TED UNIVERSITY

CMPE 491

Senior Project

Gesture Guide: Virtual Assistant for the Hearing-Impaired

Project Analysis Report

Submission Date: 26.11.2023

Team Members:

- Berke Lahna
- Canberk Aydemir
- Ceyhun Toker
- Mehmet Fatih Ülker

Advisor: Venera Adanova

Jurors: Gökçe Nur Yılmaz and Eren Ulu

Web Site: https://berkelahna.github.io/Pages

TABLE OF CONTENTS

1. Introduction

In a society where digital communication is becoming more and more important, the hearing-impaired face unique challenges that are often ignored. An innovative idea that aims to close the communication gap faced by those with hearing impairments is the Gesture Guide project. It provides an innovative approach to accessibility applications. Gesture Guide uses the power of hand gestures to improve communication among the deaf community and the hard of hearing.

For those who are hard of hearing, using traditional forms of communication can be time-consuming and restrictive in their daily lives. Relying on written communication through notepads delays natural conversation and is uncomfortable, particularly when talking to people who do not know sign language. Gesture Guide intends to change this experience by enabling smoother interactions between individuals who are deaf and those who are not familiar with sign language through the use of machine learning models for interpretation.

The analysis examines thoroughly the fundamental elements of the Gesture Guide project, analyzing the difficulties that the hearing-impaired now encounter, showcasing the innovative features of the proposed system, and offering a thorough examination of both functional and non-functional requirements. We want to establish the basis for a system that is not just technically strong but also ethically and socially conscious through careful examination. The following sections will provide insight into the complexities of the current and planned systems, providing a thorough understanding of the techniques utilized, the models built, and the expected impact on the lives of the hard-of-hearing.

As we begin on this analysis journey, we become aware of the significant consequences of our work, which go far beyond traditional software development. Gesture Guide aims to be more than just a technological solution; it's an example of how creativity may advance inclusivity, reduce obstacles to communication, and promote a more connected and understanding community.

2. Current System

If we take a general overview of our current project system, it is designed to support the daily life of hearing-impaired individuals. Therefore, our main functions involve the recognition and adaptation of sign language, as well as the recognition of hand gestures to perform desired functions in the operating system. Currently, the process of detecting sing language through video, one of the key functions we aim to implement, is not at the desired performance level. However, the prototype is successfully operational and accessible through our website. In the future, we plan to improve this process by enhancing the data training procedure to make it faster and more stable.

Another crucial point is the creation of the interface that the user will use and seek assistance from. The login system, which we are currently working on and is close the completion, is an essential step before accessing this interface. We consider this part crucial because we value user privacy rights.

In conclusion, the current system of our project is in the form of a prototype demonstrating the significant functions and features promised by our project. Our main goals for the upcoming period include the creation and improvement of the interface, increasing the model accuracy to make it stable and efficient, and customizing and enhancing user preferences.

3. Proposed System

3.1 Overview

Gesture Guide project is projected to be a complex accessibility application that uses machine learning and computer vision technologies to assist hearing impaired. This assistance contains both sign language interpretation and gesture-based personal assistant as mentioned in the specification report. With this section, we aim to better demonstrate possible use cases and goals of this project by using a variety of methodologies ranging from displaying system models used in software engineering to underlining requirements of the project. Overall, this section will be the blueprint for both the project implementation and the design document for the project.

3.2 Functional Requirements

- **Sign language Interpretation:** The Gesture Guide application should be able interpret sign language as one of the biggest concerns of the application. User Interface should lead the users to a sign language interpretation option which would require the usage of device camera. Then using the

device camera, the sign shown by the user should be classified and displayed via the user interface. This requirement aims to help users with daily communication and establishing small talks.

- **Virtual Assistant (Gesture Recognition)**: Gesture Guide application should provide users with a virtual assistant that is specifically targeted for hearing-impaired. The target users of this application require this virtual assistants to understand hand gestures and create basic responses to this hand gestures. Users should be provided an option to choose virtual assistant through the user interface and just like the sign language interpretation feature, it should demand the usage of device camera. Then through device camera application should classify the hand gesture and call the function mapped to the given gesture. As an example, if users do a wave gesture the application should open the calculator app which is the function mapped to the wave gesture.
- **Personalized Accounts:** The application should provide personalized accounts in order to establish security and to be in compliance with data ethics. The users should be able to register to the system with their personal information, set their preferences and view their virtual assistant history. Personalized accounts also require a user interface that allows users to login to the system. This requirement's biggest aim is to make sure that every user's data is stored in compliance with data ethics while also giving users new features like viewing their virtual assistant history.
- **View/ Set Preferences:** Users should be able to set and view their preferences such as font size, theme and language. Considering the fact that Gesture Guide is an accessibility application targeted for hearing impaired some features or preferences can really help users with usability of the application.
- **View Virtual Assistant History:** Users should be able to view their previous usages of the virtual assistant through the user interface. With this feature Gesture Guide aims to properly store users' usage of the virtual assistant and serve it to the users.

