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: scottdurica

Trippin

Description

Planning on traveling in or around Boston and don't want the hassle of dealing with rude drivers and gridlocked streets? Good news! Boston has one of the world's best mass transit systems and Trippin offers simple and useful access to everything you'll need to make life easier while traveling throughout the city.

Trippin has a simple and intuitive user interface. Some features include:

- Saving frequently traveled routes for quick access.
- Complete list of stations, searchable by name.
- Subway, commuter rail, ferry, and street maps.
- App widget to see current route on app screen.
- Easily share your itinerary with friends or family to ensure safe arrival.

Intended User

Trippin is an app designed for commuters traveling into, out of, or throughout Boston and the surrounding areas.

Features

Trippin has the following features:

- Accesses real time data from the MBTA API
- Stores saved travel itinerary from the user
- Provides maps to the user

User Interface Mocks

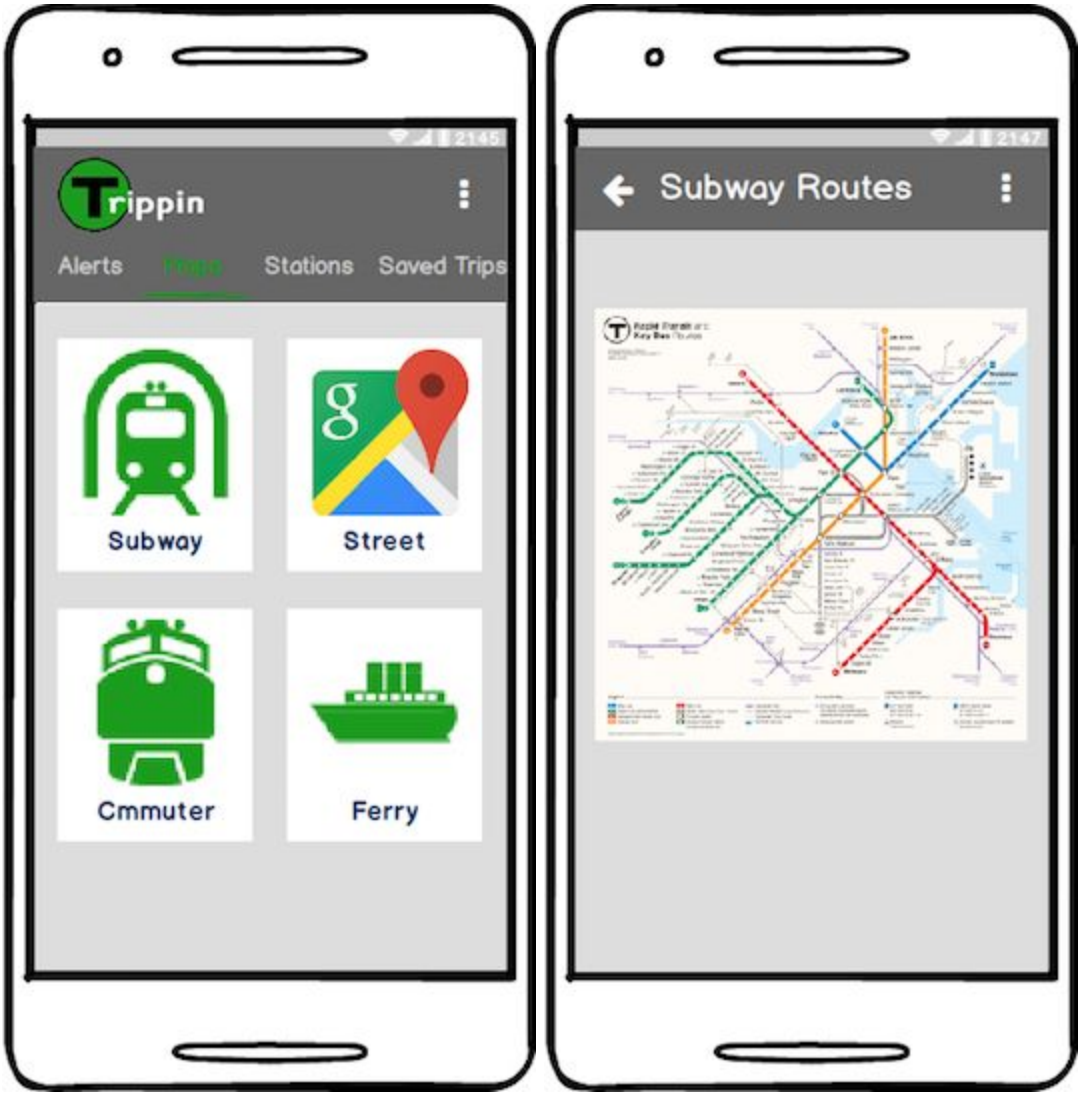
Main Screen-Tabbed layout

Launched Activity. Display current(or last selected) trip. If no persistent data exists, screen will let the user know that there are no trips selected and prompt user to choose an existing trip or create a new one. Share intent will be launched to allow user to choose an application to share their trip data with friends or family for safety's sake.

Tab 2- Service Alerts

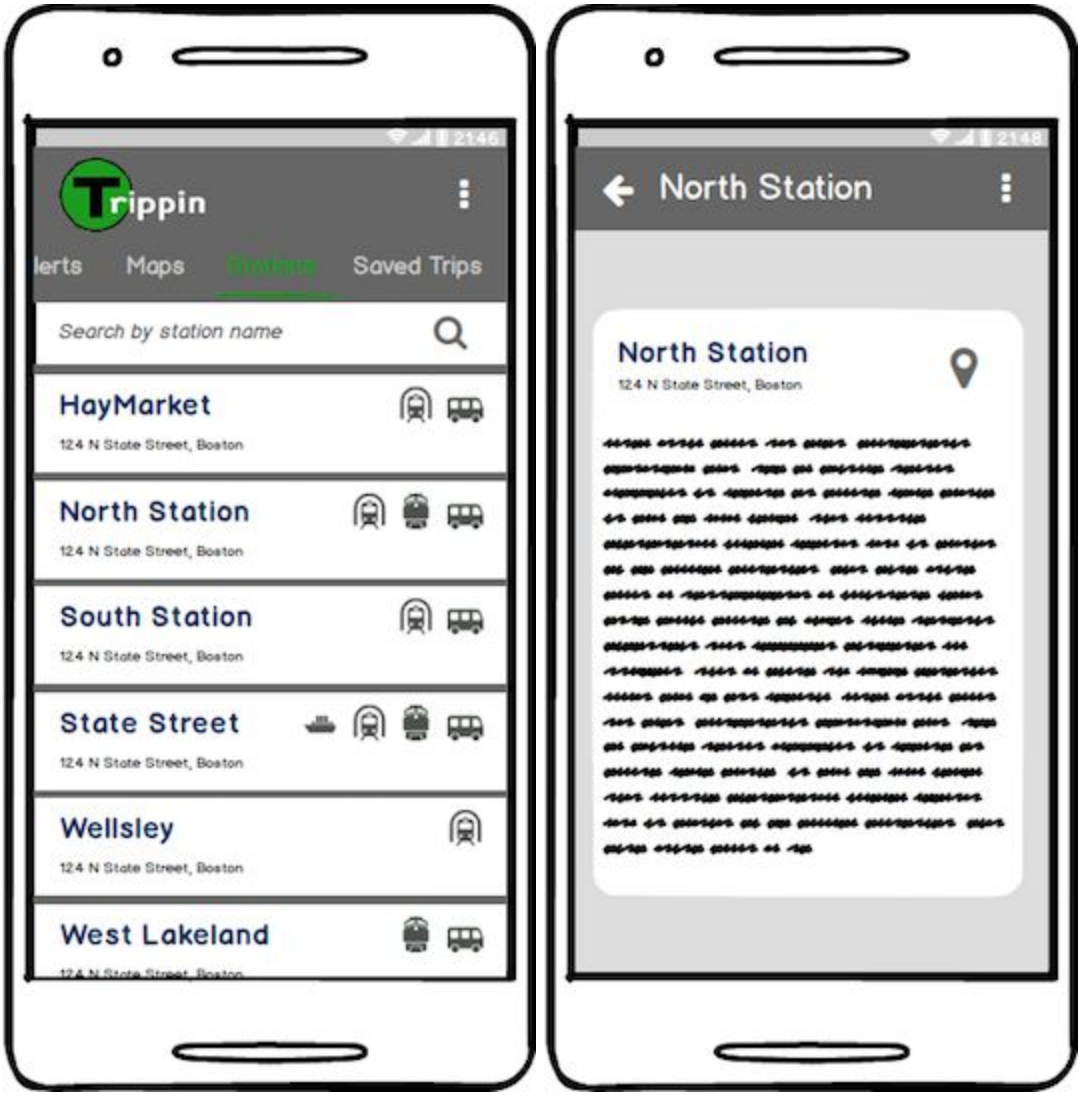
List of service alerts provided by the API. Alerts displayed are based on current selected trip. No further navigation from this page. Items are not selectable as they already display all data there is to offer.

