ScoreBoard Automation System

Project Report

Supervisor

Ms Erum Ashraf

Submitted by

Ashir Siddique

01-134201-103

Saboor Hassan

01-134201-075

Department of Computer Science,

Bahria University, Islamabad

Certificate

Acceptance is granted to the work contained in the report titled "ScoreBoard Automation System," written by Ashir Siddique and Saboor Hassan, as a confirmation of the required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

Approved by:

Supervisor: Ms. Erum Ashraf

Internal Examiner:

External Examiner:

Project Coordinator: Ms. Maryam Khalid

Head of the Department:

October 23rd, 2023

ACKNOWLEDGMENTS

The success of this project would not have been possible without the assistance of a great number of people, but first and foremost, praise is given to Allah Almighty, who endowed us with the skills and knowledge necessary to understand the project in its entirety. His blessings are immeasurable, and gratitude will always be felt towards Him. Second, acknowledgment goes to the efforts of project manager, Mrs. Erum Ashraf, who guided us through every challenge encountered. She was encouraging and had a strong, positive influence. Finally, thanks are extended to each teacher that helped during the entire degree process.

"We think someone else, someone smarter than us,
someone more capable, someone with more resources will solve that problem.
But there isn't anyone else."

Regina Dugan

Abstract

The "ScoreBoard Automation System" marks a groundbreaking advancement in local sports management by digitizing and streamlining sports data in Android application form. Traditionally reliant on manual scorecards and handwritten statistics, local tournaments faced challenges in record-keeping and accessibility. This innovative platform revolutionizes the landscape, empowering players, organizers, and spectators. It offers features such as player registration, centralized tournament management, and intelligent algorithms fostering competition among youth. Utilizing Agile Methodology, the development process prioritizes user-friendliness, efficiency, and data security. The application addresses international and local challenges, bridging gaps in accessibility and data management. Ultimately, it ensures that local sports are enjoyed, tracked, and shared with ease and efficiency, promoting youth engagement and unifying stakeholders on a global platform.

TABLE OF CONTENTS

List of Tables

LIST OF FIGURES

Chapter 1

Introduction

1.1 Objectives

Following are the objectives of the project:

- To promote sports activities among youths is the primary objective, encouraging a healthy lifestyle and fostering teamwork and discipline.
- To provide a single platform for organizers, players, and spectators.
- To enhance the user experience, sports applications often include features such as player comparisons, team creation tools, and real-time scoring updates.
- To manage data and store records on a secure platform.

1.2 Problem Statement

It addresses international problems; the main issue in existing systems is that they work on a specific game or tournament, and it is not publicly accessible. Users can only act as spectators after visiting these kinds of applications and can find out the results and match stats. The main flaw is that these types of applications or systems are only available internationally. The existing applications (e.g Cricbuzz etc) lack many features, such as users cannot make a team and add players to it, the scoreboard is not publicly accessible, and tournaments can't be created and organized by a single application. Now, if consider local problems, one of the major issues is that there isn't any automated system that can store match records, players' statistics, and organize tournaments; all of these are done manually on pages, resulting in the wastage of records because hard copies require more space and proper maintenance. Another problem is that when a game is in progress, the record is displayed on scoreboards using numerical cards, and a person would place the proper digits on a hook after making a point. With the existing system, organizers are unable to monitor hometown/local tournaments from other places due to the lack of a digitized system that provides live stats of the game. The proposed application will address this problem by making this system automated.

1.3 Solution

Agile Methodology will be the development methodology that is use for this project. Agile Methodology is the development methodology chosen for the project as it provides a systematic approach to development. The goal is to create a user-friendly design that is both simple and exquisite to ensure ease of use. The methodology consists of several processes, including requirement analysis, design analysis, development, testing, deployment, and review.

The project development process involves a thorough analysis of requirements and design to ensure the development of a user-friendly, functional, and efficient product. Module-based development using pair programming enables effective coding, while continuous testing and improvement ensure a quality end product. Continuous deployment enables the quick integration of new code to reduce lead time, and sprint review groups adjust the backlog as needed. Data backup and restoration features are incorporated to prevent any data loss resulting from hardware failure. These steps ensure the successful and seamless development of the project.

Figure 1 shows the visual representation of agile development methodology.

Figure 1.1: Project Development Methodology

1.4 Work flow Diagram

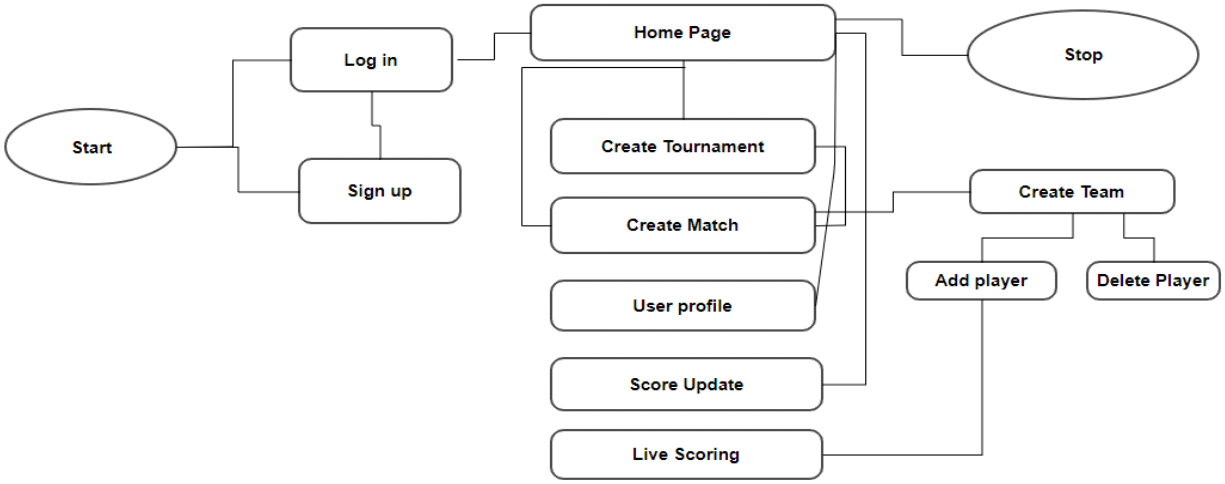

Figure 1.2 Workflow Diagram

In the scoreboard automation app, the first step is for the user to log in. Once the user is already registered, they can proceed to the login page; otherwise, they must first register using the signup option. After logging in, the user will be directed to the home page, which includes several options such as creating a tournament, creating a match, updating user profiles, updating scores, and live scoring. When the user selects "create tournament," they will be given the option to create a new tournament. When they select "create match," they can create a new match and form teams if required. If the user has played a match without creating a tournament, they can directly create a match without having to form a tournament.

1.5 Project Scope

The scope of the project is to develop an automated sports management system that includes live score tracking, player and match statistics, rankings, tournament management, and the ability to add and update players' performance records. The application will be accessible to everyone, including sports enthusiasts, coaches, players, and event organizers. It will provide notifications and graphical representation of data in the form of charts and graphs. The project will cover multiple sports played on various levels, from local to international, and will provide a centralized platform for recording and maintaining sports data. The system will be designed to address the problem of wasted records of matches played between rivals and player stats. With this solution, users can easily manage and track sports events, view performance records, and make informed decisions. [7].

1.6 Feasibility Study

The project schedule, given proper resources and a proficient working team,

- **Risks Involved**: The main risks involved include hardware design challenges, security issues, software problems, and the possibility of unknown saving of player data.
- **Resource Requirement**:

✓ Application developer.
✓ AI experts.
✓ Technology equipment's.
✓ Platform.

