NeuroCare: AI-Driven Early Diagnosis and Prevention of Neurological Disorders

Abdul Moiz Imran
01-134221-003
Ayyan Sohail
01-134221-016

Supervisor: Dr. Farman Ali

Bachelor of Science in Computer Science

Department of Computer Science
Bahria University, Islamabad

October 2025

Bahria University
Islamabad Campus
(Department of Computer Science)

CERTIFICATE

We accept the work contained in the report titled "NeuroCare: AI Driven Early Diagnosis and Prevention of Neurological Disorders" as a confirmation to the required standard for the partial fulfillment the degree of BS(CS).

Supervisor
Name: Dr. Farman Ali
Date:______________

Internal Examiner
Name: Dr. Asad Ullah
Date: 17/12/25

External Examiner
Name: Dr. Imran Ali Q.
Date: 17/12/25

Project Coordinator
Name: Dr. Asim Ali
Date:________________

Head of Department
Name: Dr. Khurram Ehsan
Date:________________

Abstract

The proposed project, NeuroCare, is designed in such a way that it will utilize the concept of Machine Learning to identify Alzheimer's Disease and predict Brain Stroke and at the same time offers a single platform where a person can seek medical assistance virtually, create lifestyle plans, and exercise plans to prevent, diagnose and manage the advancement of the brain disease and also reduce the medical expenses by a significant margin. The existing diagnostic and monitoring techniques are extremely physical and invasive meaning they are inconvenient and not readily available. The absence of cost-effective and easy to use diagnostic aids impedes prompt and correct assessment. NeuroCare is designed to address the issue with the expensive and unreachable solutions that do not allow for identifying Alzheimer's Disease (AD) and Brain Stroke Prediction in time. The project also aims at dealing with the issues that are associated with the unavailability of clinics providing neurologists within certain regions. It will act as a point of cheap identification of Alzheimer's disease and Brain Stroke forecasting, customized lifestyle programs, virtual doctor-patient conferences, preventive strategies, and simplified administrative frameworks. The major objectives of the suggested NeuroCare project include revamping healthcare through offering low-priced detection systems of Alzheimer's disease, in addition to precise detection of Brain Stroke. Also, the project will educate people to have individualized lifestyle programs and online doctor visits, which will enable them to take preemptive and preventive healthcare. By doing so, NeuroCare will be able to make mental healthcare more accessible, efficient, and patient-centered. The main advantages of the proposed NeuroCare project are ease of conducting virtual consultations, geographical barriers, waiting time, and costs, and timely access to neurologists by people with neurological disorders.

Acknowledgments

We wish to thank Allah Almighty, who has given us a chance to venture into this project. We are sure that his assistance and advice will enter the path of successful realization of this venture.

We are also tremendously grateful to our highly respected supervisor, Dr. Farman Ali, who had never once let us down, offered a brilliant vision and continued to encourage us even in the process of this project. His inspiration and faith in us has helped us to push and perform our to best. It is an honor to work under his guidance and we are appreciative of the valuable mentorship that he gives us.

Finally, but most importantly, we would like to thank our parents with all our heart and their prayers and unconditioned support. Their support and emphases have been a boost and we are moving on with this project, and even farther.

ABDUL MOIZ IMRAN AND AYYAN SOHAIL
Bahria University
E-8, Islamabad, Pakistan

October 2025

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

CNN	Convolutional Neural Network
OOP	Object Oriented Programming
AI	Artificial Intelligence
API	Application Programming Interface
GUI	Graphical User Interface
HTTPS	Hypertext Transfer Protocol Secure

Chapter 1

Introduction

Among the common neurological disorders affecting the health of patients are Alzheimer Disease (AD) and Brain Stroke, both of which are major health problems in healthcare systems across the globe.

Alzheimer Disease (AD) is a progressive brain disease that kills neuron cells, and brain cells and impairs the brain, causing a cognitive decline and has alarming statistics of increasing the number of affected persons and is expected to affect a significant section of the population by 2050. [1]. Early diagnosis of AD is still a significant issue since its manifestations tend to appear once extensive brain damage has taken place and hence the importance of proper and prompt diagnostic tools. [2].

Moreover, the stroke, which is one of the main causes of mortality and disability in the world, emphasizes the need to predict it quickly and act [3].]. Stroke symptoms are sudden and diverse, so it is necessary to predict them as early as possible and provide medical care to avoid serious consequences.

With the mind-blowing figures and the variety of these neurological conditions, there is a demand for a software system that is powered with advanced technology, artificial intelligence, and telemedicine, which can transform the diagnosis and monitoring of various neurological conditions. The system is set to overcome the complexities of dealing with neurological health problems and addressing the growing rates of problems like Alzheimer and brain stroke. The suggested system does not only provide diagnosis and monitoring systems, but also prevention measures, exercises and customized diets to reduce the further development of these diseases. Moreover, it supports the consultations of the neurons in numerous situations, which guarantees an all-encompassing support of patients with neural disorders.

1.1 Problem Description

Diagnosis and follow-up of Alzheimer Disease (AD) is a very difficult task owing to a number of reasons. To begin with, the frequent visits to the clinic are required as reliance on physical examination and observations carried out by trained medical professionals. This is a burden to the patients and their family members who may find it challenging to travel to healthcare institutions. Moreover, the best clinical invasive procedures that are effective in diagnosing at an early stage have inherent risks and are not accessible in most cases since many countries, especially the underdeveloped ones, have limited resources.

The diagnostic tests including PET (Positron Emission Tomography) imaging and cerebrospinal fluid (CSF) analysis, are costly [4]. These diagnostic procedures are very expensive, a factor that presents a significant economic barrier that prevents many individuals with the disease to get timely and accurate diagnostic procedures.

The severity of Brain Stroke usually dictates the range of challenges during the recovery process exhibited by the patients. Such lasting issues that are common in poststroke recovery include memory lapses, problems with concentration, speech disorders, emotional instabilities such as depression, motor challenges, loss of sensation and difficulties with the swallowing. Moreover, strokes may cause death or disability, which is why preventive steps are extremely important. Moreover, shortage of neurologists in the underdeveloped regions is also another barrier that can hinder the delivery of punctual and specialized care to patients. Furthermore, the absence of appropriate instructions on the exercise programs.

1.2 Objectives

This system will enable the MRI image analysis to detect and identify Alzheimer Disease in the most easy way possible. It also tries to implement preventive healthcare practices, forecast possible brain strokes using the Random Forest Classifier, and give individualized recommendations and therefore, enable the patient to take care of their health following their own choices.

Additionally, the platform aims at providing prevention and dietary suggestions, personal training, and assistance with neurologists. It is not only patient care that it focuses on, but also streamlines administrative issues and secure payment procedures. The system will have the purpose of improving the overall efficiency of healthcare by streamlining these aspects, which will save time and resources of patients and neurologists. Patient empowerment is also kept in focus, by enabling profile management, such that user needs can be changed according to their changing needs. Also, the continuous improvement of the system is supported by the elements of the user feedback and the system analytics. This makes the system very dynamic and responsive to meet the demands of the ever changing needs of patients and neurologists. The targeted goals are the following:

- To introduce a user-friendly interface to ensure ease of use for patients and neurologists, to create more user satisfaction.
- To enable easy diagnosis of Alzheimer's disease.
- To reduce the risk of incidents of brain stroke.

1.3 Project Scope

The NeuroCare project scope of work covers an all-encompassing and user friendly platform to manage the neurological health. Beginning with user registration and profile management, the system will enable a smooth process on the part of the patient, neurologist, and administrator. Patients are able to add and update their medical information, and this encourages patient engagement in their health. Instead, neurologists will be able to create, log in and maintain their profiles, which will guarantee smooth onboarding process. The disease detection modules of the system are based on machine learning algorithms, which analyse drawings and MRI pictures to identify the specific conditions with the necessary level of accuracy, such as the Alzheimer disease, and deliver diagnostics and prescriptions in time. Kaggle dataset will be utilized in training machine learning models. Moreover, the NeuroCare System will also be able to make predictions related to the possible Brain Stroke. By this proactive method, people will get a chance to work against the possible risks and practice preventive measures which will eventually help in the minimisation of the likelihood of Brain Stroke.

NeuroCare also expands its capabilities to virtual medical appointments where patients can schedule appointments through the availability of neurologists and receive e-prescriptions. The system has a payment module; it involves the use of a secure system to conduct financial transactions associated with the fees of making an appointment. The project goes further to include individualized diets and exercise regimes and patients enter preferences and get customized suggestions.

A Virtual Doctor module will allow patients to make appointments and chat with neurologists. Lastly, the Statistical Analytics module includes the creation of administrator reports of the system efficacy, the ranking of the neurologist, depending on the ratings introduced by patients, the graphical system rating, monthly reports on the virtual consultation and disease detection outcomes to preserve the constant system enhancement and testing. It will be developed based on React Native and React JS frontend, Node.js and Express.js back-end with built-in ML models via FastApi, and MongoDB to store the data.

1.4 Solution Application Areas

The NeuroCare system has a big benefit, as AI-driven potential of detecting and predicting neurological diseases early and enhancing the access to healthcare services. Through incorporation of virtual consultations and lifestyle plans, the project will provide the timely intervention, which will translate to improved patient outcomes and lower burden on healthcare systems. The target of the project is the healthcare sector, and in particular, it is a neurology and telemedicine project. It serves patients, neurologists, and healthcare providers that would want efficient, remote, and AI-assisted diagnostic and predictive solutions.

Chapter 2

Literature Review

This section offers a demonstration of the market survey which was conducted to create the NeuroCare App by reviewing various available systems, their weaknesses, and their shortcomings.

In a bid to meet the demands of our stakeholders, there is a need to possess a bit of background information regarding the nature of similar systems which are presently available in the market. This survey assisted us to categorize our project into the category of services delivery, to set our ideas in a way we plan to reach our aims and to view the great picture of this sphere.

2.1 Introduction

Neurological issues, including the Alzheimer Disease (AD) and stroke of the brain are an increasing health problem that has a considerable effect on people and the health care systems worldwide. The growing rates of these diseases prove the urgent necessity of the effective tools of the early diagnosis and the timely prediction and the constant monitoring. In the present diagnosis of AD, there is a frequent problem of late onset of the disease symptoms, which requires effective and prompt tools. The same case applies to strokes, which are acute and heterogeneous and thus necessitate prompt prediction and treatment to reduce the devastating effects.

Such difficulties encountered in the diagnosis, prediction, accessibility, and follow-up treatment indicate a big gap in the current healthcare situation of neurological diseases. The innovative solutions that can offer the correct, accessible, and integrated support to the patients and healthcare providers are needed. The chapter analyzes the existing methods of detection and treatment of neurological disorders, assesses the benefits and drawbacks of the existing digital health solutions, and points out the gaps that the NeuroCare system is expected to fill. This review provides the grounds upon which the proposed innovations of NeuroCare can be implemented by synthesizing the recent developments in the domain of AI, telemedicine, and personalized healthcare.

2.2 Comparative Analysis

There are a number of applications and systems available in the healthcare field, some of which border on areas of neurological health and health monitoring in general. But such systems nearly always possess drawbacks in their diagnostic capability, service combination, or certain functionalities in their applications in neurological disorders such as Alzheimer Disease and Brain Stroke. These are the systems and their vulnerabilities identified as follows:

1. Ada Health
2. MyChart
3. Health Tracker

2.2.1 Ada Health

Ada Health is an existing application that offers health assessment capabilities.

- **Drawbacks:** The low accuracy of Ada Health in its diagnosis is a major weakness. In complicated neurological diseases, accuracy in diagnosis has been of utmost importance in delivering prompt and efficient intervention. [5].
- **Comparison with NeuroCare:** NeuroCare will remove this weakness by using the best machine learning algorithms that are specifically tailored to enhance diagnostic accuracy on neurological disorders like Alzheimer Disease and increase the accuracy of predicting brain strokes.

2.2.2 MyChart

MyChart is a system that provides healthcare-related services.

- **Drawbacks:** Major flaws of MyChart are divided services and unpersonalized services. This will result in an in-touch user experience and might not be sufficient to address the needs of the individual and changing needs of the patients with neurological conditions.
- **Comparison with NeuroCare:** NeuroCare suggests an all-in-one and unifying system incorporating different modules of healthcare. It specializes in offering customized diets, exercise, and preventive health plans with the goal of offering a non-disjointed and personalized user experience that is missing in other systems such as MyChart.

2.2.3 Health Tracker

Health Tracker refers to general applications used for monitoring health aspects.

- **Drawbacks:** A disadvantage of Health Tracker application is that it does not have certain special functions, such as brain stroke prediction. This reduces their usefulness to persons who might be susceptible to or have been dealing with stroke.
- **Comparison with NeuroCare:** NeuroCare uses machine learning as a specific part when predicting brain strokes and offers users with the chance to get a timely risk assessment [6]. This enhanced feature is what differentiates it among the more general health tracking devices that lack such predictive features of neurological events.

2.3 Summary

This chapter has performed a literature review that has put the NeuroCare project into perspective in the context of neurological healthcare. It pointed out the increasing difficulty of such conditions as Alzheimer's and Brain Stroke and thoroughly evaluated the current diagnostic procedures and systems. The review shows that there existed big gaps, such as diagnostic accuracy and accessibility, absence of integrated and personalized care, and absence of special predictive capabilities in the existing approaches and applications. The presented system suggested as NeuroCare serves as a solution to all these shortcomings, providing a unique AI-based early diagnosis and predictive system, combined with personalized preventive treatment and lifestyle planning, as well as telemedicine services.

Chapter 3

Requirement Specifications

Requirements analysis in both systems engineering and software engineering includes the work that is involved in identifying the requirements or state of a new or modified product or project, considering the potentially conflicting requirements of the different stakeholders, analyzing, documenting, validating, and maintaining software or system requirements [7].

3.1 Existing System

The existing system for the treatment of neurological conditions, including Alzheimer's Disease and Brain Stroke, depends heavily on the classical clinical practice, including the use of physical examination and invasive diagnostic techniques, such as PET scan and CSF analyses. These techniques have significant limitations, such as inconvenience and logistical challenges of frequent clinic visits by patients and caregivers, cost, inherent risks of invasive methods, and inaccessibility, especially in areas with a shortage of neurologists and resources. Moreover, the current healthcare processes may be disjointed, and the current tools do not necessarily have expert features related to the time-prompted forecast of such events as brain strokes or offering a comprehensive, customized preventive treatment and lifestyle advice, particularly to the specific optimization of neurological health. These constraints underscore the importance of the need to have a more convenient, affordable, integrated, and technologically improved system in order to overcome the issues of diagnosing, predicting, and managing such conditions.

3.2 Proposed System

The suggested system is called NeuroCare, and it is an AI-based healthcare solution that will help overcome the deficits of the current models in the diagnosis, prediction, and treatment of neurological conditions, in this case, Alzheimer's Disease and Brain Stroke. It is projected as an end-to-end software platform, which utilizes artificial intelligence and telemedicine functions to transform neurological care. The overall objective of NeuroCare is to provide easy detection and identification of the Alzheimer Disease based on MRI images analysis and predicting possible brain strokes with the help of machine learning models. In addition to diagnosis and forecast, the system will be developed to give customized recommendations, such as diet plans, physical workouts, as well as provide virtual access to neurologists. The combination of these major features will allow NeuroCare to enhance the precision and timeliness of the diagnosis, increase accessibility to special care, empower patients with individual preventative interventions, and eventually prevent the progression and effects of neurological problems, which will fill the acute gaps revealed in the current medical environment.

3.3 Functional Requirements

The functional requirements specify the capabilities that the NeuroCare system must possess to enable users to achieve their goals. The following are the functional requirements for the system.

Module 1: Profile Management

FR1: The system shall provide an option to create a new account using email for Patients and Neurologists.

FR2: The system shall allow users (Patient, Neurologist, Admin) to login to the system with their credentials.

FR3: The system shall allow users (Patient, Neurologist) to edit their profile information such as contact info, age, etc.

FR4: The system shall allow users (Patient, Neurologist) to reset forgotten passwords.

FR5: The system shall provide neurologist option to enter their PMC registration number during account creation.

FR6: The system shall verify if the entered email address is unique during account creation.

FR7: The system shall allow neurologists to add their banking details to receive appointment fees.

Module 2: Disease Detection (Alzheimer's)

FR1: The patient will be in a position to provide brain MRI images for the detection of Alzheimer's Disease.

FR2: The software will process uploaded MRI images using CNNs to identify various stages of Alzheimer's disease.

FR3: The system shall generate a diagnostic report that shows the presence and progression of Alzheimer's disease.

FR4: The system will offer preventive measures or recommendations depending on the result of the detection of Alzheimer's.

Module 3: Brain Stroke Prediction

FR1: The system will enable patients to enter essential health information, such as age, blood pressure, and smoking status, to predict brain stroke.

FR2: The system will be able to authenticate medical data that is entered to provide accuracy and reliability in predicting a brain stroke.

FR3: The system will analyze the input medical data and forecast the chances of occurrence of brain stroke.

FR4: To the user, the system will show the prediction outcome of the brain stroke probability occurrence.

FR5: In case the patient will provide the medical information in a wrong format (alphabetic characters when the user is asked to provide numerical values), then the system should show an error message and request the user to provide the appropriate format.

FR6: The system will offer a "View Recommendations" button to patients who were already predicted to have a stroke to receive personalized prevention recommendations delivered by the system.

FR7: Upon clicking the "View Recommendations" button, the system shall present personalized preventive measures for brain stroke to the patient, including lifestyle modifications, risk factor management, and emergency response plans.

Module 4: Lifestyle Plans

FR1: The system will enable the patients to enter their dietary preferences, allergies, and restrictions.

FR2: The system will create custom meal plans that include recipes depending on the preferences and disease of the patient.

FR3: The system will provide personalized exercise plans for patients depending on their disease.

FR4: The system will have step-by-step instructions and video tutorials links to exercises.

Module 5: Virtual Doctor

FR1: The patient shall be able to book appointments with neurologists for virtual consultation.

FR2: The neurologist shall be able to log their availability timings.

FR3: The system shall schedule appointments according to neurologist availability.

FR4: The patient shall be able to share system-generated and other medical reports with neurologists.

FR5: The neurologist shall be able to conduct virtual consultations.

FR6: The neurologist shall be able to generate e-prescriptions for patients.

FR7: The patient shall be able to chat with neurologists about queries.

FR8: The user (Patient or Neurologist) shall be able to view their booked appointments.

Module 6: Payment Gateway

FR1: The patient shall be able to pay appointment fees securely.

FR2: The neurologist shall be able to receive appointment fees.

Module 7: Statistical Analytics

FR1: The system shall generate analytics reports for the admin indicating the effectiveness of system-generated solutions based on neurologist remarks.

FR2: The patient shall be able to rate doctors and provide reviews.

FR3: The system shall rank doctors based on user ratings for the admin.

FR4: The user (Patient and Neurologist) shall be able to rate the system and provide reviews.

FR5: The system shall generate graphical reports on system ratings for the admin.

FR6: The system shall generate monthly reports about virtual consultations (number, duration, reason).

FR7: The system shall generate monthly reports about disease detections and results for the admin.

FR8: The system shall generate monthly reports about the number of patients and neurologists who joined.

FR9: The admin shall be able to view and manage (approve, delete) neurologist accounts.

FR10: The admin shall be able to view patient accounts.

3.4 Non-Functional Requirements

This section specifies non-functional requirements of NeuroCare system [8].

3.4.1 Reliability

REL-1: The system shall be very reliable such that users can access it at all times without a problem.

3.4.2 Usability

USE-1: The system shall provide user friendly GUI interface with minimal complications.

USE-2: The system shall provide error messages that are clear, concise, and actionable to assist users in error recovery.

3.4.3 Performance

PER-1: System functions, such as uploading medical reports and scheduling virtual consultations, shall respond correctly under normal operating conditions.

3.4.4 Security

SEC-1: The system uses industry-standard encryption protocols for sharing medical reports and payment processing.

SEC-2: The system shall enforce role-based access control (RBAC) to ensure that only authorized users can access specific functionalities based on their roles (patient, neurologist, admin).

SEC-3: Security is essential, and hence all user data shall be protected.

3.4.5 Data Consistency

DC-1: The database should be able to maintain data consistency and avoid creating duplicates or loss of data.

3.5 Use Case Diagram

Below are the actors for the NeuroCare system.

3.5.1 Patient

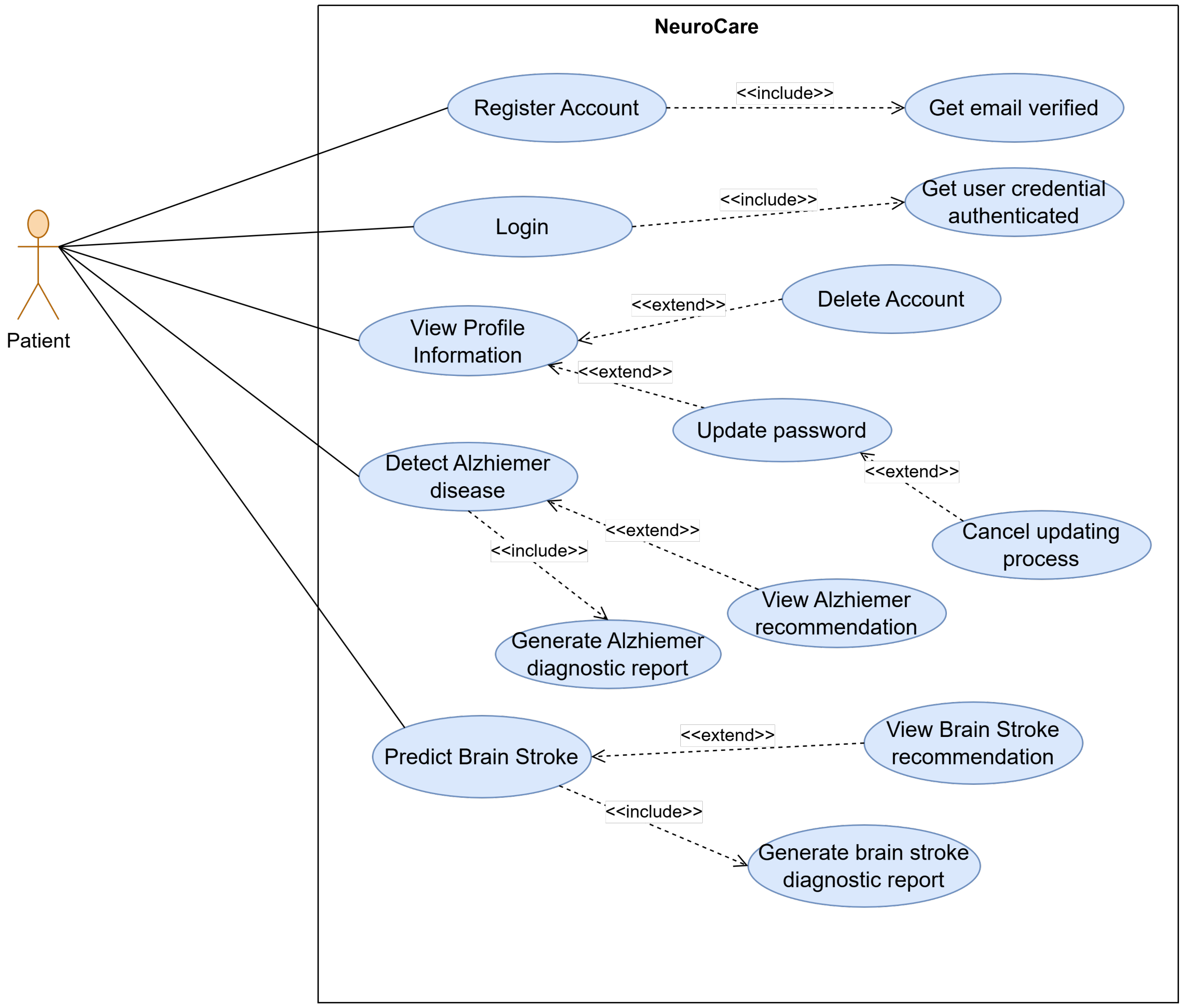


Figure 3.1: Use Case Diagram of Patient (Part 1)

Figure 3.2: Use Case Diagram of Patient (Part 2)

