

ΤΕΧΝΙΚΗ ΑΝΑΦΟΡΑ ΕΦΑΡΜΟΓΗΣ

ΑΘΑΝΑΣΙΟΣ ΚΙΒΕΡΤΖΙΚΙΔΗΣ

Τίτλος Εφαρμογής: **ForgetIt**



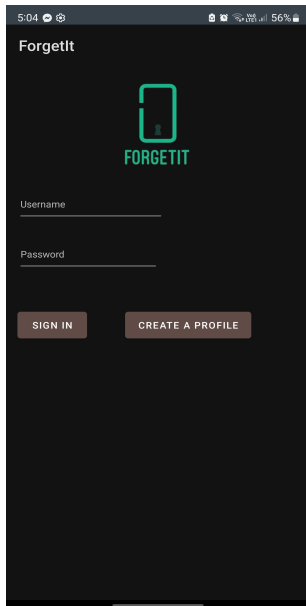
ΣΕΠΤΕΜΒΡΙΟΣ 2023

1 Το ForgetIt

Η εφαρμογή που αναπτύξαμε προσφέρει ένα περιβάλλον εύκολης και ασφαλούς αποθήκευσης κωδικών στο κινητό. Ο χρήστης μπορεί να δημιουργήσει ένα καινούργιο profile, να συνδέεται σε αυτό και να αποθηκεύει, να επεξεργάζεται, να διαγράφει ή να αντιγράφει τους κωδικούς του στο clipboard. Δίνεται η επιλογή στον χρήστη να ορίσει έναν σύνθετο ή απλό κωδικό στο profile, ώστε να καθοριστεί το επίπεδο ασφάλειας των κωδικών που θα αποθηκεύσει.

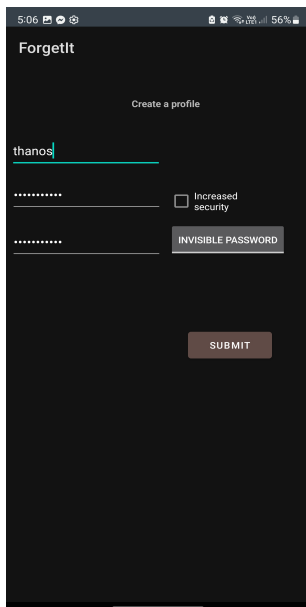
Ακολουθούν screenshots από τη λειτουργία της εφαρμογής:

1.1 Αρχική δραστηριότητα της εφαρμογής



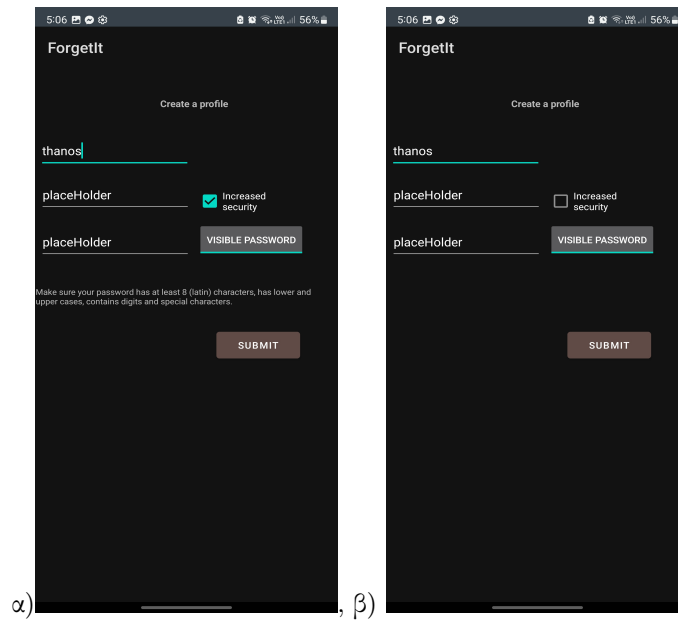
Εδώ ο χρήστης μπορεί να επιλέξει να συνδεθεί σε έναν υπάρχοντα ή να δημιουργήσει έναν νέο λογαριασμό.

1.2 Δημιουργία ενός λογαριασμού στην εφαρμογή



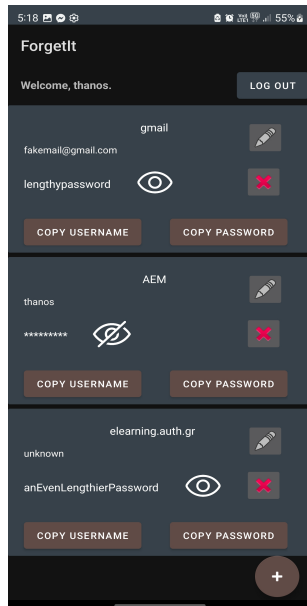
Στο πρώτο πεδίο ο χρήστης συμπληρώνει το αναγνωριστικό του λογαριασμού.

