



Bilkent University

Department of Computer Engineering

CS492 Course Project

Visp

Low-Level Design Report

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

Supervisor: Uğur GÜDÜKBAY

February 8, 2020

Contents

1. Introduction	3
1.1. Object Design Trade-offs	3
1.1.1. Performance vs Cost	3
1.1.2. Monetization vs User Experience	3
1.1.3. Functionality vs Mobility	4
1.1.4. Functionality vs Usability	4
1.1.5. Development time vs Complexity	4
1.2. Interface Documentation Guidelines	4
1.3. Engineering Standards	5
1.4. Definitions, Acronyms, and Abbreviations	5
2. Packages	5
2.1. Server Subsystem	5
2.2. Client Subsystem	6
2.2.1. Entity Package	6
2.2.1.1. Media Package	7
2.2.2. Control Package	7
3. Class Interfaces	8
3.1. Client	8
3.1.1. Visper	8
3.1.2. Asset	9
3.1.3. Media Selector	9
3.1.4. AR Controller	10
3.1.5. AreaScanner	11
3.1.6. ObjScanner	11
3.1.7. Visp	12
3.1.7. VispApp	12
3.1.8. Mini Game	13
3.2. Server	14
3.2.1. TheConnector	14
3.2.2. Vuforia	14
3.2.3. Firebase	14
4. References	15

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. Object Design Trade-offs

1.1.1. Performance vs Cost

Visp uses Vuforia for real time recognition of objects. Vuforia's cloud provides us high performance but it is very costly. In Visp, we value the user experience a lot. Therefore, we choose the performance rather than the cost while designing and implementing the project. To the extent which the group decides, the cost will be accepted.

1.1.2. Monetization vs User Experience

Visp's monetization model depends on ads. However, in modern applications putting many ads ruin user experience. In order to overcome that we will not show users direct pop-ups or ads which cover all of the phone screen, instead all the ads that will be shown are going to be put as Visps to public places.

1.1.3. Functionality vs Mobility

Visp's content sharing system depends on specific locations and objects as per its nature. This, however, may yield a hindrance for the mobility of the platform which is a crucial aspect of social media platforms. For the development of Visp we prioritize the location/object oriented content sharing schema over its ability to be used on the go since we believe that this functionality is what makes Visp unique and appealing to use. Nevertheless, our aim is to enhance Visp's usable area as much as we can by developing an efficient and neat augmented reality interface.

1.1.4. Functionality vs Usability

Our aim for Visp is to provide users with a good user experience of the application so that the users can suggest the application to their friends, workmates, etc. and the number of users of Visp can increase organically. We believe organic growth is the best one for application because otherwise we may not be able to keep the users that we have gained. If we miss the chance of first impression, in the second chance it may be harder. Therefore, we don't want users to drown in lots of functionalities and not being able to use the application easily.

1.1.5. Development time vs Complexity

Because we are not experienced on Augmented Reality apps, developing and debugging Visp may take a very long time. Also in 4 months with our other courses in Bilkent, it is not easy to get so experienced in Augmented Reality apps that we can build a complex application. Therefore, for this tradeoff we will be focusing more on the development time rather than the complexity. We should be sure that the project is completed in time and its main functionalities work.

1.2. Interface Documentation Guidelines

The guideline of interface documentation is given below.

Class	<ClassName>
Description	<ClassDescription>
Attributes	
<AttributeName>	<AttributeDescription>
Methods	
<MethodName>	<MethodDescription>

Table 1: Interface Documentation Table Example

1.3. Engineering Standards

In our design phase, we are planning to follow IEEE standards while writing the reports [2]. While creating the diagrams we will use UML standards[3] and finally for the citation and report format we will use IEEE report standards [4].

1.4. Definitions, Acronyms, and Abbreviations

Visp : Attachments that are attached on any place or object.

Attachment : Any file like image, video, text, gif, animation, etc. that you can attach to objects in Visp app. It can be used interchangeably with Visp.

Visper : People who use visp and leave visps to objects or places

Application : Visp mobile application.

2. Packages

We analyze our application in 2 main subsystem components.

