Department of Computer Science

Hand Gesture to Text Converter (DivyaAnuvadak) BTI_G05

Submitted By :

Ayushi Rajput – 2016730

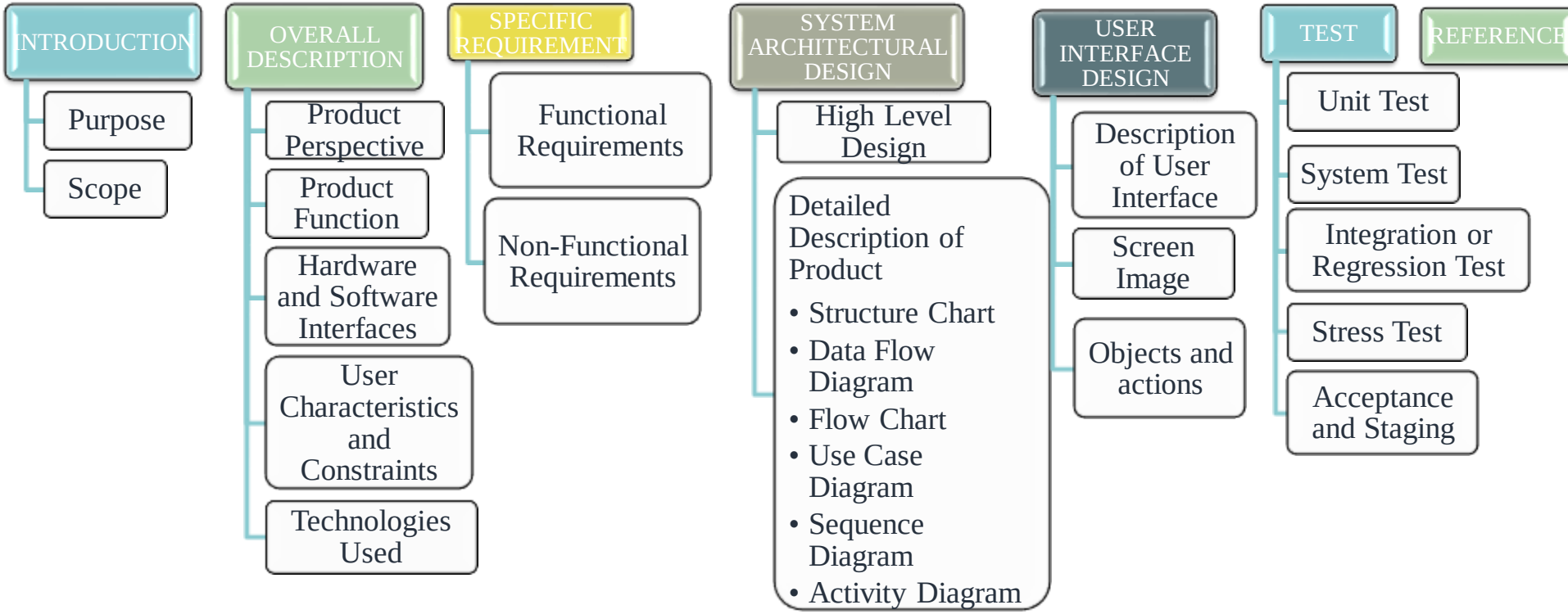
Deepali – 2016736

Shivalik – 2016821

Sneha Poddar – 2016837

© Templateswise.com

Content


INTRODUCTION
Purpose
Scope
OVERALL DESCRIPTION
Product Perspective
Product Function
Hardware and Software Interfaces
User Characteristics and Constraints
Technologies Used
SPECIFIC REQUIREMENT
Functional Requirements
Non-Functional Requirements
SYSTEM ARCHITECTURAL DESIGN
High Level Design
Detailed Description of Product
• Structure Chart
• Data Flow Diagram
• Flow Chart
• Use Case Diagram
• Sequence Diagram
• Activity Diagram
USER INTERFACE DESIGN
Description of User Interface
Screen Image
Objects and actions
TEST
Unit Test
System Test
Integration or Regression Test
Stress Test
Acceptance and Staging
REFERENCE

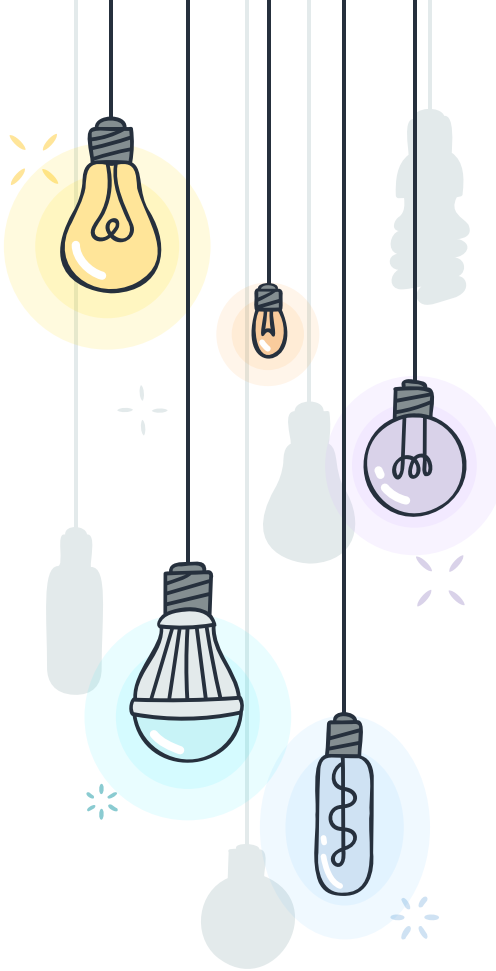
Introduction

Purpose

The main problem that the project aims to solve is to reduce the communication gap of speech or hearing impaired people with the normal ones.

DivyaAnuvadak is a web application that converts ISL to text.