Απαγορεύονται οι ειδικοί χαρακτήρες και τα κενά. Στα επόμενα δυο πεδία ο χρήστης πληκτρολογεί το συνθηματικό του νέου λογαριασμού. Για να ολοκληρωθεί η εγγραφή του λογαριασμού οι κωδικοί στα τελευταία πεδία πρέπει να ταυτίζονται και ο χρήστης πρέπει να πατήσει το κουμπί submit.



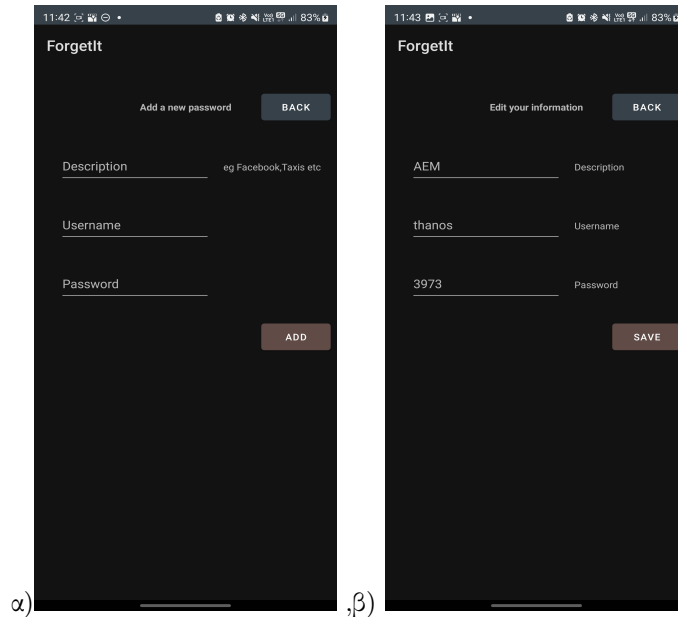
Στην εικόνα 'α' φαίνεται η δημιουργία του λογαριασμού με έναν σύνθετο κωδικό, ενώ στη 'β' η δημιουργία του χωρίς να υπάρχει ανάγκη για περαιτέρω προστασία.

1.3 Βασική δραστηριότητα της εφαρμογής



Εδώ φαίνεται η βασική οθόνη διαχείρισης των κωδίκων ενός λογαριασμού. Φαίνονται τα κουμπιά επεξεργασίας, διαγραφής και αντιγραφής κάθε εγγραφής. Μάλιστα, υπάρχει κουμπί που αφορά την εμφάνιση του κωδικού, ενώ κάτω στη δεξιά γωνία της οθόνης το κουμπί προσθήκης νέας εγγραφής.

1.4 Προσθήκη-Επεξεργασία δεδομένων



Η εικόνα 'α' αφορά την προσθήκη νέου κωδικού για αποθήκευση. Απαιτείται η συμπλήρωση τριών πεδίων, πρώτο το Description που περιγράφει την προσθήκη και τη διαφοροποιεί από τις υπόλοιπες, δεύτερο το username που συνοδεύει τον κωδικό και τέλος φυσικά το το πεδίο password ζητείται ο κωδικός.

Η δεύτερη εικόνα αφορά την επεξεργασία των πεδίων σε μια εγγραφή που υπάρχει ήδη στο σύστημα.

2 Ανάλυση της προγραμματιστικής λογικής

2.1 Βάση Δεδομένων

Η βάση δεδομένων υλοποιείται στην κλάση DataBase. Στη βάση υπάρχει τουλάχιστον ένας πίνακας, ο οποίος κρατάει τις εγγραφές των χρηστών της εφαρμογής και των συνθηματικών τους μαζί με μια σύντομη περιγραφή του χρήστη για την εγγραφή. Ο αριθμός των επιπλέον πινάκων βασίζεται στον αριθμό των τελικών χρηστών, καθώς τα δεδομένα κάθε χρήστη αποθηκεύονται σε ξεχωριστό πίνακα, πάντα με όνομα ίδιο με το όνομα του χρήστη. Κατά την ολοκλήρωση της εγγραφής ενός χρήστη καλείται η μέθοδος newUserTable() η οποία δημιουργεί ένα νέο πίνακα για τον νέο χρήστη.

Οι υπόλοιπες λειτουργίες της κλάσης αφορούν την προσθήκη, τη διαγραφή και ανάκτηση των δεδομένων και αναφέρονται παρακάτω σύμφωνα με την χρήση τους στις δραστηριότητες.

