Software Requirements Specification

for

Arena go

Version 1.0 approved

Muhammad Ahmed (21L-1836)
Hamza Amer (21L-5368)

Zaid Asif(21L-5179)
Yasoob Tahavi(21i-0395)

FAST NUCES LHR

1rd March, 2024

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document outlines the requirements for the development of the ArenaGo mobile application. ArenaGo aims to address the challenges faced by futsal and sports arena management by providing a comprehensive digital platform for booking and managing sports facilities. This document covers the scope, features, and technical specifications for the ArenaGo app.

1.2 Document Conventions

This SRS follows standard conventions for documenting software requirements. Requirements are categorized by priorities where higher-level requirements assumed to be inherited by detailed requirements unless stated otherwise.

1.3 Intended Audience and Reading Suggestions

This document is intended for various stakeholders involved in the development and use of the ArenaGo app, including developers, project manager, designers, testers and documentation writers. For developers and project managers, the detailed requirements and technical specifications will be most relevant. Designers will obtain the user interface and design requirements, while testers will focus on the functional and non-functional requirements. Users can refer to the overall descriptions and documentation sections.
Furthermore, the intended audience is sports people including rookies, professionals and middle-aged people looking out for a weekly activity.

1.4 Product Scope

The ArenaGo mobile application is designed to revolutionize the management of futsal and sports arenas. It aims to provide a user-friendly platform for owners, admins and players to efficiently manage and book sports facilities. The app's key features include real-time booking, dynamic pricing, location-based suggestions, and user reviews. By digitizing the booking process and providing a centralized platform, ArenaGo aims to enhance user experience and streamline operations for sports facility owners.

1.5 References

1. **Flutter (Dart) for the frontend**: Flutter
2. **Firebase for backend services**: Firebase
3. **Google Maps API for location-based functionalities**: Google Maps API

2. Overall Description

2.1 Product Perspective

ArenaGo is a new, self-contained product aimed at revolutionizing the management culture of futsal and other sport arenas. It is not a follow-on member of a product family nor a replacement for existing systems. The system will have interfaces with the Flutter frontend, Firebase backend services, and the Google Maps API for location-based functionalities.

2.2 Product Functions

- Digitize the process of booking and managing time slots for futsal and sports arenas.
- Allow owners to add and manage time slots, set hold amounts, and display facility rules and images.
- Enable players to browse, select, and book futsal, invite friends, and modify or cancel bookings.
- Maintain records and reviews of players, and track earnings for futsal owners.
- Update booking costs upon decreasing remaining time for an unbooked field
- Allow users to find teams to hold games against

2.3 User Classes and Characteristics

- Admins: Have comprehensive access to manage multiple futsals under a single user account.
- Owners: Act as facility owners and manage futsals, including adding time slots and setting hold amounts.
- Players: Book futsal, invite friends, and pay to play bookings.

Subdividing players into further classes, we have teenagers who have ample time and energy and thus will be our main target audience having the greatest frequency of use. Middle-aged office going men will have less time and energy but would prefer to be involved once/twice a week so they are comparatively less important to satisfy

2.4 Operating Environment

The ArenaGo app will operate on mobile devices running the Flutter framework. It will require integration with Firebase for backend services and the Google Maps API for location-based functionalities.

2.5 Design and Implementation Constraints

- Firebase for backend services require large memory sets for response from the database
- Limited access of free data through Google Maps API for location-based functionalities.
- Potential competitors jumping on the new/not yet used idea

2.6 User Documentation

User documentation components will include user manuals, on-line help, and tutorials. The delivery format will be digital, possibly in the form of PDF documents or online guides.

2.7 Assumptions and Dependencies

Assumptions:

- Assumption that users have access to mobile devices capable of running the ArenaGo app.
- Assumption that users have internet connectivity to access Firebase and the Google Maps API.

Dependencies:

- Compliance of field staff for shifting to automated systems
- Dependency on the availability and functionality of Firebase and the Google Maps API for backend services and location-based functionalities, respectively.

3. External Interface Requirements

3.1 User Interfaces

The user interfaces of ArenaGo are designed to provide an intuitive and engaging experience for different user classes, including admins, owners, and players. The interfaces will be developed using the Flutter framework, offering a consistent look and feel across various mobile devices. Key aspects of the user interfaces include:

3.1.1 Login and Registration Interface:
Users will have a secure and user-friendly interface to log in or register within the ArenaGo app. The design will incorporate fields for entering usernames/emails and passwords during login, and essential registration details during the sign-up process.

