



**THE
UNIVERSITY OF
LAHORE**

**Real-Time Sign Language Transcription
Final Year Project
PHASE – I**

Submitted by

Hamza Tariq	70113214
Hussnain Tariq	70110706
Hamza Akhtar	70111155

Project Supervisor

Sir Sayed Zishan Ali

BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING

DEPARTMENT OF SOFTWARE ENGINEERING, UNIVERSITY OF LAHORE

January 20, 2023

FINAL YEAR PROJECT PHASE-I DOCUMENTATION

STATEMENT OF SUBMISSION

Submitted to the University of Lahore in partial fulfillment of the requirement for the award of degree of Bachelors of Science in Software Engineering (BSSE)

By: Sap ID: _____

By: Sap ID: _____

By: Sap ID: _____

Project Advisor: _____

Sayed Zishan Ali

Lecturer at University of Lahore

Department of Software Engineering

Abstract

Sign-language communication plays a crucial role in fostering inclusive communication for individuals with hearing impairments. This paper introduces a pioneering application, the Sign-Language Transcription application, designed to bridge communication gaps by providing real-time transcription of sign language gestures into text. Leveraging advanced gesture recognition and machine learning technologies, the application offers an accessible and user-friendly platform for users to seamlessly transcribe sign language messages.

The Sign-Language Transcription application encompasses a comprehensive system architecture, including a user interface, gesture recognition module, transcription engine, user history tracker, and external messaging integration. The application's innovative real-time transcription feature enables users to receive instantaneous text representations of sign language gestures, fostering efficient and inclusive communication.

Dedication

This work is dedicated to individuals with hearing impairments, to whom effective communication is both a challenge and a necessity. The Sign-Language Transcription application aims to contribute to overcoming these challenges, providing a tool for enhanced communication and understanding. May it serve as a valuable resource for those seeking more accessible means of expression.

Acknowledgement

We extend our sincere gratitude to all those who contributed to the development and realization of the Sign-Language Transcription application. Our appreciation goes to the users and experts who provided invaluable insights and feedback throughout the development process. Special thanks to the dedicated team members who worked tirelessly to bring this application to fruition. Additionally, we acknowledge the support from our institutions and the broader community, without whom this project would not have been possible. Your collective efforts have played a crucial role in advancing inclusive communication for individuals with hearing impairments.

Hamza Tariq
Hussnain Tariq
Hammad Akhtar

Table of Contents

	1
	1
Chapter 1	.
Introduction to the problem	.
1.1 Introduction:	.
1.2 Purpose:	.
1.3 Objective	.
1.4 Existing Solution	.
1.5 Proposed Solution	.
Chapter 2	.
Software Requirement Specification	.
2.1 Introduction	.
2.1.1 Purpose	.
2.1.2 Scope	.
2.1.3 Definitions, acronyms, and abbreviations	.
2.1.4 References	.
2.1.5 Overview	.
2.2 Overall description	.
2.2.1 Product Perspective	.
2.2.2 Product Functions	.
2.2.3 User Characteristics	.
2.2.4 Constraints	.
2.2.5 Assumptions and dependencies	.
2.2.6 Apportioning of requirements	.
2.3 Specific Requirements	.
2.3.1 Functional Requirements	.

2.3.2 Non-Functional requirements	.
3.Appendixes	.
4. index	.
Chapter 3	.
Use Case Diagrams	.
Chapter 4	.
Design	.
4.1 Architecture Diagram	.
4.2 ERD Diagram	.
4.2.2 Data Dictionary	.
4.3 Data Flow Diagram	.
4.4 Class Diagram	.
4.5 Activity Diagrams	.
4.6Sequence Diagrams	.
4.7 Collaboration Diagrams	.
4.8 State Transition Diagram	.
4.9 Component Diagram	.
4.10 Deployment Diagram	.

CHAPTER 1

INTRODUCTION TO PROBLEM

Introduction

The objective of this Final Year Project (FYP) is to pioneer an innovative system that transcribes Sign Language into natural language text, bridging communication gaps for the Deaf and hard of hearing community. Sign Language is a rich and complex visual language used by the Deaf, and this project aims to provide a means for the wider population to understand and engage with Sign Language conversations. In this project, we embark on a unique approach where Sign Language users perform signing while our technology translates their signs into coherent and understandable natural language. The significance of this endeavor lies in facilitating inclusive communication and breaking down the barriers faced by the Deaf community in accessing information and interacting with others.

The foundation of our project begins with the compilation of a comprehensive dataset comprising video recordings of Sign Language conversations. This dataset encompasses various sign languages and dialects, allowing us to create a diverse and representative sample. We meticulously preprocess the video data, ensuring clarity and consistency in the signing gestures.

Next, we delve into the intricacies of Sign Language linguistics and cognitive features. Through advanced computer vision techniques, we extract vital information from the signing videos, including hand movements, facial expressions, and body language. These features play a pivotal role in deciphering the intended message accurately.

Our journey through the project involves the exploration of cutting-edge Natural Language Processing (NLP) models, including recurrent neural networks (RNNs), long short-term memory networks (LSTMs), and state-of-the-art transformer-based models such as BERT and GPT-3. These models serve as the backbone of our system, translating the visual nuances of Sign Language into written text.

Transparency and interpretability are at the core of our system. We dedicate considerable effort to developing methods that provide insights into how the model interprets Sign Language gestures, ensuring that the translation aligns with the intended meaning. The user interface is thoughtfully designed to be intuitive, making it accessible for Sign Language users to input their signing and receive coherent text translations.

Throughout the project's lifecycle, we remain committed to ethical principles. We prioritize user data privacy, informed consent, and mitigating potential biases in the system. Additionally, we collaborate closely with members of the Deaf community and sign language experts to validate the accuracy and cultural sensitivity of our translations.

The anticipated outcome of our FYP is a robust Sign Language-to-text transcription system. This technology serves as a vital tool for bridging the communication gap between the Deaf community and the wider society. It empowers Sign Language users to engage in conversations, access information, and participate fully in various aspects of life. However, it's important to underscore that our system should complement, not replace, the importance of human interpreters in certain contexts.

Purpose
<p>The purpose of developing our project is to address the communication challenges faced by the Deaf and hard of hearing community. The need arises from the complexity of Sign Language, which creates barriers to effective communication. Our Sign Language-to-text transcription system aims to bridge this gap by translating Sign Language gestures into coherent natural language text.</p> <p>In the market and societal context, our project offers a transformative solution. The system has the potential to improve communication accessibility for the Deaf community, allowing them to engage more effectively with the wider society. This technology goes beyond mere conversation facilitation; it can enhance access to information, education, employment opportunities, and social interactions, contributing to a more inclusive and understanding society.</p> <p>The anticipated impact is profound, as our project seeks to empower individuals with hearing impairments, enabling them to participate fully in various aspects of life. It aligns with ethical principles, ensuring user data privacy and cultural sensitivity. Importantly, our technology is designed to complement human interpreters, emphasizing a balanced integration of technological advancements and human expertise in communication. Overall, the project aims to bring about positive changes in the lives of the Deaf and hard of hearing individuals, fostering inclusivity and equal participation in society.</p>

Objectives
<ol style="list-style-type: none"> 1. Develop a Functional Sign Language-to-Text Transcription System: Create an advanced system capable of accurately transcribing Sign Language gestures into natural language text, ensuring functionality and reliability. 2. Diverse and Representative Dataset: Compile a comprehensive dataset containing video recordings of Sign Language conversations, encompassing various sign languages and dialects to ensure the system's inclusivity and adaptability. 3. Implement Advanced Computer Vision Techniques: Utilize cutting-edge computer vision techniques to extract crucial information from signing videos, such as hand movements, facial expressions, and body language, for precise interpretation of Sign Language nuances. 4. Deploy State-of-the-Art NLP Models: Explore and integrate state-of-the-art Natural Language Processing (NLP) models, including recurrent neural networks (RNNs), long short-term memory networks (LSTMs), and transformer-based models like BERT and GPT-3, to enhance the accuracy of translations. 5. Ensure Transparency and Interpretability: Develop methodologies that provide insights into the model's interpretation of Sign Language gestures, ensuring transparency and interpretability to align translations with intended meaning. 6. User-Friendly Interface Design: Design an intuitive user interface that allows Sign Language users to input their gestures effortlessly and receive coherent text translations, ensuring accessibility and usability. 7. Adhere to Ethical Principles: Prioritize user data privacy, informed consent, and address potential biases in the system, maintaining a strong commitment to ethical considerations throughout the project lifecycle. 8. Collaborate with Deaf Community and Experts: Work closely with members of the Deaf community and sign language experts to validate the accuracy and cultural sensitivity of translations, incorporating valuable feedback into system refinement. 9. Empower Deaf Community Engagement: Empower Sign Language users to actively engage in conversations, access information, and participate fully in various aspects of life, contributing to increased inclusivity and understanding within society. 10. Complement Human Interpreters: Emphasize that the system serves as a supplementary tool rather than a replacement for human interpreters, recognizing and respecting the irreplaceable role of human expertise in certain contexts.

Existing Problems

1. **Limited Availability:** Human interpreters may not be readily available at all times, leading to delays in communication for the Deaf community. This limitation hinders spontaneous and real-time interactions.
2. **Cost:** Hiring professional human interpreters can be expensive, making it difficult for some individuals or organizations to afford constant interpretation services. This financial barrier restricts access to effective communication.
3. **Scalability Issues:** Human interpreters may face challenges in scaling their services to meet the increasing demand, particularly in situations with a large number of Deaf individuals or simultaneous communication needs.
4. **Subjectivity and Variability:** Interpretation can be subjective, and individual interpreters may have different interpretations of the same Sign Language message. This subjectivity can lead to potential miscommunication and misunderstandings.
5. **Privacy Concerns:** In certain situations, relying on human interpreters may raise privacy concerns, especially when discussing sensitive or personal matters. Users may feel more comfortable with technology that ensures data privacy.
6. **Geographical Constraints:** Access to qualified interpreters may be limited in certain geographic areas, particularly in rural or remote locations. This can result in disparities in communication accessibility.
7. **Training and Certification Challenges:** Ensuring a consistent level of quality among interpreters requires standardized training and certification processes. Variability in interpreter skill levels can impact the quality of communication.

Proposed Solution

1. **Real-time Accessibility:** Our system aims to provide real-time translation, overcoming the delay associated with the availability of human interpreters. This feature is crucial for spontaneous and immediate communication.
2. **Cost-Effectiveness:** By automating the translation process, our system seeks to offer a cost-effective solution compared to hiring human interpreters. This can increase accessibility for individuals and organizations with budget constraints.
3. **Scalability:** The automated system is designed to be scalable, allowing it to handle a large number of simultaneous translation requests efficiently. This addresses the scalability issues faced by human interpreters.
4. **Consistency and Objectivity:** Machine learning models provide a consistent and objective approach to translation. By reducing subjectivity, our system aims to improve the accuracy and reliability of Sign Language translations.
5. **Privacy Considerations:** Our project places a strong emphasis on user data privacy. Unlike human interpreters, our system can ensure a level of privacy, especially in situations involving sensitive or personal information.
6. **Geographical Accessibility:** As a technology-driven solution, our system can be accessed remotely, promoting geographical accessibility in both urban and remote areas where access to qualified interpreters might be limited.
7. **Continuous Improvement:** Through machine learning, our system can continuously learn and improve its accuracy over time, adapting to various signing styles and nuances. This dynamic learning process enhances the quality of translations.

Chapter 2

Software Requirement Specification

2.1 Introduction

2.1.1 Purpose

The purpose of this Software Requirement Specification (SRS) document is to provide a comprehensive understanding of the Sign-Language Transcription application. It outlines the functional and non-functional requirements, design constraints, and interfaces necessary for the successful development and implementation of the application.

Intended Audience

The intended audience for this Software Requirement Specification includes, but is not limited to:

Development Team: Software engineers, programmers, and designers who will be involved in the development and implementation of the system.

Testing Team: Quality assurance professionals responsible for validating and verifying that the system meets the specified requirements.

Project Managers: Individuals overseeing the planning, execution, and monitoring of the project.

Stakeholders: Investors, sponsors, and any individuals or organizations with a vested interest in the successful development and deployment of the application.

Documentation Team: Writers responsible for creating user manuals, technical documentation, and other related materials.

End Users: Individuals who will interact with and benefit from the system, including drivers and administrators.

By addressing the needs of this diverse audience, this SRS aims to ensure a common understanding of the project's objectives, functionalities, and constraints, fostering effective collaboration and successful project outcomes.

2.1.2 Scope

The scope of this document encompasses the entire software development life cycle, from the

Real-Time Sign Language Transcription FYP-F23-27 Fall 2020-2024

conceptualization of the Sign-Language Transcription application to its deployment and maintenance. It serves as a guide for developers, designers, and stakeholders involved in the project.

