First Semester Project

Software Technology Engineering

Group 1

Adrian-Cristian Militaru 308842

Adrian Pompierescu 309774

Gabriel Moutinho Tristan 304376

Freja Hansen 308840

Supervisors:

Allan Henriksen

Mona Andersen

Software Technology Engineering

First semester

4th of June 2021

Table of content

List of figures and tables

Abstract

VIA Club needs a management system that can handle the data for its football team and a website that can display their match details. The main purpose of the project is to deliver a system that can handle the data of the VIA Club and to document the different stages of development.

The development was initialized with the phase of analysis. To plan the technical performance of the software, requirements, use case diagrams, use case descriptions and activity diagrams were created. The findings of those details help us to initialize the design phase. A class diagram has been created and helps us to initialize the project in Java. The GUI was created by using the JavaFX library and Scene Builder. The implementation was followed by a test phase before being hand-in to the customer.

The result is a program that will allow the manager to store in binary files all the data. The functionality of the system enables the manager to add, edit and remove data, even export in an XML file functionality that can be used on the website. The project aims to contain all the VIA Club's requirements.

1. Introduction

Bob Oldenuff is the manager of the VellCity Indoor Athletics club, also known as VIAClub, and needs a program to make his work easier and more manageable. The manager has worked with managing the club for the last 31 years by himself. VIAClub is a relatively small club and unlike what the name suggests only has had an outdoor football team for the last 57 years. The club was founded almost 100 years ago by the legendary Robert Sixpack, Roberts's great-grandson is currently one of the club's biggest sponsors.

The manager's main job is to make lists of which players to use in every match, who will start on the pitch, and who will start on the bench, including the name, number, and position of each player. It is difficult to keep track of suspended players as well as injured ones and sometimes, new players are bought, and others are sold. The manager needs to be aware of all these changes.

The manager had a problem a while ago when his hot-headed striker ate his squad formation that was on a piece of paper. Keeping track of all the things on a piece of paper has turned out to be a burden nowadays so he decided to try the digital version of the game plan. Also, the manager would like to improve the online image of the club.

Because of the previously experienced accidents and happenings, VIAClub needs a digital system that makes the manager's job easier and a website to attract new supporters. For more details on the background description, look at Appendix A.

2. Analysis

The customer wants an application that can easily manage the team and prepare it for future matches.

The extracted keywords from the interview have been used to create functional requirements which represent what the system should be able to do and works with the customer's request.

2.1 Functional Requirements

Critical Priority

1. As a manager, I need a system to make matches and upcoming matches.

2. As a manager, I want to be able to write the date of a match and, and who the opponent will be.

3. As a manager, I need a system to store a list of players that I want to include in the squad for the next match.

4. As a manager, I need to be able to make League and Cup matches that suspended or injured players cannot play.

5. As a manager, I need to be able to make Friendly matches where injured players cannot play.

6. As a manager, I need to be able to update and keep track of the player's details.

7. As a manager, I need to keep track of the injured or suspended players.

8. As a manager, I need to be able to add and remove players when they are sold or bought.

9. As a manager, I need to have an overview of the player to have a first name, last name, number, preferred position.

10. As a manager, I need to have an overview of players that are injured and try to also keep track of when and how long they might be suspended (Status).

11. As a manager, I need to be able to edit a player's position after it gets re-trained.

12. As a manager, I need to be able to edit matches if plans change.

13. As a manager, I need to be able to keep track of previous matches to be able to go back and see who has participated in which matches.

14. As a manager, I need to keep track of the number of substitute players for a specific type of match.

15. When an issue occurs, the problem pop-up a message.

High Priority

16. As a manager, I need to send the match details on the website.

17. As a manager, I need a responsive website with more details regarding the matches and the club.

18. As a supporter, I need to keep tracking the match history and the match details on the website.

2.2 Non-Functional Requirements

19. Every update in the system includes writing to a file.

20. The program is in the java language.

21. To have an interface that can be accessed with a mouse and a keyboard.

2.3 Use Case Diagram

The use case diagram has been made based on the customer requirements. This diagram represents the actions in relation to the system.

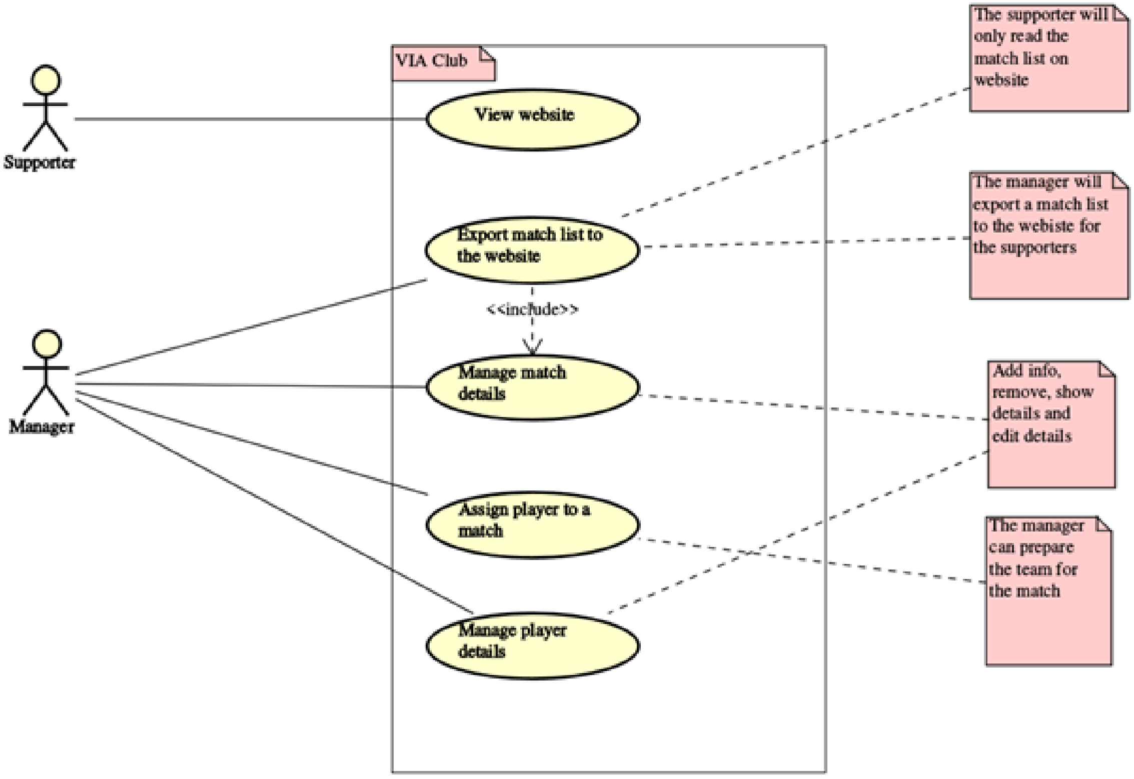

VIA Club
Supporter
Manager
View website
Export match list to the website
<<include>>
Manage match details
Assign player to a match
Manage player details
The supporter will only read the match list on website
The manager will export a match list to the webiste for the supporters
Add info, remove, show details and edit details
The manager can prepare the team for the match

Figure 1. Use case diagram

2.4 Use Case Description to manage player details

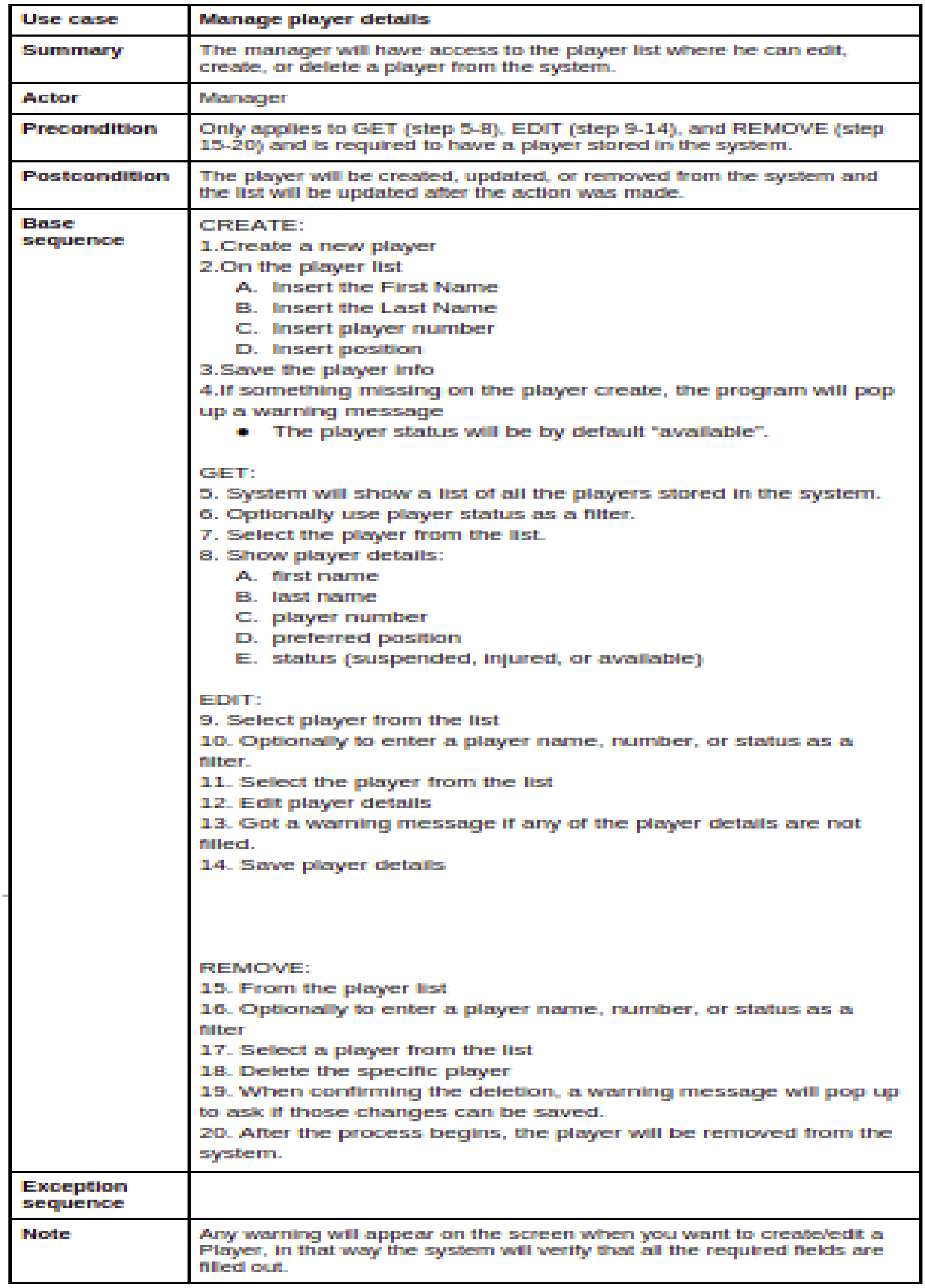
Based on the use case diagram, a use case description has been made for each action - Manage player details, manage match details, assign player to a match, export match list to the website, and view website.

The use case for Manage player details consists of creating, reading player data, editing a player, and removing a player from the system.

Based on the requirements, “manage player details” is an essential use case because it will serve for another use case, such as Assign player to a match.

For adding a new player, the process consists of adding personal information and saving it. For everything else, the data will be displayed when the operation of reading and editing will be called.

The use case of Manage player details is presented in a step-by-step description in the following use case. More details regarding the other use case descriptions will be in Appendix B.



