

Software Requirements Specification for Software Engineering: McMaster Graduate Softball League Management Platform

Team 6, Pitch Perfect
Damien Cheung
Jad Haytaoglu
Derek Li
Temituoyo Ugborogho
Emma Wigglesworth

April 4, 2025

Contents

0.1	Revision 1 Traceability	vii
1	Template Information	viii
1.1	Template Source	viii
1.2	Major Template Modifications	viii
2	Purpose of the Project	ix
2.1	User Business	ix
2.2	Goals of the Project	x
3	Stakeholders	xii
3.1	Client	xii
3.2	Customer	xii
3.3	Other Stakeholders	xii
3.4	Hands-On Users of the Project	xii
3.5	Personas	xiii
3.6	Priorities Assigned to Users	xiv
3.7	User Participation	xiv
3.8	Maintenance Users and Service Technicians	xiv
4	Mandated Constraints	xv
4.1	Solution Constraints	xv
4.2	Implementation Environment of the Current System	xv
4.3	Partner or Collaborative Applications	xv
4.4	Off-the-Shelf Software	xv
4.5	Anticipated Workplace Environment	xvi
4.6	Schedule Constraints	xvi
4.7	Budget Constraints	xvi
4.8	Enterprise Constraints	xvi
5	Naming Conventions and Terminology	xvii
5.1	Glossary of All Terms, Including Acronyms, Used by Stakeholders involved in the Project	xvii
6	Relevant Facts and Assumptions	xvii
6.1	Relevant Facts	xvii
6.2	Business Rules	xvii

6.3	Assumptions	xviii
7	The Scope of the Work	xviii
7.1	The Current Situation	xviii
7.2	The Context of the Work	xviii
7.3	Work Partitioning	xx
7.4	Specifying a Business Use Case	xxi
8	Business Data Model and Data Dictionary	xxi
8.1	Business Data Model	xxi
8.2	Data Dictionary	xxiii
9	Formal Specifications	xxiii
9.1	State Machine Model	xxiii
9.2	Formal Requirements	xxiv
9.3	Data Flow Specifications	xxv
9.4	Safety Invariants	xxv
10	The Scope of the Product	xxvi
10.1	Product Boundary	xxvi
10.2	Product Use Case List	xxvi
10.3	Individual Product Use Cases (PUC's)	xxviii
11	Functional Requirements	xxx
12	Non-Functional Requirements	xxxvi
12.1	Data Integrity and Backup Requirements	xxxvi
12.2	Session Management Requirements	xxxvi
12.3	Audit Logging Requirements	xxxvii
12.4	Security Testing Requirements	xxxviii
12.5	Schedule Verification Requirements	xxxviii
13	Look and Feel Requirements	xxxix
13.1	Layout Requirements	xxxix
13.2	Visual Feedback Requirements	xxxix
13.3	Appearance Requirements	xxxix
13.4	Style Requirements	xl
14	Usability and Humanity Requirements	xl

14.1	Ease of Use Requirements	xl
14.2	Personalization and Internationalization Requirements	xl
14.3	Learning Requirements	xl
14.4	Understandability and Politeness Requirements	xli
14.5	Accessibility Requirements	xli
15	Performance Requirements	xli
15.1	Speed and Latency Requirements	xli
15.2	Safety-Critical Requirements	xli
15.3	Precision or Accuracy Requirements	xlii
15.4	Robustness or Fault-Tolerance Requirements	xlii
15.5	Capacity Requirements	xlii
15.6	Scalability or Extensibility Requirements	xlii
15.7	Longevity Requirements	xliii
16	Operational and Environmental Requirements	xliii
16.1	Expected Physical Environment	xliii
16.2	Browser Compatibility	xliii
16.3	Requirements for Interfacing with Adjacent Systems	xliii
16.4	Productization Requirements	xliv
16.5	Release Requirements	xliv
17	Maintainability and Support Requirements	xliv
17.1	Maintenance Requirements	xliv
17.2	Supportability Requirements	xliv
17.3	Adaptability Requirements	xlv
18	Security Requirements	xlv
18.1	Access Requirements	xlv
18.2	Integrity Requirements	xlv
18.3	Privacy Requirements	xlv
18.4	Audit Requirements	xlv
18.5	Immunity Requirements	xlvi
19	Cultural Requirements	xlvi
19.1	Cultural Requirements	xlvi
20	Compliance Requirements	xlvi
20.1	Legal Requirements	xlvi

20.2 Standards Compliance Requirements	xlvi
20.3 Industry Best Practices	xlvi
21 Open Issues	xlvi
22 Off-the-Shelf Solutions	xlix
22.1 Ready-Made Products	xlix
22.2 Reusable Components	xlix
22.3 Products That Can Be Copied	xlix
23 New Problems	xlix
23.1 Effects on the Current Environment	l
23.2 Effects on the Installed Systems	l
23.3 Potential User Problems	l
23.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product	li
23.5 Follow-Up Problems	li
24 Tasks	li
24.1 Project Planning	li
24.2 Planning of the Development Phases	liii
24.2.1 Phase 1: Requirement Gathering and Initial Planning	liii
24.2.2 Phase 2: Proof of Concept and Design	liii
24.2.3 Phase 3: Core Development	liii
24.2.4 Phase 4: Verification, Validation, and Revision	liv
24.2.5 Phase 5: Final Development and Demonstration	liv
24.2.6 Phase 6: Documentation and Wrap-Up	liv
25 Migration to the New Product	liv
25.1 Requirements for Migration to the New Product	liv
25.2 Data That Has to be Modified or Translated for the New System	lv
26 Costs	lv
27 User Documentation and Training	lvi
27.1 User Documentation Requirements	lvi
27.2 Training Requirements	lvi
28 Waiting Room	lvi

29 Ideas for Solution	lvii
30 Traceability Matrices	lxiii
30.1 Requirements to Use Cases Matrix	lxiii
30.2 Requirements to Hazards Matrix	lxiv
30.3 Requirements Dependencies Matrix	lxv
31 Appendix	lxvi
31.1 Symbolic Parameters	lxvii
32 Likely Changes	lxvii
32.1 Requirements Likely to Change	lxvii
32.2 Requirements Unlikely to Change	lxviii
32.3 Impact Analysis	lxix

List of Tables

1	Traceability of Revision 1 Changes	vii
2	Requirements to Use Cases Traceability	lxiv
3	Requirements to Hazards Traceability	lxv
4	Requirements Dependencies	lxv

Revision History

Date	Version	Notes
April 3, 2024	1.1	Major revision based on feedback and requirements changes (refer to Revision 1 Traceability table for details)
Oct 11, 2023	1.0	Initial Creation

0.1 Revision 1 Traceability

The following table tracks all significant changes made to the SRS based on feedback, requirements changes, and improvements:

Table 1: Traceability of Revision 1 Changes

Change ID	Description of Change	Affected Section(s)	GitHub Issue
REV1.1	Modified team joining process: Only captains can invite players	Functional Requirements #6	Issue #468
REV1.2	Enhanced schedule view requirements: Added specific views	Functional Requirements #3	Issue #469
REV1.3	Added player profile page requirement	Functional Requirements	Issue #470
REV1.4	Update requirements to align with HA document	Non-Functional Requirements	Issue #473
REV1.5	Added formal specifications with state machine model	Formal Specifications	Issue #474
REV1.6	Enhanced data dictionary with detailed attributes	Business Data Model	Issue #476
REV1.7	Expanded introduction with clearer context	Introduction	Issue #478
REV1.8	Implemented consistent requirement numbering (UC1, FR1, NFR-LF1)	All Requirements	Issue #477
REV1.9	Replaced magic numbers with symbolic constants	All	Issue #471
REV1.10	Added introductory text to all sections	All	Issue #485
REV1.11	Added comprehensive traceability matrices	Traceability	Issue #486

Continued on next page

Table 1 – Continued from previous page

Change ID	Description of Change	Affected Section(s)	GitHub Issue
REV1.12	Enhanced diagram descriptions and references	All	Issue #487
REV1.13	Add hyperlinks to table and figure references	All	Issue #488
REV1.14	Enhanced persona descriptions	Stakeholders	Issue #489
REV1.15	Added quantifiable fit criteria to NFRs	Non-Functional Requirements	Issue #472
REV1.16	Identify requirements that are likely/unlikely to change	Likely Changes	Issue #505
REV1.17	Identify template and changes made	Template Information	Issue #506

1 Template Information

This section documents the template used and modifications made to create this SRS document.

1.1 Template Source

This document follows the Volere Requirements Specification Template by Suzanne Robertson and James Robertson. The initial draft was adapted from <https://github.com/Dieblisch/volere>, with significant modifications to suit the needs of the McMaster GSA softball league platform project.

1.2 Major Template Modifications

The following significant modifications were made to the original Volere template:

1. Removed Sections

- Risks section removed as it is covered in the separate Hazard Analysis document
- Cost estimates section simplified due to academic nature of project
- Solution constraints section modified to focus on technical constraints

2. Added Sections

- Comprehensive symbolic parameters table for configuration values
- Detailed traceability matrices linking requirements, use cases, and hazards
- Formal specifications section with state machine model

2 Purpose of the Project

This section outlines the core business objectives and goals of the McMaster GSA softball league platform, detailing how the system will serve its users and enhance league operations.

2.1 User Business

The primary business of the McMaster GSA softball league is to facilitate recreational softball activities for students, staff, and alumni. The league aims to provide an organized platform for scheduling games, managing teams, and tracking player performance, enhancing the overall experience for participants. By streamlining league operations, the platform will foster a more inclusive and efficient environment, enabling better communication among players, captains, and commissioners.

- Team Management: captains will have the ability to create and manage their teams, inviting players to join, assigning roles, and ensuring that all team members are informed about practices and games.
- Game Scheduling: users will utilize the scheduling tools to view upcoming games, propose new times, and request rescheduling, ensuring that all participants are aware of their commitments and can manage their time effectively.

