Project for 2nd semester of Bachelor of Information Technology

**TransferMaster**

****

**Durga Budha (321010)**

**Prajjwal Maharjan (321021)**

**Rabin Pulami Magar (321026)**

**KIST College of Information Technology**

**Faculty of Science and Technology**

**Purbanchal University,Nepal**

**February 14, 2025**

**STUDENT’S DECLARATION**

We hereby declare that the project in the report entitled

**TransferMaster**

Submitted in partial fulfillment of the requirement for the award of Bachelor of Information Technology (BIT) of the Purbanchal University is our original work and has not been submitted for award of any other degree or other similar title or prize.

|  |  |  |  |
| --- | --- | --- | --- |
| S.N. | Name | Registration No. | Symbol No. |
| 1. | Durga Budha | 058-3-2-02419-2023 | 321010 |
| 2. | Prajjwal Maharjan | 058-3-2-02422-2023 | 321021 |
| 3. | Rabin Pulami Magar | 058-3-2-02436-2023 | 321026 |

**RECOMMENDATION**

This is to certify that this project entitled **Transfer Master** submitted by **Durga Budha, Prajjwal Maharjan and Rabin Pulami Magar,** in partial fulfillment of the requirements of the project of Bachelors of Information Technology awarded by Purbanchal University, has been completed under my supervision. I recommend the same for acceptance by Purbanchal University.

Deepak Khadka

BIT Coordinator

KIST College of Information Technology

14 February, 2025

**CERTIFICATE**

This project entitle **TransferMaster** prepared and submitted by **Durga Budha, Prajjwal Maharjan and Rabin Pulami Magar** has been examined by us and is accepted as a part of Bachelor of Information Technology by Purbanchal University.

**Sailendra Basnet**

**External Examiner Signature Date signed**

**Deepak Khadka**

**Supervisor Signature Date signed**

**Samir Sharma**

**Principal Signature Date signed**

**ACKNOWLEDGEMENT**

We are deeply grateful to everyone who has supported us in completing this project.

At the very first, a big thank you to our **BIT Coordinator/Mentor, Deepak Khadka** for continuous support, encouragement, insightful feedback and valuable guidance throughout this project.

We are extremely grateful to our Principal, **Mr. Samir Sharma** as well as our college, **KIST College of Information Technology** for giving us this golden opportunity and for providing the necessary facilities and resources that were crucial foe this project.

We would like to take this opportunity to express our gratitude to all of our group members **Durga Budha, Prajjwal Maharjan and Rabin Pulami Magar**. The project would not have been successful without their cooperation and inputs.

We are also thankful to our colleagues and friends, for their constant support, constructive discussions and for always being there to offer help when needed. We are profoundly grateful to our family for their unconditional love, patience, and support.

Sincerely

Durga Budha (321010)

Prajjwal Maharjan (321021)

Rabin Pulami Magar (321026)

February 14,2025

**ABSTRACT**

TransferMaster is an advanced system designed to simplify player transfers, contract management, and financial operations in football. It efficiently stores player information, transfer history, and club records using a structured database. By integrating AI, the system predicts player market values and provides real-time transfer updates, ensuring clubs make informed decisions.

Security is a top priority, with secure login mechanisms safeguarding sensitive data. The system's user-friendly interface allows easy access to transfer records, contract details, and financial insights. Future developments include a mobile and web version to enhance accessibility and usability for club managers, scouts, and agents.

By optimizing data management, transfer tracking, and financial security, TransferMaster makes football transfers faster, safer, and more efficient.

**Keywords:**

* **Player Transfers**
* **Contracts**
* **AI Predictions**
* **Real-time Updates**
* **Secure Login**
* **Football Management**
* **Data Storage**
* **Transfer History**
* **Financial Management**

**TABLE OF CONTENTS**

[1. BACKGROUND viii](#_Toc190266750)

[**1.1 Introduction** viii](#_Toc190266751)

[**1.2. Objectives** viii](#_Toc190266752)

[**1.3. Features** ix](#_Toc190266753)

[**1.4. Future Implementation** ix](#_Toc190266754)

[**1.5. Introduction to software/language used** x](#_Toc190266755)

[2. SYSTEM RECOMMENDATION x](#_Toc190266756)

