Final Report Group 3

UMGC CMSC 495 Section 6381

Team Members:
Cook, Sean
Coutros, Peter
Daley, Tasciana
Geisler, Nicholas
Holman, Trevor
Jefferson, Michael
Malixi, Matthew Arvhie
Mussa, Nezifa
Suarez, Karolyn

Table of Contents

1. Overview

TaskCare Calendar is a comprehensive task and event management application designed to help users:

- Track deadlines
- Organize schedules
- Manage their time efficiently

The software specifically allows users to create, update, and delete tasks and events, ensuring a smooth workflow for both personal and professional use. TaskCare Calendar is built using Java with Spring Boot, Thymeleaf for the front-end, and an H2 database and was designed to be both a lightweight and powerful tool for managing daily, weekly, and monthly tasks.

2. Project Plan

The finalized Version 3 of the Project Plan is embedded in this document. Open it by clicking on the icon below.

Group 3 Project Plan

3. Requirements Specification

Requirement specifications are covered in the finalized project plan. Access them by clicking on the icon in the previous section.

4. System Specification

The application consists of a React-based frontend, a Java-based backend, and a MySQL database. The system is designed to create, edit, and manage tasks and calendar events. It supports CRUD (Create, Read, Update, Delete) operations for tasks and events and includes a reminder notification system. The MySQL database stores all user data, and the application is accessible through modern web browsers such as Chrome and Firefox.

5. User's Guide

The Calendar and Task Manager User Guide is a comprehensive document, vetted through multiple revisions by the team. It provides a step-by-step guide for each function of the application. Each process is accompanied by visuals to guide the user through the interface. The finalized Version 3 of the User Guide is embedded in this document. Open it by clicking on the icon below.

User Guide Group 3

6. Test Plan and Results

The test plan documents and results have been critical in ensuring that the application functions as expected. By systematically testing the application's frontend and backend, potential issues were identified and resolved. All problems discovered during testing have been fixed, and the test plan ran successfully with no failures.

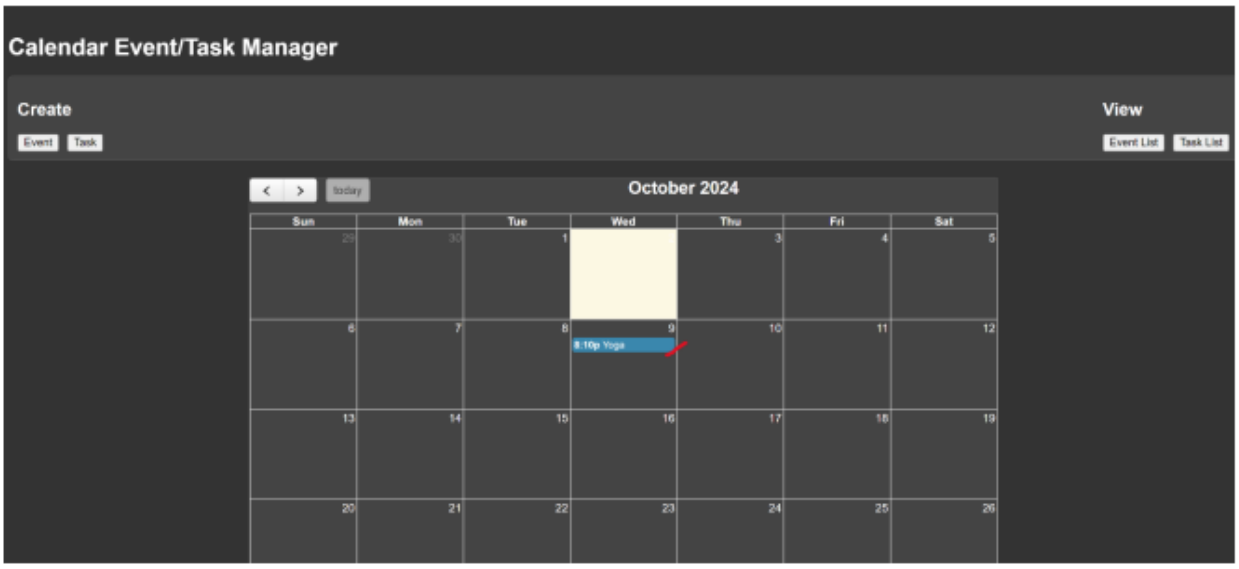
The Finalized Test Plan and its attachments are embedded in this document. Open them by clicking on the icon below.

