

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
Τμήμα Ψηφιακών Συστημάτων



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

***Μελέτη και Ανάπτυξη Εφαρμογής Σχολικής Ηγεσίας σε περιβάλλον
Android***

Ζαπαλίδη Αγγελική, Ε/11034

Επιβλέπων Καθηγητής: Απόστολος Μηλιώνης

Πειραιάς, Σεπτέμβριος 2017



Επιτελική Σύνοψη

Στα εκπαιδευτικά περιβάλλοντα παγκοσμίως, γίνεται τα τελευταία χρόνια πολύ έντονη η ανάγκη για ικανή και επαρκή οργάνωση και διοίκηση του κάθε τέτοιου οργανισμού, καθώς αποτελεί μια εκτός από εκπαιδευτική αλλά και επιχειρησιακή μονάδα. Η σωστή διαχείριση (management) στην εκπαίδευση εξελίσσεται και δέχεται διαρκώς νέες μεθόδους διοίκησης, που βασίζονται στα δεδομένα του οργανισμού (data-based leadership). Μετά την Ανάλυση σε Ακαδημαϊκό και Γνωσιακό επίπεδο, για ανώτερες βαθμίδες εκπαίδευσης που έχουν αναπτυχθεί τα τελευταία χρόνια, μελετήθηκε και η εφαρμογή διοικητικής με βάση τα δεδομένα και στην πρωτοβάθμια εκπαίδευση, καθώς η ανάγκη εφαρμογής της σε συνολικότερο επίπεδο είναι εμφανής. Η Σχολική ηγεσία αποτελεί τον κλάδο εκπαιδευτικής ηγεσίας που εφαρμόζει τις γνώσεις και τις μεθόδους ανάλυσης δεδομένων που προϋπήρχαν, καθώς και της πολυεπίπεδης διοίκησης, στο πιο περιορισμένο αλλά ρευστό περιβάλλον της πρωτοβάθμιας εκπαίδευσης. Σκοπός είναι η συλλογή δεδομένων εντός του οργανισμού, προς εξόρυξη γνώσεων για την καλύτερη διαχείριση των πόρων, τη διατήρηση της καλής φήμης του σχολείου, την ομαλή λειτουργία του και την αποτελεσματικότητά του. Στην παρούσα εργασία αναλύσαμε την έννοια και τη σημασία της Σχολικής Ηγεσίας, και το σχεδιασμό και την υλοποίηση μιας εφαρμογής σε περιβάλλον Android που να εξυπηρετεί την πρακτική εφαρμογή των εννοιών που μελετήσαμε σε ένα πραγματικό σχολικό περιβάλλον.

Λέξεις-κλειδιά:

Σχολική ηγεσία, Εκπαιδευτική Ηγεσία, Διοίκηση Σχολικού Περιβάλλοντος, Ανάπτυξη Εκπαιδευτικού Λογισμικού, Εφαρμογή Σχολικής Ηγεσίας

Learning Analytics, Academic Analytics, School Analytics, School Complexity Leadership, Data Driven Decision Making, Android Software Development, School Leadership Android Application



ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ.....	3
ΚΕΦΑΛΑΙΟ 1^ο: ΕΙΣΑΓΩΓΗ	6
1.1 Πρόλογος.....	6
1.2 Περίληψη.....	6
1.2.1 Περιγραφή του υπό μελέτη προβλήματος	6
1.2.2 Σκοπός και στόχοι της εργασίας.....	7
1.2.3 Βασικοί ορισμοί.....	7
1.3 Παραδοτέα της Εργασίας.....	8
1.4 Δομή της Εργασίας.....	9
ΚΕΦΑΛΑΙΟ 2^ο: ΑΝΑΣΚΟΠΗΣΗ ΒΙΒΛΙΟΓΡΑΦΙΑΣ ΚΑΙ ΜΕΛΕΤΗ ΘΕΩΡΙΑΣ	11
2.1 Ανάλυση Δεδομένων.....	11
2.1.1 Ακαδημαϊκή Ανάλυση	11
2.1.2 Μαθησιακή Ανάλυση	12
2.2 Πολύπλοκα Προσαρμοστικά Συστήματα	13
2.2.1 Ηγεσία βάσει Δεδομένων – Data Driven Decision Making	13
2.2.2 Το Πολύπλοκο Προσαρμοστικό Σύστημα	14
2.2.3 Ηγεσία Πολυπλοκότητας.....	15
2.3 Σχολική Ηγεσία	16
2.1.3 Σχολική Ανάλυση	16
2.3.2 Σχολική Ηγεσία – Το σχολείο ως Πολύπλοκο Προσαρμοστικό Σύστημα.....	17
2.4 Η Τεχνολογία στην υπηρεσία της Εκπαίδευσης.....	21
ΚΕΦΑΛΑΙΟ 3^ο: ΣΧΕΤΙΚΑ ΜΕ ΤΟ ANDROID	23
3.1 Ιστορικά.....	23
3.2 Η τεχνολογία.....	24
3.2.1 Components Εφαρμογής.....	24
3.2.2 Κύκλος Ζωής Activity	24
3.2.3 Layouts.....	27
3.2.4 Views	28
3.3 Χρησιμότητα.....	31
3.4 Το Android Studio και η ανάπτυξη της εφαρμογής μας	32
ΚΕΦΑΛΑΙΟ 4^ο: ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ.....	33
4.1 Σκοπός και Χρησιμότητα της Εφαρμογής	33



4.2 Ανάλυση Λειτουργικών Απαιτήσεων	33
4.3 Ανάλυση Μη Λειτουργικών Απαιτήσεων.....	34
ΚΕΦΑΛΑΙΟ 5^ο: ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ – PARSE SERVER AND DATABASE	36
5.1 Αρχικός Σχεδιασμός Βάσης Δεδομένων.....	36
5.2 Parse Server – Υλοποίηση BackEnd.....	37
ΚΕΦΑΛΑΙΟ 6^ο: ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ	41
6.1 Οπτική Σχεδίαση - Mock Up design.....	41
6.2 Λειτουργική Σχεδίαση	46
6.3 Διάγραμμα Πιθανών διαδρομών Activities	47
ΚΕΦΑΛΑΙΟ 7^ο: ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ	48
ΚΕΦΑΛΑΙΟ 8^ο: ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΗΣ	55
ΚΕΦΑΛΑΙΟ 9^ο: ΣΥΜΠΕΡΑΣΜΑΤΑ	65
ΒΙΒΛΙΟΓΡΑΦΙΑ	68
ΠΑΡΑΡΤΗΜΑ – Πηγαίος Κώδικας της Εφαρμογής.....	71

ΠΙΝΑΚΑΣ ΕΙΚΟΝΩΝ

Εικόνα 1 - Μαθησιακή Ανάλυση	12
Εικόνα 2 - Μοντέλο Πολύπλοκου Προσαρμοστικού Συστήματος	15
Εικόνα 3 - Σχολική Ηγεσία	16
Εικόνα 4 - Η Συνεργασία όλων των Παραγόντων του Οργανισμού είναι αναγκαία για τη σωστή Σχολική Ηγεσία	18
Εικόνα 5 - Internet of Things - Δυνατότητες στη Σχολική Ηγεσία.....	21
Εικόνα 6: Το λογότυπο του Android.....	23
Εικόνα 7: Κύκλος ζωής activity.....	26
Εικόνα 8: Εντολή Intent.....	27
Εικόνα 9: ScrollView	29
Εικόνα 10 Η εφαρμογή Gmail αποτελεί χαρακτηριστικό παράδειγμα RecyclerView	30
Εικόνα 12: Το λογότυπο του Android Studio	32
Εικόνα 14 - Σχήμα Βάσης δεδομένων	36
Εικόνα 15 - Δημιουργία βάσης στο mLab	37
Εικόνα 16 - Parse Database Dashboard	38
Εικόνα 17 - Mock-up αρχικής οθόνης	41
Εικόνα 18 - Mock-up οθόνης Μαθημάτων	41
Εικόνα 19 - Mock-up οθόνης Ημερολογίου	42
Εικόνα 20 - Mock-up οθόνης Βιβλιοθήκης.....	42
Εικόνα 21 - Mock-up οθόνης Ανακοινώσεων	42
Εικόνα 22 - Mock-up οθονών Βαθμών.....	43



Εικόνα 23 - Mock-up οθονών Απουσιών.....	43
Εικόνα 24 - Mock-up οθόνης Συχνών Ερωτήσεων	44
Εικόνα 25 - Mock-up οθόνης Στοιχείων Επικοινωνίας.....	44
Εικόνα 26 - Mock-up Αρχικής οθόνης Μαθήματος.....	44
Εικόνα 27 - Mock-up επιμέρους οθονών Μαθήματος	45
Εικόνα 28 - Mock-up οθονών Feedback.....	45
Εικόνα 29 - Διάγραμμα Use case.....	46
Εικόνα 30 - Διάγραμμα Πιθανών Διαδρομών – Wireframing	47
Εικόνα 31 - Αρχική Οθόνη Εφαρμογής.....	55
Εικόνα 32 - Εγγραφή Χρήστη	56
Εικόνα 33 - Είσοδος Χρήστη	56
Εικόνα 34 - Το κύριο μενού	57
Εικόνα 35 - Τα μαθήματα	57
Εικόνα 36 - Οθόνη μαθήματος.....	57
Εικόνα 37 - Βιβλίο Ύλης - Εικόνα 38 - Εργασίες.....	58
Εικόνα 40 - Το κουμπί Βιβλίο παραπέμπει στο browser	58
Εικόνα 39 - Κουίζ.....	58
Εικόνα 42 - Η Βιβλιοθήκη μου.....	59
Εικόνα 41 - Τα προφίλ ανά ρόλο χρήστη	59
Εικόνα 43 - Βαθμοί Καθηγητή	60
Εικόνα 44 - Βαθμοί Μαθητή.....	60
Εικόνα 45 - Πίνακας Ανακοινώσεων καθηγητή και μαθητή	61
Εικόνα 46 - Απουσίες Καθηγητή	61
Εικόνα 47 - Απουσίες Μαθητή	62
Εικόνα 48 - Το ημερολόγιο καθηγητή – μαθητή.....	62
Εικόνα 49 - Συχνές Ερωτήσεις - Εικόνα 50 - Στοιχεία Επικοινωνίας.....	63
Εικόνα 51 - Ανενεργή επιλογή Feedback	63
Εικόνα 52 - Ερωτήσεις Feedback Μαθητή	64

ΠΙΝΑΚΑΣ ΠΙΝΑΚΩΝ

Πίνακας 1- Θεμελιώδεις Λειτουργίες Σχολικής Ηγεσίας	20
Πίνακας 2 - Μερικά από τα Βασικά views του Android.....	29
Πίνακας 3 - Η Βάση Δεδομένων μας στο Parse.....	40



ΚΕΦΑΛΑΙΟ 1^ο: ΕΙΣΑΓΩΓΗ

1.1 Πρόλογος

Η παρούσα εργασία αποτελεί το έντυπο παραδοτέο μέρος της Πτυχιακής Εργασίας με θέμα τη μελέτη της Σχολικής Ηγεσίας, την εξόρυξη γνώσης προς εξαγωγή συμπερασμάτων για τη διοίκηση του σχολικού και εκπαιδευτικού περιβάλλοντος, καθώς και την ανάπτυξη λογισμικού που θα εφαρμόζει στην πράξη τις θεωρητικές έννοιες που μελετήθηκαν. Η εργασία αυτή πραγματοποιήθηκε στα πλαίσια της εκπόνησης της Πτυχιακής εργασίας της φοιτήτριας του τμήματος Ψηφιακών Συστημάτων του Πανεπιστημίου Πειραιά, Ζαπαλίδη Αγγελικής, υπό την επίβλεψη του Επίκουρου Καθηγητή του τμήματος, κ. Απόστολου Μηλιώνη.

1.2 Περίληψη

Η παρούσα πτυχιακή εργασία περιλαμβάνει δύο σκέλη: τη θεωρητική μελέτη επί του αντικείμενου της Σχολικής Ηγεσίας, και την πρακτική εφαρμογή της μελέτης αυτής μέσα από την κατασκευή ενός Android application. Η Σχολική Ηγεσία αποτελεί έναν κλάδο της Εκπαιδευτικής Ηγεσίας, δηλαδή της επιστήμης της Διοικητικής Εκπαιδευτικού Περιβάλλοντος μέσω συλλογής και ανάλυσης δεδομένων, που επικεντρώνεται συγκεκριμένα σε περιβάλλον πρωτοβάθμιας εκπαίδευσης. Για το πρακτικό σκέλος της εργασίας, αναπτύχθηκε μια εφαρμογή σχολικής ηγεσίας σε native Android με χρήση Android Studio.

1.2.1 Περιγραφή του υπό μελέτη προβλήματος

Η Εκπαιδευτική Ηγεσία αποτελεί κλάδο της διοικητικής οργανισμών, και εφαρμόζεται σε εκπαιδευτικούς οργανισμούς και περιβάλλοντα. Βασίζεται στη συλλογή δεδομένων και την Ακαδημαϊκή και Μαθησιακή Ανάλυση, με σκοπό την εξαγωγή συμπερασμάτων και την εξόρυξη γνώσεων για την καλύτερη διαχείριση των πόρων του οργανισμού, την αποτελεσματική και ομαλή λειτουργία του και τη διατήρηση της καλής του φήμης. Προφανώς, ο κλάδος αυτός επεκτείνεται και στους οργανισμούς πρωτοβάθμιας εκπαίδευσης. Η παρούσα εργασία εξετάζει τη σημασία της Σχολικής Ηγεσίας και την εφαρμογή της στο σχολικό περιβάλλον, που διαρκώς εξελίσσεται και προσαρμόζεται σε νέες παραδοχές, απαιτήσεις και συστήματα. Σκοπός είναι η μελέτη των εννοιών και των απαιτήσεων της Σχολικής Ηγεσίας,



καθώς και της αποτελεσματικότητας της σε ένα σχολικό περιβάλλον. Επίσης, μελετώνται οι δυνατότητες χρήσης της τεχνολογίας και του διαδικτύου στην εκπαίδευση, αλλά και στην ίδια τη Σχολική Ηγεσία. Στη συνέχεια, με βάση τις γνώσεις που αποκομίσθηκαν από τη μελέτη αυτή, θα γίνει μια προσπάθεια πρακτικής εφαρμογής της θεωρητικής μελέτης με το σχεδιασμό, την ανάπτυξη και την υλοποίηση μιας εφαρμογής Android που θα μπορεί να χρησιμοποιείται πλήρως στο σχολικό περιβάλλον, να συνεισφέρει στην εκπαιδευτική διαδικασία με τις πιο σύγχρονες μεθόδους της και να καλύπτει τις ανάγκες συλλογής δεδομένων για τη Σχολική Ηγεσία, τουλάχιστον σε πρωτογενές επίπεδο.

1.2.2 Σκοπός και στόχοι της εργασίας

Οι στόχοι της εργασίας περιλαμβάνουν τα ακόλουθα:

1. Η μελέτη των διαφόρων εννοιών και επιπέδων της Σχολικής Ηγεσίας.
2. Η θεωρητική μελέτη της θεωρίας που αφομοιώθηκε σε ένα εκπαιδευτικό περιβάλλον.
3. Η υλοποίηση και πρακτική μελέτη της γνώσης που αποκομίσθηκε μέσω της κατασκευής μιας εφαρμογής Σχολικής ηγεσίας σε λογισμικό Android.

1.2.3 Βασικοί ορισμοί

- **Μαθησιακή Ανάλυση:** Κατανόηση και βελτιστοποίηση της διαδικασίας της μάθησης, εξορύσσοντας δεδομένα από την παρακολούθηση της προόδου του μαθητή, με σκοπό τη βελτίωση της διδασκαλίας (micro layer).
- **Ακαδημαϊκή Ανάλυση:** Διαχείριση των ιδρυμάτων ανωτάτης εκπαίδευσης, για την επιχειρησιακή διοικητική του οργανισμού (meso layer).
- **Σχολική Ανάλυση:** Συνδυασμός των δύο παραπάνω, συνολικό πολυεπίπεδο πλαίσιο ανάλυσης με σκοπό την ανάλυση τόσο των επιχειρησιακών δεδομένων όσο και των εκπαιδευτικών/μαθησιακών, με σκοπό την παροχή feedback loops στη σχολική ηγεσία.
- **Εκπαιδευτική Ηγεσία:** Διοικητική που βασίζεται στη λήψη αποφάσεων βάσει δεδομένων, στην εκπαίδευση.



- **Σχολική Ηγεσία:** Εκπαιδευτική Ηγεσία που εφαρμόζεται στην πρωτοβάθμια εκπαίδευση (K-12 schools), με συνδυασμό συλλογής δεδομένων από την Ακαδημαϊκή και Μαθησιακή Ανάλυση (Σχολική Ανάλυση).
- **Πολύπλοκο Προσαρμοζόμενο σύστημα (Complex Adaptive System):** Σύστημα που αποτελείται από επιμέρους μονάδες ή πράκτορες που αλληλεπιδρούν μέσα σε αυτό και συνεργάζονται, ενώ το σύστημα εξελίσσεται, βελτιώνεται και προσαρμόζεται με σκοπό την επίτευξη της βέλτιστης δυνατής λειτουργίας του συνολικά.
- **Ηγεσία Πολυπλοκότητας (Complexity Leadership):** Ηγεσία όπου η στρατηγική σχεδίαση και τα αποτελέσματα δεν προκύπτουν αποκλειστικά από τη σχεδίαση μόνο ενός πράκτορα (διαχειριστής - administrative leadership) αλλά και από απρόβλεπτες δράσεις και διαδράσεις μεταξύ όλων των πρακτόρων και του εξωτερικού περιβάλλοντος (adaptive leadership).
- **Android:** Λειτουργικό σύστημα έξυπνων κινητών, tablets και άλλων συσκευών, που χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής στο πρακτικό σκέλος της εργασίας.

1.3 Παραδοτέα της Εργασίας

Στην ενότητα αυτή αναφέρουμε επιγραμματικά τα παραδοτέα της εργασίας. Με τον όρο παραδοτέα εννοούμε οτιδήποτε συμπεριλαμβάνεται στην εργασία. Τα παραδοτέα της συγκεκριμένης εργασίας είναι:

1. Το παρόν έντυπο κείμενο της πτυχιακής εργασίας, το οποίο περιλαμβάνει την επισκόπηση της σχετικής βιβλιογραφίας, τα αποτελέσματα της μελέτης που πραγματοποιήθηκε, τα συμπεράσματα που προέκυψαν από τη μελέτη αυτή, η περιγραφή του σχεδιασμού και της υλοποίησης του συνοδευτικού λογισμικού .
2. Το λογισμικό που υλοποιήθηκε, η συνοδευτική εφαρμογή, η απομακρυσμένη βάση δεδομένων που αναπτύχθηκαν βοηθητικά στο πλαίσιο της εργασίας σας.
3. Η συλλογή πηγών και σχετικής βιβλιογραφίας για τη δημιουργία μίας βάσης γνώσης σχετικά με το θέμα.



1.4 Δομή της Εργασίας

Στο 1^ο κεφάλαιο της εργασίας, γίνεται μια εισαγωγή στο γενικότερο θέμα το οποίο μελετήθηκε κατά την εκπόνηση της εργασίας αυτής, και διασαφηνίζονται συνοπτικά ορισμένες έννοιες που θα αναλυθούν περισσότερο σε επόμενες ενότητες. Επιπλέον, περιγράφονται τα περιεχόμενα, τα παραδοτέα και η δομή της εργασίας.

Στο 2^ο κεφάλαιο γίνεται η ανασκόπηση της βιβλιογραφίας που μελετήθηκε και η μελέτη της θεωρίας στην οποία βασίστηκε η εργασία αυτή. Παρουσιάζεται η γνώση που αποκομίστηκε από τη μελέτη αυτή, και αναλύονται με περισσότερη σαφήνεια οι διάφορες έννοιες και ορισμοί.

Στο 3^ο κεφάλαιο γίνεται μια συνοπτική παρουσίαση του λειτουργικού συστήματος Android, της πλατφόρμας όπου υλοποιήσαμε το πρακτικό μέρος της εργασίας, την εφαρμογή Σχολικής Ηγεσίας.

Στο 4^ο κεφάλαιο παρουσιάζεται ο Σχεδιασμός της Βάσης Δεδομένων της Εφαρμογής, αλλά και οι πλατφόρμες στις οποίες στηρίζεται το Back-end μέρος του λογισμικού που υλοποιήθηκε.

Στο 5^ο κεφάλαιο γίνεται η ανάλυση απαιτήσεων και η μελέτη λειτουργικότητας της εφαρμογής Σχολικής Ηγεσίας, με βάση τις έννοιες που μελετήθηκαν στο 2^ο κεφάλαιο.

Στο 6^ο κεφάλαιο παρουσιάζεται ο σχεδιασμός της ίδιας της εφαρμογής, η λειτουργική σχεδίαση με use case diagrams, η οπτική σχεδίαση και η διάταξη με χρήση mock-up designs και η διασύνδεση των διαφόρων activities που τη συνθέτουν (wire framing).

Στο 7^ο κεφάλαιο παρατίθεται ο κώδικας της εφαρμογής, με περιγραφή των λειτουργιών του ανά activity.

Στο 8^ο κεφάλαιο της εργασίας, κατασκευάστηκε ένα εγχειρίδιο χρήσης, βάσει μιας περίπτωσης χρήσης (use case), που περιλαμβάνει πρακτικά παραδείγματα ανά ρόλο χρήστη, στη λειτουργία της εφαρμογής στην πράξη.



ΚΕΦΑΛΑΙΟ 2^ο: ΑΝΑΣΚΟΠΗΣΗ ΒΙΒΛΙΟΓΡΑΦΙΑΣ ΚΑΙ ΜΕΛΕΤΗ ΘΕΩΡΙΑΣ

2.1 Ανάλυση Δεδομένων

2.1.1 Ακαδημαϊκή Ανάλυση

Με τον όρο Ακαδημαϊκή Ανάλυση αναφερόμαστε στη λήψη λειτουργικών αποφάσεων βάσει δεδομένων σε Ανώτατα Εκπαιδευτικά συστήματα, παρέχοντας στα ηγετικά στελέχη τους υποστήριξη στη διαχείριση των λειτουργιών μεσαίου επιπέδου (meso institutional layer) του ιδρύματος. Οι λειτουργίες αυτές αφορούν λήψεις αποφάσεων σε θέματα διοικητικής, επιχειρησιακών και οικονομικών. (Sampson, School Analytics - A Framework for Supporting School Complexity Leadership, 2015)

Η Ακαδημαϊκή Ανάλυση περιλαμβάνει τα εξής αντικείμενα:

- Μαθητική Διαχείριση (Student Management – AA-O1): Διευκόλυνση της Ηγεσίας της Ανώτατης εκπαίδευσης στη διαχείριση δεδομένων των μαθητών τους μέσω δημογραφικών, προηγούμενων ακαδημαϊκών καταχωρήσεων, καταστάσεων εγγραφών σε μαθήματα και δραστηριότητες, και διακρίσεων που τους δόθηκαν. Έτσι προσφέρονται στην ηγεσία προτάσεις βάσει προβλέψεων πάνω στο ποιοι μαθητές αποτελούν καλύτερους υποψήφιους (για το εκάστοτε θέμα) βάσει του ιστορικού τους.
- Διαχείριση Εσωτερικών Δομών (Infrastructure Management – AA-O2): Διαχείριση των διαθέσιμων δομών προς διαχείριση και ενημέρωση, βάσει ποσότητας και ποιότητας υλικών και ψηφιακών resources και των επιπέδων χρήσης τους.
- Διαχείριση Σχολής/Ιδρύματος (Faculty Management – AA-O3): Διευκολύνει την ηγεσία στην επίβλεψη και υποστήριξη της λειτουργίας της, με χρήση δημογραφικών, μελέτης ποσότητας/ποιότητας και φυσικά ακαδημαϊκής προόδου των μαθητών και στατιστικά εγγραφών τους, προς ανάλυση επιδόσεων του εκπαιδευτικού προσωπικού και βελτίωσης της διδακτέας ύλης.
- Οικονομική Διαχείριση (Financial Management – AA-O4): Παρέχει υποστήριξη αποφάσεων για δημιουργία οικονομικού πλάνου με βάση το ύψος διδάκτρων, δαπανών του ιδρύματος, της διατήρησης αλλά και εξωτερικών εισόδων και εξόδων.
- Διατήρηση Μαθητών (Student retention – AA-O5): Με βάση τα δεδομένα, βοηθά στην πιθανή έγκαιρη πρόβλεψη αποχώρησης/απεγγραφής μαθητών.

2.1.2 Μαθησιακή Ανάλυση

Η Μαθησιακή Ανάλυση αναφέρεται στη μέτρηση, συλλογή, ανάλυση και καταγραφή δεδομένων των μαθητών και των περιεχομένων αυτών που μαθαίνουν, με σκοπό την κατανόηση και βελτίωση τόσο της ίδιας της διαδικασίας της μάθησης, όσο και του περιβάλλοντος εντός του οποίου πραγματοποιείται. Όπως είναι προφανές, η Μαθησιακή Ανάλυση επικεντρώνεται σε διαφορετικό σημείο από αυτό της Ακαδημαϊκής, και πιο συγκεκριμένα στοχεύει στο *micro layer* ενός ακαδημαϊκού ιδρύματος, δηλαδή στη διδασκαλία, τη μάθηση και πρακτικής αξιολόγησης με σκοπό τη βελτίωση της διαδικασίας της μάθησης και την υποστήριξη της προόδου. Βασίζεται στα δεδομένα των μαθητών και δεν ασχολείται τόσο με το *meso layer*, λειτουργεί ωστόσο υποστηρικτικά και ως προς αυτό. Εφαρμόζεται σε όλες τις βαθμίδες και τα επίπεδα εκπαίδευσης, ακόμα και απομακρυσμένων ή ανοιχτών Online μαθημάτων, λόγω της γενικής και καθολικής της φύσης.



Εικόνα 1 - Μαθησιακή Ανάλυση

Πηγή: <https://tl.hku.hk/2017/06/improving-learning-with-data-lessons-learnt-from-a-learning-analytics-conference/>

Η Μαθησιακή Ανάλυση περιλαμβάνει τα εξής αντικείμενα:

- Μοντέλο Μαθητή (Student Modelling – LA-O1): Αποτελεσματική υποστήριξη αποφάσεων στο *micro layer* με βάση τα δεδομένα του μαθητή (δημογραφικά, συμπεριφορές, χαρακτηριστικά, προσωπικά)



- Πρόταση εκπαιδευτικών πηγών (Educational Resources Recommendation – LA-O2): Ταυτοποίηση και πρόταση εκπαιδευτικών πηγών για μαθητές και καθηγητές με βάση τα δεδομένα από το LA-O1, την αξιολόγηση και αλληλεπίδραση των μαθητών, και αντίστοιχα των καθηγητών. Έτσι προσαρμόζεται η διαδικασία και οι πηγές για τον κάθε μαθητή, και προτείνονται εκπαιδευτικά σενάρια και πηγές στον καθηγητή.
- Αξιολόγηση μαθητή (Student Assessment/Feedback – LA-O3): Αξιολογούνται οι επιδόσεις, η συμμετοχή, η συνεργασία του μαθητή, εξάγονται συμπεράσματα για την ψυχολογία του και τις εκπαιδευτικές του ανάγκες.
- Αξιολόγηση από τον καθηγητή (Teacher Feedback Provision – LA-O4): Τα ίδια δεδομένα από τη μεριά του καθηγητή, και επιπλέον η αλληλεπίδραση και επικοινωνία του με το μαθητή και η κριτική και βελτίωση των εκπαιδευτικών μεθόδων τους. (Sampson, School Analytics - A Framework for Supporting School Complexity Leadership, 2015)

2.2 Πολύπλοκα Προσαρμοστικά Συστήματα

2.2.1 Ηγεσία βάσει Δεδομένων – Data Driven Decision Making

Εκτός της «παραδοσιακής» διοικητικής ηγεσίας (administrative leadership), που βασίζεται στη λήψη αποφάσεων κατά την κρίση ενός ατόμου – διαχειριστή, έχουν εξελιχθεί πολλά συστήματα Ηγεσίας με βάση την καταγραφή δεδομένων για τη βοήθεια και υποστήριξη λήψης αποφάσεων. Επιπλέον, τα συστήματα αυτά βοηθούν και στην επιλογή κατάλληλου διευθυντή – διαχειριστή της ηγεσίας, με βάση το ιστορικό και το προφίλ του κάθε υποψήφιου. Μετά την επιλογή του υπεύθυνου ηγεσίας, διαμορφώνεται μια ομάδα ηγεσίας και περιλαμβάνονται χρήστες πληροφορίας. Είναι ζωτικής σημασίας το εκπαιδευτικό προσωπικό να συνεργάζεται πρόθυμα και αποτελεσματικά με την ομάδα υλοποίησης για συντήρησης του εκάστοτε συστήματος πληροφοριών για την ηγεσία βάσει δεδομένων. Στη συνέχεια, καθορίζονται οι στόχοι του συστήματος ηγεσίας, κατασκευάζονται οι δομές δεδομένων και η λειτουργικότητα του συστήματος το οποίο διαμορφώνεται για να γίνει εύχρηστο και να συνεργάζεται και με άλλα συστήματα. Ξεκινώντας από μια μικρή ομάδα χρηστών, η εφαρμογή του συστήματος επεκτείνεται σταδιακά σε όλο το σύνολο των χρηστών, το λογισμικό που θα χρησιμοποιείται υιοθετείται κατάλληλα και το σύστημα δοκιμάζεται ξανά και ξανά. Μπορεί να περιλαμβάνονται πολλές πλατφόρμες εφαρμογών, αρκεί να γίνεται σωστή



συλλογή και αξιολόγηση των δεδομένων. (*The Wallace Foundation - Leader Tracking Systems - Turning Data into Information in School leadership, 2017*)

Η λήψη αποφάσεων βάσει δεδομένων (Data-driven decision making - DDDM) στην εκπαίδευση αποκτά παγκόσμιο ενδιαφέρον. Πρόκειται για τη συλλογή, ανάλυση και ερμηνεία δεδομένων από το εσωτερικό του εκπαιδευτικού ιδρύματος, με σκοπό την εξαγωγή συμπερασμάτων και την εξόρυξη γνώσεων για τη σωστή ηγεσία και τις ρυθμίσεις στον εκπαιδευτικό τομέα. Τα δεδομένα προκύπτουν από τους διάφορους χρήστες – πράκτορες εντός του συστήματος, τα οποία αποκομίζονται από δύο επίπεδα: *micro layer* (το οποίο αφορά στην εκπαίδευση, τη μάθηση, την αξιολόγηση του μαθητή και του καθηγητή προς υποστήριξη καλύτερης διαδικασίας της εκπαίδευσης) και *meso layer* (το οποίο αφορά στο επιχειρησιακό κομμάτι του εκπαιδευτικού οργανισμού και στη διαχείρισή του).

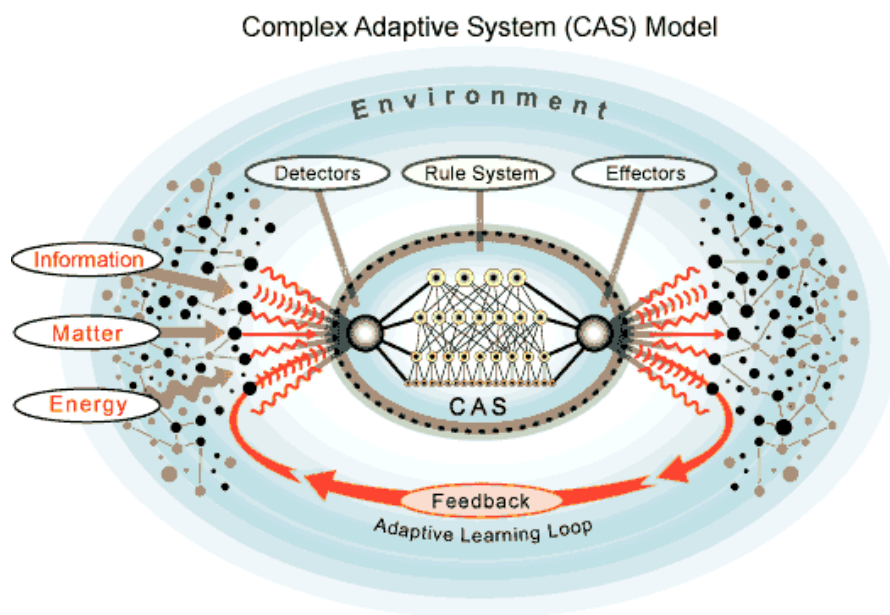
Στον τομέα της ανώτατης εκπαίδευσης, η DDDM εφαρμόζεται με τις διαδικασίες ανάλυσης που περιγράψαμε νωρίτερα, και λειτουργούν αποτελεσματικά στη σωστή εφαρμογή ηγεσίας πολυπλοκότητας, χάρη στη δημιουργία βρόχων πληροφοριών. Έτσι, η εκπαιδευτική ηγεσία προσαρμόζεται συνεχώς στις ανάγκες των εσωτερικών πρακτόρων και τα ερεθίσματα του περιβάλλοντος του κάθε ιδρύματος, είναι δυναμική και συνεχώς εξελισσόμενη. (*Sampson, School Analytics - A Framework for Supporting School Complexity Leadership, 2015*)

2.2.2 Το Πολύπλοκο Προσαρμοστικό Σύστημα

Ένα Πολύπλοκο Προσαρμοστικό Σύστημα (complex adaptive system – CAS) είναι ένα σύστημα που απαρτίζεται από πολλά επιμέρους μέρη ή πράκτορες. Οι πράκτορες αυτοί ακολουθούν μερικούς απλούς κανόνες εντός του συστήματος: δεν υπάρχει ένας ηγέτης ή διαχειριστής που να ρυθμίζει τις πράξεις των άλλων, μέσω της αυθόρμητης αλληλεπίδρασης των πρακτόρων προκύπτουν μοτίβα, και τέλος εφόσον αλλάξουν τα στοιχεία του συστήματος, τότε αυτό προσαρμόζεται ή αντιδρά στην αλλαγή. Το σύστημα αυτό δεν είναι γραμμικό, όπως προκύπτει από τους βρόχους δεδομένων μέσω *feedback*. Επίσης, ένα τέτοιο σύστημα αυτό-οργανώνεται ακολουθώντας τα μοτίβα που προκύπτουν, και φυσικά οι πράκτορες ακολουθούν τους κανόνες. Χαρακτηρίζεται επίσης από «χαοτική συμπεριφορά», καθώς μικρές

αλλαγές σε αρχικές συνθήκες μπορεί να επιφέρουν τη δημιουργία τεράστιων αλλαγών στο αποτέλεσμα του συστήματος. Όπως είναι προφανές, ένα τέτοιο σύστημα είναι επίσης στοχαστικό, καθώς υπάρχει τυχαιότητα στις πράξεις των πρακτόρων και συνεπώς στις αλληλεπιδράσεις τους.

Οι βρόχοι feedback είναι ένα πολύ σημαντικό χαρακτηριστικό των συστημάτων αυτών, καθώς με βάση τα δεδομένα τους το σύστημα μπορεί να διατηρηθεί ή να προσαρμοστεί κατάλληλα. Είναι σημαντικό να διακρίνουμε την έννοια του feedback από αυτή τη κριτικής του χρήστη ως θετική ή αρνητική. Εδώ, οι πράκτορες καλούνται να δώσουν μια κριτική στη συνολική λειτουργία του συστήματος, πιθανώς ακόμα και των άλλων πρακτόρων, αλλά και προσωπικές απόψεις σχετικά με τρόπους βελτίωσης της λειτουργίας ή αναγκαίων αλλαγών που θεωρούν ότι πρέπει να πραγματοποιηθούν. (*«What is a Complex Adaptive System?» - Project Guts*)



Εικόνα 2 - Μοντέλο Πολύπλοκου Προσαρμοστικού Συστήματος

Πηγή: datuopinion.com

2.2.3 Ηγεσία Πολυπλοκότητας

Η Ηγεσία πολυπλοκότητας (Complexity Leadership - CL) αφορά στην ηγεσία Πολύπλοκων προσαρμοστικών συστημάτων, επιτρέποντας μια διαφορετική προσέγγιση στην ηγεσία από αυτήν ενός μοναδικού πράκτορα – διαχειριστή (administrative leadership). Αντ' αυτού, η ηγεσία είναι αποτέλεσμα και των διάφορων πράξεων και



αλληλεπιδράσεων των εσωτερικών πρακτόρων, επιτρέποντας στο σύστημα να προσαρμόζεται κάθε φορά σε κάθε μικρή ή μεγάλη αλλαγή (adaptive leadership). Αυτού του είδους η ηγεσία βασίζεται όχι μόνο στη συλλογή δεδομένων και την εξαγωγή συμπερασμάτων, αλλά και στη λογική ότι η ηγεσία δεν έχει κατεύθυνση «από πάνω προς τα κάτω» ή το αντίστροφο, αφού και στις δύο αυτές περιπτώσεις δεν προκύπτουν θέματα σε σχέση με τη λήψη και υιοθέτηση αποφάσεων. Στα συστήματα αυτά, η ηγεσία βασίζεται στη λήψη αποφάσεων βάσει δεδομένων, επομένως απαιτείται η δυνατότητα και διατήρηση ροής δεδομένων στο σύστημα, και η αλληλεπίδραση των πρακτόρων. Με τη δημιουργία και χρήση των feedback loops και την προσαρμογή του συστήματος επιτυγχάνεται η ρευστότητα των δυναμικών αυτών συστημάτων και η διαρκής προσαρμογή τους στην τρέχουσα κατάσταση. Επομένως, γνωρίζουμε ότι η ηγεσία πρέπει να δίνει σημαντική βαρύτητα στη συλλογή, μοντελοποίηση και ανάλυση των αλληλεπιδράσεων και τα δεδομένα διαμέσου του οργανισμού. (Sampson, *School Analytics - A Framework for Supporting School Complexity Leadership*, 2015)

2.3 Σχολική Ηγεσία

2.1.3 Σχολική Ανάλυση

Η Ακαδημαϊκή Ανάλυση βασίζεται στην οργανωσιακή διαδικασία που στοχεύει κυρίως



Εικόνα 3 - Σχολική Ηγεσία

Πηγή: www.bcis.cn

στο meso layer των λειτουργιών ενός εκπαιδευτικού οργανισμού και σε εκπαιδευτικό επίπεδο, όπου επικεντρώνεται σε ανώτατα εκπαιδευτικά ιδρύματα. Ωστόσο, παρά την ικανή επιχειρηματική κάλυψη, δεν πραγματοποιεί επαρκή επισκόπηση σε επίπεδο λειτουργιών micro layer. Για το λόγο αυτό δεν μπορεί να επιφέρει επαρκή ηγετική δυνατότητα σε εκπαιδευτικά ιδρύματα, αφού δε λαμβάνεται υπόψη η αποτελεσματικότητα της εκπαιδευτικής διαδικασίας καθαυτής και το πόσο επηρεάζεται το meso layer από αυτήν. Επίσης, καθώς εφαρμόζεται κατά βάση σε ανώτατα εκπαιδευτικά ιδρύματα, η απευθείας εφαρμογή της στην πρωτοβάθμια εκπαίδευση δεν είναι δυνατή, ωστόσο αποτελεί μια ικανή βάση για άλλες πιθανές προσεγγίσεις ανάλυσης του επιπέδου αυτού. Όσο για τη Μαθησιακή Ανάλυση, επικεντρώνεται στο micro layer με σκοπό την αρωγή εκπαιδευτικών και



μαθητών για τη βελτιστοποίηση της αποτελεσματικότητας της διδασκαλίας και της διαδικασίας της μάθησης. Όμως οι υπάρχουσες προσεγγίσεις αυτής της μεθόδου ανάλυσης δεν καλύπτουν τις λήψεις αποφάσεων επιπέδου meso layer, και επομένως δε μπορούν να καλύψουν τις επιχειρηματικές διαδικασίες. Μπορούν όμως ως ένα βαθμό να εφαρμοστούν αποτελεσματικά στην πρωτοβάθμια εκπαίδευση, όχι όμως να καλύψουν πλήρως τις ανάγκες.

Με γνώμονα τα παραπάνω, είναι προφανές ότι απαιτείται μια νέα συνολική προσέγγιση για την υποστήριξη καθολικής και πολύπλοκης ηγεσίας για εκπαιδευτικά ιδρύματα, όπως απαιτείται για την πρωτοβάθμια εκπαίδευση όπου τα ηγετικά στελέχη αναλαμβάνουν καθήκοντα από την επίβλεψη και οργάνωση λειτουργιών επιπέδου micro layer μέχρι την περάτωση καθηκόντων επιπέδου meso layer. Όσο για τη λήψη αποφάσεων, δεν απαιτείται μόνο η συγκέντρωση των προαναφερόμενων δεδομένων αλλά και η ανάλυση τους με τρόπο που να παρέχει και βρόχους δεδομένων (feedback loops) πάνω στην απόδοση των πρακτόρων του συστήματος και να λαμβάνει υπόψη τις αλληλεπιδράσεις τους. Η λήψη τέτοιων αποφάσεων δεν καλύπτεται από καμία από τις δύο προηγούμενες Αναλύσεις επαρκώς, έτσι προέκυψε η ανάγκη δημιουργίας μιας νέας συνδυαστικής προσέγγισης που να στοχεύει στις απαιτήσεις της Ηγεσίας Πολυπλοκότητας και να παρέχει τις επιπλέον διαδικασίες. Η Σχολική Ανάλυση δημιουργήθηκε για την κάλυψη αυτών των απαιτήσεων, ακολουθώντας τις αρχές της Ηγεσίας Πολυπλοκότητας εφαρμοσμένες σε ένα σχολικό περιβάλλον. Οι στόχοι της Ανάλυσης αυτής βασίζονται στις κύριες υποχρεώσεις των Σχολικών Ηγετών και σε έναν συνδυασμό των αντικειμένων των υπάρχουσών Αναλύσεων (Ακαδημαϊκή κι Μαθησιακή) προσαρμοσμένων ώστε να υποστηρίζουν τα πολύπλοκα σχολικά περιβάλλοντα της πρωτοβάθμιας εκπαίδευσης παρέχοντας τη ροή δεδομένων εσωτερικά του συστήματος, όπως απαιτείται, για τη λήψη αποφάσεων βάσει δεδομένων και τη δημιουργία κατάλληλων στρατηγικών. (*Sampson, School Analytics - A Framework for Supporting School Complexity Leadership, 2015*)

2.3.2 Σχολική Ηγεσία – Το σχολείο ως Πολύπλοκο Προσαρμοστικό Σύστημα

Τα σχολεία και οι οργανισμοί πρωτοβάθμιας εκπαίδευσης περιγράφονται και αυτά ως Πολύπλοκα Προσαρμοζόμενα Περιβάλλοντα, καθώς λειτουργούν σαν «ζωντανά» οικοσυστήματα, που εξελίσσονται και προσαρμόζονται βάσει των αλληλεπιδράσεων

των παραγόντων τους. Ένα τέτοιο «οικοσύστημα» έχει όλα τα χαρακτηριστικά που συνθέτουν ένα CAS, αφού οι εσωτερικοί του πράκτορες αλληλεπιδρούν και παράγουν συνολικά αποτελέσματα από τις πράξεις και αλληλεπιδράσεις τους. Οι γονείς, οι μαθητές, οι εκπαιδευτικοί, οι εσωτερικές δομές του οργανισμού συνθέτουν ένα Εκπαιδευτικό Σύστημα Πολυπλοκότητας, που απαιτεί και την αντίστοιχη διαχείριση, κληρονομώντας τις αρχές των γενικών Συστημάτων Πολυπλοκότητας, και εφαρμόζοντας διαδικασίες «από πάνω προς τα κάτω», για συντονισμό των λειτουργιών και διαχείριση των resources.

Η προσαρμοστική σχολική ηγεσία, όμως, ακολουθεί συνδυαστική προσέγγιση «από πάνω προς τα κάτω» και το αντίστροφο, για μέγιστη αποτελεσματικότητα. Συγκεκριμένα, στηρίζεται στη ρευστότητα και προσαρμοστικότητα των αλληλεπιδράσεων των πρακτόρων του συστήματος. Οι δύο αυτοί κλάδοι ηγεσίας δρουν συμπληρωματικά, και μπορούν να επιφέρουν βέλτιστα αποτελέσματα αρκεί η συμπληρωματική λειτουργία τους να γίνει σωστά, μέσω της



Εικόνα 4 - Η Συνεργασία όλων των Παραγόντων του Οργανισμού είναι αναγκαία για τη σωστή Σχολική Ηγεσία

Πηγή: <http://www.nps.k12.nj.us/SOU/our-school/school-leadership-council/school-leadership-committee/>

δημιουργίας, διατήρησης και εκμετάλλευσης μηχανισμών για τη συλλογή δεδομένων μεταξύ των πρακτόρων, σε κάθε layer. Πιο συγκεκριμένα, καθιστούν αναγκαία την παρακολούθηση της κατάστασης του σχολείου μέσα από feedback loops, οι οποίοι θα έχουν άμεση επίδραση στη δράση κάθε πράκτορα του συστήματος, που με τη σειρά του αποκρίνεται και προσαρμόζεται στην αλλαγή αυτή. Με αυτό τον τρόπο, η συνολική εικόνα του συστήματος προσαρμόζεται σε κάθε εσωτερική αλλαγή, και οι συντονιστές της ηγεσίας του σχολείου μπορούν να σχεδιάσουν στρατηγικές παρατηρώντας την εξέλιξη των καταστάσεων του συστήματος.



