

Orbit: Personal Well-being Web Application

Fundamentals of Software Engineering

Deliverable 4: Sprint 3 (Iteration 3)

Submission Date: 26th April 2025

Project Team

Dania Waseem 23I-2622

Munaza Tariq 23I-2545



Department of Data Science

**National University of Computer and Emerging Sciences
Islamabad, Pakistan**

,

Contents

0.1	Introduction	1
0.2	Sprint 3 Backlog	1
0.2.1	User Stories for Sprint 3	1
0.3	All User Stories	3
0.4	Scrum Board Snapshots for Sprint 3	5
0.5	Scrum Board Snapshots Overall	8
0.6	Design	10
0.7	Architecture Pattern	18
0.7.1	Architecture Diagram	18
0.8	Implementation	19
0.9	Product Burndown Chart	27
0.9.1	Burndown Chart Description	27
0.9.2	Burndown Chart	27
0.10	Boundary Value Analysis Testing of the Sign-up and Login Interfaces . .	28
0.10.1	Boundary Value Analysis for Sign-up Form	28
0.10.2	Boundary Value Analysis for Login Form	29
0.11	Work Division	30
0.12	Learning from this Project	32

List of Figures

1	Snapshot 1: Sprint Backlog	5
2	Snapshot 2: Mid-Sprint Progress (Half of User Stories Completed)	6
3	Snapshot 3: Sprint Completion	7
4	Sprint 2 Backlog	8
5	Sprint 2 Backlog	8
6	Sprint 3 Backlog	9
7	Activity Diagram for To Do List	11
8	Activity Diagram for Meditation Page	12
9	Activity Diagram for Monthly Overview	13
10	Use Case Diagram for the Entire System	14
11	Sequence Diagram for To Do List Page	15
12	Sequence Diagram for Journal Page	16
13	Sequence Diagram for Monthly Overview Page	16
14	Class Diagram for System Architecture	17
15	Model-View-Controller (MVC) Architecture Diagram	18
16	Login Page	19
17	Sign up Page	20
18	Home Page	21
19	Meditation	22
20	To do list	23
21	Habit tracker	24
22	Mood tracker	25
23	Journal your thoughts	26
24	Product Burndown Chart for the Project	27

List of Tables

1	Boundary Value Analysis for Password Length in Sign-up Form	28
2	Boundary Value Analysis for Username in Sign-up Form	29
3	Boundary Value Analysis for Login Form	29
4	Character Requirement Testing for Password	30
5	Work Division	31

Orbit: Personal Wellbeing Application

0.1 Introduction

Orbit is a personal well-being desktop application made using C and Windows Forms. It also uses a local SQL database to save the user's data safely. The main idea of this project is to help users manage their daily activities, emotions, and habits easily. In daily busy life, people often forget to organize their tasks, relax their mind, and keep track of good habits. Orbit brings all these important features together in one simple and user-friendly app.

The application offers a Homepage with easy navigation, a To-Do List where users can add, edit, and complete daily tasks, a Journal to write and save their thoughts, and a Meditation section. The Meditation page has a timer for guided meditation and plays soft music to help users feel relaxed and focused. In addition, there is a Habit Tracker to record daily good habits like drinking water, exercising, and reading, and a Mood Tracker to record how the user feels each day.

Orbit is fully functional, meaning users can interact with all features easily — they can add new entries, update old ones, and delete anything when needed. The design uses soft colors like pink shades and large, clear buttons to make it easy for everyone to use. The application works offline after installation and does not need an internet connection. Orbit is built to improve productivity, encourage mindfulness, and help users maintain a balanced emotional life through a simple and smooth desktop experience

0.2 Sprint 3 Backlog

Module for Sprint 3: Meditation, Journal Your Thoughts, Mood Tracker

0.2.1 User Stories for Sprint 3

- **User Story 1: Meditation Page**

- As a *user*, I want to *listen to calming meditation music* so that I can *relax and reduce stress*.

Sub-User Stories:

- As a *user*, I want to *start and stop a meditation timer* so that I can *manage my meditation session easily*.
- As a *user*, I want to *save completed meditation sessions into the database* so that I can *track my mindfulness progress*.

- **User Story 2: Journal Your Thoughts Page**

- As a *user*, I want to *write and save my daily thoughts* so that I can *reflect on my emotions and experiences*.

- Sub-User Stories:**

- As a *user*, I want to *edit my past journal entries* so that I can *update my reflections later*.
- As a *user*, I want to *save journal entries in the database* so that I can *retrieve them anytime*.

- **User Story 3: Mood Tracker Page**

- As a *user*, I want to *select and record my mood* so that I can *analyze my emotional patterns over time*.

- Sub-User Stories:**

- As a *user*, I want to *choose my mood through a slider or emojis* so that I can *express how I feel easily*.
- As a *user*, I want to *save my mood with the date into the database* so that I can *track my mental wellbeing history*.

- **User Story 4: Login Page (Database Integration)**

- As a *registered user*, I want to *log in using my saved credentials from the database* so that I can *securely access my data*.

- Sub-User Stories:**

- As a *user*, I want to *get an error if login credentials don't match database records* so that I can *fix mistakes*.

- **User Story 5: Signup Page (Database Integration)**

- As a *new user*, I want to *register and store my information into the database* so that I can *create my account successfully*.

- Sub-User Stories:**

- As a *user*, I want to *validate that username is unique in the database* so that I can *avoid duplicate registrations*.

0.3 All User Stories

- **User Story: Login**

- As a *registered user*, I want to *log in securely using my username and password* so that I can *access my productivity and wellness data*.

Sub-User Stories:

- As a *user*, I want to *enter my username and password* so that I can *authenticate myself*.
- As a *user*, I want to *see an error message if my credentials are wrong* so that I can *correct them*.

- **User Story: Signup**

- As a *new user*, I want to *create an account* so that I can *save my tasks and wellbeing data*.

Sub-User Stories:

- As a *user*, I want to *enter my name, username, and password* to *register*.
- As a *user*, I want to *see a confirmation message after registration* so that I know my account is created.

- **User Story: Homepage**

- As a *user*, I want to *view a welcoming homepage* so that I can *understand the app's features easily*.

Sub-User Stories:

- As a *user*, I want to *see a navigation bar* so that I can *easily switch between sections*.
- As a *user*, I want to *read a short app description* so that I can *know its purpose*.

- **User Story: To-Do List**

- As a *user*, I want to *add tasks to a to-do list* so that I can *stay organized every day*.

Sub-User Stories:

- As a *user*, I want to *mark tasks as completed* so that I can *track my progress*.
- As a *user*, I want to *delete tasks* so that I can *remove unnecessary items*.

- **User Story: Meditation**

- As a *user*, I want to *start meditation sessions with a timer and music* so that I can *relax and de-stress*.

- Sub-User Stories:**

- As a *user*, I want to *set a meditation timer* so that I can *practice for a fixed time*.
- As a *user*, I want to *play relaxing music during meditation* to *enhance calmness*.

- **User Story: Journal Your Thoughts**

- As a *user*, I want to *write and save my thoughts daily* so that I can *reflect on my feelings*.

- Sub-User Stories:**

- As a *user*, I want to *edit previous journal entries* so that I can *update them later*.
- As a *user*, I want to *save journal entries to database* to *review them later*.

- **User Story: Habit Tracker**

- As a *user*, I want to *add daily habits* like "drink water" or "exercise" so that I can *track my routine*.

- Sub-User Stories:**

- As a *user*, I want to *mark habits as done* so that I can *monitor consistency*.
- As a *user*, I want to *view my weekly habit record* to *see my growth*.

- **User Story: Mood Tracker**

- As a *user*, I want to *select my mood daily using emojis or slider* so that I can *track emotional health*.

- Sub-User Stories:**

- As a *user*, I want to *save my mood with date* so that I can *view mood history*.
- As a *user*, I want to *see previous moods on calendar* to *analyze emotional trends*.

