**CHATBOT FOR MENTAL HEALTH SUPPORT**

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfilment of the requirements to award the degree of

**Bachelor of Technology**

In

**Computer Science and Engineering**

**School of Engineering and Sciences**

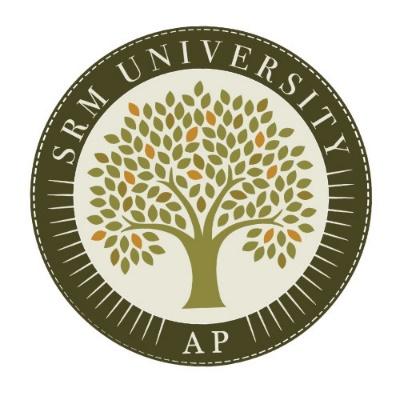
Submitted by

**Attar Imran Basha AP21110011384**

**Gorla Murali Krishna AP21110011388**

**Tatiparthi Charitha Sree AP21110011389**

**Mahanthi Rahul AP21110011393**



Under the Guidance of

**Dr. Sanjay Kumar**

**SRM University–AP**

**Neerukonda, Mangalagiri, Guntur**

**Andhra Pradesh – 522 240**

**April, 2024**

**Certificate**

Date: 30-04-2024

This is to certify that the work present in this Project entitled **“Chatbot For Mental Health Support”** has been carried out by **Attar Imran Basha, Gorla Murali krishna, Tatiparthi Charitha Sree, Mahanthi Rahul** under my supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor Technology/Master of Technology in **School of Engineering and Sciences.**

**Supervisor**

(Signature)

Prof. / Dr. Sanjay Kumar

Designation,

Affiliation.

**Acknowledgment**

I would like to express my special thanks to our mentor Dr. Sanjay Kumar for his time and efforts he provided throughout the year. Your useful advice and suggestions were helpful to me during the project’s completion. In this aspect, I am eternally grateful to you.

We would like to take this opportunity to express my gratitude to all of our group members. The project would not have been successful without their cooperation and inputs.

Signature

A Imran

G Murali

T charitha

M Rahul

**Abstract**

Health care is very important to start a good life. In recent years, the demand for mental health services has increased exponentially. The need for a better health care facility is highlighted more than ever. Besides physical health, mental health conditions have become a significant concern. Unfortunately, there are few opportunities for people to receive mental health care. There are inadequate facilities for seeking mental health support in remote areas. With this goal in mind, we want to provide mental health support to people remotely from anywhere worldwide. This project aims to develop a virtual CHAT BOT. chatbot therapist is designed to provide relief to people suffering from mental illness (anxiety, depression or stress) with daily conversations and motivating them for better mental health.Key functionalities include personalized conversation flows, real-time crisis intervention protocols, and integration with professional support services. Initial evaluations suggest promising user engagement and satisfaction levels, indicating the chatbot's potential to fill critical gaps in mental health support infrastructure. Future iterations will focus on refining the chatbot's algorithms, expanding language capabilities, and fostering partnerships with mental health organizations to ensure ongoing efficacy and relevance in addressing diverse user needs.

**Table of Contents**

**S.No Section Pg No**

**1. Introduction** 3

1.1 Purpose 3

1.2 Scope 3

1.3 Definitions, Acronyms, and Abbreviations 3

1.4 References 3

1.5 Overview 3

**2. The Overall Description** 4

2.1 Product Perspective 4

2.2 Product Functions 4

2.3 User Characteristics 5

2.4 Constraints 6

**3. External Interface Requirements** 6

3.1 User Interfaces 6

3.2 Hardware Interfaces 7

3.3 Software Interfaces 7

3.4 Communications Interfaces 8

**4. System Features** 9

**5. Other Non-Functional Requirements** 9

5.1 Performance Requirements 9

5.1.1 Capacity 10

5.1.2 Dynamic Requirements 10

5.1.3 Quality 10

5.2 Software System Attributes 10

5.2.1 Reliability 10

5.2.2 Availability 10

5.2.3 Security 10

5.2.4 Maintainability 10

5.3 Business Rules 11

**6. Other Requirements**  11

Appendix A: Glossary 12

**7. List of Diagrams**

7.1 Level – 0 DFD 13

7.2 Level – 1 DFD 13

7.3 Level – 2 DFD 14

7.4 Data Dictionary 14

7.5 Use Case Diagram 15

7.6 Entity Relationship Diagram 16

7.7 Class Diagram 17

7.8 Activity Diagram 18

7.9 Collaboration Diagram 19

7.10 State Diagram 20

**8. Testing** 21

8.1 Functional Testing 21

8.2 Path Testing 22

**9. Technology stack** 23

9.1 Front End 24

9.2 Back End 25

**10. Functionalities** 26

**11. Conclusion** 27

**12. Limitations** 28

**13. Future Work** 29

**14. References** 30

**List Of Figures**

1. Level 0 DFD
2. Level 1 DFD
3. Level 2 DFD
4. Use Case Diagram
5. ER Diagram
6. Class Diagram
7. Activity Diagram
8. State Diagram
9. Collaboration Diagram

**List of Tables**

1. Definitions, Acronyms and Abbreviations.
2. Data Dictionary.
3. **Introduction**

The discourse surrounding mental health has gained significant traction, shedding light on the importance of proactive support and destigmatizing conversations around psychological well-being. As the prevalence of mental health challenges continues to rise globally, innovative solutions leveraging technology have emerged to bridge the gap in accessible support services. Among these solutions, chatbots have garnered attention as a promising avenue for providing immediate, confidential, and personalized mental health support.

**1.1 Purpose**

The purpose of this document is to outline the development and functionality of a chatbot designed specifically to provide mental health support. It will elucidate the rationale behind creating such a tool, its intended use, and the benefits it offers to users and stakeholders alike.

* 1. **Scope**

The product scope encompasses the features, functionalities, and limitations of the mental health support chatbot. It includes:

* Providing a platform for users to express their emotions and seek guidance in a confidential environment.
* Offering personalized responses and recommendations based on user input and sentiment analysis.
* Implementing crisis intervention protocols to address urgent mental health concerns.
* Integrating with external resources and professional services for further assistance when needed.
  1. **Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| **SRS** | Software Requirement Specification |
| **API** | Application Programmable Interface |
| **UI** | User Interface |
| **UX** | User Experience |
| **DB** | Data Base |
| **HTTPS** | Hypertext Transfer Protocol Secure |

Table 1: Definitions, Acronyms, and Abbreviations

1. **The Overall Description**

**2.1 Product Perspective**

* Centralized platform in the Mental health support.
* Connects users, Admins, and Mental Health facilities.
* Offers:
  + User Registration
  + User Login
* Aims to:
  + Optimize resource allocation
  + Enhance accessibility to Mental Health
  + Save lives through digital innovation and collaboration within the healthcare community.

