

Baseball League Registration System Documentation

by

Project Group:

Zach Butler

Catlin Kramka

Rhett Godwin

INFO-C451 System Implementation

Jafrina Jabin

May 4, 2024

Table of Contents:

Customer Problem Statement and System Requirements.....	3
Functional Requirement Specification	7
System Sequence Diagram.....	11
Activity Diagram.....	13
User Interface Specification.....	17
Traceability Matrix.....	25
System Architecture and System Design.....	28
User Interface Design and Implementation.....	31
Design of Tests.....	34
Project Plan.....	36

Customer Problem Statement and System Requirements

Problem Statement

Almost all signup programs today involve some kind of electronic signup system to keep track of those who register for an event, and that's exactly what is needed for Innovative Solutions Corporation's Summer Baseball League. A good signup system includes a login, ensuring only those affiliated with the host of the event can get involved. It also enables users to sign up for a position within the event. In this case, each user is allowed to sign up for one position on a baseball team within the league. Finally, it is also helpful to see who has already signed up, especially when each person signing up is given one unique position that, once occupied, cannot be taken. An overview of the teams and available positions will be available for users to view once they pass the login page.

Glossary

- **Username:** A username will be defined by an existing employee username within the company.
 - **Password:** Passwords will be defined by the password assigned to the company employees.
 - **Team-Sign Up:** The user will select the desired position and team. This information will be recorded and entered into the database. If the position and team are full, the user will be notified.
 - **Baseball Teams:** The company will have multiple baseball teams for employees to join.
-

System Requirements

No.	Priority Weight	Description
REQ-1 Signup for the desired team	High	The user will be able to select the team that they would like to join.
REQ-2 Signup for the desired	High	The user will be able to select the team position

position		that they would like to have.
REQ-3 Notify the team if full	High	The system will send a notification if the team is full.
REQ-4 Notify position filled	High	The system will send a notification to the user if the desired position has already been taken.
REQ-5 View Team and Players	High	The system will allow users to view the teams, the players, and their positions.
REQ-6 Login	High	The login will allow only an employee/owner of the company to login to the sign-up system.
REQ-7 Edit	Low	The option of being able to edit team info.

Nonfunctional Requirements

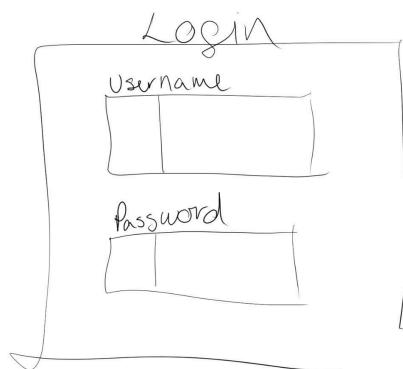
Type	Priority Weight	Description
Functionality	High	Provide accurate data at real-time to ensure users can sign up for positions that are truly available.
Usability	High	Ensure the system is easy to use and that the given information is not hard to

		read. Proper documentation included.
Reliability	High	Ensure the server running the signup system remains stable so signup is always available.
Performance	Medium	Guarantee only needed resources are used to support the signup system.
Supportability	Medium	Release continuous updates that support the scalability of the signup system, allowing it to meet the increased/decreased number of participants/teams in future leagues.

User Interface Requirements

1. Login Page

- **Priority:** High
- **Description:** Screen that displays a big, easy-to-read box where users can enter their username and password to log into the system. The company logo is at the top of the page.
- **Sketch:**



2. Home/Baseball Team Signup Page

- **Priority:** High
- **Description:** Home page split into multiple sections. One section is dedicated to position and team selection/signup; the other is an overview of the current teams and their players.
- **Sketch:**

Innovative Solutions Corp
Annual Baseball Tech Tourney Signup Sheet

Team One	Team Two	Team Three	Team Four
First Name	Last Name	Position	
Rhett	Godwin	Short Stop	
Sam	Smith	Pitcher	
test	test	test	

Plan of Work

Weeks 1- 2: Setup the development environment and become familiar with project requirements. Establish the framework and architecture of the system, construct databases, and wireframe an initial UI design for the page.

Weeks 3-6: Development of the system. Code and implement a login system. Code and implement the Home page. Code and implement the backend.

Weeks 7-8: Test and Deployment: Rigorous testing, debugging and final testing.

Weeks 9-11: Review and improve any implemented features based on customers' feedback.

Functional Requirement Specification

Stakeholders:

- Company CEO
- Company owners
- Company president
- Company managers
- Company employees

Actors and Goals:

Primary:

- Staff, Owners, Managers and Subordinates will be able to login to the Baseball Registration system and sign-up to be a position of a team.

Secondary:

- Admin - Will be able to login to the Baseball Registration system and edit team information. Remove players from the team.
- System - Will give details on available teams and positions, also displaying when a team, or the league itself, is full.

Use Cases:

(2 points indicates one engineer day's workload, like in assignment prompt)

Admin (total: 32)

- Edit team information: Change info about the team, like team name, etc. (4)
- Edit player information: Change info about player, like name, position, etc. (4)
- Remove player: Remove player from a team/the league. (4)
- Login: Login to admin account. (2)
- Logout: Logout of admin account. (2)
- View registered players: Look at players registered for different teams. (2)
- View current teams: Look at teams currently available for signup. (2)
- View team details: Look at details of team, like number of open positions, etc. (4)
- Search for players: Search the system for currently registered players. (4)
- View player details: View details of player, like name, etc. (4)

Staff (Owners, Managers, Subordinates) (total: 28)

- Login: Login to league signup system. (2)
- Logout: Logout of league signup system. (2)
- Sign-up for team: Initiate signup sequence. (2)
- Select position and team: Select the desired position to play, and desired team to play on. (2)
- Drop out: Drop out of team/league. (4)
- Search for players: Search the system for currently registered players. (4)

