Fayoum University

كلية الحاسبات والذكاء الاصطناعى
Faculty of Computers and Artificial Intelligence
كلية معتمدة

Sign-Talk

Ahmed Mohammed Ahmed

Omar Essam EL-Dien

Ziad Ahmed Abd EL-Nour

Ahmed Ihab Ali

Khaled Gamal EL-Sayed

Yasmeen Khaled Qotb

Ahmed Salah Ibrahem

Shadan Khaled Ibrahim

Supervised by: Dr.Esraa El-Hariri , Dr.Asmaa Hashem

Acknowledgement

Nothing could be done without God’s permission, and no success could be gained without his mercy. Thanks to Dr. Esraa M. El-Hariri for her very generous support and encouragement to accomplish this project in a professional and valuable way. She provided us with invaluable advice and helped us in difficult periods, her motivation and help contributed tremendously to the successful completion of the project. Thanks to Dr. Asmaa Hashem for her valuable help.

Abstract

Deaf and dumb Their world could seem silent but it's lively and filled with events. Communication is the only thanks to affect people, but the deaf and dumb face difficulty in handling ordinary people. Dumb and deaf people using sign language can find it difficult to connect with standard people. They can't understand sign language, so there's a barrier and gab between the people that disability and therefore the ordinary people.

SIGN-TALK To overcome the matter using smart gloves to assist them to communication between the people that disability (deaf and dumb) and therefore the ordinary people. The smart glove is an electronic device which may translate the sign hand gestures (sign languages) into text that the traditional person can understand, by which smart glove devour the gesture of hand and translate it into text.

SIGN-TALK Approximately 61 samples of hand movements and accompanying sentences are available for this class **,** the dataset contains around 60 samples of hand gestures and corresponding sentences from dumb individuals , there are approximately 60 samples of hand movements and sentences from deaf individuals, the classification accuracy achieved by our machine learning model is 91%, demonstrating its effectiveness in understanding and interpreting gestures and speech from individuals using the smart glove.

SIGN-TALK the mobile development aspect involved creating a user-friendly interface and implementing various features using the Flutter framework with an MVC (Model-View-Controller) architecture. Additionally, we incorporated mode selection options to customize sentence generation based on the user's current environment (home, school, office).

Contents

Acknowledgement 2

Abstract 3

1 Introduction 7

1.1 overview 7
1.2 Problem Definition 7
1.3 Project Motivation 8
1.4 Project Objectives 9
1.5 Organization 9

2 Literature Review 10

2.1 Overview 10
2.2 Intelligent Approach Based on Deep Learning Using Gloves (IDLG) 10
2.3 Biomedical Signal Processing and control 11
2.4 Intelligent gloves: An IT intervention for deaf-mute people 11
2.5 The Sign Language Interpreting Gloves 11
2.6 Assistive Data Glove for Isolated Static Postures 12
2.7 Data Preprocessing 12

3 Project Planning 13

3.1 Introduction 13
3.2 Project Plan 13

3.2.1 Gathering Requirements 13
3.2.2 System Analysis 14
3.2.3 System Design 14
3.2.4 Implementation 14
3.2.5 Project Schedule Gantt Chart 15

3.3 Feasibility Study 17

3.3.1 Focus 17
3.3.2 Strategies 17
3.3.3 KPIs 18

4 Proposed System 19

4.1 Overview........ 19
4.2 proposed framework 20
4.3 System Analysis........ 21

4.3.1 Process Modeling........ 21
4.3.2 Requirements 23
4.3.3 Use Case Diagrams........ 24
4.3.4 Use Case Scenario 25
4.3.5 Activity Diagram 27
4.3.6 Non-Functional Requirements........ 28

4.4 System Design 29

4.4.1 Sequence Diagram 29
4.4.2 Relation schema........ 30
4.4.3 Relational Database Diagram (ERD)........ 31

4.6 Understanding the Data Model 32

5 Experimental Results and Comparative Analysis........ 34

5.1 Overview........ 34
5.2 Evaluation Metrics........ 34
5.3 Results of The System 35

5.3.1 Dataset Description........ 35

6 System Development........ 40

6.1 Overview........ 40

6.2 Methodological assumptions........ 40

6.2.1 User Requirement 41
6.2.2 System Requirements........ 41

6.3 Used Technologies........ 41

6.3.1 Dart 41
6.3.2 Flutter........ 41
6.3.3 Arduino Uno 42

6.4 Embedded Development........ 43
6.5 Mobile Development........ 45
6.6 Backend-Machine Learning........ 54
6.7 Data Models........ 58

6.7.1 Technologies Used........ 59
6.7.2 Data Models........ 60

6.8 API Endpoints........ 61
6.9 Integration........ 64

7 Conclusions & Future Work **66**
Conclusions 66
Future work 68

Bibliography **70**

Chapter 1 Introduction

1.1 overview

In an era defined by technological innovation and a commitment to inclusivity, the concept of smart gloves has emerged as a revolutionary solution, redefining the boundaries of communication for individuals with hearing impairments. These innovative wearables represent a convergence of cutting-edge hardware and sophisticated software, designed to empower the deaf community by providing an intuitive and independent means of expression. Smart gloves go beyond mere technological advancement; they embody a transformative shift in addressing the communication challenges faced by deaf individuals, particularly those born to hearing parents who may not be proficient in sign language. This technological marvel, often comprising a strategic combination of flex sensors, accelerometers, and Bluetooth connectivity, stands as a testament to the power of innovation in fostering autonomy. As these gloves seamlessly translate intricate gestures into text and speech, they not only break down communication barriers but also challenge societal stigmas attached to the hearing-impaired.

1.2 Problem Definition

The two main forms of traditional communication are hearing and speaking. However, deaf people often have severe communication hurdles, which can result in social marginalization, loneliness, and limited chances for personal relationships, career advancement, and education.

Several things cause this communication gap:

1. Language barriers: the primary language for deaf people is sign language, which is not widely understood by hearing people, this gap of communication leaves deaf people excluded from mainstream conversation and decision-making.
2. Inaccessible environments: Public places, Workplaces educational places, etc. These places are not accessible for deaf people due to the absence of sign language.
3. Limited opportunities: The lack of communication often leads to missed opportunities, which makes it hard for them to find employment, or at least get a higher education.
4. Social isolation: The inability to communicate effectively can lead to feelings of isolation, loneliness, and hard to form meaningful relationships.

1.3 Project Motivation

We believe that life is about helping others and giving back to our society! And we also believe in the power of technology and using it in this project could help patients to have more time. Being a human is about meeting new people, making new memories, and being with the ones who care about you, and you care about them. Imagine that you can help someone and give him a method to overcome his disability. You help a father to keep communicating with his family. Making it easier for caregivers to take care of their loved ones. However, the proposed mobile application can determine a sentence after a specific hand motion.

1.4 Project Objectives

The main objectives of this project are to be accomplished within a timeline and with available resources by:

1. Developing technology to close the gap between deaf people and hearing people.
2. Empower deaf individuals to communicate independently without relying on interpreters or written communication by:
    - Creating a low-cost system that uses Smart Gloves to provide speech and text to voiceless People.
    - Providing a mobile application that connects to the glove via Bluetooth, allowing them to converse Wirelessly and easily.

1.5 Organization

We have structured the rest of this project as follows: Chapter(3) presents Related work including a brief history of some international Papers and their establishment, Chapter(4) presents the planning Phase is the fundamental process of understanding why a system should be built and also determining how the project team will go about building the system, Chapter(5) presents project analysis and reviews the Development methodology, the Agile Software Development Life Cycle also reviews the functional and non-functional requirements.

Chapter 2 Literature Review

2.1 Overview

This chapter includes a brief history of some international Papers and their establishment, the technology involved in improving and facilitating deaf people's communications, what problems happen and what they reach and their limitations, and references.

This project presents:

- Used a Machine learning approach to recognize and interpret hand gestures.
- Embedded system components used.

2.2 Intelligent Approach Based on Deep Learning Using Gloves (IDLG)

This paper explores the use of variables from flex and accelerometer sensors. The results introduce new types of parameters that can be utilized by sensors during neural training, creating a solid foundation for continued research. The technologies used include flex sensors, accelerometers, ARDUINO UNO, microcontrollers, and deep learning.

Data were collected from flex sensors and accelerometers, resulting in a dataset with 27 classes: 26 representing ASL alphabets and one representing a custom gesture for space. In this study, a multilayer neural network easily outperforms a “K-nearest neighbors” classifier, achieving excellent accuracy. While a “support vector machine” classifier performs poorly, a “decision tree” classifier performs well, but only with small data sets.

2.3 Biomedical Signal Processing and control.

They made a new architecture based on the old one (RNN) called GRU (Gate Recurrent Unity), they used a couple of sensors, on one hand, they used 5 flex sensors, a three-axis accelerometer, and a three-axis gyroscope, the dataset was collected from ten volunteers and it was separated 80% training and 20% test, the best result for manual feature extraction is for the classifier of type (KNN) with accuracy 88.5% for 25 feature and 96.8% for non-manual (GRU).

2.4 Intelligent gloves: An IT intervention for deaf-mute people

The researchers invented Intelligent Gloves (IG), a prototype of a two-way communication glove developed to facilitate communication between deaf-mutes and non-deaf-mutes. The Intelligent Gloves (IG), a prototype of a two-way communication glove, was developed to facilitate communication between deaf-mutes and non-deaf-mutes, which consists of a pair of gloves, flex sensors, an Arduino nano, a screen with a built-in microphone, a speaker, and a SD card module, the data they used is the signals that were generated from the sign language and started to process it and transform this data into text, at the end the result they got is making deaf-mute people easily communicate with society and express their feelings, and people could understand them easier. They used the signals that were generated from the sign language and started to process it and transform this data into text. They make deaf-mute people hope to communicate with society and easily express their feelings, and they make people not understand the sign language to understand the deaf-mute people.

2.5 The Sign Language Interpreting Gloves

The problem is significant, with a majority of deaf children born to parents lacking sign language proficiency, leading to communication challenges and strained relationships. Societal misconceptions persist, especially in regions like the Middle East, where 3% of the population faces hearing loss. The objective is clear: technology, specifically smart gloves, aims to empower deaf individuals by enabling independent communication without interpreters. The methodology involves a mixed-method approach, combining hardware and software to recognize and display American Sign Language (ASL). The hardware includes flex sensors and an MPU6050 rotation sensor, yielding promising results in detecting ASL and converting it into text and speech. This breakthrough fosters independence and communication between speech-impaired individuals and the hearing community.

2.6 Assistive Data Glove for Isolated Static Postures Recognition in American Sign Language Using Neural Network

Neural network-based sign language recognition model using an assistive glove. The glove is designed with sensors attached to each finger and palm to capture gesture data. A microcontroller is used to analyze the sensor values corresponding to different American Sign Language (ASL) postures. The author analyzes the impact of different activation functions on the performance of the neural models. The assistive glove can capture real-time data to create datasets for numeric, alphabet, and alphanumeric signs. The neural network models show promising accuracy for the different datasets. In the future, the author plans to collect more ASL samples using the glove and train other models to improve sign language recognition. The assistive glove and neural network model provide a cost-effective solution for sign language recognition compared to systems with many sensors.

2.7 Data Preprocessing

Data Acquisition and Preprocessing Phase

Data Acquisition

The data acquisition phase involves gathering the raw data required to train the machine learning model. In this project, the data consists of sign language gestures captured using the smart glove. The smart glove is equipped with sensors that capture various parameters such as finger positions, hand orientation, and movement patterns. These parameters are essential in distinguishing between different sign language gestures.

Steps in Data Acquisition:

- Sensor Calibration: Before collecting data, ensure that all sensors in the smart glove are calibrated correctly to provide accurate readings.

- Data Collection Sessions: Conduct multiple data collection sessions where individuals perform sign language gestures. Ensure a diverse set of participants to account for variations in gesture performance.

- Recording Data: For each gesture, record the sensor data. This includes:
    - Finger bend angles (from flex sensors)
    - Hand orientation and movement (from accelerometers and gyroscopes)
    - Any additional relevant parameters captured by the glove

- Data Labeling: Label each recorded data sequence with the corresponding gesture name. This labeling is crucial for supervised learning where the model learns to associate sensor data with specific gestures.

- Data Storage: Store the collected data in a structured format, such as CSV or JSON files, ensuring that each entry includes both the sensor readings and the corresponding gesture label.

Data Preprocessing

Data preprocessing is a critical step in preparing the raw data for training the machine learning model. This phase involves cleaning the data, transforming it into a suitable format, and performing any necessary augmentation to improve model performance.

Steps in Data Preprocessing:

- Data Cleaning:
    - Removing Outliers: Identify and remove any outlier readings that may result from sensor malfunctions or incorrect gesture performance.
    - Handling Missing Values: Address any missing values in the dataset, either by imputing them with appropriate values or removing the affected entries.

- Normalization:
    - Normalize the sensor data to a consistent scale, ensuring that all features contribute equally to the model training process. This can be done using techniques such as min-max scaling or z-score normalization.

- Segmentation:
    - Segment the continuous sensor data into discrete time windows or frames. Each frame represents a snapshot of sensor readings over a short time interval, allowing the model to capture temporal patterns in the gestures.

- Feature Extraction:
    - Extract relevant features from the raw sensor data. This may include calculating statistical measures (mean, standard deviation) for each sensor reading over the segmented frames or deriving new features that capture the dynamics of the gestures.

- Dimensionality Reduction:
    - Apply dimensionality reduction techniques, such as Principal Component Analysis (PCA), to reduce the number of features while retaining the most informative aspects of the data. This step helps in reducing computational complexity and mitigating the risk of overfitting.

Data Postprocessing

After the model makes predictions based on the preprocessed data, postprocessing steps are necessary to refine the results and ensure they are interpretable and actionable.

Steps in Postprocessing:

- Smoothing Predictions:
    - Apply smoothing techniques, such as moving average filters, to the predicted gesture sequences to eliminate any spurious or noisy predictions. This step ensures that the output gestures are consistent and smooth over time.

