ÇANKAYA UNIVERSITY COMPUTER ENGINEERING

CENG 407

PROJECT REPORT

GROUP 1

TEAM MEMBERS

Alperen Kaan SALT 201911052

Mehmet Emir HOCAOĞLU 201911029

Nadide SOLMAZ 201911056

Seyit KOYUNCU 201911036

Zeynep Deniz DÖNMEZ 202011012

Contents

Abstract

This report delves into the comprehensive analysis and integration of network, mobile technology, GPS tracking, body segmentation, and an AI module within a mobile application framework. The amalgamation of these components aims to propel the functionality and user experience of mobile applications to new heights. The network section assesses the robustness and efficiency of network connectivity, emphasizing the pivotal role it plays in the seamless operation of mobile applications. Subsequently, the mobile segment scrutinizes the evolving landscape of mobile technology, focusing on enhancing user interactions and accessibility. Moreover, the GPS tracking section elucidates the significance of location-based services, exploring their integration and impact on mobile applications, particularly in improving user engagement. The body segmentation component investigates cutting-edge technology employed in mobile applications to discern and interpret body movements, contributing to augmented reality experiences. Lastly, the AI module expounds on the incorporation of artificial intelligence, elucidating its role in optimizing and personalizing user experiences within the mobile application domain. This report provides a comprehensive overview of the intricate interplay between network infrastructure, mobile technology, GPS tracking, body segmentation, and artificial intelligence, highlighting their synergistic contribution to the evolution of mobile applications.

1. Introduction

The gaming industry is on the brink of a significant evolution with the advent of a multiplayer mobile gaming experience that emulates the thrill of laser tag and paintball within a digital platform. This project endeavors to create an innovative FPS (First Person Shooter) game, merging the excitement of these real-world activities with cutting-edge technology, specifically designed for mobile devices.

Distinct from conventional FPS games, this multiplayer experience sidesteps the necessity for virtual environments and three-dimensional player models. Instead, it leverages real-time body segmentation for shooting detection, allowing damage calculations based on segmented body parts such as head, limbs, and torso. Tensorflow Lite (tflite) is the proposed tool for the segmentation, while the game's structure could be developed using platforms that support tflite, including Flutter, React Native, or Unity.

In addition to the segmentation feature, the game will integrate a 3D model of a weapon within the game scene, complete with animations. Upon hitting a player, a visual indicator in the form of a red frame surrounding the screen will warn the affected player, accompanied by an in-game shooting effect. Sound effects, particularly firing sounds, will complement the visual cues. As the game seeks to create an immersive experience, real-world sound effects suffice, eliminating the need for additional sound generation.

Player identification within the game scene will be determined using a combination of GPS location, the orientation of the player's device, and an estimated distance from the captured image. This identification will influence gameplay dynamics, where damage incurred and points gained by the shooter will be calculated accordingly.

To heighten the gaming experience, the game will also support firing commands via a Bluetooth controller, allowing players to use compatible Bluetooth weapon controllers during gameplay.

The game setup emulates dynamics akin to popular games like Kahoot, where one player initiates the game while others join and form teams.

For real-time in-game information sharing, the game will employ an AMQP-based message queue structure, ensuring seamless communication between players. Alternatively, different structures may be considered during development stages.

While these outlined features serve as the baseline, the game's potential for future expansions includes the ability to purchase skins for the in-game weapon models and later diversifying features with different weapon types (machine guns, shotguns, flamethrowers, etc.). Design considerations for these modifiable in-game dynamics are under contemplation, ensuring adaptability and variability within the gaming environment.

2. Literature Review

2.1. Mobile Development

Mobile development in Unity and Flutter refers to the process of creating mobile apps using the Unity game engine and the Flutter cross-platform UI framework.

Unity is a popular game engine that provides a powerful set of tools for creating 2D and 3D games. It also has a number of features that make it well-suited for mobile development, such as support for native rendering and input, as well as a number of pre-built assets and tools.

Flutter is a cross-platform UI framework that allows developers to create native-looking apps for iOS and Android from a single codebase. It is known for its fast development speed, expressive syntax, and rich widget library.

2.1.1. Flutter

Flutter is a popular choice for cross-platform mobile app development due to its versatility and consistent user experience on both Android and iOS.

Frontend Programming: Flutter's front-end development centers around creating the user interface (UI) and managing user interactions. It utilizes the Dart programming language, known for its speed and reliability.

Widgets: Flutter's UI is built using widgets, which are highly customizable and combinable building blocks for designing various app components. This modularity makes it easy to create visually appealing and responsive interfaces.

Backend Programming: While Flutter primarily focuses on the client-side (frontend), the server-side or backend of a mobile app is equally crucial. Flutter can be integrated with various backend technologies and services.

REST APIs: Flutter apps often communicate with remote servers through RESTful APIs for data retrieval and updates. Backend developers use technologies like Node.js, Python, Ruby on Rails, etc., to create these APIs.

Databases: Data storage and management are critical on the backend. Developers typically use databases such as MySQL, PostgreSQL, or NoSQL databases like MongoDB.

Server-Side Logic: The backend handles business logic, authentication, and data processing. Technologies like Express.js, Django, or Flask can be used for this purpose.

Authentication and Security: Backend development ensures data security and implements user authentication systems, safeguarding user data.

TenserFlow Lite in Flutter:TensorFlow Lite [1] is a lightweight version of Google's machine learning framework optimized for mobile and edge devices. Combining Flutter with TensorFlow Lite allows developers to integrate machine learning capabilities seamlessly into their mobile applications.

AR in Flutter:Augmented Reality (AR) in Flutter involves integrating AR features into mobile applications using the Flutter framework. AR enhances the real-world environment by overlaying digital content, creating interactive and immersive experiences. It supports both ArCore(for android) and Arkit(for ios)[2]

In summary, Flutter offers a comprehensive solution for both frontend and backend development, making it a well-rounded choice for cross-platform mobile app development.

2.1.2. Unity

Unity is a versatile game engine and development platform that can be extended to create augmented reality (AR) applications by integrating TensorFlow, an open-source machine learning framework. Here's an extended overview:

Cross-Platform Development: Unity supports a wide range of platforms, including mobile devices, desktop, consoles, and AR/VR headsets, making it an ideal choice for creating AR experiences that reach a broad audience.

AR Foundation: Unity's AR Foundation framework simplifies AR app development by offering a consistent environment for both Android ARCore and iOS ARKit, reducing platform-specific challenges.

TensorFlow: TensorFlow, developed by Google, is a robust machine learning framework suitable for various tasks, including image recognition and natural language processing. When integrated with Unity, it brings machine learning capabilities to AR applications.

Object Recognition: TensorFlow can be used to train models for object recognition, allowing AR applications to understand and interact with the real world, providing context-aware experiences.

Augmented Reality (AR): AR enhances the user's perception of the real world by overlaying digital content through a device's camera.

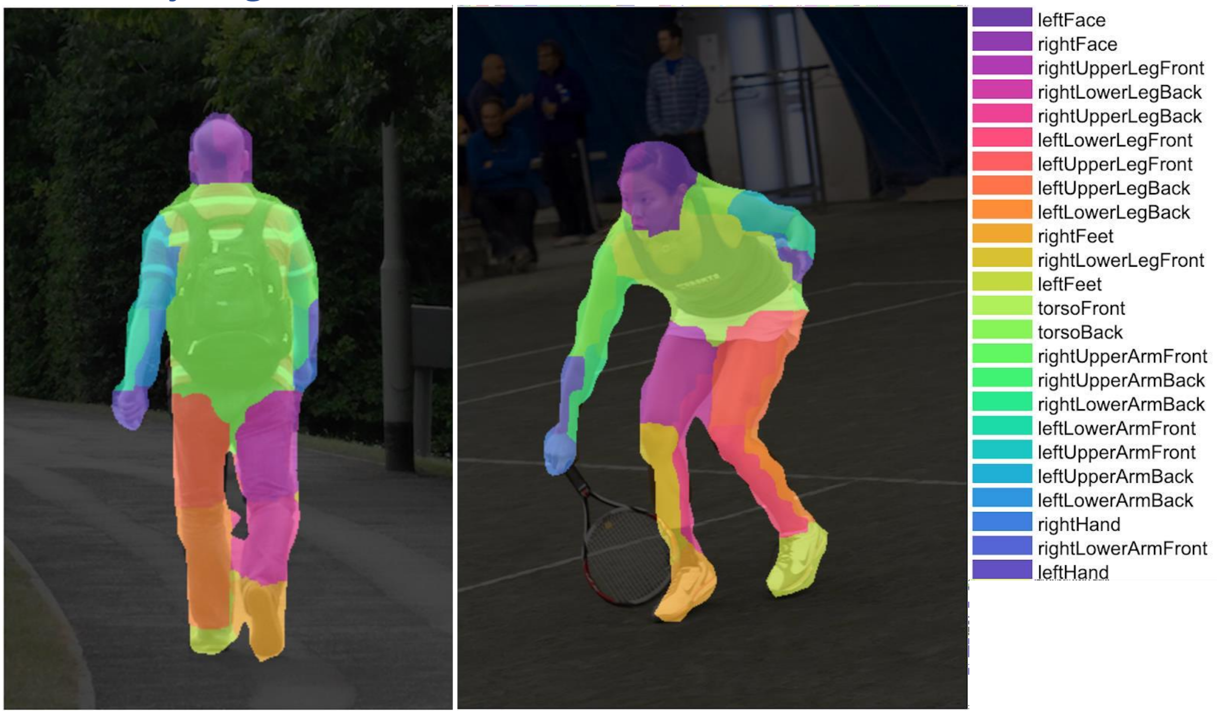
Object Tracking: AR apps can track and recognize physical objects in real time, providing information, instructions, or entertainment.

Gesture Recognition: Gesture recognition in AR applications enables users to interact with virtual objects and control them.

Machine Learning in AR: TensorFlow's [3] machine learning capabilities can make AR applications smarter by analyzing real-world data and providing context-aware experiences, such as real-time language translation or object identification.

In summary, Unity, in conjunction with TensorFlow, empowers developers to create versatile and intelligent AR applications with object recognition, tracking, and gesture control, enhancing the user's interaction with the real world.

2.2. Body Segmentation



Body segmentation refers to the process of identifying and isolating individual body parts or segments in an image or video. It is also known as human parsing. This typically includes tasks such as segmenting the human body into different regions, such as the head, torso, arms, and legs [4]. In the body segmentation we can use a lot of different Deep Learning tools. Such as, Tensorflow, PyTorch, YOLO etc. We will examine all three tools, but our main focus will be tensorflow.

2.2.1. PyTorch

PyTorch is a machine learning framework based on the Torch library, used for applications such as computer vision and natural language processing, originally developed by Meta AI and now part of the Linux Foundation umbrella. It is free and open-source software released under the modified BSD license. In PyTorch we can do image segmentation with using Convolutional Neural Network (CNN).

2.2.2. YOLO

You Only Look Once (YOLO) is one of the most popular model architectures and object detection algorithms. It uses one of the best neural network architectures to produce high accuracy and overall processing speed, which is the main reason for its popularity. In YOLIO we can do image segmentation with using YOLO-NAS.

YOLO-NAS is the latest state-of-the-art object detection model released by Deci AI. The model was generated by deci technology, Deci's Neural Architecture Search engine. YOLO-NAS surpasses other object detection models in terms of speed and accuracy. In building YOLO-NAS, Deci sought to high-quality some key limiting factors of current YOLO models, such as inadequate quantization support and insufficient accuracy-latency tradeoff.

2.2.3. Tensorflow

TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.

2.2.3.1. Tensorflow Lite

TensorFlow Lite lets you run TensorFlow machine learning (ML) models in your Android apps. The TensorFlow Lite system provides prebuilt and customizable execution environments for running models on Android quickly and efficiently, including options for hardware acceleration. We can use BodyPix [5] and MediaPipe [6] with Tensorflow Lite in the mobile devices.

2.2.3.2. BodyPix

BodyPix is an open-source machine learning model developed by Google that allows for the estimation of human bodies in images and videos. It enables the real-time segmentation of a person's body into distinct parts, such as the left arm, right arm, torso, and legs, without the need for specialized depth sensors. BodyPix is particularly known for its ability to provide detailed and accurate body segmentation in real time.

Key features of BodyPix include:

1. Real-Time Performance: BodyPix is designed to operate in real time, making it suitable for various interactive applications and experiences.
2. Detailed Body Segmentation: It can accurately identify and segment various body parts, allowing for precise analysis and manipulation of body-related data.
3. Compatibility and Flexibility: The model is compatible with various platforms and can be integrated into different applications and frameworks.

   Customization and Training: BodyPix allows for model customization and further training to adapt to specific use cases and requirements.

BodyPix has found applications in diverse fields, including virtual try-on experiences, fitness applications, augmented reality effects, and virtual backgrounds in video conferencing. Its accurate body segmentation capabilities have made it a popular tool for developers and researchers working on human-centric applications and projects.

2.2.3.3. MediaPipe

MediaPipe is an open-source framework for building multimodal (audio, video, and other time series data) applied machine learning pipelines. It was developed by Google Research and offers customizable solutions for researchers and developers. MediaPipe provides cross-platform support and processing capabilities for a variety of use cases, including object detection, face detection, hand tracking, pose estimation, and more.

Some key features and components of MediaPipe include:

1. Cross-Platform Support: MediaPipe supports various platforms, including Android, iOS, and desktop environments, allowing developers to deploy machine learning models across different devices.
2. Modular Architecture: It provides a modular and flexible architecture that enables the development of complex pipelines for processing multimedia data.
3. Pre-Trained Models and Customization: MediaPipe offers pre-trained models for various tasks, as well as the ability to customize and train models to meet specific requirements.
4. Real-Time Processing: It is designed to support real-time processing of audio and video data, making it suitable for applications that require immediate analysis and feedback.

2.3. Network

A network is a system of interconnected devices or computers that enables them to communicate and share data with each other, either locally or over long distances. In the context of gaming, networks play a crucial role in connecting players, servers, and gaming platforms together. They facilitate online multiplayer experiences, allowing players from around the world to compete or cooperate in real-time, enhancing the social aspect of gaming. A robust and low-latency network is essential for smooth gameplay, ensuring that game data, such as player actions and updates, can be transmitted quickly and accurately. Without a reliable network, online gaming experiences would be hindered by lag, disconnections, and poor performance, making it a fundamental component for the success and enjoyment of modern video games.

2.3.1. TCP (Transmission Control Protocol)

TCP is not always suitable for gaming networks due to its inherent reliability features. While TCP ensures that data packets are delivered accurately and in the correct order, it also introduces latency, or delay, in the transmission process. In fast-paced online games, such as first-person shooters or real-time strategy games, low latency is crucial for a responsive and smooth gaming experience.

2.3.2. UDP (User Datagram Protocol)

UDP is suitable for gaming networks when low latency is crucial. It offers faster data transmission but sacrifices some reliability. In real-time, fast-paced games, such as first-person shooters, low latency provided by UDP ensures a more responsive and smoother gaming experience, even if occasional packet loss may occur.

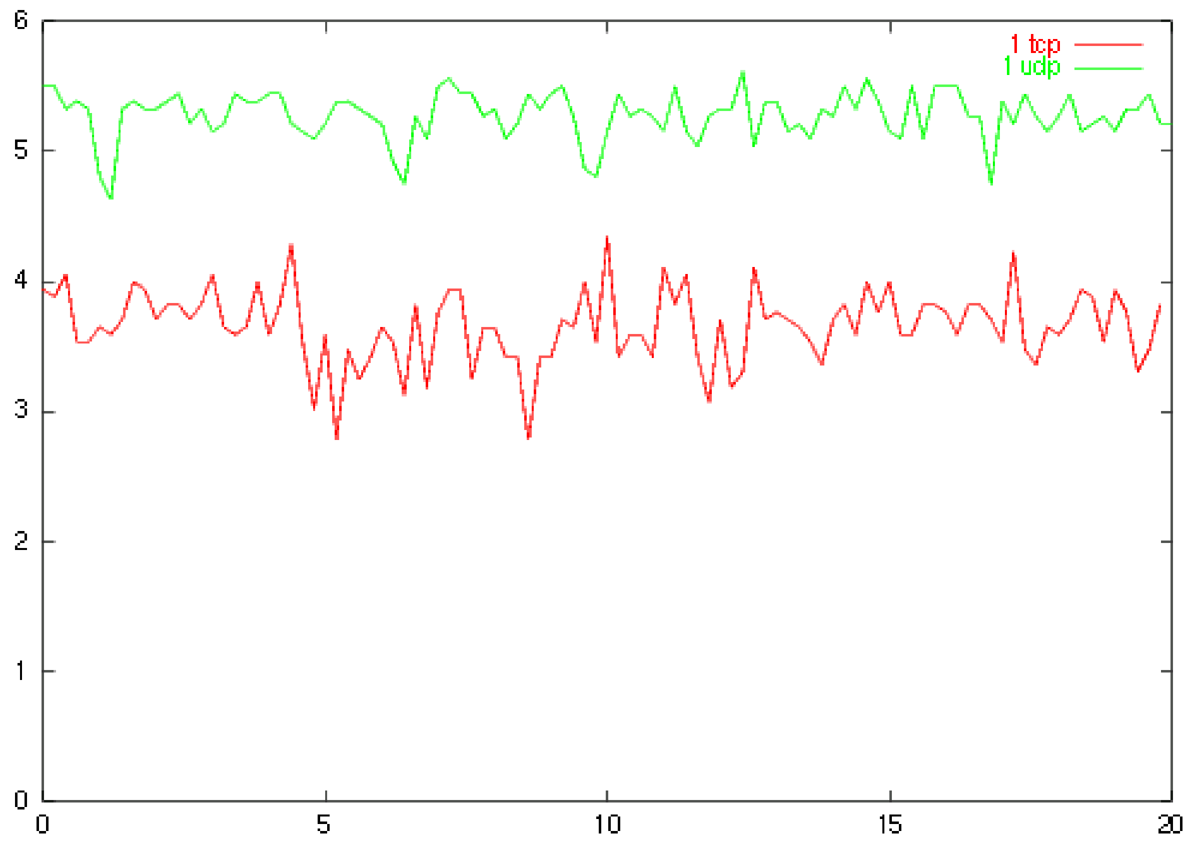
2.3.3. AMQP (Advanced Message Queuing Protocol)

AMQP is suitable for gaming networks because it enables efficient and reliable communication between game servers, players, and other components. It provides asynchronous messaging, allowing game systems to exchange data and events without waiting for immediate responses. AMQP's queuing mechanism ensures reliable message delivery, making it ideal for handling real-time events, player interactions, and game state updates. By facilitating seamless communication and coordination in gaming networks, AMQP enhances the overall gaming experience for players.

2.3.4. Why AWS?

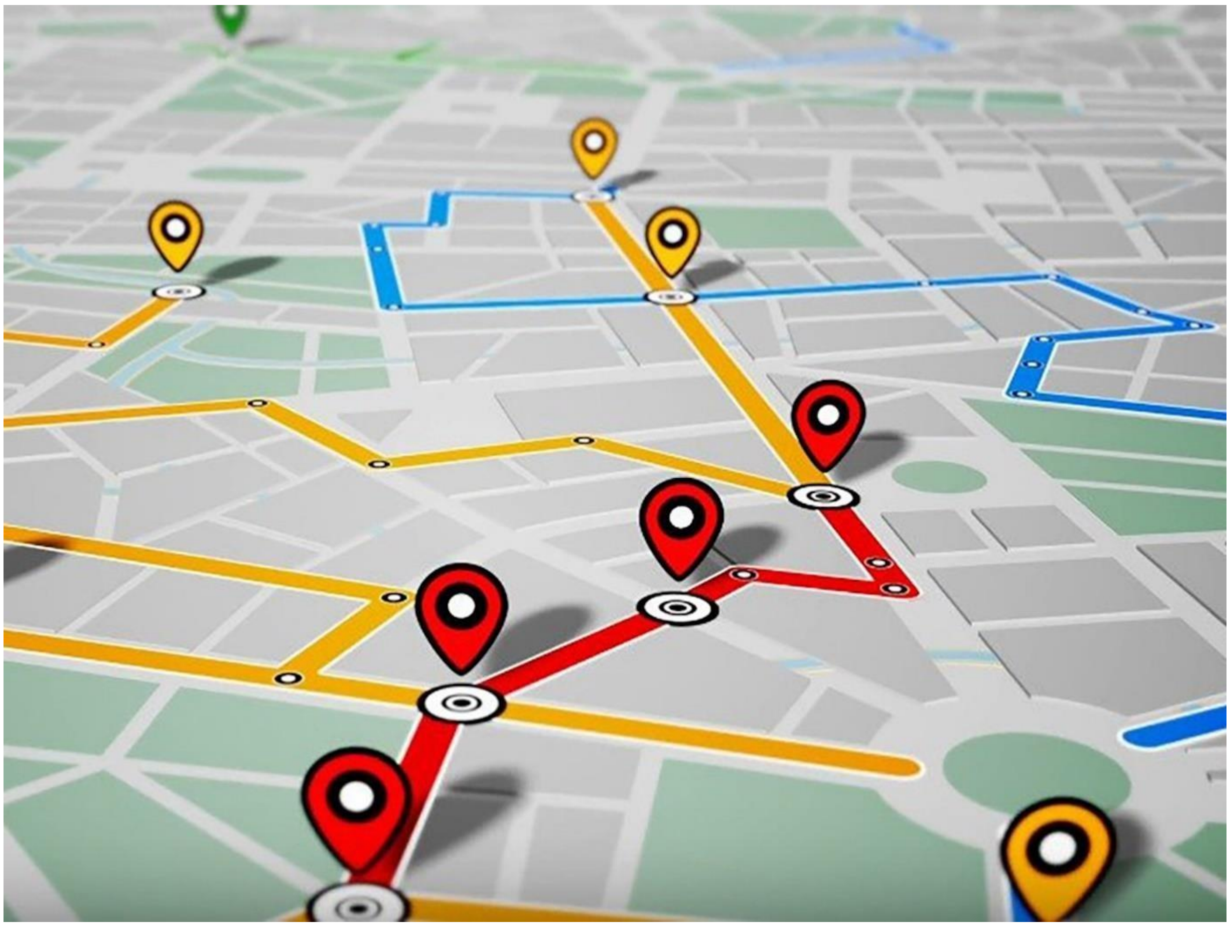
AWS has everything that's needed for this infrastructure. Of course, a similar infrastructure could be built using other big cloud providers, but AWS has a huge amount of documentation and a large community, so it doesn't take much time to solve every small problem. But if we neglect this dynamic scaling, DigitalOcean looks like a better option thanks to its simple virtual machine configuration and pricing policy [7].

