GEORGE MASON UNIVERSITY

ECE 590 Mobile Systems and Applications

# Project Title: TripMate

## Team: BUGDROID

**By:**
**Nithin Kumar Kotte (G01035945)**
**Saurabh Deshpande (G01036463)**
**Devang Motwani (G01040288)**
**Mihir Garude (G00969831)**

# Design Introduction:

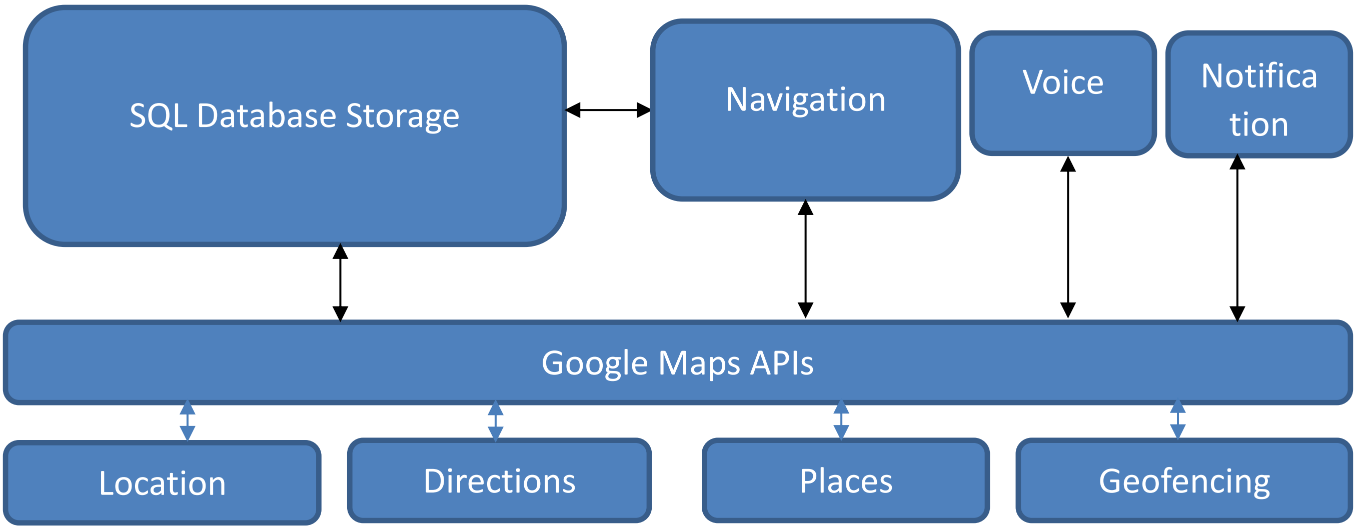
## Application scenario:

TripMate is an application for everyday use. The main purpose of the application is to help the user create a list of the number of places that the user wants to visit and when the user will Start the route, optimized route will be displayed to the user and as the user starts the Navigation, the application will use Google Maps for real time Navigation. Also, with multiple locations in a list, the application will monitor the user's location and provide a notification when the user is in the proximity of any location.

## Solution description:

- To achieve this task in the application, we have used SQLite database to save the lists dynamically to support N waypoints in a list.
- We have also implemented Google Places API to offer autocomplete suggestions while the user enters the address of the location.
- We have used Google Maps Location Services API to obtain the current location of the user. Also, it has been used for the purpose of obtaining GPS locations from addresses entered by user and then an array-list of the path is prepared from the response to draw a polyline on the Map. Moreover, Fetched GPS locations are displayed on the Map using various markers and other details are fetched from JSON file obtained as response from the API.
- Obtaining the GPS location, a service has been implemented that runs using Geofence API of Google to notify the user when the user is in the proximity of the location.
- The GPS locations have seen saved using Shared Preferences to keep them intact even when the application has been killed.
- Notification Manager and Text-to-Speech Class has been used to implement voice notification as a response to GPS location.
- For the purpose of Navigation, we are passing intent to the Google Maps along with the location addresses that are in the list that the user is using.

## Design overview:



# Design Details:

## Itemized Tasks, Task Description and Solution Details:

- User-Interface Module Development
  - Task Description: To design a user interface for the proposed application to provide effective navigation which depicts how to plan out the high-level screen hierarchy for application.
  - Solution details:
    - Firstly, to provide professional look for the application, splash screen was implemented which was displayed for the boot time of the application.
    - The first screen of the application after splash was the tabbed activity comprising of the Lists activity and the details activity in xml. The lists activity displays the stored lists names from the database and updates it every time a new list is entered.
    - The Addlist activity provided user to enter the list name, destination name and adding the halts for this list. Here user can add any number of halts in his list. To provide this facility, in the row.xml file dynamically Edittexts were generated and also each halt can be removed not affecting other halts in the application. Scroll views were used to accommodate all the halts and lists in the activity.
  - Also, starting any list for navigation or removing them options are provided on the lists activity of the application. To view the details of any lists, show details provides all the halts included for that list on the details activity.
  - Further on starting any list, first the overview of the route was generated which displays current location, halts and destination locations with different markers. And after that option for starting the navigation was provided. All other modules were integrated with provided blueprint of the application.

