



BSCS-S25-003

03-134202-013 ASAD ALI

03-134221-033 NAZAR ALI

AI Assisted Caretaker For Neurodegenerative Disorder

In partial fulfilment of the requirements for the degree of
Bachelor of Science in Computer Science

Supervisor: Dr. Iram Noreen

Co-Supervisor: Ms. Zoobia Fatima

**Department of Computer Sciences
Bahria University, Lahore Campus**

January 2026



C e r t i f i c a t e

We accept the work contained in the report titled
“AI Assisted Caretaker For Neurodegenerative Disorder Patients”

written by
ASAD ALI
NAZAR ALI

as a confirmation to the required standard for the partial fulfilment of the degree of
Bachelor of Science in Computer Science.

Approved by:

Supervisor: Dr. Iram Noreen

Co-Supervisor: Miss Zoobia Fatima

January 05, 2026

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-134202-013	Asad Ali	
03-134221-033	Nazar Ali	

Date: January 05, 2026

Specially dedicated to
My beloved Mother, Father and Teachers
(ASAD ALI)
My beloved Mother, Father and Teachers
(NAZAR ALI)

ACKNOWLEDGEMENTS

We would like to thank everyone who had contributed to the successful completion of this project. We would like to express our gratitude to our supervisor, Dr. Iram Noreen and co-supervisor, Ms. Zoobia Fatima for their invaluable advice, guidance and for their patience throughout the development of this project.

In addition, we would like to express our gratitude to our respected parents and teachers who had helped and given us encouragement.

ASAD ALI
NAZAR ALI

AI Assisted Caretaker For Neurodegenerative Disorder Patients

ABSTRACT

Neurological disorders encompass a broad spectrum of diseases affecting the brain, spinal cord, and nervous system. Almost 600 neurological disorders exist globally. Early diagnosis is critical for mitigating the societal and economic impact of Neurodegenerative Disorders (NDD). Alzheimer's Disease (AD) is one of the most prevalent neurodegenerative disorder. Additionally, there is no cure to AD. The early and precise diagnosis of AD remains a significant challenge in clinical neuro-science. Neuroimaging is one of the significant biomarker for early diagnosis of AD. The research work comprises of Magnetic Resonance Imaging (MRI) data acquired from Kaggle. Dataset consists of 4 classes: Mild Demented (8960), VeryMildDemented (8960), Moderate Demented (6464), NonDemented (9600) , 33706 samples in total. Pillow (PIL) is used for data preprocessing to greyscale, resize and normalize the image samples. For classification Convolutional Neural Network (CNN) is used with 3 convolutional blocks and 1 hidden layer. Grid Search is used for hyper-parameter tuning. Accuracy achieved is 97%. RAG with HuggingFace Transformer model embeddings is used to generate context-aware and intelligent responses. Notification system to notify patients and doctors about important events. Appointment Scheduling module is used to schedule and keep track of appointments between doctor and patients. Prescription table is used in system to keep record of patient prescription after an appointment.

Keywords: Neurological disorders, neurodegenerative diseases, Alzheimer's disease, Parkinson's disease, early diagnosis, economic burden, public health.

TABLE OF CONTENTS

CHAPTER 1	1
1 INTRODUCTION	1
1.1 Background	1
1.2 Problem Statements	3
1.3 Aims and Objectives	3
1.4 Scope of Project	4
CHAPTER 2	5
2 LITERATURE REVIEW	5
2.1 Introduction	5
2.2 Areas of Research	7
2.3 User Classes and Characteristics	10
2.4 Operating Environment	11
2.5 Design and Implementation Constraints	11
2.6 Assumptions and Dependencies	12
2.7 External Interface Requirements	12
CHAPTER 3	14
3 DESIGN AND METHODOLOGY	14
3.1 Methodology	14
3.2 System Design	16
3.3 Use Case Description	16
3.4 Use Case Diagram	28
3.5 Sequence Diagram	30
3.6 Collaboration Diagrams	34

3.7	Operation Contracts	40
3.8	Design Class Diagram	43
3.9	Data Model	44
CHAPTER 4		45
4	DATA AND IMPLMENTATION	45
4.1	Software Development Methodology	45
4.2	MRI Model Implementation	46
4.3	Chatbot Implementation	50
4.4	Web-Application	51
4.5	APIs Implementation	52
4.6	Integration	52
CHAPTER 5		53
5	RESULTS AND DISCUSSIONS	53
5.1	MRI Model Results	53
5.2	Confusion Matrix	54
5.3	Graphical User Interface	55
CHAPTER 6		65
6.1	CONCLUSION AND RECOMMENDATIONS	65
6.2	REFERENCES	66
6.3	Appendix	67

LIST OF TABLES

TABLE	TITLE	PAGE
Table 2.1:	Software Interfaces	12
Table 2.2:	Software	12
Table 3.1:	UC Self-Register User	17
Table 3.2	UC user login	18
Table 3.3:	UC Patient Upload Image	22
Table 3.4:	UC User Logout	27
Table 3.5:	OC User Login	40
Table 3.6:	OC User Logout	40
Table 3.7:	OC Patient	41
Table 3.8:	OC Doctor	41
Table 3.9:	OC Admin	42
Table 3.10:	OC Admin (Register User)	42

LIST OF FIGURES

FIGURE	TITLE	PAGE
Figure 3.1:	Feature Driven Development (FDD)	Error! Bookmark not defined.
Figure 3.2:	System Use Case	Error! Bookmark not defined.
Figure 3.3:	Doctor (Upload Image) Use Case	Error! Bookmark not defined.
Figure 3.4:	Patient Use Case	Error! Bookmark not defined.
Figure 3.9:	SD Logout	Error! Bookmark not defined.
Figure 3.10:	CD Admin	Error! Bookmark not defined.
Figure 3.11:	CD Admin Collaboration	Error! Bookmark not defined.
Figure 3.12:	CD Doctor/Patient (Login)	Error! Bookmark not defined.
Figure 3.13:	CD Doctor Query System	Error! Bookmark not defined.
Figure 3.14:	CD Doctor MRI Analysis	Error! Bookmark not defined.
Figure 3.15:	CD Patient Interacts System	Error! Bookmark not defined.
Figure 3.16:	CD User Logout	Error! Bookmark not defined.
Figure 3.17:	Class Diagram	Error! Bookmark not defined.
Figure 3.18:	Data Model	Error! Bookmark not defined.
Figure 4.1:	Top Level Architecture	Error! Bookmark not defined.
Figure 4.2:	Architecture of CNN	Error! Bookmark not defined.
Figure 4.2.3:	Model Summary	Error! Bookmark not defined.
Figure 4.2.4:	Train and Test Data samples	Error! Bookmark not defined.
Figure 4.2.5:	Training Information	Error! Bookmark not defined.
Figure 5.1:	Confusion Matrix of CNN Model	Error! Bookmark not defined.
Figure 5.3.4.:	Admin Login UI	Error! Bookmark not defined.
Figure 5.3.5:	Admin Dashboard UI	Error! Bookmark not defined.

LIST OF SYMBOLS / ABBREVIATIONS

AD	Alzheimer's disease
MCI	Mild Cognitive Impairment
MRI	Magnetic Resonance Imaging
ADNI	Alzheimer's disease Neuroimaging Initiative
CDR	Clinical Dementia Rating
UML	Unified Modelling Language
OASIS	Organization for the Advancement of Structured Information Standards
CNN	Convolutional Neural Network
OC	Operation Contract
UC	Use Case
SD	Sequence Diagram
UI	User Interface
ROC	Receiver Operating Characteristics
CD	Collaboration Diagram

CHAPTER 1

INTRODUCTION

1.1 Background

Neurological diseases are a large category of diseases that affect the brain, spinal cord, and nerve system, resulting in deficits in cognition, movement, sensation, and behavior. These diseases are classified into six primary categories: neurodegenerative disorders due to loss of neurons [1], neurodevelopmental disorders are conditions affecting the growth and maturity of the nervous system and the brain [2], neurological disorders caused by injury or infection, psychiatric disorders, functional neurological disorders, and autoimmune/inflammatory neurological disorders. Each group involves distinct issues in diagnosis, treatment, and care, frequently necessitating a multidisciplinary approach. Figure 1.1 depicts the categories of cognitive disorders and their subtypes. However neurodegenerative diseases require extensive research since they are progressive and irreversible, with a considerable impact on patients' quality of life. The rising frequency of these diseases emphasizes the need for advances in early identification and efficient treatment. The inherent difficulties and acute need for answers in neurodegenerative diseases motivate targeted research efforts. Enhancing the life quality for those impacted is consistent with the SDG (Sustainable Development Goal) of fostering health and well-being for everyone. Neuro-degenerative disorders harm neurons and nervous system associations, affecting strength, movement, cognition, and the ability to coordinate in the body[3]. NDs are one of the main causes of lifelong disability and mortality globally and impact the life quality of people [4][5][6]. According to the WHO's studies, over a billion humans globally suffer from neurological diseases [7]. Per the National Institute of Neurological Diseases and Stroke, there are over

600 neurologic diseases, with over 50 million Americans impacted annually. Furthermore, patients and caregivers have a significant emotional strain on financial expenditures. As the old population grows, so will society's costs[8]. Research suggests that a mix of genetic, environmental, and lifestyle factors likely causes neurological disorders[9][10].

Historically, environmental factors were primarily associated with cancer risk. Cancer and neurodegeneration are opposing extremes of the continuum. Cancer is excessive cell growth, whereas neurodegeneration is the outcome of cell death, either through necrosis or apoptosis. Environmental chemicals can damage the neurological system, leading to neurodegenerative disorders. Risk factors for neurological diseases include genetic variations and advancing age. Other factors that may contribute include gender, inadequate education, endocrine problems, oxidative stress, inflammation, stroke, hypertension, diabetes, smoking, head trauma, depression, infection, tumors, vitamin deficiencies, immunological and metabolic diseases, and chemical exposure. The pathophysiology for numerous of these disorders is yet unclear[8].

In addition to the 'direct' costs involved with the (para) medical care of neurodegenerative diseases, it is critical to recognize that the entire financial burden, which includes expenses due to a reduced standard of life also employment, may be considerably larger. This is especially important in diseases like MS (multiple sclerosis), Alzheimer's dementia (AD), Parkinsons, etc., which are common and severe and can last for decades. Furthermore, it is estimated that up to 80% of the subjects affected by Sclerosis become unemployed within the first 15 years following diagnosis, leading to a cost that is about 4000 USD per year higher than healthy age- and gender-matched controls, whereas most of the yearly total patient care expenses in AD (approximately USD 300) [11]. Alzheimer's care expenses in the US are expected to reach \$1.1 trillion (inflation-adjusted) by 2050, from an estimated \$500 billion in 2020[5]. For Parkinson's, annual expenditures per patient varied from around \$1750 in Canada to \$17,560 in Germany [5]. As an outcome, neurological diseases not only reduce the life quality but also impose a major cost on their families.

Accurate detection of neurological disorders is important to mitigating their societal and fiscal burden. Early identification enables prompt therapies, which can decrease disease development and enhance patients' standard of life. It enables

clinicians to design tailored treatment regimens, suggest lifestyle adjustments, and support family members and patients more. An accurate diagnosis also aids research by allowing scientists to gain additional insight into the origins, risk factors, and potential therapies. This can result in better medicines and patient care. Early diagnosis also allows patients and their families to prepare, seek help, and make educated decisions. Furthermore, this can improve doctors' and researchers' understanding of disease states, their causes, and treatment possibilities.

Built using the MERN stack for web development and Python-based AI Models, this system ensures accessibility, efficiency and accuracy. By automating essential care-giving tasks, it reduces the burden of caregivers & family members while improving the quality of life for patients. The project aims to demonstrate how AI can play a crucial role in healthcare by providing personalized assistance, early diagnosis and better patient management.

1.2 Problem Statements

Neurodegenerative disorders particularly Alzheimer need continuous, early diagnosis and personalized care. Traditional health monitoring struggles with lack of awareness, patient monitoring and accessibility to the specialist. These disorders cannot be cured, but can be slowed down if predicted early. To address these challenges the project proposes a novel AI- health-care system for neurodegenerative disorders that integrates Notifier System, Chatbot, MRI Scan Analysis & Prediction and Specialist Recommendation to provide efficient, proactive and personalized care system.

1.3 Aims and Objectives

The main goal of this project is to develop an AI-assisted caretaker system that helps patients with neurodegenerative disorders by providing real-time support, early disease detection, and personalized recommendations. To achieve this, the project focuses on the following key objectives:

1. Develop an AI-powered chat-bot using Natural Language Processing (NLP) to assist patients with daily activities, answer their queries, and engage them in meaningful conversations to improve cognitive function.

2. Implement a smart notifier system that sends automated reminders for medication, doctor appointments, and other important tasks to ensure patients stay on track with their treatment plans.
3. Create an MRI analysis module using Convolutional Neural Networks (CNNs) to detect early signs of neurodegeneration, helping doctors diagnose and monitor diseases like Alzheimer's and Parkinson's more effectively.
4. Design a Patient Appointment System that analyze patient history and symptoms to suggest the most relevant neurologists, therapists, or medical specialists, ensuring timely and accurate medical care.
5. Develop a history management and disease progression monitoring module to track patient symptoms, MRI scan results, and treatment progress over time, allowing doctors and caregivers to make better-informed decisions based on a patient's medical history.
6. Build a user-friendly web-based platform using the MERN stack (MongoDB, Express.js, React.js, Node.js) to integrate all AI-powered features and provide an accessible, efficient, and seamless experience for patients, caregivers, and healthcare professionals.

These objectives ensure that the system not only assists patients in their daily routines but also enhances early diagnosis, reduces caregiver burden, and improves the overall management of neurodegenerative disorder patients.

1.4 Scope of Project

A web-based AI-powered platform integrating:

- Chatbot
- Notifier System
- MRI Scan Analysis and prediction
- History analysis and disease progression monitoring

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Neuro-degenerative disorders harm neurons and nervous system associations, affecting strength, movement, cognition, and the ability to coordinate in the body[3]. NDs are one of the main causes of lifelong disability and mortality globally and impact the life quality of people[4]. According to the WHO's studies, over a billion humans globally suffer from neurological diseases [1]. Per the National Institute of Neurological Diseases and Stroke, there are over 600 neurologic diseases, with over 50 million Americans impacted annually. Furthermore, patients and caregivers have a significant emotional strain on financial expenditures. As the old population grows, so will society's costs[8]. Research suggests that a mix of genetic, environmental, and lifestyle factors likely causes neurological disorders[4].

Historically, environmental factors were primarily associated with cancer risk. Cancer and neurodegeneration are opposing extremes of the continuum. Cancer is excessive cell growth, whereas neurodegeneration is the outcome of cell death, either through necrosis or apoptosis. Environmental chemicals can damage the neurological system, leading to neurodegenerative disorders. Risk factors for neurological diseases include genetic variations and advancing age. Other factors that may contribute include gender, inadequate education, endocrine problems, oxidative stress, inflammation, stroke, hypertension, diabetes, smoking, head trauma, depression, infection, tumors, vitamin deficiencies, immunological and metabolic diseases, and chemical exposure. The pathophysiology for numerous of these disorders is yet unclear[4].

In addition to the direct medical costs of neurodegenerative diseases, it is critical to recognize that the overall financial burden — including reduced quality of life and lost employment or productivity — may be much larger. This is especially

important for chronic, progressive conditions such as Multiple sclerosis (MS), Alzheimer's disease (AD), and Parkinson's disease, which are often severe and can persist for decades. For instance, in MS the economic burden per patient (direct + indirect) has been estimated at tens of thousands of euros per year depending on disability status. [6]

Employment and work-capacity are heavily impacted: many MS patients stop working or reduce work hours following diagnosis. For Alzheimer's disease and other dementias, annual per-person costs rise dramatically with disease severity — from under US \$500 in mild cases to over US \$170,000 in severe cases (including formal care, social care, and informal care) according to a recent systematic review. [7].

Accurate detection of neurological disorders & proper patient management is important to mitigating their societal and fiscal burden. Early identification enables prompt therapies, which can decrease disease development and enhance patients' standard of life. It enables clinicians to design tailored treatment regimens, suggest lifestyle adjustments, and support family members and patients more. An accurate diagnosis also aids research by allowing scientists to gain additional insight into the origins, risk factors, and potential therapies. This can result in better medicines and patient care. Early diagnosis also allows patients and their families to prepare, seek help, and make educated decisions. Furthermore, this can improve doctors' and researchers' understanding of disease states, their causes, and treatment possibilities. Built using the MERN stack for web development and Python-based AI Models, this system ensures accessibility, efficiency and accuracy. By automating essential caregiving tasks, it reduces the burden of caregivers & family members while improving the quality of life for patients. The project aims to demonstrate how AI can play a crucial role in healthcare by providing personalized assistance, early disease detection and better patient history management.

2.2 Areas of Research

2.2.1 Digital Health & Hospital Information Systems

In recent years, the use of digital healthcare and information system have transformed how hospitals and caregivers manage patient data, communication and workflows. Specifically for Alzheimer's, it has shown good results that show the significance of Digital health records in keeping a consistent patient history and monitor disease progression [11]. Which supports our initiative to store E-records and history of patients.

