



## Μάθημα «Προγραμματισμός II»

Γ' Εξάμηνο

Θέμα:

**Τεχνική Αναφορά**

**StudyBuddyApp**

Βλάσση Πελαγία	8220016
Γιαγιά Ιωάννα	8230022
Δόμπρου Άννα-Ελένη	8230035
Καβαθά Μαρίνα	8230220
Καρατόσιου Αθηνά	8220057
Πένγκ Νταν	8230218
Νικολάου Μαρία	8220211
Τραϊφόρου Αντωνία	8200168

## Table of Contents

Εισαγωγή .....	3
Λειτουργικότητα Λογισμικού .....	4
Περιγραφή χαρακτηριστικών .....	4
Απαιτήσεις Λογισμικού .....	5
Περιορισμοί .....	6
Ροές Εργασίας (Workflows) .....	7
Αρχιτεκτονική και Σχεδίαση .....	11
Αρχιτεκτονική Συστήματος - UML .....	11
Σχέδιο βάσης δεδομένων .....	15
Διάγραμμα ροής δεδομένων .....	17
Προδιαγραφές Λογισμικού .....	17
Τεχνολογίες Υλοποίησης .....	17
Αλγόριθμος .....	18
Απαιτήσεις Συστήματος .....	19
Λειτουργικές απαιτήσεις .....	19
Μη λειτουργικές απαιτήσεις .....	21
Δοκιμές .....	21
Σχέδιο δοκιμών .....	21
Σχέδιο Υλοποίησης .....	22
Χρονοδιάγραμμα .....	22
Παραδοτέα .....	23
Τεκμηρίωση .....	24
Οδηγίες χρήσης για τελικούς χρήστες .....	24
Τεκμηρίωση λογισμικού .....	24
Συμπεράσματα και Μελλοντικές Εκδόσεις .....	24
Παραπομπές .....	25
Βιβλιογραφία .....	25

## Εισαγωγή

Ο σκοπός του εγγράφου είναι η χρήση του ως πλήρους οδηγού για τη σχεδίαση, ανάπτυξη και αξιολόγηση μιας προσωποποιημένης εφαρμογής διαχείρισης μελέτης, απευθυνόμενης σε φοιτητές και μαθητές. Στο έγγραφο περιλαμβάνεται η ανάλυση των αναγκών που σχετίζονται με τις καθημερινές προκλήσεις της εκπαιδευτικής ζωής, όπως η διαχείριση του χρόνου, οι προθεσμίες εργασιών, οι ημερομηνίες εξετάσεων και η ανάγκη οργάνωσης της μελέτης. Οι τεχνικές και λειτουργικές απαιτήσεις της εφαρμογής παρουσιάζονται λεπτομερώς, με αναφορά στη μεθοδολογία ανάπτυξης, στις χρησιμοποιούμενες τεχνολογίες και στη δομή της αρχιτεκτονικής.

Οι βασικές λειτουργίες της εφαρμογής, περιλαμβανόμενες στο έγγραφο, αναφέρονται στη δημιουργία προσωποποιημένων πλάνων μελέτης, την παρακολούθηση προθεσμιών, την αποθήκευση δεδομένων και την ανάλυση της προόδου μέσω στατιστικών στοιχείων. Η εφαρμογή αναπτύσσεται με βάση τη συνδυασμένη χρήση της δομημένης προσέγγισης του Rational Unified Process (RUP) και της ευέλικτης μεθοδολογίας Scrum, εξασφαλίζοντας την οργάνωση του έργου σε φάσεις, με έμφαση στη συνεχή προσαρμογή και επαναληπτική βελτίωση.

Οι πληροφορίες για τη διαχείριση του έργου, όπως η εκτίμηση πόρων, τα χρονοδιαγράμματα, οι εκτιμήσεις κόστους και οι χρονολογικές εξαρτήσεις, καταγράφονται με ακρίβεια. Περιλαμβάνονται διαγράμματα ροής δεδομένων (DFDs) και UML, σχεδιασμένα για να αποτυπώνουν τη λειτουργικότητα της εφαρμογής. Η διαδικασία δοκιμών περιγράφεται αναλυτικά, εξασφαλίζοντας την ποιότητα και τη λειτουργικότητα του προϊόντος, ενώ οι βασικοί δείκτες απόδοσης (KPIs) χρησιμοποιούνται για την αξιολόγηση της επιτυχίας του.

Τέλος, τονίζεται η δυνατότητα μελλοντικής εξέλιξης της εφαρμογής, με έμφαση στη συνεχή βελτίωση της εμπειρίας χρήστη και την προσθήκη νέων λειτουργιών, ώστε να καλύπτονται οι ανάγκες μαθητών και φοιτητών για αποτελεσματική οργάνωση και βελτιστοποίηση της ακαδημαϊκής απόδοσης.

Το πρόβλημα που αντιμετωπίζεται αφορά τις δυσκολίες στη διαχείριση του χρόνου από φοιτητές, λόγω των πολλαπλών υποχρεώσεων της ακαδημαϊκής ζωής. Οι προθεσμίες εργασιών, οι ημερομηνίες εξετάσεων και η ανάγκη καθημερινής μελέτης, συχνά συνδυαζόμενες με ανεπαρκή οργάνωση, οδηγούν σε άγχος και μειωμένη αποδοτικότητα. Η απουσία εργαλείων για προσωποποιημένο προγραμματισμό, παρακολούθηση προθεσμιών και αξιολόγηση της απόδοσης επιδεινώνεται από την αδυναμία σωστής οργάνωσης της ύλης και της προετοιμασίας.

Η ανάγκη για προσωποποιημένα πλάνα μελέτης, σχεδιασμένα με βάση των ατομικών αναγκών, γίνεται προφανής, με την έλλειψη στατιστικών αναλύσεων να δυσχεραίνει την παρακολούθηση της προόδου. Το άγχος, επιδεινούμενο από την έλλειψη εργαλείων αυτοαξιολόγησης, καθιστά την επίτευξη στόχων πιο δύσκολη.

Μέσω της εφαρμογής, παρέχονται προσωποποιημένα πλάνα μελέτης, παρακολουθούμενες προθεσμίες και δυνατότητες επαναληπτικής προετοιμασίας. Ενσωματώνοντας στατιστικά στοιχεία και εργαλεία αξιολόγησης, μειώνεται το άγχος, ενώ βελτιώνεται η οργάνωση και η ακαδημαϊκή απόδοση. Με αυτόν τον τρόπο, καλύπτονται οι ανάγκες φοιτητών για καλύτερη διαχείριση του χρόνου και βελτιστοποίηση της μελέτης.

# Λειτουργικότητα Λογισμικού

## Περιγραφή χαρακτηριστικών

Οι χρήστες μπορούν, μέσω της εφαρμογής, να δημιουργήσουν το κατάλληλο πρόγραμμα λαμβάνοντας τα μαθήματα, τις εργασίες και τη διαθεσιμότητά τους κάθε ημέρα. Η εφαρμογή διαθέτει πλήθος λειτουργιών που στοχεύουν στη δημιουργία του βέλτιστου προγράμματος του χρήστη. Κύρια χαρακτηριστικά είναι τα εξής:

1. Εισαγωγή Μαθημάτων και Εργασιών
  - Ο χρήστης μπορεί να εισάγει τα μαθήματα που έχει να διαβάσει, καταχωρώντας τα παρακάτω πεδία:
    - ο Όνομα μαθήματος.
    - ο Αριθμό συνολικών σελίδων διαβάσματος.
    - ο Ημερομηνία εξεταστικής.
    - ο Δυσκολία του μαθήματος.
    - ο Χρόνο (σε λεπτά) που χρειάζεται για να διαβάσει 20 διαφάνειες.
  - Ο χρήστης μπορεί να καταχωρήσει και εργασίες, συμπληρώνοντας τα εξής:
    - ο Τίτλος εργασίας.
    - ο Εκτιμώμενες ώρες για την ολοκλήρωση της εργασίας.
    - ο Ημερομηνία παράδοσης.
    - ο Δυσκολία εργασίας.
2. Προσδιορισμός Διαθεσιμότητας Χρήστη
  - Ο χρήστης ορίζει τις διαθέσιμες ώρες μελέτης για την κάθε ημέρα.
  - Μια ακόμη δυνατότητα που του παρέχεται είναι, μέσω ενός ενσωματωμένου ημερολογίου, να επιλέξει τις ημερομηνίες στις οποίες δεν είναι διαθέσιμος.
3. Ευρετικός Αλγόριθμος για Βελτιστοποίηση Προγράμματος
  - Χρησιμοποιείται ένα ευρετικός αλγόριθμος ο οποίος:
    - ο Υπολογίζει 50 πιθανά προγράμματα μελέτης.
    - ο Βαθμολογεί κάθε πρόγραμμα με συγκεκριμένα κριτήρια.
    - ο Παρουσιάζει στον χρήστη το πρόγραμμα με την υψηλότερη βαθμολογία.
4. Υπολογισμός Επαναλήψεων βάσει της τεχνικής Spaced Repetition
  - Μετά από κάθε συνεδρία διαβάσματος του χρήστη δημιουργείται αυτόματα μια σειρά από επαναλήψεις ενισχύοντας την αποθήκευση της πληροφορίας στη μνήμη.
5. Προβολή και ανανέωση εργασιών στο Ημερολόγιο
  - Κάνοντας κλικ σε κάθε πεδίο της ημέρας στο ημερολόγιο, εμφανίζεται ένα αναδυόμενο παράθυρο με αναλυτικές πληροφορίες για την προγραμματισμένη του δραστηριότητα.
  - Με την ολοκλήρωση κάθε δραστηριότητας, μπορεί να ενημερωθεί το σύστημα ότι έχει ολοκληρωθεί το διάβασμα του.

- Παρέχεται ακόμη η δυνατότητα ανανέωσης του προγράμματος εάν ο χρήστης κρίνει πως δεν τον εξυπηρετεί ή δεν του ταιριάζει.

#### 6. Προβολή Στατιστικών μέσω ενός Dashboard

- Ο χρήστης μπορεί να παρακολουθεί σημαντικά στατιστικά στοιχεία, όπως:
  - ο Πραγματικές ώρες διαβάσματος ανά ημέρα.
  - ο Αριθμός δραστηριοτήτων που έχουν ολοκληρωθεί για κάθε μάθημα και εργασία.
  - ο Τον αριθμό των εβδομαδιαίων ολοκληρωμένων δραστηριοτήτων για μαθήματα, εργασίες και επαναλήψεις.