Επομένως, στα πλαίσια της ηγεσίας πολυπλοκότητας, οι τεχνικές σχολικής ηγεσίας προσπαθούν να συγκεντρώσουν τις ουσιώδεις διαστάσεις των σχολικών λειτουργιών που επηρεάζονται από την διοικητική ηγεσία, αλλά και να υιοθετήσουν προσαρμοστική ηγεσία. Ο εντοπισμός και η παρακολούθηση των διαστάσεων αυτών για τη διαρκή λήψη δεδομένων και αξιολόγηση της κατάστασης του οργανισμού είναι οι θεμελιώδεις στόχοι των τεχνικών σχολικής ηγεσίας. (*Sampson, School Analytics - A Framework for Supporting School Complexity Leadership, 2015*)

Θεμελιώδης Λειτουργία	Σύντομη Περιγραφή Βρόχου Δεδομένων	Institutional Layer
Παρακολούθηση Μαθησιακής Διαδικασίας	Αφορά στην παρακολούθηση της διαδικασίας, περιλαμβάνοντας δεδομένα όπως τύπους διδακτικής, σχεδιασμός και εκτέλεση διαδικασιών, χρήση υλικού και εργαλείων μάθησης	micro
Αξιολόγηση Μαθησιακής Διαδικασίας	Αφορά στη χρήση δεδομένων της προηγούμενης διαδικασίας προς ανάλυση με σκοπό τη βελτίωση της μαθησιακής διαδικασίας, πχ αξιολόγηση αποτελεσματικότητας υιοθετημένων πρακτικών με βάση την απόδοση, συμμετοχή και feedback των μαθητών.	micro
Παρακολούθηση Επίδοσης Μαθητή	Περιλαμβάνει μεμονωμένα (micro) και αθροιστικά (meso) δεδομένα σχετικά με τις επιδόσεις των μαθητών, περιλαμβανομένων συμπεριφορών, απουσιών, συμμετοχής και αλληλεπίδρασης.	micro
Αξιολόγηση Επίδοσης Μαθητή	Αφορά στην εκτίμηση της επίδοσης του μαθητή βάσει των δεδομένων παρακολούθησης της μαθησιακής διαδικασίας του, διαγνωστικά και με feedback loops.	micro
Σχεδιασμός Διδακτέας Ύλης	Αφορά στην ταυτοποίηση προβλημάτων σχετικά με το υπάρχον πρόγραμμα σπουδών, και τη βελτίωσή του, βάσει feedback loops προηγούμενων διαδικασιών	meso
Διοίκηση Εκπαιδευτικού Προσωπικού	Σχετικά με την παρακολούθηση και διαχείριση του διδακτικού προσωπικού τόσο στις επιδόσεις διδασκαλίας όσο και σχετικά με τις λειτουργίες του σχολείου.	meso
Επαγγελματική Ανάπτυξη Εκπαιδευτικού Προσωπικού	Εντοπισμός ελλείψεων του διδακτικού προσωπικού, προώθηση επαγγελματικών ευκαιριών, επιλογή και	meso



	πρόσληψη νέου προσωπικού βάσει αναγκών του σχολείου.	
Ανάληψη ευθυνών ενδιαφερόμενων μερών	Δημιουργία και διατήρηση επικοινωνιών με ενδιαφερόμενους, για λήψη δικών τους feedback loops για την επίδρασή τους στο σχολείο, τόσο σε θέματα οικονομικής πολιτικής του σχολείου όσο και σε αμφίδρομη επικοινωνία καθηγητών – μαθητών και κηδεμόνων τους.	meso
Διαχείριση εσωτερικών παροχών	Παρακολούθηση και διαχείριση εσωτερικών παροχών του σχολείου, πχ εξοπλισμός	meso
Διαχείριση οικονομικών	Παρακολούθηση και οργάνωση οικονομικών του σχολείου πχ προϋπολογισμός, λογιστικά, εξωτερική χρηματοδότηση.	meso
Διαχείριση δεδομένων μαθητών	Συνολική διαχείριση δεδομένων μαθητή, δημογραφικά, δίδακτρα, ακαδημαϊκό υπόβαθρο, για διατήρηση ιστορικού και σημείο αναφοράς της αλληλεπίδρασης πρακτόρων-συστήματος, και άρα της βελτίωσης του τελευταίου.	meso

Πίνακας 1- Θεμελιώδεις Λειτουργίες Σχολικής Ηγεσίας (Core School Leadership Tasks - Sergis & Sampson, 2015a)

Λειτουργίες Ομάδας Σχολικής Ηγεσίας:

- Διευκόλυνση συμμετοχής της σχολικής κοινότητας στο σχεδιασμό και την υλοποίηση Πλάνου Βελτίωσης Σχολείου
- Ενθάρρυνση, υποστήριξη και άνοιγμα ευκαιριών συμμετοχής γονέων
- Συντονισμών δραστηριοτήτων σχετικών με τη σχεδίαση και υλοποίηση του Πλάνου Βελτίωσης του Σχολείου

Παρόλο που η ομάδα αυτή λειτουργεί συντονιστικά μόνο στην ηγεσία και όχι στην καθημερινή λειτουργία του σχολείου, έχει ασφαλώς και εκείνη ορισμένες υποχρεώσεις:

- Διευκόλυνση ανάπτυξης του Πλάνου Βελτίωσης του Σχολείου
- Παρακολούθηση, εκτίμηση και διόρθωση του Πλάνου Βελτίωσης του Σχολείου
- Ανάπτυξη πολιτικών και διαδικασιών ενίσχυσης παραγωγικότητας και επίτευξη στόχων ασφάλειας, εκπαίδευσης και συμμετοχής κηδεμόνων
- Διευκόλυνση λήψης αποφάσεων
- Διευκόλυνση αμφίδρομης επικοινωνίας
- Διαθεσιμότητα δεδομένων
- Δυνατότητα αντιμετώπισης ανησυχιών των εκπαιδευτικών και των γονέων
- Βελτίωση Σχολείου στα εξής πεδία:
 - Διδακτέα Ύλη
 - Κλίμα στο Σχολικό Περιβάλλον



- Διαχείριση και πειθαρχία τάξης
 - Αμφίδρομη επικοινωνία
 - Συμμετοχή γονέων και κηδεμόνων
 - Επιπλέον Δραστηριότητες (Extra-curricular activities)
- Συμβουλευτική υπηρεσία στη διεύθυνση, προτάσεις προϋπολογισμού, προτάσεις προσωπικού(SouthStreetSchool)

2.4 Η Τεχνολογία στην υπηρεσία της Εκπαίδευσης

Η τεχνολογία είναι μία από τις πολλές αδιαμφισβήτητες επιρροές στην εκπαίδευση σήμερα. Ζούμε στην εποχή όπου ο πλούτος των δεδομένων και η επέκταση της ανάπτυξης νέων γνώσεων προκαλεί τα εκπαιδευτικά ιδρύματα να επαναπροσδιορίσουν τις τεχνικές διδασκαλίας, αλλά και να προετοιμάσουν τους μαθητές με όλα τα εφόδια για τη μελλοντική τους εργασία. Η τεχνολογία δρα καταλυτικά, κινώντας την εκπαίδευση από ένα γραμμικό μοντέλο σε ένα νέο σύστημα συνεργασίας, ενεργό, αυτοπροσδιοριζόμενο, που βοηθά το μαθητή να αναπτύξει τη γνώση και τις δυνατότητές του. Η τεχνολογίες αυτές όχι μόνο εξάπτουν το ενδιαφέρον του μαθητή

και τον εφοδιάζουν με χρήσιμες πρακτικές γνώσεις, αλλά δημιουργούν και διαδραστικό περιεχόμενο μάθησης, ανοίγοντας νέους ορίζοντες και δυνατότητες στα μοντέλα διδασκαλίας, και κάνοντας την αφομοίωση πιο αποτελεσματική.

Με το Internet of Things επιτυγχάνεται η σύνδεση οποιωνδήποτε συσκευών και αντικειμένων σε ένα δίκτυο, προσθέτοντας δυνατότητες και ικανότητες, και όσο περισσότεροι συνδέονται, το Internet of Things γίνεται Internet of Everything, συνδέοντας ανθρώπους, διαδικασίες και δεδομένα και δημιουργώντας έτσι



Εικόνα 5 - Internet of Things - Δυνατότητες στη Σχολική Ηγεσία

Πηγή: CISCO – Education and the Internet of Everything



νέες πρακτικές δυνατότητες που δεν είχαν ληφθεί υπόψη πιο πριν! Μια τέτοια τεχνολογία θα μπορούσε να φέρει επανάσταση στην εκπαίδευση, καθώς απλά αντικείμενα θα αποτελούν πλέον κομμάτι ενός δικτύου, φέρνοντας στο προσκήνιο νέες ιδέες, δυνατότητες και γνώσεις. Και φυσικά, συγκεντρώνοντας δεδομένα πολύτιμα για τη στρατηγική της σχολικής ηγεσίας. Επιπλέον όμως, μπορεί να διατηρεί δεδομένα για την ψυχική διάθεση των μαθητών, την ψυχολογική τους κατάσταση, την παρακολούθηση ύπαρξης αρνητικών φαινομένων στο σπίτι ή στο σχολείο, ακόμα και της διατροφής και φυσικής κατάστασης του μαθητή!

Όταν η υιοθέτηση τεχνολογιών στη μάθηση γίνει ενεργά σε περισσότερα εκπαιδευτικά ιδρύματα, επιτρέποντας και την μεταξύ τους σύνδεση, την ανταλλαγή δεδομένων και τεχνολογιών εντός και εκτός τάξης, οι δυνατότητες θα είναι απεριόριστες. Και η δύναμη και πολυτιμότητα τεχνολογιών όπως το ΙοΕ θα γίνουν εμφανείς.

Οι τέσσερις «πυλώνες» της τεχνολογίας αυτής είναι: οι άνθρωποι, οι διαδικασίες, τα δεδομένα και τα αντικείμενα. Συνεργαζόμενοι αλλά και ο καθένας χωριστά στη διαδικασία της μάθησης μπορούν να χρησιμοποιηθούν με μεγάλο αντίκτυπο σε αυτήν, αλλά και στη σχεδιάσή της. Μια νέα ιδέα στην εκπαίδευση έχει εμφανιστεί τα τελευταία χρόνια: η «ανεστραμμένη τάξη» όπου ο μαθητής παρακολουθεί το υλικό του μαθήματος σε βίντεο ή με άλλους τρόπους, στο σπίτι ή και αλλού, ενώ στην ίδια την τάξη συζητούνται μόνο απορίες και προβληματισμοί. (KhanAcademy) Σε ένα τέτοιο μοντέλο εκπαίδευσης, που φέρνει θεαματικές θετικές αλλαγές σε άτομα όπως εργαζόμενοι μαθητές, αλλά και στο οικονομικό σκέλος της εκπαίδευσης, πόσες θα ήταν οι δυνατότητες με την υιοθέτηση τεχνολογιών ΙοΕ; Αδιαμφισβήτητα, πολλές και πρωτοποριακές. Η αξία της υιοθέτησης έξυπνων τεχνολογιών στην εκπαίδευση, και η πιθανή απήχηση που μπορεί να φέρει στην εκπαίδευση, κάνοντας τη πιο θελκτική, συναρπαστική, δίνοντας κίνητρο στο μαθητή και δυνατότητες τελειοποίησης γνώσης σε ένα αντικείμενο, αλλά και οι πιθανές χρήσεις της στην ηγεσία του σχολικού περιβάλλοντος είναι βέβαιο ότι πρέπει να εξερευνηθούν και να μη μείνουν αναξιοποίητες. (Michelle Selinger, Ana Sepulveda, Jim Buchan - *Education and the Internet of Everything* - CISCO, 2013)

ΚΕΦΑΛΑΙΟ 3^ο: ΣΧΕΤΙΚΑ ΜΕ ΤΟ ANDROID

3.1 Ιστορικά

Το Android είναι λειτουργικό σύστημα για κινητά τηλέφωνα και tablet, το οποίο τρέχει σε πυρήνα του λειτουργικού Linux. Αρχικά, αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance. Η κατασκευή κώδικα γίνεται με γλώσσα προγραμματισμού Java, σε συνδυασμό με

ελέγχους βιβλιοθηκών λογισμικού της Google που αναπτύχθηκαν για το Android. Στην πορεία, εκτός από τις κινητές συσκευές, το λειτουργικό προσαρμόστηκε και εμπλουτίστηκε ώστε να εφαρμόζεται σε διαφορετικό περιβάλλον χρήσης, παραδείγματος χάριν για τηλεοράσεις (Android TV), αυτοκίνητα (Android Auto), ρολόγια χειρός και άλλα wearables (Android Wear) και σε άλλες ηλεκτρονικές συσκευές.

Η Google πραγματοποίησε την πρώτη παρουσίαση της πλατφόρμας Android στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, ενώ η πρώτη έκδοση ήταν διαθέσιμη για το κοινό από τις 23 Σεπτεμβρίου 2008. Καθώς πρόκειται για λογισμικό ανοιχτού κώδικα (open source) σε πολύ μεγάλο βαθμό, η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού. Το λογότυπο που υιοθετήθηκε από την εταιρεία για το λειτουργικό σύστημα Android είναι ένα ρομπότ σε χρώμα πράσινου μήλου. *(Βικιπαίδεια)*

Από την κυκλοφορία του λειτουργικού στο κοινό, έχουν κατασκευαστεί αρκετές νέες εκδόσεις, με βελτιώσεις και νέες παροχές.



Εικόνα 6: Το λογότυπο του Android

Πηγή: <https://el.wikipedia.org/wiki/Android>



3.2 Η τεχνολογία

Το Android είναι κατά βάση ένα προϊόν ελεύθερου λογισμικού, ωστόσο ένα κομμάτι της ανάπτυξης του λογισμικού συνεχίζεται σε ιδιωτικό παρακλάδι. Το λειτουργικό Android υποστηρίζει οθόνες αφής πολλαπλών σημείων, αλλά η δυνατότητα αυτή είχε κλειδωθεί σε επίπεδο πυρήνα (kernel - πιθανόν για λόγους αποφυγής παραβιάσεων των πατεντών λογισμικού της Apple στις αντίστοιχες τεχνολογίες). Το λογισμικό προσαρμόζεται σε πολλές αναλύσεις οθόνης (από VGA μέχρι 4K), και για μεγάλη ποικιλία συσκευών, που καλύπτουν όλη τη γκάμα της επιχείρησης. (Βικιπαίδεια)

3.2.1 Components Εφαρμογής

Τα βασικά περιεχόμενα λειτουργίας του λογισμικού Android είναι τα εξής:

- Οι εφαρμογές Android συντίθενται από ένα ή περισσότερα application components (activities, υπηρεσίες, παρόχους περιεχομένου, και δέκτες ευρείας φάσης)
- Κάθε component εκτελεί ένα διαφορετικό ρόλο στη συνολική λειτουργία της εφαρμογής
- Κάθε **activity** περιλαμβάνει μεταξύ άλλων, ένα αρχείο XML που περιγράφει το layout του, και ένα αρχείο Java που περιγράφει τις λειτουργίες του.
- Το αρχείο **manifest**, το οποίο πρέπει να δηλώνει όλα τα components της εφαρμογής, καθώς και τις απαιτήσεις της, όπως την ελάχιστη έκδοση λογισμικού Android και πιθανές απαιτήσεις υλικού.
- **Resources** της εφαρμογής που δεν αφορούν κώδικα (εικόνες, strings, αρχεία layout κλπ.) μπορούν να περιλαμβάνουν ορίσματα για κάθε συσκευή ή έκδοση λογισμικού, ακόμα και τη γλώσσα του χρήστη.

(EngineersGarage)

3.2.2 Κύκλος Ζωής Activity

Κατά τη διάρκεια της λειτουργίας μιας εφαρμογής, κάθε activity από το οποίο αποτελείται έχει έναν κύκλο ζωής. Κάθε activity «γεννιέται», «πεθαίνει» και «ξαναγεννιέται» όσες φορές χρειάζεται, αναλόγως με το τι απαιτεί από αυτό η εφαρμογή για να λειτουργήσει.



Τα διάφορα activities της εφαρμογής, κάθε φορά που καλούνται προστίθενται σε μια στοίβα (activity stack) που βοηθά στη διαχείρισή τους. Όταν καλείται το activity τοποθετείται στην κορυφή της στοίβας αυτής, και γίνεται η τρέχουσα διαδικασία (running activity ή active). Οι προηγούμενες διαδικασίες βρίσκονται στο παρασκήνιο, κάτω από την τρέχουσα στη στοίβα, έτοιμες να καλεστούν αν χρειαστεί, με το κουμπί «πίσω» (<) του Android.

Οι καταστάσεις ενός activity είναι οι εξής:

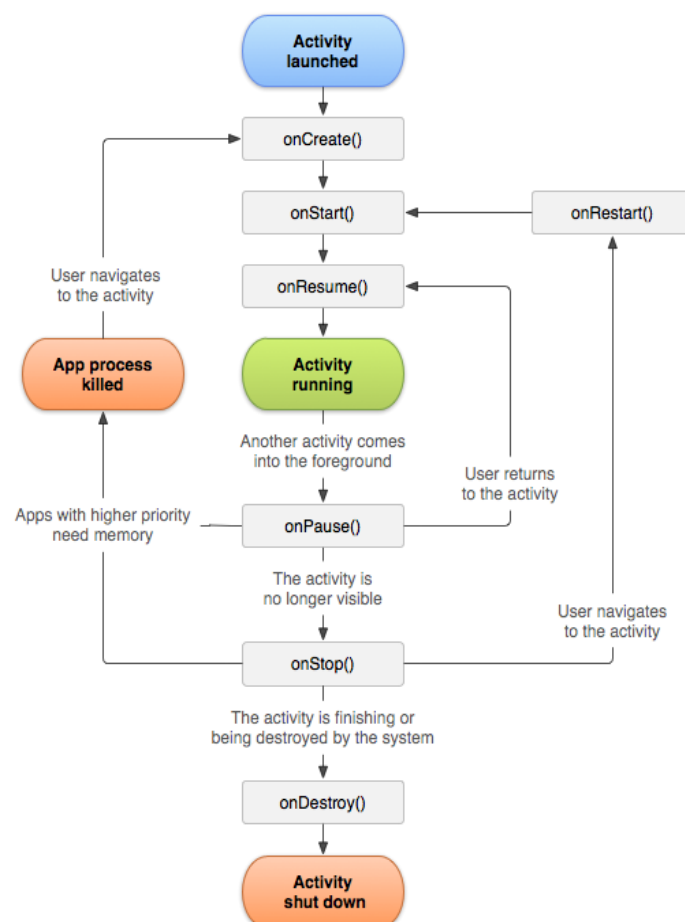
- **Running** ή **active** (τρέχον) λέγεται το activity που βρίσκεται στο προσκήνιο και εκτελείται τη συγκεκριμένη στιγμή. Πρόκειται για το activity στην κορυφή της στοίβας.
- **Paused** (σε παύση) ονομάζεται ένα activity που είναι μεν ακόμα ορατό στο χρήστη, όμως δε βρίσκεται στο προσκήνιο – δηλαδή ένα διαφανές ή μικρότερου μεγέθους activity το έχει επικαλύψει. Ένα activity σε παύση είναι ακόμα «ζωντανό», διατηρεί όλες τις πληροφορίες του και την πρωτότερη κατάστασή του και είναι έτοιμο να συνεχίσει τη λειτουργία του, ενώ δεν έχει φύγει από την οθόνη.
- **Stopped** (σταματημένο) ονομάζεται ένα activity που έχει πλήρως επικαλυφθεί από ένα άλλο. Διατηρεί την πρωτότερη κατάστασή του και τις πληροφορίες του, όμως δεν είναι ορατό στο χρήστη.
- **Dropped** ή **killed** λέγεται ένα activity που το σύστημα έδωσε εντολή να τερματιστεί ή αναγκάστηκε να «σκοτωθεί» από το σύστημα λόγω έλλειψης μνήμης. Όταν θα ξανακληθεί για να εμφανιστεί στο χρήστη, θα πρέπει να επανεκκινηθεί πλήρως και να αποκατασταθεί η προηγούμενη κατάστασή του.

Ο κύκλος ζωής ενός activity καθορίζεται από ορισμένες συγκεκριμένες εντολές. Αρχικά, κάθε activity «γεννιέται» με την κλήση της εντολής **onCreate()** και «πεθαίνει» με την κλήση της **onDestroy()**. Με την πρώτη εντολή, το activity ρυθμίζει τη συνολική του κατάσταση, ενώ με τη δεύτερη αποδεσμεύει όσα resources χρησιμοποιεί.



Το διάστημα της «ζωής» του activity όπου βρίσκεται στο προσκήνιο, καθορίζεται από την κλήση της **onStart()** μέχρι αυτήν της **onStop()**. Κατά τη διάρκεια αυτή, το activity είναι ορατή, ωστόσο μπορεί να μη βρίσκεται στο προσκήνιο (παύση) ή να μην αλληλεπιδρά με το χρήστη, αφού ενδιάμεσα μπορεί να καλούνται οι εντολές **onResume()** και **onPause()**. Αυτό το ζεύγος εντολών είναι που ορίζουν τα διαστήματα που το activity βρίσκεται στο προσκήνιο και αλληλεπιδρά πλήρως με το χρήστη. Και οι τέσσερις αυτές εντολές μπορεί να κληθούν όσες φορές θέλουμε από το σύστημα, και το activity να αλλάζει καταστάσεις για όσο χρειάζεται και να περιέχουν μεθόδους που να εκτελούν απαραίτητες για την εφαρμογή λειτουργίες, τις οποίες καλούμε πάντα σε κάποια από αυτές.

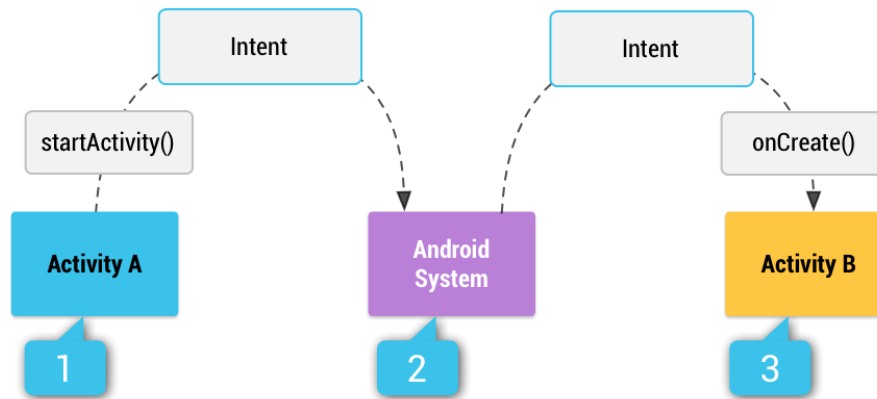
(AndroidDevelopers)



Εικόνα 7: Κύκλος ζωής activity

Πηγή: <https://developer.android.com/reference/android/app/Activity.html#ActivityLifecycle>

Ένα activity καλεί ένα άλλο μέσω της εντολής `intent(activityA.this, activityB.class)` και `startActivity()`.



Εικόνα 8: Εντολή Intent

Πηγή: <https://developer.android.com/guide/components/intents-filters.html>

3.2.3 Layouts

Κάθε βασικό layout ενός activity ονομάζεται **γονέας (parent layout)** και μπορεί να περιλαμβάνει επιμέρους layouts, τα οποία με τη σειρά τους περιέχουν widgets που ονομάζονται views. Το layout-γονέας μπορεί να είναι πάντα μόνο ένα, αλλά να περιέχει οποιονδήποτε αριθμό από εσωτερικά layouts. Τα layouts στο Android βασίζονται σε layout managers, οι οποίοι μπορεί να είναι:

- **Linear:** Γραμμική διάταξη των views, είτε οριζόντια είτε κάθετα, με βάση τον προσδιορισμό `android:orientation=""`
- **Relative:** Σχετική διάταξη των views μεταξύ τους, με βάση τα id τους. Η χρήση των id είναι πολύ χρήσιμη και για την επικοινωνία τους με το αρχείο java και τη σωστή λειτουργία της εφαρμογής.
- **Constrained:** Επιτρέπει την ιεράρχηση των views στο επίπεδο, και μπορεί να φανεί χρήσιμο στη σχεδίαση.
- **Grid:** Οργανώνει τα views σε ένα grid / table, χωρίζοντας τον «καμβά» σχεδίασης σε σειρές, στήλες και κελιά.
- **Frame:** Οργανώνει τα στοιχεία το ένα πάνω από το άλλο, για τη δημιουργία όμορφων οπτικών εφέ.

Για την ανάπτυξη της εφαρμογής μας, χρησιμοποιήθηκαν relative και linear layouts. Κάθε layout manager περιλαμβάνει διαφορετικές εντολές διάταξης, ρύθμισης και

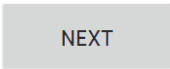


οργάνωσης των περιεχομένων του. Όλα τα layouts όμως, όπως και όλα τα περιεχόμενά τους, πρέπει οπωσδήποτε να περιγράφονται τουλάχιστον από τα attributes `layout_width` και `layout_height` που, όπως είναι ευνόητο, περιγράφουν τις διαστάσεις τους στο δισδιάστατο επίπεδο. Όλες οι υπόλοιπες εντολές, καθώς και η περιγραφή της χρήσης τους και η σωστή τους σύνταξη, μπορούν να βρεθούν στο <https://developer.android.com>, μαζί με όλο το απαραίτητο και χρήσιμο documentation που μπορεί να χρειαστεί ένας developer. Όπως προαναφέρθηκε, η περιγραφή των layouts γίνεται σε γλώσσα XML.

Επίσης, χρησιμοποιήθηκε το **DrawerLayout**, το μενού – «συρτάρι», χαρακτηριστικό του Android, για την υλοποίηση του βασικού μενού μας.

3.2.4 Views

Κάθε layout περιέχει views. Υπάρχουν πολλά είδη views, όλα όμως περικλείονται από ένα ορθογώνιο παραλληλόγραμμο στο επίπεδο του layout, και έχουν και αυτά υποχρεωτικά τιμές attributes `layout_width` και `layout_height` που περιγράφουν τις διαστάσεις τους στο δισδιάστατο χώρο. Τα βασικά είδη των views, με παραδείγματα κώδικα XML που τα περιγράφουν, είναι τα εξής:

TextView Displays text	<pre><TextView android:id="@+id/title_text_view" android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="@string/my_photos" android:textAppearance="?android:textAppearanceLarge" android:textColor="#4689C8" android:textStyle="bold" /></pre>	<h2>My Photos</h2>
ImageView Displays Image	<pre><ImageView android:id="@+id/photo_image_view" android:layout_width="match_parent" android:layout_height="match_parent" android:scaleType="centerCrop" android:src="@drawable/beach" /></pre>	
Button Button with text label	<pre><Button android:id="@+id/next_button" android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="@string/next" /></pre>	
EditText Text field that you can type into	<pre><EditText android:id="@+id/album_description_view" android:layout_width="match_parent" android:layout_height="wrap_content" android:hint="@string/album_description" android:inputType="textMultiLine" /></pre>	<h2>Album Description</h2>
Spinner Click on it to show a list of dropdown options	<pre><Spinner android:id="@+id/sort_by_spinner" android:layout_width="match_parent" android:layout_height="wrap_content" /></pre>	
CheckBox	<pre><CheckBox android:id="@+id/notify_me_checkbox" android:layout_width="wrap_content" /></pre>	<input type="checkbox"/> Notify me



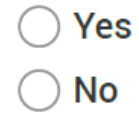
Checkbox with text label

```
android:layout_height="wrap_content"  
android:text="@string/notify_me"  
android:textAppearance="?android:textAppearanceMedium" />
```

RadioButton

Radio button (where you can select one out of a group of radio buttons)

```
<RadioGroup  
  android:layout_width="wrap_content"  
  android:layout_height="wrap_content"  
  android:orientation="vertical">  
  <RadioButton  
    android:id="@+id/yes_radio_button"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/yes"  
    android:textAppearance="?android:textAppearanceMedium" />  
  <RadioButton  
    android:id="@+id/no_radio_button"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/no"  
    android:textAppearance="?android:textAppearanceMedium" />  
</RadioGroup>
```



Πίνακας 2: Μερικά από τα Βασικά views του Android

Πηγή: Google Developers – Common Android Views - <http://labs.udacity.com/images/Common-Android-Views-Cheat-Sheet.pdf>

Ένα ακόμα, πολύ χρήσιμο είδος view είναι το **ScrollView** και το **HorizontalScrollView**, με τα οποία μπορεί να γίνει ορατό στο χρήστη όλο το περιεχόμενο, ακόμα και εφόσον δε χωράει στις διαστάσεις της οθόνης. Ένα **ScrollView**, απλό ή με οριζόντιο προσανατολισμό, μπορεί να περιέχει άμεσα μόνο ένα layout-παιδί στο εσωτερικό του, παραδείγματος χάρη ένα **linear layout** ή ένα **text view**. Σε μια περίπτωση όπως η πρώτη, το άμεσο παιδί του **ScrollView** μπορεί να περιέχει όσα στοιχεία θέλουμε στο εσωτερικό του.

```
<?xml version="1.0" encoding="utf-8"?>  
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  
  android:layout_width="match_parent"  
  android:layout_height="match_parent"  
  android:fillViewport="true"  
  android:orientation="vertical" >  
  
  <TextView  
    android:id="@+id/TextView01"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:paddingLeft="8dip"  
    android:paddingRight="8dip"  
    android:paddingTop="8dip"  
    android:text="This is a header"  
  </TextView>  
</ScrollView>
```

Παράδειγμα **ScrollView**

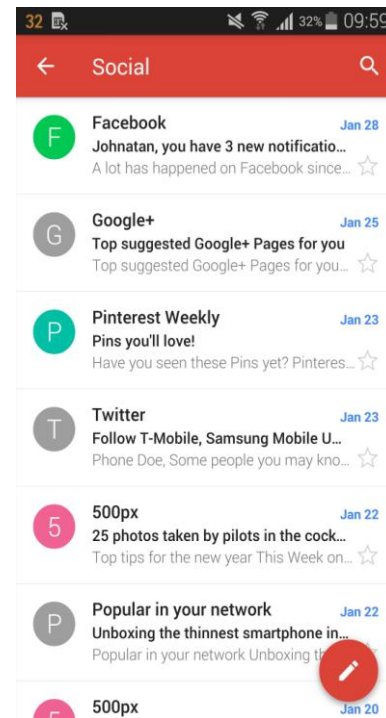


Εικόνα 9: **ScrollView**

Πηγή: Vogella - <http://www.vogella.com/tutorials/Android/article.html#resources>



Ένα πολύ χρήσιμο και σημαντικό view είναι και το **RecyclerView**. Όταν μια εφαρμογή χρειάζεται να εμφανίσει στο user-interface της μεγάλο όγκο επαναλαμβανόμενων δεδομένων ίδιου είδους, ή τύπου λίστας πολλών στοιχείων, προφανώς μόνο ένας μικρός αριθμός από αυτά μπορεί να εμφανίζεται στην οθόνη σε κάθε στιγμιότυπο χρήσης. Προηγούμενα στοιχεία προβολής δημιουργούσαν προβλήματα μνήμης ή έκαναν την εφαρμογή αργή, οπότε υπήρξε η ανάγκη δημιουργίας ενός νέου τύπου προβολής λίστας, πιο ευέλικτου αλλά και με την εξής νέα λειτουργία: η προβολή της λίστας στοιχείων γίνεται δυναμικά. Αυτό επιτυγχάνεται με μια σειρά κλάσεων και interfaces που συνθέτουν αυτή τη δυναμική διεπαφή. Υπάρχει ένα model αρχείο java, που ορίζει την κλάση αντικείμενου που συνθέτει το view, περιέχοντας διάφορα components για την προβολή των δεδομένων, ένα αρχείο XML που περιέχει το βασικό layout της κάθε γραμμής του RecyclerView, και ένα αρχείο adapter. Το αρχείο αυτό περιέχει το αντικείμενο view holder, και είναι αυτό που εκτελεί την προβολή κάθε ξεχωριστού item στο δικό του view. Το RecyclerView δημιουργεί κάθε φορά μόνον τόσους view holders όσοι είναι απαραίτητοι για την προβολή των on-screen στοιχείων συν λίγα επιπλέον. Δηλαδή, κάθε εγγραφή της λίστας αντιπροσωπεύεται πλέον από ένα αντικείμενο, στιγμιότυπο της κλάσης-μοντέλου που κληρονομεί τις ιδιότητές της, και κάθε view holder εμφανίζει ένα τέτοιο view ενός αντικείμενου. Έτσι, καθώς ο χρήστης κάνει scroll στη λίστα εγγραφών, το RecyclerView παίρνει τα off-screen views και τα επαναδεσμεύει στα δεδομένα που εισέρχονται με το scrolling στην οθόνη. (AndroidDevelopers)



Εικόνα 10 Η εφαρμογή Gmail αποτελεί χαρακτηριστικό παράδειγμα RecyclerView

Πηγή:
https://www.phonearena.com/news/How-to-delete-messages-in-Gmail-with-a-simple-swipe-Android_id65341



Παρόμοια χρήση έχει και το **Grid View**. Πρόκειται για ένα UI widget που εμφανίζει τις εγγραφές σε ένα δισδιάστατο scrollable grid (πλέγμα). (AndroidDevelopers)

Τέλος, ένα ακόμα ViewGroup που χρησιμοποιήσαμε ήταν το **CalendarView**.

3.3 Χρησιμότητα

Το Android είναι το πιο ευρέως διαδεδομένο λογισμικό στον κόσμο. Οι συσκευές με Android έχουν περισσότερες πωλήσεις από όλες τις συσκευές Windows, iOS και Mac OS X συνδυαστικά, ενώ από το 2017 ο αριθμός μηνιαίως ενεργών συσκευών που χρησιμοποιούν το λειτουργικό αυτό έχει ξεπεράσει τα δύο δισεκατομμύρια. (Android.com)

Παρά τις ανταγωνιστικές εταιρείες, το Android παραμένει πρώτο σε προτίμηση για μεγάλο μέρος του καταναλωτικού κοινού, για πολλούς λόγους. Η μεγάλη ποικιλία συσκευών και υλικού, σε μεγάλο εύρος τιμών και για κάθε προτίμηση και χρήση είναι ένας από τους βασικούς λόγους, αλλά φυσικά δεν είναι ο μόνος. Ακριβώς επειδή το λογισμικό Android δεν περιορίζεται σε μία μόνο εταιρεία και σε μία μόνο μάρκα συσκευών, είναι πολύ σύνηθες να εμφανιστούν νέες συσκευές Android στην αγορά με νέες δυνατότητες υλισμικού και επαναστατικές τεχνολογίες. Επίσης, το λειτουργικό αυτό παρέχει μεγάλη ελευθερία ρυθμίσεων και εξατομίκευσης στον κάθε χρήστη, ενώ με την πάροδο του χρόνου έγινε ένα πολύ ασφαλές και εύχρηστο σύστημα, ακόμα πιο φιλικό προς το χρήστη. Υπάρχει τεράστιος αριθμός εφαρμογών για οποιαδήποτε χρήση στο κατάστημα εφαρμογών Play Store, η τεχνολογία του Android είναι διαρκώς εξελισσόμενη, βελτιώνοντας τις υπάρχουσες παροχές της, λύνοντας τυχόν προβλήματα και προσθέτοντας νέες υπηρεσίες για το χρήστη, γεγονός που την καθιστά προτιμητέα στο καταναλωτικό κοινό, όπως φαίνεται και από τον αριθμό χρηστών Android συσκευών. (BusinessInsider)

Υπάρχουν πάνω από 706 είδη συσκευών που να υποστηρίζουν λογισμικό android, συμπεριλαμβανομένων κινητών τηλεφώνων, ψηφιακών φωτογραφικών μηχανών, media players, notebooks, tablets, υπολογιστών, gaming devices, GPS, εκτυπωτών, τηλεοράσεων και εξοπλισμού wearables, ακόμα και αυτοκινήτων. (Bluestacks)



3.4 Το Android Studio και η ανάπτυξη της εφαρμογής μας

Το Android Studio είναι το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης εφαρμογών (integrated development environment - IDE) σε λειτουργικό σύστημα Android, που κατασκευάστηκε με την υποστήριξη της JetBrains, βασίστηκε σε μεγάλο βαθμό στην «οικογένεια» IDEs της, και σχεδιάστηκε



Εικόνα 11: Το λογότυπο του Android Studio

Πηγή: <http://www.androidauthority.com/android-studio-tutorial-beginners-637572/>

ειδικά για Android development. Το πρόγραμμα αυτό είναι διαθέσιμο δωρεάν για Windows, Mac OS και Linux, και αντικατέστησε το Eclipse Android Development Tools (ADT), το οποίο μέχρι το 2014 αποτελούσε το κύριο IDE για κατασκευή λογισμικού Android. Η κυκλοφορία αυτού του νέου προγραμματιστικού περιβάλλοντος ανακοινώθηκε από την Google στις 16 Μαΐου 2013, στο Συνέδριο Google I/O, ενώ η κατασκευή του βρισκόταν ακόμα σε πρώιμο στάδιο. Η πρώτη σταθερή έκδοση του Android Studio ήταν διαθέσιμη για το κοινό το Δεκέμβριο του 2014 (έκδοση 1.0). (Wikipedia)

Το Android Studio υιοθετεί ένα νέο build system, που ονομάζεται Gradle. Το σύστημα αυτό έκανε αρκετά εύκολη και τη μετάβαση, από τη δημιουργία εφαρμογών σε Eclipse, στην ανάπτυξή τους με το νέο πρόγραμμα, καθώς και την εισαγωγή υπάρχοντων προγραμμάτων μετά από κάποια προσαρμογή. Το Gradle είναι ένα αυτοματοποιημένο σύστημα build, επίσης ανοιχτού κώδικα, το οποίο βασίζεται στα Apache Ant και Apache Maven. Επιπλέον, αντί του προσδιορισμού του project σε XML (όπως στο Apache Maven) το νέο αυτό σύστημα βασίζεται στην Groovy ως domain-specific language (DSL), ενώ η ανάπτυξη του γίνεται με γλώσσες Java και Scala, μεταξύ άλλων. Κάθε αλλαγή στο project και στα dependencies μέσω του IDE αποθηκεύεται ανάλογα και στο αρχείο build.gradle. (GitHub) Επιπλέον, για την υλοποίηση και το deployment της εφαρμογής που κατασκευάσαμε για τις ανάγκες της εργασίας, χρησιμοποιήσαμε τον virtual emulator Genymotion, καθώς και συσκευές Android που ήταν διαθέσιμες. Η υλοποίηση έγινε σε επίπεδο API 25.



ΚΕΦΑΛΑΙΟ 4^ο: ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ

4.1 Σκοπός και Χρησιμότητα της Εφαρμογής

Με βάση την αναφορά μελέτης στο paper των Sampson και Sergis, , ζητήθηκε η μελέτη και εξόρυξη πληροφοριών για την κατασκευή μιας εφαρμογής Σχολικής Ηγεσίας σε περιβάλλον Android, σε ένα αρχικό πρώτο επίπεδο. (Sampson)

Σύμφωνα με τη μελέτη αυτή, η Σχολική ηγεσία βασίζεται σε λήψη αποφάσεων βάσει δεδομένων ανάλυσης σε σχολικό επίπεδο, που περιλαμβάνει τόσο Ακαδημαϊκή όσο και Μαθησιακή ανάλυση, σε επίπεδο **micro** (γνωσιακό, εκπαιδευτικό, τάξης) και **meso** (οργανισμών εκπαίδευσης). Ο σκοπός της σχολικής ηγεσίας είναι η βελτιστοποίηση του σχολικού περιβάλλοντος τόσο όσον αφορά το μαθητή, την εκπαιδευτική διαδικασία, τις επιδόσεις, την καλή συνεργασία μαθητή-καθηγητή, την εισαγωγή μεθόδων και εξοπλισμού στην εκπαιδευτική διαδικασία, όσο και το επιχειρησιακό σκέλος ενός οργανισμού εκπαίδευσης σε θέματα οικονομικών, management και διοικητικής, πχ. κόστος εξοπλισμού, αποχώρηση μαθητών, απώλεια καλής φήμης οργανισμού.

Μελετώντας το εν λόγω paper, καταγράψαμε τις εξής απαιτήσεις μιας αρχικού σταδίου εφαρμογής σχολικής ηγεσίας, με βάση την περιγραφή ενός βέλτιστου συστήματος σχολικής ηγεσίας πρωτοβάθμιας εκπαίδευσης, όπως παρουσιάζεται στη μελέτη στην οποία βασιστήκαμε.

Όπως γνωρίζουμε, οι **απαιτήσεις** ενός συστήματος λογισμικού (στην προκειμένη περίπτωση, μιας εφαρμογής) δηλώνουν τι περιμένουμε να κάνει το σύστημα λογισμικού μας. Οι απαιτήσεις λογισμικού διακρίνονται σε **λειτουργικές** (functional) και **μη λειτουργικές** (non-functional). Οι λειτουργικές απαιτήσεις περιγράφουν τι πρέπει να κάνει η εφαρμογή μας, βάσει των δεδομένων που εισάγονται σε αυτή, και των αποτελεσμάτων που αναμένουμε να μας επιστρέφει. Οι Μη-λειτουργικές απαιτήσεις περιγράφουν ιδιότητες, περιορισμούς και υποστήριξη των λειτουργιών της εφαρμογής σε θέματα ασφαλείας, απόδοσης, χρηστικότητας κ.ά. και όχι τον τρόπο λειτουργίας της.

4.2 Ανάλυση Λειτουργικών Απαιτήσεων

- Το πρώτο άμεσα εμφανές ζητούμενο από την εφαρμογή μας είναι η καταγραφή των επιδόσεων του μαθητή, και η προβολή τους από τον καθηγητή προς εξαγωγή συμπερασμάτων και καταγραφής της προόδου του μαθητή και περαιτέρω απαιτήσεων, πχ. μαθητές με δυσλεξία.



- Παροχή online quiz για έλεγχο κατανόησης αμέσως μετά την παράδοση (με σκοπό τη βελτίωση του μαθητή και όχι το βαθμό) αλλά και σαν επιπλέον μελέτη για το σπίτι.
- Η εφαρμογή θα μπορούσε μελλοντικά να παρέχει τη δυνατότητα επικοινωνίας με το δάσκαλο άμεσα σε ειδική πλατφόρμα μηνυμάτων.
- Η εφαρμογή θα μπορούσε σε μεταγενέστερο επίπεδο να παρέχει επίσης τη δυνατότητα επικοινωνίας μεταξύ των συμμαθητών σε πλατφόρμα τύπου φόρουμ.
- Καταγραφή και αρχειοθέτηση απουσιών.
- Παροχή ατζέντας μαθήματος με βιβλίο ύλης, εργασίες για το σπίτι, προγραμματισμένα διαγωνίσματα, απώλεια μαθήματος για εκδήλωση/εκδρομή/διακοπές που θα διαχειρίζεται και θα ενημερώνει ο καθηγητής/διαχειριστής.
- Πίνακας ανακοινώσεων τάξης/σχολείου/Υπουργείου, συν ειδοποιήσεις συστήματος Android που θα διαχειρίζεται και θα ενημερώνει ο διαχειριστής.
- Πλατφόρμα εργασιών που θα οργανώνει σε ομάδες τους μαθητές και θα υποστηρίζει upload εργασιών, επεξεργασία, σχόλια και λοιπά.
- Η εφαρμογή θα απαιτεί για είσοδο authentication με e-mail γονέα ή κηδεμόνα.
- Η εφαρμογή θα παρέχει επίσης τη δυνατότητα επικοινωνίας μεταξύ καθηγητών σε ξεχωριστή πλατφόρμα.
- Αξιολόγηση καθηγητών και feedback από μαθητές και συναδέλφους.
- Σε ένα άλλο επίπεδο εφαρμογής, θα μπορούσε να προστεθεί λειτουργία καταγραφής της διάθεσης του μαθητή.
- Η εφαρμογή θα επιτρέπει ή θα εκτελεί αυτόματα Back-up και αποθήκευση δεδομένων στο Cloud ή στην απομακρυσμένη βάση δεδομένων εργασιών, υλικού μελέτης κλπ.

4.3 Ανάλυση Μη Λειτουργικών Απαιτήσεων

- Παροχή χαρτοφυλακίου μαθητή για όλα τα μαθήματα, που θα περιλαμβάνει τα σχολικά βιβλία σε PDF, καθώς και επιπλέον υλικό.
- Παρέχεται ιστορικό μαθητή από προηγούμενες τάξεις για τον δάσκαλό/καθηγητή.
- Πρέπει να υπάρχει Ειδική πλατφόρμα εργασιών.



- Μπορεί να υπάρχει Portfolio μαθητή για εργασίες/καλλιτεχνικά/πειράματα.
- Ανάθεση δικαιωμάτων επιπέδου διαχειριστή/καθηγητή/μαθητή (ελάχιστα δικαιώματα).
- Παροχή υλικού για μελέτη (διαδραστικό/βίντεο/links/παιχνίδια/επιπλέον υλικό).
- Πρέπει να υπάρχει Parental control για online συνδέσεις. (παρέχεται από τη συσκευή)
- Πρέπει να υπάρχει σελίδα FAQ, σελίδα με οδηγίες χρήσης και λοιπές πληροφορίες.
- Πρέπει να παρέχονται Στοιχεία επικοινωνίας με καθηγητή/σχολείο/υπουργείο.
- Πρέπει να υπάρχει δυνατότητα Συνδεσιμότητας με εξωτερικό εξοπλισμό, π.χ. smartboard, προτζέκτορα κ.ά. (παρέχεται από τη συσκευή)

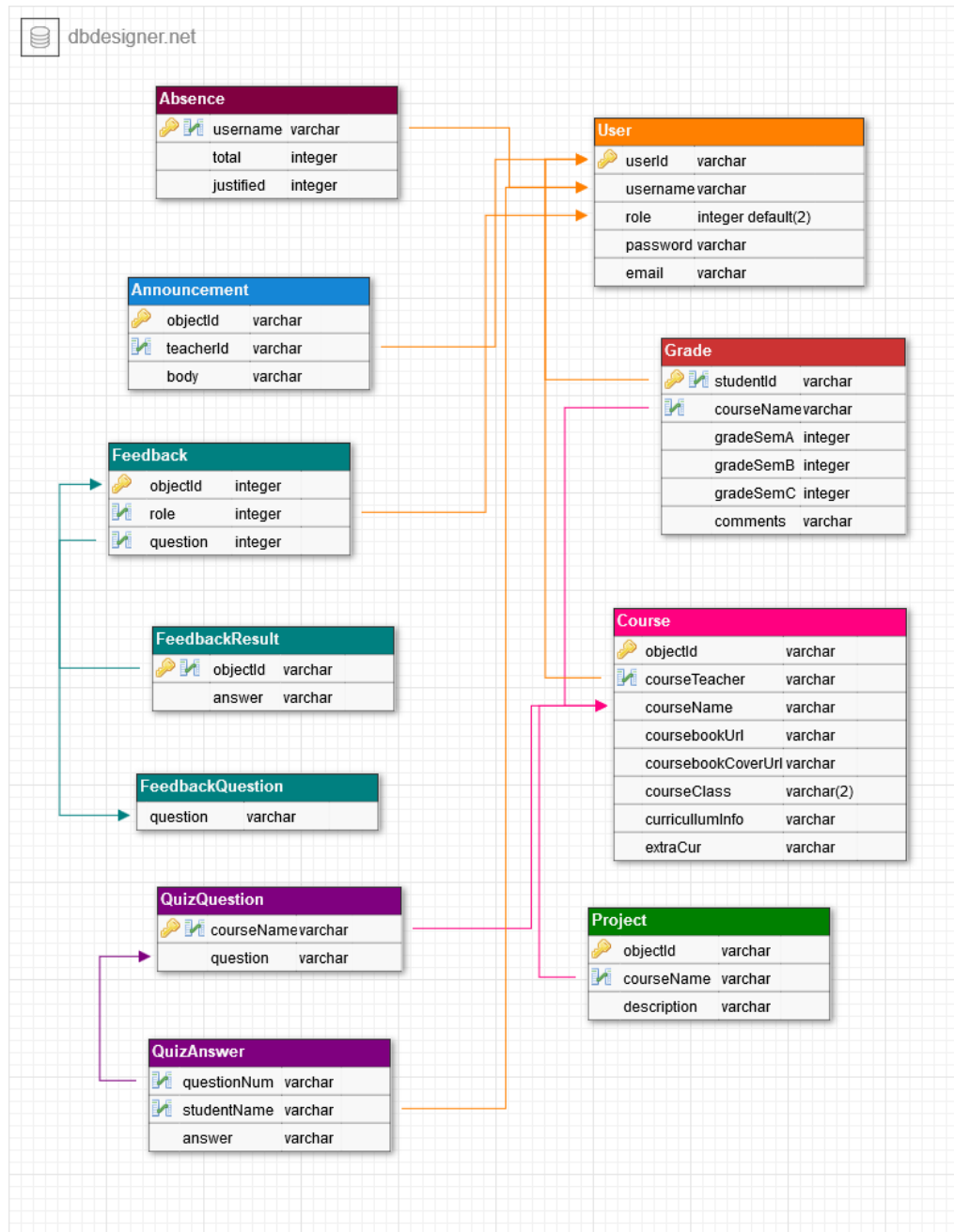
Με βάση τις παραπάνω απαιτήσεις, κινηθήκαμε σε έναν αρχικό σχεδιασμό της εφαρμογής που αναπτύξαμε στα πλαίσια της εργασίας. Προφανώς, λόγω του χρονικού περιορισμού, δεν ήταν δυνατή η υλοποίηση του 100% των απαιτήσεων, ενώ άλλες χρειάστηκε να προσαρμοστούν στις δυνατότητες του Android και του Parse. Επιπλέον, ορισμένες θα μπορούσαν να εφαρμοστούν μόνο μετά από δοκιμή σε πραγματικό χρόνο και σε περιβάλλον τάξης και σχολείου, σε επόμενο επίπεδο. Στόχος μας ήταν η αρχική σχεδίαση και υλοποίηση μιας εφαρμογής που να μπορεί να εκτελεί πρακτικά λειτουργίες σχολικής ηγεσίας, όπως τις μελετήσαμε με βάση τη βιβλιογραφία μας, προς διευκόλυνση της ομαλής λειτουργίας της τάξης και του σχολείου, και εξόρυξης δεδομένων, γνώσης και συμπερασμάτων σχετικά με τα μέλη του (καθηγητές, υπεύθυνοι, μαθητές και κηδεμόνες).