Use case	**Manage player details**
Summary	The manager will have access to the player list where he can edit, create, or delete a player from the system.
Actor	Manager
Precondition	Only applies to GET (step 5-8), EDIT (step 9-14), and REMOVE (step 15-20) and is required to have a player stored in the system.
Postcondition	The player will be created, updated, or removed from the system and the list will be updated after the action was made.
Base sequence	CREATE: 1.Create a new player 2.On the player list A. Insert the First Name B. Insert the Last Name C. Insert player number D. Insert position 3.Save the player info 4.If something missing on the player create, the program will pop up a warning message ● The player status will be by default "available". GET: 5. System will show a list of all the players stored in the system. 6. Optionally use player status as a filter. 7. Select the player from the list. 8. Show player details: A. first name B. last name C. player number D. preferred position E. status (suspended, injured, or available) EDIT: 9. Select player from the list 10. Optionally to enter a player name, number, or status as a filter. 11. Select the player from the list 12. Edit player details 13. Got a warning message if any of the player details are not filled. 14. Save player details REMOVE: 15. From the player list 16. Optionally to enter a player name, number, or status as a filter 17. Select a player from the list 18. Delete the specific player 19. When confirming the deletion, a warning message will pop up to ask if those changes can be saved. 20. After the process begins, the player will be removed from the system.
Exception sequence	
Note	Any warning will appear on the screen when you want to create/edit a Player, in that way the system will verify that all the required fields are filled out.

Figure 2. Use case description for Manage player details

2.5 Activity Diagram for export the match list to the website

The activity diagram shows how the process of exporting a match-list to an XML file is handled. The process will read every detail from the matches list and all the details about every match will be exported in an XML file that can be used on the website.

After the process has been successfully finished a confirmation message will appear and will inform the file has been exported.

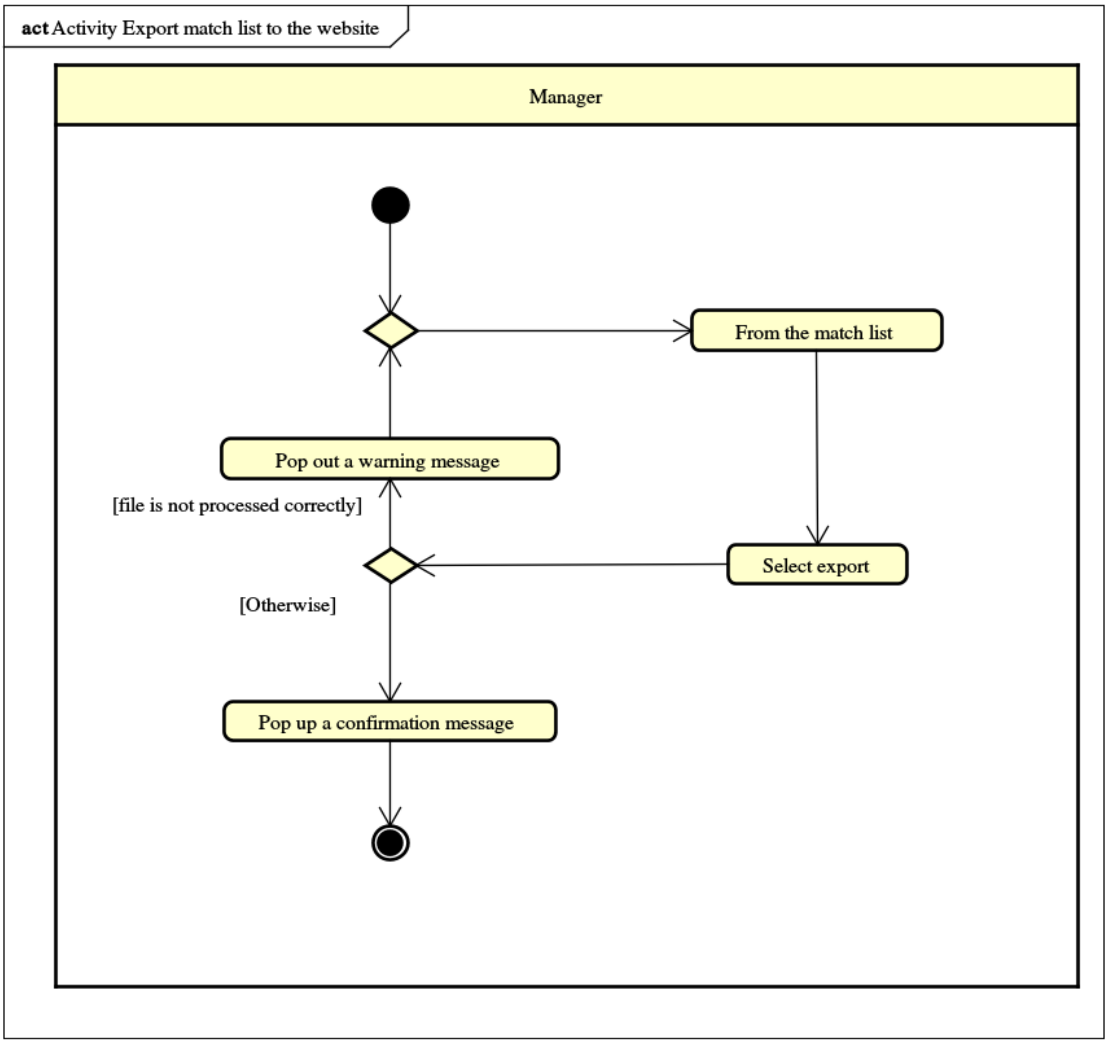

act Activity Export match list to the website
Manager
From the match list
Pop out a warning message
[file is not processed correctly]
Select export
[Otherwise]
Pop up a confirmation message

Figure 4. Activity diagram for Export match list to the website

2.6 Domain Model

A domain model has been created based on the functional requirements representing the relationship between classes. This model was generated based on the previous analysis steps.

The VIAClub application will contain a list of players and a list of matches, and for a match, it will be mandatory to have a date and location where the match will take place. The relation between Match and Player is referring to what player will be assigned to a specific match.

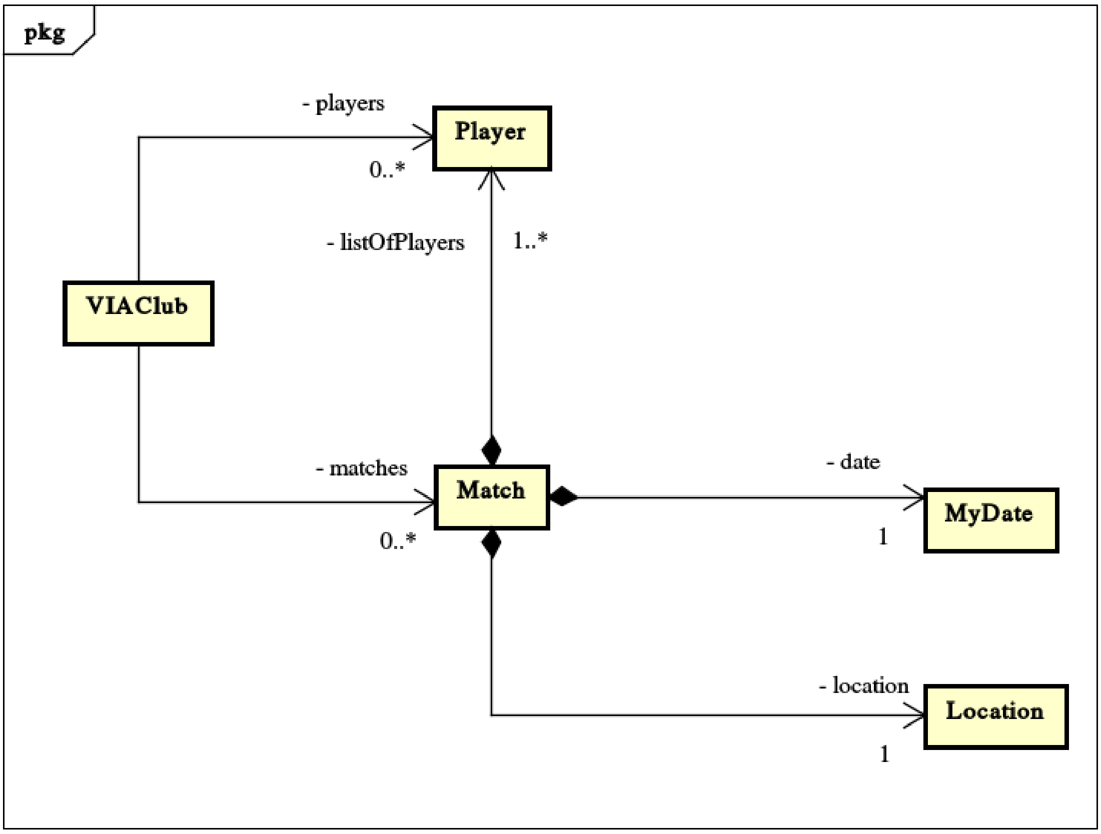


Figure 5. Domain model

3. Design

3.1 Class Diagram

The class diagram was created based upon the Domain Model and serves as the foundation for implementing the source code.

In ***Figure 6.*** We depict the Composition relation between classes. For example, the class “Match” is made up of one or more players, among other data from other classes.