3.3 Non-Functional Requirements

- **Compatibility:** As expected from every virtual assistant application the Gesture Guide application should be available for a wide range of devices. Computer users should be able to access the web site, whereas mobile device owners should be able to access the application through their mobile devices. There can be some modification that is unique to each version, but the general functionality of the applications should be the same. The primary mobile device the Gesture Guide is projected to be compatible with is android devices as mentioned earlier in the specification report. In the future, the application may also be migrated to other mobile devices.

- **Security:** Security is nonetheless one of the most important non-functional requirements. The application should handle authentication/authorization whenever users login to the system to ensure more complex problems related to data privacy do not occur.
- **Performance:** The application should be able handle request from different users received from different types of devices while also responding to all of them accurately. Performance requirement is very crucial point in order to serve multiple users in the desired way taking into account the target users and their needs.
- **Model Accuracy:** Functional requirements which are sign language interpretation and virtual assistant require usage of a combination of machine learning models and computer vision tools. In order to have these requirements work as expected the application requires models with high accuracy rates. Since both problems are classification problems, the application needs to provide models that are specifically trained in order to get the best recall (false positive), precision (false negatives) and f1 scores. Model accuracy and model evaluation is nonetheless one of the most important requirements for the gesture guide project.
- **Usability:** Gesture Guide application should have a user-friendly interface that helps users to get the maximum benefit from the application. The user interface should provide easy access to both sign language interpreter and the virtual assistant and should not confuse the users with its general layout.
- **Software Reliability:** Gesture Guide application should be reliable in the sense of software, meaning that it should have minimal downtimes and should be serving customers as continuously as possible.

3.4 Pseudo requirements

Pseudo requirements are also called constraints and can be considered as a subset of non-functional requirements. And there is not a solid border that differentiates pseudo requirements from non-functional requirements. Therefore, this section will focus on requirements that are not related to quality but rather the preferences we have chosen to stick with for the gesture guide application.

- Machine learning models will be implemented using the TensorFlow ecosystem due to its easy interface to build neural networks and compatibility with different programming languages.
- Deep learning based on deep neural networks will be used for machine learning problems.
- Convolutional neural networks, a distinct type of artificial neural networks, will be used for image classification problems.

- JavaScript will be used for the web application both for back-end and front-end in form of React.js and Node.js.
- A native approach will be used for the mobile application and web application using React Native.

3.5 System Models

3.5.1 Scenarios

Users of gesture Guide application go through 3 main scenarios. These scenarios include signing up / logging in path, sign language interpretation path and virtual assistant path.

- **Signing up/ Loggin in Scenario**: At the initial use of the application users are required create an account by entering personal information that is required for the system. After a user has created an account, every time users try to access the application, they should log in to system using their credentials like username and password. This scenario is in parallel to personalized accounts and security requirements mentioned earlier.
- **Sign Language Interpretation**: When users are logged in to the system via the user interface, they should be able to choose between virtual assistant and sign language interpreter. In the case of sign language interpreter at the first usage, users are asked to give access to usage of device camera. After getting access the application should be able detect hands in the frame and try to classify the sign using a machine learning model that is trained on the sign language. After the sign is recognized, it should write the meaning of the sign to the user. This scenario relies on model accuracy just as mentioned before.
- **Virtual Assistant:** Just as in the case of sing language interpretation, users that have been logged in to the system can choose the virtual assistant feature and they would be asked to give permission to device camera usage. After that the application will try to classify a gesture, if it is successful the application will call a functionality mapped to the gesture. All distinct hand gestures should have a distinct response providing the users with a virtual assistant that works with gestures. Model accuracy is a crucial factor for virtual assistant to serve users the best way possible.

Other scenarios include viewing/editing preferences using the user interface and viewing virtual assistant history.

