**Sandra**

Innovative Mental Health Platform

**Senior Project**

**Prepared by**

Khaled Abd-Alhameed Alshagheer

Sireen Mahmoud Altakriti

**Supervised by**

Dr. Omar Hamdoun Eng. Areej Alhawash

**Semester of**

2023-2024

Abstract

This project produced web-based platform provides comprehensive mental health resources, aimed at empowering individuals seeking support. Users can access a library of informative articles curated by experts, engage in direct conversations with licensed psychotherapists, and interact with a supportive chatbot named "Sandra."

The platform now includes several enhanced features such as appointment booking, online conferences with electronic payment, complaint handling, notifications, and a detailed patient profile.

The platform leverages React for a dynamic user interface, Laravel to manage the backend infrastructure, and Python for scripting functionalities. This combination of technology facilitates a seamless and accessible experience for users seeking mental health guidance and resources.

Contents List

[Abstract 2](#_Toc171809794)

[Contents List 3](#_Toc171809795)

[List of Figures 5](#_Toc171809796)

[List of Tables 6](#_Toc171809797)

[List of Abbreviation 7](#_Toc171809798)

[Chapter One: (Introduction) 10](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809799)

[1.1. Motivation: 11](#_Toc171809800)

[1.2. Problem description: 11](#_Toc171809801)

[1.3. project objectives: 12](#_Toc171809802)

[Chapter Two : (Background) 13](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809803)

[2.1. Theoretical Underpinnings: 14](#_Toc171809804)

[2.2. Related Works: 14](#_Toc171809805)

[2.3. the proposed solution: 15](#_Toc171809806)

[Chapter Three : (Methodology) 19](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809807)

[3.1. Kanban: 20](#_Toc171809808)

[3.2. Kanban Stages in Our Project: 21](#_Toc171809809)

[3.3. Technologies Used: 27](#_Toc171809810)

[Chapter Four : (Analysis) 32](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809811)

[4.1. Target user: 33](#_Toc171809812)

[4.2. Requirement Gathering: 33](#_Toc171809813)

[4.3. Functional and non-functional requirements: 34](#_Toc171809814)

[4.4. Class Diagram : 37](#_Toc171809815)

[4.5. Use case Diagram : 38](#_Toc171809816)

[4.6. Use case Specification : 49](#_Toc171809817)

[4.7. Activity Diagram : 67](#_Toc171809818)

[4.8. Sequence Diagram : 71](#_Toc171809819)

[Chapter Five : (Interface and system design) 73](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809820)

[5.1. System architecture: 74](#_Toc171809821)

[5.2. Block Diagram : 76](#_Toc171809822)

[5.3. Entity Relationship Diagram (ERD) : 76](#_Toc171809823)

[5.4. Interface Design : 76](#_Toc171809824)

[Chapter Six : (Develop , Testing , Deploy) 77](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809825)

[6.1. Develop: 78](#_Toc171809826)

[6.2. Testing and Deploy: 83](#_Toc171809827)

[Chapter Seven : (Conclusion) 85](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809828)

[7.1. Conclusion : 86](#_Toc171809829)

[7.2. Future Work: 86](#_Toc171809830)

[References 89](#_Toc171809831)

List of Figures

[Figure 2.1 Psychology Today Logo 14](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809519)

[Figure 2.2 NAMI Logo 15](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809520)

[Figure 2.3 Koko Logo 15](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809521)

[Figure 3.1 Kanban Principles 21](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809522)

[Figure 3.2 Kanban Stages 22](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809523)

[Figure 3.3 Kanban board (Junior) 24](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809524)

[Figure 3.4 Kanban board (Senior) 26](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809525)

[Figure 4.1 Analysis Class Diagram 37](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809526)

[Figure 4.2 Admin Use Case Diagram 39](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809527)

[Figure 4.3 Psychotherapist Use Case Diagram 43](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809528)

[Figure 4.4 Patient Use Case Diagram 47](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809529)

[Figure 4.5 Secretary Use Case Diagram 49](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809530)

[Figure 4.6 Add Article Activity Diagram 67](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809531)

[Figure 4.7 Accept / Reject Article Activity Diagram 69](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809532)

[Figure 4.8 Add Schedule Activity Diagram 69](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809533)

[Figure 4.9 Book Appointment Activity Diagram 70](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809534)

[Figure 4.10 Start Chat Sequence Diagram 71](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809535)

[Figure 4.11 upload and verify certificate Sequence Diagram 72](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809536)

[Figure 6.1 GPT3.5 Transformer (Decoder-only) 80](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809537)

[Figure 6.2 Prompt Structure 80](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809538)

[Figure 6.3 Retriever Prompt Structure 81](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809539)

[Figure 6.4 Data processing 82](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809540)

[Figure 6.5 Query processing and execution 82](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809541)

List of Tables

[Table 4.1 Function Requirements 36](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809542)

[Table 4.2 Signup Usecase Specification 49](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809543)

[Table 4.3 Email Verification Usecase Specification 50](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809544)

[Table 4.4 Login Usecase Specification 51](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809545)

[Table 4.5 Show Articles Usecase Specification 52](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809546)

[Table 4.6 Like/Unlike Article Usecase Specification 53](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809547)

[Table 4.7 Add Article Usecase Specification 55](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809548)

[Table 4.8 Accept/reject article Usecase Specification 56](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809549)

[Table 4.9 Talk To Sandra Usecase Specification 58](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809550)

[Table 4.10 Add certificate Usecase Specification 59](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809551)

[Table 4.11 Approve Appointment Usecase Specification 60](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809552)

[Table 4.12 Delete Appointment Usecase Specification 61](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809553)

[Table 4.13 Mark Appointment as paid Usecase Specification 62](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809554)

[Table 4.14 Start Session Usecase Specification 63](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809555)

[Table 4.15 Print Patient Profile Usecase Specification 65](file:///E:\sireen\study\JUP\pdf&presentation\Sandra2.docx#_Toc171809556)

List of Abbreviation

|  |  |
| --- | --- |
| LLM | Large Language Model |
| React | Library In React To Design Interfaces |
| Laravel | Framework In PHP To Create API’s |
| PTSD | Post-Traumatic Stress Disorder |
| DSM-5 | Diagnostic and Statistical Manual of Mental Disorders |
| AI | Artificial Intelligence |
| NAMI | **National Alliance On Mental Illness** |
| koko | Chat Bot Designed To provide basic support mental health topics |
| WIP | Work In Progress |
| VS Code | Visual Studio Code |
| API | Application Program Interface |
| LangChain | Open source framework for building apps based on large language model |
| Kanban | Framework To Implement Agile And DevOps Software |
| Virtual DOM | Programming concept where a virtual representation of a UI is kept in memory synced |
| Python | An Interpreted , Object-Oriented , High-Level Programming Language |
| postman | API Platform For Building And Using APIs |
| MySql | Stands For Structured Query |
| Brainstorming | A creative thinking technique for coming up with a new ideas and solving problem |

Acknowledgements

We would like to express our deepest gratitude to our supervisors, Dr. Omar Hamdoun and Eng. Areej Alhawash, for their invaluable guidance, encouragement, and support throughout the development of this project. Their expertise and insights were instrumental in shaping the direction and success of our work.

We are also profoundly grateful to the Dean of the College, Dr. Moutasem Shafaamri , for his steadfast support and encouragement. Additionally, We would like to extend our heartfelt thanks to Dr. Hiyam Khaddam and Eng. Anas Mokhallalati for their unwavering support and valuable feedback, which greatly contributed to the completion of this project.

This project would not have been possible without the collective efforts and encouragement of these individuals, and we are deeply appreciative of their contributions and dedication. Thank you all for your belief in our work and for helping us achieve our goals.

Chapter One: (Introduction)

Motivation:

* + 1. Our target audience:
       1. Support Seekers: Individuals facing psychological challenges and seeking information and support.
       2. Healthcare Professionals: Psychotherapists and specialists in mental health who wish to publish articles and communicate with patients.
       3. Administrative Professionals: Administrative officials in need of website content management and physician documentation.
    2. Enhancing Diagnostic Accessibility and Reassurance:

Our project is motivated by the crucial importance of making psychological diagnoses readily accessible and providing answers that closely align with the final diagnosis. We aim to create a diagnostic environment that delivers accurate and easily understandable information, ensuring immediate reassurance for patients and enhancing the overall mental health care experience.

Problem description:

The field of psychology often presents a significant challenge for individuals seeking information and support. Complex terminology, varying levels of accessibility, and a lack of clarity can create barriers, preventing many from fully engaging with psychological concepts and resources. This poses a substantial problem, particularly for those facing mental health challenges and seeking reliable information.

project objectives:

The aim of our project is to enhance accessibility and clarity within the realm of psychology for individuals from all walks of life. Recognizing the complexity and sometimes inaccessible nature of psychological information, our initiative seeks to bridge the gap and make psychological resources readily available to a diverse audience. By fostering a user-friendly platform, we aspire to demystify the psychological world, making it more comprehensible and accessible for everyone, ultimately contributing to a more informed and empowered community.

Chapter Two : (Background)

Mental health concerns are prevalent globally, affecting individuals of all ages and backgrounds. However, accessing qualified care can be challenging due to limited resources, financial constraints, and social stigma. This project aims to bridge this gap by providing a comprehensive online platform for mental health support.



Theoretical Underpinnings:

Mental health disorders encompass a wide range of conditions, including anxiety, depression, bipolar disorder, and post-traumatic stress disorder (PTSD). These disorders can significantly impact individuals' lives, affecting their emotional well-being, relationships, and daily functioning.

Mental health diagnosis is a complex process often involving interviews, assessments, and standardized criteria like the DSM-5 (Diagnostic and Statistical Manual of Mental Disorders). Chatbots, powered by artificial intelligence (AI), are emerging as potential tools for mental health support. However, ethical considerations and limitations, such as the potential for misdiagnosis and lack of human empathy, necessitate cautious application.

Related Works:

* + 1. ****Psychology Today:****

A well-known directory designed to help people find licensed therapists in their area. Users can tailor their search with filters such as location, specialization, and insurance coverage.



Figure 2.1 Psychology Today Logo

* + 1. ****National Alliance on Mental Illness (NAMI):****

A leading non-profit organization offering a wealth of reliable information on mental health conditions. NAMI provides educational resources, support groups, and advocacy programs for individuals and families affected by mental illness.



Figure 2.2 NAMI Logo

* + 1. ****Koko:****

An AI-based chatbot accessible 24/7, designed to provide basic support and resources on mental health topics. Koko offers a way to engage with mental health concerns and may help reduce the stigma surrounding professional help-seeking.



Figure 2.3 Koko Logo

our project establishes itself as a unique and valuable resource for individuals navigating mental health concerns, offering a **one-stop platform** for information, potential direct professional support, and AI-powered assistance with ethical considerations in mind.

**the proposed solution:**

This project proposes a website that addresses the limitations of existing solutions, offering a single platform for comprehensive mental health support. Key features include:

* + 1. Verified psychologist directory:

Licensed professionals verify their credentials with the website administrator. User profiles showcase expertise and qualifications, facilitating informed choice.

* + 1. Credible mental health articles:

Articles are authored and verified by qualified psychologists, providing reliable information and promoting self-care practices.

* + 1. Optional chat feature:

Users can interact with licensed psychologists directly (if applicable) for professional guidance and support.

* + 1. Appointment Booking:

Users can conveniently book appointments with their preferred psychotherapists through the platform, streamlining the process of seeking professional help.

* + 1. Online Conference with Electronic Payment:

The platform supports scheduling and conducting online conferences with licensed psychologists, complete with secure electronic payment options for ease and convenience.

* + 1. Complaint Handling:

A dedicated feature allows users to submit complaints, ensuring their concerns are addressed promptly and effectively by the support team.

* + 1. **Notifications**:

Users receive timely notifications about their appointments, payments, and other important updates, keeping them informed and engaged.

* + 1. Patient Profile:

A detailed patient profile section maintains comprehensive information about each user, including psychotherapists' notes from every online conference. These notes are further summarized by AI to provide a clear and concise overview, ensuring continuity of care and personalized treatment plans.

* + 1. AI-powered chatbot:

This chatbot, built with fine-tuned large language models (LLMs) like ChatGPT and LangChain in Python, aims to offer basic support, mental health information, and resources, be trained with the DSM-5 to avoid potential hallucinations inherent in OpenAI models, clearly state its limitations, emphasizing the importance of seeking professional help from verified psychologists.

Chapter Three : (Methodology)

A methodology is a framework that provides a set of principles and practices for managing and completing a software project. It helps to ensure that the project is completed on time, within budget, and to the required quality standards.



Kanban:

We've opted for the **Kanban methodology**, a member of the Agile family, to manage the development of our online mental health support platform. Kanban offers several advantages that align well with our project's needs: (adopting-agile-methodology-steps, n.d.)

* + 1. Visualization and Focus:

Utilizing a Kanban board provides a visual representation of the workflow, allowing us to track progress, identify bottlenecks, and prioritize tasks effectively. This transparency fosters clear communication within our small team.

* + 1. Flexibility and Adaptability:

Kanban readily accommodates changes and evolving requirements. New tasks can be effortlessly added and prioritized throughout the process, ensuring the project remains responsive to shifting needs.

* + 1. Continuous Flow:

Kanban emphasizes a smooth flow of work, aiming to avoid large batches and ensure steady progress. This approach suits our project well, enabling us to deliver functionalities incrementally while receiving continuous feedback for improvement.

* + 1. Efficiency for Small Teams:

Kanban is well-suited for smaller teams like ours, as it emphasizes collaboration, transparency, and prioritizing tasks based on capacity. This allows us to efficiently manage workload and deliver value through continuous progress.

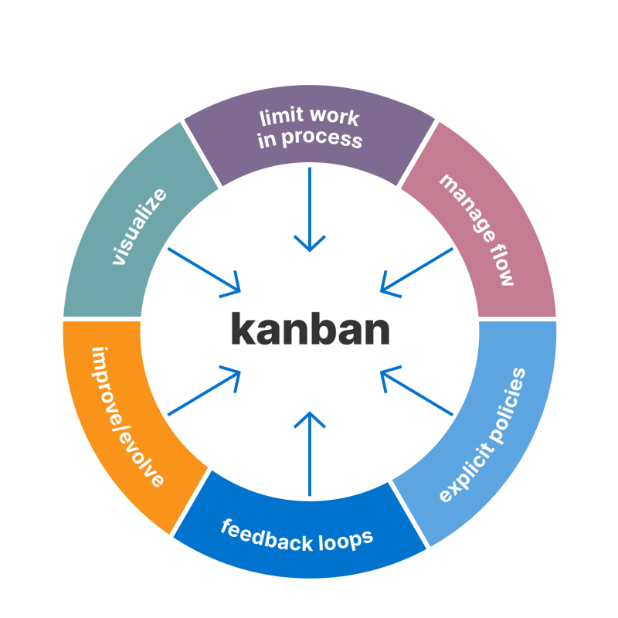


Figure 3.1 Kanban Principles

Kanban Stages in Our Project:

We'll implement Kanban through a visual board with three primary stages:

* + 1. To Do:

This stage houses all identified tasks and functionalities, each represented by a card containing details like description, priority, and estimated effort.

* + 1. In Progress:

This stage holds tasks currently being worked on by team members, with a Work in Progress (WIP) limit set to prevent overloading and ensure smooth workflow. Daily stand-up meetings facilitate discussions about progress and potential roadblocks within this stage.

* + 1. Done:

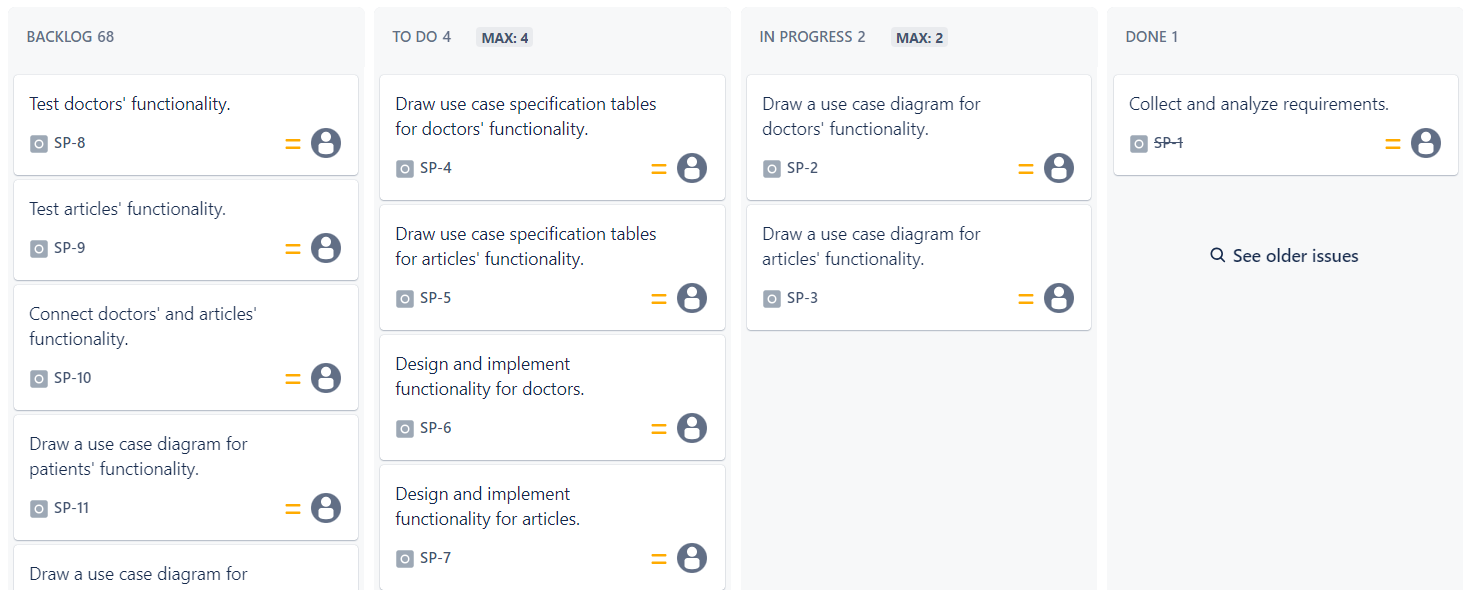
This stage showcases completed and tested tasks ready for review or integration. Cards move here upon successful completion and verification.

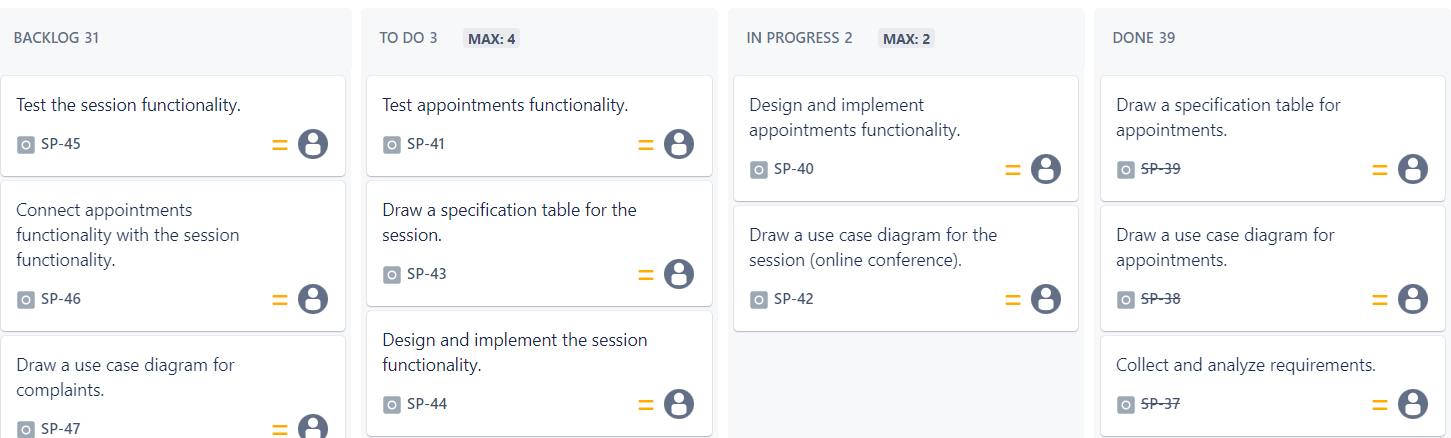
By adopting Kanban, we aim to manage our project effectively, promoting transparency, collaboration, and continuous improvement towards a successful outcome.



