Deliverable 4

Team Name: Mint

Project Name: UJ Sport Athlete Management & Booking System

Team Number: 26

Mentor Name: Mr. Daniel Ogwak

Team Members

220046244 – Pashin Soma

221060349 – Tshepo Mohale

221000696 – Morena Ramaili

221105301 – Paidamoyo Mapfuwa

Contents

Updated Problem Statement, Proposed Solution & Use Cases

Problem Background

John was excited to start his new job as a coach at UJ Sport. He had great plans for improving the team's performance but soon noticed many problems. Important information about athletes, schedules, and performance was scattered in different places, making it hard to find what he needed. The team mostly used WhatsApp and word of mouth to share updates, which caused confusion as messages were often missed or misunderstood. John sometimes didn't know about changes to the schedule, and athletes missed important sessions. Booking gym slots and sports buses was a slow process that required many phone calls, leading to scheduling conflicts and frustration. Manual data entry and different systems meant John often found duplicate or incorrect information, making it hard to track athlete performance accurately. He also couldn't easily get a full picture of each athlete's performance and schedule, making it difficult to plan effective training programs. There was no good system to track athlete injuries, making it hard to record and manage injuries, delaying recovery, and increasing the risk of re-injury. Additionally, John didn't have the right tools to give personalized feedback and training plans to athletes, making it hard to improve their skills and techniques.

Jane, a new athlete at UJ Sport, was excited to join the team but soon faced similar problems. She found it hard to keep track of her training schedule and performance because information was spread out in different places. Communication about training schedules and updates was inconsistent, causing her to miss important sessions. Manual data entry led to duplicate or outdated information, making it difficult for Jane to track her progress accurately. She also struggled to access comprehensive information about her performance trends and training schedules, which hindered her ability to make informed decisions and optimize her training efforts. Without interactive coaching tools, Jane felt that she wasn't receiving the personalized feedback and training plans needed to enhance her skills and techniques.

Problem Statement

The athlete management system at UJ Sport is plagued by inefficiencies, including fragmented systems, poor communication channels, manual booking processes, data inaccuracies, limited access to information, inefficient injury management, and lack of coaching support. These challenges hinder decision-making, disrupt workflows, and lead to dissatisfaction among athletes, coaches, and managers. Urgent intervention is needed to integrate systems, improve communication, and streamline processes for a better athlete experience.

Proposed Solution:

To address these challenges, we propose the implementation of the UJ Sport Athlete and Booking Management System. This system will provide a centralized platform for managing athlete-related activities and operational logistics, offering the following key features:

Comprehensive Athlete Profiles: Detailed profiles containing essential athlete information, demographics, medical history, and performance metrics.

Dynamic Scheduling and Calendar: Streamlined scheduling capabilities for managing training sessions, competitions, and personal appointments, integrated with external calendars and notification systems.

Performance metrics: Dedicated athlete performance tracking enabling monitoring of training progress and performance metrics.

Virtual Injury Management: Seamless communication between athletes, coaches, and medical professionals for tracking injuries.

Interactive Coaching and Feedback: Tools for personalized feedback, training plans, and performance evaluations.

Enhanced Team Management: Features for roster organization, communication, and collaboration, including group training schedules and coordination tools.

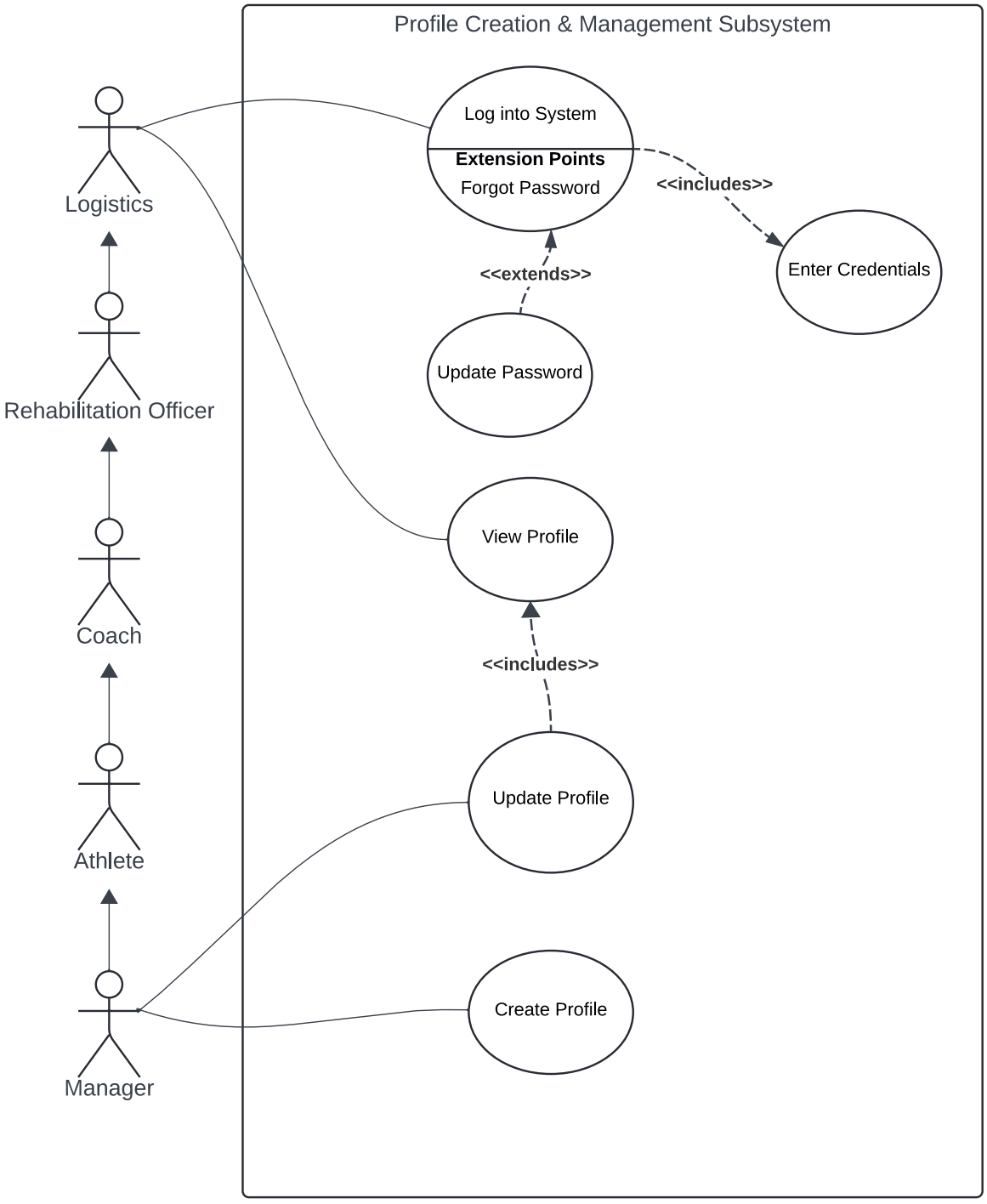
Robust Reporting and Analytics: Comprehensive reporting and analytics capabilities providing valuable insights into athlete performance, team dynamics, and strategic planning.

In addition, the system will also include efficient booking systems for gym slots and transportation, offering real-time availability updates and capacity management to optimize facility usage and streamline transportation logistics.

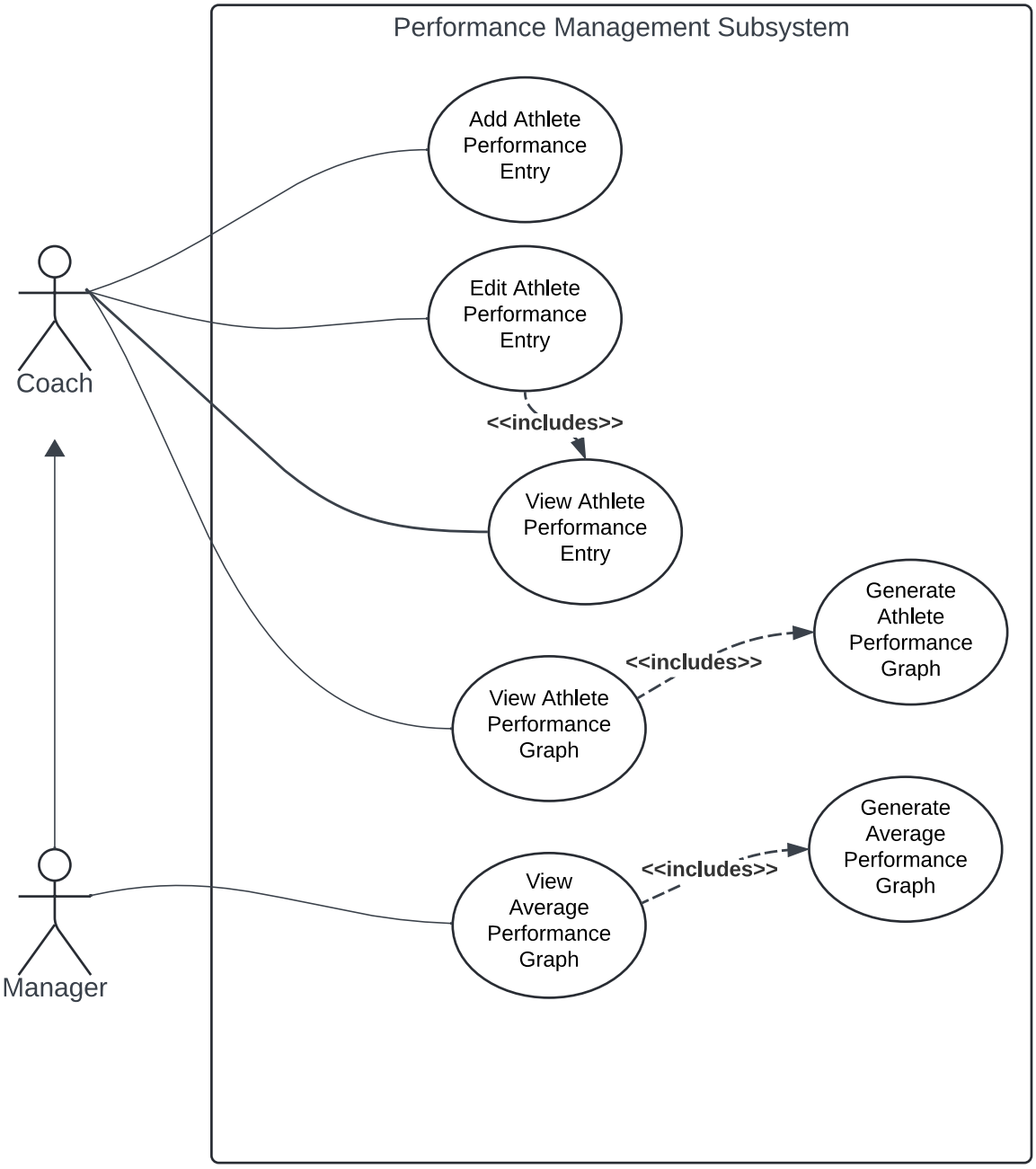
Our solution is going to significantly enhance UJ Sport’s operations, athlete performance, and overall efficiency, empowering athletes and coaches to achieve new levels of success.

