



Sandra

Innovative Mental Health Platform

Prepared by :

Khaled Abd-Alhameed Alshagheer

Sireen Mahmoud AL-Takriti

Supervised by

Dr. Hiyam Khaddam

Eng. Anas Mokhallalati

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 doctors, and interact with a supportive chatbot named "Sandra." 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	1
Contents List.....	2
List of Figures.....	4
List of Tables.....	6
List of Abbreviation	7
Acknowledgements.....	8
Chapter One : (Introduction).....	9
1.1. Motivation:.....	10
1.2. Problem description:.....	10
1.3. project objectives:.....	11
Chapter Two : (Background).....	12
2.1. Theoretical Underpinnings:.....	13
2.2. Existing Systems:	13
2.3. the proposed solution:	14
Chapter Three : (Methodology)	16
3.1. Kanban:.....	17
3.2. Kanban Stages in Our Project:.....	18
3.3. Technologies Used:	23
Chapter Four : (Analysis)	27
4.1. Target user:	28
4.2. Requirement Gathering:.....	28
4.3. Functional and non-functional requirements:	29
4.4. Class Diagram:	32
4.5. Use case Diagram:	33
4.6. Use case Specification:	38
4.7. Activity Diagram:	60
4.8. Sequence Diagram:.....	62
Chapter Five : (Interface and system design).....	64
5.1. Block Diagram :.....	65
5.2. Entity Relationship Diagram (ERD) :	66
5.3. Interface Design :.....	67
Chapter Six : (Develop , Testing , Deploy)	79

6.1. Develop:	80
6.2. Testing and Deploy:.....	80
Chapter Seven : (Conclusion)	82
7.1. Conclusion:.....	83
7.2. Future Work:	83
References.....	84

List of Figures

Figure 3.1 Kanban Principles	18
Figure 3.2 Kanban Stages	19
Figure 3.3 Kanban board	23
Figure 4.1 Class Diagram	32
Figure 4.2 AdminUseCase (Doctors , Articles).....	33
Figure 4.3 AdminUseCase (Settings,Verify).....	33
Figure 4.4 DoctorUseCase (Settings,Chat,Doctors,profile,search).....	34
Figure 4.5 DoctorUseCase (articles)	35
Figure 4.6 PatientUseCase (Doctors , Chats)	36
Figure 4.7 PatientUseCase (Search , Articles, Settings , chatbot).....	37
Figure 4.8 Add Article Activity Diagram	60
Figure 4.9 accept/reject Article Activity Diagram	61
Figure 4.10 accept/reject article Activity diagram	62
Figure 4.11 Start Chat Sequence Diagram.....	62
Figure 4.12 upload and verify certificate Sequence Diagram	63
Figure 5.1 Block Diagram	65
Figure 5.2 ERD	66
Figure 5.3 SignUp Interface	67
Figure 5.4 Login Interface.....	67
Figure 5.5 Home Page Interface	68
Figure 5.6 Home Page Interface	69
Figure 5.7 Doctors Interface.....	70
Figure 5.8 Doctor Profile Interface.....	70
Figure 5.9 Search Interface	71
Figure 5.10 ChatBot (Sandra) Interface.....	71
Figure 5.11 Settings Interface	72
Figure 5.12 Doctor Interface	72
Figure 5.13 Add Certificate Form Interface.....	72
Figure 5.14 Articles Interface	73
Figure 5.15 Add Article Form Interface	74
Figure 5.16 Verify request Interface	74
Figure 5.17 My Articles Interface	75

Figure 5.18 Pending Articles Interface	76
Figure 5.19 Chat Interface.....	76
Figure 5.20 Article Content Interface.....	77
Figure 5.21 Chats Interface	78

List of Tables

Table 4.1 Show Articles Use Case Specification	38
Table 4.2 Filter Articles Use Case Specification.....	39
Table 4.3 Show Article Content Use Case Specification.....	40
Table 4.4 Report article use Case Specification	42
Table 4.5 Delete Article Use Case Specification	44
Table 4.6 Add Article Use Case Specification	45
Table 4.7 Show Pending Articles Use Case Specification	47
Table 4.8 accept/reject article Use Case Specification	48
Table 4.9 Show All Doctors Use Case Specification.....	49
Table 4.10 Show Doctor Profile Use Case Specification.....	50
Table 4.11 Search Use Case Specification	51
Table 4.12 Signup Use Case Specification	52
Table 4.13 Email Verification Use Case Specification.....	53
Table 4.14 Login Use Case Specification	54
Table 4.15 Talk To Sandra Use Case Specification	55
Table 4.16 Send Message Use Case Specification.....	56
Table 4.17 Show Chat Content Use Case Specification.....	57
Table 4.18 Add Certificate Use Case Specification	58

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

To begin with, we extend our sincere gratitude to those who have been instrumental in the success of our project.

Firstly, our deepest appreciation goes to Dr.Hiyam Khaddam for her invaluable mentorship and guidance. Her expertise and encouragement have not only shaped the development of our project but have also inspired us to strive for excellence.

Additionally, gratitude extends to Eng.Anas Mokhallalati for his unwavering support and technical insights throughout the journey of this project. His expertise has been a cornerstone in navigating challenges and achieving our project goals.

A special thanks to the faculty and staff of Yarmouk Private University for providing a nurturing academic environment. The resources and opportunities made available have significantly contributed to the realization of our project.

To our families and friends, we express our appreciation for their understanding and encouragement. Their unwavering support has been a source of motivation throughout this endeavor.

This collaborative effort has been a profound learning experience, and we are thankful to all who have been a part of this journey, contributing to the successful completion of this project.

Chapter One :

(Introduction)

1.1. Motivation:

1.1.1. Our target audience:

- A. Support Seekers: Individuals facing psychological challenges and seeking information and support.
- B. Healthcare Professionals: Doctors and specialists in mental health who wish to publish articles and communicate with patients.
- C. Administrative Professionals: Administrative officials in need of website content management and physician documentation.

1.1.2. 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.

1.2. 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.

1.3. 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.

2.1. 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.

2.2. Existing Systems:

2.2.1. 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.

2.2.2. 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.

2.2.3. 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.

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.

2.3. 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:

2.3.1. Verified psychologist directory:

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

2.3.2. Credible mental health articles:

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

2.3.3. Optional chat feature:

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

2.3.4. 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.

3.1. 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.)

3.1.1. 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.

3.1.2. 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.

3.1.3. 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.

3.1.4. 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.

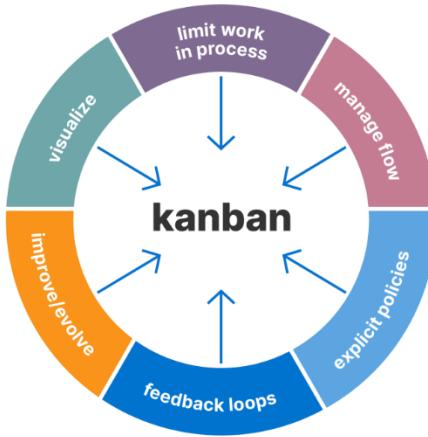


Figure 3.1 Kanban Principles

3.2. Kanban Stages in Our Project:

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

3.2.1. To Do:

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

3.2.2. 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.

3.2.3. 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.

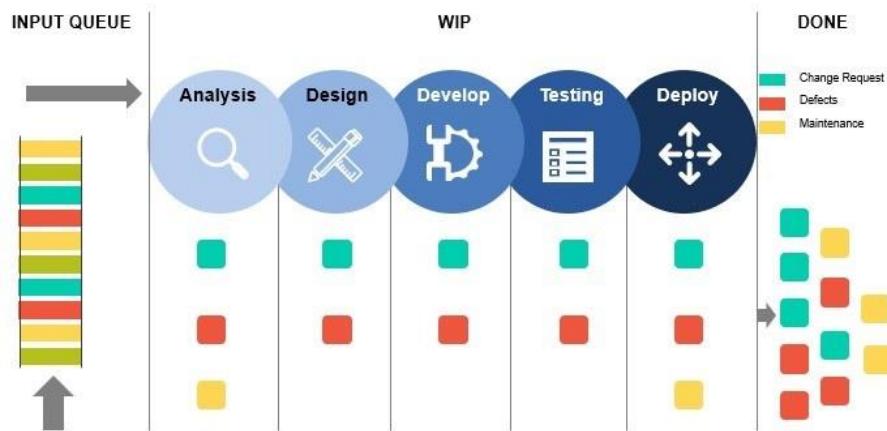
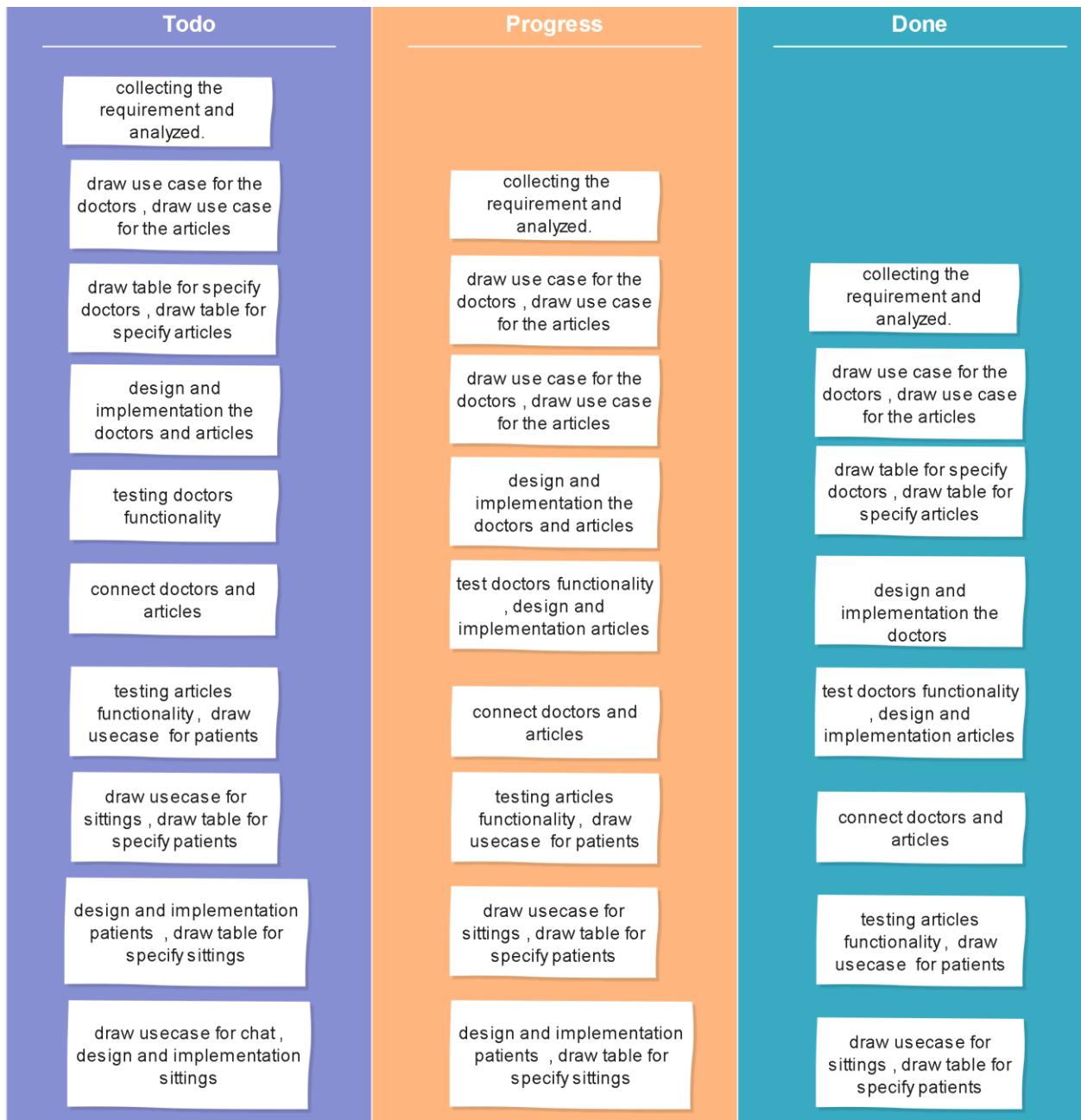


Figure 3.2 Kanban Stages

The following figures depict kanban board :



Todo	Progress	Done
draw table for specify chat	draw usecase for chat , design and implementation sittings	draw table for specify sittings , draw table for specify patients
design and implement the chat , draw usecase for the admin	draw table for specify chat , design and implement sittings	draw usecase for the caht
draw table for specify admin	design and implement chat , draw usecase for admin	draw table for specify chat , design and implement sittings
design and implement admin	design and implement chat , draw table for specify admin	draw usecase for admin
connect chat with doctors and patients	design and implement chat , design and implement admin	draw table for specify admin
testing chat functionality , connect admin with doctors	connect chat with doctors and patients , design and implement admin	design and implement chat
connect sittings with each of doctors and admin and patients	test chat functionality , connect doctor with admins	design and implement admin , connect chat with doctors and patients
testing patients functionality , connect admin with doctors	connect sittings with each of doctors and admin and patients	testing chat functionality , connect admin with doctors
draw table for specify bot , testing admin functionality	testing patients functionality , connect admin with doctors	connect sittings with each of doctors and admin and patients
design and implement chat , design and implement Bot	testing admin functionality , draw table for specify Bot	connect admin with doctors , test patient functionality

Todo	Progress	Done
Write Documentation	test sittings functionality , design and implement bot	draw table for specify Bot , connect doctors with articles
testing Bot functionality	Write documentation , design and implement Bot	test sittings functionality
	Write documentation , design and implement Bot	
integration testing	design and implement Bot , Write Documentation	design and implement Bot
	Testing Bot functionality , Write documentation	
	Testing Bot functionality , Write Documentation	Test Bot functionality
draw diagrams (activite and sequence).	integration testing , Write documentation	Integration testing
	Integration testing , write documentenion	
	draw diagrams (activite and sequence), Write documentation	
	draw diagrams (activite and sequence),Write documentation	
		draw diagrams (activite and sequence),Write documentation