Figure 3.2 Kanban Stages

The following figures depict kanban board :





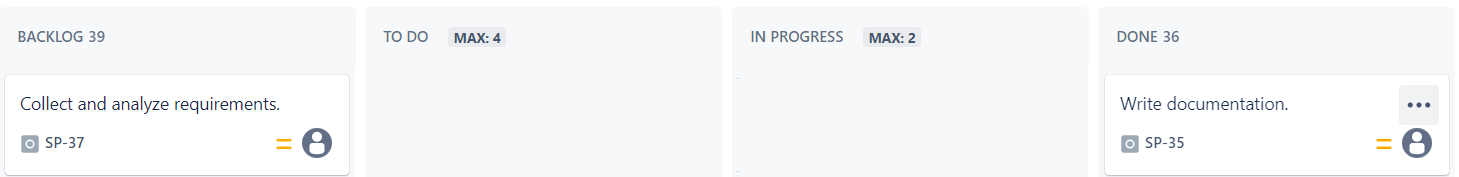
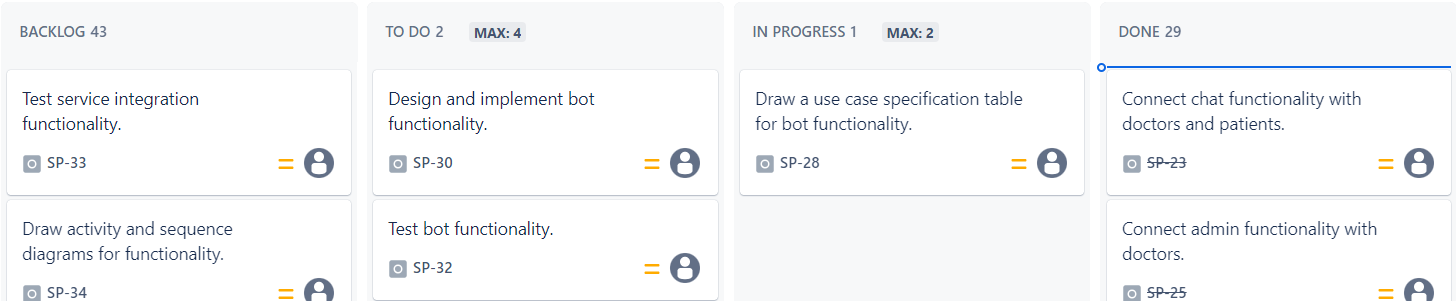
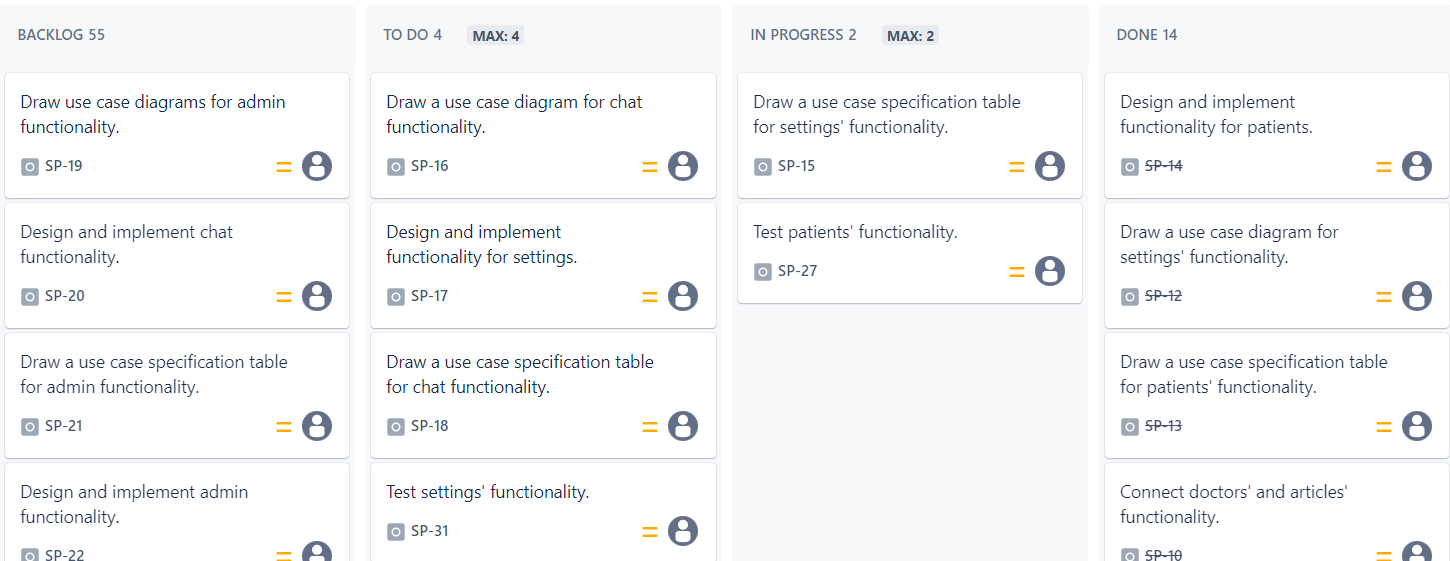


Figure 3.3 Kanban board (Junior)

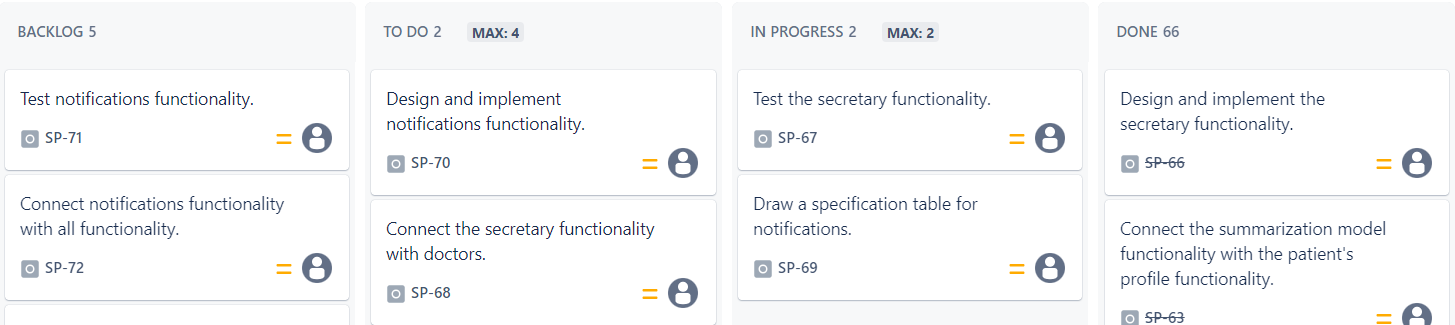
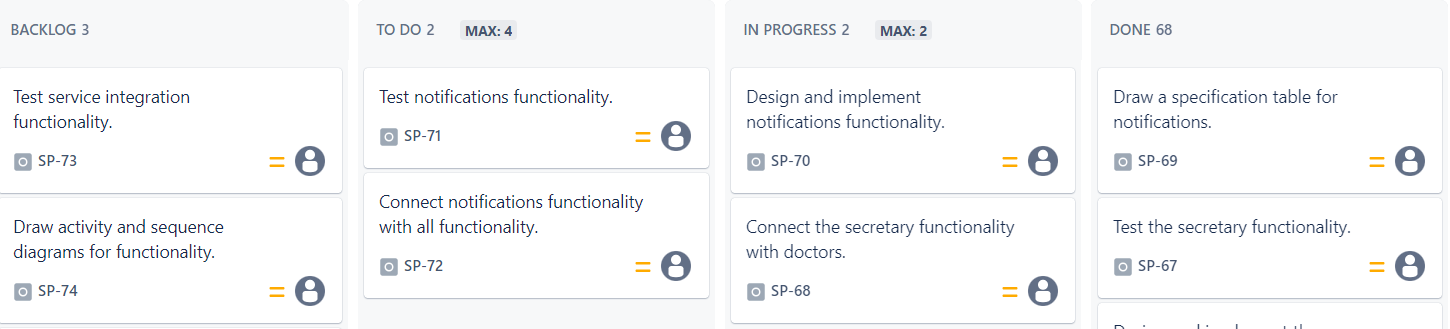
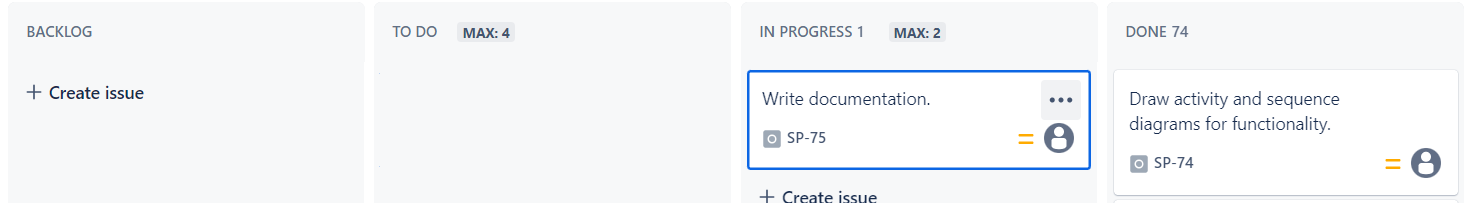
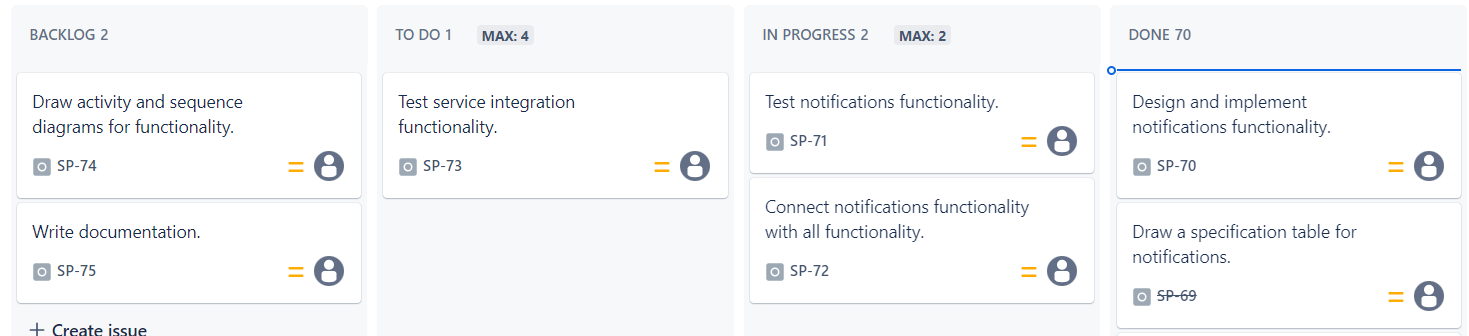
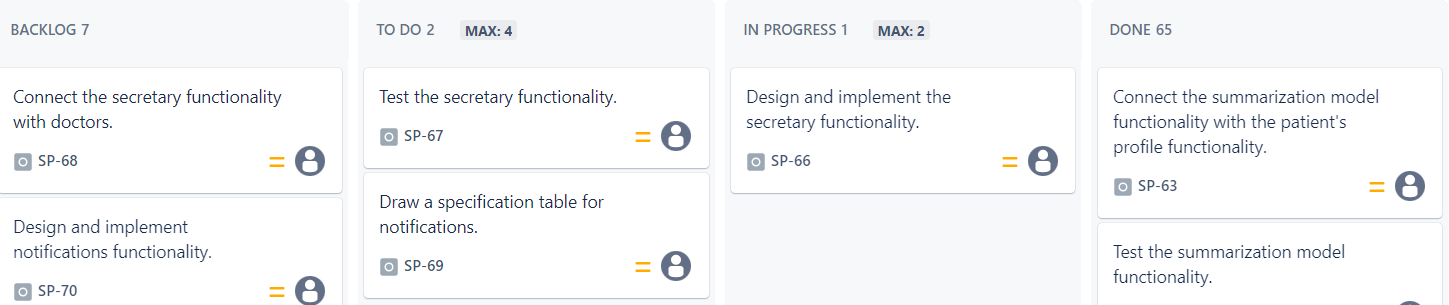
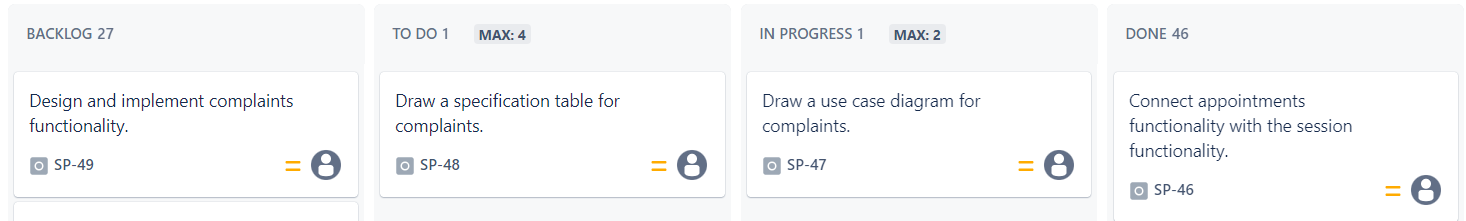
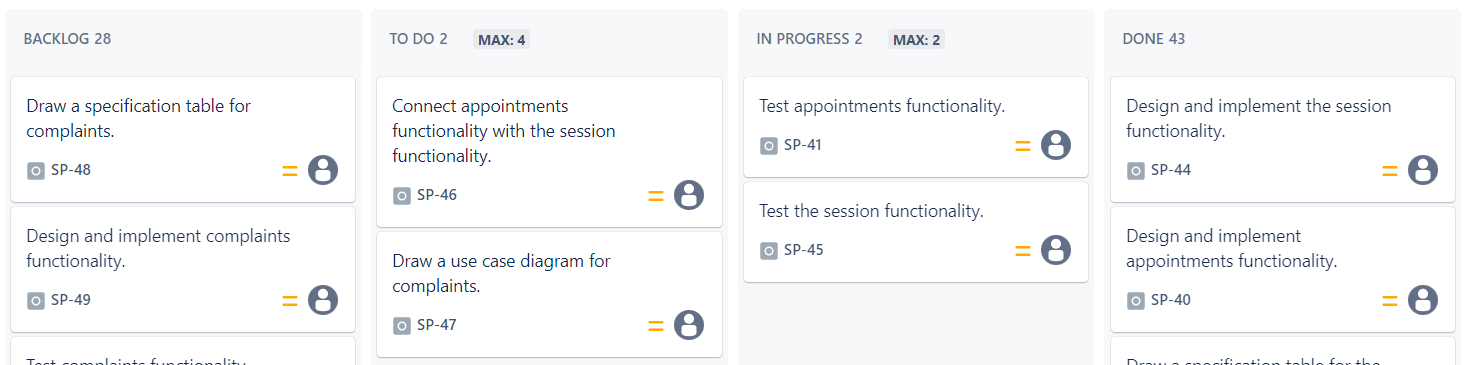
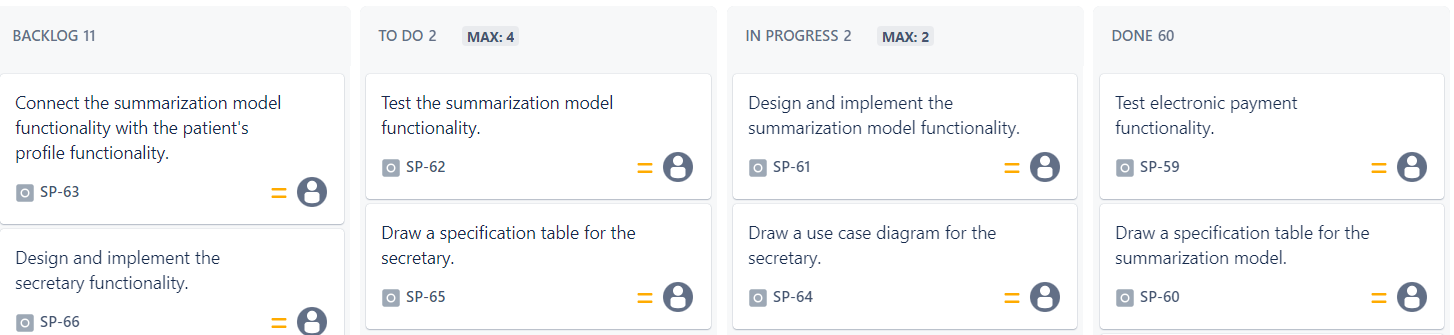
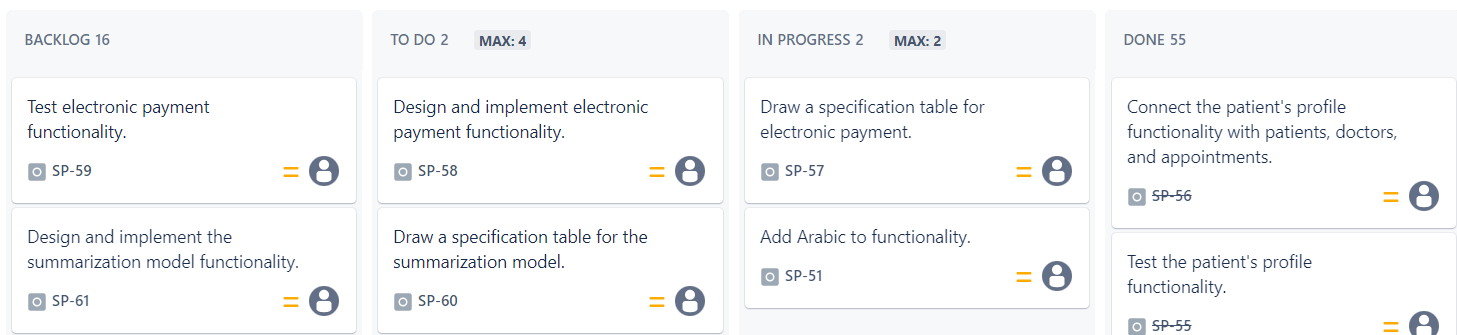
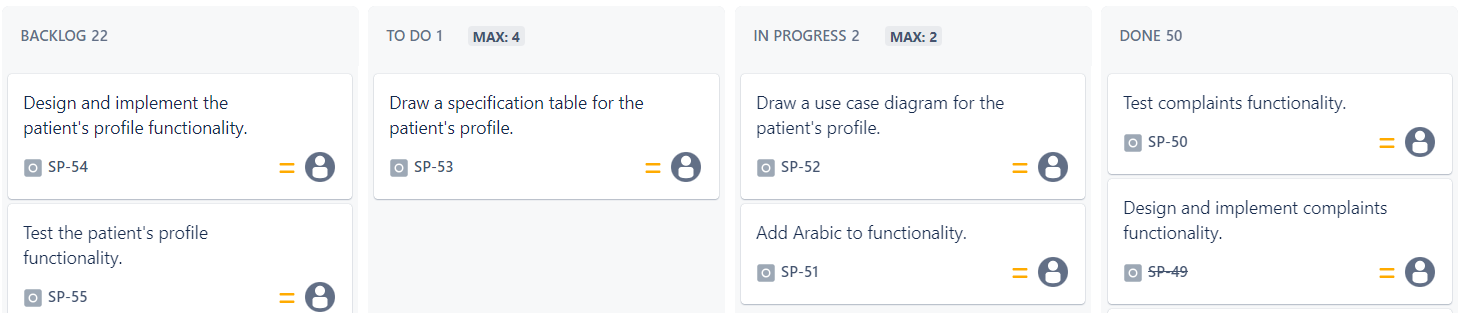
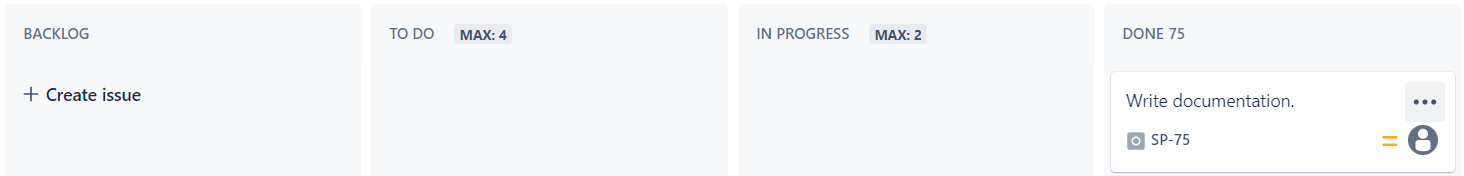


Figure 3.4 Kanban board (Senior)



Technologies Used:

* + 1. Visual Studio Code (VS Code):

Visual Studio Code (VS Code) is a widely used source code editor that has gained popularity among developers for its versatility and powerful features. Developed by Microsoft, VS Code is a free and open-source tool that supports various programming languages, making it a preferred choice for a diverse range of projects.

One of the key strengths of VS Code is its rich set of extensions, which allows users to customize and enhance their development environment according to their needs. These extensions cover a wide spectrum of functionalities, including language support, debugging, version control, and project management.

* + 1. Laravel:

a PHP web application framework, serves as the backbone of our project, managing the backend operations with efficiency and elegance. Laravel's expressive syntax and built-in features have streamlined the development of robust APIs and ensured smooth communication between the frontend and backend. Handling routing, middleware, and database interactions, Laravel facilitates the seamless flow of data and operations. Its strong emphasis on security and scalability makes it an ideal choice for creating a reliable and high-performance backend infrastructure, contributing to the overall stability and functionality of the project. (deployment, n.d.)