The aim of this presentation is to specify the features, requirements of the final product and the interface of ISL recognition



Scope

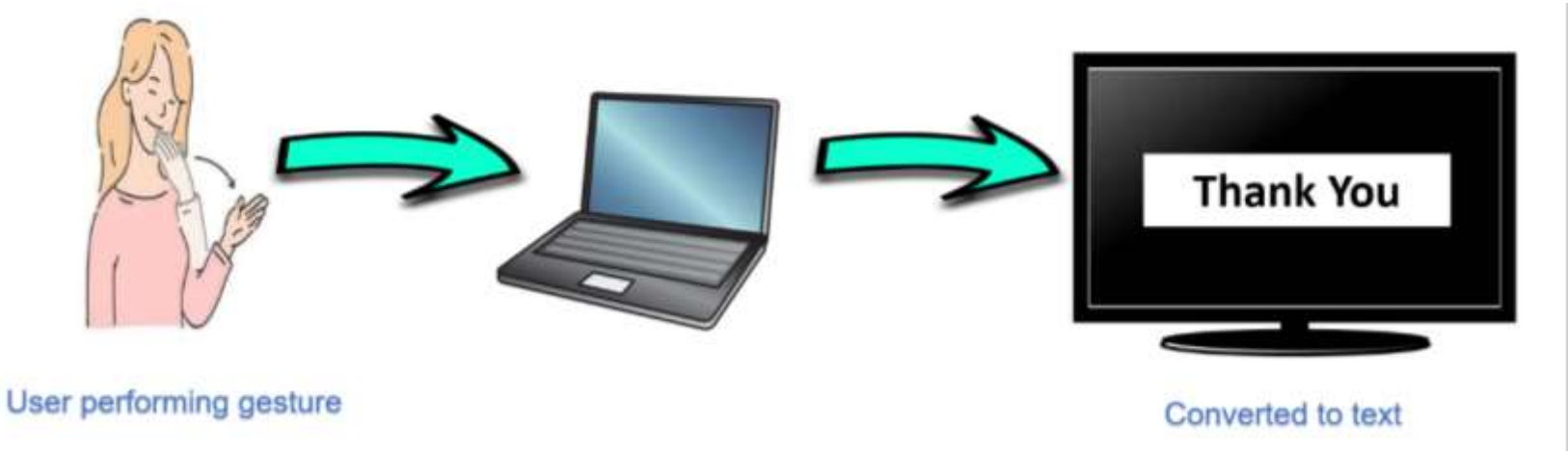
- It will facilitate speech and hearing impaired people to ease their communication.
- It can be used in offices, institutions or seminars.
- It would help both our community and university to build a startup and educate these specially-abled people.

Overall Description

Product Perspective

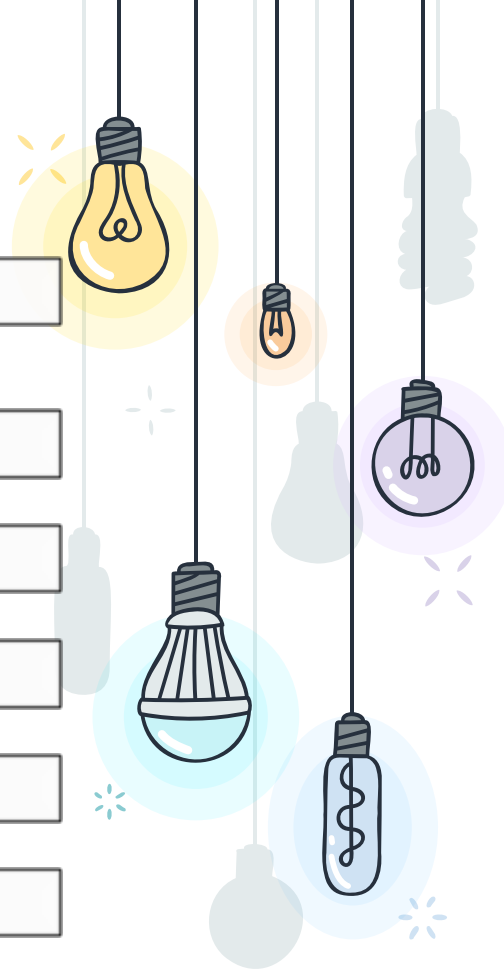
DivyaAnuvadak will serve the speech and hearing impaired people by providing their gestures converted to the equivalent text.

This system will consist of one web portal. This web portal will be used for converting gestures into its corresponding text. The predefined gestures will be stored in the memory and will be compared with the original gesture performed by the user.



Product Function

- Capture the gestures made by the user through a web camera.
- Recognizing the gestures through Tensorflow Object Detection API by identifying features.
- Pre-Processing the captured data.
- Feeding the data to the model.
- Predicting the word based on processed model.
- Selecting the word of highest possibility.
- Converting the word into text and displaying it on the screen.



Hardware Interface

- **SERVER SIDE**:

  RAM : 8 GB (minimum)

  HDD : 1 TB or more

  Processor : Intel® i3 or faster (2–4 GHz)

  GPU : 2 GHz or more

- **CLIENT SIDE**:

  RAM : 8 GB (minimum)

  HDD : 512 GB or more

  Processor : Intel® Pentium or faster (2-4 GHz)

  Camera : 720 Pixels or more

Software Interface

- **SERVER SIDE**:

  Operating System : Windows 10 or 11

  Web Server : Apache Tomcat (v10)

  IDE : Jupyter Notebook (v6.4.8), VS Code (v1.74.3)

  Programming Language : Python

- **CLIENT SIDE**:

  Operating System : Windows 7 or higher versions

  Web Browser : Any Browser

  Library : Streamlit

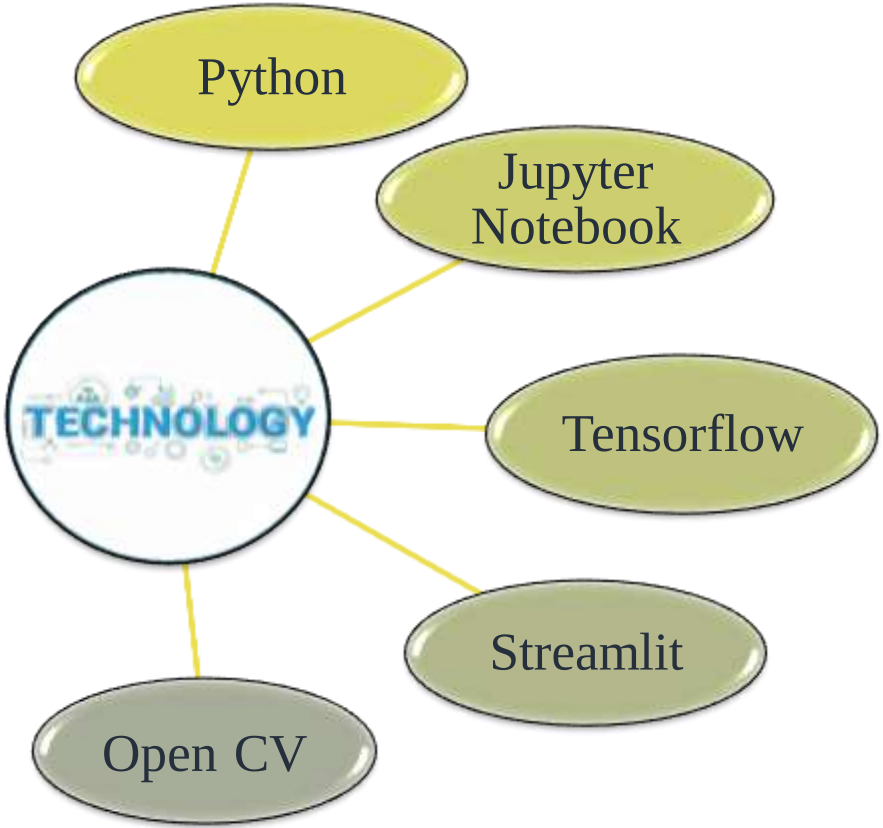
User Characteristics

- User is required to have basic knowledge of computing and internet surfing.
- User should have good knowledge of ISL and English.

Constraints

- Hardware limitation on mobile devices.
- Full-fledged translation is not possible.
- Only one-way communication is possible.
- Fast-paced communications are not possible.

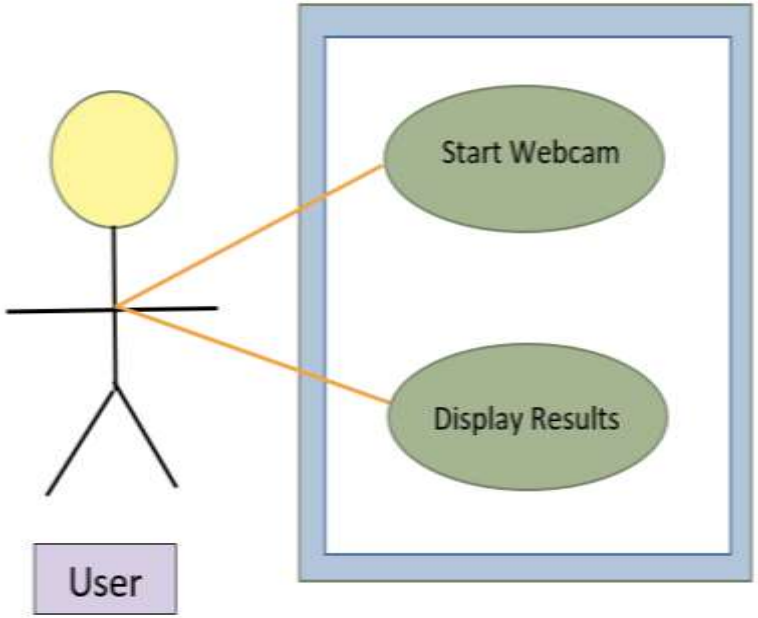
Technologies Used



Specific Requirements

Functional Requirements

Use case Name	Sign Language Recognition
Participating actors	User
Flow of events	Start the system(u) Capturing video(s) Capture gestures(s) Extract features(s) Match features(s) Recognizing gestures(s) Display result
Entry condition	Run the code
Exit condition	Displaying the text label
Quality requirements	Cam pixels clarity, good light condition



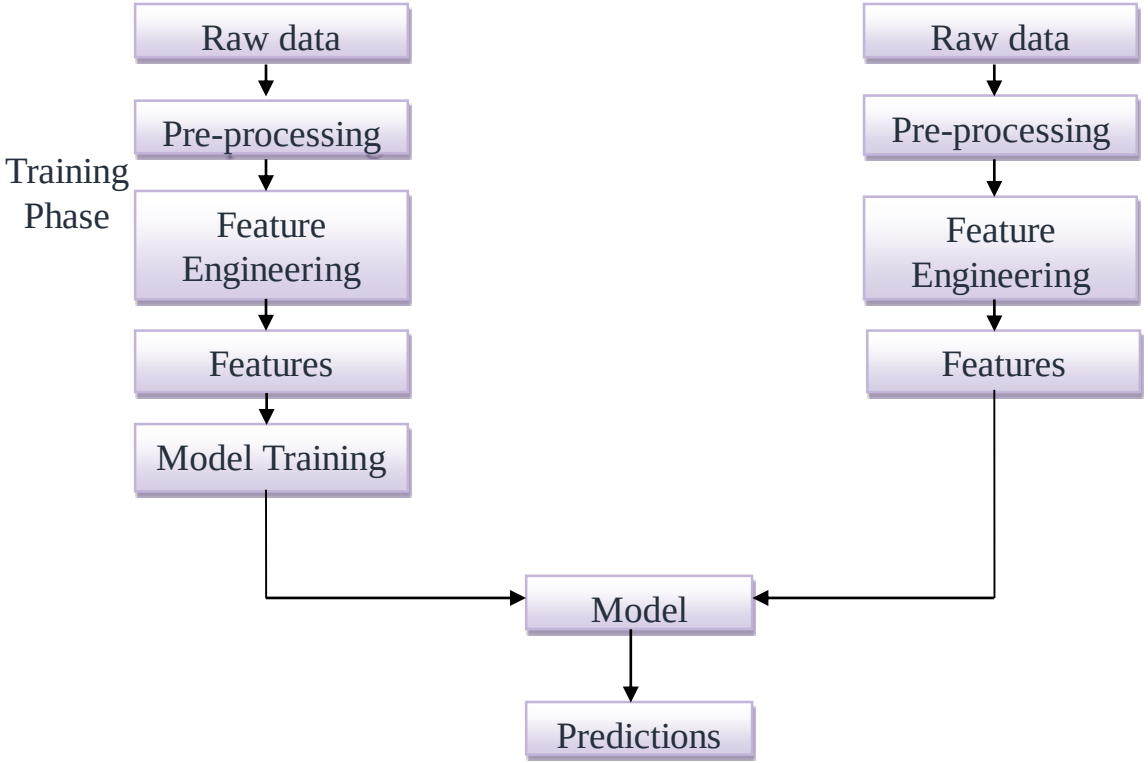
Use Case Diagram

Non-Functional Requirements

Availability	• This web application will be available free on the internet. The users do not require to login so it is user friendly.
Security	• This web application does not ask for any personal information from the user. Hence it is a secure application.
Reliability	• The application will be 99% reliable and will produce correct speech for the corresponding gestures.
Portability	• This project is a web application so it will be available online and can be accessed from any device having internet connectivity, e.g.: laptops, computers, mobile devices etc.

