



ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΠΣ

Μάθημα ροής

NNA-06-04 Ανάπτυξη Εφαρμογών Κινητών Συσκευών

Εργασία Εξαμήνου

Android App **StudyWithMe**

Δημιουργοί

3865 Μπαλακτσής Χρήστος

3875 Ανθόπουλος Ευστράτιος

3931 Μαρέδης Ιωάννης

ΜΑΪΟΣ 2023
ΘΕΣΣΑΛΟΝΙΚΗ

Περιγραφή εφαρμογής

Τι είναι και σε ποιους απευθύνεται

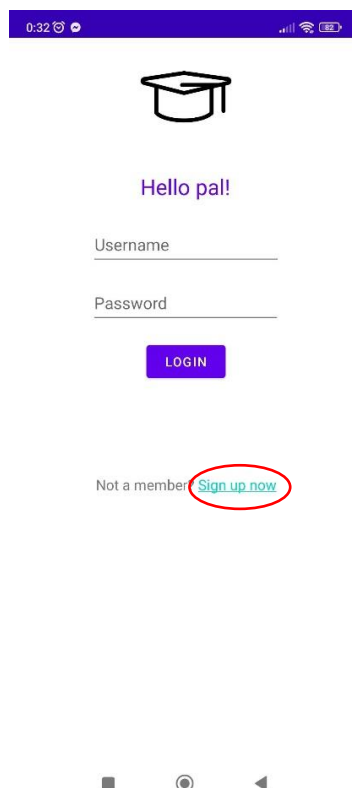
Σε κάθε εξάμηνο, οι φοιτητές διαφόρων πανεπιστημίων και σχολών καλούνται συνήθως να εξεταστούν σε περισσότερες από μία εξέταση, καθώς και να κάνουν κάποιες εργασίες, προκειμένου να περάσουν τα μαθήματά τους. Ωστόσο, κατά τη διάρκεια της μελέτης, σε πολλούς φοιτητές προκύπτει η ανάγκη να διαβάσουν κάποιο μάθημα με άλλα άτομα με στόχο την επίλυση αποριών, περισσότερη εξάσκηση και γενικότερα την καλύτερη κατανόηση του αντικειμένου που πραγματεύεται το εκάστοτε μάθημα. Προκειμένου να καλυφθεί η ανάγκη αυτή, δημιουργήθηκε το StudyWithMe. Η εφαρμογή StudyWithMe απευθύνεται σε φοιτητές που αναζητούν παρέα για διάβασμα για ένα συγκεκριμένο μάθημα, σε μία χρονική περίοδο που θέτει ο φοιτητής που αναζητά και για τους λόγους που χρειάζεται τη παρέα για το διάβασμα, λόγω χάρη για κάποια επερχόμενη πρόοδο. Στόχος της εφαρμογής είναι να εντοπίσει και να φέρει σε επαφή φοιτητές με ίδιο ή παρόμοιο αντικείμενο μελέτης.



Λειτουργίες / Χρήση της εφαρμογής

- Sign up (Εγγραφή)

Κάθε νέος χρήστης πρέπει να δημιουργήσει έναν λογαριασμό στην εφαρμογή για να την χρησιμοποιήσει. Για να το κάνει, θα πρέπει να πατήσει την επιλογή **Sign up now** που βρίσκεται στο κάτω μέρος της αρχικής οθόνης.



Στη συνέχεια, χρειάζεται να συμπληρωθούν τα πεδία με τα στοιχεία του φοιτητή ακολουθώντας τις υποδείξεις για τη σωστή συμπλήρωση των στοιχείων.

0:32

Sign up

First Name

Last Name

Username

Password

Email

Aristotle University of Thessaloniki

Department

CREATE

Εφόσον πραγματοποιηθεί η ορθή συμπλήρωση των πεδίων, είναι διαθέσιμη η επιλογή **Create**, με την οποία δημιουργείται ο νέος λογαριασμός.

0:34

Sign up

Γιώργος

Παπαδόπουλος

GeorgePap

.....

