



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών
«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Δυναμικό Σύστημα Απαντήσεων για Κινητά Dynamic Mobile Student Response System
Όνοματεπώνυμο Φοιτητή	Παντελεήμων Κρασαδάκης
Πατρώνυμο	Σταύρος
Αριθμός Μητρώου	ΜΠΣΠ17034
Επιβλέπων	Ευάγγελος Σακκόπουλος, Επίκουρος Καθηγητής

Ημερομηνία Παράδοσης **Ιούνιος 2019**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Ευάγγελος Σακκόπουλος
Επίκουρος Καθηγητής

Μαρία Βίρβου
Καθηγήτρια

Ευθύμιος Αλέπης
Επίκουρος Καθηγητής

Περίληψη

Σε αυτή την πτυχιακή, παρουσιάζεται ένα Σύστημα Αλληλεπίδρασης Χρηστών (Audience Response System) για κινητά τηλέφωνα, με σκοπό τη βελτίωση της διδακτικής διαδικασίας για τους φοιτητές αλλά και για τους καθηγητές. Μια εφαρμογή για κινητά σχεδιάστηκε και υλοποιήθηκε, την οποία μπορούν να κατεβάσουν οι φοιτητές και να χρησιμοποιήσουν από το κινητό τους για να απαντήσουν ερωτήσεις πολλαπλής επιλογής που θα δοθούν από τον καθηγητή κατά τη διάρκεια του μαθήματος. Οι φοιτητές απαντούν τις ερωτήσεις αυτές από το κινητό τους σε πραγματικό χρόνο και οι καθηγητές παίρνουν τα αποτελέσματα των απαντήσεων και βλέπουν την κατανομή αυτών σε ένα γράφημα. Το σύστημα αυτό ενθαρρύνει τη χρήση του Μοντέλου Ενεργής Μάθησης, αυξάνοντας τη συμμετοχή των φοιτητών κατά τη διάρκεια του μαθήματος και παρέχοντας παράλληλα στους καθηγητές άμεσα δεδομένα ανάδρασης.

Λέξεις κλειδιά – Ηλεκτρονική Μάθηση , Σύστημα Αλληλεπίδρασης Χρηστών, React-Native

Abstract

In this paper, we present a mobile Audience Response System (ARS) in order to improve the teaching process for both students and teachers. A mobile app was designed and created that students can download and use on their phones to answer Multiple Choice Questions (MCQs) given by the teacher during lecture. The students answer on their phones in real time and the teacher can then get the results and see a graph distribution of their answers. This is a step towards encouraging the Active Learning Model and increasing student participation during class while also providing teachers with immediate feedback.

Keywords — Audience Response System, Mobile Learning, React-Native

Περιεχόμενα

1	Εισαγωγή	8
2	Σχετικά Έργα	9
3	Σχεδιασμός	14
3.1	Πρώτο Στάδιο Σχεδίασης	14
3.2	Δεύτερο Στάδιο Σχεδίασης	16
4	Αρχιτεκτονική	31
4.1	Γενικά	31
4.2	Η μεριά του Εξυπηρετητή/Καθηγητών	32
4.3	Η μεριά του Πελάτη/Φοιτητών	41
5	Υλοποίηση	57
5.1	React-Native	57
5.2	Laravel	61
6	Αποτίμηση	65
6.1	Πειραματικά Δεδομένα	65
6.2	Συμπεράσματα	70
7	Επίλογος	71
8	Βιβλιογραφία	72

Πίνακας Εικόνων

Εικόνα 1 Clickers [22]	9
Εικόνα 2 Ποσοστά Παρακολούθησης [12]	10
Εικόνα 3 Γενική Συμβολή των Clickers.....	10
Εικόνα 4 Αρχιτεκτονική Survnote.....	10
Εικόνα 5 Ενθάρρυνση Φοιτητών για Συμμετοχή στο μάθημα	10
Εικόνα 6 Σύστημα MyVote	11
Εικόνα 7 Σύγχρονα Μοντέλα Εκπαίδευσης	12
Εικόνα 8 Αρχιτεκτονική QTvity.....	12
Εικόνα 9 Έρευνα SSCLS.....	12
Εικόνα 10 Αρχιτεκτονική Συστήματος.....	13
Εικόνα 11 Οθόνη Εφαρμογής	13
Εικόνα 12 Διάγραμμα Χρήσης 1.....	15
Εικόνα 13 Διάγραμμα Ακολουθίας 1	15
Εικόνα 14 Σελίδα Εγγραφής.....	16
Εικόνα 15 Σελίδα με Εγγεγραμμένες Τάξεις.....	17
Εικόνα 16 Σελίδα Προσθήκης Μαθήματος.....	18
Εικόνα 17 Σελίδα Λίστας Ερωτηματολογίων	19
Εικόνα 18 Σελίδα Προσθήκης Ερωτηματολογίου.....	19
Εικόνα 19 Σελίδα Ενεργοποίησης Ερώτησης.....	20
Εικόνα 20 Σελίδα Εμφάνισης Απαντήσεων	21
Εικόνα 21 Δραστηριότητα Εγγραφής Χρήστη	22
Εικόνα 22 Δραστηριότητα Επιλογής Μαθήματος.....	22
Εικόνα 23 Δραστηριότητα Εγγραφής σε Μαθήματα.....	23
Εικόνα 24 Δραστηριότητα Εύρεσης Ερωτηματολογίου	23
Εικόνα 25 Δραστηριότητα Παλιών Ερωτηματολογίων.....	24
Εικόνα 26 Δραστηριότητα Εμφάνισης Παλιών Απαντήσεων	24
Εικόνα 27 Δραστηριότητα Ερώτησης	25
Εικόνα 28 Δραστηριότητα Τέλους Χρόνου	25
Εικόνα 29 Δραστηριότητα Εμφάνισης Απάντησης	26
Εικόνα 30 Διάγραμμα Χρήσης Καθηγητή	27
Εικόνα 31 Διάγραμμα Χρήσης Φοιτητή.....	28
Εικόνα 32 Διάγραμμα Ακολουθίας Καθηγητή.....	29
Εικόνα 33 Διάγραμμα Ακολουθίας Φοιτητή	30
Εικόνα 34 Γενική Αρχιτεκτονική Συστήματος.....	32
Εικόνα 35 Οθόνη Εγγραφής.....	33
Εικόνα 36 Οθόνη Σύνδεσης.....	33
Εικόνα 37 Οθόνη Λίστας Μαθημάτων.....	34
Εικόνα 38 Οθόνη Μαθημάτων Άλλων Καθηγητών.....	35
Εικόνα 39 Οθόνη Εισαγωγής Μαθήματος	35
Εικόνα 40 Οθόνη Λίστας Ερωτηματολογίων	36
Εικόνα 41 Οθόνη Προσθήκης Ερωτηματολογίου.....	37
Εικόνα 42 Οθόνη Επεξεργασίας Ερωτηματολογίου.....	37
Εικόνα 43 Οθόνη Επιβεβαίωσης Διαγραφής	38

Εικόνα 44 Οθόνη Ερωτηματολογίου.....	39
Εικόνα 45 Οθόνη Απαντήσεων Α.....	40
Εικόνα 46 Οθόνη Απαντήσεων Β.....	40
Εικόνα 47 Εφαρμογή Expo στο e-Store.....	41
Εικόνα 48 Genymotion.....	42
Εικόνα 49 Expo Builder.....	42
Εικόνα 50 Οθόνη Σύνδεσης.....	43
Εικόνα 51 Κεντρική Δραστηριότητα.....	44
Εικόνα 52 Λίστα Επιλεγμένων Μαθημάτων.....	45
Εικόνα 53 Λίστα Μαθημάτων για Εγγραφή.....	46
Εικόνα 54 Επιλογή Καθηγητή ή Παλιών Ερωτηματολογίων.....	47
Εικόνα 55 Επιλογή Παλιών Ερωτηματολογίων.....	48
Εικόνα 56 Απάντηση Παλιού Ερωτηματολογίου.....	49
Εικόνα 57 Εύρεση Ερωτηματολογίου με βάση το Id.....	50
Εικόνα 58 Ενδεικτικό Ερωτηματολόγιο.....	51
Εικόνα 59 Ερωτηματολόγιο με επιλεγμένη απάντηση Β.....	52
Εικόνα 60 Τελική Οθόνη με Ενεργό Ερωτηματολόγιο.....	53
Εικόνα 61 Τελική Οθόνη με μη Ενεργό Ερωτηματολόγιο.....	54
Εικόνα 61 Οθόνη Ερωτηματολογίου σε iPhone.....	55
Εικόνα 61 Οθόνη Εισαγωγική σε Android.....	56
Εικόνα 61 Οθόνη Ερωτηματολογίου σε Android.....	56
Εικόνα 62 Μετατροπή JSX σε JavaScript.....	57
Εικόνα 63 Βασική Δομή Κώδικα.....	58
Εικόνα 64 StyleSheet.....	58
Εικόνα 65 Class FindPoll.....	59
Εικόνα 66 Συνάρτηση buttonPressed.....	60
Εικόνα 67 Δομή MVC της Laravel.....	61
Εικόνα 68 Μοντέλο του Poll.....	62
Εικόνα 69 API Routing.....	62
Εικόνα 70 Διαχείριση Απαντήσεων φοιτητών από StudentController.....	63
Εικόνα 71 Συλλογή Απαντήσεων φοιτητών από PollController.....	64
Εικόνα 72 Οδηγίες Εγκατάστασης/Χρήσης.....	66

1 Εισαγωγή

Την τελευταία δεκαπενταετία έχει παρατηρηθεί παγκοσμίως μια τεράστια αύξηση του αριθμού των χρηστών που χρησιμοποιούν το Διαδίκτυο. Η αύξηση αυτή μαζί με την ανάπτυξη και εξέλιξη των κινητών τηλεφώνων έχει ως αποτέλεσμα μεγάλος αριθμός χρηστών να έχουν εύκολη και συνεχή πρόσβαση στο διαδίκτυο από το κινητό τους. Έρευνες δείχνουν ότι πλέον πιο πολλοί χρήστες έχουν smart phones (έξυπνα κινητά) από υπολογιστές και ειδικά στις ηλικίες 18-29 (94% έχουν smart phones) [34].

Γίνεται λοιπόν αντιληπτό ότι οι φοιτητές ενός πανεπιστημιακού ιδρύματος ανήκουν σε αυτή την κατηγορία και μάλιστα είναι εξοικειωμένοι με τη χρήση αυτών των συσκευών. Παράλληλα βασική ανησυχία για όλα τα εκπαιδευτικά ιδρύματα αποτελεί η βελτίωση της εκπαιδευτικής διαδικασίας και της επικοινωνίας μεταξύ φοιτητών και καθηγητών. Αυτό φαίνεται από τις διάφορες έρευνες που έχουν γίνει, που ψάχνουν και αναπτύσσουν νέες μεθόδους [1] [2] και συστήματα [3][5][14] για να βελτιώσουν τις αλληλεπιδράσεις κατά τη διάρκεια του μαθήματος. Ειδικά τα τελευταία 3-4 χρόνια έχει επικεντρωθεί η έρευνα στη αξιοποίηση των κινητών τηλεφώνων μέσα στο μάθημα [9][11][13].

Το πρόβλημα λοιπόν που καλείται να αντιμετωπίσει αυτή η πτυχιακή είναι η ανάπτυξη μια βελτιωμένης εφαρμογής που θα αξιοποιεί τη σύγχρονη τεχνολογία, συγκεκριμένα τα κινητά τηλέφωνα και θα δίνει τη δυνατότητα στους χρήστες-μαθητές να συμμετέχουν πιο ενεργά στο μάθημα, βελτιώνοντας άμεσα την εκπαίδευσή τους [4]. Παράλληλα, οι χρήστες-καθηγητές θα πρέπει να λαμβάνουν άμεσα τα αποτελέσματα και τις απαντήσεις των φοιτητών τους, ώστε να προσαρμόζουν το μάθημα στις ανάγκες της κάθε τάξης. Προτεραιότητα κατά τη σχεδίαση της εφαρμογής ήταν η ευχρηστία και αμεσότητα της εφαρμογής για τους φοιτητές.

2 Σχετικά Έργα

Την τελευταία δεκαετία έχουν γίνει πολλές έρευνες και προσπάθειες για την ανάλυση και δημιουργία Συστημάτων Αλληλεπίδρασης Χρηστών (Audience Response Systems) που να εφαρμόζονται σε τάξεις. Οι αρχικές μελέτες και υλοποιήσεις έγιναν με την εισαγωγή των "clickers" σε τάξεις [12][22][24]. Οι clickers ήταν ένα ειδικά κατασκευασμένο hardware, μικροί φυσικού πομποί, με κάποιο συγκεκριμένο αριθμό κουμπιών πάνω, με το πάτημα των οποίων μεταδιδόταν η απάντηση που αντιστοιχούσε στο κουμπί. Αυτοί οι clickers συνήθως επικοινωνούσαν με το σύστημα μέσω υπέρυθρων ή σήματα Ραδιοφωνικής Συχνότητας (RF- Radio Frequency Signals) (Εικόνα 1). Αυτή η τεχνολογία όμως είχε πολλά μειονεκτήματα. Απαιτούσε ειδικό hardware να έχει αγοραστεί από το πανεπιστήμιο και μάλιστα σε πλήθος, τουλάχιστον ίσο με τον μέγιστο αριθμό φοιτητών ανά τάξη. Το ίδιο το hardware ήταν συχνά ελαττωματικό και σε περίπτωση βλάβης έπρεπε να αντικατασταθεί άμεσα διαφορετικά ο φοιτητής δεν μπορούσε να συμμετέχει στη διαδικασία. Επίσης, για την επικοινωνία με το σύστημα και τη συλλογή των αποτελεσμάτων απαιτούνταν συχνά ειδικό λογισμικό, που θα χρειαζόταν να αγοράσει ή να φτιάξει το πανεπιστήμιο. Παρόλα αυτά, η εισαγωγή τους στις τάξεις, αύξησε αισθητά τόσο την συχνότητα παρακολούθησης των φοιτητών(Εικόνα 2), όσο και την ποιότητα αυτής(Εικόνα 3).

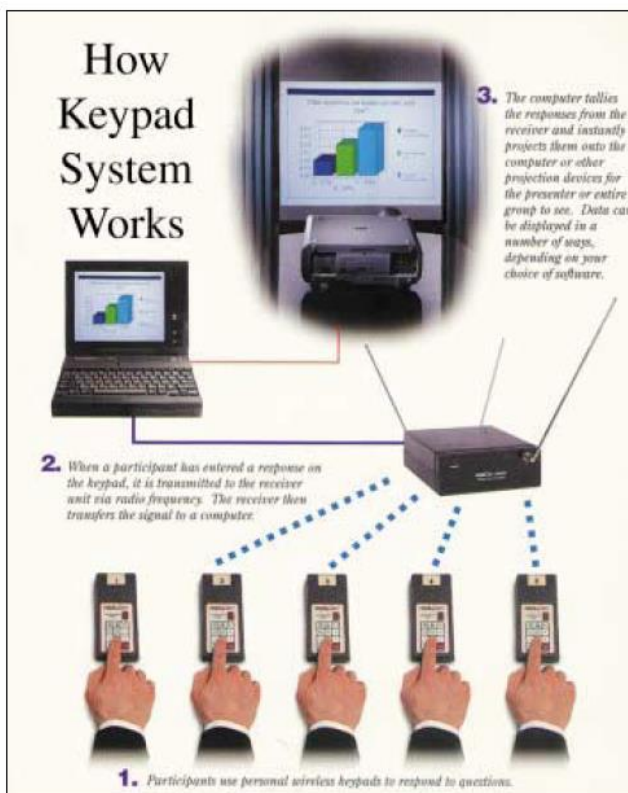


Fig. 1. Illustration of operation of wireless keypad system. (Photo: Fleetwood Group, Inc.)

lution to the problem of the passive large lecture proven to be successful, flexible, and cost-effective for a wide variety of students.

Our Keypad System

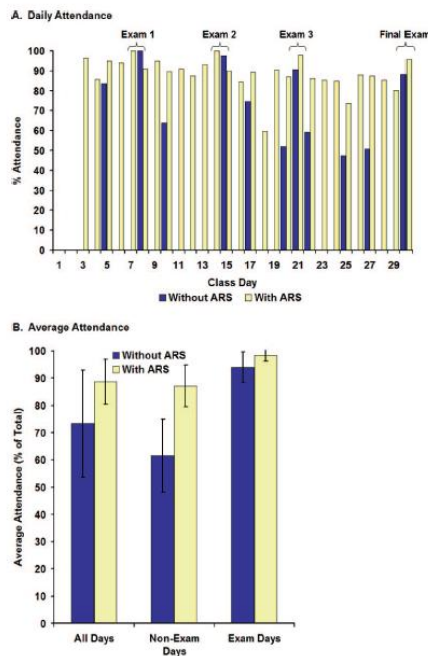
The procedure we initiated at the Illinois Institute of Technology (IIT) five years ago to enhance the participation of students in the lecture was by means of an *interactive student response system*. We used a commercial, two-way wireless (RF) keypad that the student can hold in hand and enter responses upon request during the lecture.¹² The results can be saved and displayed to the instructor or class or both within seconds. Figure 1 illustrates the keypad system and its operation.

We have made use of the wireless keypads in our elementary physics classes since the spring 1995 semester. With an average of 55 in each class, students were assigned a numbered keypad for the course.¹⁶ Before each class, they exchanged their ID card for a keypad.

Questions that could be answered by "yes"/"no" or multiple choice (1-10) were woven into the lecture and made relevant to what just happened or what was just about to happen. There were also questions to test whether students prepared for class. When appropriate, peer

Εικόνα 1 Clickers [22]

J. E. Caldwell



Εικόνα 2 Ποσοστά Παρακολούθησης [12]

force. The keypad questions were of a wide variety and the questions could accommodate a broad range of instructional philosophies and goals. They allow an instructor to determine the following:

- a) Have the students read the text before class?
- b) Are the students paying attention?
- c) Do the students remember important facts?
- d) Are the students thinking?
- e) Do the students recognize concepts?
- f) Can the student work in a group?
- g) Can the students do numerical exercises?
- h) What do the students say about the pace and the interest of the lecture?

Instructors can evaluate and get feedback on their performance in “real time” using “How am I doing?” questions where students choose between, for example,

- 1) The lecture is very clear so far, no questions.
- 2) I have a question or two.
- 3) I have a lot of questions.
- 4) I am so confused, I don't have any questions.

Student response to the last question enables the teacher to extend, repeat, or modify previous lecture topic discussions.

Εικόνα 3 Γενική Συμβολή των Clickers

Με την αύξηση των δυνατοτήτων που μας προσφέρει η σύγχρονη τεχνολογία και ειδικά η ισχύς των κινητών τηλεφώνων είναι πλέον εφικτό να μετατρέψουμε τα κινητά των φοιτητών σε “clickers”. Βασισμένη σε αυτή την ιδέα έχουν γίνει διάφορες προσπάθειες υλοποίησης. Κάποια από αυτά τα συστήματα έχουν αρχίσει να αναπτύσσονται μόλις από το 2010, όταν μια ερευνητική ομάδα πρότεινε το Survnote [6] και αντίστοιχα το σύστημα που πρότειναν στο [17], τα οποία είναι ένα δωρεάν Συστήματα Αλληλεπίδρασης με Χρήστες βασισμένα στο διαδίκτυο για έρευνες και ψηφοφορίες μέσω διαδικτύου και κινητών τηλεφώνων. Το πρώτο σύστημα το Survnote, είναι μια επέκταση του συστήματος Votapedia [7], το οποίο επιτρέπει την επικοινωνία μέσω μηνυμάτων SMS από κινητά τηλέφωνα ή μέσω υπολογιστών και διαδικτύου και του ειδικού προγράμματος που είχαν φτιάξει για τη ψηφοφορία απαντήσεων (Εικόνα 4). Αντίστοιχη δομή έχει και το project BeA και συνέβαλλαν στη μελέτη τέτοιων συστημάτων (Εικόνα 5).

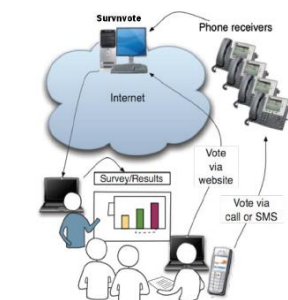


Fig.1 : Survnote General Architecture

Εικόνα 4 Αρχιτεκτονική Survnote

physical device and a specific user is established. In both cases, the system provides anonymity to students. Students increase their participation by not publicly exposed themselves to their peers at the time to respond. According to these papers, students attend more to class and the alertness is higher than in conventional classes. In addition, students can compare their answers with their peers. Teachers can test students' knowledge in class and make them talk in search of the solution or explain why they chose one option or another. This interaction brings an improvement of the quality of the students' learning. It also presents a greater commitment of students, improve their attention and improve learning achievements.

It is common for conventional voting systems to be independent of the question. The task of displaying the question and extract the data often falls on teachers. Teacher has to include her/his questions within the presentation to show the text of the question and options to the students. The remote controls do not have the ability to display all this information to students.

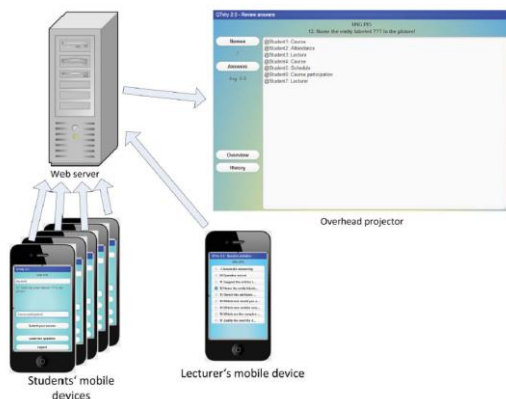
Εικόνα 5 Ενθάρρυνση Φοιτητών για Συμμετοχή στο μάθημα

Table 1: Categorisation of mobile learning applications
(adapted from (Naismith, Lonsdale, Vavoula, & Sharples, 2004))

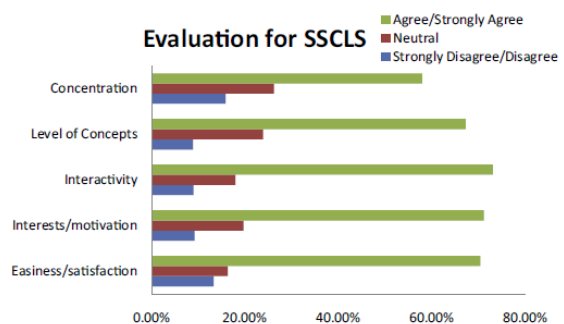
Learning Theory and Key Theorists	Descriptions	Activities
Behaviorist learning (Pavlov, 1927; Pavlov & Anrep, 2003; Skinner, 1968)	Activities promoting learning as a change in learners' observable actions.	<ul style="list-style-type: none"> • Drill and feedback • Classroom response systems (clickers)
Constructivist learning (Bruner, 1966; Papert, 1980; Piaget, 1929)	Activities in which learners actively construct new ideas or concepts based on both their previous and current knowledge.	<ul style="list-style-type: none"> • Participatory simulations
Situated learning (Brown, Collins, & Duguid, 1989; Lave & Wenger, 1991)	Activities promoting learning within an authentic context and culture.	<ul style="list-style-type: none"> • Problem and case-based learning • Context awareness
Collaborative learning (Vygotsky, 1978)	Activities promoting learning through social interaction.	<ul style="list-style-type: none"> • Mobile computer-support collaborative learning (mCSCCL)
Informal and lifelong learning (Eraut, 2000)	Activities supporting learning outside a dedicated learning environment and formal curriculum.	<ul style="list-style-type: none"> • Supporting intentional and accidental learning episodes
Learning and teaching support (No key theorists)	Activities assisting the coordination of learners and resources for learning activities.	<ul style="list-style-type: none"> • Personal organization • Support for administrative duties (e.g., attendance)

Εικόνα 7 Σύγχρονα Μοντέλα Εκπαίδευσης

Πάνω σε τέτοια συστήματα, έχουν βασιστεί και οι πιο σύγχρονες έρευνες, όπως το QTVity[9] και το SSCLS[13]. Το 2015 που υλοποιήθηκαν αυτά τα συστήματα, πλέον η τεχνολογία και οι δυνατότητες των κινητών τηλεφώνων επιτρέπουν την εύκολη χρήση τους για εκπαιδευτικές εφαρμογές. Το SSCLS χρησιμοποιεί μια εφαρμογή που έφτιαξε η ομάδα MyResponse, η οποία όμως έχει αναπτυχθεί σε Android και λειτούργησε και δοκιμάστηκε μόνο σε Android κινητά τηλέφωνα, αποκλείοντας έτσι τους μαθητές που είχαν, άλλα λογισμικά στο κινητό τους. Παρόλα αυτά από την έρευνα που έκαναν φάνηκε να είχε μεγάλη επιτυχία και απήχηση το σύστημα τους τόσο στη χρήση όσο και στη συμβολή κατά τη διάρκεια του μαθήματος.

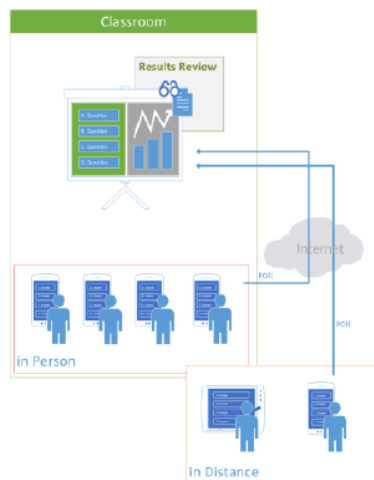


Εικόνα 8 Αρχιτεκτονική QTVity

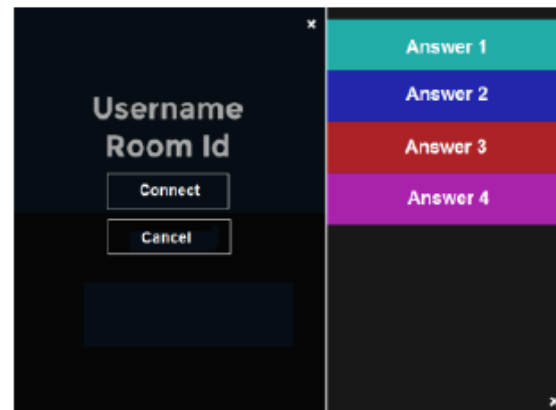


Εικόνα 9 Έρευνα SSCLS

Τέλος θα αναφέρουμε την προσπάθεια μιας ομάδας από το Ελληνικό Ανοικτό Πανεπιστήμιο [10], για τη δημιουργία ενός ARS με τη δυνατότητα να συλλέγει και να παρουσιάζει τις απαντήσεις όλων των συμμετεχόντων μιας τάξης κατά τη διάρκεια μιας διάλεξης και το σύστημα αυτό αποτελεί και τη βάση της παρούσας εργασίας. Το συγκεκριμένο σύστημα για να αντιμετωπίσει το πρόβλημα των διαφορετικών λειτουργικών συστημάτων στα κινητά, χρησιμοποίησε το nunchuck.js [40], που είναι μια βιβλιοθήκη που επιτρέπει τη δημιουργία εφαρμογών σε κινητά βασισμένη στο διαδίκτυο (Web-based Application), πρακτικά μετατρέποντας τον browser σε εφαρμογή κινητού τηλεφώνου και υποστηρίζεται από τους σύγχρονους browsers και άρα μπορεί να τρέξει σε όλα τα κινητά. Η γενική αρχιτεκτονική, όπως φαίνεται στην εικόνα 10, έχει τους χρήστες συνδεδεμένους στο διαδίκτυο είτε από κοντά (μέσα στην τάξη), είτε εξ αποστάσεως και μπορούν να συνδεθούν από τον υπολογιστή τους ή το κινητό τους. Μετά, οι φοιτητές θα βρεθούν στην κεντρική σελίδα της εφαρμογής (Εικόνα 11), η οποία ζητάει το όνομα του χρήστη και τον κωδικό σύνδεσης για την τάξη. Αν αυτά εισαχθούν σωστά, μπορεί να απαντήσει στην ερώτηση που είναι ενεργή και οι απαντήσεις συλλέγονται από το server και παρουσιάζονται μετά σε ένα από γράφημα