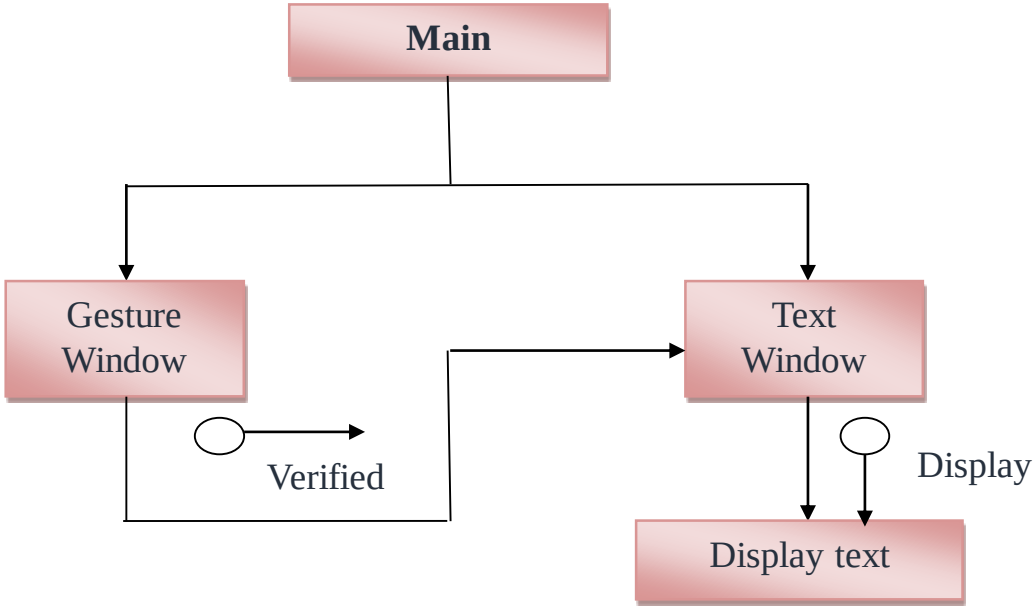
System Architectural Design

High – Level Design

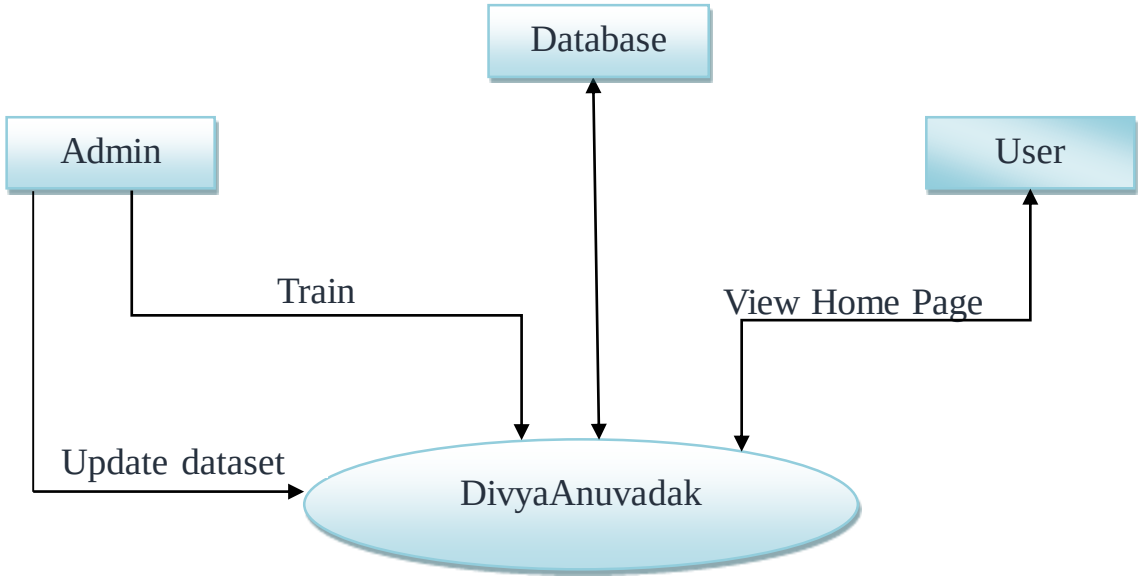


Detailed description of components

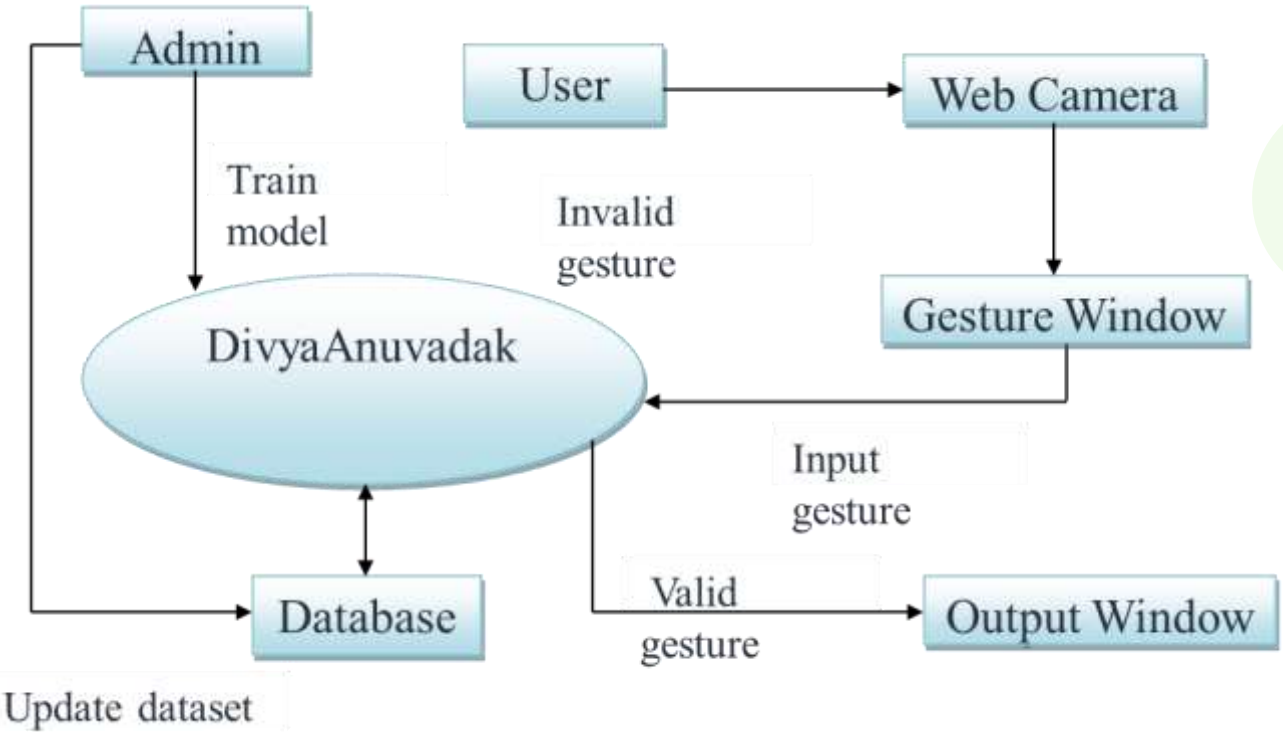
Structure Chart



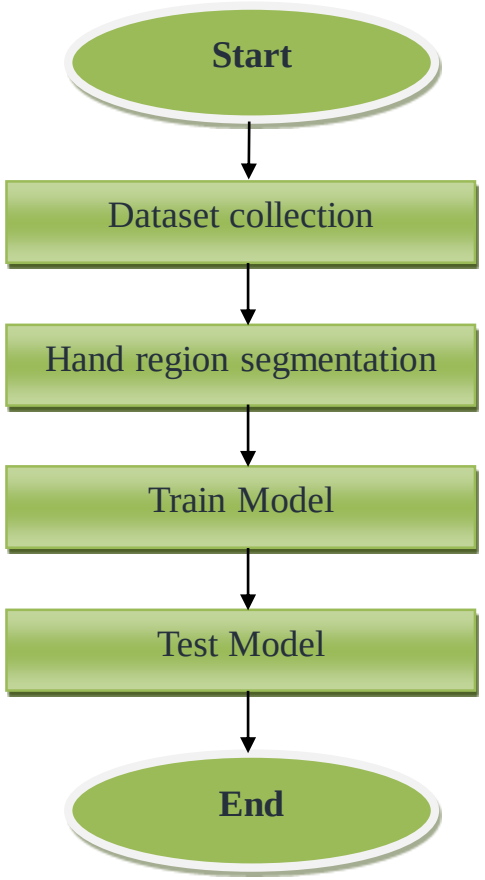
0-Level DFD


Database
Admin
