Software Requirements Specification

for

Sign Language Recognition

Noakhali Science and Technology University (NSTU)

Software Requirements Specification

for

Sign Language Recognition

Team Members :

Shoriful Habib MUH2025003M Year-3 Term-1 shoriful2515@student.nstu.edu.bd	Sanjida Akter Samanta BFH2025010F Year-3 Term-1 samanta2515@student.nstu.edu.bd
Sumaiya Begum BKH2025015F Year-3 Term-1 sumaiyabegum2515@student.nstu.edu.bd	Mehedi Hasan MUH205032M Year-3 Term-1 mehedi2515@student.nstu.edu.bd

Supervised By:

Dipanita Saha
Assistant Professor

Institute of Information Technology

Noakhali Science and Technology university

dipanita.iit@nstu.edu.bd

Submission Date: 11.06.2023

Table of Contents

1. Introduction 1

1.1 Problem Statement 1

1.2 Purpose 1

1.3 Project Scope 2

1.4 Glossary 2

1.5 Overview 3

2. Stakeholders and Characteristics 3

2.1 Deaf and Hard of Hearing Peoples 3

2.2 Non-Sign Language Users 3

2.3 Student 3

2.4 Educators 4

2.5 Speech and Hearing Professional 4

3. Design and Implementation Constrains 4

3.1 User Interface Technology 4

3.1.1 Programming Language 4

3.1.2 CSS Framework 5

3.3 Server-Side Technology 5

3.3.2 PHP 5

3.3.3 Database Server 6

3.4 Hardware Interfaces 6

3.5 Software Interfaces 6

3.5.1 OpenCV 6

3.5.2 TensorFlow 6

4 Requirement Specification 6

4.1 Functional Requirement 6

4.1.1 User login and registration 7

4.1.2 Valid Gesture 7

4.1.3 Display Text 7

4.1.4 Sign Language Translation 8

4.1.5 Gesture Database Management 8

4.1.6 Text Input Field 8

4.1.7 Display Sign 9
4.1.8 Convert Text to Sign Language 9
4.1.9 Convert Text to Speech 9
4.1.10 Edit Profile 10
4.1.11 Logout 10
4.2 Non-Functional Requirements 10
4.2.1 Performance 10
4.2.2 Scalability 11
4.2.3 Security 11
4.2.4 Usability 11
4.2.5 Accessibility 12
4.2.6 Compatibility 12
4.2.7 Maintainability 12
4.2.8 Testability 13
4.3 Usability Requirements 13
4.3.1 Simple and Intuitive navigation 13
4.3.2 Clear and Consistent Visual Design 13
4.3.3 Accessible for All 14
4.3.4 Help and Documentation 14
4.4 Security Requirements 14
4.4.1 Secure Transmission 14
4.4.2 Input Validation 15
4.4.3 Regular Software Updates 15
4.4.4 Vulnerability Scans 15
4.5 Performance Requirements 16
4.5.1 Responsiveness 16
4.5.2 Server Response Time 16
4.5.3 Concurrent Time 16
4.5.4 Optimization 17
4.6 Style Requirements 17
4.7 Legal Requirements 17
5 Requirement Engineering Process 17
5.1 Requirement Elicitation Techniques 17

5.1.1 Hold Interviews 18
5.1.2 Perform Document Analysis 18
5.1.3 System Interface Analysis 18
5.1.4 Distribute Questionnaires 18
5.2 Requirement Validation 18
5.2.1 Review the Requirements 19
5.2.2 Test the Requirements 19
5.2.3 Simulate the requirements 19
6 Use Case Diagram 20
7 Use Case Description 21
8 Activity Diagram 37
8.1 Registration 37
8.2 Log in 38
8.3 Forgot Password 39
8.4 Change Password 40
8.5 Edit Profile 41
8.6 Start Webcam 42
8.7 Enter Plain Text 43
8.8 Convert Text to Speech 44
8.9 Translate Gesture 45
8.10 Show Gesture 46
8.11 Match Feature 47
8.12 Capture Feature Point 48
8.13 Capture Video 49
8.14 Capture Gesture 50
8.15 Display Result 51
9 Appendix 52
9.1 Prioritization of requirements 52
9.1.1 Three-level Scale 52
9.1.2 Prioritization of the requirements of Sign Language Recognition 52
9.2 Traceability Matrix 52
10 References 54
11 Database Design 54

11.1 Schema of the database.....54
11.2 Schema Diagram.....55

List of Tables

Table 1: Registration 21
Table 2: Login 22
Table 3: Activation Account 23
Table 4: Forgot Password 24
Table 5: Change Password 25
Table 6: Edit Profile 26
Table 7: Start Webcam 27
Table 8: Enter Plain Text 28
Table 9: Convert Text to Speech 29
Table 10: Translate Gesture 30
Table 11: Show Gesture 31
Table 12: Match Feature 32
Table 13: Capture Feature Point 33
Table 14: Capture Video 34
Table 15: Capture Gesture 35
Table 16: Display Result 36
Table 17 :Traceability Matrix 54

List of Figures

Figure 1: Use Case Diagram 20
Figure 2: Registration 37
Figure 3: Log in 38
Figure 4: Forgot Password 39
Figure 5: Change Password 40
Figure 6: Edit Profile 41
Figure 7: Start Webcam 42
Figure 8: Enter Plain Text 43
Figure 9: Convert Text to Speech 44
Figure 10: Translate Gesture 45
Figure 11: Show Gesture 46
Figure 12: Match Feature 47
Figure 13: Capture Feature Point 48
Figure 14: Capture Video 49
Figure 15: Capture Gesture 50
Figure 16: Display Result 51
Figure 17: Schema Diagram 55

1. Introduction

Verbal communication performed by humans is one of the most unique traits in the entire animal kingdom. Humans have used communication as a tool to share and expand our knowledge of the world. It is safe to say that humans have created settlements, societies, technologies, strategies and more, only through efficient communication. In today's world, communication between individuals is essential to the development and maintenance of society. But unfortunately, some individuals with hearing/speech disabilities are unable to perform this basic human interaction. This barrier in communication alienates them from society and hinders effective communication. Since it is not feasible to assume that every person who communicates with such disabled individuals knows sign language, we need a method that will eradicate this communication barrier. In this project, we are proposing one such method.

1.1 Problem Statement

There are several potential problems that a sign language detection may aim to solve:

Difficult to communicate: Meena is a student who is unable to talk. She uses sign language to talk or communicate with others. But she feels difficulty when she tries to communicate with someone. One day Meena's friend Rabeya came to Meena's house. Rabeya wants to communicate with Meena. A sign language detection system what can convert sign language images to text or text to sign language image format can help them so that they can communicate easily.

Time-consuming manual processes: Traditional methods of communicating, sometimes not useful because of the lack of understanding what they are saying. A Sign language detection can automate many of these processes, saving time and reducing the potential for mistakes.

1.2 Purpose

The purpose of this document is to specify the features, requirements of the final product and the interface of Sign Language detection. It will explain the scenario of the desired project and necessary steps to succeed in the task. To do this throughout the document, an overall description of the project, the definition of the problem that this project presents a solution and definitions and abbreviations that are relevant to the project will be provided. The preparation of this SRS will help consider all the requirements before design begins, and reduce later redesign, recording, and retesting. If there is any change in the functional requirements or design constraint's part, these changes will be stated by giving reference to this SRS in the following documents.

1.3 Project Scope

This system is primarily intended for making an Interpreter. This will have applications in Business who want to employ deaf and mute employees can use it to convey employee messages to the end consumer. It will be used majorly by the deaf and mute to communicate. The applications can further be extended to security purposes, by developing sign language of your own. And even observing and analyzing any suspicious actions. Some other applications and scopes of this project are:

- It can be used to provide live captions for online meetings.
- It can be used to detect mistakes in sign language.
- It can be used for learning and practicing sign language.
- Text generated from this application can be converted to speech for better.
- communication.
- Use hand gestures to control and automate other devices.

1.4 Glossary

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

GUI	Graphical User Interface
API	Application Programming Interface
SRS	Software Requirement Specification
UI	User Interface
SDLC	Software Development Life Cycle
MB	Megabytes
XML	Extensible Markup Language
RESTful	Representational State Transfer
HTML	Hyper Text Markup Language

1.5 Overview

The Sign Language Detection System can be used by both deaf or hard-of-hearing individuals and those who do not understand sign language but need to communicate with them. The user opens the Sign Language Detection System on their computer, accessing the dedicated website. Then the user can choose one option from two buttons. One is the sign-to-text button and the other is the text-to-sign button. When a user selects the sign-to-text button then there is another window open with a button start-webcam. When the user selects the button start-webcam the system asks permission to allow the accessing camera of the user's computer. When the camera is open the user can give a gesture and the system captures it and recognizes the gesture and give an output in the form of text. If the user wants to convert the text into speech, then there is a button for text-to-speech. After clicking on the button, the system provides the speech of the following text. The user can convert text into a sign by using the button text-to-sign. After clicking on the button there is open a new window where a user can enter a valid text and the system will provide the gesture of the following text.

2. Stakeholders and Characteristics

2.1 Deaf and Hard of Hearing Peoples

Deaf and hard of hearing peoples are the primary stakeholders in sign language detection. They are the users who rely on sign language as their primary mode of communication. Their characteristics may include diverse signing styles, variations in regional signs, and different levels of sign language proficiency.

2.2 Non-Sign Language Users

The individuals who really carry out the task of educating kids are teachers. In their classrooms, they have a significant degree of power over what and how pupils are taught. Teachers believe they have the competence to know what to teach and how it should be taught to their pupils, which is another reason they feel strongly about education.

2.3 Student

Students, as stakeholders in a sign language detection system, can be characterized as they include both deaf or hard-of-hearing students who rely on sign language for communication and students studying sign language or training to become interpreters. They directly benefit from accurate sign language detection for their education, communication, and career goals. Their input and feedback are essential for system improvement and ensuring accessibility in educational settings.

2.4 Educators

Teachers and educators of sign language rely on accurate sign language detection for instructional purposes. They may use technology to support sign language learning, develop educational materials, or create accessible resources for students.

2.5 Speech and Hearing Professional

Speech and Hearing Professionals, as stakeholders in the field of speech and hearing may utilize sign language detection to support communication and therapy sessions with individuals who use sign language. Professionals working with individuals who are transitioning from spoken language to sign language may require detection systems that facilitate accurate translation.

3. Design and Implementation Constrains

We have employed design and implementation constraints to ensure the success of this project. It also refers to a tool that allows developers and testers to inspect and interact with the application's user interface (UI) elements.

3.1 User Interface Technology

The visual layout of the components that a user could interact with on a website or technical product is referred to as user interface design, or UI design. In other terms, it is a website's visual design.

3.1.1 Programming Language

JavaScript: JavaScript is an ECMAScript-compliant high-level, frequently just-in-time compiled language. It has first-class functions, dynamic typing, and prototype-based object orientation. It's multi-paradigm, allowing you to program in event-driven, functional, or imperative styles.

JavaScript XML is abbreviated as JSX. It's just a JavaScript syntactic extension. It allows us to create HTML directly in React (within JavaScript code). It is straightforward to generate a template in React using JSX, but it is not a simple template language; instead, it has all of JavaScript's capability. It is faster than standard JavaScript because it optimizes when converting to standard JavaScript. Rather than dividing the markup and functionality in different files.

