



Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών (ΠΜΣ) στις
«Ψηφιακές Εφαρμογές και την Καινοτομία»

Δημιουργία εφαρμογής ανάκτησης iCals και απεικόνισης των
γεγονότων τους

Ράπτης Μιλτιάδης inf.dig2203

Κέρκυρα, Φεβρουάριος 2024

Περιεχόμενα

Περίληψη	4
Εισαγωγή	5
Κεφάλαιο 1: Τεχνολογίες	7
1.1 Εργαλεία ανάπτυξης	7
1.1.1 Flutter (Software).....	7
1.1.2 Visual Studio Code (IDE)	8
1.1.3 Android Studio	8
1.1.4 Firebase	9
1.2 Packages.....	10
1.2.1 Διεπαφή χρήστη και χαρακτηριστικά ημερολογίου	10
1.2.2 Δικτύωση και αποθήκευση δεδομένων	11
1.2.3 Ενσωμάτωση Firebase.....	11
1.2.4 Βελτιώσεις διαχείρισης κατάστασης και διεπαφής χρήστη	12
1.2.5 Πλοήγηση και χειρισμός εικόνας	12
1.2.6 Διαχείριση αρχείων και βελτιστοποίηση εικόνας	13
1.3 Ασφάλεια.....	13
1.3.1 SHA certificate fingerprints	13
Κεφάλαιο 2: Υλοποίηση	14
2.1. Τεχνικά χαρακτηριστικά	15
2.2. Διαχείριση Λογαριασμού Χρήστη	20
2.3. Αποθήκευση iCal Link	22
2.4. Διάβασμα Συνδέσμων iCal	24
2.5. Εμφάνιση Συμβάντος Στο Ημερολόγιο	27
2.6. Δομή αρχείων και διαλογή πληροφοριών	30
2.7. Διεπαφή χρήστη και πλοήγηση.....	33
2.7.1 Sign-in/ Register Pages	34
2.7.1 Home Page	37
2.7.2 Calendar Page	39
2.7.3 Profile Page.....	41
2.7.4 Menu Page	42
2.7.5 Settings Page.....	43
2.7.6 Reservations Detail Page	45
2.7.7 Accommodations Page.....	47
2.7.8 Notifications Page.....	48
2.8. Δοκιμές	49

2.9 Ολοκλήρωση της εφαρμογής	50
Κεφάλαιο 3: Μελλοντικές Βελτιώσεις	50
3.1 Βελτιωμένη Διεπαφή Χρήστη	50
3.2 Χαρακτηριστικά Ημερολογίου	51
3.3 Ασφάλεια και έλεγχος ταυτότητας	51
3.4 Βελτιστοποίηση Απόδοσης.....	52
3.5 Πρόσθετες Ενσωματώσεις.....	53
3.6 Προσβασιμότητα	53
3.7 Δοκιμές και διασφάλιση ποιότητας.....	54
3.8 Βελτιώσεις Επεκτασιμότητας και Υποστήριξης.....	54
3.9 Ολοκλήρωση διαδικασίας ανάκτησης κωδικού.....	54
Συμπεράσματα.....	55
Βιβλιογραφία	56
Παραρτήματα	58
Παράρτημα Α. Τεχνικά χαρακτηριστικά.....	58
Α1. Δεδομένα Εικόνας Προφίλ.....	58

Περίληψη

Αυτό το Capstone Project παρουσιάζει την ανάπτυξη μιας εφαρμογής για κινητά χρησιμοποιώντας τη γλώσσα προγραμματισμού Flutter, σχεδιασμένη για την απεικόνιση των κρατήσεων μέσω των συνδέσεων iCal από διάφορες πλατφόρμες όπως Airbnb και Booking. Η εφαρμογή επιτρέπει στους χρήστες να δημιουργούν λογαριασμούς, να αποθηκεύουν τους iCal συνδέσμους τους στο Firebase και να απεικονίζουν τις κρατήσεις τους σε μια βολική προβολή ημερολογίου. Η εφαρμογή αντιμετωπίζει την ανάγκη για μια κεντρική λύση για τη διαχείριση διαφορετικών δεδομένων iCal, παρέχοντας στους χρήστες μια φιλική διεπαφή για να αλληλεπιδρούν με τα χρονοδιαγράμματα τους. Μέσω της ενσωμάτωσης του Flutter για τις υπηρεσίες frontend και Firebase για backend, η εφαρμογή επιτυγχάνει μια απρόσκοπτη εμπειρία χρήστη. Η εργασία αυτή περιγράφει λεπτομερώς τους στόχους του έργου, τη μεθοδολογία, το σχεδιασμό του συστήματος, τη διαδικασία υλοποίησης και την προσέγγιση των δοκιμών. Τα αποτελέσματα καταδεικνύουν την επιτυχή ανάκτηση και εμφάνιση των γεγονότων iCal, με ανάλυση των πλεονεκτημάτων, των περιορισμών και τις πιθανές μελλοντικές βελτιώσεις της εφαρμογής. Συνολικά, το έργο αυτό συμβάλλει στον τομέα της ανάπτυξης εφαρμογών για κινητά, προσφέροντας μια αποτελεσματική λύση για τους χρήστες ή τις εταιρίες διαχείρισης καταλυμάτων να οργανώνουν και να έχουν πρόσβαση στα δεδομένα ημερολογίου τους σε πολλές πλατφόρμες.

Εισαγωγή

Με την συνεχή ανάπτυξη του τουρισμού και την αύξηση του αριθμού επισκεπτών από άλλες χώρες, κύριο καθήκον του Υπουργείου Τουρισμού (Greek ministry of tourism, 2024) είναι η χάραξη της τουριστικής πολιτικής και ο σχεδιασμός της εφαρμογής της τουριστικής ανάπτυξης στο πλαίσιο της κυβερνητικής πολιτικής.

Το Υπουργείο εκτελεί το έργο αυτό με τη διαμόρφωση βραχυπρόθεσμων και μακροπρόθεσμων προγραμμάτων για την αύξηση της ανταγωνιστικότητας των τουριστικών προϊόντων, τη στήριξη των επενδύσεων, την προώθηση της τουριστικής εκπαίδευσης και κατάρτισης και τη συντονισμένη επικοινωνία για την ενίσχυση της εικόνας της Ελλάδας στο εξωτερικό.

Σε μια εποχή που χαρακτηρίζεται από την ραγδαία ανάπτυξη του τουρισμού σε συνδυασμό με την πανταχού παρουσία ψηφιακών πλατφορμών και την αυξανόμενη πολυπλοκότητα της διαχείρισης διαφορετικών χρονοδιαγραμμάτων, η ανάγκη για αποτελεσματικές λύσεις για την διαχείριση κρατήσεων έχει γίνει όλο και πιο εμφανής.

Υπάρχουν διάφορες πλατφόρμες απεικόνισης δεδομένων iCal στην αγορά όπως NoBeds (NOBEDS, 2023), Rentals United (Euroart, 2024), Beds24 (Channel Manager, Property Management System & Booking Engine, 2006) και άλλες ελληνικές και ξένες, το καθένα με τα δικά του πλεονεκτήματα και μειονεκτήματα.

Αυτό το Capstone Project αντιμετωπίζει αυτή την ανάγκη μέσω της ανάπτυξης μιας εφαρμογής για κινητά, αξιοποιώντας την ευελιξία του πλαισίου Flutter και τις ισχυρές δυνατότητες αποθήκευσης δεδομένων της Firebase.

Στα πλαίσια του μεταπτυχιακού και πιο συγκεκριμένα στο μάθημα Κινητό Υπολογίζειν και Εφαρμογές (Mobile Computing and Applications) είχε ζητηθεί από την καθηγήτρια η ανάπτυξη μιας εφαρμογής. Η εφαρμογή (εν ονόματι UPReservations) δεχόταν δεδομένα κρατήσεων από ένα αρχείο μορφής xlsx, το οποίο βρισκόταν στο Google Drive και ανανεωνόταν χειροκίνητα. Με σκοπό την εξέλιξη της ξεκίνησε αυτό το Capstone Project, για λόγους αυτοματοποίησης, αξιοπιστίας και καλύτερου ελέγχου.

Αυτή η εισαγωγή θέτει το στάδιο αναγνωρίζοντας το πρόβλημα που επικρατεί και διατυπώνει τη σημασία της προτεινόμενης λύσης. Επιτρέποντας στους χρήστες να δημιουργούν λογαριασμούς, να αποθηκεύουν συνδέσμους iCal στο Firebase και να απεικονίζουν κρατήσεις μέσω μιας διαισθητικής προβολής ημερολογίου, η εφαρμογή στοχεύει να παρέχει μια φιλική προς το χρήστη και κεντρική πλατφόρμα για τη διαχείριση διαφορετικών πληροφοριών προγραμματισμού.

Σε όλη αυτή την έκθεση, θα εμβαθύνουμε στη λεπτομερή μεθοδολογία, το σχεδιασμό του συστήματος και τη διαδικασία υλοποίησης που ακολουθήθηκε για την δημιουργία της εφαρμογής. Επιπλέον, οι συζητήσεις σχετικά με τις δοκιμές, τα αποτελέσματα και τους προβληματισμούς σχετικά με τα πλεονεκτήματα και τους περιορισμούς του έργου θα συμβάλουν στην πλήρη κατανόηση της διαδικασίας ανάπτυξης και των αποτελεσμάτων. Η

ενσωμάτωση των τεχνολογιών Flutter και Firebase υπογραμμίζει τη δέσμευσή μας να παρέχουμε μια απρόσκοπτη και αποτελεσματική εμπειρία χρήστη.

Αντιμετωπίζοντας τις προκλήσεις που σχετίζονται με τη διαχείριση συνδέσμων iCal από διάφορες πηγές, αυτό το Capstone Project όχι μόνο συμβάλλει στον τομέα της ανάπτυξης εφαρμογών για κινητά, αλλά παρέχει επίσης μια πρακτική και προσβάσιμη λύση για χρήστες που επιδιώκουν να βελτιστοποιήσουν τη διαχείριση των κρατήσεων των καταλυμάτων οι οποίες έρχονται από διαφορετικές πλατφόρμες.

Όλα τα αρχεία της εφαρμογής βρίσκονται στο Github:

<https://github.com/MiltiadisR/CapstoneProject>

Κεφάλαιο 1: Τεχνολογίες

Κατά την ανάπτυξη της εφαρμογής, ενσωματώθηκαν διάφορα πακέτα για τη βελτίωση της λειτουργικότητας, των διαδικασιών ανάπτυξης και την παροχή πλούσιας εμπειρίας χρήστη. Η ακόλουθη ενότητα παρέχει λεπτομερείς πληροφορίες για κάθε πακέτο που χρησιμοποιείται, μαζί με τα βασικά χαρακτηριστικά και τις συγκεκριμένες εκδόσεις τους.

1.1 Εργαλεία ανάπτυξης

1.1.1 Flutter (Software)

Το Flutter (Flutter - Build Apps for Any Screen, 2017), ένα εργαλείο ανάπτυξης λογισμικού UI ανοιχτού κώδικα που αναπτύχθηκε από την Google, δίνει τη δυνατότητα στους προγραμματιστές να δημιουργούν εφαρμογές για κινητές, διαδικτυακές και επιτραπέζιες πλατφόρμες από μια ενιαία βάση κώδικα, προσφέροντας ένα ευέλικτο σύνολο εργαλείων και βιβλιοθηκών

Τα ξεχωριστά χαρακτηριστικά του Flutter είναι το Hot Reload, επιτρέποντας πειραματισμό σε πραγματικό χρόνο, μια ενιαία βάση κώδικα για ανάπτυξη πολλαπλών πλατφορμών, ένα πλούσιο σύνολο προσαρμόσιμων widgets για την κατασκευή UIs (User Interface), στοιχεία UI για εκπληκτικές διεπαφές με ομαλές κινούμενες εικόνες, και τη γλώσσα προγραμματισμού Dart, σχεδιασμένη για σύγχρονες και επεκτάσιμες εφαρμογές. Επιπλέον, το Flutter διαθέτει μια ζωντανή κοινότητα, συμβάλλοντας σε ένα εκτεταμένο οικοσύστημα πακέτων και πρόσθετων.

Οι προγραμματιστές αξιοποιούν το Flutter τόσο στις πλατφόρμες iOS όσο και στο Android. Επιπλέον, το Flutter επεκτείνει τις δυνατότητές του στην ανάπτυξη ιστοσελίδων, επιτρέποντας τη δημιουργία ανταποκρινόμενων εφαρμογών ιστού. Με πειραματική υποστήριξη για πλατφόρμες υπολογιστών όπως Windows, macOS και Linux, το Flutter παρέχει μια ενοποιημένη λύση για την ανάπτυξη πολλαπλών πλατφορμών.

Το Flutter διευκολύνει την απρόσκοπτη ενσωμάτωση με εγγενείς λειτουργίες πλατφόρμας μέσω καναλιών πλατφόρμας, επιτρέποντας την επικοινωνία μεταξύ του Κώδικα Dart και του εγγενούς κώδικα. Το οικοσύστημα πρόσθετων του πλαισίου ενισχύει την αποτελεσματικότητα προσφέροντας προκατασκευασμένα πακέτα για κοινές λειτουργίες, εξασφαλίζοντας εύκολη ενσωμάτωση των δυνατοτήτων δικτύωσης, ελέγχου ταυτότητας και βάσης δεδομένων.

Υποστηρίζεται από δημοφιλή IDEs όπως το Visual Studio Code, το οποίο χρησιμοποιήθηκε για την ανάπτυξη της συγκεκριμένης εφαρμογής, και το IntelliJ IDEA, το Flutter παρέχει ένα ισχυρό περιβάλλον ανάπτυξης. Τα ολοκληρωμένα εργαλεία ανάπτυξης προσφέρουν χαρακτηριστικά όπως ολοκλήρωση κώδικα, εντοπισμό σφαλμάτων και υποστήριξη πρόσθετων. Το DevTools, μια πληθώρα εργαλείων απόδοσης και εντοπισμού σφαλμάτων, βοηθά τους προγραμματιστές στην ανάλυση και βελτιστοποίηση των εφαρμογών τους.

Το Flutter λαμβάνει τακτικές ενημερώσεις από την Google, εισάγοντας νέες δυνατότητες, βελτιώσεις και διορθώσεις σφαλμάτων. Το πλαίσιο επωφελείται από μια ενεργή κοινότητα που συμβάλλει στην ανάπτυξή του, στην ανταλλαγή γνώσεων, στη δημιουργία προσθηκών και στη συμμετοχή σε εκδηλώσεις όπως συνέδρια Flutter.

1.1.2 Visual Studio Code (IDE)

Το ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment, IDE) που χρησιμοποιήθηκε είναι ο Visual Studio Code (Visual Studio Code - Code Editing, 2015). Με το διαισθητικό περιβάλλον εργασίας χρήστη, το VS Code παρέχει μια φιλική προς το χρήστη εμπειρία, που εξυπηρετεί προγραμματιστές διαφορετικών επιπέδων εμπειρίας. Ο καθαρός σχεδιασμός και τα καλά οργανωμένα μενού συμβάλλουν σε μια βελτιωμένη ροή ανάπτυξης, προωθώντας την αποτελεσματικότητα και την ευκολία χρήσης.

Το VS Code επιδεικνύει ευελιξία προσφέροντας υποστήριξη για ένα ευρύ φάσμα γλωσσών προγραμματισμού. Ο επεξεργαστής κώδικα παρέχει βασικά χαρακτηριστικά όπως επισήμανση σύνταξης, αυτόματη συμπλήρωση (IntelliSense) και μορφοποίηση κώδικα. Αυτές οι δυνατότητες ενισχύουν την αναγνωσιμότητα του κώδικα και βοηθούν τους προγραμματιστές να γράφουν αποτελεσματικό και χωρίς σφάλματα κώδικα.

Ένα αξιοσημείωτο χαρακτηριστικό είναι το ενσωματωμένο τερματικό εντός του κώδικα VS, επιτρέποντας στους προγραμματιστές να εκτελούν εντολές και να εκτελούν σενάρια απευθείας εντός του IDE. Αυτή η ολοκληρωμένη προσέγγιση εξαλείφει την ανάγκη εναλλαγής μεταξύ πολλαπλών εφαρμογών, προωθώντας ένα συνεκτικό περιβάλλον ανάπτυξης.

Το IDE υποστηρίζει μια εκτεταμένη βιβλιοθήκη επεκτάσεων που συνεισφέρουν η κοινότητα και η Microsoft. Αυτές οι επεκτάσεις ενισχύουν τη λειτουργικότητα και επιτρέπουν στους προγραμματιστές να προσαρμόζουν το περιβάλλον τους με βάση τις συγκεκριμένες απαιτήσεις τους.

Το VS Code ενσωματώνεται απρόσκοπτα με συστήματα ελέγχου εκδόσεων όπως το Git, το οποίο χρησιμοποιήθηκε για την ανάρτηση του κώδικα στο GitHub. Οι επιλογές θεμάτων και οι δυνατότητες προσαρμογής του IDE επιτρέπουν στους προγραμματιστές να εξατομικεύσουν το περιβάλλον κωδικοποίησης τους, συμβάλλοντας σε μια άνετη και οπτικά ευχάριστη εμπειρία ανάπτυξης.

Η ενεργή κοινότητα που περιβάλλει τον κώδικα του Visual Studio είναι ένα σημαντικό πλεονέκτημα. Με εκτεταμένη τεκμηρίωση, φόρουμ και κανάλια υποστήριξης με γνώμονα την κοινότητα, οι προγραμματιστές μπορούν να βρουν βοήθεια και λύσεις στις προκλήσεις. Η δέσμευση της Microsoft για τακτικές ενημερώσεις και βελτιώσεις διασφαλίζει ότι το VS Code παραμένει ένα εργαλείο συνεχούς ανάπτυξης, αντιμετωπίζοντας ζητήματα απόδοσης και εισάγοντας νέες δυνατότητες.

Συμπερασματικά, ο Visual Studio Code έχει αποδειχθεί απαραίτητο εργαλείο για την εφαρμογή. Το πλούσιο σύνολο χαρακτηριστικών του, η επεκτασιμότητα μέσω επεκτάσεων, οι δυνατότητες συνεργασίας και η ισχυρή υποστήριξη της κοινότητας συμβάλλουν σε μια παραγωγική και ευχάριστη εμπειρία ανάπτυξης.

1.1.3 Android Studio

Το Android Studio Virtual Device Manager (AVD Manager, Android Studio & App Tools - Android Developers, 2013) είναι ένα κρίσιμο στοιχείο που χρησιμοποιήθηκε ως εικονική μηχανή (Virtual Machine) για τη δοκιμή και τον εντοπισμό σφαλμάτων. Επιτρέπει στους

προγραμματιστές να μιμούνται διαφορετικές συσκευές και διαμορφώσεις Android, στην συγκεκριμένη περίπτωση το Nexus 6, παρέχοντας ένα προσομοιωμένο περιβάλλον για δοκιμές εφαρμογών χωρίς την ανάγκη φυσικών συσκευών. Αυτή η λειτουργία εικονικής μηχανής είναι καθοριστικής σημασίας για τη διασφάλιση ότι οι εφαρμογές συμπεριφέρονται με συνέπεια σε διάφορα μεγέθη οθόνης, αναλύσεις και εκδόσεις Android.

Ο διαχειριστής AVD επιτρέπει στους προγραμματιστές να δημιουργούν και να διαχειρίζονται εικονικές συσκευές με συγκεκριμένα προφίλ υλικού, συμπεριλαμβανομένων ποικίλων ποσοτήτων μνήμης RAM, αποθήκευσης και διαφορετικών επιπέδων API. Η ικανότητα να μιμούνται διαφορετικές διαμορφώσεις συσκευών βοηθά στον εντοπισμό πιθανών ζητημάτων που σχετίζονται με την απόδοση, τη διάταξη και τη συμβατότητα. Επιπλέον, ο διαχειριστής AVD ενσωματώνεται άψογα με το Android Studio IDE, επιτρέποντας την αποτελεσματική ανάπτυξη εφαρμογών σε εικονικές συσκευές για δοκιμές και εντοπισμό σφαλμάτων σε πραγματικό χρόνο.

1.1.4 Firebase

Η Firebase, μια ισχυρή πλατφόρμα ανάπτυξης εφαρμογών για κινητά και web που δημιουργήθηκε από την Google, συνδυάζει απρόσκοπτα ένα φάσμα εργαλείων και υπηρεσιών, δίνοντας τη δυνατότητα στους προγραμματιστές να κατασκευάζουν γρήγορα εφαρμογές υψηλής ποιότητας και επεκτάσιμες. Στον πυρήνα της, η Firebase ενσωματώνει μια σειρά χαρακτηριστικών, το καθένα σχολαστικά σχεδιασμένο για να εξυπηρετεί διαφορετικές πτυχές της ανάπτυξης εφαρμογών, που κυμαίνονται από τη διαχείριση βάσεων δεδομένων και τον έλεγχο ταυτότητας έως τις λειτουργίες cloud και τη φιλοξενία.

Η βάση δεδομένων σε πραγματικό χρόνο (Realtime Database), χαρακτηριστικό γνώρισμα της Firebase, προσφέρει μια βάση δεδομένων NoSQL που βασίζεται στο Cloud και βοηθάει στον συγχρονισμό δεδομένων σε πραγματικό χρόνο μεταξύ πελατών και διακομιστών. Αυτό αποδεικνύεται ανεκτίμητο για εφαρμογές που απαιτούν ζωντανές ενημερώσεις, λειτουργίες όπως συνομιλία σε πραγματικό χρόνο και εργαλεία συνεργασίας.

Στον τομέα του ελέγχου ταυτότητας χρήστη, ο έλεγχος ταυτότητας Firebase (Firebase Authentication) βρίσκεται στο επίκεντρο, παρέχοντας μια ασφαλή και αβίαστα εφαρμόσιμη λύση. Υποστηρίζοντας πολλαπλές μεθόδους ελέγχου ταυτότητας, συμπεριλαμβανομένου του ηλεκτρονικού ταχυδρομείου/κωδικού πρόσβασης, και την ενσωμάτωση με συνδέσεις κοινωνικών μέσων όπως το Google και το Facebook, διευκολύνει μια απρόσκοπτη εμπειρία ενσωμάτωσης χρηστών.

Συμπληρώνοντας τη βάση δεδομένων σε πραγματικό χρόνο, το Firebase παρουσιάζει το Firestore, μια άλλη βάση δεδομένων NoSQL γνωστή για τις προηγμένες δυνατότητες αναζήτησης και την επεκτασιμότητα. Το Firestore γίνεται συχνά η βάση δεδομένων επιλογής καθώς οι εφαρμογές εξελίσσονται και επεκτείνονται.

Οι λειτουργίες του Firebase Cloud αναδύονται ως ένα ισχυρό εργαλείο, επιτρέποντας στους προγραμματιστές να εκτελούν κώδικα backend ως απάντηση σε συμβάντα που ενεργοποιούνται από λειτουργίες Firebase ή εξωτερικά αιτήματα HTTP. Αυτή η πλατφόρμα υπολογιστών χωρίς διακομιστές απελευθερώνει προσαρμοσμένη λογική χωρίς το επαχθές έργο της διαχείρισης διακομιστή.

Η Firebase Hosting, μια ασφαλής και επεκτάσιμη λύση, προσφέρει ένα καταφύγιο για εφαρμογές ιστού και στατικό περιεχόμενο. Εξοπλισμένο με αυτόματα πιστοποιητικά SSL, ενσωμάτωση CDN και εύκολη ανάπτυξη, εξυπηρετεί τις ανάγκες φιλοξενίας εφαρμογών.

Η υπηρεσία αποθήκευσης (Storage) επεκτείνει απρόσκοπτα τις δυνατότητες της Firebase για τη διαχείριση περιεχομένου που δημιουργείται από χρήστες, όπως εικόνες, βίντεο και αρχεία. Σχεδιασμένο για να συμπληρώνει άλλες υπηρεσίες Firebase, το Firebase Storage εξασφαλίζει μια ολιστική λύση για τη διαχείριση δεδομένων.

Στον τομέα της εμπλοκής των χρηστών, το Firebase Cloud Messaging (FCM), επιτρέπει στους προγραμματιστές να αποστέλλουν ειδοποιήσεις push σε διάφορες πλατφόρμες. Αυτό αποδεικνύεται καθοριστικό για την ενημέρωση των χρηστών και την εμπλοκή τους με έγκαιρες ενημερώσεις ή σχετικά γεγονότα.

Ο έλεγχος ταυτότητας παραμένει ο λίθος της Firebase, παρέχοντας στους προγραμματιστές ένα φιλικό προς το χρήστη σύστημα για τη διαχείριση των συνδέσεων χρηστών και της επαλήθευσης ταυτότητας. Υποστηρίζοντας διαφορετικούς παρόχους ελέγχου ταυτότητας και αρμονικά ενσωματώνοντας με άλλες υπηρεσίες Firebase, γίνεται απαραίτητο συστατικό του οικοσυστήματος Firebase.

Το Firebase ML Kit, μια απόδειξη της ευελιξίας του Firebase, φέρνει δυνατότητες μηχανικής μάθησης στους προγραμματιστές χωρίς τις πολυπλοκότητες που σχετίζονται με την εκτεταμένη τεχνογνωσία. Προσφέροντας προ-εκπαιδευμένα μοντέλα για εργασίες όπως επισημάνση εικόνας, Αναγνώριση κειμένου και ανίχνευση προσώπου, ανοίγει νέες διαστάσεις για λειτουργίες εφαρμογών.

