OOAD Project

By

Syed Irtaza Ali

i121612

Nauman Hayat

I120207

Table of Contents

* Vision Document
* Domain Model (Attached)
* Operation contracts
* Sequence Diagrams
* Class Diagram (Attached)

VISION DOCUMENT

Introduction

FOOTMASS -“Football Team Management and Scouting System” is a desktop based application that will give its users information about player performance and statistics. It is intended to be used by football Management professionals and club scouts. To assist managers, coaches and scouts

Purpose

To develop a system that records player performance data in real time and uses it to generate statistical information and performance analytics for football management (Manager/Coaching staff).   
  
This information will give the manager statistical insight, and will help him/her to improve game-tactics, team formation, and individual/team development.   
The system can also be used by a club’s scouting network, to find potential signings in the large pool of International football professionals, based on the required performance traits submitted by the scouting manager.

## Actor-Goal List

|  |  |
| --- | --- |
| Actor | Goal |
| ***Primary Actors:*** |  |
| Team Manager` | * Views player/team performance reports * Shortlist scout report and choose potential signing, finalize transfer-demand * Approach Transfer Agents. |
| Coach | * Views player/team performance reports |
| Scout | * Search for required characteristics in the player pool * Suggest potential signings shortlist |
| **Supporting Actors:** |  |
| Match Analyst | * Enter real-time performance data |
| Database-Manager | * Update player pool * Add a player * Delete a player * Edit player information i.e. name, number |
| Transfer Agent | * Sign a player |

Product Solution Overview

FTMS will be a Desktop application that provides statistical insight to managers and scouts and help solve management and scouting challenges by drawing inferences from real-time performance statistics collected by a Match Analyst.  
 Information about a player is set up by the database manager. A Match Analyst enters performance data in real time, based on his observation during the match. Commits changes to the database as soon as the final whistle blows. The system generates this information into reports for team manager/coaches and scouts.   
A manager may inquire for potential transfers from a scout, who accesses the system to obtain a shortlist of players which possess, the set of characteristics the manager has been looking for, and lets the system forward a shortlist to the manager, A manager may further shortlist the report and may choose to inquire about them from the transfer agent, the transfer agent then proceeds to signs the player. The legal process is out of system boundaries and the transfer agent only commits changes when a player has signed and has the work permit to practice his profession under a new employer, the financial process is also out of the system boundary, whereas a built-in **“player evaluation”** tool evaluates an approximate player transfer-value based on his performance-statistics .

Version 2.0

In a later version, we plan to have FTMS generate live performance data via processing images using an Aerial match camera, (tower cam) and minute by minute heat-map statistics showing time-lapse of individual/team presence on the field.   
  
Version 3.0

We also aim to extend it across all-major mobile platforms Smart-phones (android/ios/windows) and achieve to have form a cloud and increase parameters we collect (ie passes completed) taking our cloud to a “big-data” horizon. With a “big-data” cloud we aim to evolve into making big-decisions. And provide analytics on decision-time and access the choices a player makes in real time. We aim to revolutionize football management, without interfering with the ruling of the “beautiful game”.

## High level Use cases

**Use case:** update (database) players/teams/manager

**Actors:** Database manager  
**Type:** Supporting

**Description:**

A database Manager maintains an up to date system including current squads/personal information also stores club affiliations of players and team manager. A database manager commits less frequent changes to the database, twice annually (during transfer windows) or in case of a system maintenance

**Use case:** add Performance data

**Actors:** Match Analyst  
**Type:** Supporting

**Description:**

A match Analyst uses the system when a match starts and enters real-time performance data, i.e. passes played by xyz player etc. The match analyst commits final changes sometime after the final whistle.

**Use case:** View Manager Reports

**Actors:** Team Manager, Coaching Staff  
**Type:** Primary

**Description:**

A Team Manager selects which type of report he needs, queries by the manager are generated into reports and the manager leaves with information drawn from the performance data collected in the “add performance data” use case.

**Use case:** Search for characteristics

**Actors:** Scout   
**Type:** Primary

**Description:**

A scout uses the system to search for the required characteristics in a potential signings, the manager wishes to undergo, characteristics maybe statistical information i.e. *“find a CAM (central attacking midfielder), less than 25 years old, with a passing accuracy greater 75%”.*

**Use case:** Suggest potential signings

**Actors:** Scout, Team Manager  
**Type:** Primary

**Description:**

Pre-req. “Search for characteristics”. The system shows a list of players with matching or close characteristics. The scout shortlists and chooses some of them to forward to the manager suggesting potential signings.

**Use case:** Sign a player

**Actors:** Team Manager, Transfer Agent  
**Type:** Supporting

**Description:**

Pre-req. “Search for characteristics” and “Suggest potential signings”. The Manager chooses one final name after the entire scouting process, and forwards it to the transfer agent responsible for handling out the task. The transfer agent signs the player, and commits “modify player club” sub-functionality from the Database Managers role.

**Operation Contracts**

# Contract for: Record Match Stats

# 

Name: RecordMatchStats (Fouls\_Committed, Passes\_Attempted,

Passes\_Completed, Red\_Cards\_awarded, Saves, Shots\_Attempted,

Shots\_on\_Target, Tackles\_Attempted, Tackles\_Completed , Yellow\_Cards\_Awarded).

Responsibility: Adds Player Match Stats in the stats pool.

Cross Reference: UC: Add Performance Data.

Exception: If the Match Analyst enters a wrong statistical information of a player as input an error message is generated.

The Match Analyst must enter the total information of the player otherwise the system will generate error message.

Pre-conditions: A Player must exists because it describes the Player\_Stats.

# Contract for: Place Sell a Player Request

Name: Place\_Sell a Player\_Request (Player,Transfer type).

Responsibility: Team Manager places the request for a player to be sold and is

kept in a transfer request register.

Cross Reference: UC: Place Transfer Request (Sell a Player).

Exceptions: If the transfer agent is not authenticated it means he is trying to transfer a player that doesn’t belong to his affiliated club.

Pre-conditions: The Player must exist.

Post-conditions: The Transfer request instance is created and contains the reference to the “Player” that is to be transferred.

The Transfer Type attribute is set to “sell”.

All the transfer request instances are associated (contained) with a transfer request register.

# Contract for: Place sign a player request

Name: Place\_Sign a Player\_Request (Player,Transfer type).

Responsibility: Team Manager places the request for a player to be signed and is

kept in a transfer request register.

Cross Reference: UC: Place Transfer Request (Sign a Player).

Exceptions: If the transfer agent is not authenticated it means he is trying to transfer a player that doesn’t belong to his affiliated club.

Pre-conditions: The Player must exist.

Post-conditions: The Transfer request instance is created and contains the reference to the “Player” that is to be transferred to the club.

The Transfer Type attribute is set to “sign”.

All the transfer request instances are associated (contained) with a transfer request register.

# Contract for: Search by characteristics.

Name:

Search\_by\_characteristics(Min\_passing\_accuracy\_percentage,Max\_passing\_accuracy\_percentage,Min\_saves\_accuracy\_percentage,Max\_saves\_accuracy\_percentage, Min\_shooting\_accuracy\_percentage,Max\_shooting\_accuracy\_percentage, Min\_tackling\_accuracy\_percentage,Max\_tackling\_accuracy\_percentage,Min\_goals\_per\_game,Max\_goals\_per\_game,Min\_assists\_per\_game,Max\_assists\_per\_game)

Responsibility: To generate the reports containing desired player for the scout depending upon the characteristics scout entered.

Cross Reference: UC: Scout Players

Exceptions: The scout’s required characteristics don’t match.

Scout enters ambiguous query.

Pre-conditions: The stat pool must contain instances of player stats.

Post-conditions: Scout Report object is created.

Scout Report attributes are modified.

Scout Report object is associated with Scout Report Register.

# Contract for: Acquire transfer request list.

Name: Acquire\_Transfer\_request\_list()

Responsibility: To show the transfer request list of the selling players to Transfer agent that is forwarded to him by the Team Manager.

Cross Reference: UC: Sell a Player

Exceptions:

Pre-conditions: The Transfer request register must contain Transfer requests placed by the Team Manager.

Post-conditions:

# Contract for: Process Transfer Request.

Name: Process\_Transfer\_Request(Transfer\_Request)

Responsibility: To process the transfer request that is chosen by the Transfer Agent.

Cross Reference: UC: Sell a Player.

Exceptions:

Pre-conditions: The Transfer request list has been acquired by the Transfer Agent.

Post-conditions: Transfer\_Request object is initiated.

Transfer\_Request object attributes are modified.

Transfer\_Request object is associated with Transfer market.

# Contract for: Acquire transfer request list.

Name: Acquire\_Transfer\_request\_list()

Responsibility: To show the transfer request list of the signing players to Transfer agent that is forwarded to him by the Team Manager.

Cross Reference: UC: Sign a Player

Exceptions:

Pre-conditions: The Transfer request register must contain Transfer requests placed by the Team Manager.

Post-conditions:

# Contract for: Process Transfer Request.

Name: Process\_Transfer\_Request(Transfer\_Request)

Responsibility: To process the transfer request that is chosen by the Transfer Agent.

Cross Reference: UC: Sign a Player.

Exceptions:

Pre-conditions: The Transfer request list has been acquired by the Transfer Agent.

Post-conditions: Transfer\_Request object is initiated and its “player and club” attributes are matched to the corresponding attributes of the object present in Transfer Market.

If match is found the object in the transfer market is de-associated and the corresponding player object’s attribute “Club” in the player pool is modified.

# Contract for: Add a new Player

Name: Add\_a\_new\_Player(Age,Club,Player\_Name,Position,Role)

Responsibility: To add the Player along with information of the Player

Cross Reference: UC: Update Player/Team/Clubs and Manager.

Exceptions: In case the club doesn’t exist. An error message is displayed “club not found”.

Pre-conditions: Club must exist.

Post-conditions: Player instance is created.

Attributes of Player are modified.

Player is associated with the player pool.

# Contract for: Add Team Manager

Name: Add\_Team\_Manager (Club\_Affiliation,Manager\_Name).

Responsibility: To add the Team Manager along with information of the Team Manager.

Cross Reference: UC: Update Player/Team/Clubs and Manager.

Exceptions: In case the club doesn’t exist. An error message is displayed “club not found”.

Pre-conditions: Club must exist.

Post-conditions: Manager instance is created.

Attributes of Manager are modified.

Manager is associated with the Team Manager register.

# Contract for: Add Club

Name: Add\_Club(Club\_Name).

Responsibility: To add the Club along with information in the Club.

Cross Reference: UC: Update Player/Team/Clubs and Manager.

Exceptions:

Pre-conditions:

Post-conditions: Club instance is created.

Attribute of club is modified.

Club is associated with the Player.

# Contract for: Delete Player

Name: Delete\_Player(Player\_Name)

Responsibility: To delete the Player along with information in the Player

Cross Reference: UC: Update Player/Team/Clubs and Team Manager.

Exceptions: In case the Player doesn’t exist. An error message is displayed “Player not found”.

Pre-conditions: Player must exist.

Post-conditions:

If Player\_Name is matched with a Player’s “Player\_Name” present in the player pool that player is disassociated from the player pool.

# Contract for: Delete Team Manager

Name: Delete\_Team\_Manager(Team\_Manager\_Name)

Responsibility: To delete the Team Manager along with information in the Team Manager.

Cross Reference: UC: Update Player/Team/Clubs and Team Manager.

Exceptions: In case the Team Manager doesn’t exist. An error message is displayed “Team Manager not found”.

Pre-conditions: Team Manager must exist.

Post-conditions:

If Team\_Manager\_Name is matched with a Team Manager’s “Team \_Manager \_Name” present in the Team Manager register that Team Manager is disassociated from the Team Manager register.

# Contract for: Player to be edited

Name: Edit\_Player(Player\_Name)

Responsibility: To find the Player attributes of the specified player.