Use Cases

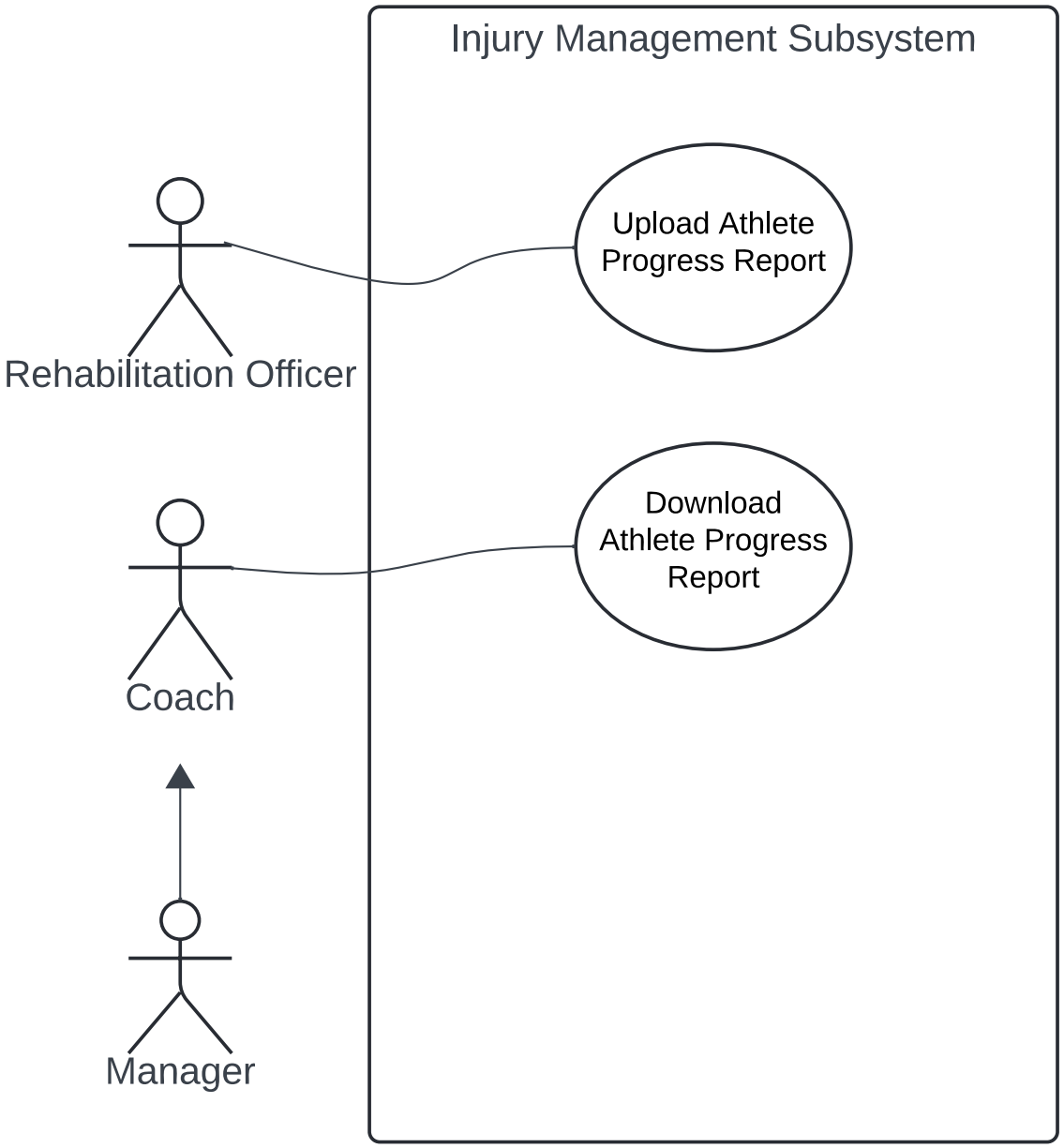
Profile Creation & Management Subsystem



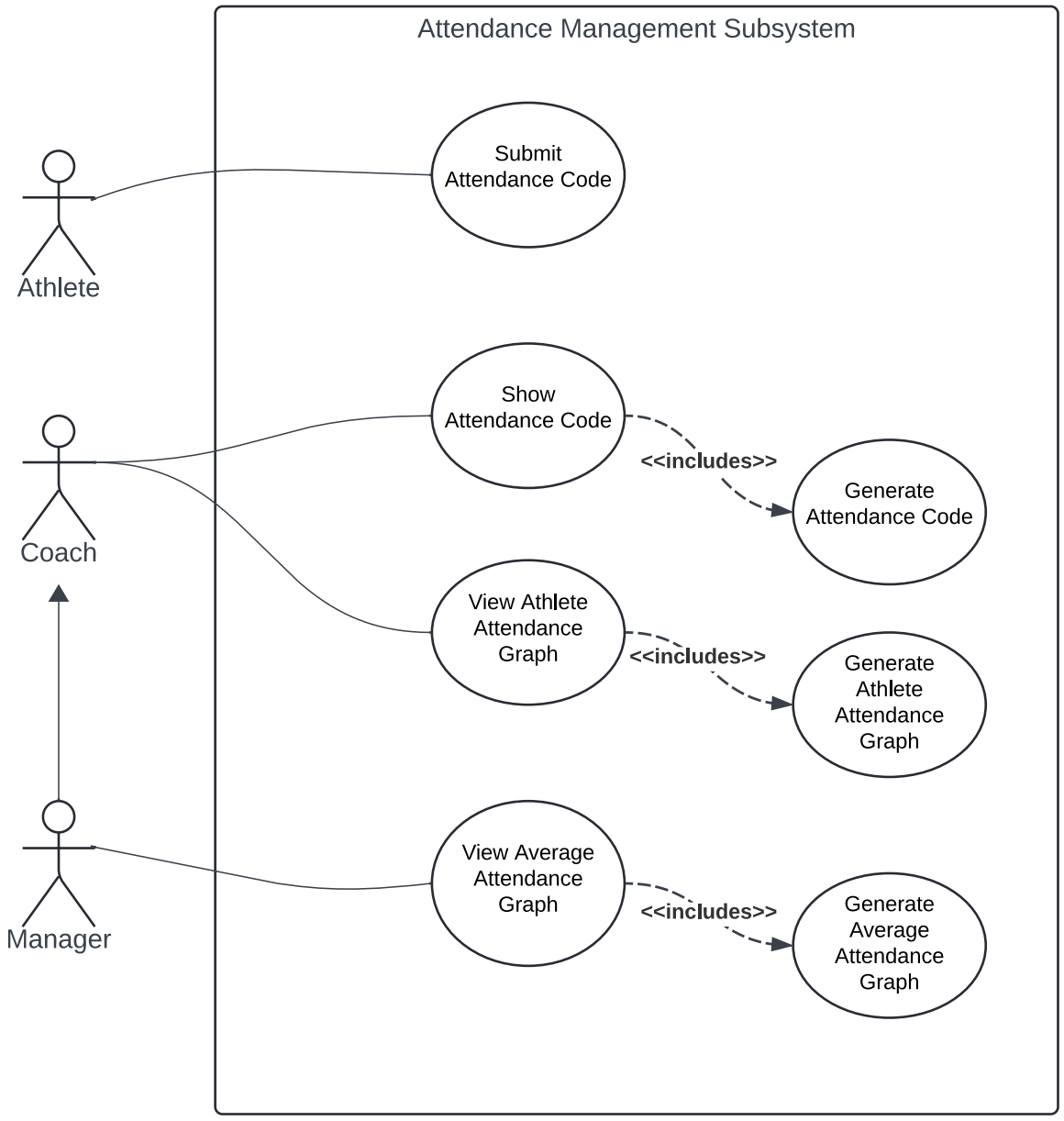
Performance Management, Reporting, and Analytics Subsystem



