



Bilkent University

Department of Computer Engineering

CS492 Course Project

Visp

Final Report

Enes Merdane, Alptekin Önder, İsmail Şahal, Muhammed Berk Köse, Muzaffer Yasin Köktürk

Supervisor: Uğur GÜDÜKBAY

Apr 30, 2021

Contents

1. Introduction	3
2.1 Functional Requirements	4
2.1.1 Scanning Mode Requirements	4
2.1.2 Registration Mode Requirements	4
2.1.3 User Functionality Requirements	4
2.1.4 Server Functionality Requirements	5
2.2 Nonfunctional Requirements	5
2.2.1 Performance	5
2.2.2 Scalability	6
2.2.3 Usability	6
2.2.4 Security	6
2.2.5 Availability	6
2.2.6 Modifiability	6
2.2.7 Legal and Regulatory Requirements	6
2.2.8 Accessibility	6
2.3 Pseudo Requirements	7
3.1 Firebase Database Design	7
3.1.1 Firebase Authentication	8
3.1.2 Firebase Database	8
3.1.2.1 Firebase Realtime Database	8
3.1.2.2 Firebase Firestore Database	9
3.1.3 Firebase Storage	9
4.1 Using Unity for Android Development	10
4.2 Using Vuforia on Unity	10
4.3 Firebase Integration	10
4.4 Maps Integration	11
6.1 Cloud Maintenance	12
6.2 Server - Database Maintenance	12
6.3 App Maintenance	13
6.4 Market Place Maintenance	13
7.1 Consideration of Various Factors in Engineering Design	13
7.1.1 Public Health	13
7.1.2 Public Safety	13
7.1.3 Economic Factors	14
7.1.4 Global Factors	14
7.1.5 Cultural Factors	14

7.1.6 Environmental Factors	14
7.2.Ethics and Professional Responsibilities	15
7.3.Judgements and Impacts on Various Contexts	15
7.4 Teamwork Details	16
7.4.1 Contributing and functioning effectively on the team	16
7.4.2 Helping creating a collaborative and inclusive environment	17
7.4.3 Taking lead role and sharing leadership on the team	17
7.4.4 Meeting objectives	17
7.4.4.1 Functional Objectives	17
7.4.4.1.1 System Functionalities	17
7.4.4.1.2 User Functionality Requirements	18
7.4.4.1.3 Server Functionality Requirements	18
7.4.4.2 Non-Functional Objectives	18
7.4.4.2.1 Performance	18
7.4.4.2.2 Scalability	19
7.4.4.2.3 Usability	19
7.4.4.2.4 Security	19
7.4.4.2.5 Availability	19
7.4.4.2.6 Modifiability	19
7.4.4.2.7 Legal and Regulatory Requirements	20
7.4.4.2.8 Accessibility	20
7.5 New Knowledge Acquired and Applied	20
7.5.1 Android Application Programming with Unity	20
7.5.2 Adobe Photoshop CC	20
7.5.3 Augmented Reality Application Development with Vuforia	20
7.5.4 Firebase with Unity Applications	21
8. Conclusion and Future Work	21
9. Glossary	21
10. User Manual	22
10.1 Login Page	22
10.2 Registration Page	23
10.3 Main Page - Add Visp Screen	24
10.4 Main Page - Scan Object Screen	25
10.5 Main Page - Scan Area Screen	26
10.6 Main Page - Menu Screen	27
10.7 Profile Page	28
10.8 Messages Page	29

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 to create 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 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.

2. Requirements Details

2.1 Functional Requirements

2.1.1 Scanning Mode Requirements

- The application should be able to recognize objects and places that are previously registered into the system.
- The application should display related attachments of recognized objects.
- The application should display a list of attachments if there is more than one for a specified object.
- The application should allow users to share videos and photos from the application to share on other social media platforms(Instagram, Whatsapp, etc).
- The application should be able to scan QR codes.

2.1.2 Registration Mode Requirements

- In registration mode, new objects should be able to be introduced to the system.
- In registration mode, new places should be able to be introduced to the system alongside their GPS information.
- The system should give positive or negative feedback about the registration process to users.

2.1.3 User Functionality Requirements

- Users should be able to save the media files or moments from recognized objects to their phone storage.
- Users should be able to attach media files(video, sound, image, gifs), texts, links to objects or places.
- Users should be able to take a picture or video of any moment in the scan mode.

- Users should be able to swipe between attached files if there is more than one for an object.
- Users should be able to add their friends to their friend list inside the application
- Users should be able to send any attachment or file from our application to their friends from the friend list
- Users should be able to see nearby memories if their GPS is active.
- Users should be able to open the application without authentication.
- Users should be able to only use scan mode without authentication.
- Users should be able to like other people's attachments to objects or places.

2.1.4 Server Functionality Requirements

- The server should try to recognize scanned objects by application and return a positive or negative message. If the server recognizes the object or places it will send attached items to the application.
- The server should save new registered objects to the database, and also send positive or negative feedback according to the registration process's success. If a place registered and registration process is successful, servers should also save GPS data to the database.
- The server should save attached files of all types to the database.

2.2 Nonfunctional Requirements

2.2.1 Performance

- The application should not consume the battery so the processes that require high computing power should be done on the server-side.
- The application should be able to display the attached media within 3 seconds.
- The user should be able to display and interact with media without any interruption so the content to be displayed should be downloaded and displayed.

2.2.2 Scalability

- The prototype of the application should be able to handle 10.000 users and 500 concurrent users.

2.2.3 Usability

- The interface of the application should be easy to adapt and use in 2 minutes for a new user.

2.2.4 Security

- The sensitive information of the user such as passwords should be sent to stored in the encrypted format.
- The ordering products process should be secure.

2.2.5 Availability

- The system should be available 24 hours 7 days and 365 days a year.

2.2.6 Modifiability

- Further implementations should be added without difficulty to the current system if needed. For that, the project should be modular.

2.2.7 Legal and Regulatory Requirements

- The terms of service and privacy should be accepted by the user before the usage of the application.

2.2.8 Accessibility

- The application should be accessible in at least 60% of Android and Apple devices.

