Software Requirements Specification

for

NeuroScan

Prepared by

Sourav barman (ASH1925030M)

Institute of Information Technology
Noakhali Science and Technology University

29.04.2024

Table of Contents

List of Figures

List of Tables

1. Introduction

The Software Requirements Specification (SRS) introduction contains the SRS's policy, scope, references, and summary. This document's goal is to gather information about proposed application as we name it as "NeuroScan" and is to give readers a greater understanding of it by outlining the issue statement in great detail. While defining the qualities of a high-quality product, it also emphasizes the advantages and requirements of the participants. Details on the "NeuroScan" can be found on this document.

1.1 Problem Statement

The growing number of brain tumors requires an effective and widely available approach for early diagnosis and treatment. Current brain tumor diagnosis methods frequently require drawn-out steps and may result in postponed intervention, which could negatively affect patient outcomes. Furthermore, it can be difficult for patients to navigate the complexity of treatment options and find appropriate medical facilities and specialists.

In order to address these challenges, we suggest developing a deep learning-powered mobile application “NeuroScan” that will improve brain tumour identification, care, and assistance. The application's goal is to accurately assess patients' brain tumour status by utilising MRI imaging technologies in combination with advanced algorithms. It will also provide users with specific suggestions for medical action and connect them with appropriate healthcare providers and experts.

1.2 Purpose

The primary purpose of the NeuroScan mobile application is to:

- **Enhance Early Brain Tumor Detection:** By leveraging deep learning analysis of MRI scans, NeuroScan aims to identify potential brain tumors at an early stage, facilitating prompt medical intervention and potentially improving patient outcomes.
- **Empower Patients through Mobile Accessibility:** NeuroScan provides a user-friendly mobile platform for brain tumor screening, offering greater accessibility and potentially reducing barriers to early diagnosis.
- **Streamline Care Navigation:** The application guides users through appropriate medical actions based on the analysis results, connecting them with relevant healthcare resources like specialists and hospitals.

1.3 Project Scope

The project is developed to provide brain tumor classification, suggestions of medicine, and many other facilities.

- Users can scan MRI brain images and get result.
- Users can get medicine suggestion and also search the medicine
- Users can view their profile and previous reports.
- Users can view doctor’s profile.
- User can view list of hospitals.

1.4 Glossary

This section provides definitions for all document names, acronyms, and abbreviations. The application domain's terms and concepts are defined.

XML	– Extensible Markup Language
SRS	– Software Requirement Specification
UI	– User Interface
API	– Application Programming Interface
MB	– Megabytes
CNN	– Convolutional Neural Network
ML	– Convolutional Neural Network
MRI	– Magnetic Image Reasoning

1.5 References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

1.6 Overview

An easy-to-use smartphone app called "NeuroScan" responds to the urgent need for better brain tumour diagnosis and treatment. It quickly and accurately determines the tumour status of patients by utilising MRI imaging and deep learning. "NeuroScan" gives patients the confidence and ease to navigate their treatment journey, thereby improving their overall quality of life, by providing them with personalised, unambiguous medical advice and seamless linkages to healthcare professionals..

2. Stakeholders and Characteristics

2.1 Acquirer

- Customers who request, purchase, and/or pay for the software product – IIT, NSTU
- End-users who actually use the product directly or indirectly – General people

2.2 Supplier

- Individual team that are part of the organization that develops the software product – Sourav barman, IIT, NSTU

2.3 Change Control Board

- Responsible for reviewing or proposing changes, approving/disapproving changes, ensuring that approved changes are implemented and validated – Faculty Members, IIT, NSTU

3. Design and Implementation Constraints

In order to ensure the project's success, we used design and implementation limitations. It can also refer to a tool that enables testers and developers to view and interact with the user interface (UI) components of an application.

3.1 Language

User interface Design, usually known as UI Design, is the visual organization of the parts of a website or technological product that a user could interact with. In other words, it is the visual layout of a Mobile Application. On the other hand, the code that enables a computer program or application to run and cannot be viewed by a user is referred to as the back end. The back end of a computer system is where the majority of data and operating syntax are kept and accessed. Typically, the code is comprised of one or more programming languages.

3.1.1 Flutter

Flutter is a framework that helps developers build UI designs for mobile applications. It is a cross-platform mobile application framework that enables us to build applications for both Android and iOS platforms. Flutter is very powerful compared to other technologies. It has impressive capabilities for handling graphics rendering with the help of the Skia library. Nowadays, Flutter is not only used for developing mobile applications but also for desktop and web applications. Its versatility and efficiency make it a popular choice among developers for creating robust, visually appealing user interfaces across various platforms. Additionally, Flutter's extensive documentation and active community support contribute to its widespread adoption and continuous evolution.

3.1.2 Dart

Dart is a programming and works along with flutter. Flutter is just a widget-based UI development framework, but behind the scene dart is used to handle all types of logics. Dart is build using c++ programming language and it also very much faster.

3.1.3 Python

Python is a high-level programming language using to build various types of applications.
Python is also very much popular for developing machine learning based application.
It has tons of libraries for application development like we use tinker for mobile application development, we use Django and flask for web application development and we use pandas, NumPy, sci-kit learn for machine learning application development. Python is important in the machine learning application for the following reasons:

- **Easy to use:** Python is a high-level programming language so that everyone can start will this language.
- **libraries:** Python has vas libraries in the field of machine learning including: pandas, NumPy, matplotlib, sci-kit learn, Keres, TensorFlow etc.
- **Backend approach:** We can not only build machine learning model but also, we can create, manage and handle the backend with the help of python's backed frameworks including Django and flask.

3.1.4 Flask

Flask is backend development framework based on Python. It is a light-weight framework.
Easy to use and helps to build machine learning application quick and fast. Using flask, we can also make API that helps to make a connection between mobile application and web applications. The deployment of flask-based machine learning application is also quite very convenient.

3.2 Server-Side Technology

When an application is used, server-side development refers to the processes that happen in the background. Databases, scripting, website architecture, backend logic, APIs, and servers are the main topics covered.

3.2.1 Firebase

Firebase is a backend technology owned by google. The is very much robust. With the help of firebase, we can build any types of application using their APIs. Firebase provides NoSQL database that is based in the forms of collections. Firebase provide sort of APIs that provides to perform CRUD operations in the database. Firebase also provides real-time database. For user authentication, we can also use firebase Auth.

3.2.2 Flask

Flask is a backend development framework. It is very light-weight. Using flask, we can make ML based application very quickly and fast.

4. Requirement Specification

4.1 Functional Requirement

Functional requirements are those that serve as examples for the system's internal operation, its description, and an explanation of each subsystem. It comprises of the task that the system should complete, the associated processes, the data that the system should store, and the user interfaces.

