

AI HEALTH ASSISTANT

A PROJECT REPORT

Submitted by

NIGAM SHASTRI [22BEIT30188]

PUJAN SHAH[22BEIT30183]

HENIL PRAJAPATI[22BEIT54009]

URMI RAMANI[22BEIT30211]

In fulfillment for the award of the degree Of

BACHELOR OF ENGINEERING

In

INFORMATION TECHNOLOGY



**LDRP INSTITUTE OF TECHNOLOGY AND RESEARCH,
GANDHINAGAR**

Kadi Sarva VishwaVidyalaya, Gandhinagar

2025-26

LDRP Institute of Technology and Research

Information Technology Department



CERTIFICATE

This is to certify that the Project Work entitled **“AI HEALTH ASSISTANT”** has been carried out by **Nigam S. Shastri(22BEIT30188)** under my guidance in fulfilment of the degree of Bachelor of Engineering in Information Technology (7th Semester) of Kadi Sarva VishwaVidyalaya University, Gandhinagar during the academic year 2025.

Guide:

Prof. Dushyant Chawada
Internal Guide,
LDRP-ITR.

Dr. Mehul Barot
H.O.D. IT Dept.,
LDRP ITR.

LDRP Institute of Technology and Research

Information Technology Department



CERTIFICATE

This is to certify that the Project Work entitled **“AI HEALTH ASSISTANT”** has been carried out by **Pujan Shah(22BEIT30183)** under my guidance in fulfilment of the degree of Bachelor of Engineering in Information Technology (7th Semester) of Kadi Sarva VishwaVidyalaya University, Gandhinagar during the academic year 2025.

Guide:

Prof. Dushyant Chawada
Internal Guide,
LDRP-ITR.

Dr. Mehul Barot
H.O.D. IT Dept.,
LDRP ITR.

LDRP Institute of Technology and Research

Information Technology Department



CERTIFICATE

This is to certify that the Project Work entitled **“AI HEALTH ASSISTANT”** has been carried out by **Henil Prajapati(22BEIT54009)** under my guidance in fulfilment of the degree of Bachelor of Engineering in Information Technology (7th Semester) of Kadi Sarva VishwaVidyalaya University, Gandhinagar during the academic year 2025.

Guide:

Prof. Dushyant Chawada
Internal Guide,
LDRP-ITR.

Dr. Mehul Barot
H.O.D. IT Dept.,
LDRP ITR.

LDRP Institute of Technology and Research

Information Technology Department



CERTIFICATE

This is to certify that the Project Work entitled **"AI HEALTH ASSISTANT"** has been carried out by **Urmi Ramani(22BEIT30211)** under my guidance in fulfilment of the degree of Bachelor of Engineering in Information Technology (7th Semester) of Kadi Sarva VishwaVidyalaya University, Gandhinagar during the academic year 2025.

Guide:

Prof. Dushyant Chawada
Internal Guide,
LDRP-ITR.

Dr. Mehul Barot
H.O.D. IT Dept.,
LDRP ITR.

ACKNOWLEDGEMENT

I express my sincere gratitude towards my guide Prof. Dushyant Chawada for her constant help, encouragement, suggestions and inspiration throughout the seminar work. Without her invaluable advice, suggestions and assistance it would not have been possible for me to complete this seminar work.

I wish to thank the Information technology Department of a LDRP-ITR for their sympathetic co-operation. My sincere thanks to all the authors whose literature I have used as a reference of my work. I am very thankful to my classmates, my Family & my Friends who supported throughout the semester.

Nigam Shastri(22BEIT30188)
Pujan Shah(22BEIT30183)
Henil Prajapati(22BEI54009)
Urmi Ramani(22BEIT30211)

ABSTRACT

The AI Health Assistant is an intelligent, user-centric healthcare application designed to enhance medical support through conversational AI and predictive analytics. Leveraging Natural Language Processing (NLP), Machine Learning (ML), and real-time chatbot interaction, the system provides users with instant responses to health-related queries, preliminary symptom analysis, and disease prediction. The assistant is integrated with advanced models such as BERT and Gemini Pro for empathetic conversations and accurate information delivery. It features modules for health risk prediction (e.g., diabetes, heart disease), a mental health companion, and document-based medical information retrieval using semantic search. Designed with both Flask and Streamlit versions for deployment flexibility, the AI Health Assistant bridges the gap between users and accessible healthcare guidance, especially in remote and underserved areas. This solution aims to reduce the burden on healthcare systems by enabling early detection, self-assessment, and health education.

LIST OF FIGURES

NO	NAME	PAGE NO
1	Work Breakdown Structure	11
2	Activity Network Requirement	11
3	Gantt Chart	12
4	Functional Diagram	21
5	Class Diagram	25
6	Usecase Diagram	26
7	Sequence Diagram	27
8	Entity Relationship Diagram	31
9	Context Level DFD	32
10	First level DFD	33
11	Activity Diagram	34

LIST OF TABLES

NO	NAME	PAGE NO
1	Users Details	29
2	Chathistorry Details	29
3	Symptoms Table	30
4	DiseasePrediction Table	30

TABLE OF CONTENTS

NO	CHAPTER NAME	PAGE NO
	Acknowledgement	Iv
	Abstract	V
	List Of Figures	Vi
	List Of Tables	Vii
	Table Of Contents	Viii
1	Introduction	1
	1.1 Introduction	2
	1.2 Scope	2
	1.2.1 Current Scope	2
	1.2.2 Future Scope	2
	1.3 Project Summary and purpose	3
	1.3.1 Project Summary	3
	1.3.2 Purpose	4
	1.4 Objectives	5
2	Technology and Literature Review	6
	2.1 Tools and Technology	7
	2.1.1 web development & UI	7
	2.1.2 NLP	7
	2.1.3 AI & ML	7
	2.2 Project Planning	8
	2.2.1 Project Development Approach	8
	2.2.2 Milestone and Deliverables	9
	2.2.3 Group Dependencies	9
	2.3 Project Scheduling	10
3	System Requirements Study	13
	3.1 User Characteristics	14
	3.2 Software and Hardware Requirements	14
	3.2.1 Software Requirements	14
	3.2.2 Hardware Requirements	15
	3.2.3 Functional Requirements	15
	3.2.4 Non Functional Requirements	16
	3.3 Constraints	17
	3.3.1 Parallel Operations	17
	3.3.2 Reliability Requirements	17

	3.3.3 Criticality of the application	17
	3.3.4 Safety and security considerations	17
	3.3.5 Hardware limitations	17
	3.3.6 Regulatory Policies	18
	3.3.7 General Constraint	18
	3.4 Assumptions and Dependencies	18
	3.4.1 Assumptions	18
	3.4.2 Dependencies	18
4	System Analysis	20
	4.1 Study of current system	21
	4.2 Modules and Functionality of proposed system	21
	4.3 Feasibility Study	22
	4.3.1 Technical feasibility	22
	4.3.2 Operational feasibility	22
	4.3.3 Economic feasibility	23
	4.4 Requirement Validation	23
	4.5 Class Diagram	24
	4.6 Use Case Diagrams	25
	4.7 Sequence Diagram	26
5	System Design	28
	5.1 Database Design	29
	5.2 Entity Relationship Diagram	31
	5.3 Data Flow Diagram	31
	5.3.1 Context level	32
	5.3.2 First level DFD	33
	5.4 Activity Diagram	34
6	Testing	35
	6.1 Testing Plan	36
	6.1.1 Planning Steps	36
	6.2 Testing Strategies	40
	6.3 TEST CASE	42
7	Limitation and Future Enhancement	46
	7.1 Limitations	47
	7.2 Future Enhancement	47
8	Bibliography	48

8.1 Conclusion	49
8.2 Bibliography	50

Appendix A

Appendix B

1 INTRODUCTION

- **INTRODUCTION**
- **SCOPE**
- **PROJECT SUMMARY AND PURPOSE**
- **OBJECTIVES**

1.1 Introduction

In today's fast-paced world, timely access to healthcare information and services is a growing necessity. With the increasing burden on healthcare infrastructure, especially in remote and underserved regions, there is a pressing need for accessible, intelligent, and scalable solutions. The integration of Artificial Intelligence (AI) in healthcare offers a promising pathway to address these challenges.

The **AI Health Assistant** is an innovative application designed to assist users in managing their health proactively. It combines the power of Natural Language Processing (NLP), Machine Learning (ML), and conversational AI to deliver personalized health insights and recommendations. Through a user-friendly chatbot interface, individuals can ask health-related questions, assess symptoms, receive potential disease predictions, and gain access to medical knowledge without the need for immediate human intervention.

1.2 Scope

1.2.1 Current Scope

The current implementation of the **AI Health Assistant** focuses on providing essential healthcare-related support through intelligent automation and conversational AI. The system is designed to operate as a virtual health companion, capable of interacting with users in real time and offering valuable insights based on user input. Key areas within the present scope include:

- Symptom-Based Disease Prediction:**
The assistant can predict common health conditions such as diabetes, heart disease, and Parkinson's disease using trained machine learning models based on user-provided health parameters.
- Conversational Chatbot:**
An interactive chatbot interface allows users to ask general health-related questions. It provides context-aware, informative, and empathetic responses using advanced NLP models like BERT or Gemini.
- Mental Health Support:**
The assistant includes a basic mental health module that identifies emotional cues from text and responds empathetically to provide comfort, suggestions, and encouragement.
- Medical Document Search (Optional Module):**
Users can upload health-related documents, and the system uses semantic search techniques (e.g., Sentence-BERT and FAISS) to answer queries based on those documents.
- Deployment Flexibility:**
The AI Health Assistant is available in both Flask and Streamlit versions, allowing

deployment as a web application or a standalone desktop tool for accessibility and user convenience.

6. User Privacy and Local Execution:

All processing can be done locally (if required), ensuring user data remains private and secure without the need for cloud-based storage.

1.2.2 Future Scope

The AI Health Assistant, in its current form, lays the foundation for an intelligent, accessible, and scalable healthcare solution. However, there are several areas where the system can be further expanded and enhanced in the future to meet growing healthcare needs and technological advancements:

1. Integration with Wearable Devices:

Future versions can integrate with smartwatches, fitness bands, and medical IoT devices to monitor real-time vitals such as heart rate, blood pressure, blood sugar levels, and oxygen saturation for continuous health tracking.

2. Voice-Based Interaction:

Incorporating voice recognition and speech synthesis will make the assistant more accessible, particularly for elderly users, visually impaired individuals, and those with limited typing ability.

3. Electronic Health Record (EHR) Integration:

The assistant can be extended to securely access and analyze users' electronic medical records, enabling more personalized and context-aware responses and recommendations.

4. Telemedicine Support:

A teleconsultation feature could allow users to directly schedule appointments or connect with healthcare professionals based on their symptom analysis and needs.

5. Advanced Mental Health Companion:

The mental health module can be expanded using deep learning models trained on empathetic and therapeutic conversations to offer deeper emotional support and crisis detection.