3.1.2 Profile Management Interface:
Users will be able to view, edit, and manage their profile information through an easily navigable interface.
The system will provide options for users to upload or change profile pictures, ensuring a personalized touch.

3.1.3 Venue Booking Interface:
The system will display a list of available futsal venues with their respective time slots.
Users will have the ability to select and book available time slots seamlessly through an intuitive interface.

3.1.4 Payment Processing Interface:
A user-friendly interface will be provided for initiating and completing payment transactions securely.

The design will accommodate multiple payment methods, including credit/debit cards, mobile wallets, and other digital payment options.

3.1.5 Notifications and Alerts Interface:

Users will have the option to customize their notification preferences, determining the types of notifications and preferred communication channels (e.g., push notifications, in-app messages, email).

The interface will allow users to view and manage their notifications effectively.

3.1.6 Review and Rating Interface:

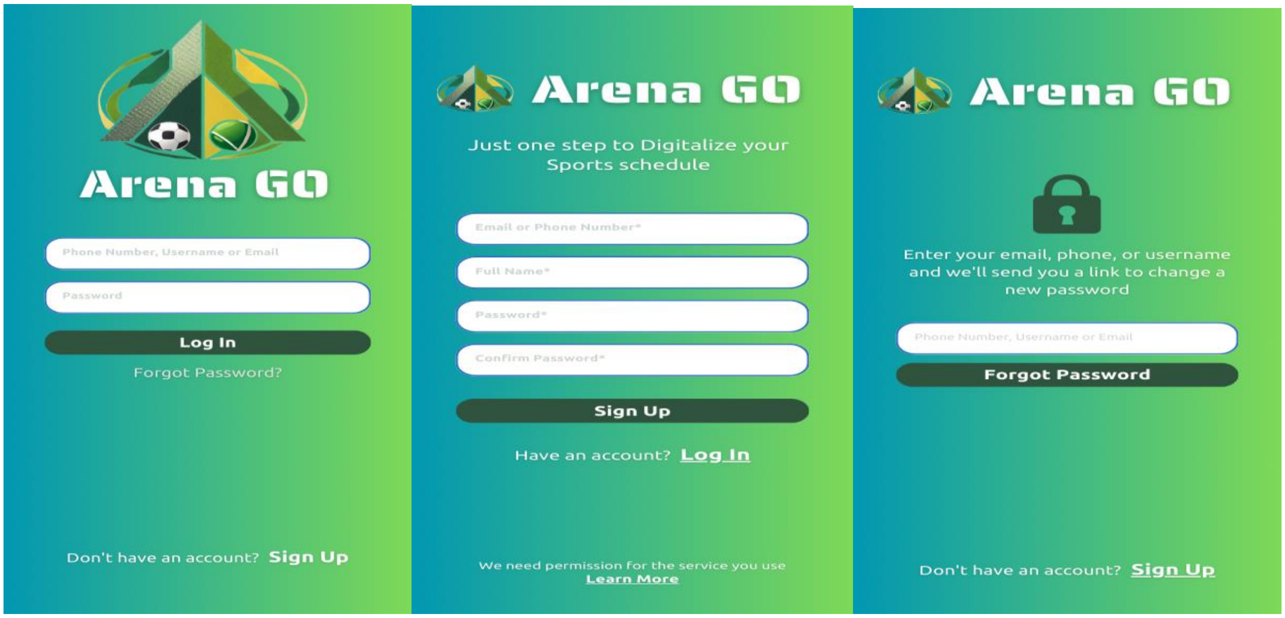
Users will be able to browse and select venues or users for review through a straightforward interface.

The system will display reviews and ratings prominently, providing users with valuable insights for decision-making.

3.1.7 Owner Interface (Add Ground) and Verification Interface:

Venue owners will have an intuitive interface for adding new grounds to the app, providing necessary details and images.

The system will display a verification interface to ensure the authenticity and quality of newly added futsal venues.


