Software Requirements Specification

for

The System for Production Of Recreational Team Scheduling (SPORTS)

Version 2.0 (Project 2)

Prepared by Wendy Meng, Evan Pochtar, Timothy Tu, and William Wang

Group 5

11/1/2024

0 Use Cases

Use Case Name: Concession Regulation by Venue (i.e. Alcohol)
Iteration: Filled
Summary: Venue managers regulate concessions based on venue-specific rules, including restrictions on alcohol sales.

Basic Course of Events:

1. The system retrieves venue-specific concession policies and accesses city regulations through an integrated database or external regulatory API.
2. The vendor requests concession approval at a specific venue.
3. The system checks if alcohol sales are allowed based on the venue's policies and city regulations.
4. The system notifies the vendor of approval or denial for concessions.
5. If approved, the concession is linked to the venue and stored in the system.

Alternative Paths:
None
Exception Paths:
In step 3, if alcohol sales are not permitted at the venue, the system denies the concession and notifies both the vendor and venue manager.
Extension Points:None
Trigger:A vendor requests approval for concession sales at a venue.
Assumptions:The system keeps track of venue-specific concession policies and city regulations.
Precondition:The vendor has applied for concession approval and submitted all necessary details.
Postcondition:The system enforces concession regulations and notifies the vendor and venue manager of approval or denial.
Author: Timothy Tu
Date: October 5, 2024

Use Case Name: Maintain Persistent Storage Including Records of Leagues and Regulations
Iteration: Filled
Summary: The system stores persistent records of leagues, including team data, schedules, and regulations.
Basic Course of Events:

1. An administrator inputs or updates league-specific regulations.
2. The system stores the regulations, along with team rosters and schedules, in persistent storage.
3. The system tracks any changes or updates to the regulations.
4. Data is retained for at least one season after the league's completion, then archived.

5. Historical league data is available for audit or future reference.

Alternative Paths:None
Exception Paths:None
Extension Points:None
Trigger**:** A league manager or authorized administrator needs to store or update league regulations.
Assumptions:The system maintains and tracks regulations, team rosters, and schedules across multiple seasons.
Precondition:The league data, including regulations, team rosters, and schedules, is available and ready for input.
Postcondition:The system maintains persistent storage of all league-related information, including historical records for future reference.
Author: Timothy Tu
Date: October 5, 2024

Use Case Name: League size constraints
Iteration: Filled.
Summary: Each league has a constraint on its size which is dependent on the specific league.
Basic Course of Events:

1. The team manager selects a league to join
2. The system displays a page for the user to input information
3. The user inputs their team's information
4. The user submits the information

Alternative Paths: None
Exception Paths: In step 1, if the league selected is already at full capacity, the system will display an error message and redirect the user to the league selection page.
Extension Points: None
Trigger: The user wishes to join a league.
Assumptions: The system keeps track of the amount of teams registered in each league.
Precondition: The user has a team and the information of each member.
Postcondition: The system accepts the user's registration and the user is now part of the league they signed up for.
Author: Wendy Meng
Date: October 5, 2024

Use Case Name: Payments and authorizations needed for league registration
Iteration: Filled.
Summary: The user registers for a league and makes the correct payments with the correct authorizations
Basic Course of Events:

1. The user selects a league
2. The user inputs the team's information and submits it

3. The system asks the user for the necessary authorizations.
4. The user uploads the needed documents
5. The system displays a payment page with the price dependent on the type of league.
6. The user makes the payment.

Alternative Paths: In step 4, if the user does not upload all the required documents and tries to proceed, the system will return to the registration page
Exception Paths: In step 6, if the payment method is incorrect, the system will display an error message and allow the user to try again.
Extension Points: None
Trigger: The user wishes to register their team
Assumptions: The user has a payment method that the system supports.
Precondition: The system is able to verify if the user’s payment method is valid.
Postcondition: The payment method is valid and the user completes the registration for their team.
Author: Wendy Meng
Date: October 5, 2024

Use Case Name: Update Game Schedule Due to Weather
Summary: This use case handles rescheduling games when weather conditions do not permit games to proceed as planned.
Basic Course of Events:

1) System detects adverse weather conditions for a scheduled game.
2) System notifies users (team managers, venue owners) of the issue.
3) The user is prompted to choose a new date and time based on available venues and weather forecast.
4) System reschedules the game after checking for potential conflicts.
5) All stakeholders are informed of the updated schedule.

Alternative Paths: Game is canceled instead of rescheduled.
Exception Paths: No venues are available for rescheduling within a reasonable timeframe.
Trigger: Weather forecast predicts unsuitable conditions for an outdoor game.
Assumptions: System integrates with a reliable weather forecasting service.
Precondition: A game has been scheduled, and weather data is being actively monitored.
Postcondition: Game is rescheduled or canceled, and all users are updated.
Author: Evan Pochtar
Date: October 5, 2024

Use Case Name: Enforce Region-Based League Eligibility
Summary: This use case ensures that teams can only register for leagues within their designated geographic region.
Basic Course of Events:

1) User (team manager) attempts to register a team for a league.
2) System prompts the user to provide the team's location information.
3) System verifies that the team’s location matches the allowed regions for the selected league.

