Capstone_Stage1

**How to Use this Template**

1. Create a new document, and copy and paste the text from this template into your new document [ Select All → Copy → Paste into new document ]
2. Name your document file: “**Capstone_Stage1**”
3. Replace the text in green

---

**GitHub Username**: Annin7y

# London Tube Schedule App

## Description

This app provides schedules for all the Tube stations within London, UK. Users will first select a Tube line to view a list of stations on the first screen. Data will be pulled from https://api.tfl.gov.uk/. Clicking on any of the lines will take users to the second screen where a list of stations in the selected line will be displayed. Then, clicking on any of the stations will take users to the third screen where train arrival and departure schedules will be displayed on the third screen.

The idea is to make a basic version for the Capstone Project, but I plan to add other features, such as Google Maps, in the near future.

## Intended User

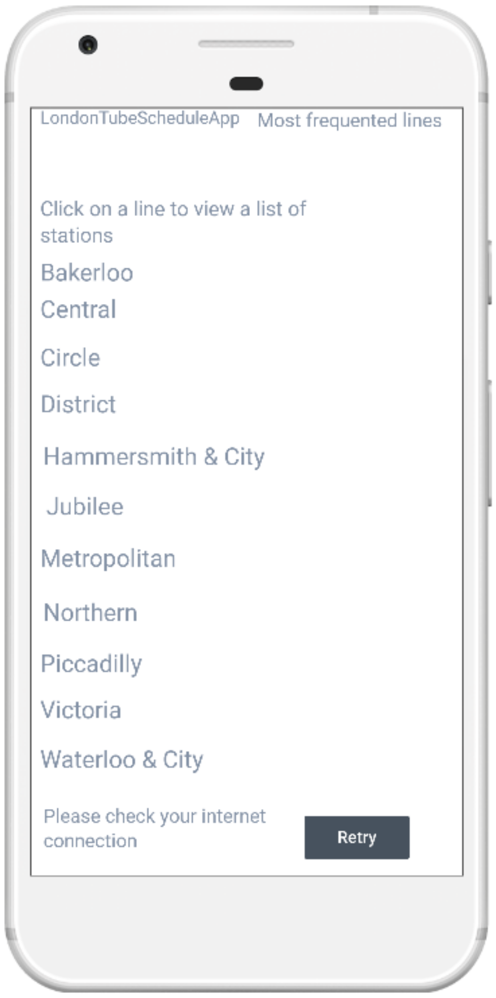
This app is designed for anyone who wishes to obtain London Tube train schedules. Target audience includes daily riders, occasional riders, and tourists.

## Features

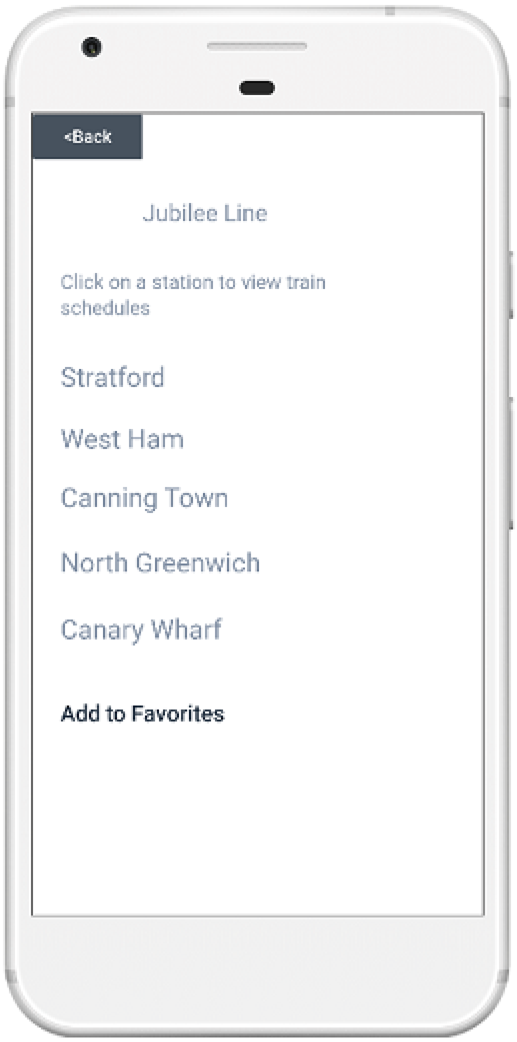
- Stores data – a list of most frequented stations - for offline viewing.
- Features Share Intent in the menu option. Allows users to share the URL of the selected station and arrival times with other apps.
- Features a Widget which displays arrival times