2.3 Pseudo Requirements

- We plan to develop the application for both Android and iOS, therefore we will use a hybrid codebase.
- The server-side of the application, where the machine learning part will run, is expected to be written mostly in Python to make use of Python libraries.
- We will be using Github in order to keep our codes clean and compatible with each other so there is no desynchronization between coders.
- For animations and mini-game that will be displayed with AR, we will use Unity3D. In Unity, we will use C#.
- The application will use GPS along with user info in order to keep track of the AR media that is left. Therefore users need a good internet connection for GPS to properly work.
- For this app to function properly in outdoor areas, users need to have a decent cellular connection since public Wi-Fi may not be present everywhere.
- The objects should be static while getting recognized by the app. Moving objects such as cars could prove difficulties.
- We plan to use open-source libraries.
- For cloud computing, we will make use of Google Collab since we are familiar with Google Collab from our machine learning courses, and they offer free GPU usage which is enough for building the prototype of our app.
- The language of the application interface will be English. However, since the application content is mostly created by users, we will have no control over the content language.

3. Final Architecture and Design

Details

3.1 Firebase Database Design

As stated in the previous sections, we have used Firebase to implement our database operations. These operations include authentication operations, database

operations, and storage operations. Because Firebase is a NoSQL database, we have stored data as objects.

3.1.1 Firebase Authentication

We have used Firebase authentication to provide users with authentication features. In the application, we have implemented only registration and login with email functionalities in the application. While registering the user to the system, Firebase stores the email, and password of the users. Besides these, it creates a unique Id for each user and stores the created date for the profile. The system also encrypts the user password so that even the developers can't see the passwords of the users to provide security.

3.1.2 Firebase Database

3.1.2.1 Firebase Realtime Database

For some data which we need to listen to on the application side, we have used Firebase Realtime Database. The reason why we didn't use Firestore for that purpose is that, the firestore wasn't supporting listening for database in its c# libraries. We have maintained the chat data of the users in that section.

The hierarchy of Realtime Database is as follows:

```
chatList/  
  sampleUserId/  
    sampleChatIdForUser/  
      chatId : sampleChatId  
      senderName : sampleName  
      ...  
    ...  
  ...  
messages/  
  sampleChatId/  
    sampleMessageId/  
      sender : sampleName  
      text : sampleMessageText  
      ...  
    ...  
  ...
```

3.1.2.2 Firebase Firestore Database

Firestore database is mainly composed of two collections: “users” and “VispCollection”. “users” collection is for keeping user information such as userId, email, and username. The hierarchy of this collection is as follows:

```
users/  
  sampleUserId/  
    email : sampleEmail  
    username : sampleUsername
```

“VispCollection” exists to know which Visp is created by which user. The hierarchy of this collection is as follows.

```
VispCollection/  
  sampleVispId/  
    cretorID : sampleUserId
```

3.1.3 Firebase Storage

Our app needs a permanent and robust cloud storage system for all the media that is going to be displayed, and that data must be stored in a proper way that will allow us to use it with the user database. We used Firebase Storage for that purpose which is very beneficial to us in terms of storing, accessing and monitoring media files also known as Visps. We store these files by their associated image target on the Vuforia Cloud and their creators. Firebase Storage does not pose any performance issues in addition to issues stemming from client upload and download speeds which is only natural.

4. Development/Implementation

Details

4.1 Using Unity for Android Development

Unity is a game engine and provides a specialised editor for game development. Due to the AR related nature of our project, we chose to develop our Android application on Unity with the choice to use Vuforia AR engine. This choice came with its consequences since Android development on Unity is a much less discovered and experienced area for mobile application development. Especially since developing a social media application has its own feature requirements, certain tools, sources and documentation were very difficult to find as they were either old, not supported anymore or less experimented on which led to having trouble finding online references.

4.2 Using Vuforia on Unity

Vuforia documentation and official examples were the main sources for our learning process. Vuforia has its own Unity package for starting work with Unity. We used that package for installation of vuforia services to the Unity environment. We use Vuforia ARCamera in order to implement AR features like tracking recognized objects with the mobile phone camera and putting 3d objects to the camera scene. Also we wrote classes to connect Vuforia Cloud Services. Cloud Services are used for recognizing objects and images. We used Unity's 'WWW' module for sending new images to save to Vuforia cloud services. Some of the difficulties we faced was tracking many objects at the same time. Since we will display different media, objects etc. at the same time, we tracked some of the objects with trackable classes and some of the objects with our UI classes.

4.3 Firebase Integration

In the implementation of Firebase database, we have used the documentations provided by Google and the example solutions provided by various developers on

the Internet. Despite all, we experienced some difficulties while implementing Firebase in our Application because it is new and even some of its features were in Beta for Unity. Some problems even didn't have exact solutions but they had some alternative solutions to apply so we have used them.

In our project, we have implemented different Firebase functionalities in different scripts such as: DatabaseAPI, AuthManager, DBManager. The purpose for this was to provide modularity on different functionalities.

4.4 Maps Integration

Google Maps integration to any application requires a billing account to be submitted. Google provides \$200 credits for free. So far, we did not require any additional payment but in the future this integration may require for us to pay for this service.

5. Testing Details

Testing of Visp includes the verification of requirements and the tests of the mobile application. The requirements part is done during the analyzing and designing parts. We have always checked the current report according to the previous ones to avoid inconsistency. If a change is needed to be done, we have gathered in an online meeting and decided what to do about the topic.

The tests of the android application can be divided into four parts, unit testing, integration testing, regression testing, and alpha testing. Unit tests are done by the developer himself while developing the classes. Integration tests are done by at least 3 people from the development team at each integration phase in the implementation. Regression tests are done by the whole development team via screen sharing in discord. Alpha tests are done again by the whole team but individually at the end of the implementation phase.

6. Maintenance Plan and Details

Visp is an AR and social media app. AR and other technologies Visp use develops and changes every day. Also, social media apps need to keep pace with the new trends. Therefore Visp requires continued maintenance and changes as long as it lasts.

