

Web-Based Health Diagnosis System

A PROJECT REPORT

Submitted by

P.V.Hemachandra [RA2211003011040]
N.Rohan Teja [RA2211003011052]

Under the Guidance of
Dr.K.Anitha
Associate Professor
Department of Computing Technologies

in partial fulfillment of the requirements for the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE ENGINEERING



SRM

INSTITUTE OF SCIENCE & TECHNOLOGY

Deemed to be University u/s 3 of UGC Act, 1956

**DEPARTMENT OF COMPUTING TECHNOLOGIES
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR- 603 203**

APRIL 2025



Department of Computational Intelligence
SRM Institute of Science & Technology
Own Work* Declaration Form

This sheet must be filled in (each box ticked to show that the condition has been met). It must be signed and dated along with your student registration number and included with all assignments you submit – work will not be marked unless this is done.

To be completed by the student for all assessments

Degree/ Course : **B-TECH/ CSE (CORE)**

Student Name : **P.V.Hemachandra, N.Rohan Teja**

Registration Number : **RA2211003011040, RA2211003011052**

Title of Work : **Web-Based Health Diagnosis System**

I / We hereby certify that this assessment complies with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly referenced / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook /University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.

If you are working in a group, please write your registration numbers and sign with the date for every student in your group.



SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of Soc. Act, 1963)

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that 21CSP302L - Minor Project report titled “**Web-Based Health Diagnosis System**” is the bonafide work of “**P.V.Hemachandra [RA2211003011040], N.Rohan Teja [RA2211003011052]**” who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr K.ANITHA

SUPERVISOR

ASSOCIATE PROFESSOR

DEPARTMENT OF

COMPUTING TECHNOLOGIES

SIGNATURE

Dr G. NIRANJANA

PROFESSOR &HEAD

DEPARTMENT OF

COMPUTING TECHNOLOGIES

ACKNOWLEDGEMENTS

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to **Dr. Leenus Jesu Martin M**, Dean-CET, SRM Institute of Science and Technology, for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor and Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We encompass our sincere thanks to, **Dr. M. Pushpalatha**, Professor and Associate Chairperson - CS, School of Computing and **Dr. Lakshmi**, Professor and Associate Chairperson -AI, School of Computing, SRM Institute of Science and Technology, for their invaluable support.

We are incredibly grateful to our Head of the Department, **Dr G. Niranjana**, Professor and Head, Department of Computing Technologies, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Project Coordinators, Panel Head, and Panel Members Department of Computing Technologies, SRM Institute of Science and Technology, for their inputs during the project reviews and support. We register our immeasurable thanks to our Faculty Advisor, **Dr M.Kandan**, Department of Computing Technologies, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide, **Dr K.Anitha**, Department of Computing Technologies, SRM Institute of Science and Technology, for providing us with an opportunity to pursue our project under her mentorship. She provided us with the freedom and support to explore the research topics of our interest. Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank all the staff members of Computing Technologies, School of Computing, S.R.M Institute of Science and Technology, for their help during our project. Finally, we would like to thank our parents, family members, and friends for their unconditional love, constant support and encouragement

P.V.Hemachandra
N.Rohan Teja

ABSTRACT

In the contemporary era of digital transformation, the healthcare sector is increasingly embracing intelligent technologies to augment diagnostic accuracy and operational efficiency. One critical challenge in early-stage disease detection is the lack of adaptive and reliable diagnostic tools that can dynamically interpret both structured and unstructured patient data. Traditional symptom checkers often operate on rigid logic or manually coded rules, which limits their adaptability, personalization, and precision. Recognizing these limitations, our project introduces an advanced Intelligent Disease Prediction System (IDPS) aimed at improving preliminary diagnostics using a hybrid of semantic reasoning and machine learning algorithms. This intelligent system is specifically designed to analyze patient health data, draw meaningful inferences, and assist healthcare administrators in making better-informed decisions. The backend leverages a semantic knowledge base the Human Disease Diagnosis Ontology (HDDO) to reason through symptom-disease relationships and uncover latent associations that are often missed in rule-based systems. HDDO enables a structured, hierarchical representation of medical knowledge, enhancing the contextual understanding of symptoms and disease manifestations. To address the challenges associated with unstructured health data, particularly free-text patient inputs or image-based inputs, the system integrates a Convolutional Neural Network (CNN). The CNN component is trained to extract deep patterns and features that contribute to more nuanced and accurate predictions. This dual approach combining semantic reasoning and deep learning allows the system to tackle a wide variety of diagnostic scenarios with increased accuracy and confidence. The system is currently implemented with admin-only access using a web-based interface developed in Java (NetBeans IDE), with MySQL databases managed via SQLyog. The admin has the authority to manage patient records, monitor model outputs, perform data preprocessing, and generate diagnostic reports. Symptom Mapping which Automatically maps patient symptoms to probable diseases using the HDDO framework.

TABLE OF CONTENTS

ABSTRACT	v	
TABLE OF CONTENTS	vi,vii	
LIST OF FIGURES	viii, ix	
LIST OF TABLES	x	
ABBREVIATIONS	xi	
CHAPTER NO.	TITLE	PAGE NO.
1 INTRODUCTION		12
1.1 Introduction to Project		12
1.2 Motivation		13
1.3 Sustainable Development Goal of the Project		13
1.4 Product Vision Statement		14
1.5 Product Goal		15
1.6 Literature Survey		15
1.7 Product Backlog (Key User Stories with Desired Outcomes)		16
1.8 Product Release Plan		17
2 SPRINT PLANNING AND EXECUTION		18
2.1 Sprint 1		18
2.1.1 Sprint Goal with User Stories of Sprint 1		18-20
2.1.2 Functional Document		21-23
2.1.3 Architecture Document		24-29
2.1.4 UI Design		30
2.1.5 Functional Test Cases		31
2.1.6 Daily Call Progress		31
2.1.7 Committed vs Completed User Stories		32
2.1.8 Sprint Retrospective		33

2.2 Sprint 2	33
2.2.1 Sprint Goal with User Stories of Sprint 2	33-35
2.2.2 Functional Document	36,37
2.2.3 Architecture Document	38,39
2.2.4 UI Design	40
2.2.5 Functional Test Cases	41
2.2.6 Daily Call Progress	41
2.2.7 Committed vs Completed User Stories	42
2.2.8 Sprint Retrospective	42
2.3 Sprint 3	43
2.3.1 Sprint Goal with User Stories of Sprint 2	43-45
2.3.2 Functional Document	46,47
2.3.3 Architecture Document	47,48
2.3.4 UI Design	49
2.3.5 Functional Test Cases	50
2.3.6 Daily Call Progress	50
2.3.7 Committed vs Completed User Stories	51
2.3.8 Sprint Retrospective	51
3. RESULTS AND DISCUSSIONS	52
3.1 Project Outcomes (Justification of outcomes and how they align with the goals)	52,53
3.2 Committed vs Completed User Stories	53
4 CONCLUSION & FUTURE ENHANCEMENT	54
REFERENCES	55
APPENDIX	56
A. SAMPLE CODING	56-58
B. PLAGIARISM REPORT	59

LIST OF FIGURES

CHAPTER NO	TITLE	PAGE NO.
1.1	MS Planner Board of Web-Based Health Diagnosis System.....	17
1.2	Release plan of Web-Based Health Diagnosis System.....	18
2.1	image of User-story for Admin Access.....	19
2.2	User Story viewing Patients Records.....	20
2.3	System Architecture Diagram	21
2.4	Entity-Relationship (ER) Diagram	25
2.5	Schema Design	25
2.6	Use-Case Diagram.....	26
2.7	Class Diagram.....	26
2.8	Communication diagram.....	27
2.9	Deployment Diagram.....	27
3.0	Data Flow Diagram (Level 0 and 1).....	28
3.1	UI design for Landing Page.....	30
3.2	UI Design for Login Page.....	31
3.3	Standup Meetings.....	32
3.4	Bar graph for Committed Vs Completed User Stories.....	33
3.5	Sprint Retrospective for the Sprint 1.....	34
3.6	User Story for Generating Diagnosis Report.....	35
3.7	User story for Updating Patient Records.....	35
3.8	Architecture Diagram for Sprint 2.....	38
3.9	UI design for Viewing, maintaining and Updating the dataset.....	40
4.0	UI design for generating the Report.....	40
4.1	Standup Meeting.....	41
4.2	Bar graph for Committed Vs Completed User Stories.....	42
4.3	Sprint Retrospective for the Sprint 2.....	43
4.4	User Story for Data Security and Privacy.....	44
4.5	user story for Checking System Performance Monitoring.....	45
4.6	Updated Architecture Diagram for Sprint 3.....	48

4.7	I design for Overall Report.....	49
4.8	UI design for generating Prescription.....	49
4.9	standup Meeting.....	50
5.0	Bar graph for Committed Vs Completed User Story.....	51
5.1	Sprint Retrospective for sprint 3.....	51
5.2	Bar graph for Committed Vs Completed User Stories.....	53

LIST OF TABLES

CHAPTER NO	TITLE	PAGE NO.
1.1	User Stories Table	16
2.1	Detailed User Stories of sprint 1.....	18
2.2	Access level Authorization Matrix.....	23
2.3	Detailed Functional Test Case.....	31
2.4	Detailed User Stories of Sprint 2.....	33
2.5	Access Level Authorization Matrix.....	37
2.6	Detailed Functional Test Case.....	41
2.7	User Stories for Sprint 3.....	43
2.8	Access level Authorization Matrix.....	47
2.9	Detailed Functional Test Case.....	50

ABBREVIATIONS

ABBREVIATIONS	Full Form
AI	Artificial Intelligence
CNN	Convolutional Neural Network
EHR	Electronic Health Record
GUI	Graphical User Interface
HDDO	Human Disease Diagnosis Ontology
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
JDBC	Java Database Connectivity
JVM	Java Virtual Machine
ML	Machine Learning
OOP	Object-Oriented Programming
PHR	Personal Health Record
SQL	Structured Query Language
UI	User Interface

CHAPTER 1

INTRODUCTION

1.1 Introduction

As the field continues to grow and transform at a fast pace of digital health, there is a growing reliance on technology to aid in early disease detection and self-assessment. With the rise of mobile health apps, wearable devices, and AI-driven tools, patients are becoming more proactive in managing their well-being. However, this shift also brings challenges — especially when users depend on generic search engines or low-accuracy symptom checkers for medical insights. These tools often lack contextual understanding, leading to misinformation and unnecessary panic.

Moreover, the explosion of health-related data from electronic health records (EHRs), wearable sensors, and patient-reported outcomes has created an opportunity for data-driven diagnosis systems that go beyond static symptom matching. By leveraging advanced AI techniques and semantic models, we can connect unprocessed data with actionable insights meaningful medical insights. These intelligent systems offer the potential to identify hidden patterns in complex datasets, adapt to individual patient profiles, and provide accurate recommendations tailored to real-world medical scenarios.

In today's e-health era, individuals frequently use the internet to self-diagnose based on symptoms. However, search engines and basic symptom checkers often fail to account for a person's complete health profile, leading to misinterpretations and anxiety — a condition known as cyberchondria. These tools typically retrieve high-ranking results, not necessarily accurate or personalized medical advice.

