

## List of Figures

1	Use-case diagram . . . . .	3
2	Flow chart . . . . .	4
3	Login Page . . . . .	5
4	Admin Page (View all user data) . . . . .	5
5	Admin Page (Host a Match) . . . . .	5
6	Client Main page . . . . .	6
7	Client Bet page . . . . .	6
8	Client History Bet page . . . . .	6
9	Hall of fame . . . . .	7
10	Twitter announcing new match . . . . .	7
11	Twitter announcing result match . . . . .	7

## Contents

<b>I. Introduction</b>	<b>1</b>
<b>II. Implemented functionalities</b>	<b>1</b>
<b>III. Programs, Materials, Architecture</b>	<b>2</b>
Use-cases . . . . .	3
Flow chart . . . . .	4
<b>Appendix</b>	<b>5</b>
Login and Register View . . . . .	5
Administrator Page . . . . .	5
Client Page . . . . .	6
<b>IV. Conclusion and Future work</b>	<b>8</b>

## I. Introduction

The application that we had developed is a Football bet website with the most up-to-date matches, placing a bet on Football has never been so easy.

## II. Implemented functionalities

In this project, we have done all the required function as in the project description and also some extra features:

- Using data from Football API
- Using MVC architecture
- Administrator
  - The Administrator is responsible for the creation, modification, deletion of Bettors and Bookmakers
  - Allow Administrator to create a Bookmaker, Parieur and assign a Match for them
- Bookmaker
  - The Bookmaker is responsible for feeding the game into a match.
  - A Match is characterized by a name, a place, a list of participants
- Bettor
  - Can register an account
  - By default, everyone starts with 1000 Limcoins in their pocket.
  - The Bettor can view all proposed Matches and place Bets on all Matches that interest him in the limit of funds available
  - Changing the bet and see their history bet
  - Can search for a particular Match that interests him
- By the result of the match all the Bettors who have succeeded in their bet will receive their coin and vice versa
- A ranking of Bettors is available in the app.cation. It is made on the amount of Limcoins they have
- Advanced features:
  - Implemented REST API for Parieur
  - REST services
  - Twitter notification:

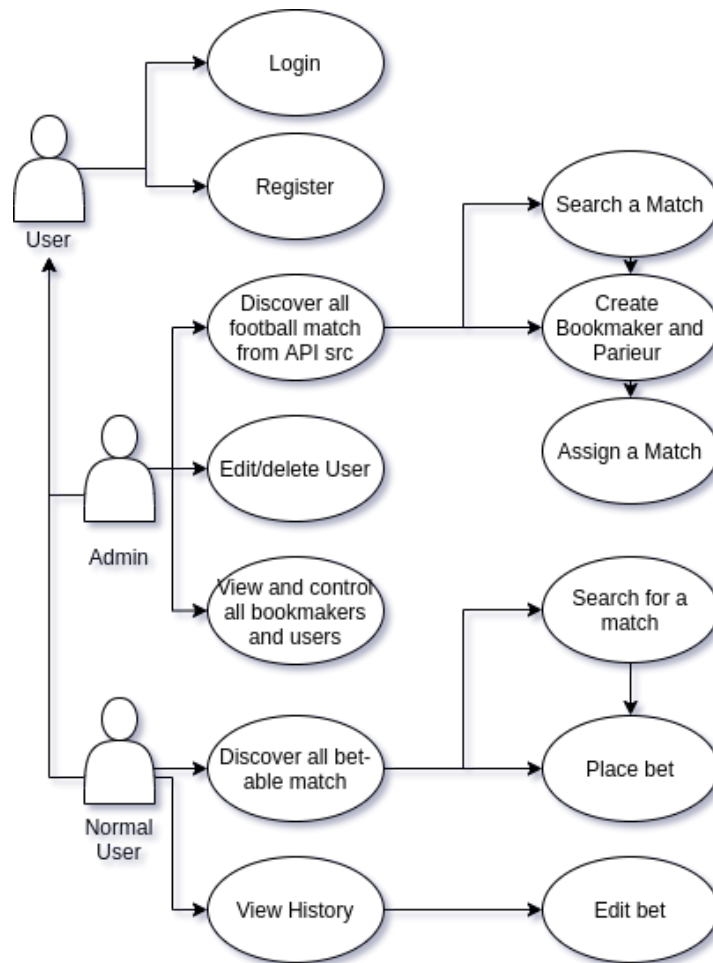
- \* Sending a message to user twitter account announcing there is a new Bet Match available
- \* Notify user the result of the match every week

### III. Programs, Materials, Architecture

In this project, we use IntelliJ with Java Platform, Enterprise Edition (Java EE) with GlassFish5 to build the web application, beside that, we also use some library for making an responsive website such as Bootstrap and NES css to quickly design and customize responsive for front-end part.

