# **Requirements: Interactive Maps**

**CS4900 - Soft Sys Dev I: Reqmts & Dsgn​**

Client: Dustin Ian Black​

Group Members: ​

Bhargav, Praveen, Brandon, Samin​

**Requirements Specifications:**

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

**Routing Algorithm:** An algorithm that determines the shortest or fastest path between two points on a map based on various factors such as distance, time, and typology.​

**Topology:** The types of roads used to create a route distinguished by traffic density, road width, presence of bike lanes, and speed limit data. The stress level of the route is on scale of 1-4, 1 being the least stressful and 4 being the most stressful these are the types of streets:

1 - Neighborhood Street (incl. LN)

2 - Neighborhood Connector (incl. NB, CB, NN, EN)

3 - Sub-Urban Connector (incl. CC)

4 - Priority/Main (incl. UC, E/F, MS)

**GPS (Global Positioning System):** A satellite-based navigation system that provides location.​

**API (Application Programming Interface):** A set of protocols, and tools for building software applications.​

**HTTPS (Hypertext Transfer Protocol Secure):** An extension of the Hypertext Transfer Protocol (HTTP) that provides encrypted communication between a web server and a web browser.

**Overview and Purpose:**

**Overview:** The primary objective of this project is to design and create a specialized application tailored specifically for the environment of Kalamazoo and Portage, Michigan. This application will serve the purpose of providing personalized and highly accurate route recommendations for bicyclists, with a focus on ensuring both safety and efficiency throughout the entire biking experience. The central feature of this custom bike routing app will be its ability to generate route suggestions based on various factors, including street typologies and stress levels.

**Purpose:** The primary purpose of the bike routing app is to actively promote and enhance the practice of active transportation within the city of Kalamazoo and Portage, Michigan. This application seeks to achieve this goal by providing a user-friendly, technologically advanced solution that places a strong emphasis on the safety, convenience, and overall experience of bicyclists.

**Objectives​:**

**Technical Objectives:​**

* The application shall employ an algorithm that accurately assesses the stress level of bike routes based on the specific street typologies. This assessment will consider factors such as traffic density, road width, presence of bike lanes, and speed limit data to provide a precise stress level rating for each route segment.
* The application shall integrate the Map Box API to generate bike routes. Users will have the ability to toggle between different road typologies using radio buttons within the interface, allowing them to customize their route preferences based on their comfort level and biking experience. ​
* The application shall implement location-based services ([GPS Location Tracking](#Topology)) to enhance user convenience. Users will have the option to:
  + Access their current location through GPS functionality.
  + Enter their destination address manually.
  + Click on a location on the map to set their destination.
  + Receive real-time updates on their current position and estimated time of arrival as they progress along their chosen route.
* Enter their destination address manually.
* Click on a location on the map to set their destination.
* Receive real-time updates on their current position and estimated time of arrival as they progress along their chosen route.

**Business Objectives:​**

* Shall provide a user-friendly bike routing application designed to provide Kalamazoo and Portage, Michigan, with an efficient solution for navigating the cities by bicycle. The app will offer a seamless and straightforward user experience, ensuring that users can easily plan, customize, and follow biking routes.​