[3. FUTURE PROSPECT xi](#_Toc190266757)

[**3.1. Solution to the above problem** xi](#_Toc190266758)

[4. SYSTEM DESIGN xii](#_Toc190266759)

[**4.1. Algorithm** xii](#_Toc190266760)

[**4.2. Flowchart** xxii](#_Toc190266761)

[5. SOURCE CODE xxxiii](#_Toc190266762)

[6. CONCLUSION xxxiv](#_Toc190266763)

[7. REFERENCE xxxv](#_Toc190266764)

List of Figures:

4.1[. Main Menu Flowchart ix](#_Toc190269144)

[**4.2. Admin Menu Flowchart** ix](#_Toc190269145)

[**4.3. Registration Flowchart** x](#_Toc190269146)

[**4.4. User Login Process Flowchart** x](#_Toc190269147)

[**4.5. Player Mangagement Process Flowchart** xi](#_Toc190269148)

[**4.6. Client Management Process Flowchart** xi](#_Toc190269149)

[4.7. Password Management Process Flowchart xii](#_Toc190269150)

[4.8. Player Transfer Process Flowchart xiii](#_Toc190269151)

[**4.9. View Player Rating and Transfer Fee Flowchart** xiv](#_Toc190269152)

[4.10. Client Menu Flowchart xiv](#_Toc190269153)

[**4.11. Login Menu Flowchart** xiv](#_Toc190269154)

# **1. BACKGROUND**

## **1.1 Introduction**

TransferMaster predicts a football player's ratings and transfer values with a unique and independently developed algorithm based on performance statistics and market trends. Using statistical analysis and machine learning methodologies, this tool offers reliable assessments of players' performance and their market value. The platform aims to help football clubs, scouts, and enthusiasts make informed decisions about player transfers. This tool provides an objective and systematic way to analyze player potential, thus minimizing reliance on subjective interpretations when making strategic decisions in football management.

## **1.2. Objectives**

The main objective of this project are as follows:

* To provide accurate player ratings based on statistical analysis.
* To predict transfer values aligned with market trends.
* To assist football clubs and enthusiasts in making informed transfer decisions.
* To create a user-friendly platform for data-driven insights.

## **1.3. Features**

* **Data-Driven Predictions:** Utilizes extensive performance statistics and market trends to generate accurate player ratings and transfer values.
* **Performance Analysis:** Analyzes player performance metrics to provide a comprehensive evaluation of player capabilities.
* **Market Trend Analysis:** Monitors market trends to adjust transfer values, reflecting the current market conditions.
* **User-Friendly Interface:** Offers an intuitive interface for users to input data and receive predictions effortlessly.

## **1.4. Future Implementation**

TransferMaster will focus on enhancing accuracy and usability through AI and machine learning for improved player rating and transfer predictions. Real-time data integration, expansion to multiple leagues, and advanced data visualization will provide more insightful analysis. A web and mobile application will make the tool more accessible, while injury prediction and blockchain-based secure transfers will add new dimensions to player valuation. AI-driven contract negotiation and fan engagement features will further broaden its impact. With multi-language support and global expansion, TransferMaster aims to become a comprehensive football scouting and transfer analysis platform.

## **1.5. Introduction to software/language used**

C++ is a general-purpose, object-oriented programming language developed by Bjarne Stroustrup in 1985 at Bell Labs. It is an extension of the C programming language, providing additional features such as object-oriented programming, better type checking, and data abstraction. C++ is widely used for system software, application software, game development, and competitive programming due to its speed, flexibility, and extensive library support. It suitable for a wide range of applications, from system-level programming to game development and large-scale software projects. Some main topic dealt in C++ programming are:

* Variable and constant
* Array and String
* Class and objects
* Vector and ctime
* File handling

# **2. SYSTEM RECOMMENDATION**

The minimum requirements for the system are:

**2.1. HARDWARE:**

* Color Monitor.
* 8 MB of RAM or more.
* PC with Intel i5 or the latest.
* Hard disk with at least 50MB of free space.

**2.2. SOFTWARE:**

* OS Windows ( Windows XP, Windows 7, Windows 8, Windows 10).