**2.2 Product Functions**

**Login Features:**

* User Authentication: Secure login mechanism for all users (Users ) and administrators to access personalized features. Password encryption and secure authentication protocols ensure the safeguarding of user credentials.
* Role-Based Access: Each user is assigned a specific role (User, Doctor, Administrator) upon registration. Role-based access controls determine the features and functionalities accessible to each user class. Admin accounts have elevated privileges for system management.

**Education Section**

Mental Health educational component is meticulously curated to provide comprehensive resources on Mental Health practices, health guidelines, and related topics. It offers valuable insights and guidance to Users and Doctors, empowering them with knowledge to make informed decisions and participate actively in the Mental Health process.

**User Interfaces:**

* **Doctor Interface:**
  + Doctor can create and manage their profiles.
  + Access educational resources and guidelines for Mental Health.

**Admin Interface:**

* Dashboard for administrators to oversee system activities.
* Manage user accounts, roles, and permissions.
* Track doctor appointments, and manage Mental Health.
* Generate reports on Mental Health, inventory levels, and system performance.

**Search and Matching:**

* Allow Users to search for doctors based on criteria such as Health type, and availability.
* Matching algorithm to connect compatible doctors with Users based on their needs and preferences.

**Health Screening:**

* Implement a health screening questionnaire for doctors to assess their eligibility for Mental Health based on medical history and current health status.
* Ensure compliance with donation guidelines and eligibility criteria.

**2.3 User Characteristics**

Understanding the diverse characteristics and needs of our users is crucial for creating a user-centric platform that caters to their requirements effectively. At Chatbot for Mental Health for we recognize the unique roles and perspectives of each user group involved in the Mental Health process.

* **Novice Doctor/Recipient:** 
  + This user is new to Mental Health Support and may lack experience with the process.
  + They require clear guidance and assistance throughout their interaction with the platform.
  + Chatbot provides simple, intuitive interfaces with explanatory screens and interactive tutorials using visual and audio aids to help users understand the Mental Health effectively.
* **Experienced Doctor/Recipient:**
  + This user is familiar with Chatbot and may have used similar platforms before.
  + They seek efficiency and speed in their interactions with Chatbot.
  + With minimal assistance needed, the platform ensures a streamlined experience, allowing users to navigate through Chatbot swiftly.
* **Administrator:**
  + This user, typically a healthcare professional is responsible for overseeing the operational aspects of Mental Health Support.
  + They are well-versed in Chatbot procedures and have additional privileges within the system.
  + Administrators have access to specialized displays and options, enabling them to manage User databases, system configurations effectively.
  + They have the authority to make changes and restrictions to features as needed, especially during maintenance and troubleshooting activities.

**2.4 Constraints**

* Technological Compatibility: The platform must be compatible with a diverse range of devices and operating systems to accommodate users accessing the system from various devices such as desktop computers, laptops, tablets, and smartphones.
* Ethical Considerations: Chatbot must uphold ethical standards in its operations, ensuring transparency, fairness, and integrity in all interactions with users, doctors, recipients, and healthcare professionals.
* Security Concerns: Chatbot must implement robust security measures to safeguard user data and protect against unauthorized access, data breaches, and cyber threats.
* User Education Needs: Ensuring users have the knowledge and skills to utilize the platform effectively requires investment in user education and training programs.

1. **External Interface Requirements**

**3.1 User Interface Requirements**

Chatbot prioritizes a seamless and user-friendly interface to enhance user engagement and accessibility. The platform's user interface requirements include:

* Intuitive Design: The user interface must be intuitive and easy to navigate, catering to users of varying technical proficiency.
* Responsive Layout: Interfaces should adapt to different screen sizes and devices, ensuring accessibility across desktop computers, tablets, and smartphones.
* Clear Navigation: Navigation menus and buttons should be logically organized and clearly labeled for effortless exploration of platform features.
* Accessible Features: The interface should incorporate accessibility features such as alternative text for images, keyboard navigation support, and screen reader compatibility to ensure inclusivity for users with disabilities.
* Interactive Elements: Interactive elements such as dropdown menus, buttons, and forms should be responsive and user-friendly, providing feedback and guidance to users as they interact with the platform.
* Rich Media Integration: Integration of rich media elements such as images, videos, and interactive maps enhances the visual appeal and engagement of the interface, providing users with immersive and informative content.
* Security Measures: Implementation of security measures such as encrypted connections, and secure authentication processes ensures the protection of user data and transactions, fostering trust and confidence among users.

**3.2 Hardware Interface Requirements**

Chatbot is designed to seamlessly integrate with a variety of hardware devices commonly used in daily life.

* Compatibility: The platform should be compatible with a wide range of hardware devices commonly used like desktop computers, laptops, tablets, and smartphones.
* Connectivity: Reliable internet connectivity is essential for accessing the platform's features and functionalities. Users should be able to connect to the platform using standard internet browsers over wired or wireless networks.
* Accessibility: The platform should be accessible to users with different hardware configurations and specifications, ensuring optimal performance and usability across diverse hardware environments.
  1. **Software Interface Requirements:**

Chatbot is engineered to seamlessly integrate with a diverse array of software environments, ensuring compatibility, interoperability, and optimal performance across various platforms.

* Operating System Compatibility: The platform should be compatible with popular operating systems such as Windows, macOS, Linux, Android, and iOS, ensuring accessibility across a wide range of devices.
* Web Browser Compatibility: Support for multiple web browsers including Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge is essential to accommodate user preferences and ensure consistent performance and functionality.
* Database Integration: Seamless integration with databases such as SQLlite is necessary for efficient data storage, retrieval, and management, supporting the platform's various functionalities and operations.
  1. **Communication Interface Requirements**

Chatbot prioritizes seamless communication between doctors, Users, and admins to facilitate efficient collaboration and enhance user engagement. The communication interface requirements include:

* Email Integration: Automated email notifications enhance user engagement with updates and promotions.
* Social Media Integration: Users can share activities and purchases, expanding the platform's reach.
* Communication Protocols: Support for communication protocols such as HTTP, and HTTPS is necessary for facilitating real-time communication between the platform and users, ensuring responsiveness and interactivity.

1. **System Features**

**1. User Registration and Login:**

- Admin, Doctor, and Patient can register and login to their respective accounts.

