



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Δημιουργία εφαρμογής για Smartphones με σκοπό την υποστήριξη της εκπαιδευτικής διαδικασίας στην Εκπαίδευση A mobile application to support teaching and learning process in Education
Όνοματεπώνυμο Φοιτητή	Νικόλαος Πλακιάς
Πατρώνυμο	Μαθαίος
Αριθμός Μητρώου	ΜΠΠΛ15060
Επιβλέπων	Ευθύμιος Αλέπης, Επίκουρος Καθηγητής

Ημερομηνία Παράδοσης

Νοέμβριος 2018

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Αλέπης Ευθύμιος Πατσάκης Κωνσταντίνος Τσιχριντζής Γεώργιος
Επίκουρος Καθηγητής Επίκουρος Καθηγητής Καθηγητής

Περιεχόμενα

1. Abstract.....	04
2. Περίληψη.....	05
3. Ευχαριστίες.....	06
4. Εισαγωγή.....	07
5. Ανασκόπηση Πεδίου.....	09
6. Εφαρμογες.....	09
7. Αρχιτεκτονική Συστήματος.....	11
8. Αρχιτεκτονική Android App.....	13
9. Παρουσίαση και Χρήση Εφαρμογής	15
9.1 Είσοδος στην Εφαρμογή.....	15
9.2 Σενάριο Λειτουργιών του χρήστη ως Μαθητής.....	17
9.3 Σενάριο Λειτουργιών του χρήστη ως Καθηγητής.....	20
9.3.1 Εξέταση ανα μαθητή.....	21
9.3.2 Εξέταση ανα τάξη.....	23
9.3.3 Διαχείριση Ερωτήσεων.....	25
9.3.4 Στατιστικά.....	27
9.3.5 Ειδοποιήσεις Απαντήσεων.....	29
10. Ανάλυση Περίπτωσης Χρήσης στο Παρασκήνιο.....	31
11. Συμπεράσματα και μελλοντικές επεκτάσεις.....	32
12. Βιβλιογραφία.....	33
13. Παράρτημα.....	34

Abstract

The aim of the "ExamPtu" application is to interact at the level of education between the teacher and students separately through chrono-scheduled questions and notifications. The role of the teacher in the application will allow him to schedule personalized custom questions on a specific date and time for each student individually. The purpose of this process will help the teacher to understand better in a cognitive level each student. At the same time the teacher by using this application in a real-time examination process, can be driven in further exaggeration of student's interest in comparison with the traditional examination methods. The role of the student in the application will be an easy and instant way for examination from the teacher without many direct interactions. The questions that are ready for examination will arrive to the user without login as push notifications. In this way, the time that required to be spend will be quite smaller and the application will be more user-friendly.

Περίληψη

Στόχος της παρούσας εφαρμογής “ExamPtu” είναι η αλληλεπίδραση σε επίπεδο εκπαίδευσης μεταξύ του καθηγητή με τον κάθε μαθητή ξεχωριστά μέσω χρόνο-προγραμματισμένων ερωτήσεων-ειδοποιήσεων. Ο ρόλος του καθηγητή στην εφαρμογή θα του επιτρέπει να προγραμματίζει εξατομικευμένες ερωτήσεις σε συγκεκριμένη ημερομηνία για τον κάθε μαθητή του ξεχωριστά. Ο σκοπός αυτής της διαδικασίας-διέπαφης θα βοηθήσει τον καθηγητή να κατανοήσει καλύτερα το γνωστικό επίπεδο του κάθε μαθητή του ξεχωριστά. Παράλληλα χρησιμοποιώντας μια εφαρμογή όπου η διαδικασία εξέτασης θα γίνεται σε πραγματικό χρόνο πιθανόν να κεντρίσει ακόμα περισσότερο το ενδιαφέρον του μαθητή του συγκριτικά με τις παραδοσιακές μεθόδους εξέτασης. Ο ρόλος του μαθητή στην εφαρμογή θα αποτελεί ένα εύκολο και άμεσο τρόπο προκειμένου να εξετάζεται από τον καθηγητή του χωρίς την άμεση επαφή τους. Οι ερωτήσεις όπου θα εξετάζεται θα φτάνουν στο κινητό του σε μορφή notification και δεν θα απαιτούν είσοδο στην εφαρμογή. Με αυτό τον τρόπο ο χρόνος που θα απαιτείται προκειμένου να δαπανήσει θα είναι σχετικά μικρότερος και συνολικότερα η εφαρμογή θα είναι πιο φιλική προς τον ίδιο.

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή του τμήματος Πληροφορικής του Πανεπιστημίου Πειραιώς κ. Ευθύμιο Αλέπη για τις εύστοχες επισημάνσεις και παρατηρήσεις του και για την πολύτιμη βοήθεια που μου παρείχε καθ' όλη την διάρκεια της εκπόνησης της Πτυχιακής. Επίσης ευχαριστώ πολύ και τα μέλη της επιτροπής για τις ουσιώδεις υποδείξεις τους.

Τέλος, θα ήθελα να ευχαριστήσω τους δικούς μου ανθρώπους για την συμπαράσταση και την υπομονή που έδειξαν κατά την εκπόνηση της παρούσας πτυχιακής εργασίας, αλλά και καθ' όλη τη διάρκεια των σπουδών μου στο Πρόγραμμα Μεταπτυχιακών σπουδών στην Πληροφορική του Πανεπιστημίου Πειραιώς.

Εισαγωγή

Ο κεντρικός άξονας της εφαρμογής βασίστηκε πάνω στο «Αρχαίο» μοντέλο ρόλων μεταξύ ενός δασκάλου με τον μαθητή του. Από την ίδρυση αυτού του μοντέλου διάδρασης το σημαντικότερο σημείο που χρήζει περισσότερης προσοχής είναι το κομμάτι της επικοινωνίας μεταξύ των συγκεκριμένων ρόλων. Είναι σημαντικό να κατανοήσουμε την αλληλεπίδραση που υφίσταται ο κάθε εμπλεκόμενος φορέας αυτής της διαδικασίας συναρτήσει του άλλου.

Ουσιαστικά αυτή η διεπαφή ήταν η αφορμή να εντρυφήσουμε και να αναλύσουμε το πώς αυτό το μοντέλο που ανά τα χρόνια εξελίσσεται και τροποποιείται μπορεί να γίνει περισσότερο αποδοτικό.

Κάπου εδώ θα ήταν φρόνιμο να αναφερθούμε στην εξέλιξη της τεχνολογίας πολύπλευρα και ποικιλοτρόπως, αυτό έχει ως αποτέλεσμα την αέναη και διαρκή ανατροφοδότηση των παγιωμένων διαδικασιών με διαρκώς καινούργιες αστάθμητες πλέον μεταβλητές που μέχρι τώρα θεωρούνταν αδιαμφισβήτητα σταθερές. Ένα παράδειγμα πλέον μεταβλητής και τέως σταθεράς είναι ο χρόνος προσπέλασης της μάθησης είτε από την σκοπιά του διδάσκοντα είτε του μαθητευόμενου. Ο τρόπος εκπαίδευσης, η άμεση εξέταση, η κατανόηση του γνωστικού επιπέδου του εκάστοτε μαθητή είναι από τα πλέον βασικά χαρακτηριστικά που οφείλει να μπορεί να διαχειριστεί οποιοσδήποτε διδάσκων προκειμένου να μπορέσει να συμβάλει αποτελεσματικά στην διαδικασία εκπαίδευσης και μόρφωσης που έχουμε αναλάβει να εντρυφήσουμε.

Ο συνδυασμός λοιπόν από όλα τα παραπάνω συμβαλλόμενα μέρη θα μπορούσε εκ των ουκ άνευ να προσδώσει μια καινούργια πολυδιάστατη σκοπιά πάνω σε αυτό το κλασικό μοντέλο ρόλων και διαδραστικότητας μεταξύ αυτών. Η ιδέα ουσιαστικά που θα πραγματευτούμε σε αυτή την εφαρμογή είναι η αλληλεπίδραση του ρόλου του καθηγητή με αυτόν του μαθητή σε πραγματικό χρόνο έξω από την αυστηρά ελεγχόμενη ώρα διδασκαλίας αλλά και κατά την διάρκεια όλης της υπόλοιπης μέρας. Επρόκειτο για μία διαδικασία όπου στην πραγματικότητα έχει σαν σκοπό να μειώσει ίσως και να εξαλείψει το χρονικό περιθώριο που δαπανάτε προκειμένου να επιτευχθεί μία έγκυρη και έγκαιρη αλληλεπίδραση εκπαίδευσης.

Σημαντικός παράγοντας λοιπόν για να αποτελέσει λειτουργική και αποδεκτή η παραπάνω διαδικασία και από τους δύο συμβαλλόμενους ρόλους είναι η γρήγορη και η φιλική προς αυτούς χρήση χωρίς να απαιτείται παραπάνω από το χρονικό διάστημα ασχολίας που χρειάζεται. Η σκέψη που επικράτησε κατά την σχεδίαση της εφαρμογής είχε σαν στόχο την μία και μοναδική εγκατάσταση - είσοδο του ρόλου του μαθητή μέσα

στην εφαρμογή, από εκεί και πέρα η μοναδική ενέργεια που θα πρέπει να ανταποκριθεί θα είναι απλά η απάντηση των χρόνο-προγραμματισμένων ερωτήσεων που θα λαμβάνει σε μορφή notification στο κινητό του. Το σκεπτικό όπου οδηγηθήκαμε σε αυτήν την απόφαση είχε να κάνει με την απλότητα χρήσης από έναν μαθητή που όμως ανταποκρινόμενος σε αυτή την ειδοποίηση θα έδινε την κατάλληλη πληροφόρηση για την εξαγωγή ωφέλιμων συμπερασμάτων και εν τέλει στην εγρήγορση αυτής της διεπαφής.

Επιπρόσθετα κάνοντας οποιαδήποτε στιγμή log in στην εφαρμογή θα μπορέσει να δει ένα activity με χρήσιμες στατιστικές πληροφορίες, όπως το σύνολο των σωστών και των λανθασμένων απαντήσεων αντίστοιχα και έτσι θα μπορούσε να πάρει μία εικόνα για το κατά πόσο το διάβασμά του είναι επαρκές.

Από την άλλη πλευρά ο ρόλος του Καθηγητή έχει περισσότερες λειτουργίες-αρμοδιότητες. Ο συγκεκριμένος ρόλος είναι πιο απαιτητικός και διέπεται από τρεις βασικούς άξονες. Οι αρμοδιότητες όπου αναλογούν στο χρήστη είναι η διαχείριση των ερωτήσεων, ο χρόνο-προγραμματισμός αυτών προς τον εκάστοτε μαθητή και εν τέλει ο έλεγχος της απάντησης για τον κάθε μαθητή ξεχωριστά. Ως προς το τελευταίο σκέλος ο καθηγητής θα έχει την δυνατότητα να συγκρίνει την απάντηση που έλαβε με την προτεινόμενη απάντηση που ο ίδιος είχε προτείνει κατά την δημιουργία αυτής και έτσι να αποφασίσει κατά πόσο η απάντηση που δέχτηκε είναι σωστή. Το αποτέλεσμα αυτής της ενέργειας θα μπορεί να είναι ορατό και στον μαθητή (θα μπορεί να δει τα δικά του στατιστικά) αλλά και στον ίδιο (να μπορεί να δει στατιστικά των ειδοποιήσεων που έχει στείλει ανά μαθητή).

Ανασκόπηση Πεδίου

Στο Play Store υπάρχουν οι κάτωθι παρόμοιες εφαρμογές:

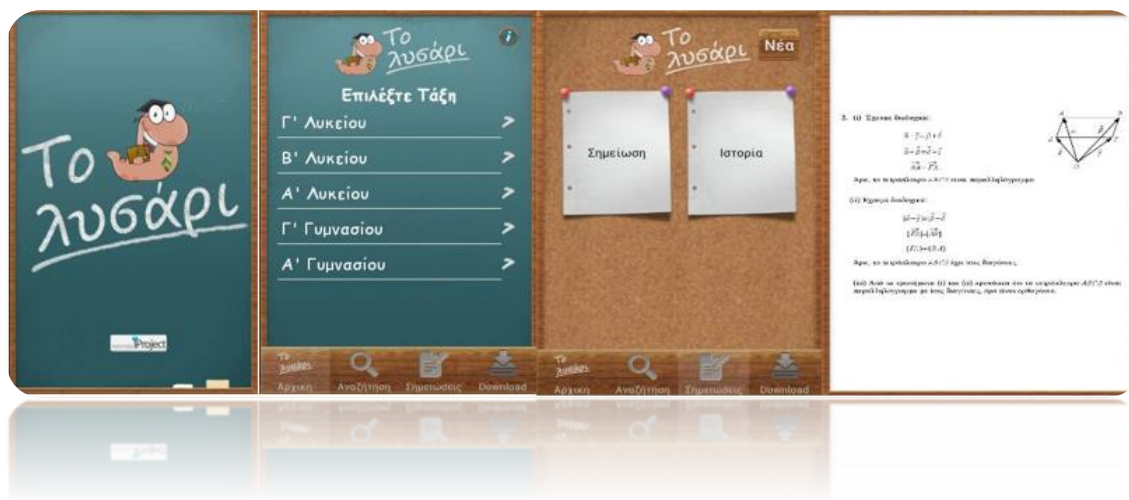
- ✓ Το Λυσάρι
- ✓ Μαθηματικά Γ Λυκείου - Πανελλήνιες
- ✓ LySTe - Τεστ Λυκείου

Οι εφαρμογές παρότι είναι πάρα πολύ καλά σχεδιασμένες και με ικανοποιητικό πλήθος ερωτήσεων, τείνουν να αντιμετωπίζουν το ίδιο πρόβλημα που θα συναντήσει και ένας καθηγητής μέσα στην τάξη του, δηλαδή την στασιμότητα που διέπει όλα αυτά τα χρόνια ο παραδοσιακός τρόπος εκπαίδευσης. Διότι παρότι οι εφαρμογές αυτές καθ' αυτές είναι user-friendly προς τους μαθητές δεν έχουν άμεση ανταπόκριση μέσα στην μέρα με τον ίδιο τον καθηγητή.