2.1.3 Definitions, acronyms, and abbreviations

- GUI – Graphical user interface
- DB – Database
- SRS – Software requirement specification
- AI – Artificial Intelligence
- ASL – American Sign-Language

2.1.4 References

REFERENCES
<p>Slobin, Dan. (1999). Sign language transcription at the morphological level: the Berkeley Transcription System (BTS).</p> <p>(2023). A Survey on Indian Sign Language Translation Using Artificial Intelligence. 10.1007/978-981-99-3963-3 33.</p> <p>M. Papatsimouli et al., "Real Time Sign Language Translation Systems: A review study," 2022 11th International Conference on Modern Circuits and Systems Technologies (MOCAST), Bremen, Germany, 2022, pp. 1-4, doi: 10.1109/MOCAST54814.2022.9837666.</p> <p>A Survey of Advancements in Real-Time Sign Language Translators: Integration with IoT Technology</p>

2.1.5 Overview

This section provides an overview of the entire Software Requirement Specification, highlighting key chapters and their respective purposes. It aims to offer a quick reference guide for readers navigating through the document.

2.2 Overall description

2.2.1 Product Perspective

The Sign-Language Transcription application is designed to operate as an independent system, employing advanced AI and machine learning algorithms for real-time sign language recognition. It interfaces with various devices and platforms, striving for seamless integration into modern communication environments in the future.

- **Admin side**
used for the administration activities just like approval of user’s accounts, managing data for

analytics, training and monitoring.

- **User side**

Used to perform the transcription either video or real-time. Users can engage in real-time sign language conversations, with the application accurately transcribing their gestures into text.

2.2.1.1 System Interfaces:

Interaction with AI and machine learning components

2.2.1.2 User Interfaces:

User-friendly screens for gesture input and transcription output.

2.2.1.3 Hardware Interfaces:

The mobile application can be used on android/iOS and web application can be used on any device like laptop, mobile phone as long as it has active internet connection and is compatible with devices featuring cameras for capturing sign language gestures.

2.2.1.4 Software Interfaces:

Software interfaces includes the operating system for mobile, Android or IOS.

For web application, it just needs to have a browser and active internet connection.

2.2.1.5 Memory:

Utilization of primary and secondary memory or database as required.

2.2.2 Product Functions

The application performs the following key functions:

- Real-time recognition and transcription of sign language gestures into text.
- User-friendly interface design for both sign language proficient users and those less familiar with sign language.
- Accessibility features, including voice output and customizable font sizes.
- Continuous improvement based on user feedback and technological

More functions are formally defined in the tables below:

2.2.2.1 User functions

Table 1: User Function - Register

ID	FR_01
Name	User Register/Signup

Description	Input	Output	Requirements	Basic Workflow
User shall able to register through application	Username/email Password Full name	Creation of a new account	Database, Internet	Enter these valid inputs for the creation of the account

Table 2: User Function - Login

ID	FR_02			
Name	Login			
Description	Input	Output	Requirements	Basic Workflow
User shall be able to login to the application	Username/email Password	Provide access to dashboard upon successful login	Input validation Account verification Input	Enter inputs if valid, the system will go to it's dashboard

Table 3: User Function - Video Transcription

ID	FR_03			
Name	Video Transcription			
Description	Input	Output	Requirements	Basic Workflow

User shall be able to upload a video for transcription	Video of valid format	If video is of acceptable quality/clarity: Text script, an error message otherwise	Internet connectivity, local storage access	User selects the option for video transcription and uploads a video.
--	-----------------------	--	---	--

Table 4: User Function – Real-Time Transcription

ID	FR_04			
Name	Real-Time Transcription			
Description	Input	Output	Requirements	Basic Workflow
Users shall be able to transcribe in real-time	Camera stream	Text transcription	Internet connectivity, camera access	User selects the option for real-time transcription

Table 5: User Function – Track History

ID	FR_05			
Name	Track History			
Description	Input	Output	Requirements	Basic Workflow

User shall be able to look-up past transcripts	Interaction with the history tab/tile	List of past transcripts	Data storage	User would press on the history tab to get a list of past transcripts
--	---------------------------------------	--------------------------	--------------	---

2.2.3 User Characteristics

Users vary in sign language proficiency, technical expertise, and educational backgrounds. The application caters to a diverse user base, ensuring usability for both sign language experts and those less familiar with signing but a familiarity with ASL.

2.2.4 Constraints

Regulatory Policies: The application must comply with data protection regulations regarding the collection, storage, and processing of user information. User consent for data processing and clear privacy policies should be implemented.

Hardware Limitations: Compatibility and optimization considerations for diverse hardware environments, the application is dependent on camera hardware some of its features.

Interfaces to Other Applications: Integration constraints with existing applications and systems.

Multiple Sign-Languages: The fact that there are multiple Sign-languages used around the world poses a constraint on the application of which on to cater too.

Reliability Requirements: Mandated reliability standards to ensure consistent performance.

Safety and Security Considerations: Implementation of measures to address safety and security concerns in the application.

2.2.5 Assumptions and dependencies

This section outlines the assumptions made and dependencies identified for the successful implementation:

Assumptions:

1. **Stable Network Connectivity:** The assumption is made that users will have stable and reliable network connectivity for real-time data exchange and communication.
2. **Camera Access:** The application assumes that camera access is always available to use certain features.
3. **Familiarity with the language:** The application assumes that the user is familiar with ASL.
4. **User Device Compatibility:** The application assumes compatibility with a range of user devices, including smartphones and tablets, for optimal accessibility.

Dependencies:

1. **Internal APIs:** The project is dependent on internal API's to communicate the data between applications.
2. **Hardware Components:** Dependencies on specific hardware components, such as cameras.
3. **Third-party Software Libraries:** Dependencies on third-party software libraries, particularly for machine learning like computer vision (CV).
4. **Data Security Protocols:** The project relies on robust data security protocols to ensure the confidentiality and integrity of user and system data.

2.2.6 Apportioning of requirements

There is a requirement we have delayed until future version of the system. This include payment method for the users to purchase the subscription.

2.3 Specific Requirements

Every system has its own specific requirements according to its nature. The requirements is of two types including functional and non-functional requirements. These are as follow:

2.3.1 Functional Requirements

This section is describing the functional requirements at a sufficient level of detail for the designers to a design a system satisfying the user requirement and testes to verify that the system satisfies the requirement.

User

- User shall be able to Register
- User shall be able to login
- User shall be able to logout
- User shall be able to receive text transcription of their signed communication
- User shall be able to select video transcription
- User shall be able to select real-time transcription
- User shall be able to track history to past transcriptions
- User shall receive clear visual cues and feedback during sign language recognition and transcription

Admin

- Admin shall be able to create, view, edit, and delete user accounts
- Admin shall be able to manage user roles and permissions
- Admin shall be able to export and download conversation data for analysis or archiving
- Admin shall be able to delete conversation data upon request or according to data retention policies

2.3.2 Non-Functional requirements

This section outlines the non-functional requirements for the application, ensuring that the system meets certain quality attributes and performance criteria:

1. Scalability:

Requirement: The system should handle a large number of concurrent users and vehicles.

Rationale: To accommodate potential growth in user base

2. Security:

Requirement: Ensure the security of user data and communication between devices.

Rationale: Protect sensitive information and prevent unauthorized access or manipulation.

3. Reliability:

Requirement: The system should operate with high accuracy and minimal false positives.

The system should operate in real-time with minimal latency and function in various lighting conditions

Rationale: Ensure continuous and reliable service to users.

4. Usability:

Requirement: Provide a user-friendly interface for both the mobile and web applications.

Rationale: Enhance user experience and accessibility, regardless of technical expertise.

5. Accuracy:

Requirement: detection systems should be highly accurate and adoptive.

Rationale: Ensure precision in the resultant transcript.

6. Performance:

Requirement: The system should in real-time with minimal latency.

Rationale: Enhance the system's responsiveness.

7. Data Storage:

Requirement: Implement efficient data storage and retrieval mechanisms for historical data.

Rationale: Optimize storage resources and ensure quick access to past data for analysis.

3. Appendixes

- a. IoT (Internet of Things): A network of interconnected devices that can communicate and share data with each other over the internet.
- b. ML (Machine Learning): A subset of artificial intelligence that enables systems to learn and improve from experience without being explicitly programmed.
- c. SRS (Software Requirement Specification): A document that outlines the functional and non-functional requirements of a software system.
- d. Regulatory Compliance: Adherence to laws, regulations, and standards relevant to the development and operation of the software.
- e. API (Application Programming Interface): A set of rules that allows one software application to interact with another.

4. index

- A: Android, ASL

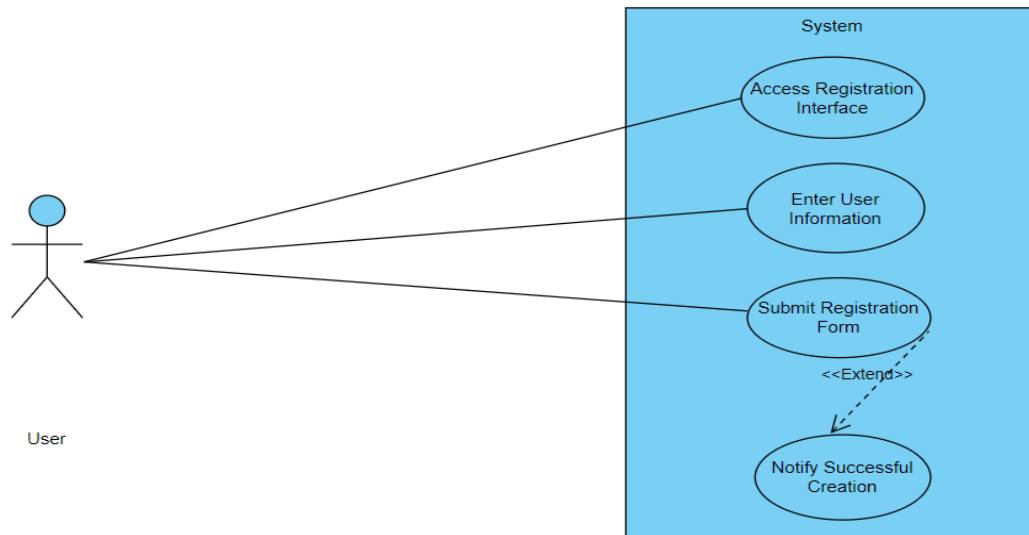
- C: Computer Vision, Communication Interfaces, Compatibility

Real-Time Sign Language Transcription FYP-F23-27 Fall 2020-2024

- H: Hardware Interfaces
- I: Interfaces, Introduction
- M: Machine Learning, Memory, Mobile Application
- N: Non-functional Requirements
- O: Operations
- P: Product Perspective, Purpose
- R: Regulatory Compliance
- S: Safety, Scalability, Security, Site Adaptation Requirements, Software Interfaces, System Interfaces
- U: User Characteristics, User Interfaces, Usability

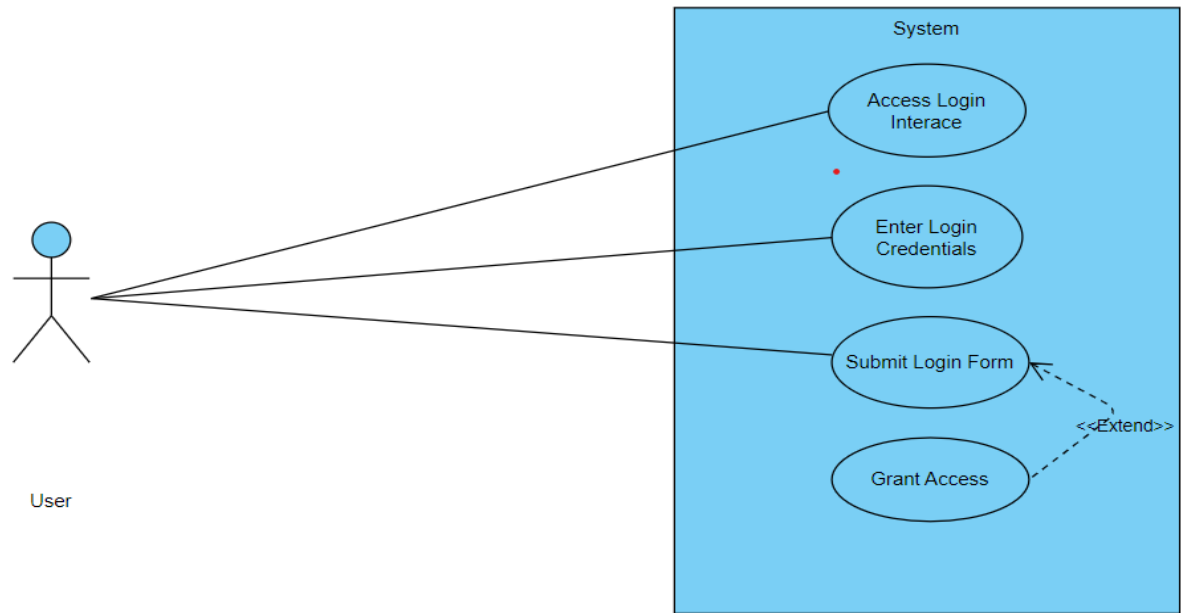
Chapter 3

Use-Cases

USE CASE 1:

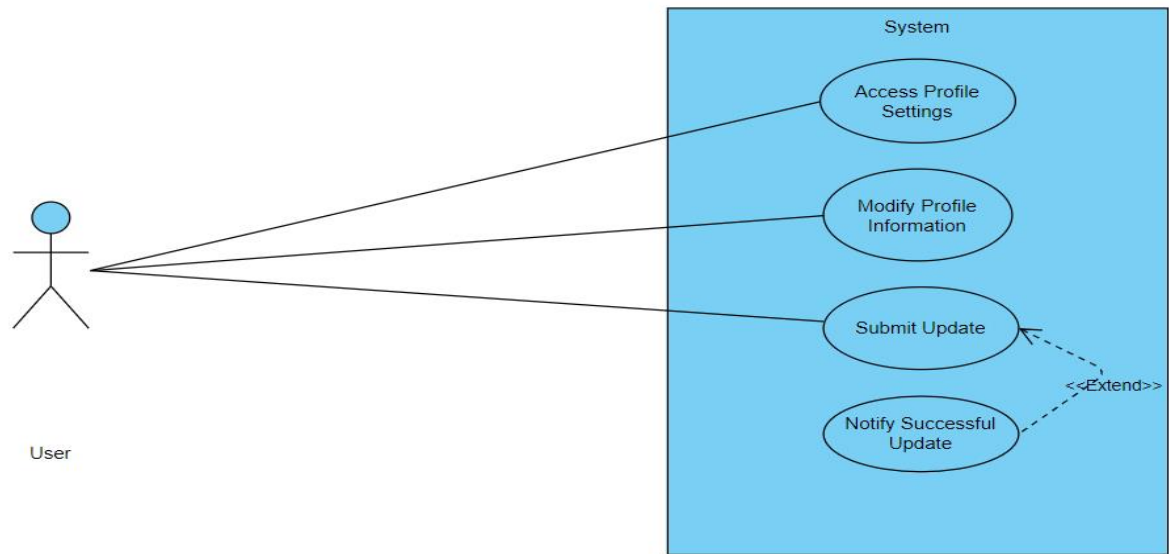
Use Case ID	UC_001	
Use Case Name	Registration/Sign up	
Description	This use case involves the process of creating a new account within the Sign Language Transcription System, allowing users to access and utilize the system's features.	
Primary Actor	User	
Secondary Actor	None	
Pre-Condition	The user must have access to the system's registration interface.	
Post-Condition	A new user account is successfully created, and the user gains access to the system.	
Basic Workflow	Actor Action	System Action
	<ul style="list-style-type: none"> ➤ The user accesses the registration interface. ➤ The user enters the required information, such as username, email, and password. ➤ The user submits the registration form. ➤ The system notifies the user of successful account creation. 	<ul style="list-style-type: none"> ➤ The system displays the account creation form. ➤ The system validates the entered information. ➤ The system processes the registration request and creates a new user account.
Alternate Flow	If the entered information is incomplete or fails validation: <ul style="list-style-type: none"> ➤ The system notifies the user of the validation error. ➤ The user corrects the information and resubmits the form. ➤ Steps 4 to 8 are repeated. 	

USE CASE 2:

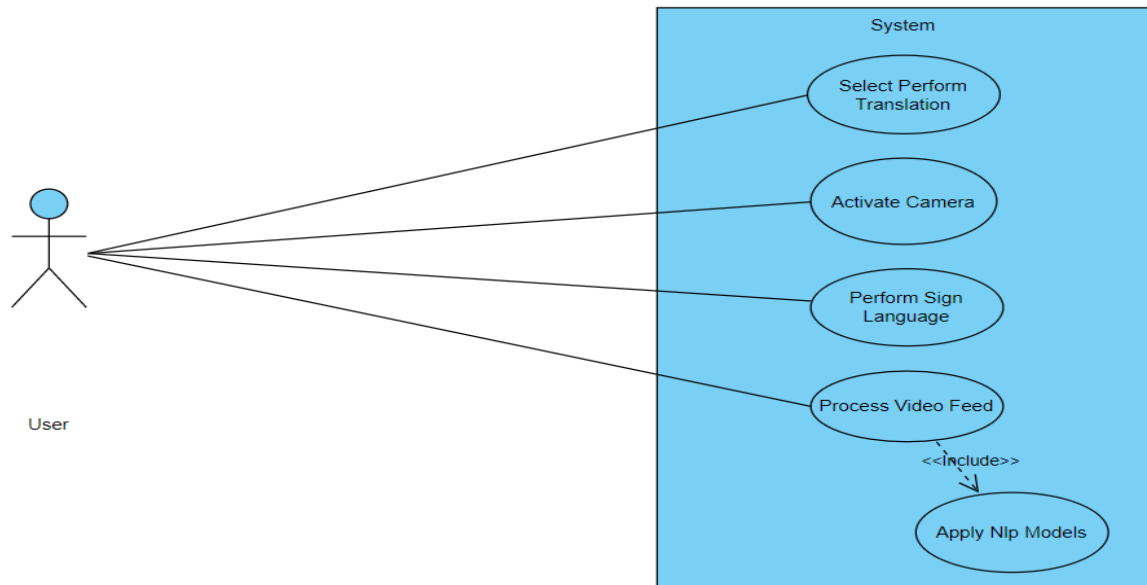


Use Case ID	UC_002	
Use Case Name	Login	
Description	User can login to the system	
Primary Actor	User	
Secondary Actor	None	
Pre-Condition	The user must have access to the system's registration interface.	
Post-Condition	A new user account is successfully created, and the user gains access to the system.	
Basic Workflow	Actor Action	System Action
	<ul style="list-style-type: none">➤ The user accesses the login interface.➤ The user enters their username/email and password.➤ The user submits the login form.	<ul style="list-style-type: none">➤ The system displays the login form.➤ The system validates the login credentials.➤ The system processes the login request and grants access to the user.
Alternate Flow	<p>If the entered credentials are incorrect:</p> <ul style="list-style-type: none">➤ The system notifies the user of the authentication failure.➤ The user retries the login with correct credentials.➤ Steps 4 to 6 are repeated.	

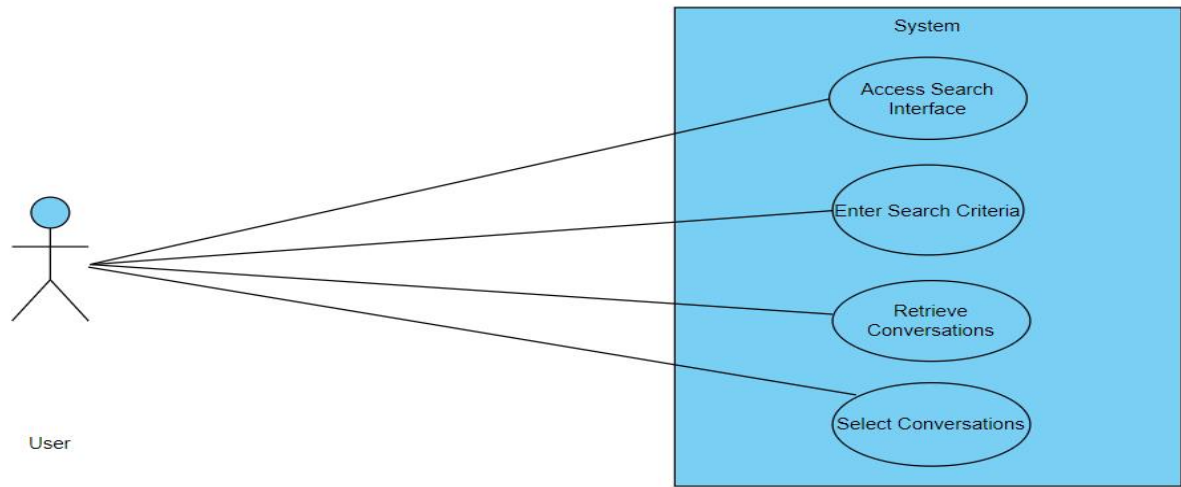
USE CASE 3:



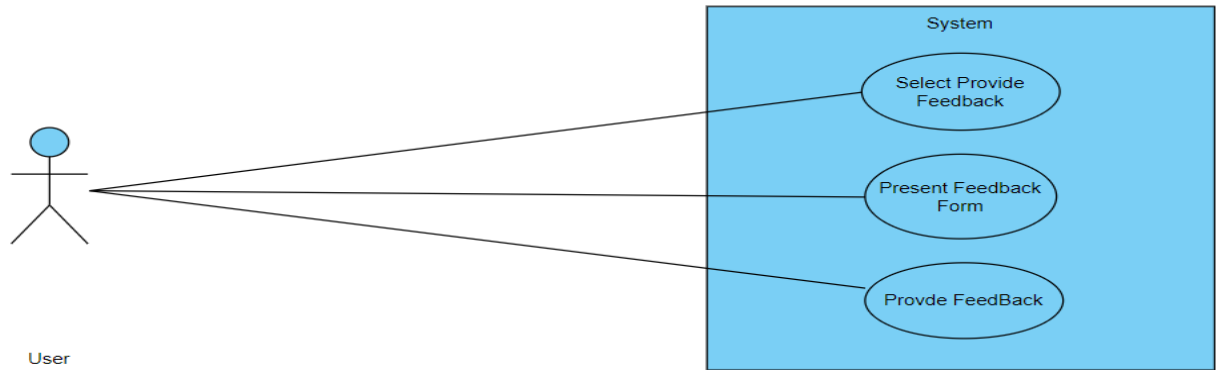
Use Case ID	UC_003	
Use Case Name	Update Profile	
Description	This use case involves the process of updating user profile information.	
Primary Actor	User	
Secondary Actor	None	
Pre-Condition	The user must be logged into the system.	
Post-Condition	The user's profile information is successfully updated.	
Basic Workflow	Actor Action	System Action
	<ul style="list-style-type: none">➤ The user navigates to the profile settings.➤ The user modifies the desired profile details (e.g., name, email, or password).➤ The user submits the updated information.	<ul style="list-style-type: none">➤ The system displays the user's current profile information.➤ The system validates the updated information.➤ The system processes the update request and reflects the changes in the user's profile.
Alternate Flow	<p>If the entered information is incomplete or fails validation:</p> <ul style="list-style-type: none">➤ The system notifies the user of the validation error.➤ The user corrects the information and resubmits the form.➤ Steps 4 to 6 are repeated.	

USE CASE 4:

Use Case ID	UC_004	
Use Case Name	Perform Sign Language Translation	
Description	This use case involves the system translating live Sign Language gestures into natural language in real-time.	
Primary Actor	User (Sign Language User)	
Secondary Actor	None	
Pre-Condition	The user must be logged into the system, and the device must have access to a camera.	
Post-Condition	The system successfully transcribes the Sign Language gestures into natural language.	
Basic Workflow	Actor Action	System Action
	<ul style="list-style-type: none"> ➤ The user selects the "Perform Translation" option. ➤ The user performs Sign Language gestures in front of the camera. 	<ul style="list-style-type: none"> ➤ The system activates the camera for live translation. ➤ The system processes the live video feed, extracting key features. ➤ The system applies Natural Language Processing (NLP) models to translate gestures into text.
Alternate Flow	If the system encounters difficulty in recognizing gestures: <ul style="list-style-type: none"> ➤ The system may prompt the user to adjust lighting or perform clearer gestures. ➤ Steps 3 to 5 are repeated until successful translation. 	

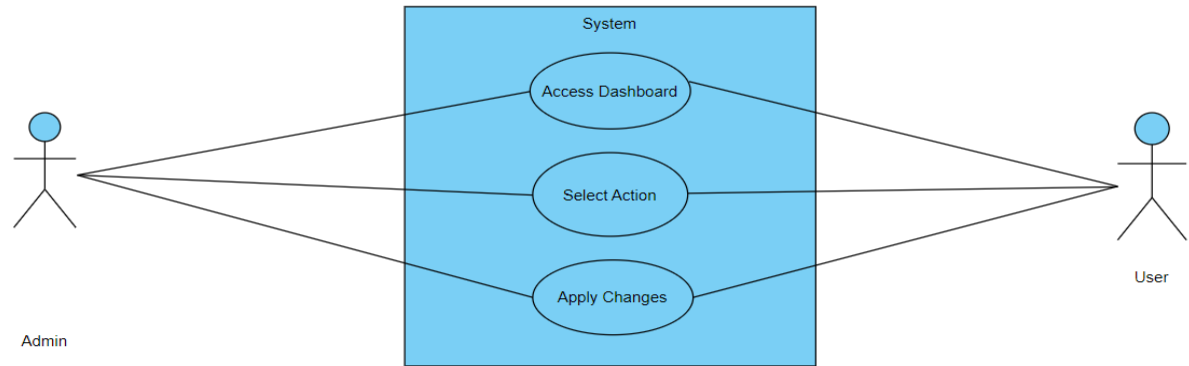
USE CASE 5:

Use Case ID	UC_005	
Use Case Name	Search Translated Conversations	
Description	This use case involves the user searching for and accessing previously translated Sign Language conversations.	
Primary Actor	User	
Secondary Actor	None	
Pre-Condition	The user must be logged into the system..	
Post-Condition	The user successfully retrieves and views previously translated conversations.	
Basic Workflow	Actor Action	System Action
	<ul style="list-style-type: none"> ➤ The user accesses the "Search Conversations" feature. ➤ The user enters search criteria, such as keywords, date, or participants. ➤ The user selects a conversation from the search results. 	<ul style="list-style-type: none"> ➤ The system displays a search interface. ➤ The system searches the database for relevant translated conversations. ➤ The system presents the selected conversation in natural language text.
Alternate Flow	If there are no matching conversations: <ul style="list-style-type: none"> ➤ The system notifies the user of no results. ➤ The user may refine the search criteria. ➤ Steps 4 to 6 are repeated. 	