- Approval is required from the admin for Doctor and Patient accounts.

**2. Admin Features:**

- Register, view, approve, reject, and delete doctors.

- Admit, view, approve, reject, and discharge patients.

- Generate and download invoices in PDF format based on treatment details.

- View, book, and approve appointments requested by patients.

3**. Doctor Features:**

- Apply for a job in the hospital (approval required).

- View assigned patient details (symptoms, name, mobile).

- View list of discharged patients.

- View appointments booked by admin.

- Delete appointments after attending them.

4**. Patient Features:**

- Create account for hospital admission (approval required).

- View assigned doctor's details (specialization, mobile, address).

- View booked appointment status (pending/confirmed).

- Book appointments (approval required).

- View/download invoice in PDF format after discharge.

**5. Chatbot for Mental Health:**

- Implement a chatbot using Botpress to provide mental health support.

- Chatbot should be accessible to patients for assistance and guidance.

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

**5.1.1 Capacity**

* The platform should handle a significant user load efficiently.
* Simultaneous access by a large number of users, including doctors and admins.

**5.1.2 Dynamic requirements**

* Dynamic adaptation to varying workloads and user interactions.
* Maintaining optimal performance during peak usage times.
* Dynamically fetch and display current weather conditions and forecasts based on the user's location or the location of their farm.
* The search and filtering functionalities should be dynamic, allowing users to refine their searches based on changing criteria, such as Feelings and Emotions.

**5.1.3 Quality**

The primary objective of the Chatbot for Mental Health Support project is to develop a robust and user-centric platform that effectively addresses the needs of doctors,Users and administrators.

* **Consistency:** Adhere to established coding standards to maintain a consistent and uniform coding style. Consistency is a key aspect that promotes readability and ease of maintenance.
* **Thorough Testing:** Testing should encompass various scenarios to identify and rectify potential issues and to verify and validate all functionalities within the Chatbot platform.
* **Scalability:** The software should be scalable, accommodating potential increases in user base and data volume.
  1. **Software System Attributes**

**5.2.1 Reliability**

* Chatbot aims to minimize downtime to ensure continuous user access to the platform.
* Robust recovery mechanism in case of system failures.

**5.2.2 Availability**

* High availability, allowing users to access the platform 24/7.
* Planned maintenance windows communicated in advance.
* Conduct regular maintenance during non-peak hours to minimize disruption to users.

**5.2.3 Security**

* Encryption protocols, secure user authentication, and continuous monitoring for potential threats.
* Stay informed about the latest security threats and proactively apply patches and updates to mitigate vulnerabilities.

**5.2.4 Maintainability**

* Design Chatbot for ease of maintenance and updates, facilitating efficient execution of regular maintenance tasks, software upgrades, and system enhancements.
* Provide comprehensive and well-organized documentation to aid developers in understanding the system architecture and codebase.
* Utilize version control systems to track changes, enabling easy rollback in case of errors.
* Establishment of clear coding standards and guidelines to promote a consistent and maintainable codebase.

**6. Other Requirements**

**Appendix A: Glossary**

* + **HTTPS (Hypertext Transfer Protocol Secure) -** A secure version of the HTTP protocol used for secure communication over a computer network, providing encryption and authentication
  + **WebSocket** - A communication protocol that provides full-duplex communication channels over a single TCP connection, allowing interactive communication between a web browser and a web server.

**7. List of Diagrams**

**7.1 Level-0 DFD**



Fig 1: Level-0 DFD

 **Entities:**

* User: Initiates a chat session with the chatbot.
* Chatbot: The mental health chatbot application.
* Doctor Appointment Application: A system that facilitates scheduling appointments with a doctor.
* Admin: Manages the chatbot and maintains the system.

 **Data Flows:**

* Query: The user submits a query to the chatbot.
* Get Results: The chatbot retrieves or generates results in response to the user’s query. This may involve data from the chatbot’s knowledge base or interaction with external systems.
* Update: The chatbot may update the user’s profile or the knowledge base based on the interaction.