Throughput (Mbps) vs time (Seconds) for noncompeting flows



2.4. GPS Tracking

GPS tracking relies on a network of 24 satellites orbiting the Earth and ground-based devices. It enables the precise determination of the location of an individual or object on Earth. The system collects and utilizes three distinct sets of data: positioning, navigation, and timing. GPS tracking systems collect location data, providing the ability to track and trace movements, analyze routes, and monitor the real-time location of assets or individuals. GPS tracking technology provides precise and near real-time location information, enabling businesses, governments, and individuals to make informed decisions, enhance safety measures, and optimize operations [8].



2.4.1. React Native (Geolocation Module)

React Native Geolocation is a module that provides an interface for accessing the device's geolocation services in React Native applications. It allows us to retrieve the device's current location (latitude and longitude). It enables them to track changes in location, monitor user movement, and integrate location-based features within their applications. Through the React Native Geolocation module, we can create location-aware applications, ranging from mapping and navigation apps to services that offer location-based content, all while adhering to user privacy and permission standards [9].

2.4.2. Flutter (Location Module)

The location module in Flutter is a package that enables us to access and manage geolocation services. This package provides functionalities to retrieve the device's location, monitor location changes, and incorporate location-based features. The location package offers a simplified and platform-independent API for handling location services across IOS and Android devices. We can request permissions, obtain the current device location (latitude and longitude), track movement, and implement various location-aware features [10].

2.4.3. Utilizing Cross-Platform Capabilities (Android and IOS)

Both the React Native Geolocation and Flutter's location modules are usable on both IOS and Android platforms. Both modules provide cross-platform compatibility, enabling the activation of location services, retrieving location data, monitoring location changes, and implementing location-based functionalities for both mobile operating systems. This allows for a cross-platform development experience with similar or identical functionalities across both IOS and Android devices.

2.5. AI Tools for Mobile Development

In the ever-evolving landscape of mobile app development, developers are constantly seeking ways to enhance user experiences and streamline development processes. In this quest for improvement, Artificial Intelligence (AI) has emerged as a transformative force. AI tools have revolutionized the way mobile apps are created and used, offering developers a powerful set of capabilities to build smarter, more efficient, and user-centric applications.

2.5.1. Google's Machine Learning Kit

Google's Machine Learning Kit stands out as the most extensively employed AI-driven software development tool in the current market. This sophisticated yet user-friendly tool serves as a conduit for Google's formidable machine learning capabilities, specifically tailored for mobile developers. Functioning as Google's Software Development Kit (SDK) for app development, ML Kit facilitates the seamless integration of machine learning functionalities into mobile applications. Compatible with both Android and iOS devices, ML Kit contributes to a more immersive and personalized user experience. Notably, the processing tasks of ML Kit occur locally on the device, ensuring swift performance for real-time applications, such as camera-input processing. Its offline functionality allows for the processing of images and text, maintaining data on the device. Additionally, ML Kit is offered free of charge and seamlessly interfaces with other cloud AI services from Google, such as Google Cloud Vision AI.

It is possible to add AI-powered features to your product, such as:
-Barcode scanning
-Text recognition
-Face recognition
-Real-time object recognition and tracking
-Digital ink recognition [11-12]

The ML Kit for Flutter by Google comprises a collection of Flutter plugins designed to empower Flutter applications in leveraging Google's independent ML Kit. Furthermore, the integration of ML Kit with Unity is also a viable option [10-13].

2.5.2. Firebase

Firebase Machine Learning serves as a mobile SDK that seamlessly integrates Google's advanced machine learning capabilities into both Android and Apple applications, offering a potent and user-friendly solution. Regardless of your level of experience with machine learning, you can easily implement the desired functionality with just a few lines of code. Additionally, Firebase ML offers convenient APIs, enabling the utilization of custom TensorFlow Lite models within your mobile apps. Furthermore, it allows the configuration of Flutter applications and Unity projects to establish connections with Firebase [14-15-16].

2.5.3. CoreML

CoreML stands as Apple's framework, providing the means to incorporate machine learning (ML) models into your iOS application. It is both cost-free and harmonizes seamlessly with Apple's array of frameworks and development tools. The key benefit of CoreML lies in its simplification of the ML model integration process into your application, featuring a user-friendly drag-and-drop interface for enhanced ease of use. CoreML allows for the utilization of your customized ML model within your application. Leveraging the Create ML feature enables the development of a tailored model trained on your specific data, and it also supports the import of models from external training libraries. The process involves converting these models into the CoreML format using CoreML Tools or seamlessly integrating the model directly into your application through top-level APIs.

It'll allow you to add functionalities to your app, such as:
-Text processing and analysis
-Converting audio to text
-Image recognition and analysis
-Sound analysis
-Sentiment analysis
-Audio transcription
-Face recognition
-Barcode recognition [11]

2.6. Conclusion

In this article, we examined our mobile game application Blast Strike which users can play interactive first person shooter game on their mobile phones with their friends or family. Blast Strike serves a great environment for playing real time first person shooter game to the users.

This report explores the integration of essential components—network infrastructure, mobile technology, GPS tracking, body segmentation, and artificial intelligence—within the framework of mobile applications. The overarching goal is to showcase how these elements can collaboratively enhance user experiences and functionality. The discussion encompasses network protocols, cross-platform mobile development using Flutter and Unity, real-time body segmentation tools like TensorFlow Lite, GPS tracking capabilities in React Native and Flutter, and the integration of artificial intelligence through tools like Google's ML Kit, Firebase, and CoreML. This holistic exploration illustrates the collective impact of these technologies in shaping the landscape of mobile applications, showcasing their versatility and potential across various applications beyond the specific example of a First Person Shooter (FPS) game.

Overall, Blast Strike offers quite unique features to the users. It's a great approach to creating a real time environment that users can play first person shooter game with their companies. If you're looking for an experience like laser tag or paintball that you can quickly play with your friends on your mobile phone, Blast Strike is worth checking out.

As mentioned, there is no mobile application game on the market like our ones. There are several distinct applications that includes body segmentation, gps tracking and AI models. However, our game is aiming to mix all important feautures and expose a great user friendly mobile application.

2.7. Related Works

In our quest for innovation, we conducted an in-depth analysis of projects that mirror our objectives and technological direction. Below, we present a curated list of select examples within the mobile application domain, each with its distinct approach and technological integration.

AR Warriors

It is a free mobile app that allows players to battle each other in the real world using their smartphones. The game uses AR technology to overlay digital characters and objects onto the user's camera view, creating a seamless experience where players can feel like they are actually fighting in a real battle. However, it is not exist on the google play store or app store.

AR Warriors features a variety of game modes, including team deathmatch, capture the flag, and free-for-all. Players can also choose from a variety of different weapons and equipment, each with its own unique strengths and weaknesses.

- https://ar-warriors-weapon-camera-augmented-shooter.softonic.com.tr/android (Youtube link :https://youtu.be/x20IrgAKuZw)

2.7.1. Related Development

- https://www.theseus.fi/bitstream/handle/10024/337183/Korhola_Samuli.pdf?sequence=2&isAllowed=y

3. Software Requirements Specification

3.1. Introduction

3.1.1. Purpose

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.

3.1.2. 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.

3.1.3. 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.

3.2. Overall Description

3.2.1. 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.

3.2.2. User Characteristics

3.2.2.1. 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.

3.2.2.2. Technical Profiency

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.

3.2.2.3. 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.

3.2.2.4. Accesibility 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.

3.2.2.5. 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.

4. Requirements Specification

4.1. External Interface Requirements

4.1.1. 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 a game room from the main menu. After users are in a lobby, they can chat with each other from the chatbox and set their status to ready. After every user is ready, the host can start the game from the start button. After the game starts, the main interface becomes the view of the front camera and some UI elements related to the game. Users who play the game will see a small crossover in the middle of the screen, shooting button, reloading button, and settings. After the game ends, users return to the lobby.

4.1.2. Hardware Interface

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

4.1.3. 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.

4.1.4. 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 the system to access their camera.

4.2. Functional Requirements

4.2.1. System and Mobile Application Requirements

4.2.1.1. 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.

4.2.1.2. 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.

4.2.1.3. 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.

4.2.1.4. Score Table

The game should track the score of each player individually and also each team to decide which team wins the game.

4.2.1.5. 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.

4.2.1.6. 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.

4.2.1.7. 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.

4.2.1.8. Friend Management

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

4.2.1.9. Join Lobbies

The system shall allow users to join existing multiplayer game lobbies via request and entering a unique room id from the join button on the menu.

4.2.1.10. 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 users.

4.2.1.11. 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.

4.2.1.12. *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.

4.2.1.13. *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.

4.2.1.14. *Quit Game*
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.

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

4.2.1.16. *Respawn Timer*
Users can respawn after a specified duration following their death.

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

4.2.1.18. *Customize Game Settings*
The host can set and customize game parameters in the lobby game mode, player count, and win conditions.

4.3. Non Functional Requirements

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

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

4.3.3. Reliability

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

4.3.4. 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.

4.3.5. Compatibility

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

4.3.6. Network Connectivity

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

4.3.7. Cross-Platform Compatibility

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

4.3.8. User Experience (UX)

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

4.3.9. Compliance

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

4.3.10. Database Scalability

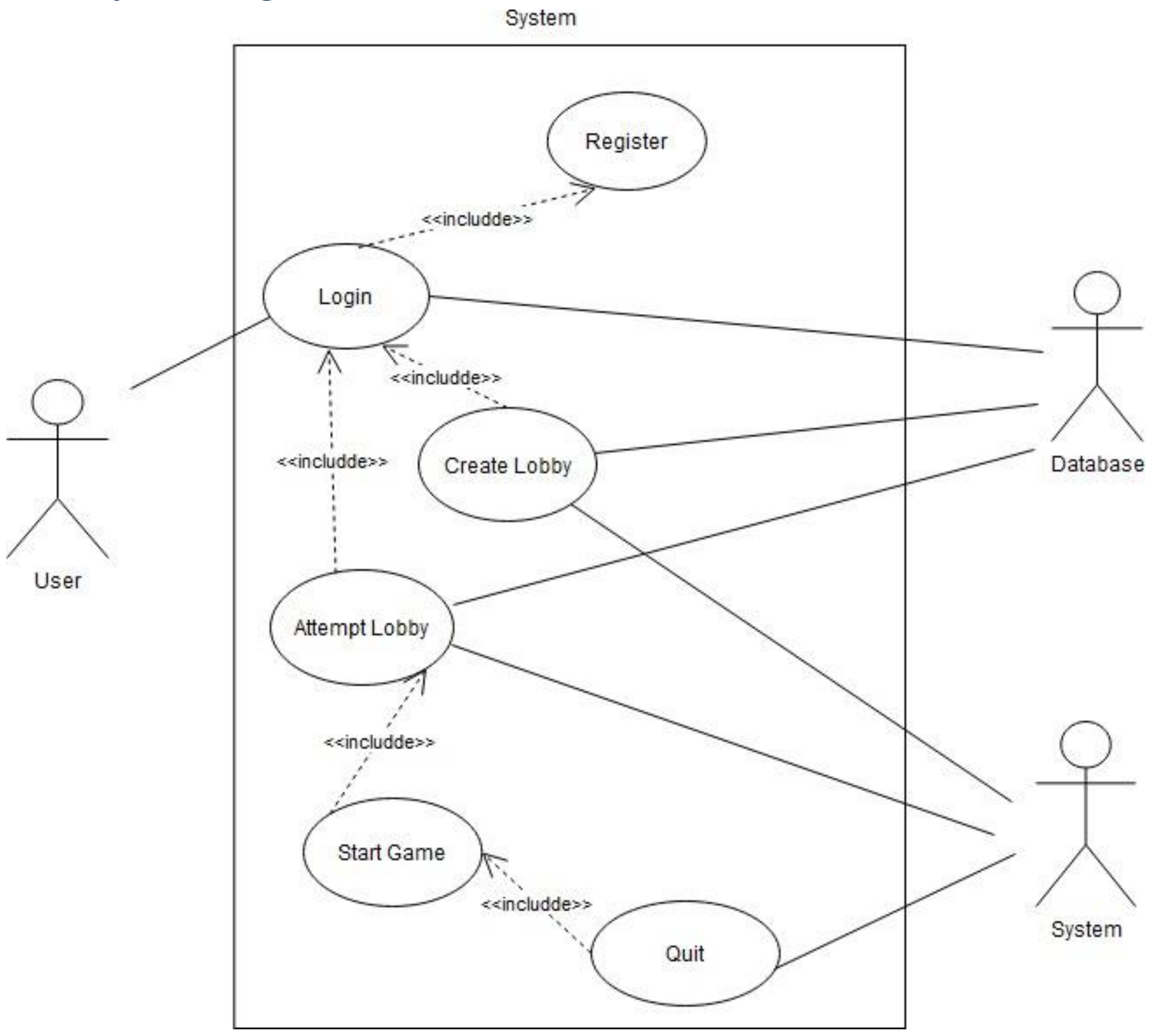
The system should be able to scale the database to accommodate a growing user base.

