Université de Bourgogne Franche-Comté

Master 2 – IoT

Mobility and Smart Cities

Project Report

Final Version

Presented by: SHREM Mohammad
ALKHALAF Mariam
IDRISSA MASSALY

Presented to: Prof. CANALDA Philippe

20/12/2024

Table of Contents

Introduction

Main Problems

The increasing significance of mobility issues cannot be denied. Factors like time, expenses, and distance play a pivotal role in shaping people's daily transportation experiences. Consequently, the development of an application that facilitates users in locating the nearest meeting point in the most time-efficient manner assumes great importance. This endeavor not only serves the individual's convenience but also contributes to the smart evolution of cities by optimizing transportation on a broader scale.

Professor Philippe Canalda's project introduces a solution that takes into account multiple locations and the respective positions of its users, with the primary goal of suggesting the most efficient meeting point, thus conserving valuable user time.

Project Idea

Our project introduces a solution aimed at enhancing mobility within smart cities through the development of a mobile application. This application is designed to enable a group of users to efficiently discover an ideal meeting point, leveraging the knowledge and skills acquired during the "Mobility & Smart Cities" course. The app is specifically tailored for Android/Web as requested.

This application takes as its input a collection of locations and the individual locations of each user. It subsequently employs an algorithm to determine the closest meeting point for all members of the group. The algorithm takes into account factors such as route optimization and minimization of travel time in order to achieve this objective.

Architecture

a. Main actors and roles

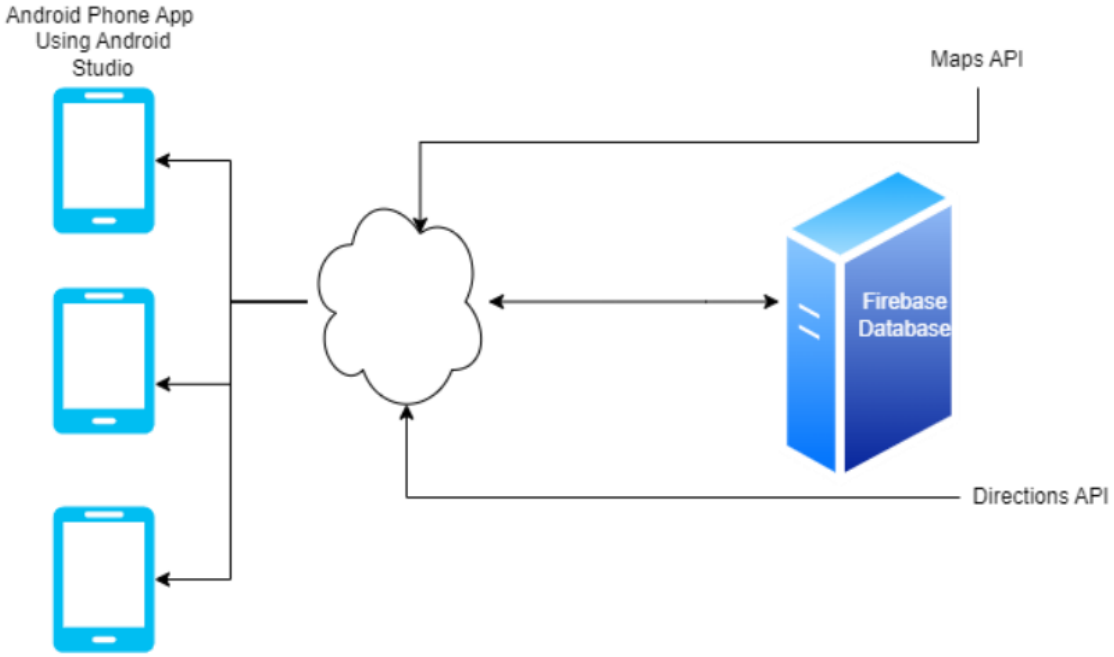
We have two main actors for the application:

- The user:
    - See friend’s location.
    - Find nearest meeting point.
    - See the route to the meeting point.
    - Get directions and instructions to that point.