Εφαρμογές

1) Το Λυσάρι

Η συγκεκριμένη εφαρμογή έχει διάφορες μεθοδολογίες και λύσεις από όλες τις τάξεις της Δευτεροβάθμιας εκπαίδευσης (με εξαίρεση την Β γυμνασίου). Μπορείς να αναζητήσεις διάφορων ειδών ασκήσεις ανά τάξη, να σημειώσεις ενδεχομένως κάποια παρατήρηση και εφόσον βρεις αυτό που ψάχνεις να κατεβάσεις την λύση τοπικά στην συσκευή σου.



Εικόνα 1 : Στιγμιότυπα από την εφαρμογή «Το Λυσάρι»

Δημιουργία εφαρμογής για Smartphones με σκοπό την υποστήριξη της εκπαιδευτικής διαδικασίας στην Εκπαίδευση

2) Μαθηματικά Γ Λυκείου – Πανελλήνιες

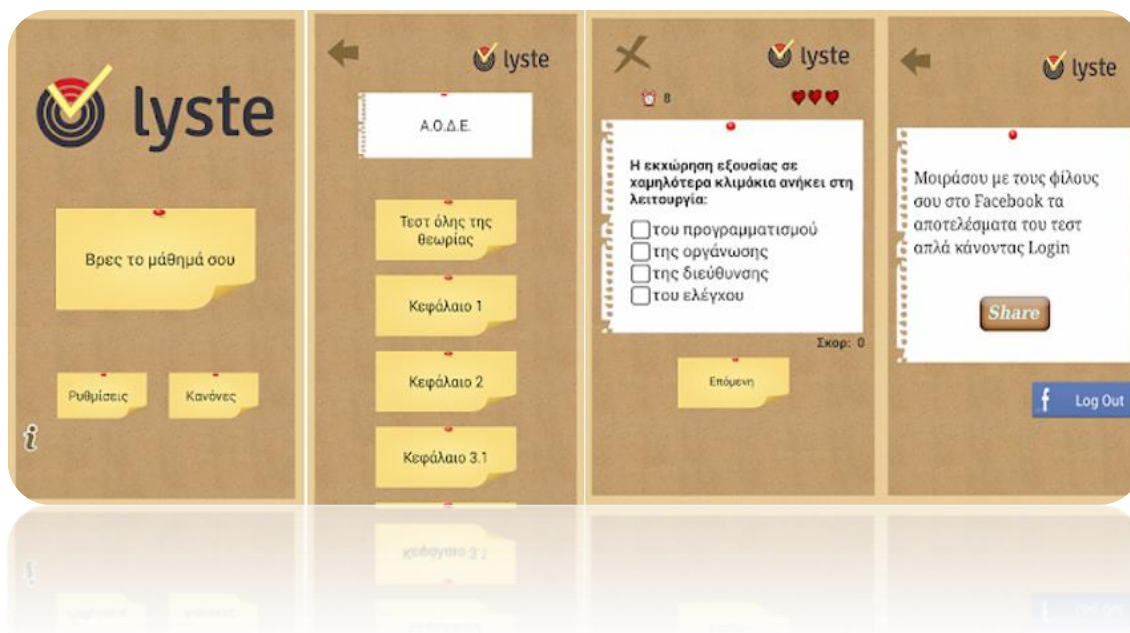
Σε αυτήν την εφαρμογή θα βρούμε ένα διαφορετικό σενάριο χρήσης από τον μαθητή. Ο σκοπός αυτής της εφαρμογής είναι ο μαθητής μέσω ενός πιο πρωτοποριακού μέσου (Smartphones - Tablets), να παρακινηθεί και να ενθουσιαστεί περισσότερο από την παραδοσιακή χρήση του σχολικού του βιβλίου. Οι απαντήσεις που θα δώσει θα είναι πολλαπλής επιλογής. Το μειονέκτημα αυτής της εφαρμογής είναι ότι ο μαθητής θα πρέπει από μόνος του να μπορέσει να βρει και να καταλάβει τα λάθη του, συν ότι ο καθηγητής δεν θα μπορεί να βρει τα πιθανά κενά που έχουν προκύψει κατά την εννοιολογική κατανόηση από τον μαθητή.



Εικόνα 2 : Στιγμιότυπα από την εφαρμογή «Μαθηματικά Γ Λυκείου - Πανελλήνιες»

3) LySTe – Τεστ Λυκείου

Τέλος σε αυτήν την εφαρμογή θα δούμε πως μαθητής θα πρέπει να επιλέξει σε ποιο μαθημα θα ήθελε να κάνει κάποια τέστ προκειμένου να εξασκηθεί ακόμη περισσότερο.



Εικόνα 3 : Στιγμιότυπα από την εφαρμογή «LySTe – Τεστ Λυκείου»

Αρχιτεκτονική Συστήματος

✚ Το Frame που υλοποιήθηκε η εφαρμογή είναι το Android Studio

Η ανάπτυξη της εφαρμογής που θα παρουσιάσουμε παρακάτω έγινε με τη χρήση του λειτουργικού συστήματος Android χρησιμοποιώντας την τελευταία έκδοση του ολοκληρωμένου προγραμματιστικού περιβάλλοντος (IDE) Android Studio. Το Android Studio είναι βασισμένο στο λογισμικό της JetBrains' IntelliJ IDEA, είναι σχεδιασμένο αποκλειστικά για προγραμματισμό Android (Ducrohet & Norbye & Chou, 2013) και αποτελεί το κύριο ολοκληρωμένο προγραμματιστικό περιβάλλον της Google για ανάπτυξη εφαρμογών Android. Είναι διαθέσιμο για Windows, Linux και Mac OS X και διατίθεται ελεύθερα προς χρήση. Η γλώσσα ανάπτυξης που χρησιμοποιήθηκε είναι Java, αντικειμενοστρεφής γλώσσα προγραμματισμού, που σχεδιάστηκε από την εταιρία πληροφορικής Sun Microsystems και πρωτοεμφανίστηκε πριν από 22 χρόνια, στις 23 Μαΐου του 1995 (Binstock, 2015).

Δημιουργία εφαρμογής για Smartphones με σκοπό την υποστήριξη της εκπαιδευτικής διαδικασίας στην Εκπαίδευση

🚦 Βασικές βιβλιοθήκες εφαρμογής

```
'com.android.support:appcompat-v7:26.1.0'
```

Η appcompat-v7 είναι η βιβλιοθήκη που μας προσφέρει τα απαραίτητα πακέτα για την υποστήριξη παλιότερων εκδόσεων Android APIs, καθώς επίσης και βασικές μεθόδους και τάξεις για την υλοποίηση της εφαρμογής.

```
'com.android.support.constraint:constraint-layout:1.0.2'
```

Η constraint-layout είναι η βιβλιοθήκη που μας επιτρέπει το σχεδιασμό δυναμικών UI (περιβάλλον χρήστη) στο Android.

🚦 Βάση Δεδομένων - Firebase Realtime

Για την υλοποίηση χρησιμοποιήθηκε η μη σχεσιακή βάση Firebase Realtime. Η βάση δεδομένων Firebase Realtime είναι μια βάση δεδομένων που φιλοξενείται σε cloud-hosted. Τα δεδομένα αποθηκεύονται ως JSON και συγχρονίζονται σε πραγματικό χρόνο σε κάθε συνδεδεμένο χρήστη. Όταν δημιουργούνται εφαρμογές πολλαπλών πλατφόρμων με τα SDK iOS, Android και JavaScript, όλοι οι χρήστες μοιράζονται ένα στιγμιότυπο βάσης δεδομένων πραγματικού χρόνου και λαμβάνουν αυτόματα ενημερώσεις με τα νεότερα δεδομένα.

Αρχιτεκτονική Android App

Οι παρακάτω κλάσεις αντιστοιχούν στους νοητούς πίνακες που δημιουργούμε στην βάση. Ουσιαστικά κάθε εγγραφή του πίνακα αντιστοιχεί σε ένα αποθηκευμένο αντικείμενο μίας εκ των παρακάτω κλάσεων.

Κατα την διάρκεια της υλοποίησης έγινε η προσπάθεια να οργανώσουμε μια μη σχεσιακή βάση με «σχεσιακό» μοτίβο. Ο σκοπός που έγινε αυτή η προσπάθεια έχει να κάνει με την πρόθεση να δημιουργήσουμε προγραμματιστικά καλούπια με σκοπό να δομήσουμε και να οργανώσουμε όσο καλύτερα γίνεται την πληροφορία που έχουμε να διαχειριστούμε.

User	Question	Notification
User_id : String Password : String Username : String Name : String Surname : String Email : String Phone : String Section : String Quality : String Age : String <hr/> getUser_id() setUser_id(String User_id) getPassword() setPassword(String Password) setUsername() setUsername(String Username) getName() setName(String Name) getSurname() setSurname(String Surname) getEmail () setEmail (String Email) getPhone () setPhone (String Phone) getSection() setSection(String Section) getQuality() setQuality(String Quality) getAge () setAge (String Age)	Question_id : String Question : String Answer : String <hr/> getQuestion_id() setQuestion_id(String Question_id) getQuestion() setQuestion(String Question) getAnswer() setAnswer(String Answer)	Notification_id : String Student : String Question : String Timestap : String Teacher : String <hr/> getNotification_id() setNotification_id(String Notification_id) getStudent() setStudent (String Student) getQuestion() setQuestion(String Question) getTimestap() setTimestap(String Timestap) getTeacher() setTeacher(String Teacher)
	AnswerStudent Answer_id : String Answer : String Student : String Status : String Notification : String <hr/> getAnswer_id() setAnswer_id(String Answer_id) getAnswer() setAnswer(String Answer) getStudent() setStudent(String Student) getStatus() setStatus(String Status) getNotification() setNotification(String Notification)	

Διάγραμμα 1 : Κλάσεις που αντιστοιχούν σε εγγραφές στην Βάση Δεδομένων

Δημιουργία εφαρμογής για Smartphones με σκοπό την υποστήριξη της εκπαιδευτικής διαδικασίας στην Εκπαίδευση

Ενδεικτικά παραδείγματα κλάσεων που αντιστοιχούν σε Activities και Background Services.

Παρατηρούμε ότι σε αυτές τις κλάσεις δεν υπάρχουν ιδιότητες, παρα μόνο λειτουργίες(function).Ο λόγος που γίνεται αυτό,είναι διότι οι συγκεκριμένες κλάσεις σχετίζονται άμεσα με τις λειτουργίες των Activities, όπως είναι εύκολα αντιληπτό πολλές απο αυτές τις λειτουργίες αντιστοιχούν σε event που χρησιμοποιούνται απο κάποιο component του εκάστοτε activity.

Η μοναδική από αυτές τις κλάσεις που απευθύνεται στο παρασκήνιο του λειτουργικού μας συστήματος είναι η κλάση backgroundtaskhandler που κληρονομεί την κλάση GcmTaskService και έχει να κάνει με την περιοδική επαφή του συστήματος ως προς τον χρήστη με πιθανές ενημερώσεις προς αυτόν.

<p>login</p> <pre>onCreate(@Nullable Bundle savedInstanceState) Task(String str) loginMethod(View view) register(View view)</pre>	<p>scheduletest2</p> <pre>onCreate(@Nullable Bundle savedInstanceState) fillspinerEverything() goCentreTeacher(View view) fillSpinnerQuestion() fillSpinnerQuestion2(List<String> list) addListenerOnSpinnerItemSelection() addListenerOnButton() onItemSelected(AdapterView<?> adapterView, View view, int i, long l) onNothingSelected(AdapterView<?> adapterView) initializeDateTime(View view) showTimePickerDialog(int h, int m) DateTimePickerDialog(View view) updateLabel()</pre>
<p>Register</p> <pre>onCreate(@Nullable Bundle savedInstanceState) fillSpinner() Submit(View view)</pre>	<p>maintainquestion</p> <pre>onCreate(@Nullable Bundle savedInstanceState) fillMyListView() onItemClick(AdapterView<?> adapterView, View view, int i, long l) goCreateQuestion(View view) deleteQuestion(View view) goCenterTeacher(View view)</pre>
<p>backgroundtaskhandler extends GcmTaskService</p> <pre>onRunTask(TaskParams taskParams) loginMethod(View view) register(View view)</pre>	

Διάγραμμα 2 : Κλάσεις που αντιστοιχούν σε Activities και σε Background Services

Δημιουργία εφαρμογής για Smartphones με σκοπό την υποστήριξη της εκπαιδευτικής διαδικασίας στην Εκπαίδευση