4.3.11. Server Scalability

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

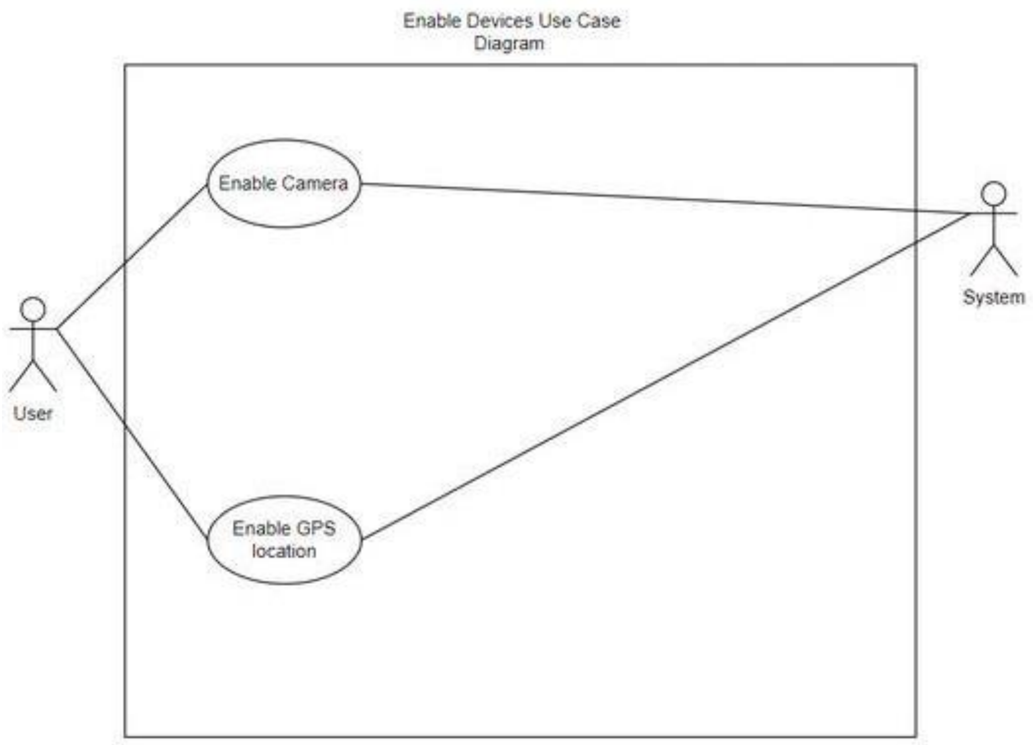
4.4. Use Cases

4.4.1. System Diagram

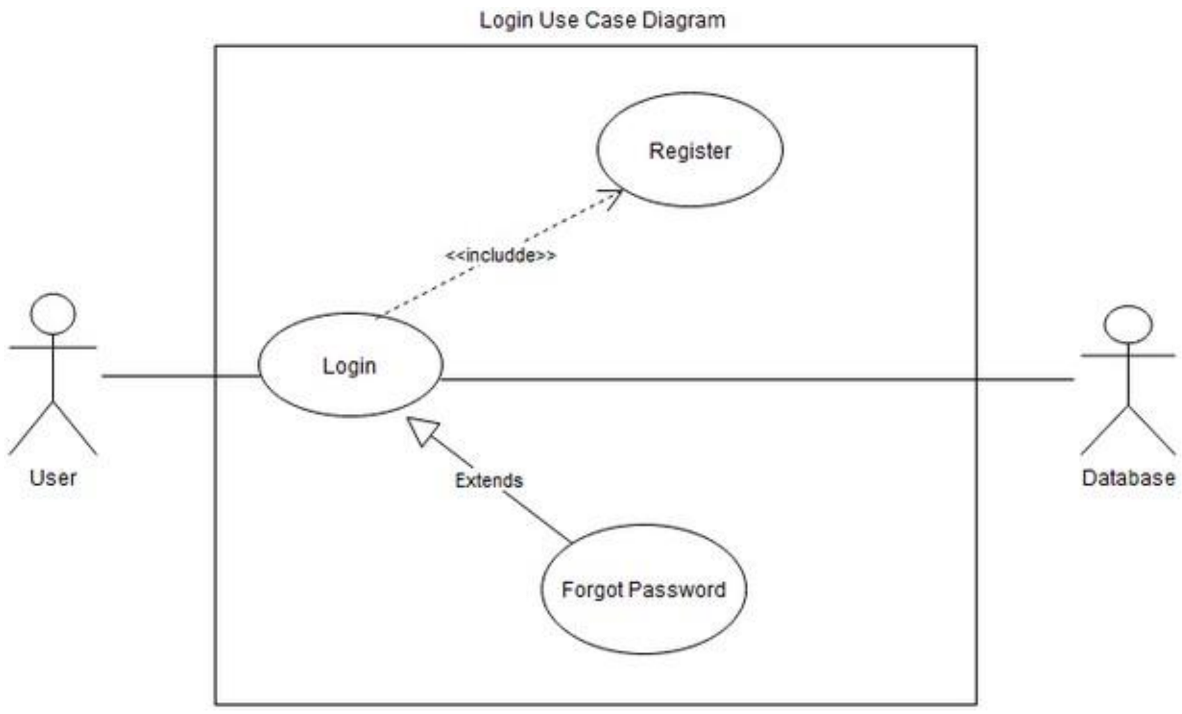


4.4.2. Use Case Diagrams

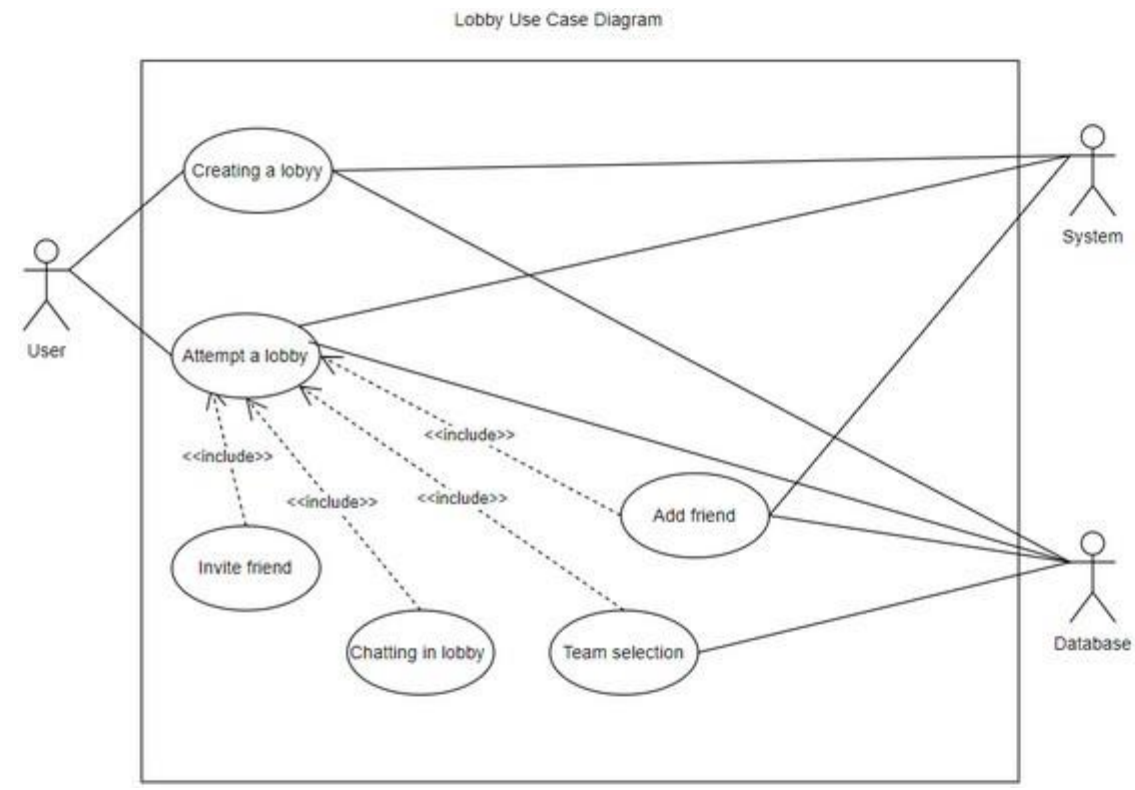
4.4.2.1. *Enable Devices*



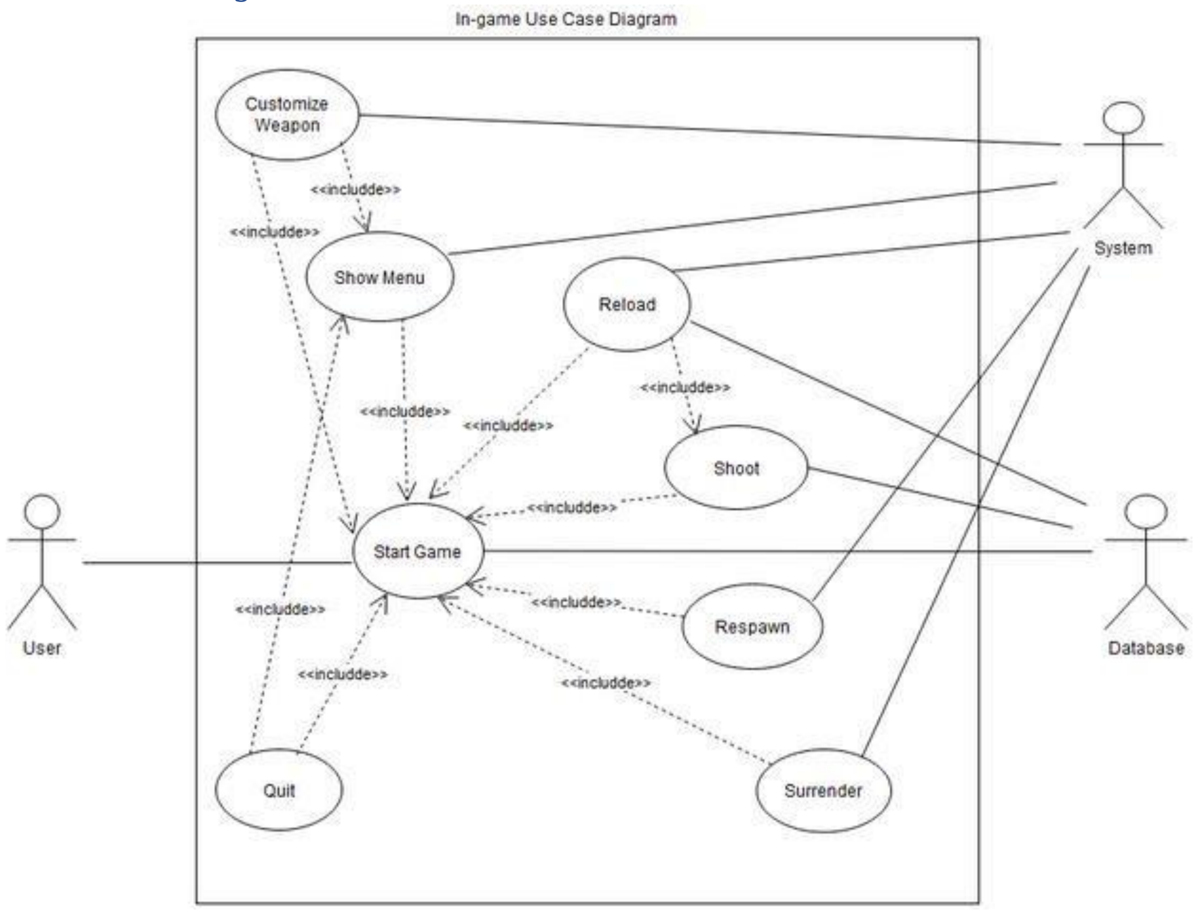
4.4.2.2. *Login*



4.4.2.3. *Lobby*



4.4.2.4. *In-game*



4.4.3. Use Cases

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 downloads the application to their phone. - The user does not need to be registered with the application before. - 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. 2. The user enters personal information such as name, surname, username, e-mail, on the registration screen. 3. The user enters the verification code into the application. 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. 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. - The user must enter the correct e-mail, and password.
Scenario	1. The user enters the application. 2. After the user enters the information for login, the system sends a verification code to the email for authentication. 3. The user enters the verification code into the application. 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 and the system displays a error message. 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. 2. The system asks the user for verification to make sure that the user made this reque 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	2.1. 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. 2. The user turns on location services on his phone. 3. The user allows the application to access the location. 4. The user starts using the application.
Postcondition	The system can access user’s location information to play game.
Exceptions	2.1. The User can not enable location service. 2.2. 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. 2. The user turns on camera services on his phone. 3. The user allows the application to access the camera. 4. The user starts using the application.
Postcondition	The system can access user’s camera information to play game.
Exceptions	3.1. The user may try to log in and use the application without turning on the camera services, this should send an error message.
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. - The User should enable gps location. - The User should enable camera.
Scenario	1. The User clicks the “Create Lobby” button. 2. System directs user to the game lobby and indicates the user as “Host”. 3. System gives a unique ID to the lobby. 4. System sends unique ID to the database. 5. Database keeps the ID.
Postcondition	- The User can invite friends and starts a game. - The User can set the game rules. - The User can make team selection. - The User can chating with other players. - The user can starting the game. - The user can quit from the lobby.
Exceptions	4.1. The User can not connect the system. 4.2. 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. 2. System opens a new pop-up screen which includes a row for user ID. 3. User enters the ID of the user that wanted to add as a friend and clicks a add friend button. 4. System sends a notification to the other user which have entered ID.

	5. Invited User clicks “Yes” button for adding a friend request. 6. System sends Users ID’s to the database. 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. 3.2. System sends error message. 5.1. Invited User clicks No button. 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. 2. System shows a pop up screen including lobby ID row. 3. The user enters lobby ID 4. System sends lobby ID to the database. 5. Database checks if there is a room with that ID. 6. Database returns true to the system. 7. System shows the lobby screen to the user. 8. The user is able to see other partipicants in the lobby.
Postcondition	- The User can play the game with other participants. - The User can make team selection. - The User can chating with other players. - The user can quit from the lobby.
Exceptions	3.1. User enters invalid ID. 3.2. System shows error message. 6.1. Database returns false to the system. 6.2. 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. - The user must have invited user's id or have it as a friend user.
Scenario	1. The user clicks invite button on lobby. 2. Pop-up shows up for select a friend or enter a user id. 3. The user selects user from friend list or enters a user id. 4. Lobby invite request shows up on other user.
Postcondition	The user which we send invite request should accept or reject invite from pop-up.
Exceptions	3.1. 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. - The user must be in a lobby.
Scenario	1. The user clicks ready button on lobby if the host is not itself. 2. The user clicks starts button on lobby if the host is itself. 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. - The user must be in a game. - The user must be alive.
Scenario	1. The user clicks shoot button. 2. System sends a message to the database to decrease user ammo. 3. Database decreases user ammo and sends current ammo information to the system. 4. System displays user ammo.
Postcondition	- If the user got hit downs below 0 health point , the system will count him as dead for specific time period. - 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	Chatiiing in lobby
Actor	User
Description	Chatting in the lobby allows users to communicate with each other and interact in the general chat before the game starts.
Precondition	- Players need to have logged into the application. - Users are required to have a game id and join the game lobby.
Scenario	1. The users enter the chat section within the lobby. 2. The user greets others or initiates a conversation in the general chat room. 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.

	- The user's communication and chat history become inaccessible once the game begins or the lobby is closed.
Exceptions	3.1. 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. - Users are required to have a game id and join the game lobby.
Scenario	4. The user is directed to the team selection screen. 5. From the database, users review available teams and team information. 6. The user selects their team. 7. The game automatically balances the teams and updates information of the team on database. 8. 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. - Once the team selection is completed, the chosen team information of users is displayed on the team lobby.
Exceptions	3.1. If a user makes an invalid selection on the team selection screen, the system sends an error message. 4.1. 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. - The User is required to have a game id.
Scenario	1. The user opens the in-game menu pop-up. 2. The user is directed to the weapon customization section from the in-game menu. 3. The user reviews the avaliable weapon options and costimization features. 4. The user selects appreance of their weapon. 5. The user saves and applies the chosen customizations. 6. The system closes the pop-up.
Postcondition	When weapon customization is completed, the changes made by players are saved.
Exceptions	4.1. If a user encounters a techical error during weapon customization, the system sends an error message. 5.1. 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. - The user must be in the game.
Scenario	1. User clicks menu/settings button. 2. In the opening menu, the user clicks to quit game button. 3. The user quit the game. 4. The System directs user to the main menu.

Postcondition	-
Exceptions	2.1. User clicks return button. 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. - The user must be in the game.
Scenario	1. User clicks reload ammo button. 2. The system sends a message to the database about refulling the users ammo with sending user ID. 3. The Database updates the record and sends a message to the system. 4. System updates amount of ammo information of user. 5. The user's ammo reloaded. 6. The user's ammo finishes, 7. System refilss user's ammo.
Postcondition	The user's ammo reloaded and ammo in maximum number.
Exceptions	6.1. 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. - The user must be in the game. - The user must be death.
Scenario	1. The System disables the reload and shoot buttons for 10 seconds. 2. The systems shows up a clock timer that represents remaining time for respawning. 3. The user wait 10 seconds. 4. The System enables the buttons. 5. User continues to play with full health and full ammo.
Postcondition	The users' scores updated.
Exceptions	3.1. When the user waits 10 seconds, if trying to use reload ammo or fire button, the system returns "You cannot 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. - Users are required to have a game id and join the game lobby.
Scenario	1. The user presses the "Show Menu" button. 2. The system opens a pop-up page. 3. The user selects one of the following options: Settings, Resume, Quit 4. The user presses the button of selected option. 5. The system closes the pop-up screen.
Postcondition	- The page is loaded according to the selection of the user. - 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. - If "Resume" button is pressed, then the user continues playing.

	- 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. - User are required to have a game id and join the game lobby. - The user must be in game.
Scenario	1. One of the users offer to surrender by clicking the “Surrender” button on the screen. 2. The system opens a pop-up page on each team member’s screen. 3. Rest of the team members vote whether to surrender or not. 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. - If not, the game continues.
Exceptions	3.1. 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.

	- The user should create a lobby.
Scenario	1. The system opens a pop-up including game rules after creating the lobby. 2. The user selects the rules of the game. 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

Use case Number	UC-21
Use case Name	User Got hit
Actor	User, Database, System
Description	The user loss health points if get shots
Precondition	Any user in the game must clicks the fire button.
Scenario	1. User clicks fire button. 2. System process AI component and detect there is a body. 3. System understands which user got hit from his location and distance. 4. System sends a message to the database about which user got hit. 5. Database decrease user health point and returns current health point. 6. System displays user’s new health point.
Postcondition	User died UC-22.
Exceptions	2.1 -System can not process AI component. 2.2- The health point of user does not decrease. 3.1- System can not get users location. 3.2- The health point of user does not decrease. 5.1- Database returns 0 for health point of user. 5.2 System understands that user has died.
Related Use Cases	UC-1, UC-2, UC-4, UC-5, UC-6, UC-11, UC-22

Use case Number	UC-22
Use case Name	User death
Actor	User, Database, System
Description	User health point becomes 0
Precondition	The user must got hit and his health point must be 0.
Scenario	1. Database returns user health point as 0. 2. System understands that user should die. 3. System displays a message about user’s death on user’s screen and disables buttons.
Postcondition	Respawning UC-17
Exceptions	-
Related Use Cases	UC-1, UC-2, UC-4, UC-5, UC-6, UC-11, UC-17, UC-21

5. Software Design Description

5.1. Introduction

5.1.1. Purpose

The purpose of this project is to develop an innovative real-world mobile multiplayer game that blends laser tag and paintball elements within a unique FPS experience. By leveraging cutting-edge technologies like TensorFlow Lite and Flutter, the aim is to create an immersive gaming environment that transcends traditional virtual setups. The focus on real-world scenarios and body segmentation for hit detection sets this game apart, promising an engaging and authentic player experience.

5.1.2. Scope

The scope of this project encompasses the development of a mobile FPS game that integrates seamlessly with the real world. The use of TensorFlow Lite for body segmentation eliminates the need for virtual environments, allowing players to engage in combat within their actual surroundings. The game will feature distinct body regions for hit detection, 3D weapon models with animations, and dynamic gameplay elements driven by player identification through GPS, phone orientation, and image distance estimation. Bluetooth controllers further enhance the gaming experience by enabling physical interactions.

5.1.3. Definitions And Acronyms, Abbreviations

Term	Definition
Activity Diagram	A flowchart showing how one activity leads to another activity.
Sequence Diagram	Shows process interactions in the area of software engineering that are time-ordered.
Class Diagram	A graphical notation for building and visualizing object-oriented systems.
Software Design Description	A description of software created to facilitate analysis, planning, implementation, and decision-making.
User Interface Design	The technique designers use to construct interfaces in software or technological devices, focused on appearances or style.
Android	An open source and free mobile operating system based on Linux, developed for use on mobile devices and mobile phones.
iOS	Apple's mobile operating system originally developed for the iPhone but later used on the iPod touch and iPad.
User	Someone that uses a product or service.
Application	

Admin	A person in charge of administration in a business or organization.
Database	A structured collection of information or data that is often saved electronically in a computer system.
Segmentation	The process of dividing a larger entity or dataset into smaller, more manageable parts, often used in networking or data analysis.
Lobby	A virtual waiting area where users gather before engaging in the main activity, common in online games and chat applications.
Screenshot	A captured image of the current display on a computer or mobile device, widely used for documentation and sharing visual information.
Health point	(HP) represent the vitality of a character or system in a game or application, indicating remaining life or well-being.
Algorithm	A step-by-step set of rules designed to perform specific tasks or solve problems, foundational in various IT processes.
Surrender	In online gaming or applications, is the voluntary act of ending a game or task before its natural conclusion.
Pop-up	A sudden graphical user interface element that appears on top of current content, commonly used to display information or options.
Host	A computer or device providing resources or services to connected devices, such as a server hosting a website or game session.

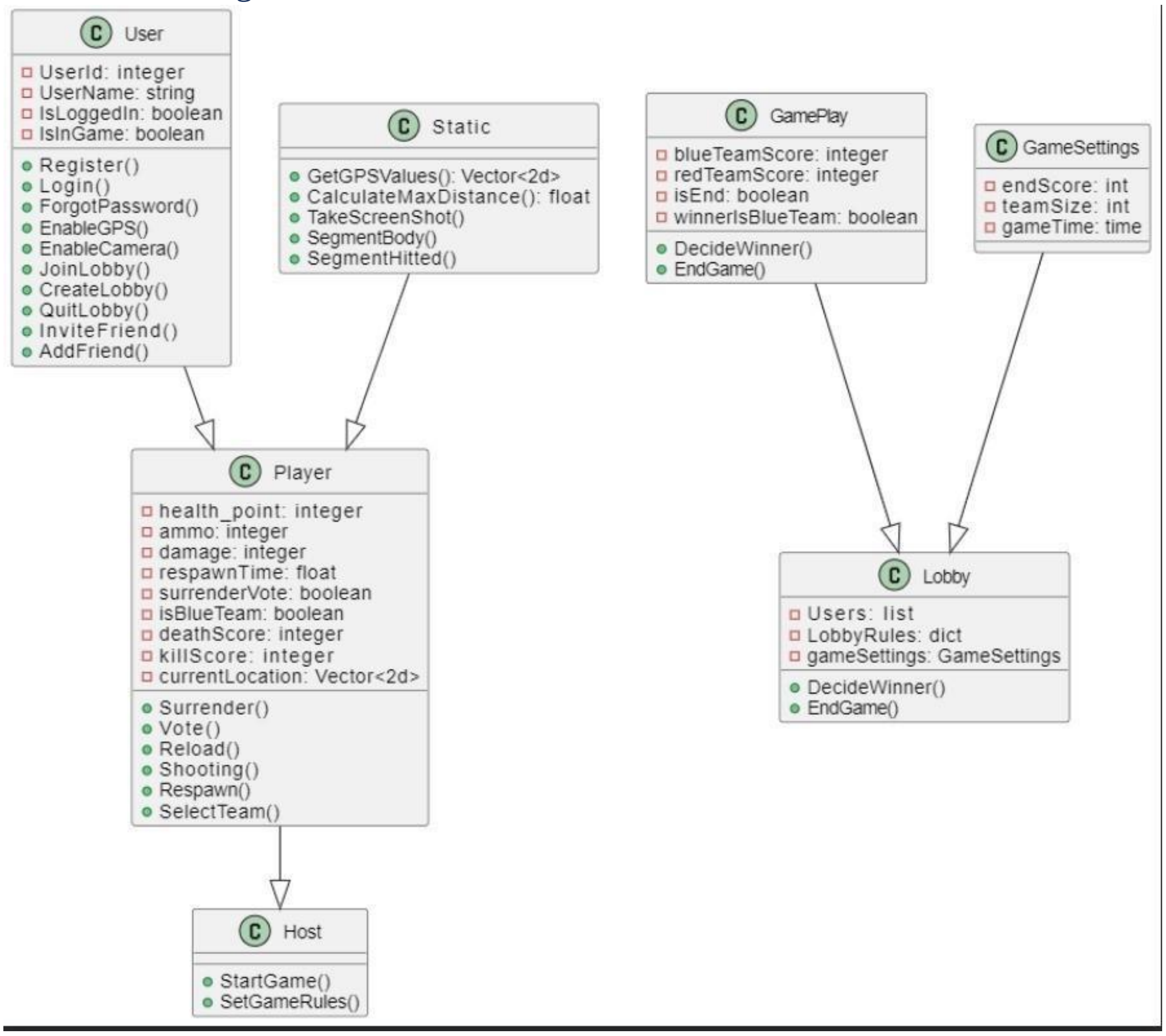
5.1.4. Motivation

The motivation behind this project is to redefine the mobile gaming experience by merging real-world elements with FPS gameplay. Traditional virtual environments can be limiting, and this project seeks to break free from those constraints. The use of TensorFlow Lite for body segmentation provides a novel approach to hit detection, while the incorporation of Bluetooth controllers adds a tangible and immersive layer to the gaming interaction. The desire is to create a game that not only entertains but also pushes the boundaries of what is possible in mobile gaming, offering players an experience that bridges the gap between the virtual and the real.

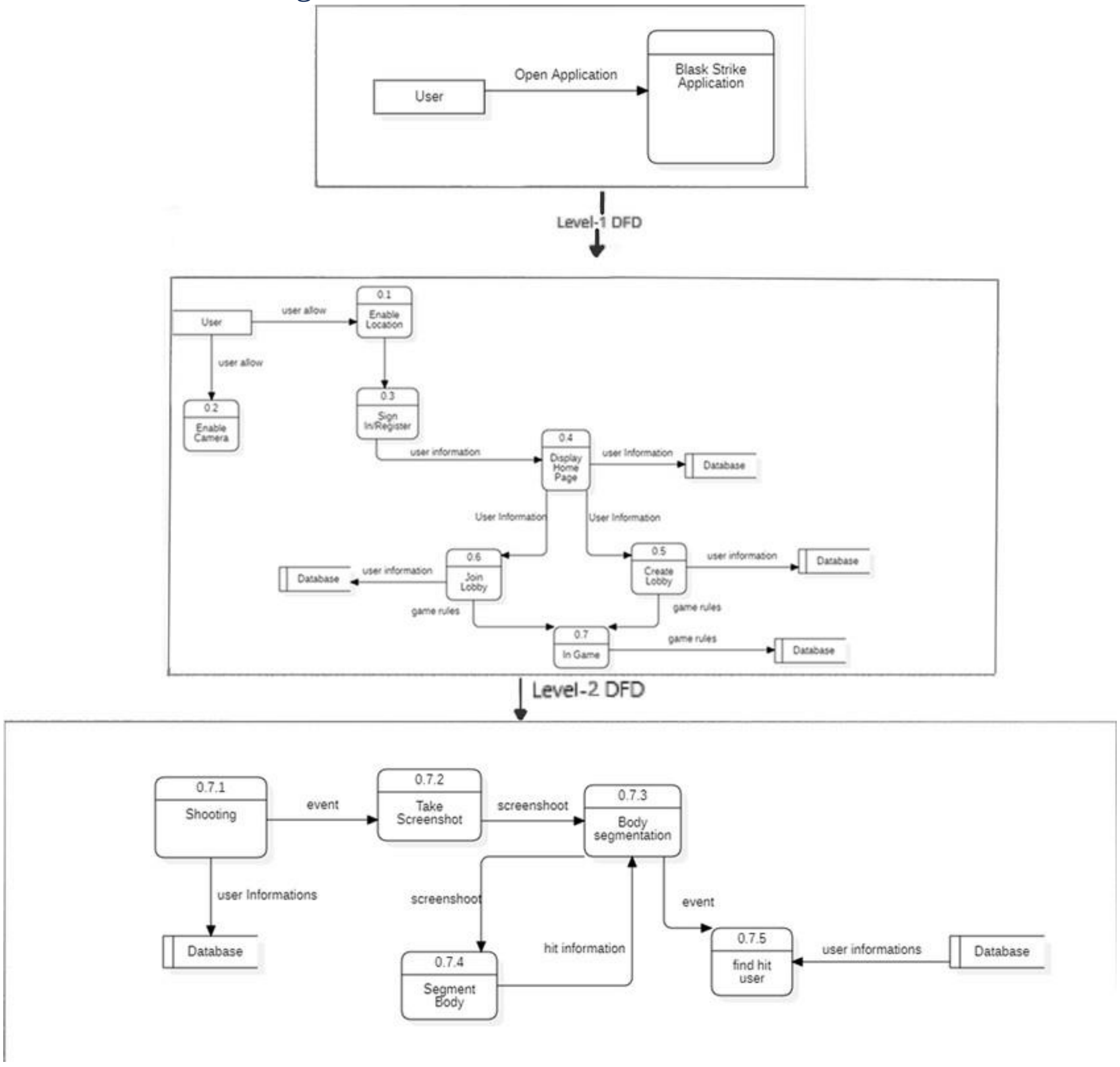
5.2. Design Approach

This section of the system design includes the system's architectural design, the definition of the problem, the technologies used, user interface design. It also includes diagrams such as Class Diagram, Activity Diagram, Sequence Diagram and Data Flow Diagram.