3.5.2 Neurologist

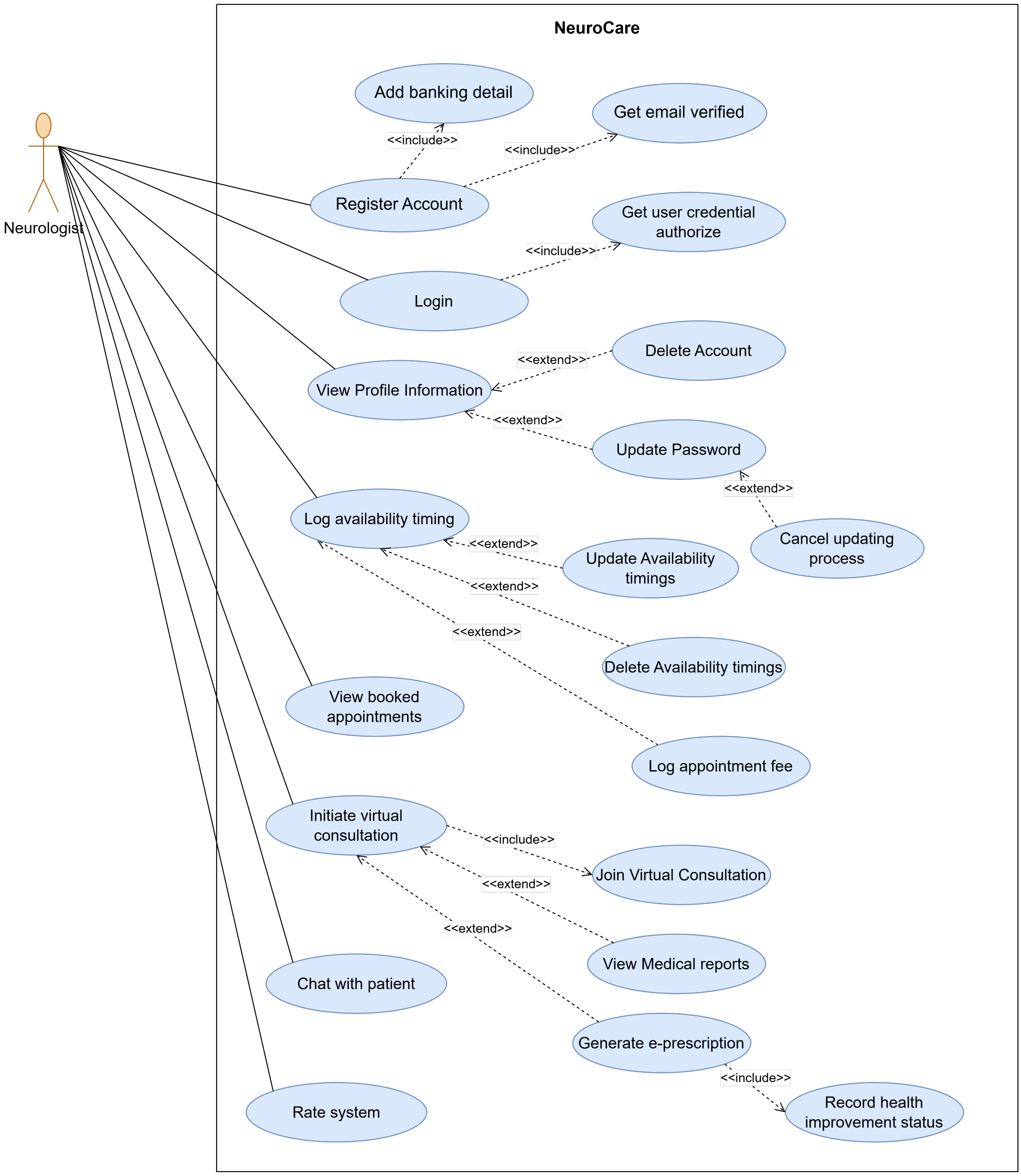


Figure 3.3: Use Case Diagram of Neurologist

3.5.3 Admin

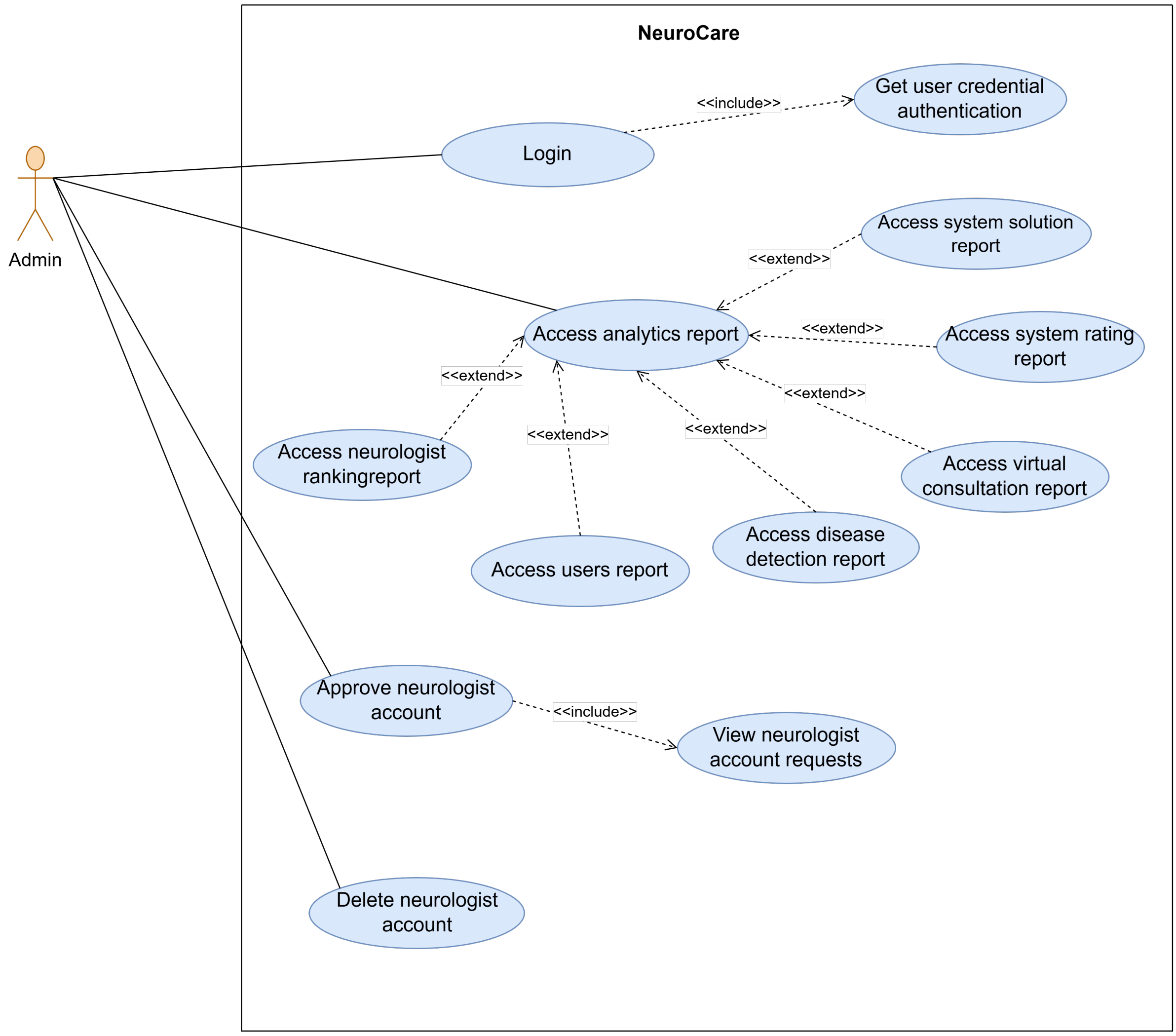


Figure 3.4: Use Case Diagram of Admin

3.6 Use Cases

This section details the functional requirements of the NeuroCare system through a set of use cases, describing the interactions between the system and its users (actors) to achieve specific goals [9]. Each use case is presented in a tabular format detailing its ID, name, actors, description, triggers, preconditions, postconditions, normal flow, alternative flows, exceptions, business rules, and assumptions.

3.6.1 Profile Management

Table 3.1: Register Account

Use Case ID	UC-1
Use Case Name	Register Account
Actors	Primary Actor: User (Patient, Neurologist)
Description	The User access NeuroCare through internet. User provide their email address and set up a password, along with additional personal details. Upon successful registration, users gain access to the system's features.
Trigger	The user clicks on the 'Register' button.
Preconditions	None
Postconditions	1. User account is successfully created in the system.
Normal Flow	1.0 Register account by patient. 1. User accesses the signup page of the system. 2. System presents options to create a new account using email address 3. User enters their email address. 4. User creates a password for their account. 5. User also enter their name, phone number, age and gender. 6. User click on submit button.
Alternative Flows	1.1 Register account by Neurologist: 1. System provide option to enter their bank name, account number, account holder name. 2. System provides option to enter PMC registration number. 3. System shows policy which Neurologist must accept before signup. 4. Return to step 2 of normal flow.
Exceptions	E1. User enters password less than 8 characters: 1. The system displays message "password length is less than ". 2. The system allows user to enter password again.
Business Rules	BR-1: The system shall ensure that the password entered by the user has a minimum length of 8 characters.
Assumptions	Users have access to a stable internet connection for creating new accounts.

Table 3.2: Get Email Address Verified

Use Case ID	UC-2
Use Case Name	Get Email Address Verified
Actors	Primary Actor: User (Patient, Neurologist)
Description	The system ensures that the email address provided during the registration process is unique and not already associated with an existing account in the system. This validation guarantees that each user has a distinct identifier within the system.
Trigger	The user submits their email address during the account registration.
Preconditions	PRE-1. User must provide an email address in the email field during the signup process.
Postconditions	POST-1. User account successfully created. POST-2. User redirected to login page.
Normal Flow	1. The system checks the entered email address against existing accounts in the system. 2. If the email address is not associated with any existing account, the system creates user account. 3. Account creation message shown to the user. 4. The system displays a message "User account successfully created."
Alternative Flow	None
Exceptions	E1. Email address entered is already associated with an existing account. 1. The system displays a message "Email address already in use". 2. The system prompts the user to enter a different email address.
Business Rules	BR-1: The system shall validate the entered email address to ensure its uniqueness within the system.
Assumptions	Users have access to a stable internet connection for creating new accounts.

Table 3.3: Login

Use Case ID	UC-3
Use Case Name	Login
Actors	Primary Actor: User (Patient, Neurologist, Admin)
Description	A user accesses the login interface of the system to input their credentials (email and password) for authentication. Upon submission, the system authenticates the provided credentials against the stored user data. If the credentials are valid, the system grants access to the user's account and redirects them to the appropriate landing page.
Trigger	The user clicks on the 'Login' button.
Preconditions	PRE-1. User must have an account.
Postconditions	POST-1. User successfully logs in to the system and is redirected to the appropriate landing page.
Normal Flow	1. User accesses the login page of the system. 2. System presents a login interface where the user can input their email and password. 3. User enters their credentials. 4. User submits the login form.
Alternative Flows	None
Exceptions	E1 Enter wrong credentials 1. Error message display. 2. System provide option to enter credentials again.
Business Rules	None
Assumptions	Assume that Users enter correct credentials during the login process

Table 3.4: Get User Credentials Authenticated

Use Case ID	UC-4
Use Case Name	Get User Credentials Authenticated
Actors	Primary Actor: User (Patient, Neurologist, Admin)
Description	The system authenticates the provided user credentials against the stored user data.
Trigger	User submits the login form with their credentials.
Preconditions	PRE-1. User must have an account.
Postconditions	POST-1. User successfully logs in to the system and is redirected to the Home page.
Normal Flow	1. User enters their credentials. 2. User submits the login form. 3. System authenticates the provided credentials against the stored user data. 4. After successful authentication, the system redirects the user to the 'Home Page'.
Alternative Flows	None
Exceptions	E1. If the provided credentials fail to authenticate against the stored user data 1. System displays an error message indicating the failure to log in. 2. System provides an option to re-enter credentials.
Business Rules	None
Assumptions	Users enter correct credentials during the login process.

Table 3.5: View Profile Information

Use Case ID	UC-5
Use Case Name	View Profile Information
Actors	Primary Actor: User (Patient, Neurologist)
Description	The user accesses the profile information section of the system to view their current profile information.
Trigger	The user clicks on 'view profile' option on the home page.
Preconditions	User must be logged in to the system.
Postconditions	POST-1. Profile information page displayed.
Normal Flow	1. User selects the option to view profile information. 2. User accesses the profile information section of the system. 3. System displays the current profile information such as name, age, phone number.
Alternative Flows	None
Exceptions	None
Business Rules	None
Assumption	Assume that Users navigate to the profile information section with the intention to view all details.

Table 3.6: Update Profile Information

Use Case ID	UC-6
Use Case Name	Update Profile Information
Actors	Primary Actor: User (Patient, Neurologist)
Description	Through this use case, the users, are allowed to update their personal information such as name, age, among others and phone number in the system. This feature will guarantee that the users will be able to keep the correct and updated information about themselves.
Trigger	The user clicks on 'update profile' option in the profile information section.
Preconditions	PRE-1. User must be logged in to the system.
Postconditions	POST-1. User successfully edits their profile information. POST-2. New information store in the database.
Normal Flow	1. User gets into to the profile editing section. 2. System offers an option of adding new profile details, such as name, age and phone number. 3. User types in new name in the input field. 4. User chooses age through a dropdown menu. 5. User enters new phone number in the input field. 6. User submits the changes. 7. System updates records of the user with the new data provided.
Alternative Flows	None
Exceptions	None
Business Rules	BR-1. System allows the user to update its personal details anytime.
Assumptions	None

Table 3.7: Cancel Updating Process

Use Case ID	UC-7
Use Case Name	Cancel Updating Process
Actors	Primary Actor: User (Patient, Neurologist)
Description	This use case enables users, to halt the process of editing their profile information. The outcome ensures that any unsaved alterations are discarded, returning the user to the profile information section without applying the changes
Trigger	The user clicks on the "cancel" option while in the process of updating their profile information.
Preconditions	PRE-1. User must be logged in to the system. PRE-2. User trying to update profile information
Postconditions	POST-1. User redirected to the profile information section.
Normal Flow	1. User clicks on the cancel button. 2. System overwrites the changes that are made by the user. 3. In the system, a message is shown on the screen indicating that the update has been cancelled. 4. The system sends the user to the profile information section.
Alternative Flows	None
Exceptions	None
Business Rules	BR-1: The system should notify the user about the outcome of the process, which includes feedback to give clear feedback in case of updates, cancellations, or any other pertinent outcomes.
Assumptions	None

Table 3.8: Delete Account

Use Case ID	UC-8
Use Case Name	Delete Account
Actors	Primary Actor: User (Patient, Neurologist)
Description	The "Delete Account" use Case enables User to delete their accounts on the system. When the deletion request is confirmed, all the related user information is erased forever within the system, this way, the User information is no longer available.
Trigger	The user clicks on 'Delete Account' option in the profile info section.
Preconditions	PRE-1. The user must be logged into the system.
Postconditions	POST-1. The account and related information of the user are deleted successfully. POST-2. A message is displayed to the user that the account is deleted.
Normal Flow	1. User visits the profile information. 2. User clicks the Delete Account option. 3. System deletes the user data and blocks the account.
Alternative Flows	None
Exceptions	None
Business Rules	BR-1: All associated user data must be permanently erased from the system.
Assumptions	Assume that Users understand the consequences of account deletion, including the loss of all associated data

Table 3.9: Update Password

Use Case ID	UC-9
Use Case Name	Update Password
Actors	Primary Actor: User (Patient, Neurologist)
Description	System assists the users to modify their existing password due to security reasons. Users are able to begin the process of updating their passwords by clicking on the profile management menu. Once the password is entered, and a new one is established, the system will update the user credentials in the database which will secure a successful change in the password and account safety.
Trigger	The user clicks on 'Update Password' option in the profile info section.
Preconditions	PRE-1. User must be logged in to the system.
Postconditions	POST-1. The user's password is successfully updated.
Normal Flow	1. User accesses the profile management menu. 2. User selects the option to change the password. 3. System prompts the user to enter the current password and the new password. 4. User enters the current password and sets a new one. 5. System authenticate user credentials. 6. After successful authentication, system updates the user credentials in the database. 7. System show successful message.
Alternative Flows	None
Exceptions	E1 If the entered current password is incorrect: 1. The system displays an error message indicating the incorrect password. 2. System allow user to re-enter current password.
Business Rules	BR-1: The system must authenticate the user's current password before allowing them to set a new one to maintain security. BR-2: Password changes must be reflected immediately in the database to ensure data consistency. BR-3: System confirm new password format is acceptable.
Assumptions	Users have logged into their accounts and wish to change their passwords.

Table 3.10: Add Banking Details

Use Case ID	UC-10
Use Case Name	Add Banking Details
Actors	Primary Actor: Neurologist
Description	Neurologists use the system to add their banking details to receive appointment fees for their services.
Trigger	The neurologist indicates that he wants to add his banking details.
Preconditions	1. The neurologist has successfully signed up for an account in the system. 2. The neurologist is logged in into the system.
Postconditions	1. The neurologist's banking details are successfully added in the system.
Normal Flow	1. The neurologist logs into the system. 2. The neurologist goes to the profile settings. 3. The neurologist chooses the option to add banking details. 4. The system will ask the neurologist to enter his banking details, such as account number, bank name, and name of account holder. 5. The neurologist fills his banking information. 6. The system stores the banking information of the neurologist.
Alternative Flows	None
Exceptions	None
Business Rules	BR-1: Neurologists must provide accurate banking details to receive appointment fees.
Assumptions	None

3.6.2 Alzheimer's Disease Detection

Table 3.11: Detect Alzheimer's Disease

Use Case ID	UC-11
Use Case Name	Detect Alzheimer's Disease
Actors	Primary Actor: Patient
Description	The system provides an opportunity to use brain MRI images uploaded by the patient and identify the signs of Alzheimer's disease. The process facilitates early diagnosis, which helps the patient and the medical team to embark on proper treatment and care plans early enough.
Trigger	The patient clicks on 'Detect Alzheimer's Disease' option.
Preconditions	PRE-1. Patient must be logged into the system.
Postconditions	POST-1. Analysis for Alzheimer's disease detection is completed. POST-2. Result shown to the user.
Normal Flow	1. The system is accessed by a patient to identify Alzheimer's disease. 2. Brain MRI images are uploaded by patients. 3. System conducts analysis of Alzheimer disease. 4. System shows detection outcome to the patient.
Alternative Flows	None
Exceptions	None
Business Rules	BR-1: Patient must upload Brain MRI for Alzheimer's disease detection.
Assumptions	Patients upload images without intentional distortion.

Table 3.12: Generate Alzheimer's Diagnostic Report

Use Case ID	UC-12
Use Case Name	Generate Alzheimer's Diagnostic Report
Actors	Primary Actor: Patient
Description	This use case explains how a diagnostic report on Alzheimer's disease will be created once the patient has been diagnosed with Alzheimer's disease using the system.
Trigger	Patient clicks on the "Generate diagnostic report" button after detecting Alzheimer's disease.
Preconditions	1. The patient is logged into the system. 2. The patient has detected Alzheimer's disease using the system.
Postconditions	A diagnostic report for Alzheimer's disease is generated and made available for the patient.
Normal Flow	1. Through the system, the patient identifies Alzheimer disease. 2. Once the diagnosis is detected, the patient clicks on the "Generate diagnostic report" button. 3. The system recalls the outcome of the detection of Alzheimer's disease. 4. The system produces a diagnostic report for the patient, including patient name, age, disease type, detection result, date, and time of detection.
Alternative Flows	None
Exceptions	None
Business Rules	BR-1: The diagnostic report should be generated promptly and made available to the patient after detection of Alzheimer's disease.
Assumptions	1. The patient's information is correctly stored and accessible within the system.

Table 3.13: View Alzheimer's Disease Recommendations

Use Case ID	UC-13
Use Case Name	View Alzheimer's Disease Recommendations
Actors	Primary Actor: Patient
Description	Patients who have detected Alzheimer's disease can access personalized recommendations for managing the disease.
Trigger	Patient clicks on view recommendation button after receiving Alzheimer's disease detection result.
Preconditions	PRE-1. Patient must be logged in to the system. PRE-2. Patient has detected Alzheimer's disease using system.
Postconditions	POST-1. Patient accesses personalized recommendations for managing Alzheimer's disease.
Normal Flow	1. A patient is provided with an outcome of the detection of Alzheimer's disease. 2. Patient clicks on the "View Recommendations" button. 3. System presents a tailor-made recommendations page to do deal with Alzheimer's disease.
Alternative Flows	None
Exceptions	None
Business Rules	BR-1: Recommendations provided by the system must be specific to managing Alzheimer's disease.
Assumptions	The system generates recommendations based on current research on Alzheimer's disease management.

3.6.3 Brain Stroke Prediction

Table 3.14: Predict Brain Stroke

Use Case ID	UC-14
Use Case Name	Predict Brain Stroke
Actors	Primary Actor: Patient
Description	This use case allows patients to provide vital medical information like age, blood pressure as well as smoking status and thus the system can evaluate the patient and determine whether he is at risk of having a brain stroke. The result is a knowledgeable forecast of the possibility of stroke, which helps patients to take preventive measures.
Trigger	The patient clicks on 'Predict Brain Stroke' option.
Preconditions	PRE-1. Patient must be logged into the system.
Postconditions	POST-1. Analysis for Brain Stroke is completed. POST-2. Prediction result shown to the user.
Normal Flow	1. Patient accesses the system to predict brain stroke. 2. Patient inputs medical details including gender, age, hypertension, heart disease, married status, work type, residence type, average glucose level, BMI, smoking status. 3. System validates the entered medical details. 4. System performs analysis for prediction. 5. System print result.
Alternative Flows	None
Exceptions	E1 If the patient enters medical details in an incorrect format (input alphabetic characters in numerical fields). 1. The system should display an error message and prompt the user to input the correct format. 2. Continue with step 2 of the normal flow.
Business Rules	BR-1: Ensure all mandatory medical details are provided for accurate brain stroke prediction. BR-2: Age must be in positive numbers.
Assumptions	Patients provide accurate medical information without intentional misrepresentation.

Table 3.15: Generate Brain Stroke Predictive Report

Use Case ID	UC-15
Use Case Name	Generate Brain Stroke Predictive Report
Actors	Primary Actor: Patient
Description	This use case describes the process by which a predictive report for Brain Stroke is generated for the patient.
Trigger	Patient clicks on the "Generate predictive report" button after predicting brain stroke using system.
Preconditions	1. The patient is logged into the system. 2. The patient has predicted brain stroke using system.
Postconditions	A predictive report for Brain Stroke is generated and made available for the patient.
Normal Flow	1. The patient predicts brain stroke using system. 2. The patient clicks on the "Generate predictive report" button. 3. The system retrieves brain stroke prediction result. 4. The system generates a predictive report for the patient, including patient name, age, disease type, prediction result, date, and time of prediction.
Alternative Flows	None
Exceptions	None
Business Rules	BR-1: The predictive report should be generated promptly and accurately based on the patient's prediction result.
Assumptions	1. The patient's information is correctly stored and accessible within the system.

Table 3.16: View Brain Stroke Recommendations

Use Case ID	UC-16
Use Case Name	View Brain Stroke Recommendations
Actors	Primary Actor: Patient
Description	Patients who have predicted brain stroke can access personalized prevention recommendations provided by the system.
Trigger	Patient clicks on view recommendation button after prediction result of brain stoke.
Preconditions	PRE-1. Patient must be logged in to the system. PRE-2. Patient must have received a prediction result for brain stroke.
Postconditions	POST-1. Patient accesses personalized prevention recommendations for brain stroke.
Normal Flow	1. Patient receives a result for brain stroke prediction. 2. Patient accesses the "View Recommendations" button. 3. System displays a personalized recommendations page for managing brain stroke.
Alternative Flows	None
Exceptions	None
Business Rules	BR-1: Recommendations provided by the system must be specific to brain stroke prevention.
Assumptions	The system generates recommendations based on current research on brain stroke prevention.

