**Software Requirements Specification (SRS)**

**Project Name**: Ticket Arena  
**Prepared By**: Ahmed Ebaid - Ahmed Yasser - Abdulkhaliq Sarwat - Mohamed Saleh - Kareem Ezzaldin   
**Date**: [9/11/2024]

**1. Introduction**

**1.1 Purpose**

The purpose of this document is to outline the functional and non-functional requirements for the Ticket Arena project. It provides a detailed description of the system’s features, constraints, and serves as a guide for developers, testers, and stakeholders.

**1.2 Scope**

Ticket Arena system will offer a convenient platform to browse, select, and book tickets for events. The system will provide functionality for:

* browsing available football matches.
* filter matches by team, stadium, and ticket category.
* Password recovery and security measures
* view real-time ticket availability for each match
* view the store of the favorite team.
* selecting seats based on availability.
* making secure payments using Fawry or Aman.
* sending email confirmation after booking by system
* viewing highlights of the favorite team matches.
* allowing users to register and log in.
* viewing booking history.

**1.3 Definitions, Acronyms, and Abbreviations**

* **API**: Application Programming Interface
* **UAT**: User Acceptance Testing
* **UI:** User interface
* **UX**: User Experience

**2. System Overview**

**2.1 Product Perspective**

Ticket Arena is a standalone system that interacts with users and external payment services. The system architecture includes:

* **Frontend**: using React,HTML,JavaScript for user interface
* **Backend**: Django will handle the back-end
* **Database**: The storage for all persistent data.

**2.2 Product Functions**

The key functions of the system include:

1. **Enabling users to browse upcoming matches.**
2. **Providing filtering options based on team, stadium, and ticket categories.**
3. **Allowing seat selection and real-time ticket availability updates.**
4. **Offering multiple secure payment options and booking confirmations.**
5. **Displaying recent team highlights for an enriched user experience.**

**2.3 User Classes and Characteristics**

Different types of users who will interact with the system include:

* **Admin Users**: Manage events, users, and handle administrative settings.
* **Regular Users**: Can browse matches, book tickets, manage bookings, view their booking history, and shop in their favorite team store.
* **Guest Users**: Can only browse matches without booking capabilities.

**2.4 Operating Environment**

The system will operate in the following environments:

* **Client Side**: Runs on web browsers like Chrome, Firefox, and Safari.
* **Server Side**: Hosted on Amazon Web Services server ,windows, running, and using Node.js
* **Database**: MySQL

**3. Functional Requirements**

**3.1 Use Case Diagrams / User Stories**

* **Use Case 1**: Ticket Booking
  + **Description**: The user can browse matches, select seats, make payments, and receive booking confirmations.
  + **Actors**: User.
  + **Steps**:
    1. Users browse upcoming matches.
    2. User selects a match and views seating options.
    3. User selects seats and proceeds to payment.
    4. System processes payment and sends a booking confirmation.
* Use case 2 : User Registration/Login
* Actors: Guest User, Regular User
* Description: A new user can sign up and log in to manage bookings and view booking history.

**3.2 Feature Requirements**

**Feature 1: Event Browsing and Filtering**

* **Description**: Allows users to view upcoming football matches and filter results based on criteria such as team, stadium, and ticket category.
* **Inputs**: team name, stadium name
* **Outputs**: list of matches with details such as date, location, and ticket options.
* **Error Handling**: If no matches are found for the criteria, display a message like "No matches available for the selected criteria."

**Feature 2: Profile Management**

* **Description**: Allows users to manage their personal information (e.g. name, email, password) and view booking history..
* **Inputs**: User information updates, login credentials.
* **Outputs**: Confirmation of updated profile information or display of booking history.
* **Error Handling**: If the update fails, display a message prompting the user to retry. For failed login attempts, display an appropriate error (e.g. incorrect password).

**4. Non-Functional Requirements**

**4.1 Performance Requirements**

* The system should respond to user requests within 2 seconds.
* It must support up to 500 concurrent users without significant degradation.

**4.2 Security Requirements**

* User data must be encrypted in transit and at rest.
* Only authorized users should have access to specific system functionalities.

**4.3 Usability Requirements**

* The user interface should be intuitive, with clear navigation and user guidance.
* Accessibility features should be included to support users with disabilities, following WCAG 2.1.

**4.4 Reliability and Availability Requirements**

* The system must have 99.9% uptime.
* In case of a failure, the system must recover within 5 minutes

**4.5 Scalability**

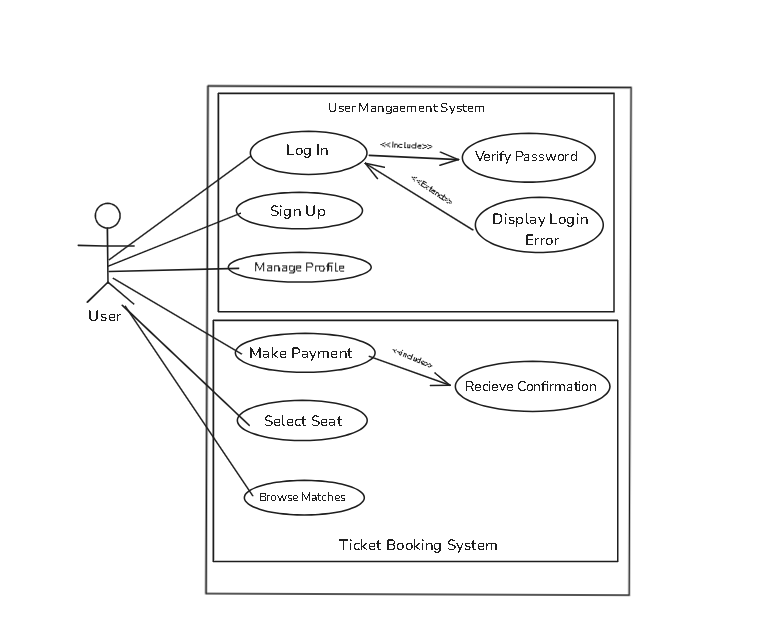
* The system should be able to scale to support 3500 users or 100 GB of data.