Arena GO
Phone Number, Username or Email
Password
Log In
Forgot Password?
Don't have an account? Sign Up
Arena GO

Just one step to Digitalize your Sports schedule

We need permission for the service you use

Enter your email, phone, or username and we'll send you a link to change a new password

Phone Number, Username or Email
Forgot Password
Don't have an account? Sign Up

User interfaces for login/sign-up panel

3.2 Hardware Interfaces

To ensure the optimal performance of ArenaGo, the software product must interface with the hardware components of users' devices. Here are the specifications:

3.2.1 **Supported Device Types:**

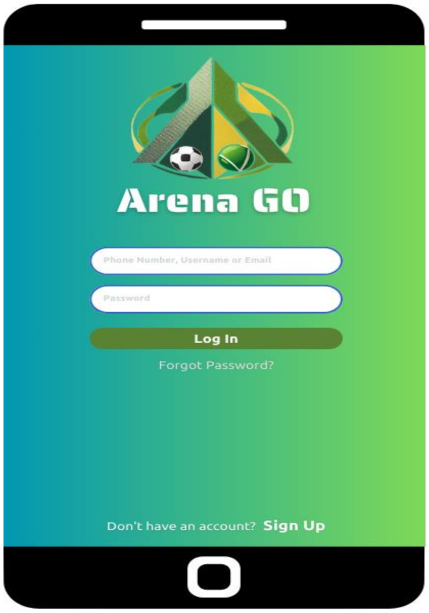
ArenaGo is designed to operate seamlessly on contemporary mobile devices, including smartphones and tablets. It is imperative that users possess up-to-date devices that meet standard industry specifications. Notably, ArenaGo, being based on the Flutter framework, is compatible with both iOS and Android platforms.

3.2.2 **Nature of Data and Control Interactions:**

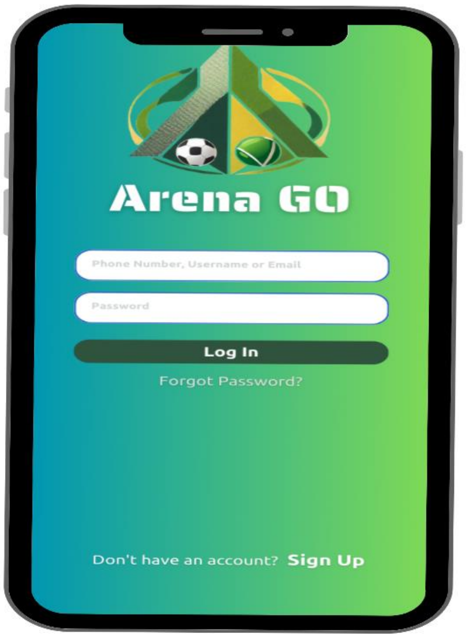
The application will leverage the hardware features of the device, such as the screen, buttons, and touch capabilities. User interactions will involve standard mobile gestures such as tapping and swiping. Additionally, the app will rely on the device's internet connectivity to establish communication with the backend (Firebase) for data retrieval and updates.

3.2.3 **Communication Protocols:**

ArenaGo will adhere to established communication protocols commonly used in mobile applications. These protocols will ensure secure and efficient data exchange between the application and the user's device.


Arena GO
Phone Number, Username or Email
Password
Log In
Forgot Password?
Don't have an account? Sign Up

ANDRIOD INTERFACE


Arena GO
Phone Number, Username or Email
Password
Log In
Forgot Password?
Don't have an account? Sign Up

IOS INTERFACE

3.3 Software Interfaces

3.3.1 **Frontend (Flutter):**

The primary user interface is built using Flutter, version [insert version if known]. Flutter serves as the framework for the ArenaGo mobile application, ensuring a consistent and responsive user experience across both iOS and Android platforms. The frontend is responsible for user interactions, displaying information, and facilitating communication with the backend.

3.3.2 **Backend (Firebase):**

ArenaGo interacts with Firebase, a cloud-based platform, for backend services. The version used is [insert version if known]. Firebase handles data storage, user authentication, and real-time updates. The software communicates with Firebase through APIs, allowing seamless integration of user data, booking information, and other relevant details.