4.1.1 User registration and login

FR-1	User (any types of users) google signup		
Description	User had better google signup with the account. If user already login to their account on the device, then user just need to select their account and he/she auto login to their account.		
Stakeholders	User (any types of users), System	**Priority**	High

4.1.2 Scan Image

FR-2	User can upload or capture image of MRI brain tumor image		
Description	To scan the image, users need to upload the image first. Then system takes some times to scan the image. At this time a circular loading indicator indicate the processing flow. After scanning completed, system will show the image with classify the brain tumor.		
Stakeholders	Users	**Priority**	High

4.1.3 Search Doctor

FR-3	Users search doctor by name
Description	The user has to google signup and goes to home page. Here user search by doctor name. System will show doctors information in the next page

Stakeholders	Users, System	**Priority**	High

4.1.4 Search medicine

FR-4	User search medicine by name		
Description	The user has to google signup and goes to home page. Then need to visit the medicine page. Here user search medicine by name. System will show medicine information.		
Stakeholders	Users, System	**Priority**	High

4.1.5 Find hospital

FR-5	The system shows the hospitals and the user will be able to see it.		
Description	The user can see the hospitals after entering home page. Then click to the hospitals button and the system will provide a detailed information of hospitals.		
Stakeholders	User, System	**Priority**	High

4.1.6 Report Generate

FR-6	The system will generate a brain tumor report with all required information.		
Description	After scanning the image, system will generate a report. Brain tumor information will be provided in this report.		
Stakeholders	System	**Priority**	High

4.1.7 View Reports

FR-7	The system will provide brain tumor report and patients required information

Description	The user has to google signup and goes to home page. Here user need to visit the report option. Here user find all the reports of the user		
Stakeholders	System	**Priority**	High

4.2 Data Requirement

In our application, firebase handles all the information of users and medical reports. Along with it also store the hospital and medicine information. Users can search medicine the and doctor when they needed.

4.2.1 Searching Doctor and Medicine

DR-1	Searching doctor and medicine using their name		
Description	An authentic user can search doctor and medicine typing their name on the search bar		
Stakeholders	Users	**Priority**	High

4.3 Performance Requirement

It is important that maintain the performance of the system. To ensure the best performance of the system we must maintain the following steps:

4.3.1 Speed and Latency Requirements

PR-1	Faster report generating		
Description	After uploading the image, the image is detected by our machine learning model which deploy in a backend server. Here need to get the response faster so that the latency become reduce		
Stakeholders	User	**Priority**	High

4.3.2 Safety Critical Requirements

For our project there is no safety critical requirements.

4.4 Maintainability and Supportability

The term "maintenance" describes how simple it is to fix, enhance, and comprehend software code. After the user has received the product, the software maintenance phase of the software development cycle begins.

4.4.1 Maintenance Requirements

MR-1	Develop maintainable code		
Description	Maintainability must be ensured so that it can be modified later and will be readable.		
Stakeholders	Developer	**Priority**	High

4.4.2 Supportability Requirements

This system satisfies the supportability requirements for testability, maintainability, compatibility, configurability, serviceability, and install ability.

4.5 Security Requirements

Information security is far more crucial for a system to gain user's trust. Here are some security requirements are given below:

4.5.1 Access Requirements

The system will apply some authorization approaches when granting access to information to make sure the right user is using the right data.

4.5.2 Integrity Requirements

Integrity requirements relate to a security system that ensures an expectation of data quality. It also ensures that no data on the system will ever be exposed to malicious modification or accidental deletion.

4.6 Usability and Human Integrity Requirements

Usability in software engineering refers to how well a piece of software may be used by a specific target audience to accomplish goals. A user-friendly environment will be provided by the system.

4.6.1 Ease of Use Requirements

Our system will be easier to use by any type of stakeholder and they don't need any training to use the system.

4.6.2 Accessibility Requirements

The system provides authorization / authentication to get access to it. Numerous modules are used in this system.

SR-1	Safeguards are provided by the system.		
Description	The system is designed in a way that allows all modules to access a mechanism that provides security services.		
Stakeholders	Developer	**Priority**	High

4.7 Look and Feel Requirements

Look and feel requirements mainly refer to how the system will appear. The "look" of a graphical user interface in software design refers to elements like colors, shapes, layouts, and typefaces. It also refers to the behavior of dynamic elements like buttons, boxes, and menus ("The Feel").

4.7.1 Appearance Requirements

AR-1	Text color and font		
Description	Our system has to be different and attractive from other existing apps using a better look and feel.		
Stakeholders	Developers , User	**Priority**	Medium

4.8 Style Requirements

There are no style requirements in our system.

4.9 Legal Requirements

Legal requirements often refer to an organization's terms and conditions or privacy policy. No third-party software or individual is permitted to use our data for commercial purposes, according to the terms and conditions of our application.

5. Requirement Engineering Process

Software requirements are established using requirements engineering (RE), which takes into account customer wants or requirements. Requirements elicitation, needs modeling, requirements analysis, requirements assurance & validation, and requirements management are all parts of the requirements engineering process.

5.1 Requirement Elicitation Techniques

Requirements elicitation, often known as "requirement gathering," is the process of investigating and discovering system requirements for users, clients, and other stakeholders. Contacting participants directly or conducting research, analysis, and testing are two ways to elicit requirements.

5.1.1 Interviews

- Conduct interviews with medical professionals, including neurologists, radiologists, and oncologists, to understand their diagnostic workflows, challenges in brain tumor detection, and requirements for an assistive technology.

- Interview patients who have undergone brain tumor diagnosis and treatment to gather insights into their experiences, concerns, and expectations from a mobile application like "NeuroScan."

- Explore the needs and expectations of potential users, such as caregivers or family members of patients, to ensure that the application addresses their informational and support needs.

5.1.2 Prototyping

- Create interactive prototypes or wireframes of the "NeuroScan" application to visually represent its interface, navigation flow, and key features.

- Share prototypes with stakeholders, including medical professionals, patients, and potential users, to solicit feedback on usability, clarity of information, and overall user experience.

- Incorporate feedback from prototype reviews to refine the design and functionality of the application iteratively, ensuring that it aligns with the needs and expectations of its intended users.

5.2 Requirement Validation

Requirement validation criteria make sure they are accurate and match the standard you desire from this program. Our requirements initially appeared to be good, but after reading them and attempting to implement them, we discovered that they contained gaps and ambiguities.

5.2.1 Review the Requirements

Among the techniques that produce the highest quality software now accessible, negative peer review, particularly the rigorous type known as evaluation, is exceptional. We carefully looked at documented needs, analysis models, and related disability information with a team of reviewers from various viewpoints.