- Thresholding:
    - Implement confidence thresholding to filter out low-confidence predictions. By setting a confidence threshold, only predictions with a confidence score above the threshold are considered valid, reducing the likelihood of incorrect gesture recognition.

- Gesture Mapping:
    - Map the predicted gestures to the corresponding words or phrases in the target language. This involves creating a lookup table or dictionary that translates recognized gestures into meaningful words.

- Error Handling:
    - Develop error handling mechanisms to manage cases where the model fails to recognize a gesture or produces ambiguous results. This may involve prompting the user to repeat the gesture or providing alternative suggestions.

- Real-Time Integration:
    - Integrate the postprocessed results into the real-time system, ensuring that the recognized words or phrases are communicated promptly and accurately. This step may involve interfacing with text-to-speech systems or display devices to convey the recognized words to the user.

By following these steps, the data acquired from the smart glove can be effectively preprocessed and postprocessed, enabling accurate and reliable gesture recognition for sign language communication.

Chapter 3

Project Planning

3.1 Introduction

Project management is a process of planning and controlling the development of a system within a specified time frame with the right functionality, and define Project Focuses, Strategies, and Action Steps, and measuring progress and performance.

3.2 Project Plan

The planning Phase is the fundamental process of understanding why a system should be built and also determining how the project team will go about building the system.

3.2.1 Gathering Requirements

The analysis of this information leads to the development of a concept for a new system.

3.2.2 System Analysis

- functional and non-functional requirements
- (Data flow diagram) DFD
- Use Case

3.2.3 System Design

- Sequence diagram
- Class diagram
- Activity diagram
- Context diagram

3.2.4 Implementation

- Embedded system
- ML model
- Back-end
- Mobile Application

3.2.5 Project Schedule Gantt Chart

A Gantt chart is a horizontal bar chart used in project management to visually represent a project plan over time.

Gantt charts typically show the timeline and status.

ID		Task Mode	Task Name	Duration	Start	Finish
1			**1-Design and Planning**	**10 days**	**Mon 10/2/23**	**Thu 10/12/23**
2			**1.1 Define Glove Functionality**	**2 days**	**Mon 10/2/23**	**Tue 10/3/23**
3			1.1.1 Brainstorm Glove Functinality	1 day	Mon 10/2/23	Mon 10/2/23
4			1.1.2 Sketch Glove Design	1 day	Tue 10/3/23	Tue 10/3/23
5			**1.2 Research Hardware Components**	**2 days**	**Mon 10/2/23**	**Tue 10/3/23**
6			1.2.1 Identify Suitable Sensors	1 day	Mon 10/2/23	Mon 10/2/23
7			1.2.2 Research for Bluetooth module for app connection	1 day	Tue 10/3/23	Tue 10/3/23
8			**1.3 Design App Functionality**	**5 days**	**Sun 10/8/23**	**Thu 10/12/23**
9			1.3.1 Define app screens	3 days	Sun 10/8/23	Tue 10/10/23
10			1.3.2 Design UI visualization and user interaction.	2 days	Wed 10/11/23	Thu 10/12/23
11			**2-Gathering Information**	**10 days**	**Sun 10/15/23**	**Thu 10/26/23**
12			2.1 Search For Common Places that Deaf-People went to	6 days	Sun 10/15/23	Fri 10/20/23
13			2.2 Analyis words that used a lot in the common places	4 days	Mon 10/23/23	Thu 10/26/23
14			**3-System Analysis**	**7 days**	**Sun 10/22/23**	**Mon 10/30/23**
15			3.1 DFD	1 day	Mon 10/23/23	Mon 10/23/23
16			3.2 Use Cases	1 day	Tue 10/24/23	Tue 10/24/23
17			3.3 Activity Diagram	1 day	Wed 10/25/23	Wed 10/25/23
18			3.4 System sequance diagram	1 day	Thu 10/26/23	Thu 10/26/23
19			3.5 Sequance Diagram	1 day	Fri 10/27/23	Fri 10/27/23
20			3.6 Class Diagram	1 day	Mon 10/30/23	Mon 10/30/23
21			**4-Prototyping**	**2 wks**	**Mon 2/12/24**	**Fri 2/23/24**
22			**4.1 Develop Glove Prototype**	**7 days**	**Mon 2/12/24**	**Tue 2/20/24**
23			4.1.1 Implement basic sensor data acquisition and transmission	3 days	Mon 2/12/24	Wed 2/14/24

Project: Sign-Talk
Date: Wed 5/15/24

Task; Split; Milestone; Summary; Project Summary; Inactive Task; Inactive Milestone; Inactive Summary; Manual Task; Duration-only; Manual Summary Rollup; Manual Summary; Start-only; Finish-only; External Tasks; External Milestone; Deadline; Progress; Manual Progress

Page 1

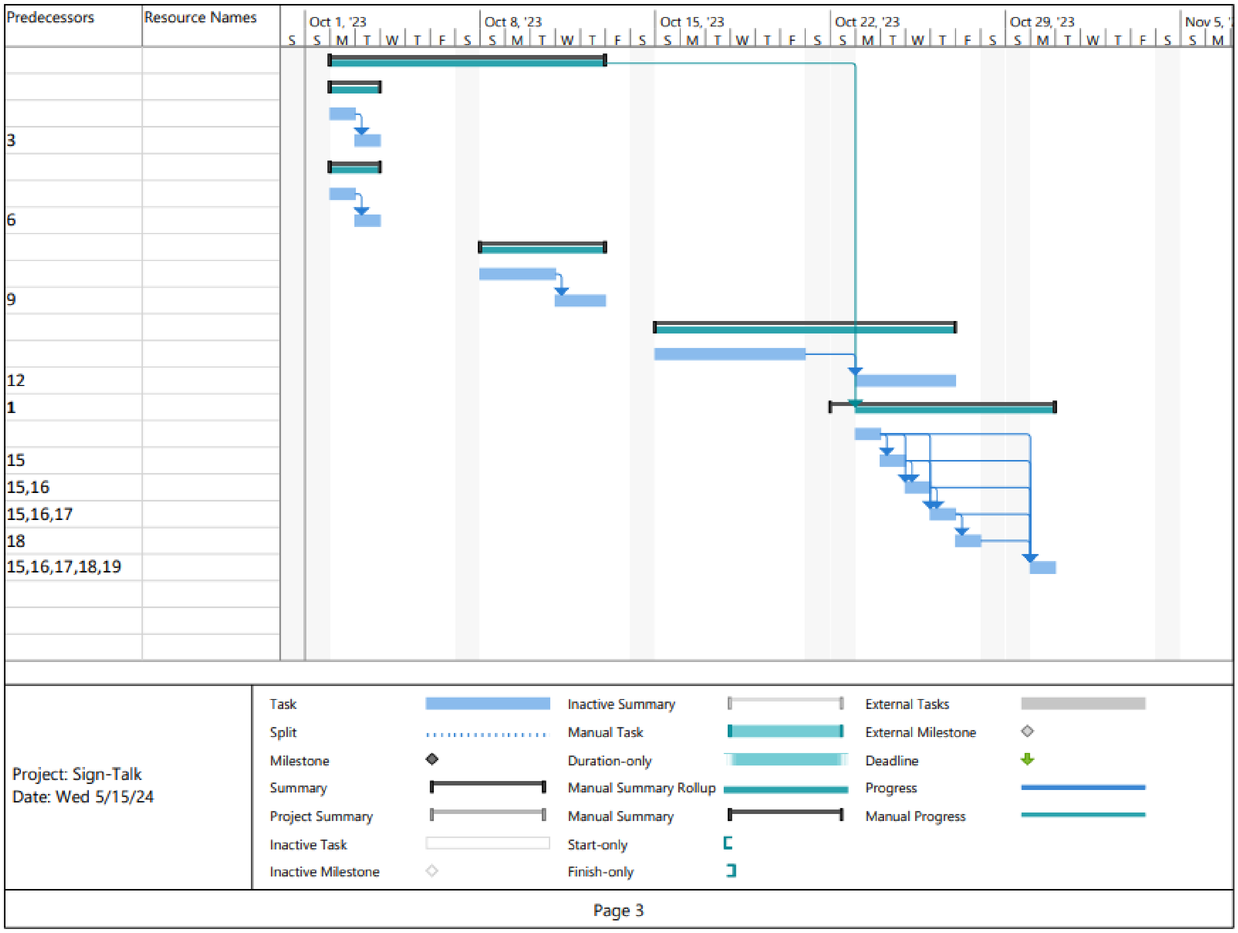
ID		Task Mode	Task Name	Duration	Start	Finish
24			4.1.2 Develop basic app functionality for data visualization and simple interactions	4 days	Thu 2/15/24	Tue 2/20/24
25			4.2 Do the UI/UX	3 days	Mon 2/12/24	Wed 2/14/24
26			4.3 Build Application to the connection between the glove and the mo	3 days	Mon 2/12/24	Wed 2/14/24
27			**5-Implemantaion**	**50 days**	**Sun 3/10/24**	**Thu 5/16/24**
28			5.1 Build Embedded Glove	10 days	Mon 3/11/24	Fri 3/22/24
29			**5.2 ML Module**	**10 days**	**Mon 3/25/24**	**Fri 4/5/24**
32			5.3 Build the Api request and the conection to the model	15 days	Mon 3/11/24	Fri 3/29/24
33			5.4 Build the Flutter Application	10 days	Fri 5/3/24	Thu 5/16/24
34			**6- Testing**	**5 days**	**Mon 11/27/23**	**Fri 12/1/23**
35			Test the project	2 days	Mon 11/27/23	Tue 11/28/23

Project: Sign-Talk
Date: Wed 5/15/24

Task	Inactive Summary	External Tasks
Split	Manual Task	External Milestone
Milestone	Duration-only	Deadline
Summary	Manual Summary Rollup	Progress
Project Summary	Manual Summary	Manual Progress
Inactive Task	Start-only	
Inactive Milestone	Finish-only	

Page 2

Project Schedule



3.3 Feasibility Study

The feasibility study outlines and analyzes several alternatives or methods of achieving business success.

3.3.1 Focus

Helping dumb and deaf.

3.3.2 Strategies

-Build a smart glove that help deaf and dumb people to facilitate their daily life.
-Create a Business Model.
-lunching Application.

3.3.3 KPIs

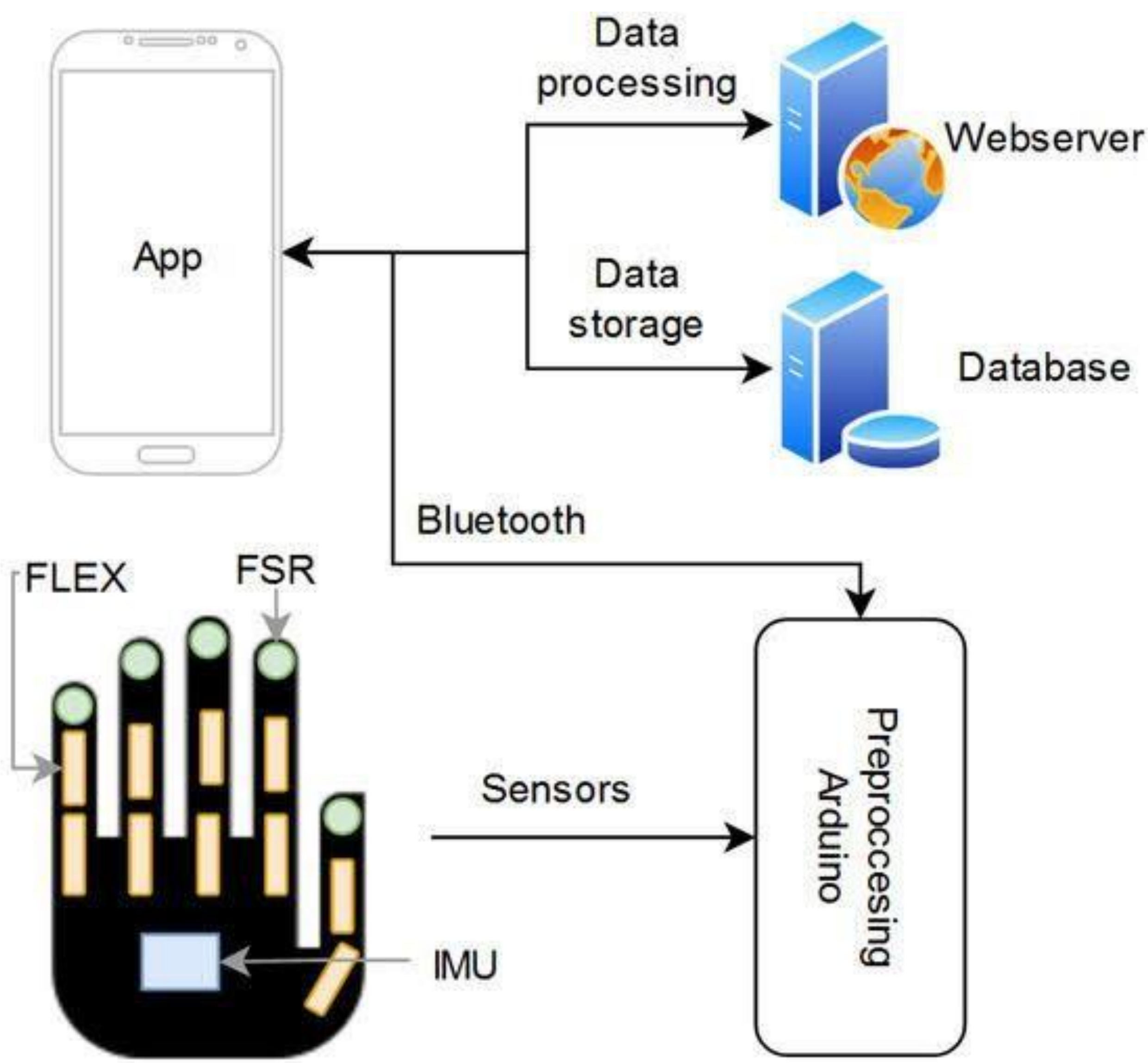
-Model classify sentence with moves correctly.
-API works properly with Mobile APP.

