

# Software Requirements Specification for Software Engineering: subtitle describing software

Team 6, Pitch Perfect

Damien Cheung

Jad Haytaoglu

Derek Li

Temituoyo Ugborogho

Emma Wigglesworth

October 20, 2024

# Contents

<b>1</b>	<b>Purpose of the Project</b>	<b>vi</b>
1.1	User Business . . . . .	vi
1.2	Goals of the Project . . . . .	vii
<b>2</b>	<b>Stakeholders</b>	<b>viii</b>
2.1	Client . . . . .	viii
2.2	Customer . . . . .	viii
2.3	Other Stakeholders . . . . .	viii
2.4	Hands-On Users of the Project . . . . .	viii
2.5	Personas . . . . .	ix
2.6	Priorities Assigned to Users . . . . .	ix
2.7	User Participation . . . . .	ix
2.8	Maintenance Users and Service Technicians . . . . .	ix
<b>3</b>	<b>Mandated Constraints</b>	<b>ix</b>
3.1	Solution Constraints . . . . .	ix
3.2	Implementation Environment of the Current System . . . . .	x
3.3	Partner or Collaborative Applications . . . . .	x
3.4	Off-the-Shelf Software . . . . .	x
3.5	Anticipated Workplace Environment . . . . .	x
3.6	Schedule Constraints . . . . .	xi
3.7	Budget Constraints . . . . .	xi
3.8	Enterprise Constraints . . . . .	xi
<b>4</b>	<b>Naming Conventions and Terminology</b>	<b>xi</b>
4.1	Glossary of All Terms, Including Acronyms, Used by Stakeholders involved in the Project . . . . .	xi
<b>5</b>	<b>Relevant Facts and Assumptions</b>	<b>xii</b>
5.1	Relevant Facts . . . . .	xii
5.2	Business Rules . . . . .	xii
5.3	Assumptions . . . . .	xii
<b>6</b>	<b>The Scope of the Work</b>	<b>xiii</b>
6.1	The Current Situation . . . . .	xiii
6.2	The Context of the Work . . . . .	xiii
6.3	Work Partitioning . . . . .	xiv

6.4	Specifying a Business Use Case . . . . .	xv
<b>7</b>	<b>Business Data Model and Data Dictionary</b>	<b>xv</b>
7.1	Business Data Model . . . . .	xv
7.2	Data Dictionary . . . . .	xvii
<b>8</b>	<b>The Scope of the Product</b>	<b>xvii</b>
8.1	Product Boundary . . . . .	xvii
8.2	Product Use Case List . . . . .	xvii
8.3	Individual Product Use Cases (PUC's) . . . . .	xix
<b>9</b>	<b>Functional Requirements</b>	<b>xxi</b>
9.1	Functional Requirements . . . . .	xxi
<b>10</b>	<b>Look and Feel Requirements</b>	<b>xxiv</b>
10.1	Appearance Requirements . . . . .	xxiv
10.2	Style Requirements . . . . .	xxiv
<b>11</b>	<b>Usability and Humanity Requirements</b>	<b>xxiv</b>
11.1	Ease of Use Requirements . . . . .	xxiv
11.2	Personalization and Internationalization Requirements . . . . .	xxiv
11.3	Learning Requirements . . . . .	xxiv
11.4	Understandability and Politeness Requirements . . . . .	xxv
11.5	Accessibility Requirements . . . . .	xxv
<b>12</b>	<b>Performance Requirements</b>	<b>xxv</b>
12.1	Speed and Latency Requirements . . . . .	xxv
12.2	Safety-Critical Requirements . . . . .	xxv
12.3	Precision or Accuracy Requirements . . . . .	xxv
12.4	Robustness or Fault-Tolerance Requirements . . . . .	xxvi
12.5	Capacity Requirements . . . . .	xxvi
12.6	Scalability or Extensibility Requirements . . . . .	xxvi
12.7	Longevity Requirements . . . . .	xxvi
<b>13</b>	<b>Operational and Environmental Requirements</b>	<b>xxvi</b>
13.1	Expected Physical Environment . . . . .	xxvi
13.2	Wider Environment Requirements . . . . .	xxvi
13.3	Requirements for Interfacing with Adjacent Systems . . . . .	xxvii
13.4	Productization Requirements . . . . .	xxvii