- **Communication and Announcements:** Commissioners will post league-wide announcements and updates, allowing all users to stay informed about important news, such as changes in scheduling, league policies, or events.
- **Performance Tracking:** players can track their individual and team performance through leaderboards and standings, fostering a sense of competition and motivation to improve.
- **Payment Management:** the application will provide visibility into players' payment statuses, enabling captains and commissioners to easily monitor team compliance with league participation requirements.
- **Community Building:** through features such as announcements and event notifications, users can engage with one another beyond the field, strengthening community ties and enhancing the overall experience of league participation.

2.2 Goals of the Project

The primary goal of the McMaster GSA softball league platform project is to develop an intuitive, user-friendly web application that enhances the management and operation of the league. Specific goals include:

- **Streamlined Scheduling and Communication:** Simplify the scheduling process for games and practices, allowing captains to manage their teams effectively while facilitating clear communication between all stakeholders.
- **Role-Based Access Control (RBAC):** Use a secure login system that allows different levels of access for different users, ensuring that users can perform actions relevant to their roles without compromising sensitive information.
- **Improved User Interface:** Design a modern, responsive user interface that enhances usability and engagement, making it easier for participants to navigate the platform and access essential features.
- **Enhanced Data Management:** Develop robust features for managing player registrations, tracking payment statuses, and maintaining accu-

rate standings and leaderboards, thereby reducing administrative overhead and increasing transparency.

- **Facilitate Community Engagement:** Foster a sense of community among participants through features such as announcements, event notifications, and an organized platform for sharing updates, enhancing the overall experience of league participation.

3 Stakeholders

This section identifies and describes the key stakeholders who will interact with or be affected by the platform, including their roles, needs, and expectations.

3.1 Client

The client for this project is the McMaster GSA, which oversees the softball league. Their primary goal is to provide a robust platform that improves the management of the league, facilitates better communication among users, and enhances the overall user experience. The GSA will benefit from increased participation and smoother operations, ultimately fostering a stronger community at McMaster.

3.2 Customer

The customers are the users of the platform, including players, captains, and commissioners. They seek an intuitive and efficient system that allows for easy management of games, team coordination, and communication. By addressing their needs, the platform aims to increase user satisfaction and engagement within the league.

3.3 Other Stakeholders

Other stakeholders include the university administration, potential sponsors, and the broader McMaster University community. These groups may have a vested interest in the successful operation of the league, as it contributes to student life and community engagement. Sponsors may seek visibility and promotional opportunities through the league's activities.

3.4 Hands-On Users of the Project

Hands-on users include the players, captains, and commissioners who will interact with the platform daily. Players will use the system to join teams, view schedules, and track their performance. Captains will manage their teams, schedule games, and communicate with players. Commissioners will

oversee league operations, post announcements, and manage administrative tasks.

3.5 Personas

- **Player Persona - Margaret Simon**

- **Background:** 2nd-year Masters student in Chemical Engineering
- **Technical Proficiency:** Moderate, comfortable with basic web applications
- **Goals:**
 - * Find and join a team that matches her skill level
 - * Easily check game schedules around her lab work
 - * Track her team’s progress throughout the season
- **Pain Points:**
 - * Limited time due to academic commitments

- **Captain Persona - Nancy Wheeler**

- **Background:** PhD candidate in Physics, 3rd year as team captain
- **Technical Proficiency:** High, experienced with various software tools
- **Goals:**
 - * Efficiently manage team roster and communications
 - * Handle game scheduling and rescheduling requests
 - * Track team performance and player attendance
- **Pain Points:**
 - * Coordinating multiple players’ schedules
 - * Needs streamlined communication tools
 - * Requires quick access to league rules and policies

- **Commissioner Persona - Dr. Philip Leroy**
 - **Background:** Postdoctoral researcher, 5+ years league experience
 - **Technical Proficiency:** Moderate to high, focuses on efficiency
 - **Goals:**
 - * Oversee league operations and maintain fairness
 - * Manage schedule conflicts and weather-related changes
 - * Ensure clear communication of league policies
 - **Pain Points:**
 - * Handling multiple rescheduling requests
 - * Needs efficient tools for league-wide announcements
 - * Requires clear oversight of all team activities

3.6 Priorities Assigned to Users

Players prioritize ease of joining teams, checking their game schedule, and tracking performance. Captains focus on scheduling flexibility, team management features, and communication tools. Commissioners prioritize administrative functionality, announcement features, and overall system reliability.

3.7 User Participation

User participation will be critical throughout the project. Stakeholders will provide feedback during the development process to ensure the platform meets their needs. User testing sessions will be conducted to gather insights and refine the interface and features.

3.8 Maintenance Users and Service Technicians

Maintenance users include any staff responsible for ongoing system updates, bug fixes, and technical support (our team). This group ensures that the platform remains functional, secure, and up-to-date. Service technicians may be

involved in troubleshooting issues and providing assistance to users, particularly during peak usage periods, such as the beginning of the season.

4 Mandated Constraints

4.1 Solution Constraints

Constraints that limit the design of the solution:

- **Scalability:** The initial version of the platform must support [MIN_TEAMS](#) to [MAX_TEAMS](#) teams, with future scalability to accommodate more teams.
- **Device Accessibility:** The system must be optimized for desktops, tablets, and mobile phones.

4.2 Implementation Environment of the Current System

The current implementation environment includes:

- **Operating Systems:** Compatibility with Windows, macOS, Android, and iOS.
- **Supported Browsers:** The platform must be fully functional on Chrome, Firefox, Safari, and Edge.
- **Hosting Environment:** The platform will continue to use the existing hosting environment, which has already been established for the current website. No changes to the hosting infrastructure are anticipated unless new features require additional support.

4.3 Partner or Collaborative Applications

There are no partner or collaborative applications required for this project.

4.4 Off-the-Shelf Software

No off-the-shelf software will be utilized for payment handling or calendar integration.

4.5 Anticipated Workplace Environment

The platform is anticipated to be used in various environments:

- **Users:** The platform will be accessed by league commissioners, team captains, players, and administrators.
- **Device Types:** Users are expected to use the platform on personal laptops, desktops, and mobile devices.

4.6 Schedule Constraints

The project is constrained by a strict timeline, with key milestones that must be met:

- **Development Timeline:** The platform must be completed within the designated course timeline (approximately 8 months).
- **Fixed Milestones:** Deliverables must be submitted according to pre-defined deadlines.

4.7 Budget Constraints

There are no budget constraints applicable to this project. The existing infrastructure and hosting environment will be reused, and no additional licensing fees are required.

4.8 Enterprise Constraints

- **Hosting Environment:** The platform must utilize the existing hosting environment already set up for the current system, which may come with certain limitations or capabilities that need to be respected.
- **No External Sponsorship:** There is no specific enterprise funding or sponsoring this project, meaning there are no additional budgetary or compliance constraints from an external organization.

5 Naming Conventions and Terminology

5.1 Glossary of All Terms, Including Acronyms, Used by Stakeholders involved in the Project

The following terms are used consistently throughout this document:

- **Commissioner:** Manages the entire league and oversees operations.
- **Captain:** Responsible for team management, including schedules and rosters.
- **Player:** A participant in a team who needs access to schedules and standings.
- **Schedule Slot:** The specific time allocated for a game.
- **Waiver:** A document that participants must sign before joining the league.

6 Relevant Facts and Assumptions

6.1 Relevant Facts

- **Role Definition:** Specific roles such as commissioner, captain, and player exist within the system, each with unique permissions and responsibilities.
- **Season Timeline:** The platform will primarily be used during the softball season from April to September.

6.2 Business Rules

- **Team Registration:** Teams must be registered by captains before the league starts.
- **Scheduling Requests:** Rescheduling requests must be made at least [RESCHEDULE NOTICE HOURS](#) hours in advance.

6.3 Assumptions

- **Internet Access:** All users are assumed to have access to a stable internet connection.
- **Mobile Usage:** It is assumed that many users will access the platform using mobile devices, so a mobile-friendly interface is required.
- **Payment Compliance:** Players are expected to make payments via e-transfer, which will be managed externally. The platform will include an interface for the designated staff member to track which players have completed their payments, without integrating any direct payment gateway. Manually updated payment status is assumed to be correct.

7 The Scope of the Work

7.1 The Current Situation

The current implementation of the GSA softball league platform is run on outdated software. This issue makes the management of the league platform difficult without a certain degree of programming knowledge. The purpose of the GSA softball platform is to manage the softball league functionalities (communication, scheduling, player logins, etc.) so it meets the needs and demands of the league.

7.2 The Context of the Work

The GSA Softball League Platform interacts with various external entities including players, captains, commissioners, and integrated systems. The system context diagram below illustrates these interaction boundaries, data flows, and system interfaces, highlighting the scope of the platform's functionality and its relationships with users and external systems. See [Figure 1](#) for the system context diagram.

