2018

SecurePass

APPLICATION REQUIREMENTS

JAVINATOR9889

JAVINATOR9889 – GITHUB.COM | https://github.com/Javinator9889/SecurePass

1. Introduction

This guide is an SRS, which details the requirements of the *SecurePass* application. Here will be described modules, specifications, use cases and more about *SecurePass*.

1.1. Purpose

This is an SRS, which wants to show the developer (and the user) how is working *SecurePass* application, its capabilities and what is it.

It aims developers and people who is studying *Java* and how to develop *Android* applications.

1.2. Scope

This software product wants to keep your passwords safe and secure, by doing some cypher techniques.

Nowadays, we use passwords for everything and we face the situation that we should keep in memory lots of information and our passwords. With *SecurePass*, we want to make it simpler and easier, by registering unlimited accounts or applications and their password.

SecurePass will allow users to use an *Android* application, web interface for adding new entries, and synchronize them between their devices. In addition, it provides compatibility between *Android* versions and *Android Wear* devices.

SecurePass is going to make it simpler in order to keep your data safe and portable, having it wherever you go. Moreover, include the possibility to restore your data if you have lost it.

1.3. Definitions, acronyms and abbreviations

1.3.1. Definitions

- **Android**: Android is "*a mobile operating system developed by Google, based on a modified version of the Linux kernel and other open source software and designed primarily for touchscreen mobile devices such as smartphones and tablets"* [1].
- **Android Wear**: "*Wear OS, previously known as Android Wear, is a version of Google's Android operating system designed for smartwatches and other wearables.*" [2]
- **Google Drive**: Google Drive is "*a file storage and synchronization service developed by Google.*" [3]
- **Java**: Java is "*a programming language and computing platform first released by Sun Microsystems in 1995. There are lots of applications and websites that will not work unless you have Java installed, and more are created every day. Java is fast, secure, and reliable. From laptops to datacenters, game consoles to scientific supercomputers, cell phones to the Internet, Java is everywhere!*" [4].
- **Google Play Services**: GPS is "*a package of APIs (things that assist programmers and allow apps to easily communicate with other apps) that ensure fewer apps are dependent upon Android OS updates to run.*" [5].
- **Terms of Service**: Terms of Service are "*rules by which one must agree to abide in order to use a service*" [6].
- **Bug**: a software bug is "*an error, flaw, failure or fault in a computer program or system that causes it to produce an incorrect or unexpected result or to behave in unintended ways*" [7].
- **Material Design**: Material Design is "*a unified system that combines theory, resources, and tools for crafting digital experiences*" [8].
- **Responsive**: Responsive Web design is "*what makes your web page look good on all devices (desktops, tablets, and phones).*
  Responsive Web Design is about using HTML and CSS to resize, hide, shrink, enlarge, or move the content to make it look good on any screen" [9].

1.3.2. Abbreviations

- **SRS**: Software Requirement Specification.
- **SP**: SecurePass (this application).
- **GPS**: Google Play Services.
- **API**: Application Programming Interface.
- **TOS**: Terms Of Service.

1.4. References

[1] Wikipedia, "Android (operating system) - Wikipedia," Wikipedia, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Android_(operating_system).

[2] Wikipedia, "Wear OS - Wikipedia," Wikipedia, 2014. [Online]. Available: https://en.wikipedia.org/wiki/Wear_OS.

[3] Wikipedia, "Google Drive - Wikipedia," Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Google_Drive.

[4] Oracle, "What is Java and do I need it?," Oracle, [Online]. Available: https://www.java.com/en/download/faq/whatis_java.xml.

[5] C. Marshall, "Google Play Services: what is it and what is it for? - AndroidPIT," AndroidPIT, 5 May 2016. [Online]. Available: https://www.androidpit.com/google-play-services-what-is-it-and-what-is-it-for.

[6] Wikipedia, "Terms of Service - Wikipedia," Wikipedia, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Terms_of_service.

[7] Wikipedia, "Software bug - Wikipedia," Wikipedia, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Software_bug.

[8] Material Design, "Material Design," material, [Online]. Available: https://material.io/.

[9] w3schools, "HTML Responsive Web Design," w3schools, [Online]. Available: https://www.w3schools.com/html/html_responsive.asp.

[10] Google, "Dashboards | Android Developers," Google, 2018. [Online]. Available: https://developer.android.com/about/dashboards/index.html.

[11] R. Ehrhardt, "Password Safe and Manager - Google Play Applications," Google Play Store, [Online]. Available: https://play.google.com/store/apps/details?id=com.reneph.passwordsafe.

[12] E. Molla, "My Passwords - Password Manager - Google Play Applications," Google Play Store, [Online]. Available: https://play.google.com/store/apps/details?id=com.er.mo.apps.mypasswords.

[13] Google, "Google Chrome," [Online]. Available: https://www.google.com/chrome/index.html.

[14] Mozilla, "Mozilla Firefox," [Online]. Available: https://www.mozilla.org/en-US/firefox/new/.