ΚΕΦΑΛΑΙΟ 5^ο: ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ – PARSE SERVER AND DATABASE

5.1 Αρχικός Σχεδιασμός Βάσης Δεδομένων

Ξεκινώντας το σχεδιασμό υλοποίησης της εφαρμογής μας, πρέπει να σχεδιάσουμε τη Βάση Δεδομένων που θα χρησιμοποιήσουμε.



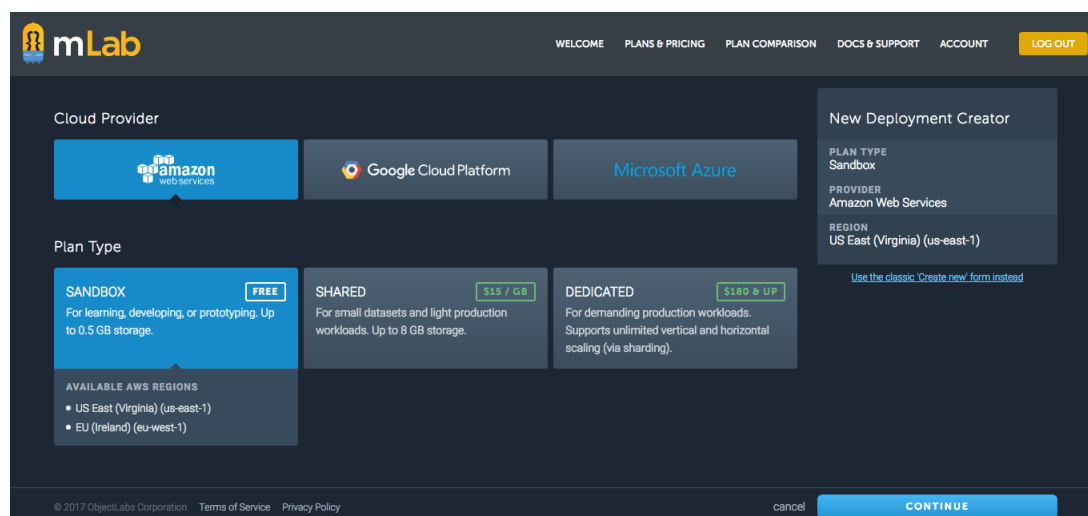
Εικόνα 12 - Σχήμα Βάσης δεδομένων

Κατασκευή Σχήματος με dbdesigner.net

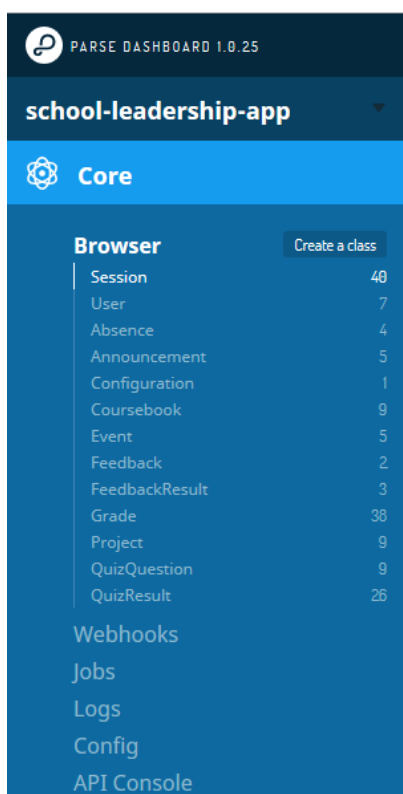


5.2 Parse Server – Υλοποίηση BackEnd

Σχετικά με το backend server της εφαρμογής, επιλέξαμε τον Parse Server, έναν open source Web Server, γραμμένο σε Node.js πάνω στο Express Framework. Προτιμήθηκε για την ευκολία χρήσης του, καθώς και το πολύ εκτενώς documented SDK για Android που παρέχει. Επιπροσθέτως, θέλαμε να έχουμε πρόσβαση στον server μας από οποιαδήποτε συσκευή, ακόμα και εκτός του τοπικού μας δικτύου. Για να επιτευχθεί αυτό, ήταν απαραίτητο να κάνουμε τον server μας deploy, σε κάποιο server host. Η πλατφόρμα Heroku φάνταζε η καλύτερη επιλογή, μιας και μας δίνει δωρεάν πλάνο, καθώς και εύκολο deploy μέσω git. Τελευταίο κομμάτι του backend, ήταν η επιλογή βάσης. Χρησιμοποιήσαμε την πιο καθιερωμένη επιλογή σε εφαρμογές Node.js, την MongoDB. Η MongoDB, είναι μια noSQL βάση δεδομένων, χωρίς ανάγκη για αυστηρά ορισμένο σχήμα. Για άλλη μια φορά, βρήκαμε δωρεάν πλάνο στο mLab.com, όπου και στήσαμε την βάση μας. Πλέον, έχοντας όλα τα κομμάτια έτοιμα, το backend μας ήταν ολοκληρωμένο. Συνδέσαμε τον server μας με την βάση μέσω της Environment Variable “MONGO_URI”, την οποία κάνουμε access από τον κώδικα του Server. Στο android, εγκαταστήσαμε το απαραίτητο gradle dependency για το Parse SDK, και μετά από το κατάλληλο setup, είχαμε συνδέσει το Android App μας με τον Parse Server μας. Για δική μας ευκολία και αναπαράσταση της βάσης, στήσαμε άλλη μια εφαρμογή στο Heroku, δίπλα από τον server μας. Αυτή ήταν ένα Parse Dashboard που συνδέσαμε με τη βάση μας, και μας επιτρέπει να κάνουμε αναπαράσταση και διαχείριση των δεδομένων της. (*ParseServer Guide*)



Εικόνα 13 - Δημιουργία βάσης στο mLab



Κατά κύριο λόγο, επεξεργαζόμαστε τη βάση μέσω του κώδικα της εφαρμογής μας, χρησιμοποιώντας το documentation ανοιχτού κώδικα, που βρίσκεται στο <http://docs.parseplatform.org>. Το Parse διαθέτει default κλάσεις αντικειμένων (πχ ParseUser, ParseObject), με τα ανάλογα χαρακτηριστικά αλλά και τις αντίστοιχες εντολές χειρισμού τους από τον κώδικα. Παραδείγματος χάριν, η κλάση χρήστη ParseUser διαθέτει πεδίο Password το οποίο αυτομάτως γίνεται hidden εντός της βάσης, καθώς και e-mail verification, και κρυφό «κλειδί» αυθεντικοποίησης χρήστη στο background, για λόγους ασφαλείας. Η κλάση αυτή και οι εντολές που τη χειρίζονται μας επιτρέπουν πολύ εύκολα να πραγματοποιήσουμε λειτουργίες εγγραφής και σύνδεσης στο σύστημα.

Εικόνα 14 - Parse Database Dashboard

Ακολουθεί η βάση δεδομένων μας:

objectid	emailVerified	ACL	updatedAt	authData	role	username	email	class
EKFBEAVkDt	(undefined)	Public Read,...	1 Oct 2017 at 20...	(undefined)	2	Μαρια Χρήστου	marxr@yahoo.com	ΣΤ
FI8BTYtF7H	(undefined)	Public Read,...	22 Sept 2017 at ...	(undefined)	2	Σοφία Παπαδοπο...	sofpap@gmail.com	ΣΤ
1uLU396b3e	(undefined)	Public Read,...	22 Sept 2017 at ...	(undefined)	2	Πέτρος Πέτρον	petrou@hotmail...	ΣΤ
WEKXij1ALr	(undefined)	Public Read,...	21 June 2017 at ...	(undefined)	2	Χρήστος Άχρηστος	axristos@gmail...	Ε
oyEiQYoaVH	(undefined)	Public Read,...	4 Oct 2017 at 02...	(undefined)	1	Χρήστος Αβαμιδης	sdds@sdsd.ds	Δ
xTNW7ucVtk	(undefined)	Public Read,...	4 Oct 2017 at 02...	(undefined)	2	Σπυρος Σπυρου	Kalinuxtakias@...	Δ
x9a0e83oF0	(undefined)	Public Read,...	6 Oct 2017 at 22...	(undefined)	0	admin	eeee@gmail.com	Δ

objectid	updatedAt	createdAt	ACL	isFeedbackVisible
d4p30G86GD	29 Sept 2017 at ...	28 Sept 2017 at ...	Public Read + Write	True

objectid	createdAt	updatedAt	ACL	total	username	justified
E1JDW78S8z	1 Oct 2017 at 01...	1 Oct 2017 at 01...	Public Read + Write	(undefined)	(undefined)	(undefined)
JLALZyeto7	7 July 2017 at 1...	7 July 2017 at 1...	Public Read + Write	(undefined)	(undefined)	(undefined)
9gyb212yPt	7 July 2017 at 1...	7 July 2017 at 1...	Public Read + Write	(undefined)	(undefined)	(undefined)
Ifi2xSKDpY	7 July 2017 at 1...	7 July 2017 at 1...	Public Read + Write	(undefined)	(undefined)	(undefined)
xFbaAxpTgy	7 July 2017 at 1...	7 July 2017 at 1...	Public Read + Write	(undefined)	(undefined)	(undefined)



CLASS										
Announcement 5 objects • Public Read and Write enabled										
objectId	String	createdAt	Date	updatedAt	Date	ACL	ACL	body	String	
<input type="checkbox"/>	E1JDWr8S8z	1 Oct 2017 at 01...		1 Oct 2017 at 01...		Public Read + Write		2 Απριλίου Παγκόσμια Ημέρα Παιδικού Βιβλίου Δευτέρα 3/4 Εκπ...		
<input type="checkbox"/>	JLALZyeto7	7 July 2017 at 1...		7 July 2017 at 1...		Public Read + Write		2 Απριλίου Παγκόσμια Ημέρα Παιδικού Βιβλίου Δευτέρα 3/4 Εκπ...		
<input type="checkbox"/>	9gyb212yPt	7 July 2017 at 1...		7 July 2017 at 1...		Public Read + Write		2 Απριλίου Παγκόσμια Ημέρα Παιδικού Βιβλίου Δευτέρα 3/4 Εκπ...		
<input type="checkbox"/>	Iffi2xSKDpY	7 July 2017 at 1...		7 July 2017 at 1...		Public Read + Write		2 Απριλίου Παγκόσμια Ημέρα Παιδικού Βιβλίου Δευτέρα 3/4 Εκπ...		
<input type="checkbox"/>	xFbaAxtpty	7 July 2017 at 1...		7 July 2017 at 1...		Public Read + Write		2 Απριλίου Παγκόσμια Ημέρα Παιδικού Βιβλίου Δευτέρα 3/4 Εκπ...		

CLASS																	
Coursebook 9 objects • Public Read and Write enabled																	
objectId	St	teacher	String	courseInfo	String	curriculumInfo	String	ACL	ACL	courseBookUrl	String	name	String	coverUrl	String	extraCur	String
3SeD67LQI8		Παπαγεωργίου		Το βιβλίο που έχ...		Unit 1. Our Mult...		Public Read + Write		http://ebooks...		ΑΓΓΛΙΚΑ ΣΤ		http://eb...		http://ebooks.	
sefttX7bHK		Άβραμίδης		Ο σκοπός της διδ...		Στο μάθημα αυτό ...		Public Read + Write		http://ebooks...		ΜΑΘΗΜΑΤΙΚΑ Ε		https://w...		http://ebooks.	
8KQV5p6xYb		Γαλανοπούλου		Ο βασικός στόχος...		ENOTHTA 1 0 φίλο...		Public Read + Write		http://ebooks...		ΓΛΩΣΣΑ Ε		http://eb...		http://ebooks.	
aBhZkISPY8		Άβραμίδης		Οι μαθητές επιδιώ...		Α ΠΕΡΙΟΔΟΣ Α Ενδ...		Public Read + Write		http://ebooks...		ΜΑΘΗΜΑΤΙΚΑ Δ		http://eb...		http://ebooks.	
zLCrmFBkWE		Γαλανοπούλου		Βασικός στόχος εί...		ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΣΤΟ...		Public Read + Write		http://ebooks...		ΓΛΩΣΣΑ Δ		http://eb...		http://ebooks.	
C4AJqLGUi3		Γαλανοπούλου		Η διδασκαλία του...		ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΣΤΟ...		Public Read + Write		http://ebooks...		ΓΛΩΣΣΑ ΣΤ		http://ec...		http://ebooks.	
2GC0JL30RA		Γαλανοπούλου		Οι μαθητές επιδιώ...		ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΣΤΟ...		Public Read + Write		http://ebooks...		ΙΣΤΟΡΙΑ ΣΤ		http://eb...		http://ebooks.	
nKPsAJHnRC		Ρέππας		Με κατάλληλες δρα...		ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΣΤΟ...		Public Read + Write		http://ebooks...		ΦΥΣΙΚΗ ΣΤ		http://eb...		http://ebooks.	
nVlIaqQUVAC		Άβραμίδης		Ο σκοπός της διδ...		Στο μάθημα αυτό ...		Public Read + Write		http://ebooks...		ΜΑΘΗΜΑΤΙΚΑ ΣΤ		http://bo...		http://ebooks.	

CLASS																			
Event 5 objects • Public Read and Write enabled																			
objectId	String	ACL	ACL	year	Num	month	Num	day	Num	updatedAt	Date	username	String	type	String	description	String	createdAt	Date
v9qF4txs3c		Public Read + Write		2017	12	6		1 Oct 2017 at 02...		Professoras		birthday		birthday		τα γενέθλια του ...		1 Oct 2017 at 02...	
iCYezD3HJq		Public Read + Write		2017	8	1		1 July 2017 at 1...		test		Diakopes		Mykonosoooooooooooo				1 July 2017 at 1...	
8ZtASZSmni		Public Read + Write		2017	1	1		1 July 2017 at 1...		test		kj		dssad				1 July 2017 at 1...	
WXo2e6jma3		Public Read + Write		2017	6	2		1 July 2017 at 1...		test		concert		bla				1 July 2017 at 1...	
5PdPr0C2pQ		Public Read + Write		2017	6	2		1 July 2017 at 1...		test		rockwave		evenescence				1 July 2017 at 1...	

CLASS															
Grade 38 objects • Public Read and Write enabled															
objectId	String	createdAt	Date	updatedAt	Date	ACL	ACL	username	courseName	String	gradeSemA	gradeSemB	gradeSemC	comment	String
<input type="checkbox"/>	pa0hGTX37V	6 Oct 2017 at 19...		6 Oct 2017...		Publi...		Επύρος...	ΙΣΤΟΡΙΑ ΣΤ		8	8	8	Καλή μνήμη, δυσκολεύε...	
<input type="checkbox"/>	QF14N9ka6B	6 Oct 2017 at 19...		6 Oct 2017...		Publi...		Επύρος...	ΑΓΓΛΙΚΑ ΣΤ		7	7	7	Δυσκολεύεται με την ξ...	
<input type="checkbox"/>	eJAWUNfNH3	6 Oct 2017 at 19...		6 Oct 2017...		Publi...		Επύρος...	ΓΛΩΣΣΑ Ε		8	8	8	Καλή έκφραση	
<input type="checkbox"/>	eeF2g1mOW2	6 Oct 2017 at 19...		6 Oct 2017...		Publi...		Επύρος...	ΓΛΩΣΣΑ ΣΤ		7	8	9	Έχει ευφρόδεια λόγου...	
<input type="checkbox"/>	KQEhMfv9o1	6 Oct 2017 at 19...		6 Oct 2017...		Publi...		Επύρος...	ΜΑΘΗΜΑΤΙΚΑ Δ		8	8	8	Τα πάει καλά με τη λο...	
<input type="checkbox"/>	vM927egE5G	6 Oct 2017 at 19...		6 Oct 2017...		Publi...		Επύρος...	ΦΥΣΙΚΗ ΣΤ		7	7	7	Δυσκολεύεται με ορισμ...	
<input type="checkbox"/>	laGuPxZT7f	6 Oct 2017 at 19...		6 Oct 2017...		Publi...		Επύρος...	ΜΑΘΗΜΑΤΙΚΑ Ε		8	8	8	Καλή εμπέδωση, δε δου...	
<input type="checkbox"/>	MdhowR0F7T	6 Oct 2017 at 19...		6 Oct 2017...		Publi...		Επύρος...	ΜΑΘΗΜΑΤΙΚΑ ΣΤ		7	7	8	Δείχνει να δυσκολεύετ...	
<input type="checkbox"/>	fKfgrjgJCO	6 Oct 2017 at 19...		6 Oct 2017...		Publi...		Επύρος...	ΓΛΩΣΣΑ Δ		8	8	8	Καλή έκφραση, κάνει ο...	
<input type="checkbox"/>	6tCAnNMGE9	6 Oct 2017 at 19...		6 Oct 2017...		Publi...		Μαρία ...	ΙΣΤΟΡΙΑ ΣΤ		9	9	9		
<input type="checkbox"/>	ZvMGwBreen	6 Oct 2017 at 19...		6 Oct 2017...		Publi...		Μαρία ...	ΑΓΓΛΙΚΑ ΣΤ		9	9	9		
<input type="checkbox"/>	iHN0C2SjJB	6 Oct 2017 at 19...		6 Oct 2017...		Publi...		Μαρία ...	ΓΛΩΣΣΑ Ε		10	10	10	Χρειάζεται περισσότερ...	
<input type="checkbox"/>	v3jXwgZjZl	6 Oct 2017 at 19...		6 Oct 2017...		Publi...		Μαρία ...	ΓΛΩΣΣΑ ΣΤ		10	10	10	Εμφανής βελτίωση από ...	
<input type="checkbox"/>	3mGS80qrin	6 Oct 2017 at 19...		6 Oct 2017...		Publi...		Μαρία ...	ΜΑΘΗΜΑΤΙΚΑ Δ		9	9	9	Εξαιρετική επίδοση - ...	
<input type="checkbox"/>	Ey0aIyraMh	6 Oct 2017 at 19...		6 Oct 2017...		Publi...		Μαρία ...	ΦΥΣΙΚΗ ΣΤ		8	8	8		
<input type="checkbox"/>	ZrjBBbuXrE	6 Oct 2017 at 19...		6 Oct 2017...		Publi...		Μαρία ...	ΜΑΘΗΜΑΤΙΚΑ Ε		9	9	9	Συμμετοχή στην τάξη, ...	
<input type="checkbox"/>	eaRvVikW576	6 Oct 2017 at 19...		6 Oct 2017...		Publi...		Μαρία ...	ΜΑΘΗΜΑΤΙΚΑ ΣΤ		9	9	10	Να παραιθεί για καμπ...	

CLASS															
Feedback 2 objects • Public Read and Write enabled															
objectId	String	ACL	ACL	role	Num	question0	String	question1	String	question2	String	question3	String	question4	String
<input type="checkbox"/>	ggVbqPzsCK	Public Read + Write		2		Πώς θα βαθμολογού...		Πώς θα βαθμολογού...		Πώς θα βαθμολογού...		Πώς θα βαθμολογού...		Ποια σχόλια έχετε...	
<input type="checkbox"/>	bRUZKAswt9	Public Read + Write		1		Πώς θα βαθμολογού...		Πώς θα βαθμολογού...		Πώς θα βαθμολογού...		Πώς θα βαθμολογού...		Ποια σχόλια έχετε...	

CLASS															
FeedbackResult 3 objects • Public Read and Write enabled															
objectId	String	ACL	ACL	updatedAt	Date	role	Num	answer0	answer1	answer2	answer3	answer4	String	createdAt	Date
<input type="checkbox"/>	UuvhZnf0vE	Public Read + Write		29 Sept 2017 at ...		2		5	4	4	3		Περισσότερη γυμνα...	29 Sept 2017 at ...	
<input type="checkbox"/>	1B5ZmI7aIa	Public Read + Write		29 Sept 2017 at ...		2		5	5	5	5		τη	29 Sept 2017 at ...	



CLASS
Project 2 objects • Public Read and Write enabled

[Add Row](#) [Refresh](#) [Filter](#) [Security](#) [Edit](#)

objectId	teacher	courseName	student_0	student1_0	student0_1	student2_0
W0kzV2LyBR	View relation	View relation	fjfgjfgj	fdsfdfsdfsd	dfgdf	fgdfgfgdgdgdg
2aIK3Uhq42	View relation	View relation	dfdf	dfsdfds	fdfsdf	dfsdfds

CLASS
QuizQuestion 9 objects • Public Read and Write enabled

[Add Row](#) [Refresh](#) [Filter](#) [Security](#) [Edit](#)

objectId	courseName	ACL	question0	question1	question2	option1_0	option2_0
9mIrk5OpVN	ΙΣΤΟΡΙΑ ΣΤ	Public Read + Write	Γράψτε μια παράγρ...	Πού πραγματοποιήθ...	Ποιος ήρωας της ...	Αστρος	Παπαφλέσσας
ryquSU8dPL	ΑΙΤΑΙΚΑ ΣΤ	Public Read + Write	Sand your pen-fr...	Where does Steve...	Where do you go ...	London, UK	The butcher's
5NIkaaZZfy	ΓΛΩΣΣΑ Ε	Public Read + Write	Τι μου αρέσει κα...	Ποια λέξη προκύπτ...	Πώς λέγεται ο επ...	ηχολήπτης	αστυνομία
LWPYe1KWdu	ΓΛΩΣΣΑ ΣΤ	Public Read + Write	Είδατε σε μια βι...	Τι περιγράφεται π...	Πώς λέγεται το α...	Το χρώμα, το σχή...	Κολοκοτρώνης
HIMx66G6r	ΜΑΘΗΜΑΤΙ...	Public Read + Write	Ποιος αριθμός είν...	Ποιο είναι το απ...	Ποιος αριθμός προ...	787	10101
XNot059D6m	ΦΥΣΙΚΗ ΣΤ	Public Read + Write	Από πού παίρνουν...	Από πού προέρχεται...	Όταν τραβάς τη χ...	Το φεγγάρι	Κινητική ενέργ
sZhub6WGYA	ΜΑΘΗΜΑΤΙ...	Public Read + Write	Γράψτε ολογράφως ...	Πόσα ψηφία έχει ...	Ποιο χρηματικό πο...	7	60 €
AndhoLULJ5	ΜΑΘΗΜΑΤΙ...	Public Read + Write	Διατυπώστε έναν κ...	Πώς λέγονται οι ...	Σε ποιον αριθμό ...	Φανταστικοί	23.679
Piyg74CvHB	ΓΛΩΣΣΑ Δ	Public Read + Write	Ποια καινούρια πρ...	Πού πέρασε τις κ...	Τι βρήκαν τα παι...	Στον Καραβόμυλο,...	Έναν ιππόκομο

CLASS
QuizResult 26 objects • Public Read and Write enabled

[Add Row](#) [Refresh](#) [Filter](#) [Security](#) [Edit](#)

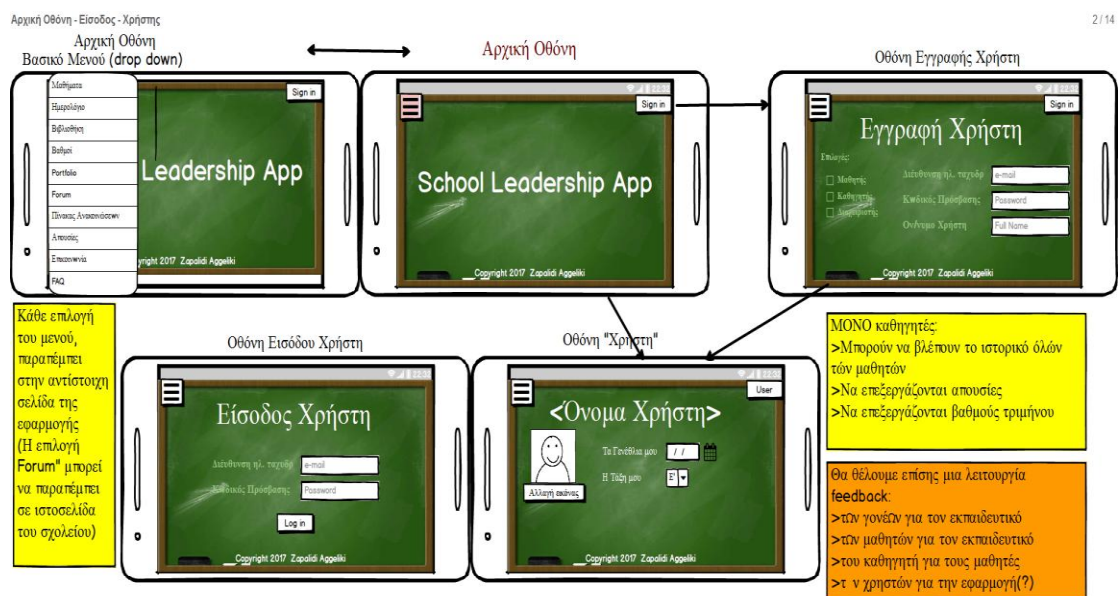
objectId	courseName	userId	ACL	updatedAt	answer0	answer1	answer2	createdAt
fbvwEUBwyD	ΑΙΤΑΙΚΑ ΣΤ	x9a0e830F0	Public Read + Write	3 Oct 2017 at 20...	η	Glaskow...	The groc...	3 Oct 2017 at 20...
8WNVb7HCIL	ΓΛΩΣΣΑ ΣΤ	x9a0e830F0	Public Read + Write	3 Oct 2017 at 12...	695	Το όνομ...	Καραϊσκάκης	3 Oct 2017 at 12...
Cs14nrR2xu	ΓΛΩΣΣΑ ΣΤ	x9a0e830F0	Public Read + Write	27 Sept 2017 at ...	5	Το όνομ...	Καραϊσκάκης	27 Sept 2017 at ...
myEzy2wBVI	ΓΛΩΣΣΑ ΣΤ	x9a0e830F0	Public Read + Write	27 Sept 2017 at ...	tt	Το όνομ...	Καραϊσκάκης	27 Sept 2017 at ...

Πίνακας 3 - Η Βάση Δεδομένων μας στο Parse

ΚΕΦΑΛΑΙΟ 6^ο: ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΠΕΡΙΓΡΑΦΗ ΕΦΑΡΜΟΓΗΣ

6.1 Οπτική Σχεδίαση - Mock Up design

Με βάση τις απαιτήσεις που εντοπίσαμε στην προηγούμενη ενότητα, προχωρήσαμε σε έναν αρχικό σχεδιασμό του user interface της εφαρμογής μας, και ορισμένων βασικών λειτουργιών της. Προφανώς, αυτό είναι ένα αρχικό προσχέδιο και κατά την υλοποίηση έγιναν αρκετές προσαρμογές. Για το σχεδιασμό του mock-up design χρησιμοποιήθηκε το σχετικό εργαλείο σχεδιασμού της Balsamiq.



Εικόνα 15 - Mock-up αρχικής οθόνης

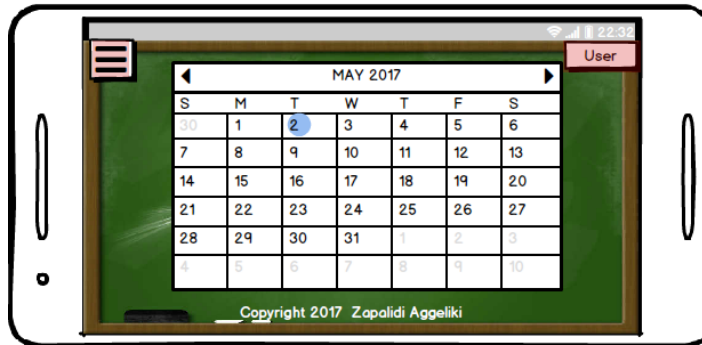


Εικόνα 16 - Mock-up οθόνης Μαθημάτων

Ημερολόγιο

4 / 14

Οθόνη "Ημερολόγιο"



Ο διαχειριστής/καθηγητής θα μπορεί να επεξεργάζεται το ημερολόγιο

Εικόνα 17 - Mock-up οθόνης Ημερολογίου

Η Βιβλιοθήκη μου

5 / 14



Περιλαμβάνει αρχεία .pdf ή συνδέσμους στα βιβλία του υπουργείου Παιδείας

Εικόνα 18 - Mock-up οθόνης Βιβλιοθήκης

Πίνακας ανακοινώσεων

9 / 14

Οθόνη "Πίνακας Ανακοινώσεων"



Ο διαχειριστής/καθηγητής θα μπορεί να επεξεργάζεται τον πίνακα ανακοινώσεων

Εικόνα 19 - Mock-up οθόνης Ανακοινώσεων

Οθόνη "Βαθμοί" μονο για καθηγητή



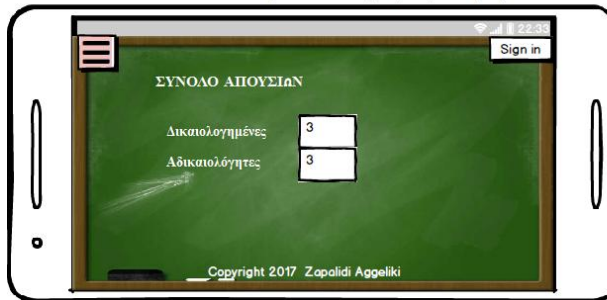
Κάθε μαθητής έχει πρόσβαση ΜΟΝΟ στους δικούς του βαθμούς. Ο καθηγητής βλέπει τους βαθμούς και το ιστορικό όλων των μαθητών, ώστε να βλέπει που χρειάζονται επιπλέον βοήθεια.

Οθόνη "Βαθμοί" μονο για μαθητή

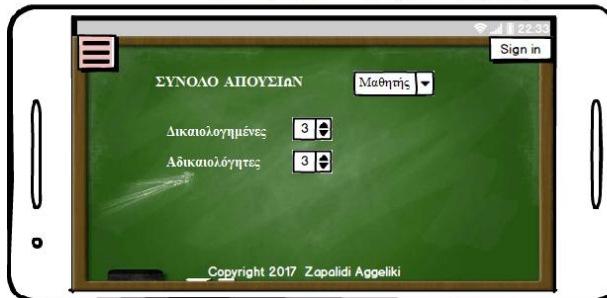


Εικόνα 20 - Mock-up οθονών Βαθμών

Οθόνη "απουσίες" για μαθητή



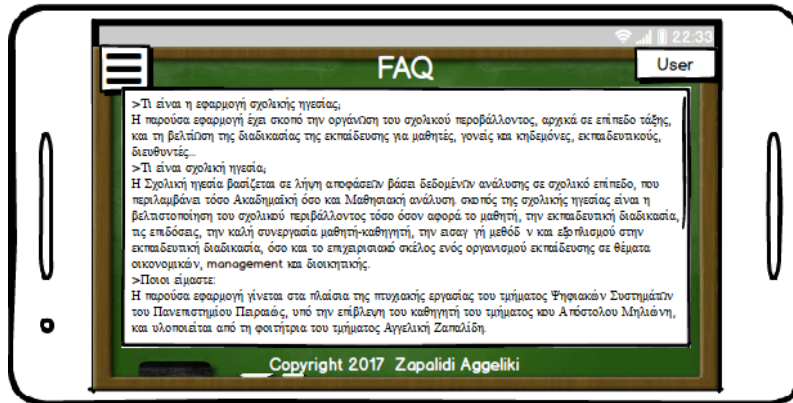
Οθόνη "απουσίες" για Καθηγητή



Read only για το μαθητή. Ο καθηγητής επεξεργάζεται τις απουσίες.

Εικόνα 21 - Mock-up οθονών Απουσιών

Οθόνη "FAQ"



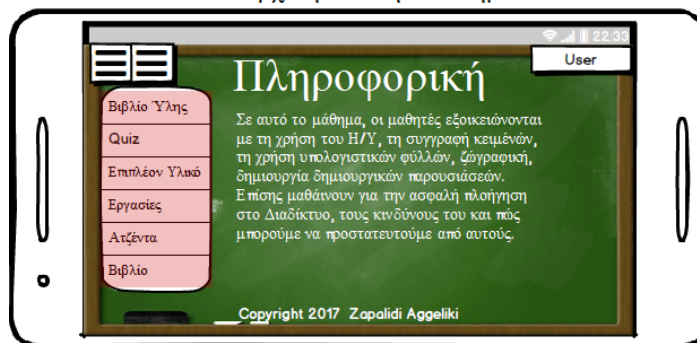
Εικόνα 22 - Mock-up οθόνης Συχνών Ερωτήσεων
Επικοινωνία

Οθόνη "Επικοινωνία"



Εικόνα 23 - Mock-up οθόνης Στοιχείων Επικοινωνίας

Αρχική Οθόνη "Μάθημα"



Κάθε μάθημα έχει τη δική του αρχική οθόνη με ένα drop down menu που θα παραπέμπει σε άλλες οθόνες σχετικές με το μάθημα.

- > Το βιβλίο Υλης, όπως ορίζεται από το Υπουργείο
- > Quiz πολλαπλών επιλογών για εξάσκηση
- > Επιπλέον Υλικό με βίντεο, links, άρθρα
- > Εργασίες, όπου θα ανακοινώνεται η εκπόνηση και (?) θα ανεβάζουν οι μαθητές
- > Ατζέντα όπου ο καθηγητής θα αναρτά πιθανή απουσία του, προγραμματισμένα διαγνώσματα, projects κλπ
- > Και παραπομπή στο βιβλίο του μαθήματος

Εικόνα 24 - Mock-up Αρχικής οθόνης Μαθήματος



Εικόνα 25 - Mock-up επιμέρους οθονών Μαθήματος

Οθόνη "Feedback" μονο για καθηγητη



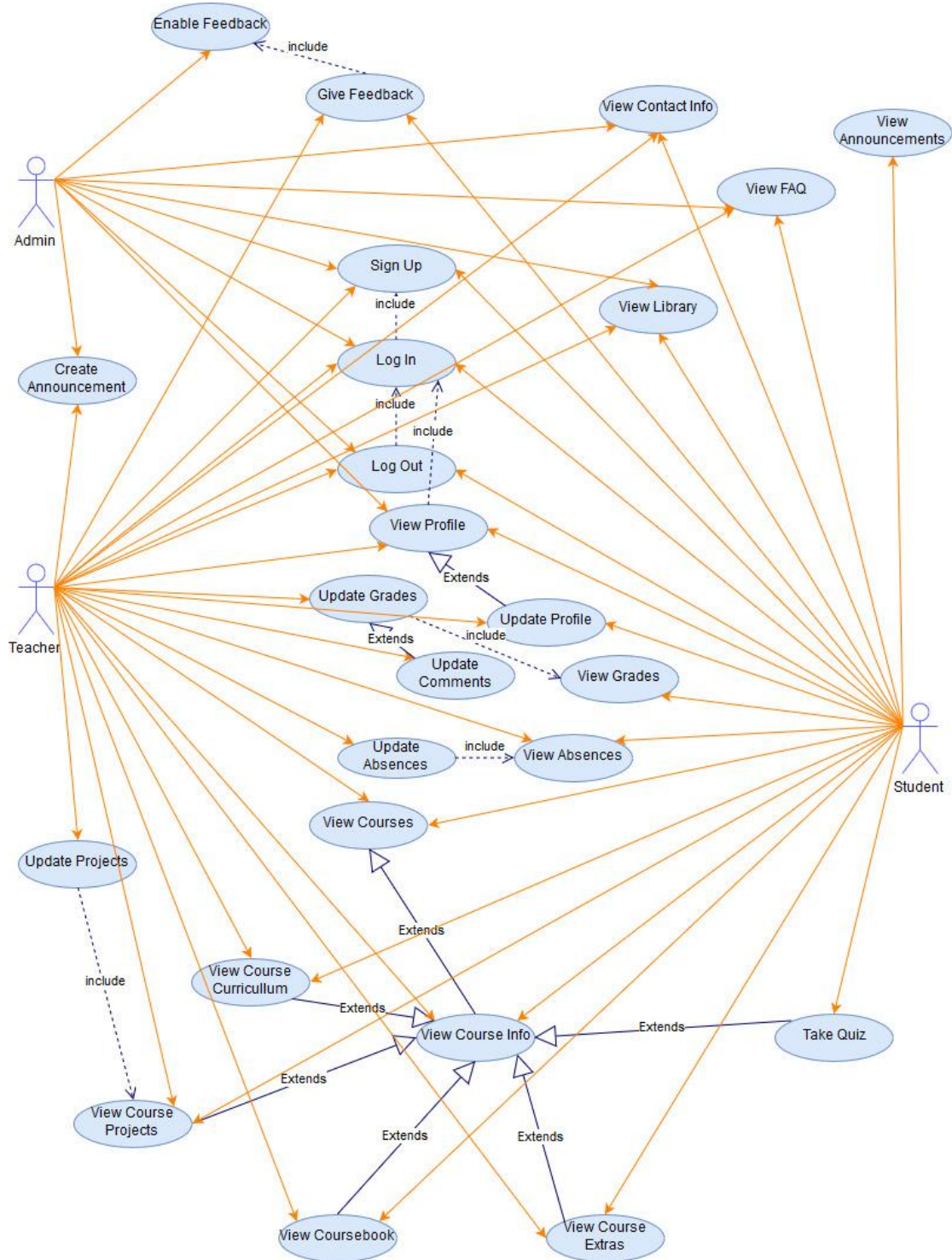
Οθόνη "Feedback" μονο για μαθητή/κηδεμόνα



Οι καθηγητές βαθμολογούν αναλυτικά το μαθητή σε κάθε τομέα και προσθέτουν σχόλια όπου χρειάζεται.
Οι μαθητές ή/και οι κηδεμόνες δίνουν κριτική για τον καθηγητή ή/και το σχολείο, προς βελτίωση και διόρθωση λαθών. Η κριτική θα είναι ανώνυμη (?)
Σε όλες τις περιπτώσεις, η κριτική θα γίνεται σε ορισμένες μόνο χρονικές περιόδους κάθε χρονιά.

Εικόνα 26 - Mock-up οθονών Feedback

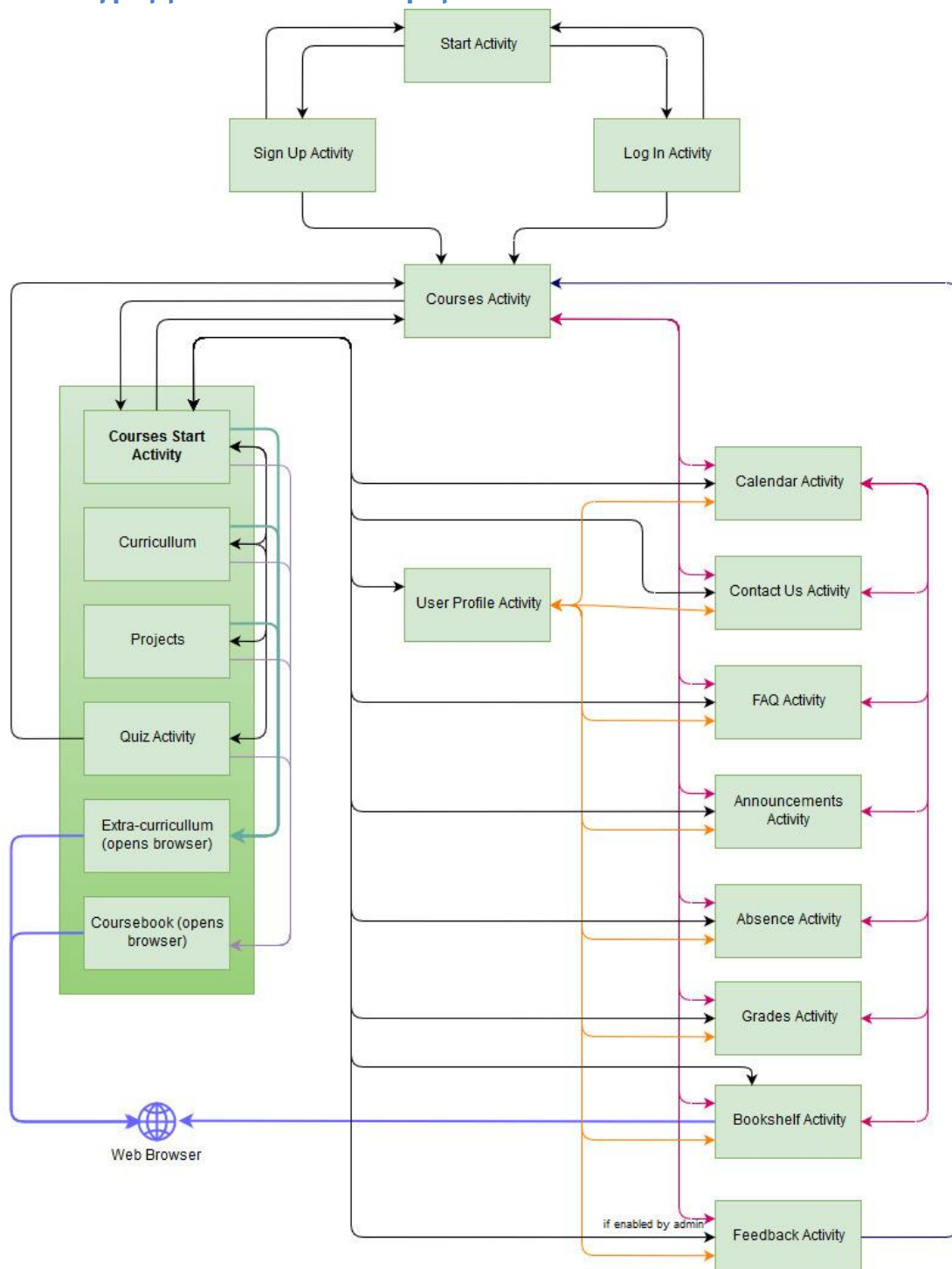
6.2 Λειτουργική Σχεδίαση



Εικόνα 27 - Διάγραμμα Use case

Κατασκευή Διαγράμματος με draw.io

6.3 Διάγραμμα Πιθανών διαδρομών Activities



Εικόνα 28 - Διάγραμμα Πιθανών Διαδρομών – Wireframing

Κατασκευή Διαγράμματος με draw.io



ΚΕΦΑΛΑΙΟ 7^ο: ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ

Στο κεφάλαιο αυτό θα αναλύσουμε τον κώδικα της εφαρμογής που αναπτύξαμε, και τις επιμέρους λειτουργίες που εκτελεί. Όπως έχουμε αναφέρει σε προηγούμενη ενότητα, για την διεκπεραίωση της εργασίας αυτής αναπτύξαμε μια πλήρως λειτουργική εφαρμογή σε Native Android, χρησιμοποιώντας τις παροχές ανοικτού κώδικα του Android και γράφοντας σε κώδικα XML και Java, για το οπτικό layout και το λειτουργικό τμήμα της εφαρμογής αντίστοιχα.

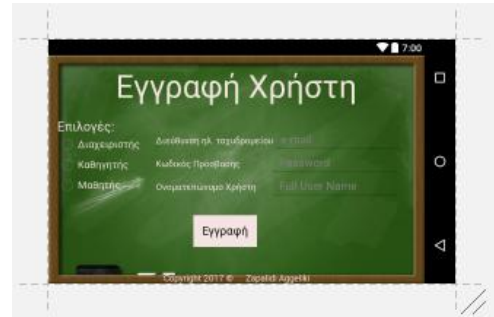
Ξεκινώντας, πρέπει να αναφέρουμε ένα προαπαιτούμενο για τη λειτουργία οποιασδήποτε εφαρμογής: η κλάση `app` περιέχει πληροφορίες απαραίτητες για τη λειτουργία της εφαρμογής. Δημιουργείται αυτόματα από το Android Studio, αλλά στη δική μας περίπτωση δημιουργήσαμε μια νέα, την οποία κάνουμε `override`, ώστε να πραγματοποιήσουμε τη σύνδεση με το Parse. (`App.java` - Παράρτημα σελ. 71) Η σύνδεση πραγματοποιείται με εντολές από το `documentation` του Parse, και με δεδομένα `credentials` που μας δόθηκαν όταν κατασκευάσαμε το `server`. Αξίζει να σημειωθεί ότι υπάρχουν και άλλοι τρόποι σύνδεσης με το Parse, που μπορεί να βρει κανείς εύκολα στη σελίδα του `documentation` του.

