Software Requirements and Design Document

for

SereniTy

Prepared by Amna Amir, Maha Qaiser, & Amna Tahir

TechPocket

26th November 2024

Table of Contents

1. Introduction

1.1 Purpose

The purpose of SereniTy is to help people improve their emotional and mental well-being through a desktop application. It combines tools like mood tracking, emotional regulation exercises, and journaling. The application is built using Java, JavaFX (specifically Scene Builder), and MySQL for storing data securely. SereniTy is designed to be simple, effective, and accessible to anyone who needs support in managing their emotions.

1.2 Product Scope

SereniTy is a desktop application that supports emotional wellness. It is designed for anybody dealing with stress, anxiety, or those who want to improve their emotional habits. The app offers:

1. *A mood tracker to record how users feel daily.*
2. *Exercises to help regulate emotions.*
3. *A journal for writing thoughts and memories.*
4. *Tracking tools like a streak calendar for long-term progress.*

1.3 Title

SereniTy: A Desktop Application for Emotional Wellness.

1.4 Objectives

1. *Help users manage emotions through guided exercises.*
2. *Provide a daily mood tracking feature for better self-awareness.*
3. *Encourage positive habits using journaling and progress tracking.*
4. *Ensure secure data storage with MySQL.*
5. *Create a simple and easy-to-use desktop application.*

1.5 Problem Statement

Worldwide, individuals struggle with mental health issues due to an array of developmental, environmental and social factors. These lead to increased stress, anxiety, and a decline in overall well being.

Considering the lack of affordable quality mental health tools in the market, there is a significant need for SereniTy, an emotional wellness tool that combines a mood tracker, emotional regulator and memory documenter all in one.

Our product tailors to individual needs through utilizing the user's own positive experiences along with approved exercises to neutralize negative emotions. With its daily streak calendar, short and *long term improvement plans, and unique journal, this application ensures access to an efficient and free mental health management system for all.*

2. Overall Description

2.1 Product Perspective

SereniTy is a new product built using JavaFX, Java, and MySQL, designed for individual users. The system is self-contained and manages all user data internally, without relying on external systems. Key components include mood tracking, journaling, emotional exercises, and admin functionality for managing users and generating reports.

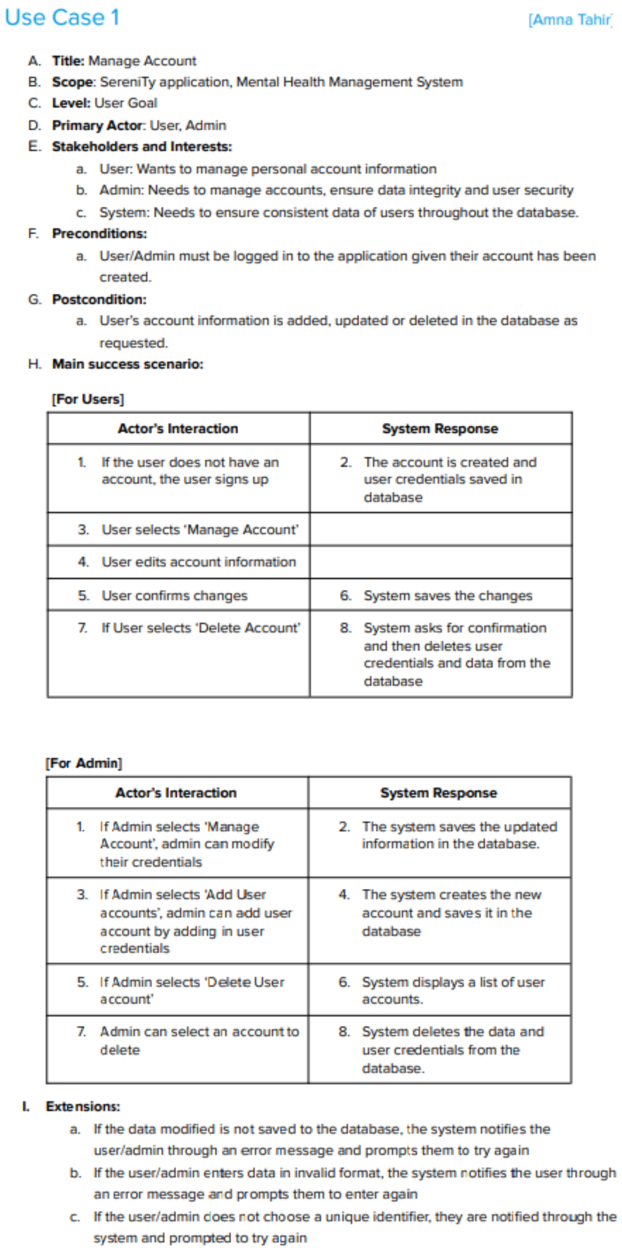
2.2 Product Functions

1. ***User Management:*** *Account creation, login, and profile updates.*
2. ***Mood Tracking:*** *Record and view moods through a calendar.*
3. ***Journaling:*** *Save and manage journal entries by date.*
4. ***Emotional Exercises:*** *Access and complete guided activities.*
5. ***Progress Tracking:*** *Monitor mood trends and streaks.*
6. ***Admin Features:*** *Manage users and generate activity reports.*

2.3 List of Use Cases

1. Manage Account (Add, Edit, Delete)
2. Enter Negative Mood
3. Enter Positive Mood
4. Manage Journal (Add, Edit, Delete)
5. View Journal
6. View Mood Calendar
7. Generate Report
8. Set Improvement Goal
9. Submit Feedback
10. Practice Exercise

2.4 Extended Use Cases



Use Case 1

[Amna Tahir]

A. **Title:** Manage Account
B. **Scope**: SereniTy application, Mental Health Management System
C. **Level:** User Goal
D. **Primary Actor**: User, Admin
E. **Stakeholders and Interests:**
   a. User: Wants to manage personal account information
   b. Admin: Needs to manage accounts, ensure data integrity and user security
   c. System: Needs to ensure consistent data of users throughout the database.
F. **Preconditions:**
   a. User/Admin must be logged in to the application given their account has been created.
G. **Postcondition:**
   a. User's account information is added, updated or deleted in the database as requested.
H. **Main success scenario:**

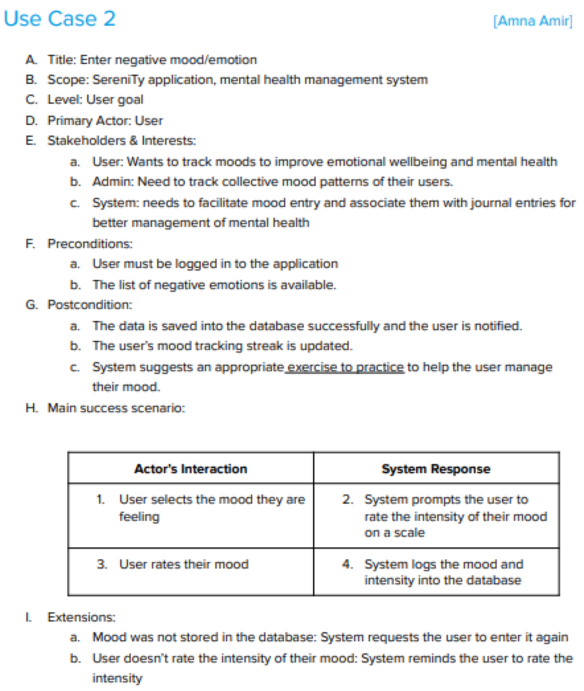
[For Users]

Actor's Interaction	System Response
1. If the user does not have an account, the user signs up	2. The account is created and user credentials saved in database
3. User selects 'Manage Account'	
4. User edits account information	
5. User confirms changes	6. System saves the changes
7. If User selects 'Delete Account'	8. System asks for confirmation and then deletes user credentials and data from the database

[For Admin]

Actor's Interaction	System Response
1. If Admin selects 'Manage Account', admin can modify their credentials	2. The system saves the updated information in the database.
3. If Admin selects 'Add User accounts', admin can add user account by adding in user credentials	4. The system creates the new account and saves it in the database
5. If Admin selects 'Delete User account'	6. System displays a list of user accounts.
7. Admin can select an account to delete	8. System deletes the data and user credentials from the database.