 **Processes:** The level 0 DFD doesn’t illustrate the specific processes that the chatbot performs.

**7.2 Level-1 DFD**

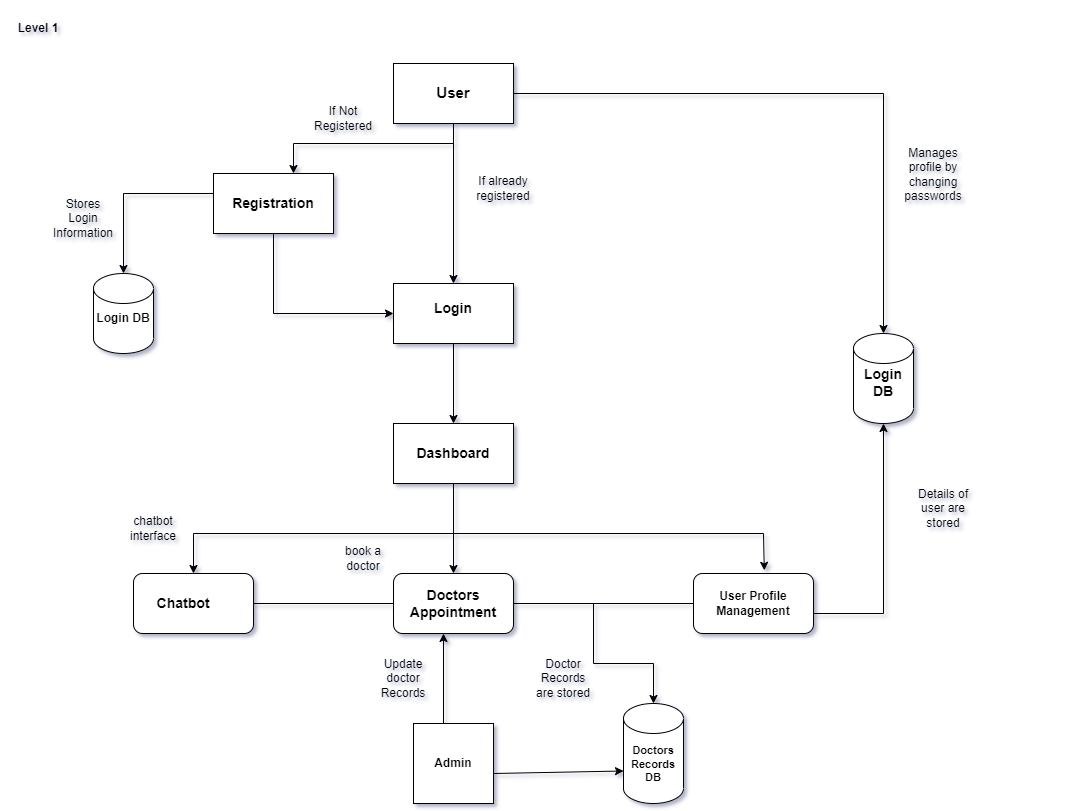


Fig 2: Level-1 DFD

**Processes:**

* **Registration** (process): This process manages user registration. It takes user input for details and stores them in the Login Database.
* **Login** (process): This process validates the user’s credentials upon login. It retrieves data from the Login Database and verifies it against the user-provided credentials.
* **Manage User Profile** (process): This process allows users to change their passwords. It presumably updates the Login Database with the new password.
* **Chatbot Interface** (process): This process presumably manages the conversation between the user and the chatbot. It likely involves processing the user’s queries, retrieving relevant responses from the knowledge base, and generating responses for the user.
* **Doctor Appointment** (process): This process facilitates scheduling appointments with a doctor, likely through interaction with an external doctor appointment system.
* **Update Doctor Records** (process): This process updates doctor records based on unspecified inputs. It stores the updated data in the Doctor Database.
* **Admin** (process): This process presumably allows system administrators to perform various tasks. The specific tasks aren’t detailed in the DFD but could include managing user accounts, managing the knowledge base, or system configuration.

**Data Flows:**

* **Login Credentials** (data flow): This data flow carries the user’s login ID and password for validation. It goes from the User to the Login process.
* **Registration Details** (data flow): This data flow carries user data for registration. It goes from the User to the Registration process and likely includes items like username, password, and potentially other user profile information.
* **Updated Password** (data flow): This data flow likely carries the new password when a user updates their profile. It goes from the Manage User Profile process to the Login Database.
* **User Query** (data flow): This data flow likely carries the user’s chat message or query. It goes from the User to the Chatbot Interface process.
* **Chatbot Response** (data flow): This data flow likely carries the chatbot’s response to the user’s query. It goes from the Chatbot Interface process to the User.
* **Appointment Details** (data flow): This data flow likely carries details about the appointment to be scheduled. It goes from an unspecified source (possibly the Chatbot Interface) to the Doctor Appointment process. Presumably, the doctor appointment system would return a confirmation or other response, but that’s not shown in this DFD.
* **Updated Doctor Records** (data flow): This data flow likely carries data used to update doctor records. It goes from an unspecified source (possibly the Admin process) to the Update Doctor Records process, and then to the Doctor Database.
* **Admin Actions** (data flow): This data flow represents various actions that the Admin can perform on the system. The specific details are not shown in the DFD.

**Data Stores:**

* **Login Database** (data store): This database stores user login credentials.
* **Doctor Database** (data store): This database stores doctor records.

**External Entities:**

* **User** (external entity): This is the person interacting with the mental health chatbot.
* **Doctor Appointment System** (external entity): A system that facilitates scheduling appointments with a doctor.

**7.3 Level-2 DFD**

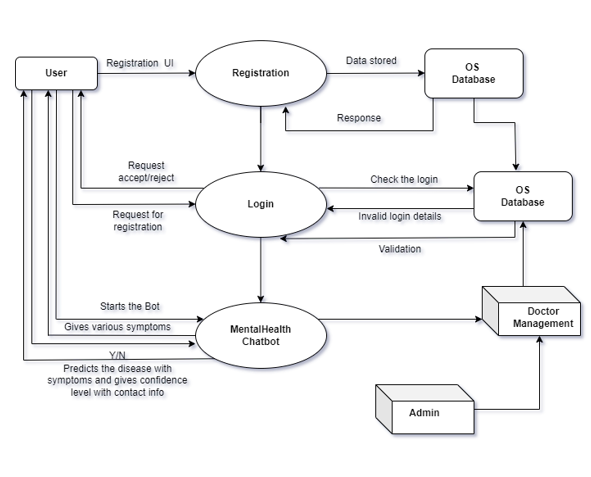


Fig 3: Level-2 DFD

**Data Flows:**

* **User Query** (data flow): This data flow likely carries the user’s chat message or query. It originates from the User.
* **Process User Query** (data flow): This data flow represents the result of processing the user’s query. It likely involves breaking the query down into keywords or concepts to determine the user’s intent.
* **Mental Health Knowledge Base** (data flow): This data flow retrieves information from the knowledge base in response to the processed user query.
* **Symptoms** (data flow): This data flow carries a list of user-reported symptoms (if any) extracted from the user query. It’s used to predict the possible condition.
* **Predicted Condition** (data flow): This data flow carries the mental health condition the chatbot predicts the user may be experiencing, based on the symptoms and the knowledge base.
* **Confidence Level** (data flow): This data flow carries the chatbot’s confidence level in its prediction of the user’s condition.
* **Chatbot Response** (data flow): This data flow carries the chatbot’s text response to the user, which may include information, advice or resources tailored to the user’s query and the predicted condition.
* **Contact Information** (data flow): This data flow likely carries contact information for mental health professionals or resources, which may be included in the chatbot response.

**Processes:**

* **Process User Query** (process): This process breaks down the user’s query to determine the user’s intent.
* **Access Mental Health Knowledge Base** (process): This process retrieves information from the knowledge base in response to the processed user query.
* **Predict Condition** (process): This process analyzes user-reported symptoms (if any) to predict a possible mental health condition.
* **Generate Chatbot Response** (process): This process generates a text response for the user, which may include information, advice or resources tailored to the user’s query and the predicted condition.
* **Provide Contact Information** (process): This process retrieves and provides contact information for mental health professionals or resources.

