Bilkent University

Department of Computer Engineering

Senior Design Project

High Level Design Report

Project: Signify

Team Members: Ali Taha Dinçer, Çağlar Çankaya, İrem Ecem Yelkanat, Muhammed Naci Dalkıran, Sena Korkut

Supervisor: Ayşegül Dündar

Jury Members: Shervin Arashloo & Hamdi Dibeklioglu

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

1. Introduction

In society, people who are hearing impaired and/or speech impaired have difficulty expressing themselves and communicating with other people because most people lack knowledge of sign language. Even though the improvements in technology have changed the way people live and made the lives of people easier by, for example, transforming mobile phones from sound devices into multi-functional devices, communicating with people who are hearing impaired and/or speech impaired continues to be a problem in many areas including social and technical contexts. These communication activities include social life, healthcare, career development, and education. Furthermore, as a result of the Covid19 pandemic that started in late 2019, obligations regulated by most countries, including wearing face masks and social and physical distancing, have increased the communication and social challenges for hearing impaired people. For example, wearing face masks has led to some negative impacts in communication with other people as it eliminates speech perception by visual features through lipreading. Additionally, a considerable amount of face-to-face communications have turned into virtual communications, which results in more hardship for the hearing impaired and/or speech impaired people as some of the most used virtual communication services like Zoom, Microsoft Teams or Skype do not support sign language.

According to the World Health Organization (WHO), 5% of the people on the earth are hearing impaired, which is more than 350 million people [1] and will exceed 700 million by 2050 [2]. Considering that, sustainability of the social lives of hearing-impaired and/or speech impaired people will be an essential issue in the future. Therefore, we propose a solution to this problem named Signify. Signify is a mobile application with the main aim of helping hearing and/or speech impaired people in their social lives by translating sign language into text along with speech and text-speech to sign language translation in real-time.

This report consists of the description of the proposed system including purpose of the system, design goals, proposed system architecture, subsystem services, and other analysis elements such as various factors in engineering design and teamwork details.

1.1. Purpose of the System

The main purpose of the application is to provide a channel among hearing-impaired or speech-impaired people and society. For this intention, two-way language transformation is ensured. The application should be able to translate a speech or text in English to sign

language through the instrumentality of animation for continuous visualization. For the reverse side, the application should also convert sign language from video to text and speech for the users who do not know sign language. For this, the user should be able to use the microphone and camera of their phones.

The application should also work close to the real-time speed to achieve practical and effective communication, especially for online meetings and conferences. Users should be able to reach the translation synchronically with the initial communicator.

For the straightforward usage of the application, guidance on the features should be done without language constraints. Guidance should be both with text and sign language to resolve the discordance happening from various impairments.

1.2. Design Goals

1.2.1. Usability

The main purpose of the application is to provide easy communication to people with impaired people. Therefore, the application should have a user-friendly interface for users to navigate and appreciate any feature within two seconds. As the target user is impared, universal icons and modern designs should be used. In this way, the application gains higher usability. Additionally, a manual is provided to the users to better understand the application and use the features straightforwardly.

1.2.2. Extensibility

Main features of Signify require utilizing machine learning models. As models can be updated with new data or the architecture can be changed, the application and its features should be extensible for any kind of change and improvement. When the learning model gets complicated, it's weights and corresponding storage requirements will also be increased. The application should be extensible in a way such that the updates cause any problems due to the user's storage limitations.

1.2.3. Performance

The performance of the application is one of the crucial parts. Machine learning models that are used for bidirectional translation should have high performance to achieve a successful communication system. Moreover, the models will have a complex structure, resulting in a

slow progress while processing text, speech or video. This process should be operated as fast as possible in order to provide a smooth communication experience.

1.2.4. Security

The application will store some personal information such as e-mail, password etc. For the best and safe experience, any kind of user related data should be prevented from leaking. Therefore, the design of the application should be encrypted in case of a data breach.

