Bilkent University

Department of Computer Engineering

# CS491 Course Project

*Vox*

## Project High Level Design Report

Furkan Başkaya, Berk Kerem Berçin, Atakan Sağlam, Oğulcan Çetinkaya, Sarp Ulaş Kaya

Supervisor: Shervin Rahimzadeh Arashloo

Jury Members: Dr. Shervin Arashloo and Dr. Hamdi Dibeklioglu

December 18, 2021

# 1 Introduction

Sign Language is a nonverbal language that people with hearing and speaking disabilities use to communicate with the rest of the world. It is built upon the visual use of hands, eyes, face and mouth. It is a complete, natural way of communication that shares similar linguistic properties as spoken languages. Sign language is adapted by deaf communities as a useful means of communication [1]. It is an integral part of their lives. According to an article on Human Rights Watch (HRW) there are more than 70 million people around the world that depend on sign language to communicate [2]. With the help of this technique these people are able to learn, work and be included in their local communities. It helps to ensure disabled people live a life on an equal basis with non-disabled people.

Even though deaf people use sign language to communicate with other people, the majority of the population don’t know how to use sign language and this creates a communication barrier between the deaf community and the rest of the population. According to HRW’s research on deaf people’s rights, lack of awareness on sign language causes deaf people to struggle while accessing public services all around the world. Deaf communities experience additional struggles in essential parts of life such as health, education and justice because of face to face communication problems [3]. Only a small fraction of these services have reasonable adjustments for disabled people to benefit from them. A recent study led by the British Deaf Association states some of the key areas affecting the deaf community with their relationship with the public. There is a lack of awareness in booking, insufficient interpreters, and a lack of understanding about deaf people and their levels of written English [4]. As a team we are determined to meet the needs of the deaf community.

One of the best ways to reduce the communication barrier that the deaf community is facing is by teaching people how to communicate with sign language. There are courses that people can attend and learn sign language but they are not always a feasible solution. In modern life not everyone has enough spare time and money to spend on such courses. Instead, people can use mobile applications to learn new languages. These kinds of applications are very common for spoken languages like English, Turkish etc. but it is not the case with sign language. Our project idea started out when we realized the lack of efficient sign language teaching applications.

Supervising students and performance measurement is very important in teaching sign language to verbal language users. It is because sign language depends on body movements and learning these movements precisely is crucial. We would like to address this problem by providing a service where we will track the hand movements and gestures of the user and then process them to check how well they have learned the sign language, measure their performance, and provide appropriate feedback based on how well they use the correct signs and gestures.

## 1.1 Purpose of the System

Vox is a service which aims to improve the communication between the deaf community and the rest of the world by teaching how to use sign language to its users. Our name represents what we provide for our users and it is Latin for voice. Our application will become the voice of the deaf and the mute communities. The main goal of this application is to be an American Sign Language (ASL) recognition engine and teach ASL to its users by asking them for the correct gestures which it then captures and translates to sign language, determining whether the user has used the correct gesture or not, using nothing but a simple camera.  With this technology we can provide an effective learning tool for any individual that wishes to learn ASL.

This application will feature a handful of different activities that will teach ASL to its users in fun and engaging ways, and they will all be accessible via a simple and clean user interface that can be easily navigated. The application will track the hand movements of its users during these activities to measure their performance, providing appropriate feedback based on how well they use the correct signs and gestures to complete the activities. The current mobile applications that teach sign language do not provide a tracking for hand movements for a smooth and well-supervised learning experience. There are prototypes which can currently understand up to 100-200 sign language sentences [5], but it is not enough yet to be used as a supervised sign language learning tool.

We plan to develop customized machine learning models for hand tracking, human pose detection and face gesture tracking. In addition to that we want to apply advanced neural network models to further improve the efficiency of our service such as YOLO, which is a Convolutional Neural Network (CNN) algorithm. As for the datasets we have found several and ready-to-use open-source datasets but we plan to further improve our system by training it with user data if the user gives permission. As we get more users, our model will improve on the fly. We will also apply computer vision techniques, image processing and deep learning.

Our mission is to radically improve the life standards of deaf and mute people via helping them be understood by non-disabled people. We wish to become their voice.

## 1.2 Design Goals

### 1.2.1 Usability

The simplicity of our design is one of the most crucial aspects of our program in terms of its usability. The easier it is to navigate the user interface and menus of our application, the less likely it will be for any user to have a hard time getting used to. The process of learning how to use the application will be minimal for our users, and each of the application’s features will only be a few clicks away for them to use. To be more specific for instance, we will allow the users to participate in any course or challenge with at most two clicks, access the tutorial video on how to use the application with a single click, and send feedback with at most two clicks. Furthermore, everything on the user interface will be easily readable. Grammar errors will be avoided and contrasting colors will not be used for texts or icons and their backgrounds. There will also not be any non-functional elements on the user interface that may falsely give the impression that they are functional pieces, i.e, no solely decorative elements that look like buttons or text fields.

### 1.2.2 Reliability

The functionalities of our application should be reliable in providing what they promise to provide to the users, that is, an easy and efficient way to correctly learn ASL. To accomplish this, we will make sure that our machine learning algorithms are as accurate as possible, and output consistent results for as much of a wide range of inputs as possible. To be more specific, we are aiming for the translations done by the system to be at least 90% accurate on average.

### 1.2.3 Supportability

Our application will be developed for the Android operating system. To maximize its supportability, we will make sure that it will be runnable on almost every device that supports this operating system, and on as many versions of Android as possible. The specific hardware components of the devices will not affect the availability of our application for them. The one exception to this will be the camera, since our application will need to use it for its features. The performance of the program may increase or decrease with respect to these hardware specifications, but they will not restrict the user from being able to use the application on their phone.

