CHAPTER 7

SYSTEM DESIGN

- Class Diagram
- Use Case Diagram
- Sequence Diagram
- Activity Diagram
- Data Flow Diagram

7.1   CLASS DIAGRAM

The **Class Diagram** represents the static structure of the system, illustrating the relationships between different objects within the application. It highlights the attributes, operations, and associations among different classes involved in the project. For the SignSpeak project, the Class Diagram models essential components like the Camera, Image Processing, GUI, and ML Model, detailing how they interact to recognize sign language gestures and convert them to text. Each class has a distinct role in capturing, processing, and displaying sign language signs.

	Class
Class Name Attributes Operations	Class
	Generalization

Table 7.1.1 Class diagram symbols

Camera

+bool isActive

+void startCamera()

+void stopCamera()

+void captureFrame()

Captures

GestureRecognition

+void detectHandGesture()

+void recognizeGesture()

+void displayDetectedGesture()

Recognizes

SignImage

+String imageName

+void displaySignImage()

+void loadSignImages()

TextConversion

+String recognizedText

+void convertGestureToText()

+void displayText()

Theme

+String currentTheme

+void toggleTheme()

Displays

Displays

Toggles

GUI

+void displaySignToText()

+void toggleDarkLightMode()

+void showSignImagesForLearning()

Fig. No. 7.1.1 Class diagram

7.2   Use Case Diagram

The Use Case Diagram shows the interaction between users and the system. It identifies all the possible actions a user can perform and how the system responds to each action. In SignSpeak, users interact with the system by starting the camera, processing the sign images, viewing the result (converted text), and learning the sign images. The Use Case Diagram demonstrates the system's functionality from a user's perspective, highlighting features such as sign language recognition, feedback, and learning modes.

	Actor
	Use-case
	System
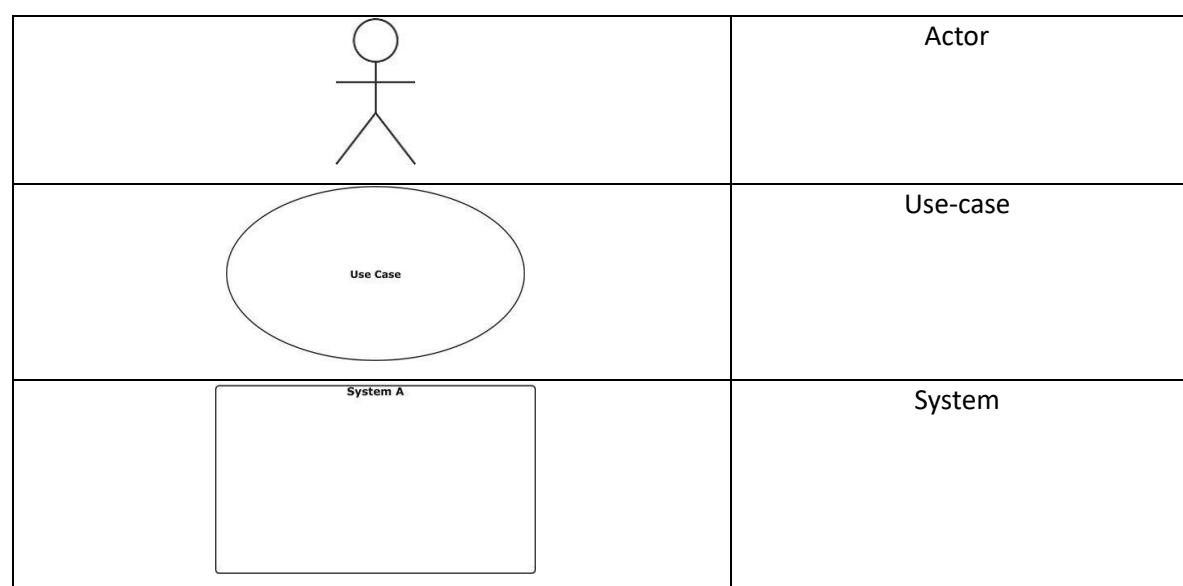
Fig. No. 7.2.1 Use-case diagram
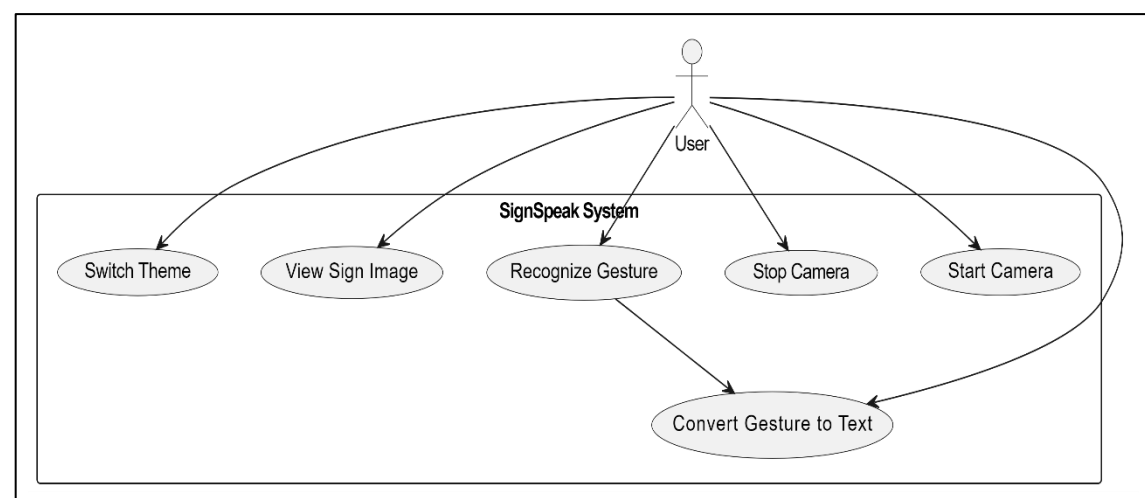
Fig. No. 7.2.1 Use-case diagram

