
**Bilkent University**

Department Of Computer Engineering

# Senior Design Project

*Project short-name: AugCards*

## High Level Design Report

Yusuf Avcı, Burak Mutlu, Çerağ Oğuztüzün, Yiğit Görgülü, Bora Kurucu

Supervisor: Prof. Dr. Uğur Güdükbay
Jury Members: Asst. Prof. Dr. Can Alkan, Assoc. Prof. Dr. Çiğdem Gündüz Demir
Innovation Expert: Prof. Dr. Veysi İşler

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 CS491/2.

# Table of Contents

# High Level Design Report

*Project short-name: AugCards*

## 1 Introduction

Mobile games are becoming more popular day by day [1]. With mobile gaming, a new era in the gaming sector has emerged. People started to lose their habit and passion of playing games with physical equipment, such as board games and card games. The emergence of mobile gaming, due to its appealing graphics, ability to play online, and the sky-high imagination of the mobile game developers, lead people to quit playing card games physically. Additionally, card games have limited assets, rigid visuals, and static rules.

The main philosophy of AugCards is reuniting the tradition and old-school fun of playing card games with your friends sitting around the table, the dynamism of mobile games. AugCard gives users the freedom to create their own cards with their own assets, introduce their own animations, and specify their own game rules.

Imagine you and your friends sitting around a table and want to have a good time. AugCard helps to limit social isolation caused by individual gaming, and bring the people together to create a game. You and your friends would first open the AugCards Desktop application and create the game cards, the event triggers, game animations, rules. Even the complex rules can simply be introduced with the help of user-friendly design which makes use of flowcharts and such structures. After the game is complete, you and your friends can open the complementary AugCards mobile application and everyone can tune in to play the game you just created, a network is established among the table and multiplayer mode is enabled. The cards and the AR versions of the assets on the cards accompanied by animations are seen on the table by everyone looking through the cam of AugCards. If you are proud of the game you have created, you can share it on the AugCards platform for other users to play, and play a game made by another user.

In this report, we will provide a high level design of the system. First, design goals of the system will be discussed. Then, the current architecture of the AugCards software and the proposed software architecture will be described,

given attention to important topics including subsystem decomposition, hardware/software control and persistent data management. Subsystem services will be explained under headers: Mobile and Desktop subsystems. Additionally, considerations of various factors such as public health, safety, welfare in engineering will be discussed in the scope of AugCards. Last but not least, the teamwork details regarding the AugCards project will be specified in this report.

## 1.1 Purpose of the System

AugCards is proposed to be a sophisticated game making software which has many purposes regarding the system. The purposes of our system can be categorized under two headings: non-technical and technical purposes.

To start with the non-technical purpose of AugCards, the major one is to enable users to make their custom games through the AugCards platform. This will make card gaming more fun among players. AugCards system will automize game making by using AR technology. Additionally, the system proposes to be user-friendly and will have an interface regarding the user experience. The system will teach the users the process of making custom games to increase the user experience. At the end of the day, AugCards will be a software where users choose to make games and play them.

Regarding the technical purposes, AugCards system's purpose is to provide a system which regards issues such as error handling, scalability, security and efficiency seriously. System will not experience any decrease in the performance due to scalability issues, regarding the user and player number growth. the error-prone system will handle any errors that arise such that the user experience will not be affected in any ways. Last but not least, the technical purpose of the system includes security such that the users' data will be kept in private in AugCards.

## 1.2 Design Goals

### 1.2.1 Usability

- Game creation should not require programming knowledge.
- Tools should be self-explanatory, shouldn't require extensive tutorials or guides to be understood.

● The expression of complex card game rules will be simplified using flowcharts.
● AR-based graphics should have a refresh rate of at least 25 Hz to not affect game experience adversely.
● Popular image formats such as PNG and JPEG should be supported as assets
● Popular graphical model/animation formats like OBJ and COLLADA should be supported.

### 1.2.2 Reliability

● Should ensure that changes in game models are not lost on connection errors.
● Should have a back-up mechanism for ongoing games in a network failure situation.
● Contradictory game rules shouldn’t be allowed to cause errors.
● Cheats should be detected via checksums.

### 1.2.3 Maintainability

● Should be modular to reduce the complexity of the codebase.
● Network maintenance costs should be lower than 50TL per month while profits are low.
● Should use design patterns that will allow changing used libraries.

### 1.2.4 Accessibility

● Should be free to download.
● Should have integration with Google Services.
● Should require less than 1GB of RAM.

### 1.2.5 Extendability

● The addition of new possible game mechanics should not require changing existing code.
● The code itself should be properly structured,using design patterns and clever modularity.Adding new features and mechanics should require minimal or zero amount of change in the code.

## 1.3 Definitions, Acronyms, and Abbreviations

AR: AR (Augmented Reality) is an interactive experience of a real-world environment where the objects that reside in the real world are enhanced by computer-generated perceptual information. Accordance between real-world environment and computer generated information is ensured via sensors (or camera) and algorithms.
Action: Actions are building blocks of events. 15 different actions with different parameters are available to the game creators.
Event: Game rules are implemented by using events. Events consist of actions. Custom events are triggered by the main event.
Card: Cards are customizable and interactive game objects.
Deck: Card Collections.
Attribute: Properties of game objects.