13.5 Release Requirements . . . . .	xxvii
<b>14 Maintainability and Support Requirements</b>	<b>xxvii</b>
14.1 Maintenance Requirements . . . . .	xxvii
14.2 Supportability Requirements . . . . .	xxvii
14.3 Adaptability Requirements . . . . .	xxviii
<b>15 Security Requirements</b>	<b>xxviii</b>
15.1 Access Requirements . . . . .	xxviii
15.2 Integrity Requirements . . . . .	xxviii
15.3 Privacy Requirements . . . . .	xxviii
15.4 Audit Requirements . . . . .	xxviii
15.5 Immunity Requirements . . . . .	xxviii
<b>16 Cultural Requirements</b>	<b>xxviii</b>
16.1 Cultural Requirements . . . . .	xxviii
<b>17 Compliance Requirements</b>	<b>xxix</b>
17.1 Legal Requirements . . . . .	xxix
17.2 Standards Compliance Requirements . . . . .	xxix
<b>18 Open Issues</b>	<b>xxix</b>
<b>19 Off-the-Shelf Solutions</b>	<b>xxx</b>
19.1 Ready-Made Products . . . . .	xxx
19.2 Reusable Components . . . . .	xxx
19.3 Products That Can Be Copied . . . . .	xxx
<b>20 New Problems</b>	<b>xxx</b>
20.1 Effects on the Current Environment . . . . .	xxx
20.2 Effects on the Installed Systems . . . . .	xxxi
20.3 Potential User Problems . . . . .	xxxi
20.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product . . . . .	xxxi
20.5 Follow-Up Problems . . . . .	xxxii
<b>21 Tasks</b>	<b>xxxii</b>
21.1 Project Planning . . . . .	xxxii
21.2 Planning of the Development Phases . . . . .	xxxiii

21.2.1	Phase 1: Requirement Gathering and Initial Planning .	xxxiii
21.2.2	Phase 2: Proof of Concept and Design . . . . .	xxxiv
21.2.3	Phase 3: Core Development . . . . .	xxxiv
21.2.4	Phase 4: Verification, Validation, and Revision . . . . .	xxxiv
21.2.5	Phase 5: Final Development and Demonstration . . . . .	xxxiv
21.2.6	Phase 6: Documentation and Wrap-Up . . . . .	xxxv
<b>22</b>	<b>Migration to the New Product</b>	<b>xxxv</b>
22.1	Requirements for Migration to the New Product . . . . .	xxxv
22.2	Data That Has to be Modified or Translated for the New System	xxxv
<b>23</b>	<b>Costs</b>	<b>xxxvi</b>
<b>24</b>	<b>User Documentation and Training</b>	<b>xxxvi</b>
24.1	User Documentation Requirements . . . . .	xxxvi
24.2	Training Requirements . . . . .	xxxvii
<b>25</b>	<b>Waiting Room</b>	<b>xxxvii</b>
<b>26</b>	<b>Ideas for Solution</b>	<b>xxxvii</b>

## Revision History

Date	Version	Notes
Oct 11, 2024	1.0	Initial Creation - Final

# 1 Purpose of the Project

## 1.1 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.

## 1.2 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): Implement a secure login system using JSON Web Tokens (JWT) that allows different levels of access for commissioners, captains, and players, 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 accurate 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.



## **2 Stakeholders**

### **2.1 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.

### **2.2 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.

### **2.3 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.

### **2.4 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.

## **2.5 Personas**

Player Persona: A student looking to join a team, participate in recreational activities/games, check their game schedule, and track their performance. Captain Persona: An organized individual responsible for managing team logistics, rescheduling games, and ensuring effective communication. Commissioner Persona: A knowledgeable overseer who ensures the league operates smoothly, creates game schedules, handles disputes, and communicates important updates to all users.