Chapter 4
Proposed System

4.1 Overview

This chapter reviews the Development methodology and also reviews the functional and non-functional requirements. It also discusses System analysis through the life cycle and its requirements showing the USE-CASE Diagram which is a list of actions or events steps typically defining the interactions between a role (known in the Unified Modeling Language as an actor) and a system to achieve a goal, Activity diagram which is a graphical representation of workflows of stepwise activities and actions and process modeling.

4.2 proposed framework



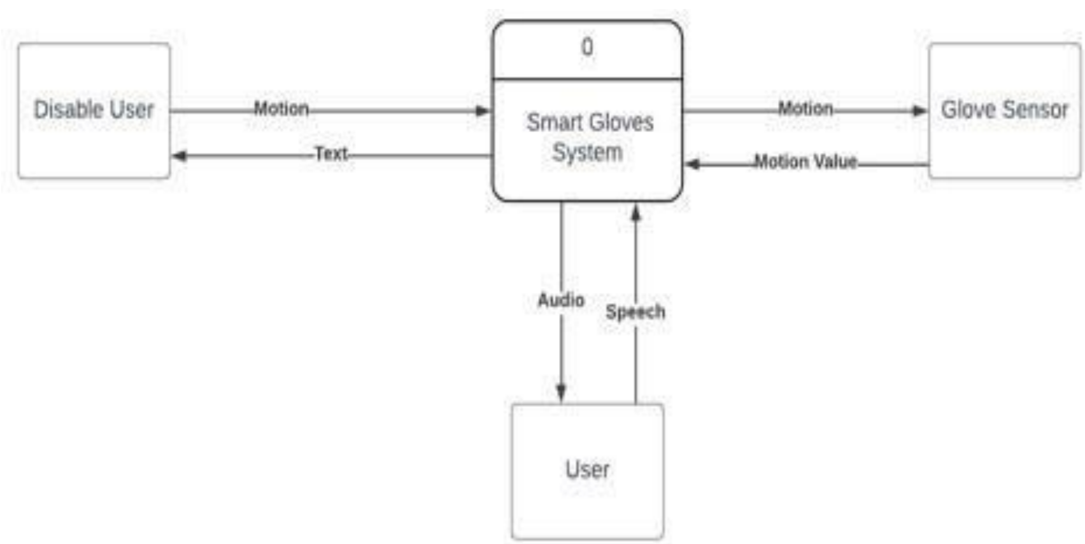
4.3 System Analysis

Through this chapter we are going through the project requirements that the system must satisfy are of two types, which are functional and non-functional requirements. Functional requirements are the requirements that define a function of the software that runs on the system. Non-functional requirements are the requirements that specify criteria that can be used to judge the operation of the system, rather than specific behaviors. In the following subsections, both functional and non-functional requirements of the proposed system are listed.

4.3.1 Process Modeling

Context Diagram

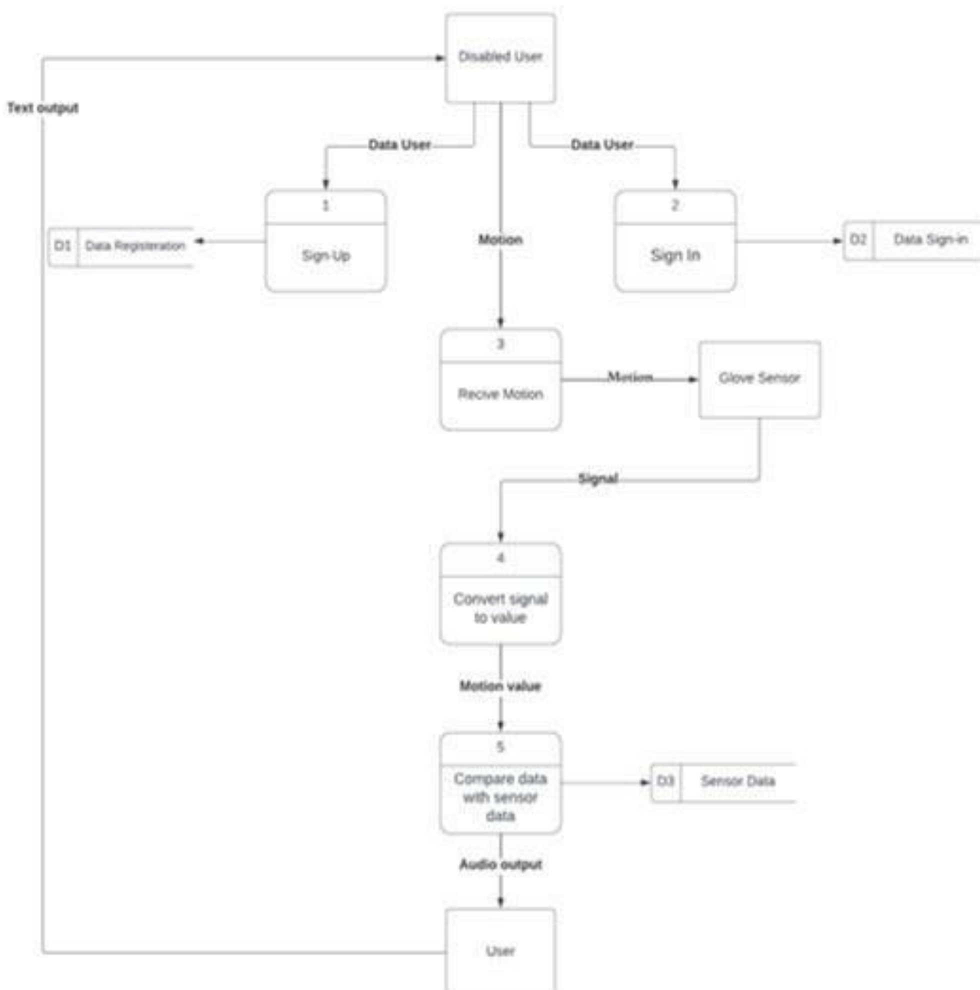
The Context Diagram shows the system under consideration as a single high-level process and then shows the relationship that the system has with other external entities.



Data Flow Diagram

A data flow diagram (DFD) illustrates how data is processed by a system in terms of inputs and outputs. As its name indicates its focus is on the flow of information, where data comes from, where it goes, and how it gets stored It provides an overview of:

- What data is the system process?
- What transformations are performed?
- What data are stored?
- What results are produced, etc.



.

4.3.2 Requirements

Functional Requirements

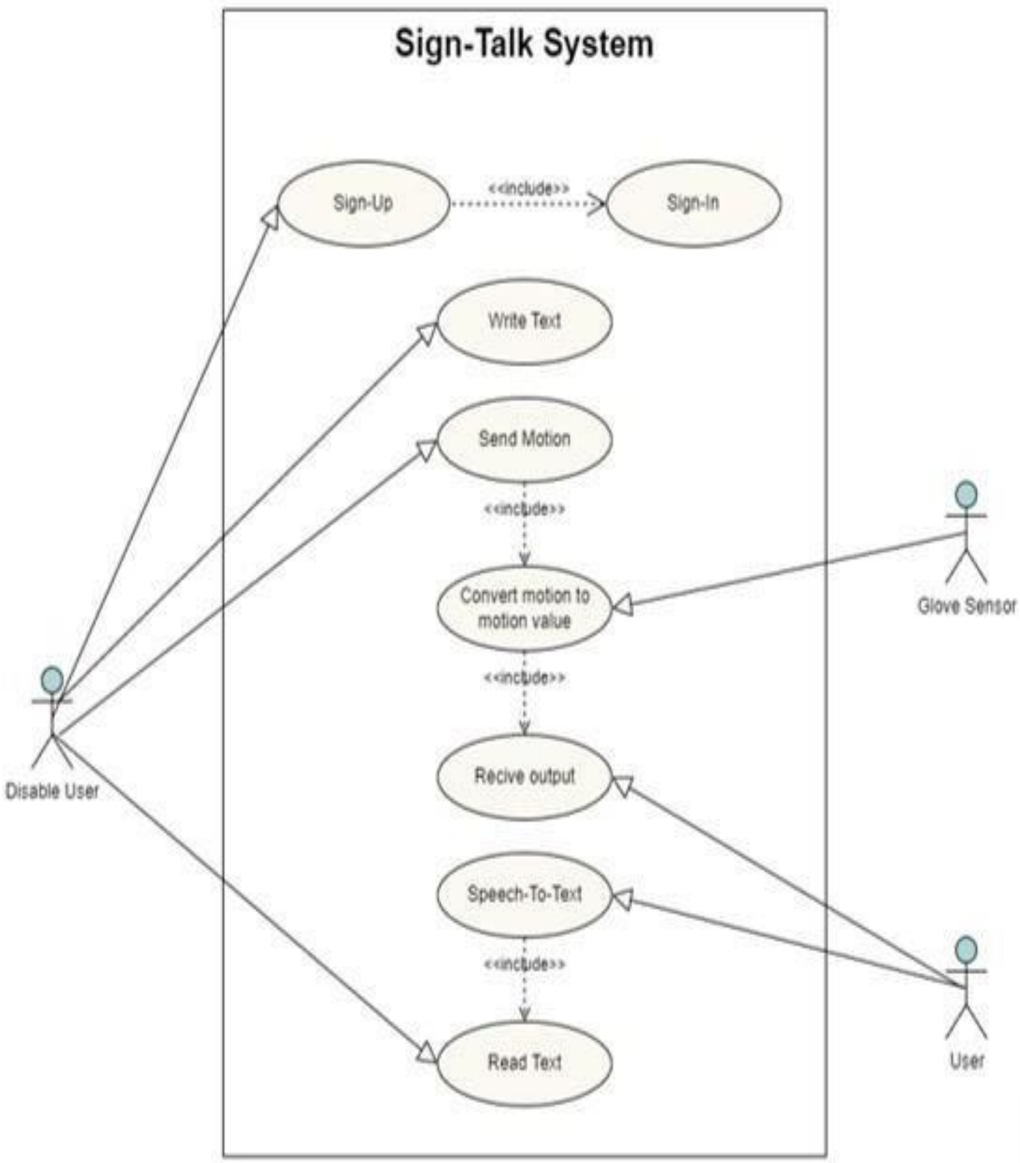
Functional requirements are features that allow the system to function as it was intended. Put another way, if the functional requirements are not met, the system will not work. Functional requirements are product features and focus on user requirements. Functional requirements are features that allow the system to function as it was intended. Put another way, if the functional requirements are not met, the system will not work. Functional requirements are product features and focus on user requirements.

4.3.3 Use Case Diagrams

• it represents system functionality from the user's perspective

• describes who will use the system and in what ways the user expects to interact with the system.

• represents the interactions between use cases and actors.



4.3.4 Use Case Scenario

A use case Scenario represents the sequence of events along with other information that relates to this use case. A typical use case specification template includes the following information:

• Description.
• Pre- and Post- interaction condition.
• Basic interaction path.
• Alternative pat.

<table>
<tr><td>Use Case Name:</td><td colspan="2">Sign Talk system</td></tr>
<tr><td>Actors:</td><td colspan="2">Disable User, User</td></tr>
<tr><td>Description:</td><td colspan="2">Disable user will send motion and User Will receive the text</td></tr>
<tr><td rowspan="2">Typical Course of Events:</td><td>Action Actors</td><td>System Response</td></tr>
<tr><td>Step 1: This use case is initiated
When the disabled user Sign in or sign up in application

Step 3: Disable user Will send motion from glove to system.

Step 7: User will receive output (voice and speech)

Step 8: User can send speech.

Step 10: Disable user will receive text.</td><td>Step 2: System Will login the user account

Step 4: System will response motion and return text from back end.

Step 5: Compare data from Api.
Step 6: App will receive response from system.

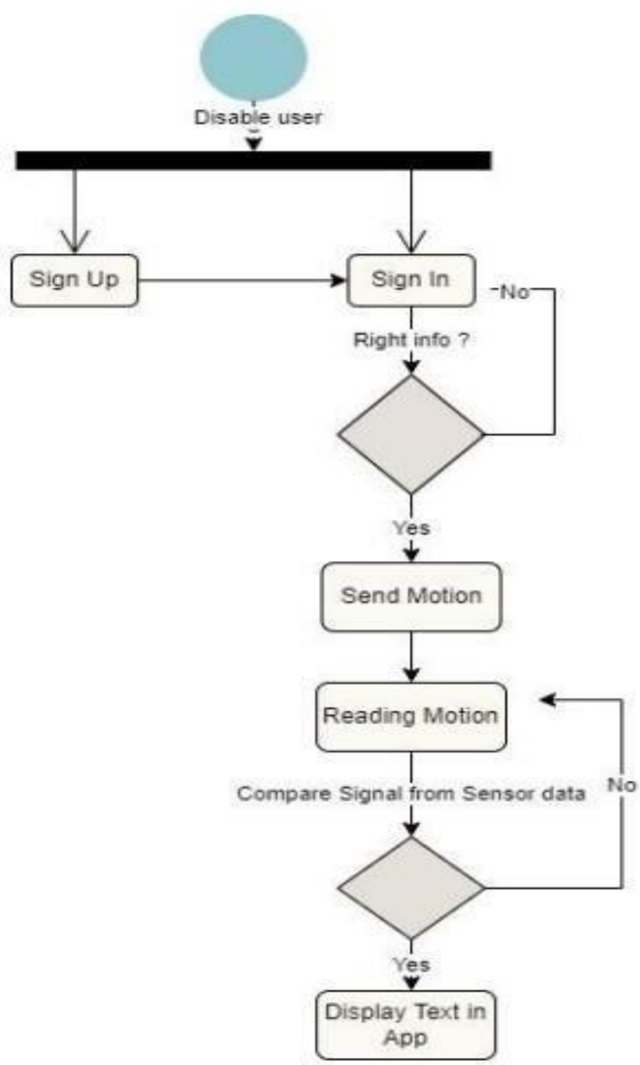
Step 9: system will convert speech to text</td></tr>
</table>

Alternate Courses:	Step 2: if the user doesn’t have an account, he will sign up. Step 6: if data doesn’t same with Api ‘s data, the sensor will read another signal
Precondition:	Signal that can only be sent by disabled user
Post condition:	Sensor will convert signal into data
Assumption:	None currently

4.3.5 Activity Diagram

An activity diagram gives a graphical representation of how data moves around an information system based on data flows.

