Capstone_Stage1

GitHub Username: amy6

Mitra

Description

Mitra is a city bus tracking app for Mysuru city. It allows a user to check for the available buses for a given source and destination, the routes covered by the buses, the expected time of arrival, price of the tickets and provides with the ability of live tracking the bus. Users can plan their commute accordingly.

Intended User

The app is intended for the general public who want to use the local city buses in Mysuru city, Karnataka, India.

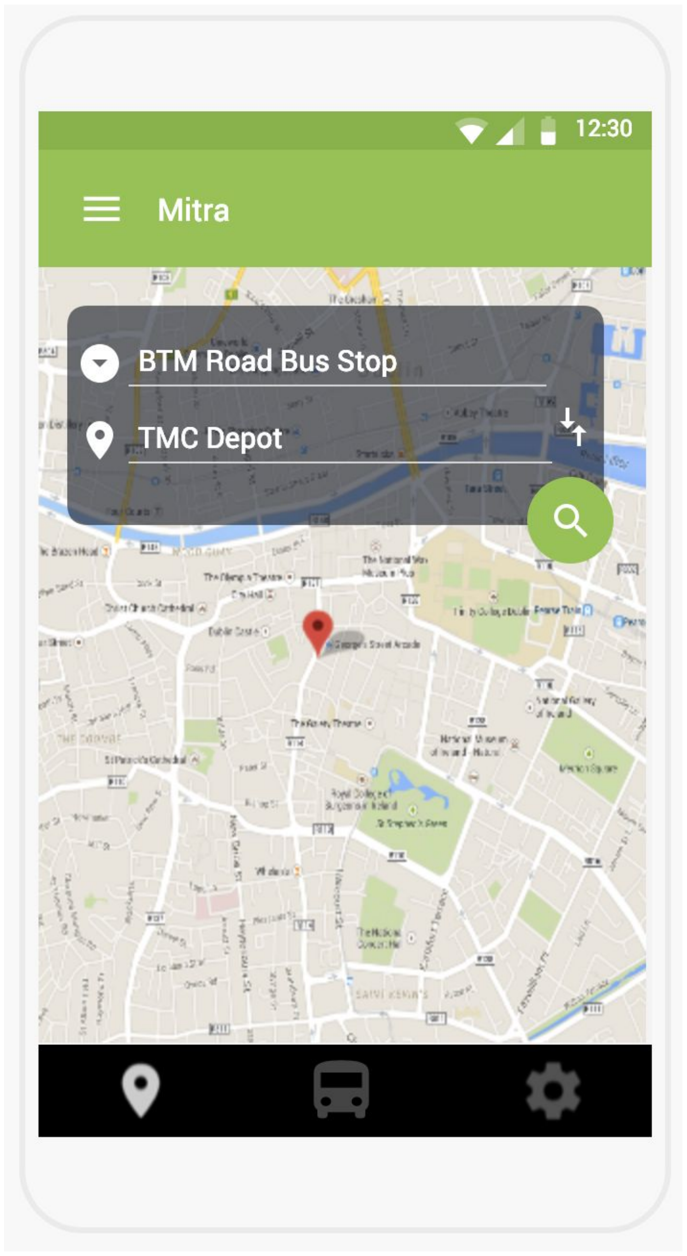
Features

- List bus numbers for a given source and destination
- List all the stops for a given bus route
- Expected time of arrival of a bus
- Check ticket fare for a given journey
- Live tracking of buses

