

Εργασία στο μάθημα Ανάπτυξη Εφαρμογών Κινητών Συσκευών 2023-2024

Όνομα Εφαρμογής:
Smokeless Journey



Συμμετέχοντες Ομάδας:

Γεώργιος Δάλλας

AEM: 4116

Κωσταντίνος Τσίγκας

AEM: 3923

Κωσταντίνος Δεμερδεσλής

AEM: 3548

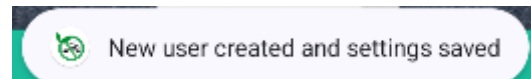
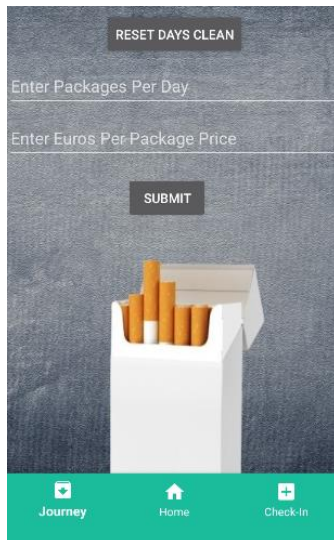
Σκοπός εφαρμογής:

Η εφαρμογή έχει σχεδιαστεί για να υποστηρίζει άτομα στην προσπάθειά τους να διακόψουν το κάπνισμα, παρέχοντας μια σειρά από χρήσιμα δεδομένα. Συγκεκριμένα, καταγράφει την ένταση των στερητικών συμπτωμάτων καθημερινά, εμφανίζει τον αριθμό των ημερών αποχής από το κάπνισμα, παρέχει ένα διάγραμμα με την πορεία των στερητικών συμπτωμάτων ανά ημέρα και υπολογίζει τα χρήματα που έχουν εξοικονομηθεί από την ημέρα διακοπής του καπνίσματος. Επιπλέον, περιλαμβάνει ένα ημερολόγιο που επιτρέπει την προβολή των καθημερινών μηνυμάτων που καταγράφονταν κατά το Check-in μαζί με την ένταση των στερητικών εκείνη την ημέρα για μια πιο ακριβείς προβολή της προόδου του χρήστη.

Οι τυπικοί χρήστες της εφαρμογής αποτελούνται από άτομα που επιδιώκουν να σταματήσουν το κάπνισμα για λόγους υγείας, οικονομίας ή προσωπικής βελτίωσης. Οι χρήστες αυτοί μπορεί να βρίσκονται σε διάφορα στάδια της διαδικασίας διακοπής, από εκείνους που μόλις ξεκινούν την προσπάθειά τους μέχρι εκείνους που έχουν ήδη επιτύχει σημαντική αποχή αλλά επιθυμούν να παρακολουθούν και να διατηρούν την πρόοδό τους.

Λειτουργικότητα εφαρμογής:

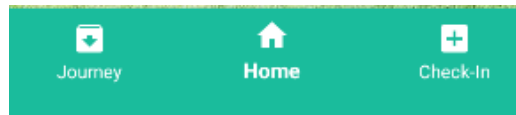
Η εφαρμογή έπειτα από την εγκατάσταση της, εμφανίζει σαν πρώτη καρτέλα την Settings, και θα την εμφανίζει για πρώτη καρτέλα μέχρι ο χρήστης να προσθέσει τις κατάλληλες ρυθμίσεις για την δημιουργία User και την προσθήκη του στην βάση δεδομένων, όπου εμφανίζεται και κατάλληλο ειδοποιητικό μήνυμα για την προσθήκη αυτή.