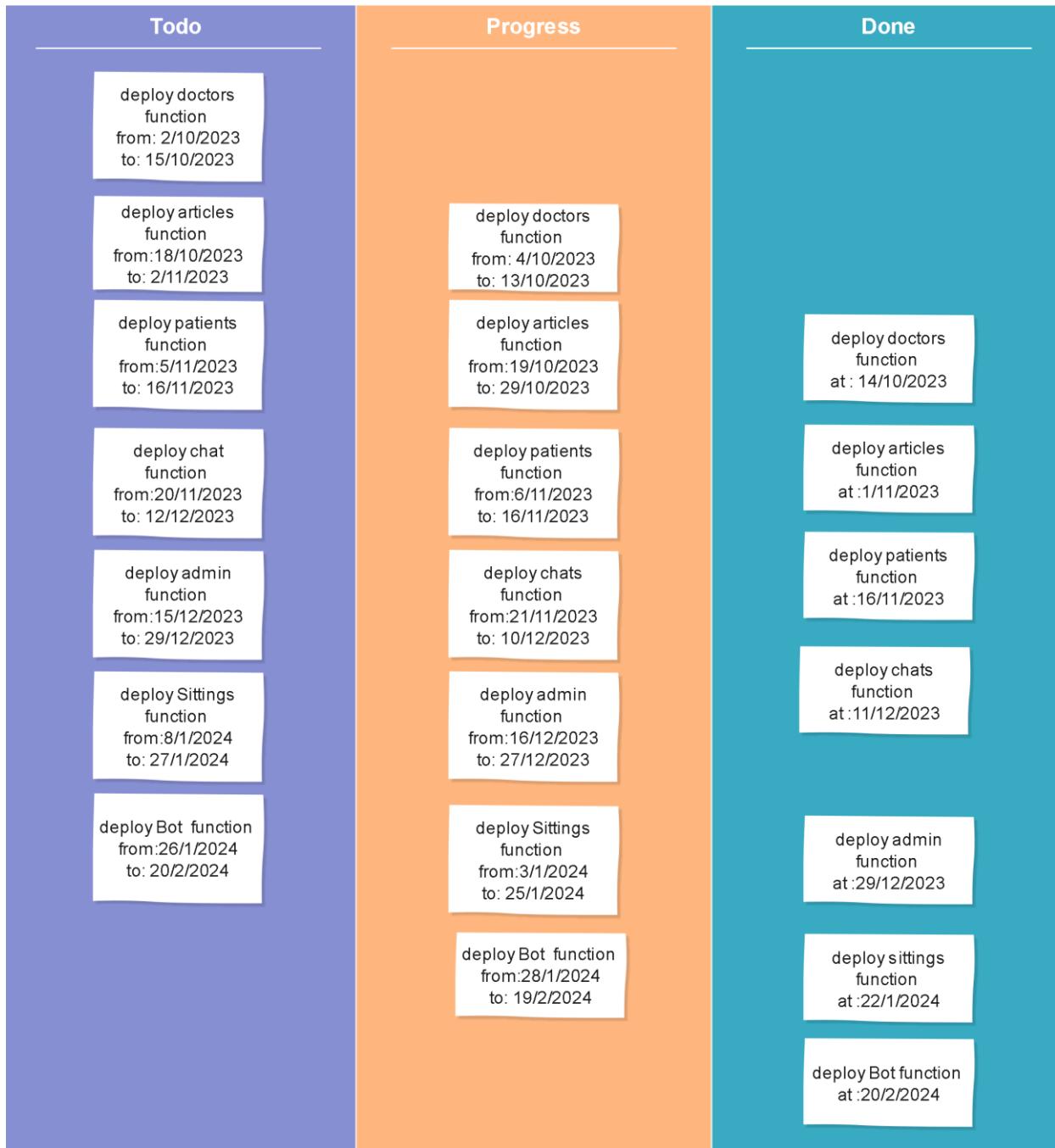


Figure 3.3 Kanban board

3.3. Technologies Used:

3.3.1. 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.

3.3.2. 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.)

3.3.3. 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.

3.3.4. 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.

3.3.5. Python:

language known for its clarity and versatility, serves as the driving force behind our project's AI-powered chatbot, a tool designed to enhance mental health assessment and support. Its intuitive syntax and extensive libraries make it an ideal choice for natural language processing, machine learning, and deep learning tasks—all integral to the chatbot's functionality.

Through Python's capabilities, we have successfully developed a chatbot that offers a sophisticated self-assessment tool. It analyzes user input, identifies potential psychological disorders, and provides initial guidance, promoting early intervention and personalized support within the mental health realm. (get_started, n.d.) (langchain-chat-with-your-data, n.d.)

3.3.6. 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)

4.1. 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 doctors and a diagnostic chat bot.

4.2. Requirement Gathering:

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

4.2.1. 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.

4.2.2. 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.

4.3. Functional and non-functional requirements:

4.3.1. Functional requirements:

Login

Signup

Show articles

Show All doctors

Show doctor profile

Show article content

Search

A. Doctor

(i) Add article

(ii) Remove article

(iii) Show pending articles

(iv) accept/reject article

(v) Show my articles

(vi) Show all chats

(vii) Show chat content

- (viii) Remove chat
- (ix) Add certificate
- (x) Delete account
- (xi) Show and edit profile information
- (xii) Show and edit account information
- (xiii) Change password
- (xiv) Contact us

B. Patient

- (i) Like/unlike article
- (ii) Report article
- (iii) Like doctor
- (iv) Start chat
- (v) Remove chat
- (vi) Show all chats
- (vii) Talk to Sandra
- (viii) Show chat content
- (ix) Delete account
- (x) Change password

(xi) Show and edit account information

(xii) Contact us

C. Admin

(i) accept certificate verification

(ii) Reject certificate verification

(iii) Delete doctor

(iv) Show pending articles

(v) Accept/reject article

(vi) change password

(vii) contact us

4.3.2. Non-functional requirements:

A. Usability

4.4. Class Diagram:

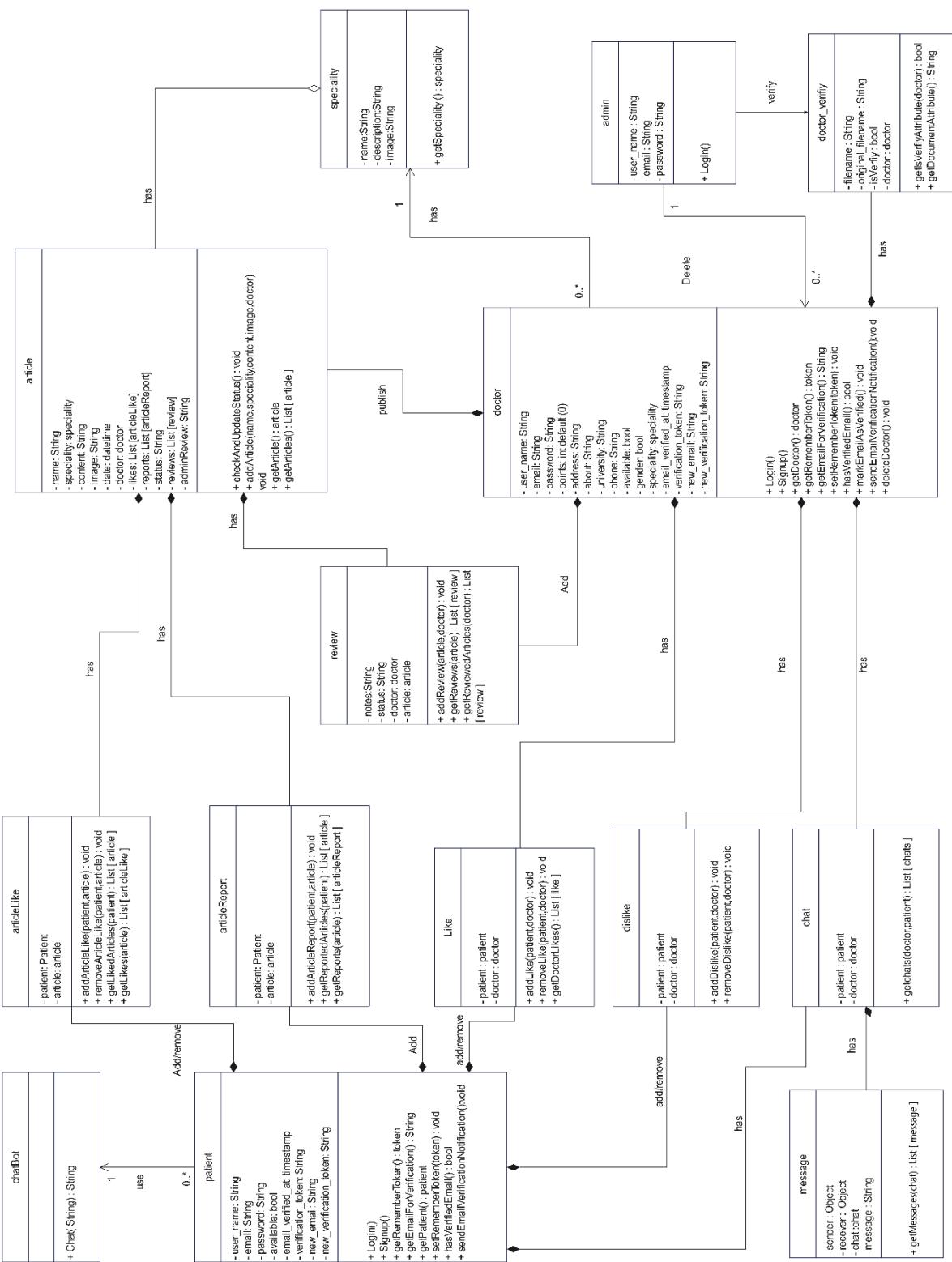


Figure 4.1 Class Diagram

4.5. Use case Diagram:

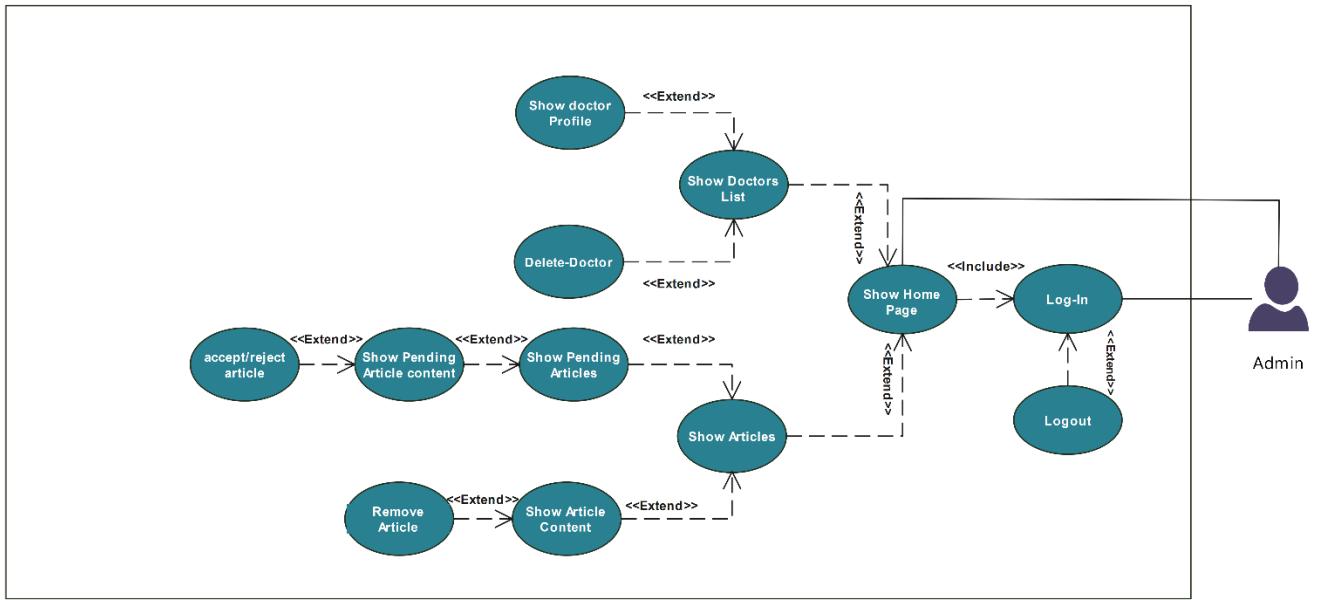


Figure 4.2 AdminUseCase (Doctors , Articles)

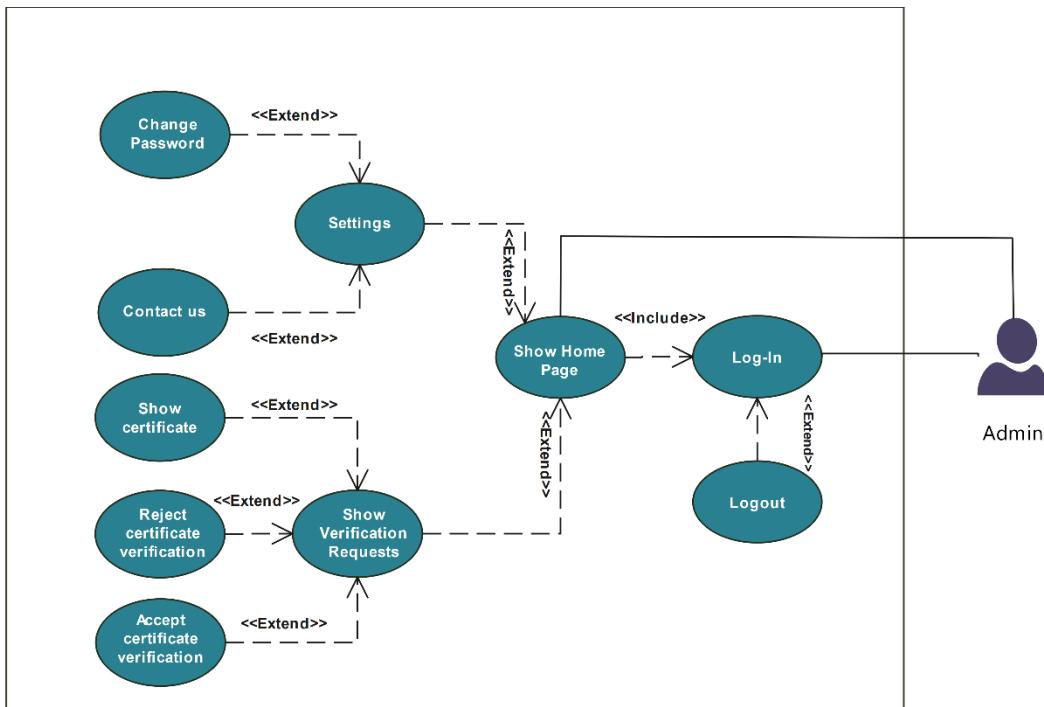


Figure 4.3 AdminUseCase (Settings,Verify)

