

CST370 Group Project

Alexandra Miller

21 October 2025

Project Report

1. **Project Title:** “SeeQuence: A Personal Task Manager”

2. **Description:**

- The project is implemented as an interactive Java-based application that allows users to keep track of daily tasks. It utilizes implementation of the QuickSort algorithm to sort tasks according to user-assigned date and time.
- Users are able to execute commands “add”, “complete”, “plan”, “remove” and “exit”. The “add” command allows the user to add a new task to the task list. The “complete” command allows the user to change the status of a task to complete. The “plan” command allows the user to display the full list of tasks, including task id number, name, date, time, priority and status sorted in a sequential manner.
- The task priority is auto-assigned by the program based on the following logic. If the task is within a day from today’s date, it receives status of “Urgent”, within a 3-day receives status of “Moderate” and beyond 3 days is assigned as “Relaxed”.
- The “remove” command allows the user to remove a task from the list by task id.
- The “exit” command saves all the tasks and exits the program.
- The list of tasks is maintained by the program inside of the tasks.csv file. Everytime the program starts it loads all the tasks from the task.csv file into List<Task> taskList and conducts operations on that list. Upon exiting the program, everything gets saved to the “tasks.csv” and is ready for the next run.
- The program allows users a fun way to interact with the SeeQuence Task Planner application, while having flexibility to plan for important tasks such as grocery shopping, homework, appointments, meetings and more.
- If users happen to manually remove the “tasklist.csv” from the folder, the program is able to handle this by assessing that the file is not present and starting fresh with no issues.

3. Benefits:

- The benefits of using the QuickSort algorithm to sort tasks are as follows:
 - Quicksort algorithm is a good choice for sorting tasks by due date due to its flexibility and in-place nature.
 - **In-Place Sorting:** Quicksort is an in-place algorithm. This means it primarily sorts the data within the original array without requiring significant extra memory. It only needs a minimal memory for the recursion stack (which is $O(\log n)$ on average), making it memory-efficient for systems with limited resources.
 - **Handling Object Sorting (Tasks):** It allows for Comparison flexibility. Quicksort works by performing comparisons. In this implementation, it is adapted to sort complex objects (Task objects) by comparing date fields. This makes it a versatile tool for planning, as the comparison logic can be easily changed to sort by name, creation date, or any other attribute.

4. Tasks.

Task Distribution	Implementation
Application design, Object and data modeling, User experience	Alexandra Miller designed the application, data and object modeling and user experience. The idea was to make it a fun and interactive tool for maintaining tasks.
Application Testing, Test Case design	Alexandra Miller tested the application functionality to ensure the user is able to utilize all application features.
Code base, Debugging, Retooling	Google Gemini AI version Flash 2.5 was utilized to generate base files. Alexandra Miller worked to re-design class composition, fields and data storage capacity of the applications. Alexandra Miller worked to debug the application while testing its functionality to ensure smooth experience.

Test Cases:

User Input	Output
“add” “buy groceries” “2025-10-29 12:00” “plan”	ID: 1 Status: Pending Priority: Relaxed Due: 15:00 on 2025-10-29 -> buy groceries
“add” “homework” “2025-10-21 23:55” “plan”	ID: 2 Status: Pending Priority: Urgent Due: 23:55 on 2025-10-21 -> homework ID: 1 Status: Pending Priority: Relaxed Due: 15:00 on 2025-10-29 -> buy groceries
“complete 2” “plan”	ID: 2 Status: Complete Priority: Urgent Due: 23:55 on 2025-10-21 -> homework ID: 1 Status: Pending Priority: Relaxed Due: 15:00 on 2025-10-29 -> buy groceries
“remove 2” “plan”	ID: 1 Status: Pending Priority: Relaxed Due: 15:00 on 2025-10-29 -> buy groceries

Important Note: In order to run the SeeQuence Task Manager application, upon unpacking the TaskManagerApp zip file, please ensure to run the **javac *.java** command inside the folder where the following application files are located:

- **Main.java**
- **Task.java**
- **TaskSorter.java**

In order to start the application please execute the **java Main.java** command after compiling.

Test Cases (Screenshots):

TC1:

```
[skynet@skynetnode java % java Main.java
Starting a new task planner.

--- Hello! I'm Seequence, a Task Planner based on QuickSort---
Commands: 'add' | 'plan' | 'complete <taskID>' | 'remove <taskID>' | 'exit'

Enter command ('add', 'plan', 'complete <taskID>', 'remove <taskID>', 'exit'): add
Enter task name: buy groceries
Enter due date/time (YYYY-MM-DD HH:MM): 2025-10-29 12:00
Task added: buy groceries (ID: 1)

Enter command ('add', 'plan', 'complete <taskID>', 'remove <taskID>', 'exit'): plan

--- Planned Tasks (Closest Due Date First) ---
ID: 1 | Status: Pending | Priority: Relaxed | Due: 12:00 on 2025-10-29 -> buy groceries
-----
```

TC2:

```
Enter command ('add', 'plan', 'complete <taskID>', 'remove <taskID>', 'exit'): add
Enter task name: homework
Enter due date/time (YYYY-MM-DD HH:MM): 2025-10-21 23:55
Task added: homework (ID: 2)

Enter command ('add', 'plan', 'complete <taskID>', 'remove <taskID>', 'exit'): plan

--- Planned Tasks (Closest Due Date First) ---
ID: 2 | Status: Pending | Priority: Urgent | Due: 23:55 on 2025-10-21 -> homework
ID: 1 | Status: Pending | Priority: Relaxed | Due: 12:00 on 2025-10-29 -> buy groceries
-----
```

TC3:

```
Enter command ('add', 'plan', 'complete <taskID>', 'remove <taskID>', 'exit'): complete 2
Task ID 2 marked as Complete.

Enter command ('add', 'plan', 'complete <taskID>', 'remove <taskID>', 'exit'): plan

--- Planned Tasks (Closest Due Date First) ---
ID: 2 | Status: Complete | Priority: Urgent | Due: 23:55 on 2025-10-21 -> homework
ID: 1 | Status: Pending | Priority: Relaxed | Due: 12:00 on 2025-10-29 -> buy groceries
-----
```

TC4:

```
Enter command ('add', 'plan', 'complete <taskID>', 'remove <taskID>', 'exit'): remove 2
Task ID 2 successfully removed.

Enter command ('add', 'plan', 'complete <taskID>', 'remove <taskID>', 'exit'): plan

--- Planned Tasks (Closest Due Date First) ---
ID: 1 | Status: Pending | Priority: Relaxed | Due: 12:00 on 2025-10-29 -> buy groceries
-----
```