**End-user Needs and Expectations​:**

* Shall develop a bike routing application that prioritizes safety and convenience for users. It will accomplish this by considering factors such as the stress level (Will be calculated based on [traffic density, road width, presence of bike lanes, and speed limit data](#Topology)) of the route, street typologies, and the distance to the user's destination.
* The application’s user interface shall be designed to maximize user-friendliness. It will not require users to create accounts or log in to access its features. This design choice ensures a seamless and hassle-free experience for all users, promoting customer retention and accessibility. Shall offer an overlay of amenities on the map, including photos and text descriptions, enabling users to add photos and reviews about the amenities, as well as view photos and reviews added by other users.​
* The application shall be designed to ensure compatibility with a wide range of devices, including smartphones and computers (IOS, Android, Mac, Windows).
  1. **Functional Requirements:​**
* The application shall utilize advanced routing algorithms, such as Dijkstra's algorithm, to calculate and display the shortest biking route from the user's specified starting point (Point A) to their designated destination (Point B). This calculation will consider factors like traffic density, road width, presence of bike lanes, and speed limit data. The user will input their destination either by manually entering an address or selecting a point on the map interface or allow the GPS to retrieve their current location. The app will then process this input and generate the most efficient biking route (1.1.1).
* The application shall have the option to customize their bike route based on preferred street typologies, such as Event/Festival Street, Urban Center Street, Enhanced Neighborhood Street, Commercial Business Street, Neighborhood business street, City connector street, Downtown Main Steet, Neighborhood Network Street, Local Neighborhood Street. The app will offer a user-friendly interface, with selectable radio buttons and dropdown menus, allowing users to specify their preferences. The application will access and analyze street typology data to provide a biking route that aligns with the user's chosen preferences. This route will optimize the biking experience by following the selected street types (1.1.2).
* The application shall feature an overlay on the map, displaying nearby amenities relevant to cyclists, such as bike repair shops, rest areas, water fountains, and parks. Each amenity will be accompanied by high-quality photos and informative text descriptions. Users can access this information while planning their route to make informed decisions about rest stops and amenities along the way. Tapping on an amenity icon will reveal its details, providing users with a comprehensive view of available resources (1.1.3).
  1. **User Interface Requirements:​**
* Shall include an interactive base map that allows zoom in and zoom out, while allowing users to view bike routes according to their preferred [stress level](#Topology), amenity locations marked by representative symbols, and location markers that shows the user’s current location (either accessed through GPS or manual input) and destination with 99% accuracy, while having the capability of zoom in and zoom out (1.2.1).​
* Shall include an amenities menu that allows users to toggle on and off which amenities are shown on the map, a topology menu which allows user to toggle on and off which [street types](#Topology) the user prefers for their biking route, and a search bar which allows users to access their current location and enter their desired destination within the Kalamazoo and Portage city limits (1.2.2).
* Shall implement responsive web design that ensures the application interface is optimized for computer (desktop and laptop) and phone (Apple and Android) screen sizes and resolutions (1.2.3).​
  1. **Capability Requirements​:**
* Shall generate the most efficient route, taking the least amount of time determined by the distance of the route, that will be the default route if all [topologies](#Topology) are selected (1.3.1).​
* Shall calculate an adjusted route according to selected [street typologies](#Topology) that prioritizes using street types that were selected unless the use of a street type that was not selected is required to reach the destination. The calculation will also take route distance into account when choosing the best fit, this is to avoid users having to go too far out of the way to reach their destination (1.3.2).​
* Shall display the route within 5 seconds of submitting your destination or changing your selection of street topologies (1.3.3).​
* Shall display amenity data, including images, reviews, and descriptions, within 1 second of selection (1.3.4).​
  1. **Software Inputs and Outputs and Data Requirements​**
* Shall require the starting and ending destination input to be in a valid alphanumeric address format with the following structure: ADDRESS, Kalamazoo or Portage, MI, ZIP CODE (1.4.1).​
* Shall limit the starting and ending destination input to ZIP codes 49001, 49002, 49004, 49006, 49007, 49008, 49009 or 49048 (1.4.2).​
* Shall set the GPS current location as the default starting destination (if the user allows GPS access) (1.4.3).​
* Shall include a backend database to store amenity data such as photos and descriptions and display this data (1.4.4).​
  1. **Software System Interface Requirements:**
* Shall show turn-by-turn instructions, including the distance until each action is to be taken, for the biking route specified by the user upon beginning the route (1.5.1).​​
* Shall access MapBox API to generate the base map and retrieve locations for location markers and amenity icons (1.5.2).
* Shall access database to retrieve amenity data such as photos and descriptions upon selection of the icon (1.5.3).
* Shall be able to use GPS to find the user's location with their permission (1.5.4).​
  1. **Security Requirements:**
* Shall allow only authorized users with permissions to access and modify sensitive information on the website/database (1.6.1).
* Shall have communication between the website and the server encrypted using HTTPS to prevent unauthorized access or data breaches (1.6.2).
  1. **Risk Control Requirements:​**
* Shall encrypt sensitive data in transit and at rest to ensure that it is protected from unauthorized access (1.7.1).
  + Shall use HTTPS to encrypt data that’s being transferred.