### 1.2.4 Efficiency

We will make sure that our program responds to the user input in an instant, and return the appropriate results based on the input as quickly as possible. To increase the efficiency of the program even further, we will also have it use as few resources as possible while maintaining a steady performance. We will achieve this by handling the translation process at the server side in order to lower the battery consumption. The application will be translating the provided footage in real-time, and during this process, the real-time footage received by the device will be sent to the server in less than 3 seconds, and the server will also respond in less than 3 seconds.

### 1.2.5 Accessibility

To maximize the accessibility of our application, it will be crucial for us to ensure that the download and installation processes of our application will be as simple as they can be, and that the application is available for as many users to download and install on their devices as possible. To achieve these goals, we will have the application available for free on the Google Play Store, where it will be just as simple to download as every other application with the click of a single button. As an alternative, we will also have it available for download on our website, https://s-guy-descend.github.io/vox/.

### 1.2.6 Privacy/Security

It is absolutely essential to maintain the safety of the users' privacy in application design. An extensive, thorough and detailed testing stage is just as crucial to spot any security oversights that may cause the users' privacy to be jeopardized as it is to find and fix bugs. Not only does it undermine the code of ethics in application design for a developer to fail to provide a sturdy set of safety protocols for the users of the application, but it may also cause backlash from the public that will tarnish their reputation, and possibly even further repercussions such as lawsuits. In our case, the oversights and/or shortcomings with the privacy and security measures of our program may end up causing the recordings of the users to be leaked, and we will make sure to build our application in a way that will prevent that. When it is first used, the application will ask the user for permission to use their device's camera to avoid recording them without their consent and violating their privacy. As the user will also be informed of, the application will only use the footage it receives for its own functionalities. It will not share any information gathered by the recording other than what is relevant, which is the translation and the assessment of the sign language being used in front of the camera. it will only share it with the user themselves and no third party groups or applications.

### 1.2.7 Stability

Handling errors and extreme conditions will be a must for our application to ensure its stability. We will do our best to build our application as stable as possible, keeping the possibility of the users encountering any software errors or broken features, or experiencing soft or hard crashes to an absolute minimum. To achieve this, we will ensure three things:

- We will make sure that the system displays appropriate error messages whenever the user doesn't follow what they are supposed to do. The user will always be properly informed about what they have done wrong and what the correct course of action is.
- The system will be able to handle any error that is the result of the user's own actions. No functionality or design element of the application will fail to work or load properly when a user doesn't follow the application's intended design.
- If any exceptions or crashes occur due to the technical errors originating from the flaws of the work done by the developers, such as logic errors in the source code, it will be apparent to the user that the error is due to a bug that must be fixed by the developers and not due to an error of their own.

### 1.2.8 Maintainability

Planning for the future of the application after the end of its initial development cycle is essential. If the tools that we use to build the project and/or the design choices we make become outdated in the near future, causing the application to run into issues that may greatly hinder its functionality, it will not only ruin the experience for the users of the application, but it may also tarnish our reputation as its developers. In our case, we will maintain our application by keeping our server up and running as long as possible, while regularly checking the user feedback to see if there are any problems with the functionalities or user interface of the application, updating both the backend and frontend as necessary.

### 1.2.9 Modularity

Rather than simply maintaining the system, implementing and publishing updates for the application is another activity that we may stay involved with after the initial implementation process if we wish for our application to maintain the interest of its users for an extended period of time. With the emergence of new technologies and constantly evolving trends in application design and maintenance, implementing the new features and/or changing existing ones without taking the risk of causing errors with the rest of the application is essential. To provide such a safe model for ourselves to build upon after its initial implementation process, we will build our application in a modular manner, which means that its functionalities will be separated into different modules that will not affect each others' functionality.

### 1.1.10 Aesthetics

The design of an application is usually just as important as its functionality in attracting its users. While the simplicity of using our application will be one of the key factors in maintaining a loyal user base, striving to achieve this goal may sometimes lead to bland and uninteresting interfaces that the users must engage with to use the application. On the other hand, putting aesthetic concerns before the readability and usability of the application may result in the opposite, making the navigation of the application convoluted and difficult to understand in order to favor unique and artistic design choices. It is important to maintain a balance between functionality and aesthetics. In our case, we recognize the importance of our application being not only easy to use, but also visually pleasant. After all, it is an application that aims to promote and teach a medium of communication that is purely visual. It would not be ideal for us to have our users not wish to look at what they see while using the application.

## 1.3 Overview

In the following sections of this report, we will be describing in detail the proposed software architecture, the subsystem services, the various considerations about engineering design that we are taking into account while working on our project, and how we are managing to work as a team.

# 2 Proposed System

## 2.1 Overview

In the following subsections, we will provide details on the detailed subsystem decomposition, hardware/software mapping, persistent data management, access control and security, global software control, and boundary conditions of our application. We will introduce relevant diagrams to visualize the subsystem decomposition and hardware/software mappings of our system to clearly show the purposes of the subsystems, services, and hardware and software components, as well as the relationships between them.