Python: Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a

scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and can be freely distributed.

3.1.2 CSS Framework

Cascading Style Sheets (CSS) is a language for specifying the appearance of a document written in a markup language like HTML. Along with HTML and JavaScript, CSS is a key component of the World Wide Web.

Tailwind: Tailwind CSS is a popular utility-first CSS framework that provides a set of pre-defined CSS classes to streamline and speed up front-end development. It takes a different approach compared to traditional CSS frameworks like Bootstrap or Foundation.

3.3 Server-Side Technology

Server-side development refers to the actions that take place behind the scenes when an application is used. It primarily focuses on databases, scripting, website architecture, backend logic, APIs, and Servers.

3.3.2 PHP

The general-purpose programming language (GPPL) is PHP. It is mostly used as a server-side scripting language for the creation of websites. Web development is also simplified by the PHP frameworks. This framework makes it easier to reuse existing code and eliminates the need to create lengthy, intricate code for web apps. The majority of PHP frameworks are free source and simple to use. Because PHP is open-source and cost-free, developers may install it easily and utilize it right away. All major operating systems, including Windows, Unix, Linux, etc., support PHP. Web applications created using PHP may simply operate on any platform. PHP makes a safe connection with databases and connects to them with ease. It features an integrated module that may be used to quickly connect to the database. The primary purposes of the PHP framework are to simplify the construction of web applications and to automatically maintain the code. The built-in tools and features of PHP frameworks make it simpler to defend online applications from outside assaults and security risks.

3.3.3 Database Server

A piece of hardware that runs database software is called a database server. Users and companies may store, manage, retrieve, update, or modify files, information logs, and other types of digital data with the use of database software. Large volumes of digital information can be easily stored, arranged, and maintained using database servers. Database servers perform by combining a database management system with memory and storage capacity for databases (DBMS).

3.4 Hardware Interfaces

If there is no embedded camera in the system, then there will be the need of an external camera sensor along with the driver needed to enable the functionality on that specific operating system and the hardware platform.

3.5 Software Interfaces

3.5.1 OpenCV

OpenCV is used to track the gestures from the input stream.

3.5.2 TensorFlow

TensorFlow is an open-source software library for high performance numerical computation. Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices.

4 Requirement Specification

All the requirements based on the elicitation process are described in this section.

4.1 Functional Requirement

Functional requirements are those requirements that are used to illustrate the internal working nature of the system, the description of the system, and explanation of each subsystem. It consists of what task the system should perform, the processes involved, which data the system should hold and the interfaces with the user.

4.1.1 User login and registration

FR-1	User should be able to register to the system and login to a registered account.		
Description	User should register his/her account for the first time and be able to log in to the account which was registered once. Already registered users will not face registration stage.		
Stakeholders	User	**Priority**	High

4.1.2 Valid Gesture

FR-2	Gestures must be Valid.		
Description	The gestures shown by the user must be valid. The user cannot give any random gesture if valid output is expected.		
Stakeholders	User	**Priority**	High

4.1.3 Display Text

FR-3	Display the recognized text to the user as the output of the sign language conversion.		
Description	According to the gestures shown by the user, the system will translate the sign to text. The translated text must be displayed to the user as output.		
Stakeholders	User	**Priority**	High

4.1.4 Sign Language Translation

FR-4	The system should accurately and efficiently translate sign language to plain text.		
Description	The system should translate recognized sign language gestures into written or spoken language to facilitate communication with non-signing individuals. The translation is crucial for bridging the communication gap between sign language users and those who do not understand sign language.		
Stakeholders	User and Developers.	**Priority**	High

4.1.5 Gesture Database Management

FR-5	There should be a database for gesture management.		
Description	The system should provide a mechanism to manage a database of recognized gestures. While not as critical as other functionalities, the ability to manage the gesture database allows users to add, edit, or delete gestures as needed for customization and maintenance.		
Stakeholders	User and Developers.	**Priority**	Medium

4.1.6 Text Input Field

FR-6	System should provide a text input field where users can type the desired text.		
Description	The system should be able to translate text to sign language. For translating the text there should be a text input field where users can type the desired text.		
Stakeholders	User and Developers.	**Priority**	High

4.1.7 Display Sign

FR-7	Display the recognized sign to the user as the output of the text to sign conversion.		
Description	According to the text entered by user, the system will translate the text to sign. The translated sign must be displayed to the user as output.		
Stakeholders	User	**Priority**	High

4.1.8 Convert Text to Sign Language

FR-8	The system should be able convert text to sign language.		
Description	The system shall accept text input from the user, representing the desired message or phrase to be translated into sign language. The system should analyze the text to determine the appropriate sign language representation.		
Stakeholders	User and Developers.	**Priority**	High

4.1.9 Convert Text to Speech

FR-9	The system should be able to convert text to speech.		
Description	The system processes conversion from sign language to natural language, English in text. The system shall generate natural and intelligible speech based on the processed text.		
Stakeholders	User and Developers.	**Priority**	Medium

4.1.10 Edit Profile

FR-10	User should be able to edit their profile.		
Description	After registration the user who wants to update their profile should be able to update their profile.		
Stakeholders	User	**Priority**	Medium

4.1.11 Logout

FR-11	Users log out from their account.		
Description	The user will be able to log out of his/her account at the end of his need. Users will need to log in again for later use.		
Stakeholders	User	**Priority**	Medium

4.2 Non-Functional Requirements

4.2.1 Performance

NFR-1	Performance		
Description	The application should load quickly and be responsive to its user interactions.		
Stakeholders	User	**Priority**	High

4.2.2 Scalability

NFR-2	Scalability		
Description	The application should be able to handle a high number of concurrent users.		
Stakeholders	User	**Priority**	High

4.2.3 Security

NFR-3	Security		
Description	The application should protect user data and prevent unauthorized access.		
Stakeholders	User	**Priority**	High

4.2.4 Usability

NFR-4	Usability		
Description	The application should be easy to navigate and use for all users, including those with disabilities.		
Stakeholders	User	**Priority**	High

4.2.5 Accessibility

NFR-5	Accessibility		
Description	The application should meet accessibility standards and guidelines, to make it available to as many people as possible.		
Stakeholders	User	**Priority**	High

4.2.6 Compatibility

NFR-6	Compatibility		
Description	The application should work on different devices and browsers		
Stakeholders	User	**Priority**	High

4.2.7 Maintainability

NFR-7	Maintainability		
Description	The code should be well-organized and easy to maintain.		
Stakeholders	Development Team	**Priority**	High

4.2.8 Testability

NFR-8	Testability		
Description	The application should have automated tests in place to ensure the app is stable and free of errors.		
Stakeholders	Development Team	**Priority**	High

4.3 Usability Requirements

4.3.1 Simple and Intuitive navigation

UR-1	Simple and Intuitive navigation		
Description	The application should have a simple, consistent and easy-to-use navigation structure.		
Stakeholders	User	**Priority**	High

4.3.2 Clear and Consistent Visual Design

UR-2	Clear and consistent visual design		
Description	The application should have a clear and consistent visual design, which helps users understand the app's structure and organization.		
Stakeholders	User	**Priority**	High

\

4.3.3 Accessible for All

UR-3	Accessible for all		
Description	The application should be designed to meet accessibility guidelines and standards, to make it available to as many people as possible.		
Stakeholders	User	**Priority**	High

4.3.4 Help and Documentation

UR-4	Help and documentation		
Description	The application should provide users with clear and easy-to-understand help and documentation.		
Stakeholders	User	**Priority**	High

4.4 Security Requirements

4.4.1 Secure Transmission

SR-1	Secure Transmission		
Description	Data transmission between the user's device and the server should be secured using HTTPS.		
Stakeholders	User	**Priority**	High

4.4.2 Input Validation

SR-2	Input Validation		
Description	The application should validate all input data and prevent malicious attacks such as SQL injection, Cross-Site scripting (XSS).		
Stakeholders	User	**Priority**	High

4.4.3 Regular Software Updates

SR-3	Regular Software updates		
Description	The application should be regularly updated with the latest security patches and fixes.		
Stakeholders	Development Team	**Priority**	High

4.4.4 Vulnerability Scans

SR-4	Vulnerability scans		
Description	The application should be regularly scanned for vulnerabilities and any issues found should be addressed promptly.		
Stakeholders	Development Team	**Priority**	High

4.5 Performance Requirements

4.5.1 Responsiveness

PR-1	Responsiveness		
Description	The application should be responsive to user interactions and provide instant feedback for actions taken.		
Stakeholders	User	**Priority**	High

4.5.2 Server Response Time

PR-2	Server response time		
Description	The server should provide a response time under a few seconds when handling requests from the application.		
Stakeholders	User	**Priority**	High

4.5.3 Concurrent Time

PR-3	Concurrent time		
Description	The application should be able to handle a high number of concurrent users without any significant performance degradation.		
Stakeholders	User	**Priority**	High

4.5.4 Optimization

PR-4	Optimization		
Description	The application should be optimized for performance, including techniques like caching, magnification and code splitting.		
Stakeholders	Development Team	**Priority**	High

4.6 Style Requirements

There are no style requirements in our system.

4.7 Legal Requirements

Legal requirements normally refer to the terms and conditions or privacy policy of any organization. The terms and conditions of our application are that no third-party software or person is allowed to use our data for their business purposes.

5 Requirement Engineering Process

Requirements Engineering (RE) determines software requirements according to customer requirements or needs. Requirements engineering process includes requirements elicitation, needs modeling, requirements analysis, requirements assurance & validation, and requirements management.

5.1 Requirement Elicitation Techniques

Requirements elicitation is the practice of researching and finding system requirements for users, customers, and other stakeholders also referred to as "requirement gathering". Requirement elicitation can be done by contacting participants directly or by doing some research, analysis, and testing. We mainly used these techniques for gathering Requirement:

- Interview
- Questionnaire
- Paper prototype

5.1.1 Hold Interviews

We do interviews with individuals who are facing problems in communicating with dumb and deaf people and who have expertise or experience in sign language, such as sign language interpreters, researchers, or native signers. We aim for a diverse range of participants to gain a comprehensive perspective. Review and analyze the data gathered from the interviews. Look for patterns, common themes, or significant insights that emerge from participants' responses. This analysis will guide the development of your sign language detection system.

5.1.2 Perform Document Analysis

Existing documentation can help to show how systems are currently operating or what they are what I should do. Documents include written information about current programs, business processes, needs specifications, and competitor research. Review once textual analysis can help determine which performance should remain and functionality that isn't in use. After the existing document. In analysis, we found several problems with the existing system.

- Existing systems cannot perform file compression.
- A user cannot share a file with others.
- No cloud storage system is provided by the existing systems.

5.1.3 System Interface Analysis

The first thing to do is to identify which systems the system-to-be shall communicate with. It could be a server on the Internet, a piece of software on the same host as the system-to-be, some hardware, or something completely different.

5.1.4 Distribute Questionnaires

The questionnaire is a useful way to investigate styles, changes in attitudes and users' ideas, and user satisfaction with priorities and preferences. Our list of questions was as short as possible. The respondent may be tired or frustrated. Had a basic reason for all the questions as well as group the topic areas together for the respondent to focus on. The main advantage of this survey responses was that they were collected in the usual way. Information was summarized by many people.