3.5.2 Use Case Model

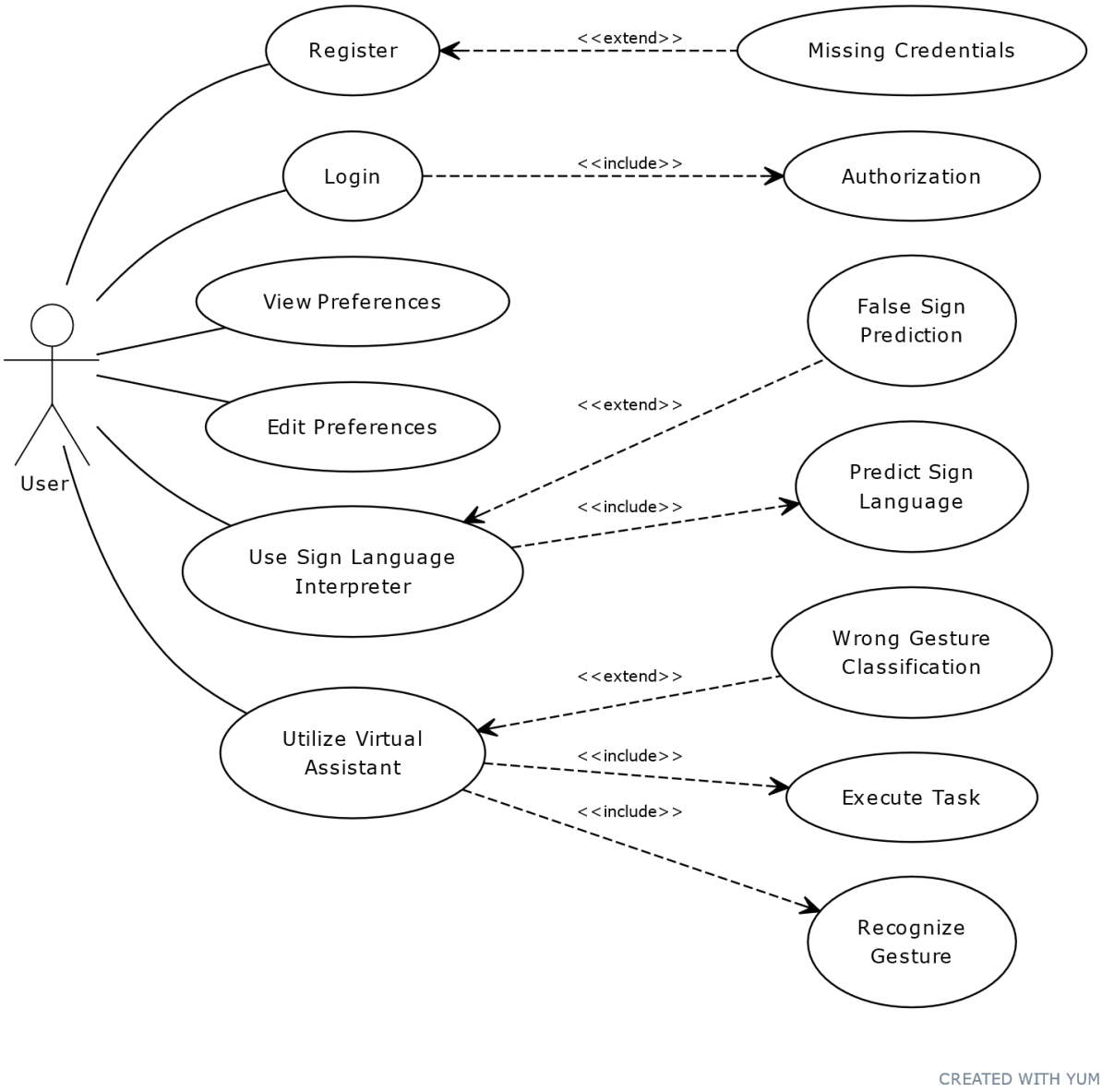

Figure 1

Figure 1 shows the use case diagram which is used to demonstrate actors of the system and how they interact with the system. Here we have 6 use cases that the actor directly interacts with, and we have 7 more use cases that are dependent on another specific use case. Extends relationship is used to illustrate a possible outcome for a use case. Includes relationship is used to display the use cases that are not started by the actor.

