LOW LEVEL DESIGN AND IMPLEMENTATION DOCUMENT

<Indian Sign Language Translation>

UE18CS390B – Capstone Project Phase – 2

Submitted by:

Akhil Chaduvula	**PES1201800282**
Alluri Nayan Varma	**PES1201800208**
Neeli Krishna Dheeraj	**PES1201800182**

Under the guidance of

Dr. Shylaja SS
Professor & Chairperson
PES University

August - December 2021

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

PES UNIVERSITY

LOW LEVEL DESIGN AND IMPLEMENTATION DOCUMENT

TABLE OF CONTENTS

1. Introduction

1.1. Overview

This is a Low Level Design and Implementation Document for the Project "Indian Sign Language Translation". This is documented as a requirement for Review - 2 as a part of our Capstone Project Phase - 2. This document is based on the requirements specified in the Project Requirement Specifications document. This document would help the reader understand the Design Considerations (Design Constraints, Assumptions and Dependencies) and certain diagrams such as Use Case diagram, master class diagram (also the data members and methods of each class present in the master class diagram), Sequence diagram and Packaging Diagram. This would really help the reader to gain insights into the thought process and the planning that was undergone to develop the application.

The whole idea is to build an Android App, that could capture video of any person signalling in sign language and translate it to english captions that would enable anyone to communicate with any person with speech disability.

1.2. Purpose

The purpose of this project is to create an application that enables any person with or without speech or hearing disabilities to communicate with any other person in our country. Models and Applications have been developed for ASL(American Sign Language), but no Application has been developed to help people communicate in ISL(Indian sign Language). This application is an effort to enable people to use such an application.

1.3. Scope

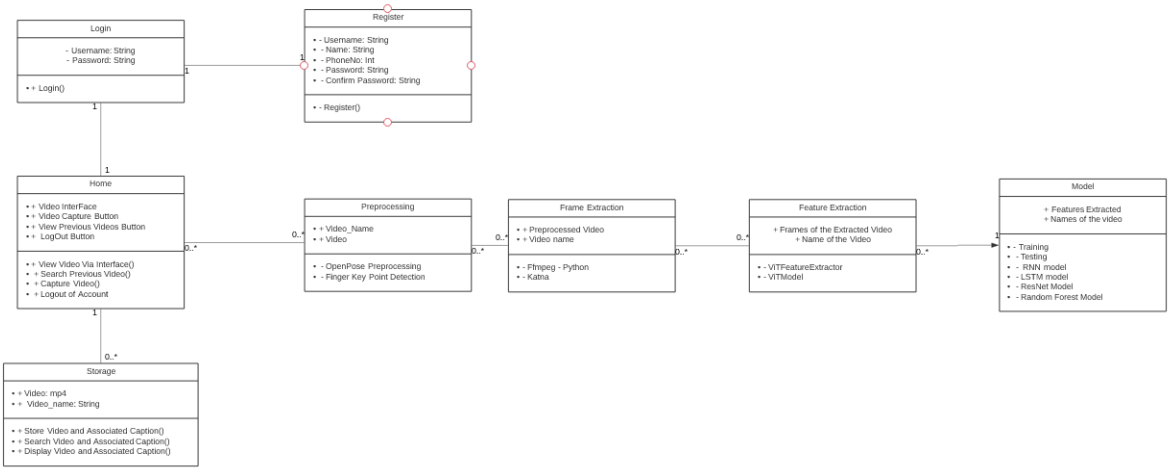
The aim of this project is to build an automatic ISL translator - i.e. given an input video of a specially abled person signing a word, the model should be able to predict which word is being signed. This translator is highly beneficial for the not just the deaf and dumb people, but also for the people trying to communicate with them. The model built should be reasonably robust to variations in camera angle, video quality, distance from the camera to the subject, variations in lighting etc. The goal of this project is to provide(coverage) a means of communication for deaf and dumb people living in India to communicate with anyone in ISL.This Application provides support for only one sign Language (i.e ISL).

2. Design Constraints, Assumptions, and Dependencies

The dependencies made on the dataset include signer diversity and that the information is correct. The constraints include the similar symbols for numerous words. The communication is similar to interprocess communications seen in the various operating systems. The dependency is that the user's phone has sufficient RAM for the application to run. The user is expected to have a working camera for live video recording or real time translation. The interface used is Android. We require the user to have an android phone for the application to run. We expect a performance of 70%. The main issue is that there are similar expressions for the same words. The symbols for FOOD and FIRE are very similar hence they can be similar in nature. The issues of scalability are managed in such a way as the basic packages are installed in the users end system. We also maintain availability through pre-installed repositories installed.The maintainability is done through periodic updates. This constantly improves the performance of the application.

3. Design Description

3.1. Master Class Diagram

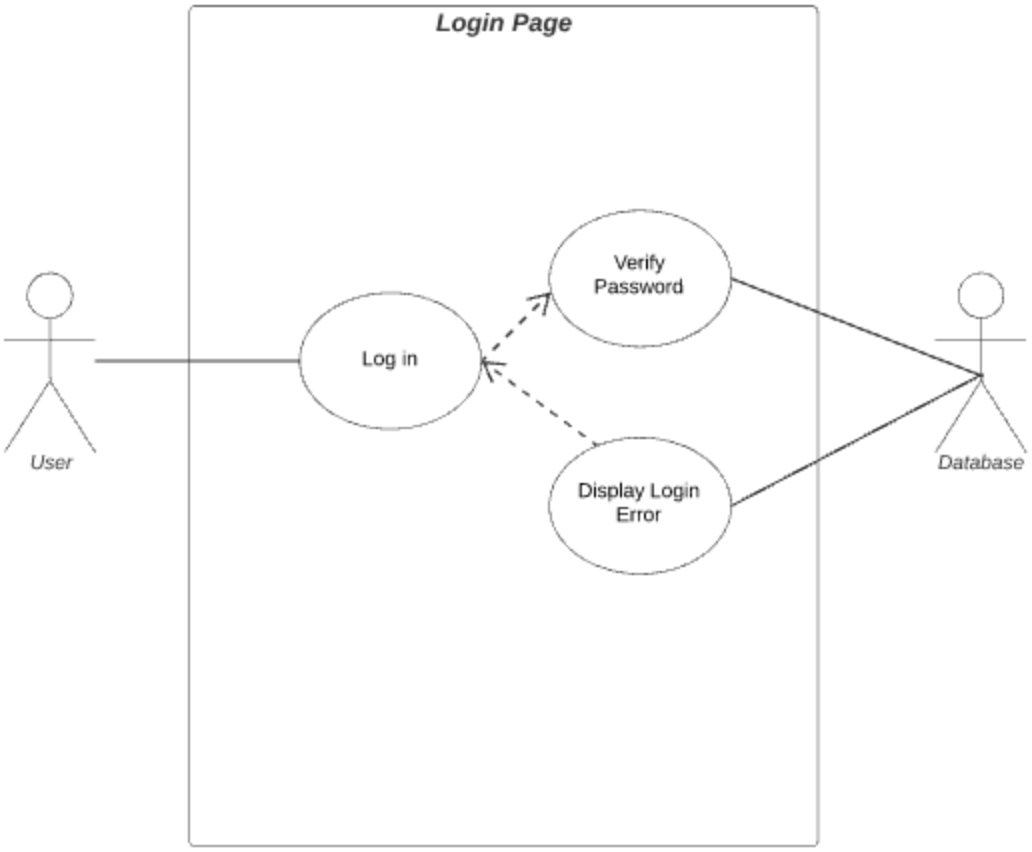


3.2. Login

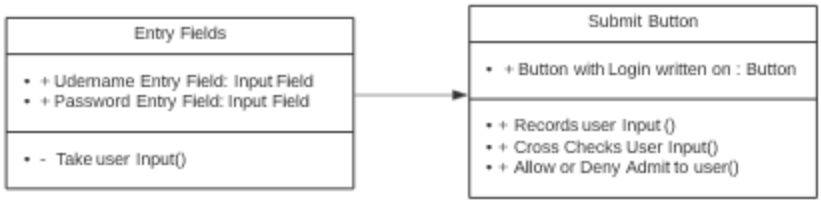
3.2.1. Description

This Module is used to help a user log into their account. Using this module every user can Login to their account so that they can access the videos that they recorded. This contains the username and Password Fields.

3.2.2. Use Case Diagram



3.2.3. Class Diagram



3.2.3.1 Entry Fields

3.2.3.1.1 Entry Fields - Description

The fields in which the user enters the username and password.

3.2.3.1.2 Entry Fields - Data members

Data Type	Data Name	Initial Value	Description
Input Field	Username Field	“”	Records UserName
Input Field	Password Field	“”	Records Password

3.2.3.1.3.1 Entry Fields - Function 1 - Take User Input

This take username and password of the user

3.2.3.2 Submit Button

3.2.3.2.1 Submit Button Description

This button is used to login a user into the software.