1.5. Global vision

The following pages will show you specific requirements and features for SP, which will be also described. You will be able to see a global description for the product and functional/non-functional requirements for this application.

2. General description

2.1. Product perspective

SP is an application that can be run on any Android device with KitKat (4.4) version or higher. Based on some statistics, it should run correctly on almost on 99’3% of devices [10].

There are some similar products in Play Store, such as *Password Safe and Manager* [11] or *My Passwords – Password Manager* [12]. The difference from this application is that:

1. SP is open source.
2. SP is free, no PRO version, all capabilities enabled.
3. SP aims to be a better application, useful for the user.

SP is *user-friendly*, this means the user will find easier to use and transparent, ergo the user knows what is the application doing and why.

The data is stored in an encrypted database, with a randomly generated password for better encryption. Also is secured by a master password, so the user is not able to access the application if he forgotten it.

The different connections are protected, so anyone can steal data during transfers. Finally, there is a functionality where the user can connect to Google Drive and store in an encrypted folder the data, in order to be able to reach it if he lost the access to the phone or forgotten the master password.

2.2. Product functions

This software can do different functions that can be divided in groups:

I. Accounts & Password storage

Once the application is installed, the user is able to create his firsts passwords and save them in a secure location, and repeat this process “infinite” times.

Once the user create a password, he can access and see it, and is able to change and modify data. For doing this, first he must indicate that he wants to change some data.

Finally, in every moment he is able to delete created passwords, by clicking them and selecting the option “Delete”.

II. QR generation

For a created password, the user can generate a QR code for sharing it. For this option, the QR must contain the necessary data so the other user can read it and have all the needed information.

III. Web access

The user is able to generate a webpage so he can access to it through a web browser at a specific location. This connection must be encrypted so no one can steal data between transfers and the master password is required for accessing to the content.

IV. Different categories

When the user creates a password, he can choose a category to save it, so then the application can sort by categories and display them organized.

V. Password generation

As an additional feature, the application is able to generate secured passwords so the user can copy and use them. These passwords are at least eight characters long and up to sixteen characters long.

VI. Security codes storage

The user is able to store and keep safe his security codes such as Google Account backup codes, GitHub Account codes, etc.

This will be also synchronized and saved in Google Drive.

2.3. User specs

The final user that will have this application installed must be the owner of the smartphone or having knowledge of the master password in order to unlock the application. Also, the owner of the smartphone is able to unlock the application with his fingerprint, making access simpler and faster.

2.4. Restrictions

The smartphone should have at least Android KitKat (4.4) and Internet connection (for web access. In other case, no need of connection).

Initially, the hardware requirements can be anyone, but at least 256MB of RAM and dual-core processor.

2.5. Supposals and dependencies

This app needs Android OS installed in order to run properly. Any other systems will not work and may cause issues. Also, it is highly recommended to have Google Play Services installed in order to store your data and synchronize your information. If they are not available, some functionalities will not be able.

2.6. Proposed requirements

The application is defined by some requirements that are the most useful, but we are open to suggestions in order to be implemented in the future. Just create an “Issue” and post what you would like to be in the application for the next upgrade: https://github.com/Javinator9889/SecurePass/issues

3. Specific requirements

Here all the specific requirements will be exhaustively described in order to avoid problems in the future.

3.1. External interface requirements

3.1.1. User interface

The application will show a list with items to the user. Each item can be modified, deleted and viewed. There will be also a main menu where the user will be able to setup the application and its services, such as Google Drive.

There will be different screens:

1. *Main screen*: the user will be able to see the data he has stored in the application. In addition, there will be tab buttons to switch between categories.
2. *Settings screen*: the user will be able to customize some parts of the application, and setup Google Drive with his Google Account. In addition, he will be able to define custom categories in order to distribute better the stored passwords and accounts. Also, the user can donate the developer if he enjoys with our work.
3. *Security codes screen*: here there will be all the account security codes that the user have created. As in the main screen, he will be able to edit, delete and view different items.
4. *WiFi screen*: In this screen, the user will be able to configure a WiFi network with strong encryption with an IP address and a necessary password. From this screen, the user will be able to connect and see data from a web interface.
5. *License screen*: here will be displayed all used libraries and TOS, also some relevant information about the application.
6. *Optional screens*: the different screens that have not been previously contemplated but exists in the application.

3.1.2. Hardware interface

As mentioned in 2.4, the app will need these hardware requirements in the Android device.

In addition, the computer that will display the web page in WiFi mode will need connection to Internet in the same network as the smartphone and a compatible navigator, such as Google Chrome [13] or Mozilla Firefox [14].

3.1.3. Software interface

The application will use Google Drive API and Firebase API. First one for backing up user-data and second one for crash reporting and useful information to developer, in order to correct *bugs*, solving problems and offering better updates to the user.

3.1.4. Communications interface

The software requires always a secured network, for backing up data and for web transmission between devices (see “WiFi screen” on 3.1.1). If this network is not available, the connection will close and stop all transfers. This is to persevere user privacy and security.

