Group 7 Individual Contribution Report

Task Manager CLI Application

Prepared by Group 7

July 25, 2025

Belinda Belange Larose

I contributed to the project proposal and software documentation, working on the mission statement, problem analysis, and comprehensive system documentation that explains our Task Manager CLI application.

For the coding part, I worked on the core logic in `TaskManager.js`, implementing the main task management features including adding, editing, and deleting tasks. I also developed the `Task.js` class that handles individual task properties and methods like marking tasks as complete and checking if they're overdue.

One of the biggest challenges I faced was understanding how to properly structure the code so that different parts of the application could work together smoothly. I learned a lot about JavaScript classes, async functions, and how to write code that other people can easily understand and maintain. This experience helped me see how important good documentation and clean code structure are for team projects.

Grace Munezero

I worked on the `FileHandler.js` and `Validator.js` modules. I implemented the file input/output operations to load, save, and back up tasks using the file system. The `FileHandler` class ensured task persistence, automatic backups, and recovery in case of data corruption.

I also created the `Validator` class to enforce strong data validation, including date format checks and input sanitization. This helped maintain data integrity across all features.

Through this project, I improved my skills in modular coding, asynchronous file operations, and validation logic, contributing to the application's stability and reliability.

Plamedi Mayala

In this project, I was in charge of designing and building the entire command-line interface (`CLI.js`) for our Task Manager application. I used Commander.js and Inquirer.js to create an interactive experience that is both smooth and user-friendly. From welcome prompts and the main menu to creating, updating, filtering, and exporting tasks, I handled it all. I also added visual touches using Chalk for a better terminal UI.

Alongside CLI development, I focused on testing. I wrote and ran unit tests to ensure that features like task creation, toggling, search, and file operations worked correctly—even in edge cases.

One of the biggest challenges was designing a CLI that is powerful yet easy to use. This pushed me to write cleaner, modular code and focus on user experience. I gained a deeper understanding of JavaScript architecture and enjoyed the development process.

Kevine Umutoni

I developed the entry point logic in `index.js` to initialize and run the CLI application, managing the startup flow and ensuring smooth interaction between modules.

I co-authored the software documentation, including an overview of the system architecture, detailed UML diagrams, the README file, and the user-friendly usage guide. I also maintained all documentation in the `docs/` directory to keep information up to date and accessible for the team.

Additionally, I supported the team by clearly communicating technical decisions and documentation updates, helping everyone stay aligned throughout development.

This work was essential in keeping the project organized, providing clear guidance on the system's design and usage, and improving team collaboration.