3.5.3 Object and Class Diagram

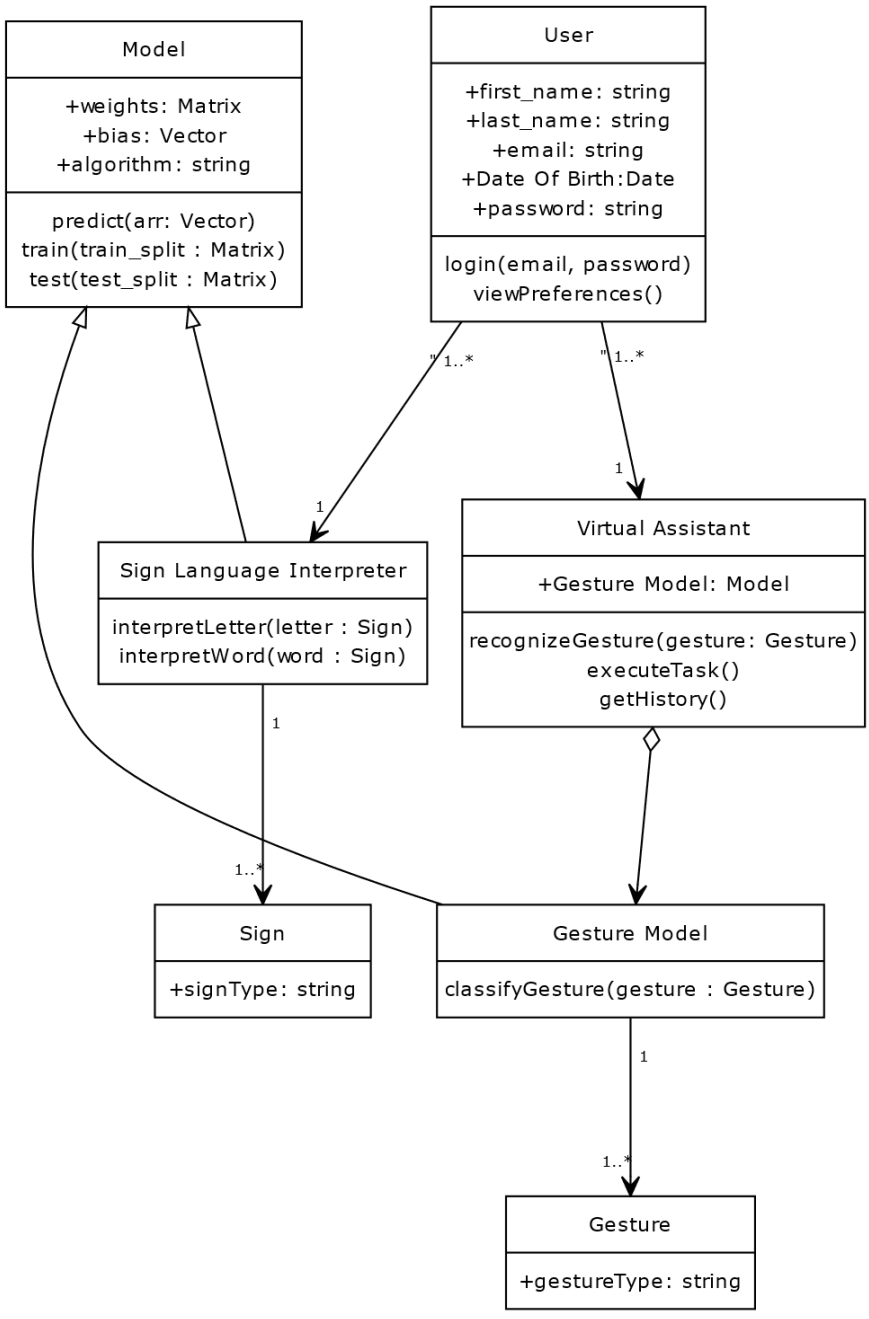

Figure 2

Figure 2 shows the UML class diagram for the Gesture Guide project. Some important notes : The diagram displays a generalization relationship where the parent class is model and sub-classes are sign language interpreter and gesture recognition model meaning that both of them are models. There is also an aggregation relationship between virtual assistant and gesture recognition model since virtual assistant has a gesture recognition model. If we need to dive deep into cardinality of relationships, many users use a single virtual assistant and a single sign language interpreter. Sign language interpreter works with many signs and gesture recognition model works with many gestures.