* + 1. React:

a JavaScript library developed by Facebook, has been employed to craft an engaging and responsive frontend for our project. With a component-based architecture, React enables the creation of dynamic user interfaces. The virtual DOM and efficient rendering empower the frontend to quickly respond to user interactions. Through React, we have developed an intuitive and visually appealing user interface that ensures a seamless and interactive experience for individuals navigating the platform. Its modular structure allows for the easy integration of new features and enhancements to optimize user engagement.

* + 1. MySQL:

an open-source relational database management system, is employed to manage and store the project's data securely and efficiently. With its proven reliability and scalability, MySQL serves as the foundation for our project's database, handling user information, chat logs, and other relevant data. Its robust querying capabilities and transactional support ensure the integrity of the data, providing a solid foundation for the project's functionality. MySQL's compatibility with Laravel facilitates seamless integration, allowing for efficient database management and retrieval of information.

* + 1. Flask Framework:

Flask, a lightweight and versatile web framework for Python, serves as the backbone behind our project's AI-powered chatbot and note summarization functionalities. Known for its simplicity and flexibility, Flask provides the necessary structure to develop robust web applications while integrating seamlessly with Python's extensive libraries for natural language processing (NLP), machine learning, and deep learning tasks.

With Flask, we have successfully implemented a sophisticated AI-powered chatbot tailored for mental health assessment and support. The chatbot analyzes user input, identifies potential psychological disorders, and offers initial guidance to promote early intervention and personalized support within the mental health realm. Additionally, Flask facilitates the automatic summarization of psychotherapists' notes after each online conference, ensuring concise information is added to the patient's profile for ongoing care and treatment.

By leveraging Flask's capabilities, our project delivers a responsive and efficient platform that enhances the accessibility and effectiveness of mental health care services, empowering users to manage their well-being proactively.

* + 1. LangChain :

LangChain, a specialized framework built on Python, serves as a pivotal component in our project's AI capabilities, particularly in enhancing mental health assessment and support. Developed to harness the power of language models effectively, LangChain integrates seamlessly with Python's libraries for natural language processing (NLP), machine learning, and deep learning tasks.

Through LangChain, we have developed advanced functionalities within our AI-powered chatbot. This includes sophisticated algorithms that analyze user input, identify potential psychological disorders, and provide initial guidance. LangChain's framework allows for precise customization and optimization, ensuring our chatbot delivers accurate assessments and personalized recommendations to users seeking mental health support.

Moreover, LangChain facilitates the automatic summarization of psychotherapists' notes after each online conference. This capability ensures that clear and concise information is efficiently integrated into the patient's profile, enhancing continuity of care and treatment planning.

By leveraging LangChain's capabilities, our project not only enhances the accessibility of mental health resources but also ensures that users receive tailored support that meets their individual needs effectively.

* + 1. GPT-3.5:

GPT-3.5 is a cutting-edge language model developed by OpenAI, renowned for its advanced natural language processing capabilities. It powers our project's AI-driven features, enhancing interactive dialogue and automating the summarization of psychotherapists' notes after online conferences. GPT-3.5's ability to understand context and generate human-like text ensures personalized and effective mental health support for users, facilitating seamless communication and continuity of care.

* + 1. ZegoCloud:

ZEGOCLOUD is employed for managing and conducting online conferences. This platform offers a reliable and high-quality video conferencing solution that is integrated into our project to facilitate virtual consultations between users and licensed psychologists. ZEGOCLOUD ensures smooth and secure communication, enhancing the accessibility and convenience of mental health support through our platform.

* + 1. Postman:

a widely-used API development and testing tool, plays a critical role in ensuring the reliability and functionality of our project's APIs. Utilized extensively during the development and testing phases, Postman allows for thorough testing of API endpoints, validating their responses and interactions. Through Postman's user-friendly interface, we conduct comprehensive tests to verify the correct functioning of our APIs, ensuring a smooth and secure flow of information between the frontend, backend, and the chatbot. Postman's robust testing features contribute to the overall quality assurance of our project, identifying and resolving potential issues before deployment.

Chapter Four : (Analysis)



Target user:

our project is designed to cater to users seeking psychological support by providing access to relevant articles, fostering community engagement through reactions and discussions, connecting users with mental health professionals, and offering a supportive environment with features like liking psychotherapists and a diagnostic chat bot.

Requirement Gathering:

The techniques used for requirement gathering were brainstorming sessions and interviews with our supervisors.

* + 1. ****Brainstorming:****

In the initial stages of our project development, my partner and I engaged in dynamic brainstorming sessions. These sessions were aimed at generating innovative ideas, features, and functionalities for our project. Through collaborative discussions, we explored various perspectives and creatively envisioned the scope of the project. The brainstorming sessions proved instrumental in inspiring creative solutions and shaping the overall direction of our project.

* + 1. ****Interviews:****

In addition to brainstorming, we conducted insightful interviews with our supervisor. These one-on-one interactions were focused on delving into our supervisor's expectations, preferences, and specific requirements for the project. The interviews provided us with valuable insights, ensuring a clear understanding of our supervisor's vision. These interactions were crucial in aligning our project goals with the expectations outlined by our supervisor, setting a solid foundation for the development process and our project report.

Functional and non-functional requirements:

This project encompasses a comprehensive mental health support platform designed to cater to the needs of psychotherapists, patients, secretaries, and administrators. Key functionalities include user authentication, article management, psychotherapist and patient profiles, appointment scheduling, secure chat features, and conference management. Psychotherapists can manage articles, certificates, schedules, and patient interactions, while patients have access to personalized profiles, appointment bookings, and secure communication with psychotherapists. Secretaries assist with schedule management and appointment logistics, and administrators oversee certificate verification, article approvals, and user management. Non-functional requirements prioritize usability and security to ensure an effective and safe user experience.

* + 1. Functional requirements:

|  |  |
| --- | --- |
| * + - 1. ****General User****: | Login - Signup - Show articles - Show all psychotherapists - Show psychotherapist profile - Show article content - Search - Change language |
| * + - 1. Psychotherapist: | Add articles - Remove articles - Show pending articles - Accept/reject articles - Show my articles - Show all chats - Show chat content - Remove chat - Add certificate - Add schedule - Edit schedule - Add banks account - Edit banks account - Add conference information - Edit conference information - Book new appointments - Show all appointments - Show appointment information - Search for appointment - Delete appointments - Approve appointment - Mark appointment as paid - Start conference - End conference - Break conference - Add conference notes - Add complaints - Delete complaints - Show all complaints - Show all patients - Show patient profile - Show patient banks account - Print patient information as PDF - Start chat - Add secretary account - Edit secretary account information - Show all notifications - Read notification - Delete account - Show and edit profile information - Show and edit account information - Change password - Contact us |
| * + - 1. Secretary: | Add schedule - Edit schedule - Add banks account - Add conference information - Book new appointments - Show all appointments - Show appointment information - Search for appointment - Delete appointments - Approve appointment - Mark appointment as paid - Add complaints - Delete complaints - Show all complaints - Show all patients - Show patient profile - Show patient banks account - Print patient information as PDF - Show all notifications - Read notification |
| * + - 1. Patient: | Like/unlike article - Report article - Like psychotherapist - Start chat - Remove chat - Show all chats - Talk to Sandra - Show chat content - Add banks account - Edit banks account - Add personal information - Edit personal information - Book new appointments - Show all appointments - Show appointment information - Search for appointment - Delete appointments - Start conference - End conference - Break conference - Add complaints - Delete complaints - Show all complaints - Show profile - Print profile as PDF - Show all notifications - Read notification - Delete account - Change password - Show and edit account information - Contact us |
| * + - 1. Admin: | Accept certificate verification - Reject certificate verification - Delete psychotherapist - Show pending articles - Accept/reject articles - Show all appointments - Show appointment information - Search for appointment - Show all complaints - Mark complaint as managed - Show all notifications - Read notification - Change password - Contact us |

Table 4.1 Function Requirements

* + 1. Non-functional requirements:
       1. Usability
       2. Security
       3. Scalability
       4. Flexibility

Class Diagram :

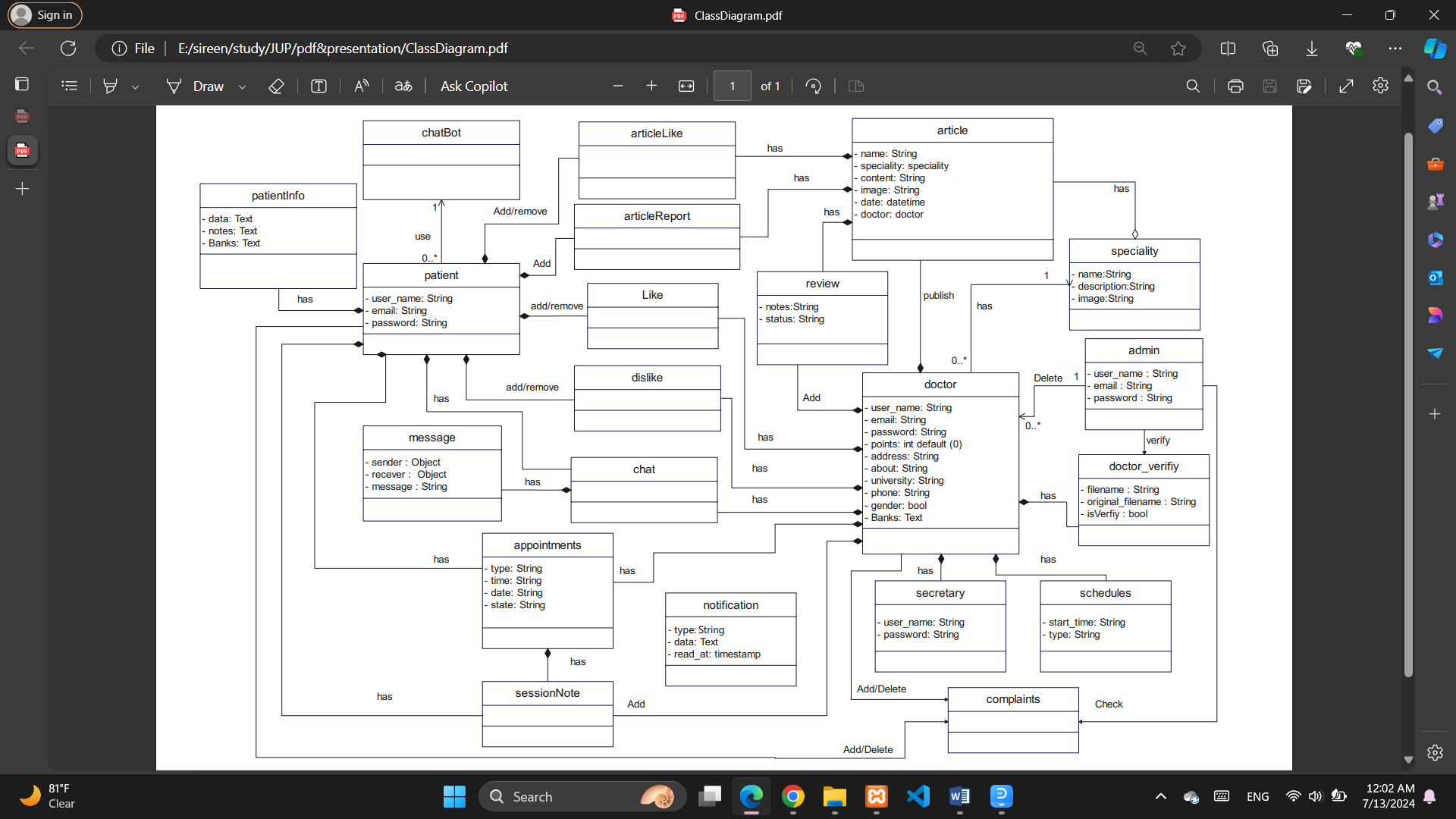
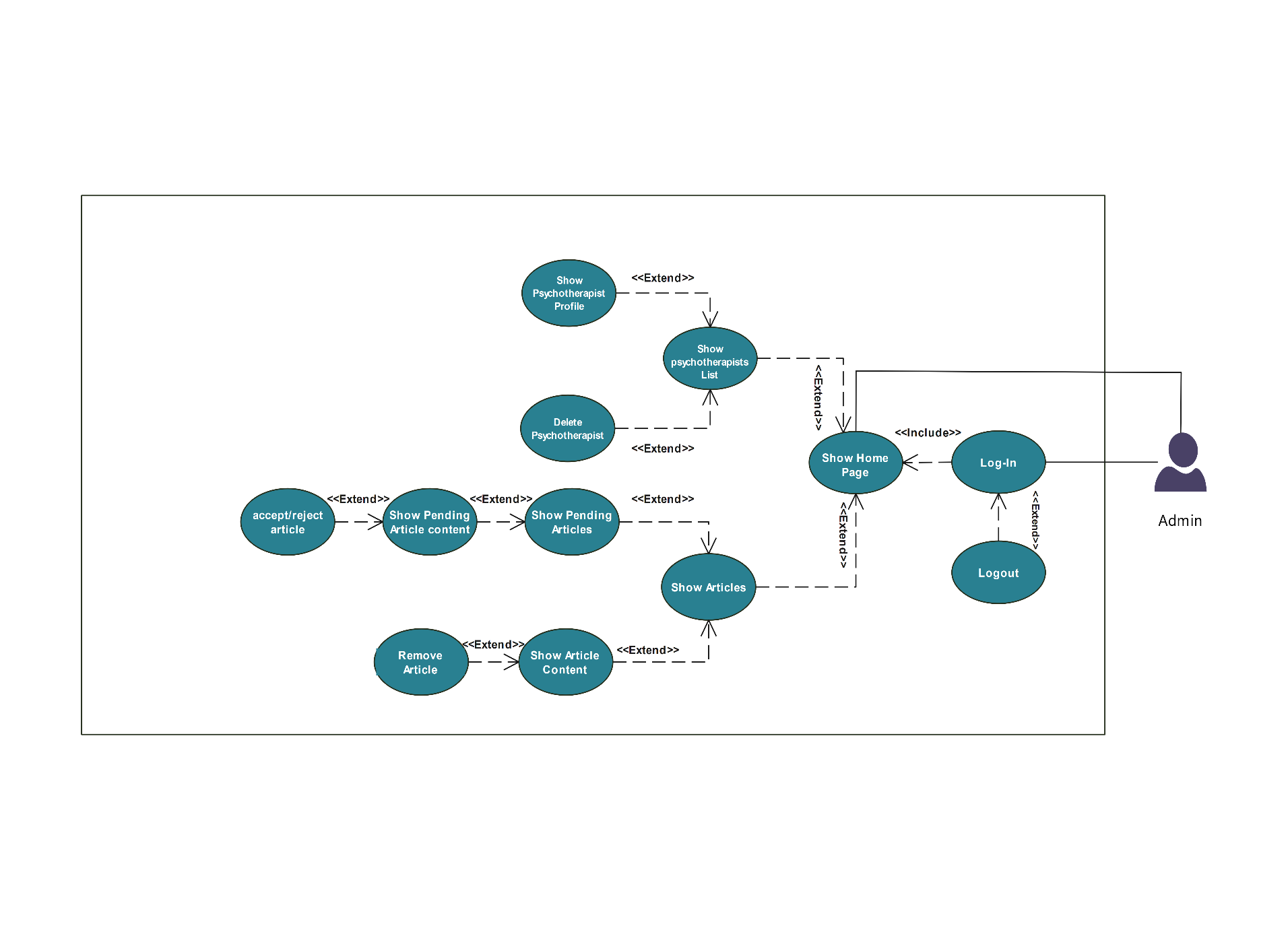
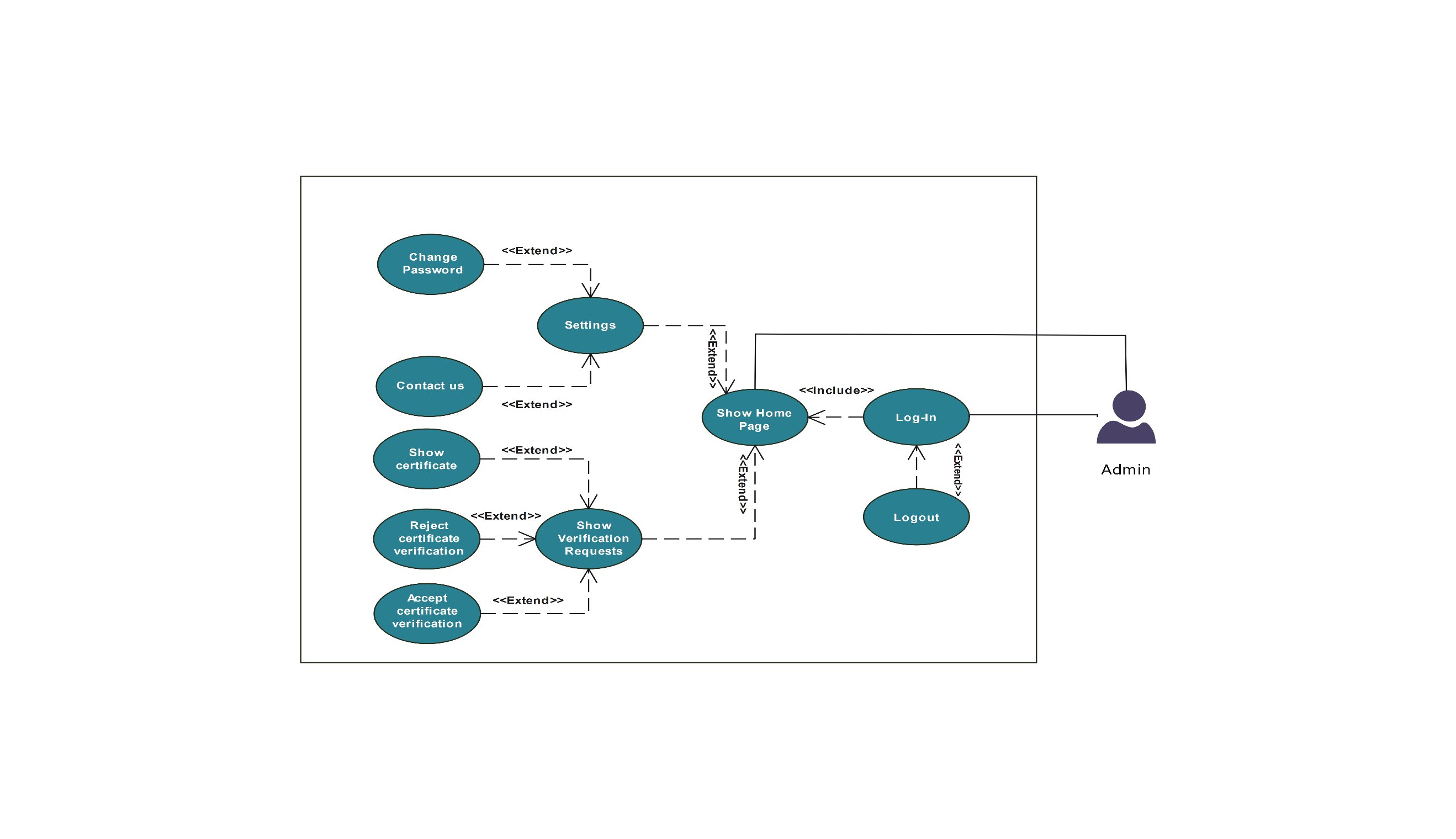


Figure 4.1 Analysis Class Diagram

Use case Diagram :