Cross Reference: UC: Update Player/Team/Clubs and Manager.

Exceptions: In case the Player doesn’t exist. An error message is displayed “Player not found”.

Pre-conditions: Player must exist.

Post-conditions:

# Contract for: Updated Player

Name: Updated\_Player(Age,Club,Player\_Name,Position,Role)

Responsibility: To modify the Player attributes of the specified player.

Cross Reference: UC: Update Player/Team/Clubs and Manager.

Exceptions: In case the Player doesn’t exist. An error message is displayed “Player not found”.

Pre-conditions: Player must exist.

Post-conditions: Attributes of Player are modified with the new attributes passed to it.

# Contract for: Team Manager to be edited

Name: Edit\_ Team Manager (Manager\_Name)

Responsibility: To find the Team Manager attributes of the specified Team Manager.

Cross Reference: UC: Update Player/Team/Clubs and Manager.

Exceptions: In case the Team Manager doesn’t exist. An error message is displayed “Team Manager not found”.

Pre-conditions: Team Manager must exist.

Post-conditions:

# Contract for: Updated Team Manager

Name: Updated\_Team\_Manager(Club\_Affiliation,Manager\_Name)

Responsibility: To modify the Team Manager attributes of the specified Team Manager.

Cross Reference: UC: Update Player/Team/Clubs and Manager.

Exceptions: In case the Team Manager doesn’t exist. An error message is displayed “Team Manager not found”.

Pre-conditions: Team Manager must exist.

Post-conditions: Attributes of Team Manager are modified with the new attributes passed to it.

# Contract for: View Individual Reports

Name: View\_individual\_report(Player\_Name)

Responsibility: To show the individual reports containing statistical data of the specified player.

Cross Reference: UC: View Squad Reports.

Exceptions: In case the Player doesn’t exist. An error message is displayed “Player not found”.

Pre-conditions: Player must exist.

Post-conditions:

# Contract for: View Team Reports

Name: View\_Team\_Report()

Responsibility: To show the Team Reports containing statistical data of the Team.

Cross Reference: UC: View Squad Reports.

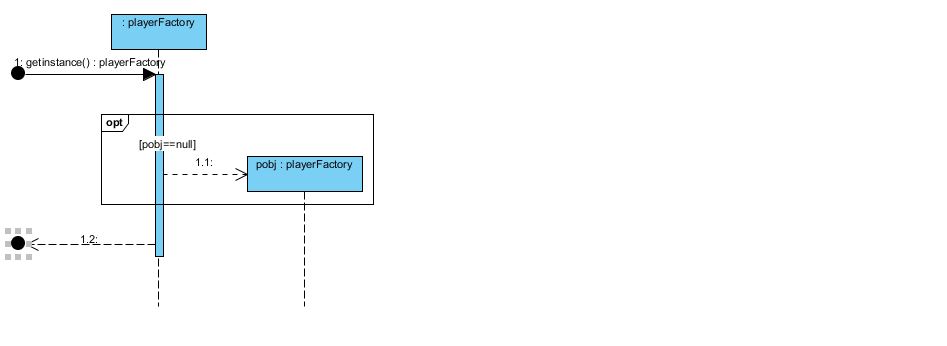
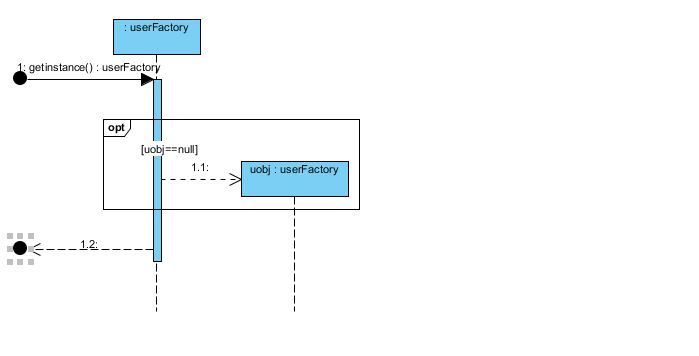
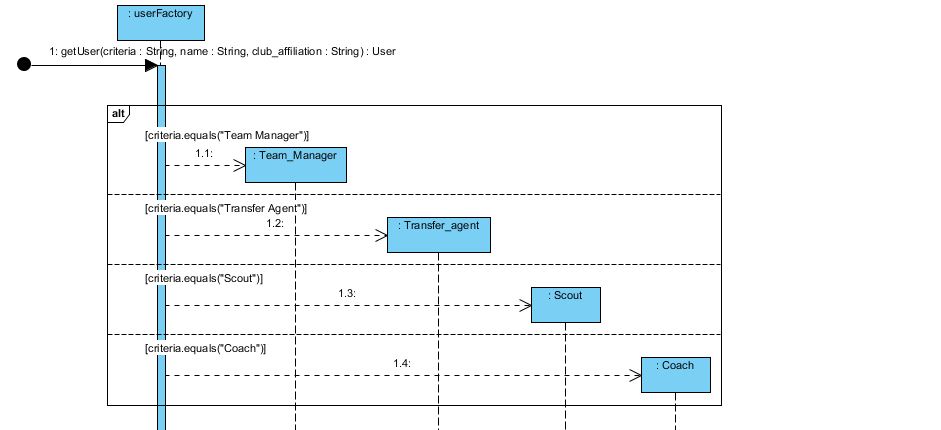
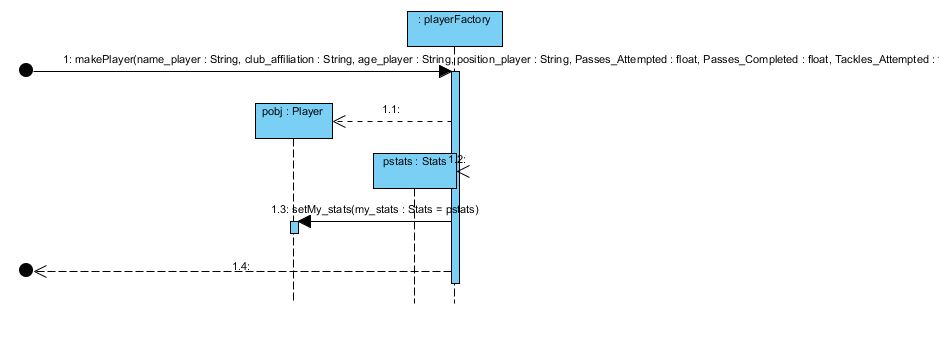
Exceptions:

Pre-conditions: Team Report must exist.

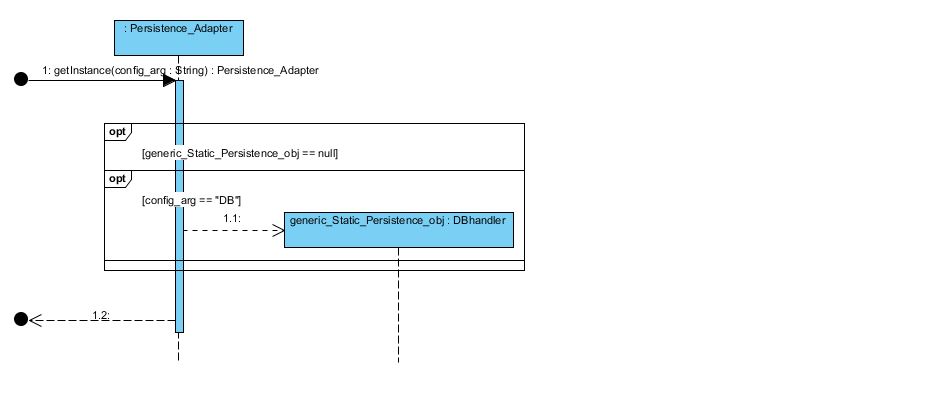
Post-conditions:

SEQUENCE DIAGRAMS

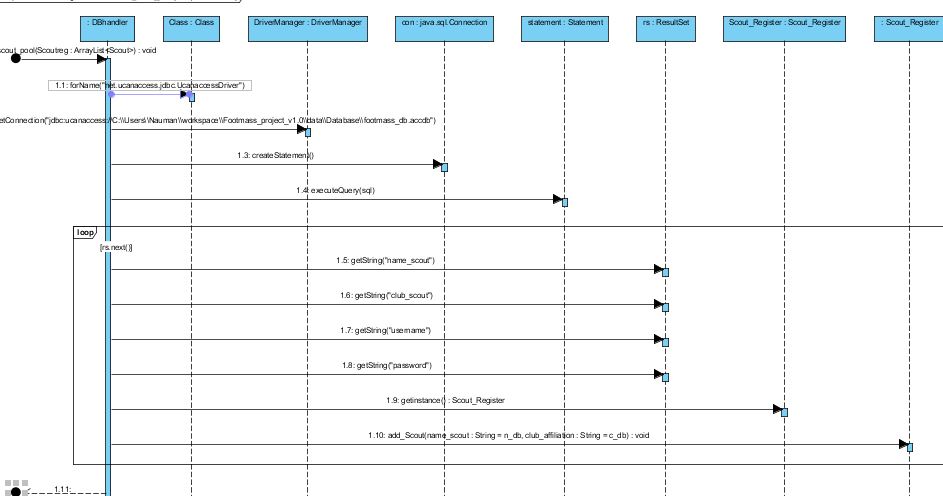
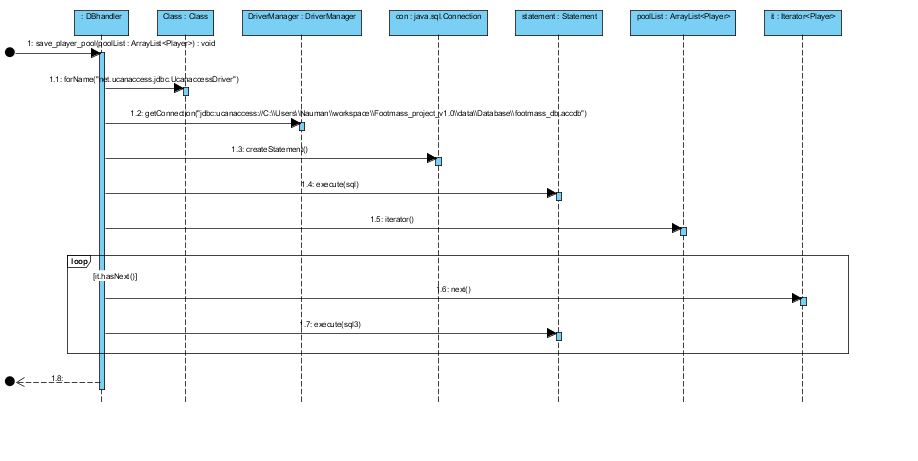
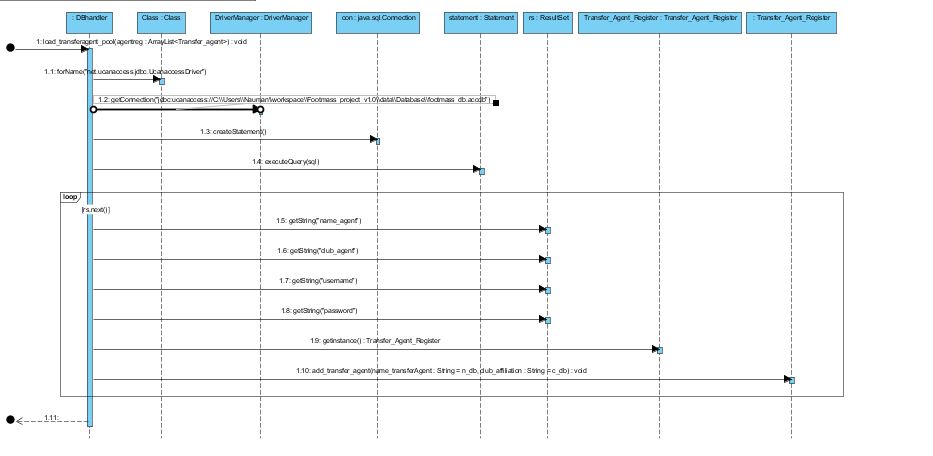
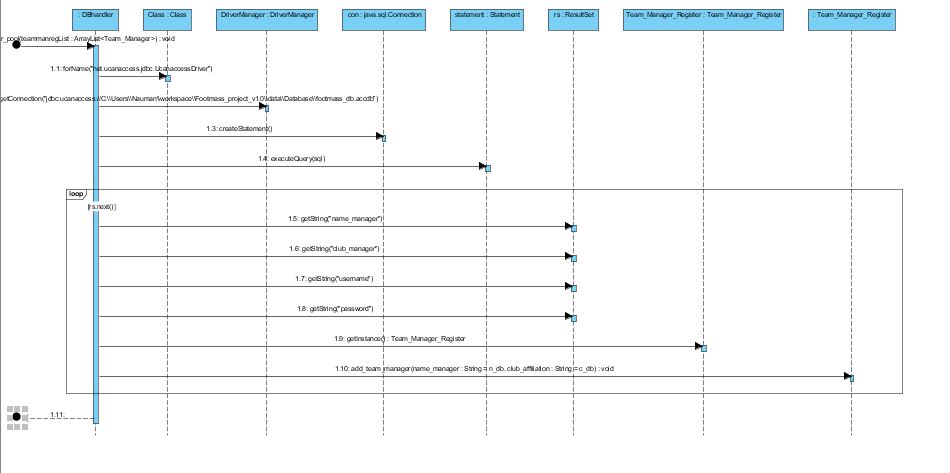
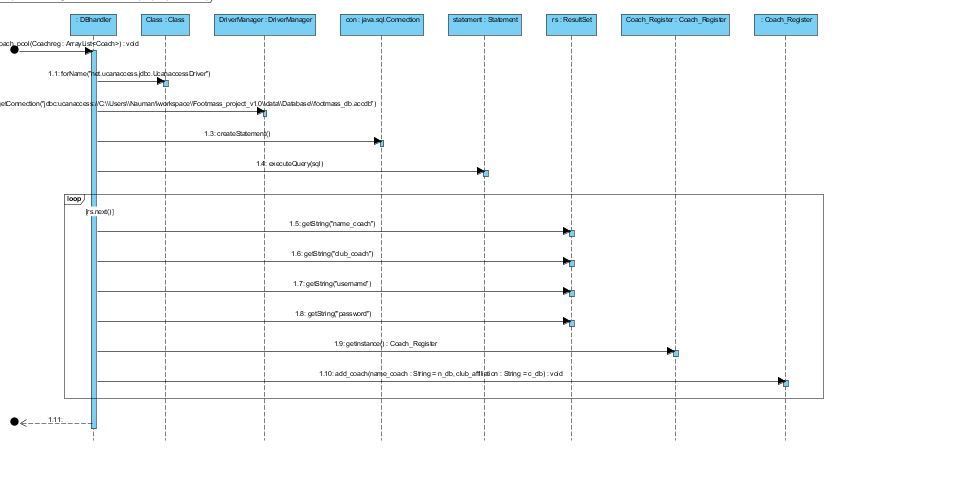
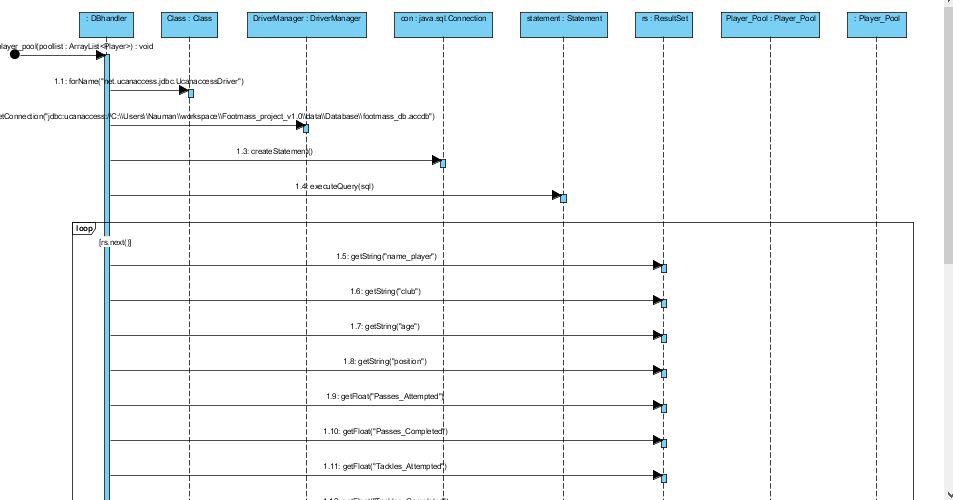
FACTORY



Persistence Adapter and Sub Classes.



DBhandler



# ACTORS

Coach







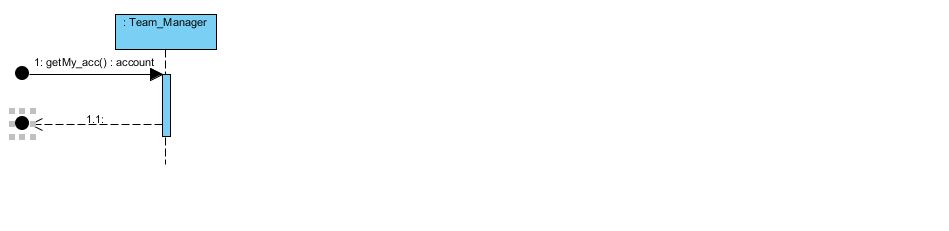
Footmass Admin



Match analyst



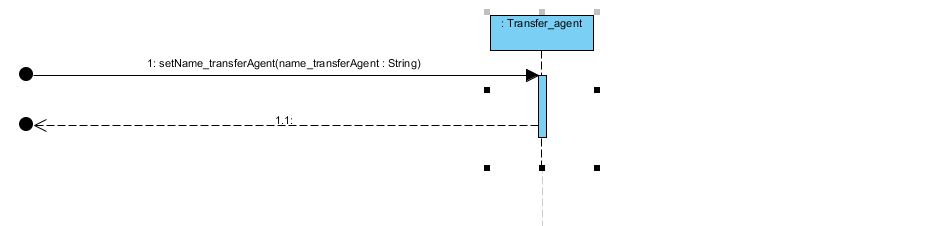
Team manager



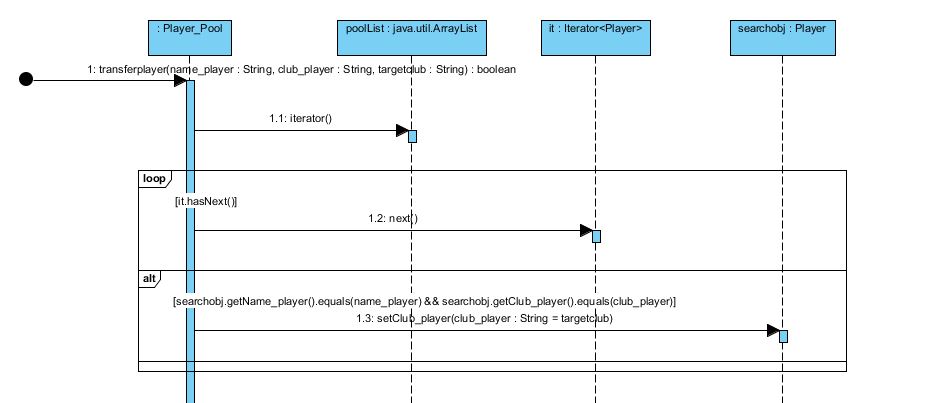
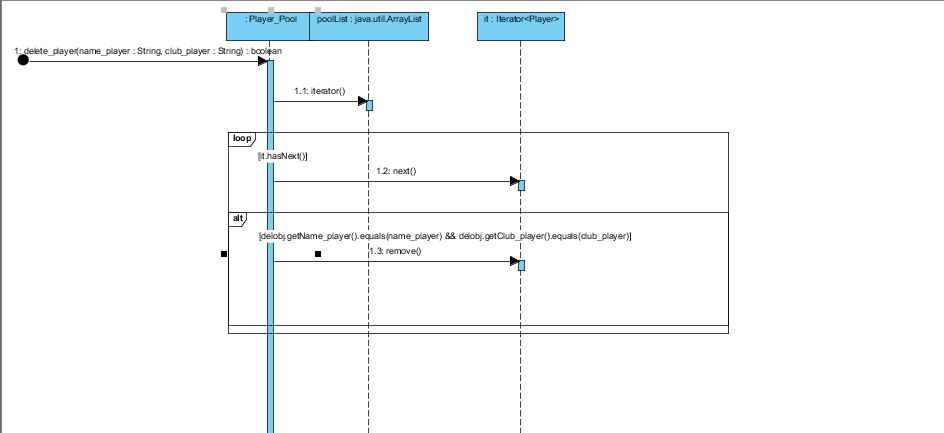
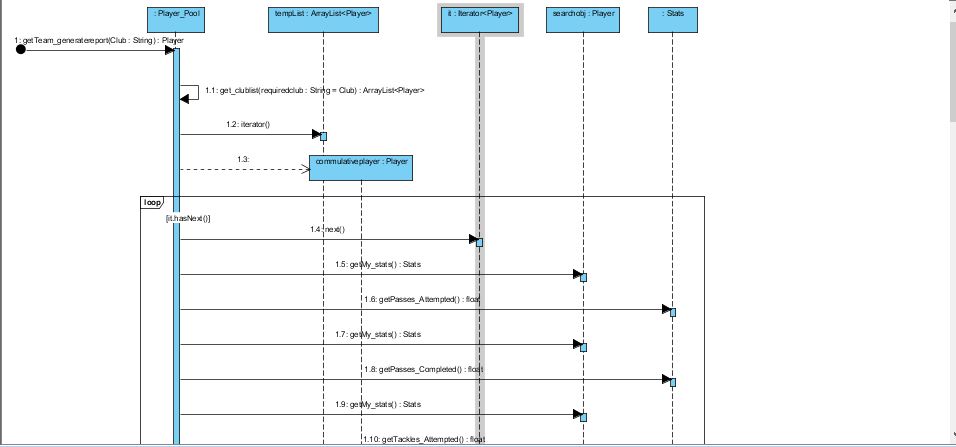
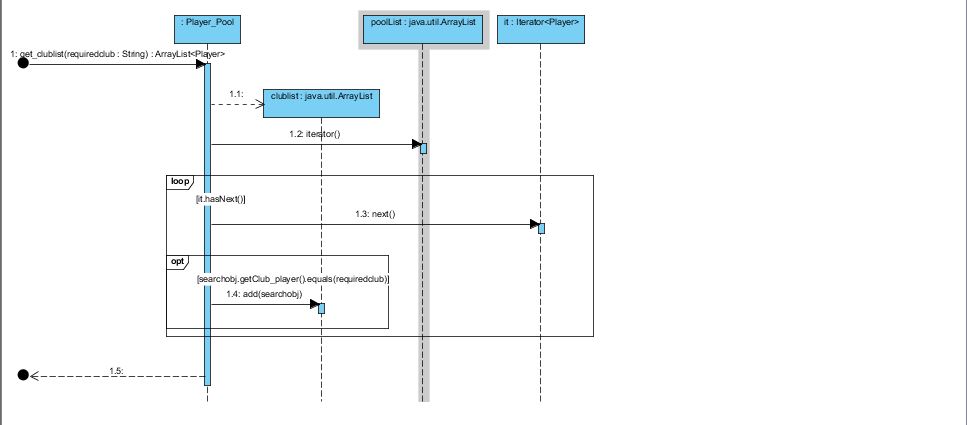
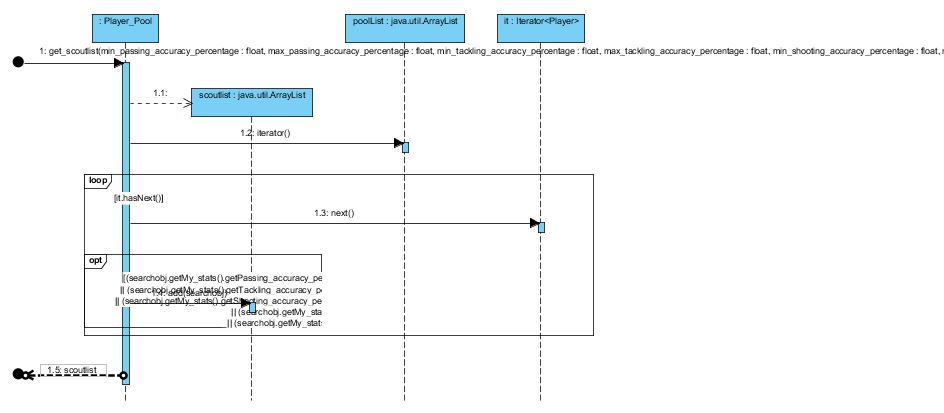
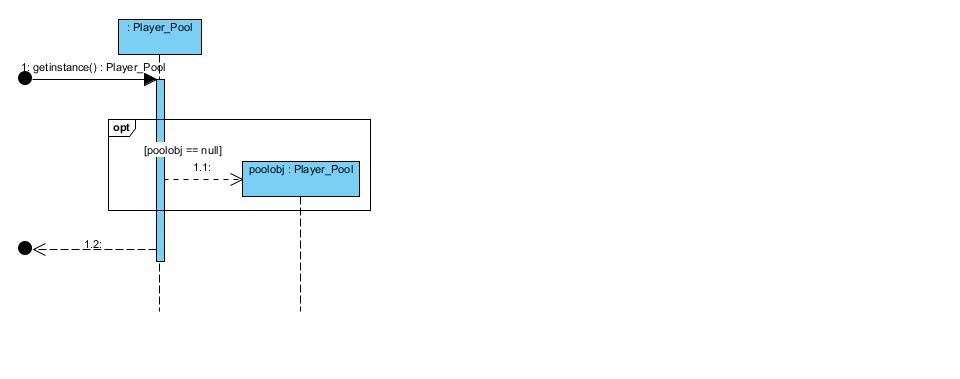
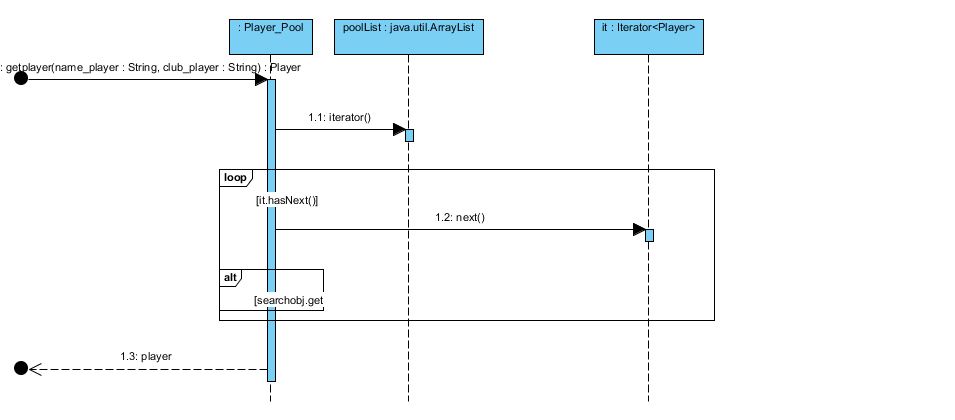
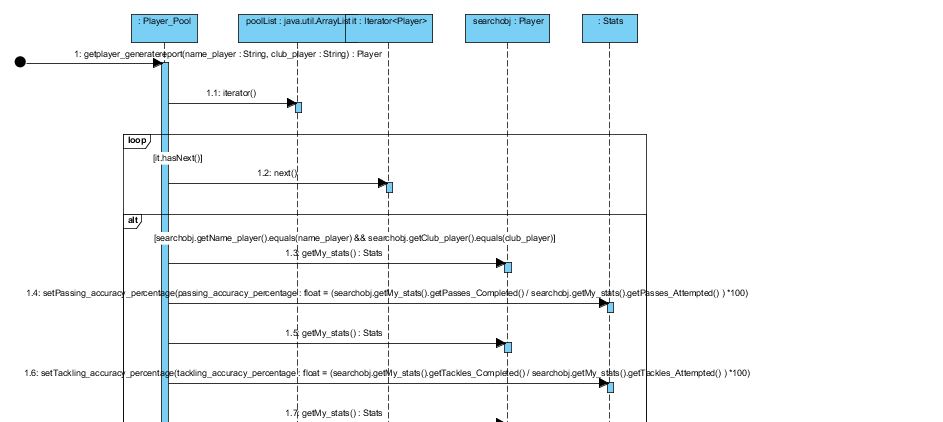
Scout

