Project Report

Project:

Taskpplication

Table of Contents

1.0 Project Overview

1.1 Concept Introduction

Taskpplication is a task planner application. Users are able to create, edit, and delete tasks as well as organize them into categories and mark tasks as complete. Users can view tasks by the month or get a more detailed view by the week. Attempting to keep track of due dates, chores, and other tasks is oftentimes a hard task in itself for many people. This application aims to give these people a tool to keep their life organized.

1.2 Motivation

After pitching several ideas we settled on this project as it has a lot of relevance to us as students. We wanted a product that we could use in our everyday lives to improve productivity, increase organizational skills, and reduce stress. The idea of a task planner application came naturally after considering these values.

To come up with some of the core features we wanted our application to have, we took inspiration from some of our personal favorite applications such as Google Calendar and Todoist and combined it with our original ideas for a planner app.

2.0 Technologies

2.1 GitHub

GitHub was used as a tool for organizing the code for the project. This tool was particularly helpful because it allowed each member to work on different parts of the project simultaneously and then facilitate the integration of each part.

2.2 H2 Database

James decided on using a serverless H2 database to keep the implementation simple, not requiring the user to have a running sql server or download any complicated jars/run complicated scripts to get the application running. The database connection is initialized on application start, and the taskDAO(data access object) handles interaction between the application and the database through sql actions.

2.3 JavaFX

In order to follow a similar workflow to the homework assignments given in this class, JavaFX was used to handle the graphics of this project. All .fxml files were created using SceneBuilder and then imported into the project, which greatly reduced the time spent on designing the layout for the GUI.

3.0 Work Distribution

3.1 Chev Kodama

Chev wrote the Task, Month, and TaskController classes. Writing the TaskController class came along with adding methods to the ControllerHelper. Writing classes included writing Javadoc.

He also wrote the Testing Manual, scripts, and readme.

3.2 Vincent Tran

Vincent worked primarily on the visuals of the project. He used SceneBuilder to generate the .fxml files for every view in the project. He then set up the controller classes for each view to allow other members to work on the GUI integration without having to fully understand how .fxml files work under the hood. Vincent also worked more in depth on the calendar view of the project implementing features like row highlighting, switching between months, and transitioning to other views such as the week view and the task creation window.

Along with the GUI, Vincent also contributed to the documentation of the project by taking on the User Manual.

3.3 James Liu

James configured the h2 database, adding a database initializer class, as well as a data access object to interface with the database using simple methods. James also worked on the week view, connecting the coding logic (backend) with the frontend through the controller. James also handled task operations like marking a task as complete or deleting a task, as well as connecting different view transitions. James also contributed to the user manual and scripts.

3.4 Kirsten Szeto

Kirsten wrote the Day and Task Creator Controller classes with Javadoc comments. This entailed modifying small parts of other

classes to ensure smooth integration during development. Kirsten also contributed to the manuals and report.

4.0 Team Organization

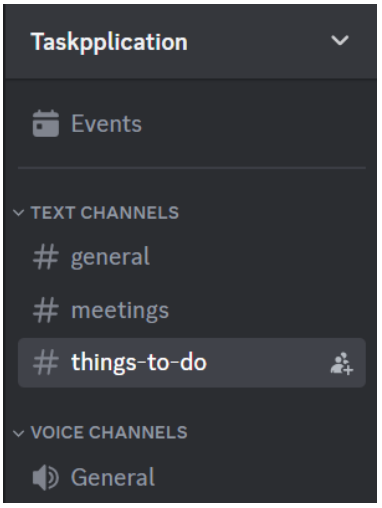
4.1 Team Meetings

We held an initial team meeting early on in the semester to craft the project pitch and delegate initial roles. This meeting was very helpful in setting up a rough plan such that the team had a general sense of what to expect moving forward, but it was also important to understand that a lot of plans created early on in the process are subject to change.

As we progressed towards the deadline of the project, there would be team meetings 1-2 times a week to discuss progress, future plans, and current obstacles. Meeting in person also allowed us to collaborate on code, which greatly reduced the time working through bugs.

4.2 Communication

A simple text group chat was used in the beginning stages of the project to facilitate any communication that needed to happen outside of meeting times. While this interface was sufficient at first, we quickly learned that a more organized form of communication was necessary as the project became more complex.



The discord server shown above was used as our primary form of communication during the actual development of the project. Discord was very useful as it gave us the ability to separate topics into different channels and organize future meetings.

5.0 Project Reflection

5.1 The Final Product

The final product accomplishes all functionality outlined in the project pitch along with additional features recommended by course staff. One of these features was to set up a database to store task information instead of using a csv or text file. This recommendation was very useful as utilizing a database is vastly more efficient when considering how the program might scale as more and more tasks are added. The other recommendation was to incorporate a month view as initially the program was only going to be able to display one week at a time. This feature vastly changed the concept of the project as initially the program was only meant to act as a weekly planner, but with this view, users are able to plan entire years in advance very quickly.

5.2 Future Improvements

If Taskpplication were to continue development there would be several functions that we would like to implement such as:

1. The ability to create new task groups
2. Add a GUI indicator in the week view to show what group each task is in
3. Add recurring tasks that automatically populate given the specifics for repetition

5.3 Takeaways

The project was very useful both in improving our knowledge of Java and included technologies as well as several non-technical skills such as time management and collaboration. Time management specifically was a very powerful lesson taken from this project as having to balance everyone’s schedules, additional assignments both inside and outside of this course, and creating a reasonable project scope was very difficult. Prioritizing the project earlier on in the semester as well as establishing an organized form of communication from the beginning would have helped with this problem.

6.0 Report Distribution

Section 1.0: Project Overview - Kirsten Szeto
Section 2.0: Technologies

- 2.1: Github - Vincent Tran
- 2.2: H2 Database - James Liu
- 2.3: JavaFX - Vincent Tran

Section 3.0: Work Distribution:

- 3.1: Chev Kodma - Chev Kodma
- 3.2: Vincent Tran - Vincent Tran
- 3.3: James Liu - James Liu
- 3.4: Kirsten Szeto - Kirsten Szeto

Section 4.0: Team Organization:

- 4.1: Team Meetings - Chev Kodma
- 4.2: Communication - James Liu

Section 5.0: Project Reflection:

- 5.1: The Final Product - Vincent Tran
- 5.2: Future Improvements - James Liu
- 5.2: Takeaways - Entire Team