5.2 Requirement Validation

Requirement validation ensures that the requirements are correct and reflect the quality you want from this program. In the beginning, our requirements looked good but when we read them and tried to work with them, they came out having ambiguities and gaps.

5.2.1 Review the Requirements

Negative peer review, especially the type of rigorous review called evaluation, is unique among the highest quality software processes available. We had a team of reviewers representing different perspectives and carefully examined written needs, analysis models, and related information on disability.

5.2.2 Test the Requirements

The test creates another view of the requirements. We also performed writing tests regarding assurance of whether the expected performance was found or not. Getting tested by the user needs to document the expected product behavior under specified conditions.

5.2.3 Simulate the requirements

To stimulate requirements, trading tools are available that we have used to simulate a proposed system in place or to add details of written requirements. The simulation takes prototyping to the next level.

6 Use Case Diagram

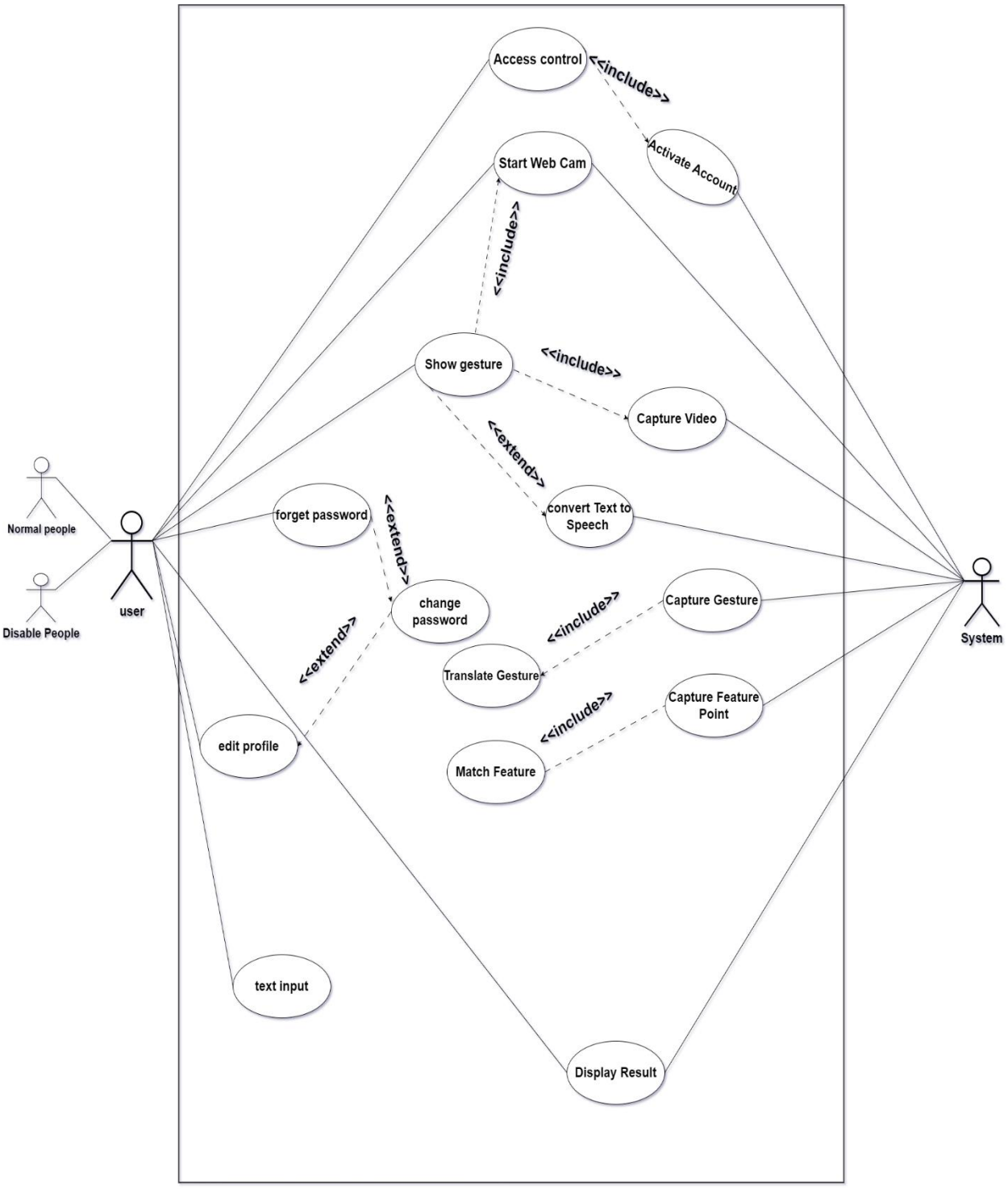

Figure 1: Use Case Diagram

7 Use Case Description

Table 1: Registration

Use Case	Registration	
Goal	User wants to create an account in the "Sign Language Detection" System.	
Preconditions	N/A	
Success End Condition	A user account is created.	
Failed End Condition	User account is not created.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	"Create Account" button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	User opens the "Sign Language Detection" application.
	2	User clicks the "Create Account" button.
	3	User provides username, email, password, repeat password.
	4	Server checks Gmail is already existed or not.
	5	User activated the account.
	6	Account is created.
Alternative Flows	**Step**	**Branching Action**
	2a	User have an account
	3a	User does not provide information.
	4a	Server shows that information invalid or used before.
	4a1	User needs to change provided information.
Quality Requirements	**Step**	**Requirement**
	4	Server will respond within 3 to 5 seconds

Table 2: Login

Use Case	Login	
Goal	User login to “Sign Language Detection” system.	
Preconditions	User Account.	
Success End Condition	Successfully login to “Sign Language Detection” system.	
Failed End Condition	Unable to login.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	“Login” Button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	User opens the “Sign Language Detection” system.
	2	User provides username and password.
	3	Server confirms the password with that Gmail.
	4	User login successful
Alternative Flows	**Step**	**Branching Action**
	2a	User does not have an account.
	2a1	User clicks the “Create Account” button to create an account.
Quality Requirements	**Step**	**Requirement**
	3	User can login whenever he/she wants (Availability). If correct credential is given, then user must be login (Integrity)

Table 3: Activation Account

Use Case	Activate Account	
Goal	Users activate account.	
Preconditions	Provide Gmail.	
Success End Condition	Successfully create account.	
Failed End Condition	Unable to create account.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	User needs to click on the provided link which was sent by the system in specified Gmail account.	
Main Success Flows	**Step**	**Action**
	1	User provides Gmail and password and clicked signup
	2	System sends an activation link to the given Gmail.
	3	Users click on that link to activate account.
	4	Account activated.
Alternative Flows	**Step**	**Branching Action**
	2a	System doesn't send activation link
	2a1	User again provide information and signup
	3a	User doesn't click on activation link
	4a	Account is not activated.
Quality Requirements	**Step**	**Requirement**
	2	Server must send an activation link to the given Gmail within 5 minutes.

Table 4: Forgot Password

Use Case	Forgot Password	
Goal	Create new password.	
Preconditions	N/A	
Success End Condition	Successfully create new password.	
Failed End Condition	Unable to create new password.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	"Forgot Password" Button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	Users click on "Forgot Password"
	2	System asks for user Gmail. User provide his/her Gmail
	3	System sends a code to that Gmail.
	4	User puts the code and click verify.
	5	System asks for new password and confirm password. Users provide new password.
	6	Password changed.
Alternative Flows	**Step**	**Branching Action**
	3a	System did not send the code
	3a1	Users again request for change password.
Quality Requirements	**Step**	**Requirement**
	3	Server must send an activation link to the given Gmail within 5 minutes.

Table 5: Change Password

Use Case	Change password	
Goal	Change user password	
Preconditions	Go to the edit profile	
Success End Condition	Change user's current password	
Failed End Condition	Unable to change password	
Primary Actors: **Secondary Actors:**	User System	
Trigger	"Change password" Button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	Users click on "Change Password"
	2	System asks current password and new password
	3	Users provide current and new password.
	4	Users click the "Save" Button.
	5	System changes the password.
Alternative Flows	**Step**	**Branching Action**
		N/A
Quality Requirements	**Step**	**Requirement**
		N/A

Table 6: Edit Profile

Use Case	Edit Profile	
Goal	Update information.	
Preconditions	N/A	
Success End Condition	Successfully update information.	
Failed End Condition	Unable to update information.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	"Update Profile" Button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	Users click the "Update Profile".
	2	Users click the "Change username", "Change Email" or "Change Password".
	3	System asks for a new username/Email/Password. Users provide new username/Email/Password.
	4	Users click the "Save" Button
	5	System updates the username/Email/Password.
Alternative Flows	**Step**	**Branching Action**
	2	Users click on change password
	3	System asks current password and new password
	3a	Users provide current and new password.
Quality Requirements	**Step**	**Requirement**
		N/A

Table 7: Start Webcam

Use Case	Start webcam	
Goal	Stakeholders wants to open webcam to capture anything to the system.	
Preconditions	• The device being used has a webcam connected or built in. • The application or platform being used supports webcam functionality.	
Success End Condition	• The webcam is actively capturing video and audio input from the device. • The user can utilize the webcam functionality within the application or platform.	
Failed End Condition	N/A	
Primary Actors: **Secondary Actors:**	User System	
Trigger	Start capture	
Main Success Flows	**Step**	**Action**
	1	Stakeholders requested to start webcam.
Alternative Flows	**Step**	**Branching Action**
		• If the conditions or triggers monitored by the external system are not met, the webcam remains inactive. • The specific conditions or triggers for activating the webcam can vary based on the requirements and design of the external system. For example, it could be based on specific events, user actions, or time-based triggers.
Quality Requirements	**Step**	**Requirement**
		Not applicable.

Table 8: Enter Plain Text

Use Case	Enter Plain Text	
Goal	To get user input as text and display the gesture image to the user.	
Preconditions	The user is logged in to the system.	
Success End Condition	Successfully show the result to user.	
Failed End Condition	N/A	
Primary Actors: **Secondary Actors:**	User System	
Trigger	"Input Text Field" needs to be clicked	
Main Success Flows	**Step**	**Action**
	1	User is presented with a user interface that allows entering text.
	2	The user enters the desired text into the provided input field.
	3	Users click the "Show" Button
	4	The system validates the entered text.
	5	The text store in the systems.
	6	The system matches the feature.
	7	Display the result to the user.
Alternative Flows	**Step**	**Branching Action**
	4a	The text is not validated.
	4a1	User will get an error message.
	4a2	User again enters the text and click "Show Button".
	6a	The feature does not match with the text.
	6a1	User will get an error message.
	6a2	User again enters the text and click "Show Button".
Quality Requirements	**Sl.**	**Requirement**
	1	System must check the input and display the result very quickly.

Table 9: Convert Text to Speech

Use Case	Convert Text to speech	
Goal	To get generated text, translate it to speech.	
Preconditions	The user is logged in to the system and the text is generated.	
Success End Condition	Play the sound.	
Failed End Condition	N/A	
Primary Actors: **Secondary Actors:**	System User	
Trigger	"Play Sound" Button needs to be clicked.	
Main Success Flows	**Step**	**Action**
	1	Get the generated text from the system.
	2	Process the text.
	3	Generate the speech.
	4	Play sound.
Alternative Flows	**Step**	**Branching Action**
		N/A
Quality Requirements	**Sl.**	**Requirement**
	1	The generated speech should be of high quality.