3.3.3. **Location-Based Functionalities (Google Maps API):**

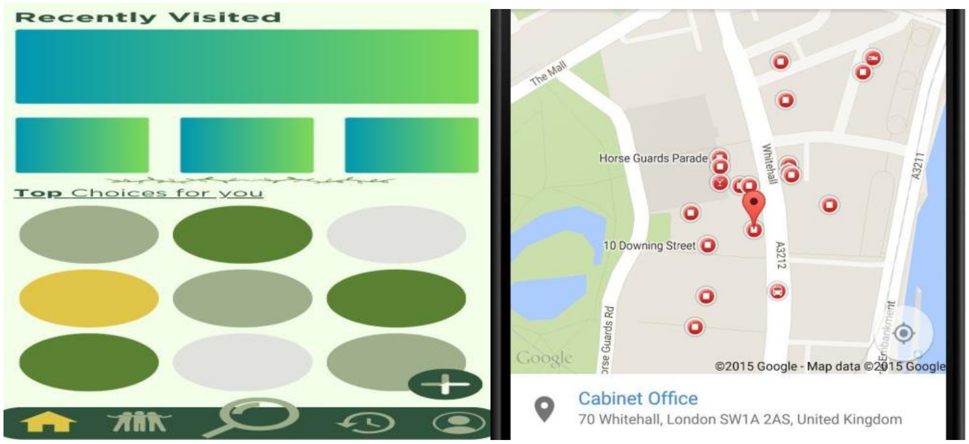
The Google Maps API is utilized for location-based features within ArenaGo. The version integrated is [insert version if known]. This interface enables the app to provide location-specific suggestions, mapping, and other geographical functionalities. Communication with the Google Maps API occurs through specified API protocols, enhancing the app's usability.

3.3.4 **Communication Protocols:**

ArenaGo uses standard communication protocols for seamless interaction between frontend and backend components. HTTPS is employed to ensure secure data transmission, preventing unauthorized access and maintaining the confidentiality of user information.

3.3.5 **Data Sharing Mechanism:**

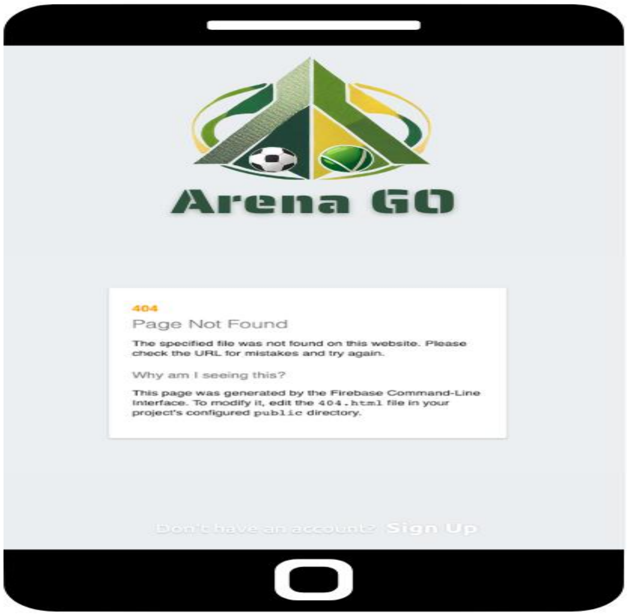
Data sharing between software components is facilitated through structured APIs. The frontend communicates with the backend, requesting and receiving data in a predefined format. Data items include user details, booking information, and venue specifics. The adherence to a standardized data format ensures consistency and compatibility between the different software components.



Access to map, previous history, add fields and other features

3.4 Communications Interfaces

- **Push Notifications:** Standard services for iOS and Android keep users updated on events.
- **Firebase Realtime Database:** Ensures instant updates on bookings and venue availability via HTTPS.
- **RESTful APIs:** For external services like payment gateways, maintaining secure communication.
- **Security:** HTTPS across interfaces ensures data encryption and user information confidentiality.



404

Page Not Found

The specified file was not found on this website. Please check the URL for mistakes and try again.

Why am I seeing this?

This page was generated by the Firebase Command-Line Interface. To modify it, edit the `404.html` file in your project's configured `public` directory.

Error message on unsuccessful HTTP request

4. System Features

4.1: Login

4.1.1 **Description and Priority**

The login feature allows users to securely access their accounts within the ArenaGo mobile app. This feature is of high priority as it is essential for user authentication and access control.
Priority: High

4.1.2 **Stimulus/Response Sequences**

User launches the ArenaGo app.
User taps on the "Login" button.
System prompts the user to enter their username/email and password.

User inputs their credentials.
System verifies the entered credentials.
If the credentials are valid, the system grants access to the user's account and navigates to the user's dashboard.
If the credentials are invalid, the system displays an error message prompting the user to try again or reset their password.