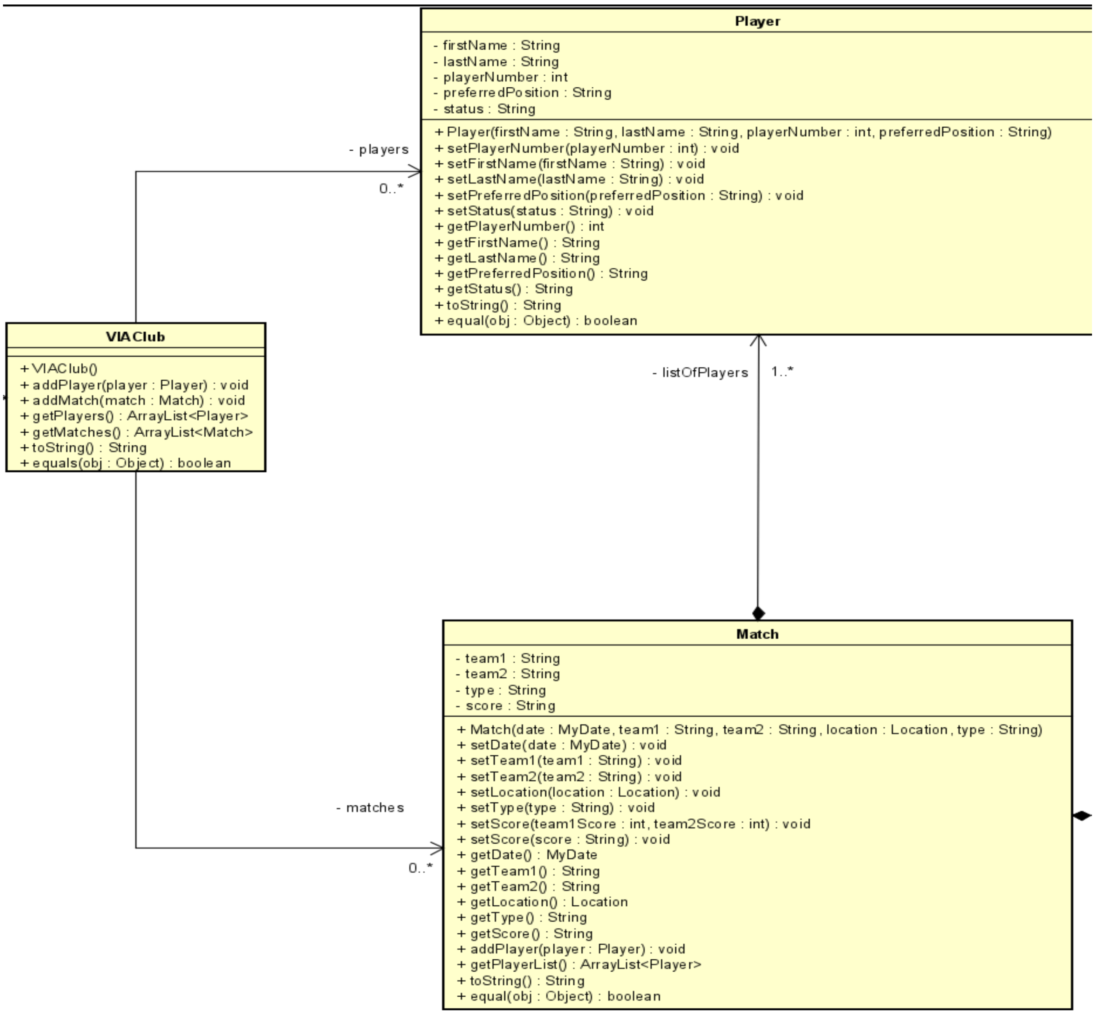

Player
- firstName : String
- lastName : String
- playerNumber : int
- preferredPosition : String
- status : String
+ Player(firstName : String, lastName : String, playerNumber : int, preferredPosition : String)
+ setPlayerNumber(playerNumber : int) : void
+ setFirstName(firstName : String) : void
+ setLastName(lastName : String) : void
+ setPreferredPosition(preferredPosition : String) : void
+ setStatus(status : String) : void
+ getPlayerNumber() : int
+ getFirstName() : String
+ getLastName() : String
+ getPreferredPosition() : String
+ getStatus() : String
+ toString() : String
+ equal(obj : Object) : boolean
- players
0..*
VIAClub
+ VIAClub()
+ addPlayer(player : Player) : void
+ addMatch(match : Match) : void
+ getPlayers() : ArrayList<Player>
+ getMatches() : ArrayList<Match>
+ toString() : String
+ equals(obj : Object) : boolean
- listOfPlayers
1..*
Match
- team1 : String
- team2 : String
- type : String
- score : String
+ Match(date : MyDate, team1 : String, team2 : String, location : Location, type : String)
+ setDate(date : MyDate) : void
+ setTeam1(team1 : String) : void
+ setTeam2(team2 : String) : void
+ setLocation(location : Location) : void
+ setType(type : String) : void
+ setScore(team1Score : int, team2Score : int) : void
+ setScore(score : String) : void
+ getDate() : MyDate
+ getTeam1() : String
+ getTeam2() : String
+ getLocation() : Location
+ getType() : String
+ getScore() : String
+ addPlayer(player : Player) : void
+ getPlayerList() : ArrayList<Player>
+ toString() : String
+ equal(obj : Object) : boolean
- matches
0..*

Figure 6. Relationship in Class Diagram

The “VIAClub” class has knowledge about “Match” and “Player” classes so we can see the Association relationship with multiplicity. To see the class diagram, look at Appendix E.

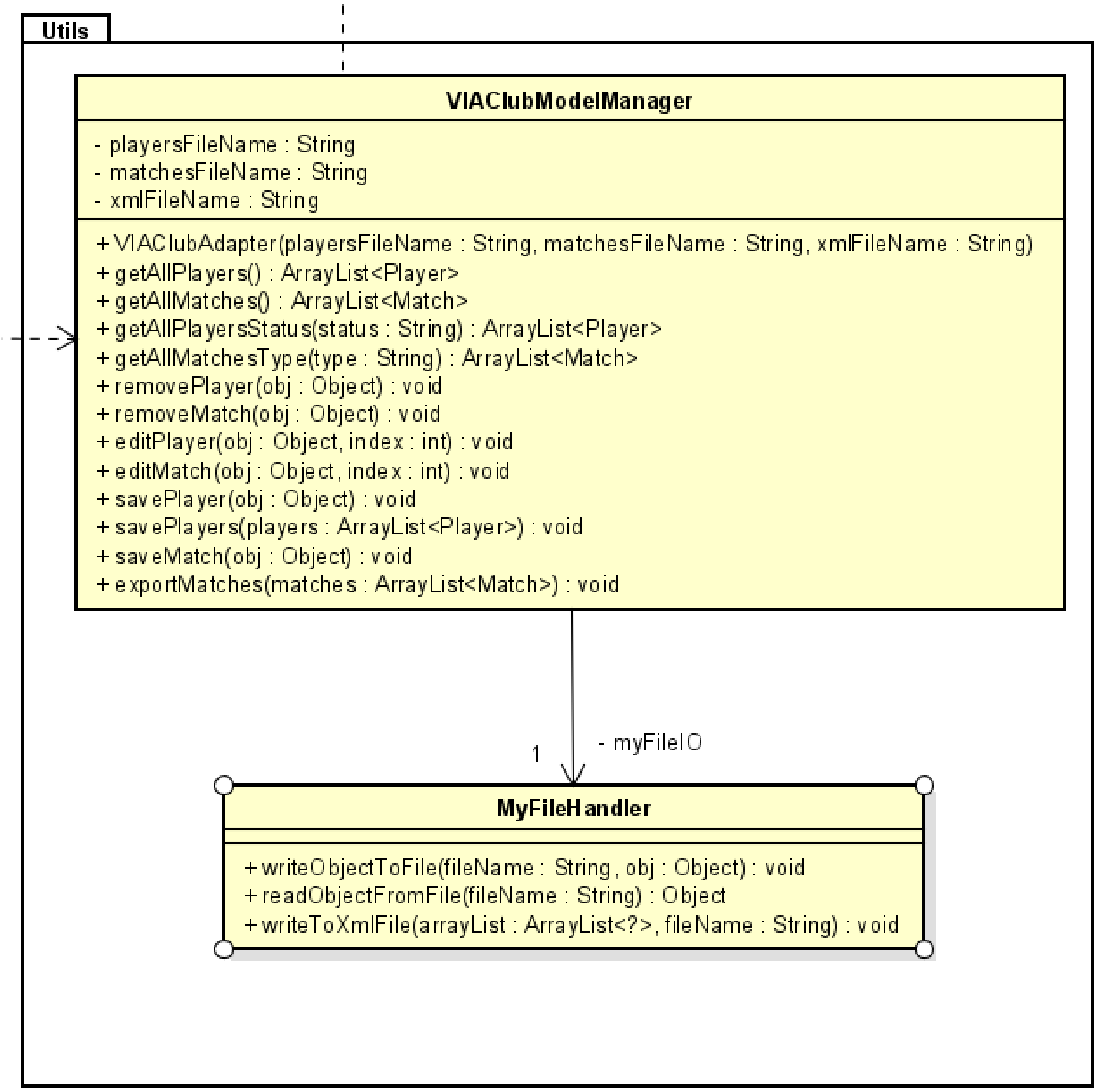

Utils
VIAClubModelManager
- playersFileName : String
- matchesFileName : String
- xmlFileName : String
+ VIAClubAdapter(playersFileName : String, matchesFileName : String, xmlFileName : String)
+ getAllPlayers() : ArrayList<Player>
+ getAllMatches() : ArrayList<Match>
+ getAllPlayersStatus(status : String) : ArrayList<Player>
+ getAllMatchesType(type : String) : ArrayList<Match>
+ removePlayer(obj : Object) : void
+ removeMatch(obj : Object) : void
+ editPlayer(obj : Object, index : int) : void
+ editMatch(obj : Object, index : int) : void
+ savePlayer(obj : Object) : void
+ savePlayers(players : ArrayList<Player>) : void
+ saveMatch(obj : Object) : void
+ exportMatches(matches : ArrayList<Match>) : void
1
- myFileIO
MyFileHandler
+ writeObjectToFile(fileName : String, obj : Object) : void
+ readObjectFromFile(fileName : String) : Object
+ writeToXmlFile(arrayList : ArrayList<?>, fileName : String) : void

Figure 7. Utils class diagram

The class “MyFileHandler” is used for writing and reading data from a binary and to a binary file, like matches and players. Since we have learned that binary files are more useful to store class objects rather than a text file. This class can also write a text file that will export in an XML format using an external library. (Appendix C) In order to import all the match details for the website.

3.2 Sequence Diagram

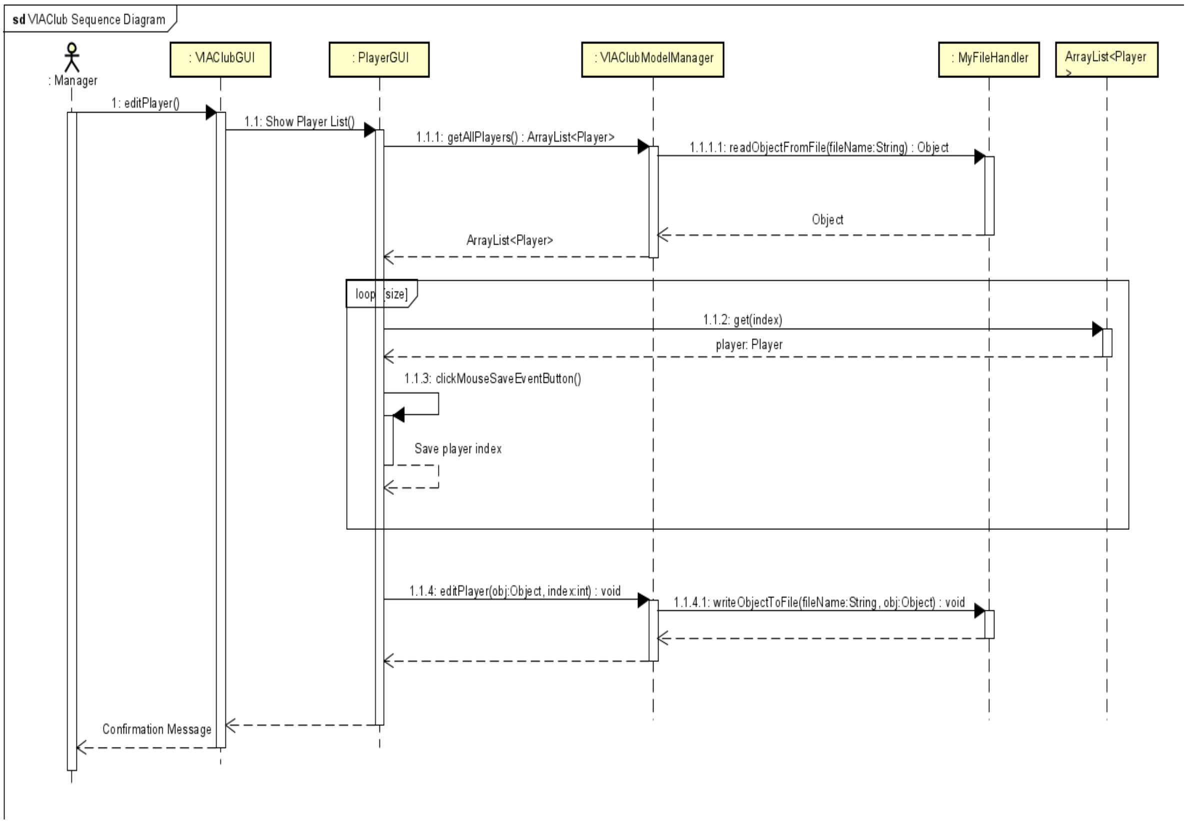

sd VIAClub Sequence Diagram
: Manager
: VIAClubGUI
: PlayerGUI
: VIAClubModelManager
: MyFileHandler
ArrayList<Player >
1: editPlayer()
1.1: Show Player List()
1.1.1: getAllPlayers() : ArrayList<Player>
1.1.1.1: readObjectFromFile(fileName:String) : Object
Object
ArrayList<Player>
loop [size]
1.1.2: get(index)
player: Player
1.1.3: clickMouseSaveEventButton()
Save player index
1.1.4: editPlayer(obj:Object, index:int) : void
1.1.4.1: writeObjectToFile(fileName:String, obj:Object) : void
Confirmation Message

Figure 8. Sequence diagram

The edit player sequence diagram explains and describes the steps that the manager will do to edit a player and the functionality of it.

