F24-143-D-HealthBridge

Project Team

Muneel Haider	21I-0640
Muhammad Abdullah	21I-0643
Abdullah Zahoor	21I-2481

Session 2021-2025

Supervised by

Mr. Aadil Ur Rehman

Co-Supervised by

Dr. Zeeshan Khan

Department of Computer Science

National University of Computer and Emerging Sciences
Islamabad, Pakistan

June, 2025

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This document outlines the requirements and the use-cases for HealthBridge, a platform that aims to provide AI diagnostics and an appointment booking system. HealthBridge focuses on accessibility by including an appointment booking system with a recommendation system and AI-assisted liver diagnostics into a single platform. The platform is designed to help patients and doctors in the process of the overall healthcare services and diagnosis, all within a single platform. Institutes will also be able to use the AI liver diagnostic model for research purposes.

This document is intended for the following group of readers:

- Developers: They are responsible to build the web and cross-platform applications (mobile application), backend services and the AI diagnostic model.

- Project Managers: They are responsible to oversee the project, including its resources, timeline, and all feature implementation.

- Marketing Staff: They are responsible to position the application in the market, so they should also have a great understanding of the platform's features and capabilities.

- Users: The platform focuses on patients who will use this system for medical consultations in the form of diagnostic report based on scans or booking an appointment with the doctor relevant to their problem. This platform also aims to assist doctors for analyzing scans and identifying patients' problem. Moreover, healthcare institutes will also be allowed to use this platform for research purposes in order to improve the overall accuracy and response of the AI model.

- Testers: They ensure the overall efficiency of the application from the technical aspect of managing appointments successfully for patients and doctors to the accuracy of the AI diagnostics for liver diagnosis.

- Documentation Writers: They are responsible for technical documentation and complete guideline for this project, along with the training resources which cover each aspect of the modules.

1.1 Problem Statement

The healthcare system in Pakistan for both public and private sectors, face various problems which effect the overall quality and accessibility of the services. These challenges create difficulties for both patients and doctors in several ways. Most of the healthcare services in rural areas consist of public hospitals and clinics. These systems provide free medication and treatment for most of the patients due to which many patients prefer such medical services. However, these facilities are mostly inadequate as they provide very limited advanced diagnostics and specialized care generally. Hospitals mainly in urban areas are mostly overcrowded as many patients are forced to wait in long queues which hinders the severity of the problem for patients. Hence, further endangering the health of the patients. Most of the hospitals especially in rural areas do not offer a pre-booking appointment system especially in underdeveloped areas. Alternatively, private hospitals appointment systems are more efficient but cannot be accessed through digital and mobile services. Delays in diagnostic reports have caused several critical issues to the healthcare system whereby almost every hospital in the country would take a number of diagnostic reports and test results for each patient, mostly in the case of specialized procedures like MRI, CT scans and related diagnostics reports. Such reports are usually processed by the diagnostic centers within hours or sometimes even days, which leads to further delays to the whole process of diagnosing a problem and recommending some treatment for the patient. Patients mostly visit doctors' multiple times for consultation and follow-ups, mainly after receiving the test results that are previously recommended by the doctors. This in-creases the wait time and leads to more logistical challenges. Patients follow a navigated healthcare system, moving from doctors to specialists and diagnostic centers without getting a final result or a recommended treatment. Such a system leads to inefficiencies as the patient is unclear about the problem and the doctor remains confused. Such problems highlight the need for improving overall healthcare accessibility and efficiency in Pakistan.

Although apps are available for appointment booking and report generation, they are hosted on different platforms. Patients need to navigate separate apps to utilize these services, making the process more complicated and time-consuming. While AI-assisted diagnostic apps exist, very few focus on liver diseases, an area that requires more attention, especially given the rising prevalence of liver-related health issues.

1.2 Scope

HealthBridge is an all-in-one platform which aims to provide a liver-related medical diagnostics by integrating an AI-Powered healthcare services. The platform allows the users to book an appointment with their relevant doctors, whether for physical or video consultations, ensuring accessibility across all regions, including those with limited facilities. It features diagnostics specifically for liver diseases such Tumor, Cancer, and Fatty Liver (hepatic steatosis). These reports are generated in real-time with basic terminologies that are easy to comprehend. Such AI model also aims to assist healthcare professionals and patients. HealthBridge is meant to be accessible on web and mobile applications to ensure the ease of use for users. This platform aims to reduce the delays which are often associated with traditional diagnostic methods by providing an instant report and treatment recommendations when necessary. Focusing exclusively on liver diagnostics, HealthBridge aims to deliver a centralized and efficient solution, ensuring the quality of healthcare is accessible and reliable for patients in need.

The following is a literature review table, which shows information about research papers that we used to identify more about the algorithms and methods used in research. We intent to practically implement these algorithms in HealthBridge.

Table 1.1: Summary of Multimodal AI Healthcare Research

Reference	**Year**	**Objective**	**Model**	**Key Findings**	**Limitations**
Multimodal for Healthcare	2024	AI diagnostics in medical imaging	RAG (Retrieval Augmented Generation)	Improved diagnostic accuracy	Limited dataset diversity and potential overfitting
Enhanced Diagnostics in Radiology	2022	Multimodal data integration in diagnostics	Multimodal Models	Enhanced diagnostic capabilities	High computational costs and complexity in model training
Multimodal RAG for Medical Radiology	2021	Large Language models in clinical support	GPT 3.5	Relevant medical text generation	Heavy reliance on fine-tuning and risk of inaccuracies
Radiologic Assistant model prediction	2023	Fact-aware generation in radiology reports	FactMM-RAG	Improved factual accuracy in reports	Dependence on retrieval systems, potential outdated information
Medical Vision Language Models	2023	Few short-learning in clinical applications	Med-Flamingo	Effective with minimal data	Limited by initial data quality and struggles with specialized knowledge

1.3 Modules

1.3.1 Module 1: Web Application

Design and implementation of Web Application.

1. Signup Page: Design and implementation of Signup Pages, unique for every type of user.
2. Login Page: Design and implementation of Login page, same for every type of user.
3. Authentication: Implement necessary authentication methods using tools like JWT or OAuth2.
4. Patient Pages: All pages related to searching, booking, communicating and using AI Model.
5. Doctor Pages: All pages related to booking, communicating and using AI Model.
6. Institute Pages: Pages only meant for various institutes, which will only use our AI Model.

1.3.2 Module 2: User Management System (Backend)

Design and implementation of User Management System with connectivity to Web Application and Chatbot.

1. Design Database: Designing our projects database, with ERDs, and relational schemas.
2. Implementation: The designed database will come into existence.
3. Backend APIs: APIs will be made for our project and be used later to connect our webapp with the server.

1.3.3 Module 3: Chatbot

Creation of HealthBridge Assistant Chatbot where users can interact with the Chatbot.

1. Chatbot Page Design: Design a single page for the Chatbot, which will flow like a conversation until a result can be concluded.
2. Connecting Chatbot: Our Chatbot will be fine-tuned and connected to our backend.

1.3.4 Module 4: Diagnostic Model

Pre-processing our collected datasets and development/training of AI Model.

1. Labelling: We will start labelling our datasets and prepare them for segmentation.

2. Model Training: Here we will start to develop our notebooks/neural networks required to train our model.

3. Segmentation: Since we do not have algorithms required for automatic segmentation, we will also need to develop algorithms required for this task before we proceed with Model Training and Data Cleaning.

1.3.5 Module 5: Android Development and System Integration

Start the development of our Android Application, and further improve communication and previous module features.

1. Portal: Algorithms developed for communication between users (patient and doctor) will be improved.

2. Android Application Development: All APIs have been made and tested on the web application, now the development of the android application will start and communicate with our backend.

3. Integration: All our modules which include different modules and their integration into one will be strengthened.

1.4 User Classes and Characteristics

Table 1.2: User Classes and Descriptions

User class	**Description**
Patients	A patient is an individual who wants medical consultancy or assistance regarding their health concerns. Patients seek medical consultancy for various problems across Pakistan, but due to inefficient and lack of resources, most of these patients are unable to get timely medical assistance. These patients seek a reliable platform, capable of providing medical consultation services and diagnostics.
Doctors	Doctors who seek an all-in-one platform which gives them an appointment management system and assisted diagnostic support. A doctor is an individual who is a professional in the medical field, specifically for a targeted organ or disease. Doctors also seek a platform which will assist them with finding more patients easily as well as have an assisted diagnostic model.
Institutes	There are many institutes which conduct medical research in liver. Institutes who intend to use the platform for their own research, as well as help us improve the accuracy of our AI diagnostic model.

Chapter 2

Project Requirements

2.1 Use-case/Event Response Table

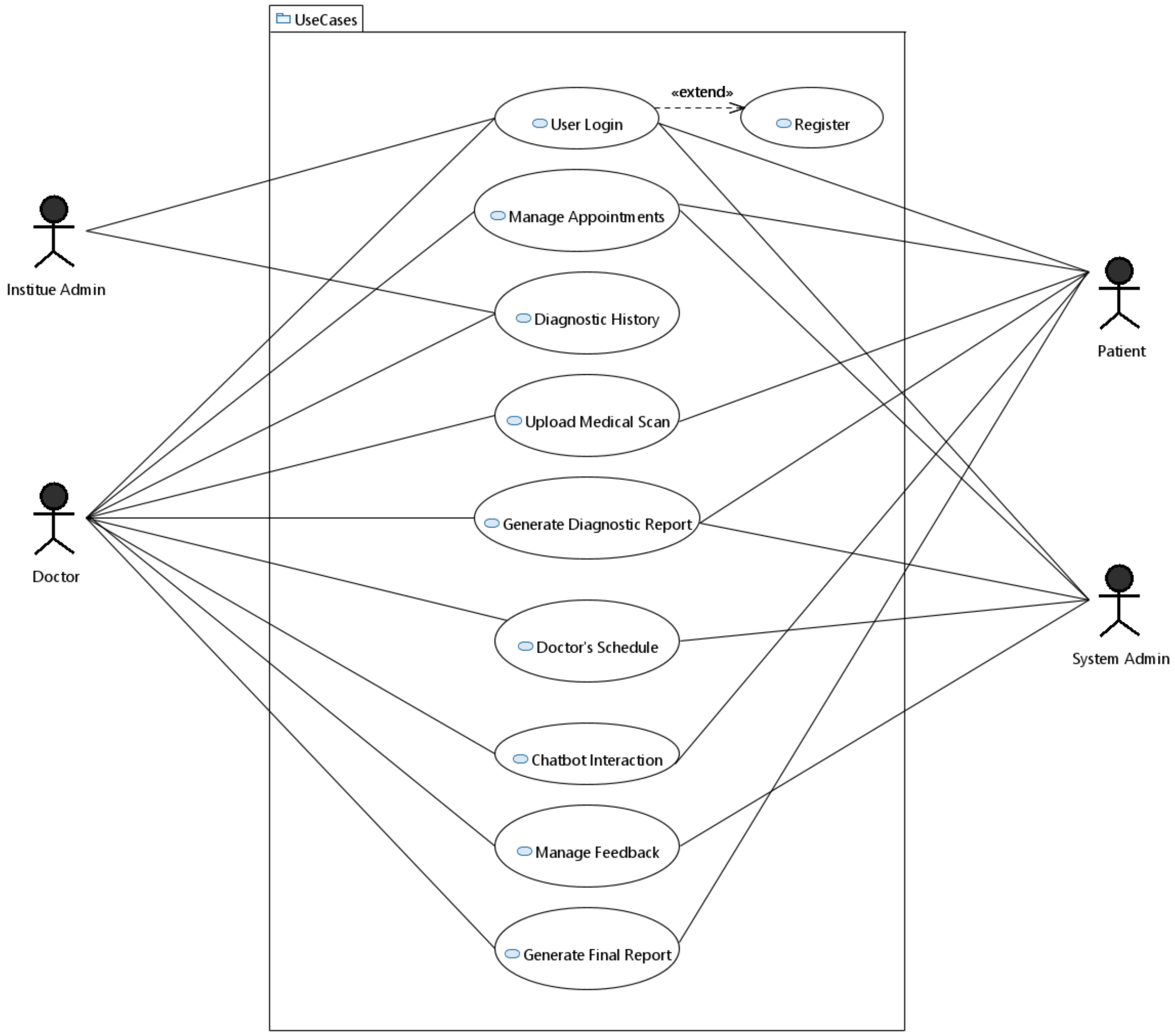


Figure 2.1: Use-case Diagram

2.1.1 Use-case 1: User Registration and Login

- Scope: HealthBridge System
- Level: User goal
- Primary Actor: Patient, Doctor, Institute Admin
- Stakeholders and Interests:

  1. Patient: Secure access to book an appointment and upload scans.

  2. Doctors: Access to schedule appointments and download assistive reports for patients safely.

  3. System Admin: To ensure that the platform is available for authorized users.

- Pre-condition:

  1. Users do not have an account on the platform.

  2. Invalid credentials.

- Post-condition:

  1. Users are successfully registered and logged in and are redirected to their dashboards.

- Main success scenario:

  1 .User navigates to the login/registration page.

  2. User enters credentials (email, CNIC, password).

  3. The system validates the entered credentials.

  4. If the user logs in, they are successfully redirected to their dashboard.

- Extensions:

  1. The system shows an error message if the CNIC/email is already registered.

  2. The system shows an error message if the credentials are invalid.

  3. If the user forgets the password, they can request password reset.

- Special Requirements:

  1. The encrypted password is stored in the database for a secure management system.

- Technology and Data Variations:

  1. Input: CNIC and password

  2. Output: Success or failure message.

2.1.2 Use-case 2: Manage Appointments

- Scope: HealthBridge System (Appointment Scheduling)
- Level: User goal
- Primary Actor: Patient, Doctor, System Admin
- Stakeholders and Interests:

  1. Patient: Authorized users can book, cancel, or reschedule appointments efficiently.

  2. Doctors: They can manage their schedule according to the given slots for patient consultations.

  3. System Admin: Ensures appointment scheduling successfully without any logical errors.
- Pre-condition:

  1. Both patients and doctors successfully log into the system.

  2. Patients can see the available slots for booking an appointment with their relative doctor.
- Post-condition:

  1. The patient successfully books the appointment with their respective doctor and both users are notified.
- Main success scenario:

  1. Patient navigate to “Book Appointment” from their dashboard.

  2. Patient is redirected to the relevant page for booking.

  3. The system displays the list of available doctors with their respective schedule.

  4. The patient is then redirected to the payment method page.

  5. Patient enters the required credentials for secure payment.

  6. System validates the given credentials and proceeds with the overall payment process.

  7. The system notifies the patient for successful payment.

  8. System updates the schedule of the relevant doctor for the appointment after the approval.

  9. Both patient and doctor are notified in case of rescheduling or canceling the appointment.

- Extensions:

  1. The system shows only empty time slots for each doctor to remove ambiguity.

  2. If a patient cancels the appointment, the system updates both patient and doctor and the time slot is updated for the doctor.

- Special Requirements:

  1. Real-time updates

  2. Confirmation notification

- Technology and Data Variations:

  1. Doctor selection with date and time

  2. Appointment confirmation

2.1.3 Use-case 3: Diagnostic History

- Scope: HealthBridge System

- Level: User goal

- Primary Actor: Patient, Doctor, Institute Admin

- Stakeholders and Interests:

  1. Patient: Upload diagnostic reports for review.

  2. Doctors: They are required to review diagnostic histories of patient for accurate consultations.

  3. Institute Admin: For history of previous reports.

- Pre-condition:

  1. All users are successfully logged in.

  2. Doctors and Institute admin can view the uploaded medical history reports of patients.

- Post-condition:

  1. Both doctors and patients can review all the details given in the uploaded medical reports.

- Main success scenario:

  1. User selects the ‘Diagnostic History’ from their dashboard.

  2. The user uploads all the medical history reports.

3. The system ensures the uploaded files can be viewed from both doctors and institute admin.

- Extensions:

1. The system notifies the user in case of no medical history records.

2. Error message in case of incorrect file path of uploaded report.

- Special Requirements:

1. Secure storage of medical history reports.

- Technology and Data Variations:

1. Previous medical reports

2.1.4 Use-case 4: Upload Medical Scans

- Scope: HealthBridge System

- Level: User goal

- Primary Actor: Patient, Doctor

- Stakeholders and Interests:

1. Patient: Upload scans for a diagnostic report.

2. Doctors: Upload scans or analyze the diagnostic report for an accurate final report generation.

3. System Admin: For patient history in case of future appointments.

- Pre-condition:

1. User is successfully logged in.

2. User has the medical scans in the correct format for uploading.

- Post-condition:

1. The AI processes the uploaded scans and provides an accurate diagnostic report for the scans.

- Main success scenario:

1. User selects the ‘Upload Medical Scan’ from the respective page.

2. The user chooses the scan files uploads.

3. The system successfully verifies the uploaded to ensure they are in the correct file format.

4. The AI model processes the scans and generates the automated diagnostic report.

5. Both patient and doctor can view the diagnostic report.

- Extensions:

  1. Error message if file format is not supported.

- Special Requirements:

  1. Secure file upload for diagnostic report.

- Technology and Data Variations:

  1. Medical scan files (dicom, nii)

  2. Various image formats for scan

2.1.5 Use-case 5: Generate Diagnostic Report

- Scope: HealthBridge System

- Level: User goal

- Primary Actor: Patient, Doctor

- Stakeholders and Interests:

  1. Patient: Quick and accurate diagnostics for uploaded scans.

  2. Doctors: Analyzes the diagnostic report of AI model for diagnosis and recommended treatment procedure.

  3. System Admin: Checks the overall accuracy and efficiency of the complete process.

- Pre-condition:

  1. The patient has uploaded the medical scan with the correct file format successfully.

- Post-condition:

  1. The AI model generates a diagnostic report based on the uploaded scans that the user can now download.

- Main success scenario:

  1. The AI model receives the medical scans.

  2. Using pre-trained models for liver diagnosis, the AI model processes the uploaded medical scans.

3. The AI models then generates an accurate diagnostic report after processing.

4. The report is sent to the patient and doctor.

- Extensions:

  1. The system notifies the patient and doctor in case of any troubleshoot or error.

- Special Requirements:

  1. Highly accurate AI model for liver diagnosis.

- Technology and Data Variations:

  1. Scanned medical data

  2. Diagnostic report

2.1.6 Use-case 6: Doctor's Schedule

- Scope: HealthBridge System

- Level: User goal

- Primary Actor: Patient, Doctor

- Stakeholders and Interests:

  1. Doctor: Updates the weekly/monthly schedule for available slots with date and time.

  2. Patient: Real-time availability of slots for booking an appointment.

  3. System Admin: Makes sure that the overall system runs smoothly without any errors and timely update of slots.

- Pre-condition:

  1. The doctor can edit their available slots and schedule of the day.

  2. The patient can view the available slots with time for the relevant doctors.

- Post-condition:

  1. The doctors schedule and availability is updated for the appointment.

- Main success scenario:

  1. The doctor selects 'Manage Schedule' option from the respective page.

  2. The doctor updates their schedule of the day along with time slots.

  3. The doctor saves changes and the system updates the complete schedule.

- Extensions:

  1. If the doctor tries to update an already scheduled appointment after approving the appointment request, the system notifies them and does not update the timeslot for the already scheduled appointment.

  2. In case of any error for updating schedule, the system suggests alternative slots.

- Special Requirements:

  1. Real-time update for appointment scheduling.

- Technology and Data Variations:

  1. Available date and time for the slots.

  2. Schedule update.

2.1.7 Use-case 7: Chatbot Interaction

- Scope: HealthBridge System

- Level: User goal

- Primary Actor: Patient, Doctor

- Stakeholders and Interests:

  1. Patients: Use of Chatbot for getting a generic report for the mentioned problem in order to get recommended doctors or a diagnostic report for medical scans.

  2. Doctors: Can use Chatbot for additional analysis of diagnostic report or report generation.

  3. System Admin: Maintains Chatbot for smooth interaction.

- Pre-condition:

  1. The user is logged in and navigates to Chatbot to start a conversation.

- Post-condition:

  1. The Chatbot assists the user in identifying the problem and recommends relevant doctors on the app.

- Main success scenario:

  1. The patient selects the option for ‘Chatbot’ interaction from the dashboard.

  2. The patient is navigated to the Chatbot page and prompts their relevant query.

3. The Chatbot processes the request and returns the message that is relevant to the patient's query. 4. The patient understands the given response and the task is completed.

- Extensions:

  1. If the system delays or stops working, the Chatbot notifies the patient.

- Special Requirements:

  1. The Chatbot must give a response which is relevant to the user query/question.

  2. The Chatbot must respond in real-time.

- Technology and Data Variations:

  1. User prompt

  2. Chatbot response

2.1.8 Use-case 8: Manage Feedback

- Scope: HealthBridge System

- Level: User goal

- Primary Actor: Patient, Doctor, System Admin

- Stakeholders and Interests:

  1. Patients: They can provide feedback for the app and doctors ratings.

  2. Doctors: Each individual can give their feedback over each appointment on the app.

  3. System Admin: Analyzes the feedback for future improvements.

- Pre-condition:

  1. The users are successfully registered and logged in to the app.

  2. The users can use the app for diagnostic reports and appointments.

- Post-condition:

  1. The users have used the application for diagnostic report and appointments and can give the overall feedback of the system.

- Main success scenario:

  1. The user logs into the app.

  2. The patient selects the option of "Provide Feedback".

3. The patient gives rating for their respective doctor after the appointment.

4. Similarly, the doctor also provides a detailed feedback for the overall system after the appointment.

5. The system saves the feedback after the submission.

- Extensions:

  1. The user is notified if the system fails to save the feedback due to any troubleshooting issues.

- Special Requirements:

  1. The system should be able to save the information in the database for each user.

- Technology and Data Variations:

  1. Ratings and comments.

  2. Confirmation message for submission.

2.1.9 Use-case 9: Generate Final Report

- Scope: HealthBridge System

- Level: User goal

- Primary Actor: Patient, Doctor, AI Model

- Stakeholders and Interests:

  1. Patient: Receives a final report based on doctor's complete analysis assisted by AI generated diagnostic report.

  2. Doctor: Uses the AI based automated report for final diagnosis for recommended treatment or prescription.

  3. AI Model: Provides a diagnostic report to assist doctor for their finalized report.

- Pre-condition:

  1. The patient uploads the medical scans.

  2. The model processes the uploaded scans and provides an initial diagnostic report.

- Post-condition:

  1. The final report is generated by the doctor with the assistance of the initial diagnostic report of scans. The patient can view the report and download it.