3.2.3.2.2 Submit Button - Data members

Data Type	Data Name	Initial Value	Description

Button	Login	NULL	Records , checks users credentials

3.2.3.2.3.1 Submit Button - Function 1 - Records User Input

This take username and password of the user

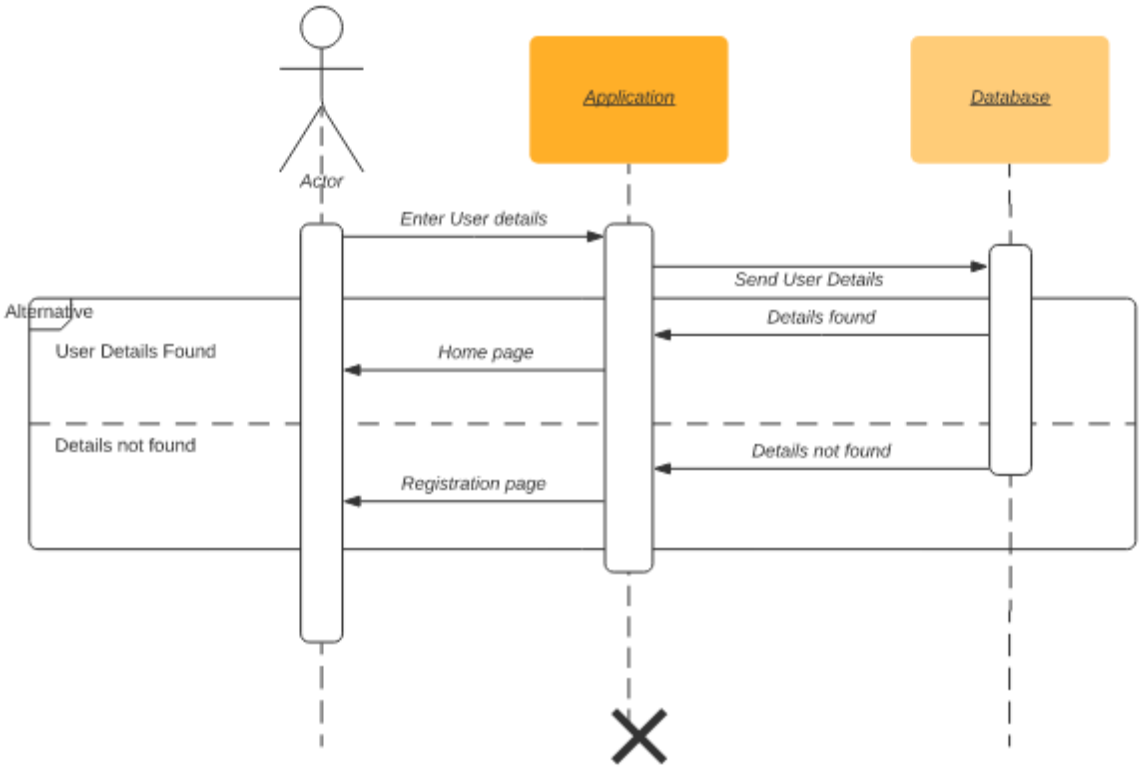
3.2.3.2.3.2 Submit Button - Function 1 - Verifies User Input

This verifies username and password of the user with the database

3.2.3.2.3.3 Submit Button - Function 1 - Allows or denies user

After Verification, the user is either allowed or denied.

3.2.4 Sequence Diagram



3.2.5 Packaging and Deployment Diagrams

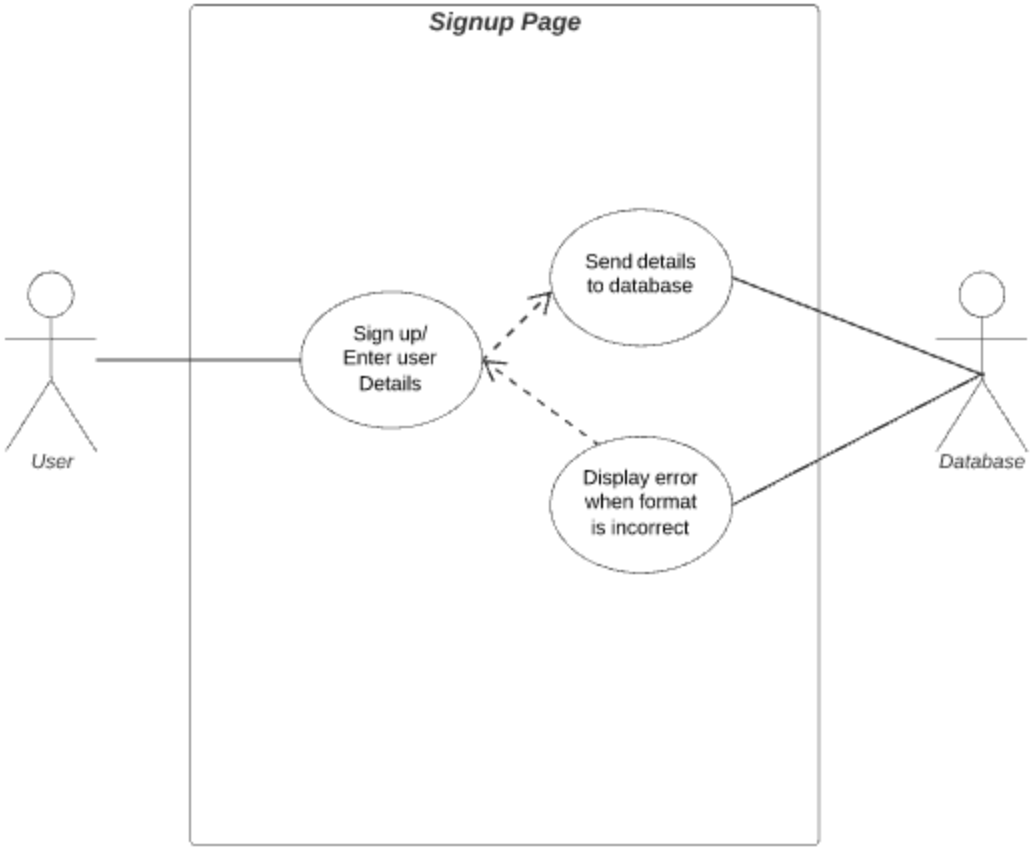
[The packaging and deployment diagrams for the system shall be presented here.]

3.3 Signup

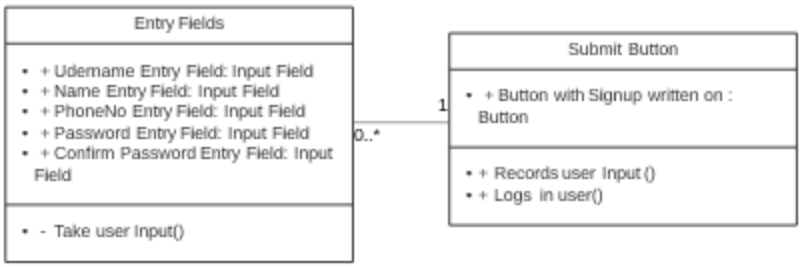
3.3.1 Description

This module is used to take in the credentials of new users and store it to the database, so that they can be cross verified when a user logs in.

3.3.2 Use Case Diagram



3.3.3 Class Diagram



3.3.3.1 Entry Fields

3.3.3.1.1 Entry Fields - Description

The fields in which the user enters the username,name, Phno and password

3.3.3.1.2 Entry Fields - Data members

Data Type	Data Name	Initial Value	Description
Input Field	Username Field	“”	Records UserName
Input Field	Name Field	“”	Records name
Input Field	PhoneNo Field	“”	records password
Input Field	Password Field	“”	Records Password
Input Field	Confirm Password Field	“”	Records Password and checks with above password

3.3.3.1.3.1 Entry Fields - Function 1 - Take User Input

This takes the username, name, PhoneNo and password of the user.

3.3.3.2 Submit Button

3.3.3.2.1 Submit Button Description

This button is used to Signup a user into the software.

3.3.3.2.2 Submit Button - Data members

Data Type	Data Name	Initial Value	Description
Button	Signup	NULL	Records user, Logs in

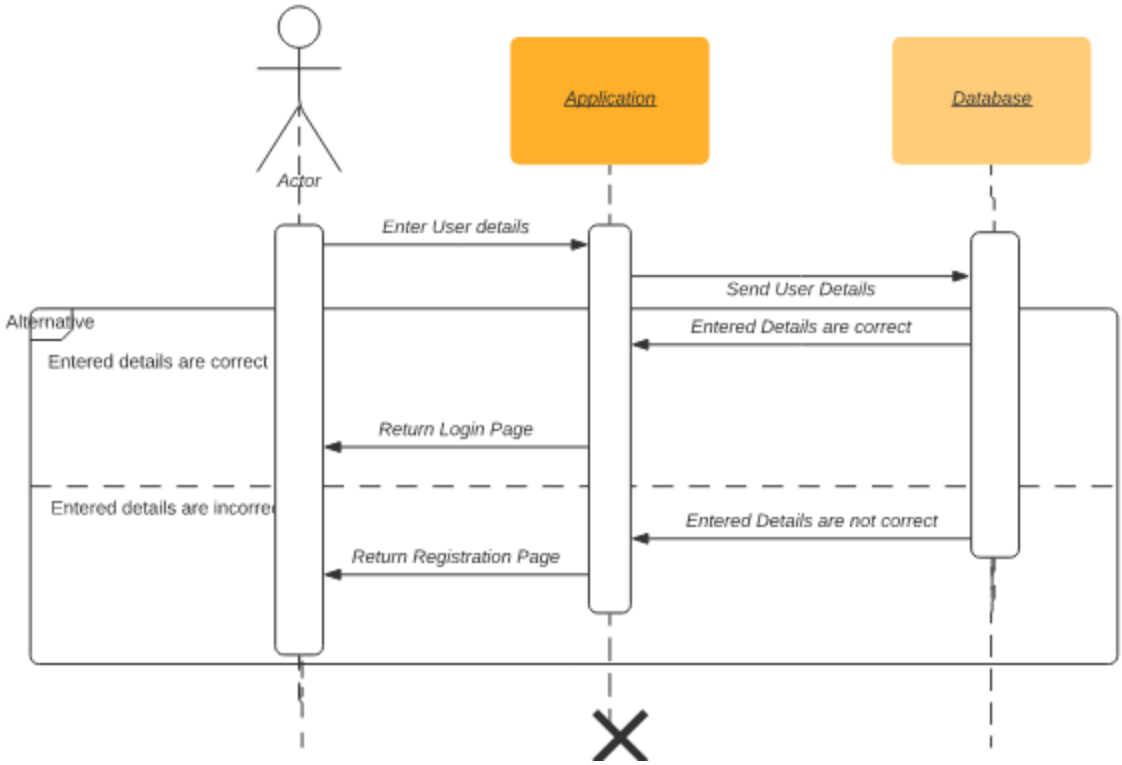
3.3.3.2.3.1 Submit Button - Function 1 - Records User Input

This take username and password of the user

3.3.3.2.3.1 Submit Button - Function 2 - Logs in user

The user is Logged in into the account

3.3.4 Sequence Diagram



3.3.5 Packaging and Deployment Diagrams

[The packaging and deployment diagrams for the system shall be presented here.]

