Bilkent University Department of Computer Engineering

CS 491- Senior Design Project

Visp

High Level Design Report

Group Members:	Enes Merdane	21600946
	İsmail Şahal	21703596
	Alptekin Önder	21602728
	Muzaffer Yasin Köktürk	21703552
	Muhammed Berk Köse	21704277

Supervisor:	Uğur Güdükbay
Jury Members:	Çiğdem Gündüz Demir
	Can Alkan

High Level Design Report

December 27, 2020

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.

1. Introduction

Since its emergence after the World Wide Web became mainstream, social media has been changing our lives in every imaginable way. In a short time span, countless social media platforms emerged and so many of them are still in use by many people of various demographics. As of 2020, it is estimated that 3.6 billion people use social media platforms worldwide [1]. Social media has become one of the most prominent ways in which people spend their time. It revolutionized the way we share information, ideas, memories and to this day, it still does. In addition to that, social media itself grows and evolves as new ideas and technologies emerge. One of the recent and exciting technologies that social media has not completely intertwined with is augmented reality. Augmented reality technology is an excellent way to enrich the environment around us to be used by people in interactive ways. Our aim in this project is to bring these two concepts together and integrate the perks of augmented reality into a platform for sharing and storing ideas. We are also very familiar with the idea of corporations using various technological platforms to engage with their audience and customers. Therefore we believe that augmented reality can also change how corporations and customers interact, bringing a new perspective to running advertisements. One of the promises of this project is creating a bridge between the digital world and real-life while allowing people to share an idea with others or save a memory related to a real-life object. We believe that introducing augmented reality to the interactivity amongst people will be the most innovative aspect of this project. A successful implementation would not only be a modern take on social media but also influence various technologies to be evolved in a similar fashion.

1.1. Purpose of the System

Visp is a social media application that utilizes augmented reality technology. The purpose of the application is to enhance the social media experience by making it more interactive with the physical environment. Using the dynamics of the social media applications, our application aims to increase the connectivity between people using common objects or locations which we believe to be a new experience for the users. During our daily experiences with our surroundings, we get connected with the objects and places around us. We believe that by allowing users to attach media objects and share what is obtained by this operation, Visps, with other people, they will be socialising in a different and more meaningful way than what is offered by today’s normal social media applications.

1.2. Design Goals

1.2.1. Reliability

Our application should be reliable. That is, the application should give consistent results when it is run in the same environment with the same inputs. If we don’t have it, then it means we are not controlling all the variables, then we have to reconsider our code and test it again.

1.2.2. Usability

Since it is a mobile application, the time to learn to use the application of the new users should be the minimum. Therefore, we should have a simple design for the application. We are aiming to achieve that by removing unnecessary pages so that the pages left are only the functional ones. Also, we have to be careful about how many functionalities we have put on one page because if it exceeds 2 or 3, that page will become too complex for the users.

1.2.3. Accessibility

The software should be accessible on most mobile platforms. We are aiming to increase accessibility by developing the application in Unity environment. That way, we will be able to deploy our product to both Android and iOS platforms. But to use the latest libraries in our application we have to give up some percentage of users by limiting the lowest version to use our application.

1.2.4. Extensibility

The design of the software should be done in a way that when new developers join the team, they should be able to adapt the code very easily. Also adding new functionalities and maintaining the code shouldn’t be difficult. To accomplish that goal, the project design should be modular so that each package will have its own job and the developer working on that package will not affect the other package’s functionalities.

1.2.5. Portability

For our application to be portable between different mobile platforms such as Android and iOS, we are planning to develop the application in Unity environment. In that way, we will be able to use most of our code for both Android and iOS.

1.2.6. Efficiency

Our application should be efficient in terms of the usage of RAM and time. The reason is, our application processes the images that are scanned by the user’s mobile device. Some computations may cost a lot of memory space so we have to utilize the jobs between the server and the user mobile device.

