

COMP4910 Senior Design Project 1, Fall 2020

Advisor: MUTLU BEYAZIT

**CYBER DRONE: Business System
Requirements Specifications Document**

23.05.2021

REVISION Final

BY:

EGE ERBERK USLU, 16070001007

BORA GÜZEL, 17070001010

SİMGE BİNNAZ ÖZDEMİR, 17070001001

Revision History

Revision	Date	Explanation
1.0	18.01.2021	Initial requirements
2.0	14.05.2021	The use case diagrams are detailed according to the last version of the system in Section 3.
Final	23.05.2021	Section 3 and 4 are detailed according to the final version of the system. Final Version of RSD.

List of figures

Figure 1:Use Case of Cyber Drone Desktop part.....	7
Figure 2:Use Case of Cyber Drone Web part.....	8
Figure 3: Class Diagram of Cyber Drone Online User Server Part.....	8
Figure 4: Class Diagram of Cyber Drone Web Server Part.	9
Figure 5:Cyber Drone Unity part-1.....	9
Figure 6: Class Diagram of Cyber Drone Unity Part-2.....	10
Figure 7: Class Diagram of Cyber Drone Desktop Canvas part.	10
Figure 8: Class Diagram of Cyber Drone Client Server Part.....	11
Figure 9:Cyber Drone Packages UML.....	15
Figure 10:Use Case Diagram of Cyber Drone Web part.....	15
Figure 11:Use Case Diagram of Cyber Drone Web application User part.....	16

List of Tables

Table 1:Requirement list of Cyber Drone Simulation Project.	14
Table 2:Actor table.....	14

Contents

List of figures	3
1. Introduction	7
2. Requirements List	11
3. Actors and Use Case Diagram	14
3.1. Actors in Cyber Drone	14
3.2. Cyber Drone Use Case Diagram Groups	14
3.3. Cyber Drone Use Case Diagrams	15
3.3.1. Web Application Subsystem Use Case Diagrams	15
3.3.2. Desktop Application Subsystem Use Case Diagrams	16
4. SYSTEM USE CASES	17
4.1. DESKTOP APPLICATION Systems Use Case Descriptions	17
4.1.1. AUTHENTICATION Use Case Descriptions	17
4.1.1.1. Use Case 1 Login Description	17
4.1.1.2. Use Case 2 Display Login Error Description	17
4.1.1.3. Use Case 3 Save Auth Token Description	17
4.1.1.4. Use Case 4 Verify Username & Password Description	18
4.1.1.5. Use Case 5 Read Related Information from Database Description	18
4.1.1.6. Use Case 6 Get Server's Public Key and Encrypt Information with RSA Description	18
4.1.1.7. Use Case 7 Navigate Page Description	18
4.1.2. GAMEPLAY Use Case Descriptions	18
4.1.2.1 Use Case 1 LitfOff Description	18
4.1.2.2 Use Case 2 Landing Description	18
4.1.2.3 Use Case 3 360DegreePerspective Description	18
4.1.2.4 Use Case 4 InitEnvandDrone Description	19
4.1.3. MAIN PAGE MANAGEMENT Use Case Descriptions	19
4.1.3.1 Use Case 1 Quit Description	19
4.1.3.2 Use Case 2 Replay Description	19
4.1.3.3 Use Case 3 Settings Description	19
4.1.3.4 Use Case 4 Start Description	19
4.1.3.5 Use Case 5 RecordOnOff Description	19
4.1.3.6 Use Case 6 UploadRecord Description	19
4.1.3.7 Use Case 7 StartMatch Description	19
4.1.4. SIMULATION MAIN CONTROL Use Case Descriptions	20

4.1.4.1 Use Case 1 Play Description	20
4.1.4.2 Use Case 2 DroneCustomize Description	20
4.1.4.3 Use Case 3 MapSelection Description	20
4.1.4.4 Use Case 4 WeatherSelection Description	20
4.1.4.5 Use Case 5 En_Disable_AI Description	20
4.1.5. SIMULATION REPORT Use Case Descriptions	20
4.1.5.1 Use Case 1 CollisionDetection Description	20
4.1.5.2 Use Case 2 BugDetection Description	20
4.1.5.3 Use Case 3 MakeReport Description	21
4.1.6. MULTIPLAYER Use Case Descriptions	21
4.1.6.1 Use Case 1 CreateandJoinRoom Description	21
4.1.6.2 Use Case 2 CreateNewRoom Description	21
4.1.6.3 Use Case 3 LeaveRoom Description	21
4.1.6.4 Use Case 4 JoinRandomRoom Description	21
4.1.6.5 Use Case 5 PhotonConnectionUsing Description	21
4.2. WEB APPLICATION Use Case Descriptions	21
4.2.1. AUTHENTICATION Use Case Descriptions	21
4.2.1.1. Use Case 1 Login Description	21
4.2.1.2. Use Case 2 Display Login Error Description	22
4.2.1.3. Use Case 3 Save Auth Token Description	22
4.2.1.4. Use Case 4 Verify Username & Password Description	22
4.2.1.5. Use Case 5 Read Related Information From Database Description	22
4.2.1.6. Use Case 6 Get Server's Public Key and Encrypt Information with RSA Description	22
4.2.1.7. Use Case 7 Change Password Description	22
4.2.1.8. Use Case 8 Navigate Page Description	22
4.2.1.9. Use Case 9 Forget Password Description	23
4.2.2. DATA MANAGEMENT Use Case Description	23
4.2.2.1. Use Case 1 ShowSystemLog Description	23
4.2.2.2 Use Case 2 Download Record Description	23
4.2.2.3 Use Case 3 Show All Users Description	23
4.2.2.4 Use Case 4 User Profile Update Description	23
4.2.2.5 Use Case 5 Replay Description	23
4.2.2.6 Use Case 6 Show User Profile Description	24
4.2.2.7 Use Case 7 Show Online User Description	24
4.2.3. USER MANAGEMENT Use Case Description	24
4.2.3.1 Use Case 1 AddUser Description	24

4.2.3.2 Use Case 2 DeleteUser Description	24
4.2.3.3 Use Case 3 UpdateUser Description	24
4.2.3.4 Use Case 4 ShowAllUser Description	24
4.2.3.5 Use Case 5 ChangeUserType Description	24
4.2.3.6 Use Case 6 Connection Description	25
4.2.3.7 Use Case 7 Get User List Description	25
4.2.3.8 Use Case 8 Handshaking Description	25
4.2.3.9 Use Case 9 Disconnect Description	25
5. Non-Functional Requirements	25
References	25

1. Introduction

The purpose of the software project is to develop a 3D simulation software in a game engine such as Unity or Unreal Engine environment, in C# or C++ programming language to do mainly the following:

- 1- Users should be able to orient the drones like up, down, right, left direction and observe the flight of drones in different environmental conditions.
- 2- Large numbers of the users would like to control and customize the drones.
- 3- Administrators should be able to access reports on events which have occurred in the simulation.