5.2.1. Class Diagram

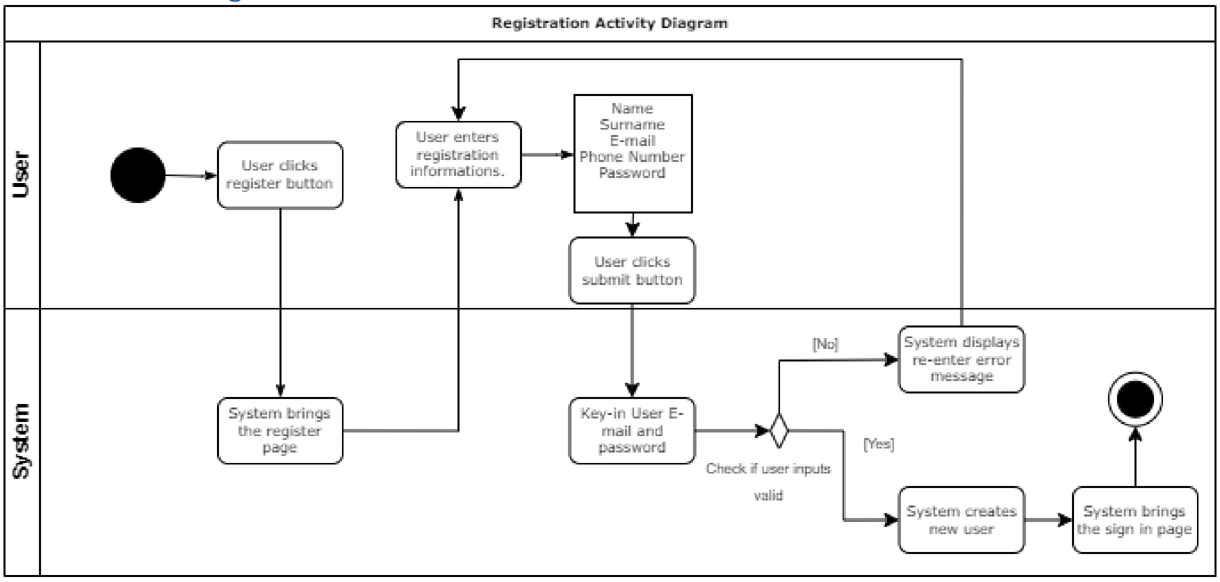


5.2.2. Data Flow Diagram

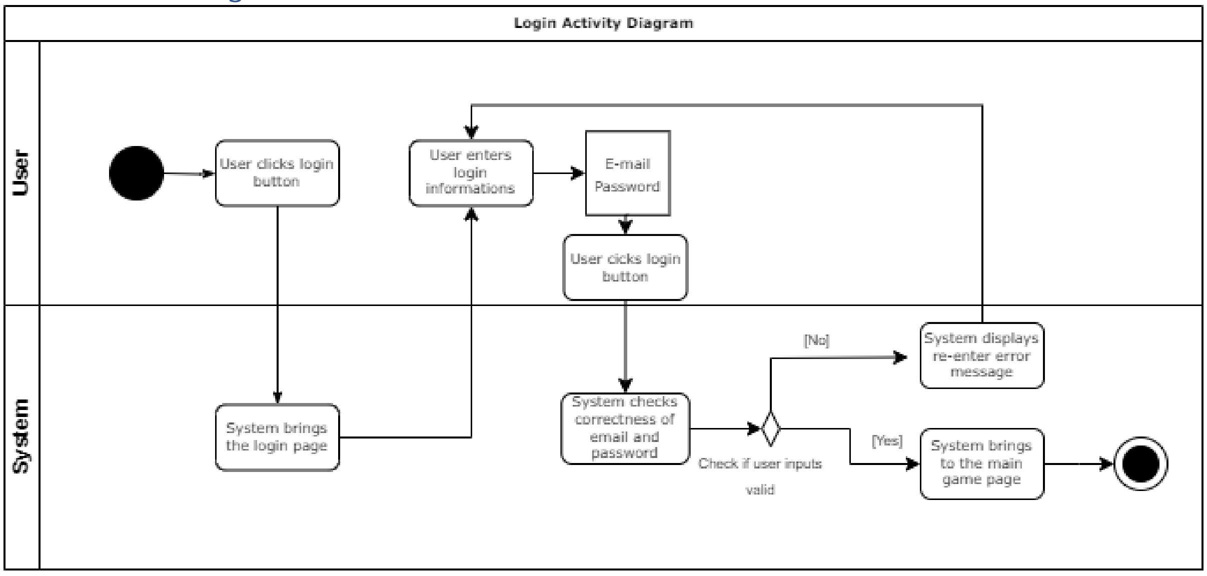


5.2.3. Activity Diagrams

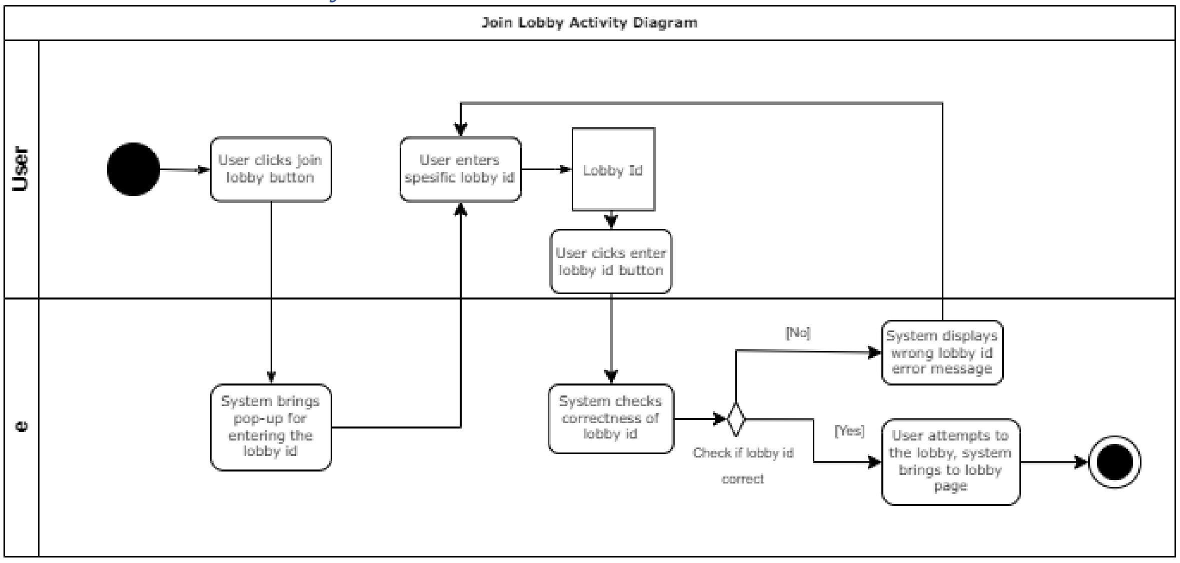
5.2.3.1. Register


Registration Activity Diagram
User
System
User clicks register button
User enters registration informations.
Name
Surname
E-mail
Phone Number
Password
User clicks submit button
System brings the register page
Key-in User E-mail and password
Check if user inputs valid
[No]
[Yes]
System displays re-enter error message
System creates new user
System brings the sign in page

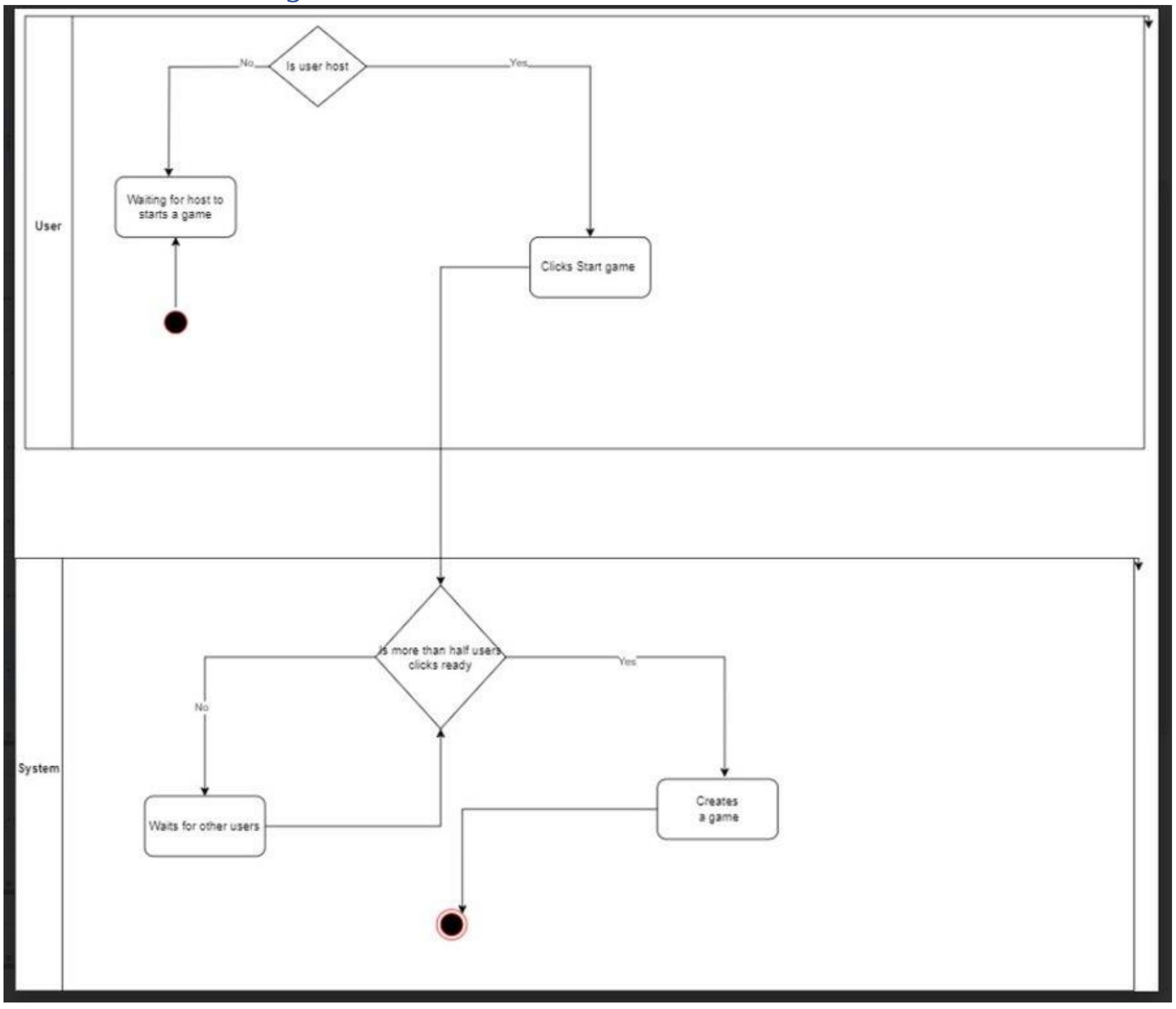
5.2.3.2. Login


Login Activity Diagram
User
System
User clicks login button
User enters login informations
E-mail
Password
User clicks login button
System brings the login page
System checks correctness of email and password
Check if user inputs valid
[No]
[Yes]
System displays re-enter error message
System brings to the main game page

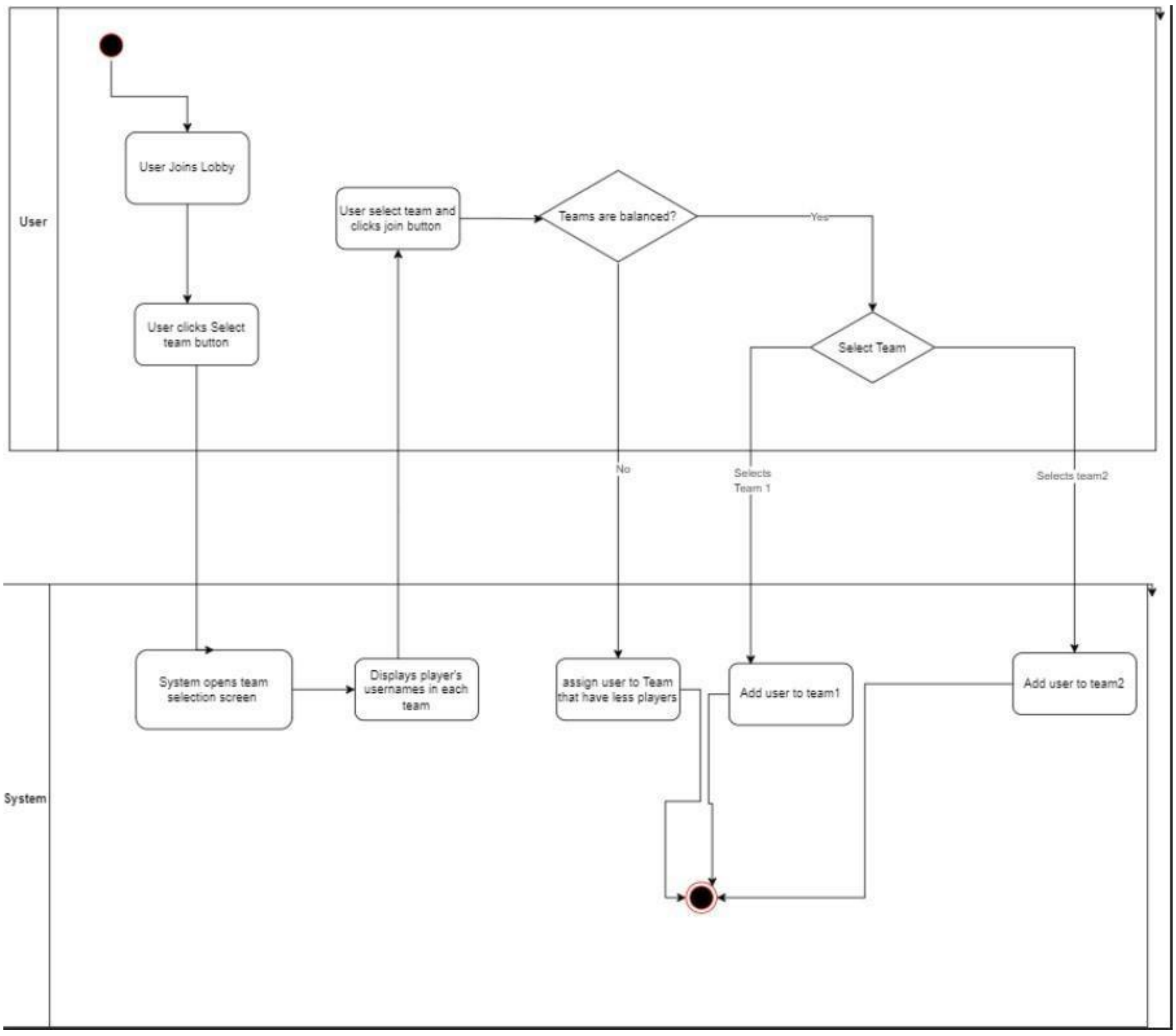
5.2.3.3. Join Lobby


Join Lobby Activity Diagram
User
e
User clicks join lobby button
User enters spesific lobby id
Lobby Id
User cicks enter lobby id button
System brings pop-up for entering the lobby id
System checks correctness of lobby id
Check if lobby id correct
[No]
System displays wrong lobby id error message
[Yes]
User attempts to the lobby, system brings to lobby page

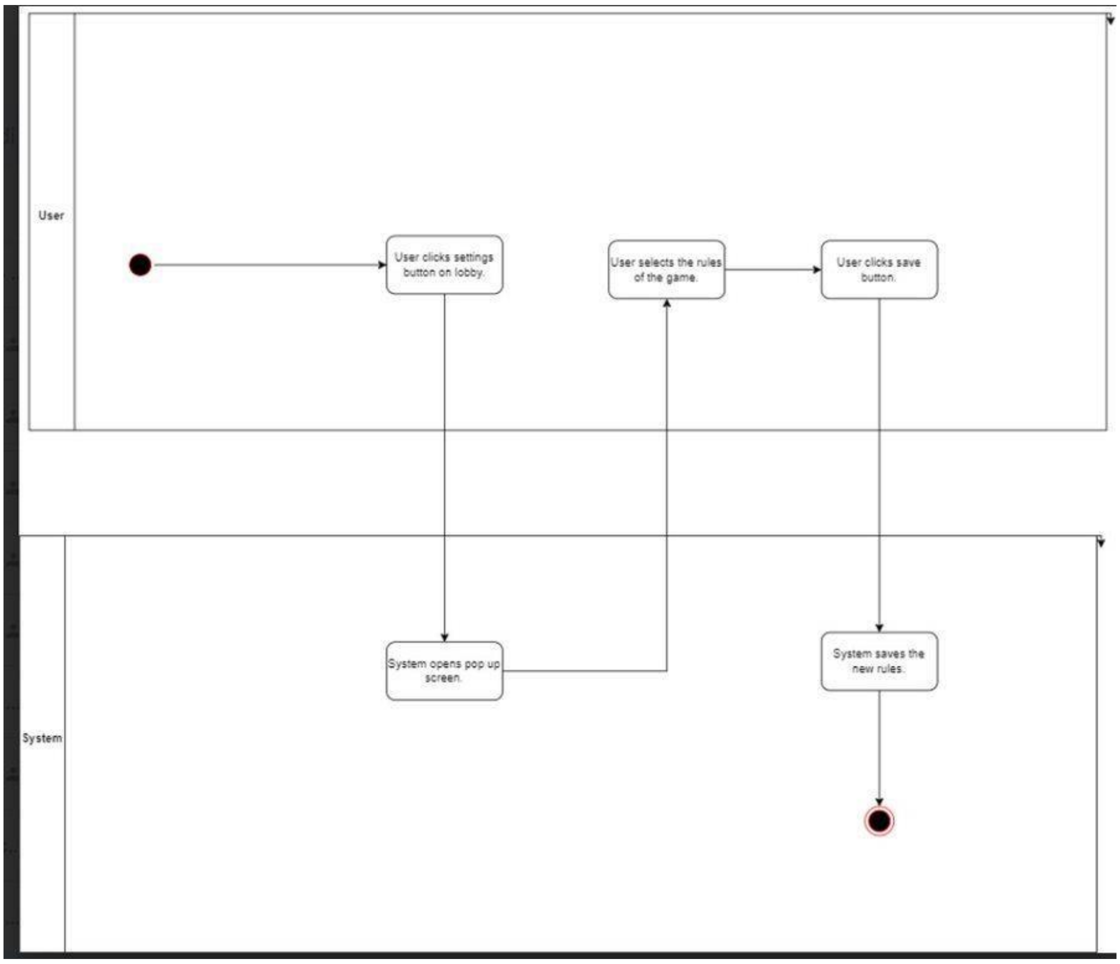
5.2.3.4. Starting Game



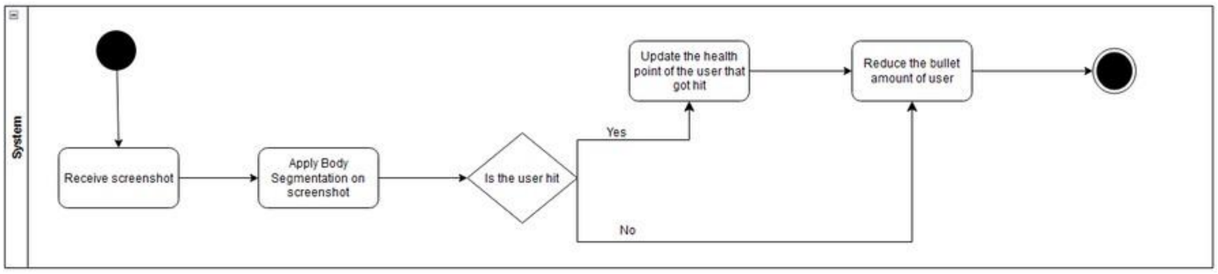
5.2.3.5. *Select Team*


User
User Joins Lobby
User clicks Select team button
User select team and clicks join button
Teams are balanced?
Yes
Select Team
No
Selects Team 1
Selects team2
System
System opens team selection screen
Displays player's usernames in each team
assign user to Team that have less players
Add user to team1
Add user to team2

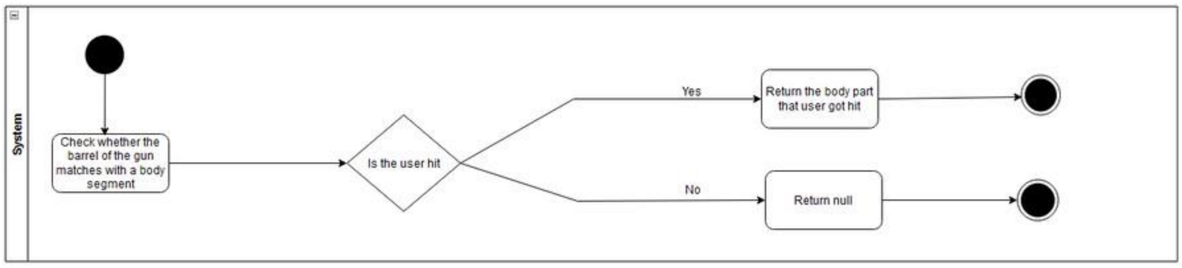
5.2.3.6. *Set Game Rules*


User
User clicks settings button on lobby.
User selects the rules of the game.
User clicks save button.
System
System opens pop up screen.
System saves the new rules.