- Location Module Development
  - Task Description: This module deals with obtaining data about user's current location from Google Maps using GPS.
  - Solution details:

    To obtain data from Google Maps API, following needs to be done:
    - Permission for Connectivity and Fine Location Access need to be provided in AndroidManifest.
    - For Android 6 and above API, permission grant access check is required and if it is not given the App will throw an error. Also, if the user is not granted permission, permission access is asked from user when the App is used for the first time.
    - To access GPS location, we need to access Location Services API and to access that we need to access Google Play Services. It monitors user's updated location and saves it. So, in the Gradle, we need to add Google Play Services in the dependencies.

- Routing Optimization and Directions API
  - Task Description: This module requires us to get N halts from the list and reorder them to optimize the route from Source to Destination. Moreover, other details of time and

distance are required to be displayed on Maps and also all the halts are marked on the Map to display them using Markers.

- Solution details:
  - N number of location addresses are obtained from the list and these locations are parsed in a URL format to be sent to Google Maps. The URL requires us to send some parameters including- 1) Required parameters such as response type, origin, destination and key obtained from Google Developer Console 2) Optional parameters include mode, waypoints, distance, time, etc.
  - The response type chosen was JSON and data extraction from this file was done by preparing a JSON parser and using it.
  - The GPS locations of all the halts is extracted and is used to set Markers on the Map and the sets of GPS locations is used to extract data and display a polyline connecting these halts.

- Navigation
  - Task Description: This module required us to help user navigate to different halts after the optimized root was obtained and show user real-time Navigation.
  - Solution details:
    - To solve this, we are passing intent to Google Maps Application.
    - The reason for using Google Maps App is that Google does not provide API for using Navigation and hence we are using Navigation by passing intent to Google Maps along with all the locations that need to be routed in optimized manner.
    - Moreover, when we pass intent to Google Maps, the API states that we can only send source and destination to the Maps and this was one of the issue faced.
    - To solve this, we tried to understand how Google Maps URL works for multiple location routing and parsed the addresses in similar manner while passing the intent and we were able to send multiple locations and were ready to start Navigation.

- Connectivity module Development:
  - Task Description: This module was developed to check with the user permissions for different APIs that is to get the user device for connectivity and location permissions and many more.
  - Solution details:
    - AndroidManifest file is included with permissions for Internet, Access Location and for fine location we need to place permission for Access Fine location.
    - We also need to get permissions for Read Services to be implemented for Places API.
    - For the maps to be implemented, we need to put the key obtained from Google Developer Console in the xml file to access maps. Moreover, we also need to place in Meta-data to define the dependencies.

- Geofencing API
  - Task Description: This module requires us to notify the user whenever the user is in the proximity of any of the locations entered by the user in the list.
  - Solution details:

    - To achieve this, we need to use the Geofencing API of Google.
    - To use that we need to use the Latitude and Longitude of all the halts and we need to save these as even if the application is not running the user needs to be notified and for that we need to run the Geofence in service.
    - So, to save this data Shared preference are being used and accessed for the Geofence to run in background.
    - For the Geofencing API to work, we need to pass a Geofence object to the API that includes the latitude, longitude and radius.
    - The main problem faced here was the fact that we had to use static instances as to access the same Shared Preferences every time and this issue took a long time to solve.

- Database Storage and Management
  - Task Description: This module is responsible in maintaining a storage repository for performing creation, deletion, updating and replacing (CRUD) operation on lists. Lists here consists of all the destinations and intermediate halts that the user want to traverse through. To provide journey waypoints and destination details to Maps activity.
  - Solution details:
    - Implementation of dbHelper class to create tables under set database. Implementing createlist() method to dynamically create multiple tables(lists) under sqlite_master which contains root page number for every other table and index in the database file.
    - Creating list/tables using auto generating ID as primary key and columns containing place specific ID's and place description.
    - Performing CRUD operations such as adding multiple halts, retrieving and deletion of lists/tables and details of the list were done using SQL raw queries through cursors.
    - Following issues were faced:
      Creation of multiple tables under one database was solved using separate method for table creation instead of onCreate().Traversing data values between activities and databases. Dynamically adding halts in the database based on required list/table.

- Google Places API
  - Task Description: This API was used to develop user-defined ArrayAdapters for autofill operations for filling destination and Halts details by user on UI.
  - Solution details:
    - Using GEO_DATA_API and GOOGLE_API_CLIENT_ID of location services to fetch and predict place details. Implementing class methods to filter the predicted results. Creating an inner class to store place name and description.
    - Following issues were faced: Limiting the number fetched details and it was solved by implementing threshold functionality. Adapting the filtered details based on the UI layout requirements.

- Text-to-Speech (Voice) Module
  - Task Description: To provide a voice notification to user when he is within closed proximity of a location. Additionally, provide another message to the user when the user wants to find the route from source to destination.

  - Solution details:
    - For Voice Notification, the TextToSpeech class was utilized. Whenever, the user reached within close proximity of the desired location, an intent would be passed from Geofencing module to Voice Notification module triggering a message "You have reached." Additionally, whenever the user click the Find Path on Google Maps Activity module another "Finding Path" message will be displayed.