1.2.5. Compatibility

This application targets the communication of people with speech/hearing impairments. Our aim is to reach as many people as possible; therefore, the application should be available for all operating systems, including Android, iOS, and web platforms, to deliver the service to all users.

1.2.6. Privacy

The application will use data to train the learning models for a better communication performance. Especially, user data such as audio, video or personal text should be processed for translation purposes. Any personal information and video and audio records should not be used or shared with any other user, a third-party company, or application.

1.3. Definitions, acronyms, and abbreviations

- Online conference: A feature where people can easily communicate online through our application.
- Real-life communication: A feature where people can easily communicate in real life through our application.
- Bidirectional translation: Translation from text to sign language and from sign language to text.
- ASL: American Sign Language, the chosen sign language for the application.
- UI: User interface of the application.
- DB: Database that holds the user related data.
- ML: Machine learning.
- GAN: Generative adversarial networks that are used to generate an ASL animation from text information.
- Model data: Data that is used for the design and improvement of ML models.

1.4. Overview

Signify is a mobile application that aims to solve social problems regarding communication, understanding, and expressing for the hearing impaired and/or speech impaired people. By this means, Signify will help these people to improve their quality of life. In that manner, the application can be seen as a communication tool.

There are already existing applications such as “Hand Talk Tradutor para Libras” [3], “ASL Translator” [4], and “S.L.A.I.T.” [5] for the hearing impaired and/or speech impaired people. However, “Hand Talk Tradutor para Libras” does not contain real-time translation over the video, which creates a one-way channel between communicators. “ASL Translator” is a paid application and works on pre-given text and video translations, in which translations are pretty limited, and for every new word or phrase, the app needs to be updated. Finally, “S.L.A.I.T.” can only be used during video calls, and it is still in the beta phase.

Signify combines these apps in terms of their translational and conversational capabilities with tools like two-way translation. With Signify, users can easily communicate through video calls and use their mobile phones to real-time translate to both text and sign language in a day-to-day conversation. With this, hearing impaired and/or speech impaired people can join conversations even if the people in the conversation do not know sign language. Along with these innovations, we aim for Signify to be a swiss-knife for hearing impaired and/or speech impaired people during their everyday lives.

2. Proposed Software Architecture

2.1. Overview

Signify will mainly work through video, text and audio inputs followed by generated video and text outputs. It will be a phone based application adaptable to different operating systems. Signify is based on a simple client-server architecture. The application uses Firebase in order to store user related data and room data which have been generated by creating calls. All the requests by the user require changes in real-time, therefore Firebase Real-Time Database has been used. The application requires all the translations handled on the device in order to preserve efficient usage, therefore TensorFlow lite has been used in order to embed our translation models into devices. Each subsystem in the implementation respects the 4 layer

MVC architecture. The detailed software architecture and more are shown and discussed in the next sections.