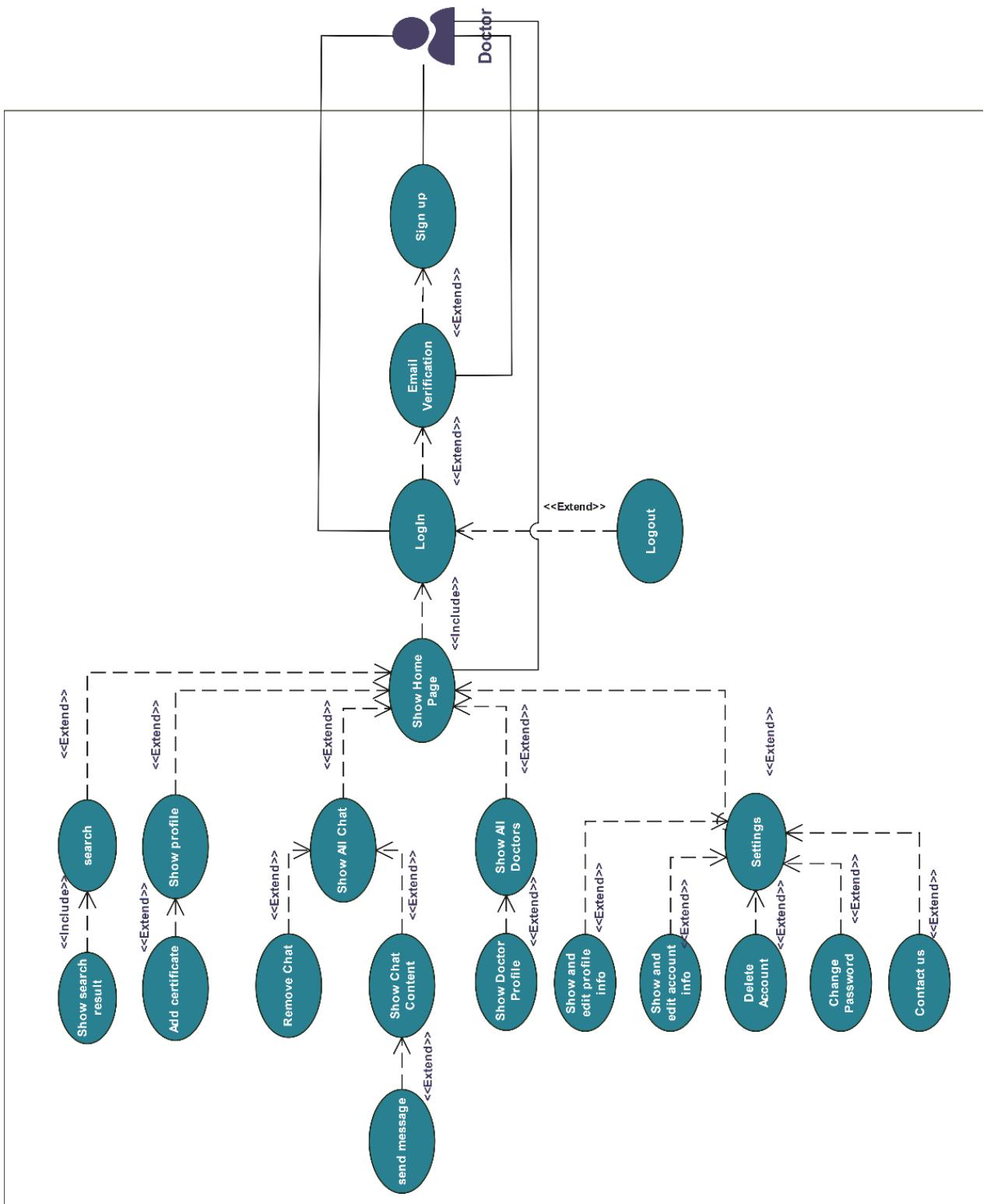


Figure 4.4 DoctorUseCase (Settings, Chat, Doctors, profile, search)

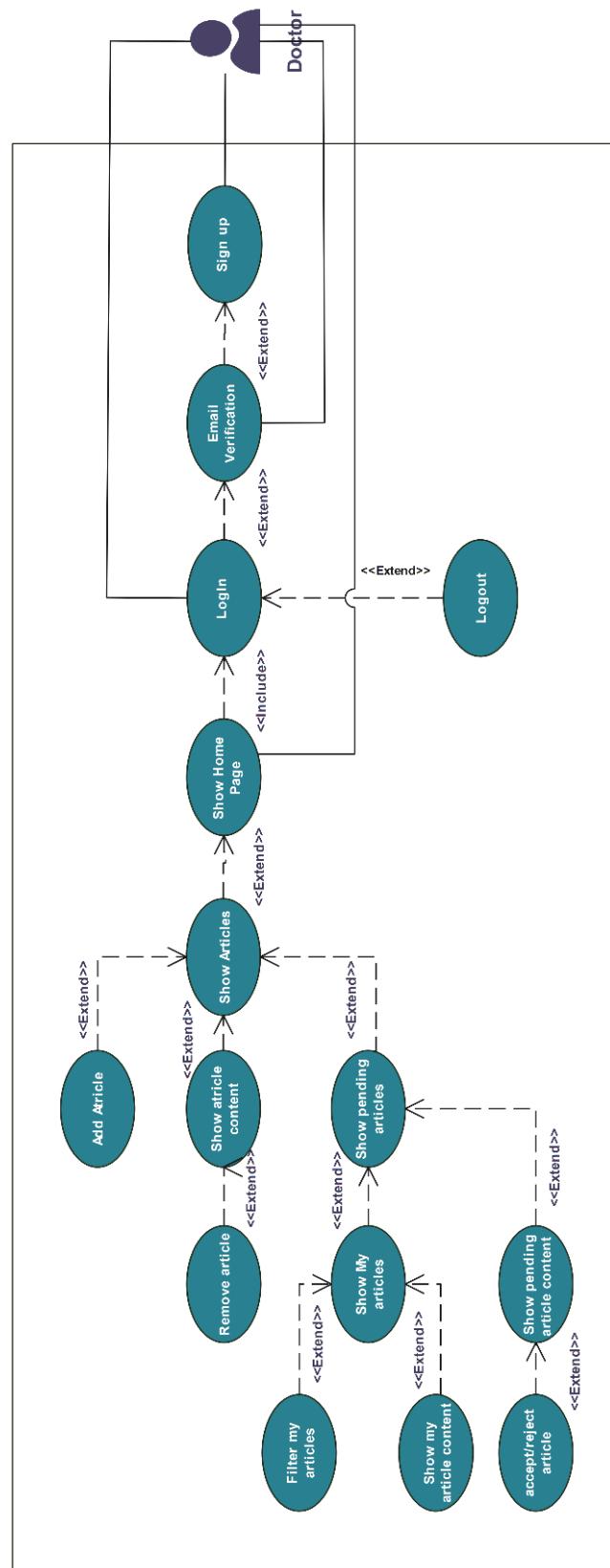


Figure 4.5 DoctorUseCase (articles)

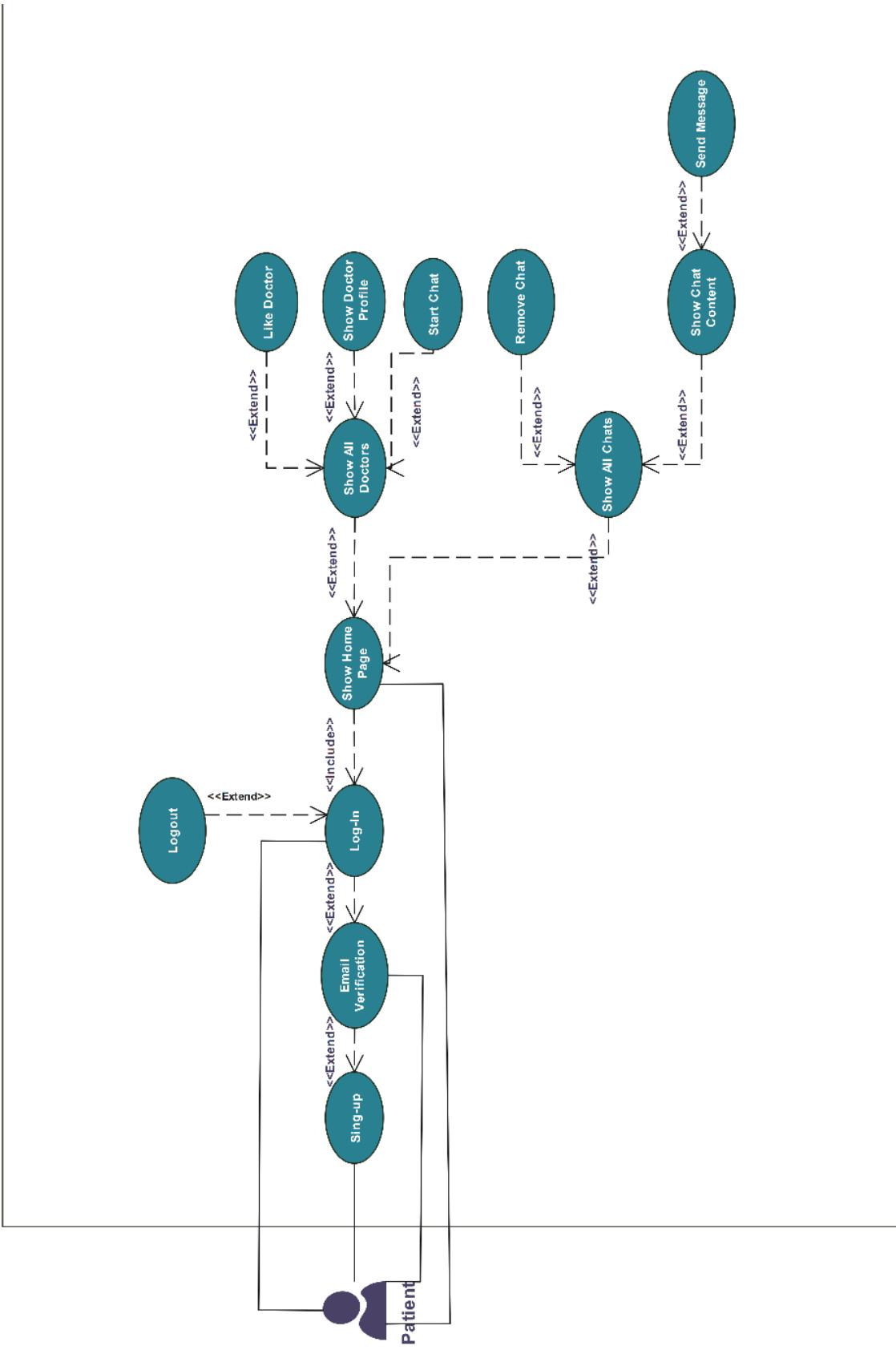


Figure 4.6 PatientUseCase (Doctors , Chats)

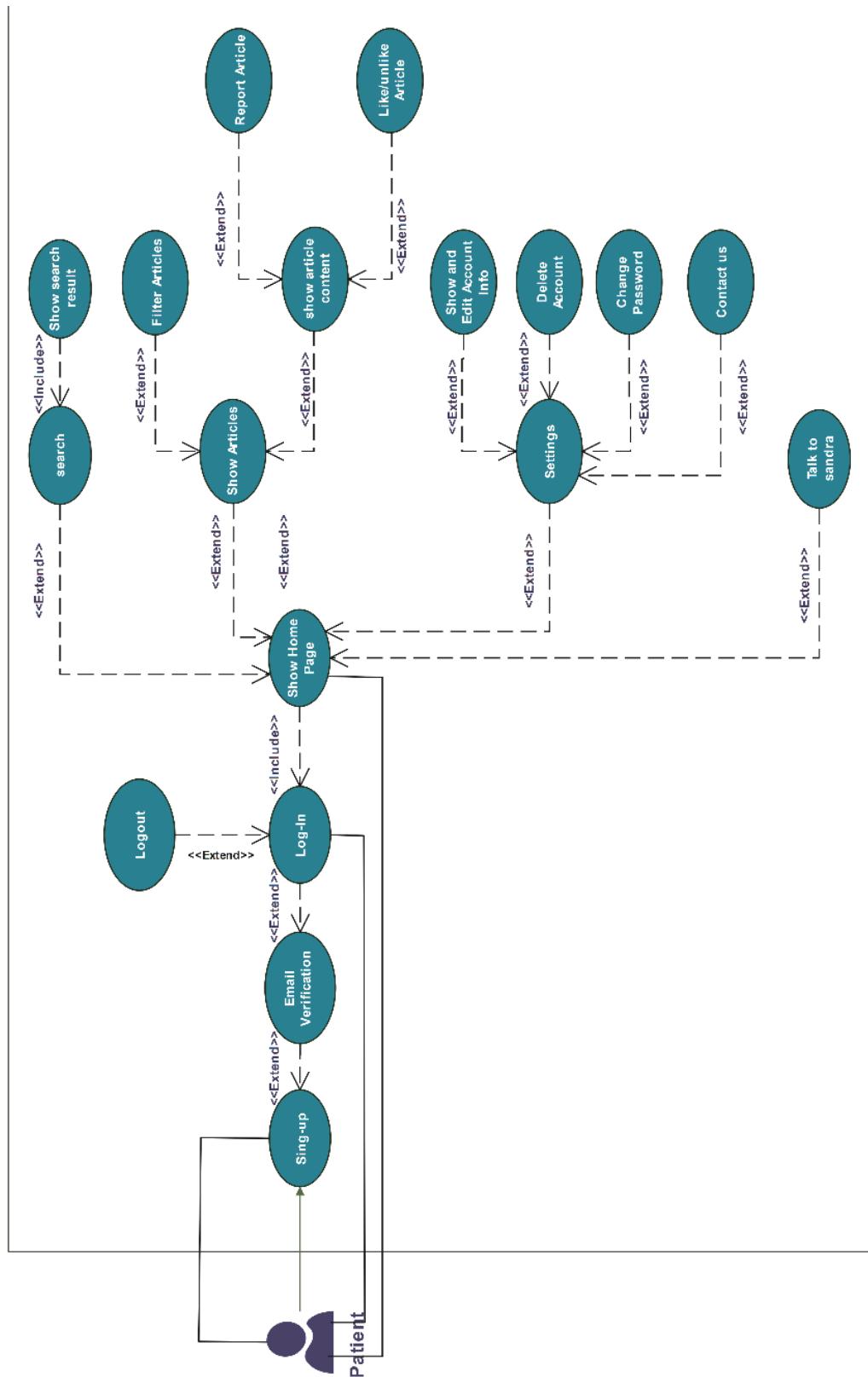


Figure 4.7 PatientUseCase (Search , Articles , Settings , chatbot)

4.6. Use case Specification:

Table 4.1 Show Articles Use Case Specification

UC-1	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:	

	<p>If the number of published articles exceeds a predefined threshold (9), the system provides pagination controls to navigate through the list.</p> <p>3. Error handling:</p> <p>The system gracefully handles potential errors, such as database connection issues or invalid data, and provides informative error messages to the user.</p>
--	--

Table 4.2 Filter Articles Use Case Specification

UC-2	Filter Articles	
Actor	Admin, Doctor, Patient, Guest User	
Brief Description	The user selects a category from a list of categories to filter the displayed articles on the articles page.	
Pre-conditions	1. User is on the articles page. 2. A list of categories is displayed on the page. 3. At least one article exists in the system.	
Post-conditions	The list of articles is updated to only show articles belonging to the chosen category.	
Flow of event	Actor	System
	clicks on a category in the list	
		System identifies the selected category.
		filters the list of articles based on the chosen category.
		Filter out reported articles from the retrieved list.

		updates the displayed list of articles
		highlights the selected category
Critical scenarios	1. Selected category has no articles: System displays an appropriate message informing the user that no articles belong to the chosen category. 2. Pagination: If the number of published articles exceeds a predefined threshold (9), the system provides pagination controls to navigate through the list. 3. Technical error occurs during filtering: System displays an error message and logs the error for debugging.	

Table 4.3 Show Article Content Use Case Specification

UC-3	Show Article Content	
Actor	Admin, Doctor, Patient, Guest User	
Brief Description	The user clicks on an article in the list on the articles page to view its full content and details. Related articles from the same category are also suggested for further exploration.	
Pre-conditions	1. User is on the articles page. 2. A list of articles is displayed.	
Post-conditions	1. A new page opens with the selected article's full content and details (title, author, date, category, etc.). 2. A list of suggested articles from the same category is displayed below the main article, with brief information like title, author, and a short excerpt.	
Flow of event	Actor	System
	clicks on an article in the list.	

		retrieves the selected article's full content and details.
		retrieves a list of suggested articles from the same category.
		displays a new page with: The full content and details of the selected article. A list of suggested articles with brief information.
Critical scenarios	<p>1. Article not found: System displays an error message indicating that the article cannot be found. System might offer options to search for the article or return to the articles page.</p> <p>2. Technical error occurs while fetching data: System displays an error message and logs the error for debugging.</p>	

Table 4.4 Like/Unlike Article Use Case Specification

UC-4	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.

