How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: “**Capstone_Stage1**”
3. Replace the text in green

Submission Instructions

1. After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
3. Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”

GitHub Username: jreyes

AniTrak

Description

There are many series being aired in Japan right now and many upcoming series as well. As an anime fan, this app will help fans to keep tabs on current and upcoming series to make sure one won’t be missing great series like Gundam, One Punch Man, or Steins Gate.

Intended User

This app is for people that likes japanese animation and would like to keep track of upcoming anime series.

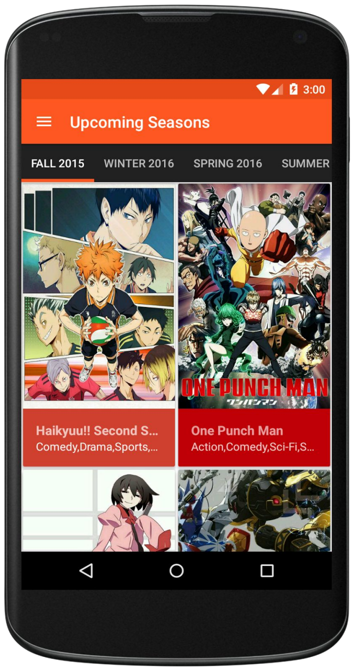
Features

List the main features of your app. For example:

- Keeps track of the current season TV series and future ones.
- You can add a serie to your "Currently Watching", "Plan To Watch", and "Completed" lists.
- Keep track of the current TV series in your homepage with a widget.
- Displays detailed information about a series to track.

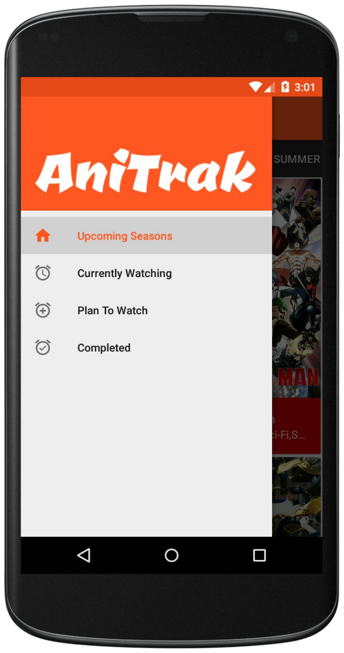
User Interface Mocks

Home Activity



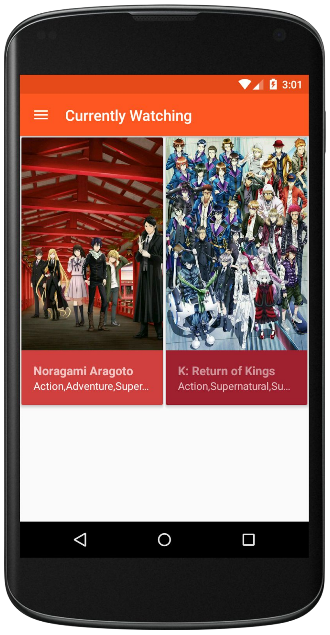
The main activity is launched when the application is started. It will contain tabs for the current season and future ones including a TBA tab (To Be Announced).

Navigation Drawer



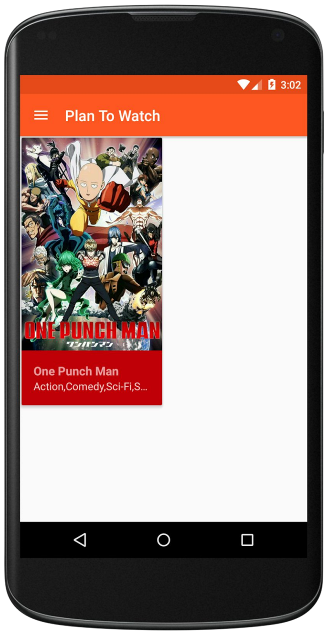
A navigation drawer that would contain links to the catalog of anime seasons, and the user's lists for "Currently Watching", "Plan To Watch", and "Completed" lists.

Currently Watching



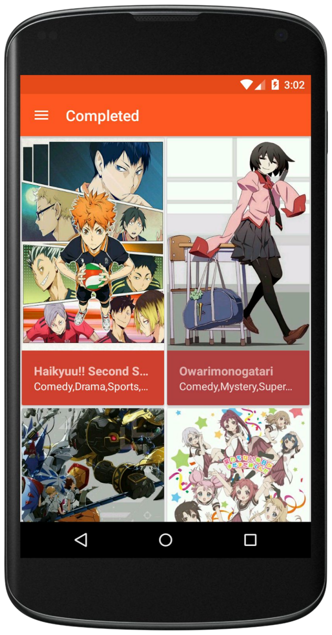
The "Currently Watching" screen that would display your currently watching series.

