**Player Management System:**

| | |
|---|---|
| **Group Leader and ID:** | **Zain Abbas 22K-5028** |
| **Member 1 and ID:** | **Abdullah Arif 22K-5071** |

## Introduction:

The aim of this project is to create a comprehensive Player Record Management System that facilitates the efficient management of player information in various sports. The system is designed to simplify the process of recording, storing, and retrieving player details for better organization and accessibility.

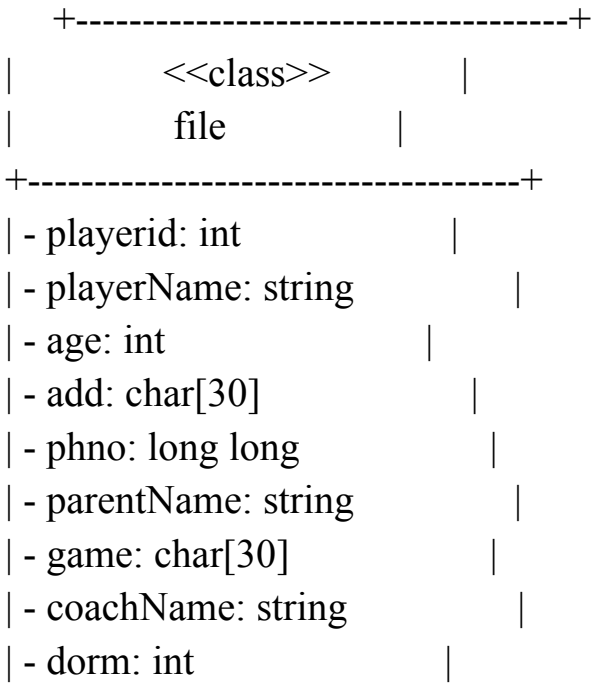
## Background (Research & Project Selection):

Before embarking on the project, a survey was conducted to identify the challenges faced in managing player records manually. The research indicated a need for an automated system to streamline the record-keeping process, reducing errors and enhancing overall efficiency.

## Project Specifications:

The project involves the development of a Player Record Management System using C++. It includes features such as adding new player records, displaying all records, deleting records, searching based on different criteria, counting players, and clearing the database. The system employs file handling for data storage and retrieval.

## Complete UML Class Diagram:



| - bed: int |
+----------+
| + input(): void |
| + output(): void |
| + retplayername(): string |
| + retplayerid(): int |
| + retage(): int |
| + radd(): char* |
| + rphno(): long long |
| + rparentn(): string |
| + rgame(): char* |
| + rcoachn(): string |
| + rdorm(): int |
| + rbed(): int |
+----------+
| + writeInFile(fobj: file): void |
| + showAllRecords(): void |
| + deleteInRecords(choice: int): void|
| + countPlayers(): void |
| + countPlayersInGame(): void |
| + searchInRecords(choice: int): void |
| + clearCompleteDatabase(): void |
+----------+
| - (static) main(): int |
+----------+

## Problem Analysis:

Manual record-keeping of player information is time-consuming and prone to errors. The lack of an organized system makes it challenging to manage and retrieve data efficiently.

## Solution Design (Project Detail, Functionality and Features):

The Player Record Management System is designed to address the challenges of manual record-keeping.

## Features:

### 1. User-Friendly Interface:

The system offers a simple and intuitive interface for easy navigation. Users interact with the system through a menu-driven approach, enhancing user experience.

### 2. Record Addition:

Users can add new player records to the system. The system prompts users to input player details such as ID, name, age, address, phone number, parent's name, preferred game, coach name, dorm, and bed.

### 3. Record Display:

The system allows users to view all player records stored in the system. The display includes comprehensive player details, providing a complete overview of each record.

### 4. Record Deletion:

Users have the capability to delete player records based on either player ID or player name. The system guides users through the deletion process, ensuring a clear understanding of the action.

### 5. Record Search:

Users can search for player records using various criteria:
Player Name
Player ID
Dorm Number
Bed Number
Parent's Name
Game Name
Persistent Storage:
Player records are stored in a file named "PlayersRecord.txt" using file handling. This ensures that records persist across program executions, maintaining data integrity.

### 6. Counting Players:

The system provides functionality to count the total number of players in the records. Users can also count the number of players enrolled in specific games.

### 7. Database Clearing:

Users have the option to clear the entire player record database.A confirmation step is implemented to prevent accidental deletion.

### 8. Dynamic Input Handling:

The system dynamically handles user input, ensuring valid and appropriate data entry. Input prompts guide users through the data input process for each player record.

### 9. Error Handling:

The system incorporates error handling mechanisms to manage unexpected input or file-related issues.Users receive informative messages in case of errors, enhancing system robustness.

### 10. Modularity:

The code is organized into functions and methods, promoting modularity and ease of maintenance. Each function encapsulates specific functionality, contributing to code readability.

### 11. Interactive Menu:

The menu-driven interface allows users to choose from various options, promoting an interactive and user-centric design. These features collectively contribute to an efficient and user-friendly Player Record Management System, addressing the challenges of manual record-keeping and enhancing the overall management of player records.

## Implementation and Testing:

The system is implemented in C++ and tested for functionality and reliability. The code undergoes rigorous testing to identify and resolve any potential bugs or issues. Unit testing, integration testing, and user acceptance testing are performed to ensure the system meets its specifications.

## Project Breakdown Structure (Workload Distribution with Timeline):

### Phase 1: Research and Planning (Week 1-2)

- Conduct research on manual record-keeping challenges.
- Define project requirements and specifications.
- Plan the development timeline.

### Phase 2: System Design (Week 3-4)

- Create UML class diagram.
- Design user interface and system functionality.

### Phase 3: Implementation (Week 5-8)

- Write code for player record management.
- Implement file handling for data storage.

Phase 4: Testing (Week 9-10)

- Conduct unit testing.
- Perform integration testing.
- Execute user acceptance testing.

Phase 5: Documentation (Week 11-12)

- Prepare a project report.
- Create user documentation.

## Results:

The Player Record Management System successfully automates the process of managing player records, leading to improved efficiency and accuracy. The system's functionality has been thoroughly tested and meets the specified requirements.

## Conclusion (Summary & Discussion):

In conclusion, the development of the Player Record Management System has addressed the challenges associated with manual record-keeping. The system provides an effective solution for sports organizations to manage player information systematically. The report summarizes the key aspects of the project, from its inception to implementation, highlighting the achieved outcomes.