1.7 Solution Application Areas

A platform is needed for local tournament organizers to manage player performance, select rivals, and preserve associated data. This proposed project will provide a platform for various sports like cricket, badminton, and ping pong. The platform will benefit management, players, and spectators who express a need for this technology. The proposed system will track player information, match statistics, and tournament information. It will help struggling local players maintain records and determine ranking

Chapter 2

Literature Review

2.1 Background

Players are always looking for rivals and a competitive atmosphere. Users desire to compete with one another and tell their tales to their near and dear ones. In free time, almost all age groups participate in sports. There are occasions when games involve teams or individuals (in shape of tournament). These contests are held on local sports fields, colleges, and schools. The management, players or even audience want that these games must be recorded somewhere and must be publicly available. For international matches, platforms are available which track all records of matches and player's performance but locally there is no such platform available which can automate scores and record matches stats. However, local players and organizers can switch to a fully automated system with the help of this application. Users can monitor and follow sports depending on their interests and favorite local areas if management wants to make the scorecard and stats publicly available [1].

Automated Scoreboard is a digitized and smart android platform for local sports. It's a feature-rich application that local players always think about. In local tournaments, a manual approach like using a pen and scorecard is employed, which is an old terminology, and afterward, records are wasted. For keeping score, calculating time, and showing statistics, new software is proposed which will overcome the problem of the existing system and has numerous features. Some of these features include multiple players can register, it will be publicly accessible, anyone can manage all tournaments, records of the tournament will be saved, and players' performance will be displayed publicly. It is not just scorecard digitization but through clever learning and scheduling algorithms, it also creates a positive competitive atmosphere for young and local players. The scorecard and statistics of the game or the tournament can be made available to the public if management so chooses. Because players' bios and stats are made public, this app also gives local players a large platform to showcase their talents to the globe, allowing selectors to choose the most effective, skilled, and top performers in their area. The main aim is not to make a scoreboard for a single category but to provide a platform for different games like cricket, badminton, and ping pong. Additionally, this application automates tournament scheduling. The desire for this type of tool is constant among management, players, and spectators.

2.2 Existing Applications

Application review is a comprehensive analysis of the previous systems environment that aims to assist developers focus their efforts and provide better solutions. There is multiple software that are used for this purpose. These systems show progress of games, scores, and stats of player. User can get to know about upcoming tournaments or matches and go through whole schedule. Whereas some systems use for adding scores between two or more players and getting result. But according to review, these systems have lack of many features as no user can add tournaments, manage, and record them. Data cannot be stored in one platform and players stats cannot be recorded [2].

2.2.1 Scoreboard

Scoreboard [3] is an android application that makes it simple to keep track of your favorite games and competitions' score points. Ideal for scoring in a wide range of multiplayer games such as card games, board games, table tennis, volleyball, and sports in general. This system allows user to select how many members will play and after selecting numbers of players can add their names. It also provides simple and intuitive screens for users. The application also has feature of finish match at any time. Scoreboard also have no limits for the scores and in last you can check out the results and champions will be declared. This application does not have feature of a choosing specific sports as it works in general and user interface is not friendly. Major flaw of this app is that it does not work for complex games or any kind of major sports an because data cannot be stored. The application also does not work on real time it's just a scoring platform for just simple games. The features of the application are listed below.

- Simple and intuitive screens.
- No score limit.
- Results generated.
- No time limit.
- Functions to add and remove points.

The limitations of the application are listed below.

- Analytics for players cannot be saved.
- Game analytics are temporarily recorded.

- Only points calculated.
- Tournament cannot be organized.
- Game cannot be followed.

2.2.2 Cricbuzz

Cricbuzz [4] is an application that is most popular for its cricket news services. It features, news, articles and live coverage matches. The application provides data about games, timetables, outcomes, editorials, and more. Users can view pictures of their preferred players or teams by visiting the photo gallery. The application is convenient to use and effective, giving users quick access to all the latest sports and gaming news. It provides notifications for live matches and breaking news. The application shows records, ranking, stats and schedule of upcoming matches. The application also features exclusive articles and user can follow all type of international cricket matches. Additionally, viewers get insight to all player information and therefore can explore further. A critical feature of this application is ball-by-ball commentary. However, because thisapplication lacks additional features and restricts the end user to a spectator role, end users may experience problems with it. The application only provides statistics of cricket and can- not accommodate multiple sports. It would have been better if there was a feature for more sports and games. The application does not have the feature of managing a local tournament on a single platform. Moreover, users are unable to create scoreboards for individual matches. User can only act as a spectator and cannot manage anything. The application functions well and have more information about cricket, but additional features need to be added to create a reliable platform for local sports and games.

The features of the application are listed below.

- Provide players stats.
- Live scores.
- Provides commentary.
- Schedules of upcoming matches.
- Notifications.

The limitations of the application are listed below.

- For single sport only.
- Unable to create scoreboard.

- Cannot organize tournament.
- Players cannot be added.
- Regional sports cannot be recorded.

2.2.3 Virtual Scoreboard

Virtual Scoreboard [5] is an android-based application that is professionallybuilt for keeping scores while playing games and sports in a simple and interactive way. This system allows its users to edit timers before game start. The application comes with many numbers of game and sports including soccer, basketball, volleyball, chess, and hockey. The user can also customize colors of interface. The application provides the option of playing alerts and buzzers to indicate when a period has expired. The user interface of this application is quite easy to use and can be an advantage for those who play simple games mostly. One of the most main features of the applicationis that user can share live scores with everyone using match codes. By using application, users can update in real-time to ever other spectator that is following. It also helps user on keeping number of fouls. However, application does not work like conventional scoreboards, which means in general first team play and set target for other in most of the games, but this system does not have compatibility and different rules for that kind of sports and games. The system can then malfunction as a result. The downside of this application is that it does not have the feature of arranging contests, recording player's data, and registering players. The user interface of this application is also not up to the mark. The ability to create graphs for player metrics and match analytic is not available in the application.

The features of the application are listed below.

- Tally game scores.
- Edit timers.
- Multiple sports and games.
- Share live scores.
- Updates in real time.

The limitations of the application are listed below.

- Record cannot be saved.
- Not a conventional scoreboard.
- No analytical graphs.

- Teams cannot be formed.
- Contests cannot be managed.

The parameters selected for the comparison are listed below.

- Graphs and Charts.
- Ranking using algorithms.
- Notifier system.
- Add and update players.
- Accessibility.
- Display Stats.
- Capacity of storing record.

A comparison table is a double-entry table that has the entities which is compared in the columns and the comparison features are compare in the rows. By highlighting the similarities and significant differences between the compared entities, the comparison table is an especially helpful tool for making decisions. Application review comparison is shown in Table 2.1.

Table 2.1 Application Comparison

Features	**ScoreBoard [3]**	**Cricbuzz [4]**	**Virtual Scorecard [5]**	**Proposed System**
Live Score	✗	✓	✓	✓
Graph & Charts	✗	✗	✗	✓
Player Stats	✗	✓	✗	✓
Match Stats	✓	✓	✓	✓
Ranking	✓	✓	✗	✓
Tournament management	✗	✗	✗	✓
Add & Update Players	✓	✗	✗	✓
Accessible by Everyone	✓	✗	✓	✓
Players Performance record	✗	✓	✗	✓
Save and Maintain data	✗	✓	✗	✓
Notification	✗	✓	✗	✓
Multiple Sports	✓	✗	✓	✓

Chapter 3

Requirement Specifications

The cricket scoreboard automation project requires specific hardware, software, and network configurations to function effectively. These include a multi-core processor, ample RAM, compatible operating systems, modern web browsers, reliable database systems, and a stable internet connection. Adherence to these requirements is essential for smooth installation, performance, and secure data management within the application.