0.4 Scrum Board Snapshots for Sprint 3

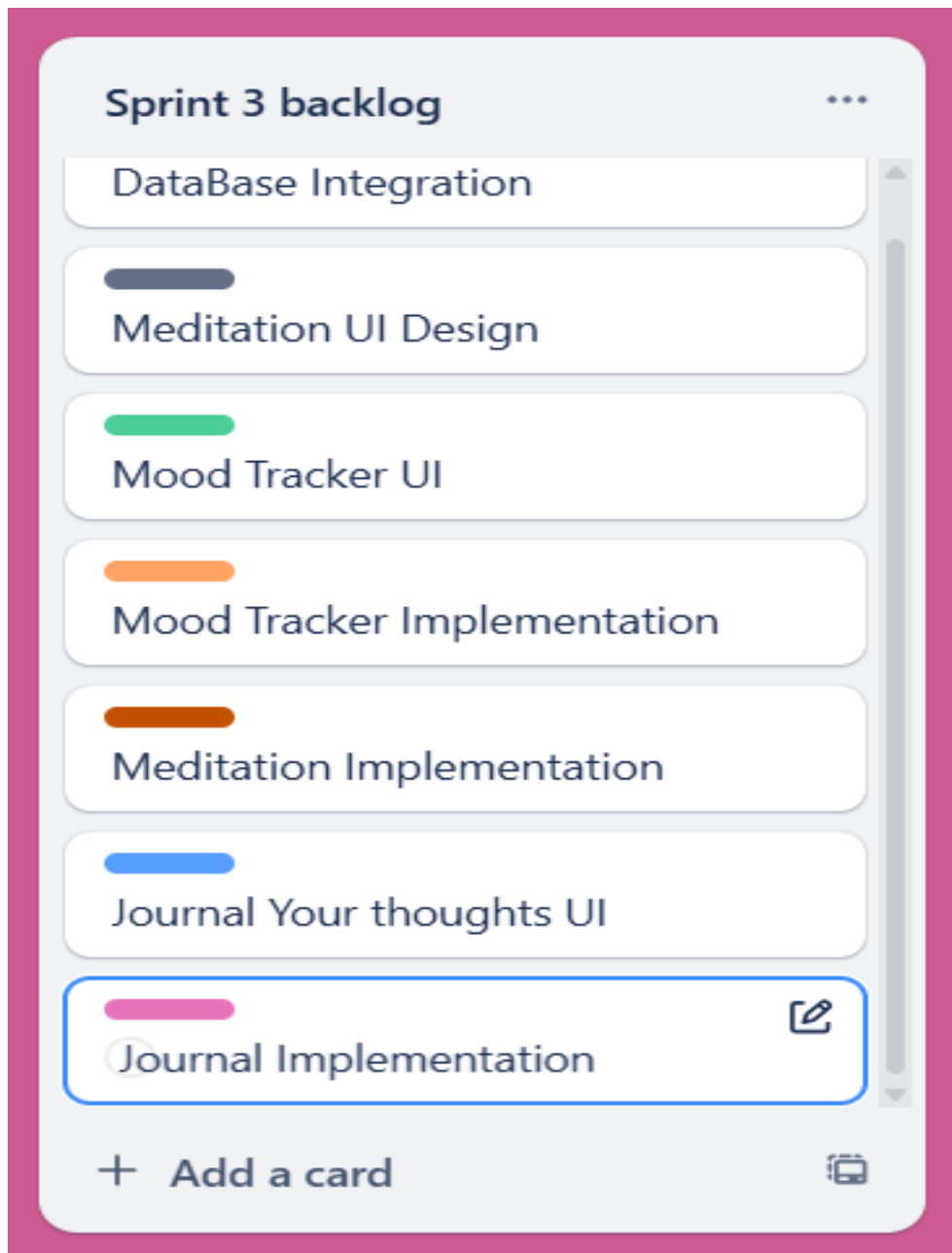


Figure 1: Snapshot 1:Sprint Backlog

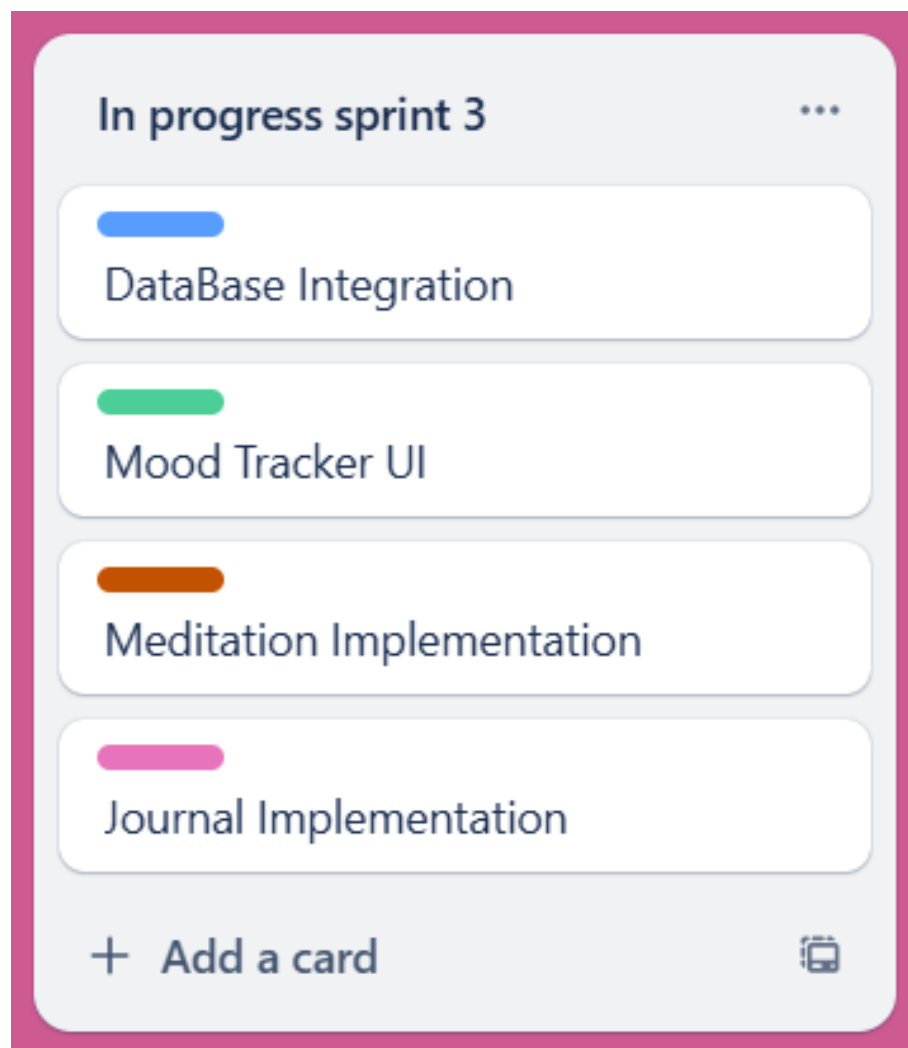


Figure 2: Snapshot 2: Mid-Sprint Progress (Half of User Stories Completed)

