
# ÇANKAYA UNIVERSITY COMPUTER ENGINEERING

# CENG 407

# BLASTSTRIKE

# SOFTWARE REQUIREMENT SPESIFICATION (SRS)

**GROUP 13**

**TEAM MEMBERS**

**Nadide Solmaz 201911056**

**Seyit Koyuncu 201911036**

**Zeynep Deniz Dönmez 202011012**

**Alperen Kaan Salt 201911052**

**Mehmet Emir Hocaoğlu 201911029**

Contents

# 1. INTRODUCTION

## 1.1 Purpose of This Document

The primary purpose of this document is to provide a clear and detailed understanding of the scope, objectives, and technical specifications of BlastStrike. It serves as a foundational reference for the development team, stakeholders, and any external entities involved in the creation and enhancement of BlastStrike. The scope of this SRS encompasses the entire development lifecycle of BlastStrike, from conceptualization to deployment. It details the core functionalities, system architecture, user interactions, and technical requirements essential for the successful implementation of the gaming application.

## 1.2 Scope of the Project

Imagine an exhilarating real-world mobile game that combines the thrill of laser tag or paintball with cutting-edge technology. This multiplayer FPS (First Person Shooter) mobile game takes the gaming experience to a new level, integrating elements of augmented reality seamlessly into the real world.

The game leverages FPS elements, creating a virtual environment without the need for 3D player models. Instead, real-time body segmentation in the scene allows for accurate hit detection. Tensorflow-lite, PyTorch, or Yolo will be utilized for segmentation, and the application structure fully supports Tensorflow-lite. The game can be developed using Flutter.

Players will experience the immersive feel of the game with a 3D model gun integrated into the scene, complete with animations. Hit players will have their screen frames turn red, providing immediate visual feedback and enhancing the overall stimulation. A shooting effect on the screen, coupled with firing sounds, creates a realistic and engaging environment. Given the real ambient sounds, there's no need for additional sound production.

Player positioning is determined by GPS location, phone perspective, and image distance estimation. Damage calculation is segmented based on head, arm/leg, and body hits. Points are awarded to the shooter, and game dynamics evolve as players engage in battles.

For a versatile gaming experience, the game accepts fire requests through Bluetooth controllers, transforming smartphones into weapon controllers. Bluetooth-enabled weapon controllers enhance the overall gaming interface.

A player initiates the game, and others join, forming teams similar to dynamics seen in games like Kahoot. This setup ensures an inclusive and engaging gaming experience.

An AMQP-based message queue structure facilitates real-time information sharing during gameplay. Alternative structures may be explored and decided upon during development for optimal performance.

While the described features form the core of the game, future updates may include additional elements such as purchasing weapon skin options and introducing different weapon models (e.g., machine gun, shotgun, flamethrower). These features promise to add diversity and excitement, altering in-game dynamics.

In essence, this game blends the real and virtual worlds, providing players with a unique and immersive gaming experience that goes beyond traditional boundaries.

# 1.3 Glossary

| TERM | **DEFINITION** |
|---|---|
| Mobile Game | A mobile game, or smartphone game, is a video game that is typically played on a mobile phone. |
| FPS(First-Person Shooter) | A type of video game in which the player views the action through the eyes of a character and has to attack enemies |
| Android | A type of operating system, designed for mobile devices, which controls the way the device works and runs apps |
| IOS | IOS is a mobile operating system developed by Apple |
| GPS | A system by which signals are sent from satellites to a special device, used to show the position of a person or thing on the surface of the earth very accurately |
| Geolocation | The process or technique of finding the exact location of a person or device using the internet |
| Flutter | Flutter is an open-source UI software development kit created by Google. |
| NodeJS | Node.js is a cross-platform, open-source JavaScript runtime environment that can run on Windows, Linux, Unix, macOS, and more. |
| Database | An organized set of data that is stored in a computer and can be looked at and used in various ways |
| SRS | A software requirements specification is a description of a software system to be developed. |
| User Interface(UI) | The way a computer gives information to a user or receives instructions from a user |

| | |
|---|---|
| AMQP | The Advanced Message Queuing Protocol is an open standard application layer protocol for message-oriented middleware. |
| Tensorflow Lite | TensorFlow Lite is a set of tools that enables on-device machine learning by helping developers run their models on mobile, embedded, and edge devices. |

# 2. OVERALL DESCRIPTION

## 2.1 Product Perspective

BlastStrike emerges as a cutting-edge mobile gaming phenomenon, seamlessly fusing the excitement of laser tag and paintball into a captivating real-world experience. This multiplayer extravaganza is meticulously crafted to deliver an unparalleled blend of physical and virtual interactions, promising an adrenaline-pumping adventure for players of all levels.

## 2.2 User Characteristics

2.2.1. **Demographics:**

**Age Group:**

The target age group for *BlastStrike* is 18-35, catering to young adults and gaming enthusiasts.

**Geographical Location:**

Primarily aimed at users in North America, Europe, and Asia where online multiplayer gaming is popular.

2.2.2. **Technical Proficiency:**

**Technical Background:**

Users are expected to have a moderate to beginner level of gaming experience, comfortable with FPS elements.

**Device Familiarity:**

Assumes familiarity with mobile devices and gaming platforms like iOS and Android.

2.2.3. **Gaming Preferences:**

**Genre Preferences:**

Targeting users who enjoy action-packed FPS and multiplayer gaming experiences.

