COS 397: Computer Science Capstone I

Critical Design Review Document

UMaine Athletic Department Inventory Management System

Version 0.1

Collin Rodrigue, Brennan Poitras, Graham Bridges, Gabe Poulin, Sean Radel

14 December 2023

Table of Contents

1. Introduction

The purpose of the product is to fulfill the customer needs of an inventory management solution. Our system will replace the customer's previous solutions for managing their equipment. The customer previously used Front Rush and currently uses a combination of Excel spreadsheets and word of mouth to track inventory. Our product will allow the customer to organize their inventory by associating equipment with teams and players. The system will be designed with simplicity in mind so that they do not need experienced developers to maintain their product following the delivery date. Equipment, player accounts, and teams will be able to be made on demand to allow the system to scale to the customer's needs. Our customer is the University of Maine Athletic Department, and specifically Jude Killy, Nick Fox, and Kevin Ritz.

1.1 Purpose of This Document

The purpose of this document is to review the design of the equipment inventory system for the UMaine athletic department. It will be read by the clients from the UMaine athletic staff for evaluation on whether or not the project is ready to begin development. It will assess the design of the system as specified by previous document artifacts created by IMSG; this includes the System Requirements Specifications, the System Design Document, and the User Interface Design Document. It will review the Problem Statement, Requirements, System Architecture, User Interface, DevOps, Milestones, and Feedback for the equipment inventory project.

1.2 References

1. Google Sheets. *Athletic Inventory*. University of Maine Athletic Department, (non-acessable to public), 15 Nov. 2023

2. IMSG. “System Requirements Specification” November 1 2023, https://docs.google.com/document/d/1LnOj2DEyu8DPbKXBTDBm2y6UbePr_AXC/edit

3. IMSG. “System Design Document” November 15 2023, https://docs.google.com/document/d/1_dkbb3zXQzOxVbdIr3nbizQc0r-f6yC2/edit?usp=sharing&ouid=105664753662702248466&rtpof=true&sd=true

4. IMSG. “User Interface Design Document” December 4 2023, https://docs.google.com/document/d/1b77rhypShsIfxY-nvZ0mERQRSm4YoxEq/edit?usp=sharing&ouid=105664753662702248466&rtpof=true&sd=true

2. Problem Statement

Through our initial meetings with our client, we were able to identify a few pressing issues that our client has with their current solution to inventory management. Identifying the main issues, and deriving solutions to those problems is extremely important, and one of the first steps taken to ensure a good product is presented to our client.

2.1 Identified Problems

The Athletic Department for the University of Maine has been currently using a Google Spreadsheet to keep track of their massive collection of inventory across all sports groups. This is clearly an issue, as the security, efficiency and usability of Google Spreadsheets for such a large operation is less than ideal. The main issues being efficiency, security, user-friendliness and customization. These problems are the issues that IMSG has set out to fix for the betterment of the University of Maine Athletic Department.

2.2 Proposed Solution

There have been attempts to use third-party software to better manage the inventory in the past, but all attempts have failed due to lack of a user-intuitive interface and lack of customization. IMSG has created the bones and outlines of an inventory management system, using the Google Cloud Platform for their security benefits, that allows the inventory team at the University of Maine to manage their inventory more efficiently, easily and freely. This solution will be scalable to the needs of a growing inventory, as well as increase the efficiency of everyday operations.

3. Requirements

3.1.1 Functional Requirements

Functional requirements define the features and use cases that the system possesses to meet customer needs. The functional requirements describe how users can interact with the system and what the expected preconditions and postconditions will be. Furthermore, we define test cases to verify that our system meets requirements. Our top level use case diagram is shown here:

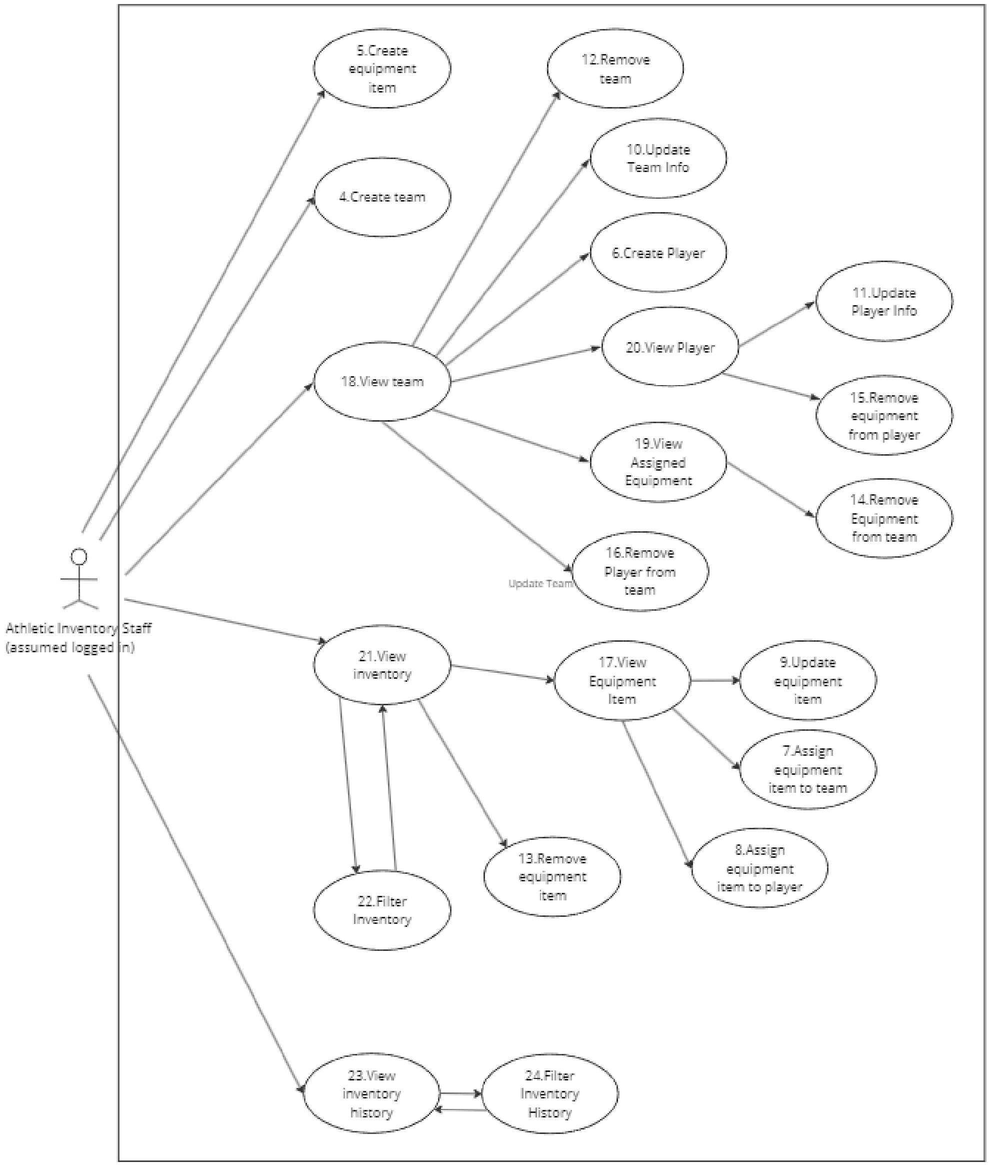

5.Create equipment item
12.Remove team
10.Update Team Info
4.Create team
6.Create Player
11.Update Player Info
20.View Player
18.View team
15.Remove equipment from player
19.View Assigned Equipment
14.Remove Equipment from team
16.Remove Player from team
Update Team
Athletic Inventory Staff (assumed logged in)
21.View inventory
17.View Equipment Item
9.Update equipment item
7.Assign equipment item to team
8.Assign equipment item to player
13.Remove equipment item
22.Filter Inventory
23.View inventory history
24.Filter Inventory History

Figure 3.1 - Top Level Use Case Diagram

Our functional requirements can be broken down into three categories: account management, CRUD operations, and inventory management.

Account Management:

These requirements are related to account management including registering, logging in, and logging out.

FR1: The system shall allow a user to register an account.
FR2: The system shall allow a user to log in.
FR3: The system shall allow a user to log out.

Create, Retrieve, Update, Delete Operations (CRUD):

This is the manner in which data will flow in and out of our system. These will include things like inputting new inventory product, viewing a team's information and roster, and updating player information. All CRUD operations will apply to all the objects in our system including players, teams, and sports equipment.

FR5: The system shall allow a user to add a new equipment item.
FR11: The system shall allow a user to update player information.
FR18: The system shall allow a user to view a a team's roster.

Inventory Management:

These requirements encapsulate the core functionalities of an inventory management system with additional functionality specific to our client's needs. Users will have the ability to assign equipment to players and teams, allowing for easy tracking of product. The system will also allow users to view all the product currently in inventory and a history of all previous inventory transactions and assignments.

FR7: The system shall allow a user to assign an equipment item to a team.
FR8: The system shall allow a user to assign an equipment item to a player.
FR21: The system shall allow a user to view general inventory.
FR22: The system shall allow a user to filter the general inventory.
FR23: The system shall allow a user to view the inventory history.
FR24: The system shall allow a user to filter the inventory history.

3.1.2 Testing Functional Requirements

Since our application is data-oriented and record-based, a majority of the tests involve querying our database to ensure Create, Read, Update, and Delete (CRUD) operations worked successfully. Our tests involve defining an expected outcome for each functional requirement.

FR5: The system shall allow a user to create a new equipment item.

Expected Outcome: The equipment table in our database contains a new record matching the created equipment object.

This is an example of a functional requirement and its expected outcome. Our system must do two things to ensure this expected outcome is met.

- Verify the insert operation was successful

- Retrieve the new record and verify the data matches the input

If these two conditions are met then our test will pass. These tests will apply to all CRUD operations.

3.2 Non-Functional Requirements

Non-functional requirements will be covered in this section. It will provide the list of each NFR, go over the aspects that each NFR covers, and how we plan to use test cases to ensure that each NFR is met. Below is the system's NFRs:

NFR-001	**Priority:** 3
Description: The system must provide data security for athlete and inventory information via encryption at rest and in transit.	
Test Case: Query the data without encryption keys to ensure the data is encrypted	

NFR-002	**Priority:** 3
Description: The system should ensure a response time of under 3 seconds for the inventory item filter.	
Test Case: By running a repeated access attempt on different items and filters, we should be able to graph the response time of each access attempt.	

NFR-003	**Priority:** 5
Description: The system must be available for use 99.9% of the time.	
Test Case: Having a downtime detector will allow for accurate graphing of real-time downage.	

NFR-004	**Priority:** 5
Description: The system should be compatible with all major web browsers using Chromium 80.0.3987.33 and later.	
Test Case: Running the application on major browsers such as Edge, Chrome, and OperaGX will show compatibility with other browsers running on Chromium.	

NFR-005	**Priority:** 5
Description: The system should support at least 10 concurrent users.	
Test Case: By using third-party software, we can write scripts to have multiple ghost users sign into the application and use it at will, testing a number higher than 10 is ideal.	

NFR-006	**Priority:** 5
Description: The system should not have any major vulnerabilities.	
Test Case: We will conduct a penetration test of the service that scans the system dependencies for CVEs, checks for XSS attack capabilities, and validates there are no unsanitized data inputs. The team will leverage the MITRE ATT&CK matrix to track what additional TTPs we need to address.	

NFR-007	**Priority:** 5
Description: The system should have user authentication through a “maine.edu” email address and password combination.	
Test Case: Testing accounts not affiliated with the university to make sure they cannot get access is ideal, as well as making sure the “maine.edu” accounts sync with the UMaine database for password syncing.	

NFR-008	**Priority:** 3
Description: The system must have a responsive and user-friendly interface, ensuring that user actions result in fast visual feedback.	
Test Case: Having continuous tests with end users to get feedback and improve until the desired product is achieved.	

NFR-009	**Priority:** 5

Description: The system must allow for the addition of new items and users.
Test Case: General testing for adding items and users through the user interface, as well as testing the overall speed at which you can add those items and users.

NFR-010	**Priority:** 4
Description: The system shouldn’t have an error rate adding or modifying data exceeding 5%.	
Test Case: Write scripts to modify data at mass and have the script quantify how many fail.	

NFR-011	**Priority:** 4
Description: The system should allow for multiple administrative users.	
Test Case: We will create multiple administrative users using scripts, then have those users do admin-level activities and see if there are any issues.	

NFR-012	**Priority:** 5
Description: The system should be able to be maintained by administrators through the user interface.	
Test Case: Having an end user quality test our product as an admin would be best here.	

NFR-013	**Priority:** 4
Description: The system should be able to handle an increasing number of inventory items without any performance degradation.	
Test Case: By writing scripts to add items constantly, we can see the time it takes to create those items and the time it takes to access as more and more are added.	

NFR-014	**Priority:** 2
Description: The system should be maintained in the most cost-effective manner.	
Test Case: By remaining within the budget we are allotted.	

NFR-015	**Priority:** 3
Description: The system shall be easily configurable by user administrators.	

Test Case: End user testing.

NFR-016	**Priority:** 1
Description: The system should provide accurate data analytics with timetables.	
Test Case: By having both the timestamps of items created and/or modified, we can compare the timestamps and tables in the graphs to the actual data using scripts.	

These non-functional requirements cover four aspects of our system:

System Availability

- NFR-003: The system must be available for use 99.9% of the time
- NFR-004: The system should be compatible with all major web browsers using Chromium 80.0.3987.33 and later
- NFR-005: The system should support at least 10 concurrent users
- NFR-011: The system should allow for multiple administrative users

System Efficiency

- NFR-002: The system should ensure a response time of under 3 seconds for the inventory item filter
- NFR-009: The system must allow for the addition of new items and users
- NFR-010: The system shouldn't have an error rate adding or modifying data exceeding 5%
- NFR-013: The system should be able to handle an increasing number of inventory items without any performance degradation
- NFR-014: The system should be maintained in the most cost-effective manner

System Security

- NFR-001: The system must provide data security for athlete and inventory information via encryption at rest and in transit

- NFR-006: The system should not have any major vulnerabilities

- NFR-007: The system should have user authentication through a "maine.edu" email address and password combination

System Usability

- NFR-008: The system must have a responsive and user-friendly interface, ensuring that user actions result in fast visual feedback

- NFR-012: The system should be able to be maintained by administrators through the user interface

- NFR-015: The system shall be easily configurable by user administrators

- NFR-016: The system should provide accurate data analytics with timetables

The test cases for each of our tests shall be thorough and accurate to ensure the availability, efficiency, security and usability of our system. To test availability we plan to implement a downtime detector, test how the system runs on various browsers, and create a script that will add mock users onto the system consecutively. For efficiency we will test how the system responds to repeated requests to view various items and filters in a fast manner, time how fast it takes to add objects to the database when given multiple requests, and record errors and time when mass data is modified and/or entered in the database. Security will be tested by querying for information without encryption, conducting penetration tests looking for CVEs, and trying to access the system with a non-maine.edu email. Usability will be tested by using end users that will give feedback.

4. System Architecture

The system architecture abstractly defines the design, structure, and software choices of our system. In this section we describe the workflow of user actions and at which part of the system they are handled by. Next, we describe the database design and schema.

4.1 System Design

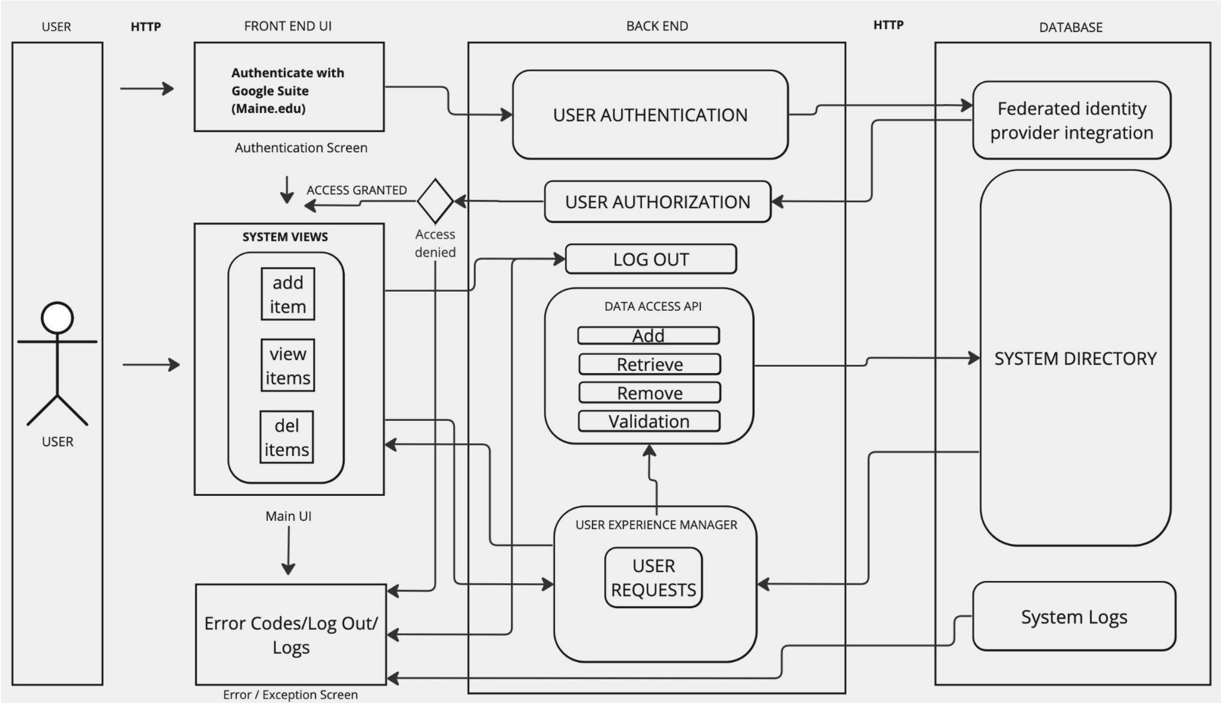


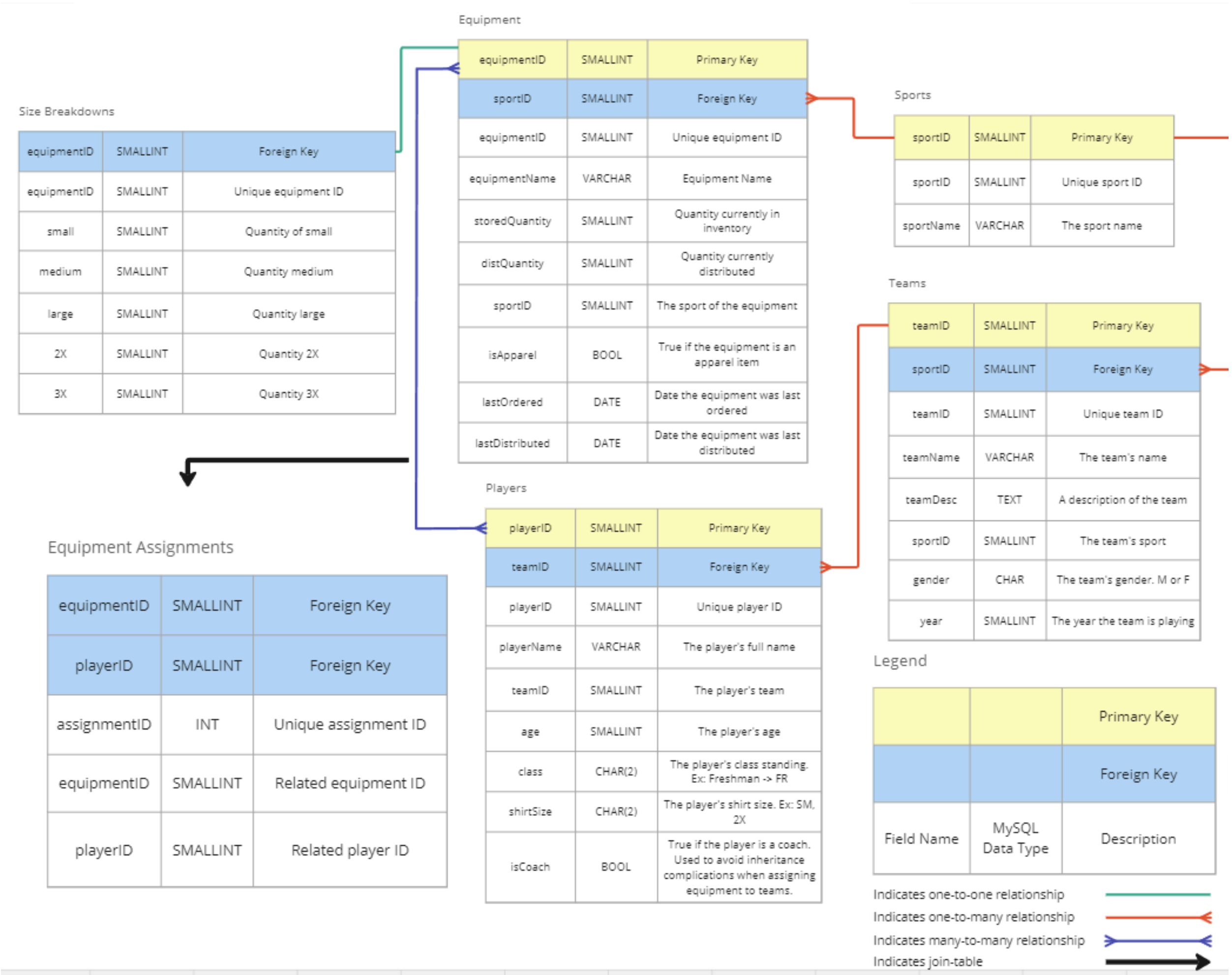
Figure 4.1 - System Design

In our system's implementation, the user's will first be prompted to login with their maine.edu email. The authenticated user will then be able to create, read, update, and delete items from the database. The user interface will be written using React, and we will use NextJS as our backend. Our backend is where we can create reports, calculate analytics, and make the API calls to google cloud platform. We will be using Google Cloud Platform to host our database as they have a range of database types and we can easily implement the Google login solution.

4.2 Database Design

The database used by the system will be a MySQL database. We will be a relational database because it will allow us to establish relationships between all the objects in our system. The record-based, relational nature of the database will allow us to fulfill functional requirements related to equipment tracking and assignment. This will also promote strong data integrity and avoid data redundancy. Following the MVC design pattern, the database serves as the Model, effectively mapping out the data flow of our application.

The database achieves this through the use of primary and foreign keys. It will allow users to establish one-to-one, one-to-many, or many-to-many relationships between Players, Teams, Sports, and Equipment. The database will contain indexes on fields that are a primary or foreign key, allowing faster data retrieval for commonly used queries. The database schema is shown below using an entity-relationship diagram.



Equipment

equipmentID	SMALLINT	Primary Key
sportID	SMALLINT	Foreign Key
equipmentID	SMALLINT	Unique equipment ID
equipmentName	VARCHAR	Equipment Name
storedQuantity	SMALLINT	Quantity currently in inventory
distQuantity	SMALLINT	Quantity currently distributed
sportID	SMALLINT	The sport of the equipment
isApparel	BOOL	True if the equipment is an apparel item
lastOrdered	DATE	Date the equipment was last ordered
lastDistributed	DATE	Date the equipment was last distributed

Sports

sportID	SMALLINT	Primary Key
sportID	SMALLINT	Unique sport ID
sportName	VARCHAR	The sport name

Size Breakdowns

equipmentID	SMALLINT	Foreign Key
equipmentID	SMALLINT	Unique equipment ID
small	SMALLINT	Quantity of small
medium	SMALLINT	Quantity medium
large	SMALLINT	Quantity large
2X	SMALLINT	Quantity 2X
3X	SMALLINT	Quantity 3X

Teams

teamID	SMALLINT	Primary Key
sportID	SMALLINT	Foreign Key
teamID	SMALLINT	Unique team ID
teamName	VARCHAR	The team's name
teamDesc	TEXT	A description of the team
sportID	SMALLINT	The team's sport
gender	CHAR	The team's gender. M or F
year	SMALLINT	The year the team is playing

Players

playerID	SMALLINT	Primary Key
teamID	SMALLINT	Foreign Key
playerID	SMALLINT	Unique player ID
playerName	VARCHAR	The player's full name
teamID	SMALLINT	The player's team
age	SMALLINT	The player's age
class	CHAR(2)	The player's class standing. Ex: Freshman -> FR
shirtSize	CHAR(2)	The player's shirt size. Ex: SM, 2X
isCoach	BOOL	True if the player is a coach. Used to avoid inheritance complications when assigning equipment to teams.