2.1. Server Subsystem

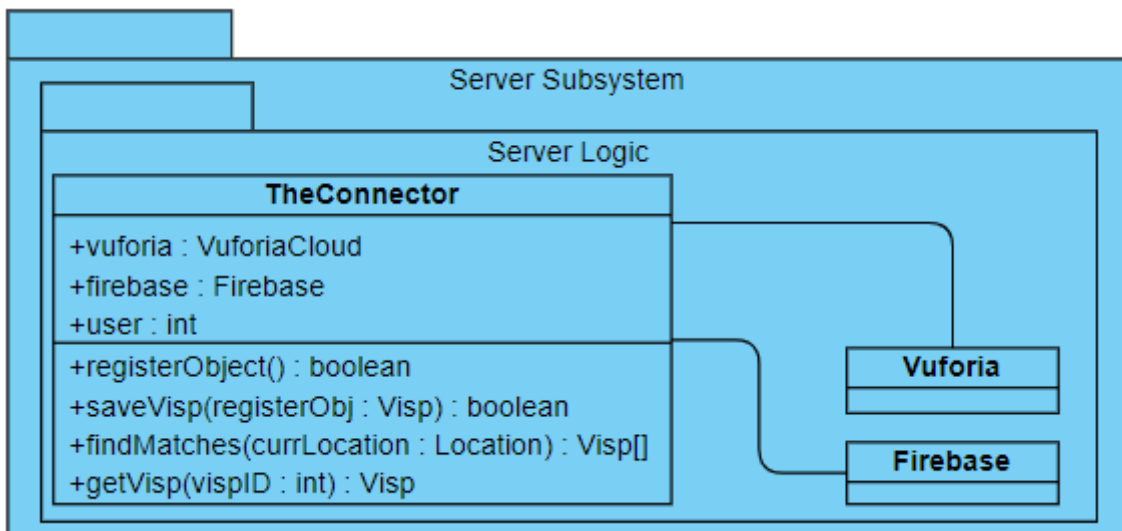


Figure 1: Server Subsystem

First of our subsystems is the Server Subsystem which consists of the part of our application that runs on the server side. TheConnector class is the class that communicates with the client side and handles the requests. Vuforia class is the class which is connected to the vuforia cloud and Firebase is the database system that we will use for storing information about users and assets.

