Bilkent University

Department of Computer Engineering

Senior Design Project

Project short-name: Augma

High Level Design Report

Ahmet Burak Şahin, Can Bayraktar, Çağdaş Han Yılmaz, Fırat Sivrikaya, Utku Oymak

Supervisor: Ercüment Çiçek
Jury Members: Bülent Özgüç and İbrahim Körpeoğlu

Progress Report

December 22, 2017

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.

Contents

High Level Design Report

Project short-name: Augma

1. Introduction

New tools and ideas emerge in the profession of computer science with advancements in technology. Some of these tools or ideas start to grow and become more popular over time. Augmented Reality [1] [2] [3] is one of these emerging technologies which makes it possible to look at the world from a new perspective. Even though it is a relatively new technology, it holds many great possibilities like helping doctors in a surgery, a visual navigation that doesn't need for the driver to look away from the road, just to name a few. Unfortunately, being new also comes with many drawbacks like most users not willing to change what they are using or lack of tool kits and native platforms. Most AR applications today are designed to work on mobile devices [4] [5] like tablets or phones but true potential of this technology can only be unlocked with devices like Google Glass. We expect to see much more advanced AR applications [6] [7] [8] as native AR platforms [9] like Google Glass starts to become affordable by the general public and this can only happen if developers keep creating new and interesting Augmented Reality applications, further increasing the value of the technology and attracting more investors to the area.

Social media applications are the most used day-to-day mobile applications in the market. This is why we saw a need and a potential for a new kind of social media application in an augmented space. Augma creates a world where people can interact with each other in an exciting and creative way. Users will be able to post location based notes on anywhere in the world. Other people who are near one of these notes will be able see it using Augma like a scope, looking into another dimension from their phone's camera. With Augma, we aim to achieve a deeper level of empathy between the users than any other social media application today by putting the readers into the exact same environment where the post was written.

In this report, we aim to provide an overview of the architecture and design of the system we will develop. First of all, the existing systems that are similar to ours, their qualities, and the missing features of the available systems are described. Then the details of our system design are listed. Subsystem decomposition, architectural plans of subsystems, and hardware/software mapping of these components are illustrated. Design decisions such as persistent data management, access control and security, boundary conditions are reported in detail. Finally, the functions of subsystem services and their interactions are presented.

1.1. Purpose of the System

Augma aims to provide another layer to the world we are living in for people who would like to take notes using Augmented Reality technology. As it would be fun to drop notes and possibly share these notes to other people using the Circle system you would actually check what notes you or other people post to Augma as oppose to forgetting about your notes.

Augma also aims to be an advertisement platform for companies, concerts or other notable events. This way the real world would not suffer from the environmental pollution as AugmaAds will be digital. Also, use of historical buildings or other places that are restricted in real world to put up advertisements will be usable which will make the layer of Augma not too complicated with all ads pile up on the same place.

1.2. Design Goals

1.2.1. Usability

- The system should have a user-friendly UI.
- The system should display in-app assets in an understandable fashion. Users should be able to distinguish notes, pictures and other in-app assets from one another.
- Users should be able to use Augma without the real time AR function if their phones aren't compatible with AR.
- The system should obey the Usability standards (ISO 9241-11) as much as possible.

1.2.2. Extensibility

- The system should be easily extended to support new functionalities.

1.2.3. Reliability

- The system should persist uploaded user notes. It should be able to keep user notes without corrupting any of it.
- The system should load and display the same note to multiple clients consistently.
- The system should not drop frames while refreshing UI.

1.2.4. Efficiency

- Data exchange between the client and the database for the notes should be fast enough to allow users to walk around while using AR without major buffering issues.
- Application boot up time period should be unnoticeably slim.

1.3. Definitions, acronyms, and abbreviations

AugmaAds: Advertisements that can appear in Augma.

REST: Representational State Transfer

AR: Augmented Reality

UI: User Interface

API: Application Programming Interface

HTTP: Hypertext Transfer Protocol

TCP: Transmission Control Protocol

Server: The part of the system responsible from logical operations, scheduling, and data management

Client: The part of the system the users interact with.

S3: Simple Storage Sevice by Amazon

EC2: Elastic Compute Cloud by Amazon

MsSQL: Microsoft SQL Server

AWS: Amazon Web Service

1.4. Overview

We are living in an era where conveying information is everything. The way you convey that information is just as important as the information itself. We felt like writing 120 characters or even posting a picture with a lengthy paragraph under it isn't enough. We wanted to add another dimension to our conversations. This is where Augma was born. Augma adds a new layer to our reality and lets our posts come to life.

Augma is a mobile-app which lets user leave location based notes that come to life with the power of Augmented Reality technology. Users can leave pretty much anything they want ranging from a plain text to pictures. After users have left their note to a location, other people who look through their phone cameras will be able to see their creation. These notes will be time-limited and people who see these notes will be able to rate these notes by upvoting or downvoting.

Users will have the choice to make their notes visible to all other people or just certain groups by using Augma's Circle system. Users can create their own small scale circles to leave private messages to their friends and family. Users will have the option to leave notes only they can see for personal use such as a reminder for an upcoming project. With the usage of the Circle system, users will be able to personalize their window to this new layer of reality. Augma will provide the option to see the notes that are only from the certain circles users have specified to filter the notes in the augmented world.