**7.4 Data Dictionary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity | Attribute | Data Type | Description | Notes (Duplicate Removal) |
| User | id (PK) | int (auto) | Unique identifier for each user. | - Ensure emails are unique during entry to prevent duplicates. |
| User | first\_name | string | User's first name. |  |
| User | last\_name | string | User's last name. |  |
| User | username | string | User's chosen username for login. | - Consider case-insensitive username comparison to avoid duplicates based on capitalization. |
| User | Email | string | User's email address for communication and login. | - Enforce email uniqueness during entry to prevent duplicates. |
| User | password | string | User's hashed password for login authentication. |  |
| User Profile | id (PK) | int (auto) | Unique identifier for each user profile. |  |
| User Profile | user\_id (FK) | int | Reference to the user associated with the profile. |  |
| User Profile | phone\_number | string | User's phone number for contact purposes. | - Validate phone number uniqueness for the same user\_id during entry to avoid duplicates. |
| User Profile | notification\_preference | bool | User's preference for receiving notifications (e.g., email, SMS). |  |
| Appointment | id (PK) | int (auto) | Unique identifier for each appointment. |  |
| Appointment | user\_id (FK) | int | Reference to the user who booked the appointment. |  |
| Appointment | professional\_id (FK) | int | Reference to the mental health professional for the appointment. |  |
| Appointment | date\_time | datetime | Date and time of the appointment. |  |
| Appointment | Status | string | Status of the appointment (e.g., booked, cancelled). | - Consider handling duplicate appointments with different statuses (discuss further if needed). |

Table 2 Data Dictionary

**7.5 Use Case Diagram**

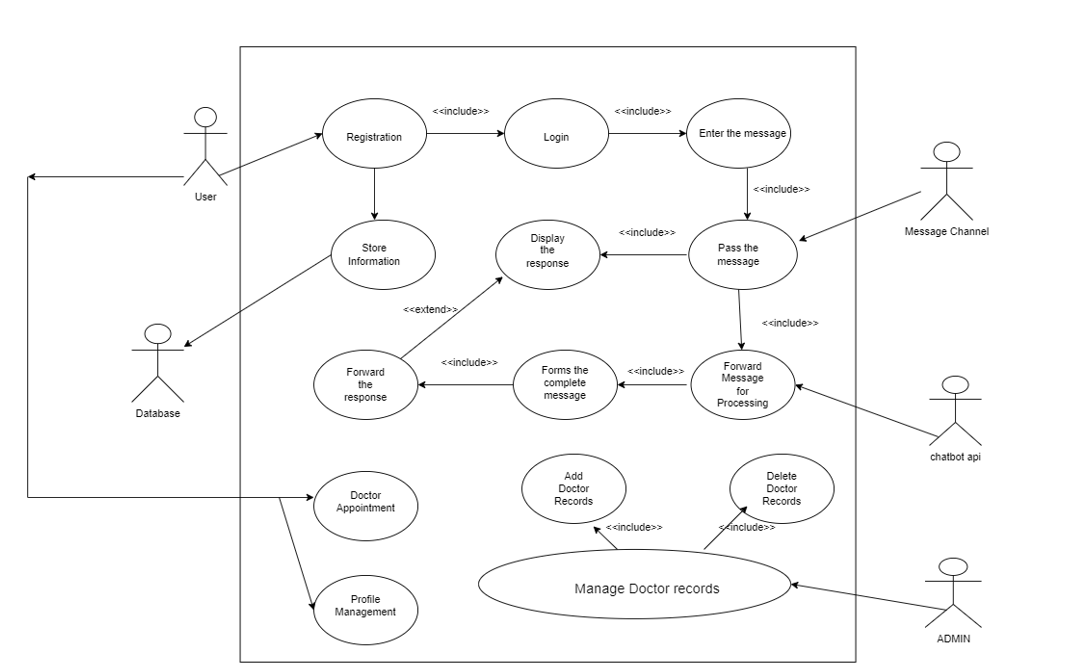
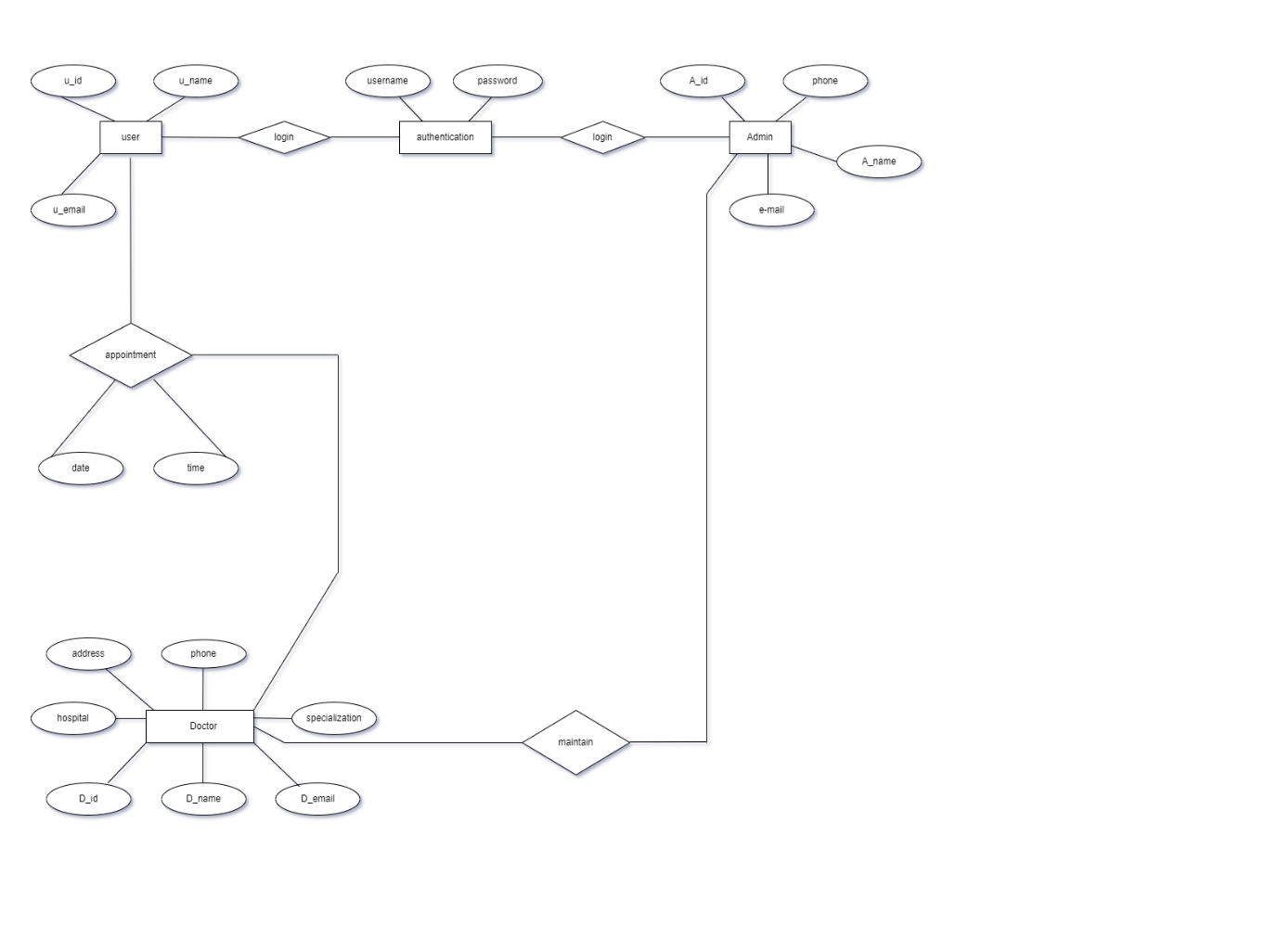