2.2. Client Subsystem

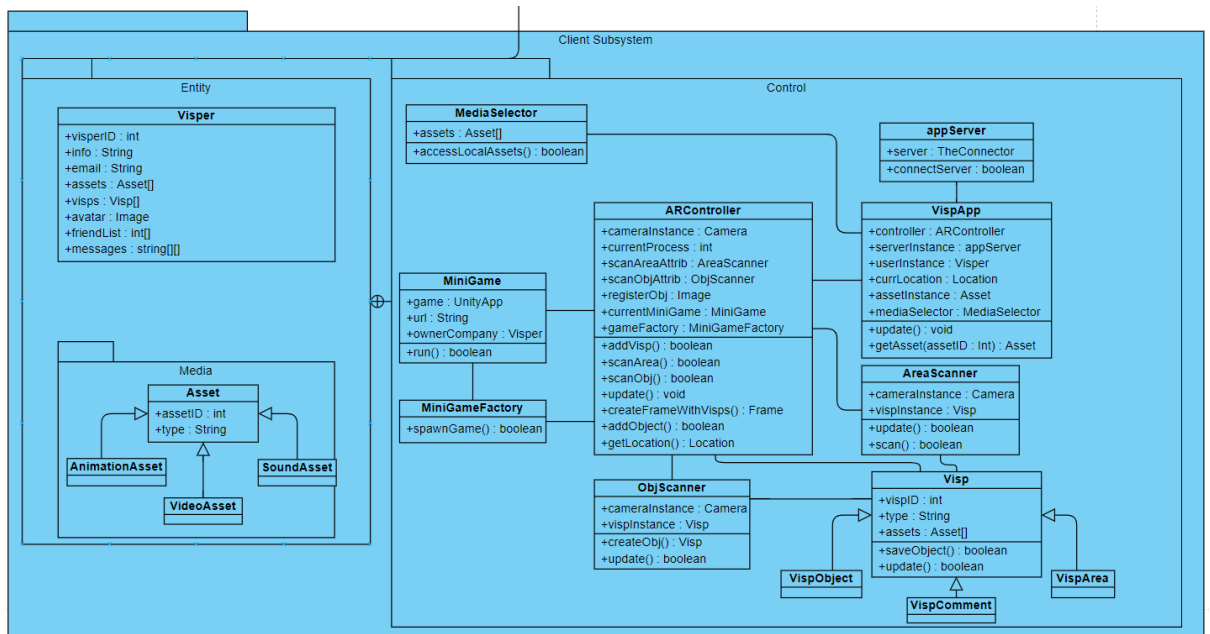


Figure 2: Overview of the Client Subsystem

Client Subsystem consists of the parts of our application that runs/stored on the user's phone.

2.2.1. Entity Package

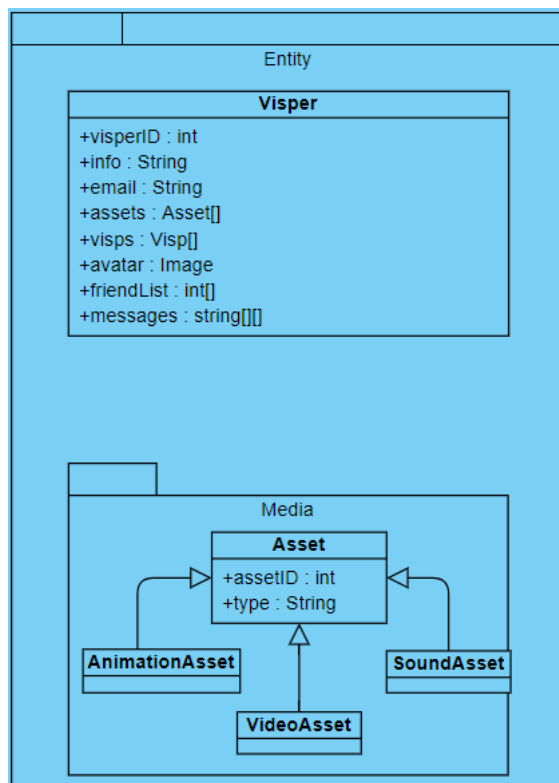


Figure 3: Entity Package

This package consists of the classes that are communicated across the client side and the server side. Mainly this package addresses the user related media and information.

2.2.1.1. Media Package

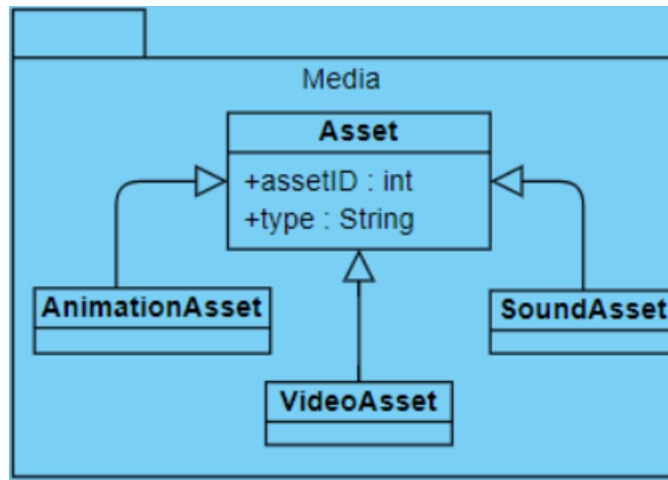


Figure 4: Media Package

Media package consists of the classes that represent the attachments that the user can add to a created Visp.