Augma will also feature a heat map in which users can see where the places with lots of notes are. When users get to a “Hot Zone” like this, they will receive a notification on their phones prompting them to have a look around even when the app is running in the background. If they have the map open and got close enough to a note, they will be able to see an indication and a small preview of the note. With this function they will be able to determine if there are notes they want look at without actually having to open their camera. If users have low battery, their phones aren’t powerful enough or maybe they don’t like having to turn around with their phone to actually see the Augma’s world. They can click on these previews to see the notes like a picture taken from the perspective of their creators.

Another one of Augma’s big features will be the Augma ads. Companies will be able to leave creative ads in the Augma’s world for a much cheaper price than in real life. Think about seeing a giant yellow M on the moon when you look through your camera at night if you are near a McDonald’s. This will allow for much more interesting ads and this way the big companies won’t have to pollute the scenery with their giant ads.

2. Current Software Architecture

AR is a fairly new technology and because of this, what we are trying to do hasn’t been tested thoroughly. There were only 2 similar applications on the market that we could find.

- **WallaMe** is an augmented reality location based message app that lets you “Hide messages in the real world using augmented reality.” [10] [11]
    - WallaMe is multi-platform. It is available in both IOS and in Android.
    - It lets you leave location based augmented reality messages on walls for certain people you chose. (You can also leave public messages but this is not WallaMe’s main purpose)
    - It supports writing, drawing and painting tools, images, emojis and stickers.
    - WallaMe also includes a like and comment system.
    - It uses image recognition to display the AR messages.
- **Mirage** is an app that lets you create and leave interactive compositions on top of the real world with augmented reality. [12] [13]
    - Users can create mirages by placing gifs, objects, text, hashtags, photos, and drawings.
    - It only lets you place your mirages only on to walls and some flat surfaces.
    - Users can upvote other people’s mirages and share any mirage you want with your friends.

- It uses image recognition to display the AR messages.

Both of these applications are very well made but they have one common weakness and that is the image recognition technology in order to display AR images. This creates major problem for usability since users need to select a mirage/message first then the application displays it alone. Also users need to find that exact background image in order to make the message appear and they can only display one image at a time. Our system aims to improve these shortcomings by not using image recognition, instead by only using location data to display the images. This will allow for better and longer battery life during usage, ability to display more than one note at a time and a much better user experience.

3. Proposed Software Architecture

3.1. Overview

In the subsystem decomposition, the subsystem structure of our system is described thoroughly. Partitions of the system are shown altogether with classes in each layer are shown. Then, system's mapping of hardware/software will be provided which presents different parts of our system and how it works using different hardware components. Persistent data management is also present that explains how we store our data. Access control and security defines access and exit boundaries of our systems. In global software control, how our server acts as main controller is discussed as well as the general flow in the system. In the end, boundary conditions such as initialization, termination and failure conditions.

3.2. Subsystem Decomposition

Augma follows a Client-Server architectural style to effectively respond/process concurrent user requests. On the client side, our phone application handles most of the processing which is mainly AR. Rendering AR elements, data transmission and GPS localization are the primary operations client-side will be dealing with. On the server side, the system will be managing the database which should be efficient and effective as much as possible. Main goal is to achieve the highest performance while sustaining the lowest response time for each user request.

Our system falls into 3-Tier system architecture which are: Presentation Tier, Logic Tier and Data Tier. The Presentation Tier, which is the topmost tier, contains the visual components of the system and it can be found mostly on our client-side application. This tier is responsible for managing the interactions with the user such as displaying the environment with AR elements via camera, showing the map of near vicinity, etc. The Logic Tier contains the fundamental operations behind the Augma. It mostly resides on

the client-side and includes operations such as AR rendering, GPS localization. The Data Tier is responsible of database management and primarily resides on the server side.