- Notification Module
  - Task Description: To pop up notification whenever user in the proximity of the halt and play the voice reminder. To specify actions on the notifications.
  - Solution details:
    - Notification Manager is used to display a persistent notification presented to the user. When you tell the system to issue a notification, it first appears as an icon in the notification area. To see the details of the notification, the user opens the notification drawer.
    - Both the notification area and the notification drawer are system-controlled areas that the user can view at any time. Action to start the navigation is provided in the notification area and play the voice notification is provided.

# Execution Situation:

## Task finished by each team member:

### Saurabh:

I was dealing primarily with the User Interface module development, notification module, connectivity and database UI integration, where I decided the flow of application to improve the layout performance. This included updating user interface with changes in the database. I was also responsible for the connectivity module where I was required to take of device permissions for different API levels. Framework for integration of different modules was provided. I did implement the notification module where different actions on voice reminder were provided in the notification drawer.

### Mihir:

I was primarily responsible for developing Directions module and routing module. This involved writing a JSON parser to parse data obtained from Google Maps and plot the path from source to destination. Additionally, a decoding function was developed to decode the encrypted path obtained from Google Maps.

Secondly, I was responsible for developing the voice module. This was achieved using an in built TextToSpeech class, providing voice message which are triggered based on Geofence events and user button events.

### Nithin:

I was responsible for designing and implementing of SQLite Database which primarily consists of structuring of database based on application requirements, designing user specific databases for storing multiple tables, Rendering UI based on database transaction (Shared with Saurabh). Setting data format for ease in UI interaction and setting location addresses for map integration.

Implementation of user defined methods involving Google places API for predicting locations, filtering them and populating the results in autocomplete textview. Also, I did implement modules related to Geofencing functionality to solve static java dependencies.

### Devang:

I was primarily responsible for developing Location module, Navigation and Geofence module code. Location module was used to obtain current location from Google Play Services and later I was involved in integrating this module with Routing and Directions. In the Routing and Directions development, I was involved in modifying JSON parser to plot multiple way-point and find multiple paths between source to destination. Navigation Module does the work of obtaining the optimized route of multiple way points and passes over to Google Maps to navigate. Geofencing module works as a service that calls the notification module to notify when the user is in the proximity of the location.

## Project time line:

|  | 09-Mar | 15-Mar | 22-Mar | 29-Mar | 05-Apr |  | 12-Apr | 19-Apr | 26-Apr | 03-May | 10-May |
|---|---|---|---|---|---|---|---|---|---|---|---|
| UI module development | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | 🟦 | Achieved |
| Location module development | 🟨 | 🟨 | 🟨 | 🟨 | Achieved |  |  |  |  |  |  |
| Routing optimization and Directions |  | 🟥 | 🟥 | 🟥 | 🟥 | 🟥<br>🟨 | 🟥<br>🟨 | 🟥<br>🟨 | Achieved |  |  |
| Navigation |  |  |  |  | 🟨 | 🟨 | 🟨 | 🟨 | Achieved |  |  |
| Connectivity module development | 🟦 | 🟦 | Achieved |  |  |  |  |  |  |  |  |
| Geo Fencing |  | 🟨 | 🟨 | 🟨 | 🟨 | 🟨 | 🟨 | 🟨 | 🟪 | 🟪 | Achieved |
| Database Storage and Management |  |  | 🟦 | 🟪<br>🟦 | 🟪<br>🟦 | 🟪<br>🟦 | 🟪<br>🟦 | 🟪<br>🟦 | 🟪<br>🟦 | Achieved |  |
| Google Places API |  |  | 🟪 | 🟪 | 🟪 | 🟪 | Acheived |  |  |  |  |
| Voice |  |  |  |  |  | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | Achieved |
| Notification | 🟦 | 🟦 | 🟦 |  |  |  |  |  |  |  |  |
| UI and Database Integration |  |  |  |  |  | 🟦<br>🟪 | 🟦<br>🟪 | 🟦<br>🟪 | 🟦<br>🟪 | 🟦<br>🟪 | Achieved |
| Direction, Navigation, Location and Optimize path Integration |  |  |  |  |  | 🟥<br>🟨 | 🟥<br>🟨 | 🟥<br>🟨 | 🟥<br>🟨 | 🟥<br>🟨 | Achieved |
| Geofence and Voice Notification |  |  |  |  |  |  | 🟥<br>🟨 | 🟥<br>🟨 | 🟥<br>🟨 | 🟥<br>🟨 | Achieved |
| UI and Database Integration + Direction, Navigation, Location and Optimize path Integration |  |  |  |  |  |  |  | 🟪 | 🟪 | 🟪 | Achieved |
| Testing |  |  |  |  |  | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 |
| Final Report and Presentations | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 |

| 🟦 | 🟥 | 🟨 | 🟪 | 🟩 |
|---|---|---|---|---|
| Saurabh | Mihir | Devang | Nithin | Entire team |

***Code:*** https://github.com/d27saurabh/TripmateTry1
***Video:*** https://www.youtube.com/watch?v=BjdJoKicdTA