User
Train
View Home Page
Update dataset
DivyaAnuvadak

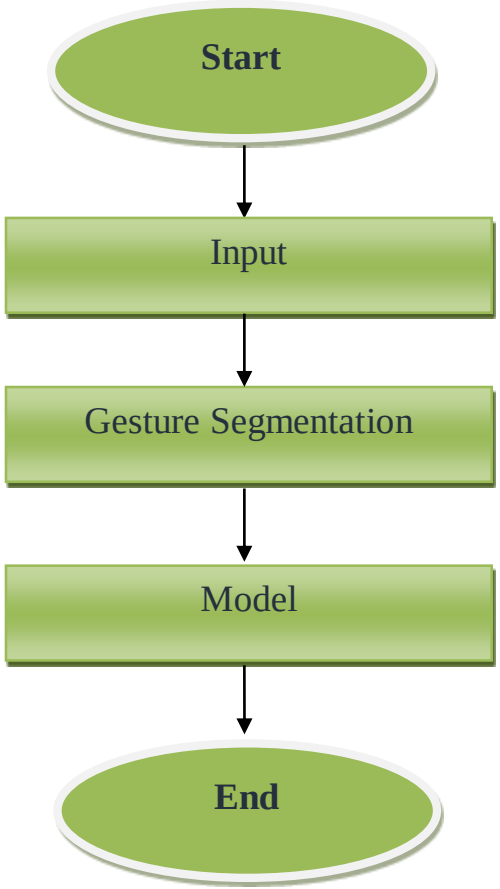
1-Level DFD

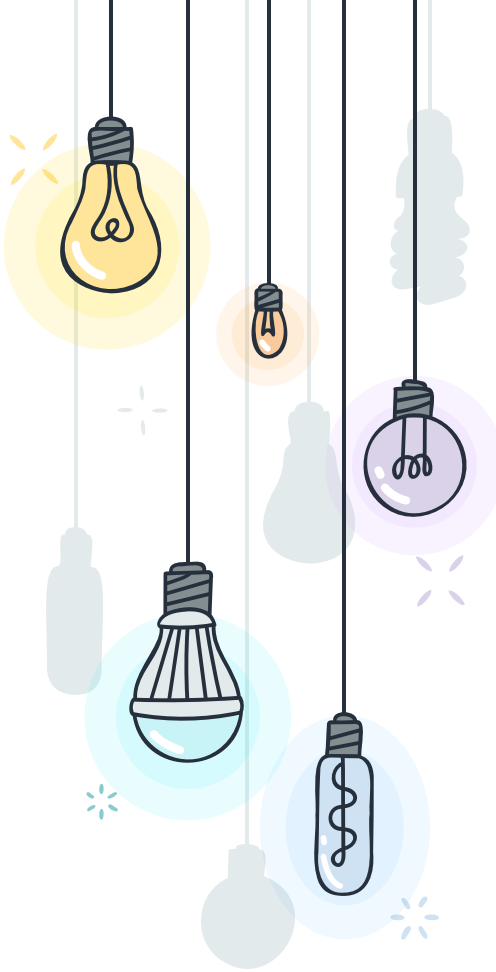

Admin
User
Web Camera
Train
model
Invalid
gesture
Gesture Window
DivyaAnuvadak
Input
gesture
Database
Valid
gesture
Output Window
Update dataset

