

ΤΕΧΝΙΚΟ ΕΓΧΕΙΡΙΔΙΟ

Εκπαιδευτικό Λογισμικό για την προπαίδεια των Μαθηματικών μέχρι το 10

ΑΚΑΔΗΜΑΙΚΟ ΕΤΟΣ 2019 – 2020

Π16036 – ΙΩΑΝΝΙΔΗΣ ΠΑΝΑΓΙΩΤΗΣ

TEAMS: P16036@UNIP1.GR

EMAIL: PANOS277@HOTMAIL.COM

Π16112 – ΠΑΡΑΒΑΝΤΗΣ ΑΘΑΝΑΣΙΟΣ

TEAMS: P16112@UNIP1.GR

EMAIL: THANOSPARAVANTIS@GMAIL.COM

1 Πίνακας Περιεχομένων

1	Πίνακας Περιεχομένων	1
1	Εισαγωγή.....	2
1.1	Στόχοι της εργασίας.....	2
1.2	Ορισμός του προβλήματος προς επίλυση.....	3
2	Σύντομη παρουσίαση της RUP	4
2.1	Φάση Σύλληψης.....	4
2.2	Φάση Επεξεργασίας.....	4
2.3	Φάση Κατασκευής	5
2.4	Φάση Μετάβασης.....	5
3	Οδηγίες εγκατάστασης	6
4	Φάση: Έναρξης (Inception)	8
4.1	Σύλληψη απαιτήσεων.....	8
4.2	Ανάλυση-Σχεδιασμός.....	9
4.2.1	Διάγραμμα Περιπτώσεων Χρήσης (1 ^η έκδοση)	9
5	Φάση: Εκπόνησης Μελέτης (Elaboration)	10
5.1	Ανάλυση-Σχεδιασμός.....	10
5.1.1	Διαγράμματα Περιπτώσεων Χρήσης (2 ^η έκδοση)	11
5.2	Υλοποίηση-Έλεγχος.....	12
5.2.1	Υλοποίηση: 1 ^η εκτελέσιμη έκδοση	12
5.2.2	Αναφορά ελέγχου για την 1 ^η εκτελέσιμη έκδοση	12
6	Φάση: Κατασκευή (Construction).....	13
6.1	Ανάλυση-Σχεδιασμός.....	13
6.1.1	Διαγράμματα Περιπτώσεων Χρήσης (3 ^η έκδοση)	14
6.2	Υλοποίηση-Έλεγχος.....	15
6.2.1	Υλοποίηση: Τελική εκτελέσιμη έκδοση	15
6.2.2	Αναφορά ελέγχου για την τελική εκτελέσιμη έκδοση	15

1 Εισαγωγή

1.1 Στόχοι της εργασίας

Ζητείται να γίνει ένα αλληλεπιδραστικό λογισμικό εκπαίδευσης μαθητών Πρωτοβάθμιας εκπαίδευσης για την προπαίδεια των Μαθηματικών Δημοτικού. Η εργασία αυτή θα περιλαμβάνει τρόπους παρουσίασης της προπαίδειας με στόχο να γίνει το θέμα κατανοητό και να μπορεί να απομνημονευθεί από τους μαθητές και να εμπεδωθεί η ύλη μέσω ασκήσεων. Ο κύριος σκοπός της εργασίας είναι ο καλός σχεδιασμός και υλοποίηση του εκπαιδευτικού λογισμικού (διδασκαλία – αξιολόγηση του μαθητή).

Συγκεκριμένα ζητούνται τα παρακάτω:

ΒΑΣΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ

1. ΠΑΡΟΥΣΙΑΣΗ ΔΙΔΑΣΚΑΛΙΑΣ

2. ΕΡΩΤΗΣΕΙΣ / ΤΕΣΤ ΑΥΤΟΑΞΙΟΛΟΓΗΣΗΣ

2.1 Κατασκευή των τεστ.

Θα πρέπει να κατασκευάζονται τεστ για την αυτοαξιολόγηση των μαθητών, όπου θα παρουσιάζονται με τυχαία σειρά κάποιες πράξεις πολλαπλασιασμού για κάθε αριθμό ξεχωριστά μέχρι το 10.

Στα επαναληπτικά τεστ, θα πρέπει να παρουσιάζονται πράξεις πολλαπλασιασμού από όλους τους αριθμούς μέχρι το 10.

2.2 Αποθήκευση στατιστικών στοιχείων προόδου του μαθητή.

Θα πρέπει να υπάρχει μια βάση δεδομένων όπου να αποθηκεύονται στοιχεία για κάθε μαθητή σχετικά με την πρόδοό του. Τα στοιχεία θα βασίζονται στην απόδοση των μαθητών στα τεστ αυτοαξιολόγησης.

2.3 Διάγνωση λαθών του μαθητή και αλληλεπίδραση.

