

Chapter 4: System Analysis and Design

4.1 Introduction

This chapter mostly deals with the analysis and design of the system. To be specific this chapter divides the requirements into functional and non-functional requirements to understand the user requirements and the system requirements. Moreover, the chapter also includes details about the specific system architecture used for handling request and also the design diagrams that help the developers to know what to code. The specific diagrams include use case diagram, class diagram, system sequence diagram, sequence diagram, entity relationship diagram, activity diagram and lastly the database schema. The list of diagrams mentioned above are those that are required if the analysis and design approach chosen is Object-oriented. However, on top of the diagrams there are a few wireframes also created that shows blueprints of the system.

4.2 System Requirements

System requirements is required for a software engineer to help in facilitate designing the diagrams for the developers so that they can know what to code in order to develop a system that meets the user requirements. Thus, the same applies to the player performance prediction model. These requirements are divided into functional requirements and non-functional requirements. Specific functional and non-functional requirements are mentioned below.

4.2.1 Functional Requirements

Functional requirements refer to the specific abilities of a system specifically conducted by a specific user. The player performance prediction model will have 3 users that is the admin, the club and the spectators. Starting with the specific requirements of the admin that are:

- i. To login and access the admin module after inputting the correct credentials. This happens after a verification and validation process with the credential match.
- ii. Register a club so that the club can add their players.
- iii. Edit and view clubs and their associations.
- iv. Start and end a season or a tournament.
- v. View leaderboards of players.
- vi. View scores of ongoing matches and previous matches.
- vii. Generate leaderboards of players at the end of a season or a tournament.

The second user is the club and the requirements of this user is:

- i. To login and access the club's module after entering the correct credentials.
- ii. To register new players to their club.
- iii. Edit the registered players information.
- iv. To score for their matches basically for matches played by this club.
- v. View scores of ongoing and previous matches.
- vi. View leaderboards of players in the current and previous seasons and tournaments.

The functional requirements of the final user that is the spectator is:

- i. To register themselves to the system.
- ii. To login and accessing the spectator module after entering the correct credentials.
- iii. To view scores of ongoing matches and previous matches.
- iv. To search clubs, players and view their statistics.

4.2.2 Non-Functional Requirements

Non-Functional requirements refer to the intangible requirements of the system that define the user experience. Although the systems can function without these requirements, but it is necessary to attract new users and retain the existing users. For the player performance prediction model, the non-functional requirements are:

- i. The system needs to be reliable meaning that the data stored in the database needs to be accurate, like for scores of players they need to be accurately stored since the prediction of the players performance will be done based on this score so if the scores stored are inaccurate then the predictions will be inaccurate. This is done by picking the exact score of a player and once a match is over the scores should not be editable and should be exported to the database the way it is.
- ii. Availability is also required so that whenever the user wants to register or log in to the system, he/she is able to do so without facing any difficulties.
- iii. The aspect of security also comes in although the player performance prediction model has no integration of money hence less prone to be attacked by the hackers but still, they can hack into the system and play around with the scores and statistics which is very risky if

the data stored is used for decision making. Thus, the viewing module should be restricted to registered users and should be only view only data and not editable.

- iv. Should be user friendly for a non-IT user or a naïve user to ease their activity with the system. There can be a recorded tutorial on how to use the model and also on the main page there can be a question button that can provide any help to users.

4.3 System Analysis Diagrams

4.3.1 System Architecture

System architecture refers to the structural design of the system trying to show what happened at the back end when a request to access a service from the system is made by a client. The proposed player performance prediction model has a 3-tier system architecture which is a client-server architecture. As evident from the figure below from the system architecture diagram the user can make a request to access service from the system by send a request from the web browser after being connected to the internet. The web browser then requests the web server for the request via the internet from which the request is taken to the database server that responds accordingly to the request made.