Test Plan Group 3

7. Design and Alternate Designs

The TaskCare calendar application was designed to be as easy to us as possible, without too many additional features that could take away from the key functionality of the software. Below are images of the application with explanations of how each feature works.

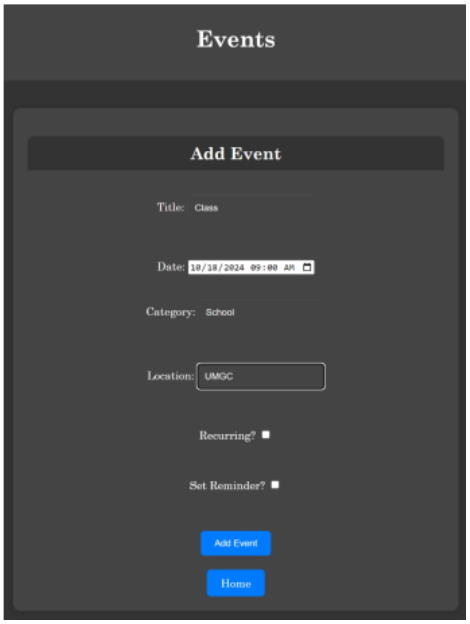
7.1 Main User Interface



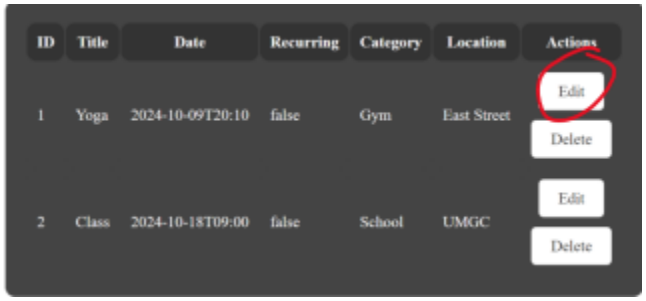
The main interface was designed to look like a standard calendar, where users can find events and tasks they have created. In the top left, there are buttons that allow the creation of new tasks and events, and the top right allows users to access a list of already created tasks and events. While viewing their task or event lists, users can edit and delete their tasks.

7.2 Task/Event Creation

To create a task or event, users can click one of the two buttons in the top left of the main interface. From there, the user will be directed to the menu pictured to the right and prompted to input all relevant information into the event or task. Once the event or task is complete, the user can add it to their calendar.



7.3 Editing/Deleting Existing Events



To edit or delete events, the user can click on the event list or task list in the top right corner of the main interface. All existing events and tasks will be provided in a list format, each with the option to be either edited or deleted. If the user is to click the edit button, circled in the image above, they will be taken to a menu similar to the create event menu seen in section 7.2.