# **3. FUTURE PROSPECT**

* **Data Management Issues** – Difficulty in storing and retrieving large player records.
* **Real-Time Updates** – Challenges in integrating live transfer data via APIs.
* **Security Concerns** – Need for secure handling of contracts and transactions.
* **Scalability Issues** – Managing multiple clubs and transfers efficiently.
* **Limited User Interface** – Console-based UI may not be user-friendly.

## **3.1. Solution to the above problem**

SQL databases can be used for better data storage, integrating football APIs for real-time transfer updates, and implementing encryption and user authentication for security. To improve scalability, optimized data structures and multithreading will be used. Upgrading to a Graphical User Interface (GUI) will enhance usability, while machine learning can help predict player values. Finally, using official data sources or web scraping will ensure accurate player and club information. These will make the system faster, secure, and easier to use.

# **4. SYSTEM DESIGN**

## **4.1. Algorithm**

Algorithm is a process of set of rules to be followed in calculation or other problem-solving operations, especially by a computer.

1. **Registration**

Step 1: Start

Step 2: Prompt user for Name, Password, Agent Name

Step 3: Validate the password:

Ensure it contains at least 8 characters, an uppercase letter, a lowercase letter, a digit and a special character

Step 4: Store user information in Club.txt

Step 5: Display confirmation message

Step 6: End

1. **User login**

Step 1: Start

Step 2: Request ID and Password

Step 3: Check if inputs are not empty

Step 4: Retrieve stored password from Club.txt

Step 5: Compare entered password with stored password

Step 6: If match found, authenticate user and proceed

If mismatch, display error message

Step 7: End

1. **Admin Menu Process**

Step 1: Start

Step 2: Display Admin Menu options:

a) Player Management

b) Club Management

c) Change Admin Password

d) View Game Rules

e) View Transfer Guidelines

f) Back to Main Menu

Step 3: Prompt Admin for selection

Step 4: Read input

Step 5: Handle selected option

If Admin chooses an invalid option, display error and retry

Repeat Steps 3-5 until Admin exits

Step 6: End

1. **Player Management Process**

Step 1: Start

Step 2: Display Player Management options:

a) Add Player

b) Edit Player Details

c) Delete Player

d) Search Player

e) Update Player Stats and Status

f) View All Players

g) Release Player from Club

h) Back to Admin Menu

Step 3: Read user input

Step 4: Handle selected option

If Admin selects an invalid option, display an error

Repeat Steps 3-4 until Admin exits

Step 5: End

1. **Client (Club) Management Process**

Step 1: Start

Step 2: Display Client Management options:

a) Approve Club

b) Edit Club

c) Delete Club

d) Search Club

e) Update Balance of Club

f) View All Clubs

g) Back to Admin Menu

Step 3: Read user input

Step 4: Handle selected option

If invalid, display an error and retry

Repeat Steps 3-4 until Admin exits

Step 5: End

1. **Change Admins Password Process**

Step 1: Start

Step 2: Prompt Admin to enter current password

Step 3: Verify the entered password

Step 4: If password is correct, prompt for a new password

Ensure it meets security criteria

Step 5: Update stored password in adminPassword.txt

Step 6: If password is incorrect, display an error message and retry

Step 7: End

1. **Initiating Player Transfer Process**

Step 1: Start

Step 2: Display list of available players

Step 3: Prompt user to enter Player ID

Step 4: Check if the Player ID exists and player is active

Step 5: Validate the club's balance

If sufficient funds, deduct transfer fee and complete the transfer

If insufficient funds, display an error message

Step 6: Update contract details and player status

Step 7: Save changes to file

Step 8: Display confirmation message

Step 9: End

1. **View Player Ratings & Transfer Fee Process**

Step 1: Start

Step 2: Fetch and calculate player ratings using:

Goals, Assists, Matches, Tackles, Key Passes, Injuries

Step 3: Calculate transfer fee using:

Rating, goals, assists, matches, injuries, and age

Step 4: Display list of players with ratings and fees

Step 5: End

1. **Manage Squad (Roster) Process**

Step 1: Start

Step 2: Display Squad Management Menu:

a) Add Player to Squad

b) View Squad Players

c) Remove Player from Squad

d) Back to Club Menu

Step 3: Read user input

Step 4: Perform selected action

If invalid, display an error and retry

Repeat Steps 3-4 until user exits

Step 5: End

1. **Displaying a Loading Animation**

Step 1: Start

Step 2: Initialize Console Handle

Step 3: Set Initial Text Color

Step 4: Print Initial Loading Text

Step 5: Display Dots in a Loop

Step 5: Print New Line

Step 6: Clear the Screen

Step 7: End

1. **Display About Us**

Step 1: Start

Step 2: Clear the screen

Step 3: Set text color

Step 4: Print program introduction and mission statement

Step 5: List features and benefits for clubs and administrators

Step 6: Display developer information

Step 7: Reset text color

Step 8: End

1. **Display the Privacy Policy**

Step 1: Start

Step 2: Clear the screen

Step 3: Set text color for headers

Step 4: Print Privacy Policy header

Step 5: Print security measures

Step 6: Explain encryption techniques

Step 7: Print commitment to user data protection

Step 8: End

1. **Player Information in a Tabular Format**

Step 1: Start

Step 2: Retrieve player details from class attributes (ID, Name, Age, etc.)

Step 3: Determine the maximum column width for proper alignment

Step 4: Print the table header with labels ("Attribute" and "Value")

Step 5: Print a separator line to distinguish the header from the data

Step 6: For each player attribute:

1. Print the attribute name in the first column
2. Print the corresponding value in the second column, aligned properly

Step 7: Ensure consistent spacing for a structured display

Step 8: End

1. **Player Rating Calculation**

Step 1: Start

Step 2: Retrieve player statistics: goals, assists, matches, tackles won, key passes, injuries, and age

Step 3: Compute the initial rating:

1. Multiply goals by 0.4
2. Multiply assists by 0.3
3. Multiply matches by 0.1
4. Multiply tackles won by 0.1
5. Multiply key passes by 0.1

Step 4: Apply deductions:

1. Subtract injuries multiplied by 0.05
2. Subtract age multiplied by 0.02

Step 5: Ensure the rating is within the range 0 to 100:

1. If rating is greater than 100, set it to 100
2. If rating is less than 0, set it to 0

Step 6: Return the final rating

Step 7: End

1. **Transfer Fee Calculation**

Step 1: Start

Step 2: Retrieve the player’s rating from calculateRating()

Step 3: Retrieve player statistics: goals, assists, matches, tackles won, key passes, injuries, and age

Step 4: Compute the transfer fee:

1. Multiply rating by 1,200,000
2. Add contributions from goals, assists, matches, tackles won, and key passes multiplied by 60,000
3. Subtract the impact of injuries and age, reducing the fee accordingly

Step 5: If the transfer fee is negative, set it to zero

Step 6: Return the final transfer fee

Step 7: End

1. **Games Rules**

Step 1: Start

Step 2: Set the display color to yellow

Step 3: Display the heading "TransferMaster - Game Rules"

Step 4: Display the following game rules:

1. A football match is played by two teams, each consisting of no more than 11 players.
2. The match is played on a rectangular field with a goal at each end.
3. The objective is to score by getting the ball into the opposing goal.
4. The match is won by the team that scores the most goals.
5. Players can be assigned to different roles such as Goalkeeper, Defender, Midfielder, Forward, or Striker.

Step 5: Reset the display color

Step 6: Display "Press any key to return..."

Step 7: Wait for the user to press any key (using \_getch())

Step 8: End

1. **Client (Club) Menu Process**

Step 1: Start

Step 2: Display Client Menu options:

    a) Register

    b) Login

    c) Change Password

    d) Forgot Password

    e) Check Login Status

    f) Back to Main Menu

Step 3: Read user input

Step 4: Handle selected option:

    If invalid, display an error and retry

    If valid, execute the corresponding function

Repeat Steps 3-4 until the user selects "Back to Main Menu"

Step 5: End

1. **Club Login Menu Process**

Step 1: Start

Step 2: Display Club Login Menu options:

    a) Update Club Information

    b) Squad Overview

    c) Initiate Transfer

    d) View Player's Rating and Transfer Fee

    e) View Transfer Guidelines

    f) View Game Rules

    g) View Club Information

    h) Back to Main Menu

Step 3: Read user input

Step 4: Handle selected option:

    If invalid, display an error and retry

    If valid, execute the corresponding function

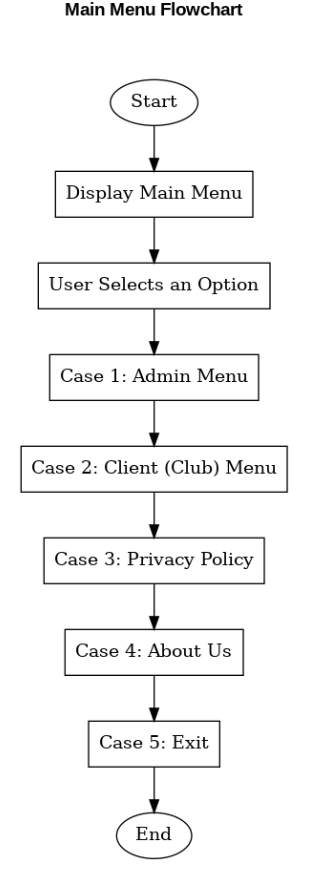
Repeat Steps 3-4 until the user selects "Back to Main Menu"

Step 5: End

## **4.2. Flowchart**

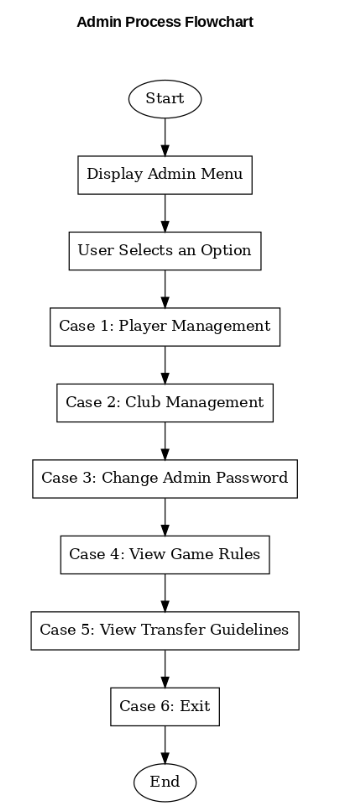
A flowchart is a visual representation of the steps involved in a process, showing the sequence of operations, decisions, and interactions.