6. Multi-Language Support:

Expanding language capabilities will enable users from diverse linguistic backgrounds to interact with the system comfortably in their native language.

7. Real-Time Disease Outbreak Monitoring:

With real-time data aggregation from users, the assistant could help identify and report potential disease outbreaks or public health issues.

8. Cloud-Based Analytics and Reporting Dashboard:

For clinics or institutions, a backend analytics dashboard could be developed to monitor user trends, risk clusters, and predictive reports for large-scale health monitoring.

1.3 Project summary and Purpose

1.3.1 Project Summary

The **AI Health Assistant** is an intelligent healthcare support system designed to offer users personalized, real-time medical guidance using Artificial Intelligence. It combines machine learning, natural language processing, and conversational AI to assist users in symptom checking, disease prediction, and mental health support.

The platform enables users to interact through a chatbot interface, ask health-related queries, and receive informative and empathetic responses. It supports disease risk prediction for conditions like diabetes, heart disease, and Parkinson's based on user input. Additionally, a mental health module analyzes emotional tone and provides supportive conversation.

Developed with deployment options in both **Flask** and **Streamlit**, the assistant ensures flexibility for both desktop and web environments. The application also features optional semantic document search capabilities, allowing users to extract information from uploaded medical content using SBERT and FAISS.

By providing proactive, accessible, and intelligent health support, the AI Health Assistant contributes to improved health literacy, early risk detection, and reduced strain on traditional healthcare systems, especially in remote or underserved areas.

1.3.2 Purpose

The primary purpose of the **AI Health Assistant** is to provide accessible, intelligent, and real-time healthcare support to users by leveraging the capabilities of Artificial Intelligence and Machine Learning. This system is designed to:

1. **Empower users** to assess symptoms and receive preliminary health guidance without immediate dependence on healthcare professionals.
2. **Assist in early detection** of diseases such as diabetes, heart disease, and Parkinson's through predictive modeling based on user input.
3. **Offer mental health support** through emotionally aware conversations to promote psychological well-being.
4. **Bridge the gap** between users and medical knowledge by delivering personalized, accurate, and understandable health information through a chatbot interface.
5. **Enhance healthcare accessibility** for individuals in remote, underserved, or resource-limited environments.
6. **Encourage health awareness** and proactive management of one's well-being using technology-driven solutions.

By fulfilling these objectives, the AI Health Assistant aims to reduce diagnostic delays, support informed decision-making, and improve the overall efficiency of healthcare delivery.

1.4 Objectives

The AI Health Assistant project aims to fulfill the following key objectives:

1. **Develop an AI-powered chatbot** that can interact with users in natural language to address health-related queries and concerns effectively.
2. **Implement machine learning models** to predict the likelihood of common diseases such as:
 - Diabetes
 - Heart disease
 - Parkinson's diseasebased on user input parameters (e.g., age, symptoms, vitals).
3. **Integrate a mental health module** capable of:
 - Detecting emotional tone using NLP techniques
 - Responding empathetically to support users' mental well-being
4. **Enable document-based query resolution** by implementing semantic search (e.g., SBERT + FAISS) to retrieve relevant information from uploaded medical documents.
5. **Ensure multi-platform deployment** through both:
 - **Flask** (for web application)
 - **Streamlit** (for rapid prototyping and desktop execution)
6. **Ensure user data privacy and local processing** to maintain confidentiality and security of health-related information.
7. **Promote health awareness and early intervention** by providing users with accurate, AI-generated insights that encourage timely medical consultation if needed.
8. **Create a scalable and modular system** that can be extended with features like voice input, multilingual support, real-time vitals from wearables, and integration with electronic health records.

2 TECHNOLOGY AND LITERATURE REVIEW

- **TOOLS AND TECHNOLOGY**
- **PROJECT PLANNING**
- **PROJECT SCHEDULING**
- **COST ESTIMATION**

2.1 Tools and Technology

2.1.1 Web Development & UI:

The AI Health Assistant features a clean, intuitive, and responsive user interface designed to provide a seamless experience across devices. The application is developed using two main frameworks—**Streamlit** and **Flask**—to offer flexibility in deployment and usage. The **Streamlit version** offers a desktop-style interface ideal for quick prototyping and user-friendly interaction. It allows users to input health parameters, receive predictions, chat with the assistant, and upload documents through a sidebar-driven layout. On the other hand, the **Flask-based web application** provides a more customizable and scalable frontend, built using **HTML, CSS, and JavaScript**. This version supports a multi-page structure with responsive design powered by **Bootstrap** or **Tailwind CSS**, enabling better styling, animations, and user control. Both versions support features like real-time chatbot communication, health prediction forms, and secure document upload functionality. The interface is designed to be simple yet powerful, making healthcare assistance accessible even to users with minimal technical background.

2.1.2 Natural Language Processing (NLP):

Natural Language Processing (NLP) plays a central role in the AI Health Assistant by enabling the system to understand, interpret, and respond to user queries in a human-like manner. The chatbot is powered by advanced NLP models such as **BERT** and **Gemini Pro**, which allow it to comprehend the context of health-related questions and generate informative, empathetic responses. Text preprocessing techniques such as tokenization, stemming, and stop-word removal are applied using libraries like **NLTK** and **SpaCy** to clean and prepare user input for analysis. For more advanced use cases, the assistant utilizes **Sentence-BERT (SBERT)** for generating semantic embeddings and **FAISS** for performing similarity-based searches across uploaded medical documents. This allows users to ask natural language questions and receive relevant information extracted from the documents. Additionally, sentiment and emotion analysis modules help in identifying the user's emotional state during conversations, enabling the system to respond in a supportive and sensitive manner. Overall, NLP transforms the AI Health Assistant from a basic Q&A tool into an intelligent, conversational healthcare companion.

2.1.3 Artificial Intelligence & Machine Learning:

Artificial Intelligence (AI) and Machine Learning (ML) form the core of the AI Health Assistant, enabling it to provide intelligent, data-driven healthcare insights. The system leverages supervised learning algorithms such as **Logistic Regression**, **Decision Trees**, and **Random Forests**—implemented using **Scikit-learn**—to predict the risk of diseases like diabetes, heart disease, and Parkinson's based on user-provided health data. These models are trained on publicly available healthcare datasets and optimized for

accuracy, interpretability, and real-time performance. For more advanced capabilities, deep learning frameworks like **TensorFlow** and **Keras** can be integrated to build emotion detection or complex diagnostic models. AI is also used in the chatbot module to generate context-aware replies and guide users toward appropriate health actions. Furthermore, the system incorporates data preprocessing techniques, model evaluation metrics, and visualizations using libraries like **Pandas**, **NumPy**, and **Matplotlib** to enhance transparency and user understanding. By embedding machine learning into the platform, the assistant transforms raw health inputs into meaningful predictions, offering users valuable early warnings and recommendations for better health management.

2.2 Project Planning

2.2.1 Project Development Approach:

The model that is referred for the development of the project is INCREMENTAL model. It combines elements of the waterfall model applied in an iterative fashion. In this process the phases are same as waterfall but the advantage is that when first phase is done it is incremented and then the other phases are carried with the same cycle. Here in this add ons on each phase can be added according to the need of the client and the project.

Phases are as follows:

1. Communication
2. Planning
- 3 .Modeling: Includes Designing
4. Construction
5. Deployment: Feedback,Delivery

Each phases are iteratively carried out. Main reason for using this then any other is waterfall has the drawback of iterations, if there is any other requirement added later on then this is not possible to add up in it, Spiral model has disadvantage that it need more manpower and even it is for multiple transactions or multiple tasks handling projects and so does the time consumption is more in it for those projects.

Planning is essential cause multiple software teams works in parallel on different system functions. Scalability should be obtained in any of the project selected but it is not available in waterfall cause of few drawbacks.

2.2.2 Milestones and Deliverables

➤ Month1:- Milestones & Deliverables

Milestones	Deliverables
Study about our web application requirement, planning	Analysis Report
Understand a project definitions and basic terms and logic for Parameter Evaluation.	
Gathering the requirements of the project using different fact finding techniques.	Analysis Report
Still Continue with Requirement's study.	

➤ Month 2-4: -Milestones & Deliverables

Milestones	Deliverables
System Analysis	Analysis Report
System Design including various diagrams	SRS

2.1.3 Group Dependencies

The team structure depends on the management style of the organization, number of people in the team, their skill levels and the problem difficulty. Our team organization is *democratic decentralized* which doesn't have a team leader. Decision is made by all of us and the problems were discussed and solved by all of us after consulting and discussing with our external guide and project guides.

2.3 Project Scheduling

Project scheduling involves separating the total work in a project into separate activities and judging the time required to complete these activities. Usually, some of these activities are carried out in parallel.

I. Work Breakdown Structure

Work Breakdown Structure is used to decompose a given task set recursively into small activity [Fig (2.3)].

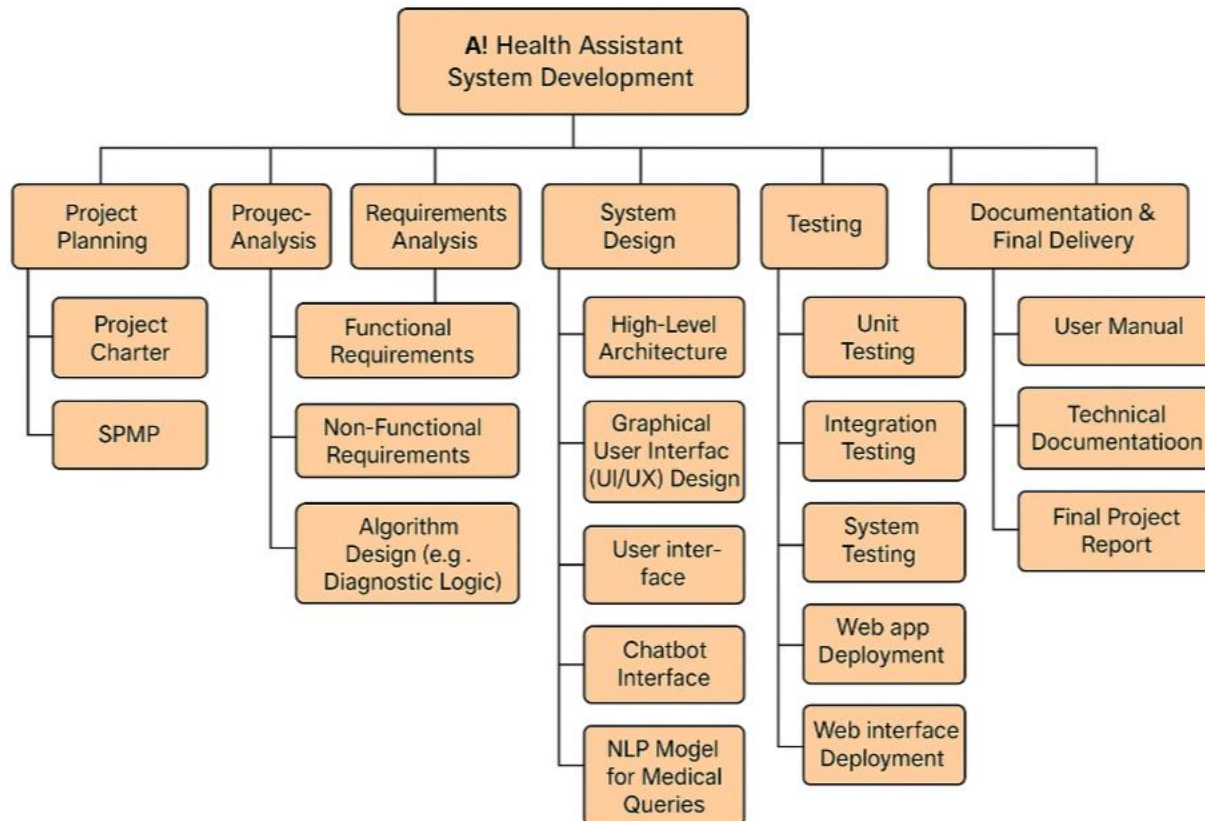
II. Activity Network Representation

An activity network shows the different activities making of a project, their estimated duration and interdependencies [Fig (2.4)].