Equipment Assignments

equipmentID	SMALLINT	Foreign Key
playerID	SMALLINT	Foreign Key
assignmentID	INT	Unique assignment ID
equipmentID	SMALLINT	Related equipment ID
playerID	SMALLINT	Related player ID

Legend

		Primary Key
		Foreign Key
Field Name	MySQL Data Type	Description

Indicates one-to-one relationship

Indicates one-to-many relationship

Indicates many-to-many relationship

Indicates join-table

Figure 4.2 - Database Schema

5. User Interface Design

In this section we will give a walkthrough of how the user interface will appear and go over the general user flow from the main pages. Below is a chart of every page and how the user can generally move between them:

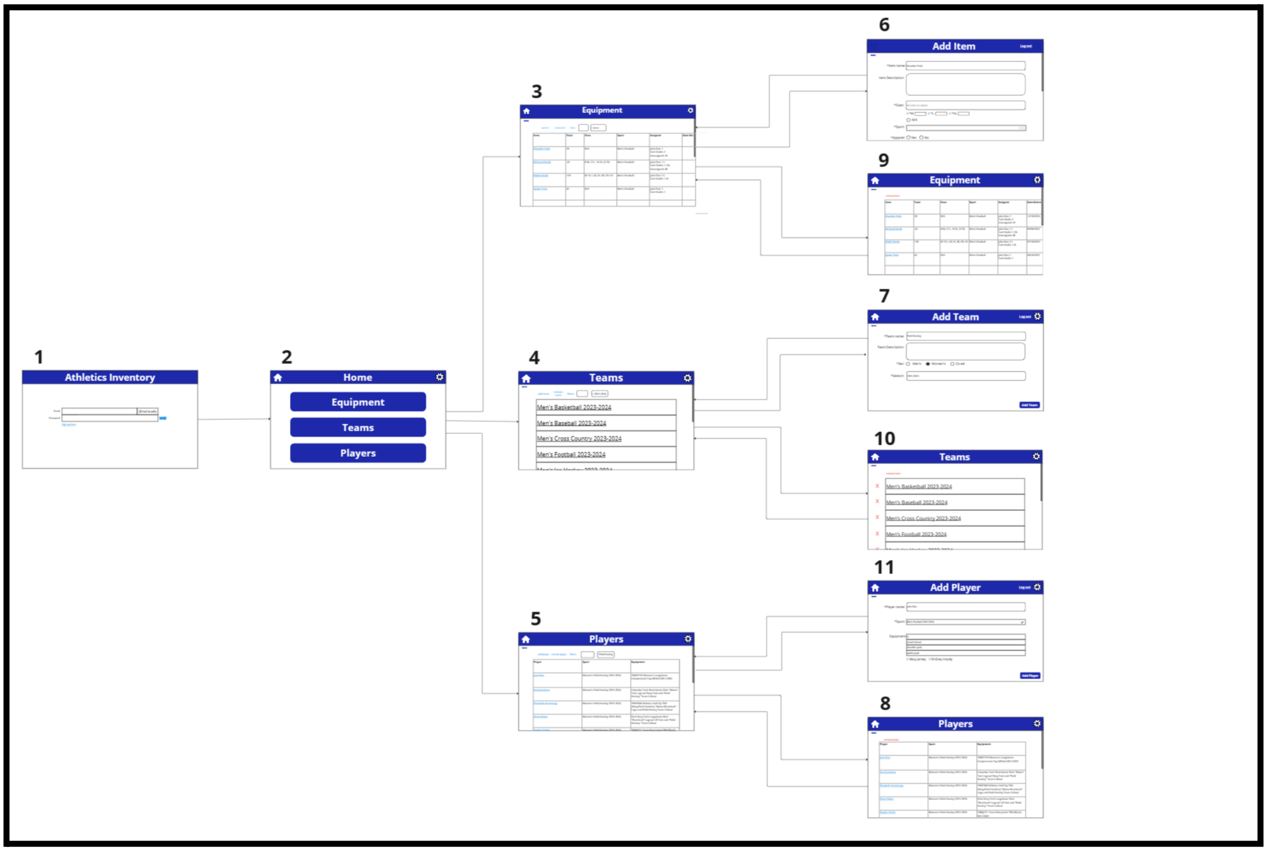


Figure 5. Navigation of Inventory System

The user must first log in to the system using their maine.edu email and password:

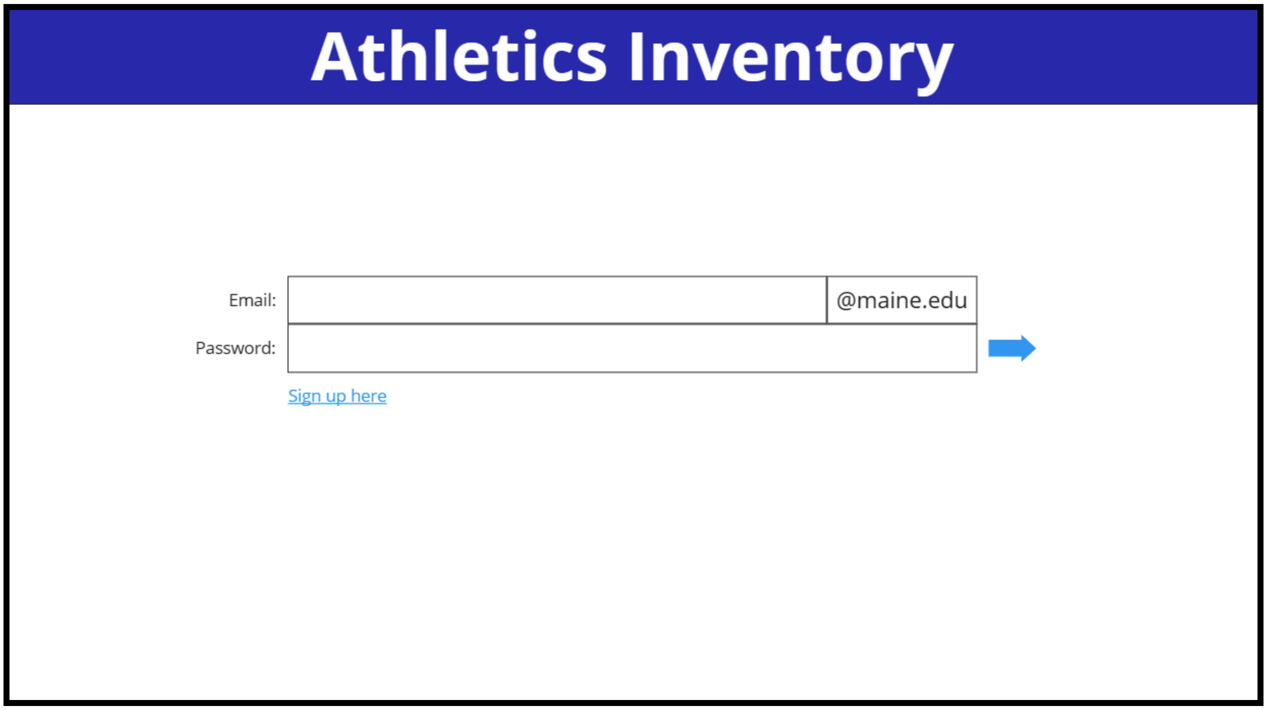


Figure 5.1. Login Screen

The user will be presented with the home page once they're logged in:

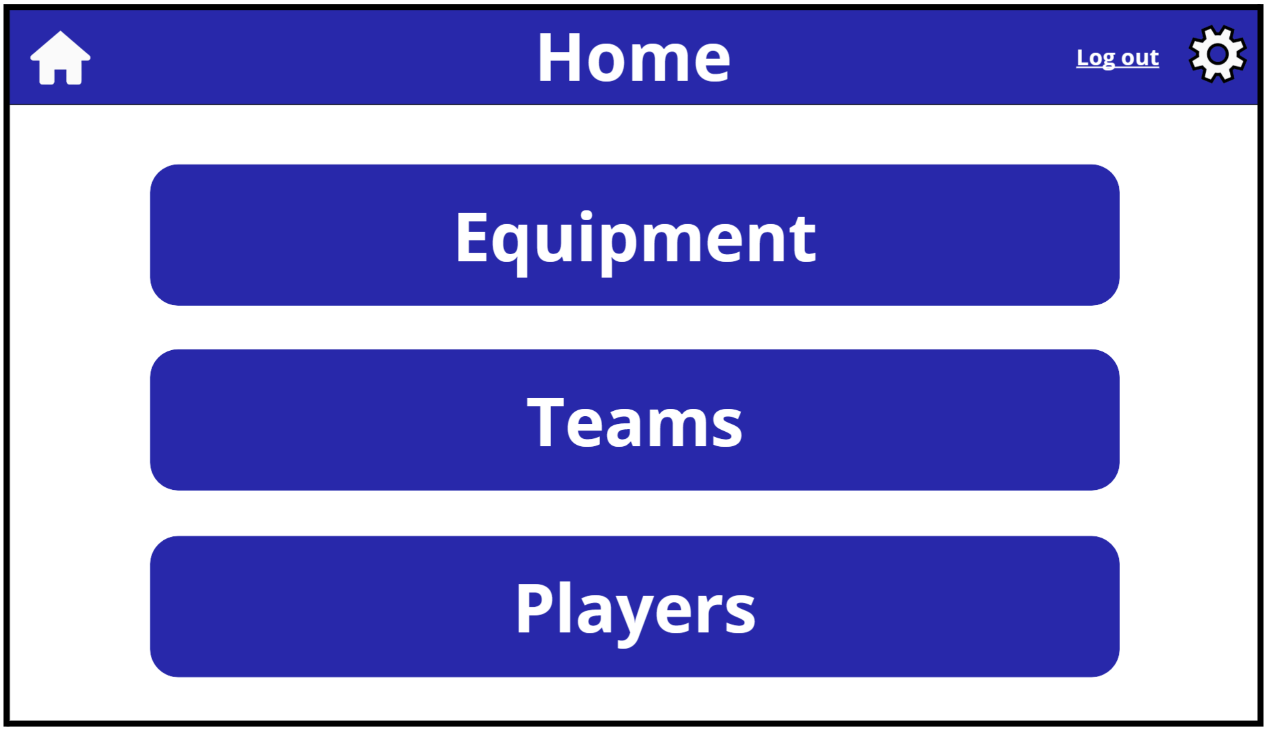
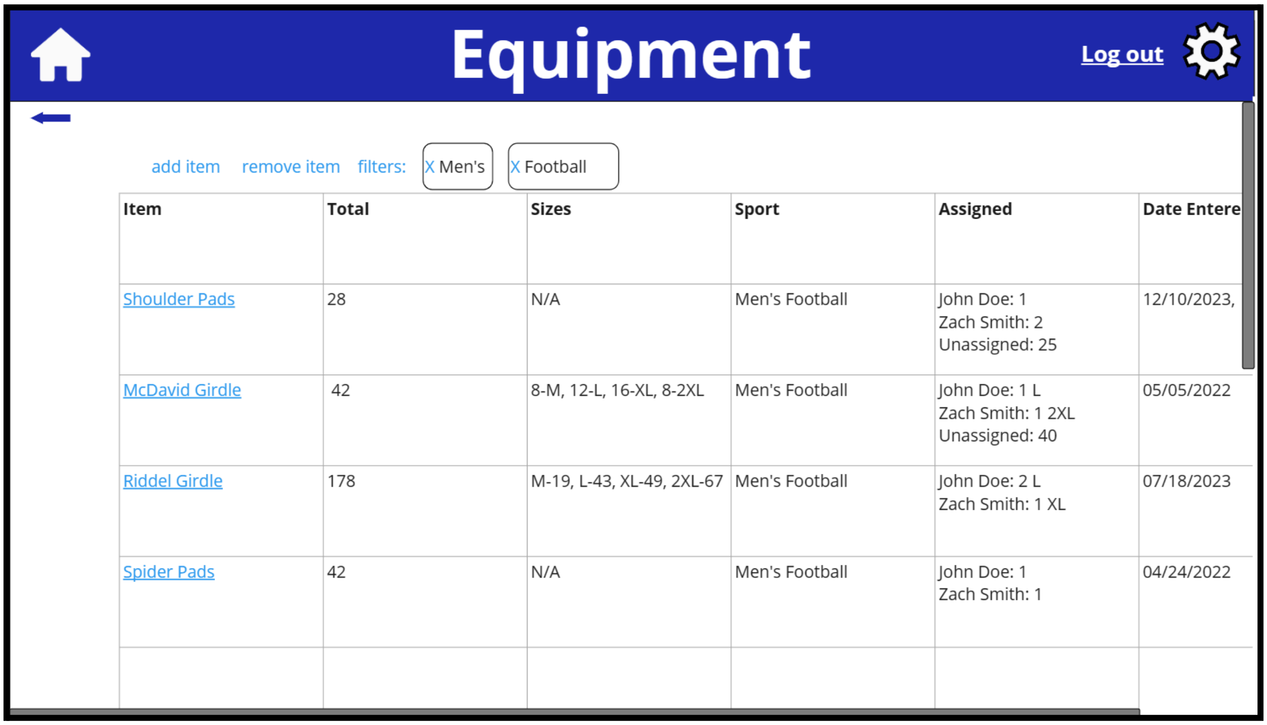


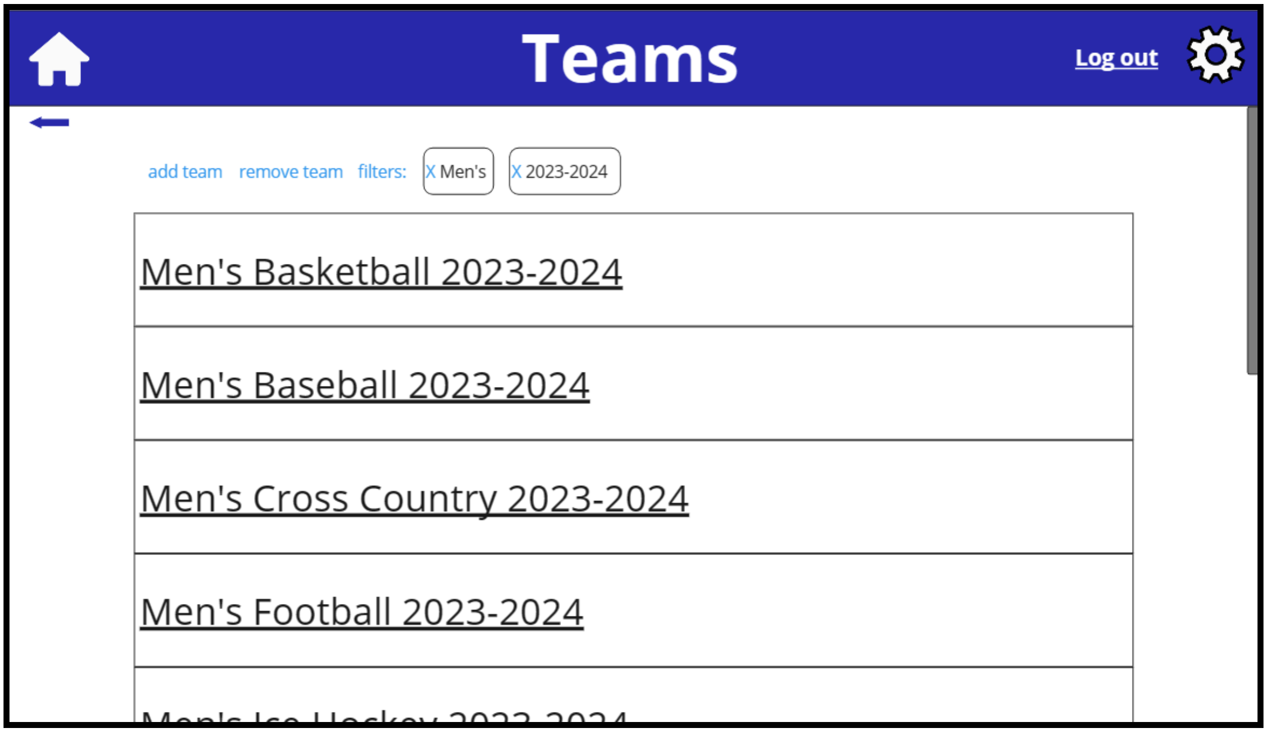
Figure 5.2. Home Screen

The navigation bar at the top of the screen will be present on every other page. From left to right, it includes a home logo that will bring the user back to the home page, a title that tells the user what page they're currently on, a logout button that will open a pop up confirming if the user wants to log out, and a gear cog button that will bring the user to the settings page. On the home screen the user can decide to go to either the equipment, teams, or players page by clicking on the corresponding blue button. These three screens are depicted below in Figures 5.3, 5.4, and 5.5:



Item	Total	Sizes	Sport	Assigned	Date Entere
Shoulder Pads	28	N/A	Men's Football	John Doe: 1 Zach Smith: 2 Unassigned: 25	12/10/2023,
McDavid Girdle	42	8-M, 12-L, 16-XL, 8-2XL	Men's Football	John Doe: 1 L Zach Smith: 1 2XL Unassigned: 40	05/05/2022
Riddel Girdle	178	M-19, L-43, XL-49, 2XL-67	Men's Football	John Doe: 2 L Zach Smith: 1 XL	07/18/2023
Spider Pads	42	N/A	Men's Football	John Doe: 1 Zach Smith: 1	04/24/2022

Figure 5.3. Equipment View Screen



Men's Basketball 2023-2024

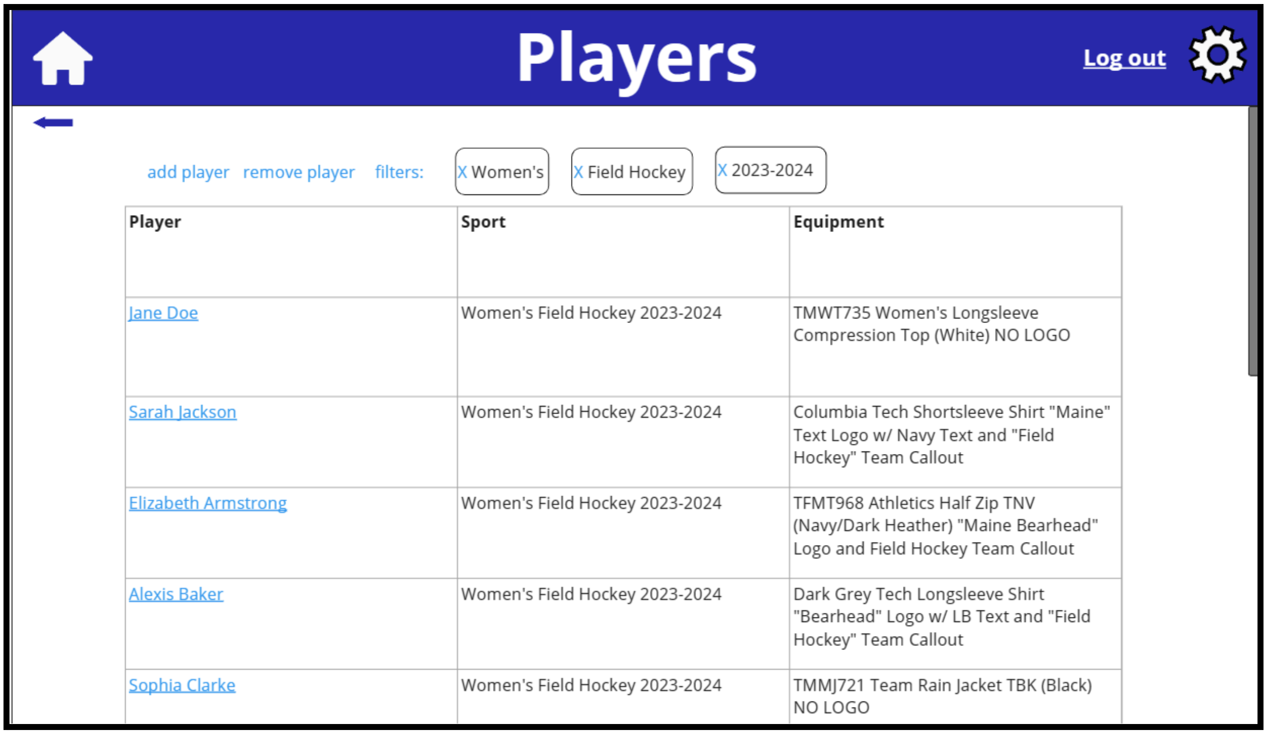
Men's Baseball 2023-2024

Men's Cross Country 2023-2024

Men's Football 2023-2024

Men's Ice Hockey 2023-2024

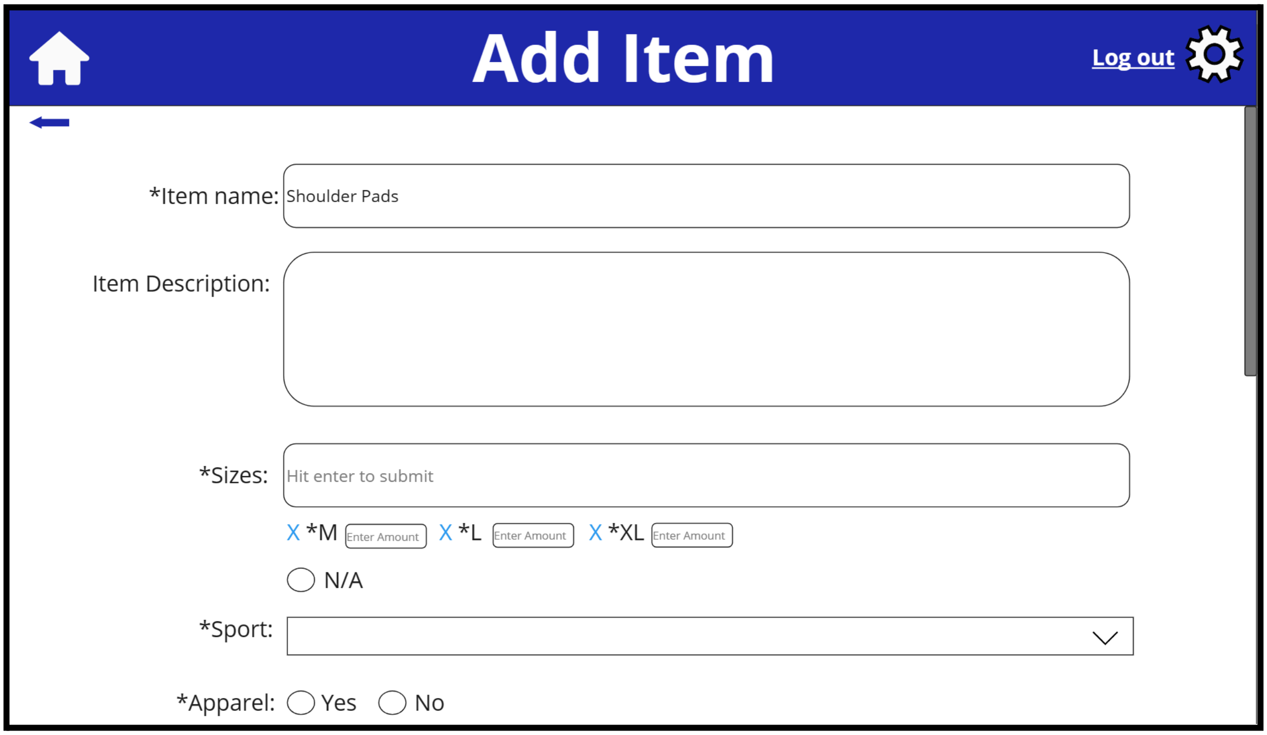
Figure 5.4. Team View Screen



Player	Sport	Equipment
Jane Doe	Women's Field Hockey 2023-2024	TMWT735 Women's Longsleeve Compression Top (White) NO LOGO
Sarah Jackson	Women's Field Hockey 2023-2024	Columbia Tech Shortsleeve Shirt "Maine" Text Logo w/ Navy Text and "Field Hockey" Team Callout
Elizabeth Armstrong	Women's Field Hockey 2023-2024	TFMT968 Athletics Half Zip TNV (Navy/Dark Heather) "Maine Bearhead" Logo and Field Hockey Team Callout
Alexis Baker	Women's Field Hockey 2023-2024	Dark Grey Tech Longsleeve Shirt "Bearhead" Logo w/ LB Text and "Field Hockey" Team Callout
Sophia Clarke	Women's Field Hockey 2023-2024	TMMJ721 Team Rain Jacket TBK (Black) NO LOGO

Figure 5.5. Player View Screen

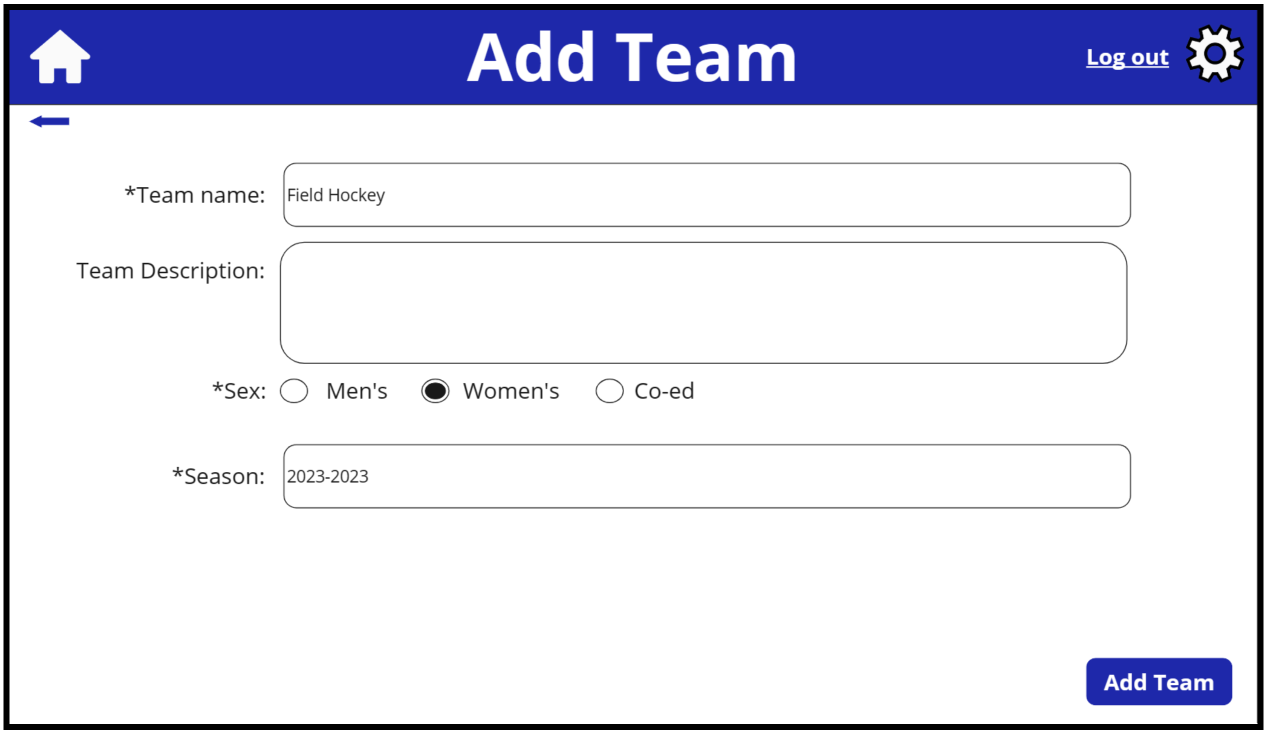
Each page will display objects in the database in the given category. The user will be directed to pages for adding or removing objects to the database by clicking on the blue add and remove buttons(“add item”, “remove item”). The user will be able to add filters to the list by clicking on the filters button. If they want to remove a filter they can click the blue “X” next to the designated filter. If the user wants to edit a specific object, such as “Jane Doe” for players, they can click on the name. The next three figures depict the pages where users can add equipment, teams, and players respectively:



*Item name: Shoulder Pads

*Sizes:

Figure 5.6. Add Equipment Screen



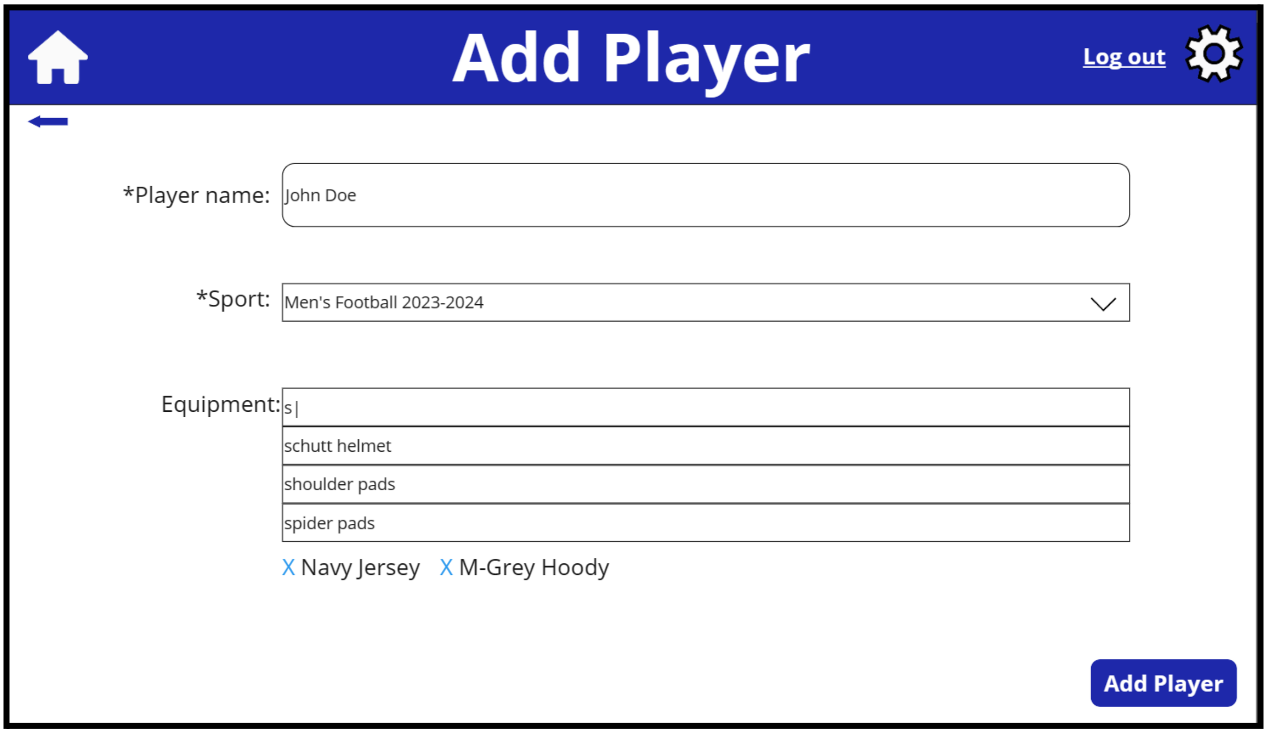
*Team name: Field Hockey

Team Description:

