

## 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:** issa12345

# App : Anime Center

## Description

Application Anime Center is an application that combines the best developments and fame in the world of anime. This app is easily defined. Top user rating and popularity This application deals with the json file, provides the user with the story of Anime and contains an assessment of the name of anime, In determining the development and evaluation.

## Intended User

The user of this application are lovers of follow-up anime so that it is easy to know the highest ranking and knowledge of the story of the anime and details

## Features

- Write the application only in Java programming language
- The pictures appear
- Handle the Jason file
- Details appear
- Anime classification
- Other features

## User Interface Mocks

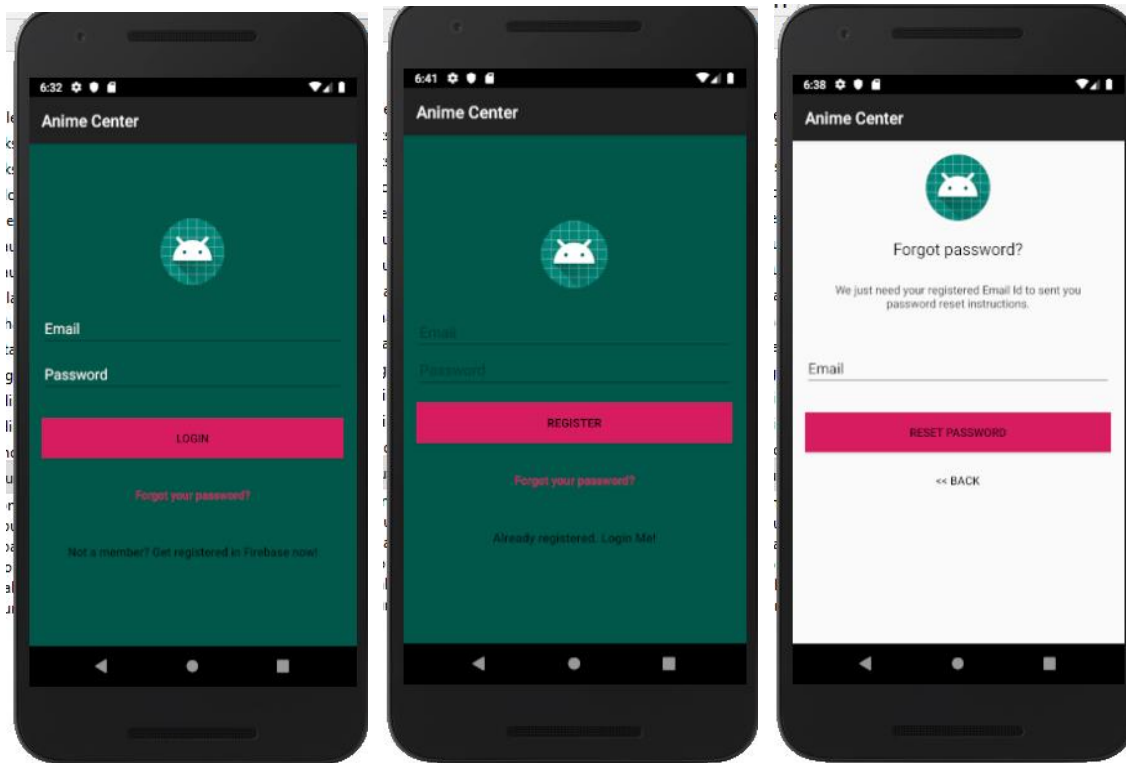
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, [www.ninjamock.com](http://www.ninjamock.com), Paper by 53, Photoshop or Balsamiq.