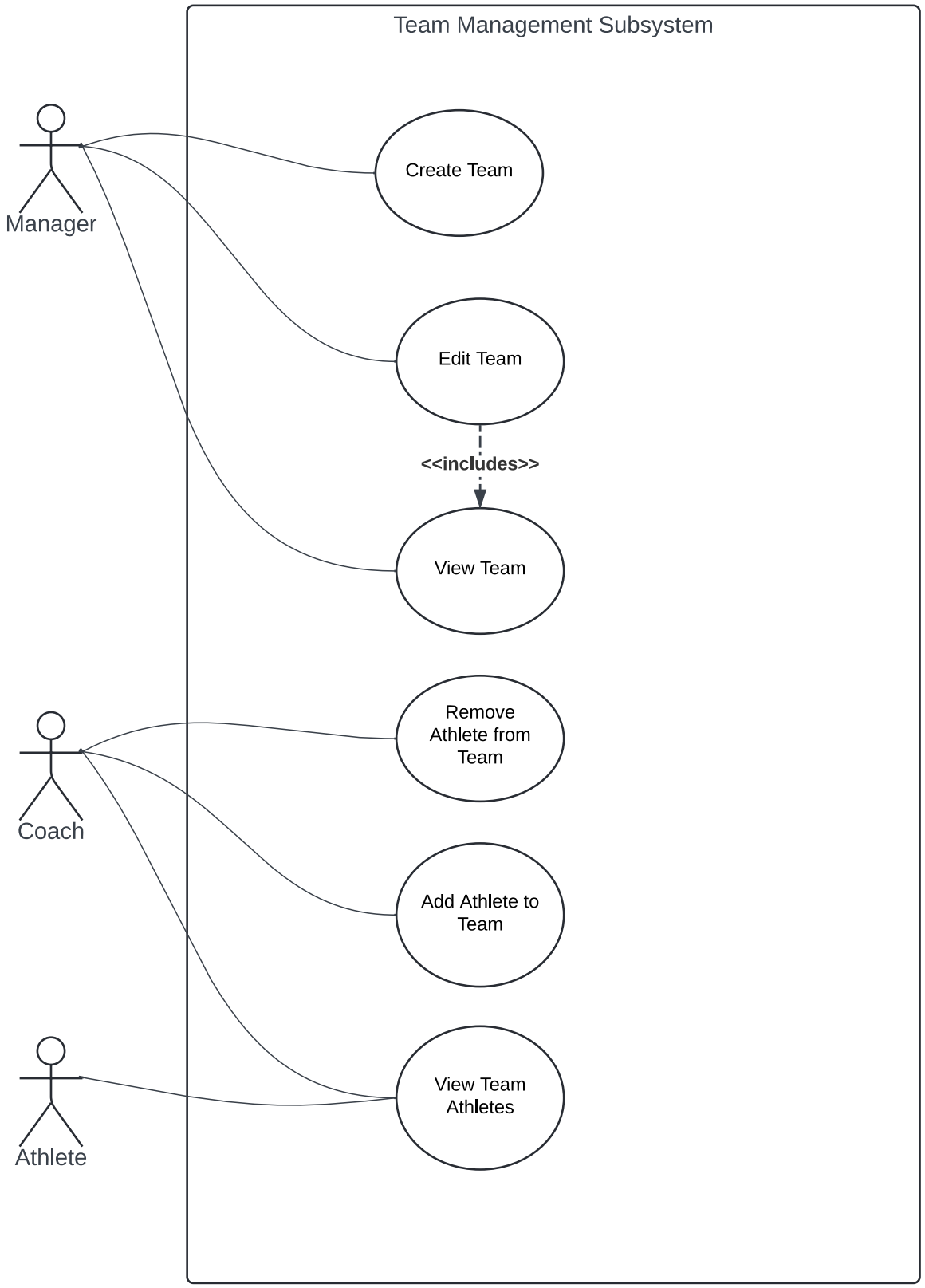
Injury Management Subsystem



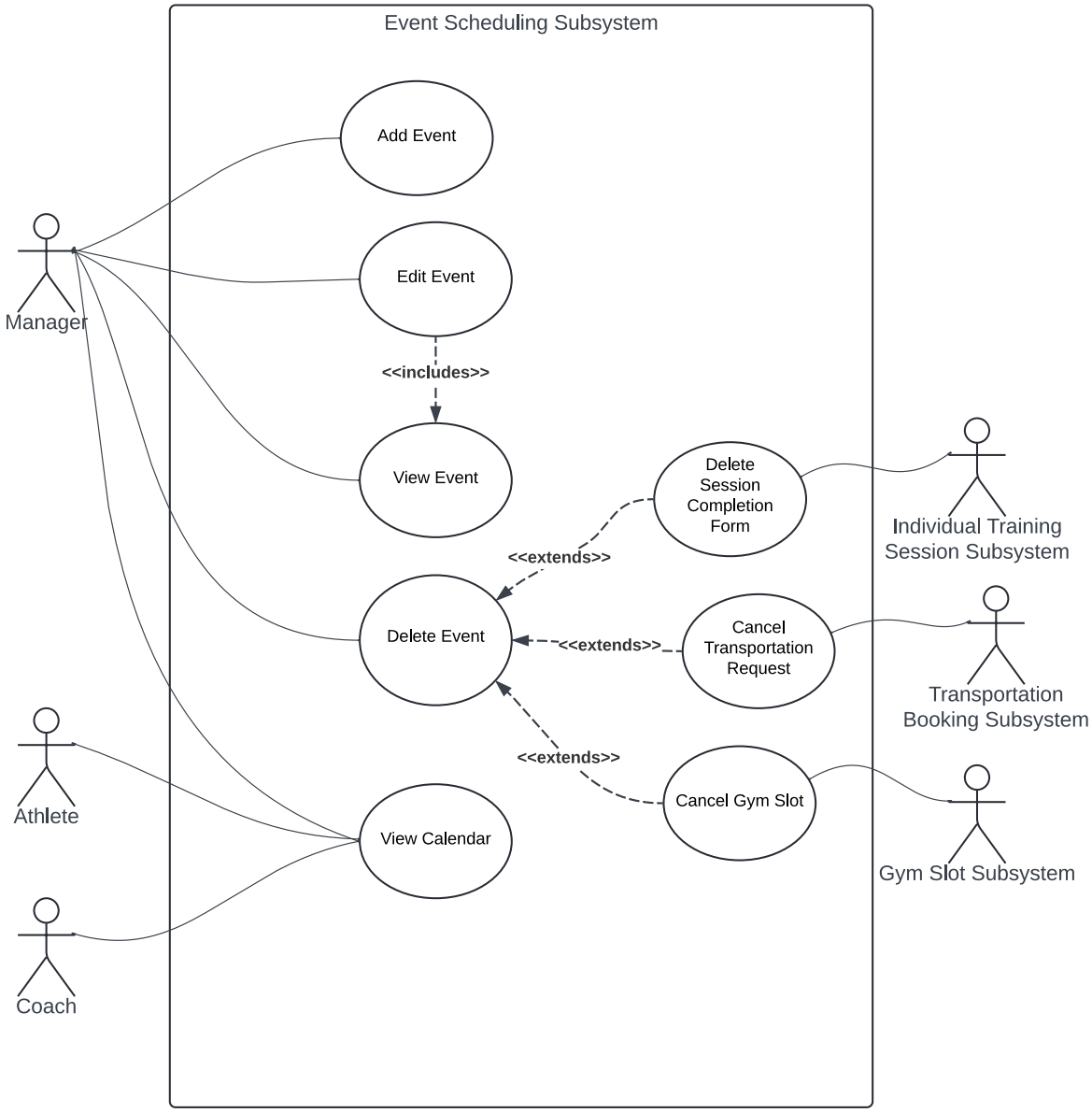
Attendance Management Subsystem



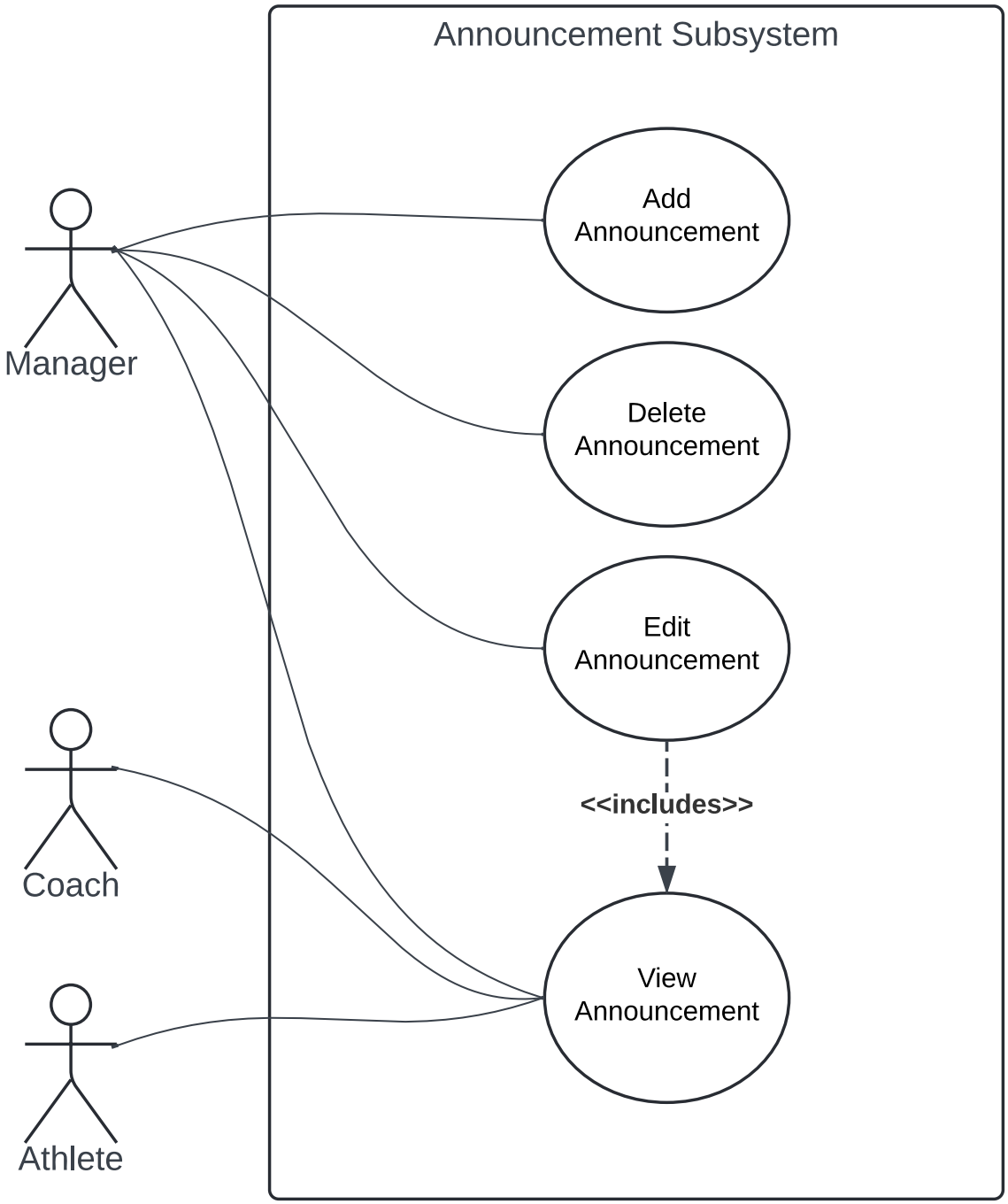
Team Management Subsystem



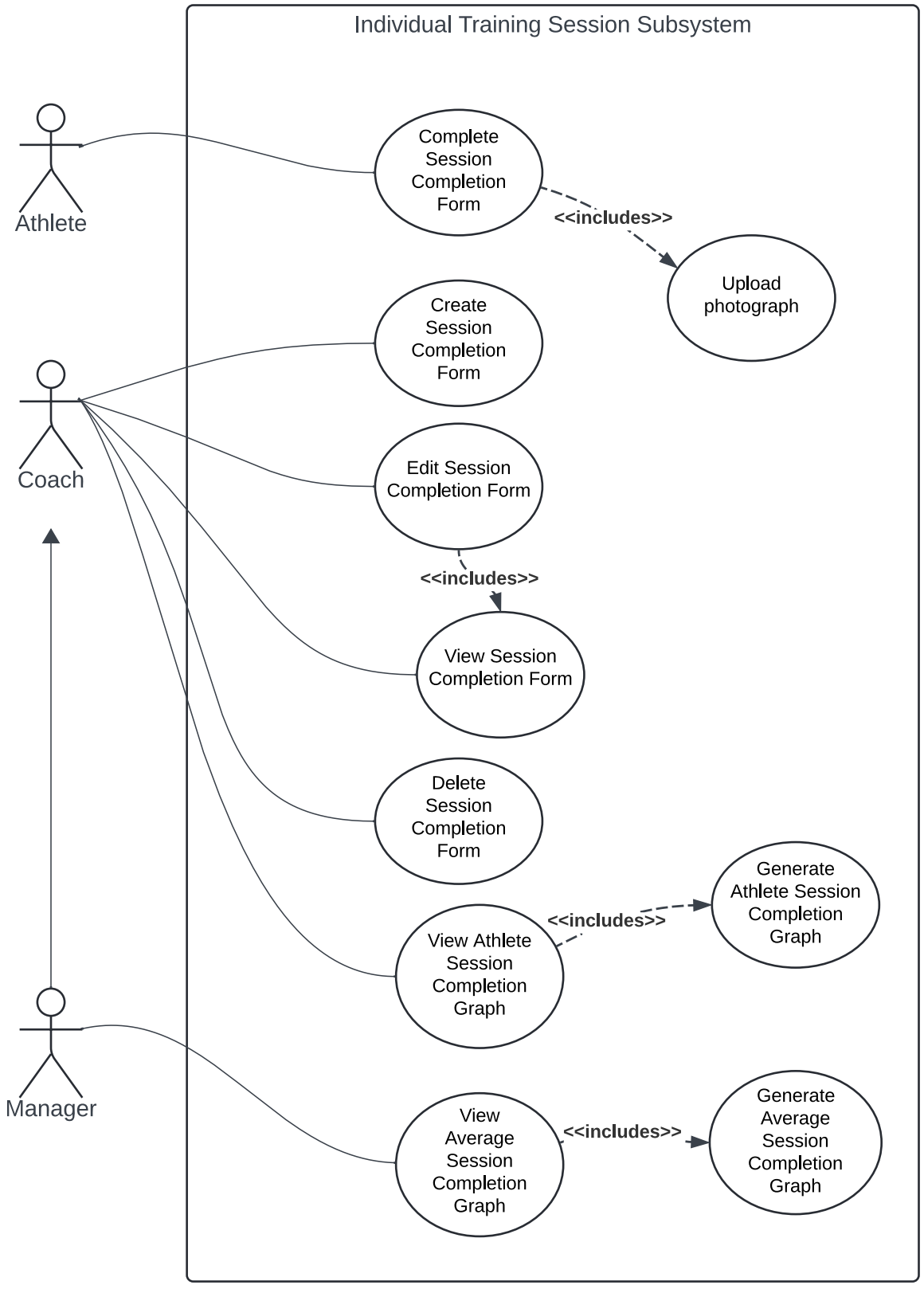
Event/Calendar Scheduling Subsystem



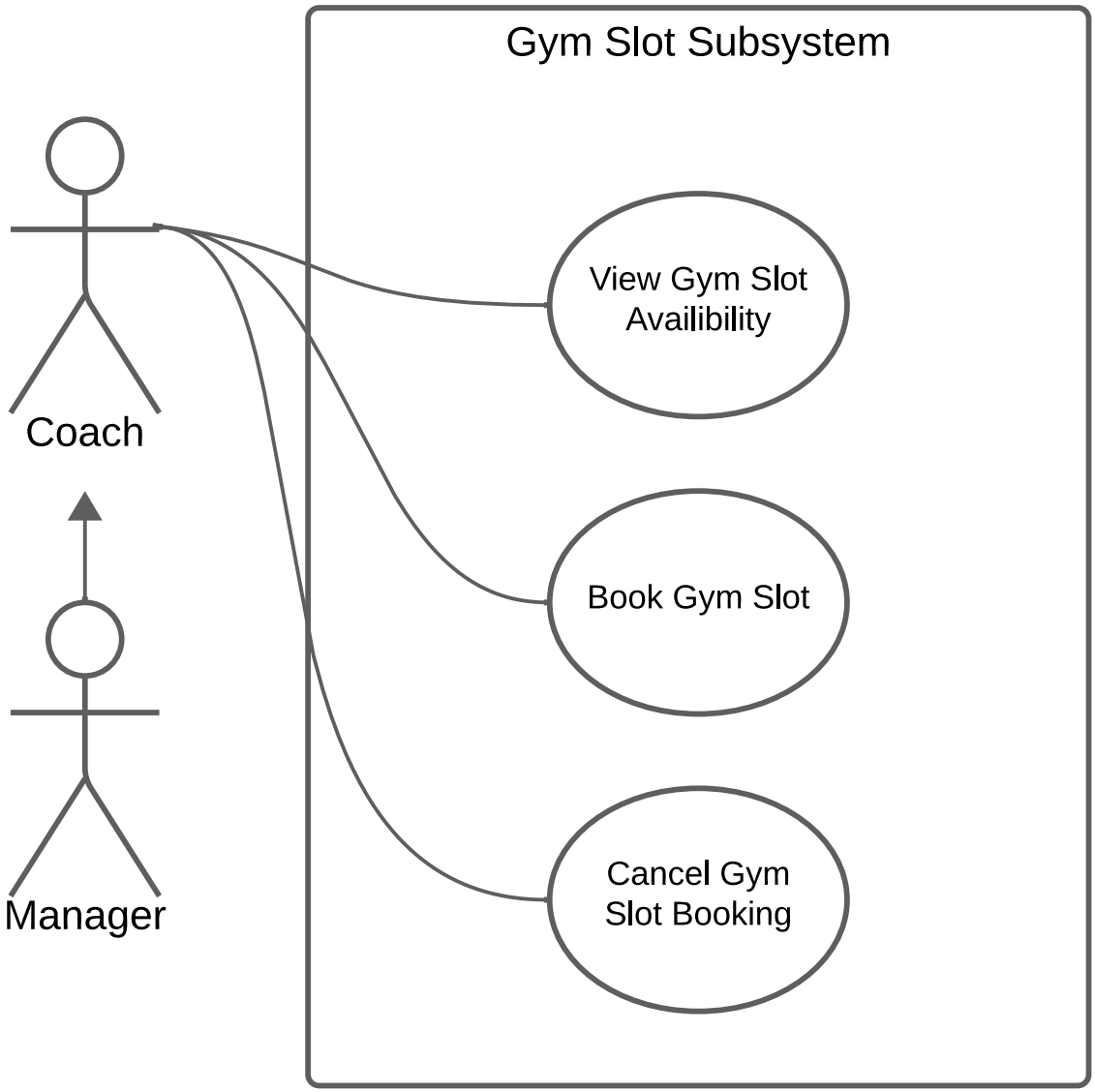
Announcements Subsystem



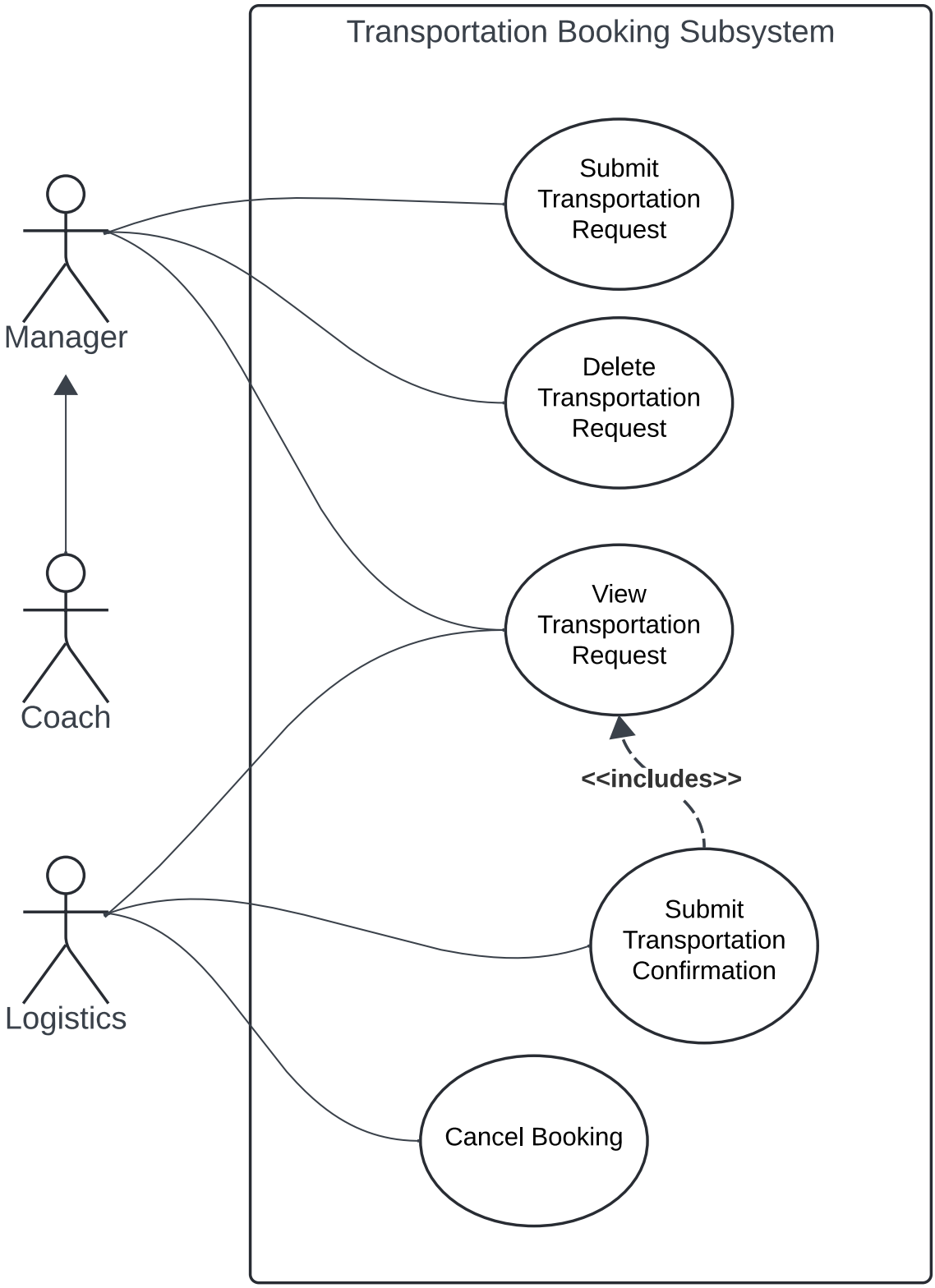
Individual Training Session Subsystem



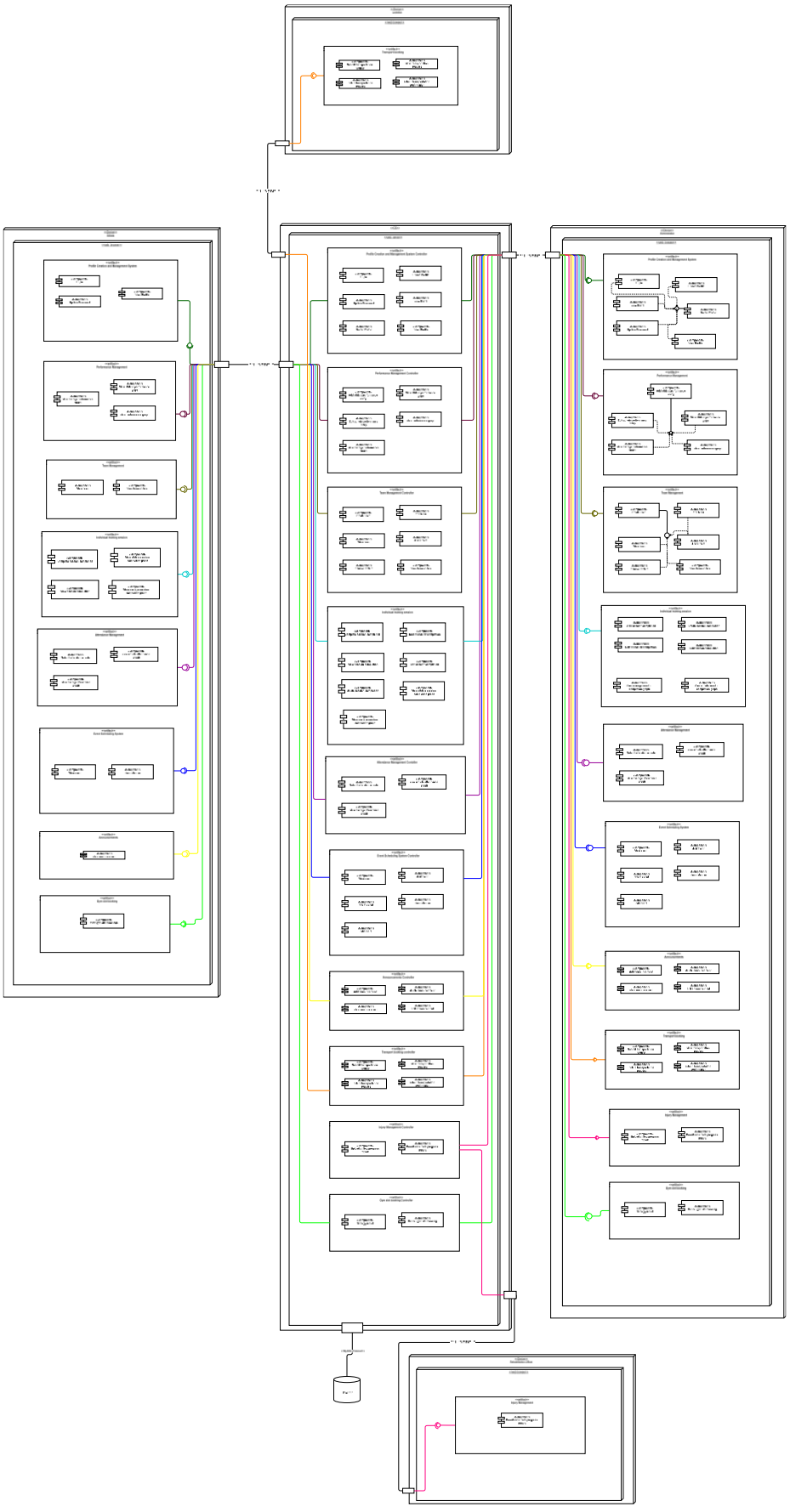
Gym Slot Booking Subsystem



Transportation Booking Subsystem



Component & Deployment Diagram



Comprehensive Verification and Validation (V&V) plan

Validation

Requirements checks:

1. **Profile Creation and Management**

   Completeness: Ensure that all necessary profile fields are included and that these fields are different for different users. Ensure that an admin (Manager) can update profile information.

   Correctness: Ensure that the subsystem follows the flow of steps of inputting user information, verifying the information, and storing it. Verification of the information will be handled by a manager. Also, ensure that the subsystem's functionality reflects real-world interactions that a user will have with their profile.

   Consistency: Ensure that only a manager user type can update and create a profile. Ensure that the profile creation and update process is consistent for all Manager user types. The subsystem should also have a consistent use of terminology.

   Accuracy: Ensure that profile creation and update fields are defined and that steps for profile creation and update are specified. Describe the expected system behaviours after a profile creation or update procedure.

2. **Performance Management**

   Completeness: Ensure that the performance metrics for a sport include all fields related to the performance of the athletes in that sport. Ensure that performance graphs can be generated for different groups (Athletes and Teams). Users can also generate graphs for different groups.

   Correctness: Ensure that a user should first be allowed to input performance data before generating a graph report output.

   Consistency: Ensure that graph generation formats are consistent for different period specifications. Ensure there are consistent uses of terminology.