The accuracy of existing online symptom checkers is alarmingly low. Studies show that they produce correct diagnoses in only about 34% of cases. This lack of reliability undermines public trust and highlights the urgent need for more precise, context-aware diagnosis tools. To address this issue, this project proposes an intelligent disease prediction system that integrates structured and unstructured health data. Admins can upload patient data, and the system uses an ontology-driven approach enhanced with machine learning (e.g., CNN) to suggest possible conditions.

1.2 Motivation

The motivation for this project originates from the growing public reliance on web-based symptom checkers and their consistent failure to deliver accurate diagnoses. While convenient, most tools lack context-awareness and are unable to handle the diversity in patient health conditions, leading to incorrect or generalized results.

Furthermore, there is a vast volume of medical data being generated in hospitals and health portals, yet much of it remains underutilized. By incorporating both structured (e.g., symptoms, demographics) and unstructured (e.g., patient notes) data, this project aims to create a tool that bridges the gap between data availability and meaningful diagnosis.

Another strong motivator is the need for administrative tools in hospital environments that allow controlled access to sensitive medical data. Our system is currently designed for **admin-only use**, enabling authorized personnel to upload, view, and analyze patient records securely. This ensures both privacy and centralized control while allowing the system to grow and improve with real-world data input.

1.3 Sustainable Development Goal of the Project

This project supports the vision of the United Nation's Sustainable Development Goal 3 (SDG 3), aiming to enhance public health and encourage overall well-being for people of all ages. Our intelligent health diagnosis system contributes to this goal by improving access to reliable health information, enabling early detection of diseases, and reducing misdiagnoses caused by generic symptom checkers.

The system's focus on accurate and data-driven diagnosis promotes **preventive care**, which is a critical component of public health sustainability. By leveraging patient health records and symptom mapping using ontology and AI, the platform supports clinical decision-making and contributes to better patient outcomes.

Moreover, the project supports **universal healthcare access** in digital form by laying the foundation for a scalable tool that can be deployed across hospitals, telemedicine platforms, and even rural healthcare networks in the future.

1.4 Product Vision Statement

Audience:

- **Primary Audience:** Healthcare administrators and analysts seeking to manage and analyze patient data.
- **Secondary Audience:** Doctors, clinical researchers, and future health app users.

Needs:

- Reliable disease prediction based on patient-specific data.
- Accurate diagnosis support using structured and unstructured health data.
- Secure admin-level data access and control.
- Scalable integration with future patient-facing platforms.

Product:

- Core Product: An intelligent, admin-controlled health diagnosis tool using ontology models and CNN for disease prediction.
- Features:
 - Dataset upload and management
 - Preprocessing and symptom mapping
 - Search and diagnosis analysis
 - Diagnostic result logging
 - Secure access for authorized personnel

Values:

- Accuracy: Uses structured medical ontology (HDDO) for reliable predictions.
- Security: Admin-based access control to maintain data privacy.
- Scalability: Designed for integration into broader hospital systems or public health networks.

1.5 Product Goal

The primary goal of this project is to create a **trustworthy, intelligent disease diagnosis system** that improves on existing symptom checkers by integrating advanced machine learning and medical ontology. It aims to provide **high-accuracy, context-aware diagnostic suggestions** based on real patient data.

The system focuses on:

- Supporting healthcare professionals with reliable predictions
- Assisting admins with centralized control of health data
- Enhancing diagnosis accuracy using CNN-based learning models
- Enabling detailed progress tracking of patient diagnostics

1.6 Literature Survey

Aspects	Description
Technological Progress in Healthcare	Technological advancements have enabled the creation of digital tools like symptom checkers to aid users in understanding potential health issues.
Core Challenge	Despite being widely used, these tools often exhibit low diagnostic precision due to several underlying constraints.
Manual Knowledge Base Dependency	Many platforms rely on static, manually curated medical knowledge bases, which are time-consuming to update and vulnerable to errors or outdated content.
Inaccuracy in Diagnosis	Over-reliance on outdated or overly simplistic models can lead to incorrect health predictions.
Oversimplified Diagnostic Logic	Conventional symptom checkers often analyze basic symptom-disease links, neglecting deeper personalized inputs or complex interrelations.
Lack of Individualization	Many systems overlook individual health backgrounds and demographic data, limiting the accuracy of personalized assessments.

1.7 Product Backlog

Table 1.1 UserStories

S.No	User Stories
#US 1	<p>This user story targets the system admin responsible for managing patient health records.</p> <p>The admin wants to log in securely to the web application to access and manage patient records.</p>
#US 2	<p>This user story targets the system admin responsible for retrieving patient information.</p> <p>The admin wants to search for a patient using their unique patient ID.</p> <p>The main benefit is that the admin can quickly access patient health records for review and diagnosis.</p>
#US 3	<p>This user story targets the system admin responsible for generating reports based on patient symptoms.</p> <p>The admin wants to generate a diagnostic report based on a patient's symptoms and medical history.</p> <p>The main benefit is that the system can provide an accurate diagnosis to assist in patient treatment.</p>
#US 4	<p>This user story targets the system admin responsible for keeping medical records up to date.</p> <p>The admin wants to update a patient's health records with new test results and diagnoses.</p> <p>The main benefit is that patient records will be kept accurate for future reference and treatment.</p>
#US 5	<p>This user story targets the system admin to ensure compliance with data security regulations.</p> <p>The admin wants to ensure that only authorized personnel can access patient records.</p> <p>The main benefit is that patient privacy is maintained, preventing unauthorized access.</p>
#US 6	<p>This user story targets the system admin responsible for ensuring system reliability.</p> <p>The admin wants to monitor the system's performance and detect errors.</p> <p>The main benefit is that issues can be resolved quickly to ensure smooth system operation.</p>

The product backlog for the Web-Based Health Diagnosis System was organized using Microsoft Planner's Agile Board, as illustrated in Figure 1.1. It contains all the user stories relevant to the project. Each story outlines key functional and non-functional requirements, along with clearly defined acceptance criteria and associated tasks.

Figure 1.1 MS Planner Board of Web-Based Health Diagnosis System

1.8 Product Release Plan I

The release timeline for the project is illustrated in Figure 1.2

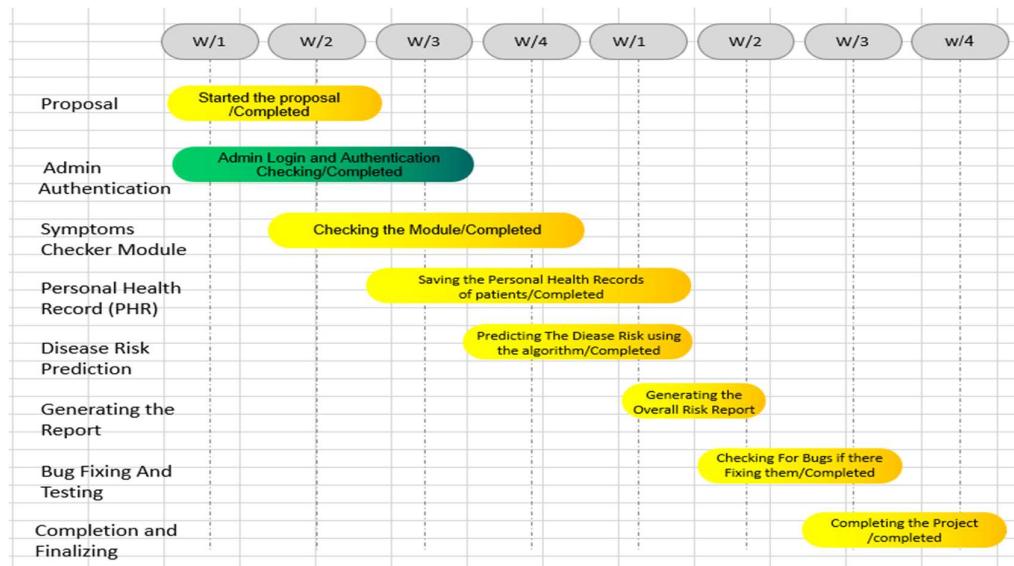


Figure 1.2 Release plan of Web-Based Health Diagnosis System

CHAPTER 2

SPRINT PLANNING AND EXECUTION

2.1 Sprint 1

2.1.1 Sprint Goal with User Stories of Sprint 1

The primary objective of Sprint 1 is to develop the Admin landing page and implement key search functionalities, including viewing datasets, performing preprocessing, and generating reports. Table 2.1 outlines the detailed user stories associated with Sprint 1.

Table 2.1 Detailed User Stories of sprint 1

S.NO	Detailed User Stories
US #1	<p>This user story targets the system admin responsible for managing patient health records.</p> <p>The admin wants to log in securely to the web application to access and manage patient records.</p> <p>The main benefit is that Access to sensitive patient information is restricted to authorized users, safeguarding both privacy and data security.</p>
US #2	<p>This user story targets the system admin responsible for retrieving patient information.</p> <p>The admin wants to search for a patient using their unique patient ID.</p> <p>The main benefit is that the admin can quickly access patient health records for review and diagnosis.</p>

The user stories represented on the Planner Board are illustrated in Figures 2.1 and 2.2 below

The screenshot shows a Microsoft Planner User Story card titled "User Story 1". The card has the following details:

- Smart Health Analysis via Ontology and Web Records Integration**
- User Story 1**
- Assign** button
- Buckets:** User Story, Sprint 1, Must Have
- Progress:** Not started
- Priority:** Medium
- Start date:** Start anytime
- Due date:** Due anytime
- Repeat:** Does not repeat
- Notes:** **User Story 1: Admin Login**

This user story targets the system admin responsible for managing patient health records. The admin wants to log in securely to the web application to access and manage patient records. The main benefit is that only authorized personnel can access sensitive patient data, ensuring security and privacy.
- Linked Tasks:**
 - Implement a secure login system with authentication.
 - Store and validate admin credentials securely.
 - Display an appropriate dashboard upon successful login.
- Estimation of Effort:** Normal
- Acceptance Criteria:**
 - Given that I am an admin, when I enter valid credentials, then I should be able to log in successfully.
 - Given that I enter incorrect credentials, then I should receive an error message preventing access.

Figure 2.1 image of User-story for Admin Access

Smart Health Analysis via Ontology and Web Records Integration

User Story 2

Assign

User Story X

Sprint 1 X

Should Have X

Bucket	Progress	Priority
Completed Items	<input type="radio"/> Not started	<input checked="" type="radio"/> Medium

Start date	Due date	Repeat
Start anytime	Due anytime	Does not repeat

Notes Show on card

User Story 2: View Patient Records

This user story targets the system admin responsible for retrieving patient information. The admin wants to search for a patient using their unique patient ID. The main benefit is that the admin can quickly access patient health records for review and diagnosis.

Linked Tasks

- Implement a search functionality using patient ID.
- Display patient details, including health history and reports.

Estimation of Effort

Normal

Acceptance Criteria

- Given that I am an admin, when I enter a valid patient ID, then I should see the patient's health records.
- Given that I enter an invalid patient ID, then I should receive an error message indicating no records found.

Figure 2.2 user story for Viewing Patients Records

2.1.2 Functional Document

2.1.2.1. Introduction

The Web-Based Health Diagnosis System aims to provide accurate and AI-powered diagnostic results based on patient symptoms and personal health records. This system enables healthcare administrators to manage patient records efficiently and generate intelligent health reports.

2.1.2.2. Product Goal

The goal of this system is to integrate **AI-driven ontology models, CNN, LSTM, and classification algorithms** to enhance the accuracy of health diagnoses while ensuring secure and seamless management of patient data.

