

How to Use this Template

1. Create a new document, and copy and paste the text from this template into your new document [Select All → Copy → Paste into new document]
2. Name your document file: “**Capstone_Stage1**”
3. Replace the text **in green**

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

GitHub Username: Slupik

SavePass

Description

Log in into different services with a different password may become cumbersome. Maybe on a few websites do you have multiple accounts? It's even more passwords! It's really hard to remember all these long, random chains of chars. But as you may know, that's not everything. All passwords should be changed in the short period of time. Can you imagine that every month all your passwords should be changed? How long would you try to remember all of them?

With this app, all your passwords will be saved on your device and available at any time.

Functions:

- Online backup

- Passwords encryption
- Reminding about changing passwords
- Fast & stable app
- Ergonomic design

Intended User

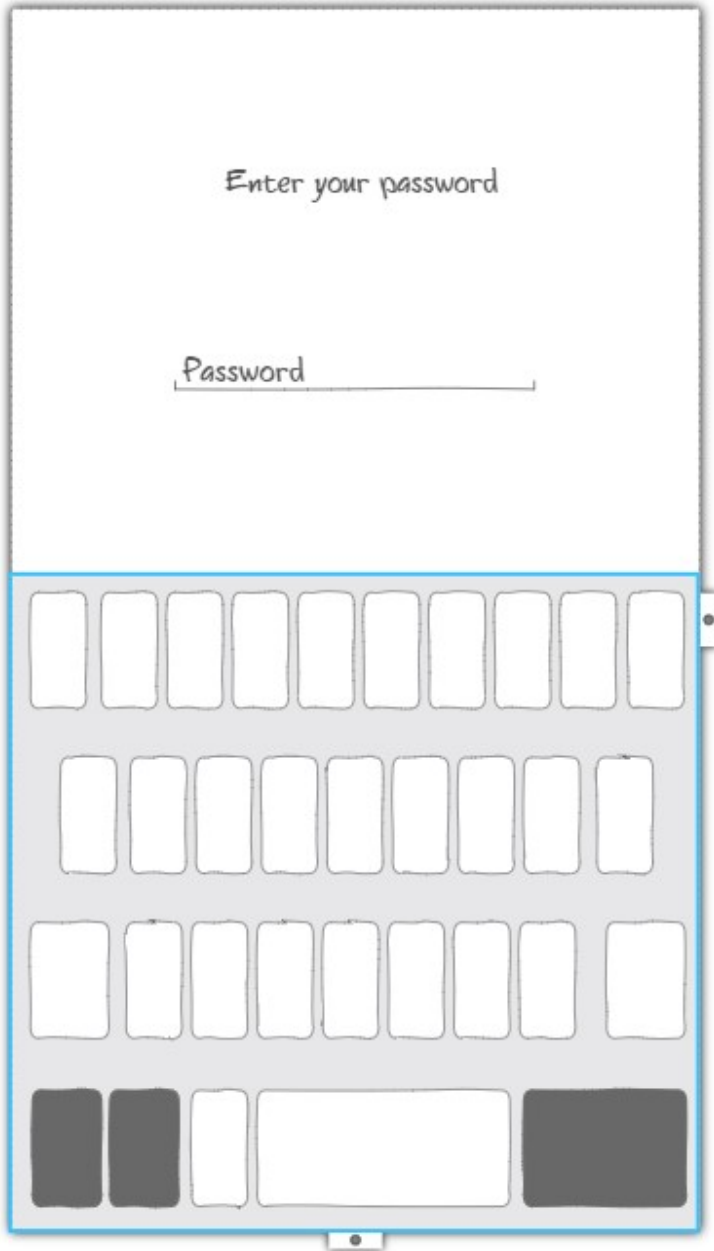
Average internet user especially with the big amount of accounts. It seems to me that main part of my users will be people 14-30 years old.