1.2.7. Scalability

Because we are aiming to create an application that will be used globally, the application should scale up while the number of users increases. However, in the short run, handling 100-500 users is enough.

1.2.8. Security

Because our application will store some user information, data protection should be considered during the design and implementation phase.

1.3. Overview

In this report, proposed software architecture, subsystem services, various considerations about engineering design and teamwork details are discussed respectively.

2. Proposed Software Architecture

2.1. Overview

In this section, we will be presenting the detailed subsystem decomposition of Visp. Below, you will find the diagrams that show this decomposition. Diagrams are important to understand which subsystem and service is responsible for which operations and the relationships between them.

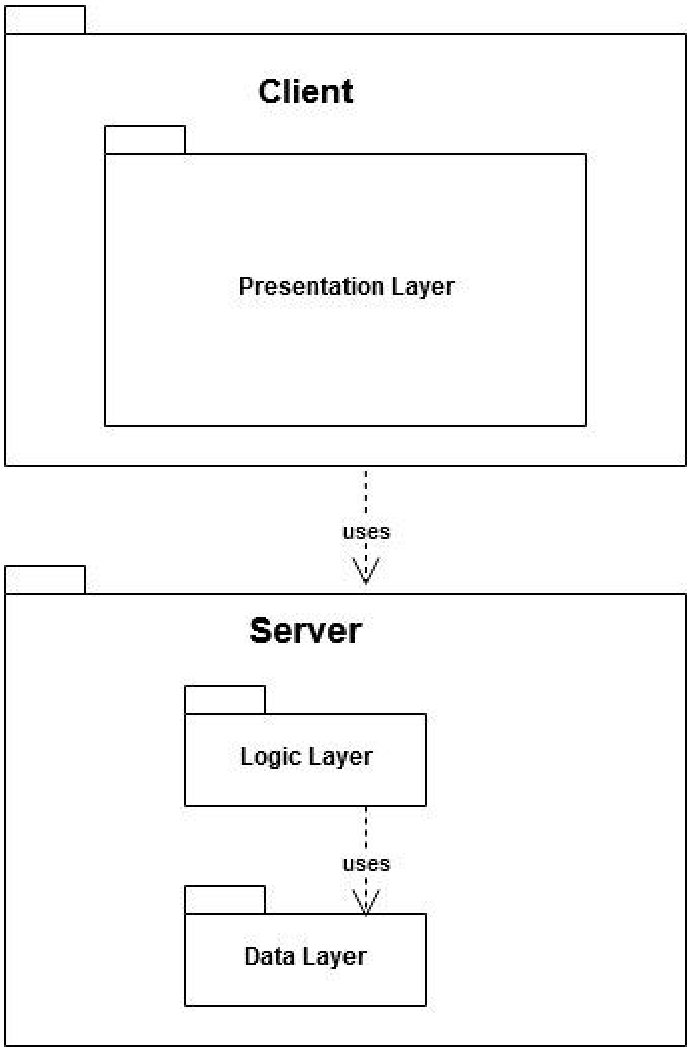

Figure 1: Overall System Decomposition (Client - Server Architecture)

2.2. Subsystem Decomposition

Visp will be implementing a client-server architecture in which the server side will be hosting the logic and the data services while the client side will be responsible for view operations for the user. Visp has a Presentation Layer that has 3 subsystems, controller, data and view. The presentation layer will run on the phone. The client will be able to take a picture and crop a part of it with CropImageManager and create a simpler data to send to the server. This process will be doable within the application’s client communication. Control subsystem will be connected to the Data Subsystem where the asset-app communications will happen. View subsystem will control the user-app interactions and navigation of the program between different screens.

Client Layer is connected to the Server Layer, which has 2 subsystems within it, which are Logic Layer and Data Layer. Logic layer will handle different processes such as communications with the client side, handling database operations and Vuforia. Data Layer will store the users data and their visps as well as the information about friends.