**Multiplayer Experience:**

Designed for users who prefer intense and competitive multiplayer battles.

2.2.4. **Accessibility Requirements:**

**Device Accessibility:**

Compatible with a range of modern smartphones to ensure accessibility for a broad user base.

**Language Preferences:**

Initially available in English, with plans for localization based on user demand.

2.2.5. **Gaming Behavior:**

**Frequency of Play:**

Intended for regular gameplay, with features that cater to both short and extended gaming sessions.

**Preferred Session Length:**

Designed for engaging sessions of 15-30 minutes, suitable for on-the-go gaming.

# 3. REQUIREMENTS SPECIFICATION

## 3.1 External Interface Requirements

### 3.1.1 User Interface

The user interface of our application will be clear and user-friendly. This application can work on both IOS and Android based mobile devices. Users who will play the game firstly select or create game room from main menu.After user in a lobby users can chat with eac other from chatbox and set their status to ready.After every user is ready .Host can start the from start button. After game starts the main interface becames view of front camera and some UI elements related to game.Users who will play the game will see a small crossover on middle of the screen, shooting button ,reloading button and settings.After game ends users returns to lobby.

### 3.1.2 Hardware Interface

This application works on all mobile devices based on IOS and Android.Server and client will be communicate through TCP protocol on the register,login ,chat and lobby operations.AMQP will be used for in game communication between server and client.

### 3.1.3 Software Interface

In our mobile application, it is written with React native, which is used with React.js for the front end. HTML, CSS, JavaScript programming languages are mainly used in the development process of the application and node.js is used for the back end.

### 3.1.4 Communication Interface

IOS and Android based mobile devices must have an internet connection in order for the application to reach locations. Besides, users should allow system for accessing their camera.

## 3.2 Functional Requirements

### 3.2.1 System and Mobile Application Requirements

#### 3.2.1.1 Geolocation Integration:

The system should utilize the mobile device's GPS capabilities to determine the player's real-world location. Geolocation data should be accurate and updated in real-time during gameplay.

#### 3.2.1.2 Camera Integration:

The game should access the mobile device's camera to capture the real-world environment.The camera should provide a clear and responsive feed for an immersive gaming experience.

#### 3.2.1.3 Multiplayer Support:

The game may support multiplayer functionality, allowing players to compete against each other in real-time. Multiplayer features might include cooperative missions, competitive challenges, or team-based gameplay.

#### 3.2.1.4 Score Table:

The game should have track the score of each player individual and also each team to decide which team is win the game.

#### 3.2.1.5 User Registration:

Users can register by providing necessary information (name,email ,password).
The system validates input .Upon successful registration, a unique user ID is generated.Clear error messages are provided for registration failures.

#### 3.2.1.6 User Login:

Registered users can log in using their email address and their password.The system verifies the entered credentials against the stored user information.Successful login grants the user access to the application.In case of unsuccessful login attempts, the system provides appropriate error messages.

**3.2.1.7 Lobby Creation:**
The system shall allow logged-in users to create multiplayer game lobbies.Upon lobby creation, the system shall generate a unique ID for each lobby.

**3.2.1.8 Friend Management:**

The system displays the online/offline status of users friends.Notifications are sent for friend requests to the users.

**3.2.1.9 Join Lobbies:**
The system shall allow users to join existing multiplayer game lobbies via request and entering unique room id from join button on menu.

**3.2.1.10 Shoot Mechanism:**

Users can initiate a shooting action by clicking the "shoot" button during a match. The system shall detect if there is a player at the targeted area via using gps location of current players and distance between 2 user.

**3.2.1.11 Lobby Chat Feature:**

The system shall provide a chat feature in the lobby for users to communicate before the game starts.Users shall be able to send and receive messages in a general chat within the lobby.

**3.2.1.12 Team Selection in Lobby:**

The system shall include a team selection feature in the lobby, enabling players to choose their teams before the game begins.Users shall have the ability to select the team of their choice during the team selection phase.

**3.2.1.13 Team Selection in Lobby:**

The system shall include a team selection feature in the lobby, enabling players to choose their teams before the game begins.
Users shall have the ability to select the team of their choice during the team selection phase.

**3.2.1.14 Weapon Customization:**

The system shall provide a weapon customization feature, enabling users to personalize their weapons. Users shall have access to a menu or interface where they can customize their weapons.

#### 3.2.1.15 Game Quit Option:

The system shall provide a "Quit Game" option for users during an ongoing game.
Users shall have the ability to initiate the game quit process.

#### 3.2.1.16 Ammo Reload Functionality:

Users shall trigger the ammo reload by a specific input command or button The system shall include a functionality that allows users to reload their ammunition during gameplay.

#### 3.2.1.17 Respawn Timer:

Users can respawn after a specified duration following their death.

#### 3.2.1.18 Surrender Mechanism:

If a majority of team members vote to surrender, the opposing team may concede the match.

#### 3.2.1.19 Custom Game Settings:

The host can set and customize game parameters in the lobby game mode,player count and win conditions.

## 3.3 NON FUNCTİONAL REQUİREMENTS

### 3.3.1. Performance:

The game shall provide a smooth and responsive experience, with minimal latency between real-world actions and in-game responses.Loading times for game should be waitable to ensure quick entry into matches.