Στη διάγνωση λαθών ζητείται να μπορεί το σύστημα να εντοπίσει αν ο μαθητής έχει πρόβλημα σε συγκεκριμένη προπαίδεια κάποιου αριθμού (π.χ. του 7). Αν εντοπιστεί κάτι τέτοιο θα πρέπει το σύστημα να παρουσιάζει πάλι τη θεωρία και περισσότερες ερωτήσεις στο συγκεκριμένο θέμα. Επίσης θα πρέπει να καταγράφεται η συγκεκριμένη αδυναμία στα στατιστικά προόδου του μαθητή και να σβήνεται όταν ο μαθητής φαίνεται ότι έχει πια μάθει το συγκεκριμένο θέμα

ΠΡΟΣΘΕΤΕΣ ΛΕΙΤΟΥΡΓΙΕΣ (ΠΡΟΑΙΡΕΤΙΚΕΣ)

1. Διαχείριση από Καθηγητή
2. Λειτουργία στο Web

Συνοδευτικά εγχειρίδια

Η εφαρμογή θα πρέπει να συνοδεύεται από τα εξής εγχειρίδια:

1. Εγχειρίδιο χρήση (user manual)
2. On-line help (που να παρέχεται τρέχοντας την εφαρμογή)
3. Εγχειρίδιο Ανάλυσης και Σχεδιασμού της εφαρμογής (Τεχνικό Εγχειρίδιο).

- Η εργασία θα πρέπει να γίνει από ομάδες των 2 ατόμων.
- Η υλοποίηση της εργασίας να γίνει σε γλώσσα οπτικού-παραθυρικού προγραμματισμού (π.χ. Visual C#, Visual Basic, Visual J++ κ.τ.λ.).
- Η παράδοση της εργασίας θα γίνει στα εργαστήρια του Τμήματος σε ημερομηνία που θα οριστεί από τους υπεύθυνους καθηγητές εντός των εξεταστικών περιόδων.
- Η ισχύς της παρούσας εργασίας είναι για το ακαδημαϊκό έτος 2019-2020.

1.2 Ορισμός του προβλήματος προς επίλυση

Η εκμάθηση της προπαίδειας είναι ένα πολύ σημαντικό στοιχείο στην εκπαίδευση των παιδιών. Παρακολουθώντας τις τελευταίες τεχνολογικές εξελίξεις και τις τάσεις που κυριαρχούν, οφείλουμε να παρέχουμε σύγχρονα και χρήσιμα εργαλεία για την αποτελεσματική εκμάθηση της προπαίδειας. Στις μέρες μας, ο καθένας έχει από έναν υπολογιστή, κινητό ή tablet, και προπαντός είμαστε συνέχεια συνδεδεμένοι στο διαδίκτυο. Για να καλύψουμε τις ανάγκες της εργασίας, κρίθηκε απαραίτητη η ανάπτυξη μιας διαδικτυακής εκπαιδευτικής εφαρμογής, που θα παρέχει πλούσιο αλληλεπιδραστικό υλικό τόσο για την εκμάθηση όσο και για την εξάσκηση πάνω στη προπαίδια.

Η ανάπτυξη μιας τέτοιας εφαρμογής με αρκετές τεχνολογικές απαιτήσεις, θέτει πολλαπλά ζητήματα στο «τραπέζι» τα οποία καλούμαστε να λύσουμε. Μερικά από αυτά συνοψίζονται στα παρακάτω:

- Ανάπτυξη διαδικτυακής υπηρεσίας (back-end) για τη παροχή της αναγκαίας λειτουργικότητας και την υλοποίηση των απαιτήσεων της εργασίας.
- Ανάπτυξη διαδικτυακής διεπαφής (front-end) για τη απρόσκοπτη πρόσβαση στην εφαρμογή με σύγχρονα τεχνολογικά μέσα.
- Σχεδίαση των λειτουργιών με τρόπο προσιτό προς τους μαθητές δημοτικού έχοντας υπ' όψη τις αρχές σχεδίασης εκπαιδευτικού λογισμικού.
- Ασφάλεια των πληροφοριών για τη προστασία των μαθητών αλλά και των διαχειριστών που έχουν πρόσβαση στην εφαρμογή με οποιονδήποτε τρόπο.
- Παραγωγή μαθησιακού υλικού για την εκμάθηση της προπαίδειας με διασκεδαστικό αλλά ταυτόχρονα αποτελεσματικό τρόπο.
- Επιβράβευση των μαθητών για τις καλές επιδόσεις τους ως κίνητρο να συνεχίσουν την εκπαιδευτική διαδικασία μέσω της εφαρμογής.
- Καταγραφή στατιστικών στοιχείων που προκύπτουν άμεσα από το χρήση της εφαρμογής για τη συνεχή βελτίωση του συστήματος και παρακολούθηση των επιδόσεων των μαθητών.

Όλα τα παραπάνω ζητήματα αναλύονται εκτενώς στις επιμέρους φάσεις σχεδίασης της εφαρμογής.

2 Σύντομη παρουσίαση της RUP

Ο κύκλος ζωής του λογισμικού (software life-cycle) σκιαγραφεί τη ζωή του προγράμματος λογισμικού από τη στιγμή της γέννησής του μέχρι τη στιγμή της αντικατάστασης ή της εγκατάλειψής του. Ο κύκλος ζωής του λογισμικού στην RUP υποδιαιρείται σε τέσσερις συνεχόμενες φάσεις, οι οποίες είναι:

- Η Φάση Σύλληψης (Inception Phase)
- Η Φάση Επεξεργασίας (Elaboration Phase)
- Η Φάση Κατασκευής (Construction Phase)
- Η Φάση Μετάβασης (Transition Phase)

Ο κύκλος ανάπτυξης είναι ένα πέρασμα από τέσσερις φάσεις. Το κάθε πέρασμα από αυτές τις φάσεις ονομάζεται γενιά (generation).