7.3   SEQUENCE DIAGRAM

The **Sequence Diagram** provides a dynamic view of the system, depicting the sequence of interactions between objects over time. For the SignSpeak project, this diagram focuses on how different components of the system (camera, image processing, ML model, and GUI) work together. It shows the step-by-step process involved in capturing an image, processing it, running recognition algorithms, and returning the recognized sign as text to the user interface.
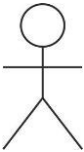
	Object
	Message
	Actor

Table 7.3.1 Sequence diagram symbols
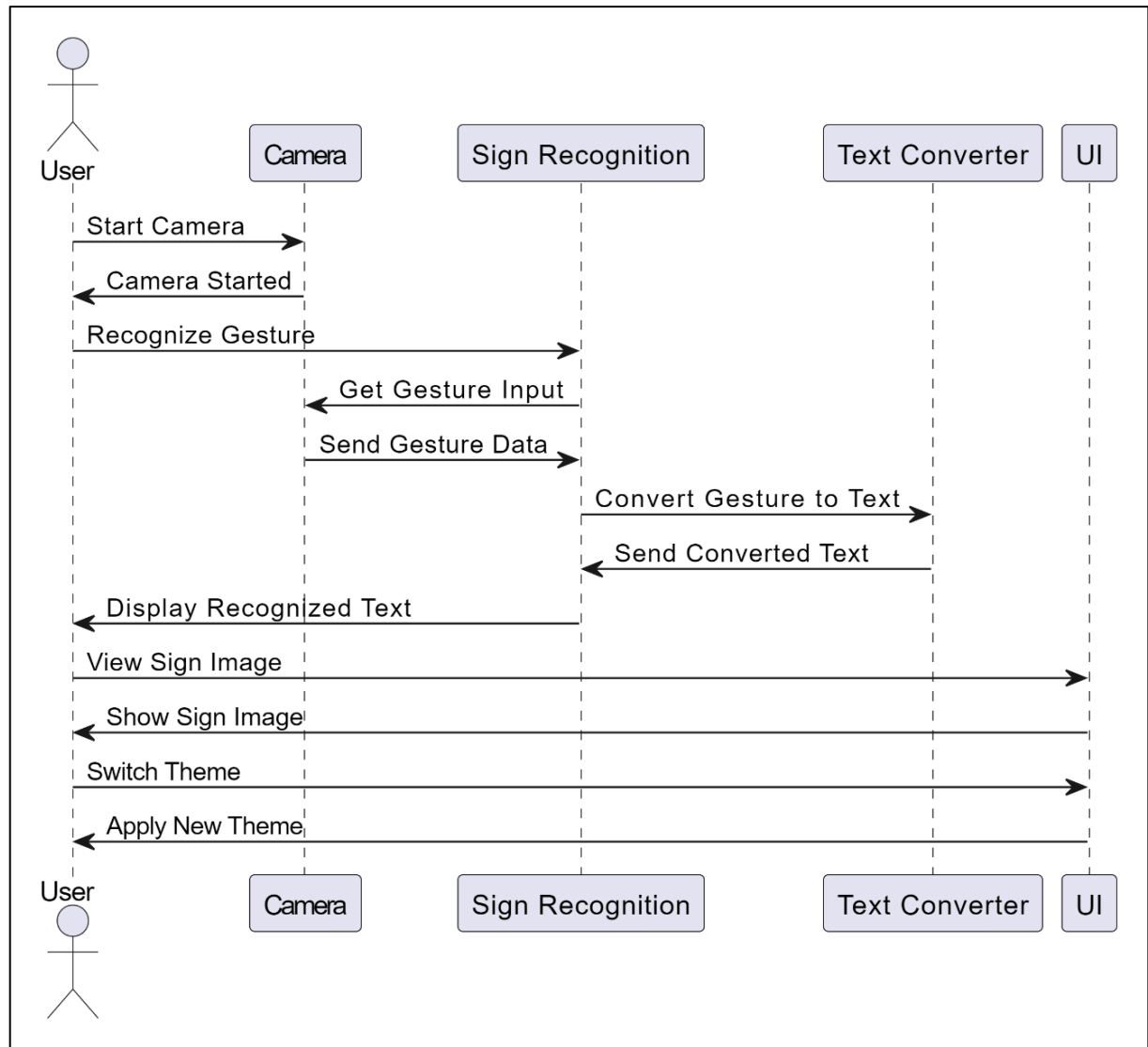
Fig. No. 7.3.1 Sequence diagram