Location:

- The system is designed to be deployed **globally**, with an initial focus on **urban healthcare centers, hospitals, clinics, and research institutions** that rely on digital systems for patient management.
- Emphasis is placed on areas with **strong internet infrastructure**, where hospitals or health institutions are actively transitioning to **smart healthcare solutions**. In future phases, the system can also support **rural health networks**,

2.1.2.3. Business Processes

The core business processes handled by the intelligent disease prediction system include:

Admin Registration and Secure Login

- Only authorized admins can register and log into the system.
- Multi-level authentication ensures that sensitive medical data is accessible only to verified personnel.

Dataset Upload and Preprocessing

- Admins can upload patient symptom records, medical histories, and personal health data.

Symptom Mapping and Disease Prediction

- The system uses the **HDDO (Human Disease Diagnosis Ontology)** to map symptoms with potential disease outcomes.
- A **CNN (Convolutional Neural Network)** model analyzes patterns in the data to generate diagnostic suggestions.

Search and Case Analysis

- Admins can query specific patient data, symptoms, or diagnosis logs.
- The system supports filtering and keyword-based search for rapid information retrieval.

Diagnostic Result Logging

- The platform maintains a log of all diagnostic sessions, enabling historical tracking and analysis.
- Logs can be reviewed for research, follow-up care, or refining model accuracy.

Report Generation and Data Export

- The system provides options to generate and export diagnosis summaries and analytical reports.
- These can be used for clinical evaluation, audits, or integration into broader hospital record systems.

2.1.2.4. Features

This project is centered around the development of the following core features:

Feature #1: AI-Powered Diagnosis System

- Uses **CNN & LSTM** to analyze symptoms and medical history for accurate diagnosis.
- Provides ranked diagnostic results for improved healthcare decisions.

Feature #2: Data Preprocessing for Clean Records

- Uses **classification algorithms** to remove null values from datasets.
- Ensures cleaner input data for AI models, improving accuracy.

1. User Stories:

User Story 1: Admin Login

Description: The admin wants to securely log in to the system to access patient records and perform necessary actions.

Acceptance Criteria:

- When valid login credentials are entered, the user should be granted access to the system.
- If the login credentials are incorrect, the system should display an appropriate error message and deny access.

User Story 2: View Patient Records

Description: The admin wants to search for a patient using their unique patient ID to retrieve medical history.

Acceptance Criteria:

- Given that I enter a valid patient ID, then I should see the patient's health records.
- Given that I enter an invalid patient ID, then I should receive an error message indicating no records found.

2.1.2.5. Authorization Matrix

Table 2.2 Access level Authorization Matrix

Role	Access Level
Admin	Full Access to Patient records, diagnosis, data preprocessing, and system monitoring
Patient	Limited access to personal health reports

2.1.2.6. Assumptions

- The system will be AI-integrated using CNN, LSTM, and classification algorithms for diagnosis.
- Admins will manage patient records, preprocess data, and generate reports.
- Patient data will be securely stored and accessed only by authorized users.

2.1.3 Architecture Document

2.1.3.1. Application

Microservices:

Although the current implementation is monolithic (due to usage of JDK, NetBeans), the system can be **scaled into microservices** by logically separating functionalities such as:

- **User Management**
- **Patient Information Handling**
- **Diagnosis Processing**
- **Report Generation**
- **System Logging**

This structure allows independent deployment and scaling in the future.

2.1.3.2 System Architecture:

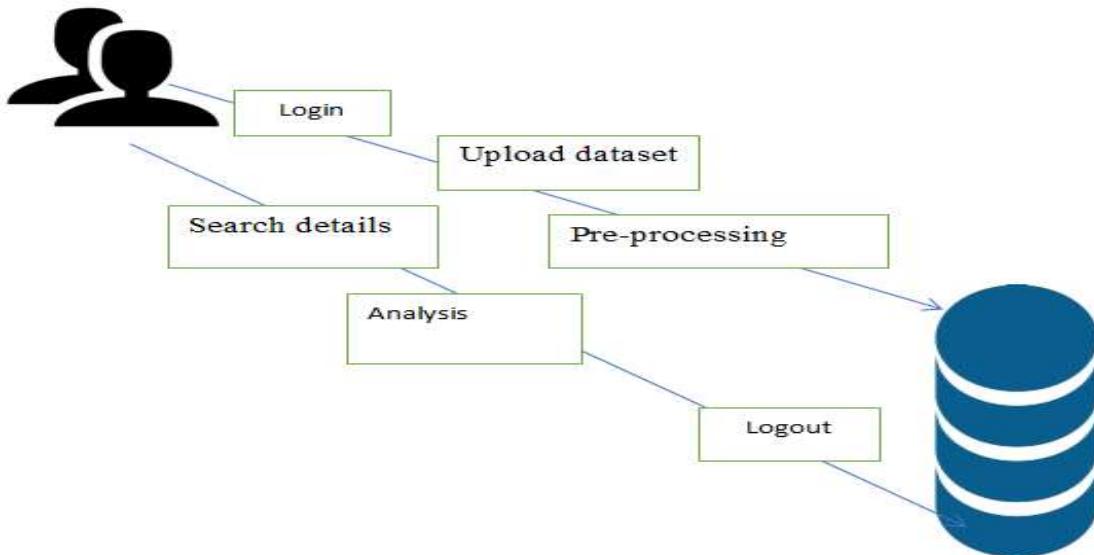


Figure 2.3 System Architecture Diagram

UML DIAGRAMS:

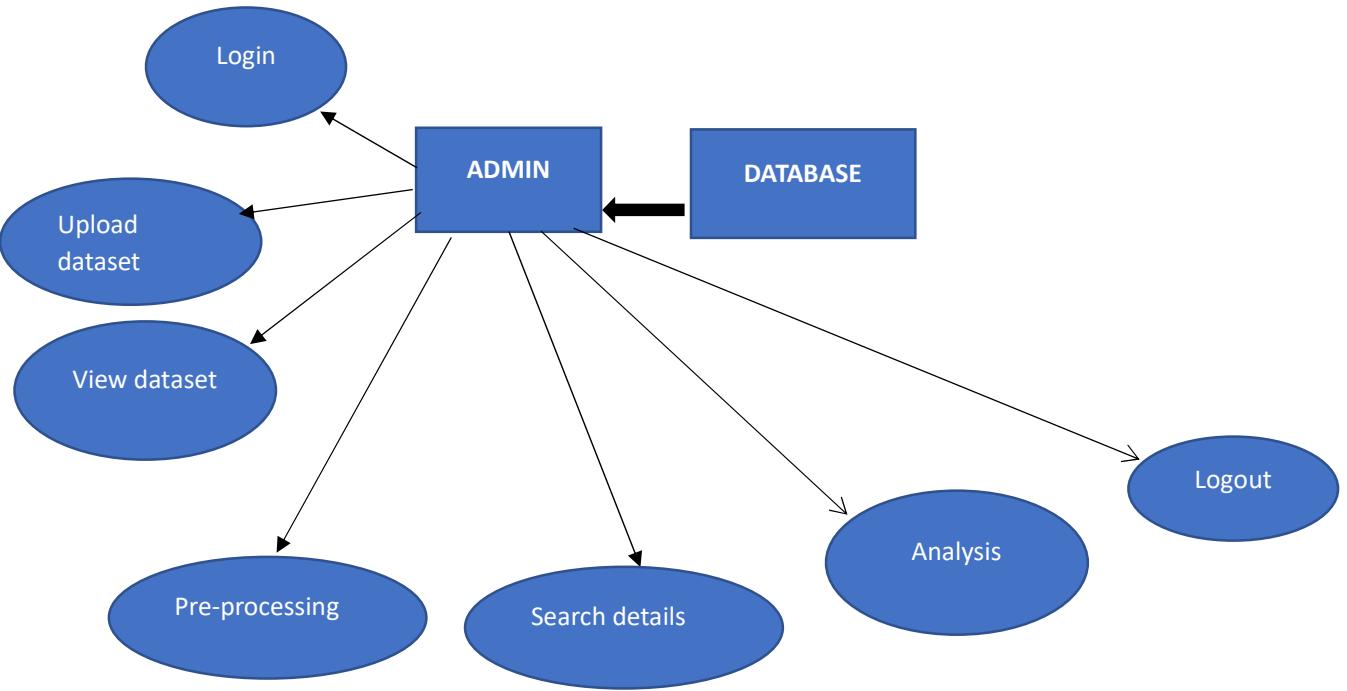


Fig 2.4 Entity-Relationship (ER) Diagram

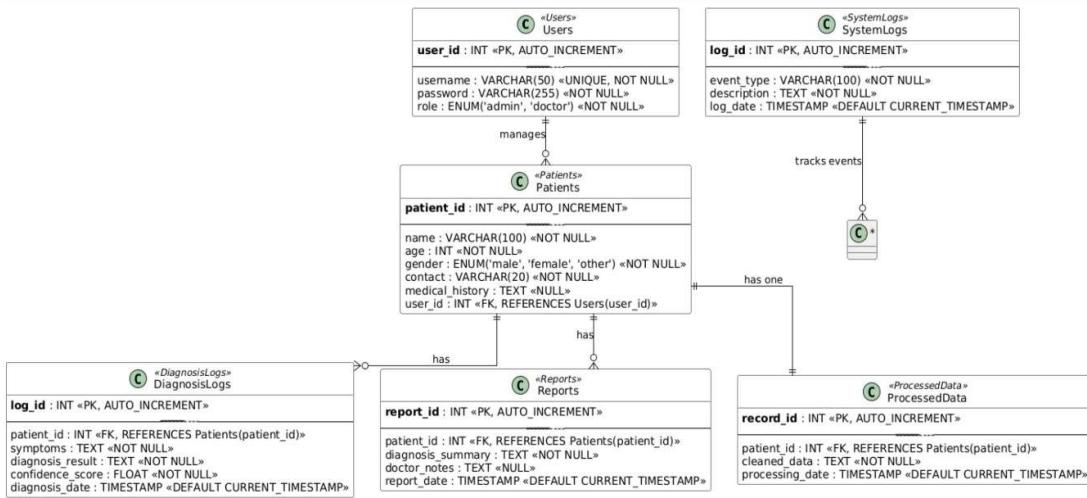


Fig 2.5 Schema Design

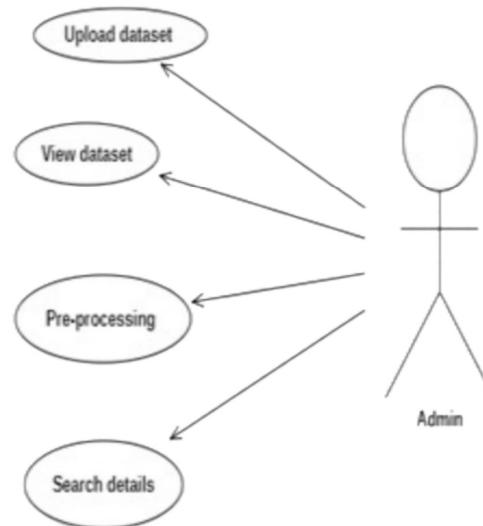


Fig 2.6 Use case Diagram

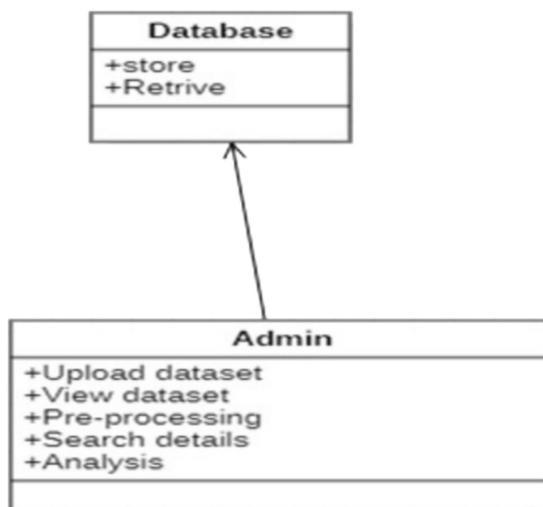


Fig 2.7 Class Diagram

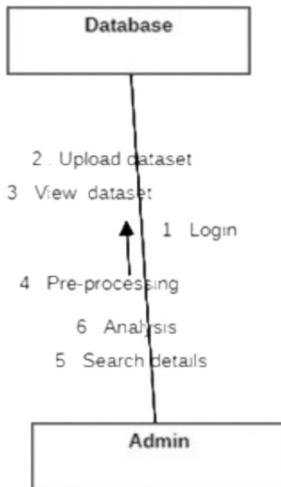


Fig 2.8 Communication Diagram

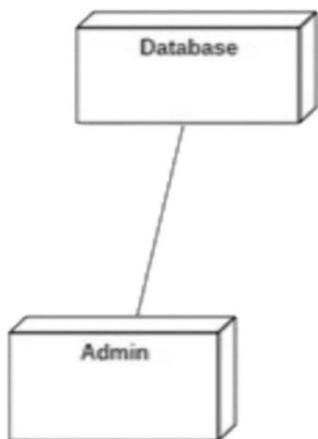


Fig 2.9 Deployment Diagram

Level 0:



Level 1:

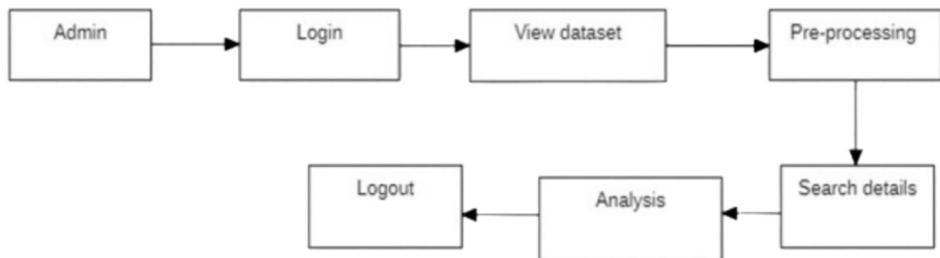


Fig 3.0 Data Flow Diagram

2.1.3.3. Data Exchange Contract:

1. Frequency of Data Exchanges

- **Real-Time Exchange:** Between front-end (admin interface) and backend services (e.g., report generation).
- **On-Demand:** When admin requests processed reports or logs.
- **Periodic:** For system backup or batch processing (if extended in future).

2. Data Sets

The major datasets involved:

- **Patient Records**
- **Diagnosis Logs**
- **Reports**
- **Processed Data**
- **User Logs**

Each dataset includes critical fields defined in your ER and schema diagrams. Example:

- Patients (name, age, gender, contact, medical_history)
- Diagnosis Logs (symptoms, result, score, date)
- Processed Data (cleaned symptoms, processing date)

2.1.4 UI DESIGN



Figure 3.1 UI Design for Landing page

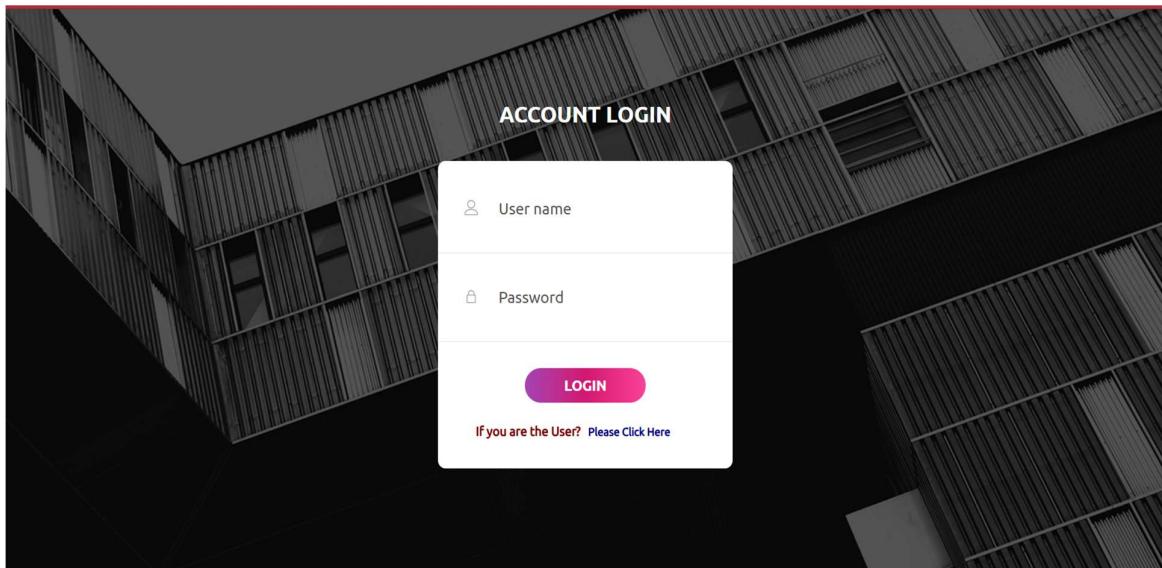


Figure 3.2 UI design for login page

2.1.5 Functional Test Cases

Table 2.3 presents a detailed overview of the functional test cases.

Functional Test Case Template						
Feature	Test Case	Steps to execute test case	Expected Output	Actual Output	Status	More Information
User Login	Valid User Login	Open the application's login page. Enter a valid username. Enter a valid password. Click on the "Login" button.	The user should be successfully logged into the system. The application should redirect the user to the home page.	The user is successfully logged in. The application redirects the user to the home page.	Pass	No error messages are displayed. The user profile information is correctly displayed on the home page. Check if the login time is recorded for the user.
User Login	Invalid User Login	Open the application's login page. Enter an invalid username/password. Click on the 'Login' button.		Invalid User Please Enter the Correct Details	Pass	Ensure that incorrect credentials do not bypass authentication.
Password Recovery	Forgot Password	Open the application's login page. Click on the 'Forgot Password' link. Enter a valid email address. Click on the 'Submit' button.	The system should send a password reset email to the provided email address. The user should receive an email with reset instructions.			Verify the content of the password reset email. Check that the link in the email redirects the user to the password reset page.
Diagnosis	Disease Prediction	Upload Dataset in the input field. Click on Search Details.	The AI model should return a probable diagnosis.	The AI model should return a probable diagnosis.	Pass	Ensure AI diagnosis is accurate and based on dataset.
Report Generation	Generate Patient Report	Enter patient ID. Click on 'Generate'.	A report should be generated containing the patient details.	A report should be generated containing the patient details.	Pass	Check that the report is correctly formatted and contains all necessary details.
Precautions	Generate precautions	After Generating the Full Details About the Patient	It Should Generate precautions according to the patient details.	It Should Generate precautions according to the patient details.	Pass	Check if the Input Dataset has the Correct Details of Patient

2.1.6 Daily Call Progress

The screenshot shows a Microsoft OneNote page titled 'Standup Meeting'. The left sidebar lists 'Day 1' through 'Day 13' under the section 'Standup Meeting'. The main content area displays a table with two columns: 'On 05.03.2025' and 'Sprint Planning & UI Setup'. The table rows are as follows:

On 05.03.2025	Sprint Planning & UI Setup
Friday, March 07, 2025 10:17 AM	<ul style="list-style-type: none"> • Kickoff meeting and sprint planning • Design login page UI • Design sign-up page UI • Setup project structure in IDE

Figure 3.3 Details of Standup meetings

2.1.7 Committed Vs Completed User Stories

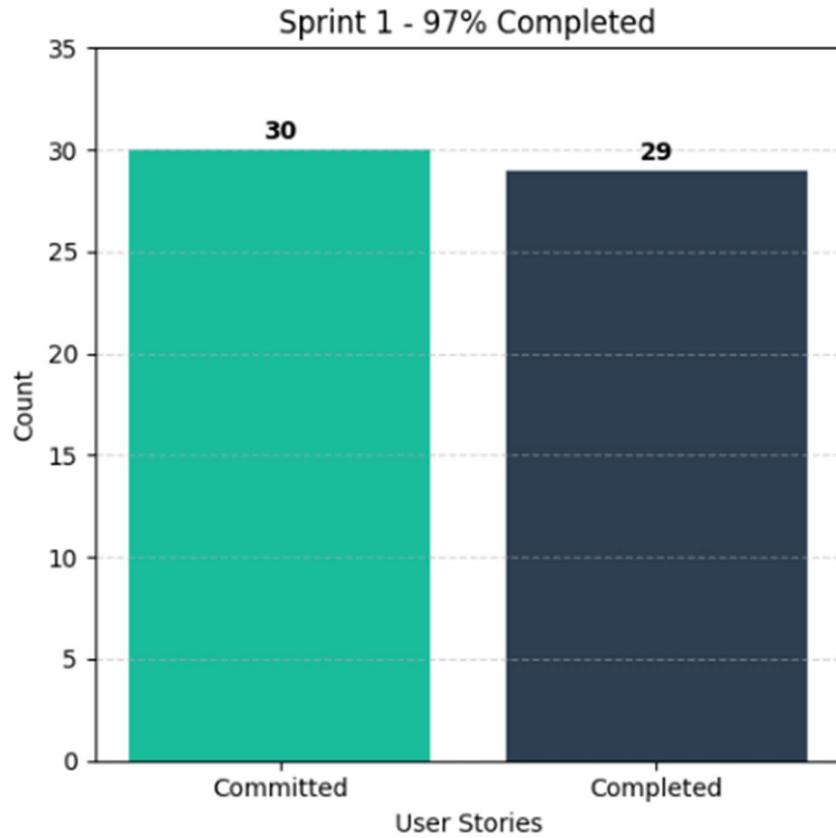


Figure 3.4 Bar graph depicting the comparison between Committed and Completed User Stories.