   Accuracy: Clearly define each performance metric and its measurement unit. Describe the expected outcomes and visuals for a performance report.

3. **Injury Management**

Completeness: Ensure that all fields for an athlete’s recovery are included in an athlete's progress report. Ensure that fields for tracking an athlete's progress are included in the progress report, for a coach or manager. There should be an ability to view historical reports.

Correctness: Ensure that the process of injury tracking has a logical flow from incident to recovery tracking.

Consistency: Ensure that there are consistent procedures for reporting an injury and that only a coach can report an injury, and a Rehabilitation Officer can upload a progress report.

Accuracy: Clearly define each injury reporting field and its format. Specify steps for updating injury status.

4. Attendance Management

Completeness: Include the different attendance types. Ensure that attendance records are kept and that fields to be used for generating the attendance graph, are included.

Correctness: Ensure that a Coach is first allowed to mark attendance before tracking attendance, thus allowing for a logical flow of processes.

Consistency: Only a Coach or Manager can mark or track attendance.

Accuracy: Ensure that the steps for marking attendance are intuitive and have a logical flow, thus reducing the possibility of user error.

5. Team Management

Completeness: Ensure that all team details are included in team creation fields, that would be used for team management. Ensure that there is a Manager and Coach assigned to a team.

Correctness: Ensure that a Coach is first given the option to view and select a team, before adding a user to the team.

Consistency: Only a manager can create a Team, and only a Coach can add and remove athletes from a Team. Ensure that the procedures for creating a team are consistent.

Accuracy: Define the steps for creating a team and adding or removing athletes from a team.

6. Event Scheduling

Completeness: Include all event types. Ensure that requirements for checking and resolving schedule conflicts are included in the fields for event creation, as well as the requirements for calendar display.

Correctness: Allow a manager to view the calendar before creating an event. This would enable the Manager to view available slots before creating an event.