- Main success scenario:

  1. The model provides and initial diagnostic report based on the uploaded scans.

  2. The doctor analyzes the diagnostic report based on the scans.

  3. After analyzing the diagnostic report in detail along with the recommended tests, the doctor will then generate a final report based on the patient details along with the description of the diagnosed problem and the relevant steps for the treatment.

  4. The patient will be notified when the final report will be generated. They can proceed to view it or download the finalized report if necessary.

- Extensions:

  1. The diagnostic report generated by the model is incomplete.

  2. The diagnostic report has ambiguities.

  3. If the model based report is not generated completely, the system will restart it.

- Special Requirements:

  1. The final report must consist of all the details of the respective patient, the diagnosed problem according to the scans and tests in detail, and the recommended treatment signed by doctor.

- Technology and Data Variations:

  1. Medical scans.

  2. Doctors detailed analysis of the diagnosed problem.

  3. Final report.

Table 2.1: System Events and Responses

Event	**System Response**
New user registers for the first time	1. User enters all the required details for registration. 2. The system verifies whether the information is correct with all constraints. 3. The system sends an email verification for authentication. 4. The user is successfully registered with the given information.
User logs in with valid credentials	1. The system validates the entered credentials. 2. The system authorizes the user to log in and the user is navigated to their respective dashboard.
User logs in with invalid credentials	1. The system shows an error message for invalid credentials. 2. The user can select the option of password reset if necessary.
User requests for password reset	1. The system sends a password reset link/code on the registered email. 2. The user is required to check their email for updating the password securely and successfully.
Patient books an appointment	1. The patient searches for relevant doctors. 2. The system navigates the patients to the searched doctors page based on the selected option (specialty, rating, region). 3. The patient selects the preferred doctor. 4. The system displays the available timeslots of each day for the specific doctor. 5. The patient selects the timeslot for the appointment with the relevant doctor. 6. The doctor chooses the option for accepting or rejecting the pending appointment. 7. The system saves the information and navigates the patient to the payment page. 8. The patient proceeds to the payment page and enters payment details. 9. The system notifies both users for the confirmation of the appointment.
Doctor accepts/rejects the appointment	1. The system updates the timeslots based on doctor's action in real-time. 2. Both users are notified with the updated status of the appointment.
Patient/Doctor cancels or reschedules appointment	1. The system allows both users to cancel or reschedule the appointment within 24 hours after the confirmation of the appointment. 2. The system updates the schedule and slots of doctor and notifies both end users.
Patient uploads medical history	1. The patients can upload previous medical history reports and tests that the doctor can access after the confirmation of the appointment for future reference.

Table 2.2: System Events and Responses (2)

Patient uploads medical scans	1. The patient selects the option for interacting with the diagnostic model. 2. The patient uploads the relevant liver scans for diagnosis with the correct file format. 3. The system validates the file format of the uploaded scans. 4. The model processes the uploaded scans and generates a diagnostic report. 5. Both users can view the report after the completion of the process.
Diagnostic model generates diagnostic report	1. The diagnostic model processes the information and generates a diagnostic report for the uploaded scans. 2. The user is notified after the completion of the report.
Doctor analyzes diagnostic report	1. The doctor analyzes the report generated by the diagnostic model and either confirms the diagnosis or recommends further tests/scans. 2. The system allows the doctor to add their comments for the diagnosis in the generated report.
Patient views/downloads diagnostic report	1. The patient can view the diagnostic report and the comments added by the doctor after the report analysis. 2. The system ensures that the report is accessible from the patient's dashboard.
User interacts with Chatbot	1. The Chatbot responds to the user inquiries, such as the problem explained, recommending the available doctors, or a diagnostic report. 2. The system processes the request in real-time and provides the appropriate response.
Doctor generates finalized patient report	1. After a detailed analysis of the report generated by the diagnostic model, the doctor finalizes the report which comprises of the complete details of the patient, description of the diagnosed problem, and the recommended treatment plan. 2. The system notifies the patient once the finalized report is completed and uploaded by the doctor. 3. The patient can view and download the final report.
User provides feedback	1. The system allows users to provide feedback after using the application. 2. The patient can provide feedback after the appointment with the doctor. 3. The feedback and rating is stored in the database for analysis and for future improvement. 4. The doctors can provide feedback for the overall provided system.
Doctor updates the schedule	1. The doctor can update their available timeslots for the complete schedule of a week. The doctor clicks on "Save Changes". 2. The system updates the complete schedule of the respective doctor with the timeslots for each day. 3. The system reflects these changes onto the patient's side for the future appointments.
User stays inactive for 15 minutes	1. The system automatically logs out the user if the user stays inactive for more than 15 minutes. 2. A message is displayed to log in again for every user.

2.2 StoryBoarding

The storyboard for the HealthBridge platform outlines the complete process for patients, doctors, and institute admin.

2.2.1 Patient's Workflow

2.2.1.1 Registration and Login

1. User Action: The patient enters the required information for registering as a new user. In case of already registered user, the patient enters valid credentials for login (CNIC and password).
2. System Response: The system validates the entered credentials of the user and navigates them to their respective dashboard which confirms the user login successfully.
3. Visual Elements: A login and register page with the relevant forms and input fields clearly displayed. For specific fields, the required format will also be given as an example for patient's ease.

2.2.1.2 Chatbot Interaction

4. User Action: The patient clicks the Chatbot icon and navigates to the Chatbot page. The patient enters the respective query in a message form and receives the relevant response related to the platform. e.g. identified problem and related available doctors with specialty and ratings.
5. System Response: The Chatbot responds in real-time for the defined query (medical scans, appointment related query, and recommended doctors). The Chatbot will provided the solution to the given query.
6. Visual Elements: A Chatbot icon is displayed on the patient dashboard which navigates the patient to a new page specifically for messages and response of Chatbot.

2.2.1.3 Appointment Booking

7. User Action: The patient sees the available doctors in their respective region. In case of an appointment with a specific doctor, the patient can search by name or specialty. After selecting a doctor, the patient chooses the timeslot of their choice and proceeds to the payment page. If the doctor accepts the appointment, the payment

process will be completed with the payment information given. The appointment will be confirmed and patient will be notified.

8. System Response: The system will provide a real-time update with each step taken by the patient to book an appointment. The system will then proceed with updating the time slots of the respective doctor after the confirmation of the appointment and payment update.

9. Visual Elements: A graphical interface which shows list of doctors with experience, rating, specialty, and rating. The patient can also use the search option for selecting a specific doctor for an appointment, the doctor details after selection and their relevant timeslots of each day, and all the required information for payment. The patient will then get a notification for confirmation.

2.2.1.4 Upload Medical Scans

10. User Action: After the patient clicks the Chatbot icon, the patient will see the option of diagnostic model. The patient will be navigated to diagnostic model page. The patient uploads their scans with the correct file format. The patient then receives the diagnostic report of the uploaded scans. The patient can also upload the scans and the diagnostic report in the portal for doctors' ease.

11. System Response: The system checks the file format of the uploaded scans and notifies the patient if the file format is incorrect. The diagnostic model processes the uploaded scans thoroughly with a loading animation. Once the report is ready, the system will allow the patient to view the uploaded report.

12. Visual Elements: The diagnostic model page will consist of interactive buttons for uploading the medical scans. The loading animation will be shown during the processing. The diagnostic report will be shown on the same page once ready.

2.2.1.5 Consultation and Final Report

13. User Action: The platform will provide a seperate portal for patient and doctor appointment. The patient will be given the option of call in case of online appointment along with relevant tools (in person or online).

14. System Response: The system will show the final report uploaded by the doctor. The patient can successfully view the final generated report and download it.

15. Visual Elements: The platform will display the visually appealing options for consultation (reports and online) along with the final uploaded report.

2.2.1.6 Provide Feedback

16. User Action: The patient selects the feedback option given on the dashboard for their overall system application experience.
17. System Response: The system stores the submitted feedback of the patient.
18. Visual Elements: The feedback page will be interactive, providing a comment box and star ratings.

2.2.2 Doctor's Workflow

2.2.2.1 Registration and Login

1. User Action: The doctor enters the login credentials (CNIC and password) for accessing dashboard. For registration, the doctor enters all the required information in detail for the approval from the system admin.
2. System Response: The system validates the entered credentials and navigates and them to their respective dashboard which confirms the user login. The system displays the detailed dashboard for scheduled appointments and reports along with the timeslots in real-time.
3. Visual Elements: A simple login interface (same as patients) with a detailed registration page for a new page for storing more information with more input fields.

2.2.2.2 Managing Appointments

4. User Action: The doctor accepts/rejects the upcoming appointments according to their availability.
5. System Response: The system updates the timeslots for the doctor in real-time and notifies both users.
6. Visual Elements: The doctors interface visually shows the timeslots of availability and already selected timeslots for the appointments highlighted.

2.2.2.3 Diagnostic Report Analysis

7. User Action: The doctor analyzes the model generated diagnostic report before or while consulting with the patient.

8. System Response: The system allows the doctor to view or download the diagnostic report for comprehensive analysis.

9. Visual Elements: The doctor will be able to see the uploaded scans of the respective patient and the diagnostic report.

2.2.2.4 Generate Final Report

10. User Action: The doctor will generate the finalized report after consultation and analysis of the diagnostic report and upload it.

11. System Response: The system will allow the user to see the uploaded final report.

12. Visual Elements: The final report will be uploaded on the doctor-patient portal.

2.2.2.5 Provide Feedback

13. User Action: The doctor submits the feedback of the overall platform.

14. System Response: The system stores the submitted feedback of the doctor.

15. Visual Elements: The feedback page will be interactive, providing a comment box.

2.2.3 Institute Admin's Workflow

2.2.3.1 Registration and Login

1. User Action: The institute admin enters the login credentials (Email and password) for accessing dashboard. For registration, the admin enters all the required information in detail.

2. System Response: The system validates the entered credentials of the user and navigates them to their respective dashboard which confirms the user login successfully. The admin will be able to see the previous diagnostic reports that were generated using the diagnostic model.

3. Visual Elements: A signup login and register page with the relevant forms and input fields clearly displayed. For specific fields, the required format will also be given as an example for admin's ease.

2.2.3.2 Diagnostic Model for Research

4. User Action: The admin will select the Diagnostic Model icon and will be navigated to the relevant page. The admin will upload available scans with correct file format.

5. System Response: The system checks the file format of the uploaded scans and notifies the patient if the file format is incorrect. The diagnostic model processes the uploaded scans thoroughly with a loading animation. Once the report is ready, the system will allow the patient to view the uploaded report.

6. Visual Elements: The diagnostic model page will consist of interactive buttons for uploading the medical scans. The loading animation will be shown during the processing. The diagnostic report will be shown on the same page once ready.