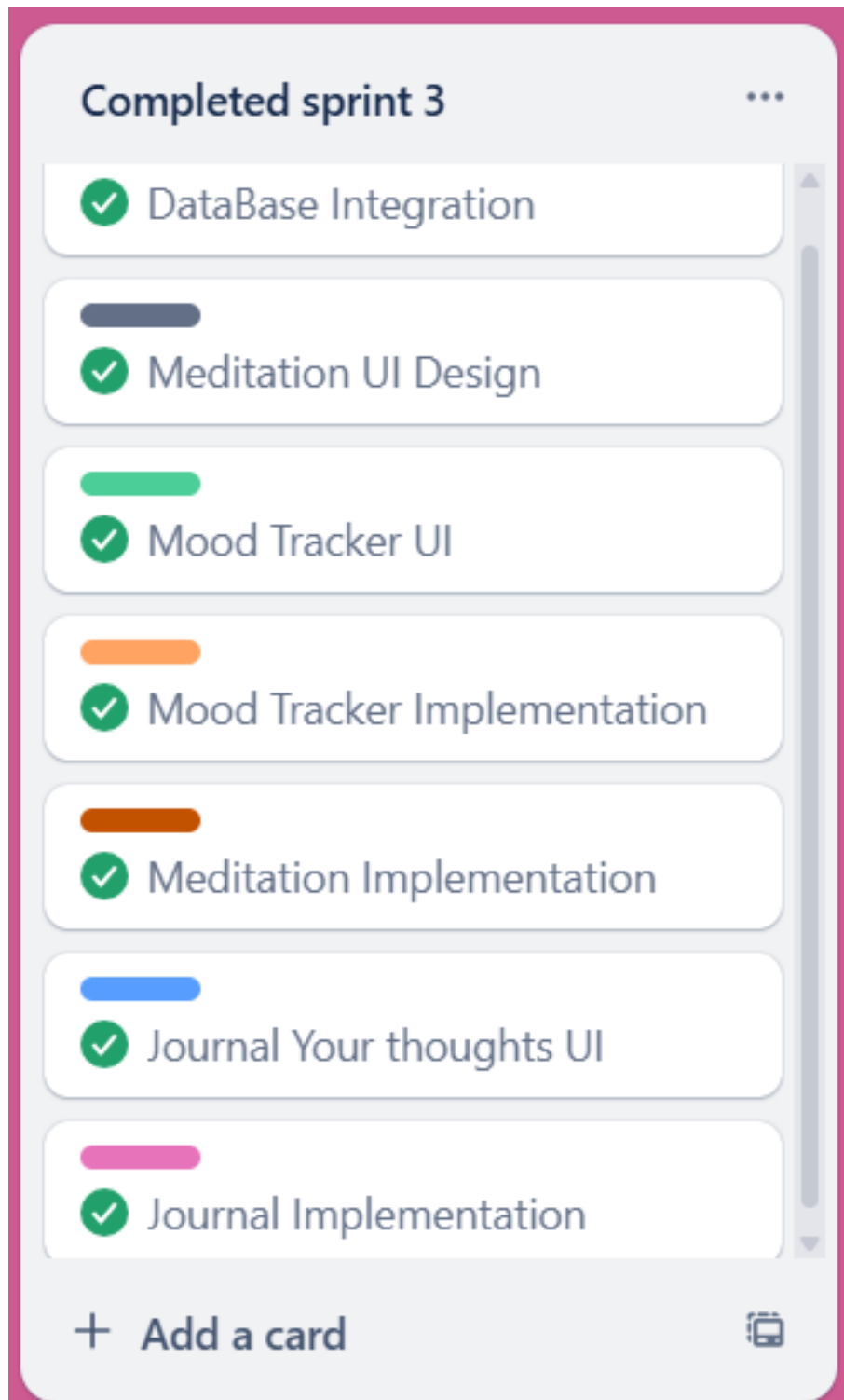


Figure 3: Snapshot 3: Sprint Completion

0.5 Scrum Board Snapshots Overall

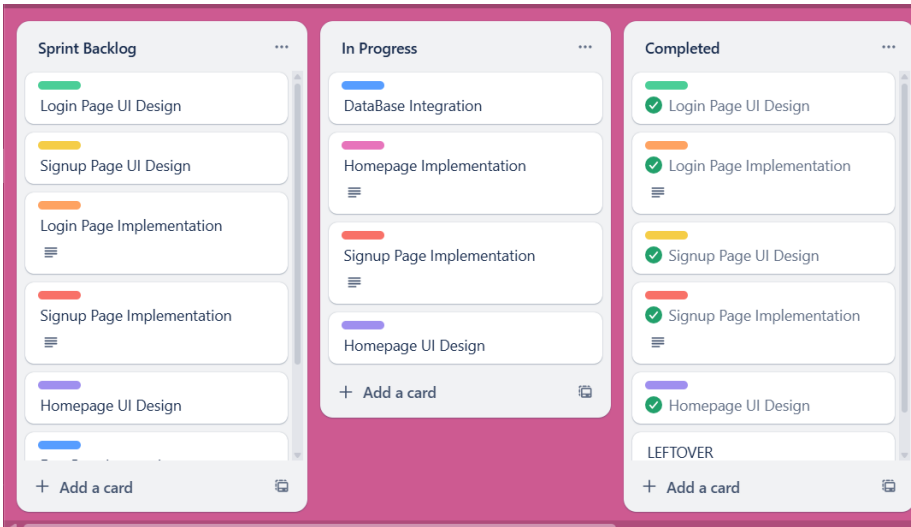


Figure 4: Sprint 2 Backlog

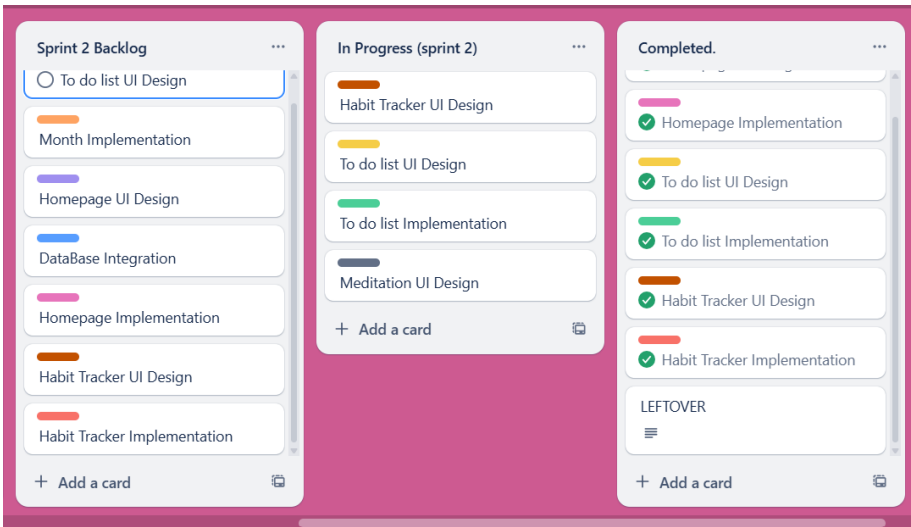


Figure 5: Sprint 2 Backlog

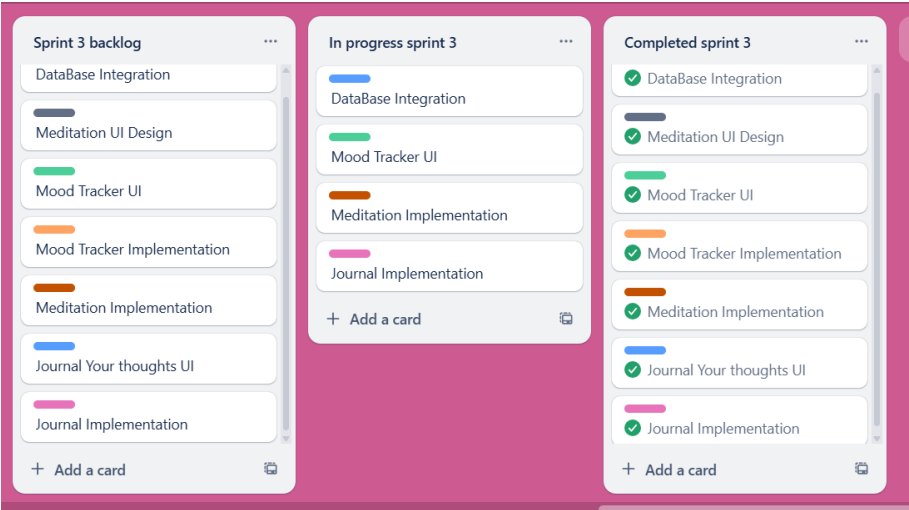


Figure 6: Sprint 3 Backlog

0.6 Design

In this section, we present the design components of our productivity and wellness application, which is structured around the main modules highlighted in the user stories: the To-Do List, Journal, and Monthly Overview. These designs help visualize the system's structure, data flow, interactions, and user engagement. The following diagrams are included:

Diagram Type	Description
Activity Diagrams	Diagrams illustrating the flow of control for the three main processes of the system: To-Do List, Journal, and Monthly Overview.
Use Case Diagram	A high-level representation of the system's functionality by showing interactions between users (actors) and the system's use cases.
Sequence Diagrams	Detailed interaction sequences for each major activity (To-Do List, Journal, Monthly Overview), demonstrating the flow of messages between objects.
Class Diagram	Structural blueprint of the system, displaying classes, their attributes and methods, and the relationships among them.