1.2 Packages

1.2.1 Διεπαφή χρήστη και χαρακτηριστικά ημερολογίου

cupertino_icons (^1.0.2):

Αυτό το πακέτο παρέχει εικονίδια σε στυλ Cupertino (Cupertino_Icons | Dart Package, 2023) για την εφαρμογή iOS, ενισχύοντας την οπτική του γοητεία και διασφαλίζοντας μια συνεπή γλώσσα σχεδιασμού.

intl (^0.18.1):

Διευκολύνοντας τη διεθνοποίηση και τον εντοπισμό, το πακέτο intl διασφαλίζει ότι η εφαρμογή μας είναι προσαρμόσιμη για χρήστες με διαφορετικές προτιμήσεις γλώσσας και περιοχής.

table_calendar (^3.0.9):

Η ενσωμάτωση του πακέτου table_calendar (Table_Calendar | Flutter Package, 2023) έπαιξε καθοριστικό ρόλο στην εφαρμογή της προβολής ημερολογίου στην εφαρμογή μας για κινητά. Αναπτύχθηκε και συντηρήθηκε από άτομα της κοινότητας του Flutter, αυτό το ευέλικτο πακέτο Flutter αποδείχθηκε καθοριστικό για τη δημιουργία ενός διαδραστικού και οπτικά ελκυστικού UI ημερολογίου. Το πακέτο προσφέρει ένα ευρύ φάσμα χαρακτηριστικών, όπως Επιλογή ημερομηνίας, χειρισμό συμβάντων και ισχυρή ανταπόκριση. Ένα από τα αξιοσημείωτα πλεονεκτήματά του είναι το ολοκληρωμένο σύνολο επιλογών προσαρμογής, επιτρέποντάς μας να προσαρμόσουμε την εμφάνιση του

ημερολογίου ώστε να συμπίπτει με την οπτική ταυτότητα της εφαρμογής μας. Η ενσωματωμένη υποστήριξη για το χειρισμό συμβάντων σε συγκεκριμένες ημερομηνίες αποδείχθηκε ανεκτίμητη για την εφαρμογή μας, ιδιαίτερα στο πλαίσιο της διαχείρισης και προβολής συμβάντων iCal. Το πακέτο `table_calendar` ενσωματώνεται απρόσκοπτα στο ευρύτερο οικοσύστημα Flutter, εξασφαλίζοντας συμβατότητα με άλλα widgets Flutter και παρέχοντας ένα ανταποκρινόμενο σχέδιο που βελτιώνει τη συνολική εμπειρία του χρήστη. Η συντήρησή του από την κοινότητα Flutter και οι ενεργές συνεισφορές στο GitHub υπογραμμίζουν την αξιοπιστία και την προσαρμοστικότητα του πακέτου. Συνοπτικά, η συμπερίληψη του πακέτου `table_calendar` συνέβαλε σημαντικά στην ανάπτυξη μιας ισχυρής και φιλικής προς το χρήστη δυνατότητας ημερολογίου στην εφαρμογή μας.

1.2.2 Δικτύωση και αποθήκευση δεδομένων

http (^1.1.0):

Το πακέτο `http` (Http | Dart Package, 2023) απλοποιεί τη διαδικασία υποβολής αιτημάτων HTTP και ανάκτησης δεδομένων από εξωτερικά API. Παρέχει έναν βολικό και αποτελεσματικό τρόπο χειρισμού της επικοινωνίας δικτύου στην εφαρμογή μας.

icalendar_parser (^2.0.0):

Το πακέτο `icalendar_parser` (Icalendar_Parser | Dart Package, 2023) διευκολύνει την ανάλυση των δεδομένων iCalendar, επιτρέποντας στην εφαρμογή μας να χειρίζεται και να διαχειρίζεται αποτελεσματικά τα συμβάντα iCal.

sqflite (^2.3.0):

Για την τοπική αποθήκευση δεδομένων, το πακέτο `sqflite` (Sqflite | Flutter Package, 2023) ενσωματώνεται στο έργο μας. Αυτό το plugin Flutter για το SQLite διευκολύνει την εφαρμογή τοπικών βάσεων δεδομένων, εξασφαλίζοντας αποτελεσματική και ασφαλή αποθήκευση δεδομένων εφαρμογών.

1.2.3 Ενσωμάτωση Firebase

Η ενσωμάτωση της Firebase είναι απαραίτητη για τις υπηρεσίες backend της εφαρμογής μας και χρησιμοποιούνται τα ακόλουθα πακέτα που σχετίζονται με την Firebase:

firebase_core (^2.24.0):

Το πακέτο `firebase_core` (Firebase_Core | Flutter Package, 2023) παρέχει τη βασική λειτουργικότητα για την ενσωμάτωση Firebase, αρχικοποιώντας τις υπηρεσίες Firebase στην εφαρμογή μας.

firebase_auth (^4.15.0):

Το πακέτο `firebase_auth` (Firebase_Auth | Flutter Package, 2023) χρησιμοποιείται για τον έλεγχο ταυτότητας χρήστη, επιτρέποντας στην εφαρμογή μας να διαχειρίζεται με ασφάλεια τις ταυτότητες των χρηστών.

cloud_firestore (^4.13.3):

Για αποθήκευση βάσης δεδομένων σε πραγματικό χρόνο, ενσωματώνεται το πακέτο `cloud_firestore` (Cloud_Firestore | Flutter Package, 2023). Επιτρέπει την απρόσκοπτη αλληλεπίδραση με το Firestore, τη βάση δεδομένων NoSQL που βασίζεται στο σύννεφο της Google.

firebase_storage (^11.5.6):

Το πακέτο `firebase_storage` (Firebase_Storage | Flutter Package, 2023) διευκολύνει την αποθήκευση αρχείων στο Firebase, επιτρέποντάς μας να διαχειριζόμαστε και να ανακτούμε αποτελεσματικά τα αποθηκευμένα αρχεία.

1.2.4 Βελτιώσεις διαχείρισης κατάστασης και διεπαφής χρήστη provider (^6.1.1):

Το πακέτο `provider` (Provider | Flutter Package, 2023) απλοποιεί τη διαχείριση της κατάστασης στην εφαρμογή Flutter. Προσφέρει μια βολική και απλή λύση για τη διαχείριση και την πρόσβαση στην κατάσταση εφαρμογής, ενισχύοντας τη συνολική συντηρησιμότητα και οργάνωση του κώδικα.

flutter_spinkit (^5.2.0):

Το πακέτο `flutter_spinkit` (Flutter_Spinkit | Flutter Package, 2023) χρησιμοποιείται για να βελτιώσει την εμπειρία του χρήστη κατά τη διάρκεια των διαδικασιών φόρτωσης. Παρέχει μια συλλογή κινούμενων δεικτών φόρτωσης, συμβάλλοντας σε μια οπτικά ελκυστική και ανταποκρινόμενη διεπαφή.

flutter_otp_text_field (^1.1.1):

Για Ασφαλείς και αποτελεσματικές διαδικασίες ελέγχου ταυτότητας χρήστη, το πακέτο `flutter_otp_text_field` (Flutter_Otp_Text_Field | Flutter Package, 2022) είναι ενσωματωμένο. Βελτιστοποιεί την εφαρμογή των πεδίων κειμένου OTP στην εφαρμογή μας.

1.2.5 Πλοήγηση και χειρισμός εικόνας get (^4.6.6):

Το πακέτο `get` (Get | Flutter Package, 2023) παρέχει μια βολική λύση για την πλοήγηση και την έγχυση εξάρτησης στο πλαίσιο του έργου Flutter. Απλοποιεί τις διαδικασίες πλοήγησης και ενισχύει τη συνολική δομή της εφαρμογής μας.

image_picker (^1.0.4):

Το πακέτο `image_picker` (Image_Picker | Flutter Package, 2023) απλοποιεί τη διαδικασία επιλογής και λήψης εικόνων στην εφαρμογή μας. Ενισχύει τη διαδραστικότητα των χρηστών και διευκολύνει τη δημιουργία περιεχομένου παρέχοντας έναν απρόσκοπτο μηχανισμό επιλογής εικόνας.

carousel_slide (^4.0.0):

Το πακέτο `carousel_slider` (Carousel_Slider | Flutter Package, 2021) στο Flutter παρέχει μια βολική λύση για την εφαρμογή καρουζέλ ή ρυθμιστικών εικόνων σε κινητές εφαρμογές.

Με την ενσωμάτωση του πακέτου, οι προγραμματιστές μπορούν εύκολα να δημιουργήσουν οπτικά ελκυστικές διεπαφές που παρουσιάζουν μια σειρά αντικειμένων ή εικόνων με απρόσκοπτο και διαδραστικό τρόπο.

1.2.6 Διαχείριση αρχείων και βελτιστοποίηση εικόνας path_provider (^2.1.1), path (^1.8.3):

Ο συνδυασμός των πακέτων path_provider (Path_Provider | Flutter Package, 2023) και path (Path | Dart Package, 2022) βοηθά στην απόκτηση της τοπικής διαδρομής καταλόγου και στο χειρισμό των διαδρομών αρχείων. Αυτά τα πακέτα διαδραματίζουν καθοριστικό ρόλο στην αποτελεσματική διαχείριση αρχείων εντός της εφαρμογής μας.

cached_network_image (^3.3.1):

Για τη βελτιστοποίηση της φόρτωσης και της προσωρινής αποθήκευσης εικόνων από διευθύνσεις URL δικτύου, χρησιμοποιείται το πακέτο cached_network_image (Cached_Network_Image | Flutter Package, 2023). Εξασφαλίζει μια ομαλή και αποτελεσματική εμπειρία προβολής εικόνων για τους χρήστες μας.

1.3 Ασφάλεια

1.3.1 SHA certificate fingerprints

Στον τομέα της ανάπτυξης εφαρμογών Android, τα δακτυλικά αποτυπώματα πιστοποιητικών SHA (Secure Hash Algorithm, (Bhattacharya, 2023)) αναδύονται ως σύνδεσμος για την εξασφάλιση της ασφαλούς επικοινωνίας μεταξύ εφαρμογών και εξωτερικών υπηρεσιών, όπως το Firebase. Αυτά τα δακτυλικά αποτυπώματα, ουσιαστικά κρυπτογραφικές τιμές κατακερματισμού που προέρχονται από ψηφιακά πιστοποιητικά, εξυπηρετούν έναν βαθύ σκοπό στην επαλήθευση της νομιμότητας, της αυθεντικότητας και της ακεραιότητας των εφαρμογών Android. Τα ψηφιακά πιστοποιητικά, που λειτουργούν ως κρυπτογραφικά κλειδιά, διαδραματίζουν καθοριστικό ρόλο στην ασφάλεια των εφαρμογών και συχνά αποθηκεύονται με ασφάλεια στο Android Keystore, έναν προστατευμένο θύλακα για κρυπτογραφικά περιουσιακά στοιχεία.

Οι κρυπτογραφικοί αλγόριθμοι που χρησιμοποιούνται για τη δημιουργία αυτών των δακτυλικών αποτυπωμάτων διαδραματίζουν καθοριστικό ρόλο στον προσδιορισμό της ευρωστίας του πλαισίου ασφαλείας. Ιστορικά, ο αλγόριθμος SHA-1 χρησιμοποιήθηκε ευρέως. Ωστόσο, λόγω αναγνωρισμένων τρωτών σημείων, η βιομηχανία έχει μετατοπίσει την πίστη της προς τον πιο ανθεκτικό αλγόριθμο SHA-256. Αυτή η μετατόπιση οφείλεται στην ανάγκη ενίσχυσης της στάσης ασφαλείας ολόκληρου του συστήματος, καθώς το SHA-256 παράγει Μεγαλύτερες τιμές κατακερματισμού, προσφέροντας αυξημένη αντίσταση έναντι πιθανών κρυπτογραφικών τρωτών σημείων που σχετίζονται με ασθενέστερους αλγόριθμους.

Πρακτικά, στο VS Code, οι προγραμματιστές μπορούν να εξαγάγουν δακτυλικά αποτυπώματα πιστοποιητικού SHA κατά τη διαδικασία υπογραφής της εφαρμογής. Αυτό το διαδικαστικό βήμα είναι καθοριστικό, ειδικά κατά τη διαμόρφωση του ελέγχου ταυτότητας με εξωτερικές υπηρεσίες. Η εντολή keytool, που ασκείται μέσα στο τερματικό του υπολογιστή,

```
keytool -list -v -keystore keystore.jks -alias keyAlias -storetype PKCS12 -sigalg  
SHA256withRSA -keyalg RSA
```

γίνεται το εργαλείο του τεχνίτη για τη δημιουργία αυτών των δακτυλικών αποτυπωμάτων. Ανάλογα με τις προϋποθέσεις ασφαλείας της εξωτερικής υπηρεσίας, οι προγραμματιστές μπορούν να δημιουργήσουν είτε δακτυλικά αποτυπώματα SHA-1 είτε SHA-256, αποδεικνύοντας μια λεπτή προσέγγιση στην κρυπτογραφική ασφάλεια.

Μια εξέχουσα αρένα όπου τα δακτυλικά αποτυπώματα των πιστοποιητικών SHA μπαίνουν σε εξέχοντα παιχνίδι είναι στην εξακρίβωση ταυτότητας Firebase. Κατά την ενσωμάτωση μιας εφαρμογής Android με τις υπηρεσίες Firebase, οι προγραμματιστές καταχωρούν σχολαστικά τα δακτυλικά αποτυπώματα της εφαρμογής στην κονσόλα Firebase. Αυτή η σχολαστική διαδικασία εγγραφής εξυπηρετεί έναν διπλό σκοπό: όχι μόνο επιβεβαιώνει την ταυτότητα της εφαρμογής, αλλά λειτουργεί και ως φύλακας, διασφαλίζοντας ότι μόνο οι εφαρμογές που φέρουν τα επικυρωμένα δακτυλικά αποτυπώματα επιτρέπεται να αλληλεπιδρούν με τις υπηρεσίες Firebase. Τα διακριτικά ελέγχου ταυτότητας Firebase, που προέρχονται από αυτά τα δακτυλικά αποτυπώματα, λειτουργούν ως ψηφιακά περάσματα, ενορχηστρώνοντας ασφαλή κανάλια επικοινωνίας μεταξύ της εφαρμογής και της Firebase.

Ο πρωταρχικός λόγος ύπαρξης για την ενσωμάτωση δακτυλικών αποτυπωμάτων πιστοποιητικών SHA στο τοπίο ανάπτυξης εφαρμογών Android έγκειται στην επιτακτική ανάγκη διασφάλισης αλληλεπιδράσεων μεταξύ εφαρμογών και εξωτερικών υπηρεσιών. Με την εντολή της επαλήθευσης της ταυτότητας της εφαρμογής μέσω αυτών των δακτυλικών αποτυπωμάτων, οι εξωτερικές υπηρεσίες μπορούν να διασφαλίσουν με βεβαιότητα ότι μόνο οι εξουσιοδοτημένες και νόμιμες εφαρμογές αποκτούν πρόσβαση στους φυλασσόμενους πόρους τους. Αυτός ο μηχανισμός ελέγχου ταυτότητας δεν είναι απλώς ένα εξάρτημα; αντίθετα, αποτελεί τον προμαχώνα κατά της μη εξουσιοδοτημένης πρόσβασης, υπερασπίζοντας την αιτία της προστασίας των δεδομένων των χρηστών και διατηρώντας την γενική ακεραιότητα του οικοσυστήματος εφαρμογών.

Συνοψίζοντας, τα δακτυλικά αποτυπώματα πιστοποιητικών SHA αναδύονται ως απαραίτητοι φύλακες της ασφάλειας στην ανάπτυξη εφαρμογών Android. Ως μοναδικά αναγνωριστικά φτιαγμένα από ψηφιακά πιστοποιητικά, αποτελούν την επιτομή της ισχυρής επαλήθευσης της ακεραιότητας και της αυθεντικότητας της εφαρμογής στο εκτεταμένο τοπίο των ψηφιακών αλληλεπιδράσεων. Η σχολαστική επιλογή κρυπτογραφικών αλγορίθμων, σε συνδυασμό με την έμπειρη εφαρμογή τους σε υπηρεσίες όπως το Firebase Authentication, αναδεικνύει τη δέσμευση για τη δημιουργία ασφαλών καναλιών επικοινωνίας που ενισχύουν την εμπιστοσύνη και την αξιοπιστία στο οικοσύστημα εφαρμογών Android. Έτσι, τα δακτυλικά αποτυπώματα πιστοποιητικών SHA δεν είναι απλώς τεχνικά αντικείμενα αλλά ως φρουροί ενός ασφαλούς και αξιόπιστου περιβάλλοντος για εφαρμογές Android.

Κεφάλαιο 2: Υλοποίηση

Αυτή η ενότητα παρέχει μια εις βάθος επισκόπηση των τεχνικών πτυχών και των βημάτων που ελήφθησαν κατά την ανάπτυξη της εφαρμογής για κινητά.

2.1. Τεχνικά χαρακτηριστικά

Η φάση υλοποίησης ξεκίνησε με τη διαμόρφωση του περιβάλλοντος ανάπτυξης και την επιλογή των κατάλληλων τεχνολογιών. Το πλαίσιο Flutter επιλέχθηκε για τις δυνατότητες πολλαπλών πλατφορμών, επιτρέποντάς μας να δημιουργήσουμε μια ενιαία βάση κώδικα τόσο για πλατφόρμες Android όσο και για iOS. Το Firebase χρησίμευσε ως λύση backend, προσφέροντας λειτουργικότητα βάσης δεδομένων σε πραγματικό χρόνο για την αποθήκευση προφίλ χρηστών και συνδέσμων iCal.

Κατά την εκκίνηση της εφαρμογής, αρχικοποιείται η σύνδεση της με το firebase. Αν υπάρχει ήδη εγγεγραμμένος χρήστης, σε αυτή την συσκευή, τον συνδέει αυτόματα αλλιώς τον βγάζει στην οθόνη σύνδεσης (Sign in, Εικόνα 2), αν ο χρήστης τώρα δεν έχει λογαριασμό καλείται να πλοηγηθεί στην οθόνη εγγραφής (Register, Εικόνα 3).

Εκεί ξεκινάει το αρχικό σημείο αλληλεπίδρασης για τους χρήστες στην εφαρμογή, που είναι η διαδικασία εγγραφής, ένα κρίσιμο βήμα που δημιουργεί τα θεμέλια για μια εξατομικευμένη και ασφαλή εμπειρία χρήστη. Κατά την εγγραφή, οι χρήστες καλούνται να παρέχουν βασικές πληροφορίες, όπως το όνομά τους, την διεύθυνση ηλεκτρονικού ταχυδρομείου τους (email), τον αριθμό του τηλεφώνου τους, την διεύθυνση τους καθώς και τον κωδικό πρόσβασης τους, όλες αυτές οι λεπτομέρειες ορίζονται από το χρήστη. Είναι σημαντικό να αναφερθεί ότι υπάρχουν αυστηρές προϋποθέσεις τόσο για τις εισόδους ηλεκτρονικού ταχυδρομείου όσο και για τον κωδικό πρόσβασης για την ενίσχυση της ασφάλειας και της ακεραιότητας των δεδομένων των χρηστών.

Για την επαλήθευση μέσω ηλεκτρονικού ταχυδρομείου, η εφαρμογή ελέγχει ότι το καταχωρημένο μήνυμα ηλεκτρονικού ταχυδρομείου συμμορφώνεται με έγκυρη μορφή, διασφαλίζοντας ότι περιλαμβάνει το σύμβολο "@" και ακολουθεί μια τυπική δομή ηλεκτρονικού ταχυδρομείου, σε περίπτωση που δεν είναι σωστή η διεύθυνση ηλεκτρονικού ταχυδρομείου εμφανίζεται το μήνυμα «Please supply a valid email». Αυτό το βήμα επαλήθευσης είναι ζωτικής σημασίας όχι μόνο για την ακριβή αναγνώριση των χρηστών αλλά και για τη δημιουργία αξιόπιστων μέσων επικοινωνίας εντός της εφαρμογής.

Οι προϋποθέσεις για τον κωδικό πρόσβασης έχουν σχεδιαστεί για να δημιουργούν ισχυρά και ασφαλή διαπιστευτήρια για λογαριασμούς χρηστών. Οι χρήστες καλούνται να δημιουργήσουν κωδικούς πρόσβασης που πληρούν συγκεκριμένα κριτήρια, όπως συμπεριλαμβανομένου ενός συνδυασμού κεφαλαίων και πεζών γραμμάτων, αριθμών και ειδικών χαρακτήρων, σε περίπτωση που δεν πληρούνται οι προϋποθέσεις εμφανίζεται το αντίστοιχο μήνυμα στον χρήστη. Αυτή η προσέγγιση ενισχύει τη συνολική στάση ασφαλείας των λογαριασμών χρηστών, μετριάζοντας τον κίνδυνο μη εξουσιοδοτημένης πρόσβασης και παραβιάσεων δεδομένων.

```

TextFormField(
  decoration: textInputDecoration.copyWith(
    hintText: 'Email',
    hintStyle: TextStyle(color: Colors(0xFF1e2d4c)),
    prefixIcon: Icon(Icons.person_outline_outlined,
      color: Colors(0xFF1e2d4c)), // Icon
  ),
  validator: (val) =>
    val!.isEmpty ? 'Enter an email' : null,
  onChanged: (val) {
    setState(() => email = val);
  },
), // TextFormField
 SizedBox(
  height: 20,
), // SizedBox
TextFormField(
  decoration: textInputDecoration.copyWith(
    hintText: 'Password',
    hintStyle: TextStyle(color: Colors(0xFF1e2d4c)),
    prefixIcon: Icon(Icons.fingerprint,
      color: Colors(0xFF1e2d4c)), // Icon
  ),
  validator: (val) => val!.length < 6
    ? 'Enter a password 6+ characters long'
    : null,
  obscureText: true,
  onChanged: (val) {
    setState(() => password = val);
  },
), // TextFormField
SizedBox(
  height: 20,
), // SizedBox
Align(
  alignment: Alignment.centerRight,
  child: TextButton(
    onPressed: () {
      ForgetPasswordScreen
        .buildshowmodalbottomsheet(context);
    },
    child: Text('Forget Password?',
      style:
        TextStyle(color: Colors(0xFFacbdad))), // Text
  ), // Align
SizedBox(
  height: 20,
), // SizedBox
ElevatedButton(
  onPressed: () async {
    if (_formkey.currentState!.validate());
    setState(() {
      loading = true;
    });
    dynamic result = await _auth
      .signInWithEmailAndPassword(email, password);
    if (result == null) {
      setState(() {
        error =
          'Could not sign in with those credentials';
        loading = false;
      });
    }
  },
  child: Text(
    'Sign in',
    style: TextStyle(color: Colors.white),
  ), // Text
  style: ButtonStyle(
    backgroundColor:
      MaterialStateProperty.all<Color>(
        Colors(0xFFA3AB92)), // ButtonStyle
  ), // ElevatedButton
SizedBox(
  height: 12,
), // SizedBox
Text(
  error,
  style: TextStyle(color: Colors.red, fontSize: 14),
) // Text

```

Κώδικας 1: Sign in Code

```

TextFormField(
  controller: controller.name,
  decoration: textInputDecoration.copyWith(
    hintText: 'Name',
    hintStyle: TextStyle(color: Colors(0xFF1e2d4c)),
    prefixIcon: Icon(Icons.person_outline_rounded,
      color: Colors(0xFF1e2d4c)), // Icon
  ),
  validator: (val) =>
    val!.isEmpty ? 'Enter a name' : null,
  onChanged: (val) {
    setState(() => name = val);
  },
), // TextFormField
SizedBox(
  height: 20,
), // SizedBox
TextFormField(
  controller: controller.email,
  decoration: textInputDecoration.copyWith(
    hintText: 'Email',
    hintStyle: TextStyle(color: Colors(0xFF1e2d4c)),
    prefixIcon: Icon(Icons.email_outlined,
      color: Colors(0xFF1e2d4c)), // Icon
  ),
  validator: (val) =>
    val!.isEmpty ? 'Enter an email' : null,
  onChanged: (val) {
    setState(() => email = val);
  },
), // TextFormField
SizedBox(
  height: 20,
), // SizedBox
TextFormField(
  controller: controller.phoneNo,
  keyboardType: TextInputType.number,
  decoration: textInputDecoration.copyWith(
    hintText: 'Phone number',
    hintStyle: TextStyle(color: Colors(0xFF1e2d4c)),
    prefixIcon: Icon(Icons.numbers,
      color: Colors(0xFF1e2d4c)), // Icon
  ),
  validator: (val) => val!.length < 10
    ? 'Enter a valid phone number'
    : null,
  onChanged: (val) {
    setState(() => phone = val);
  },
), // TextFormField
SizedBox(
  height: 20,
), // SizedBox
TextFormField(
  controller: controller.password,
  decoration: textInputDecoration.copyWith(
    hintText: 'Password',
    hintStyle: TextStyle(color: Colors(0xFF1e2d4c)),
    prefixIcon: Icon(Icons.fingerprint,
      color: Colors(0xFF1e2d4c)), // Icon
  ),
  validator: (val) => val!.length < 6
    ? 'Enter a password 6+ characters long'
    : null,
  obscureText: true,
  onChanged: (val) {
    setState(() => password = val);
  },
), // TextFormField
SizedBox(
  height: 20,
), // SizedBox
TextFormField(
  controller: controller.address,
  decoration: textInputDecoration.copyWith(
    hintText: 'address',
    hintStyle:
      TextStyle(color: Colors(0xFF1e2d4c)),
    prefixIcon: Icon(Icons.location_on_outlined,
      color: Colors(0xFF1e2d4c)), // Icon
  ),
  onChanged: (val) {
    setState(() => address = val);
  }, // TextFormField
SizedBox(
  height: 20,
), // SizedBox
ElevatedButton(
  onPressed: () async {
    if (_formkey.currentState!.validate());
    setState(() {
      loading = true;
    });
    dynamic result =
      await _auth.registerWithEmailAndPassword(
        email,
        password,
        name,
        phone,
        imageUrl,
        address);
    if (result == null) {
      setState(() {
        loading = false;
        error = 'Please supply a valid email';
      });
    }
  },
  child: Text(
    'Register',
    style: TextStyle(color: Colors.white),
  ), // Text
  style: ButtonStyle(
    backgroundColor:
      MaterialStateProperty.all<Color>(
        Colors(0xFFA3AB92)), // ButtonStyle
  ), // ElevatedButton
SizedBox(
  height: 12,
), // SizedBox
Text(
  error,
  style: TextStyle(color: Colors.red, fontSize: 14),
) // Text

```

Κώδικας 2: Register Code

Σε περίπτωση που ο χρήστης δεν θυμάται τον κωδικό πρόσβασης του μπορεί να τον συνδεθεί στον λογαριασμό του με χρήση OTP (One Time password) είτε μέσω τηλεφώνου, είτε μέσω email και να αλλάξει τον κωδικό του κατά την σύνδεση.

