

Project Title

A Project Report Submitted to Fulfill the Requirements of CS289

By Group (6)

Rana Alenezi (SID:4455088)

Nuha Ahmed Almuqit (SID:4454973)

Linah Abdullah Almozaini(SID:4451095)

Raghad Ahmed Alharbi (SID: 4454854)

Weam Fahad Alhafdi(SID: 4454594)

Supervised By

Ohood Dakheelallah Alrohili

10/5/ 2025

Abstract

We proposed a system that uses NLP to solve this problem. The system categorizes the user's condition as simple or serious. If it is simple, it resolves it through therapy sessions provided by artificial intelligence. If it is serious, it refers the user to a psychologist. The system also allows for appointment bookings via the Sehaty app. It also conducts therapy sessions for patients based on their records. At the end of the session, it sends a report to the doctor detailing the user's condition. The results demonstrate that artificial intelligence is capable of organizing and managing appointments, leading to increased user satisfaction by performing routine tasks and providing rapid and early intervention. This system is revolutionizing digital health services.

Keywords: Mental Health; Artificial Intelligence; Natural Language Processing; Chatbot; Sehaty App

Table of Content

Abstract.....	2
Chapter 1: Introduction	4
1.1 Overview	4
1.2 Client Context	4
1.3 Similar Projects	4
1.4 Problem Statement	5
Chapter 2: Requirements Elicitation	5
2.1 Methodology	5
2.2 Requirements Listing	6
Chapter 3: System Models	8
3.1 Use Case Model	8
3.2 Object Model	12
3.3 Dynamic Model	12
Chapter 4: Architecture Models & Design Patterns	14
4.1 Architecture Models.....	14
4.2 Design Patterns	15
Chapter 6: Conclusion.....	18
6.1 Conclusion	18
6.2 Lessons Learnt	18
References	19

Chapter 1: Introduction

1.1 Overview

Sehaty is a health app operated by the Saudi Ministry of Health, providing health information and services such as appointment booking and consultations. The app faces a problem with the lack of appointments at psychiatric clinics due to a shortage of specialists. To address this problem, we propose an AI-powered system to accurately classify psychiatric cases and expedite the booking process, improving the user experience and reducing pressure on doctors.

AI contributes to automating case classification based on user data, automatically booking emergency appointments, and managing virtual therapy sessions without direct medical intervention. The system also continuously monitors psychiatric cases to determine the extent of improvement or the need for additional intervention.

1.2 Client Context

Sehhaty is a government healthcare provider focused on delivering digital health services (appointments, medical info, virtual support, etc) to citizens using mobile and web platforms.

Sehhaty currently lacks mental health support. The proposed enhancement adds an AI-powered section with a chatbot using NLP to assess users' emotional states and classify them as Mild, Moderate, or difficult. Simple and Moderate cases receive simple therapeutic sessions. Moderate users are monitored using Time Series Analysis, alerting specialists if their condition worsens. This reduces strain on professionals and ensures continuous user support

1.3 Similar Projects

Youper is a mental health chatbot that started as a research project at Stanford and Harvard. Their way of helping people is by using AI to have chat-based conversations that feel like therapy. It's easy to use, private, and helpful for general emotional support. But

it has some limits—it doesn't help with serious mental health problems and doesn't connect people to real doctors. Our project is different from Youper in many ways. It uses smarter AI to better understand how the user feels. If the case is serious, the system sends the user to a real psychiatrist and even books the appointment. It's also part of the Sehaty platform, so users can get real healthcare services. We also keep checking on the user's condition over time to make sure they always get the right help at the right time

1.4 Problem Statement

Given the continuous increase in mental health issues and the limited number of available psychologists, the gap between the demand for psychological services and the ability of psychologists to respond effectively is widening. This shortage makes it difficult for psychologists to classify new cases and follow up on registered cases. We proposed a system that uses artificial intelligence and natural language processing to interact with users and classify their psychological conditions. If a patient's condition is serious, they are automatically booked into a psychiatrist. The system handles minor cases through virtual sessions, and continuously monitors cases. This system aims to reduce the pressure on psychiatrists, expedite responses to mental illnesses, and improve the quality of services.