Firstly, we run the VIAClubGUI, to edit a player the tab Players is called in the PlayerGUIController which will trigger the method getAllPlayers() from the VIAClubModeManager, then the list of all the players at VIAClub will be shown(ArrayList<Player>) and a specific index will be stored on a global variable.

The method readObjectFromFIle() is called to MyFileHandler with the purpose of reading the existing players from the binary file.

The method gets (index) will return the object that is located on the specific index. After the object is received, the clickMouseSaveEventButton() function can save the new data about the object created. The new modifications of a player object will be saved to the binary file by the writeObjectToFile() function that is called from the “MyFileHandler” class.

Finally, a confirmation message will be displayed in the PlayerGUIController for the manager. To see the sequence diagram for editing a player, look at Appendix F.

To see the original files of the diagrams, look at Appendix G.

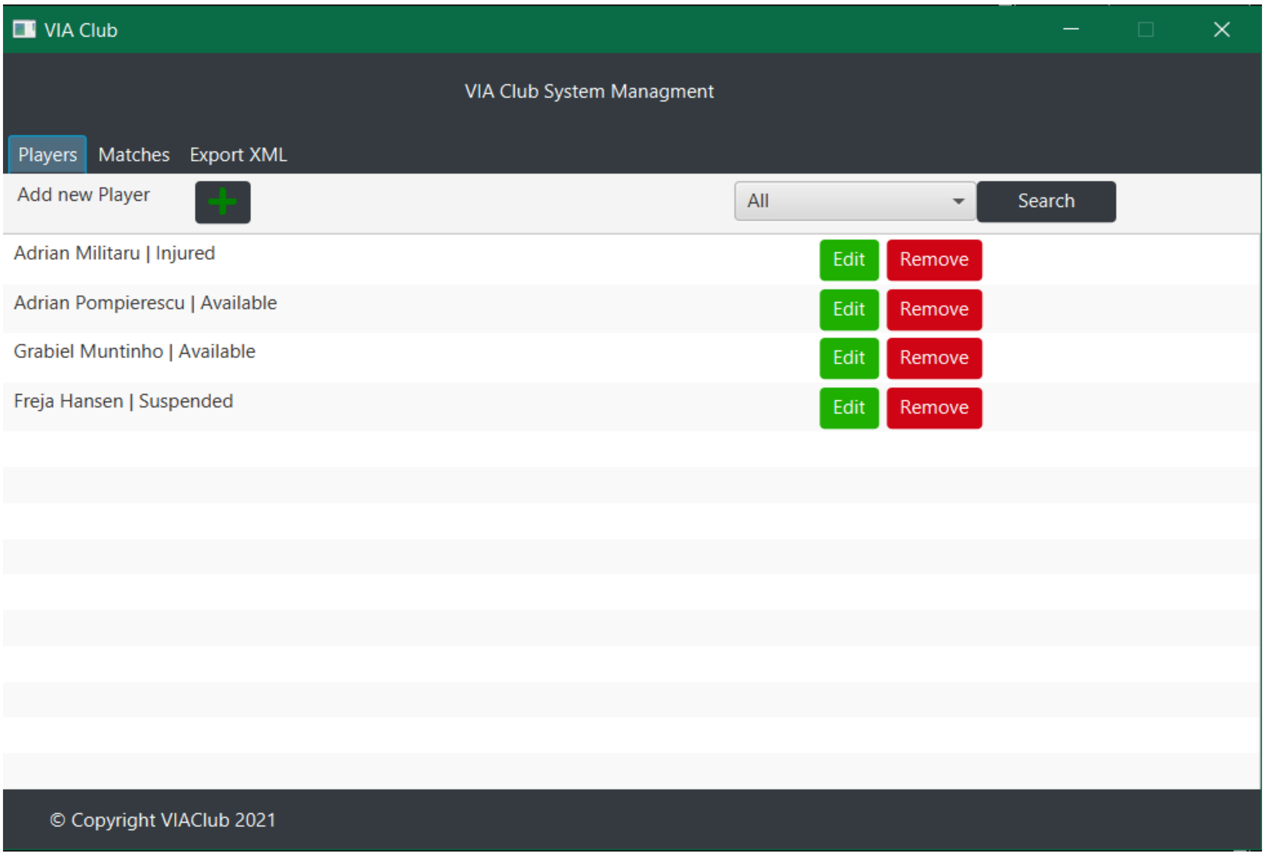
3.3 VIAClub GUI

The first page presented to the manager consists of three tabs, the Player tab, the Matches tab, and the Export XML tab.

Each tab comes with a lot of features like the player tab in which the manager can add a new player to the team, can edit the details of existing players, and remove also, but the most important thing is the search button that can search the players by their actual status.

The player tab displays the name and the status of each player.

Also from the home view, the manager can navigate through tabs.



Adrian Militaru | Injured
Adrian Pompierescu | Available
Grabiel Muntinho | Available
Freja Hansen | Suspended

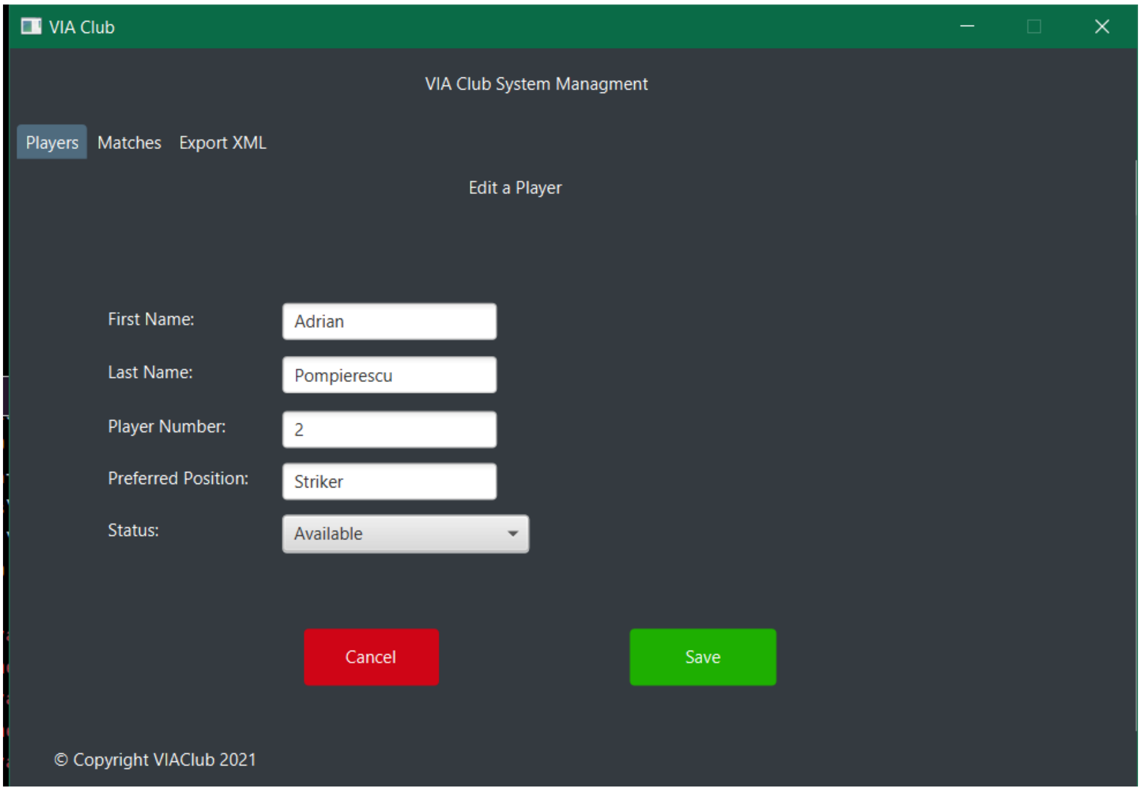
Figure 9. GUI home screen

The first page presented to the manager consists of three tabs, the Player tab, the Matches tab, and the Export XML tab.

Each tab comes with a lot of features like the player tab in which the manager can add a new player to the team, can edit the details of existing players, and remove also, but the most important thing is the search button that can search the players by their actual status.

The player tab displays the name and the status of each player. Also from the home view, the manager can navigate through tabs.

The edit button from the Player tab will display a pane where the manager can change all the information about the player including the first name, last name, number, preferred position, and status.



First Name: Adrian

Last Name: Pompierescu

Player Number: 2

Preferred Position: Striker

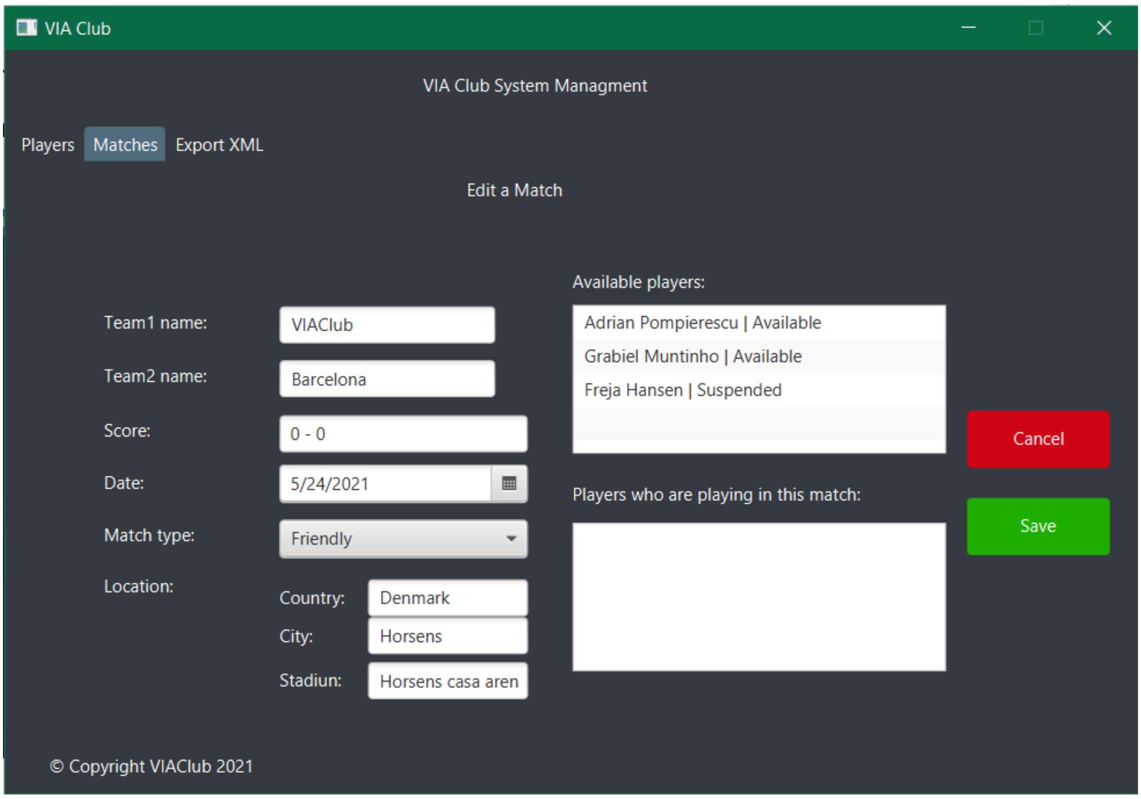
Status: Available

Figure 10. GUI edit player screen

The Matches tab is the same as the Players tab, in essence, of course, the manager can use this tab to add or remove a match or edit a match, and when the list will get crowded the search function will be useful.