*Sex: ○ Men's ● Women's ○ Co-ed

*Season: 2023-2023

Figure 5.7. Add Team Screen



*Player name: John Doe

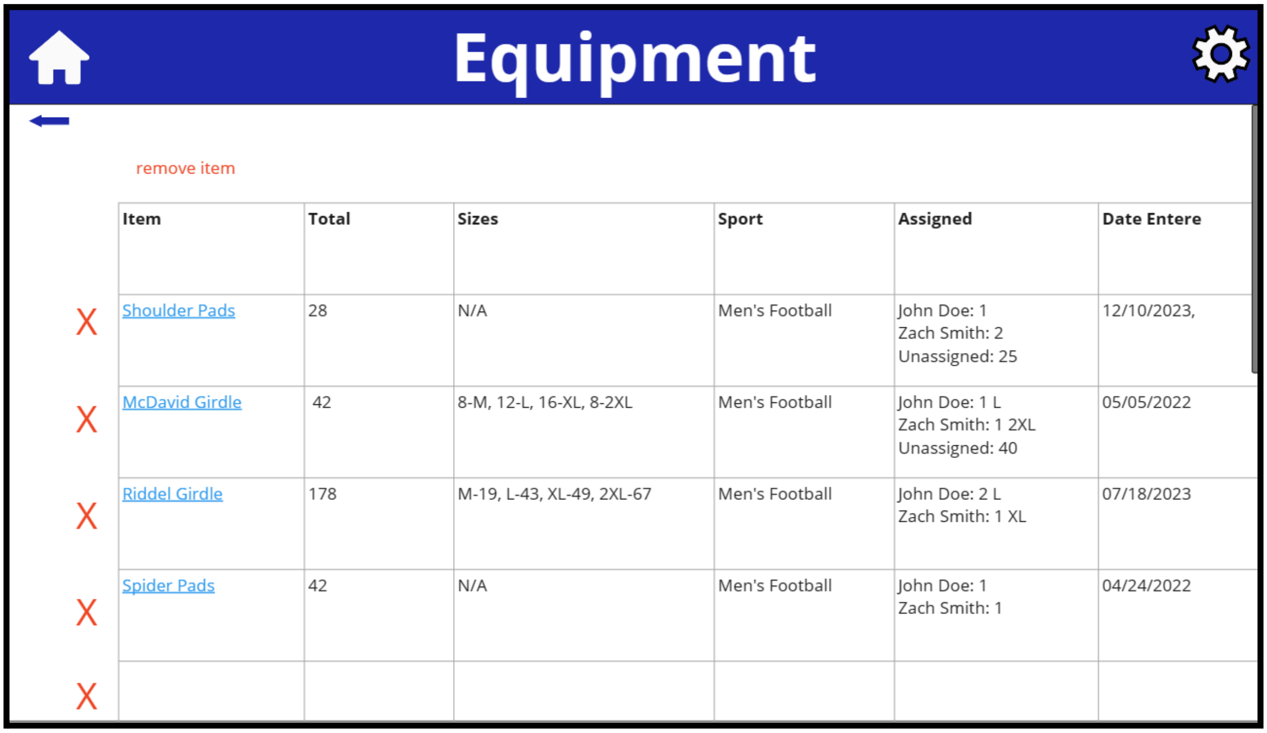
*Sport: Men's Football 2023-2024

Equipment: s|
schutt helmet
shoulder pads
spider pads

X Navy Jersey X M-Grey Hoody

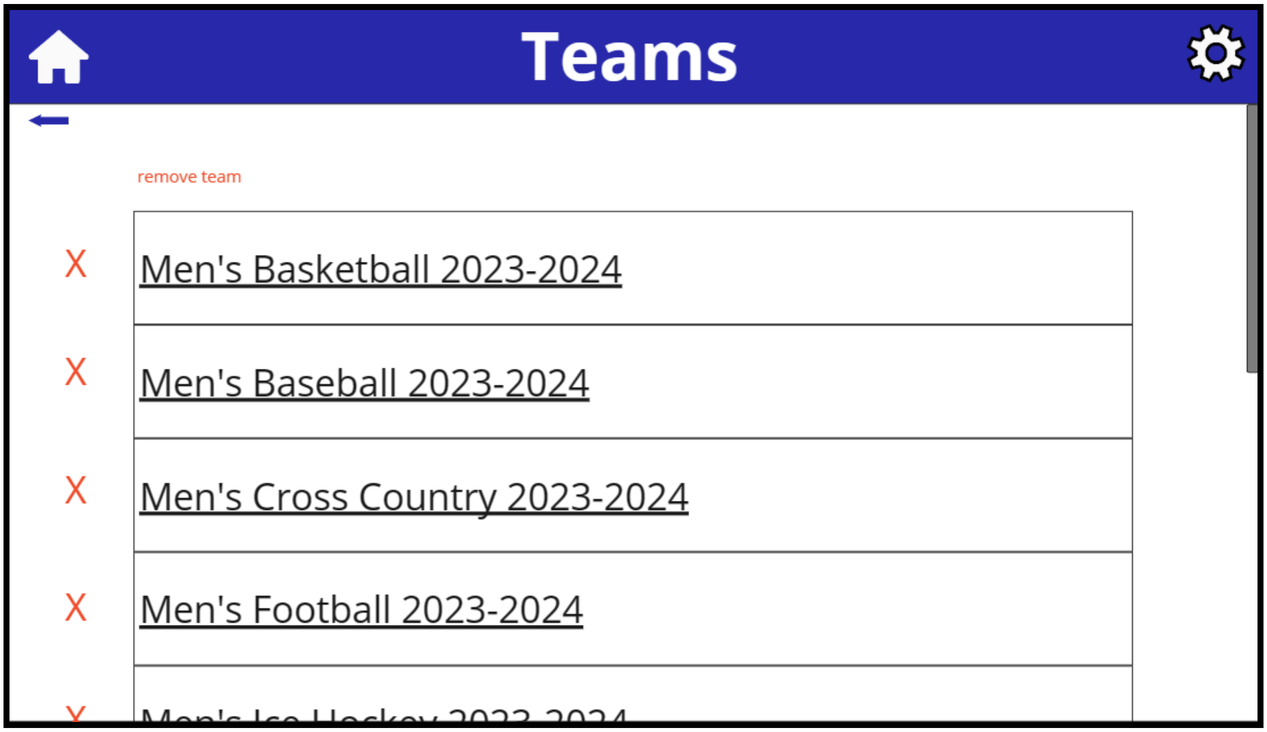
Figure 5.8. Add Player Screen

On each of these screens, the user will fill out each field for the given object. An asterisk next to a field indicates a required field. For equipment, users will give a name, an optional description, the sizes if applicable, the sport the equipment is used for, and whether or not the equipment is an apparel item. For teams, the user will give a name, optional team description, the sex of the sport, and the season. For players, the user will give a player name, the sport they play, and the equipment they are using. Once the user is done filling out the fields, they will click the blue "Add" button on the bottom right of the page to add the new object to the database. The addition of an object will be recorded in the database history. Users can always go back and edit the fields of an object. Next we will go over the pages used for removing objects from the database:



	Item	Total	Sizes	Sport	Assigned	Date Entere
X	Shoulder Pads	28	N/A	Men's Football	John Doe: 1 Zach Smith: 2 Unassigned: 25	12/10/2023,
X	McDavid Girdle	42	8-M, 12-L, 16-XL, 8-2XL	Men's Football	John Doe: 1 L Zach Smith: 1 2XL Unassigned: 40	05/05/2022
X	Riddel Girdle	178	M-19, L-43, XL-49, 2XL-67	Men's Football	John Doe: 2 L Zach Smith: 1 XL	07/18/2023
X	Spider Pads	42	N/A	Men's Football	John Doe: 1 Zach Smith: 1	04/24/2022
X						

Figure 5.9. Remove Equipment Screen



Teams

remove team

X Men's Basketball 2023-2024

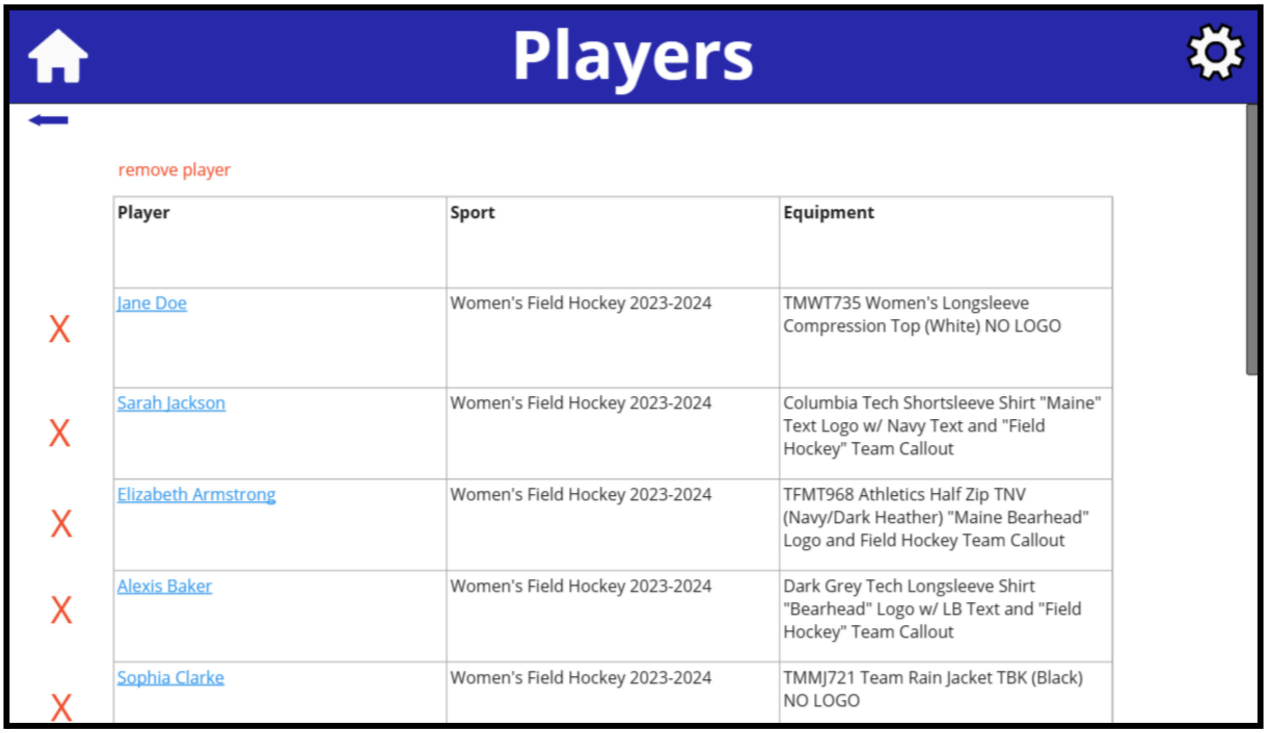
X Men's Baseball 2023-2024

X Men's Cross Country 2023-2024

X Men's Football 2023-2024

X Men's Ice Hockey 2023-2024

Figure 5.10. Remove Team Screen



Players

remove player

	Player	Sport	Equipment
X	Jane Doe	Women's Field Hockey 2023-2024	TMWT735 Women's Longsleeve Compression Top (White) NO LOGO
X	Sarah Jackson	Women's Field Hockey 2023-2024	Columbia Tech Shortsleeve Shirt "Maine" Text Logo w/ Navy Text and "Field Hockey" Team Callout
X	Elizabeth Armstrong	Women's Field Hockey 2023-2024	TFMT968 Athletics Half Zip TNV (Navy/Dark Heather) "Maine Bearhead" Logo and Field Hockey Team Callout
X	Alexis Baker	Women's Field Hockey 2023-2024	Dark Grey Tech Longsleeve Shirt "Bearhead" Logo w/ LB Text and "Field Hockey" Team Callout
X	Sophia Clarke	Women's Field Hockey 2023-2024	TMMJ721 Team Rain Jacket TBK (Black) NO LOGO

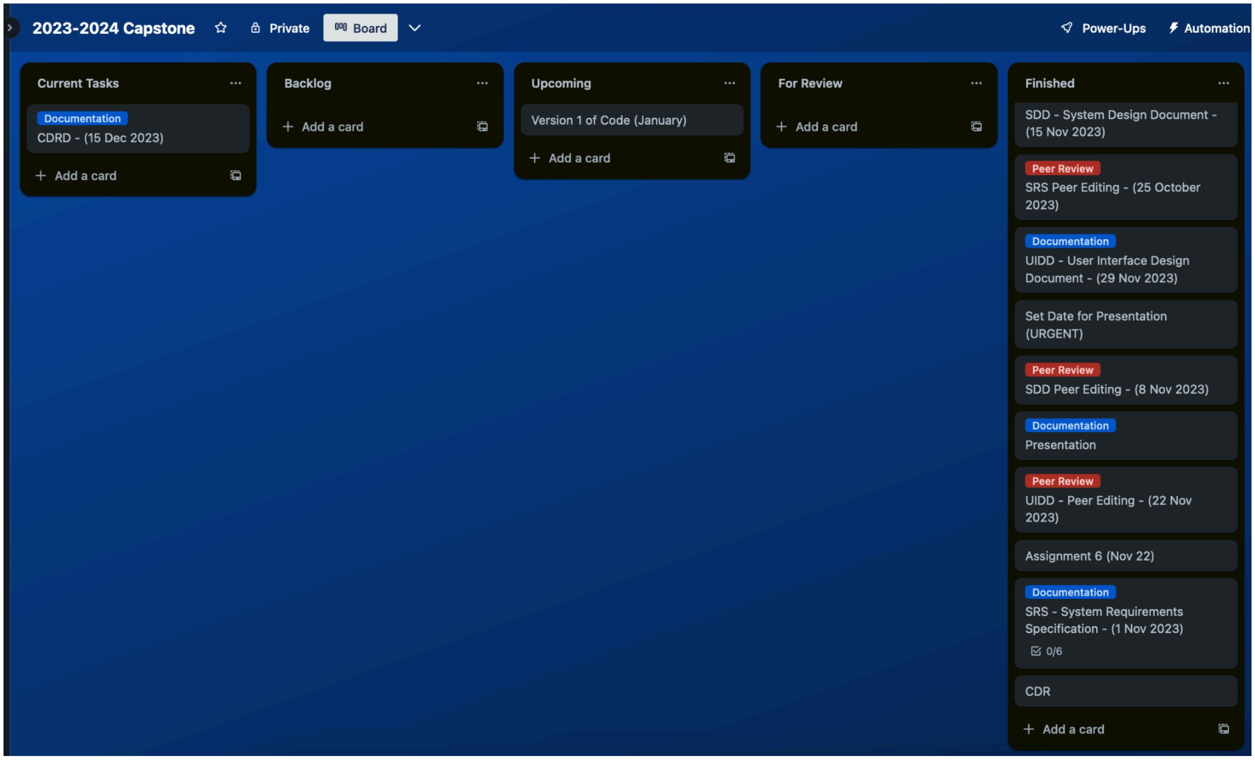
Figure 5.11. Remove Player Screen

On removal screens, there will be a red "X" to the left of every object in the category. When the user clicks the "X" next to an object, a pop up will appear to confirm the removal of the desired object. Once confirmed, the object will be removed from the database and its removal will be recorded in the database history. When equipment is removed, it is also removed from the teams and players it is assigned to. When a team is removed, the items and players assigned to that team are also removed. When a player is removed, the equipment assigned to the player is unassigned and the player is removed from the team they were assigned to.

6. DevOps

The purpsope of this section is to describe our testing, integration, and documentation strategies as the project develops.

6.1 Ticketing System



Current Tasks

- Documentation
CDRD - (15 Dec 2023)

Backlog

Upcoming

- Version 1 of Code (January)

For Review

Finished

- SDD - System Design Document - (15 Nov 2023)
- Peer Review
SRS Peer Editing - (25 October 2023)
- Documentation
UIDD - User Interface Design Document - (29 Nov 2023)
- Set Date for Presentation (URGENT)
- Peer Review
SDD Peer Editing - (8 Nov 2023)
- Documentation
Presentation
- Peer Review
UIDD - Peer Editing - (22 Nov 2023)
- Assignment 6 (Nov 22)
- Documentation
SRS - System Requirements Specification - (1 Nov 2023)
☑ 0/6
- CDR

Figure 6.1 - Trello Board

To track upcoming and current tasks across sprints, the team utilizes a Trello board help keep us organized. The board has proved to be useful in keeping track of our priorities throughout the deliverable cycle. We expect to utilize the board more when we have code sections to delegate.

6.2 Testing

The team will utilize the testing frameworks listed below. Jest, will be used to write tests that ensure our functions use the correct logic. Codacy will be used through a GitHub Action and will scan our code during pull requests to ensure that the code is written clean and secure.k6 will be used to test functionality of the system as well as load testing of our servers. The user and the team will conduct acceptance testing to ensure that system acts as intended.

Testing Type:	**Unit Testing**	**Integration Testing**	**System Testing**	**Acceptance Testing**
Testing Tool:	Jest https://jestjs.io/	Codacy Scan https://www.codacy.com/	k6 https://k6.io/	The user and team will beta test for acceptance testing

Figure 6.2 - Testing Frameworks

6.3 Source Control

The team utilizes GitHub to store all deliverable artifacts, and will continue using the same repository for our code. This ensures that all relevant documentation is stored in the same place as the code, which will best serve any future developers on the project. The repository can be found here: https://github.com/gsb02/IMSG_Capstone/tree/main.

6.4 Deployment

The team will deploy the system using automated GitHub Actions. It is not yet decided what type of server the system will be deployed to. The Github Actions will deploy the system everytime we successfully merge a pull request.

7. Project Timeline

The project timeline section outlines our key milestones and deadlines for our software. It will serve as a roadmap for our planned progression of activities from start to finish.

1. System Requirements Specification Document
   Completed November 1st, 2023
   Content
   - Initial Product Diagrams
   - Functional and Non-Functional Requirements
   - Theoretical Tests for Requirements

2. System Design Document
   Completed November 15th, 2023
   Content
   - High-level System Architecture
   - Data Flow Diagrams
   - Database Structure and Schema
   - Requirements Matrix Table

3. User Interface Design Document
   Completed December 5th, 2023.
   Content
   - Wireframes of User Interface
   - Data Validation Tables
   - Report Formats
   - User Interaction Workflows

4. Critical Design Review Document
   Estimated Date of Completion: December 16th, 2023

Content

- Problem Statement
- System Requirements Overview
- System Architecture Overview
- System Design Overview
- Dev Ops
- Timeline
- Risks
- Client Feedback

5. Development Phase
   - January 2024
     - Start coding and development
     - Build foundational components based on SRS and design documents
   - February 2024

     Documents: User Manual
     - Feature development
     - Regular development sprints with testing after each sprint
   - March 2024

     Documents: Administration Manual
     - Thorough testing of developed features
     - Address any bugs or issues
     - Begin integration testing
   - April 2024
     - System wide testing
     - Complete system hosting
     - Address any finals issues
   - May 2024
     - System deployment and presentation

8. Risks

This section outlines the primary risks that we could face during the creation and deployment of our application.

Usability - Our primary focus is that our system is easy to use. The user should be able to seamlessly navigate through the different pages of our application. They should be able easily perform routine tasks like team, player, and item creation through input fields and buttons. We also envision that our application will need minimal training, only a single training session to use the application and all its capabilities.

Security - Unauthorized users are our biggest concern. We risk potential data breaches, although this data isn't necessarily private, as a team we agree that it should only be viewable by certain people. We are also looking at unauthorized modification of data, potentially overloading our database causing our system to crash. We plan on using role based access control in combination with the UMaine login system to address this.

Data and Information Assurance - For information and data assurance, there is a risk that the user enters invalid characters into the desired field. We plan on implementing limits to certain data types such as quantity only accepting integers and names only accepting characters. We also plan on requiring additional confirmation for all user actions. For data reliability we plan on having periodic database backups. This will ensure that the data is always available and accurate.

Cost Implications - We plan to host our application in the most cost effective and efficient manner.

9. Addressing Feedback

We received most of our feedback after our presentation of the CDR. The feedback wasn't super specific, but we got enough information from our clients desires and opinions to create some general feedback points. We believe that constructive feedback is extremely important and will guide our program to fit the clients best needs.

Feedback from our client was concise, but extremely helpful. There were concerns around the types of reports and types of representation of data there would be. The client asked for more types of reports, such as new graphs other than line, and printable data sheets.

Using a good database with a strong backend will allow us to generate almost any report that our client desires. The emphasis on the types of reports, helps us prioritize the report types and the quality of those reports.

Our client also mentioned the idea of an archive. Just a space to keep old information for a set amount of time, just in case it is needed again. We agreed that it was a good idea to have such a thing, and make it easily accessible to get reports and older data.

An archive will allow our client to look back at old trends, and if issues arise with inventory disappearing, they can look back and see where things are going. An archive also ensures data security, as keeping an archive separate from the main data, we can ensure that at least past data is not lost, preferably the earliest data being within the last day.

Otherwise, our client believes that we are on the right track with our program, and likes the simple UI outline we provided him.

The audience suggested implementing an archiving system as well as reviewing our data type objects.

Figure 9.1 Recorded Feedback from CDR

Appendix A – Agreement Between Customer and Contractor

Agreement Between Customer and Contractor

1. Parties: This agreement made on “11/22/2023” is by and between

Client: University of Maine Athletic Department

AND

Contractor: Inventory Management Software Group

2. Term: The terms of this agreement shall commence on November 5th, 2023, and conclude on May, 2024.

3. Services: The Contractor agrees to provide the following services for the betterment of the Customer: The Contractor will create an inventory management system that allows the Customer to visualize and manage their inventory through an online webpage. Further details are presented in the System Design Document.

4. Expenses: There are no initial expenses. When expenses are incurred, the Contractor will communicate and get approval from the Customer to use funds allotted to them if available.

5. Agreement:
By signing this document, all parties agree to the requirements presented in this document. All parties also agree that the deadlines presented are tentative, and are subject to change as the program is developed.

Customer Signature:

	Date:	Printed:
X: *Kevin Ritz*	12/15/2023	Kevin Ritz
Contractor Signature:	Date:	Printed:
X: *Collin Rodrigue*	12/14/2023	Collin Rodrigue
X: *Gabriel A. Poulin*	12/14/2023	Gabriel A. Poulin
X: *Brennan Poitras*	12/14/2023	Brennan Poitras
X: *Sean Radel*	12/14/2023	Sean Radel
X:*Graham Bridges*	12/14/2023	Graham Bridges

Appendix B – Team Review Sign-off

Team Agreement Sign Off

Team IMSG has thoroughly reviewed the UIDD for the Athletic Inventory System and has agreed that the following information is accurate and achievable. Collectively we have no major contentions in the information stated in the document. By signing this agreement, one acknowledges all the terms and conditions outlined in the document and understands the importance of effective team collaboration, communication, and shared accountability when achieving the goals of the project. By signing below, we pledge our dedication to the success of the team and the project we plan to undertake. We agree to work collaboratively, and support each other to uphold the guidelines and expectations set forth in the agreement.

Signature:	Date:	Printed:
X: *Collin Rodrigue*	12/14/2023	Collin Rodrigue
X: *Gabriel A. Poulin*	12/14/2023	Gabriel A. Poulin
X: *Brennan Poitras*	12/14/2023	Brennan Poitras
X: *Sean Radel*	12/14/2023	Sean Radel
X:*Graham Bridges*	12/14/2023	Graham Bridges

Appendix C – Document Contributions

Name	Date	Contribution	Version
Sean Radel	12/6/23	File formatting	0.1
Gabriel Poulin	12/6/23	Feedback	0.1
Graham Bridges	12/13/23	User Interface Design, Non-functional Requirements, Purpose of the Document	0.1
Collin Rodrigue	12/14/23	Project Timeline/Risks	0.1
Sean Radel	12/14/23	Devops, System Design	0.1
Brennan Poitras	12/14/23	Functional Requirements, Database Design	0.1

Appendix D – Software Requirements Specification

COS 397: Computer Science Capstone I

System Requirements Specification

UMaine Athletic Department Inventory Management System

Version 0.3

Collin Rodrigue, Brennan Poitras, Graham Bridges, Gabe Poulin, Sean Radel

1 November 2023

Introduction

This is a capstone project for the University of Maine Athletics Department. The Inventory Management Software Group is developing a scalable cloud based solution to assist in the management and tracking of the University of Maine Athletic Department equipment and item inventory. The key stakeholders for our project are Jude Killy, Nick Fox, and Kevin Ritz. Jude Killy submitted the project on behalf of the athletic department to combat the issue of an inefficient inventory management process. Currently, Nick Fox and Kevin Ritz are tasked with managing inventory orders and distribution of items to players and teams. This process is only documented on a spreadsheet, which can be found in section 1.2 References. The group aims to develop a solution that can increase the efficiency of entering data items, storing inventory data, and tracking the distribution to athletes and teams. Our solution replaces the old inventory management solution Front Rush and the currently used strategy that's a combination of Excel spreadsheets and word of mouth to track inventory.

1.1 Purpose of This Document

The purpose of this document is to describe the core requirements of our inventory management system. After introducing our project, we describe our functional requirements using a UML diagram and further our explanation with use case specification tables. Next, we document our non-functional requirements, which describe aspects of the system like performance, reliability, scalability, compliance, security, etc. Functional requirements describe what the system should do, while non-functional requirements describe how the system should perform. Following our requirements, we describe our deliverables and preliminary schedule for those documents.

1.2. References

1. Google Sheets. *Athletic Inventory*. University of Maine Athletic Department, (non-acessable to public), 15 Nov. 2023

2. Mitre ATT&CK®. MITRE ATT&CK®. (n.d.). https://attack.mitre.org/ Accessed 15 Nov. 2023

1.3. Purpose of the Product

The purpose of the product is to fulfill the customer needs of an inventory management solution. Our system will replace the customer's previous solutions for managing their equipment. The customer previously used Front Rush and currently uses a combination of Excel spreadsheets and word of mouth to track inventory. Our product will allow the customer to organize their inventory by associating equipment with teams and players. The system will be designed with simplicity in mind so that they do not need experienced developers to maintain their product following the delivery date. Equipment, player accounts, and teams will be able to be made on the fly to allow the system to scale to the customer's needs.

1.4. Product Scope

The top level use case diagrams define the scope of our product. All product features are abstractly defined.