## 1.4 Overview

AugCards is a card game development tool where users can create their own cards and build custom games to be played on Android devices with multiplayer capabilities and AR supported visuals. The tool transforms the user-designed game logic and cards into AR multiplayer mobile games that can be played instantaneously by a group of friends around a table. Currently, there are engines that are aimed at making card games, however, there is no engine that supports mobile device integration and makes use of AR supported visual experience.
AugCards will also contain a cloud service where users can share the games they created for other people to see and play. Thanks to the AR system, the experience of playing a game will feel very similar to playing a game with a group of friends around a table, as the cards and effects will all be animated and displayed to the user. Furthermore, by using systems similar to flowcharts, we intend to make expressing complex rules of the game less of a challenge for creators. It will provide a custom diagram model which will help users design their games.
The project will consist of two applications: one desktop application and one Android application. The game making part, where the user designs their own assets, cards, events, triggers, set of game rules, and card properties, will be

provided via the desktop application. After the game is made and ready to be played, each user will use their mobile phones and use the complementary AugCards mobile app to see the AR visuals and cards on the table and play the game.

# 2 Current Software Architecture

First of all, since we are creating something similar to a card game creation engine, Unity and Unreal Engine come to mind as existing software to develop games, which also include card games. However, the issue with these game engines is, even though they are too general purpose, and require a deep knowledge of programming to even start creating a game. With AugCards, we want to simplify the process of making card games and allow people with creative ideas who may not have the required knowledge to also create their games.

For card game creation specifically, different card game development tools exist on the Internet. One of the most prominent ones is Dulst, a platform which allows its users to develop and publish card games with their own custom rules and cards [2]. The users can then get feedback on the games that they have created from the community. Dulst tries to simplify the process of creating games by allowing developers to create events and triggers easily, however, it still seems too complicated for a non-programmer to understand. We want to simplify this process even further as well as extending the visuals (through AR) and accessibility (mobile gameplay) of the generated game.

# 3 Proposed Software Architecture

AugCards uses both mobile and desktop platforms. Hence, AugCards is composed of many subsystems which enable the information exchange between platforms and enable a smooth implementation for the software. In this section of the report, AugCards' substems, their interactions and the classes will be shown to the reader.

## 3.1 Overview

AugCards' subsystem decomposition consists of a Mobile subsystem and a Desktop subsystem regarding the platforms which will be used in the software

by the users. A subsystem diagram and composition carries great importance for our project, as It displays the interactions and information exchange between the systems, and how It composes a system of harmonically working subsystems. The information exchange that is happening between desktop and mobile systems includes the compilation of the game created on the desktop, in the mobile system. For example, the clarification of this process is of great importance regarding the implementation of both of the systems such that less error is faced by the developers. Regarding these issues, in our report, we gave great consideration into subsystem composition and software architecture designs.

## 3.2 Subsystem Decomposition

Our system is divided into three main subsystems: desktop, mobile and cloud. The explanation of the subsystems and their components is provided in detail in Section 4 of the report. Our subsystem decomposition diagram is provided in Figure 1 (on the next page).