### 3.3.2. Scalability:

The game server infrastructure must be scalable to accommodate a growing number of players, ensuring a consistent experience even during peak usage times.

### 3.3.3. Reliability:

The game shall have robust error handling and recovery mechanisms to minimize the impact of unexpected errors or crashes.

### 3.3.4. Security:

Player data, including GPS coordinates, shall be transmitted and stored securely to protect user privacy.

The game shall implement anti-cheat measures to ensure fair play and discourage cheating or exploiting vulnerabilities.

**3.3.5. Compatibility:**

The game shall be compatible with a wide range of mobile devices, ensuring a consistent experience across different screen sizes and hardware specifications

**3.3.6. Network Connectivity:**

The game shall be optimized for various network conditions, including 3G, 4G, and Wi-Fi, to accommodate players with different connectivity options.

**3.3.7. Cross-Platform Compatibility:**

The game shall support cross-platform play between iOS and Android devices.

**3.3.8. User Experience (UX):**

The game shall have an intuitive and user-friendly interface, with controls optimized for mobile devices.

**3.3.9. Compliance:**

The game shall comply with relevant privacy regulations and guidelines governing the collection and use of location-based data.

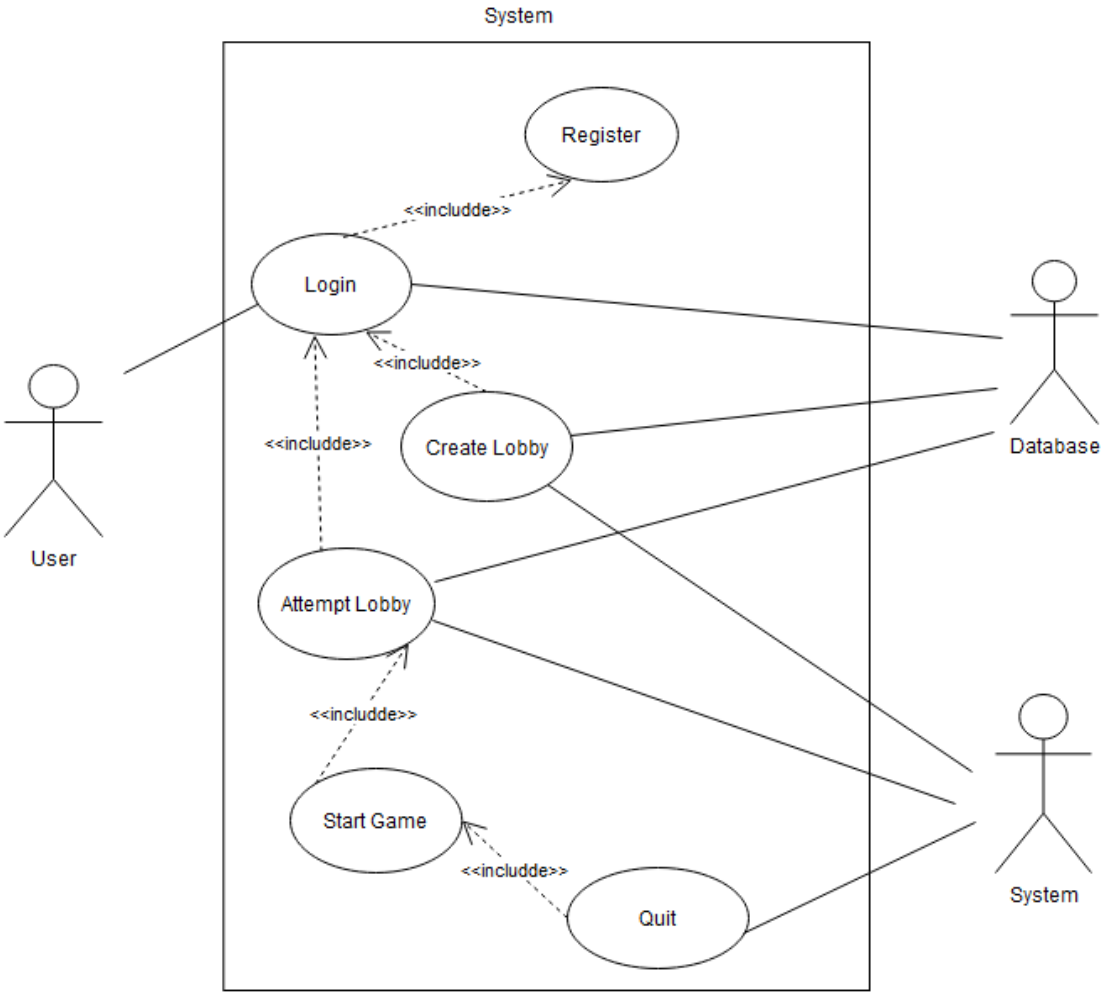
**3.3.10. Database Scalability:**

The system should be able to scale the database to accommodate a growing user base.