#### 7. Λίστες δραστηριοτήτων και πληροφοριών

- Η εφαρμογή παρουσιάζει λίστες που περιλαμβάνουν:
  - ο Μαθήματα, εργασίες και τη δηλωμένη διαθεσιμότητα του χρήστη.
  - ο Σημερινές, εβδομαδιαίες, ολοκληρωμένες και καθυστερημένες δραστηριότητες.

#### 8. Διαχείριση Δεδομένων

- Χρησιμοποιείται μια βάση δεδομένων SQLite για την αποθήκευση δεδομένων χρηστών, που περιλαμβάνουν:
  - ο Όνομα χρήστη, email και κωδικό πρόσβασης ώστε να υπάρχουν προσωπικοί λογαριασμοί.
  - ο Μαθήματα, εργασίες, διαθεσιμότητα καθώς και όλα τα στοιχεία του προγράμματος.

## Απαιτήσεις Λογισμικού

### Απαιτήσεις Λογισμικού:

Η λειτουργία της εφαρμογής απαιτεί την εγκατάσταση του Java Development Kit (JDK) 17, το οποίο χρησιμοποιείται για την εκτέλεση και τη διαχείριση εφαρμογών Java και JavaFX 17. Το JDK πρέπει να είναι σωστά διαμορφωμένο στο PATH του συστήματος. Υποστηρίζονται μόνο λειτουργικά συστήματα με 64-bit αρχιτεκτονική, όπως τα Windows 10 ή νεότερα, MacOS 11 (Big Sur) ή νεότερα, και διανομές Linux, όπως το Ubuntu 20.04 και το Fedora 34. Τα αρχεία JAR εκτελούνται μέσω του Java Runtime Environment (JRE) ή του εγκατεστημένου JDK 17.

Για τη διαχείριση των αρχείων JAR, μπορούν να χρησιμοποιηθούν προγράμματα όπως το 7-Zip ή το WinRAR. Η JavaFX 17, απαραίτητη για τη γραφική διεπαφή, ενσωματώνεται στο JDK 17 ή εγκαθίσταται ξεχωριστά.

### Απαιτήσεις Υλικού:

Για την εκτέλεση της εφαρμογής, απαιτείται σύγχρονο σύστημα με τις εξής ελάχιστες προδιαγραφές:

- **Windows (64-bit):**
  - Επεξεργαστής διπύρηνος ή ανώτερος.
  - Μνήμη RAM τουλάχιστον 4 GB, με συνιστώμενα 8 GB.

Ελεύθερος αποθηκευτικός χώρος 500 MB για την εφαρμογή και το JDK, με συνιστώμενο 1 GB.

- **MacOS:**

Επεξεργαστής Apple Silicon (M1/M2) ή Intel (64-bit).

Μνήμη RAM τουλάχιστον 4 GB, με συνιστώμενα 8 GB.

Αποθηκευτικός χώρος ελάχιστος 500 MB, συνιστώμενος 1 GB.

- **Linux:**

Διπύρηνος επεξεργαστής ή ανώτερος (64-bit).

Μνήμη RAM τουλάχιστον 4 GB, με συνιστώμενα 8 GB.

Ελεύθερος χώρος αποθήκευσης 500 MB, συνιστώμενος 1 GB.

Η υποστήριξη για συστήματα 32-bit αποκλείεται, λόγω της ανάγκης για JDK 17 και νεότερες εκδόσεις, οι οποίες λειτουργούν μόνο σε αρχιτεκτονικές 64-bit. Η εκτέλεση του JAR αρχείου πραγματοποιείται μέσω της εντολής:

```
java -jar <όνομα_αρχείου>.jar
```

## Περιορισμοί

### Προϋποθέσεις λειτουργίας εφαρμογής:

- Μπορεί να εκχωρήσει 1 exam ανά μάθημα.
- Δεν μπορεί να υπάρχει μάθημα χωρίς exam αλλά υφίσταται να υπάρχει μάθημα χωρίς εργασία.
- Ο χρήστης μπορεί να προσθέσει και μια εργασία η οποία δεν εξαρτάται από μάθημα.
- Οι εργασίες δε σχετίζονται με κάποιο μάθημα. Αν ο χρήστης θέλει να εκχωρήσει περισσότερες από μια εργασίες για ένα μάθημα μπορεί να εκχωρεί τις υπόλοιπες ως ανεξάρτητες (ο χρήστης έχει τη δυνατότητα να ονοματίσει 2 ή περισσότερες εργασίες με το ίδιο όνομα. Ωστόσο αυτό θα καθιστά τη διάκρισή τους ανέφικτη. Συνεπώς συνίσταται η χρήση διαφορετικών ονομάτων για διαφορετικά μαθήματα και εργασίες).
- Οι εργασίες ανατίθενται στο πρόγραμμα ανεξαρτήτως διαθεσιμότητας. Αναλυτικότερα: αν ο χρόνος του χρήστη επαρκεί για την κατανομή των εργασιών του διαβάσματος τότε η κατανομή των εργασιών χρονικά μέσα στη μέρα εξαρτάται από τη διαθεσιμότητα. Διαφορετικά κατανέμονται όσες είναι επιτρεπτό βάση ημερήσιας χρονικής διάθεσης του χρήστη και ακολούθως τοποθετούνται τα υπολειπόμενες εργασίες επιπρόσθετα των ημερήσιων χρονικών ορίων του χρήστη προκειμένου να ανατεθούν όλα πριν την ημερομηνία της παράδοσης της εργασίας.
- Σημ. το συνολικό προτεινόμενο πρόγραμμα μιας ημέρας ενδέχεται να ξεπερνά αυτό τον αριθμό συμπεριλαμβανομένων και των εργασιών διαβάσματος και επαναλήψεως που έχουν ανατεθεί στο χρήστη που έχει δηλώσει ο χρήστης γιατί οι εργασίες ανατίθενται στο πρόγραμμα υποχρεωτικά.
- Το πρόγραμμα που προκύπτει δεν είναι το βέλτιστο μεταξύ όλων των πιθανών διαφορετικών συνδυασμών προγραμμάτων αλλά το καλύτερο δυνατό βάση ενός δείγματος προγραμμάτων που περιορίζεται σε αριθμό πενήντα. Αυτό γιατί η φύση του προβλήματος δεν επιτρέπει την εύρεση ενός βέλτιστου προγράμματος που θα κατανείμει με τις εργασίες διαβάσματος και εργασιών καταλαμβάνουν 2 ώρες ενώ τις εργασίες επανάληψης που αντιστοιχούν σε κάθε εργασίες

διαβάσματος ανέρχονται στα 20λ ( όταν υπάρχουν πολλές εργασίες επανάληψης του ίδιου μαθήματος σε μια ημέρα (>2) αυτά συνενώνονται).

- Σε περίπτωση μονού αριθμού ωρών που απαιτούνται για μια εργασία, οι εργασίες που υπολογίζονται στρογγυλοποιούνται στον αμέσως επόμενο ακέραιο αριθμό ( πχ αν για μια εργασία απαιτούνται 13 ώρες των εργασιών που αναθέτει το πρόγραμμα στο χρήστη είναι 7.  $13/2=6,5$  και με στρογγυλοποίηση 7).
- Η εφαρμογή δεν διαθέτει μηχανισμούς κρυπτογράφησης για τη βάση δεδομένων. Δεν έχουν υλοποιηθεί πρόσθετα μέτρα ασφαλείας για την προστασία των στοιχείων των χρηστών, γεγονός που καθιστά τα δεδομένα ευάλωτα.
- Δεν υπήρξε επαρκής χρόνος για εκτεταμένες δοκιμές (testing), έτσι ενδέχεται να προκύψουν σφάλματα που δεν έχουν εντοπιστεί
- Σε περίπτωση που μια δραστηριότητα επανάληψης προγραμματιστεί για μια ημερομηνία μη διαθεσιμότητας του χρήστη, η δραστηριότητα παραλείπεται και θα εμφανιστεί η επόμενη προγραμματισμένη δραστηριότητα επανάληψης
- Το δεξί panel που εμφανίζεται στις σελίδες Calendar και Dashboard δεν λειτουργεί πλήρως, καθώς απαιτείται η ολοκλήρωση της σύνδεσής του με τον αντίστοιχο κώδικα.

## Ροές Εργασίας (Workflows)

### Εισαγωγή νέου χρήστη:

**Σκοπός:** Αυτή η ροή εργασίας περιγράφει τη διαδικασία δημιουργίας νέου χρήστη, συμπεριλαμβανομένων των βημάτων που ακολουθεί ο χρήστης και των ελέγχων του συστήματος για να διασφαλιστεί η εγκυρότητα των δεδομένων.

### Ενέργειες Χρήστη:

1. Ο χρήστης εισέρχεται στην εφαρμογή.
2. Επιλέγει είτε: "Σύνδεση" (αν διαθέτει ήδη λογαριασμό), ή "Δοκίμασε το!" (για εγγραφή).
3. Το σύστημα εμφανίζει μια φόρμα με 4 πεδία:
  - Name
  - Email
  - Password
  - Confirm Password
4. Ο χρήστης εισάγει τα δεδομένα του στα πεδία, σύμφωνα με τους παρακάτω περιορισμούς:
  - Όνομα: Επιτρέπονται μόνο γράμματα (ελληνικά ή αγγλικά), αριθμοί και παύλες.
  - Email: Πρέπει να έχει τη μορφή [test@example.com](mailto:test@example.com).
  - Password:
    - Τουλάχιστον 8 χαρακτήρες.
    - Ένα κεφαλαίο γράμμα.
    - Ένα πεζό γράμμα.
    - Ένα ειδικό σύμβολο (π.χ., @, #, \$).

- ο Έναν αριθμό.
5. Επιλέγει το κουμπί "Εγγραφή".

#### Ενέργειες συστήματος:

1. Όταν ο χρήστης επιλέγει το κουμπί "Εγγραφή" Το σύστημα ελεγχθεί τα στοιχεία του χρήστη και στην περίπτωση μη αποδεκτών στοιχείων του εμφανίζει ένα αναδυόμενο παράθυρο με εξήγηση για την δομή των επιτρεπόμενων στοιχείων εισαγωγής.
2. Στην περίπτωση που τα δεδομένα είναι έγκυρα τότε εμφανίζεται στον χρήστη ένα αντίστοιχο αναδυόμενο παράθυρο σαν αυτό των error με ένα μήνυμα επιτυχίας . Το οποίο υποδουλώνει πως ο χρήστης εισήχθη επιτυχώς στην βάση δεδομένων.
3. Εισαγωγή Μαθήματος, Εργασιών και διαθεσιμότητας του νέου χρήστη.