Plan To Watch



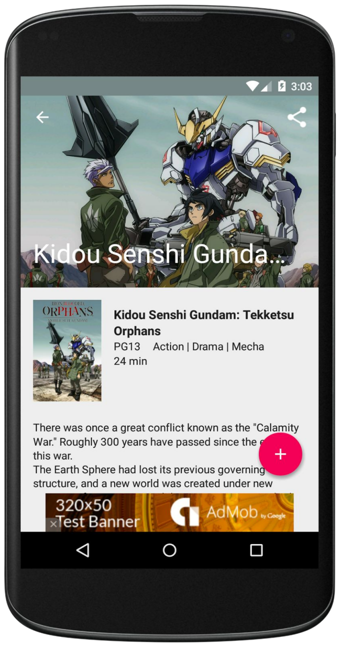
The “Plan To Watch” screen that would display your current plan to watch series.

Completed



The “Completed” screen that would display your current completed series.

Anime Details



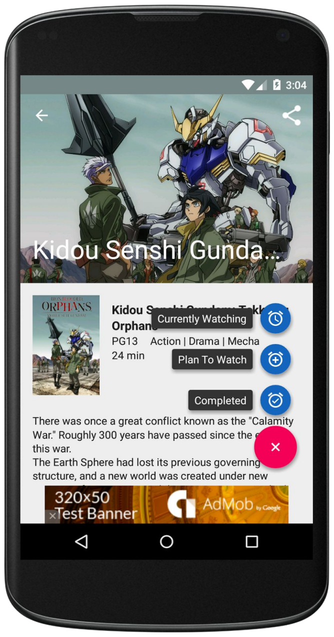
Kidou Senshi Gunda...

Kidou Senshi Gundam: Tekketsu Orphans

PG13 Action | Drama | Mecha

24 min

There was once a great conflict known as the "Calamity War." Roughly 300 years have passed since the e this war.
The Earth Sphere had lost its previous governing structure, and a new world was created under new



Kidou Senshi Gunda...

Kidou S Orphan

PG13 Action | Drama | Mecha

24 min

There was once a great conflict known as the "Calamity War." Roughly 300 years have passed since the this war.
The Earth Sphere had lost its previous governing structure, and a new world was created under new

The anime details screen where there will a poster for the series and detailed information like a synopsis, genres and episode duration. Also from this screen one would be able to add this serie to the “Currently Watching”, “Plan To Watch”, and “Completed” lists.

Key Considerations

How will your app handle data persistence?

Data persistence will be handle by the app’s Content Provider. At the launch of the application, an IntentService will be launched to synced that current anime series by season.

Describe any corner cases in the UX.

The app start in the MainActivity, from there there is a menu icon to expand the navigation drawer or the user can swipe to see the navigation drawer. From the navigation drawer you can navigate to the rest of the application. From the anime details screen there should be a FAB button that when clicked options get expanded to display options to add the current anime to a user list.

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

- Support libraries: cardview, recyclerview, design, palette.
- OkHttp for network communication
- Glide for image loading
- Retrofit for Rest support
- Play services: ads and analytics
- Clans' FAB: FAB menu

Next Steps: Required Tasks

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

Task 1: Project Setup

Configure libraries

Add libraries to build.gradle:
- com.android.support:appcompat-v7
- com.android.support:cardview-v7
- com.android.support:recyclerview-v7
- com.android.support:design
- com.android.support:palette-v7
- com.google.android.gms:play-services-ads
- com.google.android.gms:play-services-analytics
- com.github.clans:fab
- com.squareup.retrofit:retrofit
- com.squareup.retrofit:converter-gson
- com.squareup.okhttp:okhttp
- com.squareup.okhttp:okhttp-urlconnection
- com.github.bumptech.glide:glide
- com.github.bumptech.glide:okhttp-integration
- com.github.florent37:glidepalette

Get API keys

- Get keys for Google Ads and Analytics

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for NavigationDrawer
- Build UI for MainActivity
- Build UI for AnimeActivity
- Build UI for SeasonAdapter
- Build UI for SeasonPagerAdapter
- Build UI for SeasonPagerFragment
- Build UI for SeasonFragment
- Build UI for CurrentlyWatchingFragment
- Build UI for PlanToWatchFragment
- Build UI for CompletedFragment
- Build UI for AnimeFragment
- Build UI for SeasonWidgetService

Task 3: Implement ContentProvider

Define a class to hold the current content (Anime)

- Age rating
- Bayesian rating
- Community rating
- Cover image url
- Episode length
- Episode count
- Finished airing date
- Genres
- Id
- Poster image url
- Producers
- Season
- Show_type
- Slug
- Started airing date
- Synopsis
- Title canonical
- Title japanese

- Title romaji
- Title english
- User type ("currently_watching", "plan_to_watch", "completed")
- Youtube video id

Create the ContentProvider (AniTrakDB)

Create the contract (AnimeContract)

Create a CursorLoader (AnimeLoader)

Classes to create
- AniTrakDB
- AnimeContract
- AnimeLoader

Task 4: Build the UpdaterService

Service in charge to sync with a remote server the current series.

Task 5: Build the Rest client

Using Retrofit build a service to connect to a remote server and sync content (AniTrakApi)

Submission Instructions
1. After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
3. Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”