3.5.4 Dynamic Models

1. Sequence Diagram

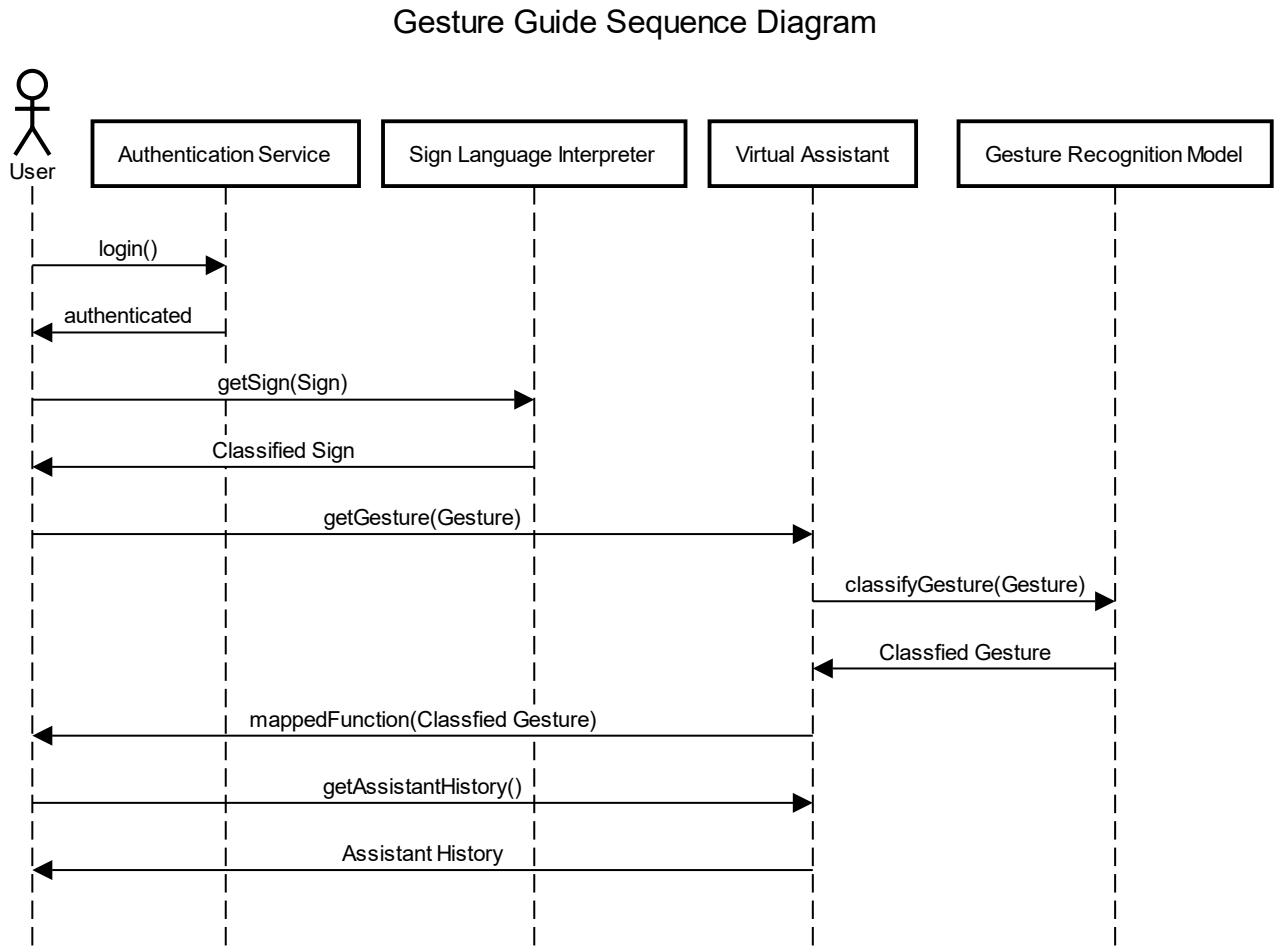

Figure 3

Figure 3 shows the sequence diagram for the basic workflow of the proposed gesture guide application and its scenarios. To better tell the workflow let us go through the diagram sequence by sequence.

- **Login**: Users should login to the system by entering their credentials, then this credentials are checked by the authentication service and returns successful if credentials are correct.
- **Sign Language Interpreter**: In the next sequence, users can use the sign language interpreter by displaying a sign using their hands which is then classified by the sign language interpreter and returned back to the user.
- **Virtual Assistant**: In another sequence users can use the virtual assistant and perform a gesture using their hands which is sent to the gesture recognition model by the virtual assistant. Then the gesture recognition model classifies the gesture and returns it to the virtual assistant. Then the virtual assistant executes the mapped function for the classified gesture. Another sequence is when users request the virtual assistant to get their history then the virtual assistant returns the records of past virtual assistant usages.

2. Activity Diagram

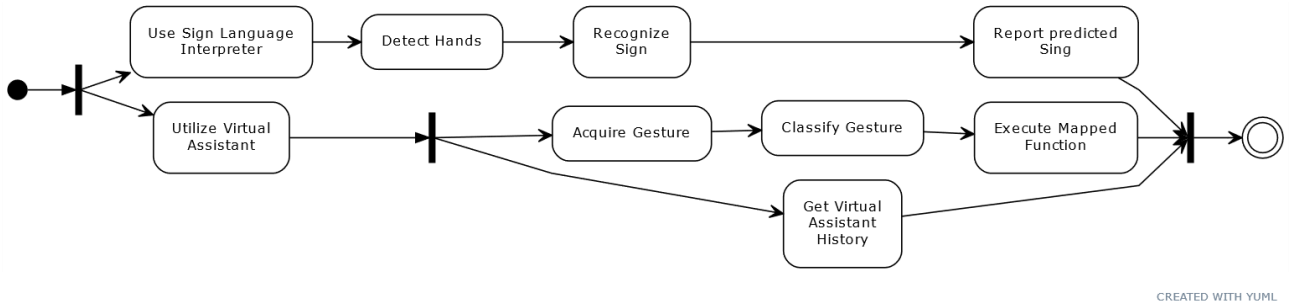

Figure 4

Figure 4 shows the scenarios surrounding the sign language interpreter and the virtual assistant as an activity diagram.

3.5.5 User interface - navigational paths and screen mock-ups

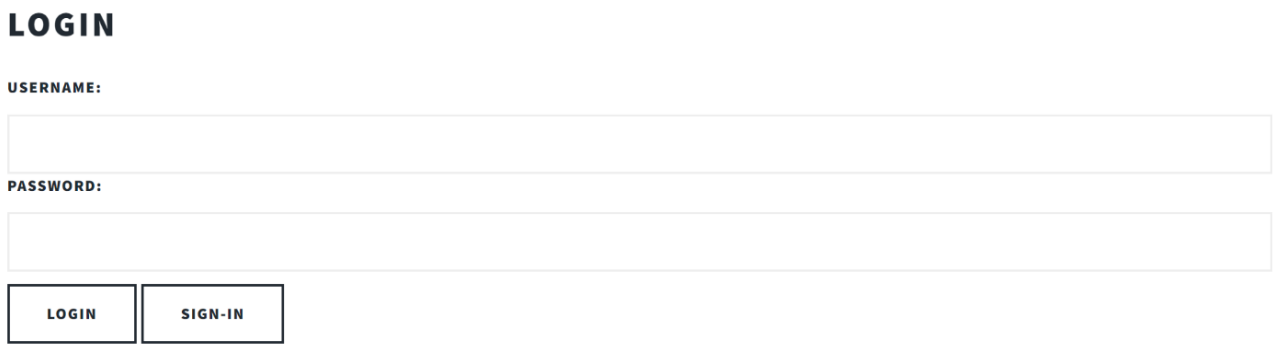

Figure 5 (The login page)