Εικόνα 10 Αρχιτεκτονική Συστήματος



Εικόνα 11 Οθόνη Εφαρμογής

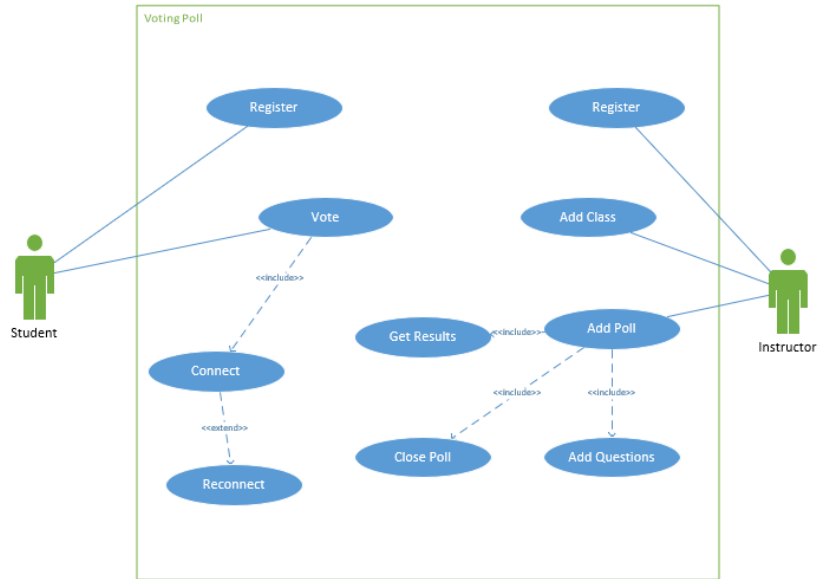
3 Σχεδιασμός

3.1 Πρώτο Στάδιο Σχεδίασης

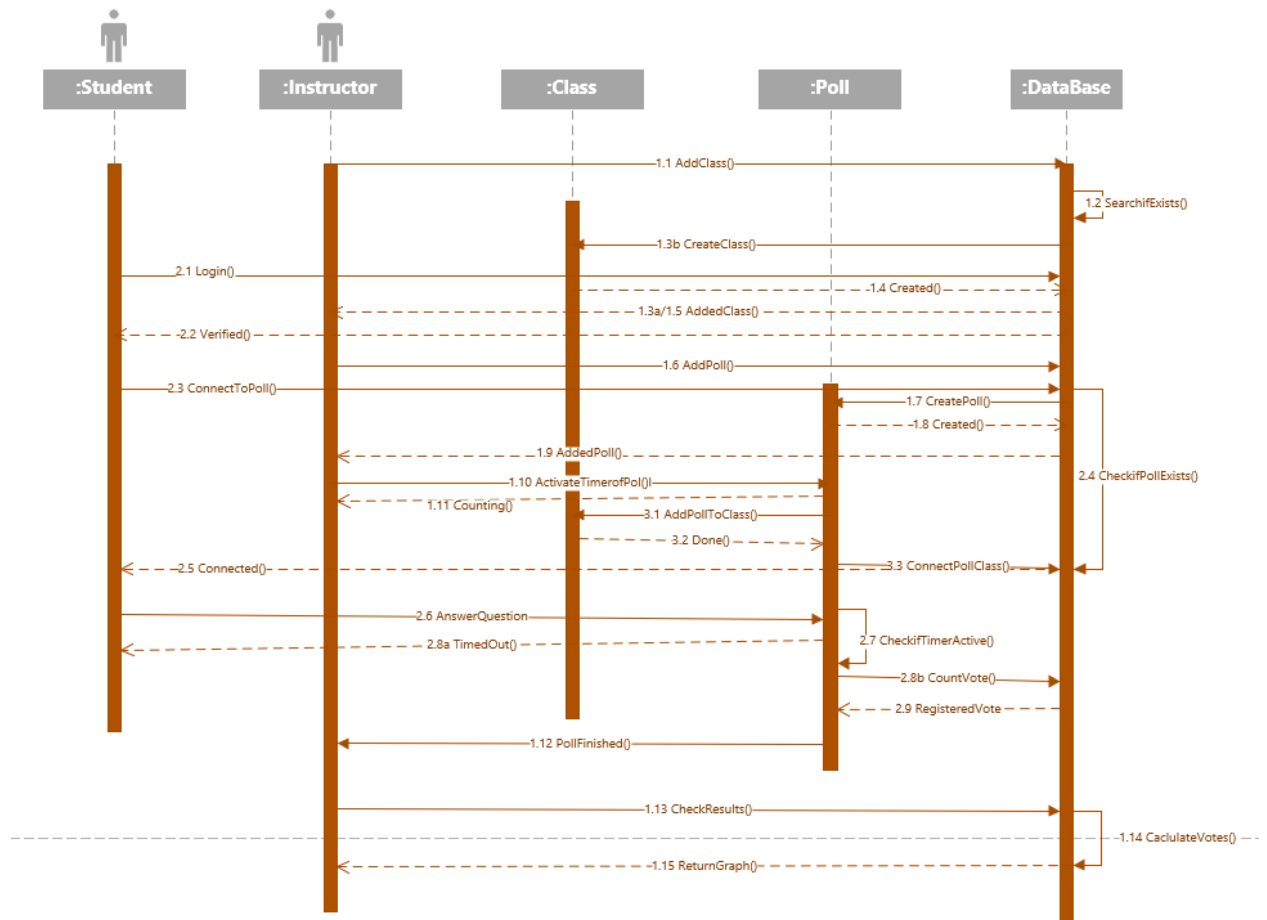
Έχοντας ορίσει λοιπόν το πρόβλημα, αρχίσαμε τη σχεδίαση των δυνατοτήτων και λειτουργιών που θα έχουν οι χρήστες. Αρχικά, οι χρήστες μας χωρίζονται σε 2 ξεχωριστές κατηγορίες, τους φοιτητές (Students) και τους καθηγητές (Instructors). Οι φοιτητές αφού γραφούν στην εφαρμογή μας θα χρειάζεται να συνδεθούν στην ερώτηση και να ψηφίσουν την απάντηση που θεωρούν σωστή. Αντίστοιχα, οι καθηγητές αφού εγγραφούν θα πρέπει να δημιουργήσουν το μάθημά τους. Το μάθημα θα έχει ερωτηματολόγια (Polls) τα οποία έχουν μορφή Πολλαπλής επιλογής, με 4 επιλογές για κάθε ερώτηση από τις οποίες μόνο μία είναι σωστή. Αφού τεθεί η ερώτηση πρέπει να μπορεί να πάρει τις απαντήσεις των φοιτητών. Ακολουθεί η τεκμηρίωση των περιπτώσεων χρήσης και τα διαγράμματα Use Case (Χρήσης) και Sequence (Ακολουθίας) UML, στις εικόνες 1 και 2 αντίστοιχα [27][28].

Τεκμηρίωση Περιπτώσεων Χρήσης:

- a) Σύντομη Περιγραφή: Προσθήκη ερωτηματολογίων στο σύστημα από καθηγητές και εύρεση και απάντηση αυτών των ερωτηματολογίων από φοιτητές.
- b) Προ-συνθήκες: Οι φοιτητές θα πρέπει να έχουν κατεβάσει την εφαρμογή στο κινητό τους και να έχουν συνδεθεί μέσω του διαδικτύου και αντίστοιχα οι καθηγητές να εγγραφούν στην ιστοσελίδα που θα τους παρέχουμε.
- c) Κύρια Ροή Γεγονότων:
 1. Ο χρήστης συνδέεται με τον κωδικό του
 2. Εμφάνιση στην κεντρική οθόνη (για προσθήκη ή εύρεση ερωτηματολογίου ανάλογα)
 3. Προσθήκη Ερωτηματολογίου από καθηγητή
 4. Απάντηση ερώτησης από φοιτητές
 5. Κλείσιμο Ερωτηματολογίου από καθηγητή
 6. Εμφάνιση Αποτελεσμάτων
- d) Εναλλακτική Ροή Γεγονότων :
 - I. Σε περίπτωση αδυναμίας σύνδεση στο δίκτυο δεν μπορούν να συνδεθούν και να χρησιμοποιήσουν οι χρήστες το σύστημα.
 - II. Σε περίπτωση που ο συνδυασμός email-κωδικός του καθηγητή είναι λάθος, ζητείται να ελέγξει την εισαγωγή των στοιχείων και δεν συνεχίζει.
 - III. Σε περίπτωση που οι φοιτητές προσπαθούν να βρουν ένα ερωτηματολόγιο που δεν υπάρχει, εμφανίζεται κατάλληλο μήνυμα και παραμένουν στην κεντρική οθόνη.
- e) Μετά-συνθήκες: Μετά την επιτυχή ολοκλήρωση της διαδικασίας της ψηφοφορίας, οι ψήφοι θα πρέπει να έχουν καταμετρηθεί και υπολογιστεί σωστά και να παρουσιαστούν στον καθηγητή σε γράφημα.



Εικόνα 12 Διάγραμμα Χρήσης 1

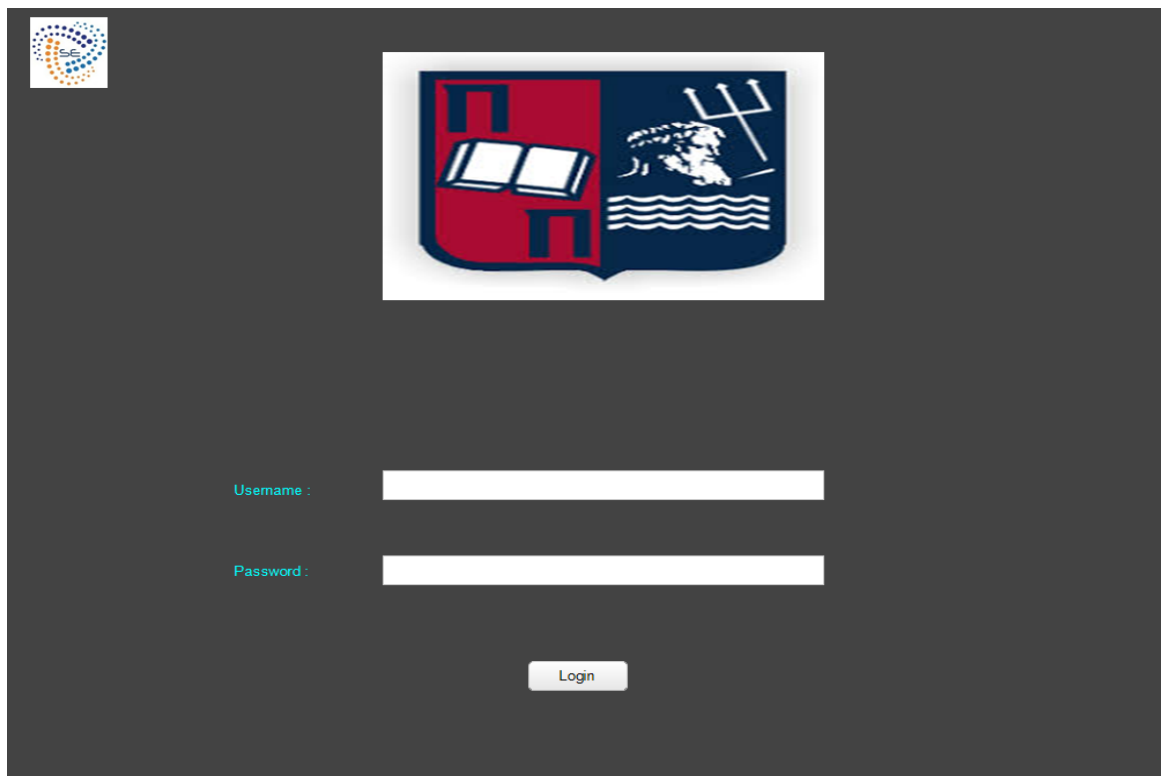


Εικόνα 13 Διάγραμμα Ακολουθίας 1

3.2 Δεύτερο Στάδιο Σχεδίασης

Στη συνέχεια, έγιναν πρωτότυπα σχέδια για τη σελίδα που θα χρησιμοποιούν οι καθηγητές και για την εφαρμογή στο κινητό των φοιτητών. Το εργαλείο που χρησιμοποιήθηκε για τη δημιουργία αυτών των πρωτοτύπων είναι το JustInMind [29].

Πρώτα έχουμε τη σελίδα εγγραφής ή σύνδεσης των χρηστών-καθηγητών. Μέσω αυτής τη σελίδα θα μπορούν οι χρήστες να συνδεθούν στο σύστημά μας μέσω του ονόματος και κωδικού τους για να χρησιμοποιήσουν τις λειτουργίες του (Εικόνα 3). Αφού συνδεθούν οι χρήστες θα οδηγούνται στη κεντρική σελίδα, στην οποία βλέπουν τη λίστα με τα μαθήματα τους, τα οποία χαρακτηρίζονται από το όνομα, το εξάμηνο, την αίθουσα που γίνονται και το μοναδικό Id τους(Εικόνα 4). Πάνω υπάρχει μπάρα πλοήγησης που περιέχει ένα κουμπί με το Logo του εργαστηρίου Τεχνολογίας Λογισμικού του Πανεπιστημίου Πειραιά που λειτουργεί σαν Home button, οδηγεί δηλαδή στην κεντρική σελίδα. Δίπλα στο logo, υπάρχουν μπάρα αναζήτησης για τα μαθήματα ή τα ερωτηματολόγια, σύνδεσμος για τα μαθήματα παλαιότερων ετών και για τα μαθήματα άλλων καθηγητών. Στα δεξιά της σελίδας βλέπει ο διδάσκοντας λίστα με τα τελευταία ερωτηματολόγια ανά μάθημα που έχει κάνει. Οι δύο αυτές μονάδες(μπάρα αναζήτησης και λίστα τελευταίων ερωτήσεων) υπάρχουν σε όλες τις υπόλοιπες σελίδες. Υπάρχει, τέλος, κουμπί για προσθήκη νέου μαθήματος που οδηγεί στην επόμενη σελίδα .



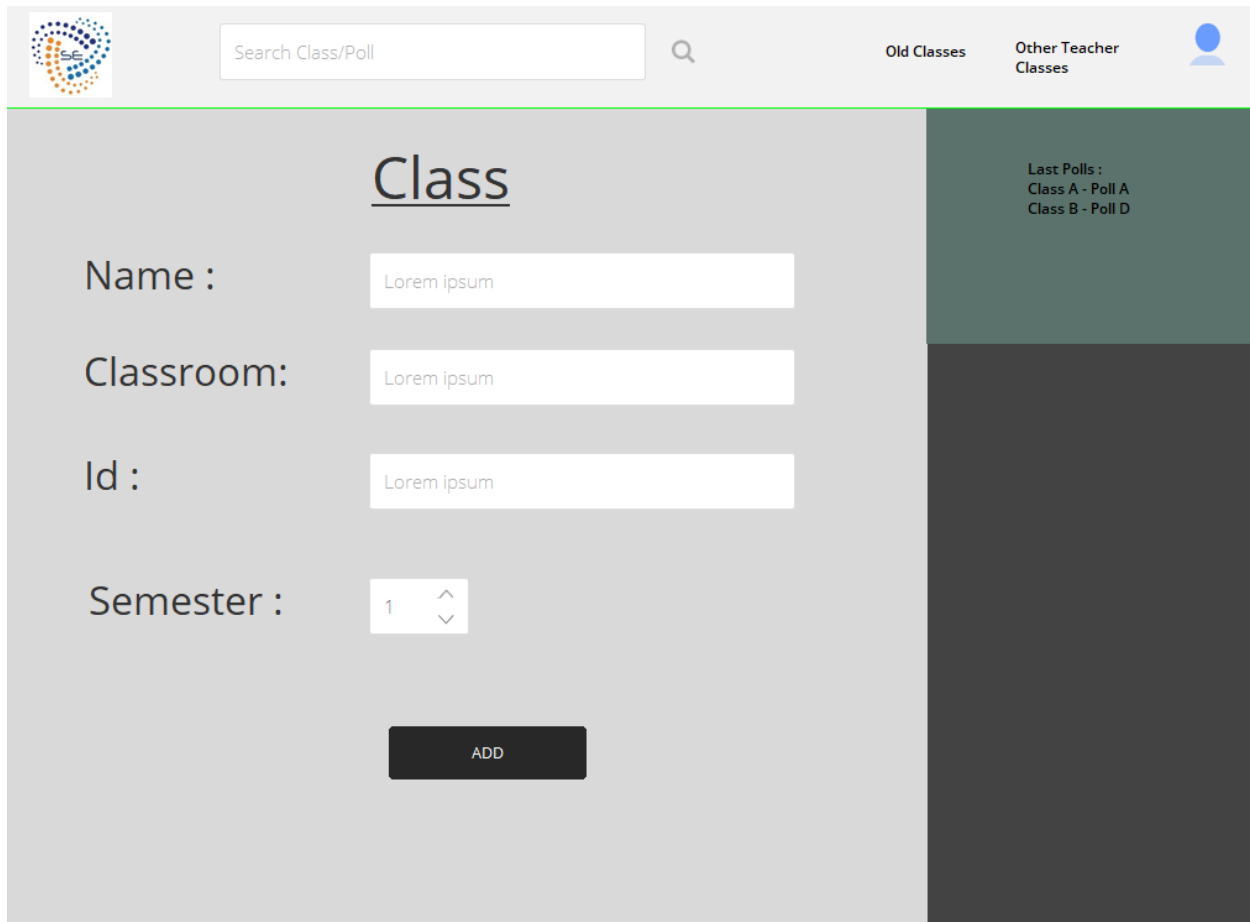
Εικόνα 14 Σελίδα Εγγραφής

The screenshot shows a web application interface for managing classes. At the top, there is a search bar labeled 'Search Class/Poll' and a magnifying glass icon. To the right, there are two tabs: 'Old Classes' and 'Other Teacher Classes', along with a user profile icon. The main content area is titled 'My Classes' and features an 'Add Class' button. Below the title is a table with the following columns: NAME, Semester, ID, and Classroom. The table contains seven rows of class data. The first two rows, 'Class A' and 'Class B', have red 'X' marks in the Classroom column. The right sidebar displays 'Last Polls' for 'Class A - Poll A' and 'Class B - Poll D'.

NAME	Semester	ID	Classroom
Class A	1	1	Lorem ipsum ✖
Class B	2	2	Lorem ipsum ✖
Class C	2	3	Lorem ipsum
Class D	3	4	Lorem ipsum
Class E	3	5	Lorem ipsum
Class F	4	6	Lorem ipsum
Class G	5	7	Lorem ipsum

Εικόνα 15 Σελίδα με Εγγεγραμμένες Τάξεις

Στη σελίδα προσθήκης μαθήματος, ζητείται από τον καθηγητή να προσθέσει το όνομα του μαθήματος, την αίθουσα στην οποία θα γίνεται το Id του και το εξάμηνο. Αφού τα προσθέσει και πατήσει το κουμπί Add(προσθήκη) οδηγείται στην προηγούμενη σελίδα (Εικόνα 5).



Class

Name :

Classroom:

Id :

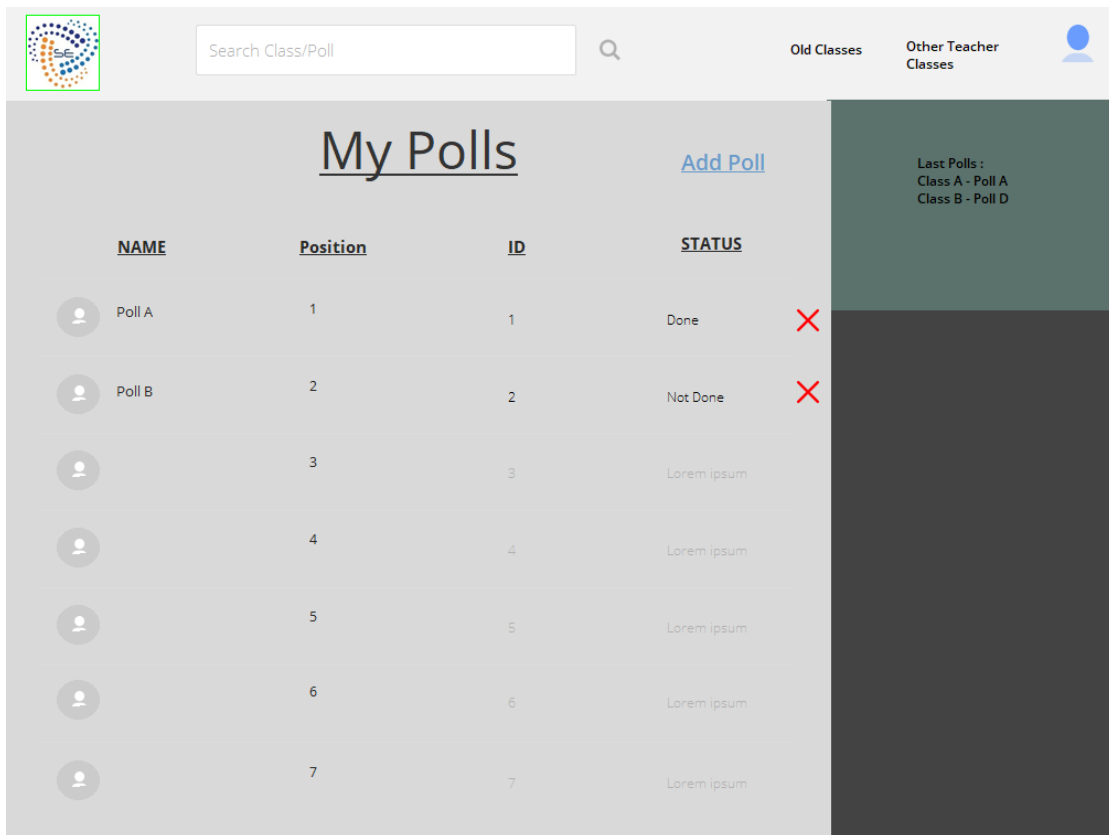
Semester :

ADD

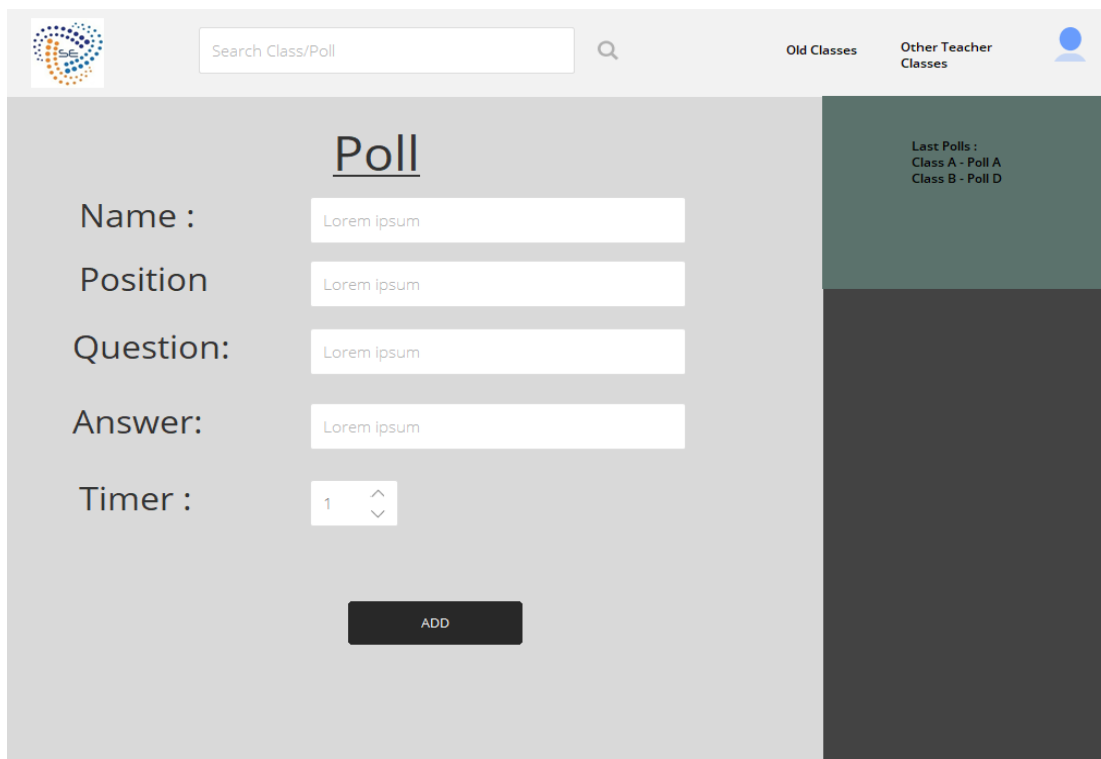
Last Polls :
Class A - Poll A
Class B - Poll D

Εικόνα 16 Σελίδα Προσθήκης Μαθήματος

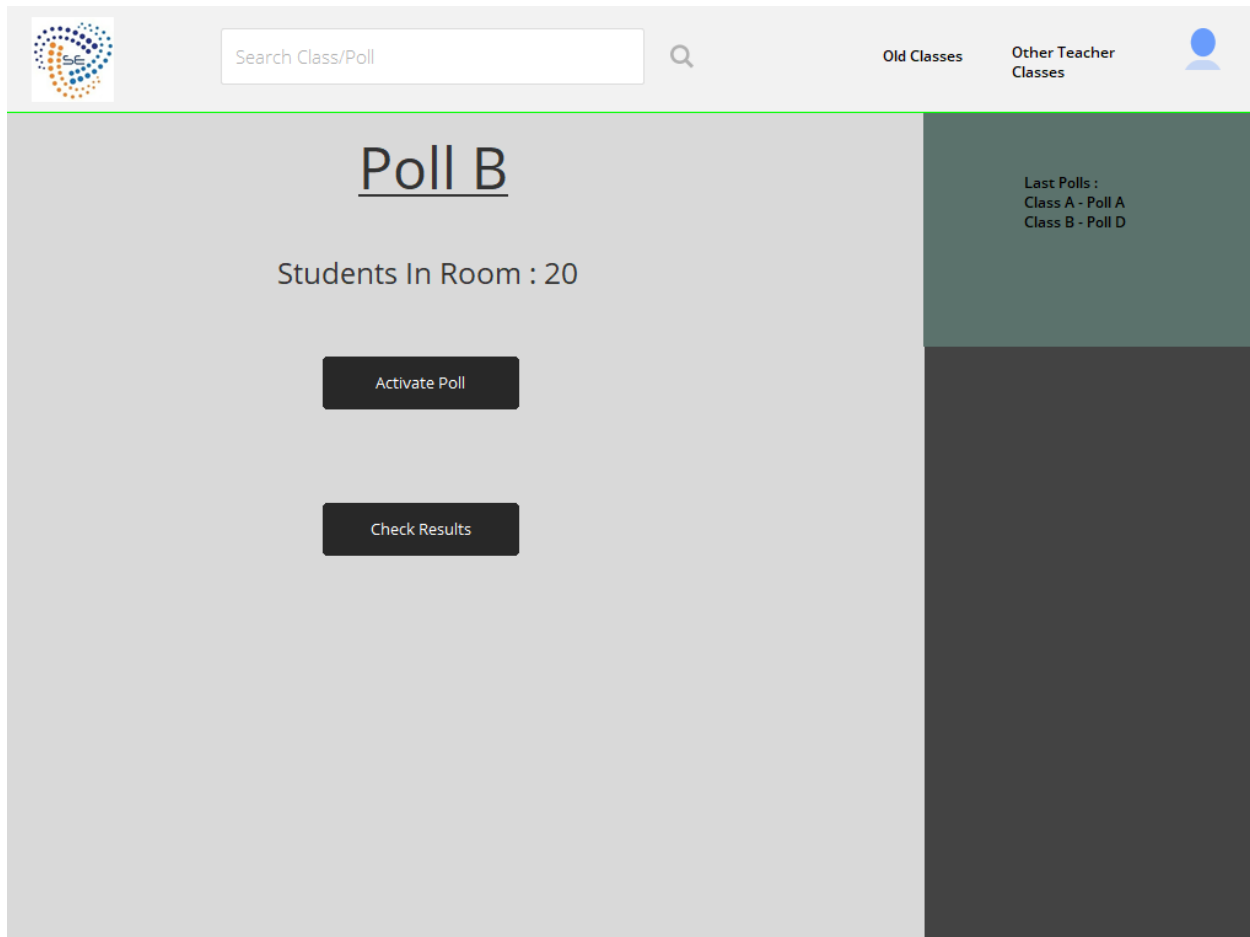
Στη κεντρική σελίδα τώρα αν πατήσει ο χρήστης το όνομα κάποιου μαθήματος θα οδηγηθεί στη σελίδα του μαθήματος. Σε αυτή βλέπει τη λίστα με τα ερωτηματολόγια που υπάρχουν για αυτό το μάθημα. Πιο συγκεκριμένα, βλέπει το όνομα του ερωτηματολογίου, τη θέση του, το Id του και την κατάσταση του (αν έχει ήδη γίνει ή όχι). Υπάρχει πάνω από τη λίστα σύνδεσμος για την προσθήκη νέου ερωτηματολογίου(Εικόνα 6). Για να προσθέσει νέο ερωτηματολόγιο ο χρήστης θα πρέπει να συμπληρώσει το όνομα, την ερώτηση, τη θέση, τις απαντήσεις και το χρόνο του ερωτηματολογίου και κατά την προσθήκη οδηγείται πάλι στην προηγούμενη σελίδα (Εικόνα 7). Αν επιλέξει τώρα μία από τις ερωτήσεις της λίστας οδηγείται στην επόμενη σελίδα, όπου μπορεί να ενεργοποιήσει τη συγκεκριμένη ερώτηση για να την απαντήσουν οι φοιτητές (ή να την απενεργοποιήσει αν ήταν ήδη ενεργή) και να επιλέξει να δει τις απαντήσεις των φοιτητών(Εικόνα 8).