5.2.2 Simulate the Requirements

We can use trading tools to simulate a suggested system in place or to add specifics to textual specifications in order to stimulate requirements. The simulation advances the concept of prototyping.

6. Use Case Diagram

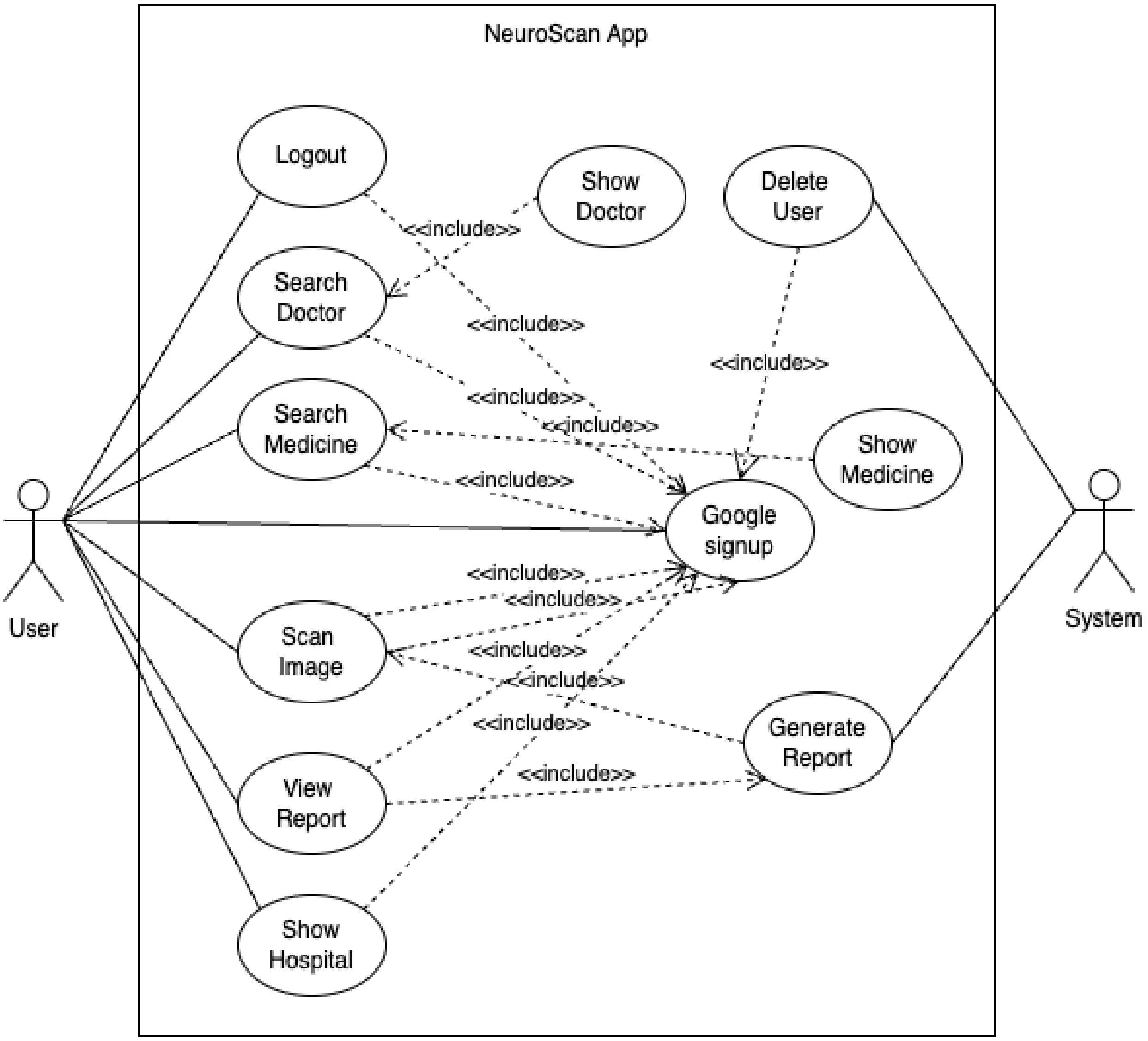

Figure 1 Use Case Diagram

7. Use Case Descriptions

Table 01 Google Signup

Use Case No.	01	
Use Case	Google signup	
Goal <a longer statement of the goal in context if needed>	A user login using their google account or add google account if user has already logged in.	
Preconditions <what we expect is already the state of the world>	User must download our application	
Success End Condition <the state of the world upon successful completion>	User logged in successfully.	
Failed End Condition <the state of the world if goal abandoned>	User doesn’t log in	
Primary Actors: **Secondary Actors:**	User System	
Trigger <the action upon the system that starts use case>	The user clicks the “Google signup” button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The user enters the welcome page of the NeuroScan App.
	2	The user clicks the “Google signup” button.
	3	The user selects this/her account from the pop-up option
	4	The system adds the account to firebase
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	3a	No account pop-up appears
	3a1	The user needs to add a google account.
	3a2	The user enters Gmail and password of his/her account
	3a3	The user’s google account added successfully
Quality Requirements	**Step**	**Requirement**
	3a2	The user should give valid information for google login

Table 02 Logout

Use Case No.	02	
Use Case	Logout	
Goal <a longer statement of the goal in context if needed>	A user logout from the App	
Preconditions <what we expect is already the state of the world>	User must Log in to the system.	
Success End Condition <the state of the world upon successful completion>	The user logout from the app successfully.	
Failed End Condition <the state of the world if goal abandoned>	The user could not log out	
Primary Actors: **Secondary Actors:**	User System	
Trigger	The user clicks the “log out” button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The user enters the NeuroScan home page after login.
	2	The user clicks the “Log out” button.
	3	The system shows a “Logout Successful” message.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	2a	Logout doesn’t work use to no internet connection.
Quality Requirements	**Step**	**Requirement**
	2	Must need Internet connection

Table 03 Search Doctor

Use Case No.	03	
Use Case	Search Doctor	
Goal <a longer statement of the goal in context if needed>	A user search for doctor by his/her name.	
Preconditions <what we expect is already the state of the world>	A user must login to the system	
Success End Condition <the state of the world upon successful completion>	The user has got the intended search request	
Failed End Condition <the state of the world if goal abandoned>	No search results are found	
Primary Actors: **Secondary Actors:**	User System	
Trigger <the action upon the system that starts use case>	The user clicks the "Search doctor" button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The user enters the NeuroScan home page after login.
	2	The user clicks to "Search Doctor" text field
	3	System redirects to doctor page with intended result
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	2a	Empty search
	2a1	System redirects to doctor page and shows nothing
	2b	Search doctor that are not store in database
	2b1	System redirects to doctor page and shows nothing
Quality Requirements	**Step**	**Requirement**
	2	Searching doctor should be fast and accurate.