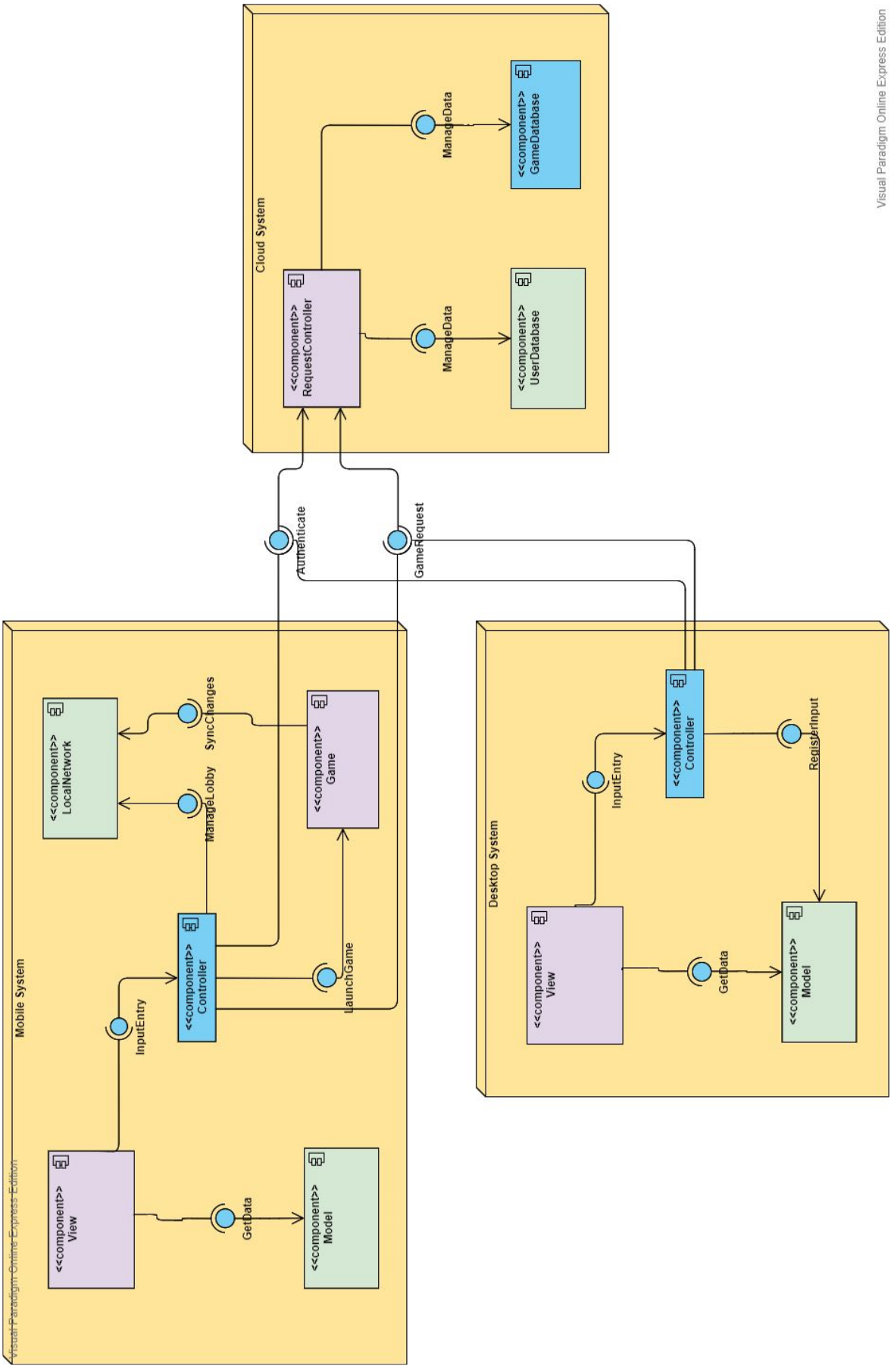


Figure 1: Subsystem decomposition diagram

## 3.3 Hardware/Software Mapping

Our project consists of a game creation environment which will require a desktop computer and a game consuming environment which will require an Android device. The project will require Android version 8+ because we will use Google's ARCore[3] which requires up-to-date devices with high processing power. Phones will require a camera as well to use the AR. Desktop applications will need internet connection to upload created games to the server and the Android devices will require internet connection to download game data and to play games. We will use Firebase[4] as our database to store the game data in JSON format. Our hardware/software mapping is provided in Figure 2.

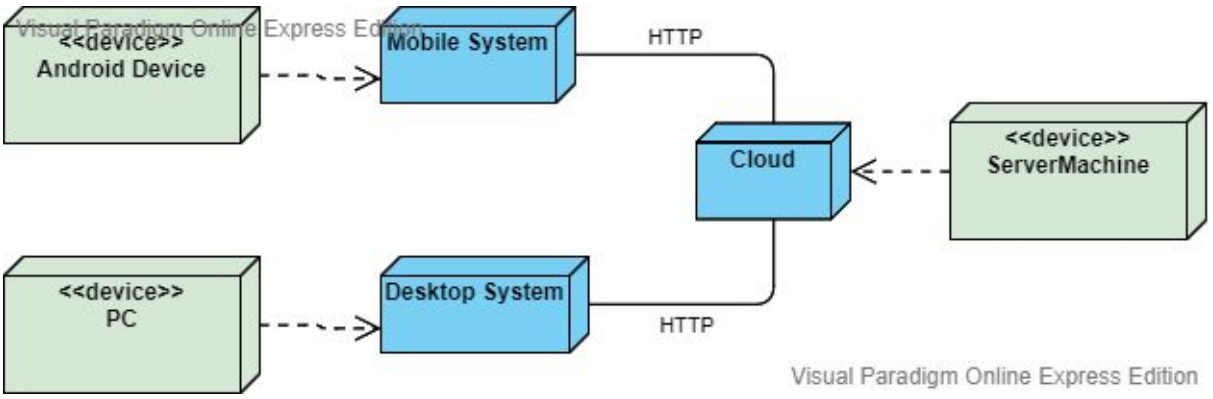


Figure 2: Hardware/Software Mapping

## 3.4 Persistent Data Management

Data management is a critical part of our application. In AugCards application, there are two types of data as user data and game models data. Both of them, data corruption constitutes a significant problem since subsystems are likely to fail in such cases. Any corruption in user data may result in that user cannot login to desktop/mobile systems and they cannot access their accounts, games, etc. Any corruption in game models data may lead to design errors in desktop system and it may lead to malfunctioning of the game in mobile system. In our systems, it is not a crucial requirement to access data fast because login and game downloading are not minimum transmission-rate requiring tasks. However, data storage efficiency is an important consideration, especially for game models data. This type of data is likely to occupy large storage space with game

variables, rules, graphical assets, etc. and there will be a high storage cost for increasing space requirement since a cloud system will be used for storage.
Managing user data, it is best to use a relational database to store/access all information like users' email, password, favorite games, game comments and games' identifiers, creators.
Managing game models data, as mentioned earlier, Json format will be used to store required information. Each game will be stored in their own repository. The development and shared version of games will be kept in separate directories to ensure continuous development progress of shared games. There will be a compression applied on graphical assets of games to reduce space requirements. While developing a game in the desktop system, a local repository will be created to prevent any data loss in development progress. While playing a game in the mobile system, a local repository will be used to launch games instantly and to prevent excessive internet usage. The games will be installed into such repositories when downloading and they will be updated if necessary.

## 3.5 Access Control and Security

The access control and security we took are different for the desktop application and our mobile application, thus, these will be discussed in two different sections. One part that is the same between the two systems is the way we will keep user accounts. We will keep the passwords in a hashed format, and the email addresses will be kept secret and only used for account creation and verification purposes.

### 3.5.1 Desktop Application Access Control and Security

In the desktop application, developers will be required to create an account and log in to be able to keep the games they develop. This will ensure that each developer will be able to access their own game files and nothing else on the server. This will help to avoid cases of plagiarism between games and access to unreleased games. Each user will be assigned a folder on our server, which they can use to keep relevant game files.