Chapter 2: Requirements Elicitation

2.1 Methodology

- This technique involves creating specific user scenarios that reflect real-life situations to determine user needs and system requirements.
- Why we used it: We use this **biased scenarios** to help us clarify the solution to the problem by mentioning the cases that the system can receive.

Scenario 1: When the user logs in for the first time, they will have a conversation with the AI. Based on this conversation, the AI will classify their condition. If it is simple, it

will offer them therapy sessions. If it is more complex, it will book an appointment with a doctor for diagnosis and treatment.

Scenario 2: Upon login, the system reviews the user's previous data and recorded mental health status. A virtual therapy session is initiated using artificial intelligence based on previous health records and an analysis of the user's current mental health status. During the session, the system monitors the user's response and adjusts the session to suit their needs.

Scenario 3 :At the end of every session, artificial intelligence generates a detailed report on the user's mental health progress. This report is automatically sent to the specialist physician for review. The physician reviews the report and determines whether it is necessary to modify the treatment plan based on the analysis.

Scenario 4: The system continues to monitor the user's condition and schedules subsequent sessions based on ongoing needs.

2.1 Methodology



We used prototype to visualize our system to gather feedback.

2.2 Requirements Listing

2.2.1 Functional Requirement:

1. The system shall allow users to log in using a username and password.
2. The system shall allow the user to initiate a virtual therapy session.

3. The system shall generate and present a personalized treatment plan for each patient based on their condition.
4. The system shall automatically schedule an appointment with a psychologist when the user's condition is assessed as deteriorating.
5. The system shall store a structured, and detailed medical history for each user.
6. The system shall generate clear and well-structured reports for psychologists, highlighting key updates in the patient's condition
7. Psychologists shall be able to change settings such as session frequency and type of therapy used

2.2.2 Non-Functional Requirements

1. Usability: The interface must be easy to use and find what it's search.
2. Security and protection of user data and maintaining privacy
3. The system should be able to accommodate a large number of users at the same time.
4. The system should work smoothly and not crash or show errors often.
5. The system shall process and respond to user input within a few seconds.
6. System support both language Arabic and English.
7. Technical support mustn't take more than 5 minutes.

Chapter 3: System Models

3.1 Use Case Model

Use Case Name	register and classify the patient's condition with the help of artificial intelligence.
Goal	Allowing the user to log in for the first time, have a conversation with the AI, and classify their mental health status.
Primary Actor	User (patient)
Secondary Actor	psychologist
Preconditions	The user must have the Sehaty app installed. The user must be a new user (first-time login)
Postconditions	The user has successfully logged in, and their mental health status is categorized. Either an appointment is booked with a psychiatrist or a treatment plan is presented based on their mental health condition.
Scenario	1. The user opens the app. 2. register 3. The system verifies the information and logs the user in. 4. The AI initiates a virtual conversation with the user to assess their psychological state. 5. Based on the user's answers, the AI classify their condition as mild, moderate, or severe. 6. If the condition is simple, the system offers virtual therapy sessions. 7. If the condition is difficult / serious the system schedules an appointment with a (psychologist)for further diagnosis and treatment.
Alternative	If the user encounters difficulty logging in due to incorrect information, they will be asked to try again after correcting the information. If the user's classification cannot be determined due to insufficient answers, the system will ask them to provide more details or refer them to a psychiatrist.
Exceptions	Invalid Data Entry: If the user enters incorrect data (e.g., an incorrect username), the system displays an error message and allows the user to try again. Insufficient Data for Classification: If the AI is unable to classify the user's mental health status due to insufficient or unclear answers, the system prompts the user to provide additional information or book an appointment with a doctor for further assessment.