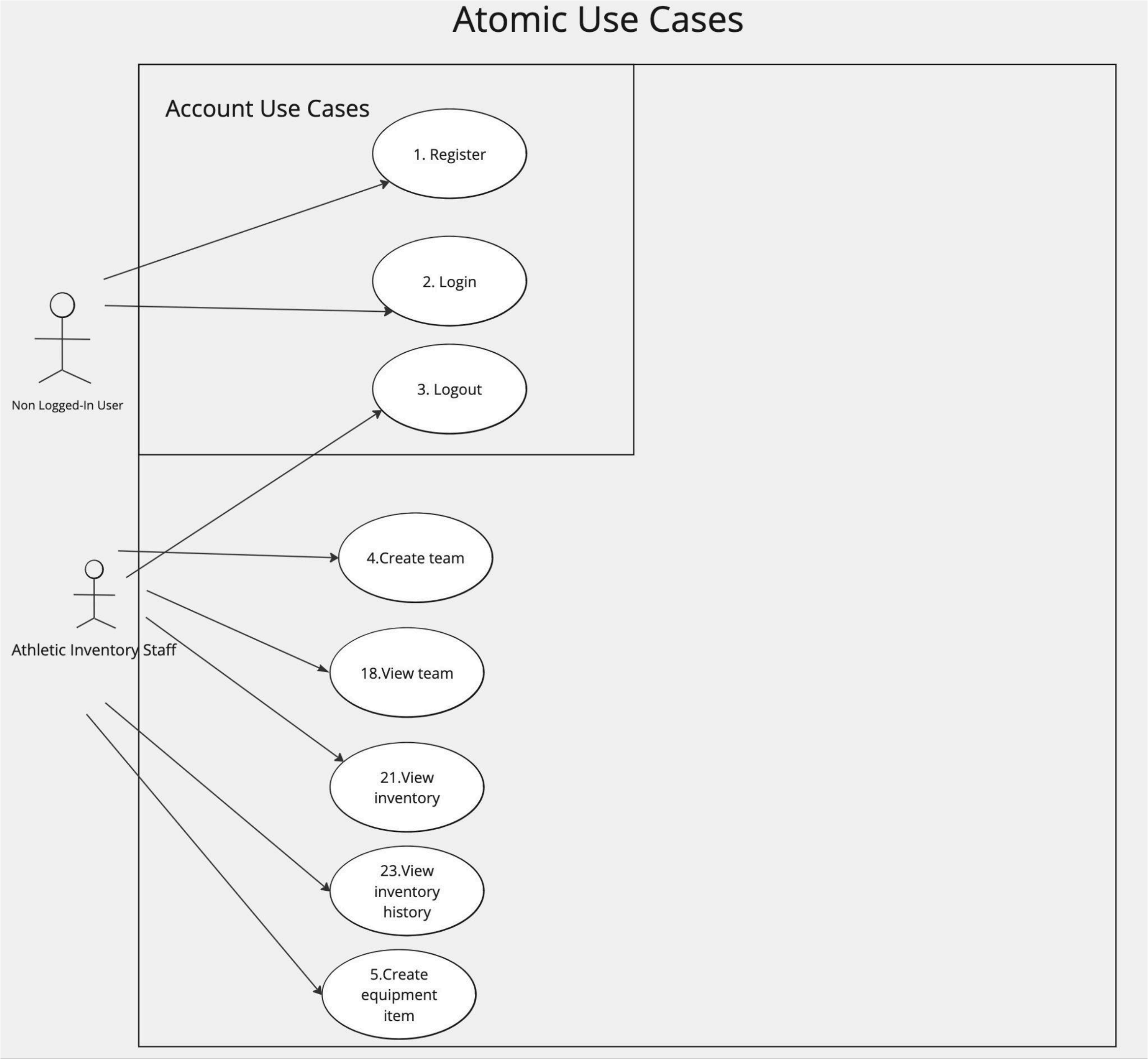


Figure 1. Atomic Use Case Diagram

The atomic use case diagram represents the most critical features and elements of the application. These features will be used regularly and will be easily accessible through the user interface.

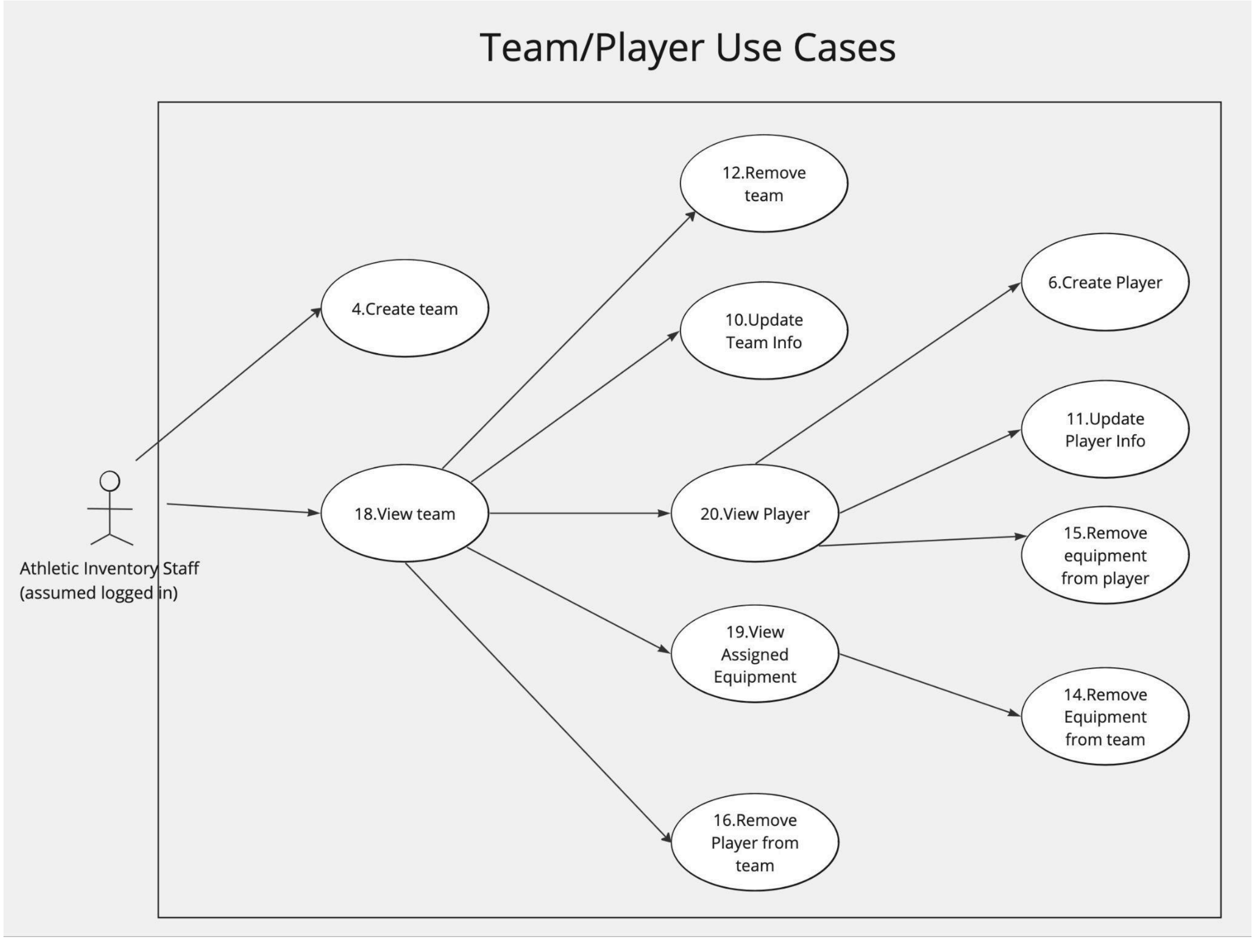

Team/Player Use Cases
12.Remove team
4.Create team
10.Update Team Info
6.Create Player
11.Update Player Info
18.View team
20.View Player
15.Remove equipment from player
Athletic Inventory Staff (assumed logged in)
19.View Assigned Equipment
14.Remove Equipment from team
16.Remove Player from team

Figure 2. Team and Player Use Cases

The Team and Player use case diagram represents product features related to sports teams and the players on those teams. These actions mainly include the removal and update of equipment, teams, and players.

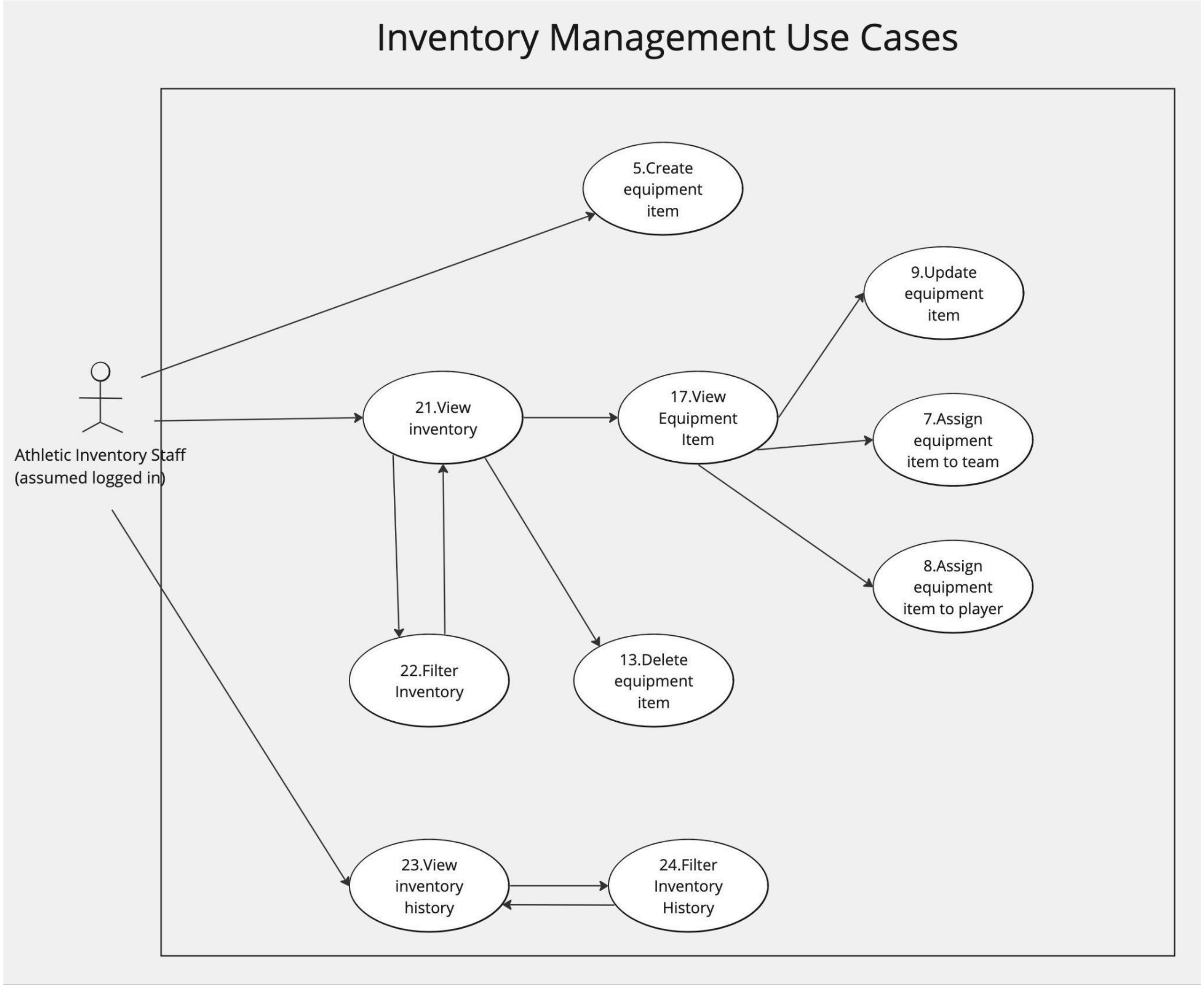

Inventory Management Use Cases
Athletic Inventory Staff (assumed logged in)
5.Create equipment item
9.Update equipment item
21.View inventory
17.View Equipment Item
7.Assign equipment item to team
8.Assign equipment item to player
22.Filter Inventory
13.Delete equipment item
23.View inventory history
24.Filter Inventory History

Figure 3. Inventory Management Use Cases

The Inventory Management use case diagram represents the standard behaviors of an inventory system including input of new inventory, output of inventory to teams and players, and viewing all products that are currently being stored.

2. Functional Requirements

Functional requirements define the features and use cases that the system possesses to meet customer needs. Our use case descriptions expand on the top level use case diagrams in Section 1.4. The functional requirements describe how users can interact with the system and what the expected preconditions and postconditions will be. Furthermore, we define test cases to verify that our system meets requirements.

2.1 Functional Requirements

FR1: The system shall allow a user to register an account.
FR2: The system shall allow a user to log in.
FR3: The system shall allow a user to log out.
FR4: The system shall allow a user to add a new sports team.
FR5: The system shall allow a user to add a new equipment item.
FR6: The system shall allow a user to create a new player.
FR7: The system shall allow a user to assign an equipment item to a team.
FR8: The system shall allow a user to assign an equipment item to a player.
FR9: The system shall allow a user to update an equipment item.
FR10: The system shall allow a user to update team information.
FR11: The system shall allow a user to update player information.
FR12: The system shall allow a user to remove a team.
FR13: The system shall allow a user to remove an equipment item from inventory.
FR14: The system shall allow a user to unassign an equipment item from a team.
FR15: The system shall allow a user to unassign an equipment item from a player.
FR16: The system shall allow a user to remove a player from a team.
FR17: The system shall allow a user to view an equipment item.
FR18: The system shall allow a user to view a a team's roster.
FR19: The system shall allow a user to view a team's assigned equipment.
FR20: The system shall allow a user to view a player.
FR21: The system shall allow a user to view general inventory.
FR22: The system shall allow a user to filter the general inventory.
FR23: The system shall allow a user to view the inventory history.
FR24: The system shall allow a user to filter the inventory history.

2.2 Use Case Descriptions

Number	1	
Name	Register	
Summary	User registers account in the system's database	
Priority	5	
Preconditions	User must be on the website User must have a valid maine.edu email	
Postconditions	User has created a registered account in the system	
Primary Actor	Athletic Inventory Staff	
Secondary Actors		
Trigger	User clicks "Register" on web page	
Main Scenario	**Step**	**Action**

	1	User clicks “Register” button
	2	User enters maine.edu email
	3	User creates password for account
	4	Account creation is approved by system admin
	5	Account is successfully created
Extensions	**Step**	**Branching Action**
	2a	Chosen maine.edu email already has an account: - System displays a message stating “That maine.edu email already has a registered account. Choose another email or log in with the selected account”.
	2b	Chosen email isn’t a maine.edu email: - System displays message saying “Please enter a maine.edu email”
	3a	Users password doesn’t contain an uppercase letter, lowercase letter, number, and a special character contained in it, and/or it is less than nine characters long: - System requests that the user changes the password so that it meets the requirements above
	4a	Account creation isn’t approved by system admin: - User’s account is not established in the system
Open Issues	Will registering and logging in be done through a portal to the maine.edu system, such as with brightspace and maine street? Or will users be able to set up a new password different from their umaine.edu password?	
Test Case	Query the database to confirm there is a record associated with the created account.	

Number	2	
Name	Log in	
Summary	User logs into website in order to access other functions	
Priority	5	
Preconditions	User must be on the website User must not be logged in already User must have a valid account	
Postconditions	User is logged in	
Primary Actor	Athletic Inventory Staff	
Secondary Actors		
Trigger	User clicks “Log in” button on web page	
Main Scenario	**Step**	**Action**
	1	User clicks “Log in” button
	2	User inputs maine.edu email and password
	3	User confirms log in and presses “continue”

	4	User is redirected to homepage logged in
Extensions	**Step**	**Branching Action**
	2a	User clicks “forgot password”: - “Reset Password” use case is triggered and user is redirected
	3a	User’s login credentials are incorrect: - User is redirected to log in page and message saying “invalid credentials” appears
Open Issues	Decide how many attempts are allowed for the user to log in and what should be done when the threshold is met	
Test Case	Scan the HTML document to see if the user is no longer on the log in page.	

Number	3	
Name	Log out	
Summary	User logs out of website and returns to homepage signed out	
Priority	5	
Preconditions	User must be on the website User must be logged in	
Postconditions	User is logged out	
Primary Actor	Athletic Inventory Staff	
Secondary Actors		
Trigger	User clicks “Log out” button on web page	
Main Scenario	**Step**	**Action**
	1	User clicks “Log out” button
	2	User is redirected to homepage logged out
Extensions	**Step**	**Branching Action**
	N/A	
Open Issues	Should users be logged out of their account after a certain amount of time idle on the website? If so, how long should this timer be?	
Test Case	Scan the HTML document to see if the user is on the log in page.	

Number	4
Name	Create a new team
Summary	Input a new sports team along with team details
Priority	5
Preconditions	User is logged in.
Postconditions	The database will contain a record matching the created team object.
Primary Actor	Athletic Inventory Staff

Secondary Actors		
Trigger	The user clicks a button titled “Add a new team”	
Main Scenario	**Step**	**Action**
	1	User clicks “Add a new team”
	2	User inputs team details (sport, gender, season, etc.)
	3	User clicks a confirmation button
Extensions	**Step**	**Branching Action**
	2a	A team already exists with the exact same parameters(sport, gender season, etc.): - System displays a message saying “A team with these parameters already exists. Please choose different parameters or delete the previous team” - Creation of the new team is prevented until the parameters aren’t the same as another team
	3a	Not all required parameters are set for the team: - System tells the user they must fill out all required parameters before the team can be created
Open Issues	What parameters should be required when making a team?	
Test Case	Query the database to confirm it contains a record matching the created team object.	

Number	5	
Name	Create an equipment item	
Summary	Input a new equipment item along with its equipment details	
Priority	5	
Preconditions	User is logged in.	
Postconditions	The database will contain a record matching the created equipment item object.	
Primary Actor	Athletic Inventory Staff	
Secondary Actors		
Trigger	The user clicks a button titled “Add new equipment item”	
Main Scenario	**Step**	**Action**
	1	User clicks “Add new equipment item”
	2	User inputs equipment details (name, description, quantity, etc.)
	3	User clicks a confirmation button
Extensions	**Step**	**Branching Action**
	2a	An item already exists containing the same name: - System informs the user that an item with the same name already exists - System prevent creation of the new item until the name is changed to something not already taken

Open Issues	
Test Case	Query the database to confirm it contains a record matching the created equipment item object.

Number	6	
Name	Create a new player	
Summary	Input a new athlete along with their details, and assign them to a sports team.	
Priority	5	
Preconditions	At least 1 sports team must have been created. User is logged in.	
Postconditions	The database will contain a record matching the created player object.	
Primary Actor	Athletic Inventory Staff	
Secondary Actors		
Trigger	The user clicks a button titled “Add a new player”	
Main Scenario	**Step**	**Action**
	1	User clicks on the team they wish to assign a player.
	2	User clicks “Add a new player”
	3	User inputs player details (name, height, weight, class, etc.)
	4	User clicks a confirmation button
Extensions	**Step**	**Branching Action**
	N/A	
Open Issues	What if two players have the same name? Should they automatically be numbered 1 and 2 in the system? Or should the user be responsible for differentiating the two players?	
Test Case	Query the database to confirm it contains a record matching the created player object.	

Number	7
Name	Assign equipment item to team
Summary	An existing equipment item will be assigned to a sports team.
Priority	5
Preconditions	At least 1 equipment item must have been created. At least 1 team must have been created. User is logged in.
Postconditions	
Primary Actor	Athletic Inventory Staff

Secondary Actors		
Trigger	The user clicks a button titled “Assign equipment to team”	
Main Scenario	**Step**	**Action**
	1	User clicks on an equipment item
	2	User clicks “Assign equipment to team”
	3	User clicks the team they wish to assign equipment to
	4	User enters desired amount
	5	User clicks a confirmation button
Extensions	**Step**	**Branching Action**
	4a	The desired amount of the item isn’t available: - System displays message saying “Only X amount of this item is available for assignment”, X being the quantity of the item unassigned to teams
Open Issues		
Test Case	Query the database to confirm it contains a relationship between the desired team and the desired equipment item.	

Number	8	
Name	Assign equipment to player	
Summary	A player will be assigned equipment specific to them	
Priority	4	
Preconditions	At least 1 equipment item must exist At least 1 team must exist At least 1 player must have been assigned to a team User is logged in.	
Postconditions		
Primary Actor	Athletic Inventory Staff	
Secondary Actors		
Trigger	The user clicks a button titled “Assign equipment to player”	
Main Scenario	**Step**	**Action**
	1	User clicks on an equipment item
	2	User clicks “Assign equipment to player”
	3	User clicks on the players associated team
	4	User clicks on the player they wish to assign equipment to
	5	User enter quantity of the item to be assigned to the player
	6	User clicks a confirmation button
Extensions	**Step**	**Branching Action**
	5a	The desired amount of the item isn’t available: - System displays message saying “Only X amount of this item is available for assignment”, X being the quantity of the

		item unassigned to players out of the team's assigned equipment
Open Issues		
Test Case	Query the database to confirm it contains a relationship between the desired player and the desired equipment item.	

Number	9	
Name	Update equipment item	
Summary	Update the details of a previously created equipment item	
Priority	4	
Preconditions	At least 1 equipment item must have been created User is logged in.	
Postconditions	The database will contain an updated record matching the updated equipment object.	
Primary Actor	Athletic Inventory Staff	
Secondary Actors		
Trigger	The user clicks a button titled "Update equipment item"	
Main Scenario	**Step**	**Action**
	1	User clicks on an equipment item
	2	User clicks "Update equipment item"
	3	User updates fields with desired detail changes.
	4	User clicks a confirmation button
Extensions	**Step**	**Branching Action**
	3a	User changes name of the item to a name already being used by another item: - System displays message saying "Name is already in use, please choose another name" - System prevents item from being updated until the name is different from other names in the database
Open Issues		
Test Case	Query the database to confirm if the record associated with the equipment item was updated correctly.	

Number	10
Name	Update team info
Summary	Update details of a sports team

Priority	4	
Preconditions	At least 1 team must have been created User is logged in.	
Postconditions	The database will contain an updated record matching the updated team object.	
Primary Actor	Athletic Inventory Staff	
Secondary Actors		
Trigger	The user clicks a button titled “Update team details”	
Main Scenario	**Step**	**Action**
	1	User clicks on the sports team they wish to update
	2	User clicks “Update team details”
	3	User updates desired team details (sport, gender, season, etc.)
	4	User clicks a confirmation button
Extensions	**Step**	**Branching Action**
	3a	Updated parameters are the same as another existing team: - System displays a message saying “A team already exists with the same parameters. Please choose different parameters” - System prevents user from updating the team before changing the parameters
Open Issues		
Test Case	Query the database to confirm if the record associated with the team was updated correctly.	

Number	11	
Name	Update player info	
Summary	Update details of an athlete	
Priority	4	
Preconditions	At least 1 player must have been created User is logged in.	
Postconditions	The database will contain an updated record matching the updated player object.	
Primary Actor	Athletic Inventory Staff	
Secondary Actors		
Trigger	The user clicks a button titled “Update player info”	
Main Scenario	**Step**	**Action**
	1	User clicks on team containing player that they wish to update
	2	User clicks on the player they wish to update
	3	User clicks “Update player info”
	4	User updates desired player details (name, height, weight, class, etc.)
	5	User clicks a confirmation button

Extensions	**Step**	**Branching Action**
	N/A	
Open Issues		
Test Case	Query the database to confirm if the record associated with the player was updated correctly.	

Number	12	
Name	Remove Team	
Summary	User removes athletic team from the system	
Priority	4	
Preconditions	Team desired to be deleted is in the system User is logged in	
Postconditions	Deleted team is removed from the system	
Primary Actor	Athletic Inventory Staff	
Secondary Actors		
Trigger	User clicks “remove team” below team display	
Main Scenario	**Step**	**Action**
	1	System asks user to enter their password
	2	User must confirm they wish to remove the team once more
	3	Team is deleted from the system
	4	User is redirected back to “view teams” page
Extensions	**Step**	
	1a	User’s password is incorrect: System displays “incorrect password and lets the user try again”
Open Issues	Decide how many attempts are given to the user to enter their password and what should be done when the threshold is met Are players assigned to that team also deleted? Is equipment assigned to the deleted team and its players deleted or made available for assignment to other teams?	
Test Case	Query the database to confirm a record does not exist for the desired team.	

Number	13
Name	Remove equipment item from inventory
Summary	Removing an equipment item from the entire inventory and the teams and players that were assigned the equipment.
Priority	5
Preconditions	An equipment item must have been created User is logged in.

Postconditions	The database will no longer contain a record matching the equipment item. The equipment item will no longer be assigned to any teams. The equipment item will no longer be assigned to any players.	
Primary Actor	Athletic Inventory Staff	
Secondary Actors		
Trigger	The user clicks a button titled "Delete equipment item"	
Main Scenario	**Step**	**Action**
	1	User clicks on an equipment item
	2	User clicks "Delete equipment item"
	3	User receives a warning and clicks a confirmation button
Extensions	**Step**	**Branching Action**
	N/A	
Open Issues		
Test Cases	Query the database to confirm a record does not exist for the desired equipment item.	

Number	14	
Name	Remove equipment item from team	
Summary	Removing an equipment item from a team to which it was assigned.	
Priority	4	
Preconditions	At least 1 equipment item must have been assigned to the team. User is logged in.	
Postconditions		
Primary Actor	Athletic Inventory Staff	
Secondary Actors		
Trigger	The user clicks a button titled "Remove equipment item from team"	
Main Scenario	**Step**	**Action**
	1	User clicks on a team they wish to remove equipment from.
	2	User clicks on "Assigned Equipment"
	3	User clicks on the equipment item they wish to remove
	4	User clicks "Remove equipment item from team"
	5	User receives a warning and clicks a confirmation button
Extensions	**Step**	**Branching Action**
	N/A	
Open Issues		
Test Case	Query the database to confirm a relationship does not exist between the desired equipment item and the desired team.	

Number	15

Name	Remove equipment item from a player	
Summary	Removing an equipment item from a player to which it was assigned	
Priority	4	
Preconditions	At least 1 equipment item must have been assigned to the player User is logged in.	
Postconditions		
Primary Actor	Athletic Inventory Staff	
Secondary Actors		
Trigger	The user clicks a button titled “Remove equipment item from player”	
Main Scenario	**Step**	**Action**
	1	User clicks on a team with the desired player
	2	User clicks on the player they wish to remove equipment from
	3	User clicks on the equipment item they wish to remove
	4	User clicks “Remove equipment item from player”
	5	User receives a warning and clicks a confirmation button
Extensions	**Step**	**Branching Action**
	N/A	
Open Issues		
Test Case	Query the database to confirm a relationship does not exist between the desired equipment item and the desired player.	

Number	16	
Name	Remove player from team	
Summary	Removing player from the team they were assigned	
Priority	5	
Preconditions	At least 1 player must be assigned to a team User is logged in.	
Postconditions		
Primary Actor	Athletic Inventory Staff	
Secondary Actors		
Trigger	The user clicks a button titled “Remove player from team”	
Main Scenario	**Step**	**Action**
	1	User clicks on the team containing the player they wish to remove
	2	User hovers over player they wish to remove
	3	User clicks “Remove player from team”
	4	User receives a warning and clicks a confirmation button
Extensions	**Step**	**Branching Action**
	N/A	
Open Issues	What happens to equipment assigned to the deleted player? Is it returned to the team unassigned or deleted?	

Test Case	Query the database to confirm a relationship does not exist between the desired player and the desired team.

Number	17	
Name	Viewing Equipment Item	
Summary	The user can view equipment item details, including name, description, quantity, image, color, etc.	
Priority	5	
Preconditions	An equipment item must have been created User is logged in.	
Postconditions		
Primary Actor	Athletic Inventory Staff	
Secondary Actors		
Trigger	The user clicks on an equipment item	
Main Scenario	**Step**	**Action**
	1	User clicks on an equipment item
Extensions	**Step**	**Branching Action**
	N/A	
Open Issues		
Test Case	Scan the HTML document and query the database to confirm the data displayed on the screen matches the data associated with the equipment record.	

Number	18	
Name	Viewing a teams roster	
Summary	View all players listed on the roster for team	
Priority	5	
Preconditions	A team must have at least 1 player assigned to it User is logged in.	
Postconditions		
Primary Actor	Athletic Inventory Staff	
Secondary Actors		
Trigger	The user clicks the team for which roster they want to view	
Main Scenario	**Step**	**Action**
	1	User clicks on a team
Extensions	**Step**	**Branching Action**
	N/A	
Open Issues		

Test Case	Scan the HTML document and query the database to confirm the data displaying on the screen matches the data associated with the team record.