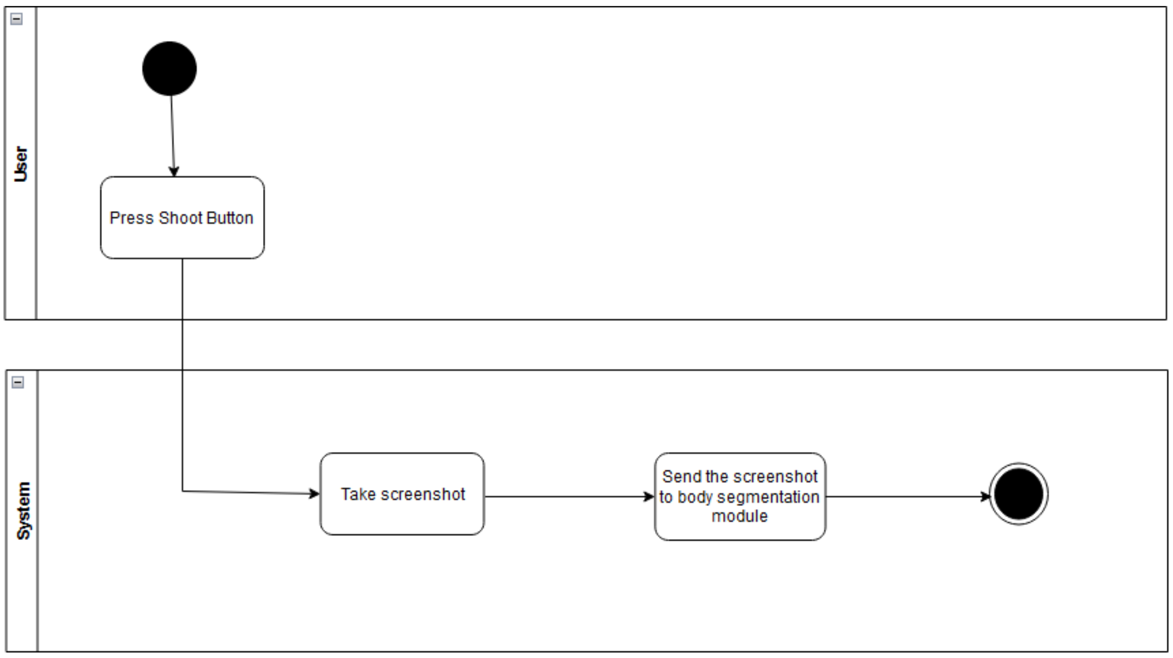
5.2.3.7. *Segment Body*


System
Receive screenshot
Apply Body Segmentation on screenshot
Is the user hit
Yes
No
Update the health point of the user that got hit
Reduce the bullet amount of user

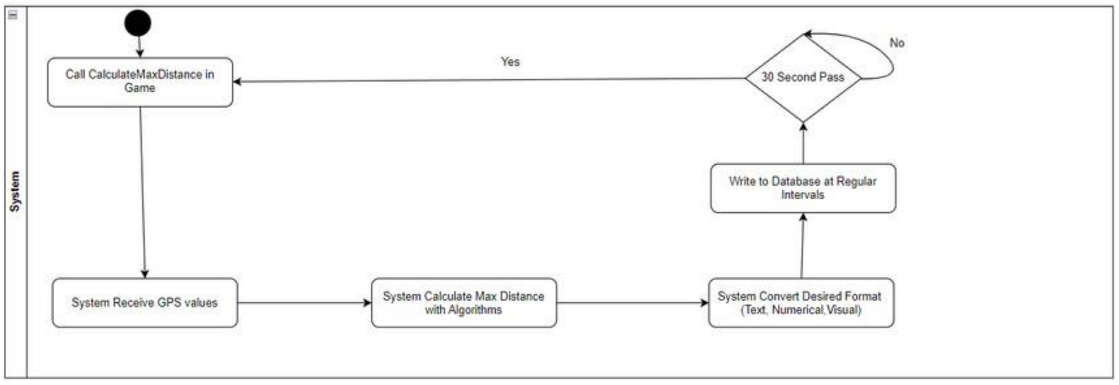
5.2.3.8. *Segment Hit*


System
Check whether the barrel of the gun matches with a body segment
Is the user hit
Yes
No
Return the body part that user got hit
Return null

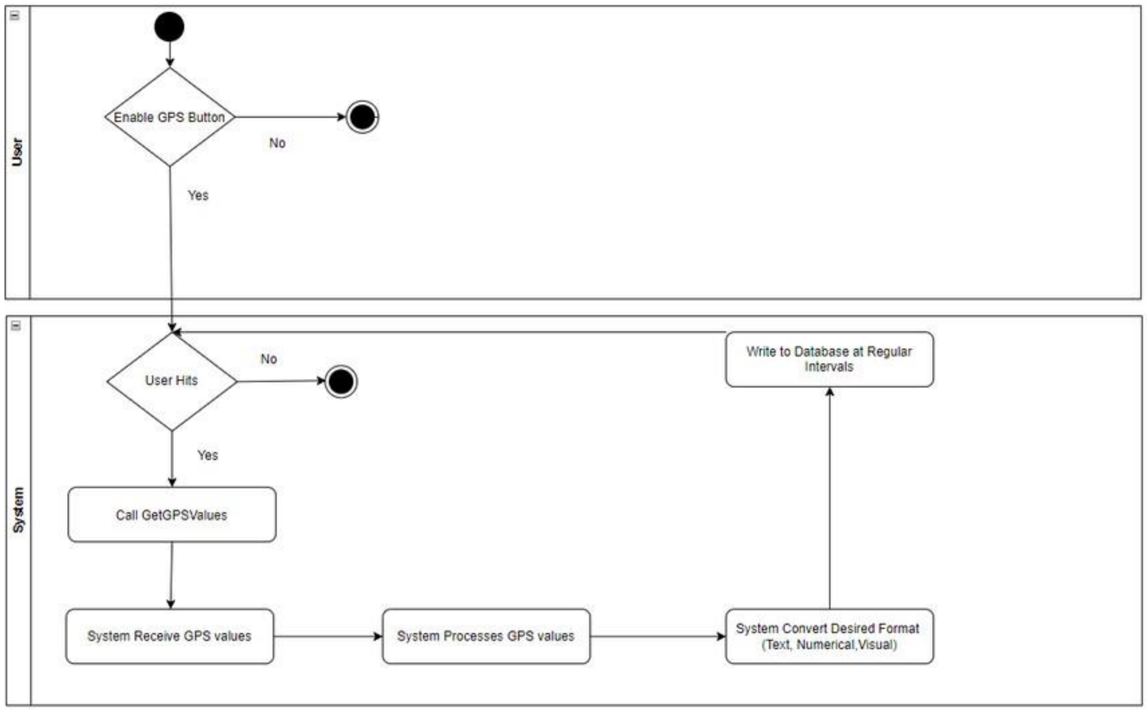
5.2.3.9. *Take Screenshot*



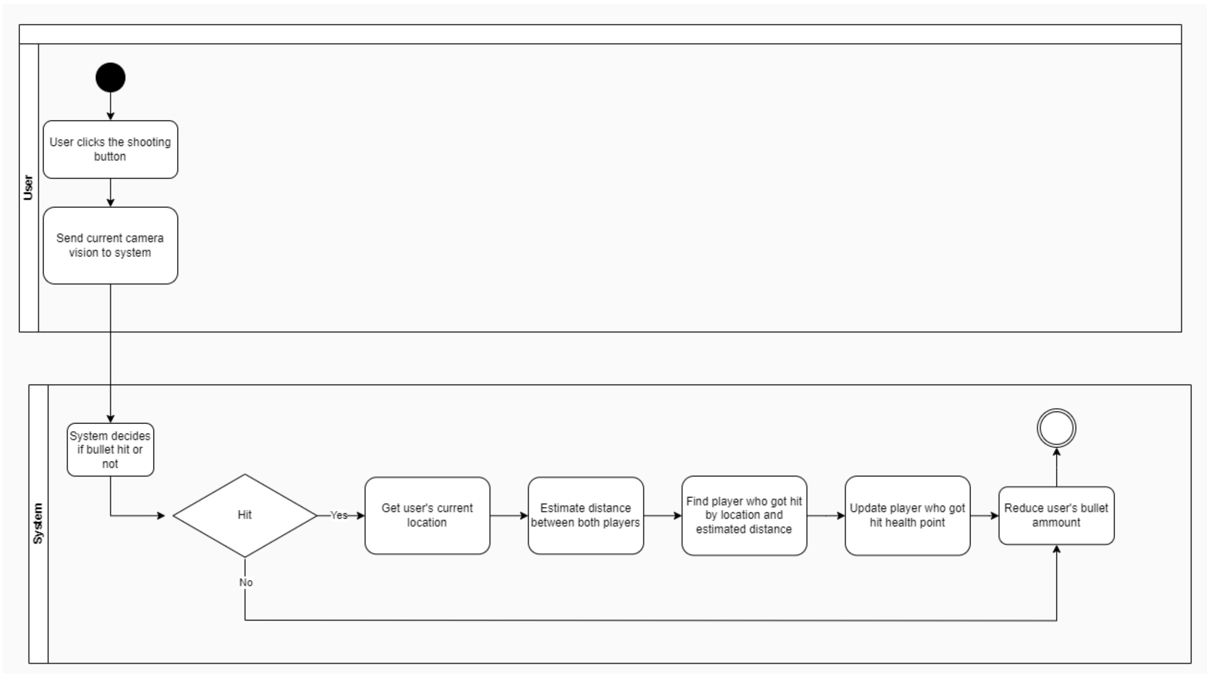
5.2.3.10. *Calculate Max Distance*


System
Call CalculateMaxDistance in Game
Yes
No
30 Second Pass
Write to Database at Regular Intervals
System Receive GPS values
System Calculate Max Distance with Algorithms
System Convert Desired Format (Text, Numerical,Visual)

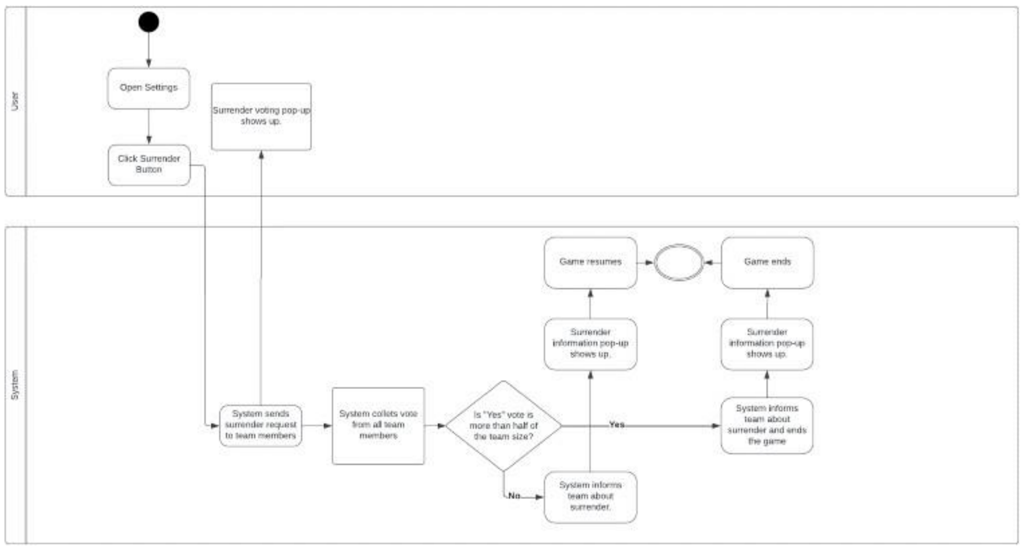
5.2.3.11. Get GPS Values


User
Enable GPS Button
No
Yes
System
User Hits
No
Yes
Write to Database at Regular Intervals
Call GetGPSValues
System Receive GPS values
System Processes GPS values
System Convert Desired Format (Text, Numerical,Visual)

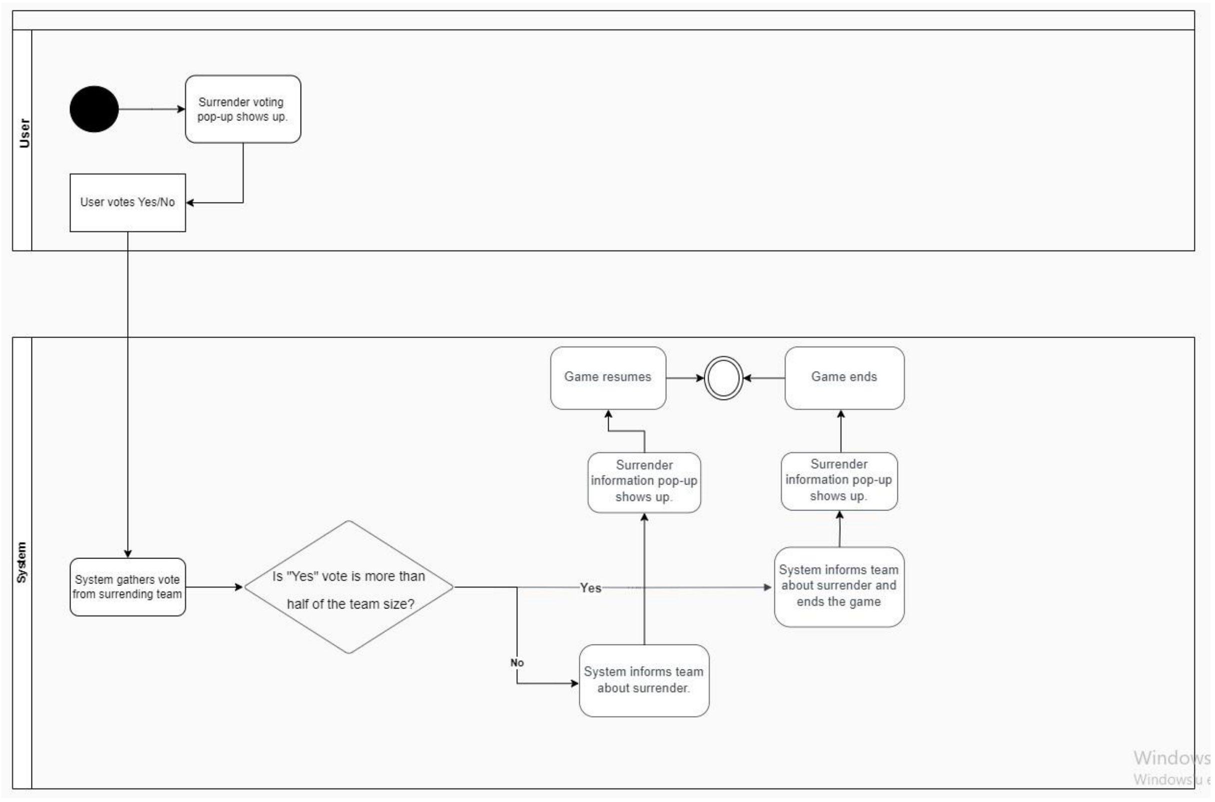
5.2.3.12. Hit


User
User clicks the shooting button
Send current camera vision to system
System
System decides if bullet hit or not
Hit
Yes
No
Get user's current location
Estimate distance between both players
Find player who got hit by location and estimated distance
Update player who got hit health point
Reduce user's bullet ammount

5.2.3.13. Surrender

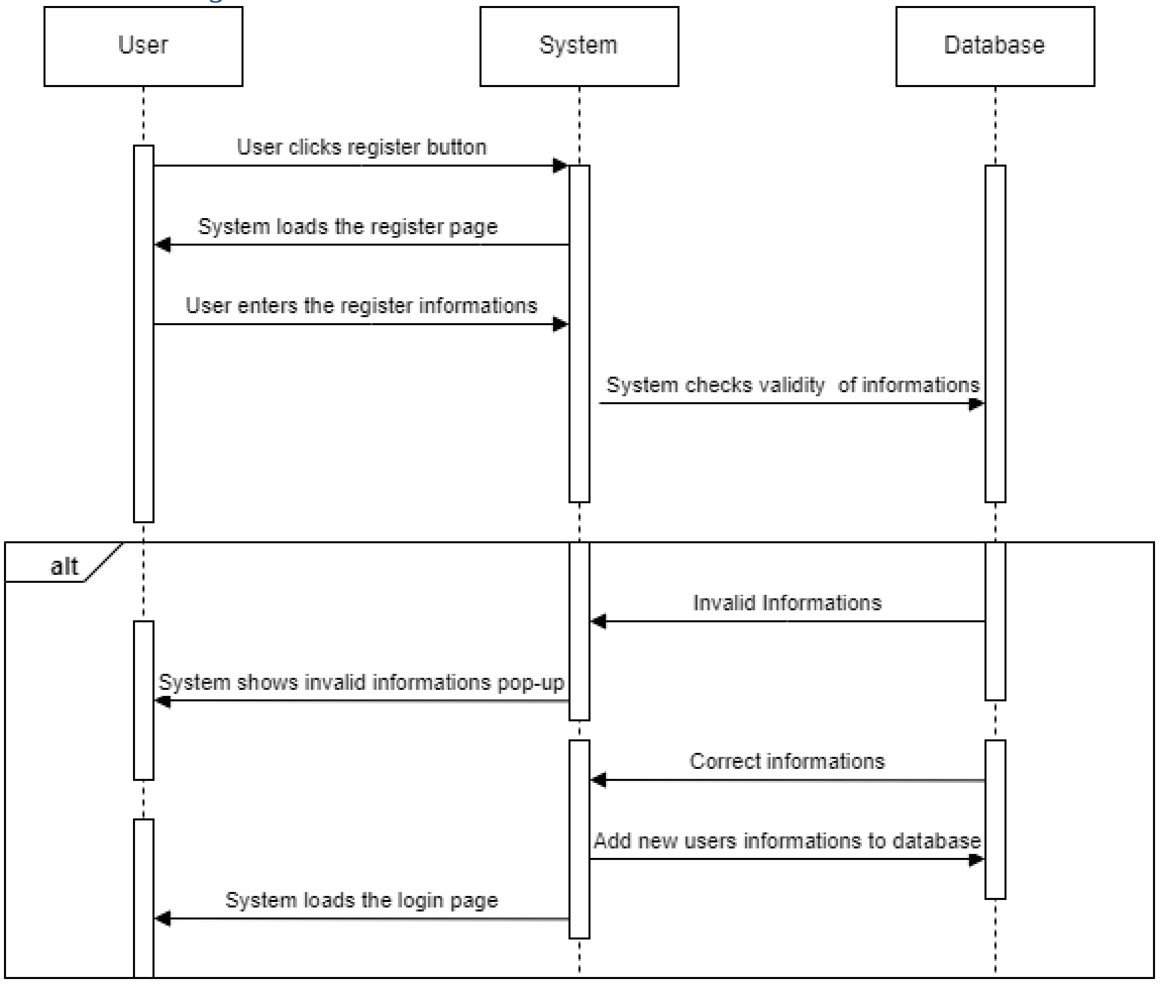

User
Open Settings
Click Surrender Button
Surrender voting pop-up shows up.
System
System sends surrender request to team members
System collets vote from all team members
Is "Yes" vote is more than half of the team size?
Yes
No
System informs team about surrender.
System informs team about surrender and ends the game
Surrender information pop-up shows up.
Surrender information pop-up shows up.
Game resumes
Game ends

5.2.3.14. Vote

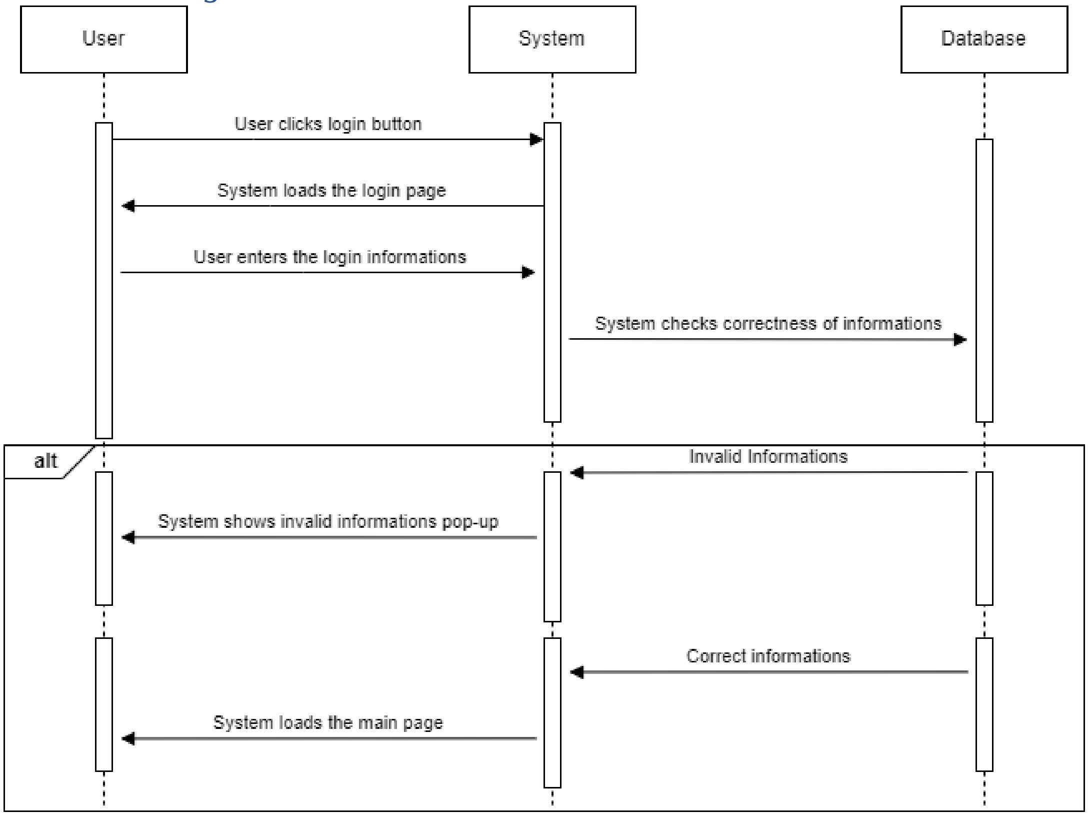

User
Surrender voting pop-up shows up.
User votes Yes/No
System
System gathers vote from surrending team
Is "Yes" vote is more than half of the team size?
Yes
No
System informs team about surrender.
System informs team about surrender and ends the game
Surrender information pop-up shows up.
Surrender information pop-up shows up.
Game resumes
Game ends