### 3.5.2 Mobile Application Access Control and Security

In the mobile application, the users may log in or they can use our application as a guest. If they choose to log in, their liked and downloaded game information will be kept on our servers privately. This information will only be used to restore user downloads on a new phone. To be able to like games, users will be required to be logged in, however, they can download and play games as guests. This will be done so that bots to promote games are not created. The JSON files that are transferred to the mobile application will be encrypted so that not everyone can read the details of the games.

## 3.6 Global Software Control

In this part of the report, the process of how the whole system is controlled will be discussed, on a global scale. The synchronization of the mobile and desktop subsystems will be discussed.

In AugCards, there are two main subsystems, which are Mobile and Desktop subsystems. The game is created by the user using the Desktop subsystem of AugCards, the information of the game is compiled and to be run on the Mobile subsystem of the AugCards software. We divided this process in a number of phases for clarification:

### 3.6.1 Game Creation Phase

This phase is the phase where the user implements the custom games through the Desktop subsystem of AugCards. The user adds game components, provides assets and establishes rules. This information of the game is translated into a code to be parsed by the Mobile subsystem to be run. A JSON file specific to the custom game is created containing this information.

### 3.6.2 Processing Phase

After the JSON file is received by the Desktop subsystem, the Mobile subsystem reads the information of the game created by the user and establishes the game to be played on mobile platforms. The information read regarding the game is joined by AR libraries to be implemented in the Mobile platform implementation of AugCards. Additionally, an ideal configuration of ports is aimed to be found regarding the prevention of delays in the client-side of this connection.

## 3.7 Boundary Conditions

Different types of boundary conditions exist in AugCards, which are initialization, error and termination.

### 3.7.1 Initialization

On the desktop side, to create games, the user should have an account in AugCards desktop and login. On the mobile side similarly, the user should have an account in the relative app store and login. On the desktop side to upload the created games the connection with the cloud should be established, as well for the mobile side to download the games.

### 3.7.2 Error

There may be multiple reasons for error. Some of them may occur when devices run out of battery while the app is open. To prevent this, both the desktop and the mobile app are not usable when the battery level is below a specified value. Another error may occur when internet connection fails while uploading/downloading a game to/from the cloud. When that happens, the operation is cancelled, and the already uploaded/downloaded content is removed automatically.

### 3.7.3 Termination

Upon termination, on the desktop side, any unsaved changes will not be saved and will be removed automatically from the system. Similarly, upon termination, on the mobile side, user's game progress will not be saved.

# 4 Subsystem Services

These subsystems are divided into further components, the details of which are provided below.

## 4.1 Desktop Subsystem

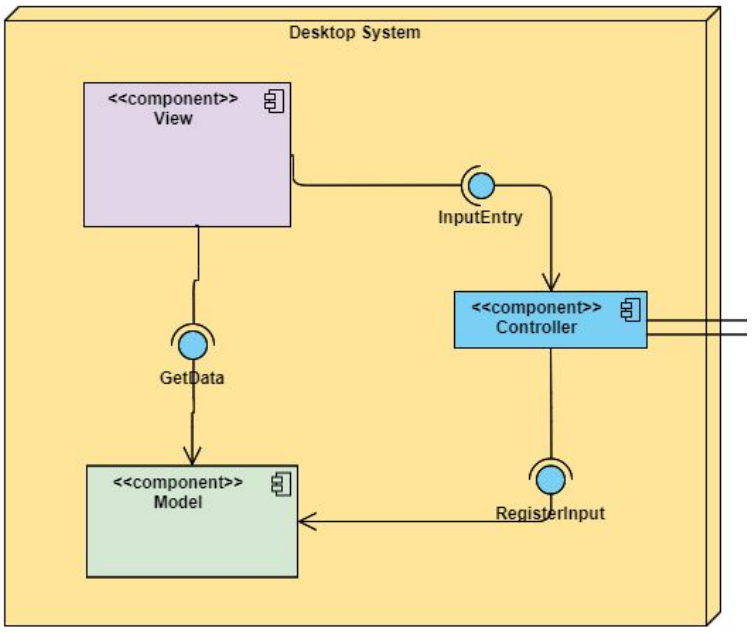


Figure 3: Desktop subsystem's components

The desktop subsystem is responsible for handling the Model and View components of the desktop application of AugCards which uses a Controller component for receiving and providing information from the Mobile subsystem of AugCards.

View: The View component is responsible for the UI operations of the Desktop application. It is related to the Controller via input entries.
Model: The Model component implements the logic of the Desktop application which shares data between the View component and is related to Controller with its register input.
Controller: The Controller subsystem is responsible for handling the information exchange between the Mobile and Desktop applications.

### 4.1.1 View Component

The View component manages the UI of our desktop application, communicating with the Model component through the Controller component.

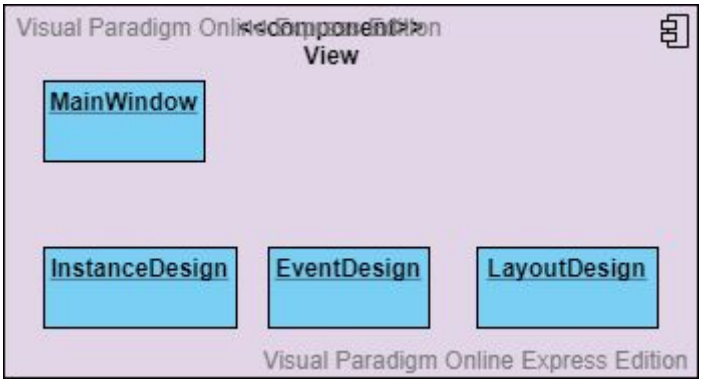


Figure 4: View component and its classes

MainWindow: The main window of the application. It contains other components and lets the user see what they are doing.
InstanceDesign: Where the user designs instances.
EventDesign: Where the user designs events.
LayoutDesign: Where the user designs layouts of their game.