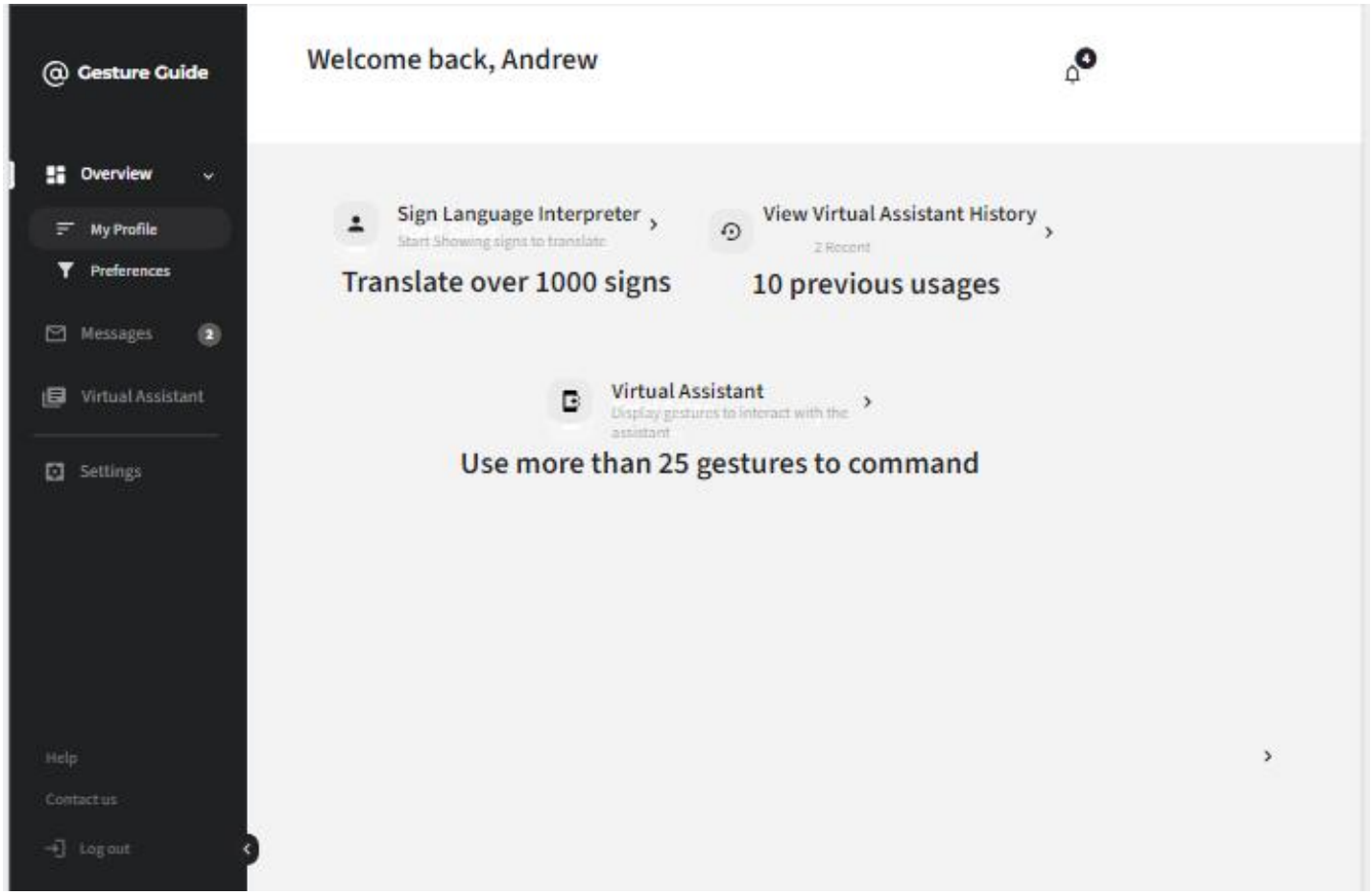

Figure 6 (The main page of the Gesture Guide app)

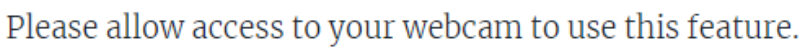

Figure 7 (Demo screen when the webcam access is denied.**)**

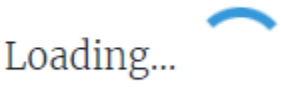

Figure 8 (Demo screen while it is initializing in the background.)