Backend Flowchart

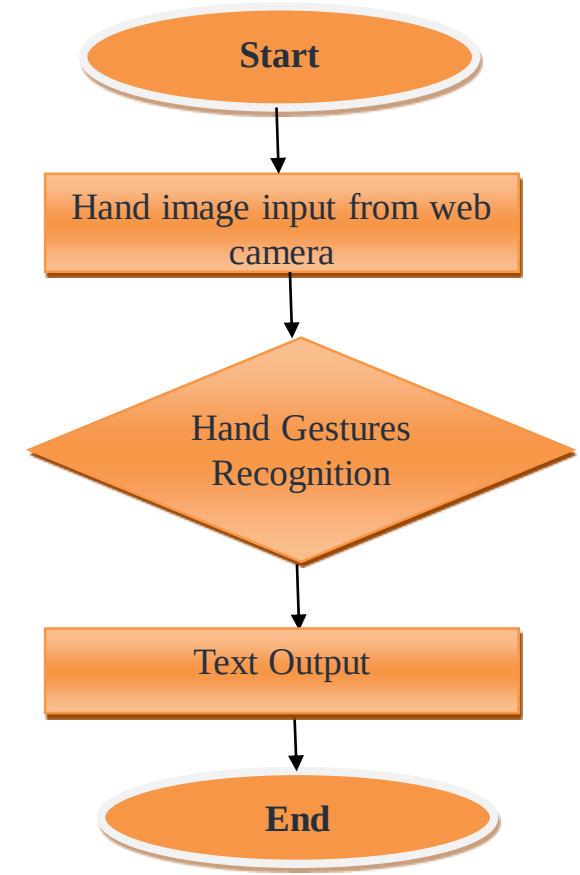


Testing

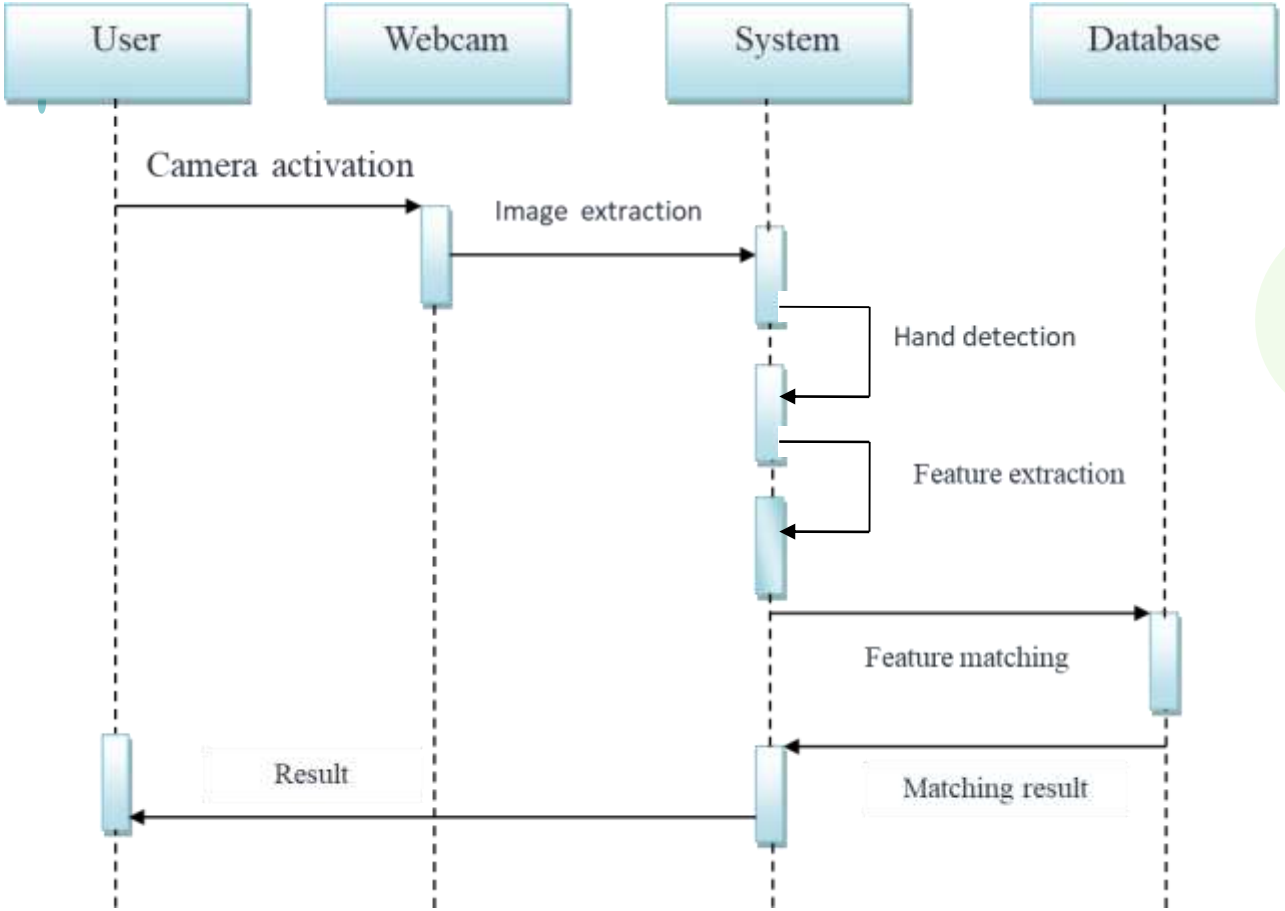




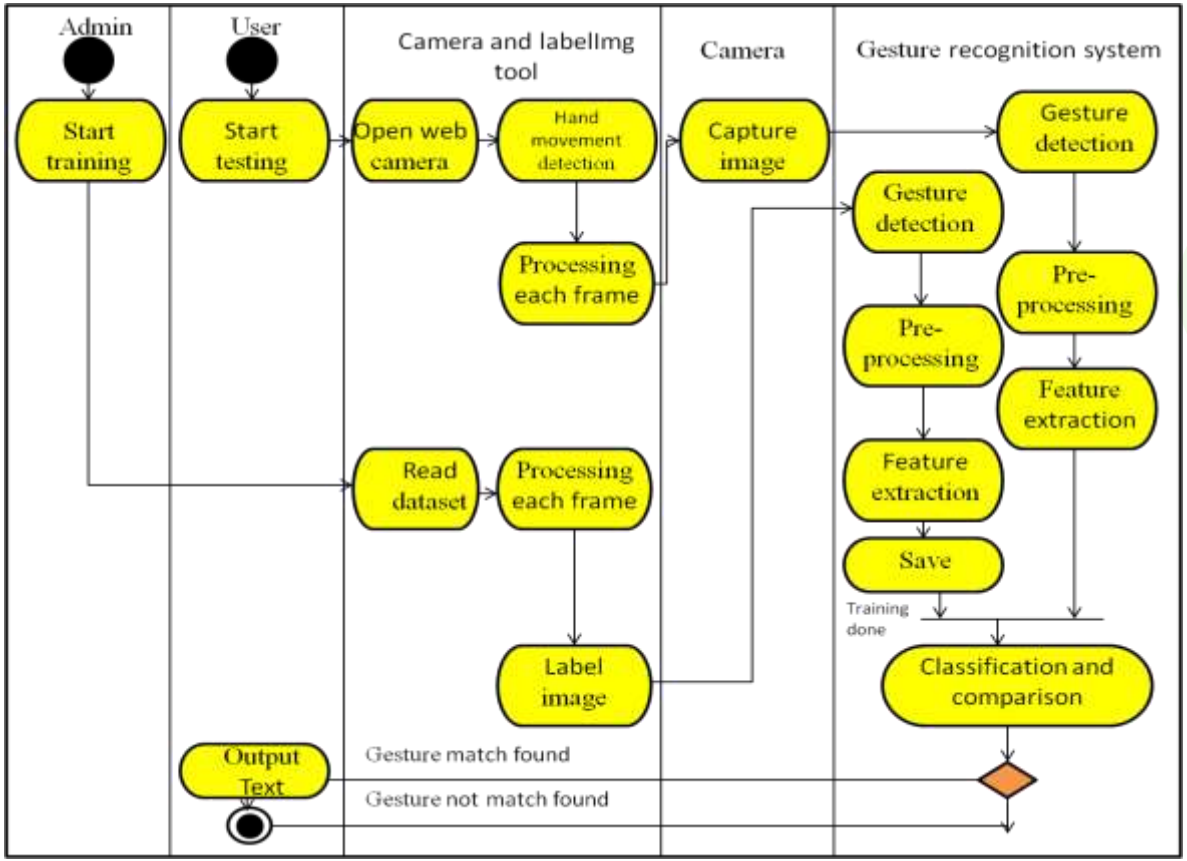
Frontend Flowchart



Sequence Diagram



Activity diagram



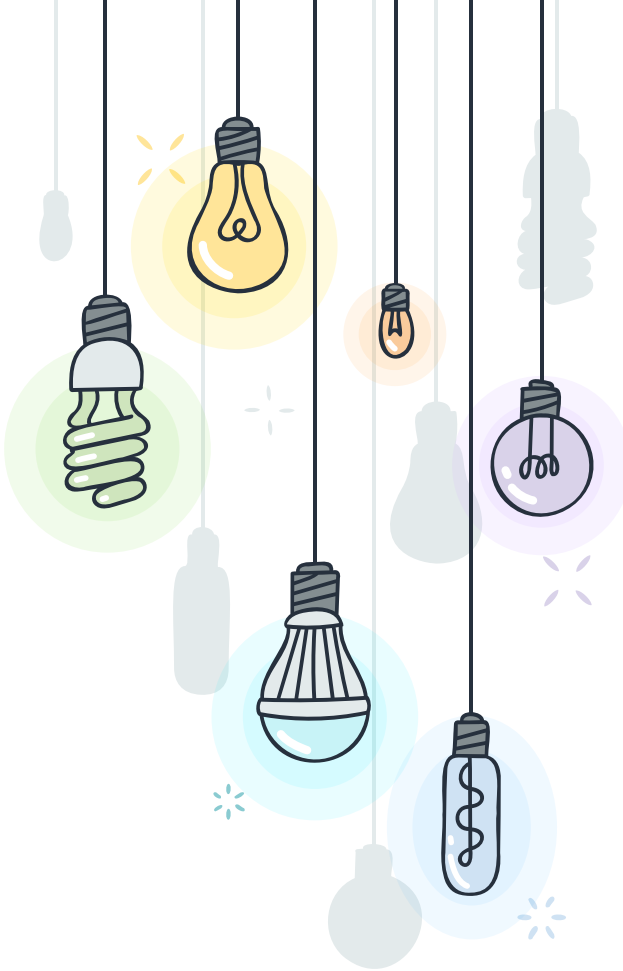
User Interface Design

Description of the user interface

- **Input –** Hand Gesture
- **Output –** Corresponding Text.

Processing Details

- The hand gestures given by the user through web-camera will be processed.
- Then it will be checked whether the input is valid or not.
- Eventually, the system would generate the corresponding output text in the text window.



Description of Component

Gesture Window

In this window, user would provide hand gesture as input which would be further verified and processed by the system.