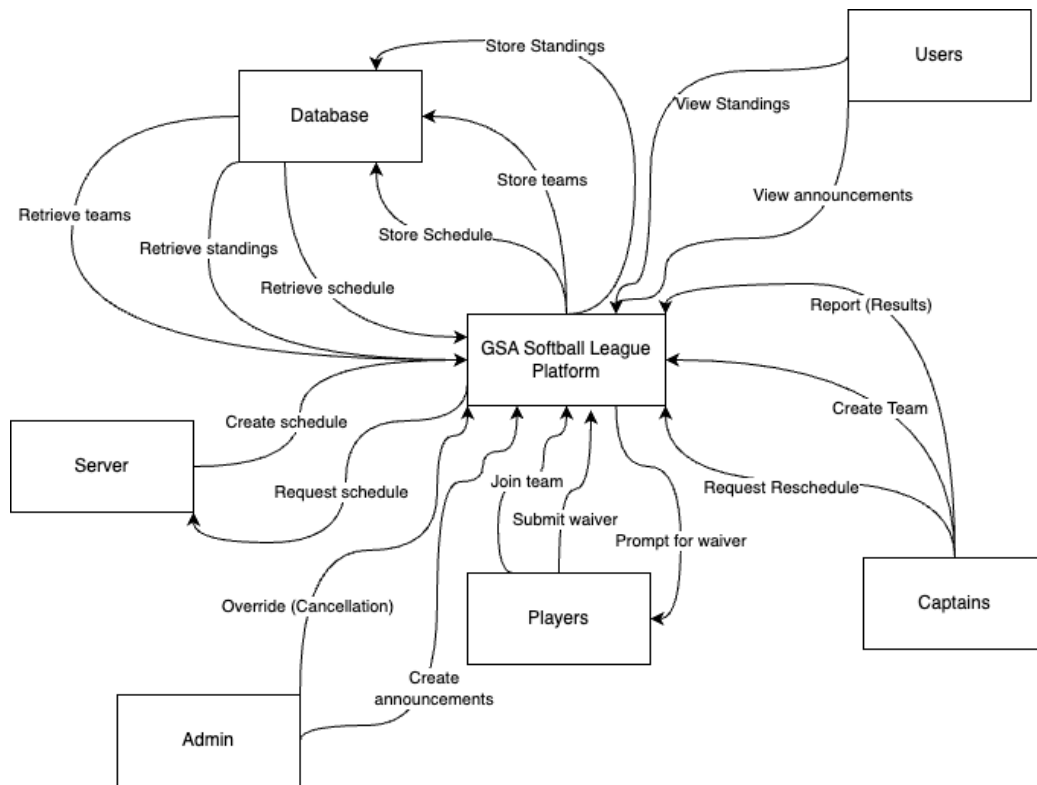


Figure 1: System Context Diagram

7.3 Work Partitioning

Event Name	Input/Output	Summary
Login	User ID (IN) Password (IN)	Enter user ID and password. User is created and user data is stored in the database.
Join team	User Login (IN) Prompt for waiver (OUT) Submit waiver (IN)	User logs in as a player, then completing the waiver allows the user to join a team
Create team	User Login (IN) Team Info (IN)	User logs into the system as a captain, then the user inputs the team info to create a team. The team is created and stored in the database.
Create Schedule	Season Schedule (OUT)	System uses all teams info to create season-long schedule with all open slots available.
Override	Admin credentials (IN)	User inputs admin credentials and gets access to complete override.
Create announcement	Admin credentials (IN) Announcement details (IN) Publish announcement (OUT)	User inputs admin credentials to gain access then enters in the announcements details. Then the announcement is published on the web page.
Request reschedule	User Login (IN)	User logs into the system as a captain, then creates a request to reschedule a game in an open slot.
Report (Results)	User Login (IN) Game results (IN) Update standings (OUT)	User logs in as a captain then inputs the results of the game. Update standings in the database with reported results.

7.4 Specifying a Business Use Case

The primary business use case for this platform is to efficiently manage all aspects of league operations from player registrations to game scheduling in a user-friendly and streamlined manner. The platform will provide commissioners, team captains, and players with a centralized system to interact with the league. The system allows commissioners to oversee the entire league, manage team and player information, and handle payments. Team captains will be able to manage their rosters, submit rescheduling requests, and report game scores, while players can join teams, view schedules, and observe league rankings and scores. For example, team captains can log in to the platform and create their teams. Players can join a team by requesting approval from the captain, while commissioners can monitor the overall progress and approve team formations. Throughout the season, captains can report game scores directly into the system, which automatically updates the standings. If a team needs to reschedule a game, the captain can request a change through the system, selecting from available timeslots. The system will queue the request for commissioner approval and update the schedule once the rescheduling is confirmed.

8 Business Data Model and Data Dictionary

8.1 Business Data Model

- **User**
 - UserID: Unique identifier for each user.
 - Name: Full name of the user.
 - Email: Email address used for communication and login.
 - Role: The role of the user in the system (commissioner, captain, player).
 - TeamID: The ID of the team the user is affiliated with (for captains and players).
- **Team**
 - TeamID: Unique identifier for each team.

- TeamName: Name of the team.
- Division: The division the team is assigned to.
- CaptainID: The UserID of the team captain.
- Roster: List of players (UserIDs) on the team.

- **Game**

- GameID: Unique identifier for each game.
- TeamAID: The ID of the first participating team.
- TeamBID: The ID of the second participating team.
- Date: The date the game is scheduled.
- Time: The time the game is scheduled.
- Result: The outcome of the game (win/loss/tie) reported by captains.
- ScoreTeamA: The score of Team A in the game.
- ScoreTeamB: The score of Team B in the game.

- **Schedule**

- ScheduleID: Unique identifier for the schedule.
- GameID: The ID of the game scheduled.
- SlotNumber: The slot number for the game (time and location).
- AvailableSlots: List of available time slots for potential rescheduling.

- **Standing**

- TeamID: The ID of the team.
- Wins: Number of games won by the team.
- Losses: Number of games lost by the team.
- Ties: Number of tied games.
- Rank: Current rank of the team in the league.

- **TotalScore:** Cumulative score across all games, used to break ties in standings.

8.2 Data Dictionary

- **Users:** There are three types of users: commissioners, team captains, and players. Commissioners oversee league operations, team captains manage their respective teams, and players join and participate in games. Each user has attributes such as name, email, role, and team affiliation.
- **Teams:** Teams consist of players and a captain. Teams are scheduled to play games, and the results of those games contribute to league standings. Teams have attributes like team name, captain, and roster.
- **Games:** Games represent the scheduled matches between two teams. Each game has attributes like date, time, and the teams involved, as well as the final score for both teams, which is reported by captains after the game.
- **Schedules:** The schedule defines the timing and location of games. It also includes available time slots for rescheduling. The system tracks preferences for game days and times, and generates the schedule accordingly.
- **Standings:** Standings reflect the rankings of teams based on the results of their games. Game scores feed into the standings, updating the number of wins, losses, and ties for each team, which determine the overall ranking.

9 Formal Specifications

9.1 State Machine Model

The system can be modeled as a state machine with the following states and transitions:

- **States:**
 - S_0 : Initial state (not logged in)

- S_1 : Logged in as Player
- S_2 : Logged in as Captain
- S_3 : Logged in as Commissioner
- S_4 : Team Creation
- S_5 : Game Scheduling
- S_6 : Score Reporting

• **Transitions:**

- $T_{0,1}$: Login as Player
- $T_{0,2}$: Login as Captain
- $T_{0,3}$: Login as Commissioner
- $T_{2,4}$: Create Team
- $T_{2,5}$: Schedule Game
- $T_{2,6}$: Report Score
- $T_{1,0}$: Logout
- $T_{2,0}$: Logout
- $T_{3,0}$: Logout

• **State Invariants:**

- $\forall s \in S : s \neq S_0 \implies \text{user is authenticated}$
- $\forall s \in \{S_4, S_5, S_6\} : s \implies \text{user has appropriate permissions}$

9.2 Formal Requirements

The system must satisfy the following formal requirements:

1. **Authentication Requirements:**

$$\forall u \in Users : \exists s \in Sessions : s.user = u \implies s.isAuthenticated = true$$

2. Team Management Requirements:

$$\forall t \in Teams : \exists c \in Captains : t.captain = c \implies c.id = t.captain.id$$

3. Scheduling Requirements:

$$\begin{aligned} & \forall g \in Games : \forall t \in Teams : g.teams \subseteq t \\ & \implies \neg \exists g' \in Games : g'.time = g.time \wedge g'.teams \cap g.teams \neq \emptyset \end{aligned}$$

4. Score Reporting Requirements:

$$\forall g \in Games : g.status = \text{"submitted"} \implies \exists s \in Scores : s.game = g \wedge s.isValid = true$$

9.3 Data Flow Specifications

The system follows these data flow rules:

- **User Data Flow:**

$$\forall u \in Users : \forall d \in UserData : d.owner = u \implies d.accessLevel \leq u.role.accessLevel$$

- **Team Data Flow:**

$$\forall t \in Teams : \forall m \in TeamMembers : m.team = t \implies m.hasAccess(t.data) = true$$

- **Game Data Flow:**

$$\forall g \in Games : \forall t \in Teams : t \in g.teams \implies t.hasAccess(g.data) = true$$

9.4 Safety Invariants

The system maintains these safety invariants:

- **Access Control:**

$$\forall a \in Actions : \forall u \in Users : u.performs(a) = true \implies u.hasPermission(a) = true$$

10 The Scope of the Product

10.1 Product Boundary

The product boundary defines the scope of interactions between the system and its external entities. The context diagram illustrates how the system interfaces with users (players, captains, and commissioners), external services (payment processing, email notifications), and data storage systems. It clearly shows the system's boundaries and the flow of information between the system and its environment. See [Figure 1](#) for the system context diagram.

10.2 Product Use Case List

The use case diagram provides a comprehensive view of the system's functionality and user interactions. It shows three primary user roles (players, captains, and commissioners) and their relationships with core system features. The diagram illustrates the hierarchical nature of permissions, with commissioners having access to all features, captains having team management capabilities, and players having basic access to view schedules and standings. See [Figure 2](#) for the use case diagram.