A: 0.99

Figure 9 **(**Hand gesture demo screen.**)**

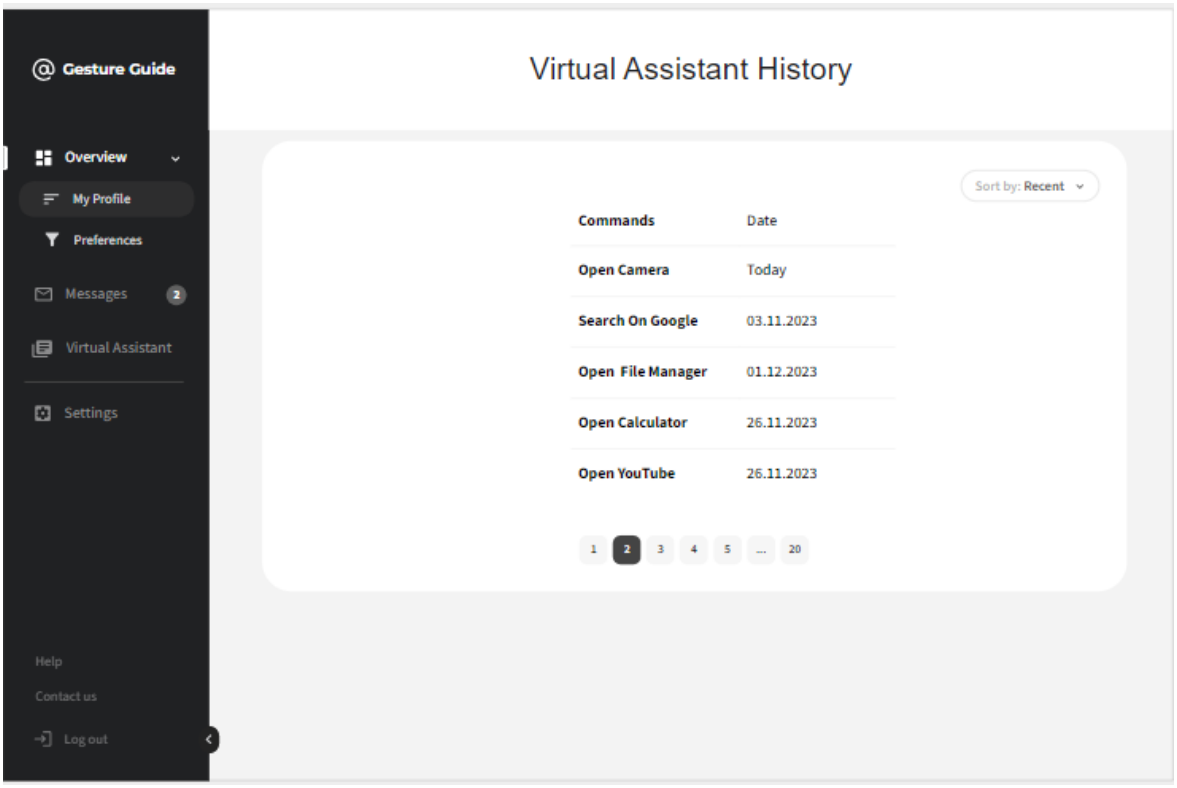

Figure 10 (Virtual Assistant history page in correspondence to “view virtual assistant history" requirement)

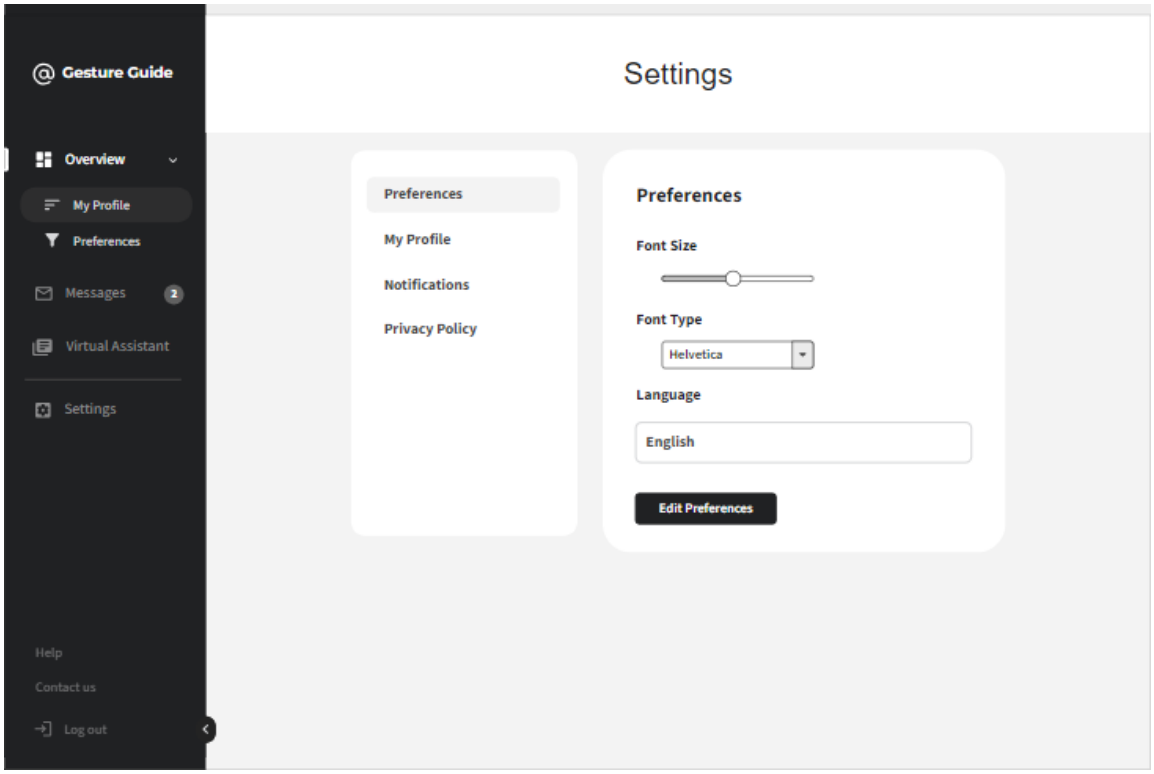

Figure 11 (view / edit preferences requirement as a part of the settings page.)