3.6.4 Virtual Doctor

Table 3.17: Search Appointment

Use Case ID	UC-17
Use Case Name	Search Appointment
Actors	Primary Actor: Patient
Description	The patient searches for available appointment slots with a neurologist for a virtual consultation.
Trigger	The patient clicks on 'search' option in the Appointment section.
Preconditions	PRE-1. The patient is logged into the system.
Postconditions	POST-1. All the available appointment slots for the entered date display on screen.
Normal Flow	1. The patient accesses the book appointment interface for searching appointments. 2. The patient inputs the date for which they want to search for appointments. 3. The system displays available appointment slots for the entered date, including times, durations, fees, neurologist's name, and ratings.
Alternative Flows	None
Exceptions	E1: Entered date is before the current date. 1. The system asks the patient to enter the current or future date. 2. Continue with step 2 of the normal flow.
Business Rules	None
Assumptions	None

Table 3.18: Book Appointment

Use Case ID	UC-18
Use Case Name	Book Appointment
Actors	Primary Actor: Patient
Description	The patient schedules an appointment with a neurologist for a virtual consultation.
Trigger	The patient clicks on book appointment button in home page.
Preconditions	PRE-1. The patient is logged into the system.
Postconditions	POST-1. The patient is redirected to the payment page.
Normal Flow	1. The patient searches for available appointment slots as per the "Search Appointment" use case. 2. The patient selects a suitable appointment slot from the available options. 3. The system redirects the patient to the payment page.
Alternative Flows	None
Exceptions	E1: Selected appointment slot overlaps with an existing appointment for the patient. 1. The system asks the patient to choose another appointment slot. 2. If the patient cancels the appointment booking process, then the use case terminates. 3 Else if the patient requests another appointment slot, then continue with step 2 of the normal flow.
Business Rules	BR-1: Payment must be completed before the appointment is considered booked.
Assumptions	None

Table 3.19: Log Availability Timings

Use Case ID	UC-19
Use Case Name	Log Availability Timings
Actors	Primary Actor: Neurologist
Description	Neurologists use the system to log their availability timings, indicating the dates and time slots they are available for consultations.
Trigger	The Neurologist clicks on 'log Availability Timing' option.
Preconditions	1. The neurologist is logged into the system.
Postconditions	1. The neurologist's availability timings are successfully logged in the system. 2. Logged timings before current system time are deleted.
Normal Flow	1.0 Neurologist logs single availability timing 1. The neurologist accesses the system interface showing his availability timings. 2. The neurologist requests to add availability timing. 3. The neurologist inputs the date he is available 4. The neurologist inputs the time slots for entered date. 5. The system saves the entered availability timings. 6. The system retrieves current time and delete all previous logged timings.
Alternative Flows	1.1 Neurologist logs multiple availability timing 1. Neurologist requests to add another availability timing. 2. Return to step 2 of normal flow.
Exceptions	E1 Entered date is before current date 1. The system asks neurologist to enter current or future date. 2. Continue with step 2 of normal flow. E2 Entered time is before current time 1. The system asks neurologist to enter future time. 2. Continue with step 3 of normal flow.
Business Rules	BR-1. Appointments must be scheduled based on neurologist's availability.
Assumptions	1. Neurologists have consistent availability patterns for consultations.

Table 3.20: Update Availability Timings

Use Case ID	UC-20
Use Case Name	Update Availability Timings
Actors	Primary Actor: Neurologist
Description	Neurologists use the system to update their availability timings for scheduling appointments.
Trigger	The neurologist clicks on update button in log availability timing section.
Preconditions	PRE-1. The neurologist is logged into the system.
Postconditions	POST-1. The selected availability timings are updated in the system.
Normal Flow	1. The neurologist accesses the system interface showing his availability timings. 2. The system displays options to edit existing availability timings for which no appointment has been booked. 3. The neurologist selects an availability timing to update. 4. The neurologist modifies the selected availability timing, such as changing the start or end time. 5. The neurologist confirms the changes. 6. The system updates the availability timing with the new information.
Alternative Flows	None
Exceptions	None
Business Rules	BR-1. Availability timings can only be updated if no appointments are booked during the updated time slot.
Assumptions	None

Table 3.21: Delete Availability Timings

Use Case ID	UC-21
Use Case Name	Delete Availability Timings
Actors	Primary Actor: Neurologist
Description	Neurologists use the system to delete those logged availability timings for which no appointment has been booked.
Trigger	The Neurologist clicks on 'Delete timing' option in log availability section.
Preconditions	1. The neurologist is logged into the system.
Postconditions	1. The selected availability timings are deleted from the system.
Normal Flow	1. The neurologist accesses the system interface showing his availability timings. 2. The shows an active delete button with each availability timing for which no appointment has been booked. 3. The neurologist clicks on active delete button for an availability timing. 4. The system deletes selected availability timing.
Alternative Flows	None
Exceptions	None
Business Rules	BR-1. Availability timing for which appointment has been booked cannot be deleted.
Assumptions	None

Table 3.22: Log Appointment Fee

Use Case ID	UC-22
Use Case Name	Log Appointment Fee
Actors	Primary Actor: Neurologist
Description	Neurologists use the system to log the appointment fee for his services.
Trigger	The neurologist indicates that he wants to log the appointment fee.
Preconditions	1. The neurologist is logged into the system.
Postconditions	1. The appointment fee is successfully logged in the system.
Normal Flow	1.0 No appointment has been booked for selected availability timing 1. The neurologist accesses the system interface for logging the appointment fee. 2. The neurologist selects an availability timing slot. 3. The neurologist inputs the appointment fee for his services. 4. The system saves the entered appointment fee details.
Alternative Flows	1.0 Appointment has been booked for selected availability timing 1. System informs neurologist that appointment has been booked so new appointment fee cannot be added.
Exceptions	None
Business Rules	BR-1: The appointment fee must be accurately logged for each consultation. BR-2: Appointment fee can only be logged if no appointment has been booked for selected availability timing.
Assumptions	None

Table 3.23: View Booked Appointments

Use Case ID	UC-23
Use Case Name	View Booked Appointments
Actors	Primary Actor: User (Patient, Neurologist)
Description	Patient uses the system to view appointments that he has booked and neurologists use the system to view appointments for which he need to conduct virtual medical consultation.
Trigger	The User clicks on 'view appointment' option in book appointment section.
Preconditions	1. The user is logged into the system. 2. The user has associated appointments with his account.
Postconditions	1. Appointment associated with the user's account is displayed to him.
Normal Flow	1. The user accesses the system interface for viewing appointments. 2. The system retrieves and displays a list of appointments associated with the user's account. 3. For each appointment, relevant details such as date, time, patient name and neurologist name are displayed.
Alternative Flows	None
Exceptions	None
Business Rules	None
Assumptions	None

Table 3.24: Initiate Virtual Consultation

Use Case ID	UC-24
Use Case Name	Initiate Virtual Consultation
Actors	Primary Actor: Neurologist
Description	The neurologist initiates a virtual consultation session according to booked appointment time so that patient is able to join that session.
Trigger	Booked appointment time comes and neurologist clicks on start virtual consultation session button.
Preconditions	1. Neurologist is logged into the system. 2. Virtual consultation appointment is scheduled.
Postconditions	1. Virtual consultation session is initiated. 2. Patient is promoted on his interface to join virtual consultation session.
Normal Flow	1. Neurologist accesses the system interface for virtual consultation. 2. Neurologist requests to initiate virtual consultation session. 3. Virtual consultation session gets initiated.
Alternative Flows	None
Exceptions	None
Business Rules	BR-1: Neurologists are eligible to initiate virtual consultations.
Assumptions	1. Neurologist have device with webcam.

Table 3.25: Join Virtual Consultation

Use Case ID	UC-25
Use Case Name	Join Virtual Consultation
Actors	Primary Actor: Patient
Description	The patient joins a virtual consultation initiated by the neurologist.
Trigger	Patient clicks on join consultation button.
Preconditions	1. Patient is logged into the system. 2. Virtual consultation session has been initiated by the neurologist.
Postconditions	1. Patient has joined virtual consultation session. 2. Virtual consultation session has been ended.
Normal Flow	1. System prompts patient to join virtual consultation session initiated by neurologist. 2. Patient accesses the system interface for virtual consultation. 3. Patient joins virtual consultation session. 4. Neurologist conducts consultation with patient. 5. Either neurologist or patient ends the consultation.
Alternative Flows	None
Exceptions	None
Business Rules	None
Assumptions	1. Patient have device with webcam.

Table 3.26: Share Medical Reports

Use Case ID	UC-26
Use Case Name	Share Medical Reports
Actors	Primary Actor: Patient
Description	The patient is able to share medical reports with the neurologist after virtual consultation session if neurologist asks for it during consultation.
Trigger	The patient clicks on upload medical reports option.
Preconditions	PRE-1. Patient is logged into the system. PRE-2. A virtual consultation session has been ended.
Postconditions	POST-1. Medical reports are successfully shared with the neurologist.
Normal Flow	1. Patient accesses the system interface to share medical reports with neurologist. 2. Patient selects the option to upload medical reports. 3. Patient selects the medical reports from his device. 4. Patient confirms to send the reports. 5. System sends upload medical report to neurologist.
Alternative Flows	None
Exceptions	None
Business Rules	None
Assumptions	1. Patients share medical reports received from authentic sources.

Table 3.27: View Medical Reports

Use Case ID	UC-27
Use Case Name	View Medical Reports
Actors	Primary Actor: Neurologist
Description	The neurologist is able to view medical reports that different patients have shared with him.
Trigger	The neurologist clicks on patient reports option in virtual consultation section.
Preconditions	PRE-1. Neurologist is logged into the system. PRE-2. Patient has shared medical reports with neurologist.
Postconditions	POST-1. Medical reports are displayed to the neurologist.
Normal Flow	1.0 View Single Report 1. Neurologist accesses the system interface to view medical reports. 2. A list of shared medical reports with patient details are displayed. 3. Neurologist selects the medical reports from list. 4. Selected medical report is displayed to the neurologist.
Alternative Flows	1.1 View Multiple Reports 1. Neurologist wants to view another report. 2. Return to step 2 of normal flow.
Exceptions	None
Business Rules	None
Assumptions	1. Patients share medical reports received from authentic sources.

Table 3.28: Generate E-Prescription

Use Case ID	UC-28
Use Case Name	Generate E-Prescription
Actors	Primary Actor: Neurologist
Description	After a virtual consultation, the neurologist can generate an e-prescription for a patient using a predefined template having necessary details. The system validates the e-prescription to ensure all required fields are filled and stores the details. Once finalized, the system sends the e-prescription to the patient.
Trigger	Neurologist clicks on E-Prescription button in consultation section.
Preconditions	PRE-1. Neurologist is logged into the system. PRE-2. A virtual consultation session has been ended.
Postconditions	POST-1. E-prescription is successfully generated and stored by the system. POST-2. E-prescription is sent to the respective patient.
Normal Flow	1. Neurologist accesses the system interface to generate e-prescription. 2. Neurologist selects the appointment for which the prescription is to be generated. 3. Neurologist fills out the relevant fields in the e-prescription template, including patient information, prescribed medication, dosage, duration, any special instructions. 4. Neurologist finalizes the e-prescription. 5. System validates the e-prescription to ensure all required fields are filled. 6. System stores the e-prescription details. 7. System sends the e-prescription to the patient.
Alternative Flows	None
Exceptions	None
Business Rules	BR-1: The e-prescription template must include fields for patient information, prescribed medication, dosage, duration, any special instructions.
Assumptions	None

Table 3.29: Record Health improvement status

Use Case ID	UC-29
Use Case Name	Record Health improvement status
Actors	Primary Actor: Neurologist
Description	After a follow-up consultation, the neurologist can record the health status of the patient. This status includes details about the patient's health progress since the previous consultation.
Trigger	Neurologist select record health improvement status option in e-prescription.
Preconditions	PRE-1. Neurologist is logged into the system.
Postconditions	POST-1. Health improvement status is successfully recorded in the system.
Normal Flow	1. Neurologist accesses the e-prescription. 2. Neurologist selects health improvement status option. 3. System display 3 remarks (select: improving, not improving, getting worse). 4. Neurologist select option. 5. System stores the health improvement status details.
Alternative Flows	None
Exceptions	None
Business Rules	BR-1: The health improvement report must include fields for recording details about the patient's health progress since the previous consultation.
Assumptions	None

Table 3.30: View E-Prescription

Use Case ID	UC-30
Use Case Name	View E-Prescription
Actors	Primary Actor: Patient
Description	Patient can view E-Prescription which was prescribed for him by neurologist and get medication reminders if he wants.
Trigger	Patient clicks on view E-Prescription option in book appointment.
Preconditions	PRE-1. Patient is logged into the system. PRE-2. Neurologist has generated an E-Prescription for the patient.
Postconditions	POST-1. E-prescription is displayed to the patient.
Normal Flow	1. Patient accesses the system interface to view E-Prescription. 2. A list of E-Prescriptions with appointment ids are displayed to the patient. 3. Patient selects one the E-Prescription. 4. Selected E-Prescription details is displayed to the patient.
Alternative Flows	None
Exceptions	None
Business Rules	None
Assumptions	None

Table 3.31: Chat with Neurologist

Use Case ID	UC-31
Use Case Name	Chat with Neurologist
Actors	Primary Actors: Patient
Description	Patients can have a chat with a neurologist to discuss his queries. Patient types a message and it will be sent to the neurologist.
Trigger	Patient clicks on chat icon shown with neurologist detail.
Preconditions	PRE-1. Patient is logged into the system.
Postconditions	POST-1. Message is sent to the neurologist.
Normal Flow	1.0 Patient sends one message 1. Patient accesses the chat feature within the system. 2. System displays a list of neurologists. 3. Patient selects one of the neurologists. 4. Patient types a message. 5. Patient sends message to the neurologist.
Alternative Flows	1.1 Patient sends multiple messages 1. Return to step 4 of normal flow.
Exceptions	None
Business Rules	None
Assumptions	None

Table 3.32: Chat with Patient

Use Case ID	UC-32
Use Case Name	Chat with Patient
Actors	Primary Actors: Neurologist
Description	Neurologist can reply to the messages received from patients. Neurologist selects one the patient and reply to his message.
Trigger	Neurologist clicks on chat icon shown with patient detail.
Preconditions	PRE-1. Neurologist is logged into the system. PRE-2. Patient have sent message to the neurologist.
Postconditions	POST-1. Reply from neurologist is sent to the patient.
Normal Flow	1.0 Neurologist sends a single reply message 1. Neurologist accesses the chat feature within the system. 2. System displays a list of patients who have sent messages. 3. Neurologist selects one of the patients. 4. Neurologist types a message. 5. Neurologist sends message to patient.
Alternative Flows	1.2 Neurologist sends multiple reply messages 1. Return to step 4 of normal flow.
Exceptions	None
Business Rules	None
Assumptions	None

3.6.5 Payment Gateway

Table 3.33: Pay Appointment Fee

Use Case ID	UC-33
Use Case Name	Pay Appointment Fee
Actors	Primary Actor: Patient
Description	After the patient has selected an appointment for booking, the system collects banking details from the patient and processes the payment using the banking system and sends it to the neurologist's bank account.
Trigger	Patient select an appointment for booking and proceeds to pay the appointment fee.
Preconditions	PRE-1. Patient is logged into the system. PRE-2. Patient has selected an appointment for booking.
Postconditions	POST-1. Appointment fee is successfully transferred to the neurologist's bank account. POST-2. Patient's appointment is booked.
Normal Flow	1.0 Patient confirms to process payment 1. System redirects patient to payment interface after he/she has selected an appointment for booking. 2. System prompts the patient to provide banking details including bank name, card number, and cardholder name. 3. Patient enters the required banking details. 4. System confirms from patient to process payment. 5. The system processes the payment using banking system. 6. Banking system processes the transaction and transfers the appointment fee to the neurologist's bank account. 7. The system displays a payment and appointment confirmation to the patient.
Alternative Flows	1.1 Patient cancel to process payment 1. Use case terminates.
Exceptions	E1 Payment process fails 1. The system provides an error message to the patient. 2a. If patient cancels the payment process, use case terminates. 2b. Else if patient requests to try again then return to step 2 of normal flow.
Business Rules	BR-1: Patient's banking details must be validated before processing the payment.
Assumptions	1. Patients have valid banking details and sufficient funds for the appointment fee.

3.6.6 Lifestyle Plans

Table 3.34: Access Diet Plans

Use Case ID	UC-34
Use Case Name	Access Diet Plans
Actors	Primary Actor: Patient
Description	Patients input their dietary preferences into the system. The system generates personalized diet plans based on the patient's disease and entered preferences.
Trigger	Patient clicks on generate diet plan option.
Preconditions	PRE-1. Patient is logged into the system. PRE-2. Patient has been detected/predicted positive for a disease.
Postconditions	POST-1. Personalized diet plan is generated for the patient.
Normal Flow	1. Patient detected positive for a disease accesses the diet plan generation feature in the system. 2. Patient inputs dietary preferences such as vegetarian or non-vegetarian choices. 3. The system generates a personalized diet plan based on the entered preferences and patient's disease.
Alternative Flows	None
Exceptions	None
Business Rules	BR-1: Diet plans must be personalized based on the patient's disease and entered dietary preferences.
Assumptions	1. Patients provide accurate and complete dietary preferences and disease information.

Table 3.35: Access Exercise Plans

Use Case ID	UC-35
Use Case Name	Access Exercise Plans
Actors	Primary Actor: Patient
Description	The system generates personalized exercise plans for patients based on his disease. The exercise plans include step-by-step instructions for each exercise along with links to video tutorials for better understanding and execution.
Trigger	Patient clicks on generate exercise plan option.
Preconditions	PRE-1. Patient is logged into the system. PRE-2. Patient has been detected/predicted positive for a disease.
Postconditions	POST-1. Personalized exercise plans are generated for the patient.
Normal Flow	1. Patient detected positive for a disease accesses the exercise plan generation feature in the system. 2. System generates personalized exercise plans based on patient's disease. 3. Personalized exercise plans with step-by-step instructors and links to video tutorials are displayed to the patient.
Alternative Flows	None
Exceptions	None
Business Rules	BR-1: The exercise plans must be tailored to the patient's disease.
Assumptions	1. Patients have access to the necessary equipment and resources to perform the exercises.

3.6.7 Statistical Analytics

Table 3.36: Access Analytics Reports

Use Case ID	UC-36
Use Case Name	Access Analytics Reports
Actors	Primary Actor: Admin
Description	Admin navigates to the section of analytics reports where system shows him options to view different types of analytics reports.
Trigger	Admin clicks on system analytics report option.
Preconditions	PRE-1. Admin is logged into the system.
Postconditions	POST-1. Options to view different types of analytics reports are displayed.
Normal Flow	1. Admin accesses analytics report feature within the system. 2. System displays options to view different types of analytics reports.
Alternative Flows	None
Exceptions	None
Business Rules	BR-1: System analytics can be accessed by admin.
Assumptions	None

Table 3.37: Access System Solutions Report

Use Case ID	UC-37
Use Case Name	Access System Solutions Report
Actors	Primary Actor: Admin
Description	The system generates system solutions analytics reports for the admin based on neurologist remarks on e-prescriptions indicating whether patients' health is improving or not. The report includes total counts of patients with different health status categories (improving, not improving, getting worse) and evaluates the effectiveness of system-generated solutions based on the comparison of these counts.
Trigger	Admin clicks on system solution report option.
Preconditions	PRE-1. Admin is logged into the system.
Postconditions	POST-1. System solutions report is displayed to the admin.
Normal Flow	1. Admin accesses system solutions report feature within the system. 2. System retrieves e-prescriptions data containing neurologist remarks. 3. System analyzes neurologist remarks to categorize patients into groups: improving, not improving, getting worse. 4. System generates a report showing total patients who have received health remarks and total counts of patients for each health status category. 5. System displays the system solutions report to the admin indicating system solutions effectiveness is positive or negative.
Alternative Flows	None
Exceptions	None
Business Rules	BR-1: The system solutions report must accurately reflect the neurologist's remarks on e-prescriptions. BR-2: The effectiveness evaluation should be based on the comparison of counts of patients' health statuses.
Assumptions	1. Neurologists provide accurate remarks on e-prescriptions indicating patients' health status. 2. The system-generated solutions are the primary intervention for patients' health improvement.

Table 3.38: Access System Rating Report

Use Case ID	UC-38
Use Case Name	Access System Rating Report
Actors	Primary Actor: Admin
Description	The system generates a graphical report on system ratings, displaying a bar graph with ratings ranging from 1 to 5 on the x-axis and the total number of ratings for each rating category on the y-axis. Admin can view ratings and reviews submitted by patients and neurologists.
Trigger	Admin clicks on system rating report option.
Preconditions	PRE-1. Admin is logged into the system.
Postconditions	POST-1. System rating report is displayed to the admin.
Normal Flow	1. Admin accesses system rating report feature within the system. 2. System displays a graphical report displaying a bar graph with ratings (1 to 5) on the x-axis and the total number for each rating on the y-axis. 3. System displays ratings and reviews submitted by patients and neurologists.
Alternative Flows	None
Exceptions	None
Business Rules	BR-1: The graphical representation should display ratings on the x-axis from 1 to 5 and the total number for each rating on the y-axis.
Assumptions	1. Ratings and reviews submitted by patients and neurologists are genuine and reflect their true opinions.

Table 3.39: Access Virtual Consultation Report

Use Case ID	UC-39
Use Case Name	Access Virtual Consultation Report
Actors	Primary Actor: Admin
Description	The system generates monthly reports on the total number of consultations, duration per consultation, and reason (if patient has already detected disease via system) for each consultation.
Trigger	Admin clicks on virtual consultation report option.
Preconditions	PRE-1. Admin is logged into the system. PRE-2: System has collected monthly data including the number of consultations, their duration, and reason for each consultation (if patient has already detected disease via system).
Postconditions	POST-1. Virtual consultation report is displayed to the admin.
Normal Flow	1. Admin accesses virtual consultation report within the system. 2. Admin inputs month for which report is to be generated. 3. System displays the total number of consultations per month, duration per consultation, and reason (if applicable) for each consultation.
Alternative Flows	None
Exceptions	None
Business Rules	None
Assumptions	1. Data collected on virtual consultations is accurate and up-to-date. 2. The system consistently records the duration and, optionally, the reason for each consultation.

Table 3.40: Access Disease Detection Report

Use Case ID	UC-40
Use Case Name	Access Disease Detection Report
Actors	Primary Actor: Admin
Description	The system generates monthly reports summarizing the total number of detections for each disease and the number of positive and negative results for each disease.
Trigger	Admin clicks on 'Disease Detection report' option.
Preconditions	PRE-1. Admin is logged into the system. PRE-2. System has collected monthly data having total number of detections for each disease and the number of positive and negative results for each disease.
Postconditions	POST-1. Disease detection report is displayed to the admin.
Normal Flow	1. Admin accesses the disease detection report within the system. 2. Admin inputs month for which report is to be generated. 3. System displays a report summarizing the total number of detections for each disease and the number of positive and negative results for each disease for the selected month to the admin.
Alternative Flows	None
Exceptions	None
Business Rules	BR-1: The system must accurately collect and summarize data on disease detections.
Assumptions	1. Data collected on disease detections is accurate and up-to-date.

Table 3.41: Access User Report

Use Case ID	UC-41
Use Case Name	Access User Report
Actors	Primary Actor: Admin
Description	The admin can generate a user report for a selected month. The report includes the total number of patients and neurologists who joined the system during that month. Moreover, Admin can enter user-type and detailed information about the selected user type presented in tabular format. The admin can also search for a specific user by entering their email.
Trigger	Admin clicks on 'User report' option.
Preconditions	PRE-1. Admin is logged into the system.
Postconditions	POST-1. User report is displayed to the admin.
Normal Flow	1. Admin inputs the month for which the report is to be generated. 2. System retrieves and displays the total number of patients and neurologists who joined the system during the selected month. 3. Admin selects the user type (doctor or patient). 4. Admin views all details of the selected user type presented in tabular format. 5. Admin searches for a specific user by entering their email. 6. System displays the user details if found.
Alternative Flows	None
Exceptions	None
Business Rules	BR-1: The system must accurately collect and present user data.
Assumptions	1. User data is accurate and up-to-date.

Table 3.42: Access Neurologist Ranking Report

Use Case ID	UC-42
Use Case Name	Access Neurologist Ranking Report
Actors	Primary Actor: Admin
Description	The admin can view a report that categorizes neurologists based on their rankings.
Trigger	Admin clicks on 'Neurologist Ranking Report' option.
Preconditions	PRE-1. Admin is logged into the system. PRE-2. Rankings for neurologists have been collected.
Postconditions	POST-1. Neurologist ranking report is displayed to the admin.
Normal Flow	1. Admin accesses the neurologist ranking report feature within the system. 2. System retrieves and displays neurologists' details by categorizing them on the basis of their rankings.
Alternative Flows	None
Exceptions	None
Business Rules	None
Assumptions	1. Neurologist rankings are accurate and up-to-date.