USE CASE 6:

Use Case ID	UC_006	
Use Case Name	Provide Feedback on Translations	
Description	This use case involves users providing feedback on the accuracy and quality of Sign Language translations.	
Primary Actor	User	
Secondary Actor	None	
Pre-Condition	The user must be logged into the system and have accessed a translated conversation.	
Post-Condition	User feedback is recorded and may be used for system improvement.	
Basic Workflow	Actor Action	System Action
	<ul style="list-style-type: none"> ➤ While viewing a translated conversation, the user selects the "Provide Feedback" option. ➤ The user provides feedback on the accuracy and clarity of the translation. 	<ul style="list-style-type: none"> ➤ The system presents a feedback form. ➤ The system records the user's feedback.
Alternate Flow	If the user chooses not to provide feedback: <ul style="list-style-type: none"> ➤ The system proceeds without collecting feedback. ➤ The user continues with their interaction. 	

USE CASE 7:



Use Case ID	UC_007	
Use Case Name	System Administrator Management	
Description	This use case involves the actions performed by a system administrator for managing user accounts and system configurations.	
Primary Actor	System Administrator	
Secondary Actor	User	
Pre-Condition	The system administrator must be logged into the system..	
Post-Condition	Changes to user accounts and system configurations are successfully applied.	
Basic Workflow	Actor Action	System Action
	<ul style="list-style-type: none">➤ The system administrator accesses the administrator dashboard.➤ The system administrator selects an action, such as managing user accounts or configuring system settings.	<ul style="list-style-type: none">➤ The system displays the administrator tools and options.➤ The system processes the administrator's request and applies changes.
Alternate Flow	If an error occurs during the administrator's action: <ul style="list-style-type: none">➤ The system notifies the administrator of the error.➤ The administrator takes corrective actions.➤ Steps 3 to 4 are repeated.	

CHAPTER 4

DESIGN

4.1. Architecture Diagram

This architecture diagram outlines the components of a Sign Language Transcription System:

1. **User:** Represents individuals interacting with the system, providing user data and submitting sign language videos.
2. **SignLanguageVideo:** Represents videos of sign language submitted by users.
3. **UserDatabase:** Stores user information and sign language videos.
4. **UserInterface:** Web interface for users to interact with the system, displaying user information, and submitting sign language videos.
5. **MessageQueue:** Manages the queue of sign language videos for processing.
6. **TranslationService:** Utilizes NLP models to process sign language videos and generate text.
7. **CloudServices:** Integrates the message queue and translation service.
8. **Administrator:** Represents administrators with access to system management.
9. **AdministratorInterface:** Web interface for administrators to access the dashboard, select actions, and apply changes.

Key Interactions:

- Users submit sign language videos through the UserInterface.
- User data and videos are stored in the UserDatabase.
- MessageQueue manages the queue of videos for processing.
- TranslationService processes videos using NLP models.
- CloudServices integrate the message queue and translation service.
- Administrators manage the system through the AdministratorInterface.

This architecture facilitates the transcription of sign language videos into text, promoting inclusivity and accessibility.

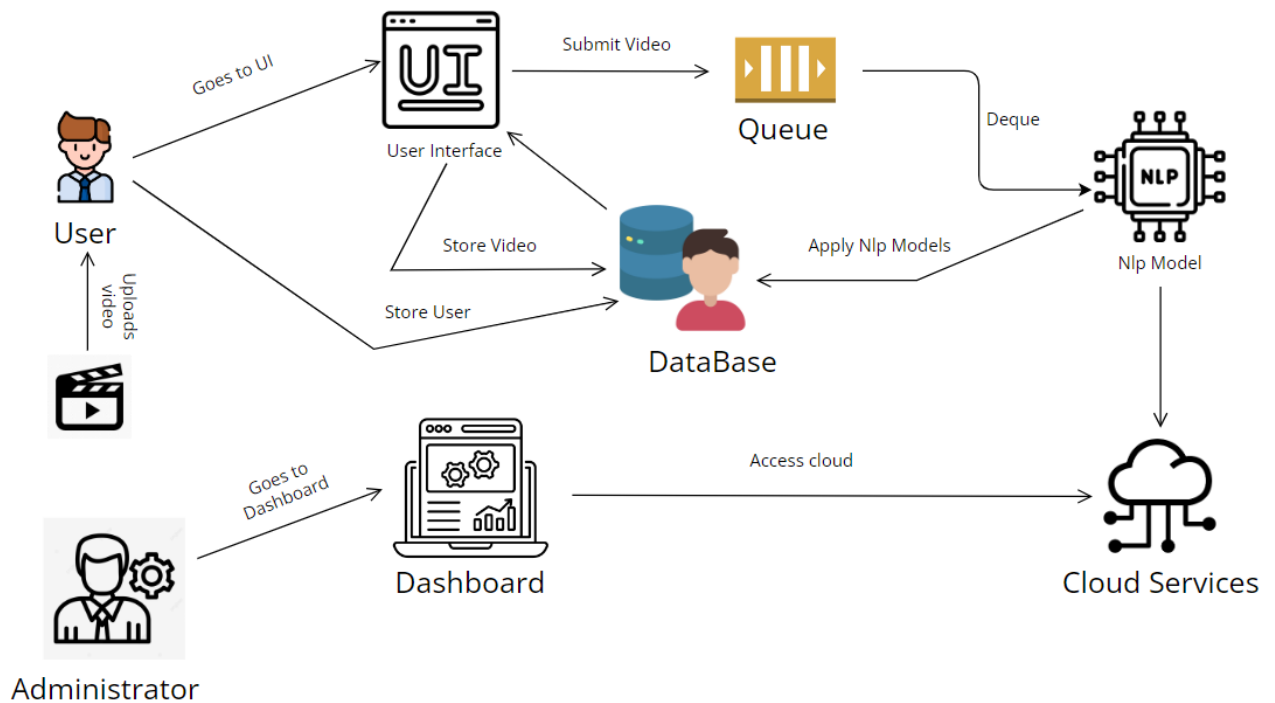


Figure 1 - Architecture Diagram

4.2.ER Diagram

4.2.1. User Entity:

Attributes: **UserId**, **UserName**, **FirstName**, **LastName**, **Email**, **Password**, **RegistrationDate**, **LastLoginDate**, and other user-specific attributes.

Relationships:

A User can submit multiple SignLanguageVideos ("Submits").

A User can provide feedback on multiple Translations ("Provides Feedback").

4.2.2. SignLanguageVideo Entity:

Attributes: **VideoId**, **UserId** (foreign key), **VideoFile**, **Timestamp**, **SubmissionDate**, **Duration**, **Resolution**, and other video-specific attributes.

Relationships:

A User can submit SignLanguageVideos ("Submits" relationship).

Each SignLanguageVideo is associated with a Translation through **VideoId**.

4.2.3. Administrator Entity:

Attributes: **AdminId**, **AdminName**, **Email**, **Password**, **Role**, and other admin-specific attributes.

Relationships:

An Administrator can review multiple Translations ("Reviews" relationship).

4.2.4. Translation Entity:

Attributes: **TranslationId**, **VideoId** (foreign key), **Transcription**, **TranslationDate**, **FeedbackCount**, **AverageFeedbackRating**, **Language**, and other translation-specific attributes.

Relationships:

A Translation is associated with a SignLanguageVideo through **VideoId**.

A Translation can have multiple Feedback entries ("Has" relationship).

4.2.5. Feedback Entity:

Attributes: **FeedbackId**, **UserId** (foreign key), **TranslationId** (foreign key), **Rating**, **Comment**, **FeedbackDate**, and other feedback-specific attributes.

Relationships:

A User can provide feedback on Translations ("Provides Feedback" relationship).

A Translation can have multiple Feedback entries ("Has" relationship).

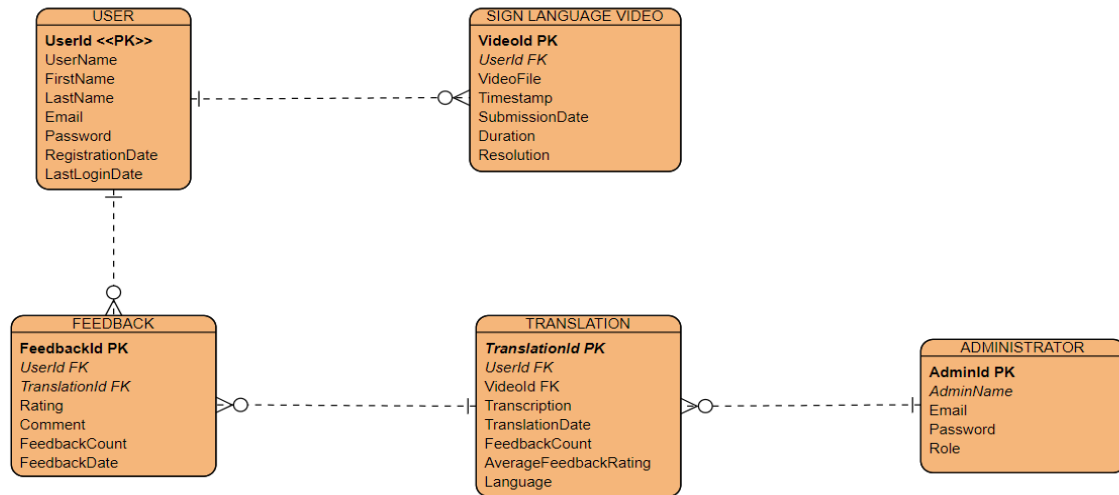


Figure 1 -ER Diagram

Data Dictionary:

User Entity

FEILD	Data Type	Description
UserId (PK)	INT	Unique identifier for a user.
UserName	VARCHAR	User's username.
FirstName	VARCHAR	User's first name.
LastName	VARCHAR	User's last name.
Email	VARCHAR	User's email address.
Password	VARCHAR	User's hashed password.
RegistrationDate	DATE	Date when the user registered in the system.
LastLoginDate	DATE	Date of the user's last login.

SignLanguageVideo Entity

FEILD	Data Type	Description
VideoId (PK)	INT	Unique identifier for a sign language video.
UserId (FK)	INT	Foreign key referencing the User entity.
VideoFile	VARCHAR	File path or identifier for the sign language video.

Timestamp	DATETIME	Timestamp of when the video was submitted.
SubmissionDate	DATE	Date when the video was submitted.
Duration	INT	Duration of the sign language video in seconds.
Resolution	VARCHAR	Resolution of the video (e.g., HD, 4K).

ADMINISTRATOR ENTITY

FEILD	Data Type	Description
AdminId (PK)	INT	Unique identifier for an administrator.
AdminName	VARCHAR	Administrator's name.
Email	VARCHAR	Administrator's email address.
Password	VARCHAR	Administrator's hashed password.
Role	VARCHAR	Role or position of the administrator.

Translation Entity

FEILD	Data Type	Description
TranslationId (PK)	INT	Unique identifier for a translation.
VideoId (FK)	INT	Foreign key referencing the SignLanguageVideo entity.
Transcription	TEXT	Transcription of the sign language video.
TranslationDate	DATE	Date when the translation was generated.
FeedbackCount	INT	Number of feedback entries for the translation.
AverageFeedbackRating	FLOAT	Average rating from user feedback for the translation.
Language	VARCHAR	Language of the transcription/translation (e.g., English).

Feedback Entity

FEILD	Data Type	Description
FeedbackId (PK)	INT	Unique identifier for a feedback entry.
UserId (FK)	INT	Foreign key referencing the User entity.
TranslationId (FK)	INT	Foreign key referencing the Translation entity.
Rating	INT	User's rating for the translation (e.g., 1 to 5).
Comment	TEXT	User's comments or feedback on the translation.
FeedbackDate	DATE	Date when the feedback was submitted.

4.3.Data Flow Diagram**User (External Agent):**

Represents the external user interacting with the Sign Language Transcription System.

User Interface Process:**Responsibilities:**

Handles user interactions and requests.

Functions:

Accepts user input for video submission, feedback, and other system interactions.

Provides a user-friendly interface for seamless interaction.

Output: Passes user requests and data to the Sign Language Video Processing and Feedback Handling System.

Sign Language Video Processing Process:**Responsibilities:**

Processes sign language videos submitted by users.

Functions:

Extracts key features from sign language videos, including hand movements, facial expressions, and body language.

Prepares the video data for further analysis and translation.

Output: Sends processed video data to the Translation and Transcription Engine.

Translation and Transcription Engine Process:**Responsibilities:**

Utilizes advanced Natural Language Processing (NLP) models to transcribe sign language into natural language text.

Functions:

Applies NLP techniques to understand sign language gestures and expressions.

Translates sign language features into coherent and understandable natural language text.

Output: Provides the transcribed text for further use and analysis.

Feedback Handling System Process:**Responsibilities:**

Manages user feedback for system improvement.

Functions:

Collects and processes user feedback on transcriptions and system performance.

Analyzes feedback to identify areas for improvement.

Output: Implements system enhancements based on user feedback.

In summary, this DFD outlines the main components and their interactions in the Sign Language Transcription System. Users interact with the User Interface to submit videos and provide feedback. The system processes videos, applies advanced NLP for transcription, and manages feedback to enhance system performance. The directional arrows represent the flow of data and control between these components, providing a visual representation of the system's functionality.