## User Interface Mocks

### Screen 1(Main Activity)



## Screen 2(StationListActivity)



Jubilee Line

Click on a station to view train schedules

Stratford

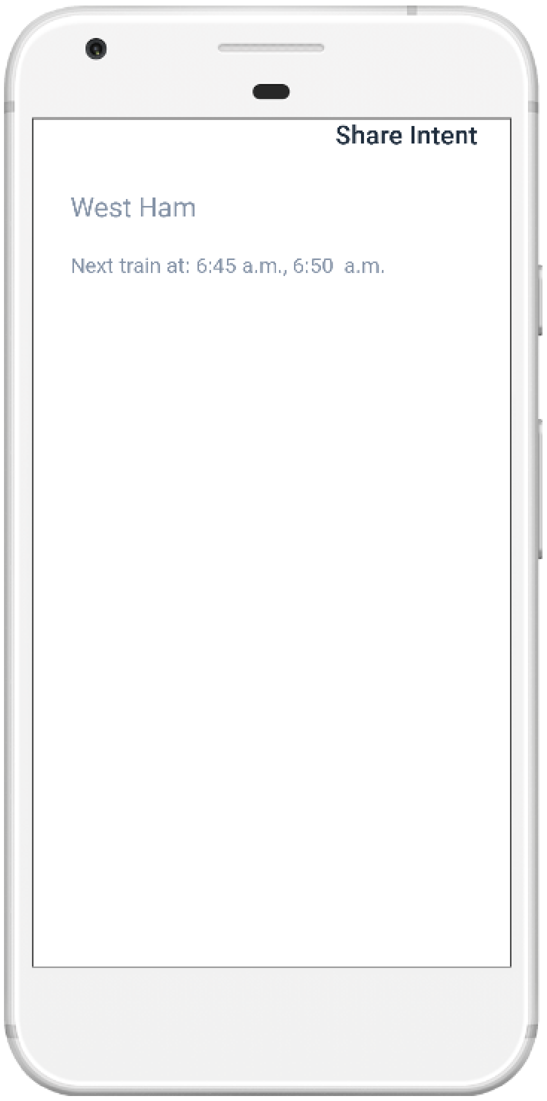
West Ham

Canning Town

North Greenwich

Canary Wharf

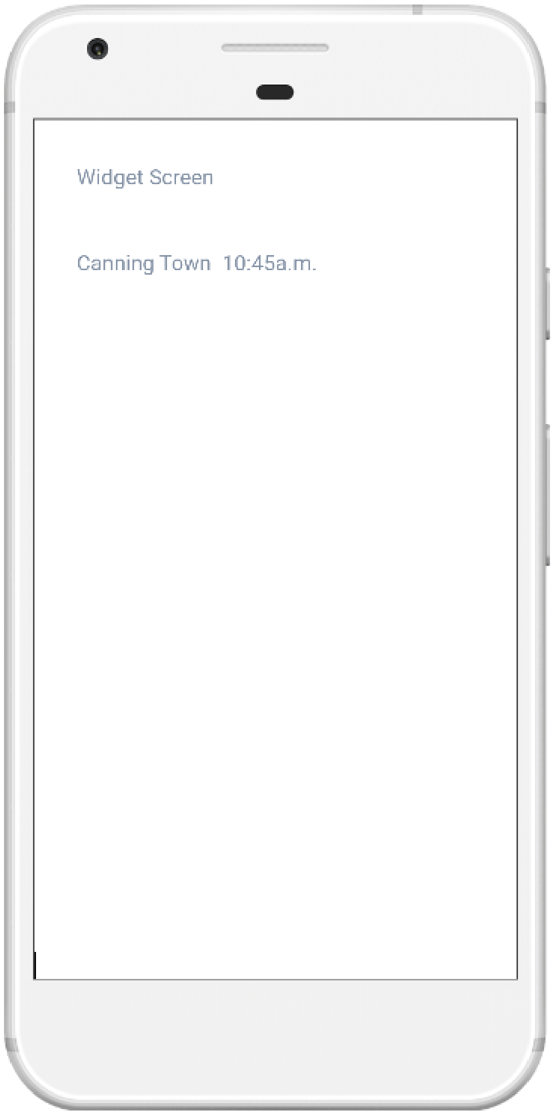
## Screen 3(StationScheduleActivity)



West Ham

Next train at: 6:45 a.m., 6:50  a.m.