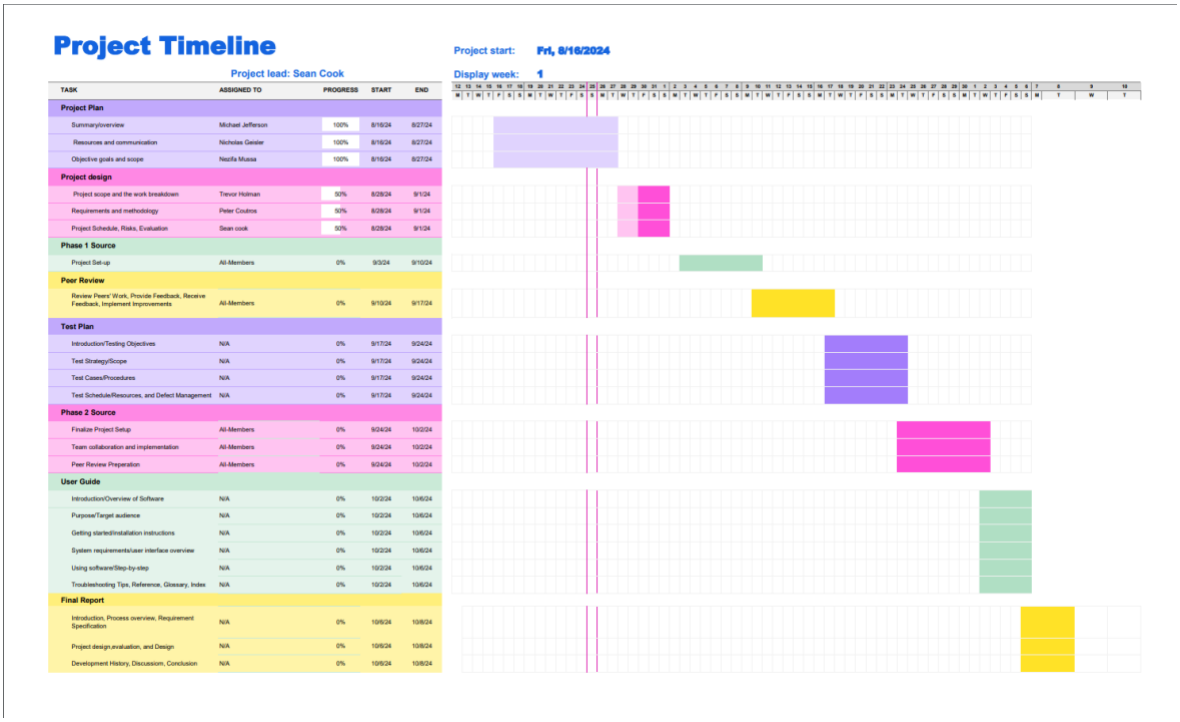
7.4 Project Design Document

A more detailed explanation of the project design can be found in the Project Design Document linked below.

Group 3 Project Design.docx

8. Development History

The best way to highlight the development history of the TaskCare calendar application is by simply examining the timeline chart created during the project design step of the planning process.



The vast majority of tasks were completed well in advance of their deadlines. The team worked well to delegate tasks and made sure that each goal was accomplished in a timely manner.

8.1 Relevant Documents

More detailed information about the development history can be found in the phase 1 and 2 source code documents, as well as the test plan document. All three of these documents can be found below.

9. Conclusions

Over the course of this project, Team 3 successfully designed, developed, and tested the TaskCare Calendar application, meeting our original objectives. The application serves as an effective tool for task and event management, with a user-friendly interface and essential functionality like task creation, editing, and deletion. By adhering to our project plan and

timelines, we were able to deliver a functional and reliable product. Through collaboration and troubleshooting, we enhanced our technical skills and project management capabilities. Moving forward, TaskCare has the potential for future enhancements, making it a flexible platform for users seeking efficient task organization.

9.1 Lessons learned

Throughout the development of the **TaskCare Calendar** project, several key lessons were learned. First, we realized the importance of flexibility in design. As the project progressed, we encountered design changes that required immediate attention, and being adaptable allowed us to address these issues without compromising the overall progress. Second, adhering to timelines proved to be crucial. By setting realistic milestones and sticking to them, we managed to balance the project tasks with personal commitments, ensuring that unforeseen issues didn't cause delays. Lastly, we learned to leverage available resources effectively. When we encountered knowledge gaps, we turned to research and online resources, discovering that many solutions were within reach with persistence and determination.

9.2 Design Strengths

Our design incorporated several strengths that enhanced the performance and usability of the application. First, we emphasized code efficiency by organizing reusable code into functions, which streamlined the development process and improved maintainability. Second, the graphical user interface (GUI) was designed to be intuitive, guiding users with highlighted input areas and providing clear notifications when errors occurred, ensuring a smooth user experience. Additionally, the program was future-proofed by making the interface easily modifiable, allowing for future updates and expansions based on user feedback. This ensured the software's continued relevance, adaptability, and accessibility for users who may not be familiar with setting up these environments.

9.3 Suggestions for Future Improvement

Several areas for future improvement were identified during the project. Expanding the customization options by offering more plate backgrounds beyond the current eight versions would enhance user experience. Another key improvement would be the automation of

expiration tracking in the database, relieving users from the burden of manual updates. Additionally, compiling the program into a standalone executable would simplify installation, making it more accessible to non-technical users, ultimately broadening the program's appeal and ease of use.