

OverKlock Password Manager

By-

Vishnu Priy Agnihotri(BT18CSE018)

Ujjwal Khanduri(BT18CSE007)

Himanshu Pandey(BT18CSE020)

Utkarsh Bhandari(BT18CSE006)

Software Requirements Specification Document

*Under the supervision of:
Dr. Maheep Singh*

Version: (1.0)

Date: (23/03/2021)

Table of Contents

1. Introduction	3
1.1 Purpose	3
1.2 Scope	3
1.3 Definitions, Acronyms, and Abbreviations	3
1.4 References	4
2. The Overall Description	4
2.1 Product Perspective	4
2.1.1 System Interfaces	4
2.1.2 Interfaces	5
2.1.3 Hardware Interfaces	5
2.1.4 Software Interfaces	5
2.1.5 Communications Interfaces	5
2.1.6 Memory Constraints	5
2.2 Assumptions & Dependencies and Apportioning of Requirement	5
3. Specific Requirements	6
3.1 External interface	6
3.2 Logical Database Requirements	6
4. Functional Requirements	6
5. Non-Functional Requirements	6
6. Change Management Process	8

1. Introduction

Nowadays, with the increasing number of registrations in various applications and websites there are increasing login credentials that we may forget time to time. Since we are prone to forget things over time we are always stuck on pages where our login credentials are required. In a fast pace world we have no time to waste of such problems. Therefore, our project is made while keeping such idea in mind so that our users have no difficulty in accessing their user names and password for any application they may use.

1.1 Purpose

.While, asking your browser to save your login credentials may be an option but when it comes to the security, this is not a good approach. There can be many third party services that may try to steal this information as you are unsecure over your internet connection. The purpose of our app is to find an effective solution for this particular problem which allows you to access all your passwords by remembering just one **Master Password**. This project provides an effective solution for everyone may it be college students or office workers.

1.2 Scope

- This project is a simple password manager for day to day use on an individual basis.
- This will encrypt all your login credentials using a hashing algorithm and store them in database and decrypt them as per your requirements.
- This version of the software will only contain simple user friendly interface which will perform the basic operations.
- Development of the user interface shall be a part of future development.

1.3 Definitions, Acronyms, and Abbreviations.

The components of the development phase makes use of the following:

- **SQL(Structured Query Language):** It is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). It is particularly useful in handling structured data, i.e. data incorporating relations among entities and variables.
- **Flutter:** Flutter is an open-source UI software development kit created by Google. It is used to develop applications for Android, iOS, Linux, Mac, Windows, Google and the web from a single codebase.

1.4 References

All the references used in the the project are as follows:

- **Flutter Documentation:** <https://flutter.dev/docs>
- **Wikipedia:** <https://en.wikipedia.org/>
- **W3Schools:** <https://www.w3schools.com/>
- **Coding with Flutter:** <https://www.youtube.com/watch?v=pTJJsmejUOQ>

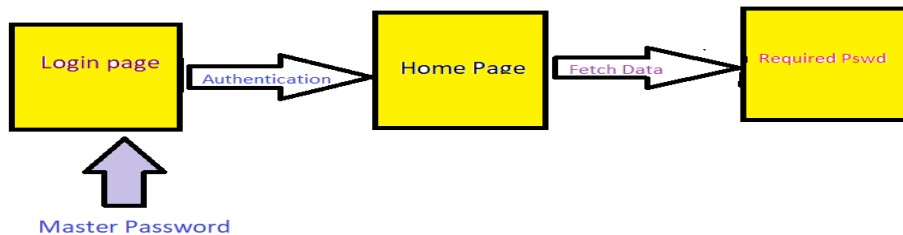
2. The Overall Description

2.1 Product Perspective

This application is a simple password manager. It is having a master password to unlock the application and the rest part will be user passwords. Initially a local storage will be created to test the functionality which then will be linked to an SQL database.

2.1.1 System Interfaces

The product will have a UI that will be asking you to enter your password (master password). When the password is verified by backend it will authenticate the session and home page will open where you can see your saved login passwords.



2.1.2 Interfaces

The interface is a simple version where the person will be entering data on the app on his phone. He will be viewing the Login page with an option to enter master password. If the entered password is correct next page containing list of the password will open. Select the appropriate and then it will be fetched and shown to you.