3.1 Existing System

In the contemporary landscape, technology has permeated every sphere, effectively shrinking the world into a global village where data travels at the speed of light. This digital evolution has empowered users, making them more discerning and time-conscious, valuing efficient shortcuts in their interactions. While numerous mobile applications have been designed to automate various facets of cricket management, there has been a disproportionate focus on aspects such as live scoring and match schedules, with insufficient attention paid to ensuring the quality of game statistics and player performance data. The existing cricket scoreboard automation systems, while functional to a degree, suffer from usability challenges. Some lack essential features, hindering negotiations or impeding the registration process for players and teams. Moreover, a significant gap exists in generating legally binding agreements. Thus, there exists a pressing need for a comprehensive solution—a unified platform that encompasses all essential features, addressing the existing gaps and facilitating seamless settlements between organizers, players, and stakeholders in the cricketing community.

3.2 Proposed System

Our cricket scoreboard automation app, tailored to address diverse needs, offers a range of cutting-edge features enhancing the user experience. Similar to CropFit, our application streamlines various aspects of cricket management.

Effortless Player and Team Management:

Our app allows effortless player and team management, enabling organizers to search for players seamlessly. There is no need for extensive field visits; users can easily access profiles of top-rated players and teams. The app facilitates the selection of players based on their performance statistics, ensuring teams are composed of the best talent.

Dynamic Match Insights:

Live Score, Graphs & Charts, Player Stats, and Match Stats features provide real-time match insights. Live Score keeps users updated on ongoing matches, while Graphs & Charts offer visual representations of game data, aiding in trend analysis. Detailed

Player Stats and Match Stats provide comprehensive performance metrics, enhancing strategic decision-making.

Comprehensive Tournament Management:

Our application boasts robust tournament management capabilities. Organizers can efficiently create, organize, and manage tournaments. The app includes a sophisticated Ranking system, allowing users to track team and player rankings, fostering healthy competition and skill development.

User-Friendly Interface:

The app's intuitive interface ensures accessibility for everyone, from casual fans to seasoned cricket enthusiasts. Users can effortlessly add, update, and manage player profiles. The system's simplicity ensures that even novice users can navigate the app seamlessly.

Accessible Data and Record-Keeping:

Accessible by Everyone, our app ensures that match data, player statistics, and tournament records are easily available. This inclusivity promotes transparency and engagement within the cricketing community. The app's Players Performance Record feature maintains detailed statistics, providing valuable insights for players and team managers.

Secure and Reliable:

Data Security is paramount. Our app employs state-of-the-art security measures to saf eguard user information. Utilizing secure technologies, the app guarantees the confidentiality and integrity of user data, providing a safe platform for all users.

Efficient Data Management:

Our application integrates Firebase for efficient and real-time data storage. Match results, player performances, and tournament details are systematically stored and maintained, ensuring a seamless experience for users while promoting accurate record-keeping and data management.

3.3 Requirement Specification

The requirements specification for our cricket scoreboard automation app is pivotal in ensuring a shared understanding among all stakeholders, encompassing users and developers alike.

3.3.1 Functional Requirements

- **Sign Up**

Description: New users should have the ability to sign up for the app by providing their essential information.

Output: Successful registration and creation of a user account.

Processing: The application should validate the provided user details, ensuring that all required fields are filled correctly. It should also check if the chosen username and email-id are unique and not already registered in the system.

Input: User's first name, last name, desired username, email-id, password, toddler details (name, disorder if any, age).

- **Log In**

Description: Registered users can log in using their username and password.

Output: Successful login and access to the app's features.

Processing: The app verifies the provided credentials against the stored user data.

Input: User's username and password.

- **Record Live Scores**

Description: The app allows authorized users to record live scores during a match.

Output: Real-time updated scores for ongoing matches.

Processing: The app should provide a user interface to input and update scores. It should calculate and display current scores accurately.

Input: Scores for each team/player during the match.

- **Manage Player Statistics**

Description: Users with appropriate access can manage and update player statistics.

Output: Updated player statistics including performance details.

Processing: The app should allow authorized users to add, edit, and view player statistics. It should perform calculations for player averages and other relevant metrics.

Input: Player performance data (runs, goals, assists, etc.).

- **Rankings Generation**

Description: The app generates rankings based on player or team performance.

Output: Updated rankings list for players/teams.

Processing: The app calculates rankings using predefined algorithms and displays them in a clear format.

Input: Player or team performance data.

- **Tournament Management**

Description: Authorized users can create, manage, and organize tournaments.

Output: Scheduled and managed tournaments with fixtures and details.

Processing: The app should allow users to create tournaments, add teams/players, set schedules, and manage brackets.

Input: Tournament details, teams, schedules.

- **Data Visualization**

Description: The app provides graphical representation of match and player data.

Output: Graphs and charts displaying match statistics and player performance.

Processing: The app should convert raw data into visually appealing graphs and charts for easy understanding.

Input: Match and player performance data.

- **Notifications**

Description: The app sends notifications to users for match updates, results, and other relevant information.

Output: Notifications/alerts about match events, results, and important updates.

Processing: The app should push notifications based on match progress and user preferences.

Input: Match progress, user preferences.

- **Accessibility and User Roles**

Description: The app assigns different access levels (admin, coach, player, spectator) with appropriate permissions.

Output: Role-based access and privileges for different user types.

Processing: The app should authenticate users and provide access based on their roles and permissions.

Input: User roles and permissions.

- **Data Security**

Description: The app ensures the security and privacy of user data.

Output: Safeguarded user data and information.

Processing: The app should implement encryption, secure authentication, and data protection measures.

Input: User data and login credentials.

- **User-Friendly Interface**

Description: The app should have an intuitive and user-friendly interface.

Output: Easy-to-navigate app interface for users.

Processing: The app should be designed with a user-centric approach, considering ease of use and navigation.

Input: User interactions and preferences.

3.3.2 Non Functional requirements

- **Performance:**

  Description: The application's responsiveness and efficiency under varying load conditions.

  Output: The application shall respond to user interactions within 2 seconds on average, even during peak usage.

  Input: Simulated user interactions generating variable levels of load, including concurrent user actions.

  Processing: The application shall employ caching mechanisms, query optimization, and load balancing to maintain optimal response times.

- **Scalability:**

  Description: The system's ability to handle increased user load without compromising performance.

  Output: The application should seamlessly accommodate a 50% increase in user registrations and usage over a 6-month period.

  Input: Gradual increase in user activity and registration rates over a specified timeframe.

  Processing: The system architecture shall support horizontal scaling and dynamic allocation of resources to handle growing user demands.

- **Security:**

  Description: Measures to safeguard user data and protect against unauthorized access.

  Output: User data shall be stored and transmitted securely, and only authorized users shall have access to their accounts.

  Input: User authentication details, sensitive personal information, login attempts.

Processing: The application shall implement encryption, secure authentication protocols, and role-based access control to ensure data integrity and user privacy.

- ## Availability:

  Description: Ensuring the application is accessible and operational for users.

  Output: The application shall have a minimum uptime of 99.5% over a calendar year, excluding scheduled maintenance windows.

  Input: Monitoring of uptime and availability metrics.

  Processing: The system shall employ redundant servers, automated failover mechanisms, and regular maintenance schedules to minimize downtime.