Table 3.43: Rate Neurologist

Use Case ID	UC-43
Use Case Name	Rate Neurologist
Actors	Primary Actor: Patient
Description	Patients can rate neurologists on a scale of 1 to 5 and submit written reviews to express their satisfaction with the medical services provided.
Trigger	Patient clicks on 'Rate Doctor' option.
Preconditions	PRE-1. Patient is logged into the system. PRE-2. Patient has conducted a virtual consultation with that specific neurologist.
Postconditions	POST-1. Rating and review submitted by the patient are stored in the system.
Normal Flow	1. After virtual consultation session ends, system prompts the patient to rate the neurologist on a scale of 1 to 5. 2. Patient selects the rating. 3. System prompts the patient to submit a written review. 4. Patient submits rating and review.
Alternative Flows	None
Exceptions	None
Business Rules	BR-1: Ratings must be on a scale of 1 to 5.
Assumptions	1. Patients provide honest and accurate ratings and reviews.

Table 3.44: Rate System

Use Case ID	UC-44
Use Case Name	Rate System
Actors	Primary Actor: User (Patient, Neurologist)
Description	User can rate the system on a scale of 1 to 5 and submit written reviews to express satisfaction with the system.
Trigger	User clicks on 'Rate System' option.
Preconditions	PRE-1. User is logged into the system.
Postconditions	POST-1. Rating and review submitted by the user are stored in the system.
Normal Flow	1. User accesses the system interface to rate system. 2. System prompts the user to rate the system on a scale of 1 to 5. 3. User selects the rating. 4. System prompts the user to submit a written review. 5. User submits rating and review.
Alternative Flows	None
Exceptions	None
Business Rules	BR-1: Ratings must be on a scale of 1 to 5.
Assumptions	1. Users provide honest and accurate ratings and reviews.

Table 3.45: View Neurologist Account Requests

Use Case ID	UC-45
Use Case Name	View Neurologist Account Requests
Actors	Primary Actor: Admin
Description	Admin can view neurologist account requests along with all details entered during signup.
Trigger	Admin clicks on 'Neurologist Request' option.
Preconditions	PRE-1. Admin is logged into the system.
Postconditions	POST-1. Neurologist account requests and their details are displayed to admin.
Normal Flow	1. Admin accesses the system interface to view neurologist account request. 2. System displays a list of neurologist account requests. 3. Admin selects a specific request to view detailed information. 4. System displays all details entered by the neurologist during signup.
Alternative Flows	None
Exceptions	None
Business Rules	None
Assumptions	1. Neurologists provide accurate information during signup.

Table 3.46: Approve Neurologist Account

Use Case ID	UC-46
Use Case Name	Approve Neurologist Account
Actors	Primary Actor: Admin
Description	Admin approves the neurologist's account upon successful verification of the PMC registration number by himself.
Trigger	Admin clicks on 'Neurologist Request' option.
Preconditions	PRE-1. Admin is logged into the system. PRE-2. Neurologist has submitted his PMC registration number.
Postconditions	POST-1. Neurologist's account is approved. POST-2. Neurologist receives an email notification about the approval.
Normal Flow	1.0 PMC registration number is valid 1. Admin view a neurologist account request. 2. Admin verifies the PMC registration number provided by the neurologist on his own means. 3. Admin approves the neurologist's account. 4. System marks the neurologist's account as approved.
Alternative Flows	1.1 PMC registration number is not valid 1. Admin disapproves the neurologist's account. 2. System removes neurologist's account request.
Exceptions	None
Business Rules	BR-1. Only PMC verified neurologists must be able to create account.
Assumptions	1. Neurologists provide accurate PMC registration numbers.

Table 3.47: Delete Neurologist Account

Use Case ID	UC-47
Use Case Name	Delete Neurologist Account
Actors	Primary Actor: Admin
Description	Admin deletes a neurologist's account due to policy violations.
Trigger	Admin clicks on 'Neurologist Request' option.
Preconditions	PRE-1. Admin is logged into the system. PRE-2. Neurologist account exists.
Postconditions	POST-1. Neurologist's account is deleted. POST-2. Neurologist receives an email notification about the deletion, including the reason.
Normal Flow	1. Admin accesses the system interface to delete neurologist accounts. 2. Admin selects a neurologist's account due to delete his account. 3. System displays a confirmation prompt to ensure the action is intentional. 4. Admin confirms the deletion. 5. System deletes the neurologist's account.
Alternative Flows	None
Exceptions	None
Business Rules	BR-1. Admin has the authority to delete neurologist accounts.
Assumptions	None

Chapter 4

Design

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. This chapter includes the entire design for the above-mentioned model, as well as an in-depth study of the needs and requirements.

4.1 System Architecture

For the NeuroCare project, we have used a Three-Layered architectural design, which has divided our system into three levels to organize and manage it efficiently [10]. In the presentation tier, both the web and mobile applications serve as user interfaces, facilitating seamless communication between users and the system. Application logic tier will consist of the core logic of our project which will involve machine learning algorithms such as CNN and random forest classifier to detect and predict disease respectively. This level also accepts the user input, interprets medical data and creates the correct diagnostic report and appointment booking and payment logic. The data tier is where all database operations are carried out to store and retrieve the required information with the help of MongoDB and Mongoose to provide interactions with the database through the schemes.

4.1.1 Architectural Structure

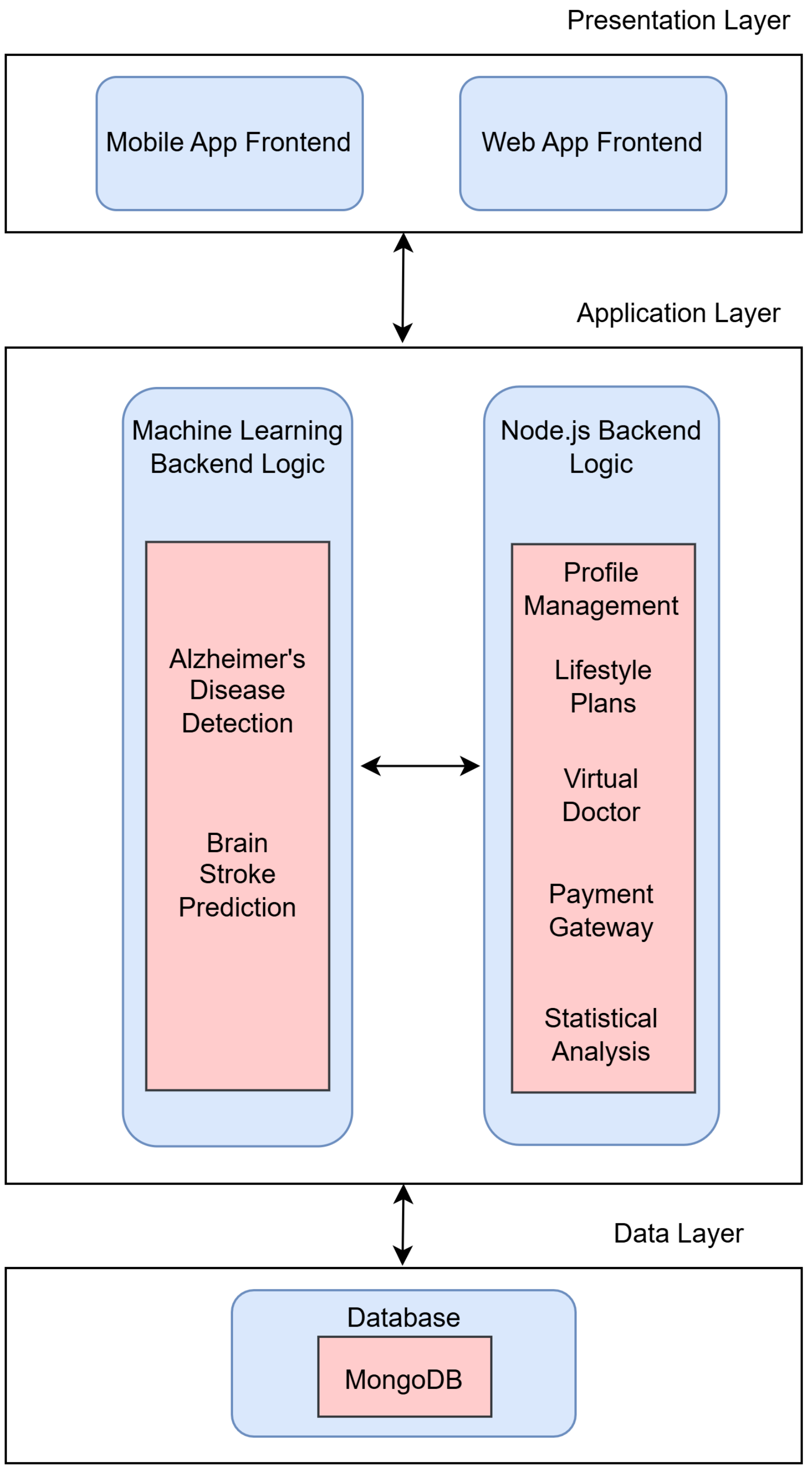


Figure 4.1: Architectural Structure of NeuroCare

4.2 Design Constraints

The design constraints of NeuroCare are as given below:

CON-1: MongoDB is the most flexible, scalable and compatible database that will be used as the primary database.

CON-2: NeuroCare frontend must use React for web app and React Native for mobile app, ensuring unified user experiences and leveraging component-based UI development.

CON-3: The language used in the system is English and hence individuals who do not understand English cannot use the system.

CON-4: For this service, the internet connection should be smooth.

4.3 Design Methodology

In our project, we are applying the procedural programming approach since it is commonly easier to implement, and most especially in simple tasks such as disease-detecting algorithms and online consultation. This is an appropriate project methodology because our project is a medium-sized project. Procedural programming is also likely to provide a quicker execution speed as it does not have the overhead of creating objects and the inheritance that object-oriented programming shares. This performance efficiency would be essential in any task that involves disease detection, in which speed and accuracy are paramount.

4.3.1 Software Process Model

We have concluded that we can use the incremental process model as the most appropriate to develop the NeuroCare system. This choice is informed by the fact that we plan to launch the system through a minimal viable product and sequentially add new features to the system in subsequent releases. In doing so, we will be able to guarantee that the system is developed in an incremental manner, each development building on the base of the preceding ones. This incremental strategy will allow us to test the system at every step of development and identify the possible problems as early as possible and overcome them. Additionally, it offers an organized system of dealing with the complexity of the project despite being flexible enough to meet the changing requirements and stakeholder feedback. On the whole, the incremental process model provides a dynamic and logical approach that is in line with our aim of providing a robust and responsive healthcare business solution with NeuroCare.

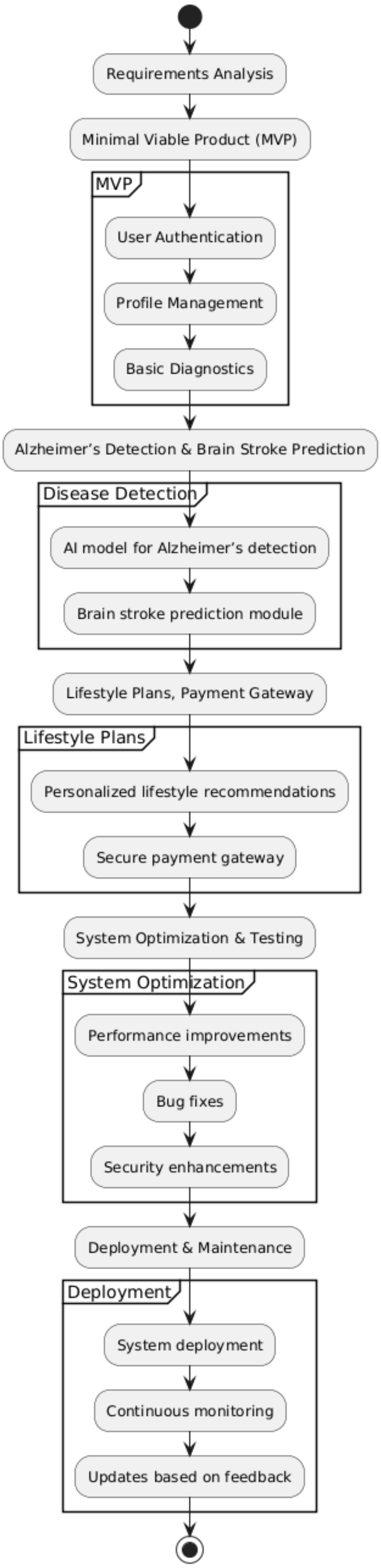


Figure 4.2: Software Process Model Diagram

4.4 High Level Design

The high-level design of the system is described using various diagrams from the Unified Modeling Language (UML) to provide different architectural views [11]. Below are the component and deployment diagrams for NeuroCare.

4.4.1 Component Diagram

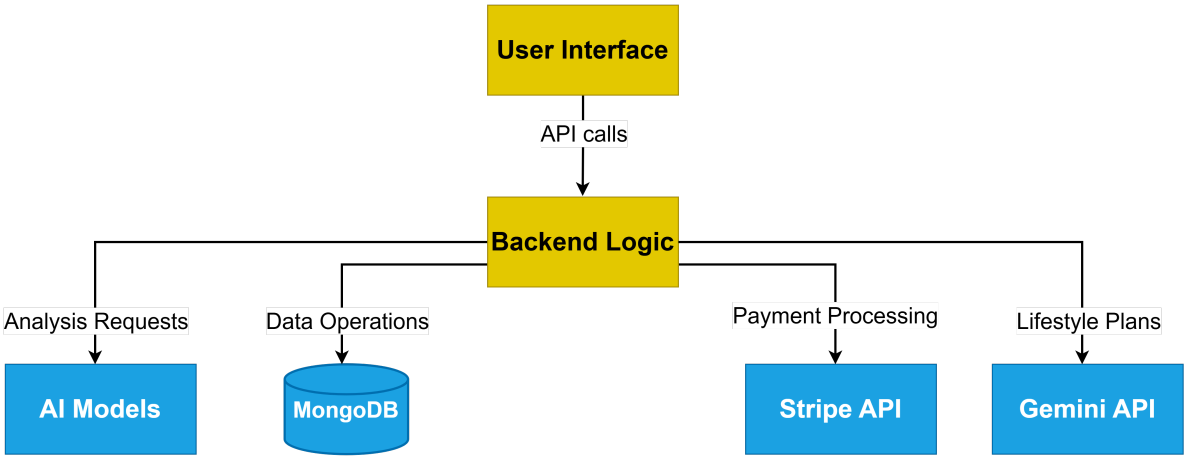


Figure 4.3: Component Diagram

4.4.2 Deployment Diagram

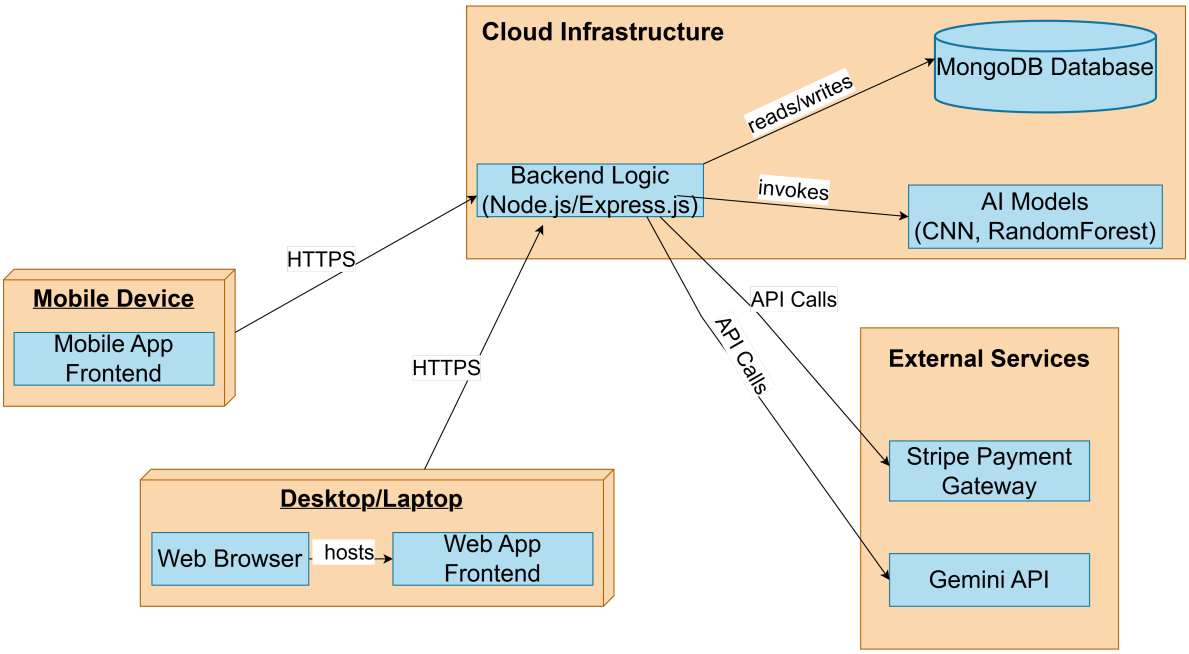


Figure 4.4: Deployment Diagram

4.4.3 Activity Diagram

Below are the activity diagrams of NeuroCare.