- The developer:
    - Add location data using maps services.
    - Create the mobile application.
    - Use an algorithm in the app to find optimal meeting point.
    - Add locations to the database

b. Software and Hardware

- Visual Studio is used for the development (React Language)
- Firebase to store the data.
- Directions API.
- Maps API (From MapBox)
- Example of graph

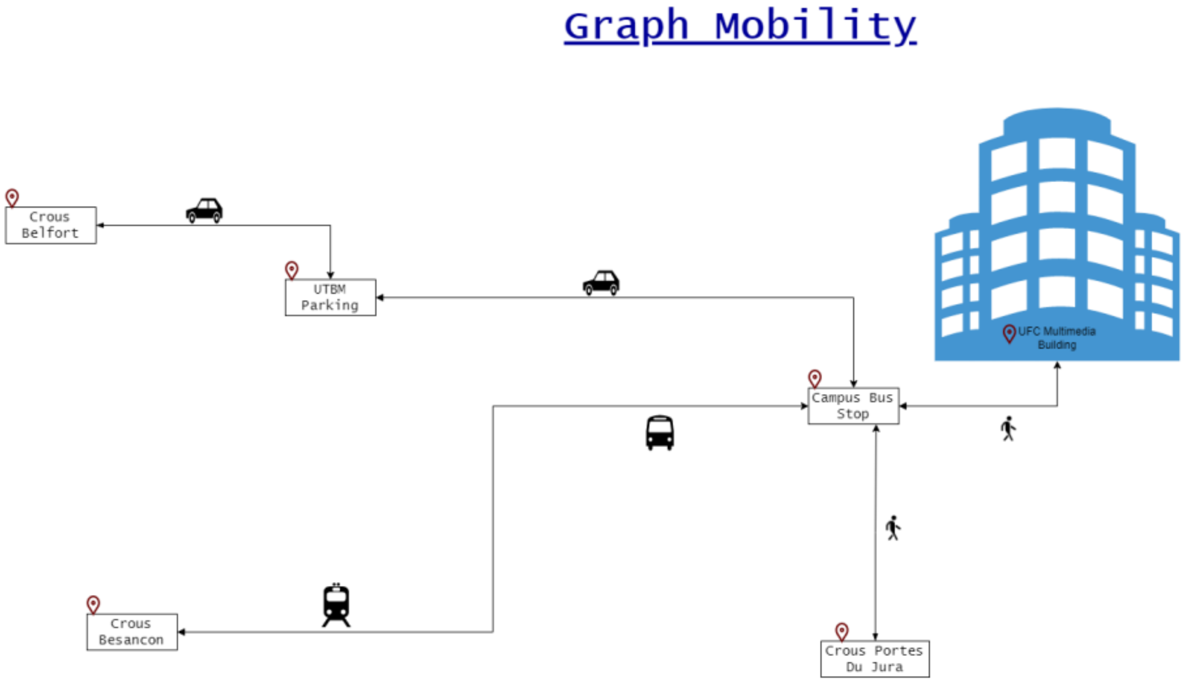


c. Main Functionalities

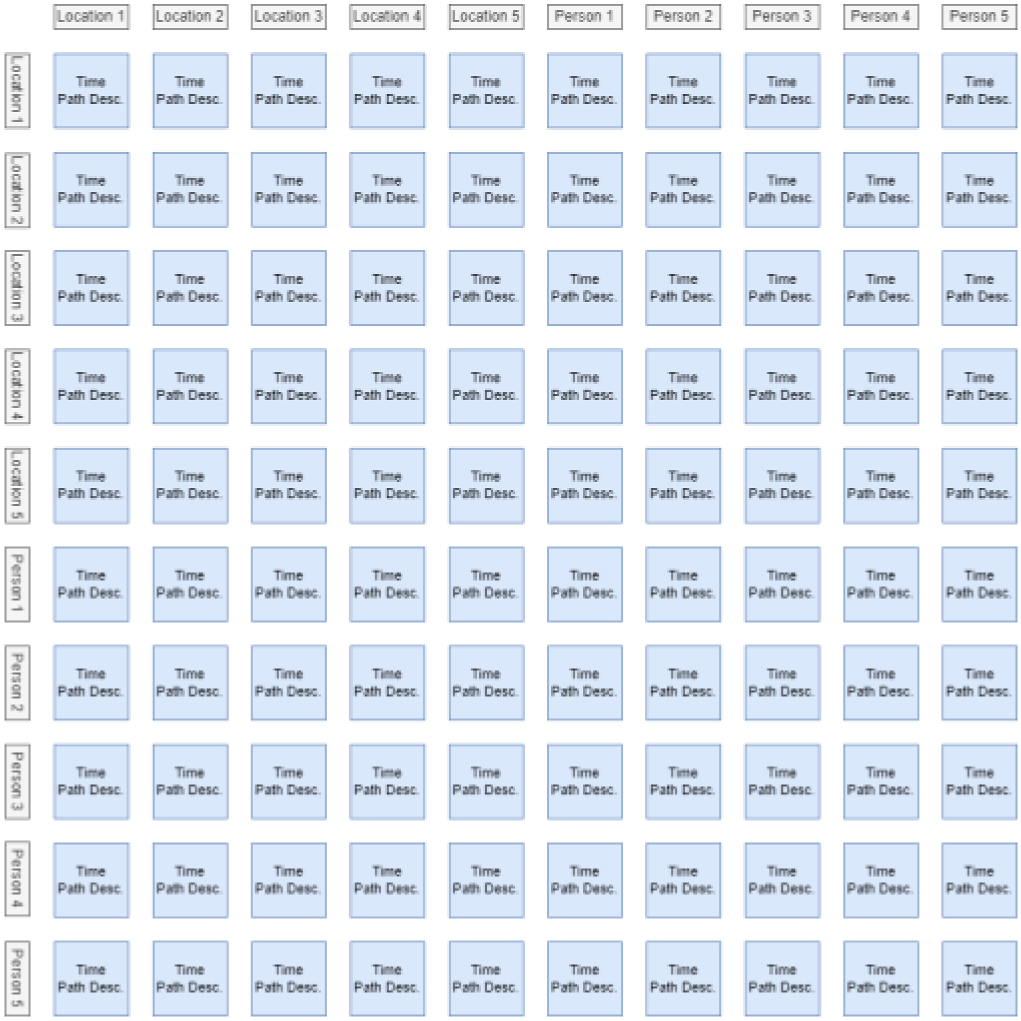
1. Upon launching the application, a marker will indicate the user's current location.
2. Once the app is opened, markers representing the locations of the user's friends will be displayed on the map.
3. Subsequently, the application will determine the optimal meeting point for the group of friends and showcase it on the map once I click “Optimize Button”
4. A designated route will be presented on the map, guiding users to the calculated meeting point.
5. Detailed instructions and directions will then be provided to facilitate reaching the designated meeting point.

d. Steps to find the meeting point

1. A graph mobility is drawn showing all the available locations.



2. From that graph mobility, an origin destination matrix will be built using route description and time to reach every location.



