Smart Cab Allocation

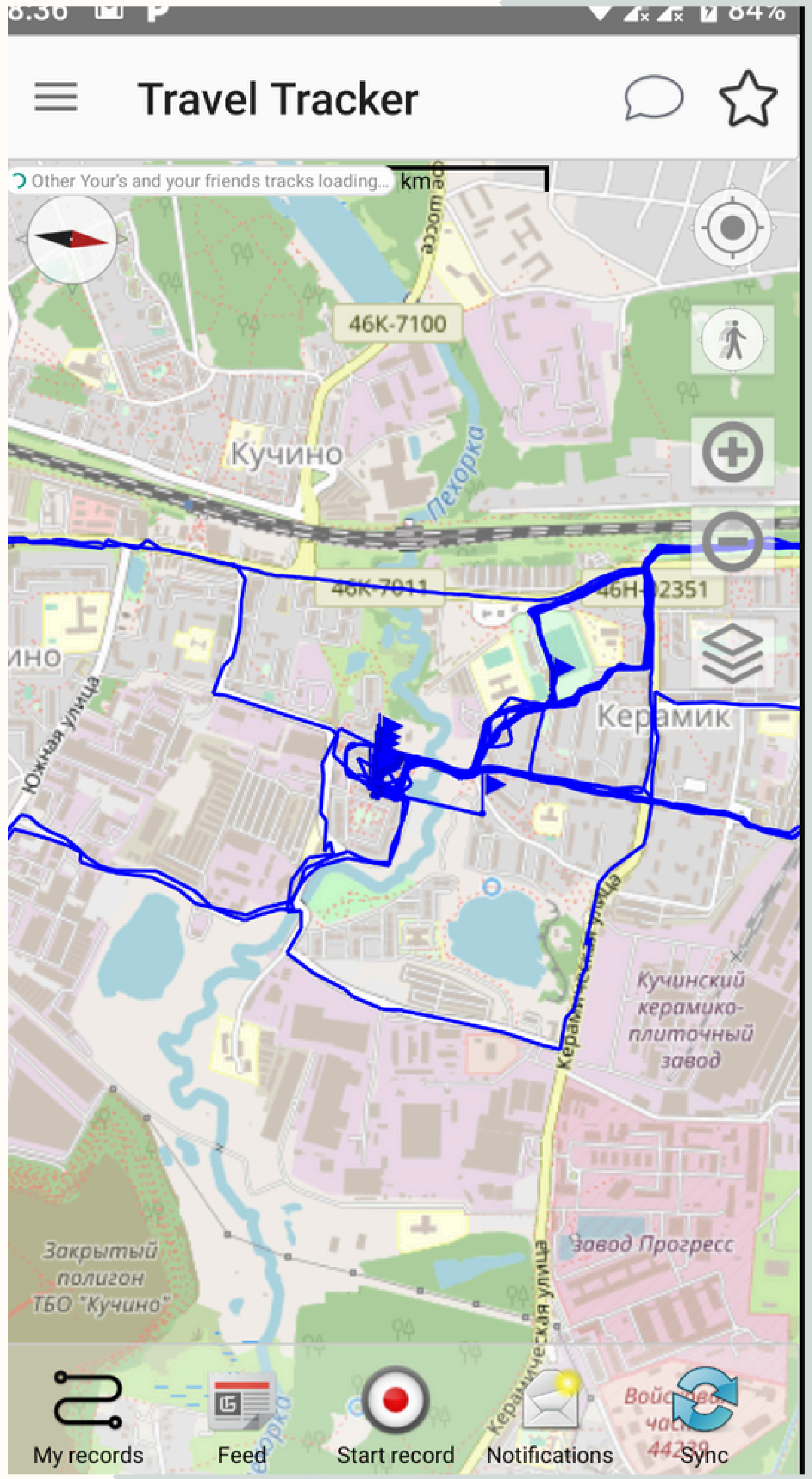
Algorithm Overview

1. Determine currently available nearby cab and the current cab route
2. Compute the effective distance for each nearby cab
3. Allocate the optimal cab

Road Network:

osmdroid - OpenStreetMap

OpenStreetMap (osmdroid) is an open-source Android library that enables developers to integrate OSM maps into their applications Key features includes offline map support, real-time location tracking etc.



