
Software Requirements and Design Document

for

Next Gen Soccer

Prepared by Muhammad Zayan (22i-1010)

Muhammad Rohaan Khalid (22i-1034)

Talha Khurram (22i-0790)

FAST NUCES

11/25/2024

Table of Contents

Table of Contents	ii
1. Introduction.....	1
1.1 Purpose.....	1
1.2 Product Scope.....	1
1.3 Title.....	1
1.4 Objectives	2
1.5 Problem Statement	2
2. Overall Description.....	3
2.1 Product Perspective.....	3
2.2 Product Functions	3
2.3 List of Use Cases	4
2.4 Extended Use Cases	4
2.5 Use Case Diagram	19
3. Other Nonfunctional Requirements.....	21
3.1 Performance Requirements	21
3.2 Safety Requirements.....	22
3.3 Security Requirements.....	23
3.4 Software Quality Attributes.....	25
3.5 Business Rules	27
3.6 Operating Environment	29
3.7 User Interfaces	31
4. Domain Model	34
5. System Sequence Diagram	35
6. Sequence Diagram.....	56
7. Class Diagram.....	102
8. Package Diagram.....	Error! Bookmark not defined.
9. Deployment Diagram.....	Error! Bookmark not defined.

1. Introduction

In urban recreational facilities, managing football turfs and player activities often involves outdated and disjointed systems. Turf bookings are typically handled manually, using phone calls or spreadsheets, while player data such as attendance, skill levels, and performance metrics are tracked separately. This fragmented approach leads to inefficiencies, such as booking conflicts, underutilized facilities, and increased workload for managers and coaches.

Next Gen Soccer aims to address these challenges by providing an integrated, automated solution. This system simplifies turf booking with real-time availability updates, prevents scheduling conflicts, and streamlines player management by centralizing data on attendance, skills, and performance. It bridges the gap between facility management and player coordination, ensuring seamless communication among coaches, players, and administrators.

Built with scalable and modern technologies like cloud databases, web interfaces, and mobile apps, this system offers a user-friendly platform to optimize resource utilization and improve overall performance tracking. By combining turf management with player-related features, the solution is set to enhance the operational efficiency of sports complexes and football academies, paving the way for more organized and effective management practices.

1.1 Purpose

The purpose of the **Next Gen Soccer** is to streamline the management of football facilities and player-related activities through an integrated, automated platform. By addressing common challenges like booking conflicts, inefficient resource utilization, and fragmented player tracking, the system aims to enhance operational efficiency and coordination. It provides real-time turf availability, simplifies scheduling, and consolidates player data, enabling coaches, players, and managers to focus on improving performance and maximizing facility usage.

1.2 Product Scope

The **Next Gen Soccer** is designed to serve urban sports facilities, including recreational complexes and football academies, by integrating turf booking and player management into a single, cohesive platform. It will feature real-time turf availability updates, automated scheduling to prevent booking conflicts, and tools to manage player attendance, skill levels, and performance metrics. The system is scalable, allowing future integration with additional facilities or sports. By combining ease of use with robust functionality, the product aims to optimize resource utilization, enhance coordination, and improve the overall experience for managers, coaches, and players.

1.3 Title

Streamlining Facility Scheduling and Player Performance Tracking for Sports Efficiency.

1.4 Objectives

1. **Simplify Turf Booking:** Develop a user-friendly system to automate turf scheduling, display real-time availability, and eliminate booking conflicts.
2. **Integrate Player Management:** Provide tools to track player attendance, skill levels, and performance metrics within the same platform.
3. **Enhance Operational Efficiency:** Reduce manual workload for facility managers and coaches by centralizing turf and player data management.
4. **Improve Resource Utilization:** Optimize the use of football turfs through efficient scheduling and better organization of activities.
5. **Enable Seamless Communication:** Foster collaboration between facility managers, coaches, and players through an integrated and accessible platform.
6. **Scalable Solution:** Create a system that can be expanded to include additional features or adapted for other sports facilities in the future.

1.5 Problem Statement

Managing football turfs and player activities in urban sports facilities often involves inefficient, manual processes. Turf bookings are frequently handled through phone calls, spreadsheets, or paper records, which are prone to errors such as double bookings or miscommunication. These inefficiencies lead to poor resource utilization, with some turfs being overbooked while others remain underutilized. Furthermore, the lack of integration between booking systems and player management tools adds to the administrative burden on facility managers and coaches. Tracking player attendance, skill development, and performance metrics is often done manually or using separate systems, making coordination difficult and time-consuming.

The proposed **Integrated Football Turf Booking and Player Management System** addresses these challenges by automating and centralizing these processes. It provides real-time updates on turf availability to prevent scheduling conflicts and includes features for tracking player data, such as attendance and performance metrics, in a single platform. This integrated approach reduces manual work, enhances organization, and improves overall efficiency for sports complexes and football academies.

Title

Leveraging modern technologies such as cloud databases, mobile apps, and web interfaces makes this system both feasible and scalable. Similar systems implemented in other sports industries serve as a reliable framework for this project. By utilizing these technologies, the system can provide robust, real-time functionality while being cost-effective and adaptable for future enhancements or expansions.

2. Overall Description

2.1 Product Perspective

Next Gen Soccer is a new, standalone solution designed to address the limitations of existing turf booking and player management processes. Unlike current systems that focus solely on one aspect—such as basic booking tools or standalone player tracking applications—this product integrates both functionalities into a unified platform.

The system operates as an independent application, providing tools for scheduling turf usage and managing player data, such as attendance, skills, and performance metrics. It connects facility managers, coaches, and players through a seamless interface, ensuring that all stakeholders can access real-time updates and relevant information.

Additionally, the system is built to be scalable, allowing it to integrate with larger sports management solutions or expand to include additional features, such as equipment tracking or analytics dashboards. Using cloud databases and web/mobile interfaces, the product ensures accessibility and ease of use across devices.

System Diagram (Plain Text Explanation)

- **User Interfaces:** Interfaces for players, coaches, and managers to book turfs, track performance, and access schedules.
- **Cloud Database:** Central repository for storing turf schedules, player profiles, and performance metrics.
- **Booking Engine:** Automated module for handling turf scheduling and resolving conflicts.
- **Player Management Module:** Tracks player attendance, skill levels, and performance data.
- **Reporting & Analytics:** Tools for generating usage reports, performance trends, and optimization suggestions.
- **Integration APIs:** Interfaces to connect with larger sports management systems or external applications if needed.

2.2 Product Functions

Next Gen Soccer provides the following high-level functions:

- **Turf Management:**
 - Real-time turf availability display.
 - Automated scheduling to prevent double bookings.
 - Conflict resolution for overlapping reservations.
- **Player Management:**
 - Maintain player profiles, including attendance, skill levels, and performance data.
 - Track player participation in events or matches.
 - Generate individual and team performance reports.
- **User Roles and Access Control:**
 - Role-based access for managers, coaches, and players to ensure secure and relevant access to information.
- **Reporting and Analytics:**
 - Generate usage reports for turfs and facilities.
 - Analyze player performance and attendance trends.
- **Notification System:**

- Send reminders for upcoming bookings or events.
 - Notify users of any changes or conflicts in schedules.
- **Integration and Scalability:**
 - API support for integration with larger sports management systems.
 - Scalability to include more facilities, sports, or advanced analytics features in the future.

This functionality ensures a streamlined workflow for managing football facilities and players, improving coordination and operational efficiency.

- **Diagram Suggestion (Plain Text Explanation)**
- **Input:** User inputs (booking requests, player data updates).
- **Processing:** Turf management and player tracking modules.
- **Output:** Schedule confirmations, player reports, and analytics insights.
- **Data Storage:** Cloud database for centralized information management.
- **Interfaces:** Web and mobile applications for end-user interaction.

2.3 List of Use Cases

- 1) **Registration/Login**
- 2) **Upload Turf Information**
- 3) **Finding and booking a turf**
- 4) **Cancelling a turf booking**
- 5) **Turf Removal**
- 6) **Weather Monitoring**
- 7) **Player performance tracking and rankings**
- 8) **Match Performing history**
- 9) **Cancelling turf booking**
- 10) **Turf Removal**

2.4 Extended Use Cases

Use Case Name: Registration/Login
Scope: Field Management System
Level: User Goal
Primary Actor: Player

Stakeholders and Interests:

- **Players/Team Managers:** Want to securely create accounts or log in to manage turf bookings and access personal profiles.
- **System Administrator:** Wants to ensure that user data is handled securely and that access control is enforced.

Pre-Condition:

- The user must have access to the Turf Booking System interface.
- The system must be operational and connected to the database.

Post-Condition:

- The user is either successfully registered or logged in, gaining access to their account and system features.

Main Success Scenario:**User Action**

1. User navigates to the registration/login page.
2. User selects the option to register.
3. User enters valid information and submits the form.
4. System creates a new user account and sends a confirmation email.
5. User returns to the login page.
6. User enters credentials and submits the login form.
7. User gains access to their account.

System Response

1. System displays the registration/login form.
2. System prompts for user details (username, email, password).
3. System validates input and checks if the email is unique.
4. User receives a registration confirmation message.
5. System displays the login form.
6. System verifies credentials and authenticates the user.
7. System redirects the user to their dashboard or homepage.

Extensions:

1a. User submits the registration form with a duplicate email or invalid data.

1a. TheSystem displays an error message indicating the issue (e.g., "Email already in use" or "Invalid input").

- **Special Requirements:**
- The system should implement secure password storage (e.g., hashing).
- The system should support email confirmation to verify user registration.

Use Case Name: Upload turf information													
Scope: Turf booking system													
Level: User Goal													
Primary Actor: Turf owner													
Stakeholders and Interests: <ul style="list-style-type: none"> • Turf Owner/Manager: Wants to provide accurate and detailed information about the turf for potential bookings. • Players/Teams: Seek to find available turfs with relevant details for their bookings. • System Administrator: Aims to maintain the integrity and accuracy of turf information within the system. 													
Pre-Condition: <ul style="list-style-type: none"> • The Turf Owner/Manager must be authenticated and authorized to upload turf information. • The system must be operational and connected to the database. 													
Post-Condition: <ul style="list-style-type: none"> • The turf information is successfully uploaded and stored in the system, making it available for players/teams to view and book. 													
Main Success Scenario: <table border="0"> <thead> <tr> <th>User Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1. Turf Owner navigates to the Upload Turf page.</td><td>1. System displays the turf upload form.</td></tr> <tr> <td>2. Turf Owner fills in the turf details (name, location, dimensions, availability, price, etc.).</td><td>2. System validates the entered information.</td></tr> <tr> <td>3. Turf Owner uploads images of the turf.</td><td>3. System confirms the image format and size.</td></tr> <tr> <td>4. Turf Owner submits the turf information.</td><td>4. System saves the turf information to the database.</td></tr> <tr> <td>5. System displays a success message confirming the upload.</td><td>5. Turf information is now available for browsing.</td></tr> </tbody> </table>		User Action	System Response	1. Turf Owner navigates to the Upload Turf page.	1. System displays the turf upload form.	2. Turf Owner fills in the turf details (name, location, dimensions, availability, price, etc.).	2. System validates the entered information.	3. Turf Owner uploads images of the turf.	3. System confirms the image format and size.	4. Turf Owner submits the turf information.	4. System saves the turf information to the database.	5. System displays a success message confirming the upload.	5. Turf information is now available for browsing.
User Action	System Response												
1. Turf Owner navigates to the Upload Turf page.	1. System displays the turf upload form.												
2. Turf Owner fills in the turf details (name, location, dimensions, availability, price, etc.).	2. System validates the entered information.												
3. Turf Owner uploads images of the turf.	3. System confirms the image format and size.												
4. Turf Owner submits the turf information.	4. System saves the turf information to the database.												
5. System displays a success message confirming the upload.	5. Turf information is now available for browsing.												

Extensions: 1a. Turf owner submits invalid data	1a. System displays error message
<ul style="list-style-type: none"> • Special Requirements: • The upload form should have input validations to ensure data integrity (e.g., correct format, required fields). • The system should provide guidelines for image uploads (dimensions, format) to ensure consistency. • The uploaded turf information should be reviewed by an administrator before it goes live to prevent inappropriate content. 	

Use Case Name: Finding and booking a turf
Scope: Player performance management System
Level: User Goal
Primary Actor: Player
Stakeholders and Interests: <ul style="list-style-type: none"> • Player/Team Captain: Wants to find available turfs based on location, time, and turf quality for upcoming matches or practice sessions. • Turf Owner/Manager: Wants to ensure that their turf is visible to potential players and that the booking system is utilized effectively. • System Administrator: Ensures that the turfs listed are maintained and updated with accurate availability and details.
Pre-Condition: <ul style="list-style-type: none"> • The Player/Team Captain is authenticated and authorized to search for available turfs. • A database of turfs is available, showing their details (e.g., location, availability, quality).
Post-Condition: <ul style="list-style-type: none"> • The Team Captain successfully finds a suitable turf
Main Success Scenario:

User Action	System Response
<p>1. Team Captain logs into the system.</p> <p>3. Player chooses a time slot and confirms availability.</p> <p>5. Player confirms the booking</p> <p>7. Player receives confirmation</p>	<p>2. The system displays available time slots, booking costs, and turf details (size, type, condition, etc.).</p> <p>4. System checks the availability of the chosen time slot and holds it for a short duration while the Player completes the booking process.</p> <p>6. System processes the booking, confirms the availability, and generates a payment request.</p> <p>7. System sends a confirmation notification (via email or app) with the booking details and receipt.</p>
Extensions:	
<p>1a. If no turfs are available</p>	<p>1a. The system displays a message that no turfs are available</p>
<p>Special Requirements:</p> <ul style="list-style-type: none"> The system must support multiple payment options (e.g., credit cards, digital wallets). The system should prevent double booking conflicts by temporarily locking a slot while it's being booked. 	

Use Case Name: Cancelling a turf booking	
Scope: Turf management System	
Level: User Goal	
Primary Actor: Player	
Stakeholders and Interests: <ul style="list-style-type: none"> • Player/Team Manager: Wants to cancel a booking easily and get a refund, if applicable. • Turf Owner: Needs to release the turf slot for other potential bookings. • System Administrator: Manages the processing of cancellations and adjustments to the booking schedule. 	
Pre-Condition: <ul style="list-style-type: none"> • The player or team manager has an active booking for a turf. • The system must allow cancellations based on predefined rules (e.g., cancellation deadline, refund policy). 	
Post-Condition: <ul style="list-style-type: none"> • The booking is canceled, the turf is made available for others, and any refunds are processed according to the policy. 	
Main Success Scenario:	
User Action	System Response
<ol style="list-style-type: none"> 1. Player/Team Manager accesses the "My Bookings" section. 3. Player/Team Manager clicks the "Cancel Booking" button. 5. Player/Team Manager receives confirmation of cancellation. 	<ol style="list-style-type: none"> 1. The system displays the booked turfs 4. System shows the cancellation option 6. System makes the turf available again

Extensions: 1a. If the cancellation is not available	1a. The system displays a message that notifies that cancellation is not an option and also displays the reason
Special Requirements: <ul style="list-style-type: none"> • The system should notify the turf owner and adjust the booking calendar in real-time. • The refund policy should be clearly displayed to the player before confirming the cancellation. 	

Use Case Name: Removing Turf
Scope: Field Management System
Level: User Goal
Primary Actor: Turf owner
<ul style="list-style-type: none"> • Stakeholders and Interests: • Groundskeeper: Needs to remove the turf efficiently, ensuring that the area is prepared for future use or replacement. • Sports Facility Manager: Aims to ensure that turf removal is completed without damaging the underlying field, allowing for quick and cost-effective turf replacement or other usage. • Players/Teams: Want the removal process to be done smoothly and quickly to avoid prolonged field downtime. • System Administrator: Ensures the system provides accurate scheduling and task management for turf removal.
<ul style="list-style-type: none"> • Pre-Conditions: • The Groundskeeper is authenticated and authorized to remove turf. • Turf health data, availability, and maintenance history are accessible to inform the turf removal decision. • The system is operational and connected to the database to track and record the removal process.

<ul style="list-style-type: none"> • Post-Conditions: • The turf is successfully removed, and the system logs the task for future reference. • The area is marked as unavailable until the field is ready for use again (e.g., new turf installation, alternative activity). 	
Main Success Scenario:	
User Action	System Response
1. Groundskeeper navigates to the Turf Removal section.	1. System displays available fields and current turf conditions.
2. Groundskeeper selects a specific turf area for removal.	2. System displays a detailed view of the selected turf, including past maintenance activities.
3. Groundskeeper initiates the turf removal process.	3. System prompts for confirmation, ensuring that the groundskeeper is aware of any dependencies (e.g., scheduled events, weather conditions).
4. Groundskeeper confirms the removal task.	4. System validates that there are no conflicting activities (e.g., booked events) and that weather conditions are suitable for turf removal.
5. Groundskeeper assigns staff to the task.	5. System updates the task schedule and assigns resources to the task.
6. Groundskeeper monitors the removal process via the system.	6. System tracks the progress and updates the field status to "Under Maintenance" or "Unavailable."
7. System logs the task completion once removal is finished.	7. The system updates the database, marking the turf area as "Removed" and providing options for the next steps (e.g., field restoration, new turf installation).
Extensions:	
1a. Turf removal is initiated when the turf is in use.	1a. System displays an error
<ul style="list-style-type: none"> • Special Requirements: • Integration with event scheduling to prevent removal during critical usage periods (e.g., games or tournaments). • Real-time weather monitoring to ensure the optimal timing for turf removal. • Detailed logging and reporting features for tracking completed removal tasks and preparing for future field restoration or turf installation. • Ability to track resource usage (e.g., staff and equipment) during the turf removal process. 	

Use Case Name: Weather Monitoring and Forecasting
Scope: Field Management System
Level: User Goal

Primary Actor: Groundskeeper	
Stakeholders and Interests: <ul style="list-style-type: none"> • Groundskeeper: Uses weather data to optimize maintenance scheduling and avoid turf damage. • Sports Facility Manager: Interested in preventing weather-related damage and ensuring field readiness for events. • Players/Teams: Rely on well-maintained fields, even in adverse weather conditions. 	
Pre-Condition: The Groundskeeper is authenticated and authorized to access weather data. The system is integrated with a reliable weather forecasting service.	
Post-Condition: Groundskeeper uses the weather forecast to adjust maintenance schedules and avoid overuse during adverse conditions.	
Main Success Scenario:	
User Action	System Response
1. Groundskeeper accesses the Weather Monitoring module. 3. Groundskeeper reviews upcoming weather patterns. 5. Groundskeeper adjusts maintenance schedules based on weather data.	2. System displays current and forecasted weather data. 4. System provides detailed forecasts, including precipitation, temperature, and wind speed. 6. System confirms changes and updates the schedule accordingly.

Extensions: 1a. Groundskeeper encounters an unexpected weather event that is not reflected in the system.	1a. System sends an alert and provides alternative options for field maintenance.
Special Requirements: <ul style="list-style-type: none"> • Real-time weather data integration. • System should support automatic adjustments to maintenance schedules based on severe weather warnings. 	

Use Case Name: Player performance tracking and rankings	
Scope: Player performance management System	
Level: User Goal	
Primary Actor: Player	
Stakeholders and Interests: <ul style="list-style-type: none"> · Player: Wants to track personal performance metrics and see how they rank compared to others. · Coaches/Team Managers: Use the system to assess player performance and select top-performing players. · Turf Manager: Needs to maintain player performance records and rankings for organizational purposes. 	
Pre-Condition: The player is registered and logged into the system. Performance metrics such as goals, assists, tackles, or fitness data are available and updated regularly.	
Post-Condition: Player performance data is logged, and the system ranks players accordingly. Players can view their progression and compare their performance with others.	
Main Success Scenario:	
User Action	System Response

<p>1. Player logs into the system and accesses the performance tracking module.</p> <p>3. Player selects a specific time period (e.g., last week, last month, season).</p> <p>5. Players check the rankings tab to compare their stats with other players</p> <p>7. Player views details about their progression over time (e.g., improvement in goals or fitness scores).</p>	<p>2. The system displays the player's performance dashboard, showing key metrics such as goals, assists, tackles, and fitness data.</p> <p>4. The system updates the performance dashboard to show the player's statistics over the selected time period.</p> <p>6. The system displays a leaderboard, showing rankings based on individual performance metrics.</p> <p>8. The system generates a report of the player's performance history, highlighting areas of strength and weakness.</p>
<p>Extensions:</p> <p>1a. If the players performance data is not available</p>	<p>1a. The system displays a message that the data is missing</p>
<p>Special Requirements:</p> <ul style="list-style-type: none"> · The system should update performance metrics automatically based on game data input. · The ranking system must be dynamic, recalculating player rankings in real time as new performance data is added. · The performance history should be visualized using charts or graphs to show progression over time. · The system should ensure privacy by allowing players to opt-out of the public leaderboard. 	

Use Case Name: Match performance history

Scope: Player performance management System

Level: User Goal	
Primary Actor: Player	
Stakeholders and Interests: <ul style="list-style-type: none">· Player: Wants to track match participation to assess consistency and experience.· Coach/Team Manager: Uses the data to evaluate player reliability and contributions to the team.· Turf Manager: Maintains player records and tracks match participation statistics.	
Pre-Condition: The player is registered in the system, and match data is regularly updated.	
Post-Condition: Match participation history is logged and available for review.	
Main Success Scenario:	
User Action	System Response
<ol style="list-style-type: none">1. Player logs into the system and navigates to the match participation history.3. Player filters the matches by date or competition5. Player requests details of a specific match to review their contribution.	<ol style="list-style-type: none">2. The system displays the player's history of match participation, including date, opponent, and match outcomes.4. The system updates the match history according to the selected filters.6. The system provides detailed match stats, including minutes played, goals, assists, and other relevant data.

Extensions: 1a. If the players performance data is not available	1a. The system displays a message that the data is missing
Special Requirements: <ul style="list-style-type: none"> • The system should automatically update match participation data after each game. • Players should be able to export their match history for external analysis or record-keeping. • Privacy controls should allow players to keep participation history private or share it with the team. 	

Use Case Name: Finding a player	
Scope: Player performance management System	
Level: User Goal	
Primary Actor: Player	
Stakeholders and Interests: <ul style="list-style-type: none"> • Team Captain: Needs to find suitable players to fill specific roles on the team based on skills, experience, and availability. • Players: Want to be considered for teams where they can showcase their skills and gain match experience. • Coaches: Interested in finding the best mix of players for training and team-building purposes. 	
Pre-Condition: <ul style="list-style-type: none"> • The Team Captain is authenticated and authorized to search for players on the platform. • A database of registered players is available with their positions, skills, and stats. 	
Post-Condition: <ul style="list-style-type: none"> • The Team Captain successfully identifies and contacts suitable players for potential recruitment to their team. 	
Main Success Scenario:	
User Action	System Response

<ol style="list-style-type: none"> 2. Team Captain logs into the system. 3. Captain selects "Find Player for Team" from the menu. 5. Captain reviews player profiles and compares their stats. 7. Captain selects a player and sends a recruitment request. 9. Player accepts the request. 	<ol style="list-style-type: none"> 2. The system authenticates the Team Captain and provides access to the player search feature. 4. System displays a search form with filters like position, skill level, availability, and match stats. 6. System sends a notification to the selected player, requesting their interest in joining the team. 7. System updates the team's roster with the new player and sends confirmation to both parties.
<p>Extensions:</p> <ol style="list-style-type: none"> 1a. If no players are available 2b. If player rejects the invitation 	<ol style="list-style-type: none"> 1a. The system displays a message that no players are available 2b. Then the system notifies the captain
<p>Special Requirements:</p> <ul style="list-style-type: none"> • The system should support real-time updates to player profiles and rankings based on recent matches. • Integration with player fitness data (e.g., match history, injury status) to ensure accurate player availability. • Notifications via email or the app to inform players of team recruitment requests. 	

Use Case Name: Feedback Submission

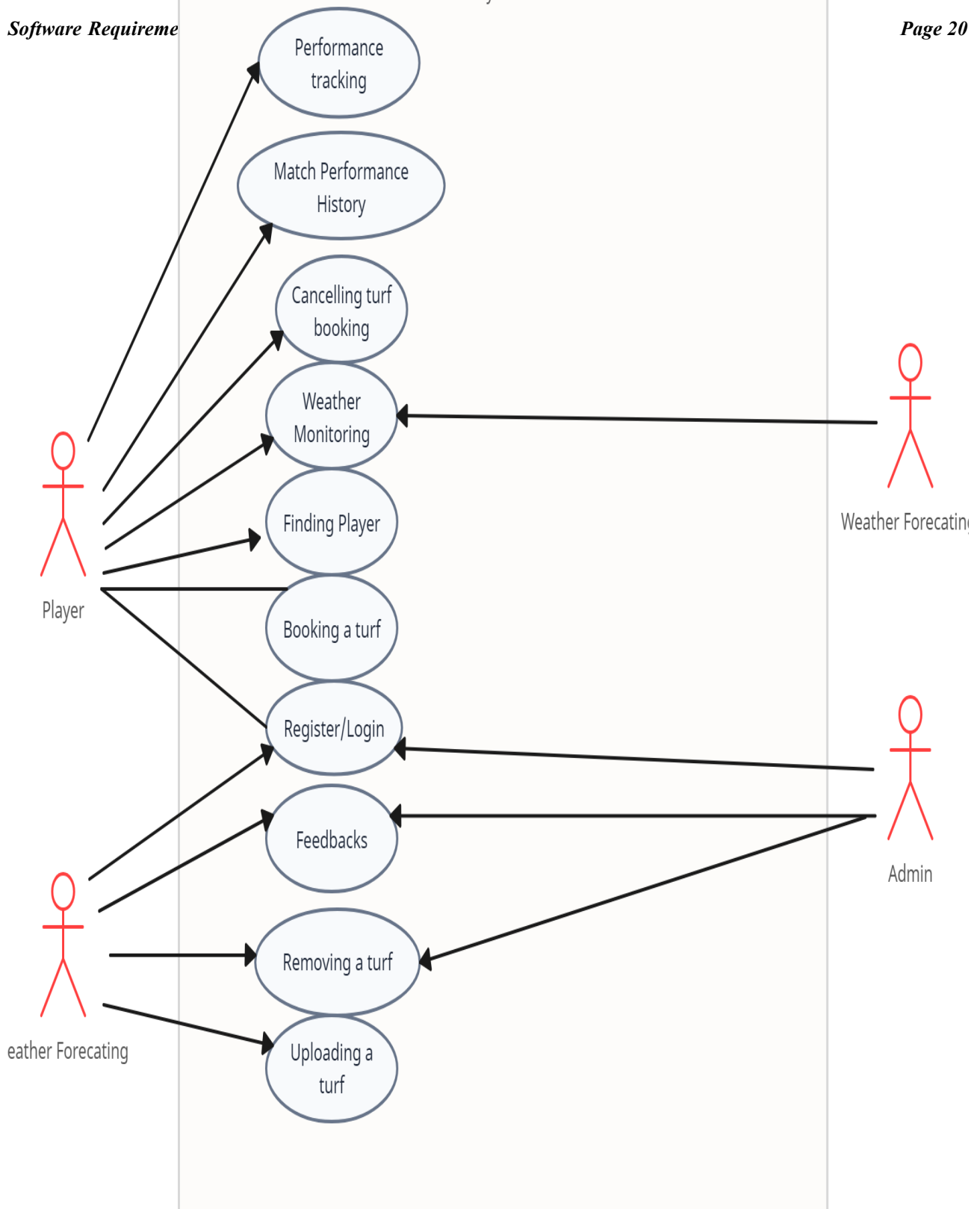
Scope: Field Management System

Level: User Goal											
Primary Actor: Player/Turf Owner											
Stakeholders and Interests: <ul style="list-style-type: none"> • Users (Players/Teams): Want to provide feedback on turfs to help others make informed decisions and improve the quality of service. • Turf Owners: Aim to receive constructive feedback to enhance their services and facilities. • System Administrator: Responsible for maintaining the feedback system and addressing user concerns. 											
Pre-Condition: <ul style="list-style-type: none"> • The User must be authenticated (logged in) to submit feedback. • The system must be operational and accessible. 											
Post-Condition: <ul style="list-style-type: none"> • The feedback is successfully submitted and stored in the system, allowing for future review and analysis. 											
Main Success Scenario: <table> <thead> <tr> <th>User Action</th><th>System Response</th></tr> </thead> <tbody> <tr> <td>1. User navigates to the Feedback section of system.</td><td>1. System displays the feedback submission form.</td></tr> <tr> <td>2. User fills out the feedback form (rating, comments).</td><td>2. System validates the input fields for completeness.</td></tr> <tr> <td>3. User submits the feedback form.</td><td>3. System stores the feedback in the database.</td></tr> <tr> <td>4. System displays a success message.</td><td>4. System may send a confirmation email to the user.</td></tr> </tbody> </table>		User Action	System Response	1. User navigates to the Feedback section of system.	1. System displays the feedback submission form.	2. User fills out the feedback form (rating, comments).	2. System validates the input fields for completeness.	3. User submits the feedback form.	3. System stores the feedback in the database.	4. System displays a success message.	4. System may send a confirmation email to the user.
User Action	System Response										
1. User navigates to the Feedback section of system.	1. System displays the feedback submission form.										
2. User fills out the feedback form (rating, comments).	2. System validates the input fields for completeness.										
3. User submits the feedback form.	3. System stores the feedback in the database.										
4. System displays a success message.	4. System may send a confirmation email to the user.										
Extensions: 1a. Invalid data entered	1a. System displays an error message.										

- **Special Requirements:**

- The feedback form should include a rating system (e.g., 1 to 5 stars) and a comment box for additional input.
- The system should ensure that users can only submit one feedback entry per turf per visit to prevent spam.
- User anonymity should be an option, allowing users to choose whether their feedback is public or private.
- An administrator interface should be available for reviewing and moderating feedback submissions.

2.5 Use Case Diagram



3. Other Nonfunctional Requirements

3.1 Performance Requirements

The performance requirements for the **Next Gen Soccer** system are outlined below, addressing system behavior under various circumstances. These requirements aim to ensure the system provides a seamless experience for users and supports the intended functionality efficiently.

- **3.1.1 User Interface Responsiveness**

- **Requirement:** All user interface interactions (e.g., form submissions, button clicks) must respond within **1 second** under normal load conditions.
- **Rationale:** Quick responses enhance user experience and prevent frustration during actions like booking or canceling a turf.

- **3.1.2 Authentication Performance**

- **Requirement:** User authentication (login/registration) should complete within **2 seconds** for 95% of requests under peak load.
- **Rationale:** Ensures smooth access for users and avoids delays that could discourage engagement.

- **3.1.3 Turf Booking and Cancellation**

- **Requirement:** The system must process turf booking or cancellation requests within **3 seconds**, including database updates and confirmation display.
- **Rationale:** Quick processing is critical during high-demand periods when users compete for available slots.

- **3.1.4 Weather Monitoring**

- **Requirement:** Weather information must be fetched from the external Weather API and displayed within **2 seconds**.
- **Rationale:** Timely updates ensure users have accurate and current weather data for planning.

- **3.1.5 Player Performance and Match History Retrieval**

- **Requirement:** Player performance data and match history should load within **3 seconds** for an individual player.
- **Rationale:** Reduces delays in data retrieval, supporting an efficient analysis and ranking experience.

- **3.1.6 Database Operations**

- **Requirement:** All database operations (e.g., queries for turf data, player stats) must execute within **100 milliseconds** for 90% of transactions.
- **Rationale:** Ensures backend processes are fast enough to support frontend responsiveness and handle concurrent requests.

- **3.1.7 Scalability**

- **Requirement:** The system must support up to **500 concurrent users** with minimal degradation in performance (<10% increase in response time under peak load).
- **Rationale:** Accommodates growth in user base and ensures a consistent experience during peak usage, such as tournament seasons.

- **3.1.8 External API Integration**

- **Requirement:** The Weather Monitoring Service must handle up to **100 API calls per minute** without exceeding a latency of **1 second** per request.
 - **Rationale:** Maintains performance even with frequent updates requested by multiple users.
-

- **3.1.9 Error Handling and Recovery**

- **Requirement:** The system must recover from minor errors (e.g., database disconnection or Weather API timeout) within **5 seconds** and display fallback messages to users.
 - **Rationale:** Ensures reliability and user trust even during transient issues.
-

- **3.1.10 Resource Utilization**

- **Requirement:** The system should utilize server resources efficiently, keeping **CPU usage below 75%** and **memory usage below 70%** during peak load.
 - **Rationale:** Prevents server overloads and ensures smooth performance for all users.
-

These performance requirements are designed to meet the expectations of diverse users, ensure reliability, and provide a high-quality user experience.

3.2 Safety Requirements

The **Next Gen Soccer** system includes several safety requirements to prevent potential harm, safeguard user data, and comply with industry standards. These requirements ensure the system is designed and used responsibly.

- **3.2.1 Data Security**

- **Requirement:** User authentication credentials (e.g., passwords) must be encrypted using industry-standard hashing algorithms (e.g., bcrypt).
 - **Rationale:** Protects sensitive user information from unauthorized access in the event of a data breach.
-

- **3.2.2 Data Backup**

- **Requirement:** The system must implement automated backups of critical data (e.g., user details, turf bookings, match histories) at least once daily.
 - **Rationale:** Prevents data loss due to system failures or cyberattacks.
-

- **3.2.3 System Access Control**

- **Requirement:** Only authorized users should access administrative functions such as turf removal or performance data updates.
 - **Rationale:** Prevents accidental or malicious alterations to sensitive information.
-

- **3.2.4 Prevention of Overbooking**

- **Requirement:** The system must verify turf availability in real time to prevent overbooking conflicts.
 - **Rationale:** Ensures fairness and avoids user disputes.
-

- **3.2.5 Weather Data Accuracy**

- **Requirement:** The system must include a disclaimer stating that weather data is sourced from external APIs and is subject to change.
 - **Rationale:** Prevents liability in case of inaccurate weather forecasts affecting users' plans.
-

- **3.2.6 User Input Validation**

- **Requirement:** All user inputs (e.g., registration forms, turf details) must be validated to prevent SQL injection, XSS attacks, or accidental data corruption.
 - **Rationale:** Ensures system integrity and protects against common cyber threats.
-

- **3.2.7 Session Management**

- **Requirement:** User sessions must expire after **15 minutes** of inactivity to prevent unauthorized access from unattended devices.
 - **Rationale:** Mitigates the risk of unauthorized use on shared or public devices.
-

- **3.2.8 Compliance with Data Protection Laws**

- **Requirement:** The system must comply with relevant data protection regulations such as **GDPR** (General Data Protection Regulation) or **CCPA** (California Consumer Privacy Act).
 - **Rationale:** Protects user privacy and avoids legal penalties.
-

- **3.2.9 System Monitoring and Alerts**

- **Requirement:** The system must log critical events (e.g., failed login attempts, turf overbooking attempts) and alert administrators in real time.
 - **Rationale:** Detects potential security threats or system misuse promptly.
-

- **3.2.10 Certification Standards**

- **Requirement:** The system must meet security and safety certifications such as **ISO/IEC 27001** for information security management.
 - **Rationale:** Ensures the system adheres to recognized safety and security standards.
-

- **3.2.11 Protection Against Server Overload**

- **Requirement:** Implement rate limiting to restrict excessive requests (e.g., API calls or turf search queries) from a single user or IP address.
 - **Rationale:** Prevents denial-of-service (DoS) attacks and ensures system availability for all users.
-

These safety requirements help mitigate risks to users, data, and the system itself while ensuring compliance with legal and industry standards.

3.3 Security Requirements

The **Next Gen Soccer** system is designed to prioritize user security and data privacy, adhering to best practices and regulatory requirements. The following security requirements ensure the system remains robust against potential threats.

- **3.3.1 User Identity Authentication**

- **Requirement:** All users must authenticate using a unique username/email and a password. Passwords must meet complexity requirements (minimum 8 characters, including uppercase, lowercase, numeric, and special characters).
 - **Rationale:** Prevents unauthorized access and enhances account security.
-

- **3.3.2 Multi-Factor Authentication (MFA)**

- **Requirement:** Support MFA as an optional security layer, requiring a one-time code sent to the user's registered email or phone.
 - **Rationale:** Reduces risks associated with compromised credentials.
-

- **3.3.3 Data Encryption**

- **Requirement:**

- All sensitive data in transit must be encrypted using **TLS 1.2 or higher**.
 - Data at rest, including user credentials and payment information (if applicable), must be encrypted with **AES-256**.
 - **Rationale:** Protects data integrity and confidentiality during transmission and storage.
-

- **3.3.4 Role-Based Access Control (RBAC)**

- **Requirement:** Implement RBAC to restrict access to specific features and data based on user roles (e.g., admin, player, manager).
 - **Rationale:** Limits exposure of sensitive features to authorized personnel only.
-

- **3.3.5 Secure API Communication**

- **Requirement:** All API endpoints must require authentication tokens (e.g., OAuth 2.0) and validate them before processing requests.
 - **Rationale:** Prevents unauthorized API usage and ensures secure client-server communication.
-

- **3.3.6 Privacy by Design**

- **Requirement:** The system must minimize data collection and only request data strictly necessary for functionality.
 - **Rationale:** Adheres to privacy principles such as **GDPR** and reduces risks of misuse or breaches.
-

- **3.3.7 Regular Security Audits**

- **Requirement:** Conduct quarterly security audits to identify vulnerabilities and ensure compliance with security policies.
 - **Rationale:** Maintains system security by addressing potential weaknesses proactively.
-

- **3.3.8 Logging and Monitoring**

- **Requirement:** Log all critical activities (e.g., login attempts, data access, booking cancellations) and monitor logs for suspicious activities.
 - **Rationale:** Provides visibility into system activities and aids in detecting breaches.
-

- **3.3.9 Protection Against Common Threats**

- **Requirement:** Implement measures to defend against common vulnerabilities, including:
 - SQL Injection
 - Cross-Site Scripting (XSS)
 - Cross-Site Request Forgery (CSRF)
 - Distributed Denial of Service (DDoS) attacks
 - **Rationale:** Ensures the system is resilient against widespread attack vectors.
-

- **3.3.10 User Data Anonymization**

- **Requirement:** Anonymize user data in analytics and backups to protect privacy and comply with data protection regulations.
 - **Rationale:** Prevents unauthorized identification of users from stored or shared data.
-

- **3.3.11 Secure Password Storage**

- **Requirement:** Store user passwords using a secure, one-way hashing algorithm (e.g., bcrypt) with appropriate salting.
 - **Rationale:** Protects user credentials even in the event of a data breach.
-

- **3.3.12 Incident Response Plan**

- **Requirement:** Develop and maintain an incident response plan to handle security breaches, including notifying affected users and relevant authorities.
- **Rationale:** Ensures swift and effective action during security incidents.