**Design Specification:**

**Mock Web GUI:**

**Map

Description automatically generated**

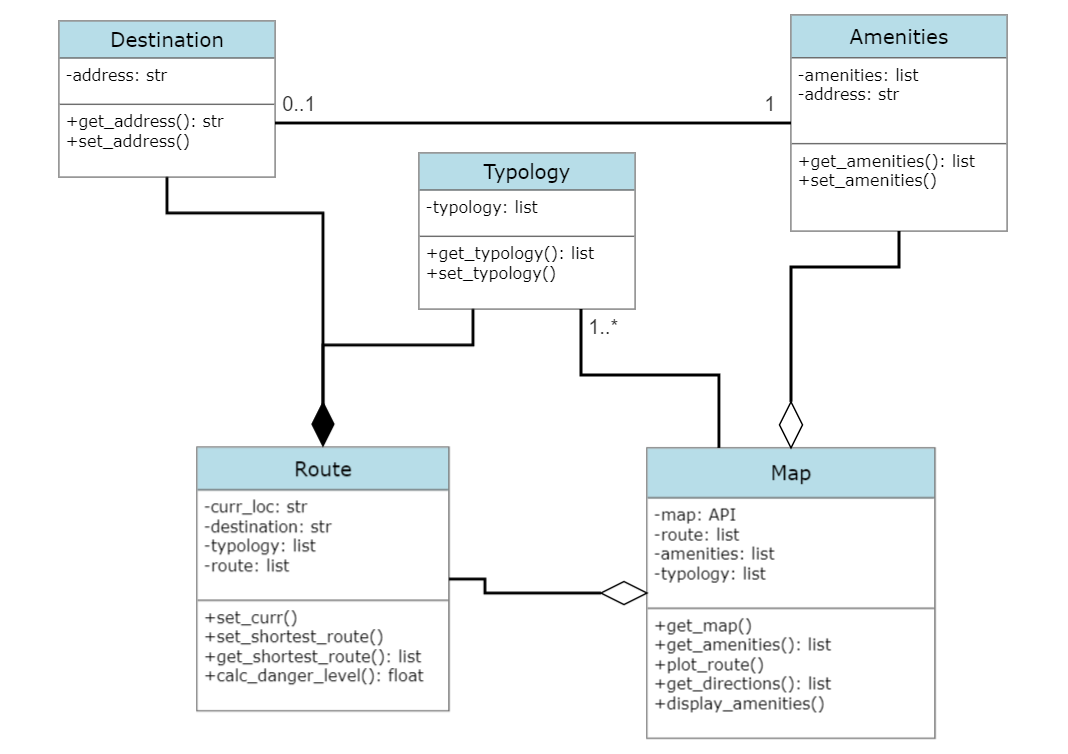
**Mock Mobile GUI:**

**Map

Description automatically generated**

* 1. **Usability Requirements​:**
* Shall be compatible on all types of devices with different screen sizes ranging from Smart Phones to Desktops (2.1.1).
  + Application shall be responsive on various devices without any overlapping icons or buttons.
  + Application shall have ham burger menu to accommodate any required links to different pages of the GUI
* Shall be able to enter their desired destination through the search bar (2.1.2).
  + Shall have a Starting and Destination field.
  + Shall have an option to add stops in between the route.
  + Shall display list of all the recommended addresses as the user types the starting/destination address.
  + Auto fill starting address if the user has enabled GPS.
* Shall be able switch between different typologies by clicking the check boxes (2.1.3).
* Shall be able to zoom in and zoom out using the buttons provided (2.1.4).
  1. **Functional Requirements​:**
* Shall allow users to input their starting location and destination, toggle between different road typologies to customize their route according to bikers’ preference (2.2.1).
* Shall display the user's route on a map interface, highlighting road typologies and amenities along the way (2.2.2).
* Shall provide feedback on the user's route, including estimated travel time, distance, and the level of stress associated with the selected route (2.2.3).
* Shall provide a responsive map interface that allows users to easily zoom in and out, pan, and switch between different map views including satellite view, dark view and light view (2.2.4).
* The system shall allow users to customize their route by toggling between different road typologies and amenities, and selecting the types of amenities they want to see along their route (2.2.5).
  1. **Interface Requirements:​**
* Shall implement an intuitive user interface layout with a map-centric design, utilizing the Mapbox mapping framework (2.3.1).​
* Shall employ a default light background theme, incorporating muted color schemes comprising pastel hues, earth tones, and subdued grays and blues, ensuring optimal visual clarity. Furthermore, it shall offer a user-configurable dark mode option for nocturnal usage scenarios (2.3.2).​
* Shall employ legible font types such as Arial or Helvetica, with a font size of no less than 12 points, consistently across all textual elements within the interface, including labels, buttons, and directional indicators. Legibility shall be maintained at small font sizes and when viewed from distances of up to 10 meters (2.3.3).
* Shall incorporate user-friendly map control mechanisms, including zoom in/out functionality and intuitive panning tools, such as drag-to-pan or pinch-to-zoom gestures, ensuring a smooth and responsive mapping experience (2.3.4)​
* Shall employ universally recognizable icons from a widely accepted package like the Google Maps icon library, accompanied by a comprehensive key or legend (2.3.5).
  1. **Performance Requirements​:**
* Shall render the biking route and map with no more than 3 seconds of delay on any supported device (2.4.1).​