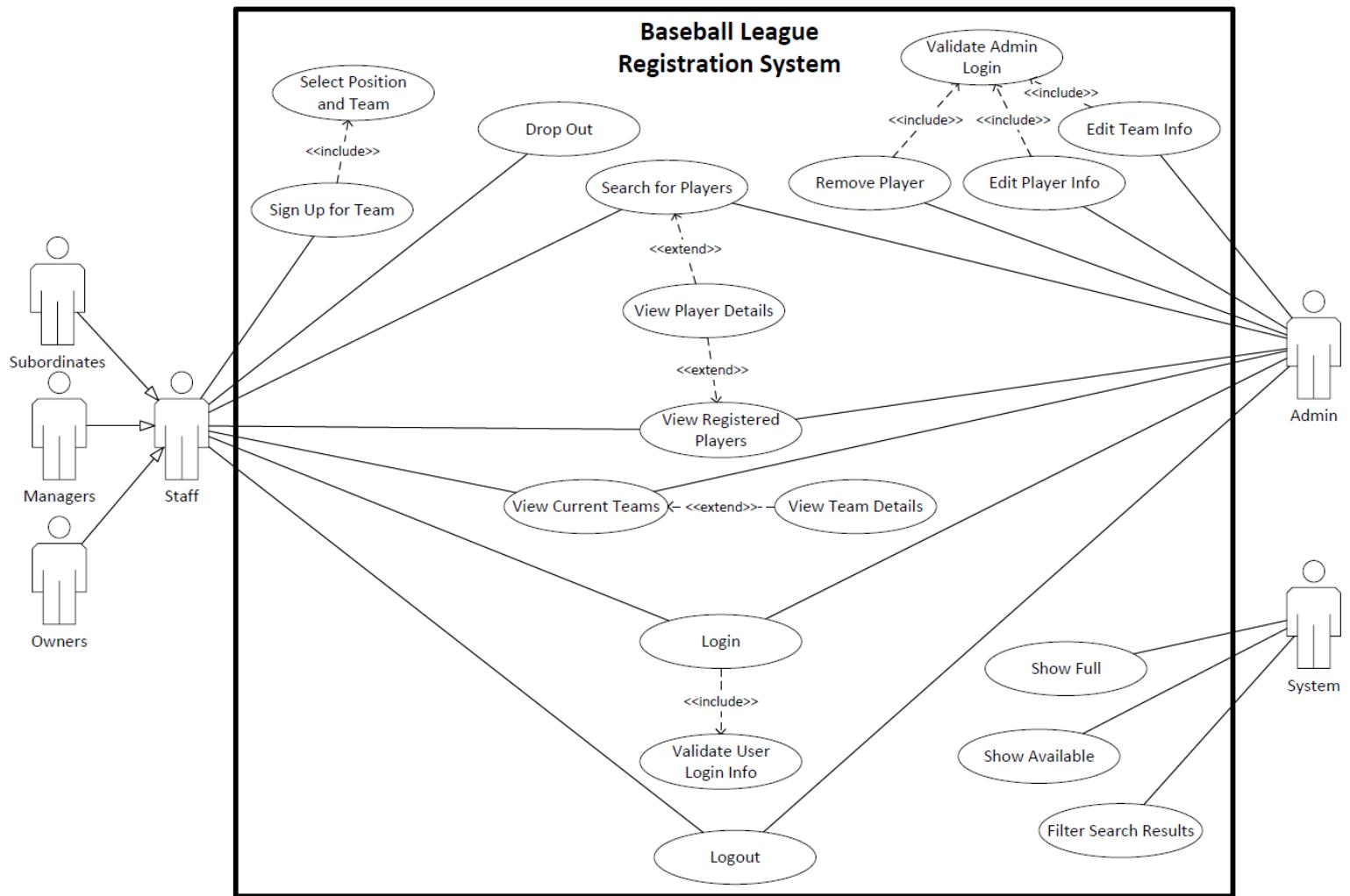
- View player details: View details of player, like name, etc. (4)
- View registered players: Look at players registered for different teams. (2)
- View current teams: Look at teams currently available for signup. (2)
- View team details: Look at details of said team, like number of open positions, etc. (4)

System (total: 10)

- Show full: Show when all teams are full and no open positions remain (4)
- Show available: Show available positions on teams (2)
- Filter search results: Filter database to return search results wanted by user (4)

Use Case Diagram:

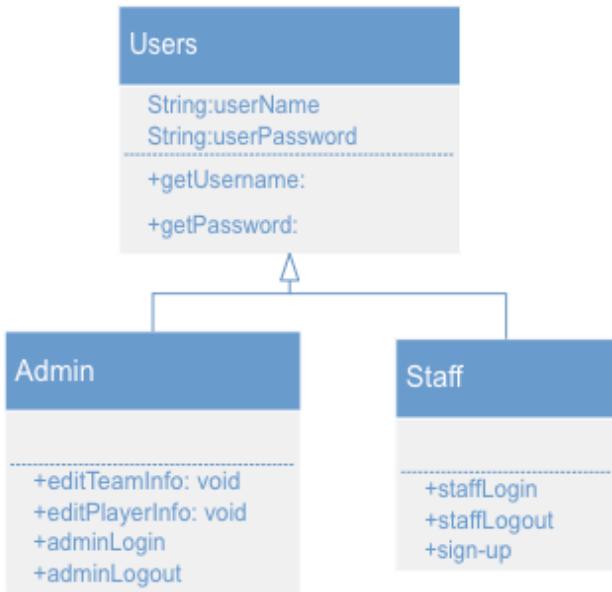
<<extend>> shows what a user can do after a use case if they choose to, while <<include>> shows what absolutely will be done after a use case, regardless of whether the user wants it to be done or not.



Class Diagram:

Users:

Based on the requirements we have two types of users, Admin and Staff. Admin and Staff extends Users. They Inert the Users class



Teams and Players Class

Based on system requirements this is the Team class. The players class is a part of the Team class.