Παρουσίαση και Χρήση Εφαρμογής

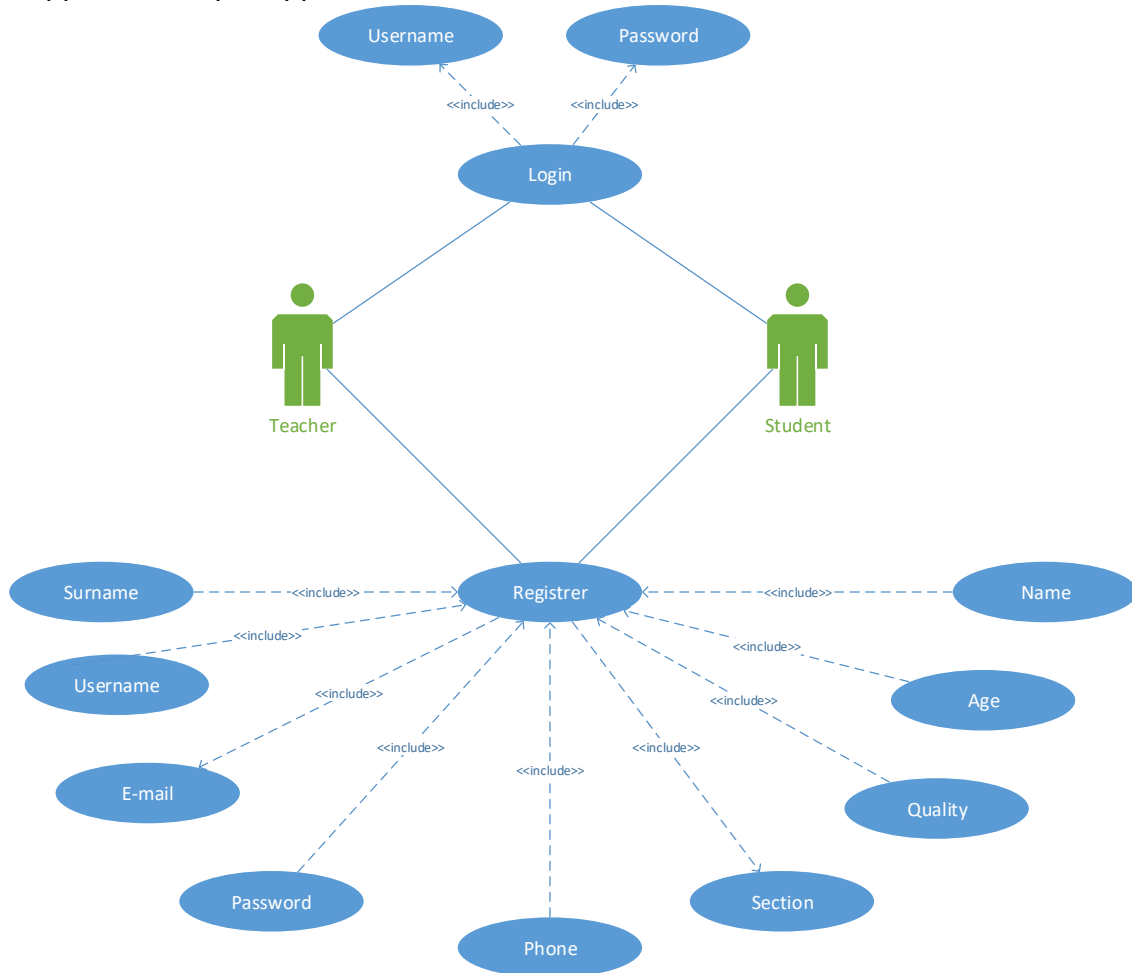
1. Είσοδος στην Εφαρμογή

Αρχικά όταν ένας χρήστης κάνει εγκατάσταση της εφαρμογής και ξεκινήσει την πλοήγηση του για πρώτη φορά θα αντικρύσει την φόρμα εισόδου, δηλαδή το activity Login. Εφόσον ο χρήστης δεν έχει κάνει ξάνα είσοδο στην εφαρμογή θα πρέπει να εισέλθει στην φόρμα Register και να καταγράψει τα στοιχεία του. Εάν ο χρήστης είναι δάσκαλος τότε θα πρέπει επιπροσθέτως να κλικάρει και το αντίστοιχο checkbox στο τέλος της φόρμας και εν συνεχεία οποιοσδήποτε χρήστης είτε μαθητής είτε καθηγητής για να ολοκληρώσει την καταγραφή του θα πρέπει να πατήσει «Submit». Αφού ο χρήστης ολοκληρώσει την καταγραφή του θα επανέλθει στην αρχική σελίδα «Login», όπου πλέον είτε ως μαθητής είτε ως καθηγητής θα πρέπει να πληκτρολογήσει σωστά τα στοιχεία του έτσι ώστε να ξεκινήσει την πλοήγηση του στην εφαρμογή.

Φόρμα 1 : Login

Φόρμα 2 : Register

Σε αυτό το διάγραμμα παρατηρούμε την πληροφορία που θα εισάγει ο χρήστης στην Φόρμα 1 και στην Φόρμα 2.

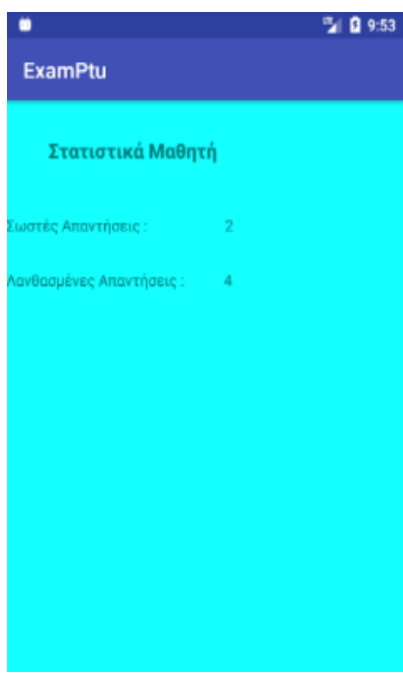


Use Case Diagram 1

Σχόλιο 1 : Σε αυτό το πρώτο βήμα θα ήτο φρόνιμο να αναφέρουμε ότι το να κάνεις εισαγωγή μέσα στην εφαρμογή έστω μία φορά είναι αναγκαίο προκειμένου να ξεκινήσουν οι διεργασίες στο παρασκήνιο του λειτουργικού μας συστήματος.

2. Σενάριο Λειτουργιών του χρήστη ως Μαθητής

Αφού εισέλθει ο μαθητής στην εφαρμογή (είτε την πρώτη είτε οποιαδήποτε άλλη φορά) το μόνο που θα μπορεί να δει είναι το πόσες ερωτήσεις έχει απαντήσει σωστά και πόσες λανθασμένα. Διότι σύμφωνα με το σχόλιο 1 (βλ. σελίδα16) η ουσιαστική διεργασία που ξεκινά κατά την αρχική είσοδο στην εφαρμογή δεν είναι απλά το ξεκίνημα μια στατικής ενέργειας στο καινούργιο μου app, αλλά ο διάυλος μίας επικοινωνίας που πλέον θα έχω σε RealTime, εφόσον ο καθηγητής μου χρονοπρογραμματίσει ερωτήσεις εξατομικευμένες ή όχι ως προς τον ίδιο ή την τάξη όπου αυτός ανήκει.

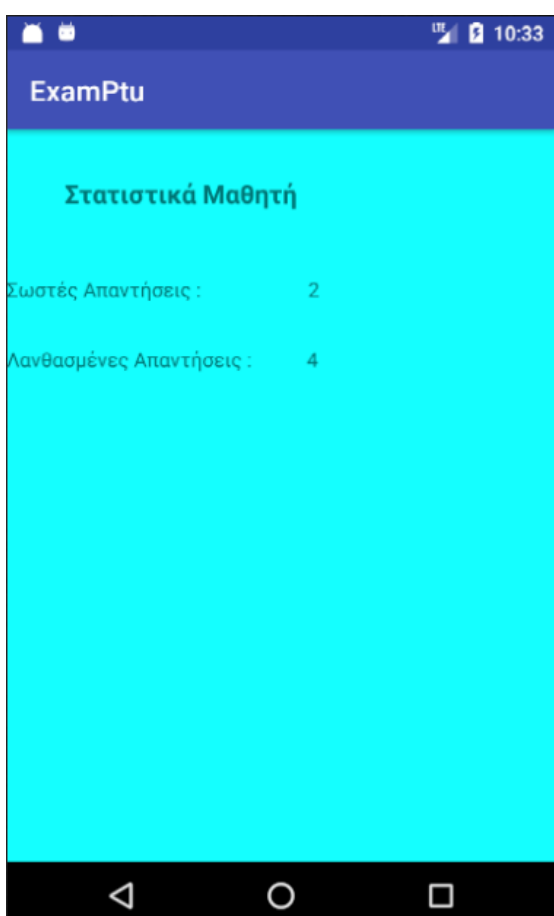


Φόρμα 1 : Στατιστικά Μαθητή

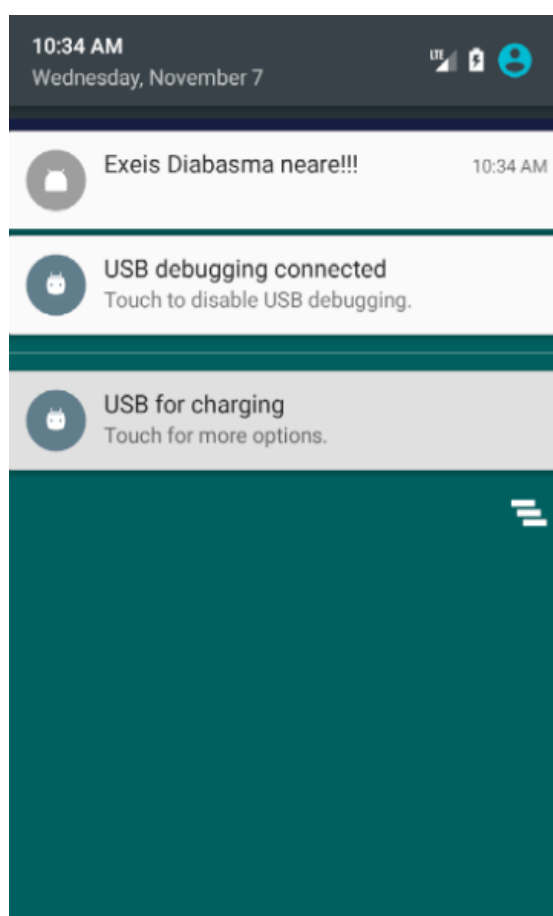
Ο μαθητής λοιπόν θα ενημερωθεί για το συνολικό αριθμό σωστών και λανθασμένων απαντήσεων που θα έχει απαντήσει και θα έχουν αντίστοιχα αξιολογηθεί από τον επιβλέπων καθηγητή του όπως βλέπουμε στην φόρμα 3.

Ωστόσο όταν θα πρέπει να απαντήσει σε μία καινούργια ερώτηση τότε θα ενημερώνεται μέσω ενός notification. Η κάθε ερώτηση αντιστοιχεί σε μία συγκεκριμένη ημερομηνία.

Παράδειγμα εάν ένας καθηγητής στείλει μια ερώτηση σε έναν μαθητή του στις 6 Δεκεμβρίου τότε και μόνο τότε θα ενημερωθεί ο μαθητής με notification. Σε περίπτωση που ο μαθητής είτε κατα λάθος είτε εξ επίτηδες θελήσει να παρακάμψει το notification χωρίς να απαντήσει στην ερώτηση τότε θα ξαναλάβει το ίδιο notification.

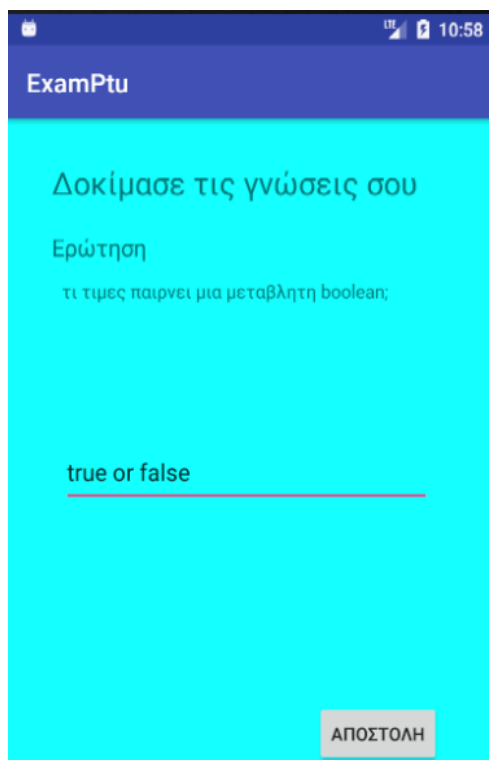


Φόρμα 3 : Ειδοποίηση με Notification



Φόρμα 2 : Άνοιγμα λίστας Notification

Εάν λοιπόν ο μαθητής ανοίξει την λίστα από τα notification που έχει παραλάβει και κλικάρει το καινούργιο notification τότε θα μεταβεί στην Φόρμα 6.



Φόρμα 4 : Απάντηση μαθητή σε χρονοπρογραμματισμένη ερώτηση

Αφού διαβάσει και απαντήσει την ερώτηση του, για να ολοκληρωθεί η διαδικασία θα πρέπει να πατήσει αποστολή. Το πώς θα ελεγχθεί η απάντηση που έδωσε ο μαθητής, θα το δούμε παρακάτω όπου θα δούμε το workflow του καθηγητή.

Είναι σημαντικό να διευκρινήσουμε ότι οι απαντήσεις που στέλνει ένας μαθητής ελέγχονται από τον καθηγητή και όχι μέσω κάποιων default απαντήσεων.