2.2.2 Alzheimer's disease and it's impact

Alzheimer's disease (AD) is the leading cause of dementia and a growing public-health challenge as populations age. I recognize that AD causes progressive cognitive decline, loss of independence, and a substantial caregiver burden; its prevalence and projected growth underline the need for earlier diagnosis and improved long-term management strategies [4]. Early detection is essential because interventions, care planning, and safety measures are most effective when implemented in preclinical or mild stages [4]

2.2.3 The promise and limitations of AI

I reviewed literature showing that artificial intelligence (AI) and machine learning (ML) can identify subtle imaging and multimodal biomarkers, improving diagnostic sensitivity compared to routine clinical reads. However, many studies remain proof-of-concept: they often use small or homogeneous datasets, report limited external validation, and struggle with clinical generalizability. Important concerns include data heterogeneity, explainability, and how tools will be integrated into clinician and caregiver workflows [5].

2.2.4 CNN for MRI Analysis & challenges

CNNs are the currently dominant deep-learning approach for MRI-based AD detection, including 2D slice-based, 3D volumetric, and hybrid architectures. Many studies report the ability of CNNs to classify AD vs. MCI and healthy controls with

encouraging accuracy, with 3D models often capturing richer spatial patterns of hippocampal and cortical atrophy. However, the methodological variability (preprocessing, cross-validation, dataset leakage) and sparse reporting of clinically interpretable features raise concerns about the generalizability of these findings, and systematic reviews show a dearth of multi-center external validation [6]. Explainability methods, such as saliency maps and region-based attributions, only partially alleviate the black-box nature of these models, and are not standardized for regulatory or bedside use [6][11].

2.2.5 NLP chatbots and RAG for healthcare communication

Conversational agents have been explored for patient education, triage, and caregiver support. Traditional rule-based chatbots offer predictable responses but lack the adaptability of large language models (LLMs). Retrieval-Augmented Generation (RAG), which combines LLMs with a document retriever improves factual accuracy and enables domain-specific knowledge injection, making it attractive for healthcare Q&A. Recent studies suggest RAG can reduce hallucinations and provide context-aware answers when tied to validated clinical content, but requires careful curation of knowledge sources, rigorous safety filters, and clarity about medico-legal responsibility for advice given to vulnerable populations [7], [11].

2.2.6 Reminder systems and medication adherence

Meta-analyses consistently find that reminders like text messages, phone calls, and notifications help people take their medications more reliably in chronic conditions. These tools work best when they're interactive or involve a caregiver. For people with neurodegenerative diseases, it's important to tailor approaches, involve caregivers, and design with usability in mind for those with cognitive challenges. The literature also notes gaps in tracking long-term adherence and in studies that specifically assess acceptability among patients with cognitive deficits [5][11].

2.2.7 History management, EHRs and continuity of care

Longitudinal history management means structuring the storage of clinical notes, imaging results, medication history, and functional assessments to track disease progression and support decisions. Electronic health records (EHRs) and integrated patient portals enable data access and provider coordination, increasing patient engagement. Integrating AI outputs such as model results raises questions about provenance, interpretability, and clinician acceptance. The literature emphasizes interoperability, data quality, and user-interface design as central to realizing benefits from digitized histories in real-world practice [8].

2.2.8 Digital appointment and prescription management

Digital tools for scheduling, prescriptions, and telehealth improve access and care coordination, especially when paired with notification systems. Implementation barriers include user digital literacy, data privacy concerns, and workflows that may misalign with clinical staff; for older adults or those with cognitive impairments, caregivers often mediate interfaces [7]. Integrating appointment and prescription modules with diagnostic outputs and reminders can close the loop from detection to care delivery, if designed around clinicians' and caregivers' needs [11].

2.2.9 Critical gaps and justification for the current project

Taken together, the literature shows solid advances: CNNs can detect structural AD signatures in MRI; RAG methods improve the safety and accuracy of domain Q&A; reminder systems enhance adherence; and EHR-based history tracking supports longitudinal care. Yet there is a clear gap in integrated, end-to-end platforms that combine validated imaging AI, clinically informed conversational support with safety layers for RAG, role-based access, history tracking, appointment and prescription workflows, and caregiver-friendly notifications—all evaluated in realistic clinical scenarios. Most existing work focuses on a single component, with few studies addressing interoperability or outcomes meaningful to patients and caregivers (quality of life, caregiver burden, real-world clinical decisions). Remaining challenges include explainability, external validation of imaging models, data governance, and medico-

legal accountability for automated advice—especially when LLMs are involved [5][6][7][9].

2.2.10 How our FYP Addresses the gaps

Our project integrates an MRI staging model, a RAG-based chatbot for patient and caregiver queries, and a role-based web application containing appointment management, prescriptions, notification/reminder modules, and persistent history tracking. By routing MRI model outputs through clinician review before clinical decisions, grounding chatbot responses in curated sources, and designing caregiver-friendly notification flows, We aim to improve usability and safety. For evaluation we will prioritize external validation of the MRI model, usability testing with patients and caregivers, and impact metrics that matter clinically — such as adherence, time-taken for clinician action, and caregiver burden. This integrated approach moves beyond single-component work and aims to test an applied solution that could be embedded into clinical pathways.

2.3 User Classes and Characteristics

The user can be divided into three classes:

- Administrator
- Doctor
- Patient

Administrator

- Must employee in an organization such as in a hospital.
- Must know how to use the system.
- Must completely understand the system

Doctor

- Doctor must know how to use the system

Patient

- Patient can be in link with doctor.
- Patient must know how to use the system.

2.4 Operating Environment

Operating environment for AI-Assisted Caretaker system for Neurodegenerative disorder patients:

- **Operating System:** Windows 10, 11, Linux & MacOS supported
- **Web Browsers:** Google Chrome and Firefox Browsers supported because these are common
- **Database Management System:** MongoDB
- **API Testing:** Postman
- **Programming Language :** Javascript, Python and Flask Framework

2.5 Design and Implementation Constraints

Identification of Mild Cognitive Impairment Dementia is a web-based system therefore front-end will be developed using Next.js, Shadcn UI, Typescript, Tailwind CSS and back-end will be developed using Node.js, Flask framework and MongoDB. The project is only implemented to make web app so it might take time to open because of synchronizing database with internet on hardware. The internet connection is also a constraint for the application. Since the application fetches data from the database & huggingface over the internet, it is crucial that there is an internet connection for the application to function.

2.6 Assumptions and Dependencies

Assumptions and dependencies:

- PC (Personal Computer) or workstation with GUI.
- A web browser with support for cookies.
- Working internet connection.

2.7 External Interface Requirements

External interface requirements specify software or database elements with which a system or component must interface.

2.7.1 Software Interfaces

Software Interfaces:

Table 2.1: Software Interfaces

Modules	Description
Operating System	We have chosen Windows operating system for its best support and user friendliness.
Database	We have chosen MONGODB for this project
Front End	We have used Next.js ,Tailwind CSS, Shaden UI and JavaScript
Back End	We have used JWT, Node.js & Flask python framework for back end.

Table 2.2: Software

Modules	Software
Front End	Visual Studio Code
Back End	Visual Studio Code
API	Visual Studio Code
API Testing	Postman

2.7.2 Performance Requirements

Since this software is going to be a web based, it requires a high band internet access. High Speed internet access is required so that it can process the data entered.

2.7.3 Safety Requirements

There is a contact feature available where users can contact or report any issue they have encountered while using the web service so the developers can resolve those issues.

2.7.4 Security Requirements

The system should provide basic security features like ID and password authentication for Admin, Doctor and Patient.

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 Methodology

This project will be developed by using agile methodology because agile models handle change in requirements easily, divide complex projects into parts (iterations) which simplifies the project and deliver working software after every iteration. In agile, there are many models but here Feature Driven Development (FDD) will be used as it is one of the simplest and easy to follow agile methodology. FDD develop project in features and divide complex features into sub parts. This model repeats last two phases i.e. Design by feature and develop by feature until all features will complete.

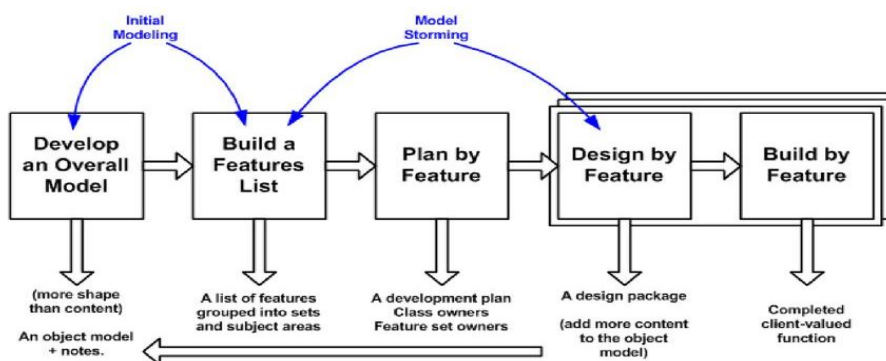


Figure 3.1: Feature Driven Development (FDD)

Following are the FDD phases that will be followed in this project.

3.1.1 Develop an Overall Model

In this step the scope of the project is defined, and a top-level architecture is developed by analyzing and identifying the phrases of the project.

3.1.2 Build a feature list

After the development of the overall model a feature list is drafted by analyzing and by fragmenting the functionality of the project.

3.1.3 Plan by feature

After drafting a feature list, next step is to draft a development plan of the features that are extracted in the previous step.

3.1.4 Design by feature

After planning, now comes the phase of designing by feature. In this phase following steps are followed.

- Create use case diagrams of each feature
- Sequence diagrams for each feature
- Collaboration diagram for each feature
- Design the user interface

3.1.5 Build by feature

The steps that are being followed for the development of the project feature by feature are as under:

- MRI Model Implementation
- Chatbot Implementation
- Web-Application Implementation
- API Implementation
- Integration
- Testing

3.2 System Design

System design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. It could be seen as the application of systems theory to product development.

3.3 Use Case Description

Description of all use cases of this project are written down in this section.

3.3.1 Use-case 1: Self-Register User

Table 3.1: UC Self-Register User

Name and ID	Self-Register
Primary Actor	User (Doctor or Patient)
Brief description	The main Objective of this use case describes that where a new user registers through the sign-up page.
Preconditions	<ol style="list-style-type: none"> 1. User has a web-browser. 2. User has web-application open. 3. User has browsed to Sign-up.
Basic flow or Happy path	<ol style="list-style-type: none"> 1. User opens the sign-up page. 2. User selects their role (Doctor / Patient). 3. System displays role-specific fields. 4. User fills out the form: <ul style="list-style-type: none"> • Full Name • Email • Password & confirm password • Profile image • Doctor fields (specialization, experience) Or • Patient fields (caretaker emergency contact) • Accept terms 5. User click sign-up 6. System validate data 7. System uploads the profile image to Vercel Blob. 8. System returns access message: <p style="text-align: center;">Registration successful! Please waiting for admin approval.</p> 9. User cannot log in until admin approval.
Trigger	User wants to create a new account.
Alternate flows	<ol style="list-style-type: none"> 1. Doctor missing specialization / experience 2. Patient missing emergency contact 3. System shows inline validation errors
Post conditions	<p>Success:</p> <ol style="list-style-type: none"> 1. User record stored with status (pending) 2. User receives conformation message 3. User cannot log in yet. 4. Admin sees the user under pending requests. <p>Failure:</p>

	<ul style="list-style-type: none"> • No User created. • No login possible
--	---

3.3.2 UC User Login

Table 3.2 UC user login

Name and ID	Login and ADN15
Primary Actor	Admin, Doctor, Patient
Brief description	Entertain the user to access the dashboard of the Application.
Preconditions	User must be registered.
Basic flow or Happy path	<ol style="list-style-type: none"> 1. Go to the login page. 2. Enter your Email and password to login. 3. Welcome to the Dashboard. 4. User is logged in to the system successfully.
Trigger	It is the event that causes use-case to be initiated.
Alternate flows	<ol style="list-style-type: none"> 1. If the Email and Password entered is incorrect then pop up message will appear.
Post conditions	User will be logged In to the Application and redirect to the Dashboard.

3.3.3 Use-case 3: Admin Login

Table 3.3: UC Admin Approve User

Name and ID	Admin-Approve-User, CALIX0
Primary Actor	Admin
Brief description	The main Objective of this use case is to approved or reject user request and also add new user i.e. Doctor or Patient.
Preconditions	Admin must have access rights to the Admin Dashboard
Basic flow or Happy path	<ol style="list-style-type: none"> 1. When Admin run application, redirects to admin panel then login to the system. 2. Admin enters their username and password and submit the form 3. System validate and redirects the admin to the Admin Dashboard 4. After login system display a list of users who signed up and are in pending status. 5. Admin chooses to approved or reject a user. 6. If approved, the system updates the user status to approved 7. If rejected, the system marks the user as rejected and access denies. 8. Admin may choose add new user, to manually register Doctor or patient. 9. Back to Dashboard/Home page. 10. Logout.
Trigger	Admin to manages the system by logging in and performing user operation i.e. approved, reject or registration
Alternate flows	<ol style="list-style-type: none"> 1. At step 2, if the admin enters incorrect login information, the system shows an error and remains in the login. 2. At any step, if Admin presses Cancel, the system returns to the Dashboard.
Post conditions	User has successfully login then redirect to the Dashboard/Home page.

3.3.4 Use-case 4: Patient Create Appointment

Table 3.4: UC User Create Appointment

Name and ID	Create Appointment, CALIX3
Primary Actor	Patient
Brief description	The objective of this use case is for a Patient to request an appointment with a doctor by selecting date and time and purpose of visit.
Preconditions	<ol style="list-style-type: none"> 1. Patient must be logged in into the system. 2. Patient must have an approved status (approved by Admin) 3. Approved Doctor list must available in the system.
Basic flow or Happy path	<ol style="list-style-type: none"> 1. When Patient logs into the system and navigates to the Appointments section. 2. Patient click the button Book New Appointment dialog. 3. Dialog display a list of available Doctors. 4. Dialog display available date and time slots for the selected Doctor. 5. Patient chooses the date, time, and reason for the appointment. 6. Patient confirms the appointment request. 7. System stores the appointment with status pending and notifies the Doctor. 8. Patients is redirected to the Appointment summery page
Trigger	Patient wants to book an appointment with a Doctor.
Alternate flows	<ol style="list-style-type: none"> 1. At step 3-4, if required fields are missing, the system shows validation error.
Post conditions	Appointment is successfully created and stored with status pending.

3.3.5 Use-case 5: Doctor Manage Appointment

Table 3.5: UC Doctor Manage Appointment

Name and ID	Doctor-Manage-Appointment, CALIX4
Primary Actor	Doctor
Brief description	The objective of this use case is to allow the Doctor to view, approve, reject or reschedule appointment requests made by patients.
Preconditions	<ol style="list-style-type: none"> 1. Doctor must be logged in into the system. 2. Doctor must have an approved status (approved by Admin)
Basic flow or Happy path	<ol style="list-style-type: none"> 1. Doctor logs into the system and navigates to the Appointments section. 2. System displays all pending, approved, rejected and reschedule appointments. 3. Doctor selects a pending appointment. 4. Doctor views appointment details (patient info, date, time, reason). 5. Doctor chooses one of the following options: 6. Approve appointment 7. Reject appointment 8. Propose new Date/Time (Reschedule) 9. System updates the appointments status based on doctor decision 10. System notifies the patient about the update status. 11. Doctor returns to the appointment page
Trigger	Doctor wants to manage patient appointment request.
Alternate flows	<ol style="list-style-type: none"> 2. At step 5, Doctor selects reject. System marks appointment as rejected and notifies the patient 3. If doctor enters invalid date/time during reschedule, system shows validation error
Post conditions	Appointment status is updated to approved, reject, or rescheduled.

3.3.6 Use-case 6: Patient Upload Profile Image

Table 3.3: UC Patient Upload Image

Name and ID	Upload Image, ADNI4
Primary Actor	Patient
Brief description	The objective of this use case is to generate the predicted stages of dementia of the current user.
Preconditions	User should be logged In to the Application.
Basic flow or Happy path	<ol style="list-style-type: none"> 1. User login to the Application. 2. After Login, a dashboard page will appear where user navigate between Menus and page. 3. Go to view profile option to see Profile. 4. Go to upload image option to add. 5. After uploading image the model will predict the dementia stages of the user. The model has been trained to generate the predicted result of the user. 6. Go to Query option and write any type of help/queries to admin. 7. Back to Dashboard/Home page. 8. Logout.
Trigger	It is the event that causes use-case to be initiated.
Alternate flows	<ol style="list-style-type: none"> 1. At step 7 of basic flow, if user had no images related to the MRI scan of the patient.
Post conditions	System perform the particular action/task and return to the dashboard/home.

3.3.7 Use-case 7: Patient MRI Upload Image

Table 3.7: UC Patient Upload MRI Image

Name and ID	Upload MRI Image, ADNI5
Primary Actor	Patient
Brief description	The objective of this use case is to generate the predicted stages of dementia of the patient.
Preconditions	Patient should be logged In to the Application.
Basic flow or Happy path	<ol style="list-style-type: none"> 1. Patient login to the Application. 2. After Login, a dashboard page will appear where user navigate between Menus and page. 3. Patient go to upload image tab 4. Patient click upload image button after click open dialog 5. Dialog show selects approved doctor and click to upload image to appropriate doctor 6. After this click to submit image. 7. System verified and submission successful 8. Back to upload image.
Trigger	Patient want to upload their report to appropriate doctor.
Alternate flows	<ol style="list-style-type: none"> 1. At step 5, if doctor not present in the list system show error doctor must be select
Post conditions	System performs the particular action/task and return to the dashboard/home.

