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

Table of Contents

1. Introduction

This is a capstone project for the University of Maine Athletics Department, in partial fulfillment of the Computer Science BS degree for the University of Maine. The goal of this project is to build a solution for managing and tracking sports equipment and apparel. 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.

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. Athletic Department managed Google Drive containing all inventory tracking spreadsheets:https://drive.google.com/drive/folders/1dclEKCke2CdXU3GNA5ee5VpROfOWX25w
2. MITRE ATT&CK Matrix: https://attack.mitre.org/

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.

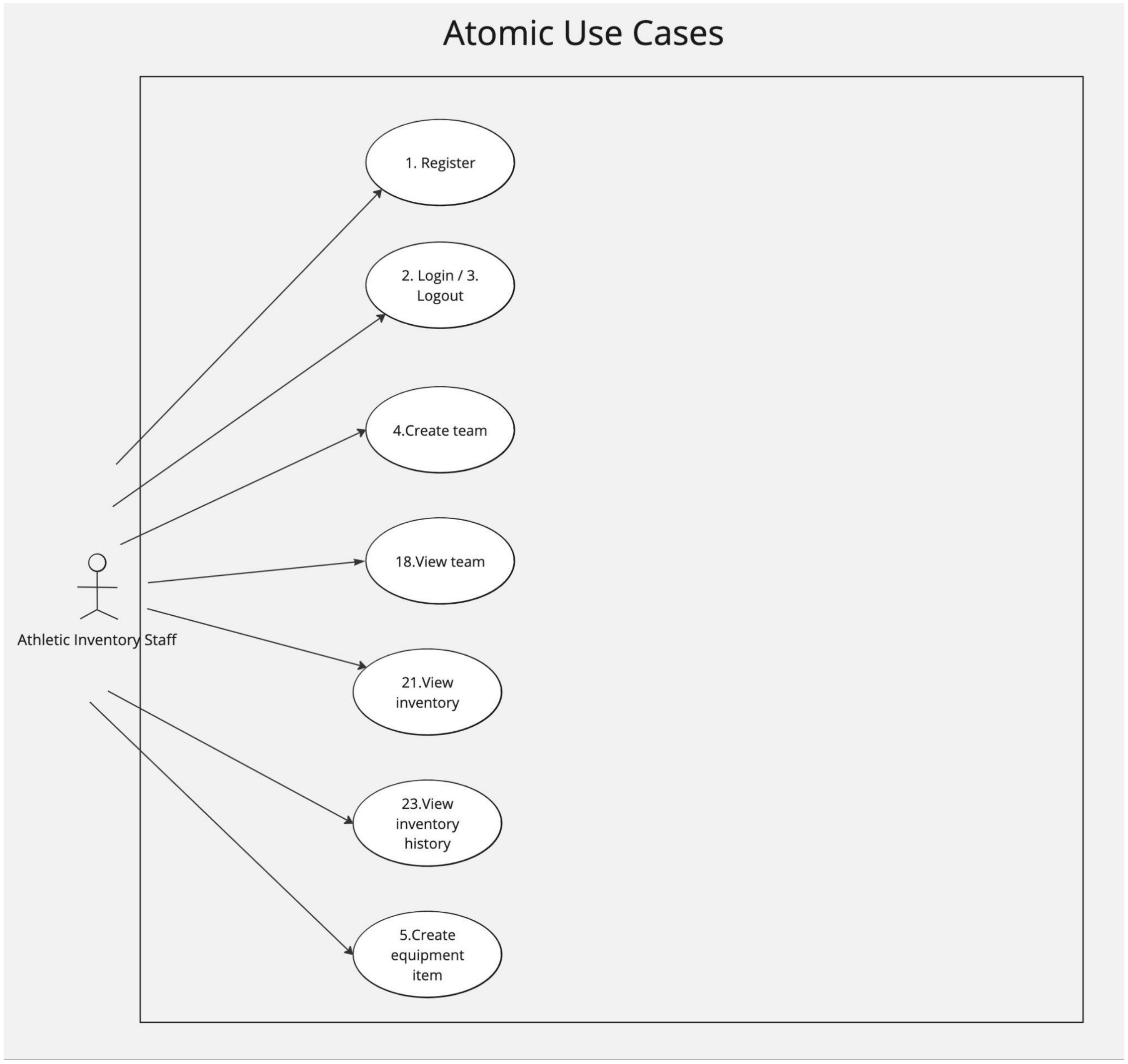

Figure 1. Atomic Use Case Diagram

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.