geopapado@gmail.com

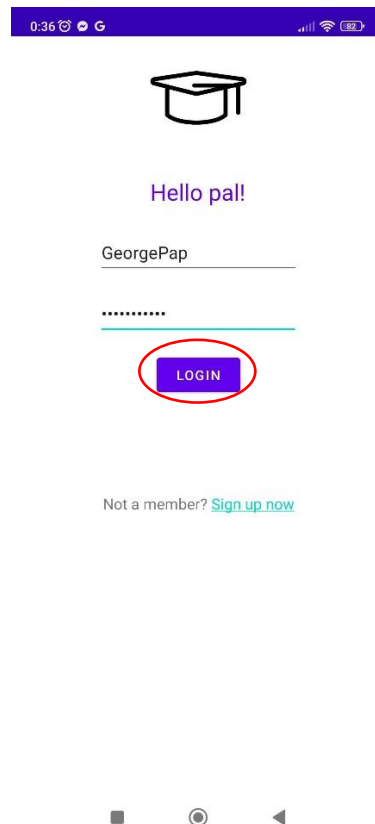
Aristotle University of Thessaloniki

Computer science


CREATE

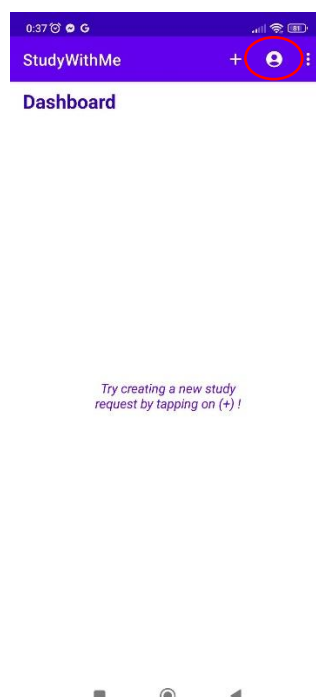
- Log in(Είσοδος)

Προκειμένου να συνδεθεί ο χρήστης στον λογαριασμό του, εισάγει το **Username** (ψευδώνυμο) του και το **Password** (κωδικό) του στην αρχική οθόνη. Στη συνέχεια, πατώντας την επιλογή **LOGIN**, εισέρχεται στον λογαριασμό του.

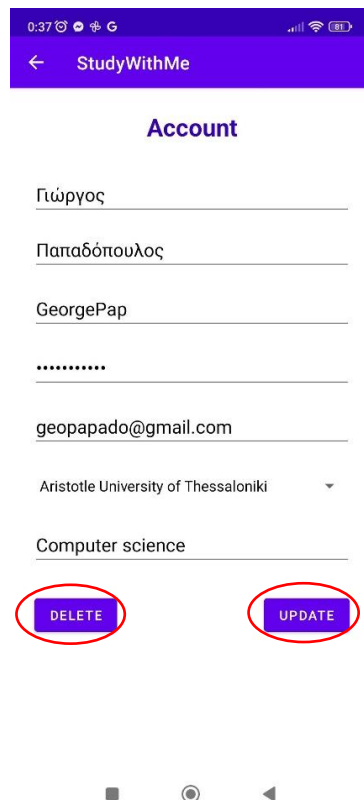


- Διαγραφή και επικαιροποίηση ενός λογαριασμού

Ένας χρήστης μπορεί να καταργήσει ή να επικαιροποιήσει τον λογαριασμό του επιλέγοντας το εικονίδιο  που βρίσκεται στο επάνω δεξιά τμήμα της οθόνης.



Μόλις ο χρήστης πατήσει το εικονίδιο, μεταφέρεται στα στοιχεία του λογαριασμού του. Κάτω από τα στοιχεία υπάρχουν οι επιλογές **DELETE** και **UPDATE**.

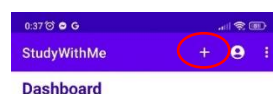


Αν ο χρήστης θέλει να ενημερώσει τα στοιχεία του, πραγματοποιεί οποιαδήποτε αλλαγή επιθυμεί και στη συνέχεια πατάει την επιλογή **UPDATE**.

Σε περίπτωση που επιθυμεί την κατάργηση του λογαριασμού του, επιλέγει το **DELETE**. Έπειτα από κάθε επιλογή, εμφανίζεται επιβεβαιωτικό μήνυμα στην οθόνη ότι η λειτουργία εκτελέστηκε επιτυχώς.

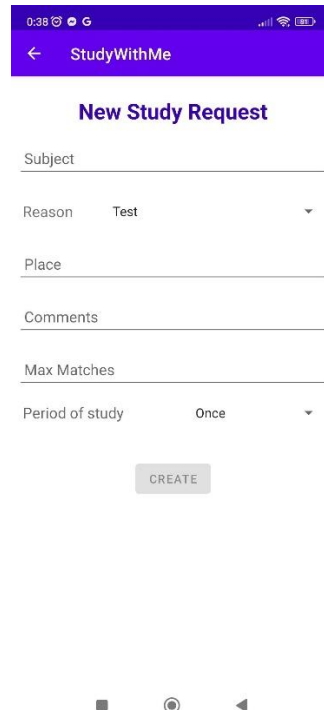
- Δημιουργία αιτήματος(request) για διάβασμα

Μόλις συνδεθεί ο χρήστης, μεταφέρεται στην αρχική σελίδα. Στο επάνω δεξιά μέρος της οθόνης υπάρχει η επιλογή του νέου αιτήματος, η οποία απεικονίζεται με το σύμβολο '+'.



Try creating a new study request by tapping on (+) !

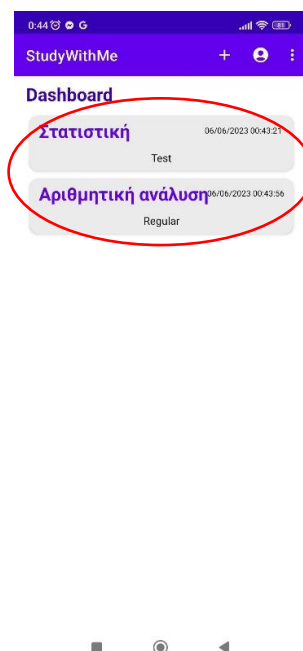
Πατώντας το '+' ο χρήστης μεταφέρεται στη σελίδα όπου απαιτείται η συμπλήρωση των πεδίων για να δημιουργηθεί το αίτημα.