**Σκοπός:** Αυτή η ροή εργασίας περιγραφή τα επόμενα βήματα που χρειάζεται να κάνει ο χρήστης αφού έχει εγγραφή για πρώτη φορά στο σύστημα.

#### Ενέργειες χρήστη:

1. Το σύστημα κατευθύνει τον χρήστη στη σελίδα εισαγωγής μαθημάτων.
2. Το σύστημα εμφανίζει μία φόρμα με 7 υποχρεωτικά πεδία προς συμπλήρωση.
3. Ο χρήστης έχει την επιλογή είτε να προσθέσει όσα μαθήματα επιθυμεί (ένα προς ένα) είτε να πατήσει το κουμπί "Επόμενο" και να εισαγάγει τα μαθήματα στη συνέχεια.
4. Αν επιλέξει να προσθέσει μάθημα, τότε πρέπει να συμπληρώσει τα 7 πεδία σύμφωνα με τους περιορισμούς:
  - ο Ονομασία Μαθήματος: Το όνομα επιτρέπεται να έχει μόνο ελληνικούς ή αγγλικούς χαρακτήρες, όπως και αριθμούς.
  - ο Είδος: Το είδος δέχεται μόνο την τιμή άγνωστο.
  - ο Σελίδες: Αυτό το πεδίο δέχεται μόνο αριθμούς.
  - ο Επανάληψη ανά (Σελίδες): Αυτό το πεδίο δέχεται μόνο αριθμούς.
  - ο Ημερομηνία Εξέτασης: Αυτό το πεδίο δέχεται μόνο ημερομηνίες της μορφής (dd/MM/yyyy).
  - ο Δυσκολία: Αυτό το πεδίο δέχεται μόνο αριθμούς μεταξύ 1 - 10.
  - ο Χρόνος ανά 20 διαφάνειες: Αυτό το πεδίο δέχεται μόνο αριθμούς.
5. Όταν ο χρήστης εισαγάγει τα δεδομένα του μαθήματος, τότε με την επιλογή του κουμπιού "ΟΚ", το σύστημα εμφανίζει, στην περίπτωση που τα δεδομένα δεν πληρούν τους περιορισμούς, ένα αναδυόμενο παράθυρο με τα πεδία τα οποία χρειάζεται να αλλάξει.
6. Αν τα δεδομένα είναι ορθά, τότε το σύστημα εμφανίζει μήνυμα επιτυχίας προσθήκης μαθήματος.



7. Στη συνέχεια, ο χρήστης μπορεί να επαναλάβει αυτήν τη διαδικασία έως ότου να προσθέσει όσα μαθήματα επιθυμεί.
8. Αφού έχουν προστεθεί τα μαθήματα από τον χρήστη, τότε, πατώντας το κουμπί "Επόμενο", ο χρήστης κατευθύνεται στη σελίδα εισαγωγής εργασιών.
9. Το σύστημα εμφανίζει 5 πεδία με τους εξής περιορισμούς:
  - ο Μάθημα Εργασίας: Το μάθημα μπορεί να πάρει 3 συγκεκριμένες τιμές.
  - ο Τίτλος Εργασίας: Ο τίτλος επιτρέπεται να έχει μόνο χαρακτήρες.
  - ο Εκτιμώμενες απαιτούμενες ώρες: Αυτό το πεδίο δέχεται μόνο αριθμούς.
  - ο Ημερομηνία Παράδοσης: Αυτό το πεδίο δέχεται μόνο ημερομηνίες της μορφής (dd/MM/yyyy).
  - ο Δυσκολία: Αυτό το πεδίο δέχεται μόνο αριθμούς μεταξύ 1 - 10.
10. Ο χρήστης σε αυτήν τη σελίδα μπορεί να εισαγάγει όσες εργασίες επιθυμεί (μία προς μία). Στην περίπτωση εισαγωγής σωστών δεδομένων, το σύστημα εμφανίζει μήνυμα επιτυχίας, αλλιώς μήνυμα με τους περιορισμούς που υπάρχουν στο πεδίο στο οποίο υπήρχε το λάθος.
11. Αφού ο χρήστης τελειώσει με την εισαγωγή εργασιών ή επιλέξει να μην εισαγάγει καμία εργασία, πατώντας το κουμπί "Επόμενο", εισέρχεται στη σελίδα διαθεσιμότητας.
12. Σε αυτήν τη σελίδα έχει την επιλογή να εισαγάγει τις ώρες που έχει διαθέσιμες κάθε μέρα της εβδομάδας, όπως και την ημέρα που είναι μη διαθέσιμος.
13. Ο χρήστης, αν το επιθυμεί, εισάγει τη διαθεσιμότητά του με τον εξής περιορισμό: οι διαθέσιμες ώρες ανά ημέρα να είναι μικρότερες του 10.
14. Στην περίπτωση που ο χρήστης επιλέξει να αφήσει κενά τα πεδία της διαθεσιμότητας των ημερών, παρ' όλα αυτά επιλέξει το κουμπί "ΟΚ", τότε του εμφανίζεται ένα μήνυμα επιβεβαίωσης αν είναι σίγουρος πως θέλει να αφήσει όλες τις μέρες κενές (δηλαδή καθημερινή διαθεσιμότητα μηδενική).
15. Αν ο χρήστης εισαγάγει σωστά τα δεδομένα, τότε του εμφανίζεται μήνυμα επιτυχίας και η διαθεσιμότητα περνάει στη βάση δεδομένων.
16. Αν ο χρήστης δεν επιθυμεί να εισαγάγει τη διαθεσιμότητά του εκείνη τη στιγμή, τότε μπορεί να επιλέξει το κουμπί "Επόμενο", όπου του εμφανίζεται ένα εισαγωγικό μήνυμα καλωσορίσματος.
17. Αφού ο χρήστης επιλέξει "ΟΚ" στο μήνυμα καλωσορίσματος, εισέρχεται στο ημερολόγιο.

#### **Εισαγωγή υπάρχων χρήστη:**

**Σκοπός:** Αυτή η ροή εργασίας περιγράφει την σύνδεση υπάρχων χρήστη στο σύστημα.

1. Ο χρήστης επιλέγει το κουμπί "Σύνδεση".
2. Το σύστημα εμφανίζει μια φόρμα με δυο πεδία:
  - ο Email

- Password

3. Επιλεγεί το κουμπί " Συνδέσου ".
4. Στην περίπτωση που ο χρήστης εισάγει κωδικό και email που δεν συμβαδίζει με εκείνα της βάσης τότε του εμφανίζει μήνυμα σφάλματος για λάθος κωδικό ή email πρόσβασης. Το ίδιο μήνυμα εμφανίζεται και στην περίπτωση που δεν υπάρχει ο χρήστης αποθηκευμένος στην βάση.

#### **Ενέργειες συστήματος:**

1. Αν η σύνδεση είναι επιτυχής το σύστημα ανακατευθύνει τον χρήστη στην σελίδα ημερολόγιο όπου μπορεί να δει το εβδομαδιαίο πρόγραμμα του μέχρι την εξεταστική.

#### **Δημιουργία Προγράμματος:**

**Σκοπός:** Αυτή η ροή εργασίας περιγράφει τα βήματα που μπορεί να κάνει ο χρήστης στο ημερολόγιο αφού έχει προηγηθεί είτε εγγραφή, είτε σύνδεση.