Σε περίπτωση που το προϊόν συνεχίσει να εξελίσσεται στην επόμενη γενιά του, έχουμε έναν νέο διαδοχικό κύκλο. Αυτοί οι κύκλοι ονομάζονται κύκλοι εξέλιξης (evolution cycles).

Καθεμία από τις φάσεις ολοκληρώνεται με την παράδοση κάποιων ορόσημων. Στο τέλος κάθε φάσης πραγματοποιείται μια αποτίμηση η οποία κρίνει εάν οι στόχοι της φάσης έχουν ικανοποιηθεί προκειμένου να αρχίσει η επόμενη φάση.

2.1 Φάση Σύλληψης

Τα αποτελέσματα της φάσης Σύλληψης είναι τα εξής:

- Μία γενική περιγραφή των βασικών απαιτήσεων του έργου, τα σημεία κλειδιά και οι βασικοί περιορισμοί.
- Ένα αρχικό μοντέλο περιπτώσεων χρήσης (ολοκληρωμένο κατά 10%-20%)
- Ένα αρχικό γλωσσάριο του έργου.
- Μια αρχική εκτίμηση των κινδύνων.
- Μία αρχική επιχειρηματική περίπτωση, η οποία περιέχει το επαγγελματικό πλαίσιο, κριτήρια επιτυχίας και οικονομική πρόβλεψη.
- Ένα σχέδιο του έργου (project plan), το οποίο δείχνει φάσεις και επαναλήψεις.
- Ένα επιχειρηματικό μοντέλο (business model) αν είναι απαραίτητο.
- Ένα ή διάφορα πρωτότυπα.

2.2 Φάση Επεξεργασίας

Το αποτέλεσμα της φάσης επεξεργασίας είναι:

- Ένα μοντέλο περιπτώσεων χρήσης (ολοκληρωμένο τουλάχιστον κατά 80%)
- Συμπληρωματικές απαιτήσεις για τις μη λειτουργικές απαιτήσεις και απαιτήσεις που δεν έχουν σχέση με μια συγκεκριμένη περίπτωση χρήσης.
- Ένα εκτελέσιμο αρχιτεκτονικό πρωτότυπο
- Μια περιγραφή της αρχιτεκτονικής του λογισμικού
- Επανεξέταση της λίστας κινδύνων και της επιχειρηματικής περίπτωσης
- Ένα σχέδιο ανάπτυξης για το ολικό έργο
- Το έργο μπορεί να ματαιωθεί ή να επανεξεταστεί σοβαρά η πραγματοποίησή του αν αποτύχει να περάσει από αυτό το ορόσημο

2.3 Φάση Κατασκευής

Αποτελείται τουλάχιστον από τα εξής:

- Το προϊόν λογισμικού
- Το εγχειρίδιο του χρήστη
- Μια περιγραφή της παρούσας άδειας έκδοσης

Η μετάβαση μπορεί να αναβληθεί αν το έργο αποτύχει να περάσει αυτό το ορόσημο.

2.4 Φάση Μετάβασης

Ο σκοπός της φάσης είναι η μετάβαση του λογισμικού προϊόντος στην κοινότητα των χρηστών. Αφού το προϊόν παραδοθεί στους τελικούς χρήστες, συνήθως εντοπίζονται προβλήματα που απαιτούν την ανάπτυξη νέων εκδόσεων.

Σε αυτό το σημείο αποφασίζεται αν οι στόχοι έχουν επιτευχθεί και αν θα πρέπει να ξεκινήσει ο επόμενος κύκλος εξέλιξης.

3 Οδηγίες εγκατάστασης

Πριν αναλυθούν οι τρεις φάσεις ανάπτυξης του λογισμικού, είναι αναγκαίο να παρουσιάσουμε τη δομή των φακέλων του project:

- Ο φάκελος “backend” περιέχει όλα τα αρχεία που υλοποιούν το business logic της εφαρμογής. Στηρίζεται στο Node.js επομένως μιλάμε για έναν εξυπηρετητή εξ’ ολοκλήρου σε JavaScript.
- Ο φάκελος “frontend” περιέχει όλα τα αρχεία που υλοποιούν τη διεπαφή χρήστη μέσω του browser. Συγκεκριμένα, το project έχει δημιουργηθεί στον υποφάκελο “mlt-elearning”. Στηρίζεται στο Angular framework επομένως όπως και στο backend είναι γραμμένο σε JavaScript.
- Ο φάκελος “database/mltelearning” περιέχει τα αρχεία προς εισαγωγή στη βάση δεδομένων MongoDB που είναι απαραίτητα για την εφαρμογή.
- Ο κυρίως φάκελος περιέχει όλα τα PDF τεκμηρίωσης που ζητούνται, δηλαδή το εγχειρίδιο χρήστη και το παρών τεχνικό εγχειρίδιο. Μαζί με αυτά στο φάκελο “docs/diagrams” περιέχονται τα UML διαγράμματα που είναι ενσωματωμένα παρακάτω ως εικόνες.