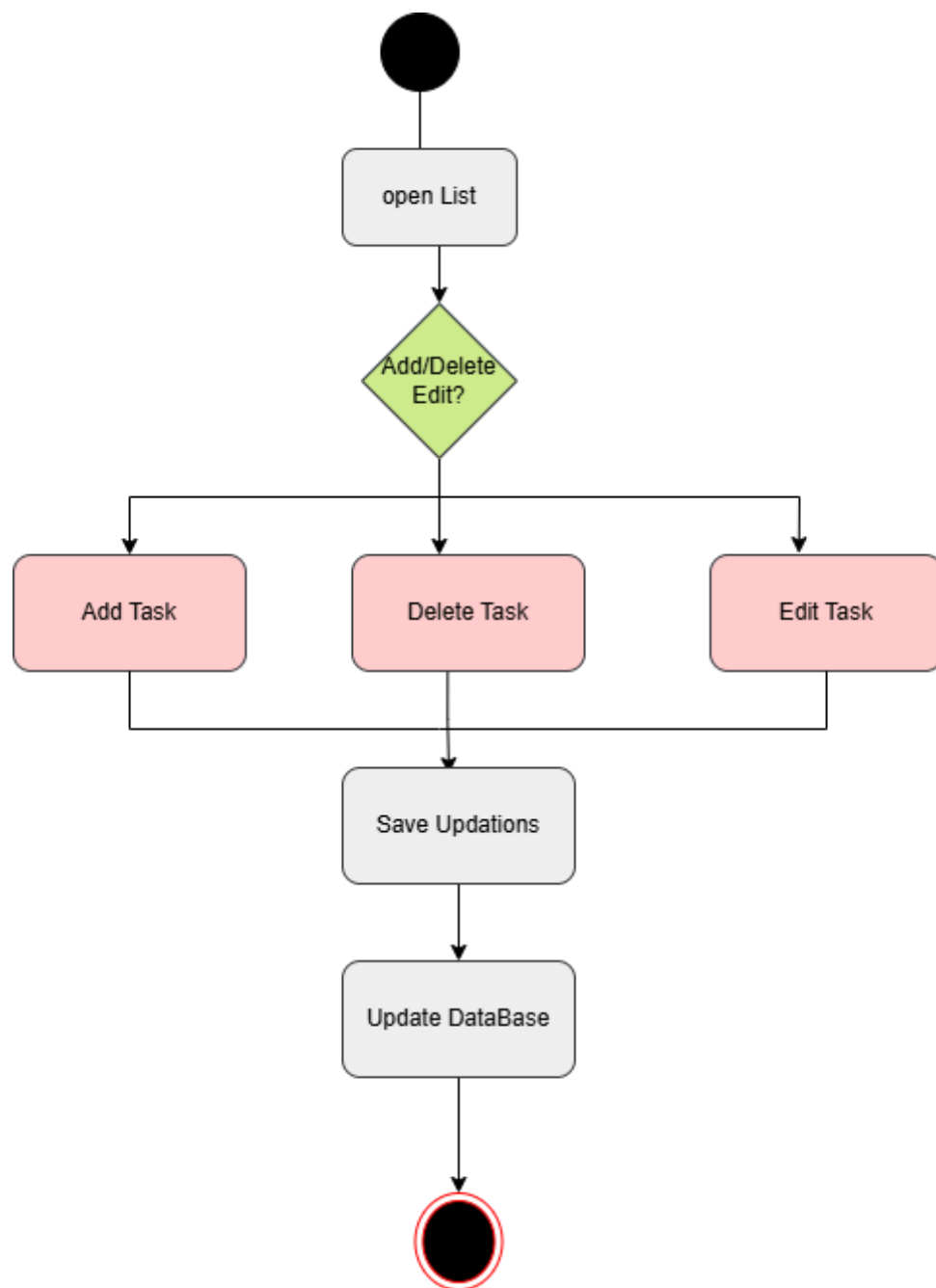


Figure 7: Activity Diagram for To Do List

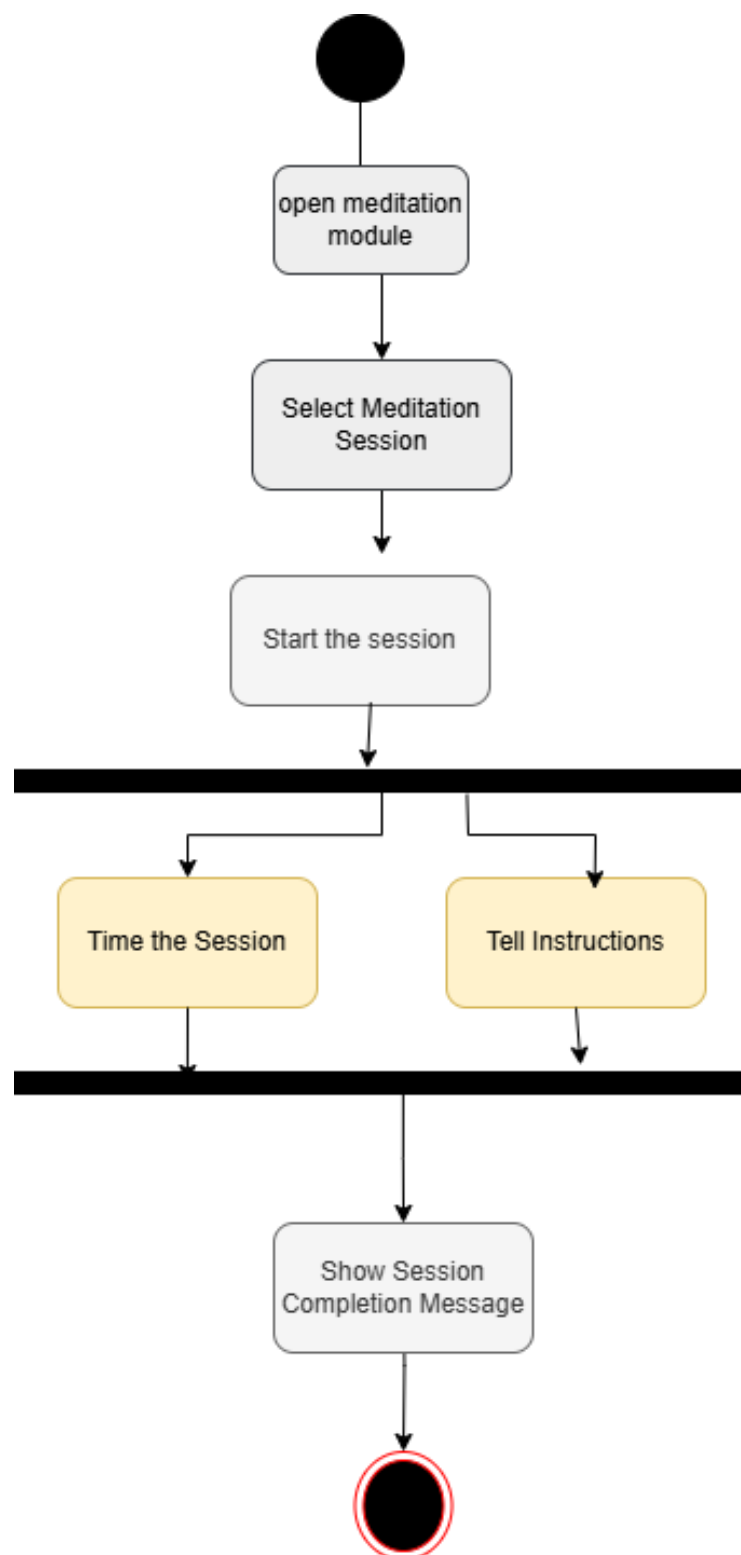


Figure 8: Activity Diagram for Meditation Page

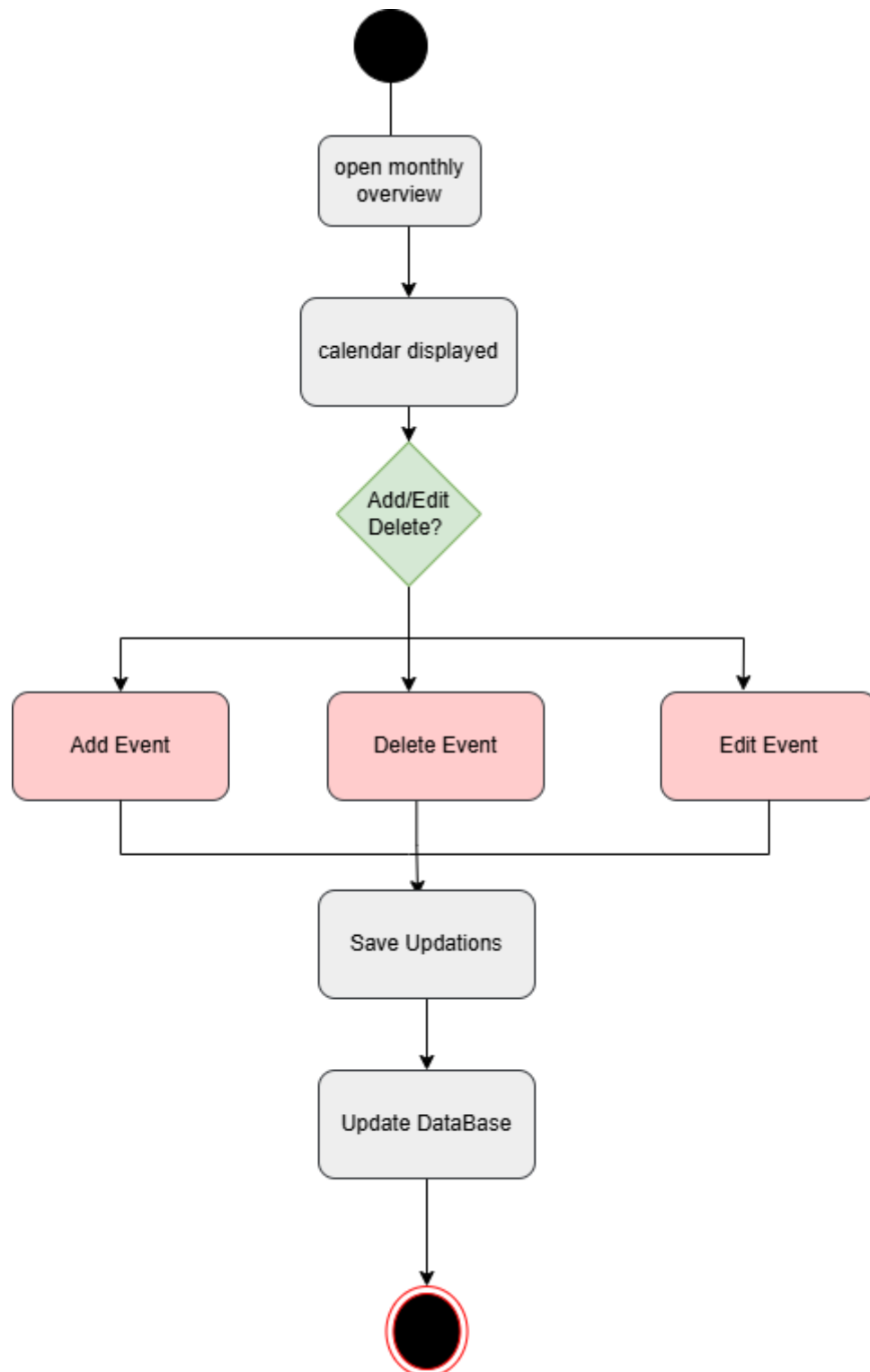


Figure 9: Activity Diagram for Monthly Overview

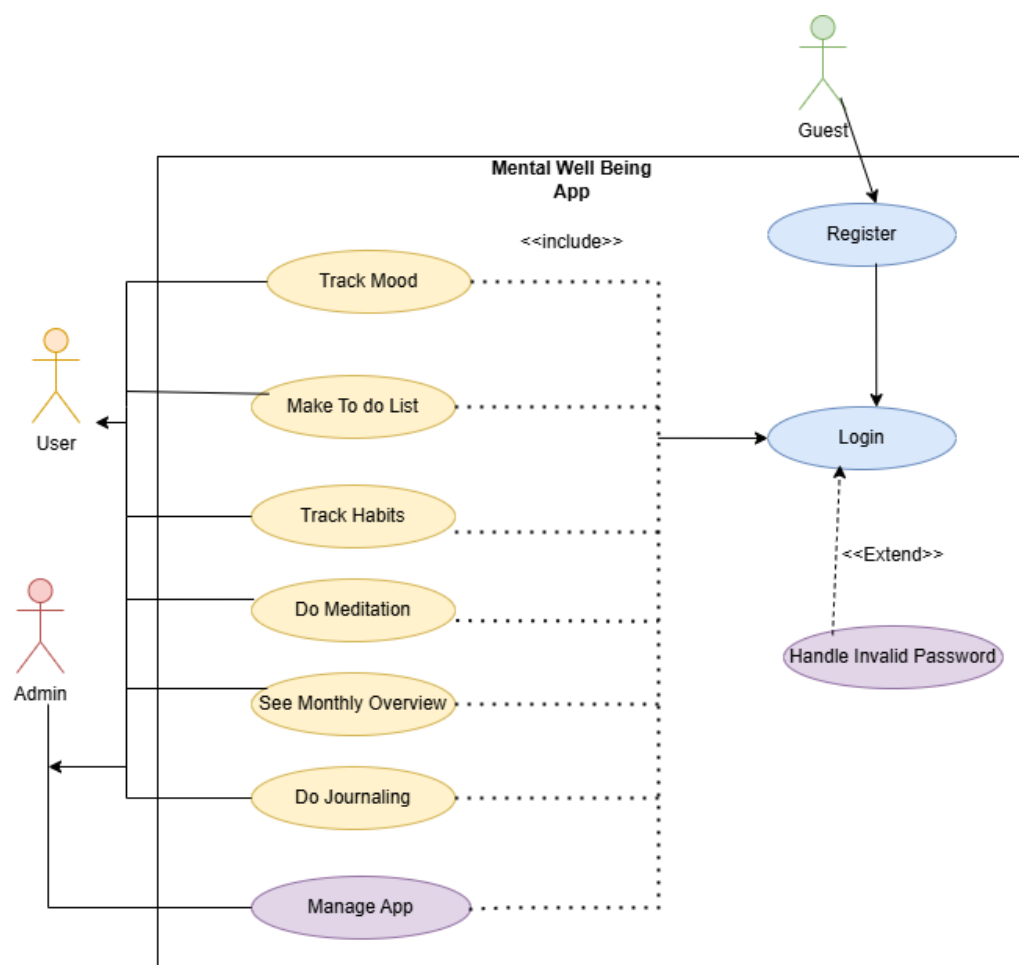


Figure 10: Use Case Diagram for the Entire System

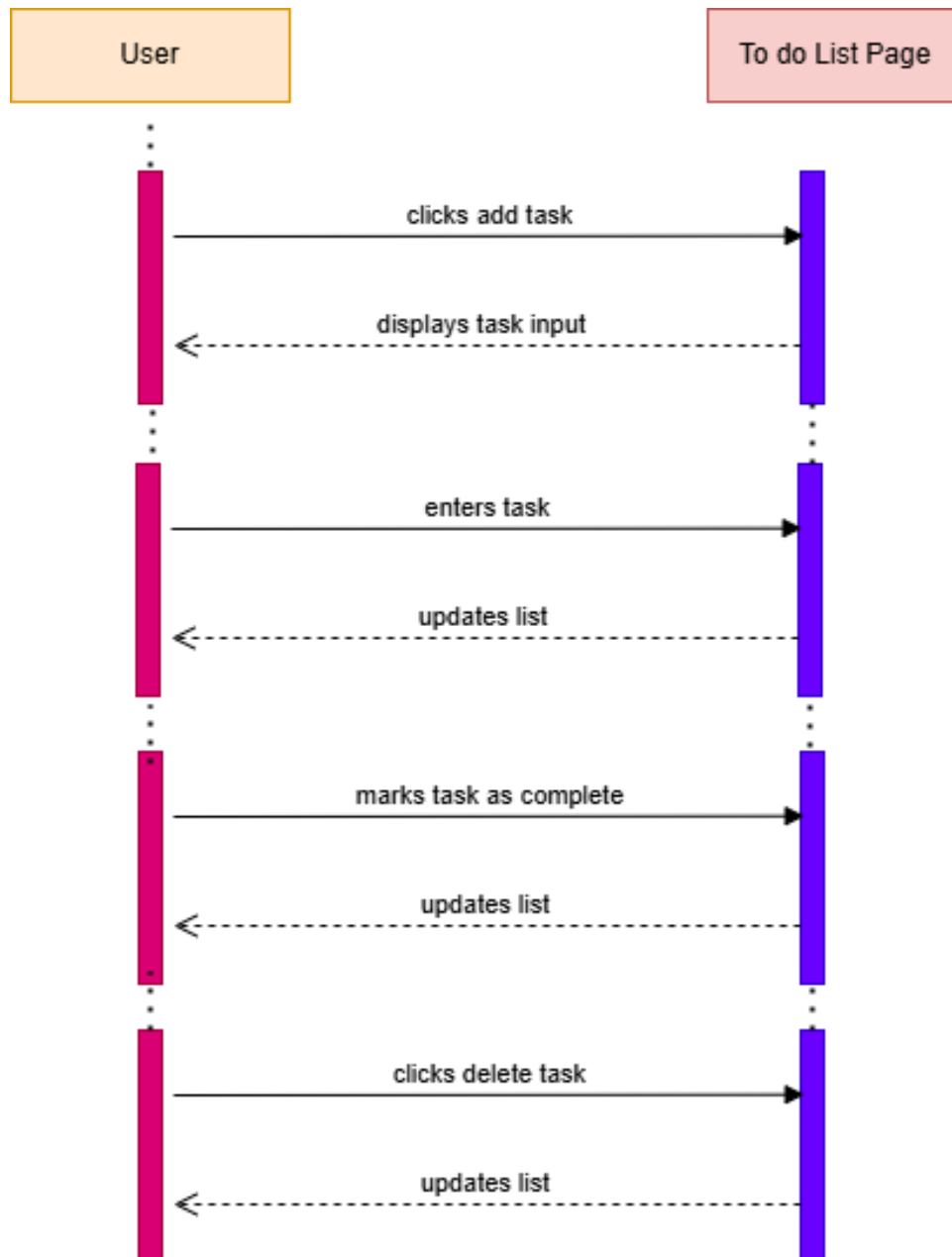


Figure 11: Sequence Diagram for To Do List Page

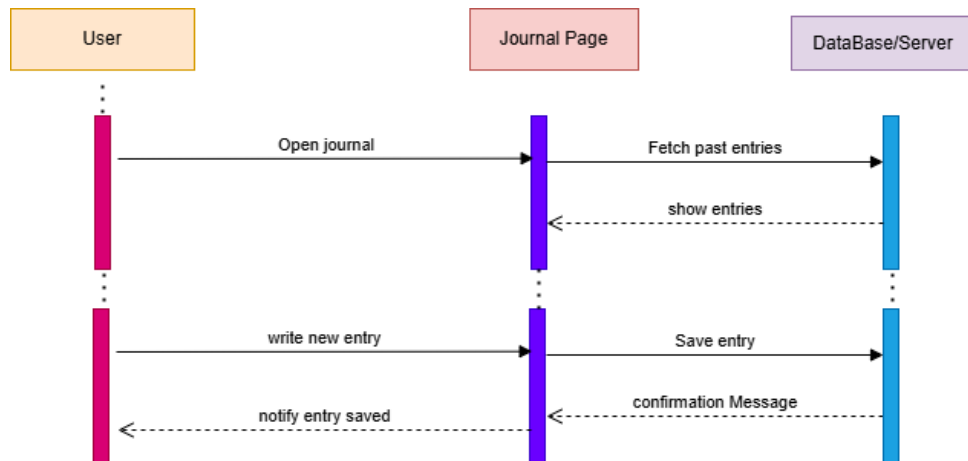


Figure 12: Sequence Diagram for Journal Page

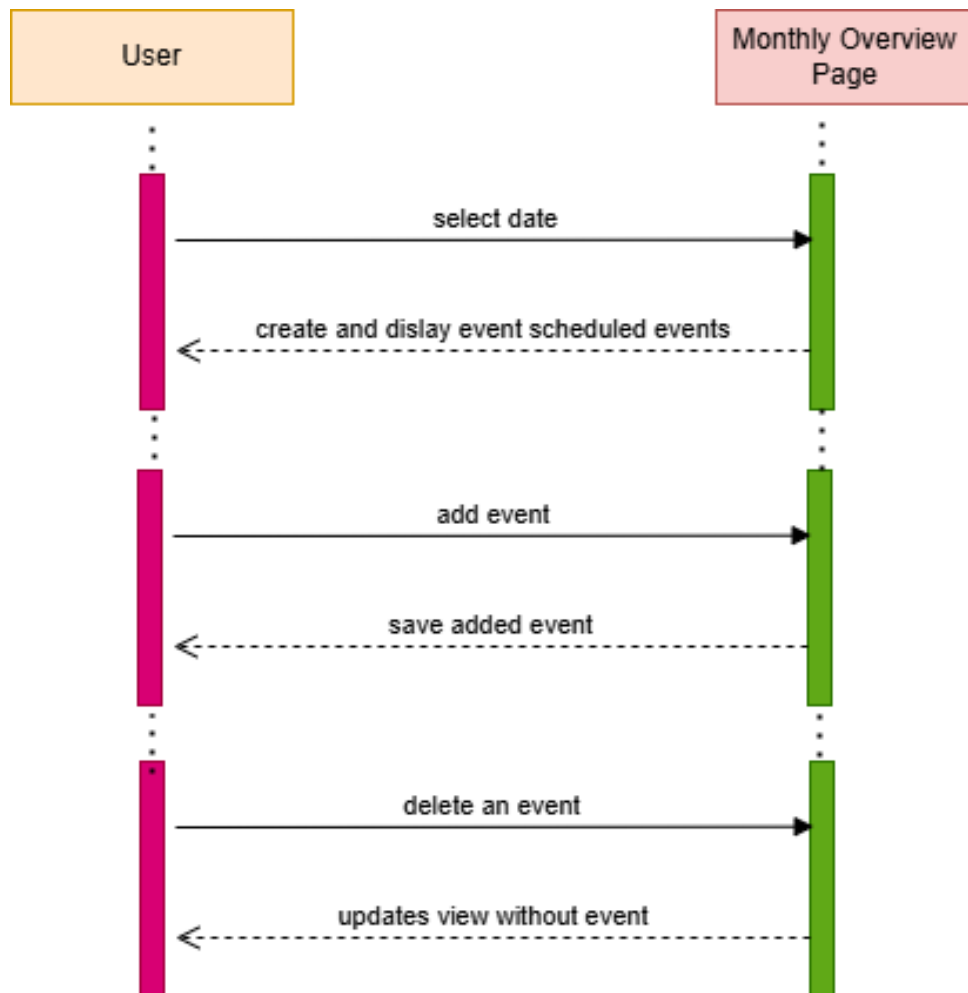


Figure 13: Sequence Diagram for Monthly Overview Page

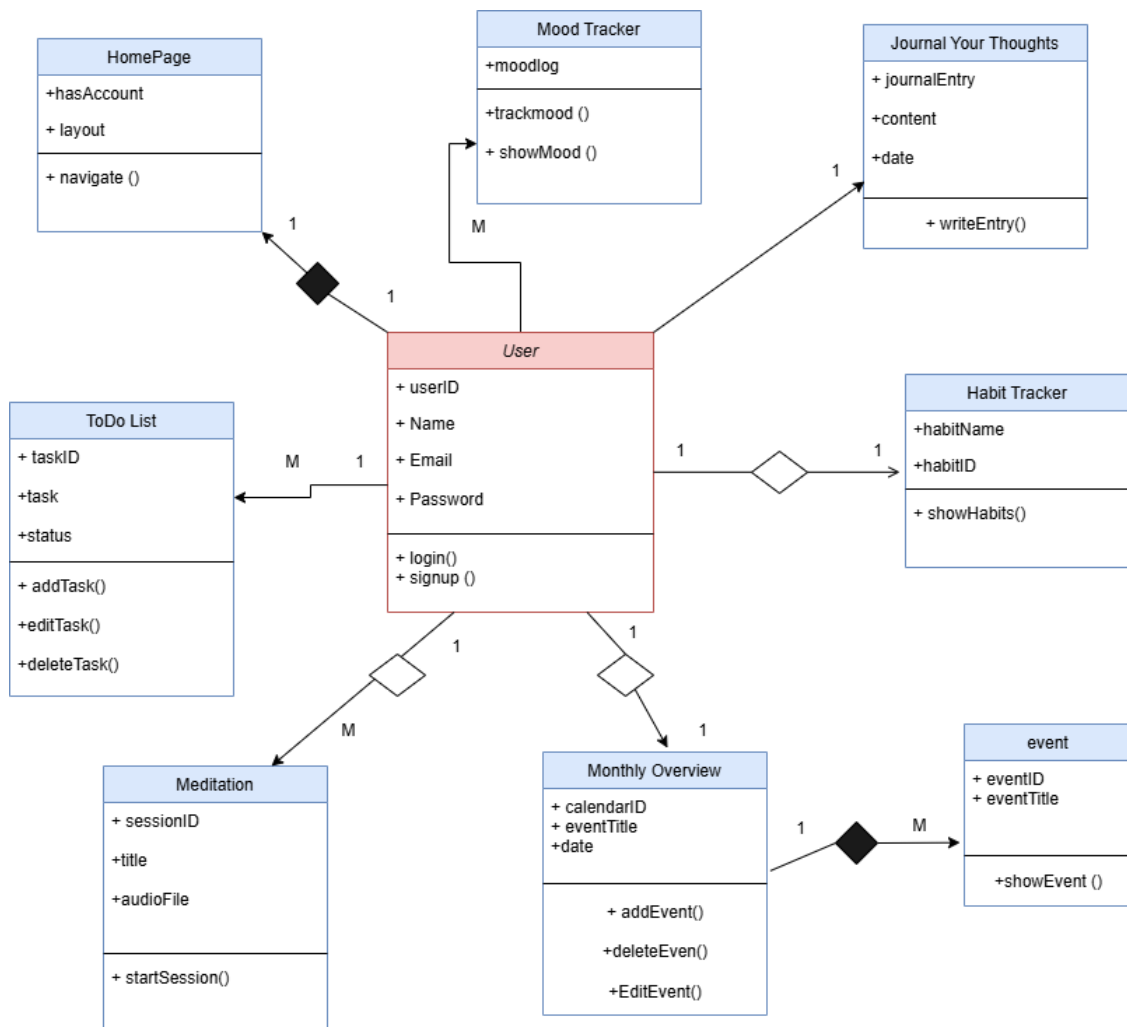


Figure 14: Class Diagram for System Architecture

0.7 Architecture Pattern