4.4.3.1 Register Account by Patient/Neurologist

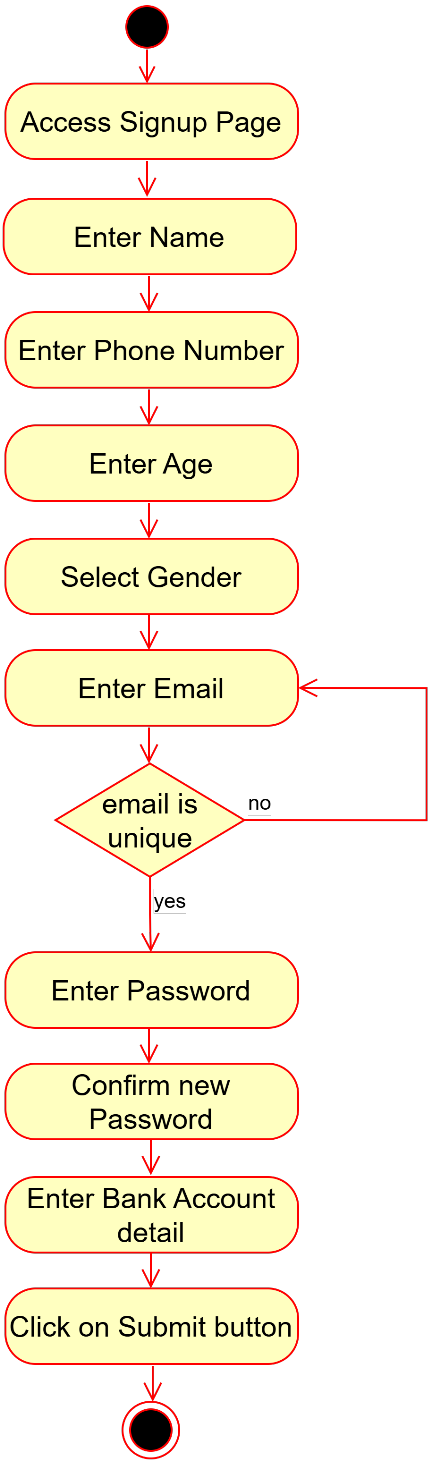


Figure 4.5: Register Account by Patient/Neurologist

4.4.3.2 Login

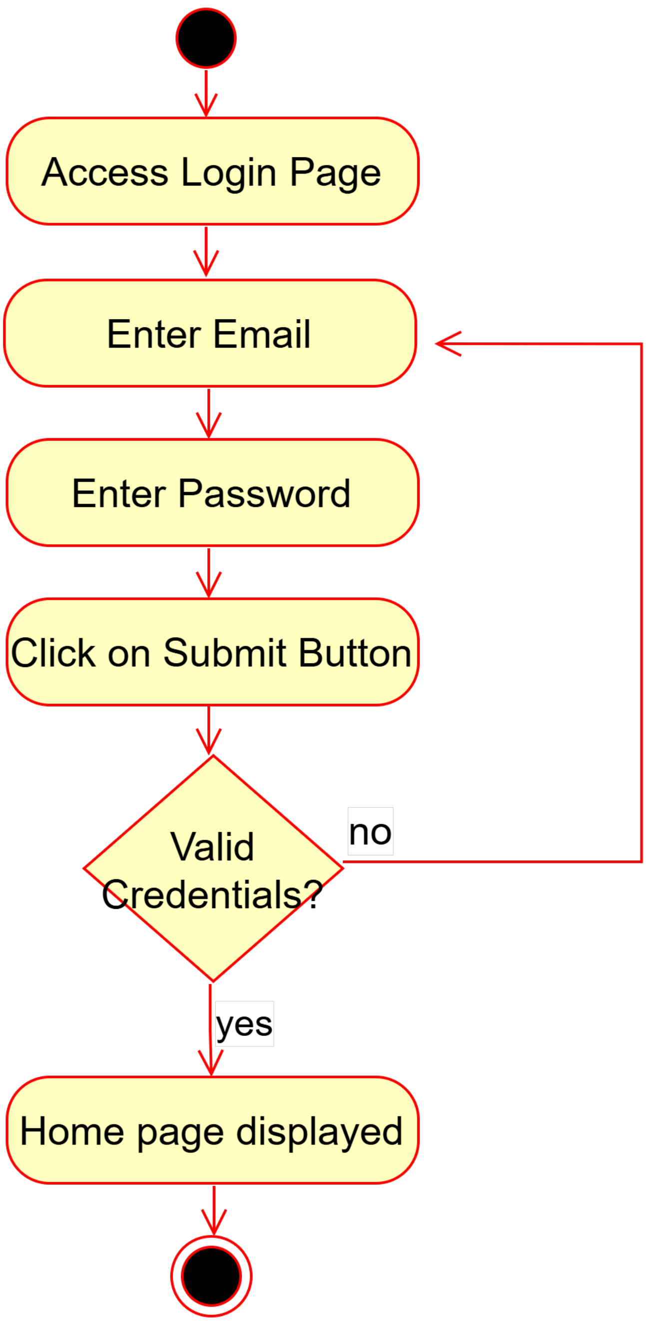


Figure 4.6: Login

4.4.3.3 View and Update Profile Info

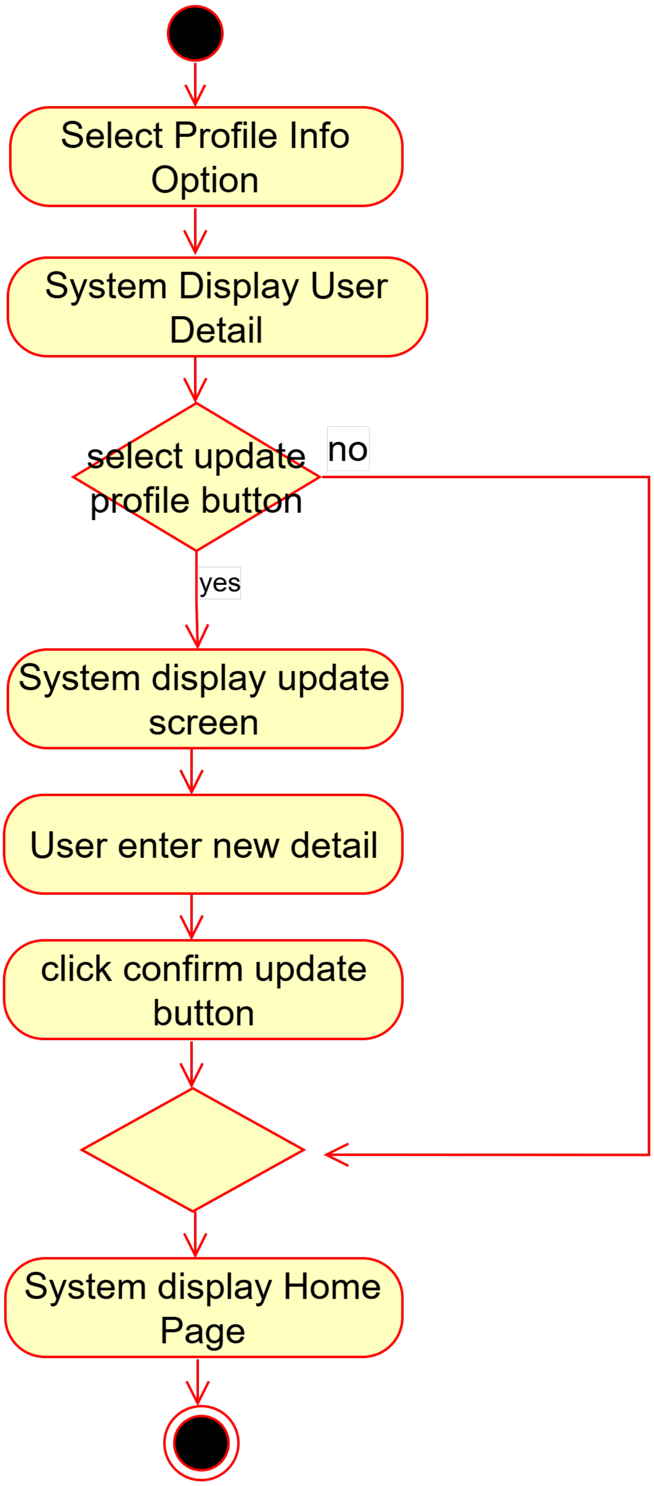


Figure 4.7: View and update profile info

4.4.3.4 Delete Account

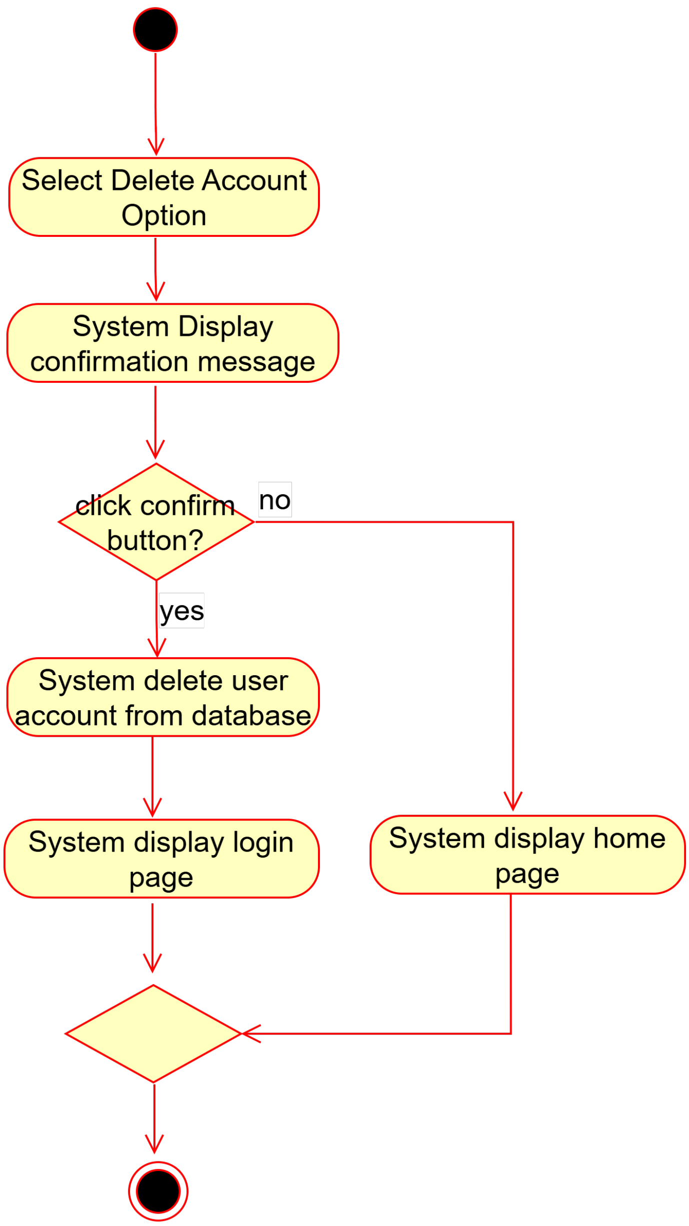


Figure 4.8: Delete Account

4.4.3.5 Update Password

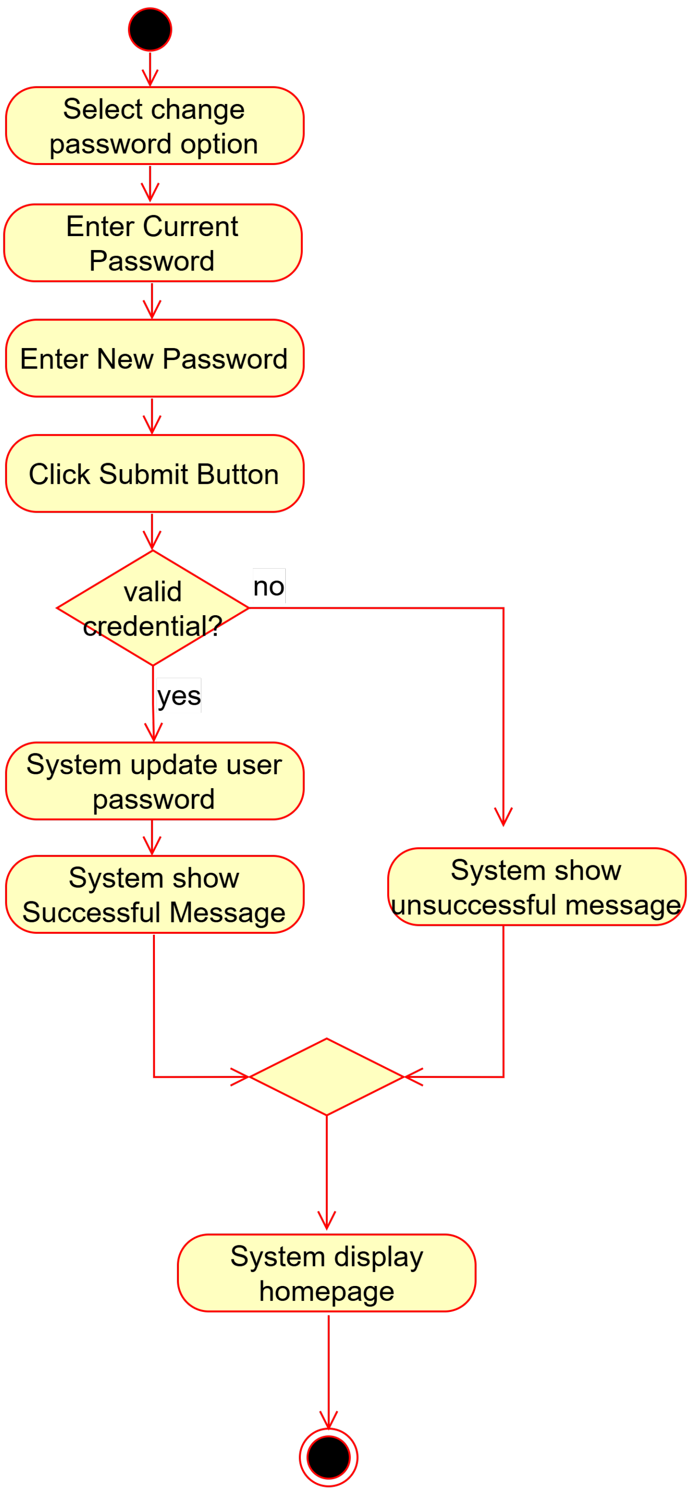


Figure 4.9: Update Password

4.4.3.6 Detect Alzheimer's Disease

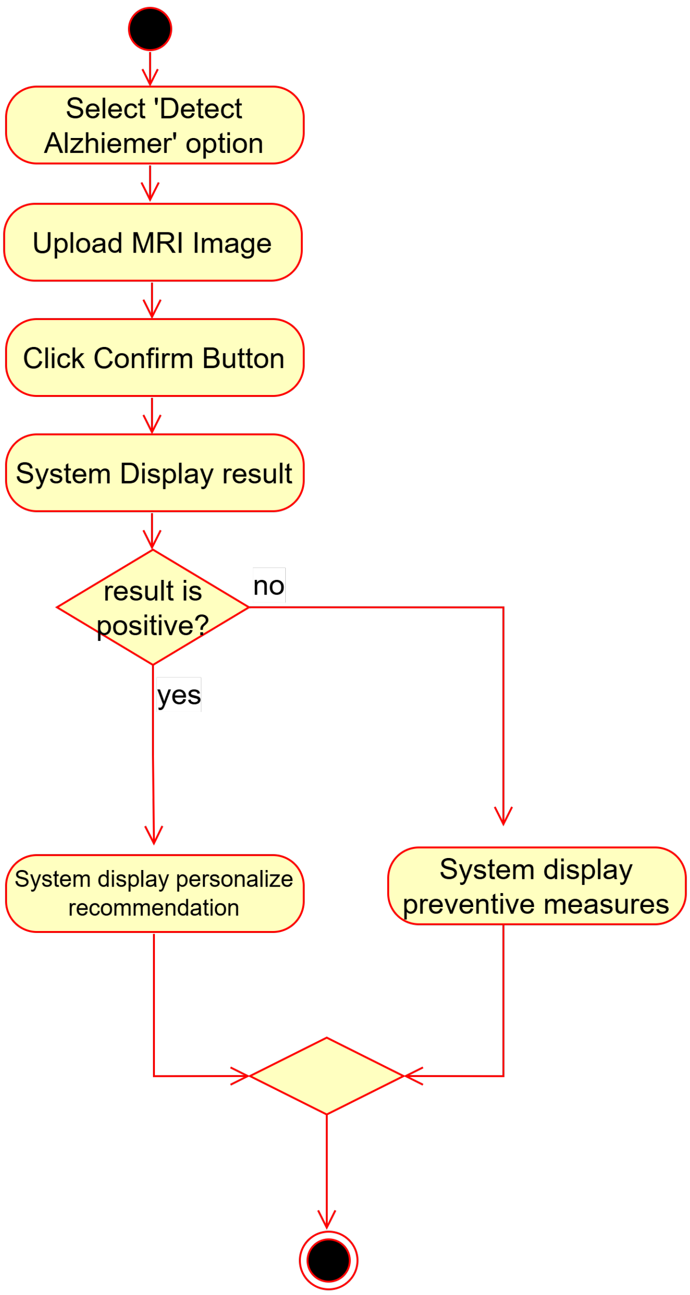


Figure 4.10: Detect Alzheimer's Disease

4.4.3.7 Predict Brain Stroke

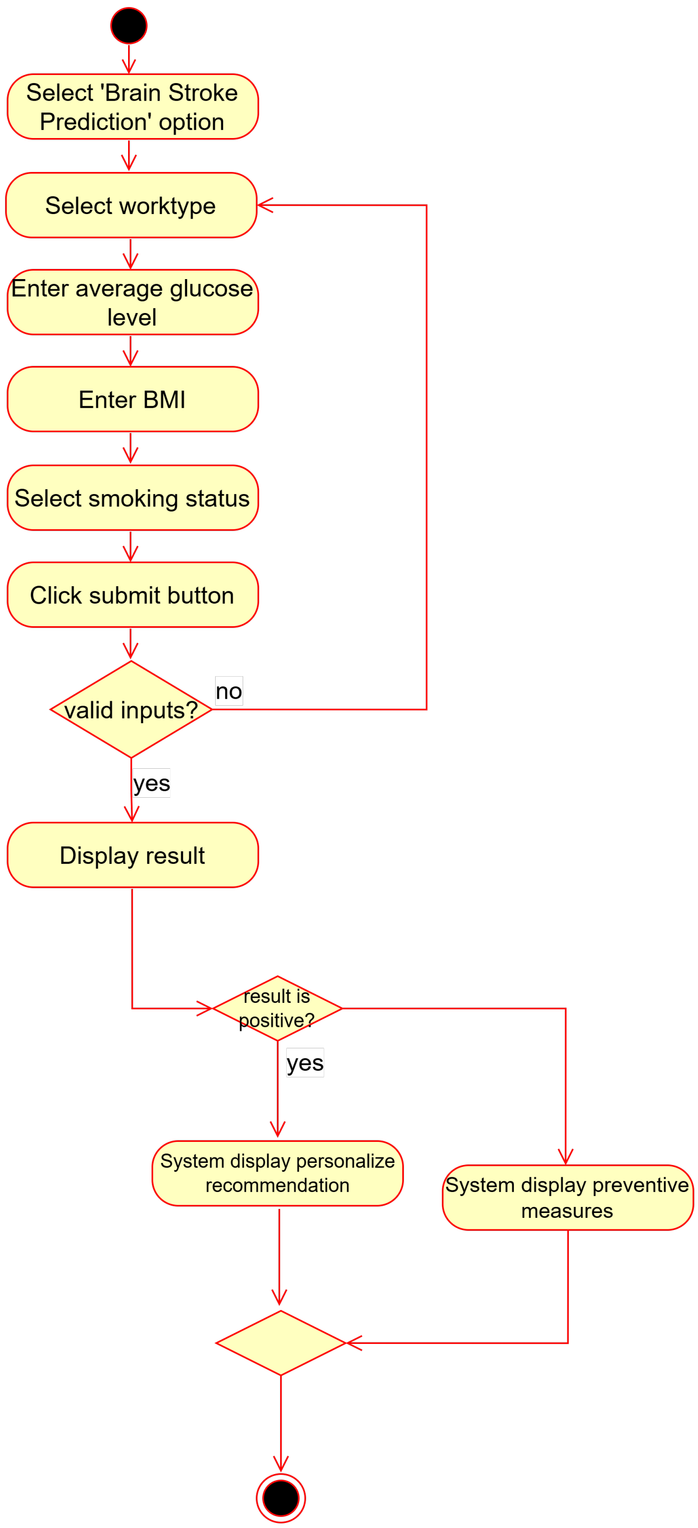


Figure 4.11: Predict Brain Stroke

4.4.3.8 Book Appointment

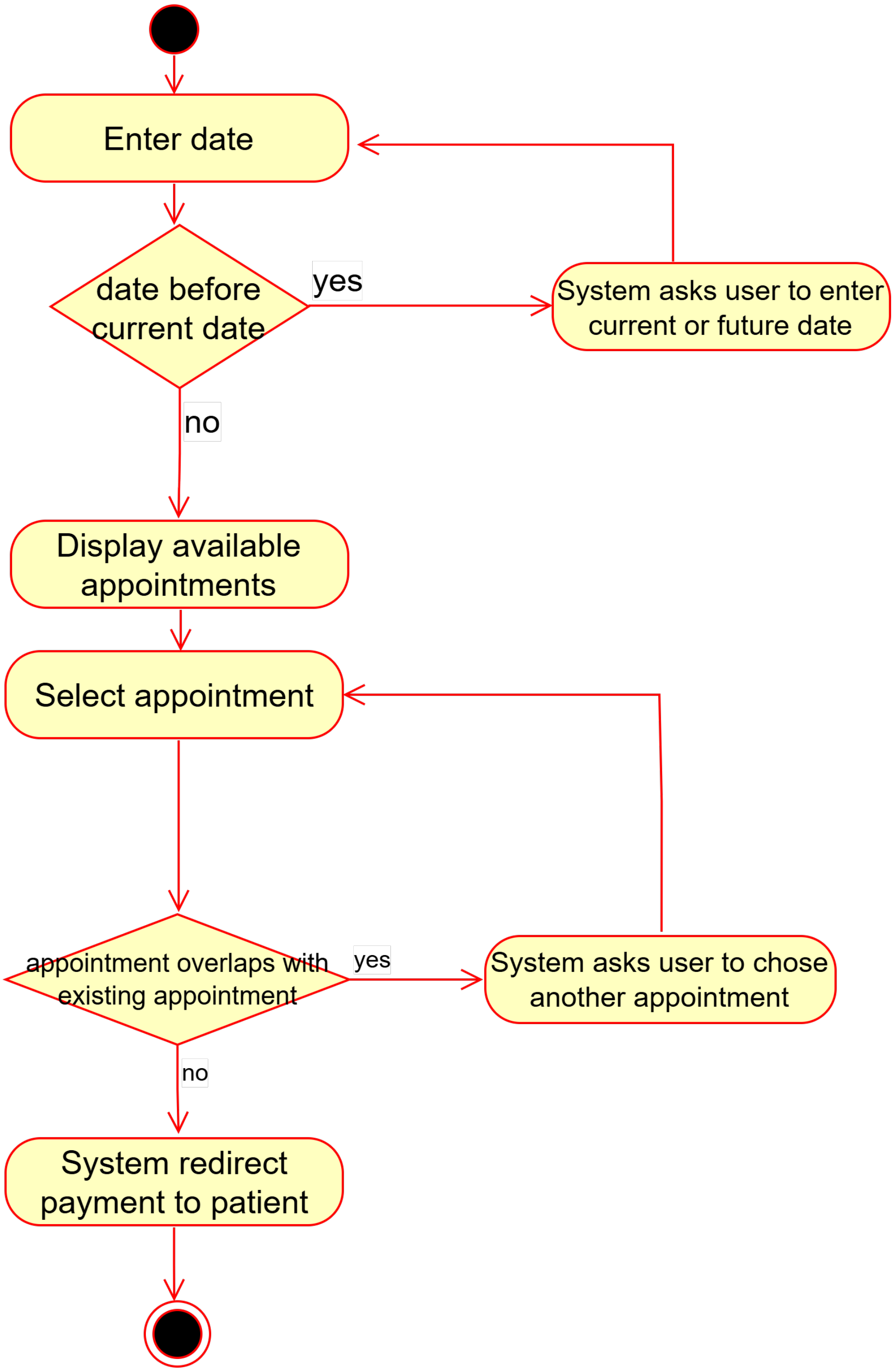


Figure 4.12: Book Appointment

4.4.3.9 Virtual Consultation

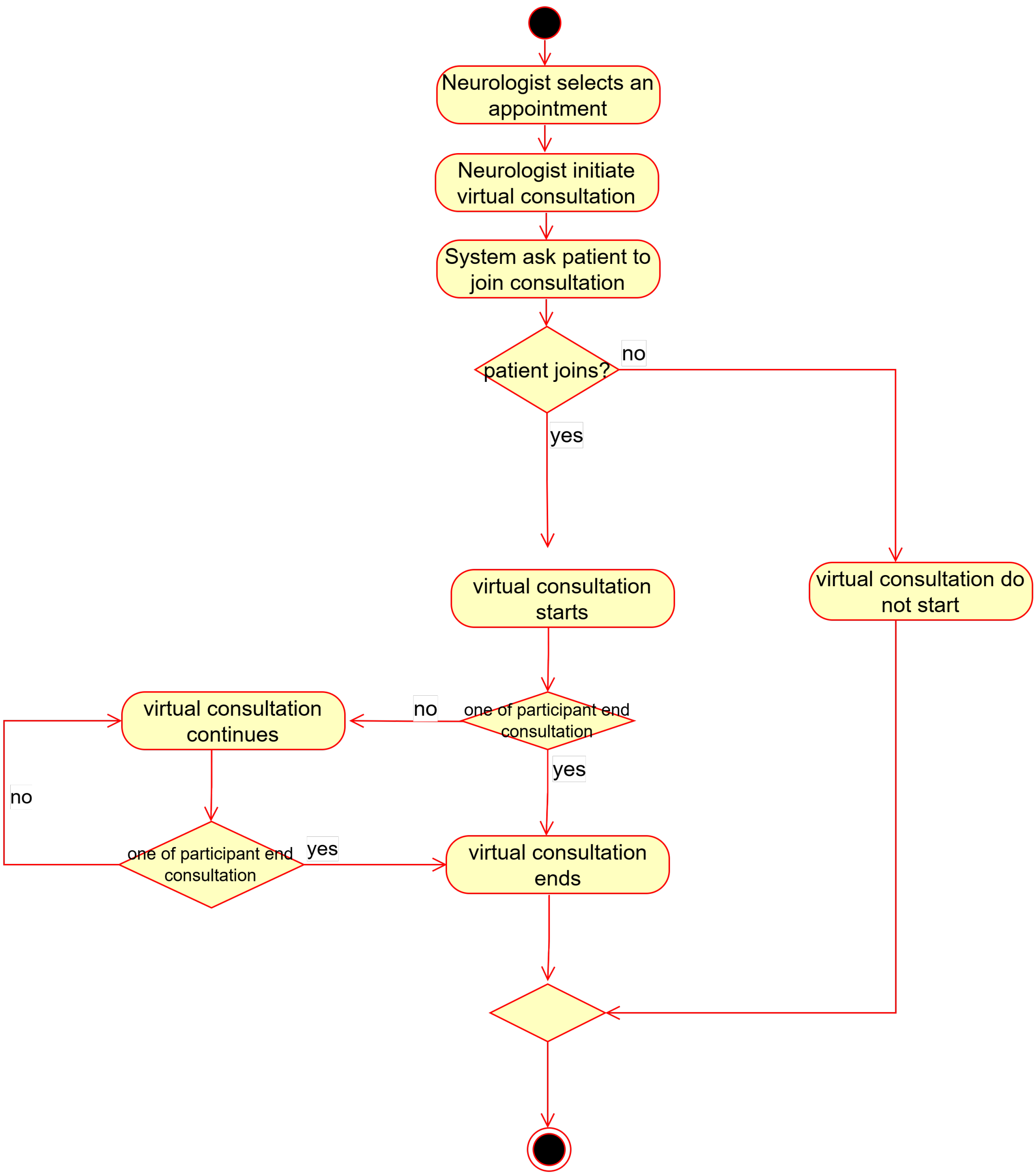


Figure 4.13: Virtual Consultation

4.4.3.10 Access Lifestyle Plans

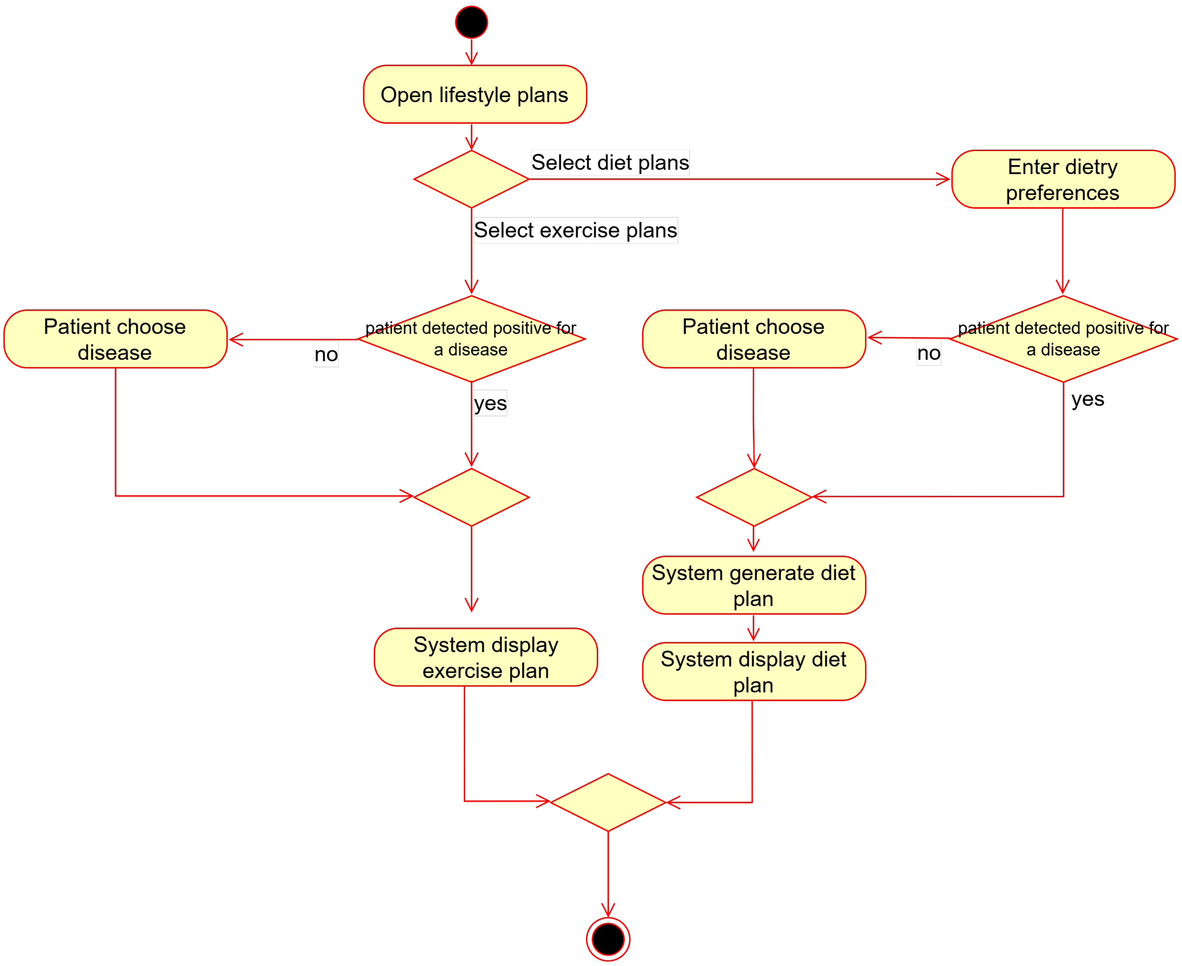


Figure 4.14: Access Lifestyle Plans

4.4.4 Sequence Diagrams

Below are the sequence diagrams of NeuroCare

4.4.4.1 Sequence Diagram for Login

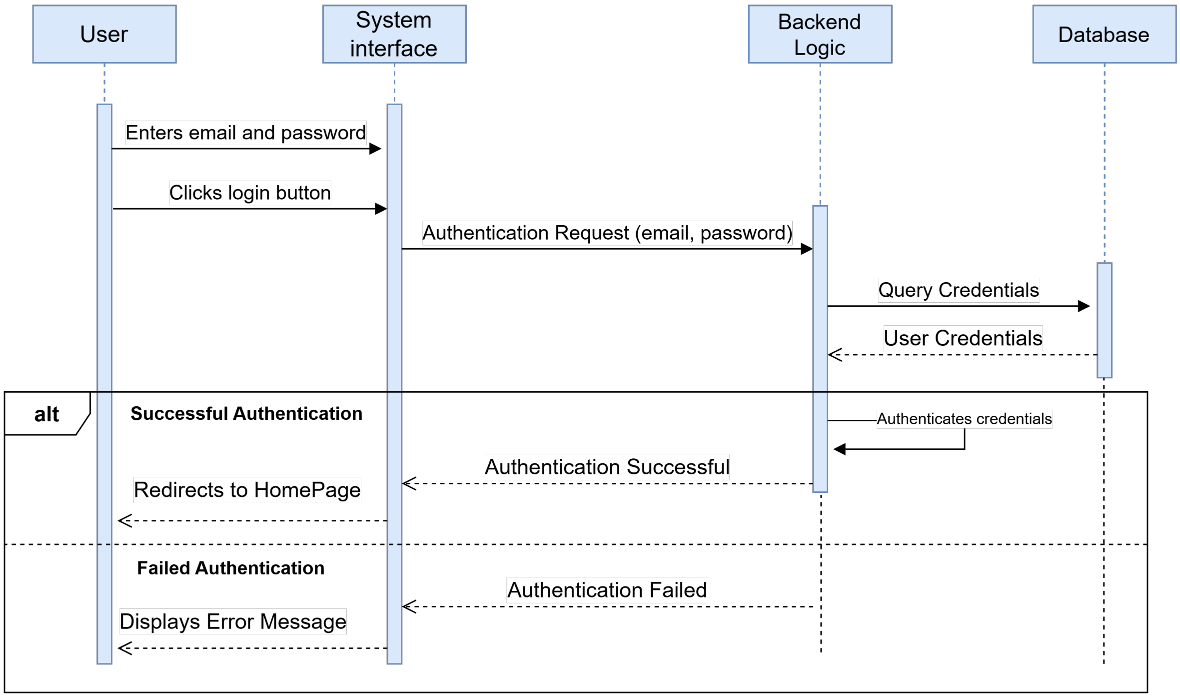


Figure 4.15: Login

4.4.4.2 Sequence Diagram for Alzheimer's Disease Detection

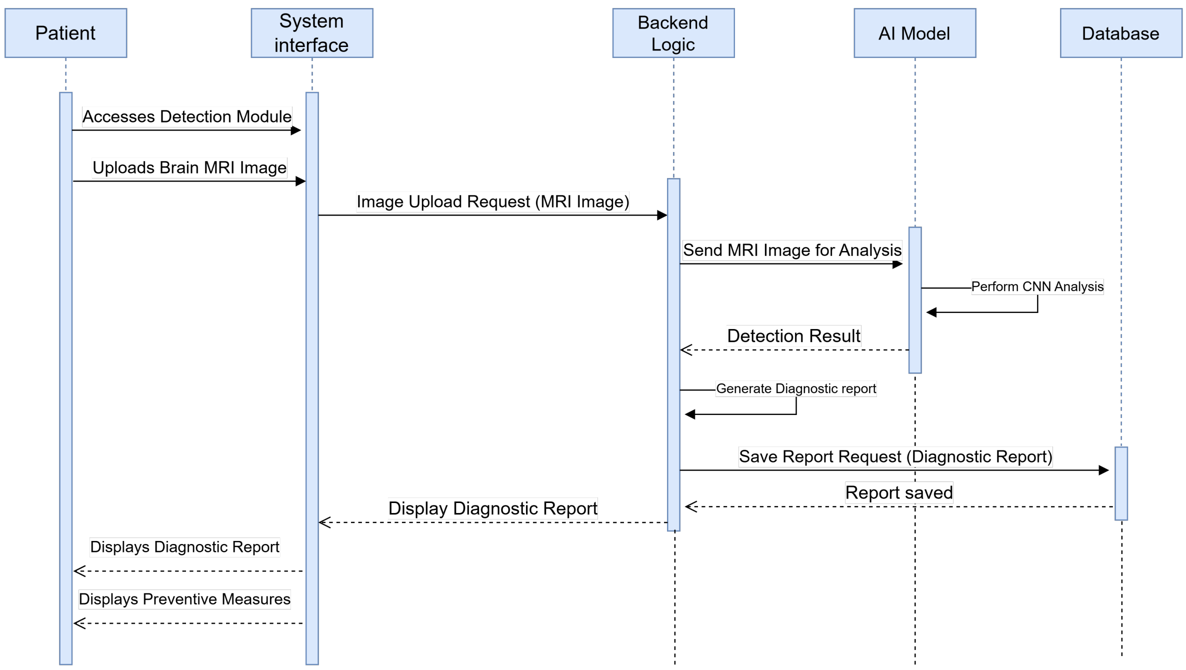


Figure 4.16: Alzheimer's Disease Detection

4.4.4.3 Sequence Diagram for Brain Stroke Prediction

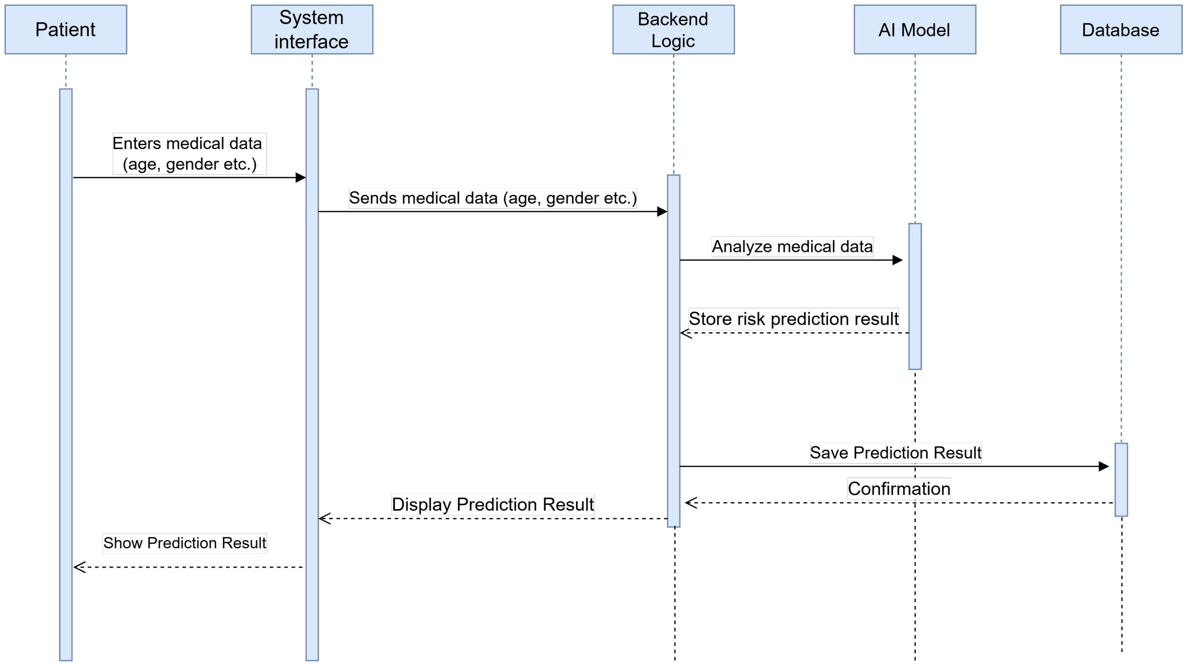


Figure 4.17: Brain Stroke Prediction

4.4.4.4 Sequence Diagram for Personalized Diet Plan

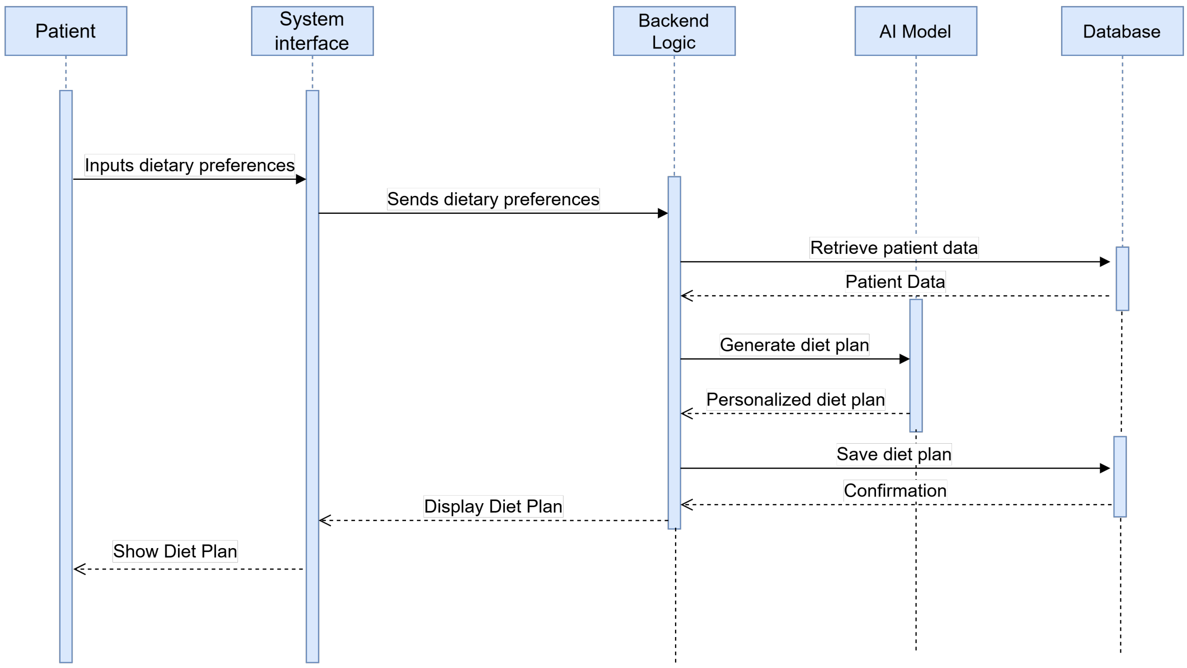


Figure 4.18: Personalized Diet Plan

4.4.4.5 Sequence Diagram for Virtual Consultation Session

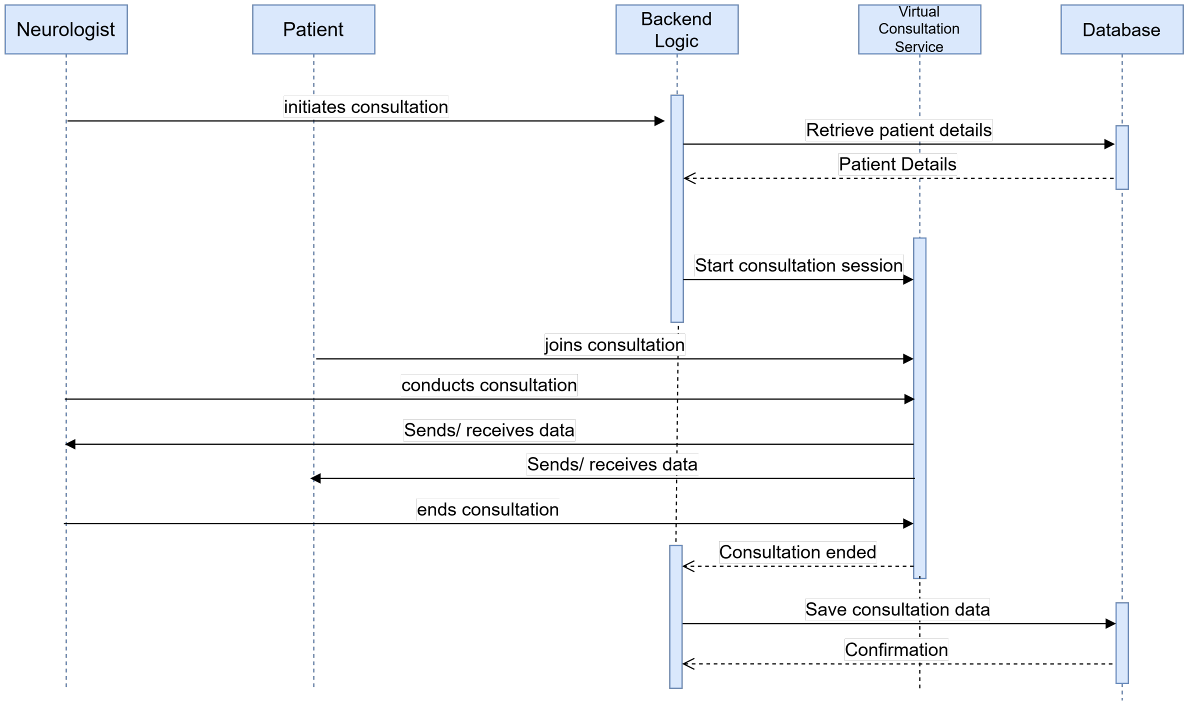


Figure 4.19: Virtual Consultation Session

4.5 Low Level Design

4.5.1 Class Diagram

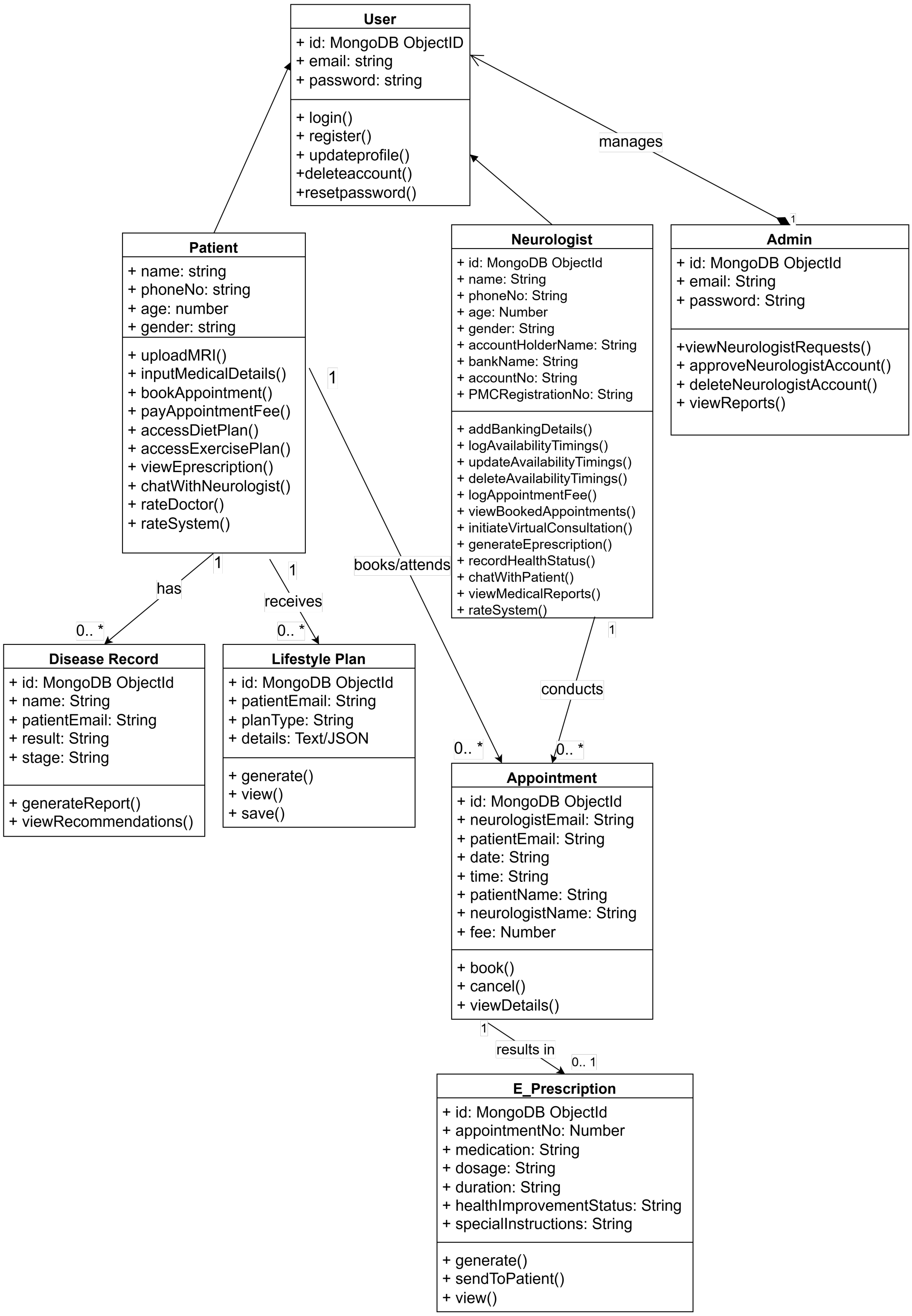


Figure 4.20: Class Diagram

4.6 Database Design

MongoDB is the database solution that our application utilizes as it is based on its NoSQL architecture that is flexible and document-oriented, as opposed to a table-centered database solution [12]. MongoDB stores the information in BSON format that is a binary form of JSON. We use Mongoose in our Node.js backend to declare the database schema with the help of the JSON syntax. The technology stack has provided the flexibility and structure in the management of data, enabling the development and management of the data layer of our application productively.

4.6.1 Data Dictionary

Following are the Data Dictionaries.

4.6.1.1 Data Dictionary for Admin

Table 4.1: Data Dictionary for Admin

Field Name	**Type**	**Description**
Id	MongoDB Object Id	Unique Id for Admin
Email	String	Email of the Admin
Password	String	Password for account security

4.6.1.2 Data Dictionary for Appointment

Table 4.2: Data Dictionary for Appointment

Field Name	**Type**	**Description**
Id	MongoDB Object Id	Unique Id for Appointment
