Algorithm Complexity Analyzer

Sprint Number 4
Date 12/1/24

Name	**Email Address**
John Gahagan	john.gahagan398@topper.wku.edu
Jay Mistry	jay.mistry627@topper.wku.edu

CS 396
Fall 2024
Project Organization Documentation

Contents

List of Figures

1 Project Team's Organizational Approach

To make sure the team has effective collaboration, our team started the project with in class meetings and used Discord as our primary communication platform outside of class. Discord allowed us to stay connected, discuss updates, and address any issues that came up during development. Additionally, we used GitHub for version control and task management. This allowed any changes to the code to be well documented and accessible to all team members. To optimize our workflow and prevent any individual from becoming overwhelmed, we carefully distributed tasks based on each member's strengths and expertise. Weekly team meetings were held to review the project's progress, identify challenges, and assign tasks for the upcoming week. A shared task list was created on Discord to outline specific responsibilities, track progress, and ensure accountability. This approach not only helped in balancing the workload but also helped with collaboration, as team members could offer help or share tips related to their assigned tasks.

2 Schedule Organization

There will be four iterations of the Gantt Chart with each iteration focusing on a specific sprint. This project is focusing on Sprint 4 and the tasks involved within this specific sprint.

2.1 Gantt Chart:

The focus of this Sprint is to design the microservices architecture for the Algorithm Complexity Analyzer. This involves outlining the structure of the three core microservices, defining their responsibilities, and planning how they will interact through RESTful APIs. The team will create clear API specifications to ensure stable communication between services. Additionally, the sprint includes planning for the deployment environment using Docker containers, which will enable isolated, scalable, and consistent setups for each service. By the end of this sprint, the project will have a solid architectural foundation, making sure of scalability and maintainability for future development phases. The Gantt Chart is located in the Project 4 folder within the zipped Project 4 directory that was submitted. After unzipping the folder, navigate to the Project 4 folder, where you will find the Gantt Chart named Gantt Chart Project 4.pdf. File path is Project4/Gantt Chart Project 4.pdf. The team also includes a image of the chart excluding the white space. This is in the project technical documentation folder/images/GanntChartImageProject4.png path.

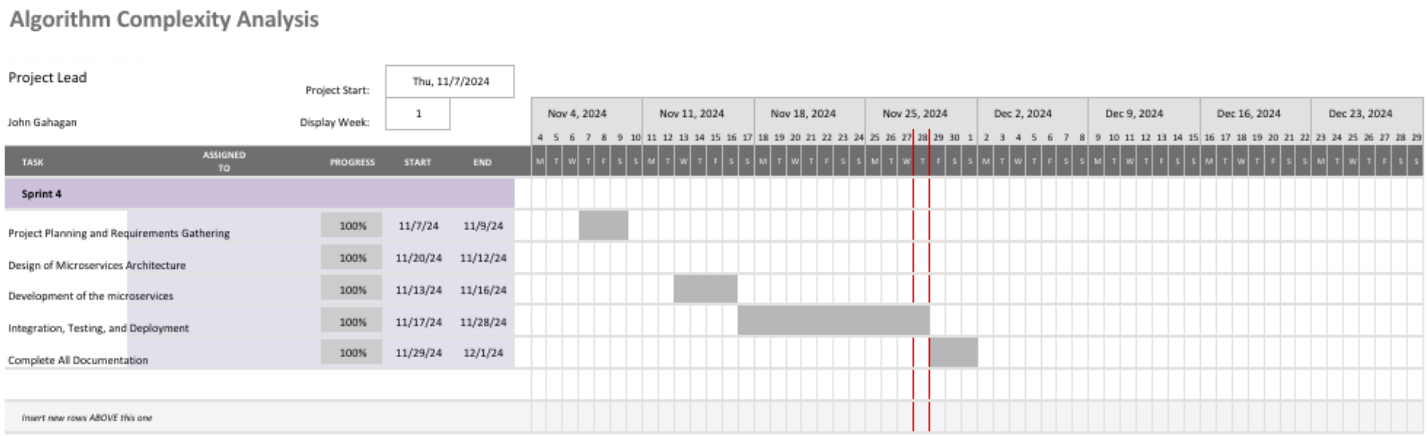

Figure 1: Gantt Chart for the Algorithm Complexity Analyzer

3 Progress Visibility

The main goal of our project is to create a tool that automates the analysis of an algorithm's time complexity. This includes developing microservices for Big-O notation determination, recurrence relation analysis, and result presentation. These components work together to give users a clear breakdown of an algorithm's efficiency. To

make sure of steady progress, tasks were assigned during weekly team meetings. In these meetings, we reviewed the project's overall progress, discussed challenges, and planned the next steps. Team members participated actively, volunteering for tasks that matched their strengths and interests. We used Discord as our main platform for task management and communication. It provided a shared space where team members could view their responsibilities, deadlines, and task dependencies. For instance, if one task depended on another, it was clearly noted to maintain transparency and avoid delays. Team members updated task progress on Discord and shared progress during meetings. If anyone faced an issue, they raised it on Discord or during discussions to get quick help. This approach kept everyone informed, organized, and flexible to adapt to any changes in the workflow.

4 Risk Management

The team took a proactive approach to risk management by identifying potential risks early during brainstorming sessions. Each risk was reviewed based on how likely it was to happen and how much impact it could have, and specific strategies were created to handle them. Risks were grouped into high, medium, or low categories, focusing on key areas like technical challenges, resource availability, integration issues, and scope creep. To address these risks, the team used strategies such as doing detailed research, keeping clear communication about availability, using continuous integration tools, and setting clear goals to avoid scope expansion. Risks were checked and discussed regularly in weekly meetings, and plans were adjusted as needed. This approach helped keep the project on track and ensured the development of a dependable tool for analyzing algorithm time complexity.