4) If eligible, the team is successfully registered.

Alternative Paths: User selects a different league that aligns with their region.
Exception Paths: System detects a mismatch between team location and league region.
Trigger: Team manager attempts to register for a league.
Assumptions: The system maintains accurate geographic data for both teams and leagues.
Precondition: The league has defined region restrictions.
Postcondition: The team is either registered successfully or prevented from registering due to regional restrictions.
Author: Evan Pochtar
Date: October 5, 2024

Use Case Name: Schedule Game Without Venue Conflict
Summary: This use case ensures that games are scheduled without conflicts at a venue by checking existing reservations and weather conditions before finalizing the schedule.
Basic Course of Events:

1) User (league manager) selects the desired league and sport.
2) System presents a list of available venues and time slots based on the game type.
3) User selects a venue and time.
4) System checks for conflicts with other games scheduled at the same venue.
5) If no conflict is found, the game is scheduled.

Alternative Paths: User changes the venue or time after seeing conflict warnings.
Exception Paths: The selected venue is unavailable (Example: construction, emergency closure).
Trigger: User attempts to schedule a game.
Assumptions: The system has access to real-time scheduling data, weather forecasts, and venue status.
Precondition: The league has been created, and the teams are registered.
Postcondition: The game is either scheduled successfully and displays in the database or is prevented from registering due to venue unavailability
Author: Evan Pochtar
Date: October 5, 2024

Use Case Name: Management of leagues and teams
Summary: This use case allows stakeholders to create constraints for multiple different sports leagues and for the teams within those leagues, as well as being able to control features of leagues and teams
Basic Course of Events:

1) Stakeholder goes into the application that houses all current leagues and teams within them
2) Stakeholder can create new constraints for said entities like removing teams from a league or limiting the amount of people on one team
3) Stakeholder can also modify old constraints like changing the maximum number of players on a team

Alternative Paths: None

Exception Paths: Stakeholder does not have the correct permissions so the options to change would not be visible
Trigger: The stakeholder wants to make changes to leagues/teams
Assumptions: The system software is able to accommodate constraints created and shows current leagues and teams
Precondition: The leagues are set up alongside all current teams
Postcondition: The system has new conditions or constraints
Author: William Wang
Date: October 7, 2024

Use Case Name: Variations of leagues
Summary: This use case ensures that the system contains a wide variety of leagues available based on sport or qualifications to enter and can fit more if needed
Basic Course of Events:

1) User enters the application
2) The application should show all open leagues and the sports that each league covers as well as information including co-ed, sport, and day of the week

Alternative Paths: None
Exception Paths: User attempts to join a league that is already full or tries to join a league that they are ineligible for which would display an error message
Trigger: Stakeholder wants to create a new league or add a new league
Assumptions: The system can accommodate new leagues and shows all current sports and leagues
Precondition: The leagues are set up alongside all current teams
Postcondition: The system displays all sports leagues as well as any new ones added
Author: William Wang
Date: October 7, 2024

1. Introduction

1.1 Purpose

This document outlines the software requirements for the System for Production of Recreational Team Scheduling (SPORTS). The scope includes the features necessary to manage sports leagues, scheduling, payments, and venue usage. This covers core functionalities needed for league registration, game scheduling, and regulation compliance across different sports and venues.

1.2 Document Conventions

Requirements are presented below with numbered headings, talking about each section of the requirements. Bold text is used to emphasize important details. Each section is created to highlight a different area of the software.

1.3 Intended Audience and Reading Suggestions

This document is intended for developers, project managers, city officials, parks and rec staff, and league managers. Readers are encouraged to start with the Introduction and Product Scope sections to get an overview of the system, followed by the detailed functional and non-functional requirements. Developers should focus on technical sections like system architecture and data management.

1.4 Product Scope

SPORTS is designed to help with the management of recreational sports leagues and tournaments, offering things such as league registration, payment processing, venue scheduling, and conflict avoidance. The system works to make operations easier for venue owners, players, and league officials, with the goal of improving user experience and reducing work needed.

1.5 References

- 2024 Fall-CSCI5801-Assignment1RequirementsDeliverable
- Elicitation Session

2. Overall Description

2.1 Product Perspective

The System for Production of Recreational Team Scheduling (SPORTS) is a system designed to meet the needs of recreational sports leagues. It's not replacing any existing system but works to combine fragmented solutions currently used in the scene. SPORTS consolidates league management, game scheduling, venue use, and payments into one system.

2.2 Product Functions

- League management: Create, update, and manage leagues and teams.
- Registration and payment: Allow players and teams to register for leagues and process payments securely.
- Venue scheduling: Schedule games and avoid conflicts based on location, time, and weather conditions.
- Game analytics: Record match outcomes, standings, and statistics for teams.
- Regulation adherence: Ensure league and venue rules are followed, including special concessions like alcohol permissions.
- Weather alerts: Integrate weather updates to warn of event disruptions.

2.3 User Classes and Characteristics

- Players: Generally nontechnical, using the system primarily for registration, payment, and viewing schedules.
- Team Managers: Have a moderate technical skill level, responsible for registering teams, organizing rosters, and managing schedules.