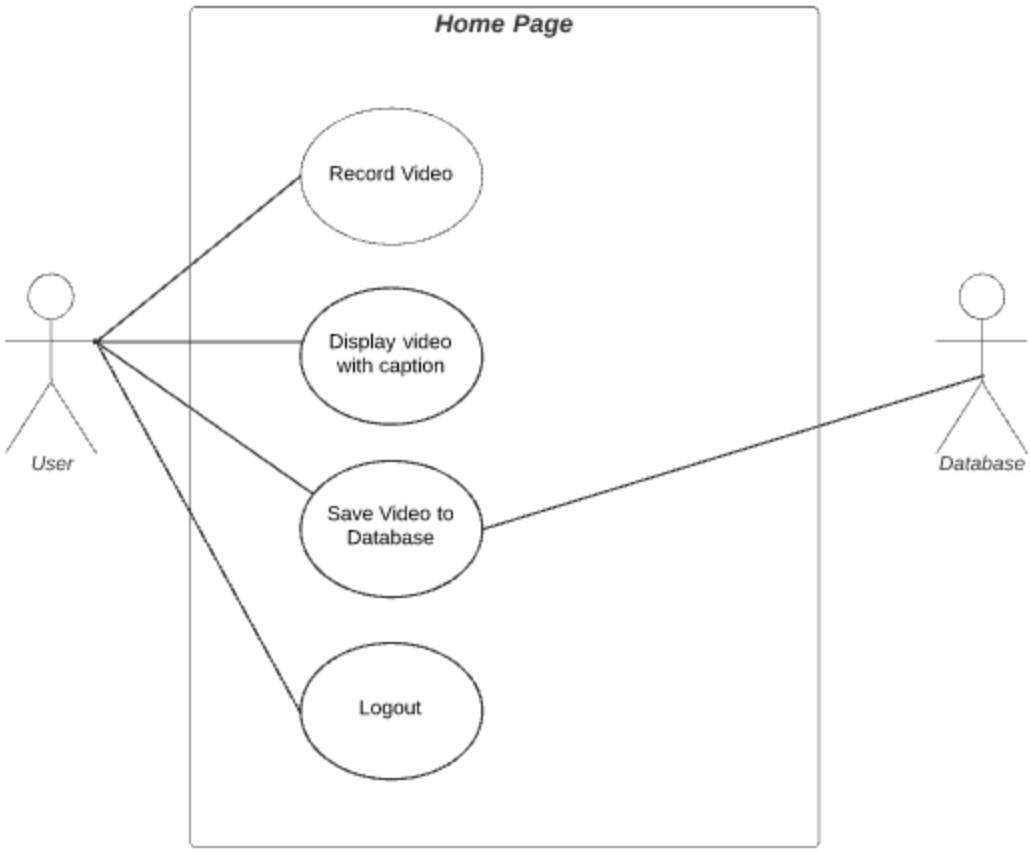
For Example:

3.4 Home

3.4.1 Description

This is the main page where video recording and captioning happens. A video interface and a capture button is present where the user can capture their video and

3.4.2 Use Case Diagram



3.4.3 Class Diagram

3.4.3.1 Video Interface

3.4.3.1.1 Video Interface - Description

The fields in which the user enters the username,name, Phno and password

3.4.3.1.2 Video Interface - Data members

Data Type	Data Name	Initial Value	Description
mp4	Video Display	NULL	Displays Video
text	Caption Space	“”	Displays Captions

3.4.3.1.3.1 Video Interface - Function 1 - Video Capture

This captures Videos from the camera of the phone.

3.4.3.1.3.2 Video Interface - Function 2 - Video Processing

It processes the video using the model that we have built

3.4.3.1.3.3 Video Interface - Function 3 - Video Captioning

The processed caption along with the video are being displayed.

3.4.3.2 Logout

3.4.3.2.1 Logout - Description

The button that helps a user to logout of their account.

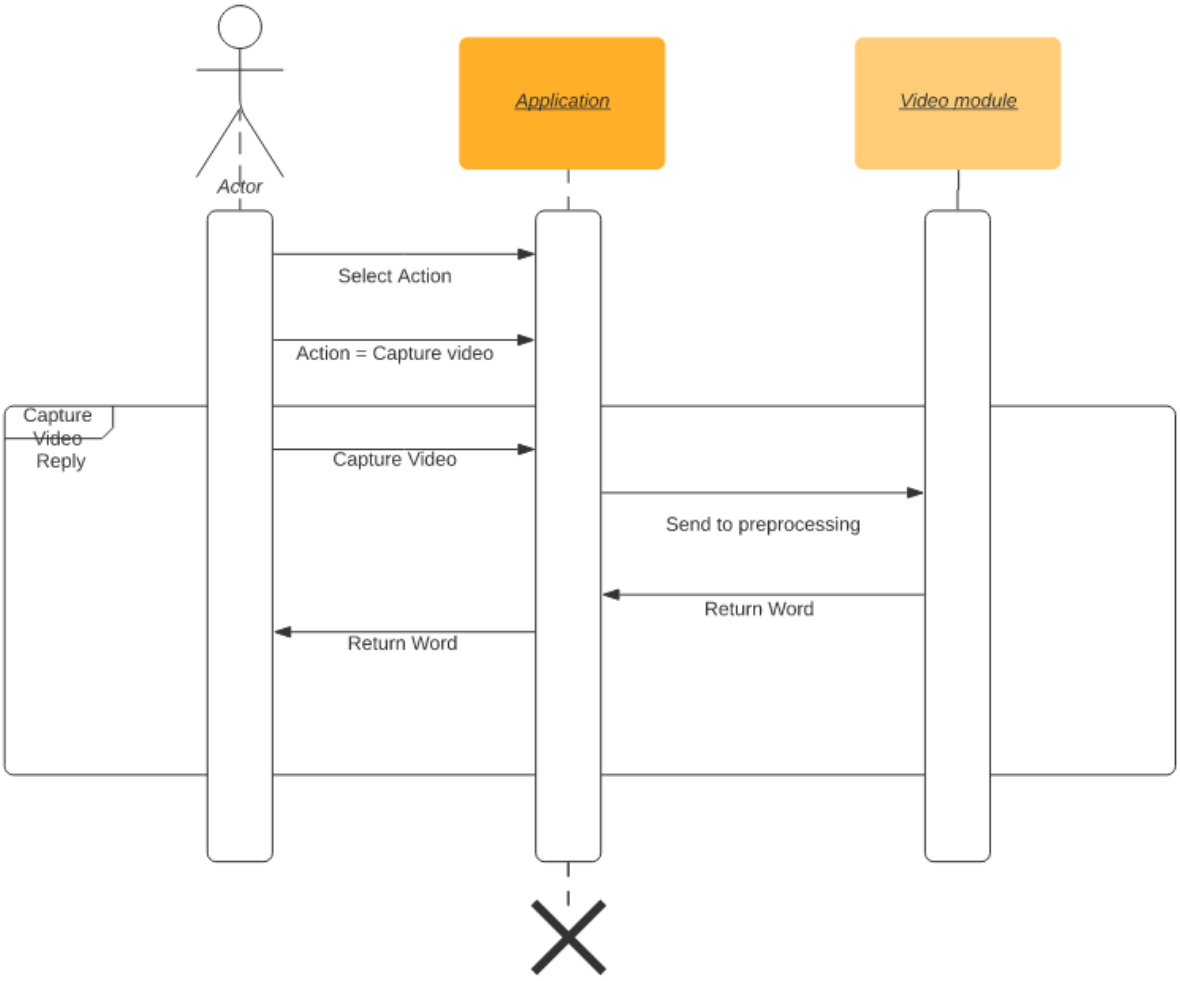
3.4.3.2.2 Logout - Data members

Data Type	Data Name	Initial Value	Description
Button	Logout	NULL	Logsout User

3.4.3.2.3.1 Logout - Function 1 - Logout of account

On clicking on the logout button the user is able to logout of their account.

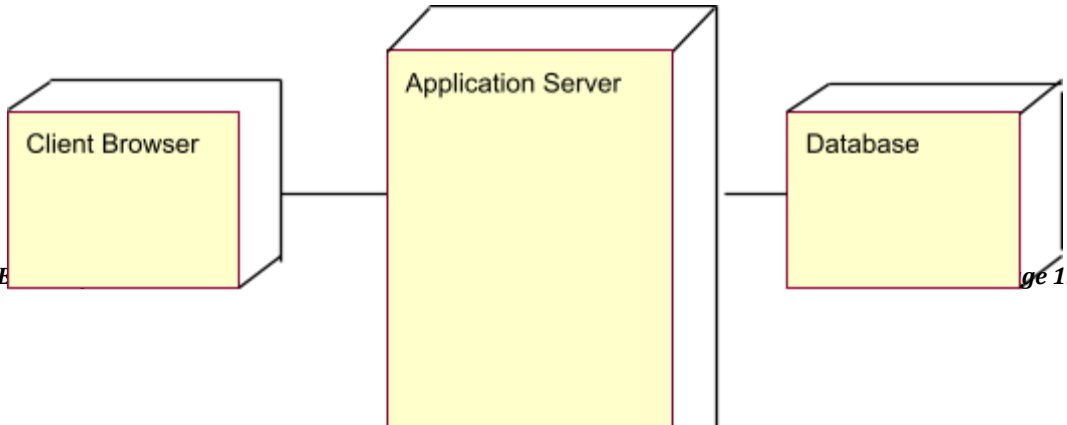
3.4.4 Sequence Diagram



3.4.5 Packaging and Deployment Diagrams

[The packaging and deployment diagrams for the system shall be presented here.]