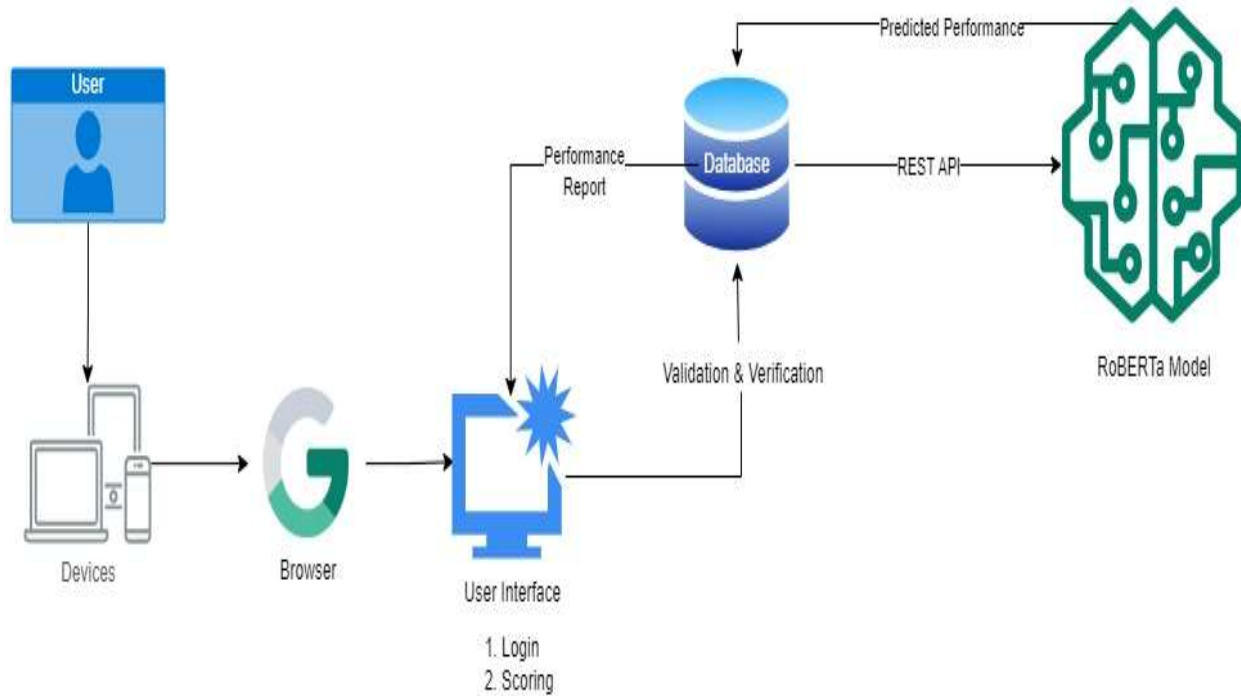


Figure 4. 1: System Architecture

4.3.2 Use Case Diagram

As evident from the diagram below the use case is used to show the roles and abilities of all the users in the system. There are three users of the system that is the Admin, the Club and the Spectator. The admin is the primary actor followed by the club and then the Spectator. The main role of the club is to do scoring for the matches so that the leaderboard is updated with the correct statistics and then this data can be retrieved by the admin so that he/she can feed this data into the prediction model that will analyse the data and make a prediction is the performance of the player will be for the future matches. The spectator is a minor user who can view all the scores of the current and previous scores of matches being played.