I. **Extensions:**
   a. If the data modified is not saved to the database, the system notifies the user/admin through an error message and prompts them to try again
   b. If the user/admin enters data in invalid format, the system notifies the user through an error message and prompts them to enter again
   c. If the user/admin does not choose a unique identifier, they are notified through the system and prompted to try again



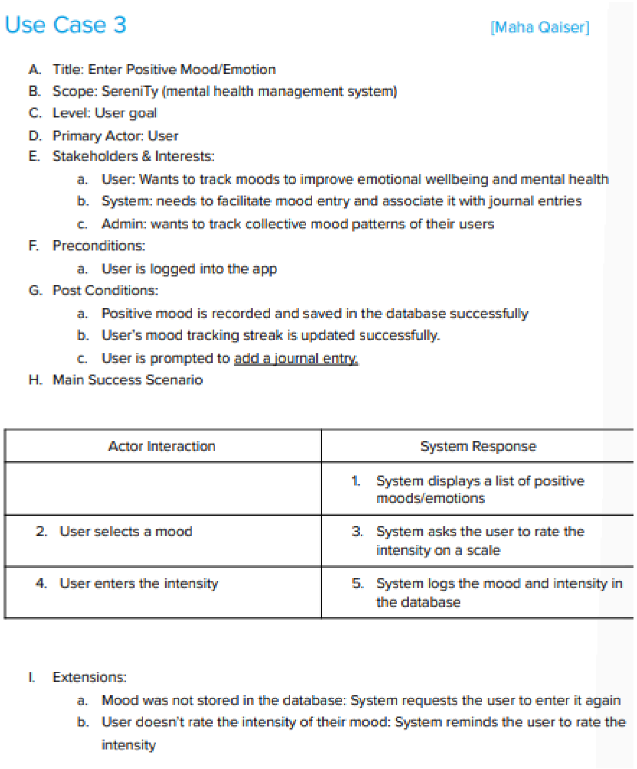
Use Case 2

[Amna Amir]

A. Title: Enter negative mood/emotion
B. Scope: SereniTy application, mental health management system
C. Level: User goal
D. Primary Actor: User
E. Stakeholders & Interests:
   a. User: Wants to track moods to improve emotional wellbeing and mental health
   b. Admin: Need to track collective mood patterns of their users.
   c. System: needs to facilitate mood entry and associate them with journal entries for better management of mental health
F. Preconditions:
   a. User must be logged in to the application
   b. The list of negative emotions is available.
G. Postcondition:
   a. The data is saved into the database successfully and the user is notified.
   b. The user's mood tracking streak is updated.
   c. System suggests an appropriate exercise to practice to help the user manage their mood.
H. Main success scenario:

Actor's Interaction	System Response
1. User selects the mood they are feeling	2. System prompts the user to rate the intensity of their mood on a scale
3. User rates their mood	4. System logs the mood and intensity into the database

I. Extensions:
   a. Mood was not stored in the database: System requests the user to enter it again
   b. User doesn't rate the intensity of their mood: System reminds the user to rate the intensity



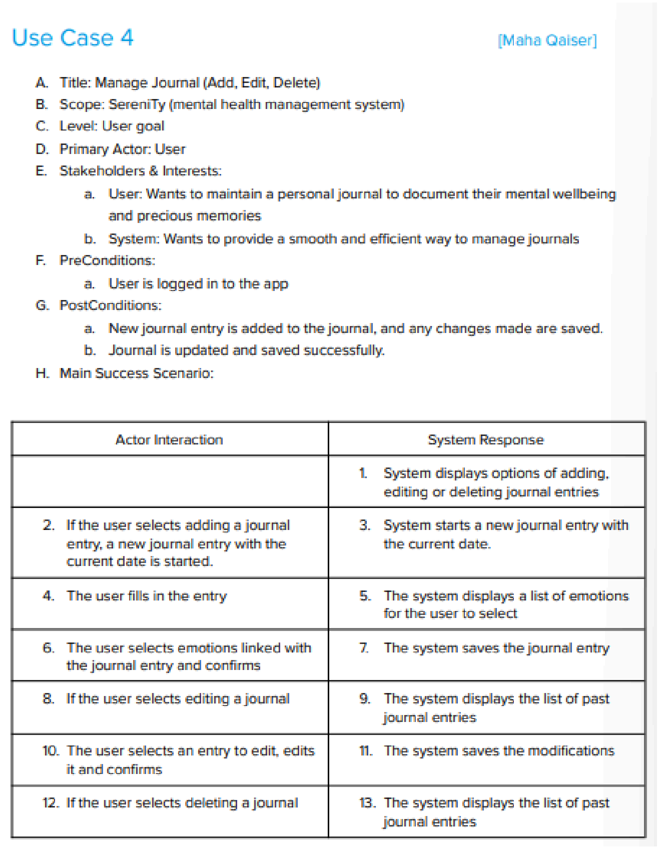
Use Case 3

[Maha Qaiser]

A. Title: Enter Positive Mood/Emotion
B. Scope: SereniTy (mental health management system)
C. Level: User goal
D. Primary Actor: User
E. Stakeholders & Interests:
   a. User: Wants to track moods to improve emotional wellbeing and mental health
   b. System: needs to facilitate mood entry and associate it with journal entries
   c. Admin: wants to track collective mood patterns of their users
F. Preconditions:
   a. User is logged into the app
G. Post Conditions:
   a. Positive mood is recorded and saved in the database successfully
   b. User's mood tracking streak is updated successfully.
   c. User is prompted to add a journal entry.
H. Main Success Scenario

Actor Interaction	System Response
	1. System displays a list of positive moods/emotions
2. User selects a mood	3. System asks the user to rate the intensity on a scale
4. User enters the intensity	5. System logs the mood and intensity in the database

I. Extensions:
   a. Mood was not stored in the database: System requests the user to enter it again
   b. User doesn't rate the intensity of their mood: System reminds the user to rate the intensity

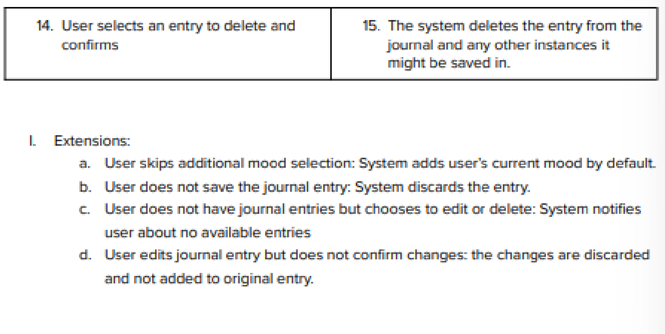


Use Case 4

[Maha Qaiser]