[Μια σημαντική παρατήρηση που πρέπει επίσης να διατυπώσουμε είναι ότι ο χρήστης θα μπορεί να λαμβάνει notification και όταν είναι εκτός εφαρμογής.](#)

Δημιουργία εφαρμογής για Smartphones με σκοπό την υποστήριξη της εκπαιδευτικής διαδικασίας στην Εκπαίδευση

3. Σενάριο Λειτουργιών του χρήστη ως Καθηγητής

Στο ρόλο του καθηγητή θα παρατηρήσουμε ότι υπάρχουν περισσότερες λειτουργίες συγκριτικά με τον ρόλο του μαθητή, κάτι το οποίο είναι πολύ φυσιολογικό. Διότι ο καθηγητής θα αναλάβει να δημιουργήσει ερωτήσεις-θέματα, θα εξετάσει στατιστικά ανά ταξη ή ανα μαθητή και θα προγραμματίσει το πότε θα λαμβανουν είτε η ταξη ολόκληρη είτε ενάς μαθητής τις ερωτήσεις που αυτός θα έχει ορίσει.

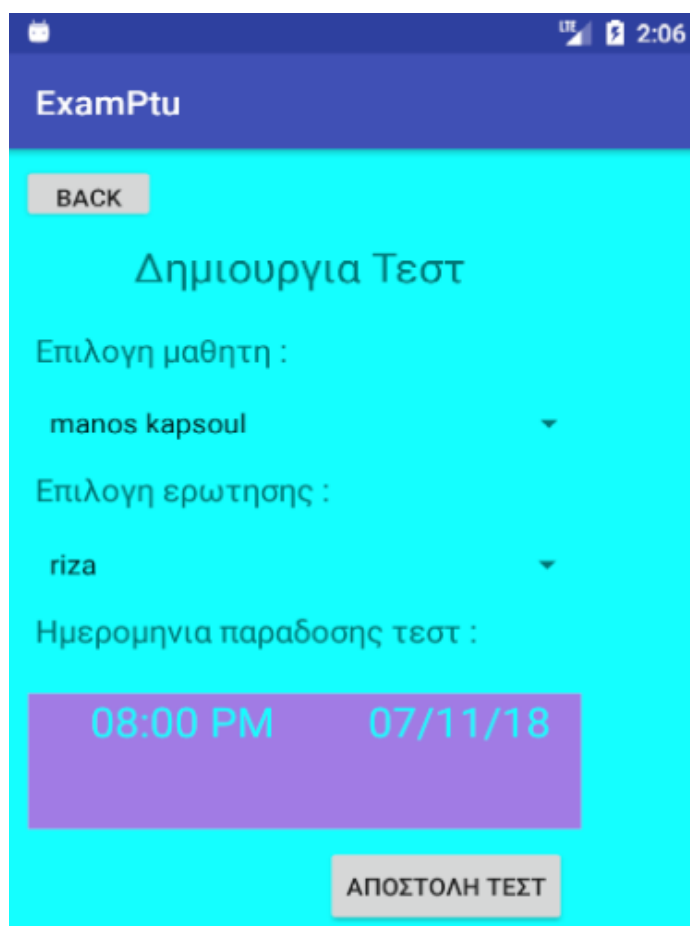


Φορμα 5 : Κεντρική φόρμα καθηγητή.

Για αυτό το λόγο θα χωρίσουμε την λειτουργικότητα του καθηγητή σε επιμέρους ενότητες.

3.1 Εξέταση ανα μαθητή

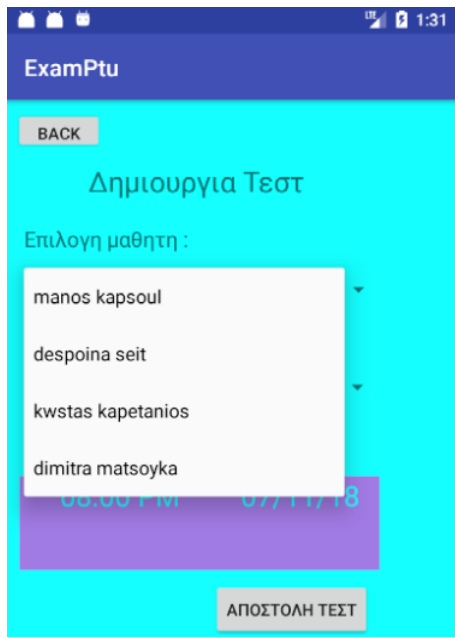
Παρατηρώντας την Φόρμα 7 βλέπουμε ότι αποτελείται από τέσσερα buttons. Εάν λοιπόν πατήσω το button «ΕΞΕΤΑΣΗ ΑΝΑ ΜΑΘΗΤΗ» θα μεταβώ στην Φόρμα 8.



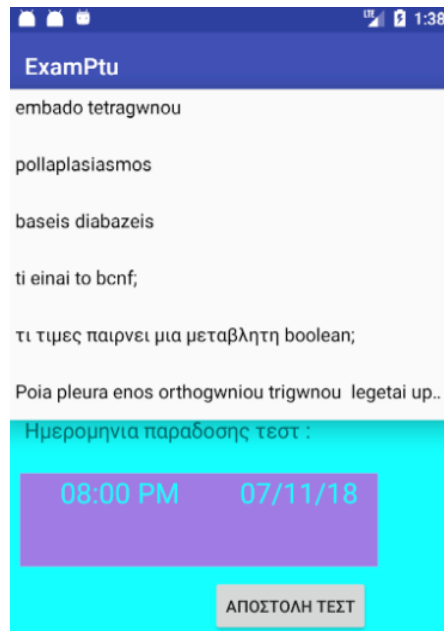
Φόρμα 6 : Προγραμματισμός ερώτησης ανα μαθητή

Εδώ ο καθηγητής θα πρέπει να επιλέξει τον μαθητή που θέλει να εξετάσει, την ερώτηση που θέλει να του αναθέσει καθώς επίσης και την ημερομηνία και ώρα που ο μαθητής θα παραλάβει το notification. Μόλις συμπληρώσει όλα τα κριτήρια, πρέπει να αποστείλει την ερώτηση που προγραμμάτισε πατώντας το button «ΑΠΟΣΤΟΛΗ ΤΕΣΤ».

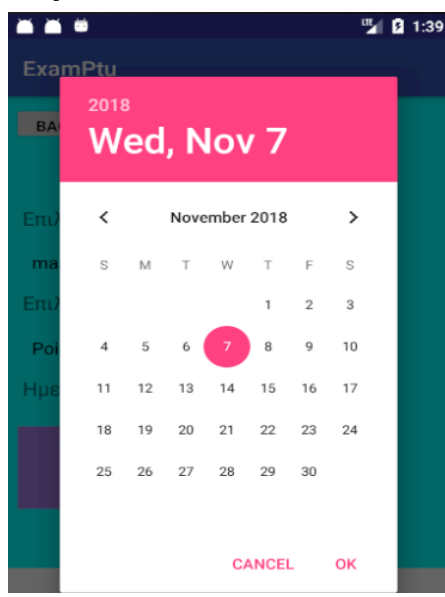
Ακολουθούν σχετικές εικόνες με αυτά τα τρία κριτήρια.



Εικόνα 4 : Λίστα Μαθητών



Εικόνα 5 : Λίστα Ερωτήσεων

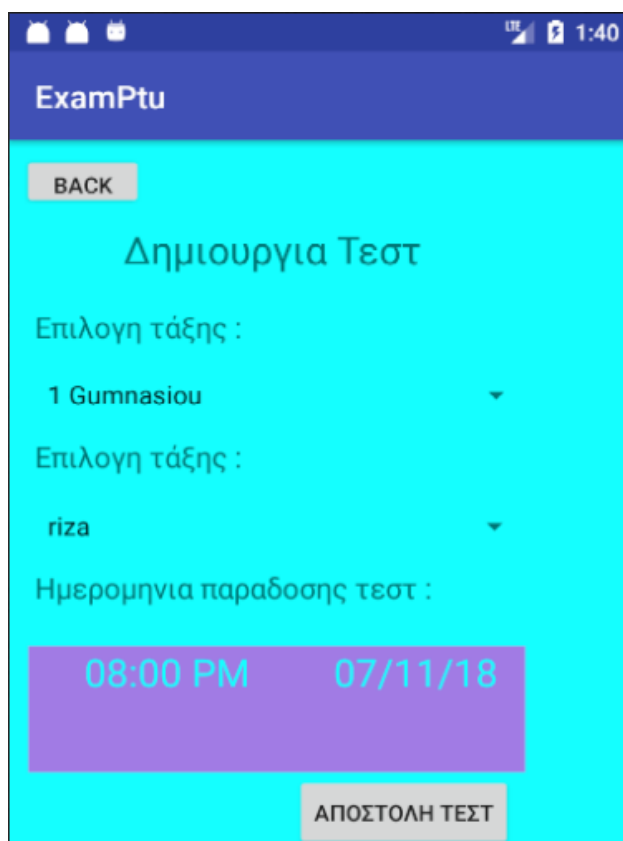


Εικόνα 6 : Επιλογή Ημερομηνίας

Δημιουργία εφαρμογής για Smartphones με σκοπό την υποστήριξη της εκπαιδευτικής διαδικασίας στην Εκπαίδευση

3.2 Εξέταση ανα τάξη

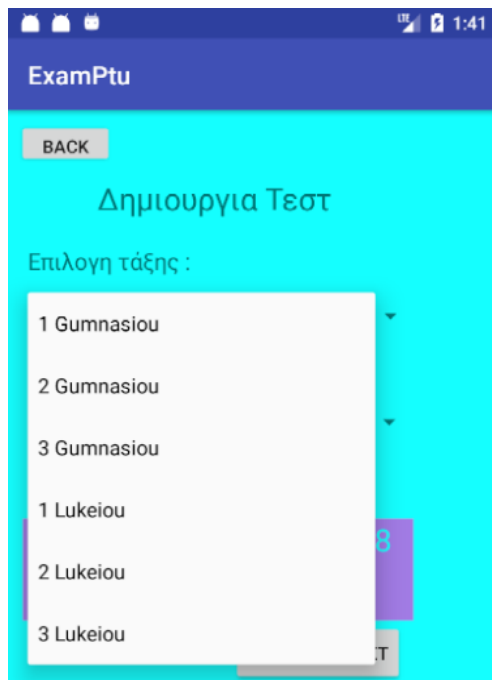
Εάν λοιπόν πατήσω το button «ΕΞΕΤΑΣΗ ΑΝΑ ΤΑΞΗ» από την Φόρμα 7 θα μεταβώ στην Φόρμα 9.

The screenshot shows a mobile application interface titled "ExamPtu". At the top left is a "BACK" button. The main heading is "Δημιουργια Τεστ". Below this, there are three form fields: "Επιλογη τάξης :" with a dropdown menu showing "1 Gymnasiou", another "Επιλογη τάξης :" with a dropdown menu showing "riza", and "Ημερομηνια παραδοσης τεστ :". Below these fields is a date and time picker showing "08:00 PM" and "07/11/18". At the bottom right is a button labeled "ΑΠΟΣΤΟΛΗ ΤΕΣΤ".

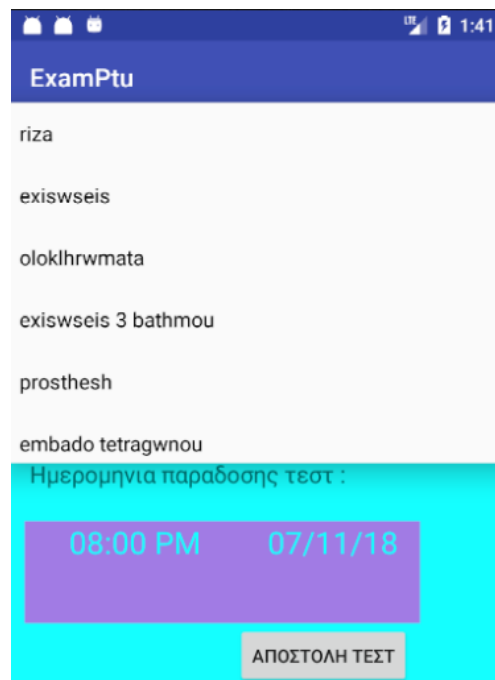
Φόρμα 7 : Προγραμματισμός ερώτησης ανα τάξη

Εδώ ο καθηγητής θα πρέπει να επιλέξει την τάξη που θέλει να εξετάσει, την ερώτηση που θέλει να τους αναθέσει καθώς επίσης και την ημερομηνία και ώρα που οι μαθητες της τάξης θα παραλάβουν το notification. Μόλις συμπληρώσει όλα τα κριτήρια, πρέπει να αποστείλει την ερώτηση που προγραμμάτισε πατώντας το button «ΑΠΟΣΤΟΛΗ ΤΕΣΤ».

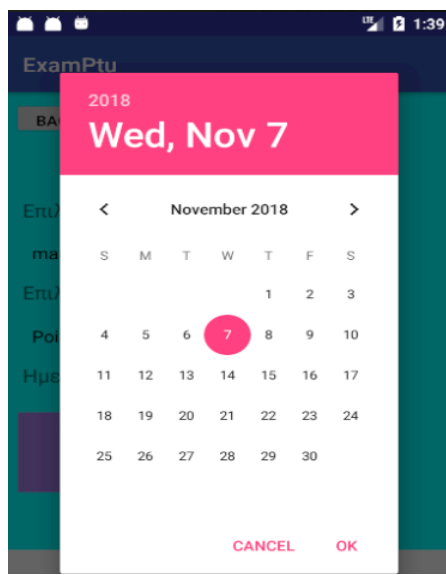
Ακολουθούν σχετικές εικόνες με αυτά τα τρία κριτήρια.