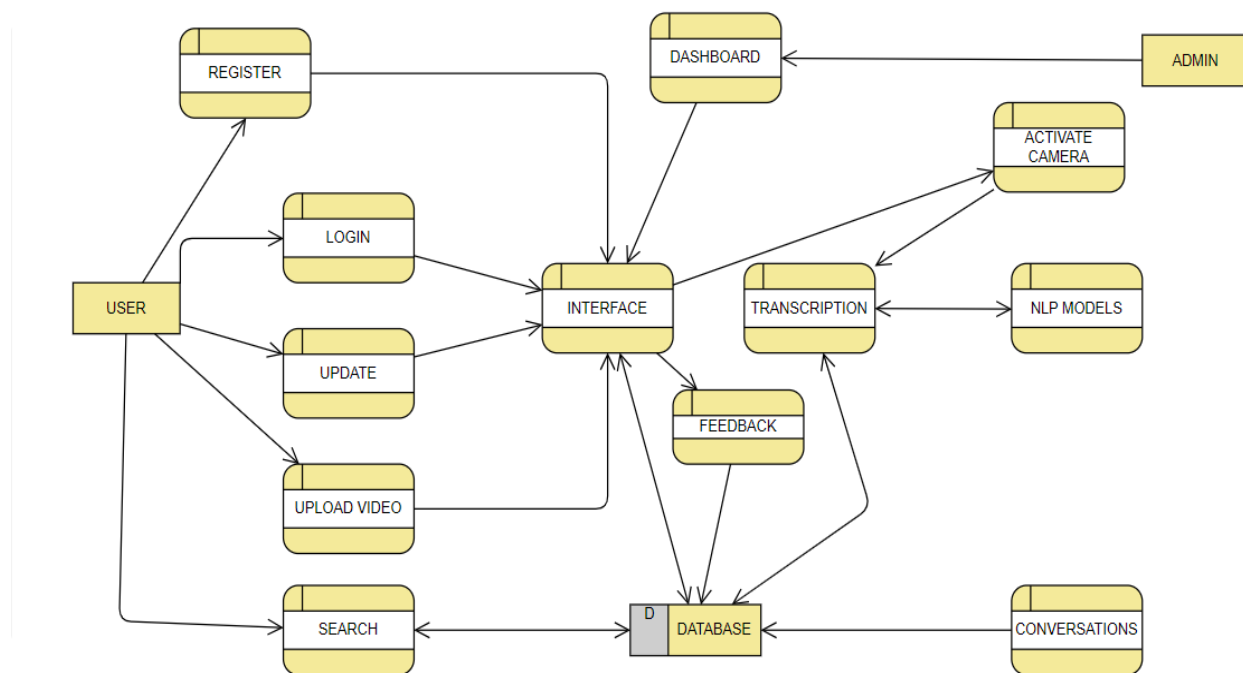


Figure 2- DFD

4.4. Class Diagram

The class diagram represents the key components and relationships in a Sign Language Transcription System:

1. User, Administrator, Signer, Translator:

- Users interact with the system. Signers and Translators are specialized users with additional attributes.

2. SignLanguageVideo:

- Represents videos submitted by users for sign language transcription.

3. Translation:

- Holds information about the transcriptions of sign language videos.

4. **Feedback:**

- Captures user feedback on transcriptions.

5. **SignLanguageTranscriptionSystem:**

- Central system orchestrating **UserInterface**, **VideoProcessing**, and **FeedbackHandling**.

6. **UserInterface:**

- Allows users to submit videos and feedback.

7. **VideoProcessing:**

- Processes sign language videos, extracting features for transcription.

8. **TranslationAndTranscriptionEngine:**

- Transcribes sign language features into written text.

9. **FeedbackHandling:**

- Analyzes user feedback and suggests system enhancements.

10. **SystemEnhancement:**

- Represents enhancements based on feedback.

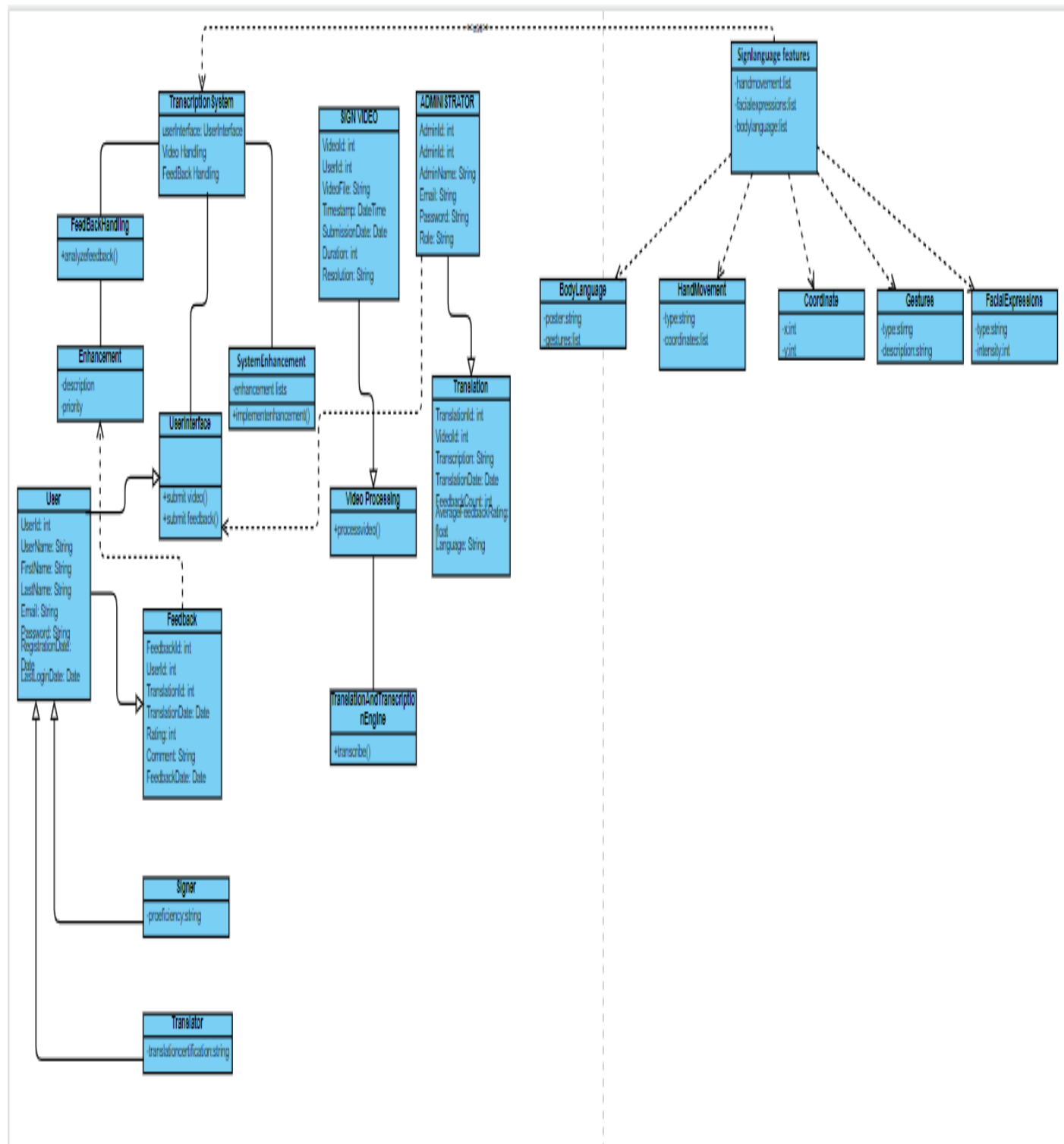
11. **Enhancement:**

- Describes specific system improvements.

12. **SignLanguageFeatures, HandMovement, FacialExpression, BodyLanguage, Coordinate, Gesture:**

- Classes related to the features extracted from sign language videos.

The diagram uses associations to show how these classes interact, and it includes inheritance to depict specialized user roles. The overall structure provides a foundation for building a comprehensive Sign Language Transcription System.