Στη συνέχεια, ακολουθεί ο κώδικας των `activities`. Το αρχικό `activity` που εκτελεί η εφαρμογή μας είναι το `MainActivity`. Πρόκειται για την αρχική οθόνη της εφαρμογής μας, η οποία παρουσιάζει στο χρήστη το όνομα της εφαρμογής και τον παροτρύνει να συνδεθεί στο σύστημα. Κατά την εκτέλεση του `main activity`, κατασκευάζονται στη βάση οι κλάσεις αντικειμένων που χρειαζόμαστε στη συνέχεια της εφαρμογής. Ενδεικτικά παρουσιάζουμε ανεπτυγμένα μόνο την `addAbsencelfNeeded()` (σειρά 59), αλλά όπως φαίνεται δίνεται η δυνατότητα διαχείρισης των δεδομένων πολύ εύκολα για την ομαλή λειτουργία της εφαρμογής μας. Οι μέθοδοι αυτοί εκτελούνται μόνο σε περίπτωση που η βάση δεν περιέχει αντικείμενα ή αυτές τις κλάσεις. Επίσης, τα δύο κουμπιά παραπέμπουν στα `activities` εγγραφής και εισόδου. (`MainActivity.java` - Παράρτημα σελ. 7271)

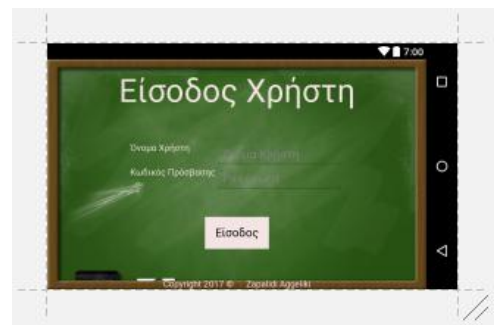
Στην εικόνα φαίνεται το `layout` που προκύπτει από τον κώδικα XML.



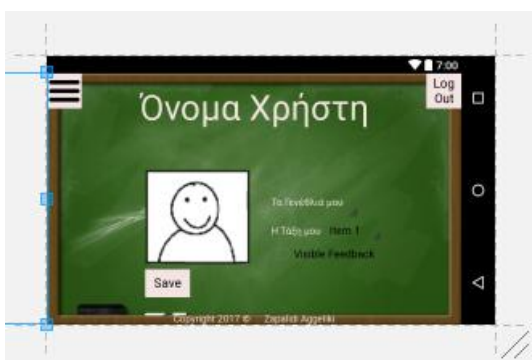
Προχωράμε στο `SignUpActivity`. Εδώ δημιουργείται ένας νέος χρήστης, με την ανάλογη εντολή του `Parse` (σειρά 52), προσθέτοντας τον αντίστοιχο ρόλο από επιλογή `radio button`, καθώς επίσης και τη δημιουργία εγγραφών απουσιών και βαθμών στις αντίστοιχες κλάσεις εάν ο χρήστης είναι μαθητής. Επιπλέον, όσο προστίθεται η εγγραφή στη βάση, εμφανίζεται στο χρήστη ένα `dialog spinner` που τον ενημερώνει ότι γίνεται αποθήκευση των στοιχείων του (σειρά 70). Σε περίπτωση που υπάρχει κάποιο λάθος (τύπος e-mail, έλλειψη password κλπ) εμφανίζεται το ανάλογο `alert` στην οθόνη του χρήστη. (σειρά 202) (`SignUpActivity.java` - Παράρτημα σελ. 7871)



Στη συνέχεια, ας αναλύσουμε το `LogInActivity`. Αντίστοιχα, εδώ ο χρήστης πραγματοποιεί είσοδο στο σύστημα χρησιμοποιώντας το ονοματεπώνυμό του (που είναι και το `username` του) και τον προσωπικό του κωδικό πρόσβασης. (σειρά 33) Σε περίπτωση κάποιου λάθους (λάθος συνδυασμός στοιχείων, έλλειψη σύνδεσης κλπ) εμφανίζεται στην οθόνη το αντίστοιχο `alert`. Επίσης, εμφανίζεται `progress dialog`, που δείχνει στο χρήστη ότι πραγματοποιείται είσοδος (σειρά 64). (`LogInActivity.java`- Παράρτημα σελ. 83)



Συνεχίζουμε με το `UserProfileActivity`. Πρόκειται για την οθόνη-προφίλ του χρήστη, όπου μπορεί να επεξεργαστεί την τάξη στην οποία ανήκει (ώστε ο μαθητής να μη δημιουργεί νέο λογαριασμό κάθε χρόνο) με ένα `spinner menu` (σειρά 70) και να αποσυνδεθεί από το σύστημα (σειρά 137). Στην οθόνη φαίνεται το όνομα του χρήστη κι ο ρόλος του. Στην περίπτωση του διαχειριστή, εμφανίζεται και ένα `checkbox` για το configuration του feedback. Το μενού στο πλάι είναι το βασικό μενού όλης της εφαρμογής μας, που περιέχει μια λίστα κουμπιών που παραπέμπουν στο ανάλογο `activity`, λαμβάνοντας υπόψη όπου χρειάζεται, το ρόλο του



συνδεδεμένου χρήστη, καθώς και εάν το πεδίο `isFeedbackVisible` είναι `true` (σειρά 47). Το μενού `Drawerlayout` βρίσκεται προφανώς εκτός της οθόνης του UI. (`UserProfileActivity.java` - Παράρτημα σελ. 7871)

Το «κεντρικό» activity λειτουργίας τάξης είναι το `CoursesActivity`, που περιέχει `recycler view`



με όλα τα μαθήματα.

(`CoursesActivity.java` - Παράρτημα

σελ.96). Επιλέγοντας το καθένα, ανοίγει ένα νέο

activity (`CourseStartActivity.java` -

Παράρτημα σελ.103) που περιέχει

συγκεντρωμένο όλο το υλικό του μαθήματος,

επιτρέποντας τη διαχείριση της τάξης από τον

καθηγητή. Περιέχει το βιβλίο, επιπλέον υλικό

κάθε μαθήματος, εκφωνήσεις και ομάδες

εργασιών, και παραπομπές σε εξωτερικές πηγές

και σε quiz που καταγράφουν την κατανόηση και

την πρόοδο του μαθητή. Τα στοιχεία του `recycler`

είναι τα παρακάτω: (`course_recycler_row.xml` - Παράρτημα σελ. 977271),

(`CourseModel.java` - Παράρτημα σελ. 9871),

(`ShowCoursesRecyclerAdapter.java` - Παράρτημα σελ. 9971) Όπως είδαμε και

στο κεφάλαιο της θεωρίας, για να λειτουργήσει σωστά το `recycler view` δημιουργεί μια

κλάση αντικειμένων που θα αποτελούν κάθε γραμμή του, και έναν αντάπτορα που να

αναθέτει το κάθε component στη σωστή θέση που ορίσαμε. Επιπλέον, η συγκεκριμένη

λίστα περιέχει `clickable` στοιχεία, που παραπέμπουν με `extras` το καθένα σε αντίστοιχο

μάθημα, με τις ανάλογες πληροφορίες.

Το activity `CourseStartActivity.java`, χάρη στα `extras` που έφερε η εντολή `intent`

στους `onClickListeners` του προηγούμενου activity (σειρά 56), εμφανίζει το αντίστοιχο υλικό

του κάθε μαθήματος που επιλέχθηκε, παραπέμποντας μεταξύ άλλων, σε κουίζ (που θα

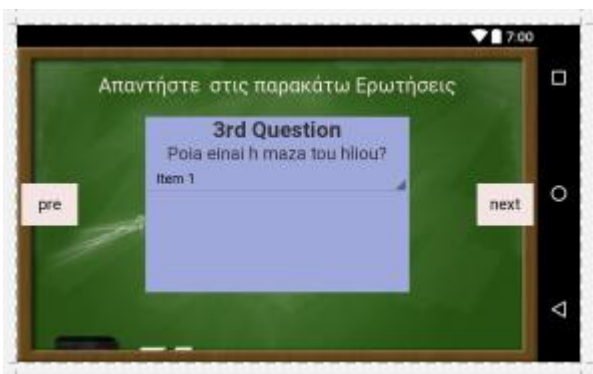
αναλύσουμε παρακάτω), στο σχολικό βιβλίο και σε επιπλέον υλικό! (σειρά 213 και εξής)

Το μόνο νέο activity στο οποίο οδηγούμαστε από το εσωτερικό μενού του `CourseStart` είναι

αυτό των κουίζ του μαθήματος. Πρόκειται για ένα φαινομενικά πολύπλοκο activity που

ωστόσο έχει πολύ απλή λογική σχεδίαση. (`QuizActivity.java` - Παράρτημα σελ.

110) Έχουμε κατασκευάσει μια στοίβα από «κάρτες» ερωτήσεων, όπου κάθε κάρτα είναι ένα `LinearLayout`. Η ορατότητα του κάθε ενός μεταβάλλεται ανάλογα με τον αριθμό του στη λίστα καρτών, με την εντολή `setVisibility(View.GONE)` (Java) ή `android:visibility(GONE)(XML)`. `activity_quiz.xml` - Παράρτημα σελ. 107)



Η εκάστοτε ορατή κάρτα εμφανίζεται χάρη στα κουμπιά `pre` και `next` τα οποία κινούνται εμπρός και πίσω στη λίστα των αριθμημένων καρτών (σειρά 75). Όταν η τελευταία κάρτα της λίστας είναι ορατή, το κουμπί `next` μετονομάζεται σε `submit`, και αποθηκεύει `onClick` τις απαντήσεις

που έδωσε ο χρήστης, στη βάση (σειρά 130).



Συνεχίζουμε με το `BookshelfActivity`. Αυτό είναι το `activity` της βιβλιοθήκης μας, που περιέχει ένα `GridView`, που αποτελείται από τα «βιβλία» και τους τίτλους του, και όταν ο χρήστης πατήσει κάποιο από αυτά, η εντολή `intent` του `onClickListener` τον παραπέμπει σε `browser`, όπου «ανοίγει» το βιβλίο ως `PDF online`, όπως και στο `activity`

του κάθε μαθήματος (`BookshelfGridViewAdapter.java` - Παράρτημα σελ. 119 - σειρά 57). Η προβολή του εξωφύλλου του κάθε βιβλίου έγινε με χρήση της βιβλιοθήκης `Glide` (`BookshelfGridViewAdapter.java` - Παράρτημα σελ. 119 -σειρά 49), για προβολή εικόνων. (`BookshelfActivity.java` - Παράρτημα σελ. 11571)



Το `activity` των βαθμών του μαθητή είναι αρκετά απλό στη σχεδίασή του: περιέχει ένα `recycler view` που φέρνει από τη βάση τους βαθμούς ανά μάθημα του μαθητή με βάση το όνομά του. Ο μαθητής δε μπορεί να δει τα σχόλια του καθηγητή, ούτε τους βαθμούς άλλων μαθητών εκτός των δικών του.

(`GradesStudentActivity.java` - Παράρτημα σελ. 12271)

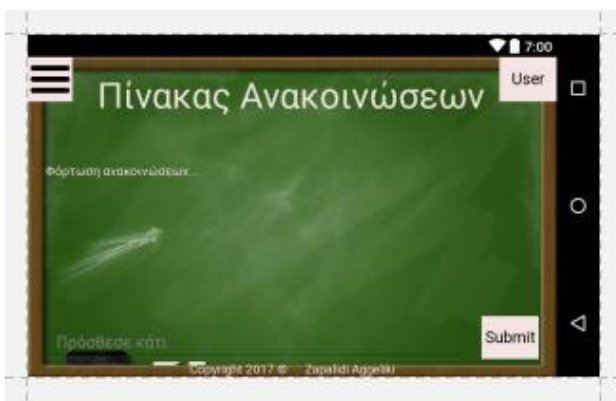
Το activity βαθμών για τον καθηγητή είναι λίγο πιο πολύπλοκο, καθώς επιστρέφει σε recycler view τους βαθμούς όλων των μαθητών, για όλα τα μαθήματα, καθώς και σχόλια για την επίδοσή τους ή οτιδήποτε κρίνει ο καθηγητής ότι πρέπει να αναφερθεί.

(GradeTeacherAdapter.java



– Παράρτημα σελ. 130) Ο καθηγητής μπορεί να δει το ιστορικό βαθμών του μαθητή, την εξέλιξη του και τα σχόλια προηγούμενων καθηγητών του, και να αναθέσει τον βαθμό που πρέπει καθώς και να ενημερώσει τα σχόλια και να αποθηκεύσει στη βάση τις αλλαγές αυτές (GradeTeacherAdapter.java – Παράρτημα σελ. 130 - σειρά 67 και εξής). (GradesTeacherActivity.java - Παράρτημα σελ. 1267271)

Αντίστοιχα, το activity ανακοινώσεων για τον καθηγητή και το διαχειριστή περιέχει ένα



TextView που θα εμφανίζει τις προηγούμενες ανακοινώσεις (σειρά 56), και ένα πεδίο EditText όπου θα μπορεί να γράφει νέες ανακοινώσεις, καθώς και ένα κουμπί Submit με το οποίο θα τις αποθηκεύει στη βάση (σειρά 86). Το κείμενο που προσθέτει ο χρήστης προσκολλάται στο τέλος

του υπάρχοντος κειμένου, ώστε κάθε φορά να εμφανίζεται το ενημερωμένο κείμενο.

(AnnouncementsTeacherActivity.java - Παράρτημα σελ. 13571)

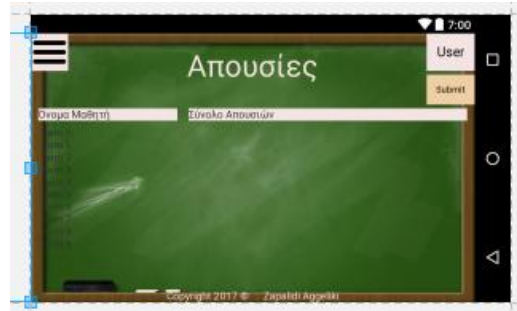
Το αντίστοιχο activity για το μαθητή του επιτρέπει απλώς να βλέπει τις ανακοινώσεις και μόνο, όχι να τις ενημερώνει.

Το activity απουσιών του μαθητή φέρνει από τη βάση τον αριθμό συνολικών και δικαιολογημένων απουσιών του μαθητή με βάση το όνομά του (σειρά 67), και υπολογίζει από τη διαφορά τους το σύνολο των αδικαιολόγητων (σειρές 75-77). Και



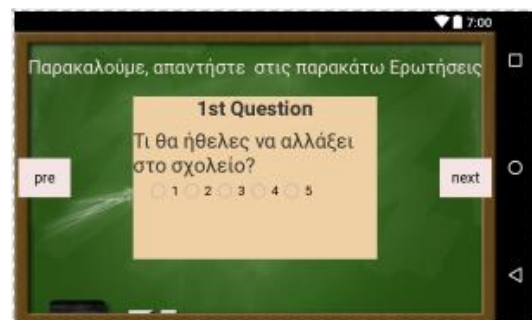
πάλι, ο μαθητής μπορεί να δει μόνο τις δικές του απουσίες, και όχι να τις επεξεργαστεί ή να δει αυτές των συμμαθητών του. (AbsenceStudentActivity.java - Παράρτημα σελ. 14071)

Στο activity απουσιών για τον καθηγητή, περιλαμβάνεται ένα RecyclerView με το όνομα κάθε μαθητή, καθώς και δύο πεδία EditText για τις δικαιολογημένες και τις αδικαιολόγητες απουσίες του καθενός. Επίσης, υπάρχει ένα κουμπί αποθήκευσης του συνόλου των αλλαγών, ως εγγραφών του ArrayList, στη βάση (σειρά 77). (AbsenceTeacherActivity.java - Παράρτημα σελ. 14371)



Συνεχίζουμε με το CalendarActivity. Στο εν λόγω activity, υπάρχει ένα CalendarView (σειρά 56), χρησιμότατο ως σχολικό εργαλείο, στο οποίο κάθε ημέρα είναι clickable και εμφανίζει pop-up μήνυμα με τα συμβάντα (events) της ημέρας αυτής (σειρά 63). Στην περίπτωση του καθηγητή και του διαχειριστή, είναι ορατό επίσης ένα κουμπί AddEvent (σειρά 114), που παραπέμπει στο αντίστοιχο activity, όπου μπορούν να προσθέσουν συμβάντα στη βάση, συμπληρώνοντας τα κατάλληλα πεδία. (CalendarActivity.java - Παράρτημα σελ. 14771)

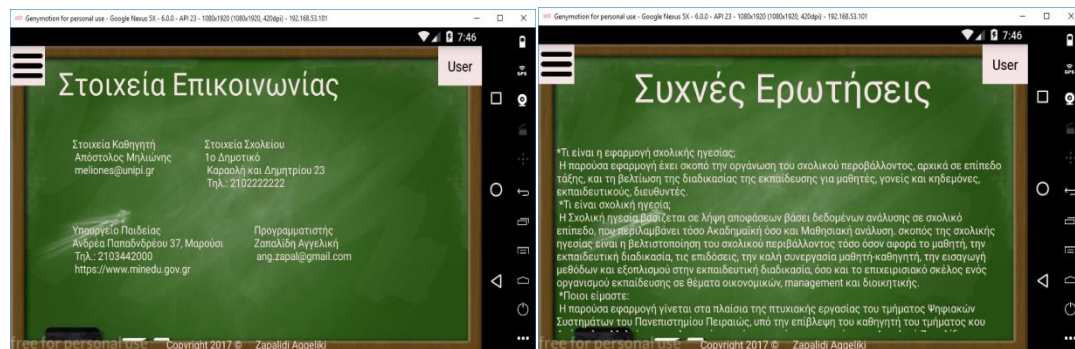
Ένα από τα πιο σημαντικά activities για τη λειτουργία της εφαρμογής σχολικής ηγεσίας είναι αυτό της κριτικής (feedback), αναγκαίας για τη διοίκηση πολυπλοκότητας. Το





συγκεκριμένο activity δομήθηκε όπως αυτό των quiz, εμφανίζοντας «καρτέλες» ερωτήσεων που εναλλάσσονται (ως λίστα καρτελών, σαρώνοντας τη λίστα εναλλάσσονται οι ορατότητες των καρτελών – σειρά 73) με κουμπιά “pre” και “next”, ενώ στο τελευταίο της λίστας το κουμπί γίνεται “Submit”. Το activity αυτό τραβά από τη βάση τις ερωτήσεις ανάλογα με το ρόλο του χρήστη (άλλες για τον καθηγητή και άλλες για το μαθητή και τον γονέα/κηδεμόνα, που απαντούν ως ένας χρήστης – σειρά 131, 139 και εξής). Το activity είναι προσπελάσιμο μόνο εφόσον ο admin έχει θέσει την τιμή του πεδίου isFeedbackVisible σε true. Οι απαντήσεις των χρηστών κατατίθενται ανώνυμα στη βάση (σειρά 163), και θα ειδοποιούνται με μέιλ (έχουν καταχωρήσει τη διεύθυνση τους κατά την εγγραφή τους) από τους διαχειριστές του συστήματος, ότι το πεδίο κριτικής είναι ανοικτό. (Αυτό θα πραγματοποιείται εκτός της εφαρμογής). Οι καρτέλες ερωτήσεων περιέχουν ερωτήσεις βαθμολόγησης με radio buttons, και ανάπτυξης όπου ο χρήστης γράφει ένα κείμενο με τη γνώμη του. (FeedbackActivity.java - Παράρτημα σελ. 15171)

Επιπλέον, υπάρχουν δύο στατικά activities που προβάλλουν πληροφορίες για την εφαρμογή (FAQ) και τα στοιχεία επικοινωνίας (Contact Us), τα οποία είναι απλά layouts με κείμενο, και φυσικά το βασικό μενού. Δεν περιέχουν κάτι νέο ή διαφορετικό από άποψη κώδικα.





ΚΕΦΑΛΑΙΟ 8^ο: ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΗΣ

Η Εφαρμογή Σχολικής Ηγεσίας που σχεδιάσαμε, υλοποιήσαμε και κατασκευάσαμε στα πλαίσια αυτής της πτυχιακής εργασίας έχει σκοπό να καλύψει σε έναν βασικό βαθμό τις απαραίτητες λειτουργίες που θα μπορεί να εκμεταλλευτεί μια ομάδα ηγεσίας, οι καθηγητές και όλοι οι πράκτορες του συστήματος που αποτελεί το σχολείο, για τη βελτιστοποίηση της λήψης αποφάσεων σε αυτό. Με βάση τη θεωρία που μελετήσαμε στη 2^η ενότητα, εντοπίσαμε τις λειτουργικές και μη λειτουργικές απαιτήσεις μιας τέτοιας εφαρμογής, ενώ στη συνέχεια αναλύσαμε τις τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της. Το αποτέλεσμα είναι μια εφαρμογή που βοηθά στην οργάνωση και την ομαλή λειτουργία του περιβάλλοντος της τάξης, και το κυριότερο στη σωστή και εύκολη συλλογή δεδομένων τόσο σε επίπεδο micro (σχετικά με το μαθητή, τη μαθησιακή και εκπαιδευτική διαδικασία, τις επιδόσεις, την κατάστασή του, την αποδοτικότητα των μεθόδων του καθηγητή κλπ) όσο και σε επίπεδο meso (γενικότερο πλαίσιο επιδόσεων του σχολείου, της φήμης του, της οργάνωσης του, του προϋπολογισμού εσόδων/εξόδων του κλπ), τουλάχιστον σε ένα ικανοποιητικό πρώτο βαθμό. Ασφαλώς, τόσο με την επιστράτευση νέων τεχνολογιών στην εκπαίδευση και στην ηγεσία συστημάτων βάσει δεδομένων, αλλά ακόμα και με τη δημιουργία και εξέλιξη απλών εφαρμογών σαν αυτήν, οι μελλοντικές προοπτικές ενός τέτοιου εγχειρήματος θα ήταν απεριόριστες και πολύ χρήσιμες στον τομέα αυτό.

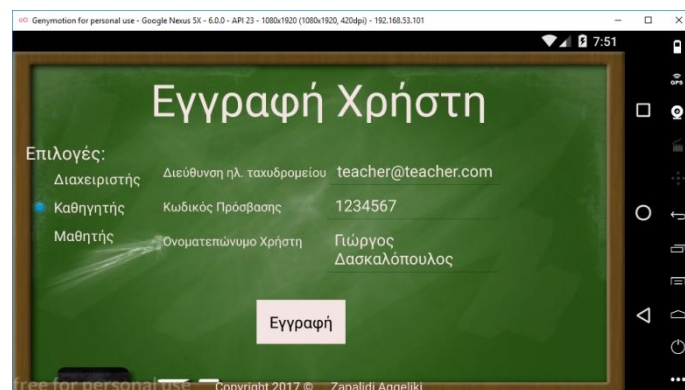
Στην ενότητα αυτή θα περιγράψουμε τη χρήση και τις λειτουργίες της εφαρμογής στην πράξη, και την ανταπόκριση τους σε θέματα Σχολικής Ηγεσίας.

Αρχικά, ο χρήστης βλέπει την αρχική οθόνη της εφαρμογής, που τον παρακινεί να δημιουργήσει λογαριασμό χρήστη στο σύστημα ή να εισέλθει με τον υπάρχοντα λογαριασμό. Κανείς δεν εισέρχεται στο σύστημα χωρίς λογαριασμό, εγκεκριμένο από το σχολείο, και εξακριβωμένο από τους διαχειριστές της εφαρμογής.



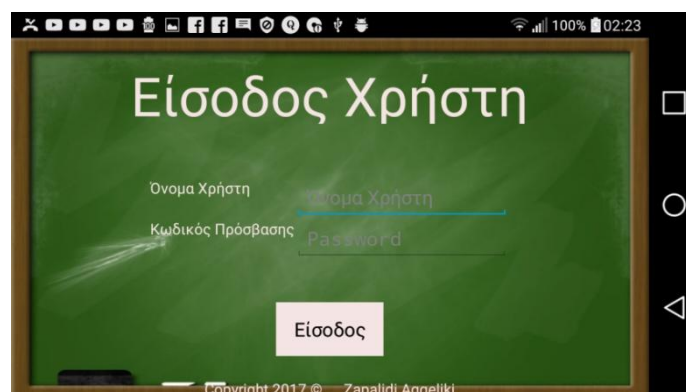
Εικόνα 29 - Αρχική Οθόνη Εφαρμογής

Κατά την εγγραφή, ζητείται από το χρήστη να εισάγει τα στοιχεία του: ονοματεπώνυμο, διεύθυνση ηλεκτρονικού ταχυδρομείου, προσωπικό κωδικό πρόσβασης και ρόλο χρήστη. Το ονοματεπώνυμο και ο κωδικός θα είναι απαραίτητα για την είσοδο του χρήστη στο σύστημα. Η διεύθυνση e-mail θα χρειάζεται να καταχωρηθεί στη βάση του συστήματος, ώστε να μπορεί το σχολείο να επικοινωνεί με τους γονείς/κηδεμόνες του μαθητή ή με τους καθηγητές ανά πάσα στιγμή, για ενημερώσεις και ανακοινώσεις που τους αφορούν. (Η αποστολή των e-mail δεν πραγματοποιείται μέσω της εφαρμογής, ωστόσο η σύνδεση οποιουδήποτε προγράμματος ηλεκτρονικού ταχυδρομείου με τη βάση μας στο Parse μπορεί να γίνει πολύ εύκολα). Η διαδικασία της εγγραφής προφανώς δε μπορεί να πραγματοποιείται χωρίς έλεγχο. Επομένως, είναι απαραίτητη η παρουσία των γονέων του μαθητή κατά την εγγραφή, και η παροχή μιας διεύθυνσης e-mail από αυτούς, και η επιβεβαίωση των στοιχείων του μαθητή. Επομένως η εγγραφή είτε γίνεται από το σπίτι, με ευθύνη των γονέων, και επιβεβαίωση από το σχολείο, είτε θα πραγματοποιείται εξ'αρχής στο σχολείο, υπό την επίβλεψη της ομάδας διαχειριστών της εφαρμογής. Οι πιθανοί ρόλοι χρηστών είναι διαχειριστής, καθηγητής και μαθητής (όπου συμπεριλαμβάνεται και ο κηδεμόνας). Πρέπει να σημειωθεί ότι ο διαχειριστής που αναφέρουμε δεν είναι διαχειριστής της ομάδας ηγεσίας, αλλά των τεχνικών υποστήριξης της εφαρμογής.

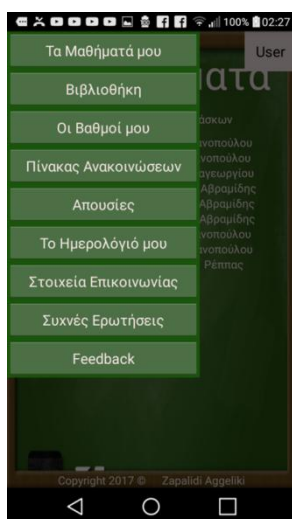


Εικόνα 30 - Εγγραφή Χρήστη

Μετά την εγγραφή ο χρήστης θεωρείται αυτομάτως συνδεδεμένος στο σύστημα, σε περίπτωση όμως που έχει ήδη λογαριασμό, πραγματοποιεί είσοδο από το αντίστοιχο activity, εισάγοντας το ονοματεπώνυμο και τον κωδικό του.

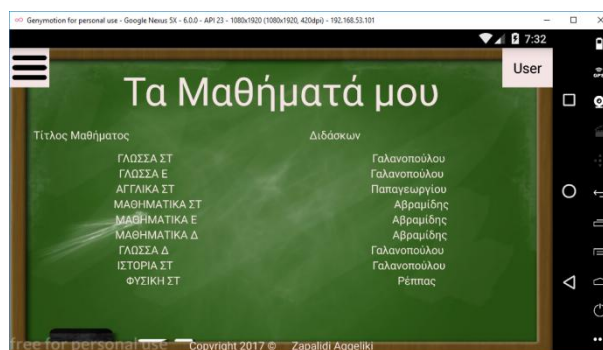


Εικόνα 31 - Είσοδος Χρήστη



Μετά την είσοδο στο σύστημα, η εφαρμογή παραπέμπει το χρήστη στην οθόνη των μαθημάτων, την πρώτη οθόνη της κυρίως εφαρμογής. Σε κάθε βασικό activity υπάρχουν ένα κουμπί που παραπέμπει το χρήστη στο προφίλ του, και ένα κουμπί πλαινού μενού (drawer navigation menu) το οποίο μενού όμως ανοίγει και με «σύρσιμο» του χρήστη από τα αριστερά. Το μενού αυτό είναι scrollable, σε περίπτωση που δε χωρά στην οθόνη της συσκευής. Το συνολικό μενού φαίνεται στην εικόνα. Όσον αφορά το συγκεκριμένο activity, παρουσιάζονται όλα τα μαθήματα του σχολείου και οι αντίστοιχοι καθηγητές τους, και ο χρήστης μπορεί να επιλέξει ένα από όλα τα μαθήματα της λίστας και να παραπεμφθεί στο activity που περιέχει αναλυτικό υλικό για το κάθε μάθημα.

Εικόνα 32 - Το κύριο μενού

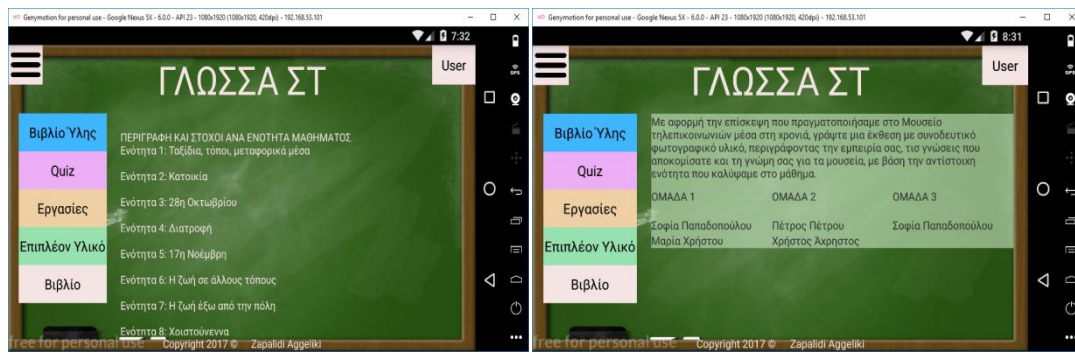


Εικόνα 33 - Τα μαθήματα

Ενδεικτικά, παρουσιάζουμε το activity μαθήματος για μία μόνο επιλογή, τη Γλώσσα ΣΤ' Δημοτικού. Όπως φαίνεται, το activity περιέχει ένα δεύτερο εσωτερικό μενού, με πέντε επιλογές χρήστη. Η αρχική οθόνη δείχνει ένα κείμενο με βασικές πληροφορίες για το μάθημα. Οι επιλογές Βιβλίο Ύλης και Εργασίες προβάλλουν αντίστοιχα το κείμενο με τις ενότητες του μαθήματος και τις εκφωνήσεις εργασιών και ομάδες μαθητών, όπως τις ορίζει ο καθηγητής.

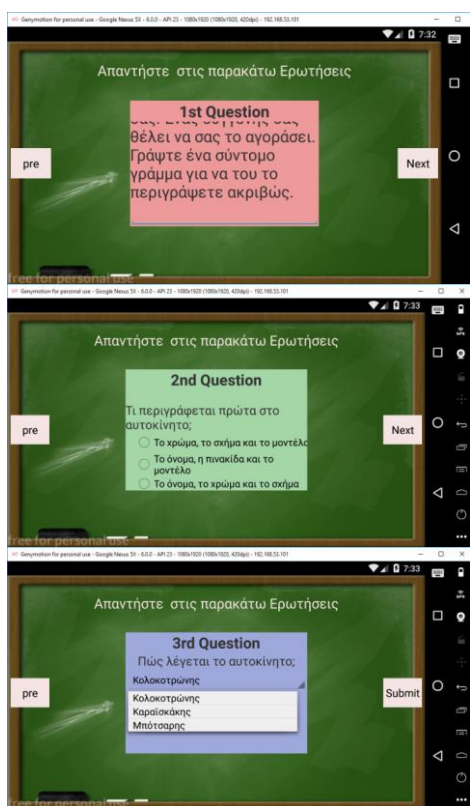


Εικόνα 34 - Οθόνη μαθήματος



Εικόνα 35 - Βιβλίο Ύλης

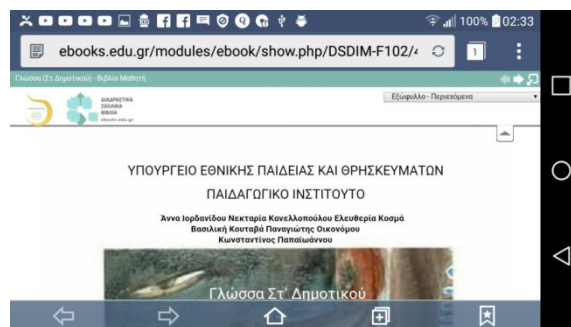
Εικόνα 36 - Εργασίες



Εικόνα 37 - Κουίζ

Το κουμπί Quiz παραπέμπει σε ξεχωριστό activity, όπου ο μαθητής απαντά σε ερωτήσεις πάνω στο μάθημα, ώστε να αξιολογήσει τις γνώσεις που αποκόμισε, την κατανόηση της ζύλης που διδάχθηκε και να εντοπίσει τυχόν κενά που χρειάζεται να καλύψει με επιπλέον μελέτη στο σπίτι ή στο σχολείο. Τα αποτελέσματα καταχωρούνται στη βάση, και επιτρέπουν τόσο στον καθηγητή όσο και στους υπεύθυνους ηγεσίας να παρατηρούν και να παρακολουθούν τις επιδόσεις του μαθητή, συγκεντρώνοντας έτσι τα απαραίτητα δεδομένα (micro layer) που χρειάζονται για τη λήψη αποφάσεων.

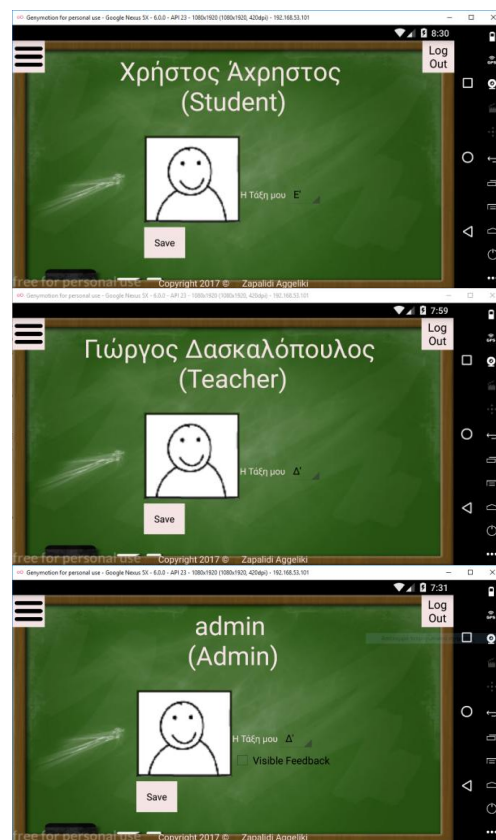
Τα κουμπιά Βιβλίο και Επιπλέον Υλικό ανοίγουν στον browser της συσκευής αντιστοίχως το σχολικό βιβλίο του μαθήματος σε PDF, και την ιστοσελίδα του υπουργείου παιδείας για το συγκεκριμένο μάθημα, με επιπλέον υλικό. Μελλοντικά, θα μπορούσαν να προστεθούν και άλλες πηγές, πχ εκπαιδευτικά βίντεο, διαδραστικά βιβλία και παιχνίδια, κατά την άποψη του καθηγητή ή και του υπουργείου παιδείας.



Εικόνα 38 - Το κουμπί Βιβλίο παραπέμπει στο browser

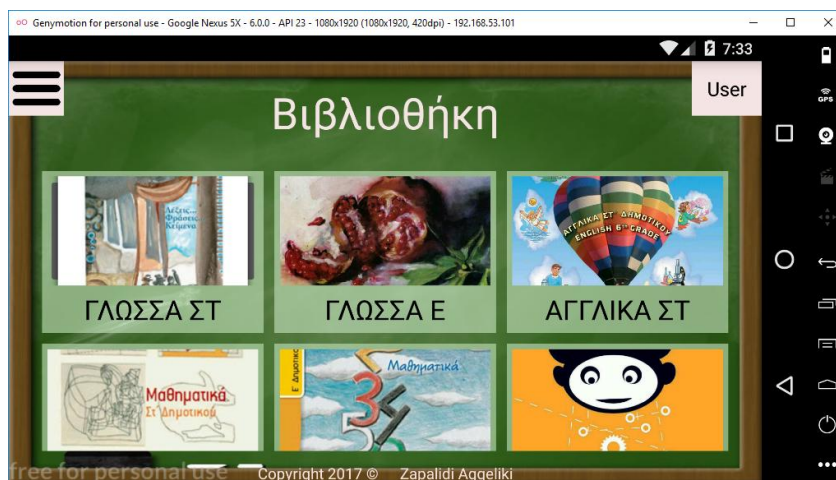


Όπως προαναφέραμε, το κουμπί Χρήστη, επάνω δεξιά, παραπέμπει στο Προφίλ του χρήστη. Στο προφίλ αυτό εμφανίζεται το όνομα και ο ρόλος του συνδεδεμένου χρήστη, η τάξη του (σε περίπτωση δασκάλου ή μαθητή) και η δυνατότητα να την ενημερώνει. Κατά αυτόν τον τρόπο δε θα χρειάζεται να δημιουργείται νέος λογαριασμός χρήστη κάθε χρονιά, αφού ο μαθητής θα αλλάζει τάξη. Στην περίπτωση που ο χρήστης είναι διαχειριστής του συστήματος, εμφανίζεται και η επιλογή μέσω checkbox του να καταστήσει την επιλογή του Feedback προσπελάσιμη στους άλλους χρήστες (configuration). Επίσης, στο προφίλ υπάρχει και η δυνατότητα αποσύνδεσης από το σύστημα, με το αντίστοιχο κουμπί. Προφανώς, σε μελλοντικές εκδόσεις της εφαρμογής θα μπορούσαν να προστεθούν και άλλα στοιχεία, κάνοντας το προφίλ πιο οικείο και προσωπικό για τον κάθε χρήστη, πχ η δυνατότητα να αλλάζει την εικόνα προφίλ του, να εισάγει πληροφορίες όπως τα γενέθλιά του κλπ.



Εικόνα 39 - Τα προφίλ ανά ρόλο χρήστη

Το activity βιβλιοθήκης περιλαμβάνει όλα τα σχολικά βιβλία που είναι διαθέσιμα στο χρήστη, προβαλλόμενα σε GridView. Όπως και στην επιλογή του βιβλίου, στο κάθε μάθημα έτσι και εδώ πατώντας στο εκάστοτε βιβλίο, αυτό «ανοίγει» σε μορφή PDF στο browser της συσκευής. Η επιφάνεια διεπαφής για αυτό το activity είναι ίδια για κάθε χρήστη, ανεξαρτήτως του ρόλου του.



Εικόνα 40 - Η Βιβλιοθήκη μου

Υπάρχουν δύο διαφορετικά activity Βαθμών για τους μαθητές και για τους καθηγητές. Προφανώς, ο καθηγητής πρέπει να μπορεί να βλέπει τους βαθμούς όλων των μαθητών του αλλά και να μπορεί να τους επεξεργάζεται ή να τους ενημερώνει σε κάθε περίοδο βαθμολόγησης. Επιπλέον, θα πρέπει να επισυνάπτει σχόλια για τις επιδόσεις του κάθε μαθητή σε κάθε μάθημα, σχετικά με τις επιδόσεις του, τις αδυναμίες του, τυχόν καταστάσεις ή προβλήματα κλπ., όχι μόνο στα πλαίσια της εκπαίδευσης αλλά και της ηγεσίας, καθώς είναι δεδομένα που συλλέγονται και βοηθούν στην παρακολούθηση των επιδόσεων των μαθητών και των αναγκών του σχολείου, και άρα στην ανάλογη λήψη αποφάσεων. Πχ. Τη δημιουργία ομάδων βοήθειας για μαθητές με δυσλεξία, εφόσον υπάρχουν αρκετοί στο σχολείο, ή την πρόταση μαθητών για συμμετοχή σε διαγωνισμούς στα μαθήματα στα οποία είναι δυνατοί.

Όνομα Μαθητή	Μάθημα	Τρίμηνο Α	Τρίμηνο Β	Τρίμηνο Γ	Σχόλια
Σοφία Παπαδοπούλου	ΑΓΓΛΙΚΑ ΣΤ	9	9	10	Μπορεί να τα καταφέρει καλύτερα στο λεξιλόγιο
Πέτρος Πέτρου	ΓΛΩΣΣΑ Δ	8	8	9	Χρειάζεται δουλειά στην έκθεση

Εικόνα 41 - Βαθμοί Καθηγητή

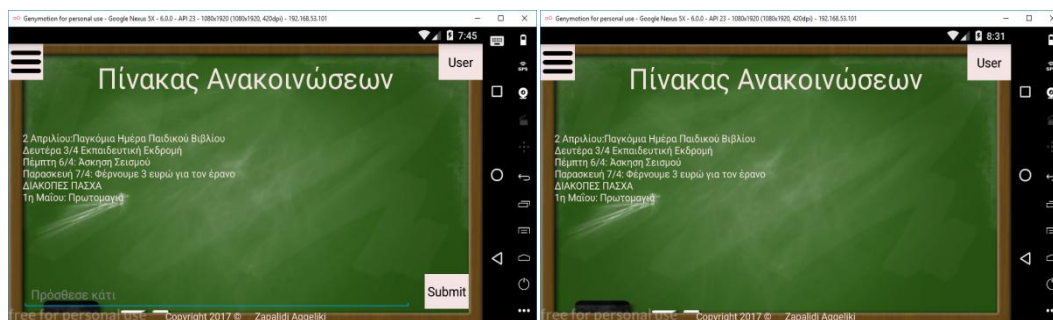
Ο μαθητής ωστόσο πρέπει επίσης να μπορεί να βλέπει τους βαθμούς του, σε όλα τα μαθήματα, να παρακολουθεί την πρόδό του και τις αδυναμίες του, χωρίς ωστόσο να μπορεί να επεξεργαστεί τους βαθμούς του ή να προσπελάσει αυτούς των συμμαθητών του.

Μάθημα	Εξάμηνο Α	Εξάμηνο Β	Εξάμηνο Γ
ΓΛΩΣΣΑ Δ	5	5	7
ΜΑΘΗΜΑΤΙΚΑ ΣΤ	5	7	7
ΜΑΘΗΜΑΤΙΚΑ Ε	5	7	7
ΦΥΣΙΚΗ ΣΤ	5	5	7
ΜΑΘΗΜΑΤΙΚΑ Δ	5	5	7
ΓΛΩΣΣΑ ΣΤ	5	5	7
ΓΛΩΣΣΑ Ε	5	5	7
ΑΓΓΛΙΚΑ ΣΤ	5	7	7
ΙΣΤΟΡΙΑ ΣΤ	5	5	7

Εικόνα 42 - Βαθμοί Μαθητή

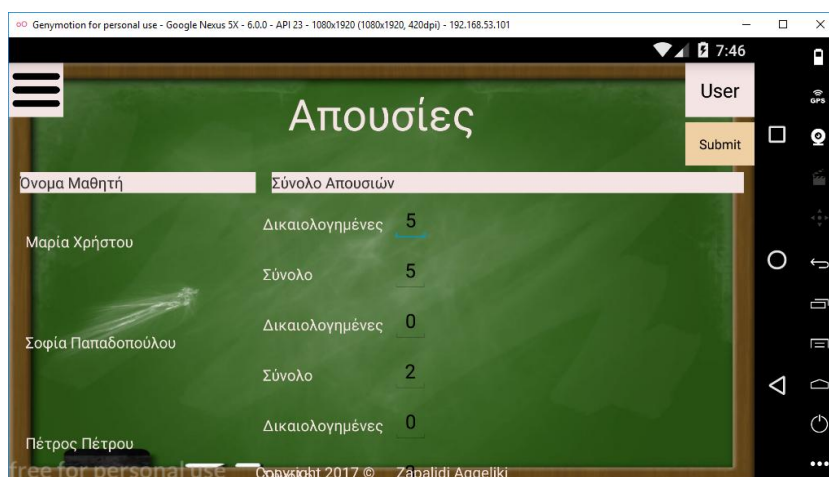
Το επόμενο activity είναι ο Πίνακας Ανακοινώσεων, όπου οι διαχειριστές, η διεύθυνση του σχολείου και οι δάσκαλοι μπορούν να ανεβάζουν συμβάντα σχετικά με την οργάνωση του

σχολείου (πχ. Σχολικές εορτές, εκδρομές) αλλά και εξωσχολικού περιεχομένου (πχ. Παγκόσμιες ημέρες, εξωτρικές εκδηλώσεις). Κάθε νέα ανακοίνωση προστίθεται κάτω από τις υπάρχουσες. Ο μαθητής μπορεί μόνο να δει τις ανακοινώσεις και όχι να προσθέσει καινούριες.



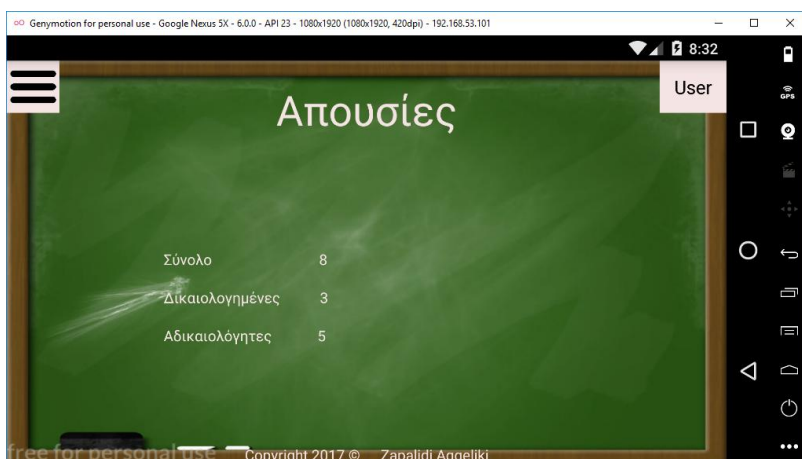
Εικόνα 43 - Πίνακας Ανακοινώσεων καθηγητή και μαθητή

Αντίστοιχα με τους βαθμούς, χειριζόμαστε και τις απουσίες των μαθητών. Κατά την εγγραφή του μαθητή, καταχωρείται αυτομάτως στη βάση ότι έχει μηδενικό αριθμό απουσιών. Ο καθηγητής ή ο υπεύθυνος τμήματος μπορεί να δει τον αριθμό απουσιών για κάθε μαθητή αλλά και να προσθέσει νέες ή να καταχωρήσει υπάρχουσες δικαιολογημένες αναλόγως με τις κινήσεις του μαθητή. Να σημειωθεί ότι οι απουσίες δεν απαιτούνται τόσο για λόγους ποινής, ειδικά στις τάξεις του δημοτικού, όσο στην καταγραφή της συμμετοχής των μαθητών και των επιπτώσεων των απουσιών στις επιδόσεις τους, δεδομένα απαραίτητα για τη σχολική ηγεσία.