Table 04 Search Medicine

Use Case No.	04
Use Case	Search Medicine

Goal <a longer statement of the goal in context if needed>	A user can search for medicine by its name.	
Preconditions <what we expect is already the state of the world>	A user must login to the system	
Success End Condition <the state of the world upon successful completion>	The user has got the intended search request	
Failed End Condition <the state of the world if goal abandoned>	No search results are found	
Primary Actors: **Secondary Actors:**	User System	
Trigger <the action upon the system that starts use case>	The user clicks the “Search medicine” button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The user enters the NeuroScan home page after login.
	2	The user clicks the “Pharmacy” button.
	3	The system redirect to pharmacy page where shows a list of available medicine.
	4	The user searched for the book from the search option.
		a. The user can Search “by Medicine Name” option. b. The user can Search “Search by Medicine Category” option.
	5	The desired book is found.
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	4a	Search by wrong Drug name and category.
	5	The desired medicine doesn’t find.
Quality Requirements	**Step**	**Requirement**
	4	Must need internet connection.

Table 05 Scan Image

Use Case No.	05
Use Case	Scan image
Goal <a longer statement of the goal in context if needed>	A user can scan the image and system generate the report
Preconditions	A user must login to the system.

<what we expect is already the state of the world>		
Success End Condition <the state of the world upon successful completion>	System will able to scan the brain tumour from image.	
Failed End Condition <the state of the world if goal abandoned>	System fails to scan the brain tumour.	
Primary Actors: **Secondary Actors:**	User System	
Trigger <the action upon the system that starts use case>	The user clicks the "Scan Image" button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The user enters the NeuroScan home page after login.
	2	The user clicks the "Scan Image" button.
	3	The system allows user to give image to the system
		a. User can take image of the MRI Brain Tumour Image
		b. User can select photo of the MRI Brain Tumour Image from gallery
	4	System scans the image.
	5	Use Case "Generate Report"
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	3a	User take a blur image of the MRI Brain Tumour Image
	3a1	The system fails to show the desired output
	3b	User select a Brain Tumour image of other format like CT scan, PET, X-rays
	3b1	The system fails to generate the report
Quality Requirements	**Step**	**Requirement**
	3	The user must give a clear MRI Brain Tumour Image to the system
	4	To scan the image must need internet connection

Table 06 Show Hospital

Use Case No.	06
Use Case	Show Hospital
Goal <a longer statement of the goal in context if needed>	User can view the list of the hospitals on the app
Preconditions <what we expect is already the state of the world>	A user must login to the system.
Success End Condition	Users view all the hospitals

<the state of the world upon successful completion>		
Failed End Condition <the state of the world if goal abandoned>	User doesn't view the hospitals	
Primary Actors: **Secondary Actors:**	User System	
Trigger <the action upon the system that starts use case>	The user clicks the "Hospitals" button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The user enters the NeuroScan home page after login.
	3	The user clicks the "Hospitals" button.
	4	The system shows the hospitals that are listed in the app
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
		N/A
Quality Requirements	**Step**	**Requirement**
		N/A

Table 7 Show Doctor

Use Case No.	07
Use Case	Show Doctor
Goal <a longer statement of the goal in context if needed>	A user sees detail information of doctor
Preconditions <what we expect is already the state of the world>	A user must login to the system and Click on the "Top doctor" button
Success End Condition	Use will see the details of the doctor

<the state of the world upon successful completion>		
Failed End Condition <the state of the world if goal abandoned>	Use will not see the details of the doctor	
Primary Actors: **Secondary Actors:**	User System	
Trigger <the action upon the system that starts use case>	The user clicks on any doctor item from the Top doctor page.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The user enters the first page of the NeuroScan App.
	2	The user clicks “Top Doctors” button
	3	System redirects to the Top Doctor page
	4	User clicks on any doctor item.
	5	System redirects to the doctor details page
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
		N/A
Quality Requirements	**Step**	**Requirement**
		N/A

Table 8 Delete Account

Use Case No.	08
Use Case	Delete Account
Goal <a longer statement of the goal in context if needed>	System can delete user account
Preconditions <what we expect is already the state of the world>	A user must google signup with their google account.
Success End Condition <the state of the world upon successful completion>	System deletes the account successfully

Failed End Condition <the state of the world if goal abandoned>	System doesn’t delete user’s account	
Primary Actors: **Secondary Actors:**	System User	
Trigger <the action upon the system that starts use case>	User’s delete account request comes in.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	User must sign up their account.
	2	Need to go profile page
	3	User request to delete the account
	4	System successfully deletes user account
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
		N/A
Quality Requirements	**Step**	**Requirement**
		N/A

Table 9 Show Medicine

Use Case No.	09
Use Case	Show Medicine
Goal <a longer statement of the goal in context if needed>	User will able to see medicine details.
Preconditions <what we expect is already the state of the world>	User must login to the system and goes to pharmacy page
Success End Condition <the state of the world upon successful completion>	User will see the medicine details successfully

Failed End Condition <the state of the world if goal abandoned>	User doesn’t see the medicine details	
Primary Actors: **Secondary Actors:**	User System	
Trigger <the action upon the system that starts use case>	The user clicks each of the “medicine item”	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The user enters the pharmacy page of the NeuroScan App.
	2	The user clicks each of the “medicine item”
	3	The user will see medicine details successfully
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	3a	User will not see medicine details
Quality Requirements	**Step**	**Requirement**
		N/A

Table 10 Generate Report

Use Case No.	10
Use Case	Generate Report
Goal <a longer statement of the goal in context if needed>	System generates the report of brain tumour image
Preconditions <what we expect is already the state of the world>	User must give MRI Brain Tumour image to the system
Success End Condition <the state of the world upon successful completion>	System will generate report successfully
Failed End Condition	System will fail generate the report

<the state of the world if goal abandoned>		
Primary Actors: **Secondary Actors:**	System User	
Trigger <the action upon the system that starts use case>	The user clicks the "Scan image" button.	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	User case "Scan Image"
	2	System will generate the report
	3	User can see the report successfully
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
	3a	User doesn't see the report without internet connection
Quality Requirements	**Step**	**Requirement**
	2	It takes up to 20 seconds to generate the report

Table 11 View Report

Use Case No.	11
Use Case	View Report
Goal <a longer statement of the goal in context if needed>	A user will be able to view the report
Preconditions <what we expect is already the state of the world>	User must give image to the system and system generate the image
Success End Condition <the state of the world upon successful completion>	User will see the report
Failed End Condition	User will not see the report