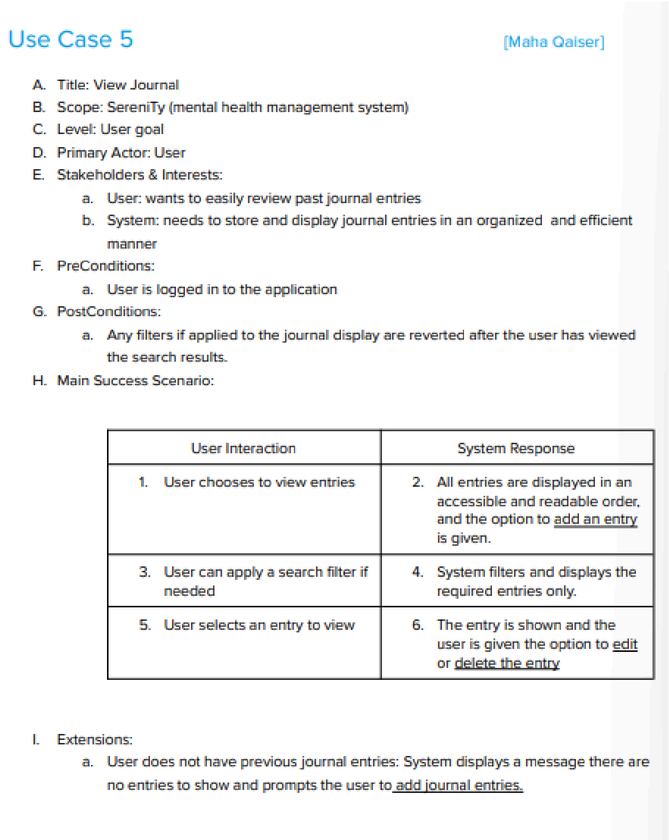
A. Title: Manage Journal (Add, Edit, Delete)
B. Scope: SereniTy (mental health management system)
C. Level: User goal
D. Primary Actor: User
E. Stakeholders & Interests:
  a. User: Wants to maintain a personal journal to document their mental wellbeing and precious memories
  b. System: Wants to provide a smooth and efficient way to manage journals
F. PreConditions:
  a. User is logged in to the app
G. PostConditions:
  a. New journal entry is added to the journal, and any changes made are saved.
  b. Journal is updated and saved successfully.
H. Main Success Scenario:

Actor Interaction	System Response
	1. System displays options of adding, editing or deleting journal entries
2. If the user selects adding a journal entry, a new journal entry with the current date is started.	3. System starts a new journal entry with the current date.
4. The user fills in the entry	5. The system displays a list of emotions for the user to select
6. The user selects emotions linked with the journal entry and confirms	7. The system saves the journal entry
8. If the user selects editing a journal	9. The system displays the list of past journal entries
10. The user selects an entry to edit, edits it and confirms	11. The system saves the modifications
12. If the user selects deleting a journal	13. The system displays the list of past journal entries



14. User selects an entry to delete and confirms	15. The system deletes the entry from the journal and any other instances it might be saved in.

I. Extensions:
  a. User skips additional mood selection: System adds user's current mood by default.
  b. User does not save the journal entry: System discards the entry.
  c. User does not have journal entries but chooses to edit or delete: System notifies user about no available entries
  d. User edits journal entry but does not confirm changes: the changes are discarded and not added to original entry.



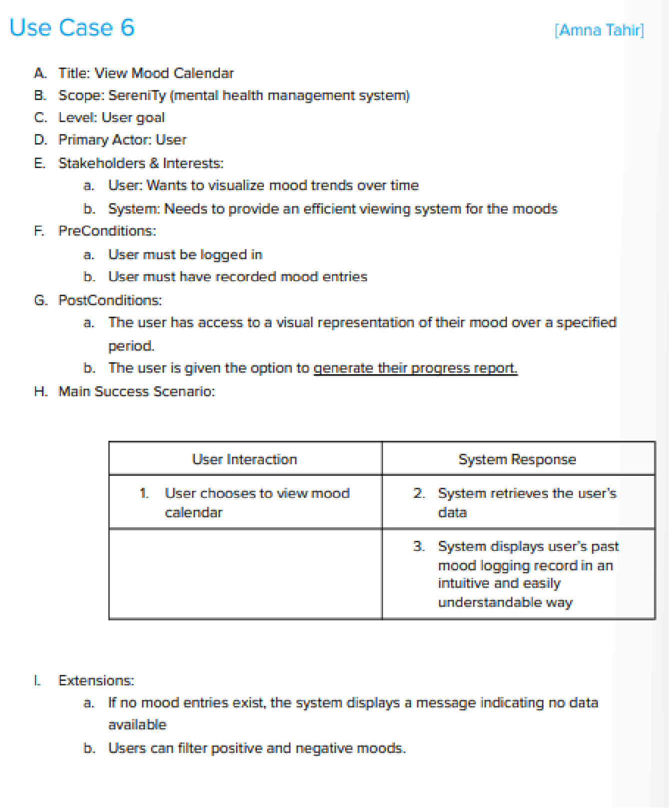
Use Case 5

[Maha Qaiser]

A. Title: View Journal
B. Scope: SereniTy (mental health management system)
C. Level: User goal
D. Primary Actor: User
E. Stakeholders & Interests:
  a. User: wants to easily review past journal entries
  b. System: needs to store and display journal entries in an organized and efficient manner
F. PreConditions:
  a. User is logged in to the application
G. PostConditions:
  a. Any filters if applied to the journal display are reverted after the user has viewed the search results.
H. Main Success Scenario:

User Interaction	System Response
1. User chooses to view entries	2. All entries are displayed in an accessible and readable order, and the option to add an entry is given.
3. User can apply a search filter if needed	4. System filters and displays the required entries only.
5. User selects an entry to view	6. The entry is shown and the user is given the option to edit or delete the entry

I. Extensions:
  a. User does not have previous journal entries: System displays a message there are no entries to show and prompts the user to add journal entries.



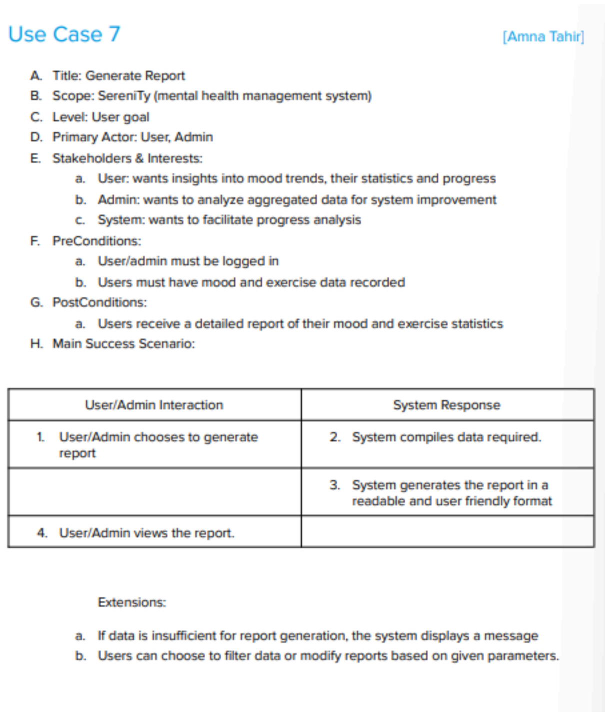
Use Case 6

[Amna Tahir]

A. Title: View Mood Calendar
B. Scope: SereniTy (mental health management system)
C. Level: User goal
D. Primary Actor: User
E. Stakeholders & Interests:
  a. User: Wants to visualize mood trends over time
  b. System: Needs to provide an efficient viewing system for the moods
F. PreConditions:
  a. User must be logged in
  b. User must have recorded mood entries
G. PostConditions:
  a. The user has access to a visual representation of their mood over a specified period.
  b. The user is given the option to generate their progress report.
H. Main Success Scenario:

User Interaction	System Response
1. User chooses to view mood calendar	2. System retrieves the user's data
	3. System displays user's past mood logging record in an intuitive and easily understandable way

I. Extensions:
  a. If no mood entries exist, the system displays a message indicating no data available
  b. Users can filter positive and negative moods.