4.5. Activity Diagrams

1. Register

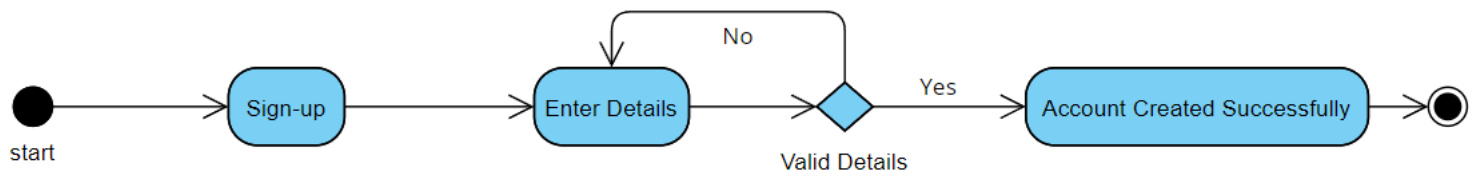


Figure 2- Requirement FR_01

2. Login

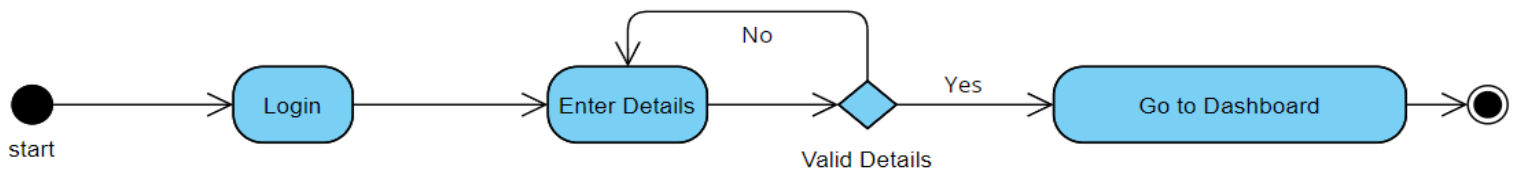


Figure 4- Requirement FR_02

3. Video Transcription

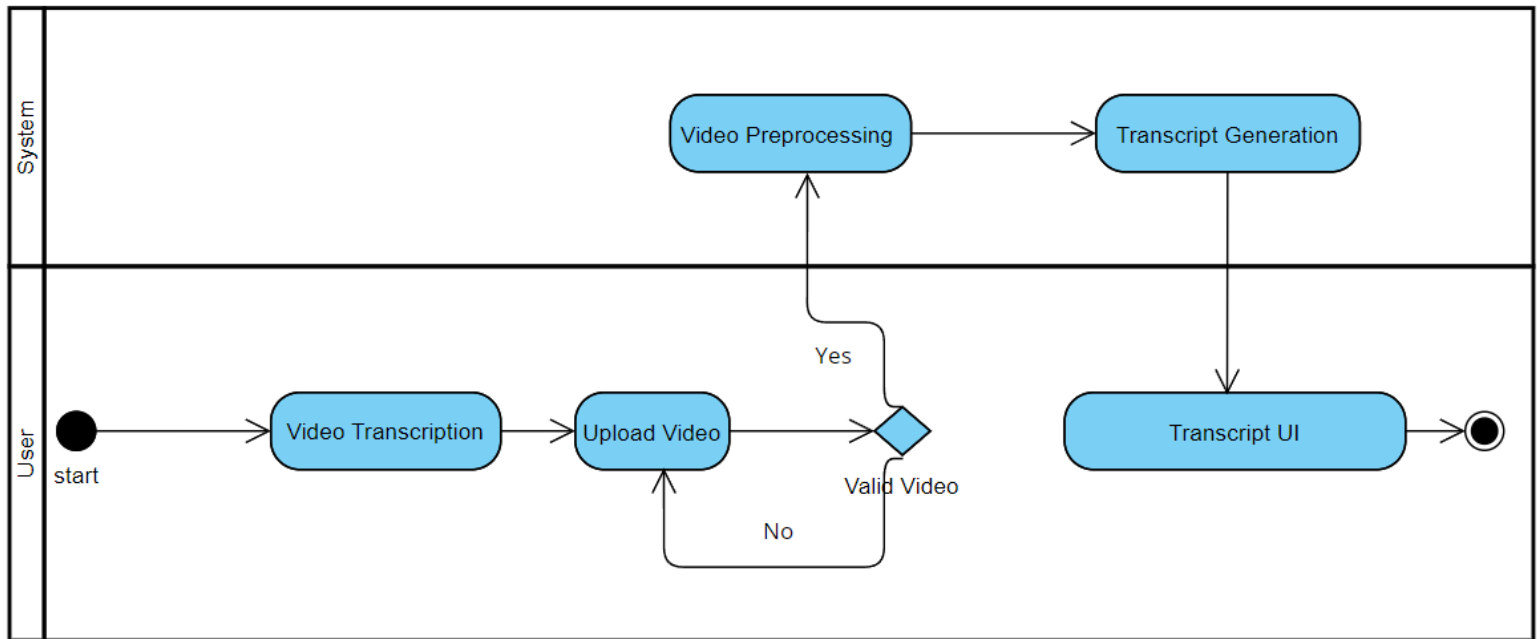


Figure 5- Requirement FR_03

4. Real Time Transcription

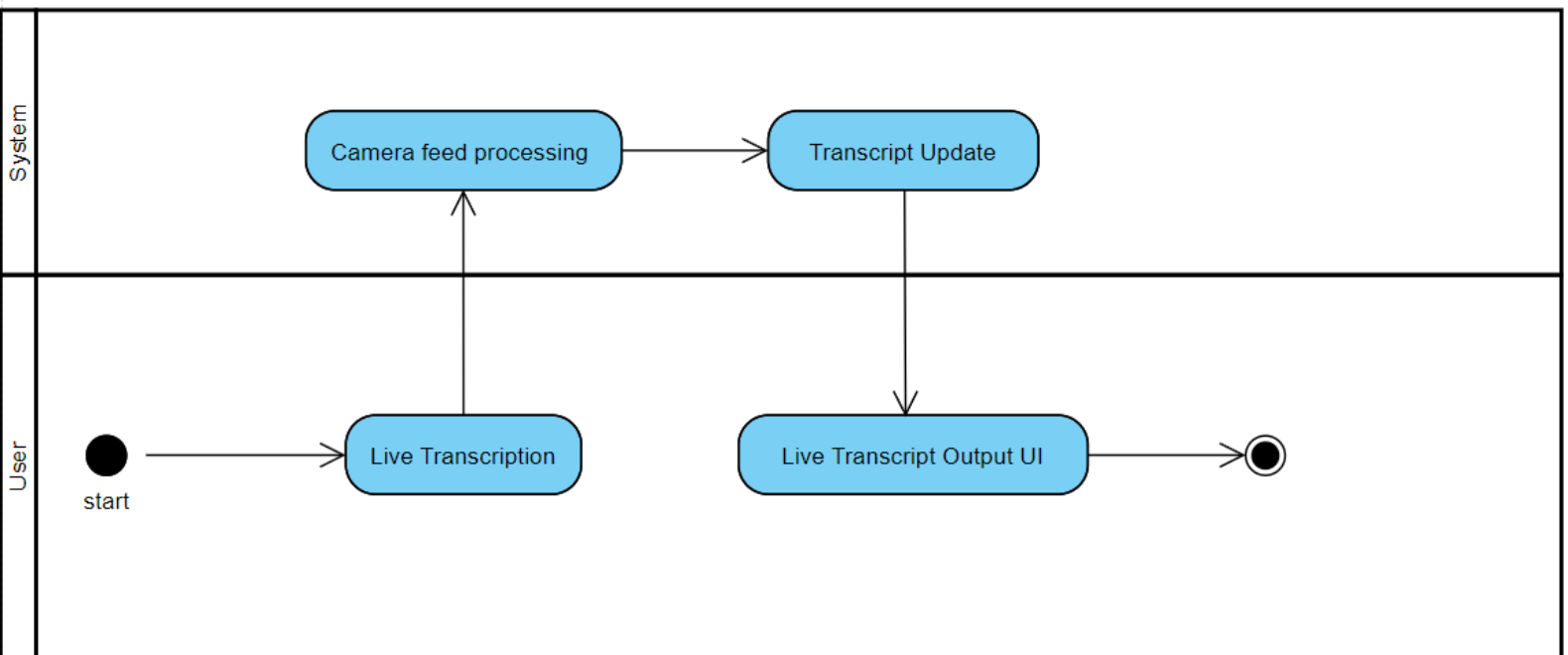


Figure 6- Requirement FR_04

5.Track History

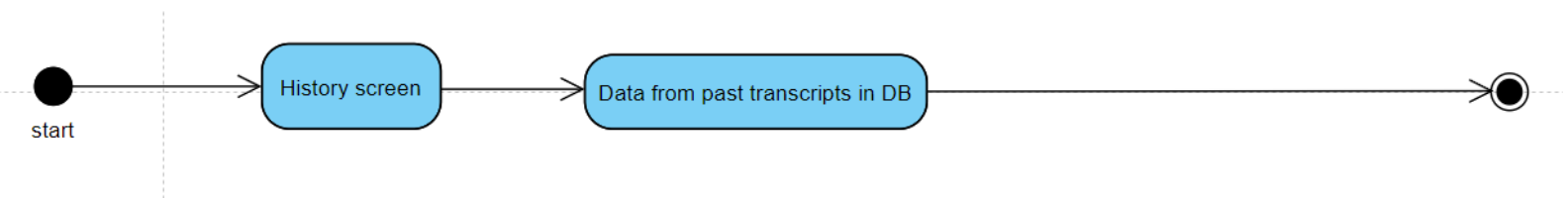


Figure 7- Requirement FR_05

4.6.Sequence Diagram

1. Sign-up

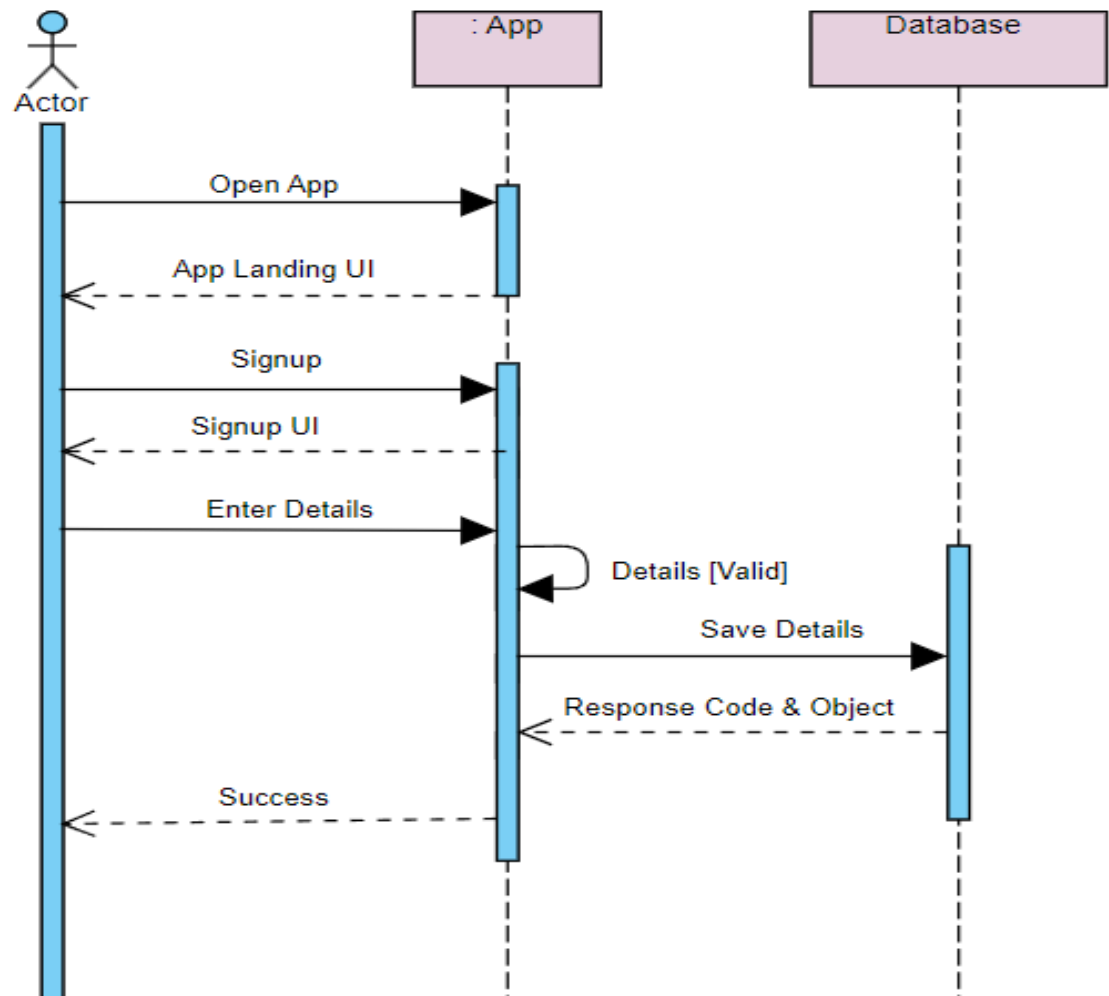
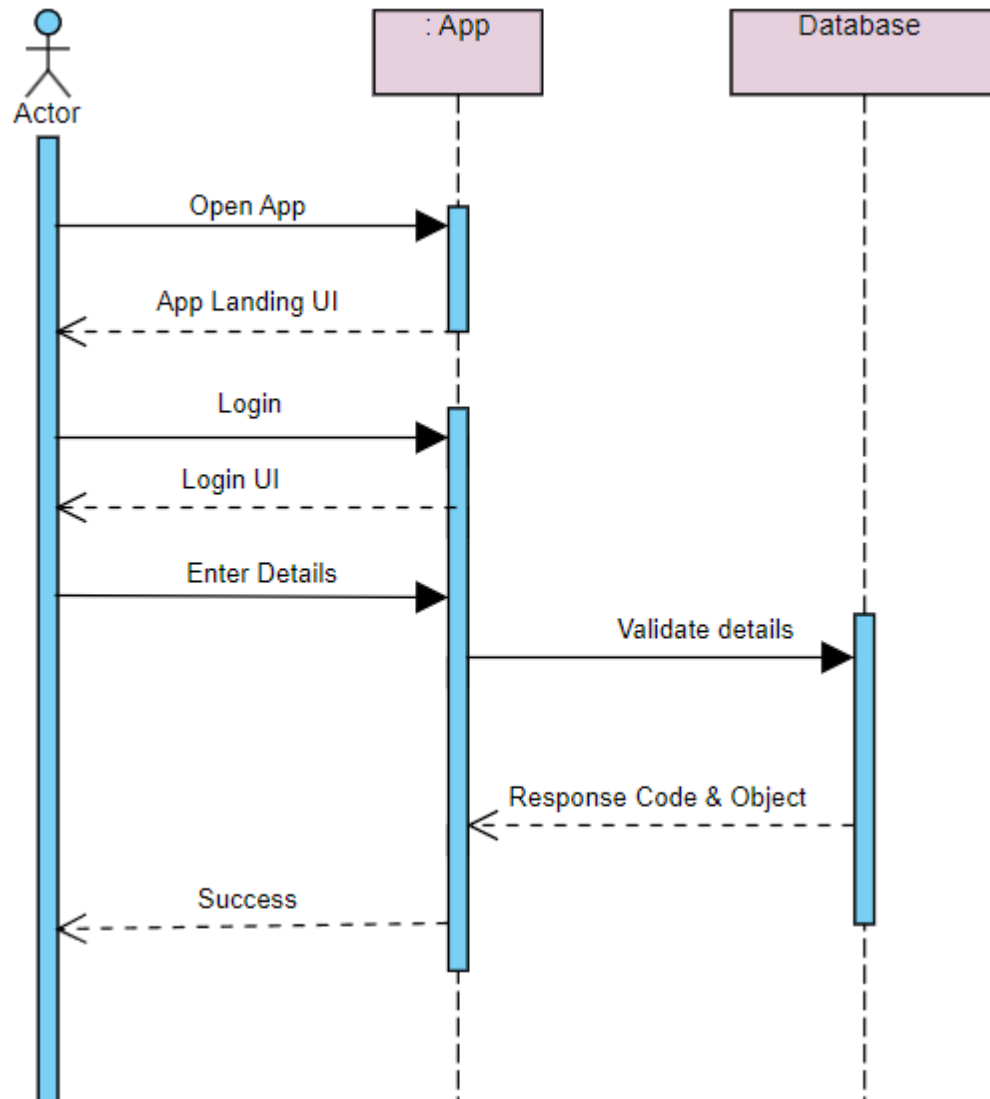


