**Use Case Document**

My League

Viet Nam

10/22/2024

**INTRODUCTION**

The Use Case Document is a business document which provides a story of how a system, and its actors, will be utilized to achieve a specific goal. An effective Use Case should provide a detailed step-by-step description of how the system will be used by its actors to achieve the planned outcome. The purpose of the Use Case is to tie the business needs of the system to the design parameters of the system to ensure that the completed system achieves the goals established by the business requirements. The level of detail in Use Cases may vary greatly depending on the size and complexity of the system being designed.

**EXPLAINATION OF USE CASE CONTENTS**

Use Case formats and contents may vary based on system requirements, organizational standards, or unique situations. However, a majority of Use Cases consist of some fundamental contents which may be applied across a wide range of system types. This section will provide explanations for each section of the Use Case.

Name of Use Case: Provide a short name for the use case which should lend itself to the objective of the system.

Description: This section should provide a description of both the reason for using the use case and the expected outcome of the use case.

Actors: Actors may be primary or secondary. Primary actors are the people who will be initiating the system described in the use case. Secondary actors are those will participate in the completion of the use case.

Precondition: This section should describe any conditions that must be true or activities that must be completed prior to executing the use case.

Postcondition: This section should described the state of the system at the conclusion of the use case. Postconditions may include conditions for both successful and unsuccessful execution of the use case.

Flow: This section should describe all actions of the user and the expected system responses for planned normal execution of the use case. The description should be sequential and provide adequate detail to understand all user actions and system responses.

Alternative Flows: Many use cases have varying or special extensions or conditions which are separate from the main flow but also necessary. Alternative flows are usually the result of options or exceptions built into the use case which may alter the primary flow.

Exceptions: When use cases are executed, there may be various conditions which result in errors. This section should describe any errors that may result during use case execution and how the system will react or respond to those errors.

Requirements: This section should describe any non-functional or special requirements for the system as the use case is executed. These requirements may consist of legal or regulatory requirements, quality standards, or organizational requirements that are outside of the functional requirements the system is expected to perform.

**USE CASE**

|  |  |  |  |
| --- | --- | --- | --- |
| Name of Use Case: | My League | | |
| Created By: | X Team | Last Updated By: | Cảnh |
| Date Created: | 10/22/2024 | Last Revision Date: | 10/23/2024 |
|  |  | | |
| Description: | Client using system for managing tournament, account, pitch, schedule, team, and operate match. | | |
| Actors: | Admin, Pitch Owner, User, Referee. | | |
| Preconditions: | 1. [System initialization] System must be online and operational, 2. [System initialization] Each actor must be successfully logged into the system using valid credentials. 3. [Actor-Specific] [Admin] Admin accounts must be created and assigned with appropriate permissions. 4. [Actor-Specific] [Admin] Admin must have set up tournament-related data such as teams, pitches, and schedules. 5. [Actor-Specific] [Pitch Owner] Pitch information, including location and availability, must be configured in the system. 6. [Actor-Specific] [Pitch Owner] Pitch owners must be assigned specific pitches for management. 7. [Actor-Specific] [User (Participant)] Users must be registered with valid profiles. 8. [Actor-Specific] [User (Participant)] Users must be assigned to teams and linked to tournaments or matches. 9. [Actor-Specific] [Referee] Must be assigned to matches. 10. [Actor-Specific] [Referee] Must be authorized to enter match results and manage match-related data. 11. [Entity Management] [Tournaments, Account, Pitches, Teams]: All entities such as tournaments, accounts, and teams must be set up by the admin before other actors can interact with them. 12. [Entity Management] [Tournaments] A valid tournament schedule must be defined by the admin. 13. [Entity Management] [Matches] Must be scheduled and linked to relevant pitches, teams, and referees. 14. [Entity Management] [Matches] Pitch availability must be verified and confirmed. 15. [System Permission] Each actor must have appropriate access rights based on their role to manage or operate relevant parts of the system. | | |
| Postconditions: | 1. Successful Execution:   - The tournament data is correctly saved and accessible to all relevant actors.  - Match results entered by the referee are saved, and the tournament standings are updated.  - Pitch availability is updated, reflecting any scheduled or completed matches.  - User accounts, teams, and schedules are updated as needed.   1. Unsuccessful Execution:   - If the tournament creation fails, no data is saved, and the system prompts the Admin to correct errors.  - If the match results cannot be saved due to invalid inputs, the system prompts the Referee to re-enter the correct data.  - If a user or actor logs out without completing an action, the system discards unsaved changes and resets to the previous state. | | |
| Flow: | 1. Admin logs in:   - Admin navigates to the login page.  - Admin enters valid credentials and submits.  - System authenticates the admin and grants access to the dashboard.   1. Admin manages tournaments:   - Admin selects the “Tournaments” section.  - Admin creates a new tournament by entering details like name, teams, schedule, and pitches.  - System saves the tournament and links it to appropriate teams and matches.   1. Pitch Owner updates pitch availability:   - Pitch Owner logs in and navigates to “Pitch Management”.  - Pitch Owner updates availability based on the current schedule.  - System confirms and displays the updated schedule.   1. User (participant) views match schedule:   - User logs in and navigates to “My Matches”.  - System displays the upcoming matches and related information like date, time, and pitch.   1. Referee submits match results:   - Referee logs in, navigates to the assigned match, and submits match results.  - System verifies the input and updates the tournament standings. | | |
| Alternative Flows: | 1. Admin creates an incomplete tournament:   - Admin forgets to assign teams or a schedule.  - System prompts the Admin to complete all mandatory fields before saving the tournament.   1. User views a canceled match:   - A match gets canceled due to pitch unavailability.  - System displays a “Canceled” status next to the match in the user’s schedule.   1. Referee is unavailable:   - Referee fails to log match results due to system outage.  - Admin assigns a new referee or reschedules the match. | | |
| Exceptions: | 1. Login failure:   - Actor enters invalid credentials.  - System displays an error message and prompts for correct credentials.   1. Pitch double-booking:   - Admin schedules two matches at the same time on the same pitch.  - System detects the conflict and prevents the match from being saved.   1. Data validation failure:   - Referee enters an incorrect match result format (e.g., alphabet instead of numbers).  - System displays a validation error and requests correction. | | |
| Requirements: | 1. Performance: The system should handle up to 1,000 simultaneous users without lag 2. Security: All user data must be encrypted, and two-factor authentication should be available for admins. 3. Legal Compliance: The system must comply with local data protection regulations (e.g., GDPR) 4. Availability: The system must be available 99.9% of the time with automated backups. | | |