2.3 Functional Requirements

2.3.1 Module 1: Web Application

1. The user is successfully able to register and login to the platform.

2. The user is properly authenticated before being able to access the platform.

3. The user's respective pages have all use cases implemented.

2.3.2 Module 2: User Management System

1. The database communicates with the web application.

2. APIs cannot be accessed outside of our web application.

2.3.3 Module 3: Chatbot

1. The chatbot responds according to the user's prompt.

2. The chatbot successfully communicates with the user, as well as our backend server.

3. The chatbot accurately suggests recommended doctors.

4. The chatbot can process various formats of the JSON body, since we will later integrate this chatbot with our AI model.

2.3.4 Module 4: Model Development

1. Datasets are correctly labeled.
2. The model returns the result in 2 different formats.
3. It should return a highlighted image and a JSON body that concludes the diagnosis made which can later be presented however we want.

2.3.5 Module 5: Android Development and System Integration

1. The android application can communicate with our backend.
2. The web application and android application are updated at the same time.

2.4 Non-Functional Requirements

2.4.1 Reliability

1. The accuracy of our AI model will be above 80
2. The system will accurately recommend respective doctors according to the problem statement of the patient.

2.4.2 Usability

3. The system servers will be running at all times, ensuring availability of platform.
4. The system will log out any users who are in-active for 15 minutes in a single session automatically, unless they choose to permanently log in.

2.4.3 Performance

5. The system will be built on fast and reliable APIs, ensuring quick responses to users.
6. The system will handle approximately 25 users in 5 seconds.
7. The Chatbot should return a response within 4 seconds for a user query in 90percent of cases.

2.4.4 Security

8. All sensitive data that is saved in the database will be encrypted.

9. The system will incorporate authorization methods, ensuring users are only granted access to features they are authorized to access.

10. The system will use an authentication method, which ensures users that are logged in are not only authorized, but also have valid token sessions through proper authentication.