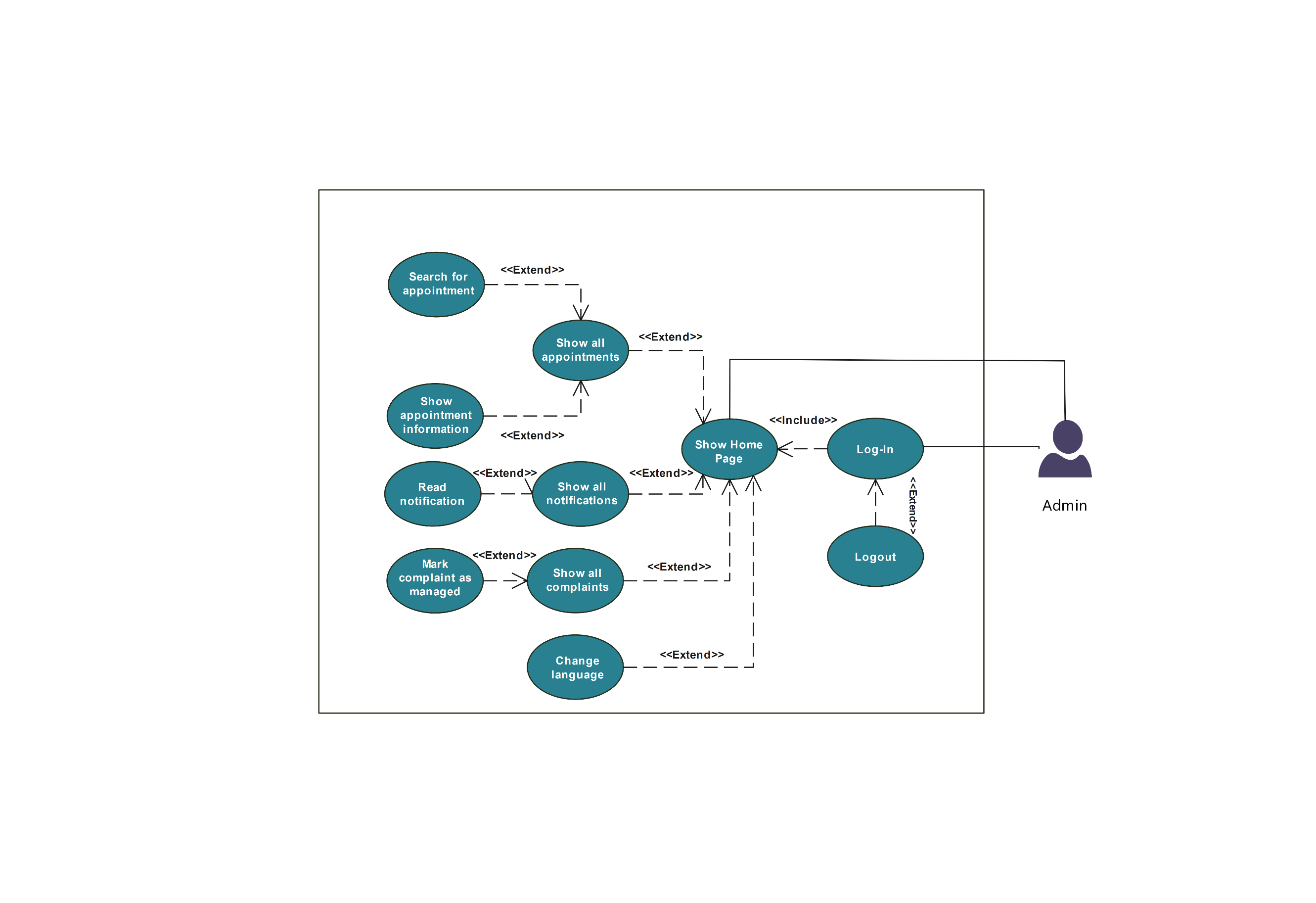
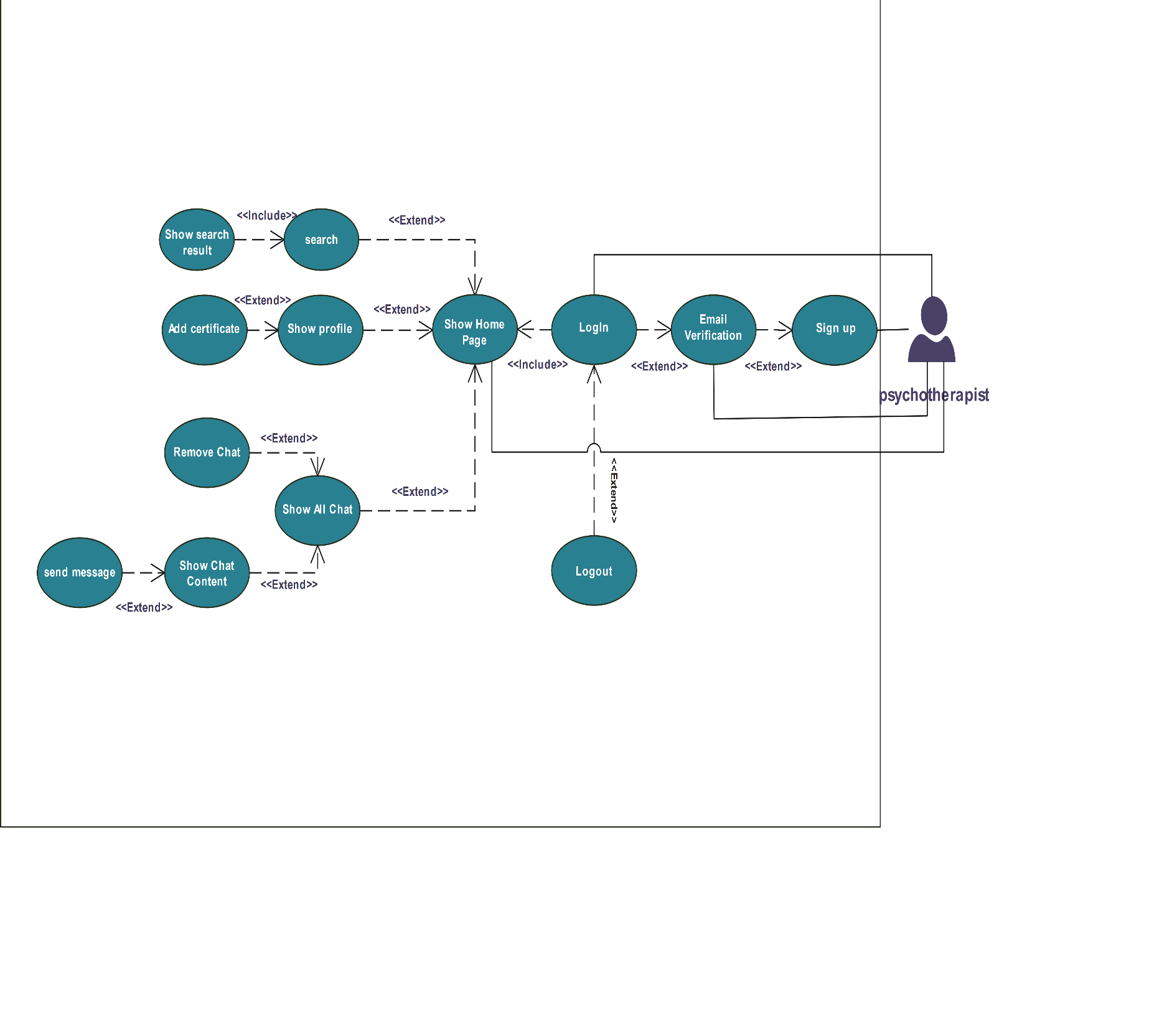
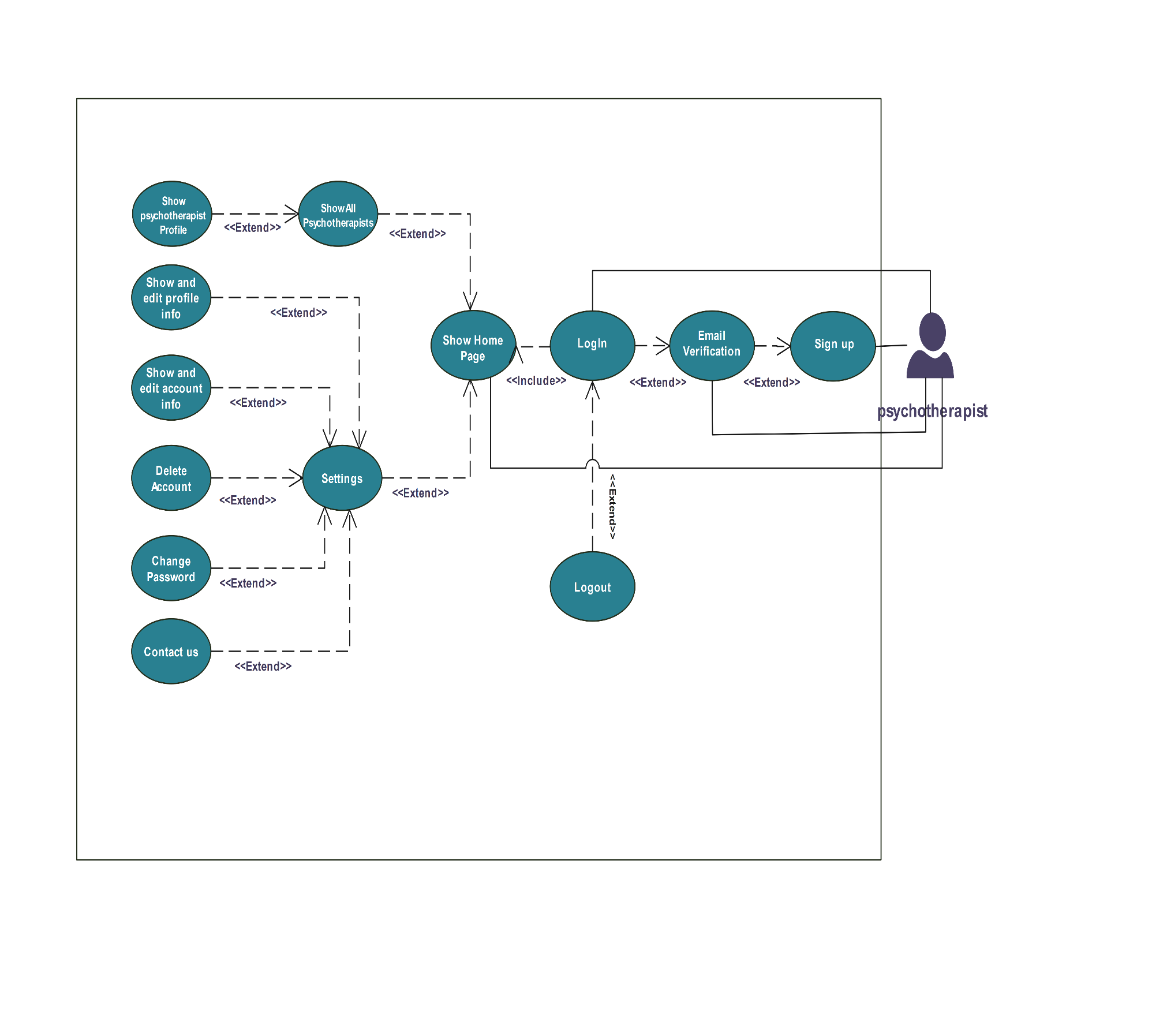
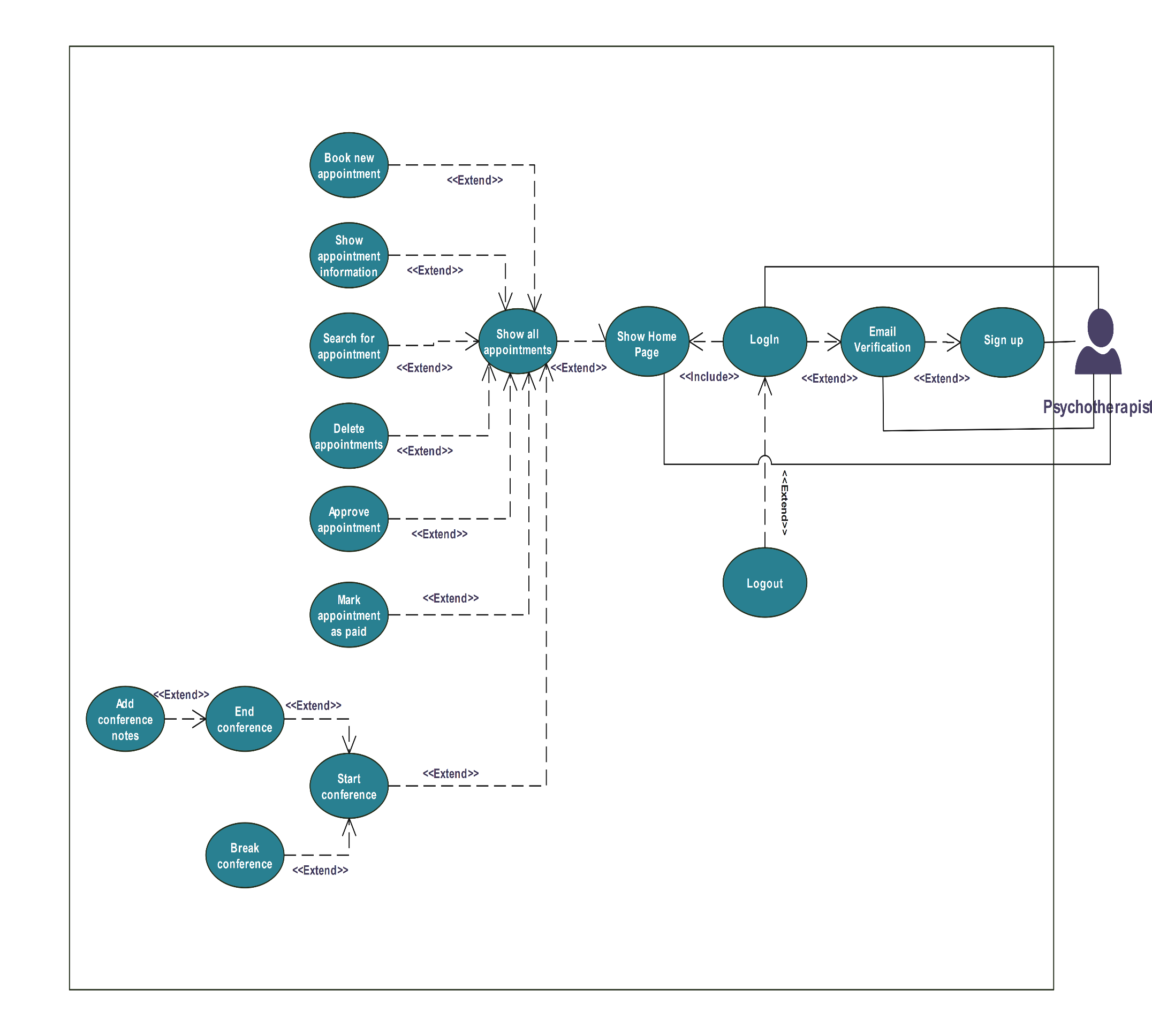