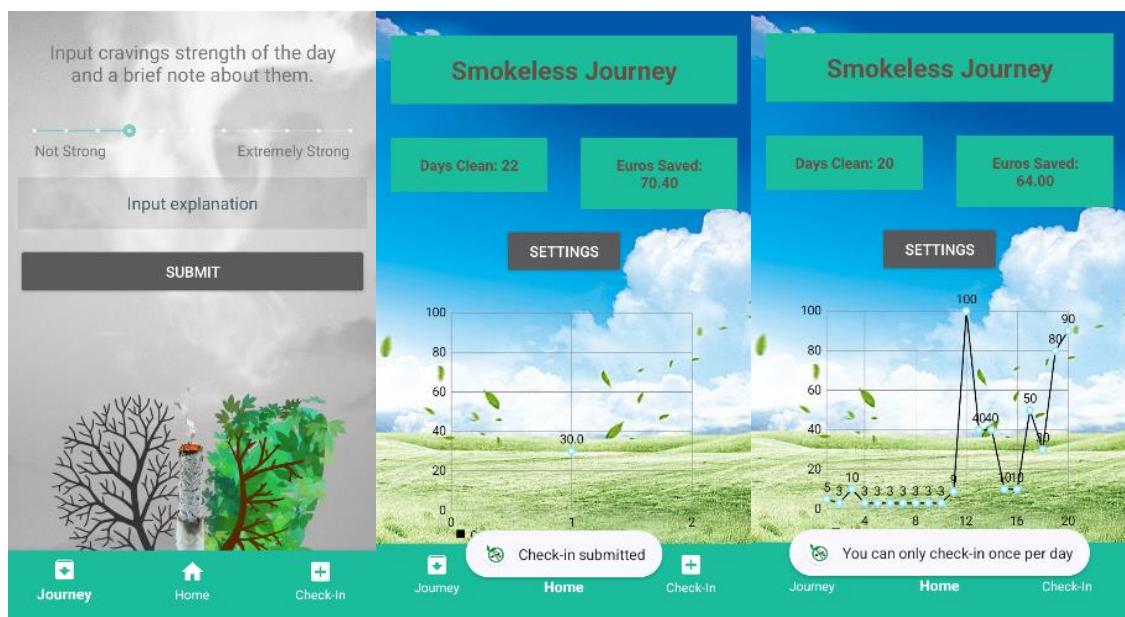
Έπειτα, έχοντας τις παραμέτρους eurosPerPackage και packagesPerDay, η πρώτη καρτέλα που εμφανίζεται κατά το άνοιγμα της εφαρμογής είναι το Home (MainActivity), όπου φαίνονται ο αριθμός ημερών που ο χρήστης δεν καπνίζει, τα λεφτά που εξοικονόμησε, ένα κουμπί πλοήγησης προς το Settings σε περίπτωση που ο χρήστης θέλει να επεξεργαστεί τις προαναφερθέντες παραμέτρους, καθώς και ένα διάγραμμα προόδου στερητικών του ανά ημέρα.



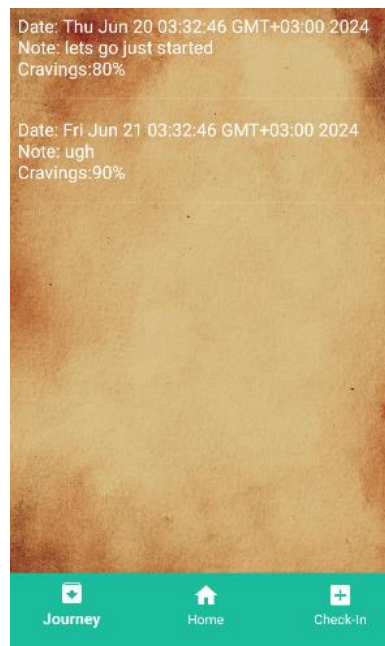
Στο μενού πλοήγησης στο κάτω μέρος της οθόνης, το οποίο είναι προς πρόσβαση σε οποιοδήποτε Activity, βρίσκονται εκτός από το Home (MainActivity), το Check In και το Journey.



Στο Check in, υπάρχει μια μπάρα με για είσοδο της έντασης επιθυμίας του καπνίσματος. Οι ενδείξεις που αναγράφονται στην μπάρα κυμαίνονται από "Not Strong" έως "Extremely Strong" και μετατρέπονται αυτόματα για την είσοδο στην βάση δεδομένων σε ποσοστό (%). Επιπλέον, υπάρχει επιλογή για προσθήκη κειμένου στο Check-In ώστε να αναδिकνύεται η πρόοδος με περισσότερες πληροφορίες κατά την πλοήγηση στο Journey. Τέλος, κατά την υποβολή του Check In υπάρχει ενδεικτικό ειδοποιητικό μήνυμα ότι η υποβολή ολοκληρώθηκε και μεταφορά πίσω στο MainActivity. Αλλιώς, κατάλληλο ενημερωτικό μήνυμα ότι δεν πέρασαν 24 ώρες από το τελευταίο Check-In, καθώς επιτρέπεται μία φορά την ημέρα.