3.3.8 Use-case 8: Alzheimer's Disease Chatbot Interaction

Table 3.8: UC Chatbot

Name and ID	Chatbot Interaction, ADNI6
Primary Actor	Patient / Doctor
Brief description	The objective of this use case is to describes how a Patient or Doctor interacts individually with the AI Alzheimer's Disease Support Chatbot.
Preconditions	Patient / Doctor should be logged In to the Application.
Basic flow or Happy path	<ol style="list-style-type: none"> 1. Patient / Doctor should be login to the Application. 2. User navigates to AI Alzheimer's chatbot Support. 3. User opens the chatbot interface. 4. User types and sends a medical or Alzheimer's-related query. 5. System loads the user's conversation history. 6. Chatbot analyse history + new input to produce an accurate response. 7. Response is displayed to the user.
Trigger	User Selects the AI Chat Support Option or sends the first message to the messages.
Alternate flows	<ol style="list-style-type: none"> 1. At step 5, if doctor not present in the list system show error doctor must be select
Post conditions	System performs the particular action/task and return to the dashboard/home.

3.3.9 Use-case 9: Doctor Create Prescription

Table 3.9: UC Doctor Create Prescription for Completed Appointment

Name and ID	Create Prescription , ADNI7
Primary Actor	Doctor
Brief description	The objective of this use case is to describes how a Patient or Doctor interacts individually with the AI Alzheimer’s Disease Support Chatbot.
Preconditions	<ol style="list-style-type: none"> 1. Doctor must be logged in. 2. Patient must have a completed appointment with the doctor. 3. Appointment must already exist in the system.
Basic flow or Happy path	<ol style="list-style-type: none"> 1. Doctor logs into the system. 2. Doctor navigates to the Appointments section. 3. System displays list of appointments with statuses: <ul style="list-style-type: none"> • Pending • Ongoing • Complete 4. Doctor selects a completed appointment. 5. System displays: <ul style="list-style-type: none"> • Patient details • Appointment summary • Symptoms or notes from patient • Attached reports if any (MRI, PDF, etc.) 6. Doctor clicks “Create Prescription” button. 7. System shows prescription form with: <ul style="list-style-type: none"> • Patient info (auto-filled) • Appointment reference number • Date/time of appointment 8. Doctor enters prescription details: <ul style="list-style-type: none"> • Medication name • Dosage • Frequency

	<ul style="list-style-type: none"> • Duration • Special instructions <p>9. Doctor submits the prescription.</p> <p>10. System validates & saves prescription linked to:</p> <ul style="list-style-type: none"> • Patient • Doctor • Appointment ID <p>11. System updates doctor's dashboard and patient's record.</p> <p>12. Patient receives notification: "New Prescription Added".</p>
Trigger	Doctor create prescription to the relevant appointment.
Alternate flows	<ul style="list-style-type: none"> • At step 4, Prescription can only be created after the appointment is completed. • Please fill all required fields
Post conditions	System performs the particular action/task and return to the prescription page

3.3.10 Use-case 10: User Logout

Table 3.4: UC User Logout

Name and ID	Logout and ADNI6
Primary Actor	Admin, Doctor, Patient
Brief description	Entertain the user to Logout of the Application.
Preconditions	User must Logged In to the System.
Basic flow or Happy path	<ol style="list-style-type: none"> 1. As user is logged in to the application. 2. Look for the logout button on the Dashboard. 3. Press this logout button to. 4. User will be logout of the system.
Trigger	It is the event that causes use-case to be initiated.
Alternate flows	<ol style="list-style-type: none"> 1. Until the logout button is not clicked, don't logout of the system.
Post conditions	User will be logout of the Application and redirect to the login page.

3.4 Use Case Diagram

3.4.1 System Use Case

System Use Case model for Alzheimer..

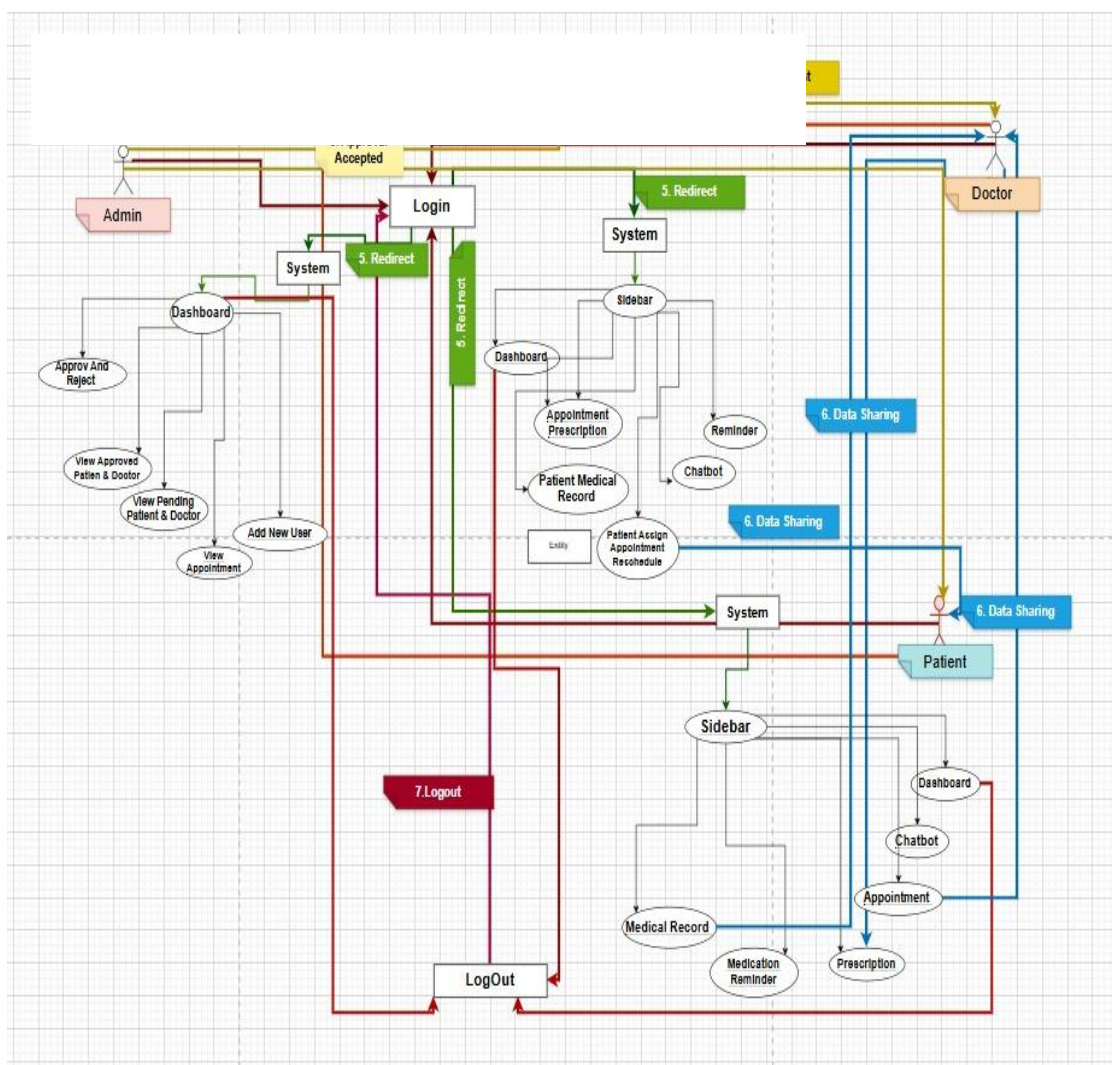


Figure 3.2: System Use Case

3.4.2 Doctor Use Case

Doctor Use Case for Alzheimer

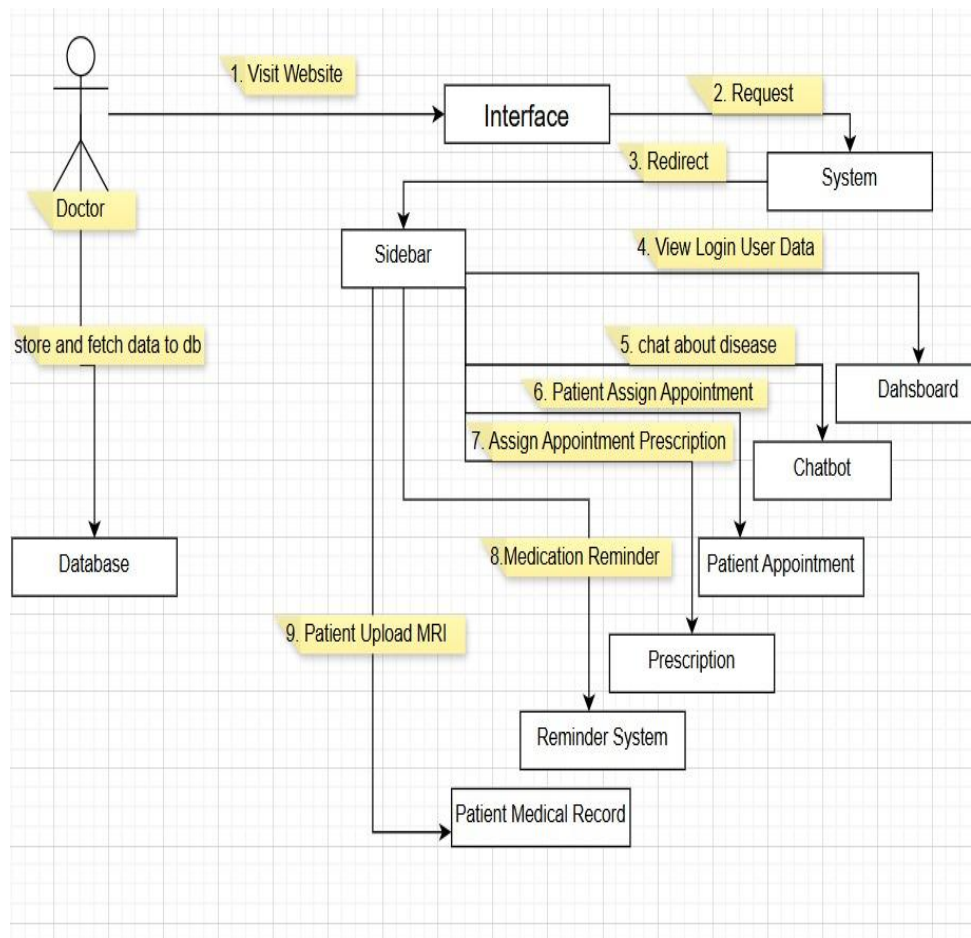


Figure 3.3: Doctor Use Case

Doctor Upload Image Use Case for Dementia.

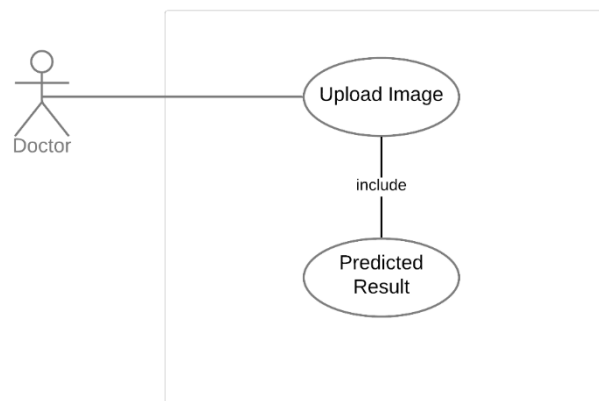


Figure 3.3: Doctor (Upload Image) Use Case

3.4.3 Patient Use Case

Patient Use Case for Dementia.

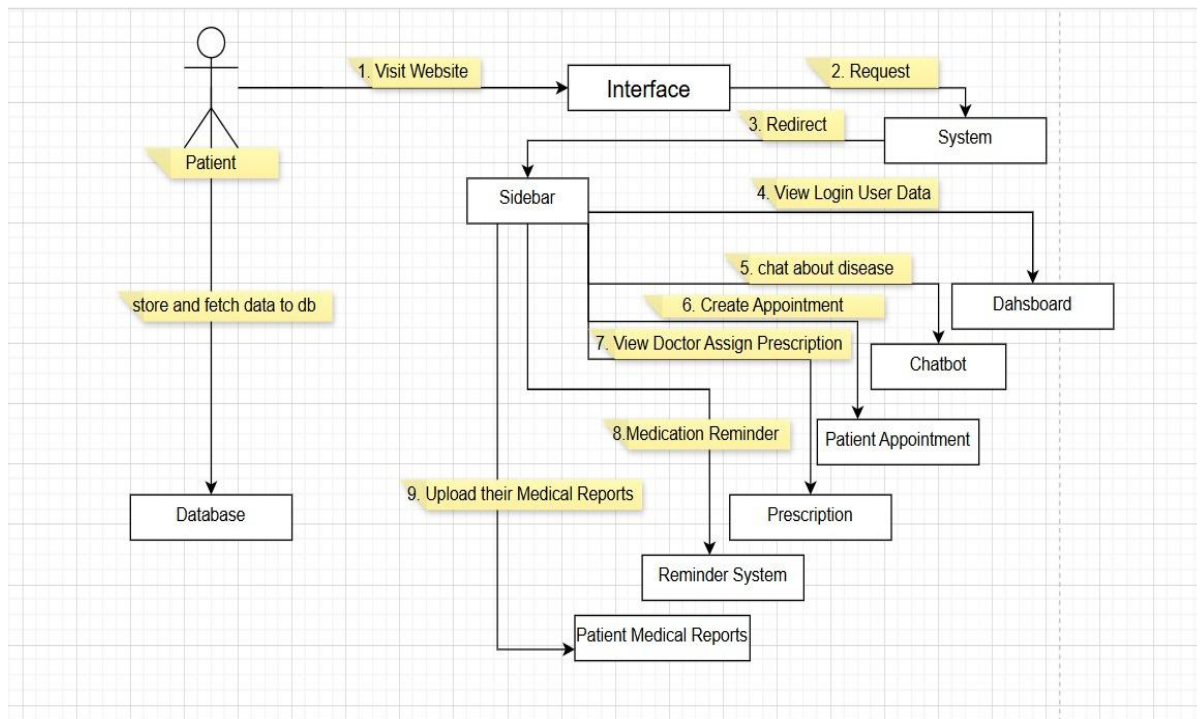


Figure 3.4: Patient Use Case

3.5 Sequence Diagram

A sequence diagram depicts the sequence of actions that occur in a system. The invocation of methods in each object, and the order in which the invocation occurs is captured in a sequence diagram. This makes the sequence diagram a very useful tool to easily represent the dynamic behavior of a system.