Number	19	
Name	Viewing a teams assigned equipment	
Summary	View all the equipment items that have been assigned to a team	
Priority	5	
Preconditions	At least 1 equipment item must have been assigned to the team User is logged in.	
Postconditions		
Primary Actor	Athletic Inventory Staff	
Secondary Actors		
Trigger	The user clicks on “Assigned Equipment”	
Main Scenario	**Step**	**Action**
	1	User clicks on a team for which they wish to view assigned equipment
	2	User clicks “Assigned Equipment”
Extensions	**Step**	**Branching Action**
	N/A	
Open Issues		
Test Case	Scan the HTML document and query the database to confirm the data displaying on the screen matches the data associated with the relationship between the team and its assigned equipment.	

Number	20	
Name	Viewing a player	
Summary	View the details associated with a player	
Priority	5	
Preconditions	At least 1 player must be assigned to a team User is logged in.	
Postconditions		
Primary Actor	Athletic Inventory Staff	
Secondary Actors		
Trigger	The user clicks on a players name on the team roster list	
Main Scenario	**Step**	**Action**
	1	User clicks on a team containing the player they wish to view
	2	User clicks on the players name they wish to view
Extensions	**Step**	**Branching Action**
	N/A	
Open Issues		

Test Case	Scan the HTML document and query the database to confirm the data displayed on the screen matches the data associated with the player record.

Number	21	
Name	Viewing general inventory	
Summary	The user can view all of the current equipment items in the inventory room, assigned to a team, or assigned to a player.	
Priority	5	
Preconditions	An equipment item must have been created User is logged in.	
Postconditions	The user is shown all equipment items in the system	
Primary Actor	Athletic Inventory Staff	
Secondary Actors		
Trigger	The user clicks a button titled “Inventory”	
Main Scenario	**Step**	**Action**
	1	User clicks on “Inventory”
Extensions	**Step**	**Branching Action**
	N/A	
Open Issues		
Test Case	Scan the HTML document to confirm the user is on the inventory screen.	

Number	22	
Name	Filter general inventory	
Summary	The user can filter the general inventory by name, team equipment, and type	
Priority	3	
Preconditions	User is on equipment inventory screen User is logged in.	
Postconditions	Inventory on screen is updated to show only equipment that matches the filters	
Primary Actor	Athletic Inventory Staff	
Secondary Actors		
Trigger	User clicks “filter”while viewing inventory	
Main Scenario	**Step**	**Action**
	1	User selects desired filters
	2	User confirms filter
	3	Display is updated to show filtered equipment
Extensions	**Step**	**Branching Action**
	3a	No equipment matches selected filters: System displays message saying no equipment matches search

Open Issues	Decide important filter options for the user
Test Case	Query the database to confirm the returned, filtered records match the equipment items displayed on the screen.

Number	23	
Name	Viewing inventory history	
Summary	Viewing a history of all previous inventory transactions including creating an item, assigning an item, updating an item, and deleting an item. This also includes team and player transactions.	
Priority	5	
Preconditions	At least 1 previous inventory action must have been taken User is logged in.	
Postconditions		
Primary Actor	Athletic Inventory Staff	
Secondary Actors		
Trigger	The user clicks a button titled "Inventory History"	
Main Scenario	**Step**	**Action**
	1	User clicks on "Inventory History"
Extensions	**Step**	**Branching Action**
	N/A	
Open Issues		
Test Case	Scan the HTML document to confirm the user is on the inventory history screen.	

Number	24	
Name	Filtering Inventory History	
Summary	Filtering the inventory history on the inventory history page so the user can search for specific transactions	
Priority	3	
Preconditions	User is on inventory history page User is logged in.	
Postconditions	System is displaying filtered inventory history	
Primary Actor	Athletic Inventory Staff	
Secondary Actors		
Trigger	The user clicks "filter" on inventory history page	
Main Scenario	**Step**	**Action**
	1	User applies desired history filters including specific items, teams, players, date range, etc.
	2	User confirms desired filters
	3	Display is updated to show filtered history
Extensions	**Step**	**Branching Action**

	3a	No inventory history matches applied filters: System displays message saying no transaction matches search
Open Issues		
Test Case	Query the database to confirm the returned, filtered records match the history records displayed on the screen.	

3. Non-Functional Requirements

Non-Functional Requirements define aspects of how the system will perform. These requirements are imperative to ensuring the system functions with the desired quality and performance.

3.1 Non-Functional Requirements List

NFR-001	**Priority:** 3
Description: The system must provide data security for athletes and inventory information via encryption at rest and in transit.	
Test Case: Query the data without encryption keys to ensure the data is encrypted	

NFR-002	**Priority:** 3
Description: The system should ensure a response time of under 3 seconds for the inventory item filter.	
Test Case: By running a repeated access attempt on different items and filters, we should be able to graph the response time of each access attempt.	

NFR-003	**Priority:** 5
Description: The system must be available for use 99.9% of the time.	
Test Case: Having a downtime detector will allow for accurate graphing of real-time downage.	

NFR-004	**Priority:** 5
Description: The system should be compatible with all major web browsers using Chromium 80.0.3987.33 and later.	

Test Case: Running the application on major browsers such as Edge, Chrome, and OperaGX will show compatibility with other browsers running on Chromium.

NFR-005	**Priority:** 5
Description: The system should support at least 10 concurrent users.	
Test Case: By using third-party software, we can write scripts to have multiple ghost users sign into the application and use it at will, testing a number higher than 10 is ideal.	

NFR-006	**Priority:** 5
Description: The system should not have any major vulnerabilities.	
Test Case: We will conduct a penetration test of the service that scans the system dependencies for CVEs, checks for XSS attack capabilities, and validates there are no unsanitized data inputs. The team will leverage the MITRE ATT&CK matrix to track what additional TTPs we need to address.	

NFR-007	**Priority:** 5
Description: The system should have user authentication through a “maine.edu” email address and password combination.	
Test Case: Testing accounts not affiliated with the university to make sure they cannot get access is ideal, as well as making sure the “maine.edu” accounts sync with the UMaine database for password syncing.	

NFR-008	**Priority:** 3
Description: The system must have a responsive and user-friendly interface, ensuring that user actions result in fast visual feedback.	
Test Case: Having continuous tests with end users to get feedback and improve until the desired product is achieved.	

NFR-009	**Priority:** 5

Description: The system must allow for the addition of new items and users.
Test Case: General testing for adding items and users through the user interface, as well as testing the overall speed at which you can add those items and users.

NFR-010	**Priority:** 4
Description: The system shouldn’t have an error rate adding or modifying data exceeding 5%.	
Test Case: Write scripts to modify data at mass and have the script quantify how many fail.	

NFR-011	**Priority:** 4
Description: The system should allow for multiple administrative users.	
Test Case: We will create multiple administrative users using scripts, then have those users do admin-level activities and see if there are any issues.	

NFR-012	**Priority:** 5
Description: The system should be able to be maintained by administrators through the user interface.	
Test Case: Having an end user quality test our product as an admin would be best here.	

NFR-013	**Priority:** 4
Description: The system should be able to handle an increasing number of inventory items without any performance degradation.	
Test Case: By writing scripts to add items constantly, we can see the time it takes to create those items and the time it takes to access as more and more are added.	

NFR-014	**Priority:** 2
Description: The system should be maintained in the most cost-effective manner.	
Test Case: By remaining within the budget we are allotted.	

NFR-015	**Priority:** 3
Description: The system shall be easily configurable by user administrators.	
Test Case: We will have our intended user base use the admin abilities as we integrate new things onto it. We will get their opinion on what they do and don't like and adjust accordingly.	

NFR-016	**Priority:** 1
Description: The system should provide accurate data analytics with timetables.	
Test Case: By having both the timestamps of items created and/or modified, we can compare the timestamps and tables in the graphs to the actual data using scripts.	

4. User Interface

See *Athletic Department Inventory Management System User Interface Design Document*

5. Deliverables

A preliminary overview of the planned deliverables for the product. This list is subject to change as the project moves forward.

5.1 List of Deliverables and Dates

Item	Date	Format
System Requirement Specification	1 November 2023	Digital PDF
System Design Document	15 November 2023	Digital PDF
User Interface Design Document	29 November 2023	Digital PDF
Critical Design Review Document	15 December 2023	Digital PDF
User Manual	Early 2024	Digital PDF
Administrator Manual	Early 2024	Digital PDF
All source code	March 2024	GitHub
Complete System Hosted on Server	April 2024	Acessable via web browser

6. Open Issues

The open issues section details what problems we have come across but are yet to address.

6.1 List of Issues and Dates

Sign agreement between customer and contractor	8 November 2023
Team Sign-off	1 November 2023

SRS - Appendix A – Agreement Between Customer and Contractor

Agreement Between Customer and Contractor

1. Parties: This agreement made on “10/30/2023” is by and between

Client: University of Maine Athletic Department

AND

Contractor: Inventory Management Software Group

2. Term: The terms of this agreement shall commence on November 5th, 2023, and conclude on May, 2024.

3. Services: The Contractor agrees to provide the following services for the betterment of the Customer: The Contractor will create an inventory management system that allows the Customer to visualize and manage their inventory through an online webpage. Further details are presented in the System Requirements Specification document.

4. Expenses: There are no initial expenses. When expenses are incurred, the Contractor will communicate and get approval from the Customer to use funds allotted to them if available.

5. Agreement:

By signing this document, all parties agree to the requirements presented in this document. All parties also agree that the deadlines presented are tentative, and are subject to change as the program is developed.

Customer Signature:		
	Date:	Printed:
X: *KEvin Ritz*	*11/8/23*	
Contractor Signature:	Date:	Printed:
X: *Collin Rodrigue*	10/30/2023	Collin Rodrigue
X: *Gabriel A. Poulin*	10/30/2023	Gabriel A. Poulin
X: *Brennan Poitras*	10/30/2023	Brennan Poitras
X: *Sean Radel*	10/30/2023	Sean Radel
X:*Graham Bridges*	10/30/2023	Graham Bridges

SRS - Appendix B – Team Review Sign-off

Team Agreement Sign Off

Team IMSG has thoroughly reviewed the System Requirements Document for the Athletic Inventory System and has agreed that the following information is accurate and achievable. Collectively we have no major contentions in the information stated in the document. By signing this agreement, one acknowledges all the terms and conditions outlined in the document and understands the importance of effective team collaboration, communication, and shared accountability when achieving the goals of the project. By signing below, we pledge our dedication to the success of the team and the project we plan to undertake. We agree to work collaboratively, and support each other to uphold the guidelines and expectations set forth in the agreement.

Signature:	Date:	Printed:
X: *Collin Rodrigue*	10/30/2023	Collin Rodrigue
X: *Gabriel A. Poulin*	10/30/2023	Gabriel A. Poulin
X: *Brennan Poitras*	10/30/2023	Brennan Poitras
X: *Sean Radel*	10/30/2023	Sean Radel
X:*Graham Bridges*	10/30/2023	Graham Bridges

Appendix C – Document Contributions

Name	Date	Contribution	Version
Sean Radel	10/22/23	Developed Multiple Sections of the document,	0.1
Collin Rodrigue	10/22/23	Non-functional requirements	0.1
Brennan Poitras	10/22/23	Functional requirements and test cases	0.1
Gabriel Poulin	10/22/23	Non Functional Requirements	0.1
Graham Bridges	10/22/23	Functional Requirements	0.1
Brennan Poitras	10/31/23	Use Case diagrams and updates to Use Case tables	0.2
Sean Radel	11/1/23	Adjusted section 1.4 and introduction of section 1. Added test cases to functional and non-functional requirements	0.2
Graham Bridges	11/1/23	Updated Use Case diagrams, specifically alternate flows and open issues	0.2
Collin Rodrigue	11/1/23	Appendix B	0.2
Graham Bridges	11/6/23	Added logo	0.3

Appendix E – System Design Document

COS 397: Computer Science Capstone I

System Design Document

UMaine Athletic Department Inventory Management System

IMSG

Version 0.2

Collin Rodrigue, Brennan Poitras, Graham Bridges, Gabe Poulin, Sean Radel

15 November 2023

1 Introduction

The purpose of the product is to fulfill the customer needs of an inventory management solution. Our system will replace the customer's previous solutions for managing their equipment. The customer previously used Front Rush and currently uses a combination of Excel spreadsheets and word of mouth to track inventory. Our product will allow the customer to organize their inventory by associating equipment with teams and players. The system will be designed with simplicity in mind so that they do not need experienced developers to maintain their product following the delivery date. Equipment, player accounts, and teams will be able to be made on demand to allow the system to scale to the customer's needs. Our customer is the University of Maine Athletic Department, and specifically Jude Killy, Nick Fox, and Kevin Ritz.

1.1 Purpose of This Document

The purpose of this document is to abstractly illustrate our software architecture, further specify our classes, files, and requirements. We aim to document how our product will work together as subsystems of one large system. This document describes the datatypes we will use in our database and how the data objects are connected to each other. This document helps to show the team what software solutions we need to implement to fulfill our requirements.

1.2 References

1. IMSG. “System Requirements Specification” November 1 2023, https://docs.google.com/document/d/1LnOj2DEyu8DPbKXBTDBm2y6UbePr_AXC/edit
2. Google. “Firebase Authentication | Firebase.” Firebase, 2019, firebase.google.com/docs/auth.
3. Vercel. “Next.js by Vercel - the React Framework.” Nextjs.org, nextjs.org/.
4. https://react.dev/
5. “How to Communicate Architecture – Technical Architecture Modeling at SAP (Part 1) | SAP Blogs.” Blogs.sap.com, blogs.sap.com/2008/01/09/how-to-communicate-architecture-technical-architecture-modeling-at-sap-part-1/.
6. “Working with Related Tables.” Help.claris.com, help.claris.com/en/pro-help/content/related-tables-files.html. Accessed 15 Nov. 2023.
7. W3Schools. “SQL Data Types for MySQL, SQL Server, and MS Access.” W3schools.com, www.w3schools.com/sql/sql_datatypes.asp.

2 System Architecture

The system architecture section describes the logical structure of our software product. In the architectural design, we illustrate how the subsystems will abstractly work together and how the user interfaces with our general system. This section will guide the team in choosing solutions for our subsystems. The decomposition description further breaks down the system into functions and objects and finally illustrates our design pattern.

2.1 Architectural Design

The architectural design abstractly illustrates how the subsystems interact with each other.

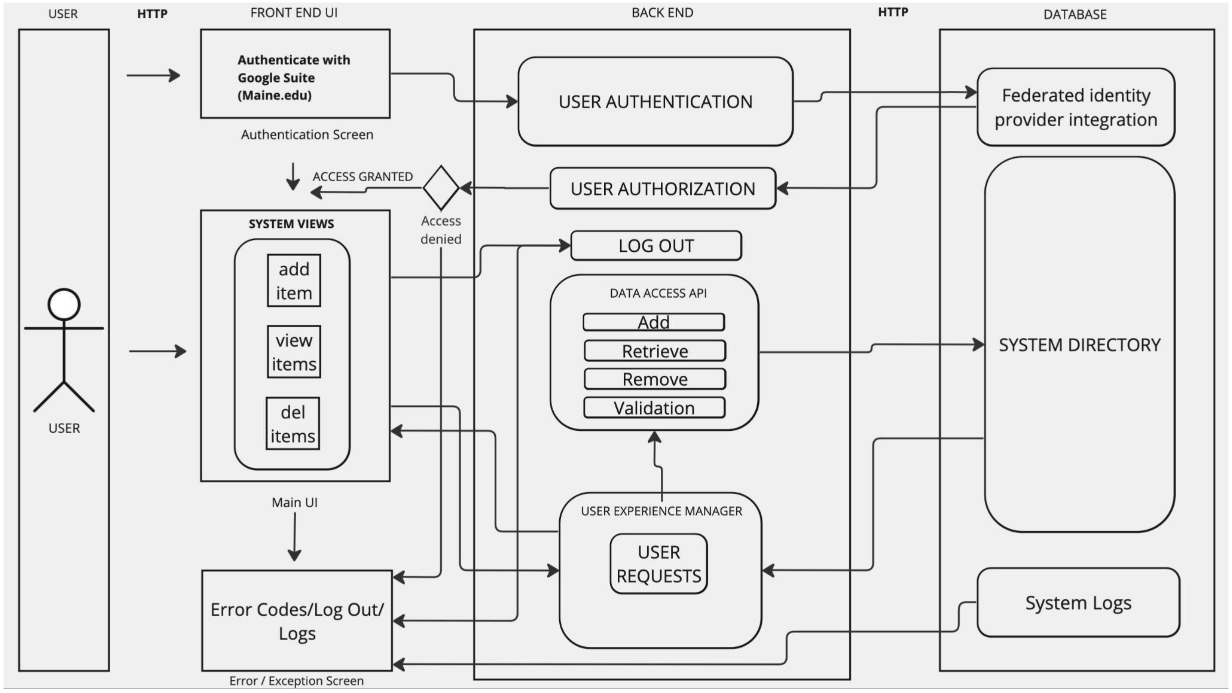


Figure 1. Architectural Design Diagram

2.1.1 Architectural Design Summary

Our system is broken down into three core components: the front end, the back end, and the database. The front end is the portion of the system that the user will see and interact with through their browser. There are three general web pages that the user will be able to see: the login page, the main user interface, and the errors and exceptions page. The user will be immediately prompted to log in on arrival to the website. The system will leverage federated identity provider integration to allow the user to utilize their University of Maine login credentials to access the system like they would with any other University of Maine service. After the authentication and authorization process, the user attains access to the rest of the user interface system. From the main UI, the user will be able to perform actions such as creating, viewing, and removing data from the database. The final interface is shown when the user logs out or reaches an error. The back-end subsystem will process user requests on the web page and handle API calls to the database. The database will be a cloud-hosted solution and will securely store all of our system data as well as manage our user authentication. Alongside storing inventory and profile data, the database will also store system logs.

The team will utilize Firebase and other Google Cloud Platform technologies (GCP) as our database and authentication management solution. Firebase is a scalable cloud-based solution owned by Google and used by leading tech companies. The team plans on using Next.JS and React to develop our back end and front end sections of our system.

The team has many options for our front/back-end solutions, but popular implementations are Next.JS / React

2.2 Decomposition Description

Our object oriented system is broken down into the following classes as shown in the diagram below.

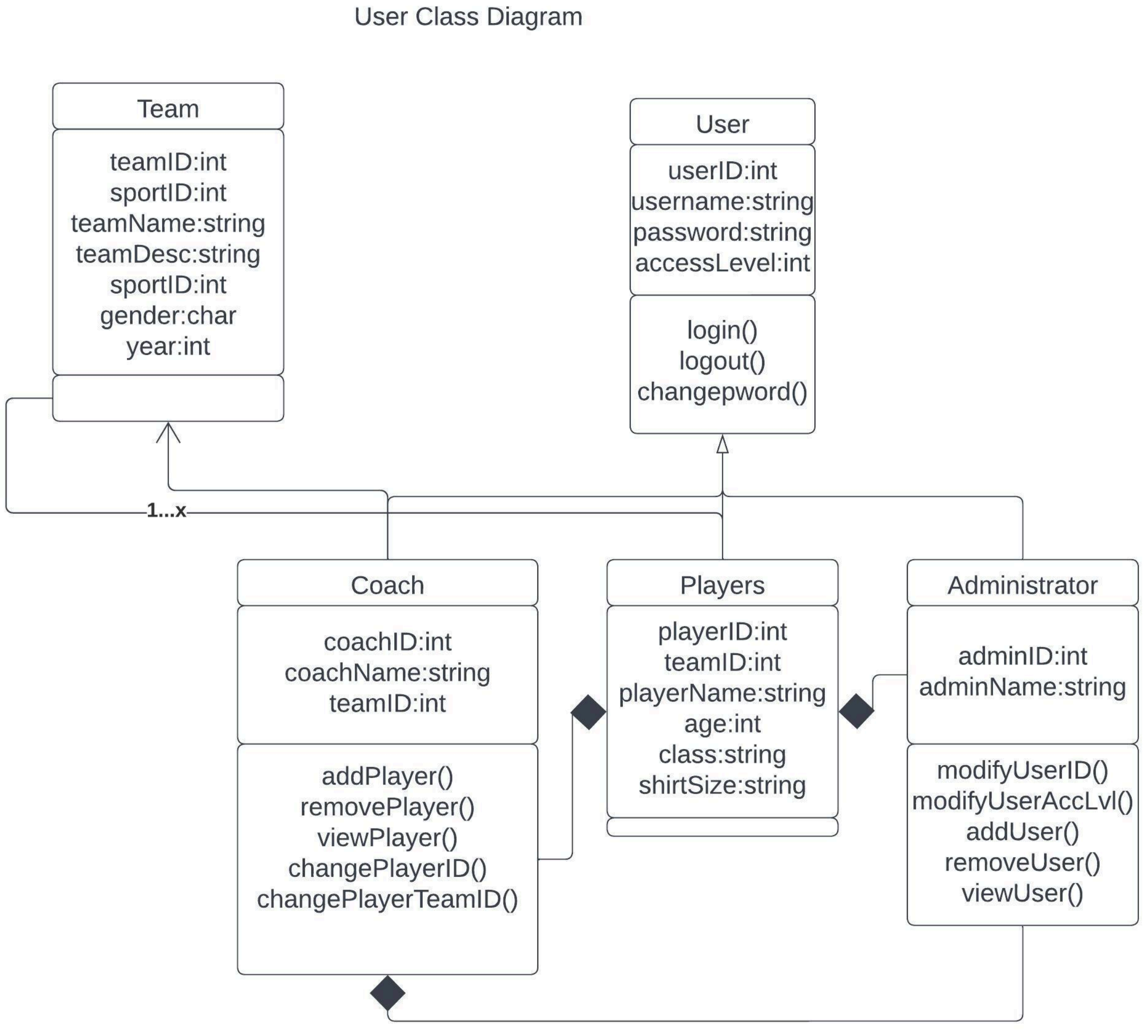


Figure 2. User Class Diagram

This class diagram shows the connection between the user and all of the subclasses that inherit the user. User being the most basic level of access, allowing for login, logout and changing of password, which will just redirect the user to a University of Maine hosted website for password resetting. The next level would be the Players, which can view their own information, as well as change some of their own attributes. Players will be able to view their coach, but not modify anything beyond their own information as allowed. Coaches inherit all of the operations and attributes of the Players, but have access to modify the players more in depth. Administrators are the next and final level of user access, which have complete control over all user accounts, and can modify anything within the scope of the system. Password resets will be hosted by the University of Maine. The Team class will be separate, but both players and coaches will be able to view the team, with coaches being able to modify.

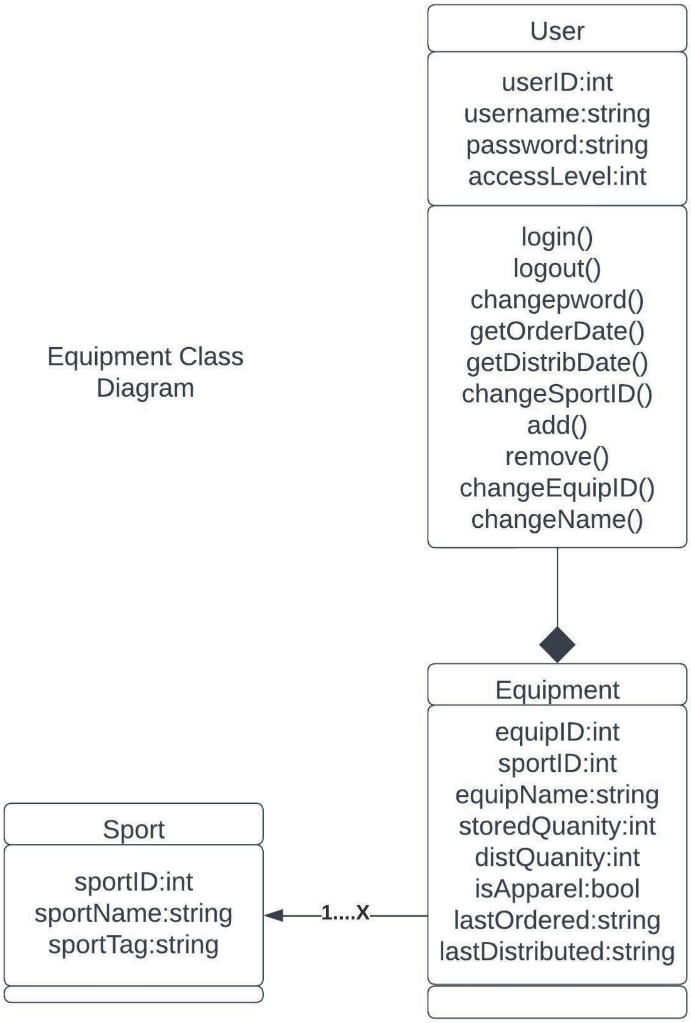


Figure 3. Equipment Class Diagram

This diagram is a little more simple, with all equipment that is listed in the databases having its own modifiers and equipment ID. The connection between the two is that there will be MANY different equipment assignments to each sport. The user is the one who owns the equipment, and is the one who will access and modify the equipment.

2.2.1 Design Pattern

We will be applying the MVC (Model View Controller) design pattern for our product. (diagram pending) We believe this has the most accurate representation of our product in the final stages. A user will interact with the controller through the UI to add, modify or remove users or items. The controller will then communicate with the model, or our database, to request the data that the user is looking for. The model, database, will then send, or deny access to the data requested. The controller will modify the data and prepare it to send to the view process, which will be what is shown to the user. The user will then see what they have done through the UI again.