## **2.6 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.

## **2.7 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.

## **2.8 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.

# **3 Mandated Constraints**

## **3.1 Solution Constraints**

Constraints that limit the design of the solution:

- **Scalability:** The initial version of the platform must support 30-40 teams, with future scalability to accommodate more teams.
- **Device Accessibility:** The system must be optimized for desktops, tablets, and mobile phones.

### 3.2 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.

### 3.3 Partner or Collaborative Applications

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

### 3.4 Off-the-Shelf Software

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

### 3.5 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.

### 3.6 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.

### 3.7 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.

### 3.8 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.

## 4 Naming Conventions and Terminology

### 4.1 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.

## 5 Relevant Facts and Assumptions

### 5.1 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.

### 5.2 Business Rules

- **Team Registration:** Teams must be registered by captains before the league starts.
- **Scheduling Requests:** Rescheduling requests must be made at least 48 hours in advance.

### 5.3 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.

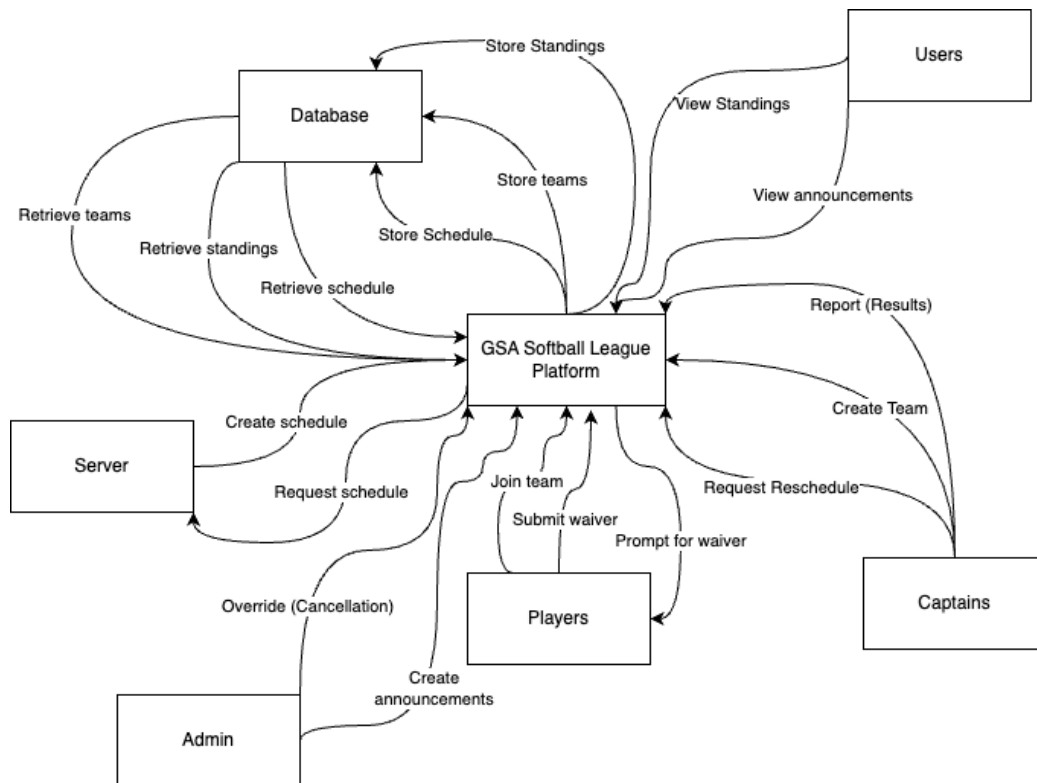


Figure 1: Context diagram.

## 6 The Scope of the Work

### 6.1 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.

### 6.2 The Context of the Work

See *Figure 1*.