### 4.1.2 Model Component

Model component is supposed to store the information for designed games, that is, defined objects and game rules.

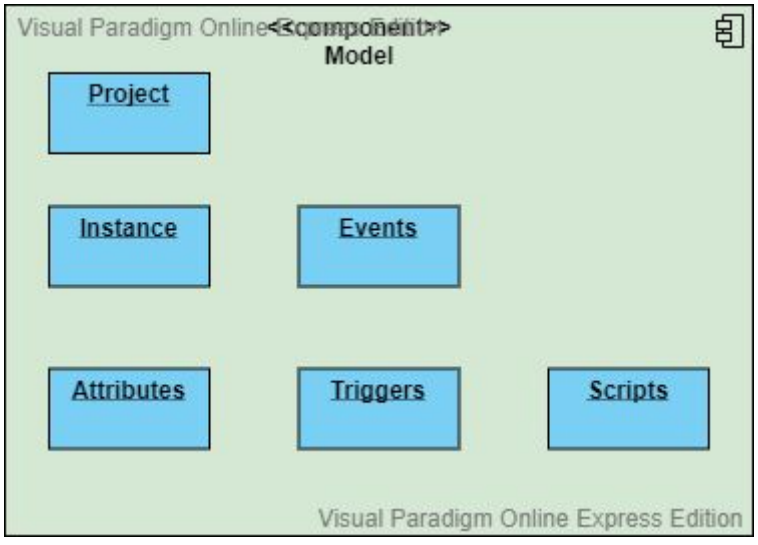


Figure 5: Model component and its classes

Project: A class to represent a particular game project. It stores all custom elements like **Instances** and **Events.**

Instance: A class to represent a specific game object like card, player, etc. It can store any custom **Attributes** defined.

Events: A class to represent a specific game event like game-init, card-attack, etc. It should store all its own **Triggers** defined.

Attributes: A class to represent a property of defined **Instance.** It should store its reference name, type and initial value.

Triggers: A class to represent a handling mechanism of an **Event.** It should store a sequence of **Scripts** to define the handling mechanism.

Scripts: A class to represent an effective change on an **Attribute.** It should store its effect type, focus attribute and modifiers.

### 4.1.3 Controller Component

Controller component basically manages the application flow according with incoming inputs from the View component. Application flow contains authentication and exporting game data.


Figure 6: Controller component and its classes

AuthenticationManager: A manager class to handle authentication requests for the desktop system. It should communicate with **Cloud Subsystem** to check users' access permission.

ModelManager: A manager class to handle manipulations on game models according to the inputs from View Component. It should apply manipulations to Data Component and generate output files for game models.

## 4.2 Mobile Subsystem

The details of the mobile subsystems is provided below.

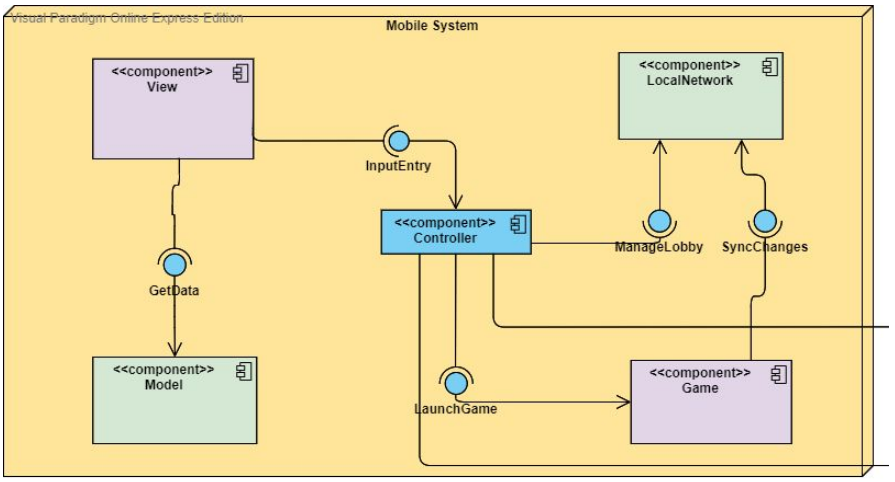


Figure 7: Mobile subsystem's components

### 4.2.1 View Component

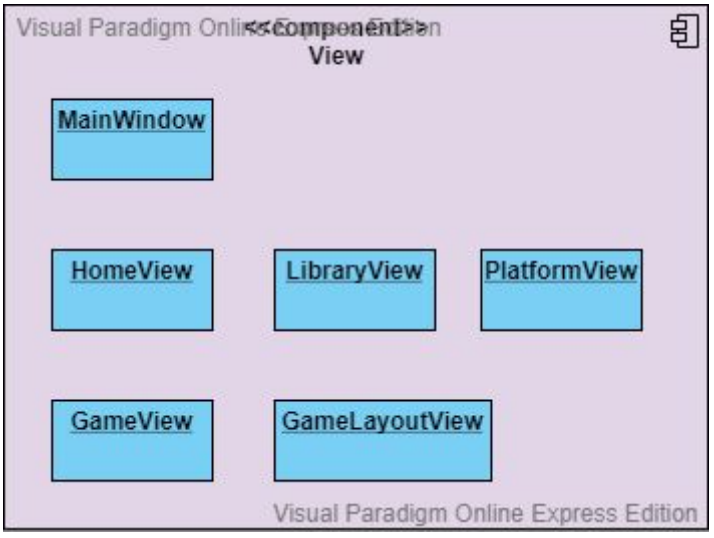


Figure 8: View component and its classes

The View component is composed of the following parts:

MainWindow: A class which provides the view of the main window of AugCards.