Για την εγκατάσταση και εκτέλεση της εφαρμογής προτείνουμε να ακολουθήσετε τα παρακάτω βήματα:

- Εγκατάσταση του Node.js από αυτή τη σελίδα:
<https://nodejs.org/en/download/>
- Εγκατάσταση της MongoDB όπως περιγράφεται λεπτομερώς σε αυτή τη σελίδα:
<https://docs.mongodb.com/manual/installation/>
- Εγκατάσταση του Angular CLI μέσω του NPM όπως περιγράφεται εδώ:
<https://angular.io/guide/setup-local>
- Εκκίνηση της MongoDB με βάση τις οδηγίες που παρατέθηκαν παραπάνω, ανάλογα με το προγραμματιστικό περιβάλλον στο οποίο βρίσκεστε.
- Εισαγωγή των δεδομένων στη βάση μας με την εντολή:
mongorestore --drop -d mltlearning ./database/mltelearning
- Εκτέλεση του αρχείου “backend/app.js” μέσω του Node.
Εφόσον είμαστε στον φάκελο “backend” τότε μέσω της γραμμής εντολών εκτελούμε τις εντολές:
 1. **npm install** (Για εγκατάσταση των απαιτούμενων βιβλιοθηκών)
 2. **node ./src/app.js** (Για εκκίνηση της backend εφαρμογής)
- Εκκίνηση του Angular project μέσω του Angular CLI για το frontend.
Εφόσον είμαστε στον φάκελο “frontend/mlt-elearning” τότε μέσω της γραμμής εντολών εκτελούμε την εντολή:
 1. **npm install** (Για εγκατάσταση των απαιτούμενων βιβλιοθηκών)
 2. **ng serve** (Για εκκίνηση της frontend εφαρμογής)

Αφού εκκινήσουμε το Angular project επισκεπτόμαστε τη σελίδα <http://localhost:4200> και συνδεόμαστε με τα παρακάτω στοιχεία χρήστη:

- **Μαθητές**
 1. *Username:* rioannidis
 2. *Username:* paravantis

- **Δάσκαλος / Διαχειριστής**
 1. *Username:* elrapa

Σε όλους τους λογαριασμούς το ***password*** είναι το ίδιο: **123456**

4 Φάση: Έναρξης (Inception)

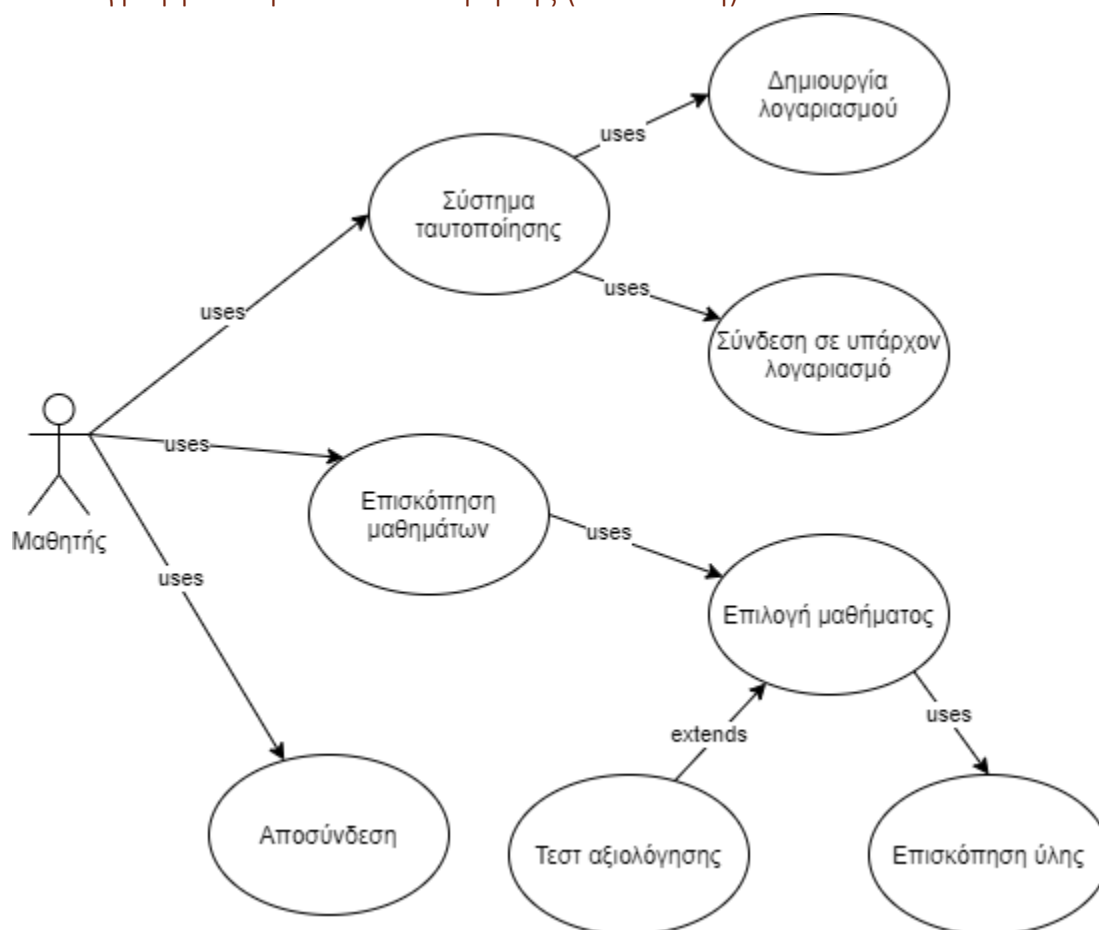
4.1 Σύλληψη απαιτήσεων

Στη πρώτη φάση σχεδίασης της εφαρμογής «Multiplication eLearning» αναλύσαμε τις πρώτες απαιτήσεις που προκύπτουν από μια τέτοια εκπαιδευτική εφαρμογή. Δίνεται έμφαση στη λειτουργικότητα και όχι τόσο στη σχεδίαση της διεπαφής ώστε να έχουμε άμεσα τα πρώτα χειροπιαστά αποτελέσματα.