Fig 4: Use Case Diagram

* **Actors:**
  + User: Initiates a chat session with the chatbot and interacts with it to get information or mental health support.
  + Doctor (optional): A doctor may be involved if the chatbot recommends the user consult a doctor.
  + Admin: Manages the system, including the chatbot’s knowledge base and user accounts.
* **Use Cases:**
  + Register: A new user registers with the system to create an account.
  + Login: A registered user logs in to access the chatbot.
  + Chat: The user interacts with the chatbot through text messages or queries. The chatbot responds with information or advice tailored to the user’s needs.
  + Manage User Profile: The user updates their profile information.
  + Doctor Appointment (optional): The chatbot may recommend the user consult a doctor, and potentially help facilitate scheduling an appointment (through interaction with an external doctor appointment system).
  + Manage Doctor Records (admin): The admin adds, updates, or deletes doctor records in the system.

The use cases with solid lines around them (Register, Login, Manage User Profile) are presumably the system’s core functionalities. The use cases with dashed lines (Doctor Appointment, Manage Doctor Records) are optional functionalities.

**7.6 ER Diagram**

Fig 5: ER Diagram

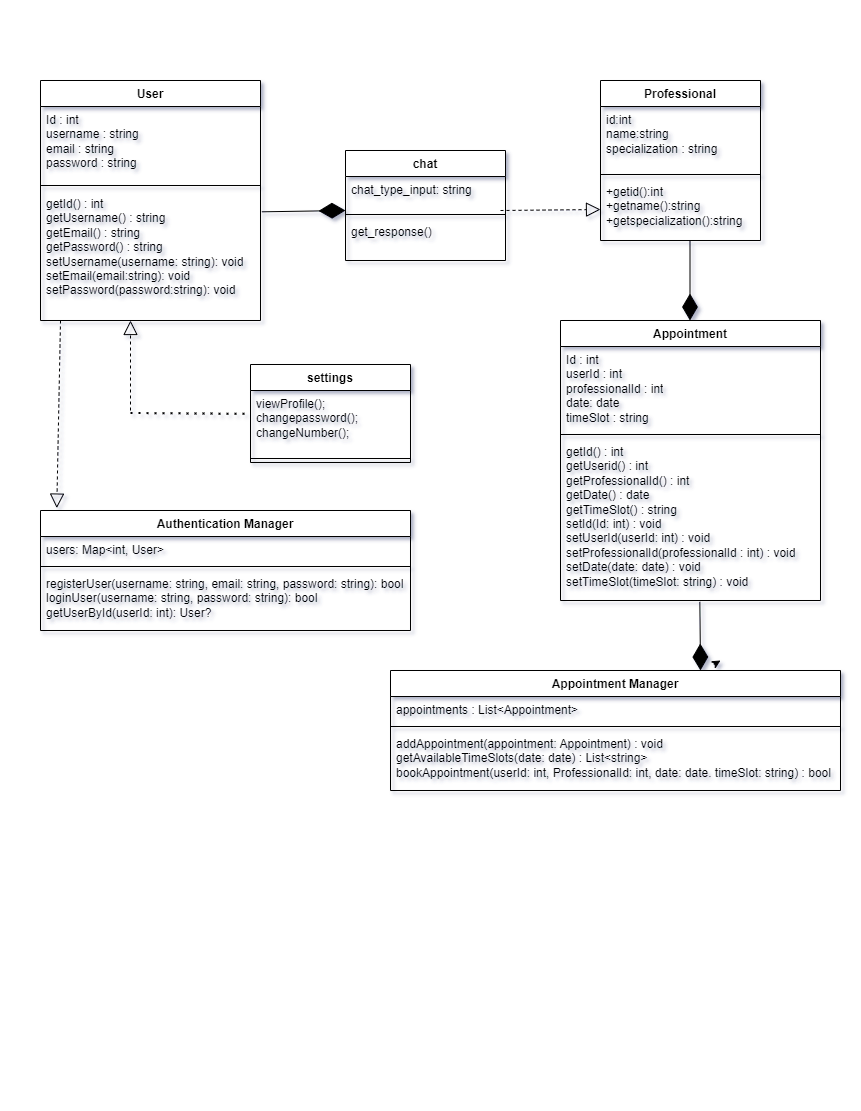
* Conceptual diagrams use broader terms and focus on the high-level relationships between concepts. In the image, we see terms like "Doctor" and "User," but not specific attributes of those entities.
* ERDs use specific terminology to depict entities, attributes, relationships, and cardinalities. An ERD would likely show tables (entities) with columns (attributes) and relationships between the tables. The image doesn't have this level of detail.

However, the conceptual diagram you provided can be a helpful starting point for creating an ERD for your mental health chatbot project. Here are some possible entities and relationships you might consider including in your ERD:

* **Entities:**
  + User
  + Doctor (optional, if the system facilitates doctor appointments)
  + Chat Session
  + Query
  + Response
  + Knowledge Base Entry
* **Relationships:**
  + A User can have many Chat Sessions.
  + A Chat Session can have many Queries.
  + A Chat Session can have many Responses.
  + A Query is مرتبط (related to) one Knowledge Base Entry (or zero, if the chatbot cannot find a relevant entry).
  + A Response is مرتبط (related to) one Knowledge Base Entry (or zero, if the chatbot provides a generic response).
  + A Doctor can have many Chat Sessions (optional, if doctors can review chat history).

**Additional Considerations:**

* You may want to include an entity to store user profile information if your system allows users to create profiles.
* You may want to include an entity to store doctor appointment information if the system facilitates appointment scheduling.

