Development Plan

Smita Singh, Abeer Alyasiri, Niyatha Rangarajan,
Moksha Srinivasan, Nicholas Lobo, Longwei Ye

September 25, 2022

Table 1: Revision History

Date	Developer(s)	Change
Sept. 21, 2022	Abeer	Introduction, Team Members Roles, Proof of Concept Demonstration, Project Scheduling, Coding Standard
Sept. 21, 2022	Smita	Worked on Team Meeting Plan, Team Communication Plan, Workflow Plan, Technology, Coding Standard
Sept. 25, 2022	Niyatha	Basic edits for grammatical issues. Suggested better development tools and risk mitigation.
Nov. 23, 2022	Abeer	update POC plan to match the preparation the team had for the presentation
April 5, 2022	Abeer	update document to match what happened throughout the project.

This document will explore the development plan of Capstone project in terms of team logistics, project design, testing plan, and project execution

1 Team Meeting Plan

The plan is to hold meetings every Monday evening from 6:30 to 7:30pm. Meetings will be conducted through Microsoft Teams or when necessary will be held in-person at McMaster's Hatch building. A meeting agenda will be established before each meeting in order to keep the meeting succinct and on track. Agendas will be recorded using OneNote on the Microsoft Team's channel. The meetings will be used for division of labour, and completing deliverables. Other meetings will be held to conduct group work sessions to complete assigned tasks. These meetings will be less formal and will be scheduled based on the project scheduling.

2 Team Communication Plan

The primary form of communication will be through Microsoft teams, but a secondary form will be discord for urgent communication. Scheduling meetings and distributing tasks will be communicated through a shared Microsoft Teams channel.

3 Team Member Roles

All team members are expected to be responsible for completing technical software component to the project. The default roles assigned for the moment is to divide general topics across the project for members to have more knowledge than others. ~~A deliverable leader will be assigned during the first meeting of each deliverable and the role will be rotated across the team. Deliverable leader is a way for the team to keep track of tasks and ensure the last version is complete for grading.~~ The team leaders kept track on the progress of each deliverable by following up on the assigned tasks and making reminders and remarks on what needs to be updated.

Name	Primary Role	Description
Moksha Srinivasan	Team Co-Leader	~~Primary~~ Ticket Creator, Developer
Abeer Alyasiri	Team Co-Leader, Scribe	Ticket Creator and managing team progress, Liason between TA and supervisor, Developer
Nicholas Lobo	Git Expert	Managing and resolving git conflicts, Developer
Niyatha Rangarajan	~~Documentation~~ Technology Expert	~~Responsible for managing documentation~~ Responsible for researching new technology, Developer
Longwei Ye	Technology and Latex Expert	Responsible for gaining knowledge on new technologies and resolving latex errors, Developer
Smita Singh	LaTeX and Documentation Expert	Responsible for resolving latex errors and for managing documentation, Developer

Table 2: Team Roles

4 Workflow Plan

The team will be using a main branch to store stable code. Every feature, bug fix or code refactor will require a new branch to be made from main and then a pull request would need to be created. Each pull request must be reviewed and approved by two other members of the team in order for it to be merged into the main branch. This will ensure a decrease in error prone code being added to the main branch. The pull request will be merged as a squash commit.

Issues will be discussed and created by all of the group members and will be stored in the GitHub issue tracker. Issues will be classified by three classes: feature, bug, refactor, and milestone feedback. Each member will be assigned issues every week to work on when the implementation process has started. The team will establish team specific milestones, in order to keep track with deadlines and deliverables.

5 Proof of Concept Demonstration Plan

5.1 Most Significant Risk

The project will include three main risks that the team must address for a successful demonstration. The first one is scalability of the software to meet the project scope. This includes module designs related to performance and maintainability. The project is tackling multiple aspects of the ARC GIS system so it is important to design modules following the separation of concerns principles to ensure modularity in the design. Building a reliable design plan

before implementation eliminates the risk. The second risk is the project's functional capabilities within the specified deadlines since the project will be based on the ARC GIS system which include various components for GPS tracking. Hence, the team must prioritize the designing and implementing modules of core functionalities before expanding. This ensures completion of a system with fundamental features. The third risk is re-implementation of existing modules and tools of the system. The team must deliver a full Python3 re-implementation the system with accessible tools for technical users to follow and use.