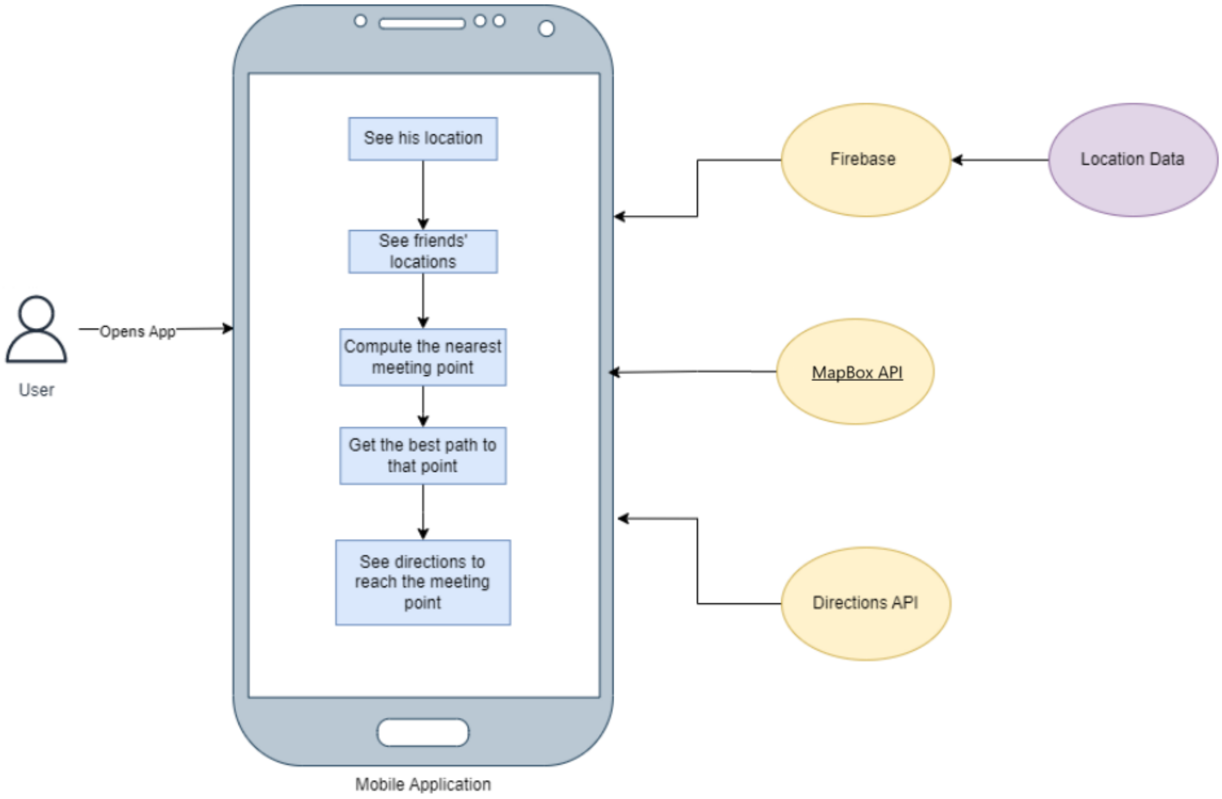
	Location 1	Location 2	Location 3	Location 4	Location 5	Person 1	Person 2	Person 3	Person 4	Person 5
Location 1	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.
Location 2	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.
Location 3	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.
Location 4	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.
Location 5	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.
Person 1	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.
Person 2	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.
Person 3	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.
Person 4	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.
Person 5	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.	Time Path Desc.

3. Transitive closure function is applied to this matrix.
4. The meeting point is then computed while minimizing the most time possible.

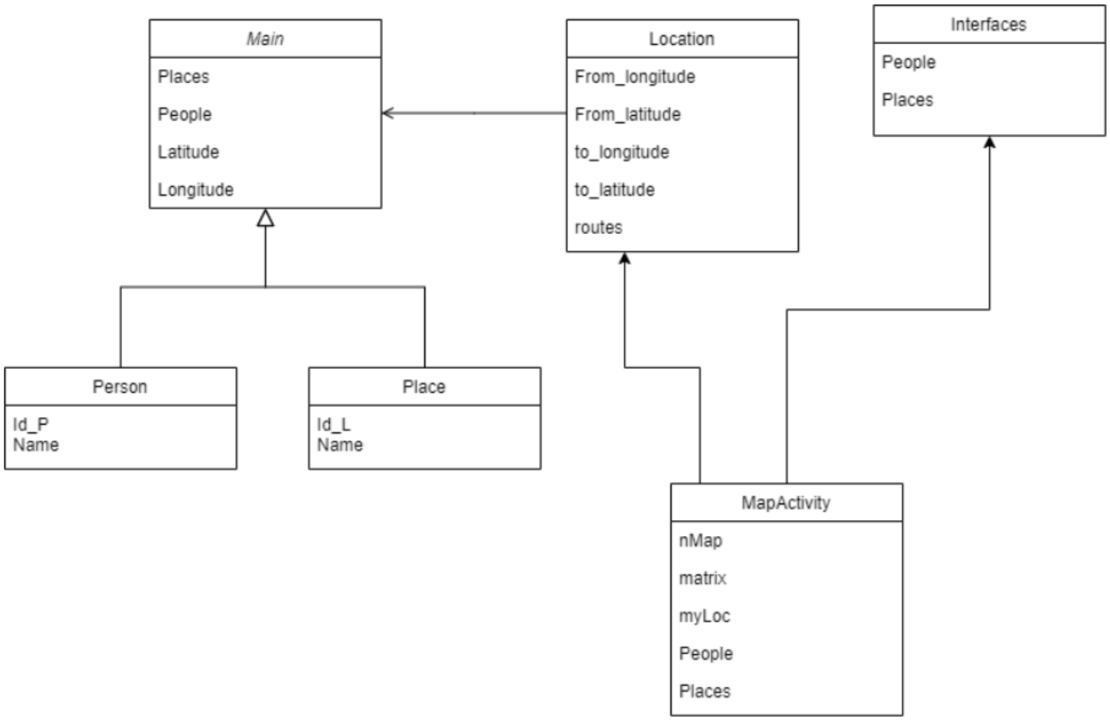
We are using MapBoxAPI to get the directions from the direction API. In order to guide the user from his location to the nearest meeting location for him and his friends.