Εικόνα 17 Σελίδα Λίστας Ερωτηματολογίων



Εικόνα 18 Σελίδα Προσθήκης Ερωτηματολογίου

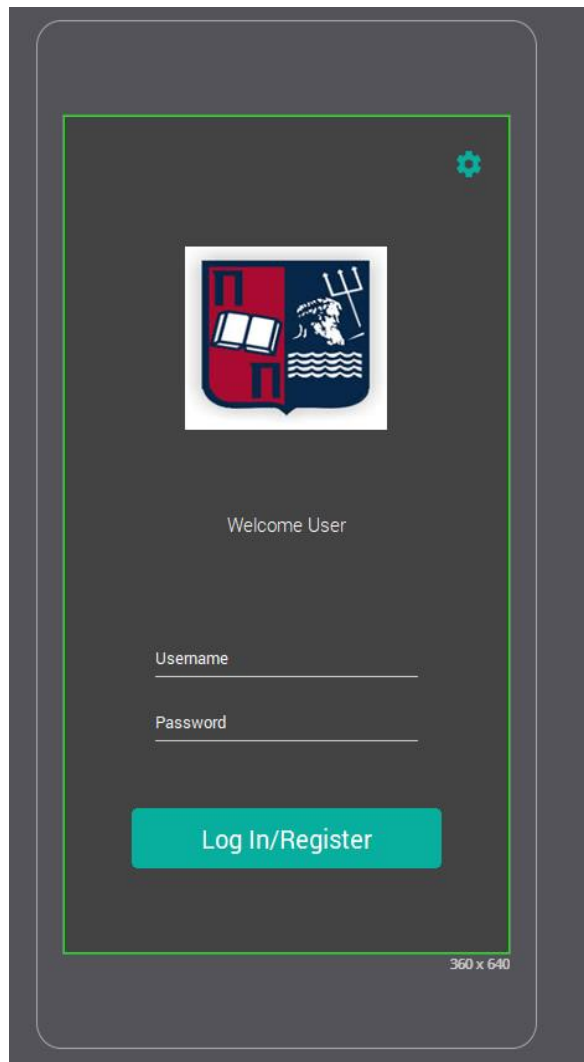


Εικόνα 19 Σελίδα Ενεργοποίησης Ερώτησης

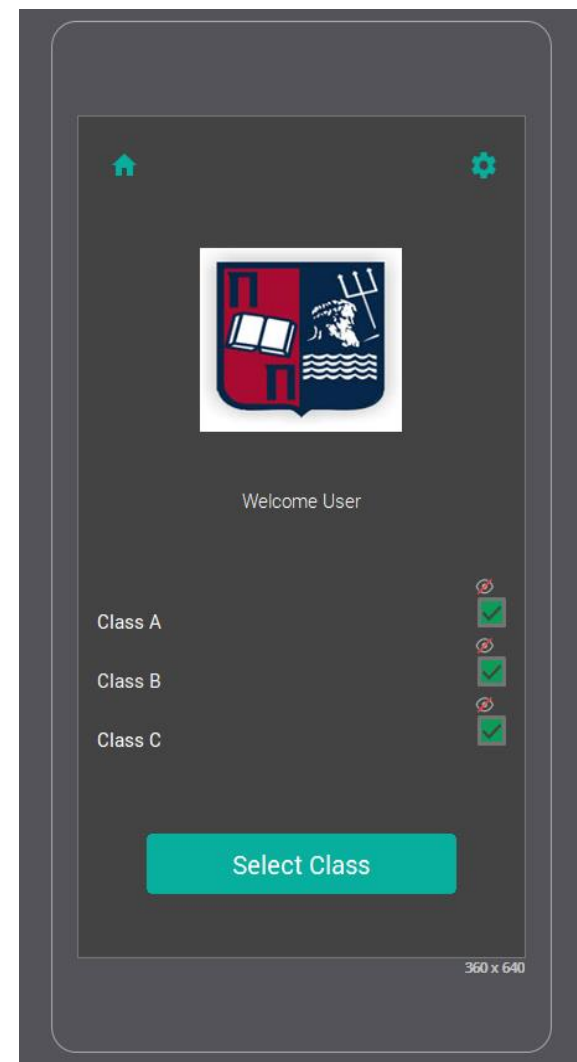
Όταν επιλέξει να δει τα αποτελέσματα του ερωτηματολογίου (Check Results κουμπί), οδηγείται στην τελική σελίδα (Εικόνα 9). Σε αυτή τη σελίδα, εμφανίζεται πάνω η ερώτηση και από κάτω το γράφημα με την κατανομή των απαντήσεων. Μετά, ένα κουμπί Show Answer με το οποίο εμφανίζεται η σωστή απάντηση. Θα δίνεται, επίσης η δυνατότητα να ελέγξει τα διαφορετικά τμήματα τι έχουν απαντήσει. Με το πάτημα του κουμπιού Next οδηγείται ο χρήστης κατευθείαν στην επόμενη ερώτηση.

Εικόνα 20 Σελίδα Εμφάνισης Απαντήσεων

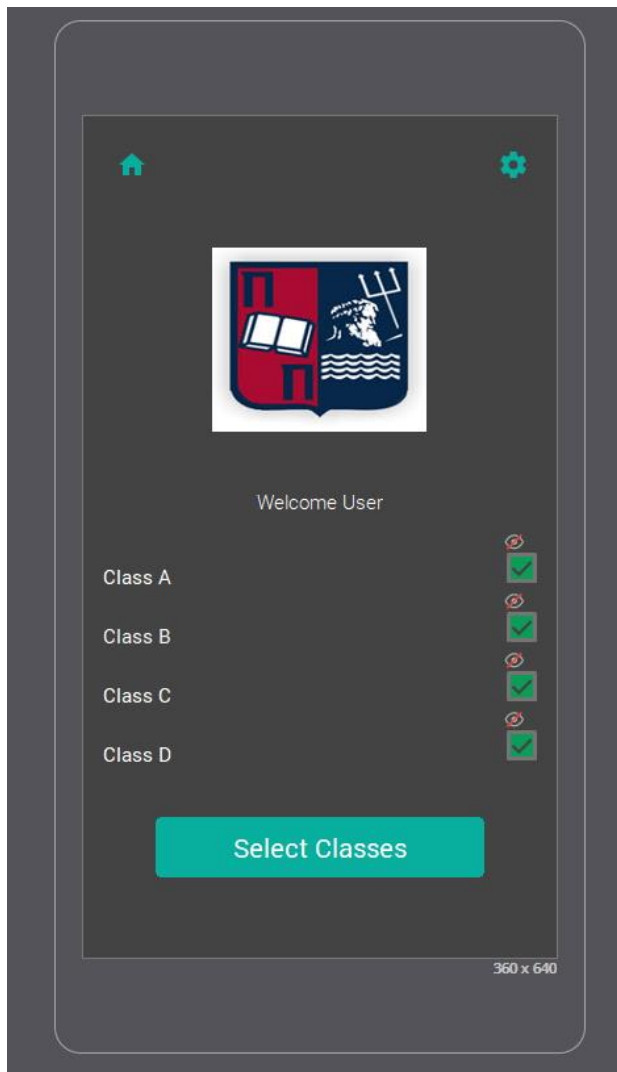
Στη συνέχεια παραθέτουμε και αναλύουμε τα πρωτότυπα που σχεδιάστηκαν για την εφαρμογή κινητού τηλεφώνου για τους φοιτητές. Αρχικά, οι φοιτητές εγγράφονται/συνδέονται στην εφαρμογή μας, με το username και τον κωδικό τους (Εικόνα 10). Με την επιτυχή σύνδεση, οδηγούνται στην κεντρική οθόνη, η οποία έχει τη λίστα με τα μαθήματα στα οποία είναι γραμμένοι οι φοιτητές. Πάνω αριστερά, υπάρχει ένα home button που οδηγεί σε αυτή την κεντρική σελίδα και πάνω δεξιά ένα κουμπί ρυθμίσεων (Εικόνα 11). Μέσα από εκεί μπορεί να γραφεί και στα μαθήματα που επιθυμεί ο χρήστης. Αυτό γίνεται στη δραστηριότητα εγγραφής μαθημάτων (Εικόνα 12), όπου ο φοιτητής απλά επιλέγει τα μαθήματα στα οποία επιθυμεί να εγγραφεί. Από την κεντρική δραστηριότητα, ο φοιτητής επιλέγοντας ένα μάθημα κατευθύνεται στη δραστηριότητα εύρεσης ερωτηματολογίου (Εικόνα 13). Εδώ έχει 2 επιλογές, είτε να βάλει τον κωδικό της ερώτησης που έχει δοθεί από τον καθηγητή, είτε να ελέγξει τις παλιές ερωτήσεις και τις απαντήσεις τους. Στη δεύτερη περίπτωση, πηγαίνει σε μία λίστα με τις παλιές ερωτήσεις του μαθήματος (Εικόνα 14) και επιλέγοντας κάποια βλέπει τη σωστή απάντηση της (Εικόνα 15). Αν εισάγει τον κωδικό της ερώτησης και η ερώτηση είναι ενεργή από τον καθηγητή, πάει στη δραστηριότητα όπου μπορεί να επιλέξει Α, Β, Γ, Δ (που αντιστοιχούν σε διαφορετικά χρώματα) για να απαντήσει την ερώτηση (Εικόνα 16). Τώρα, αν μέχρι να επιβεβαιώσει την απάντηση του, έχει λήξει ο χρόνος του ερωτηματολογίου, οδηγείται στην Εικόνα 17, όπου μπορεί είτε να πάει πίσω στο μάθημα είτε να δει τη σωστή απάντηση. Αντίθετα, αν επιβεβαιώσει πριν τη λήξη του χρόνου, πάει στην δραστηριότητα της Εικόνας 18, όπου μπορεί είτε να επιλέξει να αλλάξει την απάντηση που έχει ήδη δώσει είτε να περιμένει για την εμφάνιση της σωστής απάντησης.



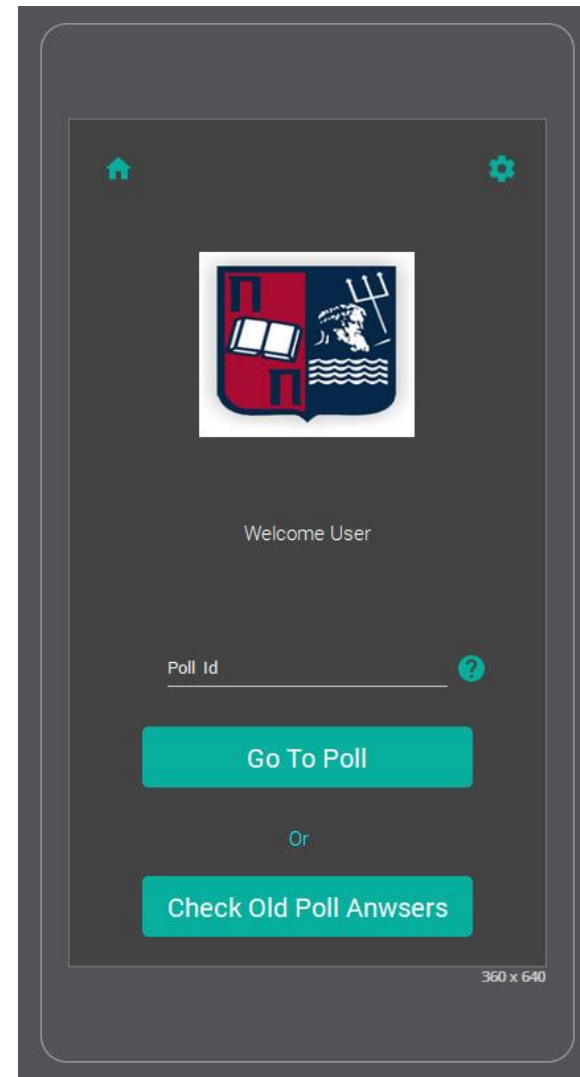
Εικόνα 21 Δραστηριότητα Εγγραφής Χρήστη



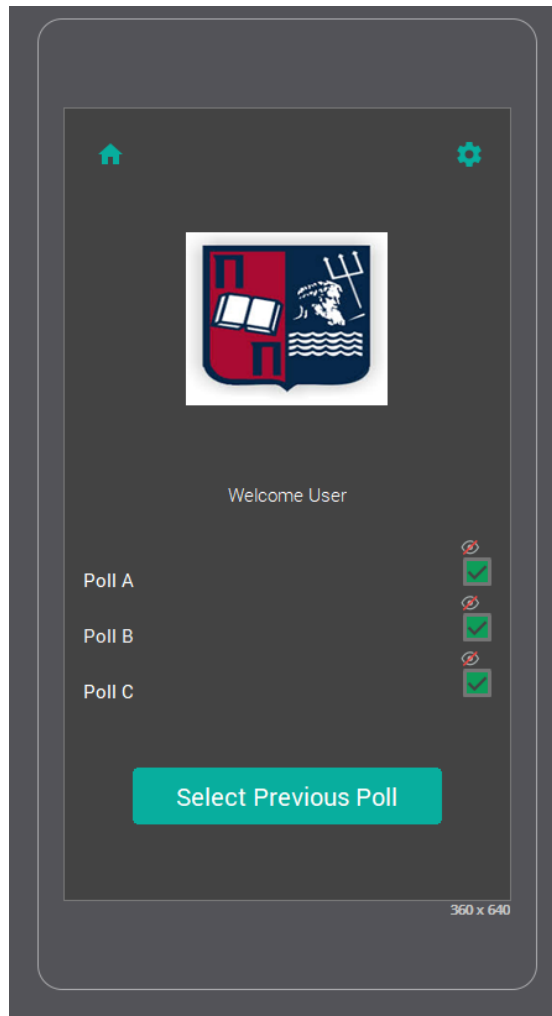
Εικόνα 22 Δραστηριότητα Επιλογής Μαθήματος



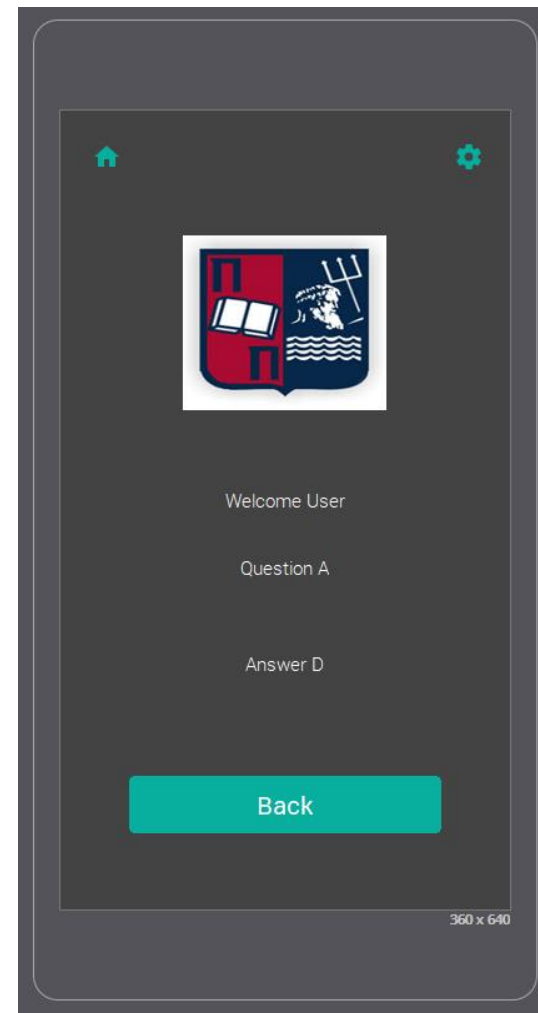
Εικόνα 23 Δραστηριότητα Εγγραφής σε Μαθήματα



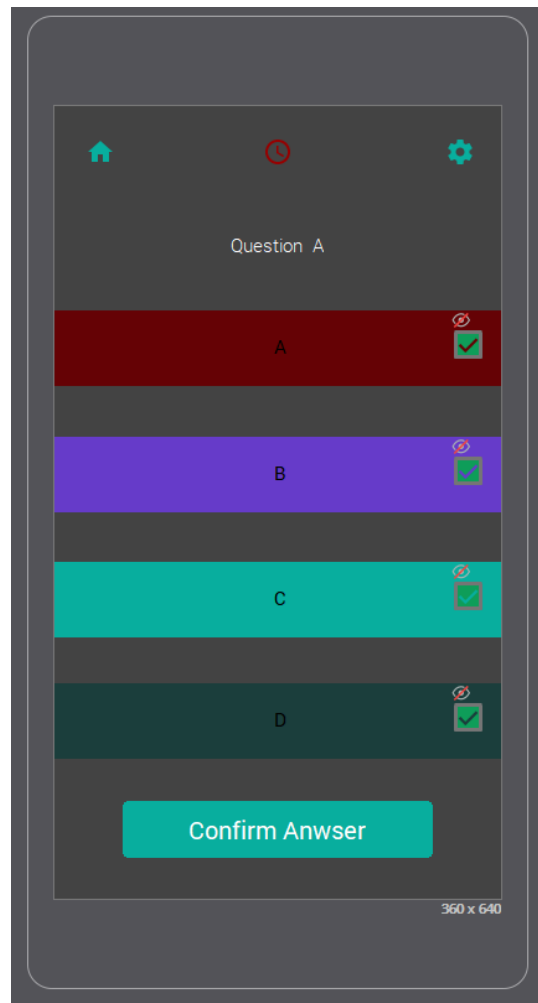
Εικόνα 24 Δραστηριότητα Εύρεσης Ερωτηματολογίου



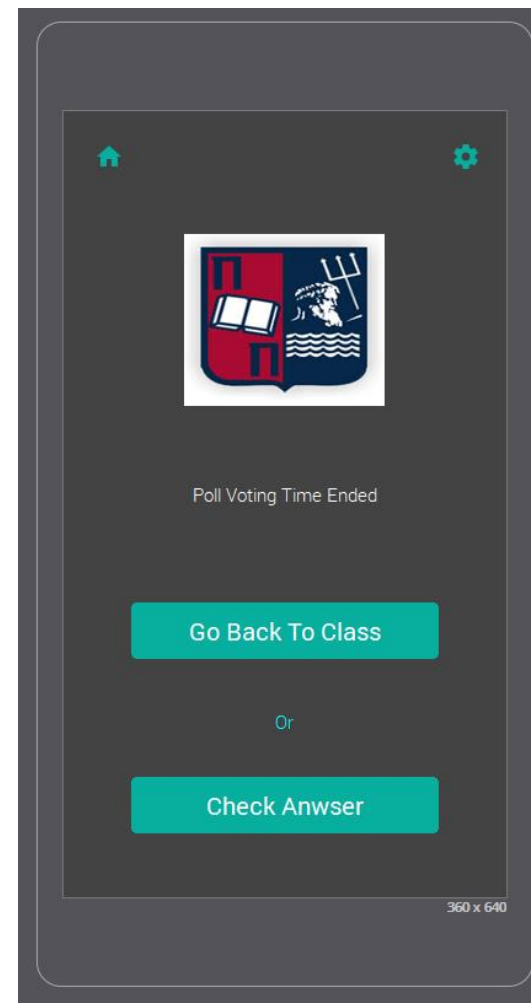
Εικόνα 25 Δραστηριότητα Παλιών Ερωτηματολογίων



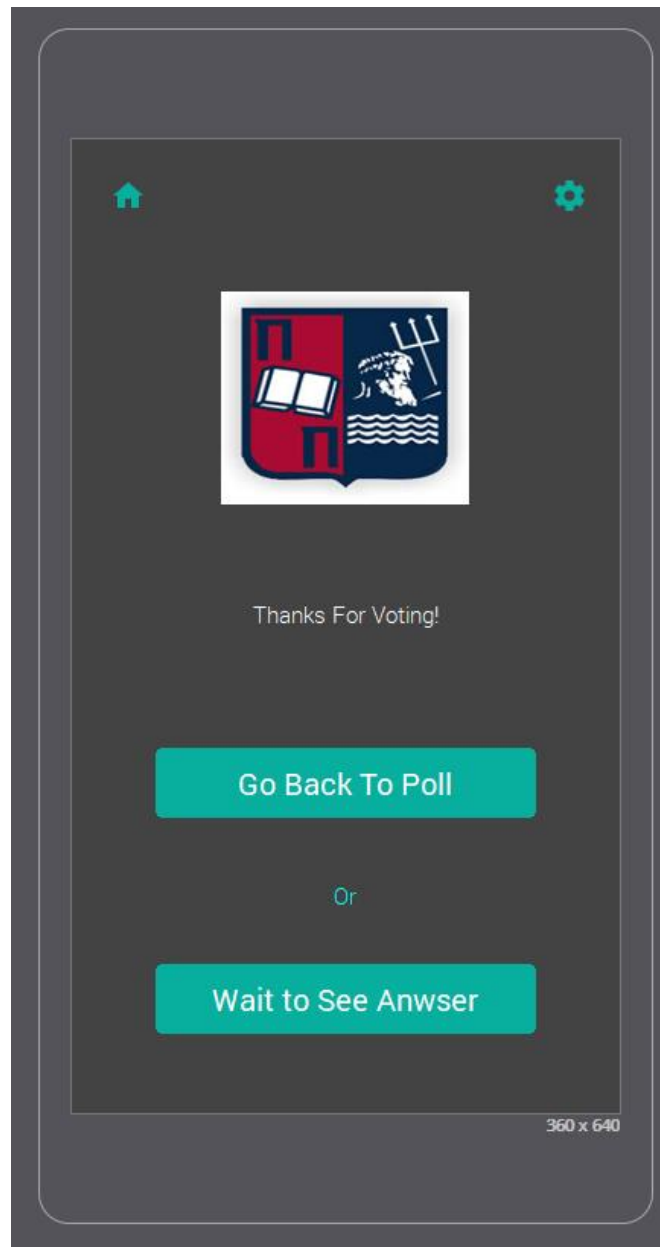
Εικόνα 26 Δραστηριότητα Εμφάνισης Παλιών Απαντήσεων



Εικόνα 27 Δραστηριότητα Ερώτησης

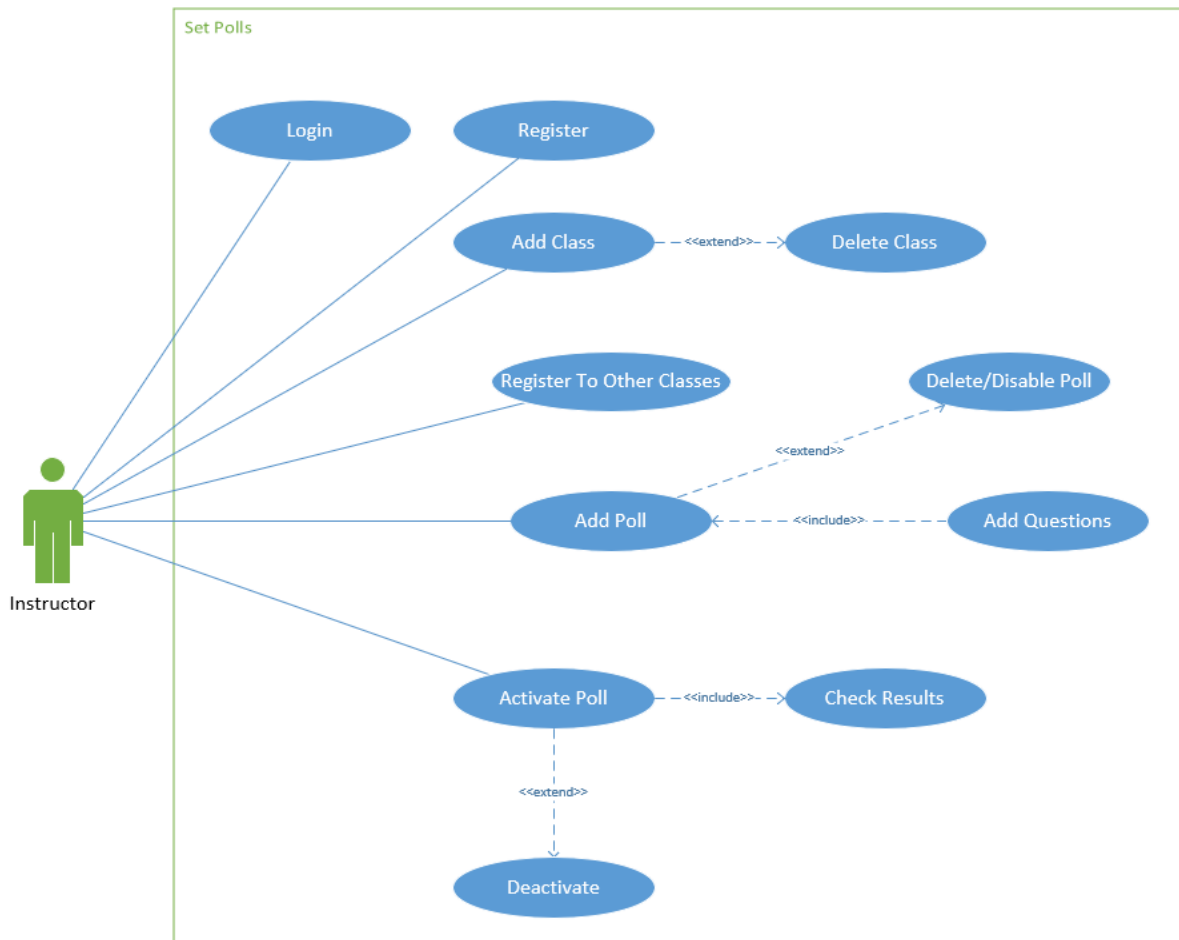


Εικόνα 28 Δραστηριότητα Τέλους Χρόνου

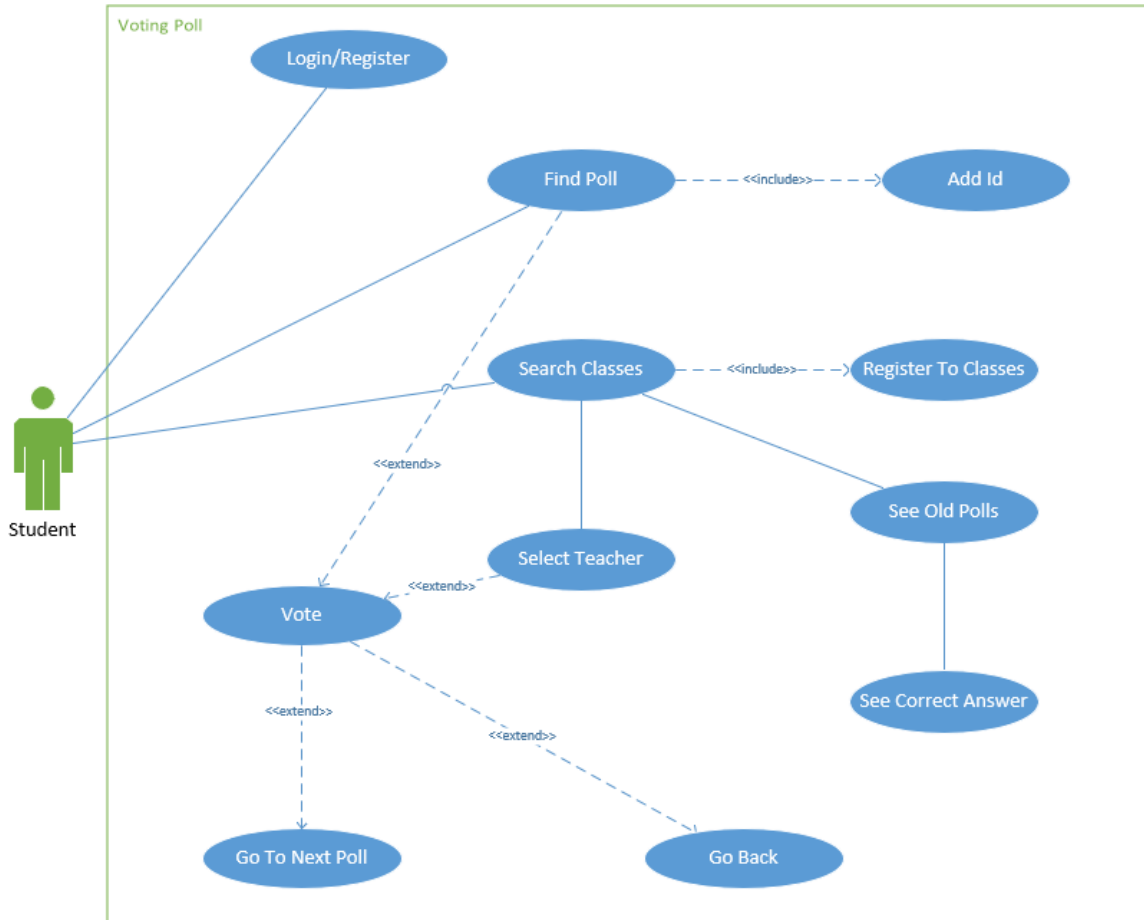


Εικόνα 29 Δραστηριότητα Εμφάνισης Απάντησης

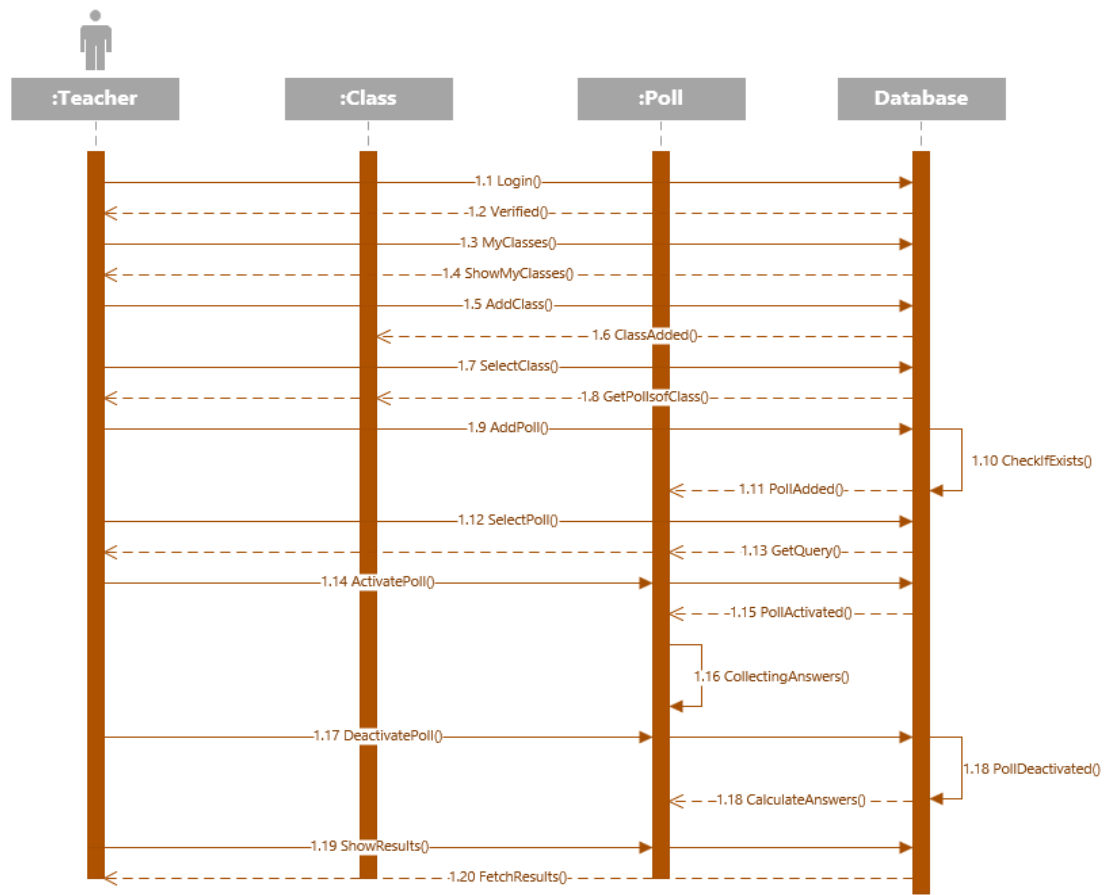
Στο δεύτερο στάδιο σχεδίασης έγιναν και πιο ξεκάθαρες οι απαιτήσεις και οι δυνατότητες των χρηστών, για αυτό και δημιουργήθηκαν εκ νέου ξεχωριστά διαγράμματα UML χρήσης και ακολουθίας για τους καθηγητές και τους φοιτητές. Παρατηρήθηκε, ότι οι καθηγητές θα χρειάζεται να συνδέονται στο σύστημα, να δημιουργούν και να διαγράφουν μαθήματα, να έχουν τη δυνατότητα να εγγράφονται σε άλλα μαθήματα, άλλων καθηγητών, να προσθέτουν Ερωτηματολόγια και ερωτήσεις και να τα ενεργοποιούν/απενεργοποιούν και να ελέγχουν τις απαντήσεις σε γραφήματα. Οι φοιτητές, με τη σειρά τους, θα πρέπει να συνδέονται στην εφαρμογή μας, να μπορούν να βρουν το ερωτηματολόγιο με βάση κάποιο Id που θα δίνεται από τον καθηγητή ή/και να αναζητούν συγκεκριμένα μαθήματα, ώστε να εγγράφονται σε αυτά και να μπορούν να δουν τις παλιές ερωτήσεις και σωστές απαντήσεις τους και τέλος να έχουν τη δυνατότητα να ψηφίζουν για τις ερωτήσεις που θέτει ο καθηγητής κατά τη διάρκεια του μαθήματος.