2.1.3 Hardware Interfaces

It will be supported by both the Android and iOS platform as we will designing it on flutter. Android phones having version higher than android 5 will be able to use it and it will be compatible with most iOS devices.

2.1.4 Software Interfaces

The other applications which whose service will be utilized will be a SQLite database and some appropriate hashing and encryption algorithm.

2.1.5 Communications Interfaces

It will be following localhost protocols as it will be having a local database and if a user opts to avail a backup at some remote server ,then both the databases can be set for a shared job.

2.1.6 Memory Constraints

For android ,minimum requirement should be 1GB RAM devices but will be very effective if the RAM is higher. For iOS, 1GB RAM is sufficient.

2.2 Assumptions & Dependencies and Apportioning of Requirements

Since, flutter is being used it can be easily deployed on both the platforms without any discrepancies. We are hopeful that the user doesn't give his master password willingly to anyone in that case he has put a nail in his coffin himself. The master password is getting stored as hashes .Thus, the security is undoubted.

As the further updates will roll out, new modifications will be done which will definitely increase the overall space requirements and some functioning but the users will be informed.

3. Specific Requirements

3.1 External Interfaces

Front end Software : Dart
Back end Software : SQLite
Operating System: Android

3.2 Logical Database Requirements

This Software uses SQLite Database.

4. Functional Requirements:

- a) **Logging In:** You will be filling your master password in the tab of login page which will then authenticate your session.
- b) **Select Menu:** The home page will have the menu from which you can view your saved password for a particular app.
- c) **Add button:** There will be an add button to add another entry for new app to save its password.

5. NONFUNCTIONAL REQUIREMENTS

5.1 PERFORMANCE REQUIREMENTS

The steps involved to perform the implementation of airline database are as listed below.

A) NORMALIZATION:

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

If a database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.

Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme. There are three different kinds of modifications of anomalies and formulated the first, second and third normal forms (3NF) is considered sufficient for most practical purposes. It should be considered only after a thorough analysis and complete understanding of its implications.

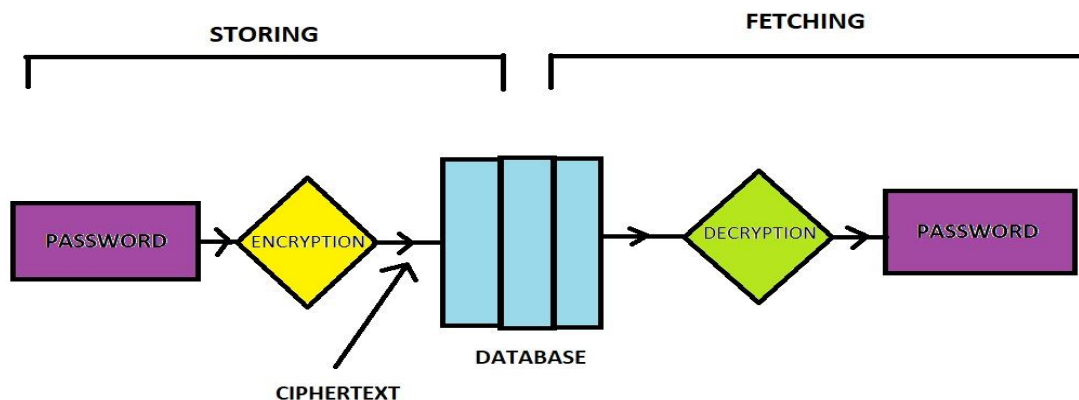
Since this project will be used by lesser number of individuals the problem of data redundancy is minute. This can be included in the future where the number of individuals is larger.

B) SAFETY REQUIREMENTS

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores the data by keeping the data in a remote database. So, if your local database is lost, you can log in from other database and retrieve your data. But this is a premium feature as maintaining a remote database will be costly.