Table 10: Translate Gesture

Use Case	Translate Gesture	
Goal	To translate the gesture to text.	
Preconditions	The user has access to a camera or sensor device capable of capturing sign language gesture.	
Success End Condition	Successfully translate the gesture to text.	
Failed End Condition	Unable to translate the gesture.	
Primary Actors: **Secondary Actors:**	System N/A	
Trigger	The camera must start capturing the gesture.	
Main Success Flows	**Step**	**Action**
	1	Capture the gesture via camera.
	2	Preprocess the gesture.
	3	Gesture recognized by the system.
	4	Translate the gesture to text.
Alternative Flows	**Step**	**Branching Action**
	3a	Gesture is not recognized.
	3a1	Show an error message.
	3a2	Again, capture the gesture
Quality Requirements	**Sl.**	**Requirement**
	1	System should recognize gesture accurately and quickly.
	2	After recognition system should take maximum 3 second to translate the gesture.

Table 11: Show Gesture

Use Case	Show Gesture	
Goal	To show the gesture by user.	
Preconditions	The user is logged in to the system and the user has access to a device with a camera for video input.	
Success End Condition	Successfully show the gesture.	
Failed End Condition	Unable to capture video by system.	
Primary Actors: **Secondary Actors:**	User System	
Trigger	The user needs to clicks "Allow" button for granting camera access.	
Main Success Flows	**Step**	**Action**
	1	The user grants camera access, allowing the system to capture video input.
	2	The system provides an interface where the user can position themselves within the camera frame.
	3	The user performs the sign language gesture, ensuring it is clearly visible within the camera frame.
	4	The system captures the video input of the user's sign language gesture.
	5	The system processes the captured video input using a sign language recognition algorithm.
	6	The system recognizes and analyzes the sign language gesture from the video input.
	7	Display the result to the user.
Alternative Flows	**Step**	**Branching Action**
	6a	The system could not recognize the gesture.
	6a1	User will get an error message.
	6a2	The system again provides an interface where the user can position themselves within the camera frame.
Quality Requirements	**Sl.**	**Requirement**
	1	System must check the gesture and display the result very quickly.

Table 12: Match Feature

Use Case	Match Feature	
Goal	To identify and match sign language features accurately within a given input.	
Preconditions	The system must be initialized and ready to receive input.	
Success End Condition	The system successfully matches the sign language features in the input and generates the corresponding interpreted message.	
Failed End Condition	The system fails to accurately match the sign language features in the input.	
Primary Actors: **Secondary Actors:**	System N/A	
Trigger	The user provides sign language input to the system.	
Main Success Flows	**Step**	**Action**
	1	The user performs the sign language gestures, which are captured by the system.
	2	The system analyzes the input and extracts the relevant sign language features.
	3	The system compares the extracted features with the pre-trained sign language feature database.
	4	If a match is found, the system generates the corresponding interpreted message
	5	The system displays the interpreted message to the user.
Alternative Flows	**Step**	**Branching Action**
	4a	The gesture is not validated.
	4a1	User will get an error message.
	4a2	User again shows the gesture to the system until find the match.
Quality Requirements	**Sl.**	**Requirement**
	1	System must check the input and display the result very quickly.

Table 13: Capture Feature Point

Use Case	Capture Feature Point	
Goal	Capture the gesture and analysis the point.	
Preconditions	User has access to a device or system capable of capturing sign language gestures	
Success End Condition	Feature points are successfully captured and processed for sign language detection.	
Failed End Condition	Gesture is unable to recognize	
Primary Actors: **Secondary Actors:**	System N/A	
Trigger	User initiates the capture feature points process.	
Main Success Flows	**Step**	**Action**
	1	User Initiates the capture feature points process through the sign language detection system.
	2	The system activates the camera or sensor to capture the user's sign language gesture.
	3	The system analyzes the captured data to identify relevant feature points in the gesture.
	4	Feature extraction algorithms or computer vision techniques are applied to extract the feature points.
	5	The system processes and maps the feature points to a predefined sign language database or model for recognition.
	6	The system interprets the captured feature points and produces a corresponding output, such as translated text or an action.
	7	The output is displayed or provided to the user, conveying the recognized sign language gesture.
Alternative Flows	**Step**	**Branching Action**
	1a.	If the captured gesture is not recognized as a valid sign language gesture, the system may prompt the user to repeat the gesture or provide feedback on the error.
	3a.	If the captured data does not contain enough feature points for accurate detection, the system may request the user to reposition their hand or improve lighting conditions.
	5a	If the feature points cannot be accurately mapped to a known sign language gesture in the database or model, the system may display an error message or prompt the user to try again.
Quality Requirements	**Step**	**Requirement**
		Not applicable.

Table 14: Capture Video

Use Case	Capture Video	
Goal	Provide input data for real-time analysis and interpretation of sign language gestures.	
Preconditions	System must have permissions to access and utilize the device's camera.	
Success End Condition	The desired video footage of the user performing sign language gestures is successfully captured.	
Failed End Condition	The system is unable to capture the video.	
Primary Actors: Secondary Actors:	System User	
Trigger	User selects the button 'Start Web Cam'.	
Main Success Flows	**Step**	**Action**
	1	The user initiates the video capture process through the application's user interface.
	2	The device's camera starts recording video footage of the user's sign language gestures.
	3	As the user performs the sign language gestures, the system continuously analyzes the video frames in real-time.
	4	The system uses computer vision techniques, such as image segmentation, motion detection, and pattern recognition to identify and classify the sign language gestures.
	5	The system generates textual or spoken language output corresponding to the recognized sign language gestures mapping to a predefined database.
	6	The system may also display visual feedback on the screen such as providing suggestions for corrections.
	7	The video capture continues until the user manually stops the process or a predefined time limit is reached.
	8	The system finalizes the process providing the user with the corresponding output.
Alternative Flows	**Step**	**Branching Action**
	1	If the system encounters technical issues or errors during video capture, it displays an error message and prompts the user to retry or seek technical support.
	2	If the user's sign language gestures are not clearly visible or recognizable in the video footage, the system may request the user to reposition themselves or adjust the lighting conditions for better capture quality.
	3	The system may provide an option for the user to review the captured video footage along with the interpreted output for further analysis or training purposes.
Quality Requirements	**Step**	**Requirement**
	1	**Proper Lighting Conditions:** Adequate lighting conditions are essential for capturing clear and accurate video data. Sufficient lighting helps enhance the visibility of sign language gestures and facilitates accurate analysis and interpretation by the
	2	**Proper Camera Placement:** The camera angle, distance, and orientation should be adjusted to ensure the user's hands and face are clearly visible within the frame.

Table 15: Capture Gesture

Use Case	Capture Gesture	
Goal	Enable the system to recognize, interpret, and process sign language gestures performed by the user.	
Preconditions	User has access to a device or system capable of capturing sign language gestures.	
Success End Condition	The desired image of the user performing sign language gestures is successfully captured.	
Failed End Condition	The system is unable to capture the gesture.	
Primary Actors: **Secondary Actors:**	System User	
Trigger	User selects the button 'Start Web Cam'.	
Main Success Flows	**Step**	**Action**
	1	The user initiates the image capture process through the application's user interface and shows gesture.
	2	The system captures an image of the user's hand gesture using the device's camera or image capturing functionality.
	3	The system analyzes the captured image using computer vision techniques, feature extraction, or other relevant algorithms.
	4	The captured image is compared against the predefined gesture database model to determine the corresponding sign language gesture.
	5	The system generates output, such as interpreted text or spoken language, based on the recognized gesture.
	6	The output is displayed to the user
Alternative Flows	**Step**	**Branching Action**
	1	If the system encounters technical issues or errors during gesture capture, it displays an error message and prompts the user to retry or seek technical support.
	2	If the user's sign language gestures are not clearly visible or recognizable in the image, the system may request the users to reposition themselves or adjust the lighting conditions for better capture quality.
	3	The system may provide an option for the user to review the captured video footage along with the interpreted output for further analysis or training purposes.
Quality Requirements	**Step**	**Requirement**
	1	**Proper Lighting Conditions:** Adequate lighting conditions are essential for capturing clear and accurate video data. Sufficient lighting helps enhance the visibility of sign language gestures and facilitates accurate analysis and interpretation by the
	2	**Proper Camera Placement:** The camera angle, distance, and orientation should be adjusted to ensure the user's hands and face are clearly visible within the frame.

Table 16: Display Result

Use Case	Display Result	
Goal	Display the converted text or images	
Preconditions	Capture the gesture or convert text to images or input the text and show the images	
Success End Condition	Successfully converted text or images, images to texts and texts to speech.	
Failed End Condition	Text or images, images to texts/text to speech not converted due to some technical condition	
Primary Actors: **Secondary Actors:**	System N/A	
Trigger	Convert	
Main Success Flows	**Step**	**Action**
	1	The user accesses the application or system interface.
	2	The user enters a text string into the input field, representing the message they want to communicate.
	3	The system receives the input text and initiates the three conversion processes simultaneously: sign language interpretation, text-to-speech synthesis, and text-to-sign conversion.
	4	The system generates the results and displays them in a table format with three columns: "Sign Language," "Speech," and "Text-to-Sign."
	5	In the "Sign Language" column, the system renders a video or animated representation of the sign language interpreter performing the corresponding gestures based on the input text.
	6	In the "Speech" column, the system plays the synthesized speech that corresponds to the input text, using the text-to-speech synthesizer.
	7	In the "Text-to-Sign" column, the system displays the textual representation of the sign language gestures produced by the text-to-sign converter.
	8	The user may provide additional input or interact with the system as needed, initiating the process again to generate new results.
Alternative Flows	**Step**	**Branching Action**
	8a	If the input text is invalid or not supported by the system, an error message is displayed, and the user is prompted to enter a different text string.
	5a	If there are technical issues with any of the conversion components (sign language interpreter, text-to-speech synthesizer, or text-to-sign converter), an error message is displayed, and the user is notified that the desired results cannot be generated at the moment.
Quality Requirements	**Step**	**Requirement**
		N/A