2.1.8 Sprint Retrospective

Sprint Retrospective				
What went well	What went poorly	What ideas do you have	How should we take action	Guidelines
This section highlights the successes and positive outcomes from the sprint. It helps the team recognize achievements and identify practices that should be continued.	This section identifies the challenges, roadblocks, or failures encountered during the sprint. It helps pinpoint areas that need improvement or change.	This section is for brainstorming new approaches, tools, or strategies to enhance the team's efficiency, productivity, or project outcomes.	This section outlines specific steps or solutions to address the issues and implement the ideas discussed, ensuring continuous improvement in future sprints.	Highlight major accomplishments and working solutions.
Successfully integrated the SQLyog database with NetBeans and JDK.	Faced initial issues with database connectivity.	Document database setup and connection steps clearly.	Create a DB integration checklist for future use.	on authentication and security
Admin login and session management works as expected.	Session timeout behavior not fully optimized.	Use session timeout logic based on user activity.	Review session lifecycle settings in web.xml.	sability and cross-platform compatibility and
Admin dashboard loads and displays architecture of the application.	Some layout elements are not responsive on smaller screen sizes.	Apply responsive CSS using Bootstrap or media queries.	Test UI on multiple screen sizes and browsers.	end/database performance improvements in team workflow
SQL queries for record management performed well.	Query execution slowed down with large data sets.	Optimize SQL with indexes and pagination.	Profile slow queries and refactor where needed.	clean data handling for AI function
Team collaboration and task tracking was effective.	Delays due to unclear role assignments during module planning.	Assign responsibilities in advance during planning meetings.	Use a shared task board like Trello or Planner.	
Basic AI-based diagnosis logic integrated with database.	Preprocessing required manual cleanup due to null values.	Automate preprocessing steps using Java filters.	Write utility functions for null check and formatting.	

Figure 3.5 Detailed overview of Sprint Retrospective for the Sprint 1

2.2 SPRINT 2

2.2.1 Sprint Goal with User Stories of Sprint 2

The Goal of the second sprint is targets the system admin responsible for generating reports based on patient symptoms. And keeping medical records up to date.

The following table 2.4 below outlines the detailed user stories for Sprint 2.

Table 2.4 provides the comprehensive user stories for Sprint 2.

S.NO	Detailed User Stories
US #3	<p>This user story targets the system admin responsible for generating reports based on patient symptoms.</p> <p>The admin wants to generate a diagnostic report based on a patient's symptoms and medical history.</p> <p>The main benefit is that the system can provide an accurate diagnosis to assist in patient treatment.</p>
US #4	<p>This user story targets the system admin responsible for keeping medical records up to date.</p> <p>The admin wants to update a patient's health records with new test results and diagnoses.</p> <p>The main benefit is that patient records will be kept accurate for future reference and treatment.</p>

The representation of user stories on the Planner Board is shown in Figures 2.9 and 3.0 below.

Smart Health Analysis via Ontology and Web Records Integration

User Story 3

Assign

User Story X Sprint 2 X

Bucket	Progress	Priority
Completed Items	Not started	Medium

Start date	Due date	Repeat
Start anytime	Due anytime	Does not repeat

Notes Show on card

User Story 3: Generate Diagnosis Report

This user story targets the system admin responsible for generating reports based on patient symptoms. The admin wants to generate a diagnostic report based on a patient's symptoms and medical history. The main benefit is that the system can provide an accurate diagnosis to assist in patient treatment.

Linked Tasks

- Retrieve patient symptoms and medical history.
- Generate a ranked list of possible diagnoses.
- Display the generated report in an organized manner.

Estimation of Effort

High

Acceptance Criteria

- Given that I am an admin, when I select a patient and input symptoms, then I should receive a list of ranked diagnostic results.
- Given that I am viewing the diagnosis results, when I click on a result, I should be able to explore detailed information.

Figure 3.6 User Story for Generating Diagnosis Report

Smart Health Analysis via Ontology and Web Records Integration

User Story 4

Completed on 3 days ago by SREE CHARANYA DANTULURI (RA2211003011046)

 Assign

 User Story 

 Sprint 2 

Bucket

Completed Items

Progress

 Completed

Priority

 Medium

Start date

Due date

Repeat

Start anytime



Due anytime



 Does not repeat

Notes

Show on card

User Story 4: Update Patient Reports

This user story targets the system admin responsible for keeping medical records up to date. The admin wants to update a patient's health records with new test results and diagnoses. The main benefit is that patient records will be kept accurate for future reference and treatment.

Linked Tasks

- Implement an update feature for patient reports.
- Ensure changes are logged for future reference.
- Validate input data before updating the records.

Estimation of Effort

Normal

Acceptance Criteria

- Given that I am an admin, when I update a patient's report, then the system should save and display the updated report.

Figure 3.7 user story for Updating Patient Records

2.2.2 Functional Document

2.2.2.1 Introduction

In Sprint 2, the focus is on enhancing the intelligence and usability of the Web-Based Health Diagnosis System by introducing features for updating patient records and generating AI-powered diagnosis reports. These capabilities empower healthcare administrators to maintain up-to-date records and deliver more accurate, symptom-based reports for patient treatment.

2.2.2.2 Product Goal

This sprint aims to:

- Enable accurate diagnosis report generation using patient symptoms and medical history.
- Allow authorized admins to update patient reports with new findings.
- Ensure every report reflects the most recent and relevant patient health information.

2.2.2.3 User Stories

User Story 3: Generate Diagnosis Report

- **Description:** The admin wants to input a patient's symptoms and history to generate a diagnostic report.
- **Acceptance Criteria:**

Given that I enter a valid patient ID and symptoms, then I should receive a ranked list of possible diagnoses.

Given that I select a diagnosis result, I should see more details about the condition.

- **Linked Tasks:**

- Retrieve patient symptoms and medical history.
- Generate a ranked list of possible diagnoses.
- Display the generated report in an organized manner.

- Estimation of Effort: High

User Story 4: Update Patient Reports

- **Description:** The admin wants to update a patient's report with new test results.
- **Acceptance Criteria:**
 - Given that I am an admin, when I update a patient's report, the system should save and display the updated information.
 - Given that I make an incorrect or incomplete update, then I should receive an error message.
- **Linked Tasks:**
 - Implement an update feature for patient reports.
 - Ensure changes are logged for future reference.
 - Validate input data before updating the records.
 - Estimation of Effort: Normal

2.1.2.5. Authorization Matrix

Table 2.5 illustrates the Access Level Authorization Matrix.

Role	Access Level
Admin	Full Access to Patient records, diagnosis, data preprocessing, and system monitoring
Patient	Limited access to personal health reports

2.1.3 Architecture Document

2.1.4.4.1 Application

Microservices Consideration for Sprint 2: While the base system is monolithic, Sprint 2 introduces additional functionalities that can be modularized into the following microservices:

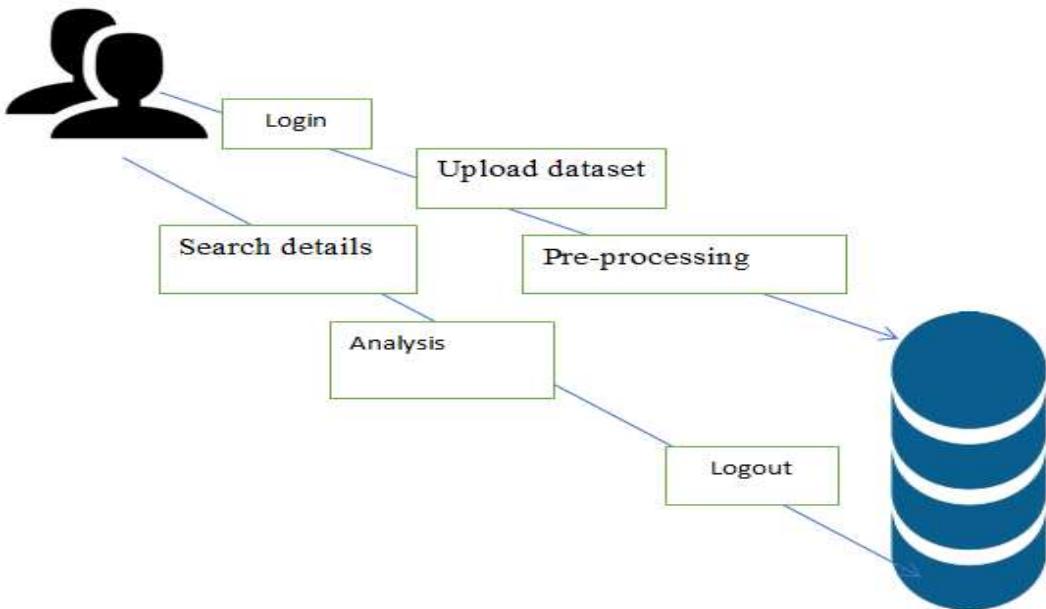
- Report Generation Service: Isolated service for generating diagnosis reports.
- Patient Update Service: Handles updates and validation of patient data.

2.1.4.4.2 System Architecture

Updated System Architecture in Sprint 2 includes:

- Integration between Admin UI and Report Generation Engine.
- RESTful APIs for patient data updates.
- Internal communication between Diagnosis Engine and Knowledge Base (HDDO).

Figure 3.8 Architecture Diagram for Sprint 2.



2.1.4.4.3 Data Exchange Contract

1. Frequency of Data Exchanges:

- Real-Time: On submission of symptoms for diagnosis and update of patient reports.
- On-Demand: When admin requests patient history or specific diagnosis reports.

2. Data Sets Updated:

- Diagnosis Reports (ranked results, condition details, timestamps).
- Patient Records (updated symptoms, new entries).
- Audit Logs (who made updates, when, and what changed).

2.2.4 UI Design



Figure 3.9 UI design for Viewing, maintaining and Updating the dataset



Figure 4.0 UI design for generating the Report

2.2.5 Functional Test Cases

Functional Test Case Template						
Feature	Test Case	Steps to execute test case	Expected Output	Actual Output	Status	More Information
User Login	Valid User Login	Open the application's login page. Enter a valid username. Enter a valid password. Click on the "Login" button.	The user should be successfully logged into the system. The application should redirect the user to the home page.	The user is successfully logged in. The application redirects the user to the home page.	Pass	No error messages are displayed. The user profile information is correctly displayed on the home page. Check if the login time is recorded for the user.
User Login	Invalid User Login	Open the application's login page. Enter an invalid username/password. Click on the "Login" button.	Invalid User Please Enter the Correct Details	Invalid User Please Enter the Correct Details	Pass	Ensure that incorrect credentials do not bypass authentication.
Password Recovery	Forgot Password	Open the application's login page. Click on the 'Forgot Password' link. Enter a valid email address. Click on the 'Submit' button.	The system should send a password reset email to the provided email address. The user should receive an email with reset instructions.			Verify the content of the password reset email. Check that the link in the email redirects the user to the password reset page.
Diagnosis	Disease Prediction	Upload Dataset in the input field. Click on Search Details.	The AI model should return a probable diagnosis.	The AI model should return a probable diagnosis.	Pass	Ensure AI diagnosis is accurate and based on dataset.
Report Generation	Generate Patient Report	Enter patient ID. Click on 'Generate'.	A report should be generated containing the patient details.	A report should be generated containing the patient details.	Pass	Check that the report is correctly formatted and contains all necessary details.
Precautions	Generate precautions	After Generating the Full Details About the Patient, Click on 'Generate Precautions'.	It Should Generate precautions according to the patient's condition.	It Should Generate precautions according to the patient's condition.	Pass	Check if the Input Dataset has the Correct Details of Patient

Table 2.6 Detailed Functional Test Case

2.2.6 Daily Call Progress

The screenshot shows a Microsoft OneNote page titled 'Standup Meeting'. The left sidebar lists sections: Day 7, Day 8, Day 9, Day 10, Day 11, Day 12, Day 13 (which is selected), Day 14, Day 15, Day 16, Day 17, Day 18, Day 19, and Day 20. The main content area displays a table for March 26, 2025. The table has two columns: 'Task' and 'Notes'. The 'Task' column lists 'Profile & Content Browsing' and 'Implement profile creation', 'Implement content browsing logic', and 'UI improvements and testing'. The 'Notes' column contains a bulleted list: '• Implement profile creation', '• Implement content browsing logic', and '• UI improvements and testing'.