Use Case 7

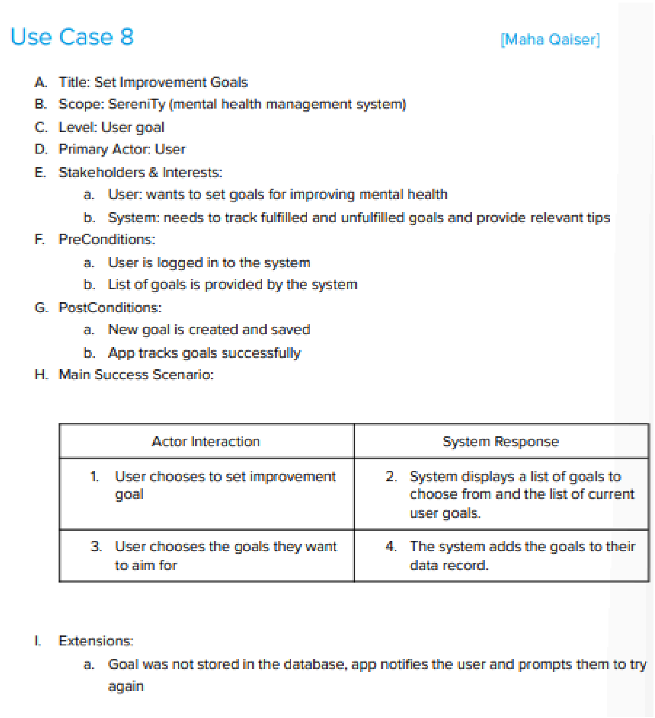
[Amna Tahir]

A. Title: Generate Report
B. Scope: SereniTy (mental health management system)
C. Level: User goal
D. Primary Actor: User, Admin
E. Stakeholders & Interests:
a. User: wants insights into mood trends, their statistics and progress
b. Admin: wants to analyze aggregated data for system improvement
c. System: wants to facilitate progress analysis
F. PreConditions:
a. User/admin must be logged in
b. Users must have mood and exercise data recorded
G. PostConditions:
a. Users receive a detailed report of their mood and exercise statistics
H. Main Success Scenario:

User/Admin Interaction	System Response
1. User/Admin chooses to generate report	2. System compiles data required.
	3. System generates the report in a readable and user friendly format
4. User/Admin views the report.	

Extensions:

a. If data is insufficient for report generation, the system displays a message
b. Users can choose to filter data or modify reports based on given parameters.



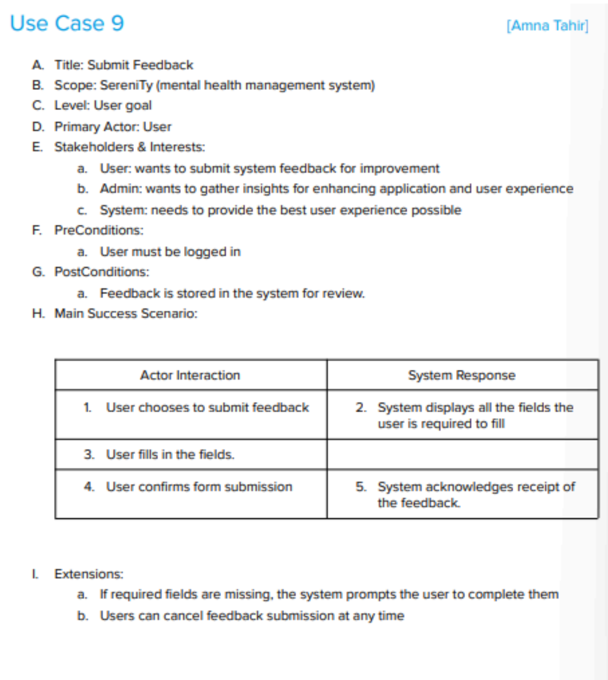
Use Case 8

[Maha Qaiser]

A. Title: Set Improvement Goals
B. Scope: SereniTy (mental health management system)
C. Level: User goal
D. Primary Actor: User
E. Stakeholders & Interests:
a. User: wants to set goals for improving mental health
b. System: needs to track fulfilled and unfulfilled goals and provide relevant tips
F. PreConditions:
a. User is logged in to the system
b. List of goals is provided by the system
G. PostConditions:
a. New goal is created and saved
b. App tracks goals successfully
H. Main Success Scenario:

Actor Interaction	System Response
1. User chooses to set improvement goal	2. System displays a list of goals to choose from and the list of current user goals.
3. User chooses the goals they want to aim for	4. The system adds the goals to their data record.

I. Extensions:
a. Goal was not stored in the database, app notifies the user and prompts them to try again



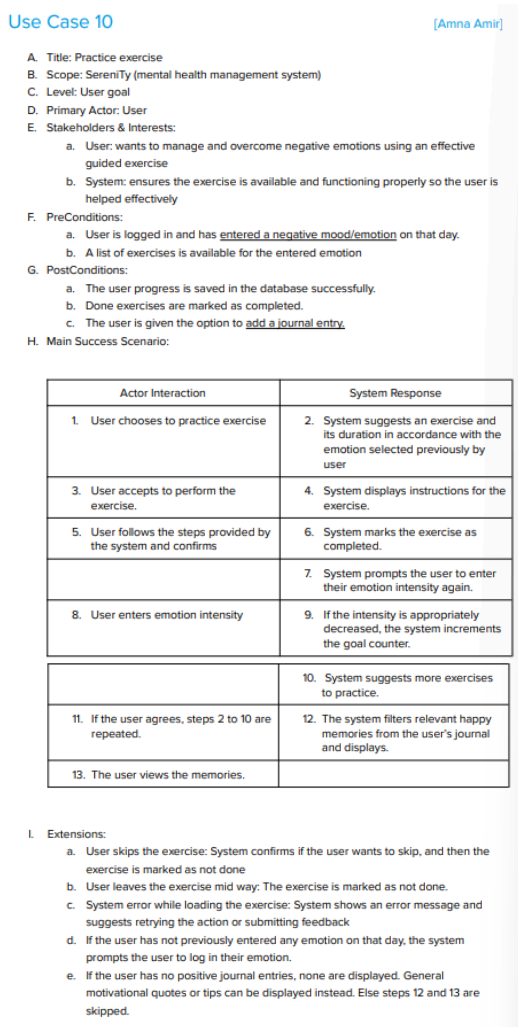
Use Case 9

[Amna Tahir]

A. Title: Submit Feedback
B. Scope: SereniTy (mental health management system)
C. Level: User goal
D. Primary Actor: User
E. Stakeholders & Interests:
a. User: wants to submit system feedback for improvement
b. Admin: wants to gather insights for enhancing application and user experience
c. System: needs to provide the best user experience possible
F. PreConditions:
a. User must be logged in
G. PostConditions:
a. Feedback is stored in the system for review.
H. Main Success Scenario:

Actor Interaction	System Response
1. User chooses to submit feedback	2. System displays all the fields the user is required to fill
3. User fills in the fields.	
4. User confirms form submission	5. System acknowledges receipt of the feedback.

I. Extensions:
a. If required fields are missing, the system prompts the user to complete them
b. Users can cancel feedback submission at any time



Use Case 10

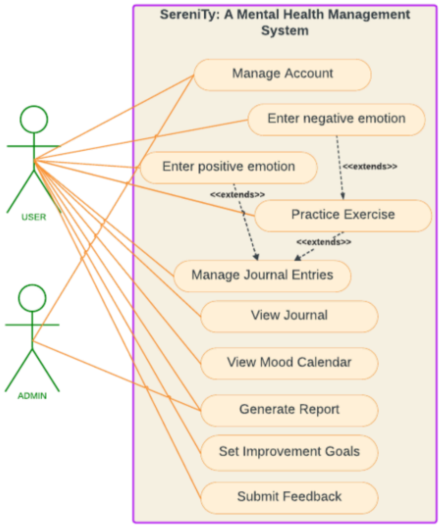
[Amna Amir]

A. Title: Practice exercise
B. Scope: SereniTy (mental health management system)
C. Level: User goal
D. Primary Actor: User
E. Stakeholders & Interests:
a. User: wants to manage and overcome negative emotions using an effective guided exercise
b. System: ensures the exercise is available and functioning properly so the user is helped effectively
F. PreConditions:
a. User is logged in and has entered a negative mood/emotion on that day.
b. A list of exercises is available for the entered emotion
G. PostConditions:
a. The user progress is saved in the database successfully.
b. Done exercises are marked as completed.
c. The user is given the option to add a journal entry.
H. Main Success Scenario:

Actor Interaction	System Response
1. User chooses to practice exercise	2. System suggests an exercise and its duration in accordance with the emotion selected previously by user
3. User accepts to perform the exercise.	4. System displays instructions for the exercise.
5. User follows the steps provided by the system and confirms	6. System marks the exercise as completed.
	7. System prompts the user to enter their emotion intensity again.
8. User enters emotion intensity	9. If the intensity is appropriately decreased, the system increments the goal counter.
	10. System suggests more exercises to practice.
11. If the user agrees, steps 2 to 10 are repeated.	12. The system filters relevant happy memories from the user's journal and displays.
13. The user views the memories.	