- **Activity Diagram:**



4.3.6 Non-Functional Requirements

Non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behavior. They are contrasted with functional requirements that define specific behaviors or functions.

• Usability: The smart glove and mobile application should be easily usable by the user.

• Accessibility: the ordinary person will receive sentences and voice from the application after a deaf or dumb person makes a specific move with the glove.

• Performance: Performance: The Main function must have specific real-time to be executed for each operation in order not to delay the whole system.

• Security: The dumb and deaf must be sure that all their details are secure in the App options.

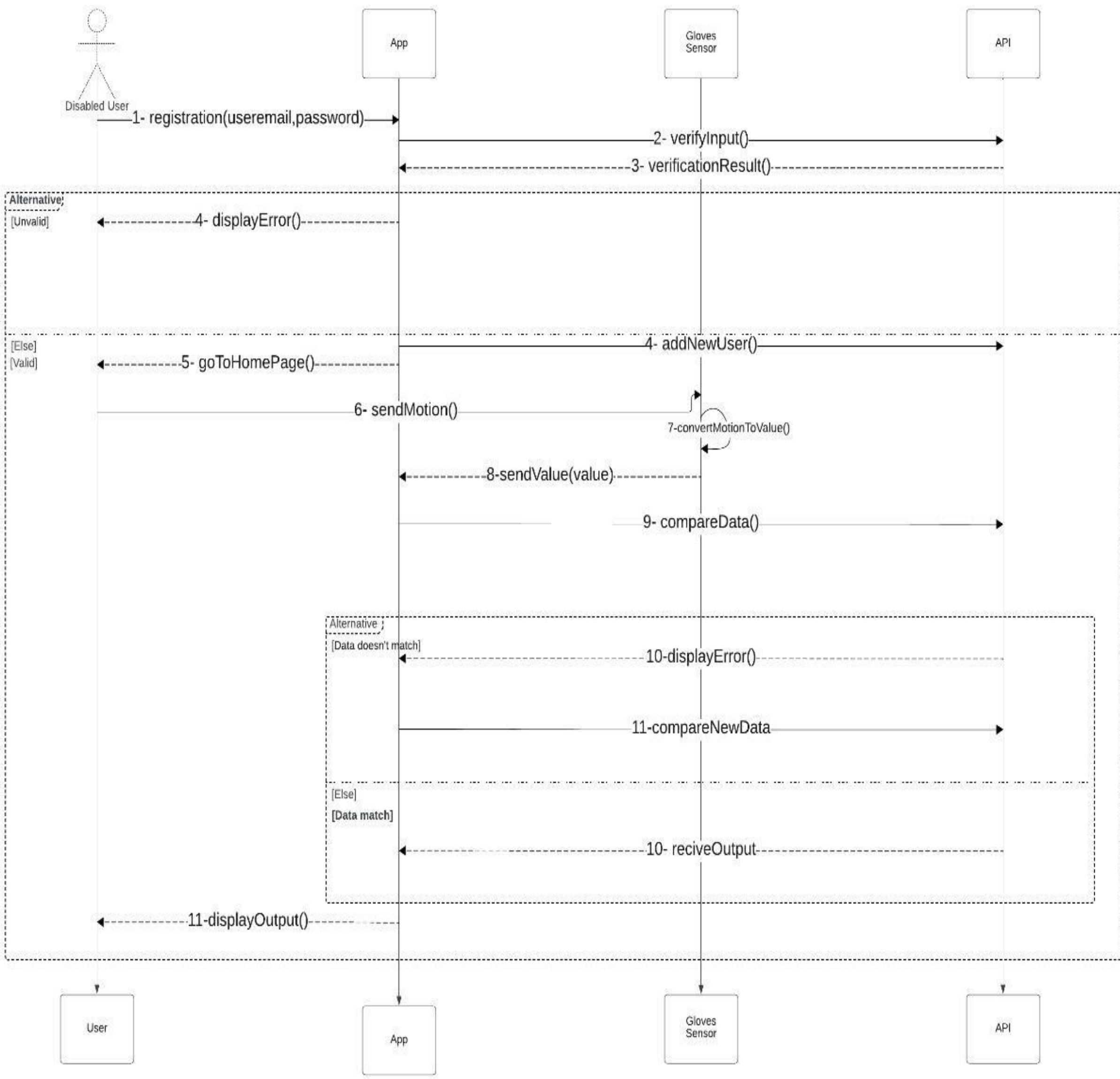
• User-friendly: Easy to use and manipulate every proposed feature.

4.4 System Design

Systems design is the process of defining elements of a system like modules, architecture, components and their interfaces, and data for a system based on the specified requirements.

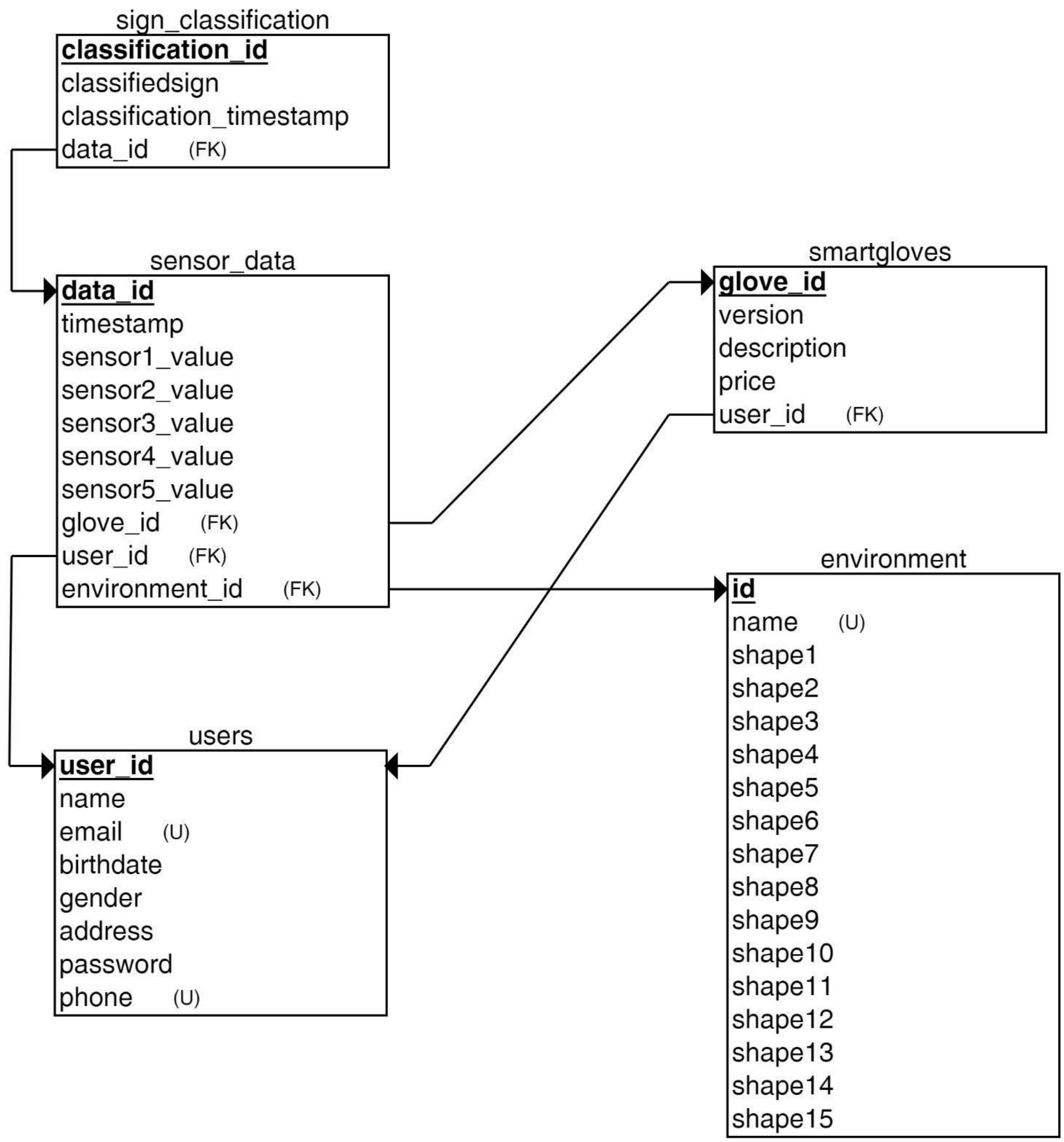
4.4.1 Sequence Diagram

sequence diagram given is used to show the interactive behavior of the system



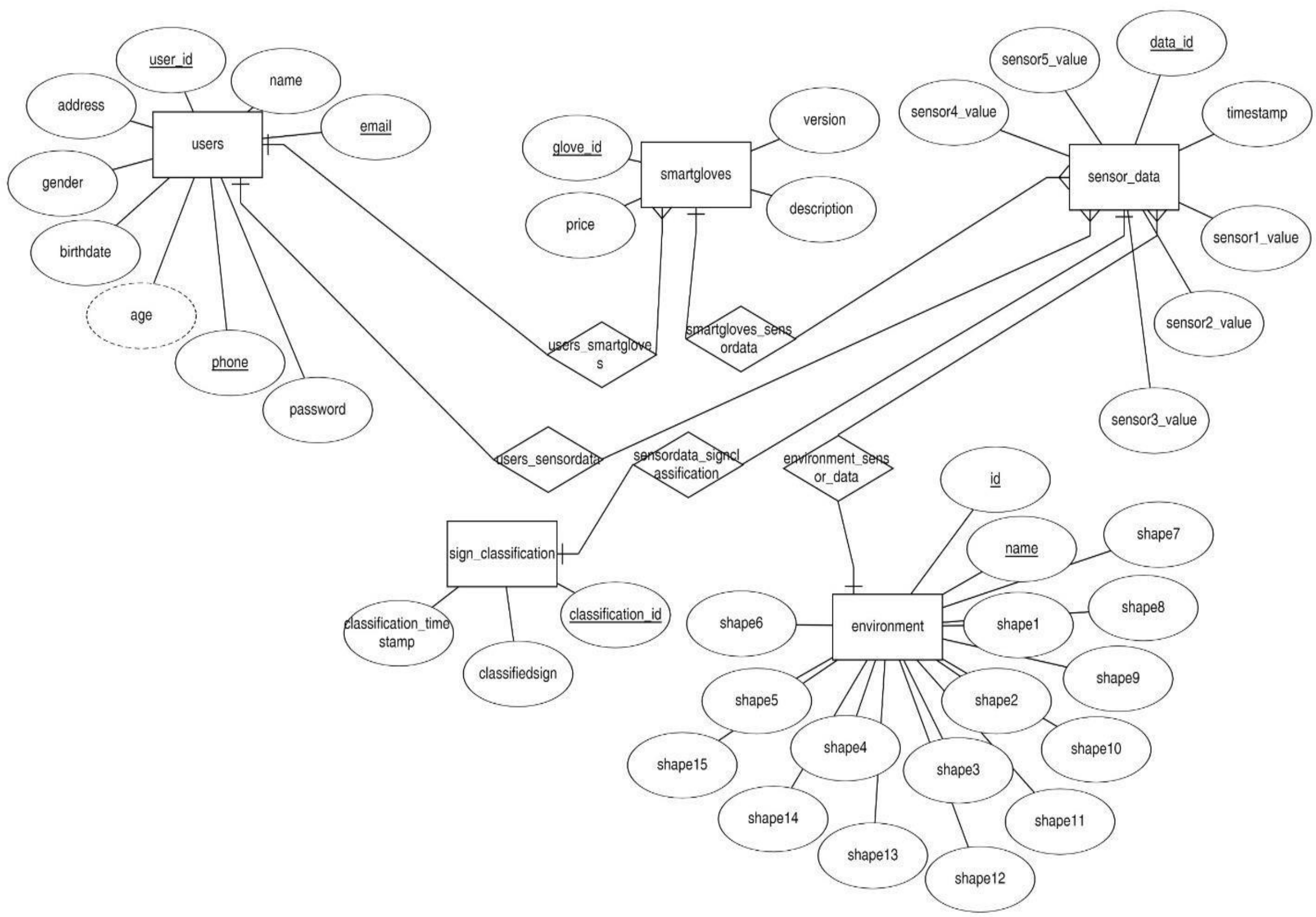
4.4.2 Relation schema

Defines the design and structure of the relation as it consists of the relation's name, set of attributes/field names/column names. Every attribute would have an associated domain.



4.4.3 Relational Database Diagram (ERD)

An entity-relationship model (ERM) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion.



4.7 Understanding the Data Model

Your data model is structured around three distinct environments: Home, Hospital, and Street. Each environment has symbols (like A, B, C, etc.) mapped to specific phrases. This mapping allows your application to provide relevant responses based on the current context or environment.

Data Model Explanation

Symbols and Phrases

- Symbols (A to N): Each symbol represents a unique identifier that corresponds to a specific phrase or message.
- Phrases in Different Environments:
  - Home: Phrases tailored for home-related situations.
  - Hospital: Phrases relevant to interactions in hospital settings.
  - Street: Phrases suited for outdoor or public environments.

Example Usages

- Symbol A: In the Home environment, symbol A maps to "I need medicine." This phrase indicates a request related to healthcare needs within a home context.
- Symbol B: Across all environments, symbol B simply maps to "Waiting...," suggesting a general state of waiting or patience required.
- Symbol C: In the Hospital environment, symbol C is mapped to "I need help with mobility," addressing a specific need related to movement or assistance within a medical facility.