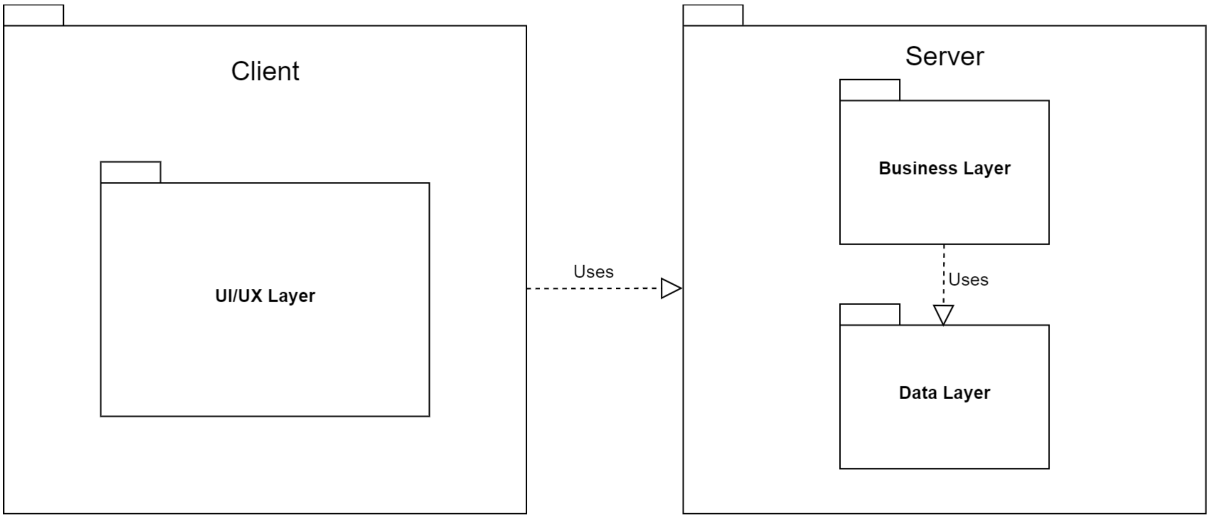

Figure 1: Overall System Decomposition (Client - Server Architecture)

## 2.2 Subsystem Decomposition

Vox will be built using a client-server architecture in which the logic resides in the server while the client side is where the view operations are executed, as well as some simple algorithms for processing the inputs sent to the server and the outputs received from it in order to display the correct results for the user. Inside the Client-Layer, which represents the client-side, Vox has a UI/UX Layer that has 2 subsystems: controller and view. This layer runs on the device of the user. The Controller subsystem controls the information to be displayed on the screens based on the user input and the server output using the aforementioned simple algorithms, while the View subsystem controls the user-app interactions and the navigation of the program between different screens. The Client Layer is connected to the Server Layer, which has 2 subsystems within it: the Business Layer and the Data Layer. The Business Layer handles the communications with the client side, as well as handling the database operations and utilizing the YOLO Algorithm with the data received from the client side as its input. The Data Layer stores the users' data, as well as the questions to be asked during the activities and their answers..