Εφόσον πραγματοποιηθεί η ορθή συμπλήρωση των πεδίων, είναι διαθέσιμη η επιλογή **Create**, με την οποία δημιουργείται το νέο αίτημα.

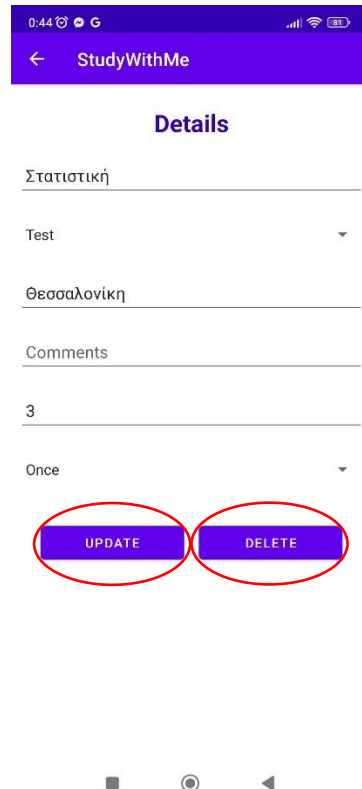
- Προβολή και διαχείριση των αιτημάτων(requests)

Μόλις συνδεθεί ο χρήστης, μεταφέρεται απευθείας στην αρχική σελίδα, στην οποία υπάρχει το Dashboard. Εκεί εμφανίζονται με γκρι χρώμα όλα τα αιτήματα που έχει κάνει, καθώς και η ημερομηνία και η ώρα δημιουργίας του αιτήματος.



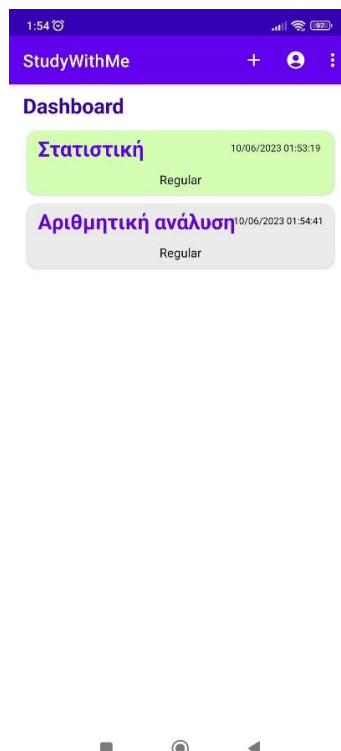
Ο χρήστης μπορεί να καταργήσει ή να επικαιροποιήσει το αίτημά του πατώντας πάνω στο αίτημα που θέλει να επεξεργαστεί. Αν ο χρήστης θέλει να προβεί σε αλλαγές του αιτήματος του, πραγματοποιεί

οποιαδήποτε αλλαγή επιθυμεί και στη συνέχεια πατάει την επιλογή **UPDATE**. Σε περίπτωση που επιθυμεί την κατάργηση του request του, επιλέγει την επιλογή **DELETE** και το αίτημά του διαγράφεται.

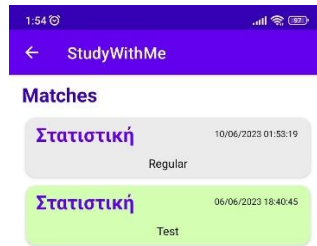


- Προβολή και διαχείριση των matches

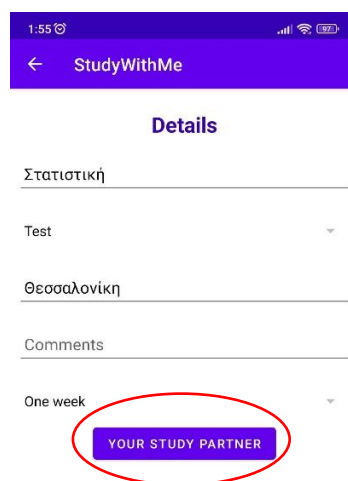
Όταν γίνει κάποιο match σε ένα από τα αιτήματά του χρήστη, τότε αυτό εμφανίζεται στο Dashboard με πράσινο χρώμα.



Επιλέγοντας αυτό το request, ο χρήστης κατευθύνεται στη σελίδα όπου βλέπει το δικό του αρχικό αίτημα που απεικονίζεται με γκρι χρώμα και με πράσινο το αίτημα με το οποίο ταίριαξε.