5.2 Will a part of the implementation be difficult?

The team is not anticipating many problems with implementation except ensuring that only accessible libraries and tools are used to build the system. It may require more research and time to use the new technology with the team's existing set of skills.

5.3 Will testing be difficult?

Automated integration testing for multiple modules will be difficult because it requires enormous amount of work to gather proper data on how the modules will be used together. Therefore, automating such cases might not be feasible for the project.

5.4 Is a required library difficult to install?

No, because any required libraries will be available in the project's git repository.

5.5 Will portability be a concern?

No, since the project will have all the files available on the git repository as it is open source. Also, the potential users for the software are assumed to have prior knowledge to download the repository and run a starting Python3 module to use the system. Therefore, any technical user with Python pre-installed will be able to use the system.

5.6 Hardware Concerns

Not applicable for the project because no hardware is involved.

5.7 Proof of Concept Plan

The proof of concept demonstration will be providing a summary of the design architecture of the project and showing the fundamental modules in the form of a use case scenario run through of the input type, input processing, and expected output format. The scenarios must provide successful system functionalities to ensure proper design decisions, mitigation method for design risks,

and technical feasibility to complete the project scope. The demonstration will showcase ~~all~~ the rudimentary functional features of the system required for re-implementation of the toolbox ~~on a rudimentary level~~ to prove that the system is suitable for scaling ~~and follows a modular design principle. Also, demonstrating 80-90%~~ The demonstration will cover 30-40% of the project's main goals. It will establish that the team is ~~successful~~ on track to implement the open source toolbox functionality by end of April. ~~at completing the project end to the end.~~

6   Technology

6.1   Programming Language

The team will be using Python3 as the main programming language. Python is a great language for handling large data sets and for computing and performing statistical analysis. ~~The team may also use some R for certain features if/when necessary.~~

6.2   Development Tools

The team will also be using VSCode ~~along with RStudio~~ as an IDE. ~~The linting tool that will be used is pylint in order to keep our code consistent and clean. The team will be likely using the R5py python library for routing different types of transportation networks.~~ The team will utilize open source libraries to extract transportation and geospatial data, such as Overpass and OSMNX libraries. As well as GeoPandas which makes geo-spatial data easier to work with in Python. The data will be stored and fetched from MongoDB. GitHub is where the repository will be stored and issues will be created and tracked.

6.3   Documentation

Doxygen will be used to create documentation for the code. ~~Quarto files will be also be used in parallel to document the code in a more thorough manner. Quarto files will include markdown as well as code in order to create well documented code that is executable.~~ Juptyer files will also be used to easily communicate the product's uses cases by included parallel markdown sections.

6.4   Testing Frameworks

The unit testing framework that will be used is pytest. Pytest also has various plugins such as pytest-regressions which will be used to regression test our code.

6.5   Continuous Integration

~~Continuous Integration will be integrated through GitHub Actions, which will build the code and run the tests on virtual machines hosted by GitHub. CI tests are needed to pass during the pull request stage for the code to be merged~~

~~in.~~ Unfortunately, continuous integration was completed because the project timeline did not accommodate enough time for it to be done.

7   Coding Standard

The project coding standard will follow a custom set of guidelines that the team worked to ensure throughout the implementation followed it. These guidelines include:

- Limited use of global variables.

- Standard headers for different modules.

- Camel naming conventions for local variables, global variables, constants and functions.

- Indentation.

- Error return values and exception handling conventions.

- Avoid using a coding style that is too difficult to understand.

- Avoid using an identifier for multiple purposes.

- Code should be well documented.

~~the Pylint rules. This is because all team members will be using VS Code for Python and will enable the standard linting tool. This will ensure consistent, clean and readable code because Pylint is able to catch a number of bugs and disable unwanted warnings. PEP 8 – Style Guide for Python Code is another tool be used to ensure standard coding conventions.~~

8   Project Scheduling

The project schedule will be showcased using a Microsoft Planner that will be accessible in the Microsoft Teams channel so all the team members have access to view and update their progress directly on there. The team decided not to follow the recommended approach with Gantt Chart for project scheduling because it is not user friendly to make changes on and during team meetings. The team will be updating the planner to assign what parts each member will be responsible for and a leader to keep track of the milestone progress. Also, each team member will spend approximately 10 hours a week to complete deliverable sections and individual capstone course work. All tasks completed will be communicated on the planner with the description section.

Figure 1: Current state of the planner while working on Problem Statement and Development Plan