Η εφαρμογή απαρτίζεται από τέσσερις δραστηριότητες activities, όπως παρουσιάστηκε παραπάνω με τα στιγμιότυπα οθόνης.

2.2 LoginActivity

Η αρχική οθόνη υλοποιείται με την κλάση LoginActivity μαζί με το αρχείο activity_login. Μόλις ο χρήστης πατήσει το κουμπί SIGN IN αφού έχουν συμπληρωθεί τα EditText που αφορούν το Username και το Password του λογαριασμού, καλείται η μέθοδος signIn().

Η μέθοδος signIn() ταυτοποιεί τα δεδομένα που δόθηκαν με τις εγγραφές της βάσης δεδομένων, μέσω της μεθόδου findProfile() σύμφωνα με την επιστροφή της και εμφανίζει ανάλογο μήνυμα στην οθόνη. Αν τα στοιχεία που δόθηκαν από τον χρήστη ταυτίζονται με μια εγγραφή της βάσης, έχουμε την έναρξη της βασικής δραστηριότητας της εφαρμογής, περίπτωση που ο χρήστης πατήσει SIGN IN, ενώ στην περίπτωση που η φόρμα είναι συμπληρωμένη με μη έγκυρες τιμές (πχ κενό) η μέθοδος getInputs() κατά την συλλογή των τιμών εισόδου εμφανίζει το αντίστοιχο μήνυμα στην οθόνη.

Όταν πατηθεί το κουμπί SIGN UP καλείται η μέθοδος signUp η οποία είναι υπεύθυνη για της έναρξη της δραστηριότητας για την εγγραφή ενός νέου χρήστη στην εφαρμογή.

2.3 SignUpActivity

Η εγγραφή ενός χρήστη υλοποιείται με την κλάση SignUpActivity και το layout activity_sign_up. Ο χρήστης πληκτρολογεί το όνομα και τον κωδικό με τα οποία επιθυμεί να συνδεθεί στο σύστημα και πατώντας το κουμπί SUBMIT ολοκληρώνει την διαδικασία.

Επιλέγοντας το checkBox καλείται η μέθοδος suggestionShow() . Αυτή θέτει ένα textView ορατό με στόχο την ενημέρωση του χρήστη για την ενεργοποίηση του επιπρόσθετου επιπέδου ασφαλείας του λογαριασμού και την ανάγκη ο κωδικός να πληροί ορισμένες προδιαγραφές (να περιλαμβάνει ειδικούς χαρακτήρες , αριθμούς, κεφαλαία κτλπ.). Ο έλεγχος της ασφάλειας του κωδικού γίνεται μέσω της συνάρτησης validPassword().

Παράλληλα, υπάρχει ένα toggleButton που πατώντας το καλεί τη συνάρτηση makePasswordVisible(), η οποία μπορεί να κάνει αναγνώσιμα ή να αποκρύψει τα πεδία των κωδικών.

Μόλις πατηθεί το κουμπί SUBMIT καλείται η μέθοδος submit(), η οποία σε πρώτη φάση ελέγχει την εγκυρότητα των τιμών εισόδου. Απαιτείται προσοχή στις τιμές που δίνονται στο πεδίο Username καθώς τα δεδομένα κάθε χρήστη αποθηκεύονται σε έναν πίνακα με όνομα το username του χρήστη. Ο έλεγχος αυτός γίνεται με την συνάρτηση validUsername(). Αν όλες οι εισοδοί είναι σωστές, τότε καλείται η μέθοδος SQLSign() η οποία ελέγχει την μοναδικότητα του username στο σύστημα και παράλληλα δημιουργεί τον πίνακα του νέου χρήστη μέσω της μεθόδου addProfile() της βάσης, ενώ παράλληλα ενεργοποιεί τη βασική δραστηριότητα.

2.4 AccountActivity

Η βασική δραστηριότητα της εφαρμογής υλοποιείται με την κλάση `AccountActivity` και το αρχείο διάταξης `activity_account`.

Το κουμπί LOG OUT καλεί την συνάρτηση `logout()` που επιστρέφει τον χρήστη στην κεντρική δραστηριότητα, θέτοντας την μάλιστα ως μοναδική στο `activity stack`. Με αυτόν τον τρόπο οι χρήστες που έφτασαν στην δραστηριότητα μέσω της `signup` δραστηριότητας δεν επιστρέφουν σε αυτή.

Το `RecyclerView` συνδέεται με έναν αντάπτορα που υλοποιείται στην κλάση `RecyclerViewAdapter`.

