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**"

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: octario23

BroadcasTV

Description

BroadcasTV is an app that allow to see tv content in your phone so you don't miss any media program, you don't need to turn on your TV to see your favorite program, and you have different channels that you can watch and share with your friends.

Intended User

This app is great for any kind of users that are 5+ years old, since there's a big variety of channel content.

Features

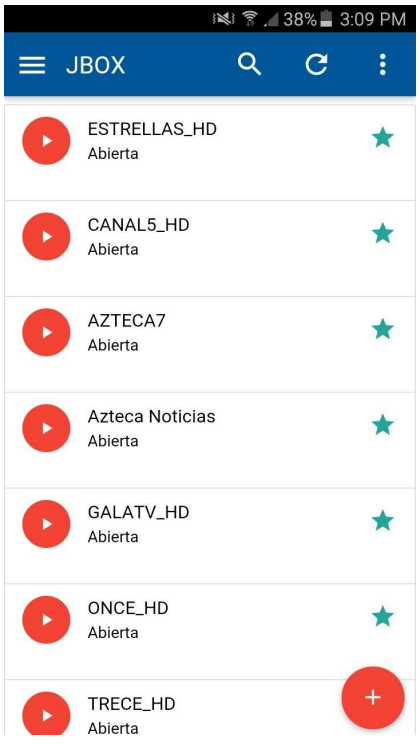
List the main features of your app. For example:

- Reproduce Online TV Channels
- Save favorite channels
- Share channels with friends
- Search channels

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 Photoshop or Balsamiq.

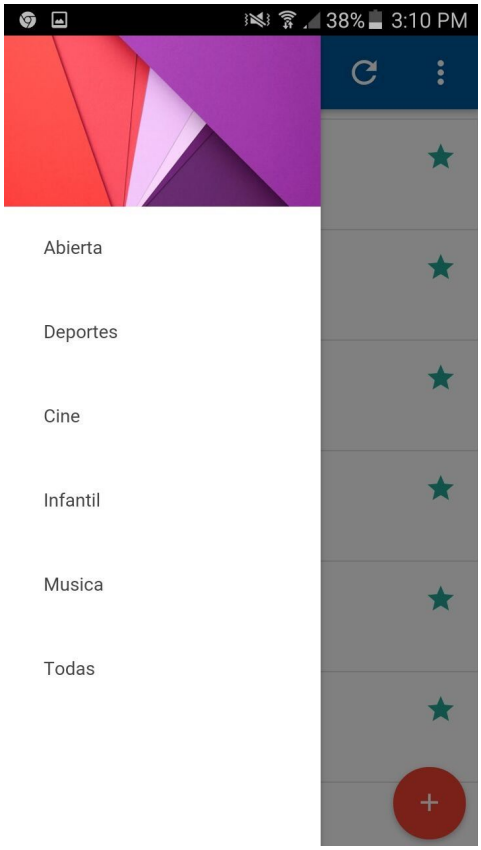
Screen 1



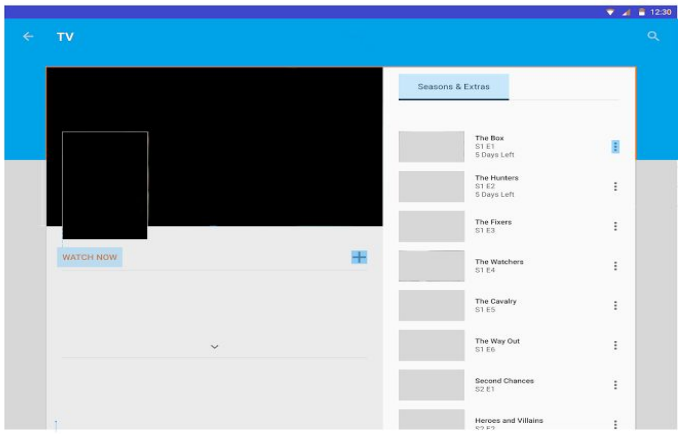
Main Screen for mobile version of app, this will be a list of channels with a FAB that will allow to add new channels, refresh button to refresh the list, search icon and a navigation drawer.

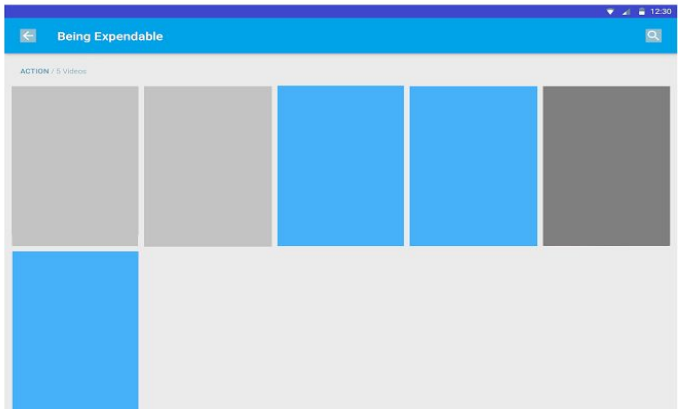
Screen 2

Video player of the application it will stream via channel URL.



Navigation Drawer with channel categories of the app.





Mock of the Tablet version of the app, it will display each channel as a card with an image and a detail of the channel with related channels based on category.

Add as many screens as you need to portray your app's UI flow.

Key Considerations

How will your app handle data persistence?

The data about the channels and the links the video player will use to stream will be stored in a sqlite database with a Content Provider and a Loader to communicate with the UI, each client will have the possibility to save favorite channels and the data of the channel will be updated biweekly.

Describe any corner cases in the UX.

The user will exit the video player if they press back button, if they want to return to the video player, then they will need to select the channel that was in playback.

User will be able to navigate channel categories using the navigation drawer so they don't have to see the full list of channels or they will see the option to go to favorite channels in case they want to see something from that category.

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

Picasso will handle the loading and caching of channel images (If available).

Material Drawer will be used to show a Material Drawer component having the channels separated by category, an option to see the favorite channels and the information of the user.

Describe how you will implement Google Play Services.

Location will be used in a future phase of this app to be capable of know the location of the user and show content based on where the user is.

Google Ads will be used to show a banner ad when the video player is about to start, and possibly a small banner ad when the user visit favorite section.

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 3rd party libraries
- Configure support library and other dependencies
- Design tables for database.
- Check API service to create table fields.
- Design content provider based on DB information.

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for MainActivity
- Build UI for Tablet version of the app
- Create Main and Detail fragment for both versions of app.
- Create Favorite channels fragment
- Store favorite information and create a list of favorite elements in the fragment
- Add transitions between fragments.

Task 3: Navigation Drawer integration and menu buttons development

- Integrate Navigation Drawer to the app
- Use webservice to get channel categories and show them as options of the navigation drawer
- Implement search functionality.
- Implement refresh button to get the latest list of available channels

Task 4: Detail Fragment development

- Based on the given channel show the detail of the channel, and suggested channels based on the category.
- Create a paralax effect when the user scroll down to see the list of suggested channels.
- Integrate Picasso library to load images
- Create a list of suggested channels

Task 5: Video player development

- Develop video player into the app
- Use URL to start streaming
- Add validation to finish video player activity once the user has press back button.

Task 6: Tablet development

- Create list of cards containing the available channels.
- Load images using Picasso
- Develop detail Fragment showing a description in the left with an image and show a list of suggested elements on the right of the screen.

Task 7: Favorite Channels App Widget

- Define information that will be displayed in the widget.
- Develop Widget and show information only if the user has at least 1 channel added to his favorite list.
-

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

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**"