1. **Main Menu**



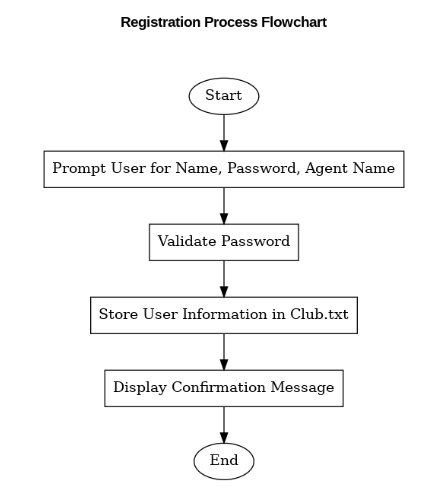
**Fig 4.1: Main Menu Flowchart**

1. **Admin Menu**



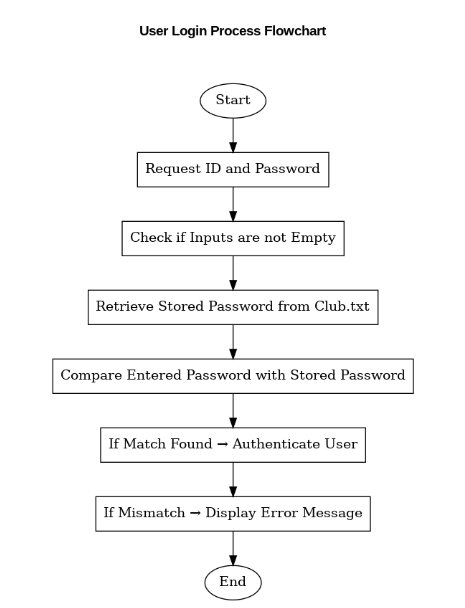
**Fig 4.2: Admin Menu Flowchart**

1. **Registration**

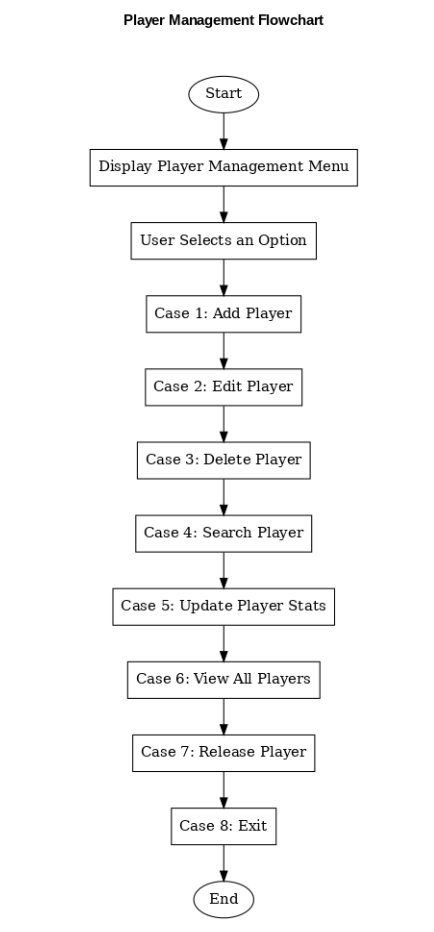


**Fig 4.3: Registration Flowchart**

1. **User Login Process**

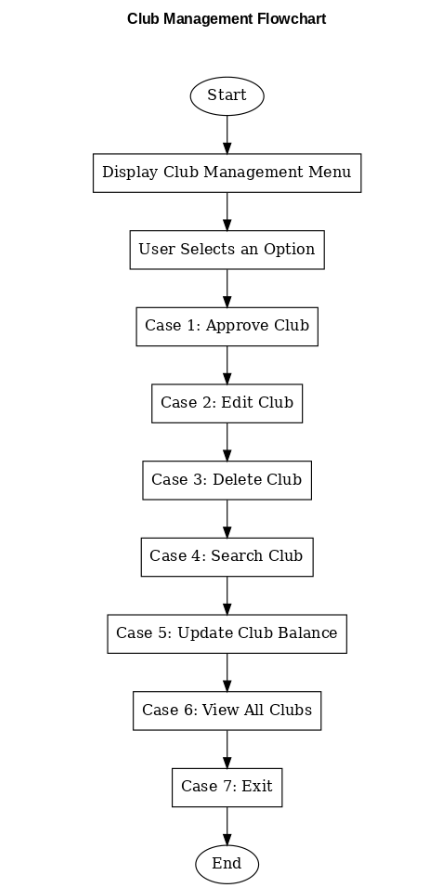


**Fig 4.4: User Login Flowchart**

**5. Player Management Process**

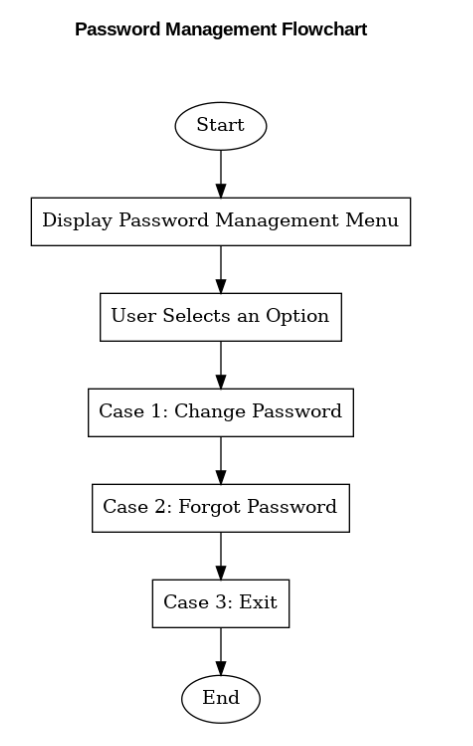
**Fig 4.5: Player Management Process Flowchart**