- ## Reliability:

  Description: The system's ability to perform consistently without errors or disruptions.

  Output: The application shall handle concurrent user registrations without crashing or losing data.

  Input: Simulated high-volume user registration scenarios, including stress tests.

  Processing: The application shall implement error handling, logging, and transactional integrity mechanisms to ensure reliability.

- ## Usability:

  Description: How user-friendly and intuitive the application is for various users.

  Output: Users shall be able to complete the registration process without requiring external assistance.

  Input: User feedback, usability testing sessions.

  Processing: The user interface shall have clear instructions, real-time validation, and logical flow to guide users through registration.

- **Compatibility:**

  Description: The application's ability to function across different platforms and devices.

  Output: The application shall render and function correctly on major web browsers and mobile devices.

  Input: Testing across multiple browsers, operating systems, and device types.

  Processing: The application shall adhere to web standards and responsive design principles to ensure cross-platform compatibility.

- **Data Privacy:**

  Description: Protecting user data and ensuring compliance with relevant privacy regulations.

  Output: User data shall be stored, processed, and transmitted in accordance with applicable data protection laws.

  Input: Privacy regulations, user data protection requirements.

  Processing: The application shall implement data encryption, consent management, and user data deletion mechanisms to ensure data privacy.

- **Maintenance:**

  Description: Ease of maintaining and updating the application over time.

  Output: Regular updates and bug fixes shall be rolled out with minimal disruption to user experience.

  Input: Frequency of updates, software patches.

  Processing: The application's codebase shall be well-organized, documented, and designed for easy modification and maintenance.

- **Performance Monitoring:**

Description: Monitoring system performance and identifying areas for optimization.

Output: System administrators shall have access to performance metrics and error logs for proactive issue resolution.

Input: Performance monitoring tools, logs, analytics.

Processing: The application shall integrate with performance monitoring tools to collect and analyze data on response times, resource utilization, and errors.

3.4 Descriptive Use case

3.4.1 Login

Table 3.1 Login Table

Use Case ID:	**UC-1**	
Use Case Name:	**Login**	
Actor(s):	Admin	
Pre-Conditions:	Admin adds information to login.	
Basic Flow:	1. Opens the mobile app. 2. Enters email and password. 3. Click Login. Logged in to the mobile app.	
Actor Actions		**System Response**
1	Admin inputs an email and password.	System check admin's provided credentials. If the admin's credentials are valid,the admin is authenticated, and the home screen is displayed.
Alternative Course of Action (if any)		
Actor Action		**System Response**
	NA	NA

3.4.2 Signup

Table 3.2 Signup

Use Case ID:	**UC-2**	
Use Case Name:	**Signup**	
Actor(s):	User	
Pre-Conditions:	User accesses the signup section.	
Priority:	High	
Basic Flow:	1. Opens the mobile app. 2. Selects the signup option. 3. Enters personal information, including email and password. 4. Clicks Sign Up. User account is created.	
Actor Actions		**System Response**

1 2 3	User selects the signup option. User enters personal information, including email and password. User clicks Sign Up.	System displays the signup form. System validates the provided information. System checks the entered data. If the information is valid, the user account is created, and a confirmation message is displayed..
Alternative Course of Action (if any)		N/A

3.4.3 Record Live Score

Table 3.3 Record Live Score

Use Case ID:	**UC-3**	
Use Case Name:	**Record Live Score**	
Actor(s):	Scorekeeper	
Pre-Conditions:	Scorekeeper accesses the live scoring section.	
Priority:	High	
Basic Flow:	1. Scorekeeper opens the mobile app. 2. Selects the live scoring option. 3. Chooses the ongoing match. 4. Enters live score updates for both teams. 5. Saves the scores. Live scores are updated.	
Actor Actions		**System Response**
1 2 3 4 5	Scorekeeper selects the live scoring option. Scorekeeper chooses the ongoing match. Scorekeeper enters live score updates for both teams. Scorekeeper saves the scores. Live scores are updated, and a confirmation message is displayed.	System displays the list of ongoing matches. System loads the match details. System validates the entered scores. System updates the live scores in real-time.
Alternative Course of Action (if any)		N/A

3.4.4 Manage Player Statistic

Table 3.4 Manage Player Statistics

Use Case ID:	**UC-4**	
Use Case Name:	**Manage Player Statistics**	
Actor(s):	Coach	
Pre-Conditions:	Coach accesses the player statistics section.	
Priority:	High	
Basic Flow:	1. Coach opens the mobile app. 2. Selects the player statistics option. 3. Chooses a player. 4. Edits player statistics such as runs scored, wickets taken, etc. 5. Saves the changes. Player statistics are updated.	
Actor Actions		**System Response**

	1. Coach selects the player statistics option. 2. Coach chooses a player. 3. Coach edits player statistics. 4. Coach saves the changes. 5. Player statistics are updated, and a confirmation message is displayed.	• System displays the list of players. • System shows the player's current statistics. • System allows the coach to modify the statistics. • System updates the player statistics.
Alternative Course of Action (if any)		N/A

3.4.5 Ranking Generation

Table 3.5 Ranking Generation

Use Case ID:	**UC-5**	
Use Case Name:	**Ranking Generation**	
Actor(s):	System	
Pre-Conditions:	Matches and player statistics are updated.	
Priority:	High	
Basic Flow:	1. System compiles match results and player statistics. 2. Applies ranking algorithms to calculate team and player rankings. 3. Generates updated rankings list. 4. Displays rankings on the app.	
Actor Actions		**System Response**
	1. System compiles match results and player statistics. 2. Applies ranking algorithms. 3. Generates updated rankings list. 4. Displays rankings on the app.	• Gathers data from matches and player performance records. • System processes the data using predefined ranking algorithms. • System calculates team and player rankings. • Rankings are shown in the appropriate section of the application.
Alternative Course of Action (if any)		N/A

3.4.6 Tournament Management

Table 3.6 Tournament Management

Use Case ID:	**UC-6**	
Use Case Name:	**Tournament Management**	
Actor(s):	Organizer	
Pre-Conditions:	Organizer accesses the tournament management section.	
Priority:	High	
Basic Flow:	1. Organizer opens the mobile app. 2. Selects the tournament management option. 3. Creates a new tournament or selects an existing one. 4. Manages fixtures, teams, and match schedules. 5. Saves the tournament details.	
Actor Actions		**System Response**
1 **2**	1. Organizer selects the tournament management option. 2. Organizer creates a new	• System displays a list of available tournaments. • System provides options for creating or editing tournament

3	tournament or selects an existing one. 3. Organizer manages fixtures, teams, and match schedules. 4. Organizer saves the tournament details.	details. • System allows the organizer to add, edit, or remove fixtures and teams. • System updates the tournament information and confirms the changes.
Alternative Course of Action (if any)		N/A

Chapter 4

System Design

4.1 System Architecture

The scoreboard automation cricket app is also based on the MVC (Model-View-Controller) architecture, which provides a secure and organized system. In this architecture, the user interacts with the rich GUI layer known as the view, while the model handles the data related to the cricket game, and the controller acts as the intermediary, managing the logic and fetching data from the model.
The following diagram shows the system architecture of Scoreboard automation.