### 6.3 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.

## 6.4 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.

# 7 Business Data Model and Data Dictionary

## 7.1 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.

## 7.2 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.

# 8 The Scope of the Product

## 8.1 Product Boundary

See Context Diagram (*Figure 1.*)

## 8.2 Product Use Case List

See *Figure 2.*

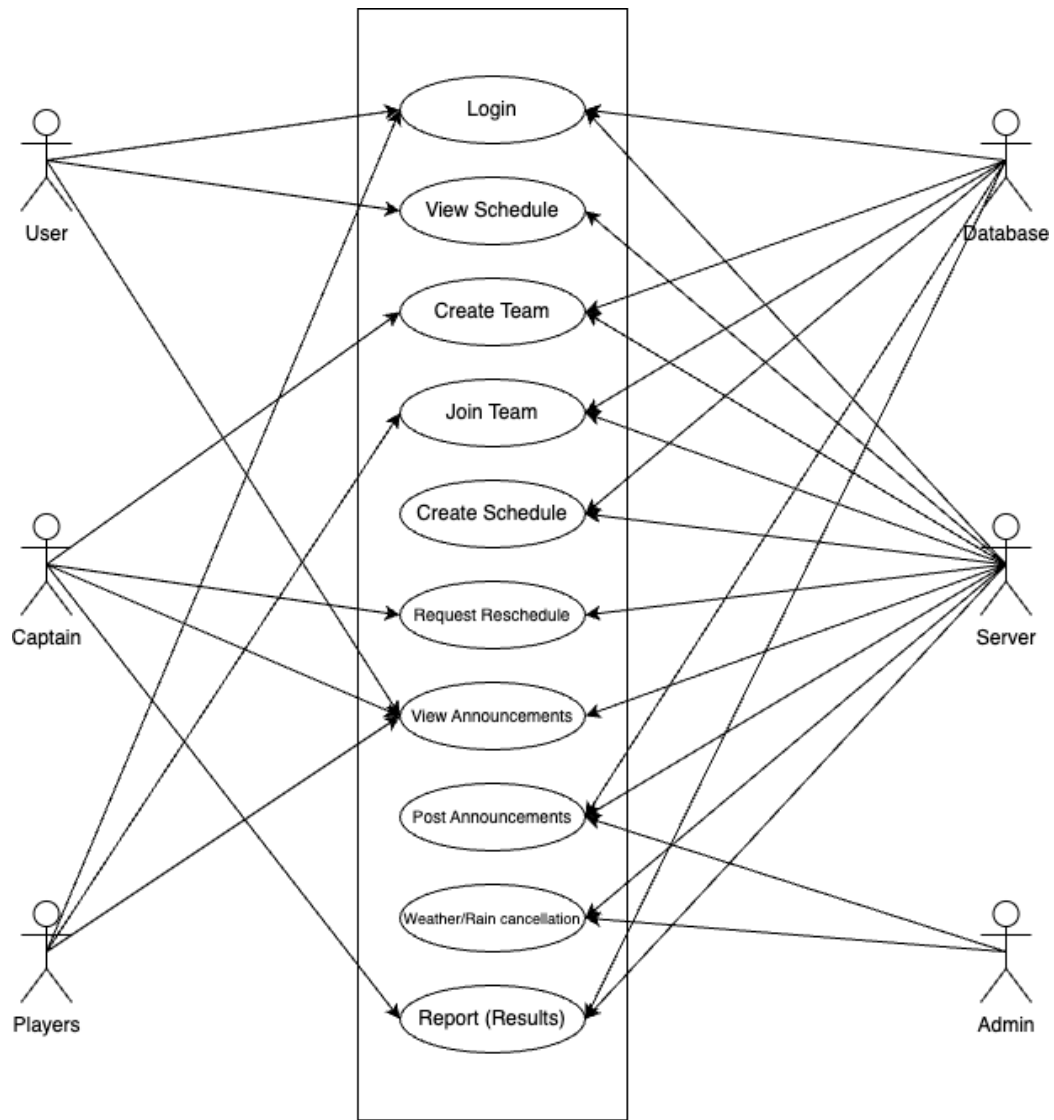


Figure 2: Use Case List.