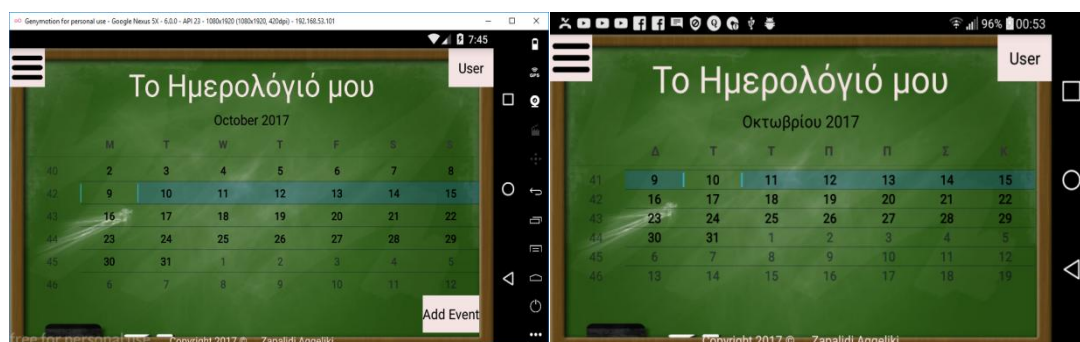
Εικόνα 44 - Απουσίες Καθηγητή

Αντιστοίχως, ο μαθητής πρέπει να μπορεί να ελέγχει και ο ίδιος το σύνολο των απουσιών του ανά πάσα στιγμή, καθώς επίσης και ο γονέας του. Ο μαθητής όμως δε μπορεί να επεξεργαστεί τον αριθμό απουσιών του, ούτε και να προσπελάσει τις πληροφορίες σχετικά με τις απουσίες άλλων μαθητών. Ο μαθητής μπορεί να βλέπει τις συνολικές, δικαιολογημένες και αδικαιολόγητες απουσίες του, και να επιφέρει εάν χρειάζεται τα απαραίτητα διακιολογητικά.



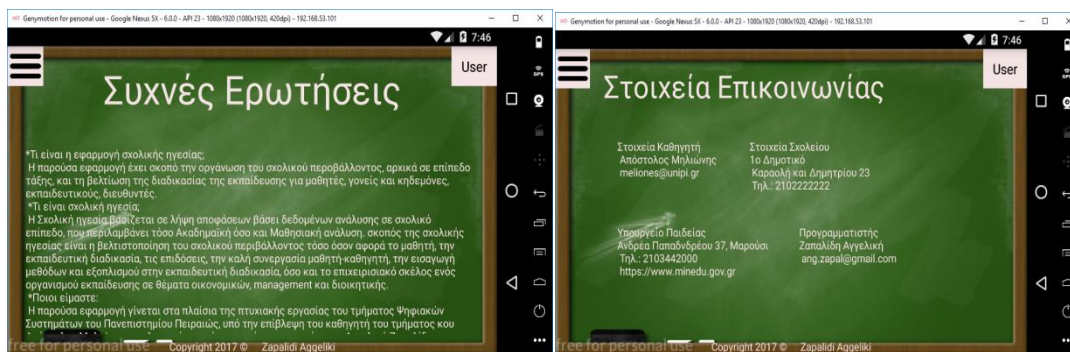
Εικόνα 45 - Απουσίες Μαθητή

Κατ' αντιστοιχία με τον πίνακα ανακοινώσεων, η εφαρμογή περιέχει και ένα ημερολόγιο, απαραίτητο εργαλείο οργάνωσης στο σχολικό περιβάλλον. Το ημερολόγιο, εκτός της χρησιμότητάς του, παρέχει επίσης στο χρήστη προβολή των συμβάντων που έχουν καταχωρηθεί, ανά ημερομηνία. Έτσι, πατώντας επάνω στην οποιαδήποτε ημέρα, το activity εμφανίζει ένα μήνυμα ενημερώνοντας το χρήστη για την ημερομηνία, τον τύπο και την περιγραφή του συμβάντος. Στην περίπτωση που ο χρήστης είναι διαχειριστής ή καθηγητής, μπορεί επίσης να προσθέτει συμβάντα σχετικά με το σχολείο ή την τάξη (πχ. Παρέλαση 28^{ης} Οκτωβρίου, διαγώνισμα μαθηματικών Δ' τάξης). Ο μαθητής έχει και πάλι τα ελάχιστα δικαιώματα, καθώς μπορεί μόνο να προβάλλει το ημερολόγιο και τα συμβάντα που εμφανίζονται, και όχι να προσθέσει.



Εικόνα 46 - Το ημερολόγιο καθηγητή – μαθητή

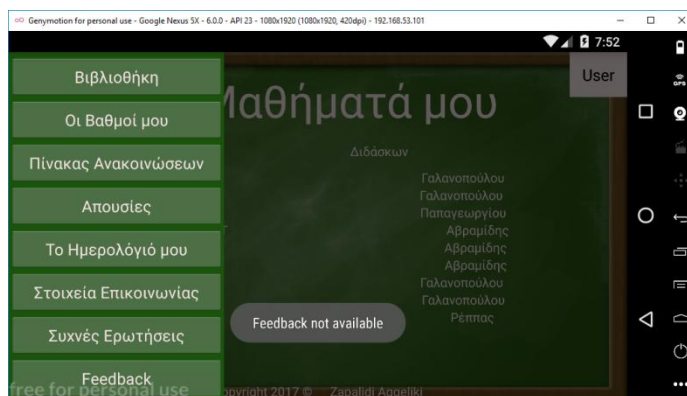
Η εφαρμογή περιέχει επίσης δύο στατικά activities με πληροφορίες για την εφαρμογή, το σκοπό της και λίγα λόγια για τη θεωρία στην οποία στηρίχθηκε (activity Συχνών Ερωτήσεων) και ένα activity με χρήσιμα στοιχεία επικοινωνίας με το σχολείο, το υπουργείο παιδείας, τους καθηγητές και το τεχνικό τμήμα διαχείρισης της εφαρμογής (activity Στοιχείων Επικοινωνίας).



Εικόνα 47 - Συχνές Ερωτήσεις

Εικόνα 48 - Στοιχεία Επικοινωνίας

Τέλος, ένα απαραίτητο και πολύ σημαντικό τμήμα της εφαρμογής είναι αυτό του feedback. Για ακόμα μια φορά, καλό θα ήταν να διευκρινίσουμε ότι δεν πρόκειται για feedback επί της εφαρμογής και της χρήσης της (αν και θα μπορούσε να περιέχεται και αυτό το ερώτημα), αλλά σε θέματα των επιδόσεων του σχολείου. Όπως ήδη αναφέραμε, το σχολείο είναι ένα πολύπλοκο προσαρμοζόμενο σύστημα (CAS), του οποίου η διοίκηση γίνεται αποτελεσματικότερα με λήψη αποφάσεων βάσει δεδομένων. Τα δεδομένα αυτά συλλέγονται με τη λειτουργία ακριβώς του feedback, καθώς ανακυκλώνονται στο εσωτερικό του συστήματος μέσω των πρακτόρων του. Στο activity αυτό, οι χρήστες παρέχουν τη γνώμη τους σχετικά με θέματα της οργάνωσης του σχολείου ως οργανισμού, των καθηγητών και διοικητικού προσωπικού σε θέματα διοίκησης αλλά και στη διαδικασία της εκπαίδευσης των μαθητών από τους καθηγητές και το πρόγραμμα του σχολείου.



Εικόνα 49 - Ανεργή επιλογή Feedback



Εικόνα 50 - Ερωτήσεις Feedback Μαθητή

μαθητής καλείται να απαντήσει μαζί με τον κηδεμόνα του, βαθμολογώντας τον υπεύθυνο καθηγητή, τις μεθόδους διδασκαλίας του, τη μεταδοτικότητα, την προθυμία του κ.ο.κ., καθώς και τη λειτουργία του σχολείου, την οργάνωση, τις παροχές κλπ.

Οι διαχειριστές έπειτα συγκεντρώνουν τα δεδομένα που συλλέχθηκαν από την έρευνα αυτή, και η ομάδα ηγεσίας του σχολείου τα επεξεργάζεται ώστε να βγάλει συμπεράσματα και αποφάσεις σχετικά με τη στρατηγική και τη διοίκηση του σχολικού περιβάλλοντος.

Όπως ήδη έχουμε δει, η επιλογή του feedback δεν είναι πάντα διαθέσιμη στους χρήστες, προκειμένου να παρέχουν τις απαντήσεις τους, αλλά γίνεται διαθέσιμη με την αντίστοιχη επιλογή του διαχειριστή. Αυτό συμβαίνει ώστε να έχουν το χρόνο και τις γνώσεις που αποκόμισαν μέσα στη χρονιά οι διαχειριστές για να συντάξουν τις κατάλληλες ερωτήσεις για τους μαθητές και τους καθηγητές, αλλά και να μην υπάρχει περίσσεια δεδομένων και υπερβολική εισροή που μπορεί να κάνει τις λειτουργίες αργές και την επεξεργασία των δεδομένων δύσκολη. Οι χρήστες ενημερώνονται μέσω e-mail ότι η πλατφόρμα feedback είναι ανοιχτή και για πόσο διάστημα, ενώ παρακαλούνται να συμμετάσχουν στην παροχή κριτικής, καθώς είναι αναγκαία και πολύτιμη για τη σχολική ηγεσία.

Στην περίπτωση που η επιλογή είναι ανενεργή, το κουμπί του μενού δεν παραπέμπει κάπου το χρήστη, ενώ εμφανίζεται μήνυμα (toast) που τον ενημερώνει ότι δεν υπάρχουν ερωτήσεις προς απάντηση.

Στην περίπτωση όμως που υπάρχει η δυνατότητα απάντησης στις ερωτήσεις, εμφανίζονται στο χρήστη καρτέλες ερωτήσεων ανάλογα με το ρόλο του, οι οποίες αφορούν σε διάφορα θέματα και επίπεδα του σχολείου. Ο καθηγητής καλείται να βαθμολογήσει την καλή συνεργασία, επικοινωνία και συμμετοχή συναδέλφων του, την αποτελεσματικότητα της διεύθυνσης του σχολείου και την αποδοτικότητα των μεθόδων διδασκαλίας και οργάνωσης σε διάφορα θέματα διαχείρισης. Ο



ΚΕΦΑΛΑΙΟ 9^ο: ΣΥΜΠΕΡΑΣΜΑΤΑ

Μετά την ενδελεχή μελέτη μας, την ανασκόπηση υπάρχουσών θεωρητικών μελετών και πληθώρας πληροφοριών πάνω στη σχολική ηγεσία, γίνεται προφανής η μεγάλη σημασία της στην ομαλή λειτουργία ενός σχολικού περιβάλλοντος και ενός εκπαιδευτικού οργανισμού πρωτοβάθμιας εκπαίδευσης. Η εφαρμογή των μεθόδων που μελετήσαμε βοηθά στη **διευκόλυνση** λήψης αποφάσεων, αλλά και στη **βελτίωση** των **αποφάσεων** αυτών βάσει των αναγκών του σχολείου και των μαθητών. Με την εφαρμογή της σχολικής ηγεσίας σε ένα σύστημα πολυπλοκότητας όπως είναι το σχολικό περιβάλλον, φαίνεται καθαρά η χρησιμότητα ηγεσίας βάση συμπερασμάτων. Όπως μάθαμε, η σχολική ηγεσία αποτελεί συνδυασμό της διαχειριστικής ηγεσίας και της ηγεσίας πολυπλοκότητας. Η υβριδική αυτή προσέγγιση επιφέρει βέλτιστα αποτελέσματα σε ένα σχολικό περιβάλλον, αφού συνδυάζει λήψη αποφάσεων βάση της ροής δεδομένων εντός του συστήματος και την υποστήριξη του διαχειριστή του συστήματος, δηλαδή τις διεύθυνσης του σχολείου. Με αυτό τον τρόπο ο συντονισμός γίνεται αποτελεσματικά αλλά και η διαχείριση δεν είναι γραμμική αλλά **προσαρμοστική** και ευέλικτη. Η ηγεσία στηρίζεται στην παρατήρηση των καταστάσεων και της βελτίωσης του σχολείου, όπως φαίνεται από τα δεδομένα που συλλέγονται και επεξεργάζονται κατά τη ροή τους στο σύστημα. Σε κάθε αλλαγή μεταξύ των συντελεστών του συστήματος τις αλληλεπιδράσεις τους μεταξύ τους και με το ίδιο το σχολείο, αλλά και σε εξωτερικές μεταβολές, το σύστημα προσαρμόζεται και η στρατηγική σχεδιάζεται αναλόγως. Μπορούμε με ασφάλεια να πούμε ότι πρόκειται για μία **αποτελεσματικότερη** μέθοδο ηγεσίας, σχεδιασμένη τόσο για συστήματα πολυπλοκότητας όσο και για συγκεκριμένα συστήματα σχολικού περιβάλλοντος της πρωτοβάθμιας εκπαίδευσης.

Όσον αφορά την **εφαρμογή σύγχρονων τεχνολογιών** στην εκπαίδευση, η υιοθέτησή τους δεν είναι καινούργιο φαινόμενο. Εδώ και πολλά χρόνια εκπαιδευτικοί σε κάθε τομέα επιστρατεύουν τεχνολογίες κατά τη διάρκεια τόσο της διδασκαλίας όσο και της αξιολόγησης, και γενικότερα κατά τη διαδικασία της μάθησης. Οι νέες τεχνολογίες εξάπτουν του ενδιαφέροντος του μαθητή και διευκολύνουν τη διαδικασία της εκπαίδευσης, ενώ φέρνουν νέο διαδραστικό υλικό στη διαδικασία της μάθησης. Όσον αφορά δε τη σχολική ηγεσία, η εφαρμογή των τεχνολογιών αυτών θα μπορούσε να έχει θετικά αποτελέσματα και σε αυτό τον τομέα. Συγκεκριμένα, να διευκολύνει τόσο την



αλληλεπίδραση μεταξύ των παραγόντων του συστήματος όσο και την συλλογή και αποθήκευση δεδομένων που χρειάζονται για τη λήψη αποφάσεων. Όπως αναφέραμε ήδη, η ροή δεδομένων εντός του συστήματος του σχολικού οργανισμού είναι ζωτικής σημασίας για τη σχολική ηγεσία. Η εφαρμογή που κατασκευάσαμε στα πλαίσια της εργασίας αυτής είναι μεν ένα μικρό παράδειγμα ενός πρώτου βαθμού σχεδιασμού και υλοποίησης και κατασκευής μιας εφαρμογής σχολικής ηγεσίας, ωστόσο δείχνει πως ακόμα και μία απλή σχετικά εφαρμογή μπορεί να βοηθήσει σε μεγάλο βαθμό τη συγκέντρωση των βασικών λειτουργιών διδασκαλίας, αλλά και στη συγκέντρωση δεδομένων που είναι απαραίτητα σε όλα τα επίπεδα για τη λήψη αποφάσεων και σχεδιασμό στρατηγικής. Επομένως, με τη διαρκή εξέλιξη των τεχνολογιών και του Internet of Things, μπορεί κανείς να προβλέψει ότι σε λίγα μόνο χρόνια η υιοθέτηση τεχνολογιών στην εκπαίδευση όχι μόνο θα είναι ο κανόνας και όχι η εξαίρεση, αλλά επίσης θα φέρει σπουδαίες καινοτομίες στο χώρο της εκπαίδευσης αλλά και της σχολικής ηγεσίας. Μάλιστα, μια μεταγενέστερη εφαρμογή θα μπορούσε ακόμα και να διαχειρίζεται πλήρως αυτά τα δεδομένα για κάθε οργανισμό, να εκτελεί προβλέψεις και να φέρνει στην ηγετική ομάδα όλα τα δεδομένα που χρειάζονται για το πλήρη σχεδιασμό πλάνου βελτίωσης του σχολείου.

Όπως είναι προφανές, η υλοποίηση σε εφαρμογή σχολικής ηγεσίας επιδέχεται ακόμα πολλές προσθήκες και βελτιώσεις, καθώς και την υιοθέτηση επιπλέον λογισμικού, τεχνολογιών και εξοπλισμού. Ορισμένα παραδείγματα βελτιώσεων σε επόμενο στάδιο της εφαρμογής θα μπορούσαν να είναι:

- Το να γίνει πιο προσωπική για τον κάθε χρήστη, παρέχοντας περισσότερες επιλογές εξατομίκευσης του προφίλ, της ίδιας της εμφάνισης της εφαρμογής, προσθήκης επιπλέον χαρακτηριστικών όσον αφορά το χρήστη κλπ.
- Η προσθήκη πλατφόρμας επικοινωνίας τύπου forum συζητήσεων, υπό την επίβλεψη των υπευθύνων για την καταλληλότητα του περιεχομένου, καθώς και ίσως μιας πλατφόρμας άμεσων μηνυμάτων, και επικοινωνίας με τον καθηγητή για επίλυση αποριών, απομακρυσμένη εκπαίδευση και άλλα.
- Συμπερίληψη επιπλέον συσκευών hardware (προτζέκτορες, έξυπνους πίνακες) και άλλων αντικειμένων (τεχνολογιών IoE) για εμπλουτισμό της εκπαιδευτικής διαδικασίας αλλά και πληρέστερης συλλογής δεδομένων.



- Επιπλέον επιλογές παρατήρησης συμπεριφοράς και διάθεσης του μαθητή, προς εξόρυξη γνώσεων (πχ bullying εντός του σχολείου, προβλήματα στο σπίτι κ.ά.)
- Εκτός της συλλογής δεδομένων, θα μπορούν να επιστρατευθούν και άλλες τεχνολογίες, λογισμικό και υλικό για την ανάλυση των δεδομένων, ώστε να γίνεται αυτόματα ο προϋπολογισμός εσόδων/εξόδων και δαπανών, η έγκαιρη καταγραφή ζημιών κι αναγκών του σχολείου, ακόμα και η πλήρης πρόβλεψη μελλοντικών καταστάσεων που επηρεάζουν τη λειτουργία του σχολικού οργανισμού, πχ. αποχώρηση μαθητών, ανάγκη ανανέωσης ή αντικατάστασης εξοπλισμού, δυσσαρέσκεια κοινού (γονέων) κ.ά.
- Επίσης, θα είναι δυνατή η επιστράτευση τεχνολογιών στη διαδικασία εκπαίδευσης ατόμων με ειδικές ανάγκες (πχ. τυφλοί/κωφοί μαθητές).

Θα μπορούσε κανείς εύλογα να αναρωτηθεί πόσο εύκολη θα ήταν υιοθέτηση τέτοιων τεχνολογιών στα υπάρχοντα εκπαιδευτικά συστήματα, πόσο μάλλον σε χώρες όπως η Ελλάδα. Ωστόσο, καθώς τα πλεονεκτήματα τέτοιων τεχνολογιών γίνονται ήδη φανερά από μικρότερης κλίμακας χρήσης τους στα εκπαιδευτικά συστήματα άλλων χωρών, αλλά και όπως φαίνεται από τις δυνατότητες συνεχούς εξέλιξης τους, είναι βέβαιο ότι το επόμενο σκαλί στην κλίμακα εξέλιξης της εκπαίδευσης θα περιλαμβάνει τέτοιες τεχνολογίες, όπως επίσης και στην διοίκηση και ηγεσία συστημάτων σε κάθε τομέα, συμπεριλαμβανομένου και του σχολικού περιβάλλοντος πρωτοβάθμιας εκπαίδευσης.



ΒΙΒΛΙΟΓΡΑΦΙΑ

1. **Demetrios G. Sampson** (Curtin University), **Stylios Sergis** (University of Piraeus) - «*School Analytics: A Framework for Supporting School Complexity Leadership*»: ResearchGate, December 2015
https://www.cisco.com/c/dam/en_us/solutions/industries/docs/education/education_interint.pdf
2. **Michelle Selinger, Ana Sepulveda, Jim Buchan** - «*Education and the Internet of Everything*»: (CISCO) Cisco Consulting Services and Cisco EMEAR Education Team, October 2013
https://www.researchgate.net/publication/282673322_School_Analytics_A_Framework_for_Supporting_School_Complexity_Leadership
3. **Kenneth Leithwood, Karen Seashore Lewis, Stephen Anderson, Kyla Wahlstorm** – «*How Leadership Influences student learning*» - The Wallace Foundation, 2004
<http://www.wallacefoundation.org/knowledge-center/school-leadership/Pages/default.aspx>
4. **Leslie M. Anderson, Brenda J. Turnbull, Erikson R. Arcaira** – «*Leader Tracking Systems: Turning Data into Information for School Leadership*» - The Wallace Foundation, June 2017
<http://www.wallacefoundation.org/knowledge-center/Pages/Leader-Tracking-Systems-Turning-Data-Into-Information-for-School-Leadership.aspx>
5. «What is a Complex Adaptive System? » - Project Guts -
https://code.org/curriculum/science/files/CS_in_Science_Background_papers.pdf
6. **Lars Vogel** - «*Getting started with Android development - Tutorial*» - Vogella GmbH, version 14.4, 20.06.20161.
<http://www.vogella.com/tutorials/Android/article.html>
7. **Karen Hawley Miles, Stephen Frank** – «*The Strategic School: Making the Most out of People, Time and Money*» - Corwin Press, 2008



8. **Hobarat L. Harmon, Kai Schafft** - «*Rural School Leadership for Collaborative Community Development* » - The Rural Educator, 2009
9. **Α. Πρέντζα - Ε. Στουγιάννου** - Διαφάνειες μαθήματος «(ΨΣ-528) ΤΕΧΝΟΛΟΓΙΑ ΛΟΓΙΣΜΙΚΟΥ ΚΑΙ ΑΝΑΛΥΣΗ ΣΥΣΤΗΜΑΤΩΝ»,
<https://evdoxos.ds.unipi.gr/modules/document/?course=DS102>
10. **Jeff Dunn** – «*6 reasons to choose Android over iPhone*» – Business Insider -
<http://www.businessinsider.com/why-android-over-iphone-2016-10>
11. *Complex adaptive system* - From Wikipedia, the free encyclopedia -
https://en.wikipedia.org/wiki/Complex_adaptive_system
12. *Android*: Από τη Βικιπαίδεια, την ελεύθερη εγκυκλοπαίδεια -
<https://el.wikipedia.org/wiki/Android>
13. *Ιστορικό Εκδόσεων Android*: Από τη Βικιπαίδεια, την ελεύθερη εγκυκλοπαίδεια -
https://el.wikipedia.org/wiki/Ιστορία_εκδόσεων_του_Android
14. *Android Studio*: From Wikipedia, the free encyclopedia -
https://en.wikipedia.org/wiki/Android_Studio
15. *Using Android Studio*: GitHub Wiki -
https://github.com/codepath/android_guides/wiki/Using-Android-Studio
16. *Eclipse Andmore* - <https://projects.eclipse.org/projects/tools.andmore>
17. *Android Developers* – Activity -
<https://developer.android.com/reference/android/app/Activity.html>
18. *Android Developers* – Activities -
<https://developer.android.com/guide/components/activities/index.html>
19. *Android Developers* – Activity Lifecycle -
<https://developer.android.com/reference/android/app/Activity.html#ActivityLifecycle>



20. *Google Developers* – Common Android Views cheat sheet – Udacity -
<http://labs.udacity.com/images/Common-Android-Views-Cheat-Sheet.pdf>
21. *Bluestacks* – List of Android Devices - <https://www.bluestacks.com/blog/app-reviews/archive/list-of-android-devices.html>
22. *AndroidBeat* - <http://www.androidbeat.com/2014/07/install-android-l-fonts-device/>
23. *Android Developers* – RecyclerView -
<https://developer.android.com/guide/topics/ui/layout/recyclerview.html>
24. *Android Developers* – GridView –
<https://developer.android.com/guide/topics/ui/layout/gridview.html>
25. *Phone Arena* – Gmail - https://www.phonearena.com/news/How-to-delete-messages-in-Gmail-with-a-simple-swipe-Android_id65341
26. *StackOverflow* – GridView vs. GridLayout -
<https://stackoverflow.com/questions/11307218/gridview-vs-gridlayout-in-android-apps>
27. *Balsamiq Mockup* - <https://balsamiq.com/products/mockups/>



ΠΑΡΑΡΤΗΜΑ – Πηγαίος Κώδικας της Εφαρμογής

App.java

```
1 package com.unipi.angzap.schoolleadershipapp;
2
3 import android.app.Application;
4
5 import com.parse.Parse;
6
7 public class App extends Application{
8     @Override
9     public void onCreate() {
10        super.onCreate();
11        Parse.initialize(new Parse.Configuration.Builder(this)
12            .applicationId("hkjbmsSSDAFSAjhfkhfk")
13            .server("https://school-leadership-
14 app.herokuapp.com/parse/")
15            .clientKey("UIOUFOLKJLk1kjsdlfj") // set explicitly
16            unless clientKey is explicitly configured on Parse server
17            .build()
18        );
19    }
20 }
```

activity_main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:background="@drawable/blackboard">
7
8     <ImageView
9         android:layout_width="match_parent"
10        android:layout_height="wrap_content"
11        android:layout_margin="30dp"
12        android:id="@+id/title"
13        android:src="@drawable/logo"/>
14
15
16     <TextView
17         android:layout_width="wrap_content"
18         android:layout_height="wrap_content"
19         android:text="@string/welcometext"
20         android:id="@+id/welcometext"
21         android:layout_marginTop="50dp"
22         android:textSize="20sp"
23         android:layout_below="@+id/title"
24         android:layout_centerHorizontal="true"
25         android:textColor="@color/chalk"/>
26
27     <LinearLayout
28         android:layout_width="wrap_content"
29         android:layout_height="wrap_content"
30         android:orientation="horizontal"
31         android:layout_below="@+id/welcometext">
```



```
32 android:layout_centerHorizontal="true">
33 <Button
34 android:id="@+id/activity_main_sign_up_button"
35 android:layout_width="wrap_content"
36 android:layout_height="wrap_content"
37 android:layout_margin="10dp"
38 android:text="Sign Up"
39 android:background="@color/chalk"/>
40 <Button
41 android:id="@+id/activity_main_login_button"
42 android:layout_width="wrap_content"
43 android:layout_height="wrap_content"
44 android:layout_margin="10dp"
45 android:text="Log in"
46 android:background="@color/chalk"/>
47 </LinearLayout>
48
49 <LinearLayout
50 android:layout_width="wrap_content"
51 android:layout_height="wrap_content"
52 android:layout_alignParentBottom="true"
53 android:layout_centerHorizontal="true">
54 <TextView
55 android:layout_width="wrap_content"
56 android:layout_height="wrap_content"
57 android:text="@string/copyright_2017"
58 android:layout_marginRight="10dp"
59 android:textColor="@color/chalk" />
60 <TextView
61 android:layout_width="wrap_content"
62 android:layout_height="wrap_content"
63 android:text="@string/footerMyName"
64 android:layout_marginLeft="10dp"
65 android:textColor="@color/chalk" />
66 </LinearLayout>
67
68 </RelativeLayout>
```



MainActivity.java

```
1 package com.unipi.angzap.schoolleadershipapp;
2 import android.app.Activity;
3 import android.content.Intent;
4 import android.os.Bundle;
5 import android.util.Log;
6 import android.view.View;
7 import android.widget.Button;
8 import com.parse.FindCallback;
9 import com.parse.ParseException;
10 import com.parse.ParseObject;
```



```
11 import com.parse.ParseQuery;
12 import com.parse.ParseUser;
13 import java.util.ArrayList;
14 import java.util.List;
15
16 public class MainActivity extends Activity {
17     private Button mLoginButton;
18     private Button mSignUpButton;
19     @Override
20     protected void onCreate(Bundle savedInstanceState) {
21         super.onCreate(savedInstanceState);
22         setContentView(R.layout.activity_main);
23
24         if(ParseUser.getCurrentUser() != null){
25             Log.d("conn", "I am logged in!");
26             Intent i = new Intent(MainActivity.this, CoursesActivity.class);
27             startActivity(i);
28             finish();
29         }
30         else{
31             Log.d("conn", "I am not logged in! You should take me to the login
32             screen.");
33         }
34
35         mLoginButton = (Button) findViewById(R.id.activity_main_login_button);
36         mLoginButton.setOnClickListener(new View.OnClickListener() {
37             @Override
38             public void onClick(View v) {
39                 Intent i = new Intent(MainActivity.this, LoginActivity.class);
40                 startActivity(i);
41             }
42         });
43
44         mSignUpButton = (Button) findViewById(R.id.activity_main_sign_up_button);
45         mSignUpButton.setOnClickListener(new View.OnClickListener() {
46             @Override
47             public void onClick(View v) {
48                 Intent i = new Intent(MainActivity.this, SignUpActivity.class);
49                 startActivity(i);
50             }
51         });
52
53         addCourseBooksIfNeeded();
54         addQuizQuestionsIfNeeded();
55         addProjectIfNeeded();
56         addFeedbackQuestionsIfNeeded();
57         addAbsenceIfNeeded();
58         addGradeIfNeeded();
59     }
60     private void addAbsenceIfNeeded(){
61         ParseQuery <ParseObject> query = ParseQuery.getQuery("Absence");
62         query.findInBackground(new FindCallback<ParseObject>() {
63             @Override
64             public void done(List<ParseObject> objects, ParseException e) {
65                 if (e == null) {
```



```
65         Log.d("absence", "e = null");
66         if (objects.size() == 0){ // 0 absences
67             Log.d("absence", "size 0");
68             String[] username = {"Μαρία Χρήστου", "Σοφία
69             Παπαδοπούλου", "Πέτρος Πέτρου", "Χρήστος Αχρηστος"};
70             int[] total = {5, 2, 3, 8};
71             int[] justified = {5, 0, 0, 3};
72
73             ArrayList<ParseObject> absenceToUpload = new
74             ArrayList<>();
75
76             for (int i = 0; i < username.length; i++){
77                 ParseObject tempParseObject = new
78                 ParseObject("Absence");
79                 tempParseObject.put("username", username[i]);
80                 tempParseObject.put("total", total[i]);
81                 tempParseObject.put("justified", justified[i]);
82
83                 absenceToUpload.add(tempParseObject);
84             }
85             for(int i = 0; i < absenceToUpload.size(); i++){
86                 Log.d("absence", "inside for loop");
87                 absenceToUpload.get(i).saveEventually();
88             }
89         }
90         else{ // Move on.
91             Log.d("absence", "size > 0");
92         }
93         else {
94             Log.d("absence", "e != null");
95         }
96     });
97     private void addFeedbackQuestionsIfNeeded() {...}
143    private void addQuizQuestionsIfNeeded() {...}
272    private void addProjectIfNeeded() {...}
332    private void addCourseBooksIfNeeded() {...}
668    private void addGradeIfNeeded() {...}
743 }
```

activity_sign_up.xml

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <RelativeLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:background="@drawable/blackboard">
7
8     <TextView
9         android:layout_width="wrap_content"
10        android:layout_height="60dp"
11        android:text="@string/signUpHeader"
12        android:textSize="50sp"
```




```
13         android:layout_centerHorizontal="true"
14         android:layout_margin="20dp"
15         android:id="@+id/header"
16         android:textColor="@color/chalk"/>
17
18     <LinearLayout
19         android:layout_width="match_parent"
20         android:layout_height="wrap_content"
21         android:layout_below="@+id/header"
22         android:id="@+id/form">
23
24         <LinearLayout
25             android:layout_width="140dp"
26             android:layout_height="wrap_content"
27             android:orientation="vertical"
28             android:layout_marginLeft="14dp"
29             android:id="@+id/selection"
30             android:layout_marginStart="14dp">
31
32             <TextView
33                 android:layout_width="wrap_content"
34                 android:layout_height="wrap_content"
35                 android:text="Επιλογές:"
36                 android:textColor="@color/chalk"
37                 android:textSize="20sp"/>
38
39             <RadioGroup
40                 android:layout_width="fill_parent"
41                 android:layout_height="wrap_content"
42                 android:id="@+id/activity_sign_up_role_radio_group"
43                 android:orientation="vertical" >
44
45                 <RadioButton
46                     android:id="@+id/activity_sign_up_admin_radiobutton"
47                     android:layout_width="wrap_content"
48                     android:layout_height="wrap_content"
49                     android:text="@string/admin"
50                     android:textColor="@color/chalk" />
51                 <RadioButton
52                     android:id="@+id/activity_sign_up_teacher_radiobutton"
53                     android:layout_width="wrap_content"
54                     android:layout_height="wrap_content"
55                     android:text="@string/teacher"
56                     android:textColor="@color/chalk" />
57                 <RadioButton
58                     android:id="@+id/activity_sign_up_student_radiobutton"
59                     android:layout_width="wrap_content"
60                     android:layout_height="wrap_content"
61                     android:text="@string/student"
62                     android:textColor="@color/chalk" />
63
64             </RadioGroup>
65     </LinearLayout>
66
```



```
67     <LinearLayout
68         android:layout_width="wrap_content"
69         android:layout_height="wrap_content"
70         android:layout_alignParentRight="true"
71         android:orientation="vertical"
72         android:layout_margin="15dp">
73
74     <LinearLayout
75         android:layout_width="wrap_content"
76         android:layout_height="wrap_content">
77
78         <TextView
79             android:layout_width="wrap_content"
80             android:layout_height="wrap_content"
81             android:textColor="@color/chalk"
82             android:text="Διεύθυνση ηλ. ταχυδρομείου" />
83
84         <EditText
85             android:layout_width="200dp"
86             android:id="@+id/activity_sign_in_email_edittext"
87             android:textColor="@color/chalk"
88             android:layout_height="wrap_content"
89             android:hint="@string/email" />
90
91     </LinearLayout>
92
93     <LinearLayout
94         android:layout_width="match_parent"
95         android:layout_height="wrap_content">
96
97         <TextView
98             android:layout_width="wrap_content"
99             android:layout_height="wrap_content"
100            android:textColor="@color/chalk"
101            android:text="Κωδικός Πρόσβασης" />
102
103         <EditText
104             android:layout_width="200dp"
105             android:id="@+id/activity_sign_inpassword_edittext"
106             android:layout_height="wrap_content"
107             android:textColor="@color/chalk"
108             android:hint="@string/pswd" />
109
110     </LinearLayout>
111
112     <LinearLayout
113         android:layout_width="match_parent"
114         android:layout_height="wrap_content">
115
116         <TextView
117             android:layout_width="wrap_content"
118             android:layout_height="wrap_content"
119             android:textColor="@color/chalk"
120            android:text="Όνοματεπώνυμο Χρήστη" />
121
122         <EditText
123             android:layout_width="200dp"
```



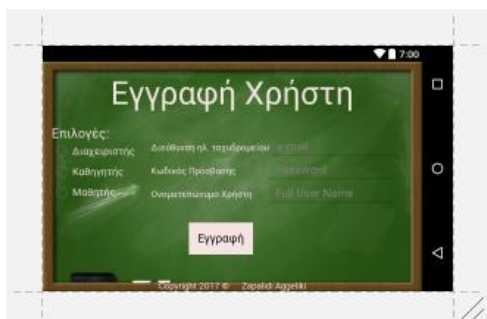
```
121         android:id="@+id/activity_sign_in_fullusername_edittext"
122         android:layout_height="wrap_content"
123         android:textColor="@color/chalk"
124         android:hint="Full User Name" />
125
126     </LinearLayout>
127
128 </LinearLayout>
129
130 </LinearLayout>
131
132 <Button
133     android:id="@+id/activity_sign_up_signUp_button"
134     android:layout_width="100dp"
135     android:layout_height="50dp"
136     android:layout_marginLeft="47dp"
137     android:layout_marginStart="47dp"
138     android:text="@string/signUp"
139     android:background="@color/chalk"
140     android:layout_marginTop="12dp"
141     android:layout_below="@+id/form"
142     android:layout_alignLeft="@+id/linearLayout"
143     android:layout_alignStart="@+id/linearLayout" />
144
145 <TextView
146     android:layout_width="wrap_content"
147     android:layout_height="wrap_content"
148     android:id="@+id/activity_sign_up_error_alert_textview"
149     android:textColor="@color/alert"
150     android:layout_toRightOf="@+id/activity_sign_up_signUp_button"
151     android:layout_alignTop="@+id/activity_sign_up_signUp_button"
152     android:layout_marginLeft="10dp"
153     android:layout_marginStart="10dp"
154     android:layout_marginRight="20dp"
155     android:layout_marginEnd="20dp"
156     android:layout_marginBottom="20dp"/>
157
158 <LinearLayout
159     android:layout_width="wrap_content"
160     android:layout_height="wrap_content"
161     android:layout_alignParentBottom="true"
162     android:layout_centerHorizontal="true"
163     android:id="@+id/linearLayout">
164     <TextView
165         android:layout_width="wrap_content"
166         android:layout_height="wrap_content"
167         android:text="@string/copyright_2017"
168         android:textColor="@color/chalk"
169         android:layout_marginRight="10dp" />
170     <TextView
171         android:layout_width="wrap_content"
172         android:layout_height="wrap_content"
173         android:text="@string/footerMyName"
174         android:textColor="@color/chalk"
```



```

175         android:layout_marginLeft="10dp" />
176     </LinearLayout>
177
178 </RelativeLayout>

```



SignUpActivity.java

```

1  package com.unipi.angzap.schoolleadershipapp;
2
3  import [ . . . ]
4
25 public class SignUpActivity extends Activity {
26
27     private EditText emailET, passwordET, fullnameET;
28     private Button signUpButton;
29     private RadioGroup roleRG;
30     private RadioButton adminRB, teacherRB, studentRB;
31     private TextView errorAlertTV;
32     private ProgressDialog mProgressDialog;
33
34     @Override
35     protected void onCreate(Bundle savedInstanceState) {
36         super.onCreate(savedInstanceState);
37         setContentView(R.layout.activity_sign_up);
38
39         emailET = (EditText) findViewById(R.id.activity_sign_in_email_edittext);
40         passwordET = (EditText)
41             findViewById(R.id.activity_sign_inpassword_edittext);
42         fullnameET = (EditText)
43             findViewById(R.id.activity_sign_in_fullusername_edittext);
44         roleRG = (RadioGroup) findViewById(R.id.activity_sign_up_role_radio_group);
45         adminRB = (RadioButton)
46             findViewById(R.id.activity_sign_up_admin_radiobutton);
47         teacherRB = (RadioButton)
48             findViewById(R.id.activity_sign_up_teacher_radiobutton);
49         studentRB = (RadioButton)
50             findViewById(R.id.activity_sign_up_student_radiobutton);
51         errorAlertTV = (TextView)
52             findViewById(R.id.activity_sign_up_error_alert_textview);
53
54         signUpButton = (Button) findViewById(R.id.activity_sign_up_signUp_button);
55         signUpButton.setOnClickListener(new View.OnClickListener() {
56             @Override
57             public void onClick(View v) {
58                 ParseUser user = new ParseUser();
59                 user.setUsername(fullnameET.getText() + "");
60                 user.setPassword(passwordET.getText() + "");
61                 user.setEmail(emailET.getText() + "");
62
63                 ParseQuery<ParseObject> query = ParseQuery.getQuery("Absence");

```



```
58 query.whereEqualTo("username", (fullnameET.getText() + ""));
59 query.findInBackground(new FindCallback<ParseObject>() {
60     public void done(List<ParseObject> absenceList, ParseException
61     e) {
62         if (e == null) {
63             Log.d("absence", "Does not exist" );
64         } else {
65             Log.d("absence", "Absence exists");
66         }
67     }
68 });
69
70 mProgressDialog = new ProgressDialog(SignUpActivity.this);
71 mProgressDialog.setMessage("Αποθήκευση...");
72 mProgressDialog.setCancelable(false);
73 mProgressDialog.show();
74
75 int checkedRadioButtonId = roleRG.getCheckedRadioButtonId();
76 int choice;
77 if(checkedRadioButtonId ==
78 R.id.activity_sign_up_admin_radiobutton){ // eimai Admin
79     choice = 0;
80 }
81 else if (checkedRadioButtonId ==
82 R.id.activity_sign_up_teacher_radiobutton) { //teacher
83     choice = 1;
84 }
85 else { // Student
86     choice = 2;
87 }
88
89 user.put("role", choice);
90
91 if (choice == 2) {
92     ParseObject absence = new ParseObject("Absence");
93     absence.put("username", (fullnameET.getText() + ""));
94     absence.put("total", 0);
95     absence.put("justified", 0);
96     absence.saveInBackground();
97
98     ParseQuery<ParseObject> queryGrade =
99     ParseQuery.getQuery("Grade");
100     queryGrade.whereEqualTo("username", (fullnameET.getText() +
101     ""));
102     queryGrade.findInBackground(new FindCallback<ParseObject>() {
103         public void done(List<ParseObject> gradeList,
104         ParseException e) {
105             if (e == null) {
106                 Log.d("absence", "Does not exist" );
107             } else {
108                 Log.d("absence", "Grade exists");
109             }
110         }
111     });
112     ParseObject grade1 = new ParseObject("Grade");
113     grade1.put("username", (fullnameET.getText() + ""));
114     grade1.put("courseName", "ΓΛΩΣΣΑ Ε");
115     grade1.put("gradeSemA", 0);
```



```
111         gradel.put("gradeSemB", 0);
112         gradel.put("gradeSemC", 0);
113         gradel.put("comment", "");
114         gradel.saveInBackground();
189     }
190     user.signUpInBackground(new SignUpCallback() {
191         public void done(ParseException e) {
192             mProgressDialog.dismiss();
193             if (e == null) {
194                 Log.d("conn", "successful sign up!");
195
196                 Intent i = new Intent(SignUpActivity.this,
197                                     CoursesActivity.class);
198                 startActivity(i);
199                 finish();
200             } else {
201                 // Sign up didn't succeed. Look at the ParseException
202                 // to figure out what went wrong
203                 Log.d("conn", e.getLocalizedMessage());
204
205                 if(e.getLocalizedMessage().equals("java.lang.IllegalArgumentException: Username
206                 cannot be missing or blank")){
207                     errorAlertTV.setText("Username cannot be missing or
208                     blank.");
209                 }
210                 else
211                 if(e.getLocalizedMessage().equals("java.lang.IllegalArgumentException: Password
212                 cannot be missing or blank")){
213                     errorAlertTV.setText("Password cannot be missing or
214                     blank.");
215                 }
216                 else if(e.getLocalizedMessage().equals("i/o failure")){
217                     errorAlertTV.setText("Ελλειψη σύνδεσης.");
218                 }
219                 else{
220                     errorAlertTV.setText(e.getLocalizedMessage());
221                 }
222             }
223         }
224     });
225 }
```

activity_log_in.xml

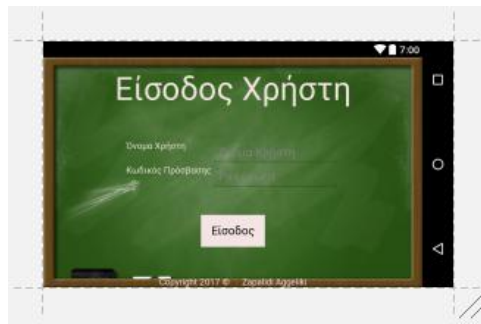
```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:background="@drawable/blackboard">
6
7     <TextView
8         android:layout_width="wrap_content"
9         android:layout_height="60dp"
10        android:text="Είσοδος Χρήστη"
11        android:textColor="@color/chalk">
```




```
12         android:textSize="50sp"
13         android:layout_centerHorizontal="true"
14         android:layout_margin="20dp"
15         android:id="@+id/headerLogIn" />
16
17     <LinearLayout
18         android:layout_width="wrap_content"
19         android:layout_height="wrap_content"
20         android:layout_centerInParent="true"
21         android:layout_below="@id/headerLogIn"
22         android:id="@+id/activity_log_in_form_linearlayout"
23         android:layout_margin="30dp"
24         android:layout_centerHorizontal="true"
25         android:gravity="center_horizontal"
26         android:orientation="horizontal">
27
28     <LinearLayout
29         android:layout_width="wrap_content"
30         android:layout_height="wrap_content"
31         android:orientation="vertical">
32         <TextView
33             android:layout_width="wrap_content"
34             android:layout_height="38dp"
35             android:text="@string/username"
36             android:textColor="@color/chalk"/>
37         <TextView
38             android:layout_width="wrap_content"
39             android:layout_height="38dp"
40             android:text="Κωδικός Πρόσβασης"
41             android:textColor="@color/chalk" />
42     </LinearLayout>
43
44     <LinearLayout
45         android:layout_width="match_parent"
46         android:layout_height="match_parent"
47         android:orientation="vertical">
48
49         <EditText
50             android:layout_width="200dp"
51             android:inputType="textWebEmailAddress"
52             android:id="@+id/activity_log_in_email_edittext"
53             android:layout_height="wrap_content"
54             android:textColor="@color/chalk"
55             android:hint="@string/username"
56             android:layout_weight="0.23" />
57         <EditText
58             android:layout_width="200dp"
59             android:layout_height="wrap_content"
60             android:id="@+id/activity_log_in_password_edittext"
61             android:textColor="@color/chalk"
62             android:hint="@string/pswd"
63             android:inputType="textPassword"
64             android:layout_weight="0.18" />
65
```



```
66     </LinearLayout>
67 </LinearLayout>
68
69 <Button
70     android:id="@+id/activity_log_in_logIn_button"
71     android:layout_width="100dp"
72     android:layout_height="50dp"
73     android:layout_margin="10dp"
74     android:text="@string/logIn"
75     android:background="@color/chalk"
76     android:layout_gravity="center_horizontal"
77     android:layout_centerHorizontal="true"
78     android:layout_below="@+id/activity_log_in_form_linearlayout"/>
79
80 <TextView
81     android:layout_width="wrap_content"
82     android:layout_height="wrap_content"
83     android:id="@+id/activity_log_in_error_alert_textview"
84     android:textColor="@color/alert"
85     android:layout_toRightOf="@+id/activity_log_in_logIn_button"
86     android:layout_alignTop="@+id/activity_log_in_logIn_button"
87     android:layout_marginLeft="10dp"
88     android:layout_marginStart="10dp"
89     android:layout_marginRight="20dp"
90     android:layout_marginEnd="20dp"
91     android:layout_marginBottom="20dp"/>
92
93 <LinearLayout
94     android:layout_width="wrap_content"
95     android:layout_height="wrap_content"
96     android:layout_alignParentBottom="true"
97     android:layout_centerHorizontal="true">
98
99     <TextView
100         android:layout_width="wrap_content"
101         android:layout_height="wrap_content"
102         android:textColor="@color/chalk"
103         android:text="@string/copyright_2017"
104         android:layout_marginRight="10dp" />
105
106     <TextView
107         android:layout_width="wrap_content"
108         android:layout_height="wrap_content"
109         android:textColor="@color/chalk"
110         android:text="@string/footerMyName"
111         android:layout_marginLeft="10dp" />
112
113 </LinearLayout>
114 </RelativeLayout>
```



