Ben Molloy (Individual Project)

Student Information

- molloyb1@mymail.nku.edu (mailto:molloyb1@mymail.nku.edu)
- https://github.com/btmolloy/ASE456-Individual-Project (https://github.com/btmolloy/ASE456-Individual-Project)

Planning the Individual Project

Summary of the Project

This project is to develop a personal time management system using Flutter and Firebase for a user. The system will allow the user to record and track their time using flexible date and time formats. Additionally, the user will be able to query the recorded data based on date, task, or tag.

Goals

First Part Goal:
Develop a user-friendly graphical user interface (GUI) where the user can easily record and track their time.

Second Part Goals:

- Implement a flexible input system where the user can record time using various date and time formats (e.g., AM/PM and "YYYY/MM/DD" formats).
- Enable the user to view the five most imminent tasks prominently on the home page.
- Allow users to query their time usage based on specific criteria, including date, task name, or tag.
- Develop a feature for generating reports that summarize the user's time allocation by tags and activities.
- Add a time usage analytics page that visually represents the total time spent per tag in a bar graph for easy analysis.

Why?

The ability to record, track, query, and analyze time usage is crucial for effective time management. Key features like the "Upcoming Tasks" section on the home screen and the time usage analytics page provide a clear snapshot of current and past activities. Querying and reporting functionalit enable users to filter and analyze their data, while the visual representation in analytics helps users make informed decisions about improving their time allocation.

Features

- Task Recording

  - As a user, I want to input data in the format DATE, FROM, TO, TASK, TAG (e.g., today 09:30 10:30 studied Java STUDY) to create a new record and save it to the database.

- Task Display and Prioritization

  - As a user, I want to see the top five upcoming tasks on the home screen, so I can quickly identify what’s next.
  - As a user, I want the upcoming tasks to be organized by start time, ensuring the soonest tasks appear first.

- Task Querying

  - As a user, I want to input a specific date and see all activities that occurred on that day.
  - As a user, I want to input a task name and retrieve all activities related to that task.
  - As a user, I want to input a tag and see all activities associated with that tag.
  - As a user, I want to query tasks by a start and end date to view all activities that fall within that range.

- Time Reporting

  - As a student, I want to generate a report of my time usage for a defined time period.
  - As a student, I want to review the tasks I spend the most time on, organized by tag, so that I can better understand how I allocate my time.

- Time Usage Analytics

  - As a user, I want to see a graphical representation of how much time I’ve spent on different tags in a bar graph, with each bar representing a tag and its corresponding time.
  - As a user, I want the bar graph to use different colors for each tag for better visualization.

Test Plans:

Objective

Verify the complete task workflow, from adding a task to querying and generating reports.

Manual Unit Test Plans

1. Confirm application can launch

Expected: Task is added and confirmed.

2. Add Task

Add a task with:
Date: 2024-11-11
Time: 10:00 AM - 11:00 AM
Name: Manual Test Task
Tag: Test Tag
Expected: Task is added and confirmed.

3. Query by Date

Search for tasks on 2024-11-11.
Expected: Manual Test Task appears.

4. Query by Name

Search for Manual Test Task.
Expected: Task details are accurate.

5. Query by Tag

Search for Test Tag.
Expected: Task is displayed.

6. Generate Report

Generate a time usage report.
Expected: Task duration (1 hour) appears correctly.

7. View Analytics Page with time usages.

View the Analytics Page.
Expected: Analytics Page gets populated with a visual representation of the users time usage.

8. View Upcoming Tasks.

View the Upcoming Tasks Box.
Expected: On launch of the application, the 5 closest upcoming tasks gets populated.

9. Query by Time Period

Search for Tasks between 2024/11/11 and 2024/12/12.
Expected: Test Task is displayed.

Manual Integration Test Plans

- Add a task, query it by name, and generate a time usage report.

Expected: The task is successfully added, retrieved accurately through queries, and included in the generated report.