6.1 Cloud Maintenance

Visp uses cloud technologies for image recognition. Vuforia cloud services are an important part of the project since it is integrated with our AR systems in Unity. Vuforia can recognize objects, images, and places. It is still being developed with new updates, therefore the Visp team will follow these updates. Visp will update its own app, cloud communication systems, and cloud architecture according to updates of Vuforia systems. Also, if any changes are required because of newly implemented features, these changes need to be done in cloud systems.

Visp uses a free version of the Vuforia Cloud services. It is free of charge for 1000 recos (image recognition) and 1000 image targets per month. After the launch of the app, we need to purchase a monthly license, which is around 100 dollars every month, for more recos, image targets, and for removing the vuforia watermark on the App.

6.2 Server - Database Maintenance

Server maintenance is mostly done by Google Firebase. For the database maintenance, because we are using the free version of Firebase database, the user capacity of Visp is limited by 10 thousand people monthly, which is equal to the goal we set in the requirements part. For the usage part, Google allows 20k document writes and 50 k document reads per day. If we assume that the application reaches one thousand daily active users, the users should not exceed daily approximately 20 operations. That number would be too low for such a popular and widely used application so after the total number of users exceeds 5k, applying “Pay as you go”

pricing for the Firebase database part should be considered (<https://firebase.google.com/pricing>).

6.3 App Maintenance

Visp mobile apps need maintenance and updates regularly. These updates are required for adding new features to the app, improving security, and making performance optimizations. Also, an increasing number of users can require maintenance in order to keep pace with the growth. For now, it is only for android mobile apps. However, in the future, we will also publish for the Apple App Store.

6.4 Market Place Maintenance

Visp is going to be deployed to Google Play and Apple App Store. After deployment our team will be monitoring for install, daily active user statistics as well as comments and feedback. We are going to use Google Analytics which we integrated into our app through Firebase for this purpose as well. To comply with Google Play's policies of app deployment, a Terms of Service is going to be generated that is going to be presented to the user upon installation.

7. Other Project Elements

7.1 Consideration of Various Factors in Engineering Design

7.1.1 Public Health

In terms of Public Health, the only effect that Visp may have is motivating people to go near some locations on foot to see what other people have shared on specific objects or locations. That is, it may motivate them to walk.

7.1.2 Public Safety

In terms of Public Safety, the application may lead people to attach media files to some private locations or objects and it can make the people who own these

properties uncomfortable. That is, it may disturb some people because it gives people to share any content by relating to any object or place.

7.1.3 Economic Factors

In the next iterations of Visp, it is planned that the markets, restaurants, and enterprises can use the AR feature of the application to advertise their products in a better way so it could increase the sales and so the economy will be affected positively.

7.1.4 Global Factors

In terms of globalization, Visp may have an effect on tourism in a way that people may travel to other cities or countries to view and share things on objects or places.

7.1.5 Cultural Factors

In terms of culture, because we are planning to enter the market first in Turkey, we have made some search about how Turkish social media users differ from other countries and found that the Turkish users are using social applications mainly to follow others in order to learn and interact socially [2]. Because Visp is a social platform that people share and view the media shared by people, we have forecasted that Visp will be affected by cultural factors strongly.

7.1.6 Environmental Factors

In terms of environment, Visp will not have any effect on the environment.

Factors	Effect Level	Effect
Public Health	Low	People can walk to near places to see the shared content but it is a low possibility so the effect is considered as low.
Public Safety	Medium	The application has the possibility to

		lead people to invade the privacy of other people.
Economic Factors	High	With augmented reality technology, the advertisement methods will evolve and the economy will be affected positively.
Global Factors	High	Visp may have a major impact on people to travel to other countries.
Cultural Factors	Medium	The culture of the users affects the direction of Visp strongly.
Environmental Factors	None	Visp will not have any effect on the environment.

Table 1: Factors Summarized

7.2.Ethics and Professional Responsibilities

Firstly, we have to be sure that the user information is not accessible by anyone so the information should be stored encrypted in the database. Secondly, the users should not be able to share any violent, racist, and sexual content in the application.

7.3.Judgements and Impacts on Various Contexts

	Impact Level	Impact Description
Impact in Global Context	Medium	We are creating a social media application that does not have boundaries and can reach anywhere in the globe.

Impact in Economic Context	Medium	We are allowing companies to post advertisements on everyday objects.
Impact in Environmental Context	Low	Our project is only virtual, not physically using any real-life objects.
Impact in Societal Context	High	Any social media application with a growing user base will have an impact on the lives of its users.

7.4 Teamwork Details

7.4.1 Contributing and functioning effectively on the team

Alptekin Önder: Worked on AR part, classes for tracking and displaying different media and objects. Worked on the cloud image-recognition part. Wrote classes that work with Vuforia API and Vuforia cloud services. Debugged other features.

Enes Merdane: Made research on Firebase usage on Unity. Created the templates of Login, Registration, Messaging, ChatList scenes and implemented their connection with Firebase.

İsmail Şahal: Worked on the login and register pages, implemented the map scene, helped debugging and helped with general integration and other implementations.

Muhammed Berk Köse: Worked on vuforia and AR , Unity, UI design and implementation, some extra specific interface options such as taking a picture and cropping etc.

Muzaffer Yasin Köktürk: Worked on integrating Firebase Database to the project. Implemented file browsing and Firebase Storage functionalities. Helped implement Visp display UI.

7.4.2 Helping creating a collaborative and inclusive environment

We used our Discord server that is created for our project team to keep the meetings in and to record the decisions that we have taken in the meetings.

For the task management we used Trello to keep track of each task using a Kanban table which contains the sections “To be done”, “In Progress”, “To be reviewed”, “Done” and “History” sections.

7.4.3 Taking lead role and sharing leadership on the team

We have developed our project with the equal contribution of each team member. But the teammembers specialized themselves in different parts of the project so we can consider this natural discrimination as taking lead on different parts of the project.

7.4.4 Meeting objectives

7.4.4.1 Functional Objectives

7.4.4.1.1 System Functionalities

- Application is able to recognize objects and places that are previously registered into the system.
- The application displays related attachments of recognized objects.
- The application displays a list of attachments if there is more than one for a specified object.
- The application allows users to share videos and photos from the application to share on other social media platforms(Instagram, Whatsapp, etc).
- In registration mode, new objects can be introduced to the system.
- In registration mode, new places can be introduced to the system alongside their GPS information.
- The system gives positive or negative feedback about the registration process to users.