Στο Journey, παρουσιάζεται μια scrollable λίστα από όλες τις ημέρες που έχει γίνει Check-In. Ποιο συγκεκριμένα, για κάθε μέρα, εμφανίζεται η ημερομηνία και η ώρα του, ένα σημείωμα (π.χ. στερητικά συμπτώματα), καθώς και το ποσοστό (%) επιθυμίας για κάπνισμα.



Τέλος, στο Settings, πλοηγήσιμο από το αρχικό activity (Home/MainActivity), κατά το άνοιγμα του, γίνεται φόρτωση των τελευταίων τιμών που εισχώρησε ο χρήστης στο packages Per Day και Euros Per Package ώστε να είναι ευκολότερη η τροποποίηση. Επιπλέον το κουμπί RESET DAYS CLEAN, μηδενίζει μόνο το counter στον User και δεν διαγράφει όλες τις προηγούμενες μέρες που έχουν γίνει Check-In, ώστε σε περίπτωση που ο χρήστης καπνίσει, και θέλει να μηδενίσει το days clean μετρητή του στην αρχική σελίδα, να μην χάνεται η προβολή όλης της προόδου που έχει κάνει ως τώρα.



Περιγραφή του κώδικα της εφαρμογής:

User.java:

Η κλάση User, χρησιμοποιείται για την αναπαράσταση των χρηστών στην εφαρμογή και αποτελείται από τις μεταβλητές : double eurosForTobaccoPackage, int packagesPerDay ,int daysClean , έναν βασικό constructor με τις προαναφερόμενες μεταβλητές καθώς και από setters και getters για κάθε μια από αυτές.

Day.java:

Η κλάση Day, χρησιμοποιείται για την αναπαράσταση ημέρας που έχει γίνει Check-In και έπειτα για την προβολή αυτής στο Journey. Αποτελείται από τις μεταβλητές Date checkInDate, String note, int cravingsPercentage, έναν βασικό constructor με όλες τις προαναφερόμενες μεταβλητές, καθώς και από setters και getters για κάθε μία από αυτές.

MyDBHandler.java:

Η κλάση αυτή αποτελεί την υλοποίηση της βάσης δεδομένων της εφαρμογής. Παρέχει μεθόδους για την δημιουργία της βάσης, για την προσθήκη, ενημέρωση καθώς και ανάκτηση δεδομένων. Πιο συγκεκριμένα:

- Η μέθοδος onCreate(SQLiteDatabase db), είναι υπεύθυνη για την δημιουργία των πινάκων κατά την πλάση της βάσης δεδομένων. Οι πίνακες που δημιουργούνται είναι ο User και ο Day και αποτελούνται από ένα primary key ο καθένας, καθώς και από όλα τα στοιχεία των αντίστοιχων κλάσεων.
- Η μέθοδος onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion), χρησιμοποιείται όταν αλλάζει η έκδοση της βάσης και διαγράφει τους παλιούς μη συμβατούς πίνακες.
- Η μέθοδος long getLastCheckInTimestamp(), πάει στο τελευταίο day που έχει εισαχθεί στην βάση, και επιστρέφει από την στήλη checkIn_Date, ουσιαστικά το τελευταίο timestamp που έχει γίνει Check-In.
- Η μέθοδος addUser(User user), εισάγει έναν χρήστη στον πίνακα user της βάσης.
- Η μέθοδος addDay(Day day), εισάγει μια μέρα που έγινε Check-In στην βάση αντιστοίχως.
- Η μέθοδος User getUser(), επιστρέφει τον χρήστη που εμφανίζεται στην πρώτη γραμμή από τον πίνακα user στην βάση.
- Η μέθοδος List<Day> getAllDays(), παίρνει όλες τις καταχωρημένες ημέρες που έγινε Check-In από την βάση, τις μετατρέπει σε μορφή κλάσης Day, έπειτα τις εισάγει σε μια λίστα την οποία τελικά επιστρέφει.
- Η μέθοδος updateUser(User user), παίρνει σαν όρισμα έναν χρήστη και έπειτα μορφοποιεί τον πρώτο χρήστη που βρίσκεται στον πίνακα user στην

βάση και βάσει στην θέση του τα δεδομένα του User από το όρισμα της μεθόδου.

BottomNavigationHandler.java:

Η κλάση αποτελείται από έναν constructor που παίρνει για όρισμα ένα activity και από μία μέθοδο setupBottomNavigation.