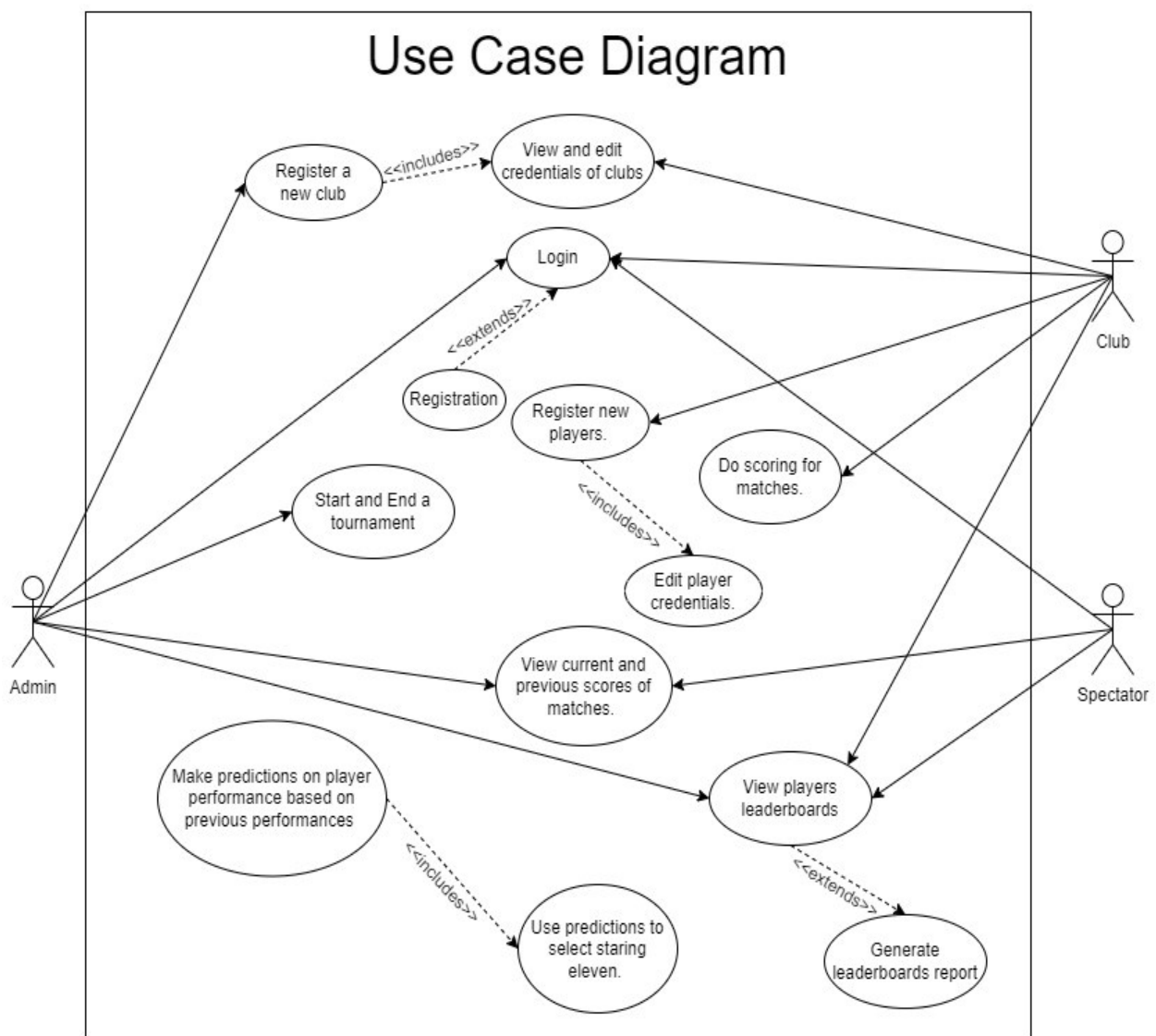


Figure 4. 2: Use Case Diagram

Table 1: Use Case Scenario 1

Use Case	Values
Actor	Admin
Pre - Condition	Must be logged in to the system.
Post - Condition	All input fields should have appropriate data before clicking the predict button or before the prediction is done.
Alternate Condition	No input field should be empty otherwise the user will be forced to fill it in first.

Table 2: Use Case Scenario 2

Use Case	Prediction
Actor	Admin
Pre - Condition	No input field should be empty
Post - Condition	A prediction verdict of good, bad and average should be displayed
Alternate Condition	In case of invalid or inconsistent information the system should display an error message prompting the admin to correct the input before generating the prediction

Table 3: Use Case Scenario 3

Use Case	Report Generation
Actor	Admin
Pre - Condition	A prediction must have been made
Post - Condition	The report should be displayed in a presentable format such as a pdf
Alternate Condition	If there are errors in the prediction data, the system should provide an error message and prevent the generation of the report

4.3.3 Sequence Diagram

A sequence diagram illustrates the sequence of message between object and how those objects interact with each other. The diagram below shoes the requests made by each other user to retrieve or access data from the database through the user interface and the series of steps followed by the users. If the actions are not followed as evident from the diagram, then there could be issues in accessing what is needed by the user.