3.5.1 Admin (Profile / Help)

Admin (Profile / Help) Sequence Diagram for Alzheimer

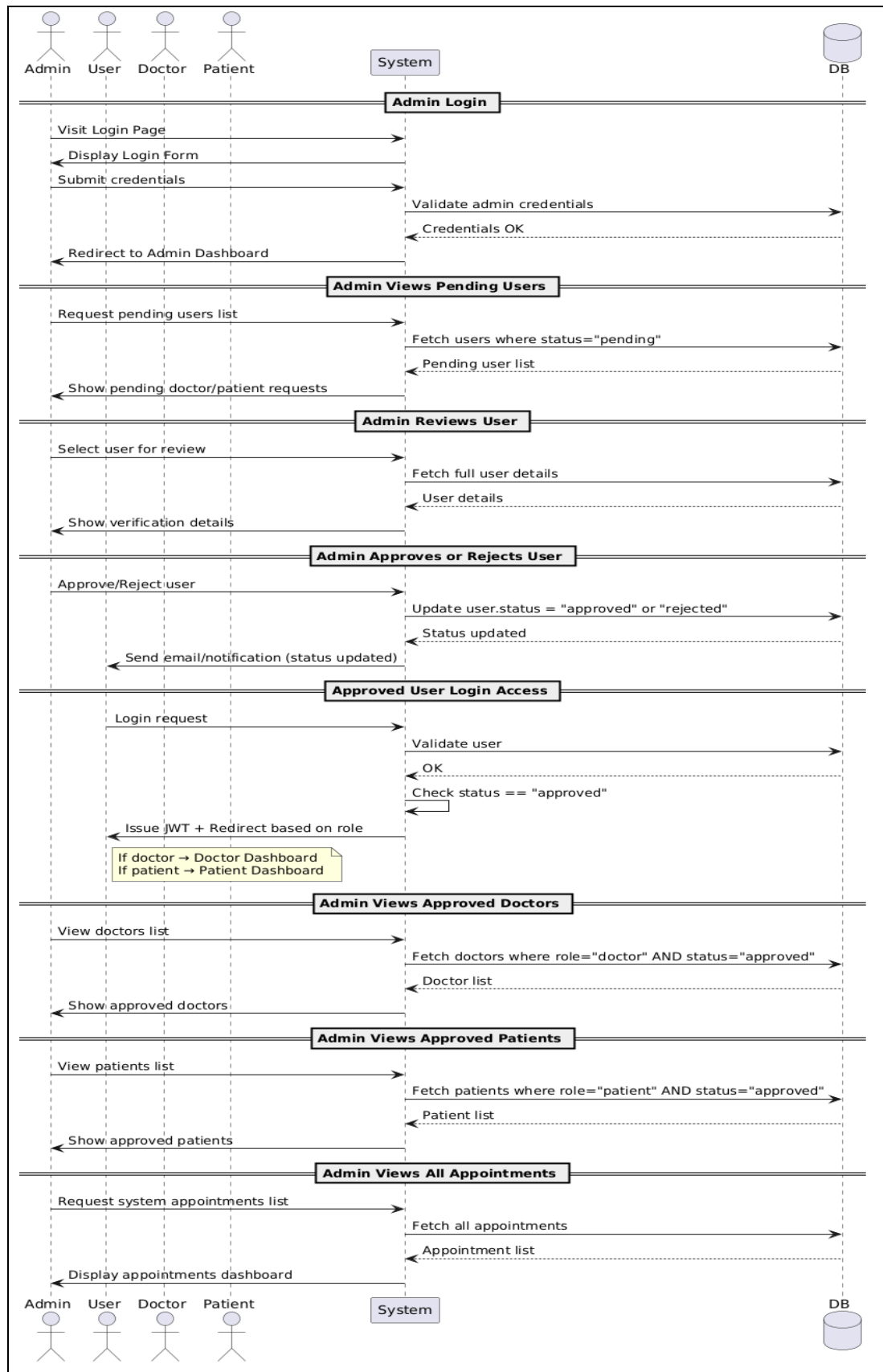
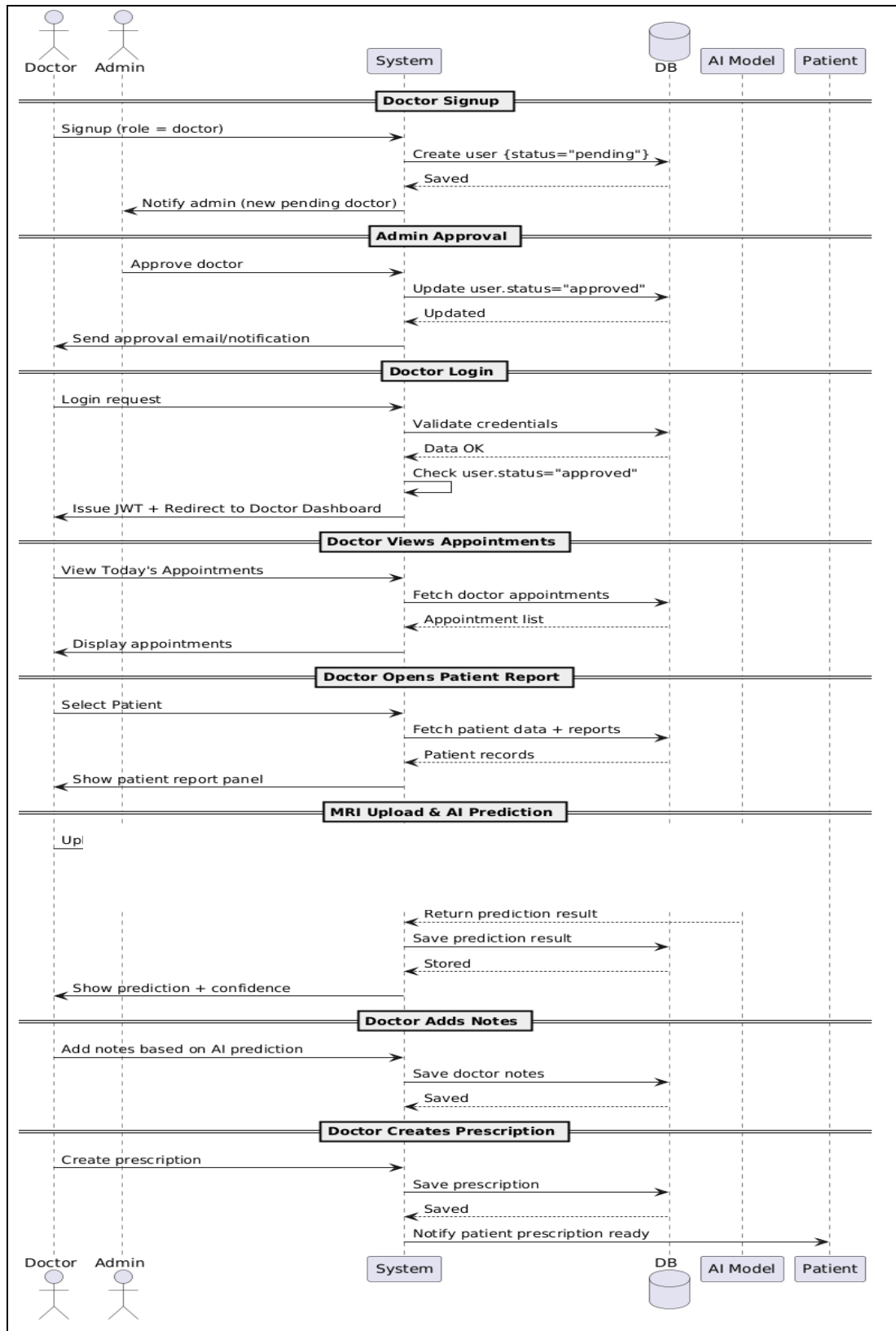


Figure 3.6: SD Admin (Profile / Help)

3.5.2 Doctor

Doctor Sequence Diagram for CALIX.

Figure 3.7: Doctor Sequence Diagram



3.5.3 Patient

Patient Sequence Diagram for Alzheimer.

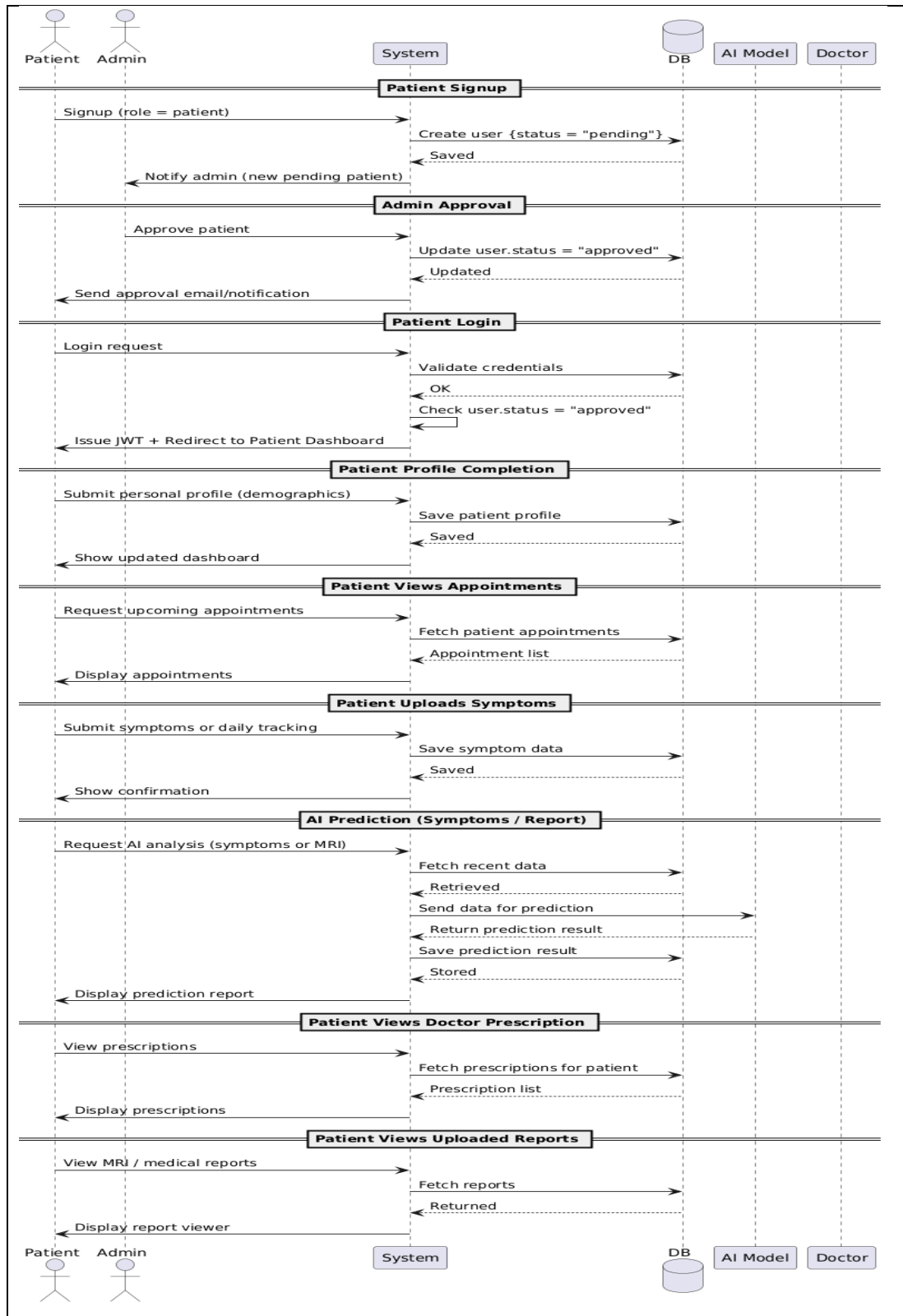


Figure 3.8: SD Patient

Logout

Logout Sequence Diagram for CALIX.

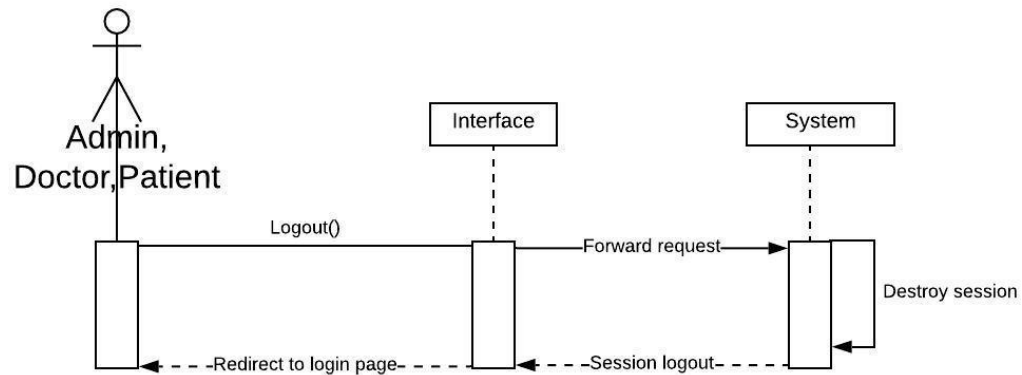


Figure 3.5: SD Logout

3.6 Collaboration Diagrams

A collaboration diagram describes a pattern of interaction among objects; it shows the objects participating in the interaction by their links to each other and the messages that they send to each other.

Collaboration diagrams are used to show how objects interact to perform

the behavior of a particular use-case, or a part of a use-case. Along with sequence diagrams, collaborations are used by designers to define and clarify the roles of the objects that perform a particular flow of events of a use-case. They are the primary source of information used to determining class responsibilities and interfaces.

3.6.1 Admin (Login)

Admin (Login) Collaboration Diagram for Dementia.

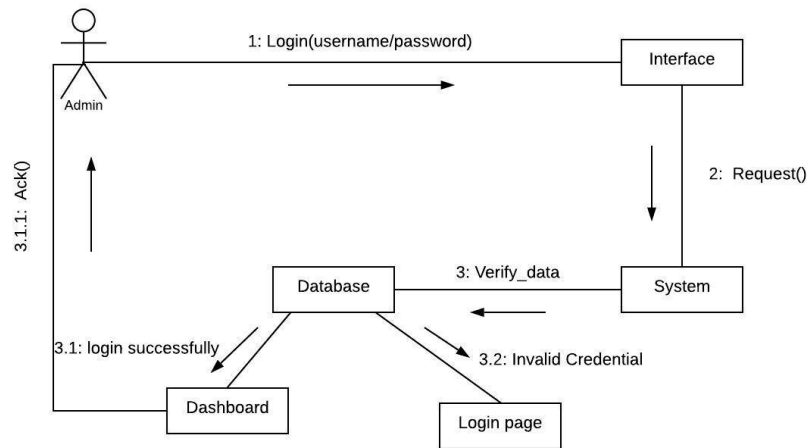


Figure 3.6: CD Admin

3.6.2 Admin Approve User

Admin Collaboration Diagram for CALIX.

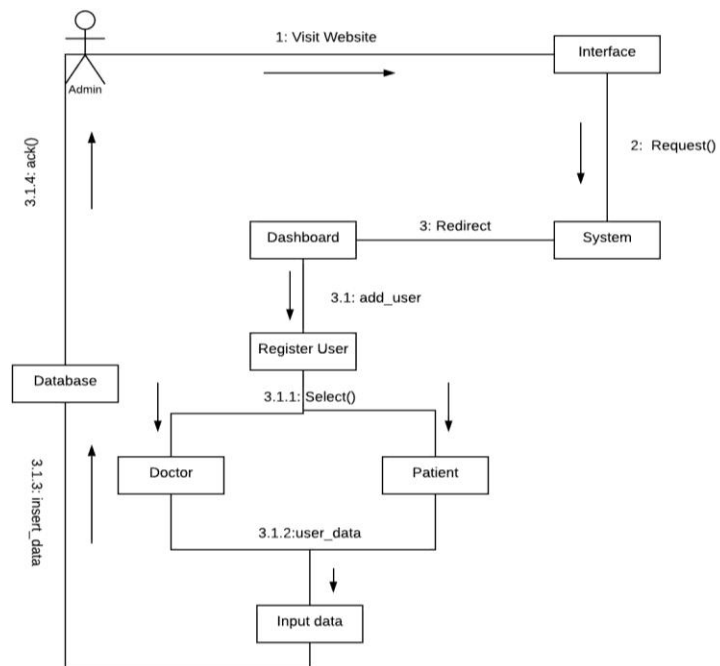


Figure 3.7: CD Admin Collaboration

3.6.3 Doctor/Patient (Login)

Doctor/Patient (Login) Collaboration Diagram for Calix.

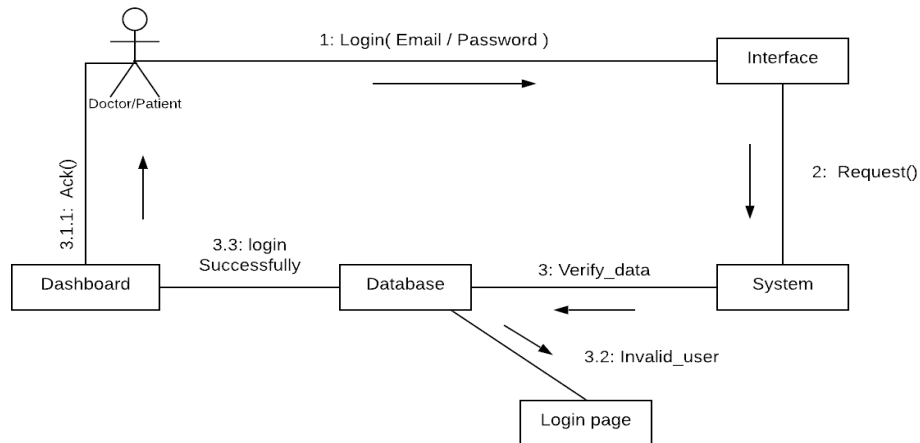


Figure 3.8: CD Doctor/Patient (Login)

3.6.4 Doctor Queries System

Doctor (Part 1) Collaboration Diagram for Calix.

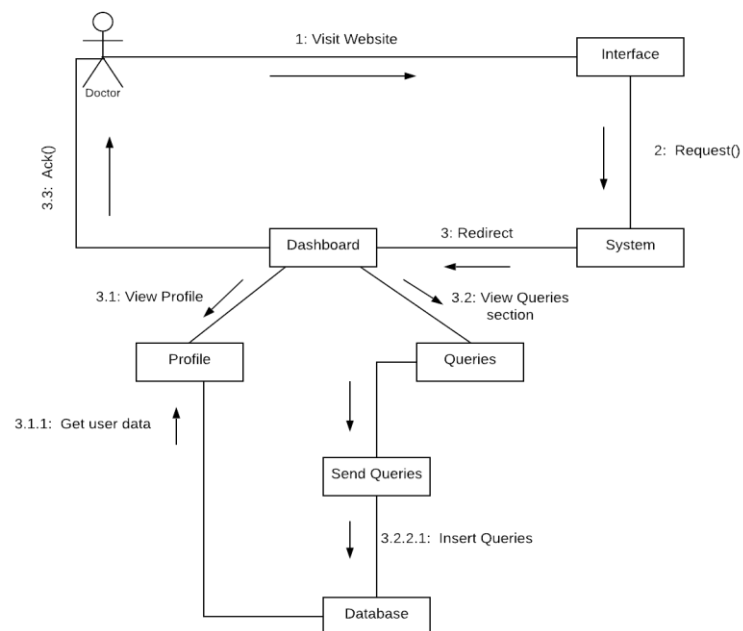


Figure 3.9: CD Doctor Query System

3.6.5 Doctor MRI Analysis

Doctor (Part 2) Collaboration Diagram for Calix.