* Shall allow users to make changes to the route, with no more than 3 seconds of delay between user action and application response (2.4.2).​
* Shall display all map elements, such as route markers and labels, with no less than 95% accuracy and precision (2.4.3).​
* Shall update real-time data on the biking route, such as distance, time, and direction at a frequency of no less than 1 second (2.4.4).​
* Shall provide turn-by-turn directions and distance with no less than 99% accuracy (2.4.5).​
* Shall display the biking route and map with a minimum resolution of 800x600 pixels on all supported devices (2.4.6).​
  1. **Capability Requirements​:**
* Shall enable users to input the origin and destination points for their biking route via either text or GPS location (2.5.1).​
* Shall provide users with options for their biking route by allowing for users to select their preferred [street typology](#Topology) (2.5.2).​
* Shall show the biking route on an interactive map with visible markers and labels for starting and ending points, as well as any selected amenities (2.5.3).​
* Shall present users with real-time data pertaining to the distance and time of the biking route (2.5.4).​
* Shall provide instructions to users for navigating the biking route, including turn-by-turn directions, distance of instruction, and street names (2.5.5).​
* Shall maintain a responsive user interface, ensuring the timely display of the biking route and map with no noticeable lag or delay (2.5.6).

**UML Diagram:**



**Method to Requirement Traceability:**

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Method** | **Explanation** |
| 1.1.1 | get\_shortest\_route() | Shortest route is calculated from starting point, destination. |
| 1.1.2 | get\_shortest\_route() | Typology is used to adjust the shortest route according to user request. |
| 1.1.3 | display\_amenities() | List of toggled amenities are displayed on the map. |
| 1.2.1 | get\_map() | Retrieves map through API. |
| 1.2.2 | get\_map() | Zoom features are included in the map API. |
| 1.2.3 | plot\_route() | In plotting the route, markers will be used for end points. |
| 1.2.4 | set\_amenities() | List of toggled amenities. |
| 1.2.5 | set\_destination() | Destination class correlates to the search bar feature. |
| 1.3.1 | get\_shortest\_route() | Generates the most efficient route |
| 1.3.2 | get\_shortest\_route() | Adjusts the typology according to users need while still providing the shortest route |
| 1.3.3 | plot\_route() | With further verification we expect to get it to load in less than 5 seconds |
| 1.3.4 | display\_amenities() | display amenity data, including images, reviews, and descriptions |
| 1.4.1 | set\_address() | Address will be verified upon declaring address. |
| 1.4.2 | set\_address() | Address will be verified upon declaring address. |
| 1.4.3 | set\_curr() | Current location will be set using GPS |
| 1.4.4 | get\_shortest\_route() | Shortest route is calculated from starting point, destination. |
| 1.5.1 | get\_directions() | List of nodes containing latitude, longitude and travel time. |
| 1.5.2 | calc\_danger\_level() | A value between 0 and 1 calculated taking into account typology and relative distance. |
| 1.5.5 | set\_curr() | Current location will be set using GPS |