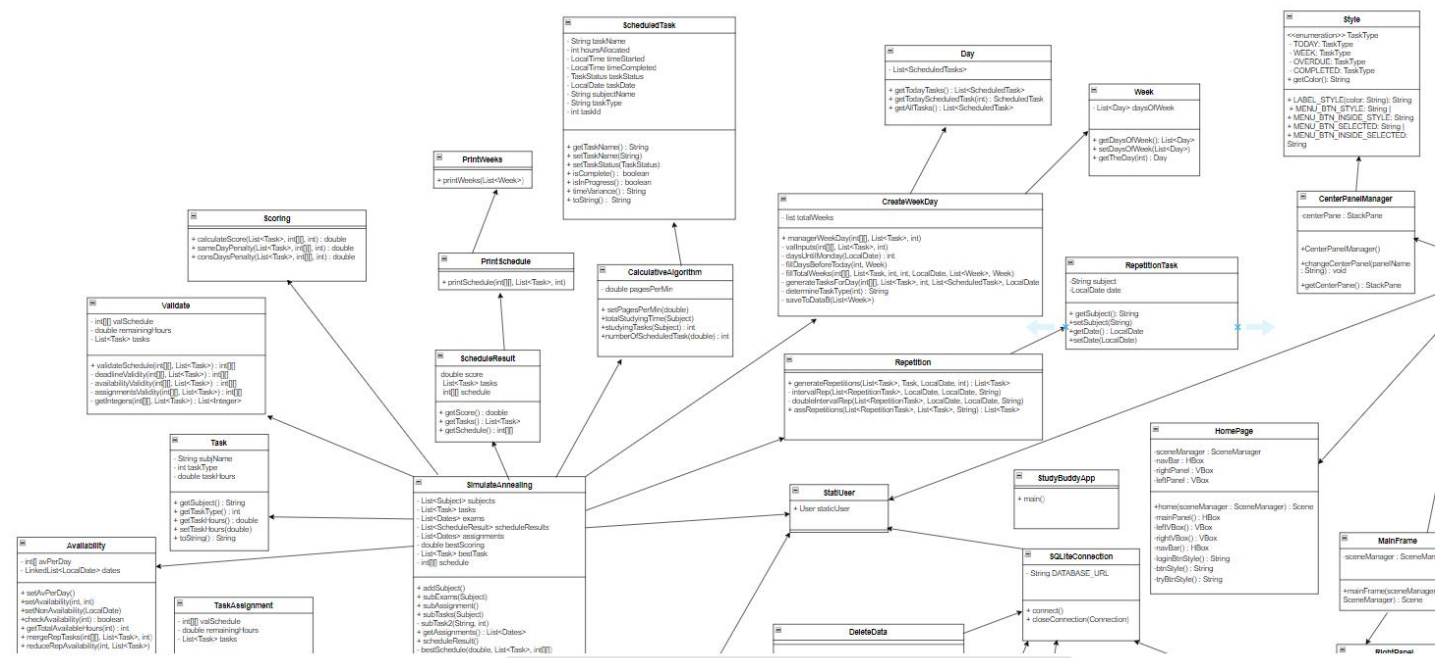
#### **Ενέργειες:**

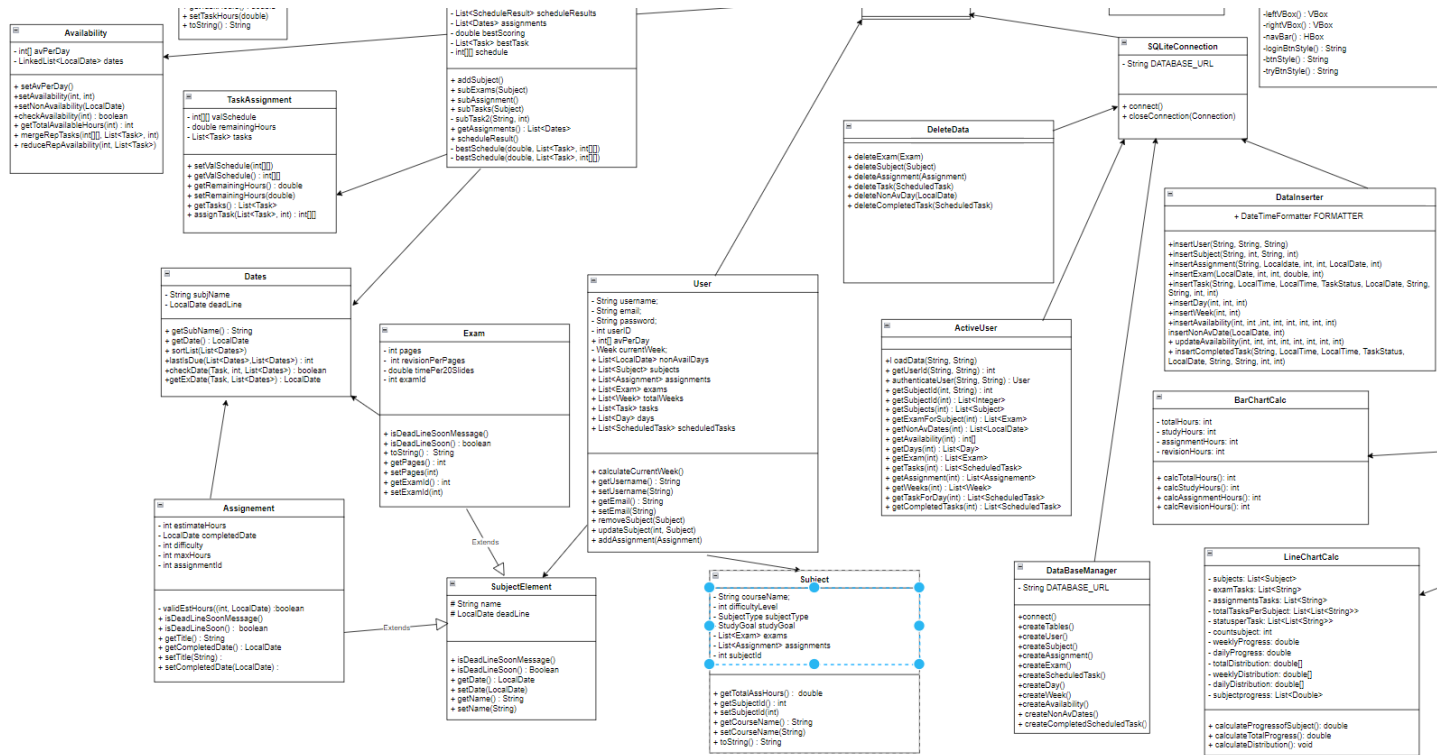
1. Ο χρήστης πατάει το κουμπί regenerate ώστε να δημιουργηθεί το βέλτιστο πρόγραμμα με βάση τον αλγόριθμο.
2. Το σύστημα του εμφανίζει ένα warning αναδυόμενο παράθυρο με τις επιλογές OK και Cancel. Στην περίπτωση που θέλει να δημιουργήσει ένα νέο πρόγραμμα ή είναι η πρώτη φορά που εισέρχεται μετά την αποσύνδεση στο σύστημα, τότε χρειάζεται να επιλέξει "OK". Αλλιώς, αν δεν θέλει να του εμφανιστεί άλλο πρόγραμμα, μπορεί να πατήσει "Cancel".
3. Το σύστημα εμφανίζει το πρόγραμμα του χρήστη για την εβδομάδα που διανύει. Από εκεί, ο χρήστης έχει την επιλογή να πατήσει πάνω σε ένα κελί με κάποιες εργασίες, ώστε να δει πόσες ώρες διάβασμα, εργασία ή επανάληψη χρειάζεται να κάνει για το συγκεκριμένο μάθημα, τότε είναι το μάθημα, καθώς και πόσες μέρες του απομένουν μέχρι να δοθεί το μάθημα/εργασία. Επίσης, αν ολοκληρώσει τις εργασίες, μπορεί να επιλέξει Finished, ώστε να φανεί η συγκεκριμένη εργασία ως ολοκληρωμένο.
4. Ο χρήστης έχει τη δυνατότητα να προηγηθεί και στις επόμενες εβδομάδες και να δει το πρόγραμμά του σε αυτές, μέχρι την εβδομάδα της εξεταστικής, επιλέγοντας το κουμπί με το σύμβολο Επόμενο (>) είτε το κουμπί με το σύμβολο Προηγούμενο (<).
5. Επίσης, αν θέλει να επιστρέψει στην εβδομάδα την οποία διανύει και να συνεχίσει την αναζήτηση από εκεί, μπορεί να επιλέξει το κουμπί Today.

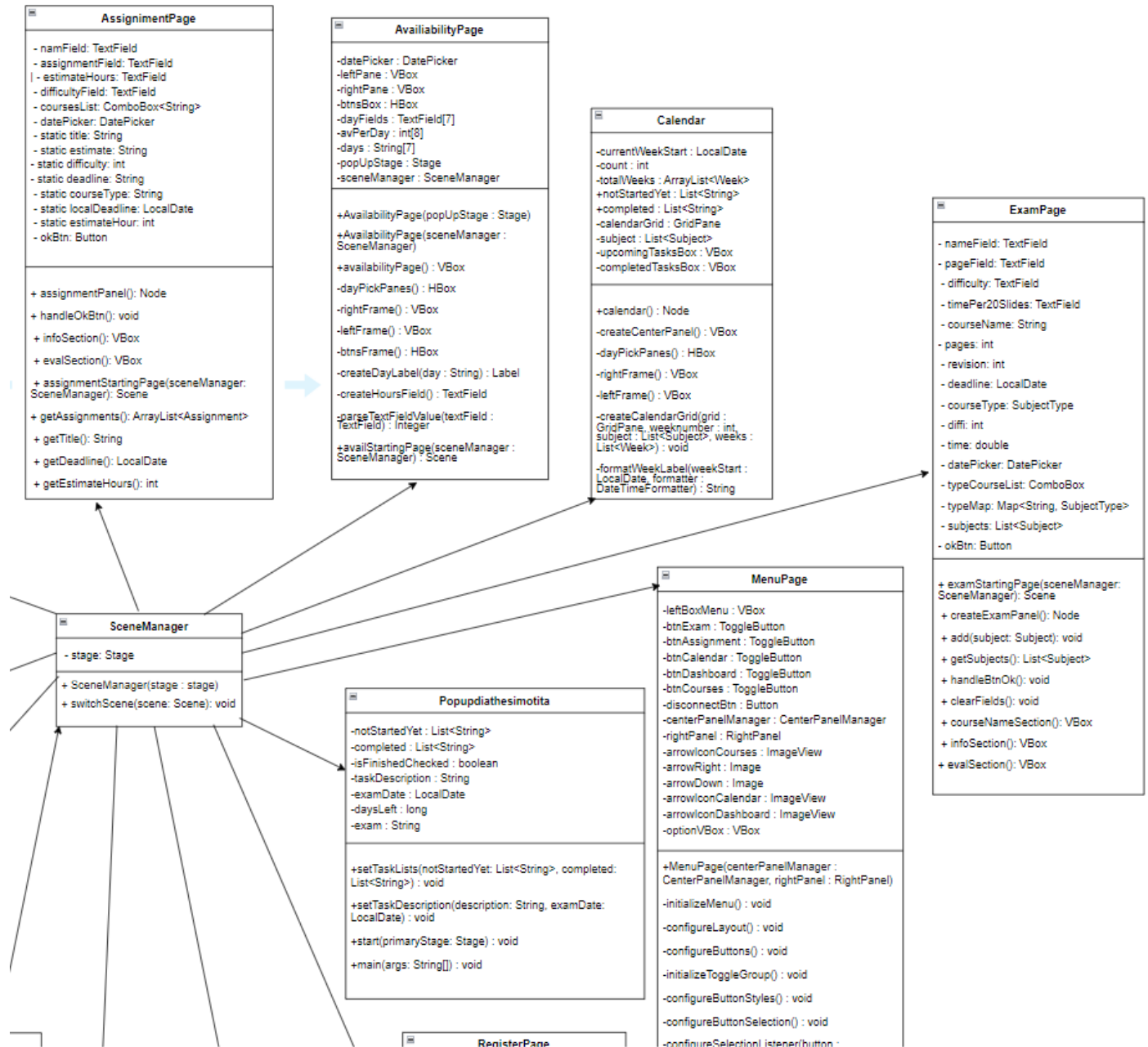
6. Από εκεί, ο χρήστης, αν θέλει να προσθέσει κάποιο μάθημα είτε κάποια εργασία, μπορεί να επιλέξει από το menu το label Courses, το οποίο θα του εμφανίσει είτε την επιλογή να πάει στη σελίδα που προσθέτεις μάθημα, είτε την επιλογή να πάει στη σελίδα που προσθέτεις εργασία. Στις δύο αυτές σελίδες εμφανίζονται τα πεδία εισαγωγής μαθημάτων/εργασιών, όπως περιεγράφηκαν πιο πάνω, καθώς επίσης και στη δεξιά πλευρά της φόρμας εμφανίζονται τέσσερα κουτιά, στα οποία μπορεί να δει τα μαθήματα που έχει εισαγάγει, τις εργασίες, τη διαθεσιμότητά του, όπως και τις ημέρες που έχει δηλώσει μη διαθεσιμότητα.
7. Από το ημερολόγιο, επίσης, ο χρήστης μπορεί να αλλάξει τη διαθεσιμότητά του και τις μη διαθέσιμες ημέρες του.
8. Επίσης, ο χρήστης μπορεί να προηγηθεί στη σελίδα “Dashboard”, αν το επιλέξει από το menu, στην οποία εμφανίζονται τα στατιστικά του στοιχεία σχετικά με τις ολοκληρωμένες του εργασίες, με τον χρόνο που καταλαμβάνει το κάθε μάθημα και εργασία συγκριτικά με τα άλλα μαθήματα/εργασίες, αντίστοιχα, αναπαριστώμενα από γραφήματα σε μορφή πίτας. Ακόμα, υπάρχει ένα γράφημα για τις συνολικές του εργασίες και ένα ακόμα για το πόσες φορές διάβασε μέσα στην εβδομάδα.
9. Τέλος, ο χρήστης από το ημερολόγιο και από το Dashboard βλέπει τις σημερινές του εργασίες, τις εργασίες όλης της εβδομάδας, καθώς και τις ολοκληρωμένες εργασίες και αυτά που είναι μη ολοκληρωμένα.