Features



















- Saves passwords and addition information
- Makes a backup of stored data
- Remembers about a password change

User Interface Mocks



MainPassword - first screen




List of passwords


SavePass	
	Page name short desc
	Two-Line List Item With Icon This is some text.
	Two-Line List Item With Icon This is some text.
	Two-Line List Item With Icon This is some text.
	Two-Line List Item With Icon This is some text.
	Two-Line List Item With Icon This is some text.
	Two-Line List Item With Icon This is some text.
	Two-Line List Item With Icon This is some text.
	Two-Line List Item With Icon This is some text.
	Two-Line List Item With Icon This is some text.
	Two-Line List Item With Icon This is some text.
	Two-Line List Item With Icon This is some text.
	Two-Line List Item With Icon This is some text.
	Two-Line List Item With Icon This is some text.
	Two-Line List Item With Icon This is some text.
	Two-Line List Item With Icon This is some text.
	Two-Line List Item With Icon This is some text.
	Two-Line List Item With Icon This is some text.

Password details




 name.com

Username



Password



Change password after

10

 days

Short desc

Lorem ipsum dolor sit amet, consectetur
adipisicing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut

Notes

Lorem ipsum dolor sit amet, consectetur
adipisicing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation

AddPassword

✕

✓

Name

Address

Username

Password

Generate password

☐ Only numbers

☒ Numbers + letters

☐ All chars

Password strength

Change password after 10 days

Notes

☒ Sync with server

Settings

Server settings

Username

Password

Synchronization settings

Sync per day

1

Password change reminder

ON

Send notifications

Default time

Widget

Accounts with old passwords	
Webpage Login	5 days
Wepage Login	4 days
Two-Line List This is some text.	4 days
Two-Line List	3 days

Key Considerations**How will your app handle data persistence?**

I will use the Room library. Passwords with other info should be saved in database and settings in shared preferences.

Describe any edge or corner cases in the UX.

It's a simple app... But if I must describe although one case then let's assume that user will enter a huge number of days in AddPassword screen. Then it's a good idea to inform him that's better to enter 0.

Describe any libraries you'll be using and share your reasoning for including them.

Picasso (2.71828) – loading webs favicon into ImageView
 Junit (4.12) – unit tests
 Espresso (3.0.2) – UI tests
 ButterKnife (8.8.1) – binding UI
 AndroidSupportLibrary (r6) – backward compatibility
 Gson (2.8.5) – parsing json
 Retrofit (2.4.0) – communication with custom server
 AdMob (15.0.0) – ads
 Google Analytics (v3-rev157-1.23.0) – collecting interesting data
 Room (1.1.1) – data persistence

Describe how you will implement Google Play Services or other external services.

Google Analytics – collecting data about app usage plus collecting crash reports

Google Mobile Ads – ads 😊

Not interesting information which I must include to pass the first stage. ;)

1. App is written solely in Java programming languages.
2. App uses stable gradle (v4.9) and Android Studio (3.1.3) version.
3. App uses JobDispatcher to upload backup in the background.
4. App will have included content descriptions, navigation using a D-pad and enabled RTL layout switching.
5. All app images will be downloading via Picasso and cached locally by this library.
6. App string will be placed in the strings.xml file.
7. App Theme will be placed in the appropriate file and will be extending the AppCompatActivity theme.
8. LiveData and ViewModel will be used when required and no unnecessary calls to the database will be made.

Next Steps: Required Tasks

Task 1: Project Setup

- Configure libraries
- Add keystroke
- Configure gradle

Task 2: Create database

- Make a database with Room library
- Add database manager to easily operate on encrypted data

Task 3: UI

- Build UI for all activities but only for portrait version

Task 4: Working UI

- Connect UI to the database

Task 5: Online backup

- Create a system for communicating with the server
- Create background service making the backup of passwords

Task 6: Widget

- Add widget (list of accounts with old passwords) connected to the local database

Task 7: Polishing UI

- Add other variations of existing UIs (landscape, tablet etc.)
- Add simple animations

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
 - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"