	<p>1.3. The article is added to the patient's "Favorite" list .</p> <p>2. If the article is already liked:</p> <p> 2.1. The "Like" button changes appearance to indicate the article is not liked.</p> <p> 2.2. The total number of likes for the article is decremented.</p> <p> 2.3. The article is removed from the patient's "Favorite" list.</p>
Flow of event	Actor
	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	<p>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.</p> <p>2. Technical error occurs during like/unlike action: System displays an error message and logs the error for debugging.</p>

Table 4.4 Report article use Case Specification

UC-5	Report Article
Actor	Patient

Brief Description	patients on the article content page can report an article they consider problematic. This triggers an alert for confirmation, and upon confirmation, the system adds the article to the reported list and updates the reputation points of the publisher and reviewers.														
Pre-conditions	1. Patient is logged in. 2. Patient is on the article content page.														
Post-conditions	1. Upon report confirmation: 1.1. The article is added to the patient's "Reported Articles" and users are redirected to the articles page. 2. If the report count reaches 3: 2.1. The reputation points of the publishing doctor, and accepting reviewers are each decremented by 1.														
Flow of event	<table border="1"> <thead> <tr> <th>Actor</th><th>System</th></tr> </thead> <tbody> <tr> <td>clicks on the "Report" option in a dropdown menu on the article content page.</td><td></td></tr> <tr> <td></td><td>displays an alert message asking the patient to confirm reporting the article.</td></tr> <tr> <td>Clicks "Yes".</td><td></td></tr> <tr> <td></td><td>adds the article to the patient's "Reported Articles" list.</td></tr> <tr> <td></td><td>checks if the report count for the article reaches 3</td></tr> <tr> <td></td><td>If report count reaches 3: System decrements the reputation points of The doctor who published the article and Each doctor who reviewed the article with "accept".</td></tr> </tbody> </table>	Actor	System	clicks on the "Report" option in a dropdown menu on the article content page.			displays an alert message asking the patient to confirm reporting the article.	Clicks "Yes".			adds the article to the patient's "Reported Articles" list.		checks if the report count for the article reaches 3		If report count reaches 3: System decrements the reputation points of The doctor who published the article and Each doctor who reviewed the article with "accept".
Actor	System														
clicks on the "Report" option in a dropdown menu on the article content page.															
	displays an alert message asking the patient to confirm reporting the article.														
Clicks "Yes".															
	adds the article to the patient's "Reported Articles" list.														
	checks if the report count for the article reaches 3														
	If report count reaches 3: System decrements the reputation points of The doctor who published the article and Each doctor who reviewed the article with "accept".														

		closes the confirmation alert and redirects the patient to the articles page.
Critical scenarios		<p>1. Patient is not logged in: System displays an error message indicating the user needs to log in to report an article.</p> <p>2. Technical error occurs during report action or point update: System displays an error message and logs the error for debugging.</p> <p>3. User interaction with alert cancelled: System closes the confirmation alert without further action.</p>

Table 4.5 Delete Article Use Case Specification

UC-6	Delete Article	
Actor	Doctor (Published article), Admin	
Brief Description	An authorized user (doctor who published the article or admin) on the article content page can delete the article. This action triggers a confirmation alert, and upon confirmation, the system removes the article.	
Pre-conditions	<p>1. Authorized user (doctor who published the article or admin) is logged in.</p> <p>2. User is on the article content page.</p>	
Post-conditions	<p>1. Upon confirmation (Yes):</p> <p>1.1. The article is permanently deleted from the system.</p> <p>1.2. The user is redirected back to the articles page.</p>	
Flow of event	Actor clicks on the "Delete" option in a dropdown menu on the article content page.	System

		displays an alert message asking the user to confirm deleting the article.
	Clicks "Yes".	
		permanently deletes the article.
		redirects the user to the articles page.
Critical scenarios	<p>1. Unauthorized user attempts to delete: System displays an error message indicating the user does not have permission to delete the article. System might offer options to contact an administrator or return to the articles page.</p> <p>2. Technical error occurs during deletion: System displays an error message and logs the error for debugging. System might offer options to retry or contact support.</p> <p>3. User interaction with alert cancelled: System closes the confirmation alert without further action.</p>	

Table 4.6 Add Article Use Case Specification

UC-7	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	<p>1. Doctor is logged in and has a verified account with a valid certificate.</p> <p>2. Doctor is on the articles page.</p>

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:	

	<p>4.1. System displays an alert with specific error messages.</p> <p>4.2. Doctor remains on the article form page to make corrections and resubmit.</p>
--	--

Table 4.7 Show Pending Articles Use Case Specification

UC-8	Show Pending Articles	
Actor	Admin , Doctor(Verified)	
Brief Description	user views all articles currently awaiting a decision (status: pending or adminchoice).	
Pre-conditions	<p>1. user is logged in and has a verified account with a valid certificate if doctor.</p> <p>2. At least one article exists with a status of “pending or adminchoice”.</p> <p>3. user is in Articles page</p>	
Post-conditions	A list of all articles with a status of "adminchoice" for admin or “pending” for doctors is displayed, including all relevant information (e.g., category, Author name , content).	
Flow of event	Actor	System
	Click on Pending Articles button	
		Retrieves a list of articles with a status of "adminchoice" for admin or “pending” for doctor
		Renders the list of pending articles on the "pending articles" page, displaying the specified information for each article.
Critical scenarios	<p>1. Doctor is not logged in:</p> <p>1.1. System redirects the doctor to the login page.</p>	

	<p>2. Doctor's account is not verified:</p> <p> 2.1. System displays an error message informing the doctor that their account needs verification to submit articles.</p> <p>3. No pending articles:</p> <p>If there are no articles with a status of "adminchoice" or "pending", the system displays a message indicating that no pending articles are found.</p> <p>4. Pagination:</p> <p>If the number of pending articles exceeds a predefined threshold (9), the system provides pagination controls to navigate through the list.</p> <p>5. Error handling:</p> <p>The system gracefully handles potential errors, such as database connection issues or invalid data, and provides informative error messages to the admin.</p>
--	--

Table 4.8 accept/reject article Use Case Specification

UC-9	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	<p>1. User is logged in.</p> <p>2. User is on the pending article content page.</p> <p>3. The user is not the doctor who published the article (applicable to doctors only).</p>	
Post-conditions	<p>1. Article status is changed.</p> <p>2. Success message is displayed to the user.</p> <p>3. System redirects the user back to the pending articles page.</p>	
Flow of event	Actor clicks the "Accept" or "Reject" button.	System

		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 Show All Doctors Use Case Specification

UC-10	Show All Doctors
Actor	Admin, Doctor, Patient, Guest User
Brief Description	All users can access a page listing doctors registered on the platform. This page displays basic information about each doctor.
Pre-conditions	The website is accessed by user.

Post-conditions	A list of all Doctors is displayed. Each doctor includes some relevant information.	
Flow of event	Actor	System
	Clicks on the "Doctors" button in the navbar or footer or Home page.	
		retrieves the list of registered doctors.
		displays the list of doctors on the page, including basic information for each doctor.
Critical scenarios	1. Technical error occurs during data retrieval or display: System displays an error message and logs the error for debugging. 2. No doctors are registered: System displays an informative message indicating no doctors are currently available.	

Table 4.10 Show Doctor Profile Use Case Specification

UC-11	Show Doctor Profile	
Actor	Admin, Doctor, Patient, Guest User	
Brief Description	All users can access the doctor's profile page. This page displays detailed information about the doctor and their published articles.	
Pre-conditions	User is on the "Doctors" page.	
Post-conditions	The doctor's profile page opens, displaying: Doctor information, Articles by this doctor	
Flow of event	Actor	System
	clicks on a doctor's name from the "Doctors" list	

		retrieves the selected doctor's profile information and list of published articles.
		System displays the doctor's profile page with the retrieved information: Doctor details. List of published articles.
Critical scenarios	1. Technical error occurs during data retrieval or display: System displays an error message and logs the error for debugging. 2. Selected doctor does not exist: System displays an error message indicating the doctor's profile is unavailable.	

Table 4.11 Search Use Case Specification

UC-12	Search	
Actor	Admin, Doctor, Patient, Guest User	
Brief Description	Any user can initiate a search for doctors or articles using the search bar in the website's navbar. The search interface allows users to specify their search terms and filter the results by type (all, doctors, articles).	
Pre-conditions	The website is accessed by user.	
Post-conditions	The search results page opens, displaying: Search filters (Three options: "All" (selected by default, "Doctors", "Articles"), Search results list.	
Flow of event	Actor	System
	clicks on the search icon in the navbar.	
		displays the search alert.

	enters their search term and selects the search type (doctors, articles)	
	clicks "Search".	
		<p>retrieves search results based on the entered keyword and selected filter type:</p> <p>If "Doctors" is selected, only matching doctors are retrieved.</p> <p>If "Articles" is selected, only matching articles are retrieved.</p> <p>If both are selected, all matching doctors and articles are retrieved.</p>
		displays the search results page with the retrieved results and filter options.
Critical scenarios	<ol style="list-style-type: none"> 1. Technical error occurs during data retrieval or display: System displays an error message and logs the error for debugging. 2. No results found: System displays a message indicating no results were found for the search term and selected filter. 	

Table 4.12 Signup Use Case Specification

UC-13	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 4.13 Email Verification Use Case Specification

UC-14	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	<ol style="list-style-type: none"> User account becomes active , granting them access to the platform's functionalities (Login) A success message is displayed to the user 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	<ol style="list-style-type: none"> The verification email is missing essential information like the verification link or the information provided is incorrect Email Spoofing : A malicious actor sends a fake verification email impersonating your platform , attempting to trick the user into clicking a malicious link 	

Table 4.14 Login Use Case Specification

UC-15	Login
Actor	Doctor, 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 4.15 Talk To Sandra Use Case Specification

UC-16	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

	Actor	System
Flow of event	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 4.16 Send Message Use Case Specification

UC-17	Send Message
Actor	Patient , Doctor
Brief Description	This service make the users (Doctor or Patient) allowed to communicate with other and exchange the information and allowed the doctors to diagnoses the different cases that will expand his experience
Pre-conditions	1. must be logged in and have activated account(Doctor ,Patient) 2. must select the Doctors page (Patient) 3. must press on the Start Chatting button(Patient) 4. write something and press “Send” button (Patient , Doctor)
Post-conditions	The message has been token for another person and display it on the screen , Refresh the page dynamic to upload a new message

	Actor	System
Flow of event	Write a text inside the “Message” input	
	Press on the “Send” button	
		Create a new record for the new message with identifier to retrieve the message
		Refresh the page dynamic for both users Sender and Receiver
		Show a new page content
Critical scenarios	1. Doctor or patient doesn't exist yet 2. Security : Send a lot of messages at the same time it might be DOS Attack and the server will prevent it	

Table 4.17 Show Chat Content Use Case Specification

UC-18	Show chat content	
Actor	Patient , Doctor	
Brief Description	Chat contain the messages that both Doctor and patient send for each other	
Pre-conditions	1. must be logged in and have activated account(Doctor ,Patient) 2. select Chat button on navigation bar (Doctor , Patient) 3. select specific chat that user want to show its content	
Post-conditions	The page shows a new messages after reload from database	
Flow of event	Actor	System
	Select specific chat from “Chats” Page	

		Convert the user for specific URL that depend on the identifier of the chat
		Represent all the messages for this chat
		Show a new page content
Critical scenarios	1. Technical Error : Delete the chat from database without order of user 2. Technical Error : Loss information when api represent the information	

Table 4.18 Add Certificate Use Case Specification

UC-19	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	

4.7. Activity Diagram:

4.7.1. Add Article :

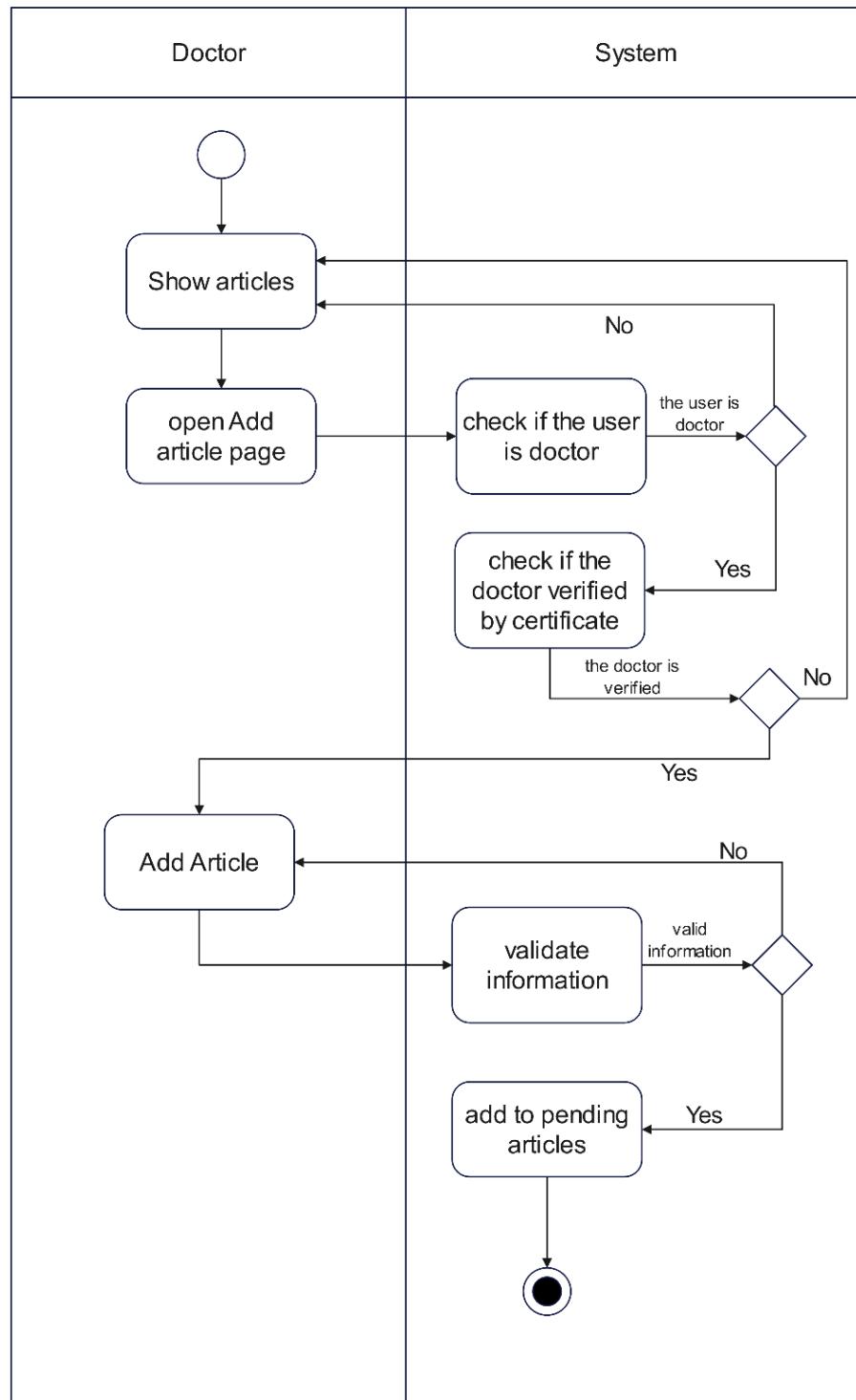


Figure 4.3 Add Article Activity Diagram

4.7.2. accept/reject Article :