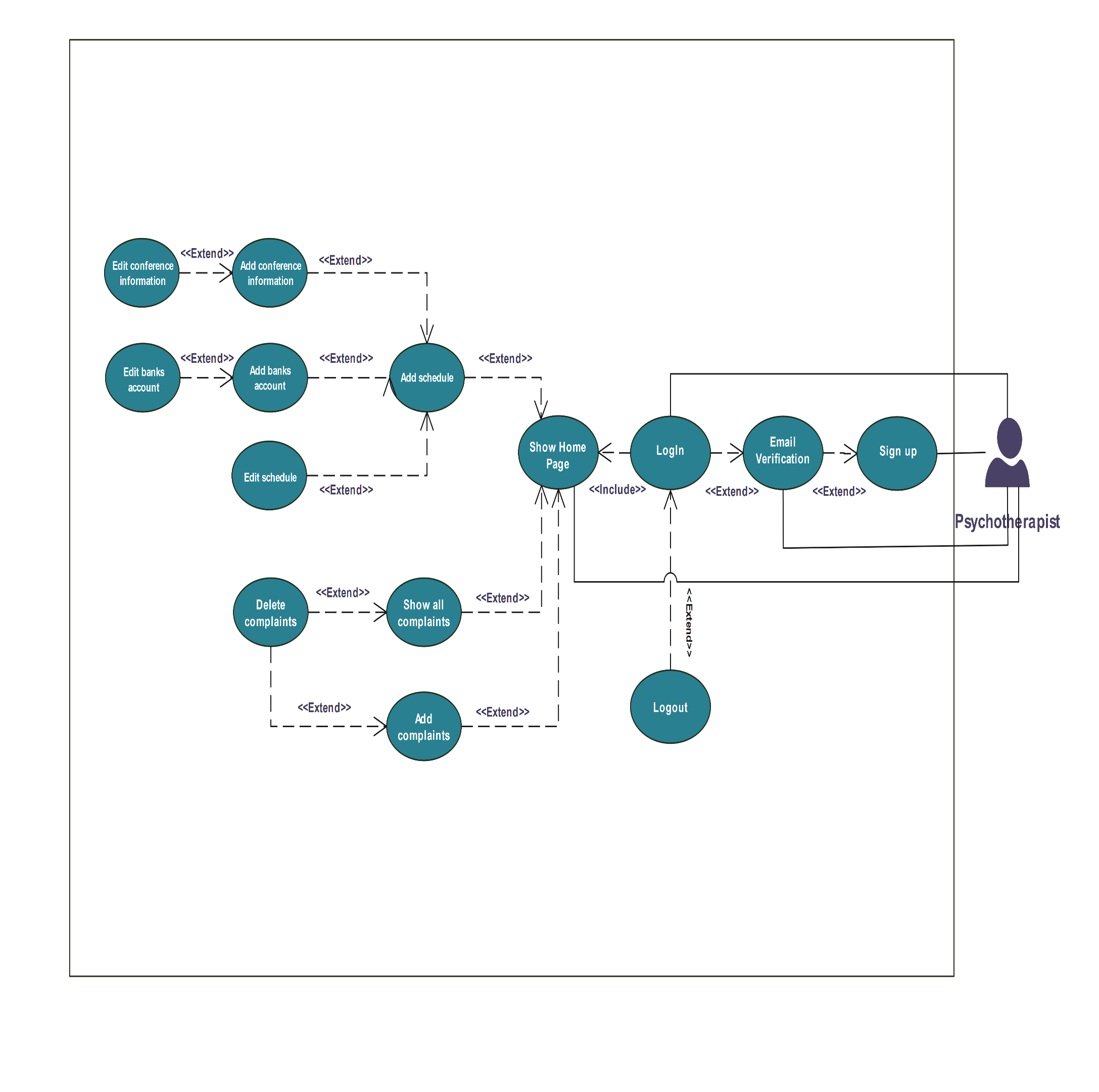
Figure 4.2 Admin Use Case Diagram











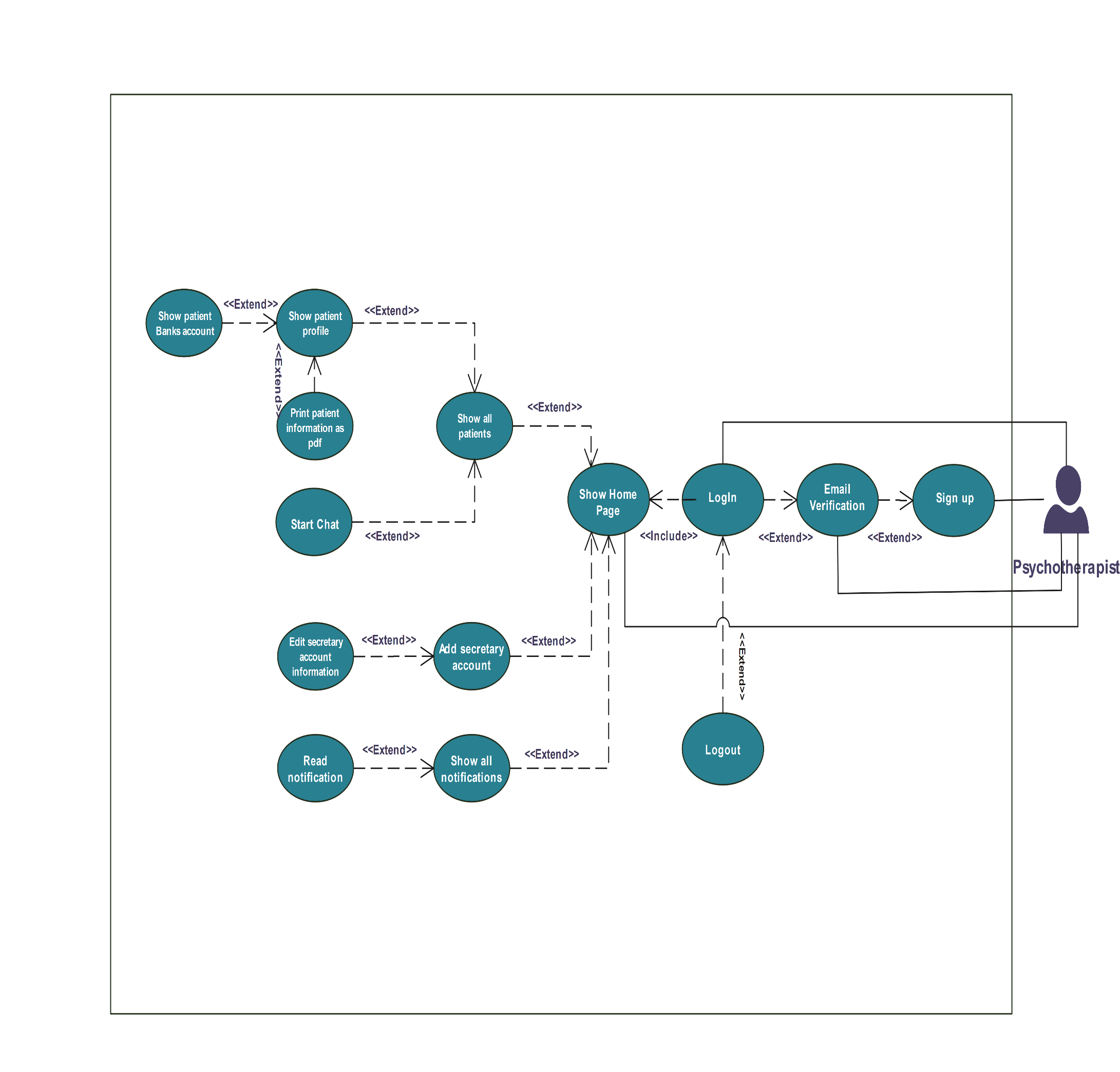
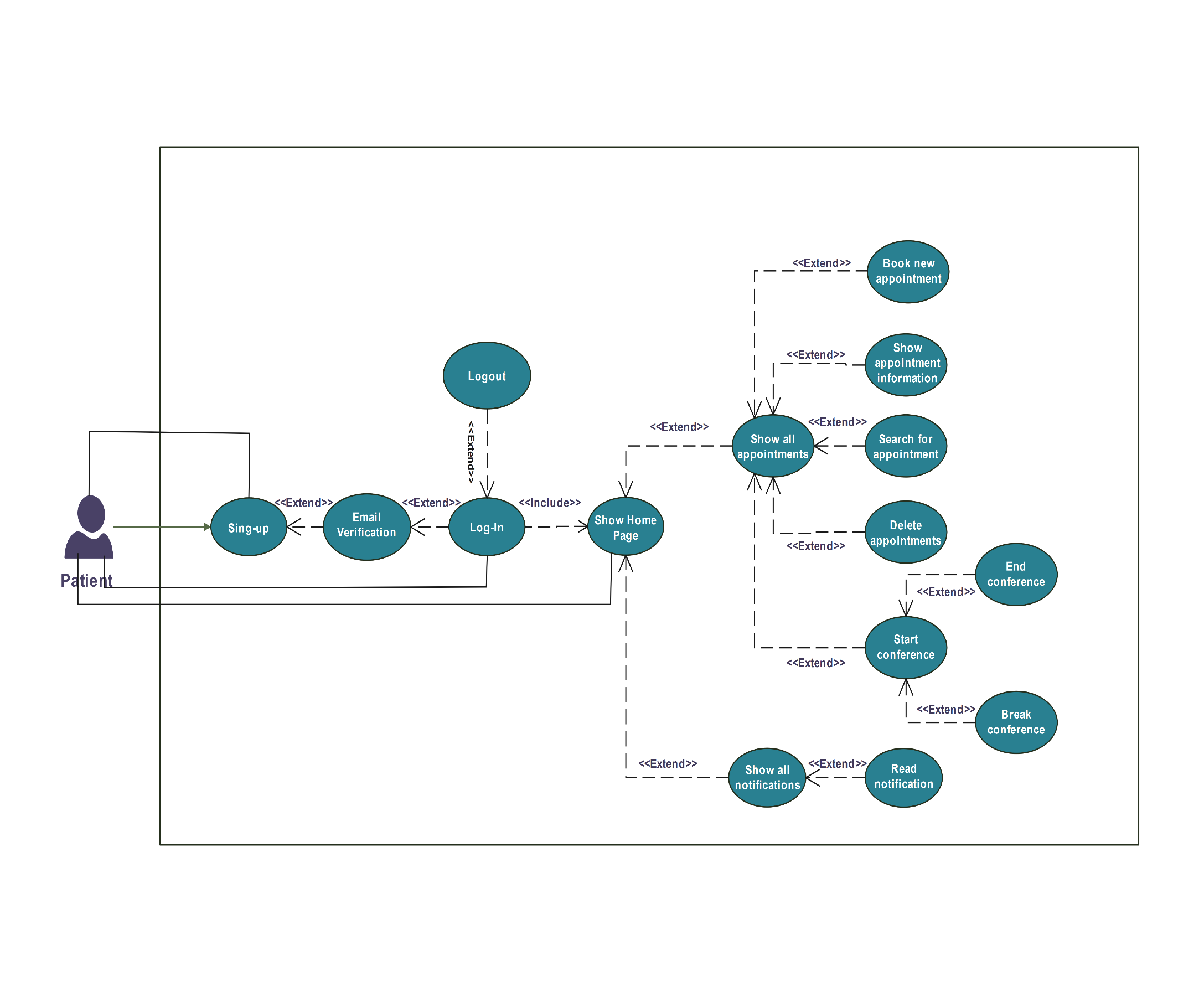
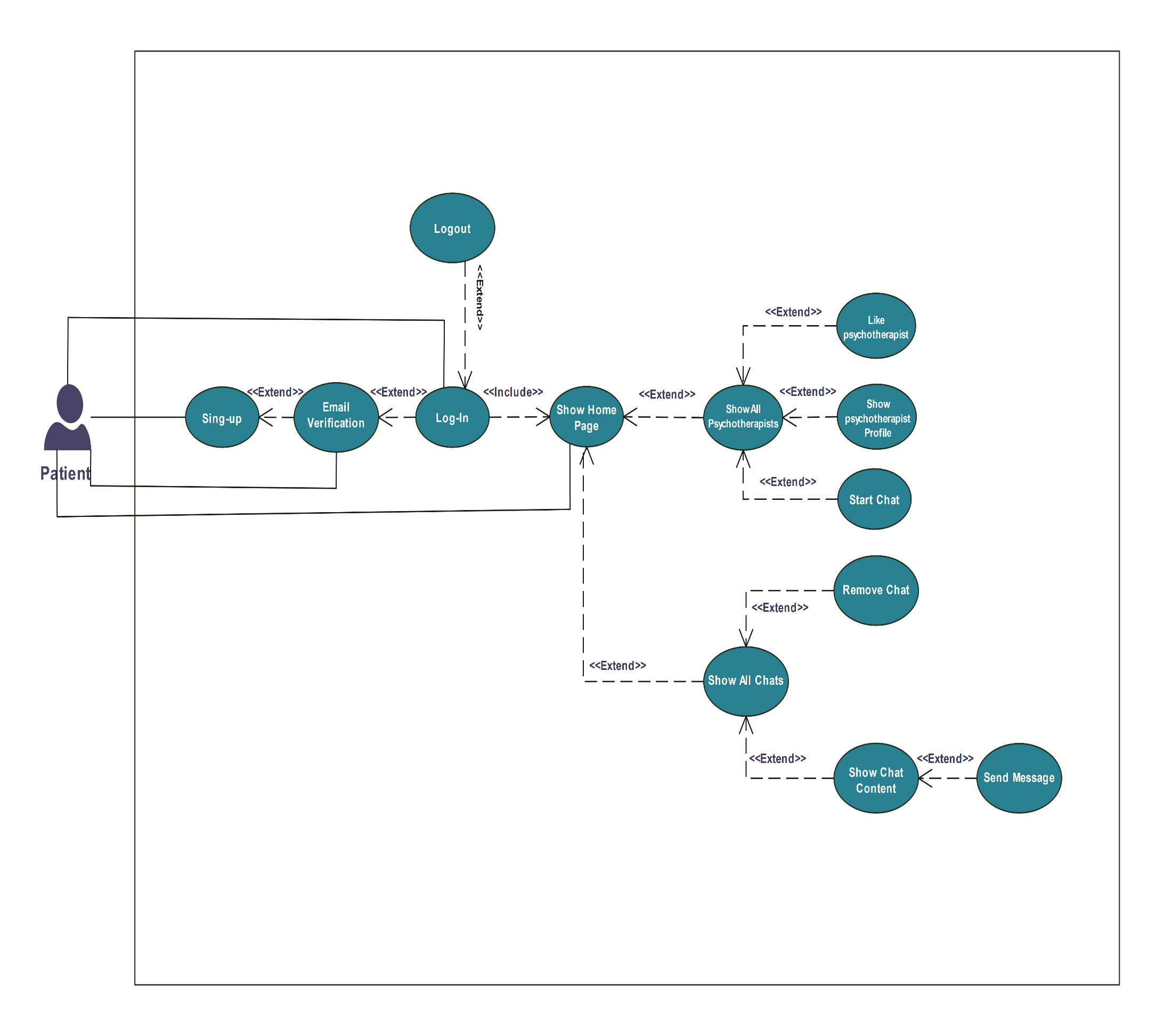
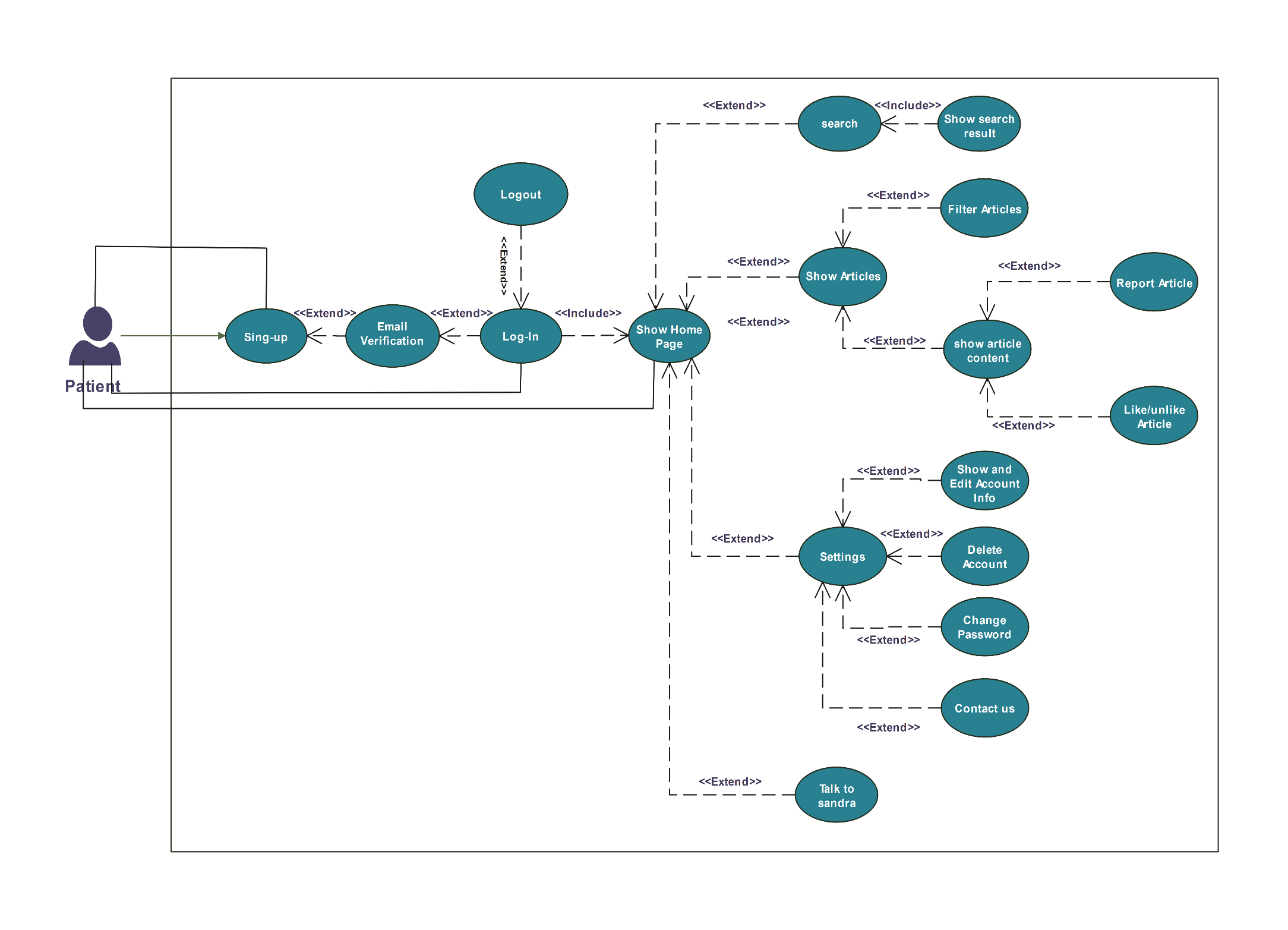


Figure 4.3 Psychotherapist Use Case Diagram







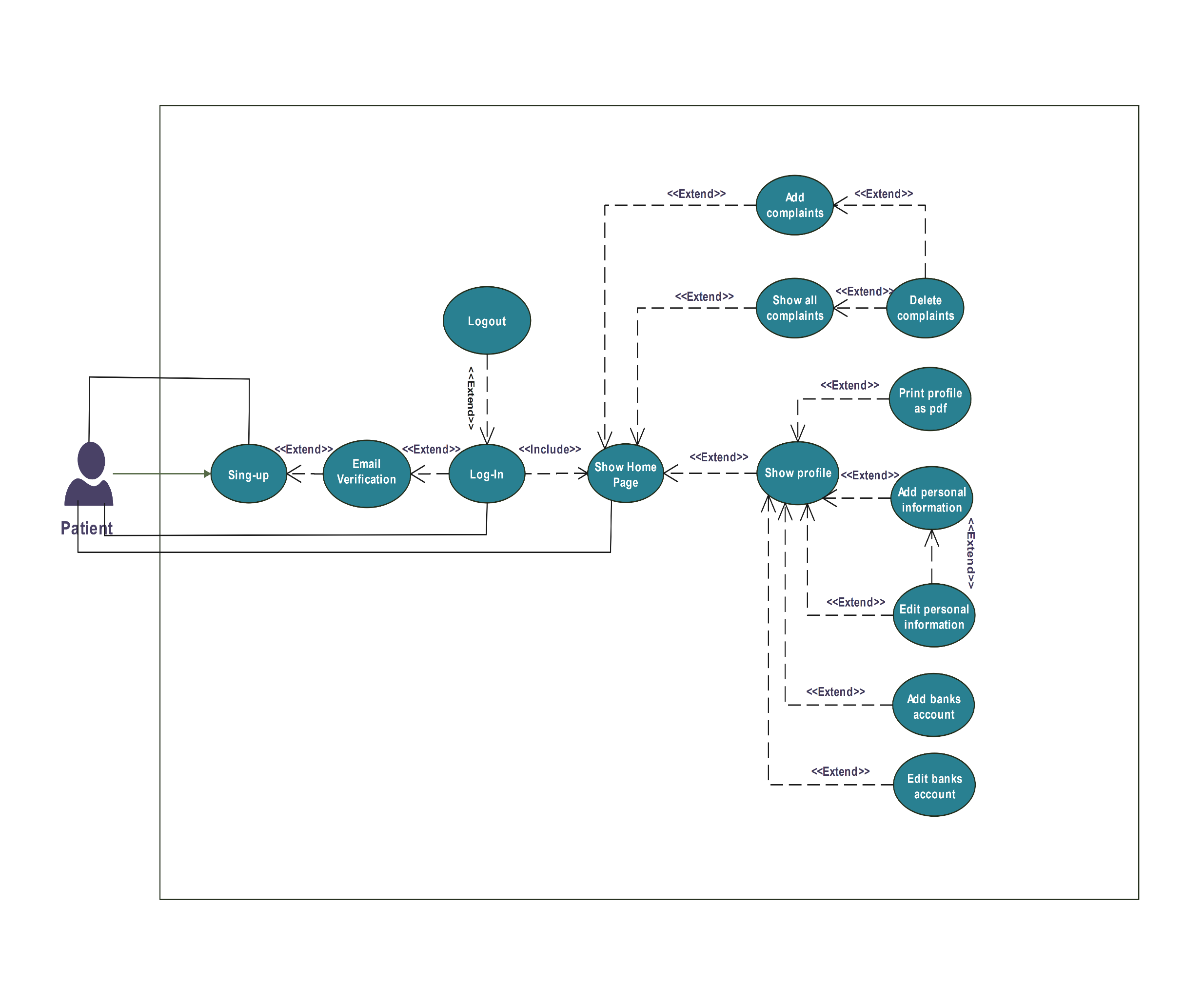
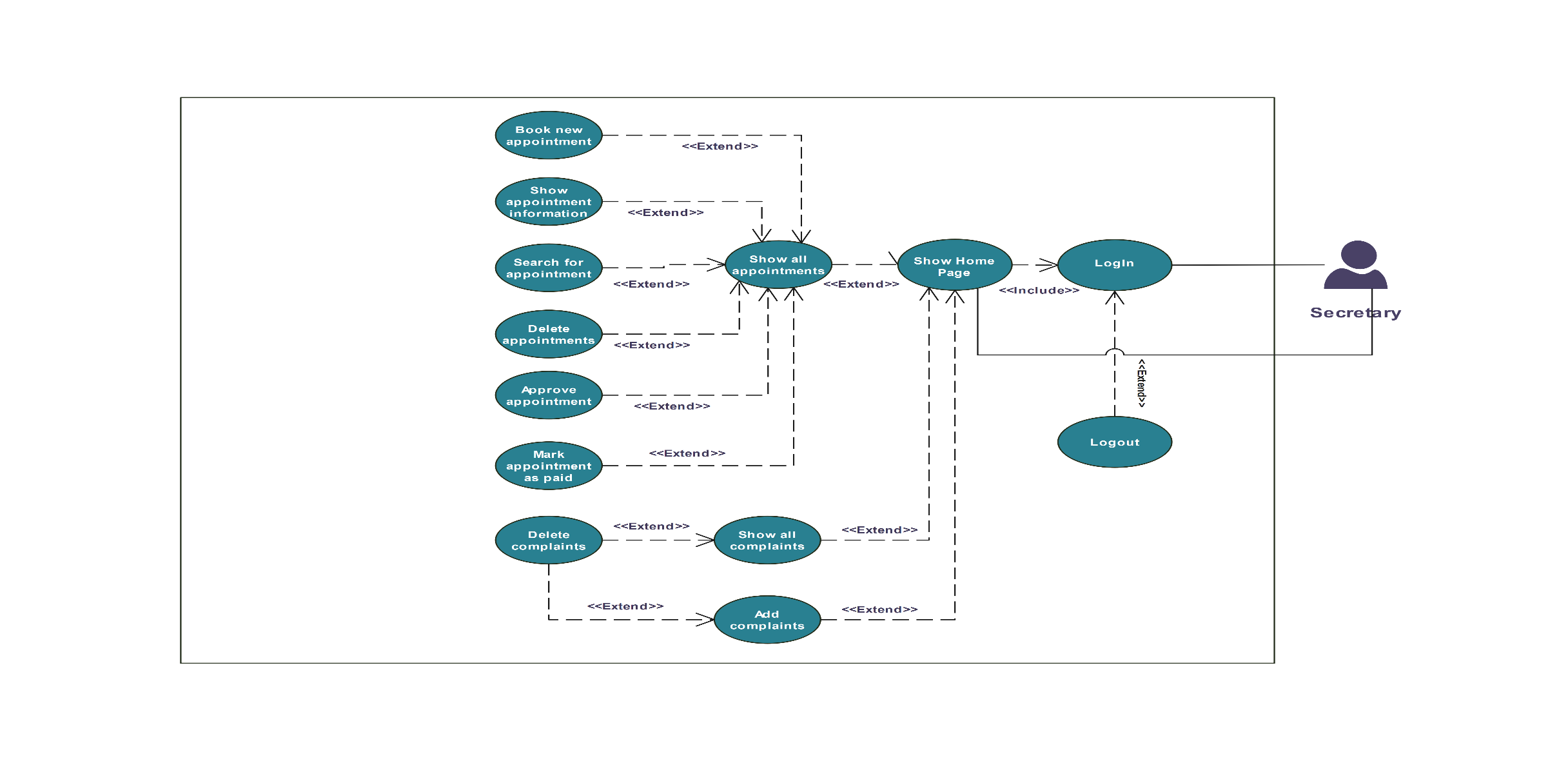
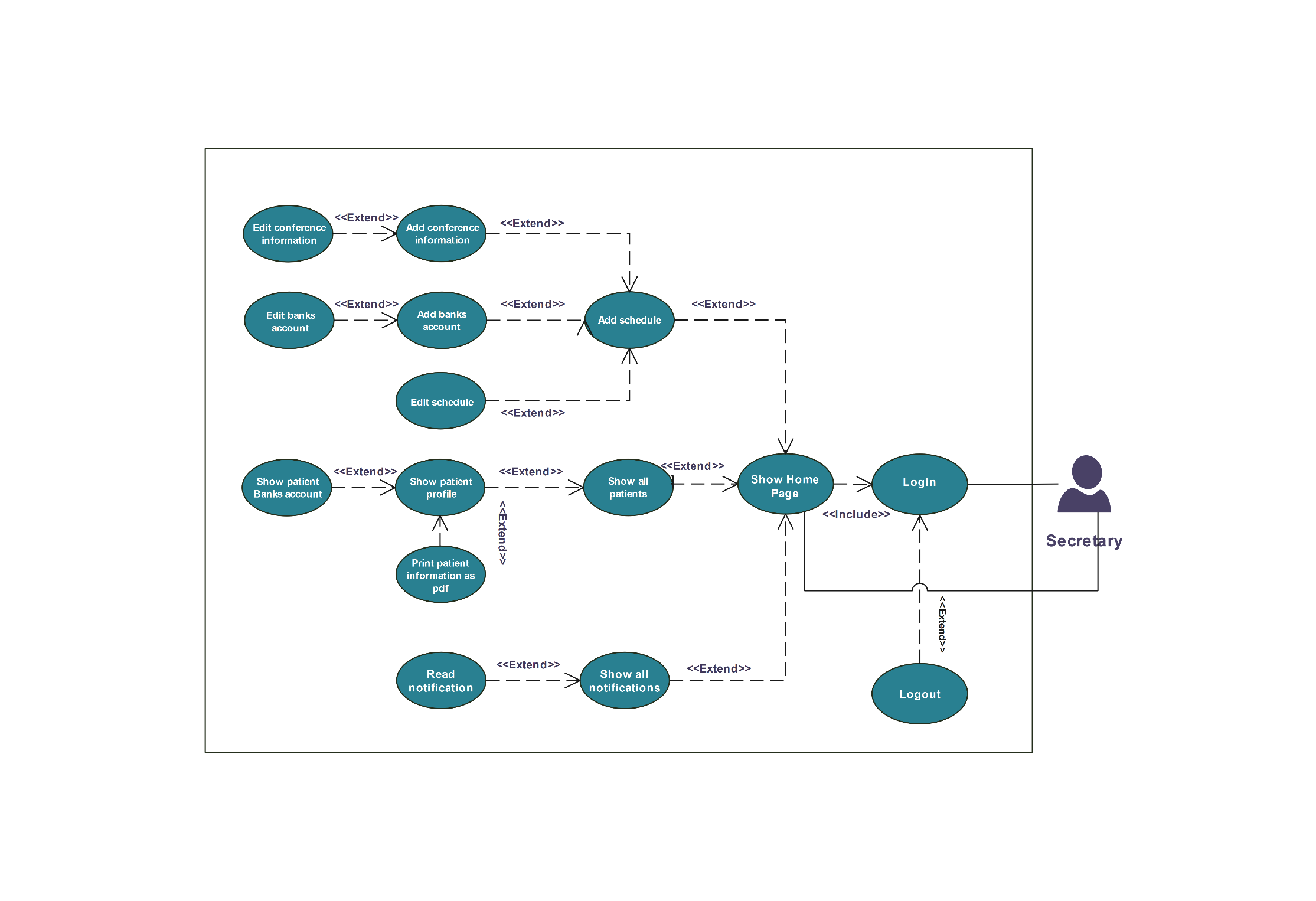