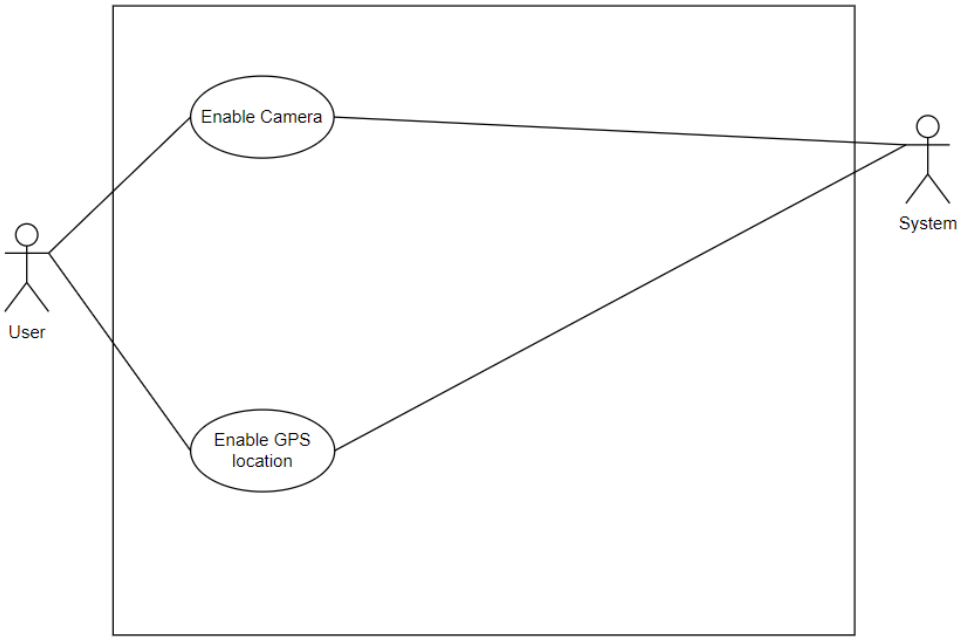
**3.3.11. Server Scalability:**

The game server infrastructure should scale horizontally to handle increasing traffic.

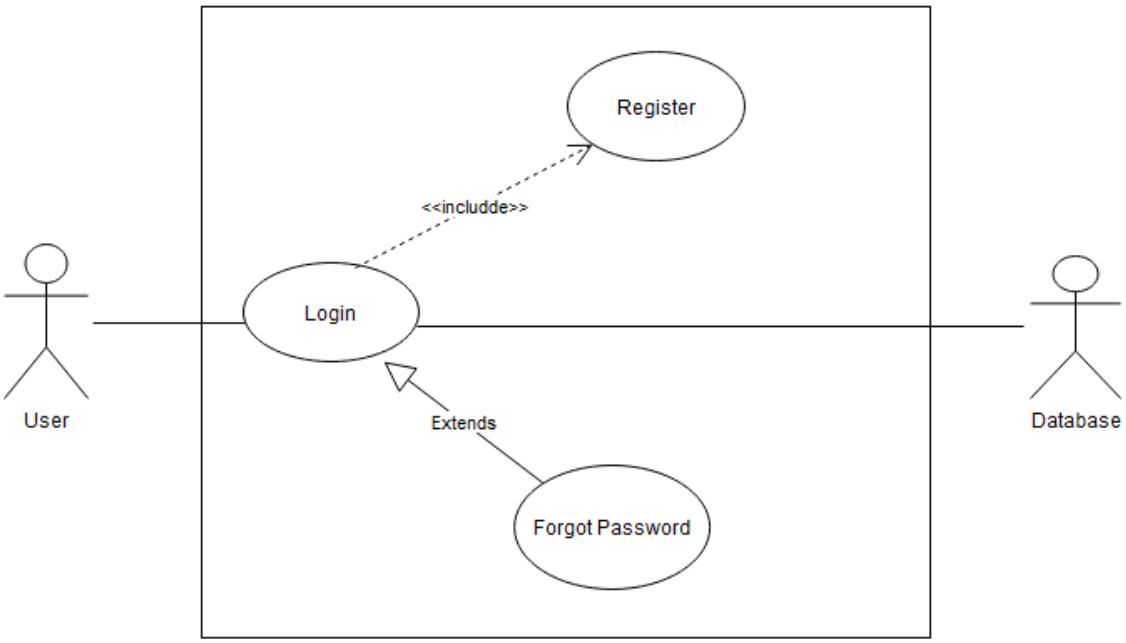
# 4. Use Cases


System
Register
<<includde>>
Login
<<includde>>
<<includde>>
Create Lobby
Attempt Lobby
<<includde>>
Start Game
<<includde>>
Quit
User
Database
System