Για την επίτευξη του ελέγχου της σύνδεσης και της εγγραφής βλέπε κώδικες 1 και 2.

Μόλις ένας Χρήστης πλοηγηθεί με επιτυχία στη διαδικασία εγγραφής και πληροί αυτές τις αυστηρές απαιτήσεις, η εφαρμογή στέλνει τα δεδομένα προφίλ του και αποθηκεύονται με ασφάλεια στο Firebase Firestore Database. Το Firebase Firestore Database λειτουργεί ως βάση δεδομένων NoSQL, που χρησιμεύει ως κεντρικό αποθετήριο για δεδομένα χρήστη. Κατά την διαδικασία αυτή δημιουργείται μια συλλογή(μόνο την πρώτη φορά) και μέσα της θα περιέχει κάθε έγγραφο (document) που ως όνομα λαμβάνει το μοναδικό αναγνωριστικό του χρήστη uid (το οποίο δημιουργείται αυτόματα και είναι τέτοιας μορφής ‘tUkYG0SLLZO8GrQg8UMWWAOpRDi2’, έτσι ώστε να μην μπορεί κανείς να τον υποκλέψει) και μέσα σε αυτό το έγγραφο δημιουργούνται τα πεδία στα οποία αποθηκεύονται τα στοιχεία του χρήστη (Κώδικας 4 και 5). Αυτό περιλαμβάνει όχι μόνο βασικές πληροφορίες προφίλ, αλλά και τυχόν συνδέσμους iCal που σχετίζονται με κρατήσεις ή εκδηλώσεις που εισάγουν οι χρήστες αργότερα. Η σχολαστική αποθήκευση των δεδομένων χρήστη στο Firebase Firestore εξασφαλίζει τη συνέπεια των δεδομένων και την προσβασιμότητα σε όλες τις συσκευές, διευκολύνοντας μια απρόσκοπτη και συγχρονισμένη εμπειρία.

Η σύνδεση με το Firebase έχει γίνει εφικτή με το API της google play services και με την ενσωμάτωση κάποιων χαρακτηριστικών της εφαρμογής όπως το SHA certificate fingerprints στο Firebase καθώς και το API Key από το Firebase στη εφαρμογή.

Προχωρώντας στο ταξίδι του χρήστη, οι χρήστες μπορούν να εμπλουτίσουν την εμπειρία του ημερολογίου τους εισάγοντας iCal συνδέσμους που σχετίζονται με τις κρατήσεις ή τις εκδηλώσεις τους. Αυτοί οι σύνδεσμοι iCal, μόλις εισαχθούν μέσω της διεπαφής της εφαρμογής, αποθηκεύονται απρόσκοπτα στο Firebase Firestore (Κώδικας 7). Η αλληλεπίδραση σε πραγματικό χρόνο μεταξύ της εφαρμογής και του Firebase Firestore διασφαλίζει ότι τυχόν ενημερώσεις ή τροποποιήσεις στους συνδέσμους iCal αντικατοπτρίζονται αμέσως στην προβολή ημερολογίου της εφαρμογής.

Αυτός ο δυναμικός συγχρονισμός συμβάλλει σε μια ενημερωμένη και ανταποκρινόμενη εμπειρία ημερολογίου, οι χρήστες μπορούν να δουν τις κρατήσεις τους τόσο στο ημερολόγιο όσο και μαζεμένες όλες μαζί ή ανά κατάλυμα.

Η αλληλεπίδραση της εφαρμογής με το Firebase Firestore εκτείνεται πέρα από τα συμβάντα ημερολογίου για να συμπεριλάβει προφίλ χρηστών. Οι χρήστες έχουν την αυτονομία να επεξεργάζονται τα προφίλ τους μέσα στην εφαρμογή, ενημερώνοντας πληροφορίες όπως το όνομά τους, την εικόνα προφίλ ή οποιεσδήποτε άλλες προσαρμοσμένες λεπτομέρειες. Συγκεκριμένα, τυχόν αλλαγές που έγιναν στο προφίλ, συμπεριλαμβανομένων των ενημερώσεων κωδικού πρόσβασης, κοινοποιούνται αμέσως στο Firebase Firestore. Αυτό επιτυγχάνεται καθώς από την είσοδο του ο χρήστης “κουβαλάει” το μοναδικό αναγνωριστικό του και η εφαρμογή διαβάζοντας το επικοινωνεί με το Firebase και κάνει τις αλλαγές που πρέπει. Αυτός ο συγχρονισμός σε πραγματικό χρόνο διασφαλίζει ότι οι ενημερωμένες πληροφορίες αντικατοπτρίζονται αμέσως τόσο στο UI της εφαρμογής όσο και στο backend της Firebase, προωθώντας μια συνεπή και ασφαλή εμπειρία χρήστη.

Όσον αφορά την αλλαγή την εικόνα προφίλ, η διαδικασία ξεκινά με την συνάρτηση `_uploadimage` (Κώδικας 3), με τη χρήση του πακέτου `ImagePicker` για να ζητήσει από τον χρήστη να επιλέξει μια εικόνα από τη συλλογή της συσκευής του. Το επιλεγμένο αρχείο υποβάλλεται στη συνέχεια σε μια σειρά ελέγχων για να διασφαλιστεί ότι είναι μια έγκυρη εικόνα με αποδεκτές επεκτάσεις όπως `"jpg"`, `"jpeg"` ή `"png"`.

Μετά την επιβεβαίωση της εγκυρότητας του αρχείου, μετατρέπεται σε αντικείμενο αρχείου και επαληθεύεται το αναγνωριστικό του τρέχοντος χρήστη. Η αναφορά αποθήκευσης `Firebase (storageRef)` κατασκευάζεται, υποδεικνύοντας τη διαδρομή με το αναγνωριστικό του χρήστη και την επέκταση αρχείου. Στη συνέχεια, το επιλεγμένο αρχείο εικόνας μεταφορτώνεται στο `Firebase Storage` χρησιμοποιώντας τη μέθοδο `putFile`. Στη συνέχεια, παίρνουμε τη διεύθυνση `URL` της μεταφορτωμένης εικόνας για περαιτέρω χρήση.

Για να διατηρήσουμε συγχρονισμένα τα δεδομένα του `Firestore`, ενημερώνουμε το αντίστοιχο έγγραφο στη συλλογή `"testdata"`, τροποποιώντας συγκεκριμένα το πεδίο `"imageurl"` με τη νέα διεύθυνση `URL` λήψης. Ταυτόχρονα, διασφαλίζουμε ότι η τοπική κατάσταση της διεύθυνσης `URL` εικόνας του χρήστη μέσα στην εφαρμογή ενημερώνεται χρησιμοποιώντας τη διαχείριση κατάστασης του παρόχου. Αυτό διασφαλίζει ότι τυχόν στοιχεία διεπαφής χρήστη που αντικατοπτρίζουν το προφίλ του χρήστη, ειδικά την εικόνα προφίλ του, ενημερώνονται αμέσως. Για την δομή των δεδομένων φωτογραφίας δείτε στο παράρτημα Α.

Δεδομένης της ασύγχρονης φύσης αυτών των λειτουργιών, ο κώδικας περιλαμβάνει χειρισμό σφαλμάτων για τη χαριτωμένη διαχείριση τυχόν προβλημάτων που αντιμετωπίζουν. Σε περίπτωση σφάλματος, ενεργοποιείται μια διακριτική αλλά ενημερωτική γραμμή λάθους για να ειδοποιήσει τον χρήστη, ενισχύοντας τη συνολική εμπειρία του χρήστη.

Πίσω από τα παρασκήνια, ο έλεγχος ταυτότητας `Firebase` παίζει καθοριστικό ρόλο στην ασφαλή διαχείριση της εγγραφής χρηστών, της σύνδεσης και των δραστηριοτήτων που σχετίζονται με το λογαριασμό. Τα διακριτικά ελέγχου ταυτότητας που παρέχονται από την `Firebase` χρησιμοποιούνται για την εξουσιοδότηση και την ασφαλή επικοινωνία μεταξύ της εφαρμογής και της `Firebase`. Αυτό το πρόσθετο επίπεδο ασφάλειας ενισχύει τις αλληλεπιδράσεις των χρηστών και διασφαλίζει ότι μόνο οι πιστοποιημένοι χρήστες μπορούν να έχουν πρόσβαση και να τροποποιούν τις πληροφορίες του λογαριασμού τους.

```

Future<void> _uploadImage(BuildContext context) async {
  try {
    final picker = ImagePicker();
    final pickedFile = await picker.pickImage(source: ImageSource.gallery);

    if (pickedFile != null) {
      final imageFile = File(pickedFile.path);

      // Check if the picked file is an image
      final allowedExtensions = ['jpg', 'jpeg', 'png'];
      final fileExtension = pickedFile.path.split('.').last.toLowerCase();

      if (!allowedExtensions.contains(fileExtension)) {
        throw Exception('Invalid file format. Please choose an image.');

```

Κώδικας 3: _uploadimage Function

Συμπερασματικά, ο περίπλοκος σχεδιασμός της εφαρμογής μας εναρμονίζει τις αλληλεπιδράσεις των χρηστών, το frontend της εφαρμογής και το Firebase ως υποδομή backend. Η σχολαστική εφαρμογή των απαιτήσεων εγγραφής, δυναμική απεικόνιση ημερολογίου, και συγχρονισμένη Επεξεργασία προφίλ αποτελεί παράδειγμα της εξελιγμένης αρχιτεκτονικής ροής πληροφοριών που στηρίζει τη λειτουργικότητα της εφαρμογής μας. Αυτή η προσέγγιση όχι μόνο διασφαλίζει μια ασφαλή βάση δεδομένων χρηστών, αλλά και ενισχύει τα θεμέλια για μια ανταποκρινόμενη, εξατομικευμένη και πλούσια σε χαρακτηριστικά εμπειρία ημερολογίου για τους χρήστες μας.

2.2. Διαχείριση Λογαριασμού Χρήστη

Η διαδικασία δημιουργίας λογαριασμών χρηστών υλοποιήθηκε χρησιμοποιώντας έλεγχο ταυτότητας Firebase. Οι χρήστες θα μπορούσαν να εγγραφούν με email και κωδικό πρόσβασης, με τα διαπιστευτήριά τους να αποθηκεύονται με ασφάλεια στο σύστημα ελέγχου ταυτότητας Firebase. Οι πληροφορίες λογαριασμού συνδέθηκαν με ένα αντίστοιχο προφίλ χρήστη στη βάση δεδομένων Firebase Realtime. Εντός της εφαρμογής ο χρήστης μπορεί να επεξεργαστεί το προφίλ του αλλάζοντας της πληροφορίες τους.

```
Future registerWithEmailAndPassword(String email, String password,
    String name, String phone, String imageUrl, String address) async {
  try {
    UserCredential result = await _auth.createUserWithEmailAndPassword(
      email: email,
      password: password,
    );
    User user = result.user!;

    // Create a new document for the user with the uid
    await DatabaseService(uid: user.uid)
      .updateUserData(name, phone, email, password, imageUrl, address)
      .whenComplete(
        () => Get.snackbar('Success', 'Your account has been created',
          snackPosition: SnackPosition.BOTTOM,
          backgroundColor: Colors.green.withOpacity(0.1),
          colorText: Colors.green),
      );
    return _userFromFirebaseUser(user);
    // ignore: unused_catch_stack
  } catch (error, StackTrace) {
    Get.snackbar('Error', 'Something went wrong. Try again',
      snackPosition: SnackPosition.BOTTOM,
      backgroundColor: Colors.redAccent,
      colorText: Colors.red);
    print(
      error.toString(),
    );
  }
}
```

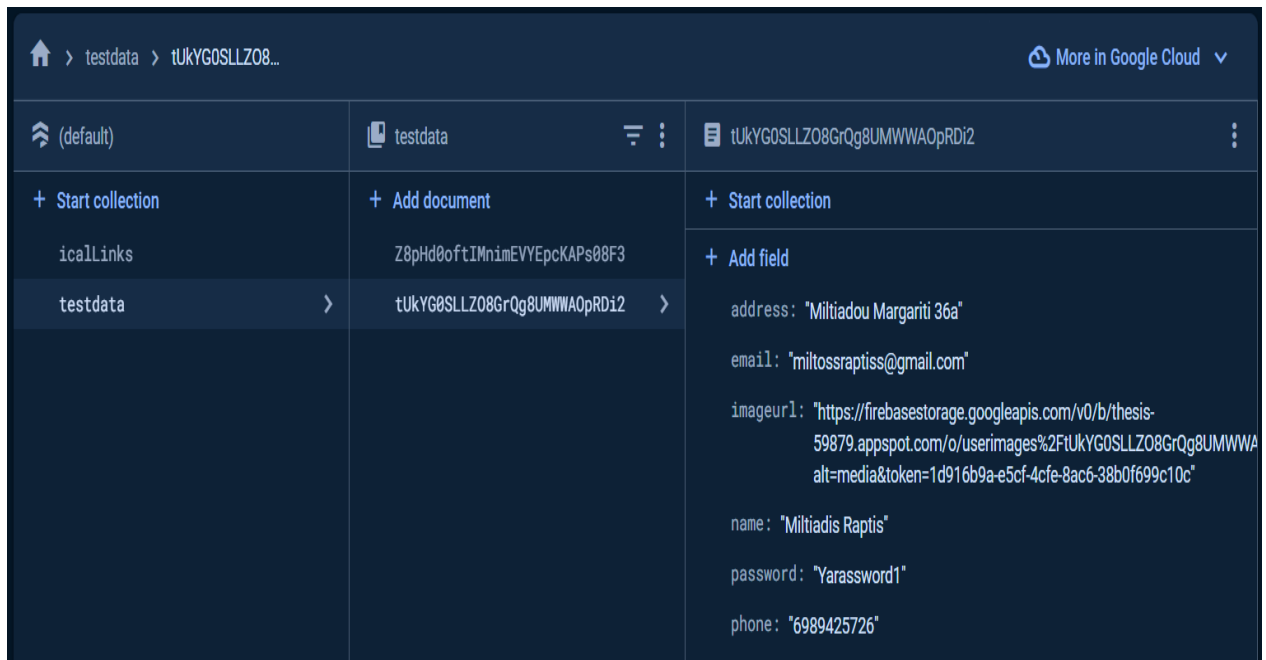
Κώδικας 4: Εγγραφή με Email και Password

Πιο συγκεκριμένα η συνάρτηση register WithEmailAndPassword έχει ως απαίτηση να λαμβάνει τα στοιχεία του χρήστη (email, password, name, phone, imageUrl και address), ελέγχει με την συνάρτηση createUserWithEmailAndPassword, του FirebaseAuth, αν υπάρχει χρήστης με τα συγκεκριμένα στοιχεία στο database. Σε περίπτωση που υπάρχει του εμφανίζει μήνυμα ότι υπάρχει χρήστης με τα ίδια στοιχεία, αλλιώς συνεχίζεται η διαδικασία και καλείται η συνάρτηση updateUserData, η οποία δημιουργεί ή διαβάζει την συλλογή που στην δικιά μας περίπτωση ονομάζεται 'testdata', για να δημιουργηθεί ο χρήστης.

```
final CollectionReference testCollection =  
    FirebaseFirestore.instance.collection('testdata');  
  
final CollectionReference icalCollection =  
    FirebaseFirestore.instance.collection('icalLinks');  
  
Future<void> updateUserData(  
    String name,  
    String phone,  
    String email,  
    String password,  
    String imageUrl,  
    String address,  
) async {  
    try {  
        await testCollection.doc(uid).set({  
            'name': name,  
            'phone': phone,  
            'email': email,  
            'password': password,  
            'imageUrl': imageUrl,  
            'address': address,  
        });  
    } catch (error) {  
        print('Error updating user data: $error');  
        throw error;  
    }  
}
```

Κώδικας 5: updateUserData Function

Η δομή των αρχείων με τα αποθηκευμένα στοιχεία στο Firebase, φαίνονται στην παρακάτω εικόνα (Εικόνα 1).



Εικόνα 1: Firebase Document Structure

Κατά την ολοκλήρωση της εγγραφής ο χρήστης καλείται να πραγματοποιήσει την σύνδεση του. Εισάγοντας το email και τον κωδικό του, η εφαρμογή ελέγχει για την ύπαρξη του χρήστη καλώντας την συνάρτηση `signInWithEmailAndPassword` (Κώδικας 6) όπως φαίνεται παρακάτω.

```
// Sign in with email and password
Future signInWithEmailAndPassword(String email, String password) async {
  try {
    UserCredential result = await _auth.signInWithEmailAndPassword(
      email: email, password: password);
    User user = result.user!;
    return user;
  } catch (error) {
    print(error.toString());
    return null;
  }
}
```

Κώδικας 6: `signInWithEmailAndPassword` Function

2.3. Αποθήκευση iCal Link

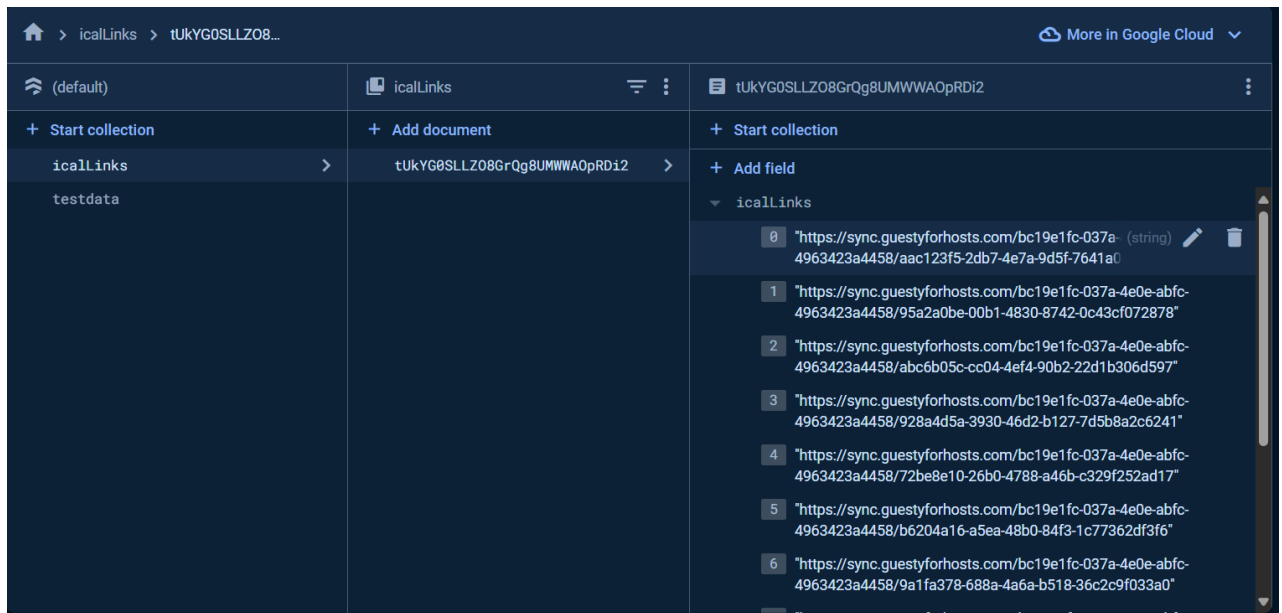
Για να μπορέσουν οι χρήστες να αποθηκεύουν συνδέσμους iCal, πρέπει να μεταβούν στην ρυθμίσεις της εφαρμογής για την εισαγωγή και την υποβολή συνδέσμων. Αυτοί οι σύνδεσμοι

στη συνέχεια αποθηκεύτηκαν με ασφάλεια στη βάση δεδομένων Firebase Realtime κάτω από το προφίλ του αντίστοιχου χρήστη.

```
ElevatedButton(  
  onPressed: () async {  
    // Save the entered iCal link to Firestore  
    String userId = AuthService().getCurrentUserId();  
  
    // Check if the document exists  
    bool documentExists = await _icalLinksCollection  
      .doc(userId)  
      .get()  
      .then((doc) => doc.exists);  
  
    if (!documentExists) {  
      // If the document doesn't exist, create a new one  
      await _icalLinksCollection  
        .doc(userId)  
        .set({'icalLinks': []});  
    }  
  
    // Update the document with the new iCal link  
    await _icalLinksCollection.doc(userId).update({  
      'icalLinks':  
        FieldValue.arrayUnion([_icalLinkController.text])  
    }).catchError((error) {  
      print('Failed to add iCal link: $error');  
    });  
  
    // Clear the text field after saving  
    _icalLinkController.clear();  
  },
```

Κώδικας 7 Save iCal Links Code

Στον κώδικα 7 φαίνεται η λειτουργία αποθήκευσης των συνδέσμων iCal στο Firebase. Αρχικά, όπως και στην δημιουργία των χρηστών, ελέγχετε αν υπάρχει η συλλογή ονόματι 'icalLinks', αν δεν υπάρχει δημιουργείτε, αλλιώς απλά δημιουργείτε ένα έγγραφο μέσα σε αυτήν την συλλογή στο οποίο αποθηκεύονται οι σύνδεσμοι σε μορφή πίνακα για να μπορούν να διαβαστούν ένας ένας, όπως φαίνεται στην παρακάτω εικόνα (Εικόνα 2).



Εικόνα 2: Firebase Document Structure iCal Links

2.4. Διάβασμα Συνδέσμων iCal

Σε αυτό το σημείο είναι σημαντικό να αναφερθεί ο τρόπος ανάκτησης των γεγονότων/κρατήσεων από τους συνδέσμους iCal που είναι αποθηκευμένοι στο Firebase. Αυτό είναι ένα από τα πιο απαιτητικά και δύσκολο κομμάτια της εφαρμογής, διότι υπήρχαν θέματα συγχρονίας. Υπάρχουν ορισμένες συναρτήσεις και λειτουργίες οι οποίες έπρεπε να γίνουν πριν από άλλες, για την αποφυγή σφαλμάτων, οπότε η σειρά με την οποία θα εντελλόντουσαν ήταν σημαντική.

Για αρχή διαβάζεται το έγγραφο της συλλογής icalLinks, το οποίο αντιστοιχεί με τον μοναδικό αναγνωριστικό του χρήστη το uid. Αυτό επιτυγχάνεται με την χρήση της συνάρτησης getUrls (Κώδικας 8), η οποία επικαλείται την getSavedIcalLinks (Κώδικας 9), και μας επιστρέφει τα Url των iCal στην μεταβλητή savedIcalLinks σε μορφή λίστας.

```
Future<List<String>> getUrls() async {
  List<String> savedIcalLinks =
    await _icalLinksService.getSavedIcalLinks(userId);
  return savedIcalLinks;
}
```

Κώδικας 8: getUrls Function

```
Future<List<String>> getSavedIcalLinks(String userId) async {
    DocumentSnapshot snapshot = await _icalLinksCollection.doc(userId).get();

    if (snapshot.exists) {
        final data = snapshot.data() as Map<String, dynamic>; // Handle null
        if (data != null) {
            List<String> icalLinks = List<String>.from(data['icalLinks'] ?? []);
            return icalLinks;
        }
    }

    return [];
}
```

Κώδικας 9: *getSavedIcalLinks Function*

Στην συνέχεια με την συνάρτηση `_getIcalDataList` (Κώδικας 10), ανακτώνται ασύγχρονα δεδομένα iCal από μια λίστα διευθύνσεων URL. Ο πρωταρχικός σκοπός του είναι να διευκολύνει την ταυτόχρονη εκτέλεση ασύγχρονων εργασιών, ειδικά την ανάκτηση δεδομένων iCal από πολλαπλές πηγές.

```
Future<List<String>> _getIcalDataList() async {
    List<String> urls = await getUrls();

    // Use Future.wait to execute requests concurrently
    List<Future<String>> futures = urls.map((url) => getIcalData(url)).toList();
    List<String> dataList = await Future.wait(futures);

    return dataList;
}
```

Κώδικας 10: *_getIcalDataList Function*

Κατά την έναρξη, η συνάρτηση ξεκινά την εκτέλεσή της αποκτώντας μια λίστα iCal URLs χρησιμοποιώντας τη συνάρτηση `getUrls`. Η χρήση της λέξης-κλειδιού `await` διασφαλίζει ότι η εκτέλεση της συνάρτησης σταματά έως ότου ανακτηθούν επιτυχώς οι διευθύνσεις URL iCal.

Μετά την απόκτηση των διευθύνσεων URL iCal, η συνάρτηση προχωρά στη δημιουργία μιας λίστας μελλοντικών αντικειμένων<String> που ονομάζονται futures. Κάθε στοιχείο future αντιστοιχεί σε μια ασύγχρονη εργασία που αντιπροσωπεύεται από τη συνάρτηση `getIcalData`, η οποία ανακτά δεδομένα iCal για μια δεδομένη διεύθυνση URL. Η συνάρτηση `map` χρησιμοποιείται για την εφαρμογή `getIcalData` σε κάθε διεύθυνση URL και όλα μετατρέπονται σε λίστα χρησιμοποιώντας το `toList`. Η συνάρτηση ολοκληρώνεται με την επιστροφή της λίστας<String> ονόματι `datalist`, ενσωματώνοντας τα δεδομένα iCal που ανακτώνται από πολλαπλές διευθύνσεις URL.