5.2.4. Sequence Diagrams

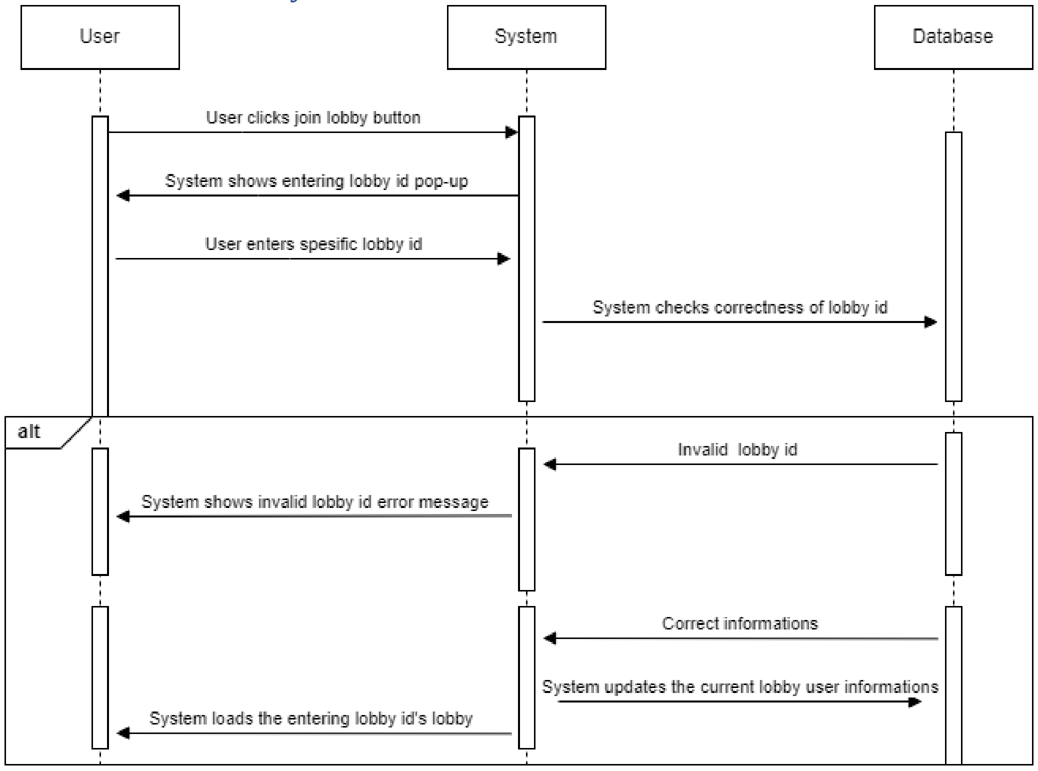
5.2.4.1. Register


User
System
Database
User clicks register button
System loads the register page
User enters the register informations
System checks validity of informations
alt
Invalid Informations
System shows invalid informations pop-up
Correct informations
Add new users informations to database
System loads the login page

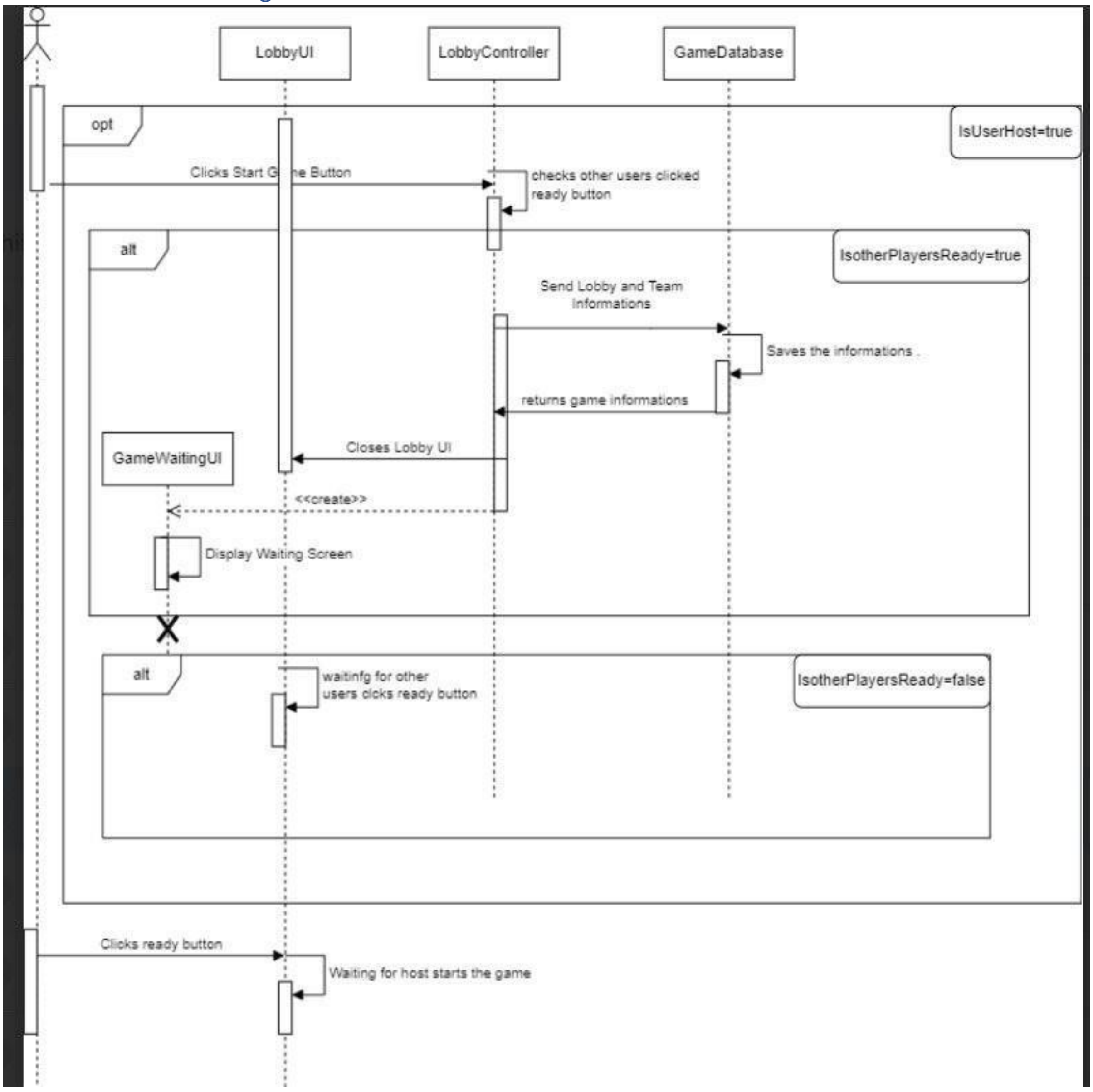
5.2.4.2. Login


User
System
Database
User clicks login button
System loads the login page
User enters the login informations
System checks correctness of informations
alt
Invalid Informations
System shows invalid informations pop-up
Correct informations
System loads the main page

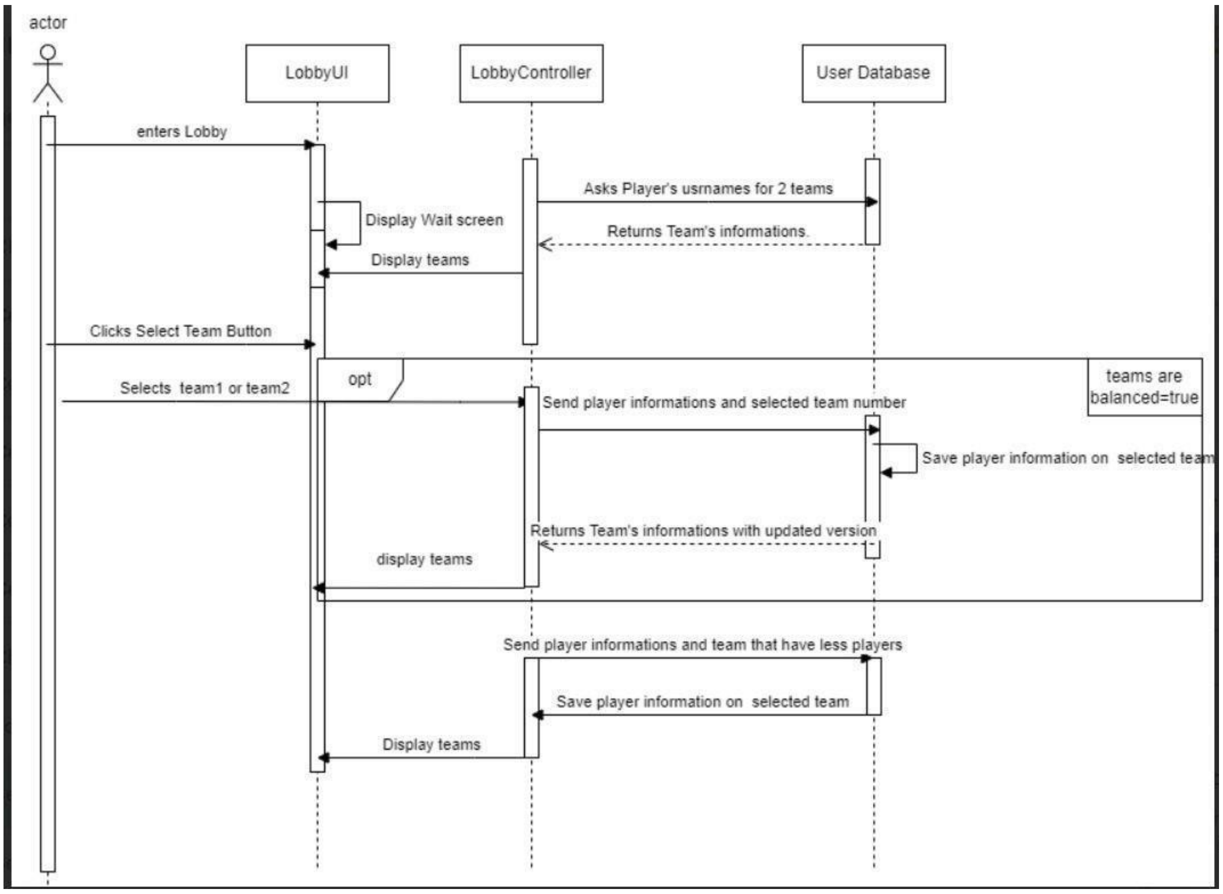
5.2.4.3. Join Lobby


User
System
Database
User clicks join lobby button
System shows entering lobby id pop-up
User enters spesific lobby id
System checks correctness of lobby id
alt
Invalid lobby id
System shows invalid lobby id error message
Correct informations
System updates the current lobby user informations
System loads the entering lobby id's lobby

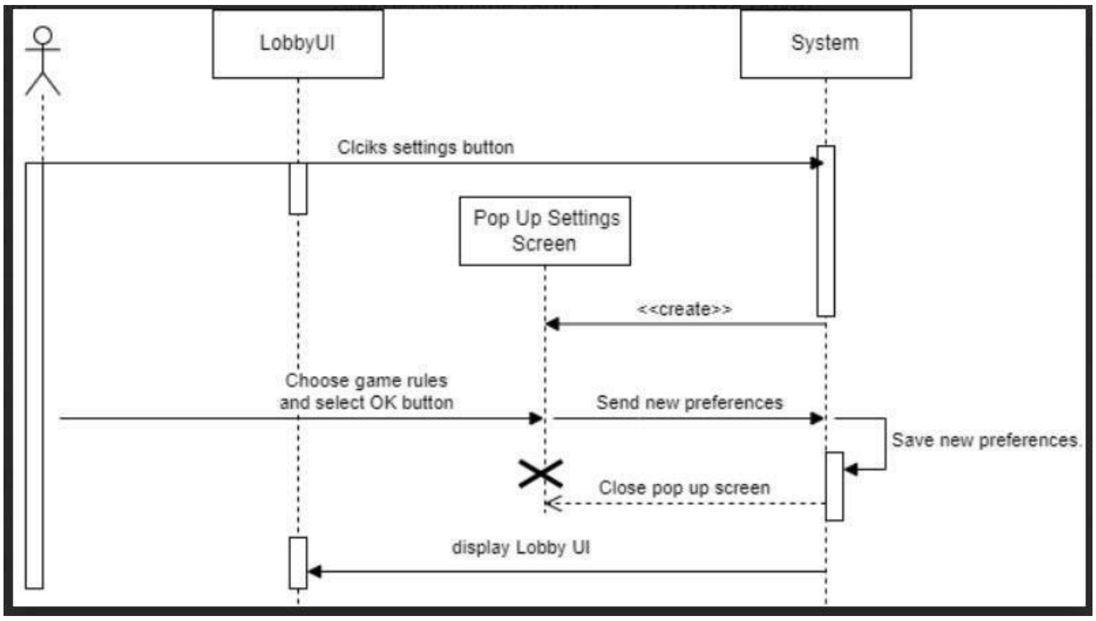
5.2.4.4. Starting Game



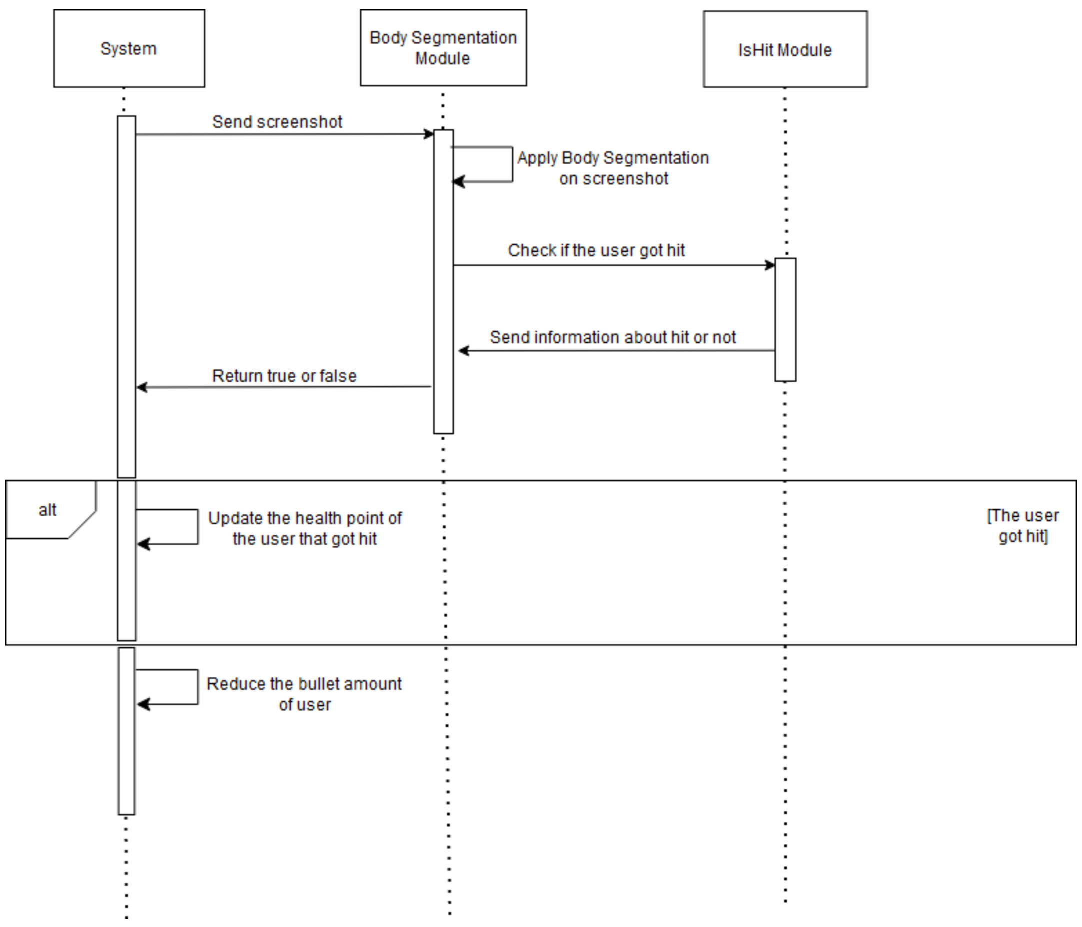
5.2.4.5. Select Team


actor
LobbyUI
LobbyController
User Database
enters Lobby
Asks Player's usrnames for 2 teams
Display Wait screen
Returns Team's informations.
Display teams
Clicks Select Team Button
opt
teams are balanced=true
Selects team1 or team2
Send player informations and selected team number
Save player information on selected team
Returns Team's informations with updated version
display teams
Send player informations and team that have less players
Save player information on selected team
Display teams

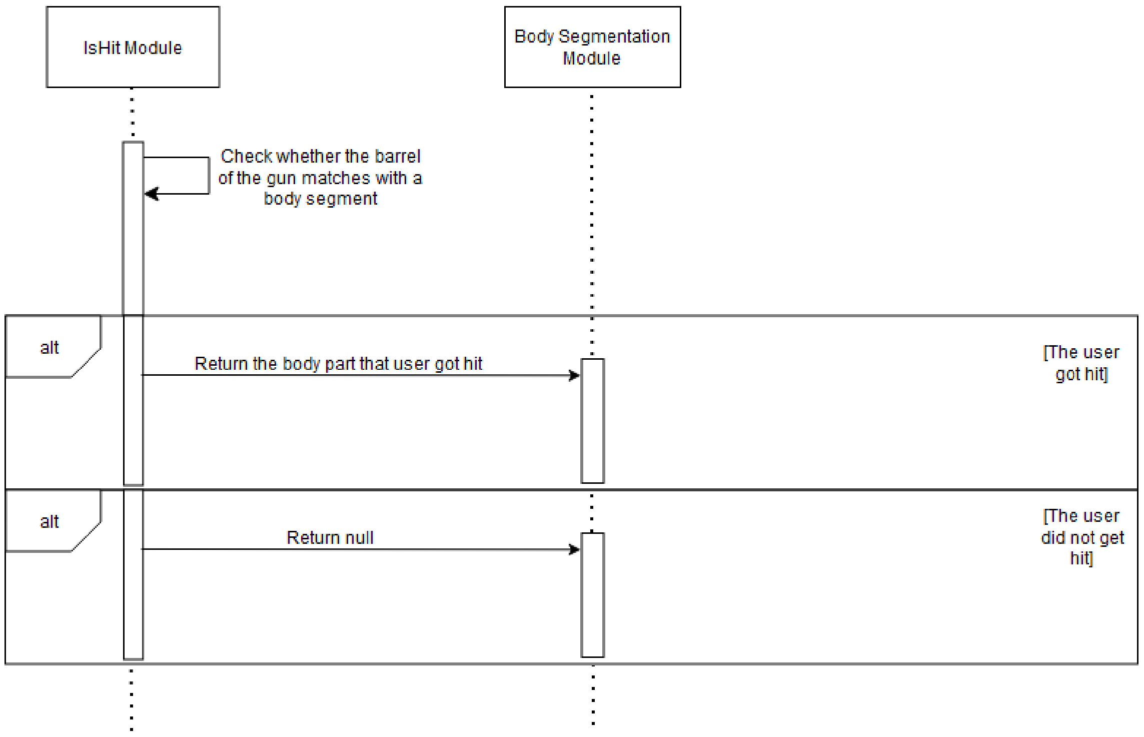
5.2.4.6. Set Game Rules



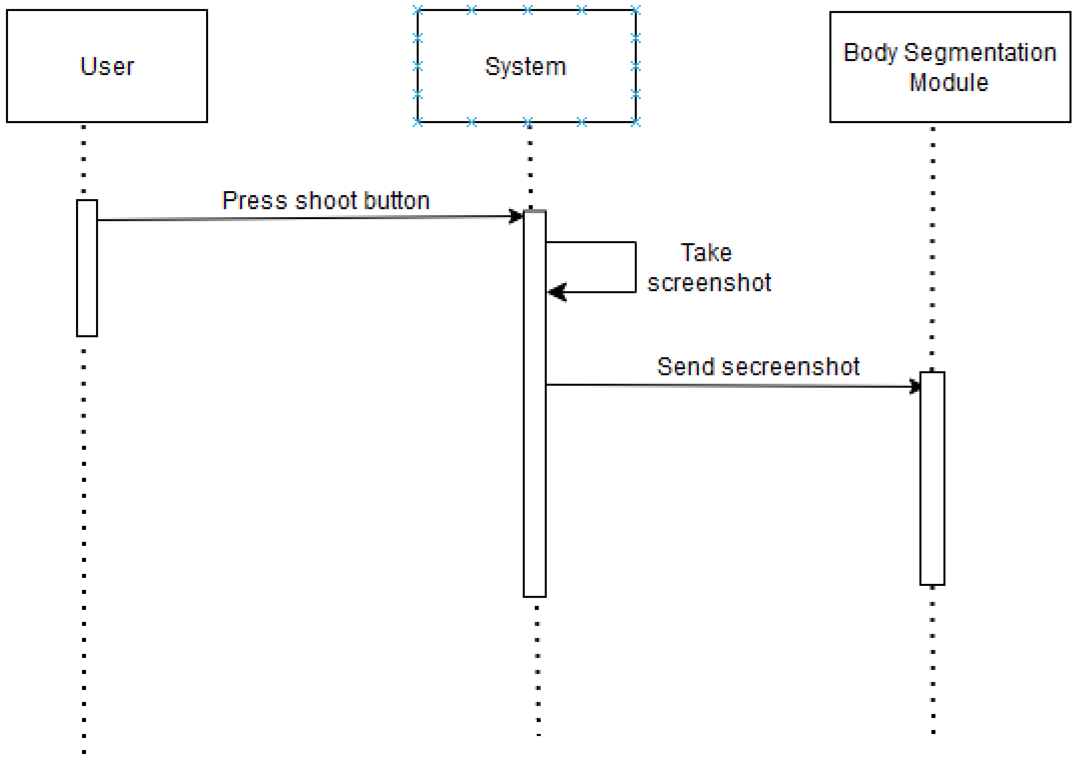
5.2.4.7. Segment Body


System
Body Segmentation Module
IsHit Module
Send screenshot
Apply Body Segmentation on screenshot
Check if the user got hit
Send information about hit or not
Return true or false
alt
Update the health point of the user that got hit
[The user got hit]
Reduce the bullet amount of user