7.4   ACTIVITY DIAGRAM

The Activity Diagram outlines the flow of activities or tasks within the system. It is used to model the workflow of the SignSpeak project, showing the different actions a user can perform, from starting the camera to displaying the recognized text. It visualizes conditional flows, user choices (e.g., choosing between dark and light themes), and the system's reactions to these inputs. This diagram is beneficial for understanding the overall flow and process from the user's perspective.
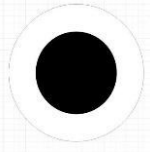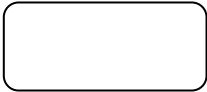
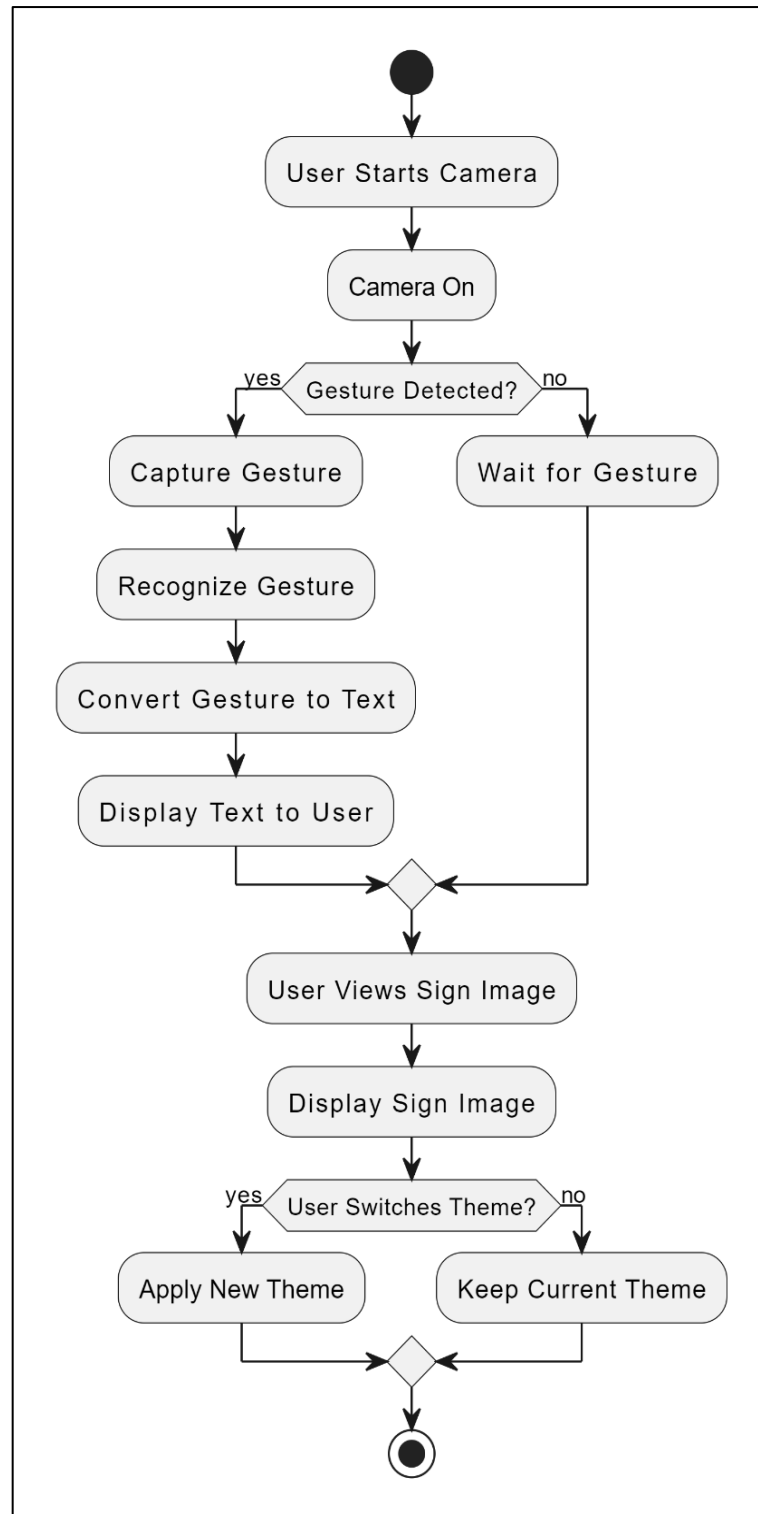|	
●	Initial node
◉	End symbol/node
⟶	Action/Control Flow
▢	Action/Activity