2.2.2. Control Package

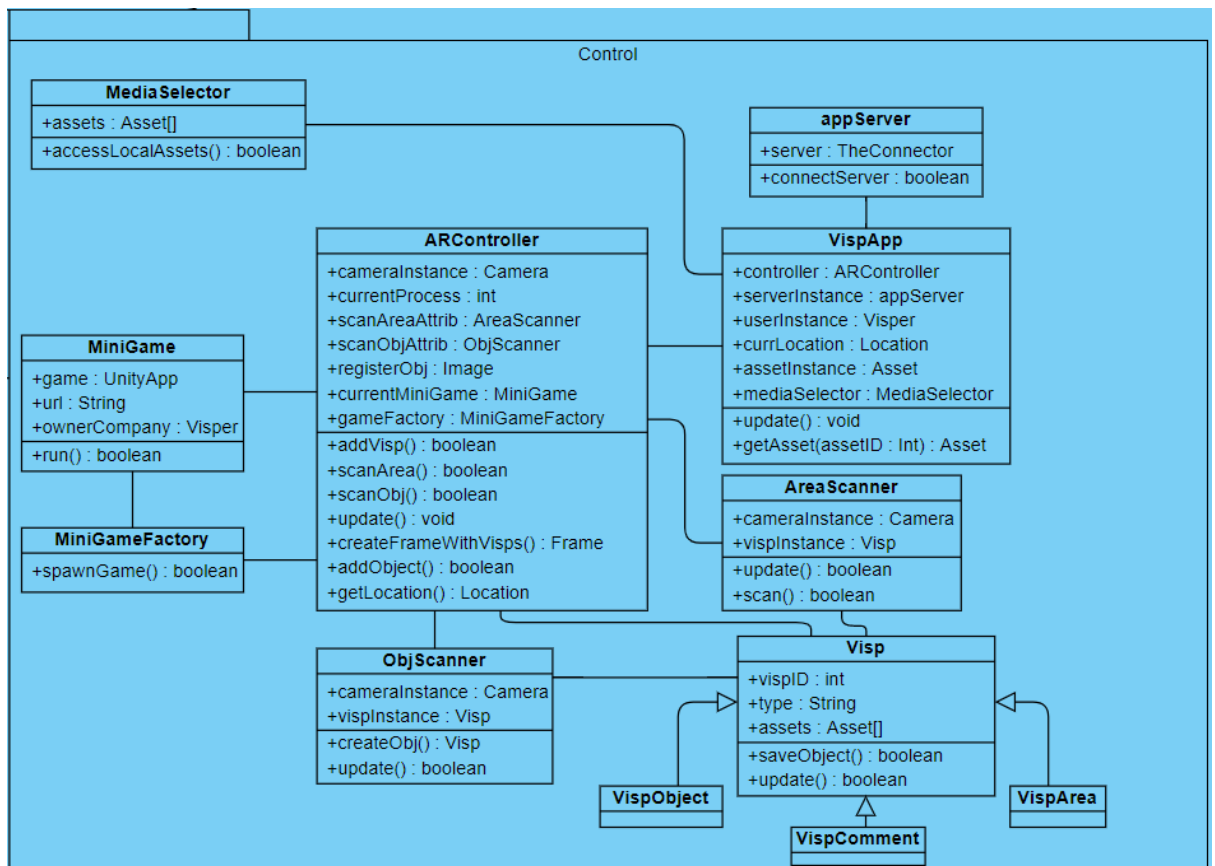


Figure 5: Control Package

Control package consists of the classes that manage the logic of our app that is running on the client side. The vispApp is connected to an ARController class and appServer class. The ARController class connects every element/class of the application together to run the system, and appServer class connects the application with the database in order to get a hold of the information stored in firebase and vuforia.

3. Class Interfaces

3.1 Client

3.1.1. Visper

Class	Visper
Description	Users of the Visp. They are called Visper
Attributes	
userName	String - Name of the user
name	String - Name of the Visper.
email	String - Email of the Visper.
visperID	String - A unique ID assigned to a user.
info	String - Description of the user
password	String - Password of Visper. It is crypted (retarded).
avatar	image - Image of the user
friendList	String[] - List of Vispers friends.
message	String[][] - List of messages from list of friends.
Methods	
change_username	Method that changes the username

change_password	Method that changes the password
change_email	Method that changes the email
change_info	Method that changes the info

3.1.2. Asset

Class	Asset
Description	Visp media that is attached to objects.
Attributes	
assetID	int - Unique ID which is assigned to assets.
type	String - Defines the type of the asset.