Tab 3- Maps (click Map to go to Map Detail Screen)



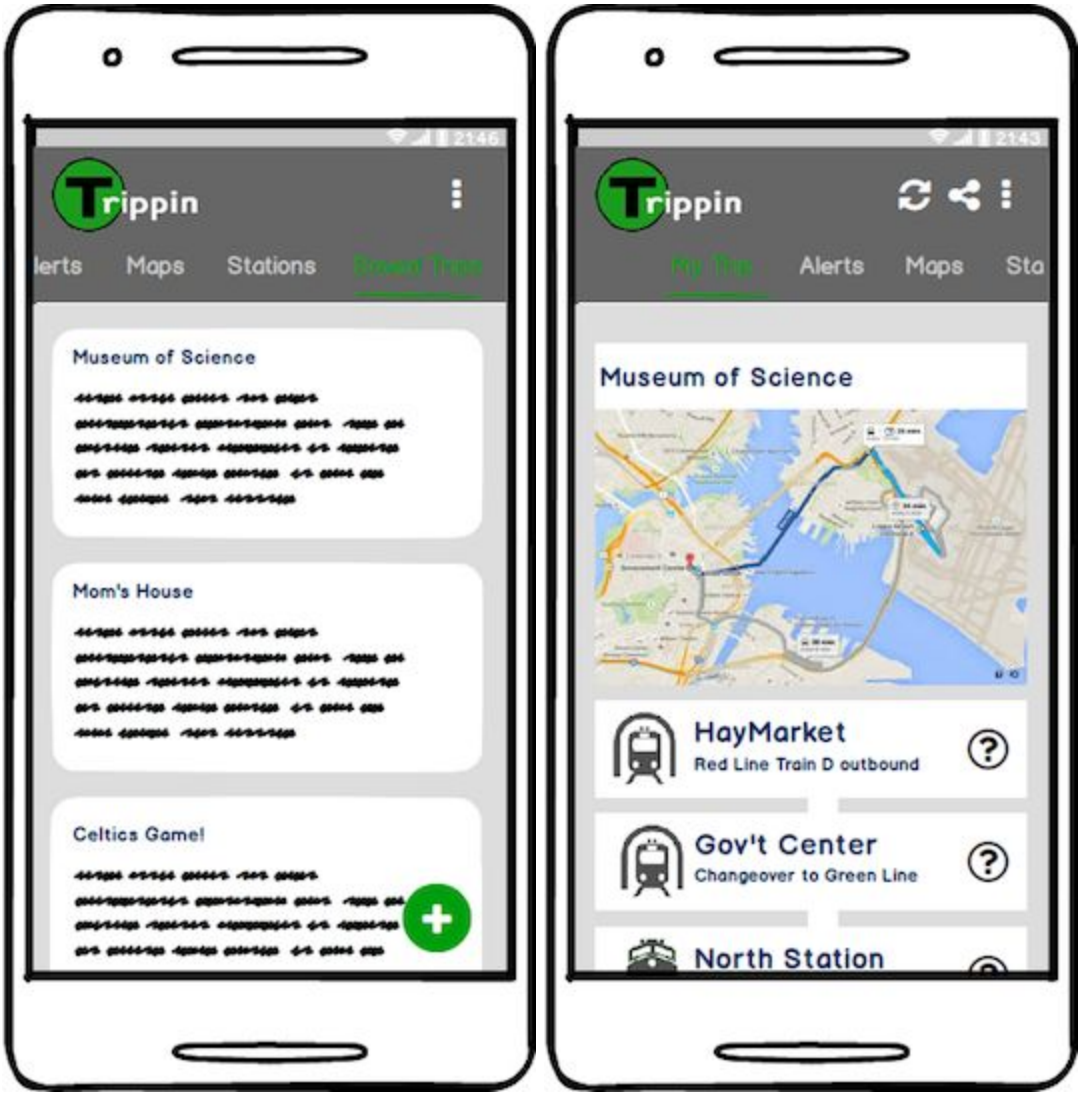
Clickable list of available maps. Click on choice to go to map detail screen.