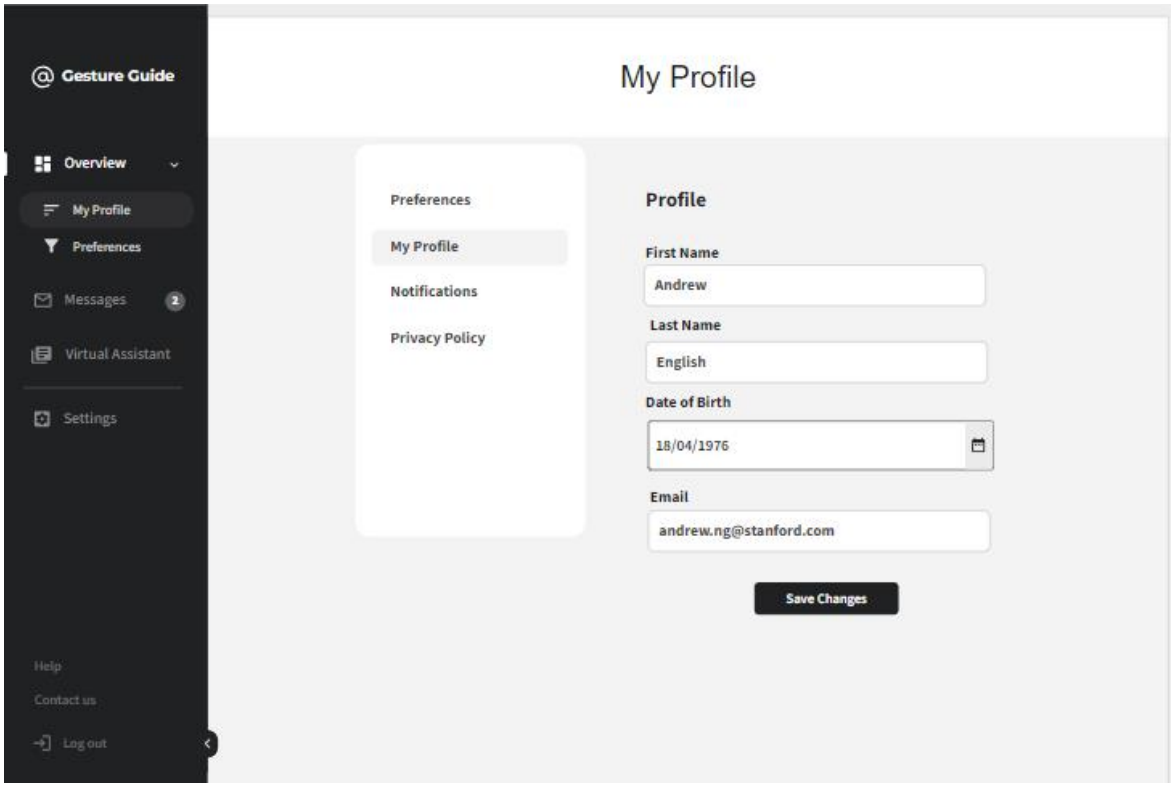

Figure 12 ("Personalized profiles" requirements as demonstrated by a personal profile page.)

4. Glossary

- **Activity Diagram**: A diagram that is used to demonstrate the general workflow of a software application.
- **Authentication**: To identify a user by checking their credentials.
- **Authorization**: To give privileges to users based on their roles in the application.
- **Class Diagram**: A diagram that is used to visualize the system as an aggregation of classes or objects in the domain of OOP.
- **Classification**: In machine learning, the process of determining which class a sample belongs to.
- **Credentials**: Personal information that is used for authentication of users. Generally, a combination of username and password for login.
- **Convolutional Neural Networks**: A unique type of neural networks that are heavily used in image classification problems. Gets its name from an extra convolution layer.
- **F1 Score**: A combination of recall and precision scores, generally the preferred method for classification model evaluation.
- **Model(Machine Learning)**: An algorithm that is used to predict on real life problems like estimating house prices.
- **Artificial Neural Networks**: Machine learning models that mimic the nervous system by using layer of nodes called neurons and interaction between layers of neurons.
- **Recall**: A model evaluation metric that focuses on number of false negative predictions (Type 2 error).
- **Precision**: A model evaluation metric that focuses on false positive prediction (Type 1 error).
- **Use Case Diagram**: Types of diagram that is used to show actors and actors' interaction with the system.
- **UML**: Stands for Unified modelling language, the universal methodology that is used for software modelling.
- **TensorFlow**: A machine learning library that is used for deep learning and construction of deep neural networks.

5. References

- Sommerville, I. (2016). Software Engineering, 10th edition. Pearson Education Limited. ISBN 10: 1-292-09613-6
- Bruegge, B et al. (2004). Object-Oriented Software Engineering, Using UML, Patterns, and Java, 2nd Edition. Pearson
- Martin, R.C . (2002). UML for Java Programmers. 1'st edition. Prentice Hall