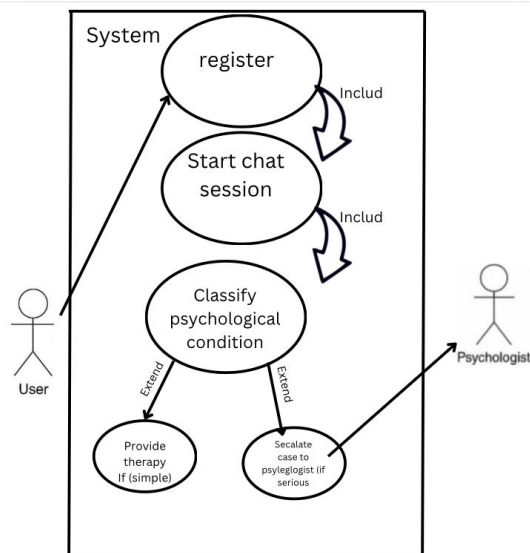


Use Case Name	Log in and track user status based on previous reports.
Goal	Allow the system to review the user's past mental health data, initiate a virtual therapy session, and adjust the session based on the user's responses.
Primary Actor	User (patient)
Secondary Actor	
Preconditions	The user must be a former user, have been previously classified by the system, and have health records in the system.
Postconditions	The patient's records are reviewed, virtual therapy sessions are initiated
Scenario	<ol style="list-style-type: none"> 1. The user logs into the system. 2. The system reviews the user's previous data and mental health status. 3. A virtual therapy session is initiated using AI based on the previous health records and analysis of the user's current status. 4. The system monitors the user's responses and adjusts the session accordingly.
Alternative	If the system fails to retrieve previous data or the user's data is insufficient, the system will ask the user to answer additional questions or refer them to a psychiatrist to verify their condition.
Exceptions	<ol style="list-style-type: none"> 1. Session Adjustment Failure: If the system fails to adjust the therapy session due to misclassification or technical issues, it asks the user to provide clarification or refer them to a psychiatrist. 2. Connectivity Issues: If there is a network failure during the session (loss of internet connection), the system informs the user of the issue and provides an option to retry later or continue using offline features until the connection is restored

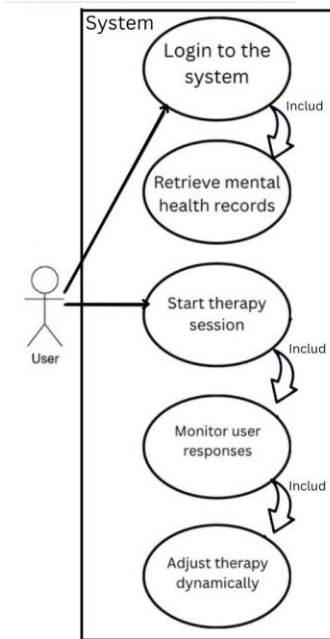
Use Case Name	Create and submit reports on user mental health
Goal	The system generates reports on the user's psychological state after the virtual session, saves them, and sends them to the psychologist
Primary Actor	User (patient)
Secondary Actor	Psychiatrist

Preconditions	The virtual therapy session must be completed. The system must have access to the user's most recent responses and progress data.
Postconditions	A progress report is successfully generated and sent to the physician for review.
Scenario	<ol style="list-style-type: none"> 1. The AI generates a detailed report based on the user's responses during the session. 2. The AI reviews the progress based on the analysis and updates the report. 3. The report is automatically sent to the assigned (psychologist)for review. 4. The psychologist evaluates the report and decides whether any modifications are necessary for the treatment plan. 5. If needed, the doctor updates the treatment plan or suggests further actions.
Alternative Path	1. If the AI fails to generate the report, the system prompts the user to retry or provide additional details.
Exceptions	1. AI Report Creation Failure: If the AI fails to create a report due to missing data or incomplete questions, the system sends a message to the user asking them to provide more information.

3.1.2 Use Case Diagram :

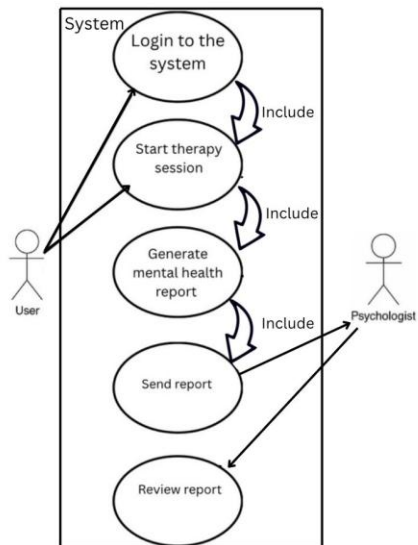


Log in for the first time and classify the patient's condition with the help of artificial intelligence.