Εικόνα 7 : Λίστα Τάξεων



Εικόνα 8 : Λίστα Ερωτήσεων

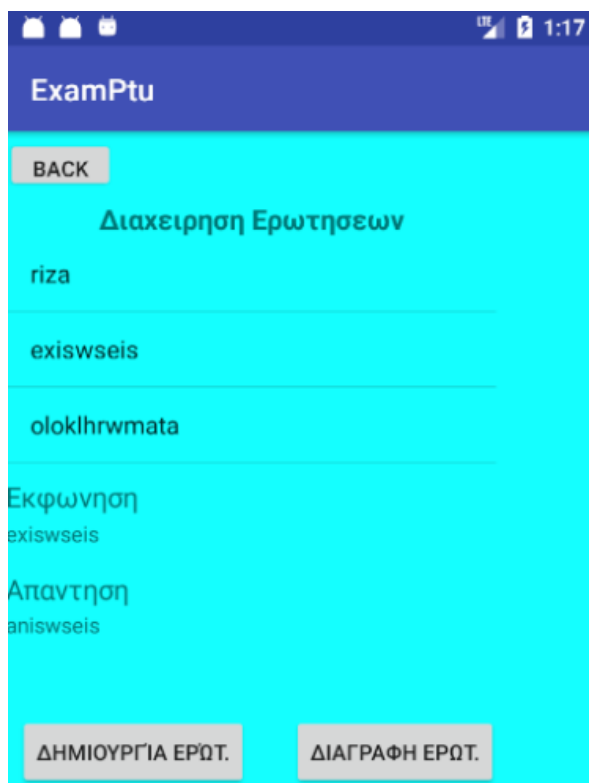


Εικόνα 9 : Επιλογή Ημερομηνίας

Δημιουργία εφαρμογής για Smartphones με σκοπό την υποστήριξη της εκπαιδευτικής διαδικασίας στην Εκπαίδευση

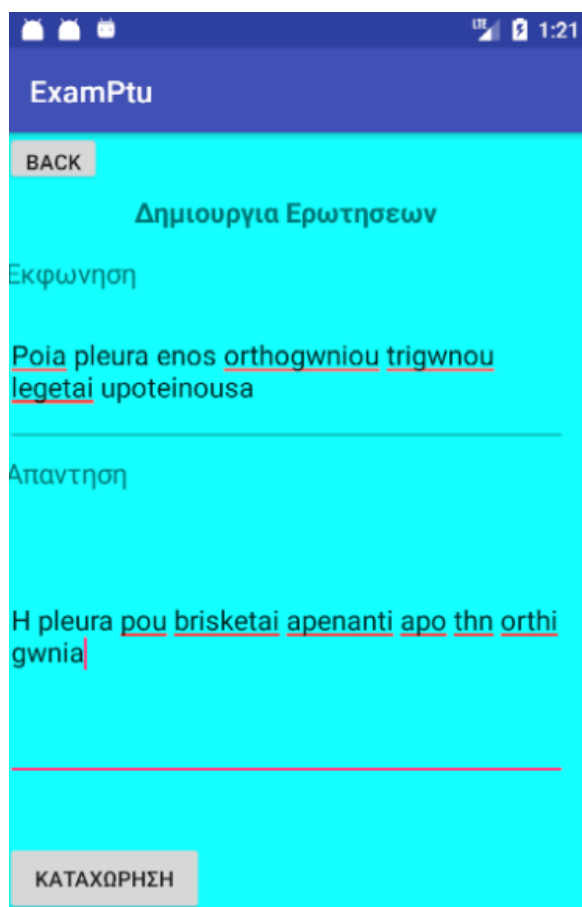
3.3 Διαχείριση Ερωτήσεων

Εάν λοιπόν πατήσω το button «ΔΙΑΧΕΙΡΗΣΗ ΕΡΩΤΗΣΕΩΝ» από την Φόρμα 7 θα μεταβώ στην Φόρμα 10.



Φόρμα 8 : Διαχείριση Ερωτήσεων

Σε αυτήν την φόρμα ο καθηγητής θα έχει την ευκαιρία να δει την λίστα από όλες τις ερωτήσεις που έχει δημιουργήσει κατά το παρελθόν. Θα μπορέσει είτε να διαγράψει κάποιες από αυτές είτε να προσθέσει κάποιες καινούργιες. Για να επιτύχει μια από αυτές τις λειτουργίες θα πρέπει να πατήσει τα αντίστοιχα button. Για να διαγράψει λοιπόν μία ερώτηση θα πρέπει πρώτα να διαλέξει μία από την λίστα που βλέπουμε στην Φόρμα 10 και στην συνέχεια να πατήσει το button «ΔΙΑΓΡΑΦΗ ΕΡΩΤ.». Εφόσον επιθυμεί να εισάγει μια καινούργια ερώτηση στην λίστα τότε θα πρέπει να πατήσει το αντίστοιχο button και να μεταφερθεί στην Φόρμα 11.

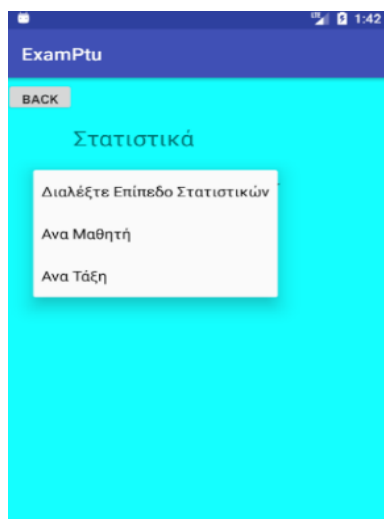


Φόρμα 9 : Δημιουργία Ερωτήσεων

Σε αυτήν την φόρμα ο καθηγητής θα μπορεί να εισάγει μια καινούργια ερώτηση και μία προτεινόμενη απάντηση.Θά σταθούμε στο κομμάτι της προτεινόμενης απάντησης,καθώς ο λόγος που υλοποιήθηκε με αυτόν τον τρόπο έχει να κάνει με τον βέλτιστο έλεγχο των πιθανών απαντήσεων που επρόκειτο να δεχθεί κατα τις απαντήσεις των μαθητών προς αυτόν.Θα δούμε παρακάτω την εφαρμογή αυτής της μεθοδολογίας και πως θα αυτοματοποιήσει το τρόπο ελέγχου.Αφού συμπληρώσει ο καθηγητής τα πεδία,πατώντας το πλήκτρο «ΚΑΤΑΧΩΡΗΣΗ» θα έχει εισάγει πλέον την καινούργια ερώτηση στην λίστα και θα μπορεί να την χρησιμοποιήσει για την διεπαφή με τους μαθητές του.

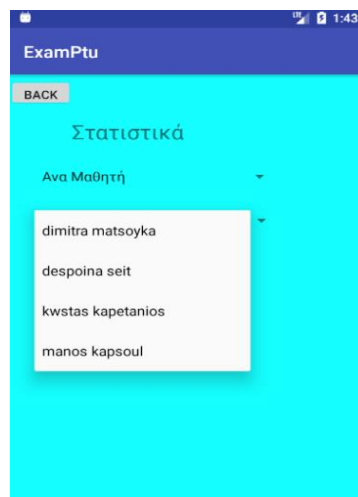
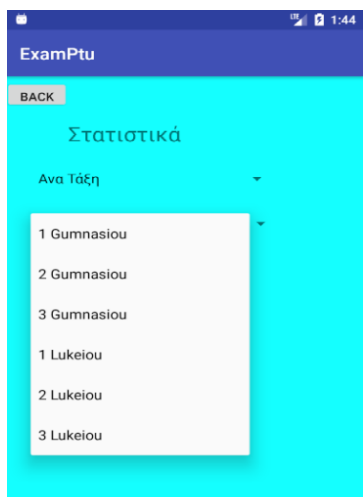
3.4 Στατιστικά

Εάν λοιπόν πατήσει το button «ΣΤΑΤΙΣΤΙΚΑ» από την Φόρμα 7 θα μεταβεί στην Φόρμα 12. Θα πρέπει σε πρώτη φάση να διαλέξει εάν θέλει να δει συγκεντρωτικά αποτελέσματα ανα τάξη ή ατομικά ανα μαθητή.



Φόρμα 10 : Στατιστικά ανα μαθητή ή ανα τάξη

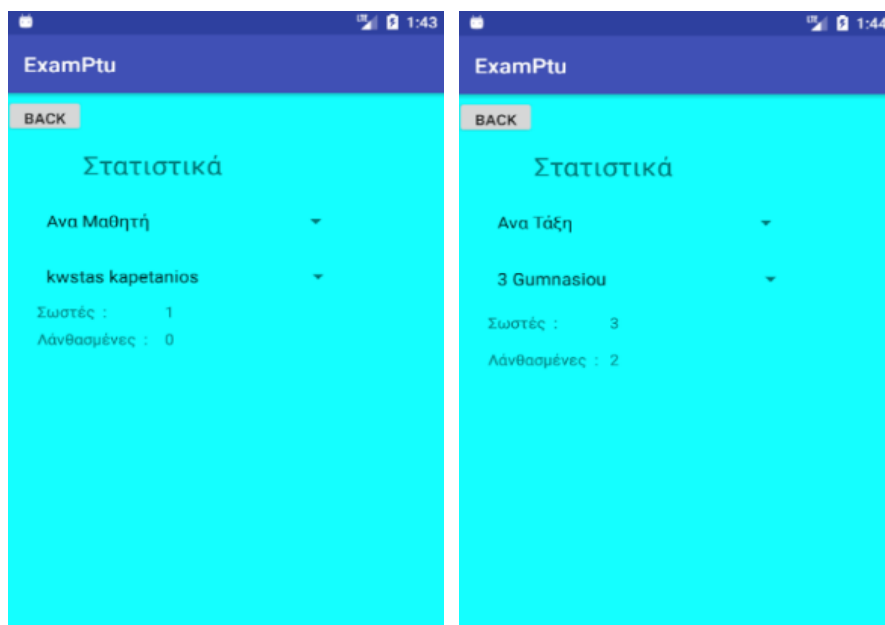
Ανάλογα με την επιλογή που θα κάνει θα πρέπει να επιλέξει κάποιο πεδίο της λίστας από την αντίστοιχη κατηγορία.



Εικόνα 10 : Επιλογή ανα τάξη **Εικόνα 11 : Επιλογή ανα μαθητή**

Δημιουργία εφαρμογής για Smartphones με σκοπό την υποστήριξη της εκπαιδευτικής διαδικασίας στην Εκπαίδευση

Τελική εμφάνιση των στατιστικών τόσο μέσα απο το φιλτράρισμα ανα τάξη όσο και για τον κάθε μαθητή μεμονωμένα.



Εικόνα 12 : Εμφάνιση Στατιστικών

Μέσα από τα στατιστικά στοιχεία που λαμβάνει ο καθηγητής τόσο σε επίπεδο τάξης όσο και σε επίπεδο μαθητή, θα έχει την δυνατότητα να αφομοιώνει και να αντιλαμβάνεται σε μεγαλύτερο βάθος, το βαθμό κατανόησης των μαθητών του. Με αυτόν τον τρόπο ο καθηγητής θα μπορεί να τους ανατροφοδοτεί άμεσα και έγκαιρα με καινούργιες ερωτήσεις αλλά και με επανάληψη παλαιότερων με σκοπό την περαιτέρω εκπαίδευση των μαθητών. Η στατιστική ανάλυση των σωστών και λανθασμένων απαντήσεων, που θα είναι καταγεγραμμένες μέσα σε μία βάση δεδομένων, θα αποτελέσει το μοχλό με τον οποίο ο καθηγητής θα μπορέσει να βοηθήσει και εν τέλει να μεταδώσει τις γνώσεις του στους μαθητές του μέσα σε ένα σύγχρονο και δυναμικό περιβάλλον όπως αρμόζει στην εκπαίδευση του σήμερα.

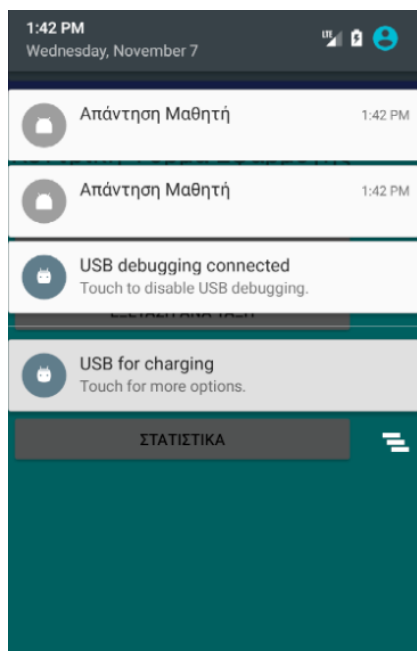
3.5 Ειδοποιήσεις Απαντήσεων

Παρατηρώντας τις περισσότερες φόρμες όπου αναλύσαμε αν κοιτάξουμε πιο προσεκτικά θα καταλάβουμε ότι ο καθηγητής είχε δύο notification όπου δεν είχε απαντήσει.



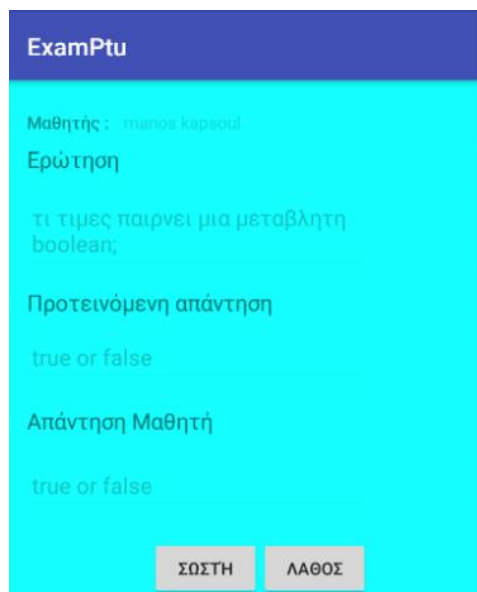
Εικόνα 13 : Εμφάνιση Notification

Τα notification ουσιαστικά ήταν ερωτήσεις που οι μαθητές έχουν απαντήσει και έχουν επιστρέψει πλέον πίσω στον καθηγητή προκειμένου να χαρακτηριστούν ως σωστές ή λάθος. Αν ανοίξουμε την λίστα των notifications θα διαπιστώσουμε το παρακάτω σενάριο.