The use case and class diagrams of the project are given in the figures below.

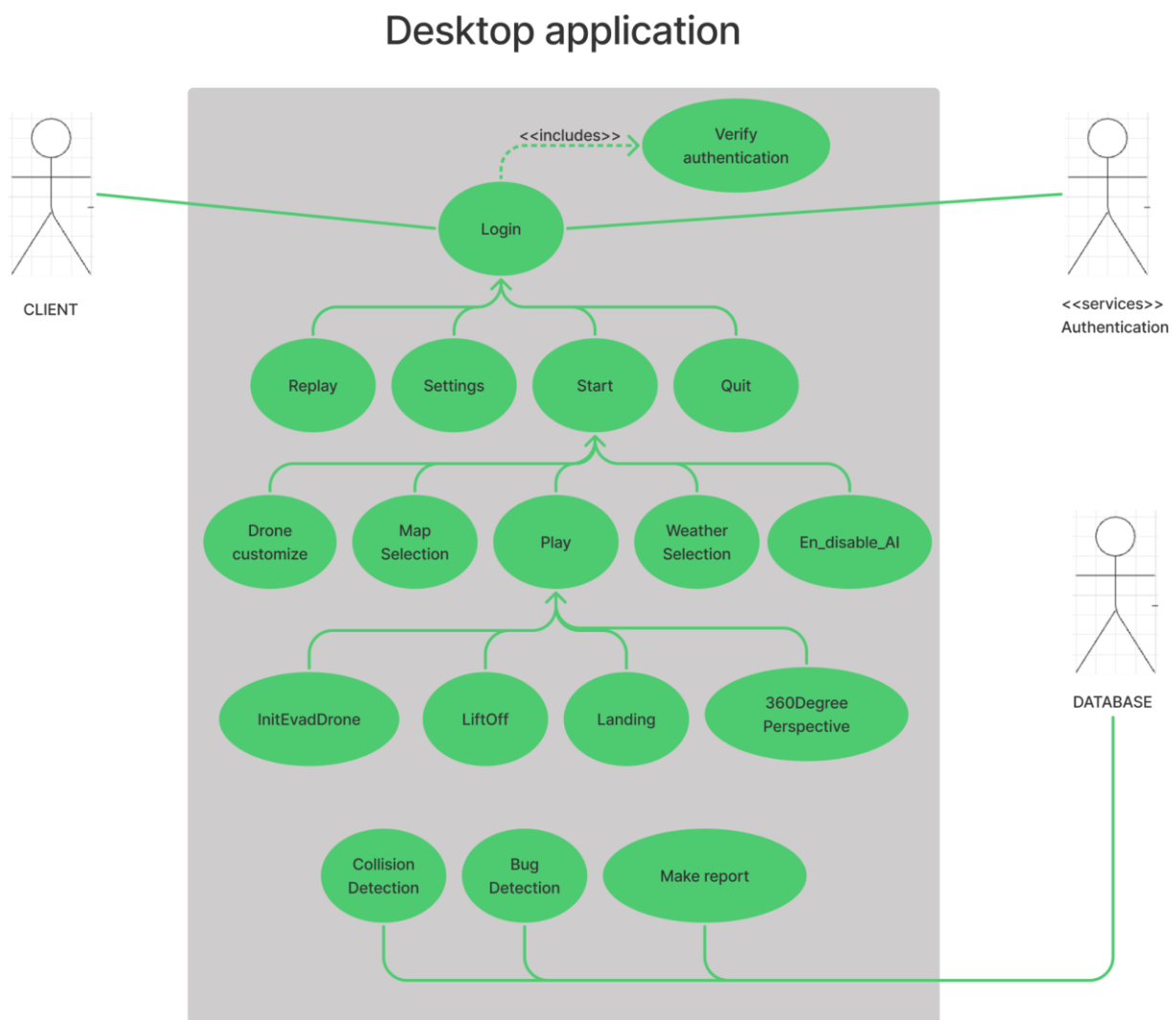


Figure 1: Use Case of Cyber Drone Desktop part.