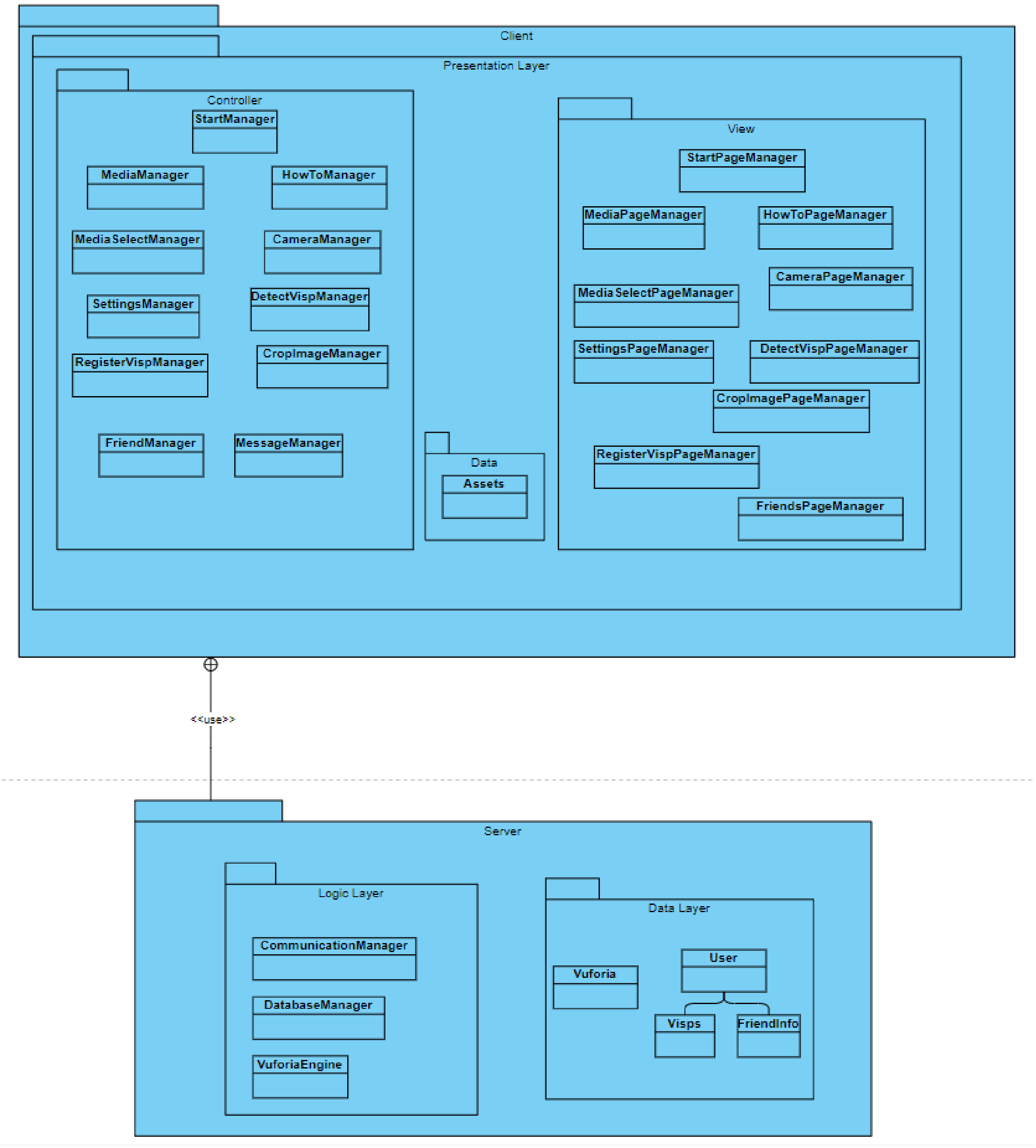

Figure 2: Detailed Subsystem Decomposition

2.3. Hardware/Software Mapping

The hardwares that will be used for the Visp project is 2 parts.