The Class Diagram is below

The relationship between classes consist of:

Composition:

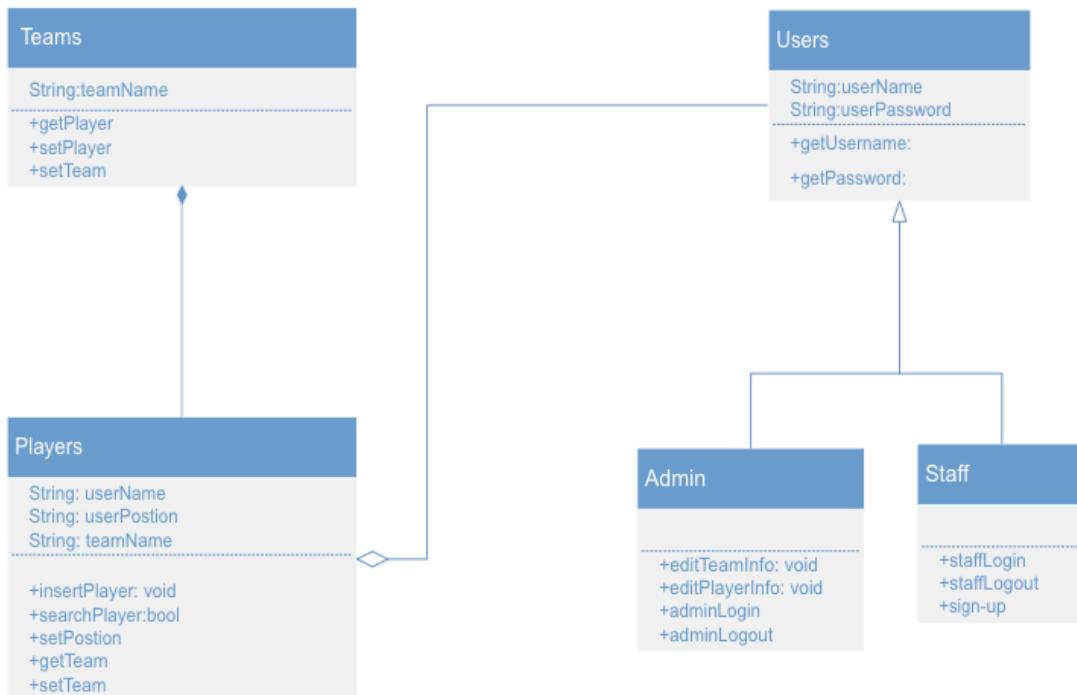
The Team class shares a composition relationship with the Players class. The team is made up of players, and the players cannot exist separately from the team. If a player doesn't have a team to play for, they are just a regular user, not a player.

Aggregation:

The User class shares an aggregation relationship with the Players class. Users can exist outside of Players, however a user is part of the Player class.

Inheritance:

The Staff class and Admin class share an inheritance relationship with the User class. Both Staff and Admin are a type of User.



Sequence Diagrams

Use Case: Sign up

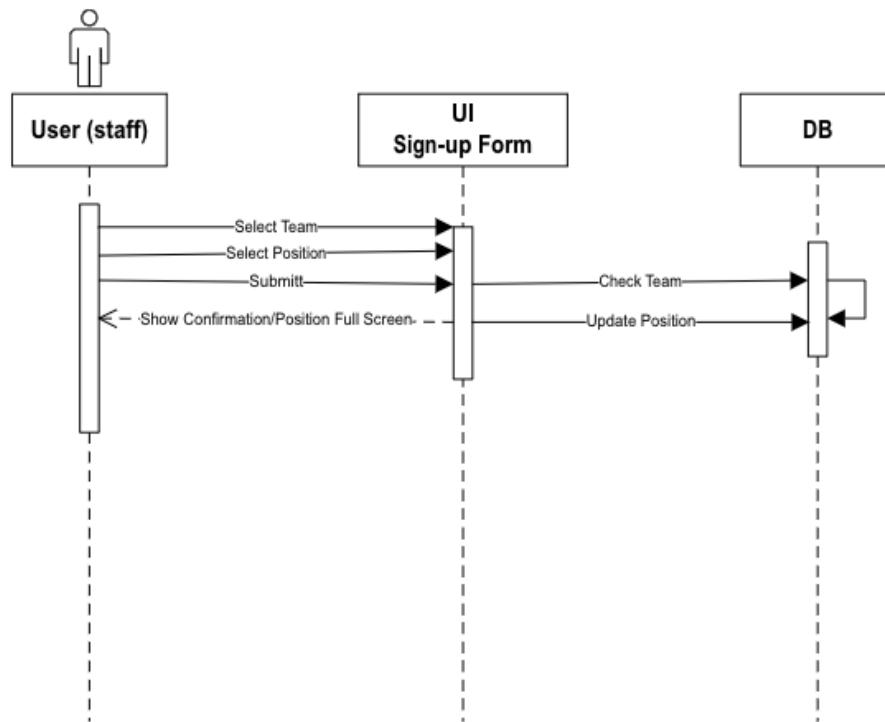
Actor: User (Staff)

Objects: UI Sign-up Form, Database

Steps:

1. User selects the team
2. User selects the position
3. User Submits
4. UI passes data to validate in the Database
5. Upon validation the Database updates
6. UI shows confirmation screen or position full.