4.1.3 **Functional Requirements**

REQ-1: The system shall provide a user interface with fields for the user to input their username/email and password.
REQ-2: The system shall securely transmit and store user login credentials using encryption protocols.
REQ-3: The system shall verify user credentials against stored user data in the database.
REQ-4: If user authentication fails, the system shall display an appropriate error message to the user.
REQ-5: The system shall provide an option for users to reset their password in case they forget it.
REQ-6: The system shall implement session management to maintain user authentication throughout the app usage session.
REQ-7: The system shall comply with relevant data protection and privacy regulations, ensuring user data security and confidentiality.

4.2: Sign Up

4.2.1 **Description and Priority**

The sign-up feature enables new users to create an account within the ArenaGo mobile app. This feature is of high priority as it facilitates user onboarding and access to the app's functionalities.
Priority: High

4.2.2 **Stimulus/Response Sequences**

User launches the ArenaGo app.
User taps on the "Sign Up" button.
System prompts the user to enter their email, username, password, and other necessary information.
User inputs their details.
System verifies the entered information for validity and uniqueness.
If the information provided passes validation, the system creates a new user account and grants access.
If there are validation errors or duplicate entries, the system prompts the user to correct the information and resubmit.

4.2.3 **Functional Requirements**

REQ-1: The system shall provide a user-friendly interface for users to input their registration details.
REQ-2: The system shall validate the entered email for correct format and uniqueness.
REQ-3: The system shall validate the entered username for uniqueness.
REQ-4: The system shall enforce password strength requirements to ensure account security.
REQ-5: Tentative: The system shall send a verification email to the user's provided email address for account activation.
REQ-6: The system shall provide error messages for invalid or duplicate entries during registration.
REQ-7: The system shall store user registration data securely in the database.

4.3: User Profile Management

4.3.1 **Description and Priority**

The user profile management feature enables users to view, update, and manage their personal information and preferences within the ArenaGo mobile app. This feature is of high priority as it empowers users to maintain accurate and relevant profile data for a personalized experience.
Priority: High

4.3.2 **Stimulus/Response Sequences**

User accesses the profile section of the ArenaGo app.
System displays the user's profile information, including name, email, contact details, etc.
User selects the option to edit profile information.
System prompts the user to input updated details or make changes to existing information.
User submits the changes.
System verifies the updated data and makes the necessary modifications.
If the update is successful, the system confirms the changes and updates the user's profile.
If there are any errors or issues with the update, the system notifies the user and prompts them to correct the information.

4.3.3 **Functional Requirements**

REQ-1: The system shall provide a user-friendly interface for viewing and editing user profiles.
REQ-2: The system shall display the user's profile information, including name, email, contact details, etc.
REQ-3: The system shall allow users to edit and update their profile information, including name, email, password, contact details, etc.

REQ-4: The system shall validate updated profile information for correctness and completeness.
REQ-5: The system shall securely store and manage user profile data in the database.
REQ-6: The system shall provide options for users to upload or change profile pictures.
REQ-7: The system shall maintain a history of profile updates for audit and tracking purposes.
REQ-8: The system shall ensure compliance with data protection regulations regarding the handling of user profile information.

This feature ensures that users have full control over their profile information, promoting transparency and trust within the ArenaGo community.

4.4 Time Slot Booking by User

4.4.1 **Description and Priority**

The time slot booking feature allows users to reserve time slots at futsal venues through the ArenaGo app. This feature is of high priority as it facilitates user engagement and enhances the user experience by enabling convenient booking processes.
Priority: High

4.4.2 **Stimulus/Response Sequences**

User navigates to the booking section of the ArenaGo app.
User selects the desired futsal venue and available time slots.
System displays available time slots and prompts the user to select one.
User selects a time slot and confirms the booking.
System verifies the booking details and updates the booking status.
If the booking is successful, the system confirms the reservation and notifies the user.
If the booking fails (e.g., slot already booked), the system prompts the user to choose another slot.

4.4.3 **Functional Requirements**

REQ-1: The system shall display a list of available futsal venues and their respective time slots.
REQ-2: The system shall allow users to select and book available time slots.
REQ-3: The system shall update the booking status in real-time to prevent double bookings.
REQ-4: The system shall provide notifications to users confirming successful bookings.
REQ-5: The system shall handle booking conflicts and provide alternative options if necessary.
REQ-6: The system shall store booking details securely in the database.
REQ-7: The system shall provide options for users to modify or cancel their bookings.