**7.7 Class Diagram**Fig 6: Class Diagram

**7.8 Activity Diagram**

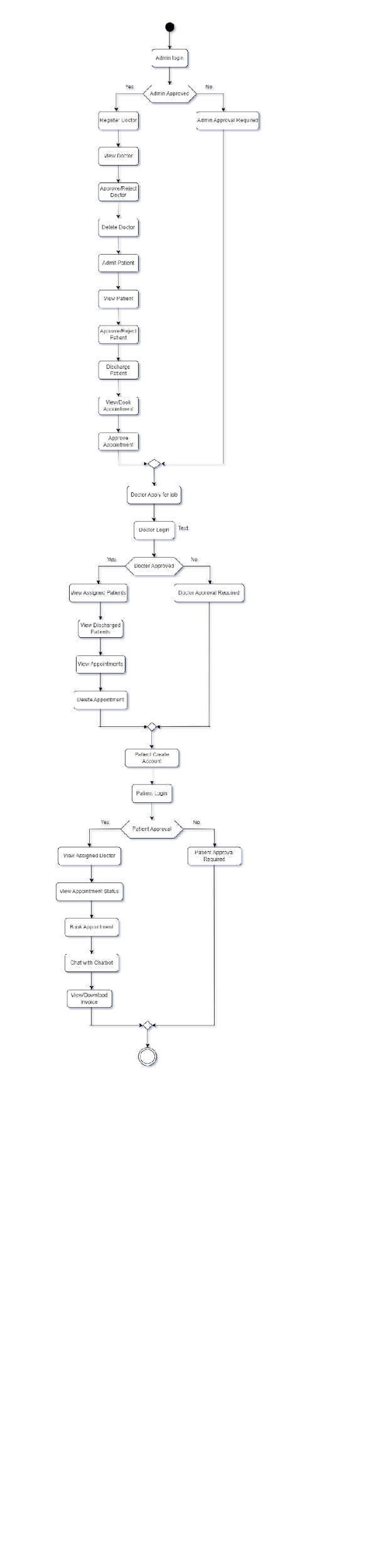
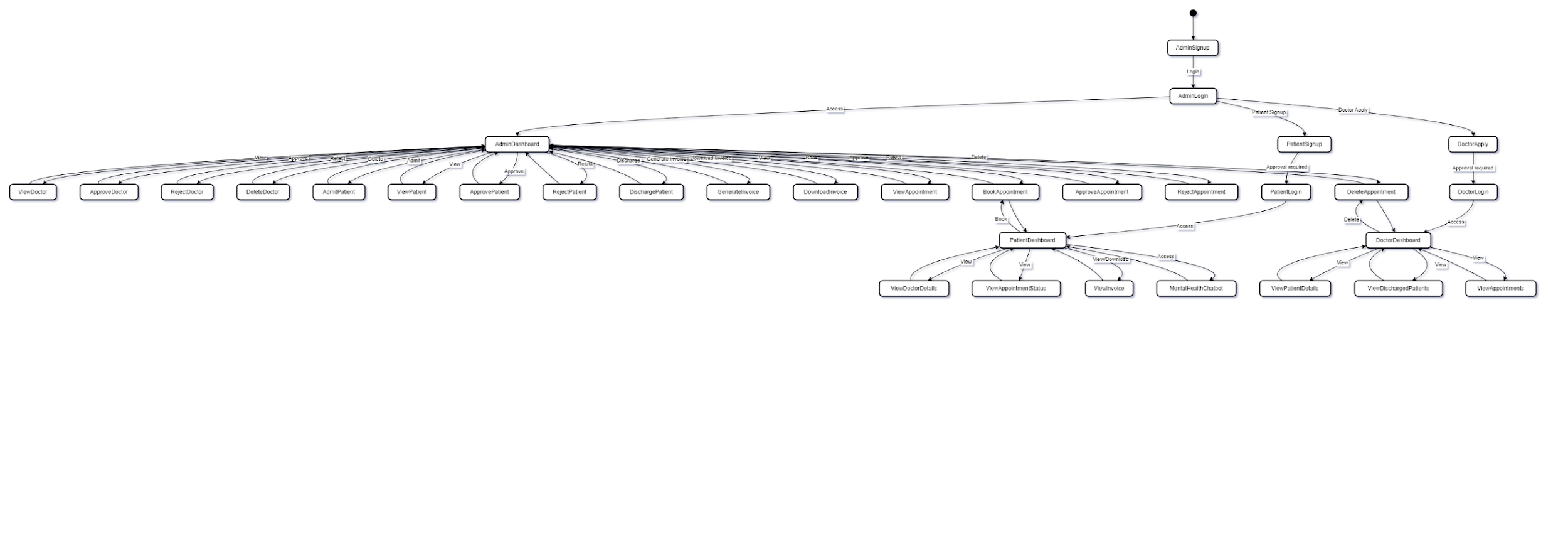
**