First part is the client part which is the mobile phones of our clients. These phones can use Android or IOS as operating systems. Visp will run on both operating systems and on their hardwares. We will use Unity's compile support for both platforms in order to make Visp work on both platforms.

Second part is the server side. Server side is used for all the computations for recognizing and scanning an environment. In these machine learning tasks we will use python libraries and tools. These tasks require fast and reliable computation power, therefore they will run on Amazon AWS servers. Also all scanned objects and Visps which are attached to them will be gathered in the database. Database and python packages for machine learning will always be in communication at the server side.

Client side and server will create their communication through Rest API, HTTP requests.

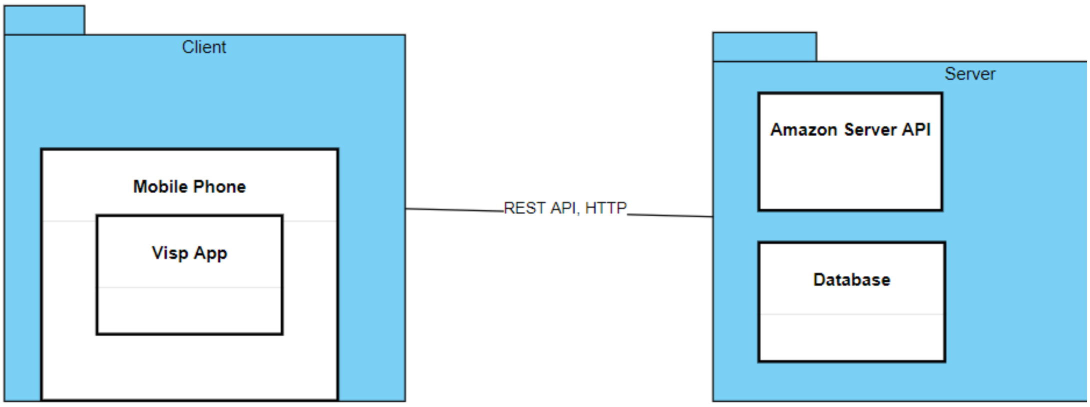

Figure 3: Hardware/Software Mapping

2.4. Persistent Data Management

The data that is going to be managed by the application include user information such as username and password, the messages that are saved by the users (visps) and the complementary data that links these messages with the locations and the backgrounds. A database is going to be used to store and retrieve these data. The application is going to connect to the database upon login and use this connection for other functionalities. The database is going to be designed in a way that it complements the user-oriented design of the application which is similar to other social media applications. All sensitive data will be encrypted and other standard security you know some security measures will be taken.

The application is going to use the aforementioned data to render the messages in real time using the input of the camera's phone. Since these messages are likely to be in multimedia format and are going to be retrieved from the database, the rate of data flow is a crucial aspect for the application. Therefore the application will be optimized to satisfy the data flow rate that will be needed between the front-end and database sides of the application.

2.5. Access Control and Security

Users need to create an account, post visps and comment on them. A user who does not create an account can browse the application to see the visps that are made public by users who posted them. The authentication process will be done via username and password. The application will use a proper encryption scheme and an authentication framework that were offered by Unity and third party libraries to exchange authentication tokens. The sensitive information of users such as passwords will be stored in encrypted form. A leak of sensitive information of our users is as likely as a breach of the cloud server of the database we are going to use.

Inside the application, users are allowed to see only the usernames and the descriptions of other users. Visps that are attached and made public by a user can be viewed by other users and users can comment on them. A user may or may not choose to display their username with a visp. Private visps can only be viewed by their creators. Public visps of a user can be browsed by their friends independent from the location.

Collecting and managing user data is a challenging and sensitive issue for a social media application like Visp. All the data management and related design choices inside the application and the database will be in compliance with KVKK (Turkish Personal Data Protection Law) [2].