Capstone_Stage1

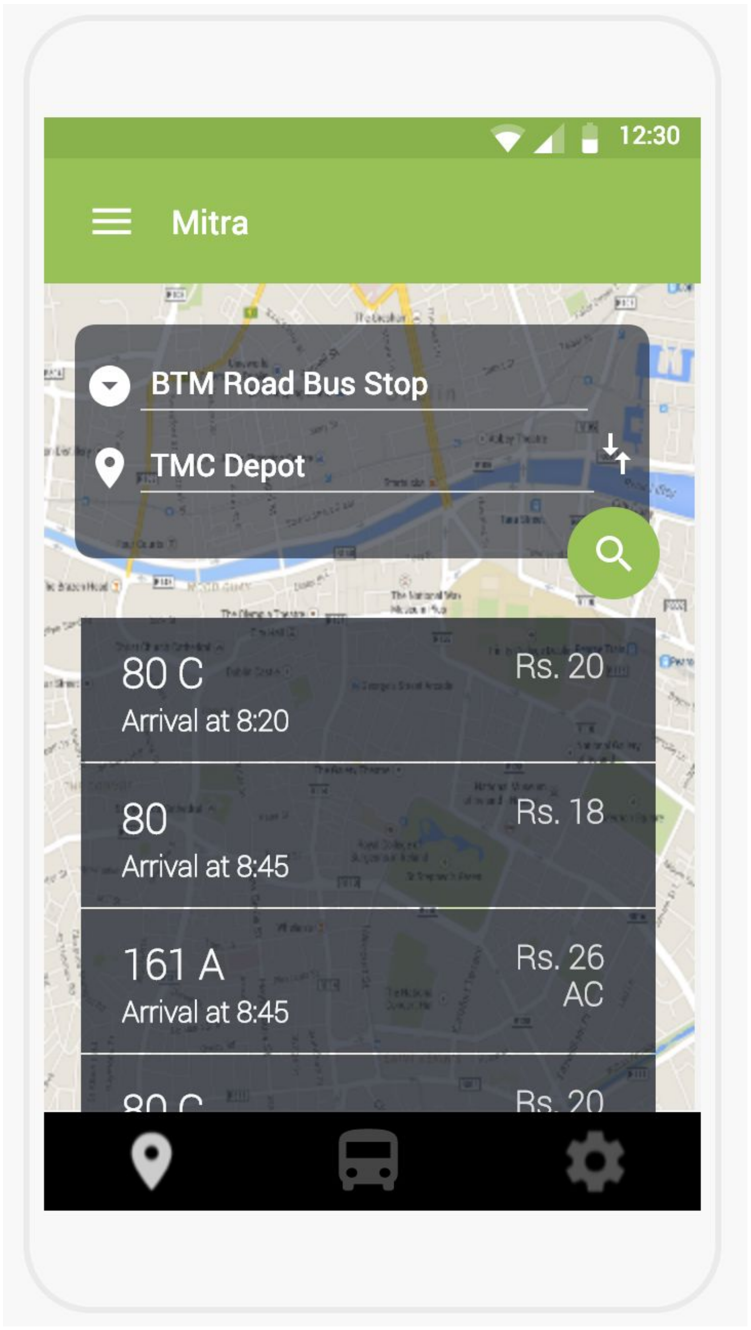
User Interface Mocks

Screen 1



Initial screen where a user can search for buses for the given source and destination

Screen 2



80 C
Arrival at 8:20
Rs. 20

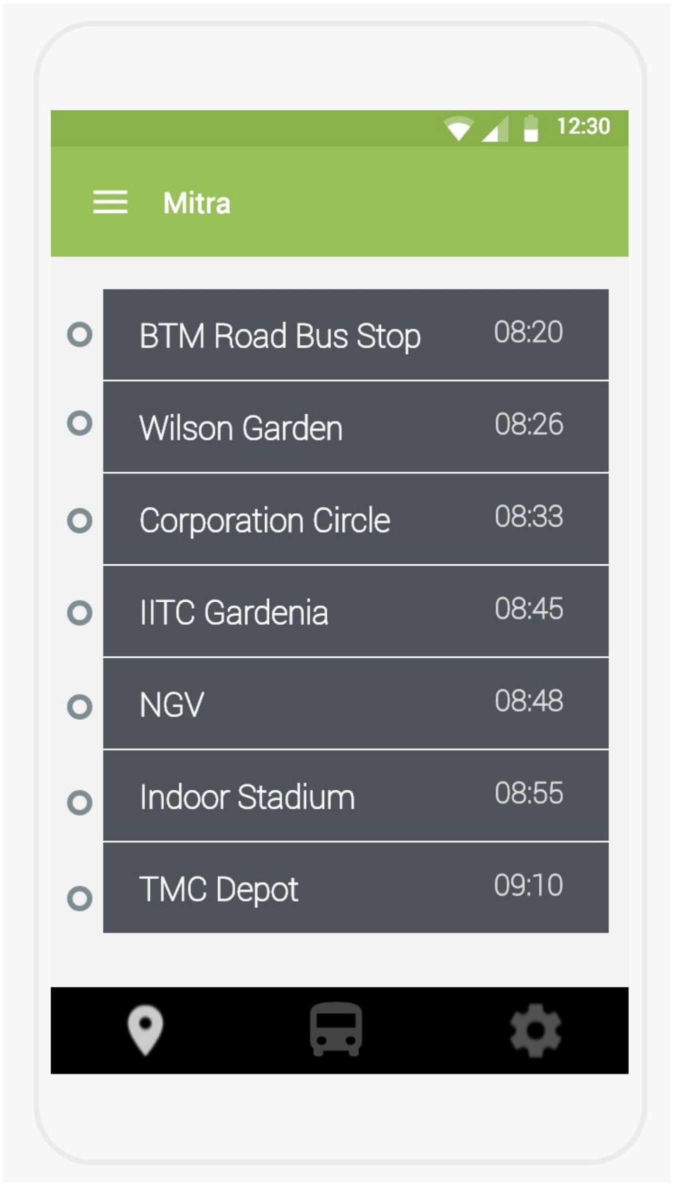
80
Arrival at 8:45
Rs. 18

161 A
Arrival at 8:45
Rs. 26
AC

List of buses for the given source and destination. The list also displays the expected arrival times for the buses and the ticket fare for the journey.