Manual Acceptance Test Plans

- Launch the application, add a task, and validate it appears on the home screen and analytics page.

Expected: The app launches without errors, the task is added successfully, and analytics are updated with the new task.

Regression Test

- Test adding a task, querying by date, and generating a report to ensure existing features are unaffected by recent changes.

Expected: All tested features work as expected, with no errors or inconsistencies observed.

Milestones and Deadline

Deadline 1:
Date: 20 OCT 2024
Goal: Complete initial prototype of the time recording and querying system.

> **Milestone 1:**
> **Date:** 30 OCT 2024
> **Goal:** Implement the query feature and begin testing for time recording and querying functionality.

Deadline 2:
Date: 10 NOV 2024
Goal: Complete Version 1.0, including full implementation of time tracking, querying, and report generation. Begin testing.

> **Milestone 2:**
> **Date:** 20 NOV 2024
> **Goal:** Create Ver2 by refactoring code for feature improvements, clean up code, and start writing documentation. Add new requirments.
>
> **Milestone 3:**
> **Date:** 25 NOV 2024
> **Goal:** Complete all testing. Finalize documentation and prepare for final submission.

Deadline 3:
Date: 02 DEC 2024
Goal: Submit the completed project, including all code, tests, and documentation.

Risk Analysis

1. **ROTC Class:** Every Thursday from 12:00 to 18:00
2. **PT/Class:** Monday, Wednesday, and Friday mornings
3. **ROTC BRM:** All day on 10/26
4. **NG Drill:** Nov 2-3
5. **Business Trip:** Nov 15-18
6. **ASE 456 Final Examination:** Sometime after Dec 1
7. **Server Team Analyst Work:** Monday, Tuesday, Wednesday, and Friday every week

Project Progress

Feature Implementation

Week 1 (10/6 - 10/12)

Lines of Code:
Loc: 0

Milestones or risks in this week:

N/A

Summary:

Canvas Page setup - Completed 10/10

Week 2 (10/13 - 10/19)

NO PROGRESS

Week 3 (10/20 - 10/26)

Lines of Code:
Loc: 489

Milestones or risks in this week:

Deadline 1: Completed (Complete initial prototype)

Summary:

This week I focused on completing the initial prototype with the goal of having it connect to firebase properly.

Week 4 (10/27 - 11/2)

Lines of Code:
Loc: 211

Milestones or risks in this week:

Milestone 1: Completed (implement query feature/testing)

Summary:

This week I focused on completing first milestone to have the application be able to query data. Working on this marked the start of working on Version 1 of the application. The feature was added successfully and testing was completed.

Week 5 (11/3 - 11/9)

NO PROGRESS

Week 6 (11/10 - 11/16)

Lines of Code:
Loc: 172

Milestones or risks in this week:

Deadline 2: Completed (Complete Version 1.0)

Summary:

This week I completed Version 1.0 by added more features and functionality. Specifically, I added full implementation of time tracking, querying and report generation. I also began testing this version of the application to make sure it functioned properly. This week I completed the second deadline.

Week 7 (11/17 - 11/23)

Lines of Code:
Loc: 293

Milestones or risks in this week:

Milestone 2: Completed (Refactor code implementing the new requirements for Ver2)

Summary:

This week I completed Version 2.0 by added more features and functionality requested by the client. Specifically, I added more query filters and I also added a new page that helps the user visualize their time usage. I also cleaned up my code a bit and started working on the documentation.

Week 8 (11/24 - 12/2)

Lines of Code:

Loc: 24

Milestones or risks in this week:

Milestone 3: Completed (Completed all testing and documentation)

Deadline 3: Completed (Fully complete the final version)

Summary:

This week I completed 1 deadline and 1 milestone. This includes finishing the rest of the final version and completing the rest of the tests and documentation. I also made some tweaks to the final version, cleaning up some code by refactoring a bit and fixed a bug or two. This week I officially finished the project.

All of my code for my features have a total of 1189 lines of code.