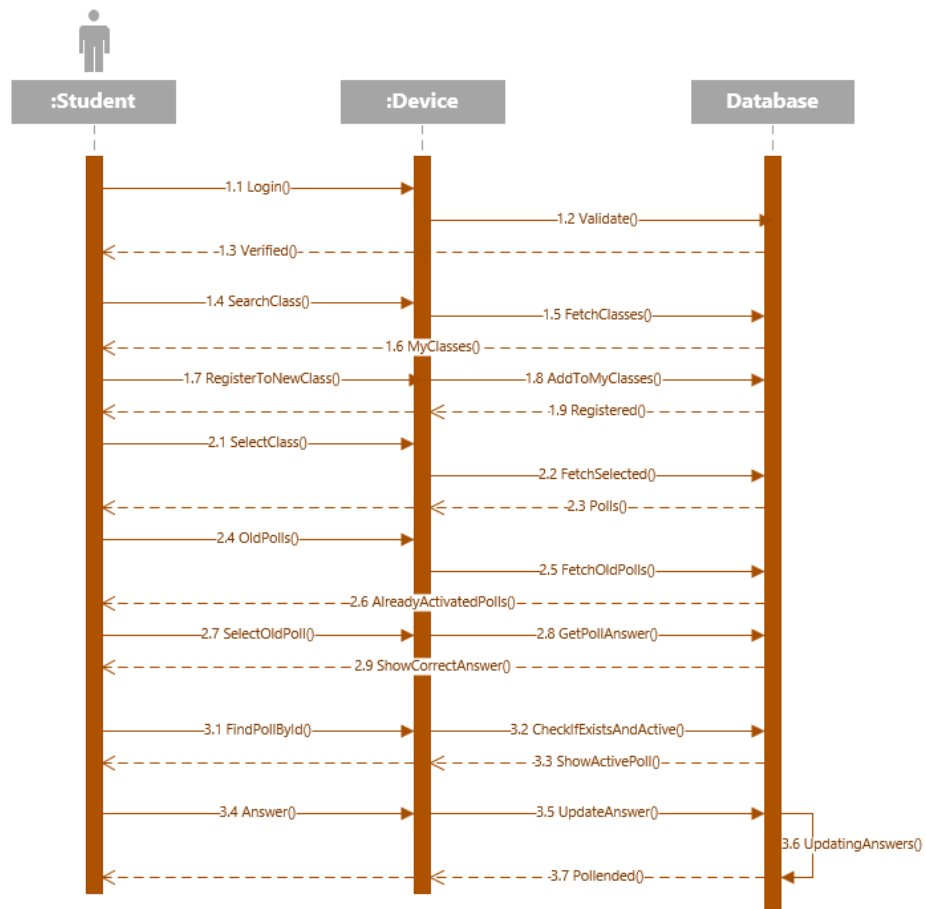
Εικόνα 30 Διάγραμμα Χρήσης Καθηγητή



Εικόνα 31 Διάγραμμα Χρήσης Φοιτητή



Εικόνα 32 Διάγραμμα Ακολουθίας Καθηγητή



Εικόνα 33 Διάγραμμα Ακολουθίας Φοιτητή

4 Αρχιτεκτονική

4.1 Γενικά

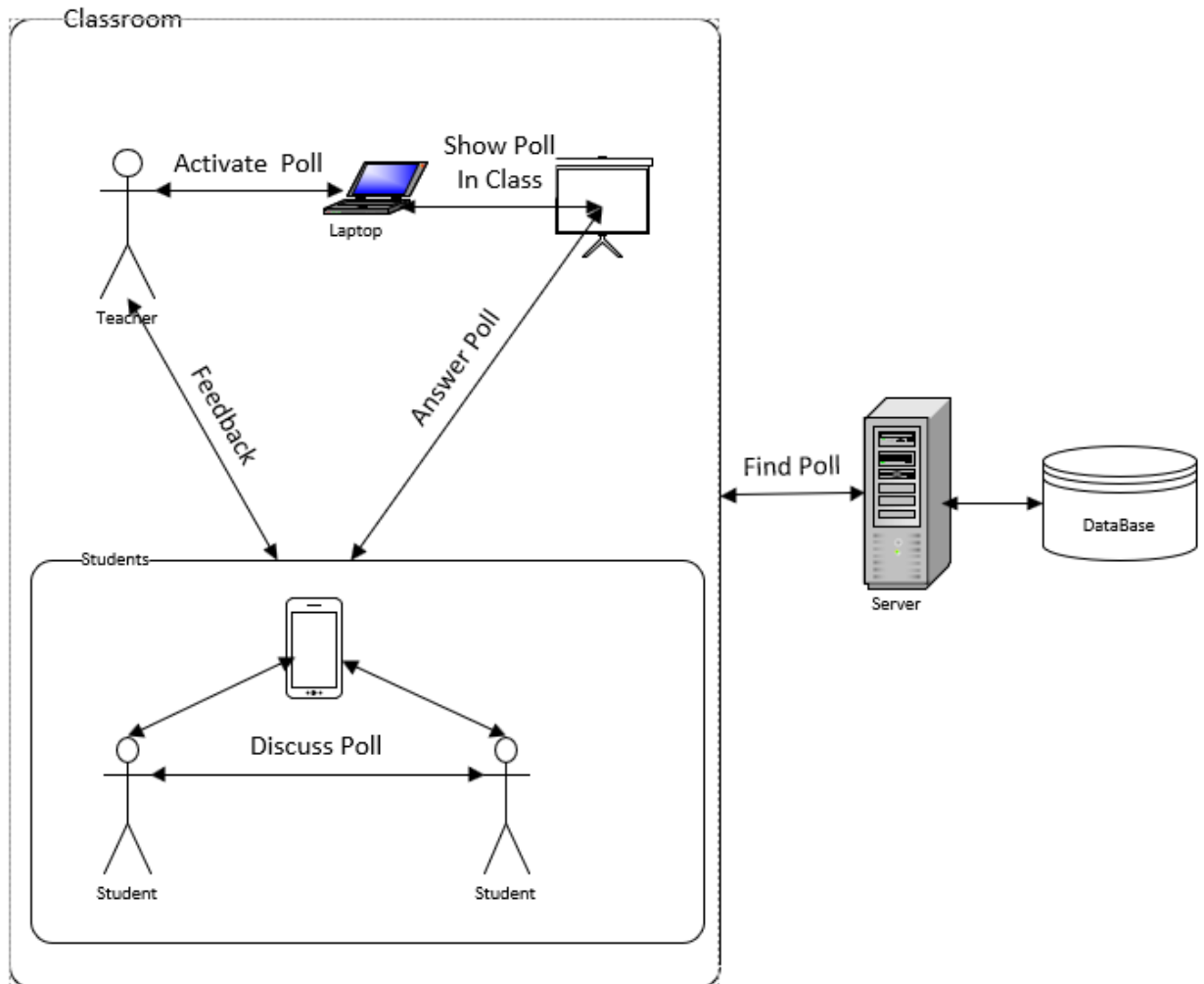
Έχοντας τελειώσει το κομμάτι του σχεδιασμού και της ανάλυσης, προχωρήσαμε στην υλοποίηση του συστήματος. Αρχικά, έπρεπε να επιλέξουμε τις τεχνολογίες που καλύπτουν βέλτιστα τις προδιαγραφές που αναλύσαμε, δηλαδή από τη μία εφαρμογή για κινητά τηλέφωνα με καθολική και εύκολη πρόσβαση και ένα εύχρηστο περιβάλλον για την ιστοσελίδα, όπου θα συνδέονται οι καθηγητές.

Για να καλύψουμε τις ανάγκες μας επιλέξαμε, λοιπόν, τη React-Native [30] και τη Laravel [31]. Η React-Native είναι ένα framework της JavaScript, που επιτρέπει με κώδικα JavaScript να γραφούν native εφαρμογές και για τα 2 βασικά λειτουργικά συστήματα των κινητών συσκευών, iOS και Android. Μας επιτρέπει πρακτικά να έχουμε ένα κώδικα που θα παίζει σε όλες τις συσκευές ανεξαρτήτως λογισμικού, κάνοντας εύκολη την υλοποίηση και τη διαδικασία δοκιμών της εφαρμογής μας. Αυτό το επιτυγχάνει με τη βοήθεια του Expo [32], μια βιβλιοθήκη εργαλείων που μετατρέπει τον JavaScript κώδικα στους αντίστοιχους native των κινητών.

Για την υλοποίηση της ιστοσελίδας επιλέξαμε τη Laravel, ένα Model-View-Controller (MVC)[35] framework της PHP, η οποία συμβάλλει στην εύκολη και σωστά δομημένη δημιουργία ιστοσελίδων. Ο εξυπηρετητής που χρησιμοποιήσαμε για την ιστοσελίδα είναι ο Internet Information Services (IIS) σε λειτουργικό Windows Server 2012 [36]. Τέλος, η βάση δεδομένων είναι MySQL και εκεί αποθηκεύονται και τα δεδομένα της εφαρμογής των φοιτητών και της ιστοσελίδας των καθηγητών [37].

Το σύστημά μας μπορεί να λειτουργήσει σε διάφορες διαμορφώσεις, αρκεί οι φοιτητές να έχουν πρόσβαση στο κινητό τους τηλέφωνο και οι καθηγητές, σε όποια συσκευή μπορεί να συνδεθεί στο διαδίκτυο και να επισκεφτεί ιστοσελίδες. Η βασική διαμόρφωση είναι που διεξάγεται κατά τη διάρκεια ενός μαθήματος. Ο καθηγητής μέσα στη ροή του μαθήματος και των διαφανειών που δείχνει από κάποιο προβολέα, παρεμβάλλει την ερώτηση που έχει ήδη εισάγει στην ιστοσελίδα μας. Οι φοιτητές συνδέονται στην εφαρμογή στο κινητό τους και ψάχνουν να βρουν την ενεργή ερώτηση. Στη συνέχεια μπορούν να την απαντήσουν και να αλλάξουν την απάντησή τους, μέσα στο χρόνο που τους διαθέτει ο καθηγητής. Ο καθηγητής, αφού απαντήσουν οι φοιτητές λαμβάνει σε πραγματικό χρόνο τις απαντήσεις και βλέπει άμεσα το γράφημα με τα ποσοστά ανά απάντηση. Αυτό επιτρέπει να αποσαφηνιστεί και για τον καθηγητή αλλά και για το σύνολο των φοιτητών το επίπεδο κατανόησης της τάξης της ερώτησης που έγινε από τον καθηγητή και να προσαρμοστεί η διδασκαλία αναλόγως. Μπορεί να συνεχίσει το μάθημα καθώς οι περισσότεροι φοιτητές απάντησαν σωστά, ή να διευκρινίσει κάτι που φαίνεται ότι δεν κατάλαβαν ή ακόμα να παροτρύνει σε ομαδικές συζητήσεις μεταξύ των φοιτητών και να ξαναψηφίσουν στο ερώτημα (Εικόνα 23). Όλα αυτά οδηγούν σε μια πιο ενεργή τάξη που με τη σειρά της, ενισχύει το ενδιαφέρον και την κατανόηση των φοιτητών επί το σύνολο της εκπαιδευτικής τους διαδικασίας [12][18][19][21][25].

Άλλες πιθανές διαμορφώσεις στις οποίες μπορεί να λειτουργήσει το σύστημά μας είναι εκπαίδευση από απόσταση ή μερικά, όπου τα ερωτηματολόγια παρουσιάζονται μέσω του διαδικτύου στους φοιτητές και εκείνοι απαντούν από τα κινητά τους, ακριβώς όπως και οι φοιτητές που βρίσκονται μέσα στην τάξη, ανεξαρτήτως από το που βρίσκονται. Αυτό διευκολύνει τη συμμετοχή των φοιτητών εξ αποστάσεως χωρίς να διαχωρίζονται από αυτό που βρίσκονται στην τάξη και χωρίς να χρειάζεται ο καθηγητής να οργανώσει ή να κάνει κάτι παραπάνω για την οποιαδήποτε από τις 2 κατηγορίες [20][26]. Θα μπορούσε επίσης να χρησιμοποιηθεί σε άλλες δομές ή ιδρύματα, όπως για παράδειγμα σε μαραθώνιους προγραμματισμού [15] ή σε τάξεις δημοτικού για εξοικείωση με την τεχνολογία εκπαιδευτικού περιεχομένου [16].



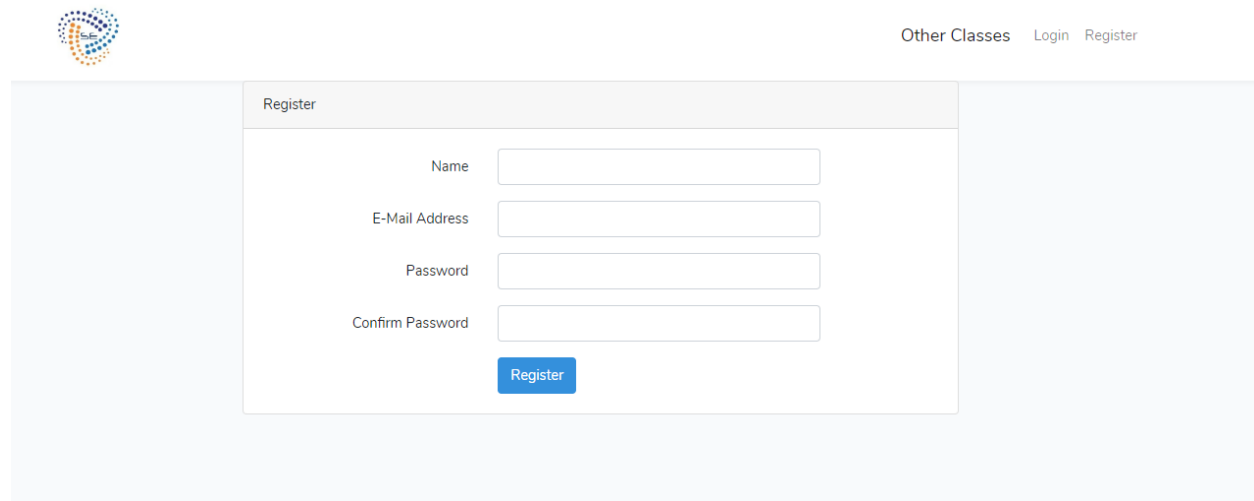
Εικόνα 34 Γενική Αρχιτεκτονική Συστήματος

4.2 Η μεριά του Εξυπηρετητή/Καθηγητών

Όπως αναφέραμε προηγουμένως, ο κύριος στόχος μας για τις δυνατότητες που θα παρέχουμε στους καθηγητές ήταν να τους επιτρέπεται άμεση ανατροφοδότηση κατά τη διάρκεια του μαθήματος, μέσα από ένα εύχρηστο και απλό σύστημα. Αυτό τους επιτρέπει να αναγνωρίζουν τις πιθανές αδυναμίες των φοιτητών σε κάποια ζητήματα και να προσαρμόζουν τη διάλεξη τους για να τις καλύψουν.

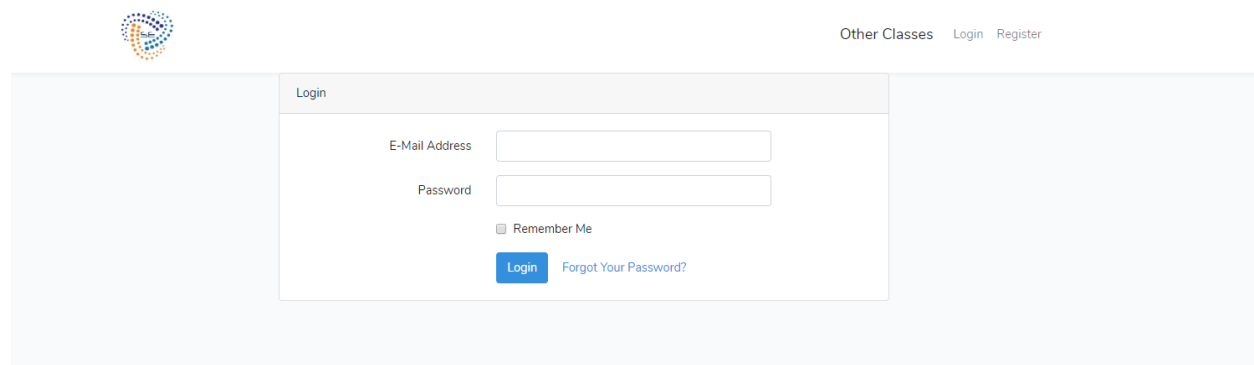
Αναπτύχθηκε λοιπόν μία ιστοσελίδα για που θα επιτρέψει στους καθηγητές τα παραπάνω και οι λειτουργίες της αναφέρονται αναλυτικά παρακάτω.

Αρχικά, οι καθηγητές πρέπει να γραφούν στο σύστημα, με το Όνομα τους, το e-mail τους και ένα κρυφό κωδικό (Εικόνα 24). Μετά την πρώτη φορά μπορούν να συνδεθούν από την σελίδα σύνδεσης, με το e-mail και τον κωδικός τους (Εικόνα 25).



The screenshot shows a web page with a logo on the top left and navigation links 'Other Classes', 'Login', and 'Register' on the top right. The main content area is titled 'Register' and contains a form with four input fields: 'Name', 'E-Mail Address', 'Password', and 'Confirm Password'. A blue 'Register' button is positioned below the 'Confirm Password' field.

Εικόνα 35 Οθόνη Εγγραφής



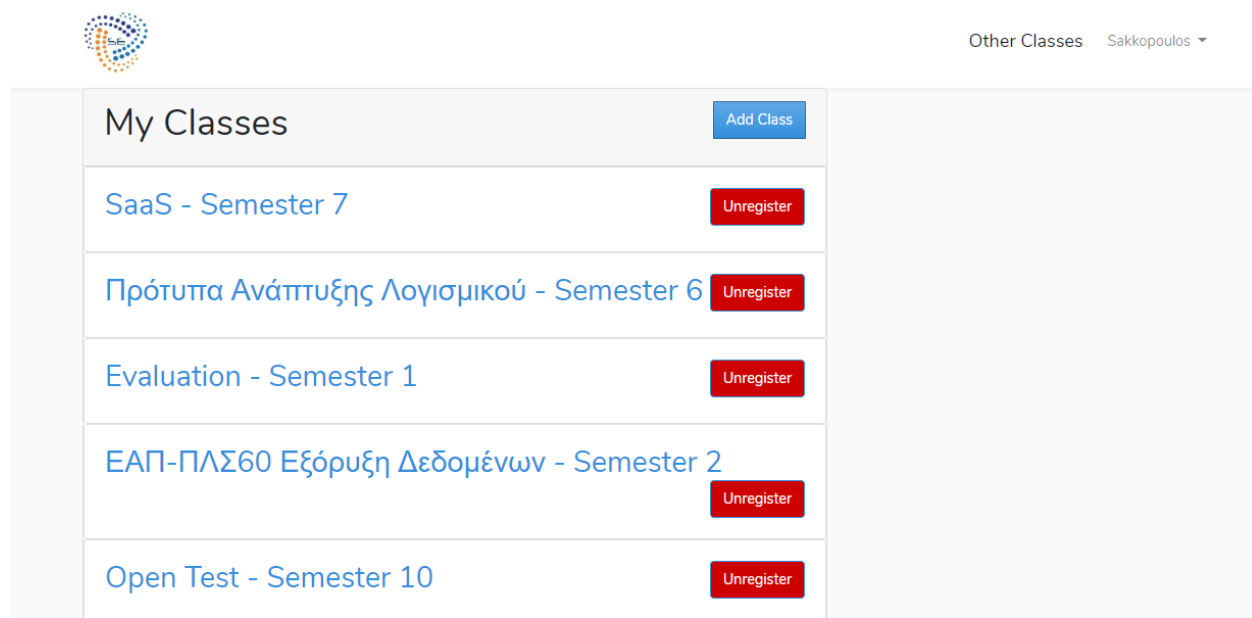
The screenshot shows a web page with a logo on the top left and navigation links 'Other Classes', 'Login', and 'Register' on the top right. The main content area is titled 'Login' and contains a form with two input fields: 'E-Mail Address' and 'Password'. Below the 'Password' field is a checkbox labeled 'Remember Me'. At the bottom of the form are a blue 'Login' button and a link 'Forgot Your Password?'.

Εικόνα 36 Οθόνη Σύνδεσης

Αφού συνδεθούν, οδηγούνται στην κεντρική σελίδα του συστήματος που περιέχει τη λίστα των εγγεγραμμένων μαθημάτων του συνδεδεμένου καθηγητή (Εικόνα 26). Στη σελίδα υπάρχει μία μπάρα πλοήγηση πάνω που παραμένει σταθερή σε όλες τις σελίδες. Αυτή έχει ένα κουμπί με το σήμα του εργαστηρίου της Τεχνολογίας Λογισμικού του Πανεπιστημίου Πειραιώς που με το πάτημα του οδηγεί στην κεντρική σελίδα. Επιπλέον, έχει ένα κουμπί Other Classes (άλλα μαθήματα), το οποίο οδηγεί στη λίστα των μαθημάτων που υπάρχουν και ο χρήστης δεν είναι γραμμένος. Τέλος, μπορεί να δει το όνομα του πάνω δεξιά το οποίο, αν το πατήσει με το ποντίκι θα εμφανιστεί από κάτω η επιλογή να αποσυνδεθεί (Logout) από το σύστημα.

Προφανώς, στην πρώτη σύνδεση του διδάσκοντα, η λίστα των μαθημάτων στη κεντρική σελίδα θα είναι κενή. Ο καθηγητής λοιπόν μπορεί είτε να δημιουργήσει ένα νέο μάθημα είτε να γραφεί σε υπάρχοντα μαθήματα. Για την πρώτη επιλογή μπορεί να πατήσει το κουμπί στη κεντρική σελίδα “Add Class” (Προσθήκη Μαθήματος) και να οδηγηθεί στη σελίδα δημιουργίας μαθήματος (Εικόνα 28). Εκεί πρέπει απλά να εισάγει το όνομα (Title) και το εξάμηνο (Semester) του μαθήματος και μετά να πατήσει το κουμπί Add (Προσθήκη). Με το πάτημα του κουμπιού το μάθημα έχει προστεθεί στη βάση δεδομένων και στη λίστα του συγκεκριμένου καθηγητή και κατευθύνεται στη κεντρική λίστα μαθημάτων, όπου πλέον θα υπάρχει το μάθημα αυτό.

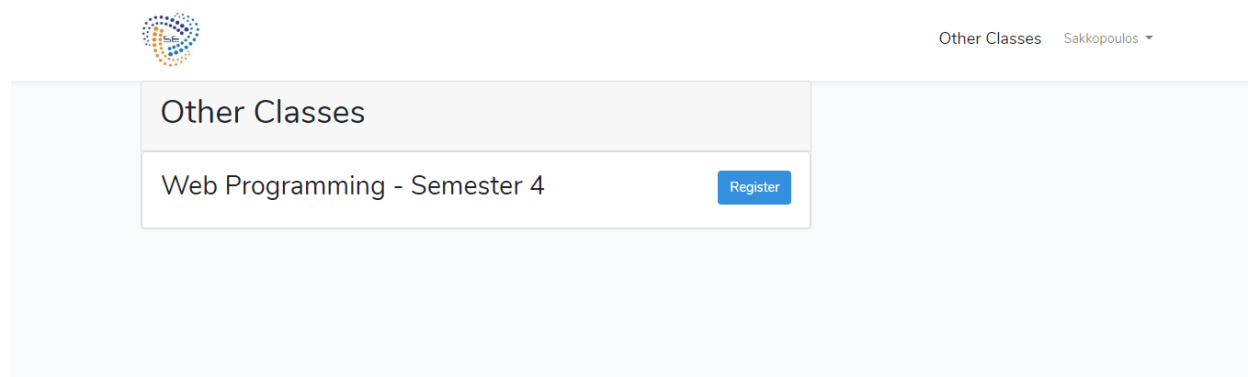
Η δεύτερη επιλογή του χρήστη για να προσθέσει μαθήματα στη λίστα του, είναι να εγγραφεί σε μαθήματα που έχουν ήδη προσθέσει άλλοι διδάσκοντες. Αυτό μπορεί να το επιτύχει πατώντας το κουμπί Other Classes και να πάει στη λίστα των υπολοίπων μαθημάτων που υπάρχουν πιθανά στο σύστημα και αυτός δεν είναι γραμμένος (Εικόνα 27). Τα μαθήματα της λίστας αυτά έχουν δίπλα ένα κουμπί Register (Εγγραφή), που με την επιλογή του μαθήματος που θέλει ο χρήστης να εγγραφεί και το πάτημα του αντίστοιχου κουμπιού., σβήνεται από η λίστα αυτή και προστίθεται στη λίστα των μαθημάτων του διδάσκοντα. Αντίστοιχα στη κεντρική σελίδα, δίπλα σε κάθε μάθημα υπάρχει ένα κουμπί Unregister (Απεγγραφή), με το πάτημα του οποίου ο χρήστης σταματάει πια να είναι γραμμένος στο μάθημα και προφανώς να εμφανίζεται αυτό στη λίστα των μαθημάτων του.



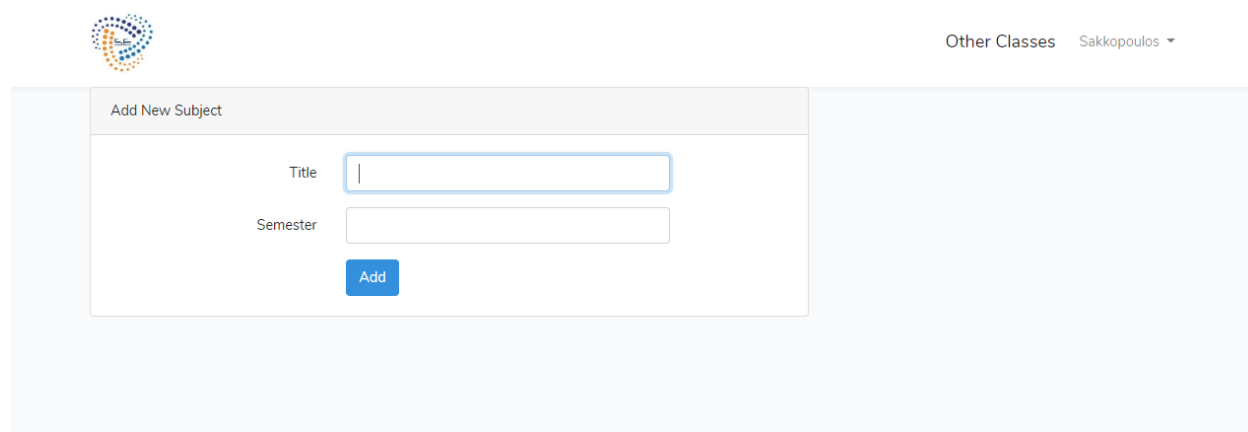
The screenshot shows a user interface for managing classes. At the top left is a logo, and at the top right are the labels 'Other Classes' and 'Sakkopoulos'. Below this is a section titled 'My Classes' with an 'Add Class' button. The main content is a table of classes, each with an 'Unregister' button.