III. Gantt Chart

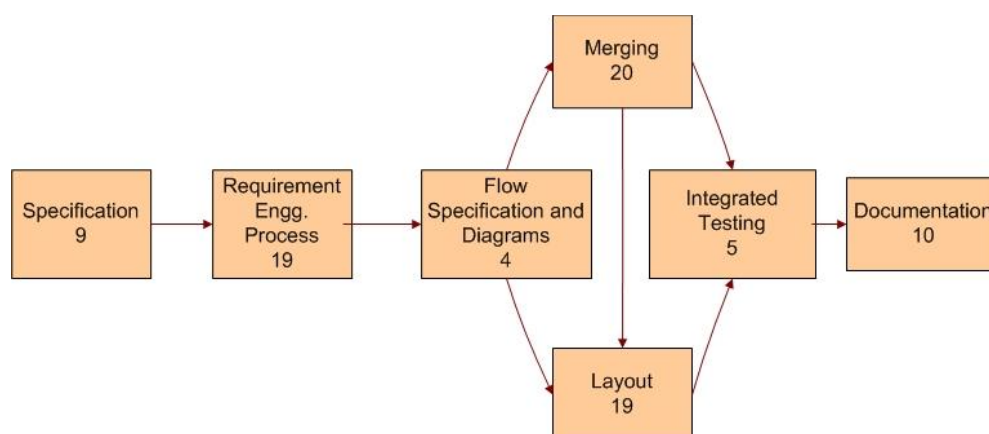
Gantt chart is mainly used to allocate resources to activities. Gantt Charts are useful for resource planning [Fig (2.5)]

(a) Work Breakdown Structure



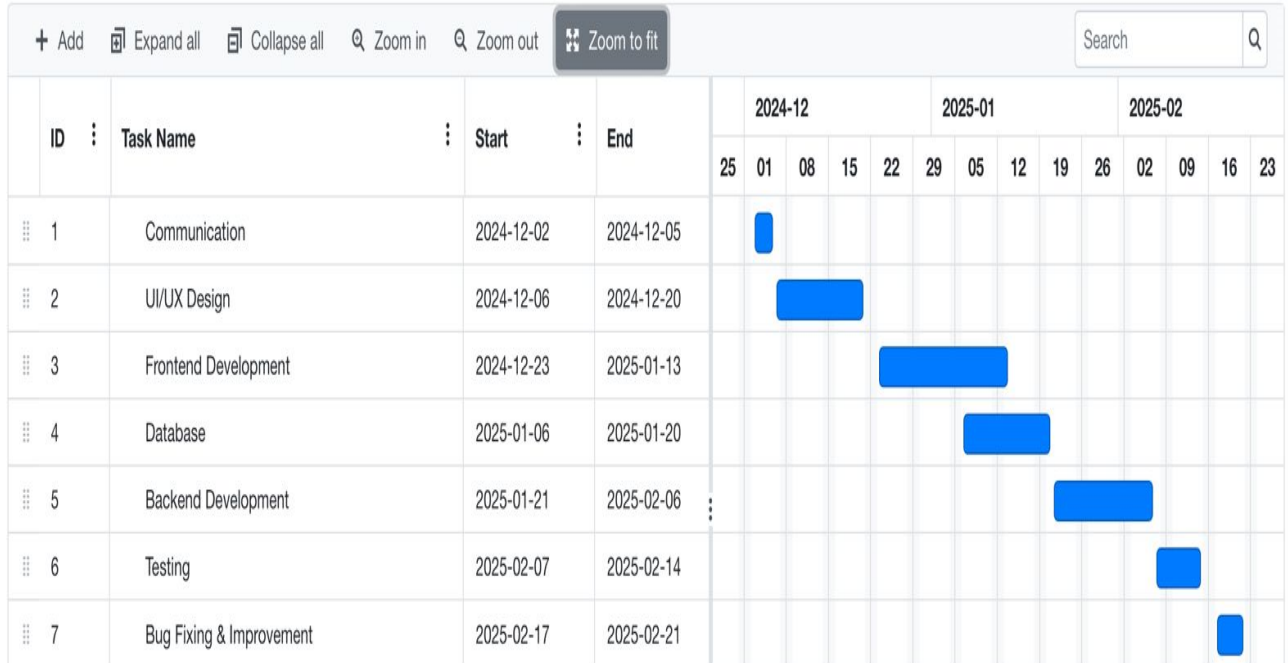
2.1 Work Breakdown Structure

(b) Activity Network Representation



2.2 Activity Network Representation

(c) Gantt chart



2.3 Gantt chart

3 SYSTEM REQUIREMENTS STUDY

- **USER CHARACTERISTICS**
- **HARDWARE AND SOFTWARE REQUIREMENTS**
- **CONSTRAINTS**
- **ASSUMPTIONS AND DEPENDENCIES**

3.1 User Characteristics

Analyzing user characteristics is an important aspect of any project. It allows us to clearly define and focus on who the end users are for the project. Also, it allows checking the progress of the project to ensure that we are still developing the system for the end users. The user must have following characteristics:

- User must have basic knowledge of Computers.
- User should understand the use of all modules.
- User can easily interact with the proposed system.
- User must know the technical terms used in the company for performing different tasks specially related to call logs, payment details, transportation details and report retrieval.
- User should be also being aware about the running process of the system.

3.2 Software and Hardware Requirements

Software and Hardware Requirements are used to describe the minimum hardware and software requirements to run the Software. These requirements are described below.

3.2.1 Software Requirements

- **Client:**
 - Operating System: Windows or Linux
 - Web Browser: Any HTML Compliant Browser
- **Server:**
 - Operating System: WINDOWS or LINUX SERVER OS
 - Database Server: Oracle
 - Technologies:Python , NLP

3.2.2 Hardware Requirements

- **Client:**
 - 1.6 GHZ CPU
 - 256 MB of RAM
 - Internet Connection
 - Monitor
 - Keyboard/Mouse
 - Printer
- **Server:**
 - 1.6 GHZ CPU
 - 1 GB of RAM
 - Internet Connection
 - Monitor
 - Keyboard/Mouse
 - Printer

3.2.3 Functional Requirements

The basic features available in the system are as follows:

- User sign-up/login via email, phone, or social media.
- Set travel preferences.
- Create, join, and manage meet-ups.
- Search and filter tours.
- Book tours and receive confirmations.
- Search and filter hotels.
- View hotel details.
- Book hotels and get confirmation.
- Multiple payment methods (cards, PayPal, wallets).
- Secure payment gateway.
- Receive payment receipts.
- Booking confirmations and reminders.

- Promotions and offers.
- In-app travel guides for destinations.
- Map integration for navigation.
- Live chat for support.
- Leave reviews and ratings for tours and hotels.
- Browse reviews from other users.
- Agency registration and management.
- Update listings and track bookings.
- Modify or cancel bookings.
- Global search bar for tours, hotels, and meet-ups.
- Filters for search results.
- Submit feedback and access customer support.

3.2.4 Non Functional Requirements

Following is a list of non functional requirements:

- **Performance:**

This system should remain accessible 24x7. At least 100 users should be able to access the system altogether at any given time.

- **Security:**

The database of system should not store any password in plain text rather the ceaser cipher text has to be stored.

- **Reliability:**

It can be accessed by the end users 24*7 as an when needed hence is highly reliable for end users.

- **Availability:**

Internet connection for the nodes with the database server is ensured and hence the application will be available any time for access.

- **Portability:**

The developed web application is portable as it can be accessed from any operation system regardless Windows, Mac, Linux provided they have a browser to access Internet.

3.3 Constraints

3.3.1 Parallel Operations

The project is on basis of multi-user. This is used for carrying out updating as well as entry by preventing the redundancy of the data.

3.3.2 Reliability Requirements

Reliability requirements of the system are one of the prime ones in the list. The system is needed to be highly reliable in terms of performance and capable of delivering robust performance. If the reports are generated within 5 seconds then the system is said to be reliable.

3.3.3 Criticality of the Application

The system can stop working on computers with very low internet connection. Other than that there won't be any issues. Apart from these the system should be able to make updates at regular time intervals.

3.3.4 Safety and Security Consideration

Safety and security too are other major concerns of any system. It is necessary to provide safety and security as the system is web application and might be intrude by security threats from the internet. Thus, the code needs to be encrypted and any transaction needs to be done securely.

3.3.5 Hardware Limitations

Hardware Limitations are other constraint of the system. Hardware Limitations should be overcome for better performance of the system. This can be achieved by using minimum and only necessary hardwares.

3.3.6 Regulatory Policies

Regulatory policy is about achieving organization's objectives through the use of regulations, laws, and other instruments to deliver better economic and social outcomes and thus enhance the life of business. Thus the system should be developed by using these regulations to provide better outcome to the company.

3.3.7 General Constraints

The service provided by consultants of Xduce is 24*7 to its client companies. Hence the oracle database server as well as the web application needs to be up 24*7. Internet availability is the major requirement for all the end users to successfully access the application. As well as being a live application security enforcement needs to be tight else there are chances of hacking or intrusion.

3.4 Assumptions and Dependencies

3.4.1 Assumptions

- Database transactions are assumed to be secure and reliable.
- User is the person having enough knowledge for the traversing operation.
- We will provide a user friendly interface so that any user can easily navigate through the system, but he/she should be capable of providing valid credentials for successful login.
- The server used for data storing is always secured.

3.4.2 Dependencies

- The system is dependent upon the user's valid credentials. If user inputs wrong username or password, he/she will not be allowed to login to the system.

- This application depends on the server and internet as all the information is collected and then stored in the server through secure internet connection.
- All the users of the system will be assigned a specific role. According to these roles each and every user will be allowed to access predefined set of features.