3.1.3. Media Selector

Class	Media Selector
Description	Helps with selecting media items from the user's phone.
Attributes	
assets	Assets[] - This is the list of assets attached to a visp.
Methods	
accessLocalAssets	Keeps the list of items that were selected.

3.1.4. AR Controller

Class	AR_Controller
Description	Controls the AR camera and AR features of the App.
Attributes	
cameraInstance	Camera - AR camera instance.
currentProcess	int - number of items AR camera shows.
scanAreaAttrib	AreaScanner- scanner areas.
scanObjAttrib	ObjScanner- scanner for objects.
registerObj	Image - image of new object or area to register
currentMiniGame	MiniGame - mini game object that can be played in AR view.
gameFactory	MiniGameFactory - mini game factory object retrieves minigames from server and loads to App.
Methods	
addVisp	adding a new visp to an area or to an object
scanArea	scans an area
scanObj	scans an object
update	updates the scene
createFrameWithVisps	add Visps to frame
addObject	adding a new object to the system.
getLocation	getLocation information from the GPS of the phone. Used when adding area to the system.

3.1.5. AreaScanner

Class	AreaScanner
Description	Class that scans and recognizes the location.
Attributes	
cameraInstance	Camera - camera to scan the area
vispInstance	Visp - visp(s) that is going to be displayed in the area
Methods	
update	Updates the liveview from the camera
scan	scans the area

3.1.6. ObjScanner

Class	ObjScanner
Description	Class that is responsible for scanning an object to be used in a Visp.
Attributes	
cameraInstance	Camera - camera to scan the area
vispInstance	Visp - visp(s) that is going to be displayed in the area
Methods	
createObj	Creates and returns Visp object.
update	updates screen according to Visps.

3.1.7. Visp

Class	Visp
Description	Main elements of our application
Attributes	
vispID	int - A unique id that is attached to an Visp object
type	It defines the type of the Visp.
assets	assets - set of assets that Visp have.
Methods	
saveObject	Saves the attached media to the specified Visp.
update	updates the Visp.

3.1.7. VispApp

Class	VispApp
Description	Class controls Visp Application
Attributes	
controller	ArController - Arcontroller object instance for App.
serverInstance	appServer - Object to communicate to server.
userInstance	Visper - User of the App.
currLocation	Location - Current location of the user.
assetInstance	Keeps the currently open asset instance.
mediaSelector	MediaSelector - object that helps to select media files from the mobile phone.

Methods	
update	Updates the apps current state according to inputs.
getAsset	Returns the Asset object with the given id.

3.1.8. Mini Game

Class	Mini Game
Description	Class for running unity games inside Visp application on AR Camera
Attributes	
game	UnityApp - Unity application to run inside Visp application
url	String - url to download game data to Visp application.
ownerCompany	Visper - Minigames are owned by companies
Methods	
run	Runs the game in Visp application

3.2. Server

3.2.1. TheConnector

Class	TheConnector
Description	Controls the visp search and match functions according to the request that were sent by the client
Attributes	
vuforia	VuforiaCloud - Vuforia cloud instance that does the image recognition and match.
firebase	Firebase - Database instance that controls queries
user	int - ID of the current user (Visper)
Methods	
registerObject	Creates and returns Visp object.
saveVisp	Saves the specified Visp data.
findMatches	Find the matches of the Visp data.
getVisp	Returns the Visp data.

3.2.2. Vuforia

Class	Vuforia
Description	This class represents vuforia libraries that we are going to use

3.2.3. Firebase

Class	Firebase
Description	This class represents firebase database functionalities that we are going to use

4. 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] Standards.ieee.org. 2021. IEEE SA - The IEEE Standards Association - Home. [online] Available at: <<https://standards.ieee.org/>> [Accessed 3 February 2021].

[3] Uml.org. 2021. Welcome To UML Web Site!. [online] Available at: <<https://www.uml.org/>> [Accessed 3 February 2021].

[4] Standards.ieee.org. 2021. IEEE SA - The IEEE Standards Association - Home. [online] Available at: <<https://standards.ieee.org/>> [Accessed 3 February 2021].