Class Name	Action
SaaS - Semester 7	Unregister
Πρότυπα Ανάπτυξης Λογισμικού - Semester 6	Unregister
Evaluation - Semester 1	Unregister
ΕΑΠ-ΠΛΣ60 Εξόρυξη Δεδομένων - Semester 2	Unregister
Open Test - Semester 10	Unregister

Εικόνα 37 Οθόνη Λίστας Μαθημάτων



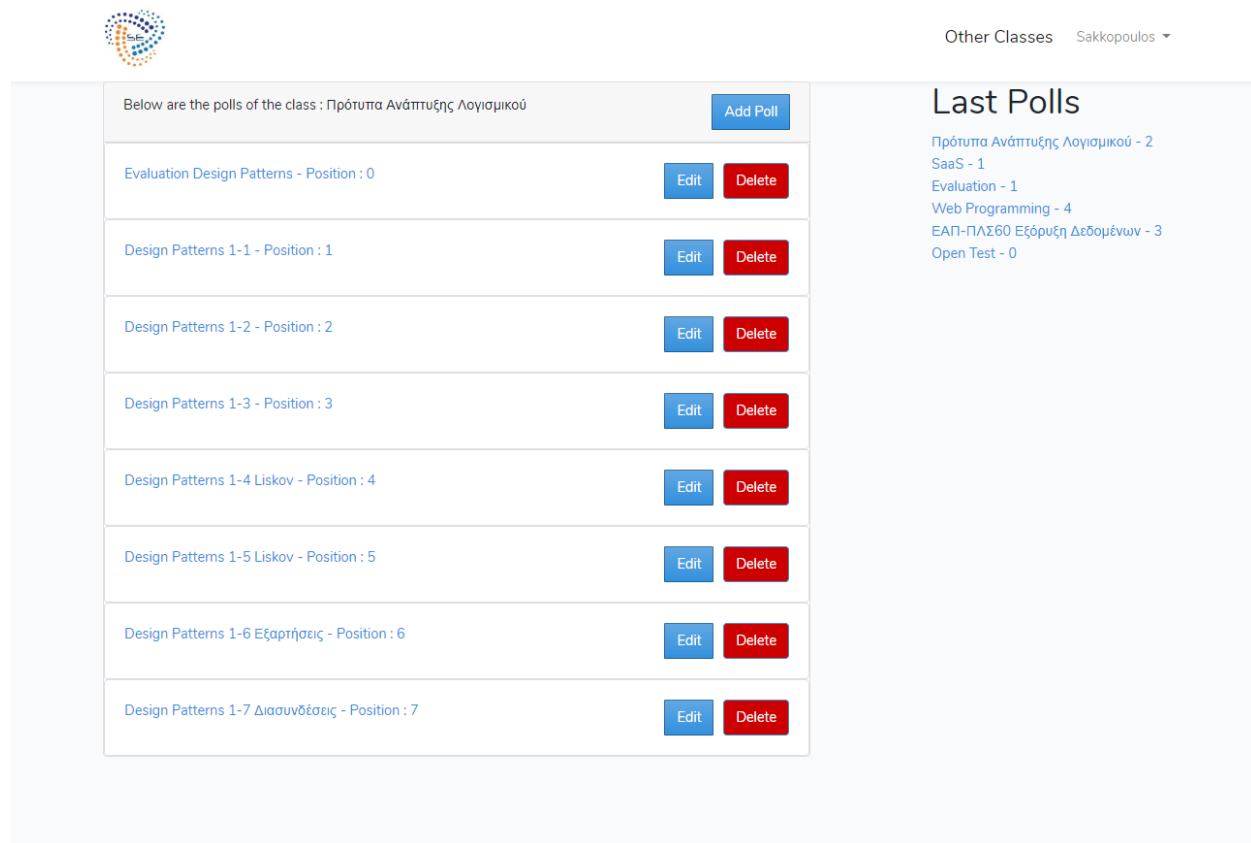
Εικόνα 38 Οθόνη Μαθημάτων Άλλων Καθηγητών



Εικόνα 39 Οθόνη Εισαγωγής Μαθήματος

Έχοντας πλέον, δημιουργήσει ή/και γραφεί στα μαθήματα που επιθυμεί ο χρήστης μπορεί να επιλέξει ένα από τα μαθήματα στη λίστα του, τα οποία λειτουργούν ως σύνδεσμοι. Θα οδηγηθεί, λοιπόν, στη σελίδα του μαθήματος που περιέχει τη λίστα των ερωτηματολογίων αυτού του μαθήματος (Εικόνα 29). Η μπάρα πλοήγησης πάνω είναι ίδια με αυτή που αναλύθηκε προηγουμένως, αλλά τώρα υπάρχει επιπλέον μία στήλη στα δεξιά της σελίδας με μια μικρή λίστα με τα τελευταία ερωτηματολόγια που έχει κάνει ο χρήστης για κάθε μάθημα του.

Τώρα, αν είναι νέο μάθημα προφανώς δεν θα υπάρχουν ερωτηματολόγια οπότε θα πρέπει να προσθέσει κάποια ο χρήστης. Για να το κάνει αυτό πρέπει να πατήσει το κουμπί Add Poll (προσθήκη ερωτηματολογίου) και θα οδηγηθεί στην κατάλληλη οθόνη. Η σελίδα προσθήκης ερωτηματολογίου απαιτεί την εισαγωγή των απαραίτητων στοιχείων του (Εικόνα 30).




The screenshot displays a web interface for managing polls. At the top left is a logo with the letters 'ΕΣΕ'. The main content area is titled 'Below are the polls of the class : Πρότυπα Ανάπτυξης Λογισμικού' and includes an 'Add Poll' button. Below this is a table listing eight polls, each with an 'Edit' button and a 'Delete' button. The polls are:

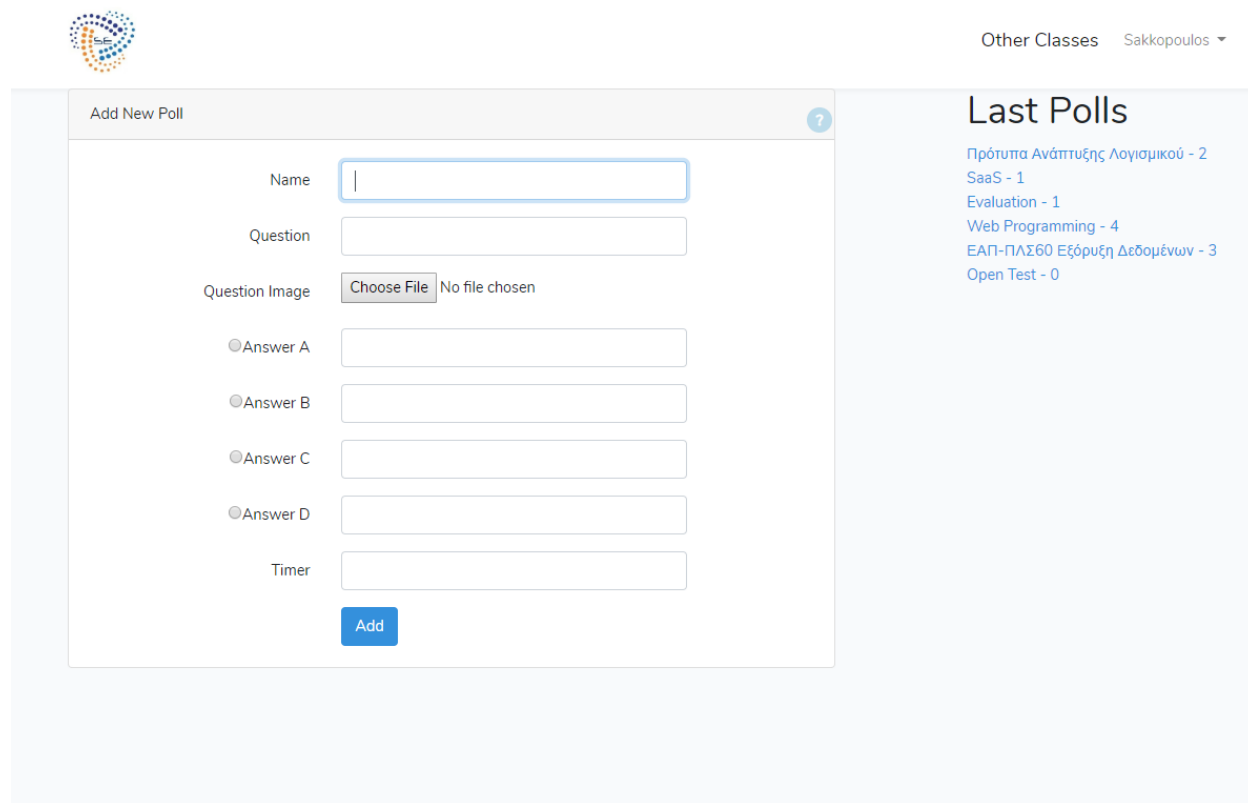
Poll Title	Position
Evaluation Design Patterns	0
Design Patterns 1-1	1
Design Patterns 1-2	2
Design Patterns 1-3	3
Design Patterns 1-4 Liskov	4
Design Patterns 1-5 Liskov	5
Design Patterns 1-6 Εξαρτήσεις	6
Design Patterns 1-7 Διασυνδέσεις	7

On the right side, there is a 'Last Polls' section listing:

- Πρότυπα Ανάπτυξης Λογισμικού - 2
- SaaS - 1
- Evaluation - 1
- Web Programming - 4
- ΕΑΠ-ΠΛΣ60 Εξόρυξη Δεδομένων - 3
- Open Test - 0

Εικόνα 40 Οθόνη Λίστας Ερωτηματολογίων

Στην οθόνη προσθήκης ερωτηματολογίου, πάνω δεξιά φαίνεται ένα βοηθητικό κουμπί, το , που όταν πάει το ποντίκι από πάνω του εμφανίζεται ένα βοηθητικό μήνυμα που λέει ότι η σωστή απάντηση του ερωτηματολογίου επιλέγεται από το Radio Button δίπλα από τις απαντήσεις. Τα απαραίτητα στοιχεία, λοιπόν που πρέπει να εισάγει ο χρήστης είναι, το όνομα της ερώτησης (για την εύρεση της από τη λίστα), την ίδια την ερώτηση, η οποία μπορεί να περιέχει κείμενο και μία εικόνα, τις 4 πιθανές απαντήσεις, επιλέγοντας δίπλα τη μία σωστή και προαιρετικά το χρόνο στον οποίο θα είναι ενεργή η ερώτηση. Στο παρόν σύστημα, επειδή είναι ακόμα σε στάδιο δοκιμής, έχει αποφασιστεί ότι ο καθηγητής θα αποφασίζει κατά τη διάρκεια του μαθήματος, πόσο χρόνο θα διαρκεί η κάθε ερώτηση και θα απενεργοποιείται από τον ίδιο. Η υλοποίηση όμως, να απενεργοποιείται με βάση το χρονόμετρο που έχει εισαχθεί υπάρχει. Με το πάτημα του κουμπού Add, προστίθεται το ερωτηματολόγιο στη βάση και συνδέεται με το μάθημα, οπότε οποιοσδήποτε καθηγητής γραφεί μετά στο μάθημα θα βλέπει αυτή την ερώτηση. Στη λίστα των ερωτηματολογίων εμφανίζονται τώρα όσες ερωτήσεις έχουν προστεθεί από τους διδάσκοντες του μαθήματος και είναι διατεταγμένα σε αύξουσα σειρά με βάση το position (τη θέση) τους, που με κάθε ερώτηση που προστίθεται να μπαίνει στο τέλος.



Other Classes Sakkopoulos ▾

Last Polls

- Πρότυπα Ανάπτυξης Λογισμικού - 2
- SaaS - 1
- Evaluation - 1
- Web Programming - 4
- ΕΑΠ-ΠΛΣ60 Εξόρυξη Δεδομένων - 3
- Open Test - 0

Add New Poll ⓘ

Name

Question

Question Image No file chosen

Answer A

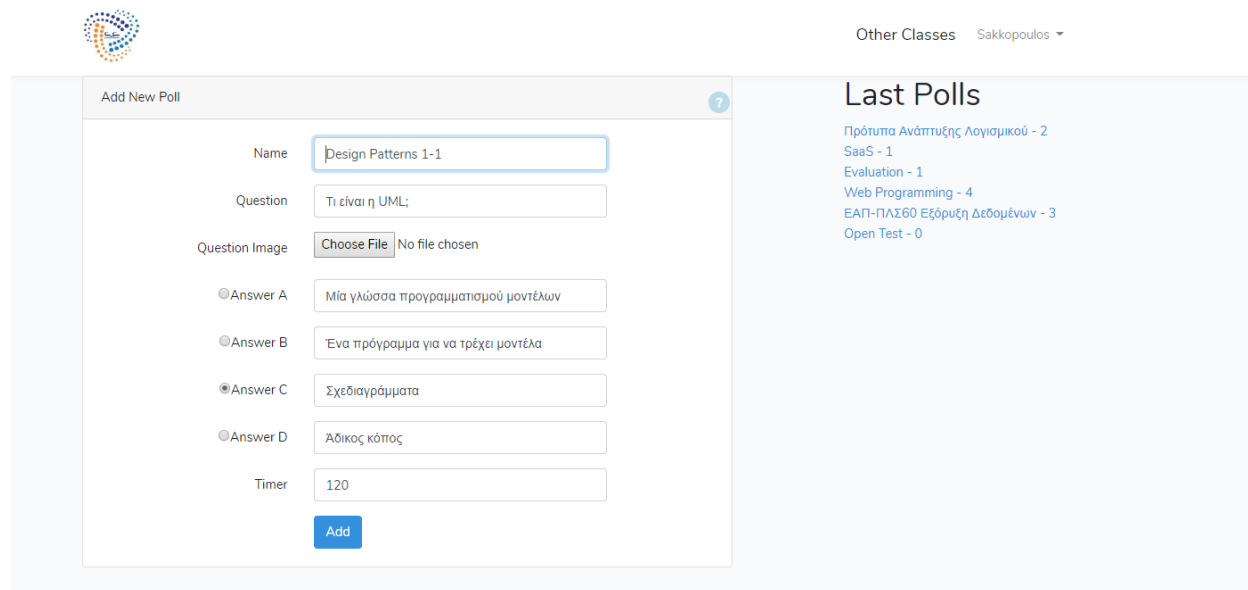
Answer B

Answer C

Answer D

Timer

Εικόνα 41 Οθόνη Προσθήκης Ερωτηματολογίου



Other Classes Sakkopoulos ▾

Last Polls

- Πρότυπα Ανάπτυξης Λογισμικού - 2
- SaaS - 1
- Evaluation - 1
- Web Programming - 4
- ΕΑΠ-ΠΛΣ60 Εξόρυξη Δεδομένων - 3
- Open Test - 0

Add New Poll ⓘ

Name

Question

Question Image No file chosen

Answer A

Answer B

Answer C

Answer D

Timer

Εικόνα 42 Οθόνη Επεξεργασίας Ερωτηματολογίου

Below are the polls of the class : Πρότυπα Ανάπτυξης Λογισμικού

Evaluation Design Patterns - Position : 0	Edit	Delete
Design Patterns 1-1 - Position : 1	Edit	Delete
Design Patterns 1-2 - Position : 2	Edit	Delete
Design Patterns 1-3 - Position : 3	Edit	Delete
Design Patterns 1-4 Liskov - Position : 4	Edit	Delete
Design Patterns 1-5 Liskov - Position : 5	Edit	Delete
Design Patterns 1-6 Εξαρτήσεις - Position : 6	Edit	Delete
Design Patterns 1-7 Διασυνδέσεις - Position : 7	Edit	Delete

Other Classes Sakkopoulos ▾

Last Polls

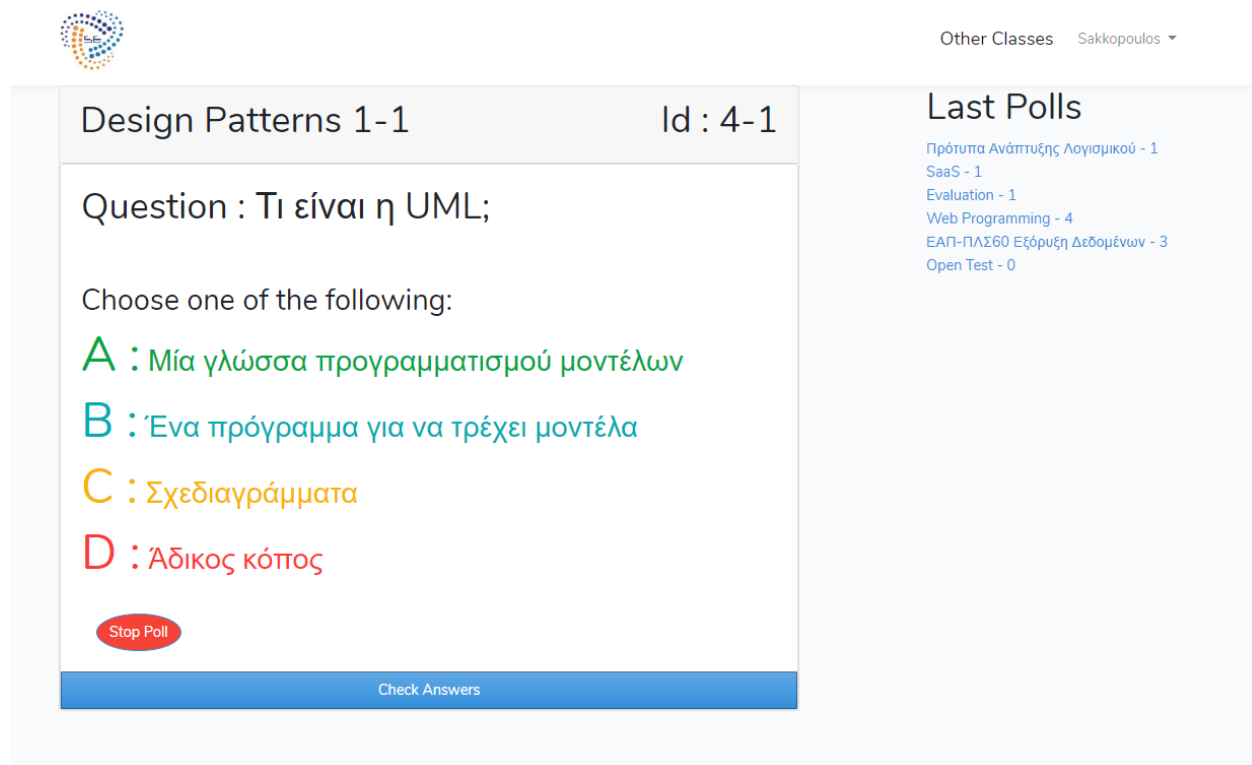
- Πρότυπα Ανάπτυξης Λογισμικού - 2
- SaaS - 1
- Evaluation - 1
- Web Programming - 4
- ΕΛΠ-ΠΛΣ60 Εξόρυξη Δεδομένων - 3
- Open Test - 0

Εικόνα 43 Οθόνη Επιβεβαίωσης Διαγραφής

Ο χρήστης από τη λίστα των ερωτηματολογίων τώρα, μπορεί είτε να επιλέξει να επεξεργαστεί κάποιο από αυτά ή να το διαγράψει ή να το ενεργοποιήσει. Στην πρώτη περίπτωση, οδηγείται στην οθόνη επεξεργασίας που πρακτικά είναι η ίδια με την οθόνη δημιουργίας, απλά υπάρχουν συμπληρωμένα τα στοιχεία του ερωτηματολογίου (Εικόνα 31). Μπορεί τώρα να επιλέξει όποιο στοιχείο αυτού θέλει και να το αλλάξει, με τα υπόλοιπα στοιχεία να παραμένουν ίδια, π.χ. να αλλάξει μόνο μία από τις 4 πιθανές απαντήσεις. Αν επιλέξει να διαγράψει το ερωτηματολόγιο εμφανίζεται ένα επιπλέον μήνυμα επιβεβαίωσης της διαγραφής, γιατί αφού επιλέξει ότι είναι βέβαιος για αυτήν διαγράφεται μόνιμα από τη βάση για το μάθημα και συνεπώς και για όλους τους καθηγητές (Εικόνα 32).

Στην τρίτη περίπτωση, όπου επιλέξει το σύνδεσμο του ερωτηματολογίου, οδηγείται στην οθόνη αυτού. Η οθόνη του ερωτηματολογίου πάνω έχει το όνομα του και το Id του (Εικόνα 33). Αυτό το id χρησιμοποιείται από την εφαρμογή των φοιτητών, αν θέλουν να βρουν το συγκεκριμένο ενεργό ερωτηματολόγιο. Το id αυτό δεν είναι μοναδικό για το συγκεκριμένο ερωτηματολόγιο αλλά αντιστοιχεί στο μάθημα και τον καθηγητή. Η λογική πίσω από αυτό είναι ότι ένας καθηγητής ανά πάσα στιγμή θα κάνει ένα μάθημα και άρα θα έχει ενεργό ένα ερωτηματολόγιο, αλλά μπορεί στο ίδιο μάθημα να το κάνουν 2 διαφορετικοί καθηγητές, οπότε δεν αρκεί μόνο να ελέγχουμε το ενεργό μάθημα. Εξασφαλίζεται ότι μόνο ένα μάθημα ανά καθηγητή μπορεί να είναι ενεργό, ακόμα και αν ο καθηγητής έχει αφήσει κάποιο μάθημα χωρίς να το απενεργοποιήσει.

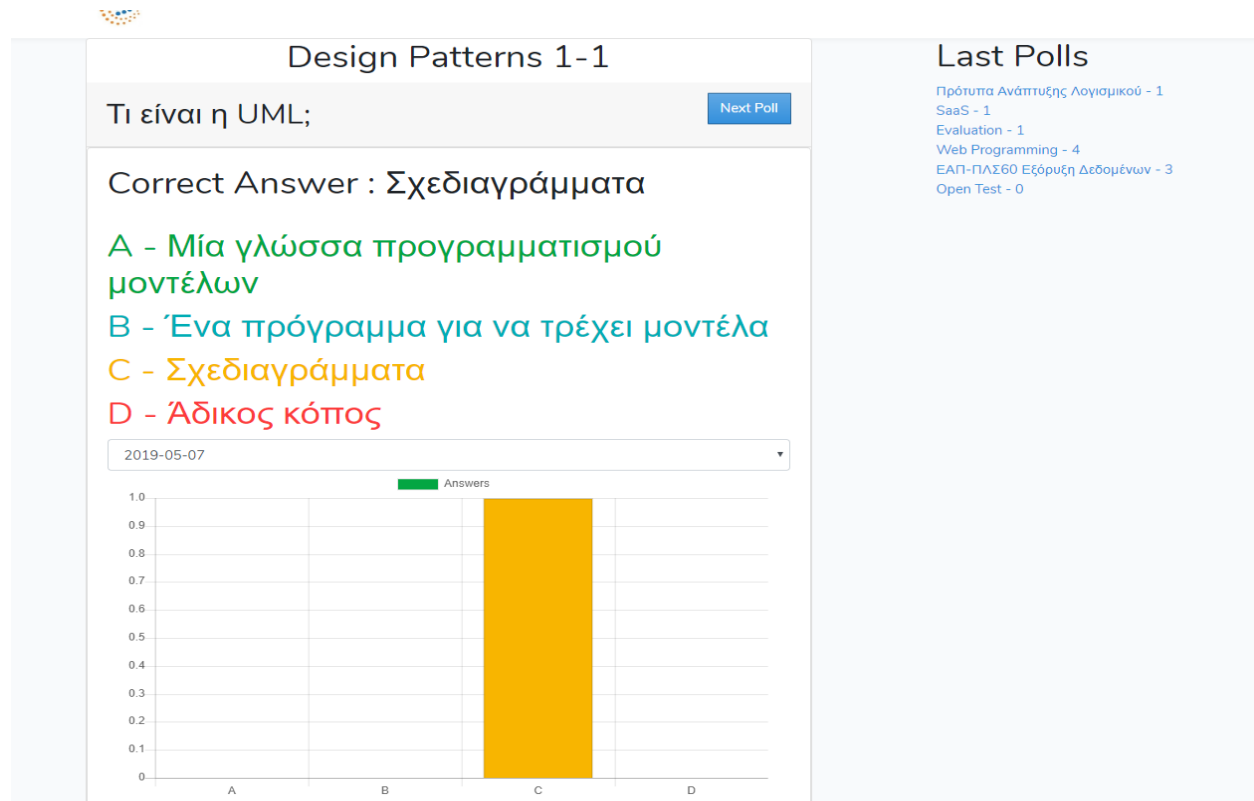
Έπειτα, υπάρχει η ερώτηση και οι 4 πιθανές απαντήσεις με 4 διαφορετικά χρώματα που αντιστοιχούν στα χρώματα των κουμπιών της εφαρμογής των φοιτητών. Ο καθηγητής μπορεί να απενεργοποιήσει (ή να ενεργοποιήσει με το ίδιο κουμπί) το ερωτηματολόγιο, οπότε επιθυμεί με το πάτημα του κουμπιού Stop Poll. Τέλος, όταν αποφασίσει να δει τις απαντήσεις των φοιτητών μπορεί να οδηγηθεί στην κατάλληλη οθόνη με το πάτημα του Check Answers (Έλεγχος Απαντήσεων). Σε κάθε περίπτωση, όταν ο καθηγητής πάει να ελέγξει τις απαντήσεις απενεργοποιούμε το ερωτηματολόγιο.



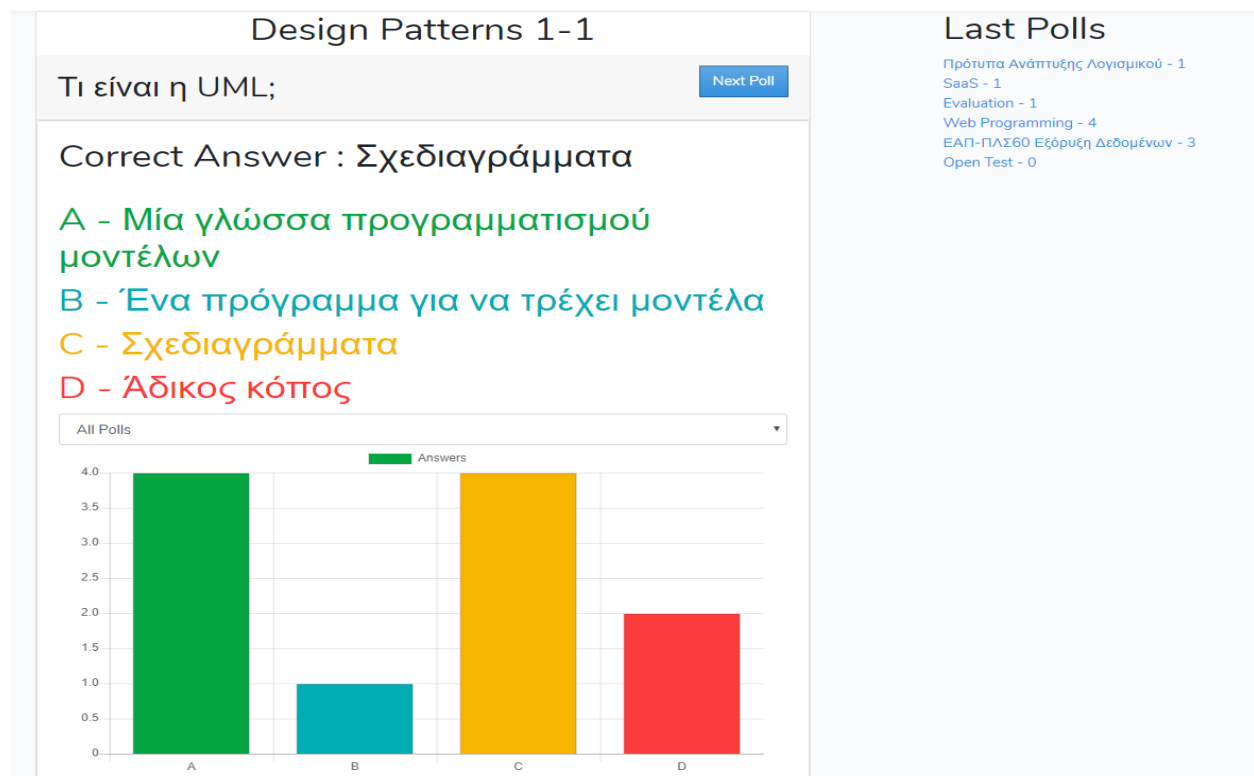
The screenshot shows a poll interface for a course titled "Design Patterns 1-1" with ID "4-1". The question is "Question : Τι είναι η UML;". Below the question, it says "Choose one of the following:". There are four options: A: Μία γλώσσα προγραμματισμού μοντέλων (green), B: Ένα πρόγραμμα για να τρέχει μοντέλα (blue), C: Σχεδιαγράμματα (yellow), and D: Άδικος κόπος (red). At the bottom left, there is a "Stop Poll" button, and at the bottom center, there is a "Check Answers" button. On the right side, there is a "Last Polls" section listing other polls: Πρότυπα Ανάπτυξης Λογισμικού - 1, SaaS - 1, Evaluation - 1, Web Programming - 4, ΕΑΠ-ΠΛΣ60 Εξόρυξη Δεδομένων - 3, and Open Test - 0.

Εικόνα 44 Οθόνη Ερωτηματολογίου

Οι απαντήσεις των φοιτητών για όση διάρκεια είναι ενεργό το ερωτηματολόγιο συλλέγονται σε πραγματικό χρόνο και αποθηκεύονται στη βάση δεδομένων και για τη συγκεκριμένη ημερομηνία. Πάνω φαίνεται το όνομα του ερωτηματολογίου, η ερώτηση και ένα κουμπί Next Poll που οδηγεί στο επόμενο με βάση τη θέση (position) ερωτηματολογίου. Αν δεν υπάρχει επόμενο ερωτηματολόγιο οδηγεί πίσω στη λίστα (Εικόνα 29). Από κάτω εμφανίζεται η σωστή απάντηση με μαύρα γράμματα και οι 4 πιθανές απαντήσεις με τα χρώματα που τους αντιστοιχούν (Εικόνα 34). Τα αποτελέσματα των απαντήσεων παρουσιάζονται μέσα από γράφημα με μπάρες, όπου πάλι αντιστοιχούν τα χρώματα κάθε μπάρας με των απαντήσεων. Τα γραφήματα αυτά έγιναν με μία βιβλιοθήκη ειδική για γραφήματα για το Laravel [33]. Πάνω από το γράφημα της εικόνας 34, μπορεί να παρατηρηθεί ένα drop-down menu με μία ημερομηνία. Από αυτή την επιλογή μπορεί εύκολα ο καθηγητής να επιλέξει παλαιότερες ημερομηνίες ή συνολικά αποτελέσματα και να συγκρίνει τα αποτελέσματα ή να δει την πρόοδο των φοιτητών του (Εικόνα 35).