For our project, we have selected the **Model-View-Controller (MVC)** architecture pattern.

- **Model:** Contains the business logic and handles communication with the local database. It manages the data operations like inserting, updating, retrieving, and deleting records.
- **View:** Consists of the Windows Forms (WinForms) that provide the graphical user interface (GUI) for the user to interact with the system.
- **Controller:** Includes the event handlers and logic that link the View and Model. It responds to user interactions, invokes model updates, and refreshes the UI accordingly.

This pattern ensures a clear separation of concerns, making the application easier to manage, scale, and test.

0.7.1 Architecture Diagram

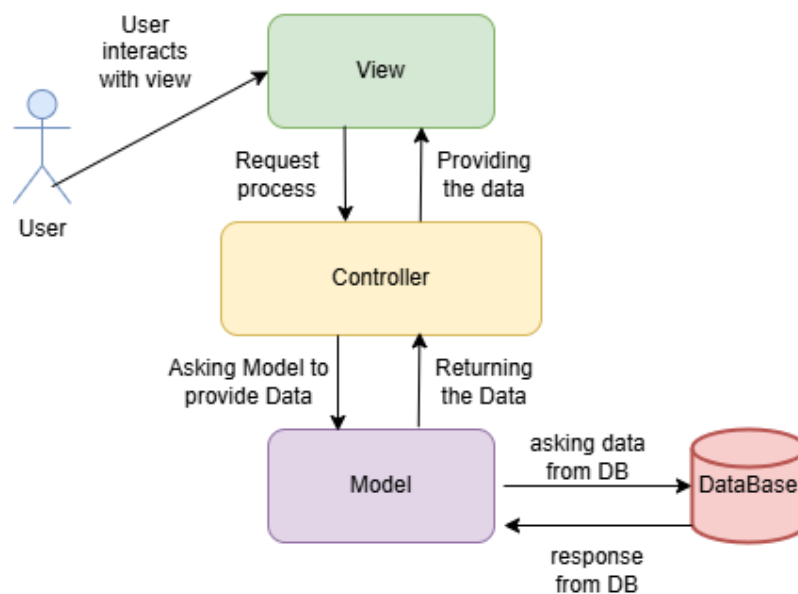
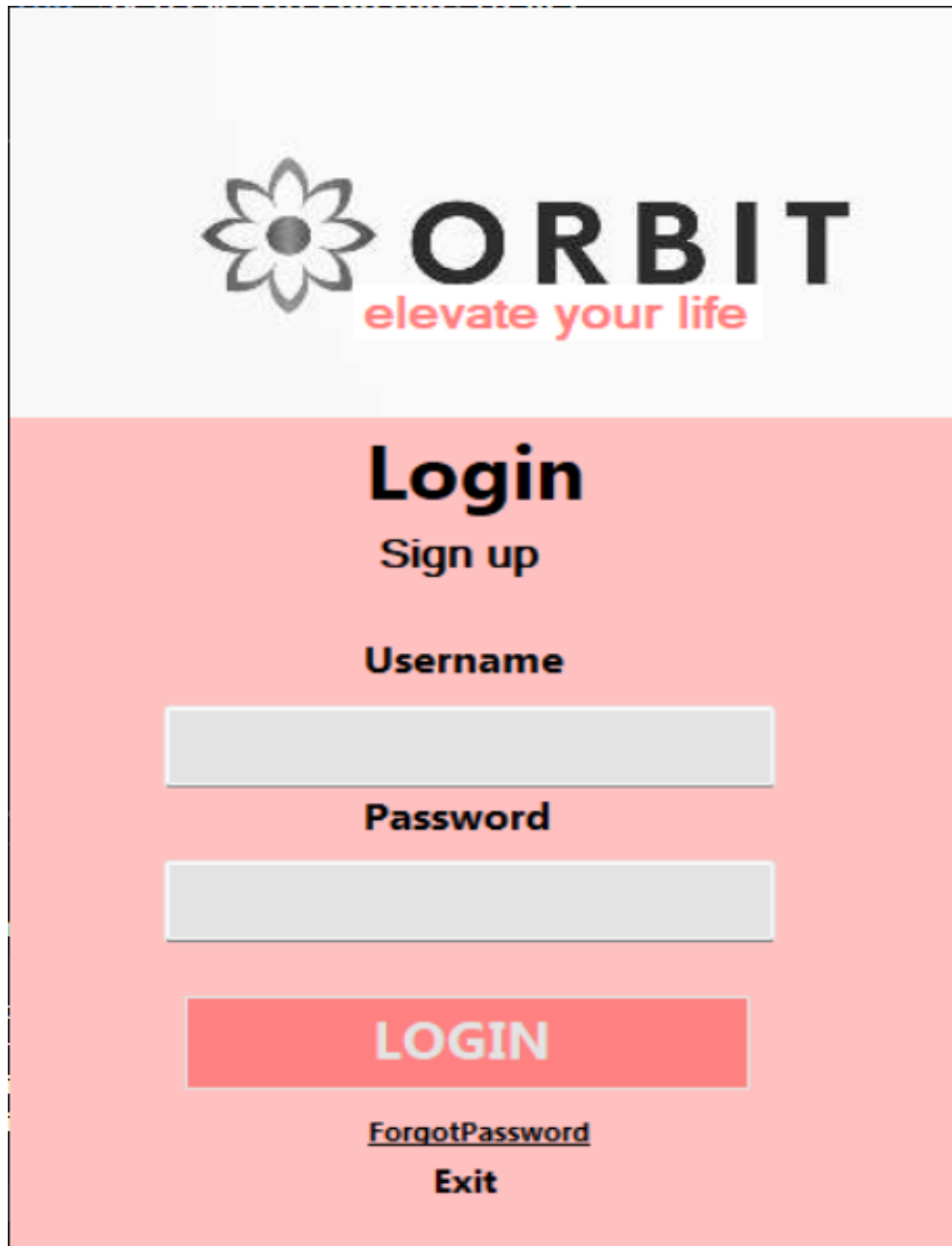


Figure 15: Model-View-Controller (MVC) Architecture Diagram

0.8 Implementation



The image shows a login page for a system named ORBIT. The page has a light gray header with the ORBIT logo, which consists of a stylized flower icon and the text "ORBIT" in large, bold, black letters, with the tagline "elevate your life" in red below it. The main content area has a light red background. It features a "Login" heading in bold black text, followed by a "Sign up" link in blue. Below these are two input fields: "Username" and "Password", each with a light gray rectangular box. A red "LOGIN" button is positioned below the password field. At the bottom, there are two links: "ForgotPassword" (underlined in blue) and "Exit" (in bold black).

ORBIT
elevate your life

Login
[Sign up](#)