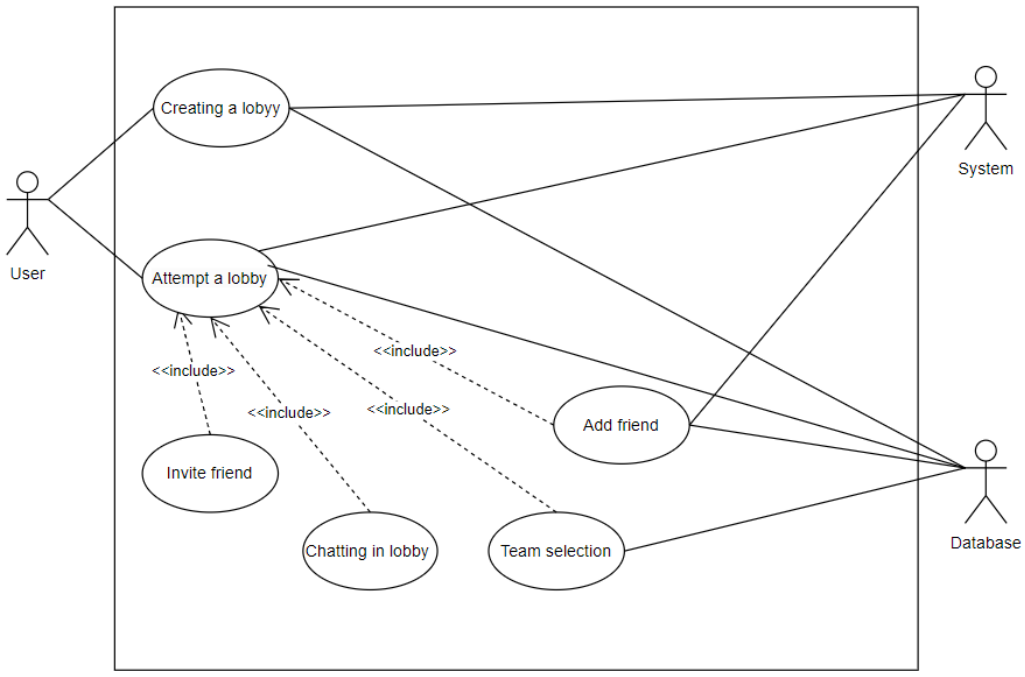
Enable Devices Use Case Diagram



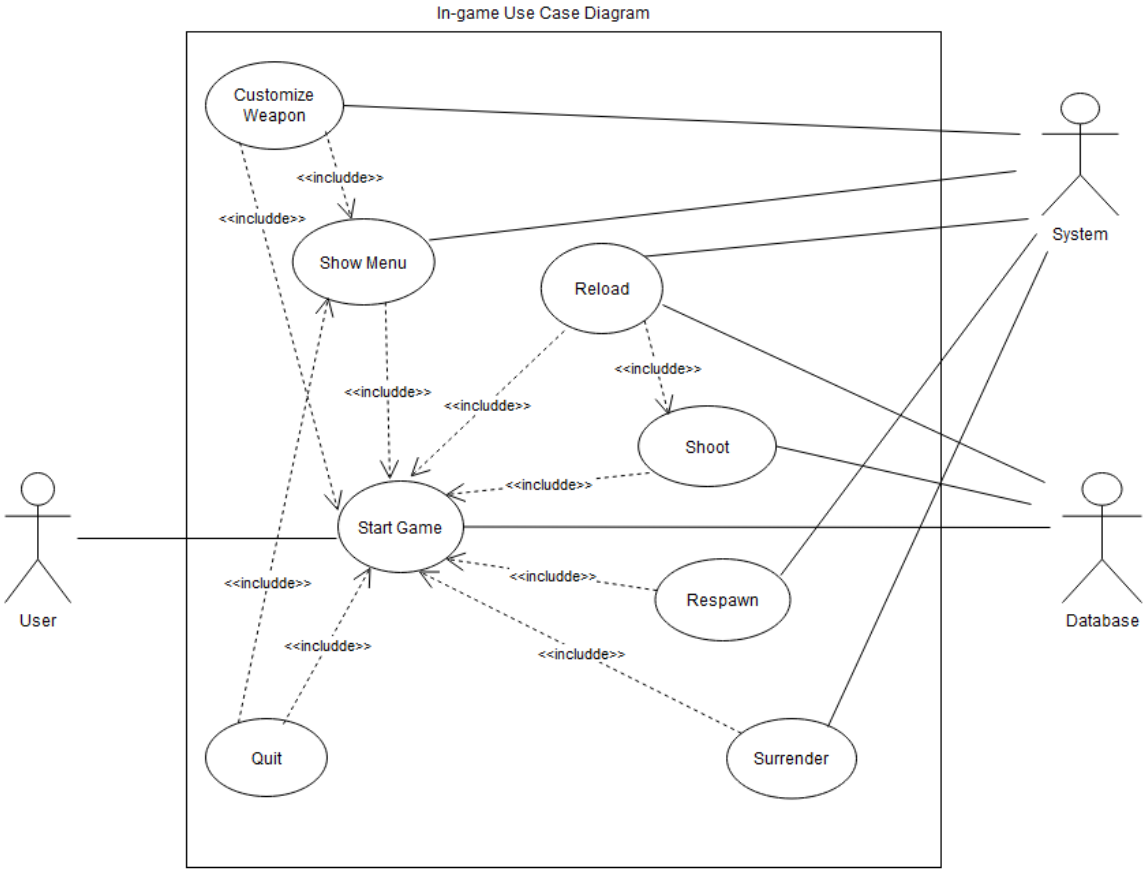

Login Use Case Diagram



Lobby Use Case Diagram




In-game Use Case Diagram



| | |
|---|---|
| Use case Number | UC-1 |
| Use case Name | Register |
| Actor | User, Database |
| Description | After downloading the mobile application to their phones, new users register to the application by providing the necessary information. If the registration is successful, users can log in to the system and start using the application. |
| Precondition | The user must download the application to their phone.<br>The user does not need to be registered with the application before.<br>User must have their own email address or phone number for the verification code. |
| Scenario | 1. The register button on the login page directs the user to the registration page.<br>2. The user enters personal information such as name, surname, username, e-mail, on the registration screen.<br>3. The user enters the verification code into the application.<br>4. The authenticated user logs into the application and informations saved to the DataBase. |
| Postcondition | The user can access the login page in the application. |
| Exceptions | 2.1-If the user enters invalid or incomplete information, the system prompts the user to renew the information with a warning.<br>2.2-When a registered user's e-mail or username is taken by other users before, the system prompts "There is a registered user for this e-mail or username". |
| Related Use Cases | - |