Capstone_Stage1

Screen 3



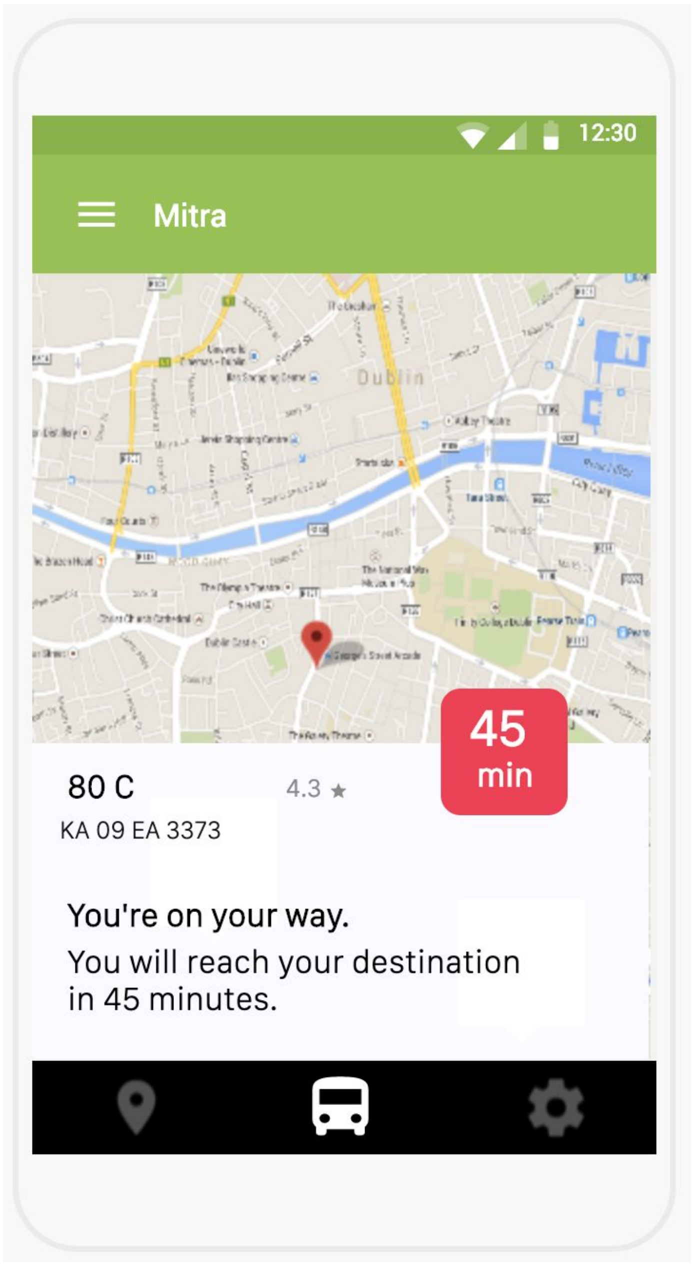
12:30

Mitra

BTM Road Bus Stop	08:20
Wilson Garden	08:26
Corporation Circle	08:33
IITC Gardenia	08:45
NGV	08:48
Indoor Stadium	08:55
TMC Depot	09:10

Route for a given bus along with expected stop times.