Consistency: Only a manager can create and manage events. Ensure that the procedures for event creation and management are consistent.

Accuracy: Clearly define each event field, as well as the steps for creating an event.

7. Announcements

Completeness: Include all necessary types of announcements. Ensure that the target group for an announcement is included in the announcement creation fields.

Correctness: Allow a manager the option to first create an announcement before viewing all announcements. This way they can view the new announcement as well.

Consistency: Only a manager can create and send out announcements. Ensure that the procedure for creating an announcement is consistent for all Managers.

Accuracy: Ensure that each announcement field is clearly defined and that the steps for creating an announcement are specified.

8. Individual Training Session Management

Completeness: Ensure requirements for generating a session completion graph are included as fields for an Individual Training Session.

Correctness: Allow a Coach or Manager the option to create a new session completion form before viewing all session completion forms.

Consistency: Only a Manager or Coach can create a session completion form. Ensure that the procedure for creating and submitting a completion form is consistent.

Accuracy: Ensure that each Session completion form field is clearly defined and that steps for creating a session completion form are specified.

9. Booking Gym Slots & Transportation

Completeness: Ensure that fields for identifying slot clashes are included in the creation of a gym slot. Ensure that the date and time of a slot are included in a Transportation request and Gym slot booking.

Correctness: Allow a Coach or Manager to view Gym slot bookings before proceeding to book gym slots. Allow Logistics to view all transportation bookings before proceeding to view and confirm booking requests. This would make the users more knowledgeable of the available gym and transportation slots.

Consistency: Managers and Coaches can book Gym slots and submit Transportation Requests. Only Logistics can view and confirm Transportation Requests. Ensure that the procedure for submitting Transportation Requests, Booking Gym Slots, and Confirming Transportation Requests is consistent.