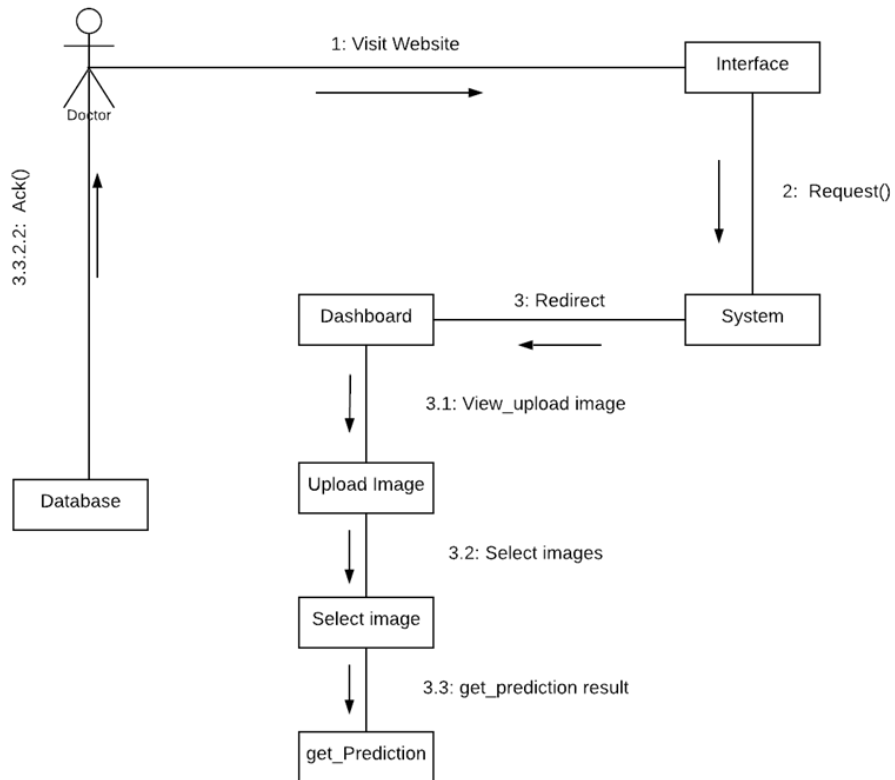


Figure 3.10: CD Doctor MRI Analysis

3.6.6 Patient Interacts System

Patient Collaboration Diagram for Calix.

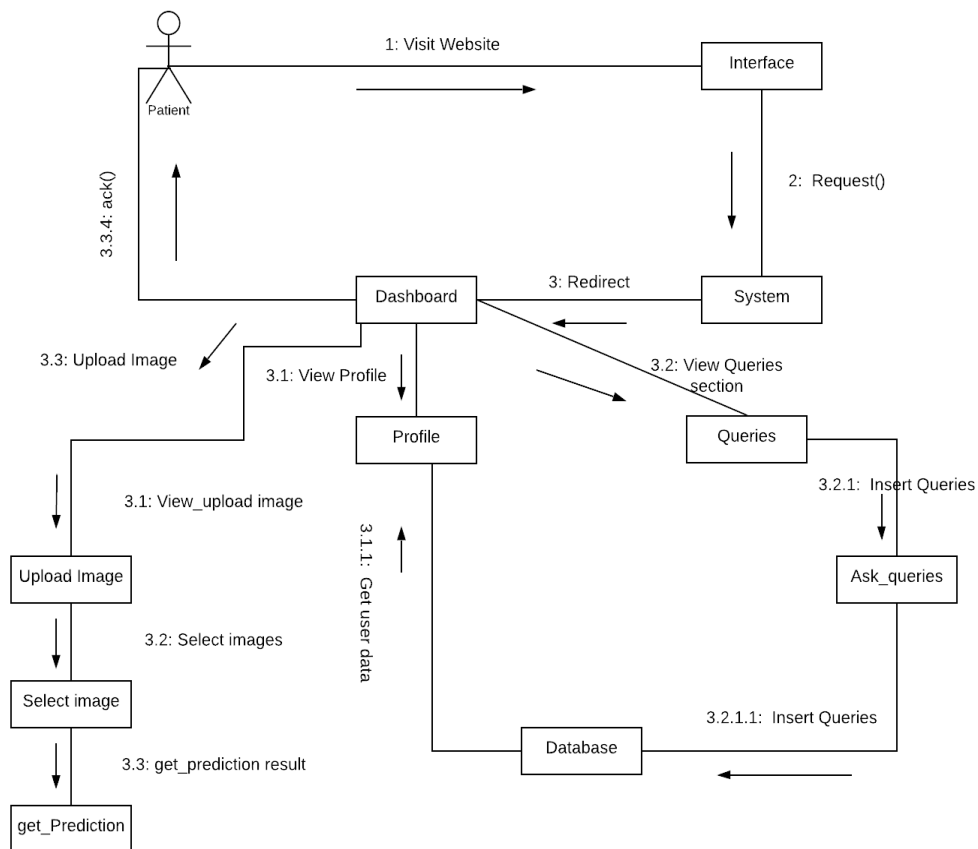


Figure 3.11: CD Patient Interacts System

3.6.7 User Logout

Logout Collaboration Diagram for Calix.

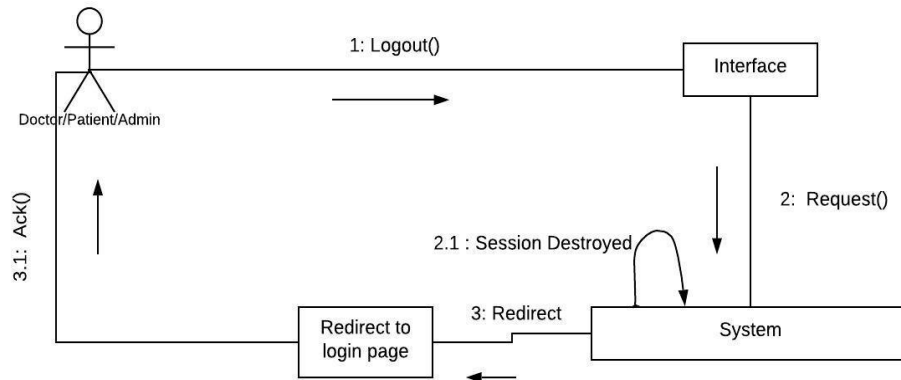


Figure 3.12: CD User Logout

3.7 Operation Contracts

A UML operation contract identifies system state changes when an operation happens. Effectively, it will define what each system operation does. An operation is taken from a system sequence diagram. It is a single event from that diagram. A domain model can be used to help generate an operation contract.

3.7.1 User Login

Login Operation Contracts for CALIX.

Table 3.5: OC User Login

Name:	Login
Responsibilities:	Admin, Doctor, Patient should be able to login.
Cross Reference:	Login use case will be used for this.
Exceptions:	None.
Pre-Condition:	Admin, Doctor, Patient must be registered first.
Post-Condition:	Admin, Doctor, Patient redirect to dashboard.

3.7.2 User Logout

Logout Operation Contracts for CALIX.

Table 3.6: OC User Logout

Name:	Logout
Responsibilities:	Admin, Doctor, Patient should be able to login.
Cross Reference:	Login use case will be used for this.
Exceptions:	None.
Pre-Condition:	Admin, Doctor, Patient must be login to the System.
Post-Condition:	Admin, Doctor, Patient redirect to login page and received an acknowledgment that the logout has successful.

3.7.3 Patient OPs

Patient Operation Contracts for CALIX.

Table 3.7: OC Patient

Name:	Patient
Responsibilities:	Patient create appointment, view prescription, use chatbot for question answering about their deases and upload MRI image
Cross Reference:	Patient use case will be used for this.
Exceptions:	None.
Pre-Condition:	Patient must be login and must be in dashboard.
Post-Condition:	Navigate between menus and pages or also download their report.

3.7.4 Doctor OPs

Doctor Operation Contracts for CALIX.

Table 3.8: OC Doctor

Name:	Doctor
Responsibilities:	Doctor upload MRI image into model and get Prediction report and also ask for query if needed and make appointment prescription and then reschedule appointment and then medication reminder send to the patient and also interact with chatbot about patient disease.
Cross Reference:	Doctor use case will be used for this.
Exceptions:	None.
Pre-Condition:	Doctor must be login and must be in dashboard.
Post-Condition:	Navigate between menus and pages or also can send report to patient.

3.7.5 Admin (Update/Delete User)

Admin (Update/Delete User) Operation Contracts for Dementia.

Table 3.9: OC Admin

Name:	Admin (Update/Delete User)
Responsibilities:	Admin get user request and admin have to right either approve nor reject the user request.
Cross Reference:	Admin (Delete User) use case will be used for this.
Exceptions:	None.
Pre-Condition:	Admin must be login and must be in dashboard.
Post-Condition:	Admin received acknowledgement whether user are Updated/deleted or not.

3.7.6 Admin (Register User)

Admin (Register User) Operation Contracts for Dementia.

Table 3.10: OC Admin (Register User)

Name:	Admin (Register User)
Responsibilities:	Navigate between menus and pages or Navigate to the option to Add user either Doctor or Patient.
Cross Reference:	Admin (Register User) use case will be used for this.
Exceptions:	None.
Pre-Condition:	Admin must be login and must be in dashboard.
Post-Condition:	Admin received acknowledgement whether user are registered or not.

3.8 Design Class Diagram

Classes are the work-horses of the design effort—they actually perform the real work of the system. The other design elements—subsystems, packages and collaborations simply describe how classes are grouped or how they interoperate.

Capsules are also stereotyped classes, used to represent concurrent threads of execution in real-time systems. In such cases, other design classes are 'passive' classes, used within the execution context provided by the 'active' capsules. When the software architect and designer choose not to use a design approach based on capsules, it is still possible to model concurrent behavior using 'active' classes.

Active classes are design classes, which coordinate and drive the behavior of the passive classes - an active class is a class whose instances are active objects, owning their own thread of control.

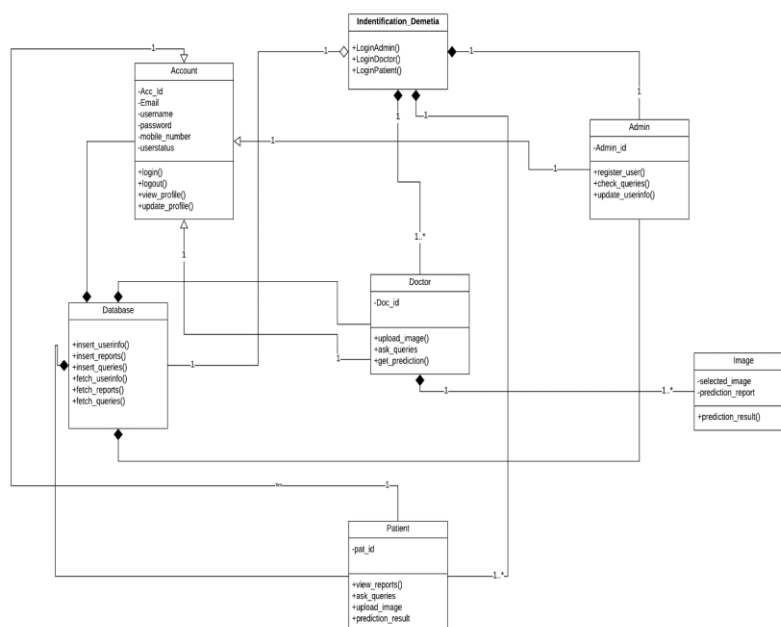


Figure 3.13: Class Diagram

3.9 Data Model

Data Model for Identification of Mild Cognitive Impairment Dementia.

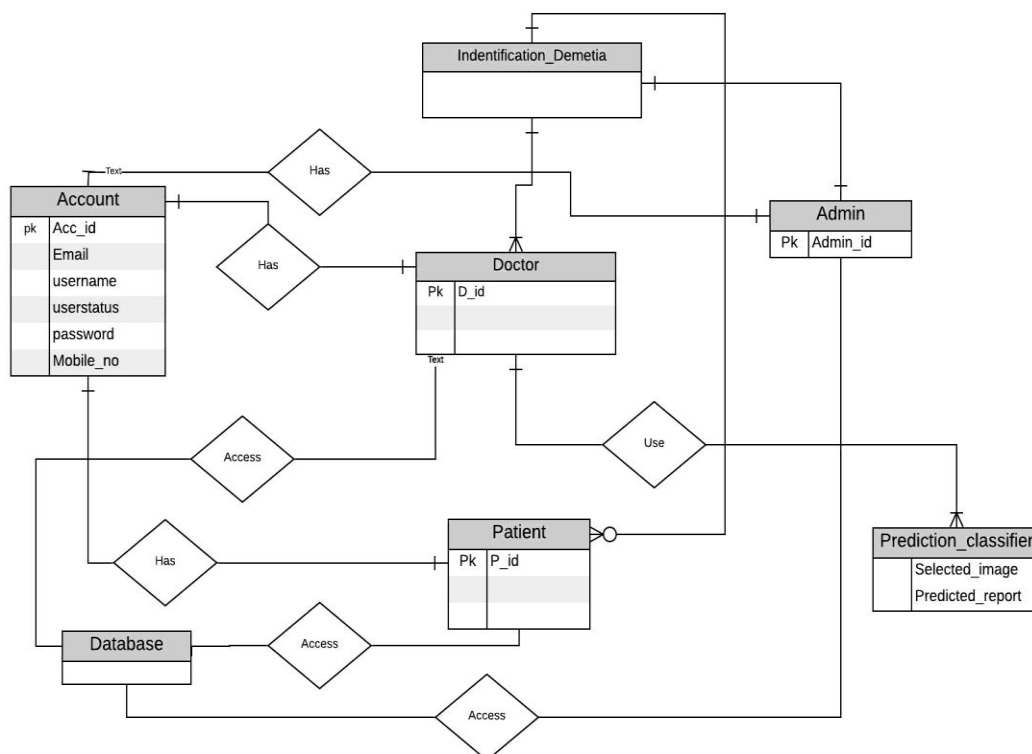


Figure 3.14: Data Model

CHAPTER 4

DATA AND IMPLEMENTATION

4.1 Software Development Methodology

In this project we used the following software development methodology to implement the system.

- MRI Model Implementation
- Chatbot Implementation
- Web-Application Implementation
- APIs Implementation
- Integration
- Testing
- Deployment

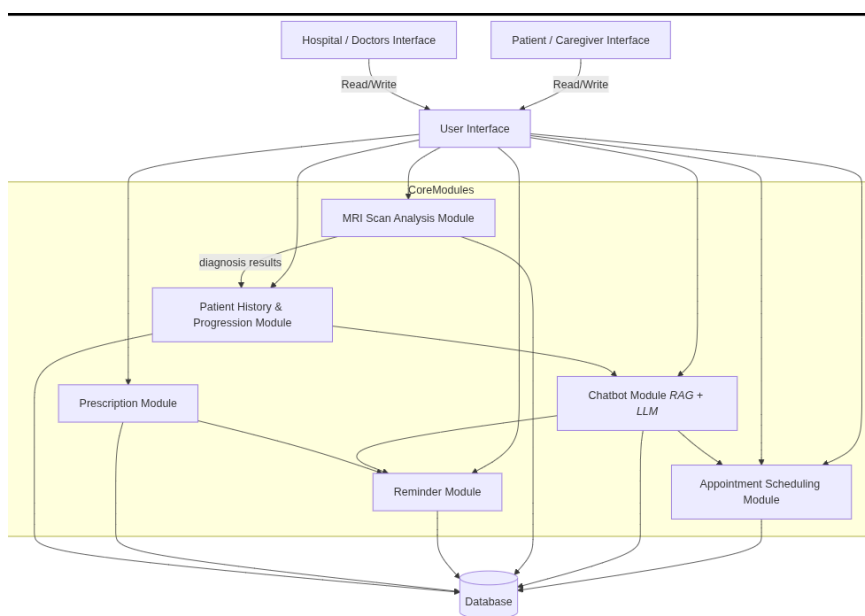


Figure 4.1: Top Level Architecture

4.2 MRI Model Implementation

This section includes the implementation details of MRI Scan classifier Model.

4.2.1 Dataset

In this project we use the “Augmented Alzheimer MRI” dataset from kaggle for training MRI scan classifier model. Important features about dataset are described below:

- 33984 MRI scans images of brain
- Dataset was labeled
- Dataset consist of 4 classes (**Mild demented, Moderate demented, Nondemented, Very Mild Demented**)

The dataset was labeled and consist of four classes. 70% of the dataset was used for training and the remaining 20% for validation & 10% for testing the trained model.

4.2.2 Model Construction (Architecture of CNN)

A Convolutional Neural Network is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. Convolutional neural networks have been known to produce admissible results for multi-classification of large image datasets and other pattern recognition problem.

We used four convolutional layers and applied reLU activation function and one fully connected hidden layer and here also we used reLU activation function. Batch Normalization, **Global Average Pooling (GAP)** & dropout layers are the additional non-linear / regularization layers present. In output layer we used SoftMax activation function.