<the state of the world if goal abandoned>		
Primary Actors:	User	
Secondary Actors:	System	
Trigger <the action upon the system that starts use case>	The user clicks on “reports” navigation button	
Main Success Flows <the steps of the scenario from trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	The user enters the home page of the NeuroScan App.
	2	The user clicks on “reports” navigation button
	3	System gives the list of reports
	4	The user successfully sees the report
Alternative Flows <a: condition causing branching> <a1: action or name of sub-use case>	**Step**	**Branching Action**
		N/A
Quality Requirements	**Step**	**Requirement**
		N/A

8. Activity Diagram

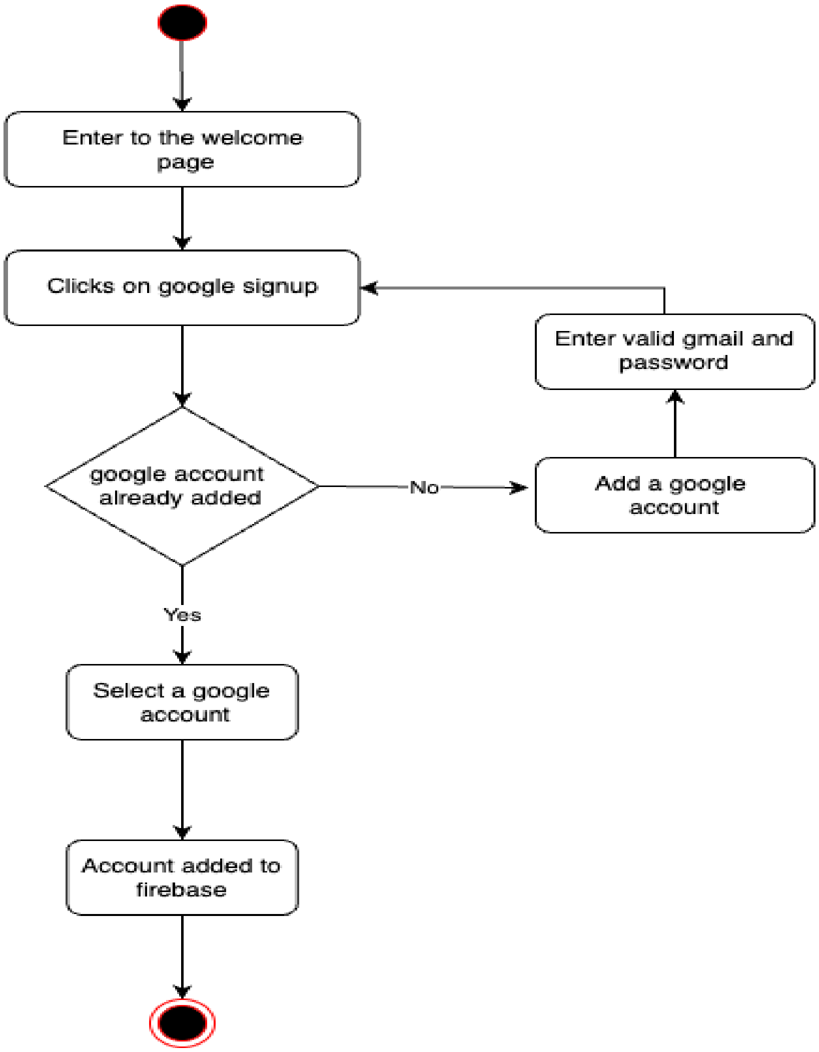
Enter to the welcome page
Clicks on google signup
Enter valid gmail and password
google account already added
No
Add a google account
Yes
Select a google account
Account added to firebase