2.2. Subsystem decomposition

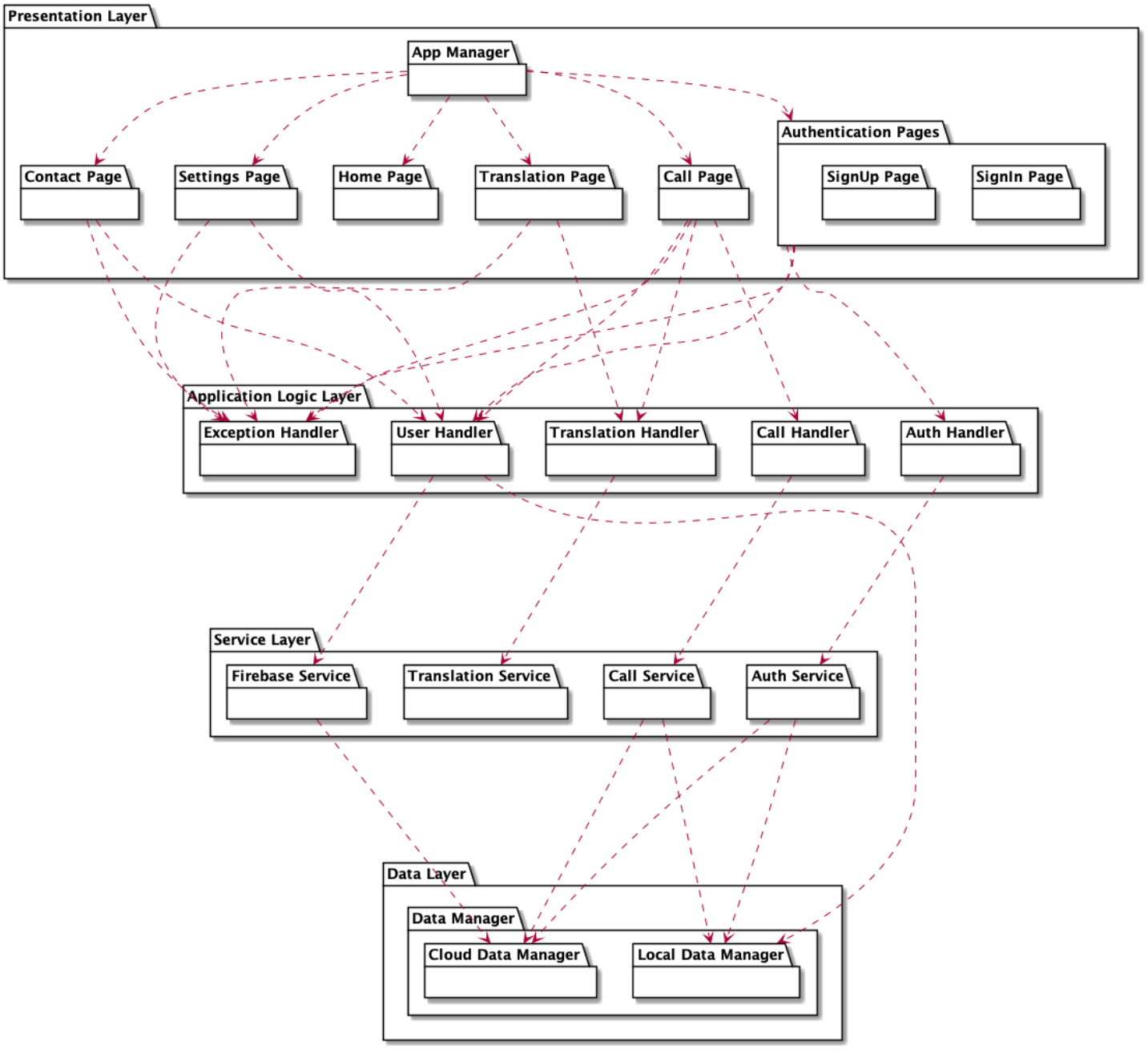

Figure 1: *Subsystem Decomposition Diagram*

2.3. Hardware/software mapping

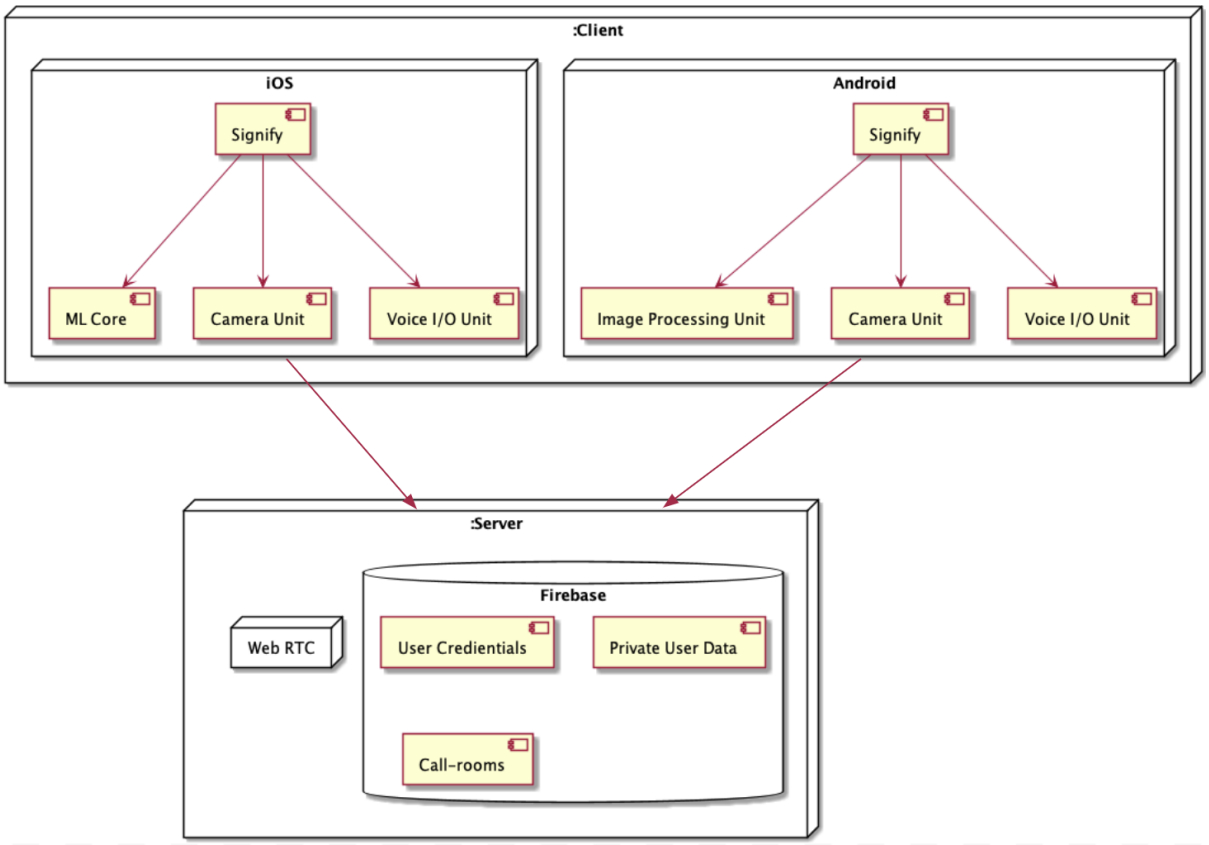

Figure 2: *Deployment Diagram*

The architecture of hardware/software mapping is shown in Figure 2. Signify architecture consists of two main parts: Client and Server. Server holds Firebase, which is the database that will be used to store the information of user credentials, private user data and call-rooms for video conferencing. Additionally, Web RTC will be used for real time communication on video conferences. It will be connected to the server along with Firebase and each process related to video calls will be operated based on the information stored on Firebase (i.e. call rooms). Every other operation is mapped on the Client side. Translation will be done by processing keyboard input, audio and video that are input by the users; thus, the application's main features require the usage of hardware systems such as camera unit and voice I/O unit for getting input data and generating a translated output. Signify will use those units on the Client side through an operating system. In addition, Signify will be adaptable to both iOS and Android. The application will also store the weights of machine learning models on the Client side, in order to utilize the retrieval time and fasten the process of translation.

2.4. Persistent data management

ML related data; such as pretrained model weights, should be stored in the locale of the user, because of simultaneous translation. The aims of the ML models are sign language to speech/text and text/speech to sign language simultaneous translation; therefore weights of the ML models should be accessed without server response delay.

User related data which are user's name, password, id are stored in the Firebase database to use them during signing in, joining meetings, inviting meetings etc. The users can have friends and he/she can invite them; therefore, user's friends data such as their ids, contact information for meeting, name etc. should be stored in the Firebase database also. However, the meetings' history, logs and outputs generated by ML models should not be stored in the database because these are personal information and storing them might cause privacy violations. Meeting related data mentioned in the previous sentence is stored in the local memory of the user.