- Venue Owners: Use the system for scheduling, facility management, and overseeing game-day operations.
- Parks & Rec Staff: Technically proficient, responsible for overseeing league operations, scheduling, and enforcing city regulations.
- City Officials: Primarily concerned with enforcing city-wide regulations, monitoring concession compliance, and handling appeals or grievances related to games.

2.4 Operating Environment

SPORTS will operate as a web-based application. The backend will use the Java programming language, and work with any type of operating system that has a browser. Compatible browsers include Google Chrome, Firefox, Safari, and Microsoft Edge.

2.5 Design and Implementation Constraints

The system must adhere to local city regulations for recreational sports, including alcohol permissions at venues and weather related game postponements. It will also need to integrate with existing payment methods. Additionally, the system must handle real time updates for venue schedules and standings.

2.6 User Documentation

- Tutorials for all users to guide them through the basics of the site.
- All documentation will be available in PDF format.

2.7 Assumptions and Dependencies

- Third-party weather APIs for real-time weather data.
- Payment processing APIs of some kind that integrate with different payment methods.
- Cloud storage providers for the data generated by leagues, games, and users.
- Dependencies include city governing sports league operations and the availability of venue data, which have to be kept up to date for accurate scheduling.
- A sufficient amount of administrative users will use the site and update the data/schedules on the site.
- All team managers will use the site to sign up for leagues.

3. External Interface Requirements

3.1 User Interfaces

- Screen layout: Each major section will have a consistent layout with a sidebar for navigation and a central content area for forms, tables, and charts.
- Standard buttons and functions: Common buttons such as Submit, Cancel, and Back will appear consistently across all screens.
- Error messages: When users encounter errors (invalid registration data or scheduling conflicts), descriptive error messages will appear at the top of the page.
- User-specific views: The system will present different interfaces based on user roles. For example, team managers will see options for registering teams and checking schedules, while umpires will have access to game assignments.

3.2 Hardware Interfaces

The system will not require specialized hardware. It will run on standard desktop computers, laptops, tablets, and smartphones. The following hardware interactions are supported:

- Input devices: Users can interact with the system via keyboards, mice, and touchscreens.
- Printers: Users may need to print schedules, registration forms, or reports, so basic printing capabilities will be supported through standard printers.

3.3 Software Interfaces

- Database: The system will use a database to store league data, team registrations, schedules, payments, and game statistics. It will handle queries for reading and writing data in real-time.
- Payment system: Integrates with common payment methods such as card or paypal for secure payment processing. Data between the system and the payment system include payment requests and confirmation of successful payments.
- Weather API: The system will communicate with an external weather service to pull weather data that can cause a warning for game scheduling in case of severe weather conditions.

3.4 Communications Interfaces

The system will support several communication methods:

- HTTP: All communication between the users and the server will use HTTPS for secure data transmission.
- Web-based forms: Users will fill out electronic forms for registration, schedule changes, etc., directly on the website. The forms will be processed as quickly as possible, so that users receive immediate feedback on the success of their submissions.
- Data transfer: The system will sync data across components with low latency, especially when interacting with external systems. Data transfer will follow standard formats like JSON.

4. System Features

4.1 Concession Regulation by Venue

4.1.1 Description and Priority

This function requires the concessions provided at a certain venue to be regulated by that venue. The regulations are based on local laws and the leagues that the venue hosts. This function is of medium priority with the benefit at 5, penalty at 7, cost at 2, and risk at 7.

4.1.2 Stimulus/Response Sequences

The system retrieves venue-specific concession policies and city regulations. The vendor requests concession approval at a specific venue. The system checks if alcohol sales are allowed based on the venue’s policies and city regulations. The system notifies the vendor of approval or denial for concessions. If approved, the concession is linked to the venue and stored in the system.

4.1.3 Functional Requirements

REQ-1: If the system does not retrieve venue-specific concession policies or city regulations, alcohol sales will not be allowed

REQ-2: The system must take in concession policies and regulations from the venue manager

REQ-3: The system must contact concessions that adhere to the provided policies

4.2 Maintain Persistent Storage Including Records of Leagues and Regulations

4.2.1 Description and Priority

The system stores persistent records of leagues, including team data, schedules, and regulations. The system also keeps track of the regulations for each league. The system will keep this information in its archive even when that league is no longer active. This feature is of high priority with the benefit: 8, penalty: 3, cost: 4, risk: 3.

4.2.2 Stimulus/Response Sequences

The user inputs or updates league-specific regulations. The system stores the regulations, along with team rosters and schedules, in persistent storage. The system tracks any changes or updates to the regulations. Data is retained for at least one season after the league's completion and then archived. Historical league data is available for audit or future reference.

4.2.3 Functional Requirements

REQ-1: The system must store each team and its members that registers for a league

REQ-2: The system must store the schedules, rankings, and scores of each league

REQ-3: The system updates the information that is already stored if it is changed

REQ-4: The system will not store information about teams that did not register correctly

4.3 League size constraints

4.3.1 Description and Priority

This function ensures that each league has a constraint on its size which is dependent on the specific league. This function has high priority. Benefit: 7, penalty: 8, cost: 7, risk: 4.