Neurologist Email	String	Email of the Neurologist
Patient Email	String	Email of the Patient
Date	String	Date of the Appointment
Time	String	Time of the Appointment
Patient Name	String	Name of patient
Neurologist Name	String	Name of neurologist

4.6.1.3 Data Dictionary for Disease

Table 4.3: Data Dictionary for Disease

Field Name	**Type**	**Description**
Id	MongoDB Object Id	Unique Id for Disease
Name	String	Name of the Disease
Patient Email	String	Email of the Patient

4.6.1.4 Data Dictionary for E-Prescription

Table 4.4: Data Dictionary for E-Prescription

Field Name	Type	Description
Id	MongoDB Object Id	Unique Id for E-Prescription
Appointment No	Number	Unique Appointment No for the Appointment
Medication	String	Medicines for Patient
Dosage	String	Dose of the medicine
Duration	String	Time to take medicine
Health Improvement Status	String	Health Improvement Status of the Patient
Special Instructions	String	Any special instructions for the patient

4.6.1.5 Data Dictionary for Neurologist

Table 4.5: Data Dictionary for Neurologist

Field Name	Type	Description
Id	MongoDB Object Id	Unique Id for Neurologist
Name	String	Name of the Neurologist
Phone No	String	Phone No of the Neurologist
Age	Number	Age of the Neurologist
Gender	String	Gender of Neurologist
Account Holder Name	String	Account Holder Name for Neurologist's bank account
Bank Name	String	Bank Name for Neurologist's bank account
Account No	String	Account No for Neurologist's bank account
Email	String	Email of the Neurologist
Password	String	Password for account security
PMC Registration No	String	PMC Registration No of the Neurologist

4.6.1.6 Data Dictionary for Patient

Table 4.6: Data Dictionary for Patient

Field Name	Type	Description
Id	MongoDB Object Id	Unique Id for Neurologist
Name	String	Name of the Patient
Phone No	String	Phone No of the Patient
Age	Number	Age of the Patient
Gender	String	Gender of the Patient
Email	String	Email of the Patient
Password	String	Password for account security

4.7 GUI Design

This section of the report describes the appearance and functionality of NeuroCare app as perceived by a user. It demonstrates what you can see on every screen, what you can press or type, and how the application can assist you in getting things done. Our primary ambition was to make the application easy to use and work well on any phone. The NeuroCare application is designed in a simplified and understandable manner. Our attention was paid to several important concepts: screens are tidy and display only what is important; everything in the app functions similarly, which is why one can learn this application without instructions.

4.7.1 Mobile App

4.7.1.1 Home Screen

Home screen where Login and Signup options are shown.

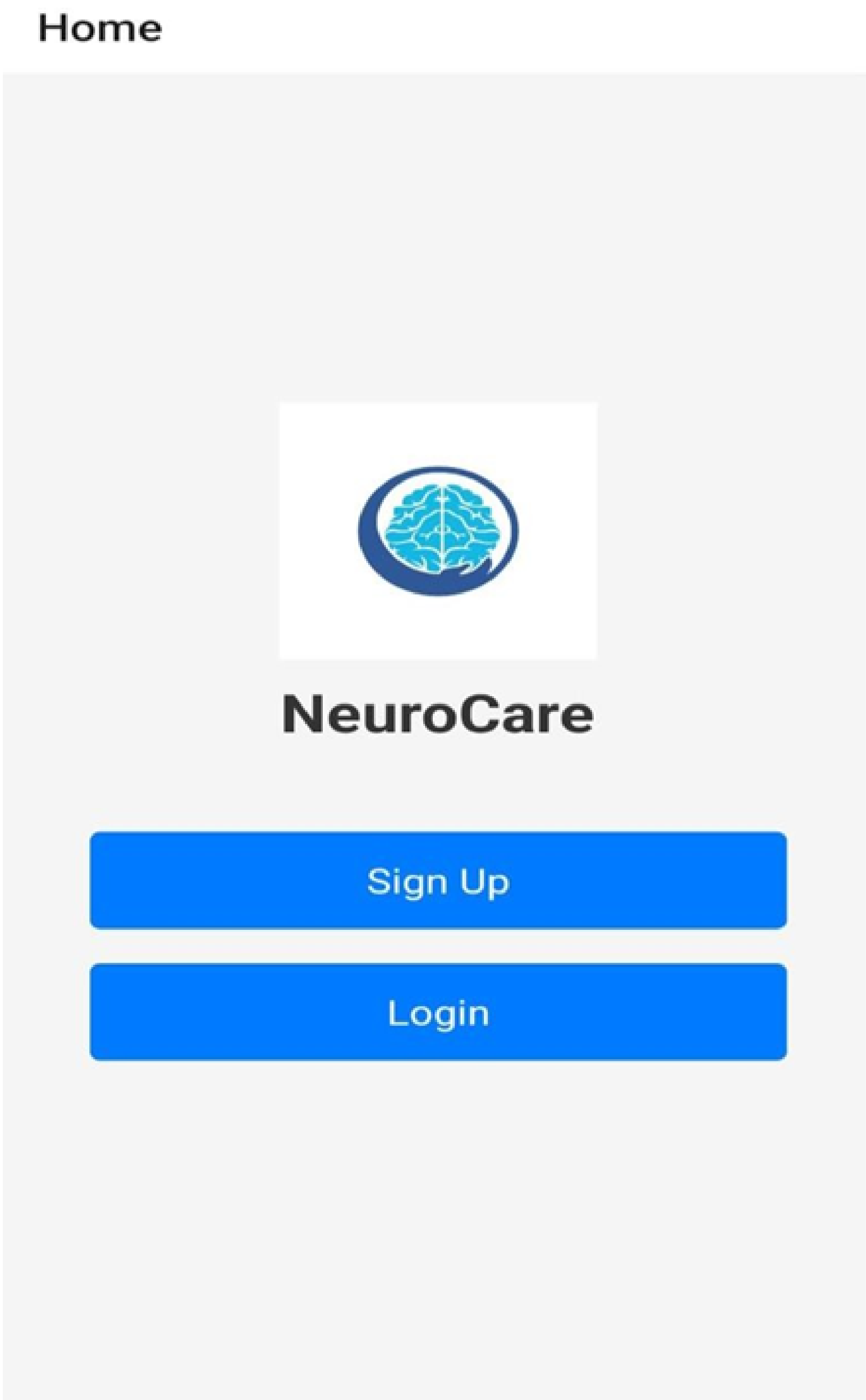


Figure 4.21: Home Screen

4.7.1.2 Login Screen

Our mobile app will have a login screen where a user will enter their email and password to access the system.