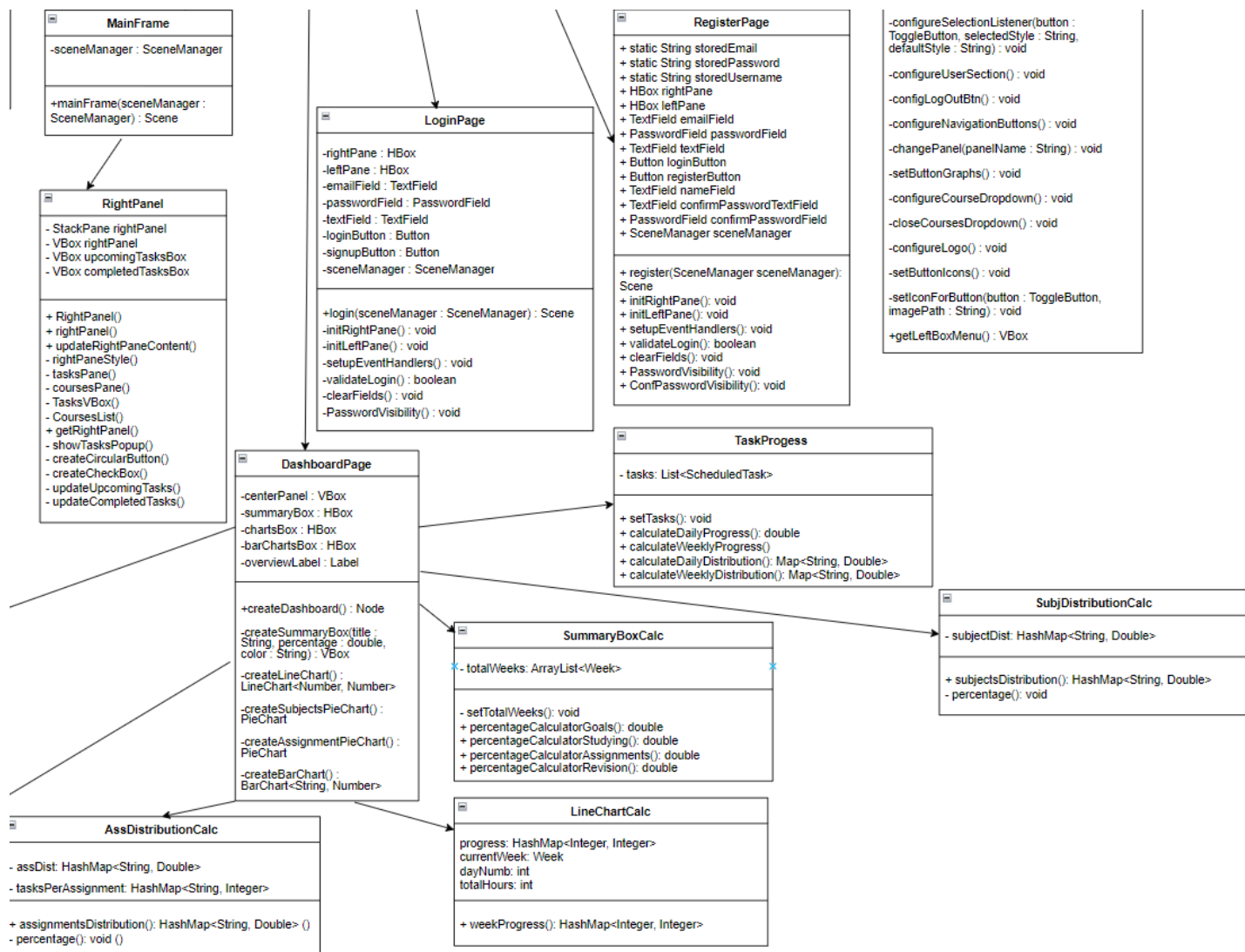
## Αρχιτεκτονική και Σχεδίαση

### Αρχιτεκτονική Συστήματος - UML



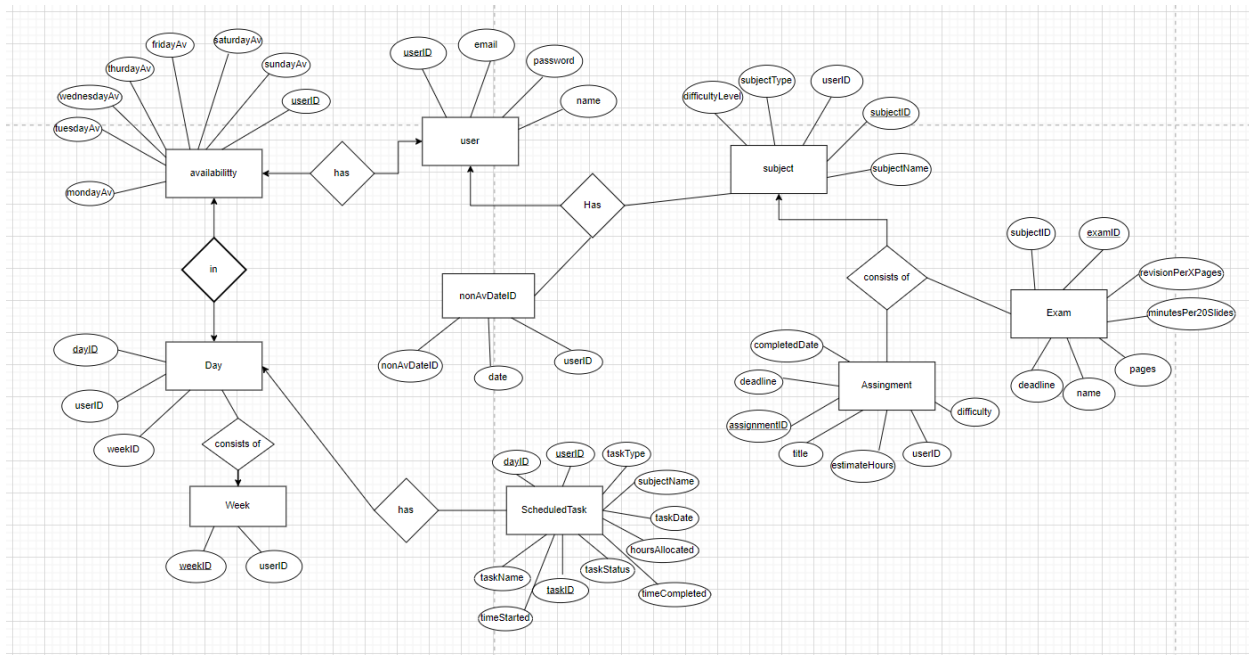






## Σχέδιο βάσης δεδομένων

Η βάση δεδομένων της παρούσας εφαρμογής έχει στόχο την αποθήκευση χρηστών, μαθημάτων και όλων των δεδομένων που χρειάζονται για την δημιουργία του προτεινόμενου προγράμματος του διαβάσματος για τον χρήστη. Το σύστημα διαχείρισης βάσης δεδομένων που χρησιμοποιείται σε αυτή την εφαρμογή είναι SQLite. Το SQLite είναι ένα ενσωματωμένο σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS) και δεν βασίζεται σε έναν ξεχωριστό διακομιστή (server). Παρακάτω είναι το διάγραμμα οντοτήτων – συσχετίσεων και το σχεσιακό μοντέλο όπου απεικονίζονται όλοι οι πίνακες που χρησιμοποιούνται και τα δεδομένα που αποθηκεύονται και μπορούν να ανακληθούν.