A clean codebase is always a pleasure to work with. A well organized codebase is easy to maintain, is robust, performs well, is testable and is self-documenting. In order to do that, the application will be written by using MVC - short for Model-View-Controller design pattern and also we use JSF alongside with javaEE.

## Use-cases



**Figure 1:** Use-case diagram

As the aim of this project is to create a football bet platform. The application has three main characters which is Normal User, Administrator and Bookmaker.

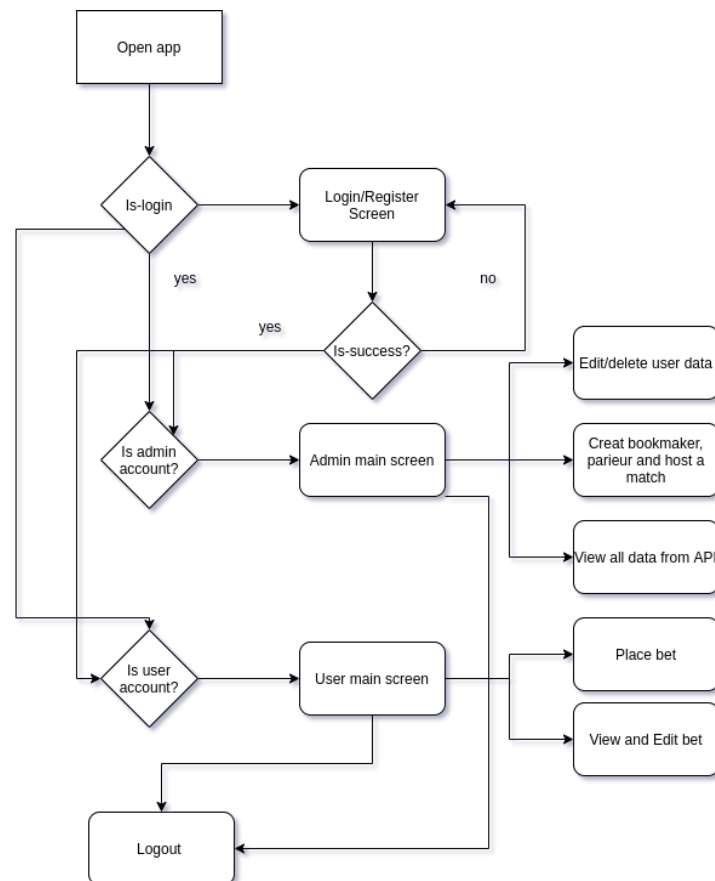
The administrator has the role to create a bookmaker and assign them with a real football match ( which is fetch from Football API) and also can handle basic user data information or delete user account.

The client can browse all the available match which was hosted by administrator and a book-maker and place a bet on it They also can edit their bet later.

The application service later will fetch the result of these match and update the user money based their bet result. And also, every time there is a new match which is available for bet, our

service application will send an message announcing to user through twitter. And also we use twitter notification to announce the match result which is an advance feature.

## Flow chart



**Figure 2:** Flow chart

## Appendix

### Login and Register View

**LocalBet**

**Login**

Username  
Enter Username\*

Password  
Enter Password\*

**Register**

Username  
Enter Username\*

Full Name  
Enter your name\*

Password  
Enter Password\*

Confirm Password  
Enter Confirm Password\*

Twitter  
Enter Twitter Username\*

Figure 3: Login Page

### Administrator Page

**Users Table**

User ID	Name	Money	
1	huy	13000	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

**Bookmakers**

**Bookmaker Table**

User ID	Username	Match ID
3	bookmaker1	303937
4	bookmaker2	303945
5	bookmaker3	303946
6	bookmaker4	303938

Figure 4: Admin Page (View all user data)

**Game list**

Username for Bookmaker  
bookmaker4

Password  
\*\*\*\*\*

**Available matches**

Search...