Εικόνα 45 Οθόνη Απαντήσεων Α

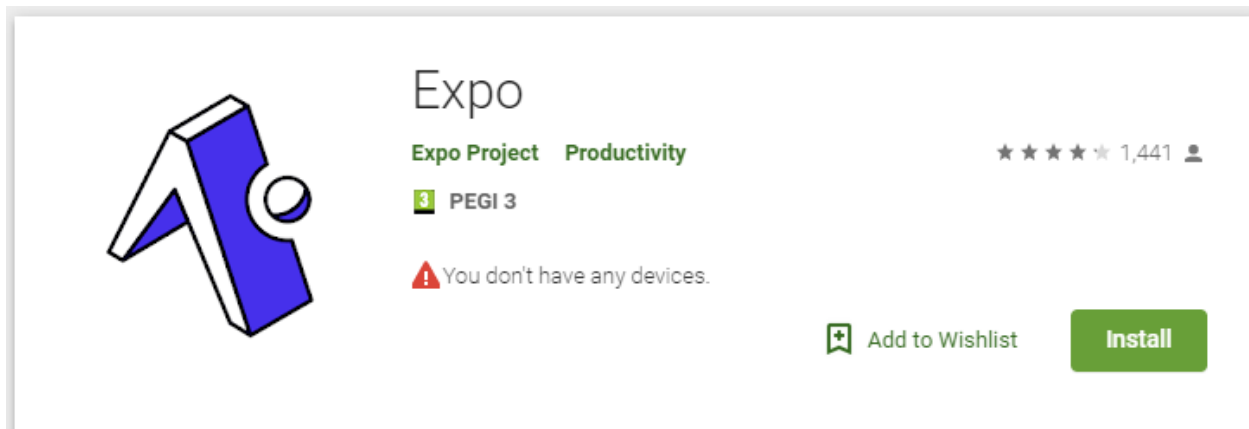


Εικόνα 46 Οθόνη Απαντήσεων Β

4.3 Η μεριά του Πελάτη/Φοιτητών

Έχοντας περιγράψει αναλυτικά τι γίνεται από τη μεριά των καθηγητών, συνεχίζουμε με την περιγραφή των δυνατοτήτων και λειτουργιών των φοιτητών. Όπως αναφέραμε, για τους φοιτητές έχει υλοποιηθεί μία εφαρμογή για κινητά τηλέφωνα σε React-Native που τρέχει μέσω του Expo και σε Android αλλά και σε iOS. Αφετηριακός μας στόχος ήταν η δημιουργία μια εφαρμογής που θα επιτρέπει στους φοιτητές να συμμετέχουν ενεργά κατά τη διάρκεια του μαθήματος. Θέλουμε να τους δώσουμε τη δυνατότητα όσο εύκολα θέτει ο καθηγητής μία ερώτηση εξίσου εύκολα να μπορούν να την απαντήσουν όλοι οι φοιτητές ταυτοχρόνως. Οι αναγκαίες προϋποθέσεις για να χρησιμοποιήσουν την εφαρμογή μας οι φοιτητές είναι να έχουν ένα κινητό τηλέφωνο μαζί τους (smart phone) και να έχουν πρόσβαση στο διαδίκτυο. Τα περισσότερα μεγάλα εκπαιδευτικά ιδρύματα παρέχουν στους φοιτητές τους δωρεάν πρόσβαση σε δίκτυο, το ίδιο ισχύει και στο Πανεπιστήμιο Πειραιά, όπου έγιναν τα πειράματα.

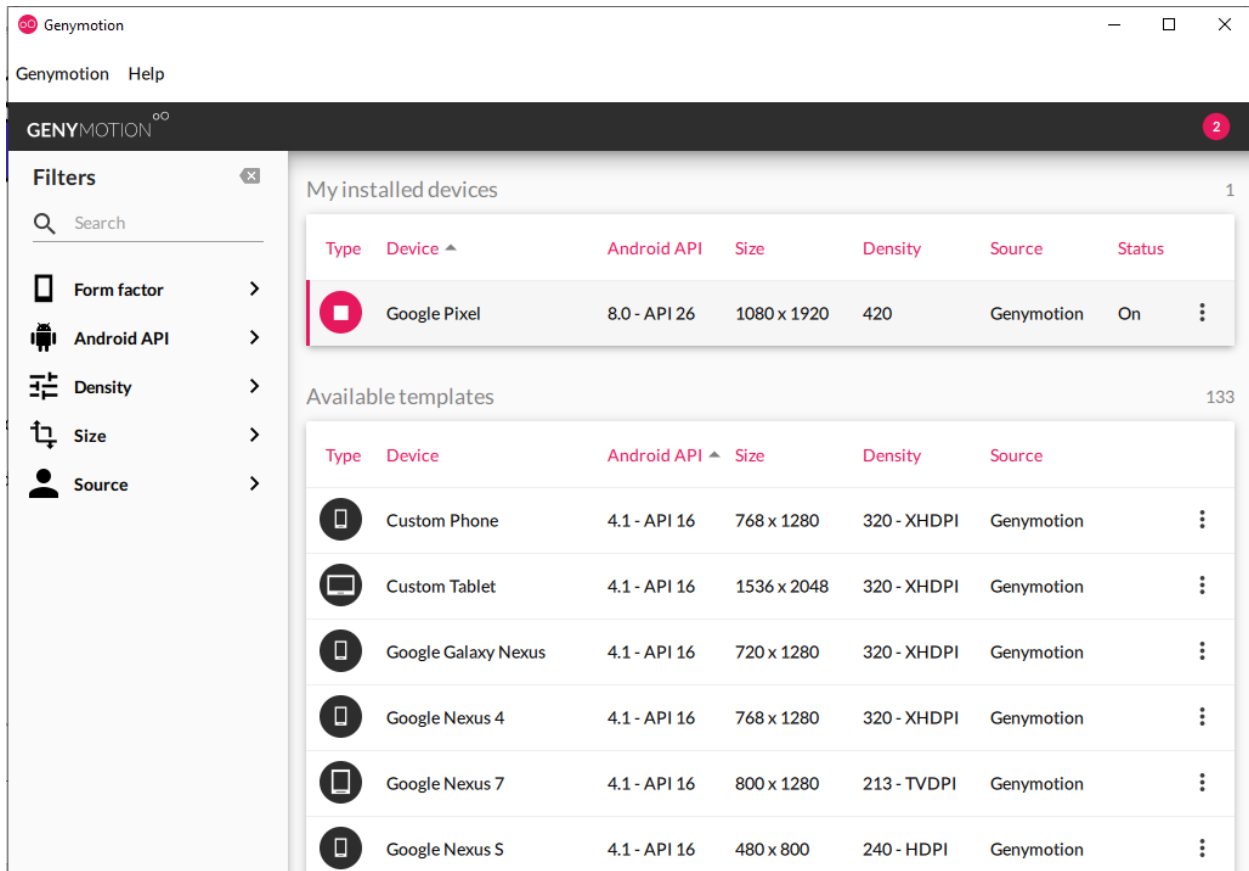
Οι φοιτητές επίσης, πρέπει να κατεβάσουν στο κινητό τους από το αντίστοιχο ηλεκτρονικό κατάστημα την εφαρμογή του Expo, η οποία παρέχεται δωρεάν και στα δύο (Εικόνα 36). Έχοντας κατεβάσει την εφαρμογή μπορούν να αναζητήσουν την εφαρμογή μας μέσα από το Expo και να την τρέξουν κατευθείαν μέσω αυτού. Στο μέλλον μπορεί να ανεβεί η εφαρμογή ξεχωριστά στα 2 e-Store, αλλά προς το παρόν η λύση μέσω του Expo καλύπτει τις ανάγκες μας.



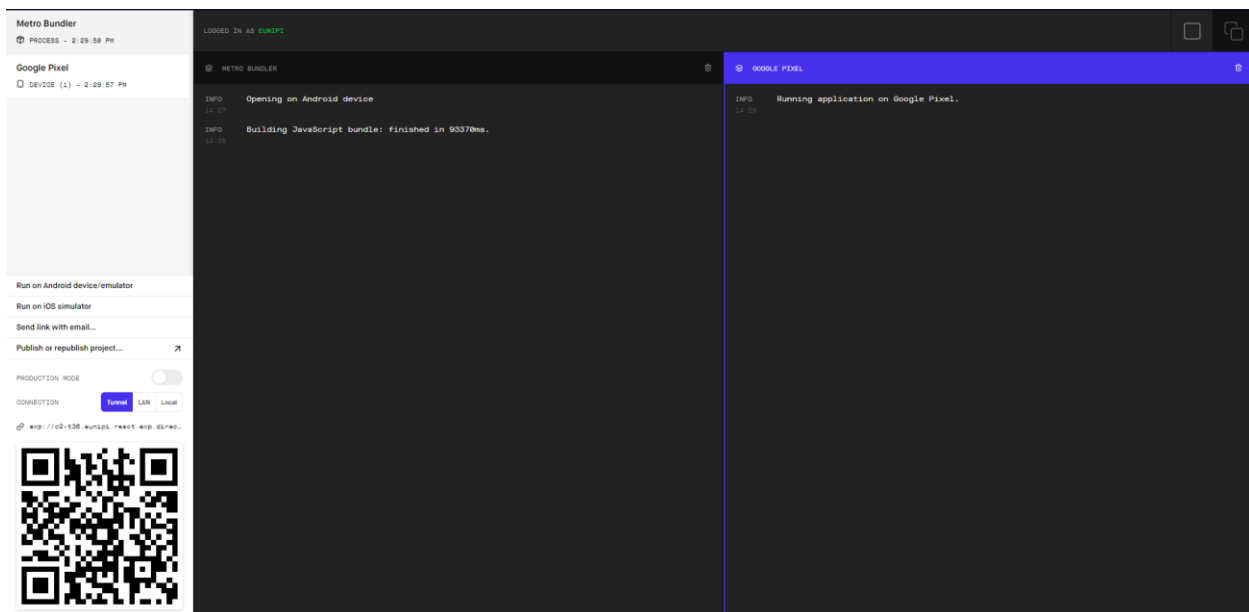
Εικόνα 47 Εφαρμογή Expo στο e-Store

Πλέον, οι φοιτητές είναι σε θέση να συνδεθούν στην εφαρμογή και να τη χρησιμοποιήσουν. Οι παρακάτω εικόνες της εφαρμογής έχουν παρθεί από τον emulator Genymotion [38], ένας ελαφρύς προσομοιωτής για Android κινητά. Το προσομοιωμένο κινητό που χρησιμοποιήθηκε είναι ένα Google Pixel με Android 8.0 API (Εικόνα 37). Την εφαρμογή την τρέξαμε τοπικά μέσω του Expo Builder (Εικόνα 38), μέσω του οποίου μπορούμε και να τη δημοσιεύσουμε (όπως και κάναμε) για δημόσια χρήση.

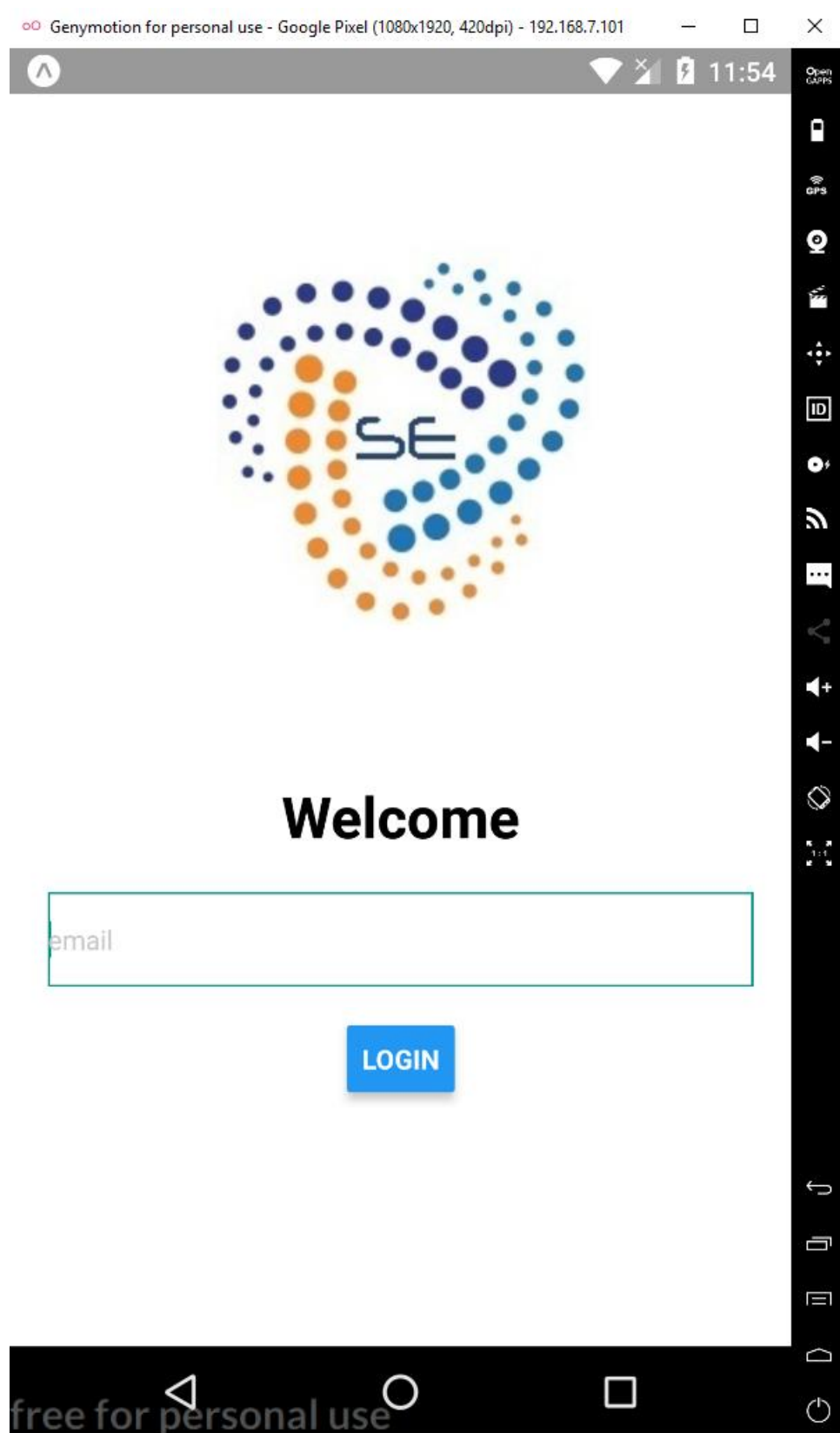
Η πρώτη οθόνη είναι αυτή που καλωσορίζει τους χρήστες και μέσω αυτής συνδέονται στην εφαρμογή. Επιλέχθηκε να μπορούν να συνδέονται οι φοιτητές με το e-mail τους απλά, καθώς δεν κρατούνται προσωπικά δεδομένα και θέλουμε να υπάρχει και ανωνυμία στις ψήφους, διότι αυτό έχει αποδειχθεί από έρευνες ότι ενισχύει τη συμμετοχή των φοιτητών [23] (Εικόνα 39).



Εικόνα 48 Genymotion




Εικόνα 49 Expo Builder



Εικόνα 50 Οθόνη Σύνδεσης

Αφού συνδεθεί ο χρήστης, οδηγείται στην κεντρική δραστηριότητα της εφαρμογής, όπου έχει 2 επιλογές (Εικόνα 40). Μπορεί να επιλέξει να αναζητήσει τα μαθήματα ή να αναζητήσει κατευθείαν ένα ερωτηματολόγιο.

Choose Class Or Search Poll



Search Classes



Search Poll

Εικόνα 51 Κεντρική Δραστηριότητα

Θα αναλύσουμε πρώτα, αν επιλέξει να αναζητήσει τα μαθήματα (Search Classes). Τότε, οδηγείται σε μία οθόνη που έχει τη λίστα με τα μαθήματα του συνδεδεμένου χρήστη (Εικόνα 41). Αν δεν υπάρχει κανένα στη λίστα, τότε μπορεί να επιλέξει να γραφεί σε μαθήματα από το κουμπί Register to Classes. Θα εμφανιστούν τότε όλα τα μαθήματα που δεν είναι γραμμένος (Εικόνα 42). Μπορεί να αναζητήσει το μάθημα που επιθυμεί, πληκτρολογώντας στη μπάρα αναζήτησης μερικά το όνομα του μαθήματος ή να κάνει scroll down στη λίστα. Με το πάτημα ενός μαθήματος-κουμπιού, αυτό προστίθεται στη λίστα του φοιτητή. Με το πάτημα το κόκκινου βέλους κάτω αριστερά οδηγείται στην προηγούμενη οθόνη. Αν επιλέξει ένα μάθημα από τη λίστα των μαθημάτων του εμφανίζεται η επόμενη οθόνη.

My Subjects

SaaS

Πρότυπα Ανάπτυξης Λογισμικού

Register to Classes



Εικόνα 52 Λίστα Επιλεγμένων Μαθημάτων

Search For Subject

Search

Select To Register

Web Programming



Εικόνα 53 Λίστα Μαθημάτων για Εγγραφή

Αφού επιλέξει κάποιο μάθημα από τη λίστα, εμφανίζεται η λίστα των καθηγητών που κάνουν το μάθημα (Εικόνα 43). Τώρα μπορεί να επιλέξει έναν από τους καθηγητές, που αντιστοιχεί στο τμήμα του φοιτητή και παρακολουθεί το μάθημα του και αν υπάρχει ενεργό ερωτηματολόγιο για αυτό το συνδυασμό μαθήματος-καθηγητή, θα οδηγηθεί σε αυτό. Σε διαφορετική περίπτωση θα εμφανιστεί στην οθόνη κατάλληλο μήνυμα που λέει ότι δεν είναι ενεργό κάποιο ερωτηματολόγιο. Έχει, επίσης την επιλογή να εξετάσει τα παλιά ερωτηματολόγια (Old Polls).

Πρότυπα Ανάπτυξης Λογισμικού

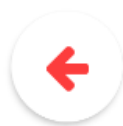
Select Teacher to go to Poll if Active

Sakkopoulos

Krasadakis

Check Old Polls for correct Answers

Old Polls



Εικόνα 54 Επιλογή Καθηγητή ή Παλιών Ερωτηματολογίων

Στα παλιά ερωτηματολόγια, βλέπει τη λίστα με τα ονόματά τους, σε σειρά με βάση τη θέση τους, όπως την έχει ορίσει ο καθηγητής και μπορεί να αναζητήσει με βάση το όνομα μέσα από τη μπάρα αναζήτησης (Εικόνα 44). Κατά την επιλογή ενός παλιού ερωτηματολογίου, εμφανίζεται η ερώτηση και η σωστή απάντηση του (Εικόνα 45).

Search For Poll

Choose Old Poll

Evaluation Design Patterns

Design Patterns 1-1

Design Patterns 1-2

Design Patterns 1-3

Design Patterns 1-4 Liskov



Εικόνα 55 Επιλογή Παλιών Ερωτηματολογίων

Question : Τι είναι η UML;

Answer : Σχεδιαγράμματα



Εικόνα 56 Απάντηση Παλιού Ερωτηματολογίου

Τώρα πάμε πίσω στην κεντρική σελίδα μέσα από τα κόκκινα βέλη, σε κάθε σελίδα κάτω αριστερά. Από την κεντρική σελίδα, επιλέγουμε τώρα να βρούμε ένα ερωτηματολόγιο κατευθείαν Search Poll. Με το πάτημα του κουμπιού εμφανίζεται στην οθόνη ένα πεδίο που ζητάει μόνο το Id του μαθήματος που δίνεται από τον καθηγητή (Εικόνα 46). Για κάποιον νέο χρήστη έχουμε και ένα δοκιμαστικό ερωτηματολόγιο που εξηγεί συνοπτικά τη λειτουργία της εφαρμογής (Εικόνα 47).

Add Poll Id

The Id will be given by the teacher

For first timers test 0-1

GO TO POLL



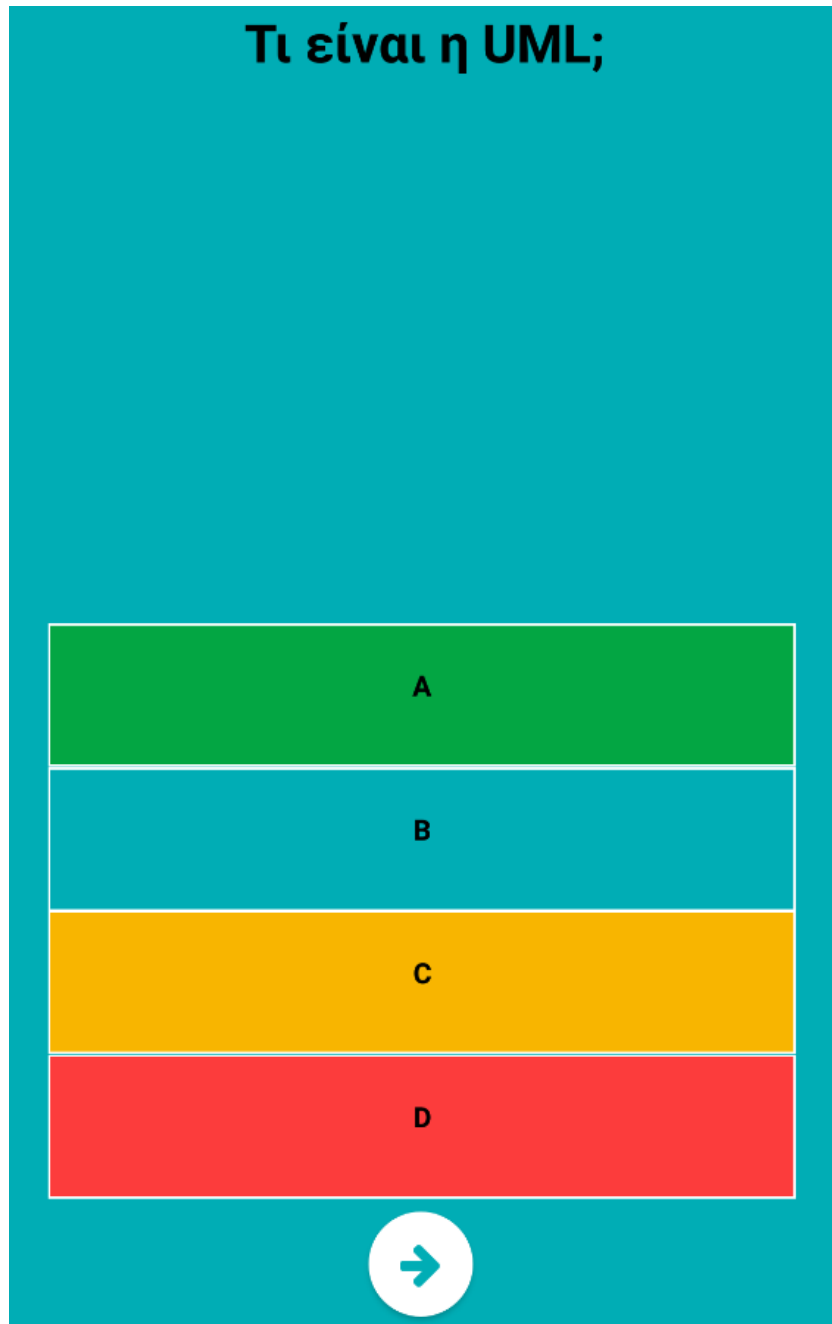
Εικόνα 57 Εύρεση Ερωτηματολογίου με βάση το Id

After Finding Poll by id or By Class and Teacher, you can answer the active poll (if there is one). The following buttons correspond to the answers. You can change your answer by simply selecting a different option.



Εικόνα 58 Ενδεικτικό Ερωτηματολόγιο

Στην οθόνη των ερωτηματολογίων εμφανίζεται πάνω η ερώτηση με bold γράμματα και από κάτω 4 κουμπιά με διαφορετικά χρώματα που αντιστοιχούν στις 4 απαντήσεις, που δείχνει ο καθηγητής στους φοιτητές (Εικόνα 48). Οι απαντήσεις έχουν τη μορφή A , B , C και D και όχι την πλήρη απάντηση, καθώς μέσα στα κουμπιά είναι δύσκολο μεγάλες απαντήσεις να χωρέσουν και να φαίνονται σωστά. Με την επιλογή κάποιας απάντησης η οθόνη παίρνει το χρώμα της επιλεγμένης απάντησης, ώστε να είναι προφανές για το φοιτητή τι έχει απαντήσει. Όσο είναι ενεργό το ερωτηματολόγιο μπορεί να αλλάξει την απάντηση του όσες φορές επιθυμεί. Αν απενεργοποιηθεί το ερωτηματολόγιο ή πατήσει το κουμπί στο κάτω μέρος της οθόνης οδηγείται στην τελευταία οθόνη.



Εικόνα 59 Ερωτηματολόγιο με επιλεγμένη απάντηση B

Στην τελική οθόνη εμφανίζεται μήνυμα που ευχαριστεί τους φοιτητές που συμμετείχαν στην διαδικασία και αναλόγως με το αν υπάρχει ενεργό ερωτηματολόγιο με το ίδιο id (συνδυασμό μαθήματος-καθηγητή) μπορούν να πάνε σε αυτό (Εικόνα 49) ή εμφανίζεται κατάλληλο μήνυμα(Εικόνα 50). Τέλος, μπορούν να πάνε πίσω στην αρχική σελίδα από το Back To Start, για να επιλέξουν άλλο μάθημα ή να δουν παλιές ερωτήσεις.

Thanks for Voting!



To Next Poll



Back To Start

Εικόνα 60 Τελική Οθόνη με Ενεργό Ερωτηματολόγιο

Thanks for Voting!

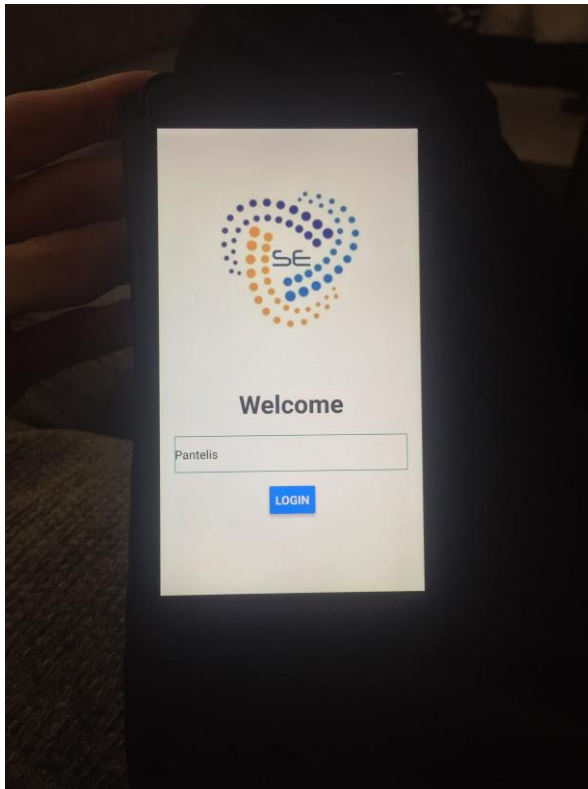
No Active Poll, please wait.

Back To Start

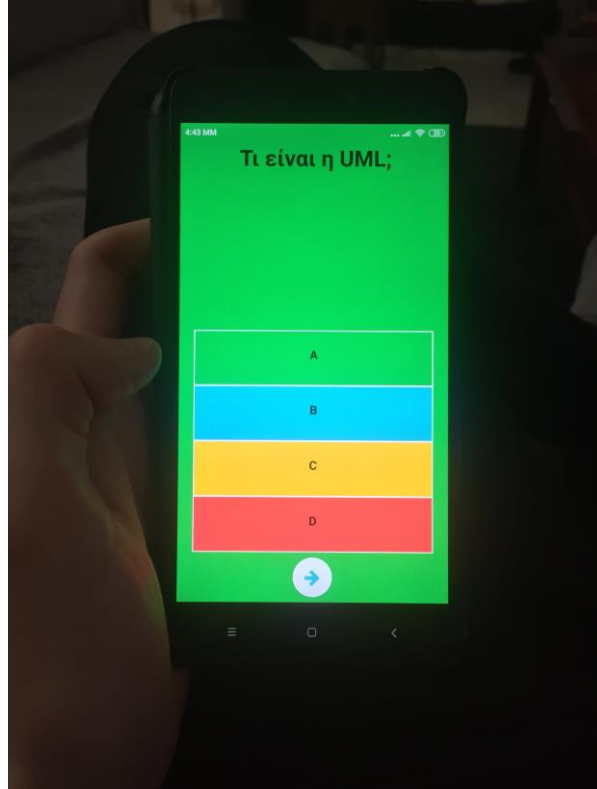
Εικόνα 61 Τελική Οθόνη με μη Ενεργό Ερωτηματολόγιο



Εικόνα 62 Οθόνη Ερωτηματολογίου σε iPhone



Εικόνα 63 Οθόνη Εισαγωγική σε Android



Εικόνα 64 Οθόνη Ερωτηματολογίου σε Android

5 Υλοποίηση

Στην ενότητα αυτή θα σχολιαστούν κάποια κομβικά σημεία της υλοποίησης μας.