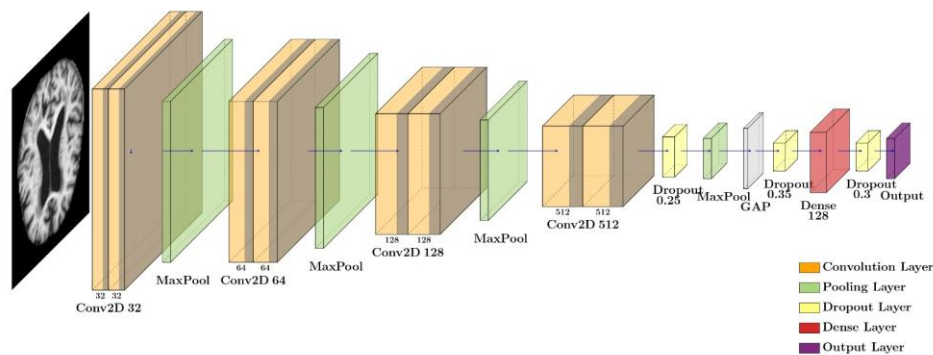


Figure 4.2: Architecture of CNN

4.2.3 Model Detail

This section include the Model details.

Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, 64, 64, 3)	0
conv2d (Conv2D)	(None, 64, 64, 32)	896
batch_normalization (BatchNormalization)	(None, 64, 64, 32)	128
max_pooling2d (MaxPooling2D)	(None, 32, 32, 32)	0
conv2d_1 (Conv2D)	(None, 32, 32, 64)	18,496
batch_normalization_1 (BatchNormalization)	(None, 32, 32, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 16, 16, 64)	0
conv2d_2 (Conv2D)	(None, 16, 16, 128)	73,856
batch_normalization_2 (BatchNormalization)	(None, 16, 16, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 8, 8, 128)	0
conv2d_3 (Conv2D)	(None, 8, 8, 512)	590,336
dropout (Dropout)	(None, 8, 8, 512)	0
max_pooling2d_3 (MaxPooling2D)	(None, 4, 4, 512)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 512)	0
dropout_1 (Dropout)	(None, 512)	0
dense (Dense)	(None, 128)	65,664
dropout_2 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 4)	516

Figure 4.3: Model Summary

4.2.4 Training and Testing

Train, Validate & Test Data samples

Total Number of training images: 23788

Total Number of validation images: 6796

Total Number of Test images: 3400

```
[OK] Using source directory: /content/alzheimer_data/AugmentedAlzheimerDataset
[INFO] OUTPUT_DIR already exists; new files will be added: /content/alzheimer_data/AugmentedAlzheimerDataset/split_AD
Class 'MildDemented': total=8960 → train=6272, val=1792, test=896 (errors=0)
Class 'ModerateDemented': total=6464 → train=4524, val=1292, test=648 (errors=0)
Class 'NonDemented': total=9600 → train=6720, val=1920, test=960 (errors=0)
Class 'VeryMildDemented': total=8960 → train=6272, val=1792, test=896 (errors=0)
Class 'split_AD': total=0 → train=0, val=0, test=0 (errors=0)

TOTALS:
source images: 33984
train: 23788 val: 6796 test: 3400

Split completed. Output directory structure:
/content/alzheimer_data/AugmentedAlzheimerDataset/split_AD
- train/<class>/*
- val/<class>/*
- test/<class>/*
```