The edit pane helps to build the team and keep track of the data about the opposing team, match type location, date, time, stadium, and score.

Also editing all the fields as the manager requires and save them in the system.



VIA Club System Managment

Edit a Match

Team1 name: VIAClub

Team2 name: Barcelona

Score: 0 - 0

Date: 5/24/2021

Match type: Friendly

Location:

Country: Denmark

City: Horsens

Stadiun: Horsens casa aren

Available players:

Adrian Pompierescu | Available

Grabiel Muntinho | Available

Freja Hansen | Suspended

Players who are playing in this match:

© Copyright VIAClub 2021

Figure 11. GUI edit match screen

To make it easier for the manager to export information about the team on the website, on the Export XML tab there is a button to Export MatchList as XML.

For more information about the graphical user interface, see Appendix H.

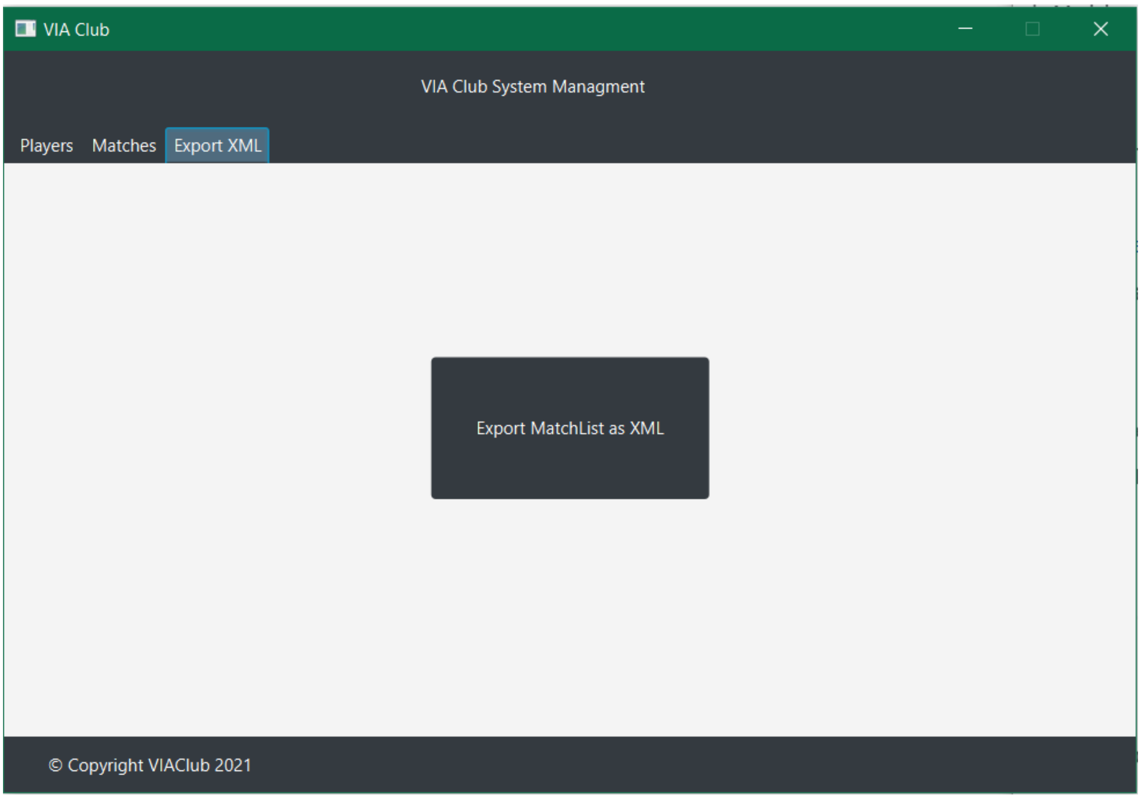


Figure 12. GUI Export XML

3.4 Website

The website for VIAClub should be available for supporters and fans. The desktop version of the website that contains a navigation bar on the top of the window, the body of the page, and a footer. The navigation bar has been split into 4 big categories that the user can navigate through with easy access to the different topics or pages. The logo comes as an extension for the navigation bar, and they are fixed so the user can access a different topic every time. The footer matches the heading in colour and contains different information.

Every page has its design and functionality based on its content. The Upcoming Events category is more special because it was specially designed to support importing external data.

The website is responsive so it can be accessed from every device with different screen sizes. On devices that have a smaller resolution, the navigation bar be collapsed and hide the other pages and reveal it at the action of the user.

For more information about the website, see the website responsive images in Appendix I.

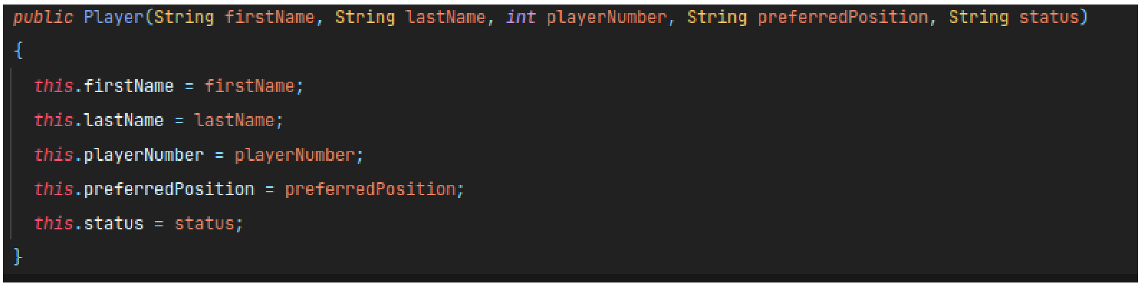
4. Implementation

Based on the design outputs, the implementation process starts. The implementation process began by implementing the basic classes from the class diagram.

4.1 GUI

4.1.1 Create a player

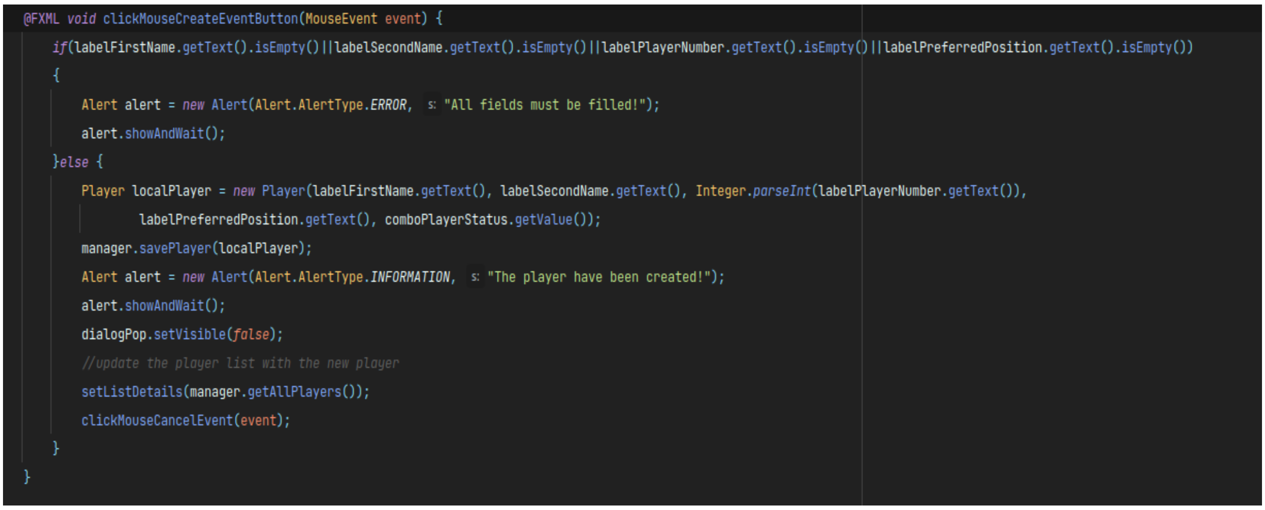
The player creation includes all the player details such a first name, last name, player number, preferred position, and player status.



```java
public Player(String firstName, String lastName, int playerNumber, String preferredPosition, String status)
{
  this.firstName = firstName;
  this.lastName = lastName;
  this.playerNumber = playerNumber;
  this.preferredPosition = preferredPosition;
  this.status = status;
}
```

Figure 13. Constructor for the Player class

A method called “clickMouseCreateEventButton(MouseEvent event)” that are located in the “PlayerGUIController” class will perform the player creation reading all the data stored in the fields and create a new “Player” class stored in the “player.bin” file, after the action was successfully made, a confirmation message will appear to inform the manager that the player has been stored in the system.



```
@FXML void clickMouseCreateEventButton(MouseEvent event) {
    if(labelFirstName.getText().isEmpty()||labelSecondName.getText().isEmpty()||labelPlayerNumber.getText().isEmpty()||labelPreferredPosition.getText().isEmpty())
    {
        Alert alert = new Alert(Alert.AlertType.ERROR,  s: "All fields must be filled!");
        alert.showAndWait();
    }else {
        Player localPlayer = new Player(labelFirstName.getText(), labelSecondName.getText(), Integer.parseInt(labelPlayerNumber.getText()),
                labelPreferredPosition.getText(), comboPlayerStatus.getValue());
        manager.savePlayer(localPlayer);
        Alert alert = new Alert(Alert.AlertType.INFORMATION,  s: "The player have been created!");
        alert.showAndWait();
        dialogPop.setVisible(false);
        //update the player list with the new player
        setListDetails(manager.getAllPlayers());
        clickMouseCancelEvent(event);
    }
}
```