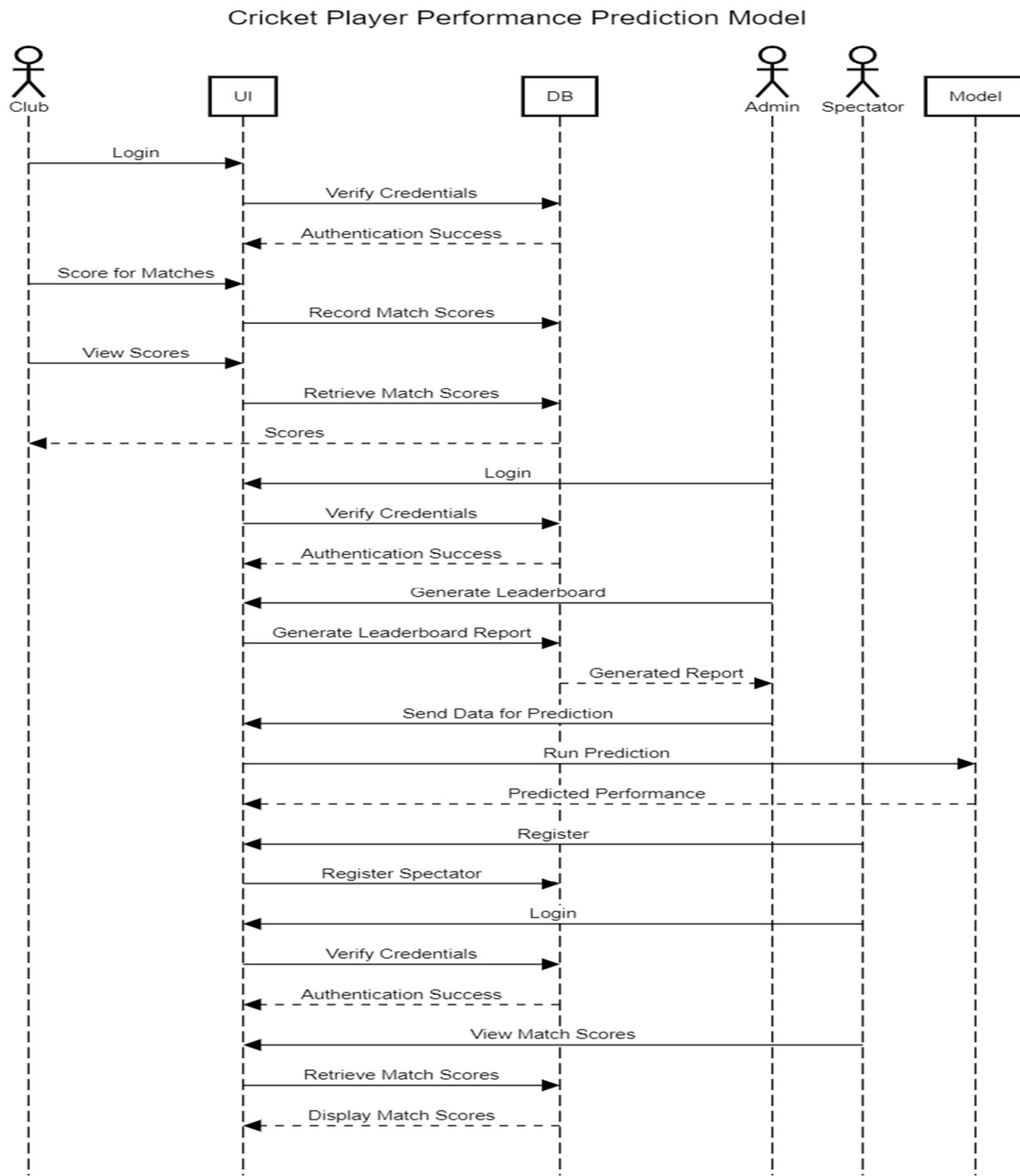


Figure 4. 31: Sequence Diagram

4.3.4 System Sequence Diagram

This diagram is used to display the interactions of between an external user and a system. It mostly focuses on the events exchanged between the system and the user to achieve a specific goal. As evident from the diagram and mentioned earlier that the admin is the main user of the system who gets the data of players performance from the leaderboards which is updated by the clubs and the admin feeds this data to the prediction model. So first the data needs to be cleaned meaning it should be made consistent to avoid any errors during the prediction. Then select the features needed for the prediction and choose a model technique that suits the data prediction model which is the decision tree or the random forest algorithm for the player performance prediction model. Use the algorithm to train the data and once the data is trained it can then successfully predict the performance and download the report.