Accuracy: Ensure that Gym slot fields and Transportation request fields are well defined and that the steps for Booking Gym slots, Requesting Transportation, and confirming Transportation are specified.

Verification

Testing Plan

Introduction

Purpose

To develop a possible testing strategy for the system and to ensure that the system's functionality meets all the specified requirements. The testing plan will also assist in identifying all the defects the system may have.

Scope

The system's modules include Profile Creation & Management, Performance Management Reporting and Analytics, Injury Management, Attendance Management, Team Management, Event/Calendar Scheduling, Announcements, Individual Training Session Booking, Gym Slot Booking, and Transportation Booking.

Objectives

- To ensure that all system's functionalities work as intended.
- To identify and fix the defects in the system.

- To verify that the system is reliable, secure, and works well under expected conditions.

Test Items

Profile Creation & Management Subsystem, Performance Management Reporting and Analytics Subsystem, Injury Management Subsystem, Attendance Management Subsystem, Team Management Subsystem, Event Scheduling Subsystem, Announcements Subsystem, Individual Training Session Booking Subsystem, Gym Slot Booking Subsystem, Transportation Booking Subsystem

Features to be Tested.

Profile Creation & Management System

- User Authentication.
- Profile creation and update.
- Profile role allocation.

Performance Management Reporting and Analytics Subsystem

- Add and edit performance entries.
- View historical performance data.
- Generate graphs from historical performance data.

Injury Management Subsystem

- Recording injuries.
- Tracking recovery.
- Viewing historical injury data.

Attendance Management Subsystem

- Capturing an Athlete’s attendance.
- Viewing attendance records.
- Generating graphs from attendance records.

Team Management Subsystem

- Creating and editing a team.
- Adding and removing athletes from a team.

Event Scheduling Subsystem

- Scheduling and editing an event.

- Checking for event date conflicts.
- Removing an event.
- Viewing all scheduled events on Calendar interface.

Individual Training Session Subsystem

- Submit Session Completion form.
- Create, Edit, and Delete Session Completion form.
- View historical data on form completions.
- Generate graphs from historical data.

Announcement Subsystem

- Add, Edit, and Delete Announcements.
- Set Announcements to target specific groups.

Gym Slot Booking Subsystem

- View available slots.
- Book and Cancel gym slots.
- Check for clashes in gym slot bookings.

Transportation Booking Subsystem

- Submitting transportation requests.
- Confirming and Cancelling transportation bookings.

Features not to be Tested.

- Non-functional requirements: Performance, Scalability, Accessibility, Security, Auditability, Portability.

Approach

Testing Type

- Unit testing: Testing each of the functions in the system's code.
- Integration testing: Testing that different components in the system are working together the way they are supposed to.
- Acceptance testing: Verifying that the system meets user requirements by allowing the user to use the system and receiving feedback.
- Regression testing: Retesting the system after changes are made to the code, to ensure that existing functionality is not affected by the changes.

- Functional testing: Verifying that the system functions as per the specified requirements that were documented.

Test Design Techniques

- Use case testing.

Pass/fail criteria.

- Testing will be said to be passed if tests are completed without errors, with the expected outputs being produced.
- Inversely testing will be said to have failed if not all the test cases can be completed, or the test cases produce unexpected outputs.

Suspension criteria and Resumption requirements

- Testing will be suspended if a critical error occurs, not allowing for further testing.
- Testing will resume once the error has been resolved.

Test Deliverables

Test plan, Test cases, Test scripts, Test data, Test results, and Defect reports.

Testing Tasks

- Create Test Cases for each subsystem.
- Prepare Test Data, for Test Case execution.
- Execute Test Cases.
- Log Defects.
- Create a Test Report.
- Share the Report with the project sponsor.

Environmental Needs

Hardware Requirements

Server:

- Database, web application, and application logic will be housed and hosted on our local machines.

Client Machines:

- Desktops/Laptops: For testers to test the web components of the test cases.
- Mobile: For testing mobile accessibility and functionalities.

Software Requirements

Operating Systems:

- Windows, Android.

Browsers:

- Any browser may be used as Mobile and Web browsers.

Database

- Microsoft SQL Server

Development Environment

- IDE: Visual Studios.
- Version Control: GitHub.

Network Requirements

A stable connection to the internet.

Responsibilities

Tshepo Mohale:

- Planning: Develop the test strategy and plan. Ensure that project goals and timelines align with the testing timeline and goals.
- Coordination: Coordinate testing activities among team members.
- Monitoring and Control: Track the testing activities, and handle issues that may arise.

Pashin Soma:

- Requirements Analysis: Review and analyze system requirements and specifications.
- Test Execution: Execute the test cases as per the test plan.
- Defect Logging: Identify, log, and track defects.

Morena Ramaili:

- Test Case Design: Create test cases covering all functional and non-functional requirements of the system.
- Test Data Preparation: Identify and prepare test data for executing test cases.
- Test Documentation: Document test cases and any other additional test artifacts.

Paida Mapfuwa:

- Reporting: Prepare and present test progress and defects reports to the project sponsor.
- Peer review: Review test cases prepared by other team members to ensure completeness and correctness.

Schedule

- Test Planning: 10 May 2024 – 12 May 2024
- Test Case Development: 13 May 2024 – 17 May 2024
- Test Execution: 16 August 2024 – 19 August 2024
- Defect Fixing and Verification: 2 September 2024 – 8 September 2024
- Final Testing and Reporting: 20 September 2024 – 29 September 2024

Risk and Mitigation

i. **Incomplete Requirements:** Requirements are not fully documented or understood, leading to gaps in test coverage.

Mitigation:

- Conduct thorough requirement analysis.
- Have regular meetings with sponsors to confirm requirements.

ii. **Limited Resources:** Insufficient testing resources.

Mitigation:

- Prioritize test cases based on risk and impact.

iii. **Test Data Issues:** Insufficient or irrelevant test data, leading to inaccurate test results.

Mitigation:

- Create test data that covers all scenarios.
- Regularly update test data to reflect real-world scenarios.

iv. **Human error:** Mistakes that would be made by us, as testers. Such as incorrect test execution or misinterpretation of results.

Mitigation:

- Use peer reviews and pair testing to reduce individual errors.

Test Cases

Profile Creation & Management Subsystem

Use Case: Log into System

#	Test Name	Input	Conditions/Tests	Expected Output
1	User Login	User email Password	User exists and password correct.	User authenticated.

				Continue to user dashboard.
2		User email Password	User does not exist.	Display error message – email is incorrect.
3		User email Password	User exists, password wrong.	Display error message – password is incorrect.
4		User email Password	Details incorrect.	Display error message – details invalid.
		User email Password	Incorrect format.	Display error message – email/password is in incorrect format.

Use Case: View Profile

#	Test Name	Input	Conditions/Tests	Expected Output
1	View User	User ID	User exists.	Continue to view User
2		User ID	User not found.	Display error message – no user found.

Use Case: Update Profile

#	Test Name	Input	Conditions/Tests	Expected Output
1	Update User	User ID	User exists. Details Correct.	Details captured. User updated. Continue to view user.
2		User ID	User not found.	Display error message – user not found.
		User type Sport First name Last name Email Address Province	Details not correct format	Display error message – enter correct details in correct format
		User type Sport First name Last name Email	Details incomplete	Display error message – fill in all fields

		Address Province		

Use Case: Create Profile

#	Test Name	Input	Conditions/Tests	Expected Output
1	Add User	User type Sport First name Last name Email Address Province	Details correct.	Details captured. Add User to database. Continue to view users
2		User type Sport First name Last name Email Address Province	Details not correct format	Display error message – enter correct details in correct format
		User type Sport First name Last name Email Address Province	Details incomplete	Display error message – fill in all fields

Use Case: Update Password

#	Test Name	Input	Conditions/Tests	Expected Output
1	Update Password	Current Password New Password Re-enter new password	Check if current password matches password stored in database. New password has at least one capital letter and numerical values. New password matches re-enter password.	Delete user from database. Continue to view users.
		Current Password	Password entered does not match current password stored in database.	Display error message – password entered is incorrect. Please re-enter password.

2		New Password Re-enter new password	New password does not meet required format.	Display error message – password is not in correct format. Please re-enter password.
3		New Password Re-enter new password	New password does not match re-enter password.	Display error message – password does not match re-enter password. Please re-enter password.

Use Case: View Users

#	Test Name	Input	Conditions/Tests	Expected Output
1	View User	-	User exists.	Get all existing users from database. Continue to view users.
2		-	User not found.	Display error message – No Users in database.

Use Case: Delete User

#	Test Name	Input	Conditions/Tests	Expected Output
1	Delete User	User ID	User exists.	Delete user from database. Continue to view users.
2		User ID	User not found.	Display error message – user does not exist.

Team Management Subsystem

Use Case: Create Team

#	Test Name	Input	Conditions/Tests	Expected Output
1	Create Team	Team Details	Valid Details	Create Team. Display success message.
2		Team Details	Invalid Details	Display error message – invalid field/fields

Use Case: Edit Team

#	Test Name	Input	Conditions/Tests	Expected Output
1	Edit Team	Team Changes	Valid Changes	Submit changes. Display success message.
2		Team Changes	Invalid Changes	Display error message – invalid field/fields

Use Case: View Team

#	Test Name	Input	Conditions/Tests	Expected Output
1	View Team	Team ID	Team exists	Display Team details.
2		Team ID	Team not found	Display error message – team does not exist

Use Case: Add Athlete to Team

#	Test Name	Input	Conditions/Tests	Expected Output
1	Add Athlete to Team	Athlete ID	Athlete exists	Add Athlete to team. Display success message.
2		Athlete ID	Athlete not found	Display error message – athlete does not exist

Use Case: Remove Athlete from Team

#	Test Name	Input	Conditions/Tests	Expected Output
1	Remove Athlete from Team	Athlete ID	Athlete is in team	Remove Athlete from team. Continue to view team athletes.
2		Athlete ID	Athlete not found in team	Display error message – athlete is not in team.