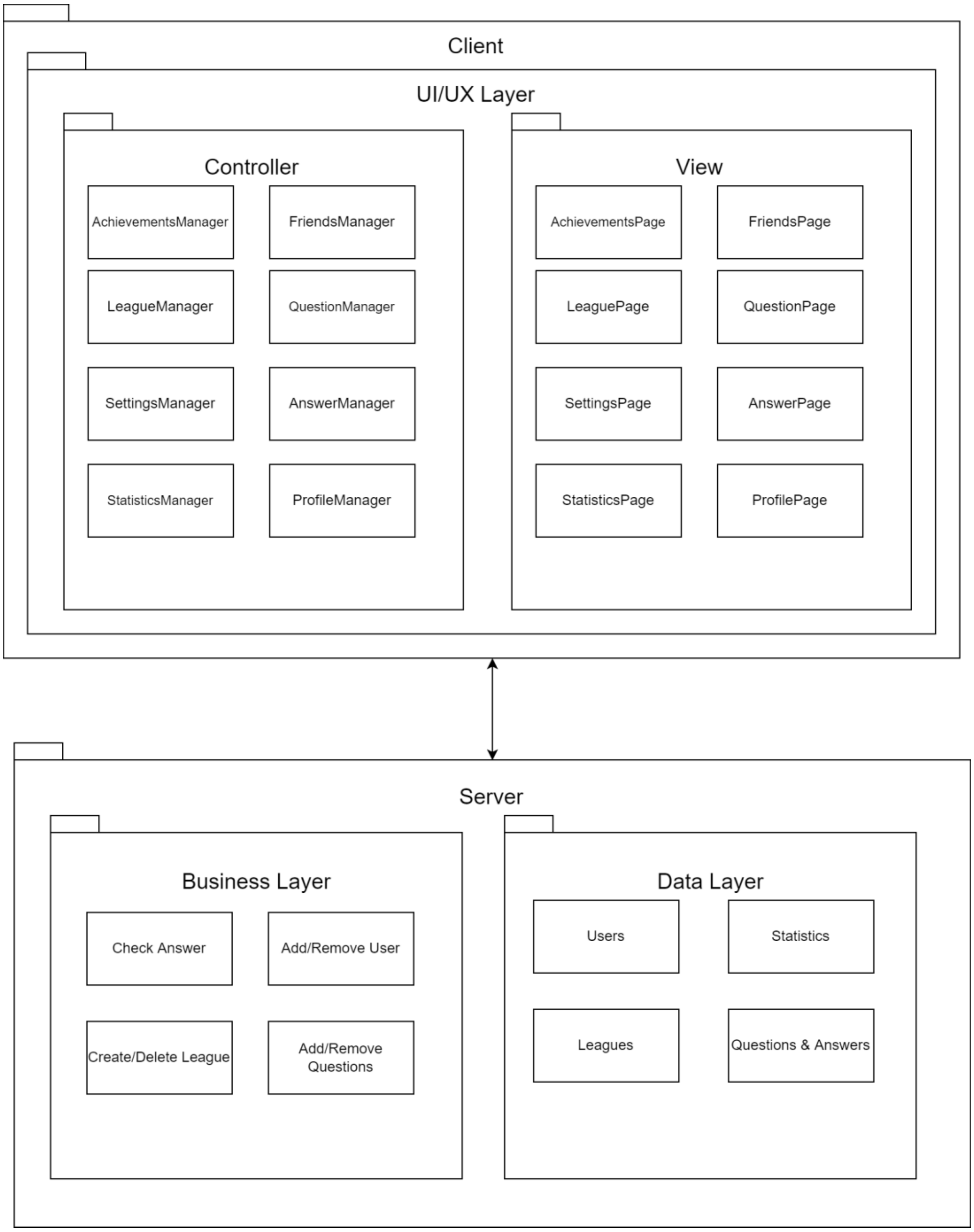

Figure 2: Detailed Subsystem Decomposition

## 2.3 Hardware/Software Mapping

Vox's application client will operate on the user's mobile phone. It will require access to the user's camera. Users will use their camera to record footage of what they desire to translate. After recording is done, this footage will then be sent to the server by the client via Rest API. The server will use a pre-trained YOLO Algorithm to assess the footage and send the result to the client by using Rest API. The database in the server will be used to keep track of the users' progress. The client will fetch data from the database to show users their information such as the last tutorial they completed and their score on various tutorials.

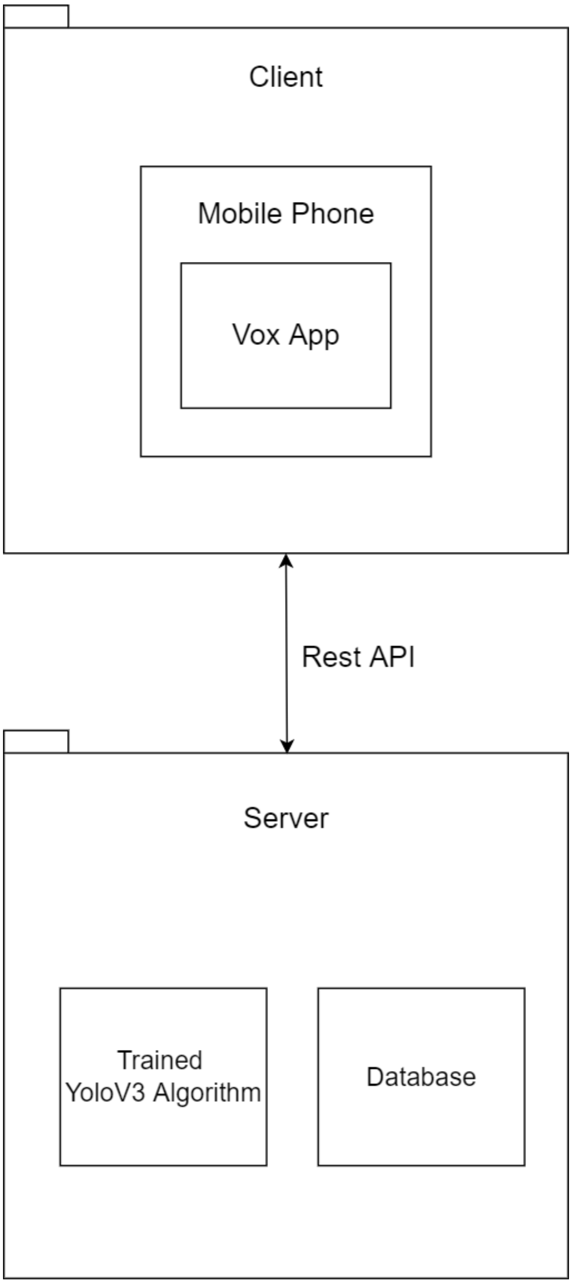

Figure 3: Hardware/Software Mapping

## 2.4 Persistent Data Management

Questions that will be asked by the application, their correct answers, user information such as their usernames and passwords, their statistics and their friend list will be stored in the database, whose connection with the application will start once the user tries to log in with a username and password, the latter of which will be encrypted. After the connection is established, the questions will be retrieved from the database and shown to the user on the client-side during activities such as courses and challenges, while the information of the user will be retrieved from the database and shown to the user on their profile page on the client-side. The information of the other users on the other hand will be retrieved and shown to the user on the friends tab on the client-side if the user is friends with them.

As for what the database will retrieve from the client-side, it will depend on the data that the client-side will receive from the camera of the device while the user participates in activities such as courses and challenges involving the aforementioned questions. The data that the client-side receives will then be sent to the database in encrypted form to be fed as input to the YOLO algorithm, whose output will then be returned back to the client-side, where it will be determined whether the user was successful in showing their answer to the device’s camera based on the YOLO algorithms output that was returned and the question that was displayed after being retrieved from the database. We will optimize our application to satisfy the data flow rate that will be needed between the client-side and the database.

## 2.5 Access Control and Security

Users will need to create an account via the registration page to use the application for the first time. If they wish to use it with the same account they have created later, they will have to log in using the user name and password combination they have created during registration. Without going through either of these authentication steps, none of the users will have access to any of the functionalities of the application beyond the login or registration screens. During authentication, the application will use a proper encryption scheme and an authentication framework to exchange the authentication tokens, having sensitive information of users such as their passwords stored in encrypted form.