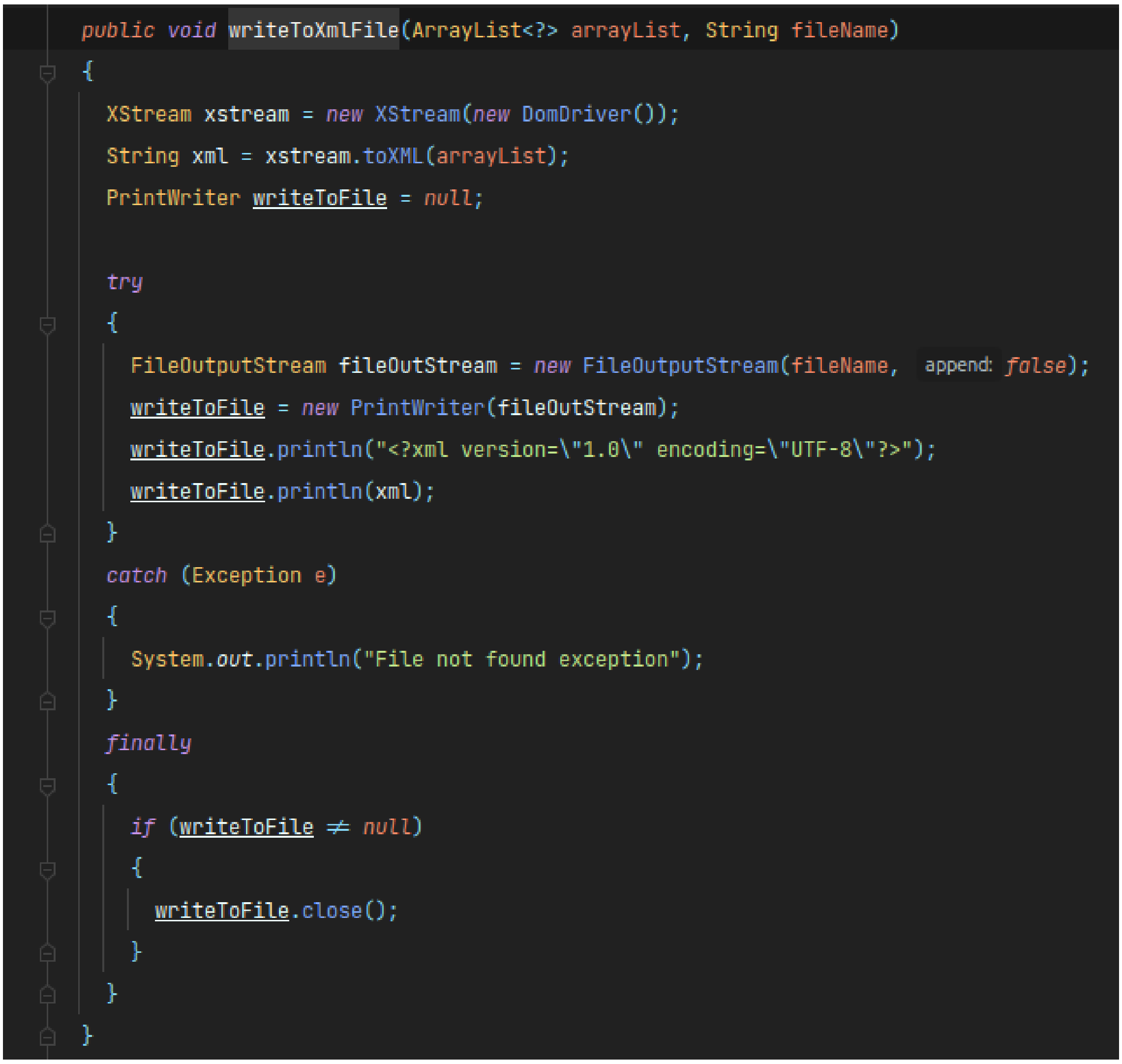
Figure 14. Create player method in PlayerGUIController

4.1.2 Export to XML

The method “exportAction(Mouse event)” that is located on the “ExportGUIController” class will allow the manager to call the function “writeToXmlFile” that is located in the “MyFileHandler” class and it will use an external library “xstream-1.4.17.jar”, this library is located on Appendix C, to export the ArrayList of matches in an XML format.

After the export was successfully done, a confirmation message will appear to inform that the match list has been successfully exported.

To see the source code for the software application and the javadocs generated, look at Appendix J and K.



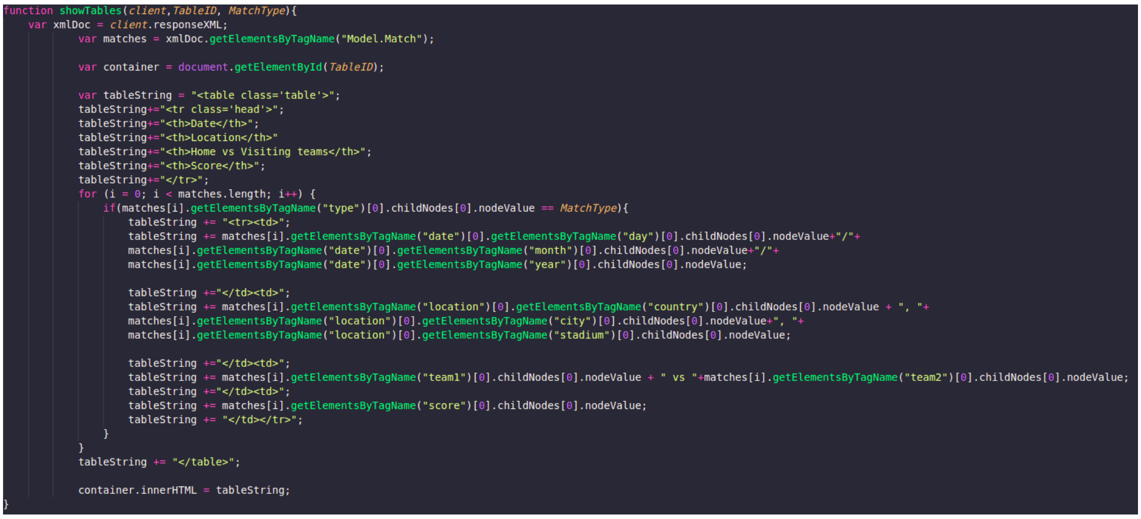
```
public void writeToXmlFile(ArrayList<?> arrayList, String fileName)
{
  XStream xstream = new XStream(new DomDriver());
  String xml = xstream.toXML(arrayList);
  PrintWriter writeToFile = null;

  try
  {
    FileOutputStream fileOutStream = new FileOutputStream(fileName,  append: false);
    writeToFile = new PrintWriter(fileOutStream);
    writeToFile.println("<?xml version=\"1.0\" encoding=\"UTF-8\"?>");
    writeToFile.println(xml);
  }
  catch (Exception e)
  {
    System.out.println("File not found exception");
  }
  finally
  {
    if (writeToFile ≠ null)
    {
      writeToFile.close();
    }
  }
}
```

Figure 15. Export Match list to an XML file

4.2 Website

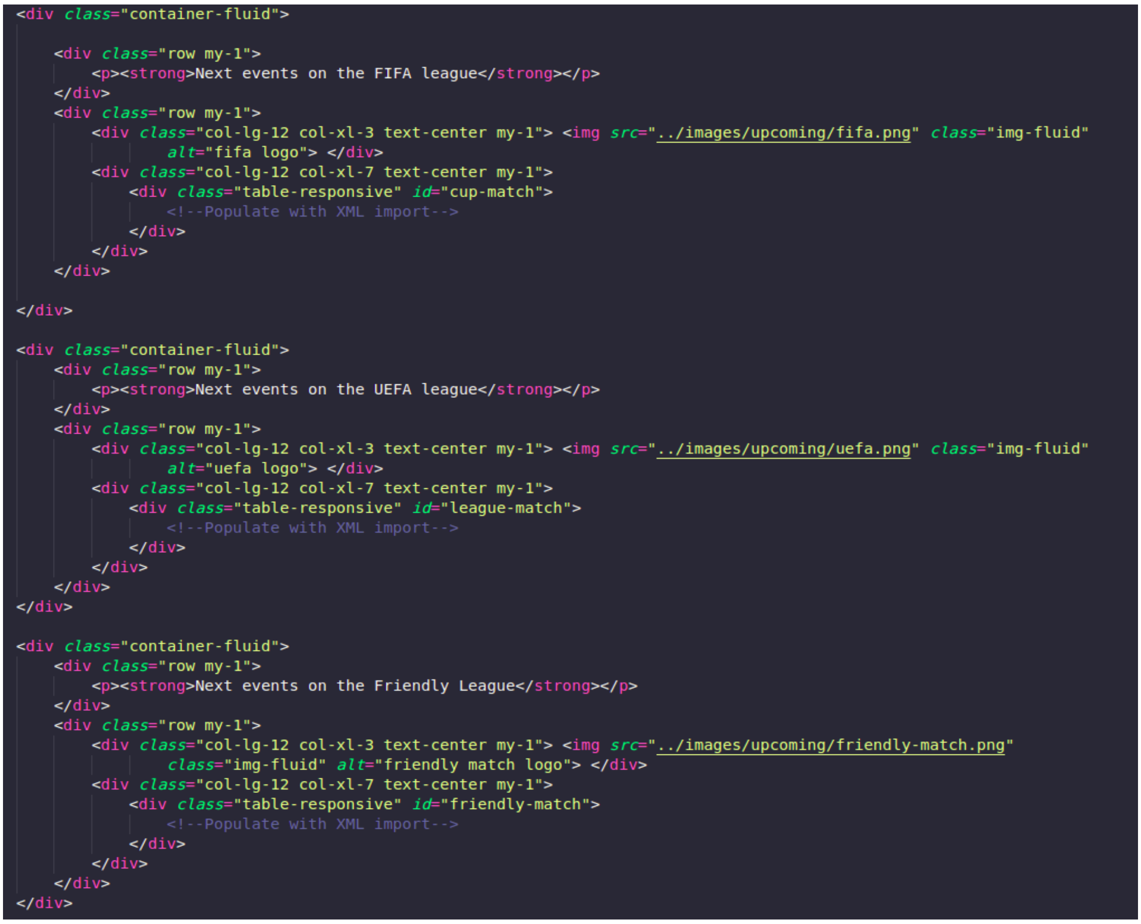
The JavaScript code has been used to import the XML file on the website and represent it under a table format in HTML code.

The function “showTables()” will read all the matches that are located in the XML file and base on the parameters will read a specific type of match and will assign it to a specific table id. To see the source code for the website, look at Appendix L.



```
function showTables(client,TableID, MatchType){
    var xmlDoc = client.responseXML;
        var matches = xmlDoc.getElementsByTagName("Model.Match");

        var container = document.getElementById(TableID);

        var tableString = "<table class='table'>";
        tableString+="<tr class='head'>";
        tableString+="<th>Date</th>";
        tableString+="<th>Location</th>"
        tableString+="<th>Home vs Visiting teams</th>";
        tableString+="<th>Score</th>";
        tableString+="</tr>";
        for (i = 0; i < matches.length; i++) {
            if(matches[i].getElementsByTagName("type")[0].childNodes[0].nodeValue == MatchType){
                tableString += "<tr><td>";
                tableString += matches[i].getElementsByTagName("date")[0].getElementsByTagName("day")[0].childNodes[0].nodeValue+"/"+
                matches[i].getElementsByTagName("date")[0].getElementsByTagName("month")[0].childNodes[0].nodeValue+"/"+
                matches[i].getElementsByTagName("date")[0].getElementsByTagName("year")[0].childNodes[0].nodeValue;

                tableString +="</td><td>";
                tableString += matches[i].getElementsByTagName("location")[0].getElementsByTagName("country")[0].childNodes[0].nodeValue + ", "+
                matches[i].getElementsByTagName("location")[0].getElementsByTagName("city")[0].childNodes[0].nodeValue+", "+
                matches[i].getElementsByTagName("location")[0].getElementsByTagName("stadium")[0].childNodes[0].nodeValue;

                tableString +="</td><td>";
                tableString += matches[i].getElementsByTagName("team1")[0].childNodes[0].nodeValue + " vs "+matches[i].getElementsByTagName("team2")[0].childNodes[0].nodeValue;
                tableString +="</td><td>";
                tableString += matches[i].getElementsByTagName("score")[0].childNodes[0].nodeValue;
                tableString += "</td></tr>";
            }
        }
        tableString += "</table>";

        container.innerHTML = tableString;
}
```

Figure 16. Import Match list from an XML file to the website



```
<div class="container-fluid">

    <div class="row my-1">
        <p><strong>Next events on the FIFA league</strong></p>
    </div>
    <div class="row my-1">
        <div class="col-lg-12 col-xl-3 text-center my-1"> <img src="../images/upcoming/fifa.png" class="img-fluid"
                alt="fifa logo"> </div>
        <div class="col-lg-12 col-xl-7 text-center my-1">
            <div class="table-responsive" id="cup-match">
                <!--Populate with XML import-->
            </div>
        </div>
    </div>

</div>

<div class="container-fluid">
    <div class="row my-1">
        <p><strong>Next events on the UEFA league</strong></p>
    </div>
    <div class="row my-1">
        <div class="col-lg-12 col-xl-3 text-center my-1"> <img src="../images/upcoming/uefa.png" class="img-fluid"
                alt="uefa logo"> </div>
        <div class="col-lg-12 col-xl-7 text-center my-1">
            <div class="table-responsive" id="league-match">
                <!--Populate with XML import-->
            </div>
        </div>
    </div>
</div>

<div class="container-fluid">
    <div class="row my-1">
        <p><strong>Next events on the Friendly League</strong></p>
    </div>
    <div class="row my-1">
        <div class="col-lg-12 col-xl-3 text-center my-1"> <img src="../images/upcoming/friendly-match.png"
                class="img-fluid" alt="friendly match logo"> </div>
        <div class="col-lg-12 col-xl-7 text-center my-1">
            <div class="table-responsive" id="friendly-match">
                <!--Populate with XML import-->
            </div>
        </div>
    </div>
</div>
```

Figure 17. HTML code where all the matches will be populated

5. Test

After the implementation of the website and GUI, the system is tested, and it is ensured that all the use cases are fulfilled.