Tab 4- Stations (click Station to go to Station Detail Screen)



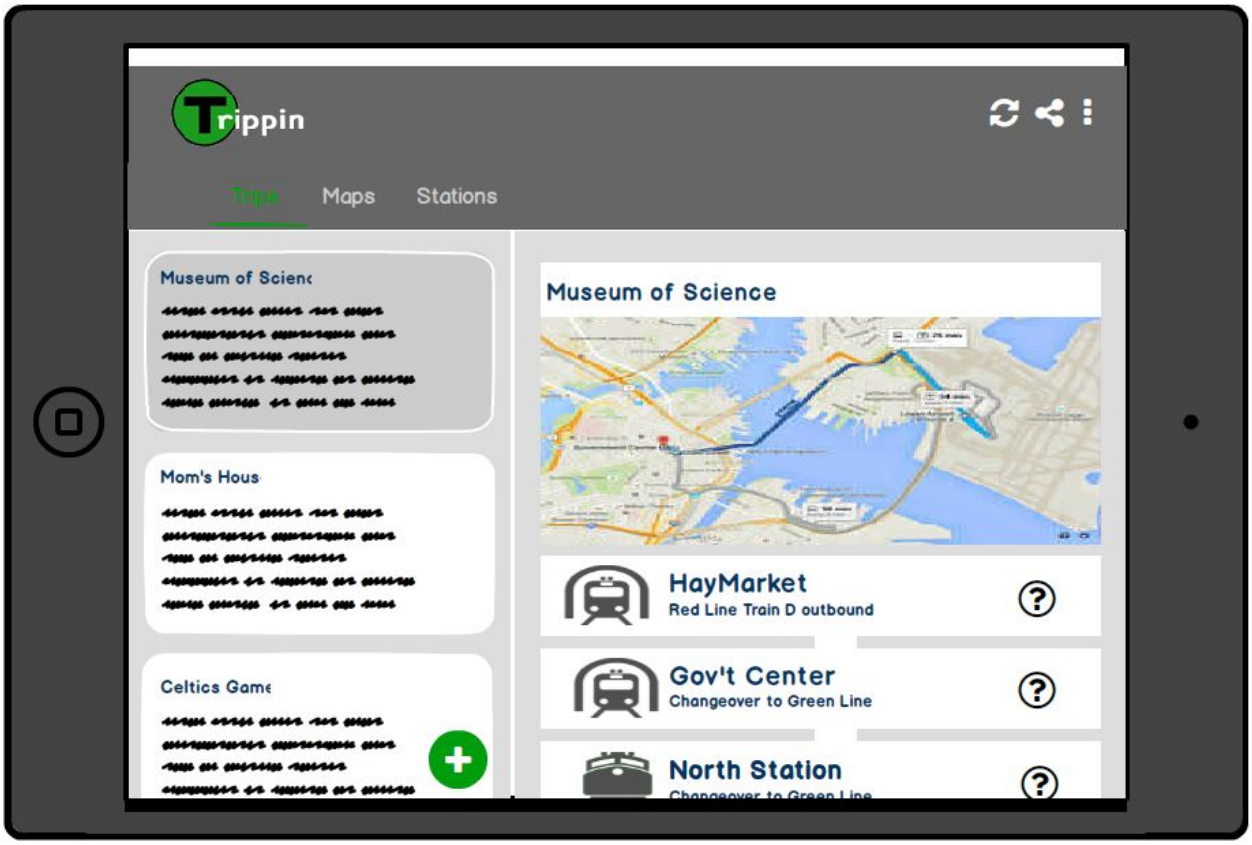
Scrollable and searchable list of stations provided from API. Click on station to go to the detail page.

Tab 4- Saved Trips (click trip to go to My Trip tab)



Provides a clickable list of saved trips. Click to choose trip as current trip. Long click to delete.