I. Extensions:
a. User skips the exercise: System confirms if the user wants to skip, and then the exercise is marked as not done
b. User leaves the exercise mid way: The exercise is marked as not done.
c. System error while loading the exercise: System shows an error message and suggests retrying the action or submitting feedback
d. If the user has not previously entered any emotion on that day, the system prompts the user to log in their emotion.
e. If the user has no positive journal entries, none are displayed. General motivational quotes or tips can be displayed instead. Else steps 12 and 13 are skipped.

2.5 Use Case Diagram



3. Other Nonfunctional Requirements

3.1 Performance Requirements

1. *Application should load within a few seconds.*
2. *Data retrieval from the database should be fast and efficient.*
3. *Application should process user inputs in real time.*

3.2 Safety Requirements

1. *The system should ensure data integrity, preventing accidental loss or overwriting of user entries*
2. *No harmful content or suggestions should be included in emotional exercises to ensure user safety and well-being.*
3. *Safeguards must be in place to prevent unauthorized access to sensitive user data.*
4. *Data should be recoverable from the last saved state in case of application failure.*

3.3 Security Requirements

1. *All accounts must be protected by password authentication.*
2. *All data must be securely stored in the database with restricted access.*

3.4 Software Quality Attributes

1. *Usability**:** The application should be user-friendly and easy to navigate.*
2. *Reliability**:** The system must operate without crashes or bugs for continuous use.*

3. *Maintainability**:** Code should be modular and well-documented for ease of updates or fixes.*

3.5 Business Rules

1. *Only authenticated users can access their personal data.*
2. *Admins can generate reports and manage users but cannot access personal journal entries or moods.*
3. *Users can only view and edit their own data and cannot access others' accounts.*
4. *Progress tracking and feedback are only displayed based on logged moods and exercises.*

3.6 Operating Environment

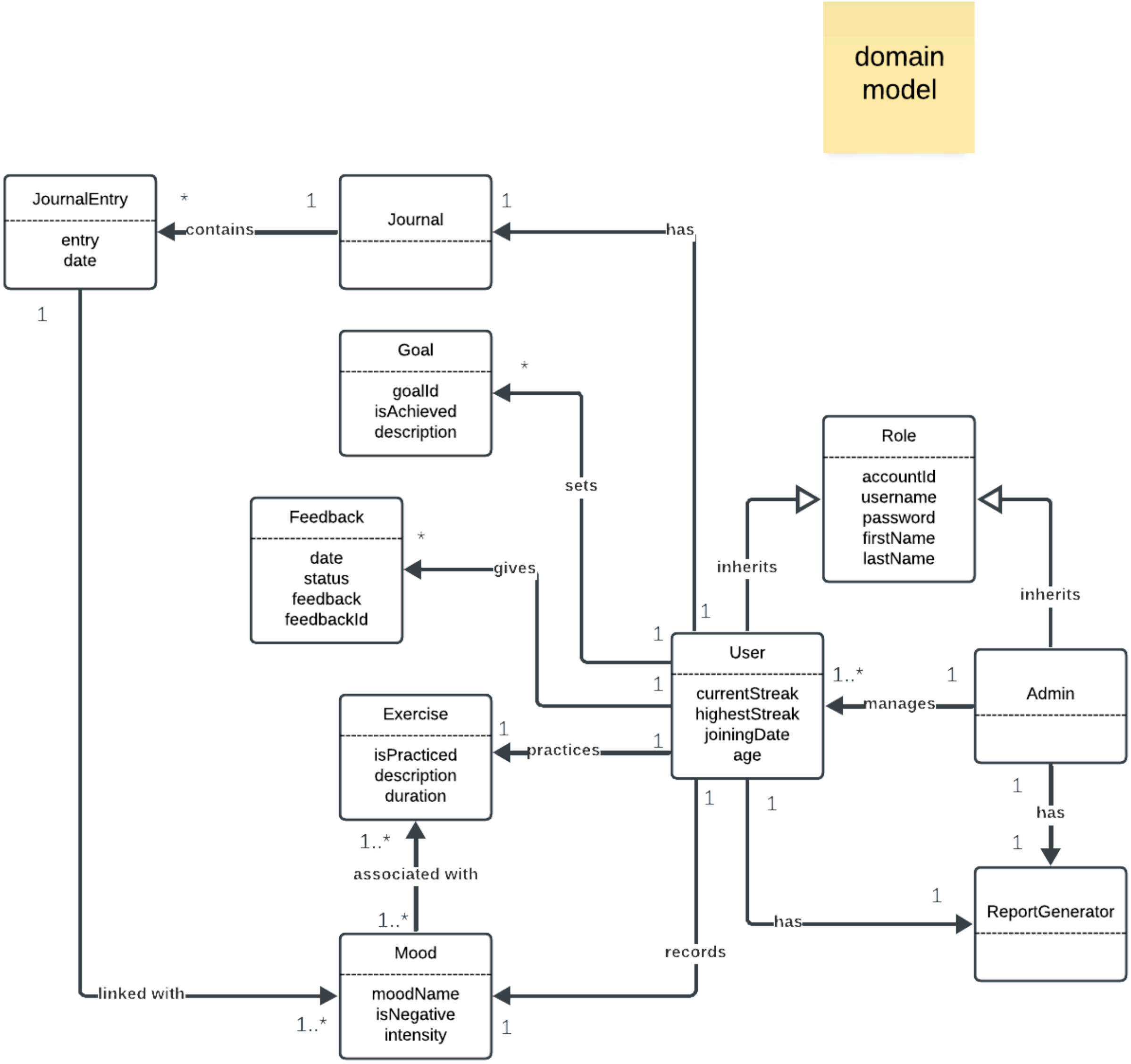
1. *Operating System: Compatible with Windows 10 or later and MacOS 10.15 or later.*
2. *Software Components: Requires Java Runtime Environment (JRE) and MySQL server.*