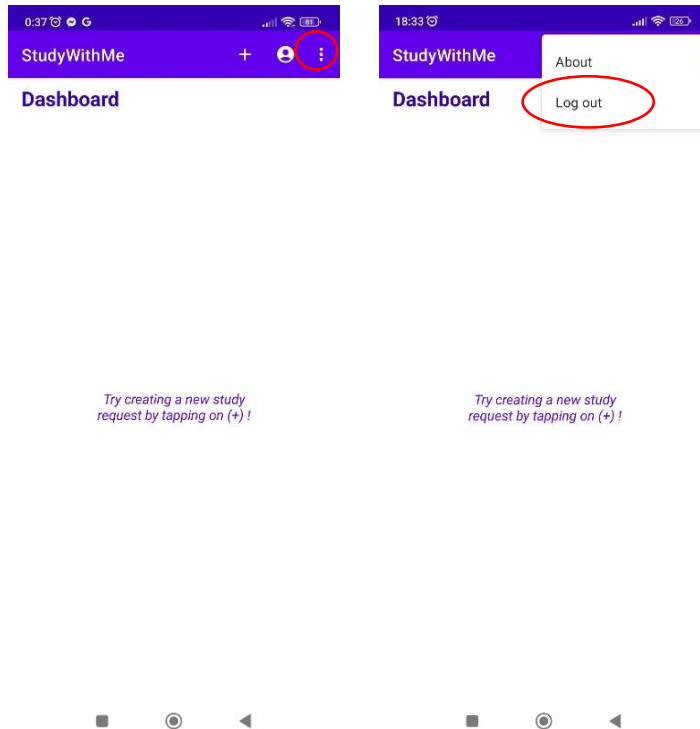


Μόλις ο χρήστης πατήσει το αίτημα με το οποίο έκανε ματς, μπορεί να δει τον νέο συνεργάτη του μέσω της επιλογής **YOUR STUDY PARTNER**.




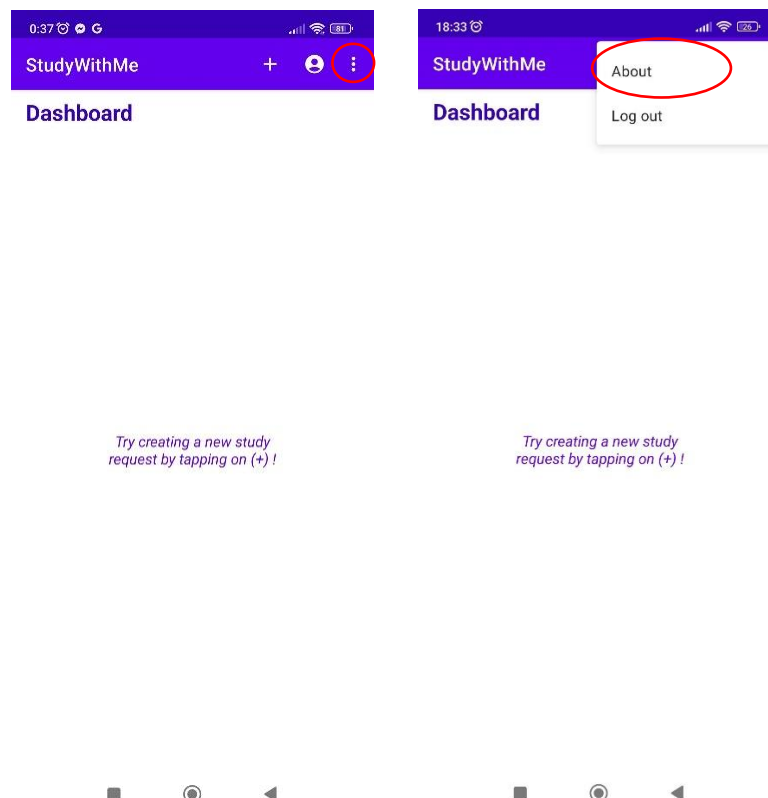
- Αποσύνδεση λογαριασμού

Ο χρήστης μπορεί να αποσυνδεθεί από την εφαρμογή επιλέγοντας τις τρεις τελείες που βρίσκονται στην πάνω δεξιά γωνία της οθόνης του κινητού ⋮ και στην συνέχεια, την επιλογή **Log out**.



- Πληροφορίες σχετικά με την εφαρμογή

Ο χρήστης μπορεί να μάθει περισσότερες πληροφορίες σχετικά με την εφαρμογή επιλέγοντας το εικονίδιο  που βρίσκεται στο επάνω δεξιά μέρος της αρχικής οθόνης και στη συνέχεια, την επιλογή **About**.



Η δομή της εφαρμογής, τη δεδομένη στιγμή, προϋποθέτει ότι όλοι χρησιμοποιούν την ίδια τοπική εφαρμογή, δηλαδή οι χρήστες και το σύνολο των δεδομένων αποθηκεύονται σε τοπική βάση (εξάρτηση από τη συσκευή). Αυτό ασφαλώς δεν είναι πρακτικά λειτουργικό, αποτελεί ωστόσο ένα proof of concept για την λειτουργία του λογισμικού. Για την πραγματική χρήση της εφαρμογής, αρκεί να αντικατασταθεί η τοπική SQLite relational database με ένα Web API και μια κεντρική βάση δεδομένων, πιθανόν μη σχεσιακή (NoSQL), τύπου Room ή MongoDB σε έναν απομακρυσμένο server. Στην τρέχουσα έκδοση της εφαρμογής υποστηρίζονται όλα τα CRUD operations τόσο για τον λογαριασμό χρήστη όσο και για τα study requests.