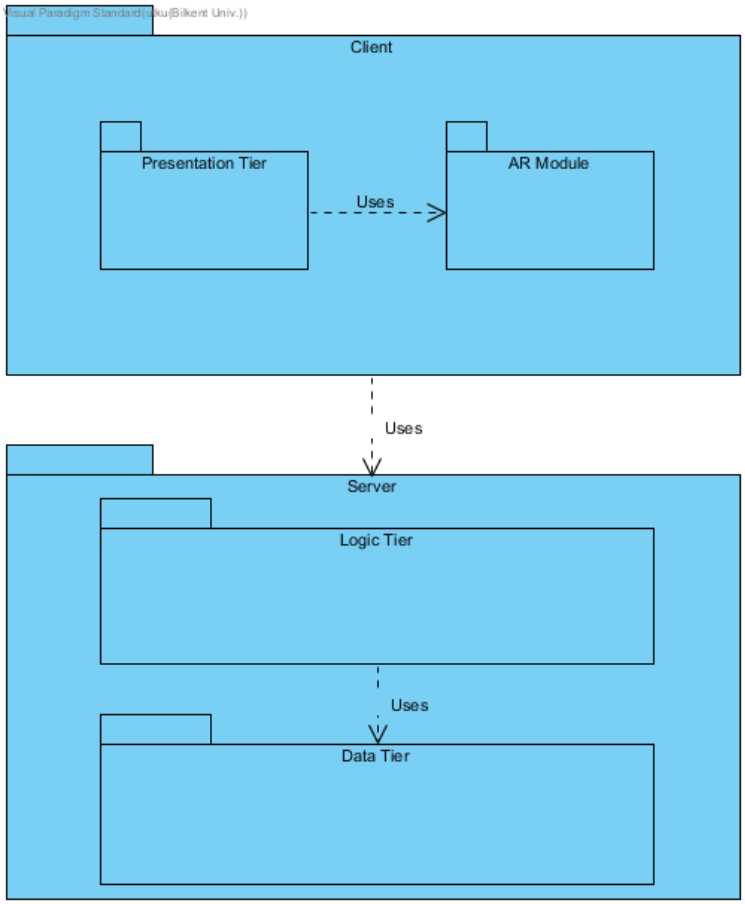

Figure 1: Subsystem Decomposition

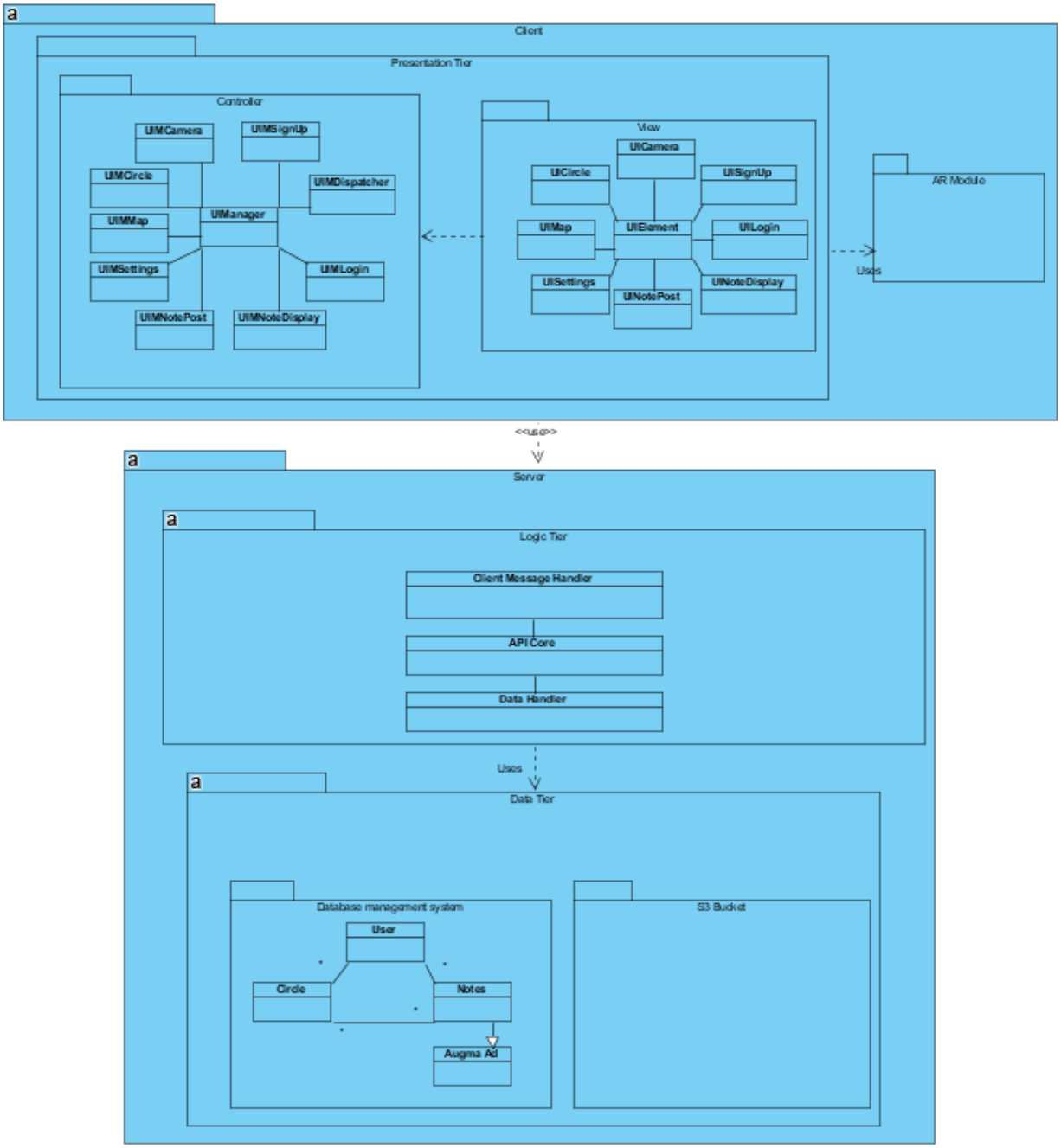

Figure 2: Subsystem Decomposition Detailed View

3.3. Hardware/Software Mapping

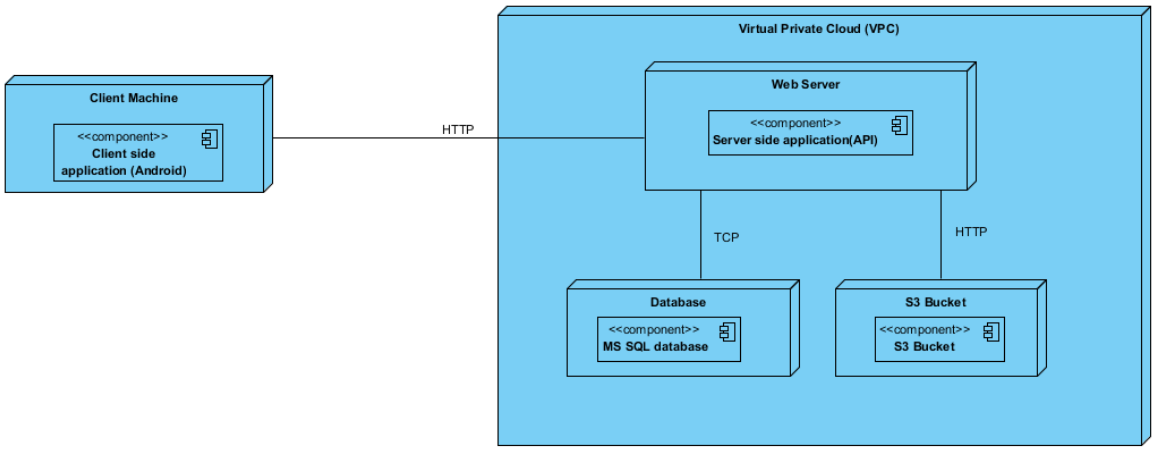

Figure 3: Component Diagram

Client side of Augma will run in a smartphone which uses Android as its operating system. The system will use screen, camera, and location services in client side. Augma will require to use significant amount of processing power as it will perform the AR-related operations on the client side. User and application data will be stored in the cloud, so they do not require additional phone memory. The app requires internet connection in order to perform data transfers between servers. HTTP [14] will be used to interact with the server.

Server side will rely on RESTful Web API [15] to serve requests coming from client. It consists of two components; Database Management System and S3 Bucket. Database part will use MsSQL [16]. S3 Bucket will be used for storing notes and other images in the system [17].

3.4. Persistent Data Management

Most of Augma's social media related features will require a data storage solution. The data can be separated into two different groups, one is user information data and other one is post image data. User info will include things like credentials, joined circle information, friends list and metadata for the posts. Post image data, as the name suggests, is the actual posts made by the users. These two types of data have different usage profiles therefore need two different data storage solutions. User info will be accessed and changed regularly as users can change their circles, create new circles of just change their credentials. Image data on the other hand will need to be accessed regularly but they will not be changed. Once a post is created and uploaded, it cannot be changed. There can be new images for a new post or deletion of an image for an older post but actual image cannot be altered.

We decided to use a relational database for the user info data since it is the most conventional and easy to handle solution for this kind of data. We will be using a Microsoft SQL Server hosted in AWS cloud infrastructure. Since our global control API will be written in C# this will make it a lot easier to access and manage the database.

Image data for the posts will be stored in a S3 bucket, again hosted in AWS. Reason for this decision is the drawbacks and strain caused on the database resulting from storing a high quality image file in a relational database table. Amazon's S3 technology is the best solution to our problem with its easy integration with other AWS hosted services and cheaper prices compared to database storage solution.

This separation causes the necessary data for showing a post on the screen to split into two different locations. Our control API's logic for serving a post to the client will be like the following:

Client will ask for a post from the API, API will go to the database and find the meta data for the post with the location data for the image, then it will get the image from the S3 bucket using the location data, after that the API will put this information in an easy to read form and serve it to the client. This design will allow Augma to keep its data safe, persistent and make it fast to access.

3.5. Access Control and Security

Users of Augma can perform various actions in the application that require access control. They can post notes, manage their custom circles, update their profiles, view notes posted in the circles that they are a member of, accept invitations. User-specific datas are separated from each other so that other users cannot make modifications on them. Users are not allowed to access and manipulate data that they do not own or they do not have permission to do so. For instance, a user is not able to read a note posted in a circle that he is not a member of. The notes posted in a custom circle are isolated from outsiders.

Security is one of the key concerns of Augma. All users register to the system with a username/email and a password. There is a character limit specified for the passwords so that the users are required to choose a password which has at least 8 characters. The passwords will be stored in hashed format by using a strong hash algorithm in order to increase data security.

Augma respects personal data and ensures its security. Location datas of the users are only accessed by the system and Google Maps API, and they not shared with third

parties. Without permission of the users, their personal data is not shared with the system.

3.6. Global Software Control

Augma will have a centralized event driven control system. Most of the controls will be done in the server side by the global control API that we will write. API will be in an EC2 [18] instance in the aws. This API will mainly work as a buffer between the clients and the data storage parts of Augma [19]. When the clients need anything from the database or the S3 bucket storage or when they need to upload something to the storage they will do so with the API. This way clients will not be in direct contact with the data storage services which will increase both security and reliability.

API will include functions to check credentials, change specific user info data, put new posts into the cloud, get existing posts and other information from storage and serve it in an easy to read and easy to handle way to the clients. Formatting the data in the cloud and serving it to the clients afterwards lightens the load on the client which will subsequently increase battery life. Also AWS's dynamic upscaling features will allow the API to handle user spikes in peak hours of use.