Once authenticated, the users will only be allowed to see the names of the other users and their progress data such as achievements, league and streaks. To collect and manage user data, which will be critical for us to protect as best as we can from being leaked, we will use a secure cloud server for our database. While the aforementioned information concerning the users that will be available for everyone using the application to see will not be encrypted other than the passwords and be kept permanently unless the users decide to delete their accounts, the sensitive data of a user that must be protected against the rest of the users such as the data collected from the recordings during the courses and challenges will only be stored for a very brief amount of time and in encrypted form, at most as long as the user stays at the same course or challenge activity.

## 2.6 Global Software Control

Vox uses an event based control model in overall structure, and as a result, can be examined in two groups: client side and server side. Client-side creates events according to the user inputs. While using the application, most interactions with the app create events for the server to handle. Registration, login, following/unfollowing users, and attempting challenges are the main functions of Vox, and each of them creates an event on the client-side for the server to handle and/or respond to.

After the client-side sends a request to the server, it waits for an answer until it receives one or a timeout occurs. In the case of a timeout, the client side sends the same event once more. This way, the client-side application does not overwhelm the server with thousands of requests of the same type at once. The server waits for a client to reach out and send a request, handling each request that arrives separately since they can come from different clients. Based on the type of request, the server starts processing the event using scripts which also involve our machine learning algorithm, YOLO, to use for the requests to translate the user's movements on their device's camera. The server uses and/or updates the database based on the type of request. After processing is over, the server sends a response to the client-side.

## 2.7 Boundary Conditions

The boundary conditions for our application can be categorized into three categories: initialization, termination and failure.

### 2.7.1 Initialization

Users have to download Vox from the Google Play Store or our website: https://s-guy-descend.github.io/vox/. It must be downloaded to a mobile device that runs on the Android operating system, and has a camera, to be initialized properly. Upon using the application for the first time, users have to give Vox permission to use their device's camera. A connection between the server and the application is then established. Upon running the application, there are 2 options that the user can choose from before they can use the features of it. They can either create an account or log in with one they had already created. Accounts can be created with a user name and a password of the user's choosing, which can then be used to log in using the same account.

### 2.7.2 Termination

Upon getting terminated by the user, all the processes that have been created by Vox are also terminated. Memory that is used by the application is released, and any necessary information is saved before termination to the device's storage.

### 2.7.3 Failure

In the case of failure, Vox will save the last code block that caused the failure into a file, create a dump, and crash the application. After crashing, an error message will tell the user what went wrong and give them the option to send the error report to the developer team, along with an optional message they can add next to the report for them to see.

# 3 Subsystem Services

Vox application will be built by using the client-server architecture. Client side of the application will be operating on the user's mobile phone. Client will contain the UI/UX layer of the application. Model-View-Controller design pattern will be used to implement the application. The Controller and the View modules will be inside the client. Server side of the application will have two layers named Business Layer and Data Layer. Business Layer will contain the classes which handle the user's requests such as signing up, checking the answer of a question, adding or removing a question to/from a quiz. Data Layer will be used to store the data of the users such as their login credentials, their leagues, the questions they prepared, their answers and their statistics.

## 3.1 Client-Side

### 3.1.1 View Layer

View modules will be used to display the desired information by the users. They will have a counterpart class in the controller module which will handle what is needed to display the information, such as fetching data from the data layer.

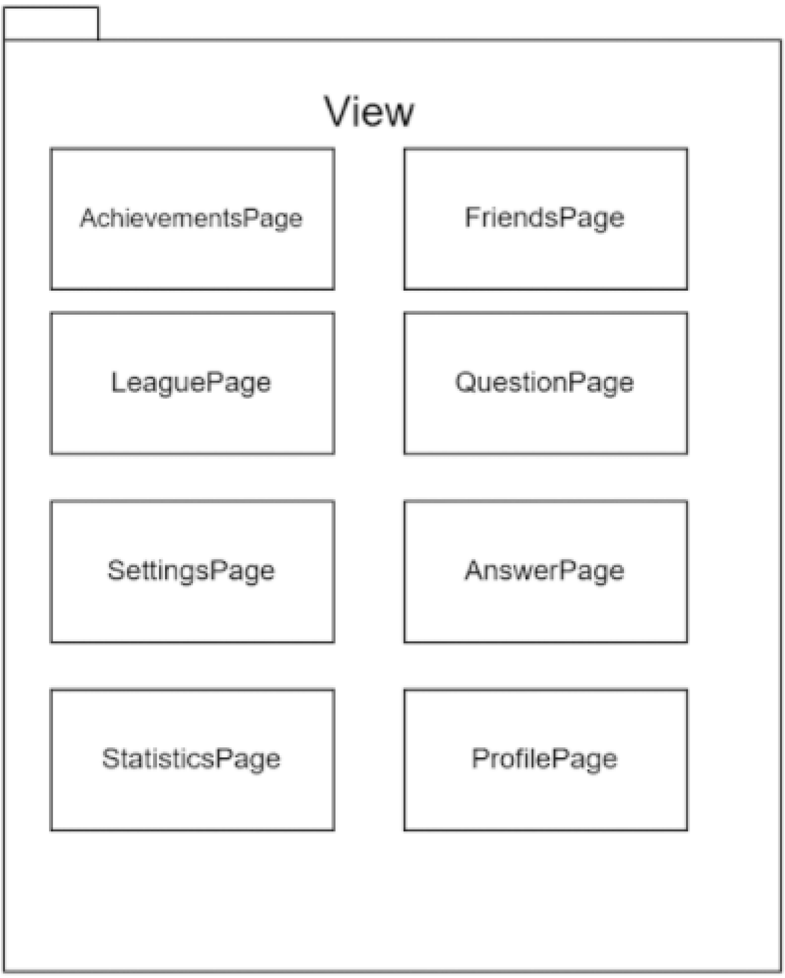

Figure 4: The View Subsystem

**AchievementsPage:** The page to display user’s their achievements.
**FriendsPage:** The page to display user’s their friends.
**LeaguePage** The page to display user’s their leagues along with their places in the league ladders.
**QuestionPage:** The page where the questions will be displayed to users.
**AnswerPage:** The page where the correct answers will be displayed.
**StatisticsPage:** The page to display the statistics of the user such as their percentage of completing a specific tutorial, the percentage of their correct answers and the total number of the tutorials they completed.
**ProfilePage:** The page to display information of the users and their friends such as their name and their league.
**SettingsPage:** The page to view settings and adjust them.