4.3.2 Stimulus/Response Sequences

The user selects a league to join. The system displays a page for the user to input information. The user inputs their team's information. The user submits the information

4.3.3 Functional Requirements

REQ-1: The system must keep track of how many teams are in each league

REQ-2: When the league is full, the system must not allow the user past the league selection page.

REQ-3: When the league is not full, the system must allow the user to register their team

4.4 Payments and authorizations needed for league registration

4.4.1 Description and Priority

This function ensures that the necessary payments and authorizations are received for a valid league registration. This function has high priority. Benefit: 9, penalty: 8, cost: 7, risk: 3.

4.4.2 Stimulus/Response Sequences

The user selects a league. The user inputs the team's information and submits it. The system asks the user for the necessary authorizations. The user uploads the needed documents. The system displays a payment page with the price dependent on the type of league. The user makes the payment .

4.4.3 Functional Requirements

REQ-1: The system must keep track of what authorizations are needed for each league as well as the amount needed to be paid

REQ-2: When the user inputs an incorrect payment method, the system must not allow the user to register.

REQ-3: When the user does not have the authorizations needed, the system must not allow the user to register their team.

4.5 Enforce Region-Based League Eligibility

4.5.1 Description and Priority

This function ensures that teams can only register for leagues within their designated geographic region. This function has low priority. Benefit: 4, penalty: 4, cost: 4, risk: 3.

4.5.2 Stimulus/Response Sequences

The user (team manager) attempts to register a team for a league. The system prompts the user to provide the team's location information. The system verifies that the team's location matches the allowed regions for the selected league. If eligible, the team is successfully registered.

4.5.3 Functional Requirements

REQ-1: The system must use location services to locate the region of the user

REQ-2: When the user selects a league outside of their geographic region, the system will notify the user through a popup that the league is outside of their region and direct the user back to the league selection page.

REQ-3: The system must store the location of each league with the rest of the league information which is in a TBD.

4.6 Schedule Game Without Venue Conflict

4.6.1 Description and Priority

This function ensures that games are scheduled without conflicts at a venue by checking existing reservations and weather conditions before finalizing the schedule. This function has high priority. Benefit: 7, penalty: 8, cost: 7, risk: 7.

4.6.2 Stimulus/Response Sequences

The user (e.g., team manager) selects the desired league and sport. The system presents a list of available venues and time slots based on the game type. The user selects a venue and time. The system checks for conflicts with other games scheduled at the same venue. If no conflict is found, the game is scheduled.

4.6.3 Functional Requirements

REQ-1: The system must store all of the current season’s scheduled games in a secure database (TBD)

REQ-2: When the user selects a venue and time slot that conflicts with currently scheduled games, the system will not allow the user to schedule that game and will display an error message in a popup, informing the user of the conflict.

REQ-3: The system must consistently keep track of the unscheduled games and display them for the user to schedule.

4.7 Management of leagues and teams

4.7.1 Description and Priority

This function allows stakeholders to create constraints for multiple different sports leagues and for the teams within those leagues, as well as being able to control features of leagues and teams. This function has low priority. Benefit: 5, penalty: 4, cost: 4, risk: 3.

4.7.2 Stimulus/Response Sequences

Stakeholders go into the application that houses all current leagues and teams within them. Stakeholders can create new constraints for said entities like removing teams from a league or limiting the amount of people on one team. Stakeholders can also modify old constraints like changing the maximum number of players on a team

4.7.3 Functional Requirements

REQ-1: The system must recognize who the stakeholders are and give them the permission to be able to edit constraints.

REQ-2: When the stakeholders edit constraints, the teams and users affected by the changes will be notified of the changes through email and will be given possible next steps.

REQ-3: Changes made by the stakeholders must not contradict overarching league regulations

4.8 Variations of leagues

4.8.1 Description and Priority

This function ensures that the system contains a wide variety of leagues available based on sport or qualifications to enter and can fit more if needed. This function has medium priority. Benefit: 7, penalty: 6, cost: 7, risk: 4.

4.8.2 Stimulus/Response Sequences

The user enters the application. The application should show all open leagues and the sports that each league covers as well as other necessary information like co-ed, etc

4.8.3 Functional Requirements

REQ-1: The system must keep a record of all leagues and variations that are available

REQ-2: When the user selects a certain league without adhering to the necessary constraints for that league, the system must notify the user through a popup message and redirect them to the league selection page

REQ-3: The system must have the same types of leagues (co-ed, female, male, etc.) available for each sport

4.9 Update Game Schedule Due to Weather

4.9.1 Description and Priority

This function ensures that games are rescheduled when weather conditions do not permit games to proceed as planned. This function has high priority. Benefit: 8, penalty: 8, cost: 7, risk: 7.

4.9.2 Stimulus/Response Sequences

The system detects adverse weather conditions for a scheduled game. The system notifies users (team managers, and venue owners) of the issue. The user is prompted to choose a new date and time based on available venues and weather forecasts. The system reschedules the game after checking for potential conflicts. All stakeholders are informed of the updated schedule.

4.9.3 Functional Requirements

REQ-1: The system must keep a record of the contact information of all users

REQ-2: When a game is canceled, an email will be sent out to the affected users notifying them of the cancellation and any rescheduling information.