### 8.3 Individual Product Use Cases (PUC's)

1. Product Use Case Name: 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.
2. Product Use Case Name: 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.
3. Product Use Case Name: 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.
4. Product Use Case Name: 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.
5. Product Use Case Name: 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.

6. Product Use Case Name: 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.
7. Product Use Case Name: 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.
8. Product Use Case Name: 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.
9. Product Use Case Name: 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.
10. Product Use Case Name: 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.

11. Product Use Case Name: Report

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.

## 9 Functional Requirements

### 9.1 Functional Requirements

**Requirement #:** 1    **Requirement Type:** 9    **Event/Use Case:** 1

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

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

**Fit Criterion:** No unauthorized users are able to log in and access role-based features after entering credentials.

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

---

**Requirement #:** 2    **Requirement Type:** 9    **Event/Use Case:** 1

**Description:** The platform shall distinguish between different user roles (commissioner, captain, player) and role-specific access after login.

**Rationale:** Provides tailored access to players, captains, admin.

**Fit Criterion:** Different user-roles only have access to the features assigned to their roles after login.

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

---

**Requirement #:** 3    **Requirement Type:** 9    **Event/Use Case:** 2

**Description:** The platform shall allow users to view the game schedule in-

cluding date, time, location, opponent.

**Rationale:** Facilitates easy access to schedule information to provide league members with aligned game information.

**Fit Criterion:** Schedules and game information are viewable by users.

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

---

**Requirement #:** 4    **Requirement Type:** 9    **Event/Use Case:** 3

**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 #:** 5    **Requirement Type:** 9    **Event/Use Case:** 4

**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 #:** 6    **Requirement Type:** 9    **Event/Use Case:** 5

**Description:** The platform shall allow players to request to join a team, and notify the captain for approval.

**Rationale:** Simplifies team membership management and formation.

**Fit Criterion:** Players can submit join requests, and captains can approve or reject them and the system automatically updates the roster.

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

---

**Requirement #:** 7    **Requirement Type:** 9    **Event/Use Case:** 6

**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 #:** 8    **Requirement Type:** 9    **Event/Use Case:** 7

**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 #:** 9    **Requirement Type:** 9    **Event/Use Case:** 8

**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 #:** 10    **Requirement Type:** 9    **Event/Use Case:** 9

**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 #:** 11    **Requirement Type:** 9    **Event/Use Case:** 10

**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 #:** 12    **Requirement Type:** 9    **Event/Use Case:** 11  
**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

## 10 Look and Feel Requirements

### 10.1 Appearance Requirements

The platform must feature a modern, intuitive interface with consistent visual elements across all views.

### 10.2 Style Requirements

The platform shall follow a consistent color palette, typography, and layout.

## 11 Usability and Humanity Requirements

### 11.1 Ease of Use Requirements

The product must be easy to use for graduate students.

### 11.2 Personalization and Internationalization Requirements

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

### 11.3 Learning Requirements

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

The platform must include easily accessible help documentation for users to learn how to perform tasks and resolve issues independently.

## **11.4 Understandability and Politeness Requirements**

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

## **11.5 Accessibility Requirements**

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

# **12 Performance Requirements**

## **12.1 Speed and Latency Requirements**

The platform must respond to user actions (e.g., scheduling, score reporting) within 1 second.

## **12.2 Safety-Critical Requirements**

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

## **12.3 Precision or Accuracy Requirements**

The platform must calculate and display the standings with 100% accuracy.

The platform must accurately match game preferences with available slots to avoid conflicts or scheduling errors and maximize adherence with team schedule preferences.

## **12.4 Robustness or Fault-Tolerance Requirements**

The platform shall handle common errors (e.g., failed database connections) without crashing, and users shall be able to retry actions if errors occur.

## **12.5 Capacity Requirements**

The platform must be able to store and handle the scheduling for at minimum 50 teams in the league.

## **12.6 Scalability or Extensibility Requirements**

The platform shall support up handling 50 teams without performance issues.