4.5: Payment Processing

4.5.1 **Description and Priority**

The payment processing feature enables users to make secure and convenient transactions for booking futsal venues and other services within the ArenaGo mobile app. This feature is of high priority as it facilitates monetization, ensures user satisfaction, and supports the growth of the platform.
Priority: High

4.5.2 **Stimulus/Response Sequences**

User initiates a payment transaction for booking a futsal venue or purchasing additional services within the ArenaGo app.
System presents the user with available payment options, such as credit/debit card, mobile wallets, or other digital payment methods.
User selects their preferred payment method and provides the necessary payment details.
System securely processes the payment transaction through integrated payment gateways or third-party payment processors.
Upon successful payment authorization, the system updates the booking status or provides access to the purchased services.
If the payment transaction fails for any reason (e.g., insufficient funds), the system notifies the user and prompts them to retry or choose an alternative payment method.

4.5.3 **Functional Requirements**

REQ-1: The system shall integrate with one or more secure payment gateways or third-party payment processors to facilitate payment transactions.
REQ-2: The system shall provide a user-friendly interface for selecting payment methods and entering payment details.
REQ-3: Tentative: The system shall encrypt payment data during transmission and storage to ensure security and compliance with payment industry standards.
REQ-4: Tentative: The system shall support multiple currencies and payment methods to accommodate diverse user preferences.
REQ-5: The system shall provide real-time payment transaction status updates to users.
REQ-6: The system shall handle payment transaction errors and provide appropriate error messages to users.
REQ-7: The system shall maintain transaction records and provide users with access to their payment history.

This feature facilitates seamless and secure transactions, contributing to user trust and satisfaction with the ArenaGo platform.

4.6: Notifications and Alerts

4.6.1 **Description and Priority**

The notifications and alerts feature enables the ArenaGo mobile app to keep users informed about important updates, such as booking confirmations, new messages, and upcoming events. This feature is of high priority as it enhances user engagement, improves user experience, and ensures timely communication between users and the platform.

Priority: High

4.6.2 **Stimulus/Response Sequences**

System detects relevant events or updates, such as booking confirmations, new messages, or upcoming events.
System generates notifications or alerts based on the detected events.
Notifications are sent to the respective users via push notifications, in-app messages, or email, depending on user preferences and the nature of the notification.
Users receive the notifications on their devices or within the app.
Users can view the notifications and take appropriate actions, such as responding to messages or confirming bookings.

4.6.3 **Functional Requirements**

REQ-1: The system shall provide options for users to customize their notification preferences, including the types of notifications they wish to receive and the preferred communication channels (e.g., push notifications, in-app messages, email).
REQ-2: The system shall generate notifications for relevant events, such as booking confirmations, new messages, upcoming events, etc.
REQ-3: The system shall send notifications to users in a timely manner.
REQ-4: The system shall ensure that notifications are delivered reliably to users' devices or within the app.
REQ-5: The system shall provide options for users to view and manage their notifications within the app.
REQ-6: The system shall implement measures to prevent spam or excessive notifications.
REQ-7: The system shall maintain a history of notifications for audit and tracking purposes.

This feature plays a crucial role in keeping users informed and engaged, thereby enhancing their overall experience with the ArenaGo platform.

4.7 Owner Interface (Add Ground)

4.7.1 **Description and Priority**

The owner interface feature allows futsal venue owners to add their grounds to the ArenaGo app, enabling them to manage their facilities and bookings efficiently. This feature is of high priority as it facilitates the expansion of the app's venue database and enhances the app's utility for both owners and users.
Priority: High

4.7.2 **Stimulus/Response Sequences**

Venue owner accesses the owner interface within the ArenaGo app.
Owner selects the option to add a new ground.
System prompts the owner to input details about the venue, such as name, location, facilities, etc.
Owner provides the necessary information and uploads relevant images.
System verifies the submitted information and validates the addition of the new ground.
If the addition is successful, the system updates the venue database and notifies the owner.

If there are any errors or missing information, the system prompts the owner to correct them.

4.7.3 **Functional Requirements**

REQ-1: The system shall provide a user-friendly interface for venue owners to add new grounds.
REQ-2: The system shall validate the entered information for completeness and accuracy.
REQ-3: The system shall allow owners to upload images and provide descriptions of their venues.
REQ-4: The system shall store added venue details securely in the database.
REQ-5: The system shall notify venue owners of successful ground additions.
REQ-6: The system shall handle errors gracefully and prompt owners to rectify any issues.