4 SYSTEM ANALYSIS

- **STUDY OF CURRENT SYSTEM**
- **MODULES AND FUNCTIONALITY OF PROPOSED SYSTEM**
- **FEASIBILITY STUDY**
- **REQUIREMENTS VALIDATION**
- **CLASS DIAGRAM**
- **SYSTEM ACTIVITY(USECASE DIAGRAM)**
- **SEQUENCE DIAGRAM**

4.1 Study of current system

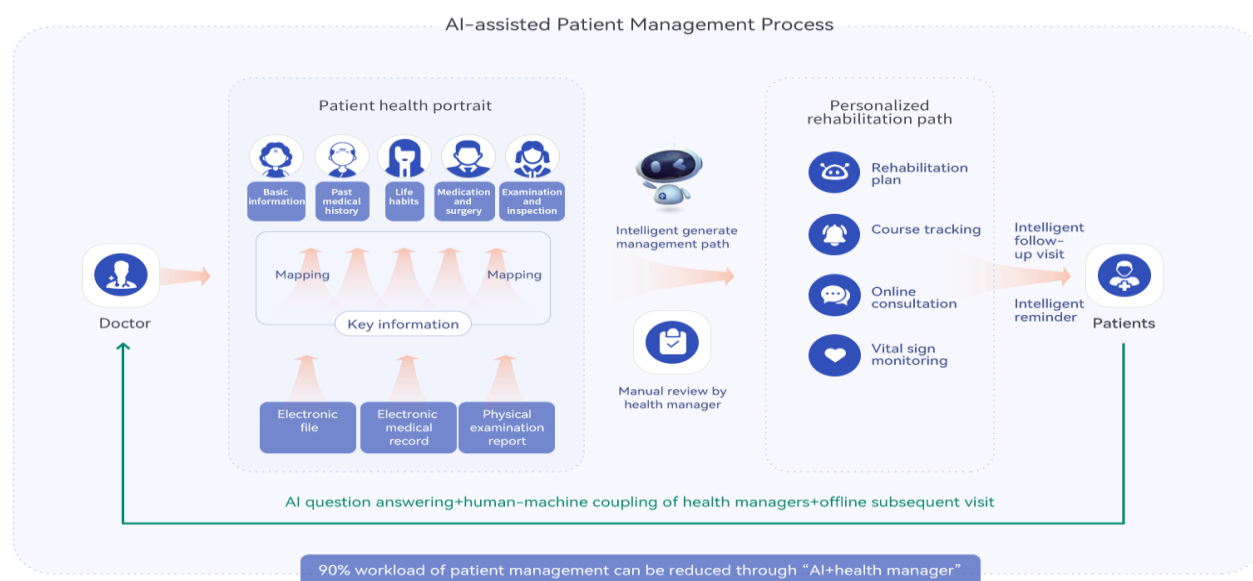
- The call log details, payment details, transportation details of the services provided by Xduce to Client companies are handled manually in current system.
- Manual systems are tedious to manage.
- It is time consuming.
- It is most difficult for report generation.

4.2 Modules and functionality of proposed system

The main functional modules would be as follows instant.

User

- **Technology:** Streamlit or React (can use Flask for backend API)
- **Functionality:**
 - Text input for user health queries
 - Display chatbot responses
 - Show disease prediction results (optional)
 - Clean, responsive design for web use



4.1 Functional Diagram

4.3 Feasibility Study

An important outcome of the preliminary investigation is the determination that the system requested is feasible. The feasibility study is carried out to examine the likelihood that the system will be useful to the organization.

There are four aspects in the feasibility study namely.

- Operational Feasibility
- Technical Feasibility
- Economic Feasibility

4.3.1 Technical Feasibility:

The main purpose of checking Technical Feasibility is to examine whether the current technology is sufficient for the development of the system.

Module	Technology	Feasibility
Chatbot	GPT-4o / Gemini API	✓ Highly feasible — APIs are well-documented and easy to integrate
Frontend (User UI)	Streamlit / React	✓ Feasible — Supports fast prototyping and deployment
Backend API (Optional)	Flask / FastAPI	✓ Feasible — Used for scalable microservices
Disease Prediction (ML)	Scikit-learn, Pandas	✓ Feasible — Simple classification models can be trained
Authentication	Firebase / OAuth	✓ Feasible — Many plug-and-play services available
Deployment	Heroku / Vercel / AWS	✓ Easily deployable using CI/CD or serverless platforms

So, this application is Technically Feasible.

4.3.2 Operational Feasibility:

The main purpose of checking Operational Feasibility is to find out whether the system will be functional after its development and installation or not.

The outcomes of the operational feasibility are as follows:

- This application provides all the data of call log services provided to client companies along with the payment and transportation details.
- They can be administered from remote locations using mail, email or telephone.
- So, it is supposed to improve the working efficiency of user.

So, this application is operationally feasible.

4.3.3 Economic Feasibility:

The main purpose of checking Economical Feasibility is to examine whether the financial investment in the system will meet the organization's requirements or not.

The outcomes of the technical feasibility are as follows:

- Proposed System is developed as web application which is freely available on WWW.
- It uses ASP.NET as a front end that is also freely available.
- The advantages of the system nullify its development cost as the scope and effect of the system are very large.

So, this application is economically feasible.

4.4 Requirements Validation

The Basic validation from user side is to detect wrong information or blank information:

- When the user has entered correct username and password, then he/she is allowed to enter information or to see the information either into the database or from the database.
- If anyone field from username or password remains blank, then user will not be allowed to enter into the system.

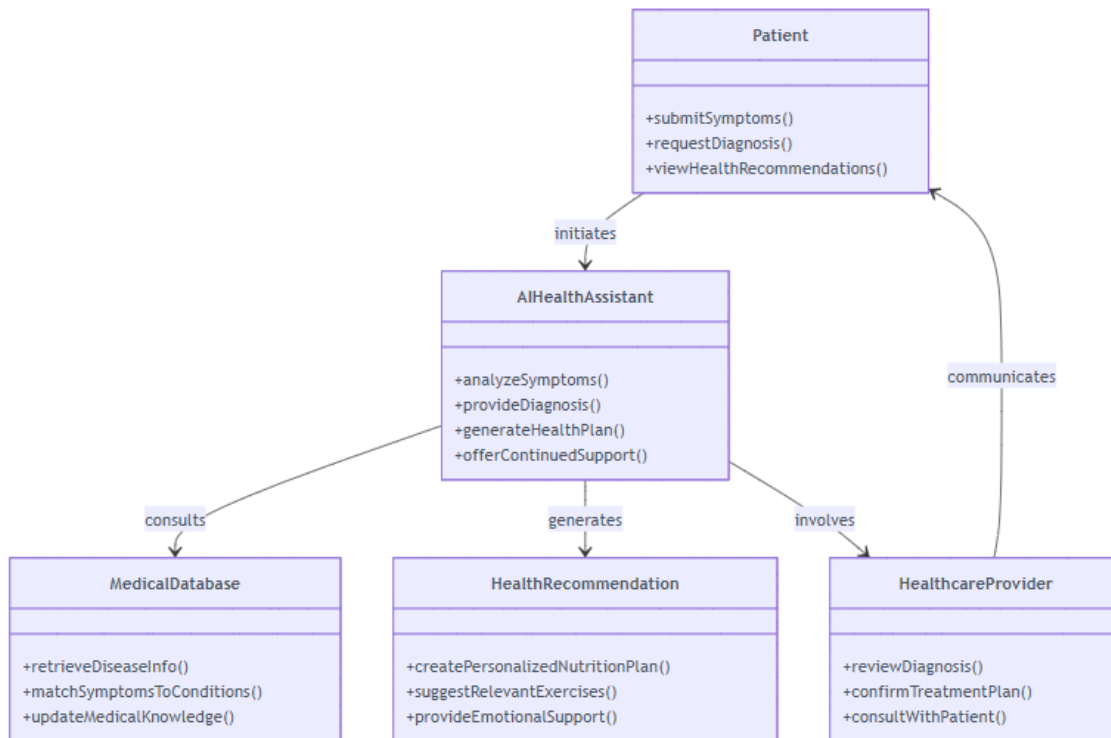
- If a consultant leaves some fields blank in call log form then he/she does not allowed submitting his/her information.
- If user has entered wrong data then accordingly message will generate automatically.

4.5 Class Diagram

A class diagram is a graph of classified elements connected by their various static relationships. It is shown here for Client Support System. This includes the System and the End-users as its main classes. Here are three different fields: Class, Attributes, and Operations.

Class shows the class name, i.e., System, End-user, Data and Sign Clip. They are connected with each other through links and their relation with each other is shown through the numbers represented on the link; here * indicates zero or more multiplicity. Here Data class is connected to System through a Composition link which is the collaboration of all participants are part of one composite class.

Attributes provide the details of the Class while Operations show all possible operations respective class can do in the system.