3.7. Boundary Conditions

Initialization

The application requires an Android device to run. The user can interact use the functionalities of the application after creating an account and logging in with that account. Trying to login with false credentials will cause the application to display an error message. Additionally, the application retrieves real-time data using an internet connection and location services, therefore an active internet connection and a permission to location services are needed.

Termination

The user can logout from the application by clicking the Logout button or by clearing the application data. If the application is not terminated, it will continue to run on the background using internet connection and location services. Terminating the program does not cause the user to log out.

Failure

The application can cause a failure if there is no internet connection. Additionally termination of the application while the application is performing an action, might cause failures.

4. Subsystem Services

This part of the report analyzes the subsystems of our system and describes the services they provide in detail.

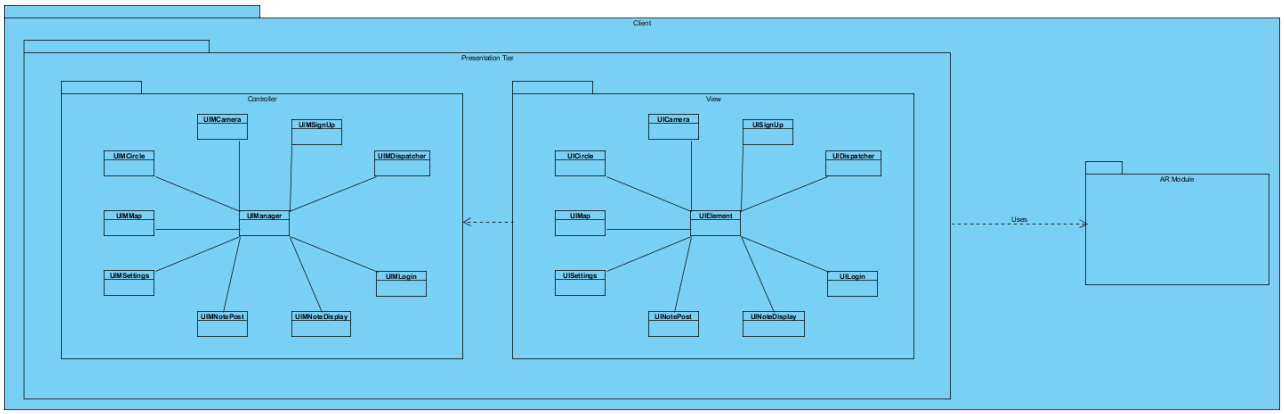

Figure 4: Detailed View of Client Subsystem