Figure 4.4 Patient Use Case Diagram



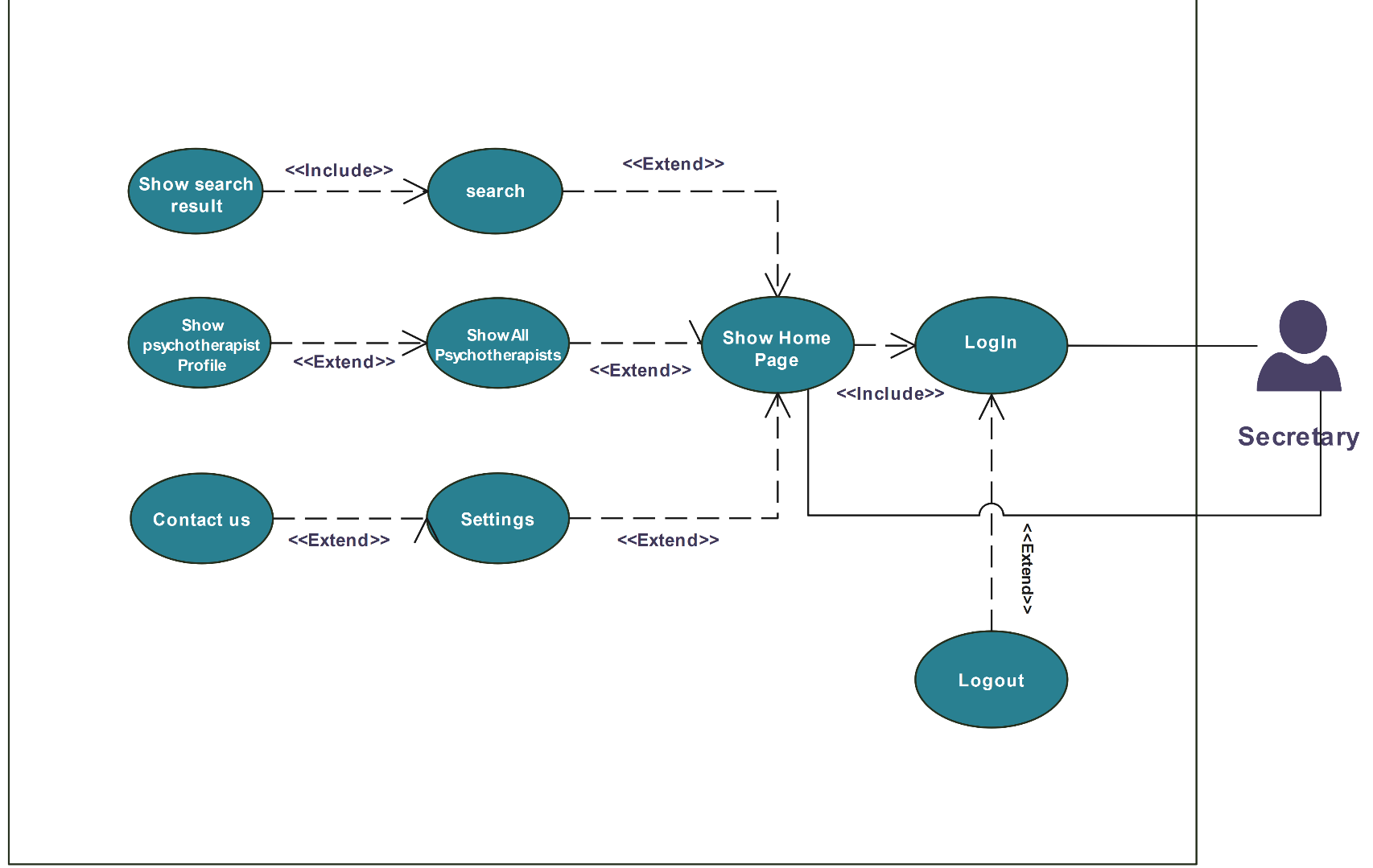


Figure 4.5 Secretary Use Case Diagram

Use case Specification :

Table 4.2 Signup Usecase Specification

|  |  |  |
| --- | --- | --- |
| UC-1 | Singup | |
| Actor | Doctor, Patient | |
| Brief Description | Allow patients and doctors to register for accounts on the website and establish user profiles. | |
| Pre-conditions | The website and service must be online and functioning properly  Users must have a stable internet connection. | |
| Post-conditions | 1. A confirmation email is sent to the user’s registered email address  2. The user account remains inactive until the email is verified  3. Upon clicking the verification link to active account | |
| Flow of event | Actor | System |
| Insert the required information |  |
| Click on Signup button |  |
|  | Create a new record in the database and save the information of the user |
|  | Sand email dynamically that content link to verify the email and activate account |
| **Critical scenarios** | 1. The user enters a valid email address during signup , but the verification email fails to deliver due to : full inbox that user’s is full , and the email bounces back .  **2.** Malicious Activity: A malicious actor attempts to create fake accounts using stolen email addresses or automated bots.  3.System Outage: The system responsible for sending or processing verification emails experiences an outage during the signup process. | |

Table 4.3 Email Verification Usecase Specification

|  |  |  |
| --- | --- | --- |
| UC-2 | Email Verification | |
| Actor | Doctor, Patient | |
| Brief Description | Confirm the validity of a user’s email address provided during signup | |
| Pre-conditions | The platform and email server must be operational  Users has successfully completed the signup process on the platform and provided a valid email address  The user’s email address should be in a format compatible with email delivery | |
| Post-conditions | 1. User account becomes active , granting them access to the platform’s functionalities (Login)  2. A success message is displayed to the user  3. The system record the verification timestamp for audit purposes | |
| Flow of event | Actor | System |
|  | Generates a unique verification link |
|  | Sand email to the user |
| User clicks the verification link in the email received |  |
|  | System updates the user’s account status based on the verification outcome |
| **Critical scenarios** | 1. The verification email is missing essential information like the verification link or the information provided is incorrect  **2. Email Spoofing** : A malicious actor sends a fake verification email impersonating your platform , attempting to trick the user into clicking a malicious link | |

Table 4.4 Login Usecase Specification

|  |  |  |
| --- | --- | --- |
| UC-3 | Login | |
| Actor | Doctor, Patient , Admin , Patient | |
| Brief Description | Allow registered users to access their accounts on the website by authenticating their identity | |
| Pre-conditions | the user must be a registered user on the platform , meaning they have successfully completed the signup process  The user’s account must be in an active state (validate his account) | |
| Post-conditions | 1. The session is created for the user  2. The user is redirected to a relevant landing page (Dashboard) | |
| Flow of event | Actor | System |
| Insert the email and password |  |
| Click the login button |  |
|  | Send the info for the database to check on it |
|  | Allowed the user to landing the dashboard |
| **Critical scenarios** | 1. Invalid login credentials : the user enters an incorrect email address or password  **2. Brute-Force Attack : A malicious actor attempts to gain unauthorized access by systematically trying different password combinations** | |

Table 4.5 Show Articles Usecase Specification

|  |  |  |
| --- | --- | --- |
| UC-4 | Show Articles | |
| Actor | Admin, Doctor, Patient, Guest User | |
| Brief Description | user views a list of published articles. | |
| Pre-conditions | The website is accessed by user. | |
| Post-conditions | A list of all published articles (status: published) is displayed, excluding articles reported by the current user. Each article includes all relevant information (e.g., category,Author name, content). | |
| Flow of event | Actor | System |
| Clicks on the "Articles" button in the navbar or footer or Home page. |  |
|  | Retrieves a list of all published articles (status: published).  If the user is a patient: Identify the user's reported articles. |
|  | Filter out reported articles from the retrieved list. |
|  | Render the list of articles on the "Articles" page, displaying the specified information for each article. |
| **Critical scenarios** | **1. No published articles:**  If there are no published articles, the system displays a message indicating that no articles are available.  **2. Pagination:**  If the number of published articles exceeds a predefined threshold (9), the system provides pagination controls to navigate through the list.  **3. Error handling:**   The system gracefully handles potential errors, such as database connection issues or invalid data, and provides informative error messages to the user. | |

Table 4.6 Like/Unlike Article Usecase Specification

|  |  |  |
| --- | --- | --- |
| UC-5 | Like/Unlike Article | |
| Actor | Patient | |
| Brief Description | patients on the article content page can click a "Like" button to either like or unlike the article. They cannot "like" the same article twice in a row. | |
| Pre-conditions | 1. Patient is logged in.  2. Patient is on the article content page. | |
| Post-conditions | 1. If the article is not liked:  1.1. The "Like" button changes appearance to indicate the article is liked.  1.2. The total number of likes for the article is incremented.  1.3. The article is added to the patient's "Favorite" list .  2. If the article is already liked:  2.1. The "Like" button changes appearance to indicate the article is not liked.  2.2. The total number of likes for the article is decremented.  2.3. The article is removed from the patient's "Favorite" list. | |
| Flow of event | Actor | System |
| clicks the "Like" button on the article content page. |  |
|  | checks the state of the article for the patient (liked or not liked). |
|  | updates the article's "likes" data. |
|  | updates the user interface to reflect the current state. |
|  | adds or removes the article from the patient's "Favorite" list accordingly. |
| **Critical scenarios** | 1. Patient is not logged in:  System displays an error message indicating the user needs to log in to like/unlike an article.  System might offer options to log in or continue without liking.  2. Technical error occurs during like/unlike action:  System displays an error message and logs the error for debugging. | |

Table 4.7 Add Article Usecase Specification

|  |  |  |
| --- | --- | --- |
| UC-6 | Add Article | |
| Actor | Doctor (Verified) | |
| Brief Description | A verified doctor can submit a new article through a dedicated form on the articles page. Upon submission, the system validates the data and displays an alert informing the doctor about the outcome. The submitted article is added to the system with a "pending" status, awaiting approval from an administrator. | |
| Pre-conditions | 1. Doctor is logged in and has a verified account with a valid certificate.  2. Doctor is on the articles page. | |
| Post-conditions | 1. The new article is added to the system with a "pending" status.  2. The doctor receives a confirmation alert indicating the article has been submitted for review.  3. The doctor is redirected to the articles page. | |
| Flow of event | Actor | System |
| clicks on the "Add Article" button on the articles page. |  |
|  | displays an article form with required fields |
| fills in the form fields and clicks the "Submit" button. |  |
|  | validates the submitted data |
|  | creates a new article entry with Content from the submitted form, "Pending" status and Association with the doctor as the author. |
|  | displays a confirmation alert informing the doctor about successful submission. |
|  | redirects the user to the articles page |
| **Critical scenarios** | 1. Doctor is not logged in:  1.1. System redirects the doctor to the login page.  2. Doctor's account is not verified:  2.1. System displays an error message informing the doctor that their account needs verification to submit articles.  3. Technical error occurs during submission:  3.1. System displays an error message and logs the error for debugging.  4. **validation fails:**  4.1. System displays an alert with specific error messages.  4.2. Doctor remains on the article form page to make corrections and resubmit. | |

Table 4.8 Accept/reject article Usecase Specification

|  |  |  |
| --- | --- | --- |
| UC-7 | Accept/reject article | |
| Actor | Admin  Doctor (Verified, not the publisher of the article) | |
| Brief Description | Doctors and admins on the pending articles page can click on any article to view its full details, including the content and other relevant information. | |
| Pre-conditions | 1. User is logged in.  2. User is on the pending article content page.  3. The user is not the doctor who published the article (applicable to doctors only). | |
| Post-conditions | 1. Article status is changed.  2. Success message is displayed to the user.  3. System redirects the user back to the pending articles page. | |
| Flow of event | Actor | System |
| clicks the "Accept" or “Reject” button. |  |
|  | If “reject” clicked Alert opens with a text field for entering rejection notes. |
| If “reject” clicked enters a note and clicks "Submit |  |
|  | Changes article status |
|  | Success message is displayed to the user. |
|  | redirects the user back to the pending articles page |
| **Critical scenarios** | 1. User is not logged in:  1.1. System redirects the user to the login page.  2. Technical error occurs during data retrieval:  2.1. System displays an error message and logs the error for debugging.  3. Selected article not found:  3.1. System displays an error message indicating the article cannot be found.  4. The user is the publisher:  4.1. System displays an informative message indicating the user cannot review their own article. | |

Table 4.9 Talk To Sandra Usecase Specification

|  |  |  |
| --- | --- | --- |
| UC-8 | Talk To Sandra | |
| Actor | Patient | |
| Brief Description | Offer basic mental health support and resources to users through an automated chat bot conversation and also initial diagnoses for our users | |
| Pre-conditions | Login successfully with as normal user not doctor and has good network connection | |
| Post-conditions | Conversation outcome : user received information on specific mental health topics , resources , or coping mechanisms | |
| Flow of event | Actor | System |
| Write on input message |  |
| Press “enter” or send to have response for the Bot |  |
|  | Generate a result question by using AI algorithm |
|  | Send the result for the user |
| **Critical scenarios** | 1. The user shares inaccurate or exaggerated information about their situation , potentially leading to inappropriate responses or advice  **2. The chat bot unintentionally provides biased or discriminatory information misinterprets user input or violates user privacy** | |

Table 4.10 Add certificate Usecase Specification

|  |  |  |
| --- | --- | --- |
| UC-9 | Add certificate | |
| Actor | Doctor | |
| Brief Description | It is a service that allows a doctor who has an account on our website to upload his graduation document for verification by the application administrator and allows him to enjoy many features. | |
| Pre-conditions | 1. complete “Signup” operation successfully  2. Verify his email  3. Login successfully and make sure his account is active | |
| Post-conditions | Admin : the admin can check his document and determine if it’s true or not (By Experience)  Doctor : after that he allows to use to many services like chatting or sharing articles | |
| Flow of event | Actor | System |
| Add his “Document” |  |
| Press “submit” button |  |
|  | Upload it if it’s one of the available files type (png,jpeg,..etc) |
|  | Create a new record in database to save this file and retrieve in another time |
| **Critical scenarios** | 1. The type of file not allowed to upload the document so we will reject and ask user to upload another file  2. Upload more than document inside one file it will rejected from the admin | |

Table 4.11 Approve Appointment Usecase Specification

|  |  |  |
| --- | --- | --- |
| UC-10 | Approve Appointment | |
| Actor | Doctor, Secretary | |
| Brief Description | The doctor or secretary approves a scheduled appointment. | |
| Pre-conditions | The user is logged into the system and navigates to the appointments page from the header. | |
| Post-conditions | The appointment status is changed to "approved," and a success message is displayed. | |
| Flow of event | Actor | System |
| The appointment status is changed to "approved," and a success message is displayed. |  |
|  | Displays the appointments page with a list of appointments and filters |
| Chooses an appointment and clicks on the "Info" button |  |
|  | Displays a container with appointment information and an "Approve" button |
|  | Clicks on the "Approve" button |  |
|  |  | Change appointment status to “Approved”. |
|  |  | Displays a success message |
| **Critical scenarios** | 1. No Appointments Available:   If there are no appointments available, the system displays a message indicating that there are no appointments to approve.   1. Invalid Appointment Selection:   If the user selects an invalid appointment (e.g., an appointment that has already been approved or cancelled), the system displays an error message and prompts the user to select a valid appointment. | |

Table 4.12 Delete Appointment Usecase Specification

|  |  |  |
| --- | --- | --- |
| UC-11 | Delete Appointment | |
| Actor | Doctor, Secretary , Patient | |
| Brief Description | The doctor , secretary or patient delete a scheduled appointment. | |
| Pre-conditions | The user is logged into the system and navigates to the appointments page from the header. | |
| Post-conditions | The appointment status is changed to "Canceled," and a success message is displayed. | |
| Flow of event | Actor | System |
| Navigates to the appointment page from the header |  |
|  | Displays the appointments page with a list of appointments and filters |
| Chooses an appointment and clicks on the "Info" button |  |
|  | Displays a container with appointment information and an "Delete" button |
| Clicks on the "Delete" button |  |
|  | Change appointment status to “Canceled”. |
|  | Displays a success message |
| **Critical scenarios** | 1. No Appointments Available:   If there are no appointments available, the system displays a message indicating that there are no appointments to approve.   1. Invalid Appointment Selection:   If the user selects an invalid appointment (e.g., an appointment that has already been cancelled), the system displays an error message and prompts the user to select a valid appointment. | |

Table 4.13 Mark Appointment as paid Usecase Specification

|  |  |  |
| --- | --- | --- |
| UC-12 | Mark Appointment as Paid | |
| Actor | Doctor, Secretary | |
| Brief Description | Doctors and secretaries can mark an appointment as paid. | |
| Pre-conditions | User is logged in and has accessed the appointment page. Appointment must be in "approved" or "waiting" status. | |
| Post-conditions | The appointment status is updated to "paid". | |
| Flow of event | Actor | System |
| Opens appointment page |  |
|  | Displays a list of appointments with their current statuses. |
| Selects an appointment with "approved" or "waiting" status. |  |
|  | Shows the details of the selected appointment, including its current status and paid button. |
| Clicks on "Paid" button. |  |
|  | Changes the status of the selected appointment to "paid". |
|  | Displays a success message |
| **Critical scenarios** | 1. Attempting to mark a non-eligible appointment as paid:   If the selected appointment is not in "approved" or "waiting" status, the system prevents the action and notifies the user.   1. Database update failure:   If the system encounters an issue while updating the appointment status in the database, it displays an error message and ensures no changes are made until the issue is resolved.   1. Error handling:   The system gracefully handles potential errors, such as database connection issues or invalid data, and provides informative error messages to the user. | |

Table 4.14 Start Session Usecase Specification

|  |  |  |
| --- | --- | --- |
| UC-13 | Start Session | |
| Actor | Doctor, Patient | |
| Brief Description | The doctor or patient starts a video call session for a scheduled appointment. | |
| Pre-conditions | The user is logged into the system, has an appointment scheduled for the current time, and navigates to the appointment information page. | |
| Post-conditions | The video call session is successfully started. | |
| Flow of event | Actor | System |
| Navigates to the appointment page from the header |  |
|  | Displays the appointments page with a list of appointments and filters |
| Chooses an appointment and clicks on the "Info" button |  |
|  | Displays the appointment information with a "Call" button. |
| Clicks on the "Call" button |  |
|  | Checks if the user has this appointment now |
|  | If the appointment is valid, opens the notes page for the user |
| Clicks on the "Accept" button |  |
|  | Displays a success message and redirects to the video call room |
|  | Generates the video call room (combination of user IDs and appointment ID) |
| Clicks "OK" |  |
|  | The session begins |
| **Critical scenarios** | 1. No Appointment at Current Time:   If the user tries to start a session outside the appointment time, the system displays an error message indicating no appointment is scheduled for the current time.   1. Invalid Appointment:   If the appointment is invalid (e.g., cancelled, already completed or not paid), the system displays an error message. | |

Table 4.15 Print Patient Profile Usecase Specification

|  |  |  |
| --- | --- | --- |
| UC-14 | Print Patient Profile | |
| Actor | Print Patient Profile | |
| Brief Description | The doctor, secretary, or patient prints the patient's profile, which includes all patient information and summarized doctor's notes. | |
| Pre-conditions | The user is logged into the system and navigates to the patient profile page.  Patients can only access and print their own profiles. | |
| Post-conditions | A PDF of the patient profile is downloaded, including all patient information and summarized doctor's notes. | |
| Flow of event | Actor | System |
| Navigates to the patient profile page. |  |
|  | Checks if the user is a patient and verifies they are accessing their own profile. |
|  | Displays the patient profile page. |
| Clicks on the "Print" button. |  |
|  | Generates a PDF with all patient information and summarized doctor's notes |
|  | Downloads the PDF. |
| **Critical scenarios** |  | |