Η συνάρτηση `getIcalData` (Κώδικας 11) είναι μια ασύγχρονη συνάρτηση που έχει σχεδιαστεί για τη λήψη iCal δεδομένων από μια δεδομένη διεύθυνση URL. Η κύρια ευθύνη του είναι να εκτελέσει ένα αίτημα HTTP GET στην καθορισμένη διεύθυνση URL, να χειριστεί την απάντηση και να επεξεργαστεί τα δεδομένα iCal.

```

Future<String> getIcalData(String url) async {
  http.Response response = await http.get(Uri.parse(url));

  if (response.statusCode == 200) {
    var icsObj = ICalendar.fromString(response.body);
    String result = jsonEncode(icsObj.toJson());
    return result;
  } else {
    print("Request failed with status Code: ${response.statusCode}");
    throw Exception("Failed to fetch data");
  }
}

```

You, 2 months ago • Initial commit

Κώδικας 11: getIcalData Function

Με τη χρήση του πακέτου http γίνεται ένα αίτημα HTTP GET στην παρεχόμενη διεύθυνση URL. Η λέξη-κλειδί await χρησιμοποιείται για να περιμένει την ολοκλήρωση του ασύγχρονου http.get λειτουργία, διασφαλίζοντας ότι η λειτουργία δεν προχωρά μέχρι να εκπληρωθεί το αίτημα HTTP. Η απόκριση αποθηκεύεται σε μια μεταβλητή που ονομάζεται result.

Μετά τη λήψη της Απόκρισης HTTP, η λειτουργία ελέγχει τον κωδικό κατάστασης. Εάν ο κωδικός κατάστασης είναι 200 (HTTP OK), προχωρά στην επεξεργασία των δεδομένων iCal. Σε αυτή την περίπτωση, η κλάση ICalendar χρησιμοποιείται για την ανάλυση των δεδομένων iCal, το οποίο θεωρείται έγκυρη μορφή iCalendar. Τα αναλυμένα δεδομένα μετατρέπονται στη συνέχεια σε μορφή JSON χρησιμοποιώντας jsonEncode. Η προκύπτουσα συμβολοσειρά JSON αποθηκεύεται σε μια μεταβλητή που ονομάζεται result, η οποία στη συνέχεια επιστρέφεται από τη συνάρτηση.

Εάν ο κωδικός κατάστασης απόκρισης HTTP δεν είναι 200, υποδεικνύοντας μια αποτυχία, η λειτουργία εκτυπώνει ένα μήνυμα σφάλματος στην κονσόλα, συμπεριλαμβανομένου του κωδικού κατάστασης. Στη συνέχεια, ρίχνει μια εξαίρεση με το μήνυμα " Request failed with status Code".

Τέλος για την απεικόνιση αυτών των δεδομένων στην προβολή ημερολογίου και στην σελίδα με όλες τις κρατήσεις, χρειάζεται να περαστούν όλα τα δεδομένα σε μια λίστα που να μπορεί να διαβαστεί, αυτό επιτεύχθηκε με την χρήση των δύο παρακάτω συναρτήσεων.

Η συνάρτηση parseEvents (Κώδικας 12) εξυπηρετεί το σκοπό της ανάλυσης μιας συμβολοσειράς JSON που αντιπροσωπεύει μια λίστα συμβάντων και τη μετατροπή της σε μια πιο δομημένη μορφή. Η συνάρτηση παίρνει ένα jsonString ως παράμετρο της και χρησιμοποιεί τη μέθοδο jsonDecode για να μετατρέψει τη συμβολοσειρά JSON σε Map<String, dynamic>. Στη συνέχεια, ελέγχει εάν περιέχεται το κλειδί "data". Εάν υπάρχει το 'data', η συνάρτηση επιστρέφει μια νέα λίστα που δημιουργήθηκε από το αυτό, όπου κάθε στοιχείο αναμένεται να είναι Map<String, dynamic>. Σε περίπτωση που δεν υπάρχει το "data", που σημαίνει έλλειψη δεδομένων, η συνάρτηση επιστρέφει μια κενή λίστα.

```

List<Map<String, dynamic>> parseEvents(String jsonString) {
  Map<String, dynamic> jsonData = jsonDecode(jsonString);
  if (jsonData.containsKey('data')) {
    return List<Map<String, dynamic>>.from(jsonData['data']);
  }
  return [];
}

```

Κώδικας 12: parseEvents Function

Η λειτουργία getallevents (Κώδικας 13) είναι προσανατολισμένη προς τη συγκέντρωση συμβάντων από μια λίστα συμβολοσειρών JSON, παρέχοντας έναν βολικό τρόπο ενοποίησης δεδομένων συμβάντων από πολλαπλές πηγές. Η συνάρτηση παίρνει το jsonDataList ως είσοδο, όπου κάθε στοιχείο αντιπροσωπεύει μια συμβολοσειρά JSON, όπως αναφέρθηκε παραπάνω, που περιέχει μια λίστα γεγονότων. Αρχικοποιεί μια κενή λίστα με το όνομα allEvents, η οποία προορίζεται για την αποθήκευση των συγκεντρωτικών συμβάντων.

```

List<Map<String, dynamic>> getAllEvents(List<String> jsonDataList) {
  List<Map<String, dynamic>> allEvents = [];
  for (String jsonData in jsonDataList) {
    allEvents.addAll(parseEvents(jsonData));
  }
  return allEvents;
}

```

Κώδικας 13: getAllEvents Function

Στη συνέχεια, η συνάρτηση εισέρχεται σε ένα βρόχο, επαναλαμβάνοντας κάθε jsonData στη λίστα εισόδου. Για κάθε επανάληψη, καλεί τη συνάρτηση parseEvents, λαμβάνοντας μια λίστα συμβάντων από την τρέχουσα συμβολοσειρά JSON. Η μέθοδος addAll χρησιμοποιείται για την προσθήκη αυτών των συμβάντων στη λίστα allEvents. Αυτή η επαναληπτική διαδικασία συνεχίζεται μέχρι να επεξεργαστούν όλες οι συμβολοσειρές JSON στη λίστα εισόδου. Το αποτέλεσμα είναι μια ολοκληρωμένη λίστα, allEvents, που περιέχει όλα τα συγκεντρωτικά συμβάντα από τη λίστα εισόδου των συμβολοσειρών JSON.

Συνοπτικά, η συνάρτηση getallevents καταδεικνύει τη χρησιμότητα της συγκέντρωσης δεδομένων από διαφορετικές πηγές, παρουσιάζοντας τη συνάφειά της σε σενάρια όπου τα γεγονότα μπορούν να ανακτηθούν από διαφορετικά τελικά σημεία ή σε ξεχωριστές παρτίδες. Η ικανότητα να χρησιμοποιείται η συνάρτηση parseEvents μέσα σε αυτή τη διαδικασία συγκέντρωσης προσθέτει στη συνολική συντηρησιμότητα και επεκτασιμότητα της βάσης κώδικα.

2.5. Εμφάνιση Συμβάντος Στο Ημερολόγιο

Η σελίδα προβολής ημερολογίου υλοποιήθηκε για τη λήψη και την εμφάνιση συμβάντων που σχετίζονται με τους αποθηκευμένους συνδέσμους iCal. Το πακέτο Flutter [table_calendar](#) (table_calendar | Flutter Package, 2023) χρησιμοποιήθηκε για τη δημιουργία μιας διαδραστικής και οπτικά ελκυστικής διεπαφής ημερολογίου.

To TableCalendar (Κώδικας 14), ένα βασικό στοιχείο που συμβάλλει στη διεπαφή χρήστη και τις διαδραστικές λειτουργίες της εφαρμογής. Αυτό widget έχει σχεδιαστεί για να παρουσιάζει μια οπτικά ελκυστική διεπαφή ημερολογίου με δυνατότητα χειρισμού συμβάντων, διαφορετικών στυλ και αλληλεπιδράσεων χρηστών.

```
child: TableCalendar<Event>(  
  daysOfWeekStyle: DaysOfWeekStyle(  
    weekdayStyle: TextStyle(color: ■Color(0xFF5FBF4)), // DaysOfWeekStyle  
  ),  
  headerStyle: HeaderStyle(  
    decoration: BoxDecoration(  
      color: ■Color(0xFF5FBF4),  
      borderRadius: BorderRadius.circular(10),  
    ), // BoxDecoration  
    headerMargin: EdgeInsets.only(bottom: 10),  
    headerPadding: EdgeInsets.all(1)), // HeaderStyle  
  firstDay: kFirstDay,  
  lastDay: kLastDay,  
  focusedDay: _focusedDay,  
  selectedDayPredicate: (day) => isSameDay(_selectedDay, day),  
  rangeStartDay: _rangeStart,  
  rangeEndDay: _rangeEnd,  
  calendarFormat: _calendarFormat,  
  rangeSelectionMode: _rangeSelectionMode,  
  eventLoader: _getEventsForDay,  
  startingDayOfWeek: StartingDayOfWeek.monday,  
  calendarStyle: CalendarStyle(  
    // Use `CalendarStyle` to customize the UI  
    weekendTextStyle: TextStyle(color: ■Color(0xFF858585)),  
    weekNumberTextStyle: TextStyle(color: ■Color(0xFF858585)),  
    markerDecoration: BoxDecoration(  
      color: ■Color(0xFFB46617),  
      borderRadius: BorderRadius.circular(20)), // BoxDecoration  
    defaultTextStyle: TextStyle(color: ■Color(0xFF5FBF4)),  
  
    outsideDaysVisible: false,  
  ), // CalendarStyle  
  onDaySelected: _onDaySelected,  
  onRangeSelected: _onRangeSelected,  
  onFormatChanged: (format) {  
    if (_calendarFormat != format) {  
      setState(() {  
        _calendarFormat = format;  
      });  
    }  
  },  
  onPageChanged: (focusedDay) {  
    _focusedDay = focusedDay;  
  },  
) // TableCalendar
```

Κώδικας 14: TableCalendar Widget

Οι αισθητικές αποχρώσεις συντονίζονται με ακρίβεια μέσω της ιδιότητας `dayofweekstyle`, επιτρέποντας την προσαρμογή των ημερών της εβδομάδας. Ταυτόχρονα, η ιδιότητα `headerStyle` συμβάλλει στη εμφάνιση της κεφαλίδας του ημερολογίου. Αυτό περιλαμβάνει τη συνετή χρήση ενός στρογγυλεμένου περιγράμματος με ακτίνα 10.

Οι βασικοί χρονικοί περιορισμοί ορίζονται από την πρώτη και την τελευταία ημέρα, οριοθετώντας το εύρος των ημερομηνιών που είναι οπτικά αντιληπτές στο ημερολόγιο. Το χαρακτηριστικό `_focusedDay` (Κώδικας 15) βρίσκεται στο επίκεντρο, ενσωματώνοντας την ημέρα που επιλέγεται από τον χρήστη.

```
final kToday = DateTime.now();  
final kFirstDay = DateTime(kToday.year, kToday.month - 3, kToday.day);  
final kLastDay = DateTime(kToday.year, kToday.month + 12, kToday.day);
```

Κώδικας 15: Days defined

Οι αλληλεπιδράσεις που βασίζονται σε συμβάντα γίνονται με χάρη μέσω του `selectedDayPredicate`, καθορίζοντας με σύνεση εάν μια συγκεκριμένη ημέρα ευθυγραμμίζεται με την επιλεγμένη ημέρα, ένα βασικό χαρακτηριστικό για επιλογές που βασίζονται στον χρήστη. Επιπλέον, η πρόβλεψη για επιλογές εύρους ημερομηνιών αντιμετωπίζεται κομψά από το `rangeStartDay` και το `rangeEndDay`, συμβάλλοντας σε έναν ολιστικό μηχανισμό επιλογής στο widget.

Η δυνατότητα διαμόρφωσης του ημερολογίου εμπλουτίζεται από παραμέτρους όπως το `calendarFormat`, το οποίο βοηθά στην οπτική αναπαράσταση του χρόνου, που κυμαίνεται από μηνιαίες έως εβδομαδιαίες μορφές. Εν τω μεταξύ, το `rangeSelectionMode` προσφέρει ένα φάσμα τρόπων επιλογής, επιτρέποντας τον ευέλικτο χειρισμό μεμονωμένων ημερών ή ολόκληρων ημερομηνιών. Η δυναμική φύση της φόρτωσης συμβάντων εκδηλώνεται μέσω της επανάκλησης του `eventLoader`, ενσωματώνοντας απρόσκοπτα εξωτερικά δεδομένα στην οπτική αφήγηση του ημερολογίου.

Η συνοχή της πλοήγησης επιτυγχάνεται με την ανάθεση της έναρξης της εβδομάδας με τη Δευτέρα, που ενσωματώνεται στην ιδιοκτησία `startingDayOfWeek`. Συμπληρώνοντας αυτό, η επανάκληση `onformatchanged` αντιδρά δυναμικά στις αλλαγές στη μορφή ημερολογίου, διασφαλίζοντας ότι το γραφικό στοιχείο παραμένει συγχρονισμένο με τις αλληλεπιδράσεις των χρηστών.

Η εμφάνιση του ημερολογίου διαμορφώνεται σχολαστικά από την ιδιότητα `calendarStyle`, επιτρέποντας ένα επίπεδο προσαρμογής εις βάθος. Αυτό περιλαμβάνει την τελειοποίηση των στυλ των αριθμών Σαββατοκύριακου και εβδομάδας, την καλλιτεχνική αναπαράσταση των δεικτών με μια ξεχωριστή χρωματική παλέτα και τη συνολική τυπογραφική συνέπεια που υποδηλώνεται από τις αποχρώσεις του στυλ κειμένου.

Συμπερασματικά, αυτή η σχολαστικά κατασκευασμένη διαμόρφωση για το widget `TableCalendar` όχι μόνο προσδίδει μια αισθητική φινέτσα στην εφαρμογή Flutter αλλά εξασφαλίζει επίσης μια απρόσκοπτη και διαδραστική εμπειρία χρήστη.

2.6. Δομή αρχείων και διαλογή πληροφοριών

Τα παρεχόμενα δεδομένα iCal προσφέρουν ένα συστηματικό στιγμιότυπο πληροφοριών σχετικά με τις κρατήσεις των καταλυμάτων, ενσωματώνοντας μια σειρά εκδηλώσεων υπό τον τύπο VEVENT. Αυτή η δομημένη μορφή συμμορφώνεται με το ευρέως αναγνωρισμένο πρότυπο iCalendar, εξασφαλίζοντας συμβατότητα και διευκολύνοντας την απρόσκοπτη ενσωμάτωση με εφαρμογές προσανατολισμένες στο ημερολόγιο.

Κάθε καταχώριση στο σύνολο δεδομένων ακολουθεί ένα συνεπές μοτίβο, περιγράφοντας λεπτομερώς βασικές παραμέτρους όπως τον τύπο συμβάντος, τη χρονική σήμανση, τις ημερομηνίες έναρξης και λήξης, Μια συνοπτική περίληψη που περιγράφει τις λεπτομέρειες της κράτησης, ένα μοναδικό αναγνωριστικό (UID) που σχετίζεται με το συμβάν και, περιστασιακά, ένα πεδίο περιγραφής που περιλαμβάνει συμπληρωματικές λεπτομέρειες. Η ευελιξία της μορφής iCalendar επιτρέπει τη συμπερίληψη ποικίλων πληροφοριών, που κυμαίνονται από βασικά στοιχεία κράτησης έως πρόσθετες σημειώσεις ή στοιχεία επικοινωνίας.

Η ευελιξία των δεδομένων είναι εμφανής στην ποικιλία των επιφυλάξεων, η καθεμία από τις οποίες αντιπροσωπεύεται από μια ξεχωριστή καταχώριση. Αυτές οι κρατήσεις ποικίλλουν όσον αφορά τα ονόματα των επισκεπτών, εύρη ημερομηνιών, και περιστασιακά περιλαμβάνουν πρόσθετες λεπτομέρειες, όπως αριθμούς τηλεφώνου. Η τήρηση του προτύπου iCalendar όχι μόνο διευκολύνει την απρόσκοπτη ενσωμάτωση αυτών των δεδομένων στην προβολή ημερολογίου της εφαρμογής, αλλά διασφαλίζει επίσης τη λειτουργικότητα με ένα ευρύ φάσμα πλατφορμών που βασίζονται σε ημερολόγια.

Παρακάτω φαίνεται η δομή αυτών των αρχείων iCal (Εικόνα 3).

```
{type: VEVENT, dtstamp: {dt: 20240125T175159Z}, dtstart: {dt: 20220611}, dtend: {dt: 20220618}, summary: Onore Villa -Reservation
HMF79HCJSD-11 Jun-18 Jun, uid: 04c6ef58-ebef-4f8e-9799-7a17b6692c62@guestyforhosts.com, description: }, {type: VEVENT, dtstamp: {dt:
20240125T175159Z}, dtstart: {dt: 20230604}, dtend: {dt: 20230618}, summary: Onore Villa -Trevor Lloyd-04 Jun-18 Jun, uid:
076b8609-8121-49aa-9c8a-5a87f2508909@guestyforhosts.com, description: Phone Number(Last 4 Digits): 1286 Phone Number: +44 79692 41286},
{type: VEVENT, dtstamp: {dt: 20240125T175159Z}, dtstart: {dt: 20230805}, dtend: {dt: 20230812}, summary: Onore Villa -Liana Miliotis-05
Aug-12 Aug, uid: 2762e9fc-9da7-478c-9adf-94bed5441025@guestyforhosts.com, description: }, {type: VEVENT, dtstamp: {dt: 20240125T175159Z},
dtstart: {dt: 20210703}, dtend: {dt: 20210710}, summary: Onore Villa -Jim Royds-03 Jul-10 Jul, uid: 37708e99-
aef9-4b8a-9319-8118636eae4a@guestyforhosts.com, description: }, {type: VEVENT, dtstamp: {dt: 20240125T175159Z}, dtstart: {dt: 20240810},
dtend: {dt: 20240817}, summary: Onore Villa -Stavros Guests-10 Aug-17 Aug, uid: 41989a42-7be4-4966-9893-c7f57f5aba42@guestyforhosts.com,
description: }, {type: VEVENT, dtstamp: {dt: 20240125T175159Z}, dtstart: {dt: 20230824}, dtend: {dt: 20230903}, summary: Onore Villa -
Magdalena Gieruszczak-24 Aug-03 Sep, uid: 43282bbd-181e-4e16-96a7-d78a51a35140@guestyforhosts.com, description: Phone Number(Last 4 Digits):
6654 Phone Number: +48505106654}, {type: VEVENT, dtstamp: {dt: 20240125T175159Z}, dtstart: {dt: 20230729}, dtend: {dt: 20230805}, summary:
Onore Villa -Annabel Dumbell-29 Jul-05 Aug, uid: 446f88d1-3fea-4035-b130-a8b1eb99e28e@guestyforhosts.com, description: Phone Number(Last 4
Digits): 4959 Phone Number: +447766174959}, {type: VEVENT, dtstamp: {dt: 20240125T175159Z}, dtstart: {dt: 20230630}, dtend: {dt: 20230707},
summary: Onore Villa -Emma McConnachie-30 Jun-07 Jul, uid: 4dbd6c99-7af1-4333-8690-fa7ffc3d8500@guestyforhosts.com, description: Phone
Number(Last 4 Digits): Phone Number: +1}, {type: VEVENT, dtstamp: {dt: 20240125T175159Z}, dtstart: {dt: 20240611}, dtend: {dt: 20240618},
summary: Onore Villa -Alannah Knutton-11 Jun-18 Jun, uid: 5f23af42-20e1-4173-b2e3-a3b449ff13fe@guestyforhosts.com, description: Phone
Number(Last 4 Digits): 6243 Phone Number: +447595256243}, {type: VEVENT, dtstamp: {dt: 20240125T175159Z}, dtstart: {dt: 20220730}, dtend:
{dt: 20220809}, summary: Onore Villa -Daniel Harris-30 Jul-09 Aug, uid: 5f97676a-e893-45c9-a715-f5ff74919011@guestyforhosts.com,
description: }, {type: VEVENT, dtstamp: {dt: 20240125T175159Z}, dtstart: {dt: 20230816}, dtend: {dt: 20230823}, summary: Onore Villa -CHRIS
MCLAUGHLIN-16 Aug-23 Aug, uid: 693ce333-90a3-45e9-873d-0b764de82f45@guestyforhosts.com, description: Phone Number(Last 4 Digits): -278 Phone
Number: +44 07540-445-278}, {type: VEVENT, dtstamp: {dt: 20240125T175159Z}, dtstart: {dt: 20240801}, dtend: {dt: 20240808}, summary: Onore
Villa -Ian Coyle-01 Aug-08 Aug, uid: 6c4b9a0b-7b43-4d61-b6f7-ca013761bd76@guestyforhosts.com, description: Phone Number(Last 4 Digits): 1437
Phone Number: +447789791437}, {type: VEVENT, dtstamp: {dt: 20240125T175159Z}, dtstart: {dt: 20230519}, dtend: {dt: 20230526}, summary: Onore
Villa -Ella Fish-19 May-26 May, uid: 72b30f68-aed8-4ca5-a678-a8e6910abf78@guestyforhosts.com, description: Phone Number(Last 4 Digits): 1259
Phone Number: +44 788431259}
```

Εικόνα 3: iCal data structure

Για την μετατροπή των δεδομένων από τα αρχεία ακολουθήθηκε η διαδικασία που αναφέρθηκε προηγουμένως καθώς και μια έξτρα επεξεργασία για την απεικόνιση τους σε μορφή λίστας στην σελίδα κρατήσεων. Όπως φαίνεται στον παρακάτω κώδικα (Κώδικας 16), η λειτουργία getAllEvents βρίσκεται στο επίκεντρο, χρησιμεύοντας ως βασικός μηχανισμός για την επεξεργασία και τη βελτίωση μιας συλλογής γεγονότων που κωδικοποιούνται ως συμβολοσειρές JSON.

```

List<Map<String, dynamic>> getAllEvents(List<String> jsonDataList) {
    List<Map<String, dynamic>> allEvents = [];
    for (String jsonData in jsonDataList) {
        allEvents.addAll(parseEvents(jsonData));
    }

    // Filter out events with end dates earlier than today
    DateTime today = DateTime.now();
    allEvents = allEvents.where((event) {
        DateTime endDate = DateTime.parse(event['dtend']['dt']);
        return endDate.isAfter(today);
    }).toList();

    // Sort the list by the 'dtstart' key
    allEvents.sort((a, b) {
        DateTime dateA = DateTime.parse(a['dtstart']['dt']);
        DateTime dateB = DateTime.parse(b['dtstart']['dt']);
        return dateA.compareTo(dateB);
    });

    // Apply property filter if selected
    if (propertyFilter != null && propertyFilter != 'All') {
        allEvents = allEvents
            .where((event) =>
                event['summary']
                    ?.toLowerCase()
                    .contains(propertyFilter!.toLowerCase()) ??
                false)
            .toList();
    }

    return allEvents;
}

```

Κώδικας 16: Filtering and sorting reservations

Το αρχικό βήμα περιλαμβάνει την εμφάνιση μιας κενής λίστας, που ονομάζεται κατάλληλα `allEvents`, έτοιμη να συγχωνεύσει τα αποτελέσματα της ανάλυσης κάθε συμβολοσειράς JSON εντός του παρεχόμενου `jsonDataList`. Η συγκεκριμένη επίκληση της λειτουργίας `parseEvents` μέσα σε ένα διακριτό βρόχο αποκωδικοποιεί αποτελεσματικά τις αναπαραστάσεις JSON, ενώ η μέθοδος `addAll` ενσωματώνει απρόσκοπτα αυτά τα αναλυμένα συμβάντα στη λίστα συσσωρευμένων `allEvents`.

Μια λεπτή αλλά κρίσιμη βελτίωση ξεδιπλώνεται στο επόμενο στάδιο, όπου τα γεγονότα με ημερομηνίες λήξης που προηγούνται της τρέχουσας ημερομηνίας φιλτράρονται σχολαστικά. Αξιοποιώντας την κλάση `DateTime`, ο κώδικας αναφέρεται διακριτικά στην τρέχουσα ημερομηνία, δημιουργώντας ένα διακριτικό χρονικό όριο. Αυτό το όριο χρησιμοποιείται στη συνέχεια για να φιλτράρει τη λίστα `allEvents`, διασφαλίζοντας ότι παραμένουν μόνο συμβάντα με ημερομηνίες λήξης στο μέλλον. Αυτή η επιμέλεια πίσω από

τα παρασκήνια ενισχύει τη συνάφεια των δεδομένων συμβάντων, παρουσιάζοντας στους χρήστες αποκλειστικά επερχόμενα και σχετικά περιστατικά.

Ο κώδικας αποκαλύπτει περαιτέρω την πολυπλοκότητά του εισάγοντας έναν διακριτό μηχανισμό ταξινόμησης, ένα βασικό στοιχείο για την παρουσίαση μιας συνεκτικής και φιλικής προς το χρήστη προβολής των γεγονότων. Χρησιμοποιώντας τη μέθοδο ταξινόμησης, τα συμβάντα στη λίστα `allEvents` οργανώνονται σχολαστικά με βάση τις ημερομηνίες έναρξης τους. Αυτή η λεπτή χρονολογική σειρά προσθέτει ένα επίπεδο διαισθητικής οργάνωσης, διασφαλίζοντας ότι οι χρήστες μπορούν να πλοηγηθούν αβίαστα στα γεγονότα σε μια απρόσκοπτη χρονική ακολουθία.