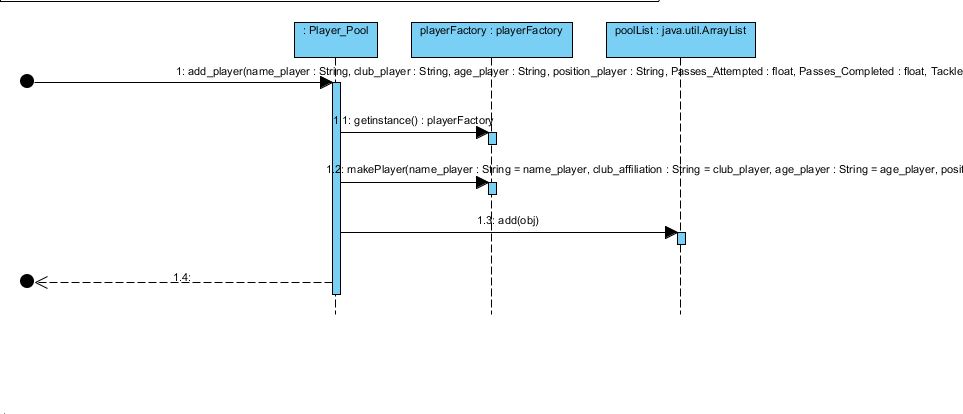


Transfer Agent

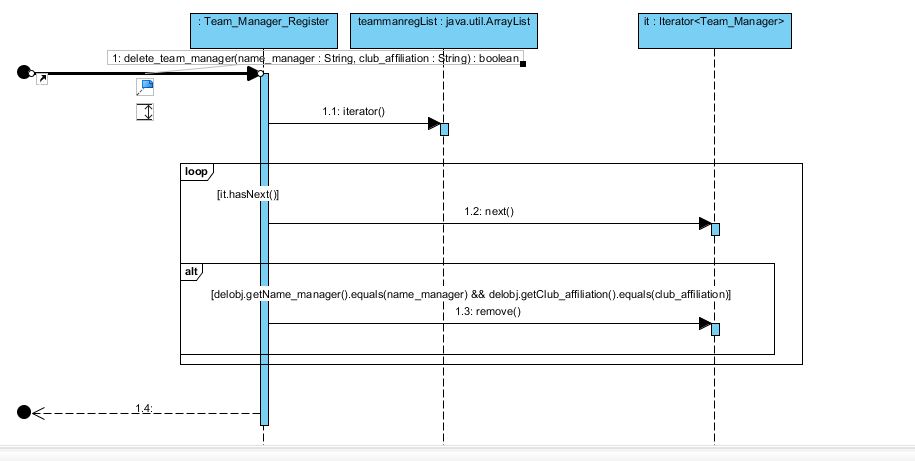
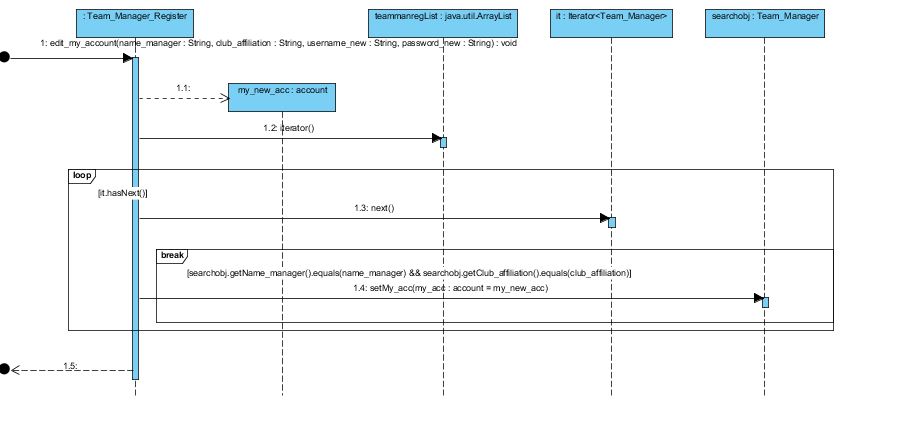
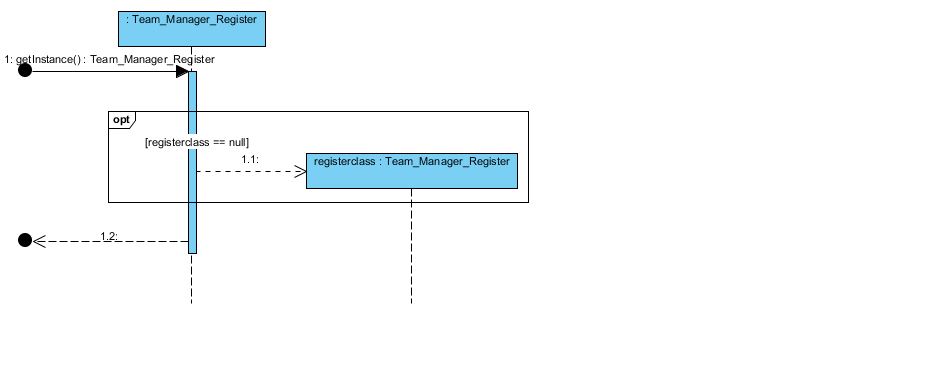
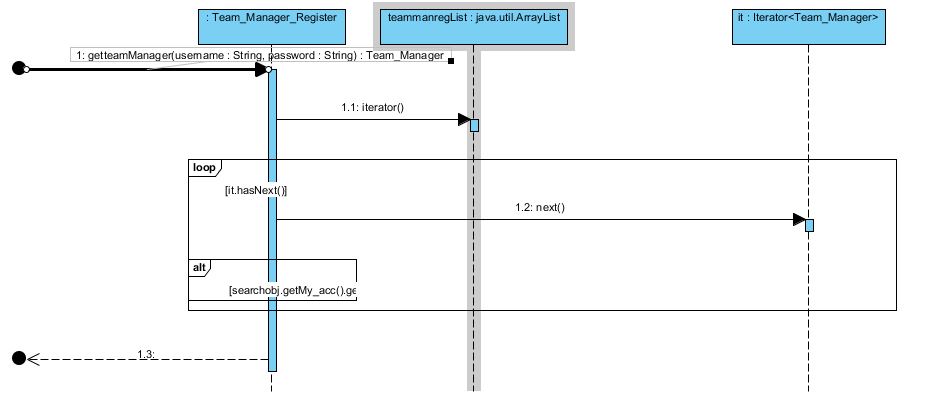
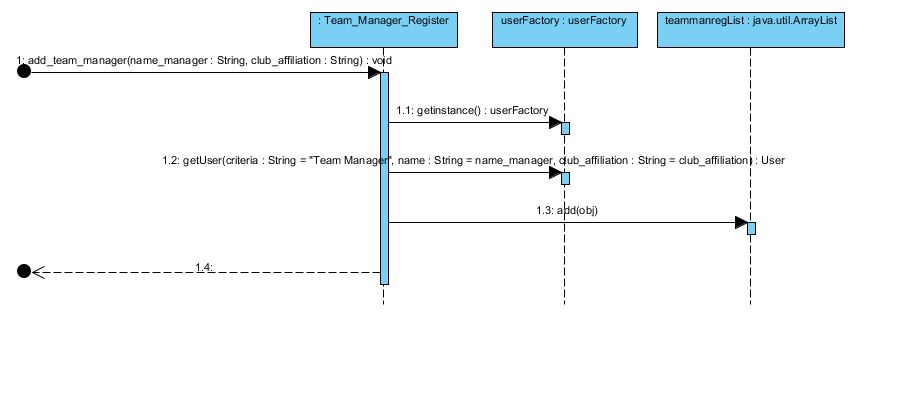
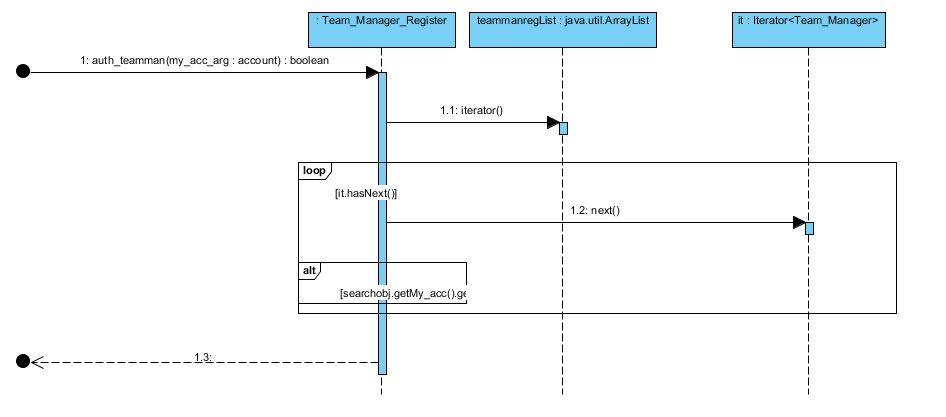