HomeView: A class that is responsible for the display of the home page.

GameView: A class that represents the view of the game to be played in AugCards.

LibraryView: A class which provides the library view of AugCards which displays a number of games.

GameLayoutView: A class to represent the layout of where each game instance will be placed for a game.
PlatformView: A class that is to display the platform where users can search for games.

### 4.2.2 Model Component

The Model component keeps information about the app itself and the game is managed by another component. This component keeps user information such as authentication information and the user's game library, as well as the game lobby.

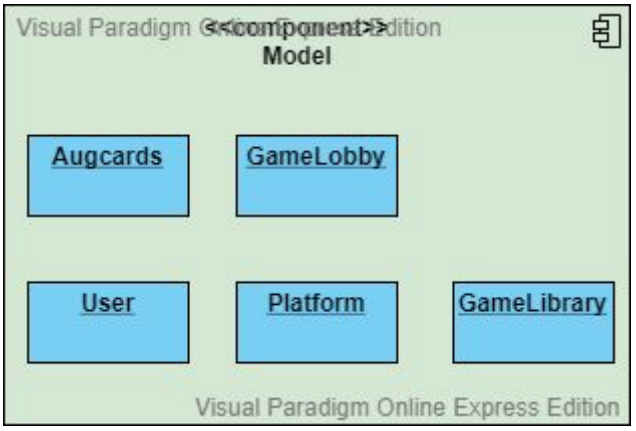


Figure 9: Model component and its classes

Augcards: The mainframe of the mobile game.
GameLobby: Information about the running game lobby is kept here.
User: Information about the user themselves will be managed by this class.
Platform: The shared games and their community pages will be kept here.
GameLibrary: The user's game library information will be kept here.

### 4.2.3 Controller Component

Controller component basically manages the application flow according with incoming inputs from the View component. Such application flow may contain authentication or creating/joining a lobby or launching a game session.

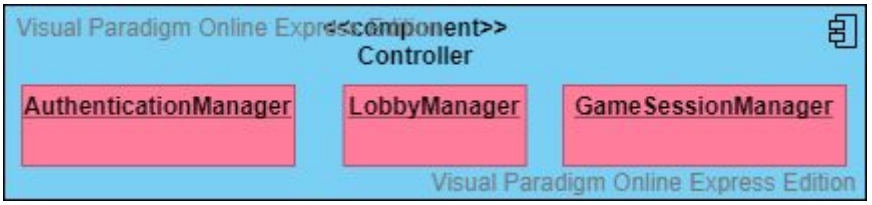


Figure 10: Controller component and its classes

AuthenticationManager: A manager class to handle authentication requests. It should communicate with **Cloud Subsystem** to check users' access permission.
LobbyManager: A manager class to handle lobby joining/creation requests. It should communicate with **Local Network Component** to establish connection with a lobby.
GameSessionManager: A manager class to handle game launch requests. It should communicate with **Game Component** to run a game execution process.

### 4.2.4 Game Component

Game component is responsible for executing the actual game. It loads a downloaded game model and it's assets and parses the data to create a game session.

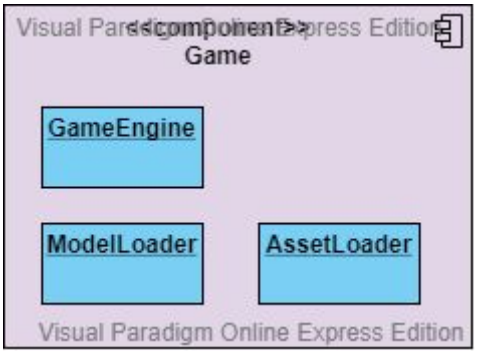


Figure 11: Game component and its classes

GameEngine: Executes the game using the game model and game assets.
ModelLoader: Loads the game model data for the game.

AssetLoader: Loads the game assets.

### 4.2.5 Local Network Component

Local Network component manages the hosting for game lobbies/sessions on a local network. Since it will use P2P structure, there will be both sending and receiving tasks to/from a connected host.

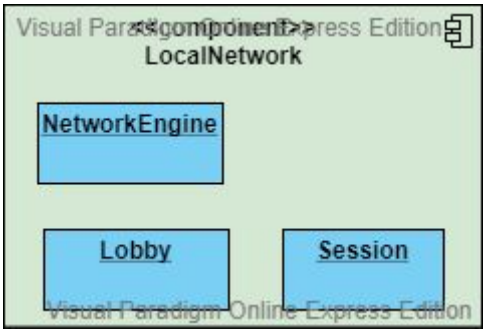


Figure 12: Local network component and its classes

Lobby: A class to represent lobby information of a host. It should contain users connected to the host.
Session: A class to represent session information of a host. It should update the state of content according with changes on the host.
Network Engine: A manager class to handle connection to the host. It executes sending and receiving tasks through the connection.