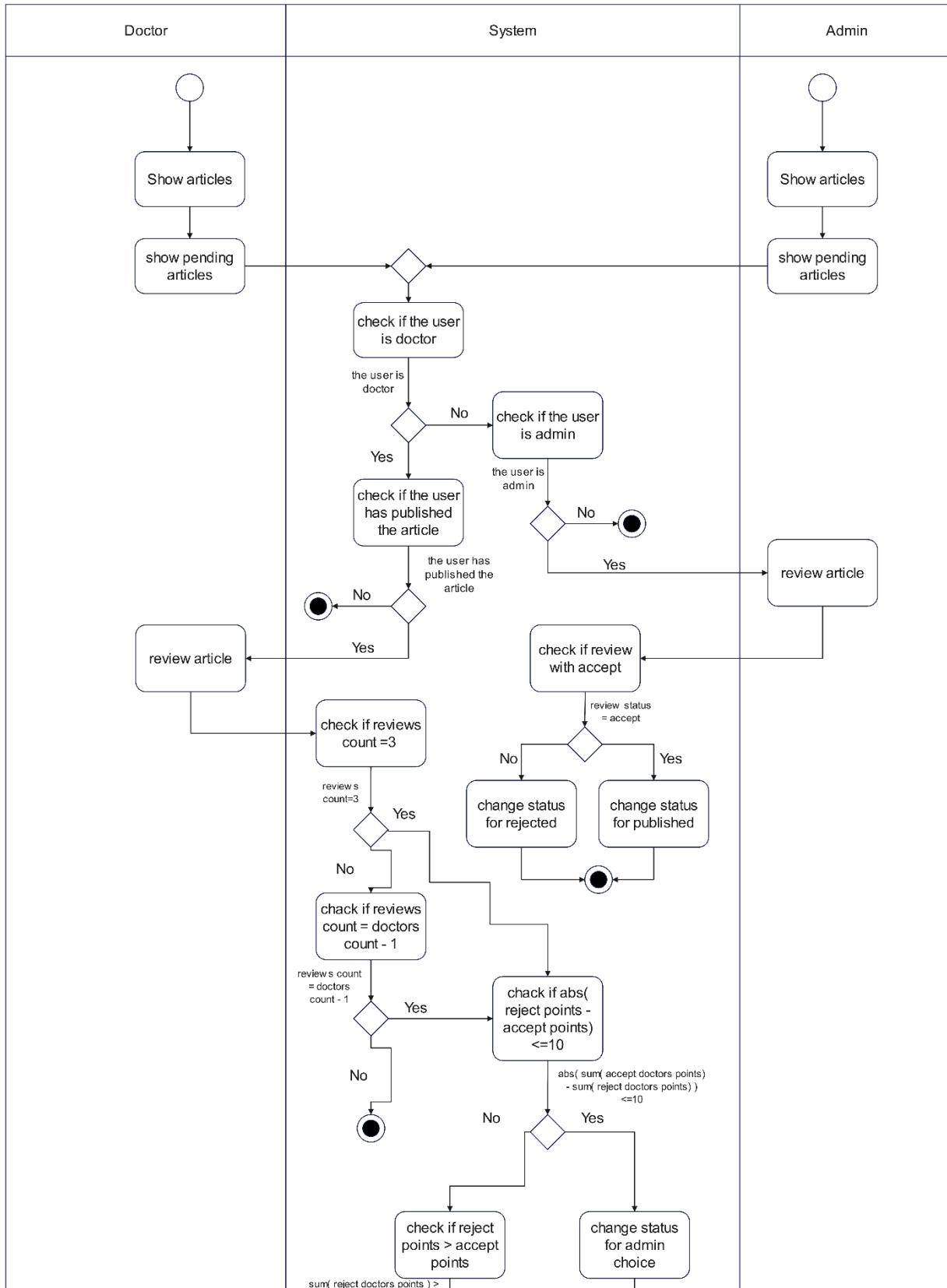


Figure 4.4 accept/reject Article Activity Diagram

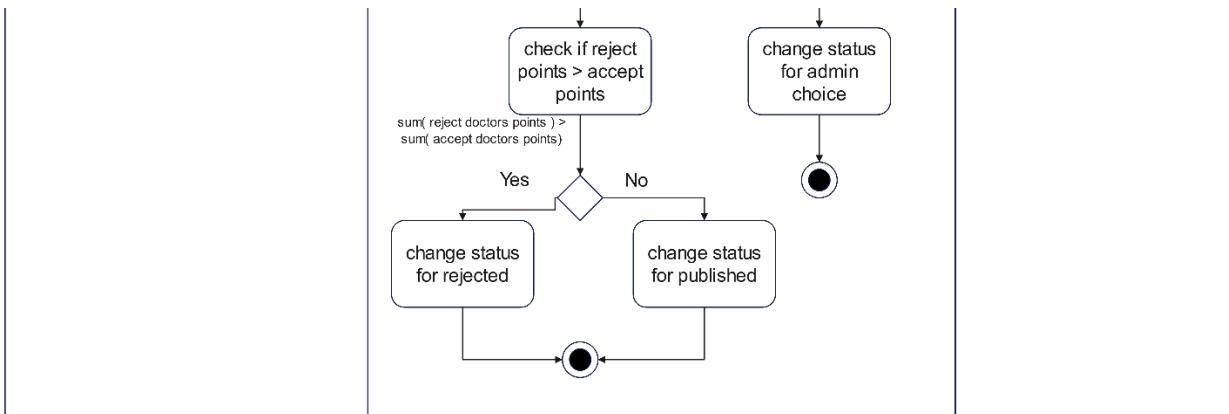


Figure 4.5 accept/reject article Activity diagram

4.8. Sequence Diagram:

4.8.1. Start Chat :

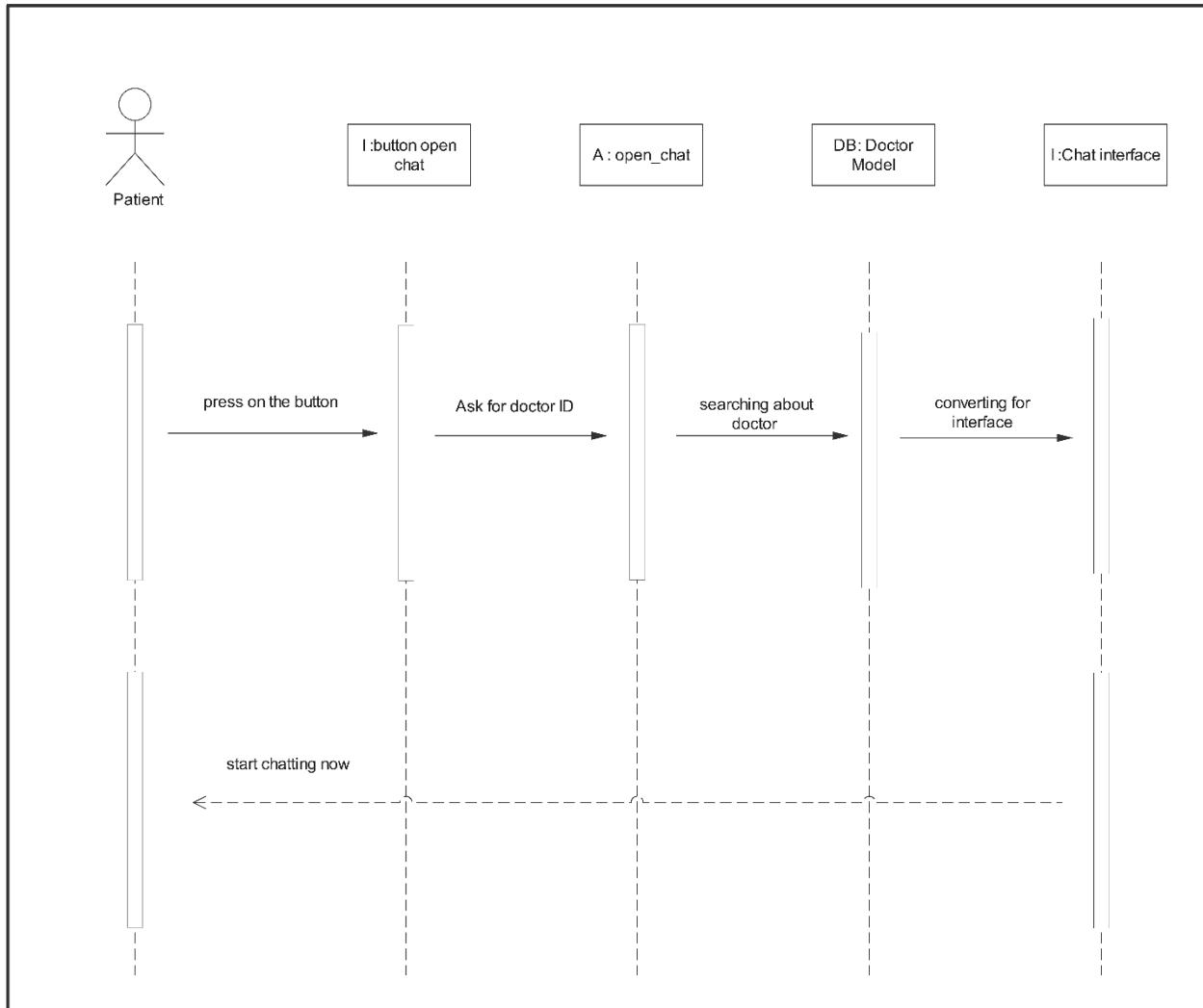


Figure 4.6 Start Chat Sequence Diagram

4.8.2. Upload and Verify certificate :

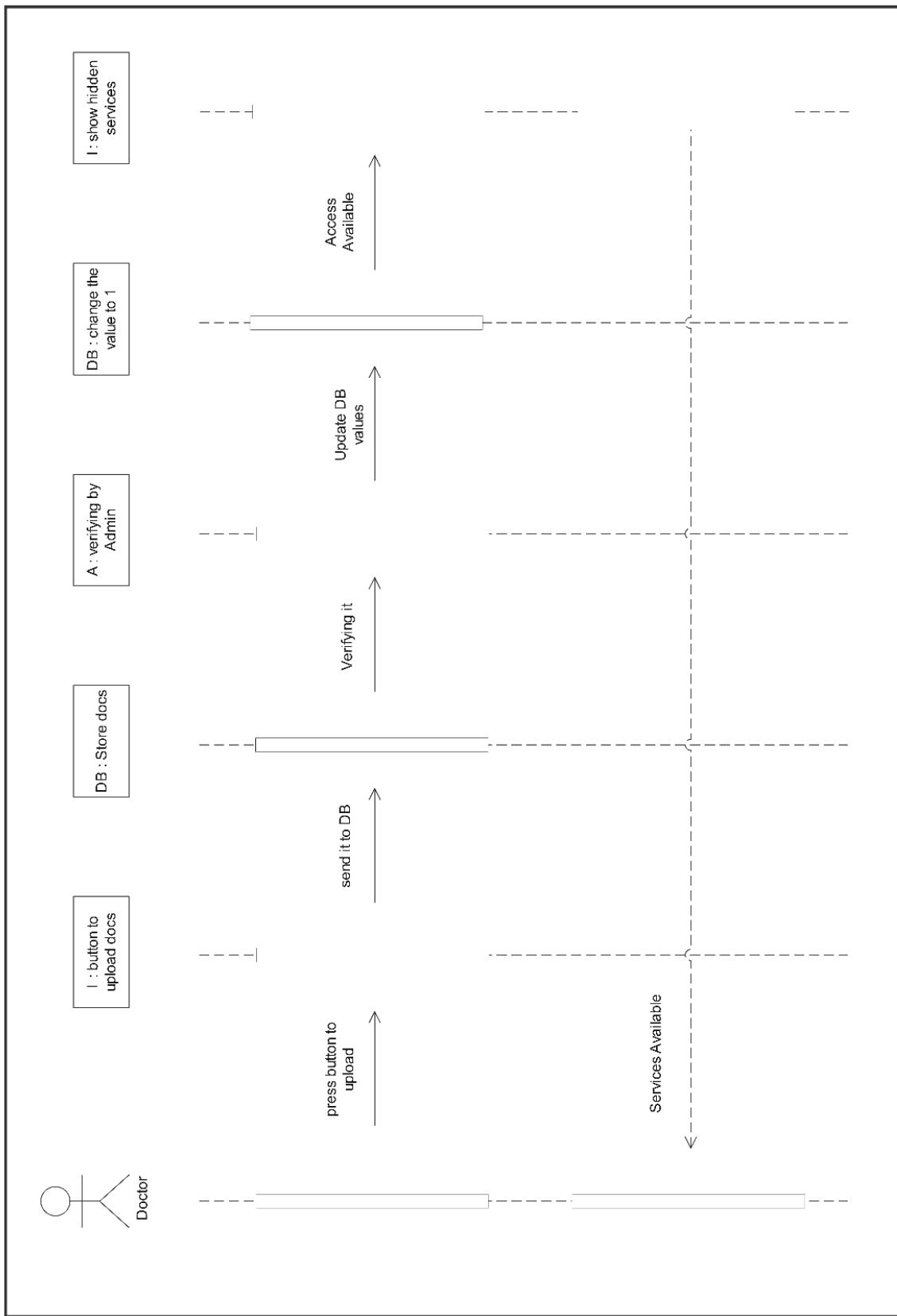


Figure 4.7 upload and verify certificate Sequence Diagram

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.