-
- **3.3.13 Compliance with Standards**
 - **Requirement:** Adhere to industry security standards and regulations, including:
 - **ISO/IEC 27001:** Information Security Management
 - **GDPR:** General Data Protection Regulation (if applicable)
 - **CCPA:** California Consumer Privacy Act (if applicable)
 - **Rationale:** Ensures the system meets legal and industry benchmarks for security.
-

These security requirements ensure that **Next Gen Soccer** remains secure, reliable, and compliant with privacy standards while fostering user trust.

3.4 Software Quality Attributes

The **Next Gen Soccer** system is designed to meet high standards of software quality to satisfy both customer needs and developer requirements. The following quality attributes are defined to guide development and ensure a robust and user-friendly product.

-
- **3.4.1 Usability**
 - **Requirement:**
 - The system must provide an intuitive interface with a user-friendly navigation structure.
 - 90% of users should be able to complete core tasks (e.g., booking a turf, checking weather) within **2 minutes** of first-time use.
 - **Rationale:** Enhances the overall user experience, particularly for non-technical users.
-

- **3.4.2 Availability**
 - **Requirement:**
 - The system must achieve **99.5% uptime** annually, with planned maintenance communicated to users in advance.
 - Recovery from downtime (e.g., server failure) must occur within **15 minutes**.
 - **Rationale:** Ensures continuous accessibility for users, especially during peak usage periods.
-

- **3.4.3 Reliability**
 - **Requirement:**
 - The system must handle up to **500 concurrent users** without performance degradation or loss of functionality.
 - All critical operations (e.g., bookings, payments) must succeed with an accuracy rate of **99.9%**.
 - **Rationale:** Builds user trust and prevents disruptions in important processes.
-

- **3.4.4 Maintainability**
 - **Requirement:**
 - The system must be modular, allowing updates or fixes to individual components (e.g., weather monitoring) without affecting other parts.
 - Developers should be able to add new features with minimal codebase changes within **2 weeks** of development effort.
 - **Rationale:** Facilitates long-term evolution and cost-effective updates.
-

- **3.4.5 Scalability**
- **Requirement:**

- The system must scale to support **1000 concurrent users** with a 10% increase in infrastructure resources.
 - Database performance must remain consistent as the dataset grows to **1 million records**.
 - **Rationale:** Prepares the system for growth in user base and data volume.
-

- **3.4.6 Interoperability**

- **Requirement:**

- The system must integrate seamlessly with external services, such as Weather APIs, and comply with **RESTful API standards** for extensibility.
 - **Rationale:** Ensures compatibility with third-party tools and future integrations.
-

- **3.4.7 Adaptability**

- **Requirement:**

- The system must allow easy configuration to support different regions (e.g., time zones, local weather APIs).
 - **Rationale:** Broadens the applicability of the system to different markets and user groups.
-

- **3.4.8 Portability**

- **Requirement:**

- The system must be deployable on **cloud platforms** (e.g., AWS, Azure) and **on-premises servers** with minimal configuration changes.
 - **Rationale:** Provides flexibility for different deployment environments.
-

- **3.4.9 Testability**

- **Requirement:**

- Each module must achieve a minimum of **90% code coverage** in automated tests.
 - Critical features (e.g., booking, authentication) must pass stress and load testing scenarios with defined thresholds.
 - **Rationale:** Ensures system correctness and resilience to edge cases.
-

- **3.4.10 Robustness**

- **Requirement:**

- The system must handle unexpected inputs (e.g., invalid turf IDs, weather API downtime) gracefully without crashing.
 - Fallback mechanisms must notify users of issues while maintaining core functionality.
 - **Rationale:** Maintains reliability under adverse conditions.
-

- **3.4.11 Correctness**

- **Requirement:**

- The system must ensure all calculations (e.g., player rankings) and data integrity operations are **100% accurate**.
 - Automated validation tests must verify data consistency after every update or maintenance.
 - **Rationale:** Builds user confidence in the accuracy of provided data and features.
-

- **3.4.12 Reusability**

- **Requirement:**

- Core components (e.g., Authentication Service, Weather Monitoring Service) must be designed as independent modules reusable in other systems.
 - **Rationale:** Reduces future development costs and promotes efficient resource utilization.
-

- **3.4.13 Flexibility**

- **Requirement:**
 - Developers should be able to modify existing functionality or add new features (e.g., team management, advanced analytics) without affecting system stability.
 - **Rationale:** Accommodates evolving business requirements and user feedback.
-

These quality attributes ensure the **Next Gen Soccer** system is adaptable, user-friendly, scalable, and robust while supporting long-term development and maintenance needs.

3.5 Business Rules

The **Next Gen Soccer** system operates under a set of business rules that define how users interact with the system and what actions are permitted under specific conditions. These rules ensure smooth operation, compliance with policies, and secure management of the platform's features.

- **3.5.1 User Roles and Permissions**

- **Admin:**

- Can manage all aspects of the system, including turf information, user accounts, and bookings.
- Can approve or remove turf details.
- Can access all match history and player performance data.

- **Manager:**

- Can upload and update turf information, but cannot delete or remove turf details.
- Can view and manage turf bookings for their specific location.
- Cannot access system-wide settings or user management.

- **Player/Customer:**

- Can register, log in, and browse available turf information.
 - Can book or cancel turf bookings within allowed timeframes.
 - Can view their own player performance and match history.
 - Cannot modify other users' data or turf details.
-

- **3.5.2 Turf Availability and Booking**

- **Booking Limit:**

- A player can only book one turf per time slot.
- A booking can be made no more than **14 days in advance**.
- A user can cancel a turf booking **up to 24 hours before** the scheduled time.
- If a player fails to show up for a booking, the system may apply a **late cancellation fee** (if applicable).

- **Turf Booking Conflicts:**

- The system must prevent double bookings for the same time slot and turf.
 - If a booking conflict occurs, the user will be notified and prompted to select an available slot.
-

- **3.5.3 Turf Removal**

- **Turf Removal Authorization:**

- Only Admins have permission to remove turf listings.
 - Managers cannot delete or remove a turf they have uploaded; they can only update its information.
 - A turf can only be removed if no active bookings are associated with it; otherwise, the system will prompt the Admin to either cancel all future bookings or transfer them.
-

- **3.5.4 Weather Data and User Notifications**

- **Weather Alerts:**

- Players will receive a notification about potential weather disruptions **12 hours before** the scheduled turf booking if adverse conditions (e.g., rain, storms) are forecasted.
 - If weather conditions change drastically within **1 hour before** a booking, the system will provide an immediate notification for cancellation or rescheduling.
-

- **3.5.5 Player Performance and Rankings**

- **Performance Tracking:**

- Player performance data is updated only by **match organizers or system-administered events** (e.g., performance in official tournaments).
 - Players cannot manually adjust their performance metrics.
 - Rankings are calculated based on predefined rules such as goals scored, matches played, and performance metrics (e.g., pass accuracy, goals per game).
-

- **3.5.6 Account Management**

- **User Registration:**

- Players must provide a valid email address for account creation, which will also be used for notifications (e.g., booking confirmation, performance updates).
- Admins can disable or suspend accounts that violate terms of service or engage in fraudulent activity.
- Passwords must be updated every **90 days** for added security.

- **Role Assignment:**

- Admins assign roles to new users.
 - A user cannot change their own role; role modifications must be done by an Admin.
-

- **3.5.7 Payment and Refunds (if applicable)**

- **Booking Payments:**

- Players must complete payment for bookings at the time of reservation.
- Payment is processed through the integrated payment gateway (e.g., credit card, online payment system).
- Players are entitled to a full refund if they cancel within **24 hours** of booking, otherwise, a cancellation fee may apply.

- **Refund Requests:**

- Refunds are processed automatically if the turf is unavailable due to technical issues or unforeseen circumstances (e.g., weather).
-

- **3.5.8 System Maintenance and Downtime**

- **Scheduled Maintenance:**

- Maintenance must be scheduled during off-peak hours to minimize disruption.
- Players will be notified in advance if the system will be unavailable for more than **1 hour** due to maintenance.

- **Unscheduled Downtime:**

- In the event of unexpected downtime, the system must notify users of the outage status, estimated resolution time, and any actions they need to take (e.g., rescheduling bookings).
-

- **3.5.9 Data Privacy and User Consent**

- **Data Consent:**

- Users must explicitly agree to the terms of service and privacy policy before using the system.
- Players can access and update their personal data, but cannot access or modify other users' information.

- User data (e.g., performance, booking history) is retained for **1 year** and automatically purged after this period unless explicitly requested to be retained for future analysis.

These business rules help ensure smooth, secure, and fair operation of the **Next Gen Soccer** system, making the platform functional and aligned with industry best practices.

3.6 Operating Environment

The **Next Gen Soccer** system is designed to operate in a cloud-based environment, supporting both individual users (players and managers) and administrators (admins) across various devices and platforms. The following describes the hardware, operating systems, software components, and infrastructure necessary for the system's effective operation.

- **3.6.1 Hardware Platform**

- **Client-Side Hardware:**

- **Desktop:** The system will be accessible on any desktop or laptop with a modern web browser (e.g., Chrome, Firefox, Safari, Edge). Recommended hardware specifications include:
 - Processor: **Intel i5 or equivalent**
 - RAM: **4 GB minimum**
 - Storage: **50 MB of free disk space** (for browser cache and temporary files)
 - Screen resolution: Minimum **1366 x 768 pixels**
- **Mobile Devices:**
 - The system will be responsive and fully functional on smartphones and tablets with Android 5.0+ or iOS 11+.
 - Recommended screen size: **4.7" or larger** for optimal user experience.

- **Server-Side Hardware:**

- **Cloud Infrastructure:** The system will be hosted on scalable cloud services (e.g., **Amazon Web Services (AWS)**, **Microsoft Azure**, or **Google Cloud Platform (GCP)**), allowing for dynamic scaling based on load and demand.
- Server configuration will be optimized for:
 - **CPU:** Multi-core processors, preferably **Intel Xeon** or equivalent, to handle concurrent user sessions.
 - **RAM:** At least **8 GB** of RAM, with scaling for higher user loads.
 - **Storage:** Cloud-based storage solutions (e.g., **AWS S3**, **Azure Blob Storage**) for user data, turf information, and booking histories.
 - **Network:** Redundant network infrastructure with **99.9% uptime** guarantee for high availability.

- **3.6.2 Operating System**

- **Server Operating System:**

- The application will run on cloud-based Linux servers, utilizing **Ubuntu 20.04 LTS** or equivalent for stability and security.
- Alternatively, **CentOS 8** or **Debian 10** could be used based on organizational preference.
- **Windows Server** (2019 or later) can be considered for certain enterprise environments, but Linux-based solutions are preferred for better performance and cost efficiency.

- **Client Operating System:**

- **Desktop Clients:** The system will be compatible with Windows 10+, macOS 10.12+.

- **Mobile Clients:** Android 5.0+ and iOS 11+.

- **3.6.3 Web Server and Application Framework**

- **Web Server:**

- **Nginx** or **Apache HTTP Server** will be used to handle HTTP/HTTPS requests, providing scalability, security, and efficient load balancing.

- **Application Server:**

- The application will run using **Node.js** or **Python** (Django/Flask) for backend services.
- **Ruby on Rails** or **Spring Boot** may be used depending on developer preferences.

- **3.6.4 Database Management System (DBMS)**

- **Database:**

- **MySQL** or **PostgreSQL** will be used for structured data storage (e.g., user accounts, turf booking information, and player performance data).
- For scalability, **Amazon RDS** or **Google Cloud SQL** will be used to host the relational database.

- **Caching:**

- **Redis** or **Memcached** will be used for caching frequently accessed data (e.g., turf availability, user sessions) to improve performance.

- **3.6.5 External Software and Integrations**

- **Payment Gateway:**

- The system will integrate with external payment providers (e.g., **Stripe**, **PayPal**, or **Razorpay**) to handle turf booking payments.

- **Weather API:**

- The system will integrate with a weather API (e.g., **OpenWeatherMap**, **Weatherstack**) to monitor real-time weather conditions and provide weather-related notifications to users.

- **Email Service Provider:**

- **SendGrid** or **Amazon SES** will be used for email notifications (e.g., booking confirmations, weather alerts, password resets).

- **Authentication Services:**

- The system will support authentication through **OAuth 2.0** for third-party integrations (e.g., Google, Facebook) and internal account creation.
- Password storage will use **bcrypt** or **argon2** for hashing and securing user credentials.

- **3.6.6 Security Infrastructure**

- **SSL/TLS Encryption:**

- The entire application will be served over HTTPS with **SSL/TLS** encryption to secure user data during transmission.

- **Firewalls and Intrusion Detection:**

- The cloud environment will be secured with **firewalls** and **intrusion detection/prevention systems (IDS/IPS)** to protect against malicious attacks.

- **3.6.7 Web Browser Compatibility**

- The system will be compatible with the following web browsers:

- **Google Chrome (latest version)**
- **Mozilla Firefox (latest version)**
- **Safari (latest version)**
- **Microsoft Edge (latest version)**

- The system will be **responsive** and optimized for both desktop and mobile views, ensuring compatibility with various screen sizes and device orientations.

-
- **3.6.8 Other Software Dependencies**
 - **Version Control:**
 - The source code will be managed using **Git**, hosted on platforms like **GitHub**, **GitLab**, or **Bitbucket** for version control and collaborative development.
 - **Containerization:**
 - **Docker** containers will be used for development, testing, and deployment, ensuring a consistent environment across different stages of the software lifecycle.
 - **CI/CD:**
 - Continuous integration and deployment will be automated using **Jenkins**, **GitHub Actions**, or **GitLab CI/CD** pipelines to ensure smooth and frequent releases.
-

- **3.6.9 Compliance and Regulatory Requirements**
 - The system must comply with applicable data protection regulations, such as:
 - **GDPR** (General Data Protection Regulation) for European users.
 - **CCPA** (California Consumer Privacy Act) for California-based users.
 - **PCI DSS** standards if payment data is processed.
-

These components, platforms, and standards ensure the **Next Gen Soccer** system operates in a scalable, secure, and efficient environment that supports both end-users and administrators.

3.7 User Interfaces

The **Next Gen Soccer** system is designed to be user-friendly, providing an intuitive experience for players, managers, and admins. The following outlines the logical characteristics of the user interfaces (UIs) for each user role, the design standards to be followed, and the key interface components that will be implemented across the system.

1. 3.7.1 User Roles and Interfaces

2. Admin Interface:

- **Purpose:** Admins manage users, turfs, payments, and system configurations.
- **Core Functions:**
 - User Management (create, update, delete user accounts)
 - Turf Management (upload turf information, update details, remove turf)
 - Viewing and generating reports (player performance, booking history)
- **Key UI Components:**
 - **Dashboard:** Provides an overview of system activity, including recent bookings, turf availability, and user activity.
 - **User Management:** List of all users with options to filter by role (Player/Manager), view details, and update account status.
 - **Turf Management:** Interface for adding and removing turf listings, viewing turf usage, and managing turf details.
 - **Reports and Analytics:** Graphical representation of user activity, turf utilization, and player rankings.

3. Manager Interface:

- **Purpose:** Managers oversee turf bookings and manage specific turf locations.
- **Core Functions:**
 - Turf Information Upload
 - Turf Availability Management
 - Booking Management (view and manage turf bookings)
- **Key UI Components:**

- **Dashboard:** A summary of turf availability, upcoming bookings, and recent activity.
- **Turf Information:** Form to upload turf details (name, size, location, price, availability).
- **Booking Management:** Interface to view current and upcoming bookings, and to manage booking confirmations or cancellations.

4. **Player/Customer Interface:**

- **Purpose:** Players interact with the system to register, view turfs, make bookings, and track their performance.
- **Core Functions:**
 - Account Management (register, log in, view profile)
 - Booking Turf (search for available turfs, book or cancel)
 - Viewing Player Performance and Rankings
 - Match History and Notifications
- **Key UI Components:**
 - **Homepage:** Provides access to turf search, player login, and registration.
 - **Turf Search and Booking:** Interface for searching available turfs by location, date, and time. Displays turf details and booking options.
 - **Performance Dashboard:** Displays personal performance metrics such as goals scored, accuracy, ranking, and history.
 - **Match History:** Allows players to view past matches, performance details, and feedback.

5. 3.7.2 Interface Design Standards

To ensure consistency and usability, the following design standards will be followed across the system:

1. **Layout:**

- **Responsive Design:** The UI will be fully responsive, adapting to different screen sizes and orientations (desktop, tablet, mobile).
- **Navigation Bar:** A sticky navigation bar will be used at the top of each page, providing quick access to major sections (Dashboard, Turf Search, Booking, Profile).
- **Action Buttons:** Common actions (e.g., "Book Turf," "Cancel Booking," "View Details") will be prominently displayed as buttons with clear labels and consistent color schemes.

2. **Color Scheme:**

- The system will follow a **professional and modern color palette** using shades of **blue** (for trust and reliability) and **green** (for sports and energy), with contrasting colors for calls to action (e.g., red for alerts).
- **Backgrounds** will be light (white/gray) with dark text for clarity.

3. **Typography:**

- The primary font will be **Roboto** (or similar sans-serif font) for a clean, modern look.
- Font sizes will follow a clear hierarchy (larger for headings, smaller for body text) to guide the user through the interface.

4. **Buttons:**

- All buttons will have consistent styling, using **rounded corners** with **hover effects** (color change or shadow).
- **Action buttons** (e.g., Book, Cancel, Submit) will be larger and placed in prominent areas of the screen (e.g., at the bottom or center).
- **Cancel buttons** or links will be clearly distinguished using a neutral or red color.

5. **Forms and Inputs:**

- **Form Fields:** All forms will use simple input fields with placeholders, tooltips, and validation messages.

- **Error Handling:** Clear, concise error messages will appear next to the field that caused the error, in red text, with an icon (e.g., exclamation mark).
- **Input Validation:** Real-time validation for fields such as email, phone number, and password strength will be implemented.

6. 3.7.3 Common Features Across Interfaces

1. Search Functionality:

- The search bar will be available across the system (e.g., for searching turfs, users, or match history) and will provide auto-suggestions as users type.
- Search results will be displayed in a clean, paginated format, with sorting options (e.g., by location, date, or popularity).

2. Help and Support:

- A **Help** icon will be available in the top navigation for all users. Clicking this will display a FAQ page or provide links to contact customer support.
- **Tooltips** and **onboarding guides** will be shown to new users to explain how to use key features.

3. Notifications:

- A notification icon will be present in the top navigation for users to view their upcoming bookings, alerts, or system messages (e.g., weather updates, booking confirmations).
- **Real-time notifications** will appear for urgent updates, such as booking changes or match results.

4. Accessibility:

- The system will adhere to WCAG 2.1 accessibility standards, ensuring that it is usable by people with disabilities.
- Features like **screen reader compatibility**, **keyboard navigation**, and **high-contrast themes** will be provided to meet diverse user needs.

5. Mobile Considerations:

- On mobile devices, the system will use a **hamburger menu** for easy access to main sections.
- **Touch-friendly** buttons and controls will be implemented to ensure smooth interaction on smaller screens.

7. 3.7.4 Sample Screens and Layouts

While a full visual specification can be provided in a separate document, the following outlines the key page structures:

1. Homepage (for Players/Customers):

- Top Navigation Bar (Login/Register, Search Turf, Profile)
- Large Hero Section (with search for turf by location/date)
- Footer with Help/Support links

2. Turf Booking Page (for Players):

- Search Results: List of available turfs with brief details (name, location, price, availability)
- Filters: Date, Time, Turf Type
- Booking Form: Selection of date/time, payment details, and booking confirmation button

3. Player Dashboard (for Players):

- Performance Metrics: Stats like total goals, accuracy, and current ranking
- Upcoming Matches: List of booked turfs and dates
- Match History: View past games, performance data

4. Admin Dashboard:

- Overview: Quick stats on user activity, bookings, and turf status
- User Management: List of all users, with filters to search by role, status, etc.

- Reports Section: Graphs and data on turf bookings, player performance, and system analytics

8. 3.7.5 Error Messages and Display Standards

- **Error Message Layout:**

- Error messages will appear in a distinct **red box** at the top or near the affected component.
- The message will be clear, concise, and suggest actions the user can take to resolve the issue (e.g., "Please enter a valid email address").

- **Confirmation Messages:**

- Success or failure messages will appear in a **green box** (for success) or **red box** (for failure), with a clear call to action or next steps.

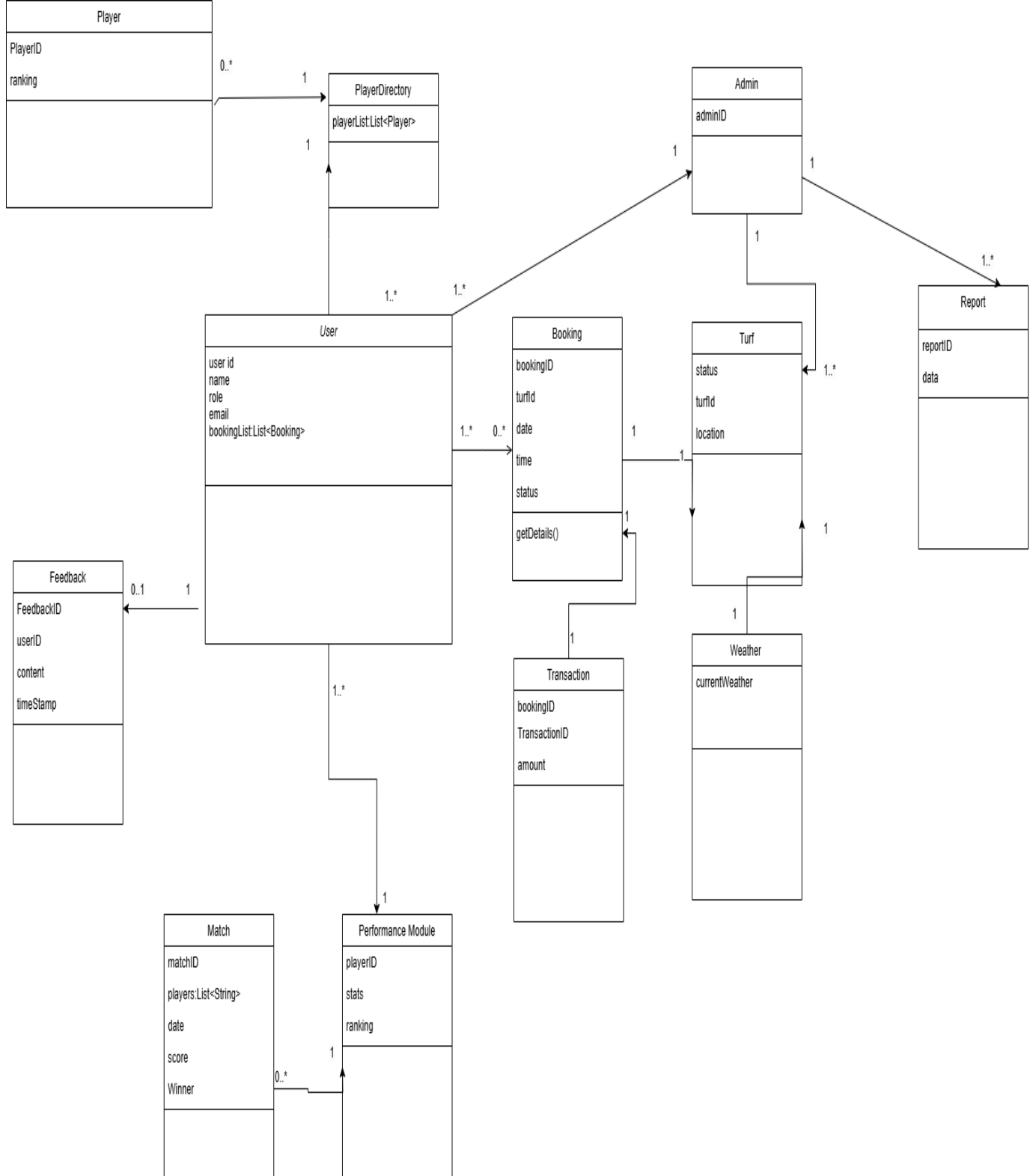
9. 3.7.6 User Interface Specification

A detailed user interface specification document will be created separately, which will include:

- Complete screen layouts and wireframes for each page
- Full UI element styling (buttons, forms, icons, etc.)
- Detailed descriptions of each UI component's behavior and interactions

These guidelines will ensure a consistent, user-friendly experience across all parts of the **Next Gen Soccer** system, enabling users to easily navigate and interact with the platform.

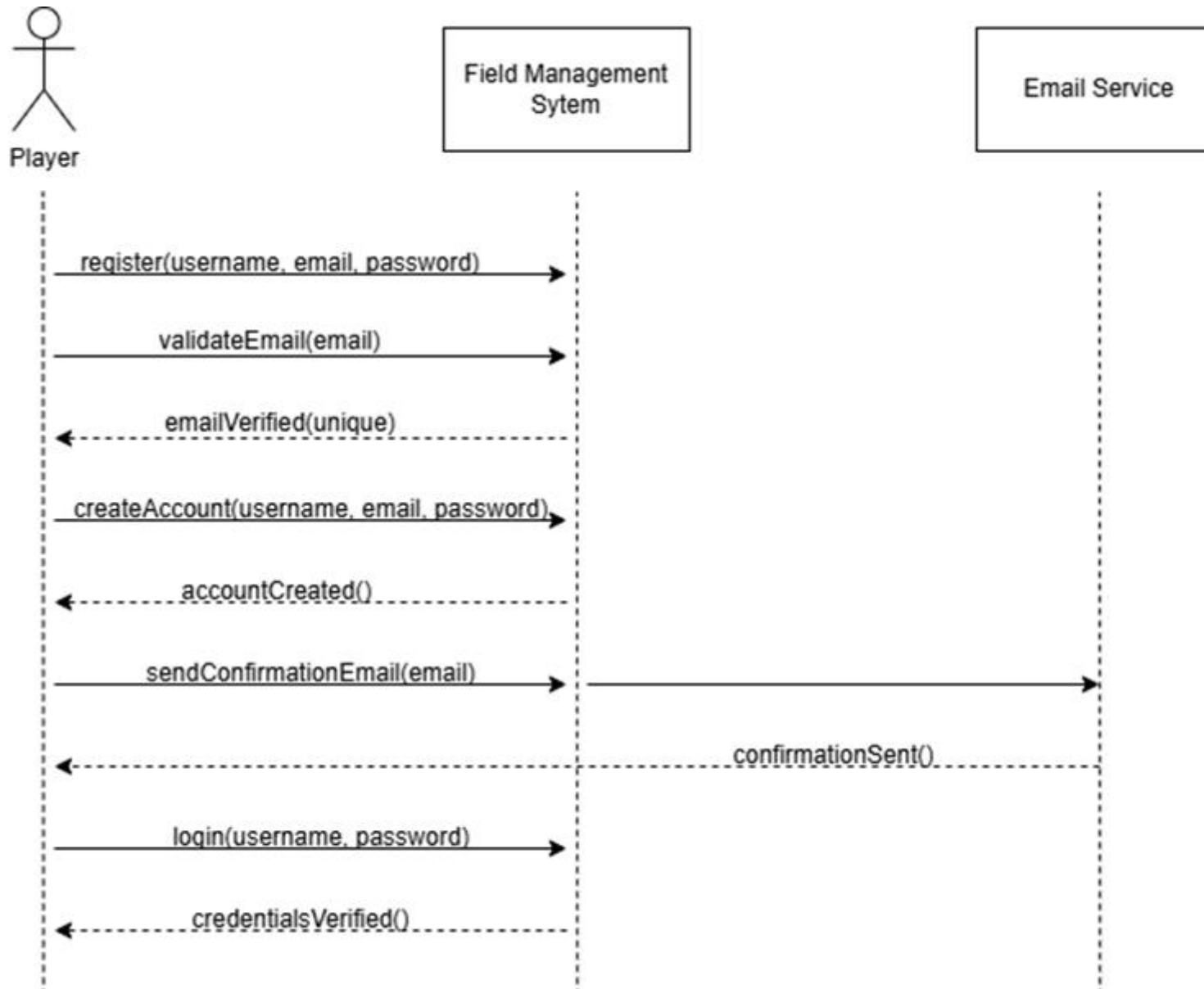
4. Domain Model



5. System Sequence Diagram

1) Registration/Login

Main Success Scenario:



Alternate Scenario:


Player/Turf Owner

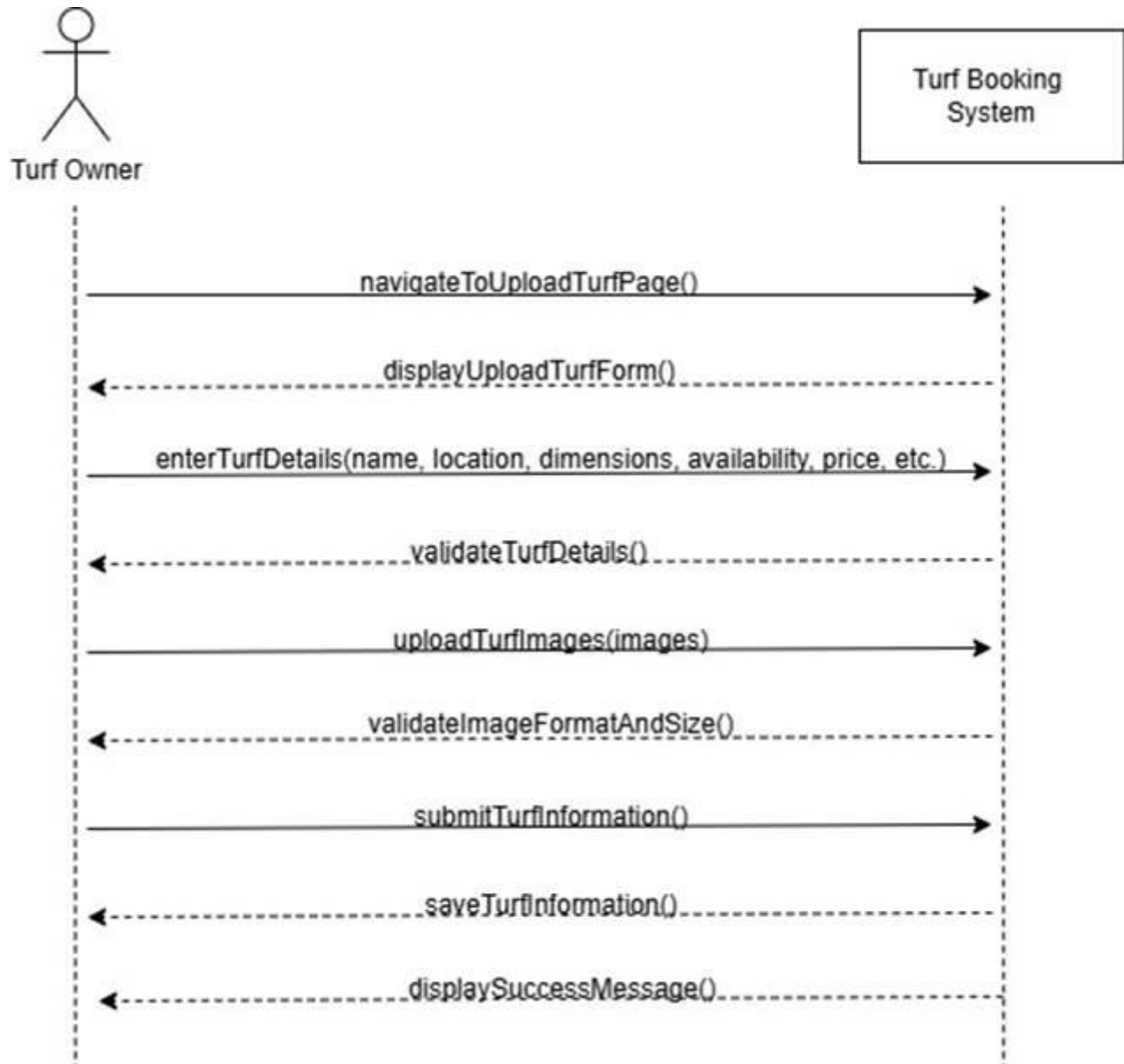
Field Management
System

Page 37

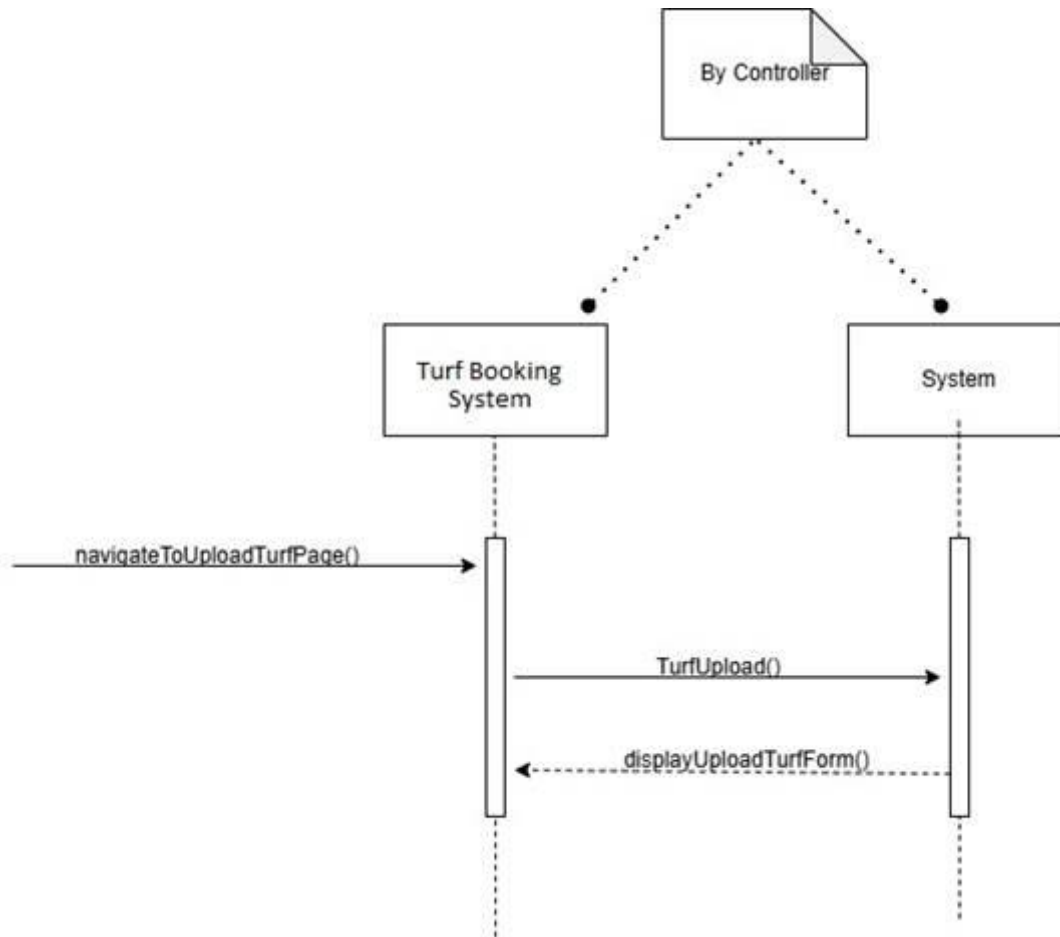


2) Upload Turf Information

Main Success Scenario:

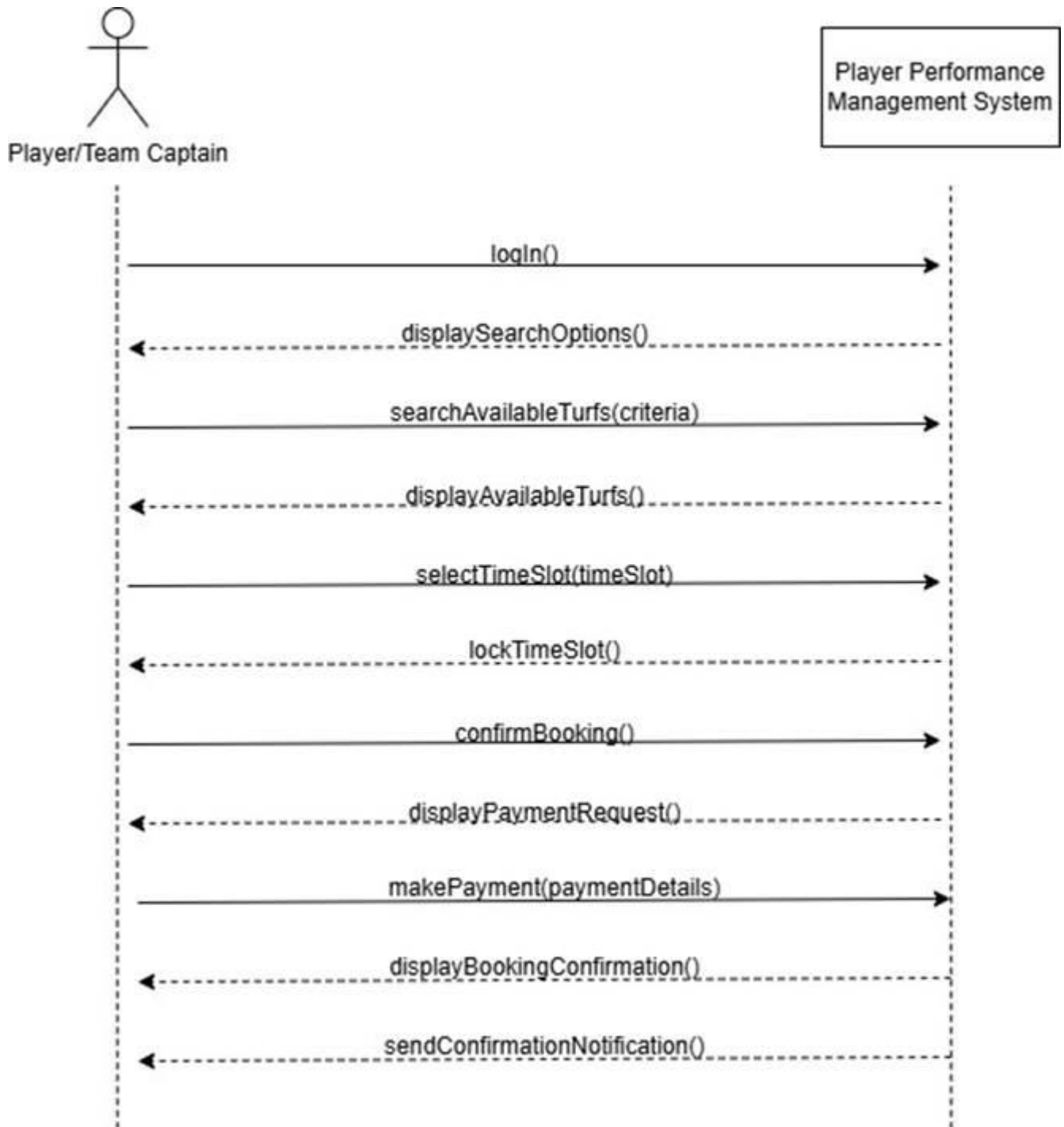


Alternate Scenario:

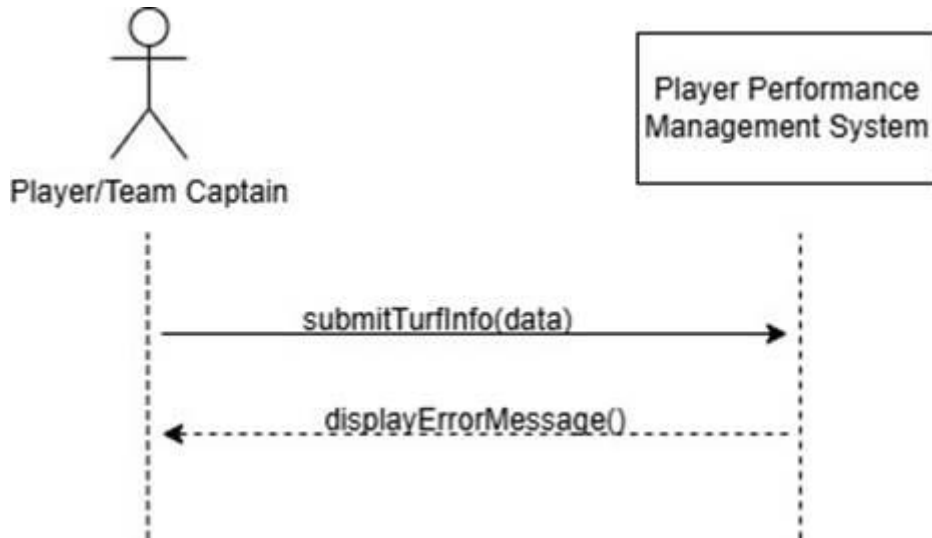


3) Finding and booking a turf

Main Success Scenario:

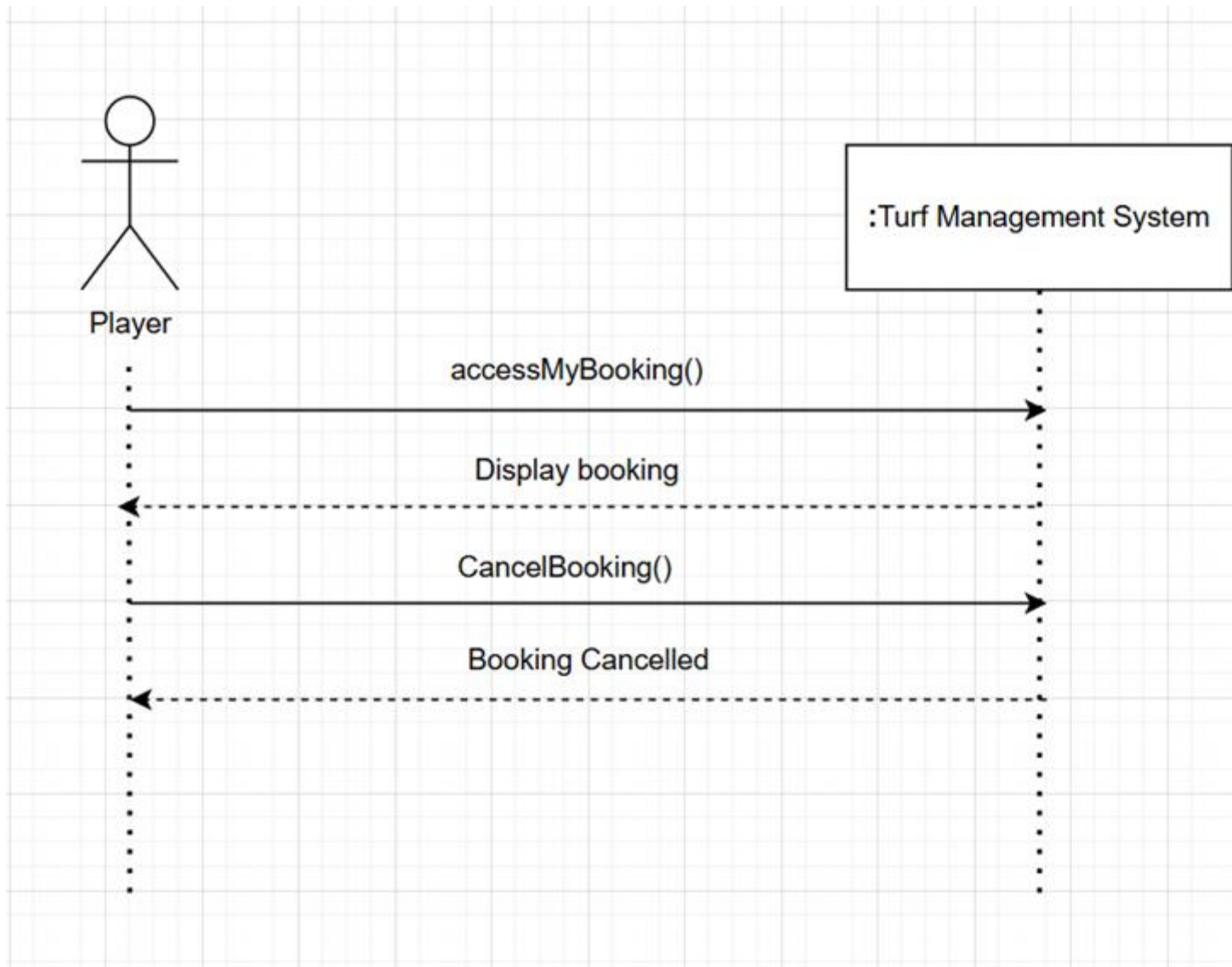


Alternate Scenario:

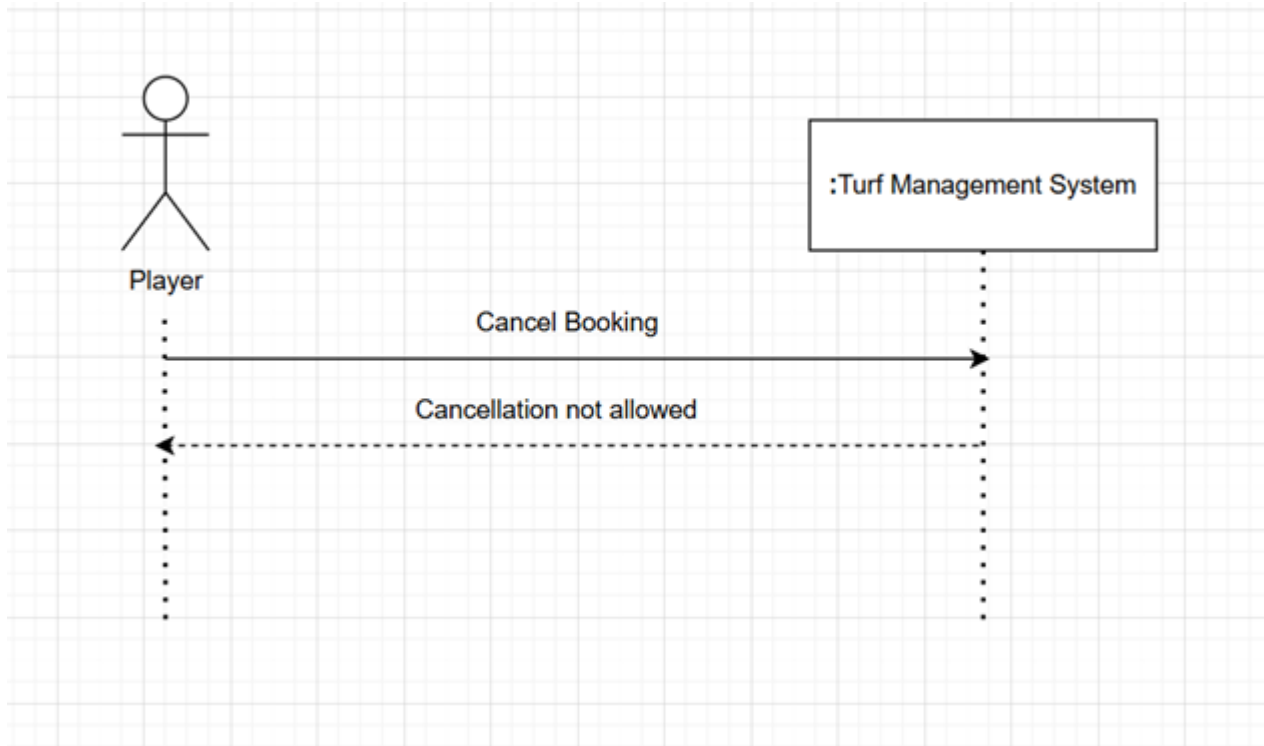


4) Cancelling a turf booking

Main Success Scenario:

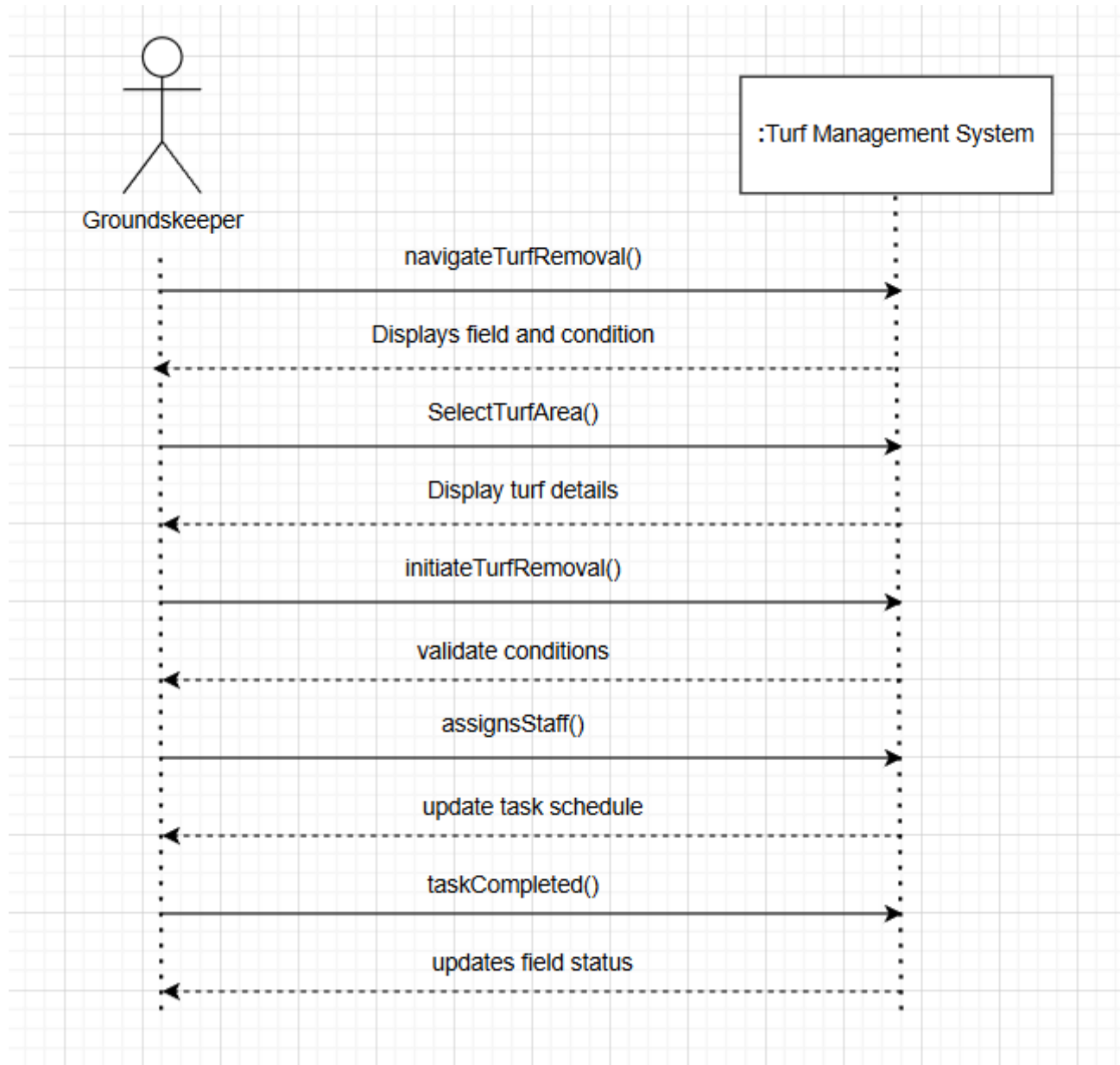


Alternate Scenario:

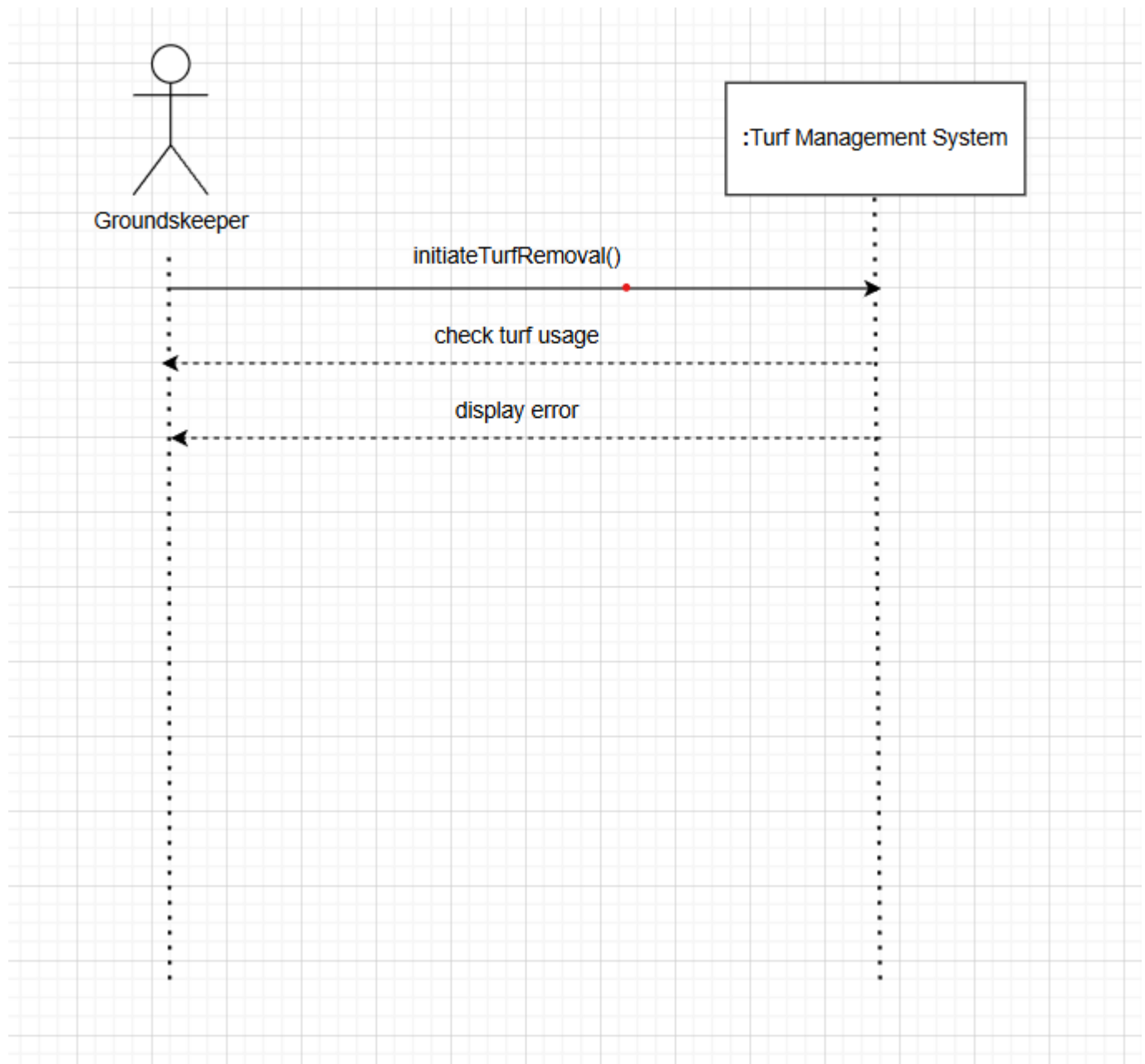


5) Turf Removal

Main Success Scenario:

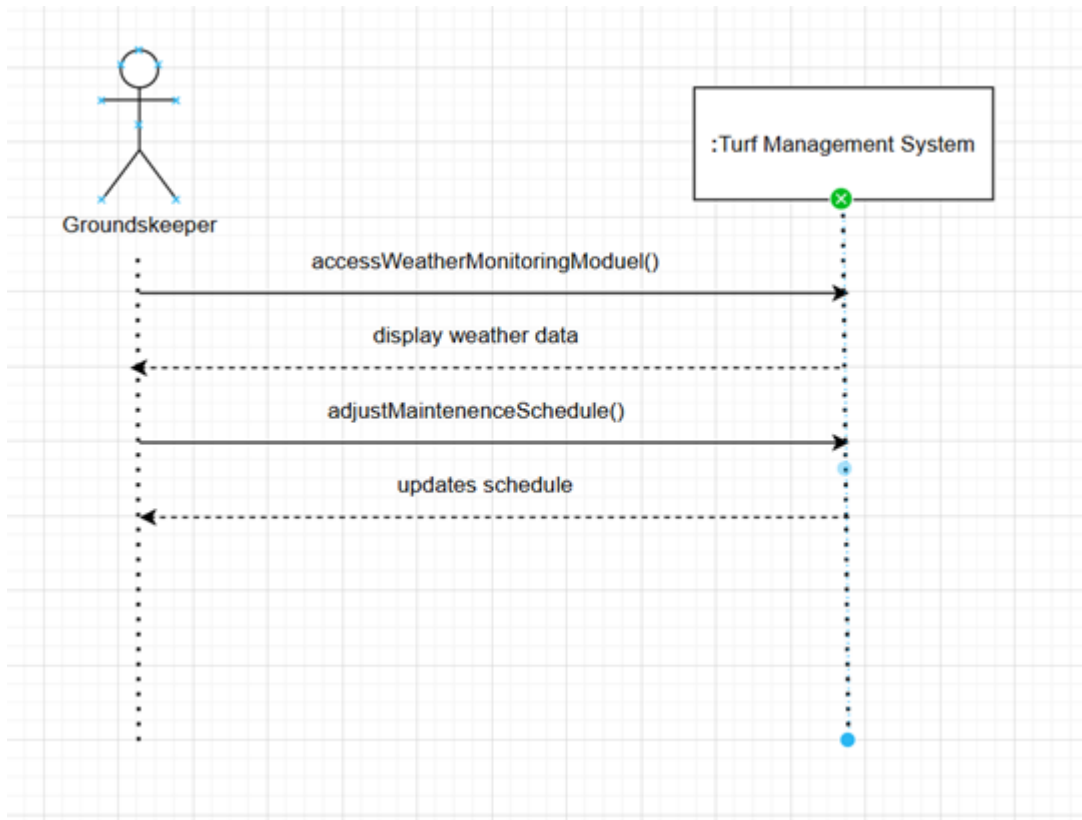


Alternate Scenario:

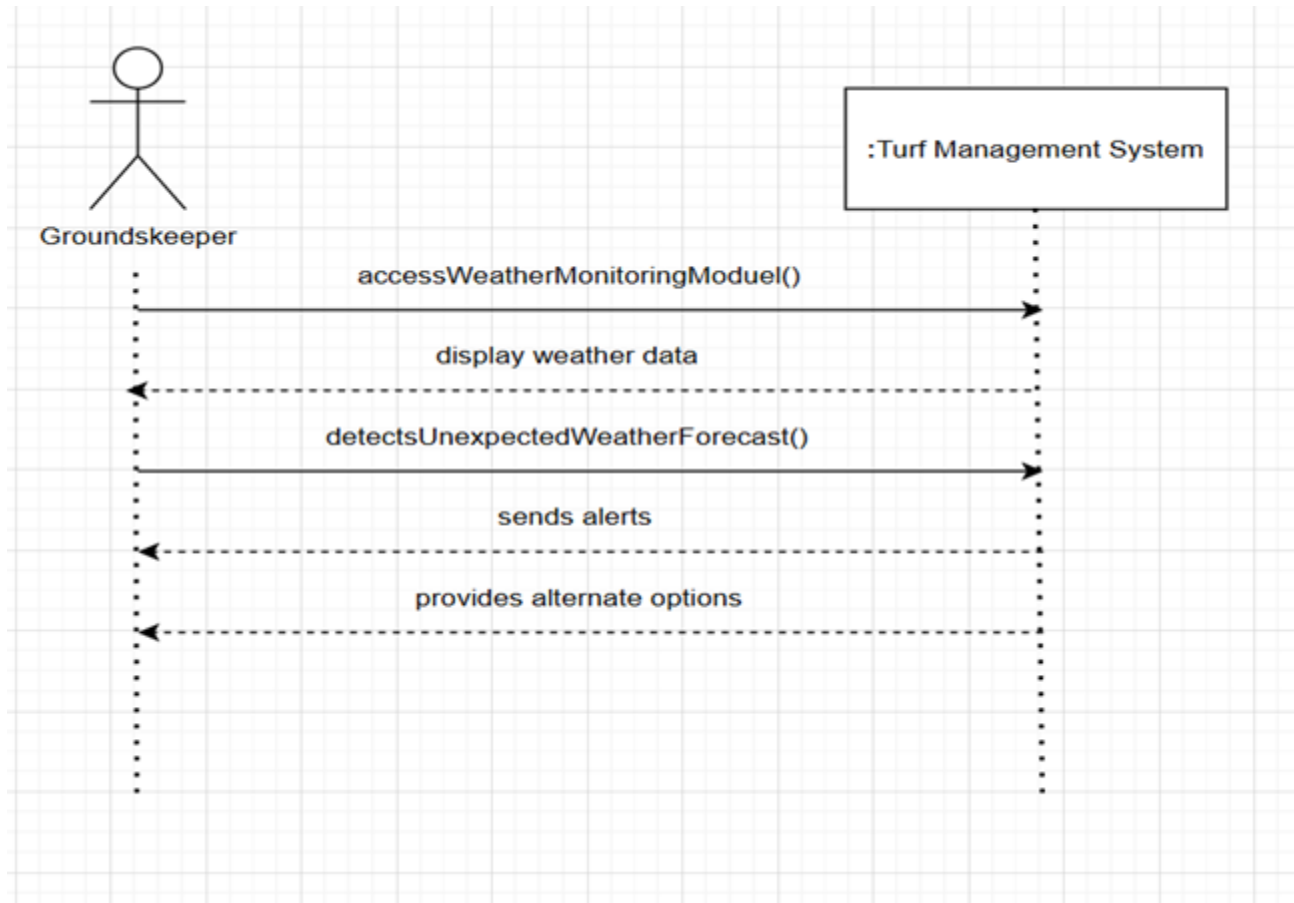


6) Weather Monitoring

Main Success Scenario:

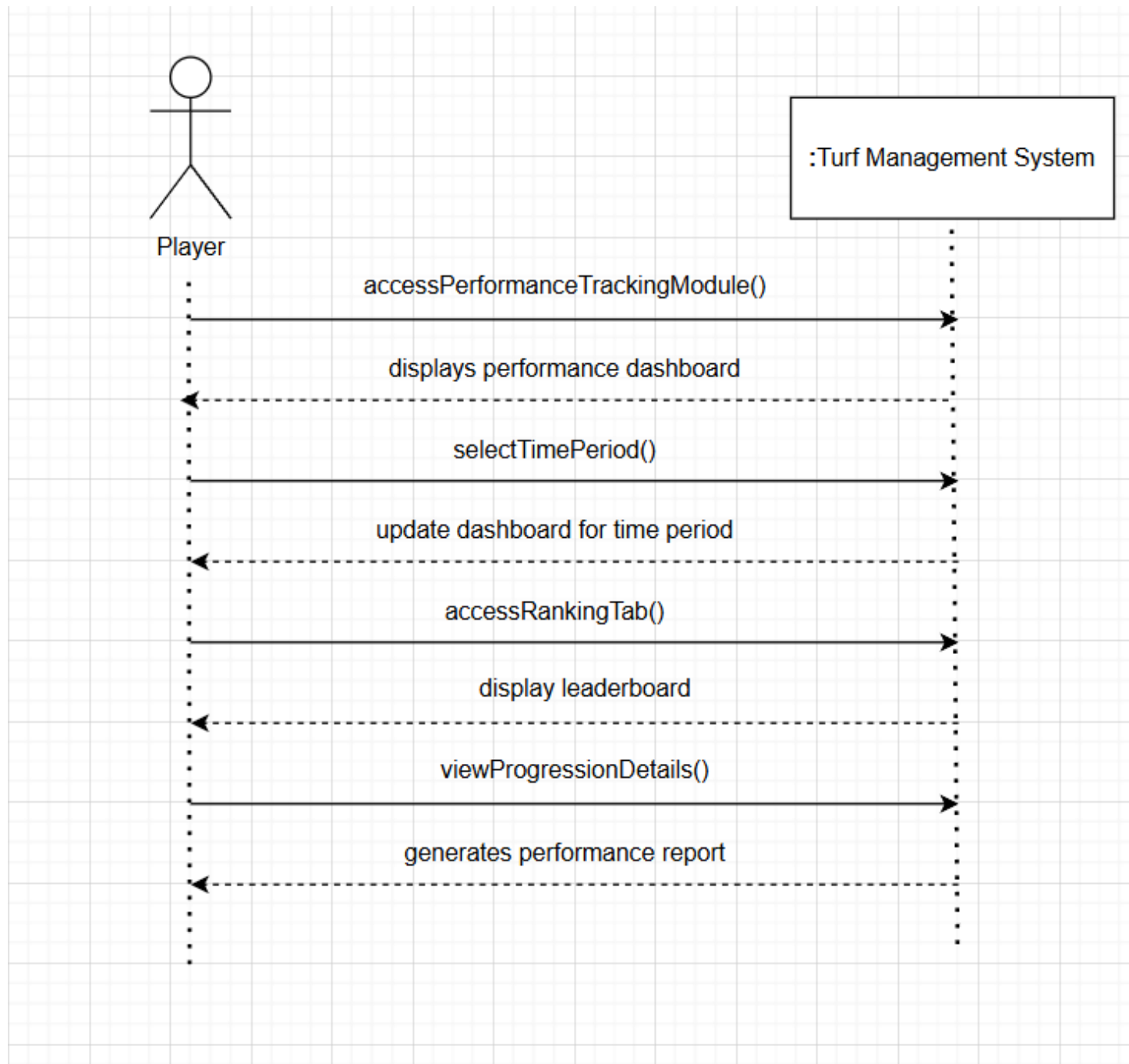


Alternate Scenario:

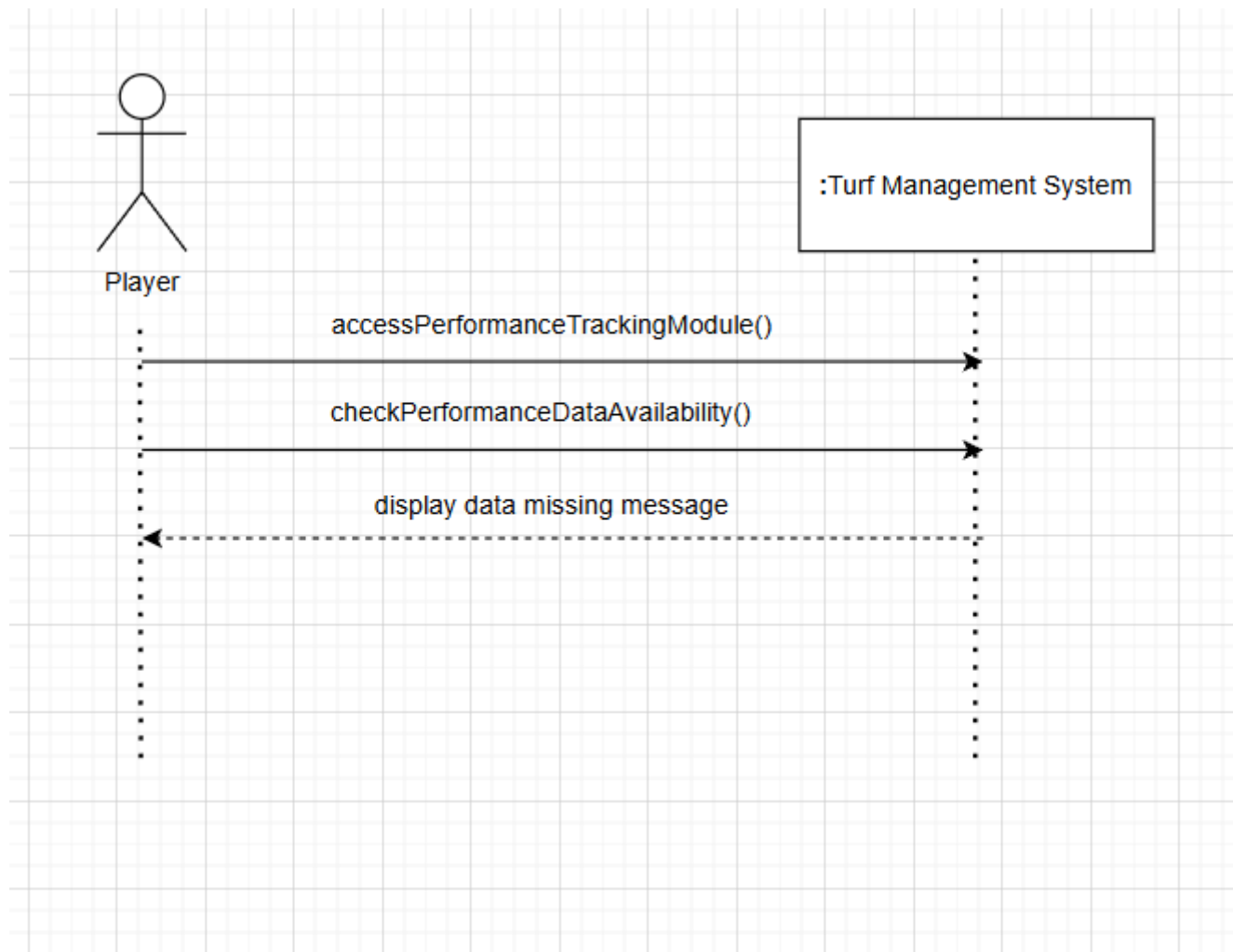


7) Player performance tracking and rankings

Main Success Scenario:

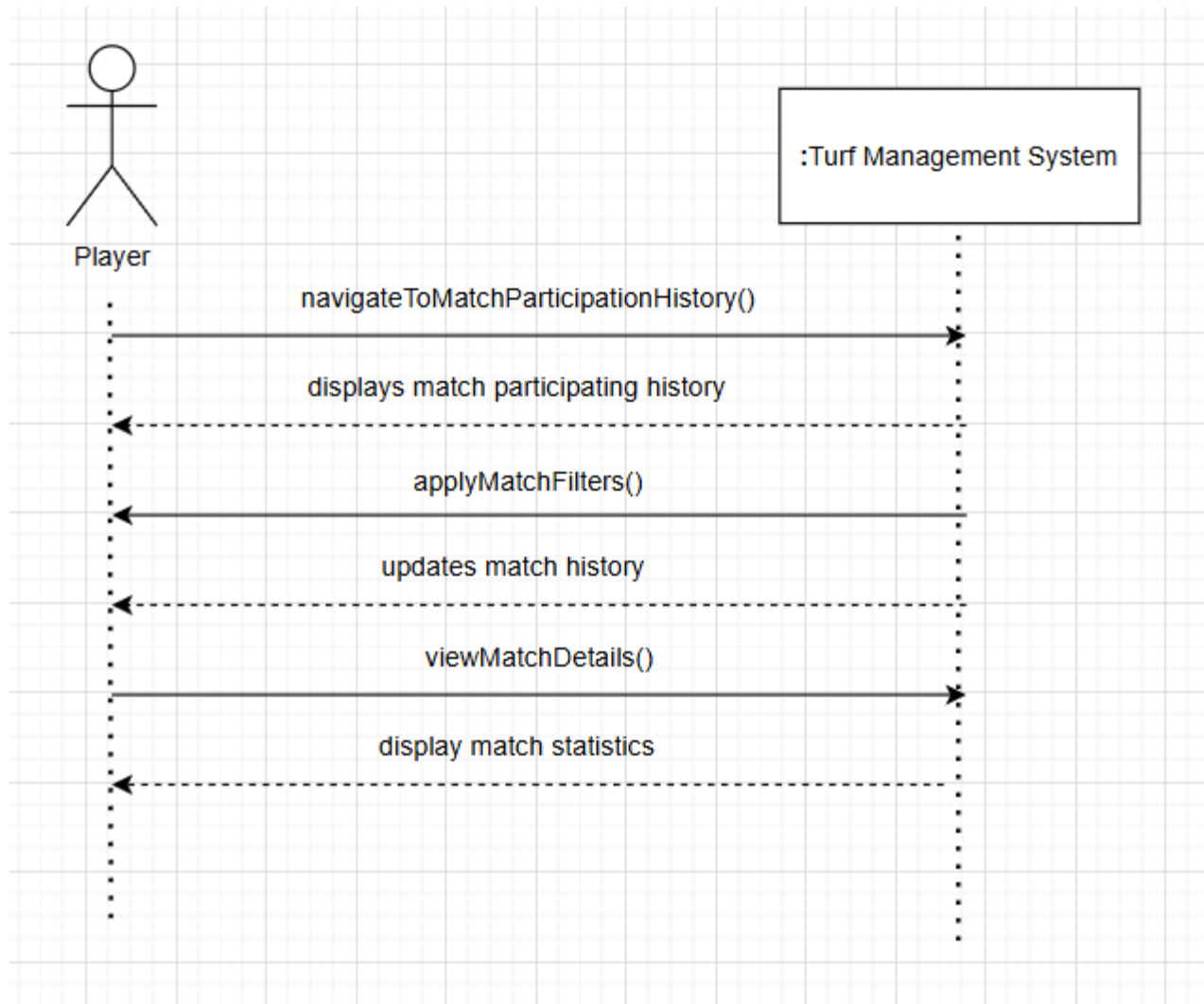


Alternate Scenario:

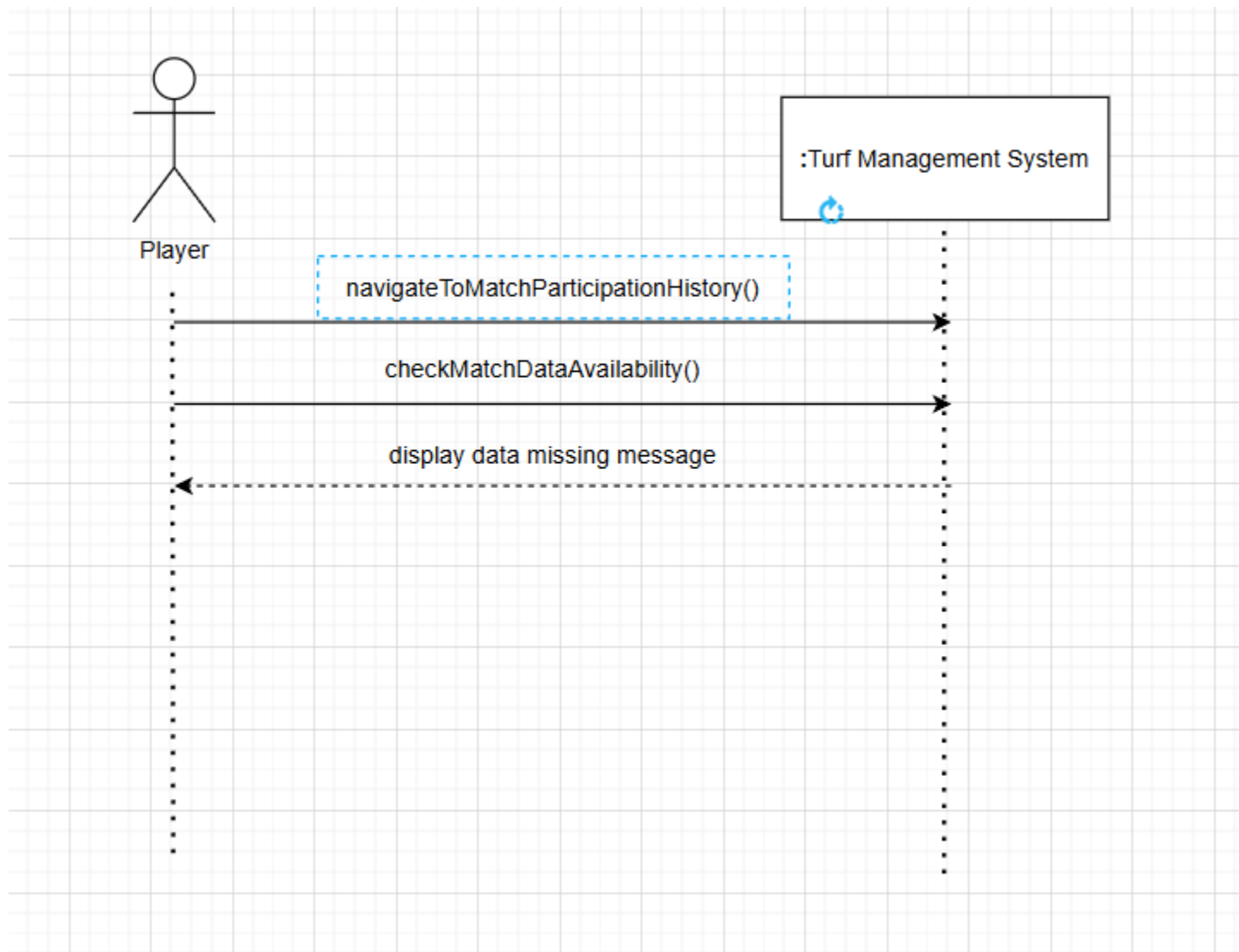


8) Match Performing history

Main Success Scenario:

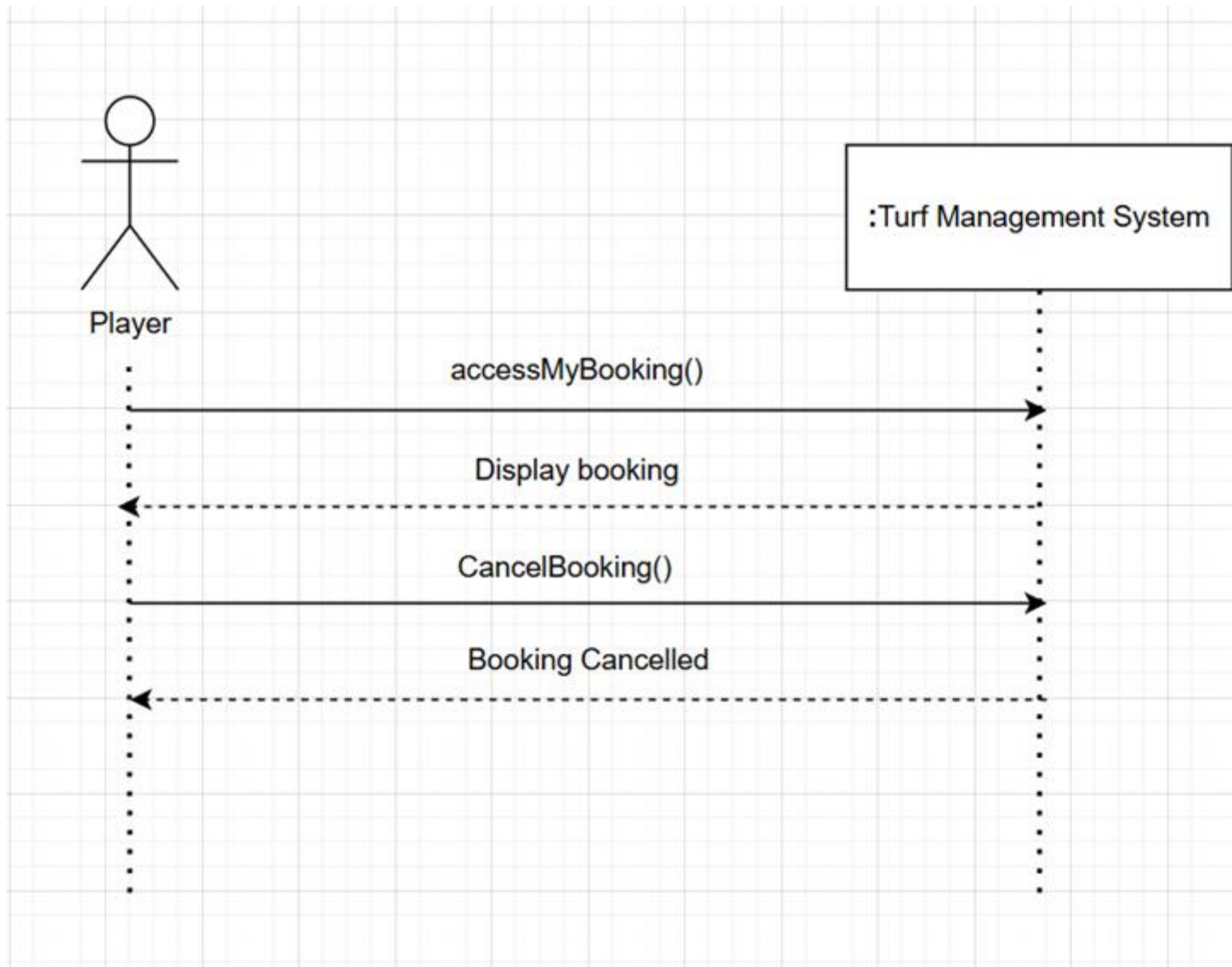


Alternate Scenario:

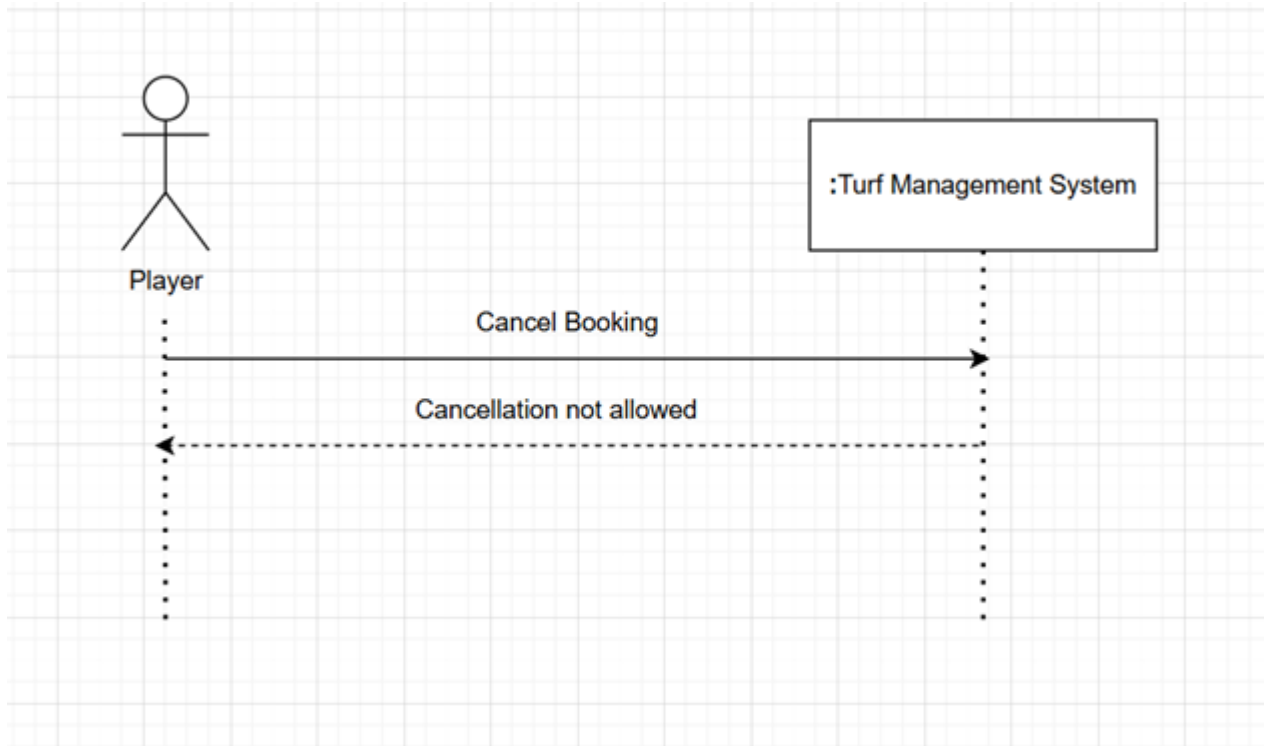


9) Cancelling turf booking

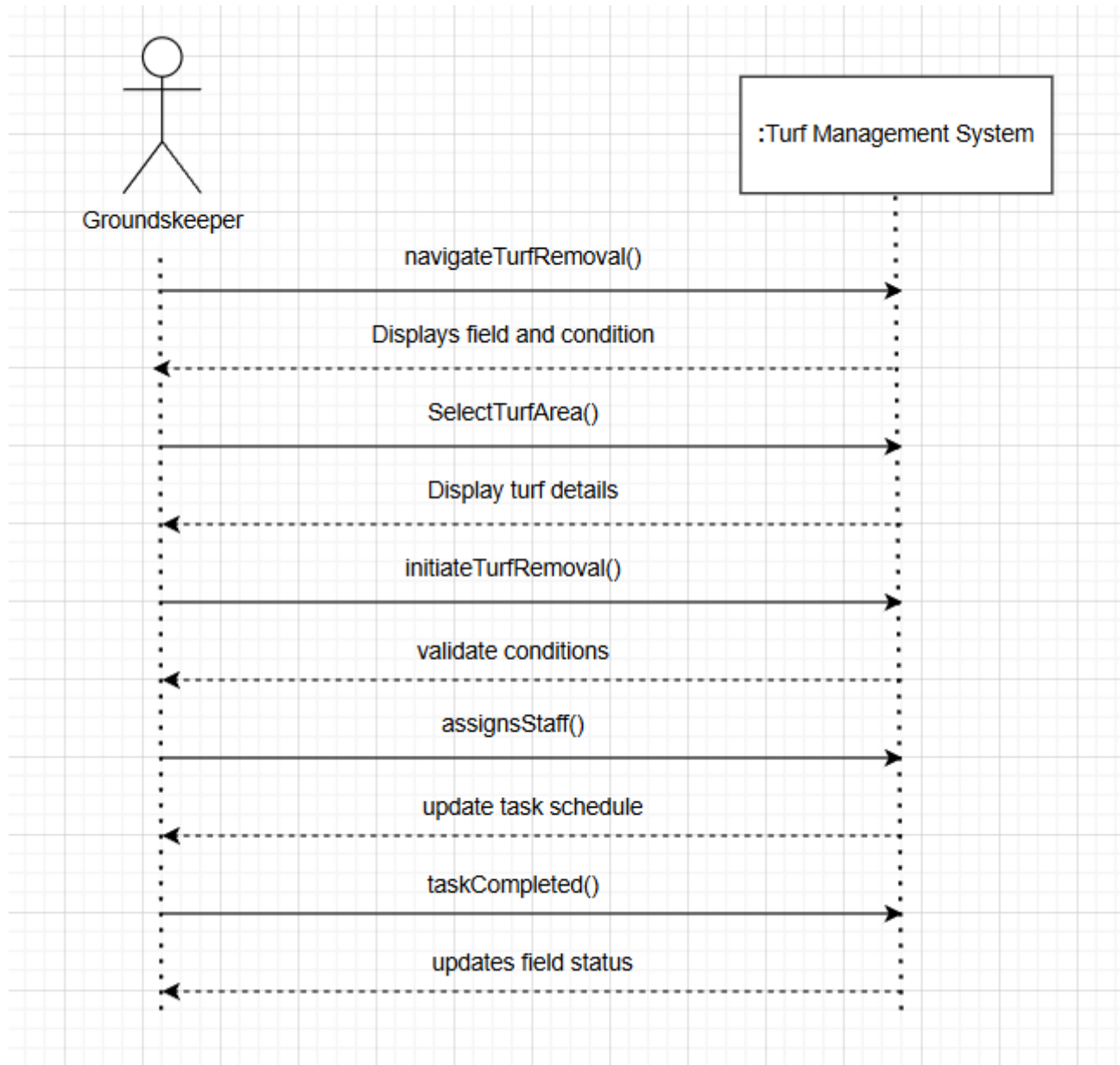
Main Success Scenario:



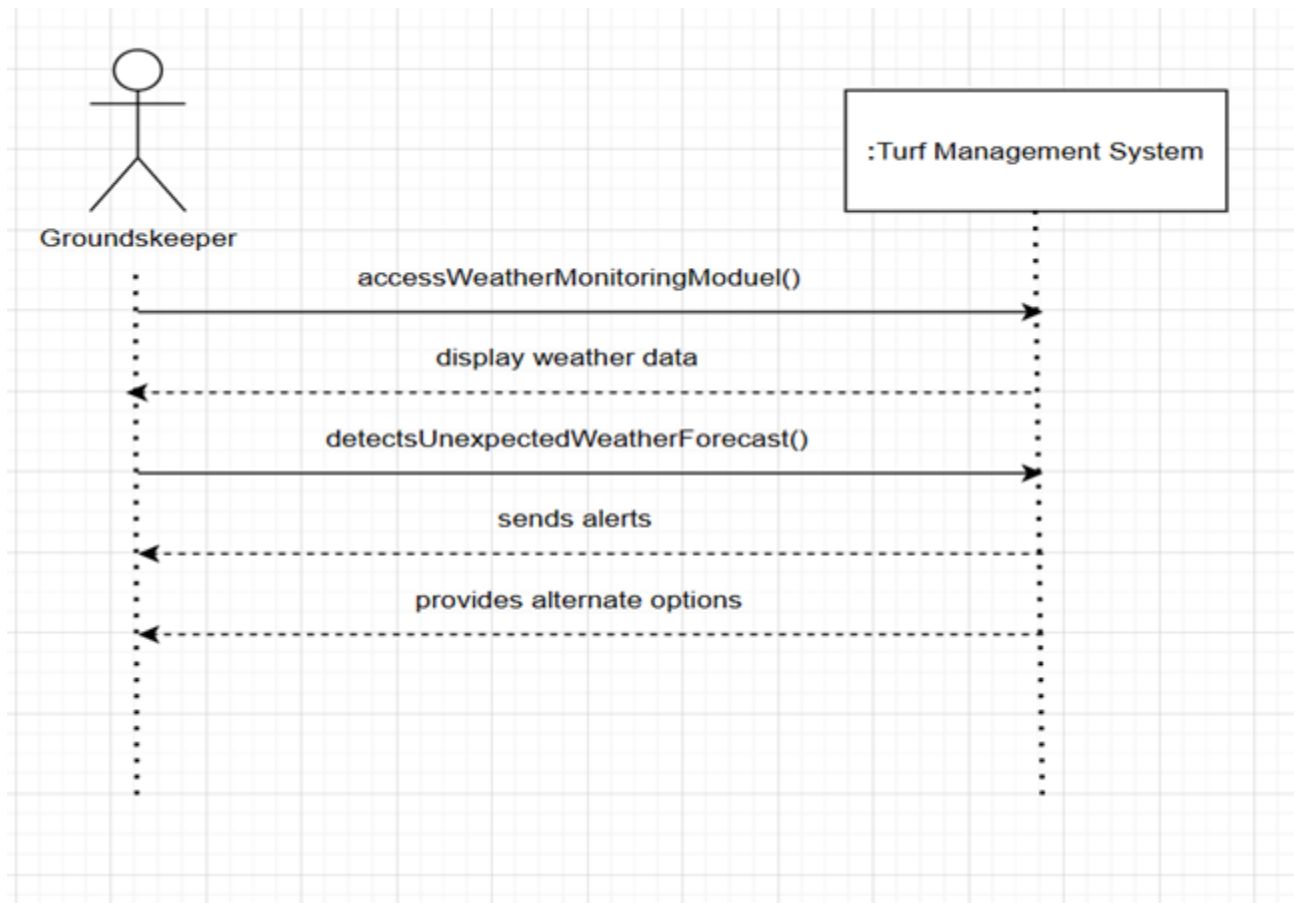
Alternate Scenario:



10) **Turf Removal**
Main Success Scenario:



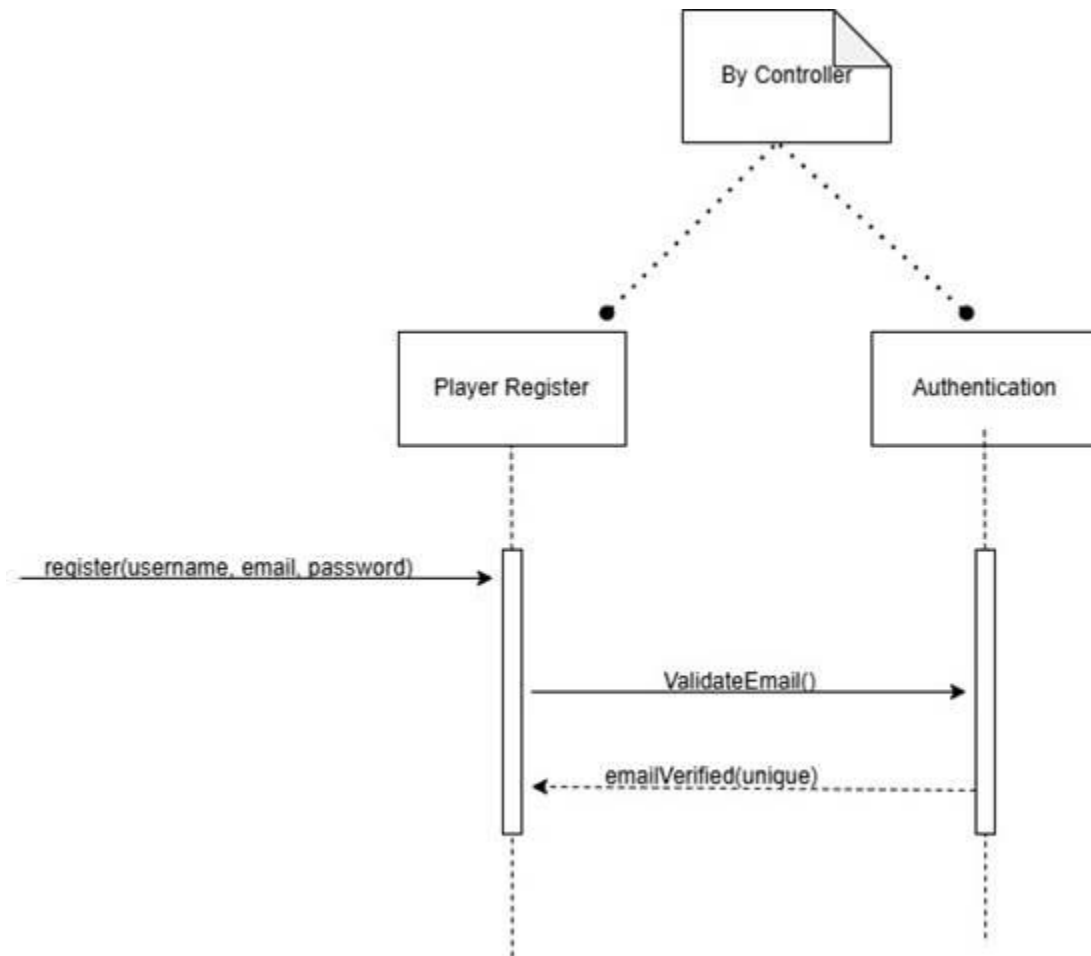
Alternate Scenario:



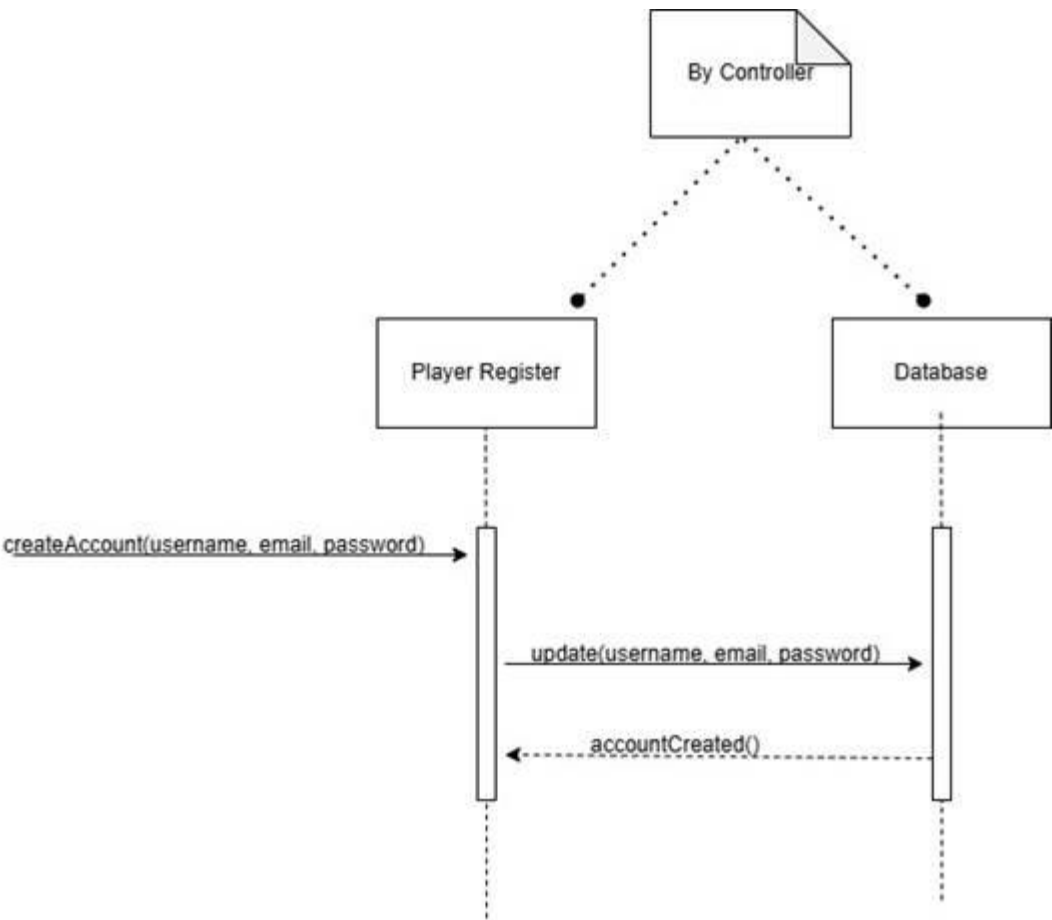
6. Sequence Diagram

1) Registration/Login

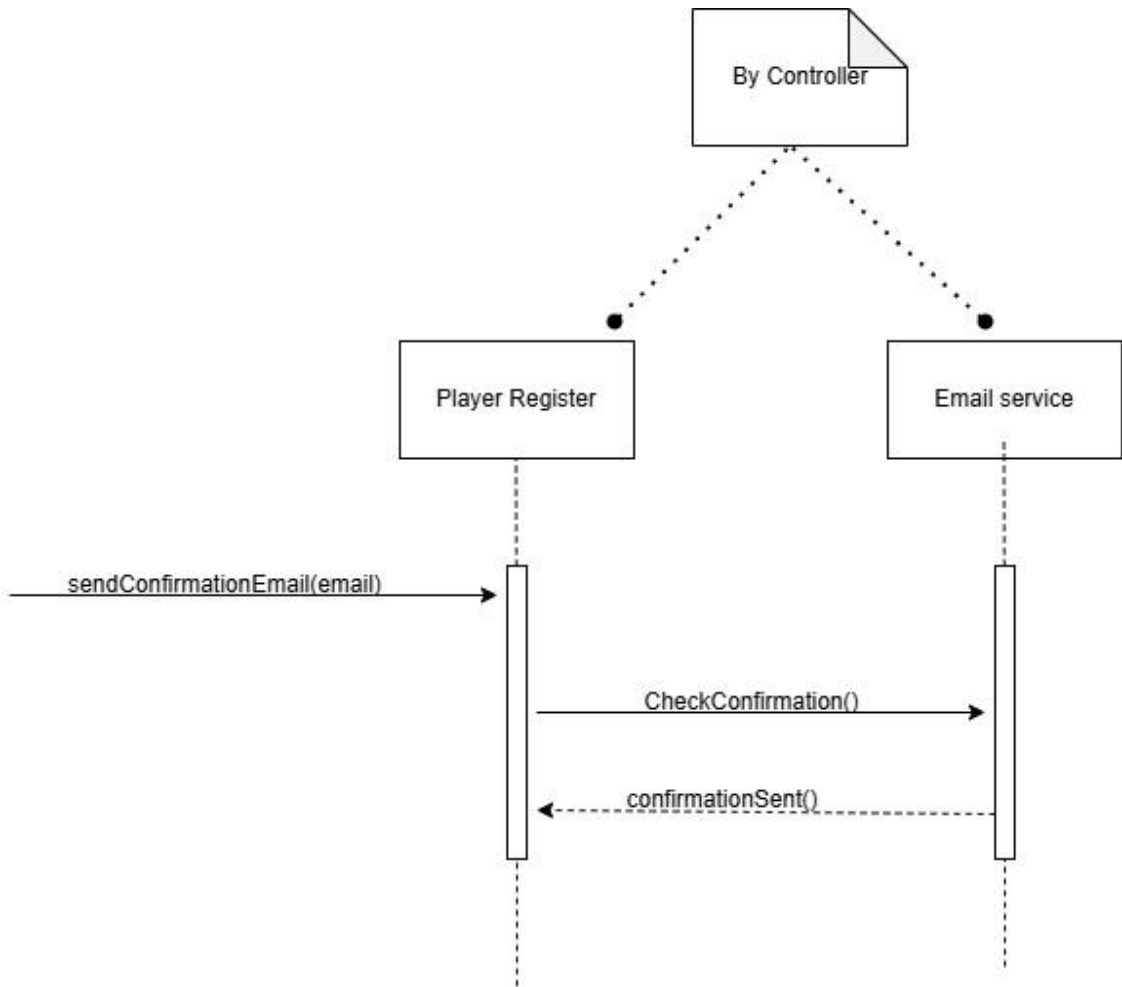
Register:



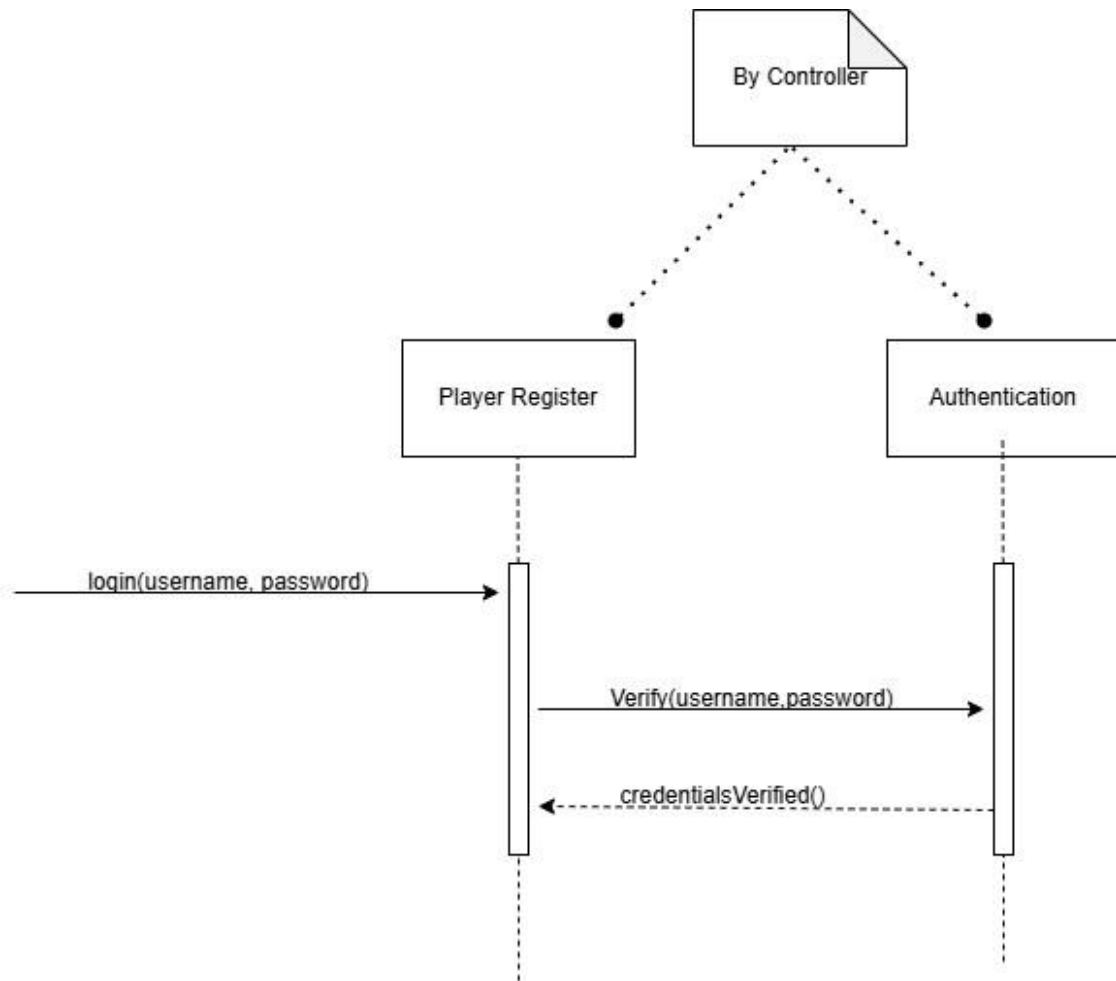
Create Account:



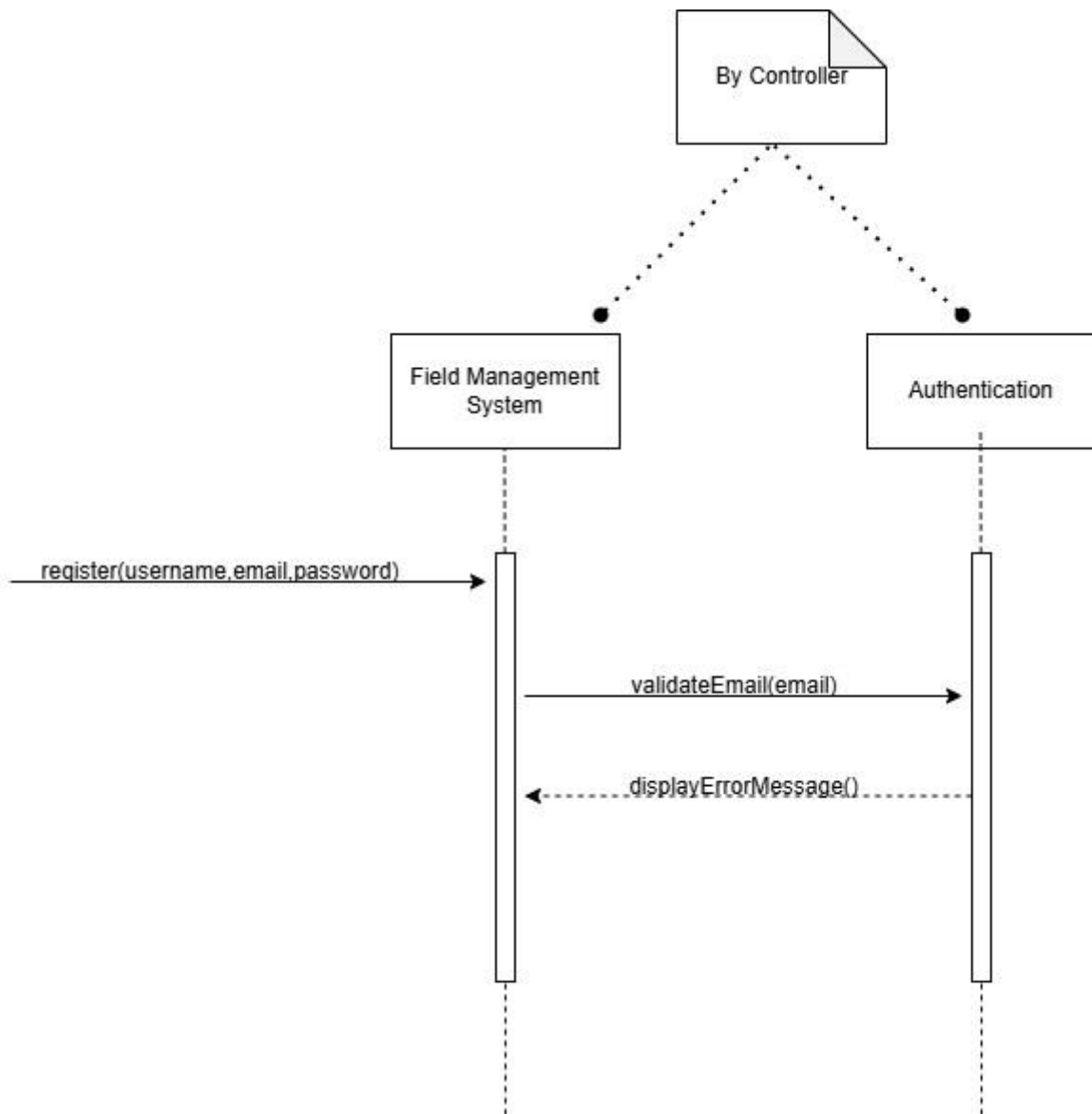
Send Confirmation Email:



Login:

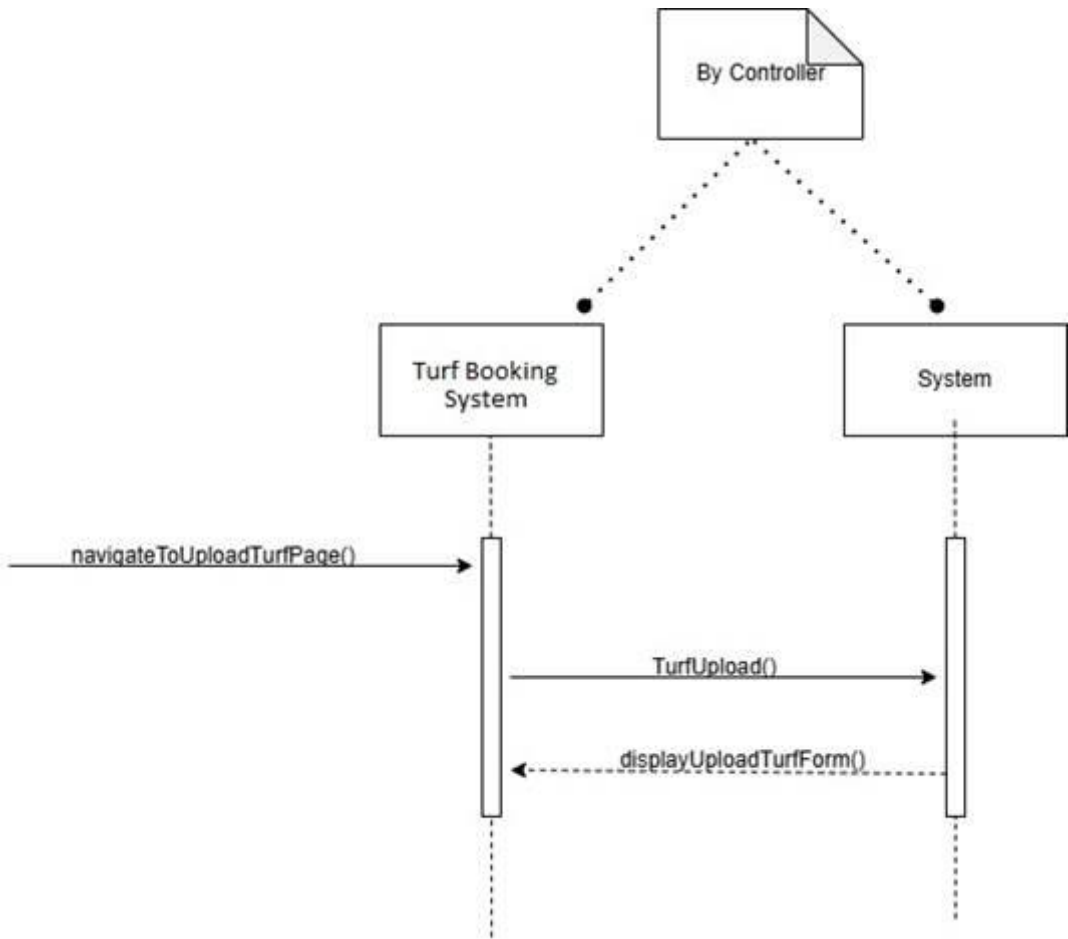


Alternate Scenario:

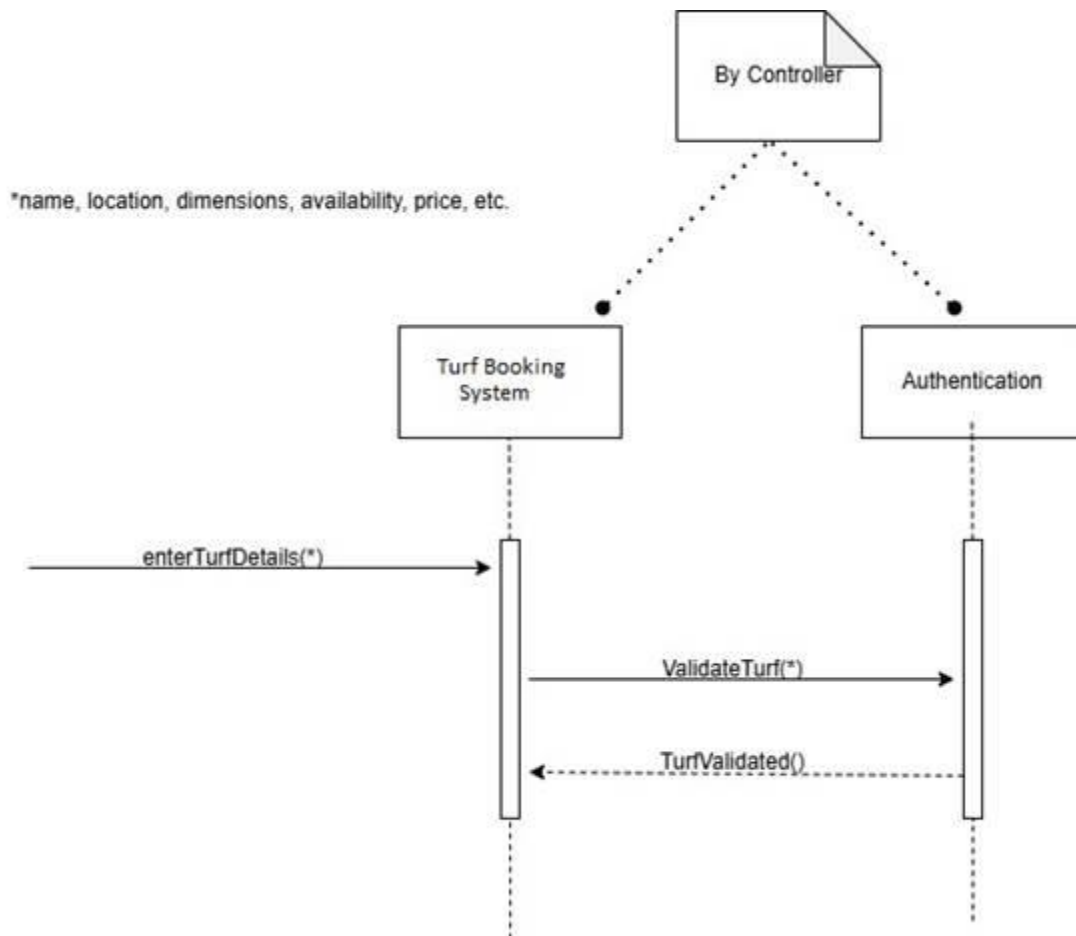


2) Upload Turf Information:

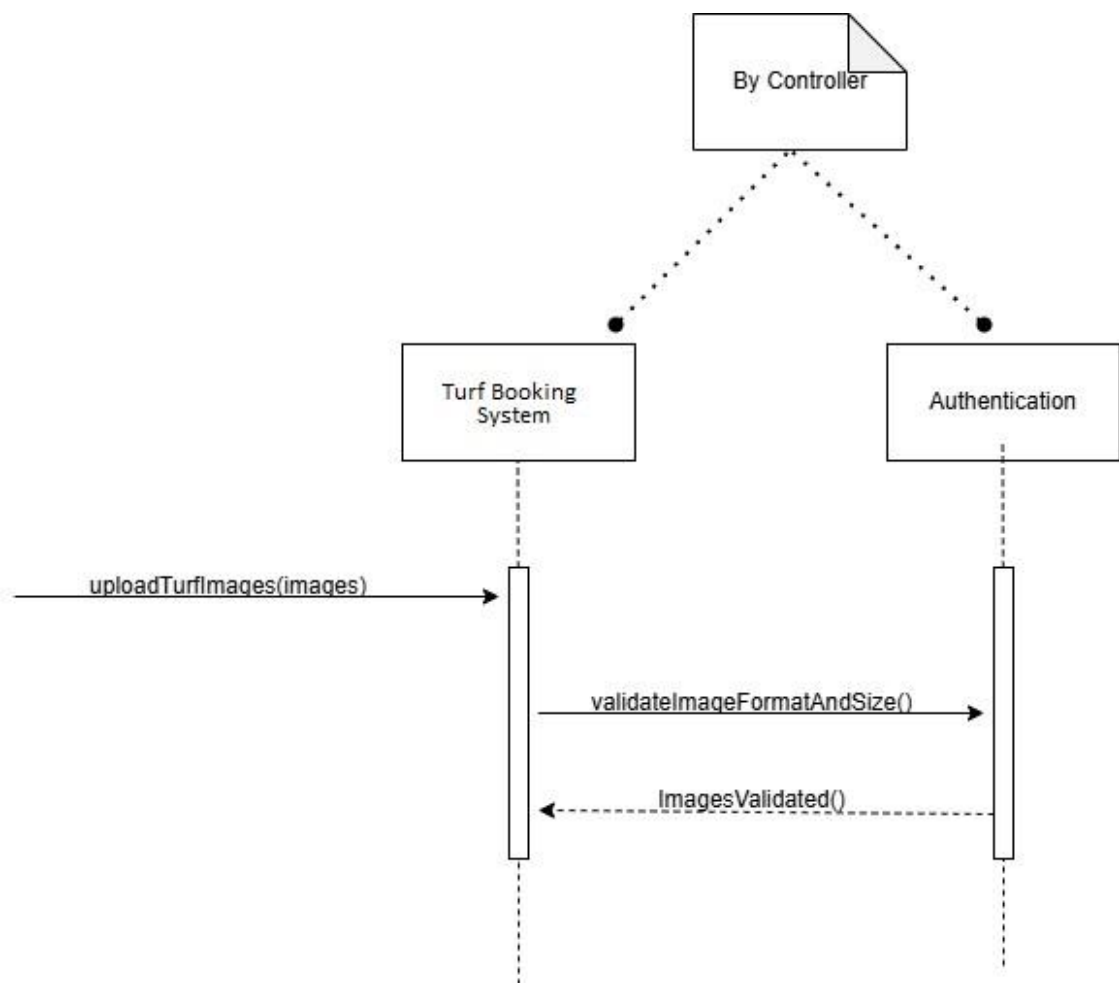
Navigate to upload turf:



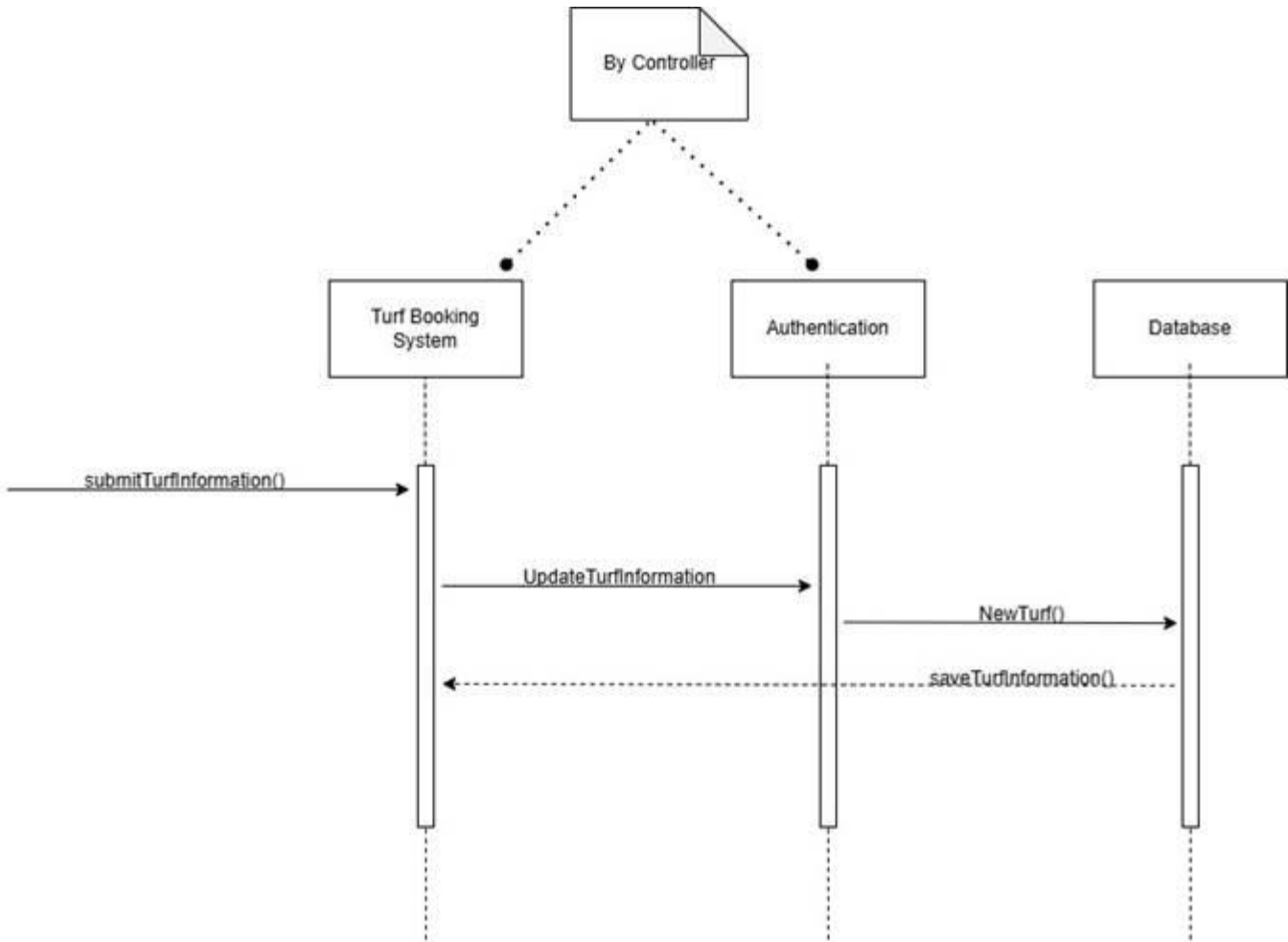
Enter turf details:



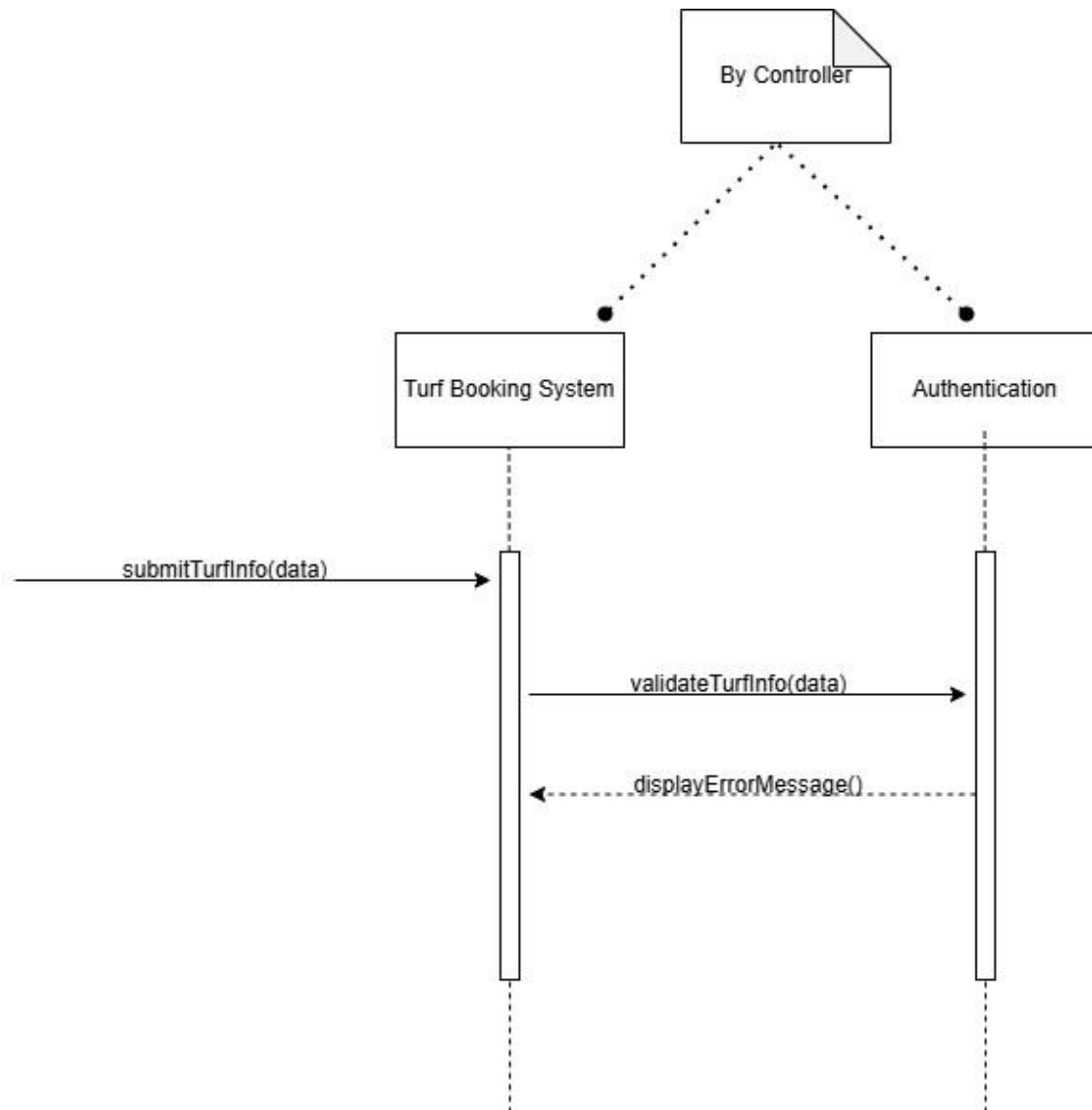
Upload turf images:



Submit Turf Information:

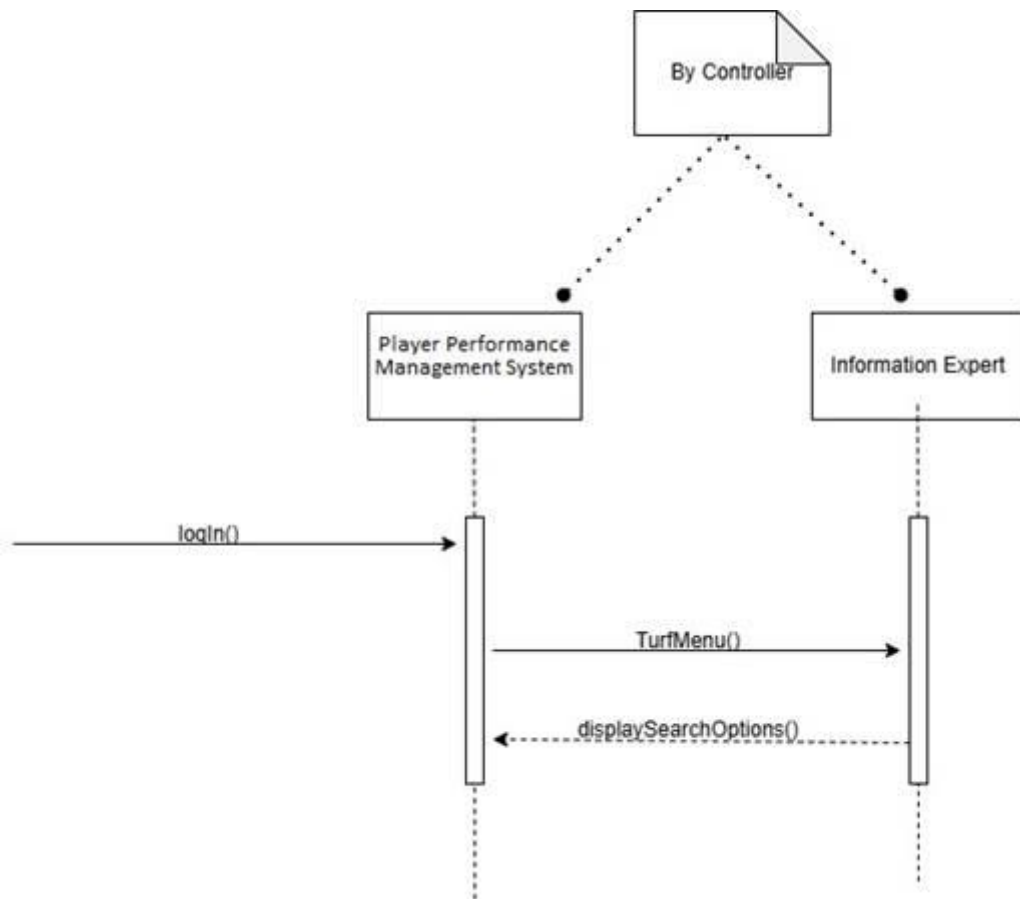


Alternate Scenario:

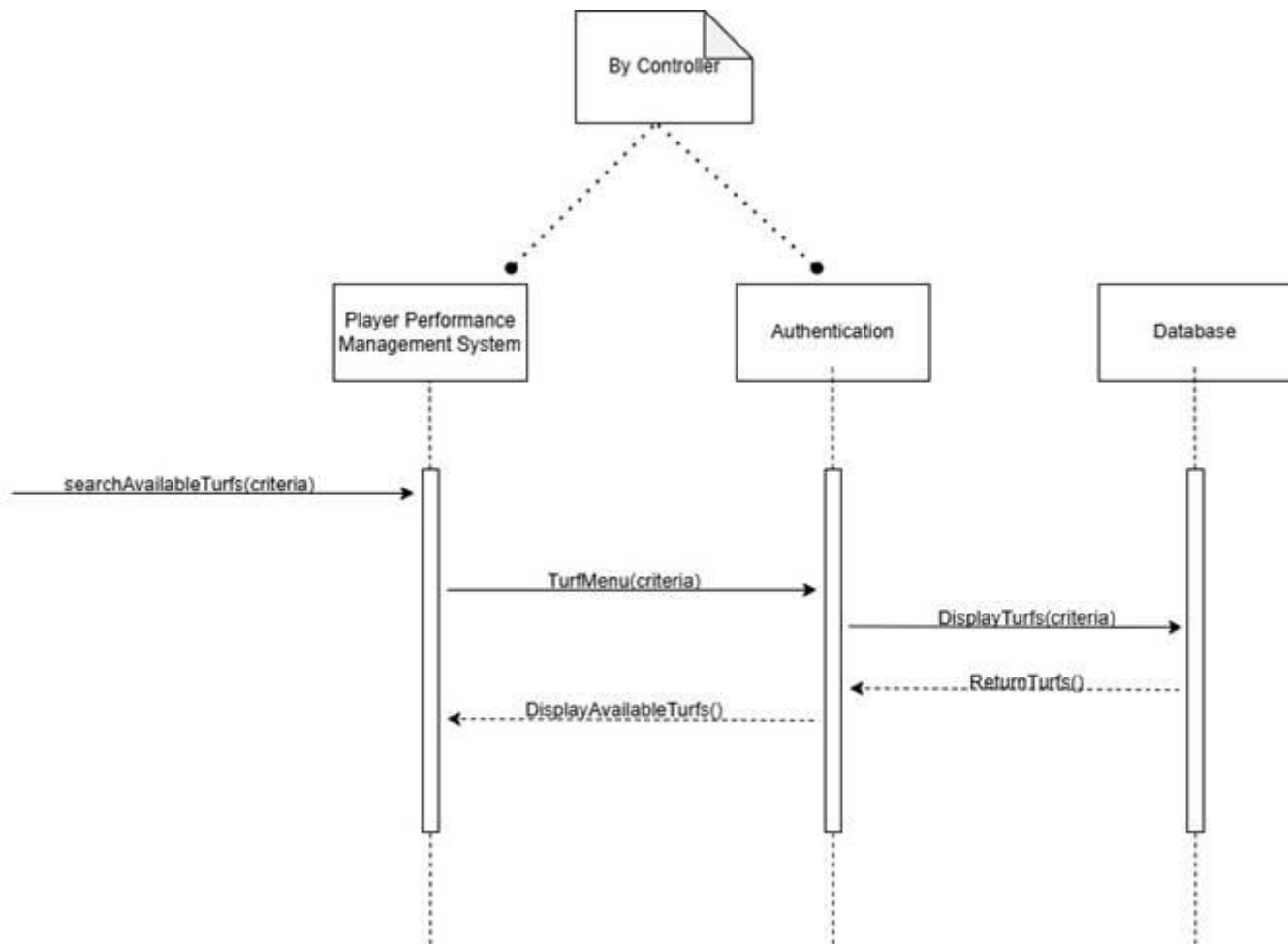


3) Finding and booking a turf:

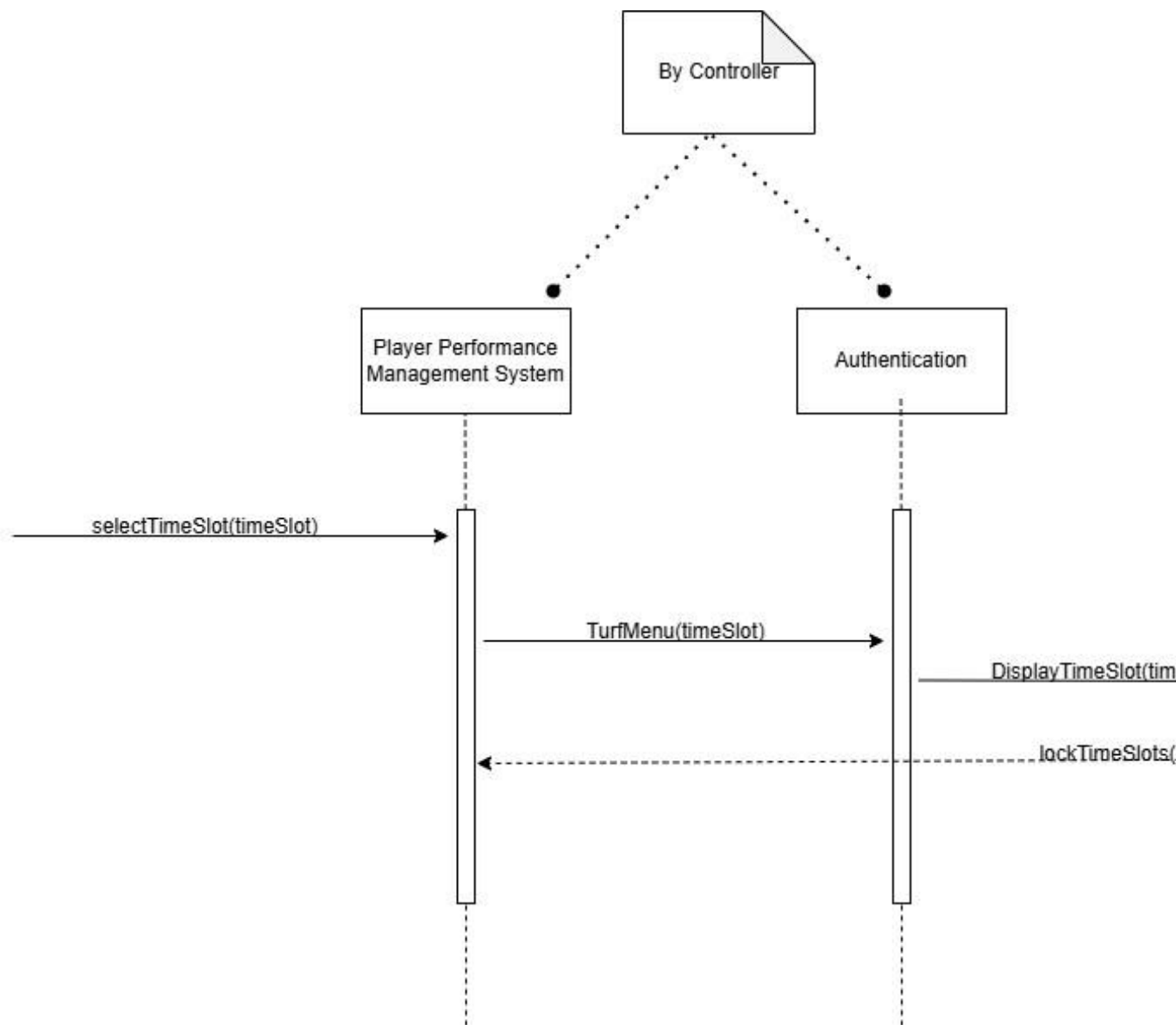
Login:



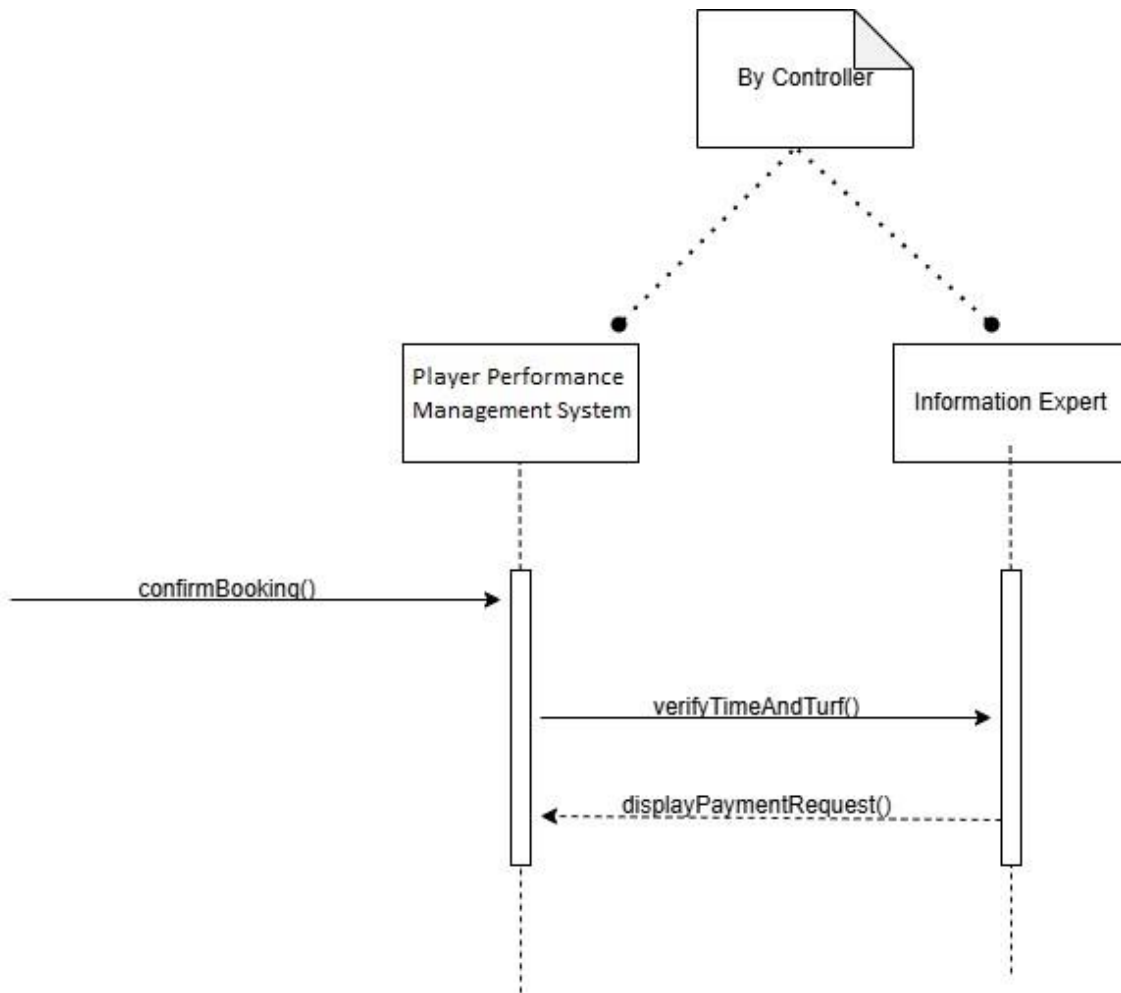
Search Available turfs:



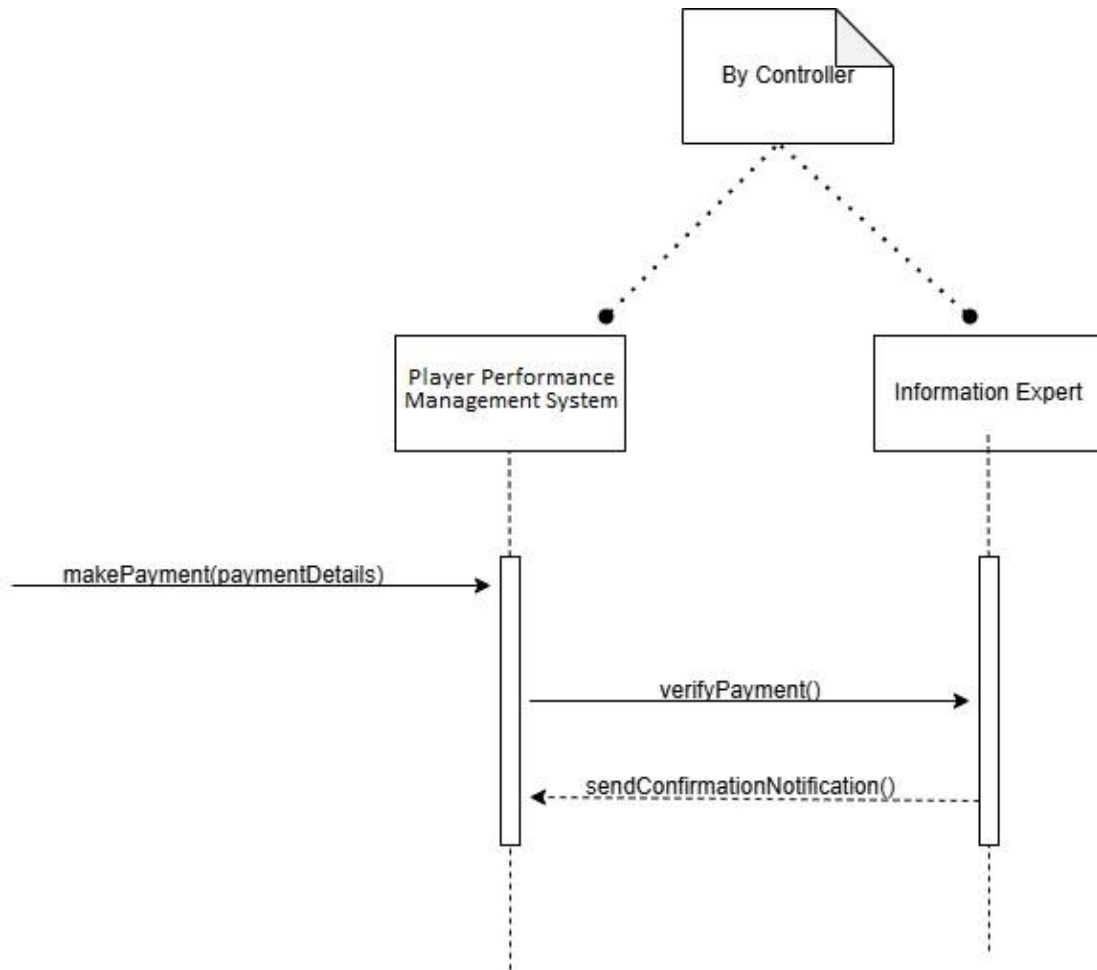
Select Time slots:



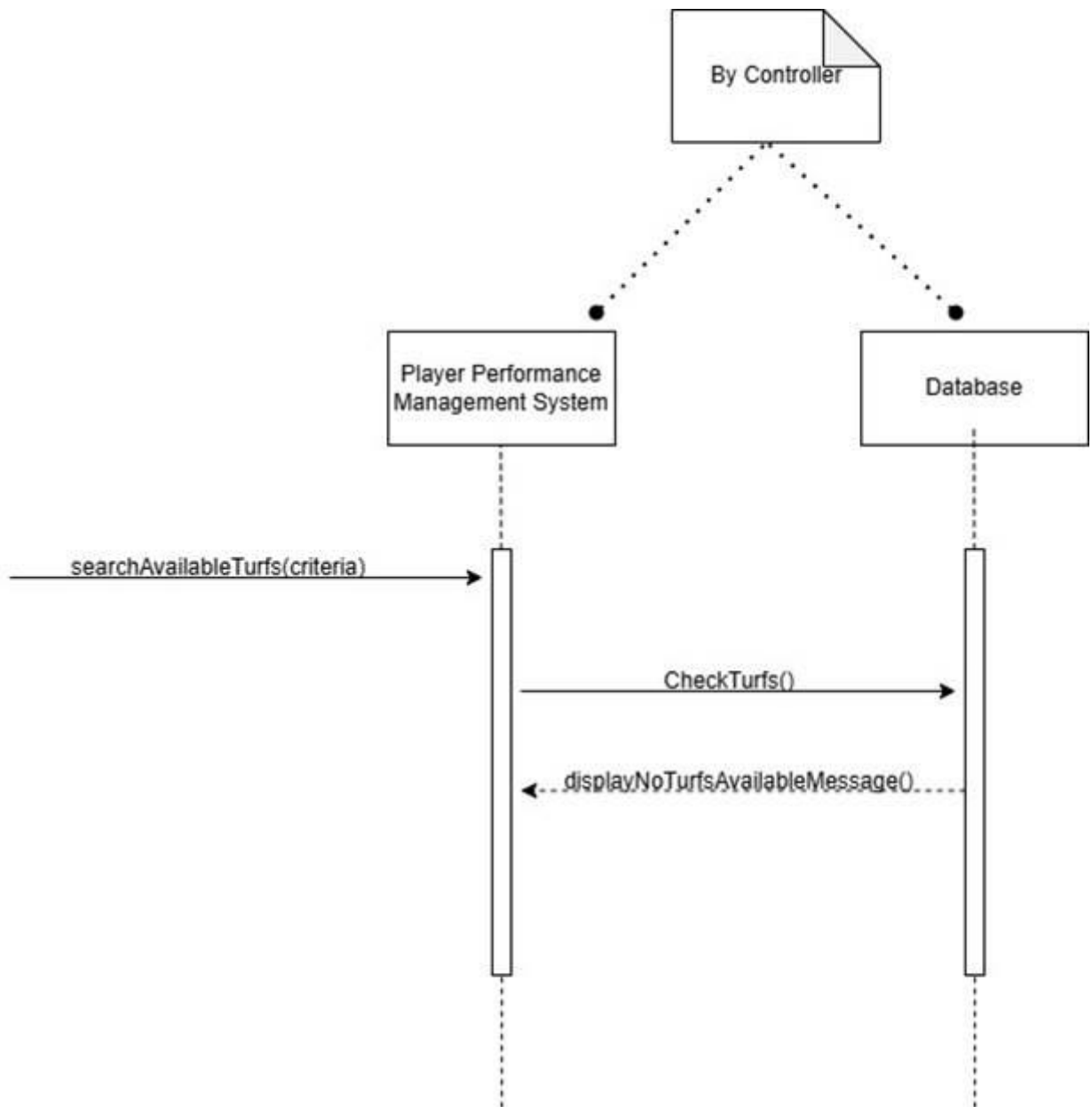
Confirm Booking:



Make Payment:

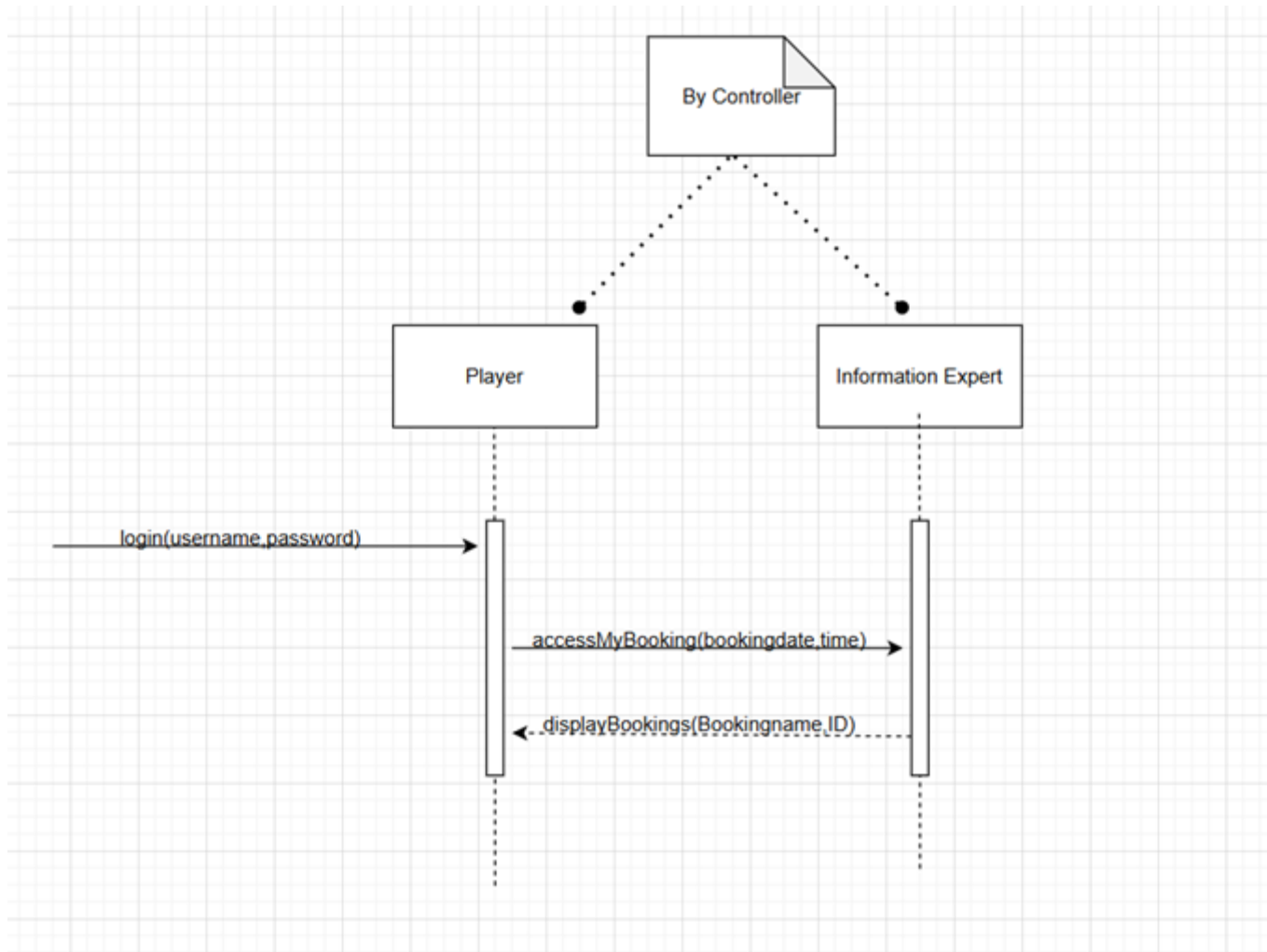


Alternate Scenario:
Search Available turfs:

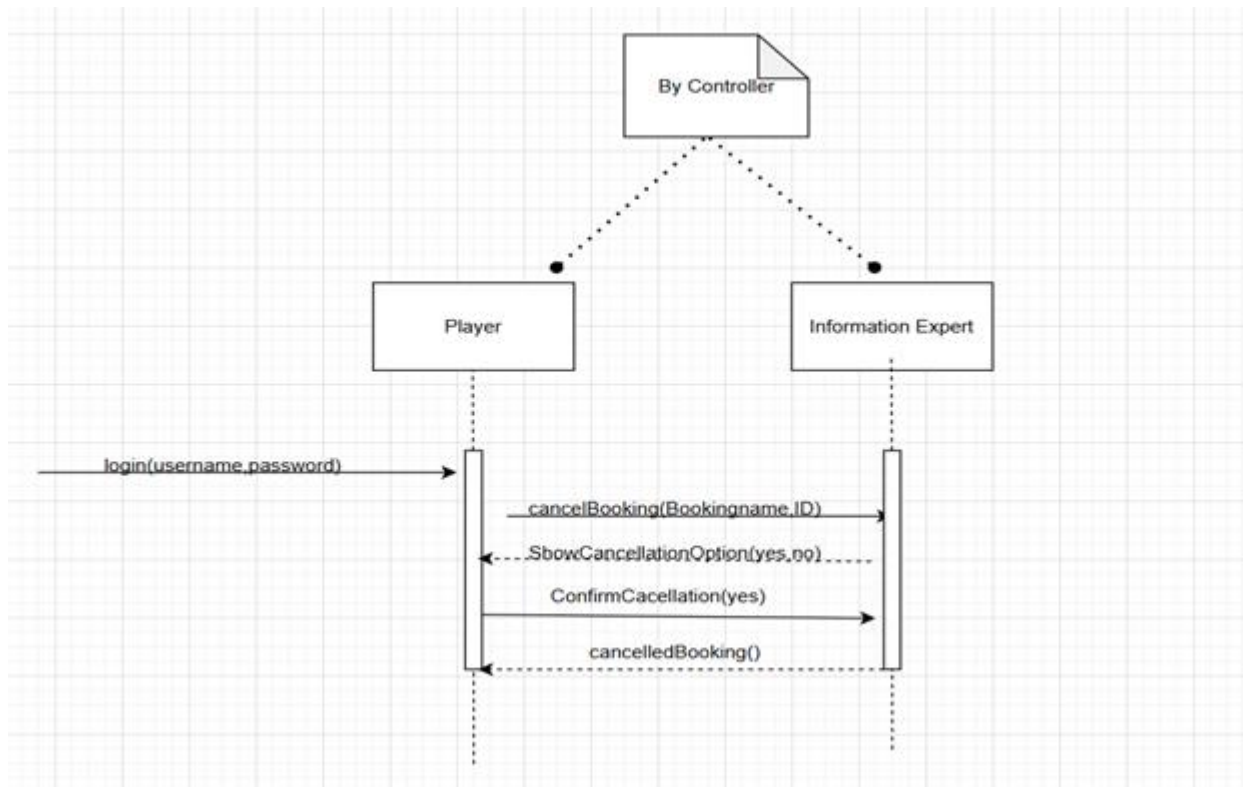


4) Cancelling Booking:

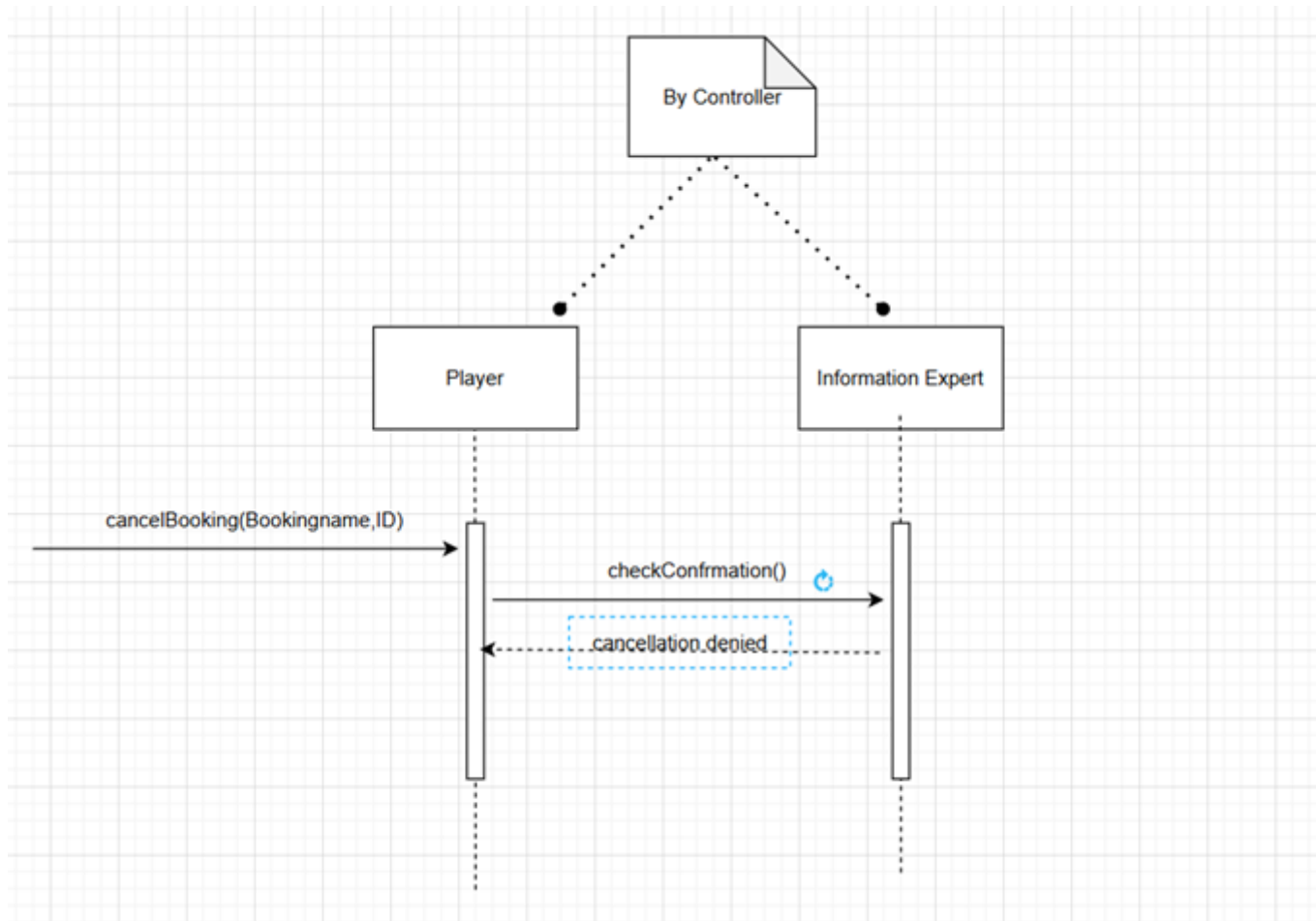
Access Booking:



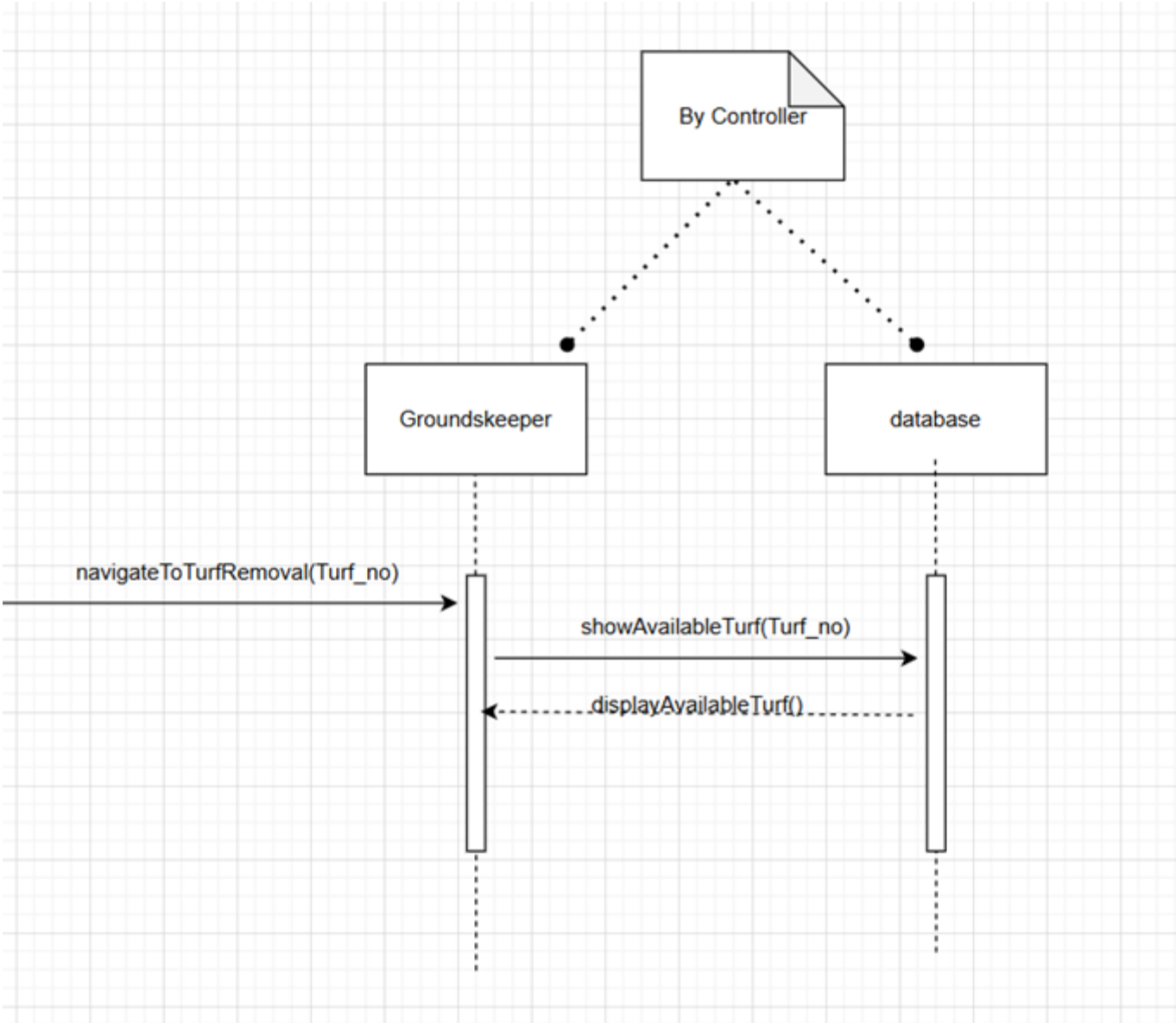
Cancelling booking:



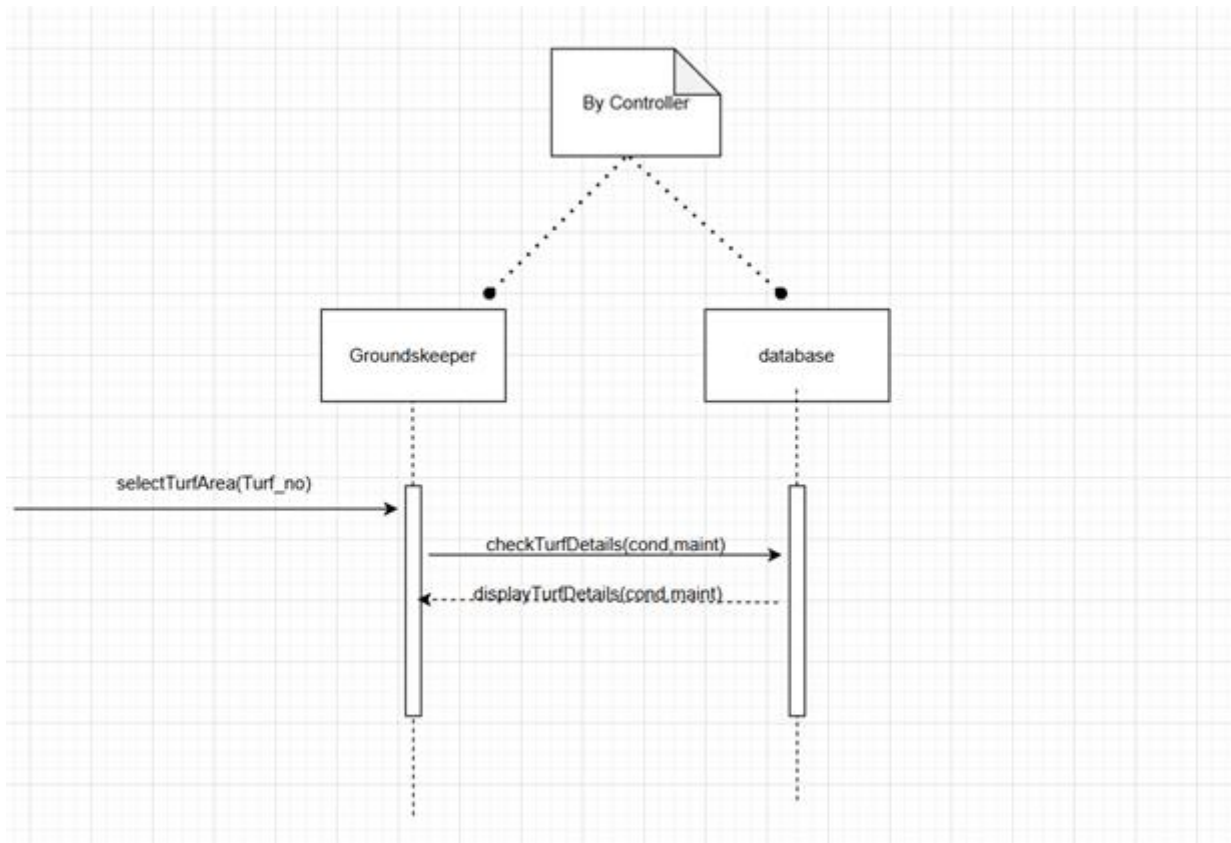
Alternate Scenario:



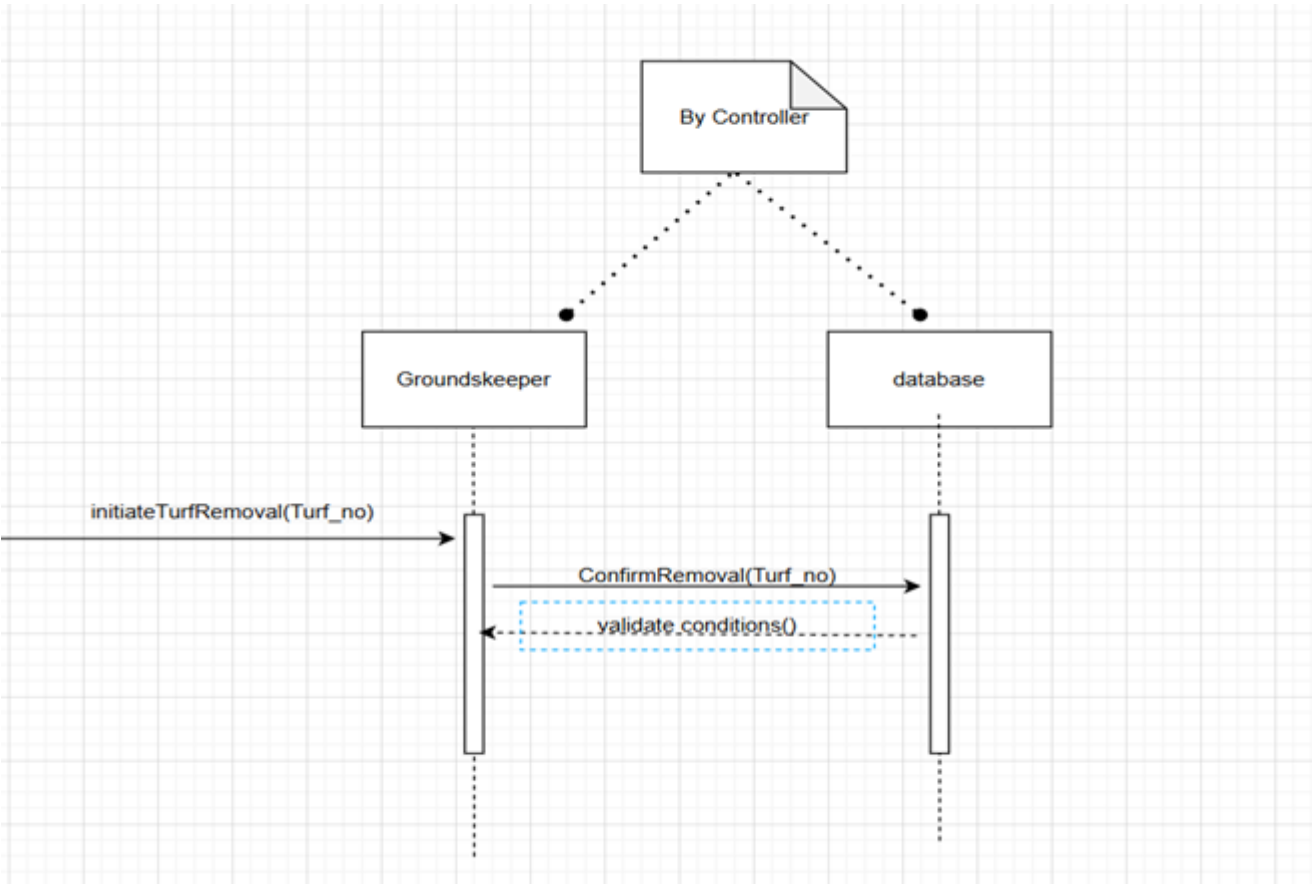
5) Turf Removal:
Navigate to turf removal:



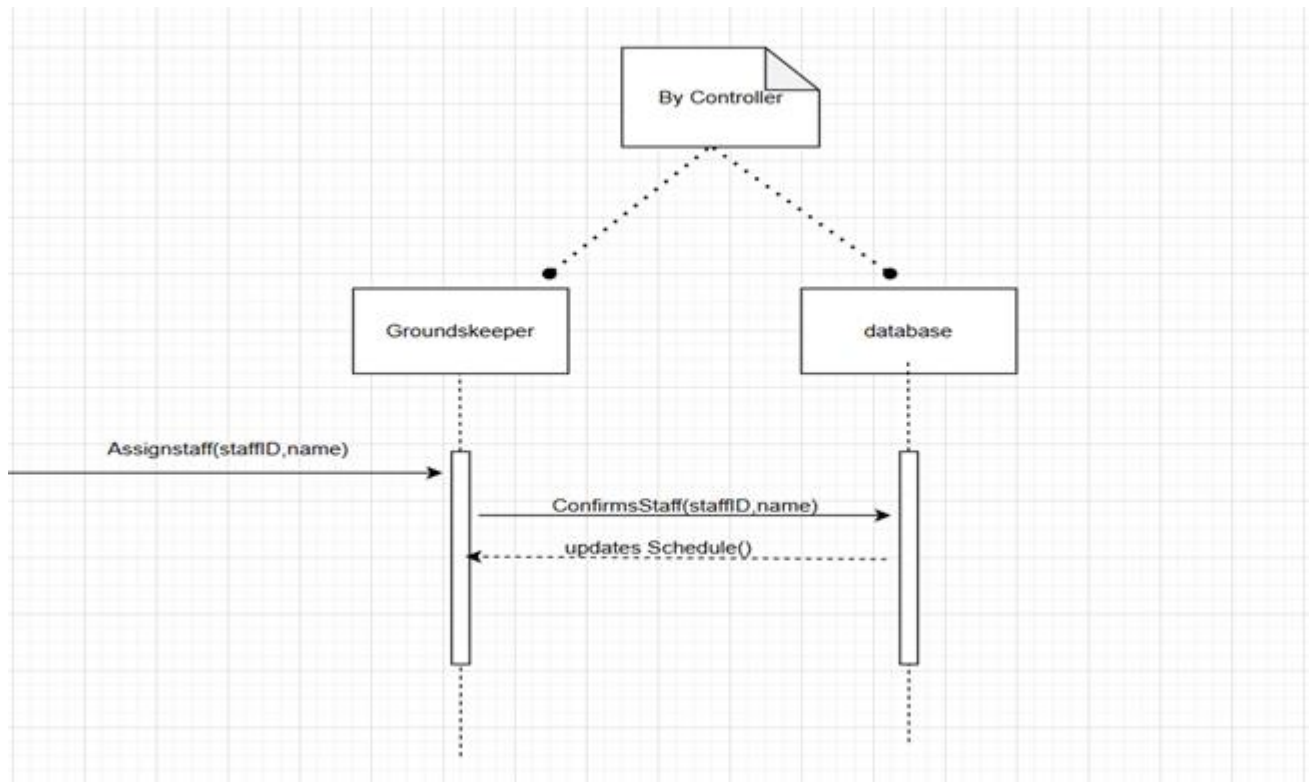
Select turf area:



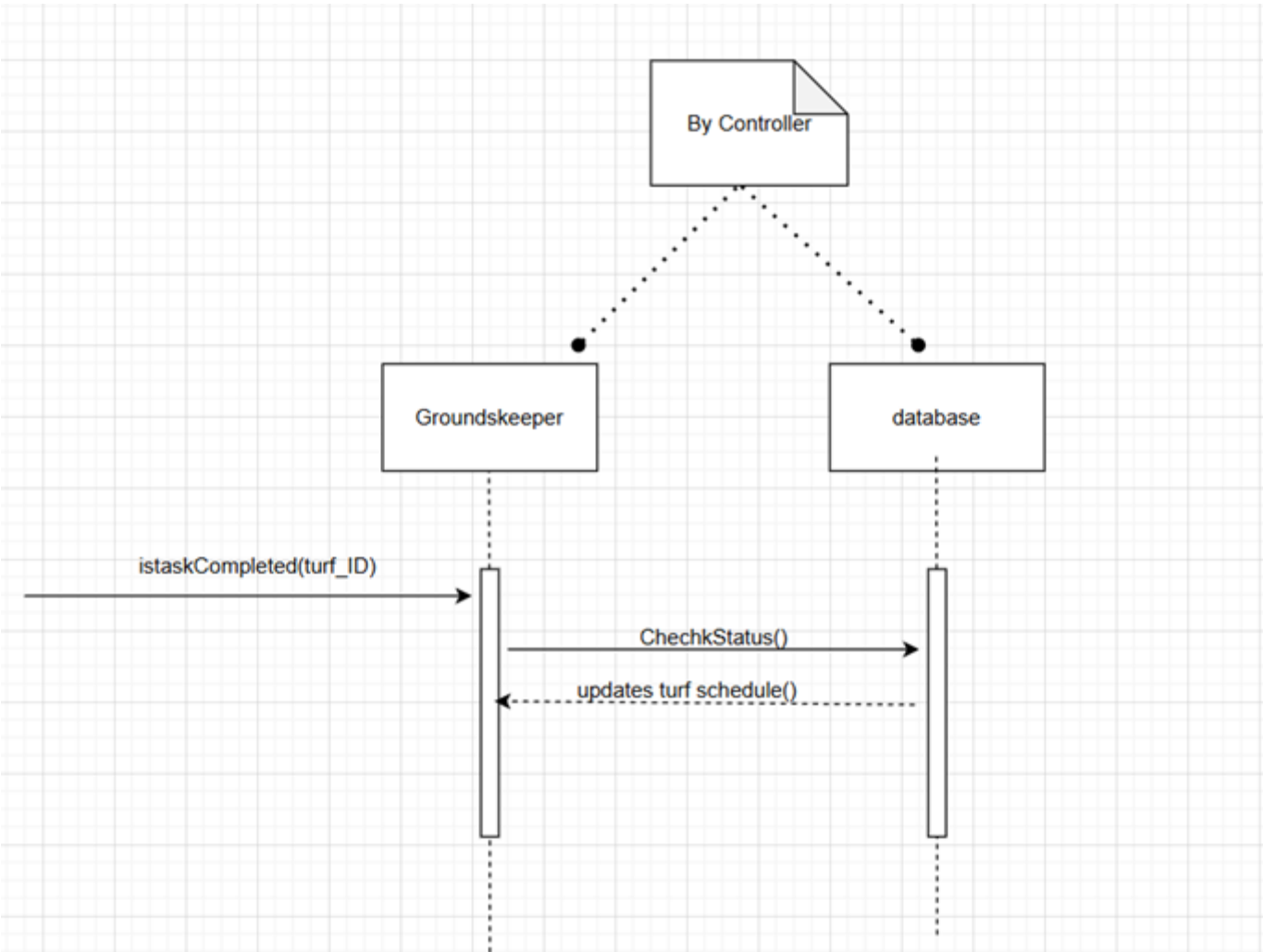
Initiate turf removal:



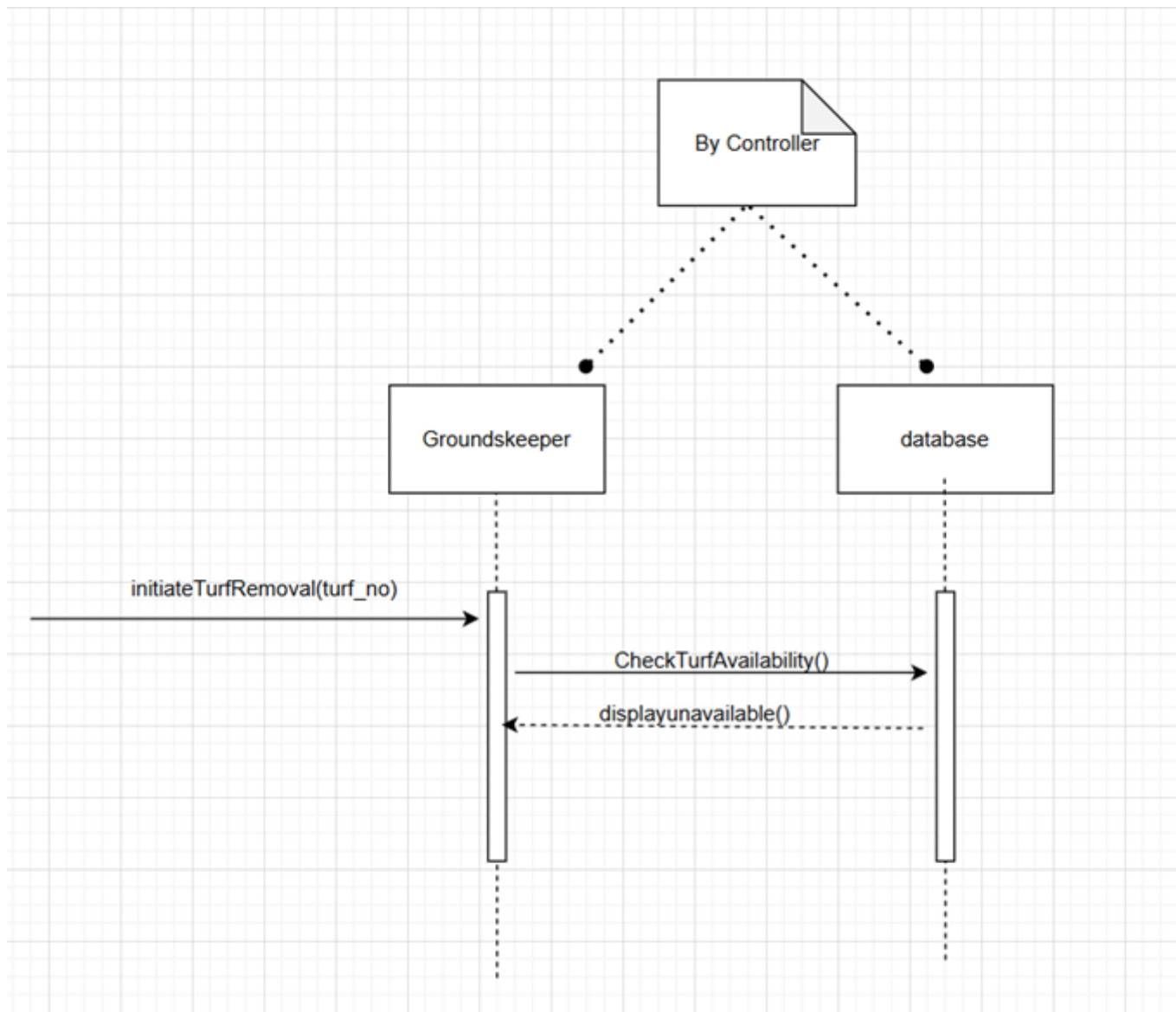
Assign staff:



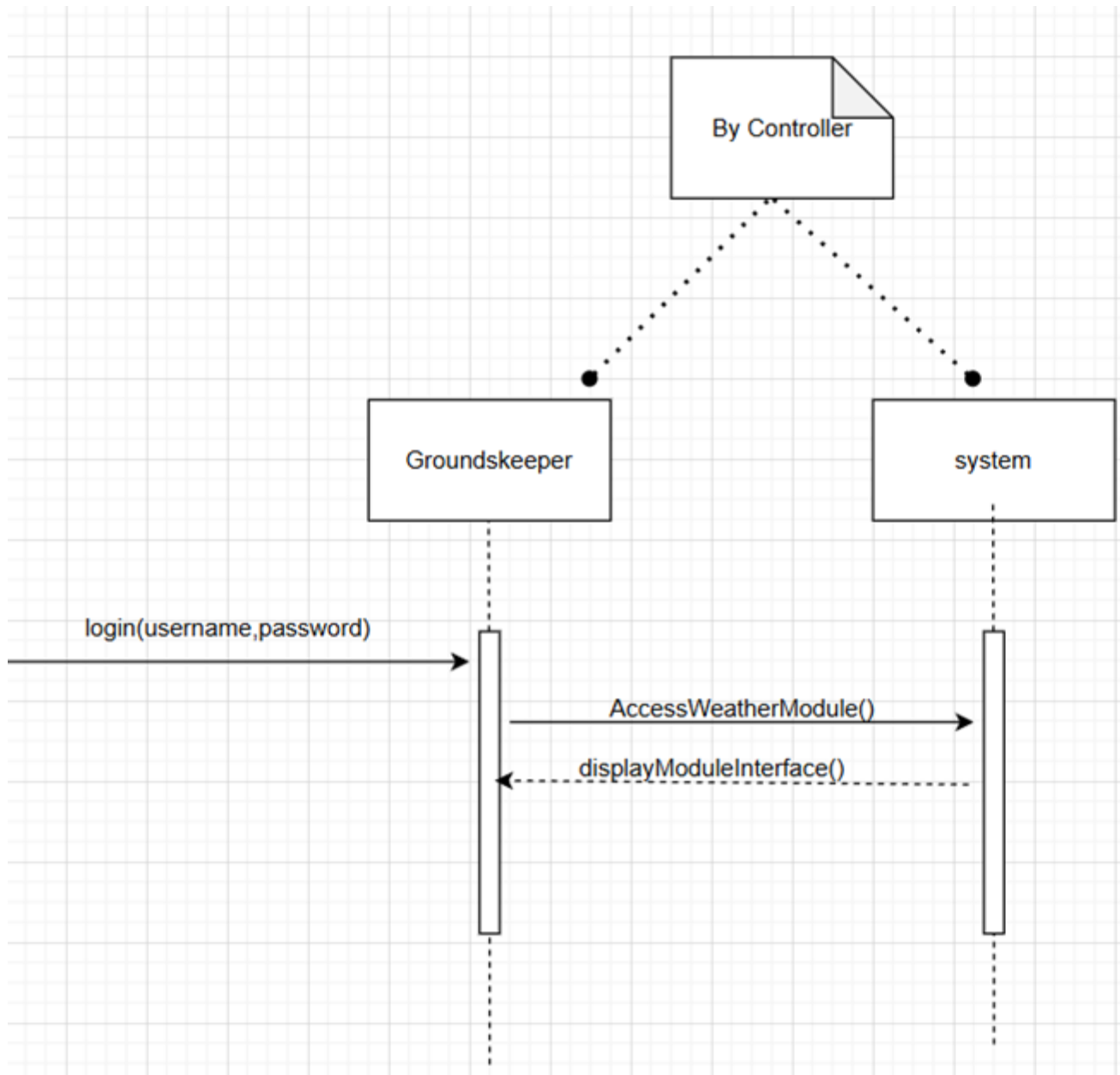
Task Completed:



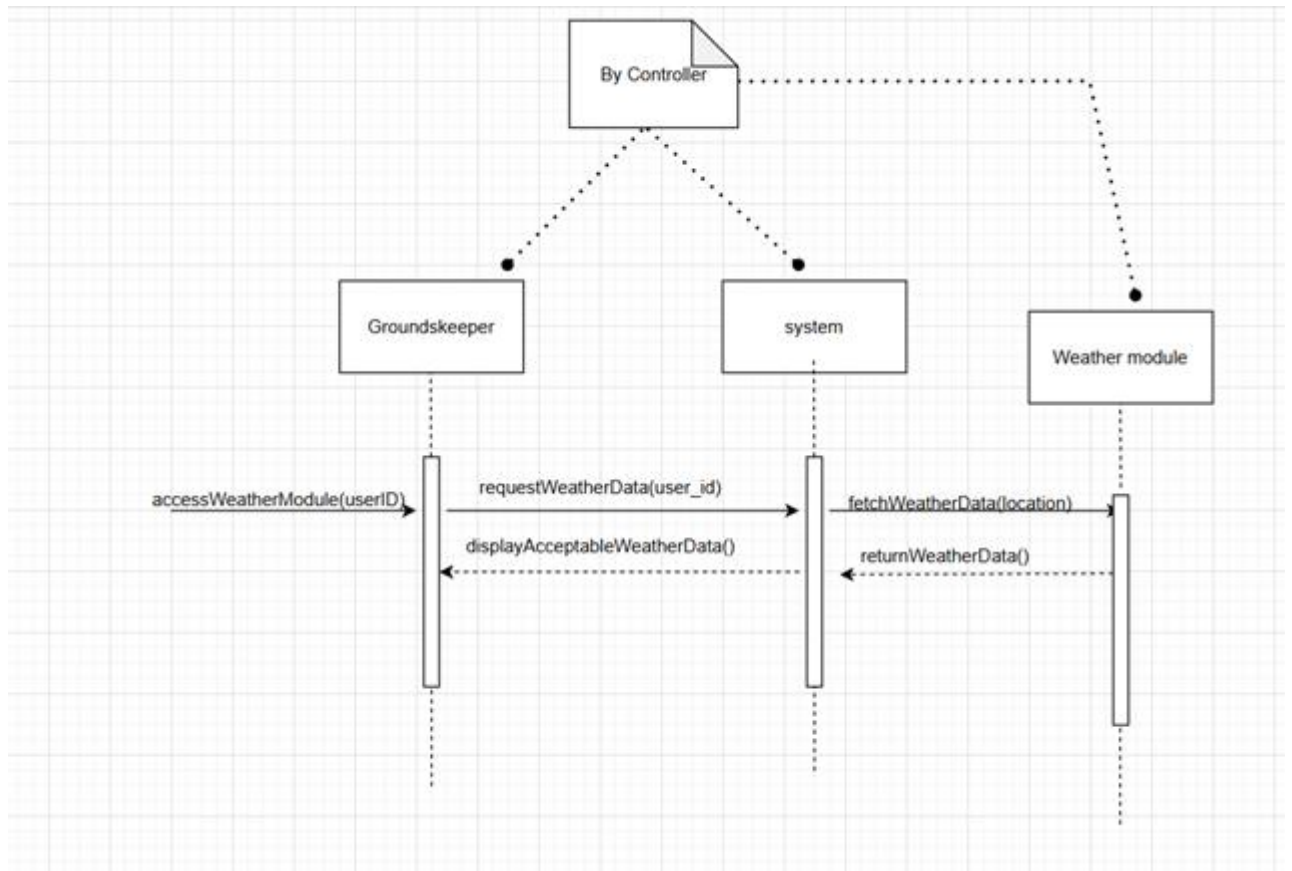
Alternate Scenario:



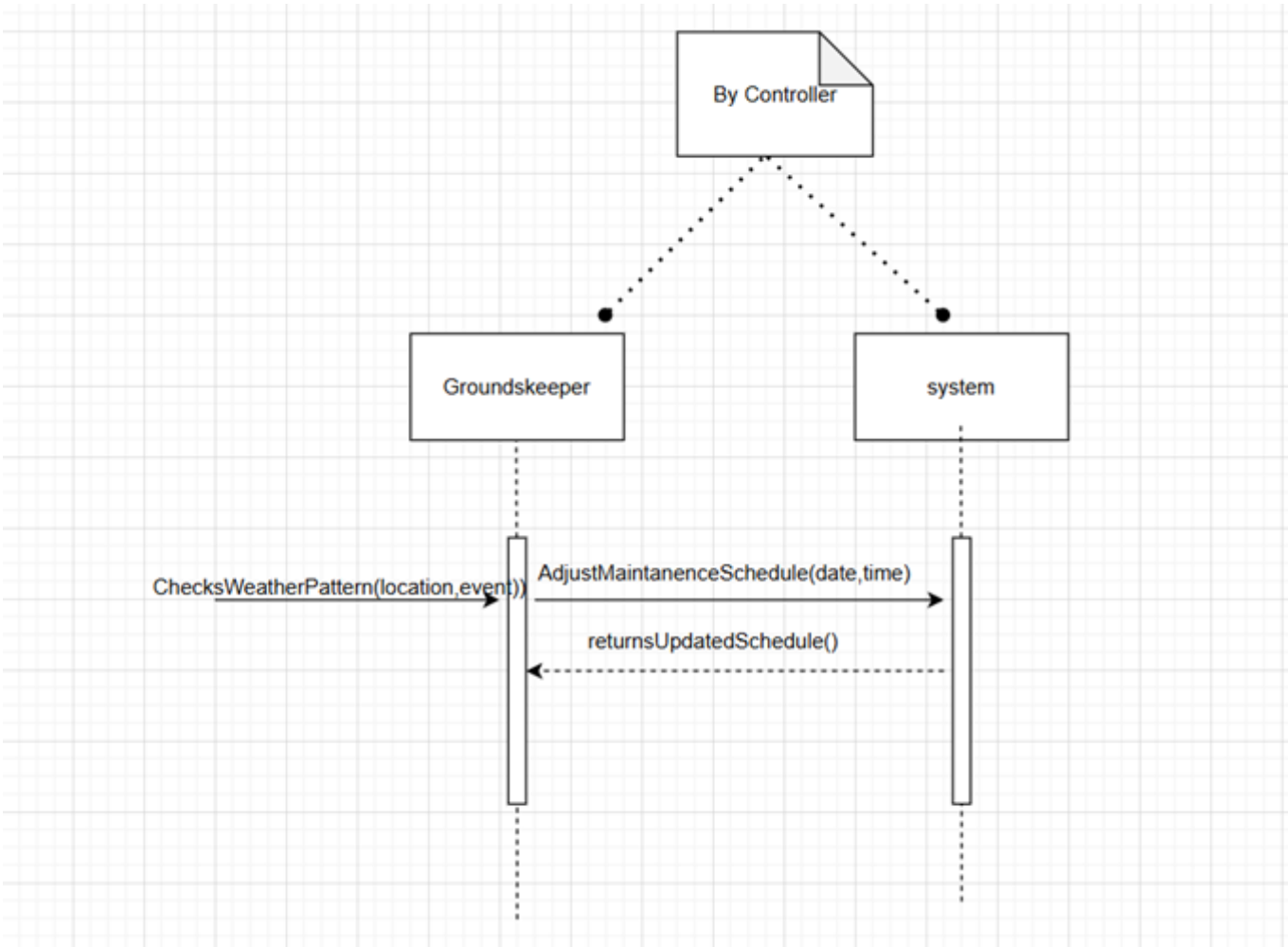
6) Weather monitoring:
Access weatehr module:



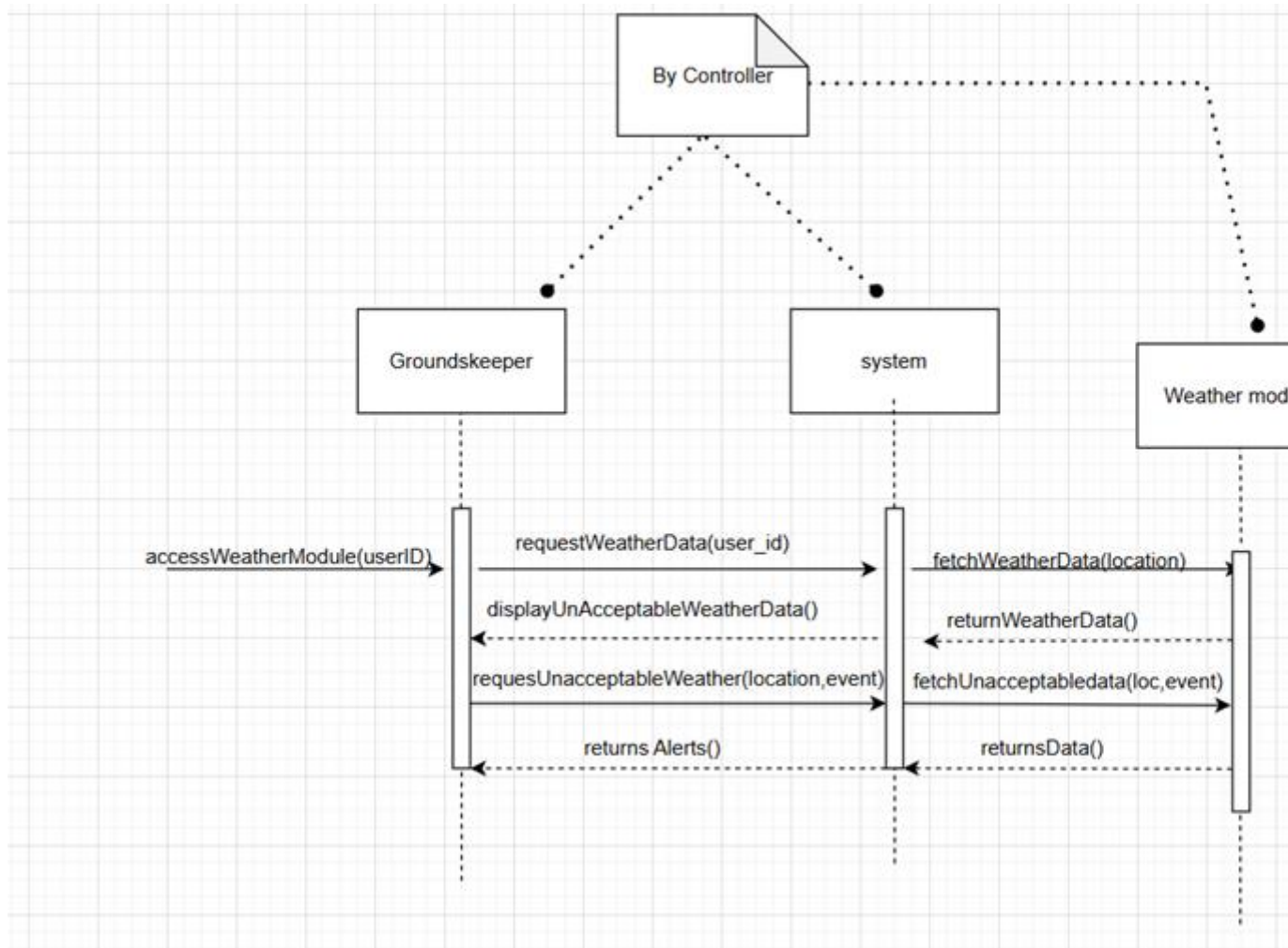
Checks weather data:



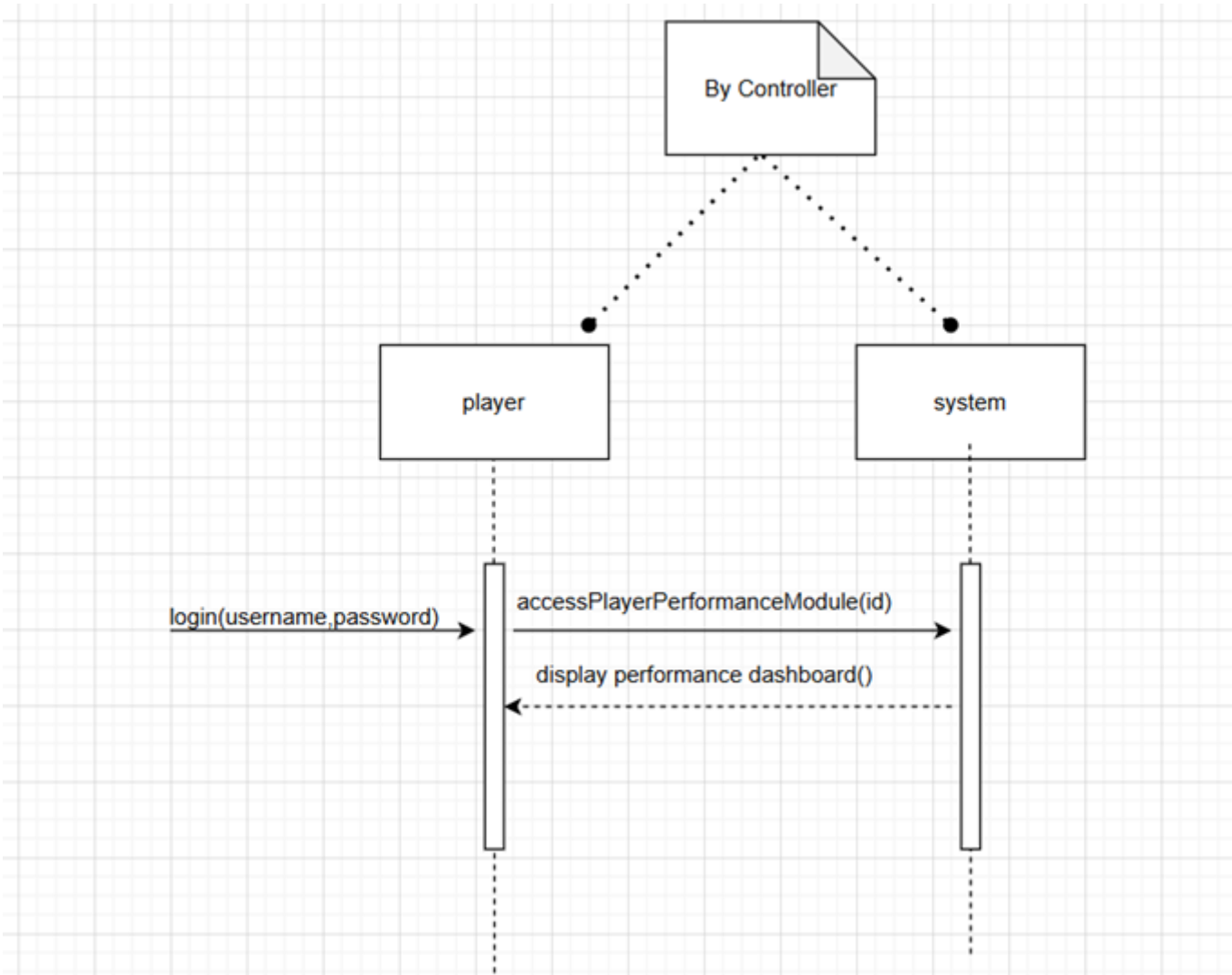
Adjust maintenance schedule:



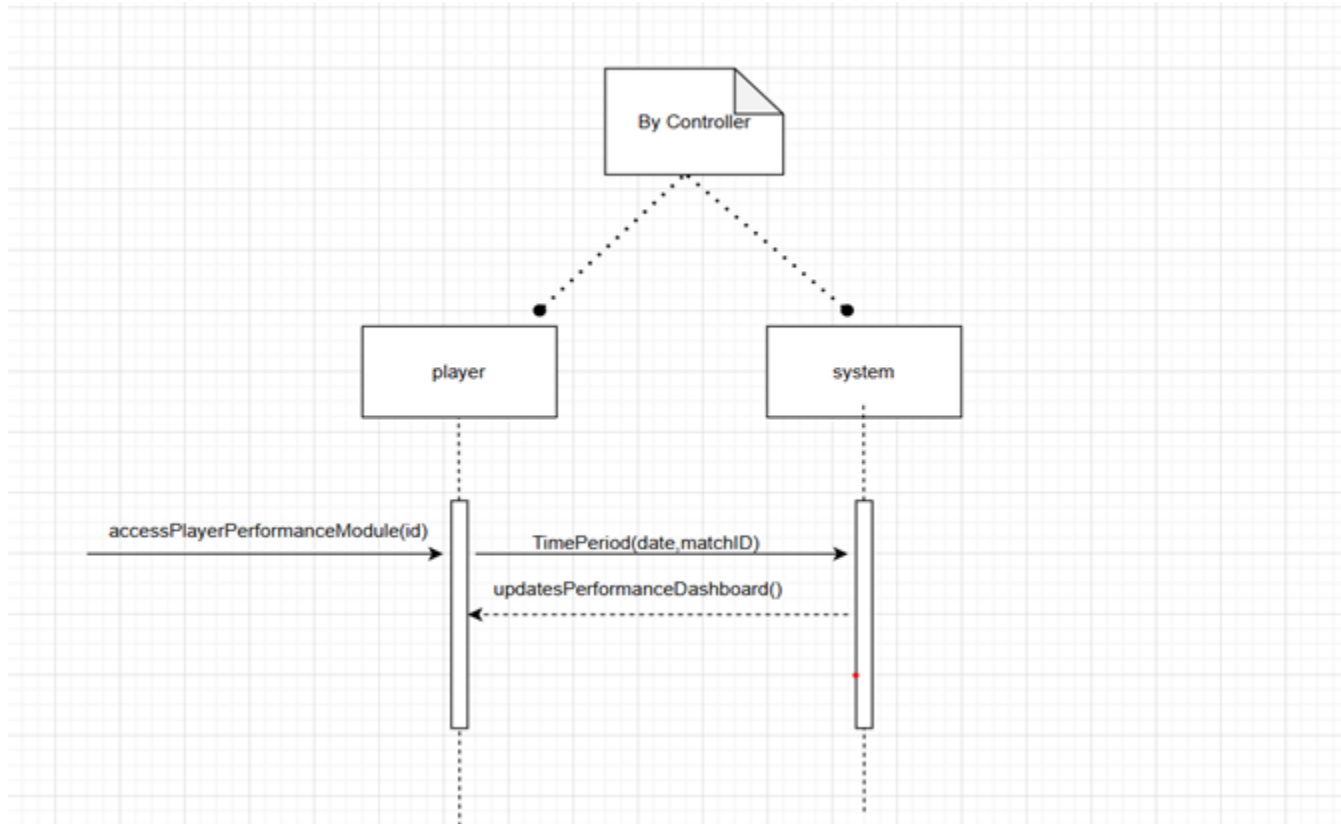
Alternate Scenario:



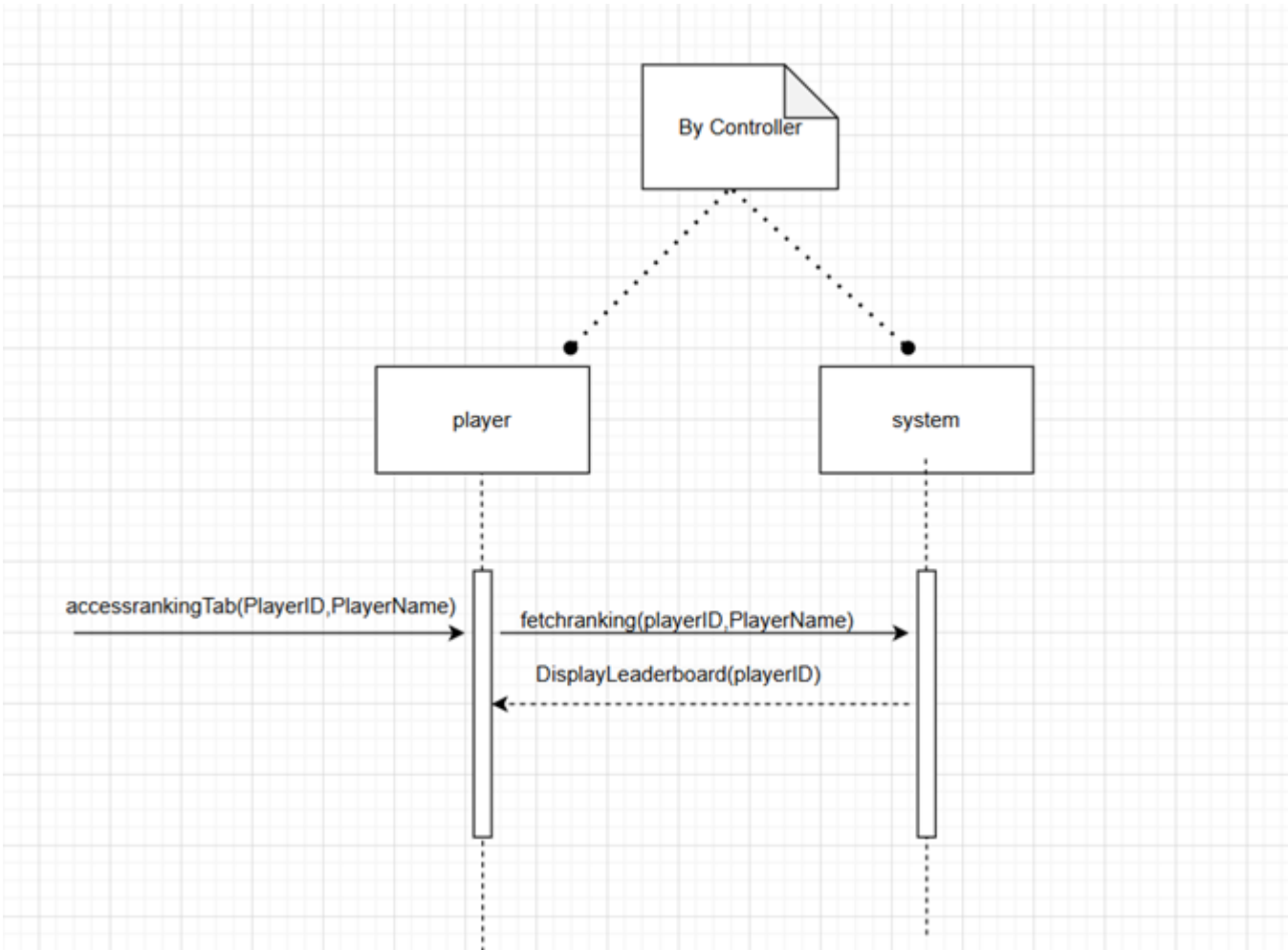
7) Player performance tracking and ranking: Access performance tracking module:



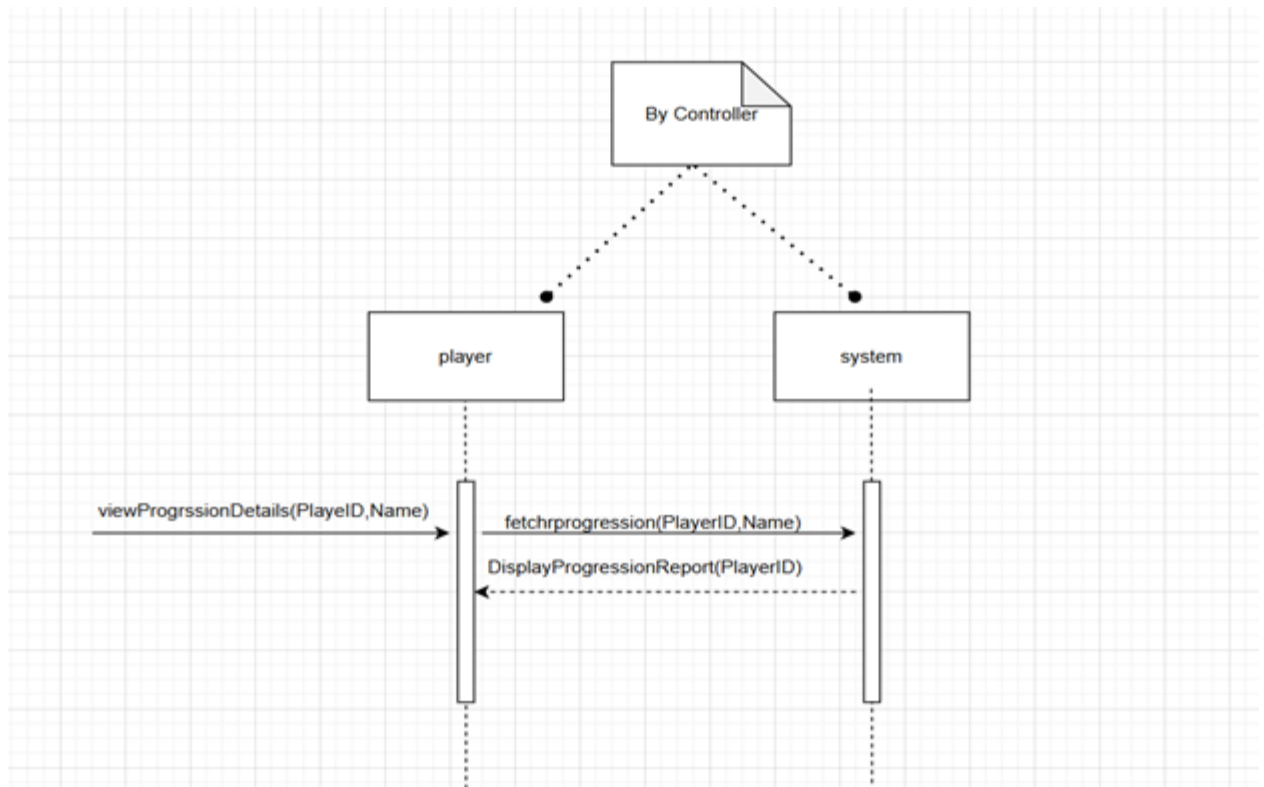
Time period:



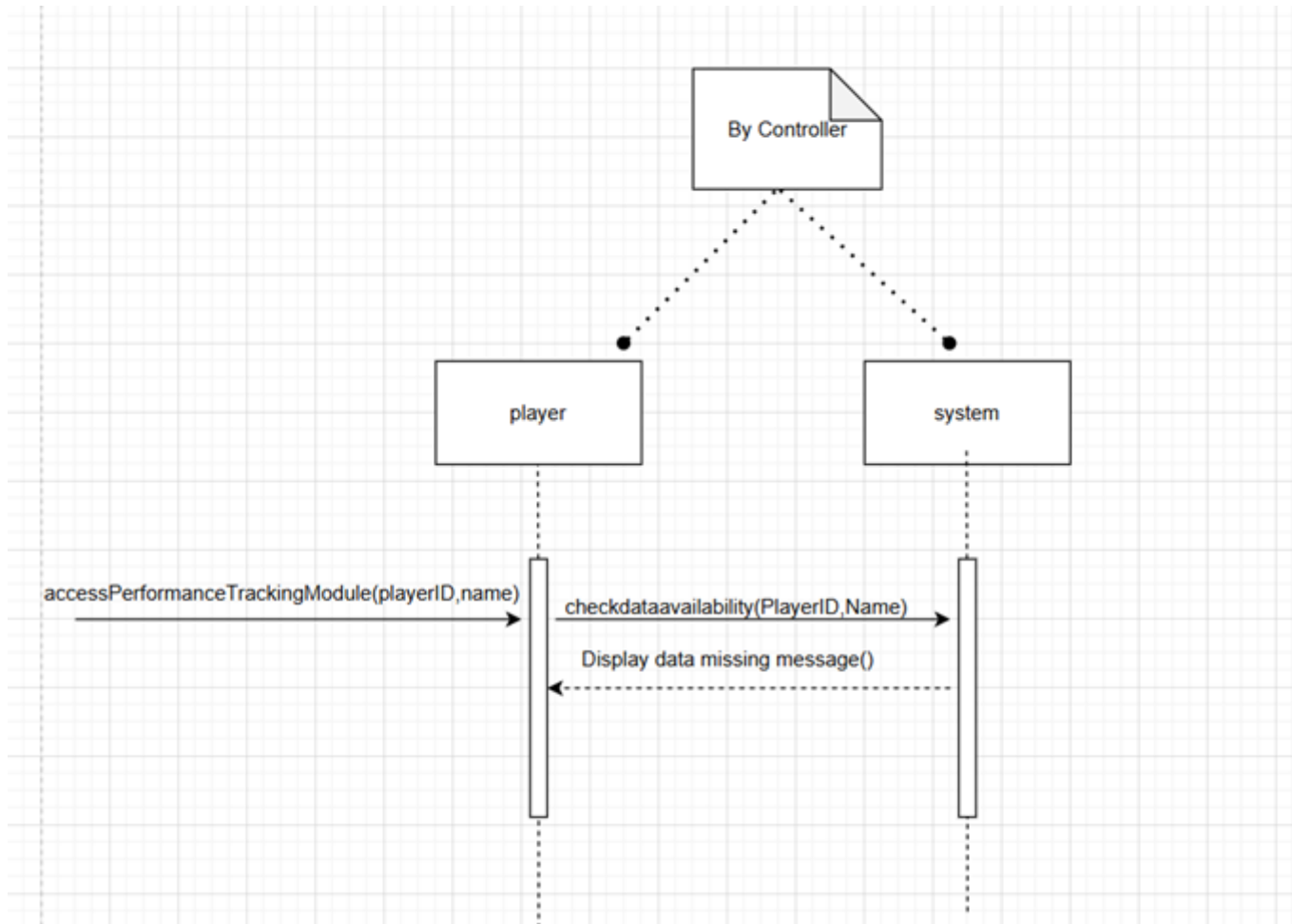
Access Ranking tab:



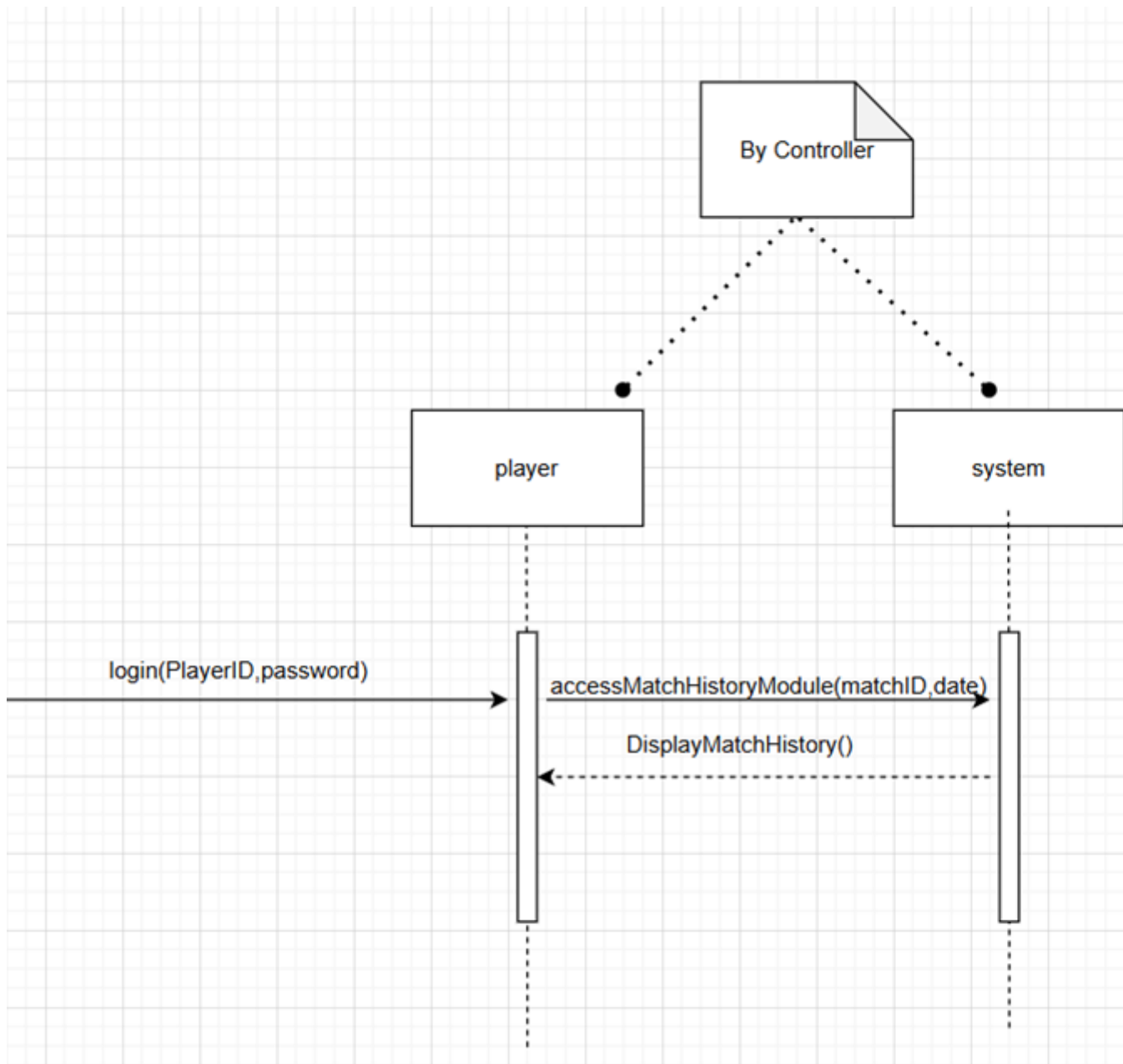
Progression details:



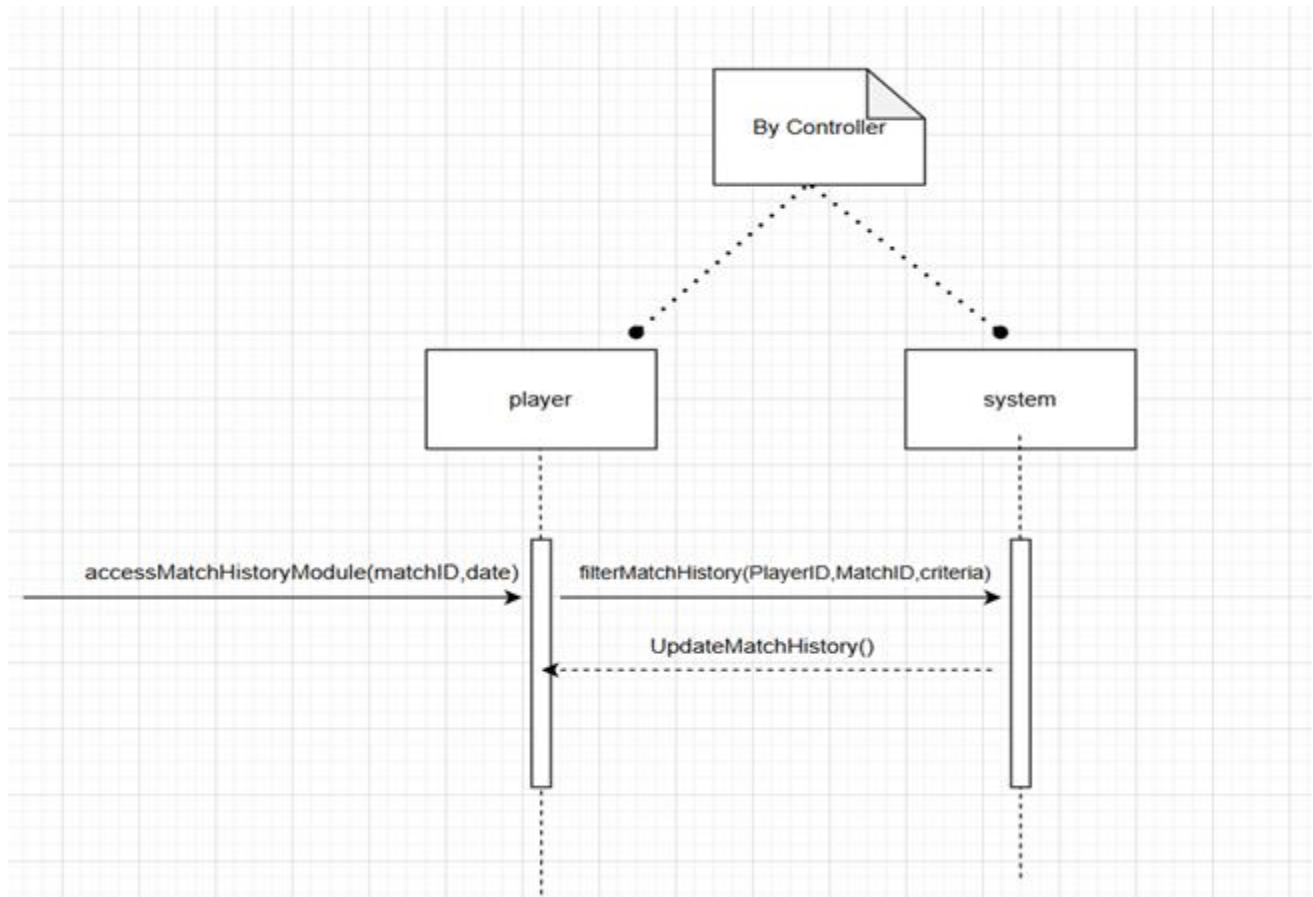
Alternate scenario:



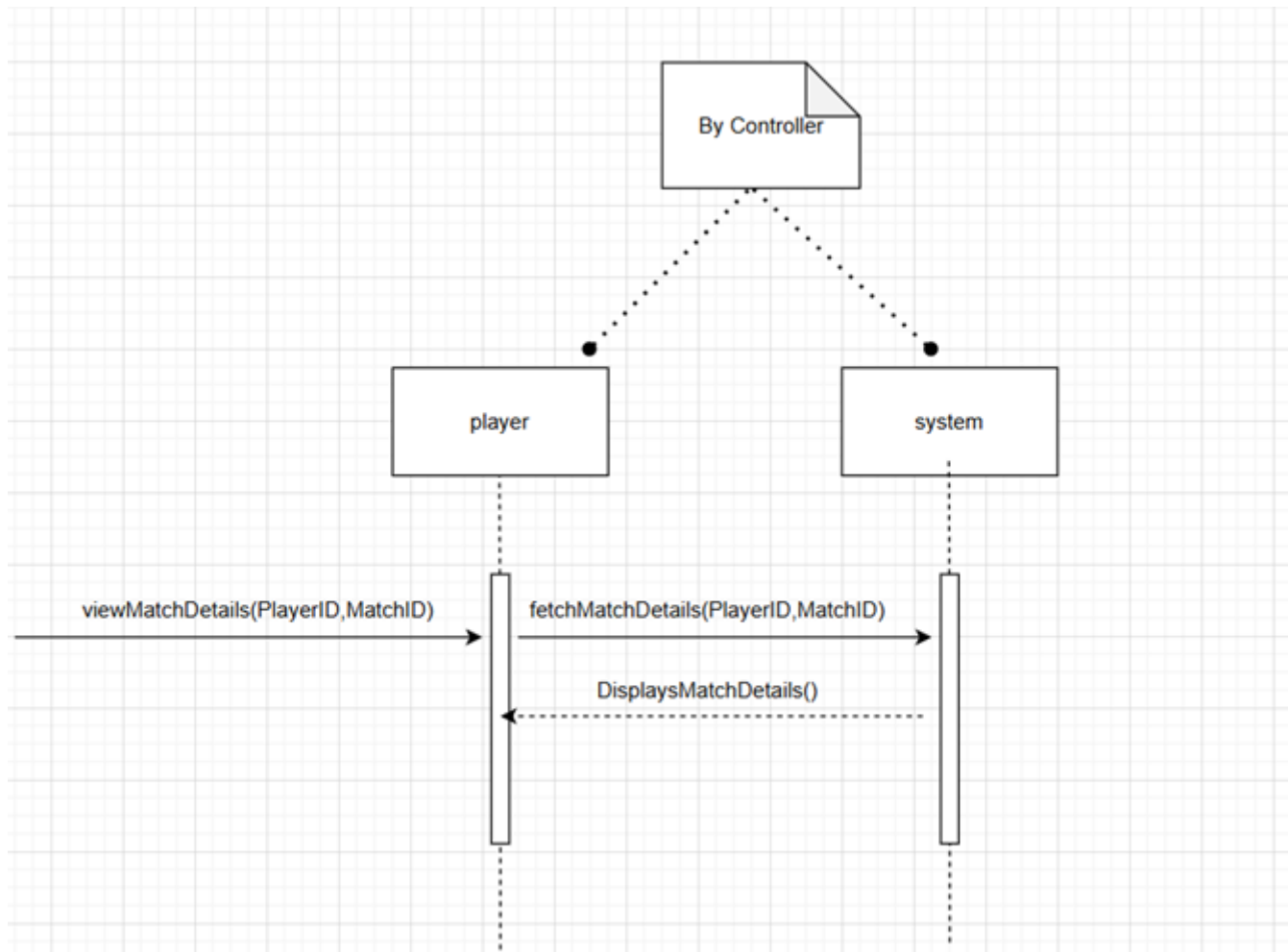
8) Match performance history:
Access match performance module:



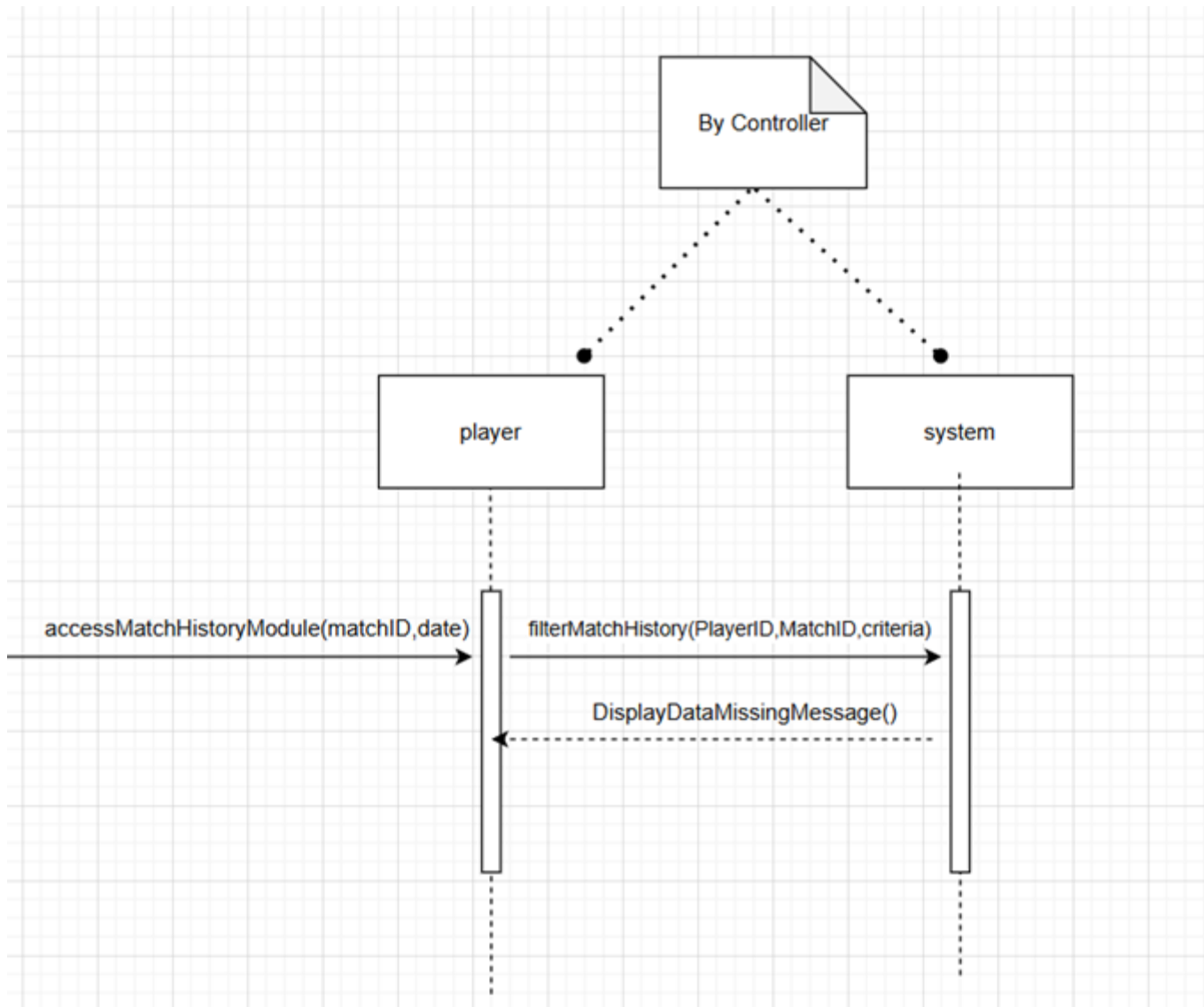
Filter match history:



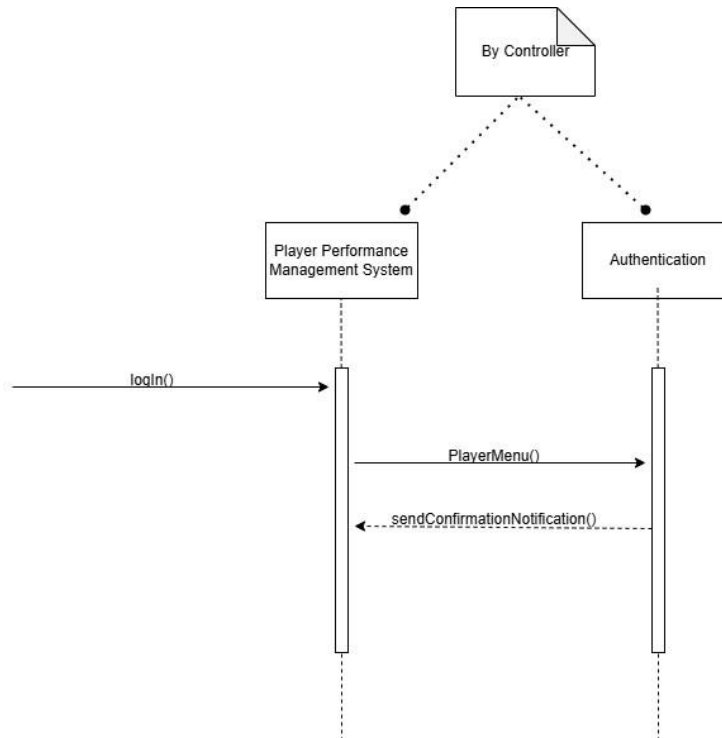
View match details:



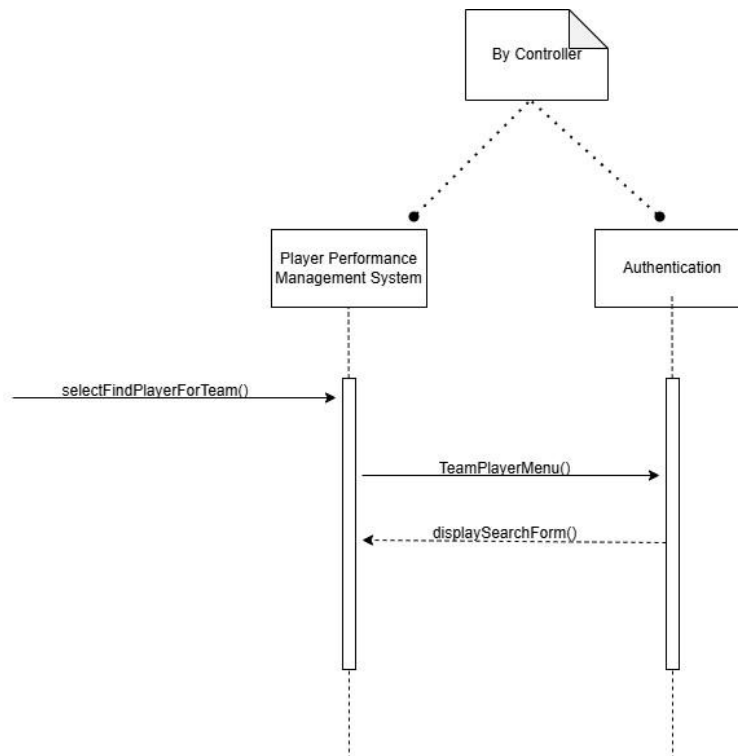
Alternate Scenario:



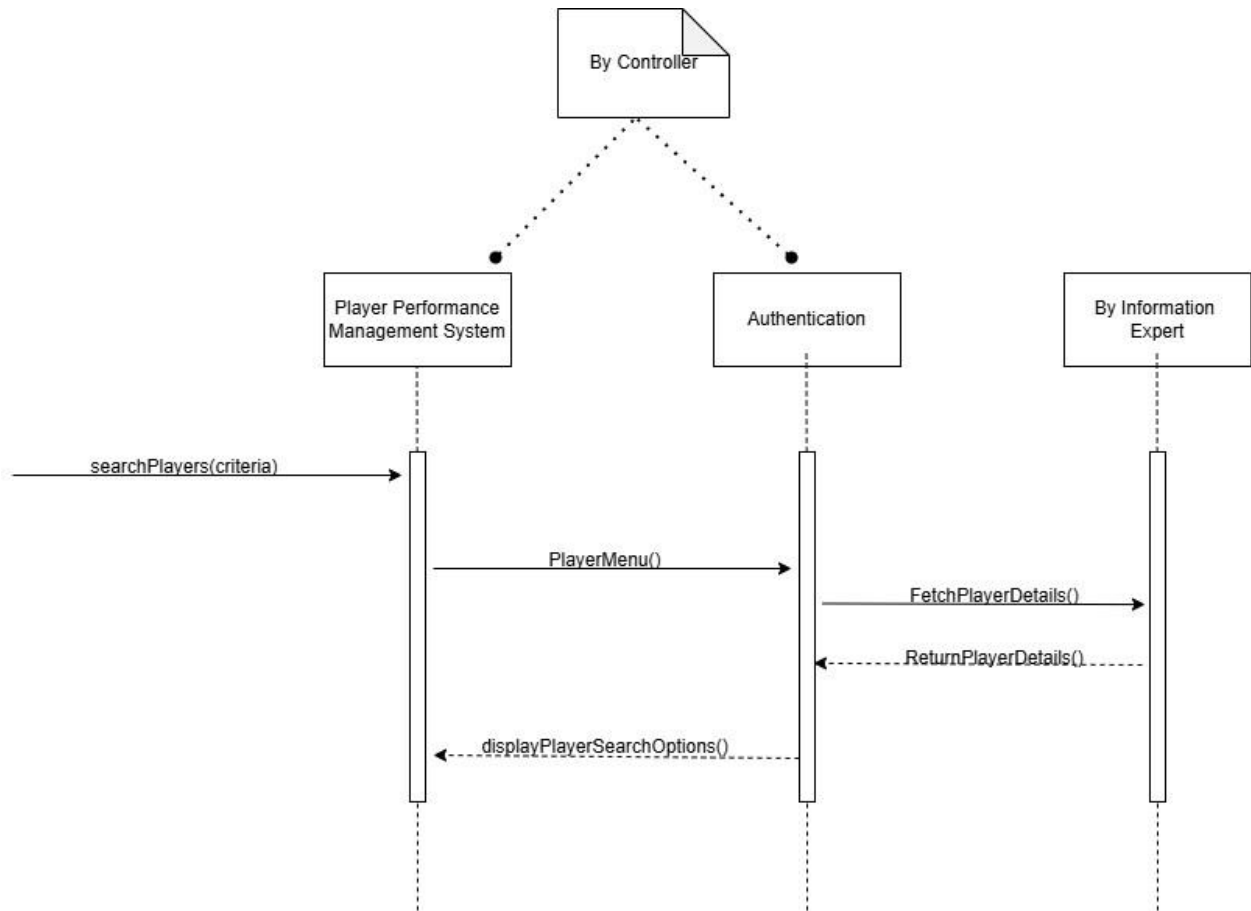
9) Find a player:
Login:



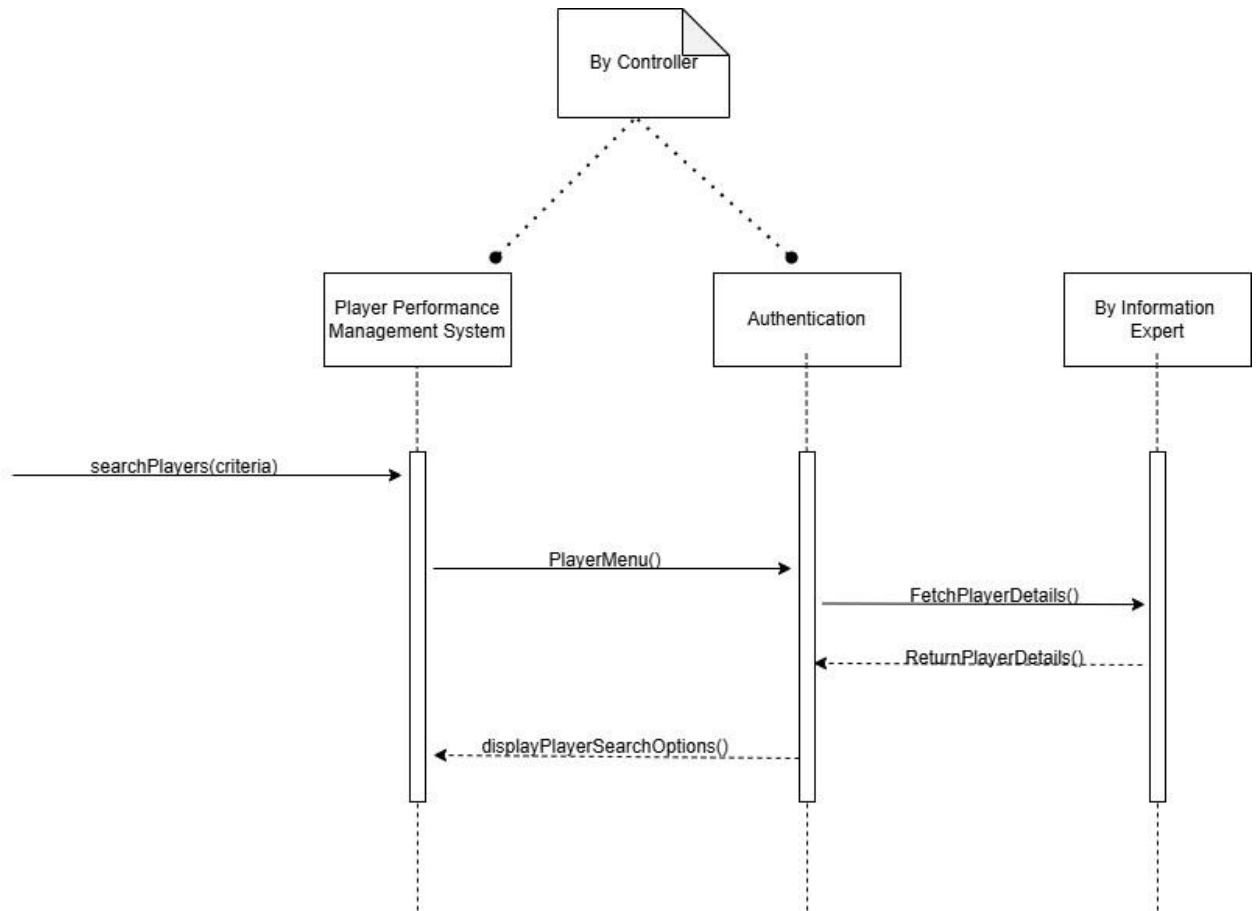
Select find player:



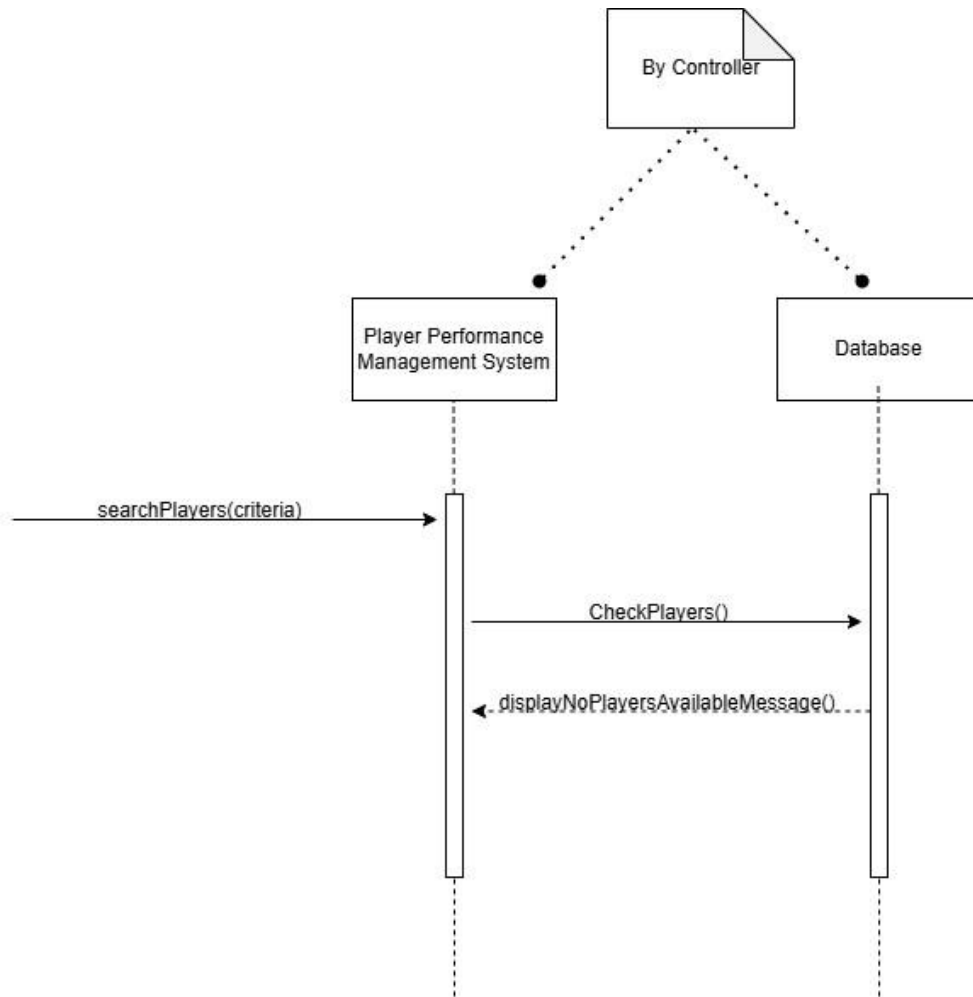
Select Player:



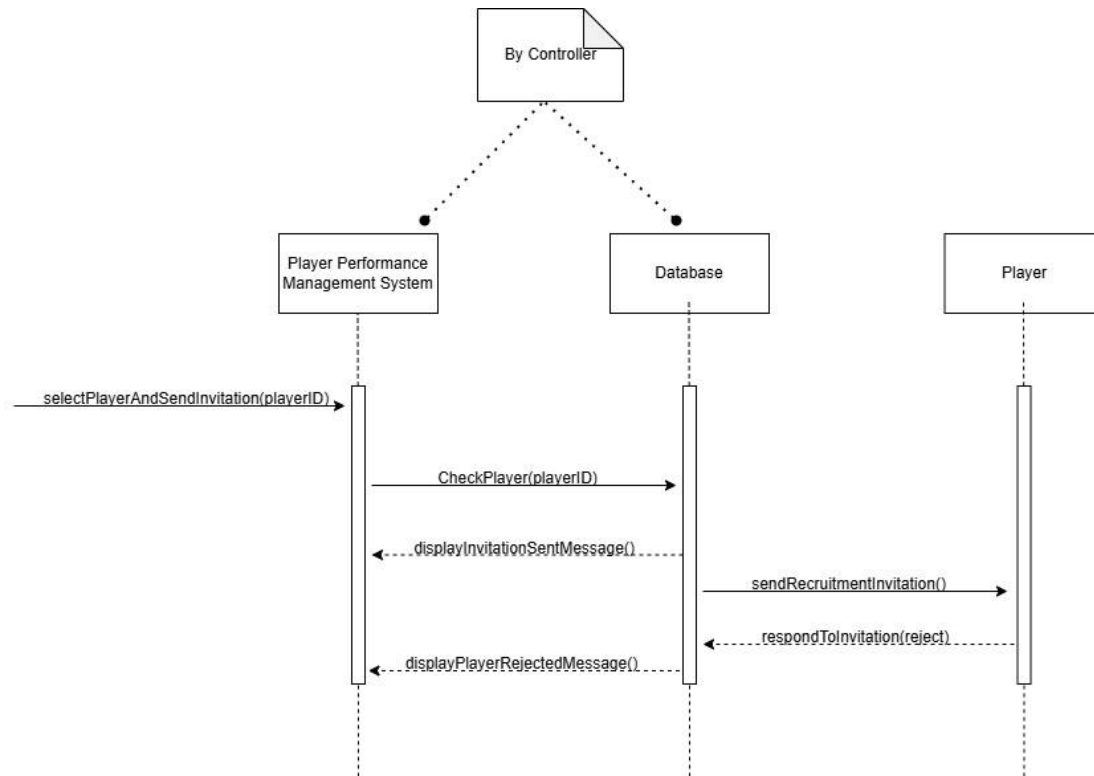
Send request to player:



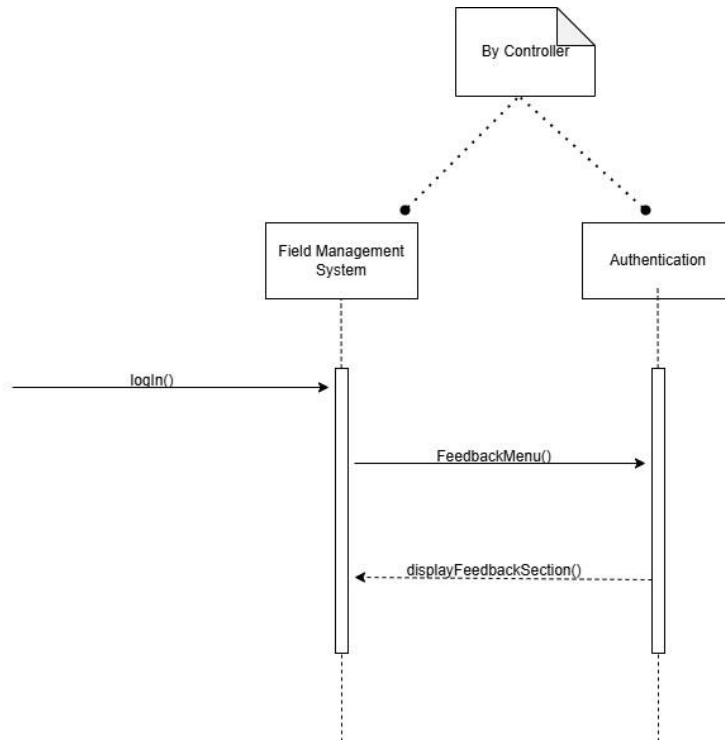
Alternate scenario 1:
No player available:



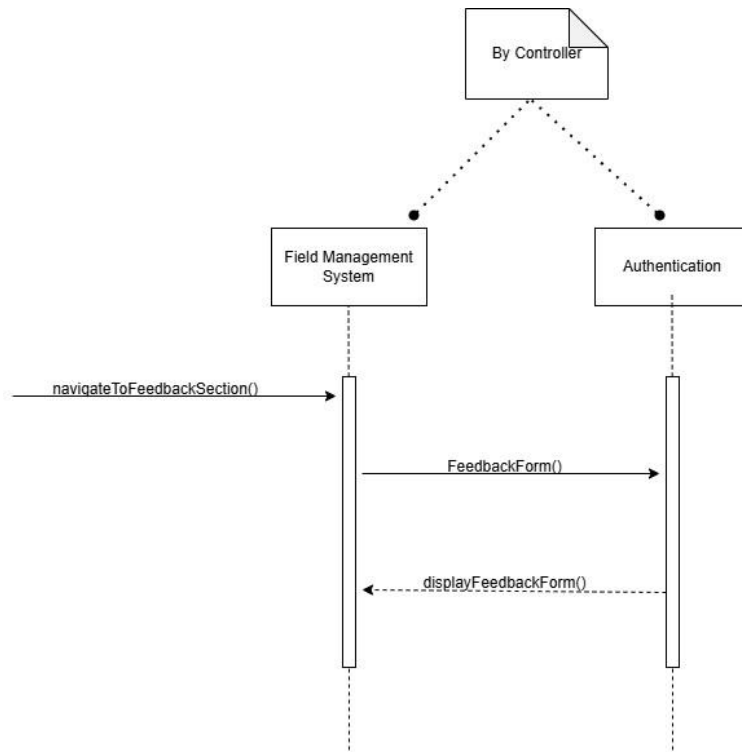
Alternate scenario 2:
Selected player rejects invite:



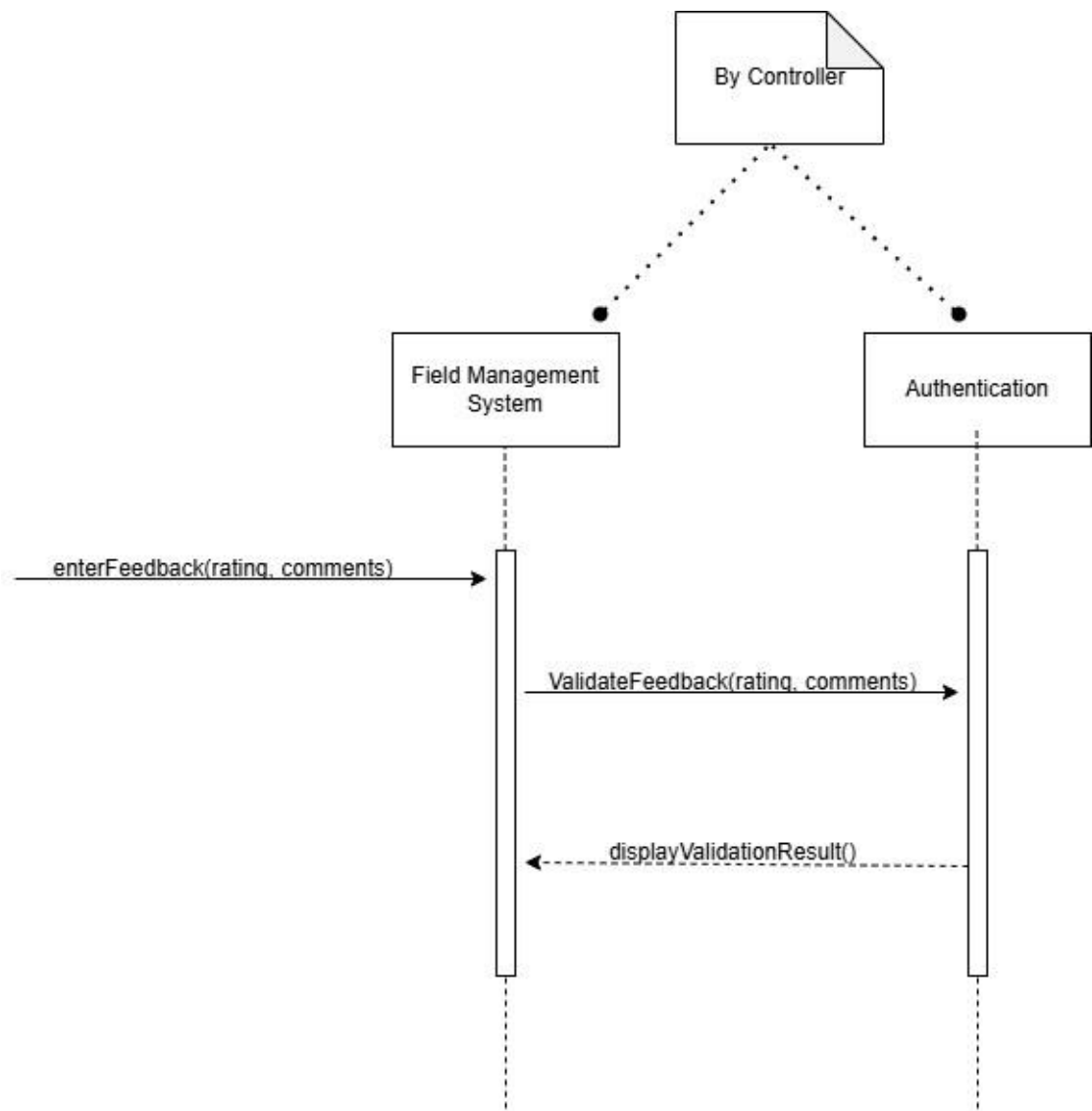
10) Feedback submission: Login:



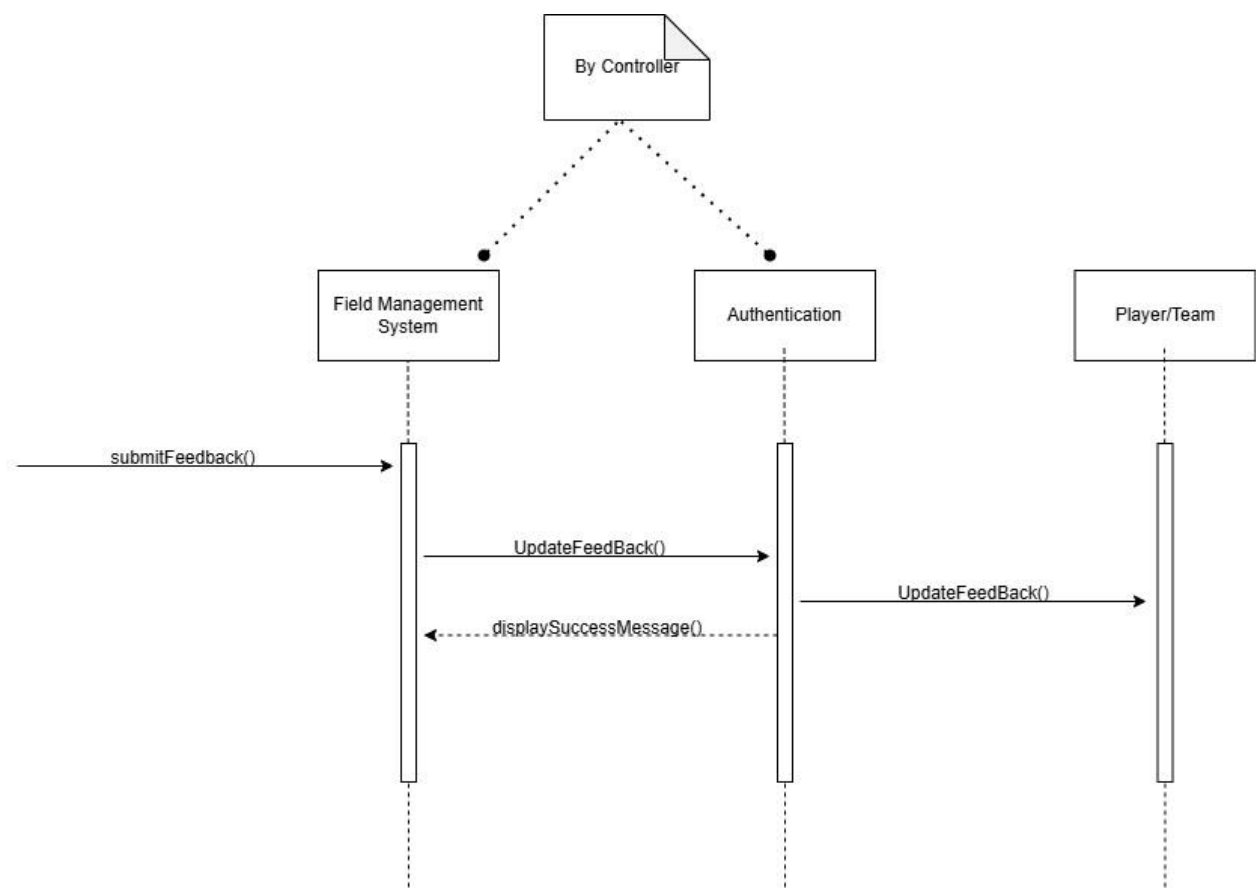
Navigate to feedback section:



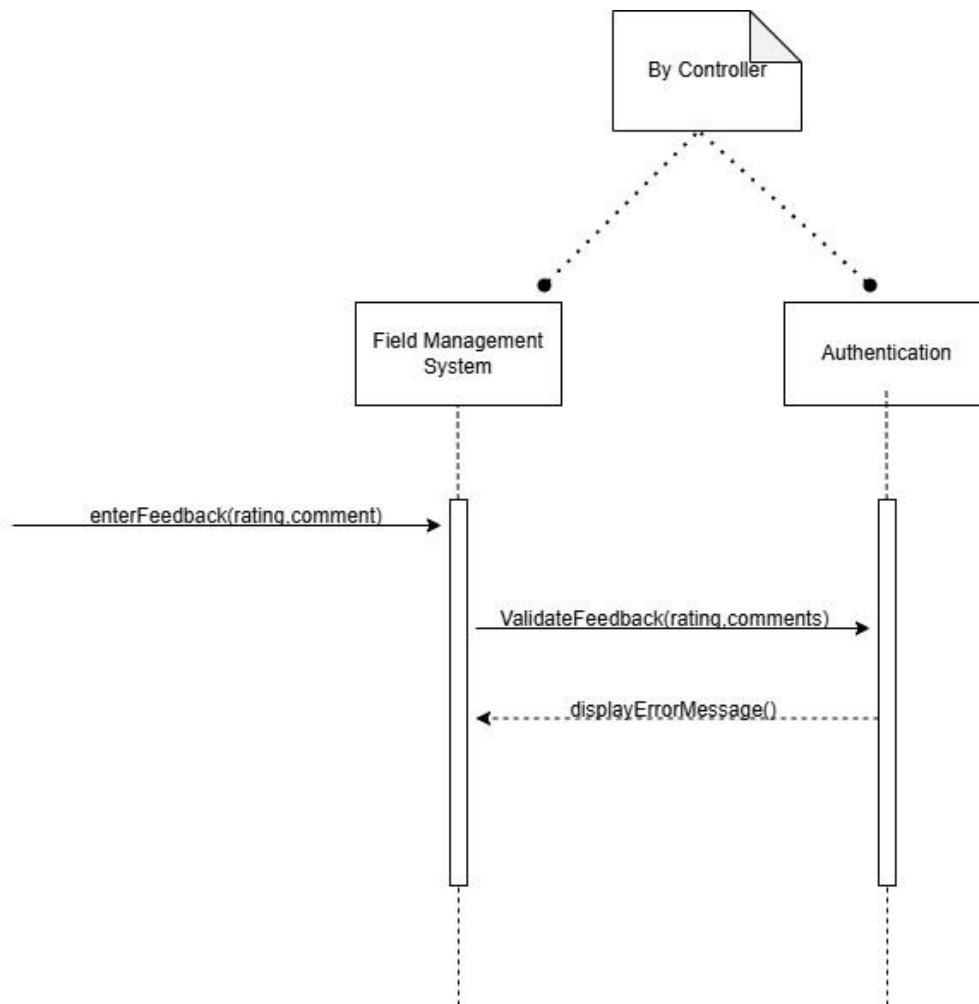
Enter feedback:



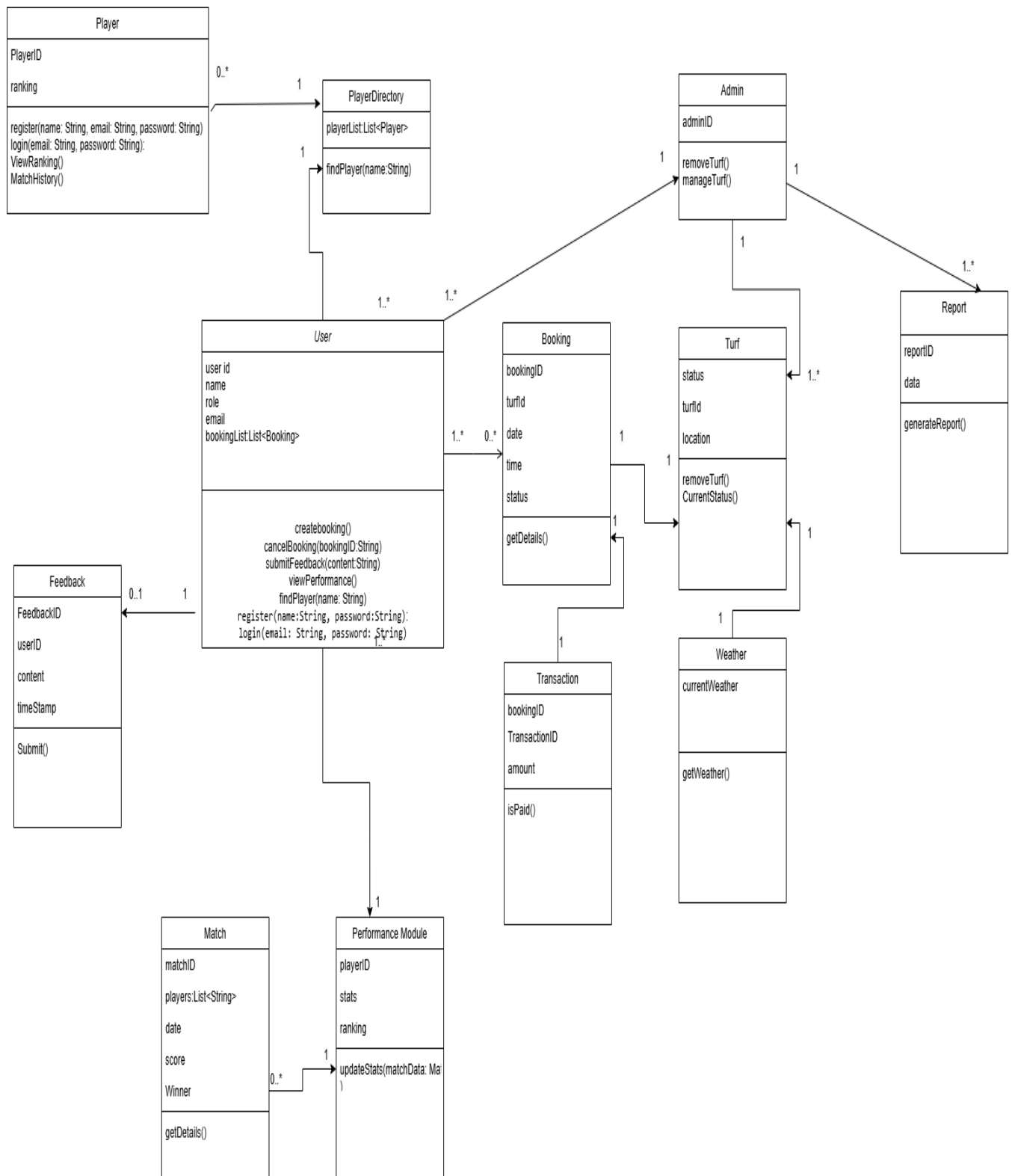
Submit feedback:



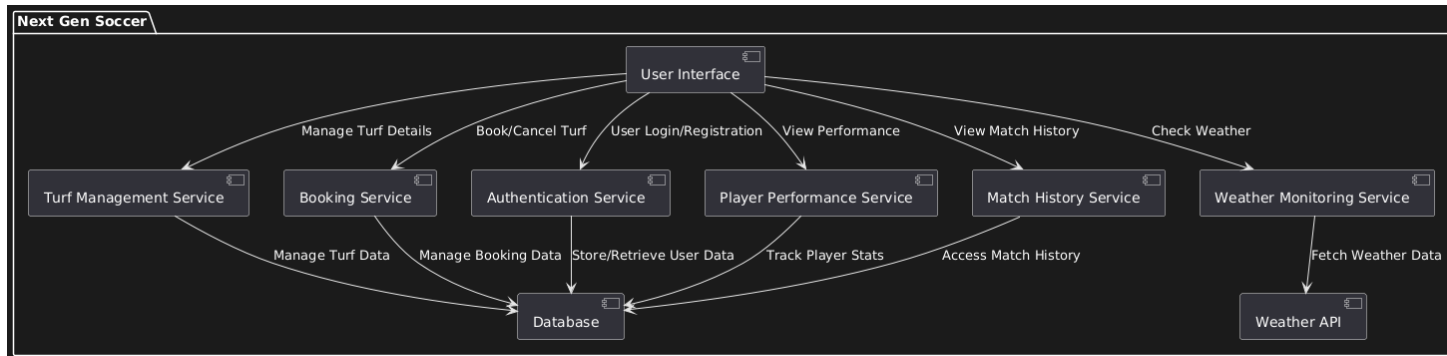
Alternate scenario:
Invalid input:



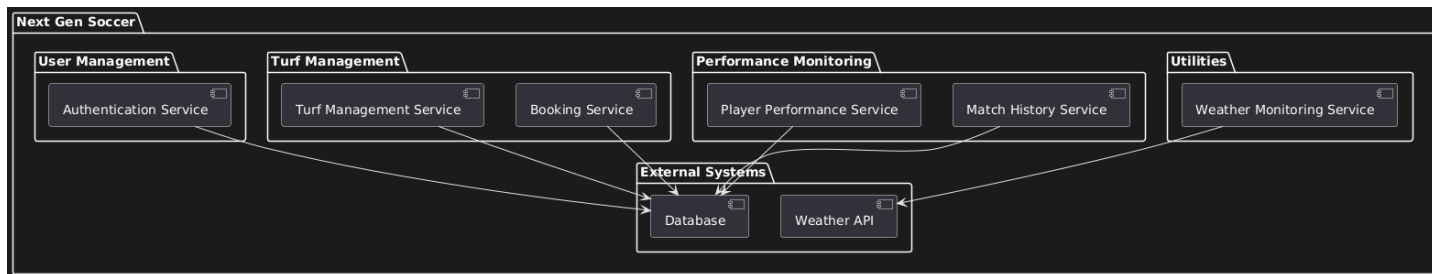
7. Class Diagram



8. Component Diagram



9. Package Diagram



10. Deployment Diagram