User (userID, password, email, name)

subject (subjectID, subjectName, difficultyLevel, subjectType, userID)

assignment (assignmentID, title, deadline, estimateHour, completedDate, userID, difficulty)

exam(examID, name, deadline, pages, revisionPerXPages, minutesPer20Slides, subjectID)

availability (userID, mondayAv, tuesdayAv, wednesdayAv, thursdatAv, fridayAv, saturdayAv, sundatAv)

nonAvDate(nonAvDateID, date, userID)

day (dayID, userID, weekID)

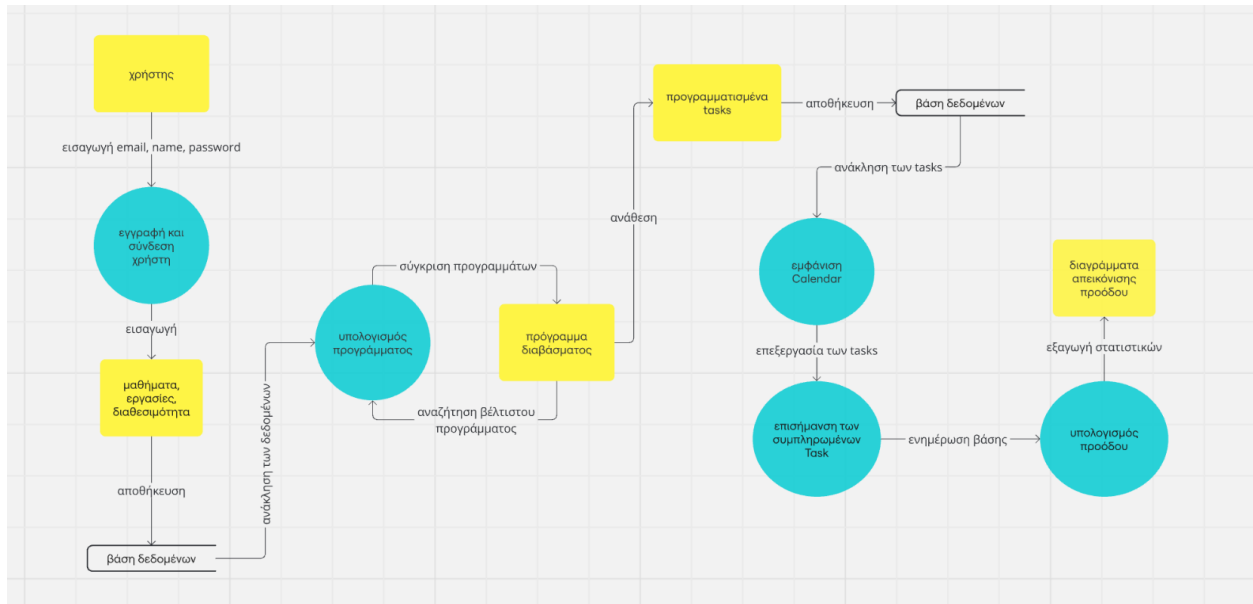
week (weekID, userID)

Task (taskID, taskName, hoursAllocated, timeStarted, timeCompleted, taskStatus, taskDate, subjectName, taskType, userID, dayID)



## Διάγραμμα ροής δεδομένων

### Data Flow Diagrams



## Προδιαγραφές Λογισμικού

### Τεχνολογίες Υλοποίησης

Λειτουργικά συστήματα, βιβλιοθήκες, γλώσσες προγραμματισμού που χρησιμοποιούνται.

Οι τεχνολογίες που χρησιμοποιούνται (π.χ., γλώσσες προγραμματισμού, βάσεις δεδομένων, πλατφόρμες).

- Για την υλοποίηση της εφαρμογής χρησιμοποιούνται οι εξής τεχνολογίες:
  - ο Γλώσσα προγραμματισμού : Java 23.
  - ο Για το γραφικό περιβάλλον χρήστη : JavaFX 17.0.6, controlsfx 11.2.1 και charm-glisten 6.2.3 για επέκταση της βιβλιοθήκης και τη δημιουργία μοντέρνων εφαρμογών JavaFx.
  - ο Για τη βάση δεδομένων : SQLite JDBC 3.47.1.0 για τη σύνδεση με τη βάση δεδομένων SQLite.
  - ο Για την κατασκευή και ανάπτυξη : maven-surefire-plugin 2.22.2 για την υλοποίηση των unit test.
  - ο Για τον έλεγχο και τις δοκιμές : junit.jupiter για την δημιουργία των τεστ του κώδικα, Jacoco 0.8.7 για την ανάλυση κάλυψης κώδικα μέσω δοκιμών, owasp 8.3.1 για τον έλεγχο των εξαρτήσεων οι οποίες έχουν αδυναμίες.
  - ο Για την ποιότητα κώδικα : checkstyle 10.21.0 για την τήρηση προτύπων κώδικα, spotless-maven-plugin 2.42.0 για την αυτόματη μορφοποίηση σε συγκεκριμένα πρότυπα στυλ κώδικα.
  - ο Χρησιμοποίηση dependency-check-maven 8.3.1 για τον έλεγχο των εξαρτήσεων για τυχόν εμφάνιση κάποιας αδυναμίας.
  - ο Χρήση maven-pmd-plugin 3.21.0 για την αυτόματη λειτουργία του PMD code εργαλείου.
  - ο Πλατφόρμες και Εργαλεία : χρήση των περιβάλλοντων ανάπτυξης IntelliJ IDEA και VSCode, το GitHub ως σύστημα ελέγχου εκδόσεων, maven ως εργαλείο διαχείρισης έργου.

## Αλγόριθμος

Ο αλγόριθμος που αναπτύχθηκε για την εφαρμογή αποσκοπεί στη δημιουργία ενός βέλτιστου προγράμματος μελέτης. Μέσω μιας αλληλουχίας βημάτων, οι εργασίες, το διάβασμα και οι σχετικές με αυτό επαναλήψεις του χρήστη κατανέμονται στις διαθέσιμες ημέρες, διασφαλίζοντας ότι τηρείται ο χρονικός περιορισμός και οι προθεσμίες που έχουν καταχωρηθεί. Η βάση του αλγορίθμου ανάγεται στη τυχαία δημιουργία πολλαπλών πιθανών και έγκυρων, με βάση τη διαθεσιμότητα του χρήστη, προγραμμάτων καθένα από τα οποία αξιολογείται λαμβάνοντας υπόψη προκαθορισμένα κριτήρια. Στην ανάπτυξη του αλγορίθμου, αξιοποιείται πληθώρα δομών δεδομένων με τις οποίες επιτρέπεται η επεξεργασία, η οργάνωση, η διαχείριση και η αποθήκευση των εργασιών (Task) του χρήστη. Στο τρέχον κεφάλαιο αναλύεται τόσο ο τρόπος που ο κύριος αλγόριθμος του συστήματος έχει δομηθεί όσο και οι δομές δεδομένων που αξιοποιεί.

Ο αλγόριθμος που δημιουργήθηκε αποτελεί έναν ολοκληρωμένο μηχανισμό δημιουργίας ενός προσωποποιημένου προγράμματος μελέτης. Για να εξαχθεί το τελικό αποτέλεσμα ο αλγόριθμος υλοποιεί πέντε (5) βασικές λειτουργίες.

Αρχικά, για την λειτουργία του κύριου αλγορίθμου απαιτούνται επιμέρους υπολογισμοί. Αυτοί προσδιορίζονται μέσω της κλάσης `CalculativeAlgorithm` όπου υπολογίζεται ο συνολικό χρόνο που απαιτείται για να ολοκληρωθεί το διάβασμα ενός μαθήματος, ο αριθμός των απαιτούμενων εργασιών (σε tasks διάρκειας 2 ωρών). Ακόμα, με την κλάση `Dates` επιτυγχάνεται η διαχείριση των ημερομηνιών και των προθεσμιών για τις εξετάσεις και τις εργασίες υπολογίζεται η τελευταία δυνατή ημέρα για κάθε εργασία (`lastIsDue`) και ελέγχεται αν μια εργασία μπορεί να προγραμματιστεί πριν την προθεσμία της (`checkDate`).

Μετά τη δημιουργία τους, οι εργασίες (Task) του εκάστοτε χρήστη κατανέμονται στις διαθέσιμες ημέρες. Η διαδικασία αυτή περιλαμβάνει τη δημιουργία εργασιών (Task) μελέτης για το διάβασμα και της εργασίες του χρήστη. Ακολούθως, γίνεται τυχαία κατανομή τους στις διαθέσιμες δηλωμένες ώρες και ημέρες ενώ συγχρόνως διασφαλίζεται η τήρηση των προθεσμιών. Παράλληλα, υπολογίζονται οι απαραίτητες επαναλήψεις βάση του όγκου διαβάσματος για την καλύτερη αφομοίωση της ύλης. Ο τρόπος υπολογισμού βασίζεται στην έννοια ότι η ικανότητα ανάκλησης πληροφοριών ενισχύεται μέσω επαναλαμβανόμενης έκθεσης σε αυτές. Με την τεχνική της Επαναληπτικής Εξάσκησης (`Spaced Repetition`), οι πληροφορίες παρουσιάζονται σε στρατηγικά χρονικά διαστήματα, βοηθώντας τον χρήστη να τις επεξεργαστεί και να τις εδραιώσει στη μακροπρόθεσμη μνήμη. Καθώς οι επαναλήψεις ενσωματώνονται στη διαδικασία μάθησης, η μνήμη ενισχύεται σταδιακά, χτίζοντας έτσι μια ισχυρότερη ικανότητα ανάκλησης.

Η παραπάνω διεργασία επαναλαμβάνεται πενήντα (50) φορές μέσω της κλάσης `SimulateAnnealing`. Παράλληλα με τη κατανομή πραγματοποιείται η λειτουργία της βαθμολόγησης. Καθένα από τα προτεινόμενα προγράμματα λαμβάνει μια βαθμολογία και στη συνέχεια επιλέγεται το αποτέλεσμα με το μεγαλύτερο βαθμό. Μέσω της κλάσης `Scoring` αποδίδεται στο τρέχον πιθανό πρόγραμμα ποινή (-0.2) όταν το ίδιο είδος εργασίας (π.χ. διάβασμα) για το ίδιο μάθημα εμφανίζεται πολλές φορές μέσα στην ίδια ημέρα και ποινή (-0.1) Όταν το ίδιο μάθημα εμφανίζεται για διάβασμα δύο συνεχόμενες ημέρες. Το καλύτερο πρόγραμμα με τη μεγαλύτερη βαθμολογία επιλέγεται για να δοθεί στον χρήστη ως αποτέλεσμα.

Ακολουθεί η λειτουργία της επικύρωσης του προγράμματος. Μέσω της κλάσης `Validate` ελέγχεται και επιβεβαιώνεται ότι το πρόγραμμα είναι έγκυρο. Πρόκειται στην ουσία για έναν περιττό έλεγχο που διασφαλίζει ωστόσο ότι καμία εργασία δεν προγραμματίζεται μετά την προθεσμία της Καμία εργασία δεν προγραμματίζεται μετά την προθεσμία της (`deadlineValidity`), ότι όλες οι εργασίες καταχωρούνται μόνο σε ώρες που είναι διαθέσιμες για μελέτη (`availabilityValidity`), καθώς και ότι σε περίπτωση που οι εργασίες (`task` τύπου 3 εργασίες) δεν έχουν κατανεμηθεί, κατανέμονται υποχρεωτικά (`assignmentsValidity`).