5.1. Block Diagram :

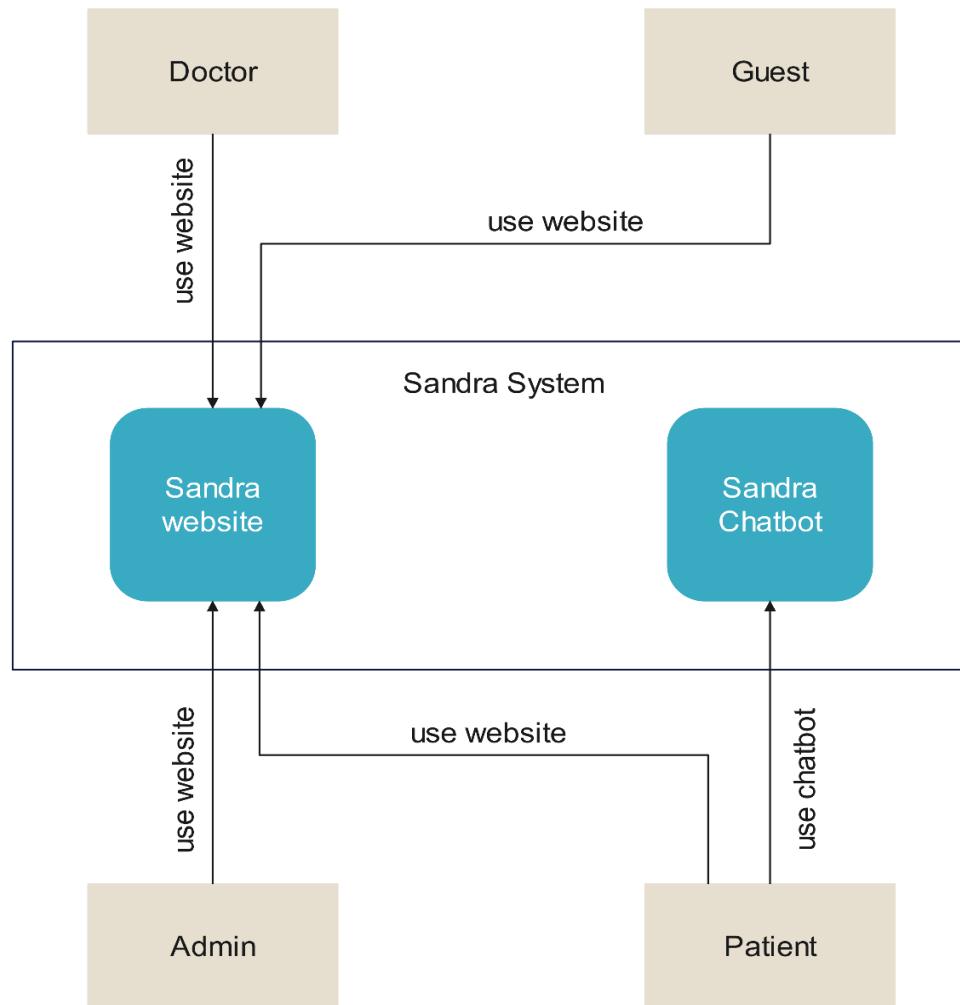


Figure 5.1 Block Diagram

5.2. Entity Relationship Diagram (ERD) :

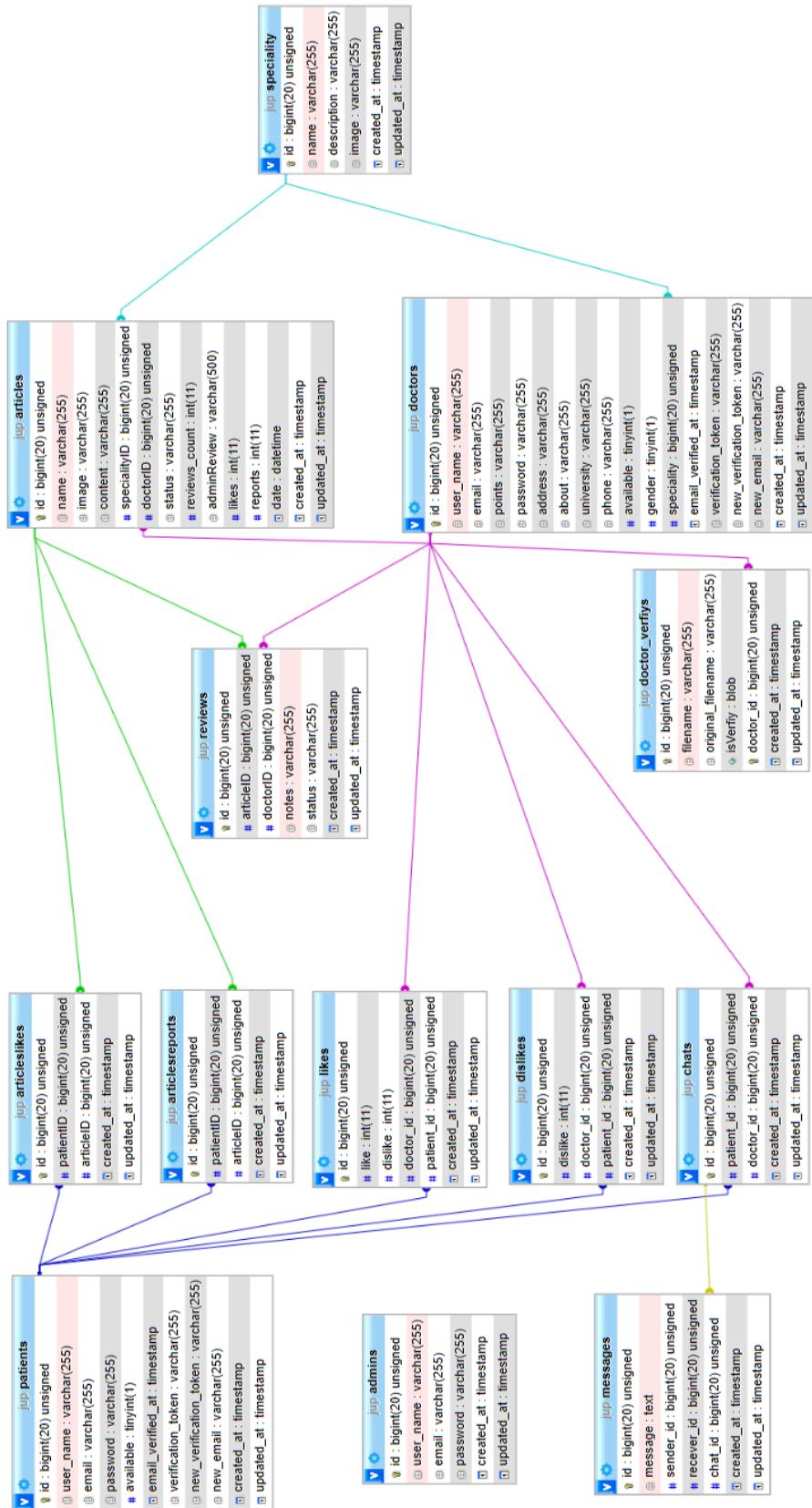


Figure 5.2 ERD

5.3. Interface Design :

5.3.1. SignUp:

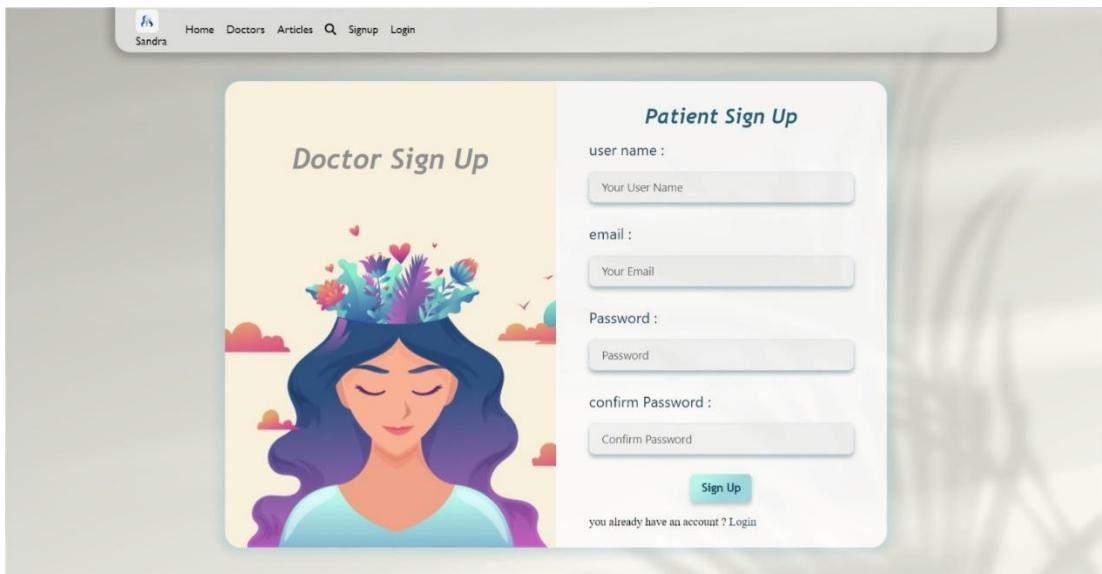


Figure 5.3 SignUp Interface

5.3.2. Login:

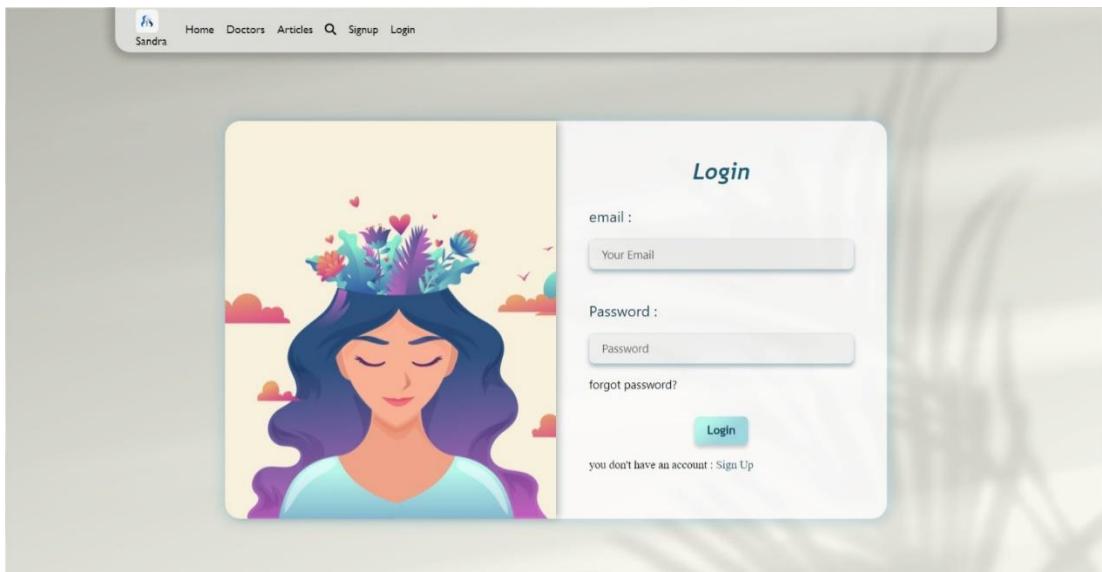


Figure 5.4 Login Interface

5.3.3. Home Page:

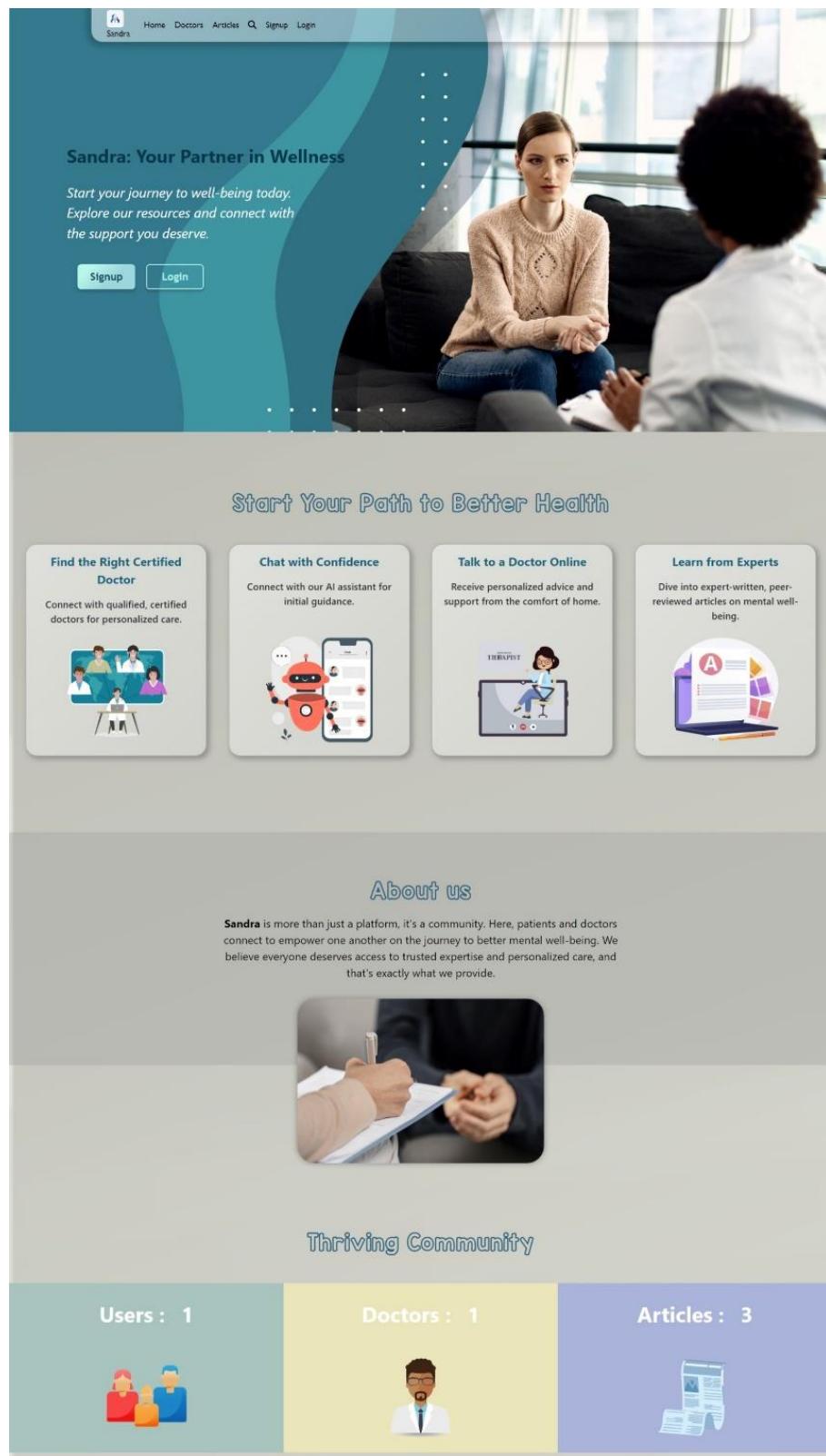


Figure 5.5 Home Page Interface

Verified Expertise

Explore our trusted doctors and connect with the support you deserve.



doctor1 
 Disorders
 points : 0 likes : 0
[profile](#)

[All Doctors](#)

Curated Content

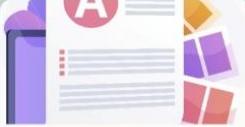
Grow & Thrive Empower yourself with the latest knowledge and practical tips. Read our top articles chosen to inspire and guide your personal growth.



s1
 doctor1
 Youth Mental Health

In the quiet recesses of my world, I sat, an enigma to those around me. Seeking comfort felt foreign, and when distress gripped me, I remained untouched by the outstretched hands ...

2024-02-25 17:18:01  0



q
 doctor1
 Disorders

In the quiet recesses of my world, I sat, an enigma to those around me. Seeking comfort felt foreign, and when distress gripped me, I remained untouched by the outstretched hands ...

2024-02-25 17:20:04  0



qfs
 doctor1
 Prevention & Wellness

jnflk nfd mdf'slkvn dsf'mv lkdfsvn;l
 ndsmf gld n dflkanv dk fngvk
 ndfkjvnkjdfsgkj nkjdsngr kjndsfkj
 nkdfjfmv kjsnfkjvnkjdfgkjnjdfkjgnv
 lkdfmfvkjngdskfjgkjdfgkjdnfgkjnsf...