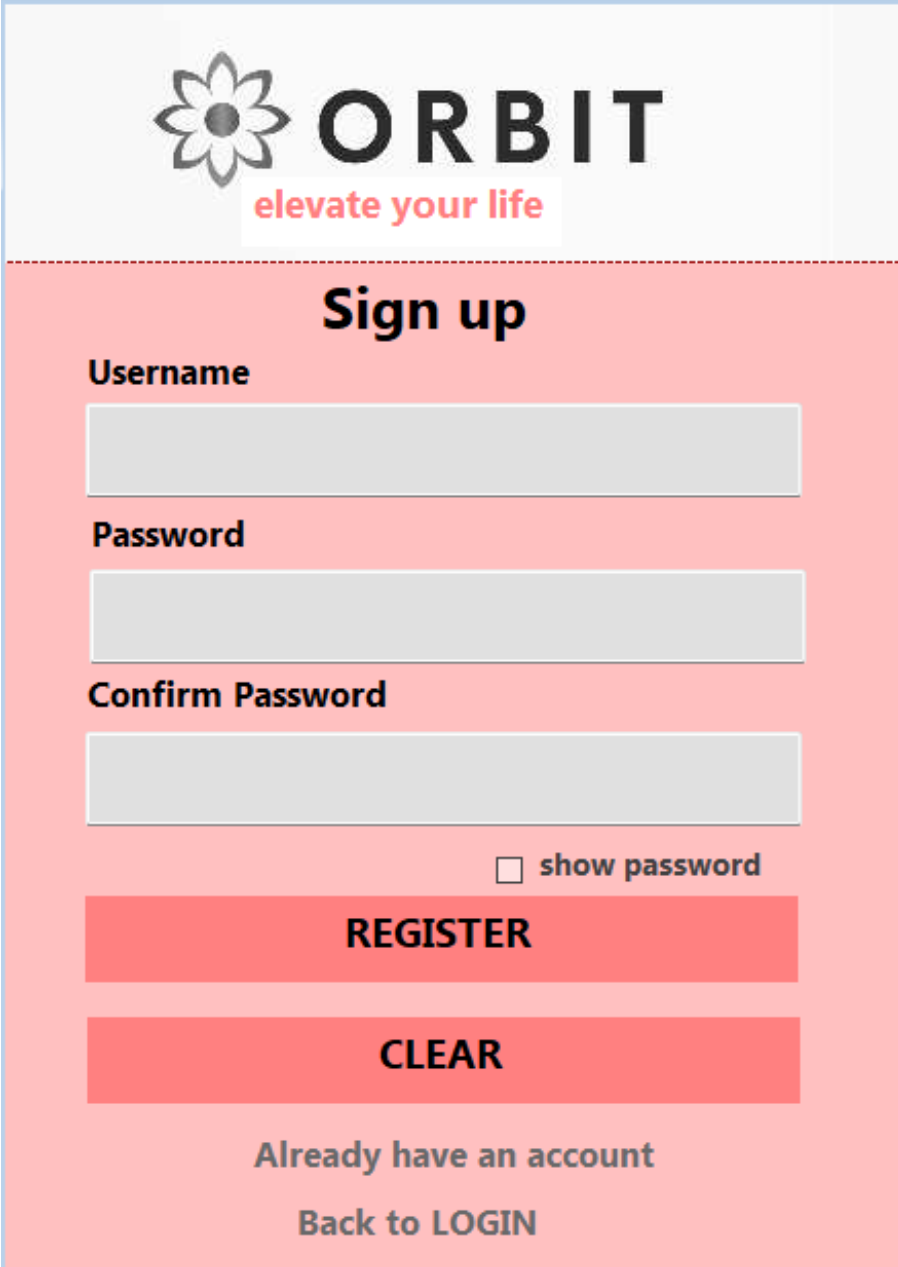
Username

Password


LOGIN

[ForgotPassword](#)
Exit

Figure 16: Login Page



The image shows a sign-up page for a service called ORBIT. At the top, there is a logo consisting of a stylized flower icon and the word "ORBIT" in large, bold, black letters. Below the logo, the tagline "elevate your life" is written in a smaller, red font. The main heading "Sign up" is centered in a large, bold, black font. Below this, there are three input fields for "Username", "Password", and "Confirm Password", each with a light gray border. To the right of the "Confirm Password" field, there is a checkbox labeled "show password". Below the input fields, there are two red buttons with white text: "REGISTER" and "CLEAR". At the bottom, there is a link that says "Already have an account" followed by "Back to LOGIN" in a smaller, gray font.

 **ORBIT**
elevate your life

Sign up

Username

Password

Confirm Password

☐ show password

REGISTER

CLEAR

Already have an account
Back to LOGIN

Figure 17: Sign up Page

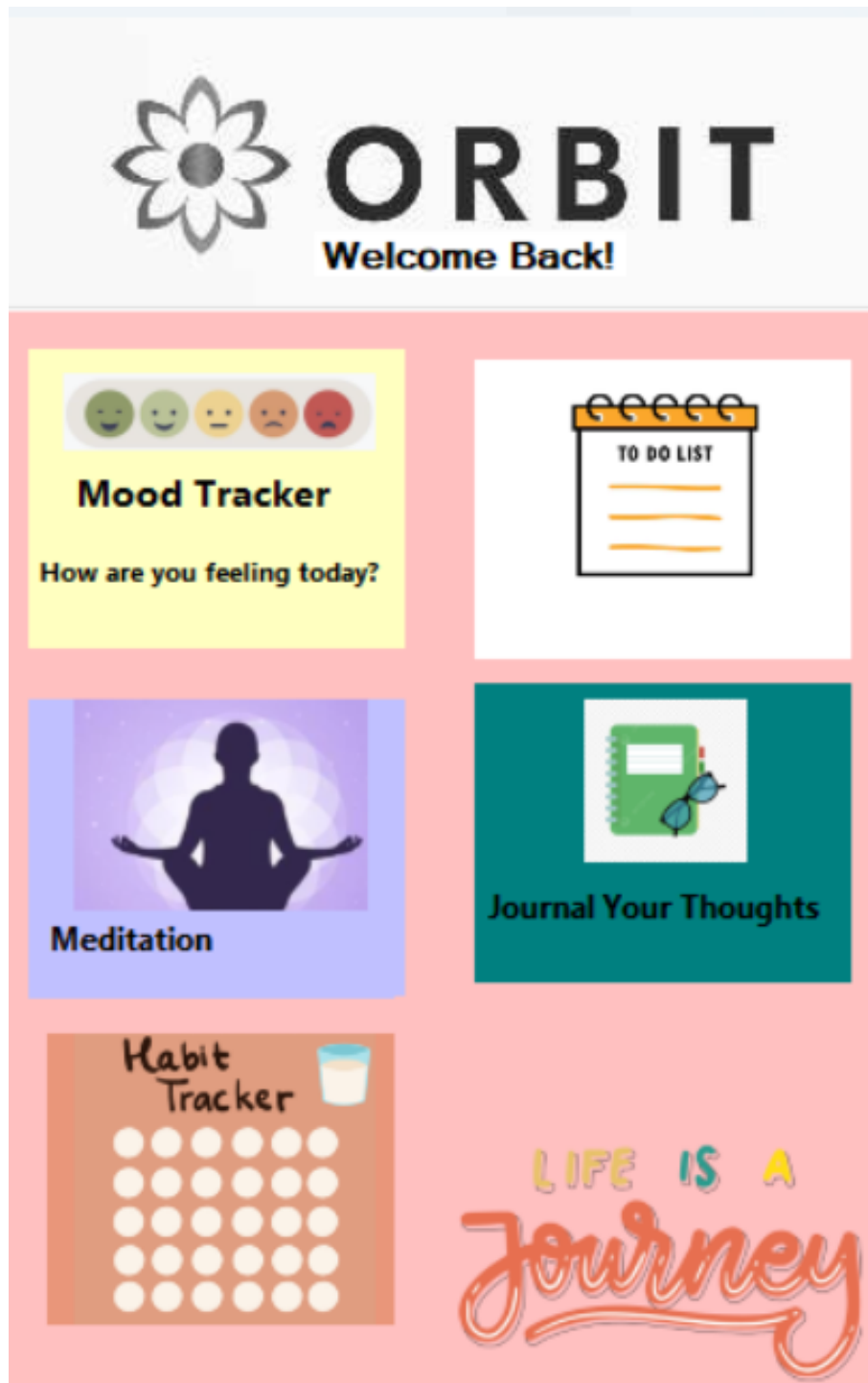


Figure 18: Home Page

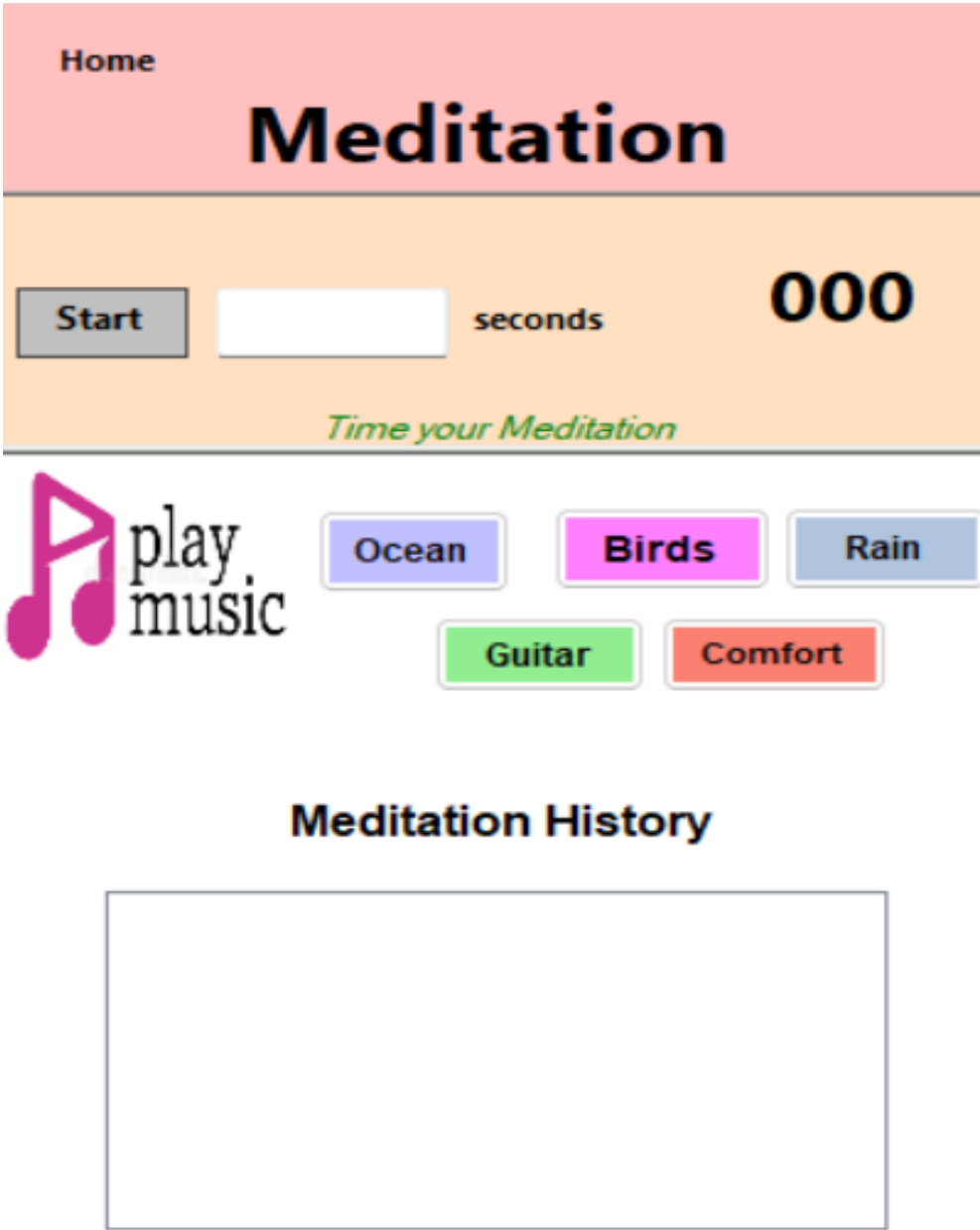



Figure 19: Meditation

Home

Todo List



Title

Description

Date and Time

New

Edit

Delete




Save

	Title	Description	Date and Time
*			

Figure 20: To do list

Home

Habit Tracker



Enter new habit

Add

Complete

View

Progress:

◀

April 2025

▶

Sun	Mon	Tue	Wed	Thu	Fri	Sat
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

Today: 4/28/2025

Figure 21: Habit tracker

[Home](#)

Select Mood

	Happy	Clear
	Relaxed	
	Sad	Add
	Angry	
	Stressed	

Description:


Figure 22: Mood tracker

Home

Journal your Thoughts

How was your day?