2.6. Global Software Control

Visp uses an event based control model in overall structure. Softwares in this structure can be examined in two groups, client side and server side.

Client side is the side which creates events according to user inputs. After starting the app, most of the interactions with the app creates an event that the server should handle. For example, recognizing, scanning objects and attaching Visps are the main functions of the Visp app. All these functions create an event that the server should handle or respond to. After the client side sends a request to the server side, it waits for the answer until a timeout occurs. If a timeout occurs, the client side again sends the same event. In this way client side application does not overwhelm the server with thousands of requests in the same type.

Server side waits until a client reaches out and sends a request. Then according to request type, the server side starts processing the event. In order to precess, the server can run python scripts for applying machine learning to new scanned objects or recognizing the object which is sent by the client. Server uses or updates the database according to request. After processing is over, the server sends a response to the client. Server handles each request separately since each request can come from different clients.

2.7. Boundary Conditions

Boundary conditions for Visp can be categorized under three subjects, initialization, termination and failure.

Initialization

Users have to download the official Visp application from the google play store or ios store. It can be downloaded with a mobile phone or a tablet. In there users have to give required permissions to the Visp App. Then when the app first opens, a connection between server and app is established.

After the user starts the application, there are 2 ways to use the application. Firstly, users can use the visp without logging in. In this case it can only see registered objects in the application. Secondly, users should log in with an account. Users with accounts can use any features of Visp. If users do not have an account, it can be created with only id and password. It depends on the user to give more detailed information when signing up.

Termination

When the Visp app gets terminated by the user, all the processes which are created by Visp are also terminated. Memory that is used by Visp is released. Visp also saves any necessary information before termination to the device's storage.

Failure

In case of any failure, the Visp application saves the last code block which crashes the app into a file and creates a dump. If the same failure happens more than 3 times, it sends an error report to the Visp support team.

3. Subsystem Services

3.1. Client

3.1.1. Controller Subsystem

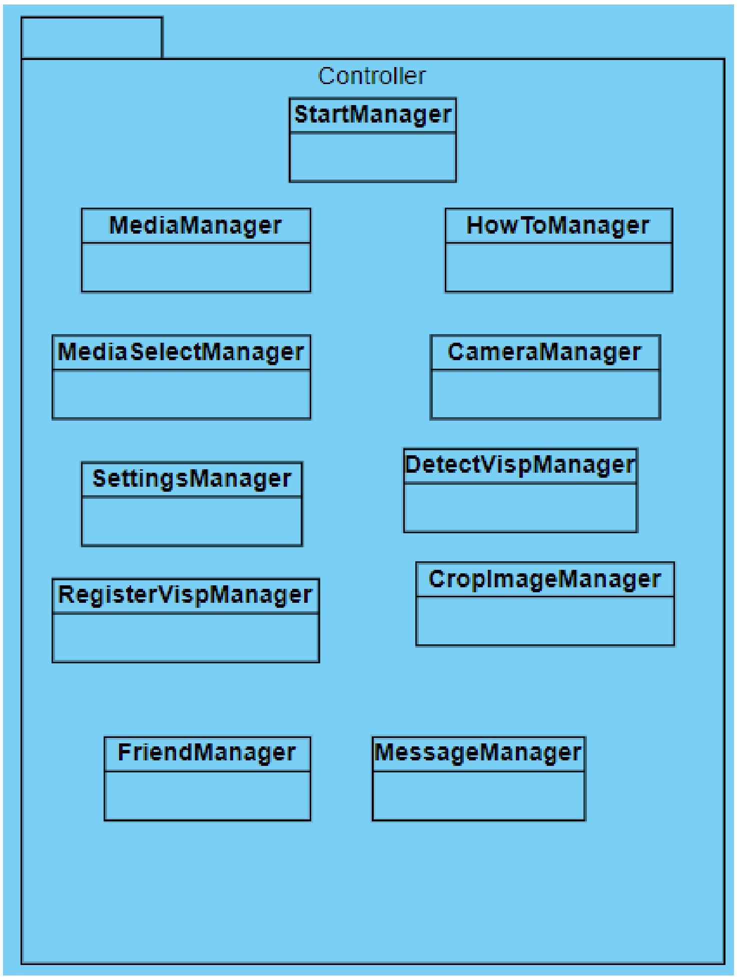

Figure 4: Controller Subsystem

startManager: This class is responsible for the startup process of the application.