Επιπλέον, παρέχεται στον χρήστη η δυνατότητα για ελεύθερη πλοήγηση μέσα στην εφαρμογή χωρίς να υπάρχει η ανάγκη για χρήση των κουμπιών της συσκευής καθώς η εφαρμογή διαθέτει δικό της back button. Παράλληλα, η εφαρμογή μπορεί να λειτουργήσει και σε landscape mode (ή εναλλαγή των modes), αλλά συνίσταται έντονα η χρήση του portrait orientation για τυπικές κινητές συσκευές.

Η ανάπτυξη του κώδικα της εφαρμογής έγινε σε γλώσσα Java, αξιοποιώντας το Android API. Για την υλοποίησή του χρησιμοποιήθηκε το εργαλείο Android Studio. Η δομή του κώδικα αποτελείται από τις 17 κλάσεις (.java αρχεία): User, StudyRequest, StorageHandler, Account, SignUpActivity, PeriodOfStudy, LoginActivity, AccountActivity, Dashboard, RecyclerViewAdapter, AboutActivity, CreateStudyRequestActivity, MatchesListActivity, ReasonOfStudy, University, UpdateStudyRequestActivity, ViewStudyRequestActivity. Στη γενική περίπτωση, οι κλάσεις τύπου activity συνδέουν αναφορικά μεταβλητές τους με στοιχεία του αντίστοιχου layout (GUI) και προσαρμόζουν τις ιδιότητες και τη λειτουργικότητά τους ανάλογα.

- User – ορίζεται η έννοια του χρήστη της εφαρμογής

Στην κλάση User, κατασκευάζονται και αρχικοποιούνται τα πεδία της κλάσης που αφορούν τον χρήστη, τα οποία είναι το username (αναγνωριστικό) του χρήστη, ο κωδικός που θα χρησιμοποιεί, το email του, το όνομα και το επίθετό του, το πανεπιστήμιο και η σχολή στην οποία φοιτά και τα requests που δέχεται. Επιπλέον, καταχωρίζεται το id του χρήστη από την βάση δεδομένων. Τα πεδία αυτά είναι private, επομένως στην κλάση υπάρχουν setters και getters για κάθε πεδίο. Σημειώνεται ότι το id του User καθορίζεται μόνο κατά την ανάληψή του από τη βάση δεδομένων και όχι γενικά κατά τη δημιουργία ενός object User. Συνεπώς, η δημιουργία χρήστη της εφαρμογής απαιτεί την κατασκευή αντικειμένου της κλάσης, την καταχώρησή του στη βάση δεδομένων και την ανάγνωσή του (προσκόμιση) με εναποτεθειμένο id.

- StudyRequest – ορίζεται η έννοια του αιτήματος εύρεσης «συνεργάτη» για διάβασμα/εργασία

Τα πεδία που σχετίζονται με το αίτημα για ομαδικό διάβασμα κατασκευάζονται και αρχικοποιούνται στην κλάση StudyRequest. Τα πεδία αυτά είναι το μάθημα για το οποίο θέλει ο χρήστης να διαβάσει με παρέα, ο σκοπός που μελετάει αυτό το μάθημα, η τοποθεσία του, κάποια επιπλέον σχόλια του χρήστη, η ημερομηνία που κάνει το αίτημα, η χρονική περίοδος κατά την οποία θέλει να γίνει η ομαδική μελέτη, ο μέγιστος αριθμός ατόμων/αιτημάτων που θέλει και ο αριθμός των χρηστών με του οποίους έκανε match. Επιπλέον, καταχωρίζεται το id του χρήστη από την βάση δεδομένων. Τα πεδία αυτά είναι private, επομένως στην κλάση υπάρχουν setters και getters για κάθε πεδίο. Σημειώνεται ότι το id καθορίζεται μόνο κατά την ανάληψη του StudyRequest από τη βάση δεδομένων και όχι γενικά κατά τη δημιουργία ενός τέτοιου object. Συνεπώς, η δημιουργία StudyRequest απαιτεί την κατασκευή αντικειμένου της κλάσης, την καταχώρησή του στη βάση δεδομένων και την ανάγνωσή του (προσκόμιση) με εναποτεθειμένο id. Στην κλάση, επίσης, υπάρχει η συνάρτηση **matchRequest()** η οποία, με βάση τα παραπάνω πεδία, υπολογίζει το ποσοστό που δείχνει πόσο ταιριάζουν δύο αιτήματα. Για το κάθε πεδίο υπάρχουν συντελεστές ανάλογα με τη βαρύτητα που έχουν στο αίτημα. Η βοηθητική συνάρτηση **datesDiff()** δέχεται δύο ημερομηνίες, την ημερομηνία δημιουργίας τους πρώτου αιτήματος και την ημερομηνία δημιουργίας του δεύτερου, και υπολογίζει και επιστρέφει το χρονικό διάστημα που μεσολάβησε μεταξύ τους. Με τη βοηθητική συνάρτηση

CalculateSimilarity() προσδιορίζεται το ποσοστό ομοιότητας των μαθημάτων και της τοποθεσίας. Η συνάρτηση δέχεται δύο συμβολοσειρές. Αν κάποια από τις δύο είναι κενή συμβολοσειρά η συνάρτηση επιστρέφει 0. Σε αντίθετη περίπτωση, υπολογίζει και επιστρέφει το ποσοστό ομοιότητας των δύο συμβολοσειρών.