303937	Wolverhampton Wanderers FC	West Bromwich Albion FC	16-01-2021 12:30	<input type="button" value="Select"/>
303945	West Ham United FC	Burnley FC	16-01-2021 15:00	<input type="button" value="Select"/>

Figure 5: Admin Page (Host a Match)

## Client Page



Figure 6: Client Main page

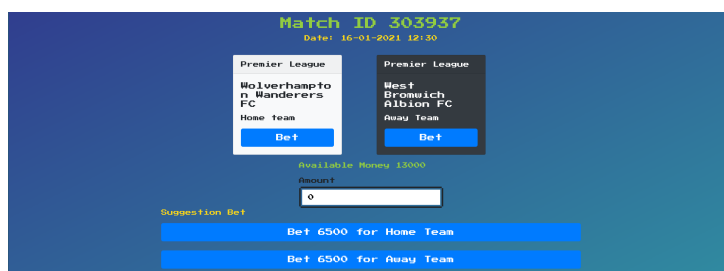


Figure 7: Client Bet page

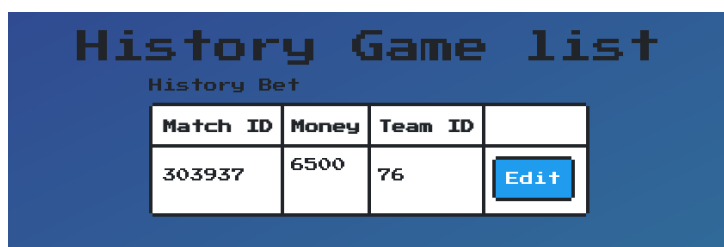


Figure 8: Client History Bet page





A screenshot of a 'Hall of Fame' interface with a dark blue background. The title 'Hall of Fame' is in a large, pixelated font, with 'Players Ranking' below it in a smaller, white font. A table with two columns, 'User Name' and 'Money', is displayed. The table has a white border and contains two rows of data.

User Name	Money
huy	6500
hoang	1000

Figure 9: Hall of fame

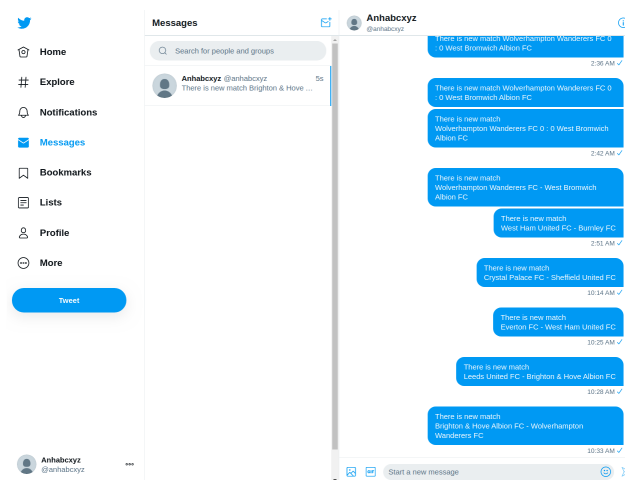


Figure 10: Twitter announcing new match

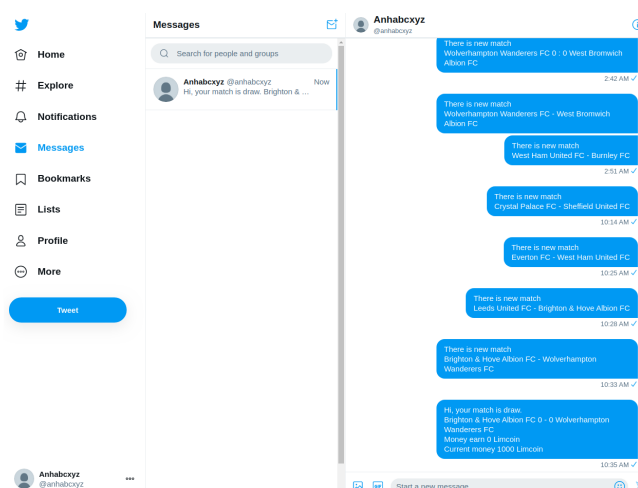


Figure 11: Twitter announcing result match

## IV. Conclusion and Future work

In this project, we have fulfilled the standard requirements and also advanced feature set out from the beginning in a short amount of time. We have presented the details of the implementation and also our special features. In the future work, we want to add Authentication security to strengthen our site , also improving the twitter notification feature and improving the web performance and scalability.