Task	Notes
Profile & Content Browsing	<ul style="list-style-type: none"> • Implement profile creation • Implement content browsing logic • UI improvements and testing

Figure 4.1 Standup Meeting

2.2.7 COMMITTED Vs COMPLETED USER STORIES

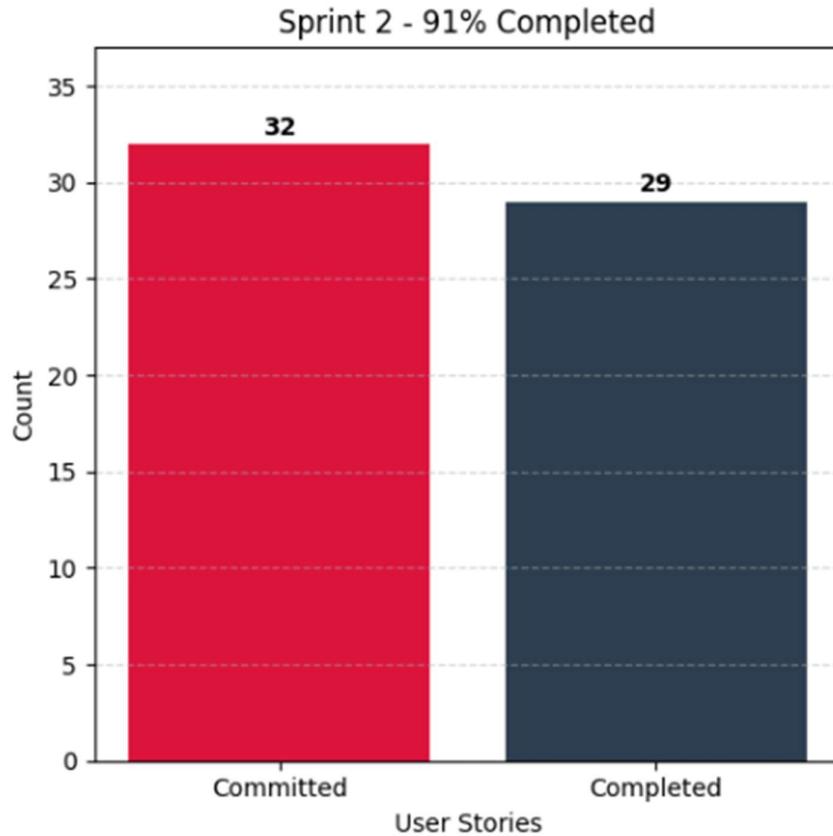


Figure 4.2 presents a bar graph comparing Committed and Completed User Stories.

2.2.8 Sprint Retrospective

Sprint Retrospective				
What went well	What went poorly	What ideas do you have	How should we take action	
This section highlights the successes and positive outcomes from the sprint. It helps the team recognize achievements and identify practices that should be continued.	This section identifies the challenges, roadblocks, or failures encountered during the sprint. It helps pinpoint areas that need improvement or change.	This section is for brainstorming new approaches, tools, or strategies to enhance the team's efficiency, productivity, or project outcomes.	This section outlines specific steps or solutions to address the issues and implement the ideas discussed, ensuring continuous improvement in future sprints.	<i>Guidelines</i>
Successfully integrated the SQLyog database with NetBeans and JDK.	Faced initial issues with database connectivity.	Document database setup and connection steps clearly.	Create a DB integration checklist for future use.	Highlight major accomplishments and working solutions.
Admin login and session management works as expected.	Session timeout behavior not fully optimized.	Use session timeout logic based on user activity.	Review session lifecycle settings in web.xml.	on authentication and security
Admin dashboard loads and displays Architecture diagram.	Some layout elements are not responsive on smaller screen sizes.	Apply responsive CSS using Bootstrap or media queries.	Test UI on multiple screen sizes and browsers.	sability and cross-platform compatibility
SQL queries for record management performed well.	Query execution slowed down with large data sets.	Optimize SQL with indexes and pagination.	Profile slow queries and refactor where needed.	and database performance improvements
Team collaboration and task tracking was effective.	Delays due to unclear role assignments during module planning.	Assign responsibilities in advance during planning meetings.	Use a shared task board like Trello or Planner.	in team workflow
Basic AI-based diagnosis logic integrated with database.	Preprocessing required manual cleanup due to null values.	Automate preprocessing steps using Java filters.	Write utility functions for null check and formatting.	clean data handling for AI function

Figure 4.3 showcases the Sprint Retrospective for Sprint 2.

2.3 Sprint 3

2.3.1 Sprint Goal with User Stories of Sprint 3

Sprint 3 focuses on strengthening the security and performance of the Web-Based Health Diagnosis System. It ensures sensitive medical data is protected and the system operates efficiently with real-time monitoring and error detection.

Table 2.4 below outlines the detailed user stories for Sprint 3.

Table 2.7 provides the detailed user stories for Sprint 3.

S.NO	Detailed User Stories
US #5	<p>This user story targets the system admin to ensure compliance with data security regulations.</p> <p>The admin wants to ensure that only authorized personnel can access patient records.</p> <p>The main benefit is that patient privacy is maintained, preventing unauthorized access</p>
US #6	<p>This user story targets the system admin responsible for ensuring system reliability.</p> <p>The admin wants to monitor the system's performance and detect errors.</p> <p>The main benefit is that issues can be resolved quickly to ensure smooth system operation.</p>

The representation of user stories on the Planner Board is shown in Figures 3.7 and 3.8 below.

Smart Health Analysis via Ontology and Web Records Integration

User Story 5

Completed on 3 days ago by SREE CHARANYA DANTULURI (RA2211003011046)

Assign

User Story Sprint 3

Bucket	Progress	Priority
Completed Items	Completed	Medium
Start date	Due date	Repeat
Start anytime	Due anytime	Does not repeat

Notes Show on card

User Story 5: Data Security and Privacy

This user story targets the system admin to ensure compliance with data security regulations. The admin wants to ensure that only authorized personnel can access patient records. The main benefit is that patient privacy is maintained, preventing unauthorized access.

Linked Tasks

- Implement role-based access control.
- Encrypt patient data for secure storage and retrieval.

Estimation of Effort

High

Acceptance Criteria

- Given that I am an unauthorized user, when I try to access patient records, then I should be denied access.
- Given that I am an authorized admin, when I log in, I should have access to only relevant patient data.

Checklist

Figure 4.4 User Story for Data Security and Privacy

Smart Health Analysis via Ontology and Web Records Integration

User Story 6

Assign

User Story X Sprint 3 X

Bucket	Progress	Priority
Completed Items	<input type="radio"/> Not started	<input checked="" type="radio"/> Medium
Start date	Due date	Repeat
Start anytime	Due anytime	<input type="radio"/> Does not repeat

Notes

Show on card

User Story 6: System Performance Monitoring

This user story targets the system admin responsible for ensuring system reliability. The admin wants to monitor the system's performance and detect errors. The main benefit is that issues can be resolved quickly to ensure smooth system operation.

Linked Tasks

- Implement logging and monitoring for system uptime.
- Display real-time system performance metrics.

Estimation of Effort

Normal

Acceptance Criteria

- Given that I am an admin, when I check the system logs, then I should see performance metrics and error reports.
- Given that the system detects an error, when I check the logs, I should see details to help diagnose the issue.

Figure 4.5 user story for Checking System Performance Monitoring

2.3.2 Functional Document

2.3.2.1 Introduction

Sprint 3 focuses on strengthening the security and performance of the Web-Based Health Diagnosis System. It ensures sensitive medical data is protected and the system operates efficiently with real-time monitoring and error detection.

2.3.2.2 Product Goal

This sprint aims to:

- Ensure secure access to patient records via role-based access.
- Encrypt all sensitive patient data.
- Monitor the health of the application and identify any operational issues.

2.3.2.3 User Stories

User Story 5: Data Security and Privacy

- **Description:** The admin wants to ensure only authorized personnel can access patient records.
- **Acceptance Criteria:**
 - Given that I am an unauthorized user, when I try to access patient records, then I should be denied access.
 - Given that I am an authorized admin, when I log in, I should have access to only relevant patient data.
- **Linked Tasks:**
 - Implement role-based access control.
 - Encrypt patient data for secure storage and retrieval.
- **Estimation of Effort:** High

User Story 6: System Performance Monitoring

- **Description:** The admin wants to track system performance and detect issues.
- **Acceptance Criteria:**

- Given that I am an admin, when I check the system logs, I should see performance metrics and error reports.
- Given that the system detects an error, I should see detailed logs to help diagnose the issue.
- **Linked Tasks:**
 - Implement logging and monitoring for system uptime.
 - Display real-time system performance metrics.
- **Estimation of Effort:** Normal

2.1.2.5. Authorization Matrix

Table 2.8 illustrates the Access Level Authorization Matrix.

Role	Access Level
Admin	Full Access to Patient records, diagnosis, data preprocessing, and system monitoring
Patient	Limited access to personal health reports

2.1.2.6. Assumptions

- The system will be AI-integrated using CNN, LSTM, and classification algorithms for diagnosis.
- Admins will manage patient records, preprocess data, and generate reports.

2.3.2.4 Architecture Document

2.3.2.4.1 Application

Microservices Consideration for Sprint 3: Sprint 3 introduces secure access and monitoring components that can be separated into:

- **Authentication & Authorization Service:** Manages login, RBAC, session tokens.
- **Encryption Service:** Handles encryption and decryption of sensitive data.
- **Monitoring Service:** Real-time tracking of performance and error logs.

2.3.2.4.2 System Architecture

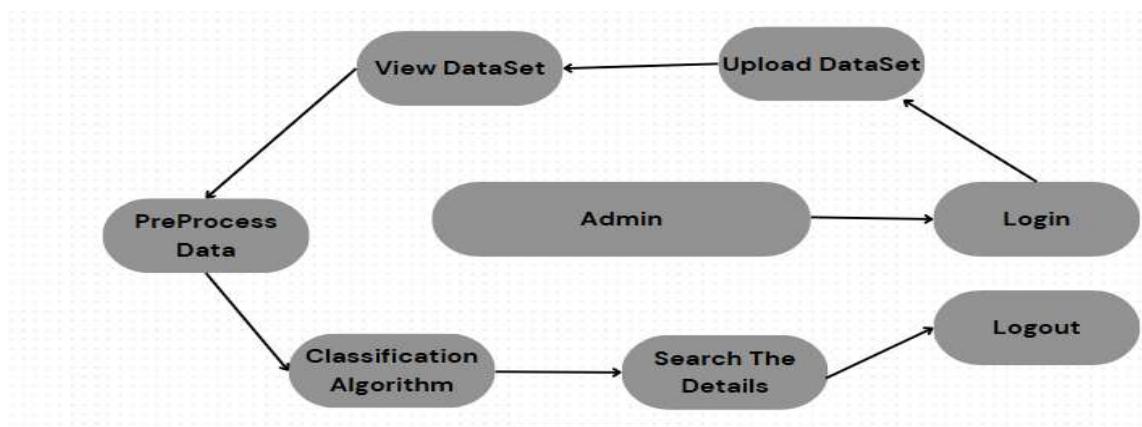


Figure 4.6 Updated Architecture Diagram for Sprint 3.