- ReasonOfStudy

Η κλάση (enum) ReasonOfStudy περιλαμβάνει τους διαφορετικούς λόγους για τους οποίους δύναται να γίνει ένα study request από έναν χρήστη. Πχ. διαγώνισμα, εξέταση, τακτικό διάβασμα κλπ.

- University

Η κλάση (enum) University αναπαριστά τις επιλογές των πανεπιστημίων που δηλώνει ένας χρήστης κατά τη δημιουργία ενός study request. Πχ. ΑΠΘ, ΠΑΜΑΚ, ΔΙΠΑΕ κλπ.

- StorageHandler

Η κλάση StorageHandler κληρονομεί την κλάση SQLiteOpenHelper. Στην κλάση αυτή δημιουργούνται οι πίνακες των αιτημάτων (request), των χρηστών (users) και των ταιριασμένων requests (matches) και ο κάθε πίνακας έχει συγκεκριμένα χαρακτηριστικά. Στους πίνακες καταγράφονται τα δεδομένα κάθε χρήστη, τα χαρακτηριστικά των αιτημάτων και τα ζεύγη αιτημάτων που έχουν ταιριάξει μεταξύ τους, αντίστοιχα. Επιπλέον, όταν εγγράφεται νέος χρήστης, γίνεται κάποιο νέο αίτημα ή νέο ταιρίασμα, τότε ενημερώνονται οι αντίστοιχοι σχεσιακοί πίνακες. Η αντίθετη διαδικασία εκτελείται όταν πρόκειται να διαγραφεί ένας χρήστης. Σημειώνεται ότι η διαγραφή χρήστη επιφέρει διαγραφή των αιτημάτων του και αφαίρεση των matching-ζευγών που περιλαμβάνουν αιτήματα του χρήστη αυτού. Σε περίπτωση που πραγματοποιηθεί οποιαδήποτε αλλαγή στον λογαριασμό ενός χρήστη ή σε ένα request, γίνεται ενημέρωση των δεδομένων στους πίνακες. Οι συναρτήσεις **fetchUserbyId**, **fetchStudyRequestbyUser** και **fetchMatchesOfStudyRequest** αποσκοπούν στην ανάκτηση του λογαριασμού του χρήστη, των αιτημάτων του χρήστη και των αιτημάτων που έχουν ταιριάξει με ένα συγκεκριμένο αίτημα. Αξίζει να σημειωθεί ότι οι περισσότερες ανακτήσεις δεδομένων γίνονται με βάση μια id ιδιότητα (χρήστη ή αιτήματος), εξασφαλίζοντας έτσι ότι υπάρχει άμεση αναφορά σε υπάρχον στοιχείο της βάσης δεδομένων.

- Account

Η abstract κλάση (activity) Account περιέχει αντικείμενα και πεδία που σχετίζονται με τον λογαριασμό του χρήστη και εκτελούνται όλες οι ενέργειες για την δημιουργία και την επεξεργασία του. Ελέγχονται τα στοιχεία που εισάγει ο χρήστης στην φόρμα εγγραφής και κατά τη διάρκεια της εισαγωγής εμφανίζονται ενημερωτικά μηνύματα στον χρήστη σχετικά με την συμπλήρωση των στοιχείων του, του username και του κωδικού του λογαριασμού. Επιπλέον, στην κλάση υλοποιούνται οι μέθοδοι τύπου **onClick** για τα κουμπιά διαγραφής/ενημέρωσης-αποθήκευσης. Η εν λόγω κλάση υλοποιήθηκε με σκοπό την αποφυγή διπλότυπου κώδικα, μιας και οι φόρμες (activities) δημιουργίας, ενημέρωσης και προβολής χρήστη περιλαμβάνουν πολλά κοινά στοιχεία. Συνεπώς, χάρη στην κληρονομικότητα μπορεί η κάθε δραστηριότητα να υλοποιήσει τροποποιήσεις στην αφηρημένη.

- SignUpActivity

Η κλάση (activity) SignUpActivity κληρονομεί την κλάση Account. Στην κλάση SignUpActivity υλοποιείται η φόρμα εγγραφής του χρήστη στην εφαρμογή. Με τη συμπλήρωση των κατάλληλων πεδίων, επιτρέπεται η επιλογή του κουμπιού STORE, που καλεί τη συνάρτηση **storeAccount()**. Η συνάρτηση συλλέγει τις εισόδους που δόθηκαν στα πεδία της εγγραφής, δημιουργεί και προσθέτει τον νέο χρήστη στη βάση δεδομένων και εμφανίζει κατάλληλο μήνυμα επιτυχούς ή ανεπιτυχούς δημιουργίας λογαριασμού στον χρήστη.

- PeriodOfStudy

Η κλάση (enum) `PeriodOfStudy` ορίζει τις περιόδους που μπορεί να αναφέρεται ένα αίτημα για ομαδική μελέτη. Πχ. μια φορά, μια φορά τη βδομάδα, όλο το εξάμηνο κλπ.