7.4.4.1.2 User Functionality Requirements

- Users can save the media files or moments from recognized objects to their phone storage.
- Users can attach media files(video, sound, image, gifs), texts, links to objects or places.
- Users can take screenshots of any moment in the scan mode.
- Users can swipe between attached files if there is more than one for an object.
- Users can add their friends to their friend list inside the application
- Users can send screenshots of Visps from our application to their friends from the friend list
- Users can see nearby memories if their GPS is active.
- Users can only use scan mode without authentication.
- Users can like other people's attachments to objects or places.
- Users can play and interact with an attached mini-game.

7.4.4.1.3 Server Functionality Requirements

- The server tries to recognize scanned objects by application and return a positive or negative message. If the server recognizes the object or places it will send attached items to the application.
- The server saves new registered objects to the database, and also sends positive or negative feedback according to the registration process's success. If a place registered and registration process is successful, servers will save also the GPS data to the database.
- The server saves attached files of all types to the database.

7.4.4.2 Non-Functional Objectives

7.4.4.2.1 Performance

- The application does not consume too much battery because the processes that require high computing power are done on the server-side.
- The application can display the attached media within 5 seconds if the user has no connection problems.

- The user can display and interact with media without any interruption so the content to be displayed should be downloaded and displayed.

7.4.4.2.2 Scalability

- Currently the system uses free versions of the technologies that we use. Therefore, our scalability is limited and could not be tested. We believe the paid versions would provide a better scalable application.

7.4.4.2.3 Usability

- The interface of the application was designed to be user-friendly. Our tests with our friends and family showed that the interface is easy to learn and not confusing. It took them 2-5 minutes to get comfortable with the user interface as it was predicted.

7.4.4.2.4 Security

- The sensitive information of the users are handled by the Firebase Authentication and Firestore Database. They are securely stored in the cloud and cannot be accessed without appropriate authentication tokens.
- The ordering products process should be secure.

7.4.4.2.5 Availability

- The application will be available for 24 hours and 7 days a week unless a third party service we use such as Vuforia and Google Firebase shuts down. Since they are proved to be reliable services our application can be considered as available at all times.

7.4.4.2.6 Modifiability

- Further implementations of new features should be added without difficulty to the current system if needed.
- For that purpose, we have implemented different main functionalities in different scripts. In that way, different team members were able to make developments on different functionalities at the same time.

7.4.4.2.7 Legal and Regulatory Requirements

- The terms of service and privacy should be accepted by the user before the usage of the application.

7.4.4.2.8 Accessibility

- Since our application's target API level is 29 (Android 10), it is going to be available for over 70% of all Android smartphones and tablets [2].

7.5 New Knowledge Acquired and Applied

7.5.1 Android Application Programming with Unity

Because Unity is not specifically designed for creating mobile applications but for creating computer and mobile games, we have struggled to find sample resources to create simple interfaces for our project. Therefore, it was harder than developing Android mobile applications with Java, Kotlin, or other cross-platform languages which are specifically designed for developing applications. However, with some youtube tutorials and example projects on Github, we have managed to solve Unity-related problems. In terms of deployment of the application to the Android platform, except for some minor Android SDK version problems, we have managed to create the APK of the application.

7.5.2 Adobe Photoshop CC

Adobe Photoshop CC is used as a design tool for general items such as icons and background. We also used photoshop for creating mockups, logo design.

7.5.3 Augmented Reality Application Development with Vuforia

Vuforia libraries and cloud services are widely used for AR applications and can be integrated with different technologies. We used Vuforia libraries for unity, in order to use AR functionalities in our application. These libraries also let us communicate to Vuforia cloud services which we use for image recognition. When we were learning these technologies our first source was documentation of Vuforia. Then for some tasks that we want to carry out, we read blogs and watched tutorials.

7.5.4 Firebase with Unity Applications

Firestore libraries in C sharp are in beta now so some of the functionalities were not available yet. Therefore, we had difficulties finding sufficient documentation resources and examples. Also, some of the features of Firestore such as some functionalities in Firestore weren't implemented in C sharp libraries yet. In such cases, we needed to find alternative solutions to these problems. But in the end, we have managed to solve all problems related to Firestore and Unity.

8. Conclusion and Future Work

We have managed to implement the main functionalities of the application such as authorizing the users, scanning objects for visps, adding visps, scanning area for visps, displaying visps, and chatting with people. In the future, when it is published in the market, a license agreement should be done with Vuforia because it will be published for commercial purposes. Also, according to our experiences on Firestore database, the free version of Firestore is not enough even for approximately 20 people to use actively so the billing account should be considered. Further tests should be conducted on the mobile to ensure the reliability and the robustness of the project. The iOS version should be deployed on a Macintosh computer to enter the Apple market as well.

9. Glossary

AR: Augmented Reality.

Vuforia: The AR engine we are using.

Visp: The combined concept of a scanned object and the attached files/name of our application.

Firestore: The database framework we have chosen.

Unity: The development environment we have chosen for our mobile development.

10. User Manual

10.1 Login Page

If you are not logged into the application already, this will be the first page you see when you open the application. If you haven't already registered to Visp, you can easily go to the registration page by clicking the Register button on the screen or if you have, you can fill the credentials on the screen and log in to the application by using the Login button.