## **12.7 Longevity Requirements**

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

# **13 Operational and Environmental Requirements**

## **13.1 Expected Physical Environment**

The platform must be accessible on various devices, such as desktops, laptops, smartphones, and tablets. The platform will also require a stable internet connection. Servers are needed to store information in the database, such as team rosters, photos, etc.

## **13.2 Wider Environment Requirements**

Although the platform will be accessible across various devices, we want our platform to be accessed across multiple browsers, such as Chrome, Safari, Firefox, Edge, etc. Even if our platform is accessible by smartphones and tablets, it must have a responsive design that allows the user to reach all features when using a mobile device.

### **13.3 Requirements for Interfacing with Adjacent Systems**

Our platform may need to integrate with payment gateways to handle payment management, such as PayPal. It would be useful if our platform could integrate with popular calendar systems, such as Google Calendar. There might be some features that could benefit from CSV / Excel import or export, such as roster data and match results.

### **13.4 Productization Requirements**

Our platform must handle scaling well, such as an increased number of divisions, teams, players, etc. Depending on the league, the managers should be able to manage anything related to game rules, so that the information is clear to users of the product (in this case, the players of the league).

### **13.5 Release Requirements**

Version control is extremely important, previous platforms must be saved in version control so we can track changes and rollback if unnecessary. Before any production release, thorough testing must be done to gather feedback. If possible, testing with real users would be the most beneficial.

## **14 Maintainability and Support Requirements**

### **14.1 Maintenance Requirements**

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

### **14.2 Supportability Requirements**

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

### **14.3 Adaptability Requirements**

An admin user should be able to change game rules within 2 minutes, such as swapping a team's game schedule with another if needed.

## **15 Security Requirements**

### **15.1 Access Requirements**

There must be role-based access control. Administrator roles must have access to everything. Team managers should have access to team management. Players should be restricted to a degree.

### **15.2 Integrity Requirements**

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

### **15.3 Privacy Requirements**

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

### **15.4 Audit Requirements**

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

### **15.5 Immunity Requirements**

The platform should be protected against malicious web attacks.

## **16 Cultural Requirements**

### **16.1 Cultural Requirements**

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.

## 17 Compliance Requirements

### 17.1 Legal Requirements

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.

### 17.2 Standards Compliance Requirements

The platform must meet typical web standards, such as readable font, colours that are easy to look at, easy to navigate, etc.

## 18 Open Issues

- **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.

## **19 Off-the-Shelf Solutions**

### **19.1 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.

### **19.2 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.

### **19.3 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.

## **20 New Problems**

### **20.1 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 pro-

vide long-term benefits by reducing maintenance and manual input required by the commissioners and administrators.

## **20.2 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.

## **20.3 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.

## **20.4 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.

## 20.5 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.

## 21 Tasks

### 21.1 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 requirement 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 #1, #2): Secure login and user roles (commissioner, captain, player).
- **Team Management** (Requirements #5, #6): Captains create teams, select divisions, and manage rosters. Players can request to join teams.
- **Schedule Generation** (Requirement #7): Automatically generate schedules, ensuring even distribution and honoring team preferences for dates/times.

- **Score Reporting** (Requirements #11, #12): Captains report game scores and forfeits, updating standings automatically.
- **Rescheduling System** (Requirements #8, #11): 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 #6): Players can log in, view team information, and request to join teams.
- **Rescheduling Queue Management** (Requirement #8): Manage multiple rescheduling requests and enforce time restrictions (e.g., within 24 hours of a game).
- **Announcements** (Requirements #9, #10): 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.

## 21.2 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:

### 21.2.1 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.

### **21.2.2 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.

### **21.2.3 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.

### **21.2.4 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.

### **21.2.5 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.

### 21.2.6 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.

## 22 Migration to the New Product

### 22.1 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.

### 22.2 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 require-

ments:

- **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.

## 23 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.

## 24 User Documentation and Training

### 24.1 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.

## 24.2 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.

## 25 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.

## 26 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.