LogInActivity.java

```
1 package com.unipi.angzap.schoolleadershipapp;
2 import [ . . . ]
14
15 public class LogInActivity extends Activity {
16     private EditText emailET, passwordET;
17     private TextView errorAlertTV;
18     private Button logInButton;
19     private ProgressDialog mProgressDialog;
20
21     @Override
22     protected void onCreate(Bundle savedInstanceState) {
23         super.onCreate(savedInstanceState);
24         setContentView(R.layout.activity_log_in);
25
26         emailET = (EditText) findViewById(R.id.activity_log_in_email_edittext);
27         passwordET = (EditText)
28         findViewById(R.id.activity_log_in_password_edittext);
29         errorAlertTV = (TextView)
30         findViewById(R.id.activity_log_in_error_alert_textview);
31         logInButton = (Button) findViewById(R.id.activity_log_in_logIn_button);
32         logInButton.setOnClickListener(new View.OnClickListener() {
33             @Override
34             public void onClick(View v) {
35                 ParseUser.logInInBackground(emailET.getText() + "",
36                 passwordET.getText() + "", new LogInCallback() {
37                     public void done(ParseUser user, ParseException e) {
38                         if (user != null) {
39                             // Hooray! The user is logged in.
40                             Log.d("conn", "Log in successful!");
41                             Intent i = new Intent(LogInActivity.this,
42                             CoursesActivity.class);
43                             startActivity(i);
44                             finish();
45                         } else {
46                             // Signup failed. Look at the ParseException to see
47                             what happened.
48                             Log.d("conn", "Log in unsuccessful!" +
49                             e.getLocalizedMessage());
50                             Log.d("conn", e.getLocalizedMessage());
51
52                             if(e.getLocalizedMessage().equals("java.lang.IllegalArgumentException: Username
53                             cannot be missing or blank")){
54                                 errorAlertTV.setText("Username cannot be missing
55                                 or blank.");
56                             }
57                             else
58                             if(e.getLocalizedMessage().equals("java.lang.IllegalArgumentException: Password
```



```
cannot be missing or blank")) {
    errorAlertTV.setText("Password cannot be missing
49 or blank.");
50 }
51 else if(e.getLocalisedMessage().equals("i/o
failure")) {
52     errorAlertTV.setText("Έλλειψη σύνδεσης.");
53 }
54 else {
55     errorAlertTV.setText(e.getLocalisedMessage());
56 }
57 }
58 }
59 });
60 }
61 });
62
63 //Progress Dialog Pop up
64 mProgressDialog = new ProgressDialog(LoginActivity.this);
65 mProgressDialog.setMessage("Είσοδος...");
66 mProgressDialog.setCancelable(false);
67 }
68 }
```

activity_user_profile.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.v4.widget.DrawerLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:id="@+id/activity_user_profile_drawer_layout"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent">
7
8     <!-- The main content view -->
9     <RelativeLayout
10        xmlns:android="http://schemas.android.com/apk/res/android"
11        android:layout_width="match_parent"
12        android:layout_height="match_parent"
13        android:background="@drawable/blackboard">
14
15        <ImageButton
16            android:id="@+id/activity_user_profile_menu_image_button"
17            android:layout_width="50dp"
18            android:layout_height="50dp"
19            android:layout_alignParentTop="true"
20            android:src="@drawable/menu"
21            android:background="@color/chalk"
22            android:scaleType="fitXY"/>
23
24        <Button
25            android:layout_width="50dp"
26            android:layout_height="50dp"
27            android:id="@+id/activity_user_profile_logOut_button"
28            android:text="Log Out"
29            android:background="@color/chalk"
```



```
30         android:layout_alignParentRight="true"/>
31
32     <TextView
33         android:layout_width="wrap_content"
34         android:layout_height="60dp"
35         android:text="@string/username"
36         android:textColor="@color/chalk"
37         android:textSize="50sp"
38         android:layout_centerHorizontal="true"
39         android:layout_margin="20dp"
40         android:id="@+id/headerUserProfile" />
41     <!--Εδώ θα μπει το όνομα κάθε χρήστη από τη ΒΔ -->
42
43     <LinearLayout
44         android:layout_height="wrap_content"
45         android:layout_width="match_parent"
46         android:layout_below="@+id/headerUserProfile"
47         android:gravity="center_horizontal">
48
49         <LinearLayout
50             android:layout_height="wrap_content"
51             android:layout_width="wrap_content"
52             android:orientation="vertical"
53             android:layout_margin="30dp">
54             <ImageView
55                 android:layout_height="150dp"
56                 android:layout_width="150dp"
57                 android:src="@drawable/usrpic"/>
58                 <Button
59                     android:layout_height="40dp"
60                     android:layout_width="wrap_content"
61                     android:text="Save"
62                     android:id="@+id/activity_user_profile_save_changes_button"
63                     android:background="@color/chalk"
64                 />
65
66     </LinearLayout>
67
68     <LinearLayout
69         android:layout_height="wrap_content"
70         android:layout_width="wrap_content"
71         android:orientation="vertical"
72         android:layout_gravity="center">
73
74         <LinearLayout
75             android:layout_height="wrap_content"
76             android:layout_width="wrap_content">
77             <TextView
78                 android:layout_height="wrap_content"
79                 android:layout_width="wrap_content"
80                 android:textColor="@color/chalk"
81                 android:text="Τα Γενέθλιά μου"
82             />
83             <Spinner
```



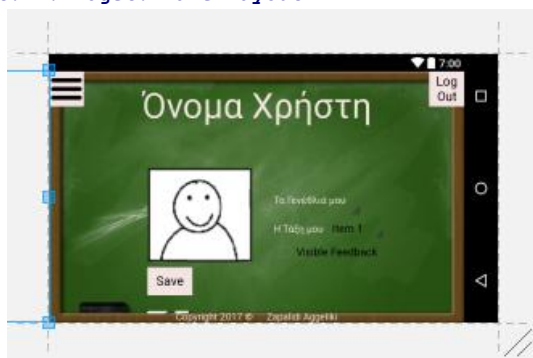
```
84         android:layout_height="wrap_content"
85         android:layout_width="wrap_content"
86     />
87 </LinearLayout>
88 <LinearLayout
89     android:layout_height="wrap_content"
90     android:layout_width="wrap_content">
91     <TextView
92         android:layout_height="wrap_content"
93         android:layout_width="wrap_content"
94         android:textColor="@color/chalk"
95         android:text="Η Τάξη μου"
96     />
97     <Spinner
98         android:id="@+id/activity_user_profile_class_spinner"
99         android:layout_width="match_parent"
100        android:layout_height="wrap_content"
101        android:spinnerMode="dialog"
102        android:entries="@array/classes_array"
103        android:prompt="@string/class_prompt" />
104 </LinearLayout>
105
106 <CheckBox
107     android:id="@+id/activity_user_profile_class_checkbox"
108     android:layout_width="wrap_content"
109     android:layout_height="wrap_content"
110     android:text="Visible Feedback"/>
111
112 </LinearLayout>
113
114 </LinearLayout>
115
116 <LinearLayout
117     android:layout_width="wrap_content"
118     android:layout_height="wrap_content"
119     android:layout_alignParentBottom="true"
120     android:layout_centerHorizontal="true">
121     <TextView
122         android:layout_width="wrap_content"
123         android:layout_height="wrap_content"
124         android:textColor="@color/chalk"
125         android:text="@string/copyright_2017"
126         android:layout_marginRight="10dp"/>
127     <TextView
128         android:layout_width="wrap_content"
129         android:layout_height="wrap_content"
130         android:textColor="@color/chalk"
131         android:text="@string/footerMyName"
132         android:layout_marginLeft="10dp"/>
133 </LinearLayout>
134
135 </RelativeLayout>
136
137 <!-- The navigation drawer -->
```




```
138 <ScrollView
139     android:layout_width="240dp"
140     android:layout_height="wrap_content"
141     android:layout_gravity="start"
142     android:id="@+id/left_drawer">
143     <LinearLayout
144         android:orientation="vertical"
145         android:layout_width="240dp"
146         android:layout_height="match_parent"
147         android:src="#eff9c181"
148         android:background="@color/colorPrimaryDark">
149
150         <Button
151             android:layout_width="match_parent"
152             android:layout_height="wrap_content"
153             android:text="@string/courses"
154             android:textColor="@color/chalk"
155             android:id="@+id/drawer_menu_courses_button"/>
156
157         <Button
158             android:layout_width="match_parent"
159             android:layout_height="wrap_content"
160             android:text="@string/bookshelf"
161             android:textColor="@color/chalk"
162             android:id="@+id/drawer_menu_bookshelf_button"/>
163
164         <Button
165             android:layout_width="match_parent"
166             android:layout_height="wrap_content"
167             android:text="@string/grades"
168             android:textColor="@color/chalk"
169             android:id="@+id/drawer_menu_grades_button"/>
170
171         <Button
172             android:layout_width="match_parent"
173             android:layout_height="wrap_content"
174             android:text="@string/announcements"
175             android:textColor="@color/chalk"
176             android:id="@+id/drawer_menu_announcements_button"/>
177
178         <Button
179             android:layout_width="match_parent"
180             android:layout_height="wrap_content"
181             android:text="@string/absence"
182             android:textColor="@color/chalk"
183             android:id="@+id/drawer_menu_absence_button"/>
184
185         <Button
186             android:layout_width="match_parent"
187             android:layout_height="wrap_content"
188             android:text="@string/calendar"
189             android:textColor="@color/chalk"
190             android:id="@+id/drawer_menu_calendar_button"/>
191
```



```
192     <Button
193         android:layout_width="match_parent"
194         android:layout_height="wrap_content"
195         android:text="@string/contactUs"
196         android:textColor="@color/chalk"
197         android:id="@+id/drawer_menu_contact_us_button"/>
198
199     <Button
200         android:layout_width="match_parent"
201         android:layout_height="wrap_content"
202         android:text="@string/faq"
203         android:textColor="@color/chalk"
204         android:id="@+id/drawer_menu_faq_button"/>
205
206     <Button
207         android:layout_width="match_parent"
208         android:layout_height="wrap_content"
209         android:text="@string/feedback"
210         android:textColor="@color/chalk"
211         android:id="@+id/drawer_menu_feedback_button"/>
212 </LinearLayout>
213 </ScrollView>
214 </android.support.v4.widget.DrawerLayout>
```



UserProfileActivity.java

```
1 package com.unipi.angzap.schoolleadershipapp;
2
3 import [ . . . ]
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28 public class UserProfileActivity extends Activity {
29     private DrawerLayout mDrawerLayout;
30     private ImageButton mMenuButton;
31     private TextView fullNameTV;
32     private Button mSaveButton;
33     private Button logOutButton;
34     private Button menuCoursesButton, menuBookshelfButton, menuGradesButton,
35     menuAnnouncementsButton, menuAbsenceButton, menuCalendarButton,
36     menuContactUsButton, menuFaqButton, menuFeedbackButton;
37     private Spinner classesSpinner;
38     private ParseUser currentUser;
39     private ProgressDialog mProgressDialog;
40     private CheckBox mCheckBox;
41     private ParseObject configObject;
42 }
```



```
41     @Override
42     protected void onCreate(Bundle savedInstanceState) {
43         super.onCreate(savedInstanceState);
44         setContentView(R.layout.activity_user_profile);
45         currentUser = ParseUser.getCurrentUser();
46         fullNameTV = (TextView) findViewById(R.id.headerUserProfile); //name
47         // (and role) text view
48         mCheckBox = (CheckBox)
49         findViewById(R.id.activity_user_profile_class_checkbox);
50         if(ParseUser.getCurrentUser().getInt("role") == 0) { //is Admin
51             mCheckBox.setVisibility(View.VISIBLE);
52             ParseQuery<ParseObject> query =
53             ParseQuery.getQuery("Configuration");
54             query.getFirstInBackground(new GetCallback<ParseObject>() {
55                 public void done(ParseObject object, ParseException e) {
56                     if (object == null) {
57                         Log.d("Checkboxerror", "No data :(");
58                     } else {
59                         configObject = object;
60                         Log.d("Checkboxesuccess",
61                         object.getBoolean("isFeedbackVisible") + "");
62                     }
63                 }
64             });
65             mCheckBox.setChecked(object.getBoolean("isFeedbackVisible"));
66         }
67         else
68             mCheckBox.setVisibility(View.GONE);
69         //Progress Dialog Pop up
70         mProgressDialog = new ProgressDialog(UserProfileActivity.this);
71         mProgressDialog.setMessage("Αποθήκευση...");
72         mProgressDialog.setCancelable(false);
73         //class select spinner
74         classesSpinner = (Spinner)
75         findViewById((R.id.activity_user_profile_class_spinner));
76         String selectedClass = currentUser.getString("class");
77         int choice = 0;
78         if(selectedClass!=null){
79             if (selectedClass.equals("ΣΤ")) {
80                 choice = 2;
81             }
82             else if (selectedClass.equals("Ε")){
83                 choice = 1;
84             }
85             else {
86                 choice = 0;
87             }
88         }
89         classesSpinner.setSelection(choice);
90         //display role in user profile layout
91         int role = ParseUser.getCurrentUser().getInt("role");
92         if(role == 0){ //Admin
93             fullNameTV.setText(ParseUser.getCurrentUser().getUsername() + " "
94             + "(Admin)");
95         }
96         else if (role ==1){ // Teacher
```



```
91         fullNameTV.setText(ParseUser.getCurrentUser().getUsername() + " "
92         + "(Teacher)");
93     }
94     else{ //Student
95         fullNameTV.setText(ParseUser.getCurrentUser().getUsername() + " "
96         + "(Std)");
97     }
98
99     mMenuButton = (ImageButton)
100    findViewById(R.id.activity_user_profile_menu_image_button);
101    mMenuButton.setOnClickListener(new View.OnClickListener() {
102        public void onClick(View v) {
103            mDrawerLayout.openDrawer(Gravity.LEFT);
104        }
105    });
106
107    mSaveButton = (Button)
108    findViewById(R.id.activity_user_profile_save_changes_button);
109    mSaveButton.setOnClickListener(new View.OnClickListener() {
110        public void onClick(View v) {
111            mProgressDialog.show();
112            String choice;
113            if(classesSpinner.getSelectedItemPosition() == 0)
114                choice = "Α";
115            else if(classesSpinner.getSelectedItemPosition() == 1)
116                choice = "Ε";
117            else
118                choice = "ΣΤ";
119            currentUser.put("class", choice);
120            currentUser.saveInBackground(new SaveCallback() {
121                @Override
122                public void done(ParseException e) {
123                    // success
124                    if(ParseUser.getCurrentUser().getInt("role") == 0)
125                        if(configObject!=null){
126                            configObject.put("isFeedbackVisible",
127                            mCheckBox.isChecked());
128                            configObject.saveInBackground(new
129                            SaveCallback() {
130                                @Override
131                                public void done(ParseException e) {
132                                    mProgressDialog.dismiss();
133                                }
134                            });
135                        }
136                    else{
137                        mProgressDialog.dismiss();
138                    }
139                }
140            });
141        }
142    });
143
144    logOutButton = (Button)
145    findViewById(R.id.activity_user_profile_logOut_button);
146    logOutButton.setOnClickListener(new View.OnClickListener() {
147        @Override
148        public void onClick(View v) {
```



```
141         ParseUser.logout();
142         Intent i = new Intent(UserProfileActivity.this,
MainActivity.class);
143         startActivity(i);
144     }
145     });
146
147     mDrawerLayout = (DrawerLayout)
findViewById(R.id.activity_user_profile_drawer_layout);
148     menuCoursesButton = (Button)
findViewById(R.id.drawer_menu_courses_button);
149     menuCoursesButton.setOnClickListener(new View.OnClickListener() {
150         @Override
151         public void onClick(View v) {
152             Intent i = new Intent(UserProfileActivity.this,
CoursesActivity.class);
153             startActivity(i);
154         }
155     });
156
157     menuBookshelfButton = (Button)
findViewById(R.id.drawer_menu_bookshelf_button);
158     menuBookshelfButton.setOnClickListener(new View.OnClickListener() {
159         @Override
160         public void onClick(View v) {
161             Intent i = new Intent(UserProfileActivity.this,
BookshelfActivity.class);
162             startActivity(i);
163         }
164     });
165
166     menuGradesButton = (Button)
findViewById(R.id.drawer_menu_grades_button);
167     menuGradesButton.setOnClickListener(new View.OnClickListener() {
168         @Override
169         public void onClick(View v) {
170             int role = ParseUser.getCurrentUser().getInt("role");
171             if(role == 0 || role ==1){ //Admin
172                 Intent i = new Intent(UserProfileActivity.this,
GradesTeacherActivity.class);
173                 startActivity(i);
174             }
175             else{ //Student
176                 Intent i = new Intent(UserProfileActivity.this,
GradesStudentActivity.class);
177                 startActivity(i);
178             }
179         }
180     });
181
182     menuAnnouncementsButton = (Button)
findViewById(R.id.drawer_menu_announcements_button);
183     menuAnnouncementsButton.setOnClickListener(new View.OnClickListener()
184     {
185         @Override
186         public void onClick(View v) {
187             int role = ParseUser.getCurrentUser().getInt("role");
188             if(role == 0 || role ==1){ //Admin
189                 Intent i = new Intent(UserProfileActivity.this,
AnnouncementsTeacherActivity.class);
```



```
189         startActivity(i);
190     }
191     else{ //Student
192         Intent i = new Intent(UserProfileActivity.this,
AnnouncementsStudentActivity.class);
193         startActivity(i);
194     }
195     }
196     });
197
198     menuAbsenceButton = (Button)
findViewById(R.id.drawer_menu_absence_button);
199     menuAbsenceButton.setOnClickListener(new View.OnClickListener() {
200         @Override
201         public void onClick(View v) {
202             int role = ParseUser.getCurrentUser().getInt("role");
203             if(role == 0 || role ==1){ //Admin
204                 Intent i = new Intent(UserProfileActivity.this,
AbsenceTeacherActivity.class);
205                 startActivity(i);
206             }
207             else{ //Student
208                 Intent i = new Intent(UserProfileActivity.this,
AbsenceStudentActivity.class);
209                 startActivity(i);
210             }
211         }
212     });
213
214     menuCalendarButton = (Button)
findViewById(R.id.drawer_menu_calendar_button);
215     menuCalendarButton.setOnClickListener(new View.OnClickListener() {
216         @Override
217         public void onClick(View v) {
218             Intent i = new Intent(UserProfileActivity.this,
CalendarActivity.class);
219             startActivity(i);
220         }
221     });
222
223     menuContactUsButton = (Button)
findViewById(R.id.drawer_menu_contact_us_button);
224     menuContactUsButton.setOnClickListener(new View.OnClickListener() {
225         @Override
226         public void onClick(View v) {
227             Intent i = new Intent(UserProfileActivity.this,
ContactUsActivity.class);
228             startActivity(i);
229         }
230     });
231
232     menuFaqButton = (Button) findViewById(R.id.drawer_menu_faq_button);
233     menuFaqButton.setOnClickListener(new View.OnClickListener() {
234         @Override
235         public void onClick(View v) {
236             Intent i = new Intent(UserProfileActivity.this,
FaqActivity.class);
237             startActivity(i);
```




```
238         }
239     });
240     menuFeedbackButton = (Button)
241     findViewById(R.id.drawer_menu_feedback_button);
242     menuFeedbackButton.setOnClickListener(new View.OnClickListener() {
243         @Override
244         public void onClick(View v){
245             ParseQuery<ParseObject> query =
246             ParseQuery.getQuery("Configuration");
247             query.getFirstInBackground(new GetCallback<ParseObject>() {
248                 public void done(ParseObject object, ParseException e) {
249                     if (object == null) {
250                         Log.d("Checkboxerror", "No data :(");
251                     } else {
252                         Log.d("Checkboxesuccess",
253                             object.getBoolean("isFeedbackVisible") + "");
254                         if(object.getBoolean("isFeedbackVisible")){
255                             Intent i = new
256                             Intent(UserProfileActivity.this, FeedbackActivity.class);
257                             startActivity(i);
258                         }
259                         else{
260                             //Toast
261                             Toast.makeText(UserProfileActivity.this,
262                                 "Feedback not available", Toast.LENGTH_SHORT).show();
263                         }
264                     }
265                 }
266             });
267         }
268     });
269 }
270
271 @Override
272 protected void onCreate(Bundle savedInstanceState) {
273     super.onCreate(savedInstanceState);
274 }
275
276 @Override
277 public void onConfigurationChanged(Configuration newConfig) {
278     super.onConfigurationChanged(newConfig);
279 }
280
281 @Override
282 public boolean onPrepareOptionsMenu(Menu menu) {
283     return super.onPrepareOptionsMenu(menu);
284 }
285
286 @Override
287 public boolean onOptionsItemSelected(MenuItem item) {
288     return super.onOptionsItemSelected(item);
289 }
290 }
```

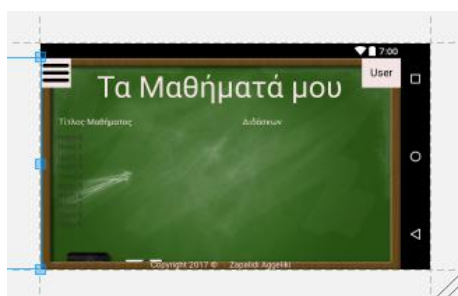


activity_courses.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.v4.widget.DrawerLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:id="@+id/activity_courses_drawer_layout"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent">
7
8     <!-- The main content view -->
9     <RelativeLayout
10        xmlns:android="http://schemas.android.com/apk/res/android"
11        android:layout_width="match_parent"
12        android:layout_height="match_parent"
13        android:background="@drawable/blackboard">
14
15        <ImageButton
16            android:id="@+id/activity_courses_menu_image_button"
17            android:layout_width="50dp"
18            android:layout_height="50dp"
19            android:layout_alignParentTop="true"
20            android:src="@drawable/menu"
21            android:background="@color/chalk"
22            android:scaleType="fitXY"/>
23        <!-- this will be our menu button-->
24        <Button
25            android:id="@+id/activity_courses_user_button"
26            android:layout_width="wrap_content"
27            android:layout_height="50dp"
28            android:layout_alignParentRight="true"
29            android:background="@color/chalk"
30            android:text="@string/user"/>
31
32        <TextView
33            android:layout_width="wrap_content"
34            android:layout_height="60dp"
35            android:text="@string/courses"
36            android:textSize="50sp"
37            android:layout_centerHorizontal="true"
38            android:layout_margin="20dp"
39            android:id="@+id/headerCourses"
40            android:textColor="@color/chalk"/>
41
42        <LinearLayout
43            android:orientation="horizontal"
44            android:layout_width="match_parent"
45            android:layout_height="wrap_content"
46            android:layout_marginLeft="30dp"
47            android:layout_marginStart="30dp"
48            android:layout_marginRight="30dp"
49            android:layout_marginEnd="30dp"
50            android:layout_marginBottom="10dp"
51            android:id="@+id/container"
52            android:layout_below="@+id/headerCourses">
53
```



```
54     <TextView
55         android:id="@+id/name"
56         android:layout_width="wrap_content"
57         android:layout_height="wrap_content"
58         android:text="@string/courseName"
59         android:layout_weight="1"
60         android:textColor="@color/chalk"/>
61
62     <TextView
63         android:layout_marginLeft="15dp"
64         android:layout_marginStart="15dp"
65         android:layout_width="wrap_content"
66         android:layout_height="wrap_content"
67         android:id="@+id/professor"
68         android:layout_weight="1"
69         android:textColor="@color/chalk"
70         android:text="@string/courseTeacher"/>
71 </LinearLayout>
72
73 <android.support.v7.widget.RecyclerView
74     android:layout_width="match_parent"
75     android:layout_height="200dp"
76     android:id="@+id/activity_courses_recycler_view"
77     android:layout_below="@id/container"
78     android:layout_marginLeft="30dp"
79     android:layout_marginStart="30dp"
80     android:layout_marginRight="30dp"
81     android:layout_marginEnd="30dp"
82     android:layout_marginBottom="30dp"/>
83
84 <LinearLayout
85     android:layout_width="wrap_content"
86     android:layout_height="wrap_content"
87     android:layout_alignParentBottom="true"
88     android:layout_centerHorizontal="true">
89     <TextView
90         android:layout_width="wrap_content"
91         android:layout_height="wrap_content"
92         android:text="@string/copyright_2017"
93         android:layout_marginRight="10dp"
94         android:textColor="@color/chalk"/>
95     <TextView
96         android:layout_width="wrap_content"
97         android:layout_height="wrap_content"
98         android:text="@string/footerMyName"
99         android:layout_marginLeft="10dp"
100        android:textColor="@color/chalk"/>
101 </LinearLayout>
102
103 </RelativeLayout>
104
105 <!-- The navigation drawer -->
106 <ScrollView... >
184 </android.support.v4.widget.DrawerLayout>
```



CoursesActivity.java

```
1 package com.unipi.angzap.schoolleadershipapp;
2 import [ . . . ]
28 public class CoursesActivity extends Activity {
29     private DrawerLayout mDrawerLayout;
30     private ImageButton mMenuButton;
31     private Button mUserButton;
32     private Button menuCoursesButton, menuBookshelfButton,
33     menuGradesButton, menuAnnouncementsButton, menuAbsenceButton,
34     menuCalendarButton, menuContactUsButton, menuFaqButton,
35     menuFeedbackButton;
36
37     @Override
38     protected void onCreate(Bundle savedInstanceState) {
39         super.onCreate(savedInstanceState);
40         setContentView(R.layout.activity_courses);
41
42         mMenuButton = (ImageButton)
43         findViewById(R.id.activity_courses_menu_image_button);
44         mMenuButton.setOnClickListener(new View.OnClickListener() {
45             public void onClick(View v) {
46                 mDrawerLayout.openDrawer(Gravity.LEFT);
47             }
48         });
49
50         mUserButton = (Button)
51         findViewById(R.id.activity_courses_user_button);
52         mUserButton.setOnClickListener(new View.OnClickListener() {
53             public void onClick(View v) {
54                 Intent i = new Intent(CoursesActivity.this,
55                 UserProfileActivity.class);
56                 startActivity(i);
57             }
58         });
59
60         //RecyclerView
61         mCoursesRecyclerView = (RecyclerView)
62         findViewById(R.id.activity_courses_recycler_view);
63
64         mAdapter = new ShowCoursesRecyclerViewAdapter(mCourseList);
65
66         RecyclerView.LayoutManager mLayoutManager = new
```



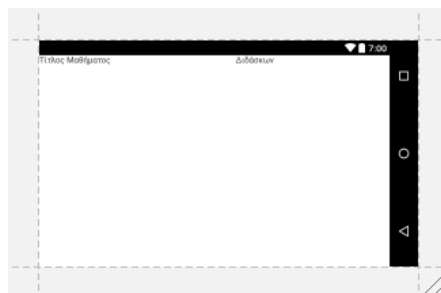
```
LinearLayoutManager(getApplicationContext());
64 //Our NavDrawer Menu buttons
        [. . .] }
65     });
66     }
67     });
68     }
69
70     @Override
71     protected void onCreate(Bundle savedInstanceState) {
72         super.onCreate(savedInstanceState);
73     }
74
75
76     private void prepareData() { //fetch data to fill the recycler
77         mCourseList.clear();
78         ParseQuery<ParseObject> query =
ParseQuery.getQuery("Coursebook");
79         query.findInBackground(new FindCallback<ParseObject>() {
80             @Override
81             public void done(List<ParseObject> objects, ParseException e)
82             {
83                 if (e == null) {
84                     Log.d("bla", objects.size() + "");
85
86                     for (int i = 0; i < objects.size(); i++) {
87                         mCourseList.add(new CourseModel(
88                             objects.get(i).getString("name"),
89                             objects.get(i).getString("teacher"),
90                             objects.get(i).getString("courseBookUrl"),
91                             objects.get(i).getString("courseInfo"),
92                             objects.get(i).getString("curriculumInfo"),
93                             objects.get(i).getString("extraCur"),
94                             objects.get(i).getString("extraCur2")
95                         ));
96                     }
97                     mAdapter.notifyDataSetChanged();
98                 } else {
99                     Log.d("Course", "Exception");
100                 }
101             }
102         });
103     }
235 }
```

course_recycler_row.xml

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:orientation="horizontal"
5     android:layout_width="match_parent"
```



```
6      android:layout_height="wrap_content">
7
8      <TextView
9          android:id="@+id/course_recycler_row_coursename_textview"
10         android:layout_width="wrap_content"
11         android:layout_height="wrap_content"
12         android:layout_weight="1"
13         android:textAlignment="center"
14         android:textColor="@color/chalk"
15         android:text="name" />
16
17     <TextView
18         android:id="@+id/course_recycler_row_courseteacher_textview"
19         android:layout_width="wrap_content"
20         android:layout_height="wrap_content"
21         android:layout_marginLeft="40dp"
22         android:layout_weight="1"
23         android:textAlignment="center"
24         android:textColor="@color/chalk"
25         android:text="teacher"/>
26
27 </LinearLayout>
```



CourseModel.java

```
1 package com.unipi.angzap.schoolleadershipapp;
2
3 public class CourseModel {
4     private String courseName, courseTeacher, courseBook, courseInfo,
5     curriculumInfo, extraCur, extraCur2;
6
7     public CourseModel(String courseName, String courseTeacher,
8     String courseBook, String courseInfo, String curriculumInfo, String
9     extraCur, String extraCur2) {
10         this.courseName = courseName;
11         this.courseTeacher = courseTeacher;
12         this.courseBook = courseBook;
13         this.courseInfo = courseInfo;
14         this.curriculumInfo = curriculumInfo;
15         this.extraCur = extraCur;
16         this.extraCur2 = extraCur2;
17     }
18
19     public String getCourseInfo() { return courseInfo; }
20     public void setCourseInfo(String courseInfo) { this.courseInfo =
21     courseInfo; }
22
23     public String getCurriculumInfo() { return curriculumInfo; }
24 }
```




```
28     public void setCurriculumInfo(String curriculumInfo) {
32         this.curriculumInfo = curriculumInfo; }
36     public String getExtraCur() { return extraCur; }
36     public void setExtraCur(String extraCur) { this.extraCur =
36         extraCur; }
40     public String getCourseName() { return courseName; }
44     public void setCourseName(String courseName) { this.courseName =
44         courseName; }
48     public String getCourseTeacher() { return courseTeacher; }
52     public void setCourseTeacher(String courseTeacher) {
52         this.courseTeacher = courseTeacher; }
56     public String getCourseBook() { return courseBook; }
60     public void setCourseBook(String courseBook) { this.courseBook =
60         courseBook; }
61     }
```

ShowCoursesRecyclerAdapter.java

```
1  package com.unipi.angzap.schoolleadershipapp;
2
3  import [ . . . ]
14
15  public class ShowCoursesRecyclerAdapter extends
15  RecyclerView.Adapter<ShowCoursesRecyclerAdapter.MyViewHolder>{
16
17      private ArrayList<CourseModel> courseList;
18
19      public class MyViewHolder extends RecyclerView.ViewHolder implements
19  View.OnClickListener{
20          private TextView name, teacher;
21          private LinearLayout container;
22          private Context ctx;
23
24          public MyViewHolder(View view) {
25              super(view);
26              name = (TextView)
26  view.findViewById(R.id.course_recycler_row_course_name_textview);
27              teacher = (TextView)
27  view.findViewById(R.id.course_recycler_row_course_teacher_textview);
28              name.setOnClickListener(this);
29              ctx = view.getContext();
30          }
31
32          @Override
33          public void onClick(View v) {
34              Log.d("clickdsfsd", name.getText().toString());
35
36              CourseModel course = courseList.get(getAdapterPosition());
37
38              Intent i = new Intent(v.getContext(), CourseStartActivity.class);
39              i.putExtra("courseBookUrl", course.getCourseBook());
40              i.putExtra("extraCur", course.getExtraCur());
41              i.putExtra("courseInfo", course.getCourseInfo());
42              i.putExtra("courseName", course.getCourseName());
43              i.putExtra("courseCurriculum", course.getCourseCurriculum());
44
45              Log.d("what error", course.getCourseBook());
46              v.getContext().startActivity(i);
```



```
47     }
48 }
49
50 public ShowCoursesRecyclerAdapter(ArrayList<CourseModel> cList) {
51     this.courseList = cList;
52 }
53
54 @Override
55 public MyViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
56     View itemView =
57     LayoutInflater.from(parent.getContext()).inflate(R.layout.course_recycler_row,
58     parent, false);
59
60     return new MyViewHolder(itemView);
61 }
62
63 @Override
64 public void onBindViewHolder(MyViewHolder holder, int position) {
65     CourseModel course = courseList.get(position);
66     holder.name.setText(course.getCourseName());
67     holder.teacher.setText(course.getCourseTeacher());
68 }
69
70 @Override
71 public int getItemCount() {
72     return courseList.size();
73 }
74 }
```

activity_course_start.xml

```
1 <?xml version="1.0" encoding="utf-8" ?>
2
3 <android.support.v4.widget.DrawerLayout
4     xmlns:android="http://schemas.android.com/apk/res/android"
5     android:id="@+id/activity_course_start_drawer_layout"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent">
8
9     <!-- The main content view -->
10    <RelativeLayout
11        xmlns:android="http://schemas.android.com/apk/res/android"
12        android:layout_width="match_parent"
13        android:layout_height="match_parent"
14        android:background="@drawable/blackboard">
15
16        <LinearLayout
17            android:layout_width="wrap_content"
18            android:layout_height="wrap_content">
19            <ImageButton
20                android:id="@+id/activity_course_start_menu_image_button"
21                android:layout_width="50dp"
22                android:layout_height="50dp"
23                android:layout_alignParentTop="true"
24                android:src="@drawable/menu"
```



```
25         android:background="@color/chalk"
26         android:scaleType="fitXY"/>
27
28     </LinearLayout>
29
30     <Button
31         android:id="@+id/activity_course_start_user_button"
32         android:layout_width="wrap_content"
33         android:layout_height="50dp"
34         android:layout_alignParentRight="true"
35         android:background="@color/chalk"
36         android:text="@string/user"/>
37
38     <TextView
39         android:layout_width="wrap_content"
40         android:layout_height="wrap_content"
41         android:text="CourseName"
42         android:id="@+id/courseheader"
43         android:textSize="50sp"
44         android:layout_margin="10dp"
45         android:layout_centerHorizontal="true"
46         android:textColor="@color/chalk"/>
47
48     <LinearLayout
49         android:layout_width="130dp"
50         android:layout_height="wrap_content"
51         android:orientation="vertical"
52         android:id="@+id/course_nav_LL"
53         android:layout_marginLeft="15dp"
54         android:layout_marginStart="15dp"
55         android:layout_below="@id/courseheader">
56         <Button
57             android:layout_width="match_parent"
58             android:layout_height="wrap_content"
59             android:background="@color/colorAccent"
60             android:id="@+id/activity_course_start_nav_LL_curriculum_button"
61             android:text="@string/curriculum_LL"/>
62         <Button
63             android:layout_width="match_parent"
64             android:layout_height="wrap_content"
65             android:background="@color/pastel1"
66             android:id="@+id/activity_course_start_nav_LL_quiz_button"
67             android:text="@string/quiz_LL"/>
68         <Button
69             android:layout_width="match_parent"
70             android:layout_height="wrap_content"
71             android:background="@color/pastel2"
72             android:id="@+id/activity_course_start_nav_LL_projects_button"
73             android:text="@string/projects_LL"/>
74         <Button
75             android:layout_width="match_parent"
76             android:layout_height="wrap_content"
```



```
77         android:background="@color/pastel13"
78 android:id="@+id/activity_course_start_nav_LL_extra_button"
79         android:text="@string/extra_LL"/>
80     <Button
81         android:layout_width="match_parent"
82         android:layout_height="wrap_content"
83         android:background="@color/chalk"
84 android:id="@+id/activity_course_start_nav_LL_coursebook_button"
85         android:text="@string/coursebook_LL"/>
86     </LinearLayout>
87
88     <ScrollView
89         android:id="@+id/activity_course_start_scrollview"
90         android:layout_width="match_parent"
91         android:layout_height="wrap_content"
92         android:layout_below="@+id/courseheader"
93         android:layout_toRightOf="@id/course_nav_LL"
94         android:layout_toEndOf="@id/course_nav_LL"
95         android:layout_margin="20dp">
96
97         <TextView
98             android:id="@+id/activity_course_start_info_tv"
99             android:layout_width="match_parent"
100            android:layout_height="wrap_content"
101            android:text="course info"
102            android:textColor="@color/chalk"/>
103        </ScrollView>
104
105    <LinearLayout
106        android:layout_width="match_parent"
107        android:layout_height="match_parent"
108        android:orientation="vertical"
109        android:visibility="gone"
110        android:id="@+id/activity_course_start_project_LL"
111        android:layout_below="@+id/courseheader"
112        android:layout_toRightOf="@id/course_nav_LL"
113        android:layout_toEndOf="@id/course_nav_LL"
114        android:layout_marginLeft="20dp"
115        android:layout_marginRight="20dp"
116        android:layout_marginBottom="20dp">
117
118        <android.support.v7.widget.RecyclerView
119            android:id="@+id/activity_course_projects_recycler"
120            android:layout_width="match_parent"
121            android:layout_height="200dp">
122
123        </android.support.v7.widget.RecyclerView>
124
125        <Button
126            android:id="@+id/activity_course_start_project_add_button"
127            android:layout_width="wrap_content"
128            android:layout_height="wrap_content"
```



```
129         android:text="Add Project"
130         android:background="@color/chalk"/>
131     </LinearLayout>
132
133
134     <LinearLayout
135         android:layout_width="wrap_content"
136         android:layout_height="wrap_content"
137         android:layout_alignParentBottom="true"
138         android:layout_centerHorizontal="true">
139         <TextView
140             android:layout_width="wrap_content"
141             android:layout_height="wrap_content"
142             android:text="@string/copyright_2017"
143             android:layout_marginRight="10dp"
144             android:layout_marginEnd="10dp"
145             android:textColor="@color/chalk" />
146         <TextView
147             android:layout_width="wrap_content"
148             android:layout_height="wrap_content"
149             android:text="@string/footerMyName"
150             android:layout_marginLeft="10dp"
151             android:layout_marginStart="10dp"
152             android:textColor="@color/chalk" />
153     </LinearLayout>
154
155 </RelativeLayout>
156
157     <!-- The navigation drawer -->
158 <ScrollView...>
159 </android.support.v4.widget.DrawerLayout>
```



CourseStartActivity.java

```
1 package com.unipi.angzap.schoolleadershipapp;
2
3 import [ . . . ]
33
34 public class CourseStartActivity extends Activity {
35
36     private DrawerLayout mDrawerLayout;
37     private ImageButton mMenuButton;
```



```
38     private Button mUserButton;
39     private Button navCurriculum, navQuiz, navExtra, navProjects,
navCoursebook;
40     private Button menuCoursesButton, menuBookshelfButton,
menuGradesButton, menuAnnouncementsButton, menuAbsenceButton,
menuCalendarButton, menuContactUsButton, menuFaqButton,
menuFeedbackButton;
41     private TextView courseInfoTV, courseNameTV;
42     private Button mAddProjectButton;
43     private LinearLayout projectLL;
44     private RecyclerView mProjectsRecyclerView;
45     private ScrollView mCenterScrollView;
46     // Extras expected
47     private String courseBook, extraCur, courseInfo, courseName,
courseCurriculum;
48     private ProjectRecyclerViewAdapter mAdapter;
49     private ArrayList<ProjectModel> projectList = new ArrayList<>();
50
51     @Override
52     protected void onCreate(Bundle savedInstanceState) {
53         super.onCreate(savedInstanceState);
54         setContentView(R.layout.activity_course_start);
55
56         courseBook =
getIntent().getExtras().getString("courseBookUrl");
57         extraCur = getIntent().getExtras().getString("extraCur");
58         courseInfo = getIntent().getExtras().getString("courseInfo");
59         courseName = getIntent().getExtras().getString("courseName");
60         courseCurriculum =
getIntent().getExtras().getString("courseCurriculum");
61
62         mCenterScrollView = (ScrollView)
findViewById(R.id.activity_course_start_scrollview);
63
64         mProjectsRecyclerView = (RecyclerView)
findViewById(R.id.activity_course_projects_recycler);
65
66         mAdapter = new ProjectRecyclerViewAdapter(projectList);
67
68         projectLL = (LinearLayout)
findViewById(R.id.activity_course_start_project_LL);
69
70         RecyclerView.LayoutManager mLayoutManager = new
LinearLayoutManager(getApplicationContext());
71         mProjectsRecyclerView.setLayoutManager(mLayoutManager);
72         mProjectsRecyclerView.setItemAnimator(new
DefaultItemAnimator());
73         mProjectsRecyclerView.setAdapter(mAdapter);
74
75         courseInfoTV = (TextView)
findViewById(R.id.activity_course_start_info_tv);
76         courseInfoTV.setText(courseInfo);
77
78         courseNameTV = (TextView) findViewById(R.id.courseheader);
79         courseNameTV.setText(courseName);
80
81         mAddProjectButton = (Button)
findViewById(R.id.activity_course_start_project_add_button);
```



```

82
83     mMenuButton = (ImageButton)
84     findViewById(R.id.activity_course_start_menu_image_button);
85     mMenuButton.setOnClickListener(new View.OnClickListener() {
86         public void onClick(View v) {
87             mDrawerLayout.openDrawer(Gravity.LEFT);
88         }
89     });
90     mUserButton = (Button)
91     findViewById(R.id.activity_course_start_user_button);
92     mUserButton.setOnClickListener(new View.OnClickListener() {
93         public void onClick(View v) {
94             Intent i = new Intent(CourseStartActivity.this,
95             UserProfileActivity.class);
96             startActivity(i);
97         }
98     });
99     mDrawerLayout = (DrawerLayout)
100    findViewById(R.id.activity_course_start_drawer_layout);
101    [...]
102
103    }
104    });
105    });
106
107    //NAV////////////////////////////////
108    navCoursebook = (Button)
109    findViewById(R.id.activity_course_start_nav_LL_coursebook_button);
110    navCoursebook.setOnClickListener(new View.OnClickListener() {
111        @Override
112        public void onClick(View v) {
113            Intent browserIntent = new Intent(Intent.ACTION_VIEW,
114            Uri.parse(courseBook));
115
116            browserIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
117            startActivity(browserIntent);
118        }
119    });
120    navCurriculum = (Button)
121    findViewById(R.id.activity_course_start_nav_LL_curriculum_button);
122    navCurriculum.setOnClickListener(new View.OnClickListener() {
123        @Override
124        public void onClick(View v) {
125            courseInfoTV.setText(courseCurriculum);
126            projectLL.setVisibility(View.GONE);
127            mCenterScrollView.setVisibility(View.VISIBLE);
128        }
129    });
130    navExtra = (Button)
131    findViewById(R.id.activity_course_start_nav_LL_extra_button);
132    navExtra.setOnClickListener(new View.OnClickListener() {
133        @Override
134        public void onClick(View v) {
135            Intent browserIntent = new Intent(Intent.ACTION_VIEW,

```




```
Uri.parse(extraCur));
236 browserIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
237         startActivity(browserIntent);
238     }
239 });
    navQuiz = (Button)
240 findViewById(R.id.activity_course_start_nav_LL_quiz_button);
241     navQuiz.setOnClickListener(new View.OnClickListener() {
242         @Override
243         public void onClick(View v) {
244             Intent i = new Intent(CourseStartActivity.this,
QuizActivity.class);
245             i.putExtra("courseName", courseName);
246             i.putExtra("courseBook", courseBook);
247             i.putExtra("extraCur", extraCur);
248             i.putExtra("courseInfo", courseInfo);
249             i.putExtra("courseCurriculum", courseCurriculum);
250             startActivity(i);
251         }
252     });
    navProjects = (Button)
253 findViewById(R.id.activity_course_start_nav_LL_projects_button);
254     navProjects.setOnClickListener(new View.OnClickListener() {
255         @Override
256         public void onClick(View v) {
257
258             projectLL.setVisibility(View.VISIBLE);
259             mCenterScrollView.setVisibility(View.GONE);
260             int role = ParseUser.getCurrentUser().getInt("role");
261             if (role == 1 || role == 0){
262 mAddProjectButton.setVisibility(View.VISIBLE);
                mAddProjectButton.setOnClickListener(new
263 View.OnClickListener() {
264                 @Override
265                 public void onClick(View v) {
266                     Intent i = new
Intent(CourseStartActivity.this, AddNewProjectActivity.class);
267                     startActivity(i);
268                 }
269             });
270         }
271     }
272 });
    getData();
274 }
275
276 private void getData() {
277     projectList.clear();
278
279     ParseQuery<ParseObject> query =
ParseQuery.getQuery("Project");
280     query.findInBackground(new FindCallback<ParseObject>() {
281         @Override
282         public void done(List<ParseObject> objects,
ParseException e) {
```