For Example:

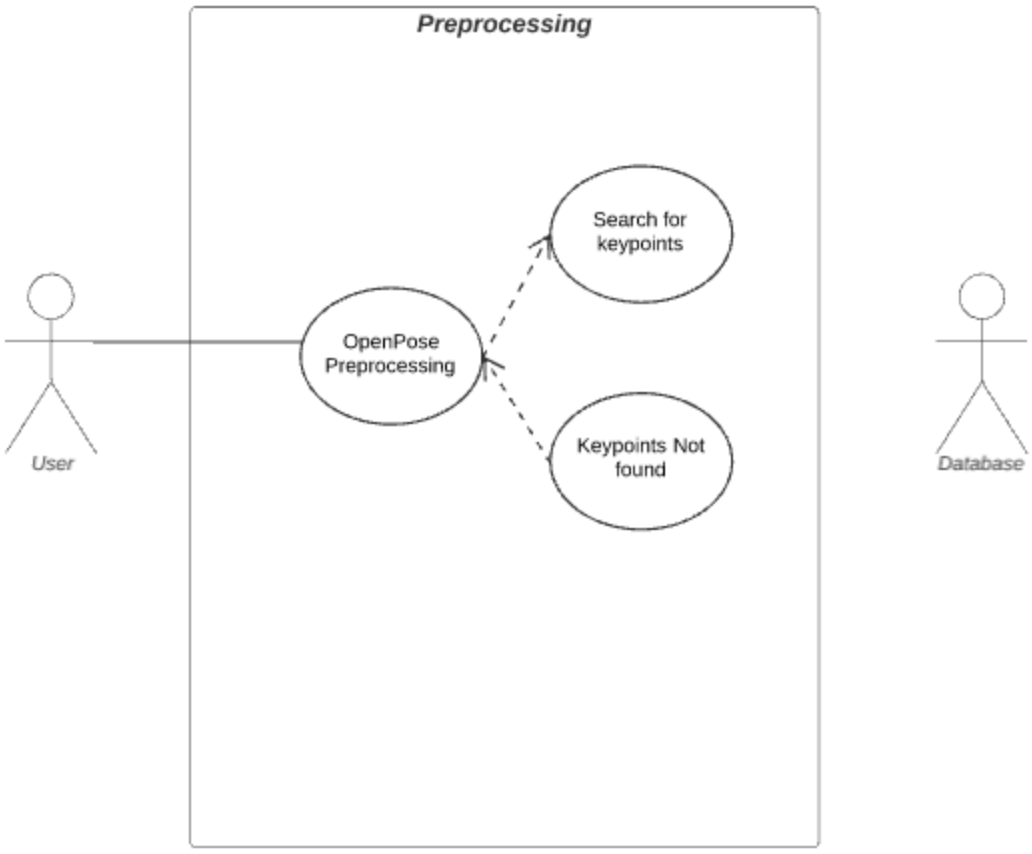


3.5 Preprocessing

3.5.1 Description

In this module the video that is captured from the home screen is preprocessed and the output i.e a openpose Stick Diagram with keypoints is pushed to the next phase

3.5.2 Use Case Diagram



3.5.3 Class Diagram

3.5.3.1 Openpose Preprocessing

3.5.3.1.1 Openpose Preprocessing - Description

The preprocessing method wherein the video can be converted to video with skeletal structures where the movement of frames can be recorded.

3.5.3.1.2 Open Pose Preprocesing - Data members

Data Type	Data Name	Initial Value	Description
mp4	Video Input	NULL	The video that is sent for preprocessing
String	Video Name	Name of each video	Name that can be stored with preprocessed result

3.5.3.1.3.1 Openpose Preprocessing - Function 1 - Openpose

The main function of this module is to convert the video of a person to a video of a skeletal structure where the movement of various key points can be recorded.

3.5.3.2 Finger Keypoint Detection

3.5.3.2.1 Finger Keypoint Detection - Description

The preprocessing method wherein the fingers in a video can be converted into Finger keypoints

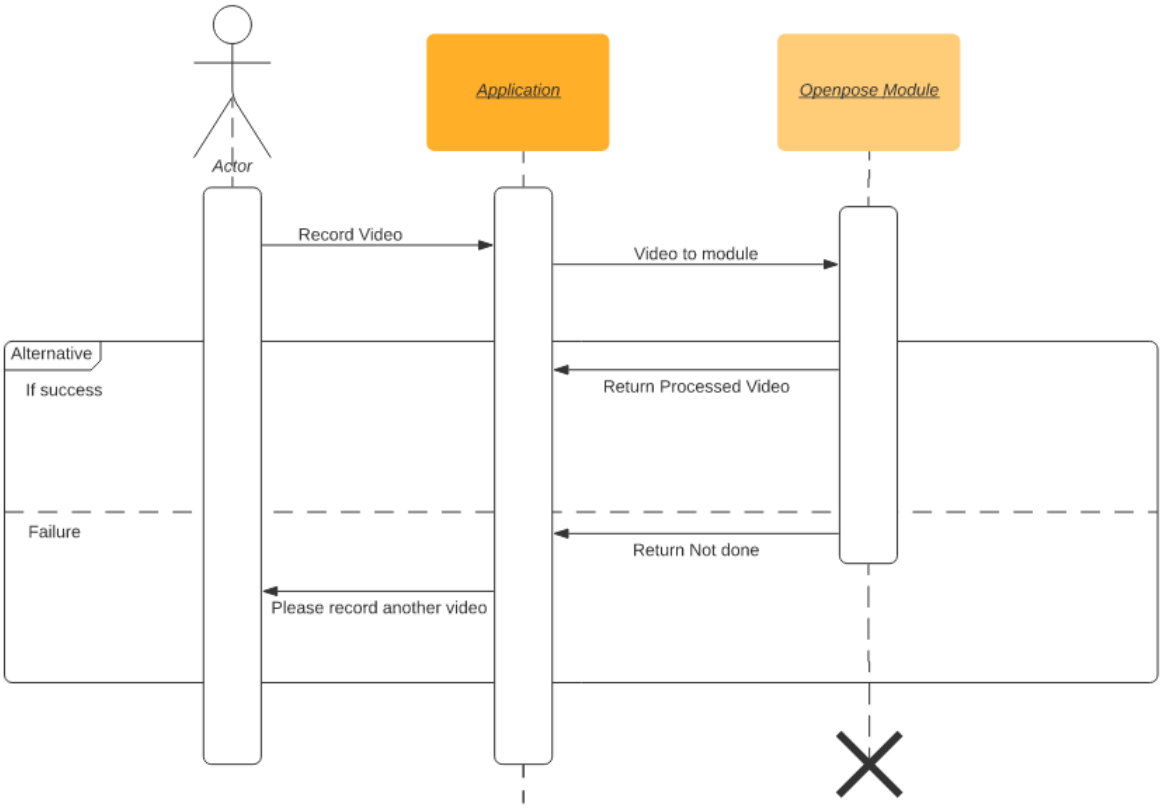
3.5.3.2.2 Finger Keypoint Detection - Data members

Data Type	Data Name	Initial Value	Description
mp4	Video Input	NULL	The video that is sent for preprocessing
String	Video Name	Name of each video	Name that can be stored with preprocessed result

3.5.3.2.3.1 Finger Keypoint Detection - Function 1 - Mediapipe

The main function of this module is to convert the video of a person to a video of a skeletal structure where the movement of various keypoints can be recorded.

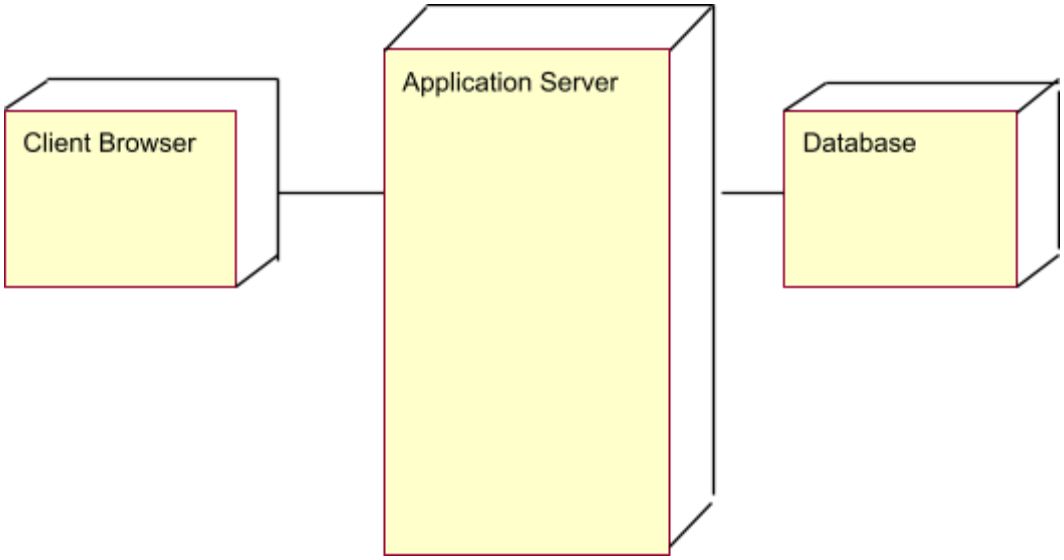
3.5.4 Sequence Diagram



3.5.5 Packaging and Deployment Diagrams

[The packaging and deployment diagrams for the system shall be presented here.]

For Example:



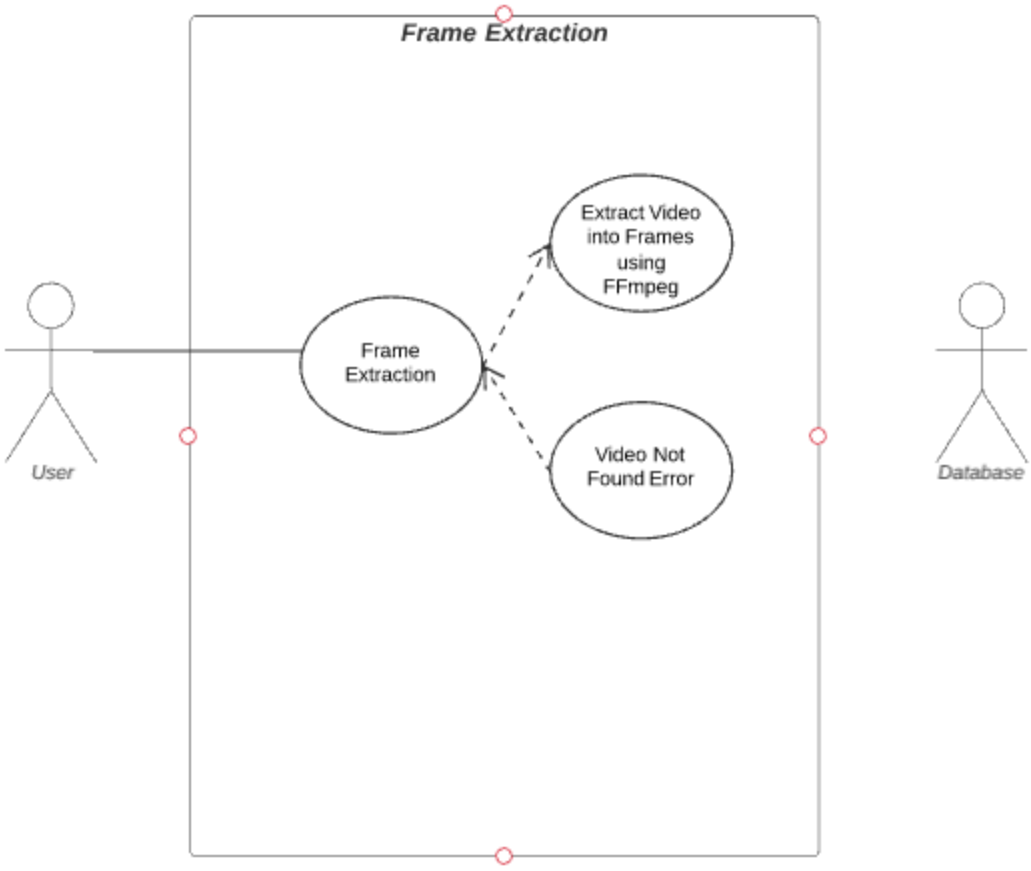
Appendix A:

3.6 Frame Extraction

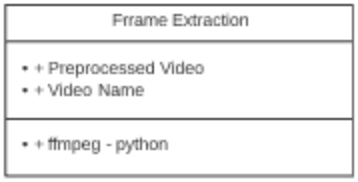
3.6.1 Description

The preprocesses video files along with the names are bought into this module where the video is divided into 15 frames.

3.6.2 Use Case Diagram



3.6.3 Class Diagram



3.6.3.1 Frame Extraction

3.6.3.1.1 Frame Extraction - Description

In this step the Video which was preprocessed in the previous step is taken and it is divided into 15 frames and each frame is named after the video.

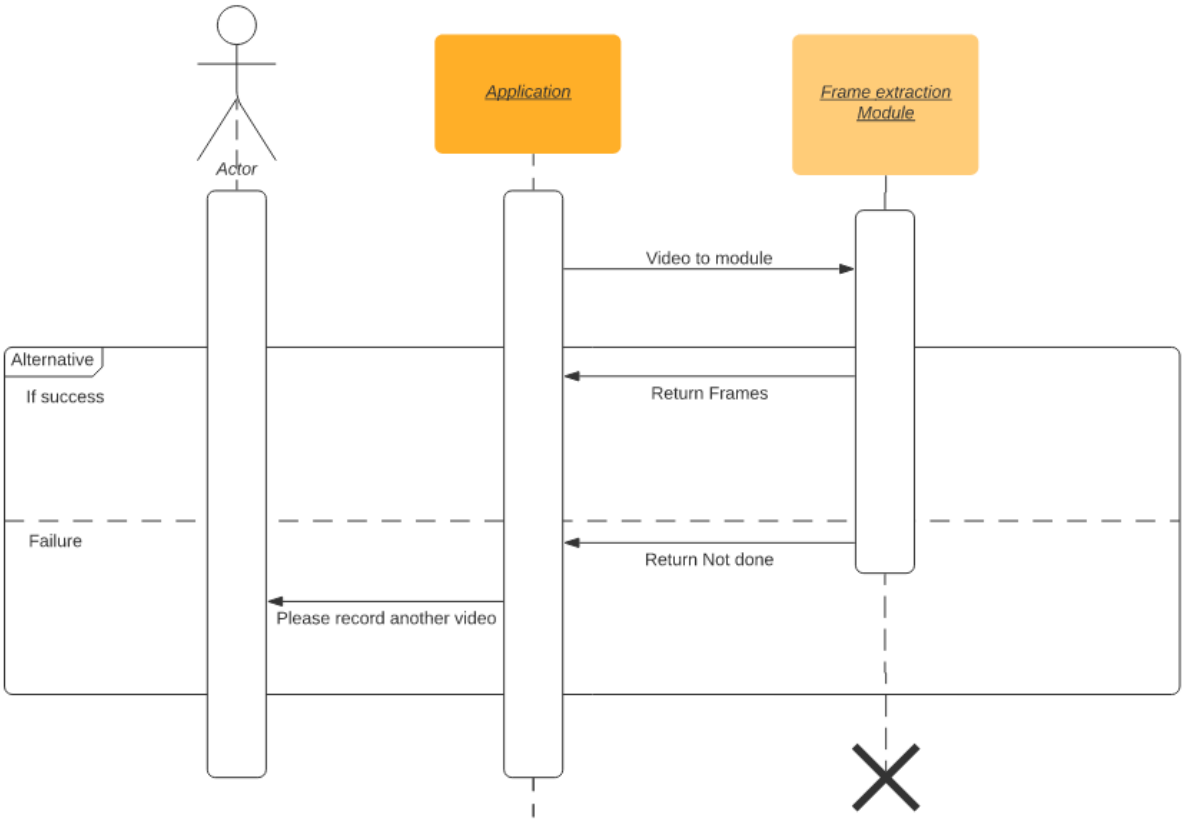
3.6.3.1.2 Frame Extraction - Data members

Data Type	Data Name	Initial Value	Description
mp4	Preprocessed Video	NULL	The video that is sent for preprocessing
String	Video Name	Name of each video	Name that can be stored with preprocessed result

3.6.3.1.3.1 Frame Extraction - Function 1 - FFmpeg - python

The main function of this module is to convert the video of a person to a video of a skeletal structure where the movement of various key points can be recorded.

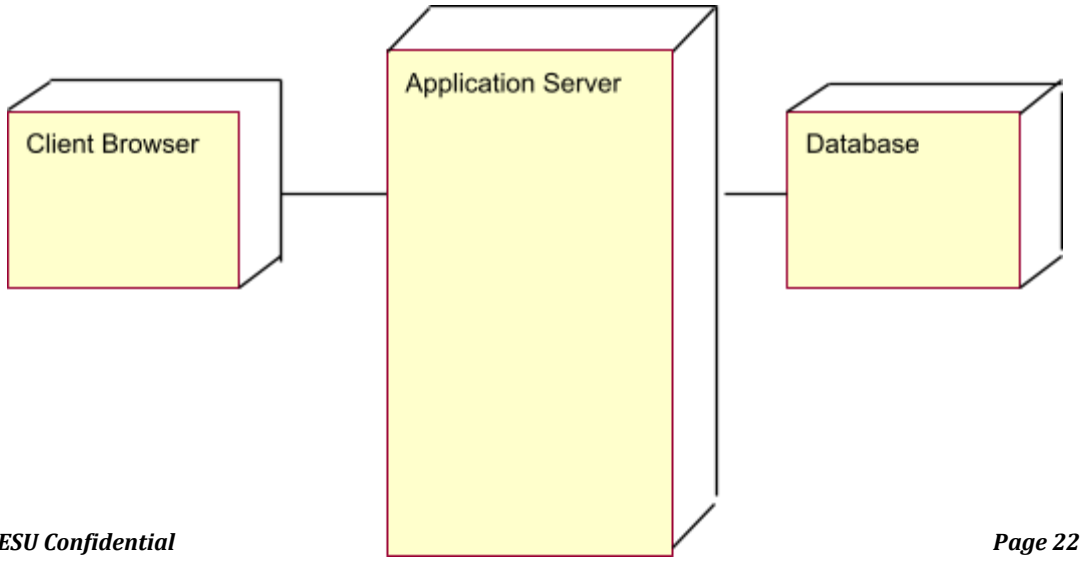
3.6.4 Sequence Diagram



3.6.5 Packaging and Deployment Diagrams

[The packaging and deployment diagrams for the system shall be presented here.]

For Example:



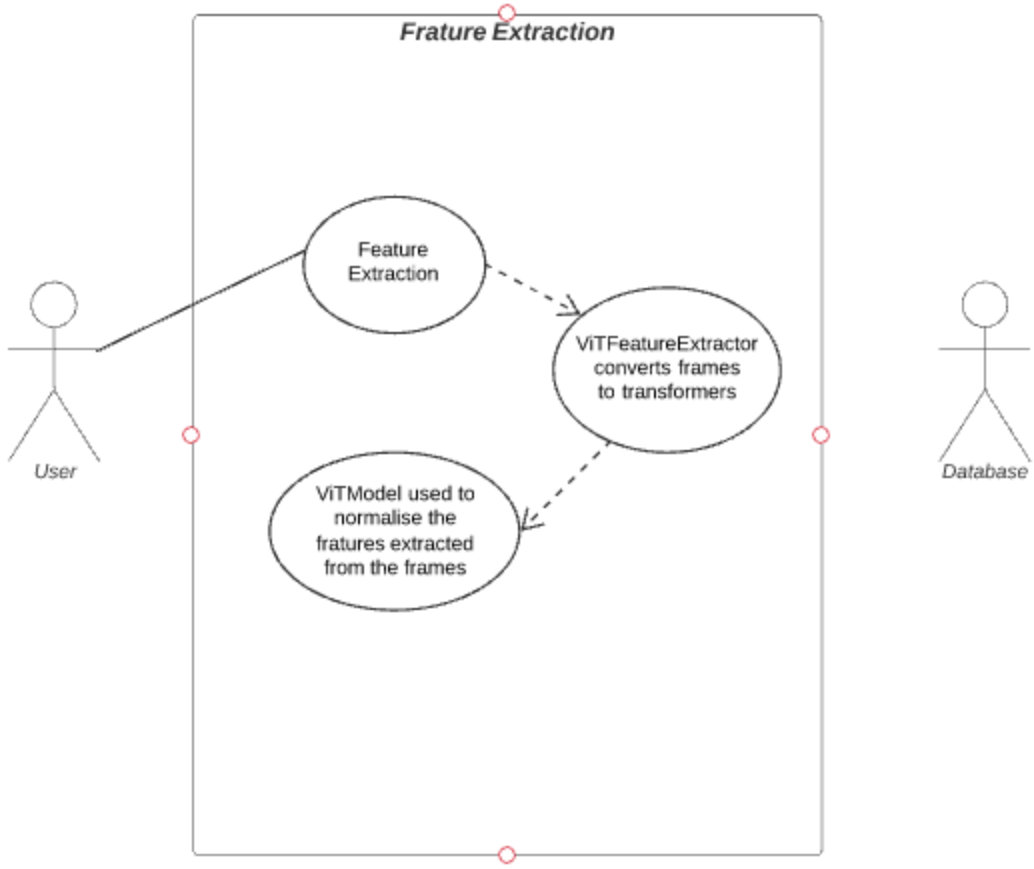
Appendix A:

3.7 Feature Extraction

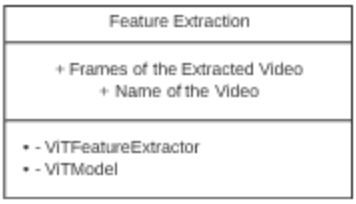
3.7.1 Description

The features are extracted from the frames and are saved in a dictionary with the name of the video as its key, so that classification can be done successfully.

