

Table 1: Revision History

Date	Developer(s)	Change
Sep 25, 2022	Priyansh, Utsharga, Sharjil, Jash, Bilal, Pranay	Rev. 0
...

Development Plan Greenway

Team #11, Roadkill
Priyansh Shah, shahp36
Utsharga Rozario, rozariou
Jash Mehta, mehtaj8
Bilal Shaikh, shaikb2
Pranay Kotian, kotianp
Sharjil Mohsin, mohsis2

[\[Put your introductory blurb here. —SS\]](#)

1 Team Meeting Plan

The team will meet once a week either Friday evening or Sunday afternoon to work on the development of Greenway. Each member will contribute equally in discussions centered around project design and development. The purpose of each meeting is to come together to make progress on the next milestone/deliverable for the project. Detailed meeting agenda points and to-do items will be specified before each meeting. A member of the team will take meeting minutes and record our discussion in our group Notion. Each meeting will end with a discussion of what needs to get completed before our next meeting.

2 Team Communication Plan

The team will communicate with one another primarily using Facebook messenger group chat. Issues regarding specifics within the source code or documentation will be raised on GitHub when necessary. Secondary documentation of our projects and conversations are recorded on the group Notion page. For urgent matters, all of our personal phone numbers and emails are also accessible.

3 Team Member Roles

There is no specific leader, we each have our own roles and responsibilities.

Jash Mehta

- Developer
- Project Manager
- Specialization in Git Version Control

Utsharga Rozario

- Developer
- Communication Officer
- Specialization in UI/UX and Frontend Development

Sharjil Mohsin

- Developer
- Minutes Taker
- Specialization in Backend Development

Priyansh Shah

- Developer
- Specialization in LaTeX and Documentation

Bilal Shaikh

- Developer
- Specialization in mapping APIs

Pranay Kotian

- Developer
- Scrum Master
- Specialization in Mathematical Calculations

4 Workflow Plan

Our teams workflow will begin with members of the team being assigned a feature of the product or a bug to fix based on the team meetings we had. We will use GitHub Issues to assign a member of a team a feature or a bug, and make sure to link the issue with the branch they will be working with. Before a developer starts working on a feature assigned to them or fixing a bug, they will pull the latest changes from the main branch and then create a new branch from it which will contain the addition of new or changed code. The developer will then commit the changes to this new branch with a commit message outlining what changes have been made, before creating a pull request when these changes are ready for review. Once a pull request is opened, GitHub Actions will run automated tests and unless all the tests have passed, the pull request will be blocked from being merged. After all the tests have passed and one other developer approves the changes being made, the changes will be ready to merge into the main branch. All these steps will be performed by all the developers of this team for the duration of the whole project.

The full workflow is summarized below:

1. Create and assign issues on the GitHub Issues interface
 - (a) Fill in the description for the issue
2. Pull changes from the main branch
3. Create a new branch off of the main branch
4. Commit changes to this branch
5. Create a pull request for the changes to the new branch
 - (a) Create a pull request for the changes to the new branch
 - (b) Add the corresponding description for the PR and link to the issue
 - (c) Have one other member of the team approves the changes
6. Merge the changes into the main branch
7. Once the changes are merged, remove the branch and issue

5 Proof of Concept Demonstration Plan

The main risks associated with POC demo is concerned with are web services which the project will depend on. One would be the API being used for MAPS services; another one would be a possible database that the app will need to connect to. We will demonstrate that the services our app depends on have a high enough uptime that function of our app will not be affected. Additionally, we will design our app to be ready for any possible service being down so the

user will be affected by a buggy application. Lastly, if primary functions are inaccessible during app downtime the user will be notified and the app will present alternate solutions or lock the user out of the app until functionality is restored. Testing for the backend of the application will be relatively easy since there will expected inputs and outputs for the application. But the frontend testing will be a relatively hard as good coverage is hard to achieve due to the number of different ways a user can interact with the website.

6 Technology

- Specific programming language
- Specific linter tool (if appropriate)
- Specific unit testing framework
- Investigation of code coverage measuring tools
- Specific plans for Continuous Integration (CI), or an explanation that CI is not being done
- Specific performance measuring tools (like Valgrind), if appropriate
- Libraries you will likely be using?
- Tools you will likely be using?

7 Coding Standard

8 Project Scheduling

This following link contains detailed information about our project scheduling and our Gantt Charts:

<https://github.com/mehtaj8/Greenway/tree/main/docs/DevelopmentPlan>