Table 7.4.1 Class diagram symbols

Fig. No. 7.4.1 Activity diagram

7.5   DATA FLOW DIAGRAM

The **Data Flow Diagram** (DFD) provides an overview of how data moves through the system. It represents how input (such as images from the camera) is processed, analyzed, and transformed into output (recognized text). In the context of SignSpeak, the DFD will illustrate how raw data (sign images) is passed through various stages—preprocessing, recognition using machine learning, and displaying text results. This helps in understanding the flow of information and the system's structure in terms of data handling.
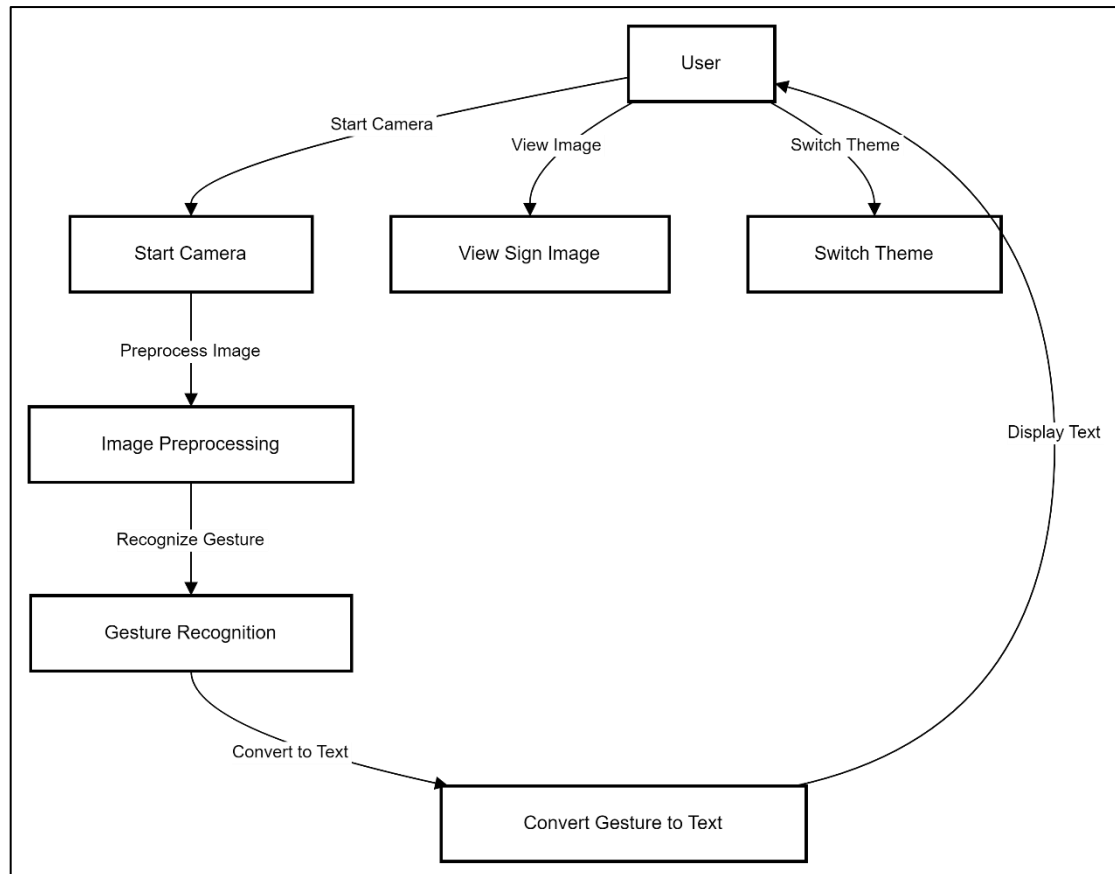
	Entity
	Data flow

Table 7.5.1 Data flow diagram symbols

Fig. No. 7.5.1 Data Flow Diagram