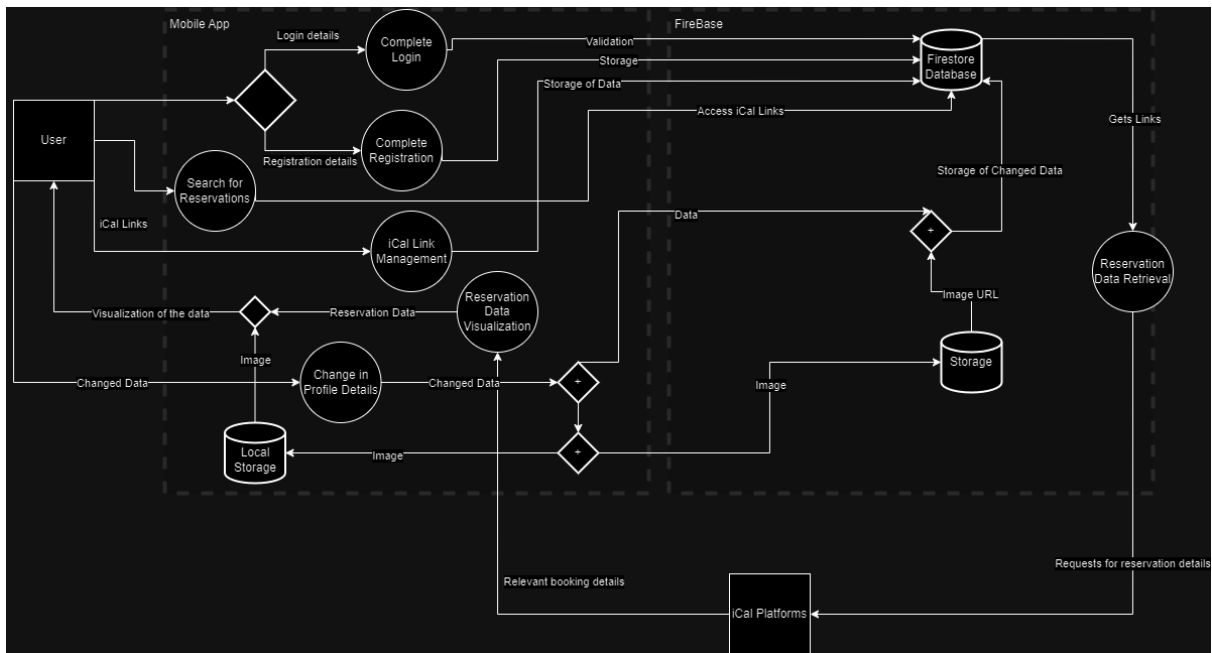
Επιπρόσθετα από την περιγραφή του γεγονότος μπορεί να διακριθεί σε πιο κατάλυμα αντιστοιχεί η κράτηση και αργότερα να ομαδοποιηθούν. Αυτή η λειτουργία επιτρέπει την επιλογή εμφάνισης των κρατήσεων ανά κατάλυμα της αρεσκείας του χρήστη ή όλων μαζί. Οι επιλογές αυτές εμφανίζονται στον χρήστη σε μορφή dropdown για να επιλέξει και επιτεύχθηκε με την συνάρτηση (Κώδικας 17).

```
List<String> getDistinctProperties(List<Map<String, dynamic>> events) {
    Set<String> distinctProperties = Set<String>();
    for (var event in events) {
        String summary = event['summary'] ?? '';
        List<String> summaryParts = summary.split('-');
        if (summaryParts.isNotEmpty) {
            distinctProperties.add(summaryParts[0].trim());
        }
    }
    return distinctProperties.toList();
}
```

Κώδικας 17: `getdistinctproperties` Function

Παρακάτω φαίνεται το διάγραμμα ροής δεδομένων (Data Flow Diagram ,Διάγραμμα 1) για την εφαρμογή, το οποίο απεικονίζει την απρόσκοπτη αλληλεπίδραση μεταξύ διαφόρων στοιχείων και εξωτερικών οντοτήτων, απεικονίζοντας μια σαφή αναπαράσταση της ροής πληροφοριών εντός του συστήματος.

Στον πυρήνα του διαγράμματος βρίσκονται οι εξωτερικές οντότητες, που περιλαμβάνουν κυρίως χρήστες και πλατφόρμες iCal. Οι χρήστες ξεκινούν αλληλεπιδράσεις με την εφαρμογή, συμμετέχοντας σε εργασίες όπως σύνδεση, εγγραφή, διαχείριση iCal link και οπτικοποίηση κρατήσεων. Από την άλλη πλευρά, οι πλατφόρμες iCal αντιπροσωπεύουν εξωτερικά συστήματα όπως το Airbnb και το Booking, ενεργώντας ως πηγές δεδομένων παρέχοντας πληροφορίες κράτησης μέσω συνδέσμων iCal.



Διάγραμμα 1: Data Flow Diagram

Η διευκόλυνση αυτών των αλληλεπιδράσεων είναι η ισχυρή υπηρεσία ελέγχου ταυτότητας Firebase, ένα βασικό εσωτερικό στοιχείο που διαχειρίζεται τον έλεγχο ταυτότητας χρήστη με ασφάλεια. Εξασφαλίζει ένα αξιόπιστο και προστατευμένο περιβάλλον για ενέργειες που σχετίζονται με το χρήστη εντός της εφαρμογής.

Η ροή δεδομένων σε όλο το σύστημα γίνεται με ακρίβεια. Οι χρήστες υποβάλλουν αιτήματα στην υπηρεσία ελέγχου ταυτότητας Firebase για σκοπούς ελέγχου ταυτότητας. Στη συνέχεια, οι πιστοποιημένοι χρήστες μπορούν να έχουν πρόσβαση και χειρισμό των δεδομένων τους, συμπεριλαμβανομένης της αποθήκευσης και ανάκτησης συνδέσμων iCal.

Ο ζωτικός σύνδεσμος μεταξύ της εφαρμογής και των εξωτερικών πλατφορμών κρατήσεων δημιουργείται μέσω της απόκτησης και επεξεργασίας συνδέσμων iCal. Το σύστημα χειρίζεται επιδέξια την είσοδο των iCal συνδέσεων, εξάγοντας τα σχετικά δεδομένα κράτησης και καθιστώντας τα διαθέσιμα στους χρήστες για να τα απεικονίσουν στην προβολή ημερολογίου της εφαρμογής.

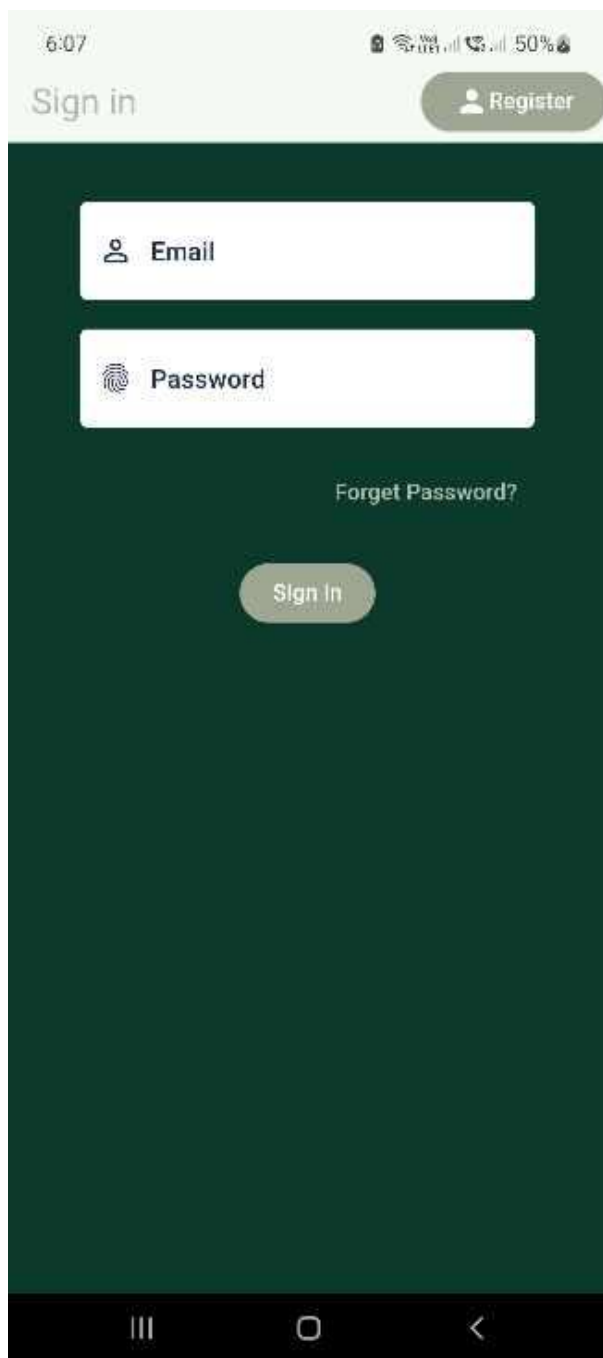
Αυτό το DFD συλλαμβάνει την ουσία της λειτουργικότητας της εφαρμογής, παρουσιάζοντας μια βελτιωμένη ροή πληροφοριών που εξασφαλίζει μια φιλική προς το χρήστη εμπειρία. Υπογραμμίζει την αρμονική συνεργασία μεταξύ εσωτερικών υπηρεσιών και εξωτερικών οντοτήτων, με αποκορύφωμα μια ολοκληρωμένη λύση για τη διαχείριση και την απεικόνιση δεδομένων ημερολογίου σε διάφορες πλατφόρμες.

2.7. Διεπαφή χρήστη και πλοήγηση

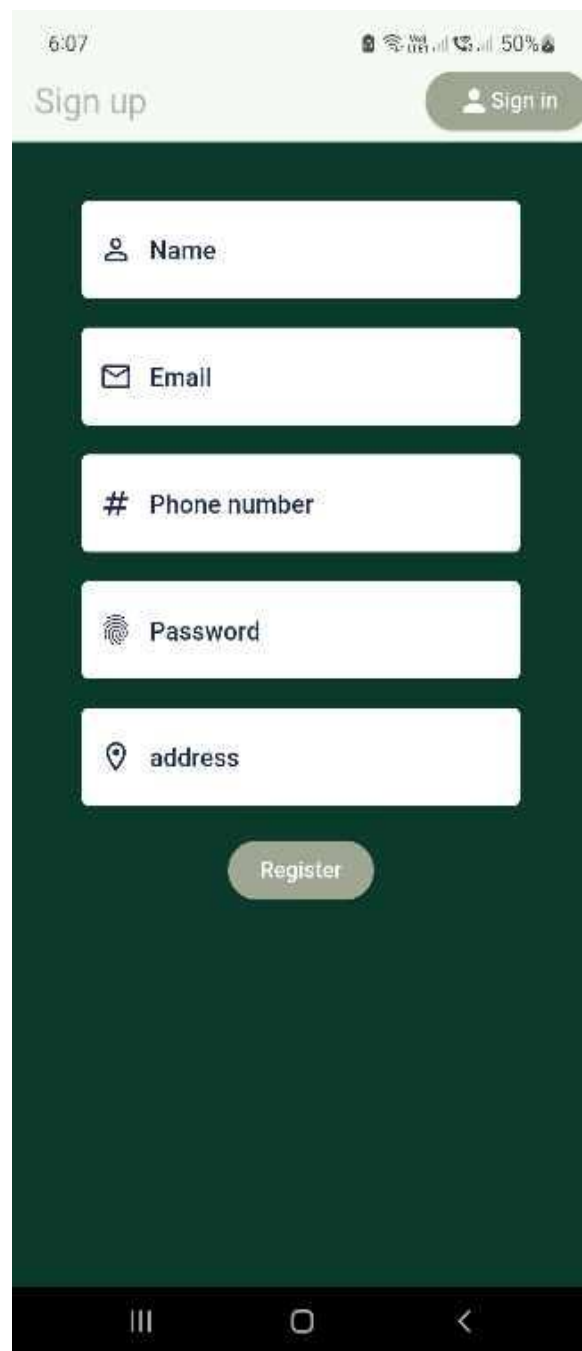
Η διεπαφή χρήστη (UI) σχεδιάστηκε για μια διαισθητική εμπειρία χρήστη, με χαρακτηριστικά όπως κουμπιά πλοήγησης, φόρμες εισαγωγής και μια ανταποκρινόμενη διάταξη. Το σύστημα widget του Flutter διευκόλυνε τη δημιουργία επαναχρησιμοποιήσιμων στοιχείων UI.

2.7.1 Sign-in/ Register Pages

Κατά την εκκίνηση της εφαρμογής εμφανίζεται η οθόνη σύνδεσης (Sign-in Page, Εικόνα 4), αν δεν έχει ήδη συνδεθεί κάποιος από αυτήν την συσκευή. Η οθόνη σύνδεσης προσφέρει μια βελτιωμένη και διαισθητική εμπειρία για χρήστες που επιθυμούν να έχουν πρόσβαση στις δυνατότητες της εφαρμογής. Οι χρήστες παρουσιάζονται με μια οπτικά ελκυστική διάταξη όπου μπορούν να εισάγουν απρόσκοπτα το email και τον κωδικό πρόσβασής τους σε καθορισμένα πεδία. Ο σχεδιασμός δίνει προτεραιότητα στη σαφήνεια και την ευκολία χρήσης, εξασφαλίζοντας μια απλή διαδικασία σύνδεσης.



Εικόνα 4: Sign-in Page



Εικόνα 5: Register Page

Για τους χρήστες που ενδέχεται να μην θυμούνται τον κωδικό πρόσβασης, ένα διακριτικό αλλά εύκολα προσβάσιμο "Forget Password?". Κατά το πάτημα, αυτός ο σύνδεσμος αποκαλύπτει τις επιλογές για ανάκτηση κωδικού πρόσβασης, με OTP είτε μέσω email είτε μέσω κινητού τηλεφώνου. Αυτή η στοχαστική ένταξη προσθέτει ένα επίπεδο λειτουργικότητας με επίκεντρο τον χρήστη, αντιμετωπίζοντας πιθανά εμπόδια πρόσβασης.

Η κύρια παρότρυνση για δράση, το κουμπί "Σύνδεση", χρησιμεύει ως πύλη για τον έλεγχο ταυτότητας χρήστη. Κατά την επιλογή, τα καταχωρημένα διαπιστευτήρια υποβάλλονται σε ταχεία επικύρωση. Εν τω μεταξύ, εμφανίζεται ένας δείκτης φόρτωσης, υποδηλώνοντας στους χρήστες ότι το αίτημά τους για έλεγχο ταυτότητας βρίσκεται σε εξέλιξη. Αυτή η οπτική ένδειξη διασφαλίζει ότι οι χρήστες ενημερώνονται κατά τη διάρκεια δυνητικά σύντομων περιόδων αναμονής.

Σε περίπτωση που ο χρήστης δεν έχει λογαριασμό μπορεί να μεταβεί στην εγγραφή χρήστη, με ένα στρατηγικά τοποθετημένο κουμπί "Sign up". Αυτή η επιλογή πλοήγησης δίνει τη δυνατότητα στους χρήστες να εξερευνούν και να δημιουργούν αβίαστα νέους λογαριασμούς, διατηρώντας μια συνεπή και λογική ροή εντός της εφαρμογής.

Σε περίπτωση προκλήσεων ελέγχου ταυτότητας, στους χρήστες εμφανίζετε ένα μήνυμα σφάλματος, παρέχοντας άμεση και σαφή ανατροφοδότηση. Αυτός ο προληπτικός χειρισμός σφαλμάτων βοηθά τους χρήστες να κατανοήσουν και να επιλύσουν τυχόν προβλήματα σύνδεσης, συμβάλλοντας σε μια συνολική θετική εμπειρία χρήστη.

Υπάρχει και μια προαιρετική δυνατότητα για ανώνυμη σύνδεση, η οποία προσφέρει μια εναλλακτική προσέγγιση για χρήστες που ενδέχεται να μην επιθυμούν να δημιουργήσουν έναν λογαριασμό αμέσως. Αυτή η ευελιξία αναγνωρίζει τις διαφορετικές προτιμήσεις των χρηστών και ευθυγραμμίζεται με τη δέσμευση της εφαρμογής για ένταξη, πρέπει να αναφερθεί ότι δεν έχει συμπεριληφθεί στο στήσιμο της εφαρμογής.

Η σελίδα εγγραφής (Register Page, Εικόνα 5) της εφαρμογής παρουσιάζει στους χρήστες μια οπτικά ελκυστική και φιλική προς το χρήστη διεπαφή που έχει σχεδιαστεί για να διευκολύνει τη διαδικασία δημιουργίας λογαριασμού. Το περιβάλλον χρήστη είναι κατασκευασμένο για να παρέχει μια διαισθητική εμπειρία, δίνοντας έμφαση στην ευκολία χρήσης και την απρόσκοπτη πλοήγηση.

Η σελίδα διαθέτει μια καθαρή διάταξη με στρατηγικά τοποθετημένα πεδία εισαγωγής κειμένου για τους χρήστες να εισάγουν τα στοιχεία εγγραφής τους. Αυτά τα πεδία, διακοσμημένα με κατάλληλα εικονίδια, ενισχύουν την οπτική έκκληση και καθοδηγούν τους χρήστες μέσω της διαδικασίας εισαγωγής πληροφοριών.

Για να διασφαλιστεί η ολοκληρωμένη εγγραφή χρήστη, η φόρμα ενσωματώνει πεδία για βασικές λεπτομέρειες όπως όνομα, ηλεκτρονικό ταχυδρομείο, αριθμό τηλεφώνου, κωδικό πρόσβασης και διεύθυνση. Κάθε πεδίο είναι εξοπλισμένο με κατάλληλους μηχανισμούς επικύρωσης για να εγγυάται την ακρίβεια και την πληρότητα των πληροφοριών που παρέχονται από τον χρήστη.

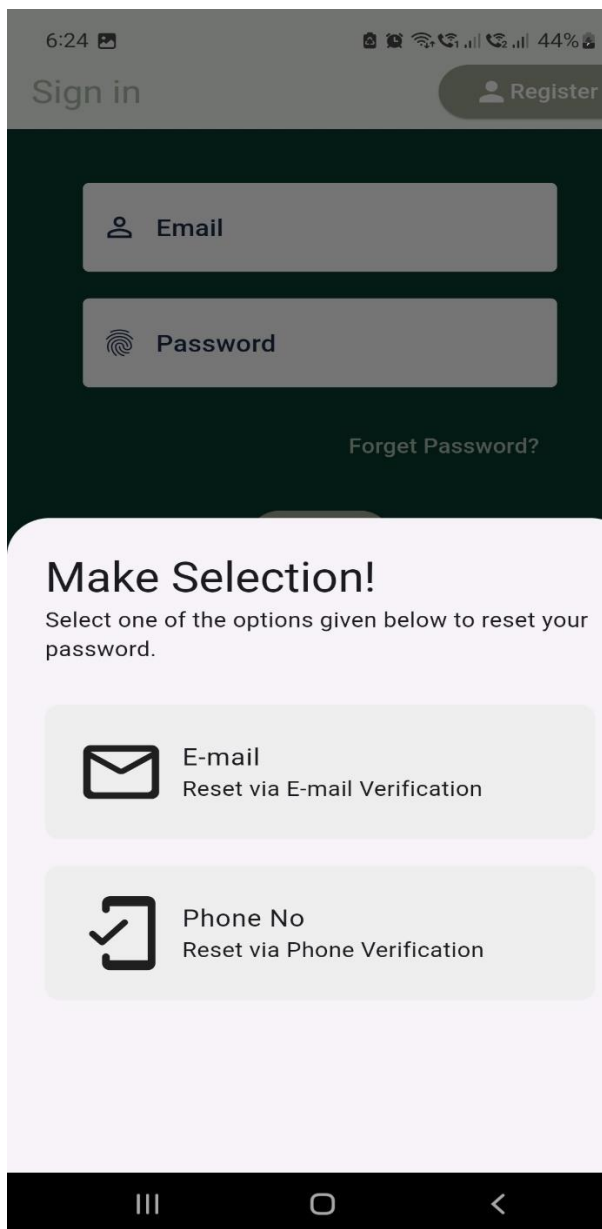
Η γραμμή εφαρμογών διατηρεί τη συνέπεια στο σχεδιασμό, προσφέροντας μια ομαλή μετάβαση μεταξύ της σελίδας εγγραφής και της οθόνης σύνδεσης. Η συμπερίληψη ενός

κουμπιού "Sign in" στη γραμμή εφαρμογών επιτρέπει στους χρήστες να περιηγούνται αβίαστα στην οθόνη σύνδεσης, παρέχοντας μια συνεκτική εμπειρία χρήστη.

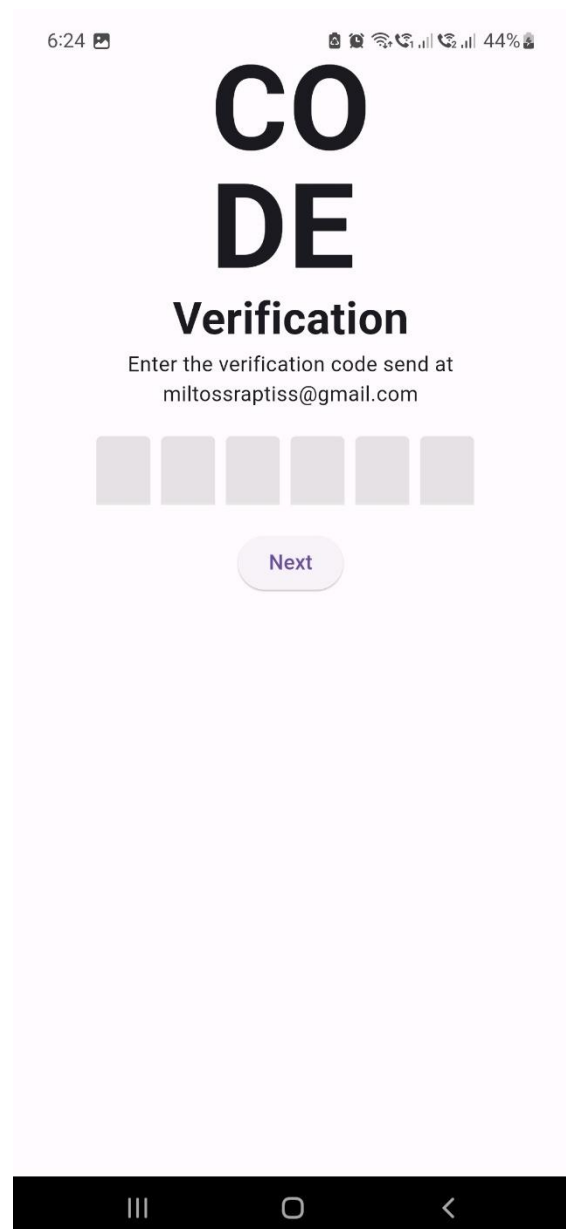
Η διαδικασία εγγραφής υποστηρίζεται από έναν δείκτη φόρτωσης, που εμφανίζεται διακριτικά για να ενημερώνει τους χρήστες για τις τρέχουσες δραστηριότητες εγγραφής. Αυτή η οπτική ένδειξη διασφαλίζει ότι οι χρήστες γνωρίζουν τις διαδικασίες υποβάθρου, συμβάλλοντας σε μια πιο διαφανή και ανταποκρινόμενη διεπαφή χρήστη.

Το κουμπί "Εγγραφή", τοποθετημένο εμφανώς στη σελίδα, χρησιμεύει ως η κύρια παρότρυνση για δράση για την έναρξη της διαδικασίας εγγραφής. Κατά την επιλογή, τα καταχωρημένα στοιχεία εγγραφής υποβάλλονται σε επικύρωση και, εάν είναι επιτυχημένα, οι χρήστες εγγράφονται απρόσκοπτα. Τυχόν πιθανά σφάλματα κοινοποιούνται μέσω ενός συνοπτικού και οπτικά διακριτού μηνύματος σφάλματος, επιτρέποντας στους χρήστες να αντιμετωπίσουν άμεσα τυχόν προβλήματα.

Στην σελίδα σύνδεσης, κατά το πάτημα του κουμπιού "Forget Password?" εμφανίζονται οι επιλογές για ανάκτηση κωδικού πρόσβασης, με OTP είτε μέσω email είτε μέσω κινητού τηλεφώνου (Εικόνα 6), επιλέγοντας τρόπο οι χρήστες εισάγουν το τηλέφωνο τους είτε το email τους για να τους σταλεί το OTP και να το πληκτρολογήσουν στην εφαρμογή για επαλήθευση (Εικόνα 7).



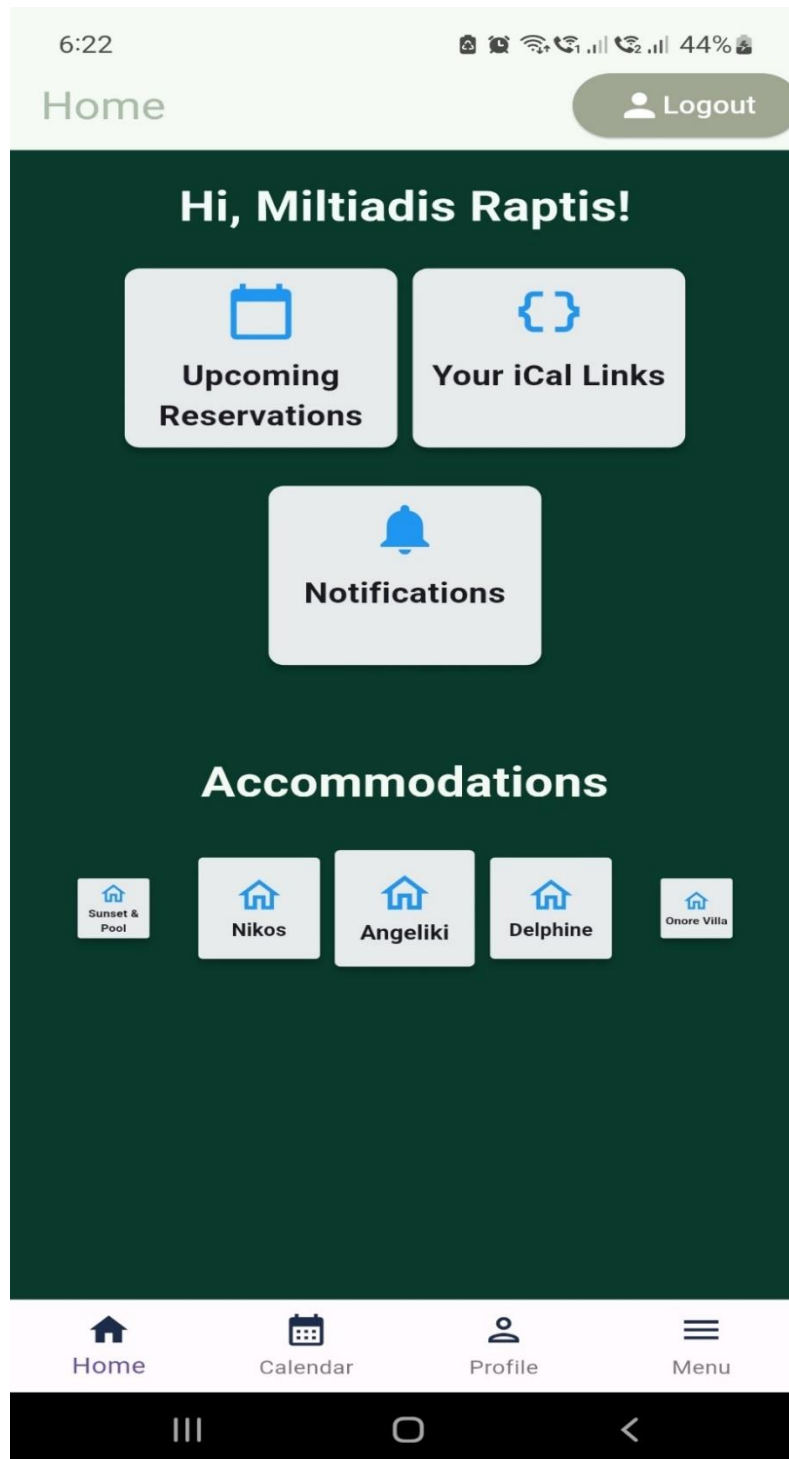
Εικόνα 6: Select Method OTP



Εικόνα 7: Enter the OTP

2.7.1 Home Page

Η αρχική σελίδα (Home Page, Εικόνα 8) της εφαρμογής χρησιμεύει ως κεντρικός κόμβος για τους χρήστες, παρέχοντας μια δυναμική και ελκυστική διεπαφή που καλύπτει τις ανάγκες τους. Δημιουργημένο με μοντέρνο σχεδιασμό και φιλική προς το χρήστη διάταξη, η σελίδα ενσωματώνει απρόσκοπτα διάφορες λειτουργίες, ενισχύοντας τη συνολική εμπειρία χρήστη.