## 4.3 Cloud Subsystem

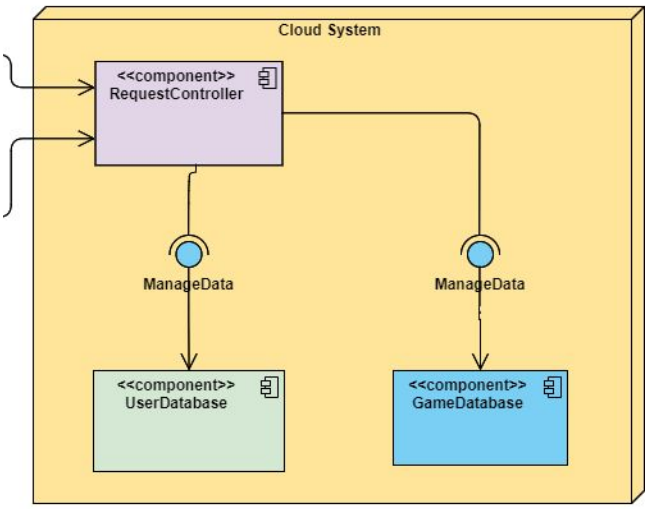


Figure 13: Cloud subsystem's components

The cloud subsystem is responsible for the delivery of services which include servers, networking and databases. AugCards will use Firebase for the purpose of database services [4].

RequestController: The RequestController component is responsible for handling the request made to the UserDatabase and GameDatabase.
UserDatabase: UserDatabase is a database consisting of the information regarding the users of AugCards.
GameDatabase: GameDatabase is a database consisting of the information regarding the games in AugCards.

### 4.3.1 Request Controller Component

The request controller manages incoming requests to the cloud subsystem and redirects them to the appropriate database after authentication.

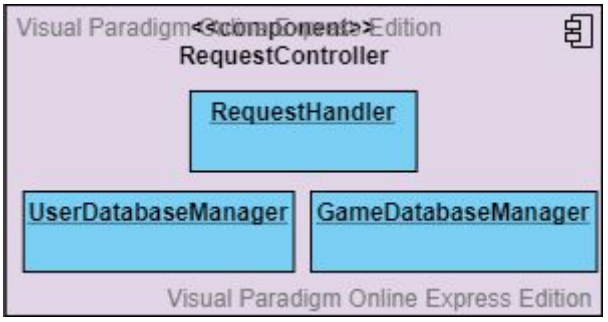


Figure 14: Request component and its classes

RequestHandler: The general request handler, manages authentication and data control.
UserDatabaseManager: The interface to the user database, used to access that database.
GameDatabaseManager: The interface to the game database, used to access that database.

### 4.3.2 User Database Component

User Database Component is basically a database structure to store user data like users' name, email, password or games' identifiers, creators, comments etc.


Figure 15: User database component

A further relational database model will be used to design this component.

### 4.3.3 Game Database Component

Game database component is used to store different games' data. These data include game events (therefore, rules), game visuals and animations and game content such as card types and cards. Data other than visuals are stored in Json format and the visuals are stored in binary format.


Figure 16: Game database component

# 5 Consideration of Various Factors in Engineering Design

Various factors were considered while designing our project. The factors are scored on a scale of 0-10, taking how much they might be affected by our project into consideration. The scores can be found in Table 1 (below). The reasonings for the scores and more details can be found in the respective subsections.

| Factor | Score (0-10; 0 not affected, 10 very crucial) |
|---|---|
| Public Health | 1 |
| Public Safety | 6 |
| Public Welfare | 5 |
| Global | 4 |
| Social | 7 |
| Cultural | 6 |
| Economic | 6 |

Table 1: How our project considers some factors

## 5.1 Public Health

We do not believe that public health will be very much affected by our application, since our application is just a way for users to create and play games with their friends. The users' mental health may improve by socialising, and that's why 1 is given as a score.

## 5.2 Public Safety

Since the app offers a variety of options to users, users may use this variety to create games, images or characters that may include offensive figures to others, which might possibly threaten public safety. In order to prevent this, AugCards will have a way of filtering the media, game style and other components that may be a potential threat to the users. Public Safety and ways to protect it must be one of our priorities, that's why 6 is given as a score.

## 5.3 Public Welfare

We believe the welfare of society will be affected by our game since we will provide a way for people to enjoy themselves and spend leisure time with their friends. This will lead to people becoming happier and releasing stress. However, welfare is not only about happiness, so 5 is given as a score.

## 5.4 Global Factors

Since our game will allow users from all around the world to create and share games, we believe our application will have a global impact. However, the card game community is not a very large niche, so we do not expect a ground-breaking impact and that's why 4 is given as a score.

## 5.5 Social Factors

We believe people will be able to socialise using our application and they will be generally happier and more social. Each game represents a community through a platform page in which players interact with the game and others. Also, custom games on AugCards application are AR featured card games and are supposed to be played with a group of people around a table, which is another socializing factor.  That's why 7 is given as a score.

## 5.6 Cultural Factors

Our application will allow users from different countries to create and share games with each other. We believe that each game will have cultural influences from the developer and when a foreigner plays their game, they will be exposed to a new culture. This will lead to a cultural exchange between different countries which is why 6 was given as a score.

## 5.7 Economic Factors

Card game creating companies that sell printed cards come to mind when talking about factors, and we need to consider their revenues as well when we are creating our application. The card game developers may also need to paid for the work that they do, which also needs to be considered, so economic factors are given a score of 6.