Use Case

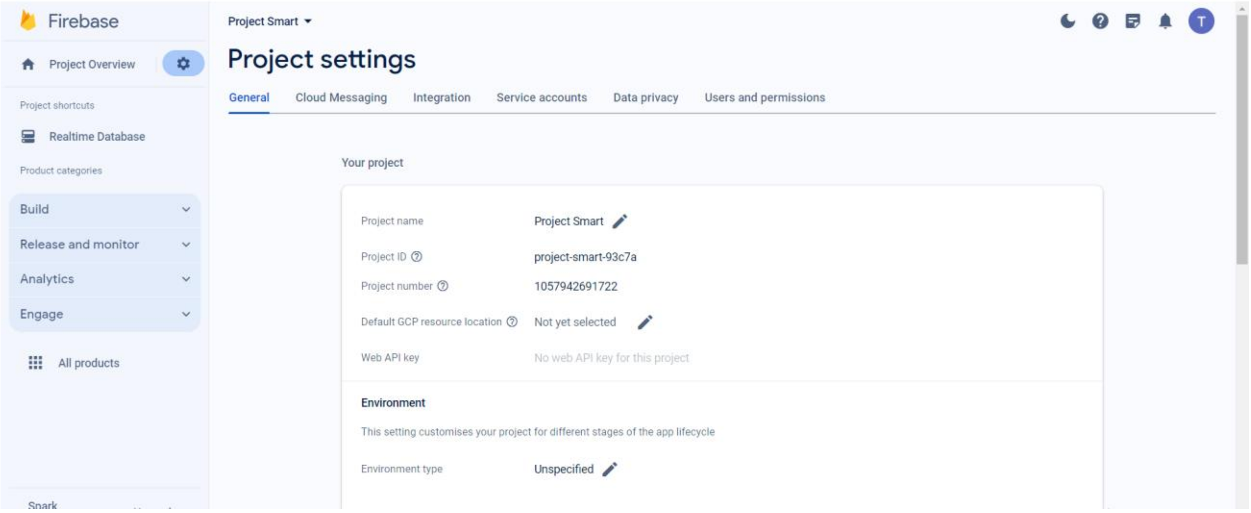
a. Activity Diagram



b. Class Diagram



c.Firebase Database



Project settings

Your project

Project name Project Smart

Project ID project-smart-93c7a

Project number 1057942691722

Default GCP resource location Not yet selected

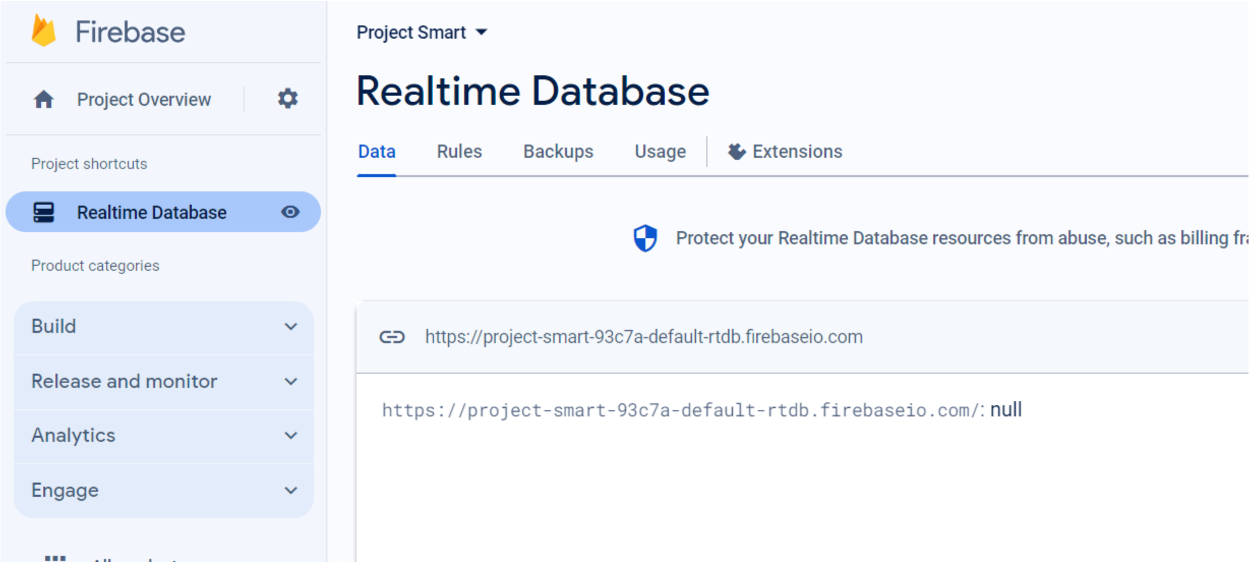
Web API key No web API key for this project

Environment