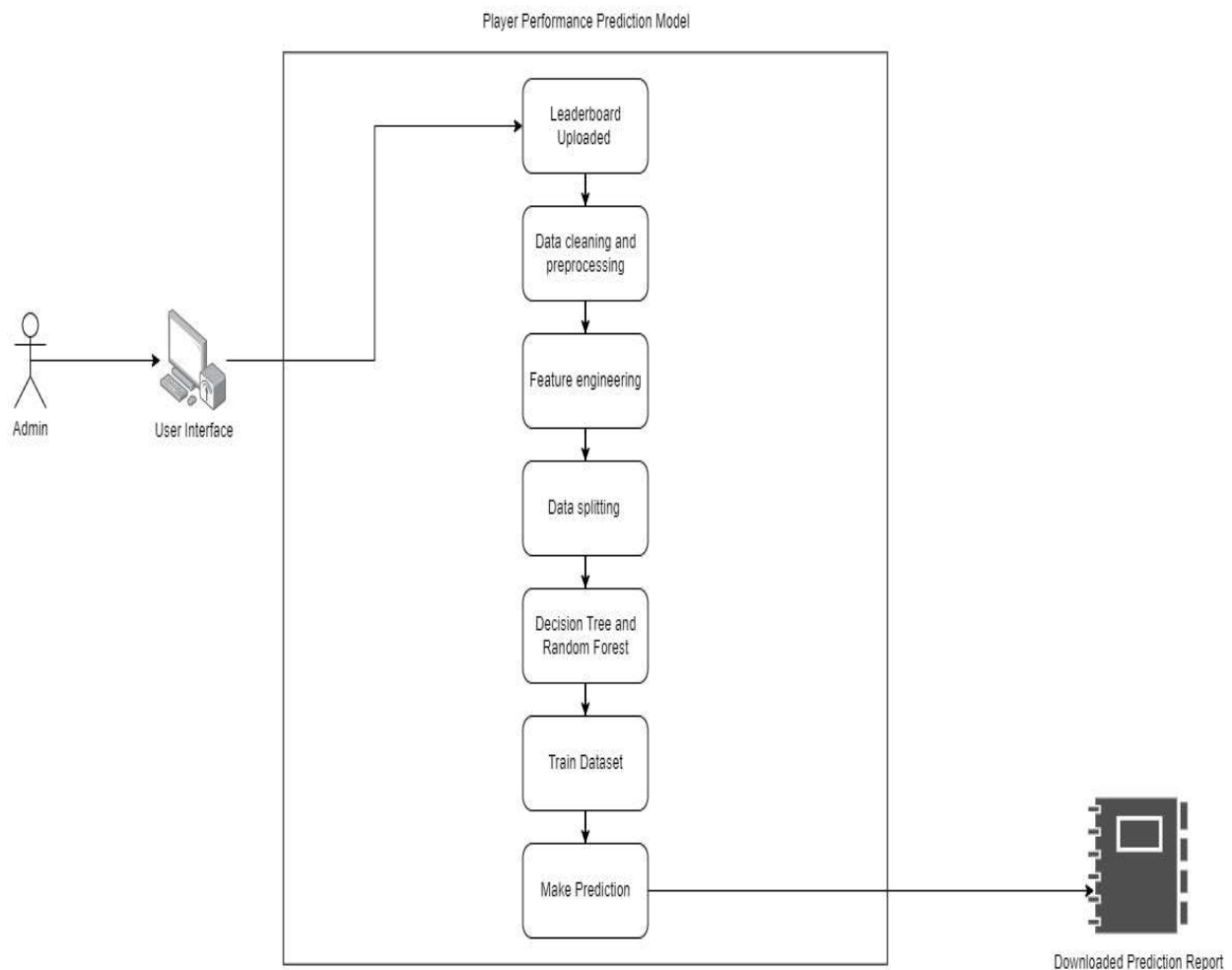


Figure 4. 42: System Sequence Diagram

4.3.5 Entity Relationship Diagram

This diagram helps to understand all the tables in the database and the relationships they carry with each other. As evident from the ER diagram below the system has nine entities represented in the rectangles. The rhombus defines the relationship between the entities, the oval represents the attributes the oval that has an underline is known to be the primary key. Moreover, the diagram also has several types of arrows pointing to entities that shows the type of relationship or cardinality of the relationship. The most used cardinalities are one to one, one to many and many to many. As evident from the diagram below one to one and one to many cardinalities are mostly used. An example of a one-to-one relationship is one user who has one role in the system either the admin, the club, or the spectator.