**6. Client (Club) Management Process**



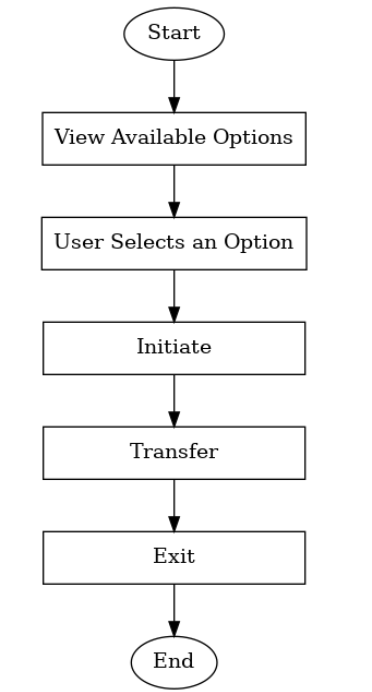
**Fig 4.6: Client Management Process Flowchart**

**7. Password Management Process**



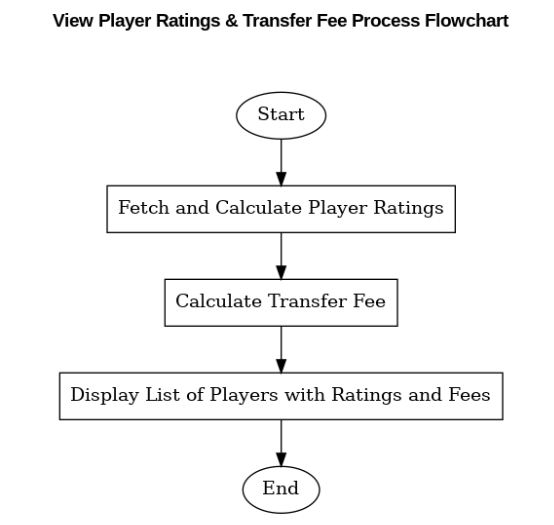
**Fig 4.7: Password Management Process Flowchart**

**8. Player Transfer Process**



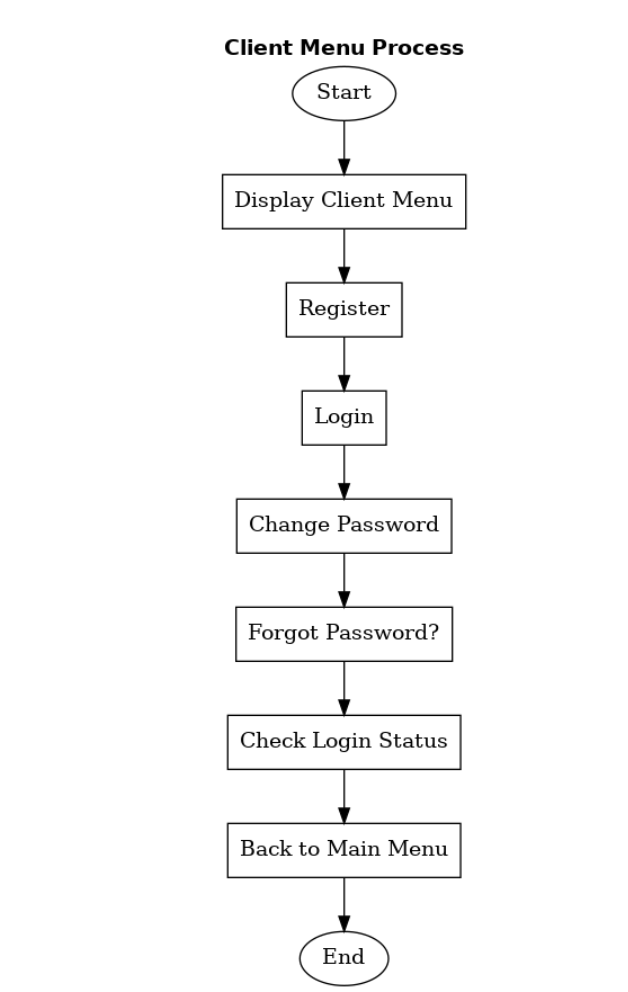
**Fig 4.8: Player Transfer Process Flowchart**