This setting customises your project for different stages of the app lifecycle

Environment type Unspecified

We created a firebase database in order to store locations for the application.

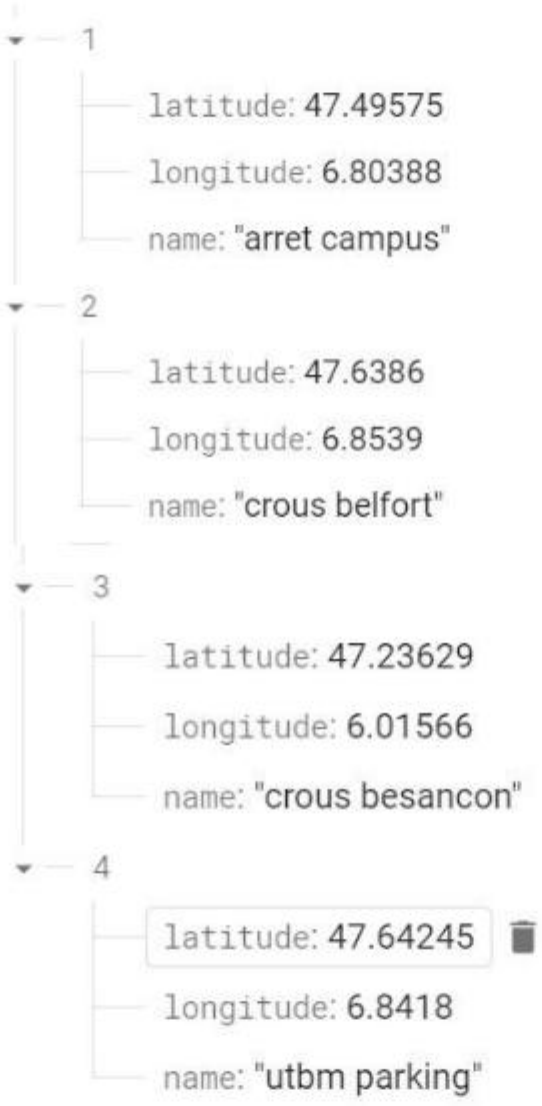


Realtime Database

Protect your Realtime Database resources from abuse, such as billing fr

https://project-smart-93c7a-default-rtdb.firebaseio.com

```
https://project-smart-93c7a-default-rtdb.firebaseio.com/: null
```