5.1 React-Native

Η React-Native που χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής είναι ένα framework που επιτρέπει την ανάπτυξη native εφαρμογών για Android και iOS με τη χρήση μόνο JavaScript. Πιο συγκεκριμένα, χρησιμοποιεί πέρα από JS και τη JSX, ένα μείγμα JavaScript με XML συντακτικό. Αυτό επιτρέπει στην εύκολη και καλά δομημένη ανάπτυξη κώδικα. Ο κώδικας της JSX παραμένει καθαρά JavaScript, απλά έχει δημιουργηθεί το πακέτο για την απλοποίηση και προς διευκόλυνση των προγραμματιστών. Στην εικόνα 51 φαίνεται ένα απλό κομμάτι κώδικα του προγράμματος μας (ένα text input για τη εισαγωγή email και ένα κουμπί σύνδεσης) που αριστερά είναι όπως το γράψαμε σε JSX και δεξιά η μετατροπή του σε καθαρή JavaScript από το Babel compiler [39]. Πρέπει να σημειωθεί, επίσης, ότι λόγω αυτής της μορφής της React και JSX, η δομή του κώδικά μας είναι εμφωλευμένη, δηλαδή κάθε επόμενη οθόνη/λειτουργία είναι παιδί/απόγονος της αρχικής. Αυτό επιτυγχάνεται με τη χρήση μίας εγγενής μεταβλητής που παρέχει η react, της state, η οποία μπορεί να ενημερωθεί δυναμικά και να αλλάξει κατάσταση χωρίς να χρειάζεται να γίνει κάποιο refresh στη σελίδα. Στην εικόνα 52 παρουσιάζεται μέρος του κώδικα της αρχικής μας σελίδας της εφαρμογής (Εικόνα 39), για να δείξουμε τη γενική ιδέα του πως δομήσαμε το πρόγραμμα μας. Βλέπουμε το αντικείμενο της state που έχει οριστεί πάνω και πριν από όλες τις συναρτήσεις και τη μεταβλητή αυτής loggedIn. Με αυτή τη μεταβλητή ελέγχουμε αν ο χρήστης είναι συνδεδεμένος στο σύστημα ή όχι. Αν δεν είναι του εμφανίζεται η αρχική σελίδα με το Welcome και τη φόρμα εισαγωγής για το email του. Διαφορετικά, είναι συνδεδεμένος και άρα εμφανίζεται κατευθείαν η κεντρική σελίδα (επιστρέφεται να γίνει render η τάξη Class). Τέλος, τα styles κάθε σελίδας μπορούν να τοποθετηθούν στο τέλος, όλα μαζί, αλλά και δίπλα από το αντίστοιχο tag (Εικόνα 53).

```

<View style={styles.container}>
  <Text style={styles.headerStyle}> Welcome </Text>
  <TextInput
    style={styles.inputStyle}
    placeholder="email"
    // JS6 to change text on real time based on input
    onChangeText={({email})=>this.setState({email})}
    value={this.state.email}
  />
  <Button
    title="Login"
    // onclick react native prop function
    onPress={this.addUser.bind(this)}
  />
</View>
)

```

```

1 "use strict";
2
3 var _this = void 0;
4
5 React.createElement(View, {
6   style: styles.container
7 }, React.createElement(Text, {
8   style: styles.headerStyle
9 }, " Welcome "), React.createElement(TextInput, {
10  style: styles.inputStyle,
11  placeholder: "email" // JS6 to change text on real time based on input
12  ,
13  onChangeText: function onChangeText(email) {
14    return _this.setState({
15      email: email
16    });
17  },
18  value: (void 0).state.email
19 })), React.createElement(Button, {
20  title: "Login" // onclick react native prop function
21  ,
22  onPress: (void 0).addUser.bind(void 0)
23 }));

```

Εικόνα 65 Μετατροπή JSX σε JavaScript

```

1  export default class App extends React.Component{
2    // state allows to change dynamically variables during execution
3    state = {
4      loggedIn:false,
5      email:"",
6      student_id:""
7    }
8
9    // in renderscreen function we use the state to change dynamically the rendered screen
10   renderScreen = () =>{
11     if(this.state.loggedIn){
12       return(
13         // return Class component to show on screen
14         <Class student_id={this.state.student_id}/>
15       )
16     }else{
17       return(
18         <Text style={styles.headerStyle}> Welcome </Text>
19         <TextInput
20           style={styles.inputStyle}
21           placeholder="email"
22           // JS6 to change text on real time based on input
23           onChangeText={(email)=>this.setState({email})}
24           value={this.state.email}
25         />
26         <Button
27           title="Login"
28           // onclick react native prop function
29           onPress={this.addUser.bind(this)}
30         />
31       )
32     }
33   }
34 }
35 render(){
36   return (
37     <View style={styles.container}>
38       {this.renderScreen()}
39       <FlashMessage position="top" />
40     </View>
41   );

```

Εικόνα 66 Βασική Δομή Κώδικα

```

35   render(){
36     return (
37       <View style={styles.container}>
38         {this.renderScreen()}
39         <FlashMessage position="top" />
40       </View>
41     );
42
43     const styles = StyleSheet.create({
44       container: {
45         flex: 1,
46       },
47       inputStyle:{
48         alignSelf:'stretch',
49         borderColor:"#009688",
50         borderWidth:1,
51         margin:20,
52         height:50
53       },
54       headerStyle:{
55         alignSelf:'center',
56         justifyContent:'center',
57         fontSize:30,
58         fontWeight:'600',
59         marginTop:60
60       },
61     });

```

Εικόνα 67 StyleSheet

Στη συνέχεια θα περιγράψουμε τη διαδικασία εύρεσης και απάντησης των ερωτηματολογίων. Για την εύρεση των ερωτηματολογίων χρειαζόμαστε κατά βάση το id που αντιστοιχεί στο ενεργό poll (συνδυασμός id μαθήματος-καθηγητή). Η βασική δουλειά εδώ γίνεται για την εύρεση από την ασύγχρονη συνάρτηση findPoll (Εικόνα 54). Αυτή λαμβάνει από το state τη μεταβλητή poll_id και αφού την κάνει encode για το να μπορεί να την εισάγει σε URI καλεί το link προς το API και το server μας, για να ελέγξει αν υπάρχει και είναι ενεργό αυτό το id. Η χρήση του async μπροστά από τη συνάρτηση μας επιτρέπει μέσα σε αυτή να χρησιμοποιήσουμε τη λέξη-κλειδί await, που περιμένει την απάντηση από το server πριν συνεχίσει τα παρακάτω βήματα της συνάρτησης. Λειτουργεί, δηλαδή σαν τα promises της JavaScript, επιβεβαιώνοντας στον προγραμματιστή, ότι θα υπάρχουν τα απαραίτητα δεδομένα στα αναγκαία σημεία κώδικα. Συνεχίζοντας, σε περίπτωση σφάλματος (απάντηση Null) ή που δεν υπάρχει (length==0) ή που υπάρχει το ερωτηματολόγιο αλλά δεν είναι ενεργό (status=="inactive"), τότε εμφανίζεται στο πάνω μέρος τη οθόνης flash message με κόκκινο φόντο που ενημερώνει ό,τι είτε δεν υπάρχει το ερωτηματολόγιο είτε είναι ανενεργό. Διαφορετικά, καλεί τη συνάρτηση που έχει πάρει από το γονέα της μέσω των props, τη waitingEnded και θα οδηγηθεί στην οθόνη του ερωτηματολογίου για να απαντήσει.

```

1 // imports
2 import React from 'react';
3 // imported Components from react native
4 import {View,Text, TextInput,Button} from 'react-native';
5 import { showMessage, hideMessage } from "react-native-flash-message";
6
7 // created components
8 export default class FindPoll extends React.Component{
9   state={
10     poll_id:""
11   }
12   // async in front of function allows us to use await which makes function wait for the return of the call before continuing
13   async findPoll(){
14     const poll_id = encodeURIComponent(this.state.poll_id);
15     try{
16       let response = await fetch(`${myLink?id=${poll_id}`);
17       let responseJsonData = await response.json();
18       if (responseJsonData.status=="inactive" || responseJsonData==null || Object.keys(responseJsonData).length ==0){
19         showMessage({
20           message: "Poll Doesn't Exist Or Not Active",
21           description: "Try Again",
22           type: "danger",
23         });
24       }else{
25         this.props.waitingEnded(responseJsonData,this.state.poll_id);
26       }
27     }catch(error){
28       console.error(error);
29     }
30   }
31   render(){
32     return(
33       <View style={{flex:1}}>
34         <Text style={styles.textStyle}>Add Poll Id</Text>
35         <TextInput
36           style={styles.inputStyle}
37           placeholder="Poll Id"
38           // JS6 to change text on real time based on input
39           onChangeText={ (poll_id)=>this.setState({poll_id})}
40           value={this.state.poll_id}
41         />
42         <Button
43           title="Go to Poll"
44           // onclick react native prop function
45           onPress={this.findPoll.bind(this)}
46         />
47       </View>
48     </View>
49   )
50 }
51 }
52

```

Εικόνα 68 Class FindPoll

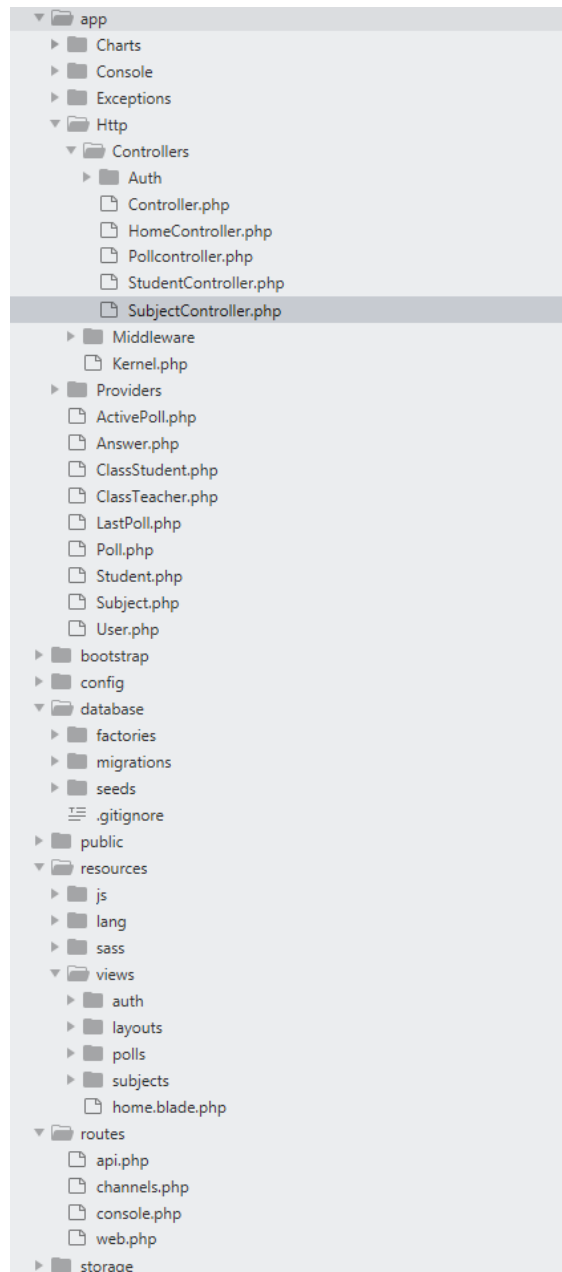
Στην οθόνη του ερωτηματολογίου (Εικόνα 47), με το πάτημα του κουμπιού παίρνουμε την απάντηση που έδωσε ο φοιτητής και την επεξεργαζόμαστε. Αυτό γίνεται από τη συνάρτηση `buttonPressed` (Εικόνα 55), που δέχεται σαν ορίσματα την απάντηση του κουμπιού, το χρώμα που αντιστοιχεί σε αυτή την απάντηση και το `id` του ερωτηματολογίου. Αρχικά, σπάμε το `id` του `poll` στα 2 που αντιστοιχούν στον καθηγητή και στο μάθημα, για να μπορέσουμε να ενημερώσουμε κατάλληλα τη βάση. Μετά, κάνουμε `POST`, την απάντηση, μαζί με τα `ids` του μαθήματος, του καθηγητή και του φοιτητή και περιμένουμε την απάντηση από το `server`. Αν είναι `OK`, τότε αλλάζουμε την οθόνη στο χρώμα της απάντησης, διαφορετικά σημαίνει ότι το ερωτηματολόγιο απενεργοποιήθηκε στο προηγούμενο χρονικό διάστημα, αφού συνδέθηκε ο φοιτητής και πριν από αυτή την απάντηση και του εμφανίζεται κατάλληλο μήνυμα και οδηγείται στην οθόνη που περιμένει να ενεργοποιηθεί το επόμενο `Poll`.

```
1  async buttonPressed(button, colour, poll){
2      let ids = this.state.poll_id.split("-");
3      try{
4          let response = await fetch('myLink/api/answer', {
5              method: 'POST',
6              headers: {
7                  Accept: 'application/json',
8                  'Content-Type': 'application/json',
9              },
10             body: JSON.stringify({
11                 data: {
12                     answer: button,
13                     poll: poll.id,
14                     teacher: ids[1],
15                     student: this.props.student_id
16                 }
17             })),
18         });
19         let responseJson = await response.json();
20         if(responseJson==0){
21             showMessage({
22                 message: "Poll Not Active Anymore",
23                 description: "Wait For Next",
24                 type: "danger",
25             });
26             this.setState({colour: "#FFFFFF"});
27             this.setState({currentScreen: "pollEnded"});
28         }else{
29             this.setState({colour: colour});
30         }
31     }catch(error){
32         console.error(error);
33     }
34 }
```

Εικόνα 69 Συνάρτηση `buttonPressed`

5.2 Laravel

Για τη μεριά του εξυπηρετητή (server), χρησιμοποιήθηκε η Laravel, ένα Model-View-Controller framework της PHP. Με τη δημιουργία ενός νέου project Laravel, υπάρχει ήδη έτοιμη αυτή η δομή του MVC μοντέλου, που κάνει πολύ πιο ξεκάθαρη τη διαφοροποίηση των στοιχείων μεταξύ τους και άρα πολύ πιο κατανοητή τη δομή και συντήρηση του κώδικα (Εικόνα 56). Τα μοντέλα αντιστοιχούν σε πίνακες στη βάση και μέσω αυτών μπορούμε πολύ εύκολα να κάνουμε SQL ερωτήματα πάνω σε αυτά ή να τα συσχετίσουμε μεταξύ τους (Εικόνα 57). Τα views, αντιστοιχούν στις PHP/HTML σελίδες που θα εμφανίζονται στο χρήστη και χρησιμοποιείται το blade template της PHP. Οι Controllers, τέλος είναι αυτοί που υλοποιούν όλες τις λειτουργίες του server μας. Όλα αυτά επικοινωνούν μεταξύ τους με το Eloquent (για τα μοντέλα και τη βάση) και το Routing (Εικόνα 58) (για τους Controllers και τις διευθύνσεις).



Εικόνα 70 Δομή MVC της Laravel

```

1  <?php
2
3  namespace App;
4
5  use Illuminate\Database\Eloquent\Model;
6  use App\Subject;
7
8  class Poll extends Model
9  {
10
11     public function subject(){
12         return $this->belongsTo(Subject::class);
13     }
14 }
15

```

Εικόνα 71 Μοντέλο του Poll

```

1  <?php
2
3  use Illuminate\Http\Request;
4
5  /*
6  |-----
7  | API Routes
8  |-----
9  |
10 | Here is where you can register API routes for your application. These
11 | routes are loaded by the RouteServiceProvider within a group which
12 | is assigned the "api" middleware group.
13 |
14 */
15
16 Route::middleware('auth:api')->get('/user', function (Request $request) {
17     return $request->user();
18 });
19
20 Route::post('student', 'StudentController@login');
21 Route::get('subjects', 'StudentController@subjects');
22 Route::get('teachers', 'StudentController@teachers');
23 Route::get('myClasses', 'StudentController@myClasses');
24 Route::get('poll', 'StudentController@poll');
25 Route::get('oldPolls', 'StudentController@oldPolls');
26 Route::post('answer', 'StudentController@answer');
27 Route::post('registerToClass', 'StudentController@registerToClass');

```

Εικόνα 72 API Routing

Τώρα, θα αναλύσουμε πως δέχεται και επεξεργάζεται ο server τις απαντήσεις των φοιτητών. Θα δούμε τις συναρτήσεις που υπάρχουν στους Controllers, StudentController_answer και PollController_answer. Ο πρώτος Controller είναι υπεύθυνος για όλες τις ενέργειες που αφορούν τους φοιτητές και άρα είναι αυτός που επεξεργάζεται σε πραγματικό χρόνο τις απαντήσεις τους. Αρχικά, δέχεται το Post Request (Εικόνα 55) και μέσω αυτού λαμβάνει το id των poll, καθηγητή και φοιτητή. Πρώτα γίνεται έλεγχος για αν είναι ακόμα ενεργό αυτό το ερωτηματολόγιο, γιατί μπορεί να έχει απενεργοποιηθεί από το χρόνο ή τον καθηγητή. Αν δεν είναι ενεργό επιστρέφει 0, ώστε να δει την κατάλληλη απάντηση ο φοιτητής και να οδηγηθεί στην επόμενη οθόνη. Διαφορετικά, η διαδικασία συνεχίζει, με το πρόγραμμα να ελέγχει αν υπάρχει ήδη απάντηση του φοιτητή καταχωρημένη στη βάση. Αν υπάρχει, έχουμε θεωρήσει από πριν, ότι σε περίπτωση που το ίδιο ερωτηματολόγιο ενεργοποιηθεί από τον ίδιο καθηγητή σε διαφορετική μέρα σημαίνει ότι θέλει να το επανεξετάσει και άρα χρειάζεται να προστεθούν ξανά όλες οι απαντήσεις των φοιτητών στη βάση. Διαφορετικά, θεωρούμε ότι είναι το ίδιο ερωτηματολόγιο και άρα οι απαντήσεις τους γίνονται update, πάνω στις προηγούμενες καθώς σημαίνει ότι απλά ψήφισαν για να αλλάξουν απάντηση (Εικόνα 59).

```

1  <?php
2  public function answer(Request $request)
3  {
4      // check if poll that is being answered is still active
5      $activePoll = ActivePoll::where('poll_id',$request->data["poll"])->first();
6      if($activePoll->status=="active"){
7          // check if student has already an answer registered in the subject-teacher combination
8          $answer = Answer::where(['poll_id' => $request->data["poll"], 'teacher_id' => $request->data["teacher"], 'student_id' => $request->data["student"]])->latest()->first();
9          if($answer){
10             // if there is an answer in db already we check when it was created.
11             $created = new Carbon($answer->created_at);
12             $now = Carbon::now();
13             // we have set a default difference in one day
14             if($created->diff($now)->days < 1){
15                 // if the dif. is less than a day it means that the student is answering the same poll again, so we update answer
16                 $answer->answer = $request->data["answer"];
17                 $answer->save();
18             }else{
19                 // else we assume that the teacher has activated on a different occasion the same poll for the same class
20                 // and wants to retest the students so we create a new answer entry
21                 $answer = new Answer;
22                 $answer->poll_id = $request->data["poll"];
23                 $answer->teacher_id = $request->data["teacher"];
24                 $answer->student_id = $request->data["student"];
25                 $answer->answer = $request->data["answer"];
26                 $answer->save();
27             }
28         }else{
29             // if no answer exists we create it
30             $answer = new Answer;
31             $answer->poll_id = $request->data["poll"];
32             $answer->teacher_id = $request->data["teacher"];
33             $answer->student_id = $request->data["student"];
34             $answer->answer = $request->data["answer"];
35             $answer->save();
36         }
37         return response()->json("ok");
38     }else{
39         // if poll not active return 0
40         return response()->json(0);
41     }
42 }

```

Εικόνα 73 Διαχείριση Απαντήσεων φοιτητών από StudentController

Ο PollController είναι υπεύθυνος για την προσθήκη και διαχείριση των ερωτηματολογίων. Η PollController_answer καλείται όταν ο καθηγητής πατήσει το κουμπί Check Answers (Εικόνα 33), για να συλλέξει τις απαντήσεις των φοιτητών στο τέλος της ερώτησης. Ξεκινάει με το να βεβαιωθεί και να απενεργοποιήσει το ερωτηματολόγιο, καθώς δεν επιθυμούμε να μπορούν να απαντούν οι φοιτητές, αφού δουν τις απαντήσεις και προσθέτει το ερωτηματολόγιο αυτό στη λίστα των τελευταίων Poll και στο αντίστοιχο session, για να εμφανίζεται σωστά η λίστα δεξιά της σελίδας, που περιέχει τα τελευταία ερωτηματολόγια ανά μάθημα για την εξυπηρέτηση των καθηγητών. Στη συνέχεια, παίρνει όλες τις μέρες που το ερωτηματολόγιο αυτό έχει απαντηθεί. Ελέγχουμε αν ο χρήστης έχει διευκρινίσει, για ποια μέρα θέλει να δει τις απαντήσεις. Αν δεν υπάρχει τέτοια επιλογή (όπως κατά την αρχική μετάβαση στη σελίδα των απαντήσεων), κρατάμε μόνο τις τελευταίες χρονικά απαντήσεις, καθώς αυτές είναι που μόλις έγιναν μέσα στην τάξη. Αν ο καθηγητής έχει επιλέξει από το drop down menu (Εικόνα 34), κάποια ημερομηνία ή το συνολικό αποτέλεσμα παίρνουμε το αντίστοιχο. Έχοντας τις απαντήσεις, φτιάχνουμε το γράφημα βάζοντας τες στις αντίστοιχες θέσεις με τα αντίστοιχα χρώματα και τα εμφανίζουμε.

```

1  <?php
2  public function answer(Request $request)
3  {
4      $user = $request->user()->id;
5      $poll = Poll::where('id',$request->poll)->first();
6      // get active poll and deactivate it
7      $activepoll = ActivePoll::where(['poll_id'=>$request->poll,'teacher_id'=>$user])->first();
8      $activepoll->status = 'inactive';
9      $activepoll->save();
10     // add this poll to last poll List for sidebar
11     $lastPolls = LastPoll::where('teacher_id',$user)->get();
12     session(['last_polls' => $lastPolls]);
13     // group answers of poll by date
14     $dates=Answer::select(DB::raw('DATE(created_at) as date'))
15     ->where(['poll_id'=>$request->poll,'teacher_id'=>$user])
16     ->groupBy('date')->get()->toArray();
17     // check if user want a specific date or all, by default we return the last date
18     if($request->date=='All' || count($dates)==0){
19         $answerA = Answer::where(['poll_id'=>$request->poll,'teacher_id'=>$user,'answer'=>1])->count();
20         $answerB = Answer::where(['poll_id'=>$request->poll,'teacher_id'=>$user,'answer'=>2])->count();
21         $answerC = Answer::where(['poll_id'=>$request->poll,'teacher_id'=>$user,'answer'=>3])->count();
22         $answerD = Answer::where(['poll_id'=>$request->poll,'teacher_id'=>$user,'answer'=>4])->count();
23         $selectedDate = "all";
24     }else {
25         if ($request->date==null){
26             // get answers of last date
27             $date = $dates[count($dates)-1]['date'];
28             $answerA = Answer::where(['poll_id'=>$request->poll,'teacher_id'=>$user,'answer'=>1])
29             ->where(DB::raw("(STR_TO_DATE(created_at,'%Y-%m-%d')"), $date)->count();
30             $answerB = Answer::where(['poll_id'=>$request->poll,'teacher_id'=>$user,'answer'=>2])
31             ->where(DB::raw("(STR_TO_DATE(created_at,'%Y-%m-%d')"), $date)->count();
32             $answerC = Answer::where(['poll_id'=>$request->poll,'teacher_id'=>$user,'answer'=>3])
33             ->where(DB::raw("(STR_TO_DATE(created_at,'%Y-%m-%d')"), $date)->count();
34             $answerD = Answer::where(['poll_id'=>$request->poll,'teacher_id'=>$user,'answer'=>4])
35             ->where(DB::raw("(STR_TO_DATE(created_at,'%Y-%m-%d')"), $date)->count();
36             $selectedDate = $date ;
37         }else{
38             // get answers of selected date by user
39             $answerA = Answer::where(['poll_id'=>$request->poll,'teacher_id'=>$user,'answer'=>1])
40             ->where(DB::raw("(STR_TO_DATE(created_at,'%Y-%m-%d')"), $request->date)->count();
41             $answerB = Answer::where(['poll_id'=>$request->poll,'teacher_id'=>$user,'answer'=>2])
42             ->where(DB::raw("(STR_TO_DATE(created_at,'%Y-%m-%d')"), $request->date)->count();
43             $answerC = Answer::where(['poll_id'=>$request->poll,'teacher_id'=>$user,'answer'=>3])
44             ->where(DB::raw("(STR_TO_DATE(created_at,'%Y-%m-%d')"), $request->date)->count();
45             $answerD = Answer::where(['poll_id'=>$request->poll,'teacher_id'=>$user,'answer'=>4])
46             ->where(DB::raw("(STR_TO_DATE(created_at,'%Y-%m-%d')"), $request->date)->count();
47             $selectedDate = $request->date ;
48         }
49     }
50     // create bar chart with A, B, C, D and corresponding colours
51     $chart = new SampleChart;
52     $chart->labels(['A', 'B', 'C', 'D']);
53     $chart->dataset('Answers', 'bar', [$answerA, $answerB, $answerC, $answerD])->options(['backgroundColor'=>["#03A643", "#00ADB5", "#F8B500", "#FC3C3C"]]);
54     return view('polls/answer',compact("poll", "dates", "chart", "selectedDate"));
55 }

```

Εικόνα 74 Συλλογή Απαντήσεων φοιτητών από PollController

6 Αποτίμηση

6.1 Πειραματικά Δεδομένα

Το σύστημά μας δοκιμάστηκε και έτρεξε σε πραγματικές συνθήκες σε δύο διαφορετικές τάξεις, μία στο Πανεπιστήμιο Πειραιά για το μάθημα Πρότυπα Ανάπτυξης Λογισμικού έκτου εξαμήνου με 5 συμμετέχοντες προπτυχιακούς φοιτητές και μια στο Ανοικτό Ελληνικό Πανεπιστήμιο για το μάθημα Εξόρυξη Δεδομένων με 10 μεταπτυχιακούς φοιτητές. Η διαδικασία που ακολουθήθηκε και στις 2 περιπτώσεις ήταν η παρακάτω.

Αρχικά, ο καθηγητής είχε ήδη συνδεθεί στο σύστημα μας και είχε δημιουργήσει τα μαθήματα. Μέσα στα μαθήματα στη συνέχεια πρόσθεσε τις ερωτήσεις πολλαπλής επιλογής που επιθυμούσε να ρωτήσει κατά τη διάρκεια του αντίστοιχου μαθήματος. Είχε προσθέσει, παράλληλα τις αντίστοιχες ερωτήσεις στις διαφάνειες του μαθήματος του για να είναι ομαλή η μετάβαση από τις διαφάνειες στη σελίδα μας και στην ενεργοποίηση της ερώτησης. Ξεκινώντας στο μάθημα παρουσιάστηκαν σύντομες οδηγίες εγκατάστασης και χρήσης της εφαρμογής στους φοιτητές (Εικόνα 51). Όλοι οι φοιτητές κατάφεραν να κατεβάσουν και να χρησιμοποιήσουν την εφαρμογή, ανεξαρτήτως κινητού τηλεφώνου και λογισμικού (Android/iOS).

Ο κάθε φοιτητής αφού κατέβασε την εφαρμογή Expro αναζητά μέσω αυτής το eUnipi/Elearning για να κατεβάσει και να μπορεί να συνδεθεί στη δική μας εφαρμογή. Στη συνέχεια συνδέθηκαν με το e-mail τους και περιηγήθηκαν λίγο στην εφαρμογή για να εξοικειωθούν και να την αξιολογήσουν. Έχοντας ξεκινήσει το μάθημα ο καθηγητής στην πορεία της διάλεξης, όταν φτάσει στην ερώτηση που επιθυμεί να θέσει στο σύνολο των φοιτητών τους ζητάει να ανοίξουν τα κινητά και να πάνε στο ερωτηματολόγιο με το Id που φαίνεται στον προβολέα. Οι φοιτητές εισάγουν το Id στην οθόνη (Εικόνα 46) και είναι έτοιμοι να απαντήσουν στο ερώτημα. Οι χρήστες απαντούσαν γρήγορα στις ερωτήσεις και παρατηρήθηκε ότι οι περισσότεροι χρησιμοποιούσαν το κουμπί για να συνεχίσουν πριν τελειώσει ο χρόνος που τους έδινε ο καθηγητής. Μπορούσαν όμως, όσο είναι ανοιχτή η ψηφοφορία να αλλάξουν την απάντησή τους. Με το τέλος της ψηφοφορίας, ο καθηγητής εμφάνιζε το γράφημα με τις απαντήσεις (Εικόνα 34), με αποτέλεσμα και οι φοιτητές αλλά και ο καθηγητής να βλέπουν το ποσοστό κατανόησης της ερώτησης. Αναλόγως, με τις απαντήσεις είτε υπήρχε περαιτέρω συζήτηση επί του θέματος (αν δεν ήταν σε ικανοποιητικό ποσοστό σωστές οι απαντήσεις) είτε το μάθημα συνεχιζόταν.