Sign-up



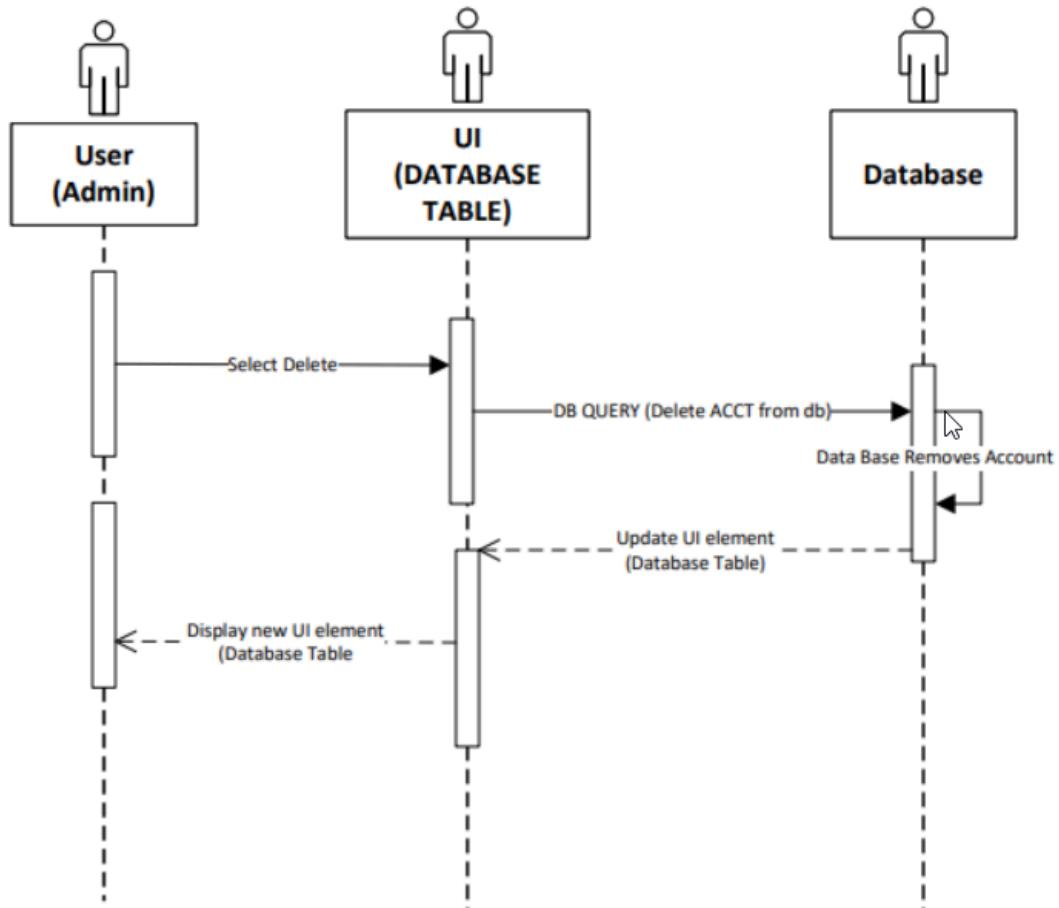
Use Case: Remove Player

Actor: Admin

Objects: UI (Database Table), Database

Steps:

1. User clicks the delete action(link)
2. UI action queries the database to remove account
3. Database Removes the account
4. Updates UI (Database Table)
5. Display new Database Table



Activity Diagrams

Use Case: Login

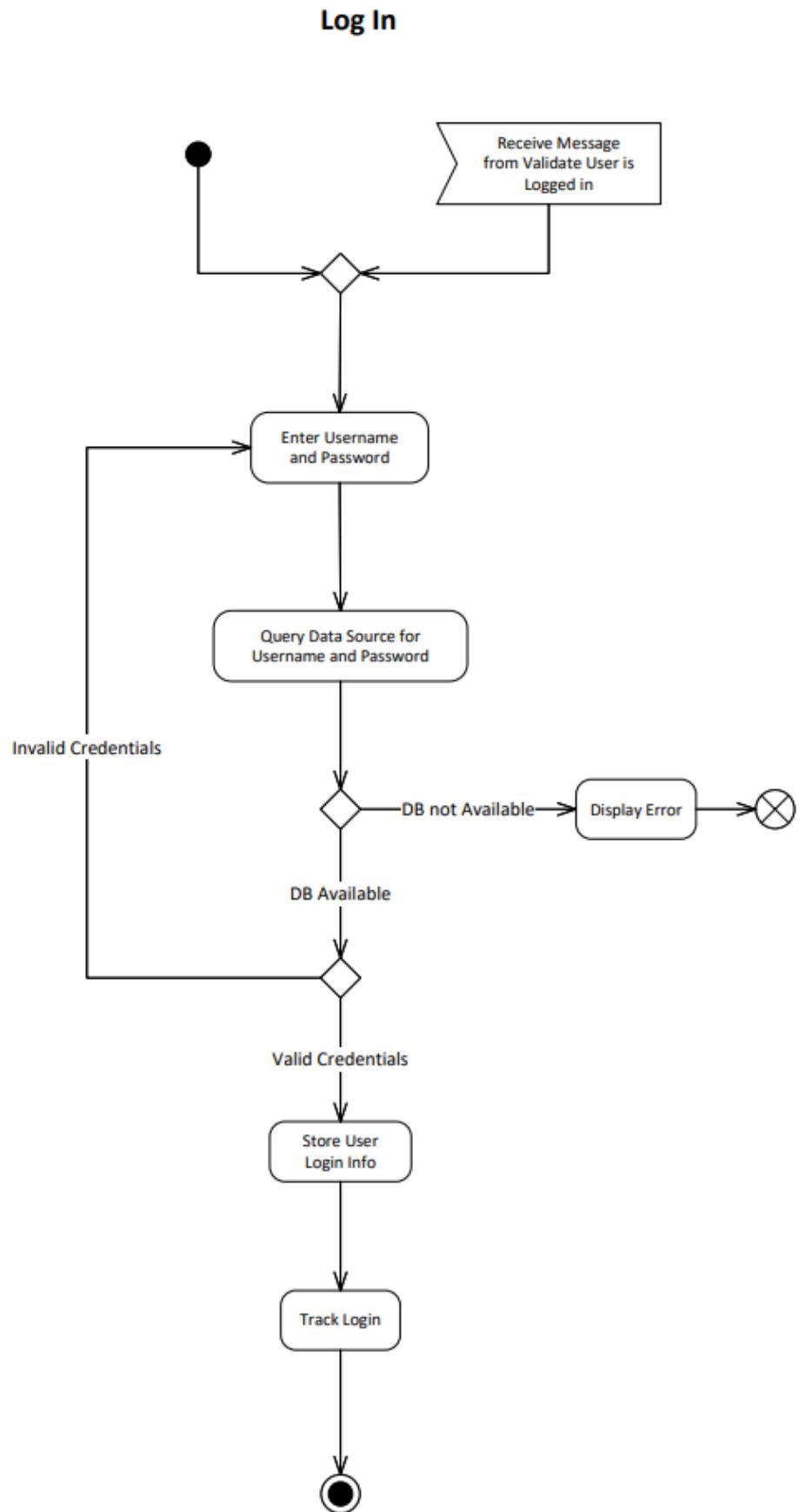
States

- Initial State: The user begins the login process.
- Final State: 1. The user's credentials are approved by the database and the user is let into the baseball signup system. 2. The user's credentials are denied and the user is not let into the system. 3. Database isn't available and the user can't login.

Actions

The user enters their company given username and password into the login page of the system. The system compares what the user enters with the login information recorded in the database. The system either approves the user's login and lets them into the registration system, or it denies the login and the user isn't let in.