Εικόνα 8: Home Page

Κατά την είσοδο στην αρχική σελίδα, Οι χρήστες καλωσορίζονται με ένα εξατομικευμένο μήνυμα με το όνομα τους. Αυτή η στοχαστική πινελιά δημιουργεί μια φιλόξενη ατμόσφαιρα, ενισχύοντας την αίσθηση σύνδεσης και εξατομίκευσης μέσα στην εφαρμογή.

Το κύριο περιεχόμενο της αρχικής σελίδας οργανώνεται σε ξεχωριστές ενότητες, καθεμία από τις οποίες εξυπηρετεί έναν συγκεκριμένο σκοπό. Το κουμπί "Upcoming Reservations" προσφέρει μια γρήγορη ματιά στις επερχόμενες κρατήσεις, επιτρέποντας στους χρήστες να

ενημερώνονται για το πρόγραμμά τους. Επιπλέον, Το κουμπί "Your iCal Links" παρέχει εύκολη πρόσβαση στους συνδέσμους iCal, που έχει αποθηκεύσει ο χρήστης.

Η πλοήγηση απλοποιείται περαιτέρω με διαδραστικές κάρτες, κομψά εμφανισμένες και διακοσμημένες με σχετικά εικονίδια. Αυτές οι κάρτες λειτουργούν ως κουμπιά πλοήγησης για βασικά χαρακτηριστικά όπως ειδοποιήσεις, επιτρέποντας στους χρήστες να έχουν εύκολη πρόσβαση σε σχετικές πληροφορίες.

Για χρήστες με συγκεκριμένες ανάγκες διαμονής, ένα οπτικά ελκυστικό καρousel παρουσιάζει τα διαθέσιμα καταλύματα. Κάθε κατάλυμα παρουσιάζεται με μια εικόνα και το όνομα του, προσφέροντας στους χρήστες μια ματιά στις κρατήσεις των διαθέσιμων καταλυμάτων. Το carousel ενσωματώνεται απρόσκοπτα στο συνολικό σχεδιασμό, ενισχύοντας την οπτική γοητεία της αρχικής σελίδας.

Ο συνολικός συνδυασμός χρωμάτων, που χαρακτηρίζεται από γήινους τόνους, συμβάλλει σε μια ήρεμη και οπτικά ευχάριστη αισθητική. Αυτή η επιλογή χρωμάτων όχι μόνο ενισχύει την αναγνωσιμότητα αλλά και αντικατοπτρίζει μια σύνδεση με τη φύση, προωθώντας μια ήρεμη εμπειρία χρήστη.

Από άποψη λειτουργικότητας, η αρχική σελίδα οδηγείται από δεδομένα σε πραγματικό χρόνο, χρησιμοποιώντας ροές για να διατηρεί τις πληροφορίες ενημερωμένες. Αυτό διασφαλίζει ότι οι χρήστες λαμβάνουν έγκαιρες και ακριβείς λεπτομέρειες σχετικά με τις κρατήσεις τους και άλλες σχετικές δραστηριότητες.

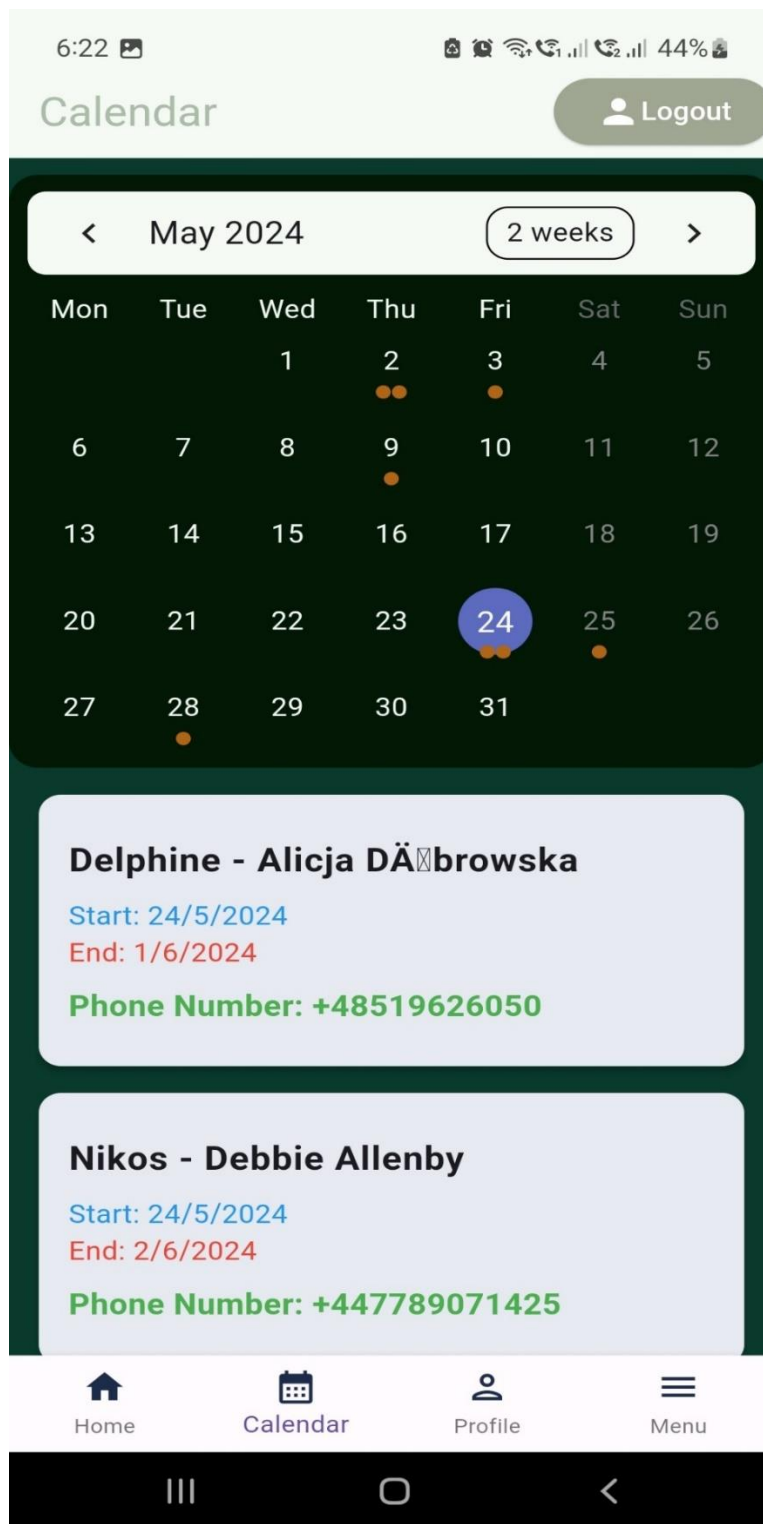
Τέλος υπάρχει το BottomNavigationBar, που αποτελεί βασικό στοιχείο για τη διευκόλυνση της πλοήγησης εντός της εφαρμογής. Τοποθετημένο στο κάτω μέρος της οθόνης, το BottomNavigationBar έχει σχεδιαστεί για να επιτρέπει στους χρήστες να μεταβαίνουν απρόσκοπτα μεταξύ διαφορετικών τμημάτων ή οθονών.

Κάθε στοιχείο γραμμής πλοήγησης αντιπροσωπεύεται από ένα BottomNavigationBarItem. Αυτά τα στοιχεία αποτελούνται από ένα συσχετισμένο εικονίδιο—που αντιπροσωπεύεται από το γραφικό στοιχείο εικονιδίου, το οποίο εμφανίζει μια οπτική αναπαράσταση της ενότητας—και μια ετικέτα, η οποία παρέχει μια περιγραφή κειμένου της ενότητας.

Περιλαμβάνει τέσσερα ξεχωριστά στοιχεία πλοήγησης: "Home", "Calendar", "Profile" και "Menu". Κάθε στοιχείο συνδέεται με ένα εικονίδιο και μια ετικέτα, συμβάλλοντας σε μια οπτικά διαισθητική και φιλική προς το χρήστη εμπειρία πλοήγησης.

2.7.2 Calendar Page

Στην σελίδα ημερολογίου (Calendar Page, Εικόνα 9) οι χρήστες μπορούν να αλληλοεπιδράσουν με το ημερολόγιο, επιλέγοντας είτε μία ημερομηνία είτε ένα εύρος ημερομηνιών. Το ημερολόγιο περιλαμβάνει λειτουργίες όπως η επισήμανση επιλεγμένων ημερών και η διαχείριση επιλογών εύρους ημερομηνιών, δηλαδή να μπορούν να επιλέξουν αν θα απεικονίζεται όλος ο μήνας ή μόνο δύο εβδομάδες ή απλά μόνο μία. Κάτω από το ημερολόγιο, υπάρχει μια κατακόρυφα κυλιόμενη λίστα που συμπληρώνεται δυναμικά με βάση τις επιλεγμένες ημερομηνίες από το ημερολόγιο. Επισημαίνονται όλες οι μέρες που υπάρχουν κρατήσεις με πορτοκαλί κουκίδα για να διευκολύνονται οι χρήστες.

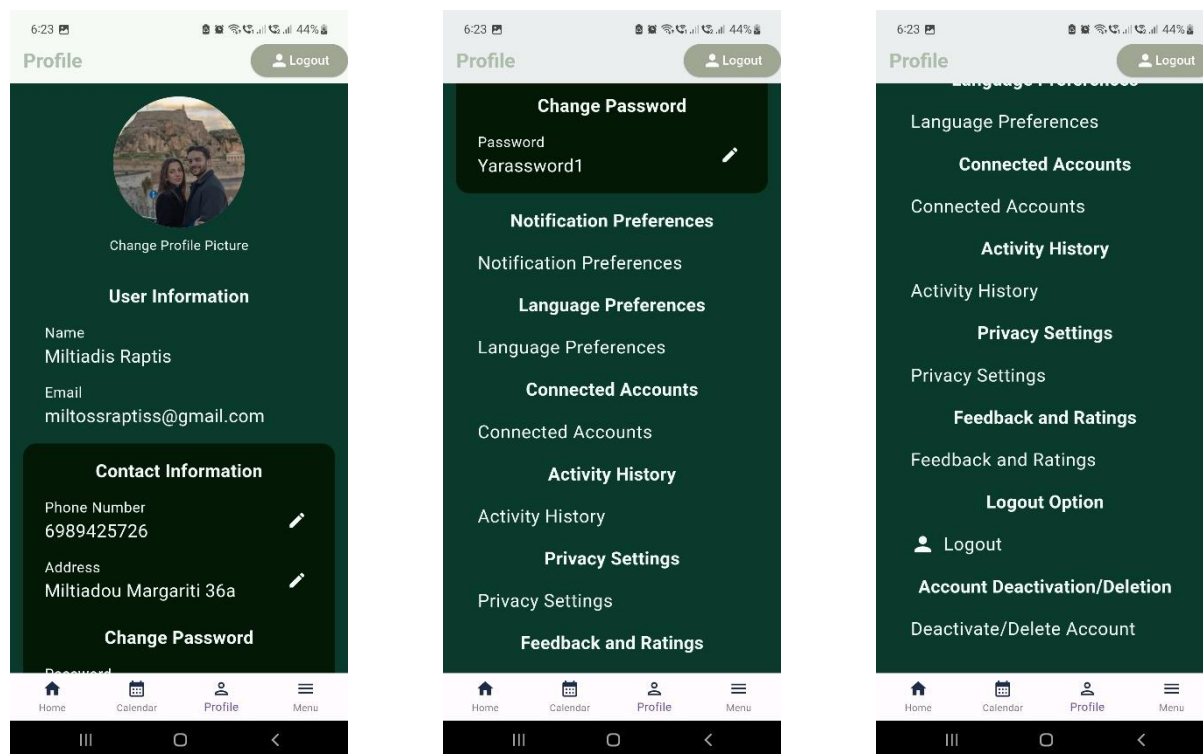


Εικόνα 9: Calendar Page

Η λίστα εμφανίζει τις κρατήσεις της επιλεγμένης μέρας και οι χρήστες μπορούν να επιλέξουν την κάθε κράτηση της λίστας για να δουν λεπτομέρειες. Ολόκληρη η διεπαφή στοχεύει στην παροχή μιας διαισθητικής και φιλικής προς το χρήστη εμπειρίας, επιτρέποντας στους χρήστες να περιηγούνται απρόσκοπτα στις ημερομηνίες και να έχουν πρόσβαση σε σχετικές πληροφορίες συμβάντων χωρίς κόπο. Ο συνδυασμός χρωμάτων και το στυλ συμβάλλουν σε έναν οπτικά συνεκτικό και αισθητικά ευχάριστο σχεδιασμό.

2.7.3 Profile Page

Στην σελίδα του προφίλ (Profile Page, Εικόνα 10), εμφανίζονται όλες οι πληροφορίες του χρήστη, είναι προφίλ είναι δομημένη σε ξεχωριστές ενότητες, η καθεμία από τις οποίες εξυπηρετεί συγκεκριμένες πληροφορίες και ενέργειες.



Εικόνα 10: Profile Page

Στο επάνω μέρος της οθόνης, οι χρήστες μπορούν να δουν την εικόνα προφίλ τους, η οποία είναι διαδραστική και τους επιτρέπει να ανεβάσουν ή να αλλάξουν την εικόνα. Κάτω από την εικόνα προφίλ, οι ενότητες εμφανίζουν βασικά στοιχεία χρήστη, όπως όνομα, ηλεκτρονικό ταχυδρομείο και άλλες προσωπικές πληροφορίες.

Η ενότητα Στοιχεία επικοινωνίας περιλαμβάνει τον αριθμό τηλεφώνου και τη διεύθυνση του χρήστη, που παρουσιάζονται με εικονίδια επεξεργασίας που επιτρέπουν στους χρήστες να ενημερώνουν αυτές τις πληροφορίες. Επιπλέον, υπάρχει μια επιλογή αλλαγής του κωδικού πρόσβασης του χρήστη, που παρέχεται με ένα εικονίδιο επεξεργασίας για ευκολία.

Η διεπαφή χρήστη οργανώνει περαιτέρω πληροφορίες σε διάφορες ενότητες όπως προτιμήσεις ειδοποιήσεων, προτιμήσεις γλώσσας, συνδεδεμένους λογαριασμούς, ιστορικό δραστηριοτήτων, ρυθμίσεις απορρήτου, σχόλια και αξιολογήσεις, επιλογή αποσύνδεσης και απενεργοποίηση/ διαγραφή λογαριασμού. Κάθε ενότητα φέρει ετικέτα και μπορεί να αξιοποιηθεί για την εκτέλεση συγκεκριμένων ενεργειών ή την πλοήγηση σε σχετικές λειτουργίες.

Για τη διαχείριση εικόνων προφίλ, οι χρήστες μπορούν να ανεβάσουν μια εικόνα πατώντας το “Change Profile Picture”. Ο κώδικας χρησιμοποιεί τη συλλογή εικόνων της συσκευής για την επιλογή εικόνας και η επιλεγμένη εικόνα μεταφορτώνεται στην αποθήκευση Firebase.

Η λειτουργία αποσύνδεσης είναι παρούσα, επιτρέποντας στους χρήστες να αποσυνδεθούν από τους λογαριασμούς τους. Η ενότητα Απενεργοποίηση/Διαγραφή λογαριασμού προσφέρει μια επιλογή απενεργοποίησης ή διαγραφής του λογαριασμού του χρήστη, ζητώντας ένα παράθυρο διαλόγου επιβεβαίωσης πριν από την εκτέλεση.

Κατά τη διάρκεια των διαδικασιών φόρτωσης δεδομένων, όπως η λήψη πληροφοριών χρήστη από το Firebase Firestore, εμφανίζεται ένας δείκτης κυκλικής προόδου, ο οποίος υποδεικνύει τη συνεχή ανάκτηση δεδομένων.

2.7.4 Menu Page

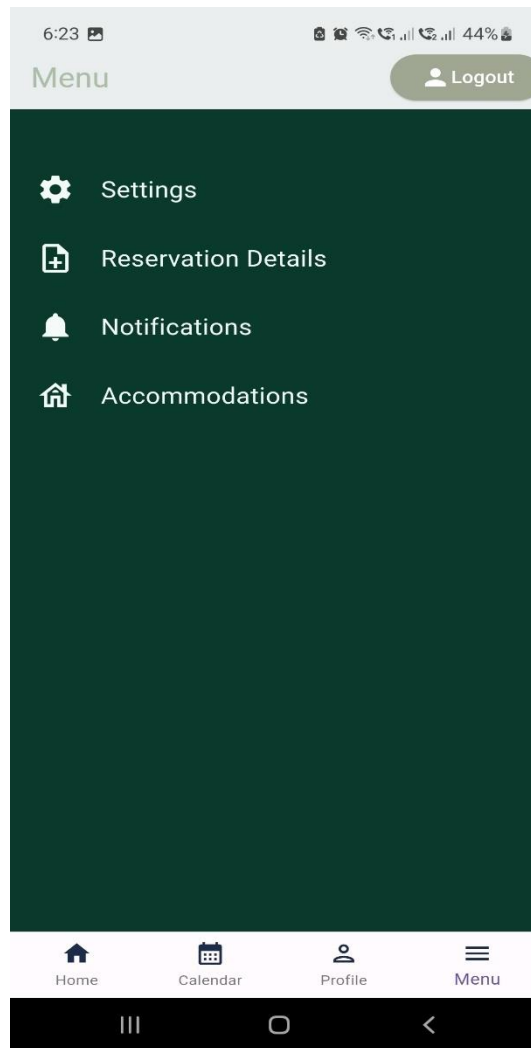
Η σελίδα Μενού (Menu Page, Εικόνα 11) της εφαρμογής παρουσιάζει μια πληθώρα επιλογών για τον χρήστη. Καθώς οι χρήστες έχουν πρόσβαση σε αυτήν την οθόνη, συναντούν ένα καλά οργανωμένο μενού, κάθε επιλογή διακοσμημένη με διακριτικά εικονίδια και σαφείς τίτλους.

Στο επάνω μέρος της οθόνης, οι χρήστες παρατηρούν μια σειρά επιλογών που έχουν σχεδιαστεί για τον εξορθολογισμό της εμπειρίας τους μέσα στην εφαρμογή. Το εικονίδιο "Settings", που μοιάζει με γρανάζι, καλεί τους χρήστες να τελειοποιήσουν τις προτιμήσεις και τις διαμορφώσεις τους, πηγαίνοντας τους στην οθόνη καταχώρισης iCal. Με την επιλογή αυτής της επιλογής, πραγματοποιείται μια μετάβαση, καθοδηγώντας ομαλά τους χρήστες σε μια ειδική οθόνη ρυθμίσεων όπου μπορούν να εξατομικεύσουν διάφορες πτυχές της αλληλεπίδρασής τους με την εφαρμογή.

Κάτω από την επιλογή Ρυθμίσεις, το εικονίδιο "Reservation Details", που θυμίζει ένα περιγραφόμενο χαρτί, τραβάει την προσοχή των χρηστών. Ένα απλό πάτημα σε αυτό το εικονίδιο οδηγεί απρόσκοπτα τους χρήστες στην οθόνη με τις λεπτομέρειες των κρατήσεων. Εδώ, μπορούν να εξερευνήσουν και να διαχειριστούν πληροφορίες που σχετίζονται με τις κρατήσεις τους, εξασφαλίζοντας μια απλή και διαισθητική εμπειρία.

Πιο κάτω στο μενού, το εικονίδιο "Notifications", σε σχήμα καμπάνας, παρουσιάζει τις ειδοποιήσεις των χρηστών. Κατά την επιλογή, οι χρήστες μεταφέρονται αβίαστα σε μια οθόνη ειδοποιήσεων όπου μπορούν να αλληλεπιδράσουν με ειδοποιήσεις και ενημερώσεις σχετικές με τις δραστηριότητές τους εντός της εφαρμογής.

Τέλος, το εικονίδιο "Accommodations", που απεικονίζει ένα περίγραμμα σπιτιού, παρουσιάζει όλα τα διαθέσιμα καταλύματα. Οι χρήστες πλοηγούνται ομαλά σε μια οθόνη αφιερωμένη στα καταλύματα, παρέχοντας έναν βολικό τρόπο περιήγησης και επιλογής των επιθυμητών χώρων διαβίωσης.



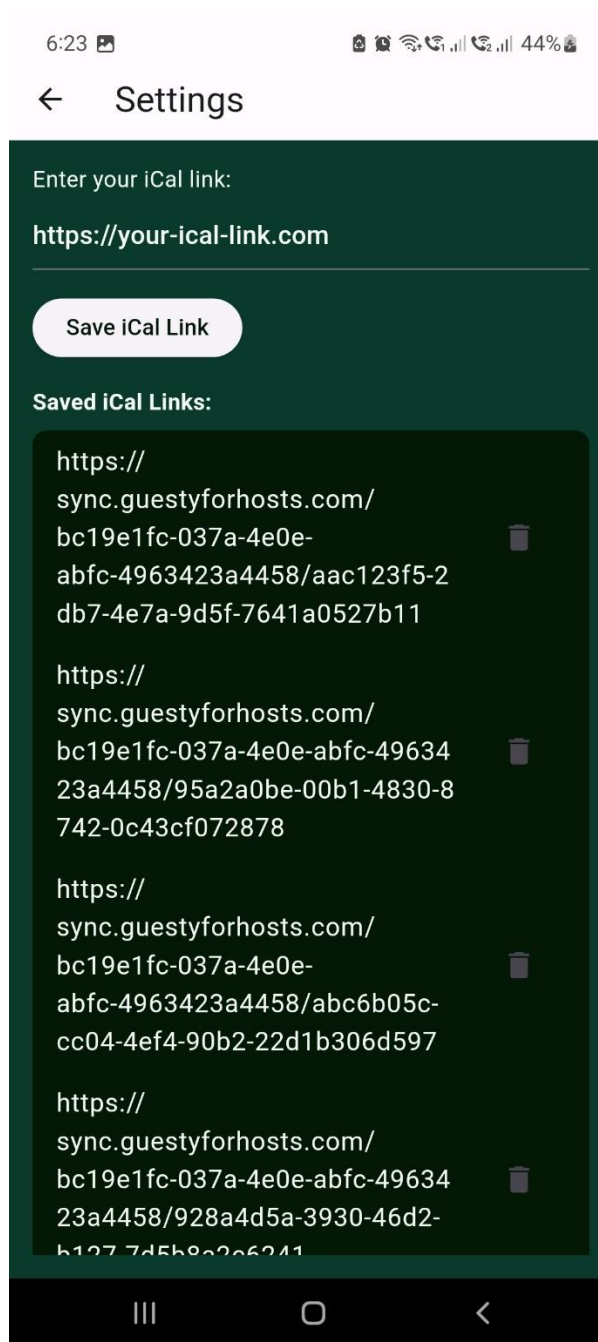
Εικόνα 11: Menu Page

Σε κάθε στοιχείο μενού, ένας προσεκτικά υλοποιημένος ανιχνευτής επιτρέπει στους χρήστες να αλληλεπιδρούν με τις επιλογές. Ο συνδυασμός αναγνωρίσιμων εικονιδίων και συνοπτικών τίτλων διευκολύνει την διαισθητική πλοήγηση, επιτρέποντας στους χρήστες να εξερευνούν και να αλληλεπιδρούν εύκολα με διαφορετικές πτυχές της εφαρμογής. Συνολικά, αυτή η προσεκτικά σχεδιασμένη σελίδα Μενού βελτιώνει την εμπειρία του χρήστη παρέχοντας μια οπτικά ελκυστική και φιλική προς το χρήστη πύλη σε διάφορες λειτουργίες εντός της εφαρμογής.

2.7.5 Settings Page

Στη σελίδα ρυθμίσεων (Settings Page, Εικόνα 12) της εφαρμογής, η κύρια εστίαση κατευθύνεται προς τις αλληλεπιδράσεις των χρηστών και τις διαμορφώσεις που σχετίζονται με τις συνδέσεις iCal.

Στο επάνω μέρος της σελίδας, μια γνωστή γραμμή εφαρμογών εμφανίζει τον τίτλο "Settings", παρέχοντας στους χρήστες μια σαφή ένδειξη του τρέχοντος περιβάλλοντος. Ακριβώς κάτω από τη γραμμή εφαρμογών, οι χρήστες συναντούν μια συνοπτική οδηγία, προτρέποντάς τους να εισέλθουν στον iCal σύνδεσμο τους. Ένα πεδίο κειμένου, συνοδευόμενο από μια λεπτή υπόδειξη, περιμένει την εισαγωγή τους, διατηρώντας μια φιλική προς το χρήστη προσέγγιση.