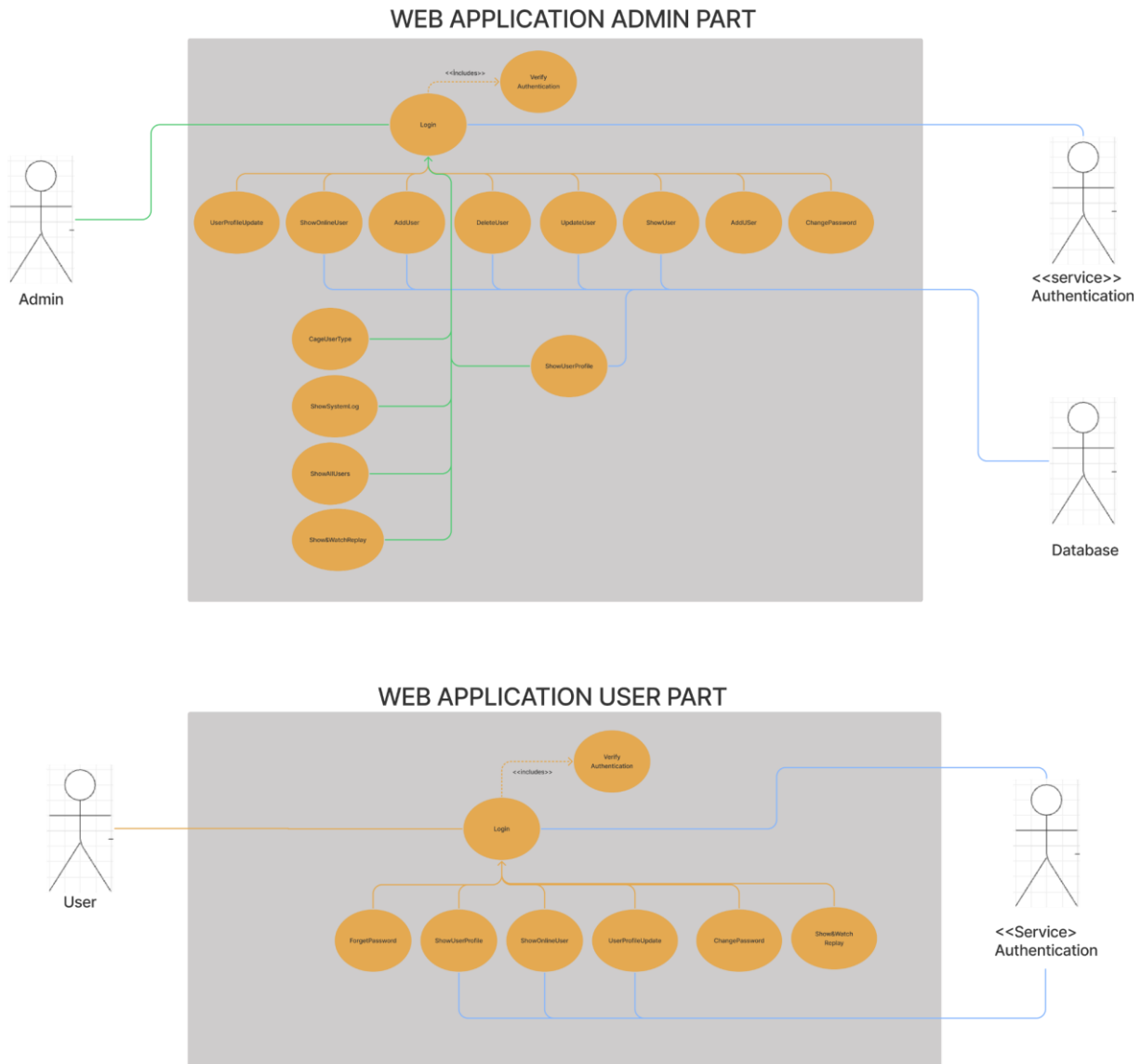


Figure 2:Use Case of Cyber Drone Web part

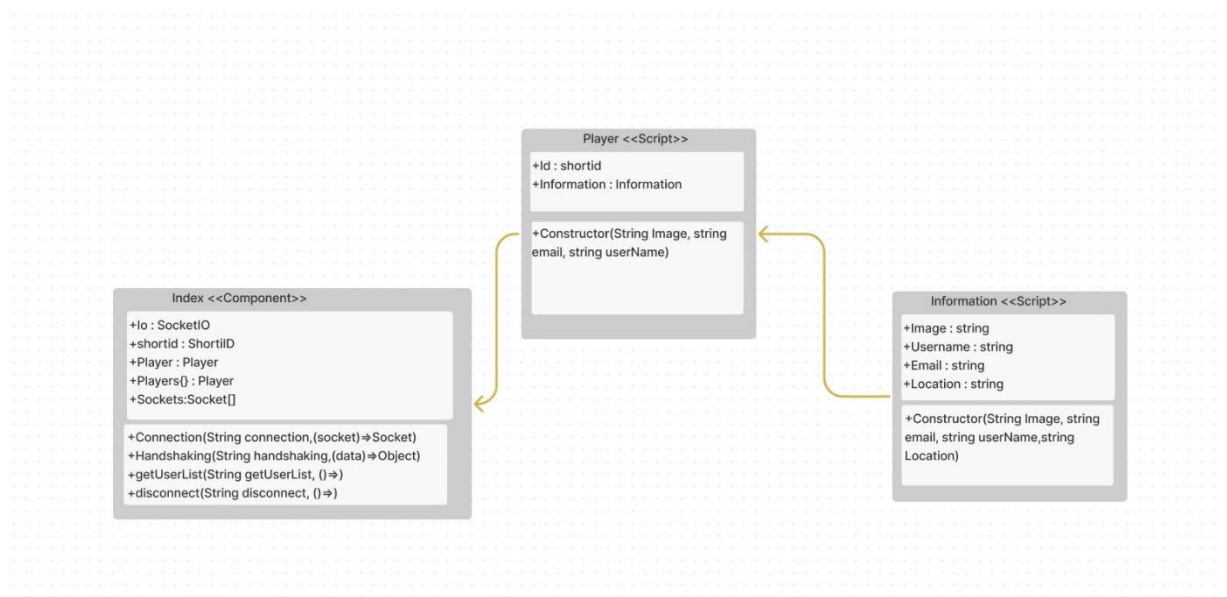


Figure 3: Class Diagram of Cyber Drone Online User Server Part.