(Diagram on following page)



Use Case: Sign up

States

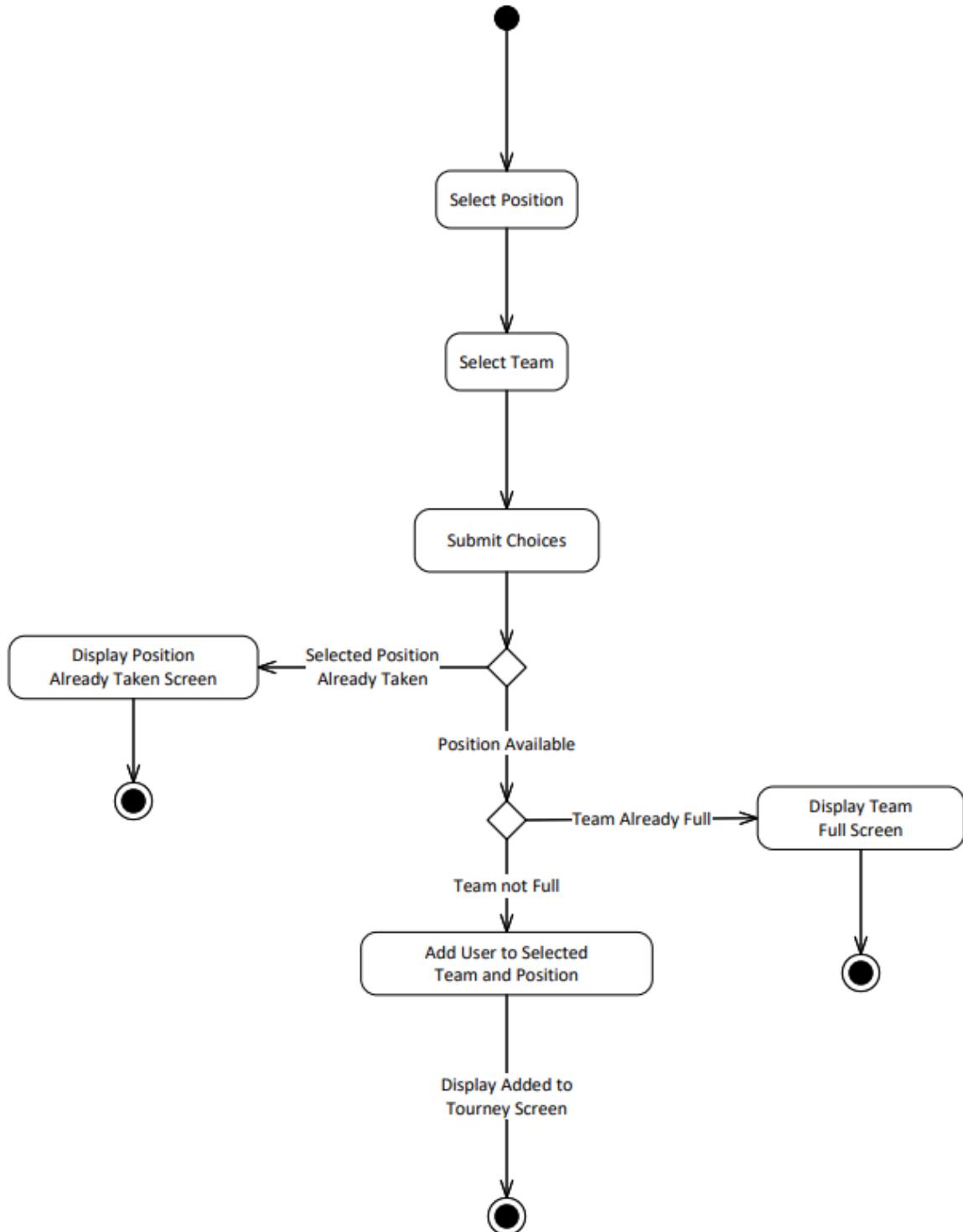
- Initial State: The customer wants to join a team.
- Final State: 1. User submits the desired position and team to the database and is placed on said team in said position. 2. User submits the desired position and team to the database but the position is already taken. 3. User submits the desired position and team to the database but the team is already full.

Actions

The user selects their desired position. The user then selects their desired team. The user clicks the signup button to sign up for the desired position and team they selected. The system approves the selection and sets them in their desired position and team if it is available, or the system declines because the selected position/team is already taken/full.

(Diagram on following page)

Sign Up



User Interface Specifications

Preliminary Design

Use Case: Sign up

Click Desired Position, Team & Sign Up button

Current Team Roster		
Player Name	Current Team	Position
Zach Butler	Cubs	Left Field
Cat Kramka	Cubs	Second Base
Rhett Godwin	Jaguars	Catcher



View the confirmation page (see next page)

↓

What team and position are you interested in?

Desired Position <ul style="list-style-type: none"> <input type="checkbox"/> Left Field <input type="checkbox"/> Center Field <input type="checkbox"/> Right Field <input type="checkbox"/> Short Stop <input type="checkbox"/> First Base <input type="checkbox"/> Second Base <input type="checkbox"/> Third Base <input type="checkbox"/> Pitcher <input type="checkbox"/> Catcher 	Display By Team <input style="margin-right: 10px; border: 1px solid #ccc; padding: 2px 10px;" type="button" value="Cougars"/> <input style="border: 1px solid #ccc; padding: 2px 10px;" type="button" value="Cubs"/> <input style="border: 1px solid #ccc; padding: 2px 10px;" type="button" value="Jaguars"/> <input style="border: 1px solid #ccc; padding: 2px 10px;" type="button" value="Reds"/>
---	--

Current Team Roster		
Player Name	Current Team	Position
Zach Butler	Cubs	Left Field
Cat Kramka	Cubs	Second Base
Rhett Godwin	Jaguars	Catcher

[Sign Up](#)

Auto Return to Sign Up page. User views table with their added name, position and team.