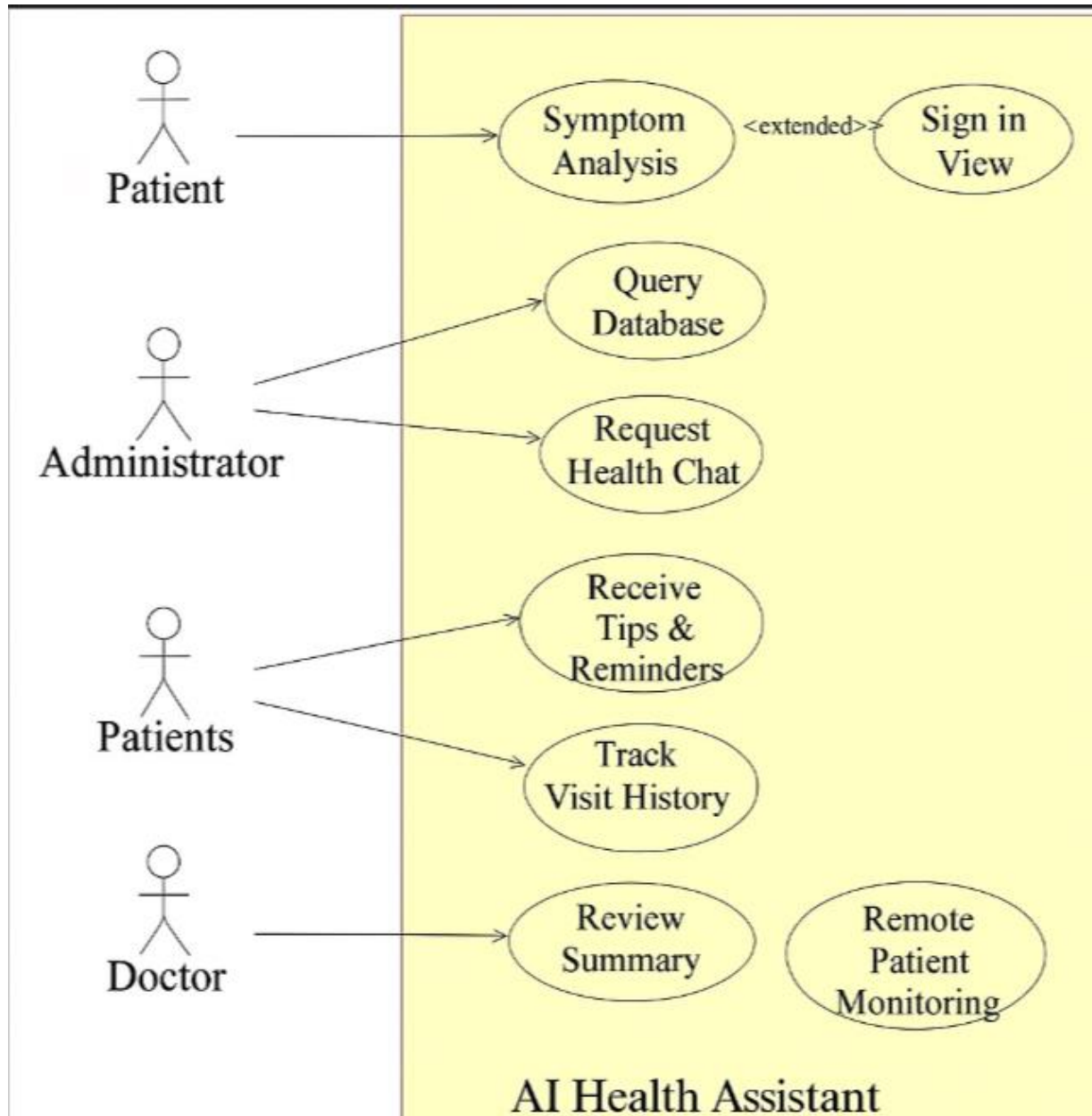
4.2 Class Diagram

4.6 System Activity

A use case diagram shows the relationship among actors and use cases within a system. Hence it provides the characteristics of the actors whose behavior and relationships can be well understood using the diagrams elaborated here.

An end-user can perform various tasks on the application; he may use signs or upload a new sign o his own. On the other hand, the system would respond to the operations done by the user. It would display the sign clips and store the uploaded sign to the defined category. Also it would provide a message whenever needed for confirmation.

Here the rectangle indicates system boundary, out of which there are actors found who perform various operations on the system which are the end-user and the system here. An elliptical shape shows the use-case while the connecting links between an actor and a use case are said to be communicates.



4.3 UseCase diagram

4.7 Sequence Diagram

A sequence diagram represents an Interaction, which is a set of messages exchanged among objects within collaboration to effect a desired operation or result. Here are the sequence diagrams for

various interactions among the user, system and the data storage. It must be noted that the rectangle box on the top of the diagram indicates the object or actor and dashed lines beneath to it shows an object's lifeline. Another rectangles following and followed by the dashed lines in a vertical manner show the activation period of the object or actor when it performs some actions. A solid arrow conveys a message while the dashed arrow gives return message. These message names are written along with their respective arrows as shown further in the diagram.

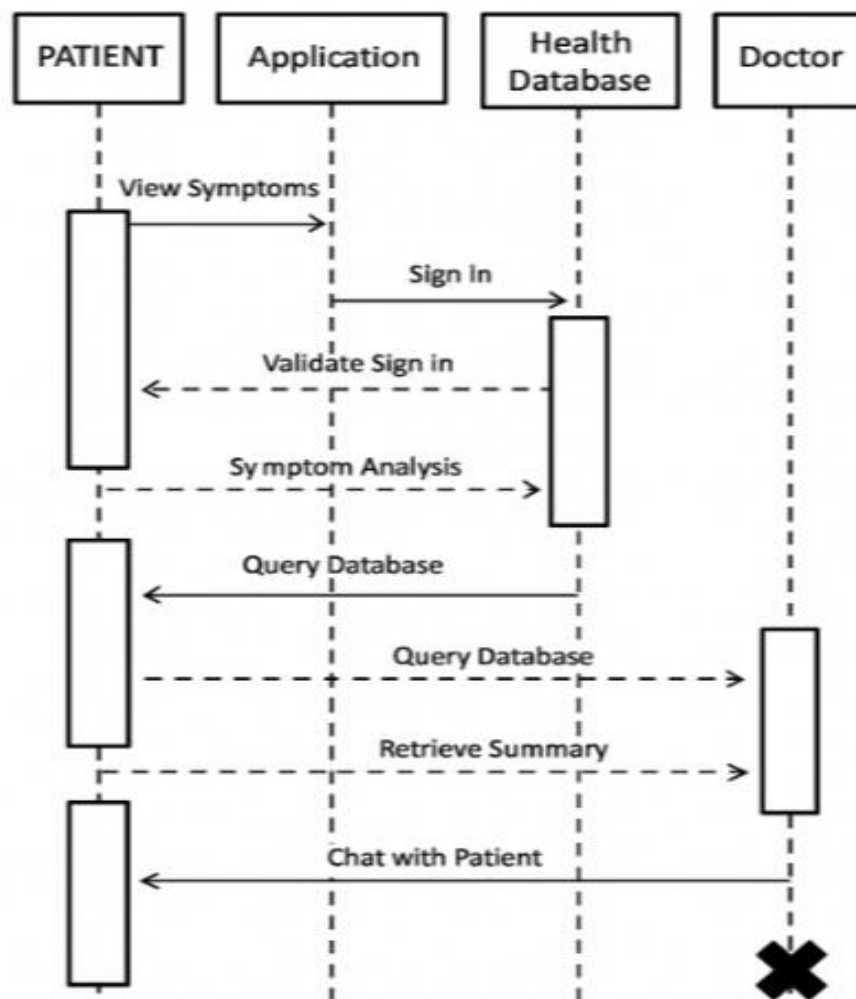


Fig-4.4 Sequence Diagram

5 SYSTEM DESIGN

- **DATABASE DESIGN**
- **ENTITY RELATIONSHIP DIAGRAM**
- **DATA FLOW DIAGRAM**
- **ACTIVITY DIAGRAM**

5.1 Database Design

The current web application uses Oracle Database to store, access and retrieve the data.

Following is the data dictionary that describes the required tables along with their fields.

5.1.1 Users:

The following table is used to store Users details

Column Name	Data type	Description
User_id	Int(Primary Key)	Unique ID for each user
Username	Varchar2(50)	User's Username
Email	Varchar2(100)	User's email address
Password	Varchar2(255)	Encrypted password
Created_at	TIMESTAMP	Account creation timestamp

Table 5.1 Users Details

5.1.2 Chathistory Details:

The following table is used to store consultant details

Field Name	Data type	Description
chat_id	Int(Primary Key)	PRIMARY KEY
User_id	INT	FOREIGN KEY
User_input	Text	NOT NULL
Bot_response	TEXT	NOT NULL

Timestamp	DATETIME	DEFAULT CURRENT_TIMESTAMP
-----------	----------	------------------------------

Table 5.2 Chathistory Details

5.1.3 Symptoms Table:

The following table is used to store agencies details:

Field Name	Data type	Constraints
Symptom_id	Int(Primary Key)	Unique ID for each agency
User_id	int	FOREIGN KEY
Symptom_list	TEXT(JSON)	NOT NULL
Submitted_at	DATETIME	DEFAULT CURRENT_TIMESTAMP

Table 5.3 Symptoms Table

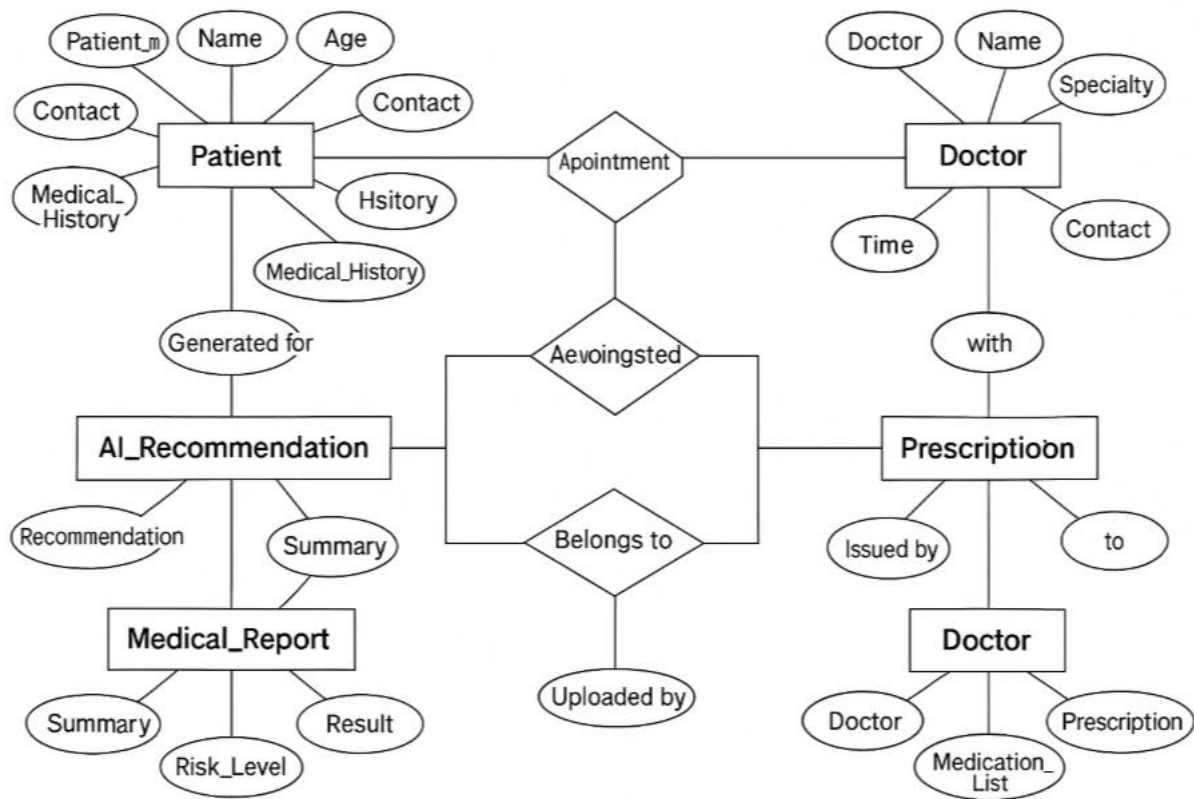
5.1.4 DiseasePrediction Table:

The following table is used to store Booking details:

Field Name	Data type	Constarints
prdition_id	Int(Primary Key)	Unique ID for each booking
User_id	Int(Foreign key)	Reference to the user making the booking (from Users)
Predicted_disease	VARCHAR(100)	NOT NULL
Confidence	FLOAT	NOT NUL
Timestamp	DATETIME	DEFAULT CURRENT_TIMESTAMP