3.7 User Interfaces

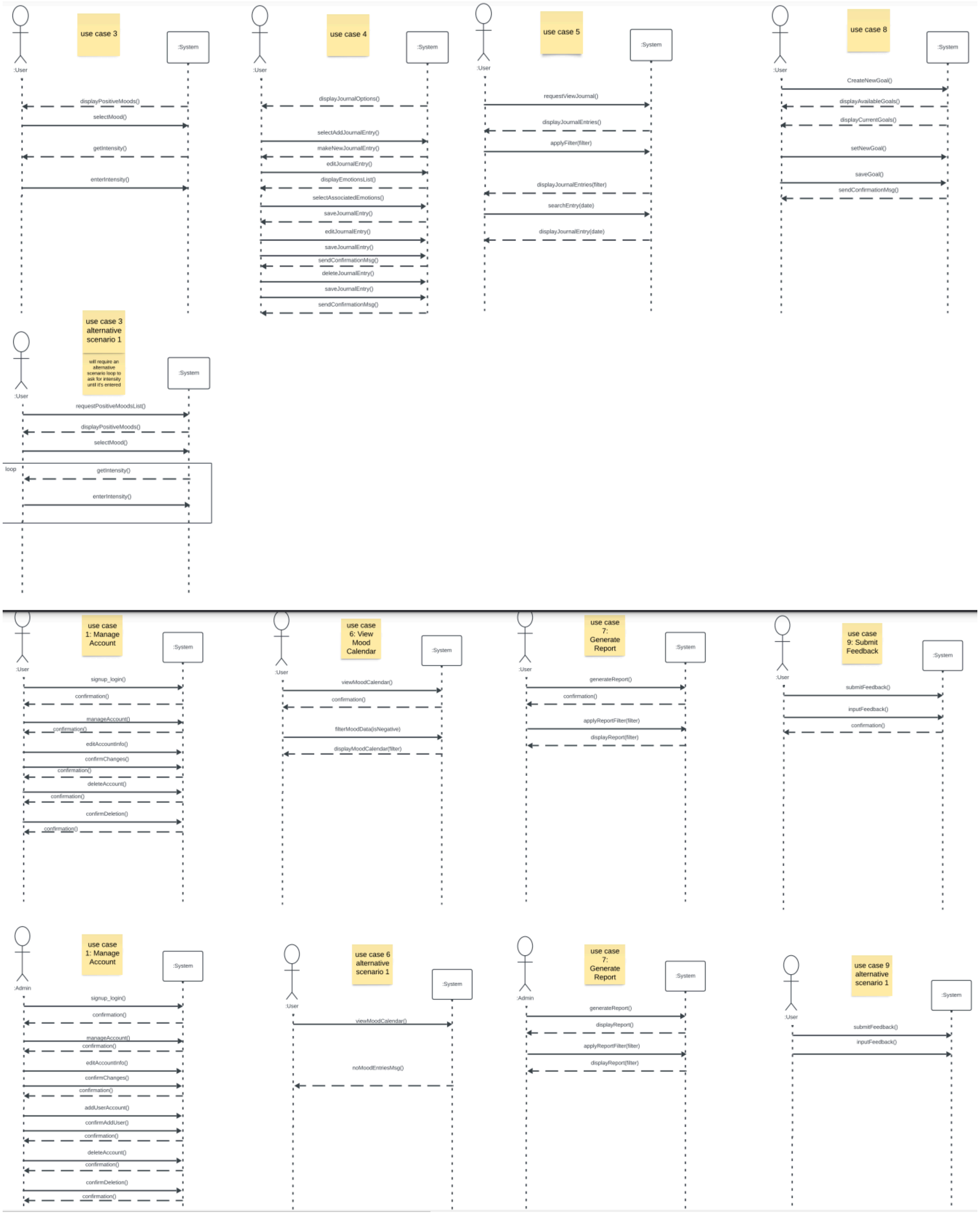
1. *Starting screen: asks for username and password, has buttons for login and signup, and a checkbox that admins will check when signing up or signing in.*
2. *Admin adding user screen: asks for user's details and has a button to add the user to the system.*
3. *Admin homepage screen: has buttons for generating reports, viewing feedback, adding user accounts, deleting user accounts, editing user accounts, viewing user accounts that are in the system*
4. *Admin signup screen: asks for admin's details and has a button to sign up.*
5. *Admin delete user screen: asks for user's details and has a button to remove the user to the system.*
6. *Admin edit user screen: asks for user's details and has a button to edit the user in the system.*
7. *User edits own account screen: asks for user's details and has a button to edit the user in the system.*
8. *Selecting daily mood screen: asks for mood through buttons and intensity through a combobox.*
9. *Mood Calendar screen: displays a table with the user's mood logs.*
10. *User's homepage screen: has buttons to view mood calendar, edit own account, delete own account, etc.*
11. *User's signup screen: asks for user's details and has a button to add the user to the system.*
12. *View accounts screen for admins: displays a table with users in the system.*
13. *Exercises screen: displays exercises the user has completed.*
14. *Highlights screen: displays happy journal entries as golden memories.*
15. *Journal entry screen: asks for the user's journal entry for the day.*
16. *Journal screen: displays all journal entries of the user.*
17. *Practice exercise screen: displays instructions/video for the exercise.*
18. *Goal screen: has buttons for add, edit, delete, and view.*
19. *Feedback screen: displays feedback from the users.*
20. *Reports screen: displays reports for the admin.*

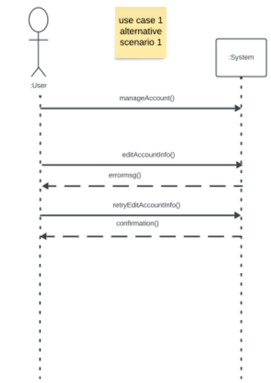
Additionally, there is a dashboard & alerts were also used in some places.

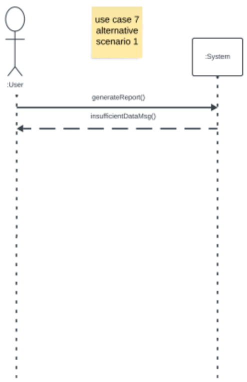
4. Domain Model

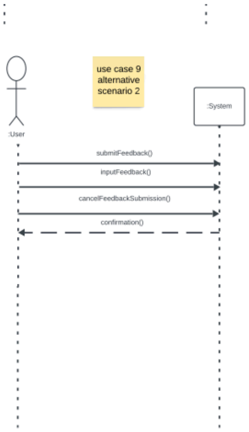


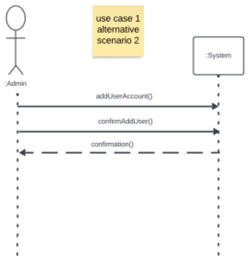
5. System Sequence Diagram

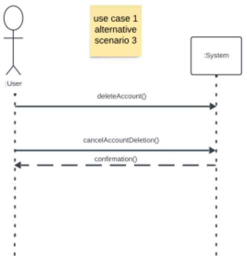


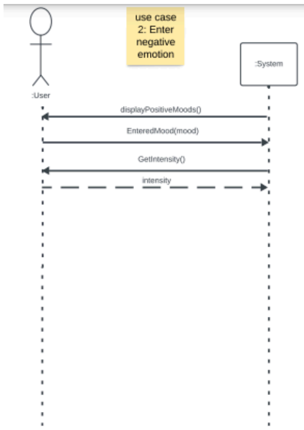

use case 1 alternative scenario 1
:User
:System
manageAccount()
editAccountInfo()
errormsg()
retryEditAccountInfo()
confirmation()

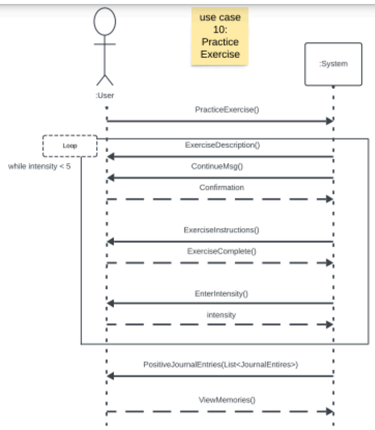

use case 7 alternative scenario 1
:User
:System
generateReport()
insufficientDataMsg()

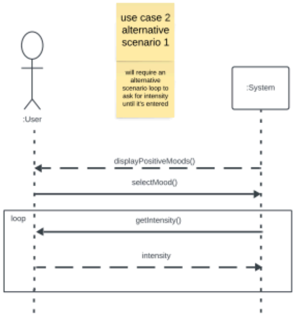

use case 9 alternative scenario 2
:User
:System
submitFeedback()
inputFeedback()
cancelFeedbackSubmission()
confirmation()

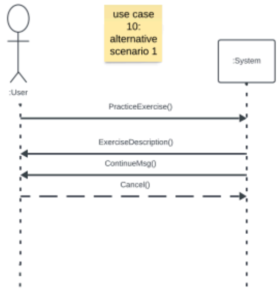

use case 1 alternative scenario 2
:Admin
:System
addUserAccount()
confirmAddUser()
confirmation()


use case 1 alternative scenario 3
:User
:System
deleteAccount()
cancelAccountDeletion()
confirmation()