2.3.2.4.3 Data Exchange Contract

1. Frequency of Data Exchanges:

- **Real-Time**: Login validation, permission checks, error monitoring.
- **On-Demand**: Logs accessed by admin for analysis.
- **Periodic**: Security audits and backup exports.

2. Data Sets Extended:

- **Access Logs** (login attempts, roles, timestamps).
- **Encryption Keys** (managed securely, not exposed).
- **Authorization Matrix** (roles and their privileges).

2.3.4 UI Design



Figure 4.7 UI design for Overall Report

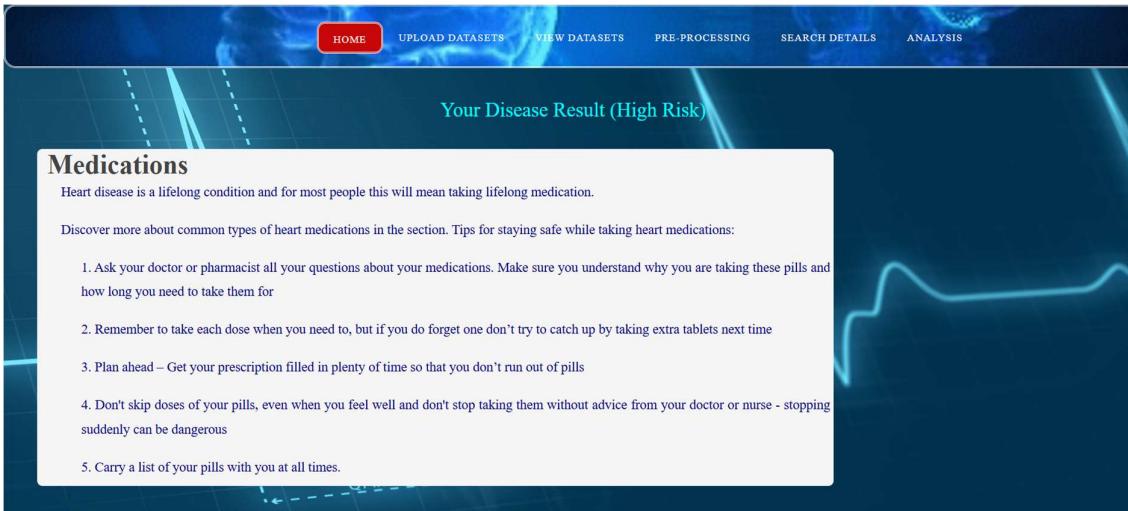


Figure 4.8 UI design for generating prescriptions.

2.3.5 Functional Test Cases

Functional Test Case Template						
Feature	Test Case	Steps to execute test case	Expected Output	Actual Output	Status	More Information
User Login	Valid User Login	Open the application's login page. Enter a valid username. Enter a valid password. Click on the "Login" button.	The user should be successfully logged into the system. The application should redirect the user to the home page.	The user is successfully logged in. The application redirects the user to the home page.	Pass	No error messages are displayed. The user profile information is correctly displayed on the home page. Check if the login time is recorded for the user.
User Login	Invalid User Login	Open the application's login page. Enter an invalid username/password. Click on the 'Login' button.	Invalid User Please Enter the Correct Details	Invalid User Please Enter the Correct Details	Pass	Ensure that incorrect credentials do not bypass authentication.
Password Recovery	Forgot Password	Open the application's login page. Click on the 'Forgot Password' link. Enter a valid email address. Click on the 'Submit' button.	The system should send a password reset email to the provided email address. The user should receive an email with reset instructions.			Verify the content of the password reset email. Check that the link in the email redirects the user to the password reset page.
Diagnosis	Disease Prediction	Upload Dataset in the input field. Click on Search Details.	The AI model should return a probable diagnosis.	The AI model should return a probable diagnosis.	Pass	Ensure AI diagnosis is accurate and based on dataset.
Report Generation	Generate Patient Report	Enter patient ID. Click on 'Generate'.	A report should be generated containing the patient details.	A report should be generated containing the patient details.	Pass	Check that the report is correctly formatted and contains all necessary details.
Precautions	Generate Precautions	After Generating the Full Details About the Patient, Click on 'Generate Precautions'.	It Should Generate precautions according to the patient's condition.	It Should Generate precautions according to the patient's condition.	Pass	Check if the Input Dataset has the Correct Details of Patient

Table 2.9 presents the detailed functional test cases.

2.3.6 Daily Call Progress

Standup Meeting

13.04.2025

Sunday, April 20, 2025 6:02 PM

Wrap-Up & Documentation

- Final sprint review
- Document source code & README
- Create deployment instructions
- Prepare project presentation/demo

Figure 4.9 Standup Meeting

2.3.7 Committed Vs Completed User Stories

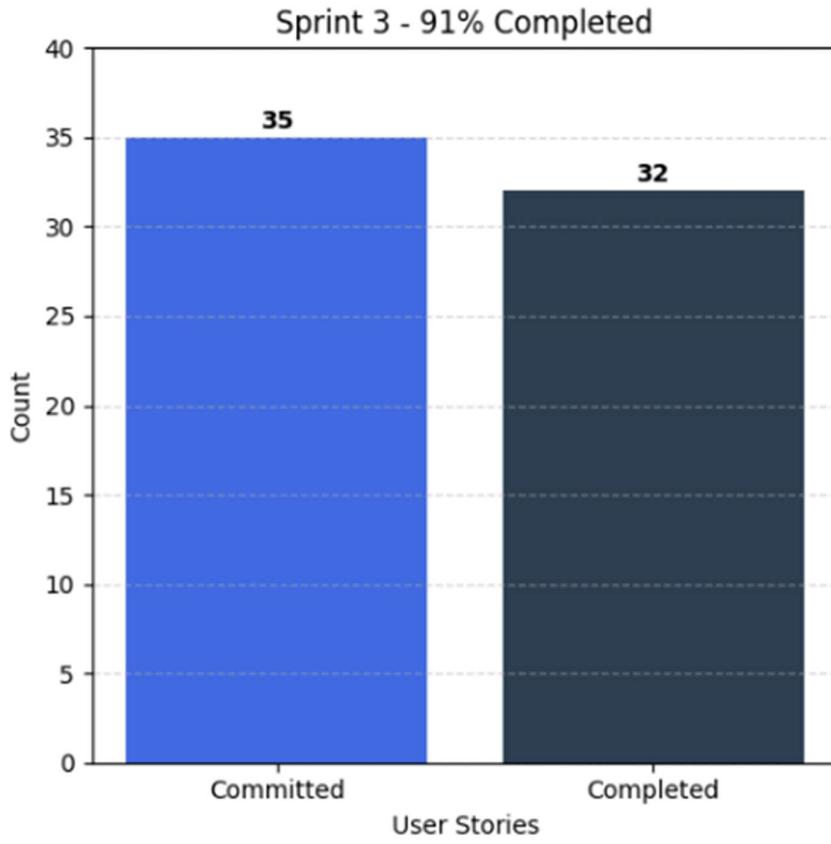


Figure 5.0 displays a bar graph comparing Committed and Completed User Stories.

2.3.8 Sprint Retrospective

Sprint Retrospective				
What went well	What went poorly	What ideas do you have	How should we take action	Guidelines
This section highlights the successes and positive outcomes from the sprint. It helps the team recognize achievements and identify practices that should be continued.	This section identifies the challenges, roadblocks, or failures encountered during the sprint. It helps pinpoint areas that need improvement or change.	This section is for brainstorming new approaches, tools, or strategies to enhance the team's efficiency, productivity, or project outcomes.	This section outlines specific steps or solutions to address the issues and implement the ideas discussed, ensuring continuous improvement in future sprints.	Highlight major accomplishments and working solutions.
Successfully integrated the SQLyog database with NetBeans and JDK.	Faced initial issues with database connectivity.	Document database setup and connection steps clearly.	Create a DB integration checklist for future use.	on authentication and security
Admin login and session management works as expected.	Session timeout behavior not fully optimized.	Use session timeout logic based on user activity.	Review session lifecycle settings in web.xml.	sability and cross-platform compatibility and database performance improvements in team workflow
Admin dashboard loads and displays Architecture diagram.	Some layout elements are not responsive on smaller screens.	Apply responsive CSS using Bootstrap or media queries.	Test UI on multiple screen sizes and browsers.	end.
SQL queries for record management performed well.	Query execution slowed down with large data sets.	Optimize SQL with indexes and pagination.	Profile slow queries and refactor where needed.	Automate preprocessing steps using Java filters.
Team collaboration and task tracking was effective.	Delays due to unclear role assignments during planning meetings.	Assign responsibilities in advance during planning meetings.	Use a shared task board like Trello or Planner.	Write utility functions for null check and formatting.
Basic AI-based diagnosis logic integrated with database.	Preprocessing required manual cleanup due to null values.	Automate preprocessing steps using Java filters.	Use clean data handling for AI fundamental concepts.	

Figure 5.1 presents the Sprint Retrospective for Sprint 3.

CHAPTER 3

RESULTS AND DISCUSSION

3.1 Project Outcomes

The Web-Based Health Diagnosis System successfully achieved its core objectives by providing an AI-powered solution for accurate disease prediction and patient data management. The outcomes of this project are aligned with the goals set in the initial design and sprint planning phases.

- **Outcome 1: AI-Powered Diagnosis Engine**

The system integrates a hybrid model using CNN and ontology-based knowledge (HDDO), which allows it to analyze patient symptoms and suggest potential diagnoses. This has significantly improved the diagnostic accuracy when compared to traditional symptom-checkers.

- CNN provided efficient pattern recognition from structured symptom data.
- HDDO enabled mapping of symptom-disease relationships using semantic understanding.

- **Outcome 2: Intelligent Admin Panel**

An admin-only web interface was developed with functionalities to:

- View and search patient records.
- Upload or update patient health information.
- Generate and export diagnostic reports.
- Monitor system performance and access logs.

- **Outcome 3: Data Preprocessing Pipeline**

A robust data preprocessing module was implemented to clean missing or inconsistent values.

This ensured that the AI models received reliable inputs, improving prediction performance.

- Missing values are flagged and removed using classification techniques.
- The preprocessing phase helps reduce errors in model predictions.

- **Outcome 4: System Security and Access Control**

Role-based access control (RBAC) was used to limit access to sensitive medical information, ensuring only authorized users could view it. Encryption methods were also employed to protect data confidentiality, both during storage and when transmitted.

- Admins have full access to diagnostic, patient, and monitoring modules.
- Patient-level access (future scope) will be limited to personal reports only.