3.7.2 Use Case Diagram



3.7.3 Class Diagram



3.7.3.1 Feature Extraction

3.7.3.1.1 Feature Extraction - Description

In this step the Video which was preprocessed in the previous step is taken and it is divided into 15 frames and each frame is named after the video.

3.7.3.1.2 Feature Extraction - Data members

Data Type	Data Name	Initial Value	Description
jpg	Frames of the preprocessed video	NULL	The frames extracted from the preprocessed video
String	Video Name	Name of each video	Name that can be stored with preprocessed result

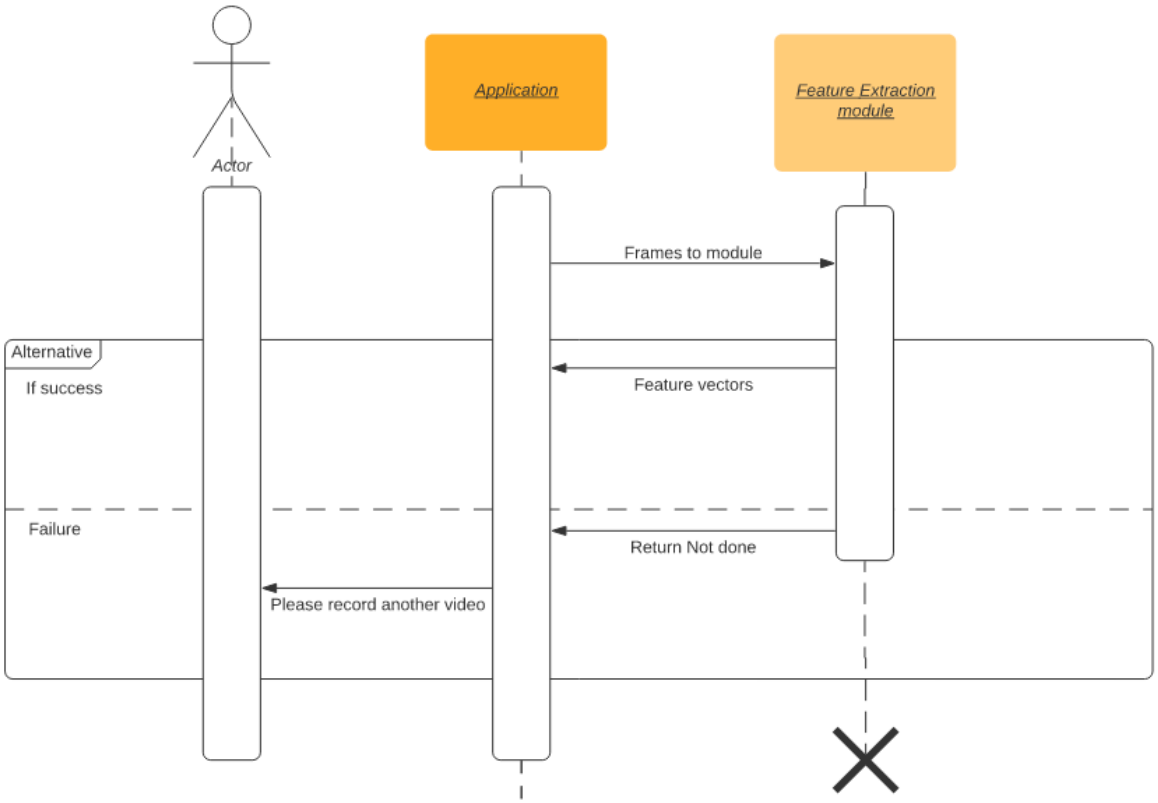
3.7.3.1.3.1 Feature Extraction - Function 1 - VIFeatureExtractor

This module takes in the frames and converts them into transformers

3.7.3.1.3.2 Feature Extraction - Function 1 - VIModel

This takes the transformers as the input and converts them into features/numbers which can later be used for training.

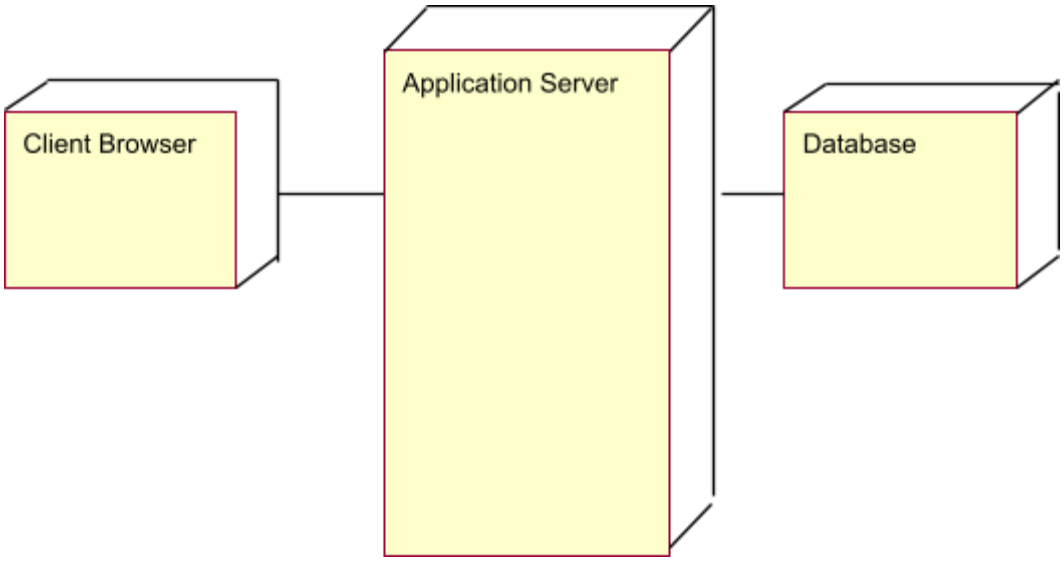
3.7.4 Sequence Diagram



3.7.5 Packaging and Deployment Diagrams

[The packaging and deployment diagrams for the system shall be presented here.]

For Example:



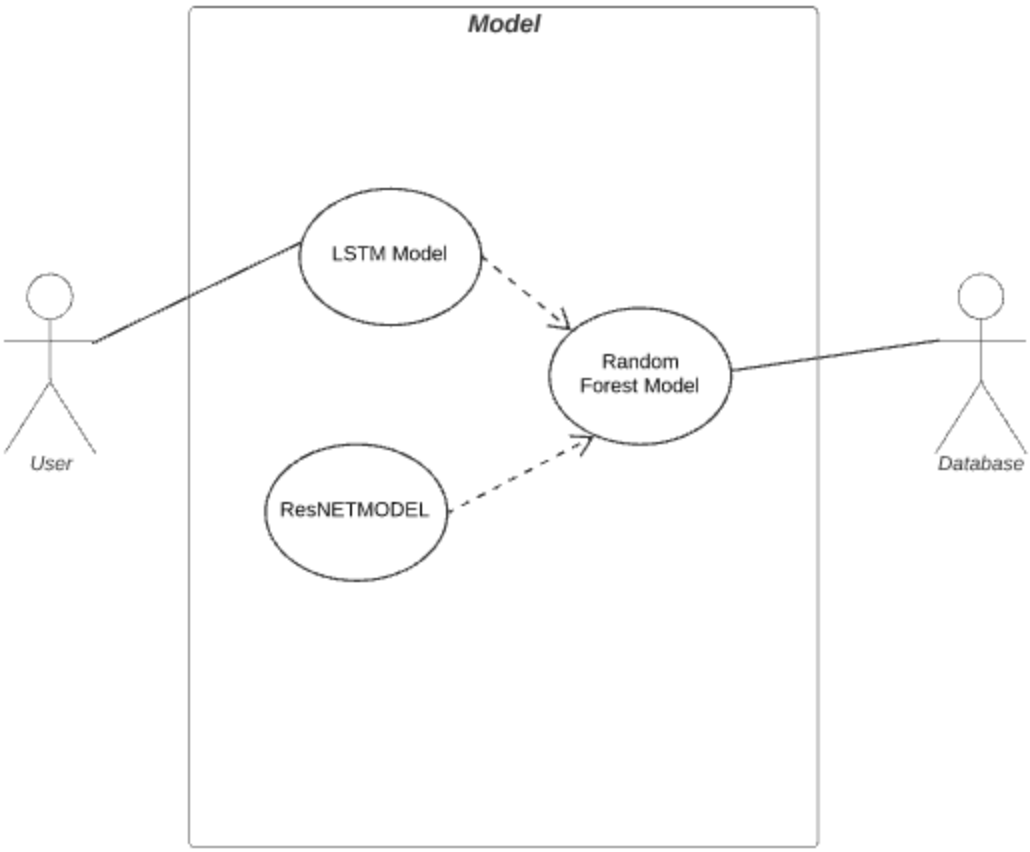
Appendix A:

3.8 Model

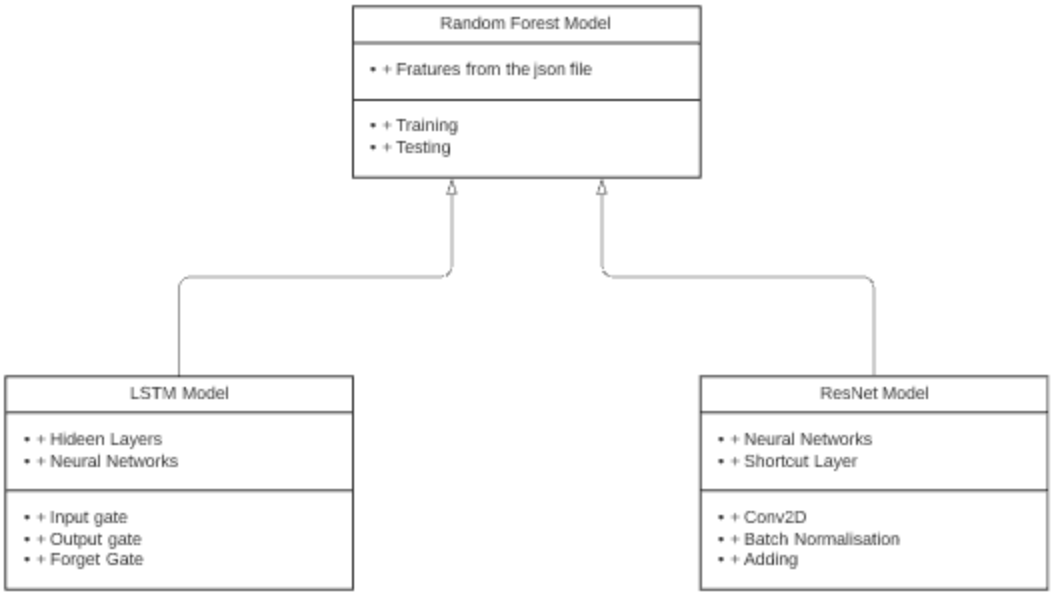
3.8.1 Description

This is the most important model of the entire project. In this step the features are being trained. The methods of modelling used here are RNN, LSTM, ResNet and Random Forest Model wherein the model all the models are combined together and are

3.8.2 Use Case Diagram



3.8.3 Class Diagram



3.8.3.1 LSTM Model

3.8.3.1.1 LSTM Model - Description

LSTM(Long Short Term Memory) models are those models that, unlike the feedforward neural networks, also have a feedback mechanism which helps the networks to also learn from the subsequent networks. It has Encoders, Decoders and Attention mechanisms which will be explained in the next section.

3.8.3.1.2 LSTM Model - Data members

Data Type	Data Name	Initial Value	Description
Layers	Frames of the preprocessed video	NULL	The network of layers formed from the perceptron
Perceptron	Video Name	Name of each video	A basic unit of neural network that consists of

			input values, weights, bais and activation values.

3.8.3.1.3.1 LSTM Model - Function 1 - Input Gate

The input gate controls the information flow to the current cell state using a pointwise multiplication operation of 'sigmoid' and 'tanh' respectively

3.8.3.1.3.2 LSTM Model - Function 2 - Output gate

the output gate decides which information should be passed on to the next hidden state

3.8.3.1.3.3 LSTM Model - Function 3 - Forget Gate

The forget gate decides which information from the previous cell state should be forgotten for which it uses a sigmoid function.

3.8.3.2 ResNet Model

3.8.3.2.1 ResNet Model - Description

ResNet Model is a model that was designed to Uphold many layers so that training can happen better and at the same time the vanishing gradient problem could also be solved.

3.8.3.2.2 LSTM Model - Data members

Data Type	Data Name	Initial Value	Description
Network of layers	Neural Network	NULL	The network of perceptrons
Layer	Shortcut	NULL	A shortcut that can skip over multiple layers to avoid overtraining

3.8.3.2.3.1 ResNet Model - Function 1 - Conv2D

This layer creates a convolution kernel that is convolved with the layer input to produce a tensor of outputs.

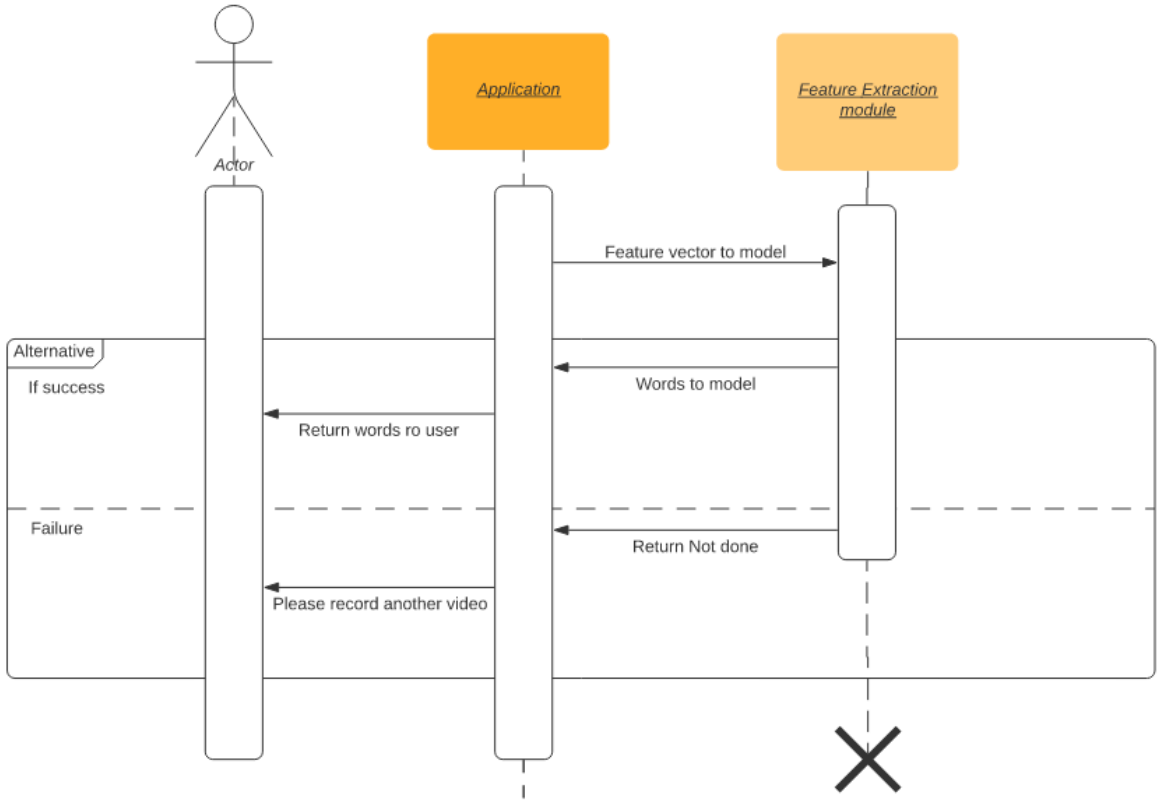
3.8.3.2.3.2 ResNet Model - Function 2 - Batch Normalisation

It is a training layer that normalises its inputs

3.8.3.2.3.3 ResNet Model - Function 3 - Adding

Adding the resultant Matrix of the Shortcut iteration and the normal path iteration

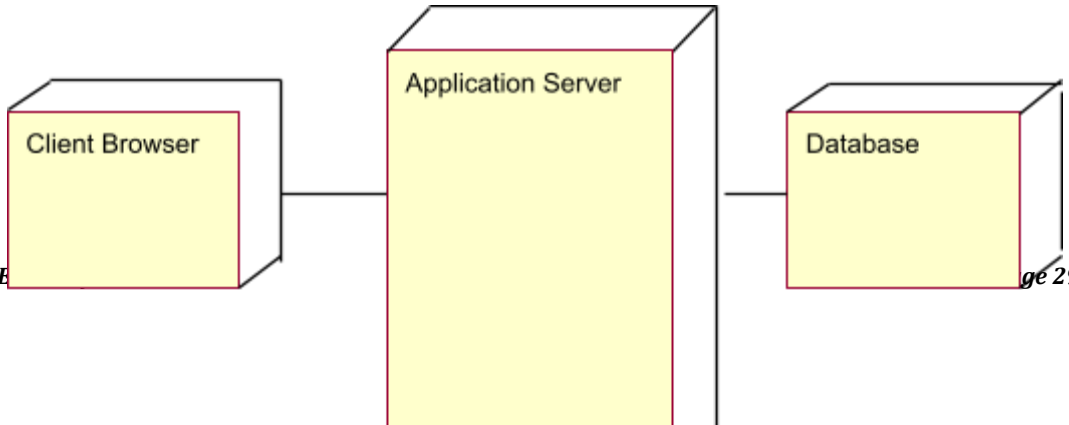
3.8.4 Sequence Diagram



3.8.5 Packaging and Deployment Diagrams

[The packaging and deployment diagrams for the system shall be presented here.]

For Example:

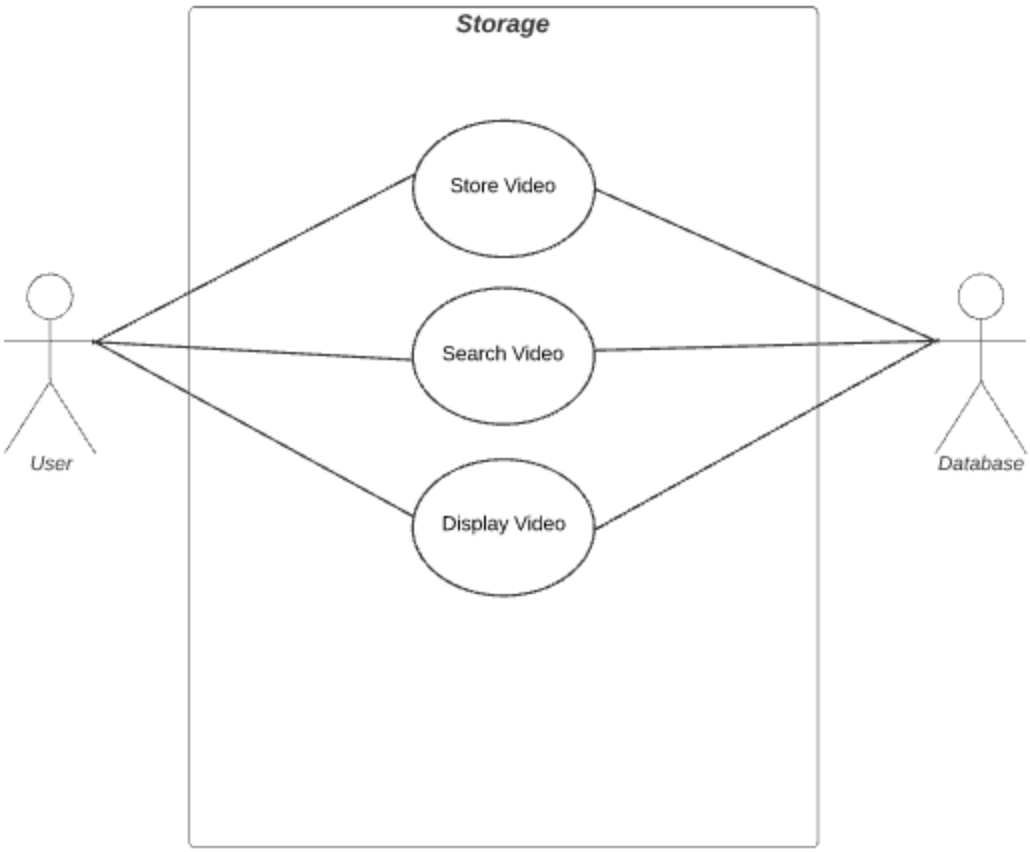


3.9 Storage

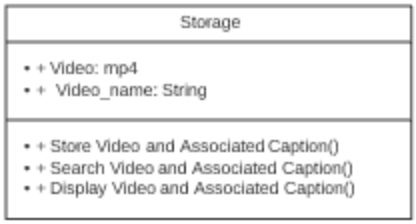
3.9.1 Description

This module is responsible for the backend saving of data where the unprepossessed, preprocessed videos along with their captions are stored and retrieved.