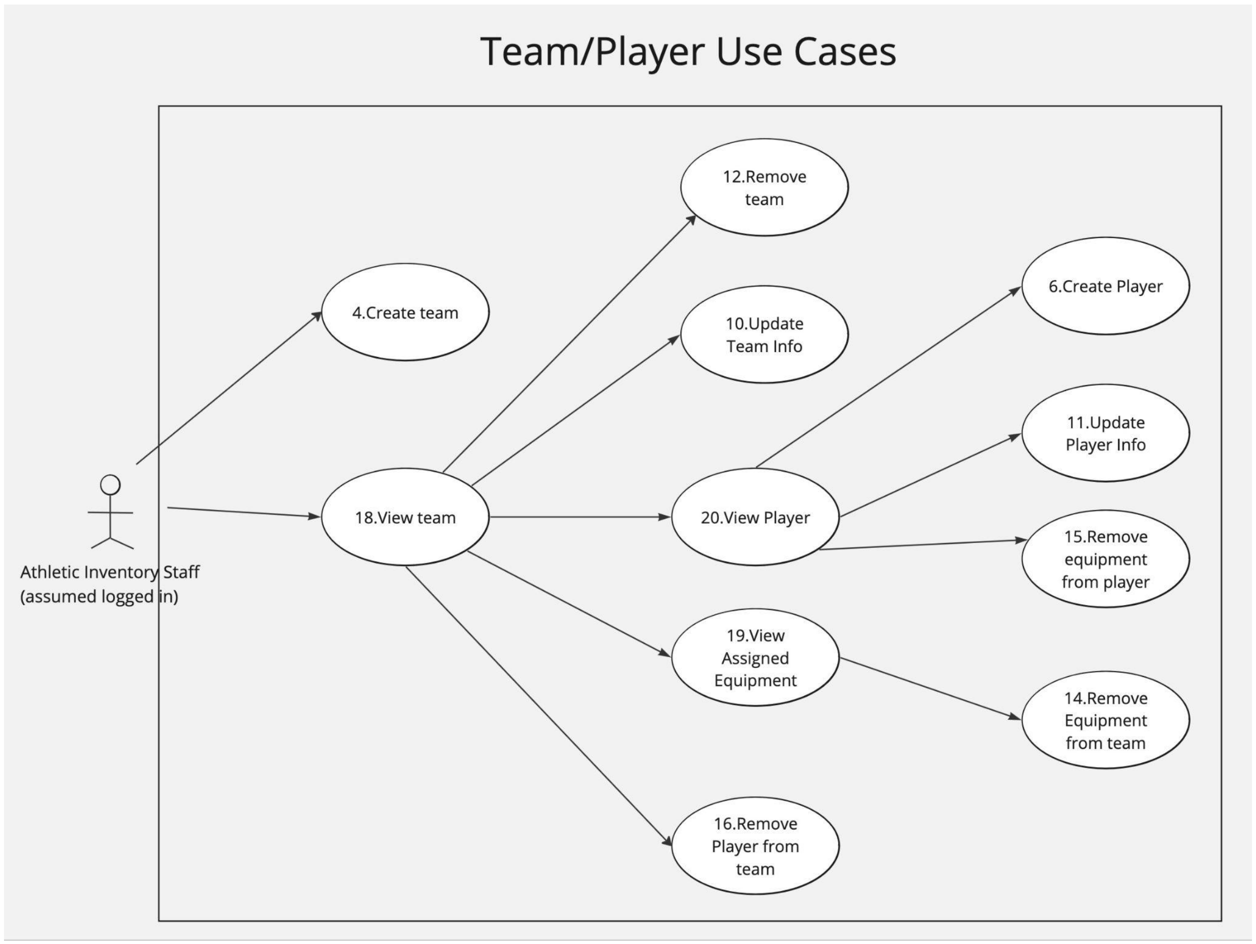
Team/Player Use Cases
12.Remove team
4.Create team
10.Update Team Info
6.Create Player
11.Update Player Info
Athletic Inventory Staff (assumed logged in)
18.View team
20.View Player
15.Remove equipment from player
19.View Assigned Equipment
14.Remove Equipment from team
16.Remove Player from team

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.