Εικόνα 14 : Notification που λαμβάνει ο καθηγητής

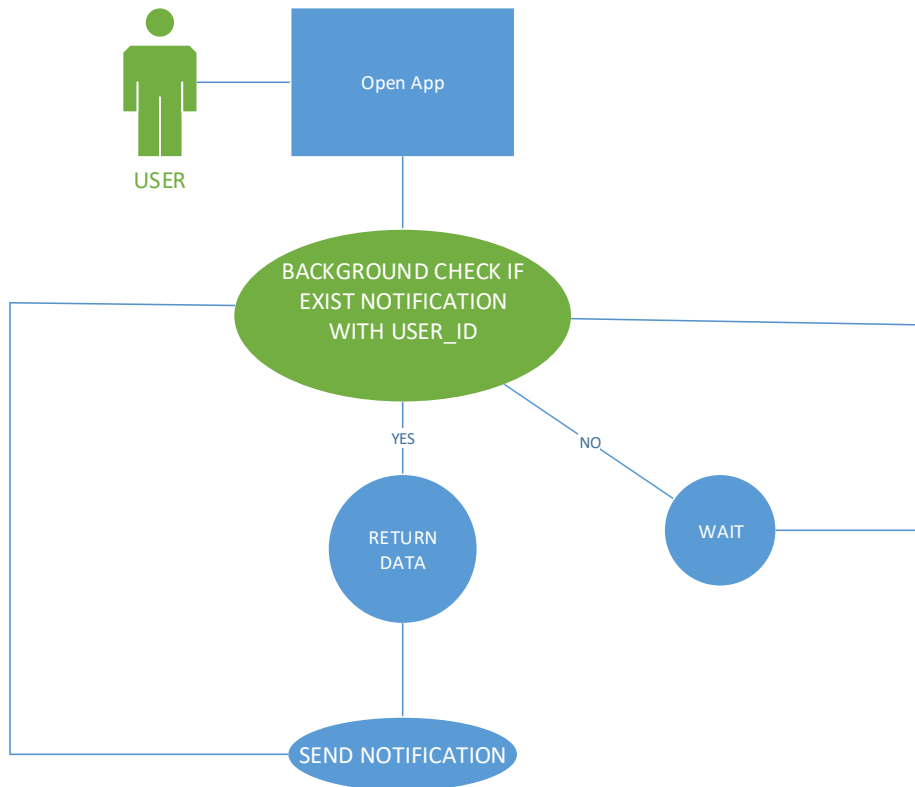
Ανοίγοντας λοιπόν ένα notification ο καθηγητής θα μεταφερθεί στην Φόρμα 12.



Φόρμα 11 : Επιστροφή της ερώτησης με την απάντηση που έχει δώσει ο εκάστοτε μαθητής

Σε αυτήν την φόρμα ο καθηγητής μπορεί να ελέγξει την απάντηση του κάθε μαθητή. Παρατηρούμε ότι ο καθηγητής γνωρίζει ποιος μαθητής έχει απαντήσει. Επιπροσθέτως όχι μόνο βρίσκει μπροστά του την απάντηση του μαθητή αλλά και την ερώτηση μαζί με την προτεινόμενη απάντηση. Ουσιαστικά αυτός είναι ο λόγος που δημιουργήθηκε το πεδίο προτεινόμενη απάντηση προκειμένου ο καθηγητής να μπορεί εύκολα να κατηγοριοποιήσει μια απάντηση κατά πόσο είναι σωστή ή λάθος. Έχοντας μπροστά του την απάντηση που ο ίδιος έχει δώσει είναι πολύ πιο εύκολο να συγκρίνει με την απάντηση του μαθητή και να αποφασίσει. Επιπλέον μπορεί κάποιος μαθητής να μην έχει απαντήσει με τον ίδιο ακριβώς τρόπο αλλά παρόλα αυτά να είναι σωστή η απάντηση του, συνεπώς συγκρίνοντας τις δύο απαντήσεις είναι ευκολότερο και γρηγορότερο να ελέγξεις και να κατηγοριοποιήσεις.

Ανάλυση Περίπτωσης Χρήσης στο Παρασκήνιο



Use Case Diagram 2

Ουσιαστικά παρατηρούμε το workflow που γίνεται στο παρασκήνιο της εφαρμογής. Θα μπορούσαμε να το ερμηνεύσουμε και σαν την πηγή έμπνευσης προκειμένου να αναλυθεί και εν τέλει να υλοποιηθεί αυτή η εφαρμογή. Αυτό που προσπαθήσαμε να αποδώσουμε συσχετιζόταν σε μία αέναη και αλληλένδετη πηγή συνεργασίας, ούτως ώστε να καταλήξουμε στον βέλτιστο και συνάμα διασκεδαστικό τρόπο εκπαίδευσης. Σκοπός είναι στην μάθηση να μην υπάρχουν όρια στον χρόνο της εκπαίδευσης ούτε δομές όπου δε συνάδουν με το σύγχρονο τεχνολογικό γίνεσθαι της επόχης. Όπως τα social media έχουν γίνει αναπόσπαστο κομμάτι της ζωής του μέσου ανθρώπου, έτσι και η εκπαίδευση θα ήταν σκόπιμο να ακολουθήσει τα ίδια τεχνολογικά μονοπάτια ή και να τα ξεπεράσει.

Συμπεράσματα και μελλοντικές επεκτάσεις

Στην παρούσα μεταπτυχιακή διατριβή αναπτύχθηκε και παρουσιάστηκε το πώς θα πρέπει να είναι απο εδώ και πέρα μία σύγχρονη εφαρμογή εκπαίδευσης. Το κομμάτι της on-line διεπαφής του καθηγητή με τον εκάστοτε μαθητή θα πρέπει να είναι το κυριάρχο θέμα που θα μας απασχολήσει ακόμα σε πιο έντονο βαθμό στο μέλλον. Οι καινούργιοι μαθητές είναι εκ γενετής εξοικειωμένοι με τον κόσμο του internet. Η διαδικασία διάδρασης ενός νέου με το κινητό του φτάνει σε επίπεδα καθημερινής εξάρτησης. Άρα θα ήταν λάθος να μην προσπαθήσουμε να αναπτύξουμε εποικοδομητικές διαδικασίες και μηχανισμούς σύγχρονης εκπαίδευσης.

Σίγουρα υπάρχουν αρκετές εφαρμογές πάνω στην εκπαίδευση και αρκετές από αυτές έχουν πάρα πολύ όμορφο front-end προκειμένου να προσελκύσουν περισσότερους χρήστες, όμως το σημαντικότερο κομμάτι που δυστυχώς υπολείπονται πολλές από αυτές είναι ο τομέας της διαδραστικότητας που θα έπρεπε να της καθορίζει. Αυτό είναι το σημαντικότερο κομμάτι που θέλει να αναζητήσει και να αναδείξει αυτή η μεταπτυχιακή διατριβή. Σίγουρα υπολείπεται πολλών πραγμάτων έτσι ώστε να μπορεί να θεωρηθεί μία εμπορική εφαρμογή, ωστόσο το θέμα που θέλει να αναδείξει είναι σημαντικότερο από την εφαρμογή αυτή καθ'αυτή. Παρόλα αυτά έγινε μια αρκετά μεγάλη προσπάθεια προκειμένου να μπορέσουμε να προσομοιώσουμε σε αρκετά ικανοποιητικό βαθμό αυτή την ιδέα.

Το σύστημα επιδέχεται αρκετές επεκτάσεις και βελτιώσεις, για να φτάσει σε επίπεδο εμπορικής εφαρμογής. Ενδεικτικά προτάσσονται τα παρακάτω :

- > Παροχή οδηγιών χρήσης εντός της εφαρμογής
- > Απεικόνιση σωστών και λανθασμένων απαντήσεων εντός του android app
- > Βελτίωση του αλγόριθμου ή επιλογή άλλου αλγόριθμου.
- > Μελέτη και εφαρμογή για συμμόρφωση με το GDPR. Το GDPR είναι ο ευρωπαϊκός κανονισμός για την προστασία προσωπικών δεδομένων ο οποίος έχει τεθεί σε ισχύ από τις 25 Μαΐου 2018.
- > Έλεγχος ύπαρξης σύνδεσης στο διαδίκτυο και ενημέρωση του χρήστη, εάν δεν υπάρχει.

Βιβλιογραφικές αναφορές-διαδικτυακές πηγές

Χρήσιμοι υπερσύνδεσμοι που χρησιμοποιήθηκαν κατά την υλοποίηση της εφαρμογής :

<https://stackoverflow.com/>

<https://developer.android.com/index.html>

<https://www.eugdpr.org/>

Υπερσύνδεσμοι από παρεμφερείς εφαρμογές :

«Το Λυσάρι»

<https://play.google.com/store/apps/details?id=com.iproject.lysari>

«Μαθηματικά Γ Λυκείου – Πανελλήνιες»

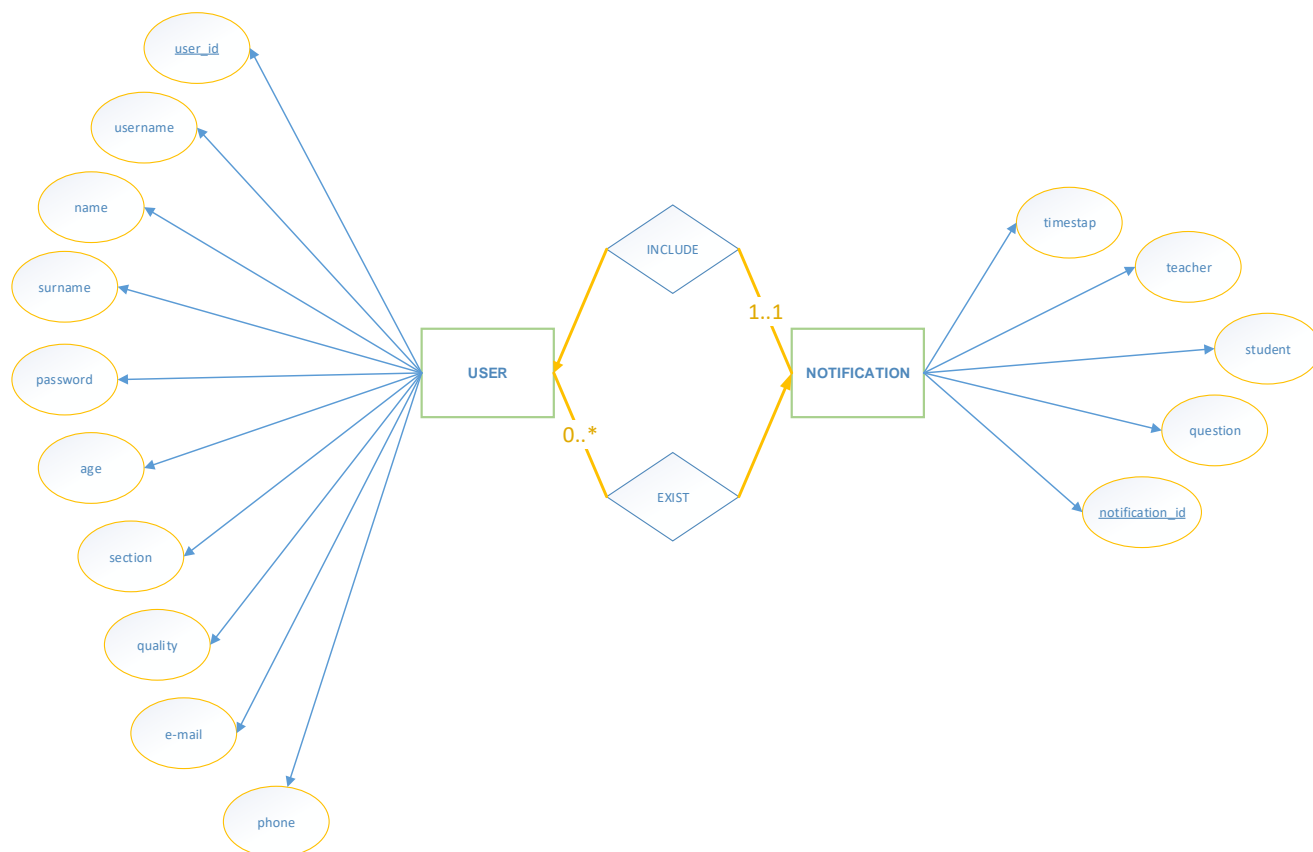
<https://play.google.com/store/apps/details?id=com.vasilis.epanalglyk>

«LySTe – Τεστ Λυκείου»