View Team Athletes

#	Test Name	Input	Conditions/Tests	Expected Output
1	View Team Athletes	Team ID	Team exists	Display Team details.
2		Team ID	Team not found	Display error message – team does not exist

Injury Management Subsystem

Use Case: Upload Athlete Progress Report

#	Test Name	Input	Conditions/Tests	Expected Output
1	Upload Athlete Progress Report	Athlete ID File	Athlete exists. All fields complete. Correct file type.	Upload file. Display success message.
2		File	Incorrect format.	Display error message – choose correct file type.
		Athlete ID File	Fields incomplete.	Display error message – complete all fields.

Use Case: Download Athlete Progress Report

#	Test Name	Input	Conditions/Tests	Expected Output
1	Download Athlete Progress Report	Athlete ID	Athlete exists with progress report ID.	Displays all reports for athlete. Downloads reports.
2		Athlete ID	Athlete exists with no progress report ID.	Display error message – athlete does not have any progress reports to download.

Attendance Management Subsystem

Use Case: Submit Attendance Code

#	Test Name	Input	Conditions/Tests	Expected Output
1	Submit Attendance Code	Attendance Code, Athlete ID	Attendance Code exists	Display Success message. Mark Session attendance.
2		Attendance Code, Athlete ID	Attendance Code doesn't exist.	Display error message – code doesn't exist

Use Case: Show Attendance Code

#	Test Name	Input	Conditions/Tests	Expected Output
1	Show Attendance Code	Session ID	Session exists	Create and allow the user to share the attendance code.
2		Session ID	Session does not exist.	Display error message – session does not exists

Use Case: View Athlete Attendance Graph

#	Test Name	Input	Conditions/Tests	Expected Output
1	View Athlete Attendance Graph	Athlete attendance entries	Athlete has attendance entries	Generate and display attendance graph
2		Athlete attendance entries	Athlete has no attendance entries	Display message – no entries

Use Case: View Average Attendance Graph

#	Test Name	Input	Conditions/Tests	Expected Output
1	View Average Attendance Graph	Team attendance entries	Team has attendance entries	Generate and display Team attendance graph
2		Team attendance entries	New Team, Team has no attendance entries	Display message – no entries

Event/Calendar Scheduling Subsystem

Use Case: Add Event

#	Test Name	Input	Conditions/Tests	Expected Output
1	Add Event	Event Details	Valid Details	Add Event. Display success message.
2		Event Details	Invalid Details	Display error message – Invalid field/fields

Use Case: View Event

#	Test Name	Input	Conditions/Tests	Expected Output
1	View Event	Event ID	Event exists	Show Event Details
2		Event ID	Event not found	Display error message – Event does not exist

Use Case: Edit Event

#	Test Name	Input	Conditions/Tests	Expected Output
1	Edit Event	Changed Fields	Valid Changes	Submit Changes. Display success message.
2		Changed Fields	Invalid Changes	Display error message – Invalid value at field

Use Case: Delete Event

#	Test Name	Input	Conditions/Tests	Expected Output
1	Delete Event	Event ID	Event exists	Delete user from database. Continue to view Events
2		Event ID	Event not found	Display error message – Event does not exist

Use Case: View Calendar

#	Test Name	Input	Conditions/Tests	Expected Output
1	View Calendar	Events	There exists some events.	Display Events on Calendar.
2		Events	There are no events.	Display message – no Events to display

Performance Management, Reporting, and Analytics Subsystem

Use Case: Add Athlete Performance Entry

#	Test Name	Input	Conditions/Tests	Expected Output
1	Add Athlete Performance Entry	Athlete ID Performance metrics Notes	All details correct.	Add new performance entry.
2	Invalid Format	Athlete ID Performance metrics Notes	Incorrect format.	Display error message – complete fields in correct format.
3	Incomplete Fields	Athlete ID Performance metrics Notes	Fields incomplete.	Display error message – please complete all fields.

Use Case: View Athlete Performance Entries

#	Test Name	Input	Conditions/Tests	Expected Output
1	View Athlete Performance Entries	Athlete ID	Performance entry exists.	View all performance entries for athlete.
2	Invalid Athlete ID	Athlete ID	No performance entries found.	Display error message – user does not exist.

Use Case: Edit Athlete Performance Entries

#	Test Name	Input	Conditions/Tests	Expected Output
1	Edit Athlete Performance Entry	Performance ID. Updated Performance metrics Notes	Performance entry exists. Fields is in the correct format. All fields complete.	Update performance entry. Display success message.
2	Invalid Format	Performance ID. Updated Performance metrics Notes	Fields is in the incorrect format.	Display error message – complete fields in correct format.
3	Incomplete Fields	Performance ID.	Fields incomplete.	Display error message – please complete all fields.

		Updated Performance metrics Notes		

Use Case: View Performance Graph

#	Test Name	Input	Conditions/Tests	Expected Output
1	View Performance Graph	Athlete ID	User exists and has performance ID.	Delete user from database. Continue to view athletes' performance graph.
2	No User data	Athlete ID	User has no performance ID	Display error message – user does not have any performance entries.

Use Case: View Average Performance Graph

#	Test Name	Input	Conditions/Tests	Expected Output
1	View Average Performance Graph	Performance ID	Performance entry exists.	View average performance entry graph.
2	No User data	Performance ID	No performance entries found.	Display error message – No Performance entries available to generate graph.

Announcements Subsystem

Use Case: Add Announcement

#	Test Name	Input	Conditions/Tests	Expected Output
1	Add Announcement	Announcement information	All the fields are correct	Add new announcement. Notify the user that the announcement was successfully added.
2	Incomplete Fields	Announcement information	Incomplete fields	Display error message and Prompt user to complete the fields.
3	Invalid Format	Announcement information	Incorrect format	Display error message and Prompt user to correct the data.

Use Case: Delete Announcement

#	Test Name	Input	Conditions/Tests	Expected Output
1	Delete Announcement	Announcement ID	Announcement does exist	Delete the announcement from database and notify the user that the announcement was successfully deleted.
2	Non-existent Announcement	Announcement ID	Announcement does not exist	Display error message – user do

Use Case: View Announcement

#	Test Name	Input	Conditions/Tests	Expected Output
1	View Announcement	None	User exists	Display list of all announcements that belong to the logged in user.
2	Unauthorized Access	None	Not logged in or unauthorized role	None

Use Case: Edit Announcement

#	Test Name	Input	Conditions/Tests	Expected Output
1	Edit Announcement	Announcement ID and Updated information	All details correct	Update the announcement and notify the user that

				changes were successfully made.
2	Incomplete Fields	Announcement ID and Updated information	Fields incomplete	Display error message and Prompt user to complete the fields.
3	Invalid Format	Announcement ID and Updated information	Incorrect format	Display error message and prompt the user to correct the data.

Individual Training Session Subsystem

Use Case: Complete Session Completion Form

#	Test Name	Input	Conditions/Tests	Expected Output
1	Complete Session Completion Form	Session details, photograph	All details correct	Add new session completion entry and notify the user of successful submission.
2	Incomplete Fields	Session details	Fields incomplete	Display error message and Prompt user to complete fields.
3	No Photograph	Session details	No Photograph	Display error message and Prompt user to upload a photo.

Use Case: Create Session Completion Form

#	Test Name	Input	Conditions/Tests	Expected Output
1	Create Session Completion Form	Form details	All details correct	Add new session completion form and notify the user of successful form creation.
2	Invalid Data	Form details	Incorrect format	Display error message and Prompt user to correct the data.

Use Case: View Session Completion Form

#	Test Name	Input	Conditions/Tests	Expected Output
1	View Session Completion Form	None	Logged as authorized role	Display list of session completion forms created.
2	Unauthorized Access	None	Not logged in or unauthorized role	None

Use Case: Edit Session Completion Form

#	Test Name	Input	Conditions/Tests	Expected Output
1	Edit Session Completion Form	Form details, updated details	All details correct	Update the session completion form and notify the user of successful changes.
2	Invalid Data	Form details, updated details	Incorrect format	Display error message and Prompt user to correct the data.

Use Case: Delete Session Completion Form

#	Test Name	Input	Conditions/Tests	Expected Output

1	Delete Session Completion Form	Form ID	Form exists and has no dependencies	Delete the session completion form and notify the user of successful deletion.
2	Dependencies Found	Form ID	Form has dependencies	Display error message and Prompt user to resolve dependencies.

Use Case: View Athlete Session Completion Graph

#	Test Name	Input	Conditions/Tests	Expected Output
1	View Athlete Session Completion Graph	Set parameters	Data for Graph generation exists	Generate and display the session completion graph for the athlete.
2	Empty Graph	Set parameters	Data for Graph generation does not exist	Display error message and display graph without values

Use Case: View Average Session Completion Graph

#	Test Name	Input	Conditions/Tests	Expected Output
1	View Average Session Completion Graph	Set parameters	Data for Graph generation exists	Generate and display the average session completion graph for the team.
2	Empty Graph	Set parameters	Data for Graph generation does not exist	Display error message and display graph without values

Gym Slot Booking Subsystem

Use Case: View Gym Slot Availability

#	Test Name	Input	Conditions/Tests	Expected Output
1	View Gym Slot Availability	Gym slots bookings.	There exists bookings in some Gym slots.	Display Gym Slots.
2		Gym slots bookings.	No Gym slot bookings found.	Display empty Gym slots.

Use Case: Book Gym Slot

#	Test Name	Input	Conditions/Tests	Expected Output
1	Book Gym Slot	Slot Details	Free slot	Book Gym slot. Display success message.
2		Slot Details	Clashing slot	Display error message – there is a clash with a pre-existing gym slot.

Use Case: Cancel Gym Slot Booking

#	Test Name	Input	Conditions/Tests	Expected Output
1	Cancel Gym Slot Booking	Slot ID	Slot is filled.	Cancel booking. Display empty/available slot.
2		Slot ID	Slot not filled	Display error message – slot is empty.

Transportation Booking Subsystem

Use Case: Submit Transportation Request

#	Test Name	Input	Conditions/Tests	Expected Output
1	Submit Transportation Request	Transportation Details	No clash with existing requests	Publish transportation request. Display success message.
2		Transportation Details	Clash with existing request.	Display error message – request clashes with existing request, cannot publish.