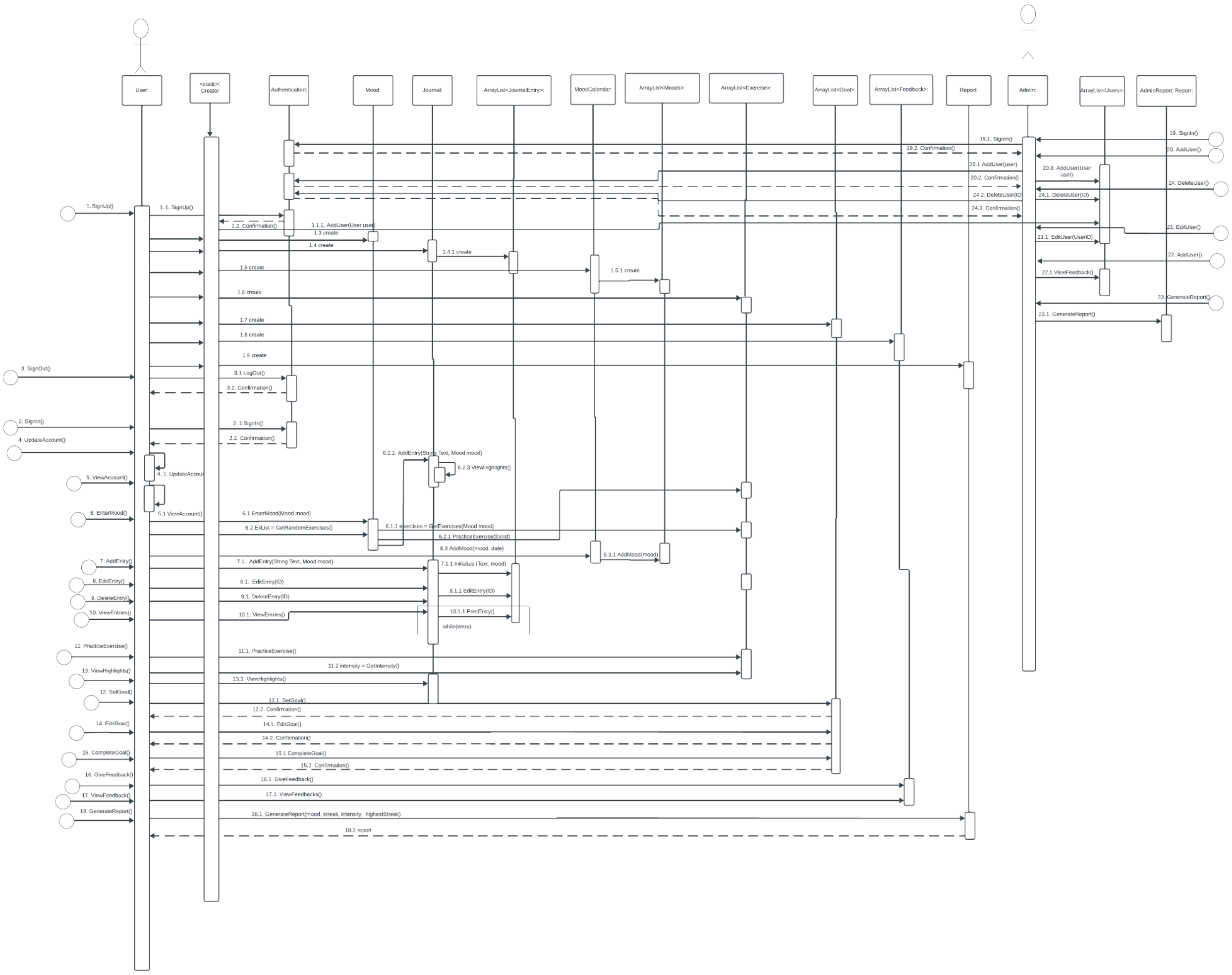

use case 2: Enter negative emotion
:User
:System
displayPositiveMoods()
EnteredMood(mood)
GetIntensity()
intensity


use case 10: Practice Exercise
:User
:System
PracticeExercise()
Loop
while intensity < 5
ExerciseDescription()
ContinueMsg()
Confirmation
ExerciseInstructions()
ExerciseComplete()
EnterIntensity()
intensity
PositiveJournalEntries(List<JournalEntires>)
ViewMemories()

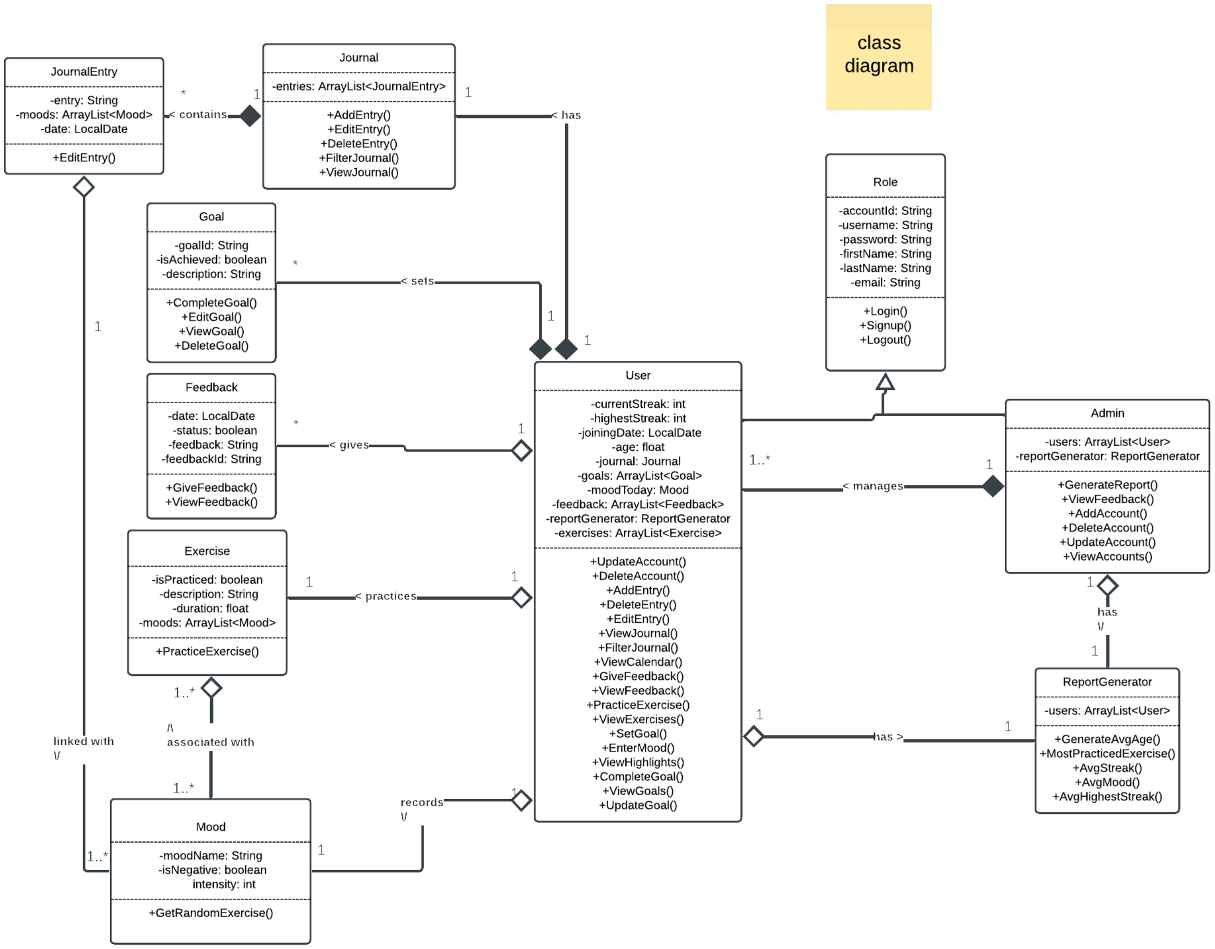

use case 2 alternative scenario 1
will require an alternative scenario loop to ask for intensity until it's entered
:User
:System
displayPositiveMoods()
selectMood()
loop
getIntensity()
intensity


use case 10: alternative scenario 1
:User
:System
PracticeExercise()
ExerciseDescription()
ContinueMsg()
Cancel()