Large Screens- Tab 1 - Main Master/Detail

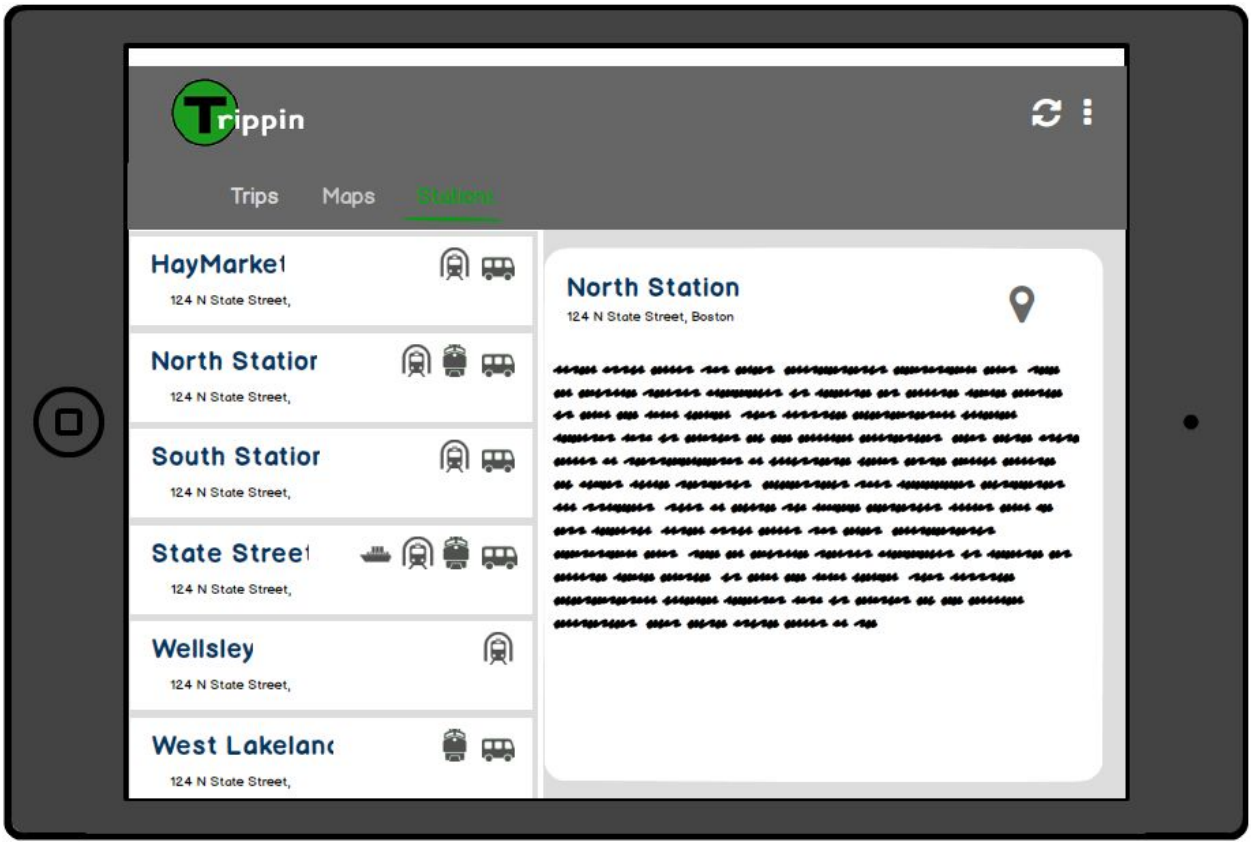


Main Screen for large screens. Shows list of saved trips and details of selected trip.

