

## 2 Project Plan

### 2.1 PROJECT MANAGEMENT/TRACKING PROCEDURES

Our group decided that we are going to use a hybrid of waterfall and agile as our project management style. For the design stage of the project, we are going to use waterfall as this stage consists mainly of the design document, and it will allow us to focus solely on each section of the design document at a time. Once we get to the implementation stage of the project, we are going to move to an agile approach that will allow us to work on the frontend and backend simultaneously, while also incorporating feedback that we get from our client/advisor during weekly meetings.

We are going to use GitLab to track version history and allow for simultaneous development. Trello will be used to assign and track tasks, and it will be used to monitor the project's overall progress. Lastly, group communication will be done through Discord.

### 2.2 TASK DECOMPOSITION

In accordance with the previous section for managing the project, here we list details of task decomposition where the shaded bullets represent our main tasks and the hollowed bullets represent the subtasks of their respective main task.

- Create the Design Document:
  - Identify use cases of project
  - Identify functional and non-functional requirements for project
  - Complete the project plan
- Build the Front-End UI:
  - Create a functional Figma Design
  - Set up base UI
  - Incorporate MapBox into the User Interface
  - Make MapBox features responsive to user input
- Build the Back-End:
  - Porting given algorithm
  - Storing data
- Connect front-end and back-end:
  - Develop API for communication between frontend and backend
  - Implement middleware to ensure frontend and backend work well together

### 2.3 PROJECT PROPOSED MILESTONES, METRICS, AND EVALUATION CRITERIA

Below are the milestones for our project broken down by semester. For estimated completion of each milestone reference the Gantt charts in section 2.4.

Fall Semester Milestones:

- Finalize tools and frameworks

- Complete preliminary system design
- Identify testing scenarios
- Refine the system design
- Refine testing scenarios and validation
- Finalize the design document

Spring Semester Milestones:

- Set up code base (Git, React project)
- Implement Visualization in MapBox
- Visualize route in MapBox
- Basic UI setup
- Use MapBox API to change map through user input
- Setup algorithms on backend
- Data is able to be stored
- Develop API for communication between frontend and backend
- Implement algorithm with frontend to change map

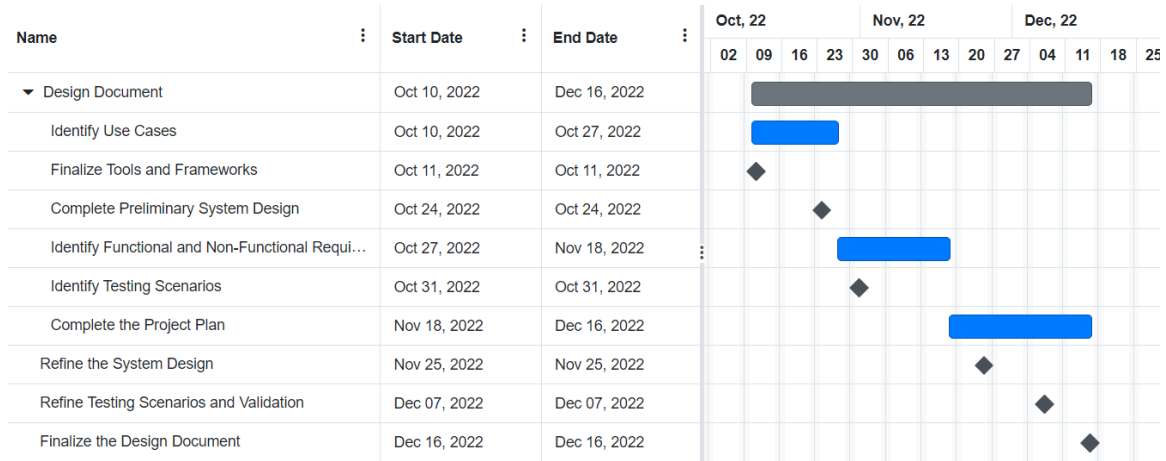
To measure the progress of the activities and completion of the milestones we will use the following metrics:

- General UI responds within a few seconds with correct information displayed. This will be evaluated by testing each of the UI's components.
- Displaying the locations and trajectory in Mapbox should correctly represent the one generated by the model. This will be evaluated by test cases corresponding to use case scenarios.
- Executing algorithms in the backend should complete within seconds. Correctness of the execution will be evaluated by using known scenarios (i.e., using some of the training data provided by the client).