8 Activity Diagram

8.1 Registration

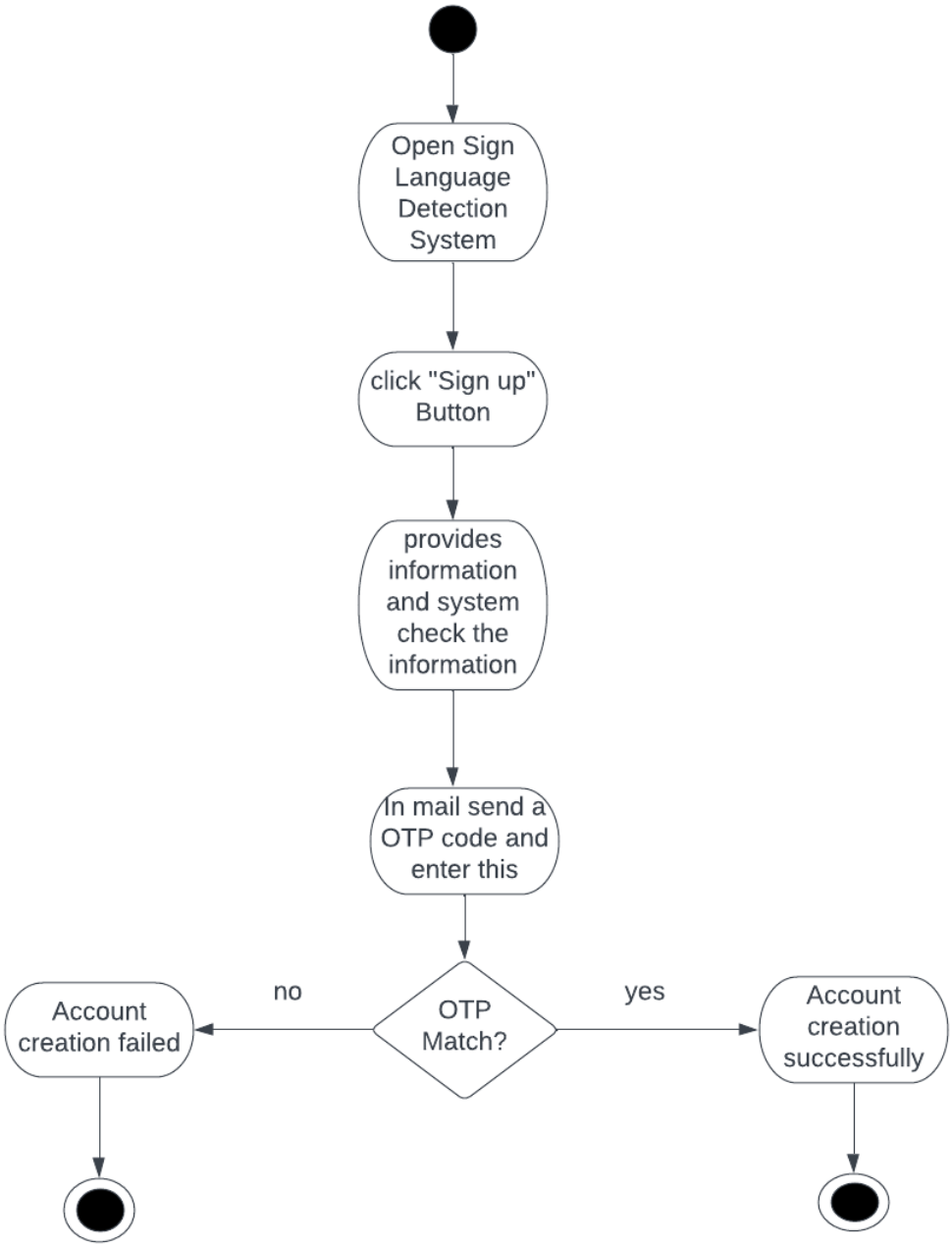

Figure 2: Registration

8.2 Log in

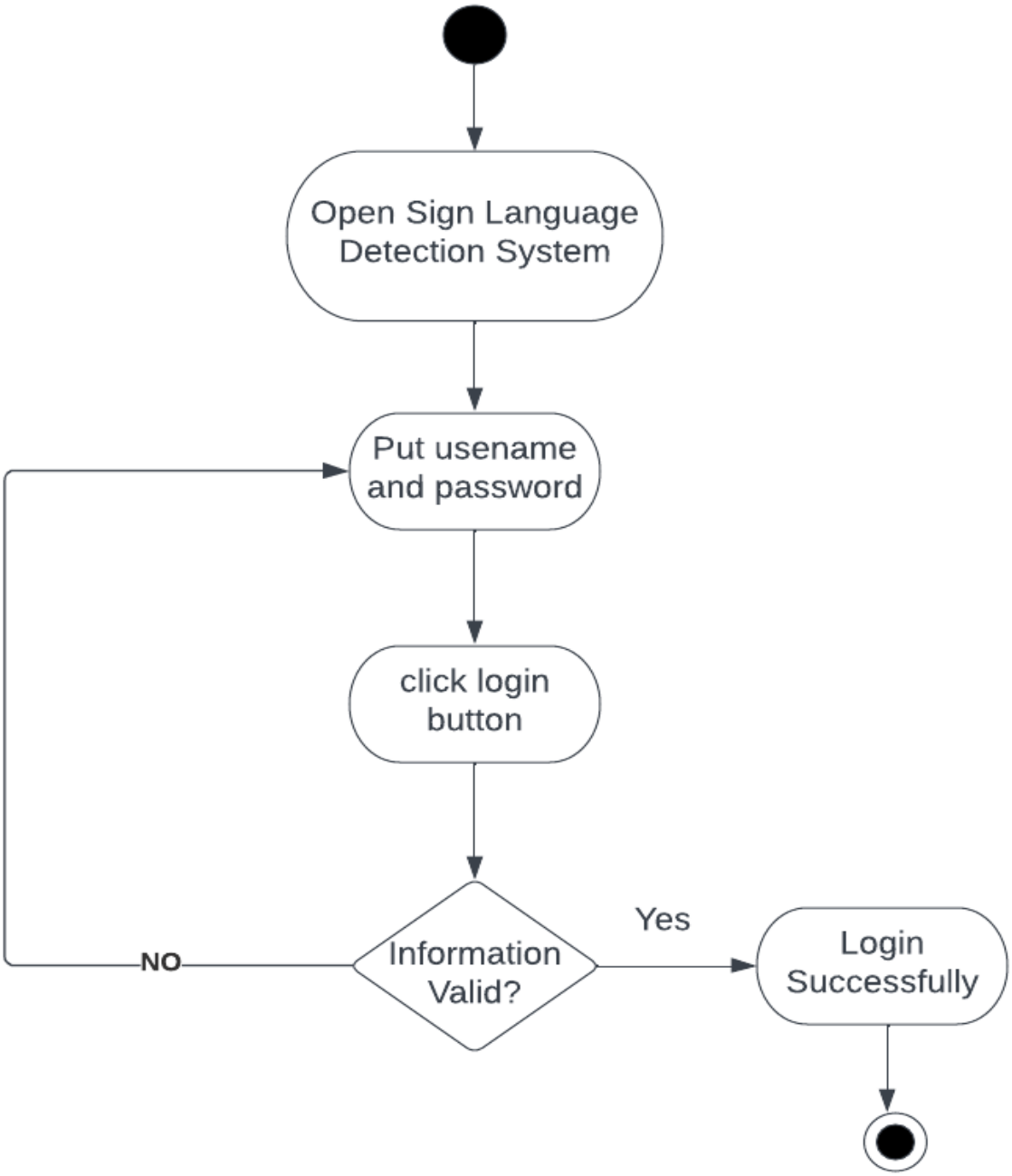

Figure 3: Log in

8.3 Forgot Password

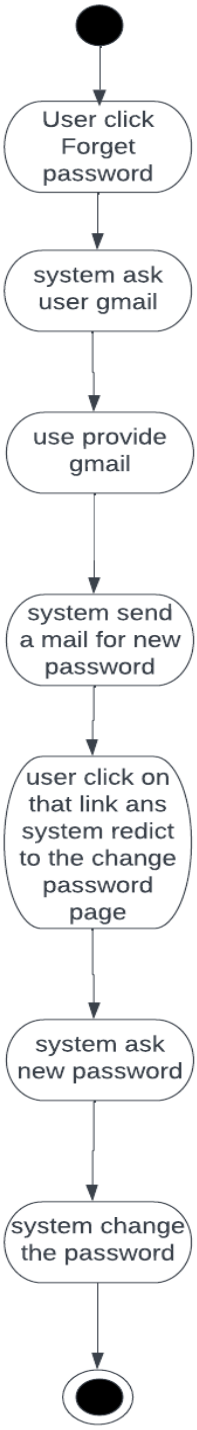

Figure 4: Forgot Password

8.4 Change Password

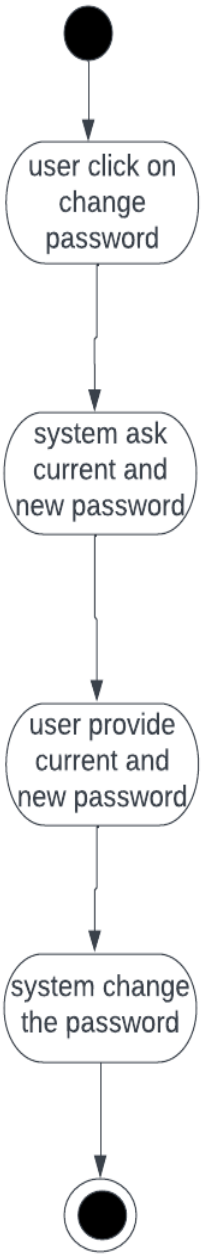

Figure 5: Change Password

8.5 Edit Profile

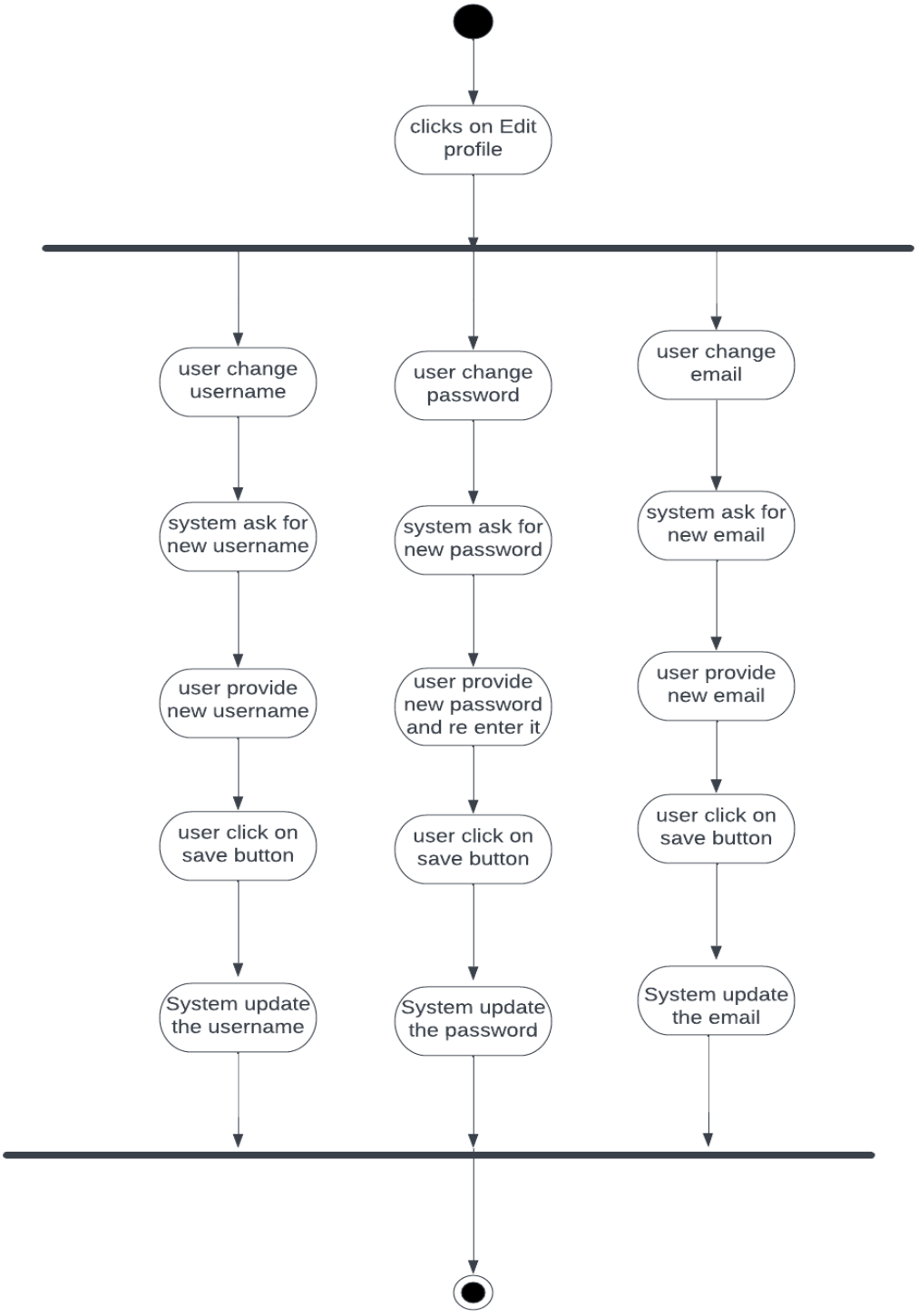

Figure 6: Edit Profile

8.6 Start Webcam

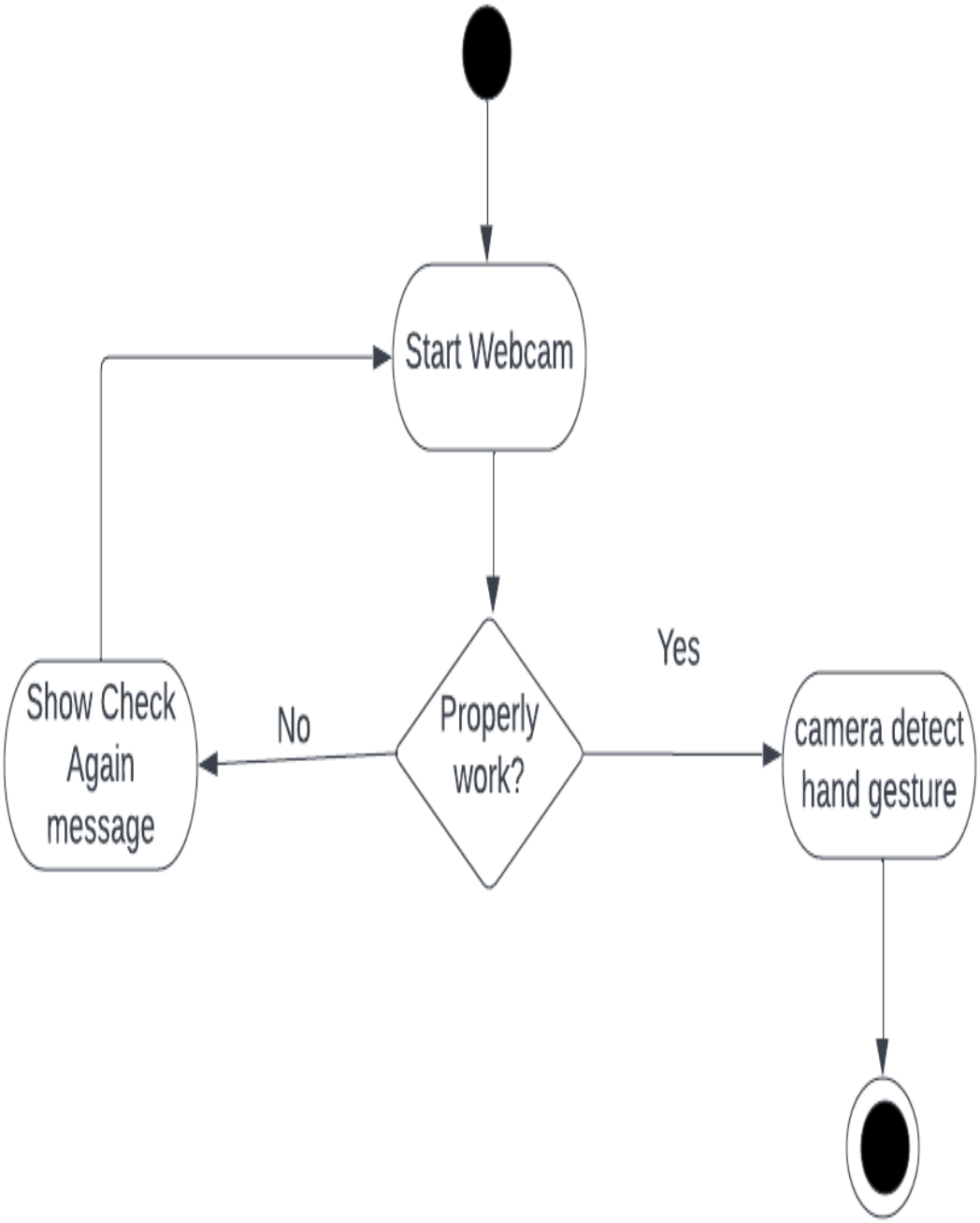

Figure 7: Start Webcam

8.7 Enter Plain Text

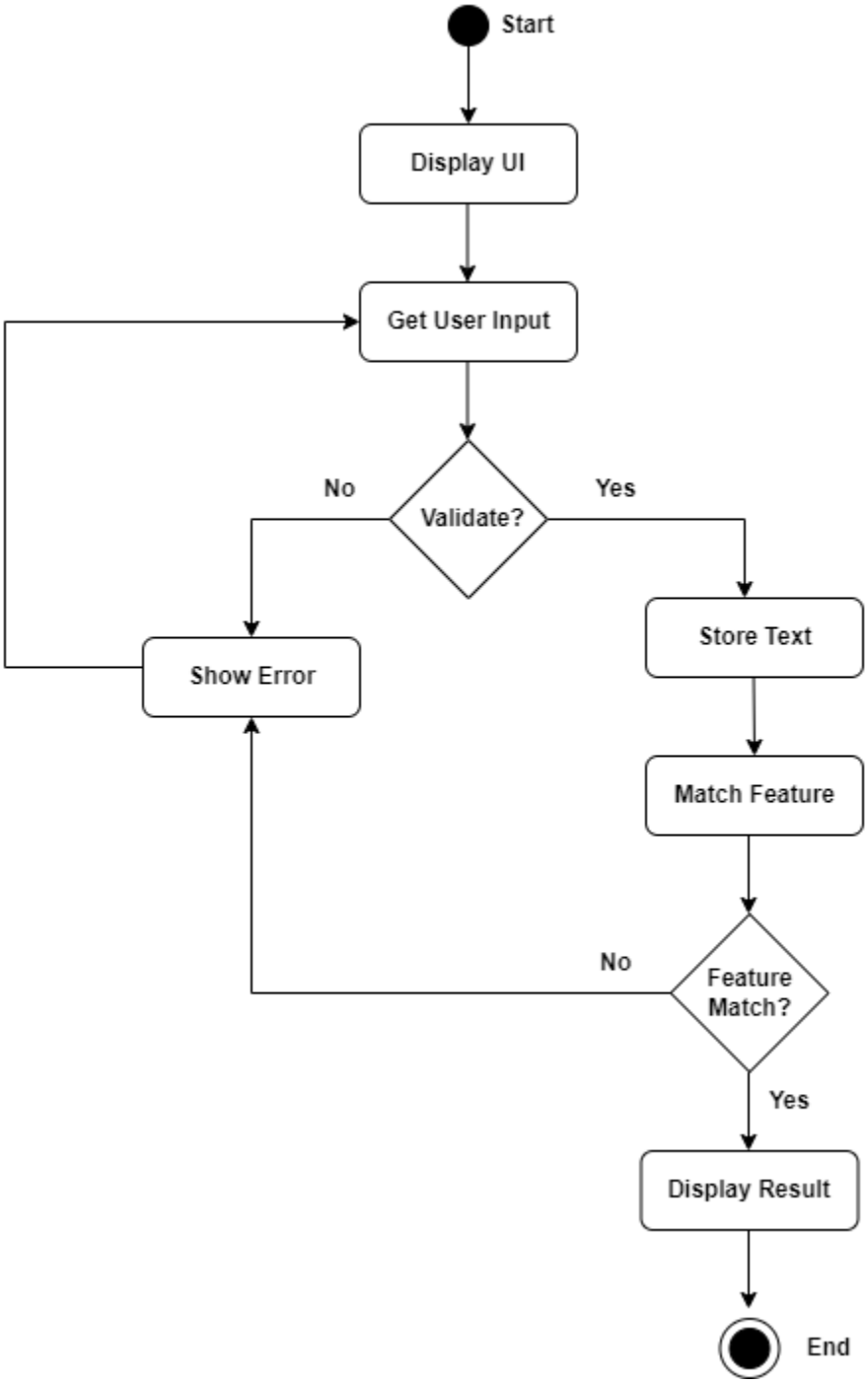

Figure 8: Enter Plain Text

8.8 Convert Text to Speech

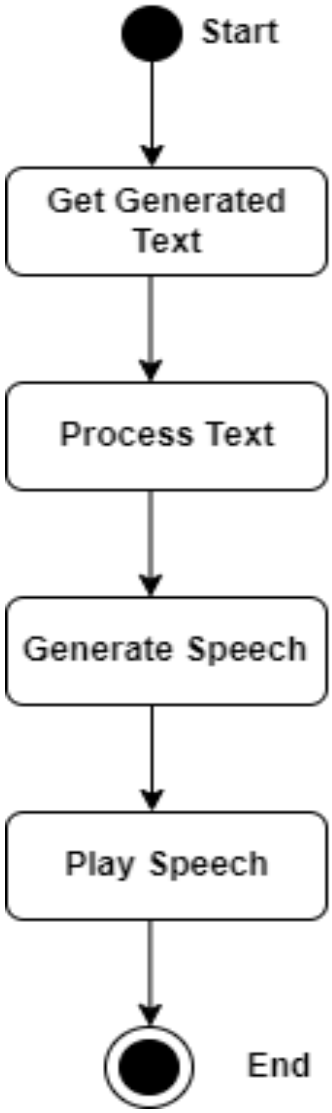

Figure 9: Convert Text to Speech

8.9 Translate Gesture

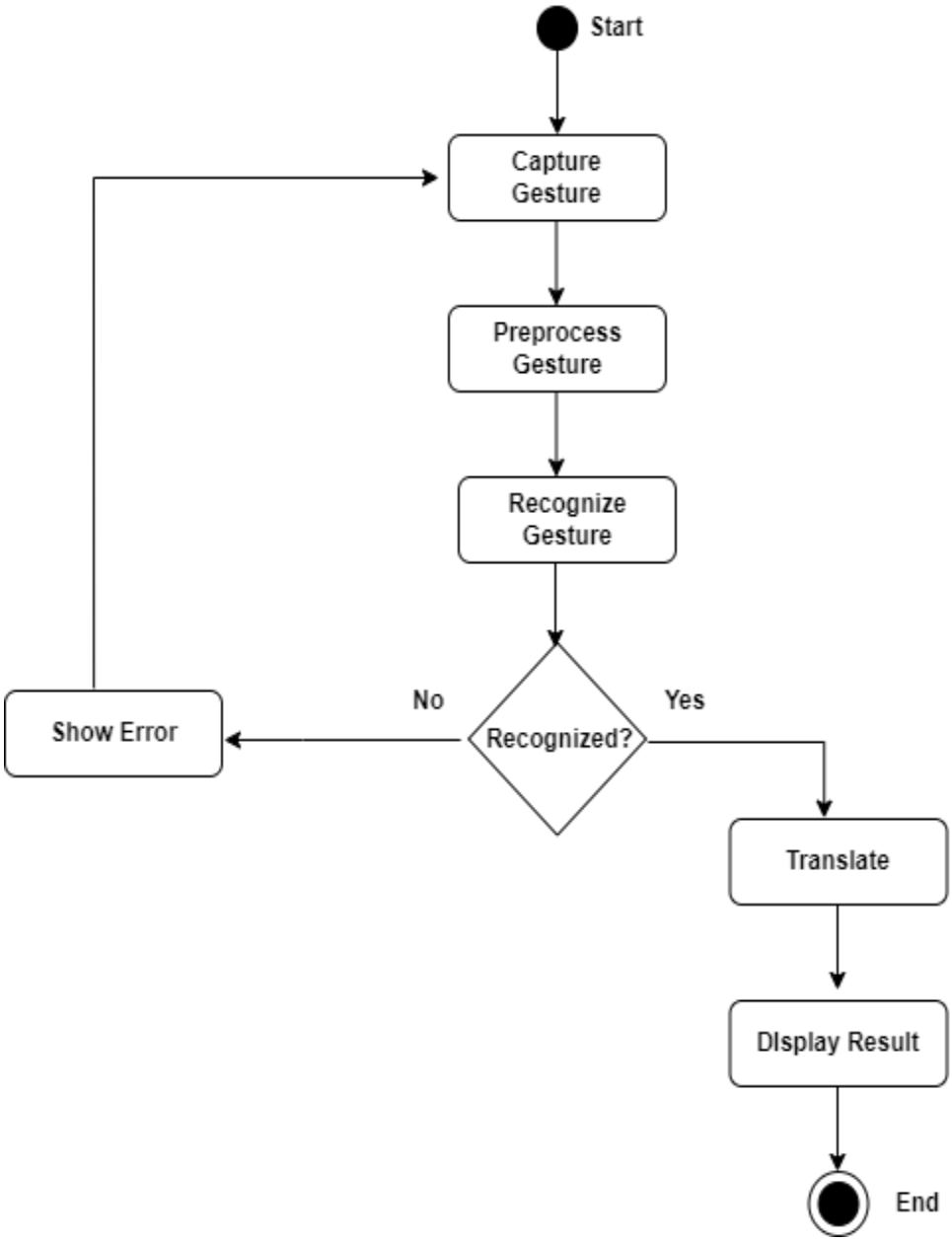

Figure 10: Translate Gesture

8.10 Show Gesture

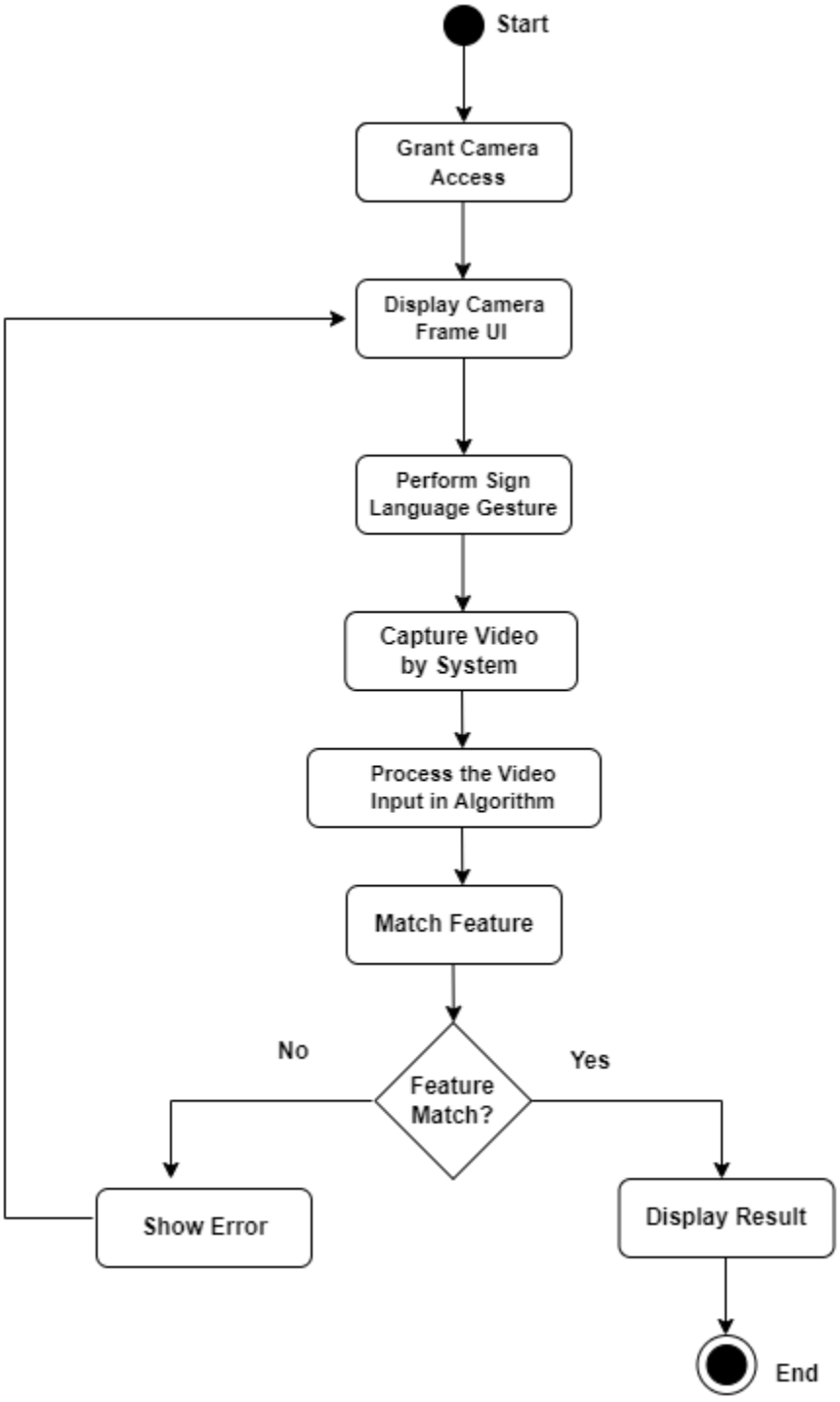

Figure 11: Show Gesture

8.11 Match Feature

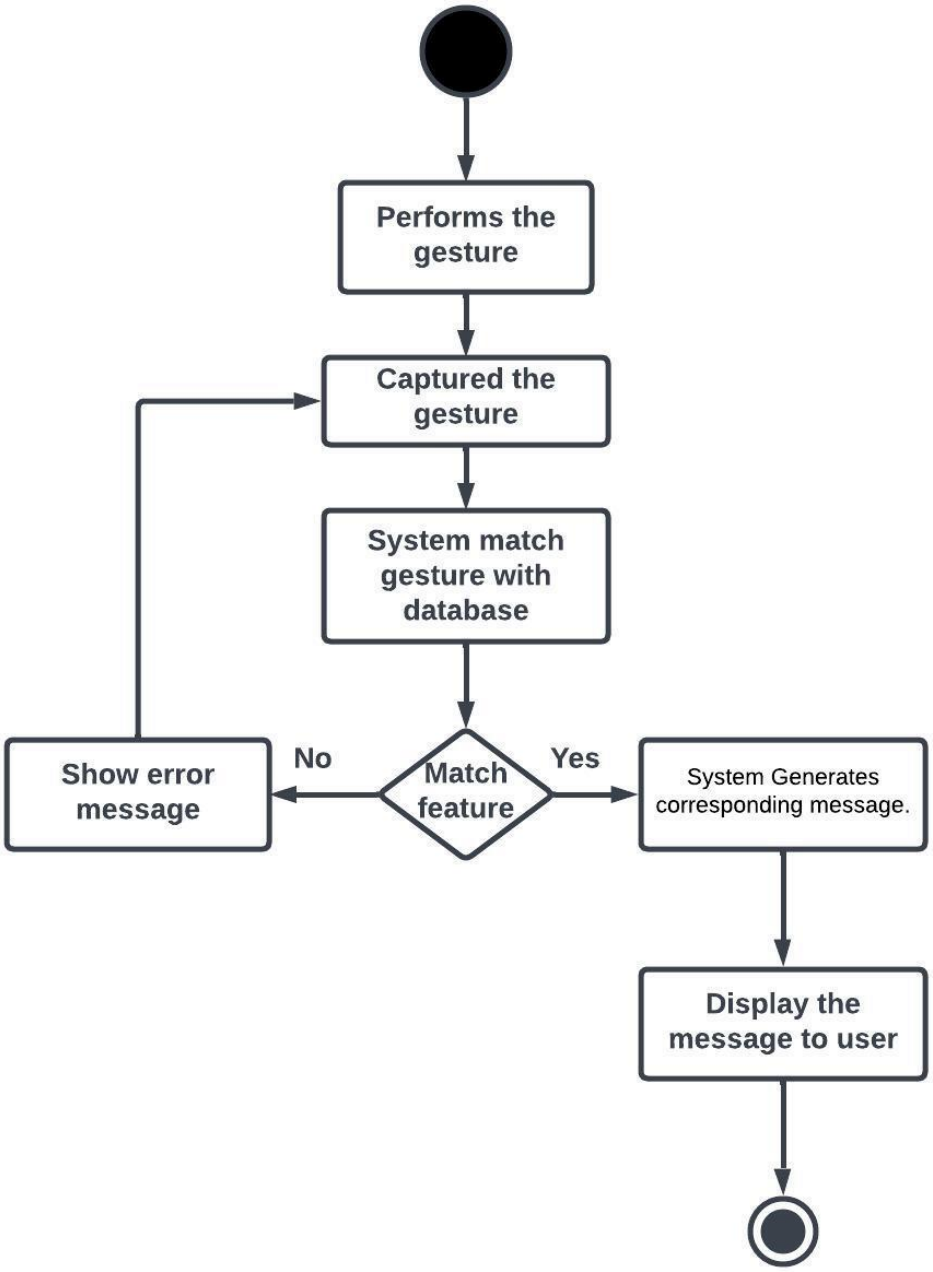

Figure 12: Match Feature

8.12 Capture Feature Point

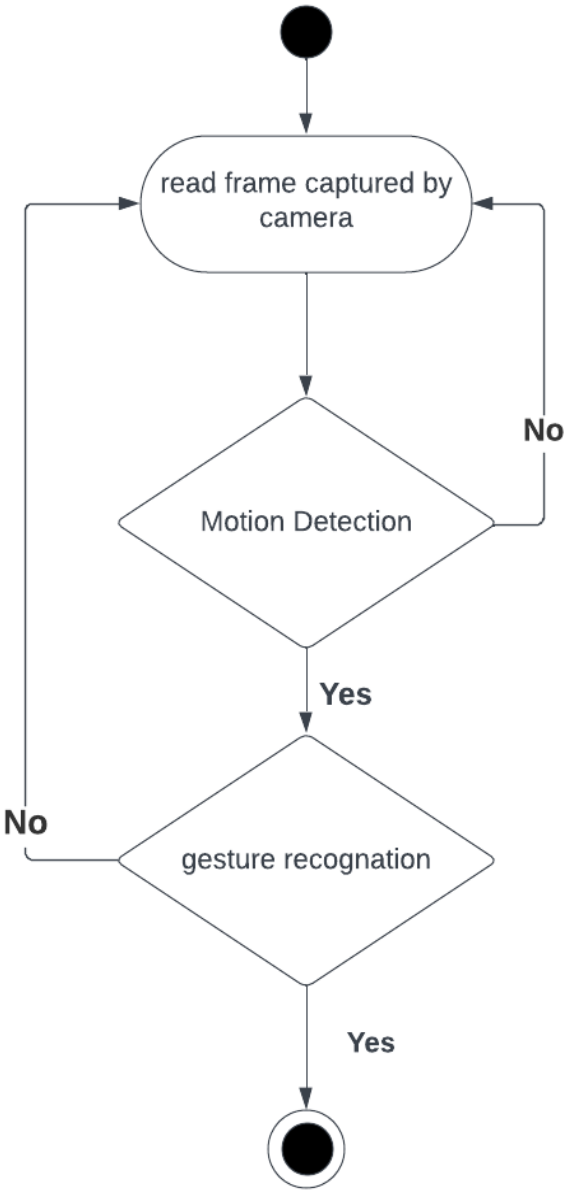

Figure 13: Capture Feature Point

8.13 Capture Video

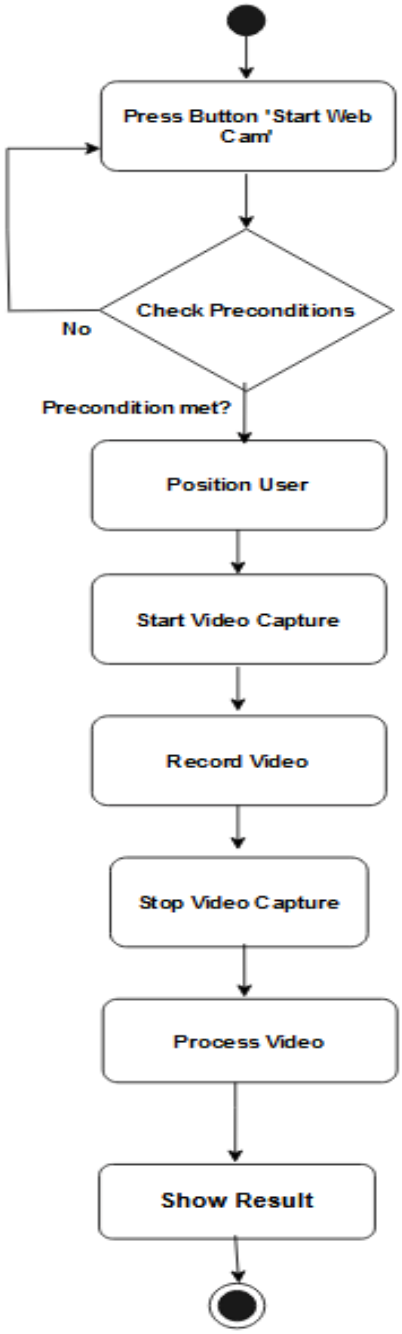

Figure 14: Capture Video