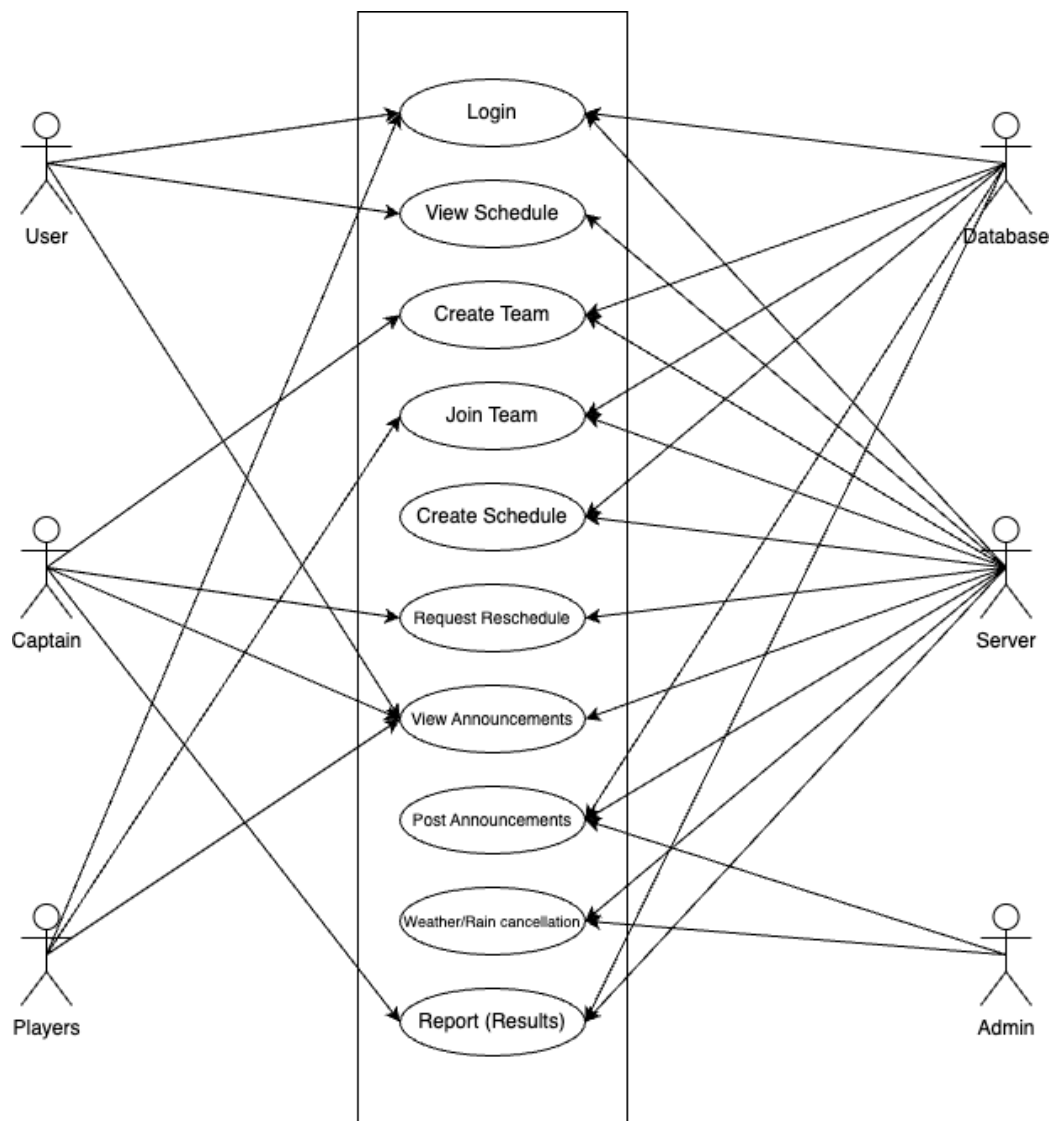


Figure 2: Use Case Diagram

10.3 Individual Product Use Cases (PUC's)

- **UC1: Login**
Trigger: User enters their username and password.
Preconditions: Client is not logged in to the system with an existing account.
Actors: User, Captain, Player, Database, Server.
Outcome: If the username and password is a match with an account in the database, the user is logged in. If not, then the user is prompted to try again, reset password (if username exists) or create an account.
- **UC2: View Schedule**
Trigger: User interacts with the view schedule prompt.
Preconditions: N/A (All users can view schedule)
Actors: User, Database, Server Outcome: Page opens up with the schedule for the GSA softball league season.
- **UC3: View Standings**
Trigger: User interacts with View standings prompt.
Preconditions: N/A (All users can view standings)
Actors: User, Database, Server.
Outcome: The standings for the GSA softball league season is displayed.
- **UC4: Create Team**
Trigger: User request to make a team with team info.
Preconditions: Client is logged in to the system as a Captain.
Actors: User, Captain, Database, Server.
Outcome: If the team name is unique and the user is not already an existing member of another team, the team is created with the user as the Captain of the team.
- **UC5: Join Team**
Trigger: User request to join an already existing team
Preconditions: User is logged into the system, the user is not part of an already existing team, and team being requested exists.
Actors: User, Player, Database, Server.
Outcome: If the user is logged in and is not a member of an already existing team, then the user will receive a request to fill out the waiver if the team exists. Upon completion, the user becomes a member of

that team.

- **UC6: Create Schedule**
Trigger: Registration deadline is past due.
Preconditions: Teams are registered in the league and meet the team conditions.
Actors: Server.
Outcome: The system uses the team preferences and creates a schedule with all open slots.
- **UC7: Request Reschedule**
Trigger: Captain interacts with the reschedule prompt at least 24 hours before the start time.
Preconditions: Team exists and has a scheduled game.
Actors: User, Captain, Database, Server.
Outcome: If the user is a Captain and makes a request on an existing game at least 24 hours before the start, the Captain is able to view available slots for the game. Upon choosing an empty slot, the game schedule is updated.
- **UC8: View Announcements**
Trigger: User interacts with view announcement prompt.
Preconditions: N/A (All users can view announcements).
Actors: User, Captain, Player, Database, Server.
Outcome: The announcements for the GSA softball league season is displayed.
- **UC9: Post Announcements**
Trigger: User interacts with the post announcement prompt.
Preconditions: User must be logged in as an Admin.
Actors: User, Admin, Database, Server.
Outcome: If the user is logged in as an admin, and enters all the necessary information to complete the required sections for the announcements, the announcements is created and published on the GSA softball league platform.
- **UC10: Rain Cancellation/Override**
Trigger: User interacts with the Override prompt.
Preconditions: User must be logged in as an Admin.
Actors: User, Admin, Database, Server.

Outcome: If the user is logged in as an admin, the user is able to cancel games or move then to open slots. Upon completion, the schedule is updated.

- **UC11:** Report Results

Trigger: User interacts with the report prompt.

Preconditions: The user must be logged in as a Captain and the duration of a game must be elapsed after the scheduled start time.

Actors: User, Captain, Database, Server.

Outcome: If the user is logged in as a captain, and the duration of a game has been elapsed since the start time, then the Captain is able to update the status of the game with results. Upon completion the standings and game results are updated on the system.

11 Functional Requirements

This section details the core functional requirements of the system, organized by feature area. Each requirement includes a unique identifier, description, rationale, and measurable fit criteria. Requirements are traced to their corresponding use cases and dependencies.

Requirement ID: FR1 **Type:** Functional **Use Case:** UC1

Priority: High

Description: The platform shall provide a secure login system where users enter credentials (email, password) to access the system. The system must:

- Validate user credentials against encrypted database records
- Lock accounts after **MAX_LOGIN_ATTEMPTS** failed attempts
- Enforce password complexity requirements

Rationale: Ensures secure access to the platform and role-specific functionality.

Fit Criterion:

- Authentication responses within **MAX_RESPONSE_TIME** second
- Zero unauthorized access incidents
- 100% successful login rate for valid credentials

Customer Satisfaction: 5 **Customer Dissatisfaction:** 5

Dependencies: Database system

Conflicts: None

Requirement ID: FR2 **Type:** Functional **Use Case:** UC6, UC7, UC9, UC10, UC11

Priority: High

Description: The platform shall implement role-based access control (RBAC) with distinct permissions for commissioners, captains, and players. The system must:

- Enforce role-specific access restrictions
- Maintain audit logs of permission changes
- Support role assignment and modification

Rationale: Ensures appropriate access levels and maintains system security.

Fit Criterion:

- 100% accurate role-based access enforcement

Customer Satisfaction: 5 **Customer Dissatisfaction:** 5

Dependencies: FR1 (Authentication System)

Conflicts: None

Requirement ID: FR3 **Type:** Functional **Use Case:** UC2

Description: The system shall provide three distinct schedule viewing options:

- Weekly Team Schedule View:
 - Display current week's games for user's team
 - Show date, time, location, and opponent for each game
- Monthly Team Schedule View:
 - Calendar view of the month showing all of the user's team games
 - Show date, time, location, and opponent for each game
 - Allow users to navigate between different months

- Monthly League Schedule View:
 - Complete calendar of all league games
 - Show date, time, location, and home/away teams for each game
 - Allow users to navigate between different months

Rationale: Multiple view options provide flexibility for different user needs and planning horizons while ensuring comprehensive schedule access.

Fit Criterion:

- All three views accurately display game information
- Users can seamlessly switch between views
- Calendar integration works with major calendar systems
- Views update within 5 seconds of schedule changes
- 100% of scheduled games appear in appropriate views

Customer Satisfaction: 5 **Customer Dissatisfaction:** 5

Dependencies: Game scheduling system, user authentication

Conflicts: None

Requirement ID: FR4 **Type:** Functional **Use Case:** UC3

Description: The platform shall provide an accurate display of league standings for all registered users.

Rationale: Keeps league members informed of team rankings.

Fit Criterion: League standings are updated accurately, automatically and visible to logged-in users.

Customer Satisfaction: 4 **Customer Dissatisfaction:** 4

Requirement ID: FR5 **Type:** Functional **Use Case:** UC4

Priority: High

Description: The platform shall allow captains to create a team by entering team details and submitting for registration.

Rationale: Enables team formation and league participation.

Fit Criterion: Captains can create a team, and details are stored in the system.

Customer Satisfaction: 5 **Customer Dissatisfaction:** 5

Requirement ID: FR6 **Type:** Functional **Use Case:** UC5

Priority: High

Description: The system shall provide team captains with the exclusive ability to invite players to their team. The invitation process must:

- Allow captains to send invitations to players via email
- Allow invited players to accept or decline the invitation
- Automatically update team rosters upon invitation acceptance

Rationale: Centralizing team formation control with captains ensures organized roster management and clear accountability for team composition.

Fit Criterion:

- Only team captains can initiate player invitations
- Players cannot join teams without a captain's invitation
- System prevents duplicate invitations to the same player
- Players already on a team cannot be invited to another team
- 100% of roster updates are accurately reflected after invitation acceptance

Customer Satisfaction: 4 **Customer Dissatisfaction:** 5

Dependencies: User authentication system, email notification system

Conflicts: None

Requirement ID: FR7 **Type:** Functional **Use Case:** UC6

Priority: High

Description: The platform shall automatically generate the season game schedule based on team availability and preferences.

Rationale: Automates scheduling for league to begin with improved efficiency.

Fit Criterion: The system generates a schedule that gives all teams equal games and factors in team preferences.