Στο τέλος των μαθημάτων ζητήσαμε από τους φοιτητές να αξιολογήσουν την εφαρμογή μας με βάση κάποιες ερωτήσεις που είχαμε ετοιμάσει. Οι ερωτήσεις ήταν οι εξής :

1. Πώς θα βαθμολογούσατε την προσθήκη αυτής της εφαρμογής στην εκπαιδευτική διαδικασία
2. Πώς θα βαθμολογούσατε την ευκολία χρήσης της εφαρμογής
3. Κατά πόσο βοήθησε η εφαρμογή στη διευκρίνιση ερωτήσεων κατά τη διάρκεια του μαθήματος
4. Κατά πόσο βοήθησε η εφαρμογή στην περαιτέρω συζήτηση θεμάτων στην τάξη

Οι απαντήσεις της μορφής Α, Β, Γ, Δ ήταν αντίστοιχα, Τέλεια, Καλά, Μέτρια, Λίγο. Τα αποτελέσματα της αξιολόγησης παρατίθενται στους παρακάτω πίνακες.

Εφαρμογή E-Learning

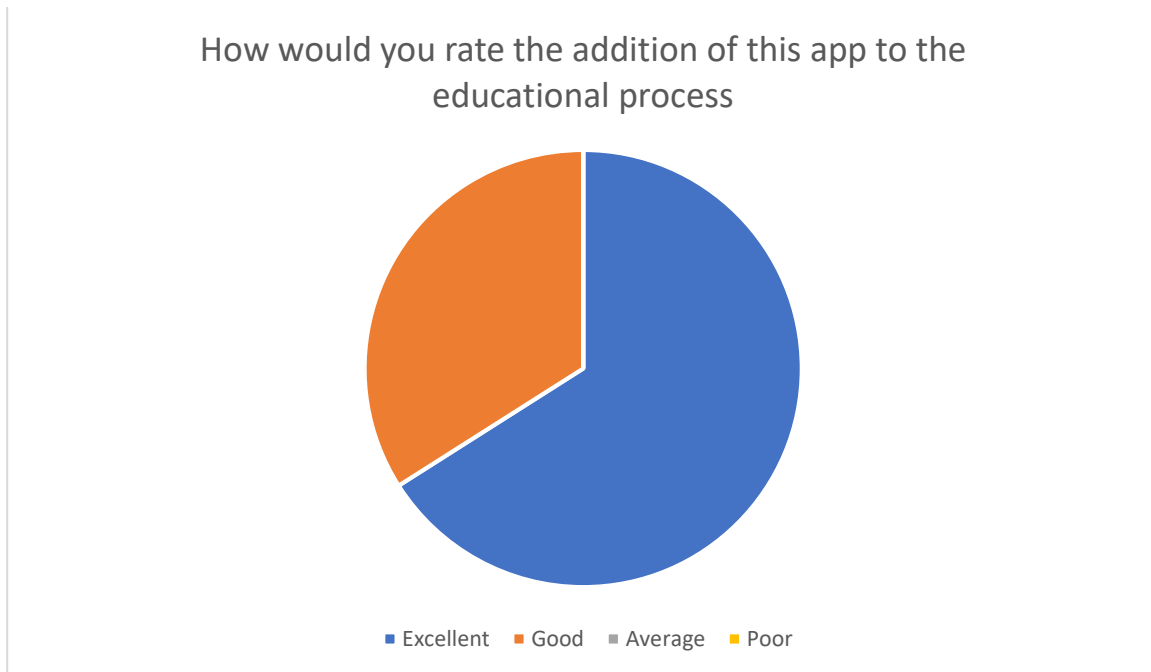
Εγκατάσταση

- 1) Κατεβάζετε την εφαρμογή Expo από Google/i Store
- 2)
 - a. Android: Αναζητήστε eUnipi/eLearning
 - b. Ios: Συνδεθείτε ως χρήστης eUnipi με password :

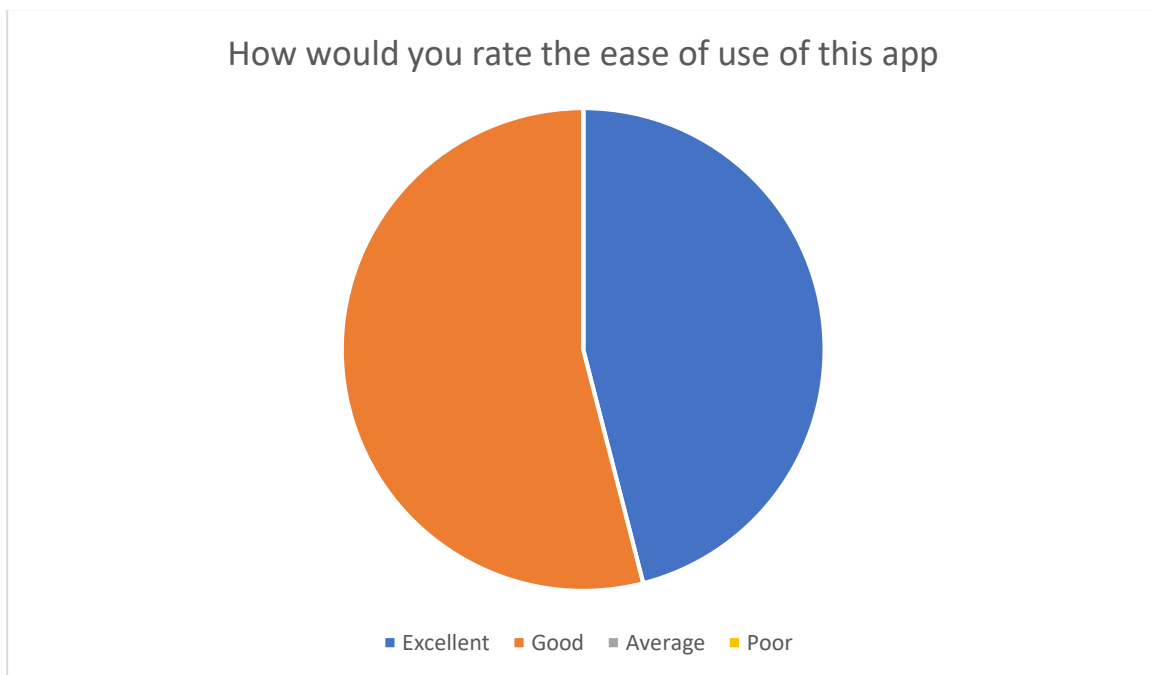
Χρήση

- 1) Συνδέεστε με το e-mail σας στην αρχική σελίδα της εφαρμογής
- 2) Επιλογές α) Αναζήτηση Μαθημάτων β) Αναζήτηση Ερώτησης:
 - A) Οδηγείστε σε οθόνη με τα μαθήματα που είστε γραμμένοι
 - i) Αν δεν έχετε κανένα μάθημα στη λίστα πηγαίνετε στην επιλογή εγγραφή σε μαθήματα.
 - ii) Σε αυτή την οθόνη επιλέγετε τα μαθήματα στα οποία θέλετε να γραφείτε
 - iii) Αφού έχετε γραφεί στα μαθήματα που επιθυμείτε και επιστρέψετε στην προηγούμενη οθόνη, επιλέγετε ένα από τα μαθήματα
 - iv) Εμφανίζεται λίστα με τους διδάσκοντες του μαθήματος και επιλέγοντας έναν από αυτούς αν έχει ανοιχτή κάποια ερώτηση για αυτό το μάθημα, πηγαίνετε σε αυτή
 - v) Διαφορετικά, μπορείτε να επιλέξετε τις παλιές ερωτήσεις και να δείτε τις σωστές απαντήσεις τους
 - B) i) Οδηγείστε σε οθόνη όπου ζητείται ο κωδικός της ερώτησης που θέλετε να απαντήσετε(δίνεται από τον καθηγητή)
 - ii) Αν είναι ενεργή η ερώτηση πηγαίνετε σε αυτή
- 3) Στην οθόνη των ερωτήσεων εμφανίζεται η ερώτηση πάνω και τα κουμπιά A, B, C, D που αντιστοιχούν στις απαντήσεις που φαίνονται στις διαφάνειες του καθηγητή.
- 4) Όσο είναι ενεργή η ερώτηση επιλέγετε την απάντηση σας και μπορείτε να την αλλάξετε
- 5) Αν απενεργοποιηθεί η ερώτηση ή επιλέξετε να πάτε παρακάτω με το κουμπί, μπορείτε είτε να επιστρέψετε στην αρχική οθόνη επιλογών (2) είτε να πάτε στην επόμενη ερώτηση εφόσον αυτή είναι ενεργή

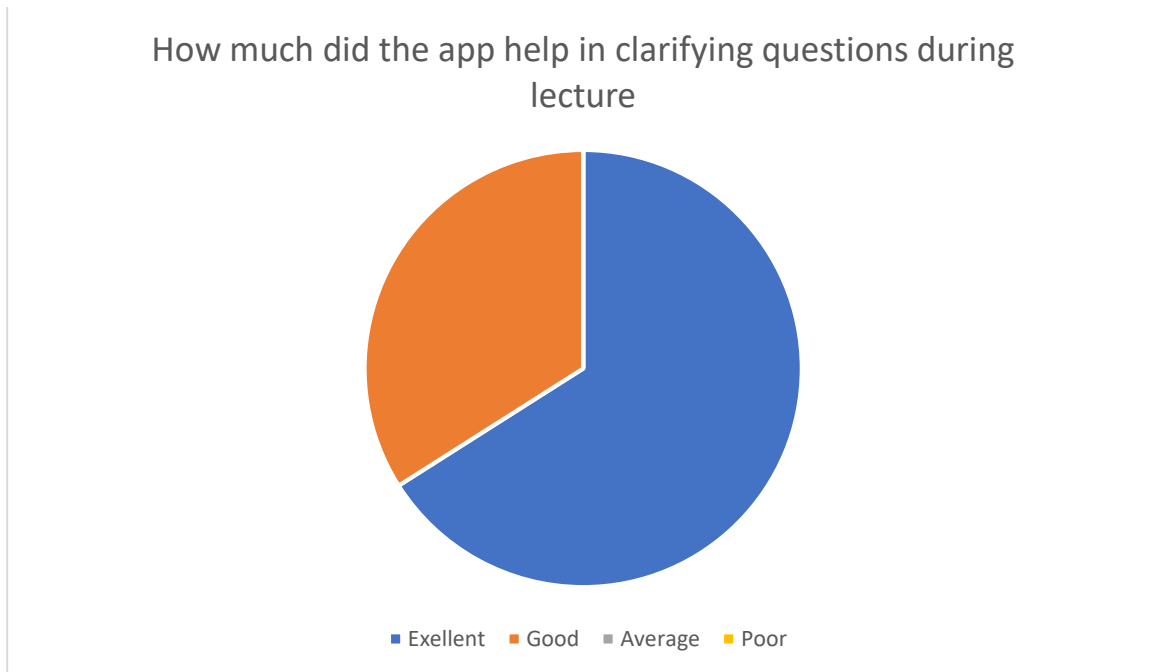
Εικόνα 75 Οδηγίες Εγκατάστασης/Χρήσης



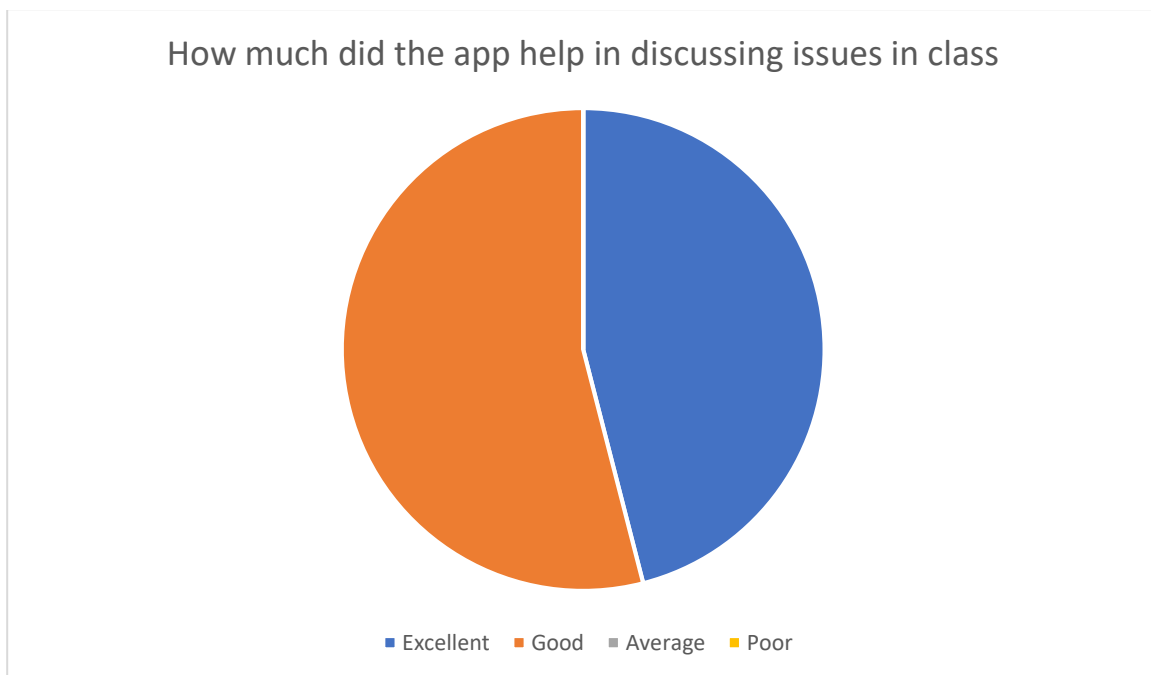
Γράφημα 1



Γράφημα 2



Γράφημα 3



Γράφημα 4

Πίνακας 1 Συνολικά Ποσοστά

Q/A	Excellent	Good	Average	Poor
How would you rate the addition of this app to the educational process	66%	34%	0	0
How would you rate the ease of use of this app	46%	54%	0	0
How much did the app help in clarifying questions during lecture	66%	34%	0	0
How much did the app help in discussing issues in class	46%	54%	0	0

Πίνακας 2 Ποσοστά Προπτυχιακών

Q/A	Excellent	Good	Average	Poor
How would you rate the addition of this app to the educational process	60%	40%	0	0
How would you rate the ease of use of this app	0%	100%	0	0
How much did the app help in clarifying questions during lecture	40%	60%	0	0
How much did the app help in discussing issues in class	60%	40%	0	0

Πίνακας 3 Ποσοστά Μεταπτυχιακών

Q/A	Excellent	Good	Average	Poor
How would you rate the addition of this app to the educational process	70%	30%	0	0
How would you rate the ease of use of this app	70%	30%	0	0
How much did the app help in clarifying questions during lecture	80%	20%	0	0
How much did the app help in discussing issues in class	40%	60%	0	0

6.2 Συμπεράσματα

Αναλύοντας τα παραπάνω μπορούμε να εξάγουμε τα συμπεράσματα μας και κατά πόσο επιτεύχθηκαν οι στόχοι μας. Από τους πίνακες φαίνεται, ότι η εφαρμογή μας είχε θετική ανταπόκριση από τους φοιτητές, καθώς κανένας δεν απάντησε για καμία ερώτηση «Μέτρια» και «Λίγο». Οι μεταπτυχιακοί φοιτητές, επίσης φαίνεται να είναι πιο εξοικειωμένοι με τέτοιες τεχνολογίες και να τους φάνηκε πιο εύκολη προς χρήση η εφαρμογή μας. Σε γενικό βαθμό, όμως, η εφαρμογή ήταν κατανοητή και δεν παρουσιάστηκε κανένα πρόβλημα στη χρήση της και όλοι οι φοιτητές είναι θετικοί στην προοπτική της τακτικής εισαγωγής της στα μαθήματα τους. Θεωρούν ότι βελτιώνει την εκπαιδευτική διαδικασία, κάνοντας το μάθημα πιο «σύγχρονο» και άρα ενδιαφέρον για αυτούς και ένα προσιτό τρόπο επικοινωνίας με τον καθηγητή, χωρίς τον κίνδυνο της έκθεσης προς τους συμφοιτητές τους. Τόνισαν, τέλος, οι φοιτητές ότι προτιμούν να υπάρχει ανωνυμία στη ψηφοφορία, για να αισθάνονται ασφάλεια να απαντήσουν.

Από τις παρατηρήσεις που κάναμε κατά τη διάρκεια των μαθημάτων, την αξιολόγηση των φοιτητών αλλά και τα σχόλια των διδασκόντων, τα θετικά και τα αρνητικά του συστήματός μας έχουν ως εξής:

Θετικά:

- Δεν χρειάζεται ειδικό hardware, αρκεί ένα κινητό τηλέφωνο, που σχεδόν όλοι οι φοιτητές πλέον έχουν
- Είναι απλό και εύκολο να χρησιμοποιηθεί και το απλό User Interface δεν αποσπά την προσοχή των φοιτητών.
- Ενισχύει την ενεργή συμμετοχή και αλληλεπίδραση των φοιτητών τόσο μεταξύ τους όσο και με τους καθηγητές.
- Βελτιώνει το ενδιαφέρον προς το μάθημα και την εκπαιδευτική διαδικασία για τους φοιτητές.
- Είναι πιο εύκολο για όλους τους φοιτητές να συμμετέχουν στο μάθημα.
- Οι καθηγητές έχουν άμεση πρόσβαση σε ανατροφοδότηση από τους φοιτητές και μπορούν να προσαρμόζουν το μάθημα τους αναλόγως.
- Οι καθηγητές έχουν επίσης πρόσβαση σε ιστορικό των ερωτήσεων, για να συγκρίνουν στο πέρασμα του χρόνου αν υπάρχει βελτίωση της τάξης ή αν υπάρχουν συγκεκριμένες ερωτήσεις που δυσκολεύουν διαχρονικά τους φοιτητές και να σχεδιάσουν ή επανεξετάσουν τις ερωτήσεις.
- Οι φοιτητές μπορούν να ξαναδούν παλαιότερα ερωτηματολόγια και τις σωστές απαντήσεις αυτών, με αποτέλεσμα να αυξάνεται η κατανόηση των θεμάτων και πιθανά να έχει θετική επίπτωση και στις εξετάσεις τους.
- Το σύστημα μπορεί να χρησιμοποιηθεί τόσο για συμβατικές τάξεις, όσο για τάξεις εξ αποστάσεως ή μεικτές. Μπορεί ακόμα να χρησιμοποιηθεί σε άλλες αντίστοιχες δομές όπως συνέδρια.

Αρνητικά:

- Την πρώτη περίοδο εφαρμογής του συστήματος αυξάνεται ο φόρτος εργασίας των καθηγητών καθώς πρέπει να προσαρμόσουν το μάθημα και τις διαφάνειες τους για να εισάγουν τις ερωτήσεις.
- Η εφαρμογή δεν λειτουργεί χωρίς πρόσβαση σε διαδίκτυο (αλλά όπως αναφέραμε, συνηθίζεται τα πανεπιστημιακά ιδρύματα να παρέχουν δωρεάν στους φοιτητές τους).
- Η χρήση κινητών τηλεφώνων μπορεί να χαλάσει τη ροή του μαθήματος ή να αποσπάσει την προσοχή των φοιτητών.
- Εξαιτίας της πιο διαδραστικής φύσης του μαθήματος, μέσω του συστήματος μας, πιθανά η ποσότητα ύλης που να καλύπτεται ανά διάλεξη να μειώνεται, αλλά το βάθος των θεμάτων που συζητούνται αυξάνεται
- Η δομή των ερωτήσεων είναι πολλαπλής επιλογής με μία μόνο σωστή απάντηση, οπότε περιορίζεται το εύρος των δυνατών ερωτήσεων που μπορεί να θέσει ο καθηγητής.

7 Επίλογος

Τα τελευταία χρόνια έχει παρατηρηθεί μεγάλη αύξηση στην αναζήτηση και έρευνα νέων τρόπων εισαγωγής ηλεκτρονικών εκπαιδευτικών λογισμικών στους πανεπιστημιακούς χώρους. Αυτοί έχουν στόχο να αξιοποιήσουν την άνοδο της τεχνολογίας και της χρήσης της καθημερινά από όλους τους φοιτητές, για να καλύψουν τα κενά ή τις αδυναμίες που παρουσιάζονται στη συμβατική εκπαίδευση. Η έρευνα, πλέον επικεντρώνεται στη χρήση των κινητών τηλεφώνων ως ένα μέσο συμμετοχής των φοιτητών στην εκπαιδευτική διαδικασία προσπαθώντας να δώσουν μία νέα οπτική στη διδασκαλία των φοιτητών.

Το σύστημα που αναπτύξαμε για την παρούσα διατριβή είναι ακόμα νέο και εξελίσσεται σαν εκπαιδευτικό εργαλείο, αλλά οι βασικοί στόχοι που θέσαμε στο στάδιο της σχεδίασης (με βάση τα αποτελέσματα και την αξιολόγηση) επιτεύχθηκαν. Καταφέραμε να εισάγουμε μία εφαρμογή εκπαιδευτικού περιεχομένου για κινητά τηλέφωνα σε μία τάξη πανεπιστημίου, με εύκολη πρόσβαση από όλους τους φοιτητές, με μηδενικό κόστος τόσο για αυτούς όσο και για το Εκπαιδευτικό Ίδρυμα. Η εφαρμογή είναι απλή και εύκολη στη χρήση της, με αποτέλεσμα από το πρώτο μάθημα να χρειάζεται ελάχιστος χρόνος εξηγήσεων, κυρίως για την εγκατάσταση αυτής. Παράλληλα, παρέχουμε μια ιστοσελίδα στους καθηγητές για την ανάρτηση ερωτηματολογίων που θα απαντούν οι φοιτητές και συλλέγουμε και παρουσιάζουμε τα αποτελέσματα των ερωτηματολογίων άμεσα στους καθηγητές. Αυτή η «ζωντανή» ανατροφοδότηση δίνει μια κατατοπιστική εικόνα στον καθηγητή για το επίπεδο των φοιτητών του κατά τη διάρκεια της διάλεξης, στον ελάχιστο χρόνο.

Η μελλοντική έρευνα και δουλειά σχετικά με την εφαρμογή μας, ξεκινά από την διεύρυνση των δυνατών ερωτήσεων που μπορεί να θέσει ο καθηγητής μέσα από το σύστημα μας, δηλαδή ερωτήσεις πέραν των πολλαπλής επιλογής με 4 πιθανές απαντήσεις εκ των οποίων μόνο μία είναι σωστή. Πιθανά, να μπορούν να υπάρχουν μερικά σωστές απαντήσεις ή πάνω από μία σωστές απαντήσεις ή και καμία σωστή απάντηση. Η δυνατότητα των φοιτητών να προσθέτουν τη δική τους απάντηση ή η αυτόματη δημιουργία ερωτημάτων ή επιπλέον απαντήσεων. Τα παραπάνω θα επιτρέψουν στους διδάσκοντες να είναι ακόμα πιο δημιουργικοί και διαδραστικοί κατά τη διάρκεια του μαθήματος και να κεντρίζουν περαιτέρω το ενδιαφέρον των φοιτητών, ενεργοποιώντας παράλληλα την κριτική τους σκέψη. Θα μπορούσε να γίνει και μια βελτίωση στο User Interface, κυρίως στο εικαστικό κομμάτι και λιγότερο στη ροή της εφαρμογής. Τέλος, μια εκτενέστερη και σε βάθος χρόνου εφαρμογή του συστήματος σε πραγματικές συνθήκες, π.χ. κατά τη διάρκεια ενός ή δύο εξαμήνων, θα παρείχε χρήσιμα επιπρόσθετα δεδομένα τόσο για την αποδοχή ενός τέτοιου συστήματος στην εκπαιδευτική διαδικασία, όσο και για τα επόμενα βήματα έρευνας του συγκεκριμένου πεδίου και των αναγκών τόσο των φοιτητών όσο και των καθηγητών.

8 Βιβλιογραφία

- [1] L. Abeyssekera & P. Dawson (2015) "Motivation and cognitive load in the flipped classroom: definition, rationale and a call for research", *Higher Education Research & Development*, 34:1, 1-14, DOI: 10.1080/07294360.2014.9343361.
- [2] V. Triglianios, S. Praharaaj, C. Pautasso, A. Bozzon & C. Hauff (2017) "Measuring Student Behaviour Dynamics in a Large Interactive Classroom Setting", *UMAP '17 Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization* Pages 212-220
- [3] S. Wang & Y. Chen (2018) "Rain classroom: a tool for blended learning with MOOCs", *Proceedings of the Fifth Annual ACM Conference on Learning at Scale* Article No. 27
- [4] Kay, R.H., LeSage, A.: (2009) "Examining the benefits and challenges of using audience response systems: A review of literature". *Computers & Education* 53, 819-827, 2009.
- [5] Antonelou, G.E., Verykios, V.S., Kalantzi, R.G., Panagiotakopoulos, C.T. and Stavropoulos, E.C., "Supporting Face-to-Face Live Sessions in Hellenic Open University based on Blended Learning Approaches and ICT tools", 8th International Conference in Open and Distance Learning 2015 (ICODL2015), Section A: theoretical papers, original research and scientific articles, pp. 72 – 81, October 2015, Athens.
- [6] Mantoro, T.; Ayu, M.A.; Habul, E.; Khasanah, A.U. "Survnote: A free web based audience response system to support interactivity in the classroom", in 2010 IEEE Conference on Open Systems (ICOS2010), pp.34-39, December 5-7, 2010.
- [7] Ayu, M.A., Taylor, K., Mantoro, T., "Active learning: Engaging Students in the Classroom Using Mobile Phones", *Proceedings of ISIEA 2009*, Kuala Lumpur, Malaysia, 2009.
- [8] C. Cheong, V. Bruno & F. Cheong (2012) "Designing a Mobile-app-based Collaborative Learning System", *Journal of Information Technology Education: Volume 11*, 2012
- [9] B. Cestnik, M. Bohanec & T. Urbančič (2015) "QTVity: Advancing Students' Engagement during Lectures by Using Mobile Devices", *CompSysTech '15 Proceedings of the 16th International Conference on Computer Systems and Technologies* Pages 334-341
- [10] C. Karachristos, E. Stavropoulos, V. Verykios, F. Lazarinis, E. Sakkopoulos & G. Androulakis (2016) "Promoting Active Learning using a web-based Audience Response System", *Proceedings of the 20th Pan-Hellenic Conference on Informatics* Article No. 38
- [11] H. Zhai, Q. Liu & X. Feng (2018) "A Comparative Study of Classroom Response Systems in China and Abroad", *ICDEL '18 Proceedings of the 2018 International Conference on Distance Education and Learning* Pages 69-73
- [12] Caldwell, J. E. (2007). "Clickers in the large classroom: Current research and best-practice tips". *Life Sciences Education*, 6(1), 9.
- [13] Yung-Ting Chuang (2015) "SSCLS: A Smartphone-Supported Collaborative Learning System". *Telematics and Informatics* Volume 32, Issue 3, Pages 463-474
- [14] D. Deb, M. Fuad, J. Etim & C. Gloster (2018) "MRS: Automated Assessment of Interactive Classroom Exercises", *SIGCSE'18*
- [15] M. Skirpan & T. Yeh (2015) "Beyond the Flipped Classroom: Learning by Doing Through Challenges and Hack-a-thons", *SIGCSE'15*
- [16] A. Garbett, D. Chatting, G. Wilkinson, C. Lee & A. Kharrufa (2018) "ThinkActive: Designing for Pseudonymous Activity Tracking in the Classroom", *CHI 2018*
- [17] M. Llamas-Nistal, M. Caeiro-Rodriguez & J. Gonzalez-Tato (2010) "Web-based Audience Response System Using the Educational Platform Called BeA"
- [18] J. Cain, E. P. Black & J. Rohr (2009) "An Audience Response System Strategy to Improve Student Motivation, Attention, and Feedback", *American Journal of Pharmaceutical Education* 2009; 73 (2) Article 21
- [19] N. Efstathiou & C. Bailey (2012) "Promoting active learning using Audience Response System in large bioscience classes", *Nurse Education Today* 32 (2012) 91–95
- [20] K. A. Clauson, F. M. Alkhateeb & D. Singh-Franco (2012) "Concurrent Use of an Audience Response System at a Multi-Campus College of Pharmacy", *American Journal of Pharmaceutical Education* 2012; 76 (1) Article 6.
- [21] S. W. Draper & M. I. Brown (2004) "Increasing interactivity in lectures using an electronic voting system", *Journal of Computer Assisted Learning* 20, pp81–94
- [22] Ray A. Burnstein, & Leon M. Lederman "Using wireless keypads in lecture classes", *The Physics Teacher* Vol. 39, January 2001
- [23] C. R. Graham, T.R. Tripp, L. Seawright & G. L. Joeckel (2007) "Empowering or compelling reluctant participants using audience response systems" 2007 SAGE Publications
- [24] W. Beekes (2006) "The 'Millionaire' method for encouraging participation" 2006 SAGE Publications

- [25] J. Stav, K. Nielsen, G. Nygard & T. Thorseth (2009)“Experiences Obtained with Integration of Student Response Systems for iPod Touch and iPhone into e-Learning Environments” Electronic Journal of e-Learning Volume 8 Issue 2 2010, (pp179 - 190),
- [26] D. Bunce, J. VandenPlas & K. Havanki (2006)“Comparing the Effectiveness on Student Achievement of a Student Response System versus Online WebCT Quizzes”, Journal of Chemical Education • Vol. 83 No. 3
- [27] Α. Ν. Χαζτηγεωργίου «Αντικειμενοστρεφής Σχεδίαση», Εκδόσεις Κλειδάριθμος
- [28] <https://www.uml-diagrams.org/>
- [29] <https://www.justinmind.com/>
- [30] <https://facebook.github.io/react-native/>
- [31] <https://laravel.com/>
- [32] <https://expo.io/>
- [33] <https://charts.erik.cat/>
- [34] <https://www.pewinternet.org/fact-sheet/mobile/>
- [35] <https://www.codecademy.com/articles/mvc>
- [36] <https://www.iis.net/>
- [37] <https://www.mysql.com/>
- [38] <https://www.genymotion.com/>
- [39] <https://babeljs.io/repl>
- [40] <https://github.com/ehzhang/nunchuck>