- Η μέθοδος setupBottomNavigation(BottomNavigationView bottomNavigationView), ρυθμίζει έναν listener για τα στοιχεία της μπάρας και όταν ένα από αυτά επιλεγθεί, ελέγχει αν δεν βρίσκεται ήδη σε αυτό το activity που επιλέχθηκε και έπειτα τερματίζει την τρέχουσα δραστηριότητα, δημιουργεί ένα intent και ξεκινάει αυτή την επιλεγμένη δραστηριότητα.

bottom_nav_menu.xml:

Το αρχείο αυτό, περιγράφει το layout του μενού πλοήγησης που βρίσκεται στο κάτω μέρος της οθόνης στην εφαρμογή. Αποτελείται από 3 items, το Journey, το Check-In και το Home. Για κάθε ένα από αυτά, παρέχεται και ένα εικονίδιο καθώς και ένα id για την αναφορά τους μέσα σε αρχεία κώδικα.

MainActivity.java:

Η κλάση αυτή αποτελείται από την υλοποίηση της κύριας δραστηριότητας / Activity της εφαρμογής, το MainActivity, ή όπως αναφέρεται από την μεριά των χρηστών της εφαρμογής Home. Πιο συγκεκριμένα αποτελείται από τις παρακάτω μεθόδους:

- Η μέθοδος onCreate(Bundle savedInstanceState), είναι υπεύθυνη για την αρχικοποίηση του activity. Ενεργοποιεί κάποιες ρυθμίσεις για καλύτερη εμφάνιση του activity στην εφαρμογή, δημιουργεί ένα αντικείμενο dbHelper της βάσης δεδομένων, ελέγχει αν η βάση έχει χρήστη και αν δεν έχει τότε μεταφέρει τον χρήστη στο activity Settings για να δημιουργήσει έναν χρήστη στην βάση, εισάγοντας τις απαραίτητες ρυθμίσεις για αυτόν, εισάγει το bottom Navigation bar για να εμφανίζεται στο κάτω μέρος του activity και τέλος, καλεί μεθόδους για την αρχικοποίηση του γραφήματος καθώς και των ενημερωτικών τιμών για την πρόοδο του χρήστη.
- Η μέθοδος setupLineChart() είναι υπεύθυνη για τον ορισμό του γραφήματος. Ορίζει τα labels, το μέγεθος και το χρώμα τους για κάθε άξονα και τέλος απενεργοποιεί τον έξτρα δεξιό άξονα y.
- Η μέθοδος plotCravingsData() είναι υπεύθυνη για την εισαγωγή τιμών από τις ημέρες στην βάση στο παραπάνω ορισμένο γράφημα. Δημιουργεί μια λίστα Days, με αρχικό index 1, το οποίο αυξάνεται για κάθε εισαγωγή μέρας και παίρνει το cravingsPercentage για κάθε μία από αυτές. Έπειτα, δημιουργεί ένα LineDataSet με αυτά τα δεδομένα, τα εισάγει στο γράφημα, επεξεργάζεται το χρώμα και το μέγεθος αυτών και τέλος ενημερώνει την κατάσταση του γραφήματος για την προσθήκη τους.

- Η μέθοδος `updateUserValues()`, ενημερώνει τα `textView` στο `activity` για το `days Clean` του χρήστη και το `Euros Saved` αυτού με τις κατάλληλες τιμές. Αρχικά εισάγει τον χρήστη από την βάση και παίρνει τα `days Clean` του για την ενημέρωση του πρώτου `textView`. Για την ενημέρωση του `Euros Saved`, πολλαπλασιάζει το `days clean` με το `packages per day` και το `euros per package`.
- Τέλος, η μέθοδος `openSettings(View view)`, ανοίγει το `activity Settings` όταν πατηθεί το αντίστοιχο κουμπί.

activity_main.xml:

Το αρχείο αυτό, περιγράφει το `layout` του `mainActivity`. Αποτελείται από ένα `ImageView` που χρησιμοποιείται για την εικόνα του `Background`, το `BottomNavigationView` για την εισαγωγή του μενού πλοήγησης, 1 `TextView` που χρησιμοποιείται απλώς σαν `Label` στην κορυφή του `Activity` με τον τίτλο της εφαρμογής, 2 `textView` για την προβολή των τιμών `Days Clean` και `Euros Saved`, ένα κουμπί για την πλοήγηση στα `Settings` και τέλος την εισαγωγή του διαγράμματος.