# FOOTY Package

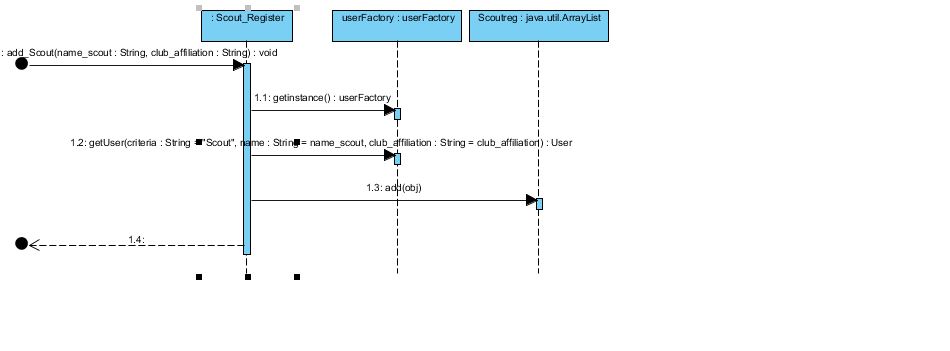
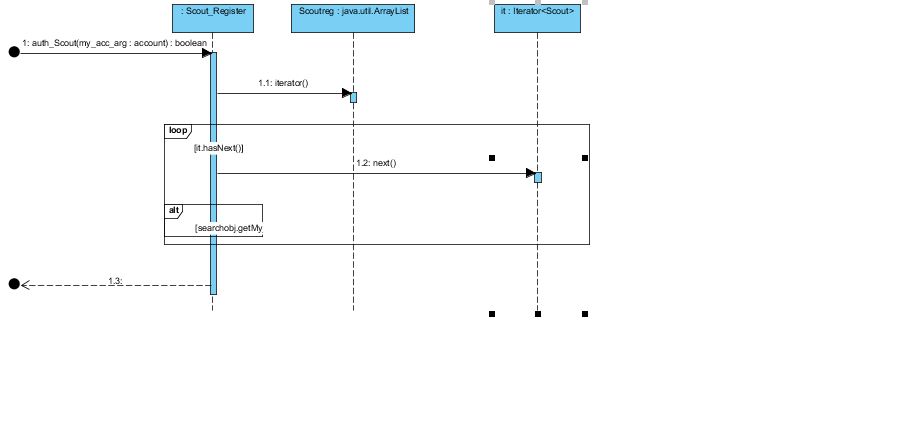
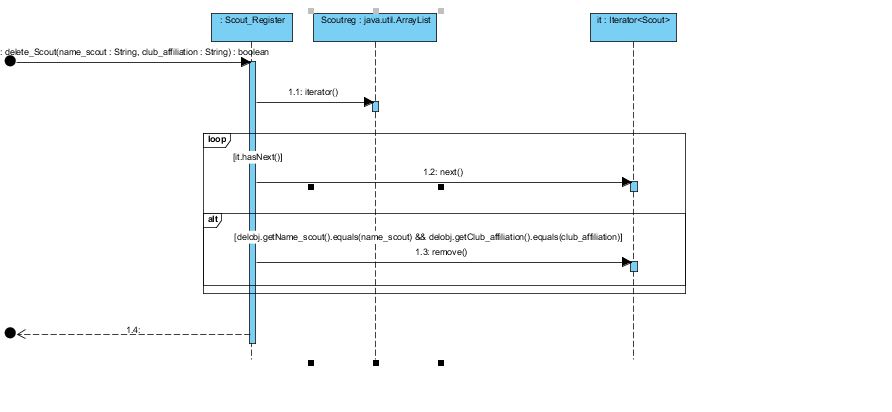
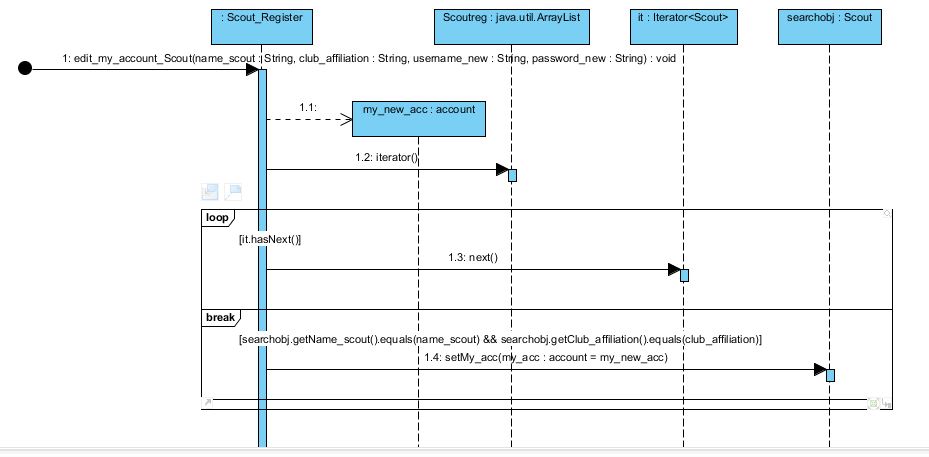
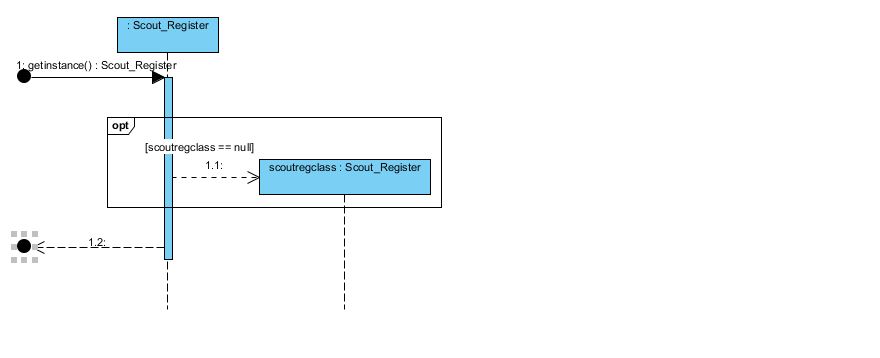
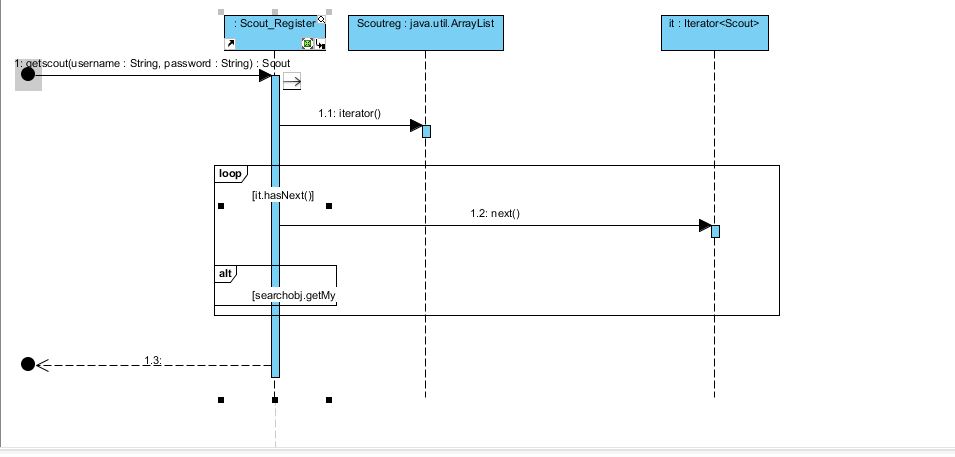
­