```
283         if (e != null) {
284             Log.d("Project", "Exception");
285         } else {
286             Log.d("bla", objects.size() + "");
287
288             for (int i = 0; i < objects.size(); i++) {
289                 projectList.add(new ProjectModel(
290                     objects.get(i).getString("courseName"),
291                     objects.get(i).getString("teacher"),
292                     objects.get(i).getString("description"),
293                     objects.get(i).getString("student0_0"),
294                     objects.get(i).getString("student0_1"),
295                     objects.get(i).getString("student1_0"),
296                     objects.get(i).getString("student1_1"),
297                     objects.get(i).getString("student2_0"),
298                     objects.get(i).getString("student2_1")
299                 ));
300             }
301             mAdapter.notifyDataSetChanged();
302         }
303     }
304     });
305 }
306 }
```

activity_quiz.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:background="@drawable/blackboard"
9     tools:context="com.unipi.angzap.schoolleadershipapp.QuizActivity">
10
11     <TextView
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content"
14         android:text="Απαντήστε στις παρακάτω Ερωτήσεις"
15         android:textColor="@color/chalk"
16         android:textSize="23sp"
17         android:layout_centerHorizontal="true"
18         android:layout_marginTop="30dp"/>
19     <!-- Quiz Cards -->
20     <LinearLayout
21         android:id="@+id/activity_quiz_first_question_layout"
22         android:layout_width="300dp"
23         android:layout_height="200dp"
24         android:layout_centerInParent="true"
25         android:gravity="center_horizontal"
26         android:orientation="vertical"
27         android:background="#ef9a9a"
28         android:visibility="gone">
```



```
29     <TextView
30         android:layout_width="wrap_content"
31         android:layout_height="wrap_content"
32         android:text="1st Question"
33         android:textSize="25sp"
34         android:textStyle="bold"/>
35     <ScrollView
36         android:layout_width="match_parent"
37         android:layout_height="wrap_content">
38         <LinearLayout
39             android:layout_width="match_parent"
40             android:layout_height="wrap_content"
41             android:id="@+id/question1contextLL"
42             android:orientation="vertical">
43             <TextView
44                 android:id="@+id/activity_quiz_first_question_text"
45                 android:layout_width="wrap_content"
46                 android:layout_height="wrap_content"
47                 android:text="Posa Bit exei ena Byte?"
48                 android:layout_marginTop="30dp"
49                 android:textSize="25sp"/>
50             <EditText
51                 android:id="@+id/activity_quiz_answer_1_et"
52                 android:layout_width="match_parent"
53                 android:layout_height="wrap_content" />
54             </LinearLayout>
55         </ScrollView>
56     </LinearLayout>
57
58     <LinearLayout
59         android:id="@+id/activity_quiz_second_question_layout"
60         android:layout_width="300dp"
61         android:layout_height="200dp"
62         android:layout_centerInParent="true"
63         android:gravity="center_horizontal"
64         android:orientation="vertical"
65         android:background="#a5d6a7"
66         android:visibility="gone">
67
68         <TextView
69             android:layout_width="wrap_content"
70             android:layout_height="wrap_content"
71             android:text="2nd Question"
72             android:textSize="25sp"
73             android:textStyle="bold"/>
74
75         <TextView
76             android:id="@+id/activity_quiz_second_question_text"
77             android:layout_width="wrap_content"
78             android:layout_height="wrap_content"
79             android:text="Poios einai o elaxistos ari8mos?"
80             android:layout_marginTop="20dp"
81             android:textSize="20sp"/>
82
83         <RadioGroup
84             android:layout_width="match_parent"
85             android:layout_height="wrap_content"
```



```
86         android:layout_marginLeft="10dp"
87         android:id="@+id/activity_quiz_answer_2_RG">
88
89         <RadioButton
90             android:layout_width="wrap_content"
91             android:layout_height="wrap_content"
92             android:id="@+id/activity_quiz_answer_2_choice_1_RB"
93             android:text="1"/>
94         <RadioButton
95             android:layout_width="wrap_content"
96             android:layout_height="wrap_content"
97             android:id="@+id/activity_quiz_answer_2_choice_2_RB"
98             android:text="2"/>
99         <RadioButton
100            android:layout_width="wrap_content"
101            android:layout_height="wrap_content"
102            android:id="@+id/activity_quiz_answer_2_choice_3_RB"
103            android:text="3"/>
104     </RadioGroup>
105 </LinearLayout>
106
107 <LinearLayout
108     android:id="@+id/activity_quiz_third_question_layout"
109     android:layout_width="300dp"
110     android:layout_height="200dp"
111     android:layout_centerInParent="true"
112     android:gravity="center_horizontal"
113     android:orientation="vertical"
114     android:background="#9fa8da"
115     android:visibility="visible">
116
117     <TextView
118         android:layout_width="wrap_content"
119         android:layout_height="wrap_content"
120         android:text="3rd Question"
121         android:textSize="25sp"
122         android:textStyle="bold"/>
123
124     <TextView
125         android:id="@+id/activity_quiz_third_question_text"
126         android:layout_width="wrap_content"
127         android:layout_height="wrap_content"
128         android:text="Poia einai h maza tou hliou?"
129         android:textSize="20sp"/>
130
131     <Spinner
132         android:layout_width="match_parent"
133         android:layout_height="wrap_content"
134         android:id="@+id/activity_quiz_answer_3_spinner"
135         android:entries="@array/quiz_answer_3_array"/>
136
137 </LinearLayout>
138 <!-- Quiz Cards END-->
139
140 <Button
141     android:layout_width="wrap_content"
142     android:layout_height="wrap_content"
143     android:id="@+id/activity_quiz_previous_button"
```



```
144     android:text="pre"
145     android:background="@color/chalk"
146     android:layout_alignBaseline="@+id/activity_quiz_next_button"
147     android:layout_alignBottom="@+id/activity_quiz_next_button"
148     android:layout_alignParentLeft="true"
149     android:layout_alignParentStart="true"
150     android:layout_marginLeft="5dp"
151     android:layout_marginStart="5dp"/>
152
153     <Button
154         android:layout_width="wrap_content"
155         android:layout_height="wrap_content"
156         android:id="@+id/activity_quiz_next_button"
157         android:text="next"
158         android:background="@color/chalk"
159         android:layout_centerVertical="true"
160         android:layout_alignParentRight="true"
161         android:layout_marginRight="5dp"
162         android:layout_marginEnd="5dp"
163     />
164 </RelativeLayout>
```

QuizActivity.java

```
1  package com.unipi.angzap.schoolleadershipapp;
2
3  Import[. . .]
27 import static android.R.attr.data;
28
29 public class QuizActivity extends Activity {
30     private Button nextButton, previousButton;
31     private LinearLayout firstQuestionLayout, secondQuestionLayout,
32     thirdQuestionLayout;
33     private int currentlyVisibleLayout;
34     private LinearLayout[] quizCardsCopy;
35     private ArrayList<String> answers = new ArrayList();
36     private EditText answer1;
37     private Spinner answer3Sp;
38     private RadioGroup answer2RG;
39     private RadioButton choice_1, choice_2, choice_3;
40     private TextView firstQuestionTV, secondQuestionTV, thirdQuestionTV;
41     private String courseName, courseBook, extraCur, courseInfo,
42     courseCurriculum, answer2;
43     private ProgressDialog mProgressDialog;
44
45     @Override
46     protected void onCreate(Bundle savedInstanceState) {
47         super.onCreate(savedInstanceState);
48         setContentView(R.layout.activity_quiz);
49
50         courseName = getIntent().getExtras().getString("courseName");
51         courseBook = getIntent().getExtras().getString("courseBook");
52         extraCur = getIntent().getExtras().getString("extraCur");
53         courseInfo = getIntent().getExtras().getString("courseInfo");
54         courseCurriculum =
55         getIntent().getExtras().getString("courseCurriculum");
56
57         firstQuestionTV = (TextView)
58         findViewById(R.id.activity_quiz_first_question_text);
```



```

        secondQuestionTV = (TextView)
55  findViewById(R.id.activity_quiz_second_question_text);
        thirdQuestionTV = (TextView)
56  findViewById(R.id.activity_quiz_third_question_text);
57
        nextButton = (Button)
58  findViewById(R.id.activity_quiz_next_button);
        previousButton = (Button)
59  findViewById(R.id.activity_quiz_previous_button);
60
        firstQuestionLayout = (LinearLayout)
61  findViewById(R.id.activity_quiz_first_question_layout);
        secondQuestionLayout = (LinearLayout)
62  findViewById(R.id.activity_quiz_second_question_layout);
        thirdQuestionLayout = (LinearLayout)
63  findViewById(R.id.activity_quiz_third_question_layout);
64
        LinearLayout[] quizCards = {firstQuestionLayout,
65  secondQuestionLayout, thirdQuestionLayout};
        quizCardsCopy = quizCards;
66
67
68  for (int i = 0; i < quizCardsCopy.length ; i++){
69      answers.add("Answer" + i); //test
70  }
71
72  currentlyVisibleLayout = 0;
73  showQuizCard(currentlyVisibleLayout);
74
75  nextButton.setOnClickListener(new View.OnClickListener() {
76      @Override
77      public void onClick(View v) {
78          currentlyVisibleLayout++;
79          showQuizCard(currentlyVisibleLayout);
80      }
81  });
82
83  previousButton.setOnClickListener(new View.OnClickListener() {
84      @Override
85      public void onClick(View v) {
86          currentlyVisibleLayout--;
87          showQuizCard(currentlyVisibleLayout);
88      }
89  });
90
        answer1 = (EditText)
91  findViewById(R.id.activity_quiz_answer_1_et);
        answer2RG = (RadioGroup)
92  findViewById(R.id.activity_quiz_answer_2_RG);
        choice_1 = (RadioButton)
93  findViewById(R.id.activity_quiz_answer_2_choice_1_RB);
        choice_2 = (RadioButton)
94  findViewById(R.id.activity_quiz_answer_2_choice_2_RB);
        choice_3 = (RadioButton)
95  findViewById(R.id.activity_quiz_answer_2_choice_3_RB);
        answer3Sp = (Spinner)
96  findViewById(R.id.activity_quiz_answer_3_spinner);
97  getData();
98  }
99
100 private void getData(){
101     mProgressDialog = new ProgressDialog(QuizActivity.this);
102     mProgressDialog.setMessage("Ετοιμασία ερωτήσεων...");

```



```
103         mProgressDialog.setCancelable(false);
104         mProgressDialog.show();
105
106         ParseQuery<ParseObject> query =
107         ParseQuery.getQuery("QuizQuestion");
108         query.whereEqualTo("courseName", courseName);
109         query.getFirstInBackground(new GetCallback<ParseObject>() {
110             public void done(ParseObject object, ParseException e) {
111                 mProgressDialog.dismiss();
112                 if (object == null) {
113                     Log.d("error", "No quizzes :(");
114                 } else {
115                     firstQuestionTV.setText(object.getString("question0"));
116                     secondQuestionTV.setText(object.getString("question1"));
117                     thirdQuestionTV.setText(object.getString("question2"));
118                     choice_1.setText(object.getString("option1_0"));
119                     choice_2.setText(object.getString("option1_1"));
120                     choice_3.setText(object.getString("option1_2"));
121                     String data[] = {object.getString("option2_0"),
122                                     object.getString("option2_1"), object.getString("option2_2")}; //ftiaxnw
123                                     ArrayAdapter<String> spinnerArrayAdapter = new
124                                     ArrayAdapter<>(
125                                         QuizActivity.this,
126                                         android.R.layout.simple_spinner_item, data);
127                                     answer3Sp.setAdapter(spinnerArrayAdapter);
128                                     }
129                                     });
130                                     }
131                                     private void showQuizCard(int index){
132                                     if(index>=quizCardsCopy.length){ // Patise End
133                                     currentlyVisibleLayout = quizCardsCopy.length-1;
134                                     //Submit results to parse and launch CoursesActivity
135                                     ParseObject quizResult = new ParseObject("QuizResult");
136                                     quizResult.put("userId", ParseUser.getCurrentUser());
137                                     quizResult.put("courseName",
138                                     getIntent().getExtras().getString("courseName"));
139                                     answers.set(0, answer1.getText() + "");
140
141                                     String answer2 =
142                                     ((RadioButton) findViewById(answer2RG.getCheckedRadioButtonId())).getText
143                                     ().toString(); //no nullable choice
144                                     answers.set(1, answer2);
145
146                                     answers.set(2, answer3Sp.getSelectedItem().toString());
147                                     for(int i = 0; i < answers.size(); i++){
148                                     quizResult.put("answer" + i, answers.get(i));
149                                     }
150                                     quizResult.saveEventually();
151
152                                     startActivity(new Intent(QuizActivity.this,
153                                     CoursesActivity.class));
154                                     }
155                                     else if (index<0)
156                                     currentlyVisibleLayout = 0;
157                                     else{
```




```
152         for(int i = 0; i<quizCardsCopy.length; i++){
153             quizCardsCopy[i].setVisibility(View.GONE);
154         }
155         quizCardsCopy[index].setVisibility(View.VISIBLE);
156     }
157     if(currentlyVisibleLayout == quizCardsCopy.length-1){
158         nextButton.setText("Submit");
159     }
160     else{
161         nextButton.setText("Next");
162     }
163 }
164 }
```

activity_bookshelf.xml

1	<?xml version="1.0" encoding="utf-8" ?>
2	
3	<android.support.v4.widget.DrawerLayout
4	xmlns:android="http://schemas.android.com/apk/res/android"
5	xmlns:tools="http://schemas.android.com/tools"
6	android:id="@+id/activity_bookshelf_drawer_layout"
7	android:layout_width="match_parent"
8	android:layout_height="match_parent">
9	<!-- The main content view -->
10	<RelativeLayout
11	xmlns:android="http://schemas.android.com/apk/res/android"
12	android:layout_width="match_parent"
13	android:layout_height="match_parent"
14	android:background="@drawable/blackboard">
15	
16	<ImageButton
17	android:id="@+id/activity_bookshelf_menu_image_button"
18	android:layout_width="50dp"
19	android:layout_height="50dp"
20	android:layout_alignParentTop="true"
21	android:src="@drawable/menu"
22	android:background="@color/chalk"
23	android:scaleType="fitXY"/>
24	
25	<Button
26	android:id="@+id/activity_bookshelf_menu_user_button"
27	android:layout_width="wrap_content"
28	android:layout_height="50dp"
29	android:layout_alignParentRight="true"
30	android:background="@color/chalk"
31	android:text="@string/user"/>
32	
33	<TextView
34	android:layout_width="wrap_content"
35	android:layout_height="60dp"
36	android:text="@string/bookshelf"
37	android:textSize="40sp"
38	android:layout_centerHorizontal="true"



```
39         android:layout_margin="20dp"
40         android:id="@+id/headerBookshelf"
41         android:textColor="@color/chalk"/>
42
43     <GridView
44         android:id="@+id/bookshelfGrid"
45         android:layout_below="@+id/headerBookshelf"
46         android:layout_marginLeft="30dp"
47         android:layout_marginStart="30dp"
48         android:layout_marginEnd="30dp"
49         android:layout_marginRight="30dp"
50         android:layout_marginBottom="30dp"
51         android:layout_width="match_parent"
52         android:layout_height="match_parent"
53         android:numColumns="3"
54         android:verticalSpacing="10dp"
55         android:horizontalSpacing="10dp"
56         android:gravity="center"/>
57
58     <LinearLayout
59         android:layout_width="wrap_content"
60         android:layout_height="wrap_content"
61         android:layout_alignParentBottom="true"
62         android:layout_centerHorizontal="true">
63         <TextView
64             android:layout_width="wrap_content"
65             android:layout_height="wrap_content"
66             android:textColor="@color/chalk"
67             android:text="@string/copyright_2017"
68             android:layout_marginRight="10dp"/>
69         <TextView
70             android:layout_width="wrap_content"
71             android:layout_height="wrap_content"
72             android:textColor="@color/chalk"
73             android:text="@string/footerMyName"
74             android:layout_marginLeft="10dp"/>
75     </LinearLayout>
76 </RelativeLayout>
77
78     <!-- The navigation drawer -->
79 <ScrollView ...>
80
81 </android.support.v4.widget.DrawerLayout>
```





BookshelfActivity.java

```
1 package com.unipi.angzap.schoolleadershipapp;
2
3 import [. . .]
27
28 public class BookshelfActivity extends Activity {
29
30     private DrawerLayout mDrawerLayout;
31     private ImageButton mMenuButton;
32     private Button mUserButton;
33     private Button menuCoursesButton, menuBookshelfButton,
34     menuGradesButton, menuAnnouncementsButton, menuAbsenceButton,
35     menuCalendarButton, menuContactUsButton, menuFaqButton,
36     menuFeedbackButton;
37
38     private GridView mGridView;
39     private BookshelfGridViewAdapter mAdapter;
40     private ArrayList<CoursebookModel> mArray = new ArrayList<>();
41
42     @Override
43     protected void onCreate(Bundle savedInstanceState) {
44         super.onCreate(savedInstanceState);
45         setContentView(R.layout.activity_bookshelf);
46
47         mMenuButton = (ImageButton)
48         findViewById(R.id.activity_bookshelf_menu_image_button);
49         mMenuButton.setOnClickListener(new View.OnClickListener() {
50             public void onClick(View v) {
51                 mDrawerLayout.openDrawer(Gravity.LEFT);
52             }
53         });
54
55         mUserButton = (Button)
56         findViewById(R.id.activity_bookshelf_menu_user_button);
57         mUserButton.setOnClickListener(new View.OnClickListener() {
58             @Override
59             public void onClick(View v) {
60                 Intent i = new Intent(BookshelfActivity.this,
61                 UserProfileActivity.class);
62                 startActivity(i);
63             }
64         });
65
66         mGridView = (GridView) findViewById(R.id.bookshelfGrid);
67         mAdapter = new
68         BookshelfGridViewAdapter(getApplicationContext(), mArray);
69         mGridView.setAdapter(mAdapter);
70         mDrawerLayout = (DrawerLayout)
71         findViewById(R.id.activity_bookshelf_drawer_layout);
72         menuCoursesButton = (Button)
73         findViewById(R.id.drawer_menu_courses_button);
74         menuCoursesButton.setOnClickListener(new View.OnClickListener()
75         {
76             @Override
77             public void onClick(View v) {
78                 Intent i = new Intent(BookshelfActivity.this,
79                 CoursesActivity.class);
```



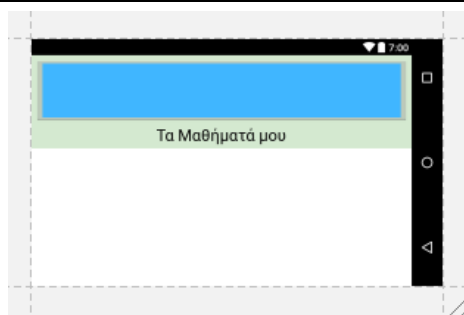
```
69         startActivity(i);
70     }
71     });
72
73     menuBookshelfButton = (Button)
74     findViewById(R.id.drawer_menu_bookshelf_button);
75     menuBookshelfButton.setOnClickListener(new
76     View.OnClickListener() {
77         @Override
78         public void onClick(View v) {
79
80     }
81     });
82     menuGradesButton = (Button)
83     findViewById(R.id.drawer_menu_grades_button);
84     menuGradesButton.setOnClickListener(new View.OnClickListener()
85     {
86         @Override
87         public void onClick(View v) {
88             int role = ParseUser.getCurrentUser().getInt("role");
89             if(role == 0 || role ==1){ //Admin // Teacher
90                 Intent i = new Intent(BookshelfActivity.this,
91                 GradesTeacherActivity.class);
92                 startActivity(i);
93             }
94             else{ //Student
95                 Intent i = new Intent(BookshelfActivity.this,
96                 GradesStudentActivity.class);
97                 startActivity(i);
98             }
99         }
100     });
101     menuAnnouncementsButton = (Button)
102     findViewById(R.id.drawer_menu_announcements_button);
103     menuAnnouncementsButton.setOnClickListener(new
104     View.OnClickListener() {
105         @Override
106         public void onClick(View v) {
107             int role = ParseUser.getCurrentUser().getInt("role");
108             if(role == 0 || role ==1){ //Admin // Teacher
109                 Intent i = new Intent(BookshelfActivity.this,
110                 AnnouncementsTeacherActivity.class);
111                 startActivity(i);
112             }
113             else{ //Student
114                 Intent i = new Intent(BookshelfActivity.this,
115                 AnnouncementsStudentActivity.class);
116                 startActivity(i);
117             }
118         }
119     });
120     menuAbsenceButton = (Button)
121     findViewById(R.id.drawer_menu_absence_button);
122     menuAbsenceButton.setOnClickListener(new View.OnClickListener()
123     {
124         @Override
125         public void onClick(View v) {
126             int role = ParseUser.getCurrentUser().getInt("role");
```



```
114         if(role == 0 || role ==1){ //Admin // Teacher
115             Intent i = new Intent(BookshelfActivity.this,
AbsenceTeacherActivity.class);
116             startActivity(i);
117         }
118         else{ //Student
119             Intent i = new Intent(BookshelfActivity.this,
AbsenceStudentActivity.class);
120             startActivity(i);
121         }
122     }
123 });
124     menuCalendarButton = (Button)
findViewById(R.id.drawer_menu_calendar_button);
125     menuCalendarButton.setOnClickListener(new
View.OnClickListener() {
126         @Override
127         public void onClick(View v) {
128             Intent i = new Intent(BookshelfActivity.this,
CalendarActivity.class);
129             startActivity(i);
130         }
131     });
132     menuContactUsButton = (Button)
findViewById(R.id.drawer_menu_contact_us_button);
133     menuContactUsButton.setOnClickListener(new
View.OnClickListener() {
134         @Override
135         public void onClick(View v) {
136             Intent i = new Intent(BookshelfActivity.this,
ContactUsActivity.class);
137             startActivity(i);
138         }
139     });
140     menuFaqButton = (Button)
findViewById(R.id.drawer_menu_faq_button);
141     menuFaqButton.setOnClickListener(new View.OnClickListener() {
142         @Override
143         public void onClick(View v) {
144             Intent i = new Intent(BookshelfActivity.this,
FaqActivity.class);
145             startActivity(i);
146         }
147     });
148     menuFeedbackButton = (Button)
findViewById(R.id.drawer_menu_feedback_button);
149     menuFeedbackButton.setOnClickListener(new
View.OnClickListener() {
150         @Override
151         public void onClick(View v) {
152
153             ParseQuery<ParseObject> query =
ParseQuery.getQuery("Configuration");
154             query.getFirstInBackground(new
GetCallback<ParseObject>() {
155                 public void done(ParseObject object, ParseException
e) {
156                     if (object == null) {
```



```
157         Log.d("Checkboxerror", "No data :(");
158     } else {
159         Log.d("Checkboxsuccess",
160 object.getBoolean("isFeedbackVisible") + "");
161
162         if(object.getBoolean("isFeedbackVisible")){
163             Intent i = new
164 Intent(BookshelfActivity.this, FeedbackActivity.class);
165             startActivity(i);
166         }
167         else{
168             //Toast
169             Toast.makeText(BookshelfActivity.this,
170 "Feedback not available", Toast.LENGTH_SHORT).show();
171         }
172     }
173 }
174 });
175 }
176
177 @Override
178 protected void onResume() {
179     super.onResume();
180     getData();
181 }
182
183 private void getData(){
184     mArray.clear();
185     ParseQuery<ParseObject> query =
186 ParseQuery.getQuery("Coursebook");
187     query.findInBackground(new FindCallback<ParseObject>() {
188         @Override
189         public void done(List<ParseObject> objects, ParseException e)
190 {
191             if (e == null){
192                 Log.d("bla", objects.size() + "");
193
194                 for (int i=0; i < objects.size(); i++) {
195                     mArray.add(new CoursebookModel(
196                         objects.get(i).getString("name"),
197                         objects.get(i).getString("courseBookUrl"),
198                         objects.get(i).getString("coverUrl"),
199                         objects.get(i).getString("courseBookClass")
200                     ));
201                 }
202                 mAdapter.notifyDataSetChanged();
203             }
204             else {
205                 Log.d("bla", "Exception");
206             }
207         }
208     });
209 }
```



BookshelfGridViewAdapter.java

```
1 package com.unipi.angzap.schoolleadershipapp;
2
3 import [ . . . ]
14 import com.bumptech.glide.Glide;
15 import java.util.ArrayList;
16
17 public class BookshelfGridViewAdapter extends
18 ArrayAdapter<CoursebookModel> {
19     Context mContext;
20     ArrayList<CoursebookModel> mArray = new ArrayList<>();
21
22     public BookshelfGridViewAdapter(Context context,
23     ArrayList<CoursebookModel> objects) {
24         super(context, R.layout.bookshelf_grid_row, objects);
25         this.mContext = context;
26         this.mArray = objects;
27     }
28
29     public class ViewHolder{
30         TextView mNameTextView;
31         ImageButton mCoverImageButton;
32     }
33
34     @Override
35     public View getView(final int position, View convertView, ViewGroup
36     parent) {
37         final ViewHolder holder;
38         CoursebookModel object = mArray.get(position);
39         if (convertView == null) {
40             LayoutInflater inflater = (LayoutInflater)
41             mContext.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
42             convertView = inflater.inflate(R.layout.bookshelf_grid_row,
43             parent, false);
44             holder = new ViewHolder();
45             holder.mNameTextView = (TextView)
46             convertView.findViewById(R.id.book_name);
47             holder.mCoverImageButton = (ImageButton)
48             convertView.findViewById(R.id.book_cover);
49             convertView.setTag(holder);
50         }
51         else{
52             holder = (ViewHolder) convertView.getTag();
53         }
54         holder.mNameTextView.setText(object.getName());
55         Glide.with(holder.mCoverImageButton.getContext())
56             .load(mArray.get(position).getCoverUrl())
```




```
51         .centerCrop()
52         .into(holder.mCoverImageButton);
53
54     holder.mCoverImageButton.setOnClickListener(new
55     View.OnClickListener() {
56         @Override
57         public void onClick(View v) {
58             Intent browserIntent = new Intent(Intent.ACTION_VIEW,
59             Uri.parse(mArray.get(position).getBookUrl()));
60             browserIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
61             mContext.startActivity(browserIntent);
62         }
63     });
64
65     return convertView;
66 }
```

activity_grades_student.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <android.support.v4.widget.DrawerLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      android:id="@+id/activity_grades_student_drawer_layout"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent">
7
8      <!-- The main content view -->
9      <RelativeLayout
10         xmlns:android="http://schemas.android.com/apk/res/android"
11         android:layout_width="match_parent"
12         android:layout_height="match_parent"
13         android:background="@drawable/blackboard">
14
15         <ImageButton
16             android:id="@+id/activity_grades_student_menu_image_button"
17             android:layout_width="50dp"
18             android:layout_height="50dp"
19             android:layout_alignParentTop="true"
20             android:src="@drawable/menu"
21             android:background="@color/chalk"
22             android:scaleType="fitXY"/>
23
24         <Button
25             android:id="@+id/activity_grades_student_user_button"
26             android:layout_width="wrap_content"
27             android:layout_height="50dp"
28             android:layout_alignParentRight="true"
29             android:background="@color/chalk"
30             android:text="@string/user"/>
31
32         <TextView
33             android:layout_width="wrap_content"
34             android:layout_height="60dp"
35             android:text="Οι Βαθμοί μου"
36             android:textSize="40sp"
37             android:layout_centerHorizontal="true"
38             android:layout_margin="20dp"
39             android:id="@+id/headerGradesStudent"
```



```
40         android:textColor="@color/chalk"/>
41
42     <LinearLayout
43         android:layout_width="match_parent"
44         android:layout_height="wrap_content"
45         android:id="@+id/activity_grades_student_recycler_header_linearlayout"
46         android:layout_below="@+id/headerGradesStudent">
47
48         <TextView
49             android:layout_width="wrap_content"
50             android:layout_height="wrap_content"
51             android:text="Μάθημα"
52             android:layout_centerHorizontal="true"
53             android:layout_weight="3"
54             android:layout_margin="20dp"
55             android:textColor="@color/chalk"/>
56         <TextView
57             android:layout_width="wrap_content"
58             android:layout_height="wrap_content"
59             android:text="Εξάμηνο Α"
60             android:layout_centerHorizontal="true"
61             android:layout_margin="20dp"
62             android:textColor="@color/chalk"/>
63
64         <TextView
65             android:layout_width="wrap_content"
66             android:layout_height="wrap_content"
67             android:text="Εξάμηνο Β"
68             android:layout_centerHorizontal="true"
69             android:layout_margin="20dp"
70             android:textColor="@color/chalk"/>
71
72         <TextView
73             android:layout_width="wrap_content"
74             android:layout_height="wrap_content"
75             android:text="Εξάμηνο Γ"
76             android:layout_centerHorizontal="true"
77             android:layout_margin="20dp"
78             android:textColor="@color/chalk"/>
79     </LinearLayout>
80
81     <android.support.v7.widget.RecyclerView
82         android:layout_width="match_parent"
83         android:layout_height="200dp"
84         android:id="@+id/activity_grades_student_recycler_view"
85         android:layout_below="@id/activity_grades_student_recycler_header_linear
86         layout"
87         android:layout_margin="20dp"/>
88
89     <LinearLayout
90         android:layout_width="wrap_content"
91         android:layout_height="wrap_content"
92         android:layout_alignParentBottom="true"
93         android:layout_centerHorizontal="true">
94         <TextView
95             android:layout_width="wrap_content"
96             android:layout_height="wrap_content"
97             android:textColor="@color/chalk"
```



```
97         android:text="@string/copyright_2017"  
98         android:layout_marginRight="10dp"/>  
99     <TextView  
100         android:layout_width="wrap_content"  
101         android:layout_height="wrap_content"  
102         android:textColor="@color/chalk"  
103         android:text="@string/footerMyName"  
104         android:layout_marginLeft="10dp"/>  
105     </LinearLayout>  
106 </RelativeLayout>  
107  
108 <!-- The navigation drawer -->  
109 <ScrollView ...>  
186 </ScrollView>  
187 </android.support.v4.widget.DrawerLayout>
```



GradesStudentActivity.java

```
1 package com.unipi.angzap.schoolleadershipapp;  
2 import [ . . . ]  
26 public class GradesStudentActivity extends Activity {  
27     private RecyclerView mGradesRecyclerView;  
28     private GradeStudentAdapter mAdapter;  
29     private ArrayList<GradeStudentModel> mGradeList = new  
30     ArrayList<>();  
31     private DrawerLayout mDrawerLayout;  
32     private ImageButton mMenuButton;  
33     private Button mUserButton;  
34     private Button menuCoursesButton, menuBookshelfButton,  
35     menuGradesButton, menuAnnouncementsButton, menuAbsenceButton,  
36     menuCalendarButton, menuContactUsButton, menuFaqButton,  
37     menuFeedbackButton;  
38  
39     @Override  
40     protected void onCreate(Bundle savedInstanceState) {  
41         super.onCreate(savedInstanceState);  
42         setContentView(R.layout.activity_grades_student);  
43  
44         mGradesRecyclerView = (RecyclerView)  
45         findViewById(R.id.activity_grades_student_recycler_view);  
46         mAdapter = new GradeStudentAdapter(mGradeList);  
47  
48         RecyclerView.LayoutManager layoutManager = new
```



```
LinearLayoutManager(getApplicationContext());
44     mGradesRecyclerView.setLayoutManager(mLayoutManager);
        mGradesRecyclerView.setItemAnimator(new
45     DefaultItemAnimator());
46
47     mGradesRecyclerView.setAdapter(mAdapter);
48
        mMenuButton = (ImageButton)
49     findViewById(R.id.activity_grades_student_menu_image_button);
        mMenuButton.setOnClickListener(new View.OnClickListener() {
50     public void onClick(View v) {
51         mDrawerLayout.openDrawer(Gravity.LEFT);
52     }
53     });
54
        mUserButton = (Button)
56     findViewById(R.id.activity_grades_student_user_button);
        mUserButton.setOnClickListener(new View.OnClickListener() {
57     public void onClick(View v) {
58         Intent i = new Intent(GradesStudentActivity.this,
59     UserProfileActivity.class);
60         startActivity(i);
61     }
62     });
63
        mDrawerLayout = (DrawerLayout)
64     findViewById(R.id.activity_grades_student_drawer_layout);
        mDrawerLayout.openDrawer(Gravity.LEFT);
65
        menuCoursesButton = (Button)
67     findViewById(R.id.drawer_menu_courses_button);
        menuCoursesButton.setOnClickListener(new
68     View.OnClickListener() {
69         @Override
70         public void onClick(View v) {
71             Intent i = new Intent(GradesStudentActivity.this,
72     CoursesActivity.class);
73             startActivity(i);
74         }
75     });
198     [...]
205     }
206 }
```

activity_grades_teacher.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.v4.widget.DrawerLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:id="@+id/activity_grades_teacher_drawer_layout"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent">
7
8     <!-- The main content view -->
9     <RelativeLayout
10        xmlns:android="http://schemas.android.com/apk/res/android"
```



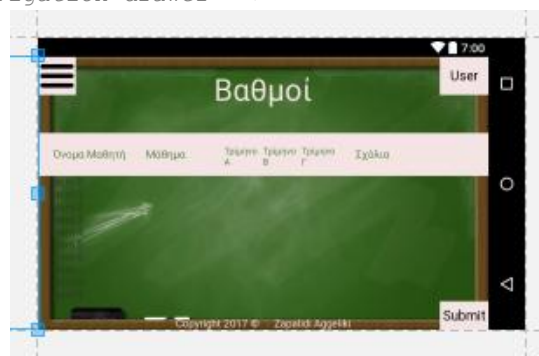
```
11     android:layout_width="match_parent"
12     android:layout_height="match_parent"
13     android:background="@drawable/blackboard">
14
15     <ImageButton
16         android:id="@+id/activity_grades_teacher_menu_image_button"
17         android:layout_width="50dp"
18         android:layout_height="50dp"
19         android:layout_alignParentTop="true"
20         android:src="@drawable/menu"
21         android:background="@color/chalk"
22         android:scaleType="fitXY"/>
23
24     <Button
25         android:id="@+id/activity_grades_teacher_user_button"
26         android:layout_width="wrap_content"
27         android:layout_height="50dp"
28         android:background="@color/chalk"
29         android:layout_alignParentRight="true"
30         android:text="@string/user"/>
31
32     <TextView
33         android:layout_width="wrap_content"
34         android:layout_height="60dp"
35         android:text="Βαθμοί"
36         android:textSize="40sp"
37         android:layout_centerHorizontal="true"
38         android:layout_margin="20dp"
39         android:id="@+id/headerGradesTeacher"
40         android:textColor="@color/chalk"/>
41
42     <LinearLayout
43         android:layout_width="match_parent"
44         android:layout_height="wrap_content"
45         android:background="@color/chalk"
46     android:id="@+id/activity_grades_teacher_recycler_header_linearlayout"
47     android:layout_below="@+id/headerGradesTeacher">
48
49         <TextView
50             android:layout_width="0dp"
51             android:layout_height="wrap_content"
52             android:text="Όνομα Μαθητή"
53             android:layout_centerHorizontal="true"
54     android:id="@+id/activity_grades_teacher_recycler_student_name_textview"
55     android:layout_marginTop="20dp"
56     android:layout_marginStart="20dp"
57     android:layout_marginLeft="20dp"
58     android:layout_weight="2"
59     android:textColor="@color/colorPrimary"/>
60
61         <TextView
62             android:layout_width="0dp"
63             android:layout_height="wrap_content"
64             android:text="Μάθημα"
```



```
64         android:layout_centerHorizontal="true"
65     android:id="@+id/activity_grades_teacher_recycler_course_name_textview"
66         android:layout_marginTop="20dp"
67         android:layout_marginStart="20dp"
68         android:layout_marginLeft="20dp"
69         android:layout_weight="2"
70         android:textColor="@color/colorPrimary"/>
71
72     <TextView
73     android:id="@+id/activity_grades_teacher_recycler_semA_textview"
74         android:layout_width="0dp"
75         android:layout_height="wrap_content"
76         android:layout_centerHorizontal="true"
77         android:layout_marginTop="15dp"
78         android:layout_weight="1"
79         android:textSize="12sp"
80         android:text="Τρίμηνο Α  "
81         android:textColor="@color/colorPrimary"/>
82
83     <TextView
84         android:layout_width="0dp"
85         android:layout_height="wrap_content"
86         android:layout_weight="1"
87         android:text="Τρίμηνο Β"
88         android:textSize="12sp"
89         android:layout_centerHorizontal="true"
90     android:id="@+id/activity_grades_teacher_recycler_semB_textview"
91         android:layout_marginTop="15dp"
92         android:textColor="@color/colorPrimary"/>
93
94     <TextView
95         android:layout_width="0dp"
96         android:layout_height="wrap_content"
97         android:layout_weight="1"
98         android:text="Τρίμηνο Γ"
99         android:textSize="12sp"
00         android:layout_centerHorizontal="true"
01     android:id="@+id/activity_grades_teacher_recycler_semC_textview"
02         android:layout_marginTop="15dp"
03         android:textColor="@color/colorPrimary"/>
04
05     <TextView
06         android:layout_width="0dp"
07         android:layout_height="wrap_content"
08         android:text="Εχόλια"
09         android:layout_weight="3"
10         android:layout_centerHorizontal="true"
11     android:id="@+id/activity_grades_teacher_recycler_comments_textview"
12         android:layout_margin="20dp"
13         android:textColor="@color/colorPrimary"/>
14 </LinearLayout>
```



```
15
16     <android.support.v7.widget.RecyclerView
17         android:layout_width="match_parent"
18         android:layout_height="wrap_content"
19         android:id="@+id/activity_grades_teacher_recycler_view"
20     android:layout_below="@id/activity_grades_teacher_recycler_header_linear
21     layout"
22         android:layout_marginLeft="20dp"
23         android:layout_marginStart="20dp"/>
24     <Button
25         android:id="@+id/activity_grades_teacher_submit_button"
26         android:layout_width="wrap_content"
27         android:layout_height="wrap_content"
28     android:layout_below="@+id/activity_grades_teacher_recycler_view"
29     android:layout_alignParentRight="true"
30     android:background="@color/chalk"
31     android:text="Submit"/>
32
33     <LinearLayout
34         android:layout_width="wrap_content"
35         android:layout_height="wrap_content"
36         android:layout_alignParentBottom="true"
37         android:layout_centerHorizontal="true">
38         <TextView
39             android:layout_width="wrap_content"
40             android:layout_height="wrap_content"
41             android:textColor="@color/chalk"
42             android:text="@string/copyright_2017"
43             android:layout_marginRight="10dp"/>
44         <TextView
45             android:layout_width="wrap_content"
46             android:layout_height="wrap_content"
47             android:textColor="@color/chalk"
48             android:text="@string/footerMyName"
49             android:layout_marginLeft="10dp"/>
50     </LinearLayout>
51 </RelativeLayout>
52 <!-- The navigation drawer -->
```



GradesTeacherActivity.java

```
1 package com.unipi.angzap.schoolleadershipapp;
```




```
2
3 import [ . . . ]
29
30 public class GradesTeacherActivity extends Activity {
31     private DrawerLayout mDrawerLayout;
32     private ImageButton mMenuButton;
33     private Button mUserButton, mSubmitButton;
34     private RecyclerView mGradeTeacherRecyclerView; //defining recycler
view class
35     private GradeTeacherAdapter mAdapter; //connecting to adapter class
36     private ArrayList<GradeTeacherModel> mGradeList = new
ArrayList<>(); //connects with model and creates arraylist
37     private List<ParseObject> fetchedGradesList;
38     private Button menuCoursesButton, menuBookshelfButton,
menuGradesButton, menuAnnouncementsButton, menuAbsenceButton,
menuCalendarButton, menuContactUsButton, menuFaqButton,
menuFeedbackButton;
39
40     @Override
41     protected void onCreate(Bundle savedInstanceState) { //executes
itself onCreate
42         super.onCreate(savedInstanceState);
43         setContentView(R.layout.activity_grades_teacher);
44         mGradeTeacherRecyclerView = (RecyclerView)
findViewById(R.id.activity_grades_teacher_recycler_view); //connects
with xml layout
45         mAdapter = new GradeTeacherAdapter(mGradeList);
46
47         mSubmitButton = (Button)
findViewById(R.id.activity_grades_teacher_submit_button);
48         mSubmitButton.setOnClickListener(new View.OnClickListener() {
49
50             @Override
51             public void onClick(View v) {
52
53                 for (int i = 0; i < fetchedGradesList.size(); i++) {
54                     fetchedGradesList.get(i).put("semGradeA",
mAdapter.getGradeTeacherList().get(i).getSemGradeA());
55                     fetchedGradesList.get(i).put("semGradeB",
mAdapter.getGradeTeacherList().get(i).getSemGradeB());
56                     fetchedGradesList.get(i).put("semGradeC",
mAdapter.getGradeTeacherList().get(i).getSemGradeC());
57                     fetchedGradesList.get(i).put("comments",
mAdapter.getGradeTeacherList().get(i).getComments());
58                     fetchedGradesList.get(i).saveEventually();
59                 }
60             }
61         });
62
63         RecyclerView.LayoutManager mLayoutManager = new
LinearLayoutManager(getApplicationContext());
64         mGradeTeacherRecyclerView.setLayoutManager(mLayoutManager);
65         mGradeTeacherRecyclerView.setItemAnimator(new
DefaultItemAnimator());
66         mGradeTeacherRecyclerView.setAdapter(mAdapter);
67
68         mMenuButton = (ImageButton)
findViewById(R.id.activity_grades_teacher_menu_image_button);
```



```
69         mMenuButton.setOnClickListener(new View.OnClickListener() {
70             public void onClick(View v) {
71                 mDrawerLayout.openDrawer(Gravity.LEFT);
72             }
73         });
74
75         mUserButton = (Button)
76         findViewById(R.id.activity_grades_teacher_user_button);
77         mUserButton.setOnClickListener(new View.OnClickListener() {
78             public void onClick(View v) {
79                 Intent i = new Intent(GradesTeacherActivity.this,
80                 UserProfileActivity.class);
81                 startActivity(i);
82             }
83         });
84
85         mDrawerLayout = (DrawerLayout)
86         findViewById(R.id.activity_grades_teacher_drawer_layout);
87         [. . .]
88
89         }
90     });
91
92     }
93
94     });
95     getData();
96 }
97
98 @Override
99 protected void onResume() {
100     super.onResume();
101 }
102
103 private void getData() {
104     mGradeList.clear();
105     ParseQuery<ParseObject> query = ParseQuery.getQuery("Grade");
106     query.findInBackground(new FindCallback<ParseObject>() {
107         @Override
108         public void done(List<ParseObject> objects, ParseException
109 e) {
110             if (e == null) {
111                 Log.d("bla", objects.size() + "");
112
113                 for (int i = 0; i < objects.size(); i++) {
114                     mGradeList.add(new GradeTeacherModel(
115                         objects.get(i).getString("username"),
116                         objects.get(i).getString("courseName"),
117                         objects.get(i).getString("comment"),
118                         objects.get(i).getInt("gradeSemA"),
119                         objects.get(i).getInt("gradeSemB"),
120                         objects.get(i).getInt("gradeSemC")
121                     ));
122                 }
123                 fetchedGradesList = objects;
124                 mAdapter.notifyDataSetChanged();
125             } else {
126                 Log.d("bla", "Exception");
127             }
128         }
129     });
130 }
```



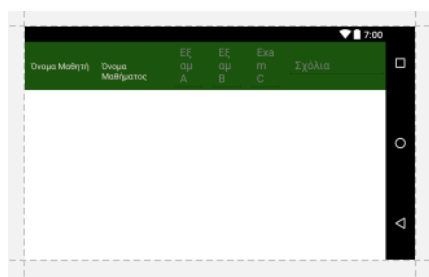
```
217     }
218   }
219   });
220 }
221
241 }
```

grade_teacher_recycler_row.xml

```
1  ?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      android:orientation="horizontal"
5      android:layout_width="match_parent"
6      android:layout_height="wrap_content"
7      android:background="@color/colorPrimaryDark">
8
9      <TextView
10         android:id="@+id/grade_teacher_recycler_row_student_name_textview"
11         android:layout_width="0dp"
12         android:layout_height="wrap_content"
13         android:layout_marginLeft="10dp"
14         android:textColor="@color/chalk"
15         android:layout_weight="2"
16         android:text="Όνομα Μαθητή" />
17
18         <TextView
19             android:id="@+id/grade_teacher_recycler_row_course_name_textview"
20             android:layout_width="0dp"
21             android:layout_height="wrap_content"
22             android:layout_marginLeft="10dp"
23             android:textColor="@color/chalk"
24             android:layout_weight="2"
25             android:text="Όνομα Μαθήματος" />
26
27         <EditText
28             android:id="@+id/grade_teacher_recycler_row_semA_textview"
29             android:layout_width="0dp"
30             android:layout_height="wrap_content"
31             android:layout_marginLeft="10dp"
32             android:layout_weight="1"
33             android:hint="Εξάμ Α"
34             android:textColor="@color/chalk" />
35
36         <EditText
37             android:id="@+id/grade_teacher_recycler_row_semB_textview"
38             android:layout_width="0dp"
39             android:layout_height="wrap_content"
40             android:layout_marginLeft="10dp"
41             android:textColor="@color/chalk"
42             android:layout_weight="1"
43             android:hint="Εξάμ Β"/>
44
45         <EditText
46             android:id="@+id/grade_teacher_recycler_row_semC_textview"
47             android:layout_width="0dp"
48             android:layout_height="wrap_content"
```



```
49         android:layout_marginLeft="10dp"
50         android:textColor="@color/chalk"
51         android:layout_weight="1"
52         android:hint="Exam C"/>
53
54     <EditText
55         android:id="@+id/grade_teacher_recycler_row_comments_textview"
56         android:layout_width="0dp"
57         android:layout_height="wrap_content"
58         android:layout_marginLeft="10dp"
59         android:textColor="@color/chalk"
60         android:hint="Σχόλια"
61         android:layout_weight="3"/>
62
63 </LinearLayout>
```