2.5 Domain Model

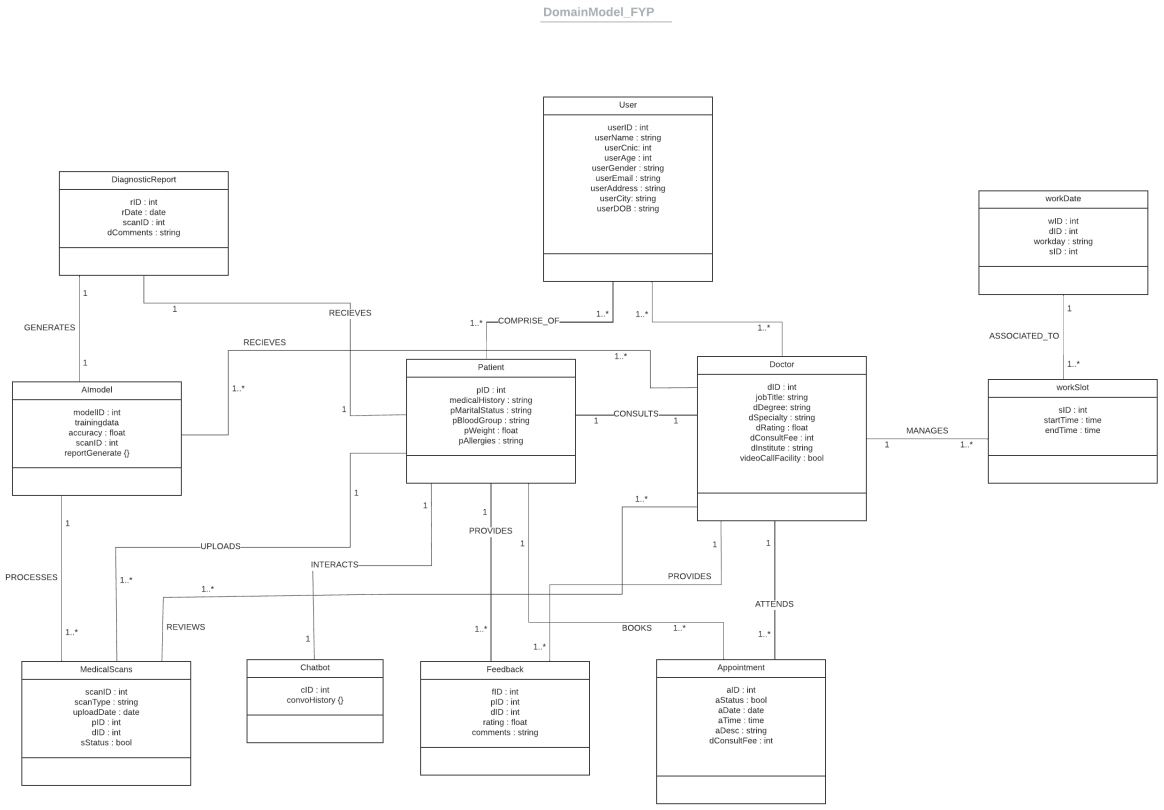


Figure 2.2: Domain Model

Chapter 3

System Overview

3.1 Architectural Design

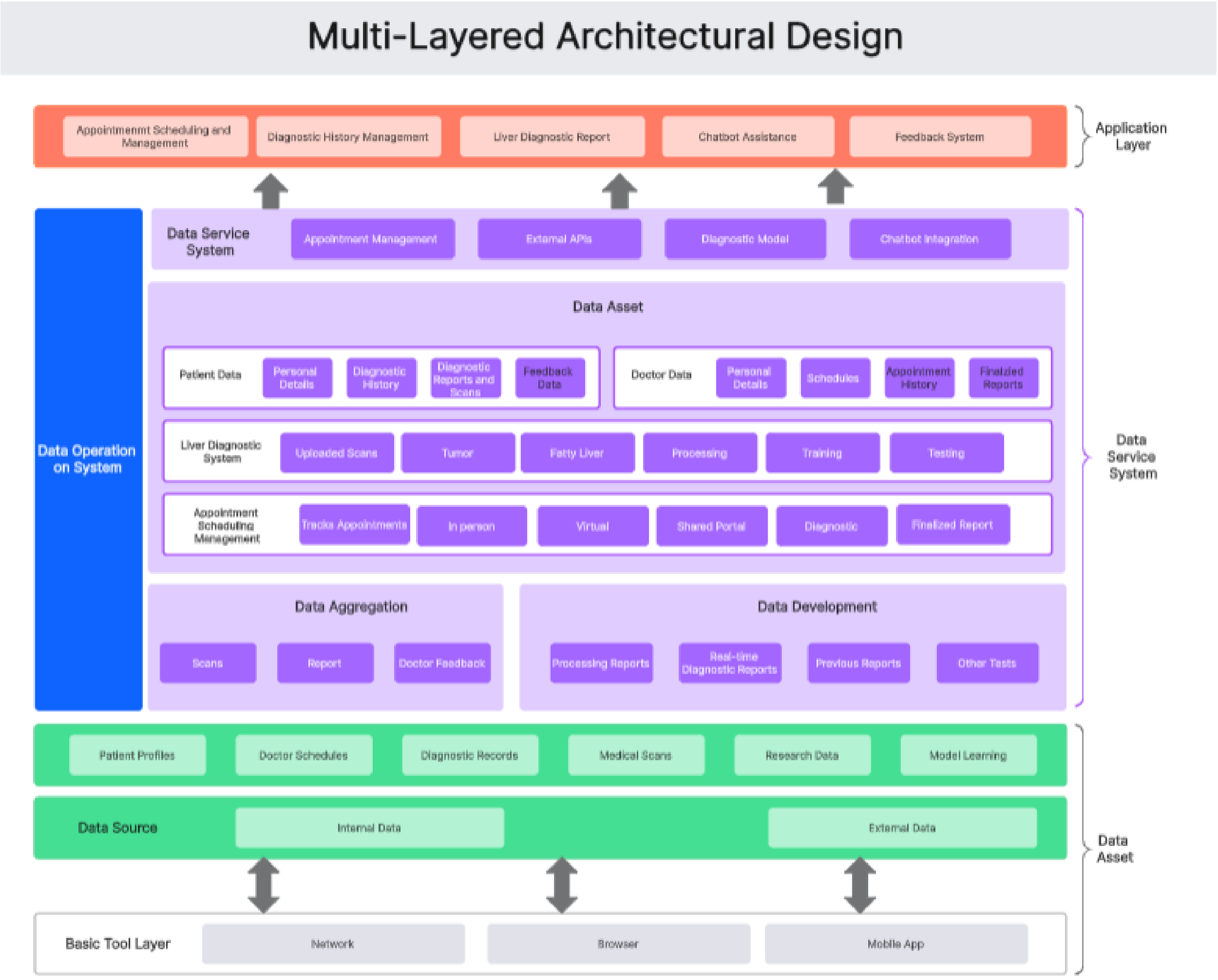


Figure 3.1: Architecture Diagram

3.2 Design Models

3.2.1 Activity Diagram

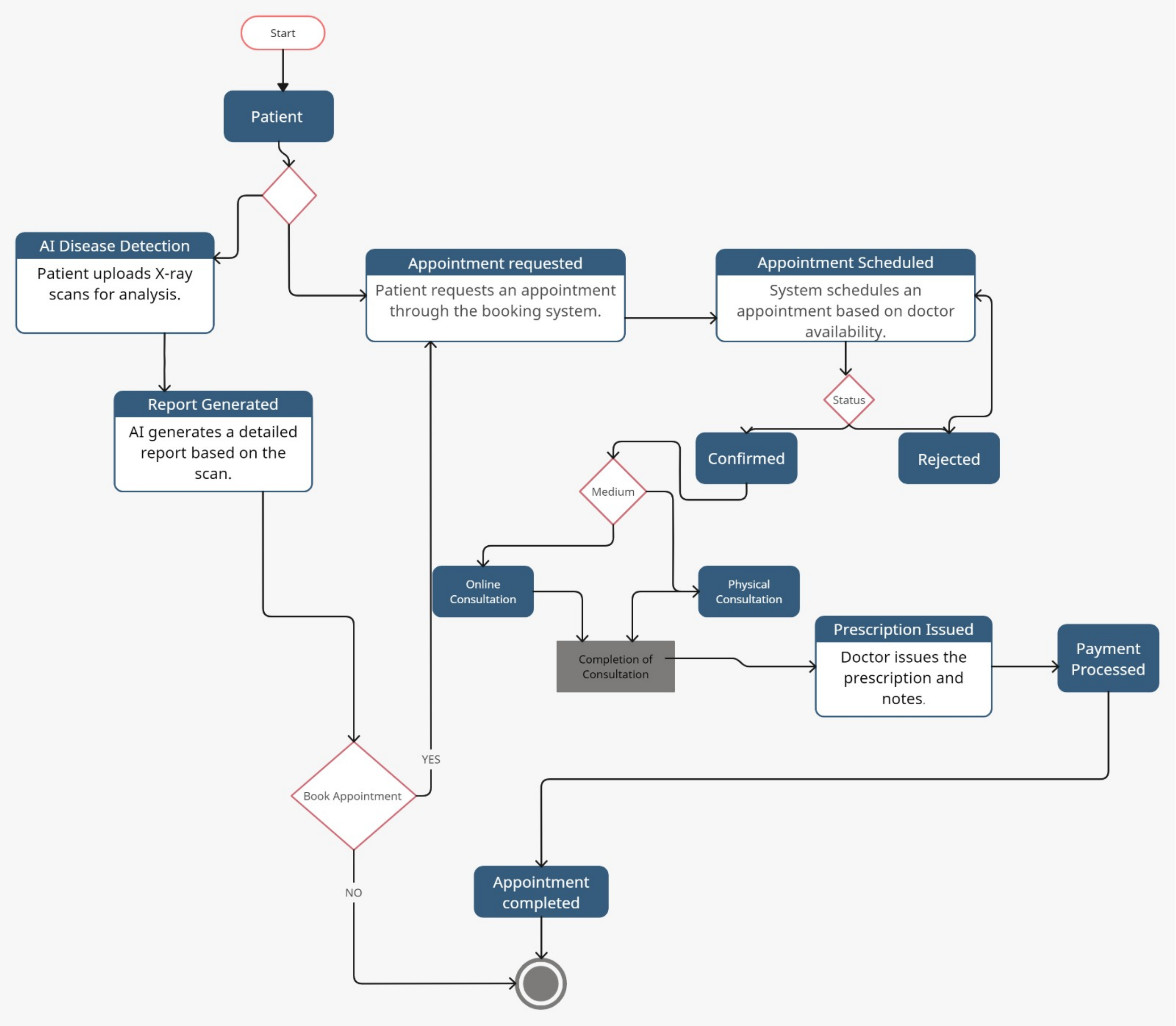

Start
Patient
AI Disease Detection
Patient uploads X-ray scans for analysis.
Appointment requested
Patient requests an appointment through the booking system.
Appointment Scheduled
System schedules an appointment based on doctor availability.
Report Generated
AI generates a detailed report based on the scan.
Status
Confirmed
Rejected
Medium
Online Consultation
Physical Consultation
Completion of Consultation
Prescription Issued
Doctor issues the prescription and notes.
Payment Processed
YES
Book Appointment
NO
Appointment completed