Use Case: Remove Player

Database Admin Page

Add New Data Logout

Players/Teams DataBase

Name	Team	Position	Action
Zach Butler	Cougars	Left Field	Edit Delete
Cat Kramka	Cubs	Second Base	Edit Delete
Rhett Godwin	Cougars	Short Stop	Edit Delete
Rhett godwin	Jaguars	Left Field	Edit Delete
Cat Kramka	Cougars	Third Base	Edit Delete
Cat Kramka	Reds	Left Field	Edit Delete

Employee Accounts Database

Add New Data

Name	Password	Email	Action
Rhett Godwin	test	Rhett@test.com	Edit Delete
Zach Butler	test	zach@test.com	Edit Delete
Cat Kramka	test	Cat@test.com	Edit Delete
admin	admin	admin@test.com	Edit Delete

Admin clicks the delete action link



Database Admin Page

Add New Data Logout

Players/Teams DataBase

Name	Team	Position	Action
Zach Butler	Cougars	Left Field	Edit Delete
Cat Kramka	Cubs	Second Base	Edit Delete
Rhett Godwin	Cougars	Short Stop	Edit Delete
Rhett godwin	Jaguars	Left Field	Edit Delete
Cat Kramka	Cougars	Third Base	Edit Delete
Cat Kramka	Reds	Left Field	Edit Delete

Employee Accounts Database

Add New Data

Name	Password	Email	Action
Rhett Godwin	test	Rhett@test.com	Edit Delete
Zach Butler	test	zach@test.com	Edit Delete
Cat Kramka	test	Cat@test.com	Edit Delete
admin	admin	admin@test.com	Edit Delete

Database sends a popup window to confirm Admin choice. Admin clicks to confirm or cancel.



Database Admin Page

Add New Data Logout

Players/Teams DataBase

Name	Team	Position	Action
Zach Butler	Cougars	Left Field	Edit Delete
Cat Kramka	Cubs	Second Base	Edit Delete
Rhett Godwin	Cougars	Short Stop	Edit Delete
Rhett godwin	Jaguars	Left Field	Edit Delete
Cat Kramka	Cougars	Third Base	Edit Delete

Employee Accounts Database

Add New Data

Name	Password	Email	Action
Rhett Godwin	test	Rhett@test.com	Edit Delete
Zach Butler	test	zach@test.com	Edit Delete
Cat Kramka	test	Cat@test.com	Edit Delete
admin	admin	admin@test.com	Edit Delete

Admin Page is updated with the player removed.

Use Case: Add New Data

Database Admin Page

Add New Data Logout

Players/Teams DataBase

Name	Team	Position	Action
Zach Butler	Cougars	Left Field	Edit Delete
Cat Kramka	Cubs	Second Base	Edit Delete
Rhett Godwin	Cougars	Short Stop	Edit Delete
Rhett godwin	Jaguars	Left Field	Edit Delete
Cat Kramka	Cougars	Third Base	Edit Delete
Cat Kramka	Reds	Left Field	Edit Delete

Employee Accounts Database

Add New Data

Name	Password	Email	Action
Rhett Godwin	test	Rhett@test.com	Edit Delete
Zach Butler	test	zach@test.com	Edit Delete
Cat Kramka	test	Cat@test.com	Edit Delete
admin	admin	admin@test.com	Edit Delete

Admin clicks Add New Data link



Add Data

MUST UPDATE ALL FIELDS

[Home](#)

Player Name	<input type="text"/>
Team	<input type="text"/>
Position	<input type="text"/>
	<input type="button" value="Add"/>

Admin enters data into each field and clicks add.



Data added successfully!

[View Result](#)

Confirmation page is shown and admin clicks View Result.



Database Admin Page

Add New Data Logout

Players/Teams DataBase

Name	Team	Position	Action
Zach Butler	Cougars	Left Field	Edit Delete
Cat Kramka	Cubs	Second Base	Edit Delete
Rhett Godwin	Cougars	Short Stop	Edit Delete
Rhett godwin	Jaguars	Left Field	Edit Delete
Cat Kramka	Cougars	Third Base	Edit Delete
Cat Kramka	Reds	Short Stop	Edit Delete

Employee Accounts Database

Add New Data

Name	Password	Email	Action
Rhett Godwin	test	Rhett@test.com	Edit Delete
Zach Butler	test	zach@test.com	Edit Delete
Cat Kramka	test	Cat@test.com	Edit Delete
admin	admin	admin@test.com	Edit Delete

Navigates back to Admin Page, with updated table.

Use Case: Display Teams

Innovative Soulutions Corporation Profile Logout



Annual Baseball Tech Tourney Signup

Welcome back, rhett godwin! Ready to play some baseball?



What team and position are you interested in?

Desired Position: <input type="checkbox"/> Left Field <input type="checkbox"/> Center Field <input type="checkbox"/> Right Field <input type="checkbox"/> Short Stop <input type="checkbox"/> First Base <input type="checkbox"/> Second Base <input type="checkbox"/> Third Base <input type="checkbox"/> Pitcher <input type="checkbox"/> Catcher	Display By Team: <input checked="" type="button" value="Cougars"/> <input type="button" value="Cubs"/> <input type="button" value="Jaguars"/> <input type="button" value="Reds"/> <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th colspan="3" style="text-align: center;">Current Team Roster</th> </tr> <tr> <th style="text-align: center;">Player Name</th> <th style="text-align: center;">Current Team</th> <th style="text-align: center;">Position</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">Zach Butler</td> <td style="text-align: center;">Cubs</td> <td style="text-align: center;">Left Field</td> </tr> <tr> <td style="text-align: center;">Cat Kramka</td> <td style="text-align: center;">Cubs</td> <td style="text-align: center;">Second Base</td> </tr> <tr> <td style="text-align: center;">Rhett Godwin</td> <td style="text-align: center;">Jaguars</td> <td style="text-align: center;">Catcher</td> </tr> </tbody> </table>	Current Team Roster			Player Name	Current Team	Position	Zach Butler	Cubs	Left Field	Cat Kramka	Cubs	Second Base	Rhett Godwin	Jaguars	Catcher
Current Team Roster																
Player Name	Current Team	Position														
Zach Butler	Cubs	Left Field														
Cat Kramka	Cubs	Second Base														
Rhett Godwin	Jaguars	Catcher														

Team Selection:
 Jaguars
 Cougars
 Reds
 Cubs

User clicks on the team button.



Player Name	Current Team	Position
Zach Butler	Cougars	Left Field
Rhett Godwin	Cougars	Short Stop
Cat Kramka	Cougars	Third Base

Current Team Roster Window opens.

Use Case: View Player Details

Innovative Soulutions Corporation [Profile](#) [Logout](#)



Annual Baseball Tech Tourney Signup

ach butler! Ready to play some baseball?

What team and position are you interested in?