### 3.1.2 Controller Layer

The controller module will have the counterpart classes for each of the view module classes.

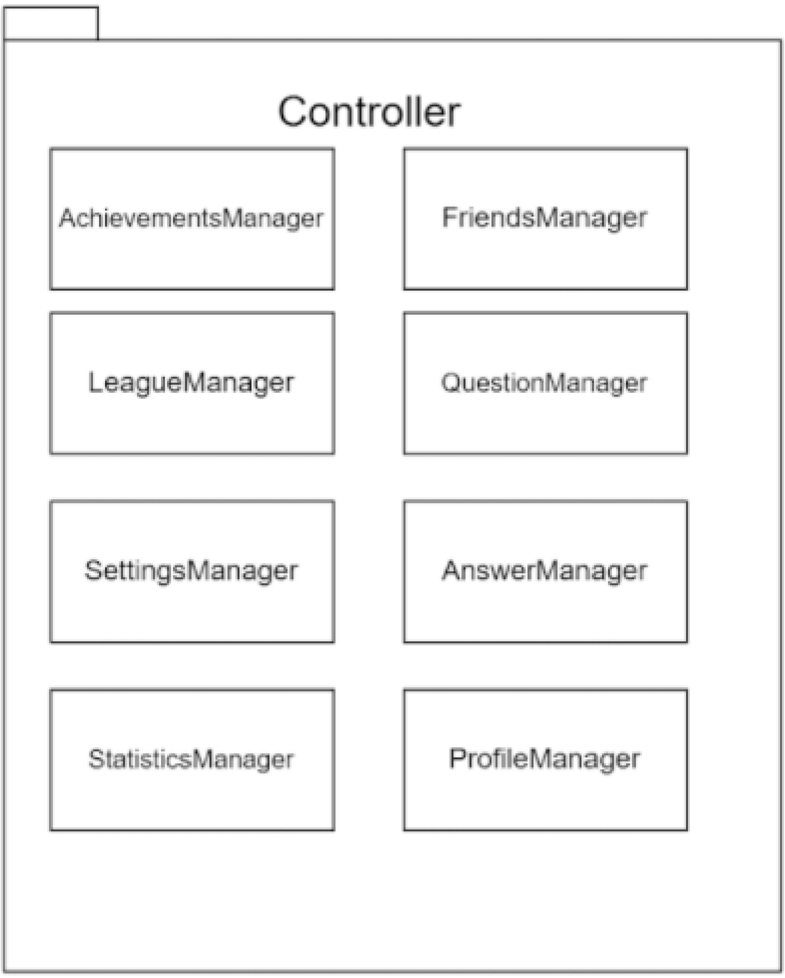

Figure 5: The Controller Subsystem

**AchievementsManager** class will be used to fetch data of the user from the server and provide it to AchievementsPage class to display it.
**FriendsManager** will be used to fetch the user’s friends data from the data layer. **LeagueManager** will be used to fetch the information of a specific user’s league from the data layer in the server.
**QuestionManager** will be used to fetch the questions from the database.
**AnswerManager** will be used to fetch the answers from the database.
**StatisticsManager** will be used to fetch the statistics of a specific user from the data layer. ProfileManager will be used to fetch the profile information of a specific user from the data layer.
**ProfileManager** will be used to fetch a specific user’s profile information from the data layer.
**SettingsManager** will be used to adjust the settings of the Vox application such as allowing notifications, changing profile settings or adjusting the volume.

## 3.2 Server-Side

### 3.2.1 Business Layer

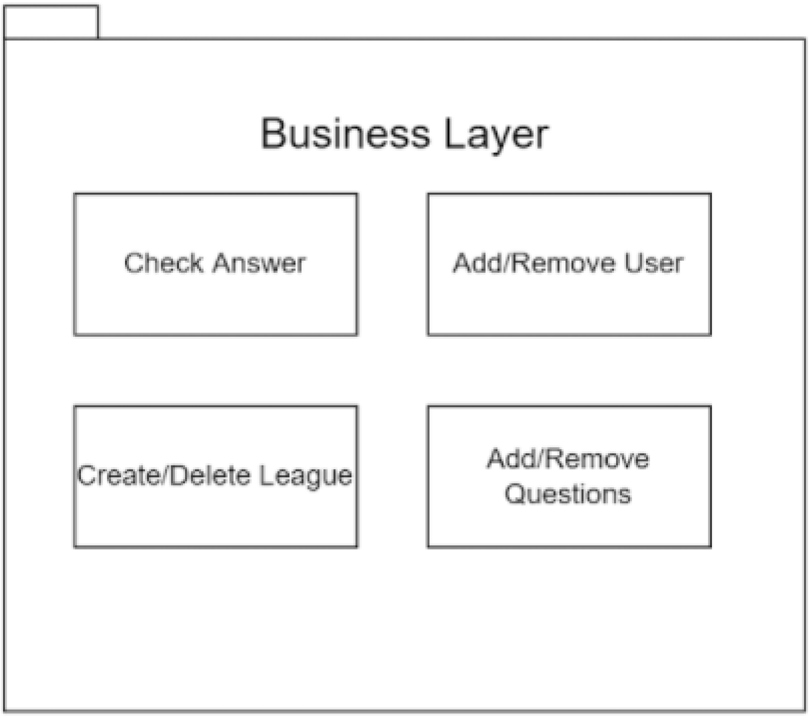

Figure 6: The Business Subsystem

**CheckAnswer** class will be the class where the Vox application will use the pre-trained YOLOv3 algorithm to assess the footage sent from the client. Then this class will send the result of the assessment to the client.