2.4.1 RecyclerViewAdapter

Ο `RecyclerViewAdapter` έχει ως `ViewHolder` `cardViews` και δέχεται τα δεδομένα του χρήστη με την μέθοδο `getUserContext()` της βάσης. Προφανώς κάθε εγγραφή του πίνακα αντιστοιχεί σε μια κάρτα του `RecyclerView`. Σύμφωνα με τα δεδομένα αυτά γεμίζει τα περιεχόμενα των καρτών (`description-username-password`) στην μέθοδο `onBindViewHolder()`, ενώ περιέχει μια εσωτερική κλάση `ViewHolder` που συνδέει προγραμματιστικά στο κάθε στιγμιότυπό της τα στοιχεία του εγγράφου διάταξης `card.layout` και αναθέτει `action listeners` στα κουμπιά της κάθε κάρτας.

Τα κουμπιά υποστηρίζουν τις λειτουργίες της διαγραφής, της επεξεργασίας των δεδομένων του χρήστη, δημιουργώντας τη δραστηριότητα επεξεργασίας και αξιοποιώντας τη μέθοδο `deleteUserContext()` της βάσης αντίστοιχα. Τέλος, ένα `imageView` εμφανίζει ή αποκρύπτει τον κωδικό με κάθε κλικ σε αυτό.

Το κουμπί `floating button` καλεί την συνάρτηση `addNewData()` οδηγώντας τον χρήστη στην δραστηριότητα προσθήκης δεδομένων.

2.5 AddActivity

Η ίδια δραστηριότητα χρησιμοποιείται για την προσθήκη και επεξεργασία των δεδομένων.

Η διάκρισή γίνεται στη μέθοδο `onCreate()` από τη λήψη των `extra` του `intent` που χρησιμοποιήθηκε κατά την δημιουργία της δραστηριότητας με ένα `flag`, αν είναι `true` τότε πρόκειται για προσθήκη δεδομένων. Αυτό σημαίνει πως το αρχείο διάταξης `activity_add` διαμορφώνεται κατάλληλα.

Αν ο σκοπός είναι η επεξεργασία, τότε ως `hints` των `editText` τοποθετούνται τα υπάρχοντα δεδομένα του χρήστη και εμφανίζονται `editText` υπενθυμίζοντας στον χρήστη τι αφορά κάθε πεδίο.

Αφού ο χρήστης έχει πληκτρολογήσει τα δεδομένα καλείται η συνάρτηση `submit()` και γίνονται οι απαραίτητοι έλεγχοι εγχυρότητας και στην συνέχεια ενημερώνεται η βάση δεδομένων μέσω της μεθόδου `addUserContext()`

3 Παραδοχές

It's not a bug its a feature: όταν ο χρήστης βρίσκεται στη βασική δραστηριότητα μπορεί να πατήσει πάνω σε ένα `imageView` μιας κάρτας, για να φανερώσει τον

κωδικό που βρίσκεται μέσα σε ένα TextView ή να τον αποκρύψει αλλάζοντας ταυτόχρονα την εικόνα του. Η αρχική κατάστασή του imageView είναι να αποκρύπτει τον κωδικό. Σε περίπτωση που ο χρήστης, ενώ έχει αλλάξει την κατάσταση του κωδικού σε ορατό, αλλάζει τον προσανατολισμό της οθόνης του, ο κωδικός αυτόματα θα επιστρέφει στην αρχική του κατάσταση. Αυτό συμβαίνει καθώς κατά την σχεδίαση του τρόπου παρουσίασης των δεδομένων στον χρήστη, θεωρήσαμε πως το TextView θα έπρεπε να παραμένει κρυμμένο όταν αλλάζει η κατάσταση της συσκευής λόγω του ευαίσθητου περιεχομένου που περιέχει .

Επιπλέον αξίζει να αναφερθεί πως στα πεδία username των δραστηριοτήτων που αφορούν την δημιουργία και την σύνδεση σε έναν υπάρχοντα λογαριασμό, αγνοείται το τελευταίο κενό της συμβολοσειράς, δηλαδή αν δοθεί ως είσοδος η συμβολοσειρά 'ΦΑΜΠΙΟΝ ' το σύστημα θα την μετατρέψει σε 'ΦΑΜΠΙΟΝ'. Αυτό συμβαίνει καθώς τα πεδία δεν υποστηρίζουν τα κενά και από τις διάφορες δοκιμές που κάναμε φάνηκε πως το κενό στο τέλος των συμβολοσειρών στα αναφερόμενα πεδία ήταν ένα συχνό σφάλμα των χρηστών. Υπεύθυνη μέθοδος στις αντίστοιχες κλάσεις είναι η μέθοδος saveTheLastSpace().