MediaManager: This class is responsible for the user's operations with media objects in the application.

HowToManager: This class is responsible for mediating the tutorial page of the application regarding usage instructions.

MediaSelectManager: This class is responsible for the process of media file selection from the phone’s gallery.

CameraManager: This class is responsible for controlling the camera operations of the user.

SettingsManager: This class is responsible for operating the settings menu and saving these settings to the local data cache.

RegisterVispManager: This class is responsible for visp registration process.

DetectVispManager: This class is responsible for detection of the nearby visps in the phone’s camera.

CropImageManager: This class is responsible for the cropping of the images that are to be sent to the server side.

FriendManager: This class is responsible for interactions with the current user’s friends in friends list.

MessageManager: This class is responsible for managing the messaging between two users.

3.1.2. View Subsystem

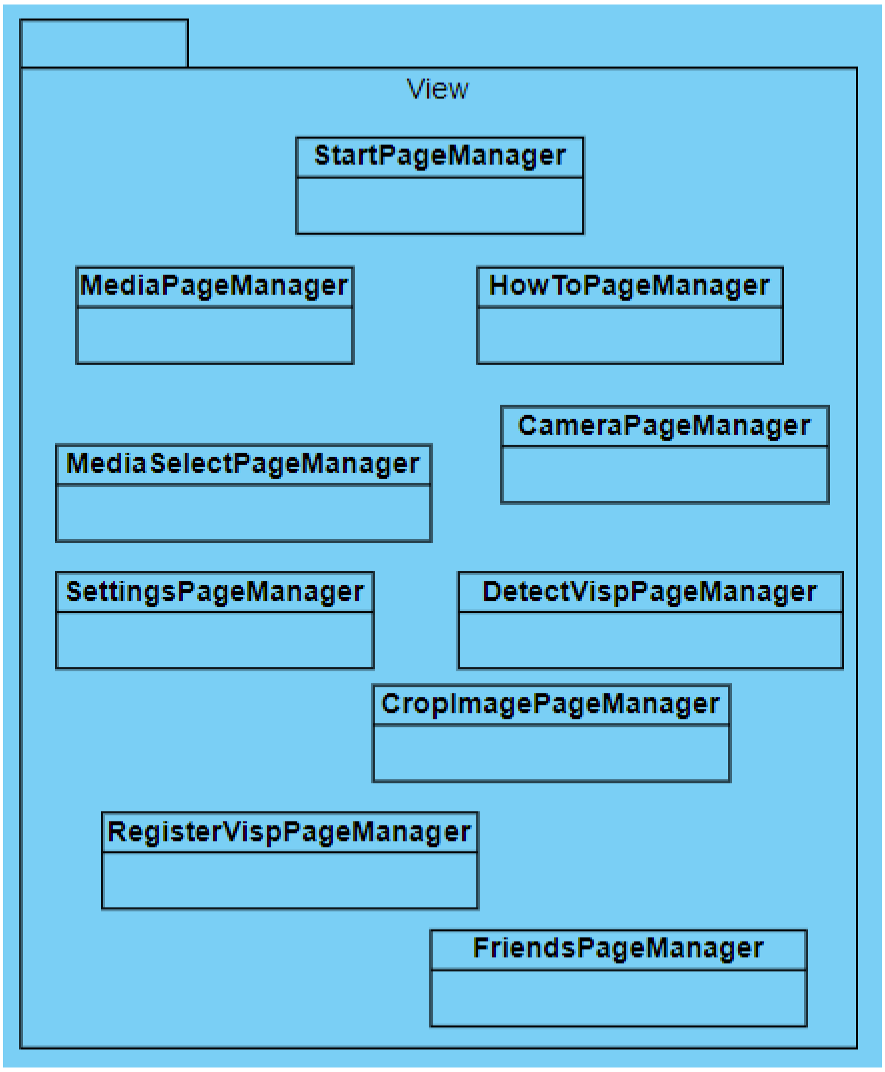

Figure 5: View Subsystem

startPageManager: This view will display the home page.