4.1. Client

4.1.1. Controller Subsystem

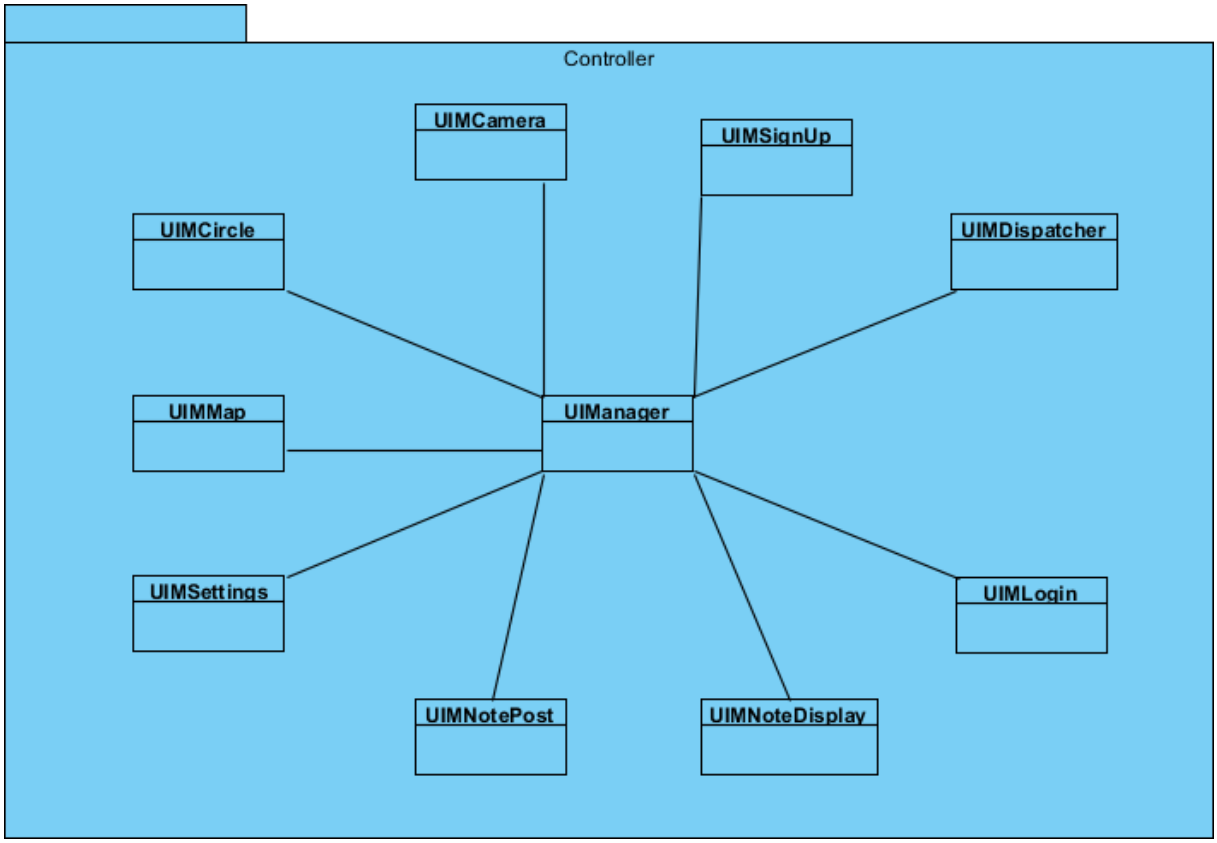

Figure 5: Controller subsystem in Client

UIManager: Abstract UI manager class that contains the most fundamental logic shared between all UI managers.

UIMMap: Manager class that is responsible of keeping the map up to date and processing inputs coming from the map interface.

UIMCircle: Manager class that contains the logic behind circle operations.

UIMSettings: Class that manages the settings of the software.

UIMNotePost: Manager that handles the operations regarding note posting.

UIMNoteDisplay: Manager class that manages the operations upon displayed note post, such as upvoting, reporting, etc.

UIMLogin: Class that manages the UI operations of user login

UIMSignUp: Class that manages the UI operations of user sign up.

UIMDispatcher: Manager class that changes the currently shown interface element.

UIMCamera: Manager class that contains the logic behind camera operations.

4.1.2. View Subsystem

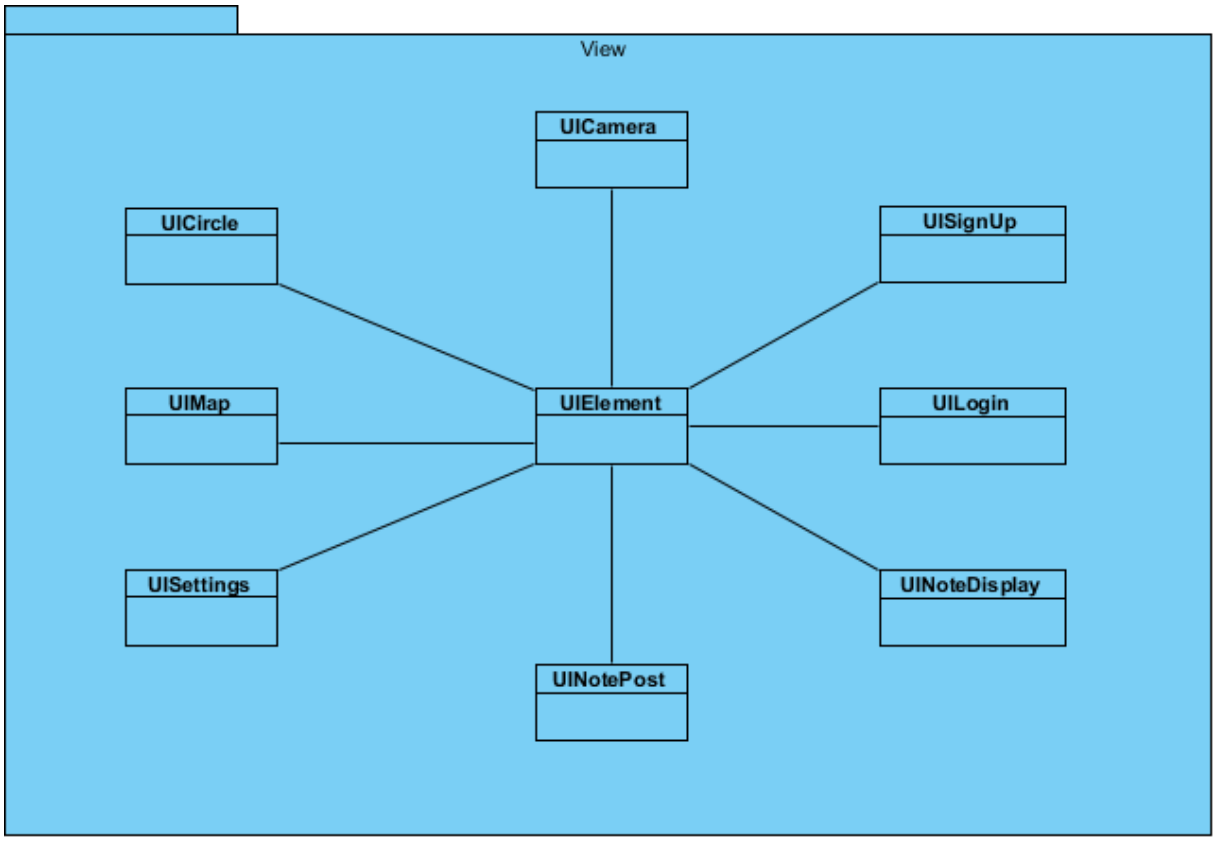

Figure 6: View subsystem in Client

UIElement: Abstract UI class that contains the most fundamental information and operations about user interface elements.