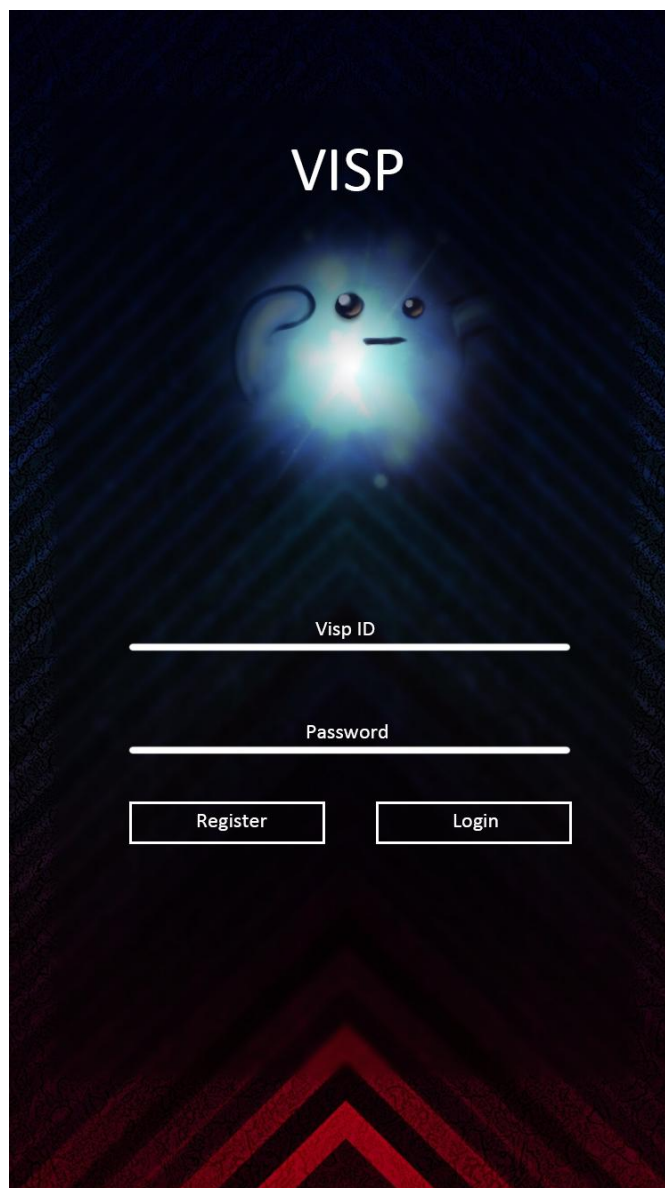
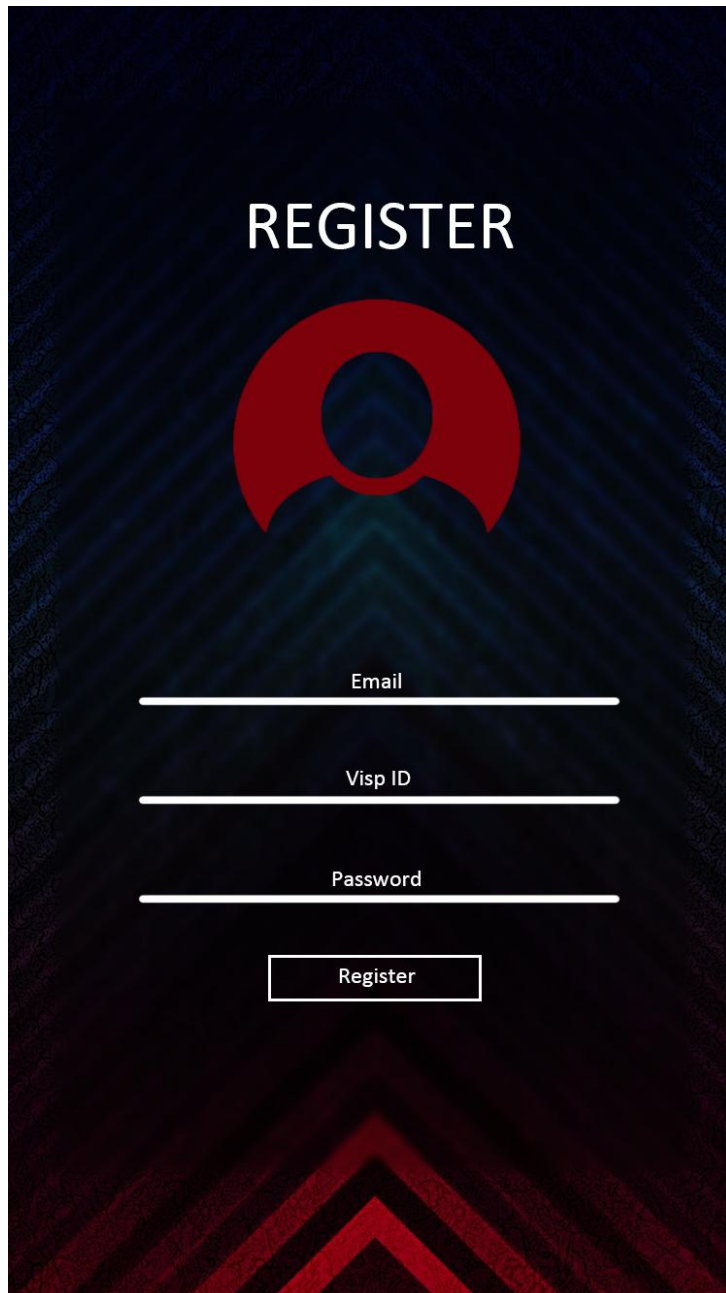


Figure 1 - Login Page

10.2 Registration Page

Using this page, you can register to Visp by filling the fields with the related information and clicking the button Register. Also, If you remember that you have an account and want to log in with that account, you can click the back button and navigate to the login page.

The image shows a registration page with a dark blue background and a red abstract logo at the top. The word "REGISTER" is written in white capital letters. Below the logo are three input fields labeled "Email", "Visp ID", and "Password". At the bottom is a "Register" button with a white border. The background features a subtle pattern of red and blue geometric shapes.

REGISTER

Email

Visp ID

Password

Register

Figure 2 - Registration Page

10.3 Main Page - Add Visp Screen

This is the first page you see after you log in to your account. If you are already logged in, you see this page when you open the application. This page opens your camera and provides functionalities related to the location or the object that your camera points to. At the bottom of the page, there is a wheel menu, which you can change its mode by sliding it to right or left. Currently, we are on Add Visp mode, which means that you can add visps to the objects and locations by specifying it on the toggle button above the Add Visp text. After you make a selection between Item and Area, you can add visp by clicking the button above them. Finally, at the top left, there is the main page menu button that opens the menu.