Customer Satisfaction: 5 **Customer Dissatisfaction:** 5

Requirement ID: FR8 **Type:** Functional **Use Case:** UC7
Priority: High
Description: The platform shall allow captains to request a reschedule by selecting a new time slots for an existing game.
Rationale: Enables flexibility and error-handling in scheduling.
Fit Criterion: Captains can successfully submit reschedule requests, and the system tracks pending requests.
Customer Satisfaction: 4 **Customer Dissatisfaction:** 5

Requirement ID: FR9 **Type:** Functional **Use Case:** UC8
Priority: Medium
Description: The platform shall allow users to view league-wide announcements, lost and found, weather updates on the dashboard.
Rationale: Keeps users informed about important league updates.
Fit Criterion: Announcements are displayed in the users announcement dashboard.
Customer Satisfaction: 4 **Customer Dissatisfaction:** 3

Requirement ID: FR10 **Type:** Functional **Use Case:** UC9
Priority: Medium
Description: The platform shall allow administrators to post announcements that are visible to all users.
Rationale: Enables effective communication with all participants.
Fit Criterion: Administrators can post announcements to be viewable by players.
Customer Satisfaction: 3 **Customer Dissatisfaction:** 3

Requirement ID: FR11 **Type:** Functional **Use Case:** UC10
Priority: High
Description: The platform shall allow administrators to manually cancel or reschedule games due to weather conditions or accepted requests.
Rationale: Ensures timely communication for game rescheduling or cancellations.
Fit Criterion: Administrators can cancel or reschedule games, which updates the schedule, and teams are notified automatically.
Customer Satisfaction: 5 **Customer Dissatisfaction:** 4

Requirement ID: FR12 **Type:** Functional **Use Case:** UC11

Priority: High

Description: The platform shall allow captains to report game scores and forfeits by selecting a completed game and submitting the final result.

Rationale: Simplifies reporting and keeps standings up to date.

Fit Criterion: Captains can submit game results, and standings are updated automatically.

Customer Satisfaction: 4 **Customer Dissatisfaction:** 3

Requirement ID: FR13 **Type:** Functional **Use Case:** UC8

Priority: High

Description: The system shall provide a player profile management interface that allows users to:

- View and update personal information:
 - First name and last name
 - Email address
 - Phone number

Rationale: Centralized profile management ensures accurate player information for league communications and emergency situations.

Fit Criterion:

- All profile updates are saved and reflected immediately
- Email addresses are validated for correct format
- Phone numbers are validated for correct format
- Changes are logged for audit purposes

Customer Satisfaction: 4 **Customer Dissatisfaction:** 4

Dependencies: User authentication system, data encryption system

Conflicts: None

12 Non-Functional Requirements

The following requirements labeled with [HA] are derived from the Hazard Analysis document.

12.1 Data Integrity and Backup Requirements

Requirement ID: NFR7 [HA4] **Type:** Non-Functional

Priority: Must Have

Description: The platform shall implement comprehensive data integrity and backup mechanisms:

- Schedule and perform regular data backups and integrity checks
- Backup all critical data (game results, standings, user profiles) daily during season
- Store backups in geographically separate location for security
- Enable data restoration from latest backup
- Test backup and recovery processes each season

Rationale: Guarantee that the platform can recover from any potential data loss or corruption.

Fit Criterion:

- 100% successful backup verification and restore testing
- Data backups must occur at least once daily during the season
- Backup and recovery processes must be tested every season

Dependencies: Database system

Conflicts: None

12.2 Session Management Requirements

Requirement ID: NFR8 [HA5] **Type:** Non-Functional

Priority: Must Have

Description: The platform shall implement robust session management:

- Automatic session timeouts after 10 minutes of inactivity
- Display countdown warning 1 minute before timeout
- Require re-authentication after timeout

Rationale: Prevent unauthorized access on shared devices and enhance security.

Fit Criterion:

- All inactive sessions must terminate within specified timeframe
- 100% of timeouts must trigger re-authentication

Dependencies: Authentication system

Conflicts: None

12.3 Audit Logging Requirements

Requirement ID: NFR9 [HA6] **Type:** Non-Functional

Priority: Should Have

Description: The platform shall implement comprehensive audit logging:

- Log all critical actions (logins, schedule changes, roster updates)
- Store logs securely with encryption
- Retain logs for minimum 90 days
- Enable admin search and filtering of logs
- Generate real-time alerts for suspicious activities

Rationale: Enable troubleshooting and security monitoring.

Fit Criterion:

- 100% logging coverage of critical actions
- Logs must be retained for minimum 90 days
- Real-time alert generation for suspicious activities

Dependencies: Storage system, encryption system

Conflicts: None

12.4 Security Testing Requirements

Requirement ID: NFR10 [HA7] **Type:** Non-Functional

Priority: Must Have

Description: The platform shall undergo comprehensive security testing:

- Perform end-to-end security testing
- Automate security tests for consistent application
- Test for common vulnerabilities (SQL injection, XSS)

Rationale: Ensure platform security against potential threats.

Fit Criterion:

- All security tests must pass before deployment
- Zero known security vulnerabilities in production

Dependencies: Testing framework

Conflicts: None

12.5 Schedule Verification Requirements

Requirement ID: NFR11 [HA8] **Type:** Non-Functional

Priority: Must Have

Description: The platform shall implement automated schedule verification:

- Verify schedule completeness automatically
- Check for missing games or conflicts
- Validate fair distribution of games

Rationale: Prevent scheduling errors and ensure fairness in game distribution.

Fit Criterion:

- Zero undetected scheduling conflicts
- Zero missing games in generated schedules
- Equal game distribution among teams

Dependencies: Scheduling system
Conflicts: None

13 Look and Feel Requirements

This section details the non-functional requirements for the look and feel of the platform.

13.1 Layout Requirements

Requirement ID: NFR-LF3 **Type:** Non-Functional

Description: The platform shall maintain a responsive layout that adapts to different screen sizes and devices, ensuring optimal viewing and interaction across desktop and mobile platforms.

Fit Criterion: The interface must maintain usability and readability across all common device sizes (mobile, tablet, desktop) with no horizontal scrolling required.

13.2 Visual Feedback Requirements

Requirement ID: NFR-LF4 **Type:** Non-Functional

Description: The platform shall provide clear visual feedback for all user actions, including loading states, success/error messages, and interactive element states.

Fit Criterion: All user actions must receive immediate visual feedback, and system status changes must be clearly indicated to users.

13.3 Appearance Requirements

Requirement ID: NFR-LF1 **Type:** Non-Functional

Description: The platform must feature a modern, intuitive interface with consistent visual elements across all views to ensure ease of use.

Fit Criterion: [USER_INTERFACE_RATING](#)% of users should rate the interface as "intuitive" or "very intuitive" in usability surveys.

13.4 Style Requirements

Requirement ID: NFR-LF2 **Type:** Non-Functional

Description: The platform shall follow a consistent color palette, typography, and layout to ensure ease of use, and improve accessibility by maintaining predictable design elements across the platform.

Fit Criterion: All pages must use the defined style guide elements with 100% compliance.

14 Usability and Humanity Requirements

14.1 Ease of Use Requirements

Requirement ID: NFR-U1 **Type:** Non-Functional

Description: Graduate students and other users shall be able to complete common tasks (login, team creation, schedule viewing) within an average of 3 minutes with no prior experience or external help.

Fit Criterion: [TASK_COMPLETION_RATE](#)% of new users should complete basic tasks within the specified time frame.

14.2 Personalization and Internationalization Requirements

Requirement ID: NFR-U2 **Type:** Non-Functional

Description: The platform must be tailored to Canadian English, using the metric system for measurements and adhering to local date and time formats.

Fit Criterion: 100% compliance with Canadian date/time formats and measurement standards.

14.3 Learning Requirements

Requirement ID: NFR-U3 **Type:** Non-Functional

Description: The platform must provide clear instructions and tooltips on all views to help new users quickly learn how to navigate and use key features.

Fit Criterion: [LEARNING_SUCCESS_RATE](#)% of new users should be able to complete basic tasks without requiring support.

14.4 Understandability and Politeness Requirements

Requirement ID: NFR-U4 **Type:** Non-Functional

Description: The platform must provide clear, straightforward navigation with simply labeled menus and buttons so that minimal prior knowledge is required.

Fit Criterion: All navigation elements must have clear, descriptive labels with no technical jargon.

14.5 Accessibility Requirements

Requirement ID: NFR-U5 **Type:** Non-Functional

Description: The platform shall provide basic accessibility features, such as keyboard navigation and sufficient text contrast, to ensure usability for a wide range of users.

Fit Criterion: Meet WCAG 2.1 Level AA compliance requirements.

15 Performance Requirements

15.1 Speed and Latency Requirements

Requirement ID: NFR-P1 **Type:** Non-Functional

Description: The platform must respond to user actions (e.g., scheduling, score reporting) within [MAX_RESPONSE_TIME](#) second to contribute to user satisfaction and efficiency.

Fit Criterion: [ACTION_COMPLETION_RATE](#)% of all user actions must complete within the specified time limit.

15.2 Safety-Critical Requirements

Requirement ID: NFR-P2 **Type:** Non-Functional

Description: The platform must ensure the secure and accurate storage of player personal information and waivers, preventing data loss or corruption.

Fit Criterion: Zero instances of data loss or unauthorized access.

15.3 Precision or Accuracy Requirements

Requirement ID: NFR-P3 **Type:** Non-Functional

Description: The platform must calculate and display the standings with 100% accuracy and accurately match game preferences with available slots to avoid conflicts.

Fit Criterion: 100% accuracy in standings calculations and zero scheduling conflicts.

15.4 Robustness or Fault-Tolerance Requirements

Requirement ID: NFR-P4 **Type:** Non-Functional

Description: The platform shall handle failed database connections by logging the error and displaying an error message to the user, and allow users to retry their actions.

Fit Criterion: All system errors must be logged and users must be able to retry failed actions.

15.5 Capacity Requirements

Requirement ID: NFR-P5 **Type:** Non-Functional

Description: The platform must be able to store and handle the scheduling for at minimum `MAX_TEAM_CAPACITY` teams in the league to ensure scheduling handling and league expansion.

Fit Criterion: Successfully manage `MAX_TEAM_CAPACITY` teams with no degradation in performance.