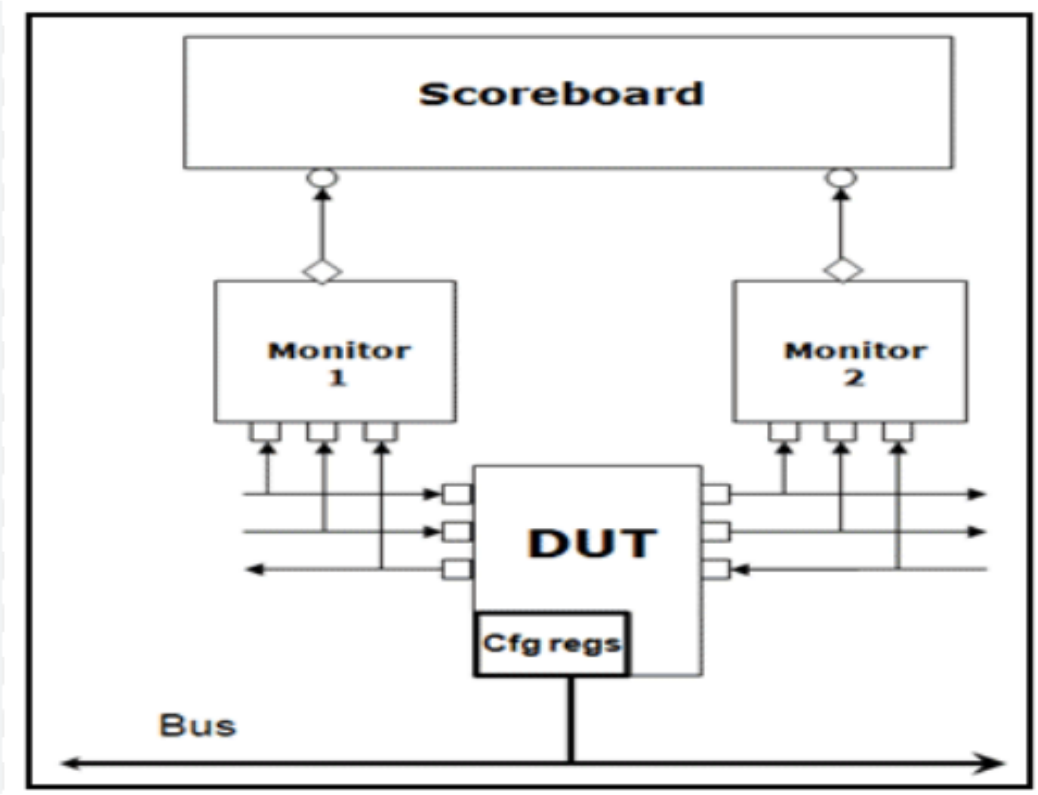

Figure 4.1 System Architecture

4.2 Use case Diagram

Use case describes the actions that can be performed by the actor. In our system we have two types of actors Admin, User. It tells the interaction between actor and the system to accomplish the goal.

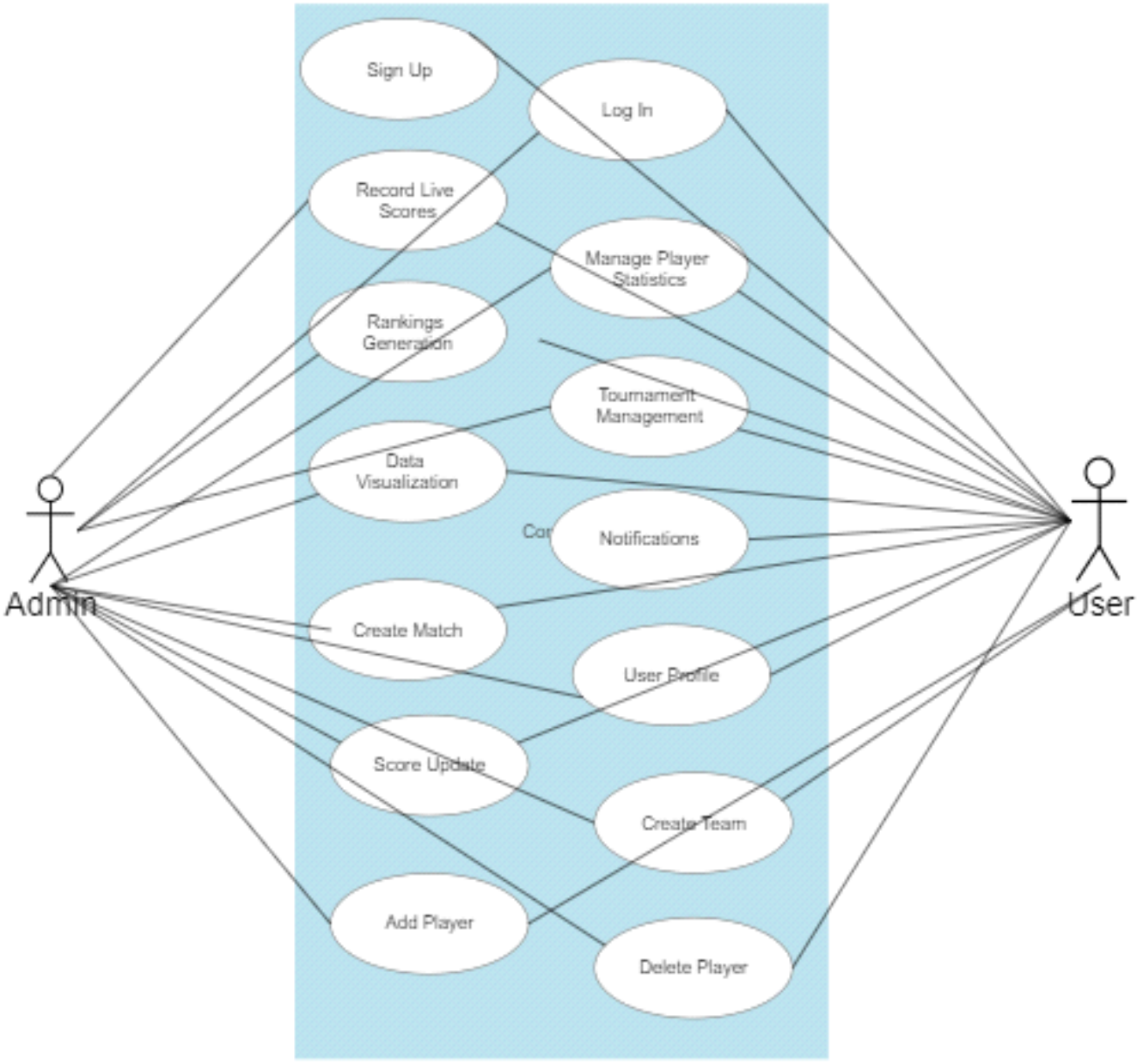

Figure 4.2 Use Case Diagram

4.2 Sequence Diagram

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. Different types if interactions will be performed by using our application. The user will interact with the system to perform various tasks. This section provides the detail view.