Log in and track user status based on previous reports.

2

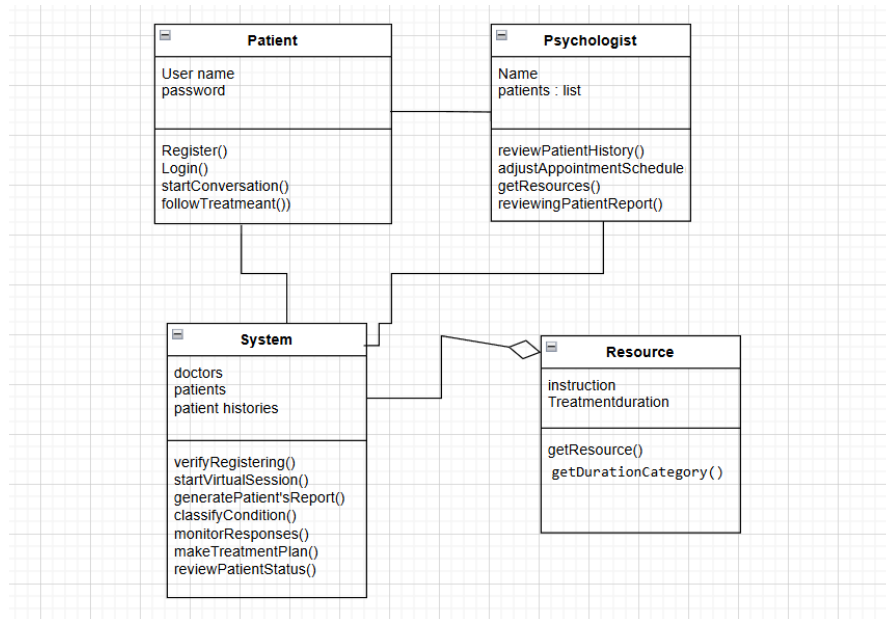


Create and submit reports on user mental health.

