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: Caroline Iacono

Mercury Test Application

Description

This is a simple Android application, following best Java and Android practices, that displays movie information.

Intended User

My company, which has sponsored my Udacity course, uses this app to evaluate potential candidates for hire. My boss requested that I complete this app so I may be judged by the same method as my co-workers.

In addition to the company-required attributes, I also implemented two Google Play services, analytics and location, as well as building a widget.

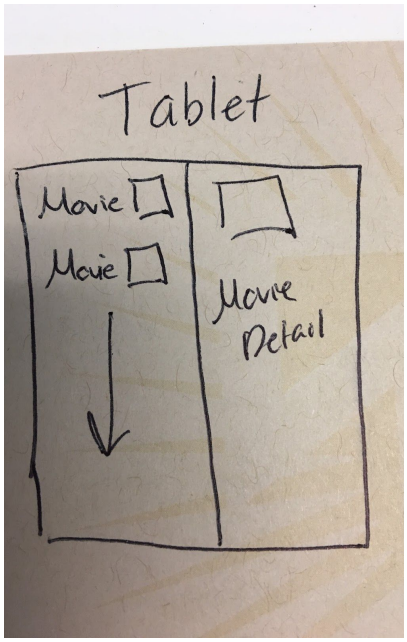
Features

- Display a list of movies items that scroll vertically. Each item should display the movie's poster art and all available summary info included in the JSON.
- The user will be able to tap a movie from the list to show detailed information about that movie on a second screen. The detail information should include all detail information included in the JSON.
- Images should be cached to reduce load on the server
- The app must support landscape and portrait orientations
- The app must handle any data or connectivity errors gracefully by displaying an error message to the user
- The minimum Android OS supported is 4.1

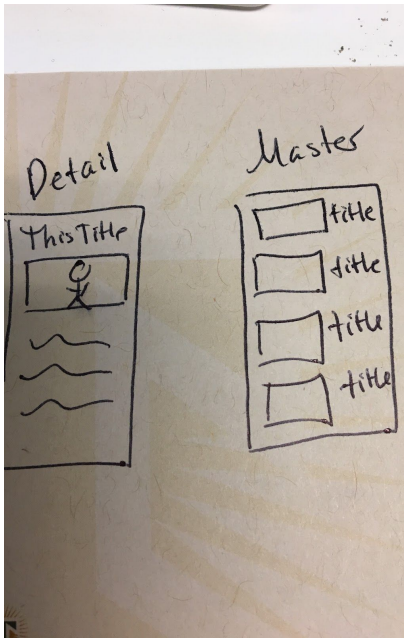
User Interface Mocks

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.

Tablet (master-detail flow)



Phone (two separate screens)



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

Key Considerations

How will your app handle data persistence?

My app will connect to a sample server using AsyncTask and store that data locally.

Describe any corner cases in the UX.

Upon exiting the app to view nearby theaters in the map app of the user's choosing, the back button returns the user to the app.

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

Picasso - image loader

Describe how you will implement Google Play Services.

I used Google Location to let the user know where nearby theaters are.
I used Google Analytics to track the usage of the app.

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

In order to make this app, I need several different components:

A Main Activity to launch the main list of movies and/or the empty detail fragment in the case of a tablet view

A class for the main MovieItems

A class for the MovieSummaries

A list adapter for the arraylist of movieItems

A DataUtil class for connecting the app to the server

Two different tasks for calling and parsing the data, since the detail's call depends on the imbd resource found in the first call

An Analytics Application for the Google Analytics

Widget information (data fetching, intent service, and view—uses Content Provider and Loaders)

And Google Location implementation.

Task 2: Implement UI for Each Activity and Fragment

- The view for the MainActivity requires a MovieListAdapter, since it's a RecyclerView.
- The data for the MovieInfoFetchTask (for the MovieSummary) is located in the MovieSummaryFragment, which also inflates the layout. That is the only fragment in the app.
- I need to create two versions of the main layout for the tablet version and the phone version (large and regular).
- Create a CardLayout for the individual MovieItems in the ArrayList.
- Create a Details_Screen layout for the details in the detail flow.
- Create a movie_widget view for the widget in the layout and an info view in the xml layout.

Other Tasks

- Add the DetailActivity and Widgets to the Android Manifest
- Add Picasso and the Google Analytics and Location services to the Gradle files
- Add RecyclerView and CardView to the Gradle files
- Add all string resources to the string files
- Add logs and checks for connectivity and errors
- Implement Google Location services and add a query for nearby theaters
- Add widget logic so the widget displays a random movie poster and title

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**"