Use Case: View Transportation Request

#	Test Name	Input	Conditions/Tests	Expected Output
1	View Transportation Request	Transportation Requests	There exists some requests	Display Transportation requests.
2		Transportation Requests	No Requests	Display message – no transportation requests

Use Case: Delete Transportation Request

#	Test Name	Input	Conditions/Tests	Expected Output
1	Delete Transportation Request	Request ID	Request exists	Delete Request. Display success message.
2		Request ID	Request not found	Display error message – request does not exist

Use Case: Cancel Booking

#	Test Name	Input	Conditions/Tests	Expected Output
1	Cancel Booking	Booking ID	Booking exists	Cancel Booking. Display success message.
2		Booking ID	Booking not found	Display error message – booking does not exist

Use Case: Submit Transportation Confirmation

#	Test Name	Input	Conditions/Tests	Expected Output
1	Submit Transportation Confirmation	Request ID	Request exists	Generate new Transportation Booking. Display success message.
2		Request ID	Request not found	Display error message – request does not exist

Gantt Chart & Resource Breakdown

Resource	Task/Phase	Quantity	Cost per Unit
Servers	Hosting	2	R50 000/year
MySQL Database	Database	1	Open Source
Third-Party APIs	Services	1	R 23 00/year
Front-end (Web) Developer	Design and Construction	850 hours	R250/hour
Front-end (Mobile) Developer	Design and Construction	720 hours	R280/hour
Back-end Developer	Design and Construction	960 hours	R300/hour
DevOps Engineer	Deployment and Maintenance	680 hours	R200/hour

Task	Estimated Hours	Cost per Hour	Total Cost
Requirements Extraction	150	R200	R30 000
Identification of Use Cases	100	R200	R20 000
Feasibility and Risk Study	96	R200	R20 000
Use Case Descriptions	80	R200	R20 000
Activity Diagrams	92	R200	R20 000
Database Design	40	R1200	R48 000
UI/UX Design	180	R250	R45 000
Class Diagram	110	R200	R22 000
Interaction Sequence Diagrams	110	R200	R22 000
Component and Deployment Diagram	90	R200	R18 000
Back-end Development	960	R300	R288 000
Front-end (Web) Development	850	R250	R212 500
Front-end (Mobile) Development	720	R280	R201 600
Unit Testing	72	R90	R6 480
User Acceptance Testing	90	R70	R6 300
Environment Setup	110	R200	R22 000
Security and Compliance Checks	200	R800	R160 000

Performance Evaluation	90	R200	R18 000
Configuration Management	72	R120	R8 640
Deployment Execution	110	R250	R27 500
Monitoring and Logging	96	R300	R28 800
User Guide	50	R60	R3 000
Total			**R1 247 820**

	Cost
Infrastructure and Services	R123 000
Operational Costs	R245 000
Detailed Task Breakdown	R1 247 820
Total Project Cost	**R1 615 820**

UJ Sport Management System

Team 26: Mint

SIMPLE GANTT CHART by Vertex42.com
https://www.vertex42.com/ExcelTemplates/simple-gantt-chart.html

Project start: Tue, 2/27/2024
Display week: 1

TASK	ASSIGNED TO	PERCENT COMPLETE	START	END
Deliverable 1: Requirements and Analysis				
Background, Problem Statement, & Proposed Solution	Palla	100%	2/27/24	3/17/24
Requirements Extraction	Tshepo	100%	3/18/24	3/23/24
Identification and Prioritization of Use Cases	Pachie	100%	3/22/24	4/1/24
Feasibility and Risk Study	Morena	100%	3/28/24	3/30/24
Problem Presentation	All	100%	4/1/24	4/8/24
Deliverable 2: Specification				
Update Problem Statement, Proposed Solution and Use Cases	All	100%	4/8/24	4/10/24
Use Case Descriptions	Tshepo	100%	4/10/24	4/13/24
Activity Diagrams	Morena	100%	4/13/24	4/18/24
Database Design	Palla	100%	4/14/24	4/20/24
Iterative Prototype	Pachie	100%	4/14/24	4/21/24
Presentation	All	100%	4/21/24	4/22/24
Deliverable 3: Architecture Design				
Update Problem Statement, Proposed Solution and Use Cases	Palla	100%	4/22/24	4/24/24
Class Diagram	Tshepo	100%	4/24/24	4/28/24
Interaction Sequence Diagrams	Morena	100%	4/28/24	5/3/24
Iterative Prototype	Pachie	100%	4/28/24	5/5/24
Presentation	All	100%	5/5/24	5/8/24
Deliverable 4: Project Management				
Update Problem Statement, Proposed Solution and Use Cases	All	100%	5/8/24	5/8/24
Component and Deployment Diagram	Palla	100%	5/8/24	5/19/24
Comprehensive Verification and Validation Plan	Tshepo	100%	5/8/24	5/19/24
Gantt Chart and Resource Breakdown	Morena	100%	5/8/24	5/19/24
Iterative Prototype	Pachie	100%	5/8/24	5/19/24
Presentation	All	100%	5/19/24	5/20/24
Updating Design				
Update Problem Statement, Proposed Solution and Use Cases	All	90%	5/20/24	5/22/24
Update Database Schema	All	80%	5/22/24	5/26/24
Update API endpoints design	All	75%	5/26/24	5/31/24
Update UI design	All	85%	5/26/24	5/31/24
Back-end Development				
User Management Subsystem	Morena	0%	6/17/24	6/24/24
Performance Management Subsystem	Morena	0%	6/24/24	7/4/24
Team Management Subsystem	Morena	0%	7/4/24	7/8/24
Injury Management Subsystem	Morena	0%	7/8/24	7/11/24
Announcements Subsystem	Morena	0%	7/11/24	7/14/24
Event/Calendar Scheduling Subsystem	Tshepo	0%	6/17/24	6/25/24
Individual Training Session Subsystem	Tshepo	0%	6/25/24	7/1/24
Attendance Management Subsystem	Tshepo	0%	7/1/24	7/8/24
Gym Slot Booking Subsystem	Tshepo	0%	7/8/24	7/11/24
Transport Booking Subsystem	Tshepo	0%	7/11/24	7/14/24
Front-end(Web) Development				
User Management Subsystem	Pachie	0%	6/17/24	6/24/24
Performance Management Subsystem	Pachie	0%	6/24/24	7/4/24
Team Management Subsystem	Pachie	0%	7/4/24	7/8/24
Injury Management Subsystem	Pachie	0%	7/8/24	7/11/24
Announcements Subsystem	Pachie	0%	7/11/24	7/14/24
Event/Calendar Scheduling Subsystem	Palla	0%	7/14/24	6/25/24
Individual Training Session Subsystem	Palla	0%	6/25/24	7/1/24
Attendance Management Subsystem	Palla	0%	7/1/24	7/8/24
Gym Slot Booking Subsystem	Palla	0%	7/8/24	7/11/24
Transport Booking Subsystem	Palla	0%	7/11/24	7/14/24
Integration with Back-end	All	0%	6/17/24	7/14/24
Front-end(Mobile) Development				
Authentication and User Profile viewing	Pachie	0%	7/22/24	8/2/24
Performance Entries Feature	Morena	0%	8/2/24	8/9/24
Attendance Submission Feature	Tshepo	0%	8/2/24	8/9/24
Event/Calendar Viewing	Tshepo	0%	8/9/24	8/16/24
Individual Training Session Completion Form	Palla	0%	8/2/24	8/9/24
Announcements Viewing	Morena	0%	8/9/24	8/16/24
Integration with Back-end	All	0%	7/22/24	8/16/24
Testing				
Unit Testing	All	0%	8/16/24	8/17/24
Integration Testing	All	0%	8/16/24	8/17/24
User Acceptance Testing	All	0%	8/17/24	8/19/24
Deployment				
Environment Setup	All	0%	8/19/24	8/20/24
Security and Compliance Checks	Morena	0%	8/20/24	8/25/24
Performance Evaluation	Tshepo	0%	8/25/24	8/27/24
Configuration Management	Pachie	0%	8/27/24	8/28/24
Deployment Execution	All	0%	8/27/24	8/29/24
Monitoring and Logging	Palla	0%	8/29/24	9/30/24
Alpha System				
Feature Completion	All	0%	8/30/24	9/2/24
Integration Testing	All	0%	9/2/24	9/4/24
Bug Fixing	All	0%	9/2/24	9/4/24
Functional and Usability Testing	All	0%	9/4/24	9/6/24
Feedback Collection	All	0%	9/6/24	9/8/24
Stabilization and Performance Optimization	All	0%	9/6/24	9/8/24
Presentation	All	0%	9/10/24	9/13/24
Beta System				
Feedback Collection and Analysis	All	0%	9/16/24	9/20/24
Bug Fixing and Refinement	All	0%	9/20/24	9/27/24
Regression Testing	All	0%	9/27/24	9/29/24
Presentation	All	0%		
Complete Documentation				
Reviewing and Editing existing Documentation	All	0%	9/22/24	9/25/24
Completing a User Guide	Tshepo	0%	9/25/24	10/3/24
Completing Installation and Deployment Guides	Pachie	0%	9/25/24	10/3/24
Completing a Developer Guide	Morena	0%	9/25/24	10/3/24
Creating Release notes(version history, change logs & upgrade instructions)	Palla	0%	9/25/24	10/3/24
Screencast Presentation				
Planning, Scripting and Tool selection	All	0%	10/3/24	10/4/24
Recording the Screencast	Pachie & Tshepo	0%	10/4/24	10/5/24
Editing	Morena & Palla	0%	10/5/24	10/7/24
Review and Feedback from mentor	All	0%	10/7/24	10/10/24
Presentation	All	0%	10/10/24	10/15/24
Marketing Website				
Market Research and Marketing Strategy Development	All	0%	9/29/24	9/29/24
Branding	All	0%	9/29/24	9/30/24
Content Creation(brochures, website)	All	0%	9/30/24	10/4/24
Advertising and promoting	All	0%	10/8/24	10/11/24
Final System				
Planning and Preparation	All	0%	10/12/24	10/12/24
Rehearsal and Demonstration practice	All	0%	10/12/24	10/14/24
Feedback Collection from supporting audience	All	0%	10/14/24	10/15/24
Presentation	All	0%	9/16/24	10/18/24