Usage in .NET ML Project

Predictive Capability

- Input and Output: Given an input symbol or sequence of symbols, your .NET ML model predicts the corresponding phrase or message the deaf individual intends to convey.
- Model Training: The model is trained on labeled data where symbols are input features, and the associated phrases are target labels. This training enables the model to learn patterns and relationships between symbols and their corresponding messages across different contexts.

Deployment and Interaction

- Real-time Interaction: In real-world scenarios, the ML model deployed in your application can interpret gestures or inputs from a deaf user and generate text-based responses using the mapped phrases from your data model.
- Adaptability: The data model can be updated or extended to include new symbols and phrases, accommodating additional expressions or evolving needs of users over time.

Conclusion

Your .NET ML project leverages a structured data model that maps symbols to phrases across multiple environments, facilitating accurate prediction of messages from gestures or inputs of a deaf individual. This structured approach ensures that your application can effectively interpret and respond to the communication needs of users in varied contexts, enhancing accessibility and usability in real-world scenarios.

Chapter 5
Experimental Results and Comparative Analysis

5.1 Overview

This section presents and discusses all the details related to the experiments carried out to investigate and evaluate the performance of the proposed approaches.

5.2 Evaluation Metrics

Calculation of Accuracy: Accuracy is calculated as the ratio of the number of correct predictions to the total number of predictions made by the model:

Accuracy= Number of Correct Predictions/ Total Number of Predictions.

5.3 Results of The System

5.3.1 Dataset Description

Data for our smart glove project was collected from individuals using the smart glove prototype developed by our team. The primary objective of this project is to facilitate communication for individuals who are dumb and deaf by translating their hand gestures and movements into understandable sentences. The dataset comprises samples obtained from ordinary individuals as well as those who are dumb and deaf, with each sample representing a combination of hand movements and sentences expressing events in daily life.

Data Collection Source: Samples were collected directly from participants who used the smart glove prototype during interaction sessions. Individuals were invited to perform various hand movements corresponding to predefined gestures, while simultaneously expressing sentences verbally or through sign language.

Dataset Composition: The dataset consists of three classes representing different user groups:

1. Ordinary Individuals (OI): This class includes samples collected from individuals who do not have any speech or hearing impairments. Approximately 60 samples of hand movements and accompanying sentences are available for this class.
2. Dumb Individuals (DI): Samples in this class were collected from individuals who have a speech impairment but normal hearing capabilities. The dataset contains around 61 samples of hand gestures and corresponding sentences from dumb individuals.
3. Deaf Individuals (DeI): This class comprises samples obtained from individuals who have a hearing impairment but normal speech abilities. There are approximately 60 samples of hand movements and sentences from deaf individuals.

Data Availability: The dataset, along with the smart glove prototype, is openly available for research and development purposes. Researchers and developers can access the dataset upon request.

Dataset Usage: The collected dataset serves as the foundation for training and evaluating the machine learning framework implemented in ml.net. The framework employs classification algorithms to accurately classify hand movements and associated sentences into their respective categories. The classification accuracy achieved by our machine learning model is (91%), demonstrating its effectiveness in understanding and interpreting gestures and speech from individuals using the smart glove.

```
Sensor1| Sensor2| Sensor3| Sensor4| Sensor5| Greeting
37| 38| 34| 32| 46| I need a medicine.
37| 38| 35| 32| 46| I need a medicine.
37| 38| 34| 32| 47| I need a medicine.
37| 38| 35| 33| 47| I need a medicine.
36| 38| 33| 31| 46| I need a medicine.
36| 37| 35| 33| 48| I need a medicine.
35| 34| 33| 31| 50| I need a medicine.
36| 36| 34| 32| 46| I need a medicine.
36| 35| 34| 32| 46| I need a medicine.
61| 61| 63| 64| 68| waiting...
61| 61| 63| 64| 67| waiting...
60| 59| 63| 63| 66| waiting...
61| 61| 62| 63| 66| waiting...
60| 59| 62| 63| 67| waiting...
61| 58| 62| 64| 63| waiting...
59| 58| 61| 63| 70| waiting...
69| 58| 61| 61| 71| waiting...
69| 58| 61| 63| 66| waiting...
59| 59| 61| 62| 66| waiting...
59| 59| 61| 63| 72| waiting...
65| 66| 67| 69| 72| waiting...
64| 65| 67| 67| 70| waiting...
64| 65| 66| 66| 68| waiting...
65| 64| 67| 67| 70| waiting...
65| 63| 65| 66| 69| waiting...
65| 66| 68| 71| 70| waiting...
63| 64| 65| 66| 68| waiting...
36| 37| 63| 64| 54| I'm thiraty.
37| 37| 63| 64| 54| I'm thiraty.
38| 38| 64| 65| 50| I'm thiraty.
36| 38| 63| 66| 53| I'm thiraty.
39| 39| 63| 64| 53| I'm thiraty.
41| 40| 63| 64| 57| I'm thiraty.
40| 39| 62| 63| 52| I'm thiraty.
41| 40| 63| 63| 51| I'm thiraty.
42| 40| 61| 62| 52| I'm thiraty.
36| 37| 62| 63| 51| I'm thiraty.
42| 41| 63| 63| 48| I'm thiraty.
43| 42| 63| 65| 47| I'm thiraty.
43| 42| 62| 63| 48| I'm thiraty.
35| 36| 62| 61| 51| I'm thiraty.
37| 36| 62| 62| 51| I'm thiraty.
39| 37| 33| 31| 66| Thank you.
38| 36| 34| 31| 67| Thank you.
37| 38| 35| 32| 66| Thank you.
39| 38| 34| 31| 66| Thank you.
39| 37| 34| 32| 67| Thank you.
42| 41| 37| 34| 66| Thank you.
40| 40| 37| 31| 66| Thank you.
38| 37| 33| 32| 66| Thank you.
39| 37| 34| 32| 66| Thank you.
39| 37| 33| 31| 66| Thank you.
39| 37| 34| 32| 67| Thank you.
39| 38| 35| 32| 67| Thank you.
65| 62| 63| 31| 47| I need more.
62| 62| 61| 32| 46| I need more.
62| 64| 65| 31| 47| I need more.
63| 63| 64| 30| 45| I need more.
62| 62| 63| 32| 47| I need more.
62| 61| 63| 37| 43| I need more.
64| 60| 60| 31| 46| I need more.
63| 62| 63| 31| 45| I need more.
62| 62| 63| 32| 46| I need more.
62| 62| 63| 32| 45| I need more.
63| 61| 65| 32| 44| I need more.
62| 61| 61| 31| 45| I need more.
```

I need medicine.

Waiting...

Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sentence
37	38	34	32	46	I need a medicine
61	61	63	64	68	Waiting
36	37	63	64	54	I`m thirsty
39	37	33	31	66	Thank you

Chapter 6
System Development

6.1 Overview

Our comprehensive solution is designed to empower individuals with hearing impairments by providing an intuitive and independent means of communication. The smart glove integrates advanced sensors with machine learning algorithms to translate sign language gestures into text and speech in real time. The system comprises a smart glove, a mobile app built using Flutter, a .NET API for data processing and management, and Arduino Uno.

This chapter includes four main categories:

- Embedded Development
- Mobile development
- Backend-machine learning
- Integration

6.2 Methodological assumptions:

User and system requirements to activate the system.

6.2.1 User Requirements

- Users should have basic computer skills in operating systems and internet browsers.
- Users must have a connection to access the internet.

6.2.2 System Requirements

- Basic computer hardware.
- Internet Browser such as Internet Explorer, Mozilla Firefox, Google Chrome, etc.

6.3 Used Technologies

6.3.1 Dart

Dart is a programming language optimized for building fast apps on any platform. It aims to be a productive language for multi-platform development, with a flexible execution runtime platform for app frameworks. Dart is versatile and well-suited for a range of development tasks, providing tools for building fast, reliable, and high-quality applications.

6.3.2 Flutter

Flutter is an open-source UI framework by Google for building cross-platform applications from a single codebase. It enables developers to create responsive applications for Android, iOS, Windows, Mac, and Linux. Flutter's native performance, hot reload feature, and comprehensive set of widgets and APIs make it a versatile and powerful tool for building high-quality applications with ease.

6.3.3 Arduino Uno

The Arduino Uno is a popular microcontroller board used for creating electronic projects. It features an ATmega328P microcontroller, operates at 5V, and can be powered via USB or an external power supply. The board has 14 digital I/O pins (6 with PWM), 6 analog input pins, and runs at 16 MHz. It's programmed using the Arduino IDE, which supports a simplified version of C++. The Arduino Uno is widely used in education, prototyping, and DIY electronics projects due to its ease of use, extensive community support, and robust design.

6.4 Embedded Development:

In the embedded development phase of our smart glove project, we utilized hardware components such as 5 flex sensors, an Arduino Uno microcontroller, a glove, a breadboard, wires for connection, and a Bluetooth module. The primary focus was on integrating these components to create a functional prototype of the smart glove. The flex sensors were strategically placed on the glove to detect hand movements, and the Arduino Uno served as the central processing unit to collect sensor data and communicate with other devices. The Bluetooth module enabled wireless connectivity between the smart glove and the mobile application.

6.4.1 Smart glove:

Recognizing motion that did by hand with flex sensor that placed on hand Components:

- Arduino (At mega 32) microcontroller:
  We use this version of microcontroller to make easy circuit and cheaper. that can use it with communication protocols HC-05

- Flex sensor that types of (Motions sensor):
  Five flex sensors are used where flex sensors are the sensors that change in resistance depending on the amount of bend on the sensor since the resistance is directly proportional to the amount of bending and they convert the change in bend to electrical resistance, The amount of resistance is changing when flex sensor is bending.

Flex sensor acts as variable resistance R1 along with the static resistance R2 and works on the principle of Voltage divider. The variable output voltage V0 is calculated as:

$$V0 = Vin(R1/(R1+R2))$$

Each flex sensor requires a voltage of +5V. Each flex sensor receives a +5V. Each flex sensor receives a +5V supply when the power is turned on.

When a user makes a letter gesture, the five signals from each flex sensor with combination of static resistance of 10KΩ gives analog resistance value and are sent to a microcontroller, where they are transformed from analogue to digital values and compare this digital data with prestored data.

The digital values are compared with already stored data in Arduino if values are matched with the database stored in Arduino, then display the output on mobile application screen.

- Resistor 10Kohm:
  To protect sensor from any high current.

- HC-05:
  HC-05 is an interesting module that allows you to add two-way (full-duplex) wireless capability to your projects. It may utilize this module to communicate between two microcontrollers, such as an Arduino, or with any Bluetooth-enabled device, such as a phone or laptop. Many Android applications are already available, making the procedure much easier. The module interacts with a microcontroller through USART at a rate of 9600 baud rate.

6.5 Mobile Development:

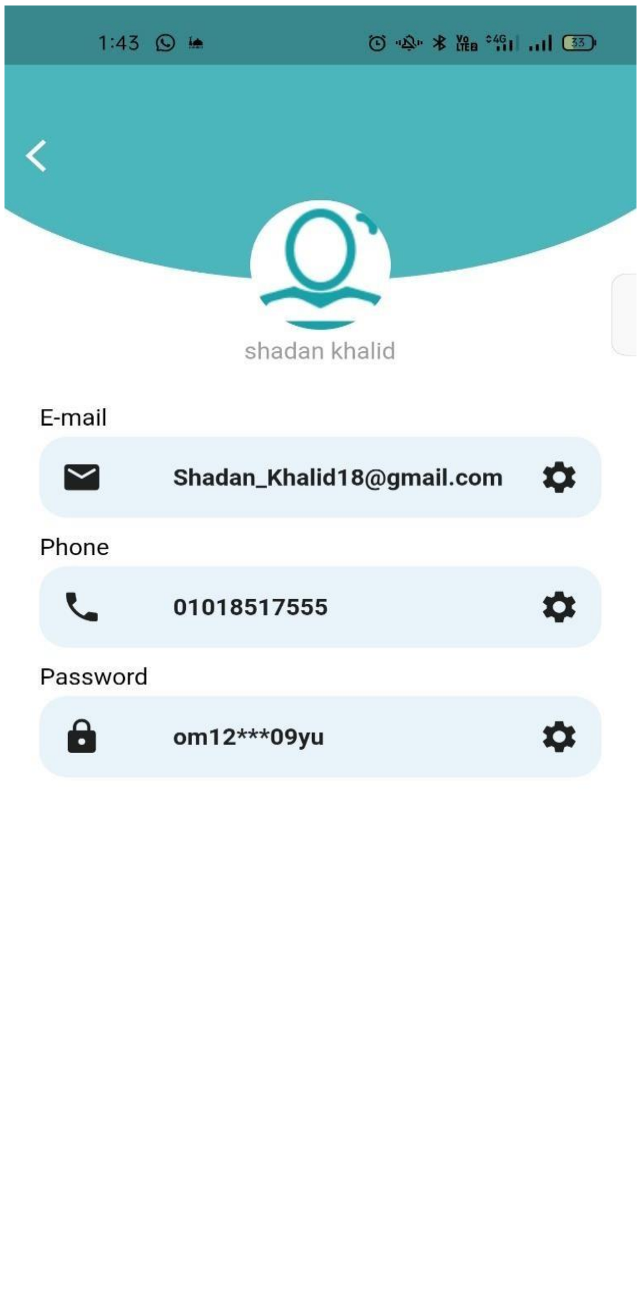
In this phase, our primary goal was to create a user-friendly interface and implement various features using the Flutter framework with an MVC (Model-View-Controller) architecture. Here's a breakdown of the key functionalities and pages incorporated into the mobile application:

1. Splash Screen:

- The splash screen is the first screen users encounter upon launching the app. It serves as a visual introduction to the application and often displays the app's logo or branding elements.
- Below is a virtual image illustrating a typical splash screen design:

2. User Profiles:

- The user profiles page allows users to view and manage their account information, including profile pictures, usernames, and settings.
- Users can update their personal information and customize their preferences.
- Here's a virtual image showcasing the user profiles page design:

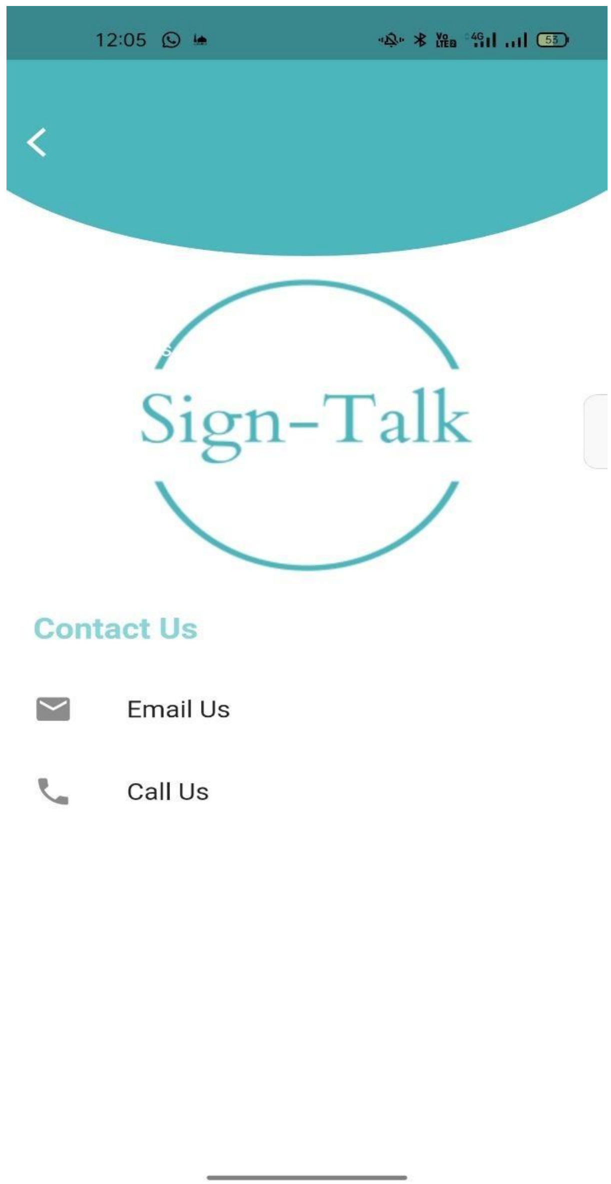


3. Tutorial Pages:

- Tutorial pages provide users with step-by-step guidance on how to use the application's features.
- These pages are especially useful for new users who may need assistance in getting started with the app.

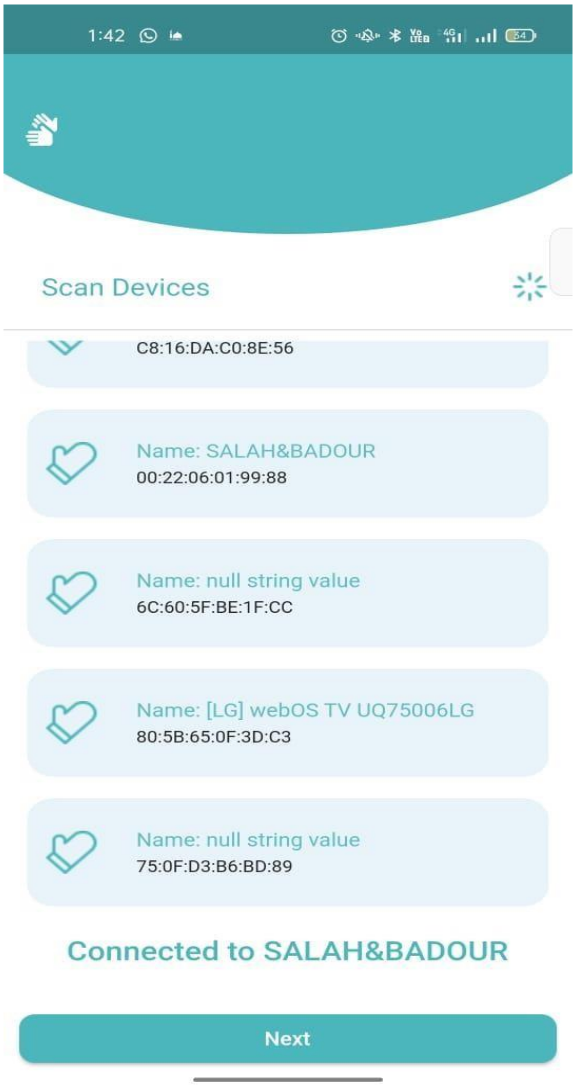
4. Help Center:

- The help center serves as a centralized hub for users to access support resources, FAQs, and contact information for customer service.
- Users can search for answers to common questions or submit inquiries directly through the app.
- Below is a virtual image representing the help center page design:



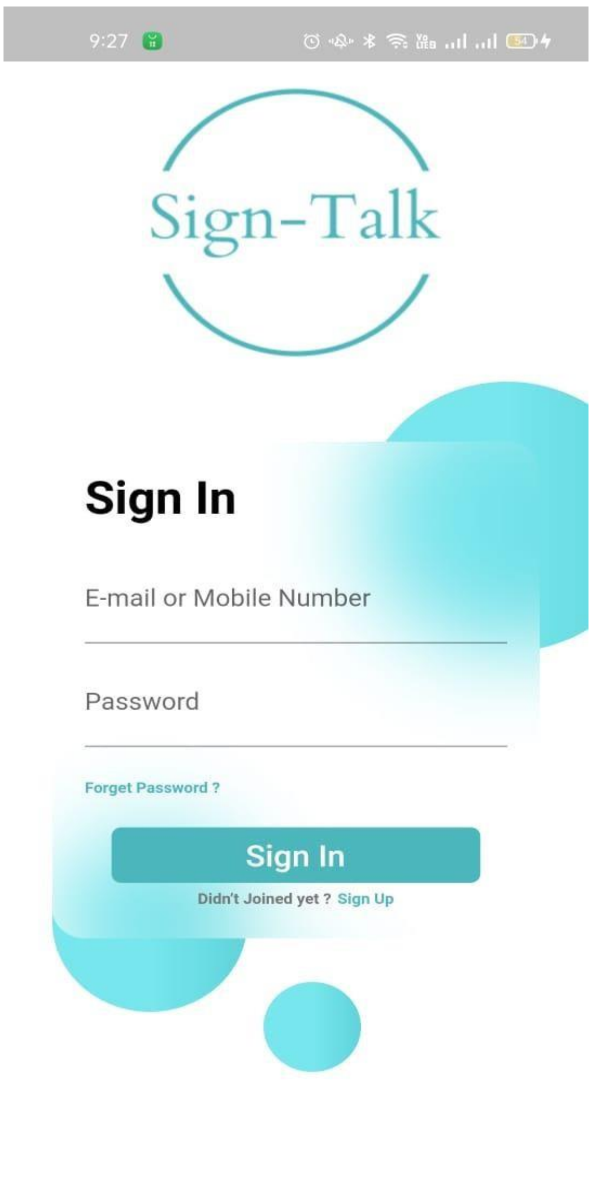
5. Bluetooth Connection Management:

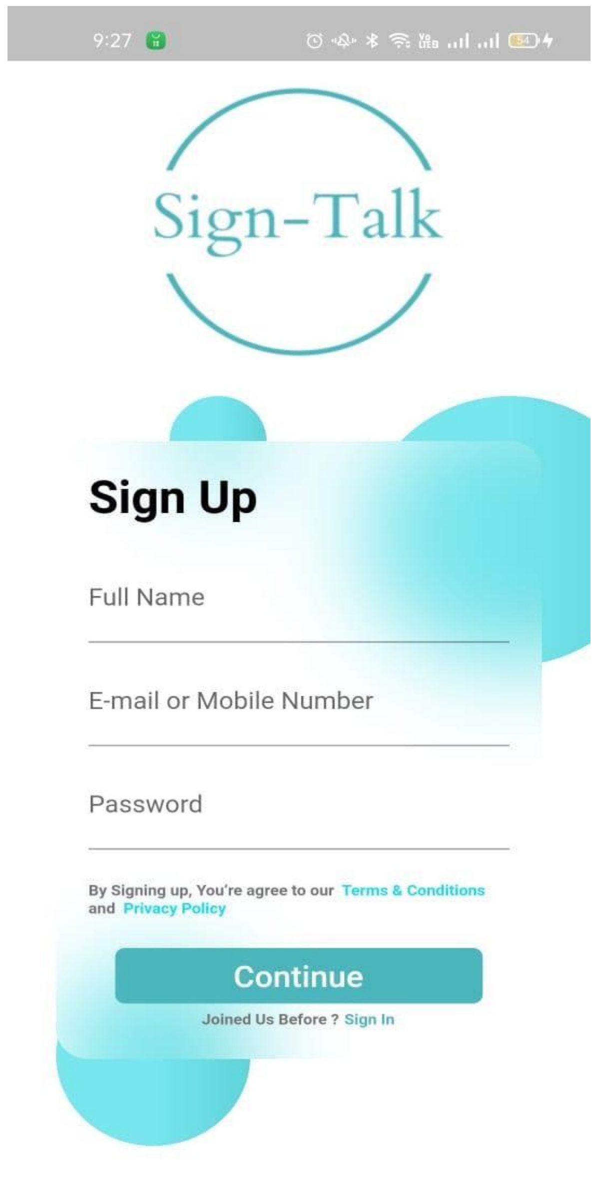
- The Bluetooth connection management feature allows users to establish and manage connections between the mobile application and the smart glove.
- Users can initiate Bluetooth pairing, view connected devices, and troubleshoot connectivity issues.
- Here's a virtual image illustrating the Bluetooth connection management interface:



6. Firebase Authentication:

- Firebase authentication provides a secure and seamless way for users to register and log in to the application.
- Users can create new accounts or sign in using existing credentials, ensuring access to personalized features and data.
- Below is a virtual image showcasing the Firebase authentication process:



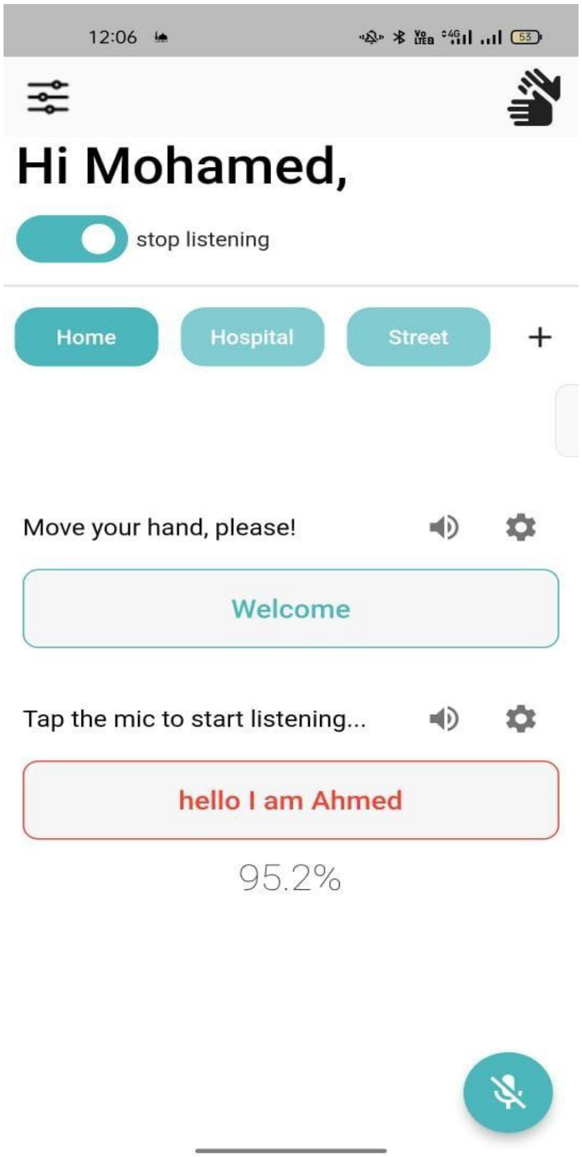


7. API Integration:

- API integration enables communication between the mobile application and the backend server, allowing for data exchange and interaction.
- Users can retrieve real-time information, submit requests, and receive updates from external sources.

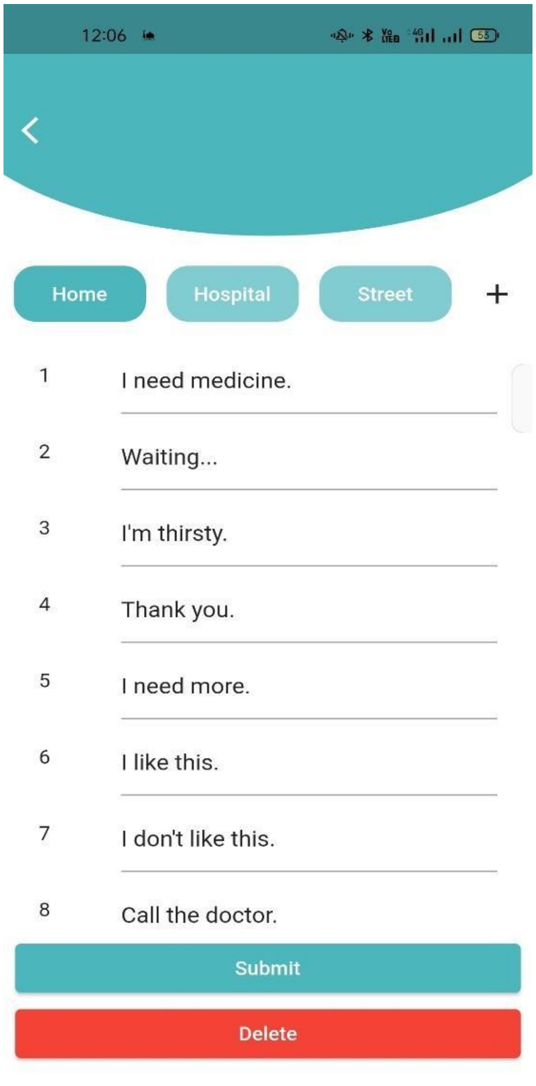
8. Text-to-Speech and Speech-to-Text Conversion:

- Text-to-speech and speech-to-text conversion features facilitate seamless communication between users wearing the smart glove and ordinary individuals.
- Users can convert spoken words into text and vice versa, enabling effective communication in various scenarios.
- Here's a virtual image demonstrating the text-to-speech and speech-to-text conversion functionalities:



9. Mode Selection Options:

- Mode selection options allow users to customize sentence generation based on their current environment and ability to create new mode or delete old ones (e.g., home, school, office).
- Users can choose the appropriate mode to tailor the conversation to specific contexts and preferences.
- Below is a virtual image showcasing the mode selection interface:



By incorporating these features and pages into the mobile application, we aimed to create a comprehensive and user-centric experience for individuals using the smart glove to communicate effectively with others.