3.9.2 Use Case Diagram



3.9.3 Class Diagram



3.9.3.1 Storage

3.9.3.1.1 Storage - Description

In this step the Video which was preprocessed in the previous step is taken and it is divided into 15 frames and each frame is named after the video.

3.9.3.1.2 Storage - Data members

Data Type	Data Name	Initial Value	Description
mp4	Video	NULL	The video that is being stored
String	Video Name	Name of each video	Name that can be stored with preprocessed result

3.9.3.1.3.1 Storage - Function 1 - Store Video and associated Caption()

This function is used to store the Video and caption

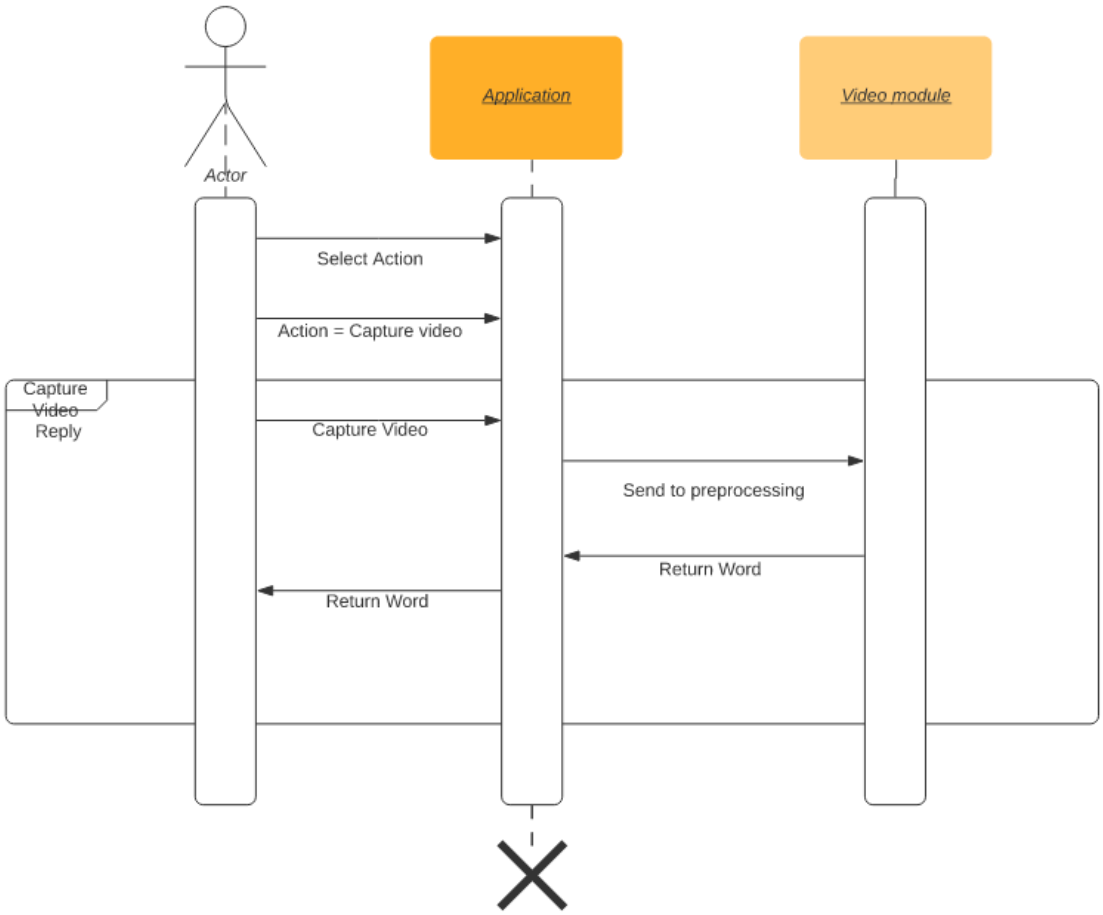
3.9.3.1.3.2 Storage - Function 2 - Search Video and associated Caption()

This function is used to search for the video and its associated caption

3.9.3.1.3.3 Storage - Function 3 - Display Video and associated Caption()

This function is used to display the video and its associated caption

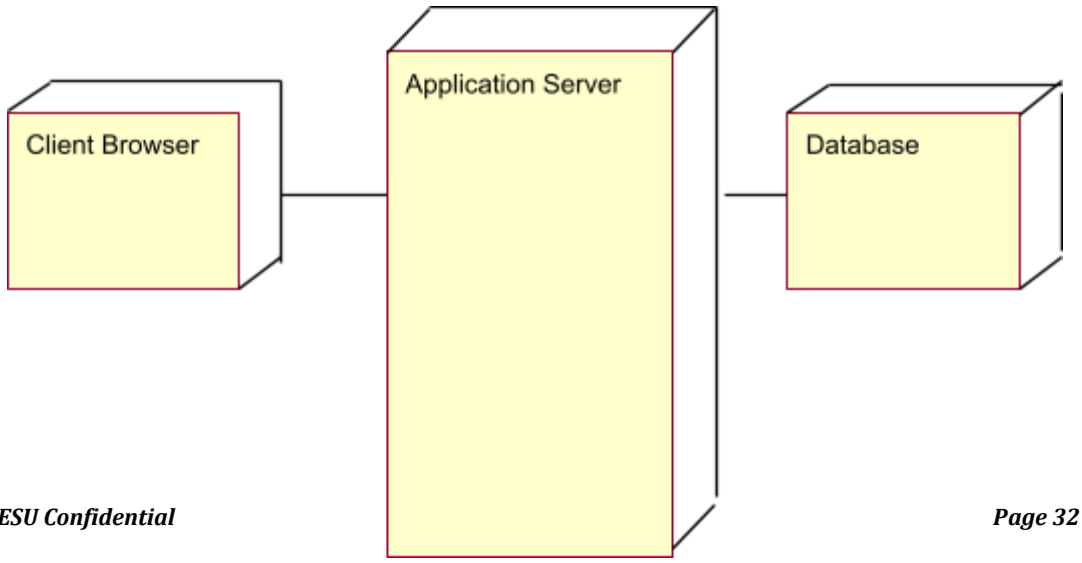
3.9.4 Sequence Diagram



3.9.5 Packaging and Deployment Diagrams

[The packaging and deployment diagrams for the system shall be presented here.]

For Example:



Appendix A: Definitions, Acronyms and Abbreviations

1) ML :- Machine Learning
2) DL : Deep Learning
3) ISL :- INdian Sign Language
4) CNN :- Convolutional Neural Network
5) RNN :- Recurrent Neural Network
6) UML :- Unified Machine Language
7) INCLUDE:- Indian Lexicon Sign Language Dataset
8) UI : User Interface

Appendix B: References

1) Danielle Bragg, Oscar Koller, Mary Bellard, Larwan Berke, Patrick Boudreault, Annelies Braffort, Naomi Caselli, Matt Huenerfauth, Hernisa Kacorri, Tessa Verhoef, Christian Vogler, and Meredith Ringel Morris.2019. Sign Language Recognition, Generation, and Translation: An Interdisciplinary Perspective. In The 21st International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '19). Association for Computing Machinery, New York, NY, USA, 16–31. DOI:https://doi.org/10.1145/3308561.3353774

2) P. C. Badhe and V. Kulkarni, "Indian sign language translator using gesture recognition algorithm," 2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS), Bhubaneswar, India, 2015, pp. 195-200, doi: 10.1109/CGVIS.2015.7449921.

3) Parton, Becky. (2006). Sign Language Recognition and Translation: A Multidisciplined Approach From the Field of Artificial Intelligence. Journal of deaf studies and deaf education. 11. 94-101. 10.1093/deafed/enj003.

4) N. C. Camgoz, S. Hadfield, O. Koller, H. Ney and R. Bowden, "Neural Sign Language Translation," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 7784-7793, doi: 10.1109/CVPR.2018.00812.

5) Akyol, S., & Alvarado, P. (2001). Finding relevant image content for mobile sign language recognition. Retrieved June 17, 2005, from

http://www.techinfo.rwth-aachen.de/Veroeffentlichungen/V001_2001.pdf

6) Word-level Deep Sign Language Recognition from Video: A New Large-scale Dataset and Methods Comparison DONGXU LI, Cristian Rodriguez, Xin Yu, HONGDONG LI; Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2020, pp. 1459-1469

Appendix C: Record of Change History

#	Date	Document Version No.	Change Description	Reason for Change
1.	**1/11/2021**	**1**	-	**-**
2.				
3.				

Appendix D: Traceability Matrix

Project Requirement Specification Reference Section No. and Name.	DESIGN / HLD Reference Section No. and Name.	LLD Reference Section No. Name
Login page	1	2
Signup Page	2	3
Home page	3	1
Help Page	4	8
How to login page	5	9
Logout Page	6	-