| | |
|---|---|
| Use case Number | UC-2 |
| Use case Name | Login |
| Actor | User, Database |
| Description | After the user registers to the application, s/he can log in to the application with her password using her e-mail address or phone number. |
| Precondition | User must be registered.<br>The user must enter the correct e-mail, and password. |
| Scenario | 1. The user enters the application<br>2. After the user enters the information for login, the system sends a verification code to the email for authentication.<br>3. The user enters the verification code into the application.<br>4. The authenticated user logs into the application If the informations matches the information in the database. |
| Postcondition | The user can access the main page of the app. |
| Exceptions | 2.1-The user who enters incorrect information while logging in is warned by the system.<br>2.2-System displays a error message.<br>3.1-If the user writes the wrong verification code, the user cannot create the account. |

| | |
|---|---|
| Use case Number | UC-3 |
| Use case Name | Forgot Password |
| Actor | User, Database |
| Description | When the user forgets the password s/he created while logging into the application, s/he can change it with the 'forgot password'. In order for the user to change his password, he must confirm what he wants with the e-mail or phone number s/he has given to the application. |
| Precondition | User should loginned. |
| Scenario | 1. The user presses the forgot password button.<br>2. The system asks the user for verification to make sure that the user made this reque<br>3. As a result of the verification, the user changes his/her password and Database update. |
| Postcondition | The user can access the login page in the application. |
| Exceptions | If the user cannot reach the e-mail address or phone number that he/she provided while registering, he/she can log in using the code the system gave the user during the first verification. |
| Related Use Cases | UC-2 |

| | |
|---|---|
| Use case Number | UC-4 |
| Use case Name | Enable GPS |
| Actor | User(Host), System |
| Description | The user must allow location access to use the app. |
| Precondition | The user needs to turn on the location services available on their phone. |
| Scenario | 1. The user opens the application.<br>2. The user turns on location services on his phone.<br>3. The user allows the application to access the location.<br>4. The user starts using the application. |
| Postcondition | The system can access user's location information to play game. |
| Exceptions | 1.2- The User can enable location service .<br>1.3-System displays a error message. |
| Related Use Cases | |

| | |
|---|---|
| Use case Number | UC-5 |
| Use case Name | Enable Camera |
| Actor | User, System |
| Description | The user must allow camera access to use the app. |
| Precondition | The user needs to turn on the camera services available on their phone. |
| Scenario | 1. The user logs into the application.<br>2. The user turns on camera services on his phone.<br>3. The user allows the application to access the camera.<br>4. The user starts using the application. |
| Postcondition | The system can access user's camera information to play game. |
| Exceptions | The user may try to log in and use the application without turning on the camera services, this should send error messages |
| Related Use Cases | UC-1,UC-2 |

| | |
|---|---|
| Use case Number | UC-6 |
| Use case Name | Creating a Lobby |
| Actor | User(Host), System, Database |
| Description | After loginned to the system, user should crate a new lobby for starting multiplayer game as a host. |
| Precondition | The User should loginned to the system .<br>The User should enable gps location.<br>The User should enable camera. |
| Scenario | 1. The User clicks the “Create Lobby” button.<br>2. System directs user to the game lobby and indicates the user as “Host”.<br>3. System gives a unique ID to the lobby.<br>4. System sends unique ID to the database.<br>5. Database keeps the ID. |
| Postcondition | The User can invite friends and starts a game.<br>The User can set the game rules.<br>The User can make team selection.<br>The User can chating with other players.<br>The user can starting the game.<br>The user can quit from the lobby. |
| Exceptions | 1.1- The User can not connect the system .<br>1.2-System displays a error message. |
| Related Use Cases | UC-1,UC-2,UC-4,UC-5,UC-8, UC-20, |

| | |
|---|---|
| Use case Number | UC-7 |
| Use case Name | Adding a Friend |
| Actor | User, System, Database |
| Description | Users should able to add new friends and see their status as Offline or Online. |
| Precondition | User should loginned. |
| Scenario | 1. User clicks add friends button.<br>2. System opens a new pop-up screen which includes a row for user ID.<br>3. User enters the ID of the user that wanted to add as a friend and clicks a add friend button.<br>4. System sends a notification to the other user which have entered ID.<br>5. Invited User clicks “Yes” button for adding a friend request.<br>6. System sends Users ID’s to the database.<br>7. Database keep records of the ID’s. |
| Postcondition | User can invite friends from their friend list on the lobby. |
| Exceptions | 3.1-User enters invalid ID.<br>3.2-System sends error message.<br>5.1-Invited User clicks No button.<br>5.2-System does not add those users as friends. |
| Related Use Cases | UC-1,UC-2,UC-4,UC-5,UC-8 |