Desired Position:

- Left Field
- Center Field
- Right Field
- Short Stop
- First Base
- Second Base
- Third Base
- Pitcher
- Catcher

Team Selection:

- Jaguars
- Cougars
- Reds
- Cubs

Sign Up

Display By Team

[Cougars](#) [Cubs](#) [Jaguars](#) [Reds](#)

Current Team Roster		
Player Name	Current Team	Position
Zach Butler	Cougars	Left Field
Cat Kramka	Cubs	Second Base
Rhett Godwin	Cougars	Short Stop
Rhett godwin	Jaguars	Left Field
zach butler	Reds	Short Stop

User clicks on the profile button at the top right of the page.



Innovative Soulutions Corporation [Profile](#) [Logout](#)

Profile Page

Your account details are below:

Username:	zach butler
Email:	zach@test.com

[Return Home](#)

Profile/player details page opens.

User Effort Estimation

Usage Scenario	Navigation	Clicks	Keystrokes
User signs-up for a position	Sign-up page, Confirmation page	3	0
Remove Player	Admin Page	2	0
Add New Data	Admin Page, Add Data Fields Page, Confirmation page	3	0-100
View Teams	Sign-up page, Team Roster page	1	0
View Player Details	Profile Page	1	0

Additional Information

All of the screens shown above are accessible to the user after they have already logged in. The screen below is what a user who is not yet logged in would see. When the user enters their username and password and hits submit, the data will be verified with the database and then the user will be able to see the sign-up welcome page.

Innovative Solutions Corporation

Employee Login

Login

[New Employee? Click here](#)

Traceability Matrix

System Requirements

No.	Priority Weight (1-5, 1: lowest, 5: highest)	Description
REQ-1	5	The user can select the team that they would like to join.
REQ-2	5	The user can select the team position that they would like to have.
REQ-3	5	The system can send a notification to the user if the team is full, and stop the user from joining said team.
REQ-4	5	The system can confirm the user has been added to the desired team.
REQ-5	5	The system can send a notification to the user if the desired position has already been taken, and stop them from taking it.
REQ-6	5	The system can allow users to view the teams, the players, and their positions.
REQ-7	5	The login will allow only an employee/owner of the company to log into and out of the sign-up system.

REQ-8	4	System admins can edit team information.
REQ-9	4	System admins can edit a player's information and remove them from the league.

Use Cases

No.	Description
UC1	Login: Login to user/admin account.
UC2	Logout: Logout of user/admin account.
UC3	Sign-up for Team: User selects the team they want to join.
UC4	Select Position: User selects the position they want on the team.
UC5	View Teams: Able to view teams, positions, players on team.
UC6	Edit Player Info: Admin can edit a player's information.
UC7	Edit Team Info: Admin can edit a team's information.
UC8	Remove player: Admin can remove player from the league.
UC9	View Player Details: Users can view their own profile details, like username and email.
UC10	Show Full: System shows what teams are already full and what positions are already taken.

Traceability Matrix

Req't	PW	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10
Req 1	5			x							
Req 2	5				x						
Req 3	5									x	
Req 4	5									x	
Req 5	5									x	
Req 6	5					x				x	
Req 7	5	x	x								
Req 8	4							x			
Req 9	4						x		x		
Max PW		5	5	5	5	5	4	4	4	5	5
Total PW		5	5	5	5	5	4	4	4	5	15

System Architecture and System Design

Architectural Style:

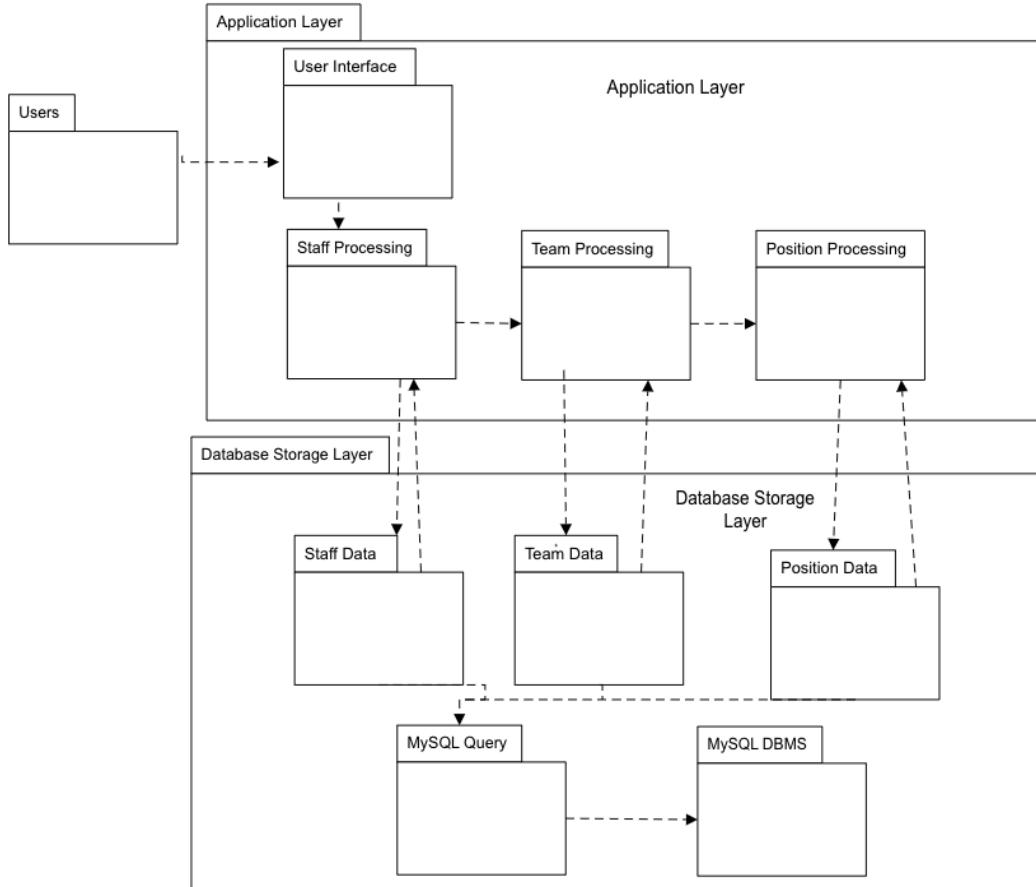
The Baseball League Registration System is a client/server system. It is a two-tiered design where the user interface is accessed via the computer at the company running the system, and the data is stored on the server or database. The user submits a request through the website to the server or database to query for information and retrieve data. The DBMS sends back data to the user. This is processed through MAMP/XAMPP with Apache, MySQL, and PHP. Apache acts as the local host through which the system pages are shown, giving the client requests to its attached PHP translator. MySQL acts as our storage to hold the employees of the company and the users currently registered in the league, including their team and position.