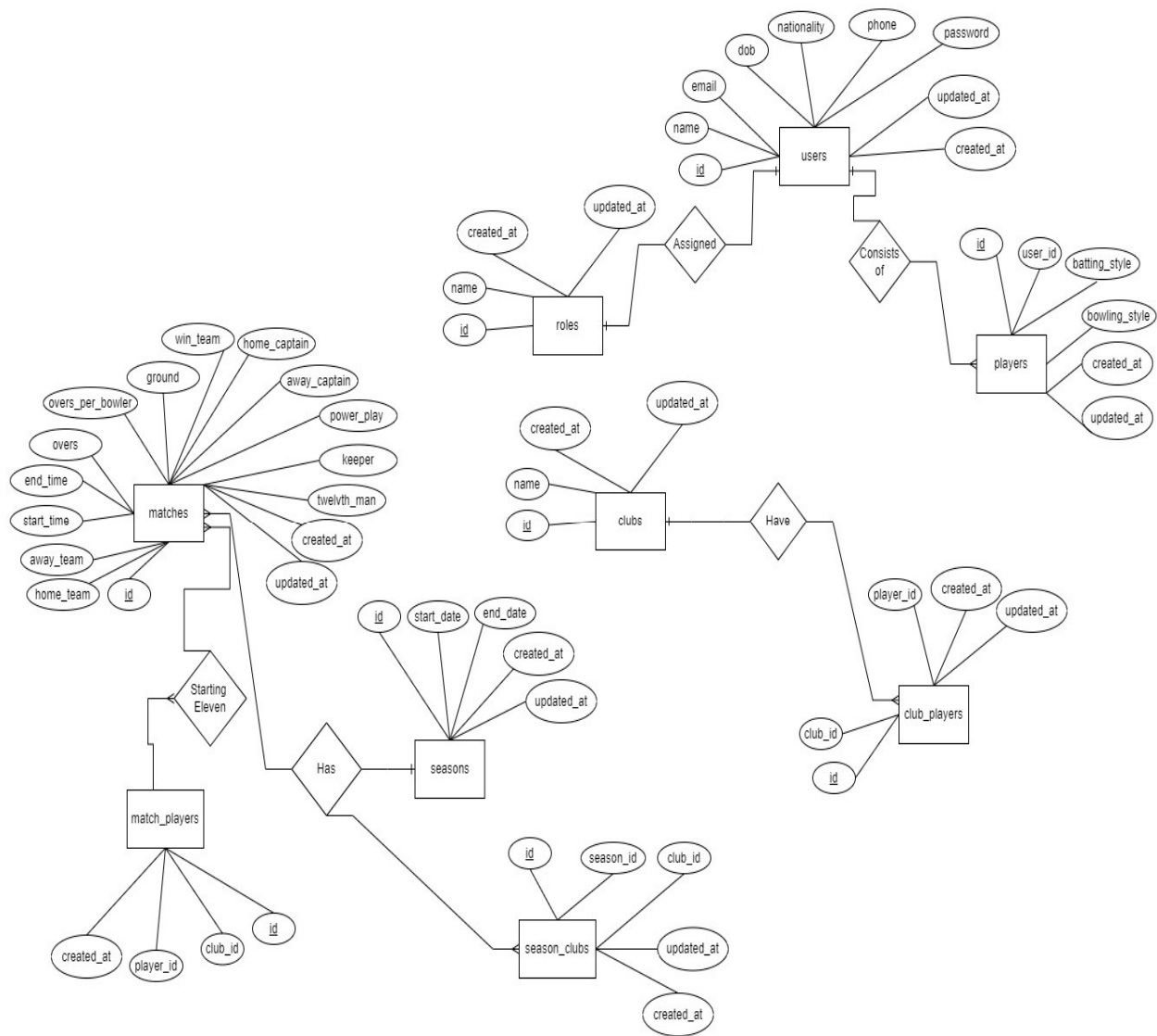


Figure 4. 5: Entity Relationship Diagram

4.3.6 Class Diagram

A class diagram illustrates the classes, entities, its methods, and the relationship amongst the classes. As evident from the diagram below the class diagram has a superclass or the parent class that is the user class that has subclasses such as the players, the match_players and the club_players. Just like the ER diagram and the Database schema this diagram also shows relationships based on their cardinalities. As evident from the diagram below the relationships are also shown with several types of arrows firstly to show a parent class there is an inheritance and pointing towards the user meaning that the user is the parent class, a composition arrow showing that the classes from which the arrows are coming out cannot exist without the class to which they are pointing at. For our case, a club can exist without players thus there is an aggregation relationship. For the composition relationship as evident that there is a relationship between the season and the clubs playing that season that is the season_clubs thus if the season is closed or deleted then the clubs playing in that season will also be removed.