Large Screens- Tab 2 - Maps Master/Detail

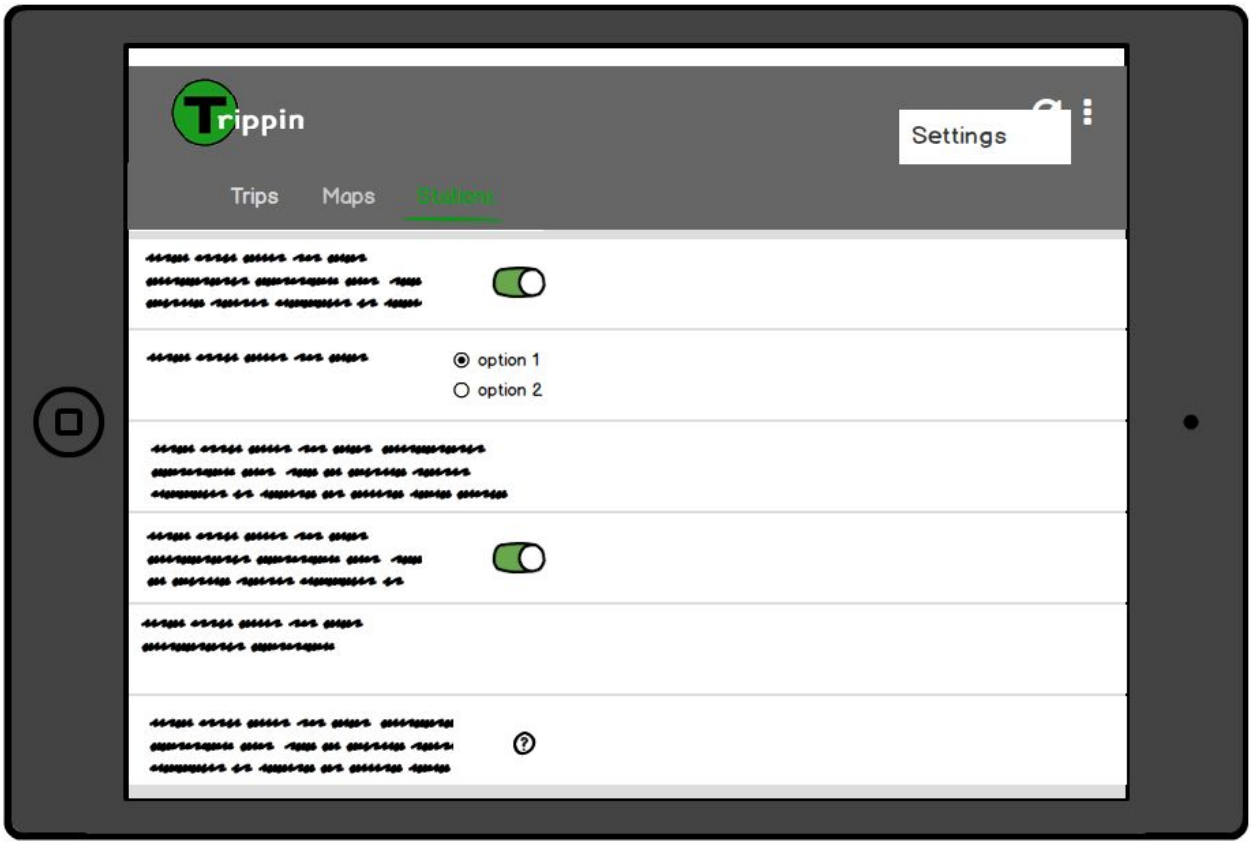
Master/Detail screen, showing map list and selected map.

Large Screens- Tab 3 - Stations Master/Detail



Master/Detail screen, showing stations list and details for selected station.

Large Screens- Settings



Settings screen.

Key Considerations

How will your app handle data persistence?

I will build a content provider for persistent data, and use SharedPreferences to persist user settings.

Describe any corner cases in the UX.

The user will be able to navigate through the main screens via side scroll or tab selection. Any of the screens that navigate away from the tabbed layout will have an back navigation to the previous screen.

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

Butterknife to simplify view binding.

Retrofit to simplify HTTP requests to the API.
Eventbus for inter-activity communication.
Picasso for any server side image loading needs.

Describe how you will implement Google Play Services.

Location services to detect and track the user's location along the chosen route.
Map services to display a proximity map to the user upon request.

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
- Configure Google Play Services
- Request key for API

Task 2: Implement UI for Each Activity and Fragment

- Build Tabbed UI for main application flow
- Build UI for detail and settings screens
- Build alternate layouts for large screens

Task 3: Build for Data Persistence

Implement the data persistence structure.

- Build ContentProvider and link to SQLite database for application data
- Configure SharedPreferences for app settings data.

Task 4: Implement Google Play Services

- Configure structure for handling calls to Location Services
- Configure structure for handling calls to the Maps Api

Task 5: Integrate Notifications

Build a notification system to help users to be updated to pertinent location based information.

Task 6: Build App Widget

Build an app widget to rest on the home screen and show the user pertinent information.

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**"