Figure 3.2: Activity Diagram

3.2.2 Data Flow Diagram

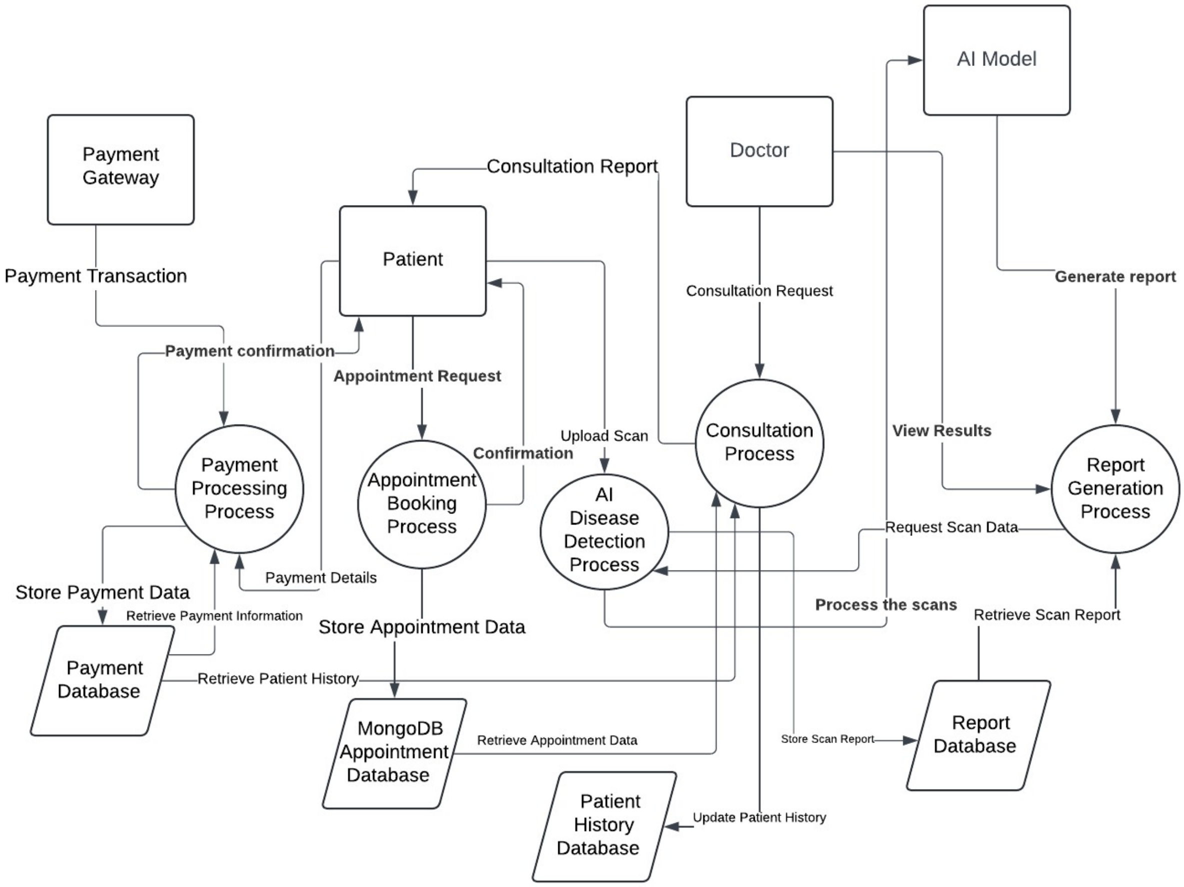


Figure 3.3: Data Flow Diagram

3.2.3 System-level Sequence Diagrams

3.2.3.1 Login

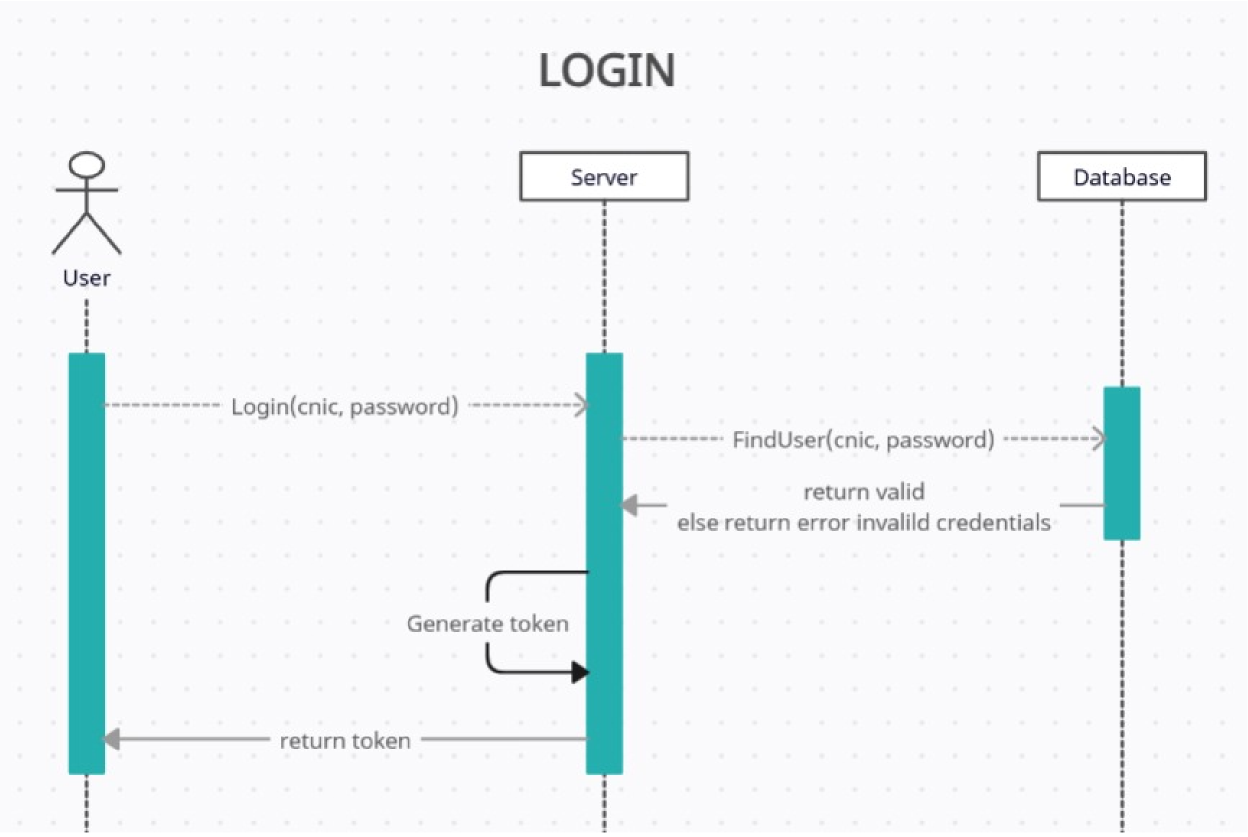


Figure 3.4: Login

3.2.3.2 Registration

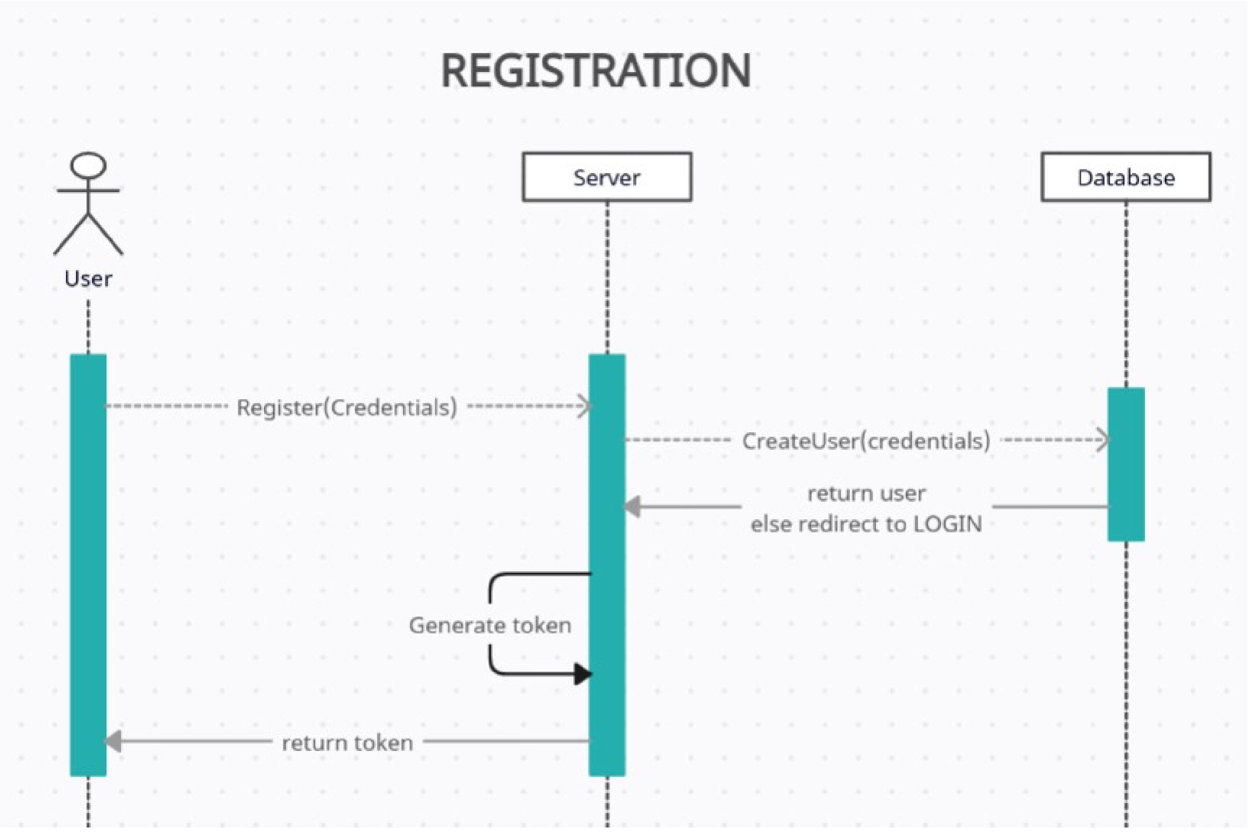


Figure 3.5: Registration

3.2.3.3 Give Feedback

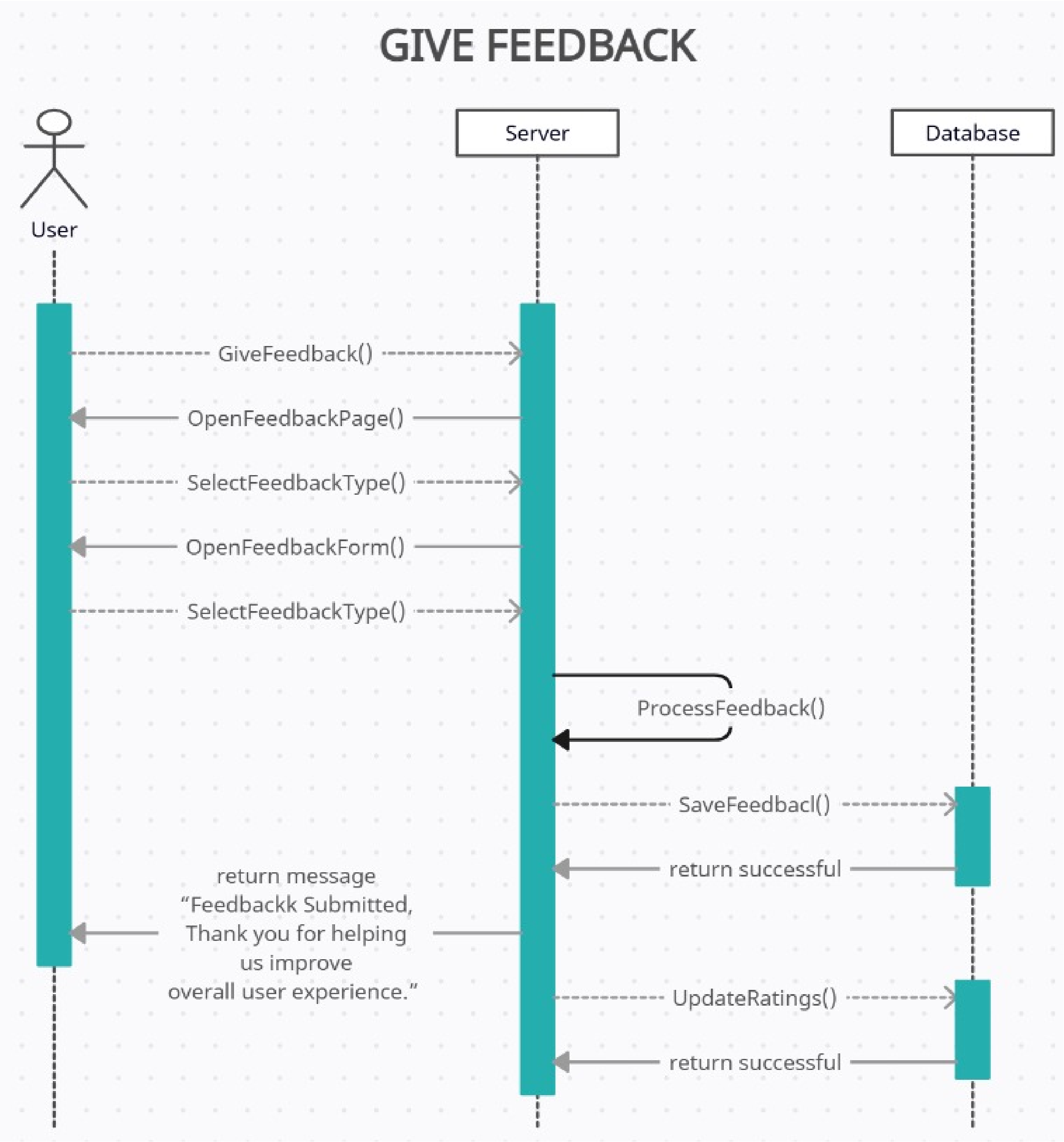
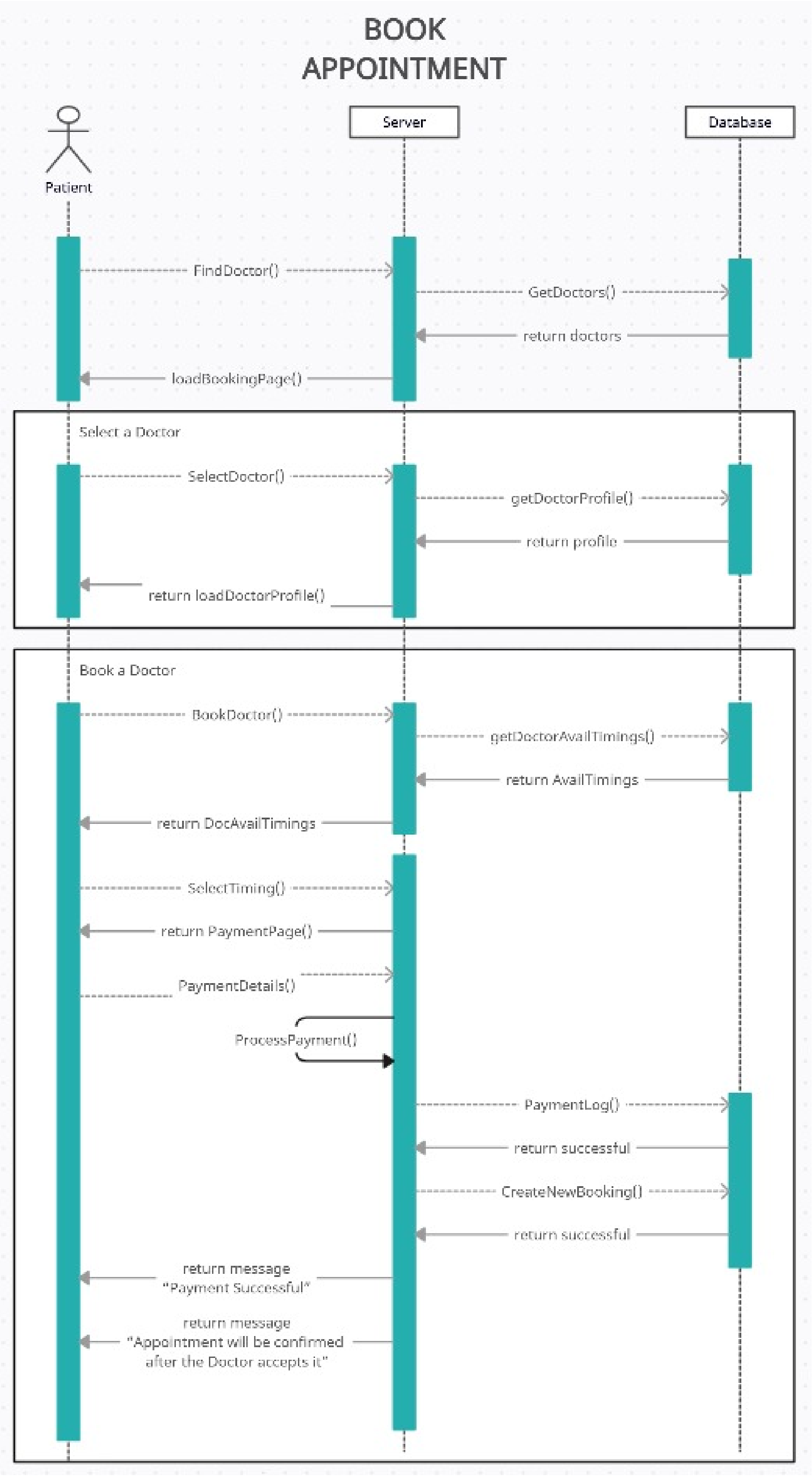


