A blue and white logo

Description automatically generated

**COMSATS University Islamabad (CUI)**

**Project Proposal**

**“FPMS”**

**(Football Player Management System)**

**Salar Wasil** **SP23-BSE-041**

**BSE-3**

***Supervisor***

**Zahid Anwar**

***Bachelor of Science in Software Engineering(2023-2027)***

FPMS

(Football Player Management System)

**Project Description:**

FPMS is a menu-driven Football Management System created in C++ using Data Structures. FPMS includes 5 teams and has basic functionalities through which you can Add Players in Teams, Display Players of Specific Teams, Search Players, Display All the Players in All the teams, Delete Players, Edit Players info etc. The Data Structures used in the project are given below.

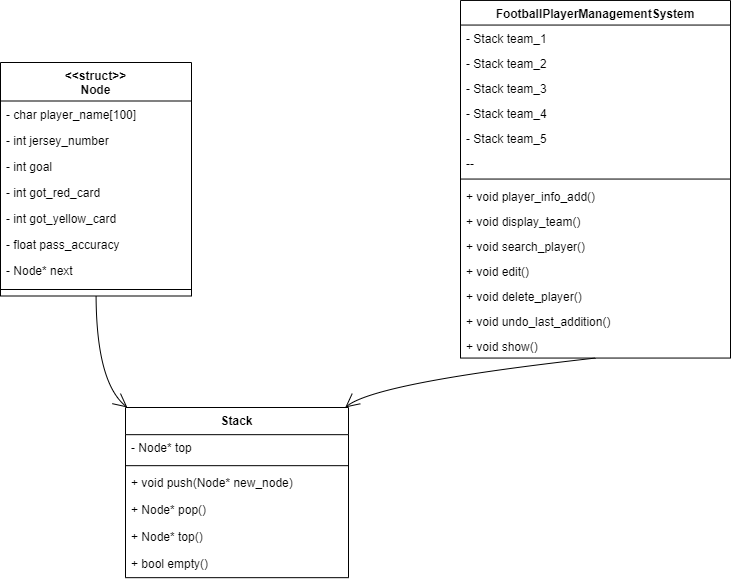
**Data Structure Used:**

* **Linked List**  
  The linked list that has been used in this project is Single linked list to traverse through the players in the team’s stack but using a Node\* next pointer to traverse forward through the linked list. Linked list has been used with stack in this project.
* **Stacks**This process makes use of the ADT stack provide by C++. Stack for this project has been implemented using linked list as in stack<NODE\*> team no. The top and pop features are mostly used in this project for the purpose of displaying and deleting and push feature has been used for adding the player information the team stack.

**Project Features:**

1. **Adding Player in Team:**Allows User of program to add player information in specific teams like name, jersey number, goals scored, red and yellow cards received, pass accuracy, and key pass status.
2. **Editing Player in Team:**Allows User of program to edit existing player details by specifying the team number and player identity. This is implemented using a temporary stack in which the specific team is stored and then data of the player is edited accordingly by the user.
3. **Displaying Players in all Teams:**Allows User of program to display list of players in all the teams, along with their performance statistics. This method is implemented using loops in which an array of temp\_stack[5] is taken in which all the 5 team stacks are added and then the data inside all the teams are displayed accordingly.
4. **Displaying Specific Team Players:**Allows User of program to display information of players in specific team. This is implemented using a while loop by storing the team in a temporary stack and then displaying accordingly.
5. **Searching Player:**Allows User of program to find player details based on their name and jersey number, searching through all the five teams respectively. For this functionality I have utilized the library of C language to use strcmp () to compare the player entered by user with players present in stack in order to search them.
6. **Deleting Player:**Allows User of program to delete the data of player present in a team and then update the stack accordingly. It uses team.pop() operation and stores the data in temporary node and then delete that node.
7. **Undoing Last Addition in Specific Team:**Allows User of program to undo the last addition of player from the stack by popping the top player from the Specific Team stack using team.pop() operation.

**Class Diagram:**

****