6.6 Backend-Machine Learning:

In the backend-machine learning phase, we focused on leveraging the ASP.NET Core framework (version 7) along with Entity Framework Core for database operations and dependency injection for managing services. Let's explore this phase further, along with virtual images for each relevant component:

1. ASP.NET Core Framework:

- We chose the ASP.NET Core framework for its robustness, scalability, and cross-platform capabilities. ASP.NET Core provided the foundation for building the backend infrastructure of our smart glove application.

2. Entity Framework Core for Database Operations:

- Entity Framework Core facilitated database operations, allowing seamless interaction with the underlying database. We utilized EF Core's ORM (Object-Relational Mapping) capabilities to define database models and perform CRUD (Create, Read, Update, Delete) operations.

3. Dependency Injection for Managing Services:

- Dependency injection played a crucial role in managing services and promoting code modularity. By registering and injecting dependencies into our application components, we achieved better separation of concerns and improved testability.

4. Model Design:

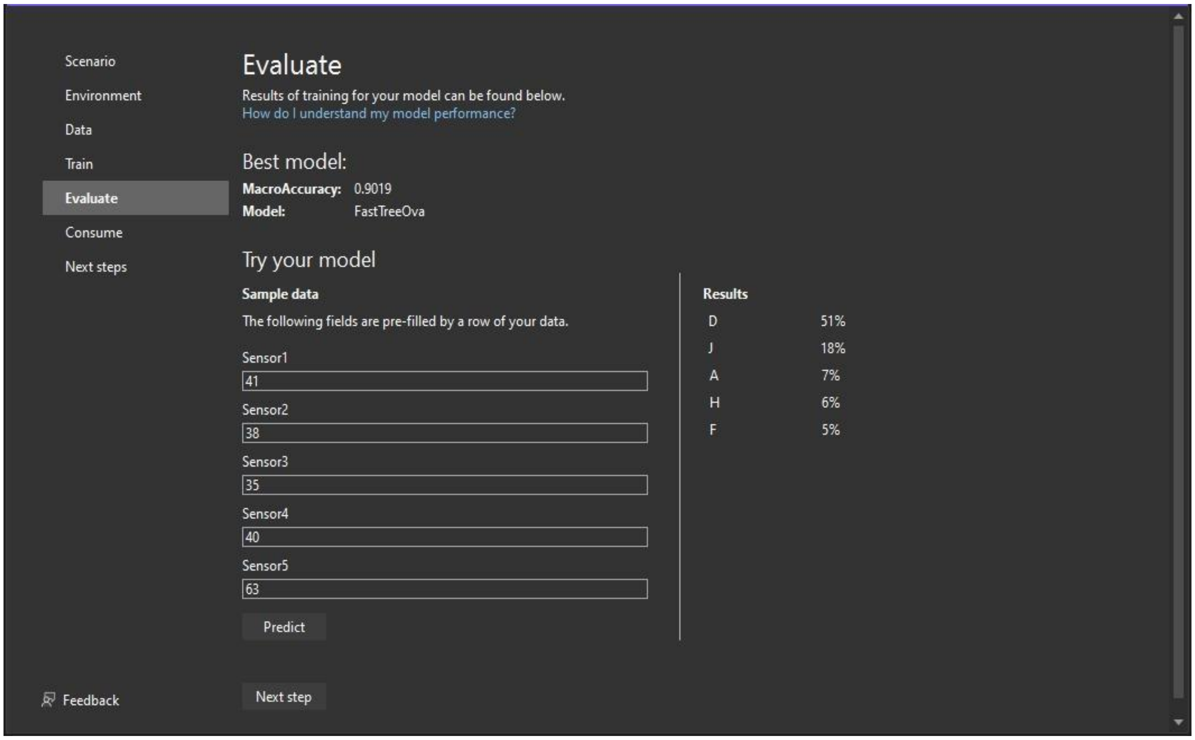
We designed models for users, sensor data, sign language classification, and glove environment. These models represented entities within our application domain and facilitated data storage and manipulation.

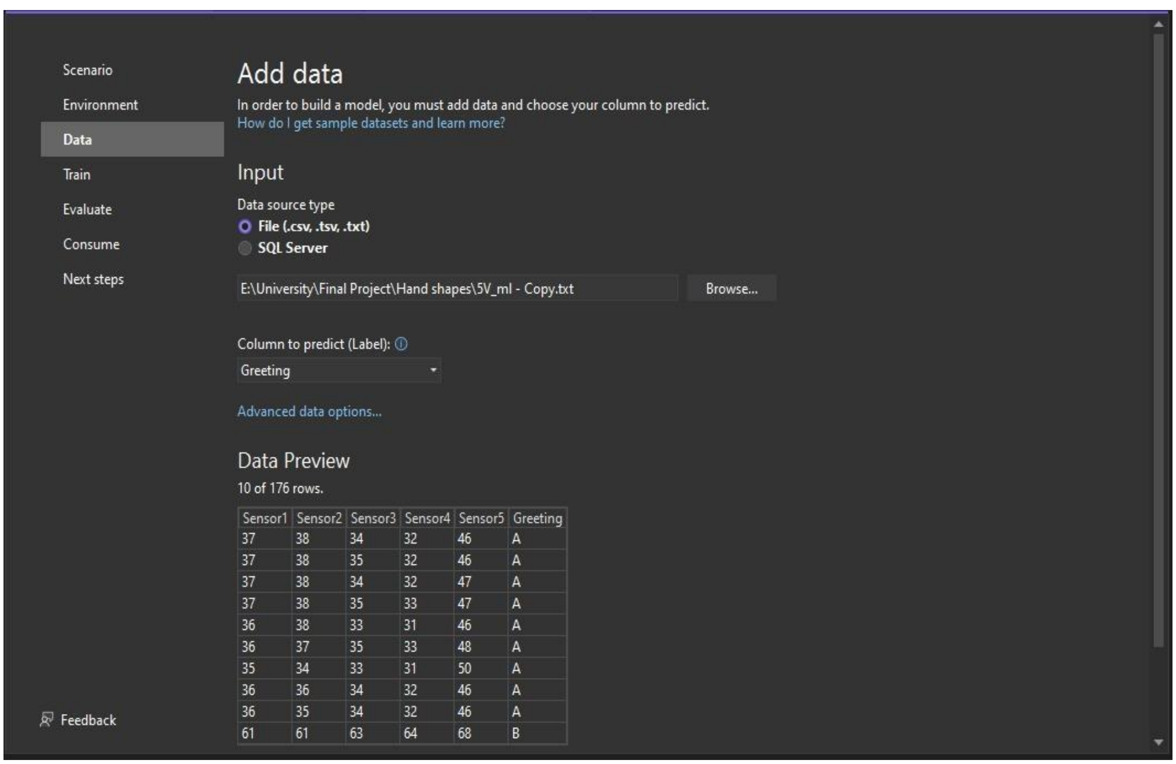
5. Controller Design:

- Each model had its corresponding controller responsible for routing communication between the front-end application and the backend server. Controllers defined actions to handle incoming requests, execute business logic, and return appropriate responses.

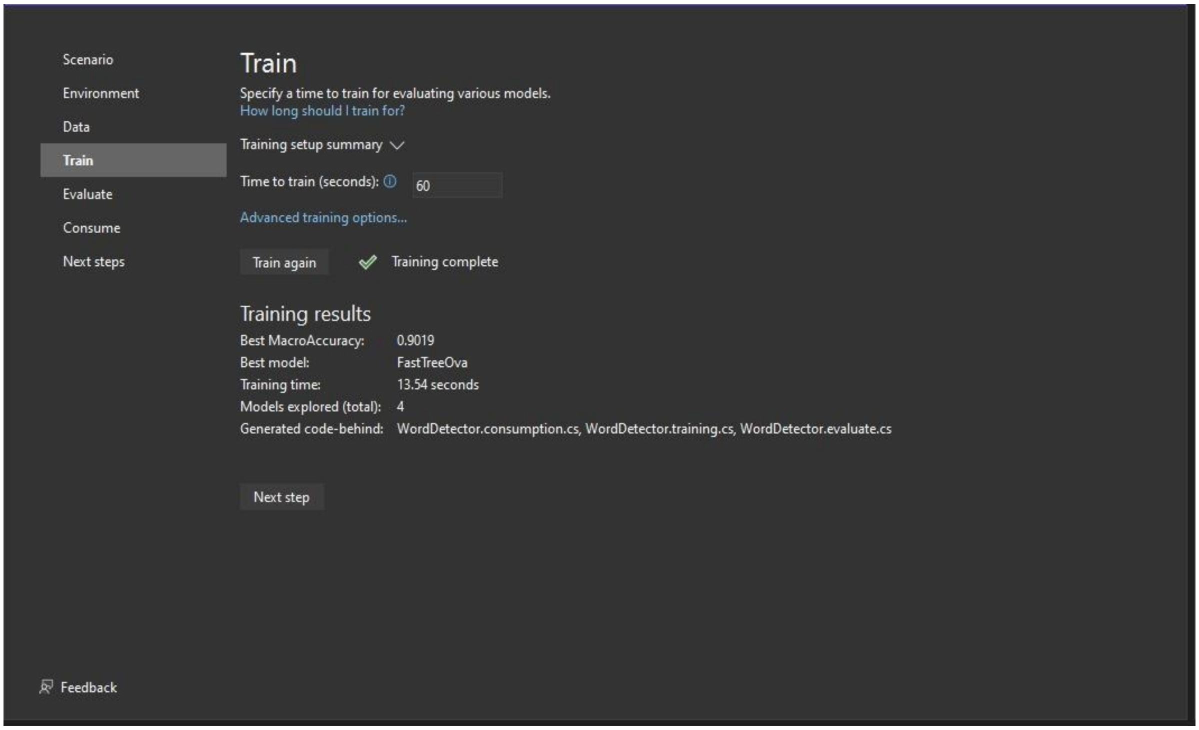
6. Machine Learning Model Integration:

- Initially, our machine learning model predicted words based on the five sensor values received from the smart glove. However, to accommodate multiple environments, we modified the model to predict symbols instead of words.
- Here's a virtual image showcasing the integration of the machine learning model into the backend:





Sensor1	Sensor2	Sensor3	Sensor4	Sensor5	Greeting
37	38	34	32	46	A
37	38	35	32	46	A
37	38	34	32	47	A
37	38	35	33	47	A
36	38	33	31	46	A
36	37	35	33	48	A
35	34	33	31	50	A
36	36	34	32	46	A
36	35	34	32	46	A
61	61	63	64	68	B


Scenario
Environment
Data
Train
Evaluate
Consume
Next steps
Train
Specify a time to train for evaluating various models.
How long should I train for?
Training setup summary
Time to train (seconds):
60
Advanced training options...
Train again
Training complete
Training results
Best MacroAccuracy: 0.9019
Best model: FastTreeOva
Training time: 13.54 seconds
Models explored (total): 4
Generated code-behind: WordDetector.consumption.cs, WordDetector.training.cs, WordDetector.evaluate.cs
Next step
Feedback

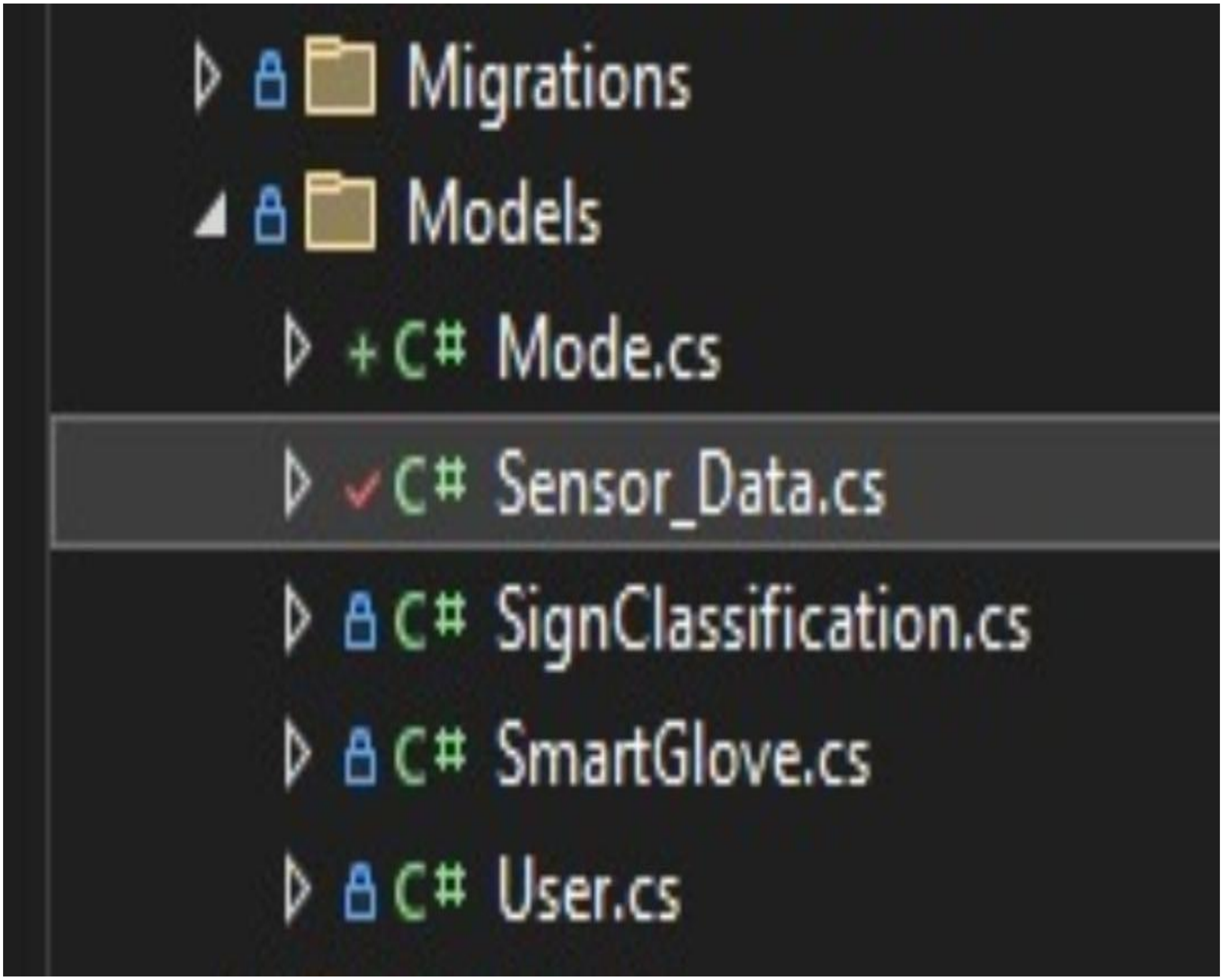
7. Mode Model for Environment-Specific Sentence Generation:

- We introduced a Mode Model to handle environment-specific sentence generation. This model mapped symbols to words based on the selected mode (e.g., Home, Hospital, Street), allowing for dynamic adaptation to different contexts.