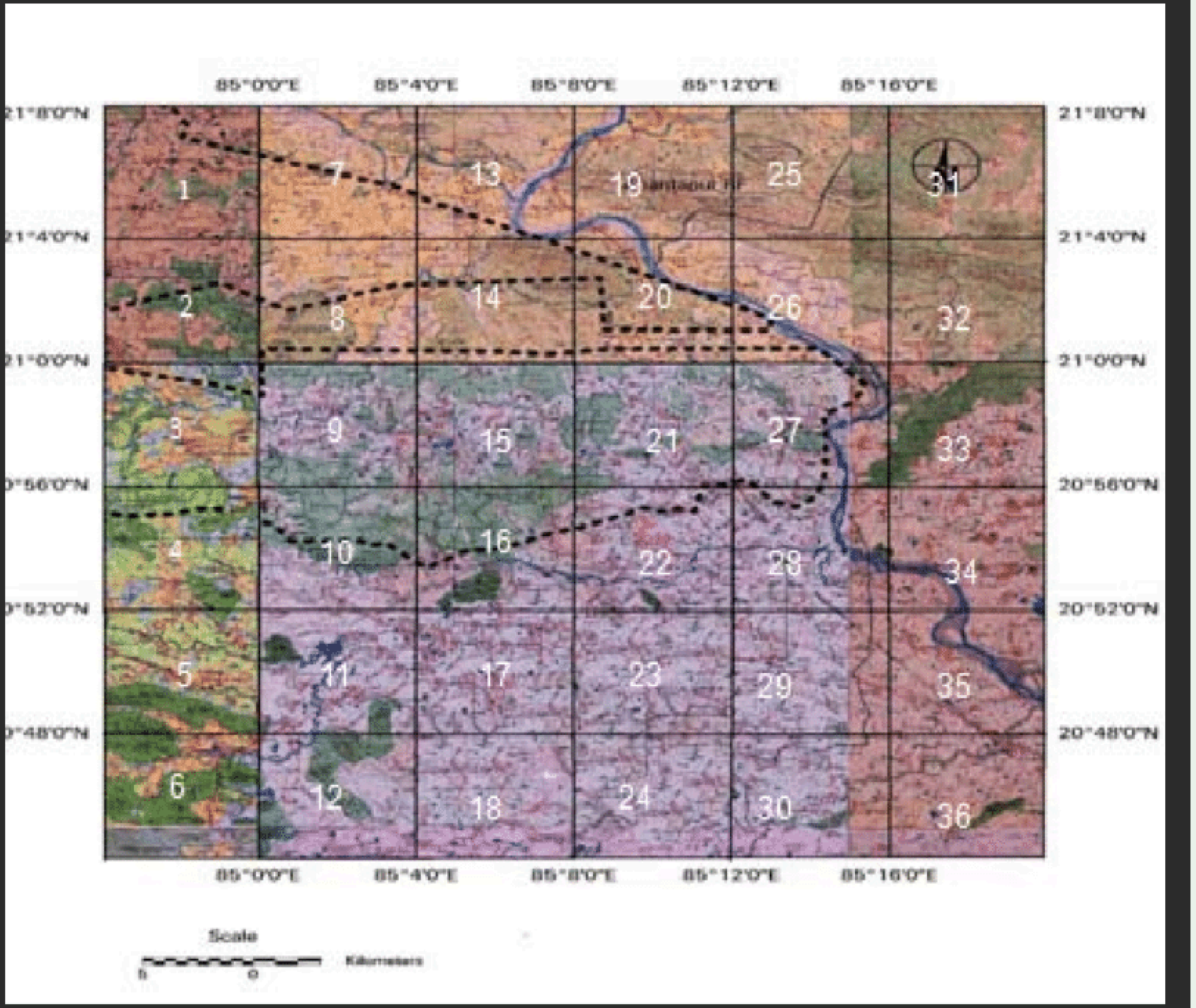
Using OpenStreetMap

1. **Spatial Division Using OpenStreetMap:**

   Utilized OpenStreetMap (OSM) data to divide the operational area into a grid of smaller, manageable cells.

2. **Cell-Based Search Algorithm:**

   When a user initiates a cab search, the algorithm identifies the specific cell corresponding to the user's current location.



Using OpenStreetMap

3 Efficient Data Handling

a. By limiting the search to a single cell, the algorithm minimizes computational overhead and accelerates search times.

b. This approach ensures faster response times and enhances the user experience by providing quick and relevant cab options.

c. Real-time data for cab locations is dynamically updated and stored in a database, indexed by cell.