Τέλος, ο αλγόριθμος ολοκληρώνεται με τη αποθήκευση του προγράμματος σε εύκολα διαχειρίσιμη μορφή. Αυτό επιτυγχάνεται διαμέσου της κλάσης `CreateWeekDay` όπου διασπάται το πρόγραμμα σε εβδομάδες (`Week`) και ημέρες (`Day`). Για κάθε μέρα δημιουργούνται αντικείμενα `ScheduledTask` για τις εργασίες που της έχουν ανατεθεί. Ακολούθως μέσω της κλάσης `StaticUser` καταχωρείται το πρόγραμμα στον τρέχον χρήστη και αποθηκεύεται στη βάση δεδομένων μέσω της μεθόδου `saveToDataB`.

Για την αποδοτικότητα του αλγορίθμου η επιλογή των κατάλληλων δομών δεδομένων είναι βαρύνουσας σημασία. Στο πλαίσιο του συστήματος οι κύριες δομές δεδομένων που αξιοποιούνται είναι οι λίστες για τη διαχείριση σειριακών δεδομένων όπως Εργασίες (`Tasks`), όπου κάθε εργασία αποθηκεύεται σε μια λίστα αντικειμένων τύπου `Task`, οι πίνακες, όπως ο πίνακας `schedule`, ένας δισδιάστατος πίνακας (`12xN`) που οι γραμμές αντιπροσωπεύουν τα διαθέσιμα χρονικά slots της ημέρας (π.χ. 12 ώρες) και οι στήλες αντιπροσωπεύουν τις ημέρες του προγράμματος. Επιπρόσθετα, αξιοποιούνται στατικά δεδομένα (όπως ο `staticUser`), που περιέχει στατικά δεδομένα του χρήστη (π.χ. λίστα μαθημάτων, ημερομηνίες διαθεσιμότητας), τα οποία είναι παγκοσμίως προσβάσιμα από άλλες κλάσεις ενώ αξιοποιούνται δομές δεδομένων όπως αντικείμενα πολλαπλών επιπέδων όπως υλοποιούνται από τις κλάσεις `Week` και `Day`.

## Απαιτήσεις Συστήματος

### Λειτουργικές απαιτήσεις

#### Λειτουργικές Απαιτήσεις

1. Εγγραφή Χρήστη:
  - Το σύστημα επιτρέπει τη δημιουργία νέου λογαριασμού χρήστη.
  - Εμφανίζει φόρμα εγγραφής με τα πεδία:
    - Όνομα (`Name`)
    - Email
    - Κωδικός πρόσβασης (`Password`)
    - Επιβεβαίωση κωδικού πρόσβασης (`Confirm Password`)
2. Ελέγχει ότι τα δεδομένα των πεδίων πληρούν τους ακόλουθους περιορισμούς:
  - Όνομα: Μόνο γράμματα, αριθμοί και παύλες.
  - Email: Έγκυρη μορφή email (π.χ., `test@example.com`).
  - Κωδικός πρόσβασης:
    - Τουλάχιστον 8 χαρακτήρες.
    - Ένα κεφαλαίο γράμμα.
    - Ένα πεζό γράμμα.
    - Ένα ειδικό σύμβολο.

- ο Έναν αριθμό.

Εμφανίζει μήνυμα επιτυχίας ή αποτυχίας με τις αιτίες αποτυχίας.

3. Σύνδεση Χρήστη:

- Το σύστημα επιτρέπει στους χρήστες να συνδεθούν με τον λογαριασμό τους.
- Εμφανίζει φόρμα σύνδεσης με τα πεδία:
  - ο Email
  - ο Κωδικός πρόσβασης.
  - ο Ελέγχει ότι τα στοιχεία του χρήστη υπάρχουν στη βάση δεδομένων.
  - ο Εμφανίζει μήνυμα επιτυχούς σύνδεσης ή σφάλματος (π.χ., "Λανθασμένος κωδικός ή email").

4. Εισαγωγή Μαθημάτων:

- Το σύστημα επιτρέπει στους χρήστες να προσθέτουν μαθήματα.
- Παρέχει φόρμα με τα πεδία:
  - ο Ονομασία Μαθήματος.
  - ο Είδος.
  - ο Σελίδες.
  - ο Επανάληψη ανά (Σελίδες).
  - ο Ημερομηνία Εξέτασης.
  - ο Δυσκολία.
  - ο Χρόνος ανά 20 διαφάνειες.

5. Ελέγχει ότι τα δεδομένα πληρούν τους περιορισμούς:

- Ονομασία Μαθήματος: Γράμματα και αριθμοί μόνο.
- Δυσκολία: Τιμές από 1 έως 10.
- Ημερομηνία Εξέτασης: Έγκυρη μορφή (dd/MM/yyyy).

6. Εισαγωγή Εργασιών

- Το σύστημα επιτρέπει στους χρήστες να προσθέτουν εργασίες.
- Παρέχει φόρμα με τα πεδία:
  - ο Μάθημα Εργασίας.
  - ο Τίτλος Εργασίας.
  - ο Εκτιμώμενες Απαιτούμενες Ώρες.
  - ο Ημερομηνία Παράδοσης.
  - ο Δυσκολία.
  - ο Εμφανίζει μήνυμα επιτυχίας ή αποτυχίας.

- ο Καθορισμός Διαθεσιμότητας
- ο Το σύστημα επιτρέπει στους χρήστες να δηλώνουν τις διαθέσιμες ώρες και μη διαθέσιμες ημέρες.
- ο Ελέγχει ότι οι διαθέσιμες ώρες ανά ημέρα είναι μικρότερες από 10.

#### 7. Δημιουργία Προγράμματος:

- Το σύστημα δημιουργεί ένα βέλτιστο πρόγραμμα για τον χρήστη, βασισμένο στα εισαγόμενα δεδομένα (μαθήματα, εργασίες, διαθεσιμότητα).
- Παρέχει δυνατότητα αναδημιουργίας προγράμματος (κουμπί "Regenerate").
- Επιτρέπει στους χρήστες να βλέπουν το εβδομαδιαίο πρόγραμμα.

#### 8. Παρακολούθηση Προόδου:

- Το σύστημα εμφανίζει στατιστικά δεδομένα (π.χ., ολοκληρωμένες εργασίες, χρόνο αφιερωμένο σε μαθήματα/εργασίες).
- Παρέχει διαγράμματα για την παρουσίαση δεδομένων.

## Μη λειτουργικές απαιτήσεις

Για αυτήν την έκδοση του προγράμματος δεν μπορούν να καθοριστούν η μη-λειτουργικές απαιτήσεις.

## Δοκιμές

### Σχέδιο δοκιμών

Στοχεύοντας στη διασφάλιση της ποιότητας και της ορθής λειτουργίας της εφαρμογής, πραγματοποιήθηκαν διάφορα είδη δοκιμών. Ωστόσο, οι δοκιμές αυτές διεξήχθησαν σε περιορισμένο χρονικό διάστημα και με περιορισμένο αριθμό χρηστών, έτσι ενδέχεται να επηρεαστεί η πλήρης κάλυψη όλων των πιθανών σεναρίων χρήσης.

### Αυτοματοποιημένες Δοκιμές

Χρησιμοποιήθηκαν JUnit tests για βασικό έλεγχο των μεθόδων και των κλάσεων της εφαρμογής, με στόχο τον εντοπισμό σφαλμάτων. Παράλληλα, με τη χρήση του JaCoCo χρησιμοποιήθηκαν οι αναφορές κάλυψης κώδικα, ώστε να διαπιστωθεί κατά πόσο οι δοκιμές καλύπτουν τα βασικά τμήματα της εφαρμογής.

### Χειροκίνητες Δοκιμές

Πραγματοποιήθηκαν χειροκίνητες δοκιμές (manual testing) κάθε φορά που κάθε μέλος της ομάδας προσέθετε καινούρια λειτουργία. Μέσα από αυτή τη διαδικασία εντοπίστηκε πληθώρα προβλημάτων στη ροή της εφαρμογής που δεν μπορούσαν να καλύψουν οι αυτοματοποιημένες δοκιμές. Λόγω περιορισμένου χρόνου, οι δοκιμές αυτές επικεντρώθηκαν στις βασικές λειτουργίες της εφαρμογής.

### Δοκιμές από Χρήστες (User Testing)

Με σκοπό την αξιολόγηση της εμπειρίας χρήσης, μια μικρή ομάδα χρηστών ανέλαβε να χρησιμοποιήσει την εφαρμογή. Οι παρατηρήσεις τους βοήθησαν στον εντοπισμό βελτιώσεων, ιδιαίτερα σε θέματα

ευχρηστίας και ροής λειτουργιών. Ωστόσο, και σε αυτό το στάδιο υπήρξε περιορισμένος αριθμός συμμετεχόντων με αποτέλεσμα να μην έχουν εντοπιστεί όλα τα πιθανά προβλήματα που μπορεί να προκύψουν κατά την εκτενέστερη χρήση της εφαρμογής.

Ο συνδυασμός αυτών των διαφορετικών προσεγγίσεων δοκιμών βελτίωσε τον εντοπισμό σφαλμάτων καθώς και τη συνολική σταθερότητα και απόδοση της εφαρμογής, αλλά η πιθανότητα να υπάρξουν σενάρια χρήσης που δεν έχουν ελεγχθεί παραμένει υψηλή.

## Σχέδιο Υλοποίησης

### Χρονοδιάγραμμα

Η μεθοδολογία που χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής είναι η RUP(Rational Unified Process) κατά την οποία το έργο χωρίζεται στις εξής τέσσερις φάσεις : έναρξη (Inception), επεξεργασία (Elaboration), κατασκευή (Construction) και μετάβαση (Transition). Επίσης, συνδυαστικά χρησιμοποιήθηκε και η μεθοδολογία scrum η οποία προβλέπει τις συχνές συναντήσεις μεταξύ των μελών της ομάδας κατά την οποία συζητούνταν η πρόοδος στην ανάπτυξη αλλά και γίνονταν συνεχείς βελτιώσεις στη δομή και στις απαιτήσεις του προγράμματος. Με δεδομένη τη μεθοδολογία οι φάσεις ανάπτυξης είναι οι εξής:

- **Ανάλυση Απαιτήσεων:**

Αρχικά πραγματοποιήθηκε ο καθορισμός των απαιτήσεων του έργου και των λειτουργικών και των μη λειτουργικών. Αποφασίστηκαν όλες οι ενέργειες που θα υποστηρίξει το πρόγραμμα αλλά και ορισμένες απαιτήσεις για τη ποιότητα και τις επιδόσεις της εφαρμογής.