Figure 3.6: Give Feedback

3.2.3.4 Book Appointment



3.2.3.5 Chatbot Interaction

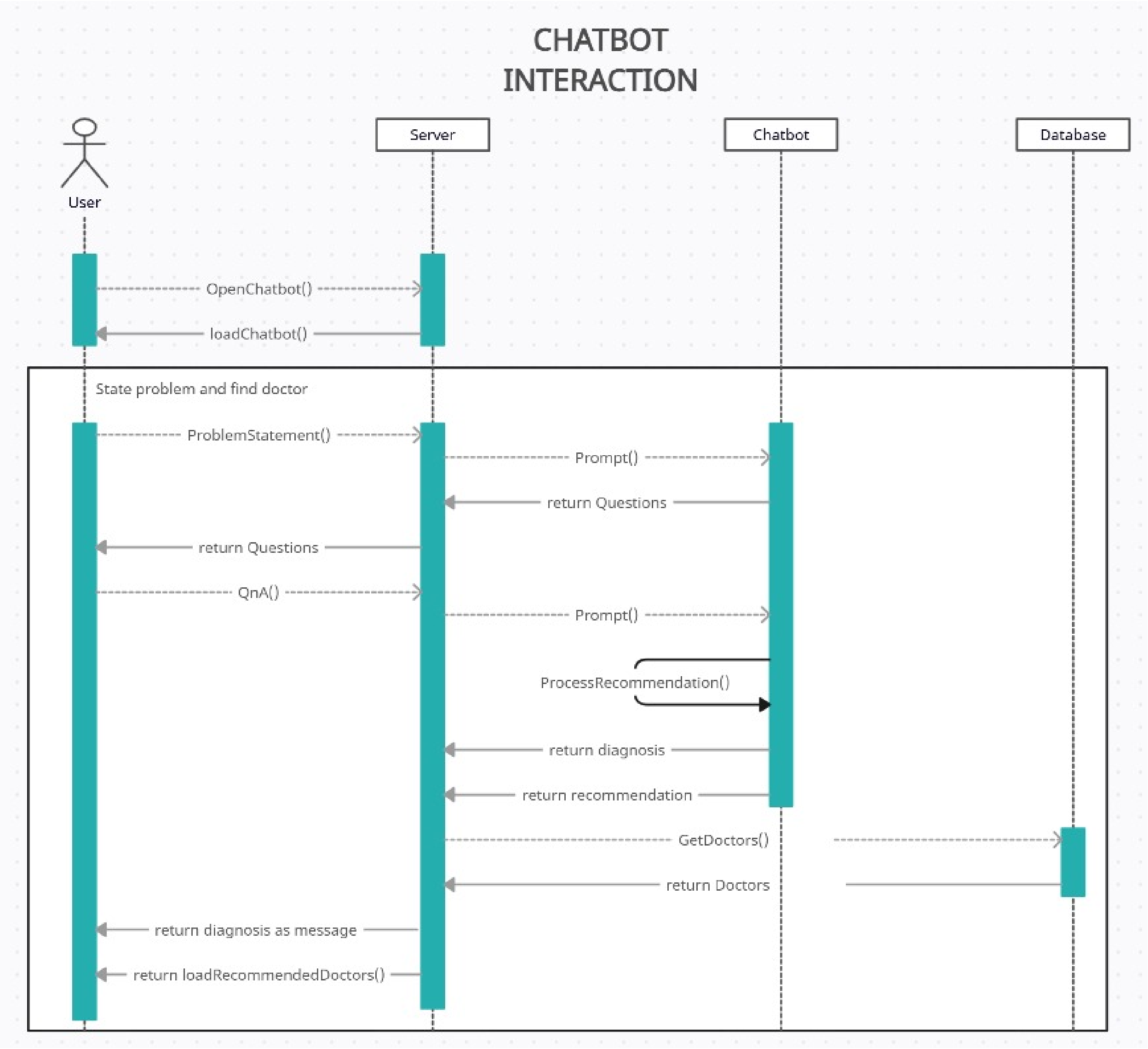


Figure 3.7: Chatbot Interaction

3.2.3.6 Diagnostic History

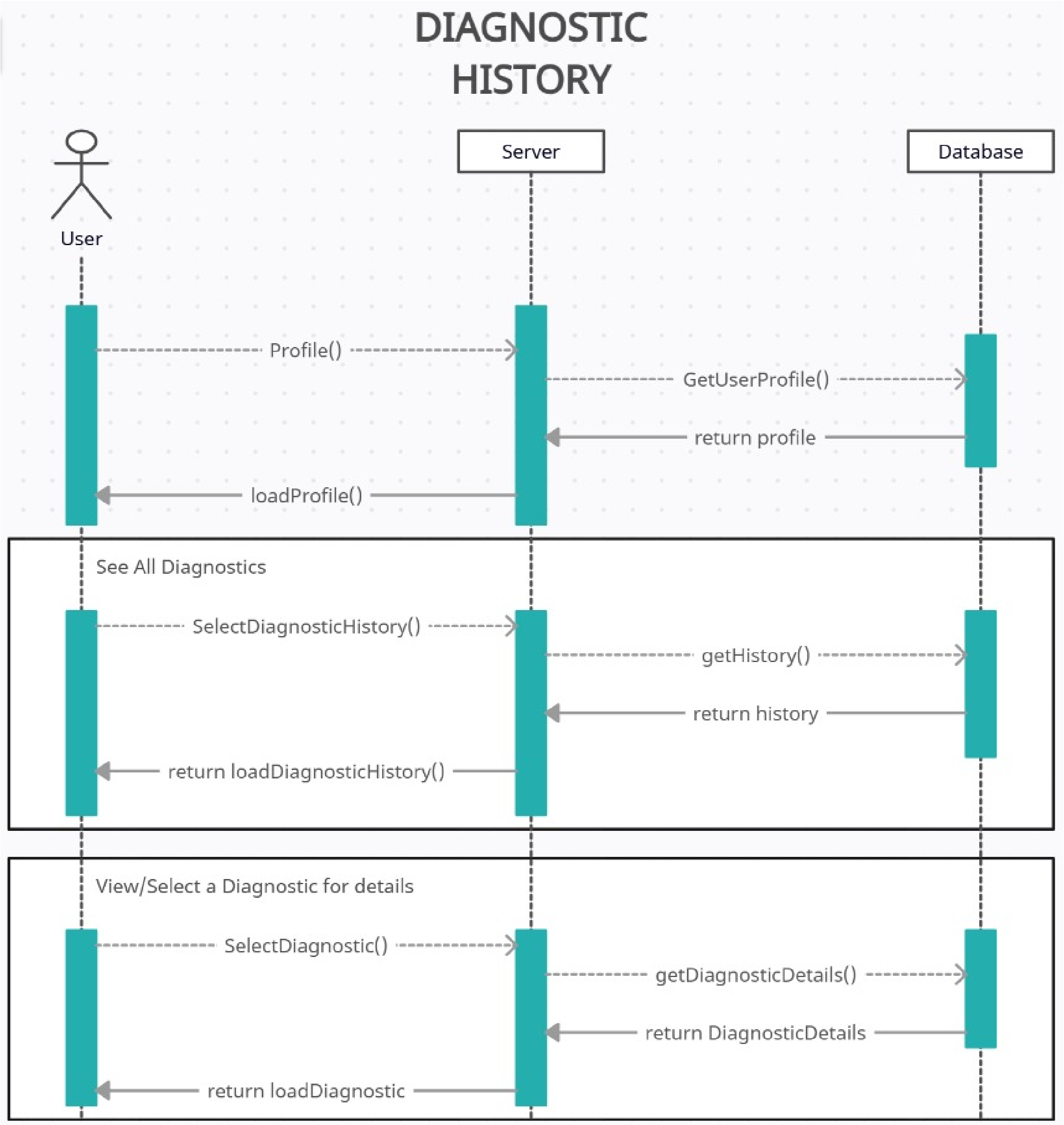


Figure 3.8: Diagnostic History

3.2.3.7 Manage Appointments

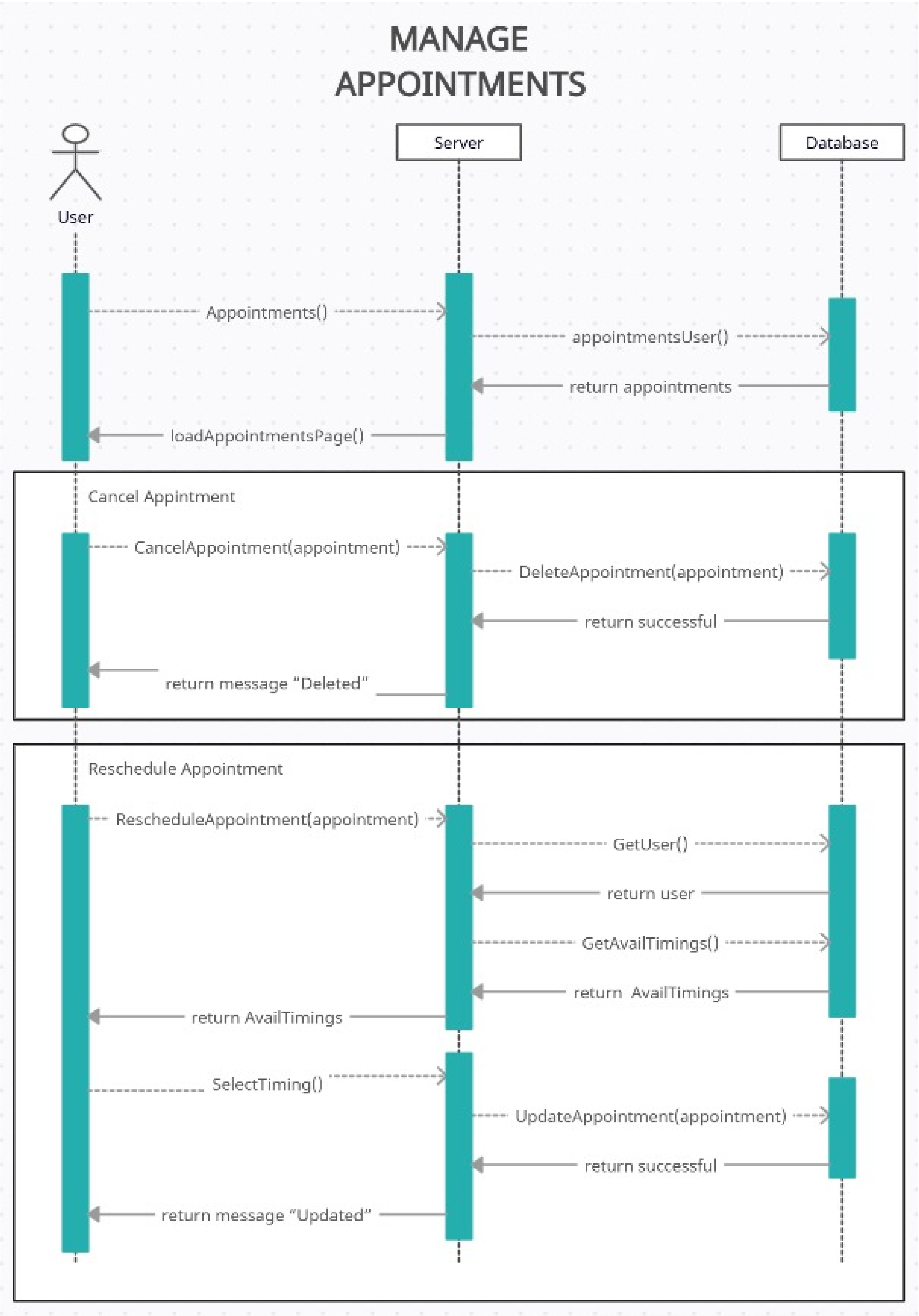


Figure 3.9: Manage Appointments

3.2.3.8 Upload Scans and generate Report

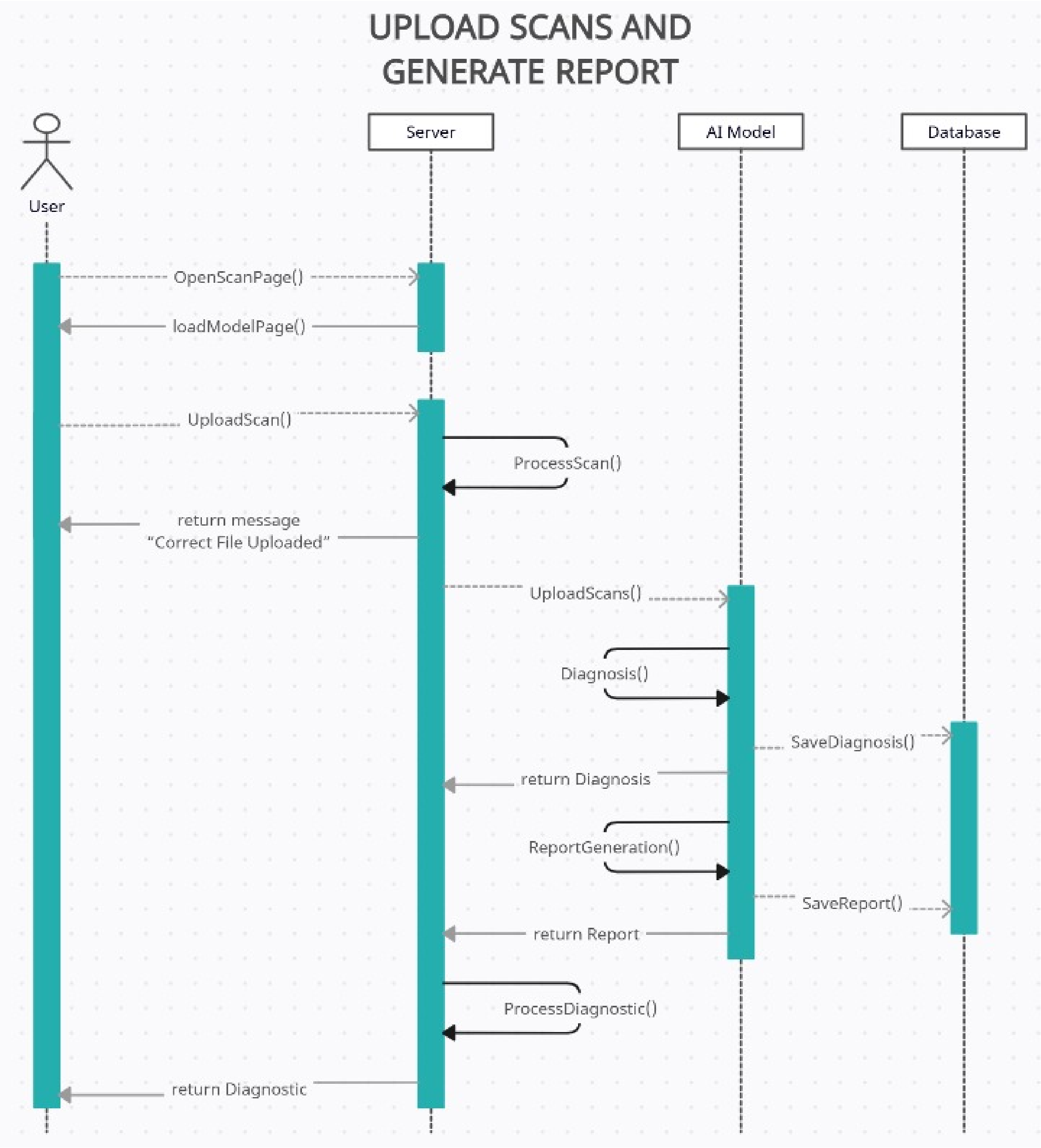


Figure 3.10: Upload Scans and generate Report

3.2.4 State Transition Diagram

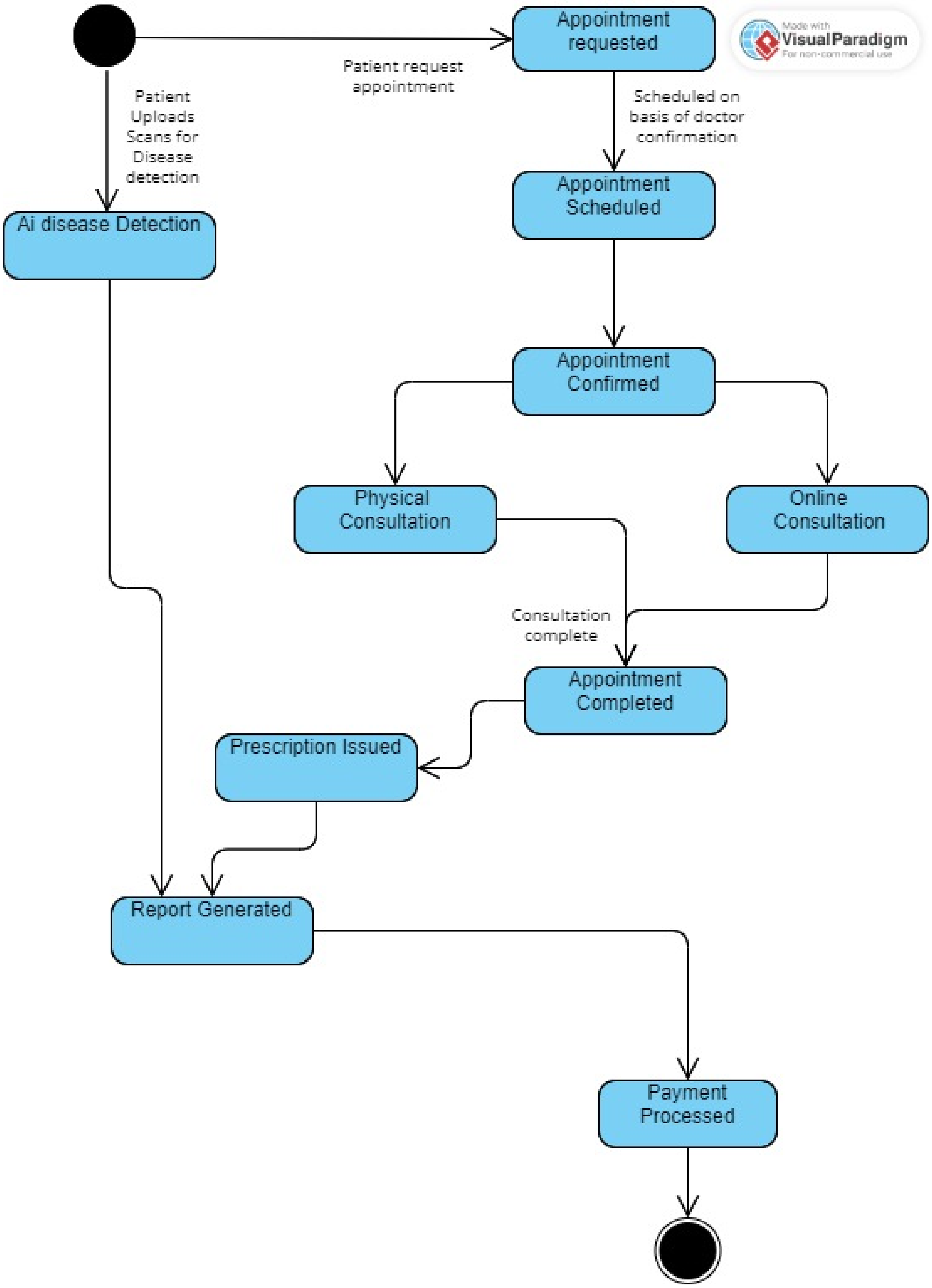


Figure 3.11: State Transition Diagram

3.3 Data Design

Data Design for Health Bridge

The Health Bridge system utilizes a MongoDB database to manage the various entities involved in the online appointment booking process, video consultations, AI-based disease detection, and automated report generation. MongoDB's flexibility and scalability are well-suited for managing the complex and dynamic nature of healthcare data, such as patient records, doctor schedules, and AI-generated reports.

The major entities in the system include **Appointments, Doctors, Patients, Payments, and Diagnosis**. These entities are stored as collections in MongoDB, each representing a key part of the healthcare interaction workflow. Below is a breakdown of the core entities, their relationships, and how they support the system's overall functionality.

3.3.1 Appointment Data

The appointment collection captures all patient-doctor appointment details. Key fields include:

- appointmentid: Unique identifier for the appointment.
- doctorid: Refers to the doctor assigned to the appointment.
- patientid: Refers to the patient booking the appointment.
- astatus, aaccepted: Manage appointment statuses (e.g., confirmed, canceled).
- adate, atime: Date and time for the appointment.
- appointmenttype: Type of consultation (e.g., "Regular Checkup" or "Video Consultation").
- areminder: Automated reminders for upcoming appointments.
- paymentid: Links to the payment details.

This entity interacts closely with **Doctors** and **Patients** collections to ensure appointment scheduling and status updates. Video consultations are facilitated through a third-party integration that links to the **appointment** entity.

3.3.2 Doctor Data

The doctor collection stores essential details about the healthcare professionals in the system. Key fields include:

- doctorid: Unique identifier for each doctor.
- name, email, phone, specialization: Personal and professional details of the doctor.
- dConsultFee, videoCallFacility: Information regarding consultation fees and video consultation availability.
- dOperatingDay, workingslots: Track the doctor's availability across different days and times.

Doctors' profiles can include **qualifications** from the qualification collection and **certifications** from the certification collection, providing complete professional details. This data supports both online appointments and video consultations, ensuring accurate scheduling and professional validation.

3.3.3 Patient Data

The patient collection holds vital information about the patients:

- patient: Unique identifier for each patient.
- name, email, phone, address, city: Personal details.
- medicalHistory, allergies, medicationList: Track the patient's medical conditions and history.
- appointmenthistory: List of past appointments with details on doctors consulted and services availed.