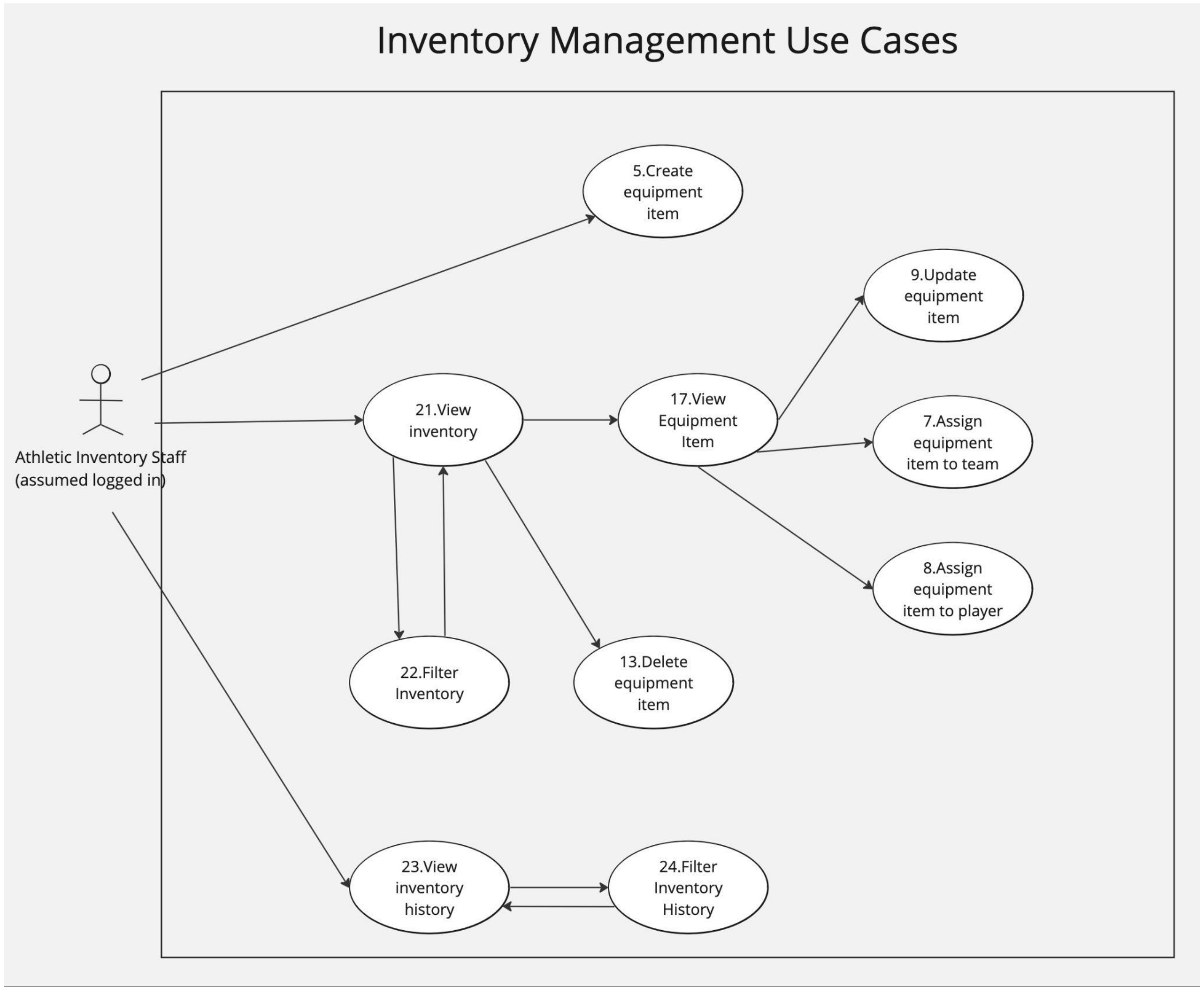
Inventory Management Use Cases
5.Create equipment item
9.Update equipment item
21.View inventory
17.View Equipment Item
7.Assign equipment item to team
8.Assign equipment item to player
Athletic Inventory Staff (assumed logged in)
22.Filter Inventory
13.Delete equipment item
23.View inventory history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

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**
System Requirement Specification	1 November 2023

System Design Document	15 November 2023
User Interface Design Document	29 November 2023
Critical Design Review Document	15 December 2023
User Manual	Early 2024
Administrator Manual	Early 2024
All source code	March 2024
Complete Program	April 2024

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

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

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