Εικόνα 12: Settings Page

Αμέσως κάτω από το πεδίο κειμένου, ένα ανυψωμένο κουμπί με την ένδειξη "Save iCal Link" καλεί τους χρήστες να αποθηκεύουν τους συνδέσμους iCal για μελλοντική αναφορά. Μόλις πατηθεί το κουμπί, ο καταχωρημένος σύνδεσμος αποθηκεύεται με ασφάλεια στο Firestore, εξασφαλίζοντας απρόσκοπτο συγχρονισμό και προσβασιμότητα.

Καθώς οι χρήστες μετακινούνται προς τα κάτω, μια ενότητα με την ένδειξη "Saved iCal Links" παρουσιάζει μια δυναμική λίστα προηγούμενων αποθηκευμένων συνδέσμων. Κάθε σύνδεσμος παρουσιάζεται σε μια τακτοποιημένη μορφή, με ένα παρακείμενο εικονίδιο διαγραφής, επιτρέποντας στους χρήστες να διαχειρίζονται αποτελεσματικά τους

αποθηκευμένους συνδέσμους iCal. Το χρώμα του φόντου διακρίνει διακριτικά αυτό το τμήμα, συμβάλλοντας στη συνολική αισθητική της σελίδας.

Οι ενημερώσεις σε πραγματικό χρόνο, χάρη στην εφαρμογή ενός StreamBuilder, διασφαλίζει ότι οι χρήστες ενημερώνονται αμέσως για τυχόν τροποποιήσεις στους αποθηκευμένους συνδέσμους iCal. Σε περίπτωση που δεν αποθηκεύονται σύνδεσμοι, ένα φιλικό προς το χρήστη μήνυμα ενημερώνει απαλά τους χρήστες για την απουσία αποθηκευμένων συνδέσμων.

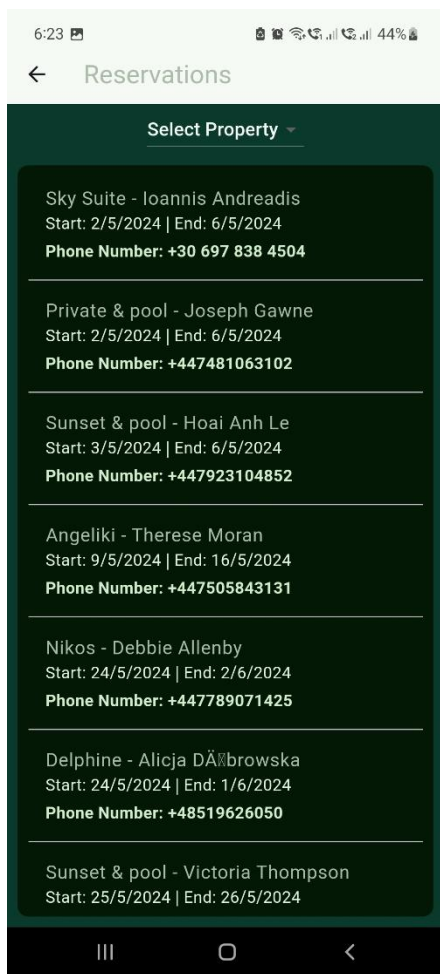
Ο διαισθητικός σχεδιασμός αυτής της σελίδας ρυθμίσεων διευκολύνει μια ομαλή και συναρπαστική εμπειρία, δίνοντας τη δυνατότητα στους χρήστες να διαχειρίζονται τους iCal συνδέσμους τους μέσα στην εφαρμογή. Είτε εισάγετε έναν νέο σύνδεσμο, είτε αποθηκεύετε είτε διαγράφετε έναν προηγούμενος αποθηκευμένος, η διεπαφή παραμένει συνεπής, παρέχοντας μια συνεκτική και επικεντρωμένη στον χρήστη αλληλεπίδραση.

2.7.6 Reservations Detail Page

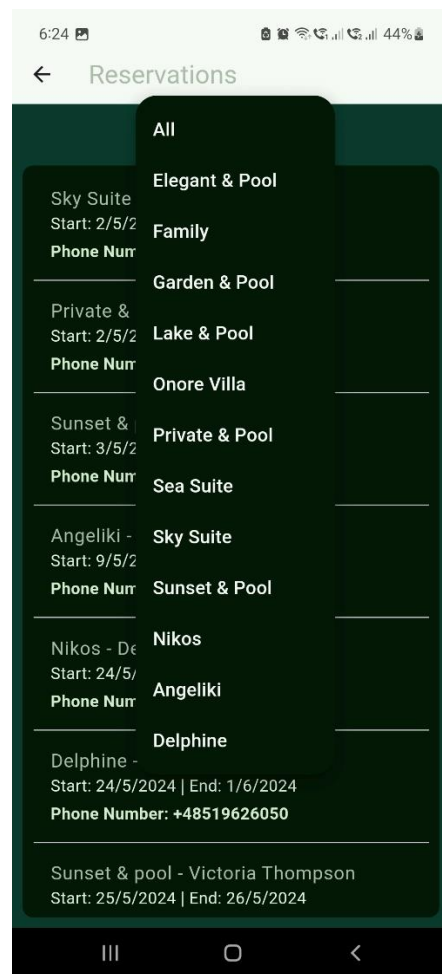
Στη σελίδα "Λεπτομέρειες κράτησης" (Reservation Details, Εικόνα 13), οι χρήστες παρουσιάζονται με μια καλά σχεδιασμένη διεπαφή σχεδιασμένη για τη διαχείριση και την προβολή πληροφοριών κράτησης. Η γραμμή εφαρμογών, φέρει τον τίτλο "κρατήσεις", σηματοδοτώντας το τρέχον πλαίσιο στους χρήστες.

Το επίκεντρο αυτής της σελίδας είναι μια δυναμική λίστα συμβάντων κρατήσεων που ανακτώνται από συνδέσμους iCal. Καθώς οι χρήστες εισέρχονται στη σελίδα, ξεκινά μια διαδικασία αυτόματης ανάκτησης, συμπληρώνοντας τη λίστα με σχετικές λεπτομέρειες κράτησης. Οι ενημερώσεις σε πραγματικό χρόνο διασφαλίζουν ότι οι χρήστες λαμβάνουν τις πιο πρόσφατες πληροφορίες σχετικά με τις κρατήσεις τους.

Για να βελτιωθεί η αλληλεπίδραση των χρηστών, παρέχεται ένα αναπτυσσόμενο μενού, επιτρέποντας στους χρήστες να φιλτράρουν τις κρατήσεις με βάση το όνομα του καταλύματος (Εικόνα 14). Οι χρήστες μπορούν εύκολα να επιλέξουν μια συγκεκριμένη ιδιότητα από το αναπτυσσόμενο μενού. Το αναπτυσσόμενο μενού, διακοσμημένο με σκούρο πράσινο φόντο, ενσωματώνεται άψογα με τη συνολική αισθητική.



Εικόνα 13: Reservation Details Page



Εικόνα 14: Dropdown list

Προς τα κάτω, υπάρχει μια λίστα κρατήσεων. Τα στοιχεία της λίστας έχουν σχεδιαστεί σχολαστικά, εμφανίζοντας κρίσιμες λεπτομέρειες όπως το όνομα του καταλύματος, όνομα επισκέπτη, ημερομηνίες έναρξης και λήξης, και έναν αριθμό τηλεφώνου επικοινωνίας. Οι πληροφορίες είναι καλά οργανωμένες, συμβάλλοντας σε μια φιλική προς το χρήστη διεπαφή.

Για να εξασφαλιστεί μια βελτιωμένη εμπειρία, οι κρατήσεις ταξινομούνται και φιλτράρονται δυναμικά με βάση τις ημερομηνίες λήξης τους. Οι προηγούμενες κρατήσεις φιλτράρονται αυτόματα, διατηρώντας την εστίαση στις επερχόμενες εκδηλώσεις. Ο κατάλογος είναι επίσης ταξινομημένος με βάση τις ημερομηνίες έναρξης των κρατήσεων.

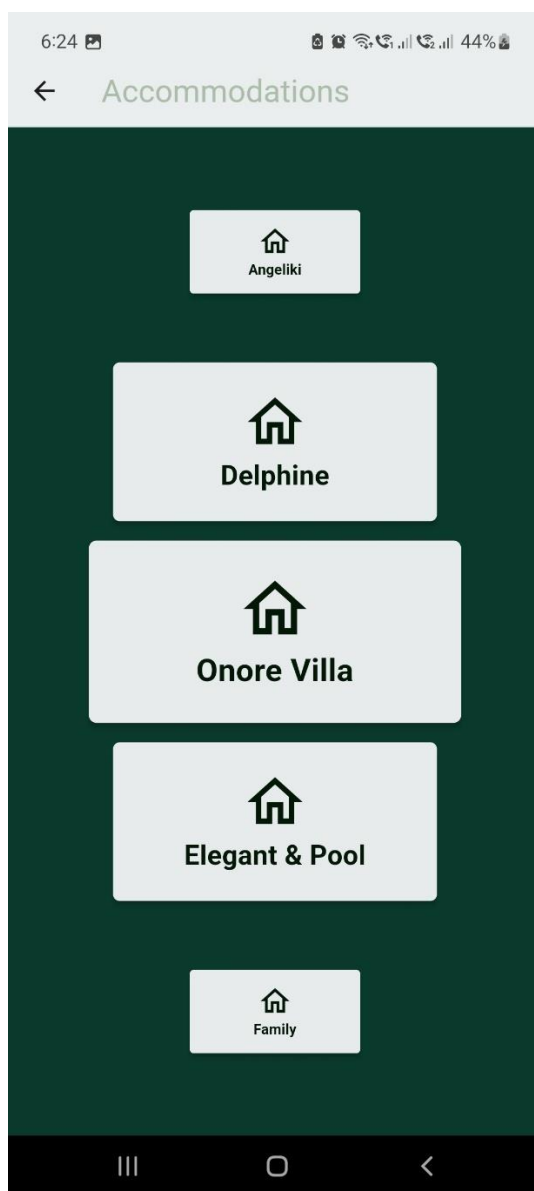
Στο πνεύμα της διαφάνειας, εμφανίζεται ένας δείκτης φόρτωσης κατά την ανάκτηση δεδομένων, παρέχοντας ανατροφοδότηση στους χρήστες σχετικά με τη συνεχιζόμενη διαδικασία. Υπάρχουν μηχανισμοί χειρισμού σφαλμάτων για τη διαχείριση με χάρη τυχόν απρόβλεπτων ζητημάτων, εξασφαλίζοντας μια ανθεκτική εμπειρία χρήστη.

Συνοπτικά, η σελίδα "Reservation Details" συνδυάζει άψογα την αισθητική με τη λειτουργικότητα, επιτρέποντας στους χρήστες να διαχειρίζονται και να εξερευνούν αβίαστα τις επερχόμενες κρατήσεις τους. Είτε φιλτράρισμα ανά τύπο ιδιότητας, κύλιση στη λίστα ή αναθεώρηση συγκεκριμένων λεπτομερειών, η διεπαφή παραμένει συνεπής και επικεντρωμένη στον χρήστη σε όλη την αλληλεπίδραση.

2.7.7 Accommodations Page

Στη σελίδα "καταλύματα" (Accommodations Page, Εικόνα 15), οι χρήστες καλωσορίζονται από μια αισθητικά ευχάριστη και διαισθητικά σχεδιασμένη διεπαφή που παρέχει μια απρόσκοπτη εμπειρία για την Εξερεύνηση διαφορετικών επιλογών δωματίου. Η γραμμή εφαρμογών, που ξεχωρίζει με ανοιχτό πράσινο χρώμα, διαθέτει τον τίτλο "Accommodations", δίνοντας στους χρήστες μια σαφή κατανόηση του περιεχομένου με το οποίο πρόκειται να αλληλεπιδράσουν.

Το κεντρικό κομμάτι αυτής της σελίδας είναι ένα δυναμικό καρουσέλ που τροφοδοτείται από το πακέτο Carousel Slider. Κάθε στοιχείο καρουσέλ αντιπροσωπεύει μια ξεχωριστή επιλογή δωματίου ή διαμονής. Οι χρήστες μπορούν εύκολα να μετακινηθούν μέσα στο καρουσέλ ή να του επιτρέψουν να παίζει αυτόματα, αποκαλύπτοντας μια επιμελημένη επιλογή καταλυμάτων. Το καρουζέλ είναι καλά ρυθμισμένο, με αρχική κατακόρυφη διάταξη, αυτόματη λειτουργία αναπαραγωγής και σχεδιασμό που ανταποκρίνεται σε διαφορετικά μεγέθη οθόνης.



Εικόνα 15: Accommodations Page

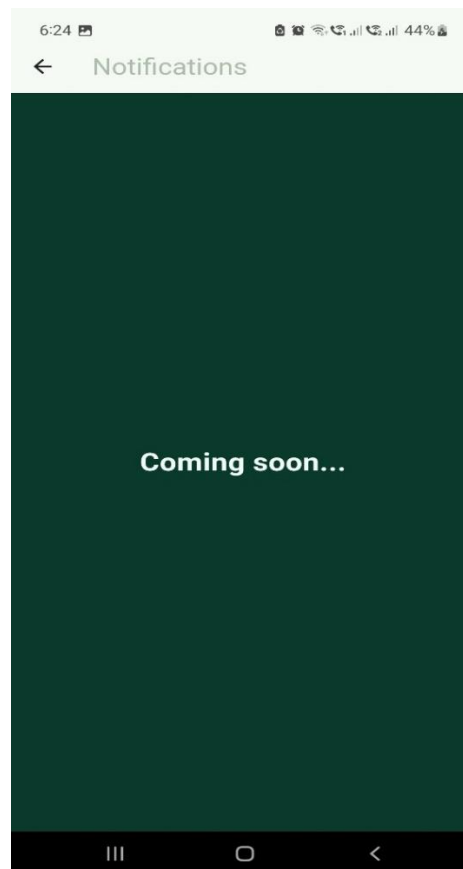
Κάθε δωμάτιο που εμφανίζεται στο καρουζέλ είναι ενθυλακωμένο μέσα σε μια όμορφα σχεδιασμένη κάρτα. Η κάρτα, που χαρακτηρίζεται από ανοιχτό πράσινο φόντο, παρουσιάζει βασικές λεπτομέρειες. Ένα εικονίδιο που αντιπροσωπεύει τον τύπο δωματίου εμφανίζεται εμφανώς, συνοδευόμενο από τον τίτλο του δωματίου με έντονη και ευανάγνωστη γραμματοσειρά.

Για να βελτιωθεί η αφοσίωση των χρηστών, κάθε κάρτα δωματίου είναι διαδραστική. Οι χρήστες μπορούν να πατήσουν σε οποιαδήποτε κάρτα δωματίου για να έχουν πρόσβαση σε λεπτομερείς πληροφορίες σχετικά με το αντίστοιχο κατάλυμα. Αυτός ο μηχανισμός επιτρέπει στους χρήστες να εμβαθύνουν σε συγκεκριμένες λεπτομέρειες δωματίου χωρίς κόπο.

Συνοπτικά, η σελίδα "Accommodations" είναι μια φιλική προς το χρήστη πύλη για να δει ποια δωμάτια είναι διαθέσιμα για κράτηση. Η διεπαφή με καρουσέλ, σε συνδυασμό με διαδραστικές κάρτες, συμβάλλει σε μια συναρπαστική και ευχάριστη εμπειρία χρήστη. Είτε περνώντας μέσα από το καρουσέλ είτε πατώντας σε ένα συγκεκριμένο δωμάτιο για περισσότερες πληροφορίες, οι χρήστες μπορούν να πλοηγηθούν αβίαστα στην εφαρμογή για να ανακαλύψουν την ιδανική διαμονή τους.

2.7.8 Notifications Page

Η σελίδα ειδοποιήσεων (Notification Page, Εικόνα 16), δεν έχει ολοκληρωθεί ακόμα. Σκοπός είναι να συνδεθεί με το Firebase Cloud Messaging και να εμφανίζει ειδοποιήσεις στους χρήστες για καινούργιες κρατήσεις, σημερινές κρατήσεις καθώς και ειδοποιήσεις σχετικές με το λογαριασμό του χρήστη.



Εικόνα 16: Notification Page

2.8. Δοκιμές

Καθ' όλη τη διάρκεια της υλοποίησης, ένα ισχυρό πλαίσιο δοκιμών ενσωματώθηκε συστηματικά, αποτελώντας θεμελιώδη πυλώνα στον αναπτυξιακό κύκλο ζωής της εφαρμογής. Σε κάθε κεντρικό στάδιο της εξέλιξης της εφαρμογής, διεξήχθη μια ολοκληρωμένη σειρά δοκιμών.

Το σχήμα δοκιμών περιλάμβανε μια πολύπλευρη προσέγγιση, αντιμετωπίζοντας ποικίλες πτυχές της αρχιτεκτονικής της εφαρμογής. Εξετάζονταν η λειτουργικότητα των μεμονωμένων εξαρτημάτων, διασφαλίζοντας ότι το καθένα εκτελούσε τις καθορισμένες εργασίες του με ακρίβεια και αξιοπιστία. Αυτή η κοκκώδης εξέταση βοήθησε στον εντοπισμό και την αποκατάσταση τυχόν πιθανών ανωμαλιών ή ασυνέπειων στη βάση κώδικα.

Ταυτόχρονα, αξιολογούταν η απρόσκοπτη συνεργασία και αλληλεπίδραση μεταξύ διαφορετικών στοιχείων εντός της εφαρμογής. Αυτό προσπάθησε να αποκαλύψει τυχόν αλληλεξαρτήσεις ή ζητήματα επικοινωνίας που θα μπορούσαν να προκύψουν όταν συγκλίνουν διακριτά στοιχεία του λογισμικού.

Η δοκιμή διεπαφής χρήστη (UI) αποτέλεσε αναπόσπαστο μέρος της διαδικασίας επικύρωσης, εστιάζοντας στην οπτική αισθητική, την ανταπόκριση και τη συνολική εμπειρία του χρήστη. Με την προσομοίωση αλληλεπιδράσεων χρηστών, οι δοκιμές διεπαφής χρήστη παρείχαν πολύτιμες πληροφορίες σχετικά με τη διεπαφή χρήστη της εφαρμογής, αποκαλύπτοντας τυχόν αποκλίσεις διάταξης, εμπόδια πλοήγησης ή γραφικές ανωμαλίες που ενδέχεται να εμποδίσουν την αλληλεπίδραση του χρήστη με το λογισμικό.

Επιπλέον, εφαρμόστηκε συστηματική δοκιμή παλινδρόμησης για να διασφαλιστεί ότι νέα χαρακτηριστικά ή βελτιώσεις δεν θα διατάρασσαν την υπάρχουσα λειτουργικότητα της εφαρμογής. Αυτή η προληπτική προσέγγιση εμπόδισε την ακούσια εισαγωγή ελαττωμάτων καθώς η εφαρμογή υπέστη επαναληπτικές βελτιώσεις.

Οι προσπάθειες δοκιμών δεν περιορίστηκαν μόνο στη φάση ανάπτυξης. Τα σενάρια πραγματικού κόσμου και τα πρότυπα χρήσης προσομοιώθηκαν για να μετρήσουν την απόδοση της εφαρμογής υπό διαφορετικές συνθήκες, επιτρέποντας τον εντοπισμό και τη διόρθωση τυχόν προβλημάτων που ενδέχεται να εμφανιστούν σε δυναμικά περιβάλλοντα.

Η δέσμευση για δοκιμές επεκτάθηκε πέρα από τον απλό εντοπισμό σφαλμάτων * περιλάμβανε μια προληπτική στάση για την προληπτική επίλυση ζητημάτων. Όταν εντοπίστηκαν αποκλίσεις, ξεκίνησε μια σχολαστική διαδικασία εντοπισμού σφαλμάτων και διορθωτικών μέτρων, διασφαλίζοντας ότι κάθε ζήτημα αντιμετωπίστηκε συστηματικά και ότι η βάση κώδικα βελτιώθηκε για να ενισχυθεί η συνολική σταθερότητα.

Στην ουσία, το πρότυπο δοκιμών που υιοθετήθηκε κατά την ανάπτυξη της εφαρμογής ήταν μια σχολαστική και συστηματική προσπάθεια, υποστηρίζοντας τις αρχές της διασφάλισης της ποιότητας. Η επαναληπτική φύση των δοκιμών σε κάθε αναπτυξιακή συγκυρία υπογράμμισε τη δέσμευση για την παροχή μιας ισχυρής, αξιόπιστης και επικεντρωμένης στον χρήστη εφαρμογής, ελαχιστοποιώντας την πιθανότητα προβλημάτων και ενισχύοντας την απόδοση του λογισμικού σε σενάρια πραγματικού κόσμου.

2.9 Ολοκλήρωση της εφαρμογής

Η φάση υλοποίησης παρέδωσε με επιτυχία μια ισχυρή εφαρμογή για κινητά, ενσωματώνοντας τη διαχείριση λογαριασμού χρήστη, την ασφαλή αποθήκευση iCal link, την εμφάνιση συμβάντων ημερολογίου και την απρόσκοπτη ενσωμάτωση με το Firebase. Η αξιοποίηση των τεχνολογιών Flutter και Firebase συνέβαλε στη δημιουργία μιας ανταποκρινόμενης και φιλικής προς το χρήστη πλατφόρμας για αποτελεσματική διαχείριση ημερολογίου.

Κεφάλαιο 3: Μελλοντικές Βελτιώσεις

Καθώς η εφαρμογή εξελίσσεται, υπάρχουν αρκετοί τομείς όπου μελλοντικές βελτιώσεις και βελτιστοποιήσεις μπορούν να εξεταστούν για περαιτέρω βελτίωση και επέκταση των δυνατοτήτων της. Τα ακόλουθα περιγράφουν πιθανές οδούς βελτίωσης και χαρακτηριστικά που θα μπορούσαν να διερευνηθούν σε μελλοντικές επαναλήψεις:

3.1 Βελτιωμένη Διεπαφή Χρήστη

Βελτίωση διεπαφής χρήστη: Η βελτίωση του σχεδιασμού της εφαρμογής συνεπάγεται σε στρατηγική εστίαση στην εμπειρία και την αισθητική του χρήστη. Μια κρίσιμη πτυχή είναι η κατανόηση του κοινού-στόχου μέσω ολοκληρωμένης έρευνας χρηστών, διασφαλίζοντας ότι ο σχεδιασμός ευθυγραμμίζεται με τις ανάγκες και τις προτιμήσεις τους. Είναι σημαντικό να δημιουργηθεί ένας σχεδιασμός με επίκεντρο τον χρήστη που απλοποιεί την πλοήγηση, καθιστώντας το διαισθητικό και φιλικό προς το χρήστη.

Η συνέπεια είναι το κλειδί στο σχεδιασμό, από στοιχεία επωνυμίας όπως χρωματικά σχέδια και γραμματοσειρές έως τη διατήρηση μιας καθαρής και οργανωμένης διεπαφής με αποτελεσματική χρήση του κενού χώρου. Η τυπογραφία παίζει σημαντικό ρόλο στην αναγνωσιμότητα, οπότε η επιλογή ευανάγνωστων γραμματοσειρών και η διατήρηση της συνέπειας σε όλη την εφαρμογή συμβάλλει σε μια καθαρή εμφάνιση. Μια συνεκτική παλέτα χρωμάτων, στρατηγικά εφαρμοσμένη, όχι μόνο ευθυγραμμίζεται με την επωνυμία της εφαρμογής αλλά και καθοδηγεί τους χρήστες μέσω της διεπαφής.

Τα διαδραστικά στοιχεία, όπως τα κουμπιά, θα πρέπει να είναι οπτικά διακριτά, παρέχοντας σαφή ανατροφοδότηση για τις ενέργειες των χρηστών. Οι μηχανισμοί ανατροφοδότησης, όπως τα μηνύματα σφάλματος και οι δείκτες φόρτωσης, ενημερώνουν τους χρήστες σχετικά με την κατάσταση των αλληλεπιδράσεών τους. Ο αποκριτικός σχεδιασμός (Responsive Design) είναι ζωτικής σημασίας για την προσαρμογή διαφόρων μεγεθών οθόνης, εξασφαλίζοντας μια απρόσκοπτη εμπειρία σε όλες τις συσκευές.

Αξιοποίηση των αλληλεπιδράσεων που βασίζονται σε χειρονομίες για μια πιο συναρπαστική εμπειρία, συμπληρωμένη από σαφή και αναγνωρίσιμη εικονογραφία. Μια ομαλή διαδικασία ενσωμάτωσης καθοδηγεί τους χρήστες μέσω των λειτουργιών της εφαρμογής.

Προσαρμογή θέματος: εφαρμογή επιλογών προσαρμογής θέματος, επιτρέποντας στους χρήστες να εξατομικεύσουν την εμφάνιση της εφαρμογής σύμφωνα με τις προτιμήσεις τους.

3.2 Χαρακτηριστικά Ημερολογίου

Σε μελλοντικές βελτιώσεις για την προβολή ημερολογίου, η εφαρμογή επιλογών εύρους για ημερομηνίες μπορεί να βελτιώσει σημαντικά την εμπειρία του χρήστη. Η δυνατότητα των χρηστών να επιλέγουν συγκεκριμένα εύρη ημερομηνιών επιτρέπει έναν πιο διαισθητικό και αποτελεσματικό τρόπο προβολής συμβάντων που διαρκούν πολλές ημέρες. Αυτή η λειτουργία διευκολύνει τους χρήστες που θέλουν να επικεντρωθούν σε ένα ευρύτερο χρονικό πλαίσιο, παρέχοντας μια ολοκληρωμένη επισκόπηση των συμβάντων εντός αυτού του επιλεγμένου εύρους.