Table 5.4 DISEASE PREDICTION TABLE

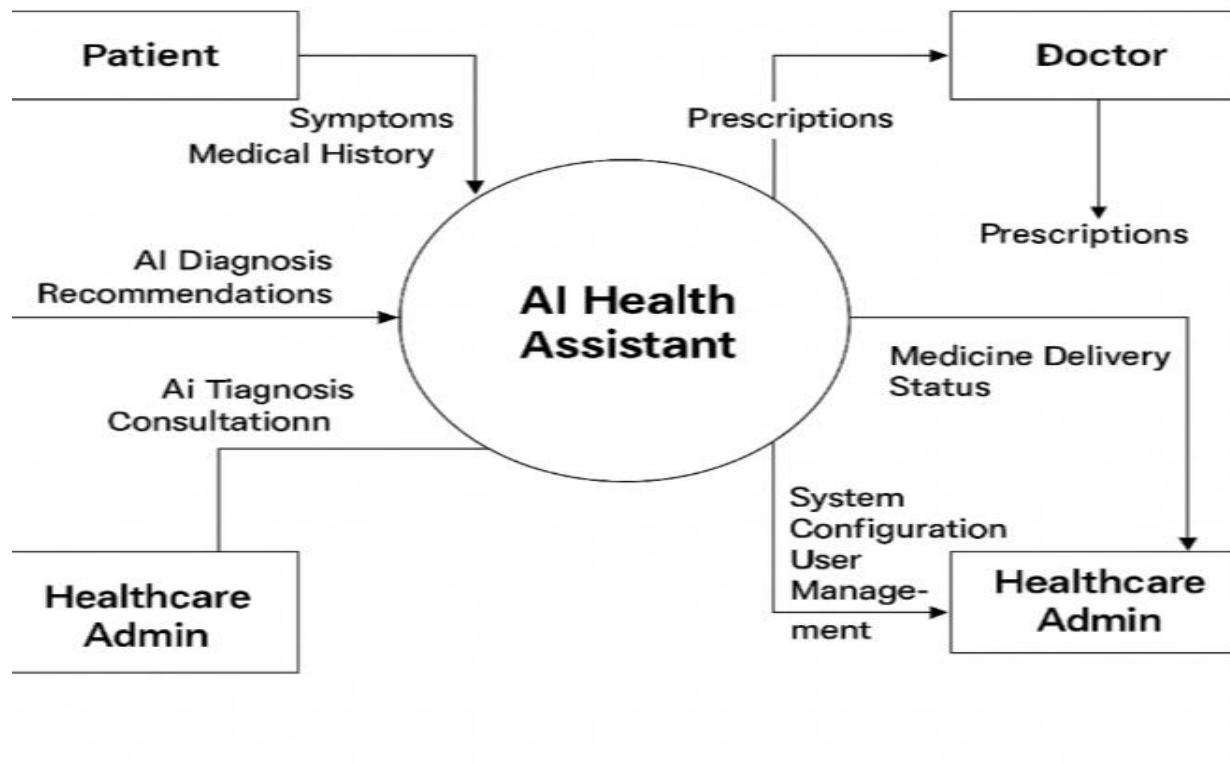
5.2 Entity Relationship Diagram



5.1 E-R Diagram

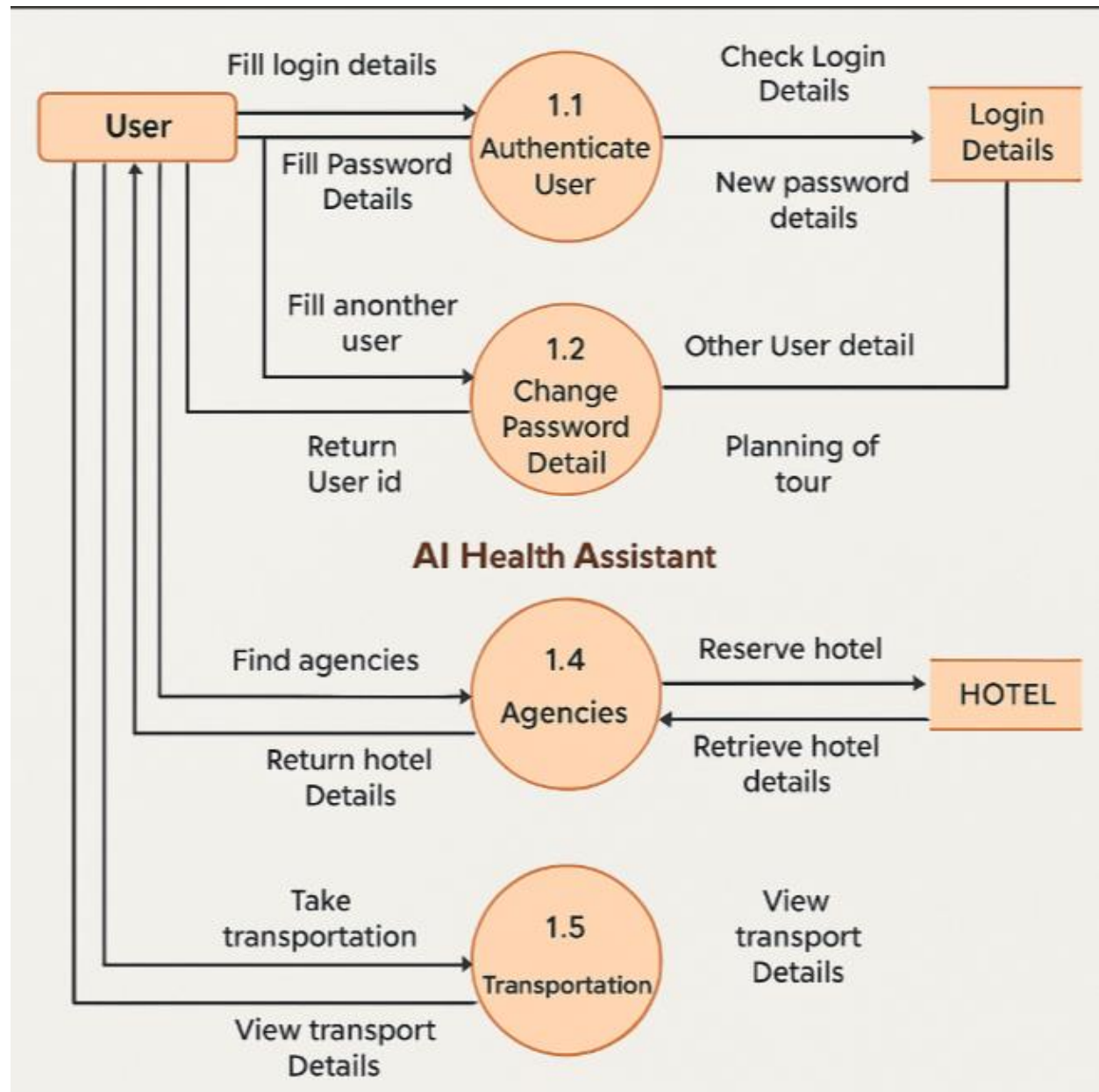
5.3 Data Flow Diagram

5.3.1 Context Level



5.2 Context Level

5.3.2 First Level DFD

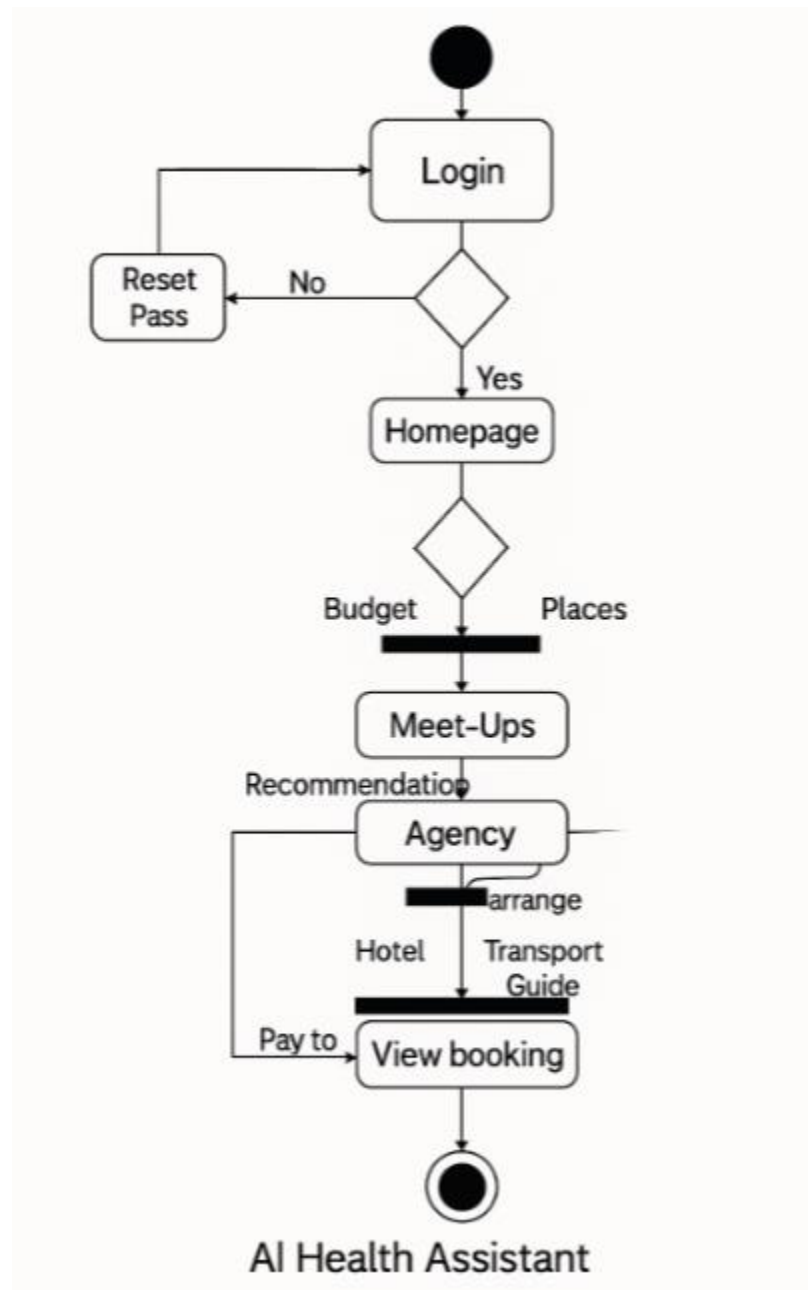


5.3 Level One DFD

5.4 Activity Diagram

An activity diagram is a special case of a state diagram in which all (or at least most) of the states are action states and in which all (or at least most) of the transitions are triggered by completion of the actions in the source states.

Below are the activity diagrams for the actions performed by the end-user and the response of the system.