## Screen 4(Widget)


Widget Screen

Canning Town  10:45a.m.

## Key Considerations

### How will your app handle data persistence?

Users will be able to store selected data – most frequented lines - by clicking on the “Add to Favorites” Button. Data will be stored in a Content Provider and will be available offline. Detailed explanation in Task 3 below.

### Describe any edge or corner cases in the UX.

Snackbar will be implemented in the MainActivity if there’s no internet connection. Users will have the option to click on the Retry button to try to connect again.

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

Butterknife will be used for binding views. Using the Butterknife library instead of using findViewById will reduce the amount of boilerplate code.

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

This app will implement Google Firebase Analytics (GFA) and a Banner Ad. The implementation of GFA is discusses in Task 5 and that of the Banner Ad in Task 6 below.

## 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

The first task will be to examine the endpoints on https://api.tfl.gov.uk/. Since the data will be downloaded from the Unified API, we first need to obtain an API key and App Id. Both will be stored (hidden) in gradle.properties. Unified API allows developers to obtain data that is easily understandable and designed to use in real time. Detailed explanation is available here: https://tfl.gov.uk/info-for/open-data-users/unified-api?intcmp=29422#on-this-page-0

Data is available in JSON and XML. We will be downloading the data in the JSON format. Parsing will be implemented in the JSONUtils class. The site provides an option for developers to test URLs directly on the site. URLs will be built in the NetworkUtils class and the data will be downloaded via Asynctask.

The code will be written entirely in Java. App will use stable release versions of all libraries, Gradle, and Android Studio. It will include support for accessibility: content descriptions,navigation using a D-pad, and if applicable, non-audio versions of audio cues. All strings will be stored in a strings.xml file and RTL layout will be enabled.

## Task 2: Implement UI for Each Activity and Fragment

- Build UI for the MainActivity
  The following URL will get a list of all the tube lines.
  https://api.tfl.gov.uk/swagger/ui/index.html?url=/swagger/docs/v1#!/Line/Line_GetByMode.

  The list will be loaded into the MainActivity via AsyncTaskInterface. LineId will be passed via Parcelable to the StationListActivity. Most frequented lines will be accessed from the Menu. Snackbar(see above) will be displayed if no internet connection.

- Build UI for the StationListActivity
  The following URL will get a list of all the stations in the selected line.

  https://api.tfl.gov.uk/swagger/ui/index.html?url=/swagger/docs/v1#!/Line/Line_RouteSequence

  This URL returns both the LineId and StopPointId(station id). The latter will be sent via Parcelable to the StationScheduleActivity.

- Build UI for the StationScheduleActivity
  The following URL will get the list of arrival predictions for given line ids based at the given stop.
  https://api.tfl.gov.uk/swagger/ui/index.html?url=/swagger/docs/v1#!/Line/Line_Arrivals

## Task 3: Implement a Content Provider

- Create the following classes: ScheduleContentProvider, ScheduleDbHelper, ScheduleContract
- Implement “Add to Favorites” button in the StationListActivity;
- Display a Toast message after a station has been added: “Station successfully added to Favorites”
- Favorite Stations can be accessed from the Menu option in the MainActivity.
- Favorites data will be loaded into the MainActivity via AsyncTask Loader.

## Task 4: Implement a Widget

The widget will display a list of arrival times for the last selected station.

- Store arrival times in SharedPreferences
- Create a widget layout.
- Create a WidgetProvider Class
- Create a RemoteViewsService Class
- Send the data to the Widget.
- Retrieve the data in the WidgetProvider Class

## Task 5: Implement Google Analytics For Firebase

- Create a log event in the MainActivity when the user selects a line
- Create a log event in the StationListActivity when the user selects a station

## Task 6: Implement a Banner Ad

The Banner Ad will be displayed in the MainActivity.

- Add AdView to activity_main.xml.
- Load the Banner Ad into the MainActivity.