Identifying Subsystems:

The below diagram represents the subsystems of the Baseball League Registration System. The subsystem consists of two layers: the application layer, or user interface, and the database storage layer. The application layer depicts the processing packages for staff, team, and position processing. The packages in the application layer interact with the corresponding database storage processing packages in the database storage layer. The user accesses the data through the user interface of the application layer by clicking and entering data into forms. The database storage is accessed, MySQL queries the information, and the DBMS returns the requested data to the user interface. The DBMS acts as the server in this client/server architectural style.

(See UML Package Diagram on the next page.)

UML Package Diagram:



Persistent Data Storage:

The persistent objects in the Baseball League Registration System are username and password, player name, team, and position. We used MySQL as our database management system.

Global Control Flow:

Execution Orders

The Baseball League Registration System is event-driven, meaning the system waits on the user to tell it what to do, mainly through button clicks. These events do different things based on what button the user clicks, like launching an SQL query that adds the user to the company's database. Most of the buttons are displayed on the main page the user sees once they get through the login screen. They involve registering for teams and positions, but buttons also exist to view currently existing teams, view profile details of the currently logged-in user, and the ability to log out. Generally, the user logs in first and then can choose the order they would like to go in.

Time Dependency

The Baseball League Registration System does not have timers. Though it does depend on information being sent to and from a database, the system is designed to run on only one computer at a time in the company. This ensures data accuracy and that the system isn't bogged down with multiple users trying to select the same team and position at the exact same time. In other words, only one computer at the company runs the system, so the system isn't worried about real-time.

Hardware Requirements:

The Baseball League Registration System will need XAMPP/MAMP software and a screen display.

- **Color display:** Min resolution: 640 x 480 pixels | Max resolution: 1024 x 1920
- **Computer:** Desktop Computer or any device with a compatible web browser
- **Memory:** Min 8 GB RAM, Max 64 GB RAM
- **Hard Drive:** Min 5 GB hard drive space

User Interface Design and Implementation:

The user interface was implemented with buttons, check boxes and forms. To make the program as user friendly as possible, all screens were designed to involve as few clicks as possible, and nothing needs to be typed beyond the login/registration screen. Each form area has clear directions written for the user to read and perform.

Because our team was ahead in terms of the design of our program, much of our interface had already been designed when module 7 was released. Therefore, many of the screens below will look very similar to the screens presented in the “User Interface Specifications” assignment given in module 7.

Login Page:

The login page features a light blue header with the text "Innovative Solutions Corporation" and "Employee Login". Below this is a white rectangular input field labeled "Login". Inside the field are two input boxes: one for "Employee Name(First Last)" and another for "Password". A blue "Login" button is centered below the boxes. At the bottom of the input field, there is a link "New Employee? Click here".

Registration Page:

The registration page features a light blue header with the text "Innovative Solutions Corporation" and "New Employee Registration". Below this is a white rectangular input field labeled "Create Employee Account". Inside the field are four input boxes: "Employee Name(First Last)", "Enter Employee Email", "Enter Password", and "Confirm Password". A blue "Create Employee Account" button is centered below the boxes.

Homepage (User can select a position and team from here):

Admin Page:

Database Admin Page

Add New Data Logout

Players/Teams DataBase

Name	Team	Position	Action
Zach Butler	Cougars	Left Field	Edit Delete
Cat Kramka	Cubs	Second Base	Edit Delete
Rhett Godwin	Cougars	Short Stop	Edit Delete
Rhett godwin	Jaguars	Left Field	Edit Delete
Cat Kramka	Cougars	Third Base	Edit Delete

Employee Accounts Database

Add New Data

Name	Password	Email	Action
Rhett Godwin	test	Rhett@test.com	Edit Delete
Zach Butler	test	zach@test.com	Edit Delete
Cat Kramka	test	Cat@test.com	Edit Delete
admin	admin	admin@test.com	Edit Delete

Update Data Page:

Add Data
MUST UPDATE ALL FIELDS

Home

Player Name	<input type="text"/>
Team	<input type="text"/>
Position	<input type="text"/>
	<input type="button" value="Add"/>

Confirmation page:

Design of Tests:

Test design implementation

Test 1: Logging in using incorrect login info.

This test will ensure that the login system prevents users from logging in without entering the proper login information.

Test 2: Logging in using correct login info.

This test will ensure that users who log in using the correct login information will be let into the system so that they can sign up for the baseball league.

Test 3: Attempt to register as an employee already in the database.

This test will be performed to ensure users already in the system can't accidentally create a duplicate account with the company.

Test 4: Register as a new employee who needs to be added to the database.

This test will ensure that users who are new to the company can create an account and sign up for the baseball league.

Test 5: Sign up for a team and position currently filled.

This test will be performed to verify the system will notify and not allow a user to sign up for a currently filled position on a team.

Test 6: Sign up for an open team and position.

This test will verify that the system allows a user to sign up for a currently open position on a team.

Test 7: Delete a user from the league using the admin account.

This test will prove the admin can delete players from the league if needed (Proof of deletion from the database will be shown).

Test 8 : Update a player's info.

This test will verify the admin can update and modify current players' data. (The user profile page will show if an update happened.)

Test 9 : Show a list of players for a team.

This test will verify that each team's roster is displayed by pressing the team button. (If the roster appears, it works).

The tests listed above will cover the functionality of our program and prove that it accomplishes what it was designed for.

Project Plan

Weeks 1- 2: Setup the development environment and become familiar with project requirements. Establish the framework and architecture of the system, construct databases, and wireframe an initial UI design for the page.
(COMPLETE)

Weeks 3-6: Development of the system. Code and implement a login system. Code and implement the Home page. Code and implement the backend.
(COMPLETE)

Weeks 7-8: Test and Deployment: Rigorous testing, debugging and final testing.
(COMPLETE)

Weeks 9-11: Review and improve any implemented features based on customers' feedback. **(COMPLETE)**