

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Screen 3](#)

[Screen 4](#)

[Screen 5](#)

[Screen 6](#)

[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: [mohammedAlefrangy](#)

AnimeApp

Description

This application is a program to display movies and anime series, where the user can find millions and millions of movies and serials, and to offer favorite movies and follow-up in the event that the Internet is not available. Users can also find many types of movies with just two of them: anime movies, drama, genres, movie details, date of publication, number of favorite users, and videos and photos.

The application solves the problem of users who do not know how to search for anime and get the best and newest anime series and determine which ones will watch and people who struggle to find new movies.

Intended User

Who is your intended user? (For example, is this an app for dog owners? Families? Students? Travelers?)

All anime lovers.

I am using Android Studio 3.1.4

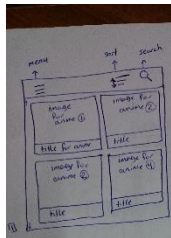
Features

- Application includes content descriptions
- The application is written only in the Java programming language
- The application can handle RTL format and change the orientation from portrait to landscape.
- Sync Adapter for regularly updated information about popular TV series
- The application retains all strings in the strings.xml file and can format RTL so that all schemas are run.
- widget provides widget to provide relevant information to user on home screen.
- The application checks all inputs from servers and users. If the data is not present or in a wrong format, the application will record this fact and will not crash.
- Information about the crew, movie, presentations, or movies.
- Search for movies or TV shows
- Tell your favorite movie user every seven days.
- Work offline for favorite movies

User Interface Mocks

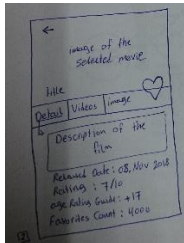
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Screen 1



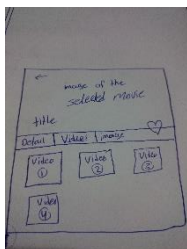
The application home page of the application where the user can select a movie that wants to know its details from the list.

Screen 2



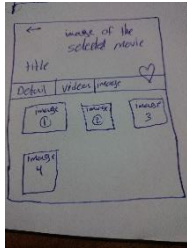
A detailed page of the selected anime film where the user can read a summary of the film and the date of publication and the number of people who like it and the legal age to watch it and can be placed as a favorite movie in the Favorites list

Screen 3



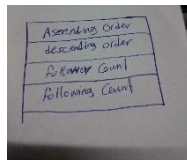
A detailed page of the selected anime film where the user can watch the related videos in the movie by YouTube

Screen 4



A detailed page of the selected anime film where the user can view the related images in the movie's scenes

Screen 5



A dialog box that allows the user to arrange movies by several options available to him

Screen 6



The list contains the types of anime movies, the favorite list, the image of the application, its name, and information about the programmer who programmed the application

Key Considerations

How will your app handle data persistence?

I will build a content provider.

Describe any edge or corner cases in the UX.

Handling the different film and the images and the videos.

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

For example, Picasso or Glide to handle the loading and caching of images.

- Gradle 3.2.1
- Picasso 2.7: image loading.
- JSON Object: web api calls.
- Android Architecture Component (Room, ViewModel, and LiveData): database

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

Admob will be added as Google Play Service

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 the project in Android Studio

- Configure Libraries such as Picasso, Android Architecture Component ,etc.
- Setup the required libraries in build.gradle files
- Setup the java folder structure in a logical manner
- Load PNG files into drawable folders including app ICON.
- Create git repo for the project
- Design database schema for user's data storage

Task 2: Implement UI for Each Activity and Fragment

- 1. Build UI for Launch activity
- 2. Build UI for different tabs on screen

Task 3: Implement Network calls

- Create classes for anime api for retrieving data
- Test the network calls
- Async Adapter in order to regularly get updated info about anime series.
- This data will be used later to feed the widget provided by the app.

Task 4: Implement logic and data

- 1. Add a favorite movie to the database for offline viewing
- 2. Add sort filters in listing page

Task 5: Implement Google Play services

- 1. Add admob for banner ads Something else
-