Figure 2 Google Signup

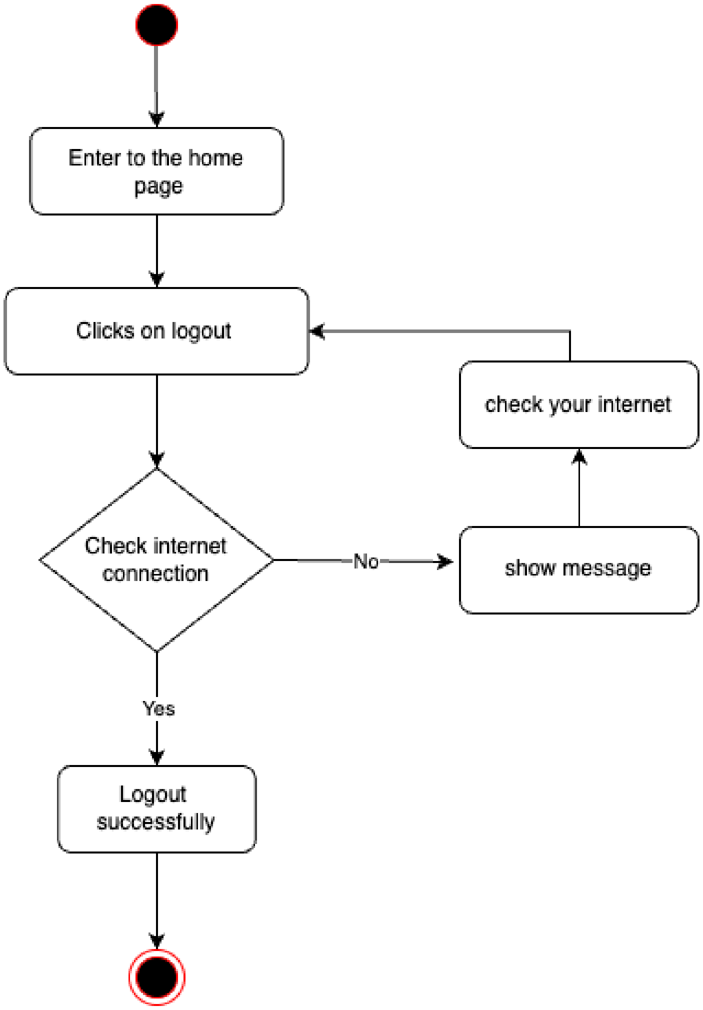

Figure 3 Logout

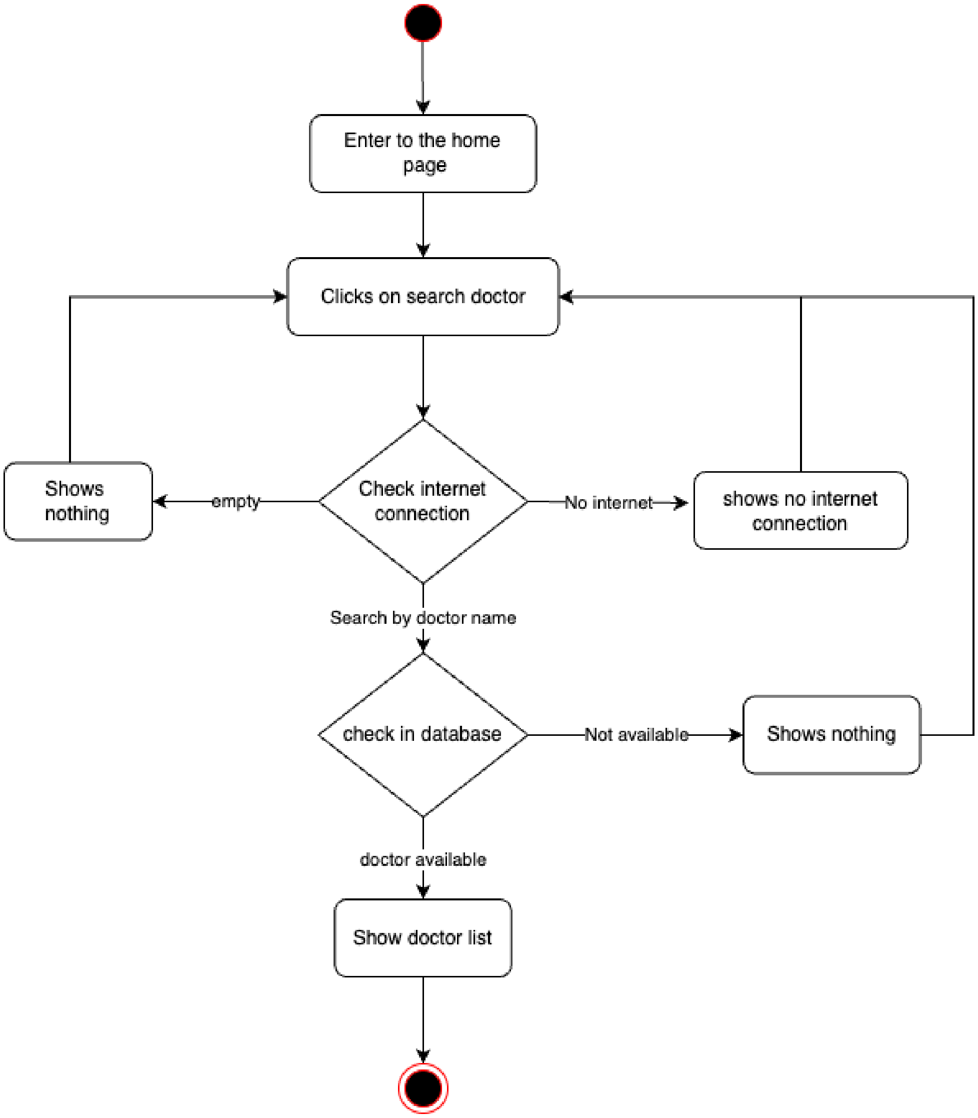

Figure 4 Search Doctor

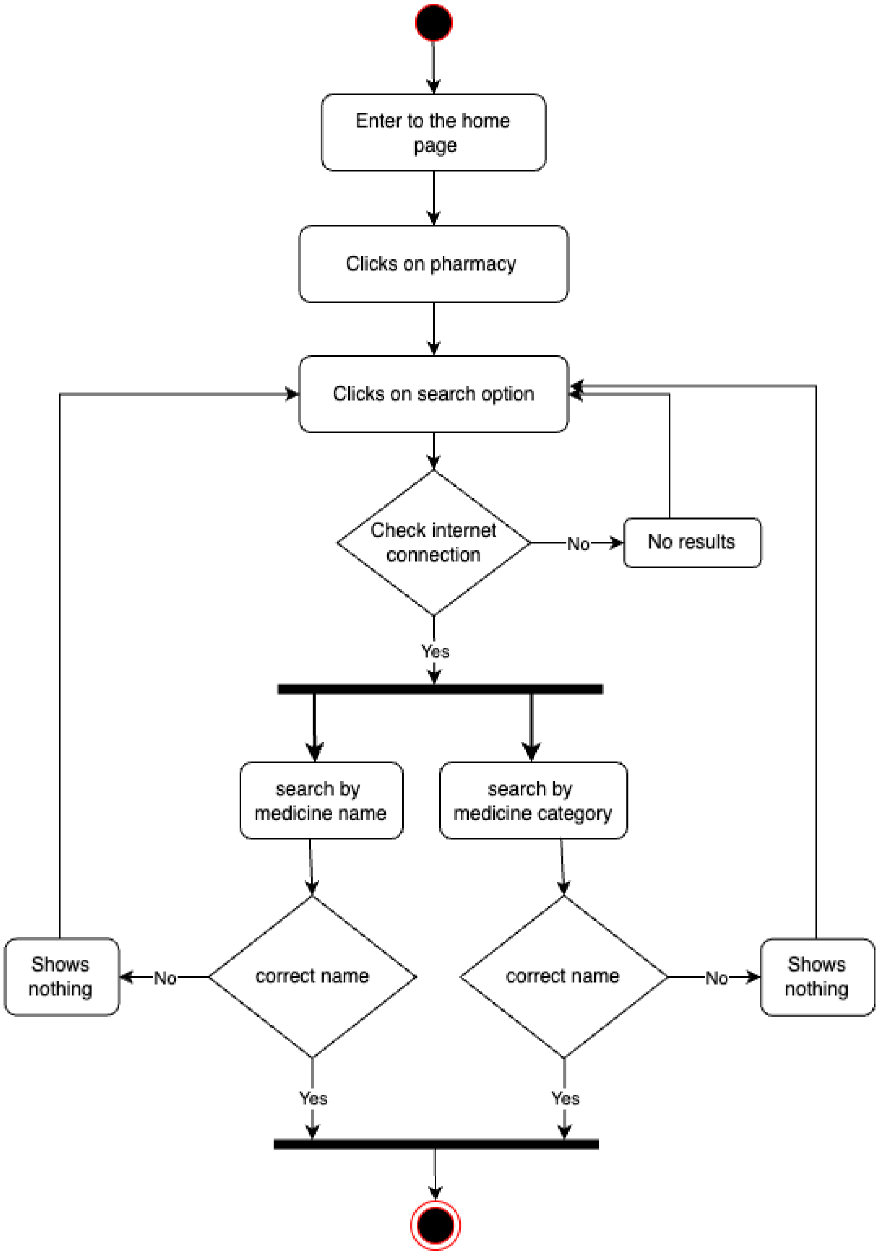

Figure 5 Search Medicine