Screen 4



80 C 4.3 ★

KA 09 EA 3373

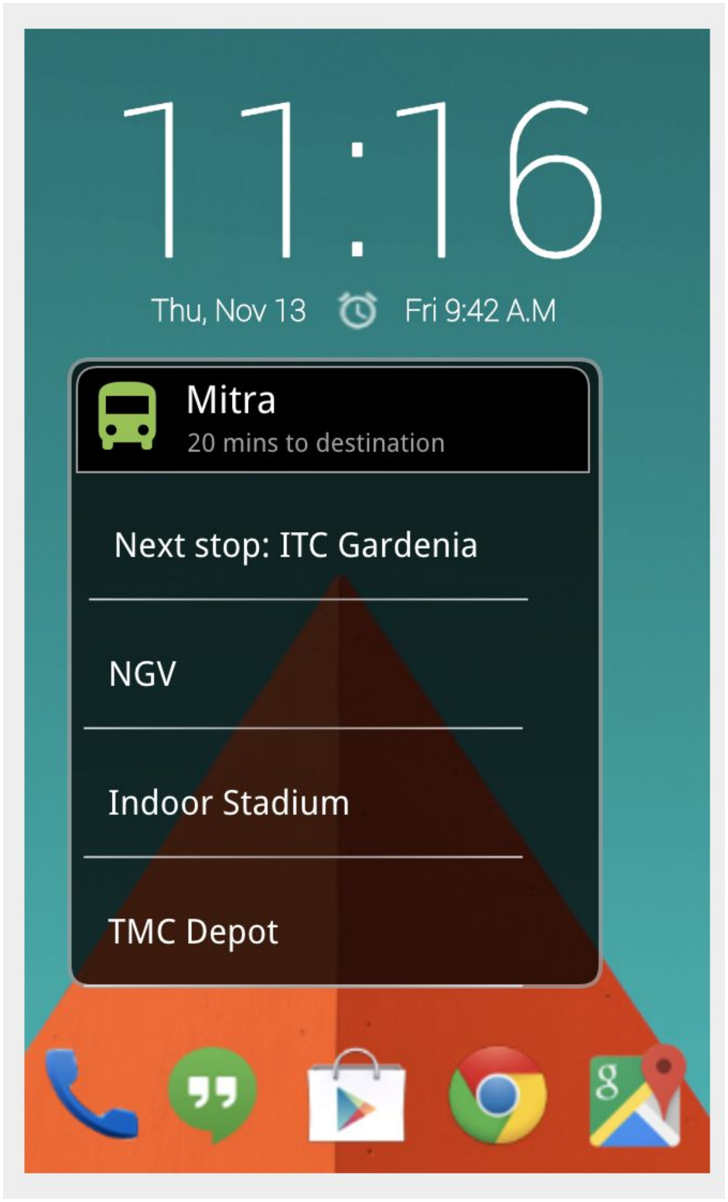
You're on your way.

You will reach your destination in 45 minutes.

Live tracking of bus. Provides details on the expected time of reach.

Capstone_Stage1

Widget



Mitra
20 mins to destination

Next stop: ITC Gardenia

NGV

Indoor Stadium

TMC Depot

Widget displays details upcoming stops during the commute and provides an estimated time to reach the destination

Navigation through the screens:
https://www.fluidui.com/editor/live/preview/cF8yTkVhRUw1OEpGR3k0ZEFSUEVVc2szWGttTlRlNlFhMw==

Key Considerations

How will your app handle data persistence?

Bus details and live tracking information will be fetched via network calls from the server. Fetched details will be stored locally in the SQLite database to support offline services. Locally stored data will be refreshed based on updates and/or user requirements.

Describe any edge or corner cases in the UX.

- Request for location access will be presented to the user as per the applicable UX guidelines.
- Care will be taken to ensure the app does not consume too much battery when the app is running with location access enabled.
- User will be able to access the app offline/low internet connectivity. He/She will be notified in cases where the app does not display real-time data but rather latest available data.

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

- Butterknife to reduce boilerplate code while referencing views.
- Room library for database implementation.
- Retrofit to handle network calls.
- GSON to simplify converting to and from JSON data structure.
- Timber for logging.
- WorkManager for handling background tasks.
- ViewModel and LiveData for building MVVM architecture in conjunction with Room.

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

Google Location services will be used to fetch user location and display live tracking information.
Google Nearby API will be used to get the information of nearby bus stops.

Next Steps: Required Tasks

Task 1: Project Setup

- List required dependencies
- Configure libraries in the project

Task 2: Implement UI for Each Activity and Fragment

- Build UI for Maps Activity
- Design UI for Search fragment, Bus Fragment, Route Fragment
- Build UI for Live Tracking Activity

Task 3: Implement networking

- Define necessary model classes for network data source
- Define Retrofit interface for making API calls

Task 4: Implement database

- Define necessary model classes for DB data source
- Define classes and interfaces required for Room database
- Handle offline caching

Task 5: Integrate data sources

- Define ViewModel and LiveData sources integrating network and DB data sources

Task 6: Data handling

- Define data validation
- Implement error handling
- Implement retry logic as required

Task 7: Multiple screen size support and accessibility

- Design UI to support multiple screen sizes
- Include accessibility support

Task 8: Define test cases

- Define unit tests and integration tests

Task 9: Create build variants

- Create required build variants
- Create app flavors as applicable
- Create keystore for release apk
- Build and deploy application to Play Store