MediaPageManager: This view will display the media related operations.

MediaSelectPageManager: This will display a helper class after selecting a media..

HowToPageManager: This will display a tutorial page for beginners on how to use the app.

CameraPageManager: This will display the camera page. Connected to both register and detect Visp.

SettingsPageManager: This will display the settings page.

RegisterVispPageManager: This will display the page which is for registering a Visp to the database.

DetectVispPageManager: This will display the page which is for detecting a Visp via camera.

CropImagePageManager: This will display an interface which will be helpful while registering a Visp by cutting a taken picture.

FriendPageManager: This will display the friend page for users view, will be connected to the message page

MessagePageManager: This will display the messages between users and their in app friends.

3.1.3. Data

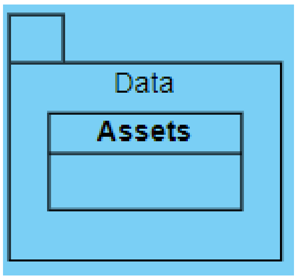

Figure 6: Data Subsystem

Assets: This will manage the Assets which are stored in the app such as stickers or 3D model for AR.

3.2. Server

3.2.1. Logic Subsystem

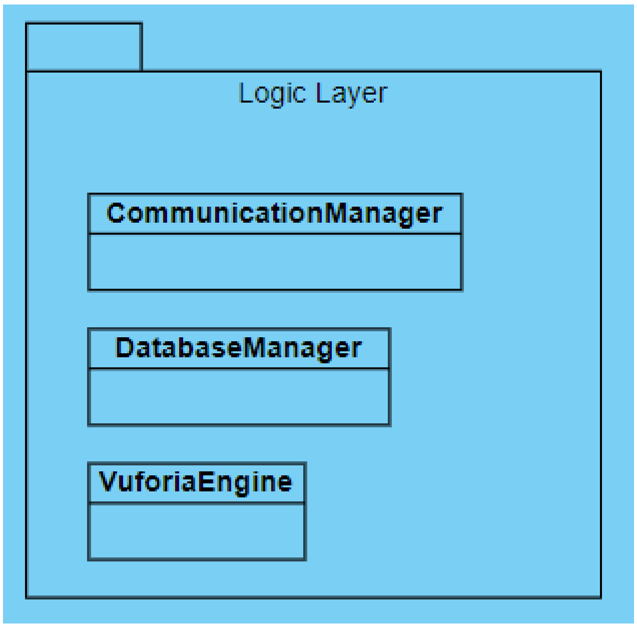

Figure 7: Logic Subsystem

CommunicationsManager: This class is responsible for communications between client side subsystems, logic layer and the data layer.

DatabaseManager: This class is responsible for data traffic between the logic layer and the data layer.

VuforiaEngine: This class is responsible for the machine learning tasks that are going to be performed for visp creation and recognition.