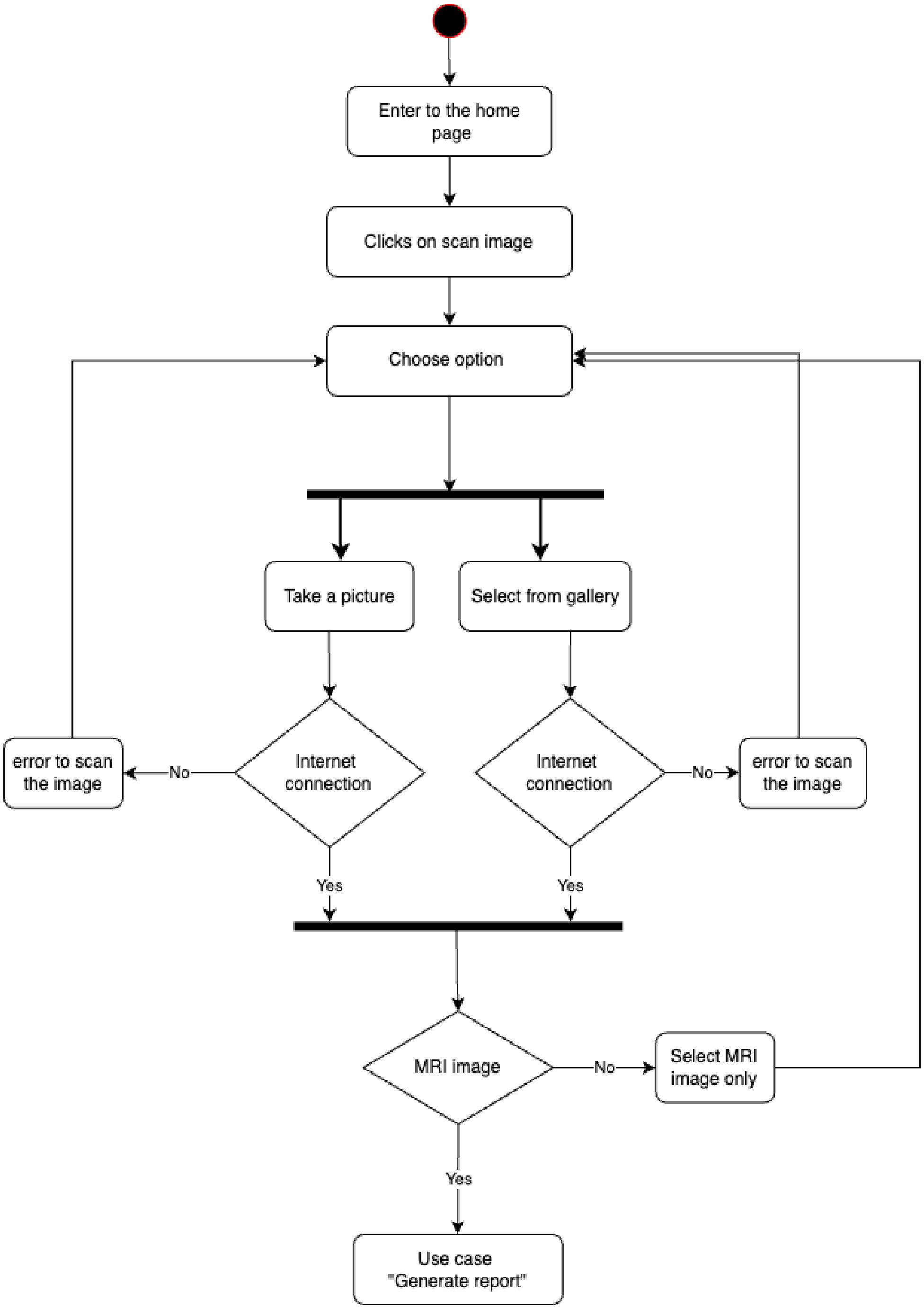

Figure 6 Scan Image

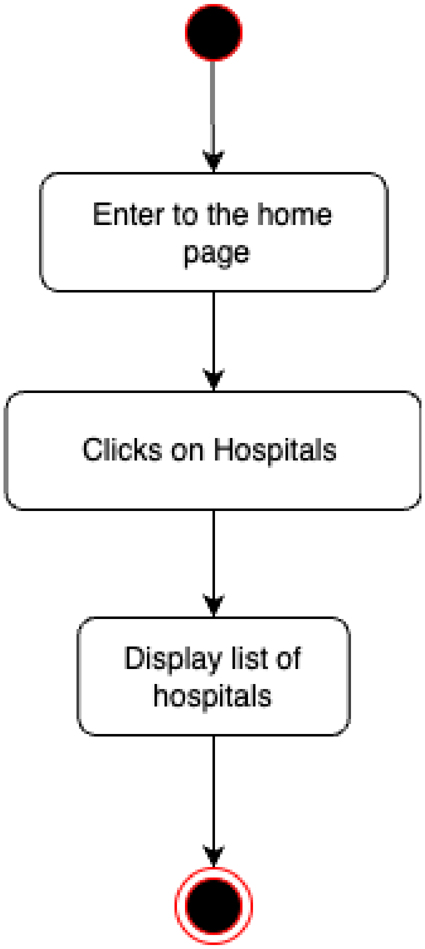

Figure 7 Show Hospital

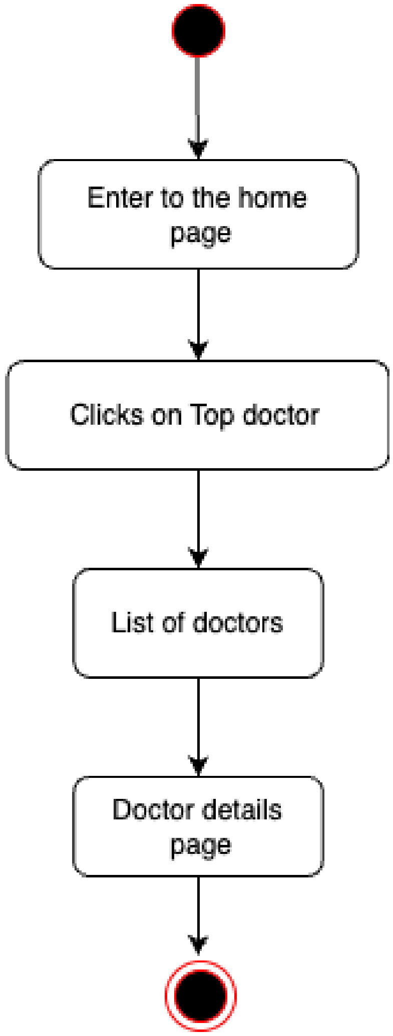

Figure 8 Show Doctor

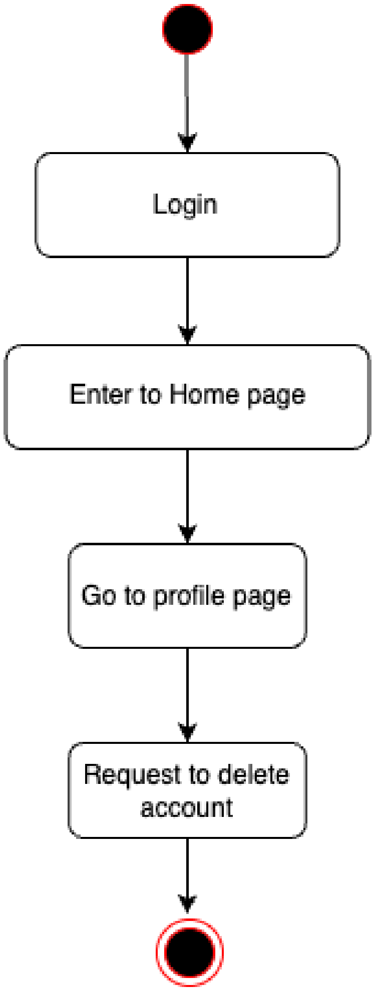

