**Kalamazoo Route Planner Requirements**

**Functional Requirements:**

1. The software shall plan a route for cycle/transit based on 2 nodes the user will enter on a map of Kalamazoo and Portage.
2. The software shall use different data structures to store the necessary data collected from the user and use a database to store augmented data retrieved from an open-source database. (What data structures are still being determined).
3. The software shall use a pathfinding algorithm with the data collected to compute either the shortest or most time efficient route.
4. The pathfinding algorithm shall also consider user input such as travel methods, road favorability, and amenities and shall reroute the user if required.
   1. Road favorability is determined by the city of Kalamazoo’s plan/worksheet to migrate roads to be more bike and pedestrian friendly by 2025.
5. The software shall then display the route to the user via a line along the roads from start node to end node and shall display turn-by-turn directions to the user.

**User Interface Requirements:**

1. The software shall display a map of Kalamazoo and Portage, the user will be able to zoom, scroll, and place nodes.
2. The software shall have a hidden navigation bar that can be opened on the left side of the window, navbar will contain type, favorability, amenities, reset, about, and update buttons and links.
3. Software shall have a radio style button displaying transportation types.
4. The software shall allow the user to use multiple sliders to dictate the favorability of certain types of roads.
5. The UI shall allow the user to turn on and off layers via checkboxes, these layers will contain amenities of the area.
6. The UI shall have a button to export the directions to a CSV or GPX.
7. The UI shall have an about link to display a popup to the user that displays info about the terminology of the web page.
8. The UI shall have a reset link in the navbar that allows the user to start over and reset the web page back to default settings.

**Capabilities Requirements:**

1. The software shall take 15-30 seconds to plan and display and will run on chrome, firefox and safari.
2. The software shall also run on Apple and Android devices by having a responsive web page based on display size.
3. The database shall use less than 10GB of space.

**Software Interface Requirements:**

1. The software shall connect to the OpenStreetMap (OSM) API, Overpass API, and a database pulling data from OSM.
2. A custom database shall be created and maintained via MySQL.
3. The user shall be able to export a GPX or CSV file that they can then use as input into other mobile GPS apps such as RideGPS.

**Software Input/Outputs and Data Requirements:**

1. The software shall take the start and end points from the user on the map.
2. The software shall use a radio style button to allow the user to pick their preferred mode of transportation.
   1. Radio button shall have options of
      1. Walking
      2. Biking
      3. Walking + Bus
      4. Biking + bus
3. The software shall allow the user to use sliders to set road preferences.
   1. Sliders will be a weighted option that will use the values in the database.
   2. These slider options will be
      1. Bike favorability
      2. Sidewalk favorability
      3. Car avoidance
      4. Other sliders to be determined
4. The software shall allow the user to select different amenities they want displayed on the map.
   1. These amenities include
      1. Stores
      2. Bike repair sites
      3. Bus Stops
      4. Restrooms
      5. Restaurants and cafes
      6. Other amenities are still to be determined
5. The software shall have a reset button in the navbar that will deselect all checkboxes and remove all markers on the map.
6. The navbar shall also have a link button that will take the user to an about page explaining how to use the site and the definitions of the terminology used on the web site.
7. The software shall output a series of directions in the web browser next to the map or drawn on the map.
8. The route will be able to be exported via CSV or GPX.
   1. These files will contain latitude and longitude coordinates that can then be used by another service or program, like the users preferred navigation app.
9. The software shall also take in data from the OSM API/database and link the data together with weights of favorability added to the roads.

**Security Requirements**

1. The update tab shall open or lead to a web page asking for a security phrase in a text field box that needs to be entered by administrators.
2. The passphrase shall be encrypted with sha-256 and salt for extra encryption. This will be used to verify that the database is up to date.
3. The database itself shall also have a username and password interface via a python script that runs a user interface for administrators to make changes to the database directly if needed.
4. The password for the database shall also be sha-256 encrypted with salt.
5. The web page shall follow the https security protocols to protect the user's information and location.

**Installation and Maintenance Requirements:**

1. The software shall be installed on a server that will require routine maintenance and bug fixing.
2. The Custom database shall be updated periodically via update link on web page, to make sure all the nodes and ways are linked properly and are up to date, this will be handled by a bash and python script that will interface with the database.
3. There shall also be a python script to interface with the database to change anything in the database if needed, manually.