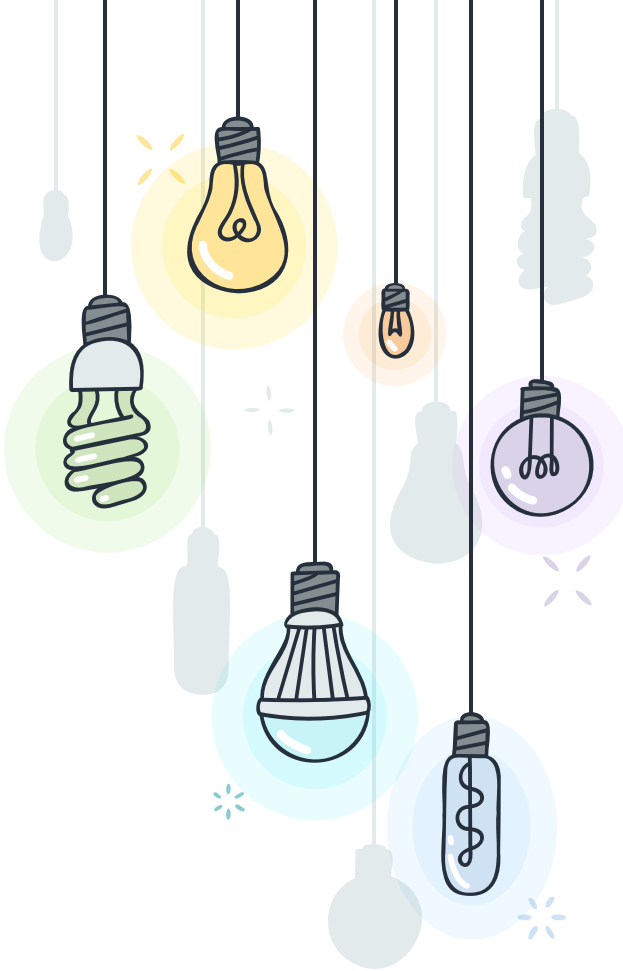
Text Window

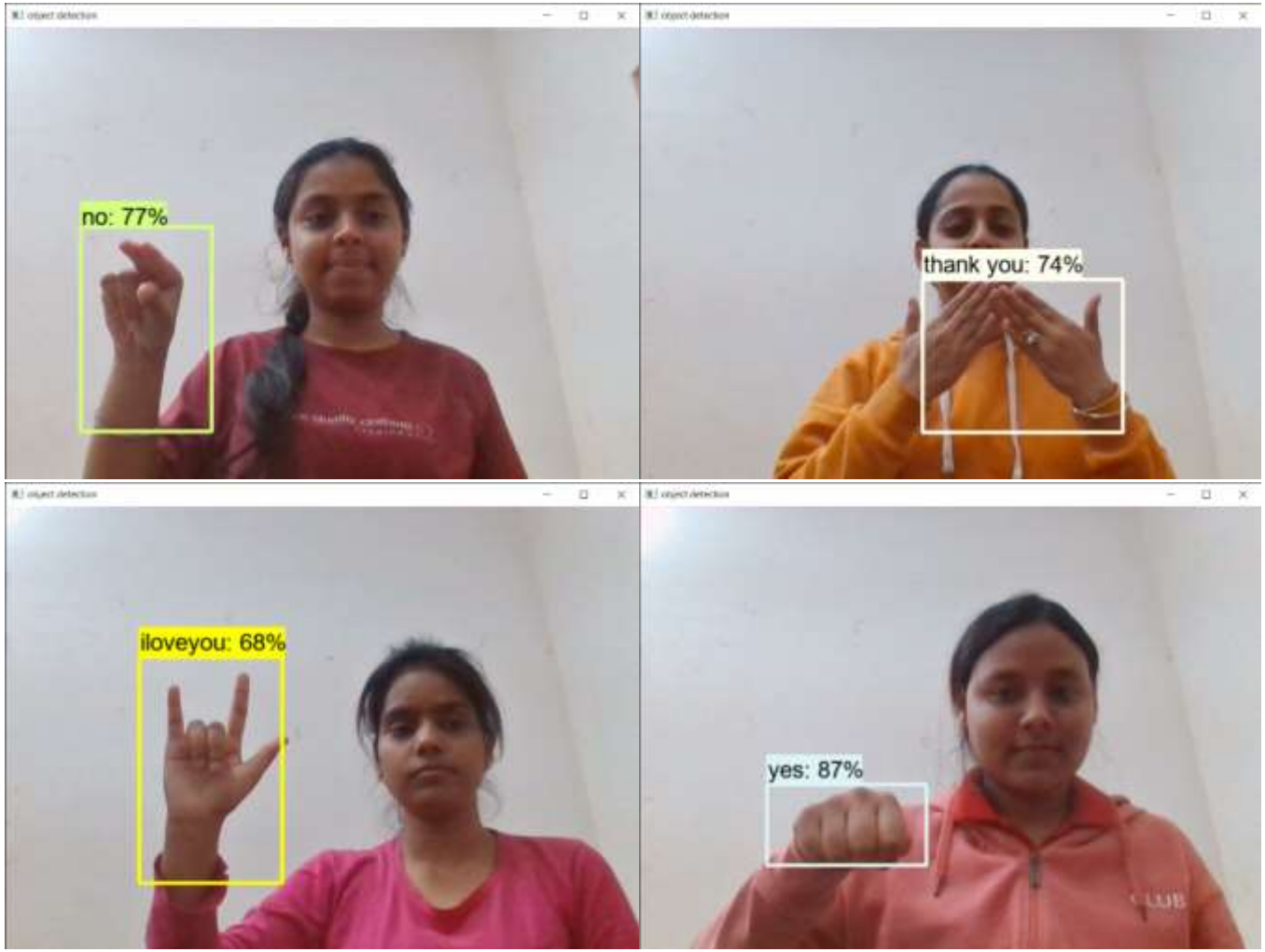
In this window, corresponding text will be displayed with respect to the valid hand gestures.

Screen Image

Objects and Actions

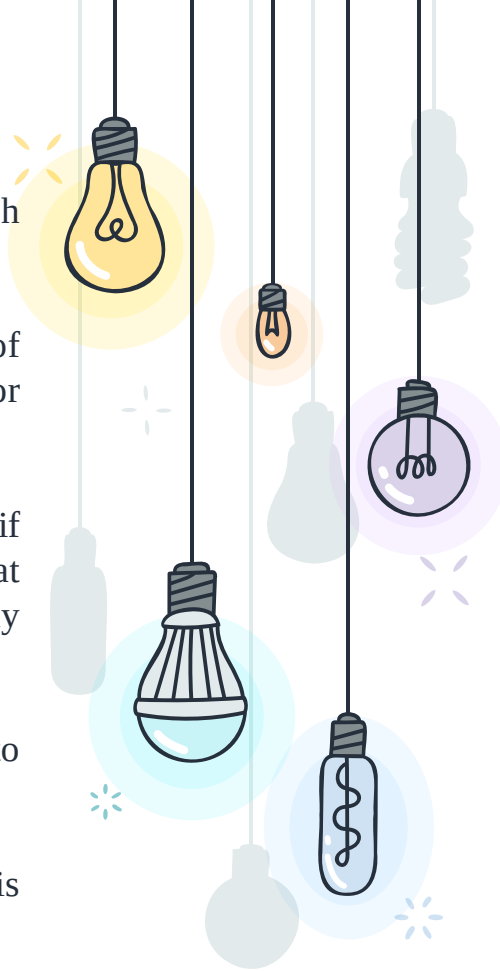
- **Object** – Textbox
- **Purpose** – To convert hand gestures into text
- **Action** – Gesture
- **Response** – Corresponding text output of gesture




no: 77%
thank you: 74%
iloveyou: 68%
yes: 87%

Types of Tests

- **Unit Tests:** This test is applied on each of the module to find whether or not each module is properly working or not.
- **System Tests:** Black box testing or system testing involves the external working of the software opposite to white box testing which checks for the internal working or code of a software application.
- **Integration or Regression Tests:** After unit test integration testing is done to test if the entire system works together correctly. While regression testing tries to ensure that a newly added feature or modified code does not break any functionality already working in the existing system.
- **Stress Tests:** Here we provide our software with some unfavorable conditions to check how they perform in such conditions.
- **Acceptance Tests and Staging:** This test will ensure that the quality of the product is not compromised.



References

- https://docs.streamlit.io/
- "A Concise Introduction to Software Engineering" by Pankaj Jalota
- Pressman Roger S., "Software Engineering - A Practitioner's Approach" Fifth Edition, McGraw-Hill Publication, 2000
- IEEE STD 830-1998, IEEE Recommended Practice for Software Requirement Specifications
- https://ieeexplore.ieee.org/document/720574

THANK YOU