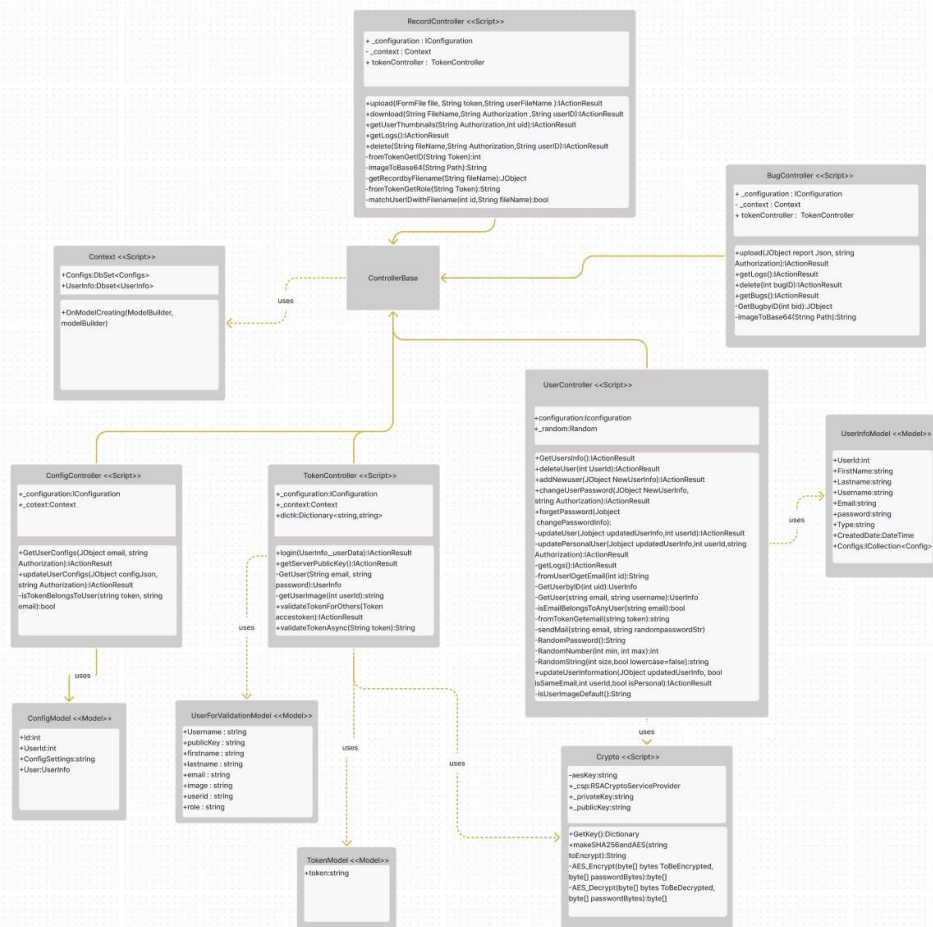


Figure 4: Class Diagram of Cyber Drone Web Server Part.

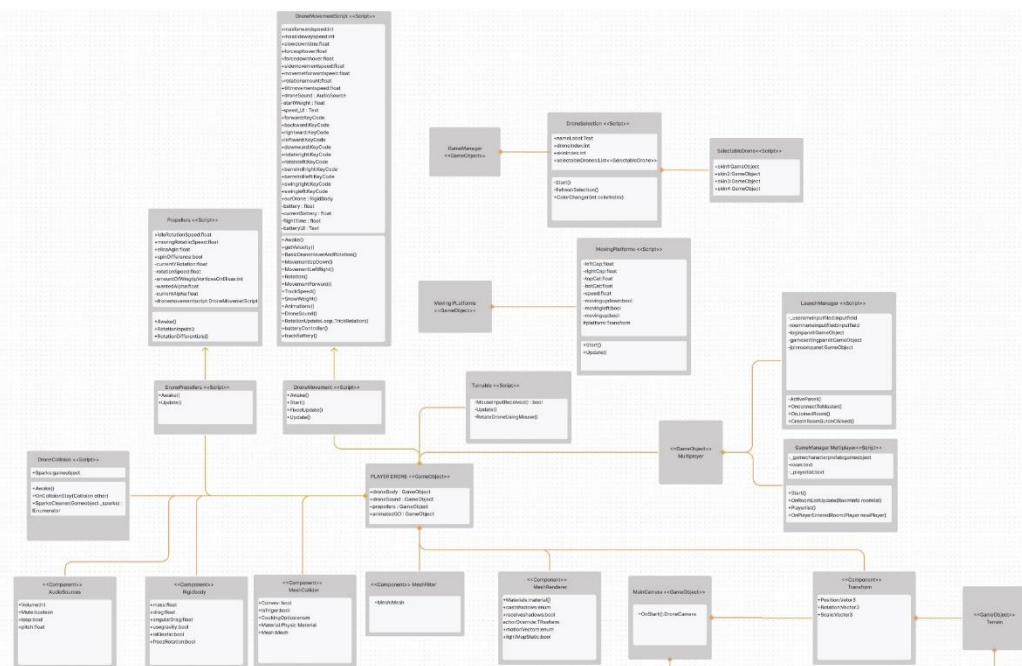


Figure 5: Cyber Drone Unity part-1.

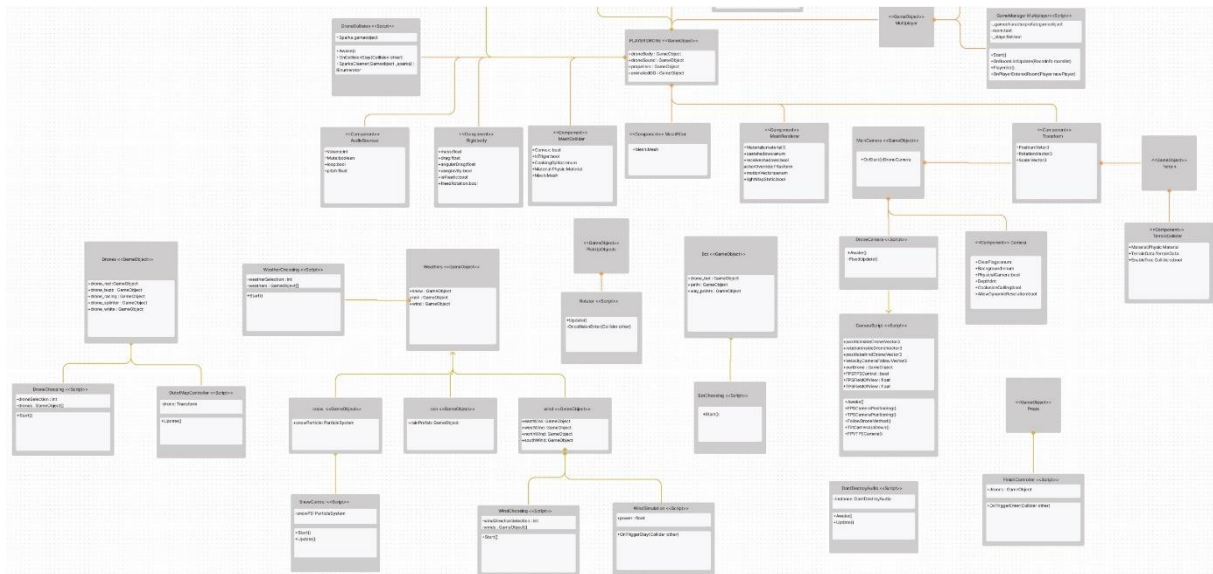


Figure 7: Class Diagram of Cyber Drone Desktop Canvas part.