Figure 4.4: Train and Test Data samples

4.2.5 Training Information

Total number of epochs = 40

Steps per epoch = 2974

```
Epoch 1/40
2973/2974 ██████████ 0s 108ms/step - accuracy: 0.4196 - loss: 1.2247
Epoch 1: val_accuracy improved from -inf to 0.60050, saving model to simple_cnn_runs/lr_5e-04_epochs_40/checkpoints/best_val_acc.keras
[Epoch 1] train_acc=0.9600 val_acc=0.9867 test_acc=0.9866666666666667 lr=None time=344.6s
2974/2974 ██████████ 347s 116ms/step - accuracy: 0.4197 - loss: 1.2246 - val_accuracy: 0.6005 - val_loss: 0.8025 - learning_rate
Epoch 2/40
2973/2974 ██████████ 0s 108ms/step - accuracy: 0.6405 - loss: 0.7647
Epoch 2: val_accuracy improved from 0.60050 to 0.69600, saving model to simple_cnn_runs/lr_5e-04_epochs_40/checkpoints/best_val_acc.keras
[Epoch 2] train_acc=0.8867 val_acc=0.8800 test_acc=0.9066666666666666 lr=None time=341.2s
2974/2974 ██████████ 343s 115ms/step - accuracy: 0.6405 - loss: 0.7647 - val_accuracy: 0.6960 - val_loss: 0.6502 - learning_rate
Epoch 3/40
2973/2974 ██████████ 0s 108ms/step - accuracy: 0.6889 - loss: 0.6808
Epoch 3: val_accuracy improved from 0.69600 to 0.73573, saving model to simple_cnn_runs/lr_5e-04_epochs_40/checkpoints/best_val_acc.keras
[Epoch 3] train_acc=0.8667 val_acc=0.8200 test_acc=0.86 lr=None time=340.2s
2974/2974 ██████████ 342s 115ms/step - accuracy: 0.6889 - loss: 0.6808 - val_accuracy: 0.7357 - val_loss: 0.5821 - learning_rate
Epoch 4/40
2974/2974 ██████████ 0s 108ms/step - accuracy: 0.7234 - loss: 0.6012
Epoch 4: val_accuracy improved from 0.73573 to 0.77192, saving model to simple_cnn_runs/lr_5e-04_epochs_40/checkpoints/best_val_acc.keras
[Epoch 4] train_acc=0.8267 val_acc=0.7733 test_acc=0.7733333333333333 lr=None time=341.1s
2974/2974 ██████████ 343s 115ms/step - accuracy: 0.7234 - loss: 0.6012 - val_accuracy: 0.7719 - val_loss: 0.5008 - learning_rate
Epoch 5/40
2974/2974 ██████████ 0s 108ms/step - accuracy: 0.7594 - loss: 0.5340
Epoch 5: val_accuracy improved from 0.77192 to 0.77693, saving model to simple_cnn_runs/lr_5e-04_epochs_40/checkpoints/best_val_acc.keras
[Epoch 5] train_acc=0.7133 val_acc=0.7133 test_acc=0.7666666666666667 lr=None time=340.5s
2974/2974 ██████████ 342s 115ms/step - accuracy: 0.7594 - loss: 0.5340 - val_accuracy: 0.7769 - val_loss: 0.4897 - learning_rate
...
Epoch 18: val_accuracy did not improve from 0.94026
[Epoch 18] train_acc=0.9533 val_acc=0.9067 test_acc=0.9133333333333333 lr=None time=338.2s
2974/2974 ██████████ 340s 114ms/step - accuracy: 0.9293 - loss: 0.1803 - val_accuracy: 0.9083 - val_loss: 0.2546 - learning_rate
Epoch 19/40
595/2974 ██████████ 4:15 107ms/step - accuracy: 0.9374 - loss: 0.1706
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Figure 4.2.5: Training Information

4.2.6 Training Accuracy

Training Accuracy: 0.9663

Validation Accuracy: 0.9766

Training Loss: 0.1654

Validation Loss: 0.035

4.3 Chatbot Implementation

This section includes the details about the implementation of RAG Chatbot.

4.3.1 Dataset

For the effective implementation of chatbots we need a lot of data, which is often called corpus. For that reason, we combined and use data from multiple datasets from kaggle.

The names of datasets are listed as follows

- | | |
|--|-----------------------|
| 1. ahmedashrafahmed/alzhemers-chat-dataset | 15833 unique values |
| 2. ahmedashrafahmed/alzheimer-chat-leader | 21244 unique values |
| 3. thedevastator/nlp-mental-health-conversations | 2480 unique values |
| 4. narendrageek/mental-health-faq-for-chatbot | 1590140 unique values |

4.3.2 Key Components

The SWEN chatbot system comprises several interconnected components designed to deliver contextually aware responses from domain-specific knowledge sources. The application is built as a Flask-based API that integrates multiple specialized services and models. Following are the key components of SWEN chatbot system

1. LLM Provider: **ChatGroq** (LangChain integration)
2. Embeddings: **HuggingFace** Embeddings (sentence-transformers/all-MiniLM-L6-v2)
3. Vector Store: **ChromaDB** (presistent)
4. Document Loaders (CSV, TXT, PDF)
5. Splitting: **RecursiveCharacterTextSplitter**

4.3.3 Working

The SWEN chatbot operates through a multi-stage pipeline that transforms user queries into informed responses grounded in domain knowledge.

When user submit a query, the system first pass the message through a history-aware retriever that reformulates the question into a standalone query using the conversation history. This contextualization step helps ensure that follow-up questions and pronouns are properly resolved against prior exchanges. The reformulated question is then used to retrieve semantically similar documents from the Chroma vector store,

with the retriever selecting the five most relevant document chunks based on embedding similarity.

The retrieved documents are combined with the user's original question and fed into a document-answering chain. This chain is powered by the llama-3.1-8b-instant language model, which has been guided by a system prompt to generate concise, empathetic responses that remain grounded in the provided context. The system prompt constrains responses to a maximum of three sentences and explicitly instructs the model to avoid speculation beyond the retrieved knowledge base.

The complete interaction chain is wrapped in a `RunnableWithMessageHistory` layer, enabling the system to maintain session-specific conversation memory and retrieve relevant history when processing new queries. The response returned to the user includes both the generated answer and source citations indicating which documents informed the response, providing transparency and enabling users to verify the information's origin.

4.3.4 Architecture Diagram

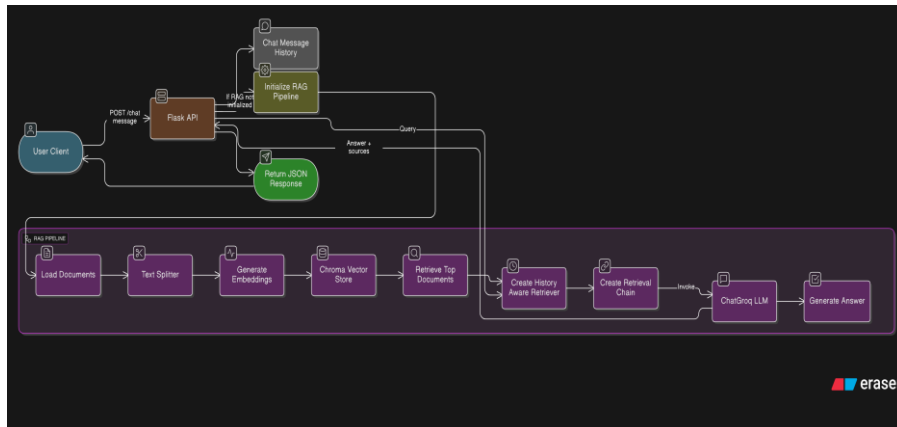


Figure 4.3.4: Swen (Chatbot) Architecture Diagram

4.4 Web-Application

This project implements the web-application utilizing following technologies:

Front-end:

- Next.js
- Shadcn/ui
- Typescript

- Tailwind CSS

Back-end:

Next.js full stack framework with MongoDB

That includes working

- Appointment System
- Prescription Table
- Patient History Module

4.5 APIs Implementation

In this phase we implemented APIs for Swen (chatbot) & MRI scan analyzer model using Flask.

4.6 Integration

In this phase we integrated Swen & MRI Scan Analyzer into the web-application.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 MRI Model Results

MRI model consists of 4 convolutional Blocks and a hidden layer. We used ReLU as activation function in the layers, and SoftMax for final classification layer.

Epochs	LR	Training Accuracy	Validation Accuracy	Validation Loss
1	5e-4	0.43	0.52	0.9937
10	5e-4	0.7878	0.8147	0.4267
20	5e-4	0.881453	0.8605	0.13
40	5e-4	0.9766670	0.98666	0.057

Accuracy:

Classification accuracy is defined as the number of correctly classified patterns to the total number of patterns.

$$(TP+TN) / (P+N)$$

Precision:

Precision is the fraction of relevant instances among the retrieved instances.

$$TP / (TP+FP)$$

Recall:

Recall is the fraction of the total amount of relevant instances that were actually retrieved.

$$TP/P$$

F-Score:

It is the harmonic mean (average) of the precision and recall.

$$2*(Recall * Precision) / (Recall + Precision)$$

Various measures such as accuracy, recall and precision are derived from the confusion matrix. These values help evaluate the model.

5.2 Confusion Matrix

Confusion matrix describes the performance of a classification model on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm. The confusion matrix is a four by four table that contains nine outcomes produced by a multiclass classifier.

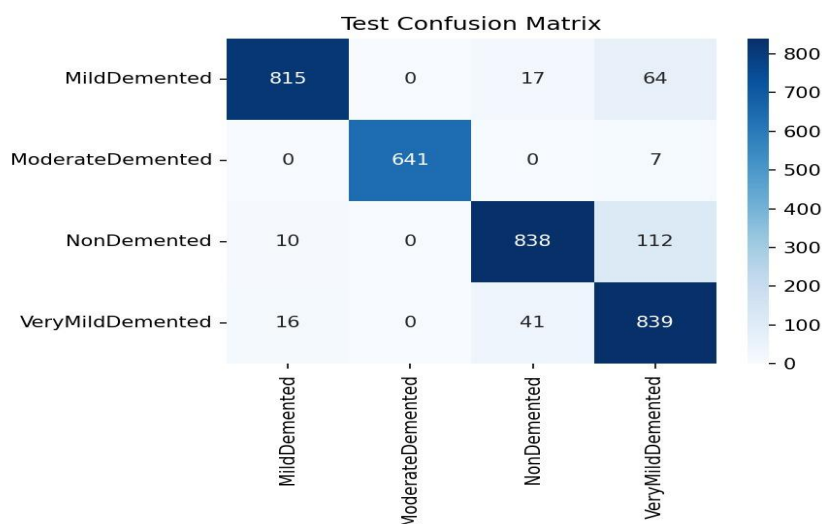


Figure 5.1: Confusion Matrix of CNN Model

True Positive:

A true positive test result is one that detects the condition when the condition is present.

False Positive:

A test result which wrongly indicates that a particular condition or attribute is present.

True Negative:

A true negative is an outcome where the model correctly predicts the negative class.

False Negative:

A false negative is a test result that indicates a person does not have a disease or condition when the person actually does have it.

5.3 Graphical User Interface

This section includes GUI of our application.