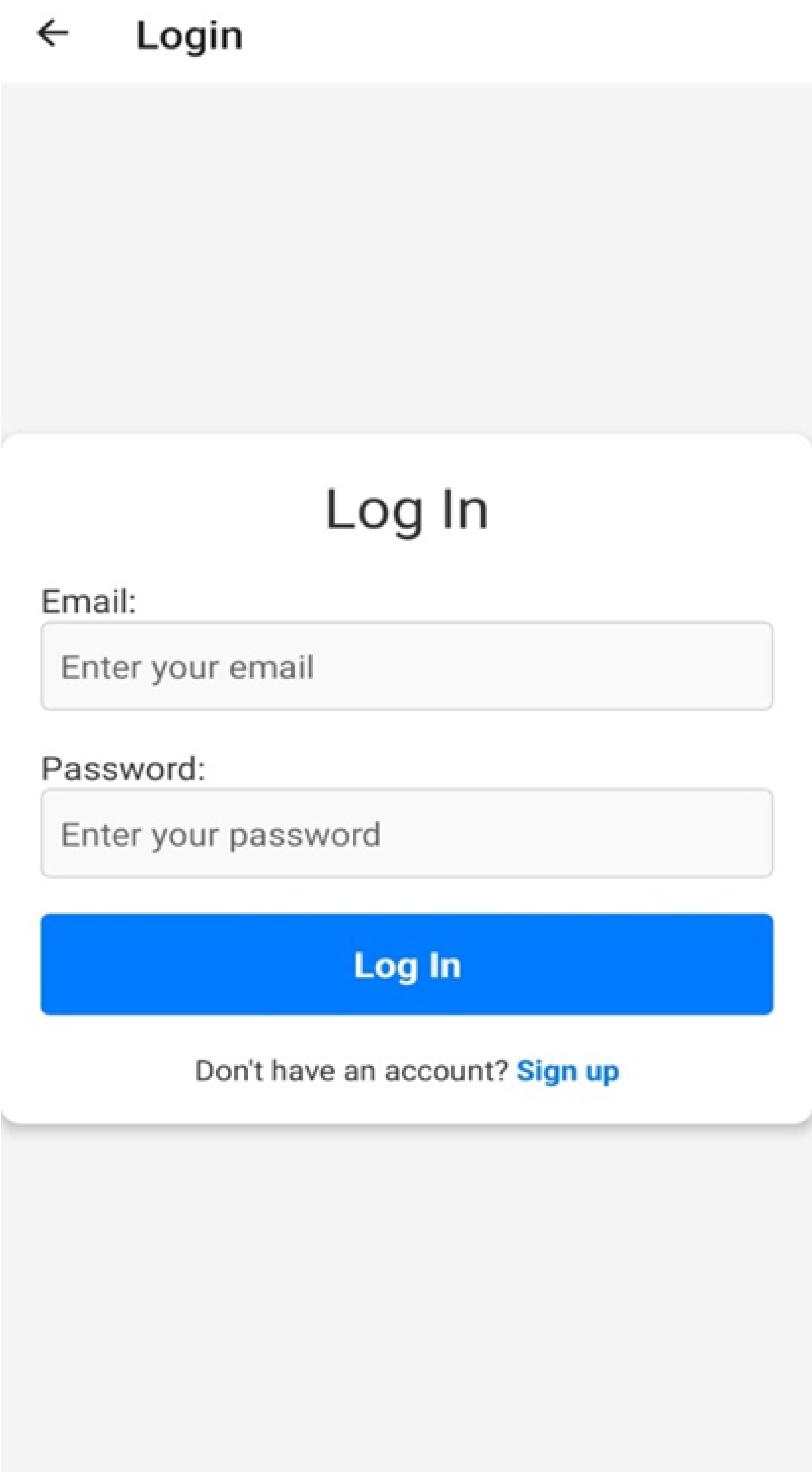


Figure 4.22: Login Screen

4.7.1.3 Brain Stroke Prediction Screen

Mobile app Brain Stroke Prediction Screen where patients predict stroke.

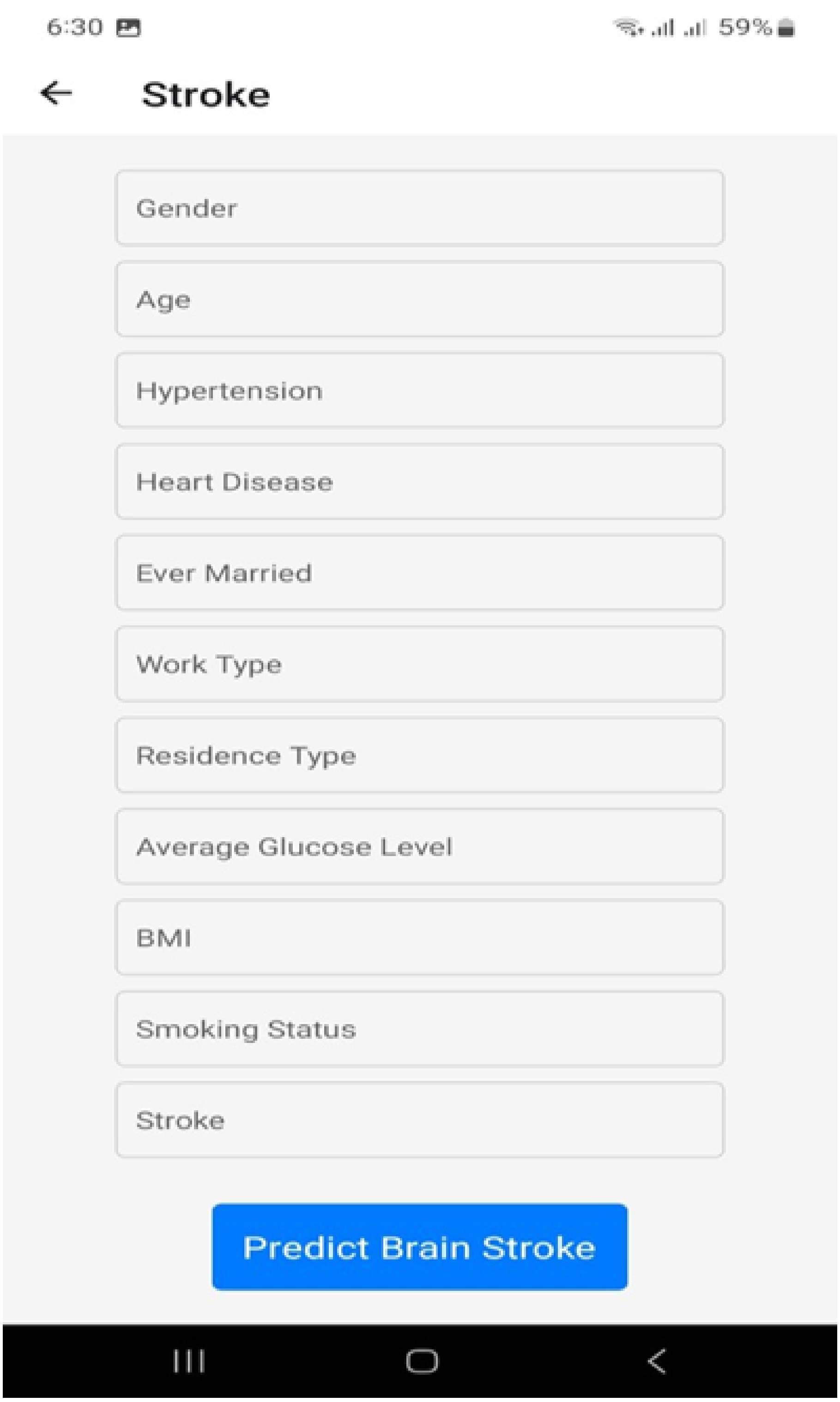


Figure 4.23: Brain Stroke Prediction Screen

4.7.1.4 Diet Plan Generate Screen

Diet plan Screen where the patient enters their preferences, allergies, and disease, and our system generates a diet plan.

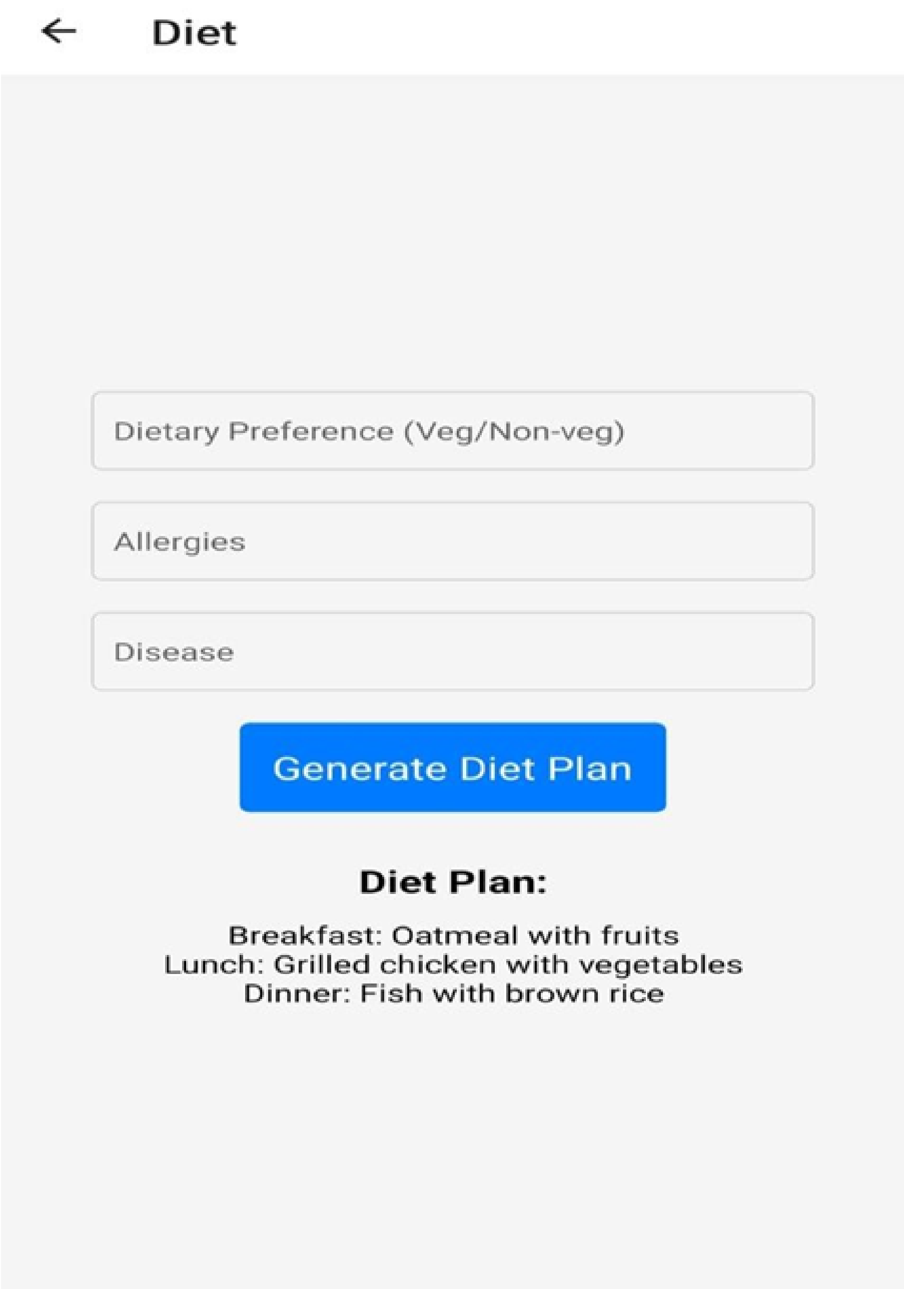


Figure 4.24: Diet Plan Generate Screen

4.7.2 Web App

4.7.2.1 Home Screen

Home Screen of web application where user enters their credentials and initiates virtual consultation.

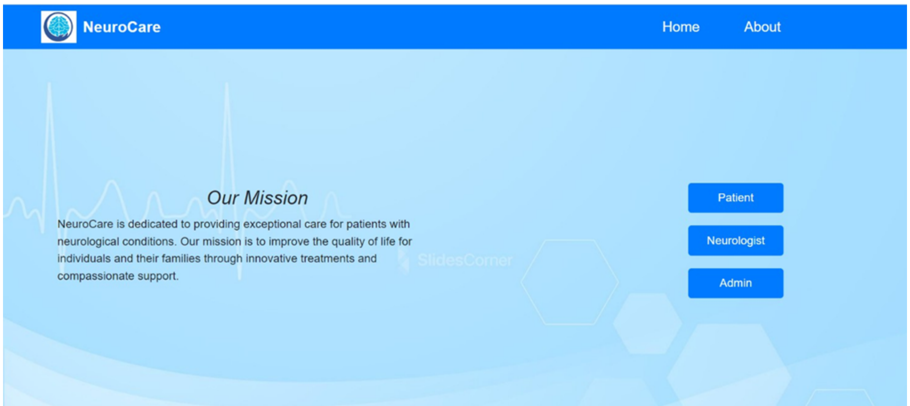


Figure 4.25: Web Home Screen

4.7.2.2 Signup Screen

Signup Screen that allows the user to enter their profile details, which are used to consult them.

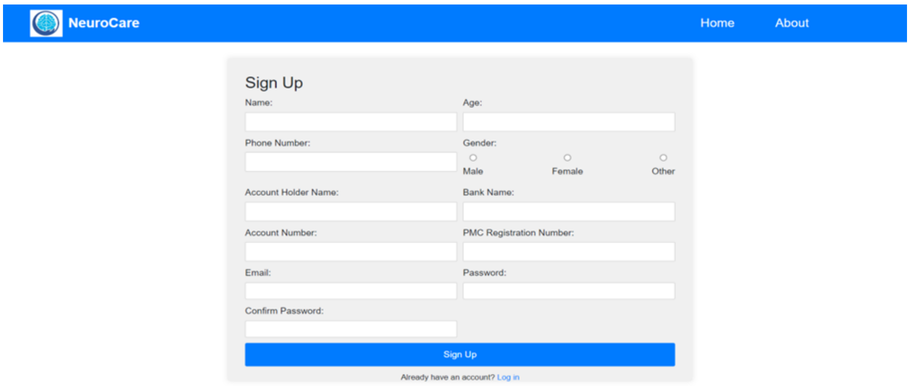


Figure 4.26: Web Signup Screen

4.7.2.3 Book Appointment Screen

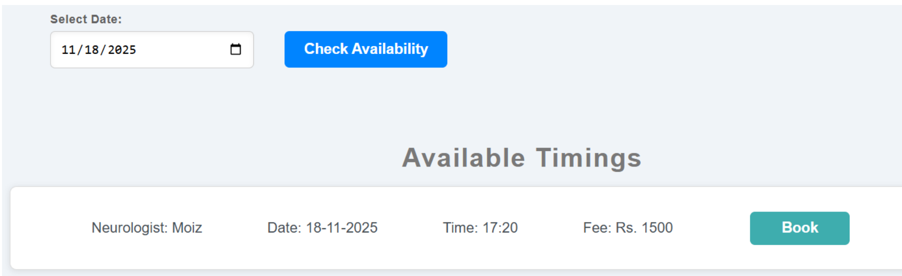


Figure 4.27: Book Appointment Screen

4.8 External Interfaces

This part explains the interfaces by which the NeuroCare system communicates with external parties, such as the users, other software systems, hardware, and communication protocols. These interfaces play an important role in the overall functionality of the system and its integration into the environment of its operation.

4.8.1 User Interfaces

The user interfaces define the interaction of the patients, neurologists, and administrators with the NeuroCare system through the web and mobile applications.

- The system application will be based on the modern web design principles and standards.
- Fonts, icons, labeling of buttons, color schemes, and layouts used in the GUI will all be uniform across all the screens.
- Screen layouts shall be responsive and optimized to work with a minimum screen resolution of 1366x768 pixels.
- Standard buttons on relevant screens shall be used to have common options like; Submit, Cancel and Save.
- The error messages will be presented in an understandable way to help users to recover the errors.

4.8.2 Software Interfaces

NeuroCare is linked with other external software and APIs to provide the whole spectrum of its functionality.

- The system will use TensorFlow to implement machine learning algorithms in the modules of disease detection to properly diagnose the diseases.
- The database management system that the system will utilize will be the MongoDB to store and retrieve patient data in a secure and efficient manner.
- The system will use Node.js and Express.js to operate the backend server, which will allow the application to have smooth communication between the client-side and server-side components.
- The system will utilize React.js as the interface to the web application.
- • The system will also incorporate React Native to create cross-platform mobile apps that will have similar user experiences on the different mobile devices and operating systems.
- **Stripe API**: Used for handling online payments and managing appointment bookings. Functions include ‘createPaymentIntent()‘, ‘retrievePaymentMethod()‘, ‘createSubscription()‘.
- **FastAPI**: A web framework for building RESTful APIs with Python, used for building APIs and handling HTTP requests.
- **Gemini API**: An AI model for generating human-like text and media, used for generating content and understanding context, such as personalized diet plans. Functions include ‘generateGeminiContent()‘, ‘handleGeminiContext()‘.

4.8.3 Hardware Interfaces

The NeuroCare system is primarily a software-based solution.

- The system does not have explicit hardware interfaces to its core functionality.

4.8.4 Communications Interfaces

Communication within and external to the NeuroCare system relies on standard protocols and methods.

- The system shall support communication over standard web protocols (HTTP/HTTPS) for data transmission between client and server components.

Chapter 5

System Implementation

Implementation is the process of moving an idea from concept to reality. The System implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through programming and deployment.

5.1 System Architecture

NeuroCare is implemented based on the three-layered design of architecture splitting the concerns into the Presentation Layer (frontend), Application Layer (backend), and Data Layer (database). This design enables modular development, scalability, and easier maintenance.

5.1.1 System Internal Components

The NeuroCare system is composed of four primary, decoupled components that work in concert:

1. **Frontend Applications (Presentation Layer):** This layer is the user's point of interaction with NeuroCare.

   - **Web Application:** Developed using React JS, this serves neurologists and administrators. It provides a comprehensive dashboard for managing appointments, viewing patient reports, conducting virtual consultations, and accessing system analytics.

   - **Mobile Application**: This is the main patient-facing interface that was created with React Native. It is user-friendly, and patients can create their profiles, post MRI scans to detect Alzheimer's, enter data to predict strokes, receive a personalized lifestyle plan, and communicate with neurologists.

2. **Backend Server (Application Layer):** The core of the system, based on Node.js and the Express.js framework. It has the following obligations:

   - **API Endpoints:** Making a RESTful API available to the frontend applications to use.

   - **Business Logic:** It deals with all the key features such as registering users, making appointments, and generating reports.

- **Authentication & Authorization:** User session management and endpoint security through a Role-Based Access Control (RBAC) system.

- **Service Coordination:** It plays the role of a central point of communication between the database and the machine learning service.

3. **Machine Learning Service:** This is a microservice written in Python and utilizing the Fast API. This isolates computationally intensive AI tasks [13].

    - **Alzheimer's Detection Module:** Hosts the pre-trained Convolutional Neural Network (CNN) model. It takes an endpoint, and it accepts an MRI image and gives out a diagnosis.

    - **Brain Stroke Prediction Module:** Hosts the trained Random Forest Classifier model. It offers a service that receives patient medical information and gives a stroke risk likelihood.

4. **Database (Data Layer):** A **MongoDB** system is used as the primary storage of data. It puts all the persistent data in binary-encoded serialization (BSON, a serialization of blogs that are in the form of JSON), which contains user profiles, medical records, diagnostic reports, appointment information, and e- prescriptions. **Mongoose** is an Object Data Modeling (ODM) library to specify the schema and manage the interactions of the database with the backend server.

5.1.2 Functionality of Components

The main functionality of the system NeuroCare is spread among many interrelated components. Every element has a particular number of tasks to accomplish to make sure that the interaction and the functioning between the patients, neurologists, and the system are smooth. The table below shows the functional details of each of the major components.

Table 5.1: Functionality of Major Components in NeuroCare System

Component Name	Description / Functionality
User Management Module	Handles user registration, login, and authentication for Patients, Neurologists, and Admins. Supports profile creation, password reset, and secure role-based access control (RBAC).
Alzheimer's Detection Module (AI)	Allows patients to upload MRI images which are analyzed using a Convolutional Neural Network (CNN) to detect and classify the stages of Alzheimer's Disease (Normal, Mild, Moderate, Severe). Generates a diagnostic report and preventive recommendations.
Brain Stroke Prediction Module (AI)	Accepts medical data such as age, hypertension, smoking habits, and glucose levels. Processes data using a Random Forest Classifier to predict the likelihood of a brain stroke. Provides a probability-based prediction and preventive guidance.
Lifestyle Recommendation Engine	Generates personalized diet and exercise plans using the Gemini API based on patient preferences, allergies, and medical condition. Aims to promote healthier lifestyles and reduce disease progression.
Virtual Consultation Module (Virtual Doctor)	Facilitates online appointments and live consultations between patients and neurologists. Supports chat, video call, and sharing of diagnostic reports. Neurologists can issue e-prescriptions directly through the system.
Payment Gateway	Integrated with Stripe API for secure payment processing. Allows patients to pay consultation fees and neurologists to receive payments in their linked accounts. Supports transaction history and receipts.
Analytics & Reporting Module	Generates statistical and graphical reports for administrators. Provides insights into system usage, number of consultations, doctor ratings, patient feedback, and model effectiveness. Helps in continuous improvement of the platform.
Database Management Module	Uses MongoDB with Mongoose to handle all CRUD operations for user data, medical reports, appointments, transactions, and logs. Ensures data consistency, backup, and recovery.

5.1.3 Communication Between Components

The NeuroCare system has a service-oriented design in a modular format with each system element communicating via safe RESTful APIs using JSON as the data interchange format [14]. This architecture allows it to be interoperable, scalable, and platform independent between the web client, mobile application, backend services, and AI modules. The flow of the communication is broken down into three significant interactions to guarantee the smooth flow of the coordination:

5.1.3.1 Frontend to Backend Communication

The frontend applications (written in React.js and React Native) interact with the backend server (Node.js + Express.js) with secure HTTPS connections. All the requests and responses are formatted in JSON to make the transmission lightweight.

- **Communication Type:** RESTful API calls over HTTPS
- **Data Format:** JSON

The backend processes the request and returns an appropriate response, such as a success status, error messages, or AI-generated diagnostic reports.

5.1.3.2 Backend to AI Microservices Communication

The backend server communicates with AI microservices through FastAPI endpoints hosted on a separate container or virtual machine. This isolation ensures scalability, fault tolerance, and efficient processing of machine learning workloads.

- **AI Services Involved:**
  - **Alzheimer's Detection Service:** Convolutional Neural Network (CNN) model for MRI classification
  - **Brain Stroke Prediction Service:** Random Forest Classifier for stroke risk estimation
- **Workflow:**
  1. The user input (MRI image or health data) is received by the backend.
  2. It preprocesses and transmits the information to the relevant FastAPI endpoint through a POST request.
  3. The AI service runs inference using pre-trained models.
  4. The result of the prediction (e.g., the stage of Alzheimer's or the risk of stroke) is sent back to the backend.
  5. This result is formatted and put into MongoDB by the backend and informed to the frontend.

5.1.3.3 Backend to Database Communication:

The backend server allows connection to the MongoDB database via the Mongoose library, which allows all Create, Read, Update and Delete (CRUD) operations.

5.2 Tools, Technology, and Development Environment

The system was developed using a range of modern tools and technologies to ensure a robust and feature-rich application.