2.5. Access control and security

Users have an account in Signify. They register the application with their name, password and email. These data are stored in the database. In the database, users have unique userIds generated automatically by the database. When the user signs in the application, they should enter her/his password and email. To control the sign in, email and password combination should be checked. The passwords of the users are encrypted in the database with authenticator key and hash map to avoid adversarial attack and account thefts. Even if the adversarials access the database server, they cannot access passwords without an authentication key. The length of the authentication key is n which cannot be broken in a real time (polynomial time boundary). For the real-time meeting, the microphone for users having no disability and video permission should be taken from the user. In the meeting user's can share personal information with other users; therefore, the chat, logs and personal data cannot be stored. As a database, we use Firebase. Firebase has its own defense system against adversarial attacks. It can be said to be trusted third party software. ML models process on the meeting data like sent and received messages, voice, video; however, these processes are done by the user's phone special unit called neural engine. Therefore, there is no privacy or security concern about the ML related process. Finally, during the meeting, user

communication is provided via Web-RTC API. This API is a trusted API and has its own defense protocol against adversaries.

2.6. Global software control

In the application, the same data might be requested for meeting the users' needs. For instance, a huge number of users want to add their friends. In this case, the database and application might encounter race conditions. To handle this issue, the server should be event-driven. Event-driven provides the server with opportunities to respond to client's request quickly and accurately. Event-driven programming is generally used in communication between server and client.

2.7. Boundary conditions

- **Initialization**

The application will be a Flutter project that will be shipped via downloading from App Store for iOS devices and Google Play Store for Android devices. After downloading and installing the application to the phone, the application can start by clicking the application.

When the user opens the application for the first time, the user will be navigated to sign in page that they can log in to the application or create a new account. After the authentication process is successfully completed and the user logs in to the application, they can navigate to the main functionality screens including home page, translation page, video call page, contacts page and profile pape.

The user must have proper Internet connection in order to use login/register functionalities, changing profile related data, manage the contacts and participate in video calls, because these operations need server connection establishment.

- **Termination**

The users can log out from the application by clicking on the logout button. When the user logs out from the application, all the data that is stored locally on the user's mobile device will be deleted permanently. When the user closes the application without logging out, when they open the application again, they will be logged in to the application automatically.

- **Failure**

While using the application, it is possible to face some errors, but it is significant to eliminate these problems before the users encounter them. In the implementation of Signify, our aim is to provide users with an extremely reliable system by utilizing exception handling mechanisms as efficiently as possible. However, the users still could encounter some failures that can happen due to the quality of Internet connection, mobile device's hardware systems and server errors. When the user faces network error, the system will disconnect the user from video call if network error could not be solved in 1 minute. In case of server errors, the application cannot connect to the database which may result in some failures in the application such as not being able to login or not being able to create/join video calls.

3. Subsystem services

This section includes the detailed explanation about each service given in the subsystem decomposition diagram shown in Section 2.2 / *Figure 1* of this report.

3.1. Presentation Layer

This layer is the starting point of the client side components which renders the UI of the application, handles the interaction between the user and the application and controls the state management.

3.2. App Manager

This is the topmost level component that handles the state management and navigation mechanics of the application. The purpose of the component is to handle several different scenarios. The first scenario is to handle the flow of the virtually persistent data throughout the application. In order to implement this, the application uses a package named "Provider" which handles the data flow with the help of observables and subscription mechanisms. The second scenario is the rendering of different UI components along with navigating between them. In order to handle this, different UI components have been implemented as path variables which makes the components persistent throughout the runtime of the application, hence, errors regarding data flow during navigation has been handled. The last scenario is the handling of persistent data on local storage. In order to handle this, the application uses a

package named “Shared Preferences” which handles the persistent data in a native way that increases efficiency both in the name of speed and storage.

3.2.1. Contact Page

This component includes contacts of the user and interactions between them. Each contact has been shown in a List View and can be interacted by tapping. Tapping to any element in the list will navigate the user to a detail page where the user can see the clicked contacts information such as name and email and see the call history made with that contact. The Contact Page component also includes a search bar at the top where users can search for their contacts by their name or email and a floating action button positioned in the bottom-leftmost part of the screen where users can add new contacts by pressing the button. Finally, the topmost element of the List View is the “Pending Invitations” button where users can see, accept or reject invitations from other users.