Journey.java:

Η κλάση αυτή αποτελεί την υλοποίηση του `activity Journey`, όπου χρησιμοποιείται για την προβολή των ημερών που έγινε `Check-In` (σημείωμα της ημέρας, ακριβείς ώρα που έγινε το `check-in`, `cravings percentage`). Πιο συγκεκριμένα αποτελείται από:

- Η μέθοδος `onCreate(Bundle savedInstanceState)`, είναι υπεύθυνη για την αρχικοποίηση του `activity`. Ενεργοποιεί κάποιες ρυθμίσεις για καλύτερη εμφάνιση του `activity` στην εφαρμογή, δημιουργεί ένα αντικείμενο `dbHandler` της βάσης δεδομένων, εισάγει το `bottom Navigation bar`, βρίσκει το αντικείμενο με `id listView` και το ορίζει με την μεταβλητή `listView` και έπειτα καλεί την μέθοδο `displayAllDays()`.
- Η μέθοδος `displayAllDays()` παίρνει μια λίστα με όλες τις ημέρες από την βάση, δημιουργεί έναν προσαρμογέα που συνδέει τα δεδομένα της λίστας `daysList` με το `listView`, παίρνει κάθε `day` και εισάγει τα δεδομένα του σε `textView` καθώς και τα μορφοποιεί για να είναι πιο ευανάγνωστα από τον χρήστη.

List_item_day.xml:

Το αρχείο αυτό, περιγράφει απλά κάποια χαρακτηριστικά των `textView` που χρησιμοποιούνται για την προβολή αντικειμένων `Day` από την κλάση `Journey`. Κυρίως έχει πληροφορίες για την ομορφότερη εμφάνιση, καθώς και το αναγνωριστικό `id` του `textViewDay`.

activity_journey.xml:

Το αρχείο αυτό, περιγράφει το layout του activity Journey. Αποτελείται από ένα ImageView που χρησιμοποιείται για την εικόνα του Background, το BottomNavigationView για την εισαγωγή του μενού πλοήγησης, και ένα ListView για την προβολή των ημερών που έγινε Check-In.

CheckIn.java:

Η κλάση αυτή είναι υπεύθυνη για την υλοποίηση του activity Check In. Αποτελείται από τις παρακάτω μεθόδους:

- Η μέθοδος onCreate(Bundle savedInstanceState), είναι υπεύθυνη για την αρχικοποίηση του activity. Ενεργοποιεί κάποιες ρυθμίσεις για καλύτερη εμφάνιση του activity στην εφαρμογή, εισάγει το BottomNavigationBar, δημιουργεί ένα αντικείμενο dbHelper της βάσης δεδομένων, και παίρνει τα αντικείμενα editTextExplanation και seekBarCravingsStrength για περαιτέρω επεξεργασία στην παρακάτω μέθοδο.
- Η μέθοδος submitCheckIn(View view), αρχικά παίρνει ένα String με το περιεχόμενο του editTextExplanation και ένα int με το ποσοστό του seekBar (καθώς και το πολλαπλασιάζει *10 για να είναι ποσοστό %), παίρνει από την βάση δεδομένων το τελευταίο timestamp που είχε γίνει submit και ξεκινάει μια σειρά υπολογισμούς για να ελέγξει αν έχουν περάσει 24 ώρες από το currentTimestamp που παίρνει από την συσκευή. Αν έχουν περάσει 24 ώρες προσθέτει μία νέα ημέρα στην βάση δεδομένων με τα παραπάνω δεδομένα και εμφανίζει ειδοποιητικό μήνυμα ότι το Check-In ολοκληρώθηκε επιτυχώς, αλλιώς βγάζει ειδοποιητικό μήνυμα πως δεν έχουν περάσει 24 ώρες. Και στις 2 περιπτώσεις μετά από το ειδοποιητικό μήνυμα ,ανακατευθύνει τον χρήστη στο MainActivity και τερματίζει το activity του Check-In.

activity_check_in.xml:

Το αρχείο αυτό, περιγράφει το layout του activity Check-In. Αποτελείται από ένα ImageView που χρησιμοποιείται για την εικόνα του Background, ένα TextView που επεξηγεί τι θα πρέπει να περιέχει η είσοδος στο κάθε πεδίο του Check-in, το SeekBar, 2 TextViews για την περιγραφή των 2 άκρων του SeekBar (Not Strong/ Extremely Strong), ένα EditText για την εισαγωγή του κειμένου Note από τον χρήστη, ένα κουμπί για το Submit (Υποβολή) του Check-In και τέλος το BottomNavigationView για την εισαγωγή του μενού πλοήγησης.