<https://play.google.com/store/apps/details?id=com.mstamos.quizv02>

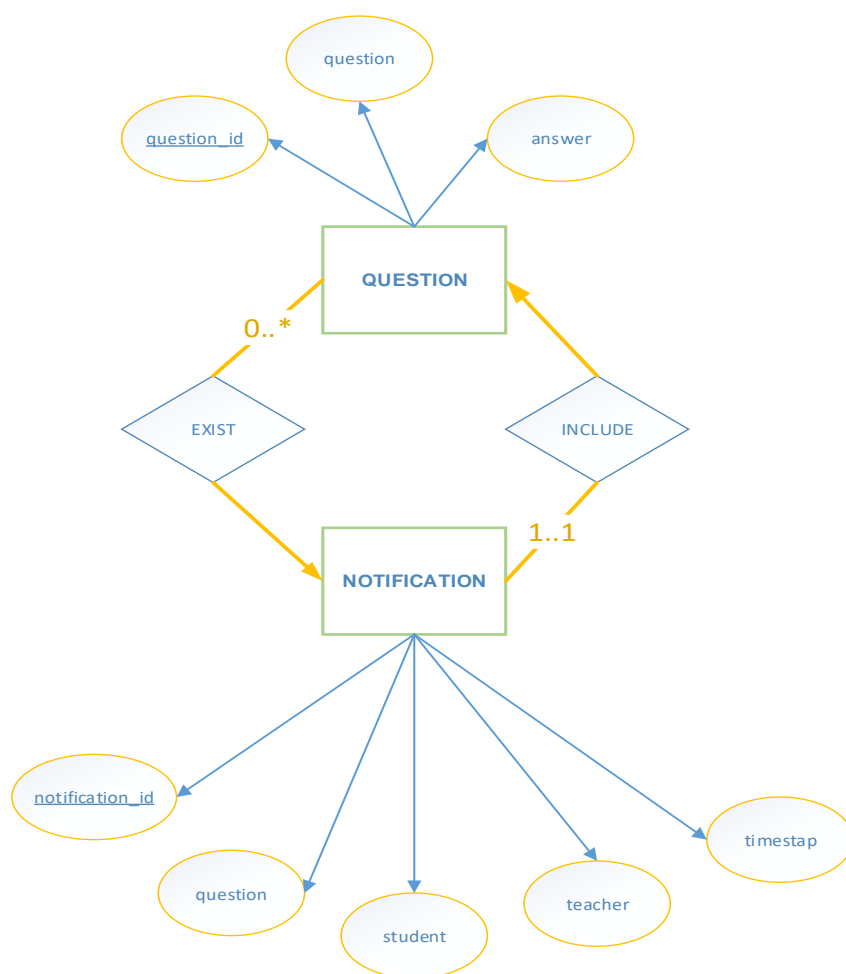
Παράρτημα

Διαγραμματική Παρουσίαση Πινάκων Βάσης



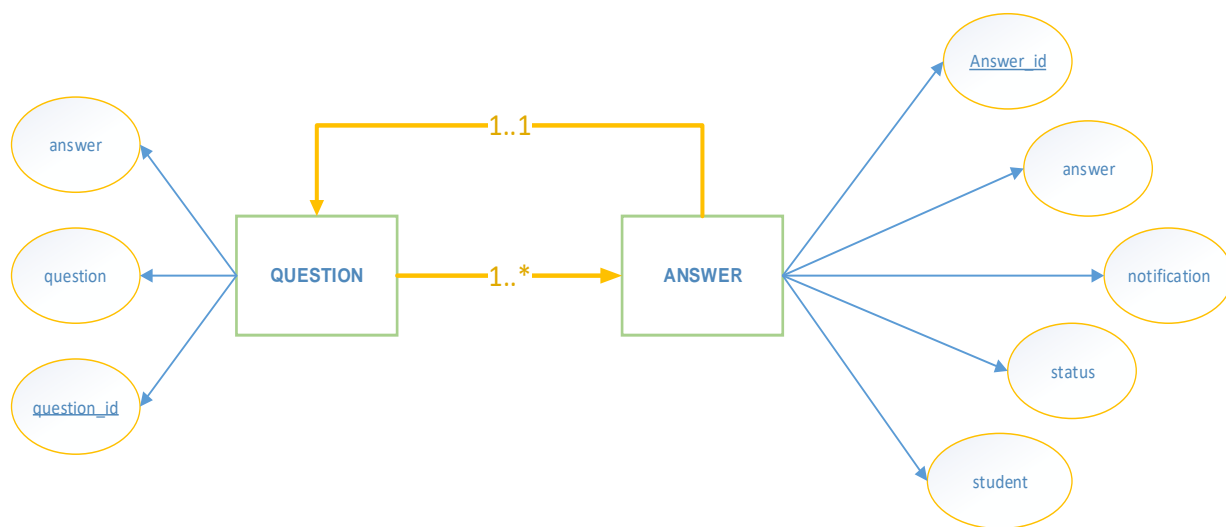
Σχισιακό Νοητό Διάγραμμα 1 : Πινάκων USER - NOTIFICATION

Παρατηρούμε ότι ένας χρήστης μπορεί να έχει από μηδέν έως άπειρες ειδοποιήσεις, ενώ μία ειδοποίηση να αντιστοιχεί κάθε φορά σε ένα συγκεκριμένο χρήστη. Η συσχέτιση «EXIST» αντιστοιχεί σε μηδέν προς πολλά, ενώ αντίστροφα η συσχέτιση «INCLUDE» σε ένα προς ένα.



Σχεσιακό Νοητό Διάγραμμα 2 : Πινάκων QUESTION - NOTIFICATION

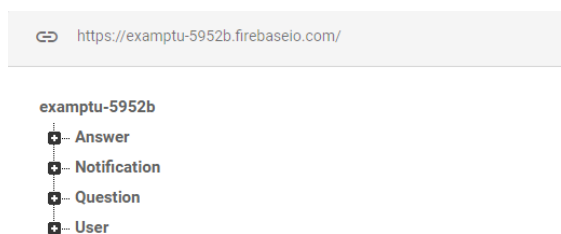
Παρατηρούμε ότι μία ερώτηση μπορεί να εντάσσει σε μηδέν έως άπειρες ειδοποιήσεις, ενώ μία ειδοποίηση μπορεί να αντιστοιχεί μόνο σε μία ερώτηση. Η συσχέτιση «EXIST» αντιστοιχεί σε μηδέν προς πολλά, ενώ αντίστροφα η συσχέτιση «INCLUDE» σε ένα προς ένα.



Σχεσιακό Νοητό Διάγραμμα 3 : Πινάκων QUESTION - ANSWER

Παρατηρούμε ότι μία ερώτηση μπορεί να αντιστοιχεί από μία μέχρι πολλές απαντήσεις, παράλληλα μία απάντηση αντιστοιχεί μόνο σε μία ερώτηση. Δηλαδή έχουμε ένα προς πολλά και αντίστροφα ένα προ ένα συσχετίσεις.

Εικονικό Interface στο Cloud



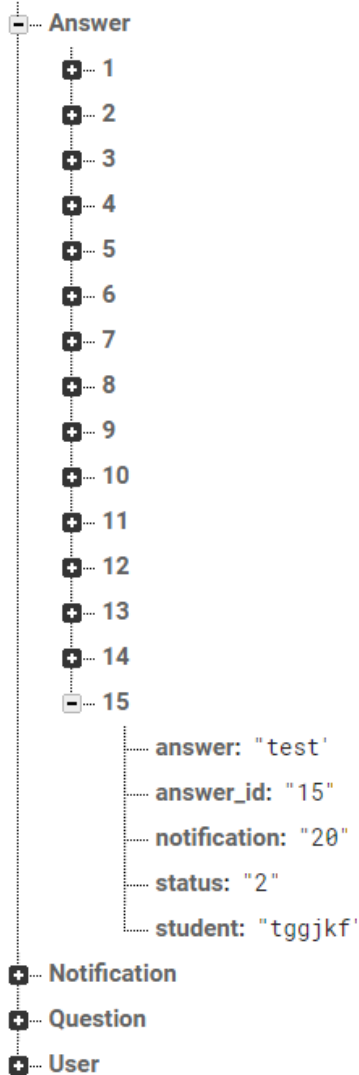
Firestore Database 1

Στο στιγμιότυπο «Firestore Database 1» παρατηρούμε την κεντρική δομή αποθήκευσης της εφαρμογής ExamPtu.

Δημιουργία εφαρμογής για Smartphones με σκοπό την υποστήριξη της εκπαιδευτικής διαδικασίας στην Εκπαίδευση

<https://examptu-5952b.firebaseio.com/>

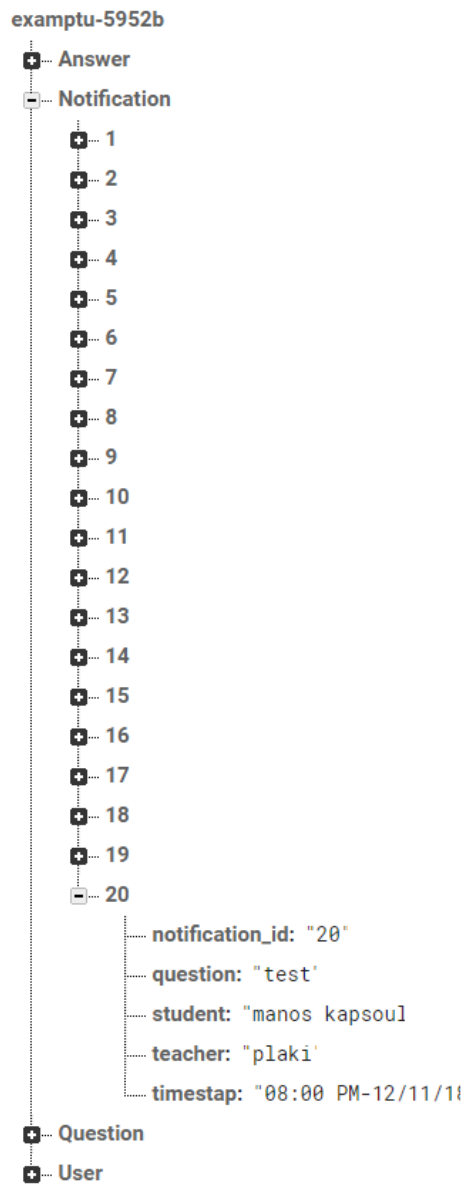
examptu-5952b



Firestore Database 2

Στο στιγμιότυπο «Firestore Database 2» παρατηρούμε πώς αποθηκεύουμε την απάντηση όπου δίνει ένας μαθητής και πώς αυτή έχει αξιολογηθεί από τον καθηγητή.

Δημιουργία εφαρμογής για Smartphones με σκοπό την υποστήριξη της εκπαιδευτικής διαδικασίας στην Εκπαίδευση



Firestore Database 3

Στο στιγμιότυπο «Firestore Database 3» παρατηρούμε πώς αποθηκεύουμε ένα notification στην βάση μας.

Δημιουργία εφαρμογής για Smartphones με σκοπό την υποστήριξη της εκπαιδευτικής διαδικασίας στην Εκπαίδευση

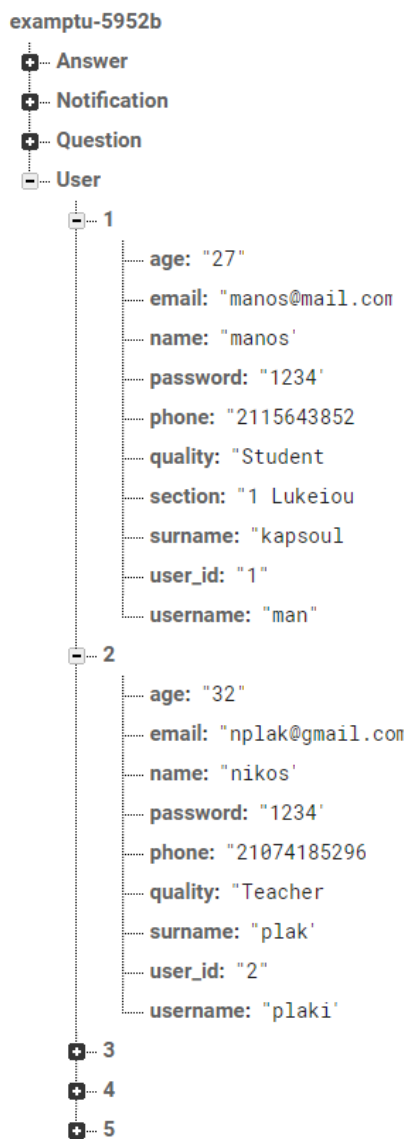
<https://examptu-5952b.firebaseio.com/>



Firestore Database 4

Στο στιγμιότυπο «Firestore Database 4» παρατηρούμε πώς αποθηκεύετε οποιαδήποτε ερώτηση δημιουργήσει ο καθηγητής.

Δημιουργία εφαρμογής για Smartphones με σκοπό την υποστήριξη της εκπαιδευτικής διαδικασίας στην Εκπαίδευση



Firestore Database 5

Στο στιγμιότυπο «Firestore Database 5» παρατηρούμε πώς αποθηκεύετε οποιωσδήποτε χρήστης δημιουργήσει προσωπικό account. Παρατηρούμε ότι ο χρήστης που είναι μαθητής έχει ένα επιπρόσθετο πεδίο όπου δηλώνει σε ποιά τάξη ανήκει.

Δημιουργία εφαρμογής για Smartphones με σκοπό την υποστήριξη της εκπαιδευτικής διαδικασίας στην Εκπαίδευση

Παρουσίαση και ανάλυση της κλάσης «BackgroundTaskHandler»

Αποφασίσαμε να απεικονίσουμε και να επεξηγήσουμε την παρακάτω κλάση καθώς αποτελεί την βάση της εφαρμογής που έχουμε παρουσιάσει έως τώρα.

```
public class backgroundtaskhandler extends GcmTaskService {

    Calendar myCalendar;
    String NotGlobal, TimestapGlobal, QuestionGlobal;
    Boolean pnt=true, PTR=false;
    final String[] GlobalVar1 = new String[1];
    DatabaseReference ref = FirebaseDatabase.getInstance().getReference();

    public backgroundtaskhandler() {
    }

    @Override
    public int onRunTask(TaskParams taskParams) {

        final String data;
        data = taskParams.getTag();
        final String[] GlobalVar = new String[1];
        final SimpleDateFormat sdf = new SimpleDateFormat( pattern: "HH:mm a-dd/MM/yy");
        final String[] test1 = new String[1];
        final String[] test2 = new String[1];
        ArrayList arrayNotification=new ArrayList();

        boolean mHandler = new Handler(Looper.getMainLooper()).post(new Runnable() {
            @Override
            public void run() {
                CountdownTimer cdt5 = new CountdownTimer( millisInFuture: 1, countdownInterval: 1) {
                    @Override
                    public void onTick(long millisUntilFinished) {
                    }
                }
            }
        });
    }
}
```

BackgroundTaskHandler 1

Η κλάση BackgroundTaskHandler όπως παρατηρούμε κάνει override την μέθοδο OnFinish() που κληρονομεί από την κλάση GsmTaskManager.