3.2. Functional requirements

Here are described exhaustively all the requirements that the application must have to work properly. If one of the following fails, the application will crash or probably some functions will not be available or incorrect.

3.2.1. User initial setup

- **Introduction**: the user is able to setup the application the first time.
- **Inputs**: for the user a master password will be recorded.
- **Process**: the hashed master password will be saved in a database in order to check that the user using the app is the owner of the smartphone.
- **Outputs**: user data will be stored and now the user is able to use the application.

3.2.2. Sync to Google Drive

- **Introduction**: once the user has created the account, he will be able to synchronize the data with Google Drive.
- **Inputs**: the user will be requested to choose a Google Account that will be used with Google Drive.
- **Process**: the chosen account will be saved and an encrypted folder will be created in Google Drive.
- **Outputs**: auto-synchronization is enabled after the user has correctly linked his Google Account.

3.2.3. Store accounts and passwords

- **Introduction**: the user will be able to save his accounts and passwords.
- **Inputs**: in the main screen, the user can create a new entry with different fields (such as "Account name", "password", etc.). The password field must not be empty.
- **Process**: the data entered by the user will be securely saved in an encrypted database.
- **Outputs**: the new entry is saved.

3.2.4. Modify accounts and passwords

- **Introduction**: in a created entry, the user can modify its data.
- **Inputs**: once the user has clicked an entry, he will be able to modify its data or delete it. If the user change its data, password field must not be empty.
- **Process**: the new data is saved, or if deleted, the entry is removed from database.
- **Outputs**: the modification is saved and the main screen has changed.

3.2.5. Save user security codes

- **Introduction**: the user can save his security codes for a specific account in the application.
- **Inputs**: the user can give the application a plain-text document ("`.txt`") or put the different codes manually into different fields.
- **Process**: if the user gives a file, it is securely stored in the application private data. Else, the fields will be saved in a database.
- **Outputs**: the security codes are saved and available for the user.

3.2.6. Categories

- **Introduction**: the application will have different categories to organize the different entries.
- **Inputs**: the user can create or delete any category. There will be always a category that will have all entries.
- **Process**: the different categories are saved in a database and synchronized with the entries. Each entry must have a category. In addition, the user can define a "default category". If not, the system will assign a default category by itself.
- **Outputs**: the categories are defined and the main screen is distributed in tabs with one category per tab.

3.2.7. Web display

- **Introduction**: the user can see his codes in a web browser by accessing to a specific IP address.
- **Inputs**: the user will define a password for the output connection.
- **Process**: the smartphone will allocate a web page and listen to a specific port. It must be connected to a WiFi network. If not, web display will not be available.
- **Outputs**: the user is able to see all his data with all capabilities if connected via web browser.

3.2.8. QR code

- **Introduction**: the user can generate a QR code with plain text for the password or one that contains all the necessary data.
- **Inputs**: the user can choose what to export in the QR code.
- **Process**: the SP app will show an image that can be exported and saved, so the user can share it lately. Also, there will be an option for share now the image.
- **Outputs**: the user has a QR code for sharing. In addition, this code will be saved so the user can restore it instead of generating it again.

3.3. Performance requirements

The application should load everything fast, and web display must be fast. It is very important to optimize processor load and access times to database as this application is designed to work with low-end devices.

3.4. Design restrictions

The SP application must be accord to the Material Design system and adapt to different screen sizes and resolutions. Also, it should work in tablets.

The web display must be responsive or, failing that, be optimized to view it correctly, allowing zoom.

3.5. Software system characteristics

SP application must provide/be:

- **Secure**: all data used by SP is sensitive, so it must be treated with the actual highly standards of security. Data will not be able to any other corporation/entity/third party user (even the developer of this application) keeping the user privacy and data safe.
- **Availability**: SP application must be available for the user whenever he wants to use it. Google Drive services depends on Google status.
- **Optimized**: the application load on system and its repercussion on Android system must be the less possible.
- **Reliability**: the application should not have any unexpected crashes or bugs.
- **Maintenance**: according to the previous characteristic, the SP app will have maintenance for solving problems or adding different features that the users request or the developer consider necessary.
- **Scalability**: the SP application must work OK independent of the number of entries or the fields provided by the user. This point has strong relation with *Optimized* point in this list.
- **User friendly**: apart of point 8 (*Design restrictions*) in this guide, the application must be honest with the user and not hide its behavior under any circumstances, so that the user is aware of the impact of his actions.

3.6. Other requirements

SP must have all its connections secured, so for the web display HTTPS must be used or any cypher method.

In addition, the application must be self-maintained and optimized, cleaning unnecessary files and keeping the database ordered (for a better performance).

4. Appendix

Different diagram models are given at the same page as you can find this document, such as ER model and database model, for implementing by yourself.

5. Index

6. Signature

24/03/2018

X Javinator9889

Javinator9889
Developer
Firmado por: Javinator9889

This signature ensures that the document you are reading has not been modified and it is the original one.

Last modifications done on: 24/03/2018 at 13:43

Javinator9889 |