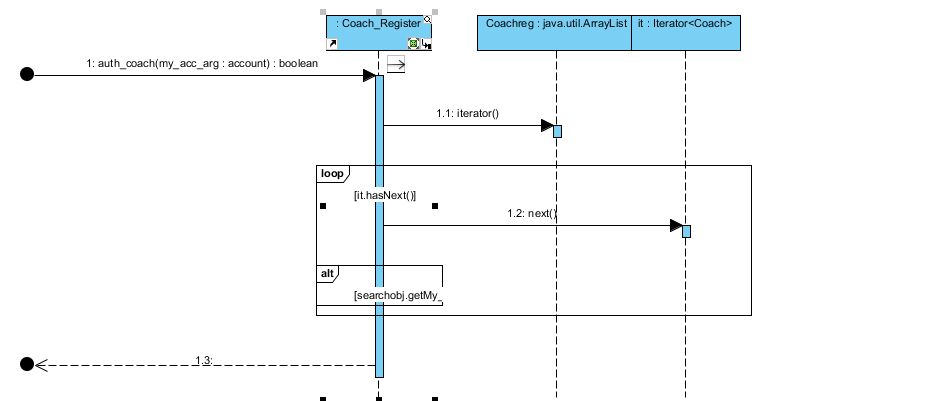
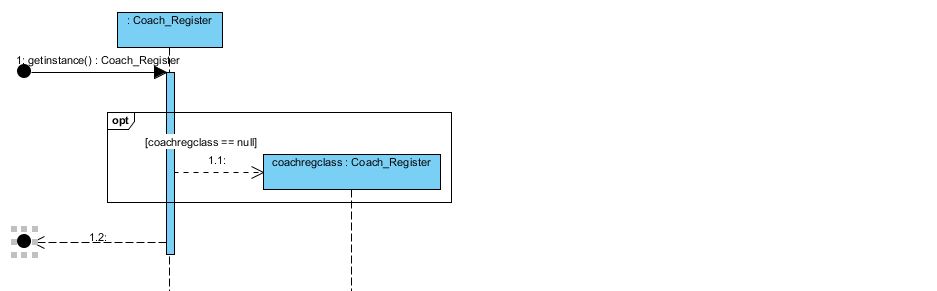
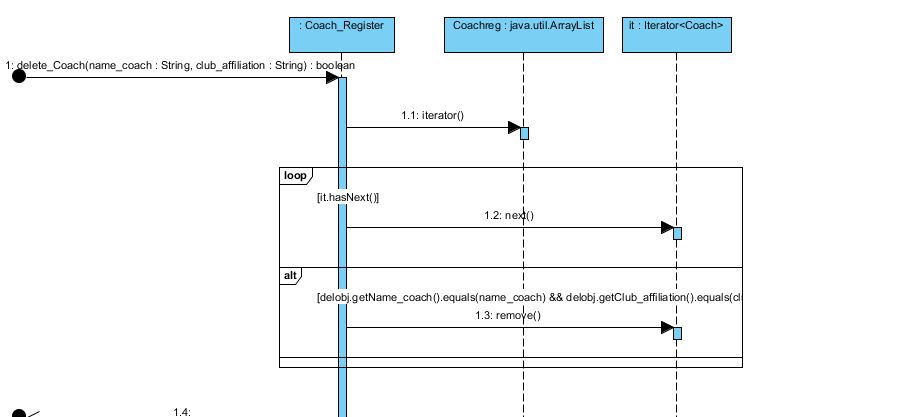
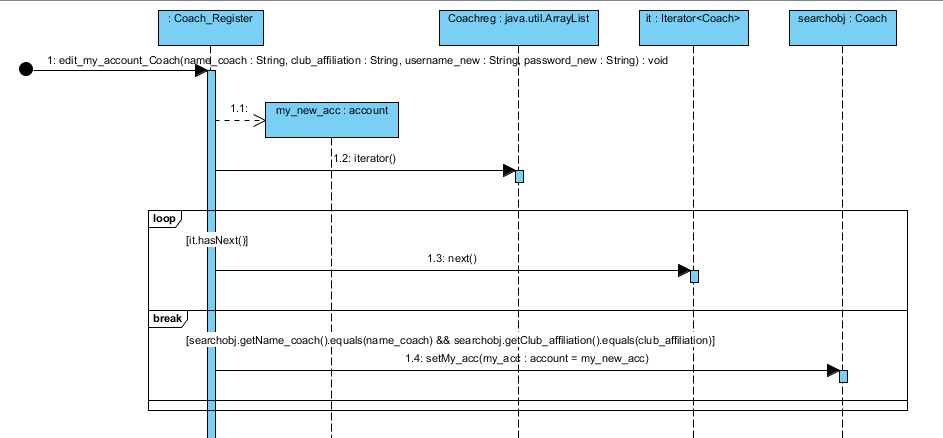
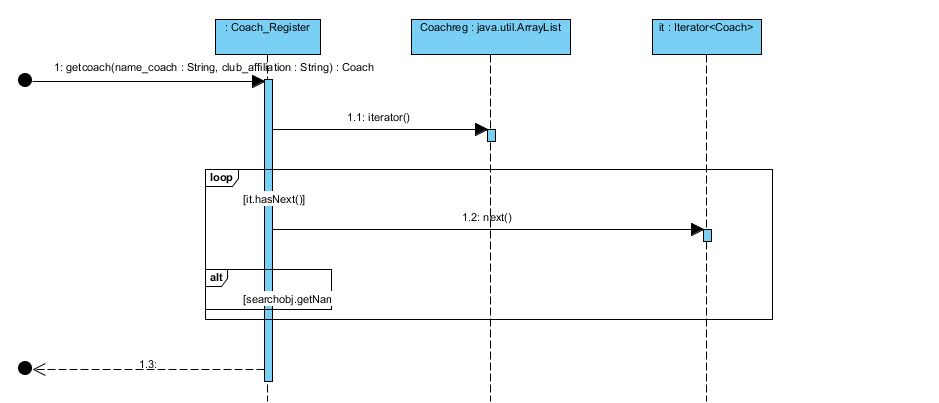
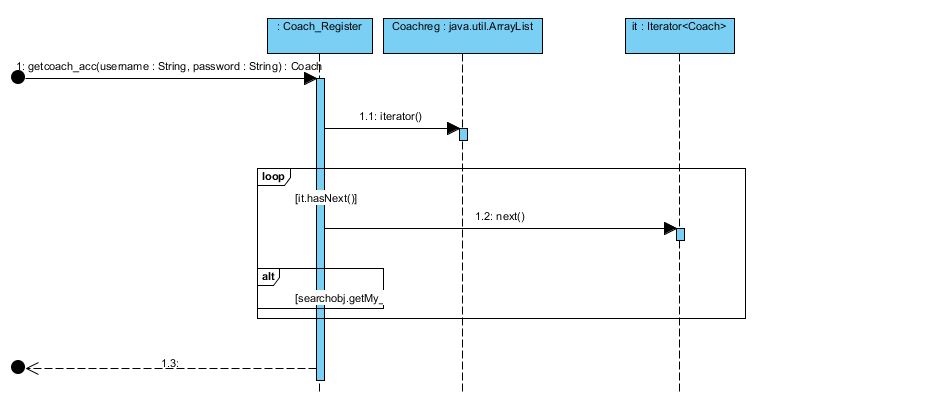
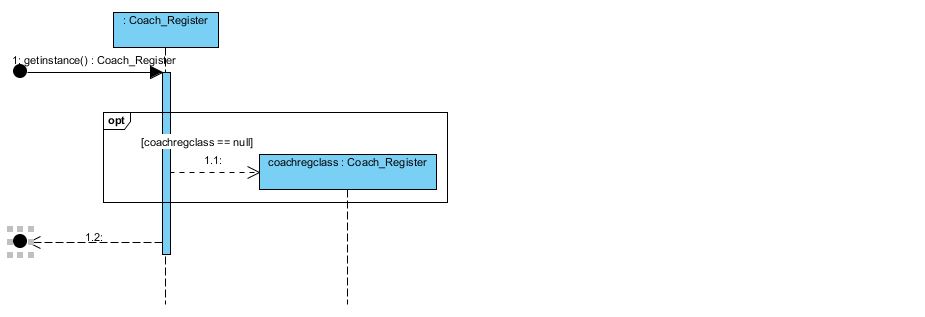
Team Manager Register



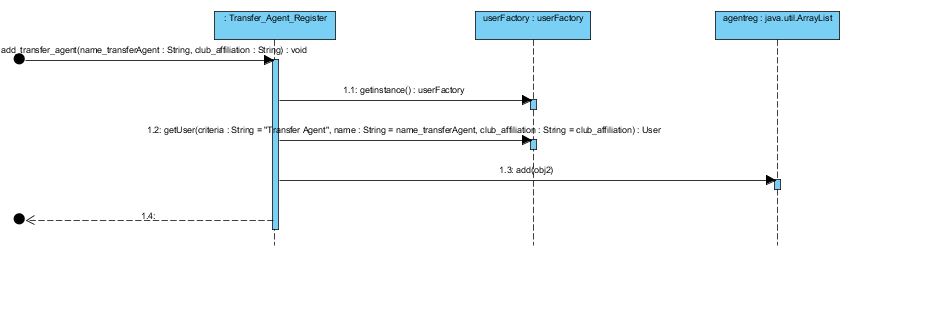
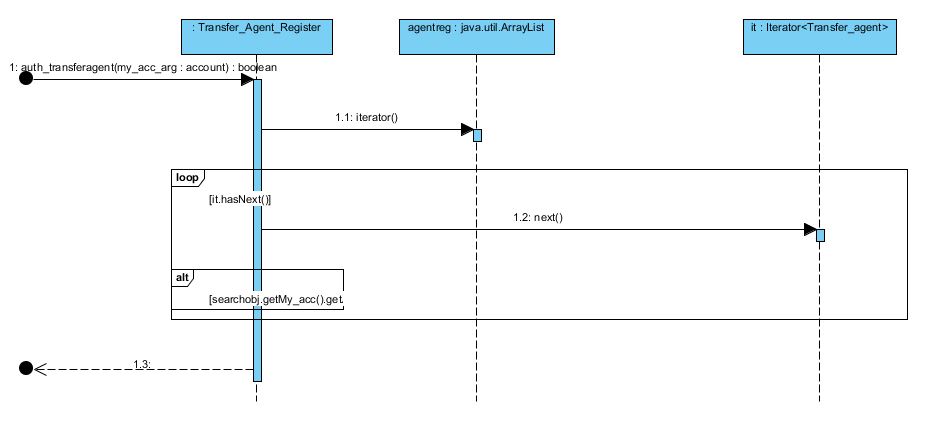
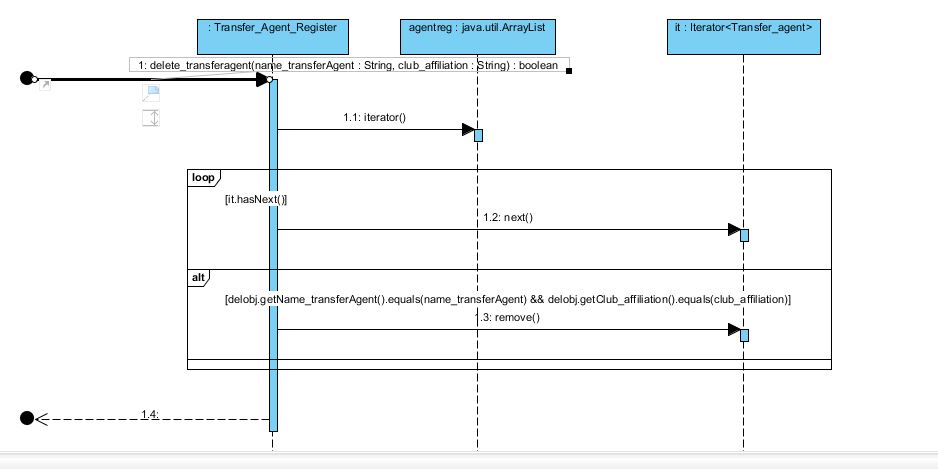
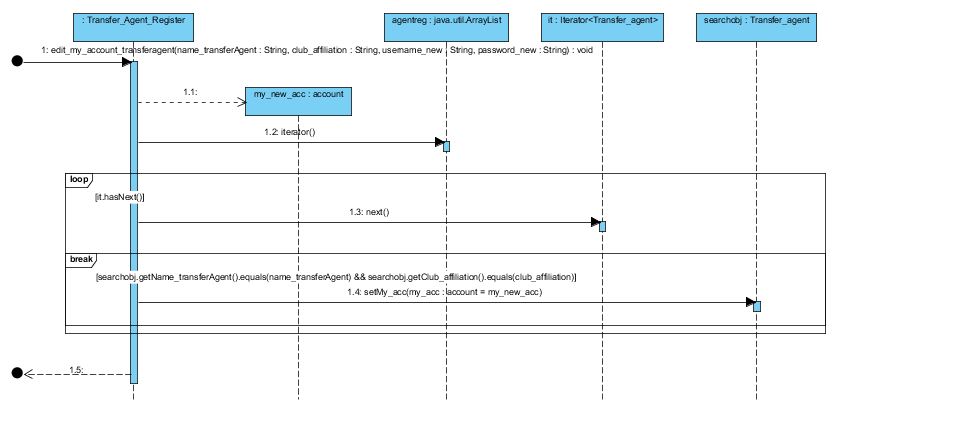
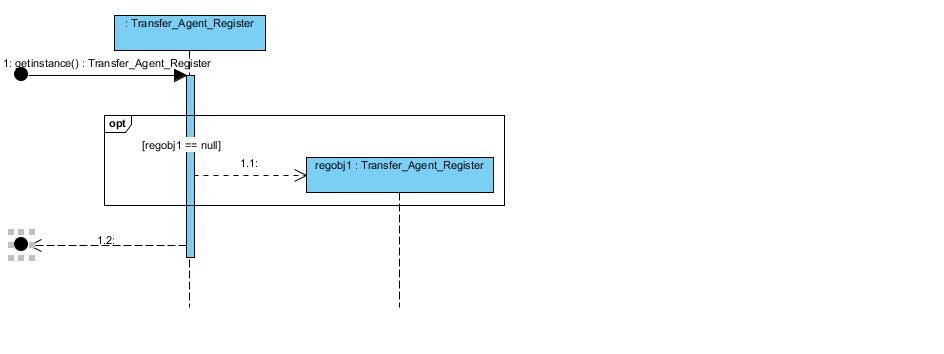
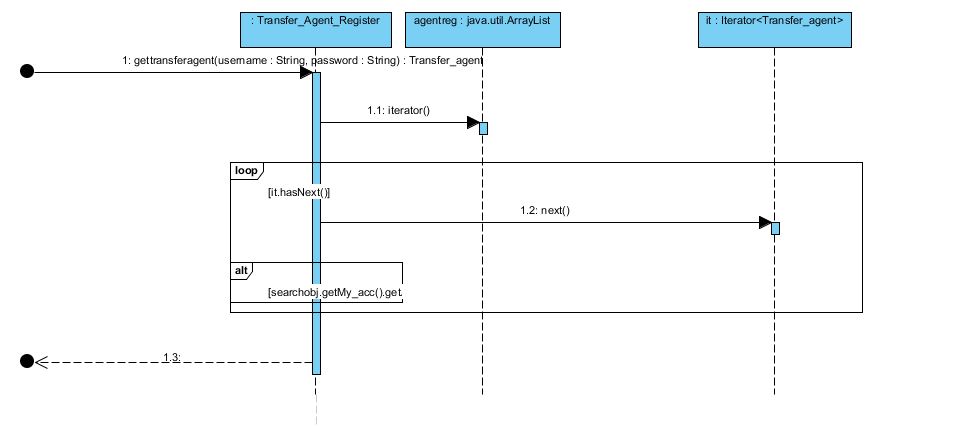
SCOUT REGISTER