Figure 9 Delete Account

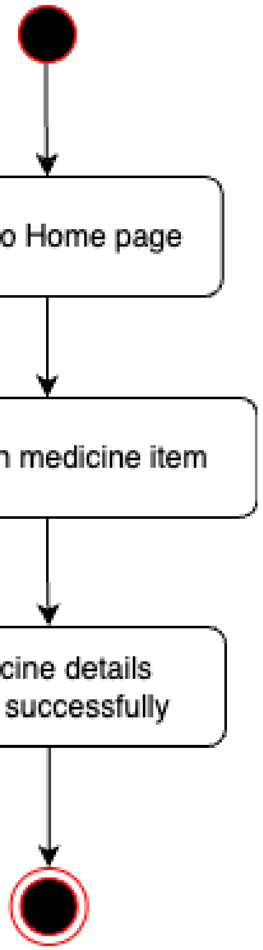

Figure 10 Show Medicine

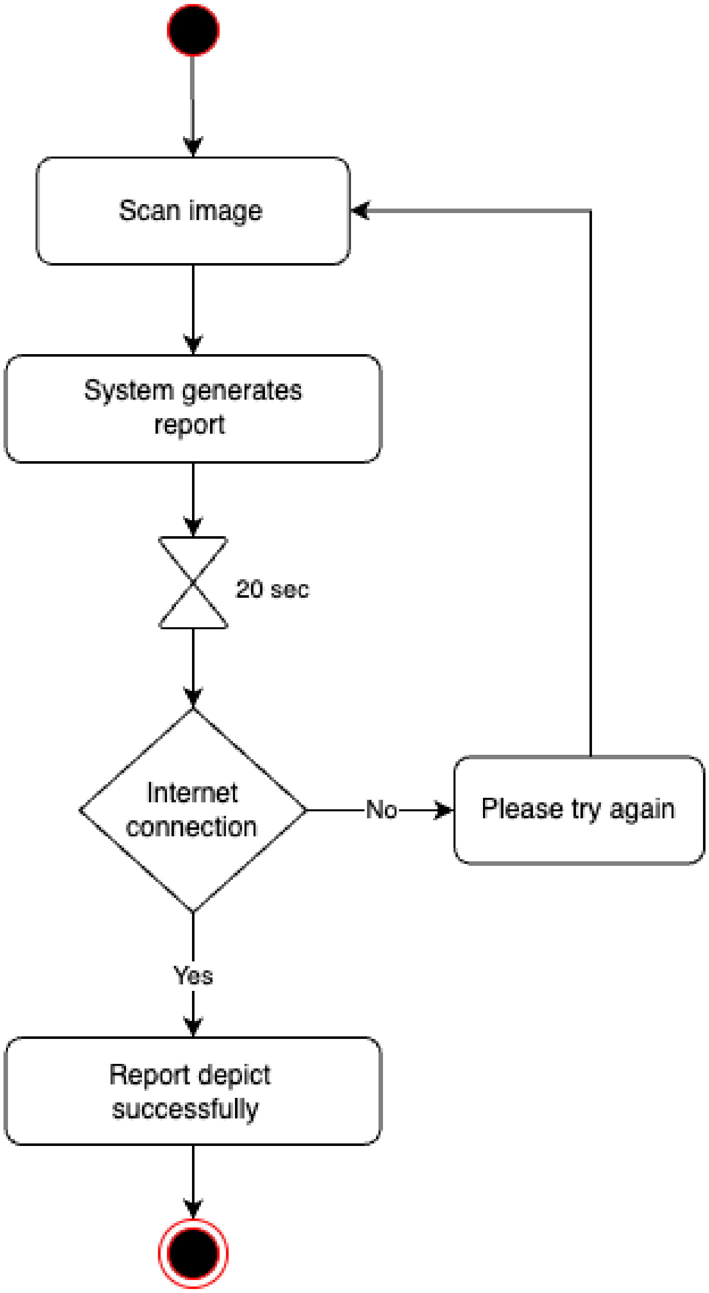

Figure 11 Generate Report

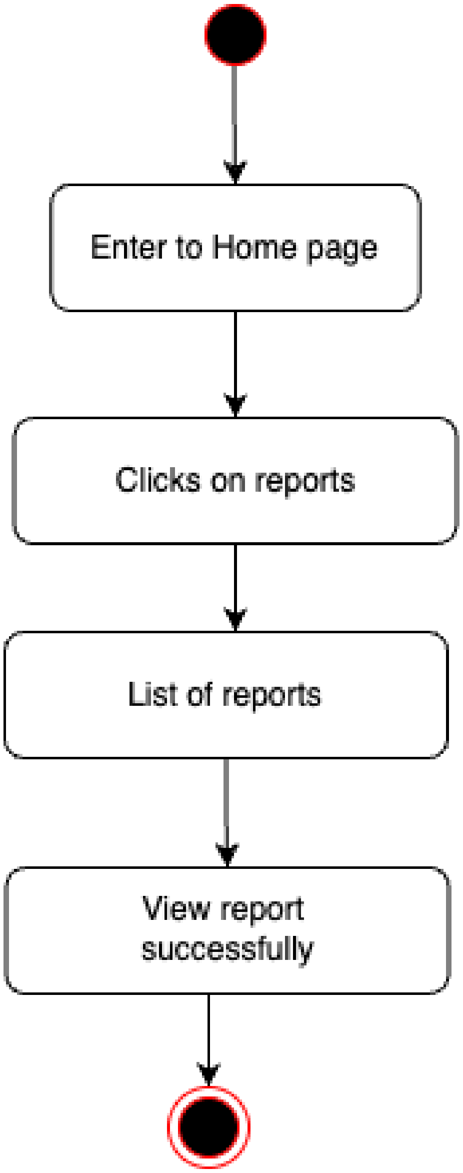

Figure 12 View Report