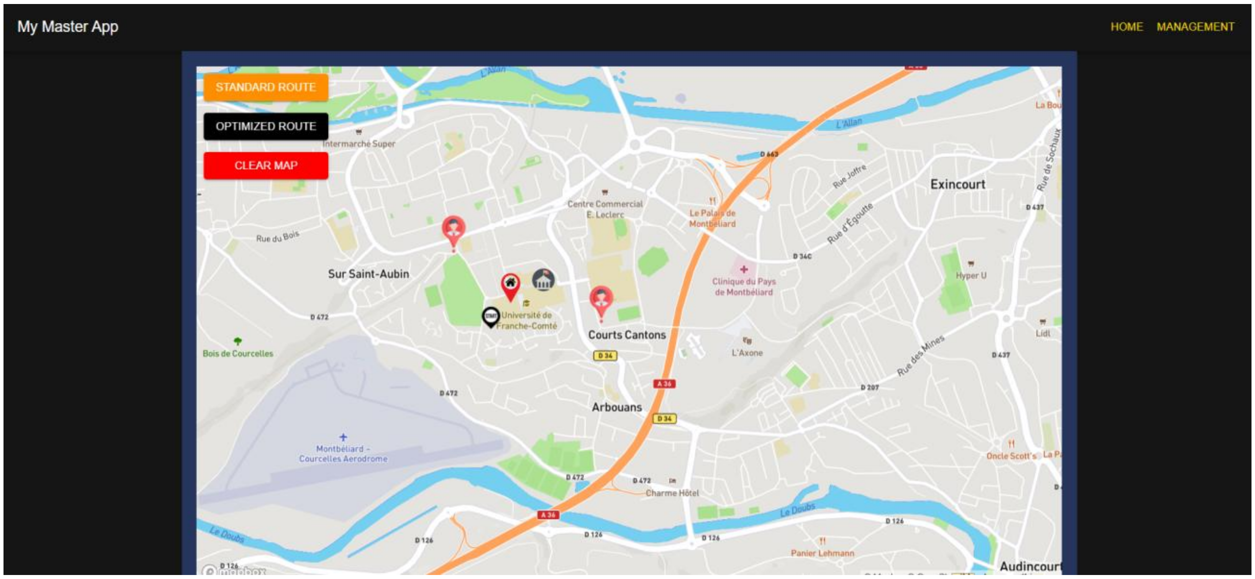


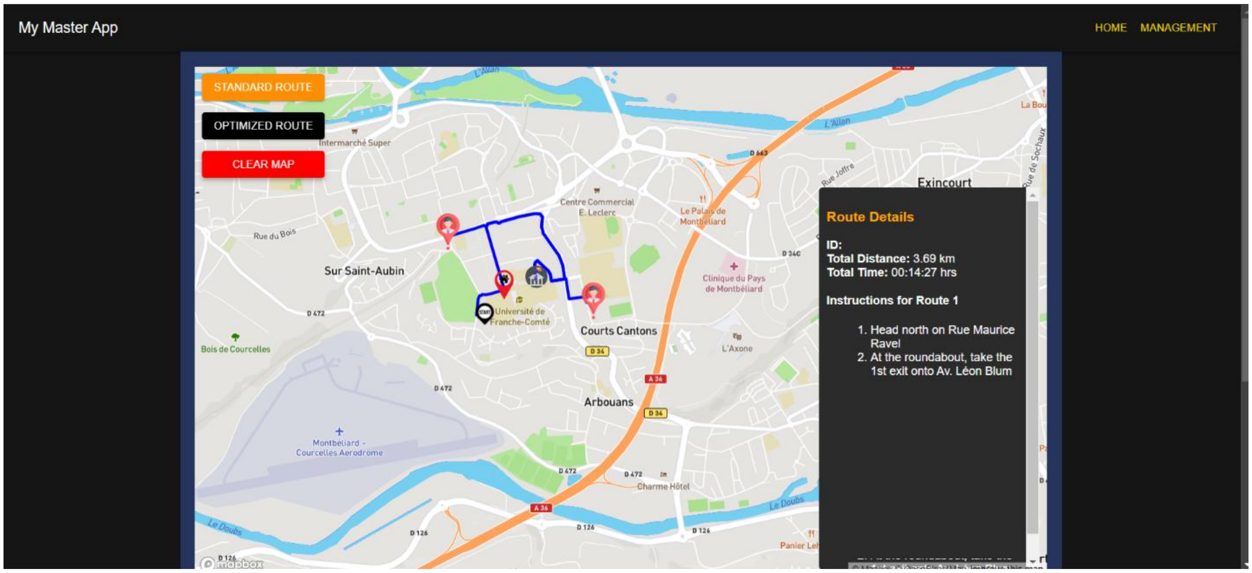
```
1
    latitude: 47.49575
    longitude: 6.80388
    name: "arret campus"
2
    latitude: 47.6386
    longitude: 6.8539
    name: "crous belfort"
3
    latitude: 47.23629
    longitude: 6.01566
    name: "crous besancon"
4
    latitude: 47.64245
    longitude: 6.8418
    name: "utbm parking"
```

d.Frontend

We are using MapBox to get the directions from the direction API. Inorder to guide the user from his location to the nearest meeting location for his and his friends.