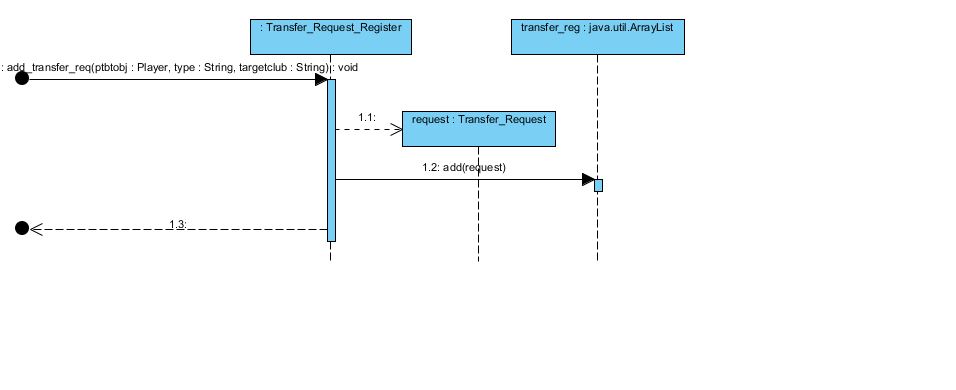
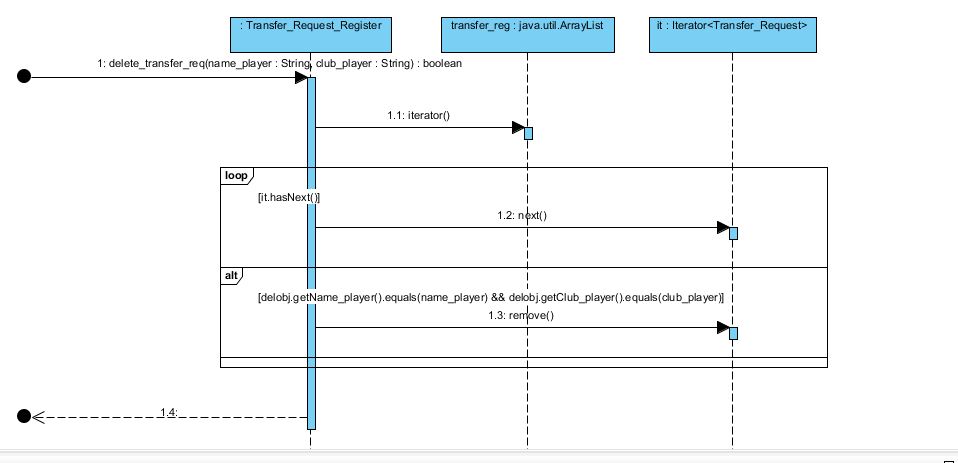
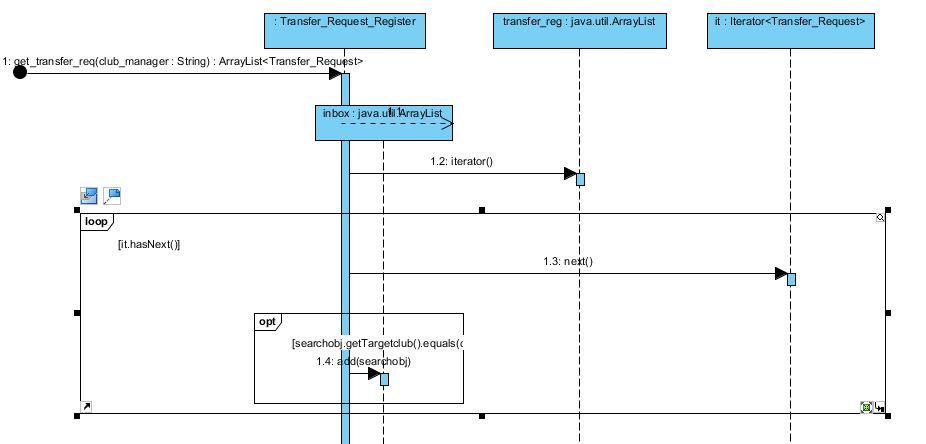
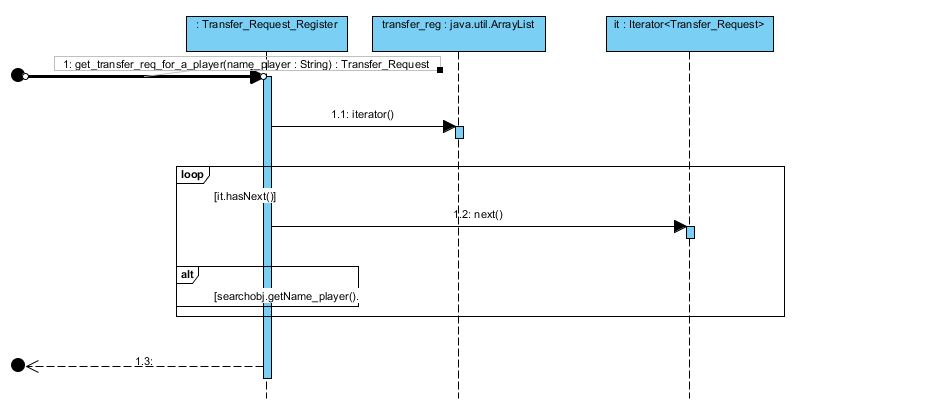
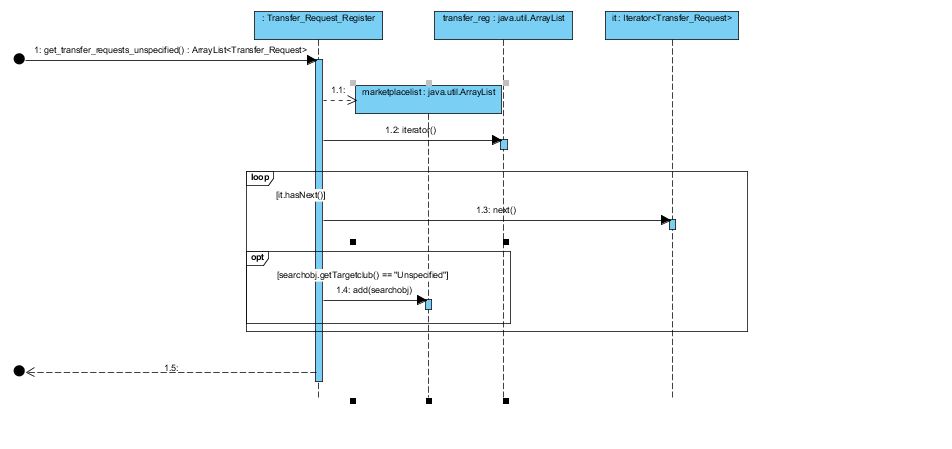
COACH REGISTER



Transfer Agent Register



Transfer Request Register



ALL EXPANDED USE CASES

## Expanded Use case:

|  |  |
| --- | --- |
| Title: View Squad Report. |  |

|  |  |
| --- | --- |
| Primary Actors | Team Manager, Coach |
| Purpose | Reports will be generated depending upon the characteristics the user entered. |
| Description | The system will ask initially what kind of report you want to generate. User will choose between the team report and individual report. Once selected the user will have to insert the characteristics on the basis of which reports will be generated. Like for individual report user has to enter the name of the player and his reports will be generated keeping track of all of his stats during the previous matches. |
| Type | Primary and essential. |
| Cross references | Add Performance Data. |
| Preconditions | Team Manager/Coach logs into the FOOTMASS system and choses the View Squad Reports.  Information can be only drawn from this use case once the database manager/match analyst have completed the Add Performance Data use case. |
| Post conditions | The Team Manager/Coach views all the squad reports. |
| Frequency of use | http://requirements.seilevel.com/blog/uploaded_images/Frquency-of-Use-by-Users-799497.JPG  This use case lies in the (D) section. Its possible users are the Coach and the Team Manager but the frequency of use is very high because the reports are required after every single match to keep in view the player performances and the team’s overall performance. |
| Priority | High. |

|  |  |
| --- | --- |
| Success Scenario |  |
| Actor’s Action | System Response |
| 1. Team Manager/Coach chooses the view squad reports. |  |
|  | 1. System will ask the user (Manager/coach) about the type of report he wants to generate.  * Individual Report * Team Report |
| 1. The user selects Individual Report. |  |
|  | 1. System will then ask about the name of the player of which reports are to be generated. |
| 1. User will enter the Player Name. |  |
|  | System Provides reports in the form of:   1. For Individual reports the system will present the reports containing the following characteristics:   1) Ball Control  2) Crossing  3) Dribbling  4) Finishing  5) Free Kick Accuracy  6) Heading Accuracy  7) Long Pass  8) Short Pass  9) Marking  10) Shot Power |
| 8) Team Manager/Coach leaves after viewing the desired reports. |  |

|  |
| --- |
| Extensions |
| 1. Team Manager/Coach chooses the view squad reports.   2. System will ask the user (Manager/coach) about the type of report he wants to generate.   * Individual Report. * Team Report.  1. The user selects Team Report. 2. For Team report the Team Manager/Coach will enter the club name for which the reports are to be generated.   5. For Team report the system will present the  reports containing the following:  characteristics  1) Goals  2) Shots  3) Yellow Cards  4) Red Cards  5) Corners  6) Fouls  7) Off sides  8) Throw ins  9) Aerial Duo’s Success  10) Possession  11) Pass Success  12) Average Rating |
| 1. For individual reports sometimes maybe no reports are generated because the user has entered the name incorrect or he doesn’t exist. |
| 1. The error message is generated “Player not found”. |
| 9. User will either input the name again or back out of this use case. |
| 10. Sometimes the system will generate an error message “No reports found” because the  Database manager/match analyst has not updated the database using Add Performance Data. |
| 1. In such a case user will either back out of this use case or try again after the database has been updated. |