15.6 Scalability or Extensibility Requirements

Requirement ID: NFR-P6 **Type:** Non-Functional

Description: The platform shall support handling up to `MAX_TEAM_CAPACITY` teams without performance issues to ensure scheduling handling and website performance for a full league.

Fit Criterion: System performance must remain consistent when managing `MAX_TEAM_CAPACITY` teams.

15.7 Longevity Requirements

Requirement ID: NFR-P7 **Type:** Non-Functional

Description: The platform must use modern, widely supported web technologies to ensure long-term compatibility with future devices and browsers.

Fit Criterion: Compatible with all major browsers and devices for at least [BROWSER_COMPATIBILITY_YEARS](#) years.

16 Operational and Environmental Requirements

16.1 Expected Physical Environment

Requirement ID: NFR-O1 **Type:** Non-Functional

Description: The platform must be accessible on various devices, such as desktops, laptops, smartphones, and tablets, with a stable internet connection and appropriate server infrastructure.

Fit Criterion: 100% functionality across all specified device types.

16.2 Browser Compatibility

Requirement ID: NFR-O2 **Type:** Non-Functional

Description: The platform must be accessible across multiple browsers with responsive design that allows users to reach all features when using a mobile device.

Fit Criterion: Full functionality in Chrome, Safari, Firefox, and Edge browsers.

16.3 Requirements for Interfacing with Adjacent Systems

Requirement ID: NFR-O3 **Type:** Non-Functional

Description: The platform must support integration with payment gateways, calendar systems, and data import/export functionality for roster data and match results.

Fit Criterion: Successful integration with specified external systems with [SYSTEM_UPTIME](#)% uptime.

16.4 Productization Requirements

Requirement ID: NFR-O4 **Type:** Non-Functional

Description: The platform must handle scaling well, such as an increased number of divisions, teams, players, etc. League managers should be able to manage game rules and related information.

Fit Criterion: System maintains performance with up to [MAX_TEAMS](#) teams and supports rule management within [ADMIN_CHANGE_TIME](#) minutes.

16.5 Release Requirements

Requirement ID: NFR-O5 **Type:** Non-Functional

Description: Version control is required for tracking changes and rollback capabilities. All production releases must undergo thorough testing with real users.

Fit Criterion: 100% of changes tracked in version control and all releases tested with minimum [MIN_TEST_USERS](#) real users.

17 Maintainability and Support Requirements

17.1 Maintenance Requirements

Requirement ID: NFR-M1 **Type:** Non-Functional

Description: The platform should be updated regularly for fixing bugs, adding new features, and security patches.

Fit Criterion: Monthly maintenance updates with zero critical bugs pending for more than [MAINTENANCE_BUG_DAYS](#) days.

17.2 Supportability Requirements

Requirement ID: NFR-M2 **Type:** Non-Functional

Description: Any user who requires support should be directed to an email that handles issues with the platform.

Fit Criterion: Support requests acknowledged within [SUPPORT_RESPONSE_HOURS](#) hours and resolved within [SUPPORT_RESOLUTION_HOURS](#) hours.

17.3 Adaptability Requirements

Requirement ID: NFR-M3 **Type:** Non-Functional

Description: An admin user should be able to change game rules within [ADMIN_CHANGE_TIME](#) minutes, such as swapping a team's game schedule with another if needed.

Fit Criterion: [ADMIN_CHANGE_SUCCESS](#)% of administrative changes completed within specified time frame.

18 Security Requirements

18.1 Access Requirements

Requirement ID: NFR-S1 **Type:** Non-Functional

Description: The platform must implement role-based access control with distinct permissions for administrators, team managers, and players.

Fit Criterion: Zero unauthorized access incidents and 100% correct permission enforcement.

18.2 Integrity Requirements

Requirement ID: NFR-S3 **Type:** Non-Functional

Description: All data entered such as team rosters, game schedules, and scores must be validated for accuracy and completeness.

Fit Criterion: 100% validation of all data entries with error logging.

18.3 Privacy Requirements

Personal user data such as emails, phone numbers, etc., should be protected by encryption.

18.4 Audit Requirements

Requirement ID: NFR-S4 **Type:** Non-Functional

Description: Logs of team history, game scores, roster updates, and game scheduling, should all be kept in case we ever need them to solve disputes.

Fit Criterion: Complete audit trail for all system changes maintained for 1 year.

18.5 Immunity Requirements

Requirement ID: NFR-S5 **Type:** Non-Functional

Description: The platform should be protected against malicious web attacks.

Fit Criterion: Zero successful security breaches and 100% compliance with OWASP security standards.

19 Cultural Requirements

19.1 Cultural Requirements

Requirement ID: NFR-C1 **Type:** Non-Functional

Description: The platform should support multiple languages so that the diverse league participants can switch to their preferred language. The platform should only display the local time of Hamilton, as it is assumed that all league participants are in the same region where the games are being hosted.

Fit Criterion: Support for English and French languages with 100% accurate time zone display.

20 Compliance Requirements

20.1 Legal Requirements

Requirement ID: NFR-L1 **Type:** Non-Functional

Description: Users who no longer want to be part of the league should be able to delete their data. The platform must accommodate for this within 2 minutes.

Fit Criterion: 100% of data deletion requests completed within specified timeframe.

20.2 Standards Compliance Requirements

Requirement ID: NFR-ST1 **Type:** Non-Functional

Description: The platform must adhere to the following standards and regulations:

- **Privacy Standards:**

- Personal Information Protection and Electronic Documents Act (PIPEDA) [1]
- Freedom of Information and Protection of Privacy Act (FIPPA) [2]
- General Data Protection Regulation (GDPR) compliance for international users [3]
- **Web Accessibility Standards:**
 - Web Content Accessibility Guidelines (WCAG) 2.1 Level AA compliance [4]
 - Section 508 of the Rehabilitation Act [5]
- **Sports League Standards:**
 - Canadian Interuniversity Sport (CIS) guidelines for recreational sports [6]
 - McMaster University Athletics and Recreation policies [7]
- **Technical Standards:**
 - OWASP Top 10 Security Standards [8]
 - ISO/IEC 27001 Information Security Management [9]
 - REST API Design Standards [10]

Fit Criterion: 100% compliance with all specified standards and regulations.

20.3 Industry Best Practices

Requirement ID: NFR-ST2 **Type:** Non-Functional

Description: The platform will implement the following industry best practices:

- **Data Management:**
 - Regular data backups every `DATA_BACKUP_FREQUENCY_DAYS` days
 - Data encryption at rest and in transit

- Secure password hashing using industry-standard algorithms
- **User Experience:**
 - Responsive design principles
 - Mobile-first approach
 - Consistent navigation patterns
 - Page load time under [PAGE_LOAD_TIME_SECONDS](#) seconds
 - Database query response time under [DATABASE_QUERY_TIME_MS](#) milliseconds
- **Performance:**
 - Database query optimization with response time under [DATABASE_QUERY_TIME_MS](#)ms
 - Regular performance monitoring and optimization

Fit Criterion: All specified best practices must be implemented and verified through testing.

21 Open Issues

This section details the open issues that are still under discussion.

- **User Interface Design:** Some aspects of the user interface are yet to be fully defined. Specific visual elements and interaction flows will be determined in collaboration with stakeholders during the design phase.
- **User Interface Testing:** Details on the testing methodology for the user interface are still under discussion. The testing plan will include usability testing with users from the target audience to ensure ease of use.
- **User Payment Tracking Feature:** The implementation specifics for the payment tracking feature for staff are still being finalized. Considerations are being made regarding how the status of each player's payment will be updated and stored.
- **Integration of Existing Features:** There are open questions regarding the best approach to integrate and improve existing features from

the current website in the new platform.

22 Off-the-Shelf Solutions

This section details the off-the-shelf solutions that will be used in the project.

22.1 Ready-Made Products

There are no ready-made products that will be a part of our solution. The objective is to update and completely redesign the current softball league web-based platform. While features from the original platform will be transferred over to maintain continuity, the entire solution will be developed in-house based on updated requirements and best practices.

22.2 Reusable Components

While no third-party reusable components will be integrated, several features from the original platform, such as user authentication, team management, and scheduling, will be carried over to ensure continuity. These components will be updated and enhanced as part of the complete redesign to meet the new requirements.

22.3 Products That Can Be Copied

There are no external products that will be directly copied for this project. However, the redesign will reference the design and functional elements of the current website to maintain familiarity for existing users. Additionally, best practices from similar platforms will be researched and adapted to improve the user experience.

23 New Problems

This section details the new problems that may arise during the project.

23.1 Effects on the Current Environment

Implementing the new system will require a complete overhaul of the existing McMaster GSA softball league platform, which may cause temporary disruption to ongoing league activities. The outdated system will be replaced, and users accustomed to the old methods may face a learning curve during the transition. Additionally, migrating any existing data (such as player information, schedules, and standings) to the new platform will need to be handled carefully to avoid loss or corruption. However, the improved system will provide long-term benefits by reducing maintenance and manual input required by the commissioners and administrators.

23.2 Effects on the Installed Systems

Since the current system is outdated integrating the new platform may require adjustments to the existing server infrastructure or cloud services. The new system will be built on modern technologies which may require updating the server environment to ensure compatibility with these technologies. If hosted on legacy systems, a migration to a cloud or updated hosting service might be required.

23.3 Potential User Problems

Users (commissioners, captains, players) who are familiar with the current platform may struggle with adapting to the new interface and features. Particularly, older users or users with less technical proficiency may have difficulty navigating the system or understanding new functionalities like the rescheduling workflow, leaderboard tracking, or player management. Proper documentation and user training will be necessary to ensure a smooth transition and reduce user frustration. Moreover, captains and commissioners may need additional guidance on using automated features like team creation, player approvals, and schedule rescheduling.

23.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

The project must be compatible with various devices (desktop, tablet, mobile), which presents challenges in ensuring consistent performance and user experience across different screen sizes and browsers. Another potential limitation is the anticipated usage spikes, such as the heavy volume of rescheduling requests at the beginning of the season, which could strain the system. The hosting environment needs to handle potential scalability issues, and the system should be optimized for responsiveness during high-demand periods. There may also be limitations in integrating certain external services, such as payment verification systems, if APIs or infrastructure constraints arise.