3.2.2. Settings Page

This component includes information related to the user along with application settings. Users can change their mail, name, password and profile picture or sign-out or delete their account by this component. In addition to that, users can change settings related to theme and language of the application from this component.

3.2.3. Home Page

The purpose of the component is to show a live feed for users where the topics are related to latest technologies and news regarding hearing/speech impairment. In addition to that, users can start the tutorial mode from this screen or navigate to the original site of ASL where users can learn American Sign Language.

3.2.4. Translation Page

This component includes the live translation future of the application. The component starts the Camera and Audio inputs when the user navigates to the component. Users can select between different types of translations. Users can also replay the translations that were generated.

3.2.5. Call Page

This component is where the users can create/join calls and see the call history. Each call in the history can be interacted with by tapping which will navigate the user to a detail page. Users can see who participated in the respective call and information related to the call such as date, duration and participant count. Creating or joining a call will navigate the user to the In-Call page.

3.2.6. Authentication Pages

This section includes detailed information related to the sub package that hpşds authentication related components. Users who have created an account or signed up to their accounts can not navigate to these components without logging out or deleting their accounts.

3.2.6.1. SignIn Page

This is the first component that users interact with when they run the application for the first time. Users can create an account using this component or navigate to the SigUp Page component by clicking a button.

3.2.6.2. SignUp Page

This is the component where users who have an existing account in the system should be navigated. Users who accidentally navigated to that component can be returned to the SignIn Page component by tapping a button or users who forgot their passwords can navigate to Forgot Password view by tapping a button.

3.3. Application Logic Layer

This layer is the bridge between client related components and services related components. Interactions of the users will be captured by this layer and will be transferred to the corresponding services with the help of the components that are available in this layer. This section includes detailed information regarding logic layer components.

3.3.1. Exception Handler

The purpose of this handler is to handle runtime related errors. The reason or purpose of the error may differ. The application can have two types of errors. First error type is the “wanted” errors, for example if the user tries to change his/her email address and enters an

existing email in the server, the information of the user should not be changed and reported to the user via toast messages rather than crashing the application. The second type of error is the “unwanted” errors where something can happen during data flow or navigation. This type of error will result in immediate fallback to the Not Found View.

3.3.2. User Handler

The purpose of this handler is to handle interactions between user and other users and the user itself. By that means, the handler will handle adding or removing other users to the contacts and changing name, email, password or profile picture of the user. The handler will carry the requested action to the Firebase Service in order to complete the requests and carry back the payload of the actions to the presentation layer to inform the changes to the App Manager, hence results by changing the state and re-rendering of the UI.

3.3.3. Translation Handler

The purpose of this handler is to handle interaction of the user in the both Translation Page and In-Call page. Changing between different translation types, carrying the Camera/Audio input to the Translation Service, and carrying back the generated translation by the Translation Service to the respective pages are handled by this handler. Each change affect the presentation layer, hence the App Manager will change states and re-renders the UI.

3.3.4. Call Handler

The purpose of this handler is to handle call related interactions in the application. Creating and joining calls will result in actions related to changing the database stored in the Firebase and creating call instances on the webRTC component. Hence, the actions will be carried to Firebase Service and Call Service by this handler and regarding payload will be carried to the App Manager component which will result in navigating to In-Call screen and changing the state of the application.

3.3.5. Auth Handler

The purpose of this handler is to handle authentication related interactions in the application. The handler works not only in the interactions caused by Authentication Pages, but as well as each request sent to Firebase Service and Auth Service as each interaction needs a check to the user if the user is valid or not. The service carries actions to Auth Service during SignUp

and SignIn related interactions and actions to Firebase Service during adding or removing contacts and finally returns the payloads to the App Manager in order to change the state and re-render the UI of the application.