3

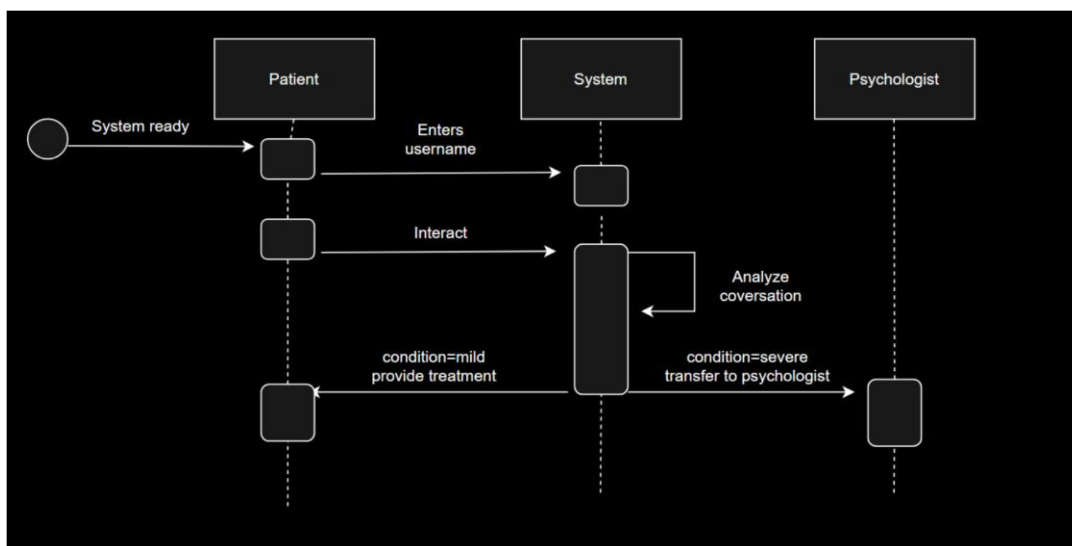
3.2 Object Model

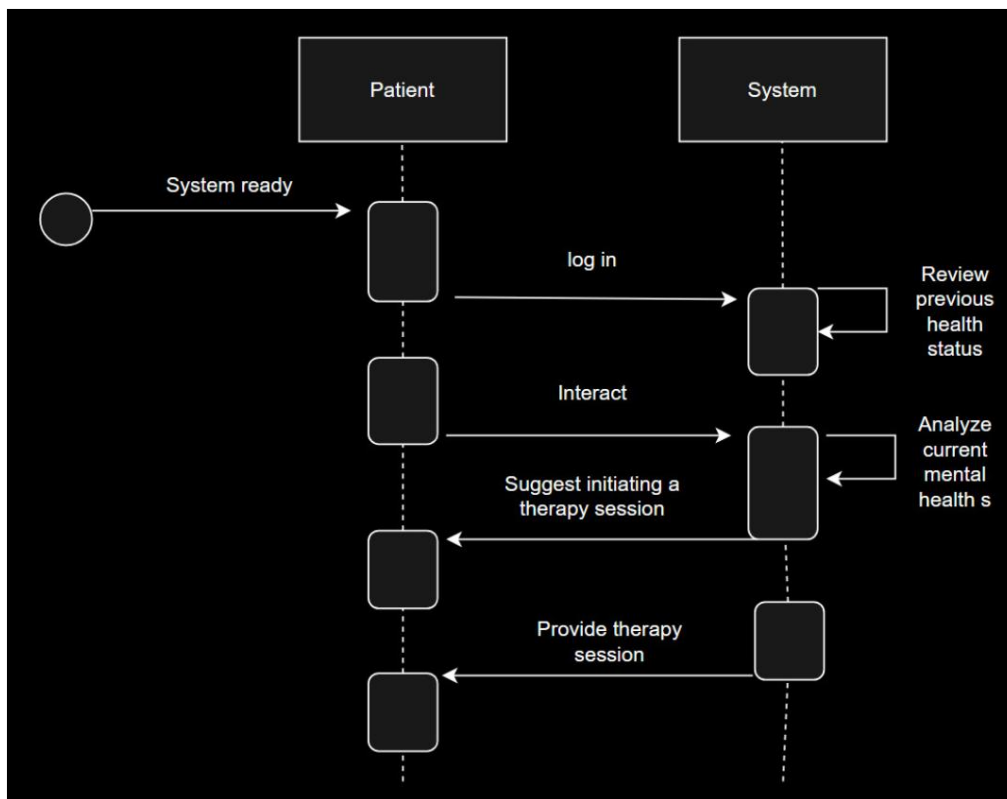
3.2.1 Class Diagram



3.3 Dynamic Model

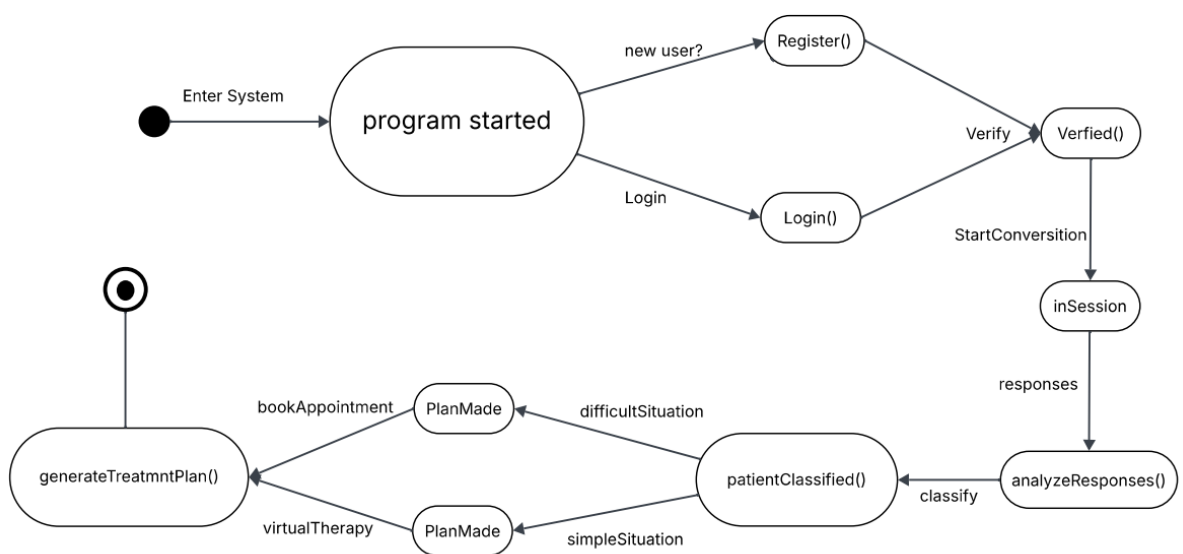
3.3.1 Sequence Diagrams





3.3.2 State Diagram

Log in for the first time and classify the patient's condition with the help of artificial intelligence.

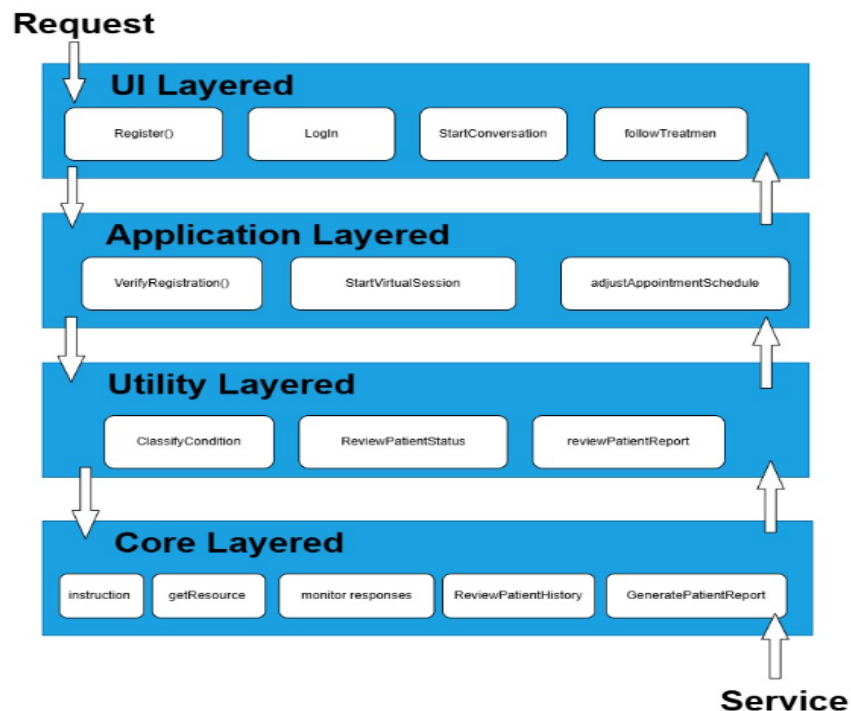


Chapter 4: Architecture Models & Design Patterns

4.1 Architecture Models

- The proposed system is based on artificial intelligence. It includes interactive interface and chatbot an artificial intelligence module using natural language to classify mental health cases, and a database for storing reports.
- We chose the layered architecture model because it divides the system into independent layers: the interface layer for interacting with the user, the application layer for managing sessions, the AI layer for classifying conditions, and the data layer for storing records. This division facilitates understanding, testing, and modifying the system. The design also includes an architectural diagram illustrating the main components, such as the user interface, business logic, the AI module, and data storage, illustrating the interactions between them layered architecture is the most convenience for our system, these separated layered support more security for the system which make harder to break and access the data system.

Layered Diagram



4.2 Design Patterns

. Creational Patterns

- Factory Method: This pattern helps the system decide what to do based on the user's mental state. Factory: SessionFactory (decides whether to start a therapy session or refer the user to a psychologist).