4.	UploadRecord	Uploads the mp4 video to web server.
5.	DownloadRecord	Download the desired recorded from web server.
6.	Replay	Shows replay files upload & Watch your replay on web site.
7.	Quit	Closes the desktop application
8.	Play	Starts the simulation
9.	DroneCustomize	Customize the drone
10.	MapSelection	Changes the maps
11.	WeatherSelection	Changes the conditions of simulation
12.	En_Disable_AI	Enables or disables AI assistance
13.	LiftOff	Detects the liftoff drone
14.	Landing	Detects the landing drone
15.	360DegreePerspective	Changes Camera and movement of drone
16.	InitEnvandDrone	Loads or Reloads the environment and drones
17.	CollisionDetection	Detects the collision of drone
18.	BugDetection	Report the bug of application
19.	MakeReport	Collect the bugs
20.	AddUser	Insert a new user to application
21.	UpdateUser	Updates user's informations
22.	ShowOnlineUsers	Shows online users as web or simulation
23.	ShowAllUsers	Shows all users to developer
24.	DeleteUser	Deletes user from application
25.	ChangeUserType	Changes the user type & Give Permission
26.	ShowsSystemLog	Shows the log of system
27.	ShowUserProfile	Shows the user's profile
28.	UserProfileUpdate	User can update profile

29.	Login	Responsible for gathering username and password and calls verification
30.	Display Login Error	After unsuccessful login, shows error message
31.	Verify Username and Password	Controls the credentials of username and password
32.	ChangePassword	User can change password
33.	ForgetPassword	New generated password is sent to the user via email.
34.	Save Auth Token	Saves the authentication token for further logins
35.	Navigate Page	After a successful process, it sends the user related page.
36.	Read Related Information from database	Reads the related username informations and sends to verification process
37.	StartMatch	User starts te multiplayer game as a match
38.	CreateandJoinRoom	Create or join the room for a multiplayer game.
39.	CreateNewRoom	User creates a new room for a multiplayer game.
40.	JoinRandomRoom	User joins a random room for a multiplayer game.
41.	LeaveRoom	The user leaves the room where he/she is located.
42.	PhotonConnectionUsing	Establishes the connection between Photon server and the application.
43.	Connection	Detects the new socket which is the newcomer to the server. Creates a new instance of Player.
44.	Handshaking	Gets the newcomer player's data to attach information with this socket and adds the player in socket and player dictionary.

45.	GetUserList	Returns all player's information that is in the server.
46.	Disconnect	Detects closing connection with socket and pops the disconnected player in socket and players dictionaries. Then it triggers disconnected for sending the player information who is disconnected to other players.
46	GetServersPublicKey&EncryptUserInformationWithRSA	Clients get Server's public key and encrypted the login information with this key by using RSA.

Table 1: Requirement list of Cyber Drone Simulation Project.

3. Actors and Use Case Diagram

3.1. Actors in Cyber Drone

The actors in this project are given in the **table 2** below.

Actor	Description
Client	Member that who is allowed to use simulation application
Admin	Admin who is assigned by a company to manage their clients. With reaching web applications, admin is able to manage their companies' clients.
Developer	Developer is responsible for the system. Developers can assign an admin to manage their company clients.
Database	Database interacts with web apps and desktop apps to store all related information.

Table 2: Actor table.

3.2. Cyber Drone Use Case Diagram Groups

The initial architecture of the system is based on the packages into which the uses cases are grouped. These use cases have been grouped into seven subsystem packages.

Web Application:

- Authentication
- User Management
- Data Management

Desktop Application:

- Authentication
- Main Page Management
- Gameplay
- Simulation Main Control Simulation Report