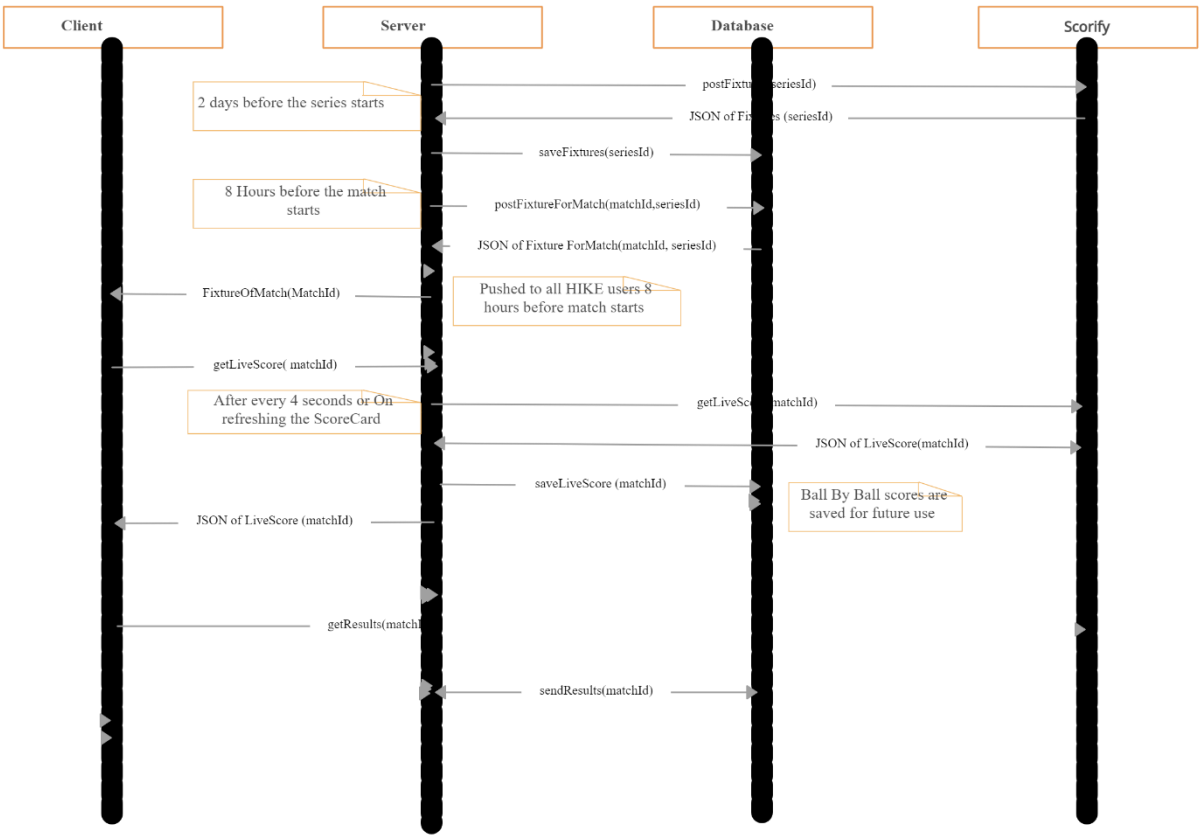

Figure 4.3 Sequence Diagram

4.3 Data Flow Diagram

It illustrates how data enters and exits the system, where it's stored, and how it's transformed as it moves through different processes. DFDs help in understanding the system's overall structure and the interactions between its components, making them essential tools for visualizing data flow and system behavior in a clear and concise manner.

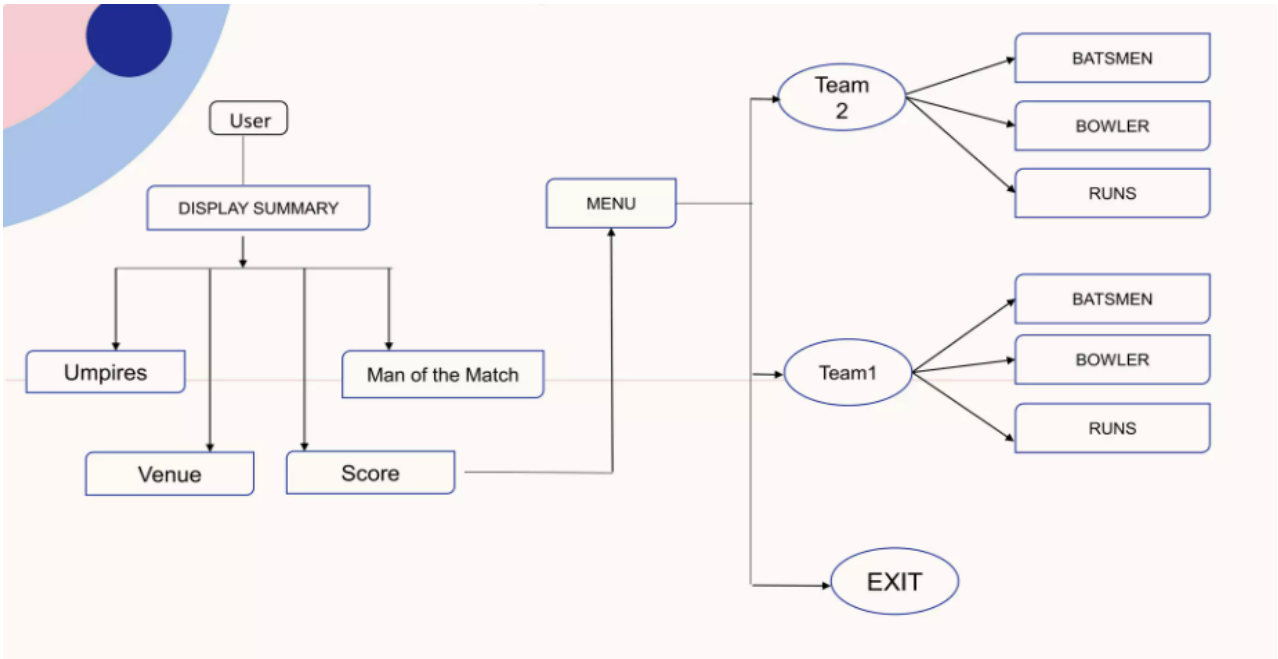

Figure 4.4 Data Flow Diagram

4.4 Activity Diagram

It illustrates the dynamic aspects of a system by focusing on the flow of actions triggered by specific events or conditions. Activity diagrams are widely used for modeling business processes, software workflows, and system behaviors, providing a detailed and intuitive view of how different activities relate to each other and the decisions made during their execution.