### Screen 1



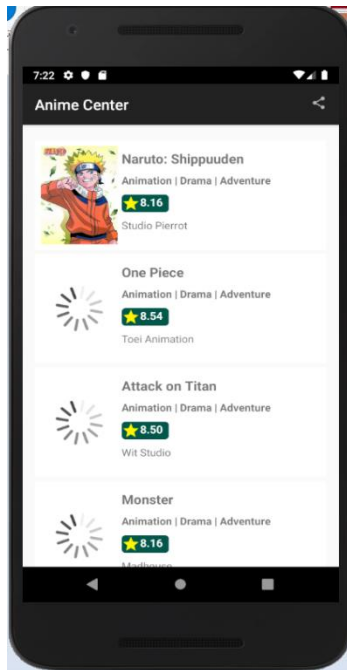
splash screen is the welcome page with an image and the text of the app name pops up for five seconds

## Screen 2



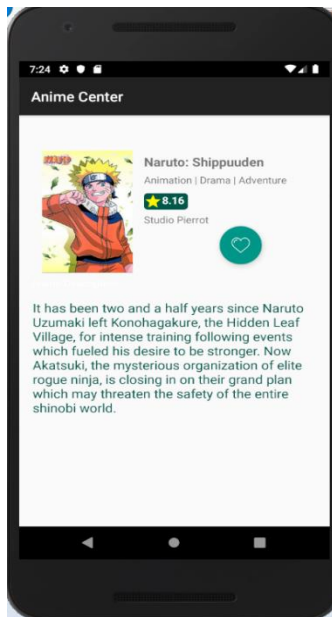
This is the screen after the splash screen which is used to authenticate users using firebase google login. The "Sing up" screen also allows the user to record his data in the user database as well as the user data recovery page

### Screen 3



This screen contains the anime information imported from the JSON file, namely, images, anime name, classification and classification, as well as at the top of the screen, the application's share button is displayed in vertical order and some buttons are added back

### Screen 4



This is a page that displays the details of anime selected from the user to an application that displays the story of anime and contains a picture name, rating, rating and anime story

## Key Considerations

### How will your app handle data persistence?

To fetch the hero details AsyncTask, to access the details of the application, the Internet connection will be used in a file JSON , and to register the user we will need the Firebase database

### Describe any edge or corner cases in the UX.

Display appropriate error messages & alerts to the user.

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

#### Library use

```
Implementation 'com.android.support:appcompat-v7:28.0.0'
implementation 'com.android.support.constraint:constraint-layout:1.1.3'
implementation 'com.github.bumptech.glide:glide:4.7.1'
implementation 'com.github.ivbaranov:materialfavoritebutton:0.1.5'
annotationProcessor 'com.github.bumptech.glide:compiler:4.6.1'
implementation 'com.android.support:design:28.0.0'
implementation 'com.android.volley:volley:1.1.0'
implementation 'com.android.support:recyclerview-v7:28.0.0'
androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
testImplementation 'junit:junit:4.12'
implementation 'android.arch.persistence.room:runtime:1.1.1'
implementation 'me.zhanghai.android.materialratingbar:library:1.2.0'
androidTestImplementation 'com.android.support.test:runner:1.0.2'
implementation 'com.google.android.gms:play-services-ads:18.0.0'
```

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

I Use – admobs to display Ads to the user.

I use –Firebase Google signIn to authenticate the users.

## Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

### Task 1: Project Setup

#### ❖ Create Project

- Create a project, name it Anime Center
- Configure the dependencies.

### Task 2: Fetch Data form the API Properly

- Implement the code to properly fetch JSON Data from URL.

### Task 4: Implement UI for Each Activity and Fragment

- Design layout for authentication with firebase database.
- Design layout for grid list of images.
- Design a layout for the application's share button activity
- Design menu
- Display ads

### Task 5: Create screen to show Widget the application

- Design and implementation of the main screen Widget to show the application's title when running the application, and then go to the user's user page and click on it and go to a page that displays a vertical list of the application content

### Task 6: Create design of the Share

The design of the Share button at the top of the application allows the user to share the application with friends

## Task 7: Create a list:

This menu contains the option to share the application and the Logoff and Settings button that appears when you click in the database.

## Task 8: Create design content

The layout of the vertical layout to show the content details of the application of images and backgrounds give a wonderful character to the application

Add as many tasks as you need to complete your app.

---

### 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**"