**Add/Remove User** class will be used to add or remove users to the data layer. It will modify the contents of the database by forming and sending queries to the database.

**Create/Delete League** class will be used to modify the contents of the League table in the database.

**Add/Remove Questions** class will be used to modify the contents of the Questions table in the database.

### 3.2.2 Data Layer

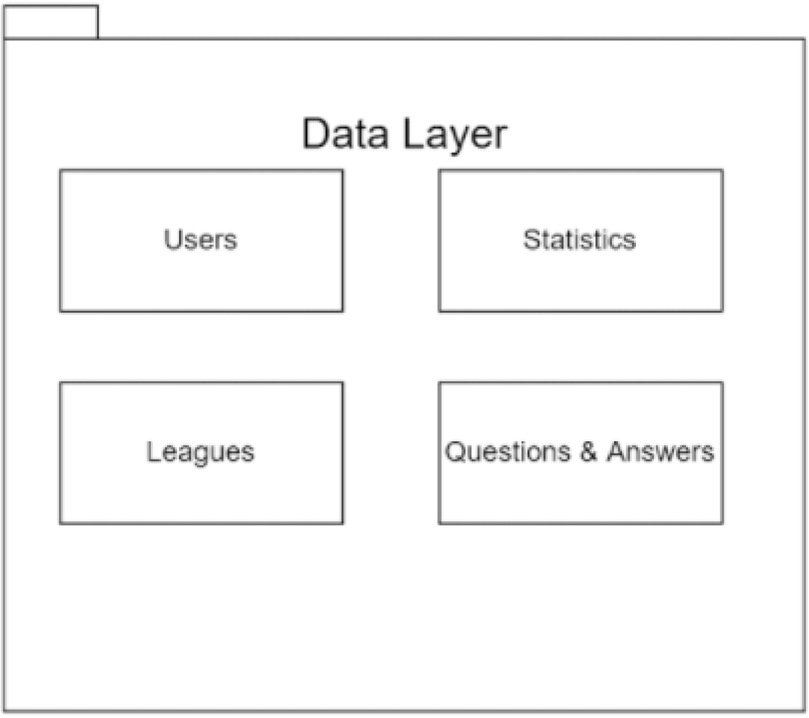

**Figure 7: The Data Subsystem**

**Users:** User information (username, password, friends of the user etc.) will be kept in the database.
**Statistics:** Statistics of each user such as their progress will be kept in the database.
**Leagues:** In a league, each participant will have a point and we will keep the points in the database.
**Questions & Answers:** There will be questions that are going to be asked to the users during quizzes. These questions and answers to them will be kept in the database..

# 4 Consideration of Various Factors in Engineering Design

## 4.1 Public Health

In terms of public health, the application is not likely to have major effects. Since the functionalities of the app are not directly associated with any health concerns, any outcomes related to the health of the users can be expected to be the very minor side effects of certain users using it rather than the expected results for the user base in general. These can emerge due to being physically active while using sign language.

## 4.2 Public Safety

It would be expected and desired for the application’s usage to have no effects on public safety in an ideal situation. However, if certain errors, oversights and/or shortcomings with the privacy and security measures of the program end up emerging, such as the user’s facial features being leaked, the safety of the users in public may become jeopardized.

## 4.3 Economical Factors

Sign language may prove to be a useful tool to utilize for marketing and transactional operations in the micro-scale. While it may not seem as a necessity due to the existence of subtitles and similar text-based assistance found in most brand advertisements, small businesses such as restaurants can find significant use in training their employees to learn sign language so that they can communicate better with some of their potential customers if needed, potentially raising their sales. Thus, the application may have a positive effect on the economic circumstances of such businesses.

## 4.4 Social and Cultural Factors

The cultural impact of the application will probably be one of the more significant wide-scale outcomes of the application, and this significance will keep increasing as more users teach themselves sign language using the application. One of the most important characteristics of an inclusive and socially just society is that it can provide equal rights and opportunities for all of their members, and allow its culture to be enriched by as many of these members. As expected, one of the key components of this characteristic is to make it as easy as possible for people with disabilities to live their lives with as little difficulties as those without the same disabilities. With the communication barrier that the deaf community faces on a daily basis being slowly lifted thanks to more and more users of the application becoming proficient sign language users, the effects of it on culture and society will be highly positive.

## 4.5 Global Factors

When the topic of globalization is brought up, language barriers are commonly regarded as one of the most significant barriers to cross, especially considering how easy it is for two people on the opposite sides of the world to connect with each other once this barrier is crossed. While the barriers between verbal languages usually come to mind first in these discussions, it is not difficult to estimate that it is likely for sign language barriers to also affect global communication, even if to a smaller extent. With the application helping its users cross this barrier, it will certainly have a positive effect on globalization.