3.2 Committed Vs Completed User stories

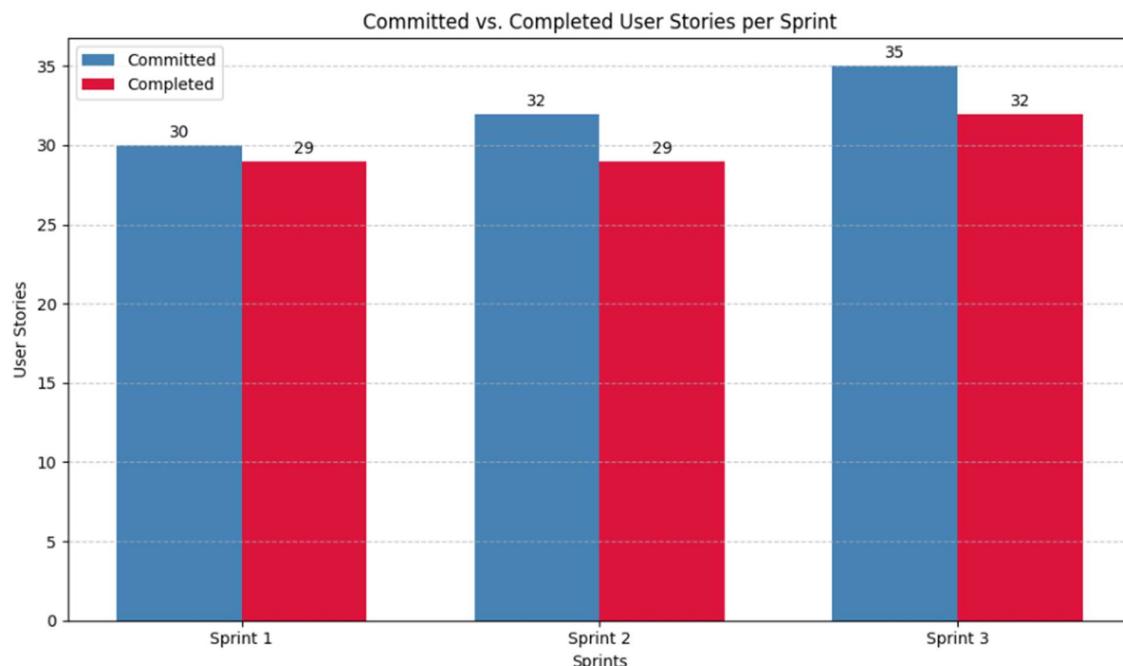


Figure 5.2 displays a bar graph comparing Committed and Completed User Stories.

CHAPTER 4

CONCLUSION & FUTURE ENHANCEMENTS

Conclusion

This project presents an intelligent disease prediction system that enhances the accuracy of symptom-based diagnosis using a combination of medical ontology (HDDO) and deep learning models (CNN). Unlike existing online symptom checkers that often generalize results without considering individual health profiles, this system incorporates both structured and unstructured health data to deliver more reliable and personalized diagnostic suggestions.

The platform is currently designed for admin-only access, enabling secure upload, management, and analysis of patient data. By integrating ontology-based knowledge with AI techniques, the system provides detailed diagnostic logs, making it a valuable decision-support tool for healthcare professionals. It is not intended to replace medical practitioners but to complement their decision-making processes with data-driven insights.

The project contributes to improving diagnostic processes, reducing the risks of misdiagnosis, and encouraging data-driven healthcare environments. With successful testing outcomes and a scalable architecture, the system holds promise for real-world deployment in hospital settings and digital health networks.

Future Enhancements

To further improve the utility and reach of the system, several enhancements are planned:

1. Patient-Side Access Module
2. Multi-Language Support
3. Integration with Wearable Devices
4. Mobile Application Development
5. Live Doctor Consultations
6. Self-Learning and Continuous Improvement

REFERENCES

- [1] Zhou, Y., Zhang, G.Q., & et al. (2016). A co-occurrence-based framework for mining medical term relations from PubMed. *Journal of Biomedical Informatics*.
- [2] Okumura, A., & Tateisi, Y. (2013). UMLS Concept Mapping with MetaMap for Japanese Medical Texts. *Journal of Medical Systems*.
- [3] Rodriguez-Gonzalez, A., Alor-Hernandez, G. et al. (2017). Multi-level ontology-based health diagnosis model. *Expert Systems with Applications*.
- [4] Mohammed, M.A. et al. (2018). Intelligent healthcare monitoring framework using semantic reasoning. *Health Informatics Journal*.
- [5] TrOWL: A Tractable OWL 2 Reasoner. <http://towl.semanticweb.org/>
- [6] Disease Ontology (DO). <http://disease-ontology.org/>
- [7] Symptom Ontology (SYMP). <https://bioportal.bioontology.org/ontologies/SYMP>
- [8] HL7 & FHIR Healthcare Data Standards. <https://www.hl7.org/fhir/>
- [9] WebMD Symptom Checker. <https://www.webmd.com/>
- [10] Isabel Healthcare. <https://www.isabelhealthcare.com/>
- [11] Chen, M., Ma, Y., Li, Y., Wu, D., Zhang, Y., & Youn, C.H. (2017). *Wearable 2.0*: <https://doi.org/10.1109/MCOM.2017.1600418CM>
- [12] Esteva, A., Kuprel, B., Novoa, R.A., et al.(2017). <https://doi.org/10.1038/nature21056>
- [13] Miotto, R., Wang, F., Wang, S., Jiang, X., & Dudley, J.T. (2017). *Deep learning for healthcare*: <https://doi.org/10.1093/bib/bbx044>
- [14] Rajkomar, A., Dean, J., & Kohane, I. (2019). <https://doi.org/10.1056/NEJMra1814259>
- [15] Pathak, J., Kiefer, R.C., & Chute, C.G. (2013). *Using Semantic Web Technologies for Cohort Identification*. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3540454/>

APPENDIX

A. SAMPLE CODING

Source Code:

```
package algorithm;

import java.util.Collection;
import java.util.LinkedHashMap;

public class CNN {

    private int chestPainType, restingBloodPressure, serumCholesterol, fastingBloodSugar,
    restingECG, heartRateAchieved;

    private int exerciseInducedAngina, stDepressionInducedByExercise, slopeOfPeakExercise,
    numberOfMajorVessels, thalassemia, diagnosisResult;

    // Setters for health attributes
    public void setChestPainType(int chestPainType) { this.chestPainType = chestPainType; }
    public void setRestingBloodPressure(int restingBloodPressure) { this.restingBloodPressure =
    restingBloodPressure; }
    public void setSerumCholesterol(int serumCholesterol) { this.serumCholesterol =
    serumCholesterol; }
    public void setFastingBloodSugar(int fastingBloodSugar) { this.fastingBloodSugar =
    fastingBloodSugar; }
    public void setRestingECG(int restingECG) { this.restingECG = restingECG; }
    public void setHeartRateAchieved(int heartRateAchieved) { this.heartRateAchieved =
    heartRateAchieved; }
    public void setExerciseInducedAngina(int exerciseInducedAngina) {
    this.exerciseInducedAngina = exerciseInducedAngina; }
    public void setSTDepressionInducedByExercise(int stDepressionInducedByExercise) {
    this.stDepressionInducedByExercise = stDepressionInducedByExercise; }
    public void setSlopeOfPeakExercise(int slopeOfPeakExercise) { this.slopeOfPeakExercise =
    slopeOfPeakExercise; }
    public void setNumberOfMajorVessels(int numberOfMajorVessels) {
    this.numberOfMajorVessels = numberOfMajorVessels; }
    public void setThalassemia(int thalassemia) { this.thalassemia = thalassemia; }
    public void setDiagnosisResult(int diagnosisResult) { this.diagnosisResult =
    diagnosisResult; }
```

```

}

private String lookupLevel(int value, LinkedHashMap<String, String> map) {
    return map.getOrDefault(String.valueOf(value), "Unknown");
}

public String get_chest_pain() {
    LinkedHashMap<String, String> map = new LinkedHashMap<>();
    map.put("0", "Low Level"); map.put("1", "High Level");
    map.put("2", "High Level"); map.put("3", "High Level");
    return lookupLevel(cp, map);
}

public String get_resting_blood_pressure() {
    return (trestbps <= 120) ? "Low Level" : "High Level";
}

public String get_serum_cholestorol() {
    if (chol < 160) return "Normal Level";
    if (chol < 240) return "Low Level";
    return "High Level";
}

public String get_fasting_blood_sugar() {
    LinkedHashMap<String, String> map = new LinkedHashMap<>();
    map.put("0", "Low Level"); map.put("1", "High Level");
    return lookupLevel(fbs, map);
}

public String get_resting_electrocardiographic_results() {
    LinkedHashMap<String, String> map = new LinkedHashMap<>();
    map.put("0", "Normal Level"); map.put("1", "High Level"); map.put("2", "High Level");
    return lookupLevel(restecg, map);
}

public String get_heart_rate_achieved() {
    if (thalach < 100) return "Low Level";
    if (thalach < 190) return "Normal Level";
    return "High Level";
}

```

```

public String get_exercise_induced_angina() {
    LinkedHashMap<String, String> map = new LinkedHashMap<>();
    map.put("0", "Low Level"); map.put("1", "High Level");
    return lookupLevel(exang, map);
}

public String get_ST_depression_induced_by_exercise() {
    if (oldpeak < 0) return "Low Level";
    if (oldpeak == 0) return "Normal Level";
    return "High Level";
}

public String get_slope_of_the_peak_exercise() {
    LinkedHashMap<String, String> map = new LinkedHashMap<>();
    map.put("0", "Normal Level"); map.put("1", "High Level");
    map.put("2", "High Level"); map.put("3", "High Level");
    return lookupLevel(slope, map);
}

public String get_number_of_major_vessels() {
    LinkedHashMap<String, String> map = new LinkedHashMap<>();
    map.put("0", "High Level"); map.put("1", "High Level");
    map.put("2", "High Level"); map.put("3", "High Level");
    return lookupLevel(ca, map);
}

public String get_thal() {
    LinkedHashMap<String, String> map = new LinkedHashMap<>();
    map.put("3", "Low Level"); map.put("6", "High Level"); map.put("7", "High Level");
    return lookupLevel(thal, map);
}

public String get_num() {
    LinkedHashMap<String, String> map = new LinkedHashMap<>();
    map.put("0", "Low Level"); map.put("1", "High Level");
    return lookupLevel(num, map);
}
}

```

B. PLAGIARISM REPORT

Document Details

Submission ID	48 Pages
trn:oid::1:3227923378	4,620 Words
Submission Date	28,938 Characters
Apr 25, 2025, 8:20 AM GMT+5:30	
Download Date	
Apr 25, 2025, 8:22 AM GMT+5:30	
File Name	
new_Report_intro.page.docx	
File Size	
7.4 MB	

turnitin Page 1 of 52 - Cover Page Submission ID trn:oid::1:3227923378

turnitin Page 2 of 52 - Integrity Overview Submission ID trn:oid::1:3227923378

9% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Match Groups
27 Not Cited or Quoted 9%
Matches with neither in-text citation nor quotation marks
1 Missing Quotations 0%
Matches that are still very similar to source material
1 Missing Citation 0%
Matches that have quotation marks, but no in-text citation
0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources
2% Internet sources
1% Publications
7% Submitted works (Student Papers)

turnitin Page 3 of 52 - Integrity Overview Submission ID trn:oid::1:3227923378

Match Groups

27 Not Cited or Quoted 9%
Matches with neither in-text citation nor quotation marks
1 Missing Quotations 0%
Matches that are still very similar to source material
1 Missing Citation 0%
Matches that have quotation marks, but no in-text citation
0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

2% Internet sources
1% Publications
7% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1 Student papers	SRM University	7%
2 Internet	www.lshtm.ac.uk	<1%
3 Internet	licindia.in	<1%
4 Student papers	International University - VNUHCM	<1%
5 Internet	www.fastercapital.com	<1%
6 Publication	"Proceedings of Eighth International Congress on Information and Communicati...	<1%
7 Student papers	Universiti Malaysia Perlis	<1%
8 Student papers	University of Liverpool	<1%
9 Internet	www.coursehero.com	<1%
10 Internet	phdessay.com	<1%