4.8 Verification of Added Ground

4.8.1 **Description and Priority**

The verification of added ground feature ensures the authenticity and quality of newly added futsal venues within the ArenaGo app. This feature is of high priority as it maintains the credibility of the app's venue database and ensures a positive user experience.
Priority: High

4.8.2 **Stimulus/Response Sequences**

System detects a newly added futsal venue submitted by an owner.
System initiates the verification process by reviewing the submitted venue details and images.
If the information meets verification criteria, the system marks the venue as verified.
Verified venues are displayed as trusted options for users to book.
If the information is insufficient or raises concerns, the system flags the venue for further review or contact with the owner.

4.8.3 **Functional Requirements**

REQ-1: The system shall automatically review newly added venue details and images for verification.
REQ-2: The system shall mark verified venues as trusted options for users to book.
REQ-3: The system shall flag venues with insufficient or questionable information for further review.
REQ-4: The system shall provide feedback to venue owners regarding the verification status of their added grounds.

4.9: Review and Rating System

4.9.1 **Description and Priority**

The review and rating system allows users to provide feedback and rate their experiences with futsal venues and other users within the ArenaGo mobile app. This feature is of high priority as it fosters community engagement, facilitates decision-making for users, and promotes accountability among venue owners.

Priority: High

4.9.2 **Stimulus/Response Sequences**

User navigates to the review section of the ArenaGo app.
System displays a list of futsal venues or users available for review.
User selects the venue or user they wish to review.
System prompts the user to provide a rating (e.g., star rating) and optional written feedback.
User submits the review.
System verifies the review data for completeness and appropriateness.
If the review is valid, the system updates the venue or user's rating and displays the review.
If there are any issues with the review (e.g., inappropriate content), the system notifies the user and prompts them to make corrections.

4.9.3 **Functional Requirements**

REQ-1: The system shall provide a user interface for browsing and selecting venues or users to review.
REQ-2: The system shall allow users to provide ratings (e.g., star ratings) and optional written feedback for venues or users.
REQ-3: The system shall validate review data to ensure completeness and appropriateness.
REQ-4: The system shall update venue or user ratings based on received reviews.
REQ-5: The system shall display reviews and ratings for venues and users within the app.
REQ-6: The system shall provide options for users to filter and sort reviews based on criteria such as date, rating, etc.
REQ-7: The system shall allow users to edit or delete their own reviews if needed.

These features encourages user participation and contributes to building a trustworthy and transparent platform for the ArenaGo community.

5. Other Nonfunctional Requirements

5.1: Performance Requirements

- The ArenaGo mobile app should load within 3 seconds on average for users with a stable internet connection, to ensure a smooth user experience.
- The booking and payment processing functionalities should respond within 5 seconds on average, to prevent user frustration during the booking process.
- The app should be capable of handling concurrent user interactions without significant latency, ensuring responsiveness even during peak usage hours.
- The system should be scalable to accommodate increasing user traffic and data volume, ensuring sustained performance as the user base grows.

5.2: Safety Requirements

- The app should implement measures to protect user data privacy and ensure compliance with relevant data protection regulations (e.g., GDPR, CCPA).
- User authentication processes should be secure to prevent unauthorized access to user accounts and sensitive information.
- Payment transactions should be encrypted and processed securely to prevent fraud and protect users' financial information.

5.3: Security Requirements

- User identity authentication should be enforced through secure login mechanisms, such as password hashing and multi-factor authentication (optional).
- The app should implement role-based access control to ensure that only authorized users can perform certain functions, such as venue management.
- Regular security audits and penetration testing should be conducted to identify and mitigate potential vulnerabilities in the app's infrastructure and codebase.

5.4: Software Quality Attributes

- Usability: The app should provide an intuitive user interface with clear navigation and minimal learning curve, ensuring ease of use for users of all skill levels.
- Reliability: The app should be stable and resilient, minimizing crashes or downtime to provide a consistent user experience.
- Maintainability: The codebase should be well-structured and documented, facilitating easy maintenance and future updates by developers.
- Portability: The app should be compatible with a wide range of mobile devices and operating systems, ensuring accessibility for a diverse user base.

5.5: Business Rules

- Only registered users should be allowed to book futsal venues or access certain features within the app, such as venue management.
- Venue owners should be able to set their own pricing and availability rules for bookings, subject to approval by the app administrators.
- Users should adhere to the app's community guidelines and terms of service, with violations resulting in account suspension or termination.

These nonfunctional requirements ensure that the ArenaGo mobile app meets performance, safety, security, and quality standards while adhering to business rules and regulations.

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>