Μια άλλη πολύτιμη προσθήκη στην προβολή ημερολογίου θα μπορούσε να είναι η ενσωμάτωση λειτουργιών λήψης σημειώσεων. Το να επιτρέπεται στους χρήστες να επισυνάπτουν σημειώσεις ή σχόλια σε συγκεκριμένα συμβάντα παρέχει έναν βολικό και συγκεντρωτικό τρόπο για τη λήψη πρόσθετων λεπτομερειών ή πληροφοριών που σχετίζονται με κάθε καταχώρηση ημερολογίου. Αυτή η λειτουργία ενισχύει τη χρησιμότητα της εφαρμογής, καθιστώντας την όχι μόνο ένα εργαλείο προγραμματισμού αλλά και μια ευέλικτη πλατφόρμα για την οργάνωση σκέψεων και πληροφοριών.

Επιπλέον, η εξέταση επιλογών για προσαρμόσιμες προβολές μπορεί να καλύψει διαφορετικές προτιμήσεις χρηστών. Η παροχή διαφορετικών τρόπων εμφάνισης ημερολογίου, όπως προβολές ημέρας, εβδομάδας και μήνα, επιτρέπει στους χρήστες να προσαρμόζουν την εμπειρία τους με βάση τις συγκεκριμένες ανάγκες τους. Αυτή η ευελιξία διασφαλίζει ότι το ημερολόγιο προσαρμόζεται σε διαφορετικά σενάρια χρήσης, φιλοξενώντας τόσο εκείνους που προτιμούν μια λεπτομερή ημερήσια προβολή όσο και εκείνους που προτιμούν μια πιο ολοκληρωμένη μηνιαία επισκόπηση.

Για την βελτίωση της εμπειρίας των χρηστών, η διερεύνηση της ενσωμάτωσης με δημοφιλή εργαλεία και πλατφόρμες παραγωγικότητας μπορεί να είναι επωφελής. Η απρόσκοπτη συνδεσιμότητα με εφαρμογές λήψης σημειώσεων, πλατφόρμες συνεργασίας ή εργαλεία διαχείρισης εργασιών μπορεί να εμπλουτίσει τη συνολική λειτουργικότητα της προβολής ημερολογίου, παρέχοντας στους χρήστες μια ολιστική λύση για τη διαχείριση του χρόνου και των εργασιών τους.

3.3 Ασφάλεια και έλεγχος ταυτότητας

η εφαρμογή βιομετρικού ελέγχου ταυτότητας για σύνδεση προσθέτει ένα επιπλέον επίπεδο ασφάλειας και ευκολίας χρήστη στην εφαρμογή. Η βιομετρική εξακρίβωση ταυτότητας αξιοποιεί μοναδικά φυσικά ή συμπεριφορικά χαρακτηριστικά, όπως δακτυλικά αποτυπώματα, χαρακτηριστικά προσώπου ή μοτίβα φωνής, για την επαλήθευση της ταυτότητας ενός χρήστη. Με την ενσωμάτωση βιομετρικών ελέγχων, η εφαρμογή ενισχύει την ασφάλεια, καθώς αυτά τα χαρακτηριστικά είναι εγγενώς πιο δύσκολο να αναπαραχθούν ή να πλαστογραφηθούν σε σύγκριση με τους παραδοσιακούς κωδικούς πρόσβασης.

Η διαδικασία βιομετρικής σύνδεσης συνήθως περιλαμβάνει την προτροπή των χρηστών να καταχωρήσουν τα βιομετρικά τους δεδομένα κατά την αρχική ρύθμιση. Μόλις καταχωρηθούν, οι επόμενες συνδέσεις απαιτούν από τους χρήστες να επαληθεύσουν τον εαυτό τους παρέχοντας το βιομετρικό δείγμα. Αυτή η απρόσκοπτη και γρήγορη μέθοδος

ελέγχου ταυτότητας όχι μόνο ενισχύει την ασφάλεια αλλά και βελτιώνει τη συνολική εμπειρία χρήστη εξαλείφοντας την ανάγκη για σύνθετους κωδικούς πρόσβασης.

Εκτός από τον βιομετρικό έλεγχο ταυτότητας, η εφαρμογή ελέγχου ταυτότητας δύο παραγόντων (2FA) ενισχύει περαιτέρω τη διαδικασία σύνδεσης. Το 2FA απαιτεί από τους χρήστες να παρέχουν δύο διαφορετικές μορφές ταυτοποίησης πριν αποκτήσουν πρόσβαση στους λογαριασμούς τους. Αυτό συνήθως περιλαμβάνει κάτι που γνωρίζει ο χρήστης (όπως έναν κωδικό πρόσβασης) και κάτι που έχει ο χρήστης (όπως έναν προσωρινό κωδικό που αποστέλλεται στο εγγεγραμμένο ηλεκτρονικό ταχυδρομείο ή στην κινητή συσκευή του).

Συνδυάζοντας τον βιομετρικό έλεγχο ταυτότητας με το 2FA, η εφαρμογή υιοθετεί μια ισχυρή προσέγγιση ελέγχου ταυτότητας πολλαπλών παραγόντων. Σε αυτό το σενάριο, οι χρήστες όχι μόνο επαληθεύουν την ταυτότητά τους μέσω βιομετρικής σάρωσης, αλλά λαμβάνουν επίσης έναν δευτερεύοντα κωδικό ελέγχου ταυτότητας, προσθέτοντας ένα επιπλέον επίπεδο προστασίας. Αυτή η προσέγγιση μειώνει σημαντικά τον κίνδυνο μη εξουσιοδοτημένης πρόσβασης, καθώς ο συμβιβασμός τόσο των βιομετρικών δεδομένων όσο και του δευτερεύοντος κώδικα ελέγχου ταυτότητας γίνεται πιο δύσκολος για πιθανούς επιτιθέμενους.

Για την αποτελεσματική εφαρμογή αυτών των χαρακτηριστικών, είναι σημαντικό να διασφαλιστούν φιλικές προς το χρήστη διεπαφές για βιομετρική εγγραφή και να κοινοποιηθούν με σαφήνεια τα βήματα που εμπλέκονται στη διαδικασία ελέγχου ταυτότητας δύο παραγόντων. Επιπλέον, η ενσωμάτωση επιλογών για τους χρήστες να διαχειρίζονται και να ανακτούν τις μεθόδους ελέγχου ταυτότητας βελτιώνει τη συνολική εμπειρία και την προσβασιμότητα των χρηστών.

Η τακτική ενημέρωση και βελτιστοποίηση της εφαρμογής βιομετρικών και 2FA με βάση τις βέλτιστες πρακτικές του κλάδου και τις αναδυόμενες τεχνολογίες είναι απαραίτητη για να παραμείνουμε μπροστά από πιθανές απειλές για την ασφάλεια. Αυτή η προληπτική προσέγγιση διασφαλίζει ότι οι μηχανισμοί ελέγχου ταυτότητας παραμένουν ισχυροί και ευθυγραμμισμένοι με τα εξελισσόμενα πρότυπα ασφαλείας.

3.4 Βελτιστοποίηση Απόδοσης

Στην επίδιωξη της βελτιστοποίησης της αποτελεσματικότητας του κώδικα και της βελτίωσης της ταχύτητας φόρτωσης, υπάρχουν αρκετές στρατηγικές που μπορούν να εφαρμοστούν στη βάση κώδικα της εφαρμογής.

Για την ανάκτηση γεγονότων, η χρήση τεχνικών ασύγχρονου προγραμματισμού, μπορεί να βελτιώσει σημαντικά την ανταπόκριση της εφαρμογής. Η χρήση ασύγχρονων μεθόδων επιτρέπει την εκτέλεση χωρίς αποκλεισμό, διασφαλίζοντας ότι η διεπαφή χρήστη παραμένει ανταποκρινόμενη κατά τη λήψη συμβάντων. Επιπλέον, με εφαρμογής μηχανισμών προσωρινής αποθήκευσης για την αποθήκευση γεγονότων που ανακτήθηκαν προηγουμένως τοπικά, μειώνεται η ανάγκη για επαναλαμβανόμενα αιτήματα δικτύου και βελτιώνεται τη συνολική ταχύτητα φόρτωσης.

Για την βελτίωση περαιτέρω την ταχύτητα φόρτωσης στις διαδικασίες μεταφόρτωσης και λήψης εικόνων, μπορεί να ενσωματωθούν τεχνικές συμπίεσης εικόνας. Η συμπίεση εικόνων πριν από τη μεταφόρτωσή τους στο διακομιστή μειώνει τα μεγέθη των αρχείων, οδηγώντας

σε ταχύτερους χρόνους μεταφόρτωσης. Από την πλευρά του χρήστη, η εφαρμογή ενός μηχανισμού προσωρινής αποθήκευσης για τις ληφθείσες εικόνες μπορεί να βελτιώσει την ταχύτητα φόρτωσης κατά τις επόμενες προβολές, καθώς οι αποθηκευμένες εικόνες μπορούν να ανακτηθούν γρήγορα χωρίς πρόσθετα αιτήματα δικτύου.

Αντί να φορτώνονται όλες τις εικόνες ταυτόχρονα, μπορούν να φορτωθούν δυναμικά καθώς έρχονται στην προβολή του χρήστη. Αυτό αποτρέπει την περιττή κατανάλωση πόρων και επιταχύνει την αρχική απόδοση στοιχείων διεπαφής χρήστη.

Η φόρτωση των κρατήσεων σε μικρότερα, διαχειρίσιμα κομμάτια και όχι όλα ταυτόχρονα ενισχύει την ανταπόκριση της εφαρμογής και παρέχει μια ομαλότερη εμπειρία χρήστη.

3.5 Πρόσθετες Ενσωματώσεις

Εισαγωγές εξωτερικών ημερολογίων θα αποτελούσε σημαντική πρόοδο στη λειτουργικότητα της εφαρμογής. Η δυνατότητα στους χρήστες να εισάγουν ημερολόγια από εξωτερικές πηγές, όπως δημοφιλείς πλατφόρμες ημερολογίου ή μορφές αρχείων, βελτιώνει την ευκολία και τη λειτουργικότητα. Αυτή η δυνατότητα θα επέτρεπε στους χρήστες να ενοποιήσουν τα χρονοδιαγράμματα τους χωρίς κόπο, προωθώντας μια ενοποιημένη και συγχρονισμένη εμπειρία σε διαφορετικές πλατφόρμες.

Επιπλέον, η επέκταση της συμβατότητας υποστηρίζοντας διάφορες δομές iCal μπορεί να διευρύνει την εμβέλεια και τη χρησιμότητα της εφαρμογής. Καθώς διαφορετικές εφαρμογές και υπηρεσίες ενδέχεται να συμμορφώνονται με διαφορετικές μορφές iCal, η ευελιξία στην ανάγνωση και την ερμηνεία διαφορετικών δομών iCal εξασφαλίζει ολοκληρωμένη συμβατότητα, διότι πολλές εταιρίες όπως η Airbnb και Booking, έχουν διαφορετικά δομημένα τα iCal τους.

Εκτός από τις εξωτερικές Εισαγωγές ημερολογίου και τη διαφοροποιημένη υποστήριξη iCal structure, η βελτιστοποίηση της απόδοσης και της ανταπόκρισης της εφαρμογής παραμένει μια κρίσιμη πτυχή των μελλοντικών βελτιώσεων. Οι βελτιώσεις στην ταχύτητα επεξεργασίας δεδομένων, ο αποτελεσματικός χειρισμός μεγάλων συνόλων δεδομένων και η ελαχιστοποίηση της καθυστέρησης συμβάλλουν σε μια ομαλότερη εμπειρία χρήστη, ειδικά όταν ασχολούνται με εκτεταμένες καταχωρήσεις ημερολογίου ή εξωτερικές Εισαγωγές.

3.6 Προσβασιμότητα

Η διαφοροποίηση των μεθόδων ελέγχου ταυτότητας για τις διαδικασίες σύνδεσης και εγγραφής μπορεί να βελτιώσει σημαντικά την προσβασιμότητα και την εμπειρία χρήστη της εφαρμογής. Μια πιθανή βελτίωση είναι η εισαγωγή πολλαπλών επιλογών ελέγχου ταυτότητας, παρέχοντας στους χρήστες ευελιξία και ικανοποιώντας τις προτιμήσεις τους.

Η υποστήριξη διαφόρων μεθόδων σύνδεσης, όπως ο παραδοσιακός έλεγχος ταυτότητας μέσω email/κωδικού πρόσβασης, οι συνδέσεις κοινωνικών μέσων (π.χ. χρήση διαπιστευτηρίων Google, Facebook ή Twitter), μπορεί να διευρύνει τη βάση χρηστών και να εξορθολογίσει τη διαδικασία ενσωμάτωσης. Κάθε μέθοδος ελέγχου ταυτότητας προσφέρει ξεχωριστά πλεονεκτήματα, με ορισμένους χρήστες να προτιμούν την ευκολία των κοινωνικών συνδέσεων και άλλοι να δίνουν προτεραιότητα στην ασφάλεια των παραδοσιακών ρυθμίσεων ηλεκτρονικού ταχυδρομείου/κωδικού πρόσβασης.

Η ενσωμάτωση των συνδέσεων κοινωνικών μέσων επιτρέπει στους χρήστες να εγγραφούν ή να συνδεθούν χρησιμοποιώντας τους υπάρχοντες λογαριασμούς τους σε δημοφιλείς πλατφόρμες, απλοποιώντας τη διαδικασία εγγραφής και εξαλείφοντας την ανάγκη να θυμούνται πρόσθετα διαπιστευτήρια σύνδεσης.

Για να εξασφαλιστεί μια απρόσκοπτη μετάβαση και φιλική προς το χρήστη εμπειρία, η εφαρμογή θα πρέπει να παρέχει σαφείς οδηγίες κατά τη διάρκεια των διαδικασιών εγγραφής και σύνδεσης. Τα στοιχεία διεπαφής χρήστη θα πρέπει να σχεδιάζονται διαισθητικά για να προσαρμόζονται σε διαφορετικές μεθόδους ελέγχου ταυτότητας, διατηρώντας τη συνέπεια και τη σαφήνεια σε όλη την εφαρμογή.

Είναι σημαντικό να τονιστεί η σημασία του απορρήτου των χρηστών και της ασφάλειας των δεδομένων κατά την εφαρμογή αυτών των βελτιώσεων. Η σαφής επικοινωνία σχετικά με τον τρόπο χειρισμού και προστασίας των δεδομένων των χρηστών, ειδικά κατά την ενσωμάτωση με τρίτους παρόχους ελέγχου ταυτότητας, συμβάλλει στην οικοδόμηση εμπιστοσύνης μεταξύ των χρηστών.

3.7 Δοκιμές και διασφάλιση ποιότητας

Αυτοματοποιημένες δοκιμές: Με αυτοματοποιημένες δοκιμές μονάδας, ολοκλήρωσης και widget για να διασφαλίσετε τη συνεχή ποιότητα και αξιοπιστία του κώδικα.

Ενσωμάτωση σχολίων χρηστών: εφαρμογή μηχανισμών για τη συλλογή σχολίων από τους χρήστες απευθείας μέσα στην εφαρμογή, επιτρέποντας τη συνεχή βελτίωση με βάση τις προτάσεις και τις εμπειρίες των χρηστών.

3.8 Βελτιώσεις Επεκτασιμότητας και Υποστήριξης

Επεκτασιμότητα Backend: Επέκταση της υποδομής backend για την προσαρμογή σε πιθανές αυξήσεις στη βάση χρηστών και τον όγκο δεδομένων.

Συνεργασία σε πραγματικό χρόνο: Λειτουργίες που επιτρέπουν τη συνεργασία σε πραγματικό χρόνο, επιτρέποντας σε πολλούς χρήστες να αλληλεπιδρούν και να συντονίζονται μέσα σε κοινόχρηστα ημερολόγια.

3.9 Ολοκλήρωση διαδικασίας ανάκτησης κωδικού

Η εφαρμογή ενός ασφαλούς και φιλικού προς το χρήστη μηχανισμού ανάκτησης κωδικού πρόσβασης μπορεί να αποτελέσει βασική βελτίωση. Μια αποτελεσματική προσέγγιση είναι η ενσωμάτωση ενός συστήματος ανάκτησης κωδικού πρόσβασης που χρησιμοποιεί κωδικούς πρόσβασης μιας χρήσης (OTP). Αυτή η λειτουργία προσθέτει ένα επιπλέον επίπεδο ασφάλειας, παρέχοντας παράλληλα στους χρήστες μια βελτιωμένη διαδικασία για την ανάκτηση πρόσβασης στους λογαριασμούς τους. Όπως αναφέρθηκε και σε προηγούμενο κεφάλαιο αυτή η λειτουργία έχει ξεκινήσει να δημιουργείται, αλλά δεν είναι ολοκληρωμένη.

Το σύστημα ανάκτησης κωδικού πρόσβασης που βασίζεται σε OTP περιλαμβάνει συνήθως την αποστολή ενός δυναμικά δημιουργημένου κώδικα στο καταχωρημένο email ή τον αριθμό τηλεφώνου του χρήστη. Στη συνέχεια, ο χρήστης εισάγει αυτόν τον κωδικό στην

εφαρμογή για να επαληθεύσει την ταυτότητά του και να προχωρήσει στη ρύθμιση ενός νέου κωδικού πρόσβασης. Αυτή η μέθοδος εξασφαλίζει ένα ασφαλές και αποτελεσματικό μέσο ανάκτησης λογαριασμού.

Η τακτική ενημέρωση και έλεγχος των πρωτοκόλλων ασφαλείας, όπως τα πρότυπα κρυπτογράφησης και οι αλγόριθμοι δημιουργίας OTP, είναι απαραίτητη για να παραμείνετε μπροστά από πιθανές απειλές για την ασφάλεια. Η διεξαγωγή διεξοδικών δοκιμών, συμπεριλαμβανομένων των δοκιμών διείσδυσης, βοηθά στον εντοπισμό και την αντιμετώπιση τυχόν τρωτών σημείων στο σύστημα ανάκτησης κωδικού πρόσβασης.

Λαμβάνοντας υπόψη τη σημασία της ασφάλειας των δεδομένων των χρηστών, είναι σημαντικό να κοινοποιούνται αυτές οι βελτιώσεις με διαφάνεια στους χρήστες. Η παροχή πληροφοριών σχετικά με τα ενισχυμένα μέτρα ασφαλείας και τα οφέλη της ανάκτησης κωδικού πρόσβασης βάσει OTP μπορεί να ενσταλάξει εμπιστοσύνη στους χρήστες σχετικά με τη δέσμευση της εφαρμογής για την προστασία των λογαριασμών τους.

Συμπεράσματα

Το αποκορύφωμα αυτού του έργου Capstone υπογραμμίζει την επιτυχή ανάπτυξη μιας εφαρμογής για κινητά που αντιμετωπίζει επιδέξια τις περίπλοκες προκλήσεις που σχετίζονται με τη διαχείριση δεδομένων iCal από διάφορες πλατφόρμες. Η εφαρμογή του Flutter για τις υπηρεσίες frontend και Firebase για backend εξασφαλίζει μια συνεκτική και φιλική προς το χρήστη εμπειρία, επιτρέποντας σε άτομα και φορείς διαχείρισης καταλυμάτων να οργανώνουν και να έχουν πρόσβαση στις πληροφορίες ημερολογίου τους.

Μέσω σχολαστικών δοκιμών σε κάθε στάδιο ανάπτυξης, η εφαρμογή επιδεικνύει ισχυρή λειτουργικότητα και διόρθωση σφαλμάτων. Η ενσωμάτωση του Flutter και του Firebase συμβάλλει σε μια ρευστή διεπαφή χρήστη, προσφέροντας στους χρήστες ένα διαισθητικό μέσο οπτικοποίησης των κρατήσεών τους. Η συστηματική προσέγγιση του Capstone, από τη σύλληψη έως την υλοποίηση, αναλύεται διεξοδικά στο έγγραφο, παρέχοντας μια διαφανή εικόνα των διαδικασιών.

Η ανάλυση των αποτελεσμάτων ρίχνει φως στην αποτελεσματικότητα της εφαρμογής στην ανάκτηση και την εμφάνιση iCal γεγονότων, τονίζοντας τη χρησιμότητα και τη συνάφεια της σε σενάρια πραγματικού κόσμου. Ενώ αναγνωρίζουμε τα επιτεύγματα, είναι επιτακτική ανάγκη να αναγνωρίσουμε τους εγγενείς περιορισμούς και να εξετάσουμε πιθανές οδούς βελτίωσης, διότι η εφαρμογή είναι ακόμα σε πολύ αρχικό στάδιο. Αυτό το έργο συμβάλλει ουσιαστικά στη σφαίρα της ανάπτυξης εφαρμογών για κινητά, προσφέροντας μια πολύτιμη λύση για την οργάνωση και τη διαχείριση δεδομένων ημερολογίου σε πολλές πλατφόρμες.

Τελος, η επιτυχής εκτέλεση του έργου αυτού επιβεβαιώνει τη σημασία του στην παροχή μιας αποτελεσματικής λύσης για τους χρήστες και τους φορείς διαχείρισης καταλυμάτων. Η ενσωμάτωση της τεχνολογίας και η προσεκτική διαδικασία υλοποίησης θέτουν τα θεμέλια για μια εφαρμογή για κινητά που ξεχωρίζει στον τομέα της οπτικοποίησης και διαχείρισης δεδομένων iCal.

Βιβλιογραφία

- table_calendar | Flutter Package. (2023, January 6). Dart Packages. https://pub.dev/packages/table_calendar
- Flutter - Build apps for any screen. (2017, May 12). <https://flutter.dev/>
- cupertino_icons | Dart Package. (2023, August 30). Dart Packages. https://pub.dev/packages/cupertino_icons
- http | Dart Package. (2023, June 26). Dart Packages. <https://pub.dev/packages/http>
- icalendar_parser | Dart Package. (2023, May 12). Dart Packages. https://pub.dev/packages/icalendar_parser
- sqflite | Flutter Package. (2023, July 26). Dart Packages. <https://pub.dev/packages/sqflite>
- firebase_core | Flutter Package. (2023, November 28). Dart Packages. https://pub.dev/packages/firebase_core
- firebase_auth | Flutter Package. (2023, November 28). Dart Packages. https://pub.dev/packages/firebase_auth
- cloud_firestore | Flutter Package. (2023, November 28). Dart Packages. https://pub.dev/packages/cloud_firestore
- firebase_storage | Flutter Package. (2023, December 12). Dart Packages. https://pub.dev/packages/firebase_storage
- provider | Flutter Package. (2023, November 10). Dart Packages. <https://pub.dev/packages/provider>
- flutter_spinkit | Flutter Package. (2023, May 1). Dart Packages. https://pub.dev/packages/flutter_spinkit
- flutter_otp_text_field | Flutter Package. (2022, September 2). Dart Packages. https://pub.dev/packages/flutter_otp_text_field
- get | Flutter Package. (2023, September 8). Dart Packages. <https://pub.dev/packages/get>
- image_picker | Flutter Package. (2023, August 30). Dart Packages. https://pub.dev/packages/image_picker
- path_provider | Flutter Package. (2023, August 30). Dart Packages. https://pub.dev/packages/path_provider
- path | Dart Package. (2022, December 5). Dart Packages. <https://pub.dev/packages/path>
- cached_network_image | Flutter Package. (2023, December 31). Dart Packages. https://pub.dev/packages/cached_network_image
- carousel_slider | Flutter Package. (2021, June 19). Dart Packages. https://pub.dev/packages/carousel_slider

Visual Studio Code - Code editing. Redefined. (2015, April 29).

<https://code.visualstudio.com/>

Android Studio & App Tools - Android Developers. (2013, May). Android Developers.

<https://developer.android.com/studio>

Bhattacharya, A. (2023, December 22). What is SHA? What is SHA used for? | Encryption Consulting. Encryption Consulting. <https://www.encryptionconsulting.com/education-center/what-is-sha/>

Greek ministry of tourism. (2024, January 9). Όραμα και Αποστολή | Ministry of Tourism. Ministry of Tourism. <https://mintour.gov.gr/ypoyrgeio/orama-apostoli/>

NOBEDS. (2023, July 17). Free hotel management system: #1 - hotel management free software. Hotel Management Free Software. <https://nobeds.com/>

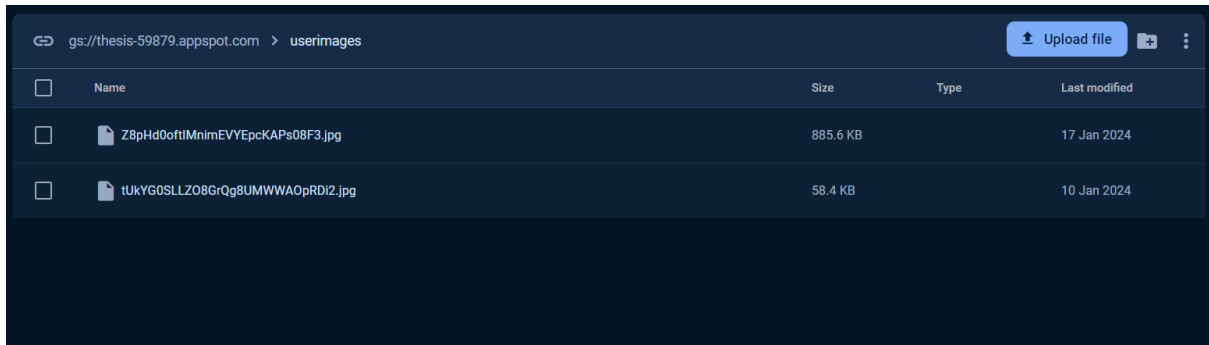
Euroart. (2024, January 19). Vacation Rental Channel Manager | Rentals United. Vacation Rental Channel Manager. <https://rentalsunited.com/>

Channel Manager, Property Management System & Booking engine. (2006). <https://www.beds24.com/>

Παραρτήματα

Παράρτημα Α. Τεχνικά χαρακτηριστικά

Α1. Δεδομένα Εικόνας Προφίλ



☐	Name	Size	Type	Last modified
☐	 Z8pHd0ofitlMnimEVYEpcKAPs08F3.jpg	885.6 KB		17 Jan 2024
☐	 tUkYG0SLlZ08GrQg8UMWWAOpRDI2.jpg	58.4 KB		10 Jan 2024

Εικόνα 17: Firebase Document Structure of the storage of images