6. TESTING

- **TESTING PLANS**
- **TESTING STRATEGIES**
- **TEST CASE**

6.1 Testing Plan

6.1.1 Planning Steps

- 1) Functionality Testing
- 2) Usability testing
- 3) Interface testing
- 4) Performance testing
- 5) Security testing

1) Functionality Testing:

Test for – all the links in web pages, database connection, forms used in the web pages for submitting or getting information from user, Cookie testing.

➤ **Check all the links:**

- Test the outgoing links from all the pages from specific domain under test.
- Test all internal links.
- Test links jumping on the same pages.
- Test links used to send the email to admin or other users from web pages.
- Test to check if there are any orphan pages.
- Lastly in link checking, check for broken links in all above-mentioned links.

➤ **Test forms in all pages:**

Forms are the integral part of any web site. Forms are used to get information from users and to keep interaction with them. So what should be checked on these forms?

- First check all the validations on each field.
- Check for the default values of fields.
- Wrong inputs to the fields in the forms.
- Options to create forms if any, form delete, view or modify the forms.

Let's take example of the search engine project currently I am working on, In this project we have advertiser and affiliate signup steps. Each sign up step is different but dependent on other steps. So sign up flow should get executed correctly. There are different field

validations like email Ids, User financial info validations. All these validations should get checked in manual or automated web testing.

➤ **Cookies testing:**

Cookies are small files stored on user machine. These are basically used to maintain the session mainly login sessions. Test the application by enabling or disabling the cookies in your browser options. Test if the cookies are encrypted before writing to user machine. If you are testing the session cookies (i.e. cookies expire after the sessions ends) check for login sessions and user stats after session end. Check effect on application security by deleting the cookies. (I will soon write separate article on cookie testing)

➤ **Validate your HTML/CSS:**

If you are optimizing your site for Search engines then HTML/CSS validation is very important. Mainly validate the site for HTML syntax errors. Check if site is crawlable to different search engines.

➤ **Database testing:**

Data consistency is very important in web application. Check for data integrity and errors while you edit, delete, modify the forms or do any DB related functionality. Check if all the database queries are executing correctly, data is retrieved correctly and also updated correctly. More on database testing could be load on DB, we will address this in web load or performance testing below.

2) Usability Testing:

➤ **Test for navigation:**

Navigation means how the user surfs the web pages, different controls like buttons, boxes or how user using the links on the pages to surf different pages. Usability testing includes: Web site should be easy to use. Instructions should be provided clearly. Check if the provided instructions are correct means whether they satisfy purpose. Main menu should be provided on each page. It should be consistent.

➤ **Content:**

Content should be logical and easy to understand. Check for spelling errors. Use of dark colours annoys users and should not be used in site theme. You can follow some standards that are used for web page and content building. These are common accepted standards like as I mentioned above about annoying colors, fonts, frames etc. Content should be meaningful. All the anchor text links should be working properly. Images should be placed properly with proper sizes. These are some basic standards that should be followed in web development. Your task is to validate all for UI testing.

➤ **Other user information for user help:**

Like search option, sitemap, help files etc. Sitemap should be present with all the links in web sites with proper tree view of navigation. Check for all links on the sitemap. “Search in the site” option will help users to find content pages they are looking for easily and quickly. These are all optional items and if present should be validated.

3) Interface Testing:

The main interfaces are:

- Web server and application server interface
- Application server and Database server interface.

Check if all the interactions between these servers are executed properly. Errors are handled properly. If database or web server returns any error message for any query by application server then application server should catch and display these error messages appropriately to users. Check what happens if user interrupts any transaction in-between? Check what happens if connection to web server is reset in between?

4) Performance Testing

Web application should sustain to heavy load. Web performance testing should include:

- Web Load Testing
- Web Stress Testing

Test application performance on different internet connection speed. In **web load testing** test if many users are accessing or requesting the same page. Can system sustain in peak load times? Site should handle many simultaneous user requests, large input data from users, Simultaneous connection to DB, heavy load on specific pages etc.

Stress testing: Generally stress means stretching the system beyond its specification limits. Web stress testing is performed to break the site by giving stress and checked how system reacts to stress and how system recovers from crashes. Stress is generally given on input fields, login and sign up areas.

In web performance testing web site functionality on different operating systems, different hardware platforms is checked for software, hardware memory leakage errors.

5) Security Testing:

Following are some test cases for web security testing:

- Test by pasting internal url directly into browser address bar without login. Internal pages should not open.
- If you are logged in using username and password and browsing internal pages then try changing url options directly. I.e. If you are checking some publisher site statistics with publisher site ID= 123. Try directly changing the url site ID parameter to different site ID which is not related to logged in user. Access should be denied for this user to view others stats.
- Try some invalid inputs in input fields like login username, password, input text boxes. Check the system reaction on all invalid inputs.
- Web directories or files should not be accessible directly unless given download option.
- Test if SSL is used for security measures. If used proper message should get displayed when user switch from non-secure http:// pages to secure https:// pages and vice versa.
- All transactions, error messages, security breach attempts should get logged in log files somewhere on web server.

6.2 Testing Strategies

- **White Box Testing:**

White box testing (WBT) is also called **Structural or Glass box testing**. White box testing involves looking at the structure of the code. When you know the internal structure of a product, tests can be conducted to ensure that the internal operations performed according to the specification. And all internal components have been adequately exercised.

- **Why we do White Box Testing?**

To ensure:

- That all independent paths within a module have been exercised at least once.
- All logical decisions verified on their true and false values.
- All loops executed at their boundaries and within their operational bounds internal data structures validity.

- **Need of White Box Testing?**

To discover the following types of bugs:

- Logical error tend to creep into our work when we design and implement functions, conditions or controls that are out of the program
- The design errors due to difference between logical flow of the program and the actual implementation
- Typographical errors and syntax checking

Limitation Of WBT:

Not possible for testing each and every path of the loops in program. This means exhaustive testing is impossible for large systems. This does not mean that WBT is not effective. By selecting important logical paths and data structure for testing is practically possible and effective.

- **Black Box Testing:**

- Black box testing treats the system as a “**black-box**”, so it doesn’t explicitly use Knowledge of the internal structure or code. Or in other words the Test engineer need not know the internal working of the “Black box” or application.
- **Main focus in black box testing is on functionality of the system as a whole.** The term ‘**behavioural testing**’ is also used for black box testing and white box testing is also sometimes called ‘**structural testing**’. Behavioural test design is slightly different from black-box test design because the use of internal knowledge isn’t strictly forbidden, but it’s still discouraged.
- Each testing method has its own advantages and disadvantages. There are some bugs that cannot be found using only black box or only white box. Majority of the application are tested by black box testing method. We need to cover majority of test cases so that most of the bugs will get discovered by black box testing.
- Black box testing occurs throughout the software development and Testing life cycle i.e. in Unit, Integration, System, Acceptance and regression testing stages.
- **Advantages of Black Box Testing**
 - Tester can be non-technical.
 - Used to verify contradictions in actual system and the specifications.
 - Test cases can be designed as soon as the functional specifications are complete
- **Disadvantages of Black Box Testing**
 - The test inputs needs to be from large sample space.
 - It is difficult to identify all possible inputs in limited testing time. So writing test cases is slow and difficult. Chances of having unidentified paths during this testing.

6.3 TEST CASE

The screenshot displays the 'AI Health Assistant - Advanced Edition' interface. On the left is a dark sidebar with a 'Navigation' menu containing: 'Health Assistant Menu', 'Diabetes Prediction' (highlighted), 'Heart Disease Prediction', 'Parkinsons Prediction', 'Prediction History', 'Trend Analysis', and 'Export Data'. The main content area has a top header 'AI Health Assistant - Advanced Edition' and a title 'Diabetes Prediction using Machine Learning'. Below the title is a dropdown menu 'About Diabetes Prediction'. The 'Enter Your Health Parameters' section contains input fields for: 'Number of Pregnancies' (0), 'Glucose Level (mg/dL)' (120.00), 'Blood Pressure (mmHg)' (80.00), 'Skin Thickness (mm)' (20.00), 'Insulin Level (uU/mL)' (80.00), 'BMI (kg/m²)' (25.00), 'Diabetes Pedigree Function' (0.50), and 'Age (years)' (30). Each field has minus and plus icons for adjustment. At the bottom center is a 'Run Diabetes Test' button, and at the bottom right is a 'Manage app' link.

Parameter	Value
Number of Pregnancies	0
Glucose Level (mg/dL)	120.00
Blood Pressure (mmHg)	80.00
Skin Thickness (mm)	20.00
Insulin Level (uU/mL)	80.00
BMI (kg/m²)	25.00
Diabetes Pedigree Function	0.50
Age (years)	30

Navigation

Health Assistant Menu

Diabetes Prediction

Heart Disease Prediction

Parkinsons Prediction

Prediction History

Trend Analysis

Export Data

Share ★ ↗ 🔊 ⋮

AI Health Assistant - Advanced Edition

Heart Disease Prediction using Machine Learning

About Heart Disease Prediction

Enter Your Cardiovascular Parameters

👤 Age (years)

50 - +

👤 Sex

Female ▾

👤 Chest Pain Type

Typical Angina ▾

👤 Resting Blood Pressure (mmHg)

120.00 - +

👤 Serum Cholesterol (mg/dl)

200.00 - +

👤 Fasting Blood Sugar > 120 mg/dl

No ▾

👤 Resting ECG Results

Normal ▾

👤 Max Heart Rate Achieved

150.00 - +

👤 Exercise Induced Angina

No ▾

👤 ST Depression

1.00 - +

👤 ST Segment Slope

Upsloping ▾

👤 Major Vessels (0-3)