Figure 8 - Requirement FR_01

2. Login

*Figure 8- Requirement FR_02*

3. Video Transcription

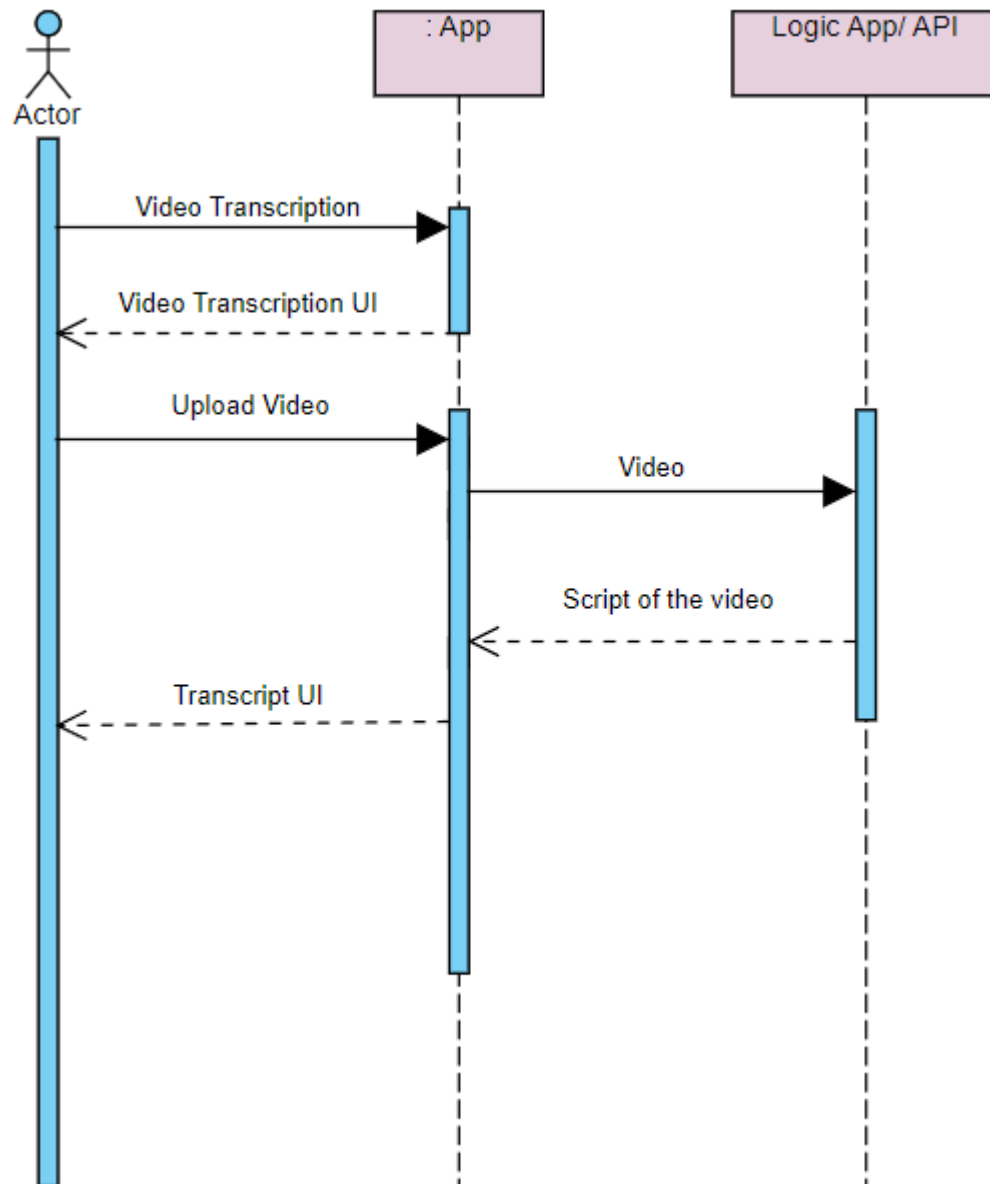


Figure 9 - Requirement FR_O3

4. Real-Time Transcription

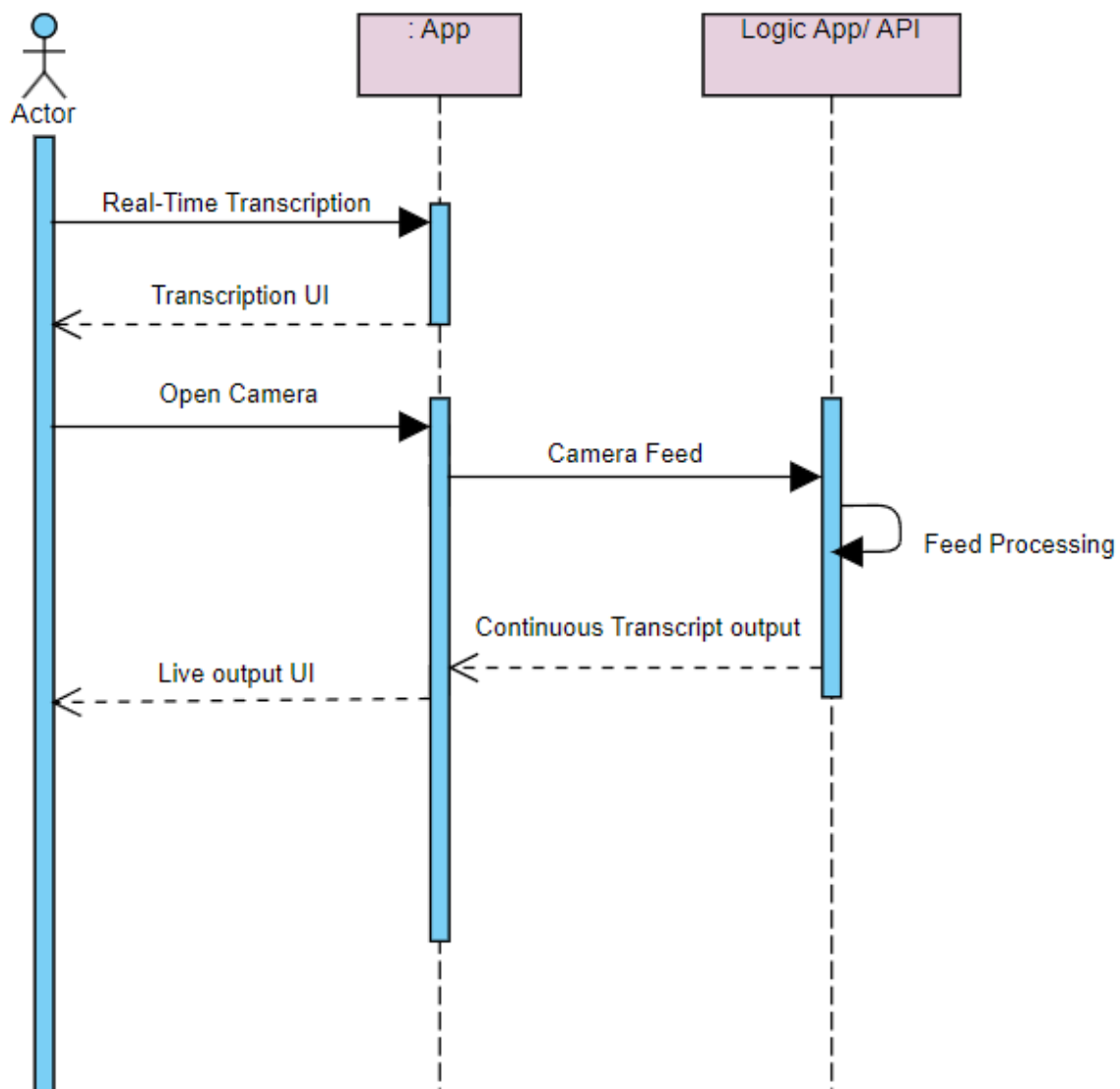


Figure 9 - Requirement FR_O4

5. Track History

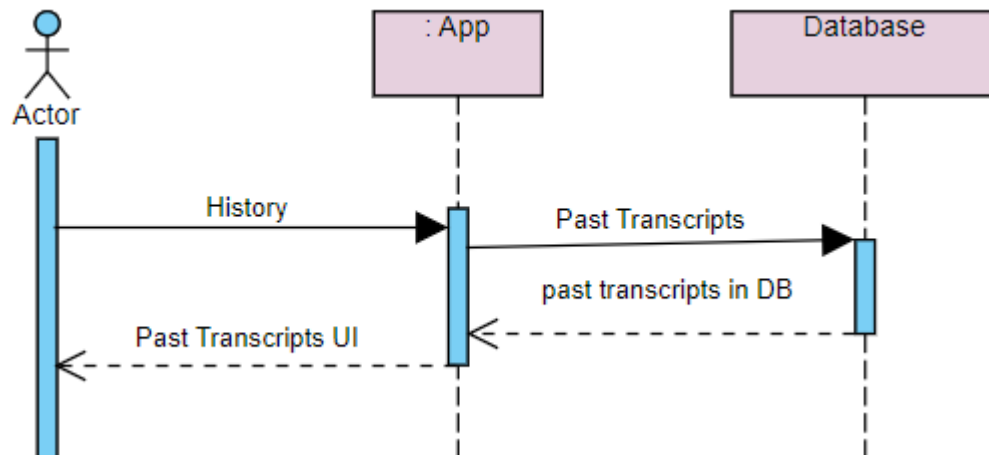


Figure 9 - Requirement FR_O5

4.6 Collaboration Diagram

1. Register

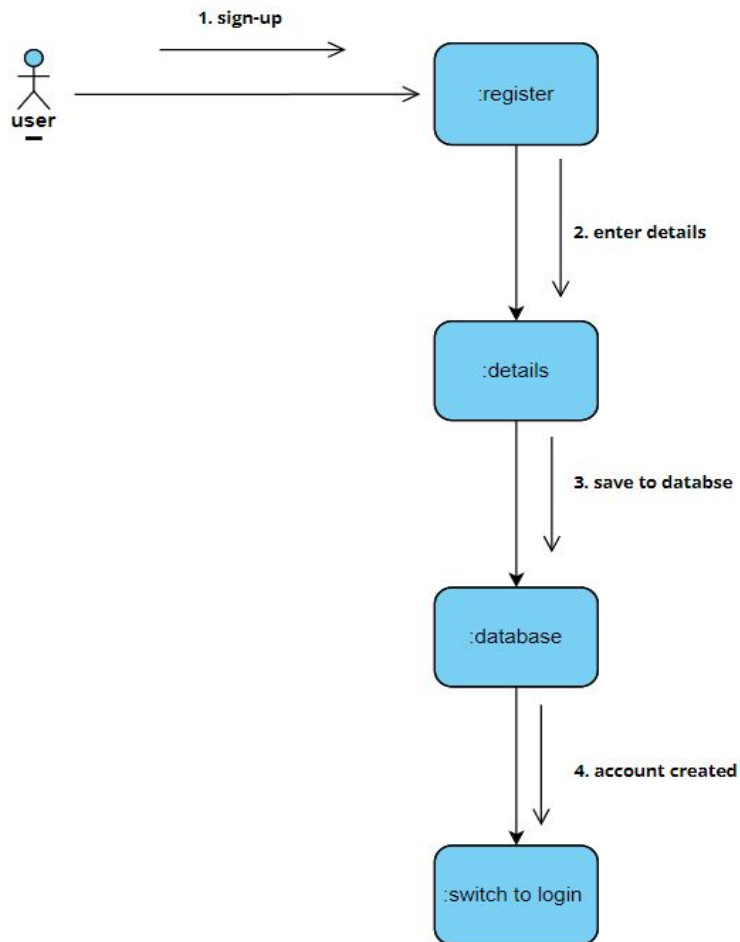


Figure 11- Requirement FR_01

2. Login

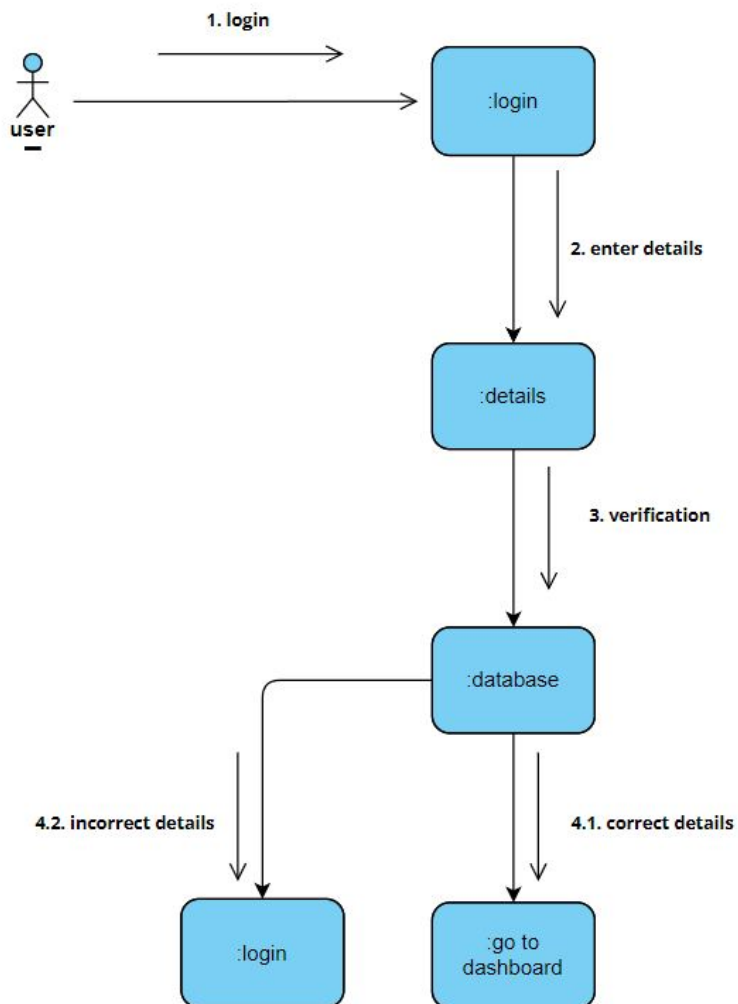


Figure 12- Requirement FR_02

3. Video Transcription