UIMap: Class that contains the information about what is shown on map screen.

UICircle: Class that contains the information about what is shown on circle screen.

UISettings: Class that contains the information about what is shown on settings screen.

UINotePost: Class that contains the information about what is shown on note posting screen.

UINoteDisplay: Class that contains the information about what is shown on note display screen.

UILogin: Class that contains the information about what is shown on login screen.

UISignUp: Class that contains the information about what is shown on sign up screen.

UICamera: Class that contains the information about what is shown on camera screen.

4.2. Server

Server side is the part of the system where data related tasks are completed. All the server side components of our application will be deployed using Amazon Web Services and its cloud infrastructure. There will be 3 distinct parts in the cloud; logic core (API), MS SQL database and S3 bucket. S3 bucket will hold the image data for the Notes and Augma Ads while the MS SQL will handle rest of the data like the user credentials, circle data and location data for the notes. Client side of the application will only be interacting with the API. Whenever there is a need to access the database or the S3 bucket, client application will request the necessary data from the API with its available functions. Then the API will fetch the requested data from either the database, S3 bucket or from both and compile it into a form that is easily understandable and readable from the client side and serve it in JSON form. This process ensures that clients are never in direct contact with the data storage parts of the application which provides an increased security and reliability. Also compiling the necessary information in the server side will lighten the already heavy load created by the AR and GPS features on the client side which will in turn increase battery life.

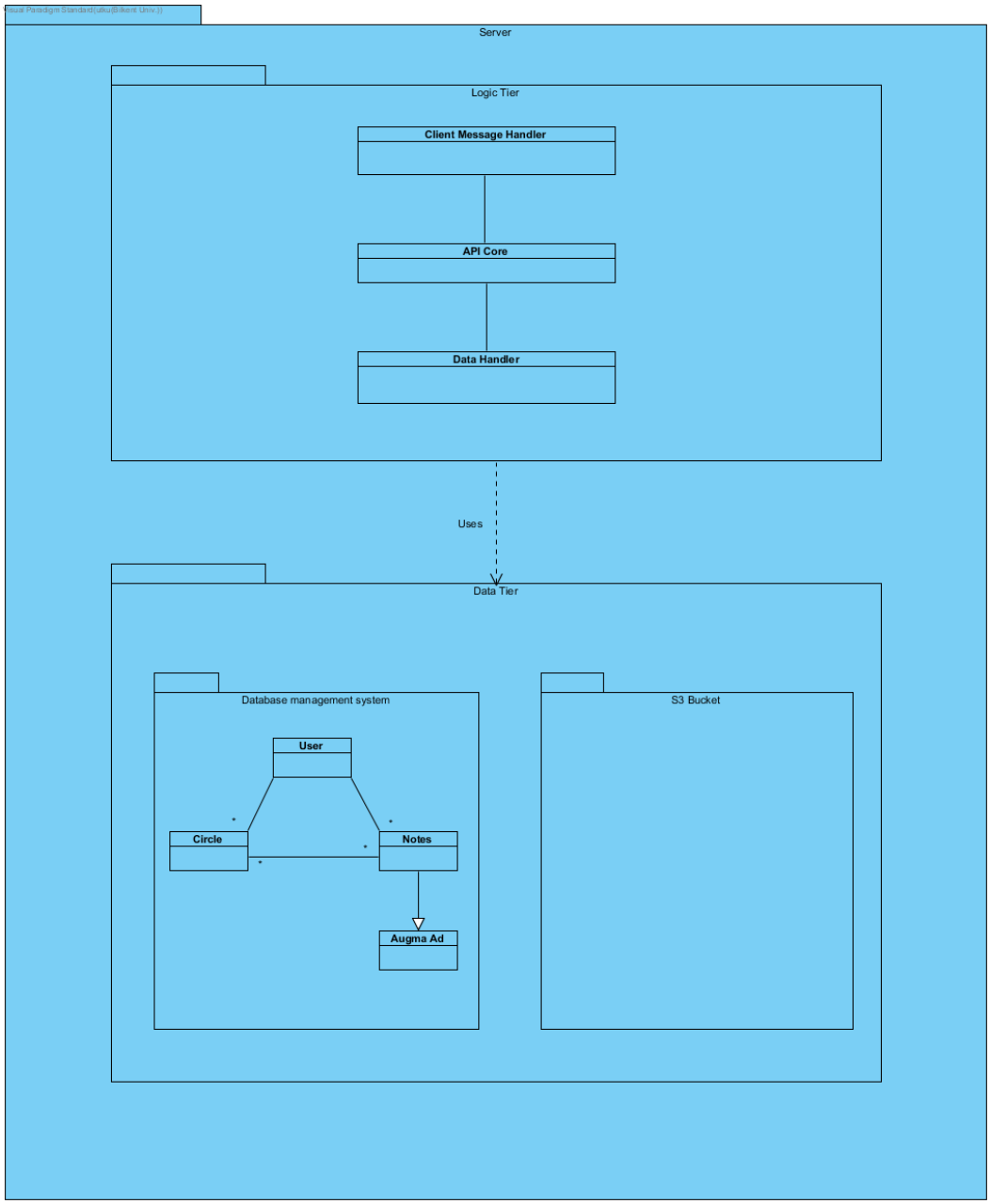

Figure 7: Detailed View of Server Subsystem

4.2.1. Logic Tier

Logic Tier is the layer which communicates with the database and client and processes the data incoming. This tier contains two handlers, Client Message Handler and Data Handler, and the API core, which deals with the core business logic. Client Message Handler gets requests from client and sends the output back to client after getting them processed in API core. Data Handler executes queries to access the data stored in database and passes the data to API core.