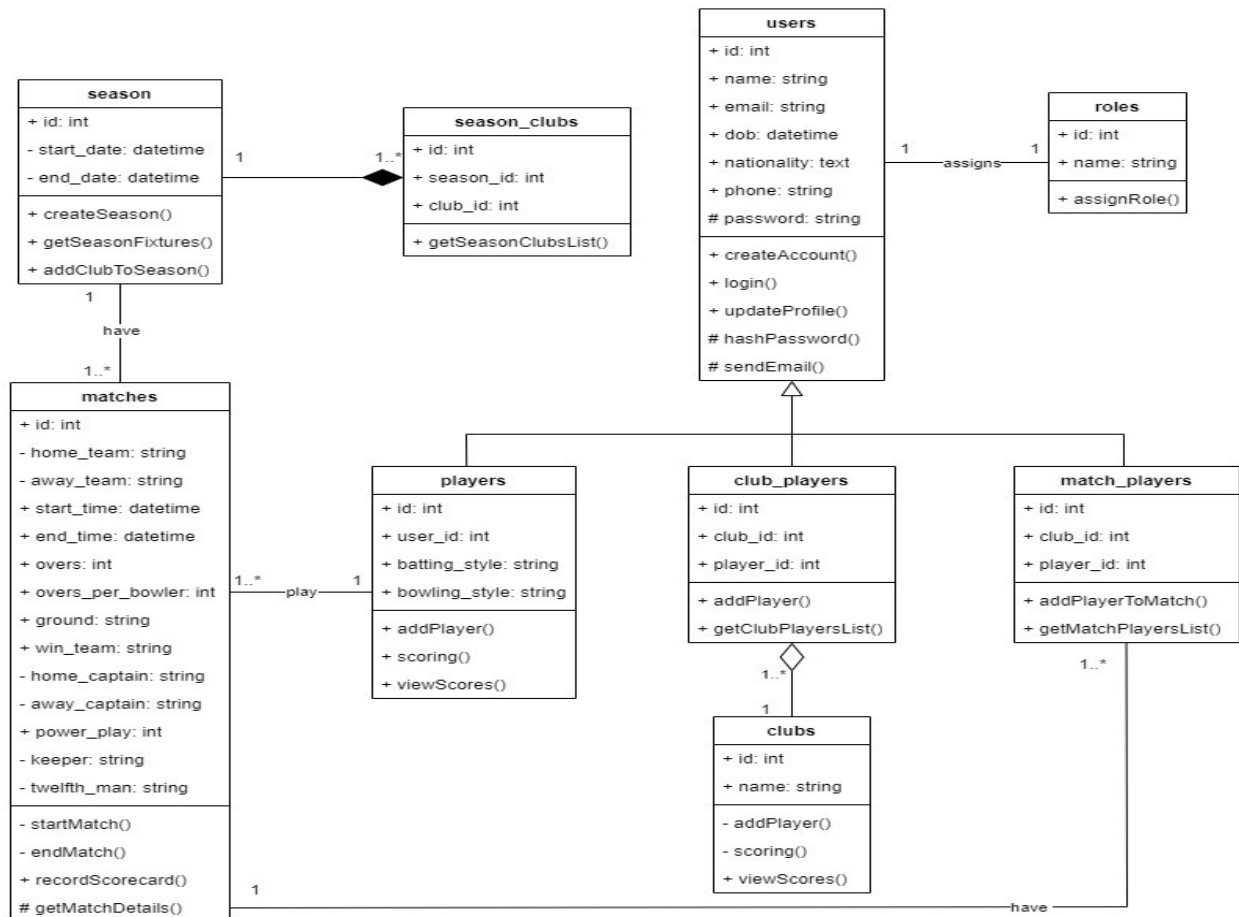


Figure 4. 6: Class Diagram

4.3.7 Activity Diagram

An activity diagram is used to illustrate the steps followed to meet the purpose of a use case. They are used to model all the steps undertaken in a system. For the prediction model the steps start by a login system which all the users have to go through to access their authorized interface and carry out their assigned roles. The rhombus is a checkpoint where the appropriate steps are taken to reach the desired output. The most activity is carried out by the admin who generates the leaderboards model and feeds it into the RoBERTa model which is a deep learning model used to help in predictive tasks and help to come up with a prediction for the players performance in the upcoming matches.

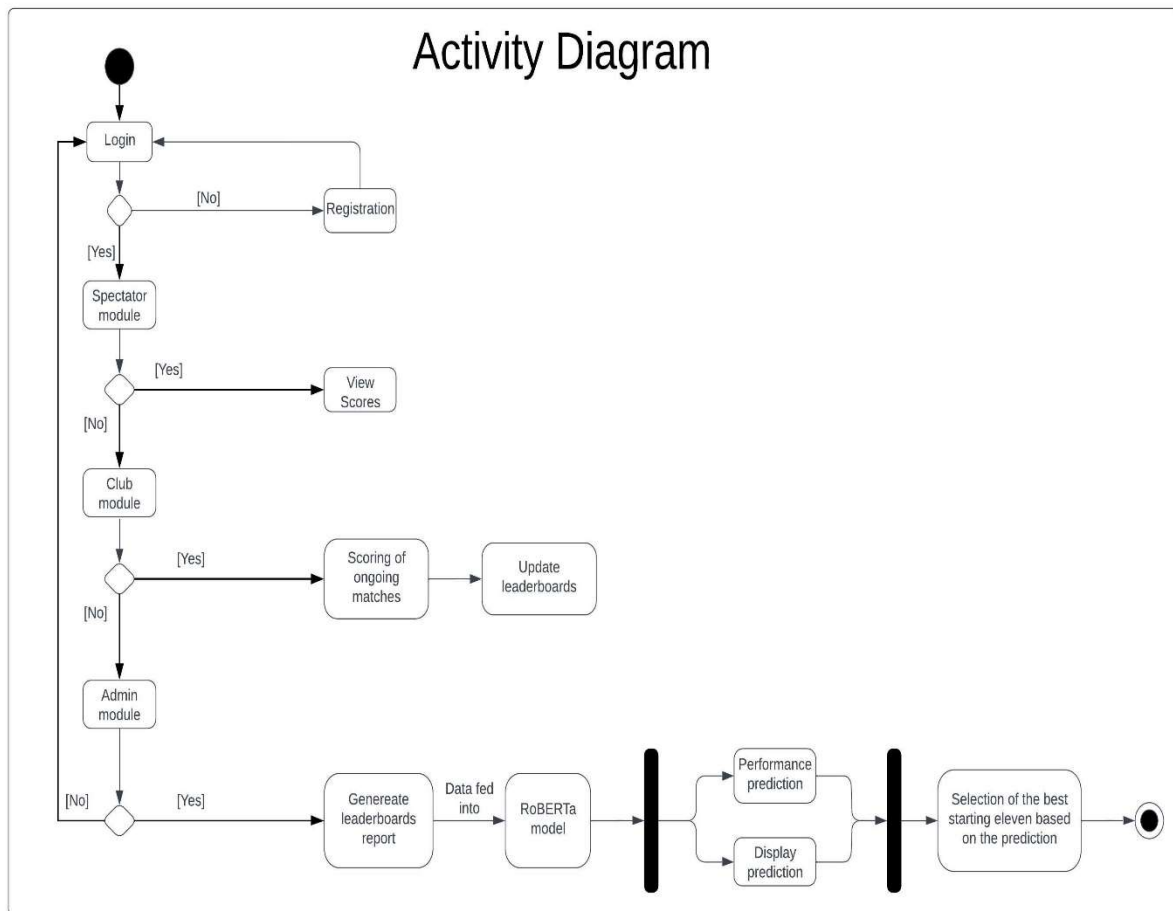


Figure 4. 7: Activity Diagram

4.3.8 Wireframes

These diagram helps with blueprints of how the system looks like so that the users and the developers can anticipate and have an early feel of what the actual system is going to look like.