GradeTeacherAdapter.java

```
1 package com.unipi.angzap.schoolleadershipapp;
2
3 import [ . . . ]
4
5 public class GradeTeacherAdapter extends
6 RecyclerView.Adapter<GradeTeacherAdapter.MyViewHolder>{
7
8     private ArrayList<GradeTeacherModel> gradeTeacherList;
9
10    public class MyViewHolder extends RecyclerView.ViewHolder {
11        private TextView studentName, courseName;
12        //define class components PRIVATE
13        private LinearLayout container; //define container as is
14        private EditText semGradeA, semGradeB, semGradeC, comments;
15
16        private Context ctx;
17
18        public MyViewHolder(View view) {
19            super(view);
20            studentName = (TextView)
21            view.findViewById(R.id.grade_teacher_recycler_row_student_name_textview);
22            //fetches component ID from recycler row layout xml file
23            courseName = (TextView)
24            view.findViewById(R.id.grade_teacher_recycler_row_course_name_textview);
25            semGradeA = (EditText)
26            view.findViewById(R.id.grade_teacher_recycler_row_semA_textview);
27            semGradeB = (EditText)
28            view.findViewById(R.id.grade_teacher_recycler_row_semB_textview);
```



```
32         semGradeC = (EditText)
view.findViewById(R.id.grade_teacher_recycler_row_semC_textview);
33         comments = (EditText)
view.findViewById(R.id.grade_teacher_recycler_row_comments_textview);
34
35         ctx = view.getContext();
36     }
37 }
38
39 public GradeTeacherAdapter(ArrayList<GradeTeacherModel> gList) {
//constructor - den pairnei orisma
40     this.gradeTeacherList = gList;
41 }
42
43 public ArrayList<GradeTeacherModel> getGradeTeacherList() {
44     return gradeTeacherList;
45 }
46
47 public void setGradeTeacherList(ArrayList<GradeTeacherModel>
gradeTeacherList) {
48     this.gradeTeacherList = gradeTeacherList;
49 }
50
51 @Override
52 public GradeTeacherAdapter.MyViewHolder onCreateViewHolder
(ViewGroup parent, int viewType) { //calls adapter
    View itemView = LayoutInflater.from(parent.getContext()).inflate
(R.layout.grade_teacher_recycler_row, parent, false);
53 //calls xml recycler row
    return new GradeTeacherAdapter.MyViewHolder(itemView);
54 //after inflating the view, it returns the new viewholder
55 }
56
57 @Override
58 public void onBindViewHolder
(final GradeTeacherAdapter.MyViewHolder holder, int position) {
59     GradeTeacherModel grade = gradeTeacherList.get(position);
    holder.studentName.setText(grade.getStudentName());
60 //calls getters
    holder.courseName.setText(grade.getCourseName());
    holder.semGradeA.setText(grade.getSemGradeA() + "");
62 //because they're integers, to be printed as string we add " "
    holder.semGradeB.setText(grade.getSemGradeB() + "");
64     holder.semGradeC.setText(grade.getSemGradeC() + "");
65     holder.comments.setText(grade.getComments());
66
67     holder.semGradeA.addTextChangedListener(new TextWatcher() {
68         @Override
69         public void afterTextChanged(Editable s) {
70             if (!(s+ "").equals(""))
71                 gradeTeacherList.get(holder.getAdapterPosition()).
setSemGradeA(Integer.parseInt(s+ ""));
72         }
73     })
```



```
74         @Override
75         public void beforeTextChanged(CharSequence s, int start,
76                                     int count, int after) {
77         }
78
79         @Override
80         public void onTextChanged(CharSequence s, int start,
81                                 int before, int count) {
82         }
83     });
84
85     holder.semGradeB.addTextChangedListener(new TextWatcher() {
86         @Override
87         public void afterTextChanged(Editable s) {
88             if (!(s+"").equals(""))
89                 gradeTeacherList.get(
holder.getAdapterPosition()).setSemGradeB(
Integer.parseInt(s+""));
90         }
91
92         @Override
93         public void beforeTextChanged(CharSequence s, int start,
94                                     int count, int after) {
95         }
96
97         @Override
98         public void onTextChanged(CharSequence s, int start,
99                                 int before, int count) {
100        }
101    });
102
103     holder.semGradeC.addTextChangedListener(new TextWatcher() {
104         @Override
105         public void afterTextChanged(Editable s) {
106             if (!(s+"").equals(""))
107                 gradeTeacherList.get(holder.
getAdapterPosition()).setSemGradeC(
Integer.parseInt(s+""));
108         }
109
110         @Override
111         public void beforeTextChanged(CharSequence s, int start,
112                                     int count, int after) {
113         }
114
115         @Override
116         public void onTextChanged(CharSequence s, int start,
117                                 int before, int count) {
118         }
119    });
120
121     holder.comments.addTextChangedListener(new TextWatcher() {
122         @Override
```



```
123         public void afterTextChanged(Editable s) {
124             if (!(s+ "").equals(""))
125 gradeTeacherList.get(holder.getAdapterPosition()).setComments(s+ "");
126         }
127
128         @Override
129         public void beforeTextChanged(CharSequence s, int start,
130             int count, int after) {
131         }
132
133         @Override
134         public void onTextChanged(CharSequence s, int start,
135             int before, int count) {
136         }
137     });
138 }
139
140 @Override
141 public int getItemCount() {
142     return gradeTeacherList.size();
143 }
144 }
```

activity_announcements_teacher.xml

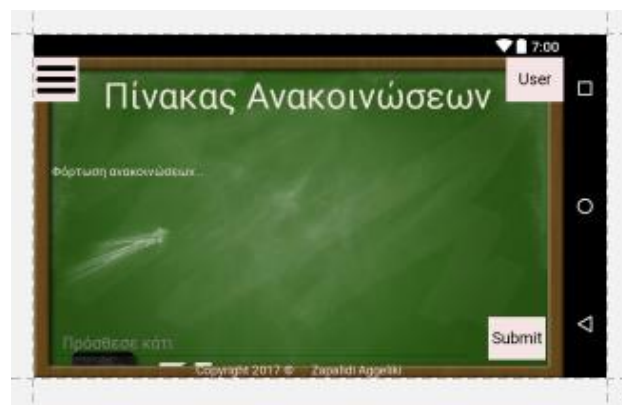
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.v4.widget.DrawerLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:id="@+id/activity_announcements_teacher_drawer_layout"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent">
7
8     <!-- The main content view -->
9     <RelativeLayout
10         xmlns:android="http://schemas.android.com/apk/res/android"
11         android:layout_width="match_parent"
12         android:layout_height="match_parent"
13         android:background="@drawable/blackboard">
14
15         <ImageButton
16             android:id="@+id/activity_announcements_teacher_menu_image_button"
17             android:layout_width="50dp"
18             android:layout_height="50dp"
19             android:layout_alignParentTop="true"
20             android:src="@drawable/menu"
21             android:background="@color/chalk"
22             android:scaleType="fitXY"/>
23
24         <Button
25             android:id="@+id/activity_announcements_teacher_user_button"
26             android:layout_width="wrap_content"
27             android:layout_height="50dp"
28             android:layout_alignParentRight="true"
29             android:background="@color/chalk">
```




```
30         android:text="@string/user"/>
31
32     <TextView
33         android:layout_width="wrap_content"
34         android:layout_height="60dp"
35         android:text="@string/announcements"
36         android:textSize="40sp"
37         android:layout_centerHorizontal="true"
38         android:layout_margin="20dp"
39         android:id="@+id/headerAnnouncements"
40         android:textColor="@color/chalk"/>
41
42     <ScrollView
43         android:layout_width="match_parent"
44         android:layout_height="match_parent"
45         android:layout_centerHorizontal="true"
46         android:layout_margin="20dp"
47         android:layout_below="@+id/headerAnnouncements"
48         android:layout_above="@+id/activity_announcements_addsomething_edittext"
49         android:id="@+id/scrollAnnounce">
50
51         <TextView
52             android:layout_height="wrap_content"
53             android:layout_width="match_parent"
54             android:id="@+id/activity_announcements_teacher_ann_TV"
55             android:textColor="@color/chalk"
56             android:text="Φόρτωση ανακοινώσεων..."/>
57         <!-- This will be a recyclerView-->
58
59     </ScrollView>
60
61     <!-- the following features should be available only for the
62     teacher-->
63
64     <EditText
65         android:layout_width="wrap_content"
66         android:layout_height="wrap_content"
67         android:hint="Πρόσθεσε κάτι"
68         android:layout_marginLeft="20dp"
69         android:layout_marginStart="20dp"
70         android:layout_marginBottom="20dp"
71         android:id="@+id/activity_announcements_addsomething_edittext"
72         android:layout_alignParentBottom="true"
73
74         android:layout_toLeftOf="@+id/activity_announcements_teacher_submit_button"
75         android:layout_alignParentLeft="true"
76         android:textColor="@color/chalk"/>
77
78     <Button
79         android:layout_width="wrap_content"
80         android:layout_height="wrap_content"
81         android:id="@+id/activity_announcements_teacher_submit_button"
82         android:background="@color/chalk"
83         android:layout_alignParentBottom="true"
84         android:layout_alignParentEnd="true"
85         android:layout_alignParentRight="true"
```



```
84         android:layout_margin="20dp"
85         android:text="Submit"/>
86
87
88     <LinearLayout
89         android:layout_width="wrap_content"
90         android:layout_height="wrap_content"
91         android:layout_alignParentBottom="true"
92         android:layout_centerHorizontal="true">
93         <TextView
94             android:layout_width="wrap_content"
95             android:layout_height="wrap_content"
96             android:textColor="@color/chalk"
97             android:text="@string/copyright_2017"
98             android:layout_marginRight="10dp"/>
99         <TextView
100            android:layout_width="wrap_content"
101            android:layout_height="wrap_content"
102            android:textColor="@color/chalk"
103            android:text="@string/footerMyName"
104            android:layout_marginLeft="10dp"/>
105     </LinearLayout>
106 </RelativeLayout>
107
108 <!-- The navigation drawer -->
109 <ScrollView . . .>
187 </android.support.v4.widget.DrawerLayout>
```



AnnouncementsTeacherActivity.java

```
1 package com.unipi.angzap.schoolleadershipapp;
2
3 import [ . . . ]
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27 public class AnnouncementsTeacherActivity extends Activity {
28
29     private DrawerLayout mDrawerLayout;
30
31     private ImageButton mMenuButton;
32     private Button mUserButton;
33
34
35     private TextView announcementsTV;
36
37     private EditText announcementsET;
38 }
```



```
        private Button menuCoursesButton, menuBookshelfButton,
        menuGradesButton, menuAnnouncementsButton, menuAbsenceButton,
        menuCalendarButton, menuContactUsButton, menuFaqButton, submitButton,
39 menuFeedbackButton;
40     private ProgressDialog mPD;
41
42     @Override
43     protected void onCreate(Bundle savedInstanceState) {
44         super.onCreate(savedInstanceState);
45         setContentView(R.layout.activity_announcements_teacher);
46
47         announcementsTV = (TextView)
48         findViewById(R.id.activity_announcements_teacher_ann_TV);
49         announcementsET = (EditText)
50         findViewById(R.id.activity_announcements_addsomething_edittext);
51
52         ParseQuery<ParseObject> query = new ParseQuery<>("Announcement");
53         query.orderByDescending("createdAt");
54         query.getFirstInBackground(new GetCallback<ParseObject>() {
55             @Override
56             public void done(ParseObject object, ParseException e) {
57                 if (e == null) {
58                     announcementsTV.setText(object.getString("body"));
59                 }
60                 else {
61                     announcementsTV.setText("Αποτυχία φόρτωσης
62                     δεδομένων.");
63                 }
64             }
65         });
66
67         mMenuButton = (ImageButton)
68         findViewById(R.id.activity_announcements_teacher_menu_image_button);
69         mMenuButton.setOnClickListener(new View.OnClickListener() {
70             public void onClick(View v) {
71                 mDrawerLayout.openDrawer(Gravity.LEFT);
72             }
73         });
74
75         mUserButton = (Button)
76         findViewById(R.id.activity_announcements_teacher_user_button);
77         mUserButton.setOnClickListener(new View.OnClickListener() {
78             @Override
79             public void onClick(View v) {
80                 Intent i = new
81                 Intent(AnnouncementsTeacherActivity.this, UserProfileActivity.class);
82                 startActivity(i);
83             }
84         });
85
86         mPD = new ProgressDialog(this);
87         mPD.setMessage("Αποθήκευση...");
88         mPD.setCancelable(false);
89
90         submitButton = (Button)
91         findViewById(R.id.activity_announcements_teacher_submit_button);
92         submitButton.setOnClickListener(new View.OnClickListener() {
93             @Override
94             public void onClick(View v) {
95                 //submit

```



```
        if (!(announcementsET.getText().equals("") ||
announcementsET.getText().equals("Φόρτωση ανακοινώσεων...") ||
announcementsET.getText().equals("Αποτυχία φόρτωσης δεδομένων."))){
91         mPD.show();
92         ParseObject annTextObject = new
93 ParseObject("Announcement");
          annTextObject.put("body", announcementsTV.getText() +
94 "\n" + announcementsET.getText());
          annTextObject.saveInBackground(new SaveCallback() {
95             @Override
96             public void done(ParseException e) {
97                 mPD.dismiss();
98                 if (e == null){
99
100
101 Toast.makeText(AnnouncementsTeacherActivity.this, "Επιτυχής προσθήκη
ανακοίνωσης",
102                 Toast.LENGTH_SHORT).show();

          announcementsTV.setText(announcementsTV.getText() + "\n" +
103 announcementsET.getText());
          announcementsET.setText("");
104     }
105     else {
106
107

          Toast.makeText(AnnouncementsTeacherActivity.this, "Αποτυχία προσθήκης
108 ανακοίνωσης",
109                 Toast.LENGTH_SHORT).show();
110     }
111     });
112 });
113 }
114 }
115 });
116
118     mDrawerLayout = (DrawerLayout)
119 findViewById(R.id.activity_announcements_teacher_drawer_layout);
123 [...]
239     }
240     });
243 }
277 }
```

activity_absence_student.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2
3 <android.support.v4.widget.DrawerLayout
4     xmlns:android="http://schemas.android.com/apk/res/android"
5     android:id="@+id/activity_absence_student_drawer_layout"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:layout_centerHorizontal="true">
9
10     <!-- The main content view -->
11     <RelativeLayout
12     xmlns:android="http://schemas.android.com/apk/res/android"
13         android:layout_width="match_parent"
```



```
14         android:layout_height="match_parent"
15         android:background="@drawable/blackboard">
16
17     <ImageButton
18     android:id="@+id/activity_absence_student_menu_image_button"
19         android:layout_width="50dp"
20         android:layout_height="50dp"
21         android:layout_alignParentTop="true"
22         android:src="@drawable/menu"
23         android:background="@color/chalk"
24         android:scaleType="fitXY"/>
25
26     <Button
27     android:id="@+id/activity_absence_student_menu_user_button"
28         android:layout_width="wrap_content"
29         android:layout_height="50dp"
30         android:layout_alignParentRight="true"
31         android:background="@color/chalk"
32         android:text="@string/user"/>
33
34     <TextView
35         android:layout_width="wrap_content"
36         android:layout_height="60dp"
37         android:text="@string/absence"
38         android:textSize="40sp"
39         android:layout_centerHorizontal="true"
40         android:layout_margin="20dp"
41         android:id="@+id/headerAbsenceTeacher"
42         android:textColor="@color/chalk"/>
43
44     <LinearLayout
45         android:layout_width="match_parent"
46         android:layout_height="wrap_content"
47         android:layout_below="@+id/headerAbsenceTeacher"
48         android:orientation="vertical"
49         android:layout_marginLeft="300dp"
50         android:layout_marginStart="300dp"
51         android:layout_marginTop="80dp"
52         android:layout_centerHorizontal="true">
53
54         <LinearLayout
55             android:layout_width="match_parent"
56             android:layout_height="wrap_content">
57
58             <TextView
59                 android:layout_width="wrap_content"
60                 android:layout_height="wrap_content"
61                 android:text="Σύνολο \n"
62                 android:textColor="@color/chalk" />
63
64             <TextView
65                 android:layout_width="wrap_content"
66                 android:layout_height="wrap_content"
67                 android:text="0"
68                 android:layout_marginLeft="95dp"
69                 android:layout_marginStart="95dp"
70                 android:textColor="@color/chalk"
71
72     android:id="@+id/activity_absence_student_total_TV"/>
73
74
```



```
75         </LinearLayout>
76         <LinearLayout
77             android:layout_width="match_parent"
78             android:layout_height="wrap_content">
79
80             <TextView
81                 android:layout_width="wrap_content"
82                 android:layout_height="wrap_content"
83                 android:text="Δικαιολογημένες \n"
84                 android:textColor="@color/chalk" />
85             <TextView
86                 android:layout_width="wrap_content"
87                 android:layout_height="wrap_content"
88                 android:text="0"
89                 android:layout_marginLeft="30dp"
90                 android:layout_marginStart="30dp"
91                 android:textColor="@color/chalk"
92         android:id="@+id/activity_absence_student_just_TV"/>
93
94     </LinearLayout>
95
96     <LinearLayout
97         android:layout_width="match_parent"
98         android:layout_height="wrap_content">
99         <TextView
100             android:layout_width="wrap_content"
101             android:layout_height="wrap_content"
102             android:text="Αδικοιολόγητες "
103             android:textColor="@color/chalk"/>
104         <TextView
105             android:layout_width="wrap_content"
106             android:layout_height="wrap_content"
107             android:text="0"
108             android:layout_marginLeft="30dp"
109             android:layout_marginStart="30dp"
110     android:id="@+id/activity_absence_student_unjust_TV"
111         android:textColor="@color/chalk"/>
112
113     </LinearLayout>
114 </LinearLayout>
115 <LinearLayout
116     <LinearLayout
117         android:layout_width="wrap_content"
118         android:layout_height="wrap_content"
119         android:layout_alignParentBottom="true"
120         android:layout_centerHorizontal="true">
121         <TextView
122             android:layout_width="wrap_content"
123             android:layout_height="wrap_content"
124             android:textColor="@color/chalk"
125             android:text="@string/copyright_2017"
126             android:layout_marginRight="10dp"/>
127         <TextView
128             android:layout_width="wrap_content"
129             android:layout_height="wrap_content"
130             android:textColor="@color/chalk"
131             android:text="@string/footerMyName"
132             android:layout_marginLeft="10dp"/>
133
134
135
136
```



```
137         </LinearLayout>
138     </RelativeLayout>
139
140     <!-- The navigation drawer -->
141     <ScrollView
142 </android.support.v4.widget.DrawerLayout>
```



AbasenceStudentActivity.java

```
1  package com.unipi.angzap.schoolleadershipapp;
2
3  import [ . . . ]
28 public class AbsenceStudentActivity extends Activity {
29
30     private DrawerLayout mDrawerLayout;
33     private ImageButton mMenuButton;
34     private Button mUserButton;
35     private TextView absenceJust, absenceUnjust, absenceTotal;
36     private Button menuCoursesButton, menuBookshelfButton,
37     menuGradesButton, menuAnnouncementsButton, menuAbsenceButton,
38     menuCalendarButton, menuContactUsButton, menuFaqButton,
39     menuFeedbackButton;
40
41     @Override
42     protected void onCreate(Bundle savedInstanceState) {
43         super.onCreate(savedInstanceState);
44         setContentView(R.layout.activity_absence_student);
45
46         //
47         mMenuButton = (ImageButton)
48         findViewById(R.id.activity_absence_student_menu_image_button);
49         mMenuButton.setOnClickListener(new View.OnClickListener() {
50             public void onClick(View v) {
51                 mDrawerLayout.openDrawer(Gravity.LEFT);
52             }
53         });
54
55         mUserButton = (Button)
56         findViewById(R.id.activity_absence_student_menu_user_button);
57         mUserButton.setOnClickListener(new View.OnClickListener() {
58             @Override
59             public void onClick(View v) {
60                 Intent i = new
61                 Intent(AbsenceStudentActivity.this, UserProfileActivity.class);
62                 startActivity(i);
63             }
64         });
65
66         mDrawerLayout = (DrawerLayout)
67         findViewById(R.id.activity_absence_student_drawer_layout);
68
69         absenceJust = (TextView)
70         findViewById(R.id.activity_absence_student_just_TV);
```




```
        absenceUnjust =
64 (TextView) findViewById(R.id.activity_absence_student_unjust_TV);
        absenceTotal =
65 (TextView) findViewById(R.id.activity_absence_student_total_TV);
66
67     ParseQuery<ParseObject> query = ParseQuery.getQuery("Absence");
        query.whereEqualTo("username",
68 ParseUser.getCurrentUser().getUsername());
69     query.getFirstInBackground(new GetCallback<ParseObject>() {
70         public void done(ParseObject object, ParseException e) {
71             if (object == null) {
72                 Log.d("My absence", "No data :(");
73             } else {
74                 Log.d("My absence", "Fetching student's absence");
75                 int total = object.getInt("total");
76                 int justified = object.getInt("justified");
77                 int unjust = total - justified;
78
79                 absenceJust.setText(String.valueOf(justified));
80                 absenceTotal.setText(String.valueOf(total));
81                 absenceUnjust.setText(String.valueOf(unjust));
82             }
83         }
84     });
85 }
86 }
87 });
88 [...]
193 }
194 });
197 }
198 });
201 }
235 }
```

activity_abasence_teacher.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2
3 <android.support.v4.widget.DrawerLayout
4     xmlns:android="http://schemas.android.com/apk/res/android"
5     android:id="@+id/activity_absence_teacher_drawer_layout"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:layout_centerHorizontal="true">
9
10 <!-- The main content view -->
11 <RelativeLayout
12     xmlns:android="http://schemas.android.com/apk/res/android"
13     android:layout_width="match_parent"
14     android:layout_height="match_parent"
15     android:background="@drawable/blackboard">
16
17     <ImageButton
18     android:id="@+id/activity_absence_teacher_menu_image_button"
19     android:layout_width="50dp"
20     android:layout_height="50dp"
21     android:layout_alignParentTop="true"
22     android:src="@drawable/menu"
23     android:background="@color/chalk"
24     android:scaleType="fitXY"/>
25
```



```
26     <Button
27         android:id="@+id/activity_absence_teacher_user_button"
28         android:layout_width="wrap_content"
29         android:layout_height="50dp"
30         android:layout_alignParentRight="true"
31         android:background="@color/chalk"
32         android:text="@string/user"/>
33
34     <TextView
35         android:layout_width="wrap_content"
36         android:layout_height="60dp"
37         android:text="Απουσίες"
38         android:textSize="40sp"
39         android:layout_centerHorizontal="true"
40         android:layout_margin="20dp"
41         android:id="@+id/headerAbsenceTeacher"
42         android:textColor="@color/chalk"/>
43
44     <LinearLayout
45         android:id="@+id/absence_teacher_legend_LL"
46         android:layout_width="match_parent"
47         android:layout_height="wrap_content"
48         android:layout_marginLeft="10dp"
49         android:layout_marginStart="10dp"
50         android:layout_marginRight="10dp"
51         android:layout_marginEnd="10dp"
52         android:layout_below="@id/headerAbsenceTeacher">
53         <TextView
54             android:id="@+id/absence_list_student_name_tv"
55             android:layout_width="0dp"
56             android:layout_height="wrap_content"
57             android:text="Όνομα Μαθητή"
58             android:layout_weight="1"
59             android:background="@color/chalk"/>
60
61         <TextView
62             android:id="@+id/absence_row_total_tv"
63             android:layout_marginLeft="15dp"
64             android:layout_marginStart="15dp"
65             android:background="@color/chalk"
66             android:layout_width="0dp"
67             android:layout_weight="2"
68             android:layout_height="wrap_content"
69             android:text="Σύνολο Απουσιών"/>
71     </LinearLayout>
74     <android.support.v7.widget.RecyclerView
75         android:layout_width="match_parent"
76         android:layout_height="wrap_content"
77         android:id="@+id/activity_absence_recycler_view"
78         android:layout_marginTop="8dp"
79         android:layout_marginLeft="15dp"
80         android:layout_marginStart="15dp"
81         android:layout_marginRight="15dp"
82         android:layout_marginEnd="15dp"
83         android:layout_below="@+id/absence_teacher_legend_LL"
84         android:layout_centerHorizontal="true" />
85     <Button
86         android:id="@+id/activity_absence_teacher_submit_button"
87         android:layout_width="wrap_content"
```



```
88         android:layout_height="40dp"
89         android:text="Submit"
90         android:layout_alignParentRight="true"
91         android:textSize="12sp"
92         android:layout_alignParentEnd="true"
93         android:layout_marginTop="5dp"
94
95         android:layout_below="@+id/activity_absence_teacher_user_b
96         utton"
97         android:background="@color/pastel12"/>
98     </LinearLayout . . .>
99 </RelativeLayout>
100 <!-- The navigation drawer -->
101 <ScrollView . . .>
102 </android.support.v4.widget.DrawerLayout>
103     });
104 }
105 }
```



AbsenceTeacherActivity.java

```
1 package com.unipi.angzap.schoolleadershipapp;
2 import [. . .]
3 public class AbsenceTeacherActivity extends Activity {
4
5     private DrawerLayout mDrawerLayout;
6     private RecyclerView mAbsenceRecyclerView;
7     private AbsenceRecyclerAdapter mAbsenceTeacherAdapter;
8     private ArrayList<AbsenceModel> mAbsenceList = new ArrayList<>();
9     private List<ParseObject> fetchedAbsencesList;
10    private ProgressDialog mProgressDialog;
11    private ImageButton mMenuButton;
12    private Button mUserButton;
13    private Button mAbsenceSubmitButton;
14    private Button menuCoursesButton, menuBookshelfButton,
15    menuGradesButton, menuAnnouncementsButton, menuAbsenceButton,
16    menuCalendarButton, menuContactUsButton, menuFaqButton,
17    menuFeedbackButton;
18
19    @Override
20    protected void onCreate(Bundle savedInstanceState) {
21        super.onCreate(savedInstanceState);
22        setContentView(R.layout.activity_absence_teacher);
23        mAbsenceRecyclerView = (RecyclerView)
24        findViewById(R.id.activity_absence_recycler_view);
25        mAbsenceTeacherAdapter = new AbsenceRecyclerAdapter(mAbsenceList);
26        RecyclerView.LayoutManager layoutManager = new
27        LinearLayoutManager(getApplicationContext());
28        mAbsenceRecyclerView.setLayoutManager(layoutManager);
29    }
30 }
```



```
53         mAbsenceRecyclerView.setItemAnimator(new DefaultItemAnimator());
54         mAbsenceRecyclerView.setAdapter(mAbsenceTeacherAdapter);
55         mProgressDialog = new ProgressDialog(this);
56         mProgressDialog.setMessage("Αποθήκευση...");
57         mProgressDialog.setCancelable(false);
58         mMenuButton = (ImageButton)
59 findViewById(R.id.activity_absence_teacher_menu_image_button);
60         mMenuButton.setOnClickListener(new View.OnClickListener() {
61             public void onClick(View v) {
62                 mDrawerLayout.openDrawer(Gravity.LEFT);
63             }
64         });
65
66         mUserButton = (Button)
67 findViewById(R.id.activity_absence_teacher_user_button);
68         mUserButton.setOnClickListener(new View.OnClickListener() {
69             public void onClick(View v) {
70                 Intent i = new
71                 Intent(AbsenceTeacherActivity.this, UserProfileActivity.class);
72                 startActivity(i);
73             }
74         });
75         mDrawerLayout = (DrawerLayout)
76 findViewById(R.id.activity_absence_teacher_drawer_layout);
77         mAbsenceSubmitButton = (Button)
78 findViewById(R.id.activity_absence_teacher_submit_button);
79         mAbsenceSubmitButton.setOnClickListener(new View.OnClickListener()
80 {
81     @Override
82     public void onClick(View v) {
83         for (int i=0;i<fetchedAbsencesList.size();i++){
84             fetchedAbsencesList.get(i).put("total",
85 mAbsenceTeacherAdapter.getAbsenceList().get(i).getAbsenceUnjust());
86             fetchedAbsencesList.get(i).put("justified",
87 mAbsenceTeacherAdapter.getAbsenceList().get(i).getAbsenceJust());
88             fetchedAbsencesList.get(i).saveEventually();
89         }
90     }
91 }); [ . . . ]
92     }
93     });
94 }
95     });
96     getData();
97 }
98     private void getData(){
99         mAbsenceList.clear();
100         ParseQuery<ParseObject> query =
101 ParseQuery.getQuery("Absence");
102         query.findInBackground(new FindCallback<ParseObject>() {
103             @Override
104             public void done(List<ParseObject> objects, ParseException e)
105 {
106             if (e == null){
107                 Log.d("bla", objects.size() + "");
108
109                 for (int i=0; i < objects.size(); i++) {
110                     mAbsenceList.add(new AbsenceModel(
111                         objects.get(i).getString("username"),
```



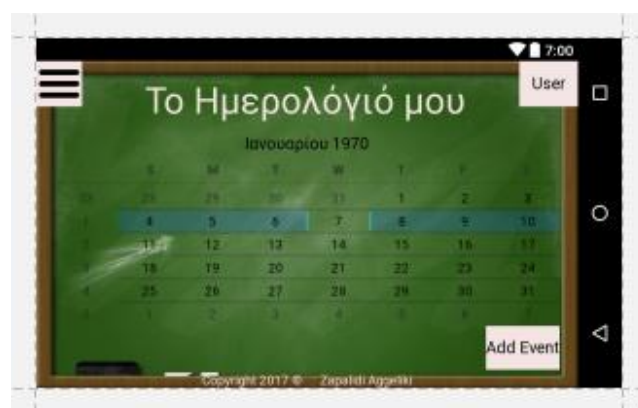
```
207         objects.get(i).getInt("total"),
208         objects.get(i).getInt("justified")
209     );
210     }
211     fetchedAbsencesList = objects;
212     mAbsenceTeacherAdapter.notifyDataSetChanged();
213 }
214 else {
215     Log.d("bla", "Exception");
216 }
217 }
218 });
219 }
241 }
```

activity_calendar.xml

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <android.support.v4.widget.DrawerLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:id="@+id/activity_calendar_drawer_layout"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent">
7
8     <!-- The main content view -->
9     <RelativeLayout
10        xmlns:android="http://schemas.android.com/apk/res/android"
11        android:layout_width="match_parent"
12        android:layout_height="match_parent"
13        android:background="@drawable/blackboard">
14
15        <ImageButton
16            android:id="@+id/activity_calendar_menu_image_button"
17            android:layout_width="50dp"
18            android:layout_height="50dp"
19            android:layout_alignParentTop="true"
20            android:src="@drawable/menu"
21            android:background="@color/chalk"
22            android:scaleType="fitXY"/>
23
24        <Button
25            android:id="@+id/activity_calendar_user_button"
26            android:layout_width="wrap_content"
27            android:layout_height="50dp"
28            android:layout_alignParentRight="true"
29            android:text="@string/user"
30            android:background="@color/chalk"/>
31
32        <TextView
33            android:layout_width="wrap_content"
34            android:layout_height="60dp"
35            android:text="@string/calendar"
36            android:textSize="40sp"
37            android:layout_centerHorizontal="true"
38            android:layout_margin="20dp"
39            android:id="@+id/headerCalendarTextView"
40            android:textColor="@color/chalk"/>
41
```



```
42
43     <CalendarView
44         android:id="@+id/activity_calendar_calendarview"
45         android:layout_width="match_parent"
46         android:layout_height="match_parent"
47         android:layout_marginTop="70dp"
48         android:layout_marginLeft="20dp"
49         android:layout_marginRight="20dp"
50         android:layout_marginBottom="70dp"/>
51
52     <Button
53         android:layout_width="wrap_content"
54         android:layout_height="wrap_content"
55         android:id="@+id/calendar_activity_add_event_button"
56         android:text="Add Event"
57         android:layout_margin="20dp"
58         android:layout_alignParentBottom="true"
59         android:layout_alignParentEnd="true"
60         android:layout_alignParentRight="true"
61         android:background="@color/chalk"/>
62
63     <LinearLayout
64         android:layout_width="wrap_content"
65         android:layout_height="wrap_content"
66         android:layout_alignParentBottom="true"
67         android:layout_centerHorizontal="true">
68         <TextView
69             android:layout_width="wrap_content"
70             android:layout_height="wrap_content"
71             android:textColor="@color/chalk"
72             android:text="@string/copyright_2017"
73             android:layout_marginRight="10dp"/>
74         <TextView
75             android:layout_width="wrap_content"
76             android:layout_height="wrap_content"
77             android:textColor="@color/chalk"
78             android:text="@string/footerMyName"
79             android:layout_marginLeft="10dp"/>
80     </LinearLayout>
81 </RelativeLayout>
82
83 <!-- The navigation drawer -->
84 <ScrollView . . .>
162 </android.support.v4.widget.DrawerLayout>
```





CalendarActivity.java

```
1  package com.unipi.angzap.schoolleadershipapp;
2
3  import [ . . . ]
25
26  public class CalendarActivity extends Activity {
27
28      private DrawerLayout mDrawerLayout;
29      private ImageButton mMenuButton;
30      private Button mUserButton, mAddEventButton;
31      private CalendarView mCalendar;
32      private Button menuCoursesButton, menuBookshelfButton, menuGradesButton,
33      menuAnnouncementsButton, menuAbsenceButton, menuCalendarButton, menuContactUsButton,
34      menuFaqButton, menuFeedbackButton;
35      private int mDay, mMonth, mYear;
36
37      @Override
38      protected void onCreate(Bundle savedInstanceState) {
39          super.onCreate(savedInstanceState);
40          setContentView(R.layout.activity_calendar);
41
42          mMenuButton = (ImageButton)
43          findViewById(R.id.activity_calendar_menu_image_button);
44          mMenuButton.setOnClickListener(new View.OnClickListener() {
45              public void onClick(View v) {
46                  mDrawerLayout.openDrawer(Gravity.LEFT);
47              }
48          });
49
50          mUserButton = (Button) findViewById(R.id.activity_calendar_user_button);
51          mUserButton.setOnClickListener(new View.OnClickListener() {
52              public void onClick(View v) {
53                  Intent i = new Intent(CalendarActivity.this,
54                  UserProfileActivity.class);
55                  startActivity(i);
56              }
57          });
58
59          mCalendar = (CalendarView)
60          findViewById(R.id.activity_calendar_calendarview); //NO MORE
61          mCalendar.setFirstDayOfWeek(2);
62          mCalendar.setOnDateChangeListener(new CalendarView.OnDateChangeListener() {
63              @Override
64              public void onSelectedDayChange(CalendarView view, int year, int month,
65              int dayOfMonth) {
66                  mYear = year;
67                  mMonth = month + 1;
68                  mDay = dayOfMonth;
69
70                  ParseQuery<ParseObject> query = ParseQuery.getQuery("Event");
71                  query.whereEqualTo("month", mMonth).whereEqualTo("day",
72                  mDay).whereEqualTo("year", mYear);
73                  query.getFirstInBackground(new GetCallback<ParseObject>() {
74                      public void done(ParseObject event, ParseException e) {
75                          if (e == null) {
76                              Log.d("event", "Retrieved " + event.get("type"));
77
78                              final AlertDialog alertDialog = new AlertDialog.Builder(
79                                  CalendarActivity.this).create();
80                              // Setting Dialog Title
```




```
78     alertDialog.setTitle(event.get("type") + " event By: " +
79     event.get("username"));
80     // Setting Dialog Message
81     alertDialog.setMessage(event.get("description") + "");
82     // Setting OK Button
83     alertDialog.setButton("OK", new
84     DialogInterface.OnClickListener() {
85         public void onClick(DialogInterface dialog, int
86         which) {
87             // Write your code here to execute after dialog
88             closed
89             alertDialog.dismiss();
90             }
91         });
92     // Showing Alert Message
93     alertDialog.show();
94     } else {
95     Log.d("event", "Error: " + e.getMessage());
96
97     final AlertDialog alertDialog = new AlertDialog.Builder(
98     CalendarActivity.this).create();
99     // Setting Dialog Title
100    alertDialog.setTitle("Error");
101    // Setting Dialog Message
102    alertDialog.setMessage(e.getMessage());
103    // Setting OK Button
104    alertDialog.setButton("OK", new
105    DialogInterface.OnClickListener() {
106        public void onClick(DialogInterface dialog, int
107        which) {
108            alertDialog.dismiss();
109            }
110        });
111    // Showing Alert Message
112    alertDialog.show();
113    }
114    }
115    });
116
117    int role = ParseUser.getCurrentUser().getInt("role");
118    mAddEventButton = (Button)
119    findViewById(R.id.calendar_activity_add_event_button);
120    if (role != 2) {
121    mAddEventButton.setOnClickListener(new View.OnClickListener() {
122        @Override
123        public void onClick(View v) {
124            startActivity(new Intent(CalendarActivity.this,
125            AddNewEventActivity.class));
126        }
127    });
128    } else {
129    mAddEventButton.setVisibility(View.GONE);
130    }
131
132    mDrawerLayout = (DrawerLayout)
133    findViewById(R.id.activity_calendar_drawer_layout);
134    [...]
135    }
136    });
137    });
138    }
139    }
140    }
```



activity_feedback.xml

```
1  <?xml version="1.0" encoding="utf-8" ?>
2  <RelativeLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      xmlns:tools="http://schemas.android.com/tools"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      android:background="@drawable/blackboard"
9      tools:context="com.unipi.angzap.schoolleadershipapp.QuizActivity">
10
11     <TextView
12         android:layout_width="wrap_content"
13         android:layout_height="wrap_content"
14         android:text="Παρακαλούμε, απαντήστε στις παρακάτω Ερωτήσεις"
15         android:textColor="@color/chalk"
16         android:textSize="23sp"
17         android:layout_centerHorizontal="true"
18         android:layout_marginTop="30dp"/>
19     <!-- Quiz Cards -->
20
21     <LinearLayout
22         android:id="@+id/activity_feedback_first_question_layout"
23         android:layout_width="300dp"
24         android:layout_height="200dp"
25         android:layout_centerInParent="true"
26         android:gravity="center_horizontal"
27         android:orientation="vertical"
28         android:background="@color/pastel2"
29         android:visibility="visible">
30
31         <TextView
32             android:layout_width="wrap_content"
33             android:layout_height="wrap_content"
34             android:text="1st Question"
35             android:textSize="25sp"
36             android:textStyle="bold"/>
37
38         <ScrollView
39             android:layout_width="match_parent"
40             android:layout_height="140dp">
41
42             <TextView
43                 android:id="@+id/activity_feedback_first_question_description_tv"
44                 android:layout_width="wrap_content"
45                 android:layout_height="wrap_content"
46                 android:text="Τι θα ήθελες να αλλάξει στο σχολείο?"
47                 android:layout_marginTop="10dp"
48                 android:textSize="19sp"/>
49
50             </ScrollView>
51
52         <RadioGroup
53             android:layout_width="match_parent"
54             android:layout_height="wrap_content"
55             android:orientation="horizontal"
56             android:layout_marginLeft="10dp"
57             android:id="@+id/activity_feedback_answer_1_RG">
58
59             <RadioButton
60                 android:layout_width="wrap_content"
61                 android:layout_height="wrap_content"
```



```
    android:id="@+id/activity_feedback_answer_1_choice_1_RB"
59         android:text="1"/>
60     <RadioButton
61         android:layout_width="wrap_content"
62         android:layout_height="wrap_content"

63     android:id="@+id/activity_feedback_answer_1_choice_2_RB"
64         android:text="2"/>
65     <RadioButton
66         android:layout_width="wrap_content"
67         android:layout_height="wrap_content"

68     android:id="@+id/activity_feedback_answer_1_choice_3_RB"
69         android:text="3"/>
70     <RadioButton
71         android:layout_width="wrap_content"
72         android:layout_height="wrap_content"

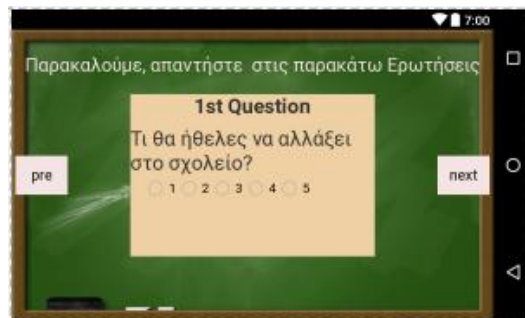
73     android:id="@+id/activity_feedback_answer_1_choice_4_RB"
74         android:text="4"/>
75     <RadioButton
76         android:layout_width="wrap_content"
77         android:layout_height="wrap_content"

78     android:id="@+id/activity_feedback_answer_1_choice_5_RB"
79         android:text="5"/>
80     </RadioGroup>
81 </LinearLayout>
82
83 <LinearLayout
84     android:id="@+id/activity_feedback_second_question_layout"
145     [...]
146
147 <LinearLayout
148     android:id="@+id/activity_feedback_third_question_layout"
207     [...]
208
209 <LinearLayout
210     android:id="@+id/activity_feedback_fourth_question_layout"
269     [...]
270
271 <LinearLayout
272     android:id="@+id/activity_feedback_fifth_question_layout"
273     android:layout_width="300dp"
274     android:layout_height="200dp"
275     android:layout_centerInParent="true"
276     android:gravity="center_horizontal"
277     android:orientation="vertical"
278     android:background="@color/pastel2"
279     android:visibility="gone">
280
281     <TextView
282         android:layout_width="wrap_content"
283         android:layout_height="wrap_content"
284         android:text="5th Question"
285         android:textSize="25sp"
286         android:textStyle="bold"/>
287
288     <TextView

289     android:id="@+id/activity_feedback_fifth_question_description_tv"
290     android:layout_width="wrap_content"
```



```
291         android:layout_height="wrap_content"
292         android:text="Πώς σου φαίνεται το σχολείο?"
293         android:textSize="20sp"/>
294
295
296     <EditText
297         android:id="@+id/activity_feedback_answer_5_et"
298         android:layout_width="match_parent"
299         android:layout_height="wrap_content" />
300 </LinearLayout>
301
302
303 <!-- Quiz Cards END-->
304
305 <Button
306     android:layout_width="wrap_content"
307     android:layout_height="wrap_content"
308     android:id="@+id/activity_feedback_previous_button"
309     android:text="pre"
310     android:background="@color/chalk"
311     android:layout_alignBaseline="@+id/activity_feedback_next_button"
312     android:layout_alignBottom="@+id/activity_feedback_next_button"
313     android:layout_alignParentLeft="true"
314     android:layout_alignParentStart="true"
315     android:layout_marginLeft="5dp"
316     android:layout_marginStart="5dp"/>
317 <Button
318     android:layout_width="wrap_content"
319     android:layout_height="wrap_content"
320     android:id="@+id/activity_feedback_next_button"
321     android:text="next"
322     android:background="@color/chalk"
323     android:layout_centerVertical="true"
324     android:layout_alignParentRight="true"
325     android:layout_marginRight="5dp"
326     android:layout_marginEnd="5dp"
327 />
328 </RelativeLayout>
```



FeedbackActivity.java

```
1 package com.unipi.angzap.schoolleadershipapp;
2
3 import [...]
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25 public class FeedbackActivity extends Activity {
26
27     private Button nextButton, previousButton;
```



```
29     private LinearLayout firstQuestionLayout, secondQuestionLayout,
        thirdQuestionLayout, fourthQuestionLayout, fifthQuestionLayout;
31     private int currentlyVisibleLayout;
33     private LinearLayout[] feedbackCardsCopy;
35     private ArrayList<String> answers = new ArrayList();
37     private TextView first_question_description_tv,
        second_question_description_tv, third_question_description_tv,
        fourth_question_description_tv, fifth_question_description_tv;
39     private EditText answer5;
41     private RadioGroup answer1RG, answer2RG, answer3RG, answer4RG;
43     private RadioButton choice_0_0, choice_0_1, choice_0_2, choice_0_3, choice_0_4;
44     [...]
48     private ProgressDialog mProgressDialog;
52     @Override
53     protected void onCreate(Bundle savedInstanceState) {
54         super.onCreate(savedInstanceState);
55         setContentView(R.layout.activity_feedback);
57         nextButton = (Button) findViewById(R.id.activity_feedback_next_button);
58         previousButton = (Button)
            findViewById(R.id.activity_feedback_previous_button);
60         firstQuestionLayout = (LinearLayout)
            findViewById(R.id.activity_feedback_first_question_layout);
61         [...]
66         LinearLayout[] feedbackCards = {firstQuestionLayout, secondQuestionLayout,
            thirdQuestionLayout, fourthQuestionLayout, fifthQuestionLayout};
67         feedbackCardsCopy = feedbackCards;
69         for (int i = 0; i < feedbackCardsCopy.length ; i++){
70             answers.add("Answer" + i);
71         }
73         currentlyVisibleLayout = 0;
74         showFeedbackCard(currentlyVisibleLayout);
78         nextButton.setOnClickListener(new View.OnClickListener() {
79             @Override
80             public void onClick(View v) {
81                 currentlyVisibleLayout++;
```



```
82         showFeedbackCard(currentlyVisibleLayout);
83     }
84 });
86     previousButton.setOnClickListener(new View.OnClickListener() {
87         @Override
88         public void onClick(View v) {
89             currentlyVisibleLayout--;
90             showFeedbackCard(currentlyVisibleLayout);
91         }
92     });
94     first_question_description_tv = (TextView)
95     findViewById(R.id.activity_feedback_first_question_description_tv);
96     [...]
101    answer1RG = (RadioGroup) findViewById(R.id.activity_feedback_answer_1_RG);
102    [...]
107    choice_0_0 = (RadioButton)
108    findViewById(R.id.activity_feedback_answer_1_choice_1_RB);
109    [...]
128    getData();
129 }
131 private void getData() {
132
133     mProgressDialog = new ProgressDialog(FeedbackActivity.this);
134     mProgressDialog.setMessage("Ετοιμασία ερωτήσεων...");
135     mProgressDialog.setCancelable(false);
136     mProgressDialog.show();
137
138     ParseQuery<ParseObject> query = ParseQuery.getQuery("Feedback");
139     query.whereEqualTo("role", ParseUser.getCurrentUser().getNumber("role"));
140     query.getFirstInBackground(new GetCallback<ParseObject>() {
141         public void done(ParseObject object, ParseException e) {
142             mProgressDialog.dismiss();
143             if (object == null) { Log.d("error", "No questions :("); }
144             else {
145                 first_question_description_tv.setText(object.getString("Question0"));
146                 [...]
147             }
148         }
149     });
150 }
```



```
155         }
156     });
157 }
159 private void showFeedbackCard(int index){
160     if(index>=feedbackCardsCopy.length){ // PAtise End
161         currentlyVisibleLayout = feedbackCardsCopy.length-1;
163         ParseObject feedbackResult = new ParseObject("FeedbackResult");
164         feedbackResult.put("role", ParseUser.getCurrentUser().getInt("role"));
167         String answer1 =
((RadioButton) findViewById(answer1RG.getCheckedRadioButtonId())).getText().toString();
168         answers.set(0, answer1);
176         answers.set(4, answer5.getText() + "");
178         for(int i = 0; i < answers.size(); i++){
179             feedbackResult.put("answer" + i, answers.get(i));
180         }
181         feedbackResult.saveEventually();
184         Toast.makeText(FeedbackActivity.this, "Σας ευχαριστούμε για τη συμμετοχή
σας!",
185             Toast.LENGTH_SHORT).show();
188         startActivity(new Intent(FeedbackActivity.this, CoursesActivity.class));
190     }
191     else if (index<0)
192         currentlyVisibleLayout = 0;
193     for(int i = 0; i<feedbackCardsCopy.length; i++){
194         feedbackCardsCopy[i].setVisibility(View.GONE); }
198         feedbackCardsCopy[index].setVisibility(View.VISIBLE);
199     }
201     if(currentlyVisibleLayout == feedbackCardsCopy.length-1){
202         nextButton.setText("End");
203     }
204     else{
205         nextButton.setText("Next");
206     }
207 }
```