| | |
|---|---|
| Use case Number | UC-8 |
| Use case Name | Attending to a Lobby |
| Actor | User, System,Database |
| Description | Users should able to join lobbies. |
| Precondition | User should loginned . |
| Scenario | 1. The user clicks join lobby button.<br>2. System shows a pop up screen including lobby ID row.<br>3. The user enters lobby ID<br>4. System sends lobby ID to the database.<br>5. Database checks if there is a room with that ID.<br>6. Database returns true to the system.<br>7. System shows the lobby screen to the user.<br>8. The user is able to see other partipicants in the lobby. |
| Postcondition | The User can play the game with other participants.<br>The User can make team selection.<br>The User can chating with other players.<br>The user can quit from the lobby. |
| Exceptions | 3.1-User enters invalid ID.<br>3.2-System shows error message.<br>5.1- Database returns false to the system.<br>5.2- System displays “Lobby is not exist” message |
| Related Use Cases | UC-1,UC-2,UC-4,UC-5,UC-9,UC-10 |

| | |
|---|---|
| Use case Number | UC-9 |
| Use case Name | Invite Friend |
| Actor | User |
| Description | The user invites friends by their unique user id’s or from the friend list . |
| Precondition | The user needs to login.<br>The user must have invited user’s id or have it as a friend user. |
| Scenario | 1. The user clicks invite button on lobby.<br>2. Pop-up shows up for select a friend or enter a user id.<br>3. The user selects user from friend list or enters a user id.<br>4. Lobby invite request shows up on other user. |
| Postcondition | 2.1-The user which we send invite request should accept or reject invite from pop-up. |
| Exceptions | The user may enter wrong or false id , in this case system will inform user about the error. |
| Related Use Cases | UC-1,UC-2,UC-4,UC-5,UC-6,UC-7 |

| | |
|---|---|
| Use case Number | UC-10 |
| Use case Name | Starting Game |
| Actor | User,Database |
| Description | The user creates or joins a room for play game. |
| Precondition | The user needs to login.<br>The user must be in a lobby. |
| Scenario | 1. The user clicks ready button on lobby if the host is not itself.<br>2. The user clicks starts button on lobby if the host is itself.<br>3. System create a game for current lobby players on database. |
| Postcondition | The users are able to play game |
| Exceptions | 3.1-System can not start a game at server,in this case lobby will be informed about error. |
| Related Use Cases | UC-1,UC-2,UC-4,UC-5,UC-8,UC-9 |

| | |
|---|---|
| Use case Number | UC-11 |
| Use case Name | Shooting |
| Actor | User,Database |
| Description | The user clicks shoot button at match. |
| Precondition | The user needs to login.<br>The user must be in a game.<br>The user must be alive. |
| Scenario | 1. The user clicks shoot button.<br>2. System detects if there is a player at the shooted area.<br>3. If there is a body,system detects user from current location and distance between 2 user.Else do nothing.<br>4. The System decides which user got hit.<br>5. The System updates database information about the user who got hit. |
| Postcondition | If the user got hit downs below 0 health point ,the system will count him as dead for specific time period.<br>The user got hit has loss health point |
| Exceptions | 3.1-System can not detect a user. |
| Related Use Cases | UC-1,UC-2,UC-4,UC-5,UC-10,UC-16,UC-17 |

| | |
|---|---|
| Use case Number | UC-12 |
| Use case Name | Chatting in the lobby |
| Actor | User |
| Description | Chatting in the lobby allows users to communicate with each other and interact in general chat before the game starts. |
| Precondition | Users need to have logged into the application.<br>Users are required to have a game identity and join the game lobby. |
| Scenario | 1-The users enters the chat section within the lobby.<br>2-The user greets other or initiates a conversation in the general chat room.<br>3- users can use private messaging to make tactical plans or share strategies with other users. |
| Postcondition | When the game starts and the lobby process concludes, the chat history within the lobby is saved.<br>The user's communicate and chat history become inaccessible once the game begins or the lobby is closed. |
| Exceptions | If a user looses connection, the system sends an error message. |
| Related Use Cases | UC-1,UC-2,UC-4,UC-5 |

| | |
|---|---|
| Use case Number | UC-13 |
| Use case Name | Team selection in lobby |
| Actor | User, Database |
| Description | Team selection in the lobby refers to a feature that allows players to choose their teams before the game starts. This case enables users to select the team of their choice and balance the teams as they see fit. |
| Precondition | Players need to have logged into the application.<br>Users are required to have a game id and join the game lobby. |
| Scenario | 1-The user is directed to the team selection screen.<br>2- From the database, users review available teams and team information.<br>3-The user selects their team.<br>4-The game automatically balances the teams and updates information of the team on database.<br>5-Before the game starts, users review the information related to their selected team. |
| Postcondition | The teams chosen by users and the number of users is organized to ensure balanced and fair gaming experience.<br>Once the team selection is completed, the chosen team information of users is displayed on the team lobby. |
| Exceptions | If a user makes an invalid selection on the team selection screen, the system sends an error message.<br>If a user looses connection, the system sends an error message. |
| Related Use Cases | UC-1,UC-2,UC-4,UC-5,UC-10 |