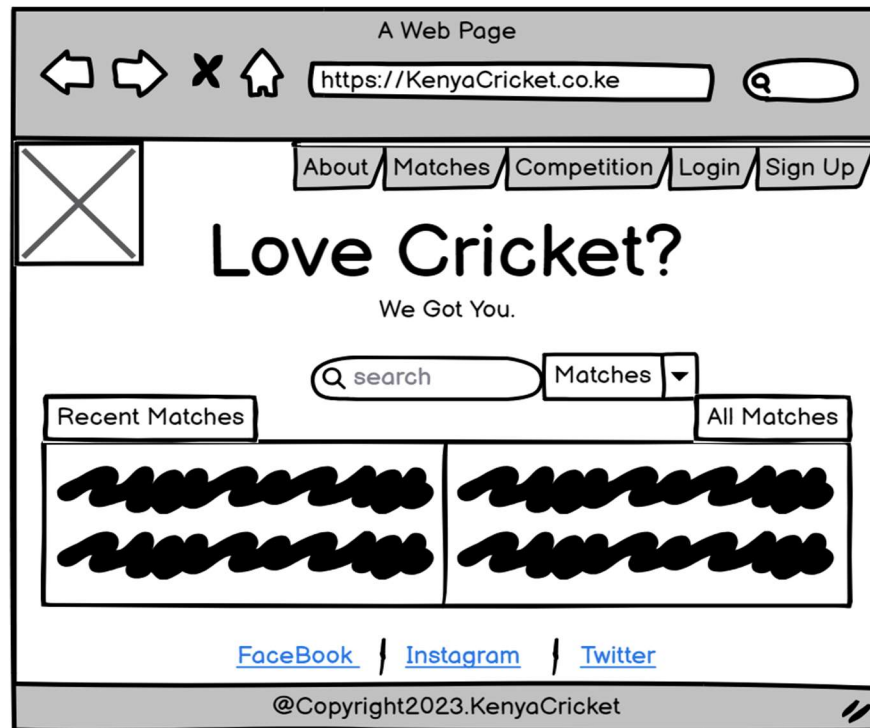


Figure 4. 8: Landing Page Wireframe

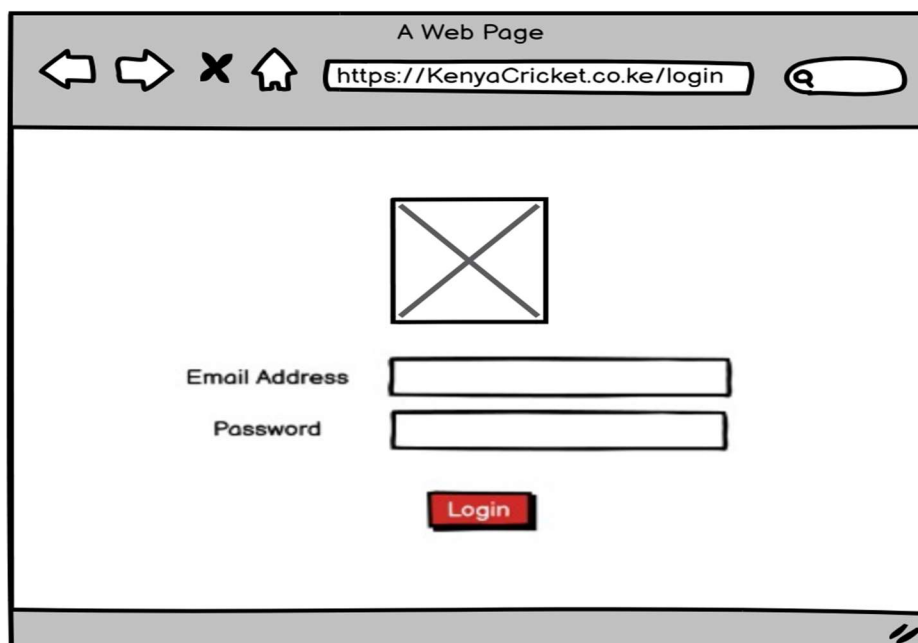


Figure 4. 9: Login Page Wireframe

A Web Page

https://KenyaCricket/scoring

Team A Team B

Team A

Team B

Batsman R, B, 4s, 6s

Bowler R, O, M, W

1 2 3

4 6 5+ END OVER ACTIONS

WIDE NO BALL BYE LEG BYE WICKET

Figure 4. 10: Scoring Page Wireframe

A Web Page

https://KenyaCricket/prediction

Player Prediction

Name:

Opposition:

Venue:

Match Type: 5 overs or 10 overs

Batting First or Bowling First: First Batting Or First Bowling

Toss:

Predict

Figure 4. 11: Prediction Model Wireframe