2024-02-25 17:20:22  0

[All Articles](#)

SANDRA

if you care about your mental health and you want control on your nerve just be quiet



QUICK LINKS

- [Home](#)
- [Login](#)
- [Sign Up](#)

SERVICES

- [Articles](#)
- [Doctors](#)
- [Talk to Sandra](#)

CONTACT

Sandra@gmail.com
 +963900547937

© 2024 Copyright SandraGroup

Figure 5.6 Home Page Interface

5.3.4. Doctors (admin):



Figure 5.7 Doctors Interface

5.3.5. DoctorProfile:

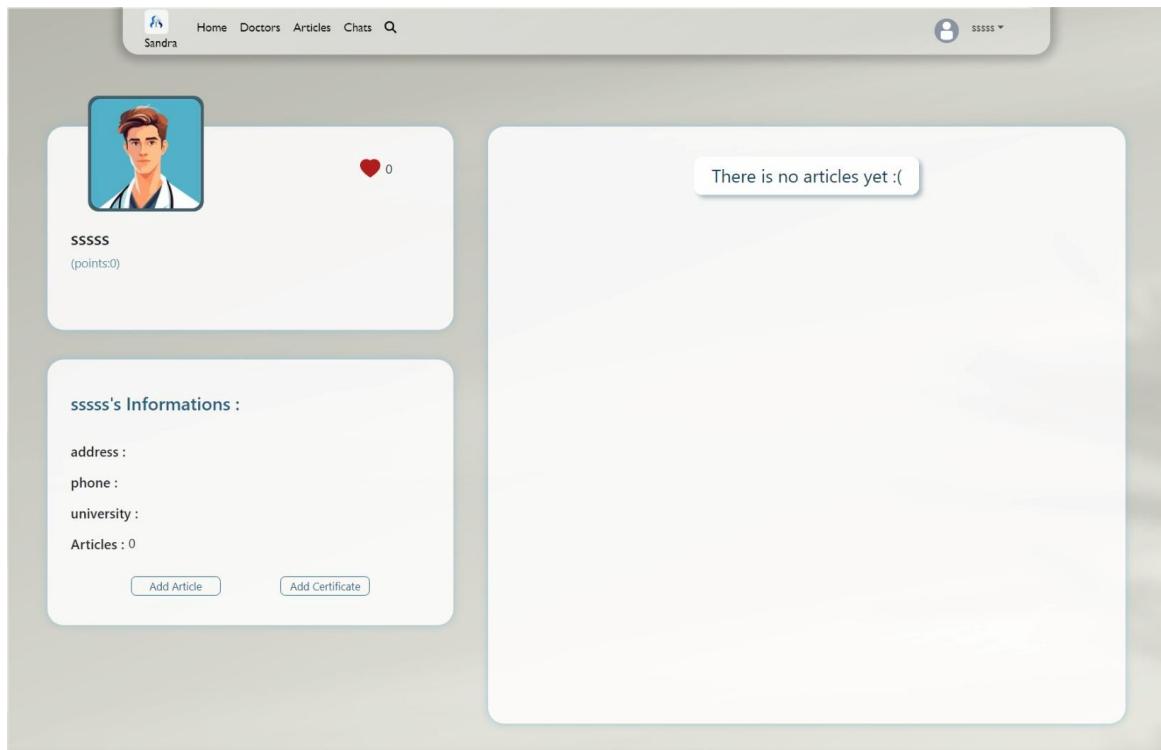


Figure 5.8 Doctor Profile Interface

5.3.6. Search:

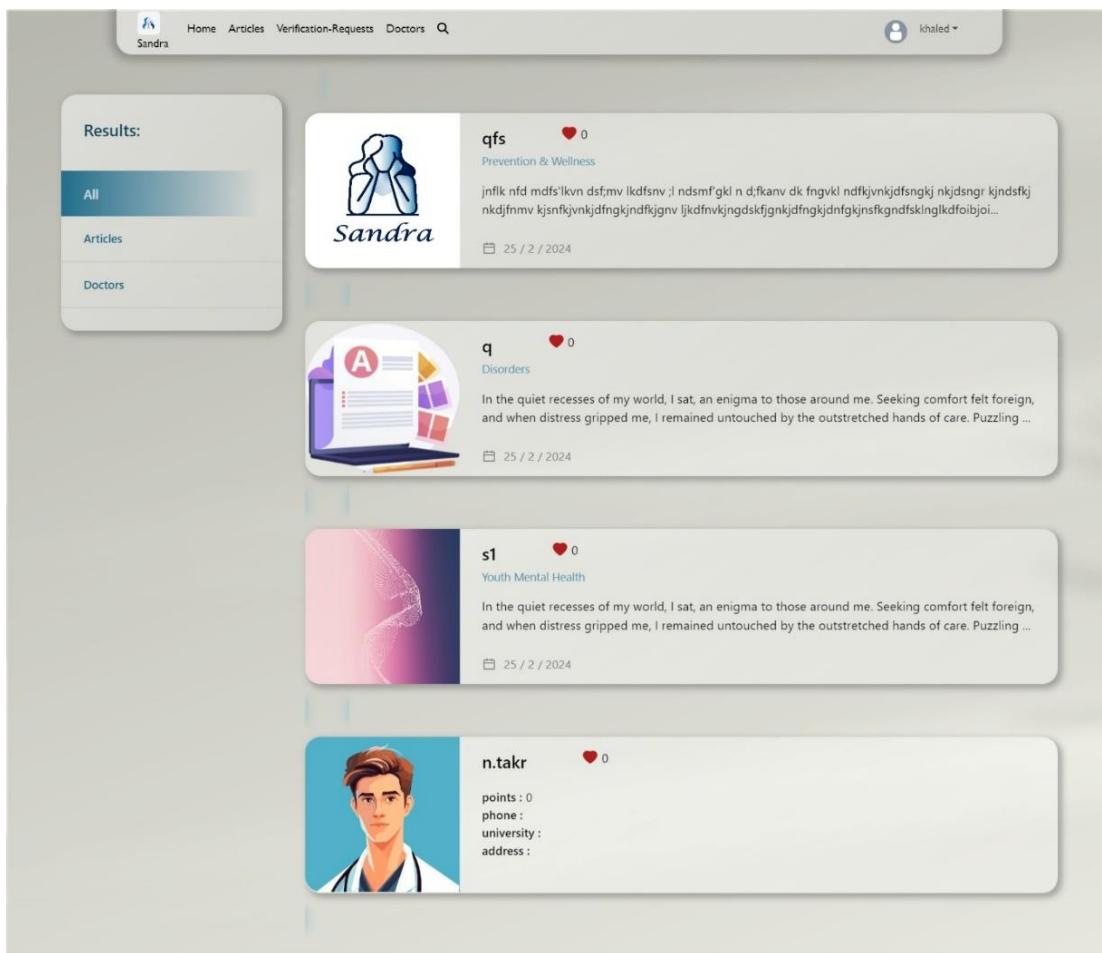


Figure 5.9 Search Interface

5.3.7. ChatBot (Sandra):

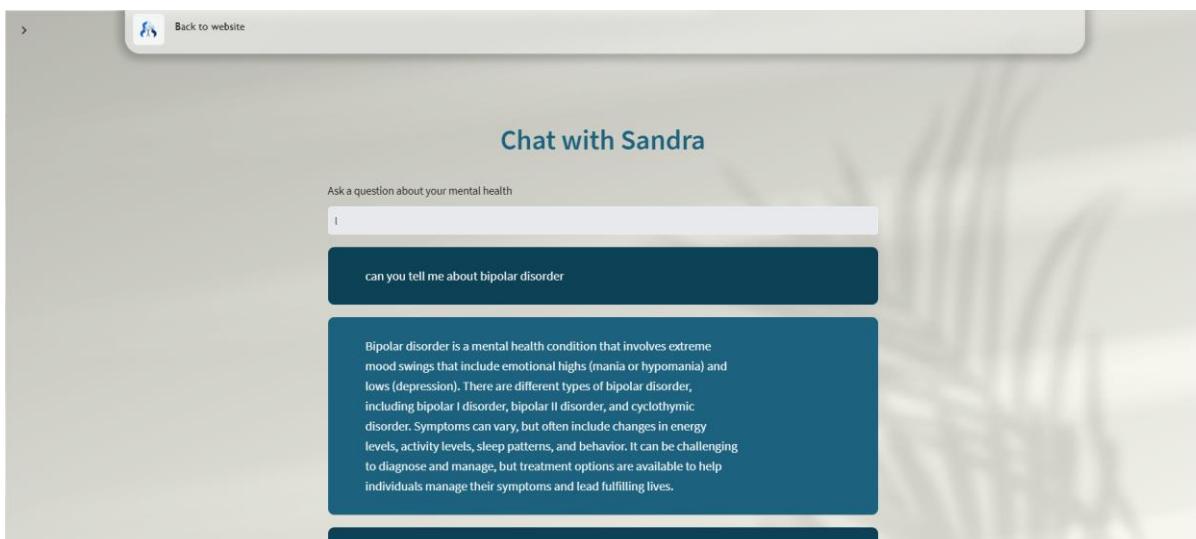


Figure 5.10 ChatBot (Sandra) Interface

5.3.8. Settings:

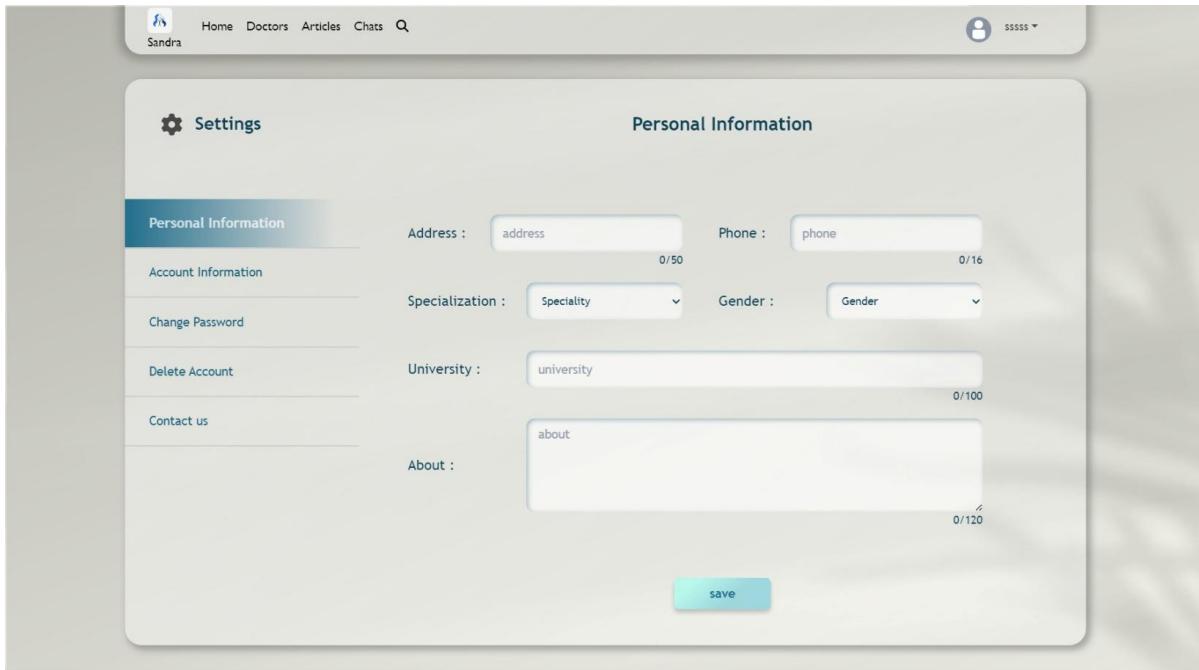


Figure 5.11 Settings Interface

5.3.9. Doctors (patient,Doctor):

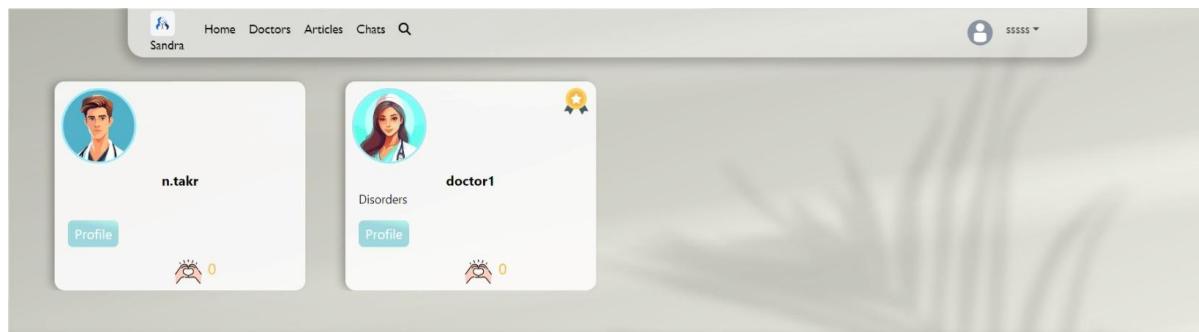


Figure 5.12 Doctor Interface

5.3.10. Add Certificate Form:

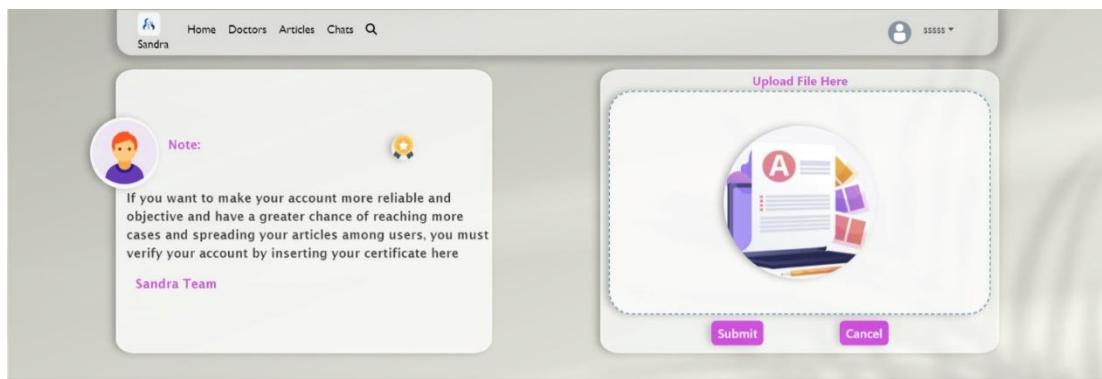


Figure 5.13 Add Certificate Form Interface

5.3.11. Articles:

The screenshot shows the 'Articles' section of a digital platform. At the top, there's a navigation bar with links for Home, Doctors, Articles, Chats, and a search icon. On the right, there's a user profile for 'Sandra' and a dropdown for 'doctor1'. Below the navigation, a large central box is titled 'Discover Nice Articles here' with a sub-instruction: 'There is hope. With the right support, you can overcome your challenges and live a happy and fulfilling life.' It also says 'Embark on an Article Category Journey to Uncover a Universe of Relevant Knowledge :'. A 'Pending Articles' button and an 'Add new Article' button are at the top right of this box. Below this, a horizontal menu lists categories: All, Disorders, Stress Management, Youth Mental Health, Relationships, Self-Improvement, Elderly Mental Health, Therapy, and Psychology Basics. Underneath, two more categories are listed: Culture & Mental Health and Prevention & Wellness.

The main feature is a large purple rectangular card for an article titled 'Relationships'. It features a central illustration of a person meditating with floating lightbulbs and documents. Below the illustration, the text reads 'Relationships' and 'Improving personal connections and communication'. At the bottom of this card are icons for a laptop and a smartphone.