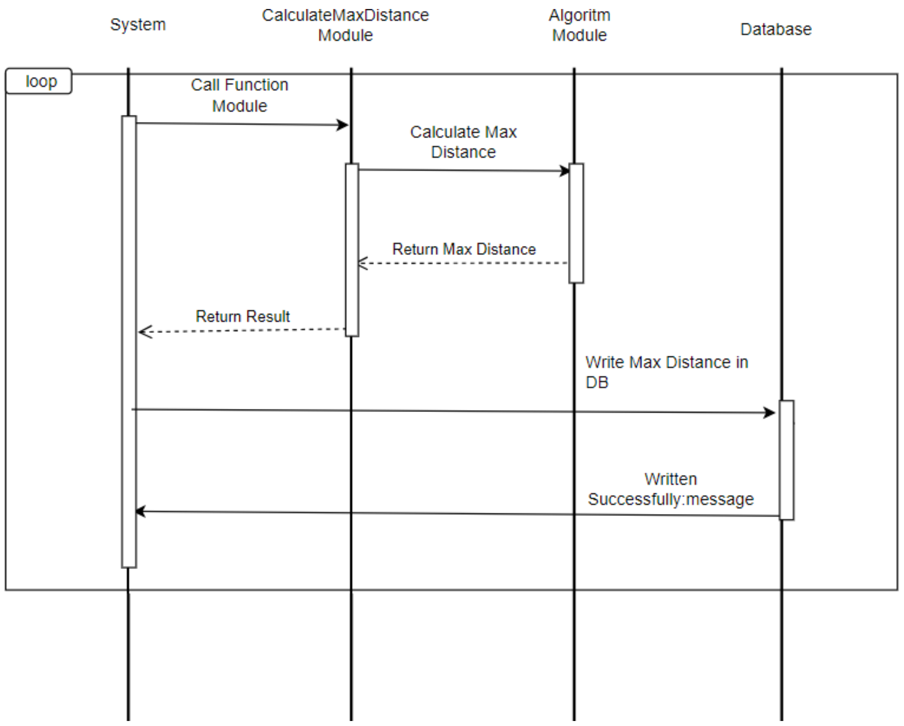
5.2.4.8. Segment Hit


IsHit Module
Body Segmentation Module
Check whether the barrel of the gun matches with a body segment
alt
Return the body part that user got hit
[The user got hit]
alt
Return null
[The user did not get hit]

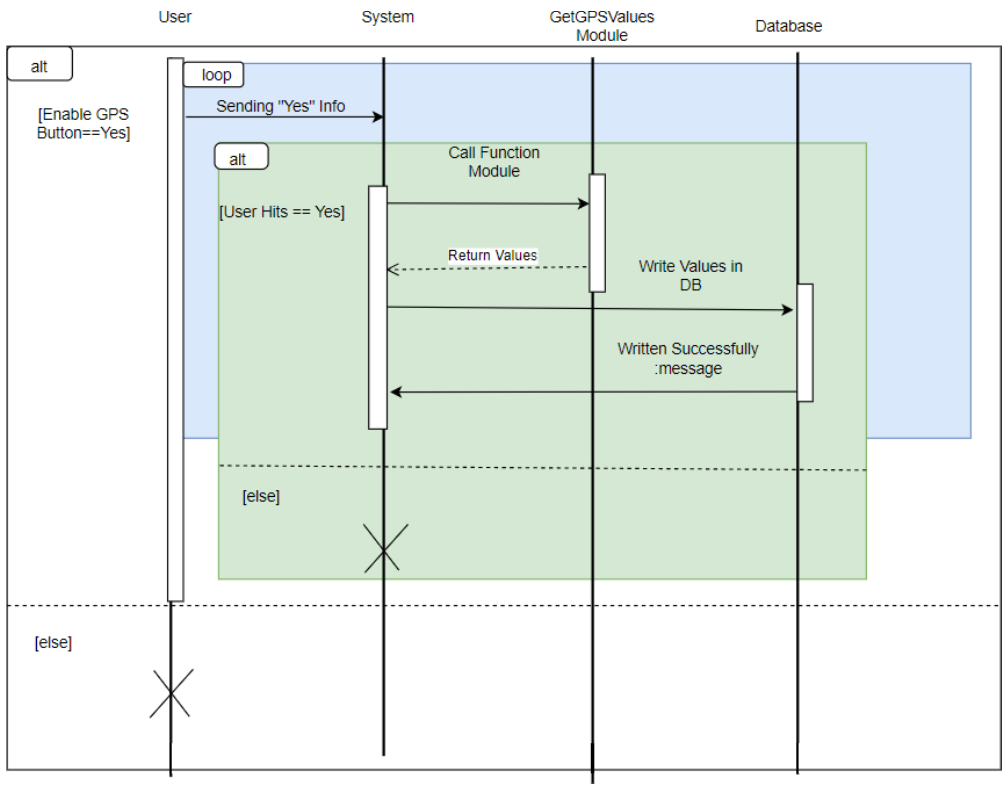
5.2.4.9. *Take Screenshot*



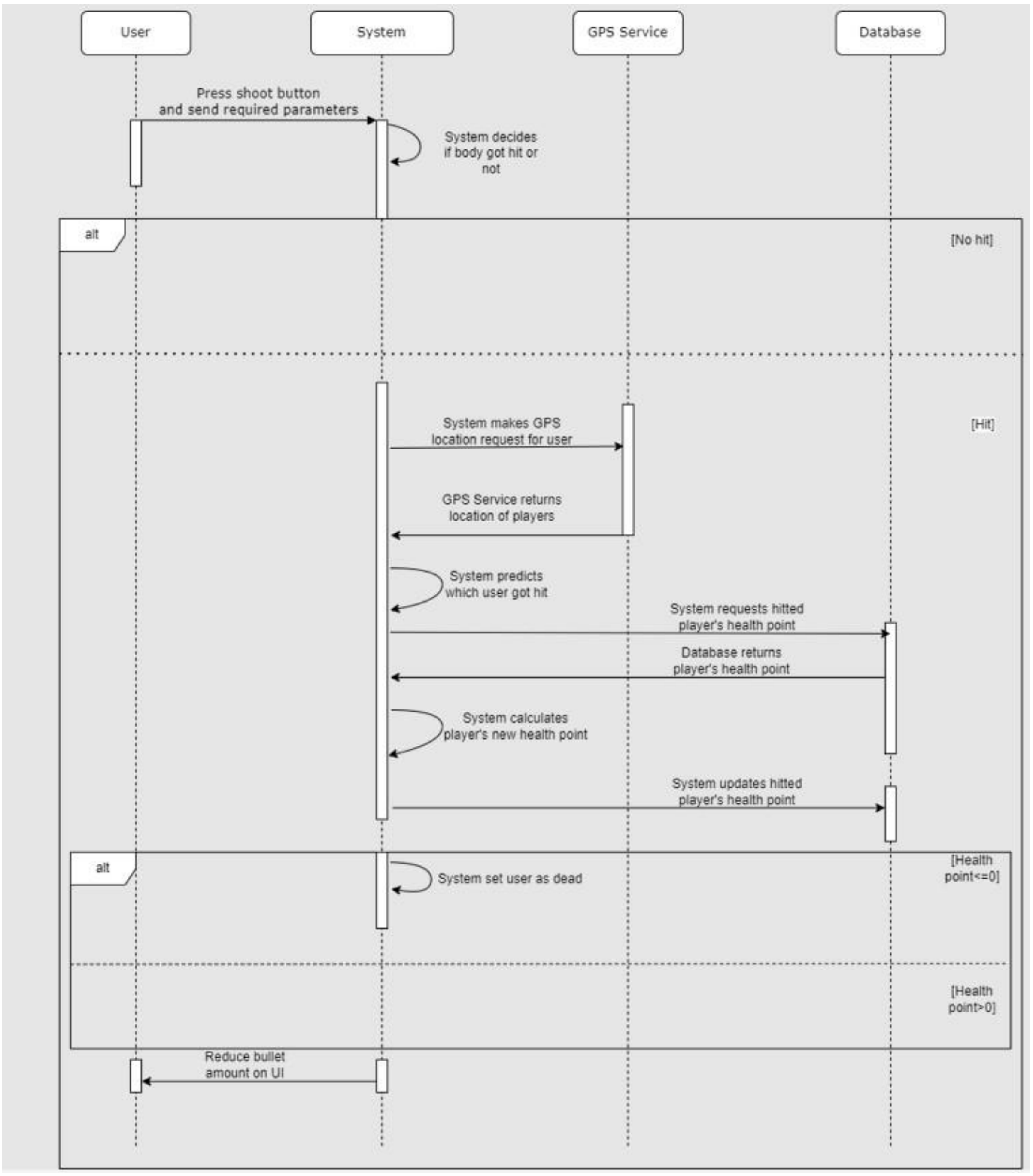
5.2.4.10. *Calculate Max Distance*



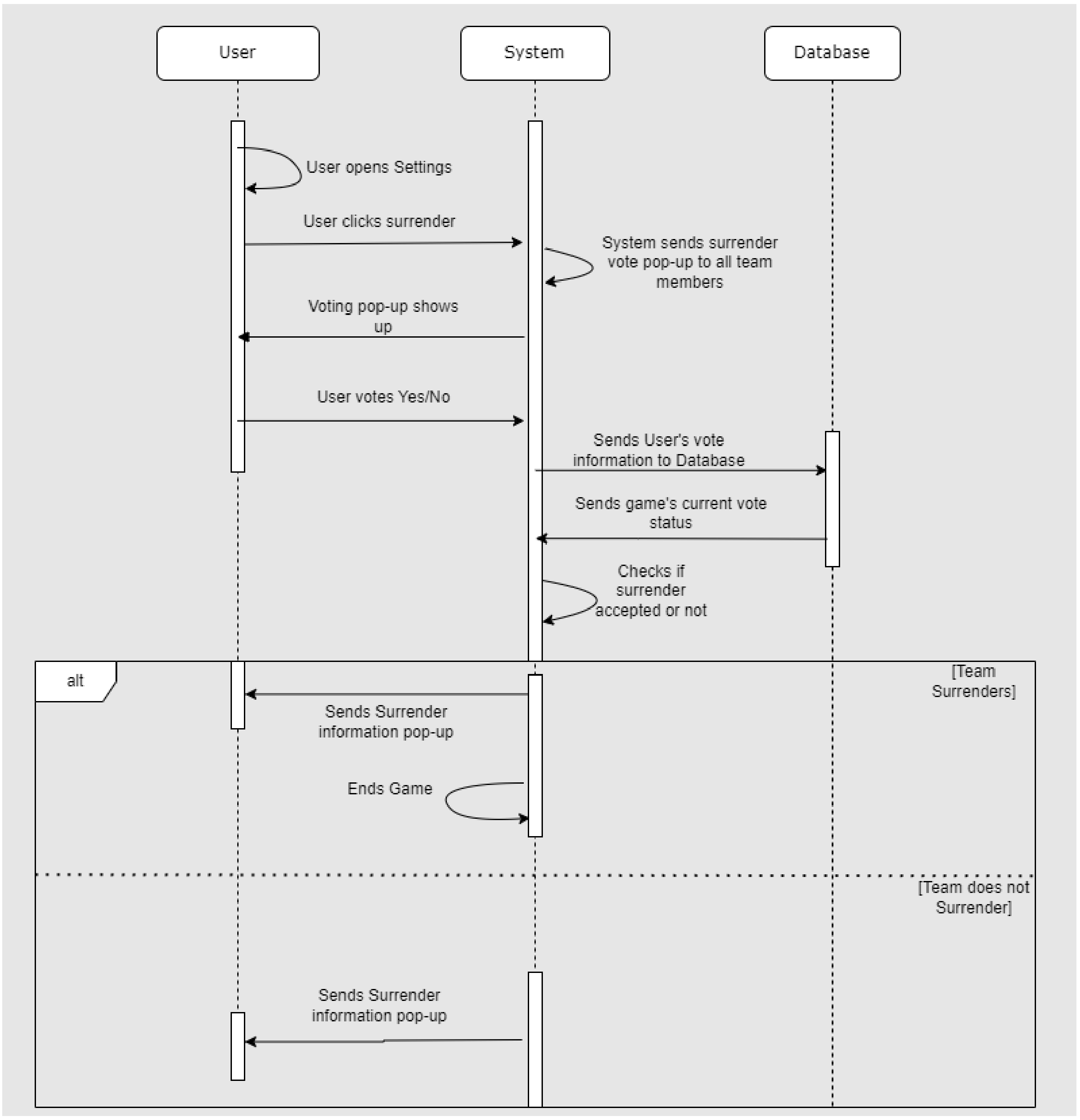
5.2.4.11. Get GPS Values



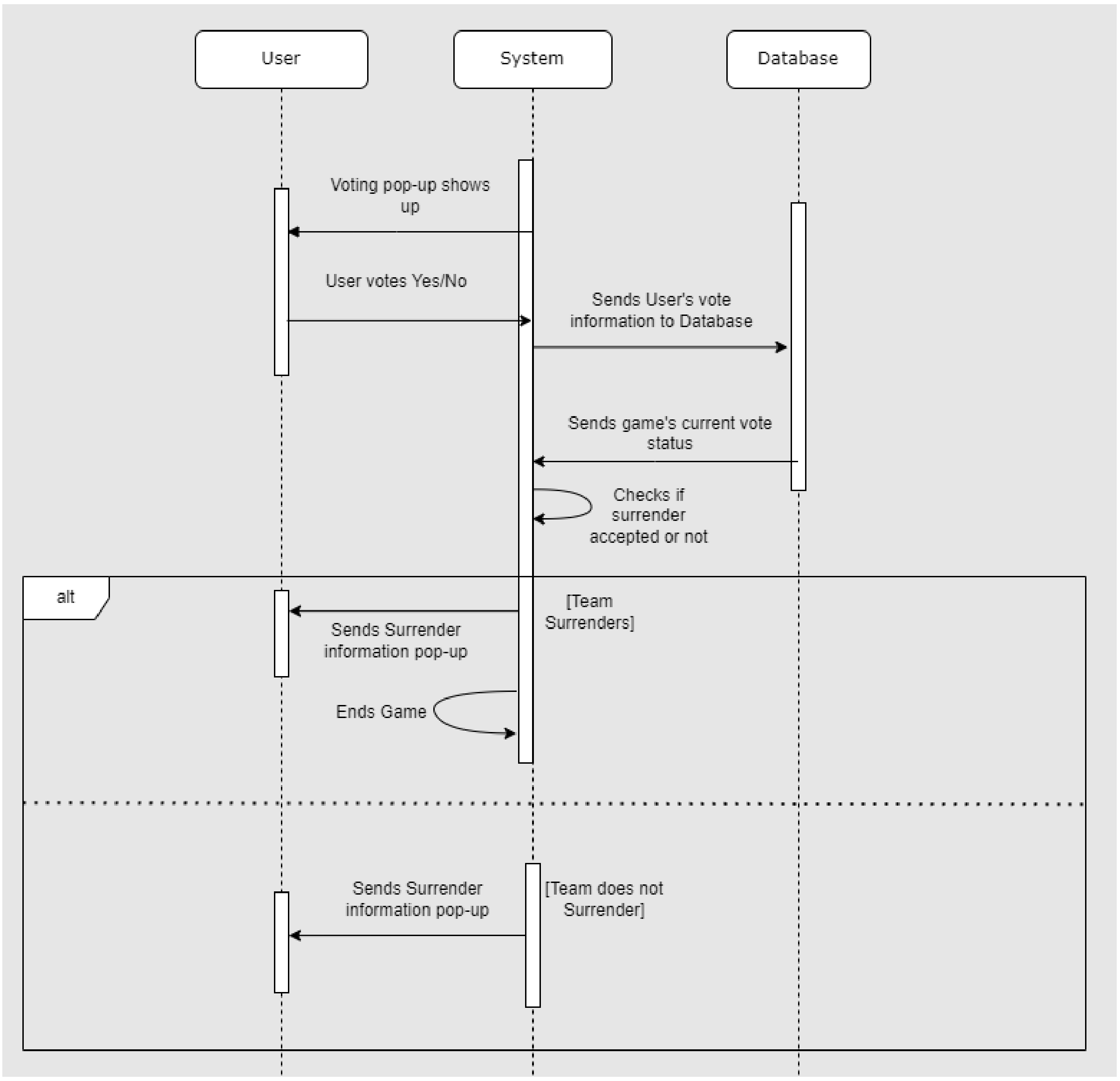
5.2.4.12. Hit


User
System
GPS Service
Database
Press shoot button and send required parameters
System decides if body got hit or not
alt
[No hit]
[Hit]
System makes GPS location request for user
GPS Service returns location of players
System predicts which user got hit
System requests hitted player's health point
Database returns player's health point
System calculates player's new health point
System updates hitted player's health point
alt
System set user as dead
[Health point<=0]
[Health point>0]
Reduce bullet amount on UI

5.2.4.13. Surrender

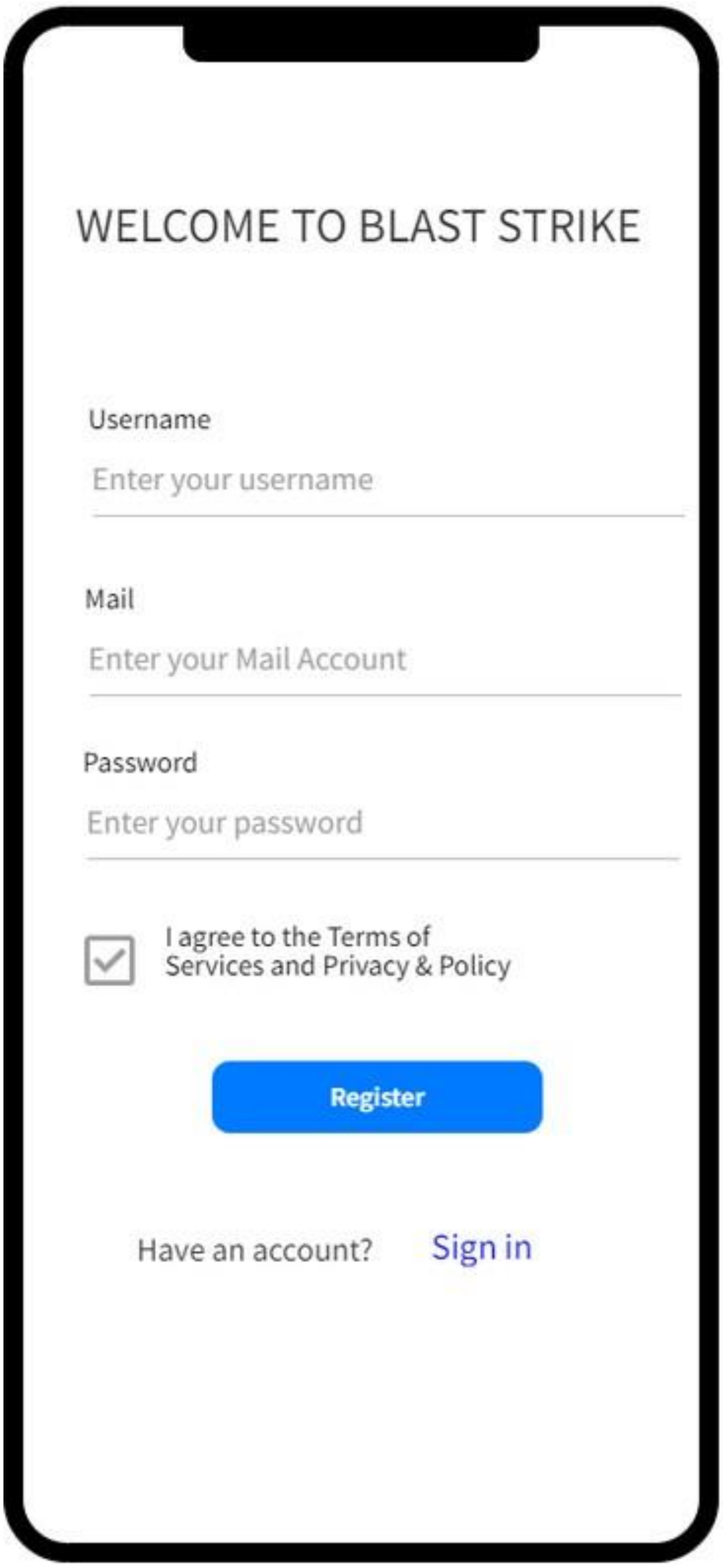

User
System
Database
User opens Settings
User clicks surrender
System sends surrender vote pop-up to all team members
Voting pop-up shows up
User votes Yes/No
Sends User's vote information to Database
Sends game's current vote status
Checks if surrender accepted or not
alt
[Team Surrenders]
Sends Surrender information pop-up
Ends Game
[Team does not Surrender]
Sends Surrender information pop-up

5.2.4.14. Vote

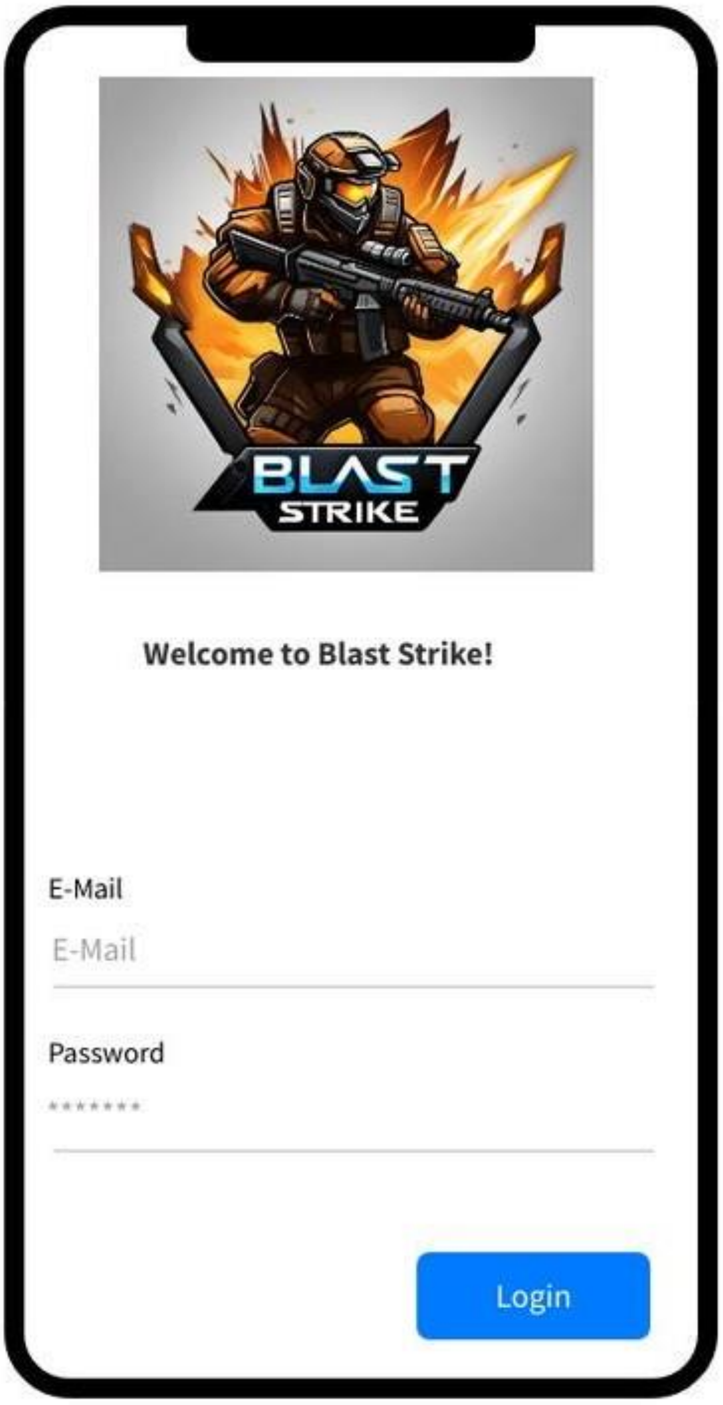

User
System
Database
Voting pop-up shows up
User votes Yes/No
Sends User's vote information to Database
Sends game's current vote status
Checks if surrender accepted or not
alt
[Team Surrenders]
Sends Surrender information pop-up
Ends Game
[Team does not Surrender]
Sends Surrender information pop-up