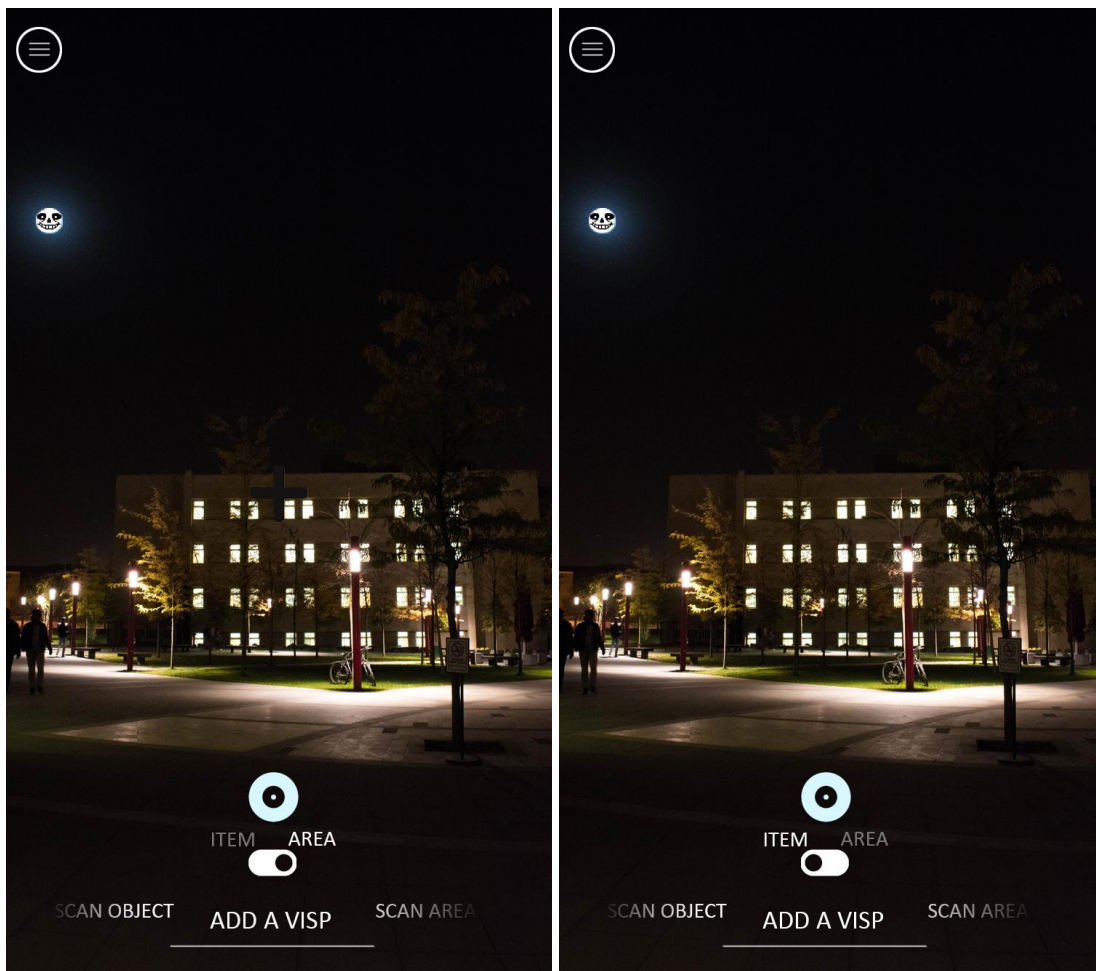


Figure 3 - Main Page Add Visp Screen Area & Item

10.4 Main Page - Scan Object Screen

This page opens when you slide the scroll button to the corresponding position below. The functionality of this page is scanning a specific object to see the visps attached to it.



Figure 4 - Main Page Scan Object Screen

10.5 Main Page - Scan Area Screen

This page opens when you slide the scroll button to the corresponding position below. The functionality of this page is scanning a specific area to see the visps attached to it. In the example below, you see a text “Summer Flowers <3 Bilkent” and a 3d flower image attached to it.