## 4.6 Sustainability and Environmental Factors

Although the effects of a mobile application on the environment may often be overlooked, especially in cases similar to ours where the features of the application are not associated in any way with sustainability, the developers should still keep an eye out for the unexpected results of their design. If the application is more wasteful with the resources it uses than it should be, such as draining a device’s battery too fast, the effects of it on the environment per user will be so small that it would almost be completely neglectable. However, it would also add up as the user base grew more and more. While developing our application, we will be conscious of the power consumption of our application, as well as any other concerns around it related to sustainability and the environment, even if the resulting effects may be small and/or indirect.

# 5 Teamwork Details

## 5.1 Creating a Collaborative Development Environment

For managing team coordination and providing the necessary environments for collaborations and version control, we are using GitHub and are planning to possibly use Jira for additional functionalities and conveniences for work distribution. Additionally, the screen-sharing features of some of the applications that we are using for communication such as Discord provide us the ability to code the same portion of the project together in real-time.

## 5.2 Contributing to the Teamwork and Functioning in a Team

While developing a project of this size, it is naturally almost impossible for all members of the group to focus their attention on developing the same portion of the application due to time constraints, which leads to a distribution of individual tasks among the team members. As a result, personal responsibility becomes just as important as cooperative responsibilities for each member of the team. As individuals, each of us knows that despite not feeling a constant pressure on himself while working on any of his individual tasks, he must not procrastinate as it will not only have negative consequences for himself, but for the rest of the team as well. As a team, we know that while working together, everyone must pull their weight instead of relying on the rest of the team to decide and/or develop everything. By adopting these principles, we all make our contributions and function flawlessly as a team.

## 5.3 Establishing and Maintaining Communication

Communication within our team is key to work as efficiently as possible during the collaborative process. Using several platforms with varying functionalities for effective communication such as WhatsApp and Discord, our team meets regularly to inform each other of the progress on our individual tasks, as well as brainstorming about the next steps to be taken or discussing what to do during group tasks as we work on them together.

## 5.4 Distributing Work Evenly and Sensibly

Maintaining a balanced distribution of work among the team members is essential, and even though distributing the work as evenly as possible is important to achieve this goal, it should also be done with the strengths of each member taken into account in order to keep the overall progress as efficient as possible. By following this principle, we have managed to keep our efficiency at a maximum while working on the deliverables thus far, and we will continue to do so.

## 5.5 Taking and Following Leadership

Similar to how the work distribution should have the unique strengths and proficiencies of the team members as an essential factor, leadership in a team should stay dynamic based on the same considerations. Rather than having a constant member in the role of leadership throughout the development process, it is more beneficial for different members of the team to take lead and guide others at different points based on their interests, experiences and proficiencies. In our case, these shifts from one member’s lead to another regularly takes place even in the smallest scale possible such as while discussing and planning a minor adjustment in the design of our project or a small section of one of our work deliverables, let alone during the larger portions where the effects of our decisions can be much more significant. While we recognize the advantage of receiving guidance from our members based on their skills, knowledge and experiences, following leadership rather than taking it has its own set of rules as well, such as always analyzing the suggestions and intuitions of whoever is leading based on one’s own set of skills and knowledge. Since mutual respect is one of the key components of the foundation of our team, adopting these principles comes easy to us, never ignoring or devaluing one’s input based on their lack of knowledge or experiences of any topic while blindly following those that are more experienced and knowledgeable on it.

## 5.6 Being Supportive and Constructive

While professionalism is the key in achieving a high-quality final product, the genuine bonds between the team members are just as helpful in motivating everyone to work with each other and in unison. A group is only as strong as the bonds between their members, and the group members should be ready to provide each other emotional support and motivation when needed. Accepting that not each member can be as well-suited for each task or complete it as quick, and that it is natural for any of the members to make mistakes or struggle to remain productive under such an extensive and stressful development process is essential for us to avoid any toxicity within the team, and support each other as necessary. At times, a member may even have to put in some extra effort to compensate for the errors of one of his teammates. Rather than lashing out in these situations, being constructive keeps everyone's morale high despite the obstacles that we face throughout the development of our project.

# 6 Glossary

ASL: American Sign Language
SQL: Structured Query Language
HRW: Human Rights Watch
Vox: The name of our application, which translates to "voice" in Latin

# 7 References

[1] "American Sign Language", *National Institute on Deafness and Other Communication Disorders*, 2019, [Online]. Available:https://www.nidcd.nih.gov/health/american-sign- language .[Accessed: 10- Oct- 2021].

[2] "Without Sign Language, Deaf People Are Not Equal", *Human Rights Watch,* 2019. [Online]. Available:https://www.hrw.org/news/2019/09/23/without-sign-language-deaf-people- are-not-equal .[Accessed: 10- Oct- 2021].

[3] "Sign Language Key to Deaf People's Rights", *Human Rights Watch*, 2018. [Online]. Available: https://www.hrw.org/news/2018/09/23/sign-language-key-deaf-peoples-rights . [Accessed: 10- Oct- 2021].

[4] "Accessing Public Services: Issues for Deaf People", *BDA*, 2014, [Online]. Available:https://bda.org.uk/wp-content/uploads/2017/03/BDA_Accessing_Public_Services-Issues_for_Deaf_People-London_Boroughs_12-2014.pdf .[Accessed: 10- Oct- 2021].

[5] "SLAIT's Real-Time Sign Language Translation Promises More Accessible Online Communication", TechCrunch, 2021, [Online]. Available:https://techcrunch.com/2021/04/26/slaits-real-time-sign-language-translationpromises-more-accessible-online-communication/ . [Accessed: 10- Oct- 2021].