Table 5.2: Tools and Technologies for NeuroCare

		Version	Rationale
Tools	MS Visual Studio Code	2024	IDE
	PyCharm	2024.3	IDE
	Jupyter	7.3.1	ML Coding
Technology	React JS	18	Front-end Development
	React Native	0.76.4	Cross Platform Mobile App Development
	Node.js	16	Back-end Development
	Express.js	4.21.2	Back-end Development
	MongoDB	8.0	Database Tasks
	ZEGOCLOUD	3.17.0	Video Calling
	Stripe	2024-11-20. acacia	Payment Integration
	Gemini Developer API	2024	AI Powered Diet Plans
	TensorFlow	2.16.1	Machine Learning, Image Processing
	Scikit-Learn	1.5.2	Machine Learning
	Matplotlib	3.9.3	Graph Plotting
	Anaconda	2024.06	ML Packages Management
	Pandas	2.2.3	Data manipulation and analysis
	NumPy	2.1.0	ML Numerical Computation

5.3 Processing Logic/ Algorithms

The following are the implementation details and algorithms of NeuroCare.

5.3.1 Alzheimer's Detection (CNN Algorithm)

Algorithm 1: Alzheimer Disease Detection

Input: Image File (Uploaded brain MRI)
Output: Detection result (String)

```
Define constants:
BATCH_SIZE ← 32
IMAGE_SIZE ← 256
CHANNELS ← 3
EPOCHS ← 50
Load dataset:
dataset ← tf.keras.preprocessing.image_dataset_from_directory(
"Alzheimer_s Dataset", seed=123, shuffle=True,
```

Continued on next page

```
image_size=(IMAGE_SIZE, IMAGE_SIZE), batch_size=BATCH_SIZE)
class_names ← dataset.class_names
n_classes ← length(class_names)
Visualize one batch of images and labels:
plt.figure(figsize=(10, 10))
for image_batch, labels_batch in dataset.take(1):
for i from 0 to 11:
plt.subplot(3, 4, i + 1)
plt.imshow(image_batch[i].numpy().astype("uint8"))
plt.title(class_names[labels_batch[i]])
plt.axis("off")
Split dataset into train, validation, and test sets:
train_ds ← dataset.take(160)
test_ds ← dataset.skip(160).take(20)
val_ds ← test_ds.skip(20)
// Function to partition dataset:
function get_dataset_partitions_tf(ds):
ds ← ds.shuffle(10000, seed=12)
train_size ← 0.8 * length(ds)
val_size ← 0.1 * length(ds)
return ds.take(train_size), ds.skip(train_size).take(val_size), ds.skip(train_size +
val_size)
train_ds, val_ds, test_ds ← get_dataset_partitions_tf(dataset)
Cache and prefetch datasets:
train_ds ← train_ds.cache().shuffle(1000).prefetch(buffer_size=tf.data.AUTOTUNE)
val_ds ← val_ds.cache().shuffle(1000).prefetch(buffer_size=tf.data.AUTOTUNE)
test_ds ← test_ds.cache().shuffle(1000).prefetch(buffer_size=tf.data.AUTOTUNE)
Define data augmentation:
data_augmentation ← tf.keras.Sequential([
layers.RandomFlip("horizontal_and_vertical"),
layers.RandomRotation(0.2)])
Apply data augmentation:
train_ds ← train_ds.map(lambda x, y: (data_augmentation(x, training=True), y))
Build and compile the CNN model:
model ← models.Sequential([
layers.Input(shape=(IMAGE_SIZE, IMAGE_SIZE, CHANNELS)),
layers.Rescaling(1.0/255),
layers.Conv2D(32, (3, 3), activation='relu'),
layers.MaxPooling2D((2, 2)),
layers.Flatten(),
```

Continued on next page

```
layers.Dense(64, activation='relu'),
layers.Dense(n_classes, activation='softmax')])
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
Train the model:
history ← model.fit(train_ds, validation_data=val_ds, epochs=EPOCHS, verbose=1)
Evaluate the model on the test dataset:
scores ← model.evaluate(test_ds)
Predict function:
function predict(model, img):
img_array ← tf.expand_dims(tf.keras.preprocessing.image.img_to_array(img), 0)
predictions ← model.predict(img_array)
return class_names[np.argmax(predictions[0])], round(100 * (np.max(predictions[0])),
Visualize predictions:
plt.figure(figsize=(15, 15))
for images, labels in test_ds.take(1):
for i from 0 to 8:
predicted_class, confidence ← predict(model, images[i])
plt.title(f"Actual: {class_names[labels[i]]}, Predicted: {predicted_class}, Confidence:
{confidence}%")
plt.axis("off")
Save the trained model:
model.save("../models/model_keras")
```

Table 5.3: Algorithm 1: Alzheimer Disease Detection

5.3.2 Brain Stroke Prediction (Random Forest Classifier)

Algorithm 2: Brain Stroke Risk Prediction

Input: Patient's demographic and health data (gender, age, hypertension, heart disease, ever married, work type, residence type, average glucose level, bmi, smoking status)

Output: Brain Stroke Risk Prediction Result (int)

```
Load the dataset:
data ← pd.read_csv('F:/Fyp Project/stoke predict.zip')
Import necessary libraries:
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
Initialize LabelEncoder objects for categorical columns:
label_encoder_gender ← LabelEncoder()
```

Continued on next page

```
label_encoder_residence ← LabelEncoder()
label_encoder_smoking ← LabelEncoder()
label_encoder_married ← LabelEncoder()
label_encoder_work ← LabelEncoder()
Encode categorical columns:
df['gender'] ← label_encoder_gender.fit_transform(df['gender'])
df['Residence_type'] ← label_encoder_residence.fit_transform(df['Residence_type'])
df['smoking_status'] ← label_encoder_smoking.fit_transform(df['smoking_status'])
df['ever_married'] ← label_encoder_married.fit_transform(df['ever_married'])
df['work_type'] ← label_encoder_work.fit_transform(df['work_type'])
Separate features and target variable:
x ← df.drop(columns=['stroke'])
y ← df['stroke']
Split the dataset into training and testing sets:
x_train, x_test, y_train, y_test ← train_test_split(x, y, test_size=0.1, random_state=42)
Print the shapes of training and testing sets:
print(x_train.shape)
print(x_test.shape)
Initialize and train the Random Forest classifier:
rf ← RandomForestClassifier(max_samples=0.95)
rf.fit(x_train, y_train)
Make predictions on the test set:
y_pred ← rf.predict(x_test)
Calculate and print the accuracy score:
accuracy ← accuracy_score(y_test, y_pred)
print(accuracy)
```

Table 5.4: Algorithm 2: Brain Stroke Risk Prediction

5.4 Application Access Security

The system prioritizes patient confidentiality, regulatory compliance, and data integrity.

5.4.1 Security Zones

- All communications occur over HTTPS.
- APIs are protected via JWT (JSON Web Tokens) for authentication.

5.4.2 Authentication and Authorization

- Role-based access control (RBAC) ensures only authorized users (patient, neurologist, admin) can access designated modules.

5.4.3 Data Encryption and Safe Storage

- AES-256 encryption for sensitive data at rest in MongoDB.
- Token-based session management for secure API calls.

5.4.4 Auditing and Access Logging

- Every login, data update, and transaction is logged with a timestamp and user ID.
- Admin dashboard includes real-time activity logs to detect anomalies.

5.4.5 Database Security

- MongoDB Atlas IP whitelisting prevents unauthorized connections.

Chapter 6

System Testing and Evaluation

Evaluation and testing of the system are essential phases in the development process of the NeuroCare system with the aim of ensuring that the application performs as planned, satisfies the requirements, and delivers quality and valid outputs [15]. In this chapter, the methodologies of testing, the nature of testing conducted, test cases and results, and the assessment of system performance and usability are explained.

6.1 Objectives of System Testing

The project was tested well to ensure that it could be moved out of development to a viable product. The objectives of the testing and evaluation were:

- To make sure that all the functional and non-functional requirements would be addressed.
- To ensure the correct functionality of modules such as AI diagnosis, virtual consultations, and payment gateway.
- To check the accuracy and efficiency of the Alzheimer Detection and Brain Stroke Prediction models.
- To determine usability and general user experience.

6.2 Types of Testing Performed

The testing was done at various levels to have a complete system validation. There are major testing categories that are described below.

Table 6.1: Types of Testing Performed

Testing Type	**Description**
Unit Testing	Validates the functionality of individual components or modules, such as APIs, ML functions, ensuring each works independently as expected.
Integration Testing	Ensures that multiple modules or components work together correctly. Focuses on data flow between the frontend, backend, database, and AI microservices.
System Testing	Evaluates the entire integrated system to confirm it meets the specified functional and non-functional requirements in an environment similar to production.
User Acceptance Testing (UAT)	Involves end-users (patients, neurologists, and admins) to validate whether the system meets user needs, expectations, and usability standards.
Performance Testing	Measures system efficiency and scalability.
Security Testing	Examines vulnerabilities and ensures protection of sensitive medical data through secure communication protocols.
Usability Testing	Evaluates the system interface for accessibility, readability, and ease of navigation, particularly for elderly users.

6.3 Graphical User Interface (GUI) Testing

The GUI testing has been conducted to guarantee a clean user experience that is intuitive and highly responsive across all the supported devices.

Table 6.2: GUI Testing

Test Focus	**Methodology**	**Outcome**
Visual Consistency	Manual checks and screenshot comparison against the final UI/UX design specifications.	All branding elements, color palettes, and typography were consistent.
Layout Responsiveness	Testing the application on emulated screens from small smartphones to large tablets.	All major layouts (search, booking forms, profile view) adjusted correctly to screen size without overflow.
Interactive Elements	Validation that all buttons, text fields, and navigation controls were correctly sized, clickable (or tappable), and provided appropriate visual feedback (e.g., ripple effects).	Zero non-responsive or misaligned buttons found in the final build.

6.4 Usability Testing

Usability testing was performed to establish the ease with which the interface of the system could be used by the users. The criteria of evaluation were the clarity of the navigation, readability, responsiveness, and accessibility.

Key Findings:

- The UI design was found to be consistent across devices.
- The older test users were capable of navigating on their own.
- Mean time to complete a complete diagnosis process (upload → predict → report) was under 1.5 minutes.

These results confirm that NeuroCare is an application fitting necessary criteria of usability and accessibility in healthcare apps.

6.5 Software Performance and Load Testing

Software Performance and Load Testing were conducted to analyze the NeuroCare system under different workloads in terms of speed, scalability, stability, and reliability. This was aimed at making the system able to sustain the acceptable response times when the system is at its maximum usage without performance decline.

6.6 Compatibility Testing

The Compatibility Testing was done to make sure that the NeuroCare system works equally well with various devices, operating systems and web browsers. This is because the system will be tailored as a web and mobile accessible healthcare platform and, therefore, users will require the same functionality and performance, no matter what environment they may prefer to use.

Table 6.3: Testing Environment for Compatibility and Performance Evaluation

Category	**Details**
Operating Systems Tested	Windows 11, Android 11
Web Browsers	Google Chrome, Mozilla Firefox, Microsoft Edge
Mobile Devices	Samsung Galaxy S23, Xiaomi Redmi Note, Google Pixel series
Display Resolutions	1366×768, 1920×1080, 2560×1440

The NeuroCare web and mobile applications were fully functionally compatible with all platforms tested. The findings establish the fact that the NeuroCare system is extremely compatible with the major operating systems, browsers, and mobile devices.

6.7 Exception Handling

Exception Handling was introduced and tested to make sure that the NeuroCare system did not go out of control when some unforeseen mistakes or invalid user data arise. Exception management was a critical design choice because the application uses sensitive healthcare data, and the application uses numerous interconnected modules (backend APIs, AI models, payment gateway, and database) to execute its functionality.

Table 6.4: Exception Handling Scenarios and Results in the NeuroCare System

Exception Scenario	Description / Cause	System Response	Status
Invalid User Input	User enters incomplete or incorrect data in registration or prediction form	Displays clear validation message (e.g., "Invalid email" or "Missing field")	Handled
Database Connection Failure	MongoDB temporarily unavailable or connection timeout	Retries connection	Handled
Model Prediction Error	AI model fails to load or process MRI image	Returns message "Unable to process image"; logs technical error for debugging	Handled
Payment Gateway Timeout	Stripe API request delayed or failed due to network issue	Automatically retries once; notifies user to reattempt transaction	Handled
File Upload Error	Unsupported file format or large image upload	Displays alert "Invalid file type or size exceeded limit"	Handled
Unauthorized Access Attempt	User tries to access restricted module	Redirects to login page; logs unauthorized access attempt	Handled
API Communication Error	Backend unable to fetch response from AI microservice	Displays message "Service temporarily unavailable"; logs error for developer action	Handled

6.8 Security Testing

Security tests were carried out to guarantee that user and medical information cannot be compromised by unauthorized access or manipulation.

- **Authentication Validation:** This was to ensure that all routes were secured and only authorized users were granted access to sensitive information.
- **Authorization Validation:** This was to ensure that only authorized users can access the system using Role-based access control (RBAC).
- **Data Transmission:** This was done to ensure that all the communications are secured using HTTPS protocols.

6.9 Functional Testing

Functional Testing was carried out to make sure that the modules of the NeuroCare system could work as per the specification functional requirements. This testing confirmed that every feature is functioning properly and is delivering the desired outcomes and that all of them are working correctly with other modules.

Table 6.5: Results of Functional Testing Conducted on the NeuroCare System

Test Case ID	Module Tested	Functionality Description	Expected Outcome	Result
TC-01	User Registration & Login	Allows users (patients, doctors, admin) to create accounts and access the system	Successful registration and secure authentication	Passed
TC-02	Profile Management	Enables users to update personal and medical details	Data saved accurately in database	Passed
TC-03	Alzheimer's Detection Module	Processes uploaded MRI images to predict Alzheimer's stage using CNN model	Correct stage prediction displayed to user	Passed
TC-04	Brain Stroke Prediction Module	Predicts stroke risk level based on patient data using Random Forest model	Risk result displayed accurately with preventive advice	Passed
TC-05	Virtual Consultation	Allows patients to book and conduct live consultations with neurologists	Appointment successfully created and joined	Passed
TC-06	Payment Gateway	Handles payment processing via Stripe API	Secure transaction with confirmation receipt	Passed
TC-07	Admin Analytics Dashboard	Displays system usage, user data, and prediction statistics	Accurate and real-time analytics displayed	Passed

6.10 Evaluation of AI Models

Because NeuroCare is extremely reliant on AI models to diagnose and predict, it was tested on both models within the standard machine learning metrics of accuracy and reliability [16].

6.10.1 Alzheimer's Detection Model (CNN)

Table 6.6: Performance Metrics for Alzheimer's Disease Detection Model (CNN)

Metric	Result
Accuracy	94.6%
Precision	93.2%
Recall	92.8%
F1-Score	93.0%

The CNN model was found to be accurate in classifying Alzheimer stages, indicating that it is robust in identifying healthy and affected brain MRI images.

6.10.2 Brain Stroke Prediction Model (Random Forest)

Table 6.7: Performance Metrics for Brain Stroke Risk Prediction Model (Random Forest)

Metric	**Result**
Accuracy	92.4%
Precision	90.5%
Recall	91.0%
F1-Score	90.7%

The Random Forest model was effective in predicting risks of stroke and giving valid results in making decisions regarding preventive health care.

6.11 User Acceptance Testing (UAT)

To ensure the system is usable and useful in real world, the system was tested by various users, such as neurologists, patients and system admins.

Table 6.8: User Acceptance Testing (UAT) Criteria and Observations

Testing Criteria	**Observation / Feedback**
Interface Design	Simple and user-friendly
Performance	Fast model response time
Accuracy	Reliable and consistent results
Ease of Use	Minimal learning curve
Overall Satisfaction	Mostly rated the system as “Highly Effective”

User feedback confirmed that the system achieved the major goals and was easily usable by both healthcare professionals and patients.

6.12 Summary

This chapter proved the testing and evaluation measures undertaken on the NeuroCare platform. By undertaking rigorous testing across GUI, usability, security and performance, as well as functional validation and UAT, the system was found to be very functional, user- centric and robust. The comprehensive testing ensured that the final product met both technical specifications and the user expectation of a trustworthy and efficient healthcare platform.

Chapter 7

Conclusion And Future Work

7.1 Conclusion

The main goal of this project, NeuroCare: AI-Driven Early Diagnosis and Prevention of Neurological Disorders, was to design and develop an intelligent healthcare system to help in the early diagnosis of neurological diseases such as Alzheimer's disease and brain stroke. The system is effective in combining contemporary artificial intelligence methods with user-friendly web-based systems to improve the process of healthcare diagnostics.

During the development cycle, the project has shone light on the fact that machine learning and deep learning models can be utilized well to process medical data and images to have a preliminary diagnosis. The Convolutional Neural Network (CNN) was used to power the Alzheimer's Disease Detection module, and a Random Forest classifier to power the Brain Stroke Prediction module.

The system offers a combined platform between patients, doctors and the administrators. Patients can enroll, add medical data, upload MRI to analyze and assess the level of stroke risk, and have virtual meetings with neurologists. The analytics dashboard allows doctors to review the past of patients, give expert advice, and track their trends. The administrators are able to control users, monitor the performance, and review the overall usage statistics of the system.

In-depth testing such as unit, integration, performance, GUI, compatibility and user acceptance testing were conducted to ensure that the NeuroCare system is efficient, data integrity is maintained and produces consistent results under many different conditions. Load testing and stress testing were used to verify the performance of the platform and ensure that it will be stable even at the highest concurrent usage rates. Also, there were strong exception handling and security such as encrypted communication through HTTPS and access control by roles.

All in all, it can be concluded that NeuroCare is a solution that manages to fill the existing gap concerning artificial intelligence and available healthcare. It does not only show how AI can be useful in medical diagnostics but also highlights the value of the user-friendly design, security, and scalability of health care system. This project is successfully implemented and validated, which can be considered a major contribution to the intelligent medical assistance tools to detect neurological disorders.

7.2 Limitations

Regardless of its effective implementation, the project has several limitations that can be considered for further improvement:

- The Alzheimer detection module is only compatible with a limited dataset and image format at the moment.
- The accuracy of the model can also be refined with more heterogeneous and balanced data.
- The system is still lacking advanced patient monitoring capabilities like continuous vitals monitors.
- It is not yet integrated with hospital management systems (HMS) and electronic health records (EHR).

7.3 Future Work

For the future development of NeuroCare, we aim to enhance the user experience by optimizing the app's interface for easier navigation and faster performance. Marketing efforts will focus on targeted campaigns, collaborations with health influencers, and participation in industry events to grow the user base. To generate revenue, we plan to integrate non-intrusive advertisements that align with the app's mission. Additionally, we will expand the platform by introducing an ecommerce section for neurological medicines and supplements, providing users with convenient access to trusted products.

Appendix A

User Manual

Hardware and Software Requirements

- **Hardware Requirements:** Laptop or PC having at least 8 GB RAM, Intel i5 processor (or equivalent), and 20 GB free disk space.
- **Operating System:** Windows 11 / Ubuntu 22.04
- **Programming Languages:** Python, JavaScript
- **Frameworks and Libraries:** TensorFlow, Keras, Scikit-learn, FastAPI, React.js, Node.js, Express.js
- **Database:** MongoDB
- **Tools Used:** VS Code, PyCharm
- **APIs:** Gemini API, Stripe API

Appendix B: Dataset Description

Alzheimer's Disease Detection Dataset

- **Source:** Kaggle — Alzheimer's MRI Dataset
- **Total Images:** 6400+ MRI scans
- **Categories:**
  - Mild Demented
  - Moderate Demented
  - Non-Demented
  - Very Mild Demented
- **Preprocessing Steps:** Image resizing, normalization, and augmentation (RandomFlip, RandomRotation).

Brain Stroke Risk Prediction Dataset

- **Source:** Kaggle — Stroke Prediction Dataset
- **Attributes Used:** Gender, Age, Hypertension, Heart Disease, BMI, Smoking Status, Glucose Level, etc.
- **Preprocessing Steps:** Null value handling, label encoding, and data normalization.

Appendix C: Tools and Technologies Used

Table A.1: Tools/Technologies

Tool / Technology	**Purpose**
Python	Machine learning model development and data preprocessing
TensorFlow / Keras	Deep learning framework for CNN model
Scikit-learn	Random Forest implementation for stroke prediction
FastAPI	Model serving and backend API integration
Node.js / Express.js	Backend service for handling user requests
React.js	Frontend interface for users and doctors
MongoDB	Database for storing user and medical records
Stripe API	Secure online payment handling

References

[1] Mia Yang, Rafael Samper-Ternent, Elena Volpi, Aval-Na'Ree Green, Maya Lichtenstein, Katy Araujo, Pamela Borek, Peter Charpentier, James Dziura, Thomas M. Gill, Rebecca Galloway, Erich J. Greene, Kristin Lenoir, Peter Peduzzi, Can Meng, Jordan Reese, Amy Shelton, Eleni A. Skokos, Jenny Summapund, Erin Unger, David B. Reuben, Jeff D. Williamson, and Alan B. Stevens. The dementia care study (d-care): Recruitment strategies and demographic characteristics of participants in a pragmatic randomized trial of dementia care. *Alzheimer's & Dementia*, 20(4):2575–2588, 2024. Cited on p. 1.

[2] Soraisam Singh, Dulumani Das, Utpal Barman, and Manob Saikia. Early alzheimer's disease detection: A review of machine learning techniques for forecasting transition from mild cognitive impairment. *Diagnostics*, 14:1759, 08 2024. Cited on p. 1.

[3] Afeez Soladoye, Nicholas Aderinto, Mayowa Popoola, Ibrahim Adeyanju, Ayokunle Osonuga, and David Olawade. Machine learning techniques for stroke prediction: A systematic review of algorithms, datasets, and regional gaps. *International journal of medical informatics*, 203:106041, 07 2025. Cited on p. 1.

[4] Keith Johnson, Nick Fox, Reisa Sperling, and William Klunk. Brain imaging in alzheimer disease. *Cold Spring Harbor perspectives in medicine*, 2:a006213, 04 2012. Cited on p. 1.

[5] Hannah Semigran, Jeffrey Linder, Courtney Gidengil, and Ateev Mehrotra. Evaluation of symptom checkers for self diagnosis and triage: Audit study. *BMJ (Clinical research ed.)*, 351:h3480, 07 2015. Cited on p. 5.

[6] Pranto Roy, Dola Robin, and Asad M Khattak. Machine learning for stroke prediction: A review. *International Journal of Advanced Computer Science and Applications*, 13(6), 2022. Cited on p. 5.

[7] Ian Sommerville. *Software Engineering*. Pearson, 10 edition, 2015. Cited on p. 7.

[8] Martin Glinz. On non-functional requirements. In *15th IEEE International Requirements Engineering Conference (RE'07)*, pages 21–26. IEEE, 2007. Cited on p. 10.

[9] Ivar Jacobson, Magnus Christerson, Patrik Jonsson, and Gunnar Övergaard. *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley, 1992. Cited on p. 14.

[10] Len Bass, Paul Clements, and Rick Kazman. *Software architecture in practice*. Addison-Wesley Professional, 3 edition, 2012. Cited on p. 42.

[11] Martin Fowler. *UML distilled: a brief guide to the standard object modeling language*. Addison-Wesley Professional, 3 edition, 2004. Cited on p. 46.

[12] Pramod J Sadalage and Martin Fowler. *NoSQL distilled: a brief guide to the emerging world of polyglot persistence*. Addison-Wesley, 2012. Cited on p. 61.

[13] Sam Newman. *Building microservices: designing fine-grained systems*. O'Reilly Media, Inc., 2 edition, 2021. Cited on p. 72.

[14] Leonard Richardson and Sam Ruby. *RESTful web services*. O'Reilly Media, Inc., 2007. Cited on p. 73.

[15] Glenford J Myers, Corey Sandler, and Tom Badgett. *The art of software testing*. John Wiley & Sons, 3 edition, 2011. Cited on p. 80.

[16] David M Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2011. Cited on p. 84.

Plagiarism and AI Report

turnitin Page 2 of 161 - Integrity Overview
Page 1 of 119 - Cover Page

Match Groups

- **385** Not Cited or Quoted 21%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** 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

8% Internet sources
1% Publications
20% Submitted works (Student Papers)

Top Sources

13% Overall Similarity

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

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text

Match Groups

- **385** Not Cited or Quoted 21%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** 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

8% Internet sources
1% Publications
20% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

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

turnitin Page 2 of 106 - AI Writing Overview

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

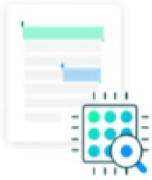
Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Disclaimer
Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.



False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.