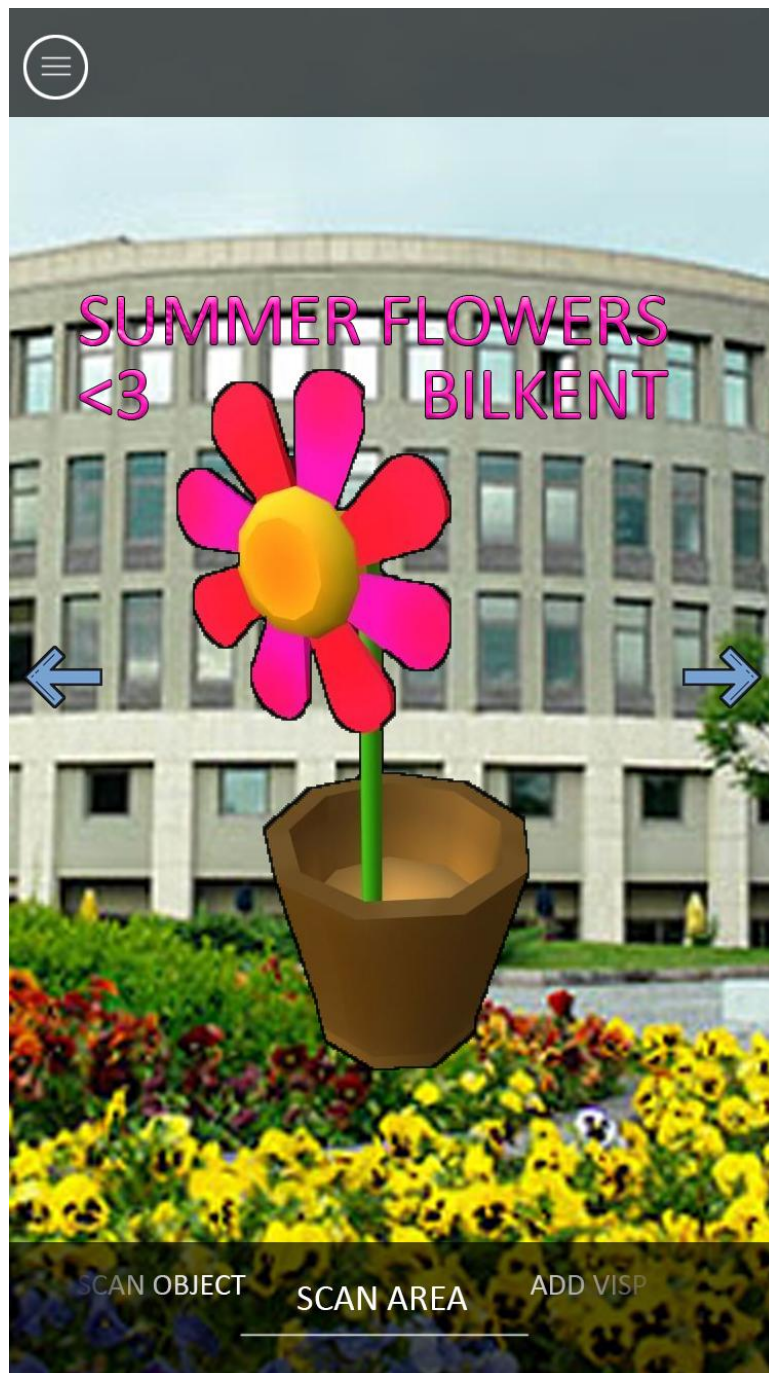


Figure 5 - Main Page Scan Area Screen

10.6 Main Page - Menu Screen

This page opens when you click the top left button on the Main Page screen. To close this user can click on the black space seen on the right or click the same button. On the menu screen, users can see their profile photo, username, and user hash id. This hash id is specific to the users and can be used to find other users. Below them, there are profile, messages, map, and settings buttons. You can use them to go to the corresponding screens.

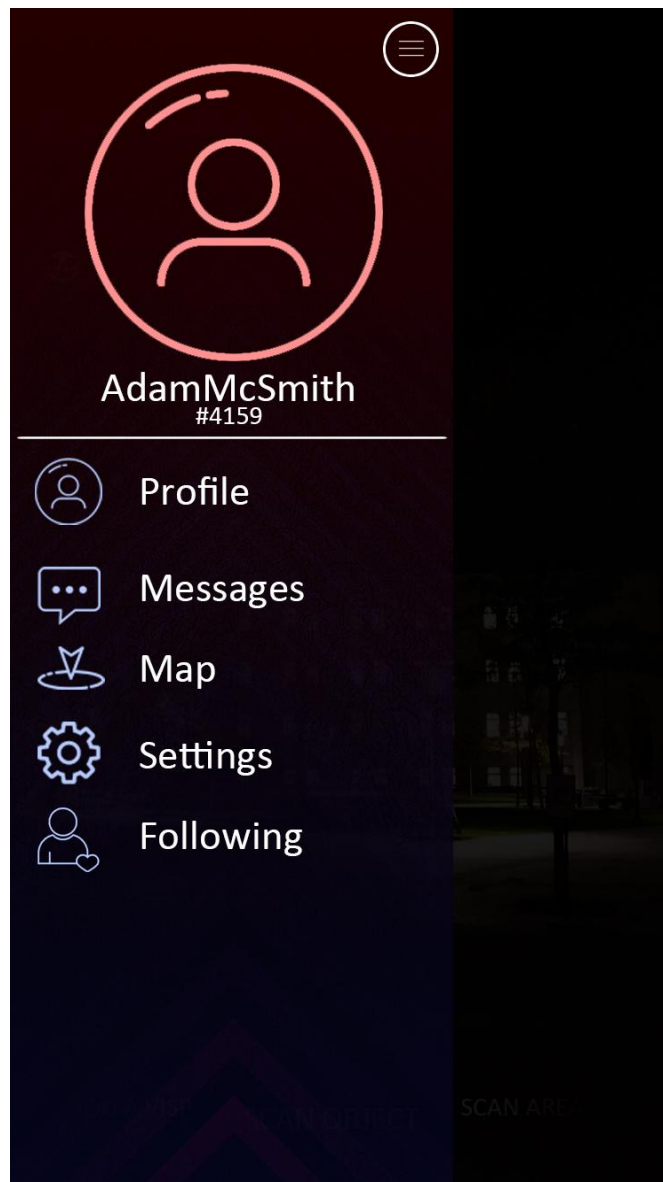


Figure 6 - Main Page Menu Screen

10.7 Profile Page

This page opens when you click the profile button on the menu screen. This page has two panels, which are info and friends. The friends panel shows the friends of the user and the info panel shows the information about the user. Other information on the screen is the same as it was on the menu screen. You can go back to the menu screen by clicking the left top button. Users can follow others to send messages to them.



Figure 7 - Profile Page

10.8 Messages Page

This page opens when you click the messages button on the main menu screen and select a chat. Here you can see your messages. you can go back by clicking the left top back button.