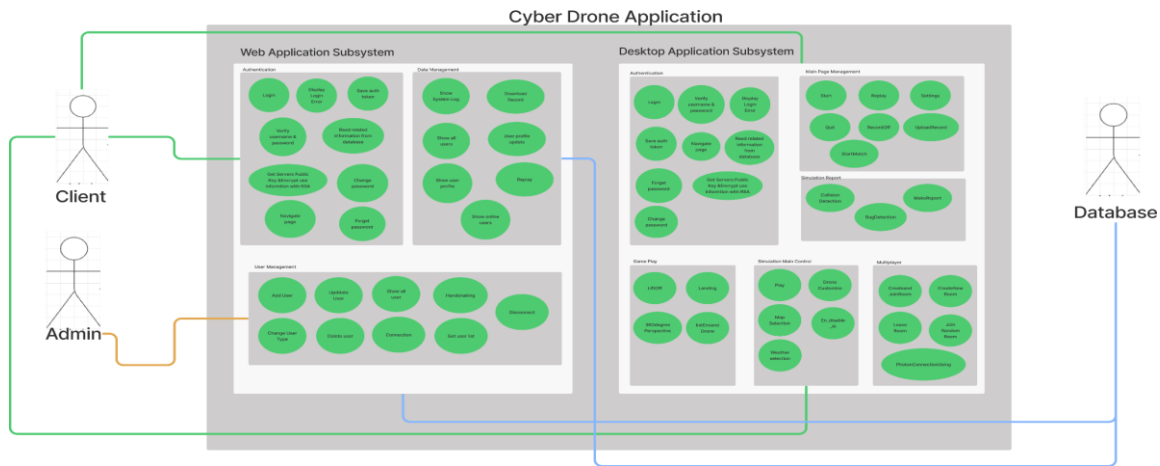


Figure 9: Cyber Drone Packages UML

3.3. Cyber Drone Use Case Diagrams

3.3.1. Web Application Subsystem Use Case Diagrams

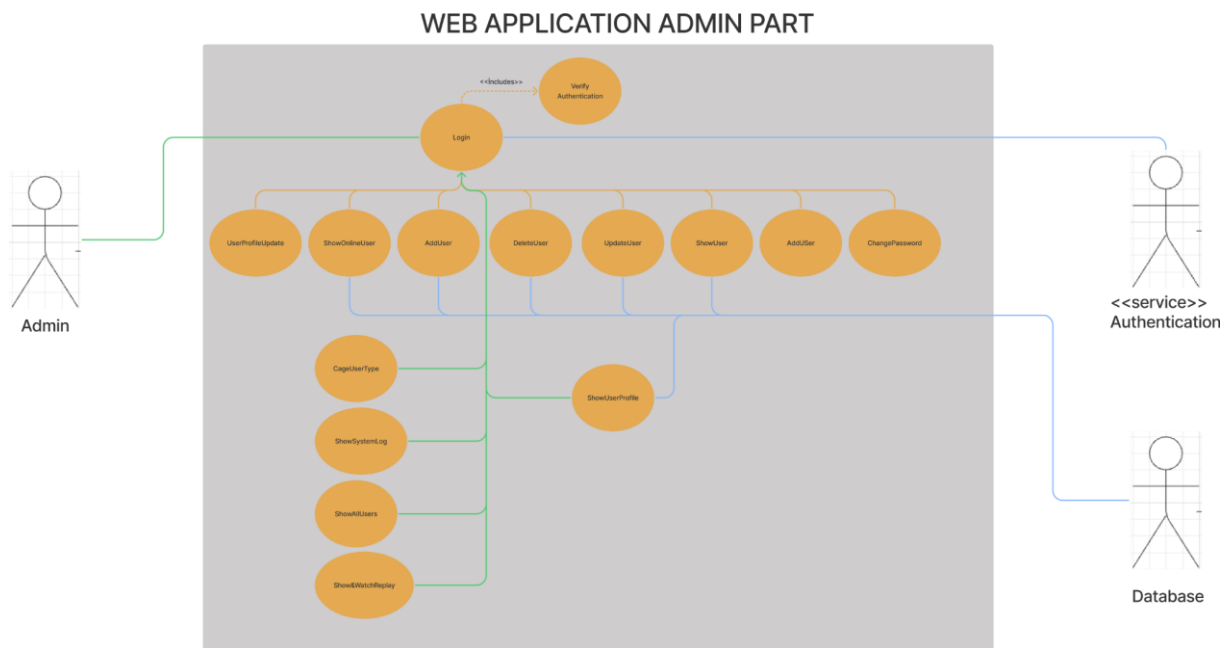


Figure 10: Use Case Diagram of Cyber Drone Web part.

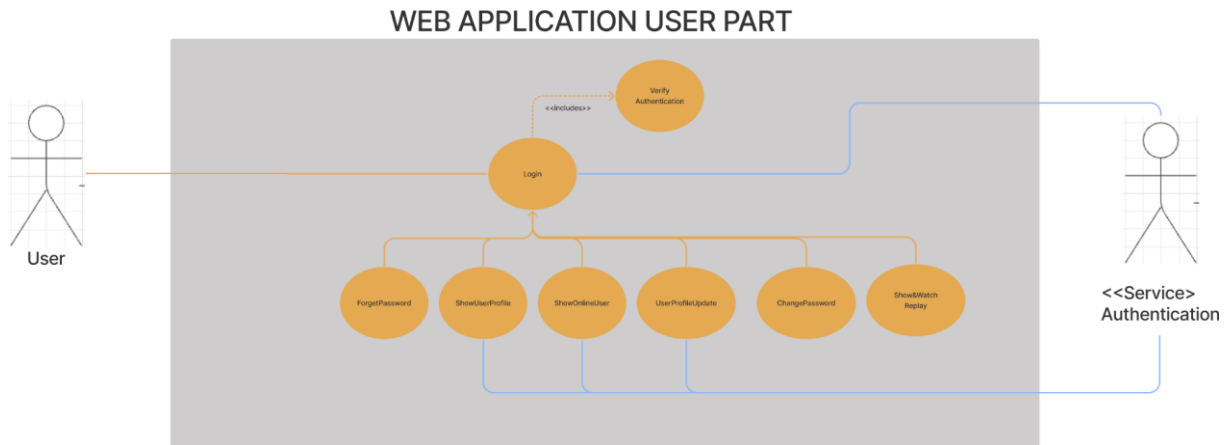


Figure 11:Use Case Diagram of Cyber Drone Web application User part.

The web application part is the part where the users add, remove and similar privileges are regulated. Changes are made to the system in line with the audited information. In order for these changes to be applied, admin or developers must log into the web application. The login process interacts with the authentication package and checks whether the user information is correct. In line with this control, they are allowed to enter the system. After entering the system, the admin is only allowed to access the information of his/her own users, while the developers have the authority to access the data and system records of all users. Admins have permissions such as adding, removing, deleting, and viewing users just as **figure 9**. In this way, the assignment of who can access the simulation environment is made. In addition, admins can access users' records. During all these operations, there is an interaction with the database. [1] In addition users have permission such as forget password, show online user in application, show own profile and edit profile, watch their replay. All interactions applicable to the user are shown in **figure 10**.

3.3.2. Desktop Application Subsystem Use Case Diagrams

The desktop application is the heart of the system in shown figure 1. Here, users whose information has been approved can log into the system. After the login, a menu will appear in front of the user, where he/she can start the simulation, change the system settings, get information about the simulation, watch the records of his/her playthroughs and exit the system. The user who wants to continue with the simulation is directed to the game screen from the start menu. Here, the user can select the desired map and weather conditions. Also, the user may select and customize any drone he/she desires. Then the user can start the simulation by clicking the play option. During the simulation, the user can take off the drone, land the drone and control the drone in 360 degrees. When a collision occurs during drone use, the system detects the collision and transmits it to the reporting system. In addition, bugs encountered by the user in the simulation environment are also caught by the system and transmitted to the reporting system. This reporting system transfers the recorded reports to the database. [1]

4. SYSTEM USE CASES

4.1. DESKTOP APPLICATION Systems Use Case Descriptions

4.1.1. AUTHENTICATION Use Case Descriptions

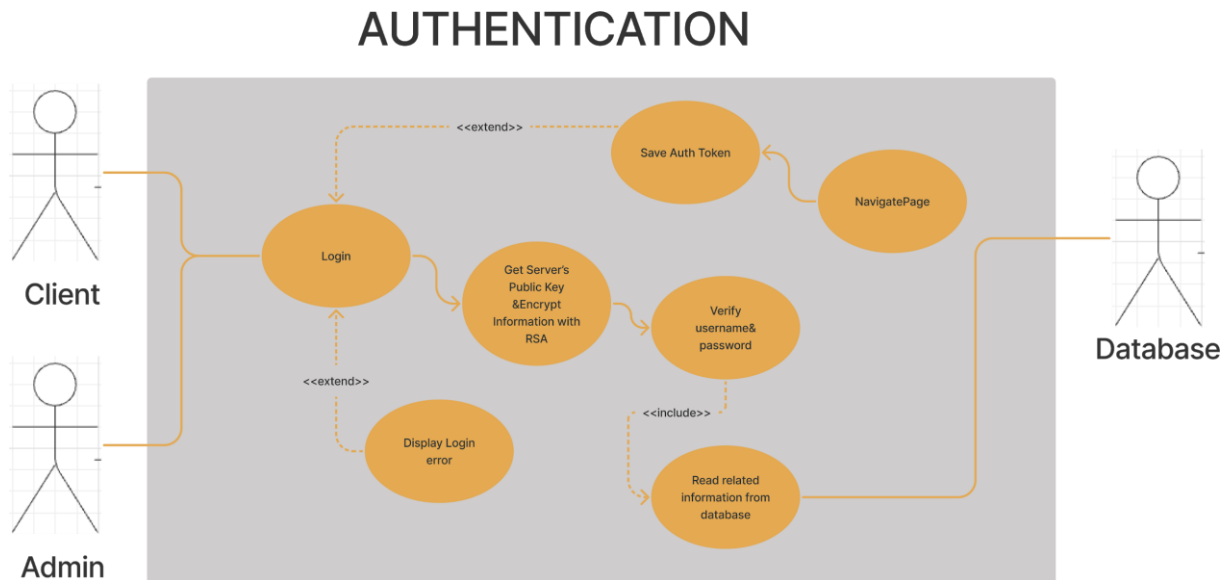


Figure 8: Authentication UML diagram.