5.2.5. User Interface Design

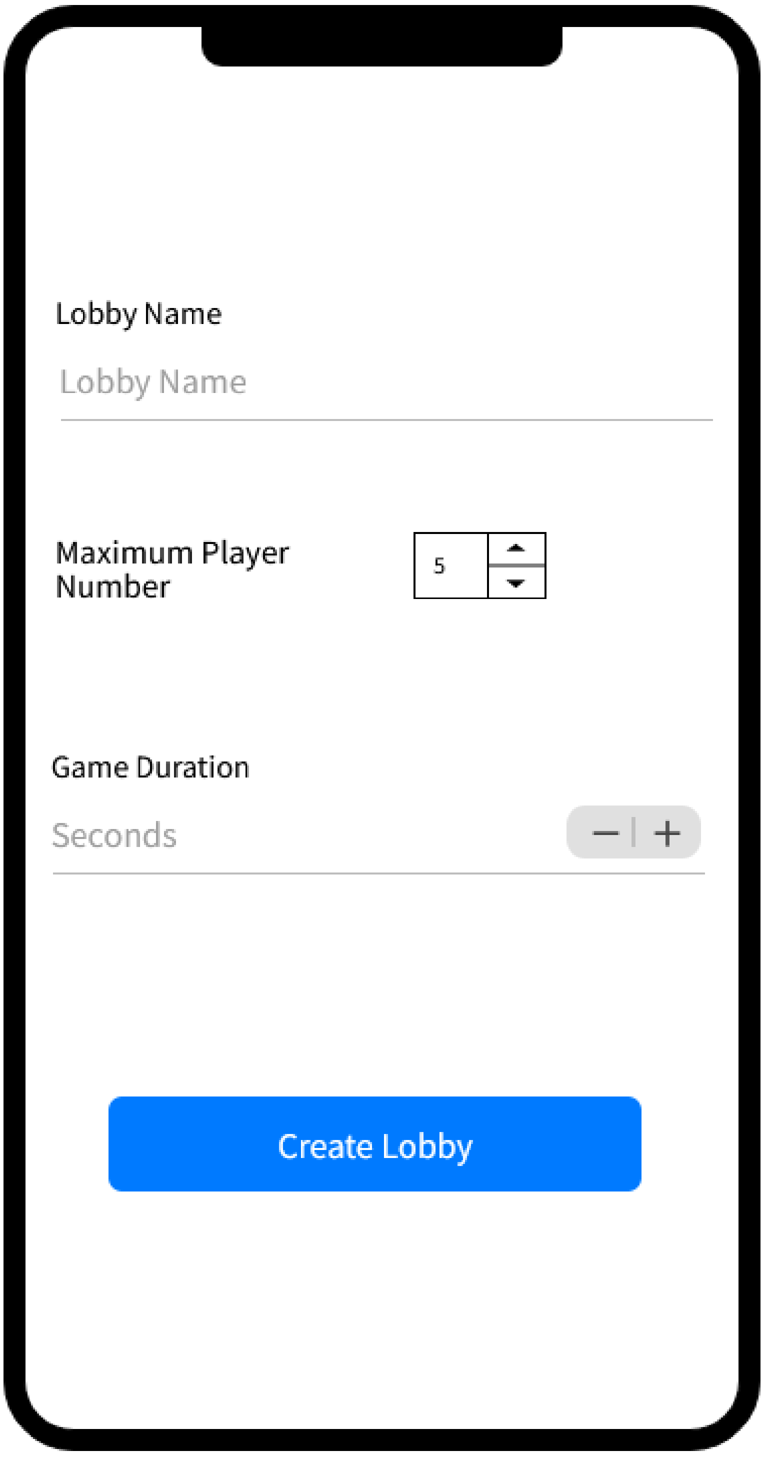
5.2.5.1. Register Page



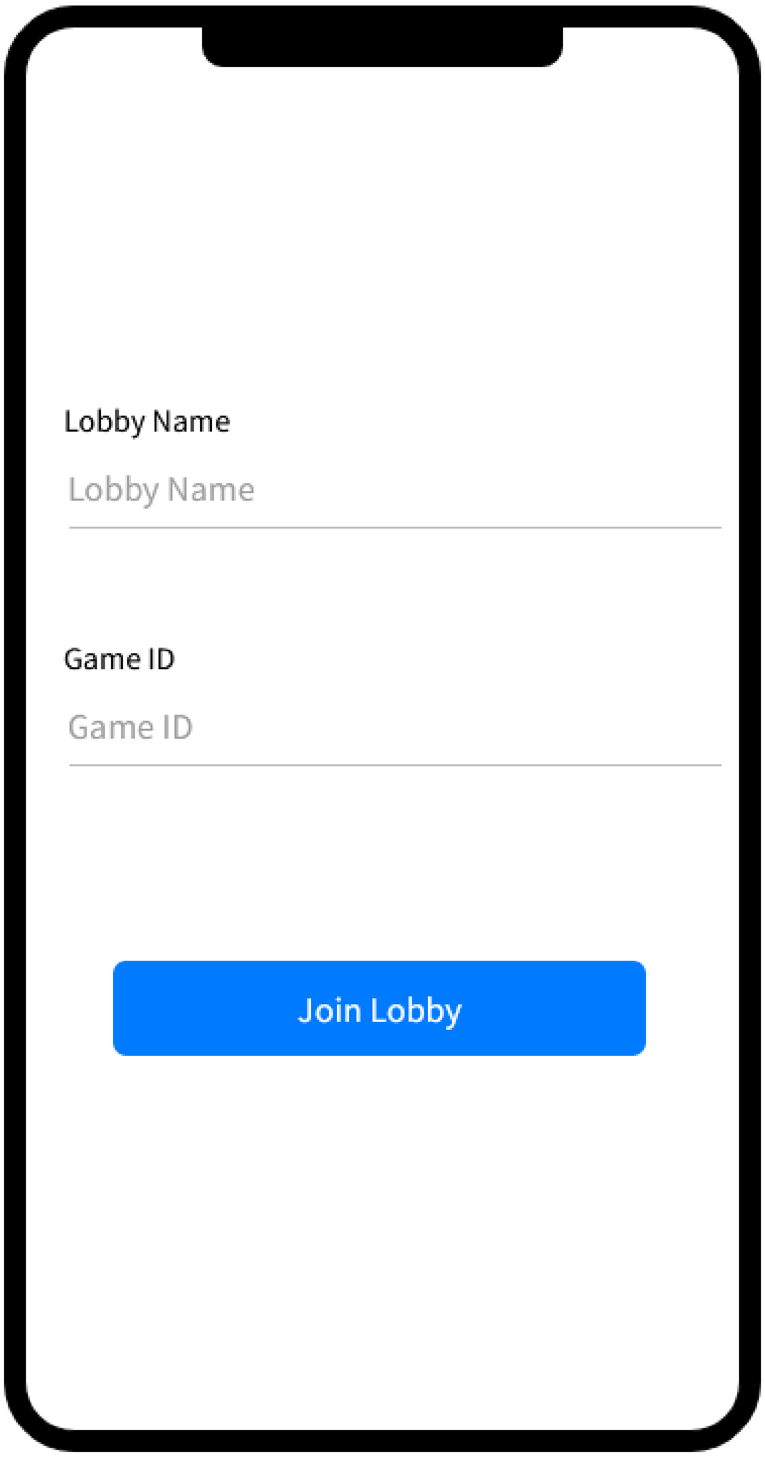
5.2.5.2. *Login Page*



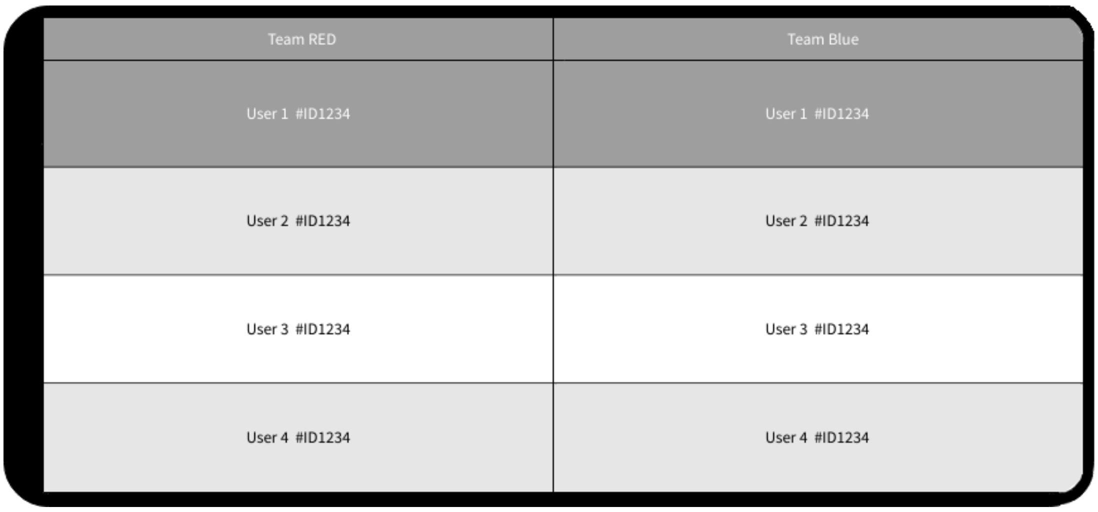
5.2.5.3. *Create Lobby Page*



5.2.5.4. Join Lobby Page

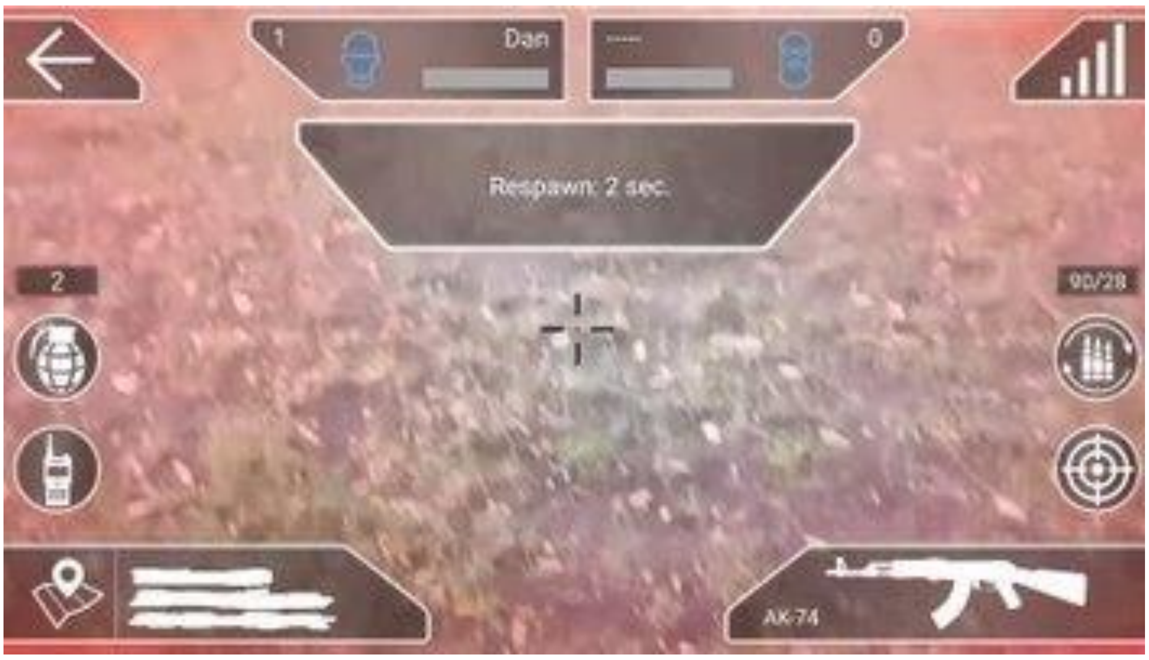


5.2.5.5. *Lobby Page*

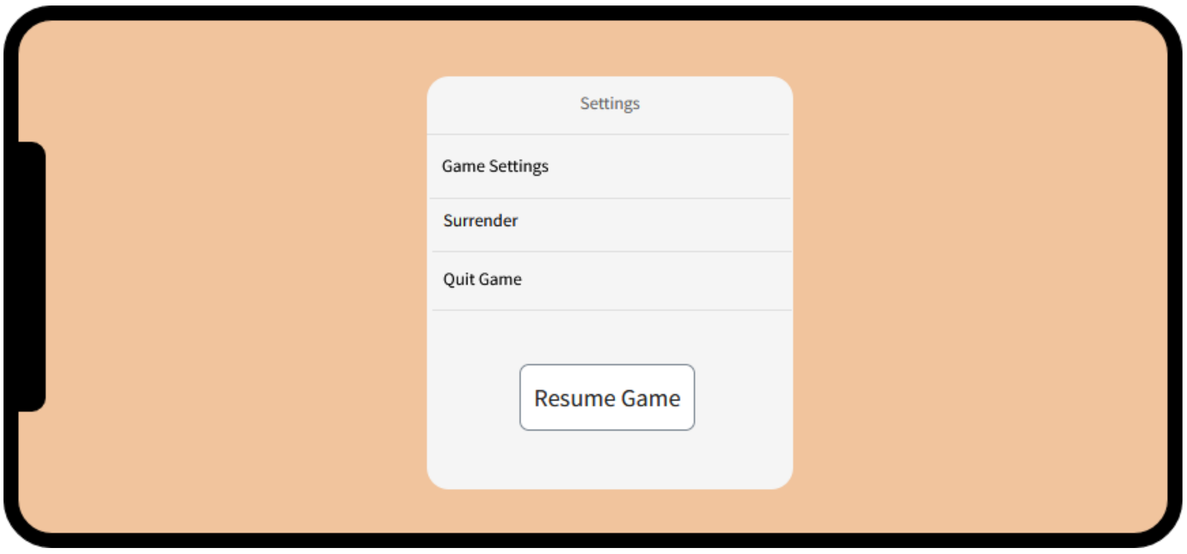


Team RED	Team Blue
User 1 #ID1234	User 1 #ID1234
User 2 #ID1234	User 2 #ID1234
User 3 #ID1234	User 3 #ID1234
User 4 #ID1234	User 4 #ID1234

5.2.5.6. *In- Game Page*



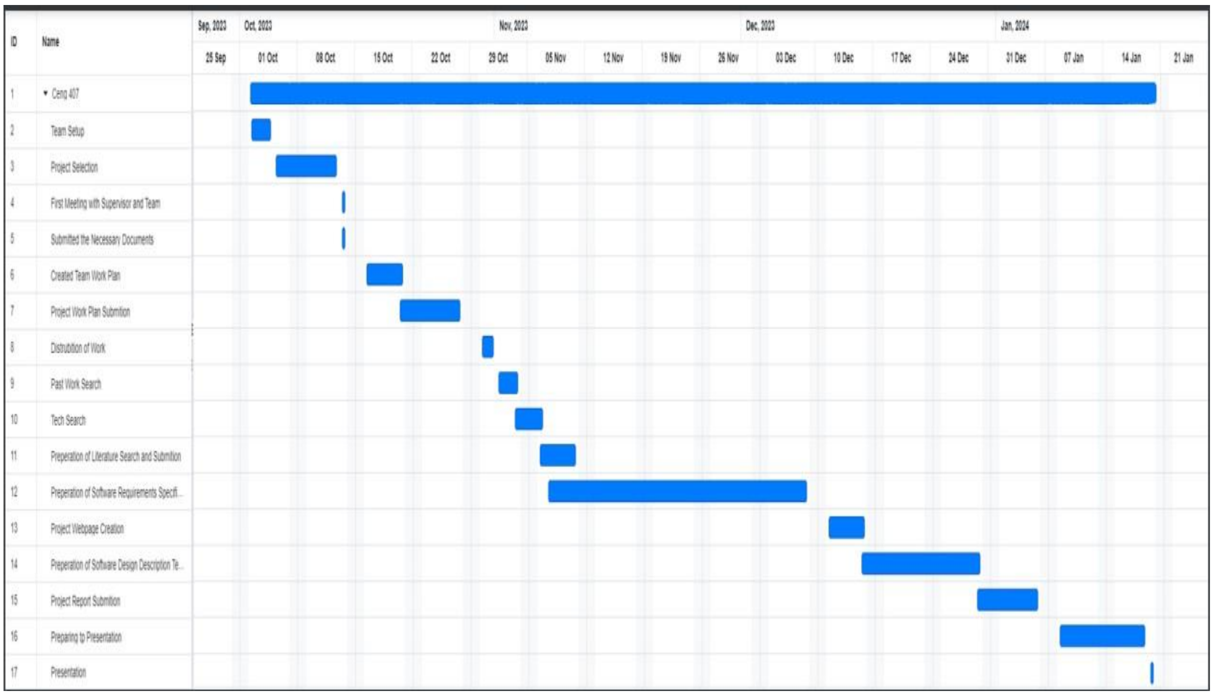
5.2.5.7. *Game Options Pop-Up Page*



6. Conclusion

This game project is developed with the goal of bringing the real-world experience and excitement of laser tag/paintball to the mobile gaming platform. At its core, the project aims to provide a laser tag/paintball experience and enhance interaction among players. The minimum feature set of the project includes real-time body segmentation using TensorFlow Lite for instant scene hit detection, player identification based on GPS location and phone orientation, the ability for players to shoot using Bluetooth controllers, and an AMQP-based message queue structure. These features will form the foundational dynamics of the game, integrating real-world interaction into the mobile gaming experience. The project can be expanded during development by adding more features. For example, additional features such as purchasing skins for the on-screen weapon model, using different weapon models, and introducing features to alter in-game dynamics could be considered. This could enhance the long-term appeal of the game and provide players with various options. Flutter, React Native, or Unity are considered as development platforms due to their support for tflite. This ensures that the game can be published on different mobile platforms. This project aims to deliver an innovative experience by combining both physical interaction and mobile gaming, offering players a unique and engaging experience.

7. Work Plan



References

[1] Tflite_flutter: Flutter Package (2023) Dart packages. Available at: https://pub.dev/packages/tflite_flutter (Accessed: 09 November 2023).
[2] Ar_flutter_plugin: Flutter Package (2022) Dart packages. Available at: https://pub.dev/packages/ar_flutter_plugin (Accessed: 09 November 2023).
[3] asus4 (no date) Asus4/TF-Lite-Unity-sample: Tensorflow Lite samples on Unity, GitHub. Available at: https://github.com/asus4/tf-lite-unity-sample#tensorflow-lite-for-unity-samples (Accessed: 09 November 2023).
[4]Body segmentation in machine learning: Applications & datasets (2023) Datagen. Available at: https://datagen.tech/guides/pose-estimation/body-segmentation/ (Accessed: 09 November 2023).
[5] [updated] BodyPix: Real-time person segmentation in the browser with tensorflow.js (no date) The TensorFlow Blog. Available at: https://blog.tensorflow.org/2019/11/updated-bodypix-2.html (Accessed: 09 November 2023).
[6] Body segmentation with MediaPipe and Tensorflow.js (no date) The TensorFlow Blog. Available at: https://blog.tensorflow.org/2022/01/body-segmentation.html (Accessed: 09 November 2023).
[7] Elivanov, A. (2022) Real-time game server internals: Basic theory, architecture, optimization, auto-scaling, Medium. Available at: https://betterprogramming.pub/real-time-game-server-internals-basic-theory-architecture-optimization-auto-scaling-b2070aa803d9 (Accessed: 09 November 2023).
[8]What is GPS tracking and how does it work?: Azuga (2023) RSS. Available at: https://www.azuga.com/blog/what-is-gps-tracking-and-how-does-it-work (Accessed: 09 November 2023).
[9] Oaikhenan, E. (2022) React native geolocation: A complete tutorial, LogRocket Blog. Available at: https://blog.logrocket.com/react-native-geolocation-complete-tutorial/ (Accessed: 09 November 2023).
[10] Location: Flutter package (2023) Dart packages. Available at: https://pub.dev/packages/location (Accessed: 10 November 2023).
[11]Agency, D. (2022) Best Mobile App Design Tools for developers, DECODE. Available at: https://decode.agency/article/mobile-app-design-tools/ (Accessed: 8 November 2023).
[12]Expertappdevs (2022) Top 13 Artificial Intelligence Mobile App Development tools, Medium. Available at: https://medium.com/@expertappdevs/top-13-artificial-intelligence-mobile-app-development-tools-b65daea09794 (Accessed: 8 November 2023).
[13]AbhijEet Kumar (2023) Is it possible to integrate a Google ML kit with Unity?, Quora. Available at: (Accessed: 8 November 2023).
[14]Firebase Machine learning | firebase ML (no date) Google. Available at: https://firebase.google.com/docs/ml (Accessed: 8 November 2023).
[15]Firebase Machine learning | firebase ML (no date) Google. Available at: https://firebase.google.com/docs/flutter/setup?platform=ios (Accessed: 8 November 2023).
[16]Firebase Machine learning | firebase for unity (no date) Google. Available at: https://firebase.google.com/docs/unity/setup (Accessed: 8 November 2023).