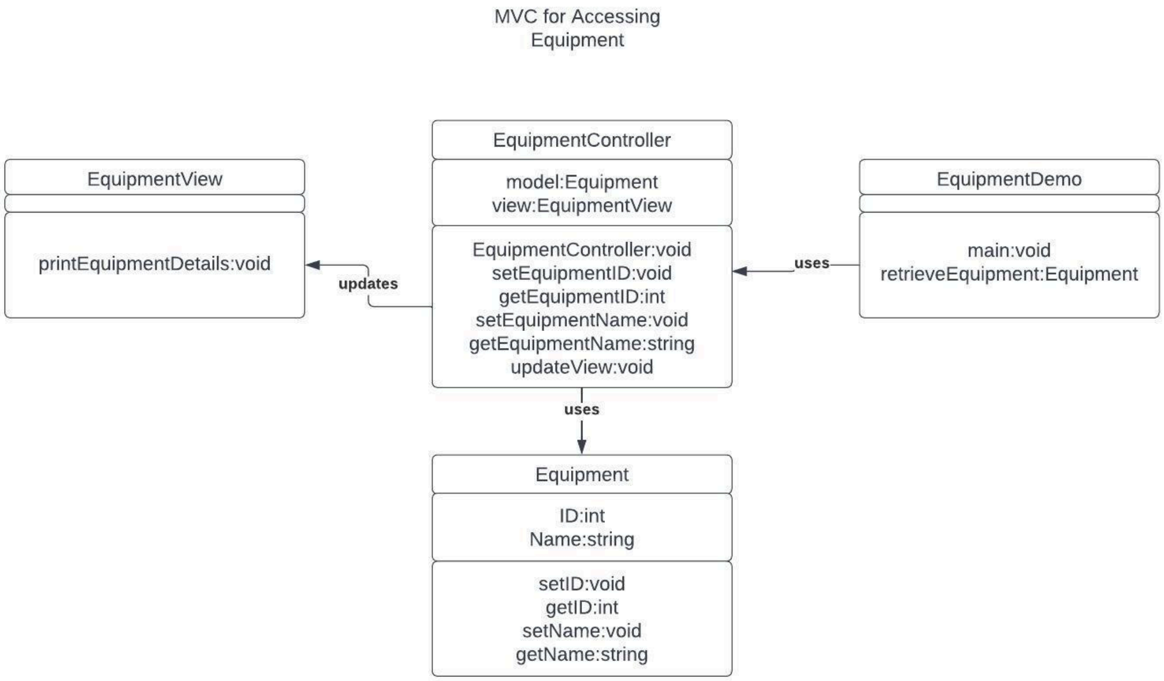


Figure 4. MVC Model for Equipment

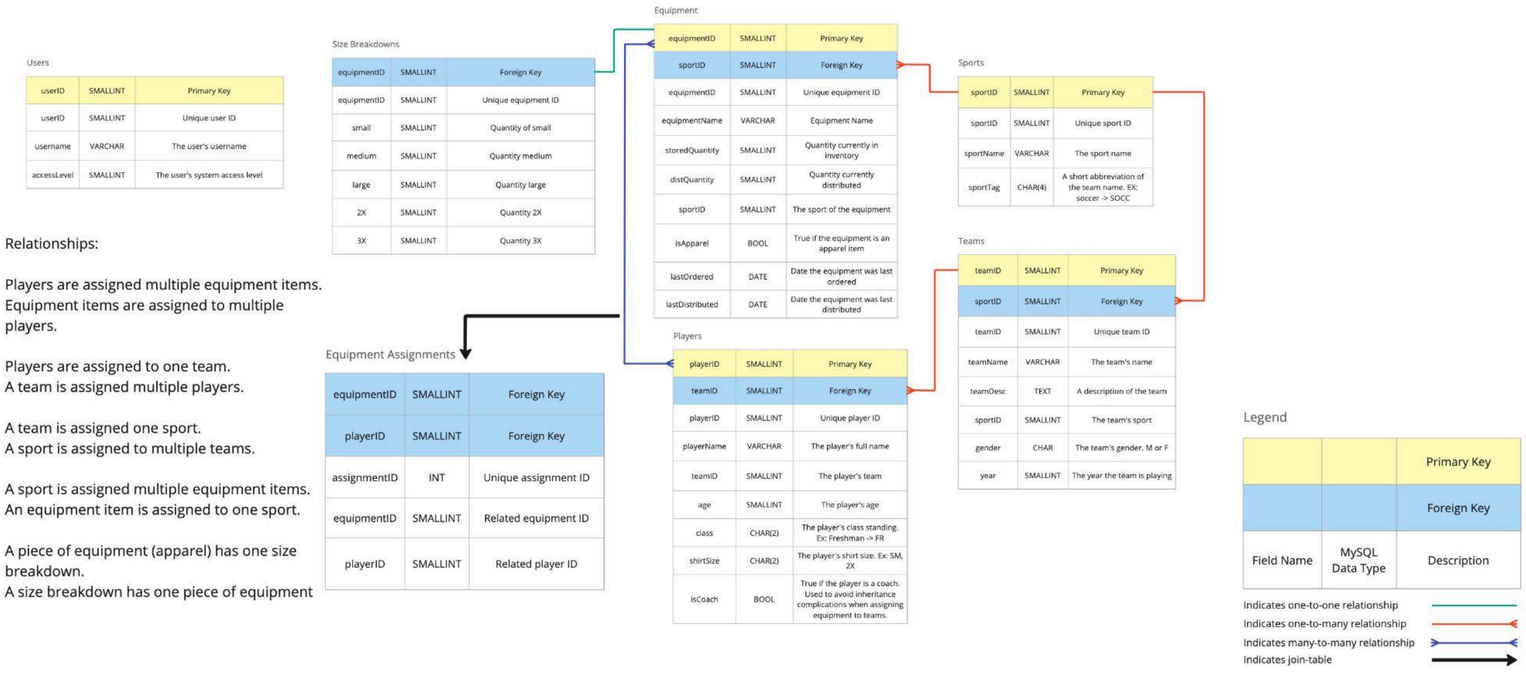
3 Persistent Data Design

The database used by the system will be a relational database using MySQL. We will be using a record-based approach that will allow us to establish relationships between the objects in our system. This will promote strong data integrity and avoid data redundancy. Following the MVC design pattern, the database serves as the Model, effectively mapping out the data flow of our application.

There is no file description section in this document because the system will be relying only on the database to keep track of the inventory. This is because the system will need a database to keep track of complex relationships between our objects in the database and the need for querying and indexing of the inventory

3.1 Database Descriptions

The database achieves this through the use of primary and foreign keys. It will allow users to establish one-to-one, one-to-many, or many-to-many relationships between Players, Teams, Sports, and Equipment. The nature of the database will allow us to perform complicated queries and joins that are necessary for tracking and filtering inventory. The database will contain indexes on fields that are a primary or foreign key, allowing faster data retrieval for commonly used queries. The database schema is shown below using an entity-relationship diagram.



Users

userID	SMALLINT	Primary Key
userID	SMALLINT	Unique user ID
username	VARCHAR	The user's username
accessLevel	SMALLINT	The user's system access level

Size Breakdowns

equipmentID	SMALLINT	Foreign Key
equipmentID	SMALLINT	Unique equipment ID
small	SMALLINT	Quantity of small
medium	SMALLINT	Quantity medium
large	SMALLINT	Quantity large
2X	SMALLINT	Quantity 2X
3X	SMALLINT	Quantity 3X

Equipment

equipmentID	SMALLINT	Primary Key
sportID	SMALLINT	Foreign Key
equipmentID	SMALLINT	Unique equipment ID
equipmentName	VARCHAR	Equipment Name
storedQuantity	SMALLINT	Quantity currently in inventory
distQuantity	SMALLINT	Quantity currently distributed
sportID	SMALLINT	The sport of the equipment
isApparel	BOOL	True if the equipment is an apparel item
lastOrdered	DATE	Date the equipment was last ordered
lastDistributed	DATE	Date the equipment was last distributed

Sports

sportID	SMALLINT	Primary Key
sportID	SMALLINT	Unique sport ID
sportName	VARCHAR	The sport name
sportTag	CHAR(4)	A short abbreviation of the team name. EX: soccer -> SOCC

Relationships:

Players are assigned multiple equipment items.
Equipment items are assigned to multiple players.

Players are assigned to one team.
A team is assigned multiple players.

A team is assigned one sport.
A sport is assigned to multiple teams.

A sport is assigned multiple equipment items.
An equipment item is assigned to one sport.

A piece of equipment (apparel) has one size breakdown.
A size breakdown has one piece of equipment

Equipment Assignments

equipmentID	SMALLINT	Foreign Key
playerID	SMALLINT	Foreign Key
assignmentID	INT	Unique assignment ID
equipmentID	SMALLINT	Related equipment ID
playerID	SMALLINT	Related player ID

Players

playerID	SMALLINT	Primary Key
teamID	SMALLINT	Foreign Key
playerID	SMALLINT	Unique player ID
playerName	VARCHAR	The player's full name
teamID	SMALLINT	The player's team
age	SMALLINT	The player's age
class	CHAR(2)	The player's class standing. Ex: Freshman -> FR
shirtSize	CHAR(2)	The player's shirt size. Ex: SM, 2X
isCoach	BOOL	True if the player is a coach. Used to avoid inheritance complications when assigning equipment to teams.

Teams

teamID	SMALLINT	Primary Key
sportID	SMALLINT	Foreign Key
teamID	SMALLINT	Unique team ID
teamName	VARCHAR	The team's name
teamDesc	TEXT	A description of the team
sportID	SMALLINT	The team's sport
gender	CHAR	The team's gender. M or F
year	SMALLINT	The year the team is playing

Legend

		Primary Key
		Foreign Key
Field Name	MySQL Data Type	Description

Indicates one-to-one relationship
Indicates one-to-many relationship
Indicates many-to-many relationship
Indicates join-table

Figure 5. Database Schema

4 Requirements Matrix

The requirements matrix is used to create a framework for what functions and methods that will be used to satisfy the functional requirements for our system. This document demonstrates the relationships between the artifacts and objects and provides an overview of the systems architecture. The requirements matrix not only brings clarity to the development process but also facilitates the actions that will sustain our application. By presenting a structured view of the relationships between the functions, methods, and requirements, this document will act as a pivotal tool between the developers and clients.

4.1 Requirements Matrix Diagrams

This table gives a framework for what functions and methods will be used for our system.

Use Case	Functional Requirement	Source	Satisfied by System Component(s)

Register	FR - 001 : Register	User	UserService.registerUser()
			checkIfExists();
			handleRegisterErrors(errorType);

Log In	FR - 002 : Log In	User	displayLoginForm();
			submitLoginCredentials(username, password);
			validateCredentials(username, password)
			createSession(user);
			handleLoginErrors(errorType);

Log Out	FR - 003 : Log Out	User	logoutUser();
			terminateSession();
			redirectToLogInPage();
			displayLogOutMessage();

Create Team	FR - 004 : Create a team	User/System	createNewTeam(teamID, teamName, teamDesc, sportID, gender, year.);
			
			checkIfTeamExists(teamID);
			If True
			displayTeamExistsMessage(errorType);
			If False
			validateTeamParameters(teamDetails);
			createTeamRecord(teamDetails);
			displayConfirmationMessage();

Create Equipment Item	FR - 005 : Create an equipment item	User/System	createEquipmentItem(equipmentID, equipmentName, storedQuantity, distQuantity, isApparel, lastOrdered, lastDistributed);
			
			checkIfItemExists(equipmentDetails);
			If True
			informItemExists(errorType);
			preventItemCreation();
			If False
			validateEquipmentParameters(equipmentDetails):
			displayConfirmationMessage();

Create New Player	FR - 006 : Create a new player	User/System	selectTeam(teamID);
			inputPlayerDetails(playerID, playerName, teamID, age, class, shirtSize, isCoach):
			checkForDuplicatePlayers(name);
			If True
			numberDuplicatePlayers(name, playerNumber);
			confirmPlayerCreation();
			If False
			confirmPlayerCreation();
			createPlayerRecord(playerDetails);

Assign an Equipment Item	FR- 007 : Assign an Equipment Item to Team	User/System	chooseTeam(teamID);
			enterDesiredAmount(amount);
			
			selectEquipmentItem(equipmentID);
			assignEquipmentToTeam(teamID, equipmentID, desiredAmount);
			
			confirmAssignment();
			checkEquipmentAvailability(equipmentID, desiredAmount);
			displayAvailabilityMessage(availableAmount);

Assign an Equipment Item	FR - 008 : Assign an Equipment Item to Player	User/System	chooseTeam(teamID);
			choosePlayer(name);
			enterDesiredQuantity(quantity);
			
			selectEquipmentItem(equipmentID);
			assignEquipmentToPlayer(name, teamID, equipmentID, desiredQuantity);
			
			confirmAssignment();
			checkEquipmentAvailability(teamID, equipmentID, desiredQuantity);
			displayAvailability(availableQuantity);

Update the details of a previously created equipment item	FR - 009 : Update Equipment Item	User/System	selectEquipmentItem(equipmentID);
			updateEquipmentItem(equipmentID, updatedDetails);
			editItemFields(updatedDetails);
			confirmUpdate();
			
			checkForDuplicateName(updatedDetails, equipmentID);
			updateDatabaseRecord(equipmentID, updatedDetails);

Update details of sports team	FR - 010 : Update Sports Team	User/System	selectTeam(teamID);
			updateTeamDetails(teamID, updatedDetails);
			editTeamFields(updatedDetails);
			confirmUpdate();
			
			checkForDuplicateParameters(updatedDetails,

			teamID);
			updateDatabaseRecord(teamID, updatedDetails);

Update details of player	FR - 011 : Update Player Info	User/System	selectTeam(teamID);
			selectPlayer(playerID);
			
			updatePlayerInfo(playerID, updatedDetails);
			editPlayerDetails(updatedDetails);
			confirmUpdate();
			updateDatabaseRecord(playerID, updatedDetails);

Remove a Team	FR - 012 : Remove Team	User/System	promptForPassword();
			confirmRemoval();
			removeTeam(teamID);
			updateDatabaseRecords(teamID);

Removing an equipment item from the entire inventory and the teams and players that were assigned the equipment.	FR - 013 : Remove Equipment Item from Inventory	User/System	selectEquipmentItem(equipmentID);
			confirmRemoval();
			removeEquipmentItemFromInventory(equipmentIID);
			unassignEquipmentFromTeams(equipmentID);
			unassignEquipmentFromPlayers(equipmentID);
			displayConfirmationMessage();
			updateDatabaseRecords(equipmentID);

Removing an equipment item from a team to which it was assigned	FR - 014 : Remove equipment item from team	User/System	selectTeam(teamID);
			selectEquipmentItem(equipmentID);
			confirmRemoval();
			removeEquipmentFromTeam(teamID, equipmentID);
			displayConfirmationMessage();
			updateDatabaseRecords(teamID, equipmentID);

Removing an equipment item from a player to which it was assigned	FR - 015 : Remove Equipment Item from Player	User/System	selectTeam(teamID);
			selectPlayer(playerID);
			selectEquipmentItem(equipmentID);
			confirmRemoval();
			removeEquipmentFromPlayer(playerID, equipmentID);
			displayConfirmationMessage();
			updateDatabaseRecords(playerID, equipmentID);

Removing a player from the team they were assigned	FR - 016 : Remove a player from team	User/System	selectTeam(teamID);
			selectPlayer(playerID);
			confirmRemoval();
			removePlayerFromTeam(playerID, teamID);
			displayConfirmationMessage();
			updateDatabaseRecords(playerID, teamID);

The user can view all equipment item details.	FR - 017 : Viewing Equipment Item	User/System	selectEquipmentItem(equipmentID);
			getEquipmentItemDetails(equipmentID);
			displayEquipmentItemDetails(equipmentID);

View players listed under a teams roster	FR - 018 : Viewing a teams roster	User/System	selectTeam(teamID);
			getTeamRoster(teamID);
			displayTeamRoster(rosterData);

Viewing a teams assigned equipment	FR - 019 : Viewing a teams assigned equipment	User/System	selectTeam(teamID);
			getAssignedEquipment(teamID);
			displayAssignedEquipment(equipmentData);

View the details associated with a player	FR - 020 : Viewing a player	User/System	selectTeam(teamID);
			selectPlayer(playerID);
			getPlayerDetails(playerID);
			displayPlayerDetails(playerData);

The user can view all the items in the inventory room	FR - 021 : Viewing general inventory	User/System	displayGeneralInventory();
			retrieveAllEquipmentItems();
			filterInventoryByCategory(category);
			
			Category can either be TeamID or PlayerID

The user can filter the general inventory by name, team equipment, and type	FR - 022 : Filter general inventory	User/System	displayEquipmentInventory();
			applyFilters(filters);
			retrieveFilteredEquipment(filters);
			updateInventoryDisplay(filteredInventory);
			
			Filter can be by name, team equipment, or type

The user can view all previous inventory transactions	FR - 023 : Viewing inventory history	User/System	displayInventoryHistory();
			retrieveInventoryTransactions():
			filterByActionType(actionType):
			
			Action types include create, assign, update, delete

The user can filter the inventory history to search for a specific transaction	FR - 024 : Filtering inventory history	User/System	displayInventoryHistory();
			applyFilters(filters);
			retrieveFilteredHistory(filters);
			updateHistoryDisplay(filteredHistory);
			
			Filters include specific items, teams, players, date range

Figure 6. Requirements Matrix Diagram

SDD - Appendix A – Agreement Between Customer and Contractor

Agreement Between Customer and Contractor

1. Parties: This agreement made on "11/15/2023" is by and between

Client: University of Maine Athletic Department

AND

Contractor: Inventory Management Software Group

2. Term: The terms of this agreement shall commence on November 5th, 2023, and conclude on May, 2024.

3. Services: The Contractor agrees to provide the following services for the betterment of the Customer: The Contractor will create an inventory management system that allows the Customer to visualize and manage their inventory through an online webpage. Further details are presented in the System Design Document.

4. Expenses: There are no initial expenses. When expenses are incurred, the Contractor will communicate and get approval from the Customer to use funds allotted to them if available.

5. Agreement:
By signing this document, all parties agree to the requirements presented in this document. All parties also agree that the deadlines presented are tentative, and are subject to change as the program is developed.

Customer Signature:		
	Date:	Printed:
X: *Kevin Ritz*	11/15/2023	Kevin Ritz
Contractor Signature:	Date:	Printed:
X: *Collin Rodrigue*	11/15/2023	Collin Rodrigue
X: *Gabriel A. Poulin*	11/15/2023	Gabriel A. Poulin
X: *Brennan Poitras*	11/15/2023	Brennan Poitras
X: *Sean Radel*	11/15/2023	Sean Radel
X:*Graham Bridges*	11/15/2023	Graham Bridges

SDD - Appendix B – Team Review Sign-off

Team Agreement Sign Off

Team IMSG has thoroughly reviewed the System Design Document for the Athletic Inventory System and has agreed that the following information is accurate and achievable. Collectively we have no major contentions in the information stated in the document. By signing this agreement, one acknowledges all the terms and conditions outlined in the document and understands the importance of effective team collaboration, communication, and shared accountability when achieving the goals of the project. By signing below, we pledge our dedication to the success of the team and the project we plan to undertake. We agree to work collaboratively, and support each other to uphold the guidelines and expectations set forth in the agreement.

Signature:	Date:	Printed:
X: *Collin Rodrigue*	11/15/2023	Collin Rodrigue
X: *Gabriel A. Poulin*	11/15/2023	Gabriel A. Poulin
X: *Brennan Poitras*	11/15/2023	Brennan Poitras
X: *Sean Radel*	11/15/2023	Sean Radel
X:*Graham Bridges*	11/15/2023	Graham Bridges

SDD - Appendix C – Document Contributions

Name	**Date**	**Contribution**	**Version**
Sean Radel	11/2/23	Formatting, introduction	0.1
Sean Radel	11/3/23	Architectural Design Section	0.1
Brennan Poitras	11/6/23	Database Description/Schema	0.1
Collin Rodrigue	11/6/23	Requirements Matrix	0.1
Gabriel Poulin	11/6/23	UML Diagrams and Design Pattern	0.1
Sean Radel	11/14/23	Fixed Architectural Design Section and formatting	0.2
Gabriel Poulin	11/14/23	Appendix A and Appendix B	0.2
Brennan Poitras	11/14/23	Additions to the DB schema	0.2

Appendix F – User Interface Design Document

COS 397: Computer Science Capstone I

User Interface Design Document

UMaine Athletic Department Inventory Management System

IMSG

Version 0.2

Collin Rodrigue, Brennan Poitras, Graham Bridges, Gabe Poulin, Sean Radel

4 December 2023

1. Introduction

The purpose of the product is to fulfill the customer needs of an inventory management solution. Our system will replace the customer's previous solutions for managing their equipment. The customer previously used Front Rush and currently uses a combination of Excel spreadsheets and word of mouth to track inventory. Our product will allow the customer to organize their inventory by associating equipment with teams and players. The system will be designed with simplicity in mind so that they do not need experienced developers to maintain their product following the delivery date. Equipment, player accounts, and teams will be able to be made on demand to allow the system to scale to the customer's needs. Our customer is the University of Maine Athletic Department, and specifically Jude Killy, Nick Fox, and Kevin Ritz.

1.1 Purpose of This Document

The purpose of this document is to illustrate the design choices and standards the team will use when developing the user interface for our product. The team will outline the accessibility and inclusion design choices to ensure we develop a fair and non-discriminatory product. This document will walk the reader through our user interface concept and display the webpages we are planning on building. The team will also document how we validate the data that users will input into the system. Finally, we document how the user can print an inventory report, and what that will look like.

1.2 References

1. Google Sheets. *Athletic Inventory*. University of Maine Athletic Department, (non-acessable to public), 15 Nov. 2023

2. IMSG. “System Requirements Specification” November 1 2023, https://docs.google.com/document/d/1LnOj2DEyu8DPbKXBTDBm2y6UbePr_AXC/edit

3. IMSG. “System Design Document” November 15 2023, https://docs.google.com/document/d/1_dkbb3zXQzOxVbdIr3nbizQc0r-f6yC2/edit?usp=sharing&ouid=105664753662702248466&rtpof=true&sd=true

2. User Interface Standards

This section will go into the User Interface Standards, which are the guidelines and principles used in the inventory system to establish a consistent and accessible user experience throughout the system. It shall serve as a reference for developers should there be a need for modification or additions to the inventory system.

2.1 Design Standards

The design standards are consistent elements used in the user interface that will be present on most if not all pages of the system. It describes the general layout of each page of the inventory and general navigation and components.

At the top of every page will be a blue navigation bar. It will include the name of the page the user is currently viewing, which will generally state what the current page's functionality is. This name will be the only item on the navigation bar before the user is logged in, as shown below:

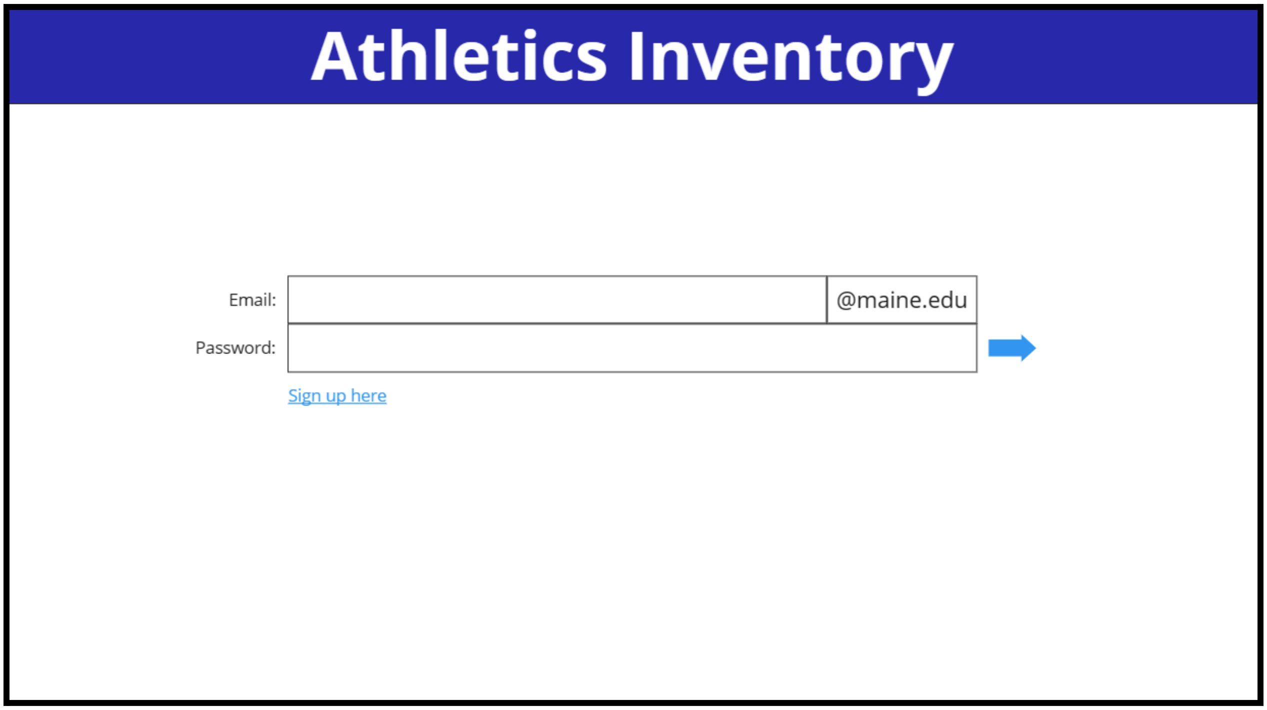


Figure 2.1

Once the user is logged in, there will be three more buttons on the navigation bar. The first will be a home button at the top left which will take the user to the home page shown below:

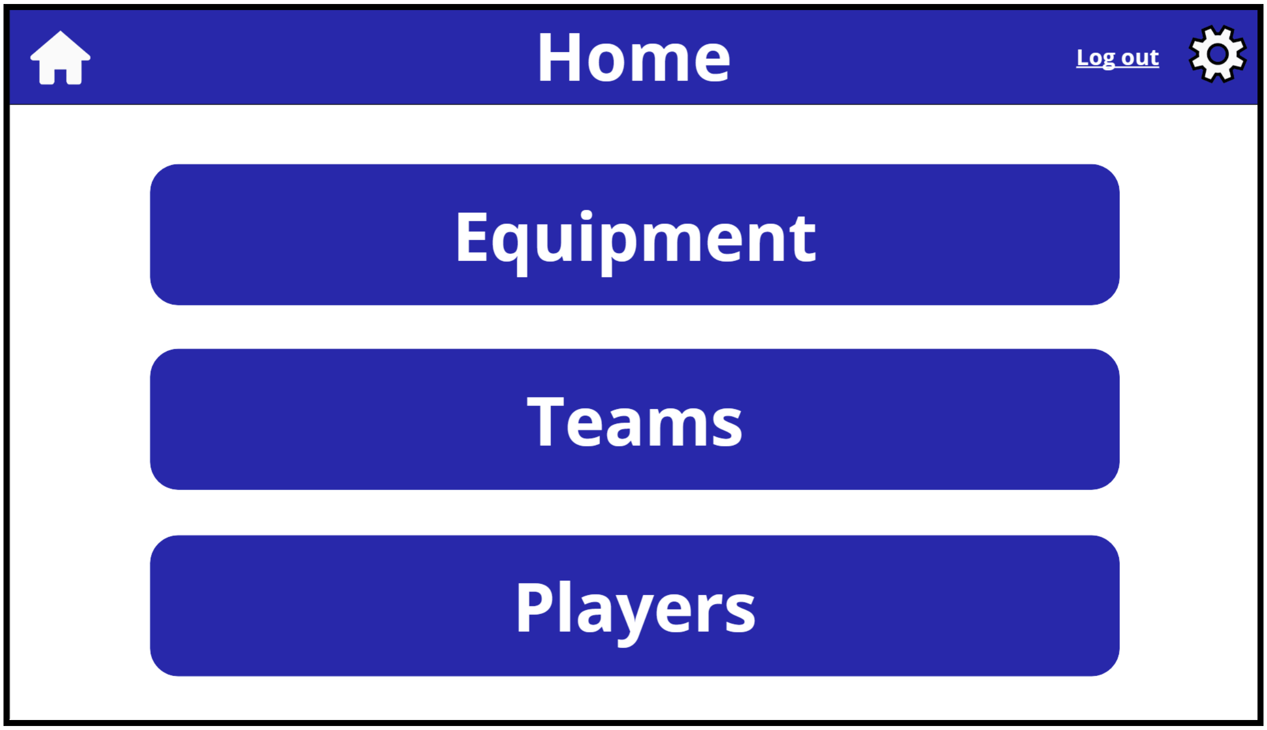
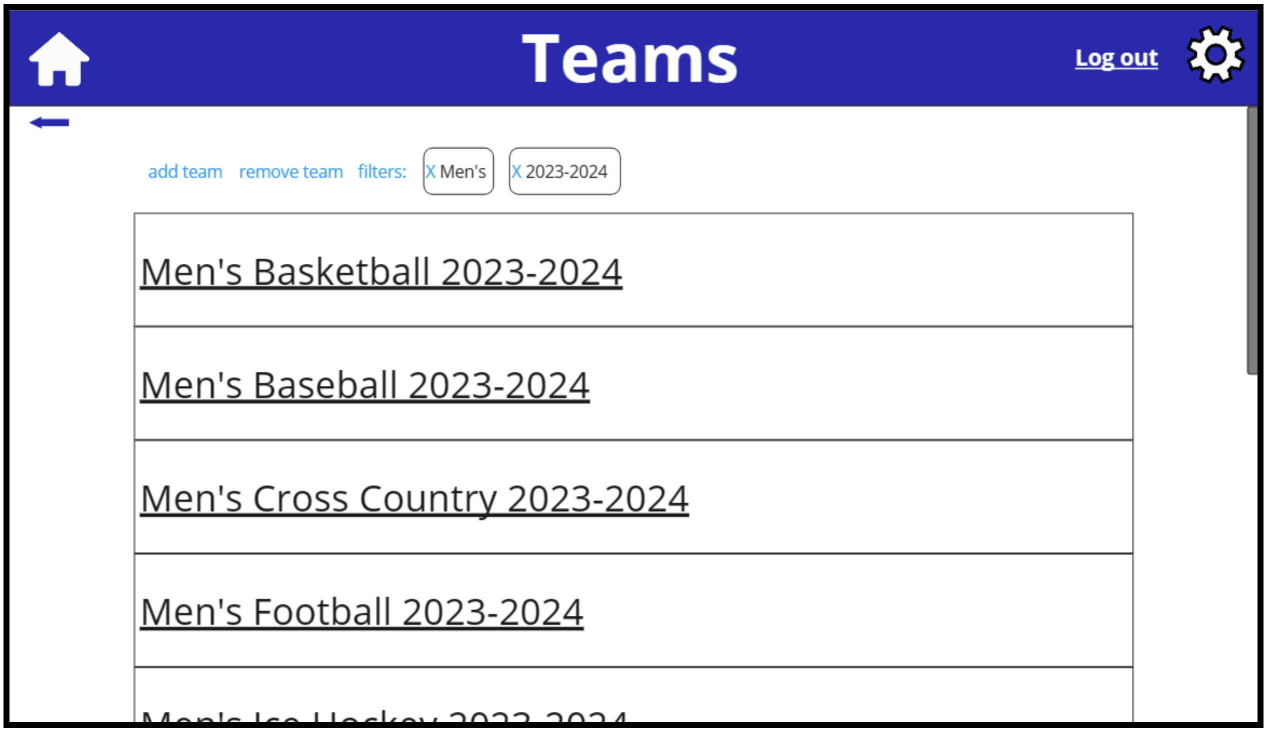


Figure 2.2

The second button is a logout function that will take the user to a screen confirming if they want to log out of their account. Clicking confirm will log out the user and take them back to the login page. The third button is a gear cog logo that will bring the user to the customizable settings page.