- **Σχεδιασμός:**

Έπειτα, σχεδιάστηκε λεπτομερώς το κάθε κομμάτι του προγράμματος. Σχεδιάστηκαν διάγραμμα UML για την δομή των κλάσεων, διάγραμμα οντοτήτων – συσχετίσεων και σχεσιακό μοντέλο για τη δομή της βάσης δεδομένων και μέσω του figma καθορίστηκε η τελική μορφή των σελίδων διεπαφής χρήστη. Ακόμη επιλέχθηκαν οι τεχνολογίες και τα εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση του προγράμματος, όπως για παράδειγμα οι τεχνολογίες για τη διεπαφή και για τη βάση δεδομένων.

- **Ανάπτυξη:**

Σε αυτή τη φάση πραγματοποιήθηκε η συγγραφή του κώδικα με παράλληλη ανάπτυξη του frontend, του backend, του αλγορίθμου και των βάσεων δεδομένων. Πραγματοποιήθηκαν οι απαραίτητες αλλαγές κα έγινε οι σύνδεση όλου του προγράμματος.

- **Έλεγχος και δοκιμές:**

Τέλος, πραγματοποιήθηκαν πολλές χειροκίνητες δοκιμές αλλά και δημιουργήθηκαν τα απαραίτητα τεστ για να εξασφαλιστεί η αξιοπιστία και η σωστή λειτουργία του προγράμματος.

## Παραδοτέα

### Παραδοτέα Σημαντικών Σταδίων Ανάπτυξης:

#### 1. Ανάλυση Απαιτήσεων

- **Παραδοτέα:**
  - ο Λίστα Λειτουργικών και Μη Λειτουργικών Απαιτήσεων.
  - ο Σενάρια Χρήσης (Use Cases) και αντίστοιχα Διαγράμματα.
  - ο Αναφορά Καθορισμού Ρόλων και Χρηστών.

#### 2. Σχεδιασμός

- **Παραδοτέα:**
  - ο UML Διαγράμματα:
    - Διάγραμμα Κλάσεων (Class Diagram).
    - Διάγραμμα Ροής Εργασιών (Activity Diagram).
  - ο Διάγραμμα Οντοτήτων-Συσχετίσεων (ER Diagram) και Σχεσιακό Μοντέλο.
  - ο Wireframes/Mockups διεπαφής χρήστη μέσω εργαλείων όπως το Figma.
  - ο Αναφορά επιλογής τεχνολογιών και εργαλείων ανάπτυξης.

#### 3. Ανάπτυξη

- **Παραδοτέα:**
  - ο Ενδιάμεσος και Τελικός Κώδικας:
    - Backend (API, Διαχείριση Βάσεων Δεδομένων).
    - Frontend (User Interface και Επικοινωνία με Backend).
  - ο Σύνδεση και Ολοκλήρωση Υποσυστημάτων.
  - ο Τεκμηρίωση κώδικα και διαδικασιών.

#### 4. Έλεγχος και Δοκιμές

- **Παραδοτέα:**
  - ο Σενάρια Δοκιμών (Test Cases) και Λίστα Ελέγχου.
  - ο Αναφορές Αποτελεσμάτων Δοκιμών.
  - ο Αναφορά Διορθώσεων (Bug Fixes Report).
  - ο Εγχειρίδιο Χρήσης και Εγκατάστασης (User Manual & Installation Guide).

#### 5. Τελική Παράδοση

- **Παραδοτέα:**
  - ο Λειτουργική Εφαρμογή (Deployable Version).
  - ο Τελική Αναφορά Έργου.
  - ο Παρουσίαση Έργου (PowerPoint ή άλλο μέσο).
  - ο Οδηγίες Συντήρησης και Τεκμηρίωσης (Maintenance Documentation).

Αυτά τα στάδια αποτελούν τα πιο σημαντικά σημεία στην ανάπτυξη ενός έργου, συνδυάζοντας τεκμηρίωση, σχεδίαση, ανάπτυξη και έλεγχο για να εξασφαλιστεί η επιτυχία της εφαρμογής.

## Τεκμηρίωση

### Οδηγίες χρήσης για τελικούς χρήστες

#### Αρχική Σελίδα / Είσοδος:

Η σύνδεση πραγματοποιείται με τον λογαριασμό του χρήστη ή μέσω εγγραφής με τα απαραίτητα στοιχεία.

#### Εισαγωγή

#### Μαθημάτων:

Μετά τη σύνδεση, η μετάβαση γίνεται αυτόματα στην επιλογή Courses. Οι πληροφορίες, όπως το όνομα του μαθήματος, ο αριθμός των σελίδων και η ημερομηνία εξέτασης, συμπληρώνονται από τον χρήστη. Με την επιλογή OK, οι πληροφορίες αποθηκεύονται και η μετάβαση πραγματοποιείται αυτόματα στην ενότητα Assignments.

#### Εισαγωγή

#### Εργασιών:

Οι λεπτομέρειες, όπως ο τίτλος, ο χρόνος ολοκλήρωσης, η ημερομηνία παράδοσης και η δυσκολία, εισάγονται από τον χρήστη. Με την επιλογή OK, οι πληροφορίες αποθηκεύονται και η μετάβαση γίνεται αυτόματα στη σελίδα Διαθεσιμότητας.

#### Διαθεσιμότητα:

Η διαθεσιμότητα ορίζεται για κάθε ημέρα της εβδομάδας. Με την επιλογή OK, η μετάβαση πραγματοποιείται αυτόματα στο ημερολόγιο.

#### Οργάνωση

#### στο

#### Ημερολόγιο:

Το προτεινόμενο πλάνο διαβάσματος εμφανίζεται στην οθόνη. Αν απαιτείται νέο πλάνο, χρησιμοποιείται η επιλογή Refresh.

#### Προβολή

#### Στατιστικών:

Στην ενότητα Dashboard, εμφανίζονται γραφήματα και στατιστικά δεδομένα, όπως:

- Η κατανομή του χρόνου ανά δραστηριότητα.
- Η παραγωγικότητα κατά τη διάρκεια της εβδομάδας.
- Τα ποσοστά ολοκλήρωσης για εργασίες, μαθήματα και στόχους.

## Τεκμηρίωση λογισμικού

[Link από το repository για τα Javadoc](#)

## Συμπεράσματα και Μελλοντικές Εκδόσεις

Συμπερασματικά, η εφαρμογή StudyBuddyApp αποτελεί ένα χρήσιμο εργαλείο διαχείρισης προσωπικών μαθημάτων και εργασιών, το οποίο έχει σχεδιαστεί για να βοηθήσει κυρίως τους φοιτητές



να οργανώσουν τις σπουδές τους με αποτελεσματικό τρόπο. Με το StudyBuddyApp, οι χρήστες μπορούν να παρακολουθούν τα μαθήματα, τις εργασίες και τη διαθεσιμότητά τους μέσω ενός διαδραστικού ημερολογίου. Η εφαρμογή προσφέρει τη δυνατότητα εισαγωγής μαθημάτων και εργασιών, ενώ επιτρέπει επίσης την παραμετροποίηση του καθημερινού προγράμματος, παρέχοντας ταυτόχρονα εργαλεία παρακολούθησης της απόδοσης του χρήστη.

Αναφορικά με τις μελλοντικές εκδόσεις της εφαρμογής, υπάρχουν πολλές ιδέες για βελτίωση και εμπλουτισμό του συστήματος. Μερικές από τις προτάσεις περιλαμβάνουν: Ενσωμάτωση με εξωτερικά ημερολόγια: Η σύνδεση της εφαρμογής με το ημερολόγιο του χρήστη, είτε μέσω Google Calendar είτε άλλων πλατφόρμων, θα επέτρεπε μια πιο εύκολη και γρήγορη παρακολούθηση των καθημερινών υποχρεώσεων.

Ανατροφοδότηση και εξατομικευμένες συμβουλές: Χρησιμοποιώντας αλγορίθμους τεχνητής νοημοσύνης, η εφαρμογή θα μπορούσε να παρέχει εξατομικευμένες συμβουλές για τη βελτίωση της απόδοσης του χρήστη, προσφέροντας υποδείξεις για καλύτερη οργάνωση του χρόνου του.

Συγχρονισμός με συστήματα διαχείρισης σπουδών (LMS): Η δυνατότητα συγχρονισμού της εφαρμογής με τα συστήματα διαχείρισης σπουδών των εκπαιδευτικών ιδρυμάτων θα διευκόλυνε την παρακολούθηση της προόδου του χρήστη και την αυτόματη ενημέρωση των δεδομένων για τα μαθήματα και τις εργασίες του.

Χρήση τεχνητής νοημοσύνης (AI): Μελλοντικά, η εφαρμογή θα μπορούσε να ενσωματώσει τεχνολογίες τεχνητής νοημοσύνης για την αναγνώριση προτύπων στις συνήθειες του χρήστη, προσφέροντας έτσι πιο ακριβείς και χρήσιμες συστάσεις για τη βελτίωση της αποδοτικότητας και της οργάνωσης. Τέλος, Το StudyBuddyApp έχει τη δυνατότητα να εξελιχθεί σε μια ακόμα πιο δυναμική και ολοκληρωμένη λύση για την προσωπική και ακαδημαϊκή οργάνωση, εφόσον συνεχίσει να προσαρμόζεται στις ανάγκες των χρηστών και να ενσωματώνει σύγχρονες τεχνολογίες.

## Παραπομπές

Η τεχνητή νοημοσύνη αξιοποιήθηκε αποκλειστικά ως εργαλείο υποστήριξης για τη συγγραφή και τη βελτίωση του κώδικα, ενώ όλες οι τελικές αποφάσεις και η υλοποίηση ανήκουν στον συντάκτη της παρούσας αναφοράς. Τα αρχεία alert.css και success.css, καθώς και τα σχόλια JavaDoc, δημιουργήθηκαν με τη συνδρομή του ChatGPT (OpenAI, 2025). Επιπλέον, παραγόμενη τεχνητή νοημοσύνη αξιοποιήθηκε συμβουλευτικά για τη συγγραφή κώδικα, κυρίως με στόχο την εκμάθηση της βιβλιοθήκης JavaFX και τη διαχείριση των βάσεων δεδομένων.

## Βιβλιογραφία

<https://doi.org/10.1109/TKDE.2023.3251721>

<https://doi.org/10.1145/3534678.3539081>