**4.6 Compatibility**

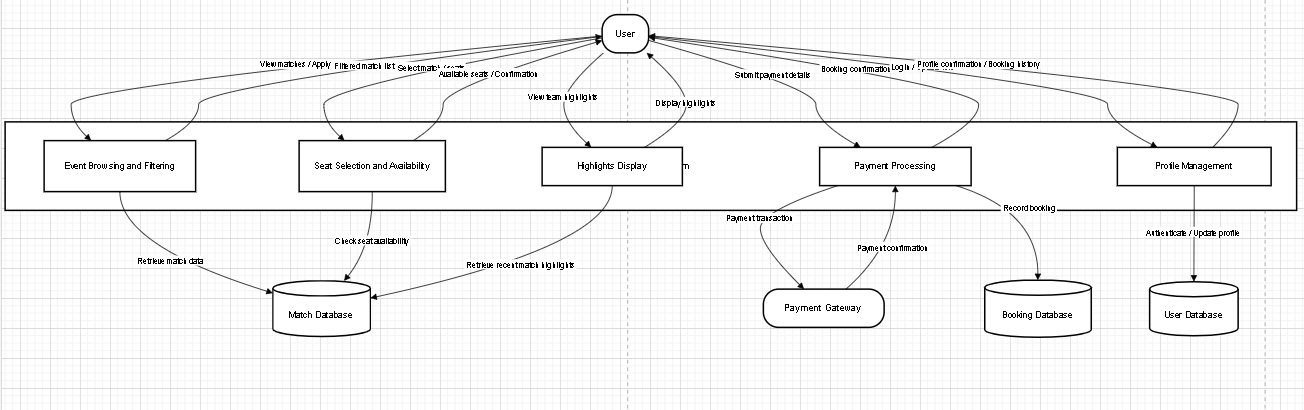
* The software should work on modern web browsers, including Chrome, Firefox, Edge, and Safari.
* The system should be compatible with mobile devices.

**5. System Models**

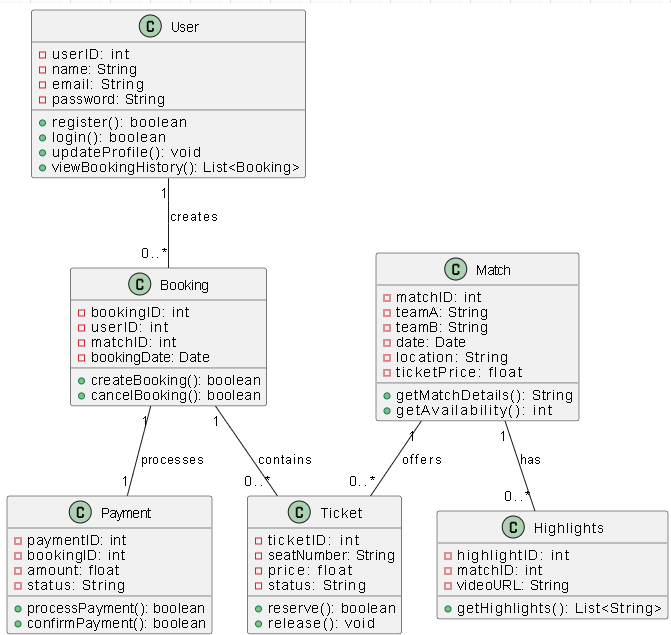
**5.1 Use Case Diagrams**



**5.2 Data Flow Diagrams**



**5.3 Class Diagrams**



**6. External Interface Requirements**

**6.1 User Interfaces**

* The system should have an intuitive UI with the following major components:
  1. **Home screen**: Displays the upcoming football matches, recent highlights, and team store and allows users to browse matches, view real-time ticket availability, and access account login or registration options.
  2. **dashboard**: Logged-in users can access features for managing their bookings, checking booking history, and updating their profiles. For admin users, there are tools available for overseeing events, user accounts, and administrative configurations.

**6.2 API Interfaces**

* RESTful APIs will manage functionalities like data retrieval, ticket booking, and user profile updates.

**6.3 Hardware Interfaces**

* No hardware will used

**7. Other Requirements**

**7.1 Legal and Regulatory Requirements**

* The system must comply with regulations like GDPR, HIPAA

**7.2 Documentation Requirements**

* Provide user manuals and API documentation for developers.

**7.3 Data Backup Requirements**

* The system should back up data daily and store backups for 30 days

**8. Conclusion**

This SRS document outlines the necessary functional and non-functional requirements for Ticket Arena. By adhering to these specifications, the development team will create a system that meets the needs of all stakeholders.