- LoginActivity

Στην κλάση (activity) `LoginActivity` υλοποιούνται τα πεδία σύνδεσης του χρήστη στην εφαρμογή και παρέχεται η δυνατότητα επιλογής εγγραφής νέου χρήστη. Τοποθετούνται πεδία εισαγωγής `username` και συνθηματικού και η επιλογή εγγραφής. Η συνάρτηση `LoginChecked()` παίρνει το `username` και τον κωδικό που εισάγει ο χρήστης και τα ελέγχει. Αν είναι ταυτίζονται με κάποιον χρήστη που έχει δημιουργηθεί και είναι αποθηκευμένος στη βάση δεδομένων, τότε μεταφέρει τον χρήστη στο `Dashboard` του (βλ. `DashboardActivity`), εμφανίζοντας μήνυμα για επιτυχή, διαφορετικά εμφανίζει στον χρήστη μήνυμα σφάλματος.

- AccountActivity

Η κλάση (activity) `AccountActivity` κληρονομεί την κλάση `Account`. Στην κλάση γίνεται διαχείριση και έλεγχος των πληροφοριών και της δραστηριότητας ενός λογαριασμού χρήστη. Συγκεκριμένα, προκειμένου ο χρήστης να μπορεί να δει και να επεξεργαστεί πληροφορίες του λογαριασμού, περνάει από έλεγχο ταυτοποίησης. Αν δεν υπάρχει ταύτιση των στοιχείων, τότε απαγορεύεται η πρόσβαση στον χρήστη. Αντίθετα, αν υπάρξει ταύτιση, τότε μπορεί να επεξεργαστεί τον λογαριασμό του και να δει τα στοιχεία του συνεργάτη του. Με την συνάρτηση `storeAccount()` γίνεται ενημέρωση των αλλαγών στο λογαριασμό.

- DashboardActivity

Η κλάση (activity) `DashboardActivity` αποτελεί το `dashboard` του χρήστη. Είναι το κεντρικό σημείο της εφαρμογής, στο οποίο μεταβαίνει αμέσως μετά την σύνδεση ή βρίσκεται αμέσως πριν την αποσύνδεση. Η εν λόγω activity παρέχει στον χρήστη πρόσβαση στις βασικές λειτουργίες της εφαρμογής. Αναλυτικότερα περιλαμβάνει ένα `menu` επιλογών που επιτρέπει τη δημιουργία νέου αιτήματος, τη διαχείριση του λογαριασμού, την προβολή πληροφοριών σχετικά με την εφαρμογή και τη δυνατότητα αποσύνδεσης (`log out`). Τα παραπάνω υλοποιούνται και δρομολογούνται από τη συνάρτηση `onOptionsItemSelected()` της κλάσης. Επιπλέον, η `DashboardActivity` περιέχει ένα `RecyclerView` από στοιχεία τύπου `card` τα οποία εμφανίζονται με τη μορφή λίστας και αποτελούν τα `StudyRequests` που έχει δημιουργήσει ο χρήστης. Διακρίνονται τα `matched`, δηλαδή τα εκπληρωμένα, αιτήματα από τα υπόλοιπα, καθώς φέρουν πράσινο χρώμα στο φόντο των καρτών τους. Η υλοποίηση της `onClick` μεθόδου για τα `requests` υπογράφεται στο `interface ISStudyRequestRecycler` της κλάσης `RecyclerViewAdapter` και υλοποιείται από την παρούσα κλάση, διακρίνοντας την περίπτωση όπου ένα αίτημα είναι εκπληρωμένο ή όχι. Η μέθοδος `showStudyRequestDetails(id)` δέχεται ως όρισμα το `id` του αιτήματος που πατήθηκε και με βάση το `matching state` του καθορίζει την οθόνη που θα μεταβεί ο χρήστης, δηλαδή την οθόνη επεξεργασίας του (`UpdateStudyRequestActivity`) ή προβολής των `matches` του (`MatchesListActivity`).

- RecyclerViewAdapter

Η κλάση `RecyclerViewAdapter` κληρονομεί την `RecyclerView.Adapter`. Η κλάση χρησιμοποιείται για να εμφανίζει στον χρήστη τα αιτήματα που έχει κάνει ο ίδιος ή τα `matches` ενός αιτήματος. Η διάκριση αυτή καθορίζεται κατά τη δημιουργία του `object`, δηλαδή στον `constructor` και επί της ουσίας εξαρτάται από το αν τροφοδοτεί `RecyclerView` του `DashboardActivity` ή του `MatchesListActivity`, αντίστοιχα. Τροφοδοτεί τα `RecyclerView layouts` της εφαρμογής λαμβάνοντας τις λίστες με τα `requests` από τους αντίστοιχους πίνακες της βάσης δεδομένων. Αξίζει να σημειωθεί ότι, πέραν των τυπικών μεθόδων και υλοποιήσεων του `Adapter`, περιλαμβάνει ως ιδιότητα το `interface ISStudyRequestRecycler` το οποίο επιτρέπει την υλοποίηση διαφορετικών μεθόδων `onClick` με βάση το `activity` στο οποίο χρησιμοποιείται ο `RecyclerViewAdapter`.