REQ-3: The system must allow the admin to post announcements for cancellations and rescheduling of games on the main page of the website.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The system shall handle league registration for up to 500 concurrent users without performance degradation.
- Venue scheduling must occur in real-time with conflict checking completed within 3 seconds to prevent double booking of fields.
- When accessing schedules, standings, or making game updates (e.g., umpire reports), the system shall respond within 2 seconds for all users, including managers, officials, and players.
- Updates made to a schedule (e.g., rescheduling games or cancellations) must update all team managers via email and update the web interface within 1 minute.

5.2 Safety Requirements

- The system shall ensure that sensitive data such as player contact information, payment details, and schedules is securely stored and only accessible by authorized users.
- Any payment processing errors shall immediately log the error and notify both the user and an admin, preventing double charges or incomplete registrations.
- In the event of an emergency, such as field closures due to unsafe weather conditions, the system must send notifications to all relevant people participating.

5.3 Security Requirements

- The system shall use secure HTTPS for all data transmissions to protect user information during online registration and game updates.

- User access shall be role-based, with different permission levels for players, managers, umpires, Parks & Rec staff, and city officials. Each role will have access only to the data and actions relevant to their role.
- User authentication shall be required for all actions involving sensitive data, such as registration, payment processing, and roster updates.
- Data retention policies shall ensure that personal information is archived or anonymized after the completion of a season, adhering to privacy regulations.

5.4 Software Quality Attributes

- Reliability: The system must achieve 99.9% uptime to ensure users can always access schedules and game information.
- Usability: The interface should be intuitive enough for users of all technical skill levels. This includes accessible schedules, game rescheduling options, and clear error messages for conflicts in registration or scheduling.
- Maintainability: The system shall be designed with modular components, allowing for easy updates when adding new sports, fields, or other league configurations without requiring a full system overhaul.
- Interoperability: The system must be compatible with external scheduling software to allow seamless booking and updating of field availability.
- Scalability: The system shall handle increasing numbers of leagues, games, and participants without performance degradation, with the capacity to scale up to handle leagues in multiple cities.

5.5 Business Rules

- Only the Parks & Rec staff have the authority to start a new league, while team managers are responsible for registering their teams and paying league fees.
- Umpires must update game reports within 24 hours of the game’s conclusion, with automated reminders sent if they fail to meet this deadline.
- Teams are automatically placed into leagues based on the availability of players, teams, and fields. Players registering as "free agents" will be directed to specific leagues or teams that need additional members.
- Concessions must be approved by city officials before they can be listed at a venue. These concessions must be scheduled separately from the game schedule to prevent overlap.
- The system must provide the option for teams to request a refund or credit if they cannot pay upfront at registration, and refunds must be processed within the system to avoid manual handling.

6. Other Requirements

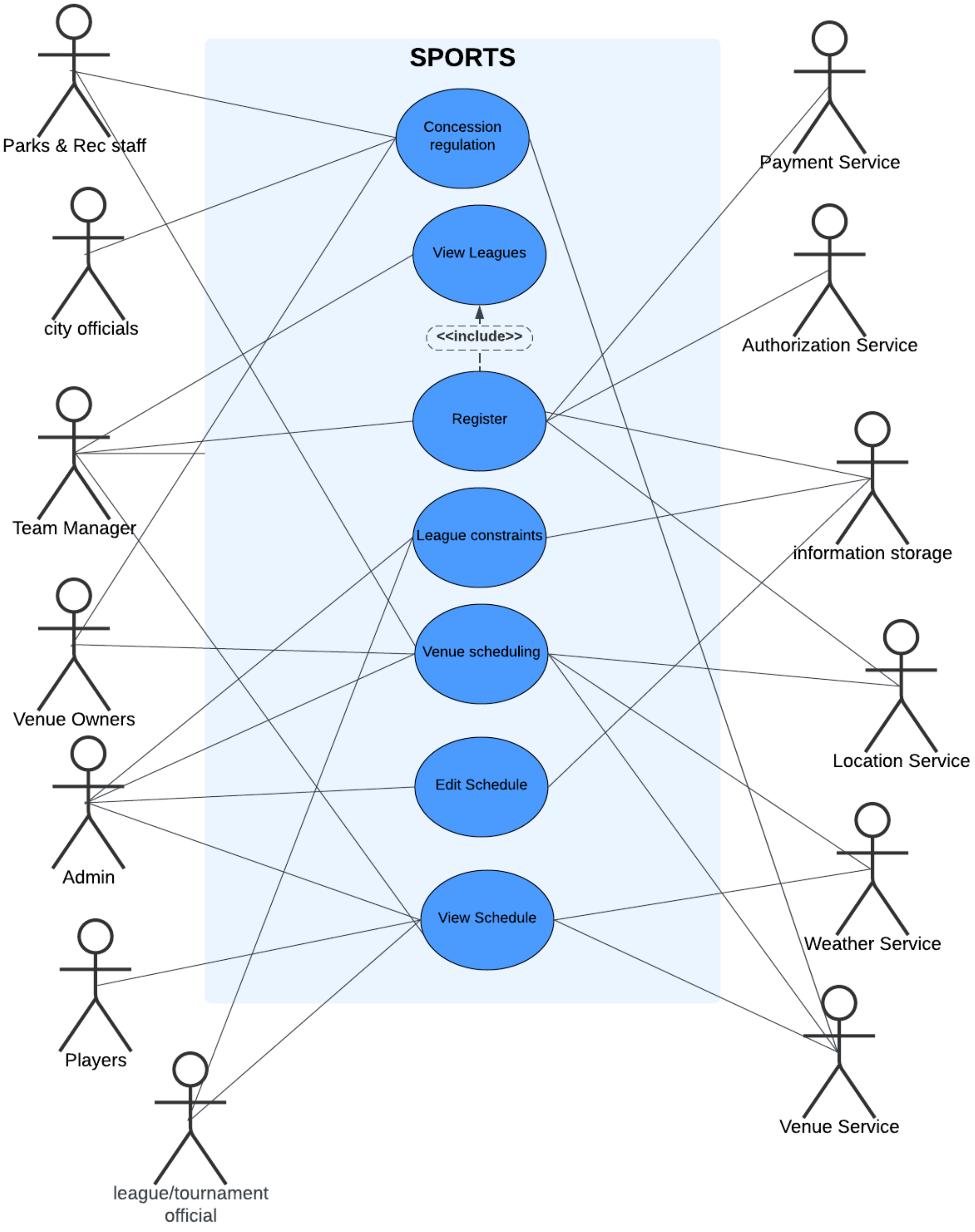
- The system shall comply with all privacy laws and regulations where applicable
- A terms of service shall be provided to users upon registration
- The system shall accurately display game times based on the local time zone of the user