Once the user is on a page other than the login or home screen, an arrow will appear on the top right corner of the screen below the navigation bar. If clicked, this arrow will take the user to the page they were previously on. For example, if a user first clicks on the teams button on the home page, then clicks on a specific team to go to that team's page, if they then click the arrow in the top right the system will bring the user back to the page listing every team. The arrow can be seen below on the teams page mock up:



Men's Basketball 2023-2024

Men's Baseball 2023-2024

Men's Cross Country 2023-2024

Men's Football 2023-2024

Men's Ice Hockey 2023-2024

Figure 2.3

2.2 Accessibility and Inclusive Design Elements

Developing an inclusive and accessible product is a focus for the team so that we can create a usable product that encourages our stakeholders and end users to use the platform. To increase the accessibility of our product, we will ensure that we use high-contrast color schemes to increase visibility, include alt-text in all images for users with screen readers, allow the website to be navigated using arrow keys for users who can not use a mouse, and include captions in any videos we may implement. We will ensure that we do not convey information through color alone to increase the understanding for users with screen readers and color blindness. To increase the inclusivity of our system, we will allow user profiles to enter whichever pronouns they feel comfortable with. We will aim to use racially neutral icons that don't discriminate against any particular identity. Our aim is to design a product that is both welcoming and usable for all communities.

Case	Design Element
Visual Impairment	- High Contrast Color Scheme - Support for Screen Readers - Convey information using multiple mediums (symbols, text, and color)
Physical Impairment	- Utilize arrow keys for movement

	around the screen
Diversity and Inclusion	- Utilize gender and racially neutral icons - Pronoun support

Figure 2.4. Inclusive Design Elements

3. User Interface Walkthrough

The User Interface Walkthrough will describe how a user will navigate to and from each screen of the system. Any components discussed in Section 2.1 will be ignored in the walkthrough and the following explanation. Below is a navigation diagram of the inventory system site:

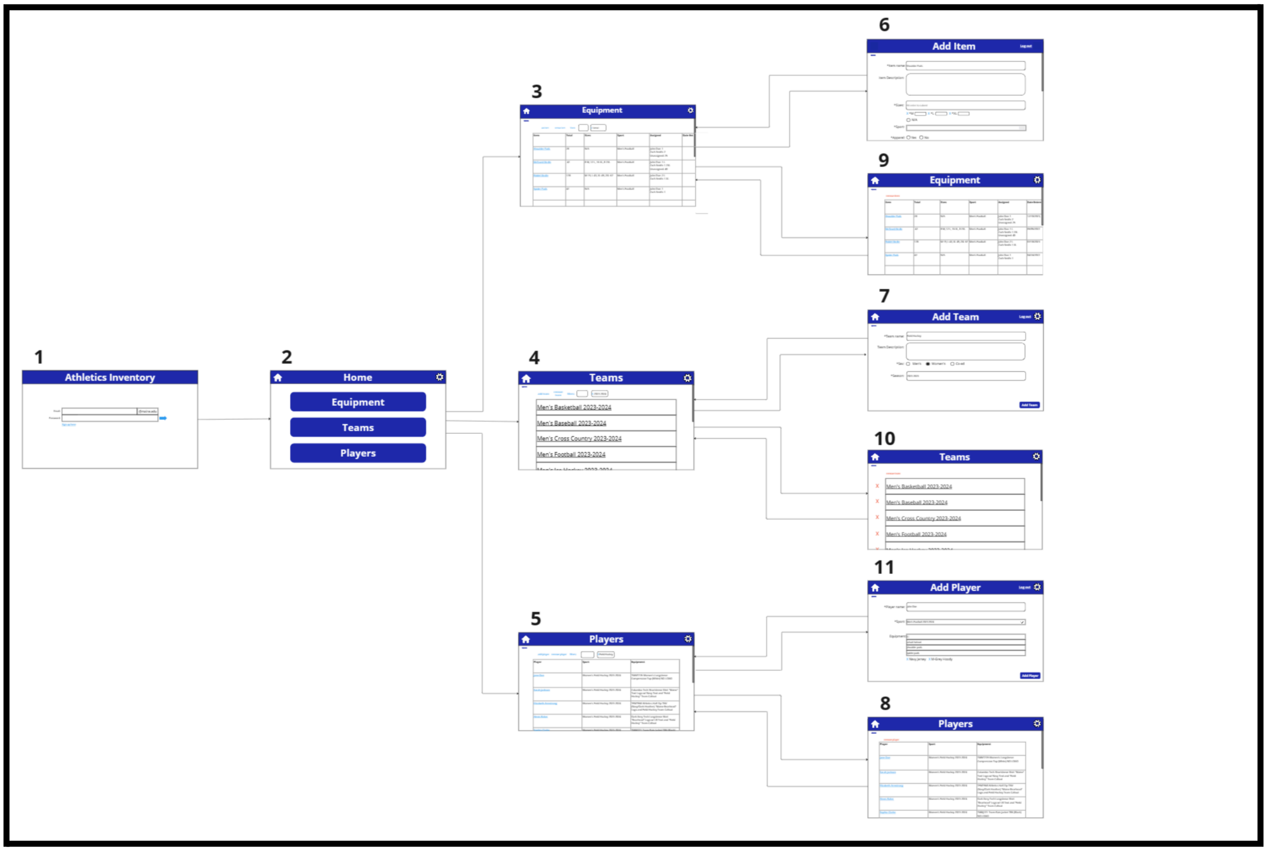


Figure 3. Navigation of Inventory System

The user will begin on Figure 3.1, the login screen:

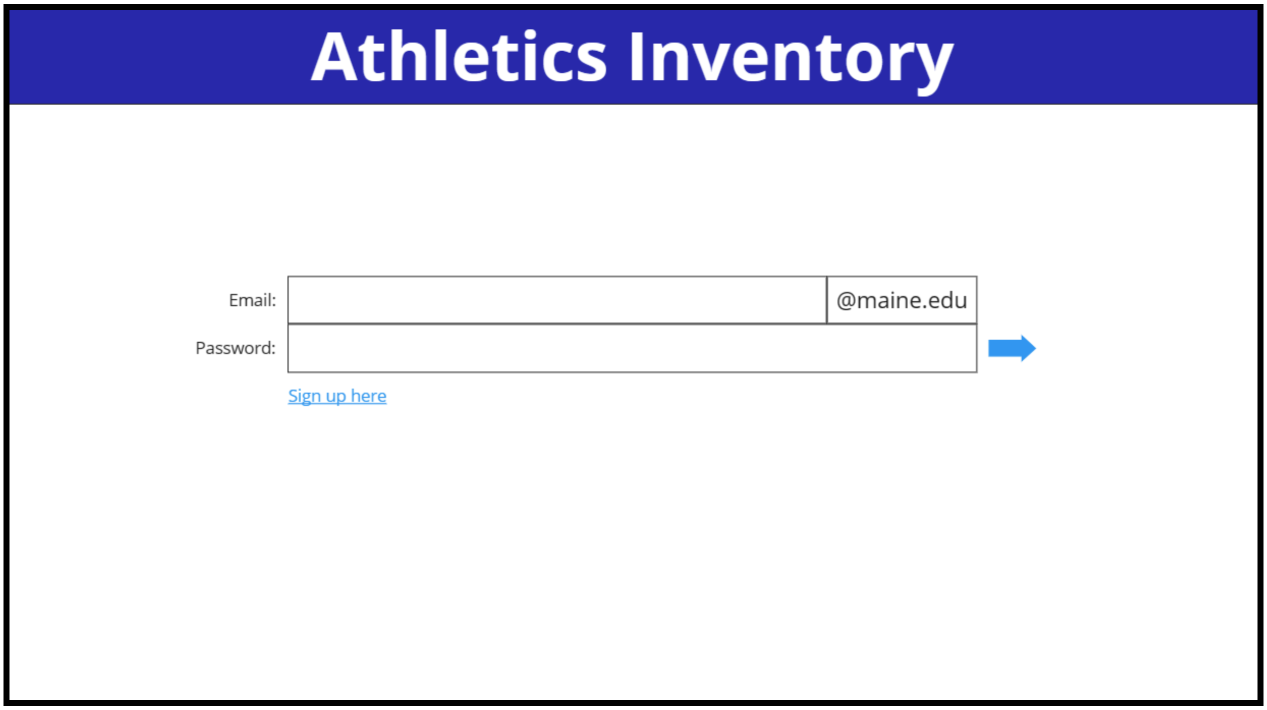


Figure 3.1. Login Screen

The user will enter their maine.edu email and password used for their University of Maine portal account. They will then hit enter or click the blue arrow to continue to the rest of the site. If they are not signed up, they can click "sign up here" and enter their maine.edu email and password to request access to the inventory from an administrator. Once logged in, the user will be presented with Figure 3.2:

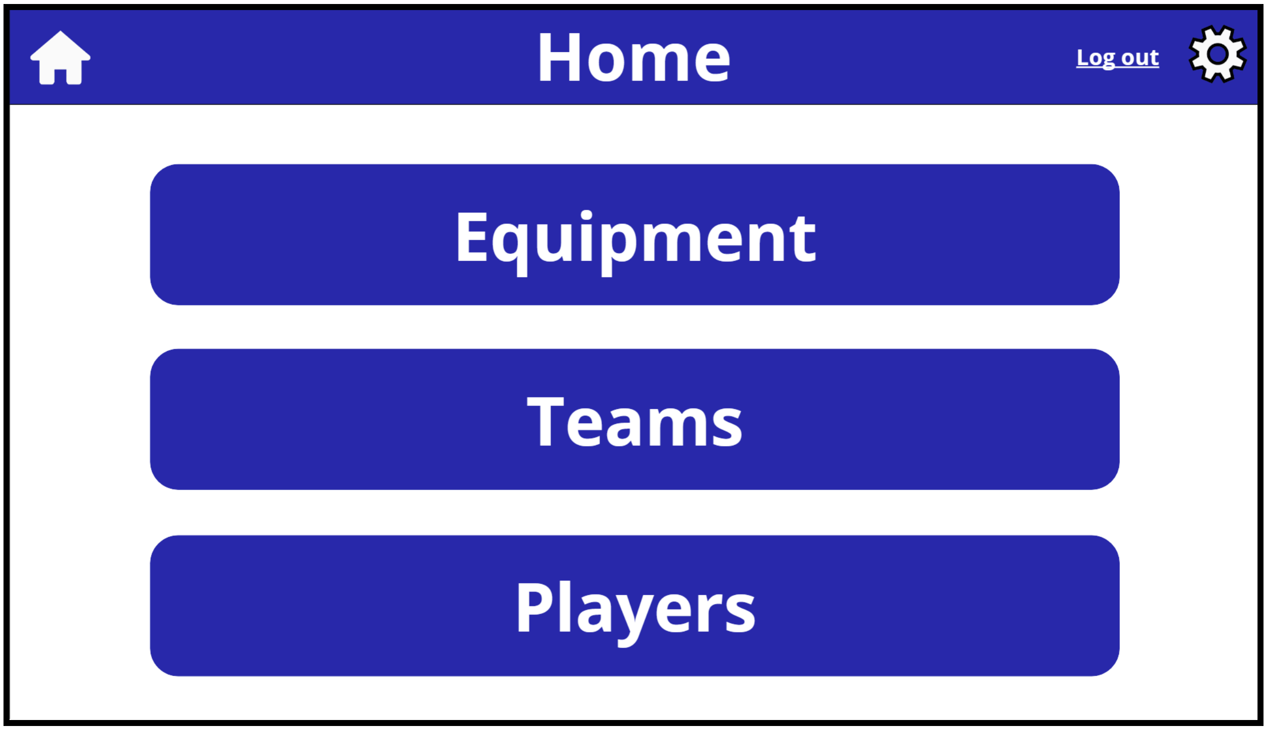
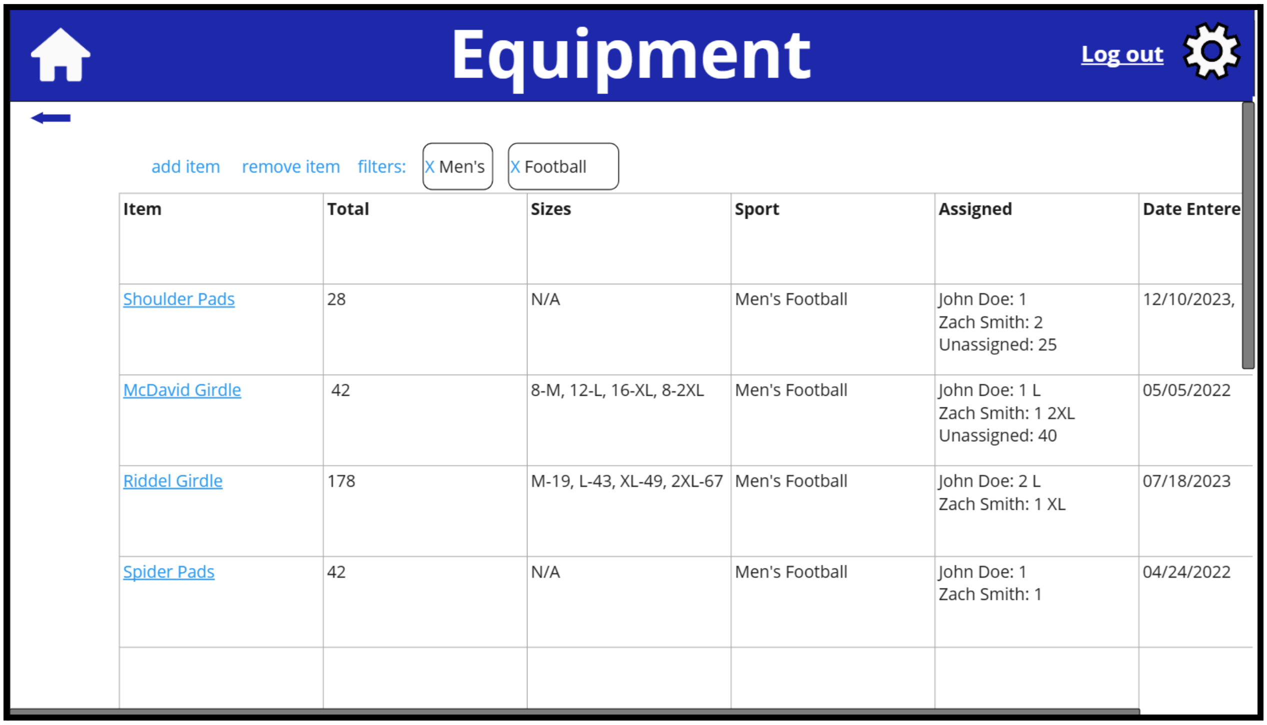


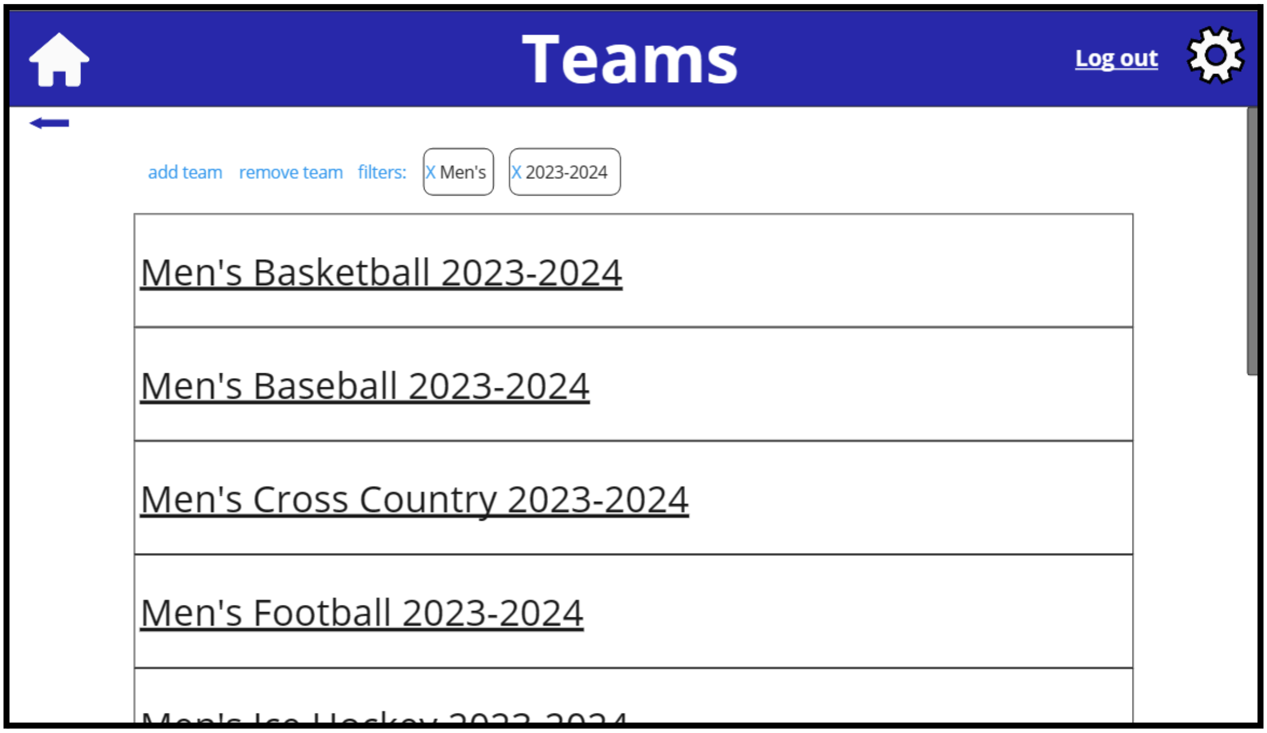
Figure 3.2. Home Screen

On this screen, the user can click on the equipment, team, player button to navigate to the given type of object's view screen. These are Figures 3.3, 3.4, and 3.5:



Item	Total	Sizes	Sport	Assigned	Date Entere
Shoulder Pads	28	N/A	Men's Football	John Doe: 1 Zach Smith: 2 Unassigned: 25	12/10/2023,
McDavid Girdle	42	8-M, 12-L, 16-XL, 8-2XL	Men's Football	John Doe: 1 L Zach Smith: 1 2XL Unassigned: 40	05/05/2022
Riddel Girdle	178	M-19, L-43, XL-49, 2XL-67	Men's Football	John Doe: 2 L Zach Smith: 1 XL	07/18/2023
Spider Pads	42	N/A	Men's Football	John Doe: 1 Zach Smith: 1	04/24/2022

Figure 3.3. Equipment View Screen



Men's Basketball 2023-2024

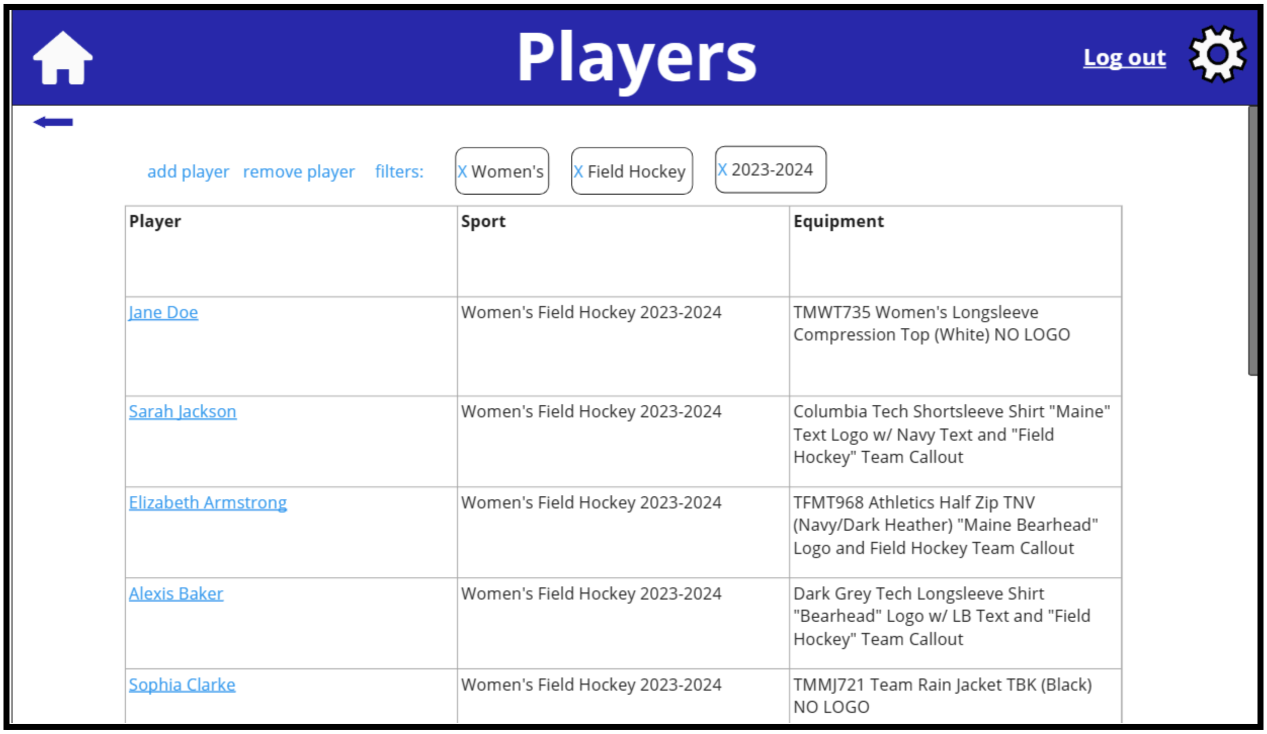
Men's Baseball 2023-2024

Men's Cross Country 2023-2024

Men's Football 2023-2024

Men's Ice Hockey 2023-2024

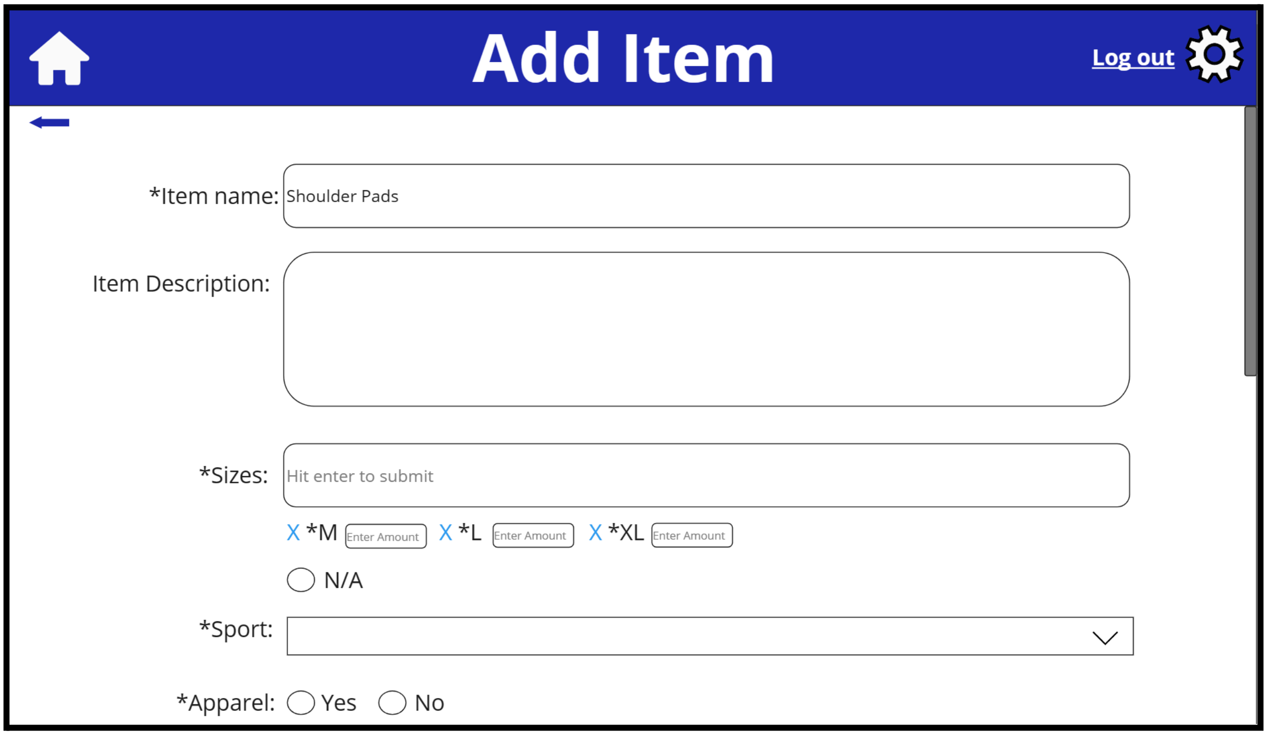
Figure 3.4. Team View Screen



Player	Sport	Equipment
Jane Doe	Women's Field Hockey 2023-2024	TMWT735 Women's Longsleeve Compression Top (White) NO LOGO
Sarah Jackson	Women's Field Hockey 2023-2024	Columbia Tech Shortsleeve Shirt "Maine" Text Logo w/ Navy Text and "Field Hockey" Team Callout
Elizabeth Armstrong	Women's Field Hockey 2023-2024	TFMT968 Athletics Half Zip TNV (Navy/Dark Heather) "Maine Bearhead" Logo and Field Hockey Team Callout
Alexis Baker	Women's Field Hockey 2023-2024	Dark Grey Tech Longsleeve Shirt "Bearhead" Logo w/ LB Text and "Field Hockey" Team Callout
Sophia Clarke	Women's Field Hockey 2023-2024	TMMJ721 Team Rain Jacket TBK (Black) NO LOGO