4.1.1.1. Use Case 1 Login Description

When the user wants to log into the desktop application, this section interacts with the authentication service and receives the username and password information that the user entered on the login screen and transmits it to the authentication service. This service connects with the database and checks the correctness of the username and password. If the entries are valid as a result of this check, this case receives a positive feedback from authentication service and the user is transferred to the main page so that the user can take action. If the user's entries are not valid, an error message is displayed.

4.1.1.2. Use Case 2 Display Login Error Description

Login error description is sent to the client after a failure login attempt. Then the client's system shows the error message that comes from the server in a popup component.

4.1.1.3. Use Case 3 Save Auth Token Description

After successful login operation, client system stores authentication token, email, lastLogin, etc... It stores these values that mentioned before, in a file called config in xml form.

4.1.1.4. Use Case 4 Verify Username & Password Description

Login information is encrypted by `getServerPublicKey&EncryptInformationWithRSA` use case. Login information is sent to the web server by using api service. This information is decrypted by the server and checks the credentials of the user by using `readRelatedInformationFromDatabase` use case. According to validity, the server sends a response message either positive or negative.

4.1.1.5. Use Case 5 Read Related Information from Database Description

After the server gets related information from the client by using an api service, it collects related information such as hashed and encrypted password from database.

4.1.1.6. Use Case 6 Get Server's Public Key and Encrypt Information with RSA Description

Before login information is sent to the server, it receives the server's public key for RSA encryption. After collecting this public key, when the client wants to login, it encrypts the client's login information with RSA.

4.1.1.7. Use Case 7 Navigate Page Description

After successfully operation, it sends the user to the related page.

4.1.2. GAMEPLAY Use Case Descriptions

4.1.2.1 Use Case 1 LitfOff Description

This case detects that the drone is taking off by the user. As a result of this detection, the system understands that the drone is in the air.

4.1.2.2 Use Case 2 Landing Description

After the drone has landed by the user, it detects the drone's landing. As a result of this detection, the system understands that the drone has landed on the ground.

4.1.2.3 Use Case 3 360DegreePerspective Description

The user can control the drone through this case and have a 360-degree view angle. In this way, the drone has 360-degree maneuverability. The user gets the chance to observe the drone in 360 degrees.

4.1.2.4 Use Case 4 InitEnvandDrone Description

Through this case, the selected external factors, the selected map and the selected drone are prepared to start the simulation. After this process, these entries are loaded and the simulation is started.

4.1.3. MAIN PAGE MANAGEMENT Use Case Descriptions

4.1.3.1 Use Case 1 Quit Description

This case is responsible for the user to exit the simulation. In this way, the user can close the simulation application.

4.1.3.2 Use Case 2 Replay Description

This case is responsible for the part where the user can see the records of the simulations performed. In this way, the user can rewatch the simulation records.

4.1.3.3 Use Case 3 Settings Description

User's settings are received from the web server and the preferences of settings applied to simulation then these settings are saved to a file called settings.json. The user can change all the settings belonging to the application with the help of this case. All changes are sensed, and updated preferences are sent to the web server for saving.

4.1.3.4 Use Case 4 Start Description

The user is transferred to the main control screen, where the user can change the data of the simulation and start the simulation.

4.1.3.5 Use Case 5 RecordOnOff Description

Users can record your simulation on time. After recording, users are able to upload automatically upload their records that saved and see their records in file called records.

4.1.3.6 Use Case 6 UploadRecord Description

Users access their recorded file called records. So they can upload their simulation records to web system and save them easily.

4.1.3.7 Use Case 7 StartMatch Description

It is used by the user who has decided to play the multiplayer game. When this method runs, the user is redirected to the lobby. Then it realizes the simulation as a multiplayer by being included in a room.

4.1.4. SIMULATION MAIN CONTROL Use Case Descriptions

4.1.4.1 Use Case 1 Play Description

This section is responsible for starting the simulation environment by taking the information of external factors, drone and map preferred by the user. Through this case, the user is transferred to the simulation environment.

4.1.4.2 Use Case 2 DroneCustomize Description

It is the part where the drones in the system are selected by the user for use in the simulation environment. In addition, the user can modify the drone of her/his choice through this case.

4.1.4.3 Use Case 3 MapSelection Description

The maps available in the system are selected by the user through this case to be used in the simulation environment.

4.1.4.4 Use Case 4 WeatherSelection Description

External factors (weather conditions) that we prefer to occur in the simulation environment are selected through this case.

4.1.4.5 Use Case 5 En_Disable_AI Description

It is the section where the user can activate and deactivate the artificial intelligence, which acts as an assistant in order to use the drone more efficiently.

4.1.5. SIMULATION REPORT Use Case Descriptions

4.1.5.1 Use Case 1 CollisionDetection Description

It is the section where the collision occurred during the simulation is detected, recorded and transferred to the reporting system for reporting.

4.1.5.2 Use Case 2 BugDetection Description

It is the section where errors that occur during the use of the system are detected, recorded and transferred to the reporting system for reporting.

4.1.5.3 Use Case 3 MakeReport Description

It reports the information received from the relevant units and sends it to a database on a server to be recorded.

4.1.6. MULTIPLAYER Use Case Descriptions

4.1.6.1 Use Case 1 CreateandJoinRoom Description

When users join a multiplayer game, it is called when entering an existing room or creating a new room.

4.1.6.2 Use Case 2 CreateNewRoom Description

This method calls when the user or system wants the new room. This method uses the room options information which are the number of players, room name, determined by user or system randomly.

4.1.6.3 Use Case 3 LeaveRoom Description

It calls when the user decides to leave the room. After calling this method, user backs to the main menu and leaves the room.

4.1.6.4 Use Case 4 JoinRandomRoom Description

If the user does not join a specific room, he or she joins any existing room through this method.