India's approximate latitude and longitude boundaries are:

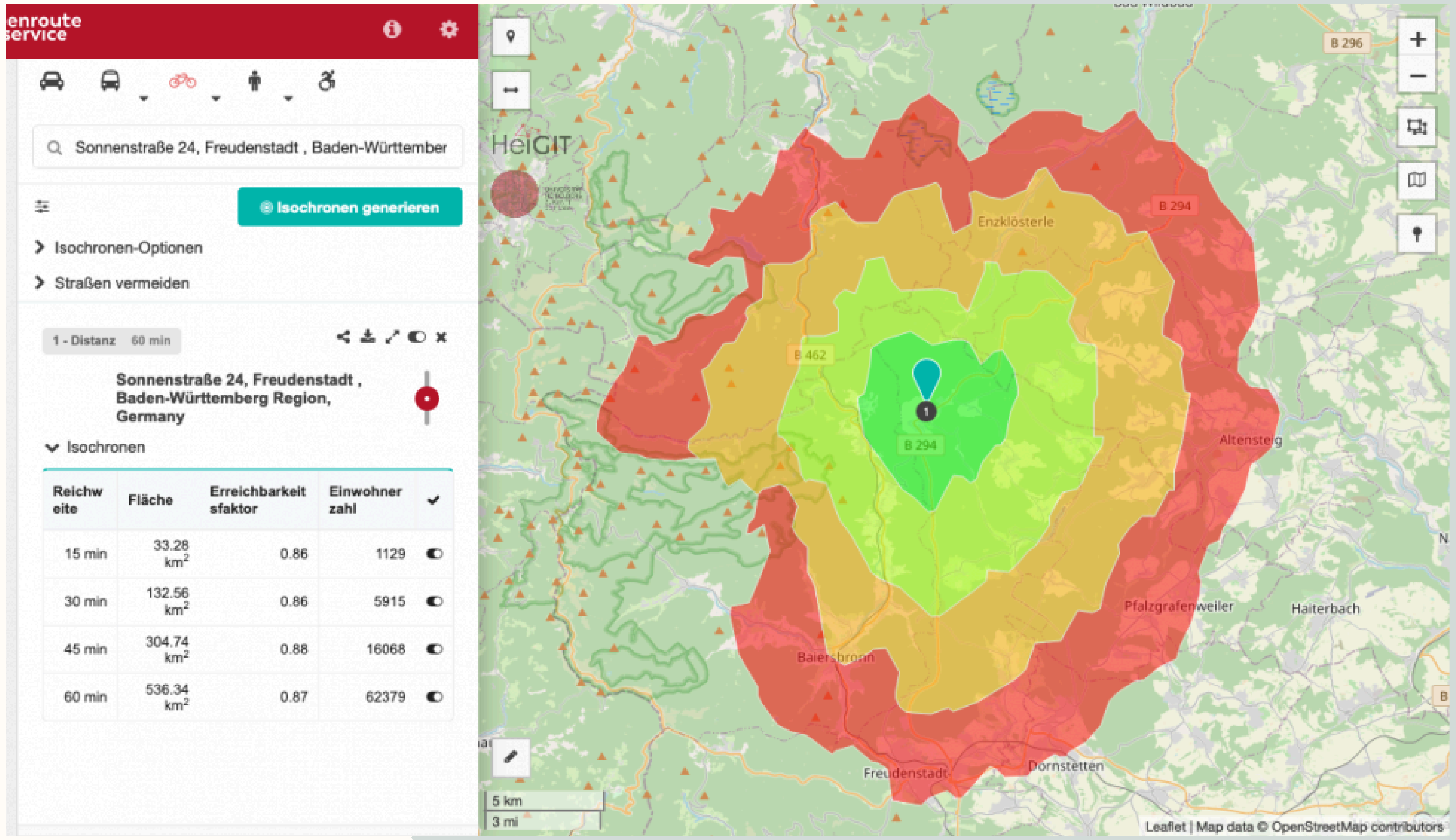
- Latitude: 8.4°N to 37.6°N
- Longitude: 68.7°E to 97.2°E

the cell size used in the grid
0.1° x 0.1°

Route Tracking

OpenRouteService API (ORS)

OpenRouteService (ORS) API is a powerful tool for integrating geospatial services into applications. It provides functionalities such as route planning, distance calculation, and geocoding. By using ORS API, developers can create efficient travel routes, optimize logistics, and enhance navigation systems with real-time data.



Sonnenstraße 24, Freudenstadt , Baden-Württemberg Region, Germany

Isochronen

Reichweite	Fläche	Erreichbarkeitsfaktor	Einwohnerzahl	✓
15 min	33.28 km^2	0.86	1129	◐
30 min	132.56 km^2	0.86	5915	◐
45 min	304.74 km^2	0.88	16068	◐
60 min	536.34 km^2	0.87	62379	◐

Using OpenRouteService API

1 **API Key**

The API key is required to authenticate requests to the ORS API and is unique to each user

2 **Request URL**

The URL for the ORS API includes the API key, starting point (start.longitude, start.latitude), and ending point (end.longitude, end.latitude).

Using OpenRouteService API

3 Processing the Response:

If the request is successful, the response will contain the route data in JSON format The features array in the JSON response contains route information.