Figure 3.5. Player View Screen

Clicking “add item”, “add team”, or “add player” will bring the user to a screen where they can add an item, team, or player respectively. Clicking on the name of an item, team, or player will bring the user to a page where they can edit the given item, team, or player; these pages will be similar to their “add item”, “add team”, and “add player” counterparts. Clicking “filters” will reveal a pop-up where the user can add features for the system to filter objects in the database for. Hitting the “X” next to any of the filters will remove that filter from the search. When the user clicks either “remove item”, “remove team”, or “remove player” the user will be taken to a page similar to screenshots Figures 3.3, 3.4, and 3.5 where they can remove a given item, team, or player from the database entirely. See figures 3.9, 3.10, 3.11 below. The screenshots below are the pages where users can add equipment, teams, and players to the database:



*Item name: Shoulder Pads

Item Description:

*Sizes: Hit enter to submit

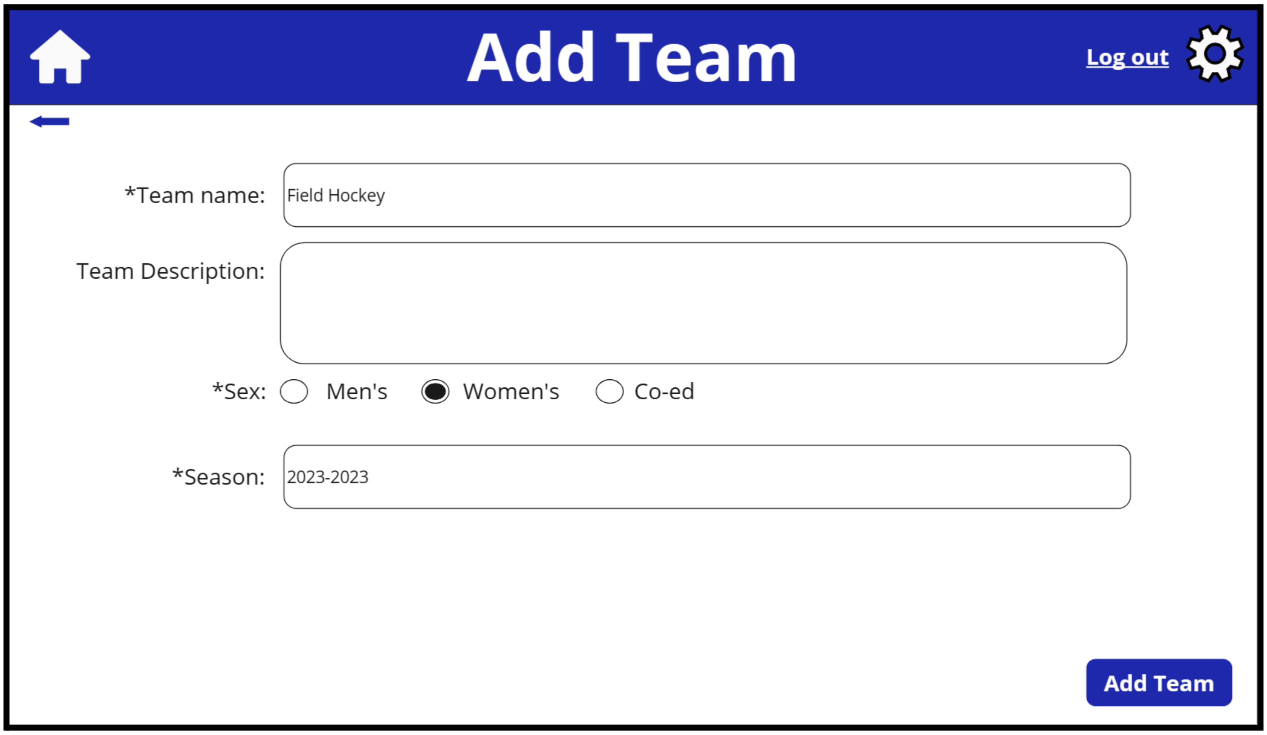
X *M Enter Amount X *L Enter Amount X *XL Enter Amount

○ N/A

*Sport:

*Apparel: ○ Yes ○ No

Figure 3.6. Add Equipment Screen



Add Team

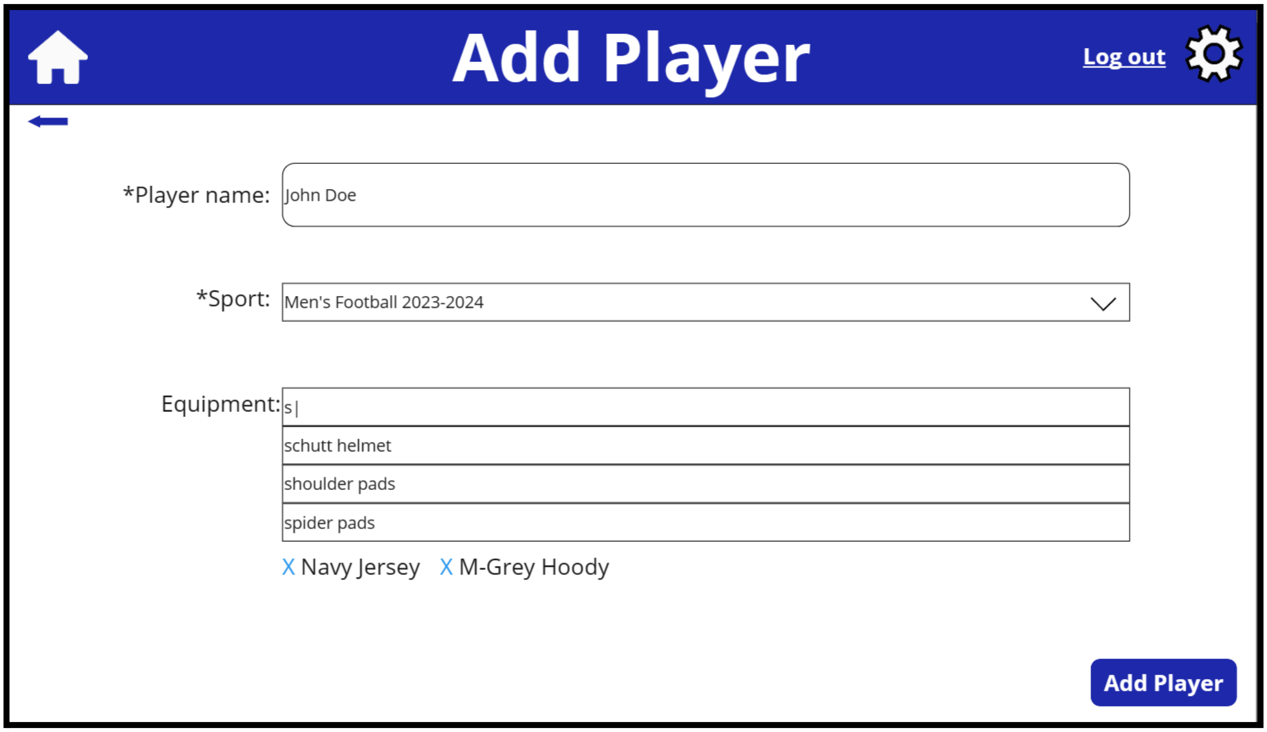
*Team name: Field Hockey

Team Description:

*Sex: ○ Men's ● Women's ○ Co-ed

*Season: 2023-2023

Figure 3.7. Add Team Screen



Add Player

*Player name: John Doe

*Sport: Men's Football 2023-2024

Equipment: s|

schutt helmet

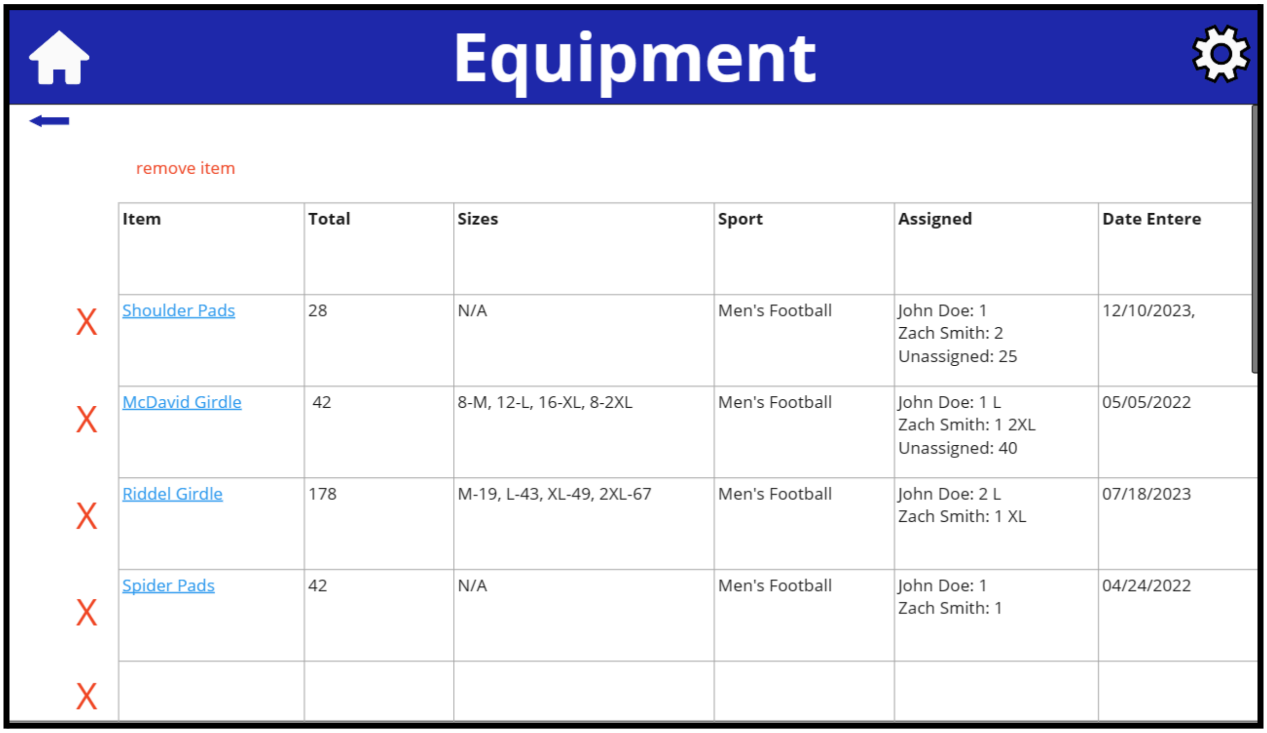
shoulder pads

spider pads

X Navy Jersey X M-Grey Hoody

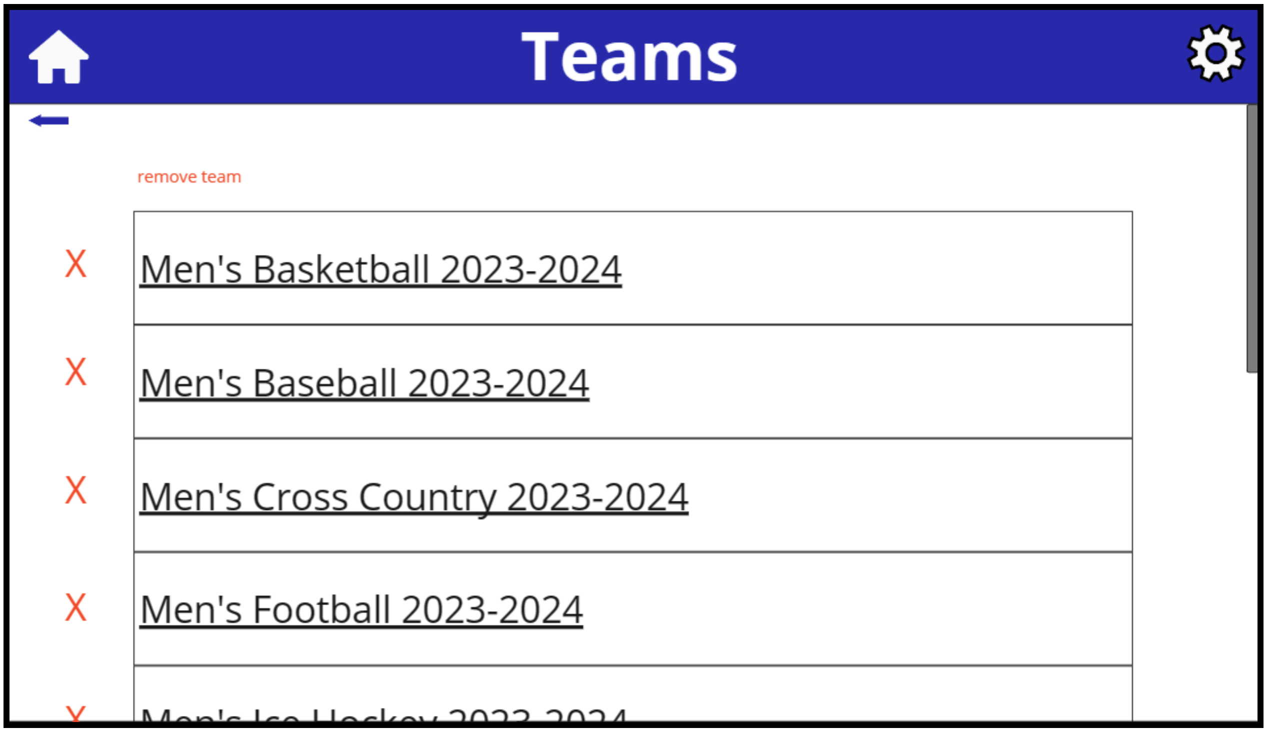
Figure 3.8. Add Player Screen

On the pages depicted by Figures 3.6, 3.7, and 3.8 users will fill out fields for the given object they wish to enter into the database. A "*" next to a field's name means it's a required field. For a type of equipment, the user must enter the item's name, the sizes if there are multiple of them(if not N/A), the sport the equipment is associated with, and the total quantity and quantity of each size. An item description is optional. When adding a team, the user must add a team name, the sex of the team, and the season for the given team. When adding a player, the user must add a name for the player, the sport the player is participating in, and the equipment assigned to the player if there is any. The user can remove assigned equipment from the player by clicking the blue "X" next to it. Once the user is done filling out the field for the given object, they will click the blue "Add" button at the bottom right of the page. The user will then be redirected to the viewing screens in Figures 3.3, 3.4, and 3.5 with the new object added to the database. The screenshots below show the pages the user is directed to when they select either, "remove item", "remove team", and "remove player":



	Item	Total	Sizes	Sport	Assigned	Date Entere
X	Shoulder Pads	28	N/A	Men's Football	John Doe: 1 Zach Smith: 2 Unassigned: 25	12/10/2023,
X	McDavid Girdle	42	8-M, 12-L, 16-XL, 8-2XL	Men's Football	John Doe: 1 L Zach Smith: 1 2XL Unassigned: 40	05/05/2022
X	Riddel Girdle	178	M-19, L-43, XL-49, 2XL-67	Men's Football	John Doe: 2 L Zach Smith: 1 XL	07/18/2023
X	Spider Pads	42	N/A	Men's Football	John Doe: 1 Zach Smith: 1	04/24/2022
X						

Figure 3.9. Remove Equipment Screen



Teams

remove team

X Men's Basketball 2023-2024

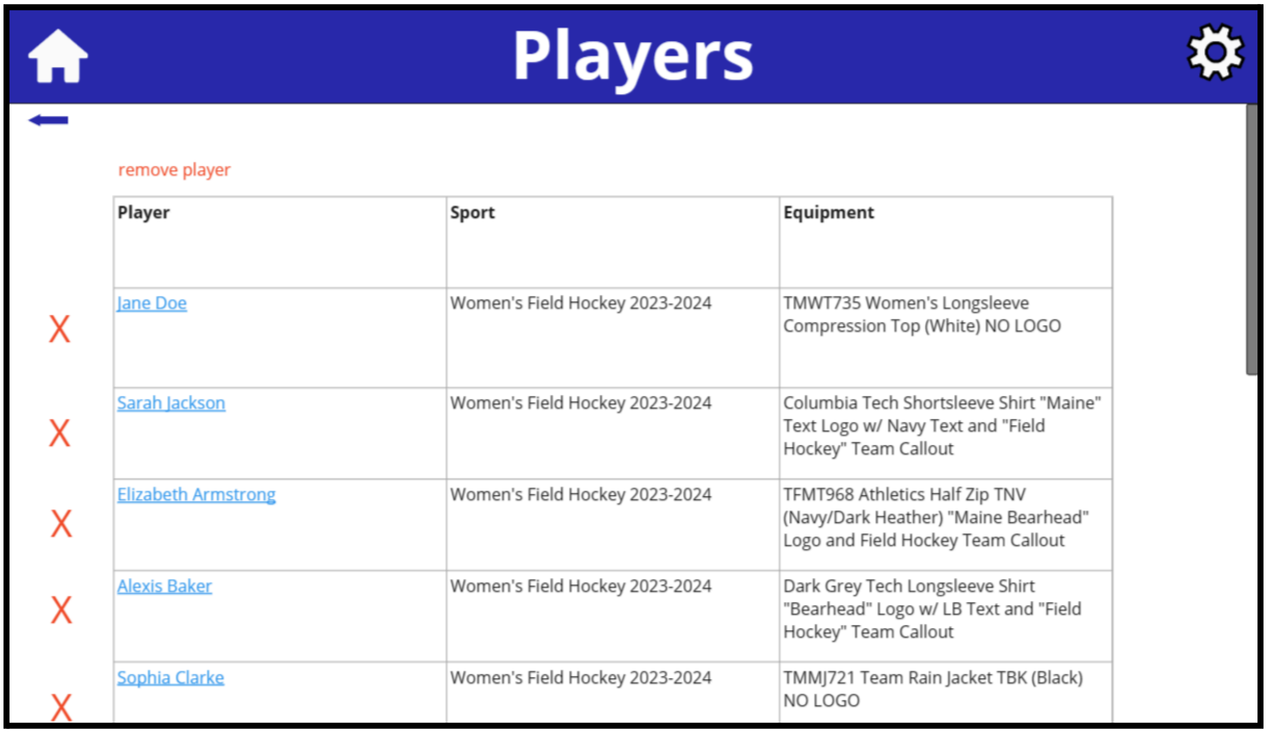
X Men's Baseball 2023-2024

X Men's Cross Country 2023-2024

X Men's Football 2023-2024

X Men's Ice Hockey 2023-2024

Figure 3.10. Remove Team Screen



Players

remove player

	Player	Sport	Equipment
X	Jane Doe	Women's Field Hockey 2023-2024	TMWT735 Women's Longsleeve Compression Top (White) NO LOGO
X	Sarah Jackson	Women's Field Hockey 2023-2024	Columbia Tech Shortsleeve Shirt "Maine" Text Logo w/ Navy Text and "Field Hockey" Team Callout
X	Elizabeth Armstrong	Women's Field Hockey 2023-2024	TFMT968 Athletics Half Zip TNV (Navy/Dark Heather) "Maine Bearhead" Logo and Field Hockey Team Callout
X	Alexis Baker	Women's Field Hockey 2023-2024	Dark Grey Tech Longsleeve Shirt "Bearhead" Logo w/ LB Text and "Field Hockey" Team Callout
X	Sophia Clarke	Women's Field Hockey 2023-2024	TMMJ721 Team Rain Jacket TBK (Black) NO LOGO

Figure 3.11. Remove Player Screen

On the pages depicted by Figures 3.9, 3.10, and 3.11, the user will decide what object to remove from the database by clicking on the red “X” next to its name. The system will then provide a pop up window that will ask the user to confirm they want to delete the selected object. Once they hit confirm, the user will be redirected to the viewing screen with the selected item removed. If the user decides they don't want to remove anything, they can click the red "remove" button to be redirected back to the viewing screen. They could also click the arrow in the top left corner, as described in section 2.1. If an equipment item is removed, any player it is assigned to will be removed from that player. If a team is removed, any players and equipment associated with that team will be removed. If a player is removed, any equipment assigned to that player will be unassigned.

4. Data Validation

This section outlines Data Validation, a crucial step in maintaining an accurate and functioning system. It clearly defines user-input data items, including data types, limits, and allowed formats. The information is presented in a tabular format for easy comprehension and accessibility.

Screen Name	**Data Type**	**Description**	**Limits**	**Format**
Equipment Name	string	The equipment's name	N/A	N/A
Size totals	Object array	array of objects where each object has a string "size" and an integer "quantity" stating the amount in that size	N/A	N/A
Quantity total	integer	The quantity being added to inventory(sum of size quantities)	Number	N/A
Sport Name	string	The name of the sport associated with the equipment	N/A	N/A
Apparel?	boolean	Whether the equipment is apparel. User will also enter size information if equipment is marked as apparel.	N/A	N/A
Today's Date	date	The date of the entry	N/A	mm/dd/yyyy

Figure 4.1 - Input of new equipment item

Screen Name	**Data Type**	**Description**	**Limits**	**Format**
Quantity Small	integer	The number of small items	N/A	N/A
Quantity Medium	integer	The number of medium items	N/A	N/A
Quantity Large	integer	The number of large items	N/A	N/A
Quantity XL	integer	The number of XL items	N/A	N/A
Quantity 2X	integer	The number of 2X items	N/A	N/A

Figure 4.2 - Input of size information

Screen Name	**Data Type**	**Description**	**Limits**	**Format**
Team Name	string	The team associated with the player.	N/A	N/A
Age	integer	The players age.	Number	XX
Name	string	The players name.	Characters	N/A
Class	string	The players class in school.	Characters	N/A
Shirt Size	string	The players recommended shirt size.	Characters	N/A
Coach	boolean	Is this person a coach?	N/A	N/A

Figure 4.3 - Input of new player

Screen Name	**Data Type**	**Description**	**Limits**	**Format**

Sport Name	string	The sport associated with the team	N/A	N/A
Team Name	string	The name of the team.	Characters	N/A
Team Description	string	A description of the team.	Characters	N/A
Gender	char	Team gender.	Characters	M/F
Year	integer	The year of the current season.	Number	XXXX

Figure 4.4 - Input of new team

Screen Name	**Data Type**	**Description**	**Limits**	**Format**
Sport Name	string	The name of the sport	N/A	N/A

Figure 4.5 - Input of new sport

5. Report Formats

This section displays rudimentary implementations of how our report system is going to look like. The information and data will be very different for each item, as some items are in much larger quantities or see much more “in and out” usage.

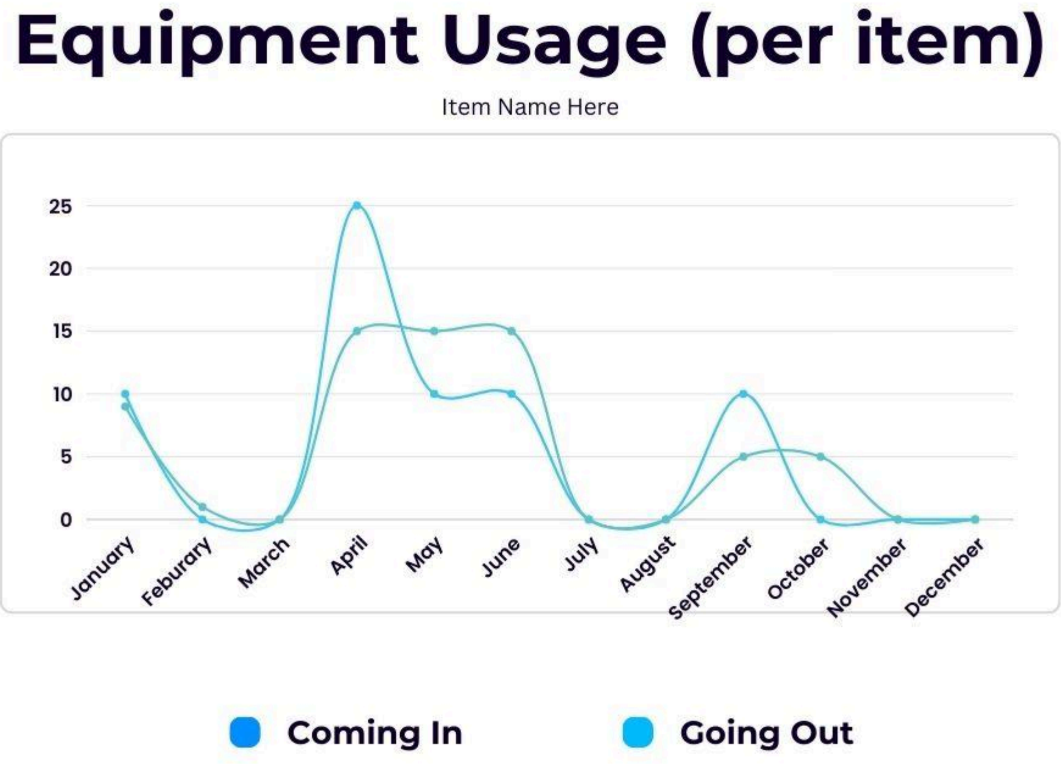


Figure 5.1

In Figure 5.1, this is a general representation of how data will be displayed on a per-item basis. Users can request that data, from any time frame, the example here is monthly for a year, and the program will then generate an “in and out” line chart showing the amount of the item is coming in and going out for use.

UIDD - Appendix A – Agreement Between Customer and Contractor

Agreement Between Customer and Contractor

1. Parties: This agreement made on “11/22/2023” is by and between

Client: University of Maine Athletic Department

AND

Contractor: Inventory Management Software Group

2. Term: The terms of this agreement shall commence on November 5th, 2023, and conclude on May, 2024.

3. Services: The Contractor agrees to provide the following services for the betterment of the Customer: The Contractor will create an inventory management system that allows the Customer to visualize and manage their inventory through an online webpage. Further details are presented in the System Design Document.

4. Expenses: There are no initial expenses. When expenses are incurred, the Contractor will communicate and get approval from the Customer to use funds allotted to them if available.

5. Agreement:
By signing this document, all parties agree to the requirements presented in this document. All parties also agree that the deadlines presented are tentative, and are subject to change as the program is developed.

Customer Signature:		
	Date:	Printed:
X: *Kevin RITZ*	12/4/2023	Kevin Ritz
Contractor Signature:	Date:	Printed:
X: *Collin Rodrigue*	11/22/2023	Collin Rodrigue
X: *Gabriel A. Poulin*	11/22/2023	Gabriel A. Poulin
X: *Brennan Poitras*	11/22/2023	Brennan Poitras
X: *Sean Radel*	11/22/2023	Sean Radel
X:*Graham Bridges*	11/22/2023	Graham Bridges

UIDD - Appendix B – Team Review Sign-off

Team Agreement Sign Off

Team IMSG has thoroughly reviewed the UIDD for the Athletic Inventory System and has agreed that the following information is accurate and achievable. Collectively we have no major contentions in the information stated in the document. By signing this agreement, one acknowledges all the terms and conditions outlined in the document and understands the importance of effective team collaboration, communication, and shared accountability when achieving the goals of the project. By signing below, we pledge our dedication to the success of the team and the project we plan to undertake. We agree to work collaboratively, and support each other to uphold the guidelines and expectations set forth in the agreement.

Signature:	Date:	Printed:
X: *Collin Rodrigue*	11/22/2023	Collin Rodrigue
X: *Gabriel A. Poulin*	11/22/2023	Gabriel A. Poulin
X: *Brennan Poitras*	11/22/2023	Brennan Poitras
X: *Sean Radel*	11/22/2023	Sean Radel
X:*Graham Bridges*	11/22/2023	Graham Bridges

UIDD - Appendix C – Document Contributions

Name	**Date**	**Contribution**	**Version**
Sean Radel	11/23/23	Formatting, introduction, delegation	0.1
Brennan Poitras	11/25/23	Data validation tables	0.1
Graham Bridges	11/26/23	Design Standards and User Interface Walkthrough	0.1
Collin Rodrigue	11/27/23	Data validation tables	0.1
Sean Radel	11/27/23	Accessibility and Inclusion	0.1
Gabriel Poulin	11/27/23	Report Example	0.1
Sean Radel	12/4/23	Peer review Revisions	0.2
Gabriel Poulin	12/14/23	Misc changes	0.3