April 28, 2025



Save

April 26, 2025: hello i fukign ns...

April 26, 2025: helloooooooooo...

April 26, 2025: hello i had fun...

April 26, 2025: my day was awexome...

April 28, 2025: advanced statistics ...

Figure 23: Journal your thoughts

0.9 Product Burndown Chart

The Product Burndown Chart provides a visual representation of the remaining work in the project versus time. It helps in tracking the progress and predicting the completion timeline based on the sprint goals.

0.9.1 Burndown Chart Description

- **X-Axis:** Represents the Sprint Days (Time).
- **Y-Axis:** Represents the Remaining Story Points or Tasks.
- **Ideal Line:** Shows the perfect scenario of task completion rate.
- **Actual Line:** Represents the real progress made by the team during each sprint.

0.9.2 Burndown Chart

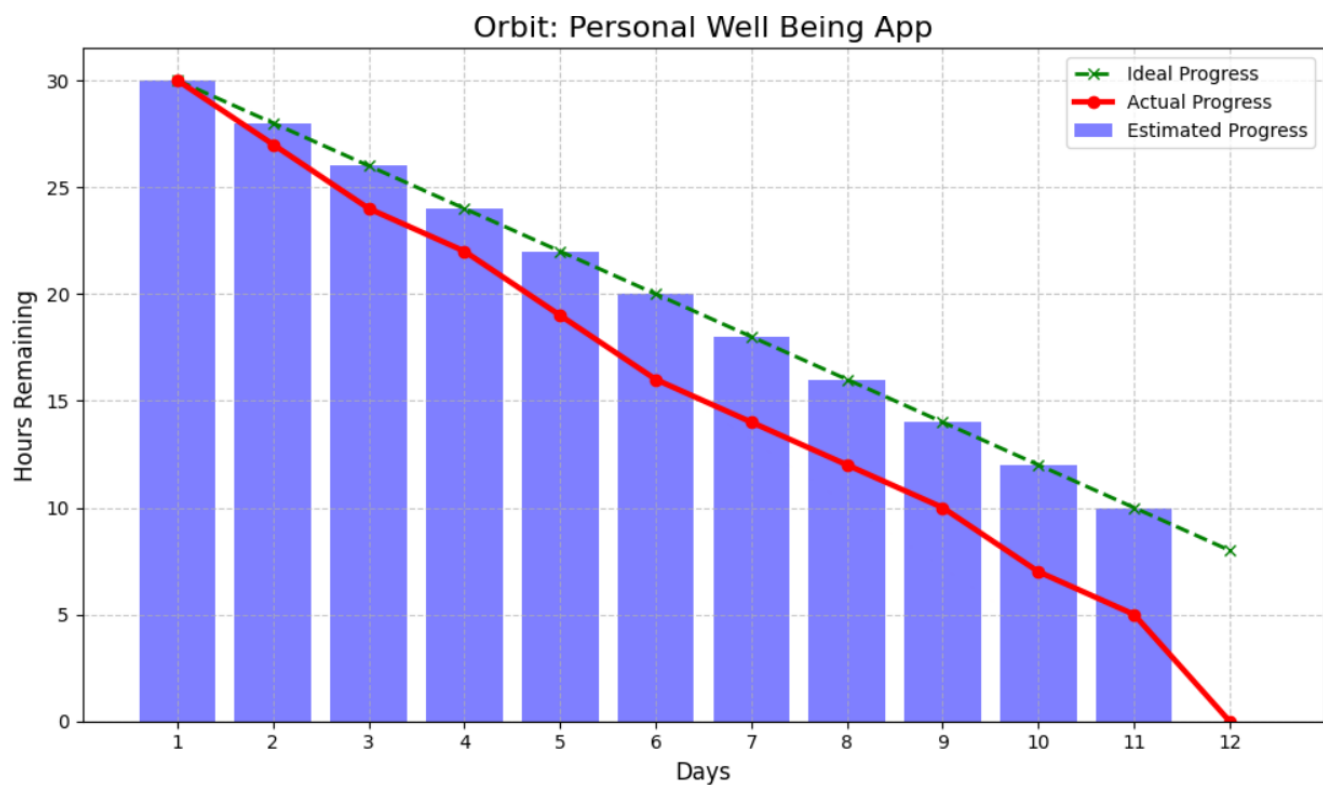


Figure 24: Product Burndown Chart for the Project

0.10 Boundary Value Analysis Testing of the Sign-up and Login Interfaces

For this project, we apply boundary value analysis to the Sign-up and Login forms to validate input fields like Username and Password. We especially focus on ensuring that:

- Username must be unique (for Sign-up).
- Password must be at least 8 characters long, with at least one uppercase letter, one lowercase letter, and one number.

0.10.1 Boundary Value Analysis for Sign-up Form

Password Field (Minimum Length 8)

The password must have a minimum of 8 characters. Thus, boundary values are tested around 8.

Test Case	Password Length	Expected Result
TC1	7 (min-1)	Fail: Password too short
TC2	8 (min)	Pass: Valid length
TC3	9 (min+1)	Pass: Valid length
TC4	15 (mid)	Pass: Valid length
TC5	19 (max-1)	Pass: Valid length
TC6	20 (max)	Pass: Valid length
TC7	21 (max+1)	Fail or Accept (depending on system limit)

Table 1: Boundary Value Analysis for Password Length in Sign-up Form

Username Field (Uniqueness)

We assume username must be unique and has a reasonable character limit (e.g., 20 characters max).

Test Case	Username Input	Expected Result
TC1	Already existing username	Fail: Username already taken
TC2	New unique username (min-1 character, empty)	Fail: Username required
TC3	1 character (min)	Pass: Valid unique username
TC4	2 characters (min+1)	Pass: Valid unique username
TC5	10 characters (mid)	Pass: Valid unique username
TC6	19 characters (max-1)	Pass: Valid unique username
TC7	20 characters (max)	Pass: Valid unique username
TC8	21 characters (max+1)	Fail: Username too long

Table 2: Boundary Value Analysis for Username in Sign-up Form

0.10.2 Boundary Value Analysis for Login Form

Username and Password Fields (Non-Empty)

Test Case	Username Input	Password Input	Expected Result
TC1	Empty	Any password	Fail: Username required
TC2	Valid username	Empty	Fail: Password required
TC3	Valid username	Password length 7 (min-1)	Fail: Password too short
TC4	Valid username	Password length 8 (min)	Pass: Successful login if correct
TC5	Valid username	Password length 9 (min+1)	Pass: Successful login if correct
TC6	Invalid username	Any password	Fail: User not found
TC7	Valid username	Incorrect password	Fail: Incorrect credentials

Table 3: Boundary Value Analysis for Login Form

Password Field (Character Requirements)

The password must contain:

- At least one uppercase letter (A-Z)
- At least one lowercase letter (a-z)
- At least one digit (0-9)

Test Case	Password Input	Expected Result
TC1	all lowercase (e.g., "password")	Fail: No uppercase
TC2	all uppercase (e.g., "PASSWORD")	Fail: No lowercase
TC3	only letters, no numbers (e.g., "Password")	Fail: No digit
TC4	only numbers (e.g., "12345678")	Fail: No letters
TC5	combination of uppercase, lowercase, numbers (e.g., "Pass1234")	Pass: ValidPassword

Table 4: Character Requirement Testing for Password

0.11 Work Division

Team Member	Roles and Responsibilities
Dania Waseem (Team Lead)	<ul style="list-style-type: none"> - Managed Trello (Scrum Board) - SignUp UI and Implementation - Meditation UI and Implementation - Made Sequence Diagrams - Made Class Diagram - Made Use Case Diagrams - Made Activity Diagrams - LaTeX Documentation - Presentation Slides - ToDo List UI - Database Integration
Munaza Tariq	<ul style="list-style-type: none"> - Designed ToDo List UI - Login UI and Implementation - Journal UI and Implementation - Mood Tracker UI and Implementation - ToDo List Implementation - Implemented ToDo List Page - Designed and Implemented Habit Tracker - Made Architecture Design - LaTeX Documentation - Presentation Slides

Table 5: Work Division

0.12 Learning from this Project

Through the development of the Orbit application, we gained hands-on experience in building a real-world software project from scratch. We learned how to plan a project by creating sprint backlogs and breaking tasks into small, manageable user stories. We improved our technical skills by working with Windows Forms in C#, designing user-friendly interfaces, and connecting our application to an SQL database. We also explored important software engineering concepts like requirements gathering, validation techniques such as boundary value analysis, and architecture design using layered patterns. Additionally, we practiced teamwork by dividing tasks and using tools like Trello for sprint management. Overall, this project taught us how to apply both programming and software engineering practices together to build a complete, functional application that solves a real-world problem.