Activity Diagram :

* + 1. Add Article :

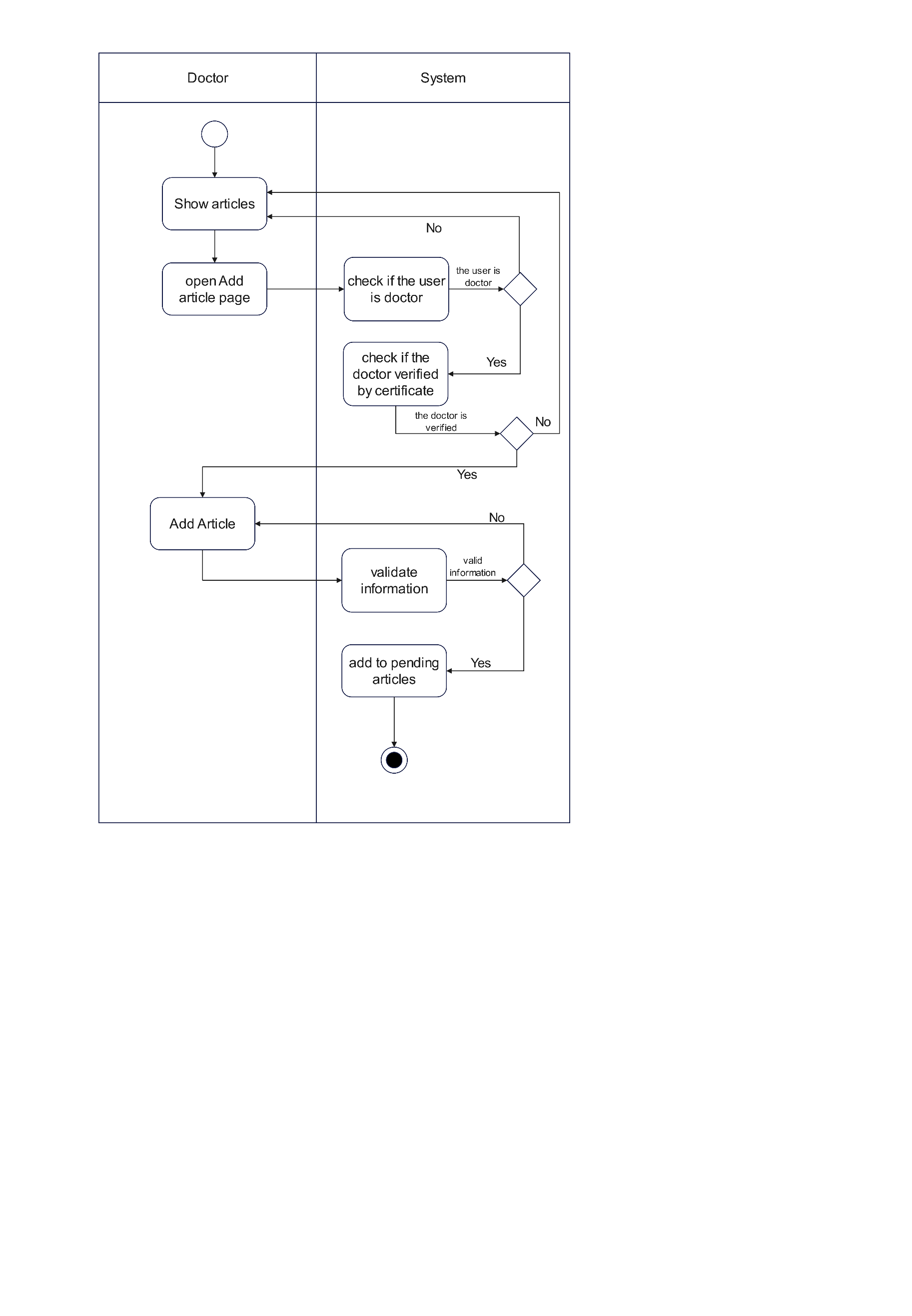


Figure 4.6 Add Article Activity Diagram

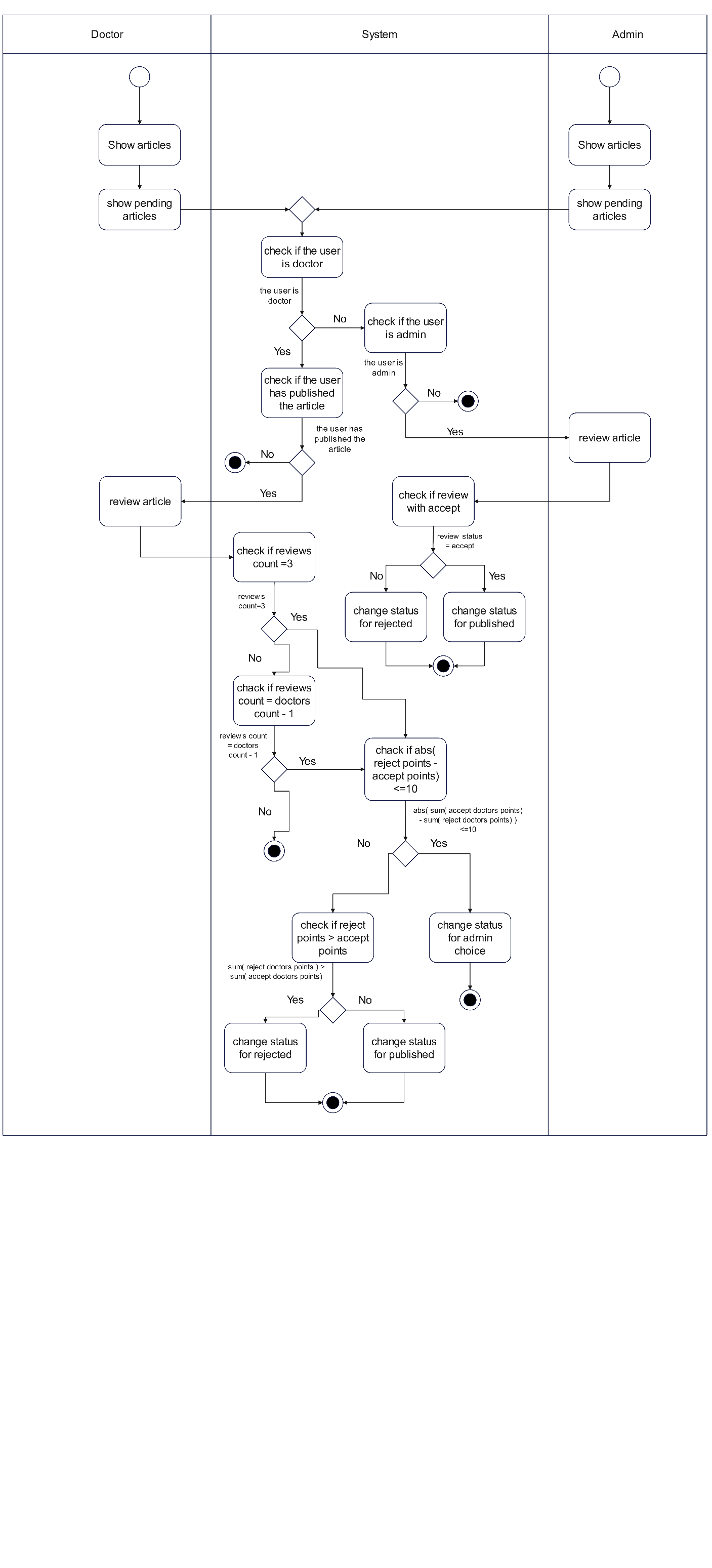
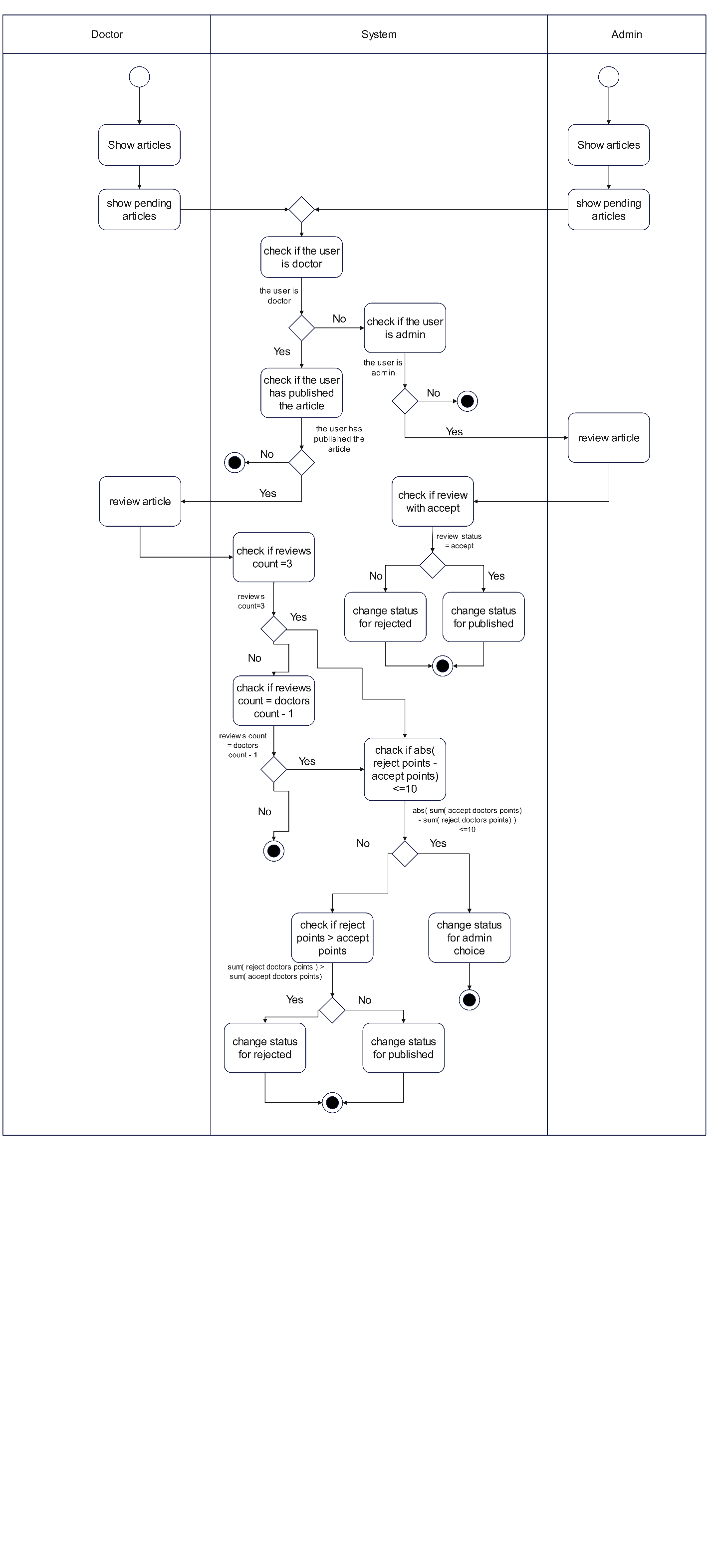
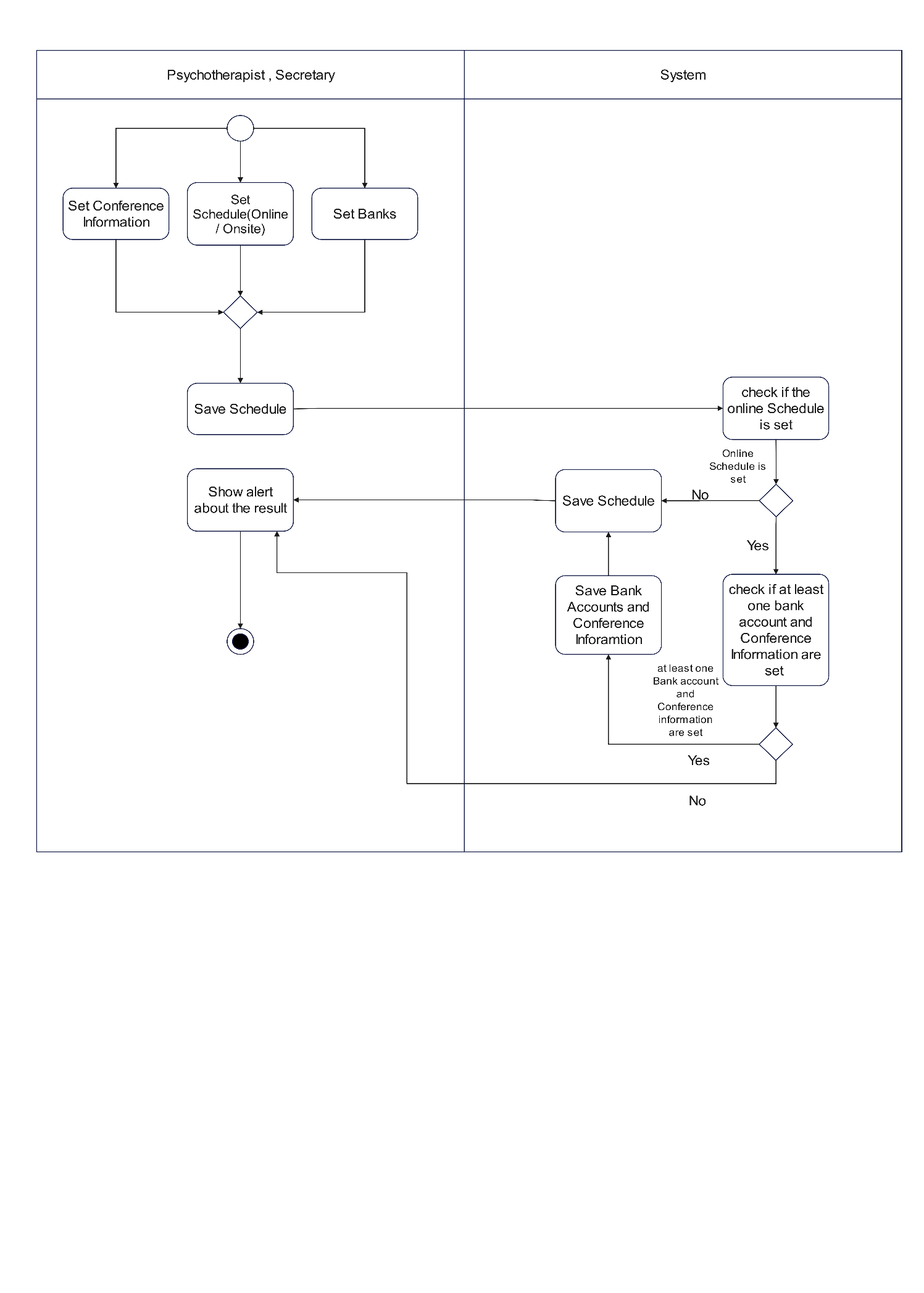
* + 1. Accept / Reject Article :

Figure 4.7 Accept / Reject Article Activity Diagram



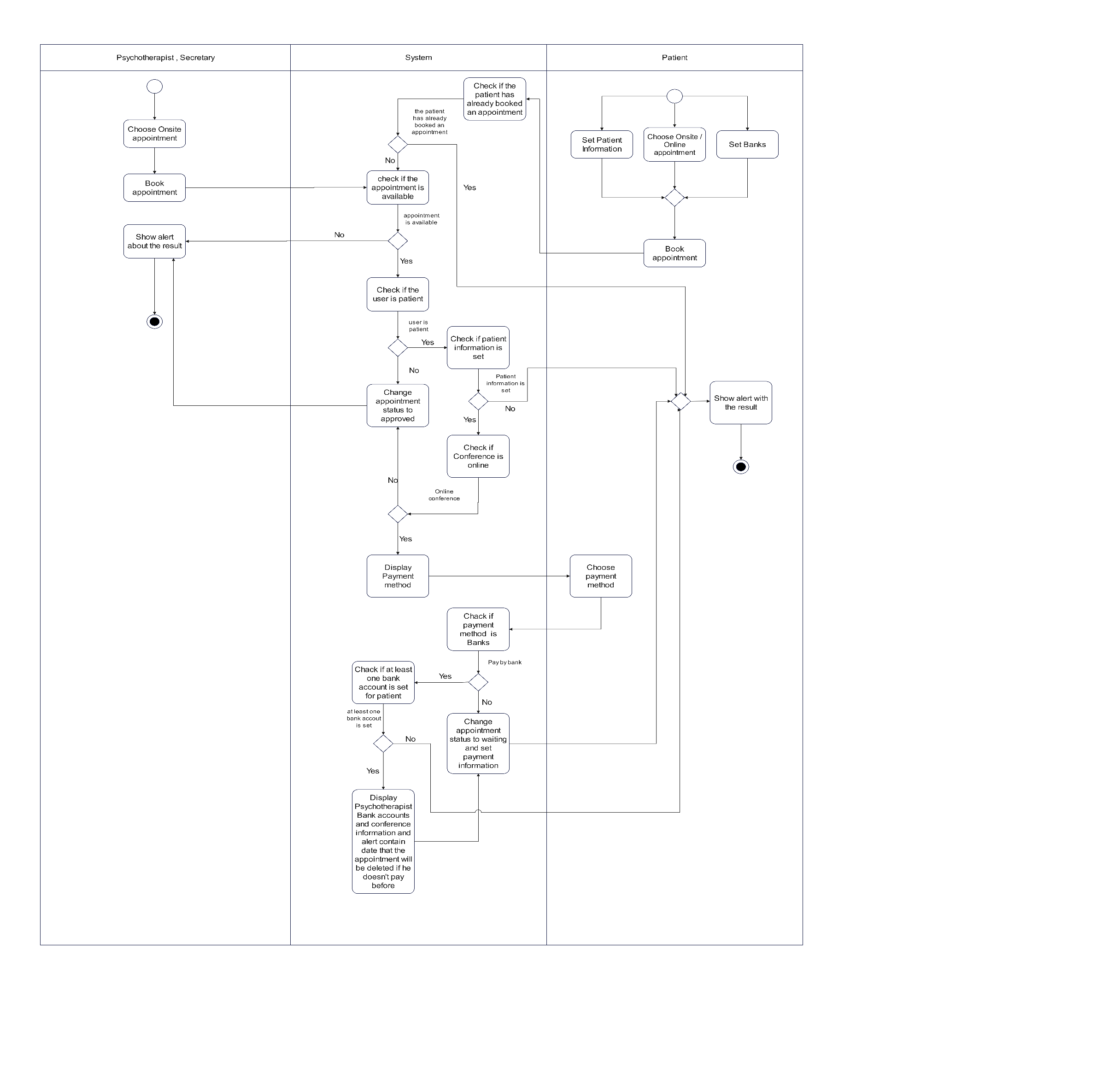
* + 1. Add Schedule :

Figure 4.8 Add Schedule Activity Diagram



* + 1. Book Appointment

Figure 4.9 Book Appointment Activity Diagram



Sequence Diagram :

* + 1. Start Chat :

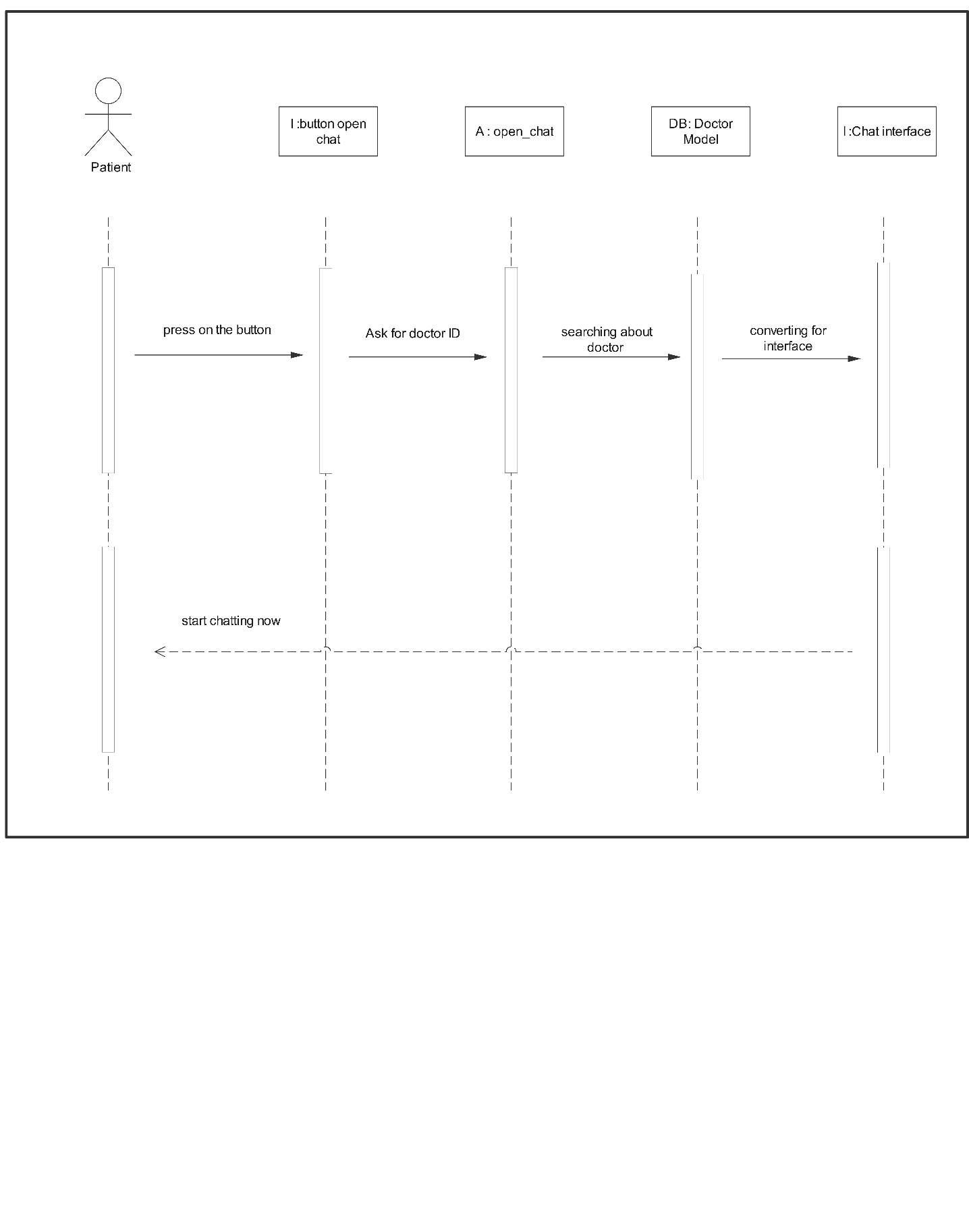


Figure 4.10 Start Chat Sequence Diagram

* + 1. Upload and Verify certificate :