- Ανάπτυξη της πρωτότυπης εφαρμογής (prototyping) για έναν μόνο αριθμό της προπαίδειας και ένα τεστ αξιολόγησης.
- Προσθήκη αρχικής ύλης για αυτό τον αριθμό και συζήτηση πάνω στη μορφή παρουσιάσής της. Θεωρούμε πως είναι ένα πολύ σημαντικό βήμα αυτό, εφόσον τα μαθήματα παρουσιάζονται πριν την αξιολόγηση.
- Προσθήκη του πρώτου τεστ αξιολόγησης πάνω στην ύλη που παρουσιάστηκε και συζήτηση για τη δομή του. Τόσο οι ερωτήσεις όσο και οι απαντήσεις πρέπει να είναι ευκολοδιάβαστες και κατανοητές.
- Ανάπτυξη συστήματος ταυτοποίησης των χρηστών με απλό login και registration ώστε να γίνει από την αρχή διαχωρισμός των ρόλων που θα προστεθούν μετέπειτα.

4.2 Ανάλυση-Σχεδιασμός

4.2.1 Διάγραμμα Περιπτώσεων Χρήσης (1^η έκδοση)



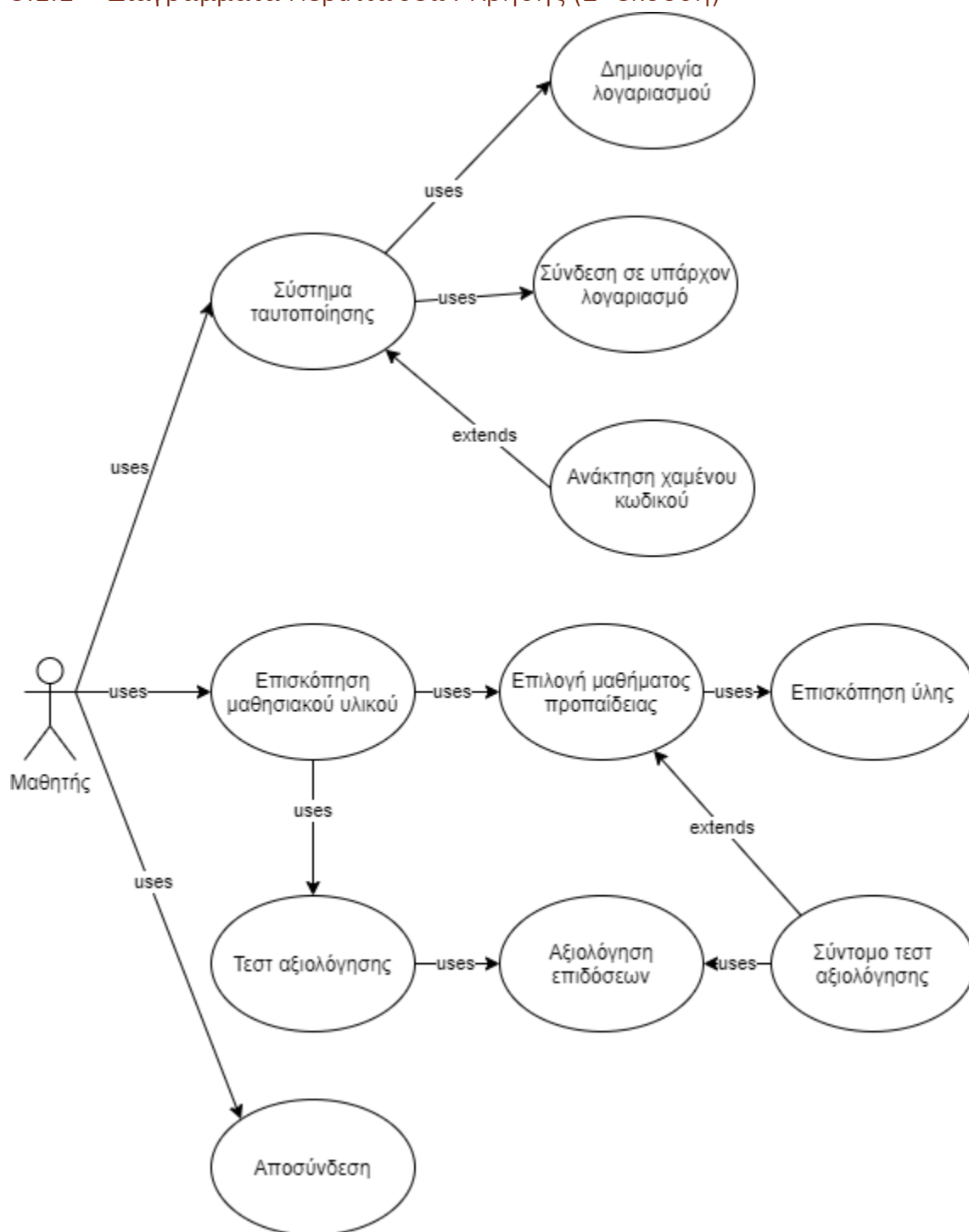
5 Φάση: Εκπόνησης Μελέτης (Elaboration)

5.1 Ανάλυση-Σχεδιασμός

Στη δεύτερη φάση σχεδίασης της εφαρμογής στόχος μας είναι η ανάπτυξη της πρώτης έκδοσης της εφαρμογής. Έμφαση δίνεται τόσο στη διεπαφή όσο και στη λειτουργικότητα, εφόσον αρκετά χαρακτηριστικά θα καταλήξουν στο τελικό προϊόν. Κρατώντας τη προεργασία της πρώτης φάσης, περιυλλέγουμε όλες τις ιδέες που προέκυψαν από συζητήσεις και εμπλουτίζουμε ακόμη περισσότερο την εφαρμογή μας.