GUI



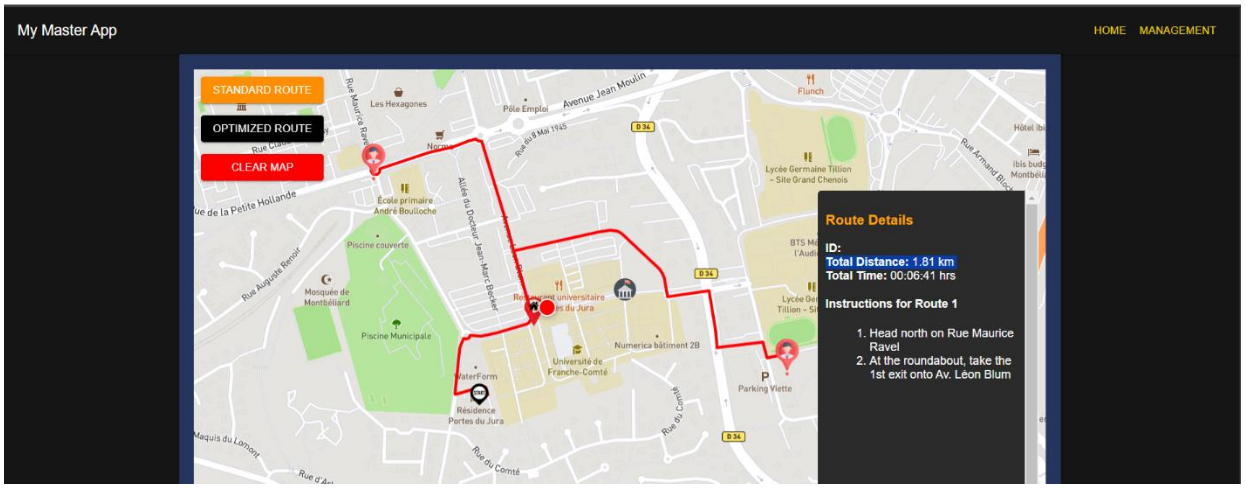


Route Details

ID:
Total Distance: 3.69 km
Total Time: 00:14:27 hrs

Instructions for Route 1

1. Head north on Rue Maurice Ravel
2. At the roundabout, take the 1st exit onto Av. Léon Blum



Route Details

ID:
Total Distance: 1.81 km
Total Time: 00:06:41 hrs

Instructions for Route 1

1. Head north on Rue Maurice Ravel
2. At the roundabout, take the 1st exit onto Av. Léon Blum