3.2.2. Data Subsystem

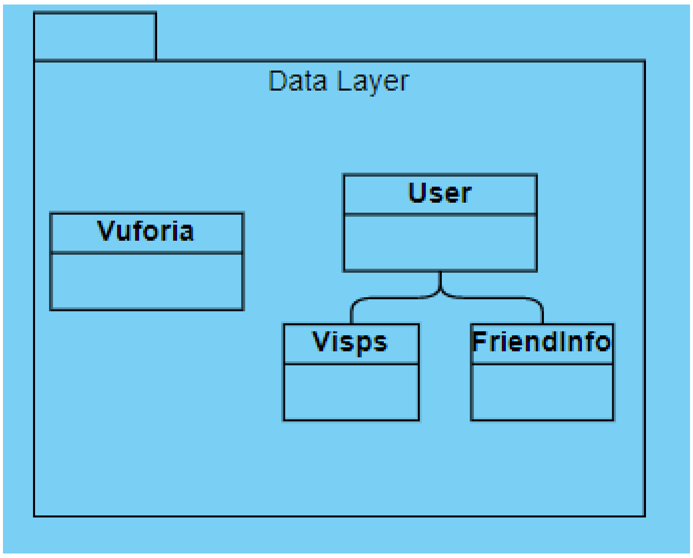

Figure 8: Data Subsystem (Server Side)

Vuforia: The machine learning models that are required for the logic layer to operate.

User: The user information that will be kept.

Visps: User created media files that are the backbones of our application.

FriendInfo: The information we keep for interactions between users.

4. Consideration of Various Factors in Engineering Design

4.1. Aesthetics

The aesthetics of the product defines the first impression of the users. Even before the functionalities of the product, users look at the feeling that the application gives them. Do they feel comfortable while using it, or do they feel like a stranger to the application? Do they use it like it's their first time using the application, so having difficulties learning, or does it come so easy for them to use the functionalities of the application. For these reasons, the aesthetics of the application should be considered in both the design and implementation phases of the application.

4.2. Modularity

Modularity is another important factor during engineering design. The reason is when there is a need for new functionalities to be added to the application or a need for changes in some functionalities of the application due to the emergence of the new technologies and mobile platforms, making these changes and additions without breaking other parts is a very critical issue. Therefore, the design and the implementation of the application should be done in a modular manner, that is, the functionalities should be separated into different modules that will not affect each others’ functionality.

4.3. Sustainability

Sustainability is another important point in application design. The designers should choose libraries and technologies that are not outdated and being developed by developers according to technological needs. If these are not taken into consideration during the design and implementation phase, after some time, the team may need to make big implementation changes on the project so it costs a lot.

5. Teamwork Details

5.1. Creating Collaborative Teamwork Environment

To create a collaborative work environment for the team, we are planning to be always in touch with each other via our communication channels such as Whatsapp and Discord. Also for the deliverables of the project, we are planning to keep at least one to three meetings a week to discuss what we have done, what are the problems we have faced, how should we solve these problems, what risks we have and what we should be doing for the next meeting.

5.2. Individual Contributions to Create a Collaborative Environment

We are working as individuals and duals for the deliverables of the project. We are planning to change the working duals for the parts of the project so that we will be in a dynamic manner. Also, we are planning to set some working as a group sessions to work together on some parts of the project. In that way, everyone will work with each other and contribute to each part.

5.3. Taking Lead in Different Parts of the Project

As we are creating the deliverables of the project, in the first stages these are the reports that we have been writing, the interests and roles of the group members become more clear. That is, on what part each member may contribute to the project at his best becomes clear. With that, each group member will take the lead on the part of the project where he will have more influence on the work and understand the job to be done better than others.

6. Glossary

Server Side: We will be running many core parts of our application (e.g. machine learning tasks) on a cloud server.
Client-Side: View part of the application that is running on a user's phone.
Vuforia: The AR library that we are planning to use [3].

7. References

[1] Statista. 2020. Number Of Social Media Users Worldwide | Statista. [Online] Available at: <https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users> [Accessed 11 October 2020].

[2] Personal Data Protection Law. [Online]. Available at: <https://www.kvkk.gov.tr/Icerik/6649/Personal-Data-Protection-Law> [Accessed 22 December 2020].

[3] Vuforia Developer Portal |. [Online]. Available at: <https://developer.vuforia.com/> [Accessed 22 December 2020].