Concrete Factories:

▪ SimpleSessionFactory: Creates RelaxationSession, MotivationTipsGenerator, DailyReminderMessage.

▪ ModerateSessionFactory: Creates CBTSelfHelpSession

▪ defucalSessionHandler: Creates ReferralNotice, and sends the case to the psychologist. The system chooses the factory after AI classifies the condition.

• Builder Pattern: We used this to build therapy plans with different steps depending on the case. Builder: TherapyPlanBuilder (adds sessions, follow-ups, and alert levels). Concrete Builders:

▪ SimplePlanBuilder: creates a simple plan with light sessions and basic reminders.

▪ ModeratePlanBuilder: includes more follow-ups and an alert system.

▪ difficultPlanBuilder: includes emergency alerts, psychologist notifications. TherapyDirector: uses these builders to build the plan based on the user's mental state.

2. Structural Patterns

• Adapter Pattern: We used adapters to make the system work with outside tools. SentimentAPIAdapter: Connects to third-party emotion analysis and returns structured mood data. HealthRecordsAdapter: Reads old records from Ministry of Health

• Container Pattern: Containers helped us keep the system parts clean and organized. TherapySessionManager: manages the flow of therapy sessions and tracks user responses. ProgressReportHandler: creates and sends reports about the user's improvement. MentalStateAnalyzer: uses AI to evaluate the user's current mental condition

3. Behavioral Patterns

- Chain of Responsibility: To make the classification more accurate. TextAnalysisClassifier: reads what the user wrote in the chat and tries to see if there are signs of sadness, stress, or anxiety. QuestionnaireClassifier: checks the answers the user gave in the mental health form or survey HistoryClassifier: Looks at previous sessions. If one fails, the next continues the process.
- Command Pattern: Commands help organize system actions. StartSessionCommand, GenerateReportCommand, NotifyDoctorCommand. MentalHealthEngine runs the correct command

Illustrative Workflow

1. User logs in.
2. AI classifies the condition.
3. Factory makes session or referral.
4. Builder creates therapy plan.
5. Adapters handle data connection.
6. Containers manage main parts.
7. Classifiers check the case step by step.
8. Commands run based on result

Creational Patterns

Factory Method: Decouples decision logic from implementation by selecting the appropriate factory based on user state, allowing new session types without changing core logic.

Builder Pattern: Builds therapy workflows suited to user condition levels, keeping creation logic modular and separate from usage.

Structural Patterns

Adapter Pattern: Enables integration with third-party tools via standardized interfaces, easing future replacements.

Container Pattern: Divides the system into clear parts (e.g., sessions, reporting), improving debugging and scalability.

Behavioral Patterns

Chain of Responsibility: Breaks classification into steps (e.g., analysis, surveys), allowing fallback and easier updates.

Command Pattern: Abstracts actions like session starts and notifications, separating execution from triggers and simplifying changes

Chapter 6: Conclusion

6.1 Conclusion

In this project, we designed an AI-powered mental health assistant integrated into the Sehaty platform to support the classification and treatment of mental health conditions. We presented an overview of our solution. We defined the client context (the Ministry of Health via the Sehaty app), studied similar systems to our app, such as Youper, and defined a clear problem statement that guided our solution.

We analyzed requirements based on potential scenarios and prototypes. We defined clear functional and non-functional requirements for the system. These requirements ensured that the solution would be practical, secure, and scalable for actual use in a healthcare setting.

In Chapters 3 and 4, we modeled the system using use case diagrams, class diagrams, sequence diagrams, and a layered architecture model. We divided the architecture into four main layers: user interface, application, utility, and core—each with its own functionality.

6.2 Lessons Learnt

This project taught us the importance of teamwork, that one hand cannot clap alone, that each member has a key role in the project's success, and that constructive criticism contributes to the development and improvement of the project's quality.

References

- [1] I. Sommerville, "Software Engineering", 11th ed., Addison-Wesley, 2015.
- [2] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed., Pearson, 2022.
- [3] Ministry of Health, Saudi Arabia. (2024). Sehaty App [Online]. Available: <https://www.sehaty.sa>