- The system must have multiple language support so as to accommodate the top three most common languages in the United States

Appendix A: Glossary

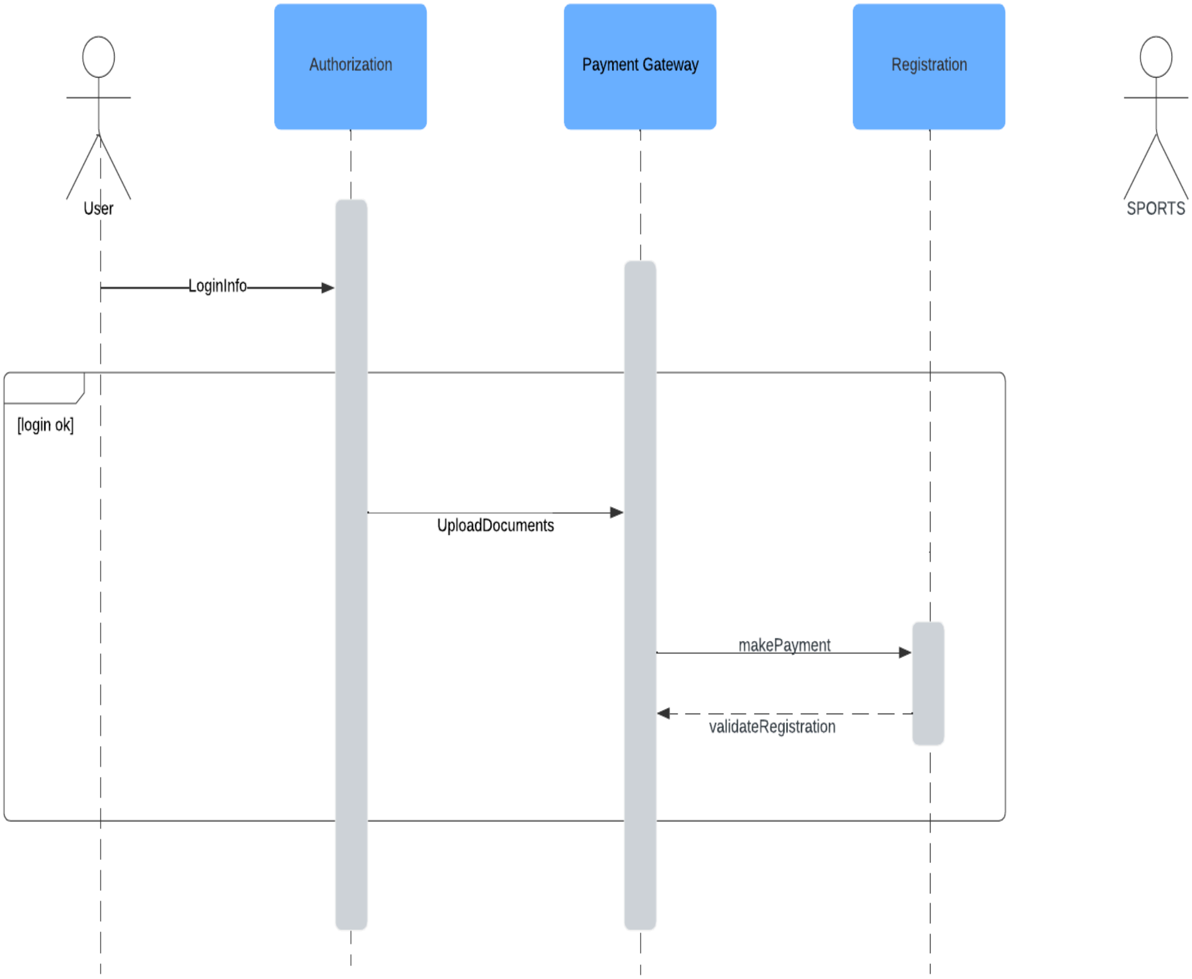
- TBD - To be determined, an item which will be determined at a later date after more information is acquired
- SPORTS - System for Production Of Recreational Team Scheduling, acronym for the application