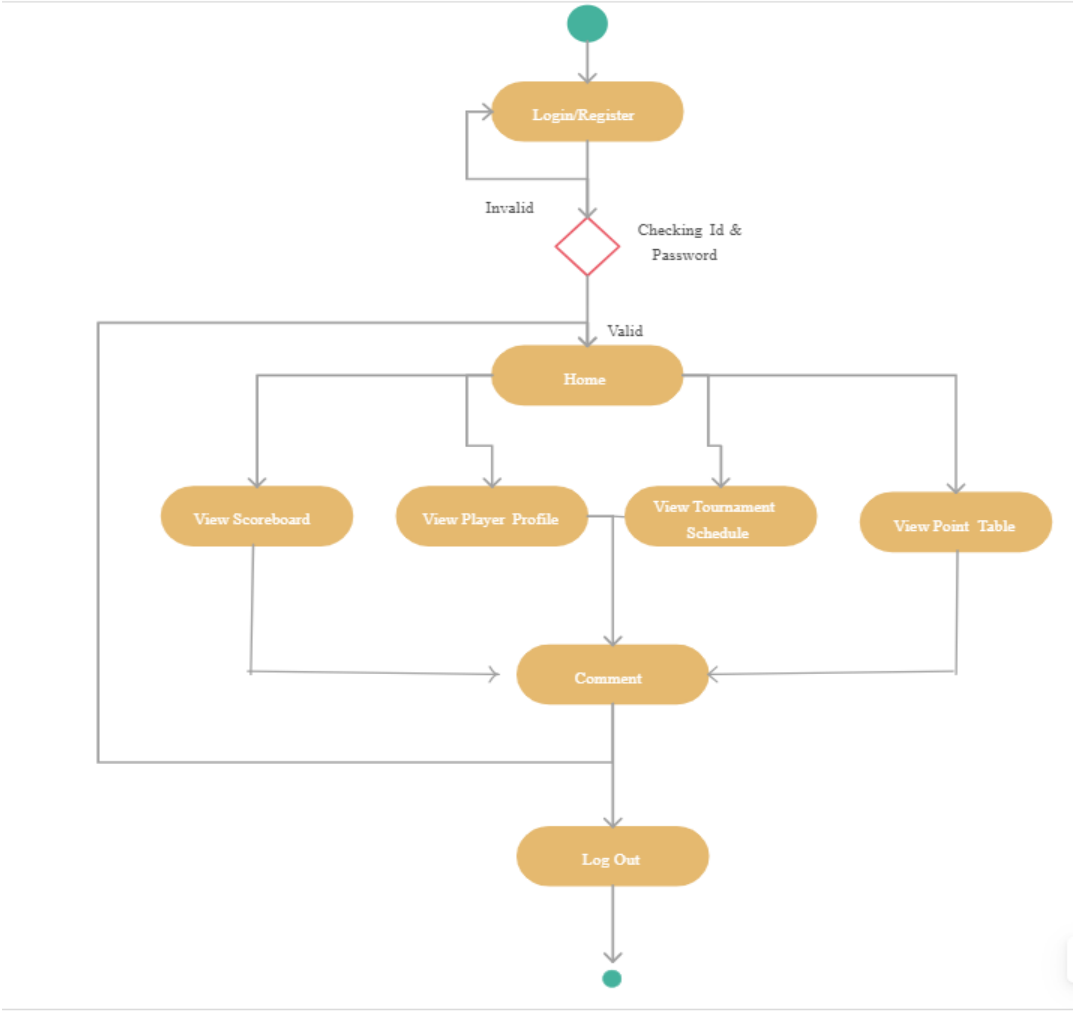

Figure 4.5 Activity Diagram

4.5 GUI Design

We have design an interface keeping in mind all the usability factors.Also we try to make it look pleasing and aesthetic. The purpose of creating such interface is that the user should feel good while using our application. As nuch importance is now given to the look and feel of interface now a days. In this section we are going to show our GUI.