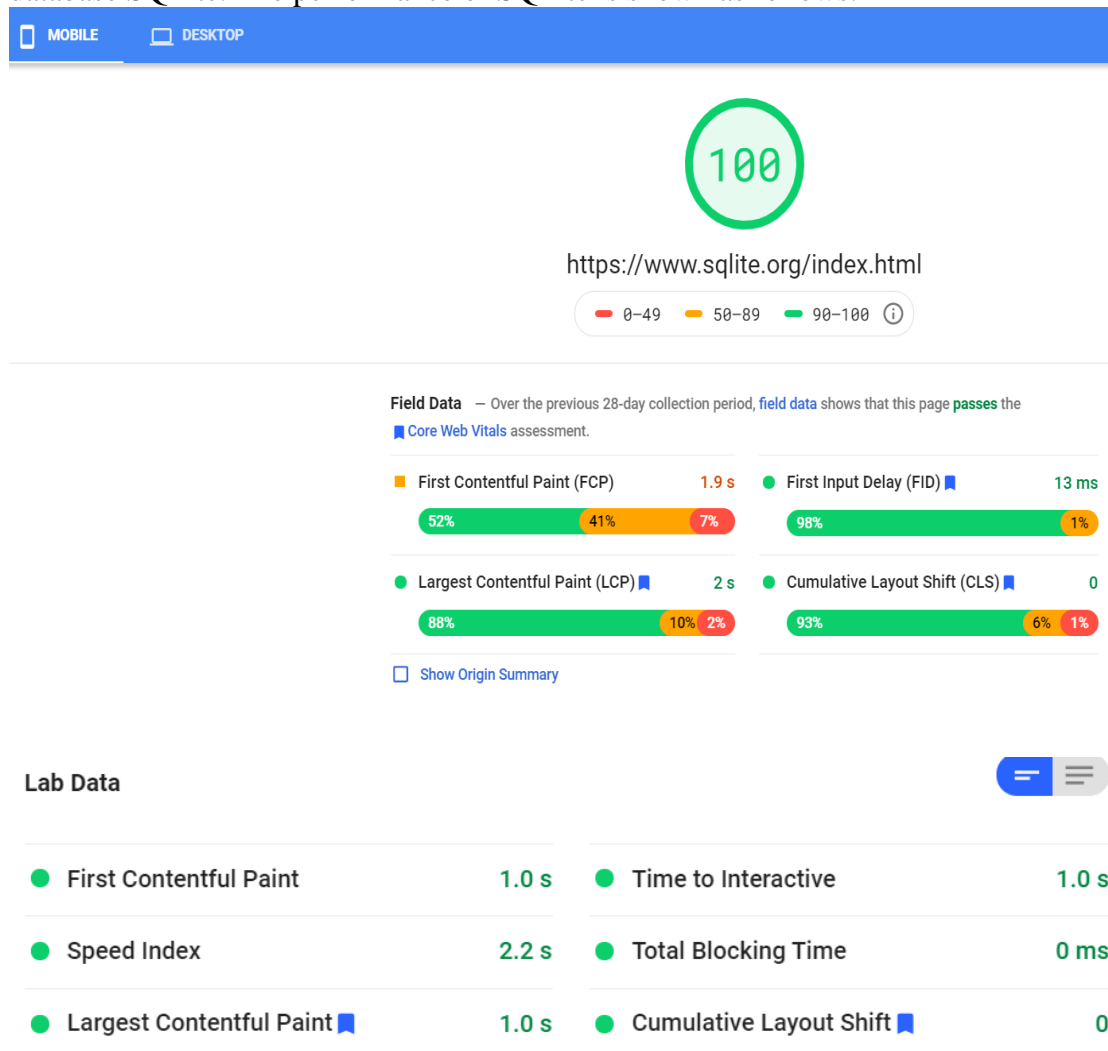
C) SECURITY REQUIREMENTS

The master password will be checked using hashing function. As for the password that will be saved in the application, we will use a suitable encryption algorithm to Encrypt the password and send them to the database. When the user wants to know the password, we will fetch the ciphertext from the database and decrypt it. Therefore, the security is increased.



D) SOFTWARE QUALITY ATTRIBUTES

AVAILABILITY: The availability of the app will be based on the availability of the database SQLite. The performance of SQLite is shown as follows:



CORRECTNESS: As long as the master password is correct all the information will be provided.

MAINTAINABILITY: It will require updates in future, but no such maintainance required.

6. Change Management Process

We are thinking of an app which will be like a stem of the tree i.e. we are first focusing on creating a basic structure for the app and in case any user wants some modification in it and the developers feel it a valid reason to be changed, then it will be deployed in the

next update. And if any user wants a customized feature and really demands it, he can purchase a premium for that.