# **Requirements: Interactive Maps**

**CS4900/CS4910 - Soft Sys Dev: Requirements Document**

Client: Dustin Ian Black​

Group Members: ​

Bhargav, Praveen, Brandon​

**Requirements Specifications:**

**Explanation of Key Terms Used in the Project​:**

**Routing Algorithm:**

This is an algorithm that finds the best way to get from one point to another. It looks at things like how far it is, how long it will take, and the type of roads to find the quickest or easiest route.

**Topology:**

This term describes the kinds of roads used for making a route. Roads are classified based on how busy they are, how wide they are, if they have bike lanes, and the speed limits. They are rated from 1 to 4 for how stressful they are to use, with 1 being calm like neighborhood streets, and 4 being busy like main roads.

Level 1 - Neighborhood Street

Level 2 - Neighborhood Connector

Level 3 - Sub-Urban Connector

Level 4 - Priority/Main

**GPS (Global Positioning System):**

This is a system that uses satellites to tell you where you are on Earth. It's commonly used in cars and smartphones to help with navigation.

**API (Application Programming Interface):**

This is a set of rules that let different software programs talk to each other. It's like a translator that helps different programs understand and use each other's features.

**Overview and Purpose:**

**Overview:**

The goal of this project is to develop an application designed for the specific environment of Kalamazoo and Portage, Michigan. This app is intended to consider the bicycling experience in these areas by offering personalized, precise route recommendations for cyclists. This application has the capability to generate bike routes based on street types and their associated stress levels. This feature aims to ensure that each biking journey is not only safe but also efficient, catering to the unique needs and preferences of each user.

**Purpose:**

The aim of this bike routing app is to further and enhance the form of active transportation in Kalamazoo and Portage, Michigan. By introducing an innovative, user-friendly solution, the app focuses on three key aspects for cyclists: safety, convenience, and an enriched biking experience. Through this application, the project endeavors to make cycling a more appealing and accessible mode of transportation.

**Objectives​:**

**Technical Objectives:​**

* The application shall utilize the Map Box API to calculate biking routes based on various factors including traffic density, road width, presence of bike lanes, and speed limits.
* The application shall provide a user-friendly interface with radio buttons that allow cyclists to choose different road typologies, such Neighborhood Street, Neighborhood Connector, Sub-Urban Connector, Priority/Main. Cyclists can personalize their routes based on their comfort levels and biking preferences.
* Through integration with the Map Box API, the application shall offer GPS location tracking capabilities. Users will have the option to access their current location, and they manually enter their destination address. Throughout the journey, users will receive ongoing updates about their position and estimated time of arrival.
* The application shall include an amenity toggle feature that displays nearby amenities such as restaurants, toilets, bike repair shops, and bike rails on the map. This enhances the route planning experience by providing information on essential services along the way.
* The application shall ensure real-time responsiveness by continuously adapting to changing traffic and road conditions. This will be achieved through the utilization of Map Box API's real-time data and traffic updates.
* The application shall be compatible with the latest versions of Android and iOS for a wide user base.

**Business Objectives:​**

* The primary objective is to develop a user-friendly bike routing application designed specifically for Kalamazoo and Portage, Michigan. The application should offer an efficient and accessible solution for bicycle navigation within these cities. The emphasis is on ease of use, allowing users to effortlessly plan, customize, and navigate biking routes.

**End-user Needs and Expectations​:**

* The application must generate personalized biking routes for users, considering road stress levels.
* The application displays amenities preferred by the user providing them an option to toggle required amenities based on their preferences and needs.
* The application's user interface must offer a user-friendly and easily accessible experience for users.
* Ensure the application is compatible across various devices, including iOS and Android.