Fig 7: Activity Diagram

**7.9 State chart Diagram**

****Fig 8: State Chart Diagram

**7.10 Collaboration Diagram**

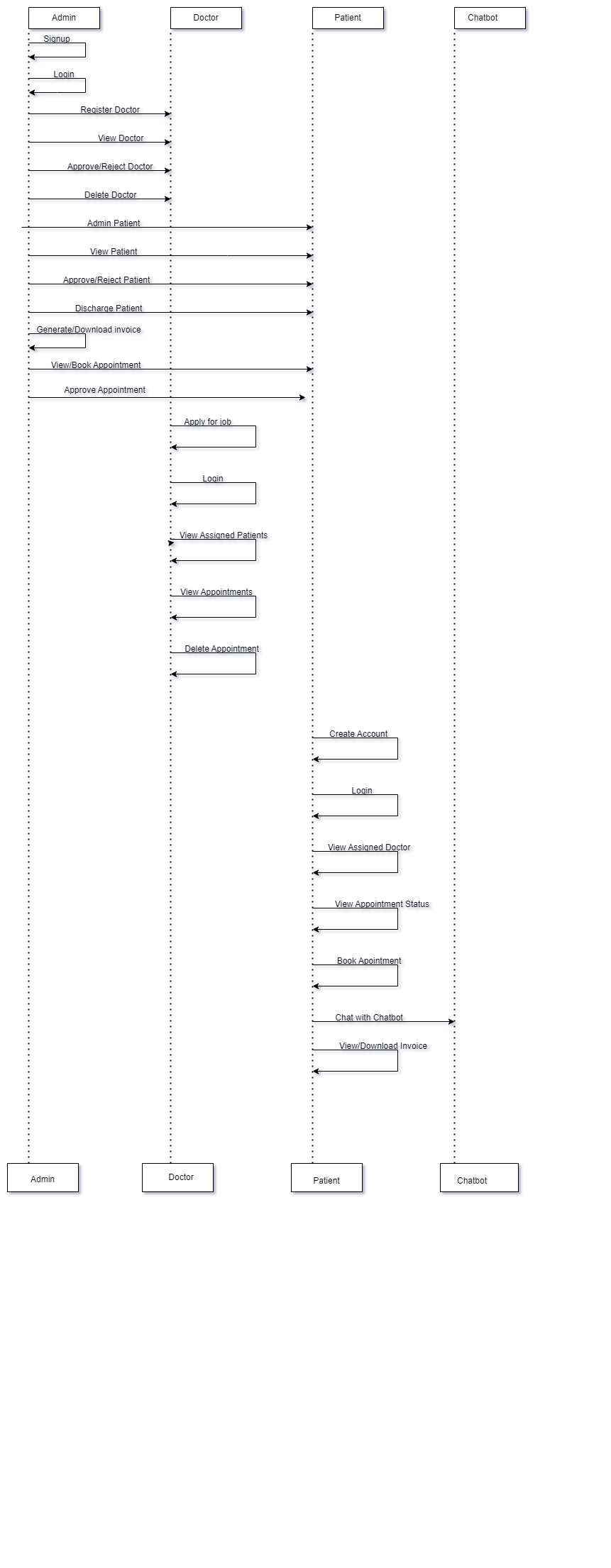


Fig 9: Colaboration Diagram

**8. Testing**

**8.1 Functional Testing**

**Login Module:**

Functional testing of the login modules ensures that users can successfully authenticate and access the Chatbot platform. The following test cases are executed to validate the functionality of the login modules:

* **Valid Credentials Test:**

Verify that users can log in successfully with valid username and password combinations.

* **Invalid Credentials Test:**

Confirm that users receive appropriate error messages when attempting to log in with invalid credentials.

* **Case Sensitivity Test:**

Test whether the system distinguishes between uppercase and lowercase characters in

usernames and passwords during login.

* **Forgot Password Test:**

Test the "Forgot Password" functionality to ensure users can reset their passwords securely.

* **Account Lockout Test:**

Validate that user accounts are locked out after a specified number of failed logins attempts to prevent unauthorized access.

* **Browser Compatibility:**

Test login functionality across different web browsers (e.g., Chrome, Firefox, Safari) to ensure consistency and compatibility.

**Searching Module:**

Test that users can input search queries for Chatbot or recipients based on criteria such as Feelings,availability, and other relevant parameters. Verify that the search functionality is prominently displayed and easily accessible from the user interface.

* **Search Filters:**

Validate that all filter options are functional and produce accurate results when applied individually or in combination.

* **Search Results:**

Confirm that search results are accurate and relevant, displaying matching Doctors or Adminstrators based on the specified criteria.

* **Error Handling:**

Validate that appropriate error messages are displayed when users encounter issues such as invalid search queries, network errors, or database connectivity problems.

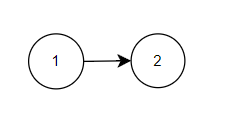
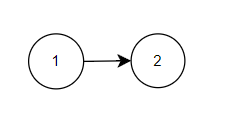
**8.2 Path Testing**

**8.2.1 Signup ( )**

**Pseudo code:**

1. Enter User name, password, email

2. Submit the credentials

**Control Flow Diagram:**

**Complexity**: **E-V+2**

=1-2+2

=1

**8.2.2 Login ( )**

**Pseudo Code:**

1. Enter loginID, Password

2. If (loginID == EnteredUserName && Password==EnteredPassword) then

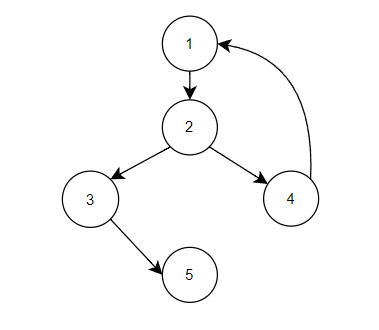
3. print (“Login Successful”)

4.else

Print (“Login Failed”)

Goto 1

5.EndIf

**Control Flow Diagram:**

E: Number of Edges = 5

V: Number of Vertices = 5

**Complexity**: **E-V+2**

=5-5+2

=2

**9. Technology Stack**

**9.1 Front End**

This tech stack offers a powerful combination of tools to develop a robust and user-friendly application like Chatbot for MentalHealth Support.

* **HTML5 and CSS3:**
  + HTML5 provides the structure and semantics for web pages.
  + CSS3 is used for styling and layout, enhancing the visual appeal of the user interface.
* **JavaScript (JS):**
  + JavaScript adds interactivity and dynamic behavior to web pages.
  + It enables client-side scripting for tasks such as form validation, DOM manipulation, and event handling.
* **Bootstrap:**
  + Bootstrap is a front-end framework that provides pre-designed templates and components.
  + It allows for rapid development of responsive and mobile-first web interfaces, ensuring consistency across different devices and screen sizes.
* **jQuery:**
  + jQuery is a fast and lightweight JavaScript library.
  + It simplifies DOM traversal, manipulation, event handling, and AJAX interactions, making front-end development more efficient.

**9.2 Back End**

* DJANGO
  + Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design.
  + It is used for building web applications and provides tools and libraries for common web development tasks.
* SQLITE
  + SQLite is a lightweight, serverless database engine that is often used for development and testing purposes.
  + It is simple to set up and is suitable for small-scale applications or when a full-scale database system is not required.
* Integration
  + Django can be integrated with SQLite or other databases supported by Django's ORM (Object-Relational Mapping) to store and manage application data.
  + The choice of database depends on the application's requirements, scalability needs, and performance considerations.

**10. Future Work**

* Enhanced Chatbot Functionality: Integrate more advanced natural language processing (NLP) features to improve the chatbot's ability to understand and respond to user queries more accurately.
* Telemedicine Integration: Implement features for virtual consultations between doctors and patients, allowing for remote diagnosis and treatment.
* Patient Health Monitoring: Integrate sensors and wearable devices to monitor patient health remotely and provide real-time data to healthcare providers.
* Machine Learning for Diagnosis: Develop machine learning models to assist doctors in diagnosing mental health conditions based on patient symptoms and history.
* Mobile Application: Develop a mobile application to complement the web-based system, providing patients and doctors with easy access to the platform on their smartphones.
* Data Analytics and Insights: Implement data analytics tools to analyze patient data and derive insights that can help in improving treatment outcomes and patient care.
* Enhanced Security Features: Implement advanced security measures to protect patient data and ensure compliance with healthcare regulations such as HIPAA.
* Integration with Electronic Health Records (EHR): Integrate with existing EHR systems to streamline data sharing and ensure continuity of care for patients.

1. **Reference**

<https://ieeexplore.ieee.org/document/10170573>

<https://ieeexplore.ieee.org/document/9456326>

<https://ieeexplore.ieee.org/document/7962482>

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10242473/>