Neha Bangari, Vrishank Bangari, Jake Frohlich, Ethan Werner
Intermediate Software Design

Project Milestone 4

Process Deliverable III

Agile Process Model

Name	Jake	Ethan	Neha	Vrishank
What went well?	Tests were applicable.	Started early and finished with much time to spare	Tests were easy to determine based on previous assignments.	Making the tests was straightforward.
What didn't go well?	Creating the ideas for the tests was difficult.	Thinking of enough different tests	Determining the test types for some was difficult.	Sometimes the inputs and outputs would be confusing to make.
What will we change next time?	Think about how to approach testing of the system from a user point of view.	Not much. This time went pretty well.	Think about potential edge cases in the system more to come up with more robust cases.	Think through the app idea more thoroughly so the inputs/outputs are more clear.

Prioritized Task List for PM5

1. Create final presentation
    a. Explain the problem we are addressing
    b. Describe how our solution will solve this problem
    c. Compile related works
    d. List 3 or more related concepts from class that we used in this project
    e. Brainstorm future additions that could improve our solution even more down the line
    f. Compile the above info into a neat and organized presentation
    g. Practice presentation and assign speaking roles
2. Develop final report
    a. Update original abstract to address feedback given
    b. Update the proposal's introduction using feedback
    c. Develop a motivating example to demonstrate how and why our solution would be used
    d. Explain our solution's relevance to software engineering

   e. Determine if we need to include a background section to define terms and concepts and develop one if needed
   f. Write a section utilizing the related works compiled for the presentation
   g. Create a section to explain our design processes and decisions throughout the development of this project
   h. Develop and a deployment plan for our solution
   i. Discuss the possible future additions brainstormed previously
   j. Finish the report with a conclusion and a works cited section
   k. Revise and edit full report to perfect grammar, spelling, and cohesion

3. Individually complete retrospective surveys about the project

Black Box Tests:

Test ID	**Description**	**Expected Results**	**Actual Results**
Test Name: TestEvenDistribution **Test Author:** Ethan Werner **Test Type:** Functional	**Preconditions:** ● A list exists of tasks and meetings for the upcoming week **Steps:** ● Select to create a new schedule ● Enter each meeting scheduled for the week including g a title, day, time, and length of each meeting ● Enter each task that must be completed for the week including a title and estimated length of each task ● Submit all items and get optimized schedule ● Calculate the total time allocated for work each each day and check if the distribution is even	**Test Outputs:** ● The distribution of work across each day is within a hour range of each other	N/A

	Test Inputs: - A list of all weekly tasks and meetings for the week		
Test Name: Schedule Task at Specific Time **Test Author:** Vrishank Bangari **Test Type:** Functional	**Preconditions:** - Task scheduler application is running. - No tasks are currently scheduled. **Steps:** - Open the task scheduler interface. - Input task name as "Send Email." - Set the execution time to 10:00 AM. - Save the task. **Test Inputs:** - Task name: "Send Email" - Execution time: 10:00 AM	**Test Outputs:** - Task "Send Email" is successfully scheduled for 10:00 AM. - Task execution log confirms the task runs at 10:00 AM.	N/A
Test Name: Handle Overlapping Tasks **Test Author:** Vrishank Bangari **Test Type:** Stress	**Preconditions:** - Task scheduler has a task already scheduled at 10:00 AM. - The system allows multiple tasks to overlap. **Steps:** - Schedule Task 1 named "Send Email" at 10:00 AM. - Schedule Task 2 named "Backup Database" also at 10:00 AM. **Test Inputs:** - Task 1 name: "Send Email" - Task 2 name: "Backup Database" - Execution time for both: 10:00 AM	**Test Outputs:** - Both tasks are scheduled for 10:00 AM. - Task execution log shows both tasks run without errors.	N/A

Test Name: Task Cancellation **Test Author:** Vrishank Bangari **Test Type:** Functional	**Preconditions:** ● Task scheduler has a task named "Send Email" scheduled for 10:00 AM. **Steps:** ● Open the task scheduler interface. ● Locate the "Send Email" task in the list. ● Select the task and choose the "Cancel" option. **Test Inputs:** ● Task to cancel: "Send Email"	**Test Outputs:** ● Task "Send Email" is removed from the schedule. ● Task execution log confirms the task does not run at 10:00 AM.	N/A
Test Name: TestInsufficientTime **Test Author:** Neha Bangari **Test Type:** Functional	**Preconditions:** ● The total estimated time for a given task or set of tasks exceeds the available time in a week. **Steps:** ● Select to create a new schedule ● Add tasks to the schedule where the total estimated completion time exceeds available work hours for the week ● Submit the schedule, to be optimized. **Test Inputs:** ● All weekly tasks for the current iteration of a project, where total hours exceed allocated hours for the week	**Test Outputs:** ● System notifies the user that the total workload of all tasks exceeds the available time ● The scheduler system provides suggestions for prioritizing or reallocating tasks	N/A
Test Name: TestConfigureWorkHours **Test Author:** Neha Bangari	**Preconditions:** ● The user can configure daily working hours. **Steps:** ● Set custom work hours for a given week	**Test Outputs:** ● The schedule keeps the user's daily hours in mind when	N/A

Test Type: Functional	● Submit hours for approval by higher-ups ● After approval, add tasks to be configured based on daily hours **Test Inputs:** ● User's work hours for a week ● Tasks for the week	allocating tasks	
Test Name: TestTaskRescheduling **Test Author:** Neha Bangari **Test Type:** Functional	**Preconditions:** ● A previously created schedule exists **Steps:** ● Load an existing schedule ● Modify a task's estimated duration, priority level, or specifications ● Resubmit the schedule to be optimized again ● Seek approval based on pending deadlines or requirements **Test Inputs:** ● Existing schedule ● Task with updated duration or specifications	**Test Outputs:** ● The system adjusts the schedule accordingly, while still maintaining an even distribution of work, if the task change is approved	N/A
Test Name: TestTaskEdit **Test Author:** Ethan Werner **Test Type:** Functional	**Preconditions:** ● A previously created schedule exists **Steps:** ● Load the existing schedule ● Choose a task to edit ● Select edit ● Change the name and description of the task **Test Inputs:** ● Task to edit ● New name and description	**Test Outputs:** ● Updated task with updated information	N/A

Test Name: TestUpdateTask **Test Author:** Jake Frohlich **Test Type:** Functional	**Preconditions:** ● A previously created schedule exists and tasks are loaded into the system ● User has access and can edit a user edit button **Steps:** ● Load the schedule ● Click on the "make an update tab" ● Modify a task (delete, reallocate estimated times it would take to complete) ● Reload UI to see if modifications were made appropriately **Test Inputs:** ● Current tasks in the schedule as a list ● A flag to indicate deletion or reallocation of estimated time for a given task ● The task to modify	**Test Outputs:** ● Updated UI to reflect changes. List of schedule should show a deletion or a new allocation of estimated time	N/A
Test Name: TestModifyColors **Test Author:** Jake Frohlich **Test Type:** Functional	**Preconditions:** ● UI object instantiated ● UI is defaulted to white background color **Steps:** ● Load the task scheduler ● Click the button on the main screen which swaps from light to dark mode **Test Inputs:** ● Color mode button	**Test Outputs:** ● Correct assertion if the background color RGB is (0,0,0), false otherwise.	N/A