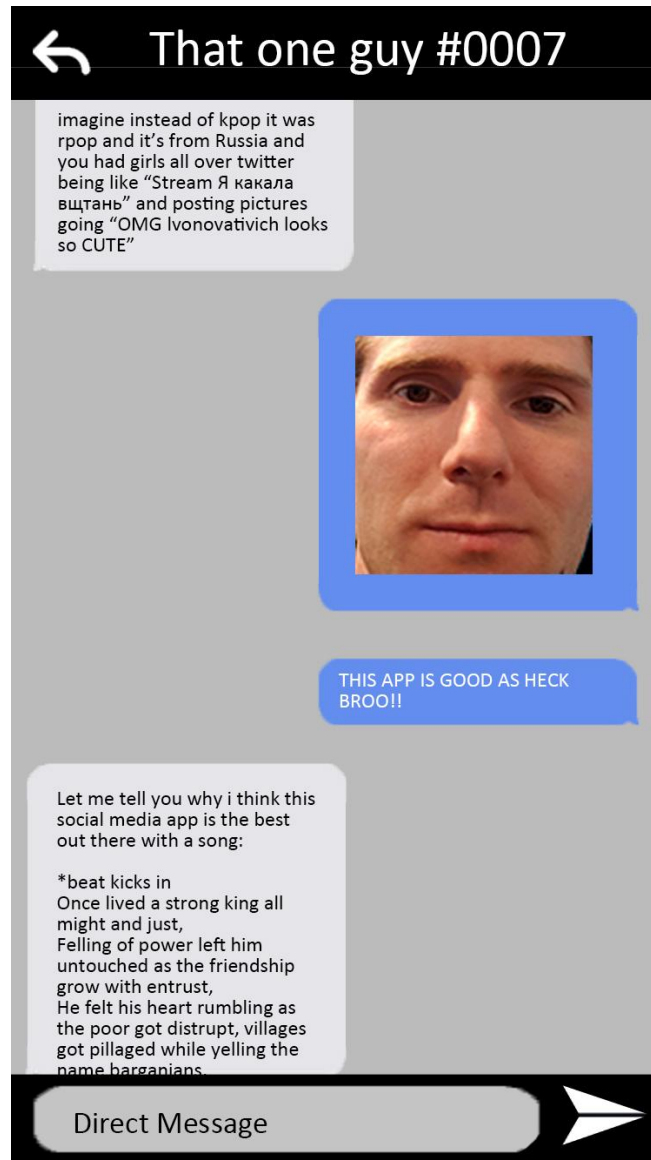


Figure 8 - Messaging with users screen

10.9 Following List Page

Here, the user can see whom they are following and can send messages to them by clicking on their name.

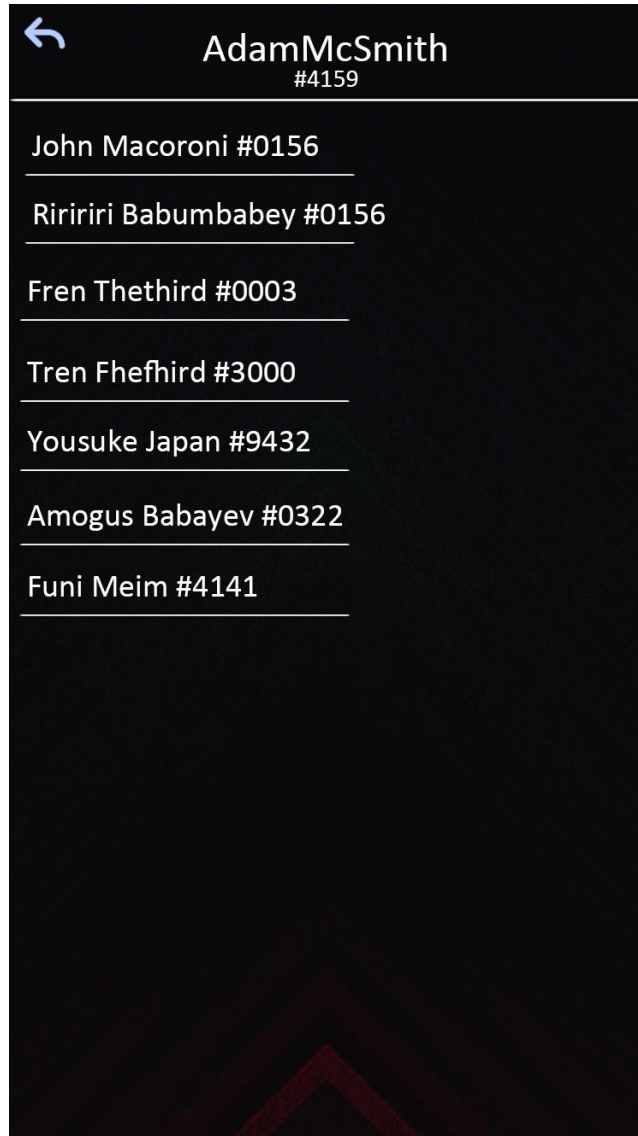


Figure 9 - Following screen

10.10 Map Page

This page opens when you click the map button on the main menu screen. Here you can see the visps around you. You can go back to the menu screen by clicking the left top back button.

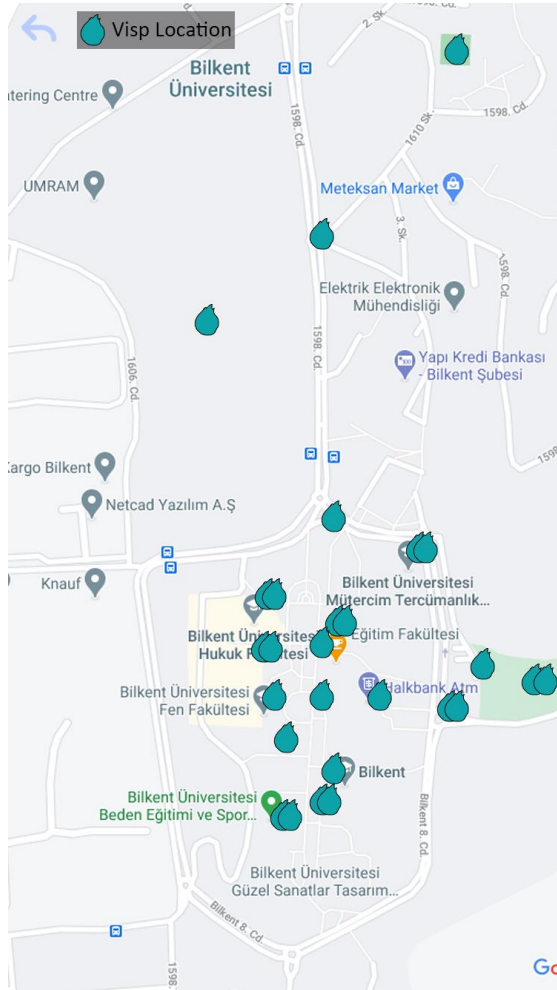


Figure 10 - Map Page

11. 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] "Mobile & Tablet Android Version Market Share Worldwide," StatCounter Global Stats. [Online]. Available: <https://gs.statcounter.com/android-version-market-share/mobile-tablet/worldwide>. [Accessed: 30-Apr-2021].