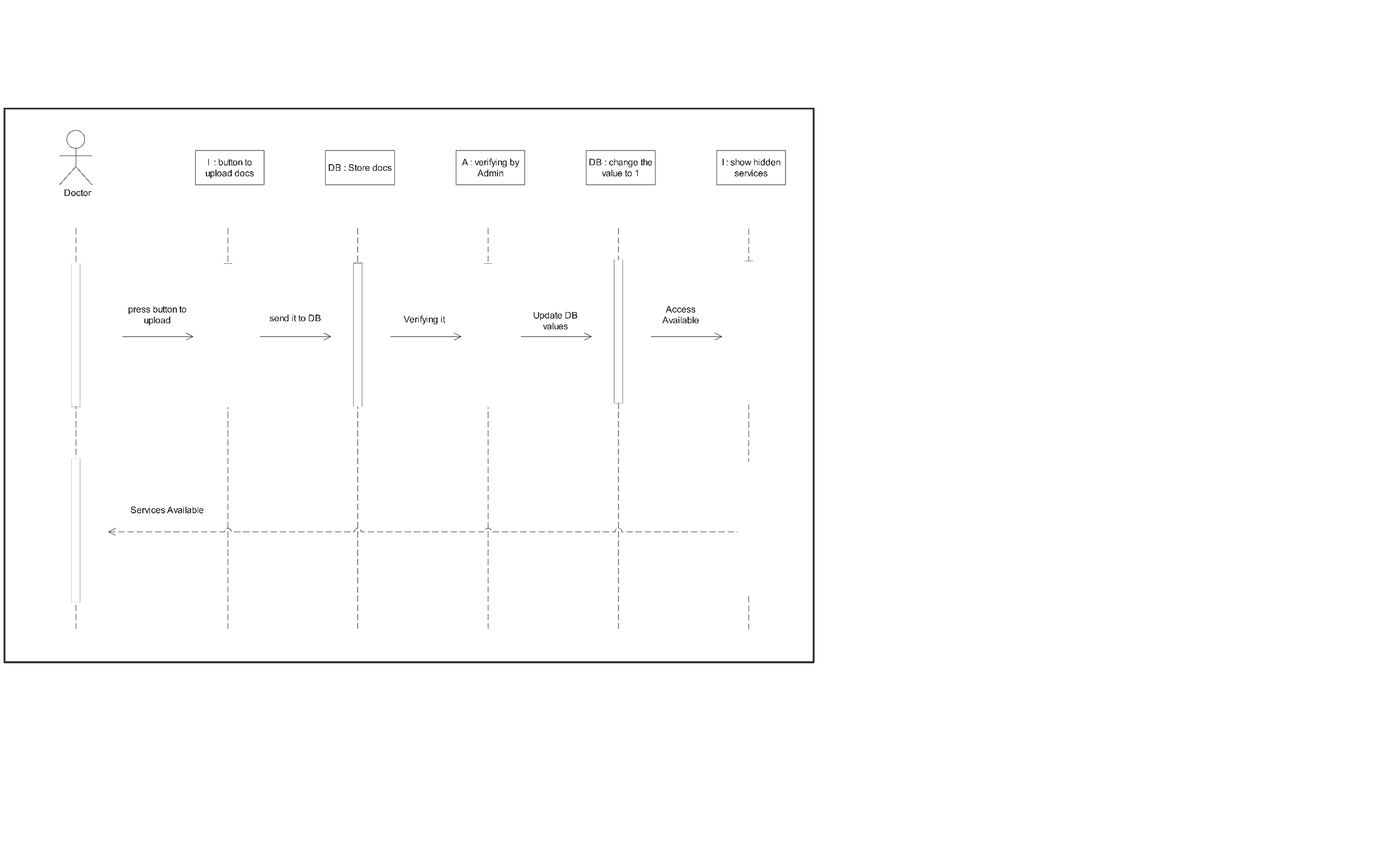


Figure 4.11 upload and verify certificate Sequence Diagram

Chapter Five : (Interface and system design)

the main purpose of the design level. Explain that its primary goal is to ensure that the work items are thoroughly analyzed, planned, and designed before development begins. This helps to minimize rework, improve efficiency, and enhance the quality of the final product.

System architecture:

This section outlines the architectural design of our mental health support platform, detailing the integration of layered and MVC architectures. We describe the fundamental principles of layered architecture and its implementation within our project, emphasizing the roles of React, Laravel, and Flask. Additionally, we explore the MVC pattern as employed by Laravel, highlighting its role in managing business logic and data interactions. The combination of these architectural styles ensures a modular, maintainable, and scalable system, enhancing the platform's overall functionality and user experience.

* + 1. Layered Architecture

Layered architecture organizes an application into distinct layers, each responsible for specific aspects of the software. These layers include the Presentation Layer, which handles user interfaces and user interaction; the Application Layer, which contains business logic and workflows; and the Data Access Layer, responsible for data storage and retrieval. This approach provides clear separation of concerns, enhances maintainability, and simplifies the development process.

* + 1. Implementation of Layered Architecture in Our Project

In our project, we implement a layered architecture to ensure modularity and ease of maintenance. The frontend, built using React, serves as the Presentation Layer, handling the user interface and interactions. Laravel, which follows the MVC pattern, acts as the Application Layer, managing the business logic and data flow. The Data Access Layer is managed by Laravel's Eloquent ORM, which interacts with the MySQL database for data persistence and retrieval. Additionally, we have integrated Flask as a specialized backend service for handling advanced natural language processing tasks using OpenAI's API, connecting directly with the React frontend.

* + - 1. Presentation Layer:

Managed by React, handling all user interface elements and interactions.

* + - 1. Application Layer:

Managed by Laravel, implementing business logic and workflows.

* + - 1. Data Access Layer:

Managed by Laravel's Eloquent ORM, interacting with the MySQL database.

* + - 1. Specialized Processing Layer:

Managed by Flask, handling advanced natural language processing tasks with OpenAI's API.

* + 1. MVC Architecture in Laravel

Laravel implements the MVC (Model-View-Controller) architecture to organize the backend components. The Model represents the application's data structure and business logic, interacting with the MySQL database through Eloquent ORM. The View is responsible for presenting data to the user, though in this project, it is primarily managed by the React frontend. The Controller acts as an intermediary, processing incoming HTTP requests, interacting with the model to fetch or manipulate data, and then determining the appropriate view to render. This structure ensures a clear separation of concerns, allowing for efficient handling of data and user interactions.

By combining layered architecture with the MVC pattern in Laravel, we achieve a robust and scalable system. React manages the frontend presentation, Laravel handles the core application logic and data management, and Flask is employed for specialized processing tasks, enhancing the overall functionality and user experience of the mental health support platform.

Block Diagram :

Entity Relationship Diagram (ERD) :

Interface Design :



Chapter Six : (Develop , Testing , Deploy)



Develop:

Following the completion of the psychotherapists' functionality, we moved on to developing the articles functionality. This feature allows psychotherapists to share informative articles and resources with other users, fostering a collaborative environment for knowledge exchange and empowering patients with valuable information to support their mental well-being.

Next, we concentrated on the patients' functionality, aiming to create a seamless and user-friendly experience for individuals seeking mental health support. This includes features like secure registration, personalized profiles, and the ability to book appointments and communicate with their assigned psychotherapists.

The chat functionality was another crucial aspect we worked on, providing a dedicated space for real-time communication between psychotherapists and patients. This feature enables secure and confidential conversations, allowing patients to express their concerns and psychotherapists to provide guidance and support.

To ensure smooth administration of the platform, we developed the admin functionality. This allows administrators to manage user accounts, handle system configurations, and monitor the overall performance of the platform.

Additionally, we focused on the settings functionality to provide users with customizable options that suit their preferences. This includes personalization settings, notification preferences, and language selection, among other features.

Lastly, we turned our attention towards the bot functionality. Leveraging Python, we developed an intelligent bot that assists with initial diagnoses, providing users with preliminary insights and suggestions based on their symptoms and concerns. This enhances the responsiveness of the platform and helps users gain a better understanding of their mental health status.

By meticulously working through these stages, we aim to create a comprehensive mental health support platform that empowers psychotherapists, patients, and administrators alike, fostering effective communication, knowledge sharing, and improved mental well-being.

* + 1. Advanced Models and Integration

In developing our mental health support platform, we utilized GPT-3.5, a cutting-edge language model by OpenAI, chosen for its exceptional natural language understanding and generation capabilities. This makes GPT-3.5 highly suitable for complex tasks such as diagnosing mental disorders based on user input and summarizing psychotherapists' notes.

* + - 1. Why GPT-3.5:

GPT-3.5 was selected for its robust performance in generating human-like text and handling diverse conversational contexts. Its transformer-based architecture, with multiple layers of attention mechanisms, enables it to capture long-range dependencies and context within text. This makes it particularly effective in understanding nuanced questions and providing coherent, contextually relevant responses.

* + - 1. Transformers in GPT-3.5:

Transformers constitute the core architecture of GPT-3.5, designed specifically with a decoder-only framework to excel in natural language processing tasks. Unlike traditional encoder-decoder transformers used for tasks like machine translation, GPT-3.5 utilizes a decoder-only setup to generate text sequentially and autoregressively. This architecture enables GPT-3.5 to process input text by attending to different parts of the sequence using self-attention mechanisms. These mechanisms allow the model to understand and respond to complex queries in a contextually aware and semantically accurate manner. The use of positional encoding further aids in understanding the sequential nature of the input, ensuring that GPT-3.5 can generate coherent and meaningful responses based on the preceding context.

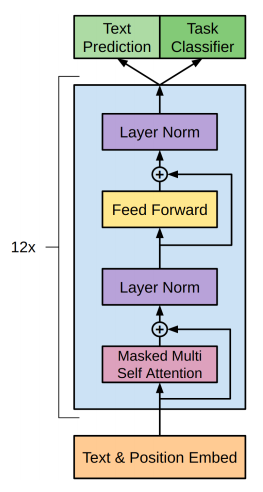
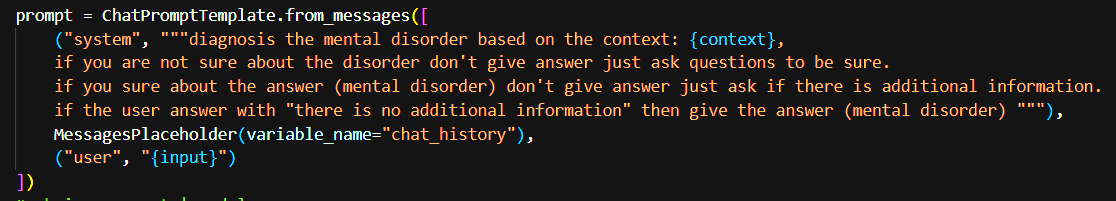


Figure 6.1 GPT3.5 Transformer (Decoder-only)

* + - 1. Prompt Design:

We designed specific prompts to guide GPT-3.5 in providing accurate and relevant responses. The primary prompt for diagnosing mental disorders is structured as follows:

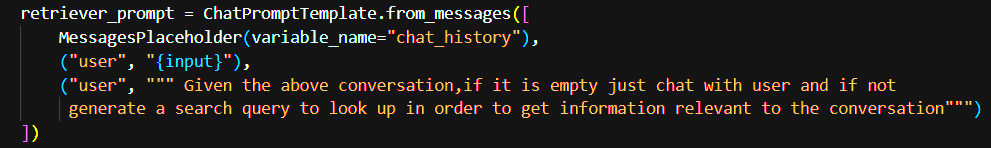
Figure 6.2 Prompt Structure



This prompt ensures that GPT-3.5 only provides a diagnosis when it is confident, otherwise prompting the user for more information to ensure accuracy. This approach is known as a **Conditional Prompt**, where the system gathers necessary information before providing a conclusive response.

We also use a retriever prompt to assist in fetching relevant information during conversations:

Figure 6.3 Retriever Prompt Structure



* + - 1. LangChain Integration:

LangChain is a powerful framework we integrated into our project to enhance the capabilities of GPT-3.5. LangChain allows us to manage and streamline interactions between various components, such as document retrieval and conversational history management. By using the GPT-3.5 API within LangChain, we ensure that our system can handle complex workflows and provide users with coherent and contextually relevant information.

* + - 1. Merging GPT-3.5 with LangChain:

The system retrieves and processes documents, creates embeddings, and sets up a retrieval chain using LangChain. We use the DSM-5 (Diagnostic and Statistical Manual of Mental Disorders) as the primary dataset to inform GPT-3.5's responses, ensuring the information provided is accurate and clinically relevant.

The DSM-5 data is integrated into the prompts, allowing GPT-3.5 to access and utilize this comprehensive resource for mental health diagnostics.

Figure 6.4 Data processing

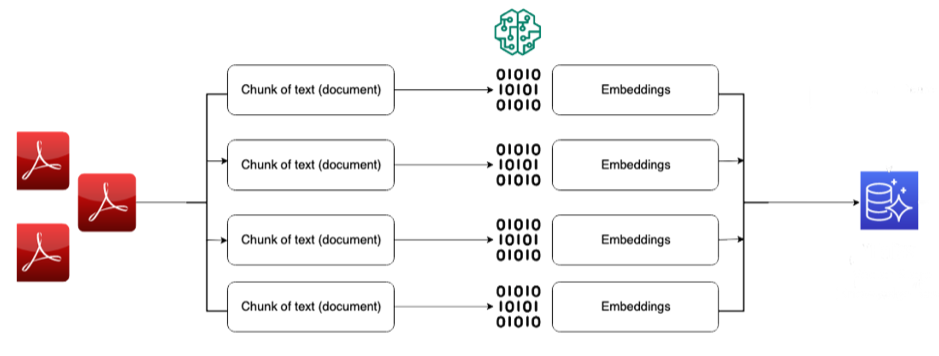
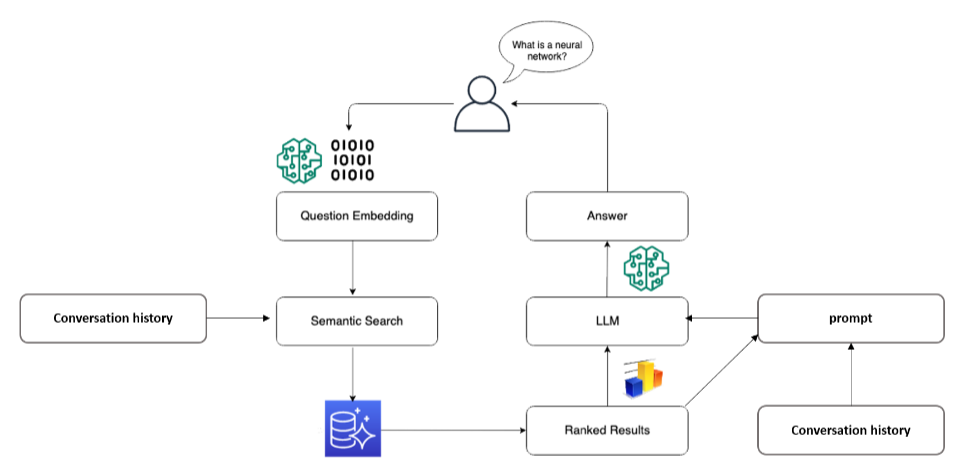


Figure 6.5 Query processing and execution



By combining the advanced capabilities of GPT-3.5 with the structured workflow management of LangChain and integrating critical data sources like DSM-5, our platform provides a comprehensive and effective solution for mental health support. This ensures users receive accurate, contextually relevant information and support tailored to their needs.

Testing and Deploy:

In addition to the extensive development process, we also place great emphasis on thoroughly testing each functionality within our project. Once a specific feature or functionality is deemed stable and reliable, we proceed to test it in a controlled environment. This allows us to ensure that all capabilities and enhancements are functioning as intended before final submission.

Our testing process involves careful planning and coordination to ensure a smooth transition from development to testing. We work closely with our team to ensure that all necessary resources are in place for successful testing.

After completing the tests, we closely monitor the performance and usage of each functionality. This allows us to gather valuable insights and make data-driven decisions to further optimize and improve the user experience.

* + 1. Performance Metrics

To maintain high performance and ensure our platform delivers a responsive and reliable user experience, we implemented several key performance metrics:

* + - 1. ****Queries per Second (QPS)****:

This metric measures the number of queries the system can handle per second, providing insight into its capacity and scalability.

* + - 1. ****Average Response Time****:

By calculating the average time taken to respond to user queries, we ensure the system delivers timely support.

* + - 1. ****Query Time****:

This measures the time taken to process individual queries, helping us optimize and improve system performance.

* + 1. User Experience

We prioritize delivering a user-friendly platform. We focus on creating intuitive interfaces and implementing features that simplify the user experience. This includes designing clear and accessible communication channels between psychotherapists and patients, as well as providing a seamless appointment booking process.

By prioritizing the testing and performance evaluation of each functionality, we ensure that our project progresses efficiently and that users can access and benefit from the platform's features in a timely manner. This approach enables us to make a significant impact on mental health support and user satisfaction within our project scope.

Chapter Seven : (Conclusion)



Conclusion :

In conclusion, our project represents a significant advancement in the field of mental health support and communication between psychotherapists and patients. By leveraging technologies such as React, Laravel, Flask, and ZegoCloud, we have successfully developed a robust and user-friendly platform. The integration of features such as article sharing, real-time chat services, AI-driven support tools, appointment scheduling with online conferencing capabilities, complaint handling, notifications, and AI-driven summarization of psychotherapists' notes has greatly enhanced the accessibility and effectiveness of mental health care delivery.

Our commitment to timely deployment ensures that users can promptly access and leverage the platform's comprehensive functionalities. By facilitating seamless interaction between patients and licensed professionals, our project not only streamlines access to crucial mental health resources but also empowers individuals to take proactive steps towards their well-being.

Overall, our project is poised to make a tangible impact by fostering improved communication, knowledge dissemination, and personalized support within the mental health landscape. With a focus on innovation and user-centric design, we are confident that our platform will continue to evolve and positively influence the lives of its users.

Future Work:

Looking ahead, there are several enhancements and new features that can further elevate the functionality and impact of our mental health support platform. One promising direction is the implementation of a recommendation system. This system could analyze user interactions and preferences to provide personalized content, such as tailored articles, suggested psychotherapists, and relevant self-help resources, thereby enhancing user engagement and satisfaction.

Another important advancement is the development of a robust model for checking articles before publishing. This model would ensure the credibility and reliability of the information provided by automatically reviewing and validating the content against established medical guidelines and current research, thereby maintaining the platform’s high standards for quality and accuracy.

In addition to these features, there are several other improvements we envision:

* + 1. **Mobile Application:**

Developing a mobile app version of the platform to increase accessibility and convenience for users, allowing them to access mental health resources and support anytime, anywhere.

* + 1. ****Multilingual Support:****

Expanding the platform to support multiple languages, making mental health resources accessible to a broader, more diverse audience.

* + 1. ****Community Forums:****

Creating community forums where users can share their experiences, support each other, and discuss mental health topics in a moderated, safe environment.

* + 1. ****PayPal Payment Integration:****

Adding support for PayPal as a payment method for online conferences and other services, providing users with a convenient and widely-used option for secure electronic payments.

By pursuing these future works, our platform can continue to innovate and provide even more effective, personalized, and accessible mental health support to its users.

References