This data helps doctors provide personalized treatment and allows patients to manage their health records easily.

3.3.4 Prescription and Reports

The prescription collection links directly to appointments and doctors. It includes:

- prescriptionid: Unique identifier for prescriptions.

- doctorid, patientid, appointmentid: Links to the corresponding doctor, patient, and appointment.
- prescriptionStatus, Prescription: Details of the medications prescribed.
- prescriptionCategory: Specifies the medical category (e.g., Cardiology).
- Similarly, the report collection stores AI-generated medical reports:
- reportid: Identifier for each medical report.
- reportDocument, reportSummary, reportReviewer: Stores the generated report document and review comments.
- diseaseDetectionResult: Stores results of AI-based scan analysis (e.g., liver disease detection based on uploaded X-rays).

These entities play a key role in automating healthcare delivery, supporting both in-person and virtual consultations, and providing diagnostic assistance through AI.

3.3.5 Payment Data

The payment collection handles all financial transactions in the system:

- paymentid, paymentamount, paymentdate, paymentmethod: Capture details of payments made by patients for appointments, lab tests, and other services.
- appointmentid: Links payments to specific appointments to ensure financial records are in sync with healthcare services provided.

3.3.6 Third-Party Integrations

Health Bridge integrates with third-party APIs to enable online video consultations between patients and doctors. These APIs are linked to the appointment collection, where the appointment type is marked as "Video Consultation" if it involves a video call. This integration ensures that online consultations are smooth and seamless, improving the overall user experience.

3.3.7 Artificial Intelligence Model

A core feature of Health Bridge is its AI-enabled disease detection model, specifically designed for liver disease detection using MRI/CT scans. The patient uploads the scan through the system, which is then processed by the AI model. The results are stored in the report collection under the field AIdiseasedetectionresult. This data is further used for generating a detailed report for the patient.

3.3.8 Conclusion

In conclusion, the Health Bridge system leverages MongoDB's document based structure to efficiently manage healthcare data. The data is organized into collections that reflect real-world entities such as appointments, doctors, patients, and payments. These data structures are further enhanced with AI and third-party integrations to provide a comprehensive and secure healthcare platform that supports both in-person and virtual consultations.

Bibliography

[1] Biomed central, 2024.

[2] D-rax: Domain-specific radiologic assistant leveraging multi-modal data and expert model predictions. *arXiv*, 2024.

[3] The express tribune, 2024.

[4] Fact-aware multimodal retrieval augmentation for accurate medical radiology report generation. *arXiv*, 2024.

[5] Ihme: Institute for health metrics and evaluation, 2024.

[6] Multimodal healthcare ai: Identifying and designing clinically relevant vision-language applications for radiology. *ACM*, 2024.

[7] Pbs government website, 2024.

[8] Radio rag: Factual large language models for enhanced diagnostics in radiology using dynamic retrieval augmented generation. *arXiv*, 2024.

[9] Rule: Reliable multimodal rag for factuality in medical vision language models. *arXiv*, 2024.