4 Extracting Coordinates and Distance

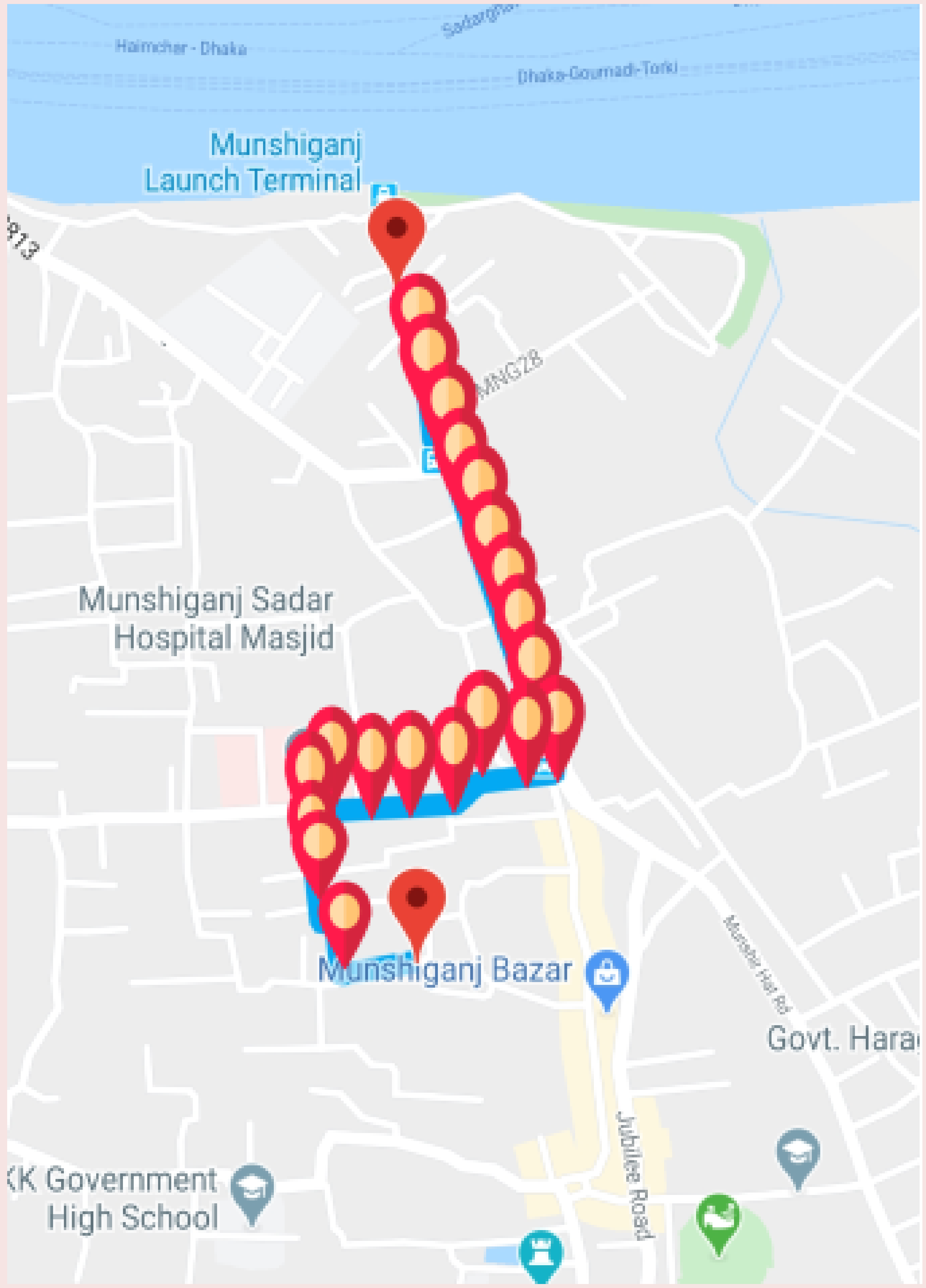
a. The coordinates from the JSON response are parsed and converted into a list of GeoPoint objects, representing the path points from the start location to the destination.

b. The distance is part of the properties object within the route's features array.

Using OpenRouteService API

5 Drawing the Route

The list of GeoPoint objects is used to draw a polyline (route) on the map, visually representing the route from the user's location to the destination.


Haimchar - Dhaka
Dhaka-Goumadi-Torki
Munshiganj Launch Terminal
MNG28
Munshiganj Sadar Hospital Masjid
Munshiganj Bazar
Govt. Hara
KK Government High School
Jubilee Road

Overall Time Complexity

Determine the user's cell and surrounding cells: O(1)

Retrieve drivers from relevant cells: O(1)

Filter drivers by exact distance: O(k)

where k = average number of drivers in a cell

Convert User Coordinates to Cell Index:

- **Latitude Index: (userLatitude - minLatitude) / cellSize**
- **Longitude Index: (userLongitude - minLongitude) / cellSize**