Use Case	Test Result	Comments
Manage Players	Works	The manager can read all the data of a player, create a new player, edit an existing player, or remove a player.
Manage Matches	Works	The manager can read all the data of a match, create a new match, edit an existing match, or remove a match.
Assign player to a match	Works	The manager can assign a player to a specific match, based on the match type and the player status
Export match list to the website	Works	The manager can export all the matches that are stored in the system in an XML file that can be imported into the VIAClub website.
View website	Works	The supporter can visit the VIAClub website and see what the next matches under the upcoming events tab.

6. Results and Discussion

The following table has been made to represent all the functional requirements used in the analysis phase.

Priority	Requirement id	Requirement	Status
Critical	1.	As a manager, I need a system to make matches and upcoming matches.	Working
Critical	2.	As a manager, I want to be able to write the date of a match and, and who the opponent will be.	Working
Critical	3.	As a manager, I need a system to store a list of players that I want to include in the squad for the next match.	Working
Critical	4.	As a manager, I need to be able to make League and Cup matches that suspended or injured players cannot play.	Working

Critical	5.	As a manager, I need to be able to make Friendly matches where injured players cannot play.	Working
Critical	6.	As a manager, I need to be able to update and keep track of the player’s details.	Working
Critical	7.	As a manager, I need to keep tracking of the injured or suspended players.	Working
Critical	8.	As a manager, I need to be able to add and remove players when they are sold or bought.	Working
Critical	9.	As a manager, I need to have an overview of the player to have a first name, last name, number, preferred position.	Working

Critical	10.	As a manager, I need to have an overview of players that are injured and try to also keep track of when and how long they might be suspended (Status).	Working
Critical	11.	As a manager, I need to be able to edit a player's position after it gets re-trained.	Working
Critical	12.	As a manager, I need to be able to edit matches if plans change.	Working
Critical	13.	As a manager, I need to be able to keep track of previous matches to be	Working
Critical	14.	As a manager, I need to keep track of the number of substitute players for a specific type of match.	Working

Critical	15.	When an issue occurs, the problem pop-up a message.	Working
High	16.	As a manager, I need to send the match details on the website.	Working
High	17.	As a manager, I need a responsive website with more details regarding the matches and the club.	Working
High	18.	As a supporter, I need to keep tracking the match history and the match details on the website.	Working
Low	19.	Every update in the system includes writing to a file.	Working
Low	20.	The program is in the java language.	Working

Low	21.	To have an interface that can be accessed with a mouse and a keyboard.	Working

The implementation process started by implementing all the methods and fields from the class diagram and fulfil all the requirements from the activity diagrams base on every use case.

After the implementation process has been successfully made, the team realizes that some methods of the “VIAClub” class are not used, and these modifications have been reflected on the class diagram.

7. Conclusions

The manager contacted the group with the issue he had while trying to create matches. He either lost the paper with his work or was unable to remember all the details, he looked for a solution to the problem.

To summarize, the goal of the project was to create a software application that can manage a football team, which will help VIAClub with storing data for players and matches and to export it to a newly designed website that will work on every device in order to make the manager's job easier.

The Analysis started with getting all the requirements that the customer provided during his interview. While going through the design and implementation of the software, every aspect of the project was formed based on the Analysis phase in order to fulfil all the customer's requirements.

From the project description the requirement list was made, which made it possible for the Class Diagram was created from which it was possible to create the program. The website was created and functioning.

To make sure the client's requirements were fulfilled a test was done and the conclusion is that the project is fulfilling the requirements and solves the client's issue.

8. Sources of information

Duckett, J., 2011. *HTML and CSS : Design and Build Websites.* First Edition ed. Indianapolis, IN46256: John Wiley & Sons, Inc..

Duckett, J., s.f. *Javascript & jQuery, Jon Duckett.* First Edition ed. s.l.:s.n.

Gaddis, T., s.f. *Starting Out with Java: Early Objects, Global Edition.* 5th Edition ed. s.l.:s.n.

LaGrone, s.f. *HTML5 and CSS3 Responsive Web Design Cookbook.* s.l.:s.n.

9. List of Appendices

Appendix A: Project Description

Appendix B: Use Case Description

Appendix C: External library used for application

Appendix D: Activity Diagrams

Appendix E: Class Diagram

Appendix F: Sequence Diagram

Appendix G: Astah File

Appendix H: User Guide

Appendix I: Website responsive images

Appendix J: Software Application Source Code

Appendix K: JavaDocs

Appendix L: Website Application Source Code

Process Report

Group 1

Gabriel Moutinho Tristan 304376

Adrian-Cristian Militaru 308842

Adrian Pompierescu 309774

Freja Kammersgaard Hansen 308840

Allan Henriksen

Mona Andersen

Software Technology Engineering

First semester

4th of June 2021

Table of content

1 Introduction

In the following report, our team reflects upon the process of making the first-semester project. We focused on how the group works based on cultural differences, challenges, communication, and time management.

Our team started the project by forming a group and establishing some basic rules for weekly team meetings to discuss problems or the next steps to do. This is mentioned in the group contract as well.

We followed the contract plan for time-schedule but also managed to stay flexible if something happened. In the last two weeks of doing the project, we worked daily by holding online meetings or monitoring the activity using GitHub for the application. The meetings were held on the Discord application because with the COVID-19 situation we are not allowed to work on VIA's campus.

As a group, we decided to discuss with a supervisor and explain the progress that has been made on the project.

2 Group Description

2.1 Cultural background

Our team has representatives from Romania, Denmark and Spain. We used tools such as the Hofstede method and E-Stimate profiles to analyse our cultural differences. The group roles and personalities were examined with the help of those two platforms.

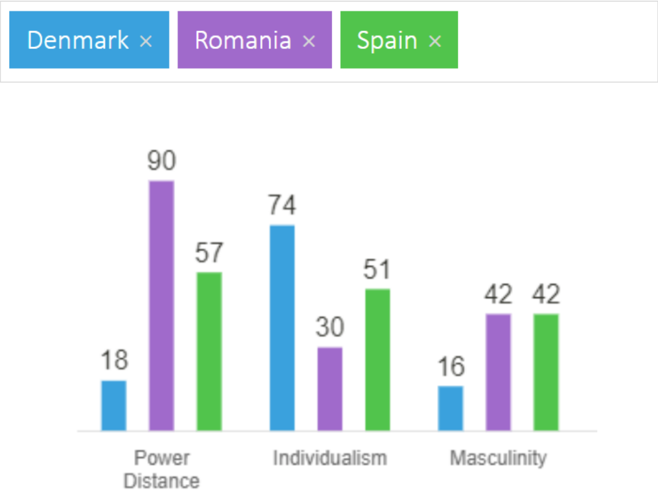


Figure 1 .source: Hofstede Insights, 2021

As per the graph from figure 1, the first difference in national culture was the power of distance. According to figure1, Romania is higher than Denmark with 72 points and higher than Spain with 33 points.

This dimension deals with the fact that all individuals in societies are not equal – it expresses the attitude of the culture towards these inequalities amongst us. Power Distance is defined as the extent to which the less powerful members of institutions and organizations within a country expect and accept that power is distributed unequally. (Hofstede Insights, 2021)

In contrast, it was concluded that all members of our team have different expectations than what the graph represents. We all believe in equality. Also, the formality in our relations and communication between us was much higher than the power distance.

The next dimension of national culture our group analysed was individualism.

This refers to whether a person is focused on doing a specific task given by other members in our weekly team meeting.

Although the individualism in Denmark, with a score of 74 is an Individualist society. This means there is a high preference for a loosely knit social framework in which individuals are expected to take care of themselves and their immediate families only. It is relatively easy to start doing business with the Danes. Small talk is kept at a minimum and you do not need to create relationships first. Danes are also known for using a very direct form of communication. (Hofstede Insights, 2021)

The final dimension discussed was masculinity. The main rules in our team were to use quality and mutual support besides the sharing of knowledge in some specific fields.

2.2 Group roles E-stimate

After we analysed our cultural differences using the Hofstede theory, we looked at our individual E-stimate profiles. This platform helps us understand our cultural differences and understand each other’s expectations and build the base of our communication as a team.

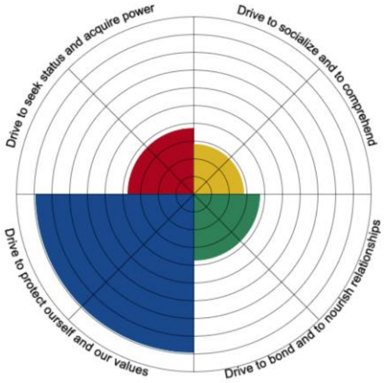


Figure 2. Adrian-Cristian Militaru

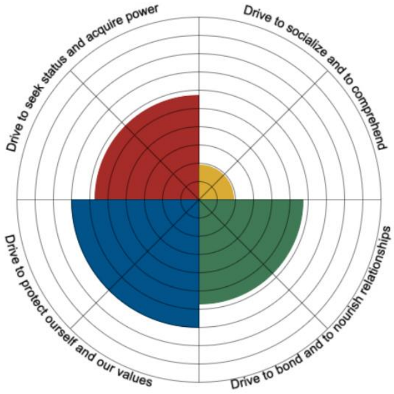


Figure 3. Adrian Pompierescu

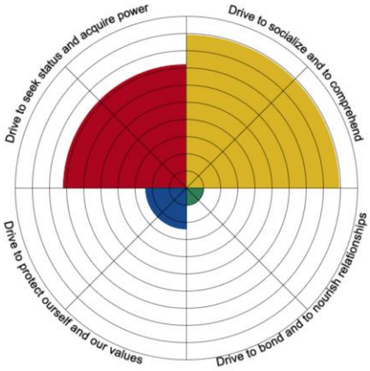


Figure 4. Gabriel Moutinho Tristan

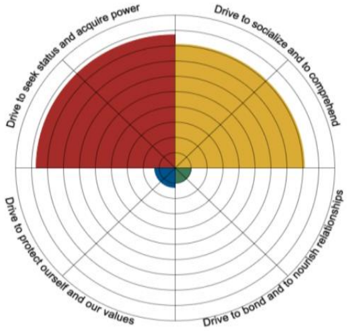


Figure 5. Freja Hansen

Our team covers mainly the blue and red with yellow, as well as green to only for one of us (Figure 3) (Wendel, 2021)

In general, the tendencies are well balanced between all the members of our group are from extreme blue to extreme red and extreme yellow (Figure 4)

Inside our group even if there are differences in tendencies, we still worked well together, and we have not encountered any negative experience.