Appendix B: Use Case Model

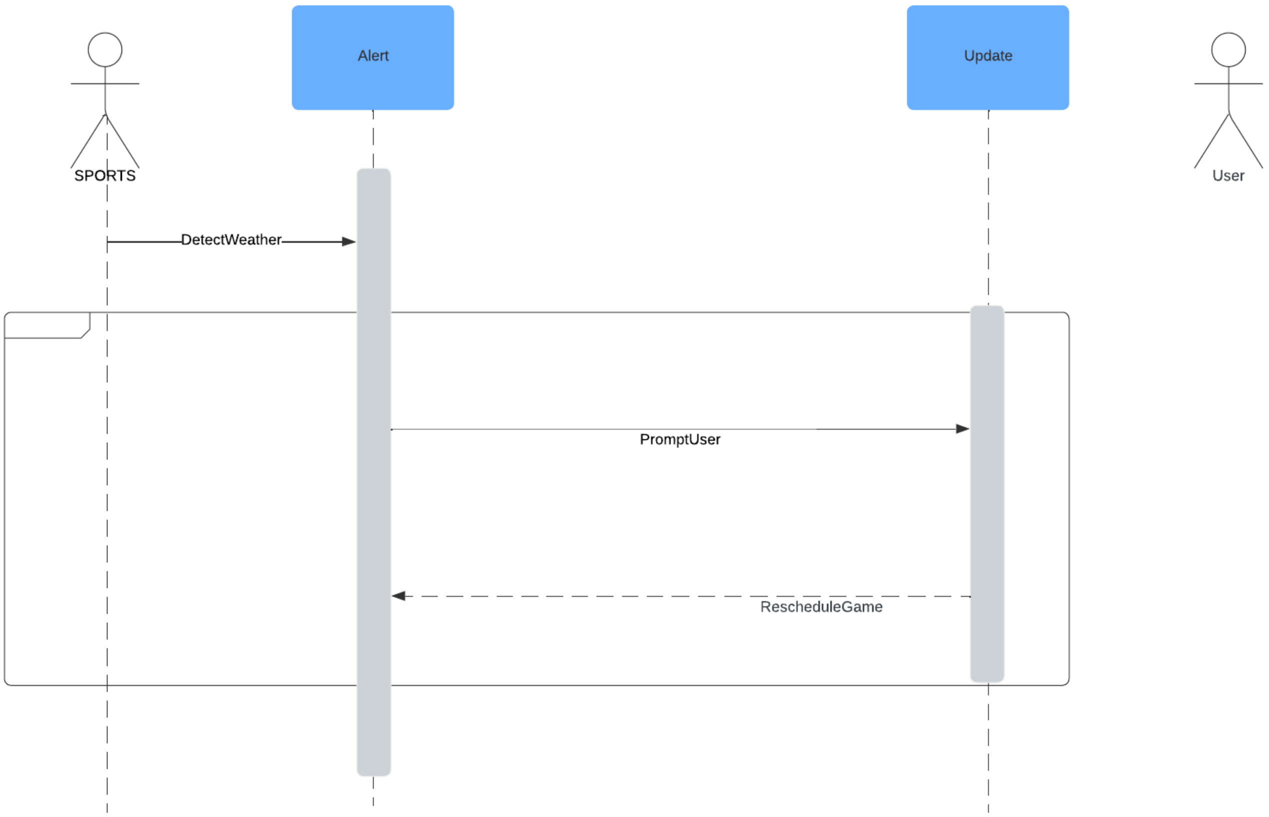


Appendix C: Analysis Models

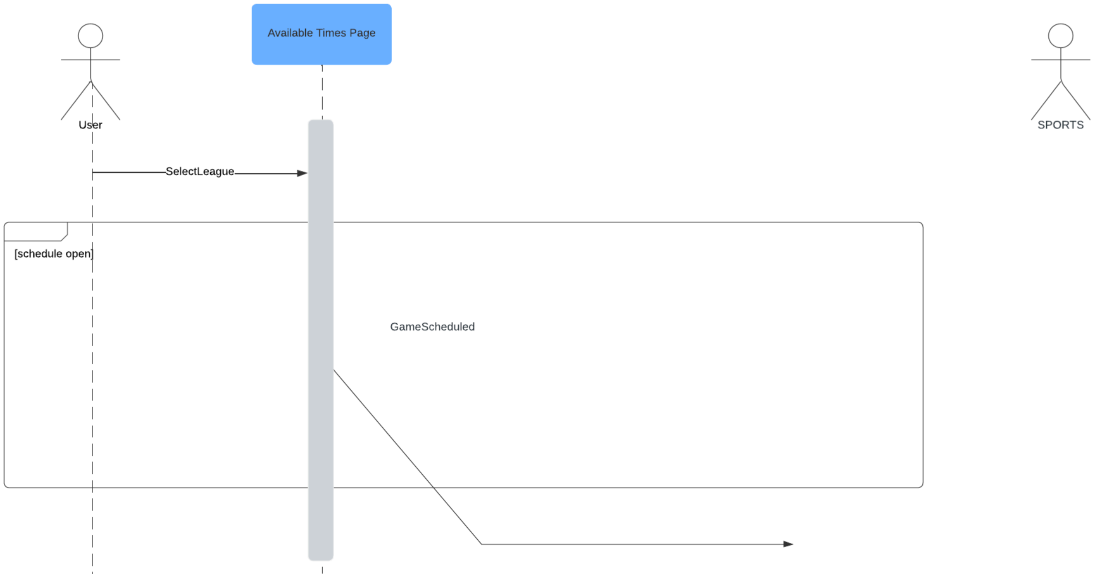
Use Case: Payment

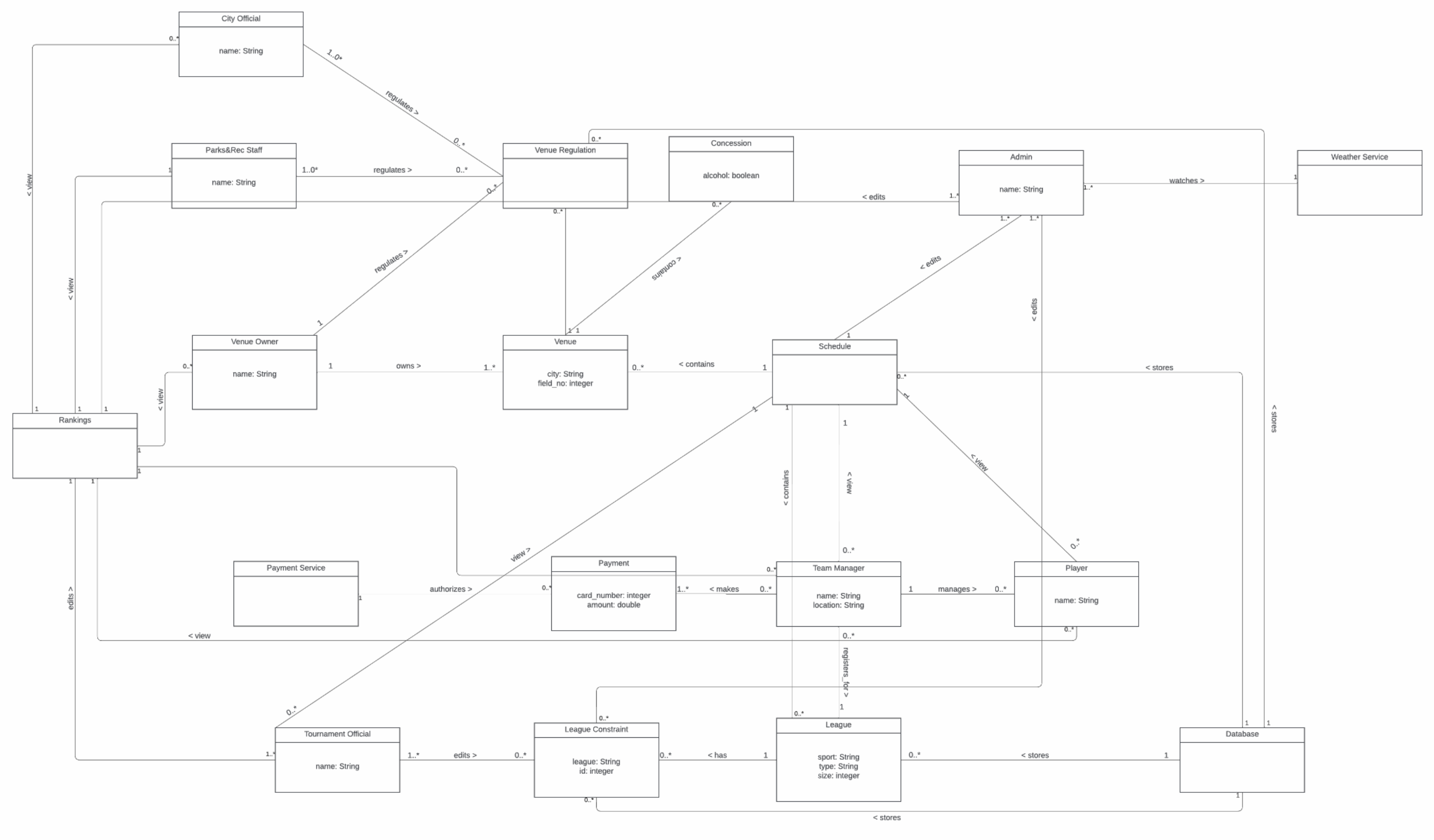


Use Case: Weather



Use Case: Schedule Game without conflict





Appendix C: TBD List

- 4.5.3 REQ-3: The system must store the location of each league with the rest of the league information which is in a TBD.
- 4.6.3 REQ-1: The system must store all of the current season's scheduled games in a secure database (TBD)