Gsm Task Service is an application service capable of running and scheduling tasks asynchronously. The service instance returned will not invoke the runnable task in the current thread .In addition to allowing pooling, this method will ensure that the threads executing the tasks will have the thread local variables copied from the calling thread.

```

@Override
public void onFinish() {
    if (String.valueOf(data).contains("Teacher"))
    {
        Query query = FirebaseDatabase.getInstance().getReference("Notification").orderByChild("teacher")
        .equalTo(String.valueOf(String.valueOf(data.replace(target: "Teacher", replacement: "").trim())));
        query.addListenerForSingleValueEvent(valueEventListener2);
    }
    if (String.valueOf(data).contains("Student"))
    {
        DatabaseReference rootRef = FirebaseDatabase.getInstance().getReference();
        DatabaseReference ref = rootRef.child("User");
        ValueEventListener eventListener = new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {
                String Name, Surname, UsernameDB;
                for (DataSnapshot ds : dataSnapshot.getChildren()) {
                    UsernameDB = ds.child("username").getValue(String.class);
                    if (String.valueOf(UsernameDB)
                        .compareTo(String.valueOf(String.valueOf(data.replace(target: "Student", replacement: "").trim())) == 0) {
                        Name = ds.child("name").getValue(String.class);
                        Surname = ds.child("surname").getValue(String.class);
                        GlobalVar1[0] = String.valueOf(Name + " " + Surname);
                    } else {
                    }
                }
            }
        };
        @Override
        public void onCancelled(DatabaseError databaseError) {
        }
    };
    ref.addListenerForSingleValueEvent(eventListener);
    Query query = FirebaseDatabase.getInstance().getReference("Notification").orderByChild("timestamp");
    query.addListenerForSingleValueEvent(fereAnswerMeStatus0);
}
}
}.start();
}
});
return 0;
}
}

```

Μέσα σε αυτήν την μέθοδο θα υλοποιηθεί η υφιστάμενη λογική που θέλουμε να προσθέσουμε. Είναι ευδιάκριτο ότι οι πρώτοι έλεγχοι γίνονται ανάλογα με την ιδιότητα του χρήστη. Αυτό γίνεται διότι η λειτουργικότητα του κάθε ρόλου εντάσσεται στην εφαρμογή με διαφορετικά κριτήρια.

Δημιουργία εφαρμογής για Smartphones με σκοπό την υποστήριξη της εκπαιδευτικής διαδικασίας στην Εκπαίδευση

```

int j = 0,k=0;
} ValueEventListener valueEventListener1 = new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {

        if (dataSnapshot.exists())
        {
            for (DataSnapshot ds : dataSnapshot.getChildren()) {

                tableAnswerStudent tableanswerStudent = ds.getValue(tableAnswerStudent.class);
                if (tableanswerStudent.getStatus().compareTo("0") == 0)
                {
                    j++;
                    System.out.println("notification apo background task : "+tableanswerStudent.getNotification());
                    //notification ετοιμο μια χαρα
                    NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(getApplicationContext())
                        .setSmallIcon(R.mipmap.ic_launcher)
                        .setContentTitle("Απάντηση Μαθητή")
                        .setContentText("")
                        .setAutoCancel(true);
                    System.out.println("mphka fora : " +j);
                    Intent[] intents=new Intent[11];
                    intents[j]=new Intent(getApplicationContext(), checkanswerfromteacher.class);
                    intents[j].putExtra( name: "username", String.valueOf(tableanswerStudent.getNotification())); //plhrofori
                    //mono notification id
                    System.out.println("prin pending item fora : "+j);
                    PendingIntent pendingIntent = PendingIntent.getActivity(
                        getApplicationContext(),
                        j,
                        intents[j],
                        PendingIntent.FLAG_UPDATE_CURRENT
                    );
                    mBuilder.setContentIntent(pendingIntent);
                    // Gets an instance of the NotificationManager service
                    System.out.println("prin notification manager fora : "+j);
                    NotificationManager mNotificationManager =
                        (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
                    mNotificationManager.notify(j, mBuilder.build());
                }
            }
        }
    }
}
}
}

```

```

    @Override
    public void onCancelled(DatabaseError databaseError) {
    }
};

ValueEventListener valueEventListener12 = new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        if (dataSnapshot.exists())
        {
            j = 0;
            for (DataSnapshot ds : dataSnapshot.getChildren()) {
                tableNotification tableNotification= ds.getValue(tableNotification.class);
                Query query = FirebaseDatabase.getInstance().getReference("Answer").orderByChild("notification").equalTo(String.valueOf(tableNotification));
                query.addListenerForSingleValueEvent(valueEventListener11);
            }
            j = 0;
        }
    }

    @Override
    public void onCancelled(DatabaseError databaseError) {
    }
};

String getTimestap()
{
    myCalendar = Calendar.getInstance();
    String day, month;
    if (String.valueOf(myCalendar.get(Calendar.MONTH)+1).length() == 1) {
        month = "0" + String.valueOf(myCalendar.get(Calendar.MONTH) + 1);
    } else {
        month = String.valueOf(myCalendar.get(Calendar.MONTH) + 1);
    }
    int mon = Integer.parseInt(month);
    month = String.valueOf(mon);
}
if (String.valueOf(myCalendar.get(Calendar.DAY_OF_MONTH)).length() == 1) {
    day = "0" + String.valueOf(myCalendar.get(Calendar.DAY_OF_MONTH));
} else {
    day = String.valueOf(myCalendar.get(Calendar.DAY_OF_MONTH));
}
String todayDate = day + "/" + month + "/" + String.valueOf(myCalendar.get(Calendar.YEAR)).replace( target: "20", replacement: "");

return String.format(todayDate, "dd/MM/yy");
}
}

```

```

3 ValueEventListener fereAnswerMeStatus0 = new ValueEventListener() {
    @Override
3 public void onDataChange(DataSnapshot dataSnapshot) {
    String values = String.valueOf(dataSnapshot.getValue());
    System.out.println("backgroundtaskhandler : ValueEventListener fereAnswerMeStatus0 "+ values +"   www.wwwwwwwwwwwwwwwwww");

    if (dataSnapshot.exists())
    {
        for (DataSnapshot dt : dataSnapshot.getChildren()){
            tableNotification obj = dt.getValue(tableNotification.class);
            if (obj.getTimestap().contains(getTimestap()) && obj.getStudent().equals(GlobalVar1[0])){
                System.out.println("to epiasa!!!!!!");

                System.out.println("backgroundtaskhandler : steile mou notification me k = "+k+" gia to timestap = "+obj.getTimestap()+" tou mathhth
                k=0;
                final tableNotification tabli = obj;
                System.out.println("backgroundtaskhandler : tableNotification = "+tabli.getTimestap() );
                System.out.println(GlobalVar1[0]);
                System.out.println(tabli.getNotification_id()+" "+tabli.getTimestap()+" "+tabli.getStudent());
                NotGlobal = tabli.getNotification_id();
                QuestionGlobal = tabli.getQuestion();
                DatabaseReference rootRef = FirebaseDatabase.getInstance().getReference();
                DatabaseReference ref = rootRef.child("Answer");
3 ValueEventListener eventListener = new ValueEventListener() {
                @Override
3 public void onDataChange(DataSnapshot dataSnapshot) {
                System.out.println("notification_id apo ton pinaka Notification : "+tabli.getNotification_id());
                Boolean pnt = false;
                String test = String.valueOf(dataSnapshot.getValue());
                System.out.println(test);
                for (DataSnapshot ds : dataSnapshot.getChildren()) {
                    tableAnswerStudent check = ds.getValue(tableAnswerStudent.class);
                    if (check.getNotification().equals(tabli.getNotification_id())) {
                        System.out.println(check.getNotification()+" "+check.getStudent()+"MESA STO IF");
                        pnt=true;
                    }
                }
            }
        }
    }
}

```

```

    }
    if (pnt==false)
    {
        k++;
        System.out.println("steile notification sto "+tabli.getNotification_id()+"kai erwthsh"+ tabli.getQuestion());
        NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(getApplicationContext())
            .setSmallIcon(R.mipmap.ic_launcher)
            .setContentTitle("Exeis Diabasma neare!!!")
            .setContentText("")
            .setAutoCancel(true);
        //Toast.makeText(getApplicationContext(), String.valueOf(QuestionGlobal), Toast.LENGTH_SHORT).show();
        Intent[] intents=new Intent[11];
        intents[k]=new Intent(getApplicationContext(), centerstudent.class);
        intents[k].putExtra( name: "username", value: "2/" + tabli.getNotification_id() + "/" + tabli.getQuestion()); //p
        PendingIntent pendingIntent = PendingIntent.getActivity(
            getApplicationContext(),
            k,
            intents[k],
            PendingIntent.FLAG_UPDATE_CURRENT
        );
        mBuilder.setContentIntent(pendingIntent);
        NotificationManager mNotificationManager = (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
        mNotificationManager.notify(k, mBuilder.build());
    }
    }

    @Override
    public void onCancelled(DatabaseError databaseError) {
    }

};
ref.addListenerForSingleValueEvent(eventListener);
}
}
k=0;
}

System.out.println("backgroundtaskhandler : Den mou fernei ta data ");
}

@Override
public void onCancelled(DatabaseError databaseError) {
}

};
};

ValueEventListener answerVL = new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        if (dataSnapshot.exists())
        {
            for (DataSnapshot ds : dataSnapshot.getChildren()) {
                tableAnswerStudent check = ds.getValue(tableAnswerStudent.class);
                System.out.println("Apo pinaka Answer notification_id :"+check.getNotification() + " apantsh mathhth" +
                    check.getStudent() + " an exei elejei o kathhgths" + check.getStatus() + " kai o kwdikos ths erwthshs" + check.getAr
            }
        }
        else {
            System.out.println("Den efere tipota");
        }
    }

    @Override
    public void onCancelled(DatabaseError databaseError) {
    }
};
};
```

Η περιοδική κλήση που έχουμε ορίσει να κάνουμε στην βάση διαφοροποιείται ανάλογα με τον ρόλο που έχει ο χρήστης στην εφαρμογή. Ας αναλύσουμε εκτενέστερα τα δύο workflow που έχουμε δημιουργήσει στην εφαρμογή μας.

Στον πρώτο έλεγχο εξετάζουμε κατα πόσο η πληροφορία που έχει έρθει στο παρασκήνιο ανταποκρίνεται στο ρόλο του καθηγητή, εφόσον λοιπόν ο χρήστης είναι ο καθηγητής θα ψάξουμε στον πίνακα Notification να βρούμε ποια από αυτά απευθύνονται στον συγκεκριμένο με βάση το username, εφόσον βρεθούν ειδοποιήσεις για αυτόν θα κοιτάξουμε στον πίνακα Answer προκειμένου να ελέγξουμε αν υπάρχει εγγραφή. Εάν δεν υπάρχει εγγραφή τότε ο μαθητής δεν έχει απαντήσει ή ο καθηγητής δεν έχει στείλει ερωτήσεις αυτός στους μαθητές του. Εάν υπάρχει εγγραφή τότε ελέγχουμε αν το status της εγγραφής είναι 0 τότε και μόνο τότε έχει απαντήσει ο μαθητής και το status της ερώτησης είναι ανοικτό, δηλαδή σε αυτήν την περίπτωση ο καθηγητής θα λάβει notification προκειμένου να διορθώσει την απάντηση που του έδωσε ο εκάστοτε μαθητής του.

Στην περίπτωση που στον πρώτο έλεγχο η πληροφορία που φτάνει στο παρασκήνιο έχει να κάνει με τον ρόλο του μαθητή τότε θα ψάξουμε στον πίνακα User προκειμένου να πάρουμε υφιστάμενη πληροφορία, στην συνέχεια θα αναζητήσουμε στον πίνακα Notification με βάση το timestamp και εφόσον βρεθεί ειδοποίηση που αφορά τον συγκεκριμένο μαθητή την συγκεκριμένη ημερομηνία συνεχίζουμε και αναζητούμε να βρούμε εάν υπάρχει εγγραφή στον πίνακα Answer με τον συγκεκριμένο κωδικό του ήδη υπάρχοντος notification. Εάν υπάρχει τότε ο μαθητής έχει απαντήσει την ερώτηση που του έχει βάλει ο καθηγητής του και προφανώς δεν θα λάβει κάποιο notification, αντίθετα εάν δεν υπάρχει η εγγραφή στον πίνακα τότε ο μαθητής θα πρέπει να ειδοποιηθεί και να λάβει την χρονο-προγραμματισμένη ερώτηση που του έχει στείλει ο καθηγητής του και προφανώς θα πρέπει να απαντήσει. Όταν ο μαθητής απαντήσει και εν συνεχεία ο καθηγητής βαθμολογήσει, δηλαδή αλλάξει το status της εγγραφής του πίνακα Answer σε 1 (σωστή απάντηση) ή 2 (λανθασμένη απάντηση) τότε και μόνο τότε έχει ολοκληρωθεί η διεπαφή των δυο ρόλων.