# 6 Teamwork Details

In this section, each member's contributions to the project will be explained.

## 6.1 Contributing and Functioning Effectively on the Team

| Member | Contributions |
|---|---|
| Çerağ | Worked in the implementation of the Desktop application as a backend developer. |
| Yusuf | Worked on designing the event generation and parsing logic, and related GUI. Worked on mobile GUI as well. |
| Yiğit | Designed the JSON format for data transfer between mobile and desktop systems. |
| Burak | Worked on backend development of Desktop application, specifically, custom game concepts and instance design. |
| Bora | Worked in the implementation of the Desktop application as a frontend developer. |

Table 2: Contributions of each member

## 6.2 Helping Create a Collaborative and Inclusive Environment

| Member | Contributions |
|---|---|
| Çerağ | Helped set up meetings and resolved most of the conflicts among group members. |
| Yusuf | Helped setting up a project management software (Trello). Suggested enforcing code reviews. |
| Yiğit | Set up the GitHub repositories and Google Drive folder. |
| Burak | Participated in group meetings and other scheduled works. |
| Bora | Created a written to do list & summary after every meeting. |

Table 3: How each member helped create a collaborative and inclusive environment

## 6.3 Taking Lead Role and Sharing Leadership on the Team

| Member | Contributions |
|---|---|
| Çerağ | As a natural leader, I took the group's thought process to another level when the members were indecisive. |
| Yusuf | Took an important part in the decisions regarding the implementation. All the final decisions were democratic, however. |
| Yiğit | Took part in the decision making process and dividing tasks. |
| Burak | Contributed into the decision making process. |
| Bora | Made sure that everyone took equal responsibility. |

Table 4: How each member took leadership roles

# 7 Glossary

*Game Instance* A specific instance in AugCards to represent the custom game objects like card, player, avatar defined by the developer.

*Game Event* A specific event in AugCards to represent the custom game events like attack, play card or navigate to the next turn, defined by the developer.

*Event Trigger* An event trigger represents the trigger mechanisms for defined events.

*Game Rules* A set of conditioners for a game specified by Developers in AugCards to represent the rules of the created game.

*Session* A session refers to a game session in which the game is played by the players.

*Asset* An abstract class to generalize the graphical elements.

*Animation* A specific Asset to represent the pre-designed transform sequence for graphical models within animation data.

*Developer* A developer of AugCards represents the type of actor in the system which creates card games.

*Player* A player of AugCards represents the type of actor in the system which attends games.

*Platform* A platform in AugCards represents the common point where users and developers meet through shared games.

*Game Library* A game library in AugCards represents the customizable game storage where users can insert new ones and pick favorites.

*Game Lobby* A game lobby in AugCards represents created and in-preparation game sessions in which players can join.

*GPU* Graphics Processing Unit. GPU is designed for handling graphics operations, including 2D and 3D calculations to render 3D graphics [5].

*Git* A version control system used for project teams for reviewing and tracing code changes.

*GitHub* An online platform which hosts software development versions for software development teams by using Git.

*Trello* Trello is a collaboration tool that organizes your projects into boards [6].

*Augmented Reality* Augmented Reality is a technology for producing an enhanced environment [7].

*Android System* The Android operating system is a mobile operating system developed for mobile platforms.

*Discord* an American VoIP, instant messaging and digital distribution platform designed for creating communities [8].

*WhatsApp* WhatsApp is a messenger cross-platform instant messaging application.

*Dulst* Dulst is an online card game playing software [9].

*UML* Unified Modeling Language, is a standardized modeling language consisting of an integrated set of diagrams [10].

Vuforia Vuforia is an engine that supports the use of AR and computer vision functionalities [11].

*Firebase* Firebase is Google's mobile platform that helps you quickly develop high-quality apps and grow your business [4].

*JSON* JSON (JavaScript Object Notation) is a lightweight data-interchange format [12].

# 8 References


[1] "Topic: Mobile gaming." [Online]. Available: https://www.statista.com/topics/1906/mobile-gaming/. [Accessed: 09-Oct-2020]

[2] "Dulst." [Online]. Available: https://dulst.com/. [Accessed: 08-Oct-2020]

[3] "ARCore overview | Google Developers," *Google*. [Online]. Available: https://developers.google.com/ar/discover. [Accessed: 27-Dec-2020].

[4] *Google*. [Online]. Available: https://firebase.google.com/. [Accessed: 27-Dec-2020].

[5] "GPU (Graphics Processing Unit) Definition." [Online]. Available: https://techterms.com/definition/gpu. [Accessed: 21-Nov-2020]

[6] Trello, "What is Trello?" [Online]. Available: https://help.trello.com. [Accessed: 21-Nov-2020]

[7] "augmented reality." [Online]. Available: https://www.dictionary.com. [Accessed: 21-Nov-2020]

[8] Contributors to Wikimedia projects, "Discord (software)," 30-Jan-2016. [Online]. Available: https://en.wikipedia.org/wiki/Discord_(software). [Accessed: 21-Nov-2020]

[9] "Dulst." [Online]. Available: https://dulst.com/. [Accessed: 21-Nov-2020]

[10] "What is Unified Modeling Language (UML)?" [Online]. Available: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/. [Accessed: 21-Nov-2020]

[11] "Vuforia Developer Portal." [Online]. Available:

https://developer.vuforia.com/. [Accessed: 21-Nov-2020]

[12] "Introducing JSON," *JSON*. [Online]. Available: http://www.json.org/json-en.html. [Accessed: 27-Dec-2020].