3.4. Service Layer

This layer is the final layer in the client side and handles the main business logic of the application. Each service is distinct from each other but correlates in runtime to handle workflow of the application. The section includes detailed information regarding services. Each service is responsible to accomplish the requested workflow and deliver the corresponding payload to regarding handlers.

3.4.1. Firebase Service

This service accomplishes the workload of Firebase related requests. As the whole application requires communication between Firebase, almost each operation in the application requires this service. Although authentication related actions also require Firebase Service, these actions have been made distinct from the service in order to reduce the workflow of the Firebase Service. In that manner, workflows of adding or removing contacts, creating or joining calls, changing user related data and deleting accounts will be done by this service.

3.4.2. Translation Service

This service accomplishes the workload of translation related requests. The service requires a decider in order to change its workflow which is the translation type (Audio to Sign or Sign to Text/Audio). The service uses TensorFlow Lite to run the pre-trained model on the client machine. Different model usage will be decided to be used by the given translation type.

3.4.3. Call Service

This service accomplishes the workload of webRTC related requests. Creating or joining a call with means of transferring the audio/visual of the users to other participants are handled by this service.

3.4.4. Auth Service

This service accomplishes the workload of authentication related requests. Basically, this service is a sub-service of the Firebase service, however, in order to reduce the workload of the Firebase service and make this service run on another thread, the services are seperated. Services such as signing in or signing up will expected to be handled by one time for this service, however, each request going to be sent to Firebase Service required to be checked in order to know about if the user has been legally authenticated or not, so each request to Firebase Service also uses Auth Service in order to be accomplished.

3.5. Data Layer

This layer encapsulates the Data Manager as a layer in order to represent data workflow as a different scheme. The detailed information has been given in the following subsection.

3.5.1. Data Manager

Purpose of this component is to handle the persistent data stored in the cloud and local storage. Whichever service wants to do any CRUD operation to the data has to communicate with that manager in order to execute the operation.

3.5.1.1. Cloud Data Manager

This manager communicates with the Firebase Real-Time Database Services in order to handle CRUD operations such as adding, updating or deleting a profile picture or creating / joining a call.

3.5.1.2. Local Data Manager

This manager communicates with persistent local storage in the user's device in order to handle CRUD operations. These operations include CRUD related to JWT Token, Firebase User ID, webRTC persistent tokens, call history and user settings such as theme and language.

4. Consideration of Various Factors in Engineering Design

To conduct a project, different factors should be considered to produce a product which provides great user experience, solves people's problems etc. The chief consideration is to organize and plan the project progress. Since it is a team-work project, team members should

collaborate and communicate with each other. The one part might have a huge impact on other parts; therefore, team members should consider the consequences of their actions. To ensure this communication and collaboration, we decided to follow agile and scrum methodology for developing the project. The team has weekly meetings to determine the road map of the week.

Another consideration is related to user experience. The application should be user-friendly. It has not only a modern user interface but also simple, flawless, and fast design to provide the best user experience.

Furthermore, the user should have no security concern. The personal and sensitive data should be protected against any adversarial attack. Nowadays, a lot of applications are exposed to adversarial attacks. The application should be protected against them. All personal and sensitive data should be encrypted and stored safely. Also, any non-required data which has no impact on application performance should not be stored or taken by the user. To comply with “General Data Protection Regulation” - GDPR and “Kişisel Verilerin Korunumu Kanunu” - KVKK, any personal and sensitive data should not be shared with third parties.

The last consideration of the project is related to finance. The application is determined as free so that everyone can use the application. The application should be affordable for not only users; but also developers. Therefore, all machine learning work is made by the phone’s gpu or special unit which is made for enhancing the computational power of the phone. If the server or any external service is used for these tasks, the server fee becomes much more expensive as the application grows.

	Effect Level	Effect
Cultural Factors	9	Change in learning model outputs due to cultural differences in spoken-sign language.
Public Safety	8	General Data Protection Regulation
Public Health	1	Impaired people’s health should be considered, however, it is not the main purpose of the application.
Public Welfare	2	Reducing the communication limits among impaired people.
Economic Factors	6	The application should be free.