23.5 Follow-Up Problems

After the initial launch of the new platform, follow-up issues may include bug fixes, additional feature requests, and ongoing user feedback that necessitates adjustments. Since the system will rely on certain automated processes (e.g., scheduling, score reporting), there may be edge cases or scenarios not accounted for during development, which would require post-launch refinements. Maintenance and updates to the platform will need to be addressed continuously to ensure smooth functionality, especially in response to league changes or growing participation numbers. Moreover, as the system expands, additional support or enhancements might be required to maintain security and efficiency.

24 Tasks

This section details the tasks that will be undertaken during the project.

24.1 Project Planning

The platform's development will follow a structured plan prioritizing the requirements as outlined in the SRS. The functional requirements have been grouped based on priority, with each group indicating the relevant require-

ment numbers. Non-functional requirements, such as look and feel, usability, and accessibility, will be applied throughout the project to ensure consistency across all phases.

High Priority – These features are critical for the functioning of the platform:

- **User Authentication and Account Management** (Requirements FR1, FR2): Secure login and user roles (commissioner, captain, player).
- **Team Management** (Requirements FR5, FR6): Captains create teams, select divisions, and manage rosters. Players can request to join teams.
- **Schedule Generation** (Requirement FR7): Automatically generate schedules, ensuring even distribution and honoring team preferences for dates/times.
- **Score Reporting** (Requirements FR11, FR12): Captains report game scores and forfeits, updating standings automatically.
- **Rescheduling System** (Requirements FR8, FR11): Allow captains to reschedule games, with requests approved by commissioners. Handle conflict checks and queues for multiple requests.

Medium Priority – These features enhance the user experience but are not critical to the platform’s core functionality:

- **Player Login and Team Requests** (Requirement FR6): Players can log in, view team information, and request to join teams.
- **Rescheduling Queue Management** (Requirement FR8): Manage multiple rescheduling requests and enforce time restrictions (e.g., within 24 hours of a game).
- **Announcements** (Requirements FR9, FR10): Commissioners post league-wide announcements for all users.

Low Priority – Features that are “nice to have” but not necessary for platform operation:

- **UI/UX Enhancements** (Non-functional requirement): A modern interface, cleaner design, weekly schedule tabs, and alternating colors for easier reading.

- **Lost and Found Feature** (Non-functional requirement): Allow users to report and check lost items.
- **Waiver System** (Non-functional requirement): Integrate waivers with payment status tracking.

24.2 Planning of the Development Phases

Development will be divided into phases, with each phase focused on implementing the functional requirements in order of priority. Each phase includes start and end dates based on your project schedule:

24.2.1 Phase 1: Requirement Gathering and Initial Planning

Start: September 10, 2024 — End: October 9, 2024

- Meet with stakeholders to determine the functional and non-functional requirements of the platform.
- Prioritize requirements based on their criticality to the project.
- Document the initial project plan, setting up deadlines for each phase.

24.2.2 Phase 2: Proof of Concept and Design

Start: October 10, 2024 — End: November 22, 2024

- Develop and demonstrate a proof of concept for high-priority features, including user authentication, team management, and scheduling (Requirements #1, #2, #5, #7).
- Finalize the design document and refine it based on stakeholder feedback.

24.2.3 Phase 3: Core Development

Start: November 23, 2024 — End: January 15, 2025

- Implement high-priority features, including team management, scheduling, and score reporting (Requirements #5, #7, #11, #12).
- Begin the integration of features from the existing platform where applicable.

24.2.4 Phase 4: Verification, Validation, and Revision

Start: January 16, 2025 — End: March 7, 2025

- Test the platform against the documented requirements.
- Present the initial version (Revision 0) and gather feedback for improvement.

24.2.5 Phase 5: Final Development and Demonstration

Start: March 8, 2025 — End: March 30, 2025

- Finalize medium- and low-priority features, such as announcements and UI/UX improvements (Requirements #9, #10, non-functional requirements).
- Prepare for the final demonstration and present the platform.

24.2.6 Phase 6: Documentation and Wrap-Up

Start: April 1, 2025 — End: April 2, 2025

- Complete all project documentation.
- Submit the final deliverables, including the user guide and source code.

25 Migration to the New Product

This section details the requirements for migrating to the new product.

25.1 Requirements for Migration to the New Product

To migrate from the existing platform to our redesign involves several key steps to ensure that the transfer of all relevant data, features, and functionalities is seamless and complete:

- **Data Preservation:** All current data from the existing platform must be preserved and migrated to the new platform. The specific data to be transferred will be confirmed through consultation with the current

platform administrator, ensuring that only the necessary and relevant data is included in the migration.

- **Feature Continuity:** Ensure that all features present in the original platform are retained in the redesigned version, with enhancements and improvements where applicable.
- **Minimizing Downtime:** The softball season begins in April, which falls around when the project is expected to be completed. Due to this tight deadline, the migration process should be planned to minimize downtime for users, ensuring that the transition is as smooth as possible with minimal impact on league operations.

25.2 Data That Has to be Modified or Translated for the New System

During the migration process, specific data elements may need to be modified or translated to align with the new platform's structure and data requirements:

- **User Accounts and Roles:** The login system needs a complete redesign. User account information, including usernames, passwords, and roles (commissioner, captain, player), will need to be reformatted to match the requirements of the new authentication system.
- **Team and Match Data:** Team rosters, match schedules, and historical results may require adjustments to fit the new database schema. This may involve reformatting data fields or translating old data structures to new formats.

26 Costs

The capstone supervisor expressed minimizing costs. Our approach will be done with free frameworks, such as React.js, and affordable hosting plans such as AWS or Google Cloud. Our team will emphasize core functionality, and avoid creating unnecessary features and costs that could undermine the product goal. By following these cost-conscious strategies, the project can stay within budget while delivering a functional and scalable platform.

27 User Documentation and Training

This section details the user documentation and training requirements.

27.1 User Documentation Requirements

- **Comprehensive User Guide:** Create a detailed user manual page tab that outlines:
 - **Getting Started:** Instructions on how to register, create teams, and navigate the platform before delving deep into more advanced features.
 - **Feature Descriptions:** Written and visual explanations of each feature, such as team management, scheduling, score tracking, etc.
 - **FAQs:** Address common questions and troubleshooting tips to help users resolve issues independently. If unresolved, prompt users to email for more support or direct them to the discussions page.
- **Interactive Tutorials:** Create pop-up icons that demonstrate key functionalities, making it easier for both new and existing users to understand how to use the platform effectively.

27.2 Training Requirements

- **Help Support Desk:** A dedicated support email or live chat feature where users can ask questions or report issues with someone real.
- **Feedback Section:** Implement a section where users can provide feedback through open-ended surveys to help improve the platform.
- **Discussion:** A section where users can publicly post their questions and other users are free to answer them.

28 Waiting Room

For the softball league platform, the "waiting room" will most likely be the feature for players who have registered, but do not have a pre-formed team.

These players are called "free agents," who may be added to teams later once all team rosters are confirmed. This concept is similar to how IMLeague is handled, which is the McMaster Intramural League platform. For example, a student desires to play softball and registers for the platform. As pre-formed teams are registered and confirmed, this student will be assigned to one of the teams that have extra space on their roster. Hence, they are "waiting" for a team to join.

29 Ideas for Solution

Players who sign up without a pre-formed team will be placed in a waiting room, designating them as "free agents." This waiting room is a holding area until all team rosters are confirmed. Other teams can choose from this holding area if they need any players before the final roster confirmation date. However, all players will eventually be auto-assigned to a team if the waiting room is still non-empty. Players from the waiting room will be notified of status updates. Players who register late will also be automatically assigned from the waiting room.

Therefore, we need a database to hold the status of each player in the waiting room. On the front user interface, the waiting room should be displayed to all users, so teams can see available free agents. For users with team manager permissions, they will be able to invite free agents from the waiting room to their teams if desired.

Appendix — Reflection

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

- 1. What went well while writing this deliverable?**

During the development of the SRS, the team was able to collaborate effectively, maintaining open communication and sharing ideas. The clear breakdown of tasks among team members helped streamline the process, ensuring that each section was tackled methodically. We used Google Docs to put down initial drafts and brainstorming so we could work together more easily in real time than if we were to draft using git commits. Utilizing LaTeX as the documentation format facilitated a consistent and professional layout, making it easy to organize and adjust content as needed.

- 2. What pain points did you experience during this deliverable, and how did you resolve them?**

One of the major challenges faced was ensuring that all requirements were accurately captured and properly categorized, especially distinguishing between functional and non-functional requirements. To resolve this, team members would bring up to the rest of the team any concerns they had in their designated sections. Another key step was comparing the requirements document with feedback from the TA. Additionally, managing the document formatting in LaTeX was initially

time-consuming, but the team grew more proficient over time through consistent practice and peer assistance. Another pain point would be the struggle to keep track of all graded content and requirements. This was because the resources we have distributed over GitLab, GitHub, and AvenueToLearn. This caused a lot of confusion for the team that we still need to get adapt to.

3. **How many of your requirements were inspired by speaking to your client(s) or their proxies (e.g., your peers, stakeholders, potential users)?**

A significant portion of the requirements—approximately 80%—were directly inspired by interactions with the client (our project advisor). Discussion with the project advisor provided valuable insights, helping the team refine specific requirements related to user roles, scheduling, and data management.

4. **Which of the courses you have taken, or are currently taking, will help your team to be successful with your capstone project?**

The following courses are expected to be instrumental in the success of this project:

- **SFWRENG 4HC3 (Human-Computer Interfaces):** This course provides essential knowledge for designing user-friendly interfaces, critical for the web platform.
- **SFWRENG 4AA4 (Real-Time Systems and Control Applications):** Offers insights into designing responsive systems, which is important for building the scheduling and management components of the platform.
- **SFWRENG 3DB3 (Databases):** Provides the necessary skills for designing and managing databases, crucial for the project's data management requirements.
- **SFWRENG 2C03 (Data Structures and Algorithms):** A foundational course that covers algorithms relevant to the scheduling functionality of the platform.
- **SFWRENG 3S03 (Software Testing & Quality Assurance):**

This course will aid in implementing rigorous testing methodologies to ensure that the platform meets the required quality standards.