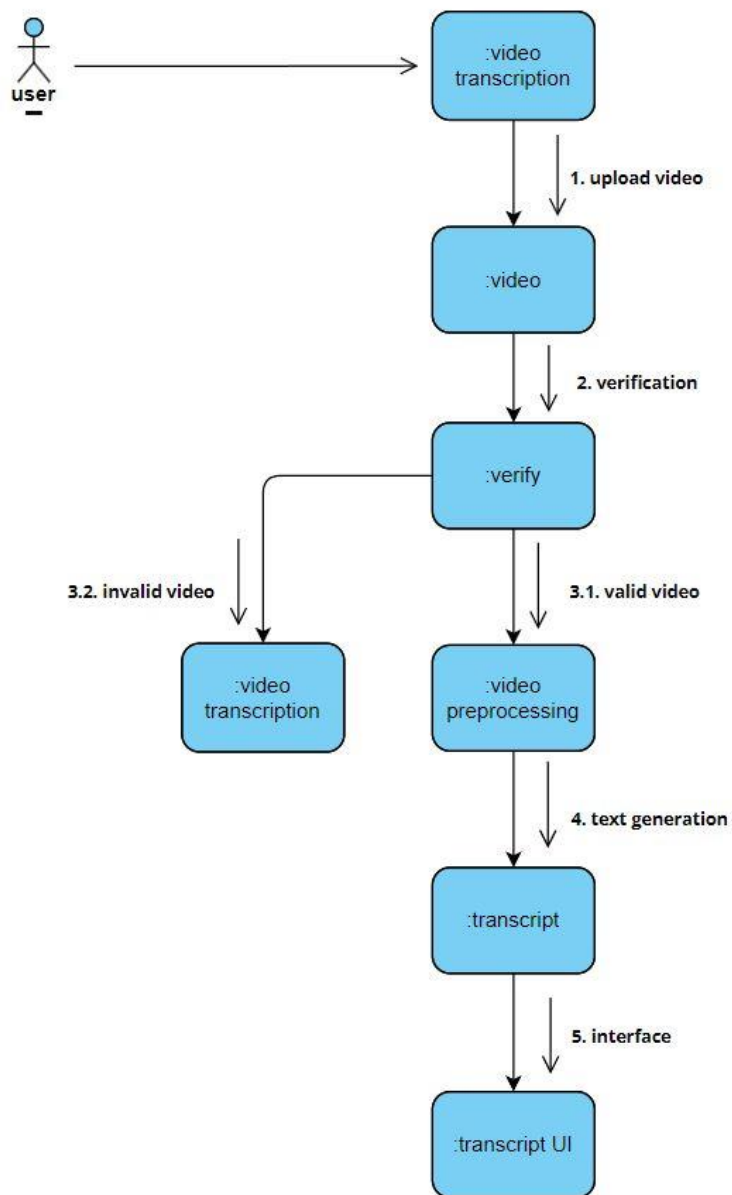


Figure 13- Requirement FR_03

4. Real-Time Transcription

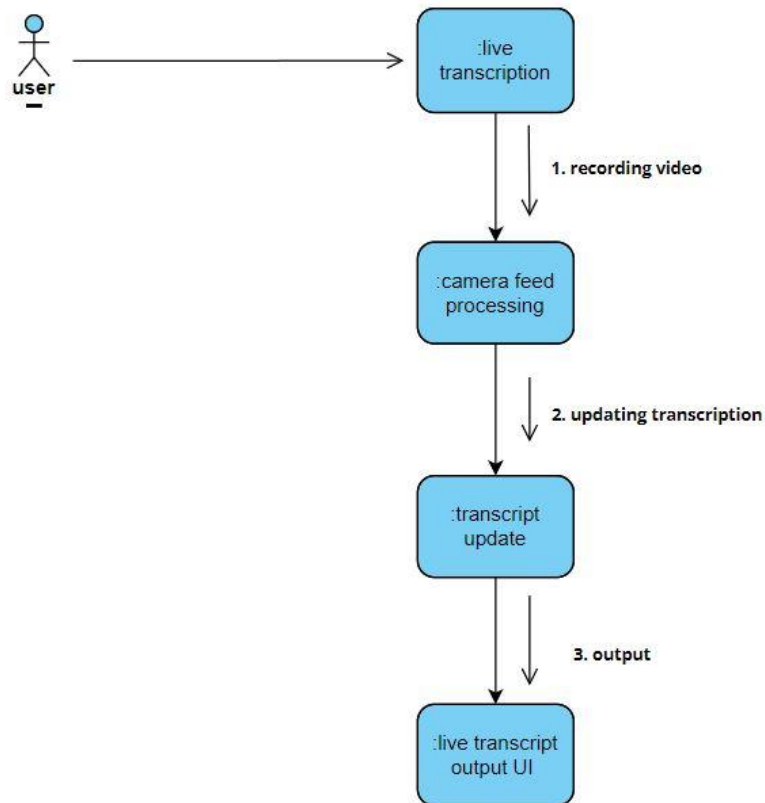


Figure 14- Requirement FR_04

5. Track History

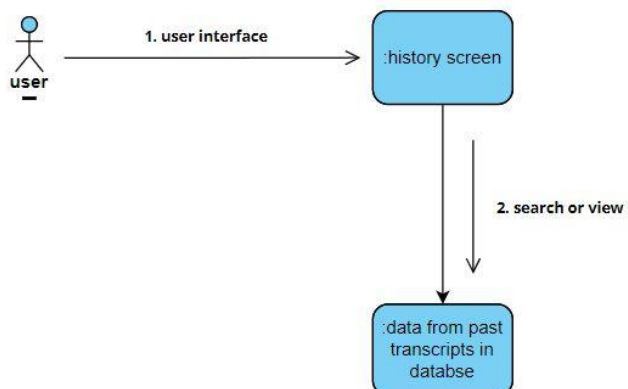


Figure 15- Requirement FR_05

4.7.State Transition Diagram

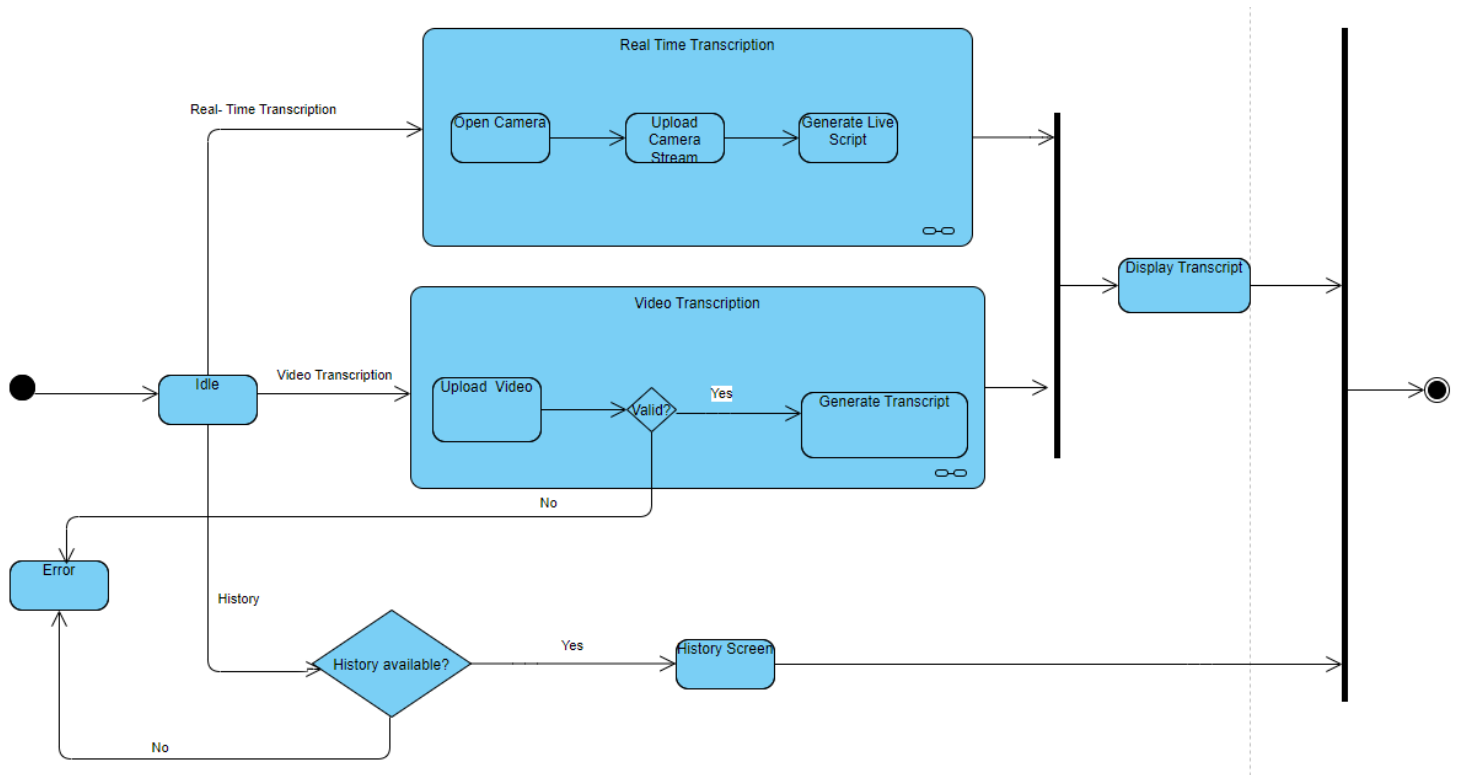


Figure 10 - State Diagram

4.8.Component Diagram

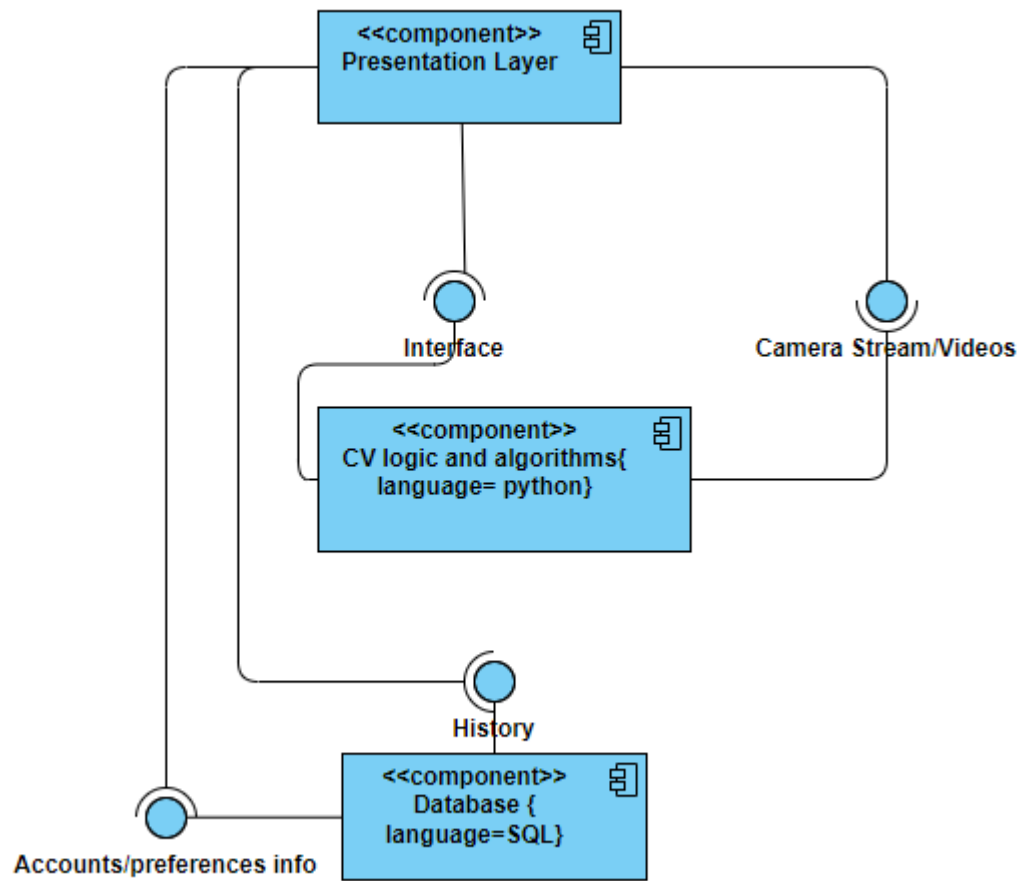


Figure 11- Component Diagram

4.9. Deployment Diagram

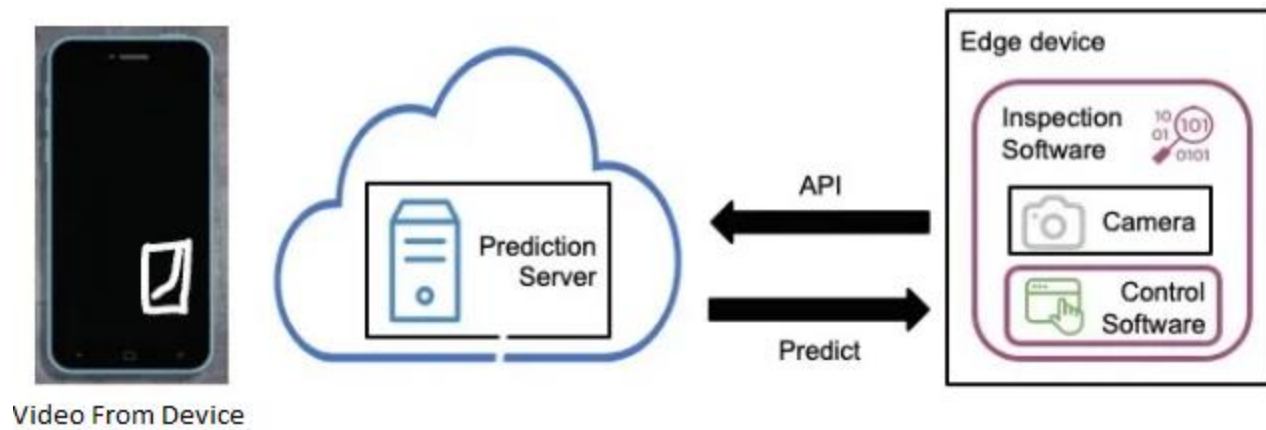


Figure 12 - Deployment Diagram