8.14 Capture Gesture

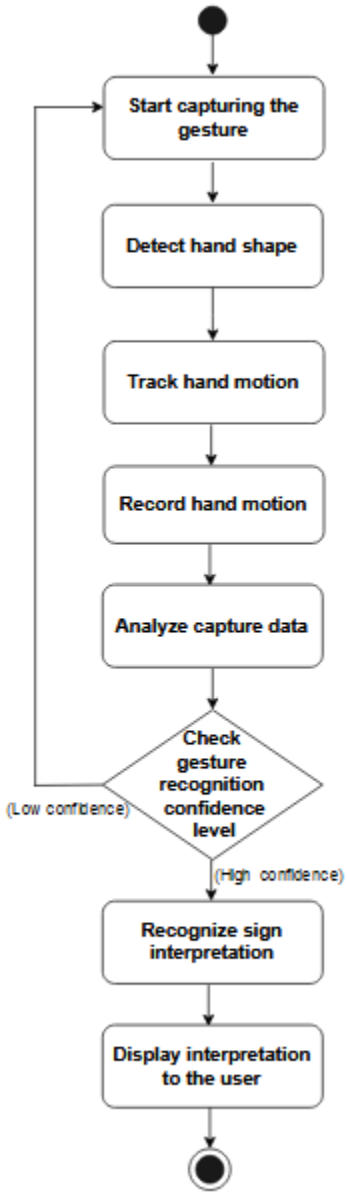

Figure 15: Capture Gesture

8.15 Display Result

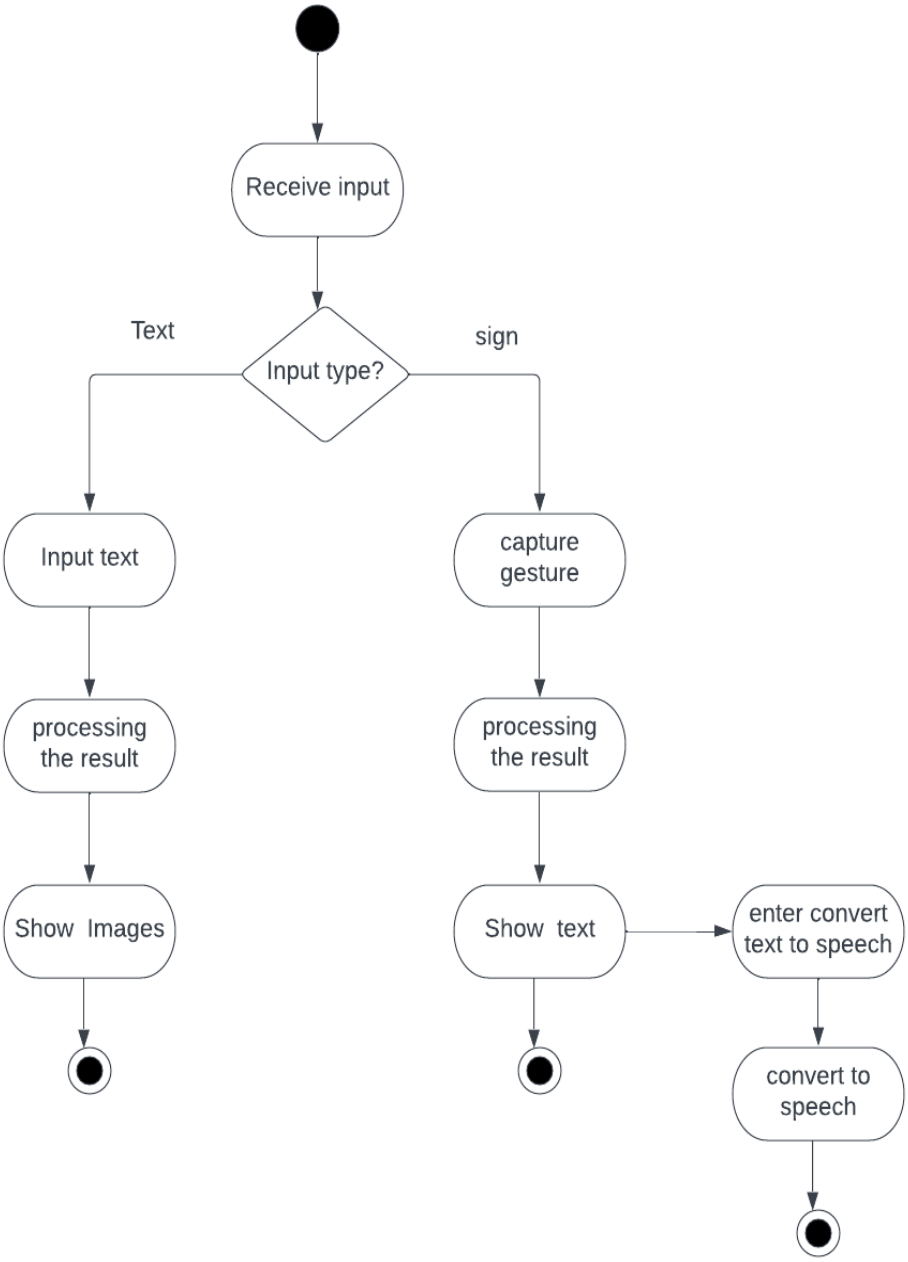

Figure 16: Display Result

9 Appendix

9.1 Prioritization of requirements

We've prioritized the functional requirements by following Three-level Scale technique.

9.1.1 Three-level Scale

When a Business Analyst categorizes the requirements in any of the ordering or ranking scale, it is subject to the analyst's understanding of the business. Many analysts suggest that this method has some drawbacks and advocate methods that have more than one scale.

9.1.2 Prioritization of the requirements of Sign Language Recognition

FR1-High priority: Its essential requirement for our system. User registration is obvious to run this application.

FR2-High priority: It is important to this system. If gesture is not valid then user doesn't able to know the result

FR3-High priority: It is important to this system. The system translates sign image to text so that it contains a meaningful word and normal people understand it.

FR4-High priority: If sign language doesn't translate properly then it doesn't show correct value so that the meaning is complex for this.