- Το σύστημα ταυτοποίησης παρέχει τη δυνατότητα σύνδεσης σε υπάρχον λογαριασμό, δημιουργία νέου λογαριασμού και ανάκτηση χαμένου κωδικού. Προφανώς, και οι τρεις αυτές διαδικασίες αφορούν την ασφάλεια της εφαρμογής επομένως πρέπει να δοθεί μεγάλη βαρύτητα.
- Διακρίνουμε το πρώτο «ρόλο» στους χρήστες της εφαρμογής, και αυτός δεν είναι άλλος από τον Μαθητή. Ο μαθητής που θα χρησιμοποιεί την εφαρμογή, έχει πια τη δυνατότητα επισκόπησης όλου του εκπαιδευτικού υλικού. Εκεί μπορεί να επιλέξει τα μαθήματα που του είναι διαθέσιμα καθώς και τα τεστ αξιολόγησης για να ξεκλειδώσει τα επόμενα.
- Τα μαθήματα προπαίδειας και τα τεστ αξιολόγησης αποκτούν τη πρώτη μορφή τους με συνεχή αξιολόγηση από δασκάλους και μαθητές. Γίνονται συνεχείς αλλαγές έτσι ώστε να καταλήξουμε σε μια μορφή προσιτή από το κοινό στο οποίο απευθύνεται η εφαρμογή.
- Τα τεστ αξιολόγησης καθώς και τα σύντομα τεστ ανά μάθημα καθιερώνουν την επίδοση κάθε μαθητή, η οποία μπορεί να αξιολογηθεί από τον εκάστοτε καθηγητή που θα επιβλέπει την πλατφόρμα.
- Η διεπαφή της εφαρμογής εμπλουτίζεται με γραφικά προσιτά προς τους μαθητές καθώς και βίντεο που συμβάλλουν στην εκπαιδευτική διαδικασία. Είναι σημαντικό οι μαθητές να επιβραβεύονται κάθε φορά που κάνουν κάτι σωστό.

Έχοντας υπ' όψη τα παραπάνω, προχωράμε στη δημιουργία της πρώτης έκδοσης του «Multiplication eLearning», έτσι ώστε να διανεμηθεί σε μερικούς μαθητές και δασκάλους. Εφόσον γίνει αυτό και αξιολογηθούν οι τρόποι χρήσης της εφαρμογής από το κοινό, θα γίνουν οι απαραίτητες αλλαγές που θα οδηγήσουν στη τελική έκδοση.

5.1.1 Διαγράμματα Περιπτώσεων Χρήσης (2^η έκδοση)

5.2 Υλοποίηση-Έλεγχος

5.2.1 Υλοποίηση: 1^η εκτελέσιμη έκδοση

Για την υλοποίηση της εφαρμογής χρησιμοποιούμε το NodeJS ως εργαλείο της υπηρεσίας back-end και το AngularJS ως βιβλιοθήκη ανάπτυξης της διεπαφής των χρηστών. Για τις ανάγκες αποθήκευσης των δεδομένων, χρησιμοποιείται η MongoDB για σύστημα διαχείρισης της βάσης δεδομένων. Η αλληλεπίδραση αυτών των συστημάτων είναι απαραίτητη έτσι ώστε να δημιουργηθεί η πρώτη έκδοση της εφαρμογής.

Για την υλοποίηση του business logic χρησιμοποιούμε JavaScript ακολουθώντας τα πρότυπα ανάπτυξης λογισμικού. Επιπλέον, αρκετές λειτουργίες είναι ήδη υλοποιημένες σε διάφορες βιβλιοθήκες που διανέμονται δωρεάν. Επιλέξαμε λοιπόν στο back-end να χρησιμοποιήσουμε τις εξής: bcrypt, cors, express, helmet, jsonwebtoken, mongoose, nodemailer, passport, passport-jwt, passport-local και uuid.

Για την υλοποίηση της διεπαφής χρήστη, χρησιμοποιούμε JavaScript ξανά, με τη βοήθεια της διάσημης πια βιβλιοθήκης AngularJS. Όπως και στο backend έτσι και στο frontend χρησιμοποιούμε τις εξής βιβλιοθήκες: @angular-devkit/build-angular, @angular/cli, @angular/compiler-cli, @types/node, @types/jasmine, @types/jasminewd2, codelyzer, jasmine-core, jasmine-spec-reporter, karma, karma-chrome-launcher, karma-coverage-istanbul-reporter, karma-jasmine, karma-jasmine-html-reporter, protractor, ts-node, tslint, typescript, @angular/animations, @angular/cdk, @angular/common, @angular/compiler, @angular/core, @angular/forms, @angular/platform-browser, @angular/platform-browser-dynamic, @angular/router, @fullcalendar/core, @fullcalendar/daygrid, chart.js, eva-icons, fullcalendar, i, primeflex, primeicons, primeng, quill, rxjs, tslib και zone.json.