## Expanded Use case:

|  |  |
| --- | --- |
| Title: Update Player/Club and Manager. |  |

|  |  |
| --- | --- |
| Primary Actors | Database/FootMass Administrator |
| Purpose | To update the player pool from which all the information is then used. |
| Description | The player pool and Manager register containing the information of all the Players/Managers is updated. Keeping in view all the ongoing transfers the Players/Manager information is updated by the FootMass Administrator. |
| Type | Primary and essential. |
| Cross references | No Cross references. |
| Preconditions | Database Administrator logs into the FOOTMASS system and chooses the Update Player/Teams and Manager. |
| Post conditions | The Database Administrator updates the information of the player he/she wants. |
| Frequency of use | http://requirements.seilevel.com/blog/uploaded_images/Frquency-of-Use-by-Users-799497.JPG  This use case can lie in any section. Its possible users is Database Administrator and it’s frequency of use is highly invariant |
| Priority | High. |

|  |  |
| --- | --- |
| Success Scenario |  |
| Actor’s Action | System Response |
| 1. Footmass Administrator chooses the Update Player/Teams and Manager. |  |
|  | 1. System will ask whether the Footmass Admin either wishes to  * Add Player/Manager/Club. * Edit/delete existing Player/Manager info. |
| 1. The Footmass admin will select Add Player as main success scenario. |  |
|  | 1. System will then ask about all the add player information. |
| 1. Footmass admin will enter the add Player information. |  |
| 6) Footmass admin leaves after  adding the information of the  Player. |  |

|  |
| --- |
| Extensions |
| 1. In case the Footmass administrator selects Add Manager/Club the System will ask the admin to enter all of the information of the Manager. |
| 1. The Footmass administrator will enter the required Manager information. |
| 1. Footmass administrator leaves after adding the information of the Manager. |
| 1. In case the Footmass administrator selects Edit Manager/Player the System will ask the admin to choose the information of the Manager/Player he wants to edit. |
| 1. The Footmass admin will choose the information he wants to edit of the Manager/Player. |
| 1. The Footmass admin will now enter the new information of the Manager/Player to update. |
| 1. Footmass administrator leaves after updating the information of the Manager/Player. |
| 1. In case the Footmass administrator selects Delete Manager/Player/Club the System will ask the admin to enter the name of the Player/Manager. |
| 1. The Footmass administrator will enter the name of the Player/Manager/Club. |
| 1. If found System will tell that the desired Player/Manager/Club is deleted. |
| 1. Footmass administrator leaves after deleting the desired Player/Manager/Club. |
| 1. If the Footmass administrator enter the wrong name of the Player/Manager/Club in case of edit/delete Manager/Player/Club. |
| 1. The error message is generated “Player/Manager/Club not found”. |
| 14. Footmass administrator will either input the name again or back out of this use case. |

## Expanded Use case:

|  |  |
| --- | --- |
| Title: Add Performance Data. |  |

|  |  |
| --- | --- |
| Primary Actors | Match Analyst. |
| Purpose | All the information is about the performance of the players and team collectively is added to FootMass Database. |
| Description | The system will ask one by one about the information of Players performances individually. System will also enquire about the Team performance. This information is stored in the database along with it the opponent team’s name and date of the fixture. |
| Type | Primary and essential. |
| Cross references | No cross references. |
| Preconditions | Team Manager/Coach logs into the FOOTMASS system and choses the Add Performance Data. |
| Post conditions | The Team Manager/Coach add all the match performance data. |
| Frequency of use | http://requirements.seilevel.com/blog/uploaded_images/Frquency-of-Use-by-Users-799497.JPG  This use case lies in the (D) section. Its possible user is only the match analyst but the frequency of use is very high because the reports are required after every single match to keep in view the player performances and the team’s overall performance so the match analyst has to update the database with the record of all |
| Priority | High. |

|  |  |
| --- | --- |
| Success Scenario |  |
| Actor’s Action | System Response |
| 1. Match Analyst chooses the add performance data. |  |
|  | 1. System will then ask about the name of the player or kit number whose match performance data is to be added. |
| 1. User will enter the Player Name or kit number. |  |
|  | 1. The system will ask the user to enter the following   Performance data measure:   1. Fouls Committed 2. Passes Attempted 3. Passes Completed 4. Red Cards Awarded 5. Yellow Cards Awarded 6. Saves 7. Shots Attempted 8. Shots on Target 9. Tackles Attempted 10. Tackles Completed |
| 8) Match Analyst leaves after adding the performance information. |  |

|  |
| --- |
| Extensions |
| 1. Sometimes the match analyst enters the name or kit number of the player incorrect. |
| 1. The error message is generated “Player not found”. |
| 4. Match Analyst will either input the name or kit number again or back out of this use case. |
| 1. If the user enter a wrong input in the performance data of the team or individual error message will be generated “Invalid parameter given in the input” and user will have to enter the data again in this case. |

## Expanded Use case:

|  |  |
| --- | --- |
| Title: Sell a Player |  |

|  |  |
| --- | --- |
| Primary Actors | Transfer Agent. |
| Purpose | To change player club and transfer him to new club. |
| Description | When the Manager places the transfer request, the transfer agent will process the transfer request. The transfer request will only be proceeded when both club agents have reached an agreement. |
| Type | Primary and essential. |
| Cross references | UC: Place transfer request |
| Preconditions | The manager must have placed the transfer request  UC: Place transfer request.  The player should exist in the player pool, and in its club |
| Post conditions | The selected player has been successfully transferred to the new club. |
| Frequency of use | http://requirements.seilevel.com/blog/uploaded_images/Frquency-of-Use-by-Users-799497.JPG  This use case lies in the (C) section. user is only the Transfer agent and the frequency of use is low because the transfer are open in a transfer window twice a year |
| Priority | High. |

|  |  |
| --- | --- |
| Success Scenario |  |
| Actor’s Action | System Response |
| 1. Transfer agent chooses to sell a player |  |
|  | 1. System will ask the Transfer agent to choose the transfer request from the list of transfer requests placed. |
| 1. The Transfer Agent will enter the transfer request he wants to proceed. |  |
|  | 1. System will ask the transfer agent for authentication to the club that appointed him. |
| 1. The transfer that was placed is initiated only if the transfer agent is authenticated. |  |
|  | 1. System will check if the outgoing request is directed to a club the request will be forwarded to that club for conformation. |
| 1. Transfer agent leaves after initiating the transfer request. |  |

|  |
| --- |
| Extensions |
| 1. If the transfer agent is not authenticated the transfer request will not proceed. |
| 1. System will check if the outgoing request is not directed to a club the request will be forwarded to the market place for everyone to view. |

## Expanded Use case:

|  |  |
| --- | --- |
| Title: Sign a player |  |

|  |  |
| --- | --- |
| Primary Actors | Transfer Agent. |
| Purpose | To change player club and transfer him to the club. |
| Description | When the Manager places the transfer request, the transfer agent will process the transfer request. The transfer request will only be proceeded when both club agents have reached an agreement. |
| Type | Primary and essential. |
| Cross references | UC: Place transfer request |
| Preconditions | The manager must have placed the transfer request  UC: Place transfer request.  The player should exist in the player pool, and in its club. |
| Post conditions | The selected player has been successfully signed for the club. |
| Frequency of use | http://requirements.seilevel.com/blog/uploaded_images/Frquency-of-Use-by-Users-799497.JPG  This use case lies in the (C) section. user is only the Transfer agent and the frequency of use is low because the transfer are open in a transfer window twice a year |
| Priority | High. |

|  |  |
| --- | --- |
| Success Scenario |  |
| Actor’s Action | System Response |
| 1. Transfer agent chooses to sign a player from the transfer requests placed by the managers. |  |
|  | 1. System will ask the Transfer agent to choose the transfer request from the list. |
| 1. The Transfer Agent chooses the transfer request from the transfer request list. |  |
|  | 1. System will ask the transfer agent for authentication to the club that appointed him. |
| 1. Transfer Agent enters confidential code given by the manager. |  |
|  | 6. The transfer that was placed is initiated only if  the transfer agent is authenticated. |
|  | 1. System will check if the sign a player request is directed to a club the request will be forwarded to that club’s Transfer agent for conformation. |
| 1. Transfer agent leaves after initiating the transfer request. |  |

|  |
| --- |
| Extensions |
| 1. If the transfer agent is not authenticated the transfer request will not proceed. |
| 1. System will check if the signing request is not directed to a club the request will be forwarded to the market place for everyone to view. |

## Expanded Use case:

|  |  |
| --- | --- |
| Title: Place Transfer Request. |  |

|  |  |
| --- | --- |
| Primary Actors | Team Manager. |
| Purpose | To inform the transfer agent about the player(s) a club wishes to sign. |
| Description | The Team Manager wishes to sign/sell a player the manger places the Transfer Request. |
| Type | Primary and essential. |
| Cross references |  |
| Preconditions | In case the Team Manager wants to sign a player the player he wishes to sign must exist in the selling club.  In case the Team Manager wants to sell a player the player must exist in the club the manager is appointed to. |
| Post conditions | The Transfer request is been placed in the Transfer register and is ready to be processed by a transfer agent. |
| Frequency of use | http://requirements.seilevel.com/blog/uploaded_images/Frquency-of-Use-by-Users-799497.JPG  It does not belong to any quadrant of the frequency diagram because of the high variance for example 10 players could be transfer in one transfer window or no player could be transfer in 10 transfer windows. |
| Priority | High. |

|  |  |
| --- | --- |
| Success Scenario |  |
| Actor’s Action | System Response |
| 1. Team Manager selects the place transfer request. |  |
|  | * System will ask whether he wants to * Sign a player * Sell a player |
| 1. The user will select sign a player. |  |
|  | 1. System will then ask about the name of the player and selling club. |
| 1. User will enter the Player Name or selling club name. |  |
| 5) Team Manager leaves after placing  the transfer request. |  |

|  |
| --- |
| Extensions |
| 1. If The user will select sign a player. |
| 1. System will then ask about the name of the player and destination club. |
| 1. User will enter the Player Name or selling club name. |
| 4. Team Manager leaves after placing the transfer request. |
| 1. If a request for a player has been already placed the system will not allow to again place the transfer. |

|  |  |
| --- | --- |
|  |  |

## Expanded Use case:

|  |  |
| --- | --- |
| Title: Scout Players. |  |

|  |  |
| --- | --- |
| Primary Actors | Scout |
| Purpose | Reports will be generated depending upon the characteristics the Scout entered. |
| Description | The system will ask initially what kind of characteristics are required to be in a new player, the scout will enter the characteristics, based on closest matches, the system will generate a shortlist of potential transfers from a large pool of player data |
| Type | Primary and essential. |
| Cross references | UC: Add Performance Data. |
| Preconditions | Scout logs into the FOOTMASS system and choses the “Scout for players” option.  Information can be only drawn from this use case once the player pool is in a stable state (a onetime process), and the Footmass admin has completed the “update player/team use case“. |
| Post conditions | The scout views the shortlist of players, and forwards it to the team manager/transfer agent. |
| Frequency of use | http://requirements.seilevel.com/blog/uploaded_images/Frquency-of-Use-by-Users-799497.JPG  This use case lies in the (C) or (D) section. Its possible users are the scouts, but the frequency of use can vary. Typically the scouts will view the reports twice a year in an open transfer window, but there are no restrictions what so ever. |
| Priority | High |

|  |  |
| --- | --- |
| Success Scenario |  |
| Actor’s Action | System Response |
| 1. Scout chooses to scout players |  |
|  | System will ask the Scout about the required characteristics. |
| 1. The scout will enter the characteristics required |  |
|  | System will then match the required characteristics in the player pool. And display the shortlisted potential signings. |
| 1. The Scout chooses to forward the shortlisted players to the manager |  |
|  | The system allows the manager to view the shortlisted player chosen by the scout. |
| 1. The Scout leaves after shortlisting the players for the Team Manager. |  |

|  |
| --- |
| Extensions |
| 1. The scout’s required characteristics don’t match. |
| 1. The error message is generated “Player not found”. |
| 3. Scout enters ambiguous query. |
| 1. The scout is prompted to enter a valid query. |