Client Message Handler: This handler communicates with the client, retrieves data from client, and send the processed data back.
API Core: API core constitutes the heart of logic tear and deals with the core business logic operations. When it is signaled by client message handler, It retrieves data from database and processes it as requested by client-side, transforms it into a deliverable format, JSON, to the client, then delivers the data to Client Message Handler for the rest of the operations. It can perform Create, Read, Delete, Update operations on database and it relies on REST.

4.2.2. Data Tier

Data Tier is the layer which manages the data of the application. This tier contains the Database Management System and S3 Bucket subsystems. Database Management System manages the relational database of the application. S3 Bucket subsystem manages the images of the notes that are uploaded to the database.

User: User data class represents the users of the application. This data class contains the data of the users like username, password, birthday, email.

Circle: Circle data class represents the circles of the application. This data class contains the data of the circles like circle name, description and users that are subscribed to the circles.

Notes: Notes data class represents the notes of the application. This data class contains the data of the notes that are uploaded such as the location of the note, the user who has left the note.

Augma Ad: Augma Ad data class represents the advertisements of the application. This data class contains the data of the advertisements such as the location of the advertisement, the company that has left the advertisement.

References

[1] WoodrowBarfield, Fundamentals of Wearable Computers and Augmented Reality, CRC Press, 2016.

[2] Matt Bower, Cathie Howe, Nerida McCredie, Austin Robinson & David Grover, «Augmented Reality in education – cases, places and potentials,» *Educational Media International,* 2014.

[3] Mark Billinghurst, Adrian Clark, Gun Lee, «A Survey of Augmented Reality,» *Foundations and Trends in Human-Computer Interaction,* cilt 8, pp. 73-272, 2014.

[4] Clemens Arth, Lukas Gruber, Raphael Grasset, Tobias Langlotz,Alessandro Mulloni, Dieter Schmalstieg, Daniel Wagner, «The History of Mobile Augmented Reality- Developments in Mobile AR over the last almost 50 years,» Inst. for Computer Graphics and Vision Graz University of Technology, Graz, 2015.

[5] Thomas Olsson, Kaisa Väänänen, «Expected user experience of mobile augmented reality services: A user study in the context of shopping centres,» *Personal and Ubiquitous Computing,* 2011.

[6] Philip Geiger, Marc Schickler, Rudiger Pryss, Johannes Schobel, Manfred Reichert, «Location-based Mobile Augmented Reality Applications,» 2014.

[7] Panos E. Kourouthanassis, Costas Boletsis, George Lekakos, «Demystifying the design of mobile augmented reality applications,» *Multimedia Tools and Applications,* 2013.

[8] Rüdiger Pryss,Philip Geiger,Marc Schickler,Johannes Schobel,Manfred Reichert, «Advanced Algorithms for Location-Based Smart Mobile Augmented Reality Applications,» *Procedia Computer Science,* cilt 94, pp. 97-104, 2016.

[9] Zhihan Lv ,Alaa Halawani , Shengzhong Feng , Shafiq ur Re´hman, Haibo Li, «PreprintTouch-less Interactive Augmented Reality Game on Vision Based Wearable Device,» 2014.

[10] «GooglePlay Store page,» Wallame Ltd, [Online]. Available: https://play.google.com/store/apps/details?id=com.wallame.

[11] «WallaMe,» Wallame Ltd, [Online]. Available: http://walla.me.

[12] «ITunes Store page,» Mirage Worlds, Inc., [Online]. Available: https://itunes.apple.com/us/app/mirage-world/id1225850968?ls=1&mt=8.

[13] «Mirage World,» Mirage Worlds, Inc., [Online]. Available: https://mirage.world.

[14] “Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing”, IETF, 2014. [Online]. Available: https://tools.ietf.org/rfc/rfc7230.txt

[15] “What is RESTful Web API?”, TechTarget, 2017. [Online]. Available: http://searchmicroservices.techtarget.com/definition/RESTful-API

[16] “SQL Server Documentation”, Microsoft, 2017. [Online]. Available: https://docs.microsoft.com/en-us/sql/sql-server/sql-server-technical-documentation

[17] “What is Amazon S3?”, Amazon, 2017. [Online]. Available: http://docs.aws.amazon.com/AmazonS3/latest/dev/Welcome.html

[18] ‘What is Amazon EC2?”, Amazon, 2017. [Online]. Available: http://docs.aws.amazon.com/AWSEC2/latest/APIReference/Welcome.html

[19] “Create a Network interface”, Amazon, 2017. [Online]. Available: http://docs.aws.amazon.com/AWSEC2/latest/APIReference/API_CreateNetworkInterface.html