Environmental Factors	1	It is not in the scope of this project.
Social Factors	5	Language barrier for hearing impaired people in the sign up process.
Global Factors	8	Translation is done only between English and ASL.
Technological Factors	10	Change in design of learning models due to the publication of new technologies.

Table 1: *Factors that can affect analysis and design*

5. Teamwork Details

5.1. Contributing and functioning effectively on the team

The most important factor of a proper teamwork is communication. As a team, we scheduled a time where everyone is available. Every week, we meet at least once and discuss the progress so far and regulate the future plans. New responsibilities are distributed to each team member considering the workload and previous experiences. While distributing the tasks, various factors are taken into account. Each team member has his/her own weaknesses and strengths in different fields, and different interests; thus, assignments of tasks are done accordingly. We have worked together before, so it is easier to decide and communicate based on our previous experiences. This way, we can bring forth the utmost effectiveness in each task.

In addition, we have scheduled the works that should be completed by a given that is set either by the department or our team. We give great importance to the deadlines and finish our tasks accordingly. Deadlines are also crucial for a functioning project and future progress.

5.2. Helping creating a collaborative and inclusive environment

The works are separated among team members; however, that does not mean we work only on the tasks given. When someone has a problem, or needs assistance, other team members become involved in the work to achieve better outcomes. Additionally, after a task is completed, it is being reviewed by other members. Collaborative reviewing is one of the significant facts of teamwork. One can overlook mistakes he/she has done; however, others can see and correct those mistakes. By reviewing each work done, we aim to improve our design and eliminate errors that can influence our future work.

5.3. Taking lead role and sharing leadership on the team

As part of the planning activity, the project goals are listed and divided into work packages (WPs) as shown below. Each work-package has its own leader and at least two students work on a WP. Although the contributions of the members are equally divided, leaders have the extra responsibility to regulate the process of the work packages they are assigned to.

WP#	Work package title	Leader	Members involved
WP1	Project Specifications Report	Çağlar	Everyone
WP2	Analysis Report	Ali Taha	Everyone
WP3	User Interface Implementation	İrem	Ali Taha
WP4	High Level Design Report	Naci	Everyone
WP5	Sign Language Translation (From Video to Text)	Sena	Çağlar
WP6	Sign Language Translation (From Text to Animation)	Çağlar	Naci
WP7	Database Connection	Ali Taha	Sena
WP8	Mid Testing	Naci	Everyone
WP9	Demo-Presentation	Sena	Everyone
WP10	Real Life Communication	İrem	Ali Taha
WP11	Video Conference Communication	Naci	İrem
WP12	Low Level Design Report	Çağlar	Everyone
WP13	Final Testing	Ali Taha	Everyone
WP14	Final Report	Sena	Everyone
WP15	Final Presentation	İrem	Everyone

Table 2: *List of Work Packages*

6. References

[1] E. McPhillips , “World wide hearing loss: Stats from around the world,” *Audicus*, 14-Sep-2021. [Online]. Available: https://www.audicus.com/world-wide-hearing-loss-stats-from-around-the-world/. [Accessed: 05-Oct-2021].

[2] “Every 4th person to suffer hearing loss by 2050: Who,” *Down To Earth*. [Online]. Available: https://www.downtoearth.org.in/news/health/every-4th-person-to-suffer-hearing-loss-by-2050-who-75718. [Accessed: 05-Oct-2021].

[3] “Hand talk translator - apps on Google Play,” *Google*. [Online]. Available: https://play.google.com/store/apps/details?id=br.com.handtalk. [Accessed: 05-Oct-2021].

[4] “ASL translator - apps on Google Play,” *Google*. [Online]. Available: https://play.google.com/store/apps/details?id=com.asltranslator. [Accessed: 05-Oct-2021].

[5] “S L A I T. real-time sign language translator with AI.,” *S L A I T. Real-time Sign Language Translator with AI.* [Online]. Available: https://slait.ai/. [Accessed: 05-Oct-2021].