5. **What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project?**

The team will need to acquire skills in several areas to ensure the successful development of the softball league platform. These include:

- **Front-End Development and User Interface Design:** Knowledge of creating user-friendly and accessible interfaces, applying Human-Computer Interfaces (HCI) principles.
- **Database Management:** Skills related to structuring, optimizing, and securing databases to handle large volumes of user data.
- **Scheduling Algorithms:** Familiarity with creating effective scheduling systems that allow team preferences and rescheduling requests to be efficiently managed.
- **Authentication and Security:** Understanding of secure login systems and data protection measures to ensure user information is kept safe.
- **System Architecture and Integration:** Skills in connecting front-end and back-end components for seamless user experience.

6. **For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?**

The approaches and team member focus areas are as follows:

- **Derek:**
 - Derek will focus on enhancing front-end development and user interface design skills by applying principles from the Human-Computer Interfaces course.
 - He will also review online tutorials on web accessibility to ensure the platform meets diverse user needs.

- Derek chose this approach due to his experience building web-based tools and his interest in creating an intuitive user interface that aligns with the capstone’s focus on user-friendliness.
- **Emma:**
 - Emma will deepen her understanding of database management by revisiting content from database courses.
 - She will explore additional online resources focused on data optimization techniques, such as indexing and data security.
 - Emma chose this focus because of her background in back-end development and her desire to ensure data integrity and efficiency, both critical to the project’s backend development.
- **Jad:**
 - Jad will refine his knowledge of scheduling algorithms and system optimization by revisiting course materials from previous software engineering courses.
 - He will also explore online tutorials on task automation to improve the efficiency of scheduling features.
 - Jad selected this approach because of his strong analytical skills and interest in making the scheduling feature as effective as possible, which is central to the platform’s functionality.
- **Tuoyo:**
 - Tuoyo will enhance his skills in authentication and security by reviewing past coursework related to software security practices.
 - He will utilize online tutorials that cover modern security protocols (e.g., OAuth, JWT) to ensure secure user access control.
 - Tuoyo selected this approach due to his interest in cybersecurity and the importance of creating a secure login system for the platform.
- **Damien:**

- Damien will focus on system architecture and integration between front-end and back-end components by referring to previous coursework on software design and development.
- He will use online tutorials to better understand system workflows and ensure seamless component interaction.
- Damien chose this approach because he enjoys working on system-level design and aims to ensure the platform runs smoothly across all components.

References

- [1] Government of Canada. (2023). Personal Information Protection and Electronic Documents Act (PIPEDA). <https://www.priv.gc.ca/en/privacy-topics/privacy-laws-in-canada/the-personal-information-protection-and-electronic-documents-act-pipeda/>
- [2] Government of Ontario. (2023). Freedom of Information and Protection of Privacy Act (FIPPA). <https://www.ontario.ca/laws/statute/90f31>
- [3] European Union. (2023). General Data Protection Regulation (GDPR). <https://gdpr-info.eu/>
- [4] W3C. (2023). Web Content Accessibility Guidelines (WCAG) 2.1. <https://www.w3.org/WAI/WCAG21/quickref/>
- [5] U.S. Department of Justice. (2023). Section 508 of the Rehabilitation Act. <https://www.section508.gov/>
- [6] U Sports. (2023). Canadian Interuniversity Sport Guidelines. <https://usports.ca/en/sports/softball>
- [7] McMaster University. (2023). Athletics and Recreation Policies. <https://www.mcmaster.ca/athletics/>
- [8] OWASP Foundation. (2023). OWASP Top 10 Security Standards. <https://owasp.org/www-project-top-ten/>
- [9] ISO/IEC. (2023). ISO/IEC 27001 Information Security Management. <https://www.iso.org/isoiec-27001-information-security.html>

- [10] Fielding, R. T. (2023). REST API Design Standards. <https://restfulapi.net/>

30 Traceability Matrices

This section provides comprehensive traceability between different aspects of the system requirements.

30.1 Requirements to Use Cases Matrix

The following matrix maps functional requirements to their corresponding use cases, showing which requirements are satisfied by which use cases.

Requirement ID	Related Use Cases
FR1 (Login)	UC1 (Login)
FR2 (Role Access)	UC1 (Login)
FR3 (Schedule View)	UC2 (View Schedule)
FR4 (Standings)	UC3 (View Standings)
FR5 (Team Creation)	UC4 (Create Team)
FR6 (Team Joining)	UC5 (Join Team)
FR7 (Schedule Generation)	UC6 (Create Schedule)
FR8 (Reschedule Request)	UC7 (Request Reschedule)
FR9 (View Announcements)	UC8 (View Announcements)
FR10 (Post Announcements)	UC9 (Post Announcements)
FR11 (Override)	UC10 (Rain Cancellation/Override)
FR12 (Report Results)	UC11 (Report Results)
FR13 (Profile Management)	UC8 (View Announcements)

Table 2: Requirements to Use Cases Traceability

[Table 2](#) shows the mapping between requirements and use cases.

30.2 Requirements to Hazards Matrix

This matrix illustrates how requirements address specific hazards identified in the hazard analysis, helping ensure all safety concerns are properly miti-

gated.

Requirement ID	Related Hazards
FR1, FR2	H1 (Authentication Failure)
FR3, FR4, FR7	H2.1 (Invalid Schedule Generation), H2.2 (Missing Games)
FR12	H3 (Failed Score Reporting)
FR5, FR6	H4.1, H4.2 (Team/Player Registration Failures)
All	H5.1, H5.2 (Data Corruption/Unauthorized Access)
FR9, FR10	H6 (Failed Announcement Delivery)

Table 3: Requirements to Hazards Traceability

As shown in [Table 3](#), requirements are mapped to specific hazards.

30.3 Requirements Dependencies Matrix

The dependencies matrix shows which requirements rely on other requirements, helping to plan implementation order and identify potential bottlenecks.

Requirement ID	Dependencies
FR2-FR13	FR1 (Login System)
FR3	FR5 (Team Creation)
FR4	FR12 (Score Reporting)
FR6	FR5 (Team Creation)
FR8	FR7 (Schedule Generation)

Table 4: Requirements Dependencies

[Table 4](#) illustrates the dependencies between requirements.

31 Appendix

31.1 Symbolic Parameters

The definition of the test cases will call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.

parameter	value	unit	description
MAX_RESPONSE_TIME	1	s	maximum acceptable response time for user actions
MAX_LOGIN_ATTEMPTS	5	n/a	maximum failed login attempts before logout
MIN_TEAMS	30	n/a	minimum number of teams supported
MAX_TEAMS	40	n/a	maximum number of teams supported
MAX_TEAM_CAPACITY	50	n/a	maximum number of teams for capacity planning
RESCHEDULE_NOTICE_HOURS	48	h	hours notice required for rescheduling
SUPPORT_RESPONSE_HOURS	24	h	hours to acknowledge support requests
SUPPORT_RESOLUTION_HOURS	48	h	hours to resolve support requests
MAINTENANCE_BUG_DAYS	14	d	days before critical bugs must be fixed
MIN_TEST_USERS	5	n/a	minimum number of real users for testing
SYSTEM_UPTIME	99.9	%	required system uptime percentage
ADMIN_CHANGE_TIME	2	min	minutes for admin to change game rules
ADMIN_CHANGE_SUCCESS	95	%	percentage of successful admin changes
USER_INTERFACE_RATING	90	%	percentage of users who should rate interface as intuitive

TASK_COMPLETION_RATE	85	%	percentage of new users completing basic tasks
LEARNING_SUCCESS_RATE	90	%	percentage of new users completing tasks without support
ACTION_COMPLETION_RATE	95	%	percentage of user actions completing within time limit
BROWSER_COMPATIBILITY_YEARS	5	y	years of browser compatibility
DATA_BACKUP_FREQUENCY_DAYS	7	d	frequency of regular data backups
PAGE_LOAD_TIME_SECONDS	3	s	maximum acceptable page load time
DATABASE_QUERY_TIME_MS	100	ms	maximum acceptable database query response time

32 Likely Changes

This section identifies requirements that are likely or unlikely to change over time, helping to guide future maintenance and development efforts.

32.1 Requirements Likely to Change

The following requirements are anticipated to evolve as the league grows and technology advances:

- **Team Capacity** (FR5, NFR-P5)
 - The maximum number of teams ([MAX_TEAMS](#)) may need to increase as the league grows
 - Team roster size limits may need adjustment based on participation trends
- **Schedule Management** (FR7, FR8)

- Scheduling algorithms may need enhancement to accommodate new game formats
- Rescheduling policies ([RESCHEDULE NOTICE HOURS](#)) may be adjusted based on user feedback
- **User Interface** (NFR-LF1, NFR-LF2)
 - Visual design elements may need updates to stay modern
 - New accessibility requirements may emerge
 - Mobile interface may need optimization for new devices
- **Performance Requirements** (NFR-P1)
 - Response time requirements ([MAX RESPONSE TIME](#)) may become stricter
 - Concurrent user capacity may need to increase
- **Security Standards** (NFR-S1, NFR-S3)
 - Authentication methods may need updates for new security threats
 - Data protection requirements may evolve with new regulations

32.2 Requirements Unlikely to Change

The following requirements are expected to remain stable:

- **Core Game Logic** (FR12)
 - Basic softball scoring and standings calculations
 - Win/loss/tie determination rules
- **Role Hierarchy** (FR1, FR2)
 - Basic roles (commissioner, captain, player)
 - Core permission structure
- **Data Model Fundamentals** (Business Data Model)
 - Basic entity relationships (teams, players, games)
 - Core data attributes for each entity

- **Communication Flow** (FR9, FR10)
 - Basic announcement structure
 - Team-player communication channels
- **Compliance Requirements** (NFR-L1)
 - User data privacy rights
 - Basic legal compliance needs

32.3 Impact Analysis

Understanding which requirements are likely to change helps in:

- Designing flexible architecture that can accommodate future changes
- Planning maintenance schedules and resource allocation
- Prioritizing modular development for components likely to change
- Establishing stable core functionality that can support evolving features