Below this central card, there are three smaller article cards, each with a thumbnail image, author information, category, content snippet, and timestamp. The first card is by 's1' (doctor1) under 'Youth Mental Health'. The second is by 'q' (doctor1) under 'Disorders'. The third is by 'qfs' (doctor1) under 'Prevention & Wellness'. Each card shows a truncated version of the full article content.

Figure 5.14 Articles Interface

5.3.12. Add Article Form:

The screenshot shows the 'Add Article' form. At the top right is a user profile icon and the name 'sssss'. The main area features a large image of a desk setup with a monitor, keyboard, mouse, and smartphone, overlaid with the text 'EMPOWER MINDS, INSPIRE CHANGE. YOUR INSIGHTS CAN SHAPE A HEALTHIER WORLD.' Below this is the form itself, which includes fields for 'Title' (with a placeholder 'Title'), 'Category' (a dropdown menu showing 'Select a category'), 'Image' (a button labeled '+Add'), and 'Content' (a file input field showing 'Select text file' and 'No file chosen'). At the bottom are 'Submit' and 'cancel' buttons.

Figure 5.15 Add Article Form Interface

5.3.13. Verify requests:

The screenshot shows the 'Verify Request' interface. At the top right is a user profile icon and the name 'khaled'. The main area displays a request from a user named 'khaled' (represented by an icon of a person with a gear). The request details are: Name: dr.ssss and Request: verify. To the right of the details are three buttons: 'See File' (with a file icon), 'Accept' (green button), and 'Reject' (red button). At the bottom left is a footer with the text 'Admin khaled' and a small profile icon.

Figure 5.16 Verify request Interface

5.3.14. My Articles:

The screenshot shows a user interface for managing articles. At the top, there is a navigation bar with links for Home, Doctors, Articles, Chats, and a search icon. On the right, there is a user profile for 'doctor1'.

The main area displays a list of four articles under the heading 'Types :'. The articles are listed vertically, each with a thumbnail image, a title, a category, a brief description, a date, and a status indicator.

- s1**
Youth Mental Health
In the quiet recesses of my world, I sat, an enigma to those around me. Seeking comfort felt foreign, and when distress ...
25 / 2 / 2024 published
- werv wrg**
Psychology Basics
jnflk nfd mdfs'lkvn dsf,mv lkdfsrv ;l ndsmf'gkl n d,fkanv dk fngvkl ndfkjvnkjdfsngk jnkjdsngr kjndsfkj nkdfjnmv k...
25 / 2 / 2024 rejected
- q**
Disorders
In the quiet recesses of my world, I sat, an enigma to those around me. Seeking comfort felt foreign, and when distress ...
25 / 2 / 2024 published
- qfs**
Prevention & Wellness
jnflk nfd mdfs'lkvn dsf,mv lkdfsrv ;l ndsmf'gkl n d,fkanv dk fngvkl ndfkjvnkjdfsngk jnkjdsngr kjndsfkj nkdfjnmv k...
25 / 2 / 2024 published

A sidebar on the left shows a list of article types: All, Pending, Published, and Rejected. The 'All' option is currently selected.

Figure 5.17 My Articles Interface

5.3.15. Pending Articles:

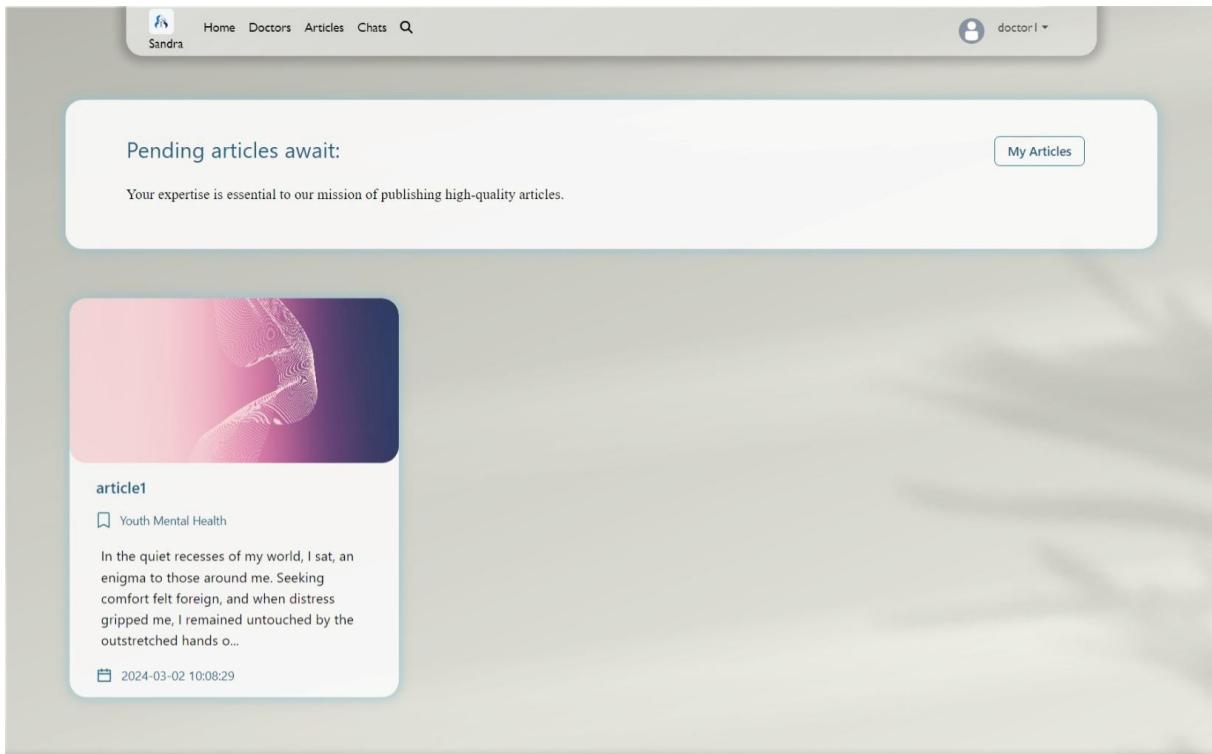


Figure 5.18 Pending Articles Interface

5.3.16. Chat:

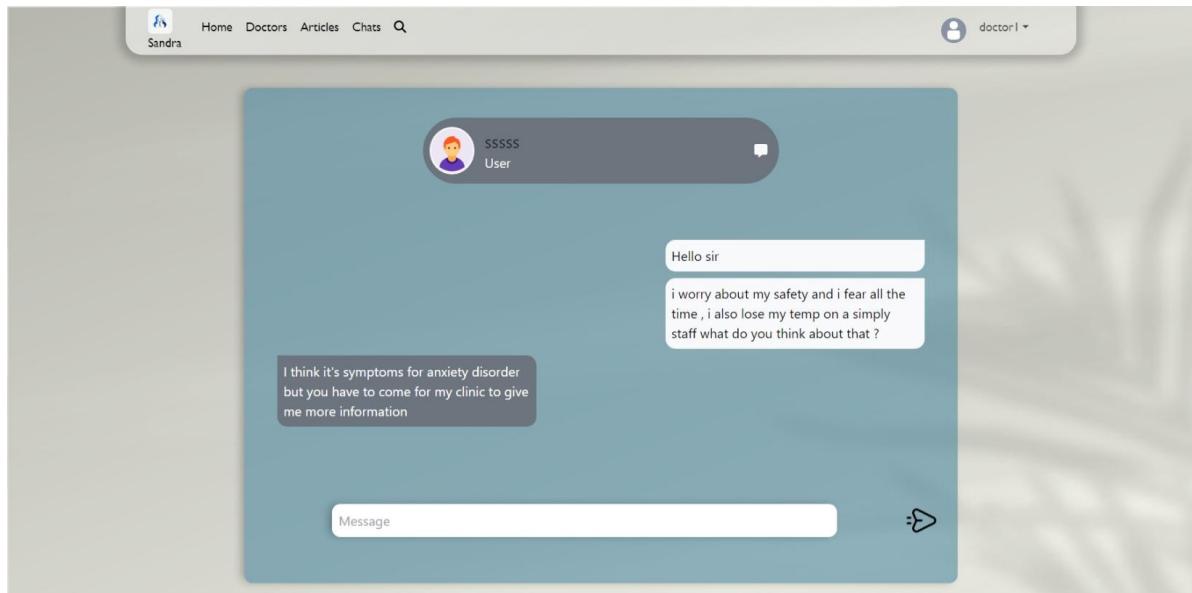


Figure 5.19 Chat Interface

5.3.17. Article Content:

The screenshot shows a user interface for managing article content. At the top, there's a navigation bar with icons for profile (Sandra), Home, Articles, Verification-Requests, Doctors, a search bar, and a user dropdown (Khaled). Below the navigation is a large, abstract background image with a pink-to-purple gradient and a wavy, translucent pattern.

article1

>Youth Mental Health

In the quiet recesses of my world, I sat, an enigma to those around me. Seeking comfort felt foreign, and when distress gripped me, I remained untouched by the outstretched hands of care.

Puzzling those who sought to engage, my responses were distant, my interactions minimal. Even in turmoil, I made no effort to seek solace.

Emotionally distant, I moved through life with minimal connection, joy elusive, my face a mask hiding inner turmoil.

Rooted in a past of neglect and instability, each transition severed fragile bonds. Raised without warmth, hungering for affection, I existed adrift.

My troubled behavior echoes my tumultuous journey, a silent reflection of my past. Despite my age, emotional growth remained stunted, trapped in the echoes of silence. Can you tell me if I have any mental disorders? Can you please provide me with the diagnostic criteria for the previous disorder that I may have? I'd like to understand the specific symptoms and characteristics outlined in the criteria

Date : 2 / 3 / 2024

Notes :

first review :
status : reject
notes : bad content
Date : 2 / 3 / 2024

accept **reject**

Figure 5.20 Article Content Interface

5.3.18. Chats:

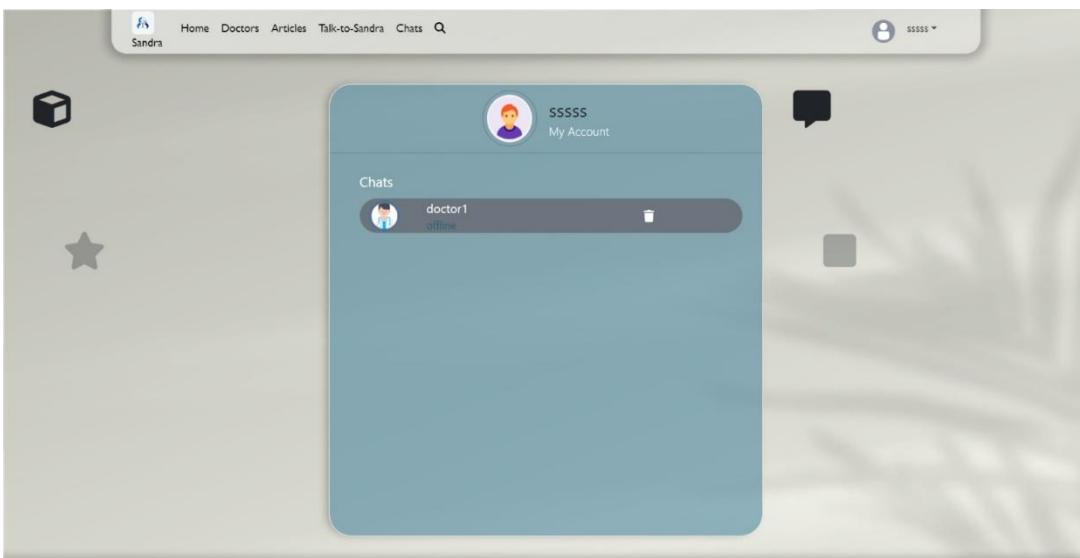


Figure 5.21 Chats Interface

Chapter Six :

(Develop , Testing ,

Deploy)

6.1. Develop:

Our project encompasses several stages that we are diligently working on to deliver a comprehensive mental health support platform. We have initially focused on developing the functionality for doctors, ensuring they have the necessary tools to provide efficient and effective care. This includes features such as patient management, appointment scheduling, and access to medical records.

Following the completion of the doctors' functionality, we moved on to developing the articles functionality. This feature allows doctors 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 doctors.

The chat functionality was another crucial aspect we worked on, providing a dedicated space for real-time communication between doctors and patients. This feature enables secure and confidential conversations, allowing patients to express their concerns and doctors 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 doctors, patients, and administrators alike, fostering effective communication, knowledge sharing, and improved mental well-being.

6.2. Testing and Deploy:

In addition to the extensive development process, we also put great emphasis on deploying and delivering each functionality within our project. Once a specific feature or functionality is deemed stable and reliable, we proceed to deploy it to our live environment. This allows users to immediately access and benefit from the new capabilities and enhancements.

Our deployment process involves careful planning and coordination to ensure a smooth transition. We work closely with our team to ensure that all necessary resources are in place for a successful deployment. This includes preparing the infrastructure, configuring the necessary servers, and ensuring compatibility with the existing system.

Once the deployment is complete, we conduct rigorous testing to verify that the functionality is working as intended in the live environment. This helps us identify and address any issues that may arise during the deployment process.

Once deployed, 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.

Furthermore, 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 doctors and patients, as well as providing a seamless appointment booking process.

By prioritizing the deployment and delivery 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 communication between doctors and patients.

Chapter Seven :

(Conclusion)

7.1. Conclusion:

In conclusion, our project represents a significant advancement in the field of mental health support and communication between doctors and patients. By leveraging technologies such as React, Laravel, and Python, we have successfully developed a robust and user-friendly platform. The integration of features such as article sharing, chat services, and intelligent bot functionalities has greatly enhanced the accessibility and effectiveness of mental health care. Our dedication to deploying each functionality in a timely manner ensures that users can benefit from the platform's capabilities as soon as they become available. Overall, our project is poised to make a meaningful impact by fostering improved communication, knowledge sharing, and support within the realm of mental health.

7.2. Future Work:

In the future, we envision expanding our project by incorporating additional functionalities to further enhance the mental health support platform. These potential features include personalized treatment plans tailored to individual users' needs, integration of wearable devices to track and monitor mental health indicators, a comprehensive resource library with access to books, podcasts, and videos on mental well-being, and the ability to schedule virtual group therapy sessions for enhanced peer support. Additionally, we aim to introduce data analytics capabilities to provide insights and trends on mental health patterns, and integrate telemedicine capabilities for remote consultations with specialists. By continuously exploring and implementing these advancements, we strive to create a comprehensive and evolving platform that revolutionizes mental health support and empowers individuals on their journey to well-being.

References

- adopting-agile-methodology-steps.* (n.d.). Retrieved from plutora: <https://www.plutora.com/blog/adopting-agile-methodology-steps>
- Association, A. P. (2013). *DIAGNOSTIC AND STATISTICAL MANUAL OF MENTAL DISORDERS FIFTH EDITION*. United States of America: American Psychiatric Association.
- deployment.* (n.d.). Retrieved from Laravel: <https://laravel.com/docs/10.x/deployment>
- get_started.* (n.d.). Retrieved from python.langchain:
https://python.langchain.com/docs/expression_language/get_started
- langchain-chat-with-your-data.* (n.d.). Retrieved from deeplearning: <https://www.deeplearning.ai/short-courses/langchain-chat-with-your-data/>