**Requirements:**

* 1. **Functional Requirements:​**
  + The application shall calculate biking routes using Map Box API, considering traffic, road width, bike lanes, and speed limits. (1.1.1)
  + The application shall utilize Map Box GPS functionality for accurate user location tracking and continuous position updates during travel. (1.1.2)
  + The application shall display route visualization, including distance and estimated arrival time, before starting turn-by-turn navigation using Map Box API data. (1.1.3)
  + The application shall provide turn-by-turn navigation instructions, including upcoming turns, distances, and street names using Map Box API. (1.1.4)
  + The application shall display real-time updates on the estimated time of arrival (ETA) based on the user's current position and route progress. (1.1.5)
  1. **User Interface Requirements:​**
* The application shall support zooming in and out for detailed locality viewing and display nearby amenities on the map. (1.2.1)
* The application shall include a GPS icon that, when activated, uses Map Box to fetch the user's GPS location. (1.2.2)
* The application shall ensure a user-friendly interface accessible on both Android and iOS devices. (1.2.3)
* The application shall have users input their destination address through a user-friendly interface. (1.2.4)
* The application shall implement interactive menus for selecting stress levels (Neighborhood Street, Neighborhood Connector, Sub-Urban Connector, Priority/Main) and toggling amenities (restaurants, toilets, bike repair shops, bike rails) using radio buttons. (1.2.5)
* The application shall provide real-time updates on the user's progress along the route during turn-by-turn navigation. (1.2.6)
  1. **Capability Requirements​:**
* The application shall calculate and display biking routes within 3 seconds of the user selecting the start route option. (1.3.1)
* The application shall promptly display detailed amenity data within 2 seconds of the user's selection. (1.3.2)
* The application shall offer audible turn-by-turn navigation instructions to promptly guide users along their selected biking route. (1.3.3)
* The application shall provide users with real-time updates on the estimated time of arrival based on their current position and route progress. These updates will ensure users have accurate timing information throughout their journey. (1.3.4)
  1. **Software Inputs and Outputs and Data Requirements​**
* Users must provide starting and ending destinations in a valid alphanumeric format: ADDRESS, City (Kalamazoo or Portage), MI, ZIP CODE. This ensures proper location information for route planning. (1.4.1)
* If users allow GPS access, their current location becomes the default starting point for route planning. (1.4.2)
* The application utilizes geojson data to store and display amenity information for Kalamazoo and Portage, MI on the map according to user’s preferences. (1.4.3)
* The application will collect and store user preferences for stress levels (e.g., Neighborhood Street, Neighborhood Connector, Sub-Urban Connector, Priority/Main) and amenities (e.g., restaurants, toilets, bike repair shops, bike rails). These preferences will be used to customize route recommendations and map displays. (1.4.4)
  1. **Software System Interface Requirements:**
* The application shall display turn-by-turn instructions, including the distance to each action point, for the user's specified biking route when they commence the route. (1.5.1)
* The application shall interface with the MapBox API to generate the base map and retrieve location information for location markers and amenity icons. (1.5.2)
* The application utilizes geojson data for Kalamazoo and Portage, MI to display amenity data. (1.5.3)
* The application shall have the capability to utilize GPS to determine the user's location, subject to their permission using MapBox API. (1.5.4)
* The application shall be compatible with multiple platforms, including Android and iOS devices. (1.5.5)

**Method to Requirement Traceability:**

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Method** | **Explanation** |
| 1.1.1 | getCyclingRouteUsingMapbox() | Calculate biking routes using Map Box API, considering traffic, road width, bike lanes, and speed limits. |
| 1.1.2 | getLocation() | Method of the location variable that pulls user current location with permission. |
| 1.1.3 | getDirectionsAPIResponse() | Display the calculated route, estimated arrival time, and distance to destination. |
| 1.1.4 | MapRouteNavigationScreenPage.buildInstructionInfo() | Provide turn-by-turn navigation instructions, including upcoming turns, distances, and street names |
| 1.1.5 | MapRouteNavigationScreenPage.calculatorUtils() | Display real-time updates on the estimated time of arrival (ETA) based on the user's current position and route progress. |
| 1.2.1 | MapboxMap() | Handles the map view and provides zoom in and out functionality. |
| 1.2.2 | controller.animateCamera() | UI element changes camera location to that of the current location data from the GPS location tracking. |
| 1.2.3 | \*Provided by Flutter\* | Ensures that the app is accessible on both Android and IOS. |
| 1.2.4 | LocationField() | Provides autocomplete functionality on user's destination address while they input the address. |
| 1.2.5 | HomeBottomBar() | Produces the bottom bar on the home screen which houses the amenity and topology menus. |
| 1.2.6 | MapRouteNavigationScreenPage.calculatorUtils() | Provides real-time updates on the user's progress along the route during turn-by-turn navigation. |
| 1.3.1 | MapRouteNavigationScreenPage() | Calculate and display navigation route. |
| 1.3.2 | MapboxMap() | Retrieves and displays amenity data on the map after user selection. |
| 1.3.3 | navigationController.setEnableAudio() | Provide audible turn-by-turn navigation instructions to promptly guide users along their selected biking route. |
| 1.3.4 | MapRouteNavigationScreenPage.calculatorUtils() | Provides users with real-time updates on the estimated time of arrival based on their current position and route progress. |
| 1.4.1 | getParsedResponseForQuery() | Gets users starting and destination address from API response. |
| 1.4.2 | \_useCurrentLocationButtonHandler() | Sets the user's current GPS location as the default start. |
| 1.4.3 | MapboxMap() | Uses geojson data to display amenity data on the map via Mapbox map options. |
| 1.4.4 | captureSelections() | Collects user preferences for personalized routing. |
| 1.5.1 | MapRouteNavigationScreenPage.buildInstructionInfo() | Provide turn-by-turn navigation instructions, including upcoming turns, distances, and street names |
| 1.5.2 | MapboxMap() | Interfaces with MapBox API for base map and location data. |
| 1.5.3 | MapboxMap() | Uses geojson data to display amenity icons on the map. |
| 1.5.4 | getLocation() | Method of the location variable that pulls user current location with permission. |
| 1.5.5 | \*Provided by Flutter\* | Ensures that the app is accessible on both Android and IOS. |