5.2.2 Αναφορά ελέγχου για την 1^η εκτελέσιμη έκδοση

Σε γενικές γραμμές η πρώτη εκτελέσιμη έκδοση ήταν αρκετά ικανοποιητική. Εφαρμόσαμε μερικές αρχικές ιδέες από το prototype της πρώτης φάσης, καθώς και νέες ιδέες που προέκυψαν στη πορεία. Οι χρήστες που δοκίμασαν την εφαρμογή πιλοτικά επισήμαναν μερικά πράγματα που ήταν δυσνόητα. Προφανώς, δεν έλειψαν και τα σφάλματα στο κώδικα, μερικά από τα οποία διορθώθηκαν και άλλα πιο ριζικά θα καταλήξουν διορθωμένα στη δεύτερη εκτελέσιμη έκδοση.

6 Φάση: Κατασκευή (Construction)

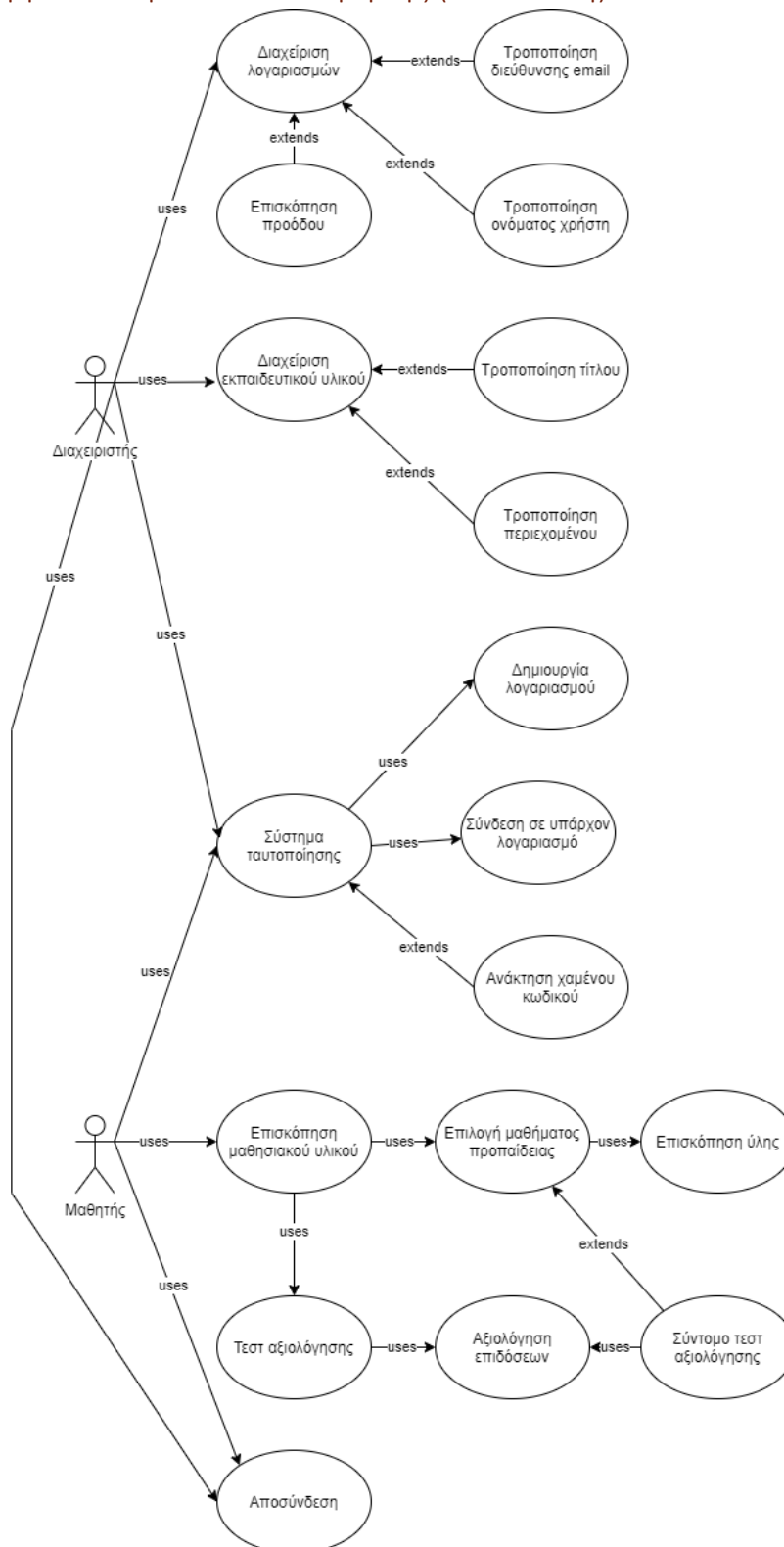
6.1 Ανάλυση-Σχεδιασμός

Στη τρίτη και τελευταία φάση ανάπτυξης της εφαρμογής προχωράμε στη δημιουργία της δεύτερης έκδοσης, βασιζόμενοι στα ελαττώματα και ζητήματα της πρώτης. Σύμφωνα με τα απαιτούμενα της εργασίας, προχωρήσαμε στη δημιουργία του δεύτερου ρόλου του διαχειριστή. Με αυτό τον τρόπο οι δάσκαλοι μπορούν να τροποποιούν τα μαθήματα αλλά και να επιβλέπουν τους λογαριασμούς των μαθητών. Συνεπώς, η ανάλυση και σχεδίαση της τελικής έκδοσης στηρίζεται στους παρακάτω άξονες:

- Διαμόρφωση της διεπαφής χρήστη με βάση τις ανάγκες των χρηστών της πρώτης έκδοσης της εφαρμογής.
- Τροποποίηση του εκπαιδευτικού λογισμικού και των τεστ αξιολόγησης με τέτοιο τρόπο ώστε να γίνει πιο προσιτό προς τους μαθητές.
- Προσθήκη επιπλέον οπτικοακουστικού υλικού για να μεγιστοποιηθεί η αλληλεπίδραση των μαθητών με την εφαρμογή, και ως εκ τούτου να είναι πιο ευχάριστη.
- Προσθήκη των απαραίτητων λειτουργιών για τους διαχειριστές της εφαρμογής. Αυτό περιλαμβάνει επισκόπηση των μαθημάτων, μαθητών και επιδόσεων.
- Διόρθωση καίριων τεχνικών λαθών και ατελειών που προέκυψαν κατά την ανάπτυξη πρώτη έκδοση.

Με βάση τα παραπάνω, προχωράμε στην ανάπτυξη της ολοκληρωμένης εφαρμογής «Multiplication eLearning» που μπορεί να χρησιμοποιηθεί για εκπαιδευτικούς σκοπούς.

6.1.1 Διαγράμματα Περιπτώσεων Χρήσης (3^η έκδοση)



6.2 Υλοποίηση-Έλεγχος

6.2.1 Υλοποίηση: Τελική εκτελέσιμη έκδοση

Όπως στη πρώτη έκδοση έτσι και στη τελική, χρησιμοποιούμε το NodeJS για το backend και το AngularJS για το frontend. Η βάση δεδομένων στη MongoDB παραμένει ως έχει, όμως εμπλουτίζεται με επιπλέον δεδομένα για την διαμόρφωση του τελικού προγράμματος προς διανομή.

Οι βιβλιοθήκες τόσο για το backend όσο και για το frontend παραμένουν ίδιες: bcrypt, cors, express, helmet, jsonwebtoken, mongoose, nodemailer, passport, passport-jwt, passport-local και uuid, @angular-devkit/build-angular, @angular/cli, @angular/compiler-cli, @types/node, @types/jasmine, @types/jasminewd2, codelyzer, jasmine-core, jasmine-spec-reporter, karma, karma-chrome-launcher, karma-coverage-istanbul-reporter, karma-jasmine, karma-jasmine-html-reporter, protractor, ts-node, tslint, typescript, @angular/animations, @angular/cdk, @angular/common, @angular/compiler, @angular/core, @angular/forms, @angular/platform-browser, @angular/platform-browser-dynamic, @angular/router, @fullcalendar/core, @fullcalendar/daygrid, chart.js, eva-icons, fullcalendar, i, primeflex, primeicons, primeng, quill, rxjs, tslib και zone.json.

6.2.2 Αναφορά ελέγχου για την τελική εκτελέσιμη έκδοση

Αξίζει να σημειωθεί πως τα εργαλεία που χρησιμοποιήσαμε μας βοήθησαν στην εξοικονόμηση χρόνου αλλά και στη χρήση καλών πρακτικών κατά την ανάπτυξη λογισμικού. Το NodeJS είναι ένα ολοκληρωμένο σύστημα εκτέλεσης κώδικα JavaScript το οποίο έχει χρησιμοποιηθεί ευρέως στη προγραμματιστική κοινότητα. Το ίδιο ισχύει και για το AngularJS, το οποίο μας επιτρέπει να φτιάξουμε δυναμικές single-page εφαρμογές χωρίς την ανάγκη refresh. Τα εργαλεία αυτά, ακριβώς επειδή έχουν χρησιμοποιηθεί από πολλούς developers, προσφέρουν μια σταθερότητα που είναι καλό να υπάρχει σε αναπτυσσόμενες εφαρμογές.

Για της ανάγκες του Multiplication eLearning, κρίθηκε απαραίτητη η χρήση «δοκιμασμένων» εργαλείων και βιβλιοθηκών, από τη στιγμή που το λογισμικό έχει προοπτικές ευρείας χρήσης στα δημοτικά σχολεία. Ένα ακόμα πλεονέκτημα αυτής της επιλογής είναι η εύκολη συντήρηση του λογισμικού μελλοντικά. Εφόσον χρησιμοποιείται η JavaScript με βάση το Node Package Manager, μπορούν να «κουμπώσουν» νέα πακέτα κώδικα χωρίς να διακινδυνεύεται η σταθερότητα της εφαρμογής.

Τέλος, η διαδικασία συγγραφής τόσο της πρώτης όσο και της δεύτερης έκδοσης έγινε σε περιβάλλον Ubuntu. Χρησιμοποιήθηκε το PyCharm ως IDE για τη συγγραφή κώδικα και το Robot3t για την οπτική αλληλεπίδραση με τη βάση δεδομένων.