0 ▾

👤 Thalassemia

Normal ▾

Run Heart Disease Test

Manage app

AI Health Assistant - Advanced Edition

Share
★
🔍
⋮

Navigation

Health Assistant Menu

Diabetes Prediction

Heart Disease Prediction

Parkinsons Prediction

Prediction History

Trend Analysis

Export Data

Parkinson's Disease Prediction using Machine Learning

About Parkinson's Disease Prediction

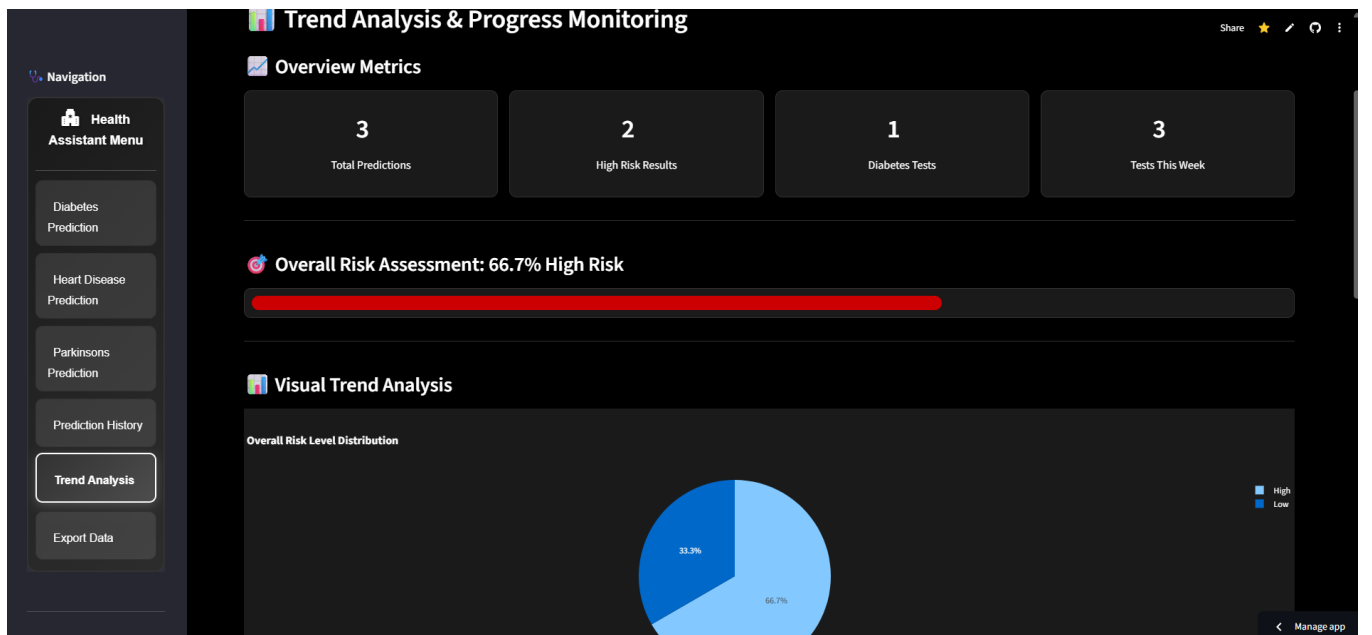
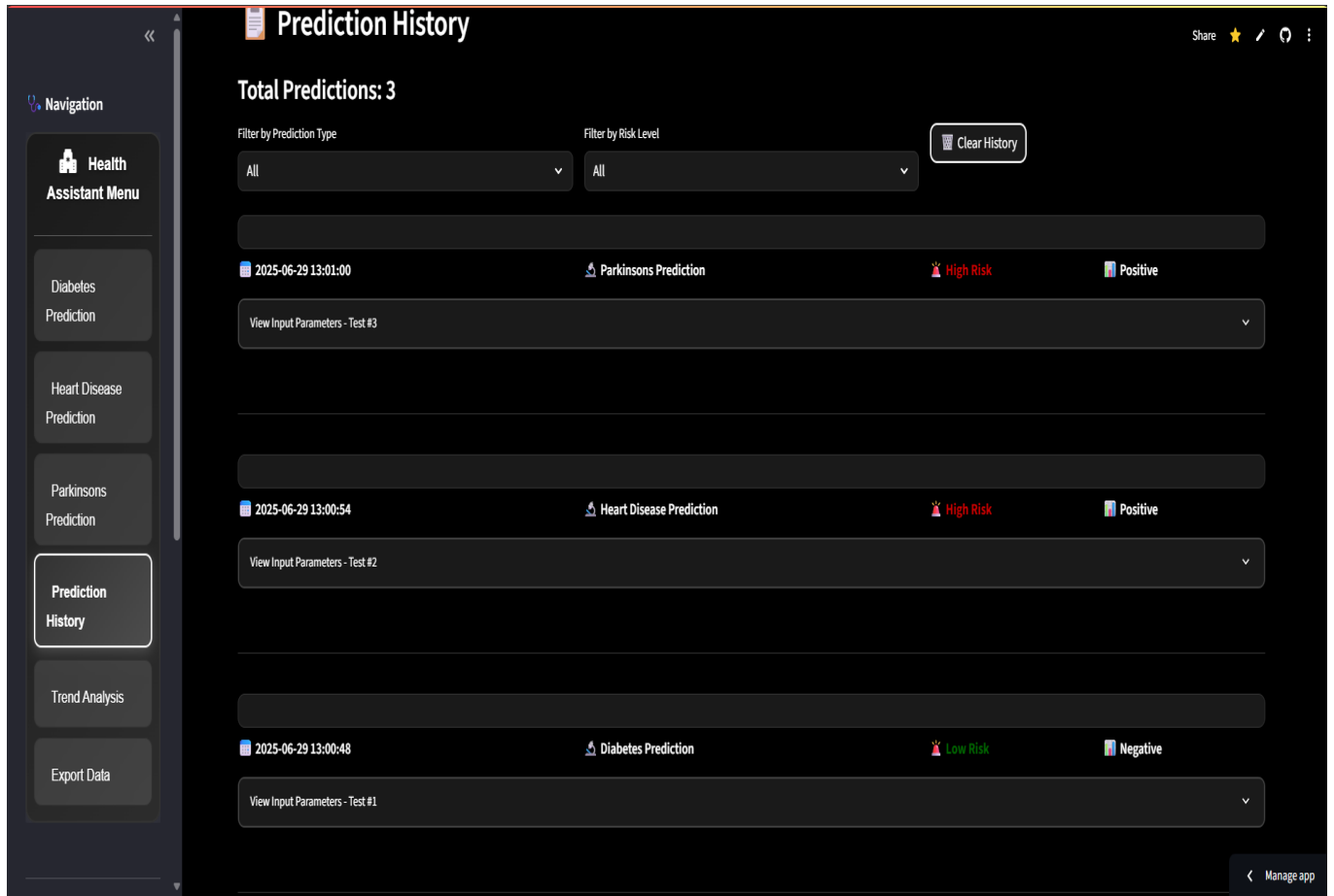
Enter Voice Analysis Parameters

These parameters are typically obtained from voice recording analysis. Consult with a healthcare professional for proper voice analysis.

MDVP:F0(Hz)	MDVP:F1(Hz)	MDVP:F2(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)
150.00	200.00	100.00	0.50	0.000100
MDVP:RAP	MDVP:PPQ	Jitter:DDP	Shimmer:APQ5	D2
0.0100	0.0100	0.0300	0.0200	2.00
MDVP:Shimmer	MDVP:Shimmer(dB)	Shimmer:APQ3	HNR	PPE
0.0300	0.30	0.0200	20.00	0.20
MDVP:APQ	Shimmer:DDA	NHR	spread2	
0.0200	0.0500	0.0200	0.20	
RPDE	DFA	spread1		
0.50	0.70	-5.00		

Run Parkinson's Test

Manage app



7 LIMITATIONS AND FUTURE ENHANCEMENT

- **LIMITATIONS**

- **FUTURE ENHANCEMENT**

7.1 Limitations

The app faces several limitations that could impact its overall user experience and scalability. It relies heavily on stable internet connectivity, limiting its functionality in areas with poor or no network access, while offering limited offline capabilities. Scalability could also become a concern as the app grows, potentially leading to performance slowdowns without a robust backend. Additionally, handling sensitive user data raises privacy and security risks, which could pose challenges in ensuring compliance with data protection regulations. The app may struggle to gain traction due to competition from established players, and inconsistent service quality from third-party providers like agencies and guides could further affect user satisfaction. Real-time updates may experience delays, leading to issues with bookings or itineraries. The lack of language and cultural support could alienate non-English-speaking users, and limited service coverage in remote areas reduces the app's effectiveness. Complex booking management might overwhelm users, while high maintenance costs for updates, partnerships, and security measures could strain resources. Finally, the app may offer limited customization options, restricting users from personalizing their travel experiences.

7.2 Future Enhancement

Future enhancements for the app could greatly improve its functionality and user experience. Implementing an offline mode would allow users to access important features like itineraries and maps even in areas with poor connectivity. Adding personalized recommendations driven by AI can offer tailored meet-ups, tours, and accommodations based on user preferences. Expanding multi-language support and including in-app chat for seamless communication with organizers and agencies will enhance the app's global appeal. Enhanced security features like two-factor authentication and end-to-end encryption would ensure safer transactions and data protection.

Further improvements could include augmented reality (AR) for interactive travel experiences, and a loyalty program to encourage repeat bookings. Providing real-time notifications for booking updates and customizable itinerary planners would enhance convenience for travelers. A user review and rating system would help users make informed choices, while expanding global service partnerships would increase coverage in remote areas. Customizable notifications and dynamic pricing could be added to offer a more personalized experience and competitive pricing. Lastly, voice assistant integration and local event/activity suggestions would improve user convenience and help discover unique experiences during their trips.

8. BIBLIOGRAPHY

- **CONCLUSION**
- **BIBLIOGRAPHY**

8.1 Conclusion

The AI Health Assistant successfully integrates machine learning algorithms, natural language processing, and health data analytics to provide preliminary diagnostics, symptom checking, and personalized healthcare recommendations. The system is designed to reduce the burden on healthcare professionals by enabling users to access basic medical advice and support remotely. With real-time data analysis and chatbot interaction, the assistant enhances patient engagement and promotes preventive care. Future improvements may include integration with wearable devices, multi-language support, and a more robust knowledge base to improve accuracy and reliability.

The AI Health Assistant makes healthcare more accessible and user-friendly by helping people check symptoms, get basic health advice, and understand when to seek medical help. It acts like a smart guide for users, especially in remote areas, offering support anytime through chat and automated tools. As technology improves, this assistant can become even more helpful by connecting with smart devices and offering more personalized care.

8.2 BibliographyS

Web Resources

- 1) <https://www.who.int/publications/i/item/9789240029200>
- 2) <https://cloud.google.com/healthcare>
- 3) <https://azure.microsoft.com/en-us/blog/using-ai-to-build-healthcare-chatbots/>
- 4) <https://www.ibm.com/watson-health>
- 5) <https://huggingface.co/docs/transformers>

Books

- **Topol, Eric.** (2019). *Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again*. Basic Books.
ISBN: 9781541644632
- **Rajkomar, Alvin, Dean, Jeff, & Kohane, Isaac.** (2021). *Machine Learning in Medicine*. Springer.
ISBN: 9783030812144
- **Yu, Kun-Hsing & Kohane, Isaac S.** (2019). *Artificial Intelligence in Healthcare*. Academic Press.
ISBN: 9780128173576
- **Kudyba, Stephan.** (2020). *Healthcare Informatics: Improving Efficiency and Productivity*. CRC Press.
ISBN: 9780367333583
- **Davenport, Thomas H., & Kalakota, Ravi.** (2019). *The Potential for Artificial Intelligence in Healthcare*. Future Healthcare Journal, Royal College of Physicians. (Also in print collections)
- **Zhang, Yu, Milinovich, Andrew, & Xu, Hua.** (2021). *Natural Language Processing in Healthcare and Biomedicine: A Practical Guide*. Springer.
ISBN: 9783030727615
- **Shortliffe, Edward H., & Cimino, James J.** (2014). *Biomedical Informatics: Computer Applications in Health Care and Biomedicine*. Springer.
ISBN: 9781447144748