By implementing these components and functionalities in the backend-machine learning phase, we established a robust and scalable backend infrastructure to support the smart glove application's core functionalities.

6.7 Data Models

The backend of the Smart Glove for deaf people project encompasses a RESTful API built on ASP.NET Core, providing essential functionalities for managing users, sensor data, smart gloves, sign classifications, and modes. This documentation outlines the structure, endpoints, and data models utilized in the API.



6.7.1 Technologies Used

- Framework: ASP.NET Core
- Database: Entity Framework Core with SQL Server
- Authentication: JWT (not implemented in provided code)
- API Documentation: Swagger/OpenAPI (not implemented in provided code)

6.7.2 Data Models

User

- Properties:
  - Id: Unique identifier for the user.
  - Name, Email, Password, Gender, Date-Time, Address, Phone: Basic user information.
  - Smart Gloves: Collection of smart gloves owned by the user.
  - Sensor Data: Collection of sensor data associated with the user.

Smart Glove

- Properties:
  - Id: Unique identifier for the smart glove.
  - Version, Description, Price: Details about the smart glove.
  - Owner-ID: Foreign key linking to the User who owns the smart glove.

Sensor Data

- Properties:
  - Data-Id: Unique identifier for the sensor data entry.
  - Time Stamp: Timestamp when the sensor data was recorded.
  - Sensor1_Value to Sensor5_Value: Values from sensors embedded in the smart glove.
  - User-ID: Foreign key linking to the User associated with the sensor data.
  - Mode-Id: Foreign key linking to the Mode associated with the sensor data.

Sign Classification

- Properties:
  - Id: Unique identifier for the sign classification entry.
  - Word: The word classification derived from sensor data.

    - Time Stamp: Timestamp of when the sign classification was generated.
    - Sensor Data-ID: Foreign key linking to the Sensor Data associated with this classification.

Mode

- Properties:
    - Mode-Id: Unique identifier for the mode.
    - Mode-Name: Name of the mode.
    - A To N: Different phrases or responses associated with the mode.

6.8 API Endpoints

Users

- GET /api/Users: Retrieves all users.
- GET /api/Users/{id}: Retrieves a specific user by id.
- POST /api/Users: Creates a new user.
- PUT /api/Users/{id}: Updates an existing user.
- DELETE /api/Users/{id}: Deletes a user by id.

Smart Gloves

- GET /api/SmartGloves: Retrieves all smart gloves.
- GET /api/SmartGloves/{id}: Retrieves a specific smart glove by id.
- POST /api/SmartGloves: Creates a new smart glove.
- PUT /api/SmartGloves/{id}: Updates an existing smart glove.
- DELETE /api/SmartGloves/{id}: Deletes a smart glove by id.

Sensor Data

- GET /api/Sensor_Data: Retrieves all sensor data entries.
- GET /api/Sensor_Data/{id}: Retrieves a specific sensor data entry by id.
- POST /api/Sensor_Data: Creates a new sensor data entry and generates sign classification.
- PUT /api/Sensor_Data/{id}: Updates an existing sensor data entry.
- DELETE /api/Sensor_Data/{id}: Deletes a sensor data entry by id.

Sign Classifications

- GET /api/SignClassifications: Retrieves all sign classifications.
- GET /api/SignClassifications/{id}: Retrieves a specific sign classification by id.
- POST /api/SignClassifications: Creates a new sign classification.
- PUT /api/SignClassifications/{id}: Updates an existing sign classification.
- DELETE /api/SignClassifications/{id}: Deletes a sign classification by id.

Modes

- GET /api/Modes: Retrieves all modes.
- GET /api/Modes/{id}: Retrieves a specific mode by id.
- POST /api/Modes: Creates a new mode.
- PUT /api/Modes/{id}: Updates an existing mode.
- DELETE /api/Modes/{id}: Deletes a mode by id.

Database Integration

The API integrates with an SQL Server database using Entity Framework Core. It includes database seeding to populate initial user and mode data for testing purposes.

Error Handling

Error handling is implemented for each endpoint to manage cases such as invalid requests, not found resources, and database update conflicts.

Conclusion

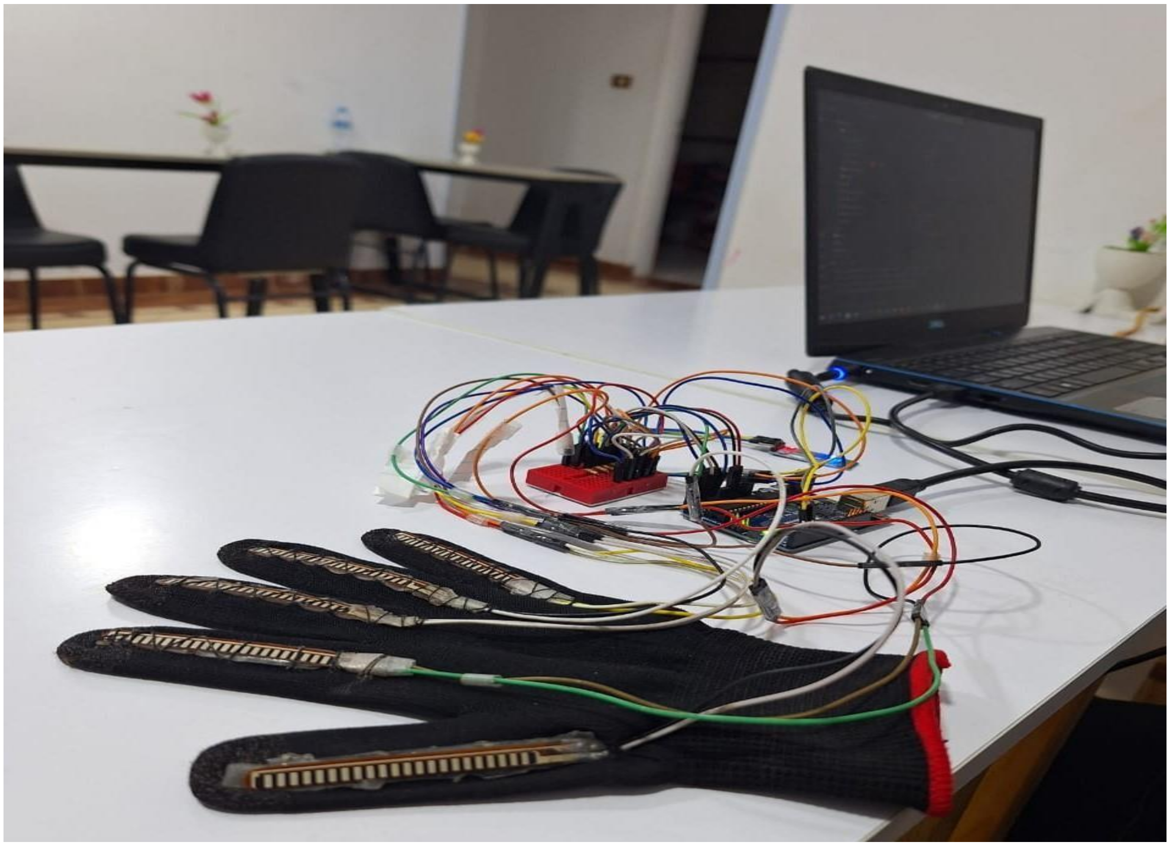
This API serves as the backend infrastructure for the Smart Glove project, facilitating data management, classification, and interaction between the embedded system, mobile app, and front-end interfaces. Future enhancements could include authentication, expanded error handling, and more sophisticated data analytics features.

6.9 Integration:

The integration phase focused on establishing seamless communication between the mobile application and the backend server, as well as connecting the application to the smart glove via Bluetooth. This involved implementing data exchange protocols, handling responses from the backend, and managing Bluetooth connectivity for real-time interaction with the smart glove. By integrating all components effectively, we ensured smooth operation and interoperability between the smart glove, mobile application, and backend infrastructure.

Overall, through a systematic approach to embedded development, mobile development, backend-machine learning, and integration, we successfully realized the smart glove project, providing a practical solution for facilitating communication for individuals who are dumb and deaf.

Here's an image showcasing the integration of the whole system:



Chapter 7
Conclusions & Future Work

Conclusions

In this project, we have introduced the structure and assessment of a smart glove system designed to translate sign language gestures into text and speech in real-time. The SIGN-TALK smart glove utilizes multimodal data from various sensors (flex sensors, IMU, pressure sensors) to accurately interpret hand movements and gestures, facilitating effective communication for individuals with hearing and speech impairments. Our process starts with the glove capturing the hand gestures, which are then transmitted to the mobile app for preprocessing. The app ensures that the data is in the correct format and size needed for accurate translation.

Subsequently, the processed data, along with the user's profile information, is stored in our database. The machine learning models employed in our system are capable of translating gestures into meaningful text and speech outputs, providing real-time communication support. Our models achieved high accuracy rate [90.1%] in gesture recognition, ensuring reliable and effective translations. Therefore, the choice of specific gestures or the full range of sign language can be left to the user, ensuring flexibility and adaptability.

Finally, SIGN-TALK offers a comprehensive solution for bridging communication gaps. The proposed system has three main stages: data acquisition and preprocessing and gesture translation. The proposed translation approach was implemented and assessed, demonstrating that the method yields high performance in both gesture recognition and translation tasks. Specifically, our method achieves high accuracy rates in both individual gesture recognition and continuous gesture translation using advanced machine learning models. Overall, the SIGN-TALK project not only showcases technical innovation but also a profound commitment to enhancing the quality of life for individuals with hearing and speech impairments, providing them with a vital tool for independent communication.

Future work

As we continue to develop and refine the SIGN-TALK smart glove, several future enhancements have been identified to further improve its functionality, user experience, and accessibility:

1. Comprehensive Sign Language Coverage: Extend the gesture recognition capabilities to cover a wider range of sign languages, including regional and less commonly used variants, to accommodate a broader user base.

2. IoT Integration: Connect the smart glove with other IoT devices in the home or workplace to enable a more integrated communication environment.

3. Third-party App Integration: Allow the glove to work seamlessly with popular communication and productivity apps, such as messaging platforms and virtual assistants.

4. Biometric Sensors: Integrate additional sensors to monitor health parameters such as heart rate, skin temperature, and stress levels, providing comprehensive health monitoring alongside communication support.

5. Emergency Alerts: Develop an emergency alert system that can notify caregivers or emergency services if the user experiences a health crisis.

6. Multilingual Support: Expand language support to include more languages and dialects, making the app accessible to a more diverse population.

7. User Community: Create a platform for users to share their experiences, tips, and custom gestures, fostering a supportive community.

Bibliography

- Thiago Simões Dias, José Jair Alves Mendes Junior ∗, Sérgio Francisco: Comparison between handcraft feature extraction and methods based on Recurrent Neural Network models for gesture recognition by instrumented gloves: A case for Brazilian Sign Language Alphabet, September 2022.

- A New Intelligent Approach for Deaf/Dumb People based on Deep Learning: Haitham Elwahsh1, Ahmed Elkhouly1, Emad Aboul Nasr2, Ali K. Kamrani3, and Engy El-shafeiy4.

- Amal Babour*, Hind Bitar, Ohoud Alzamzami, Dimah Alahmadi, Amal Barsheed, Amal Alghamdi, and Hanadi AlmshjaryIntelligent gloves: An IT intervention for deaf-mute people, December 2022.

- Hyeon-Jun Kim 1 and Soo-Whang Baek: Application of Wearable Gloves for Assisted Learning of Sign Language Using Artificial Neural Networks, March 2023.

- Muhammad Saad Amin 1, Syed Tahir Hussain Rizvi 2,*, Alessandro Mazzei 1 and Luca Anselma: Assistive Data Glove for Isolated Static Postures Recognition in American Sign Language Using Neural Network, March 2023.

- Ziad MHD Adib Burhani, Judhi Prasetyo: The Sign Language Interpreting Gloves, January 2023., Judhi Prasetyo : The Sign Language Interpreting Gloves , January 2023