FR5-Medium priority: User add gesture in database so that when match it will help also add and remove gesture.

FR6-High priority: It is important to this system. System translate text to sign image so that it contains a meaningful word for disabled person.

FR7-High priority: System display the result show to generate sign.

FR8-High priority: System must be converting the text data to image data this is our system main motive.

FR9-Medium priority: Sometime convert text to speech but always it is not necessary to communicate.

FR10-Medium priority: Users sometimes change their information. User will need to update for later use.

FR11-Medium priority: The user will be able to log out of his account at the end of his need. Users will need to login again for later use.

9.2 Traceability Matrix

Use Cases:

UC1: Registration

UC2: Log in

UC3: Activation Account

UC4: Forgot Password

UC5: Change Password

UC6: Edit Profile

UC7: Start Webcam

UC8: Enter Plain Text

UC9: Convert Text to Speech

UC10: Translate Gesture

UC11: Show Gesture

UC12: Match Feature

UC13: Capture Feature Point

UC14: Capture Video

UC15: Capture gesture

UC16: Display Result

Functional Requirements:

FR1-User Login and Registration

FR2-Valid Gesture

FR3-Display Text

FR4-Sign Language Translation

FR5-Gesture Database Management

FR6-Text Input Field

FR7-Display Sign

FR8- Convert Text to Sign Language

FR9-Convert Text to Speech

FR10-Edit Profile

FR11-Logout

Table 17 :Traceability Matrix

FR/UC	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12	UC13	UC14	UC15	UC16
FR1	✓	✓	✓	✓	✓											
FR2										✓	✓	✓	✓	✓	✓	
FR3										✓	✓	✓	✓	✓	✓	✓
FR4							✓			✓	✓	✓	✓	✓	✓	✓
FR5	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓
FR6								✓								
FR7								✓				✓				✓
FR8								✓				✓				✓
FR9									✓							
FR10						✓										
FR11		✓														

10 References

1. IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
2. Software Engineering 9th Edition by Lan Sommerville.
3. Requirements Engineering Fundamentals by Klaus Pohl
4. Database System Concepts 6th Edition by Abraham Silberschatz.

11 Database Design

11.1 Schema of the database

userinfo(email,Upassword,username)
images(name,location)

11.2 Schema Diagram

Figure 17: Schema Diagram