In our opinion, this can be because, even though the green colour is not dominant in any of us, we all took very seriously to communicate respectfully to each other's and keeping a good atmosphere inside the group.

The fact that two members are clearly blue predominant helped a lot with the structure, the organization, and the discipline of the group.

In general, we think that the fact that all member's personalities are kind of complementary to each other's made the group work an easier task.

3 Project Initiation

The topic of the project was provided to us in the form of an interview with the VIAClub Manager Bob Oldenuff, our potential customer.

The group was formed at the request of all the members and later approved by the supervisor.

We did our initial plan for the project description, and we set a series of milestones. Almost all of them had to be postponed, except for the last milestone about writing the reports.

We think that when we did the initial planning, we did not have the experience to know the workload required at every step and we were being taught along the way about the concepts, but the overall planning was realistic.

4 Project Description

The provided interview with VIAClub manager included all the client's requirements for the system we were supposed to develop, but in a format that was not easy to work with.

After studying the interview carefully, we were able convert that information into a project description, which provided our team with a better understanding of the customer requests, and we were able to use the project description as a base for the project.

The final version of the project description was evaluated by our supervisors before starting to analyse.

5 Project Execution

During the first part of the analysis, it was difficult to see the meaning behind it. Without any experience with subtracting information from interviews and making the needs and requirements more functional than subjective needs, some things seemed unnecessary at the time. With help from the supervisors, it was chosen that it would be better to have all the informational requirements. The team used the waterfall method which is a hard method to follow up on unless there are no mistakes made during the process (however, some minor changes on the class diagrams were made after the implementation), which is why we had to move away from our original plan and therefore couldn’t continue using the waterfall method.

In the second stage of the analysis, the information that was gained and compressed was now to be used in the class diagram, which made the analysis useful. In this state, it was clear why it is better to have all the information as some things that might have been deemed unimportant are rather important for the project to be complete.

After clearing out all the analysis and design, it was time to use all the compressed information that has been gained up until the making of the program. The program was written using GitHub for the team to be able to work together without compromising each other's work. GitHub has made working together as a team more effective.

The program was successful, as it follows the Class diagram as well as fulfils all the requirements that the client needed.

6 Personal Reflections

6.1 Adrian Militaru's reflections

6.1.1 Project development

Developing the project using the waterfall methodology, our team started out by following all the steps from waterfall methodology which later results in the best way to coordinate our work as a team and re-evaluate a step before the main result.

During the analysis part, our group gathered and categorized all the requirements for our system based on the interview with the customer and extracting some specific statements or keywords.

The classification of requirements made us understand what is necessarily important for the customer and what is irrelevant or optional. This step helps us to proceed into the design, implementation, and test phases.

The design phase consisted of creating the class and sequence diagram even the use case diagrams for every functionality that the customer is required.

During the implementation phase, we implement everything that we did in the design phase and edit some of the elements from the class diagram to offer an optimized version.

As our team went through the stages of development, I have improved my understanding of how each phase from the waterfall methodology works and contributes to the development of a system. In the future, it will be a base approach to contact or inform the customer about every phase and what is the evolution of the project for better communication between the team and the customer.

6.1.2 Group cooperation

After the group was formed, we spent time getting to know each other. The team was formed based on a wish to work together, which helped us easily establish some ground rules and goals that must be achieved by the end of the first semester and helped us create the group contract.

For me, the most important part of working with a team is to understand and see new options or opinions for a specific subject that helped us later in the project development.

On the other hand, sometimes some internal conflicts can appear, but hopefully, we pass everything and work as a team to finish the project on time.

6.1.3 Future improvements

As a future improvement for our group is that see every member getting more knowledge in programming and for an easy approach in the developing for the future projects

6.2 Gabriel Moutinho’s reflections

We started the project by defining a group contract at the very beginning and now that the project is finished, we can look back and say that we respected all the initial agreements. The contract, in my opinion, covered the minimum terms of group conduct and cooperation required for a group to function. Luckily, we did not have to refer to it during the project because all the members participated actively in group meetings and we split all the tasks, everybody delivered on time.

When it comes to working together, I must thank Adrian Militaru, because he helped us a lot configuring and using GitHub so we could effectively collaborate programming the website and the application and he also took kind of a “manager” role in the group that helped us a lot with coordination issues and to work all in the same direction.

Since the group contract worked well for us, I do not suggest any major changes for future group contracts. I think that the success of our group working together is because we all respected the terms of the contract, even without thinking about it. Breaking any of the rules of the contract would have negatively affected our performance as a group.

All members of the group, in my opinion, contributed satisfactorily to the project. Some of us had more experience in the fields than others, but I have to say that we all contributed with our effort in all areas of the project.

The motivation of the group changed over time because at the beginning of the semester we were all very motivated, but we did not have the knowledge to start the project, and for some weeks it was on stand-by, only working on the website as a part of the RWD curriculum.

Then, after some Wednesdays working on analysis and design, we found ourselves two weeks before the deadline with all the reports to write and all the Java programming to do. Luckily, these two weeks were full-time, and we were able to make it. We worked these last days together over discord and that really motivated us to keep working.

About the benefits and challenges of being a multicultural group, I must say that we were all working in the same direction, we did not encounter any challenge because of this situation and neither there were not any direct benefits.

I learned a lot during this project, more exactly, how to cooperate in a group context, and I think that being organized as a group is key to having all members working towards the same goals. Next time I am involved in group work I will check that we are well organized, even if in my personality the colour yellow is dominant and it is not a yellow characteristic, but I realized that it is crucial.

In my opinion, the advantages of group work and problem-based learning are that the students can apply knowledge to solve problems that can be like ones in real life and that they must solve them together, and in the process, the students can learn even more.

The disadvantages are that sometimes it is difficult for people to work together, but I think that it is an ability that can be trained.

6.3 Freja Kammersgaard Hansen’s reflections

After we got split into our SEP1 groups we wrote a group contract stating how we wanted to work together as a group as well as set up some demands that we expected of each other such as that we wanted to keep each other informed if we were unable to meet up and how the tone should be in the group as well as when it is okay to set up meetings.

I feel responsible for my part of the project as I respect my team members and their time, I feel like we all had an equal part in the project and the workload.

We all tried our best to deliver good results while some cultural differences have made the group work interesting and both easier and harder. An example of this can be that hand-ins need to follow the exact requirements, in some countries the student is always supposed to write more than the requirement and is applauded for such, while in the Danish school system this is not the case, and you should always stay within the limit of the requirement.

I suggest for a new contract that when choosing a working schedule that everyone's wishes are aligned to avoid creating issues with private life.
For the group to be multicultural along with working with completely new people it has been a delight to learn about different ways of solving the same issue, how we all have different values and how to put our values together to get a good perspective which helped us to learn how to work together to achieve a goal.

I have learned for myself to put others before my own needs in the group, to help each other. This can be by accommodating a member’s suggestion for a time even though I would rather at another time.

The motivation in the group always peaks in the beginning when we start on a new part as it is new and exciting but fades rather quickly after. We work hard and effectively for us to still maintain a private life and not wanting to work on the weekends.

I believe that the project description was a great way to analyse all the information and process the information to make sure we remember but also can go back to and reread to check up if the data is correct and that we made sure everything was accounted for and fulfilled.

6.4 Adrian Pompierescu' s Reflections

6.4.1 Group work

Our group contract was like a mandatory thing for us. I just want to think that we had to puzzle each other. We firmly did respect every criteria according to our group contract, like the deadlines, time plans, meetings and the most important was that we communicate respectfully with each other.

As I started working on the project, I felt overwhelmed by all the things that I had to put together to deliver my best performance and made me anxious and uncomfortable.

Nevertheless, this was the first project for me that comes with a lot of question marks but in this way, I realized that the waterfall methodology is a step-by-step method to be followed because each step is the foundation for the next step.

As a group we tried to split the task between us so everybody can contribute fairly to the project, of course, we are from different countries, and we do not have the same level of experience. Like some of us had more packages of knowledge and different education standards.

But in the end, our group members manage every task maybe because we show that we are very serious about this project and school in general.

I believe being in an international group did not affect us at all, after all, we all have learned under the same curriculum, the only thing we had to do was to comply with every need and was not hard at all.

The motivation of our group was like a roller-coaster with ups and downs. Some of us had work to go to and of course the private life. I do not believe I felt demotivated like a lack of motivation at all, but sometimes maybe a lack of focusing but we never did cancel a meeting or workflow.

I can say I have learned a lot this semester and doing this project group, but it is not only this project, but I also felt like all the other assignments that we have as a group turn out to solidify our group and make us stronger to surpass every challenge that we had so far.

In other words, I had come with zero knowledge about all the curriculum and finished with being part of an great group and full with knowledge and motivation to go further.

In my vision, each member has worked hard to define the problem, develop hypotheses, gather information, and arrive at a clear solution. I think that collaboration in the group makes it strong and keeps it evolving.

In the future, I would like to adopt the discussions to develop much further and to negotiate the establishment of quality criteria and construct a common solution.

6.4.2 Project organized studies and problem-based learning.

I personally think that collaboration in our group was so superb. We plan to gather more information, then do the necessary research and reconvene to share and summarize new knowledge in the group, even if there may or may not be a product. And that means we had adequate time for reflection and self-evaluation, collaboration is an essential component of PBL, Collaboration versus Individual Learning (Hoffman & Ritchie, 1997)

7 Supervision

The supervision of our project was held via Discord and e-mails or on the Zoom meeting. Most of the meetings with the supervisors were not scheduled in advance. In some cases, only one member of the group was present to talk with a supervisor regarding a specific part that was assigned by the team. After every meeting, all the other members from the group were informed of the discussion to ensure the group was on track and helped us make the right decisions until the next meeting with a supervisor.

All the group members were presented in a meeting, especially when they are multiple questions and problems regarding the project and the documentation that we need to provide.

8 Conclusions

To conclude the process and group work there have been a few complications from time to time, but the group has succeeded in working together and using the given tools to comprehend and successfully ensure that the project is completed and is living up to our own expectations. The given tools and material have been a great help in achieving the group's goals.

Some tools have been more helpful than others, such as the group contract has not really been brought up due to no intern fights in the group and everyone doing their best to succeed.

Our recommendations for a group to be successful based on our experience are respect each other’s members and communicate respectfully, take responsibility for the assigned tasks, help each other's when needed, and have a high level of structure within the group. Organization is very important because at the end simplifies the tasks.

9 References

Hoffman, B. & Ritchie, D., 1997. *Teaching and Learning Online: Tools, Templates, and Training.* s.l.:s.n.

Hofstede Insights, 2021. *Country comparison.* [En línea]
Available at: https://www.hofstede-insights.com/country-comparison/denmark,romania,spain
[Último acceso: 29 5 2021].

Wendel, M., 2021. *E-Stimate Personal profiles [pdf].* [En línea]
Available at:
https://via.itslearning.com/ContentArea/ContentArea.aspx?LocationID=19156&LocationType=1