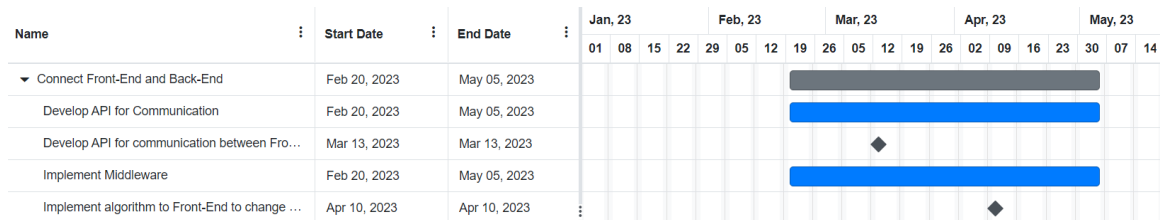
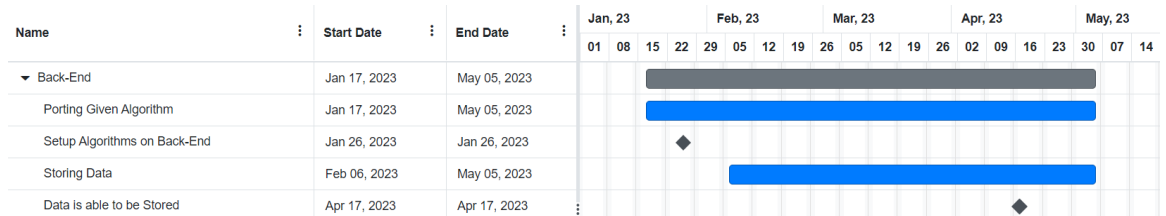
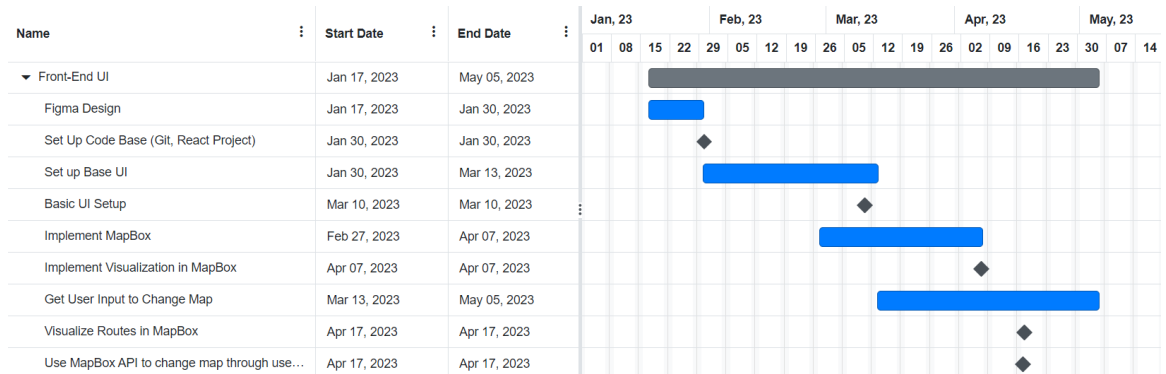
#### 2.4 PROJECT TIMELINE/SCHEDULE

The following Gantt charts represent the estimated project timeline. The bars represent the approximate length of the task and the diamonds represent the estimated completion date of milestones within each task.

## First Semester



## Second Semester



## 2.5 RISKS AND RISK MANAGEMENT/MITIGATION

Since our implementation is an Agile approach to the project we can assess risks and risk mitigation with each sprint. Below is a table of risks that we have identified surrounding the project. Each row specifies the potential impact of the risk. Each cell on the table is the probability that the specified risk occurs during our execution of the project.

Risk	Mapbox is shutdown	Given algorithm is non-functional	Chosen UI component is unavailable	Versioning error resulting in code loss	Dataset gets corrupted
Catastrophic		1%			
Major	1%				
Moderate				10%	
Minor					5%
Insignificant			20%		

## 2.6 PERSONNEL EFFORT REQUIREMENTS

Below is the projected person hours divided by task.

Task	Project number of hours	Explanation
Finalize tools and frameworks	5	Determine what technologies we will use for the front and backend
Complete preliminary system design	25	Make diagrams for the system to show how the internals will communicate
Identify testing scenarios	15	Create test cases to determine our website is visualizing routes correctly
Refine the system design	15	Refine the previous diagrams and add anything in case of additions
Refine testing scenarios and validation	15	Refine the previous test cases and process to validate them
Complete the design	60	Reviewing and revising design

document		document and discussing with advisor
Set up code base	10	Setting up React project, backend code, and connecting code bases to GitLab
Implement UI for user input	25	Creating the UI layout for all user inputs and input processing
Implement MapBox into UI and create MapBox components	45	Figuring out how MapBox API implementation works and implementing the API into our code
Use MapBox API to change map through user input	25	Figuring out all parts of the MapBox API and how we can utilize it for our project
Setup algorithms on backend	15	Setting up how algorithms work on backend
Develop API for communication between frontend and backend	35	Developing the API between the MapBox UI and the database/backend
Implement algorithm with frontend to change map	60	This will probably be more API work, but customizing it with the algorithms given on backend
Implement extra UI components	30	Extra components for more details on the routes
Stretch goal: Incorporate other attributes in ranking the selections	20	Rank the hotels by price or stars, rank museums by entry fee, etc.

## 2.7 OTHER RESOURCE REQUIREMENTS

Below are our other resources needed to fulfill the project requirements.

- Datasets
- Algorithms
- MapBox API
- Personal Computers