Project Name: Task-Sphere
Proposer's Name: Ashton McBride
Date of Submission: 2024-10-13
Version Number: v1

Task-Sphere Project Plan

I. Executive Summary

Task-Sphere is a task management system designed to boost users' productivity by managing their to-do lists and tracking their schedules.

II. Objective

To provide users with an easy-to-use tool that lets them create, manage, and track tasks, set due dates, and receive notifications for important reminders on their to-do lists

III. Background

Many people struggle with managing their tasks efficiently, leading to missed deadlines, overlooked responsibilities, and increased stress. Existing task management systems often lack integration of advanced features such as customizable reminders and comprehensive task tracking. This application offers a solution that combines task management with effective notification and deadline features, allowing users to stay up-to-date on their tasks.

IV. Features

A. User Login: Secure sign-up and login functionality.
B. Task Creation and Management
    1. Users can create, edit, and delete tasks.
    2. Tasks can be categorized into different categories.
    3. Users can create, edit, and delete categories.
C. Deadlines and Reminders
    1. Set deadlines for each task.
    2. Receive a customizable notification for each task.
D. Priority Levels
    1. Assign priority levels (Low, Medium, High) to tasks.

E. Search Functionality
    1. Search through any existing tasks based on due date, category, priority.
F. Task Completion Tracking
    1. Determine whether a task is complete or not

V. Technology Being Used

A. Programming Language: C++
B. Development Environment: Microsoft Visual Studio 2022
C. Libraries and Frameworks (Notifications): Platform-specific libraries for notifications (e.g., Windows API for Windows)
D. File I/O: Standard C++ file handling for saving and loading tasks

VI. Methodology/Approach

Software Development Methodology: Agile with iterative development (develop the project in increments, allowing for frequent adjustments based on feedback).

VII. Timeline & Milestones

- Project Duration: 8 weeks
- Key Milestones and Delivery Dates:
    - Week 1-2: Initial setup, project planning, and basic task management functionalities
    - Week 3-4: Implementation of deadlines, reminders, and priority levels
    - Week 5-6: Development of task tracking, search, and filtering features
    - Week 7: User interface design and integration of features
    - Week 8: Final testing, debugging, and preparation of project documentation

VIII. Expected Outcome

At the end of the project timeline, I expect my application to be a functional task management system with core features including task creation, deadline management, notification reminders, priority levels, and basic task tracking.

IX. Test Plan

A. This section outlines the testing approach that will be used for the Task-Sphere management system. The objective of this section is to verify that all features and functions will work as intended.

B. Scope of Testing
   1. Viewing tasks, creating tasks, and editing tasks
   2. Handling of categories (create, update, delete)
   3. Adding and removing tasks from categories
   4. Searching for tasks

C. Testing Strategy
   1. Manual testing of features based on test cases
   2. Functional testing for each action and state change
   3. Regression testing to ensure no new bugs are introduced

D. Schedule
   1. Initial functionality testing (View, Create, Edit)
   2. Category-related testing (Create, Update, Delete)
   3. Analysis and final review

E. Resources Required
   1. C++ IDE such as Microsoft Visual Studio 2022
   2. Computer system

X. Test Cases

A. Test Case 1: View Tasks
   1. Description: Confirm that the user can view all tasks, view tasks by category, and search for tasks.
   2. Preconditions: At least one task exists.
   3. Steps:
      a. Select "View Tasks" from the main menu.
      b. Select "View Categories" to view tasks by category.
      c. Use the "Search Tasks" function.
   4. Expected Result: Tasks are displayed, and search function works as intended.

B. Test Case 2: Create Task
    1. Description: Confirm that a new task can be added to an existing category or a new category.
    2. Preconditions: Application is running.
    3. Steps:
        a. Select "Create Task" from the main menu.
        b. Choose to add the task to an existing category or create a new one.
        c. Expected Result: Task is created and appears in the selected category.

C. Test Case 3: Edit Task
    1. Description: Test updating various task attributes.
    2. Preconditions: A task exists.
    3. Steps:
        a. Select "Edit Tasks" from the menu.
        b. Select which task to update or delete.
        c. Update the name, description, due date, reminder, priority level, or status of the task.
        d. Expected Result: Task updates are shown in the system.

D. Test Case 4: Edit Categories
    1. Description: Confirm categories can be created, updated, and deleted.
    2. Preconditions: Application is running.
    3. Steps:
        a. Select "Edit Categories."
        b. Create, update, or delete a category.
        c. Add or remove tasks from categories.
    4. Expected Result: Categories are created, updated, or deleted.

XI. Test Scenarios

A. Scenario 1: A user needs to update the due date of an important task and confirm that the task remains in the correct category.
    1. Steps:
        a. Launch application
        b. Select “Edit Tasks” from the main menu
        c. Select “Update Task”
        d. Select “Update Due Date”

B. Scenario 2: A user mistakenly deletes a task and creates a new task with similar details under the same category.
    1. Steps:
        a. Launch application
        b. Select “Edit Tasks” from the main menu
        c. Select “Delete Task”
        d. Navigate back to menu
        e. Select “Create Task”
        f. Select “Add to Existing Category”

XII. Traceability Matrix

Requirement	Test Case ID	Status
View Tasks	1	TBA
Create Task	2	TBA
Edit Tasks	3	TBA
Edit Categories	4	TBA

XIII. Test Report

A. Total test cases: 4
B. Results: TBA
C. Bugs Discovered: TBA