| | |
|---|---|
| Use case Number | UC-14 |
| Use case Name | Weapon Customization |
| Actor | Usern , System |
| Description | Weapon customization usage case refers to a feature that allows users to personalize their weapons. |
| Precondition | The User must be logged into the application.<br>The User is required to have a game id. |
| Scenario | 1- The user opens the in-game menu pop-up.<br>2-The user is directed to the weapon customization section from the in-game menu.<br>3-The user reviews the avaliable weapon options and costimization features.<br>4-The user selects appreance of their weapon.<br>5-The user saves and applies the chosen customizations.<br>6-The system closes the pop-up. |
| Postcondition | When weapon customization is completed, the changes made by players are saved. |
| Exceptions | If a user encounters a techical error during weapon customization, the system sends an error message.<br>If a user experinces a connection issue while saving the chosen customizations, the system sends an error message. |
| Related Use Cases | UC-1,UC-2,UC-4,UC-5,UC-10,UC-18 |

| | |
|---|---|
| Use case Number | UC-15 |
| Use case Name | Quiting Game |
| Actor | User, System |
| Description | The user can quit from current playing game. |
| Precondition | The user must logged in to app.<br>The user must be in the game. |
| Scenario | 1. User clicks menu/settings button.<br>2. In the opening menu, the user clicks to quit game button.<br>3. The user quit the game.<br>4. The System directs user to the main menu. |
| Postcondition | - |
| Exceptions | 2.1- User clicks return button.<br>2.2- The System closes the menu and user returns to the game. |
| Related Use Cases | UC-1,UC-2,UC-4,UC-5, UC-10, UC-18 |

| | |
|---|---|
| Use case Number | UC-16 |
| Use case Name | Reloading Ammo |
| Actor | User, System, Database |
| Description | The user can reload their ammo. |
| Precondition | The user must logged in to app.<br>The user must be in the game. |
| Scenario | 1. User clicks reload ammo button.<br>2. The system sends a message to the database about refulling the users ammo with sending user ID.<br>3. The Database updates the record and sends a message to the system.<br>4. System updates amount of ammo information of user.<br>5. The user's ammo reloaded.<br>6. The user's ammo finishes,<br>7. System refilss user's ammo. |
| Postcondition | The user's ammo reloaded and ammo in maximum number. |
| Exceptions | 1.1- If the users' ammo is finished, the system automatically reloads the ammo. |
| Related Use Cases | UC-1,UC-2,UC-4,UC-5, UC-10, UC-5 |

| | |
|---|---|
| Use case Number | UC-17 |
| Use case Name | Respawning |
| Actor | User, System |
| Description | The user can respawn after seconds after the death. |
| Precondition | The user must logged in to app.<br>The user must be in the game.<br>The user must be death. |
| Scenario | 1. The System disables the reload and shoot buttons for 10 seconds.<br>2. The systems shows up a clock timer that represents remaining time for respawning.<br>3. The user wait 10 seconds.<br>4. The System enables the buttons.<br>5. User continues to play with full health and full ammo. |
| Postcondition | The users' scores updated. |
| Exceptions | 1.1- When the user waits 10 seconds, if trying to use reload ammo or fire button, the system returns "You can not use buttons when you die" to the user. |
| Related Use Cases | UC-1,UC-2,UC-4,UC-5, UC-10, UC-11 |

| | |
|---|---|
| Use case Number | UC-18 |
| Use case Name | Show Menu |
| Actor | User, System |
| Description | The user presses “Show Menu” option during the game. |
| Precondition | The user must be logged in.<br>Users are required to have a game id and join the game lobby. |
| Scenario | 1. The user presses the “Show Menu” button.<br>2. The system opens a pop-up page.<br>3. The user selects one of the following options: Settings, Resume, Quit<br>4. The user presses the button of selected option.<br>5. The system closes the pop-up screen. |
| Postcondition | The page is loaded according to the selection of the user.<br>If “Setting” button is pressed, then the system opens a pop-up page including settings of the game. The user can only alter the settings like sound, brightness. Other settings can only be altered by the host of the game.<br>If “Resume” button is pressed, then the user continues playing.<br>If “Quit” button is pressed, then the user quits the game. |
| Exceptions | - |
| Related Use Cases | UC-1,UC-2,UC-4,UC-5, UC-10, UC-15 |

| | |
|---|---|
| Use case Number | UC-19 |
| Use case Name | Surrender |
| Actor | User, System |
| Description | One of the opposing teams may surrender if the majority of the team users vote for surrender. |
| Precondition | The user must be logged in.<br>User are required to have a game id and join the game lobby.<br>The user must be in game. |
| Scenario | 1. One of the users offer to surrender by clicking the “Surrender” button on the screen.<br>2. The system opens a pop-up page on each team member’s screen.<br>3. Rest of the team members vote whether to surrender or not.<br>4. The system closes the pop-up page. |
| Postcondition | If majority of team members vote for “Surrender” the game ends and the opposing team wins.<br>If not, the game continues. |
| Exceptions | If the votes for “Surrender” and “Not surrender” has a tie, then the team captain decides whether to surrender or not. |
| Related Use Cases | UC-1,UC-2,UC-4,UC-5, UC-10 |

| | |
|---|---|
| Use case Number | UC-20 |
| Use case Name | Set Game Rules |
| Actor | User |
| Description | The host sets the rules of the game in lobby before starting the game. |
| Precondition | The user must be logged in as host.<br>The user should create a lobby. |
| Scenario | 1. The system opens a pop-up including game rules after creating the lobby.<br>2. The user selects the rules of the game.<br>3. The system saves rules and closes the pop-up. |
| Postcondition | The rules of the game are set. |
| Exceptions | - |
| Related Use Cases | UC-1,UC-2,UC-4,UC-5, UC-6 |