- AboutActivity

Η κλάση (activity) `AboutActivity` αφορά την προβολή οθόνης πληροφοριών σχετικά με την εφαρμογή.

- CreateStudyRequestActivity

Η κλάση (activity) CreateStudyRequestActivity σχετίζεται με την οθόνη δημιουργίας ενός νέου αιτήματος, επιτρέποντας στον χρήστη να συμπληρώσει τα πεδία του αιτήματος. Αρχικά, ανακτά τα στοιχεία του συνδεδεμένου χρήστη από το DashboardActivity. Στη συνέχεια, δέχεται τη συμβολοσειρά που εισάγεται στο πεδίο του μαθήματος και επιλέγεται ο λόγος δημιουργίας του αιτήματος στο μενού επιλογής. Δέχεται από τον χρήστη τις εισόδους στα πεδία της τοποθεσίας, των σχολίων και του μέγιστου αριθμού για matches. Αν το πεδίο του αριθμού των matches δεν συμπληρωθεί τότε τοποθετεί by-default το 1. Έπειτα, επιλέγεται και η χρονική περίοδος από το μενού επιλογής στις οποίες αναφέρεται το αίτημα. Με τη συνάρτηση **createRequest()** παίρνει τα στοιχεία που εισήχθησαν στα πεδία, δημιουργεί ένα νέο study request με τις πληροφορίες που δόθηκαν και το αποθηκεύει στη βάση δεδομένων. Η συνάρτηση καλείται, όταν ο χρήστης πατήσει το κουμπί CREATE. Η συνάρτηση **TryActivateCreateButton()** ενεργοποιεί το κουμπί στον χρήστη μόνο αφού επιβεβαιώσει ότι όλα τα προαπαιτούμενα πεδία έχουν συμπληρωθεί (κατάλληλα).

- MatchesListActivity

Η κλάση (activity) MatchesListActivity εμφανίζει για το εκάστοτε επιλεγμένο (από το DashboardActivity) matched study request του συνδεδεμένου χρήστη λίστα αιτημάτων με τα οποία έκανε match, με τη μορφή card σε RecyclerView layout. Η υλοποίηση της onClick μεθόδου για τα requests υπογράφεται στο interface ISStudyRequestRecycler της κλάσης RecyclerViewAdapter και υλοποιείται από την παρούσα κλάση, με τη μέθοδο **showStudyRequestDetails(id)**, που δέχεται ως όρισμα το id του αιτήματος (card) που πατήθηκε και μεταβιβάζει τον χρήστη στην οθόνη ViewStudyRequestActivity, στην οποία οι διαθέσιμες ενέργειες καθορίζονται από τις παραμέτρους χρήστη και αιτήματος που έχουν διαδοθεί (βλ. παρακάτω).

- UpdateStudyRequestActivity

Η κλάση (activity) UpdateStudyRequestActivity σχετίζεται με την διαδικασία της επικαιροποίησης ή διαγραφής ενός αιτήματος. Αναλυτικότερα, προσυμπληρώνει τα πεδία που προσδιορίζουν τις ιδιότητες/πληροφορίες του επιλεγμένου αιτήματος που διαδόθηκε ως παράμετρος για την εκκίνηση του activity, και επιτρέπει την επεξεργασία τους. Με τη μέθοδο **updateStudyRequest()** λαμβάνει τις τιμές που βρίσκονται εκείνη τη στιγμή στα πεδία και αναλαμβάνει -εφόσον είναι επιτρεπτές τιμές- να ενημερώσει τα στοιχεία του αιτήματος στη βάση δεδομένων. Η μέθοδος **deleteStudyRequest()** διαγράφει το αίτημα από την βάση δεδομένων και επιστρέφει τον χρήστη στο Dashboard. Η μέθοδος **TryActivateUpdateButton()** ενεργοποιεί το κουμπί στον χρήστη μόνο αφού επιβεβαιώσει ότι όλα τα προαπαιτούμενα πεδία έχουν συμπληρωθεί κατάλληλα.

- ViewStudyRequestActivity

Η κλάση (activity) ViewStudyRequestActivity είναι υπεύθυνη για την προβολή των πληροφοριών ενός study request που ταίριαξε με κάποιο άλλο αίτημα. Εμφανίζει συμπληρωμένα τα πεδία που προσδιορίζουν το αίτημα (χωρίς δυνατότητα επεξεργασίας τόσο στην περίπτωση που το προβαλλόμενο αίτημα ανήκει στον συνδεδεμένο χρήστη όσο και στην περίπτωση των matches). Εφόσον το αίτημα δεν ανήκει στον τρέχοντα χρήστη, εμφανίζει στην οθόνη το κουμπί «Your study request» το οποίο με την μέθοδο **viewSROwner()** εκκινεί το activity προβολής στοιχείων χρήστη (AccountActivity) για την προβολή των στοιχείων του «ιδιοκτήτη» του ταιριασμένου αιτήματος. Διαφορετικά, ο χρήστης απλά βλέπει τα στοιχεία του αιτήματος που είχε δημιουργήσει και πλέον είναι matched και μη επεξεργάσιμο.