4.1.6.5 Use Case 5 PhotonConnectionUsing Description

This method provides the internet and Photon network connection by Photon.If this method does not work, the driver cannot be activated.

4.2. WEB APPLICATION Use Case Descriptions

4.2.1. AUTHENTICATION Use Case Descriptions

4.2.1.1. Use Case 1 Login Description

When the user wants to log into the web application, this section interacts with the authentication service and receives the username and password information that the

user entered on the login screen and transmits it to the authentication service. This service connects with the database and checks the correctness of the username and password. If the entries are valid as a result of this check, this case receives a positive feedback from the authentication service and the user is transferred to the main page so that the user can take action. If the user's entries are not valid, an error message is displayed.

4.2.1.2. Use Case 2 Display Login Error Description

Login error description is sent to the client after a failure login attempt. Then the client's system shows the error message that comes from the server in a popup component.

4.2.1.3. Use Case 3 Save Auth Token Description

After successful login operation, client system stores authentication token, username, role, etc. It stores these values that mentioned before, in local storage in a web client and a store state by using Vuex.

4.2.1.4. Use Case 4 Verify Username & Password Description

Login information is encrypted by `getServerPublicKey&EncryptInformationWithRSA` use case. Login information is sent to the web server by using api service. This information is decrypted by the server and checks the credentials of the user by using `readRelatedInformationFromDatabase` use case. According to validity, the server sends a response message either positive or negative.

4.2.1.5. Use Case 5 Read Related Information From Database Description

After the server gets related information from the client by using an api service, it collects related information such as hashed and encrypted password from database.

4.2.1.6. Use Case 6 Get Server's Public Key and Encrypt Information with RSA Description

Before login information is sent to the server, it receives the server's public key for RSA encryption. After collecting this public key, when the client wants to login, it encrypts the client's login information with RSA.

4.2.1.7. Use Case 7 Change Password Description

When a client attempts to change a password, it collects old and new passwords twice. Firstly, it compares two old passwords whether they are the same or not. If the result is true, it sends the new and old credentials to the server by using RSA encryption and changes password by using sha256 hash algorithm and AES encryption.

4.2.1.8. Use Case 8 Navigate Page Description

After successfully operation, it sends the user to the related page.

4.2.1.9. Use Case 9 Forget Password Description

It gets the email and username from clients and sends them to the web server, after that, the server checks the validity of these information. At the end the new generated password which comes from the web server, is sent to the user's email address.

4.2.2. DATA MANAGEMENT Use Case Description

4.2.2.1. Use Case 1 ShowSystemLog Description

This section is only accessible to admins. As a result of access, all records of the system are transferred to an interface. Through this interface, they can observe all the logs in the system.

4.2.2.2 Use Case 2 Download Record Description

All records information's such as thumbnails and name are transferred to client interface from the web server by using api service. Desired records are downloadable as mp4 format. That methodology is used by clients and admins.

4.2.2.3 Use Case 3 Show All Users Description

All user's information's such as name, last name, username, creation date are shown by interface like table. This information comes from a related api service. In that table, all users are selectable for deletion, showing their details, and updating. That methodology is used by admins.

4.2.2.4 Use Case 4 User Profile Update Description

User information is stored in the Vuex State Management System after the login process. When a client or admin accesses the profile page, stored information that is related, is received from that system and shown in interfaces such as form. Users are able to change their credentials by using this case. All new information is sent to the web server and new credentials are checked. According to validity systems send feedback to clients whether conditions are satisfied or not.

4.2.2.5 Use Case 5 Replay Description

All records information's such as thumbnails and name are transferred to client interface from the web server by using api service.

4.2.2.6 Use Case 6 Show User Profile Description

It is the part where admin can access her/his defined system via an interface and view the information of the defined users. Then admins are able to update the desired user's information's. All new information is sent to the web server and new credentials are checked. According to validity systems send feedback to clients whether conditions are satisfied or not.

4.2.2.7 Use Case 7 Show Online User Description

When a client login in server, sockets are triggered and this user information are sent to other clients that are already in system, then list of online users are updated. according to newcomer user information. Generally, this case shows online users that are in the web or simulation system to us.

4.2.3. USER MANAGEMENT Use Case Description

4.2.3.1 Use Case 1 AddUser Description

It is the part where admin can access her/his defined system via an interface and add users. In addition, in this section, also developers can access the system via an interface and add users and admins.

4.2.3.2 Use Case 2 DeleteUser Description

It is the part where the admin can reach his/her defined system via an interface and delete his / her defined users. In addition, in this section, also developers can access the system via an interface and delete all users and admins.

4.2.3.3 Use Case 3 UpdateUser Description

It is the section where admin can access her/his defined system via an interface and update the information of defined users. In addition, in this section, also developers can access the system via an interface and update the information of all users and admins.

4.2.3.4 Use Case 4 ShowAllUser Description

In this section, developers can access the system via an interface and view the information of all users and admins.

4.2.3.5 Use Case 5 ChangeUserType Description

It is the part where the developers manage the authorizations of all users and admins through an interface.

4.2.3.6 Use Case 6 Connection Description

When a new client logs in on a web or simulation system, it creates a new socket, then the user connection server senses a new socket is connected to our system.

4.2.3.7 Use Case 7 Get User List Description

When Get User List is called, the entire user list is emitted on to the client calling it.

4.2.3.8 Use Case 8 Handshaking Description

When handshaking is called, it creates a new instance to user for collecting data such as their images, usernames, state (web or simulation) and client id (generates automatically in user connection server.). At the end it adds the newcomer user to the system and server sends the user list of servers to the newcomer user also newcomer user information is broadcast to other users.

4.2.3.9 Use Case 9 Disconnect Description

When the user that connected to the system is disconnected, the entire user list is broadcasted to other users in the system. Then the user that is disconnected are popped from list of users.

5. Non-Functional Requirements

The software will be developed in Unity or game engine language such as C#. The development and operation environment are any Windows. All client-server operations through the Internet will be through SSL to provide security. All the data which will be transferred to the database, is transferred to the server via encryption. Vice-versa is also valid. Necessary data will be stored in a database with its encrypted versions. Actions of all administrators regarding user/admin deletion and creation must be logged to a file and info about such actions must be emailed to the first administrator in the list of all administrators. Related system can support multiple users without crashing. [1]

References

[1] D. Team, "Cyber Drone RSD v1.0," İzmir, 2020.