**9. View Player Rating and Transfer Fee**



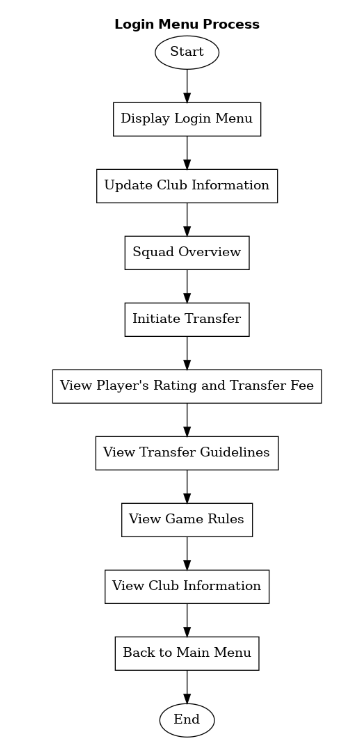
**Fig 4.9: View Player Rating and Transfer Fee Flowchart**

**10. Client Menu**



**Fig 4.10: Client Menu Flowchart**

**11. Login Menu**



**Fig 4.11: Login Menu Flowchart**

# **5. SOURCE CODE**

The source code for this project is available on **GitHub**. You can access the complete repository, review the code, contribute to the project, or report any issues directly through our **GitHub** page. Visit our GitHub Repository:

**https://github.com/Prajjwal-dev/TransferMaster**

# **6. CONCLUSION**

TransferMaster provides an efficient way to manage player transfers, contracts, and financial transactions between football clubs. AI improves in player value prediction, and databases are used to securely store player information. Clubs are kept informed of transactions in real time, and sensitive data is protected by secure login systems. It will be easier to use with a user-friendly interface (GUI), and accessibility can be enhanced with a mobile or web version. Overall, this system offers a structured, data-driven, and modern approach to managing football transfers effectively.

# **7. REFERENCE**

1. Stroustrup, B. (2013). The C++ programming language (4th ed.). Addison-Wesley
2. GeeksforGeeks. (n.d.). Object-oriented programming with C++. GeeksforGeeks. <https://www.geeksforgeeks.org/object-oriented-programming-in-cpp/>
3. GeeksforGeeks. (n.d.). Changing text color in C++. GeeksforGeeks. <https://www.geeksforgeeks.org/colorizing-text-and-console-background-in-c/>
4. Cplusplus.com. (n.d.). ctime library in C++. Cplusplus.com. <https://cplusplus.com/reference/ctime/>
5. GeeksforGeeks. (n.d.). Printing '' in place of characters for reading passwords in C\*. GeeksforGeeks. <https://www.geeksforgeeks.org/print-in-place-of-characters-for-reading-passwords-in-c/>
6. GeeksforGeeks. (n.d.). Getter and setter methods in C++. GeeksforGeeks. <https://www.geeksforgeeks.org/write-getter-and-setter-methods-in-cpp/>
7. Programiz. (n.d.). File handling in C++. Programiz. <https://www.programiz.com/cpp-programming/file-handling>
8. GeeksforGeeks. (n.d.). For-each loop in C++. GeeksforGeeks. <https://www.geeksforgeeks.org/g-fact-40-foreach-in-c-and-java/>
9. GeeksforGeeks. (n.d.). Formatted input and output in C++. GeeksforGeeks. <https://www.geeksforgeeks.org/formatted-i-o-in-c/>
10. Transfermarkt. (n.d.). Nepal transfer market. Transfermarkt. <https://www.transfermarkt.com/nepal/startseite/verein/15350>
11. Transfermarkt. (n.d.). International transfer market. Transfermarkt. <https://www.transfermarkt.com/>
12. Cplusplus.com. (n.d.). cmath library in C++. Cplusplus.com. <https://cplusplus.com/reference/cmath/>
13. GeeksforGeeks. (n.d.). Vectors in C++. GeeksforGeeks. <https://www.geeksforgeeks.org/vector-in-cpp-stl/>
14. Aruneworld. (n.d.). Windows header files in C++. Aruneworld. <https://aruneworld.com/programming-language/c/c-library/c-library-windows-h/>
15. Cplusplus.com. (n.d.). Algorithm header file in C++. Cplusplus.com. <https://cplusplus.com/reference/algorithm/>
16. GeeksforGeeks. (n.d.). Regression models and AI models. GeeksforGeeks. <https://www.geeksforgeeks.org/regression-analysis-and-the-best-fitting-line-using-c/>