**UNIVERSITI TUNKU ABDUL RAHMAN**

**Faculty of Information and Communication Technology**



**UCCD3223 Mobile Applications Development**

**(January 2022 Trimester)**

**Individual Practical Assignment**

|  |  |
| --- | --- |
| **Name** | Tan Xi En |
| **Student ID** | 19ACB04098 |
| **Course** | Computer Science |
| **Practical Group** | P12 |
| **Lecturer** | Tan Chiang Kang |

|  |  |  |
| --- | --- | --- |
| **Marking scheme** | **Marks** | **Remarks** |
| Correctness of output | **× 3** |  |
| Design of UI | **× 2.5** |  |
| User Friendliness | **× 2.5** |  |
| Neat Program Documentation |  |  |
| Report Format |  |  |
| **TOTAL** |  |  |

**Introduction**

With the advancement of modern technology, digital transformation is altering every aspect of how today’s business operate and complete. Internet services are being introduced at every second, and these internet services requires a certain level of security to prevent unauthorized access to these services. Password is used as a mean to protect user security and prevent unauthorized access. However, user-created password may either be too simple or too repetitive, as users may use the same password for each internet service. To combat this problem, password managers are introduced to manage password for different application. Therefore, for this assignment, I plan to propose a password manager that stores, generate and manage multiple passwords for multiple applications.

**Problem Statement**

1. **Password do not have strong security**

The main issues with a lot of passwords are that they are too simple. User generated password usually comprises of only number and alphabets. These passwords are too simple to be cracked as they have a low number of combinations. For a security tool used to prevent unauthorized access to important services, passwords need to meet a certain degree of security. Therefore, our password manager will generate strong combinations of password with certain length.

1. **Some Passwords are repetitive**

The main issue with user defined passwords are that user tend to use only one password for each internet service. This is because it is much easier memorising one passwords, rather multiple unique passwords for each internet service. A user should ideally have unique passwords for all of one’s services to reduce the impact of compromised passwords. The implied effect is that if one of the services is compromised, none of the user’s other services are.

**App Functionality**

While there are many instances of password managers in Google play Store (e.g. KeePass, DashLane etc.), there are universally agreed functionalities for a password manager. A basic password manager should be able to:

* Generate Secure password with multiple character sets and variable length
* Able to add, update, delete password for multiple websites
* Encrypt password when stored into database, decrypt password when retrieved from database.
* Search and sort list of passwords.
* Export Database as Backup

To store and retrieve password, we need to create a local database within our android phone. Normally, for application development, developers would use SQLite database, as it is provided in all android phone. However, for ease of development, I’ve decided to use Room Database to implement our database functionality. This is because the code are much cleaner and it is much easier to debug using Room Database.

Secondly, to show our list of password, I have also decided to implement RecycleView adapter. This is because RecycleView have its own view model and life cycle that constantly recycle old view and reuse for new elements, hence the name recycle view. Recycle view allows our android app to run smoother, by constantly recycle old elements and remove unnecessary view.

To implement these functionalities, I have designed multiple activities and fragment that accommodates these functions. The UIs that I have designed are, activity\_main.xml, fragment\_gen\_pwd.xml, fragment\_change\_mas\_pwd.xml, activity\_login.xml, activity\_add\_pwd.xml, activity\_edit\_pwd.xml.

Login Page

|  |  |
| --- | --- |
|  |  |
| Default Login Page for Password Manager  Have eye icon button to show or hide master password. | Shows toast message if master password is incorrect. |

Main Page

|  |  |
| --- | --- |
|  |  |
| Default Homepage for Password Manager. Shows list of password that users can edit.  Shows Floating Action Button that users can press to add new password  Use Recyle View Adapter to show List of password. | Custom Navigation Drawer on the right. When user press on the navigation icon, the drawer will pop up and allow user to navigate to different sections of the application |

Generate Password Page and Change Master Password Page

|  |  |
| --- | --- |
| Generate Password Page | Change Master Password Page |
|  |  |
| Helps user to auto generate password based on selected character Sets.  Offers 4 different types of Character set, that is Uppercase Characters, Lowercase Characters, Digit Characters, Symbol Characters  Allow User to generate password on variable length  Have copy to clipboard button that user may press to copy to clipboard  Have reset button that user may press to generate a new password | Allow User to change Master Password  Shows incorrect toast message if password and confirm password are not the same  Shows success toast message if password and confirm password are the same.  Have eye icon to hide or show password |

Add Password Page

|  |  |
| --- | --- |
|  |  |
| The following activity comprises of the basic property of each password. Title, Username, Password and Website.  If title, username, passwords and website are not filled in, an incorrect toast message will be shown to act as validation.  Optional field such as Emergency PIN, security questions are provided. User may choose to fill in.  Have eye icon to hide or show password field  Have save Icon to save password | |

Edit Password Page

|  |  |  |
| --- | --- | --- |
|  |  |  |
| The following edit page provides the list of field based on the password element selected.  There are three variations of Edit Page. This is based on the number of optional fields that the user have entered. For example, “Facebook” does not contain any optional field, whereas “Snapchat” and “Amazon” does.  Each of the field have a copy clipboard button, that copies the respective field into the clipboard  Each password field have a show / hide eye icon, that shows or hide password.  Each edit page have a delete and save button function, that performs its respective function. | | |

**Encrypted Password**

One of the main priorities of a password function is its ability to encrypt password. Encrypted password is important because in the scenario the original database is hacked, hackers would be unable to get the password, as they are not stored in plaintext. To achieve this, we have decided to use AES encryption with a randomly generated key that is stored within the user shared preference file. To demonstrate the functionality, we have provided two screenshot.

Graphical user interface, application

Description automatically generated

Using app inspection (tool that is provided in android studio), we are able to debug the information in our room database. As we can observe, there is a random cipher that is generate from our AES encryption. Therefore, we are successful in storing encrypted password in our Room Database model.

Text

Description automatically generated

**AES Key:**

The randomly generated key is stored in our shared preference file. The key will be retrieved and used when the password manager is required to decrypt the password.

Text

Description automatically generated

**Exporting Database**

By selecting our option menu, users are provided with the following option to export database. This option will create “PwdDb.txt”, that stores all the password in our database. This text file can be used as a backup, in the event the application is uninstalled for unspecified reason. User can then refer to the backup text file for their list of passwords stored.

|  |  |
| --- | --- |
|  |  |

The contents of “PwdDb.txt” is shown below:

Text

Description automatically generated with medium confidence

**Conclusion**

In conclusion, our password manager application comprises of all the basic functionalities needed. This password manager app is able to accommodate for variety of websites and generate strong password up to the length of 50. It provides 4 characters sets, which is sufficient security to not be cracked easily. Furthermore, each password are encrypted using AES database and a randomly generated key unique for each android phone. Therefore, our password manager app meets the basic requirements for a functioning password manager.

**Appendix:**