The Figure below represents the Home page of the user interface, this page can be access by any user i.e. Doctor, Patient or Admin.

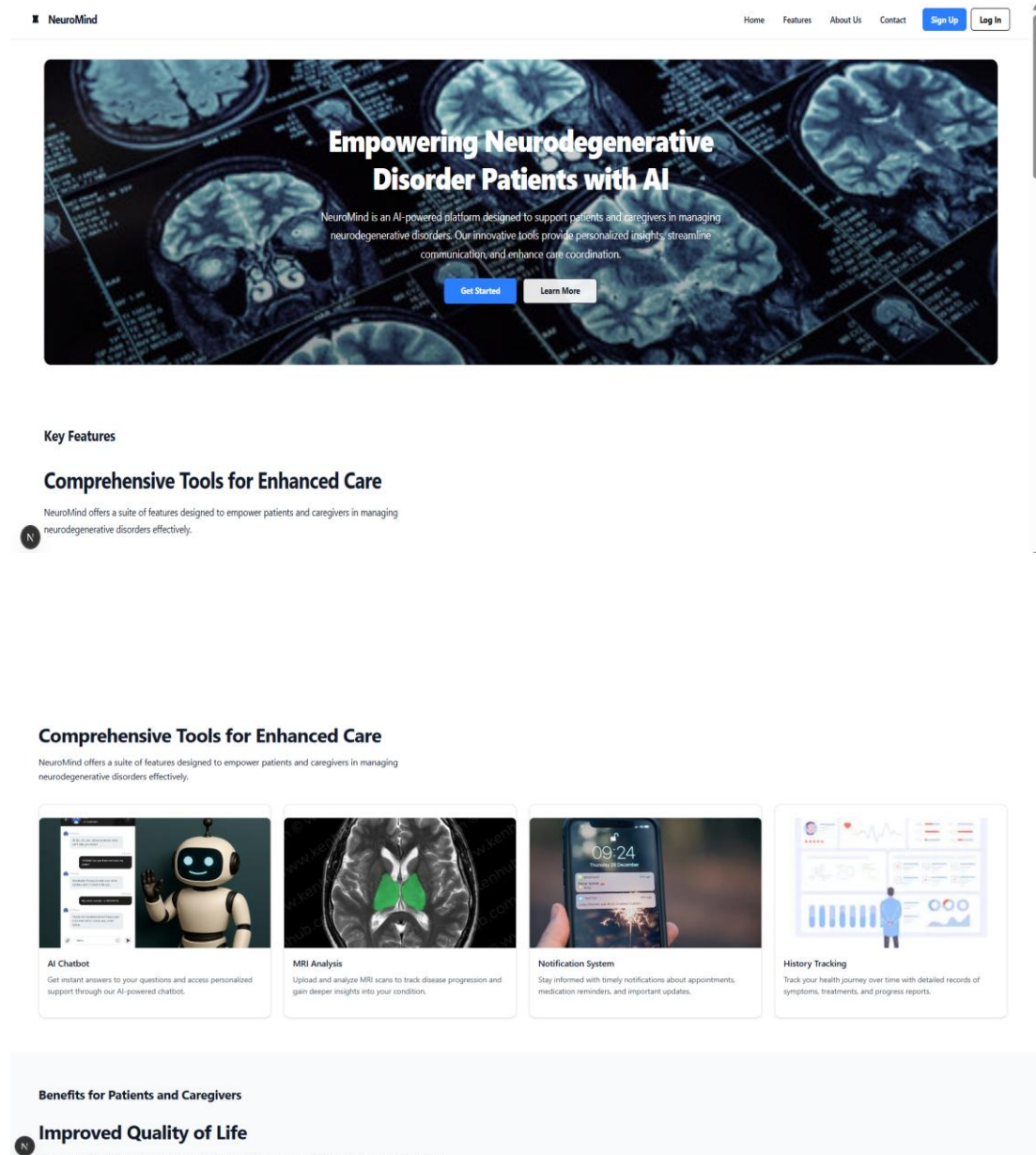


Figure 5.3.1: Landing Page UI

Doctor/Patient Sign-up page.

The screenshot shows a sign-up page for NeuroCare. At the top, it says "Welcome to NeuroCare" and "Sign in to continue to your account". Below this, there are two buttons for "Doctor" and "Patient". The "Doctor" button is selected. The form fields are: Full Name (placeholder: Enter your full name), Email (placeholder: you@example.com), Specialization (placeholder: e.g. Neurologist), Experience (years) (placeholder: 5), Password (placeholder:), Confirm Password (placeholder:), and Profile Picture (placeholder: Click or drag an image here (Max size: 5MB)). At the bottom, there is a checkbox for "I agree to the Terms of Service and Privacy Policy" and a blue "Sign Up" button.

Welcome to NeuroCare
Sign in to continue to your account

Select your role

Doctor **Patient**

Full Name
Enter your full name

Email
you@example.com

Specialization
e.g. Neurologist

Experience (years)
5

Password
.....

Confirm Password
.....

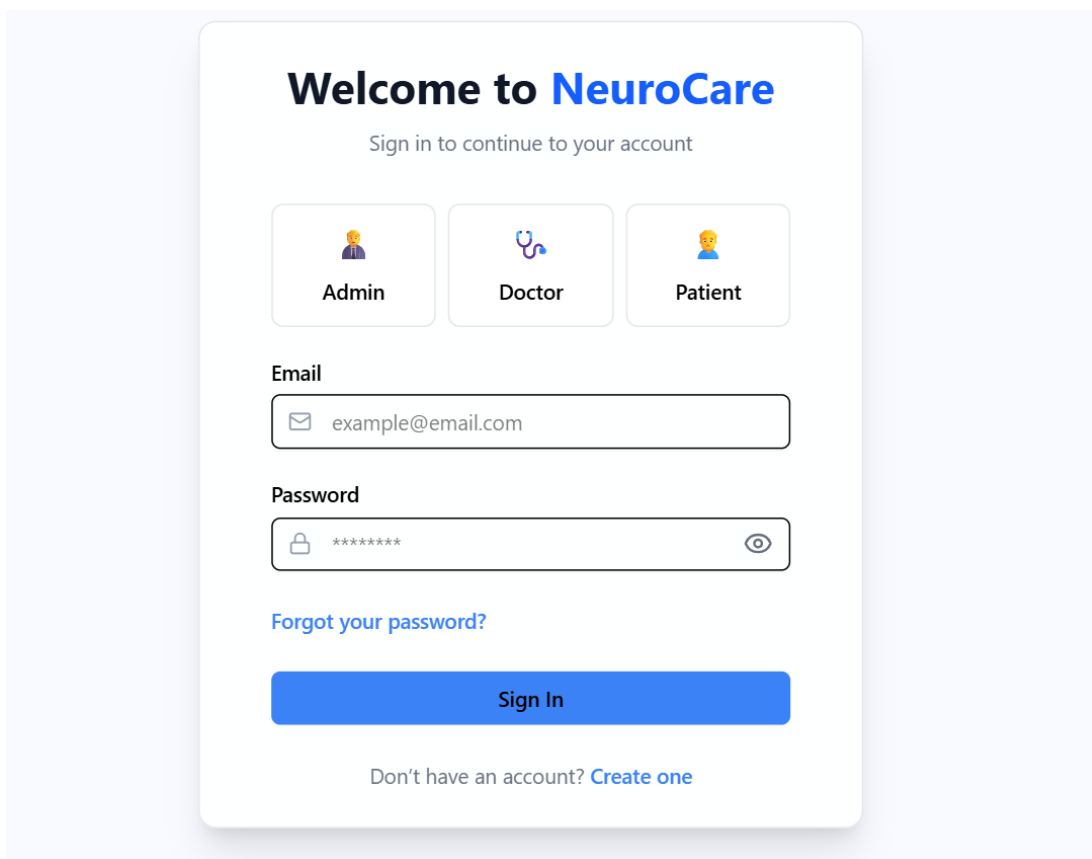
Profile Picture
Click or drag an image here
(Max size: 5MB)

I agree to the [Terms of Service](#) and [Privacy Policy](#)

Sign Up

Figure 5.3.2: Sign Up

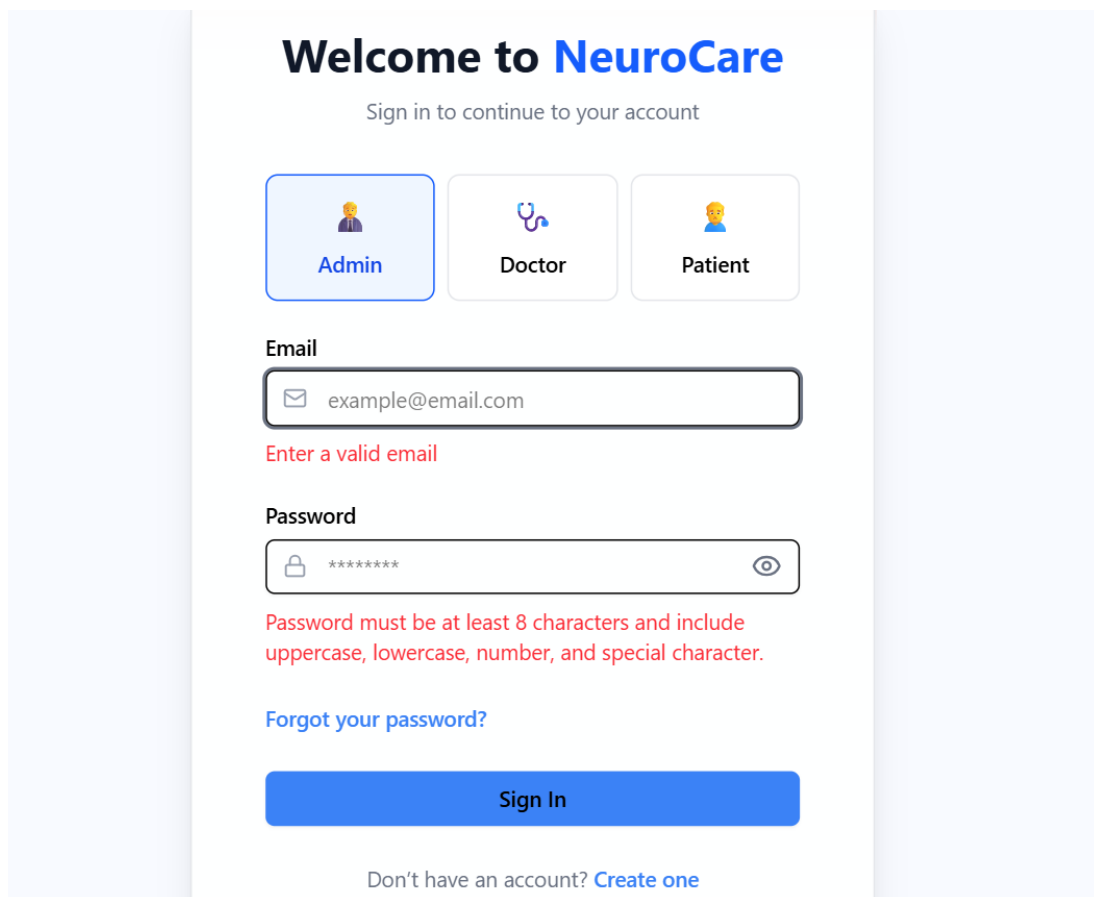
The Figure below represents the Sign In page of the user interface, this page can be accessed by any user i.e. Doctor, Patient or Admin.



The image shows a sign-in page for 'NeuroCare'. At the top, it says 'Welcome to NeuroCare' in a large blue font, followed by 'Sign in to continue to your account' in a smaller grey font. Below this are three buttons for user roles: 'Admin' (with a person icon), 'Doctor' (with a stethoscope icon), and 'Patient' (with a person icon). Underneath are two input fields: 'Email' with the placeholder 'example@email.com' and 'Password' with a masked password '*****' and a toggle eye icon. A blue link 'Forgot your password?' is positioned below the password field. A prominent blue 'Sign In' button is centered below the inputs. At the bottom, there is a link 'Don't have an account? Create one'.

Figure 5.3.3: Sign-In UI

The Figure below represents the Admin Sign In page of the user interface, this page can only be access by Admin. Admin can enter his credentials to login into the system.



The image shows a login interface for 'NeuroCare'. At the top, it says 'Welcome to NeuroCare' and 'Sign in to continue to your account'. There are three role selection buttons: 'Admin' (highlighted in blue), 'Doctor', and 'Patient'. Below these are input fields for 'Email' (containing 'example@email.com') and 'Password' (masked with asterisks). A red error message 'Enter a valid email' is shown below the email field. Another red error message 'Password must be at least 8 characters and include uppercase, lowercase, number, and special character.' is shown below the password field. A blue 'Sign In' button is at the bottom, and a link 'Don't have an account? Create one' is at the very bottom.

Welcome to NeuroCare

Sign in to continue to your account

[Admin](#) [Doctor](#) [Patient](#)

Email

Enter a valid email

Password

Password must be at least 8 characters and include uppercase, lowercase, number, and special character.

[Forgot your password?](#)

[Sign In](#)

Don't have an account? [Create one](#)

Figure 5.2: Admin Login UI

The Figure below represents the Admin Dashboard of the user interface, this page can only be access by Admin. Admin will enter his credentials to login into this Dashboard.

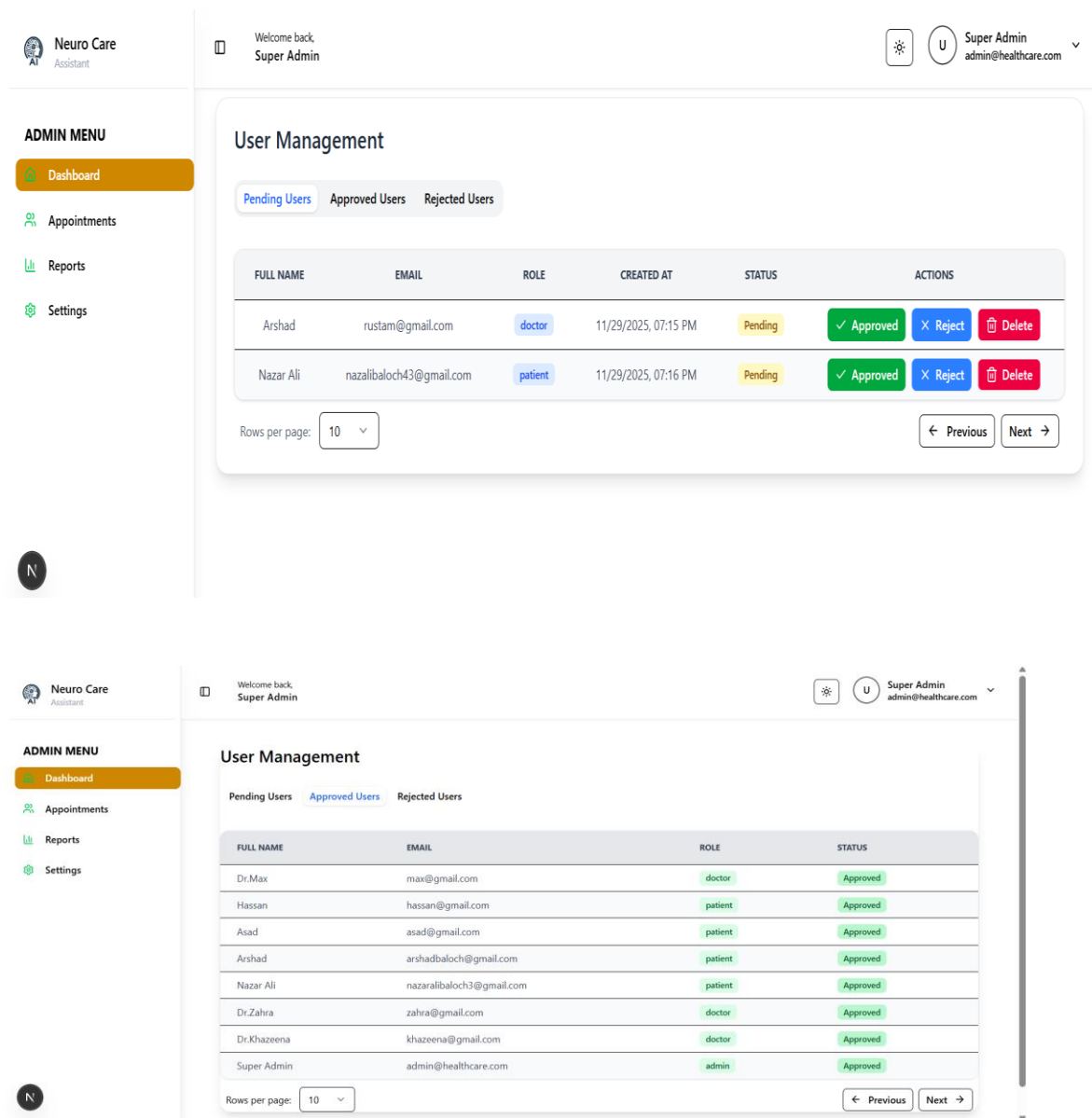


Figure 5.3.3: Admin Dashboard UI

Neuro Care Assistant

Welcome back, Dr.Khazeena

Dr.Khazeena khazeena@gmail.com

Doctor Dashboard

DOCTOR CODE	NAME	EMAIL	SPECIALIZATION	EXPERIENCE	ASSIGNED PATIENTS	ACTIONS
DR-002	Dr.Khazeena	khazeena@gmail.com	Neurologist	6 yrs	No Patients	

Rows per page: 10

[← Previous](#) [Next →](#)

Figure 5.3.6: Doctor Dashboard UI

Neuro Care Assistant

Welcome back, Dr.Khazeena

Dr.Khazeena khazeena@gmail.com

My Appointments

All Upcoming Past Cancelled

Asad
asad@gmail.com
0344232332

🕒 08:06 - 09:06 Pending [Approve](#) [Reschedule](#) [Cancel](#)

Nazar Ali
nazaraibaloch3@gmail.com
03301641135

🕒 03:08 - 10:13 Pending [Approve](#) [Reschedule](#) [Cancel](#)

Figure 5.3.7: Doctor Reschedule UI

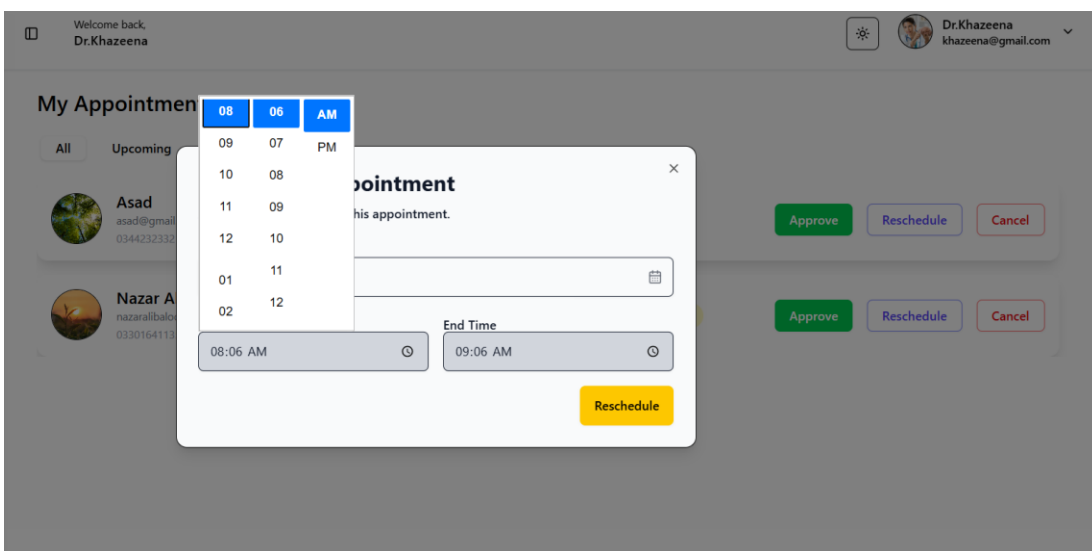
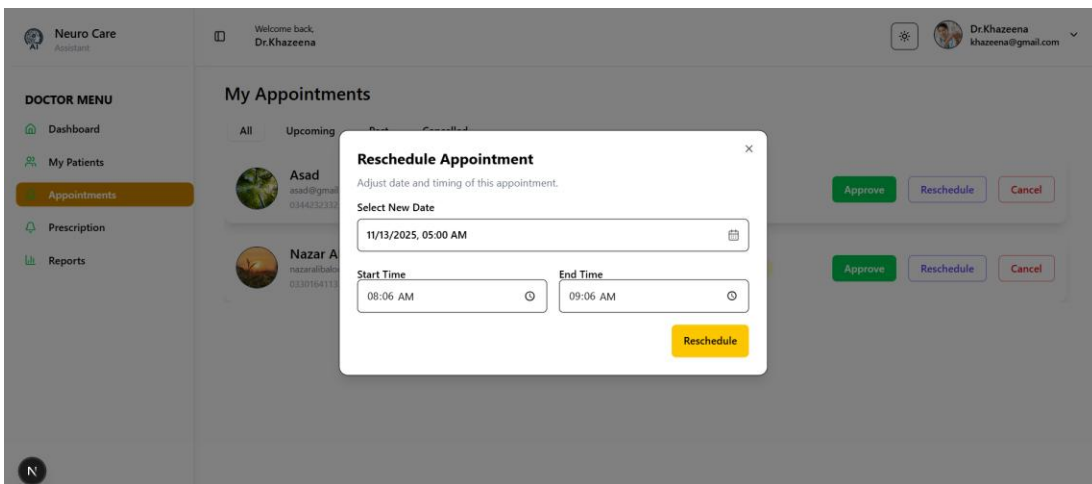
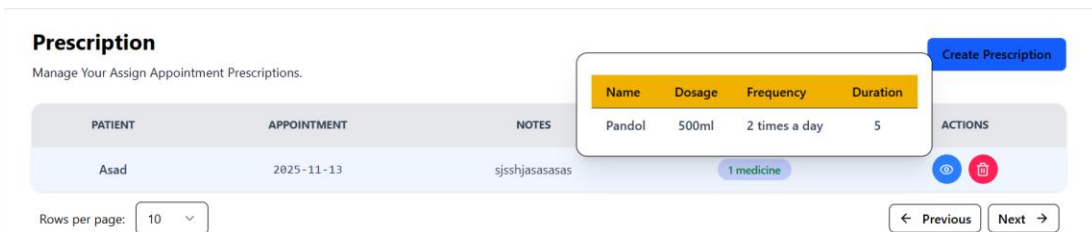


Figure 5.3.8: Doctor Reschedule UI



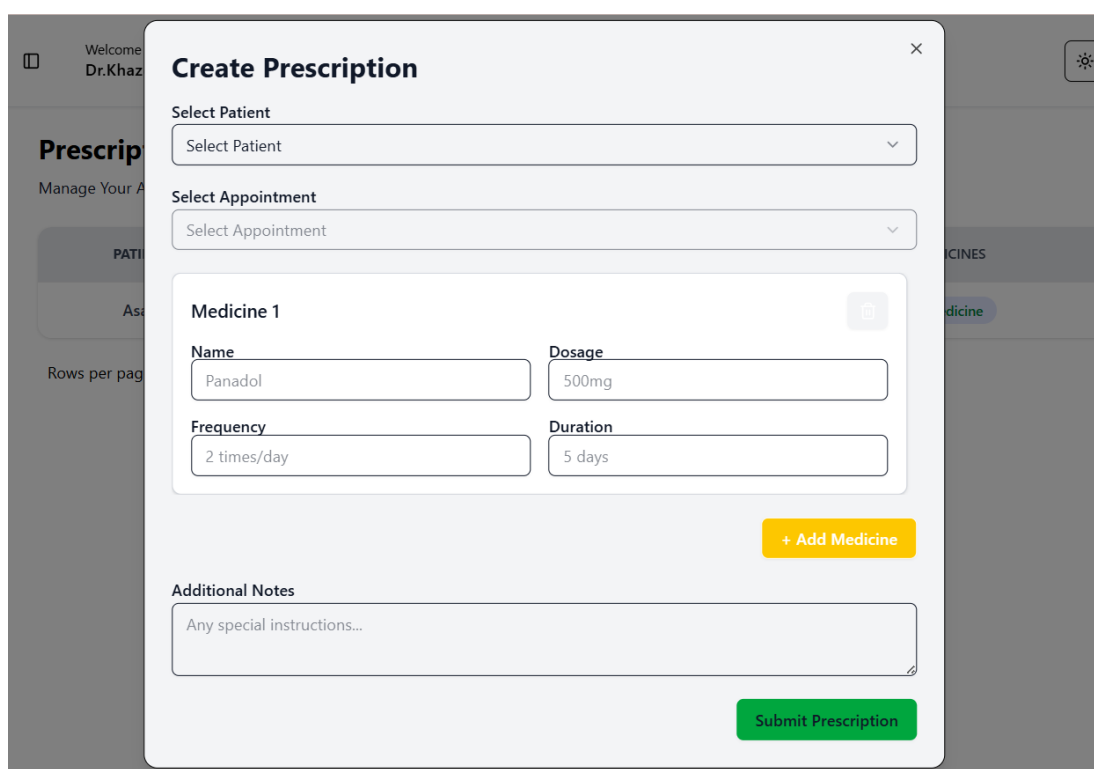
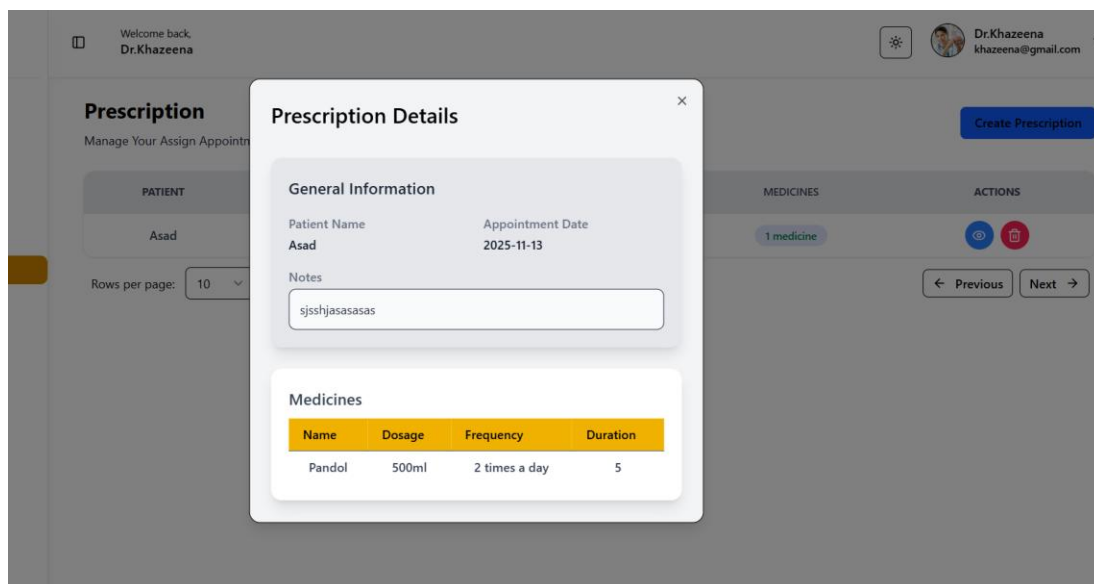


Figure 5.3.9: Create Prescription UI

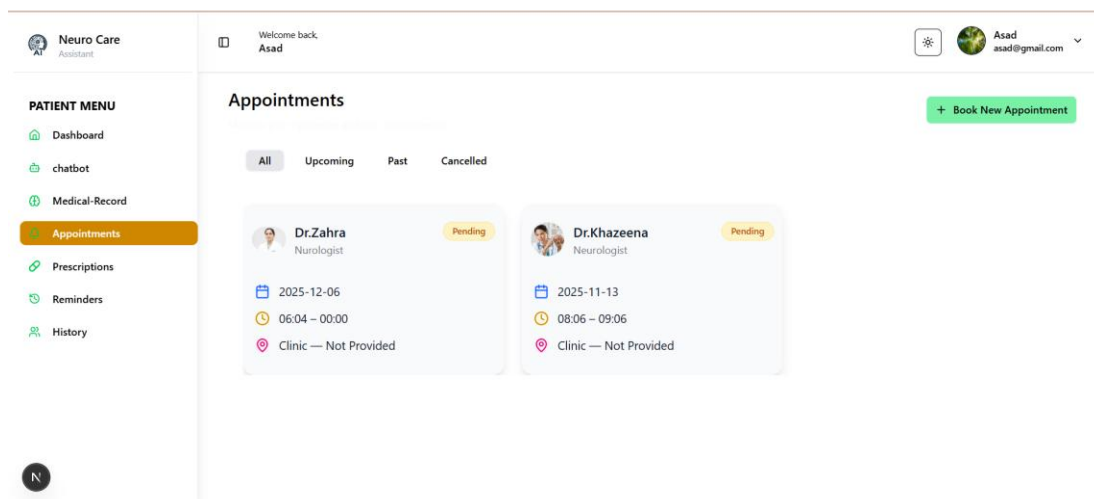


Figure 5.3.10: Patient Appointment Card Ui

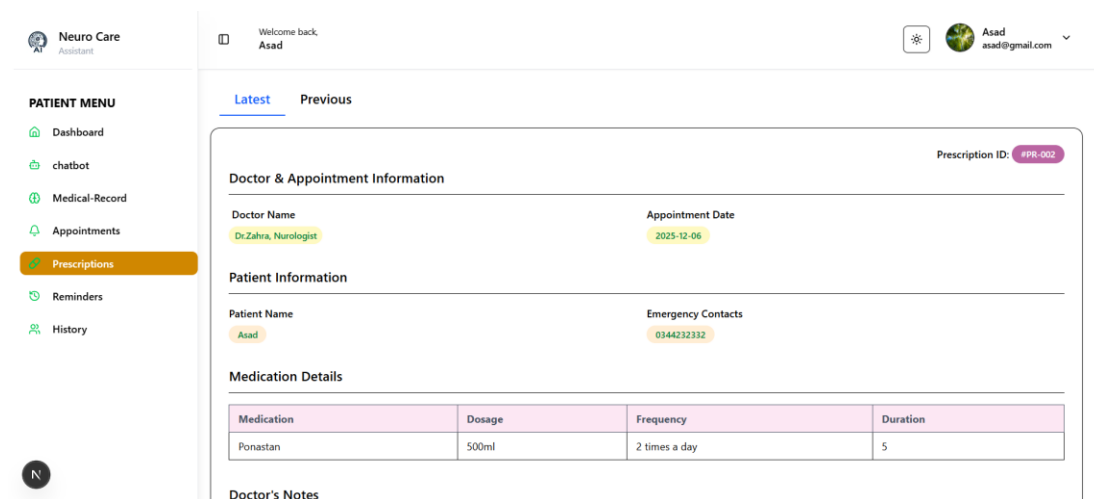


Figure 5.3.11: Patient Prescription Detail Ui

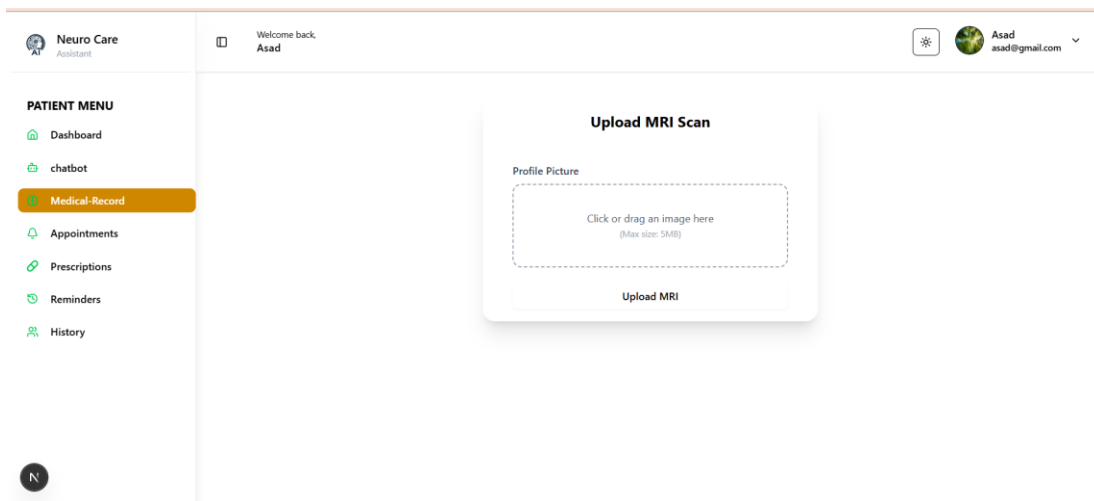


Figure 5.3.12: Patient Upload MRI UI

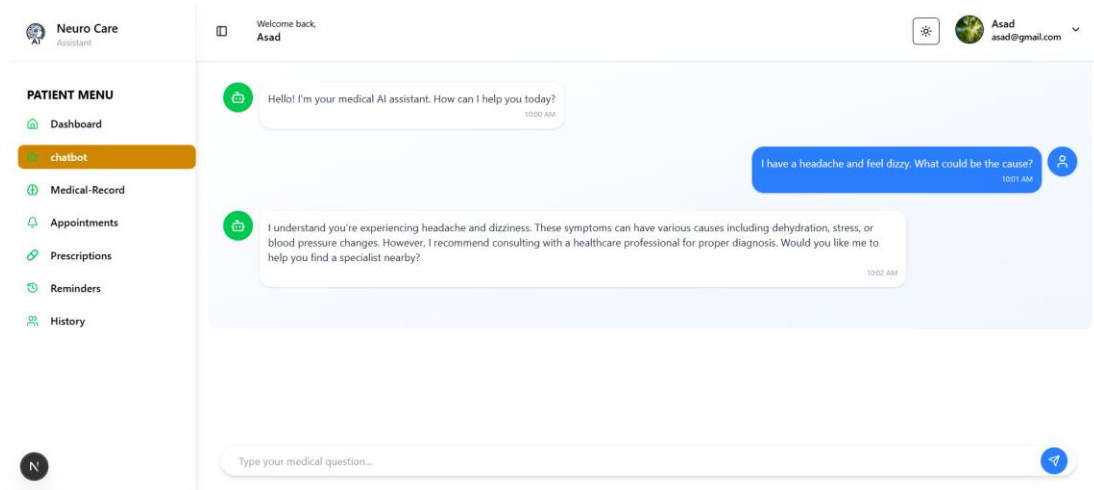


Figure 5.3.13: Patient Chatbot UI

CHAPTER 6

6.1 CONCLUSION AND RECOMMENDATIONS

This study has shown the efficiency of deep learning methods for neuroimaging analysis in the early diagnosis of Alzheimer's Disease. A proposed Convolutional Neural Network with a large MRI dataset of 33,706 samples distributed across four classes of dementia severity has successfully achieved an accuracy of 97%, confirming its feasibility as a diagnostic support tool. Advanced preprocessing with Pillow and hyperparameter optimization using Grid Search were performed to assure strength in model performance and generalizability.

Beyond classification, a comprehensive clinical decision support system using Retrieval-Augmented Generation, with HuggingFace Transformer embeddings, allows for intelligent, context-aware responses that will help healthcare practitioners as well as patients. These additional features—appointment scheduling, notification system, and management of digital prescriptions — point toward several important gaps in clinical workflow and can facilitate improved patient-doctor coordination and continuity of care. Together, these can contribute to a reduction in delays in diagnosis, improved outcomes of treatment, and a reduction in the social and economic consequences associated with NDD. Future work will involve integrating multi-modality neuroimaging data and advanced biomarkers to improve diagnostic accuracy, leveraging explainable AI techniques that allow for clinical interpretability of the system, and deployment of the system in real-world healthcare settings for clinical validation across diverse patient populations.

6.2 REFERENCES

- [1] F. A. L. W. E. C. G. M. S. M. F. Silvia Basaiaa, "Automated classification of Alzheimer's disease and mild cognitive," *Neuroimage: Clinical*, vol. 21, no. 2018, pp.1-8, 2019.
- [2] Michael W. Weiner, "The Alzheimer's Disease Neuroimaging Initiative (ADNI)," 2004-2016. [Online]. Available: <http://adni.loni.usc.edu/data-samples/access-data/http://adni.loni.usc.edu/data-samples/access-data/>. [Accessed 16th Aug 2025].
- [3] IEEE-2024-M.ABDUL ALIEM, Detecting Selected Neurodegenerative_ Diseases
- [4] Alzheimer's Association, *Alzheimer's Disease Facts and Figures*, 2025. [Online]. Available: <https://www.alz.org/alzheimers-dementia/facts-figures>. [Accessed: 12-Nov-2025].
- [5] A. M. Tăuțan *et al.*, "Artificial intelligence in neurodegenerative diseases: a review," 2021.
- [6] A. Ebrahimi *et al.*, "Convolutional neural networks for Alzheimer's disease: a review," *PeerJ/PMC*, 2021. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8083897/>. [Accessed: 28-Nov-2025].
- [7] N. F., "Retrieval-Augmented Generation (RAG) in Healthcare," *MDPI*, 2025.
- [8] S. D. Fenerty *et al.*, "The effect of reminder systems on patients' adherence: meta-analysis," 2012. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3287416/>. [Accessed: 28-Nov-2025].
- [9] "The impact of electronic health records on patient care and continuity," *Review Article*, 2024.
- [10] "Digital innovation in healthcare: effects on appointments and scheduling," *Journal/ResearchGate*, 2025.
- [11] H. S. F., "Explainable AI for Alzheimer Detection: A Review," *MDPI*, 2024

6.3 Appendix

Github Link : <https://github.com/nazar-ali>

<https://github.com/asad-10x>