Settings.java:

Η κλάση αυτή είναι υπεύθυνη για την υλοποίηση του activity Settings. Αποτελείται από τις παρακάτω μεθόδους:

- Η μέθοδος onCreate(Bundle savedInstanceState), είναι υπεύθυνη για την αρχικοποίηση του activity. Ενεργοποιεί κάποιες ρυθμίσεις για καλύτερη εμφάνιση του activity στην εφαρμογή, εισάγει το BottomNavigationBar, δημιουργεί ένα αντικείμενο dbHelper της βάσης δεδομένων, και καλεί την μέθοδο loadUserSettings().

- Η μέθοδος `loadUserSettings()`, παίρνει τον χρήστη από την βάση δεδομένων και εισάγει την τιμή ευρώ για κάθε πακέτο τσιγάρων που παίρνει καθώς και τον αριθμό πακέτων ανά ημέρα στις μεταβλητές `editEurosPerPackage` και `editPackagesPerDay` αντιστοίχως.
- Η μέθοδος `resetDaysClean(View view)`, καλείται όταν πατιέται το κουμπί `Reset Days Clean`, παίρνει τον χρήστη από την βάση, θέτει την τιμή `daysClean = 0` και ανανεώνει την βάση. Έπειτα βγάζει ενημερωτικό μήνυμα ότι η αλλαγή έγινε επιτυχώς ή βγάζει ενημερωτικό μήνυμα ότι ο χρήστης δεν βρέθηκε σε περίπτωση που κάποιος πατήσει το κουμπί όταν ακόμη το `Settings` εμφανίζεται σαν αρχική καρτέλα στους χρήστες μέχρι να εισάγουν τα αρχικά τους `Settings` μετά την εγκατάσταση.
- Η μέθοδος `submitSettings(View view)`, καλείται όταν πατιέται το κουμπί `Submit`, αρχικά παίρνει τα δεδομένα του `editEurosPerPackage` και του `editPackagesPerDay` και προσπαθεί να τα εισάγει στην βάση δεδομένων. Αν η εισαγωγή ολοκληρωθεί επιτυχώς, εμφανίζεται ενημερωτικό μήνυμα ότι οι ρυθμίσεις ενημερώθηκαν ή ότι ο νέος χρήστης δημιουργήθηκε και οι ρυθμίσεις του αποθηκεύτηκαν αν δεν υπήρχε χρήστης ως τώρα. Αλλιώς εμφανίζεται ενημερωτικό μήνυμα ότι τα νούμερα που δόθηκαν δεν ήταν σωστά.

activity_settings.xml:

Το αρχείο αυτό, περιγράφει το layout του activity `Settings`. Αποτελείται από ένα `ImageView` που χρησιμοποιείται για την εικόνα του `Background`, το `BottomNavigationView` για την εισαγωγή του μενού πλοήγησης, 2 `EditText` για την εισαγωγή των δεδομένων `editEurosPerPackage` και `editPackagesPerDay`, ένα κουμπί για το `Reset Days Clean` και τέλος ένα κουμπί για το `Submit` των δεδομένων στα `Settings`.

Συνεισφορά των μελών ομάδας:

- Ο Γεώργιος Δάλλας συνεισέφερε τα εξής: α) σχεδιασμός και υλοποίηση του mainActivity καθώς και την εισαγωγή του διαγράμματος σε αυτό. β) σχεδιασμός και υλοποίηση του μενού πλοήγησης στο κάτω μέρος της οθόνης BottomNavigationView. γ) Συγγραφή της αναφοράς.
- Ο Κωσταντίνος Τσίγκας συνεισέφερε τα εξής: α) σχεδιασμός και υλοποίηση του Journey. β) σχεδιασμός ,υλοποίηση και αρχική διασύνδεση με τα activities της βάσης δεδομένων. γ) προετοιμασία βίντεο-demo.
- Ο Κωσταντίνος Δεμερδεσλής συνεισέφερε τα εξής: α) σχεδιασμός και υλοποίηση του Check-In. β) σχεδιασμός και υλοποίηση του Settings. γ) υλοποίηση του User/Day.class.

Η υλοποίηση των γραφικών έγινε ξεχωριστά από κάθε άτομο στα activities που ανέλαβαν.