

Project Plan

Senior Design I

Project Title	Interactive evaluation of shortest path methods
Client & Advisor	Goce Trajcevski, Mengxuan Zhang
Team	sddec23-14
Team Members	Alex Blomquist, Samuel Caldwell, Selma Saric, Yadiel Johnson

1 Project Management & Tracking Procedures

The chosen methodology for this project was Agile due to some advantages that it has over the Waterfall method. Namely, the flexibility to adjust our priorities throughout the course of the project would fit better with the Senior Design course layout, and it allows for the client to have a higher influence throughout the project's development. The agile methodology provides various tools to enable this via providing a clearer understanding of the progress of the project as well as keeping team members updated on each other's work to identify potential roadblocks or issues during development.

The tools in the agile methodology we use can be found below:

- Every Tuesday involves a breakdown of current and upcoming assignments and goals. These get broken down into tasks that are submitted to our team's Trello board.
- The team maintains communication via Discord, where general discussion takes place. Team meetings are also held over voice channels.
- The team has access to a GitLab instance for the code's version control. It is reserved for the implementation phase of the project.
- Retrospectives after each sprint where the team and its members can reflect on their performance and identify areas of potential improvement.

2 Task Decomposition

Our team elected to showcase the project's task decomposition in two ways; a project conceptualization and a tabular rundown of the decomposition.

2.1 Project Conceptualization

In this section we present the team's derivation of what parts of the project need to be implemented to achieve the desired outcome. The approach that the team settled on was a Model-View-Controller design that leverages RESTful logic, and a further specification on the server side regarding an algorithm execution driver.

2.1.1 Backend

This section contains the preliminary design of the server-side components.

- Obtain and adapt implementations of the various shortest path algorithms as described in our Taxonomy document.
 - Modify algorithm implementations such that all I/O operations are standardized.
- Develop a "driver" that receives a dataset and a requested algorithm to run against it.
- Develop a server component that manages transactions with the web application and the algorithm driver.
 - Develop the "Controller" portion of the MVC pattern to communicate with the web application via a RESTful API.
 - Integrate with the driver to coordinate multiple algorithm executions.
 - Implement methods to receive, validate, and manage datasets submitted by users.

2.1.2 Frontend

This section contains the preliminary design of the client-side components.

- Create wireframes for the entirety of the web application to conceptualize its user interface.
- Create UI for the web app using HTML, CSS, and JavaScript
 - Develop a way for users to upload data sets.
 - Develop a way for the users to select algorithm(s) to run on their data sets.
 - Develop shortest-path algorithm visualizations.
 - Present algorithm runtime and metrics on the results screen.
 - Develop a method to generate reports that have comparisons between algorithms, including a method to store them.

2.2 Tabular Rundown

Below is tabular rundown of all major tasks that must be completed is presented further below. These tasks are necessary for the successful completion of this project. Note that the corresponding sprints that illustrate the agile methodology that the team has chosen is available in the Gantt chart below.

#	Description
1	Design the <i>System Architecture</i>
1.1	Design the server component
1.2	Design the driver component
1.3	Design the web app component
1.4	Adjust and adapt the algorithm-dataset suite
2	Design the <i>System Framework</i>
2.1	Design a standardized format for algorithm I/O
2.2	Design the REST endpoints
3	Design <i>Testing Framework</i>
4	Finalize Design Document
5	Prepare Server Environment
6	Implement Server Component
6.1	Add REST endpoints
6.2	Implement persistence
7	Implement Driver Component
7.1	Implement algorithm interface solution
7.2	Add “runtime and space complexity” metric gathering
8	Implement Web App Component
8.1	Implement basic UI
8.2	Add REST logic
8.3	Add user form submission
8.4	Implement algorithm output visualization
8.5	Add “comparison export” functionality
9	Implement Testing Suite
10	Final Presentation

3 Project Milestones, Metrics, and Evaluation Criteria

A detailed listing of the milestones for this project can be found below.

#	Milestone Title	Milestone Description	Metrics:
1	Finalize System Architecture Design	Working with the use-case diagram finalize a diagram that showcases how the frontend, driver, and backend of the system work together.	April 2nd
2	Finalize Design Document	Finalize the design document to work as an exhaustive summary of the details of the software's development.	April 23 rd
3	Acquire and Adapt Algorithm Code	Receive and analyze code provided by the project advisor in order to utilize it with the AED.	Sept. 10th
4	Develop the User Interface	Integrate the initial design of the UI into the frontend of the system architecture.	Nov. 11th
5	Develop Algorithm Execution Driver	Develop a module tasked with managing algorithm executions, runtime metrics, and related responsibilities.	Oct. 1 st
6	Develop the Server Component	Implement the environment and server where all web application requests will be handled, including algorithm executions and persistence.	Sept. 17th
7	Unit Testing	Ensure the functionality of each component by testing its functionality for potential software bugs during their operation.	Oct. 17th
8	Implement Algorithm Visualization	Include visualizations of algorithm results and traversal paths on the web app results screen.	Nov. 1st
9	Integration and Acceptance Testing	Test each component for their compatibility during interactions to detect any potential bugs or other potential system vulnerabilities during integration.	Nov. 17th
10	Final Software Release	Prepare the finalized version of the software system for presentation, with added emphasis on quality assessment.	Dec. 3rd
11	Final Presentation to Panel	Present the finalized version of our presentation with a demonstration of the software developed that showcases its functionality.	Dec. 8th

5 Risks, Management & Mitigation

Risks associated with the project can be found below.

#	Title	% Risk	Reason	Mitigation Strategy
1	Design System Architecture	0.3	Planning Stage	N/A
2	Design System Framework	0.3	Planning Stage	N/A
3	Design Testing Framework	0.2	Initialization Stage	N/A
4	Prepare Server Environment	0.2	Initialization Stage	N/A
5	Finalize Design Document	0.3	Documentation	N/A
6	Implement Server Component	0.2	Implementation Failure	N/A
7	Implement Driver Component	0.5	Implementation Failure	Test algorithm implementation to make sure they produce the expected results.
8	Implement Web App Component	0.4	Implementation Failure	N/A
9	Implement Testing	0.3	Testing Failure	N/A
10	Final Presentation	0.4	Documentation	N/A

6 Personnel Effort Requirements

#	Title	Hours	Explanation
1	Design System Architecture	30	The server, driver, and web app components will provide the foundations for the application to work.
2	Design System Framework	30	Design the applications functions which will interact with the
3	Design Testing Framework	40	Create a test suite for verifying the application works as intended.
4	Prepare Server Environment	40	
5	Finalize Design Document	20	Complete documentation describing the application's design in its entirety.
6	Implement Server Component	50	Implement a backend housing the algorithms utilizing a Java backend to interface with the driver component.
7	Implement Driver Component	50	Implement the algorithm functions and ensure they produce the expected results.
8	Implement Web App Component	50	This will involve implementing the UI and user functions to allow them to utilize the algorithms and data sets. It will also need to retrieve the

			metric gatherings and present them back to the user.
9	Implement Testing	40	Final round of testing all functionalities of the application.
10	Final Presentation	100	It will be necessary to develop a concluding presentation that will highlight our project's functionality along with its associated design methodology.

7 Other Resource Requirements

In unison with the client, only two other extraneous requirements were identified:

- Throughout the second semester, we may request a server from ETG since we need a server capable of handling multiple algorithm executions on large data sets.
- We do not expect to have any financial constraints for our project.