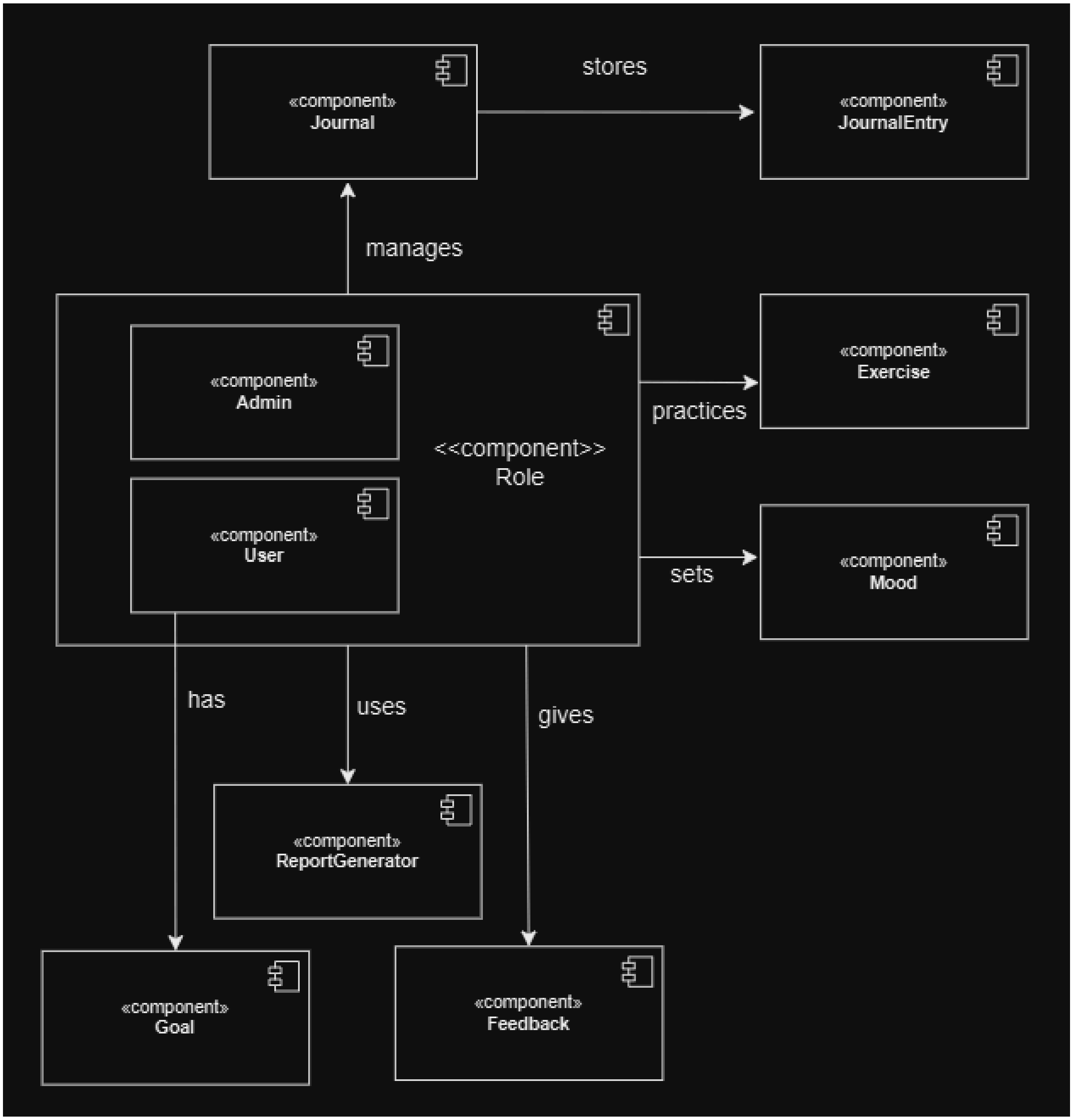
6. Sequence Diagram



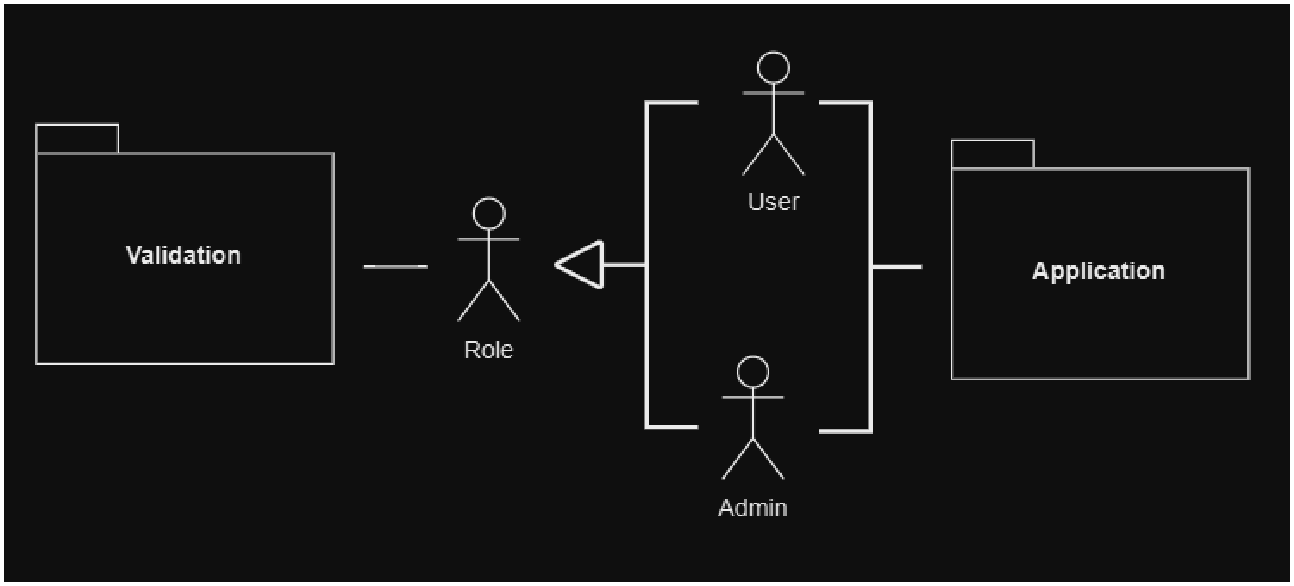
7. Class Diagram



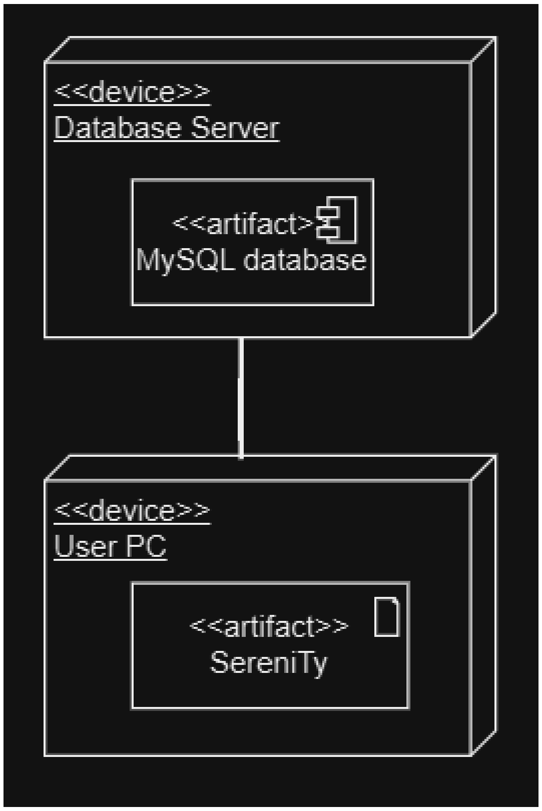
8. Component Diagram



9. Package Diagram



10. Deployment Diagram


<<device>>
Database Server
<<artifact>>
MySQL database
<<device>>
User PC
<<artifact>>
SereniTy