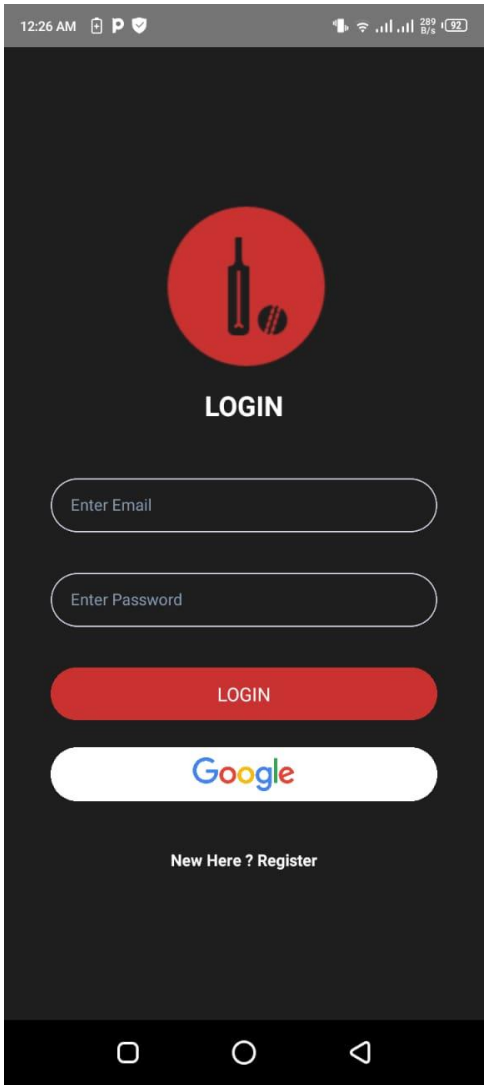

Figure 4.6 Sign in Page

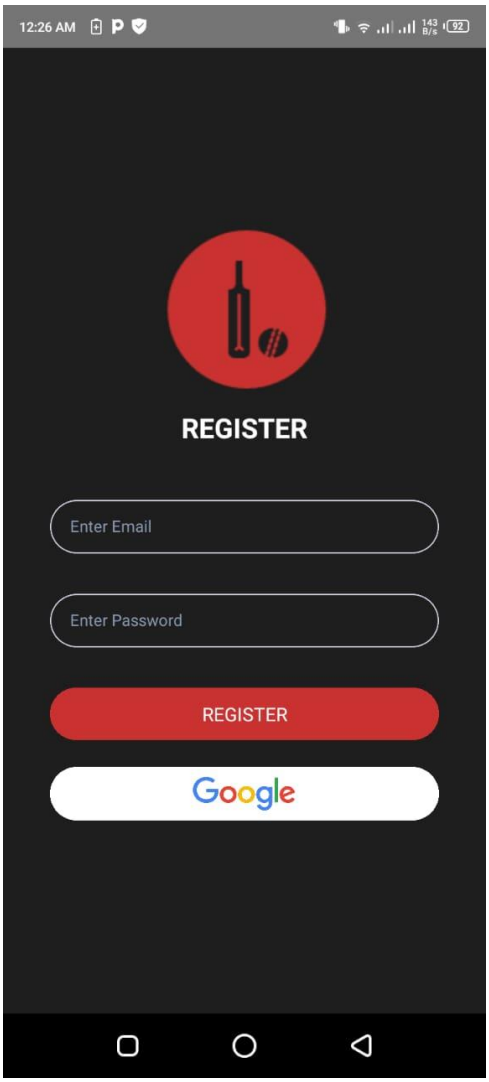

Figure 4.7 Sign up Page

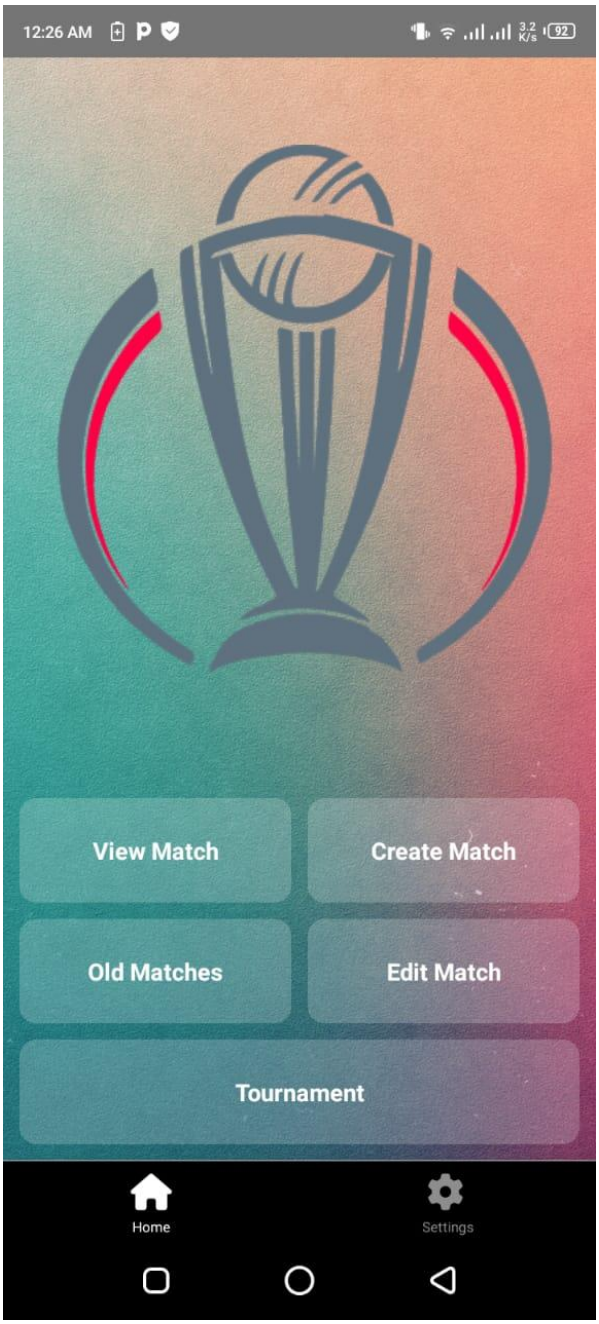

Figure 4.8 Home Page

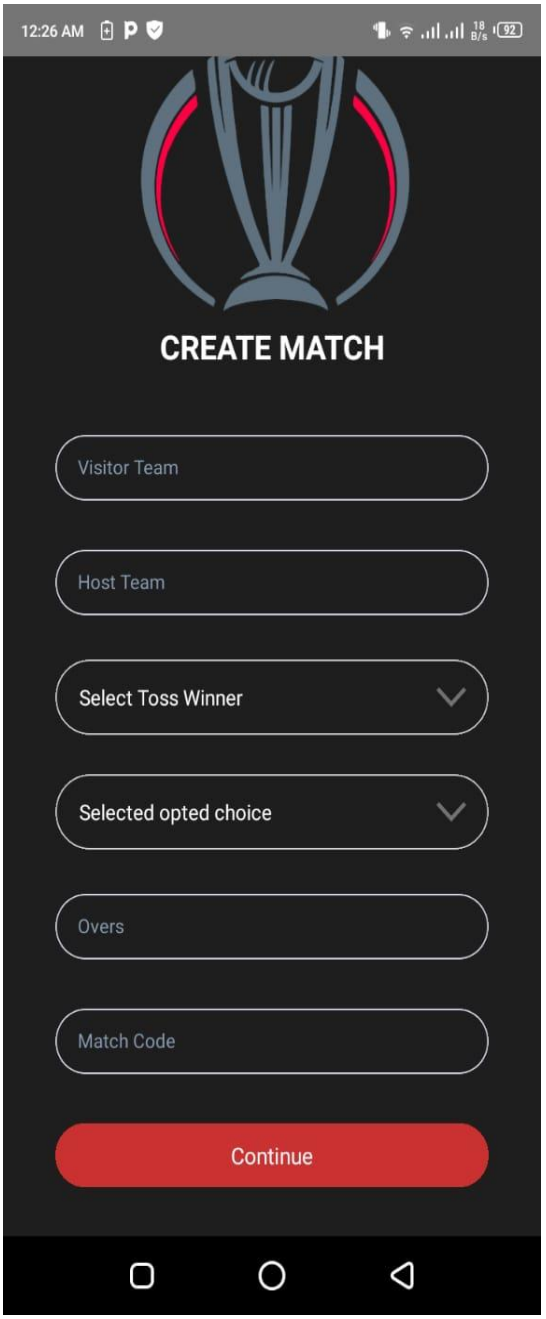

Figure 4.9 Create Match Page

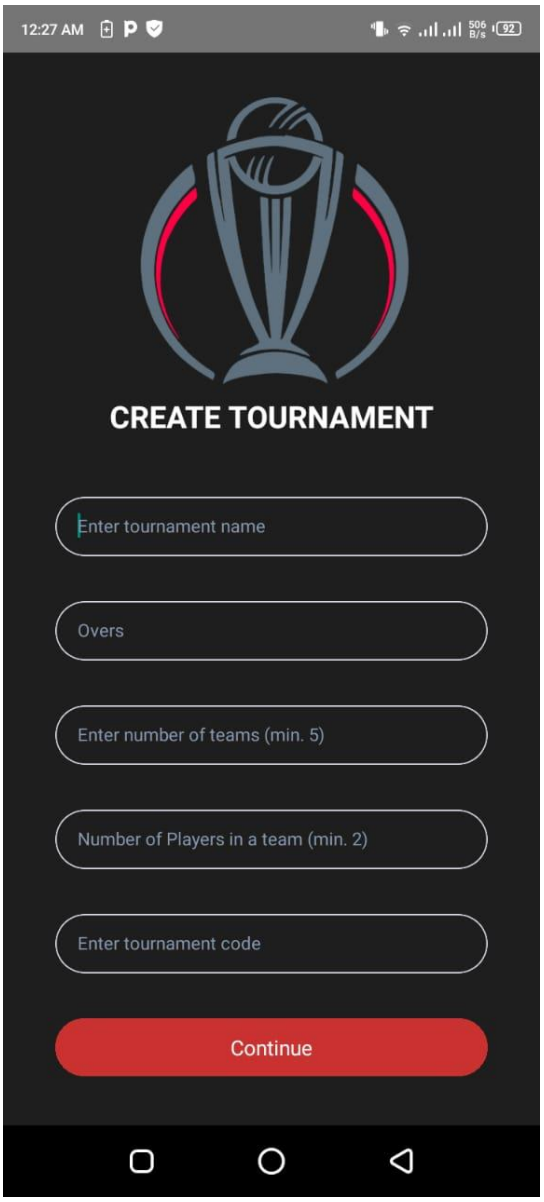

Figure 4.10 ceate tournament Page

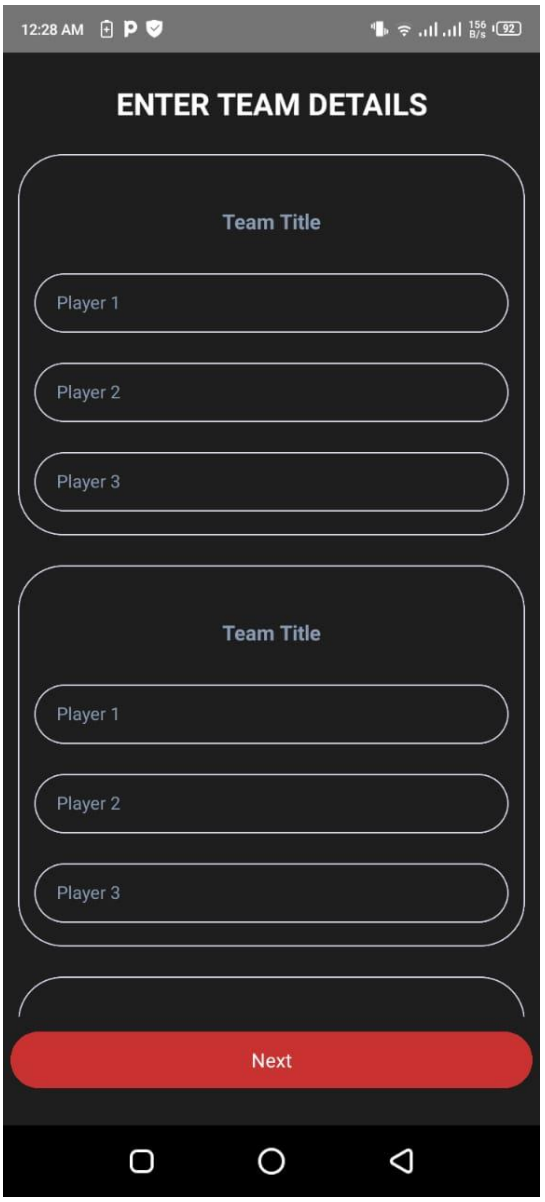

Figure 4.11 Enter team details Page

Figure 4.12 Tournament schedule Page

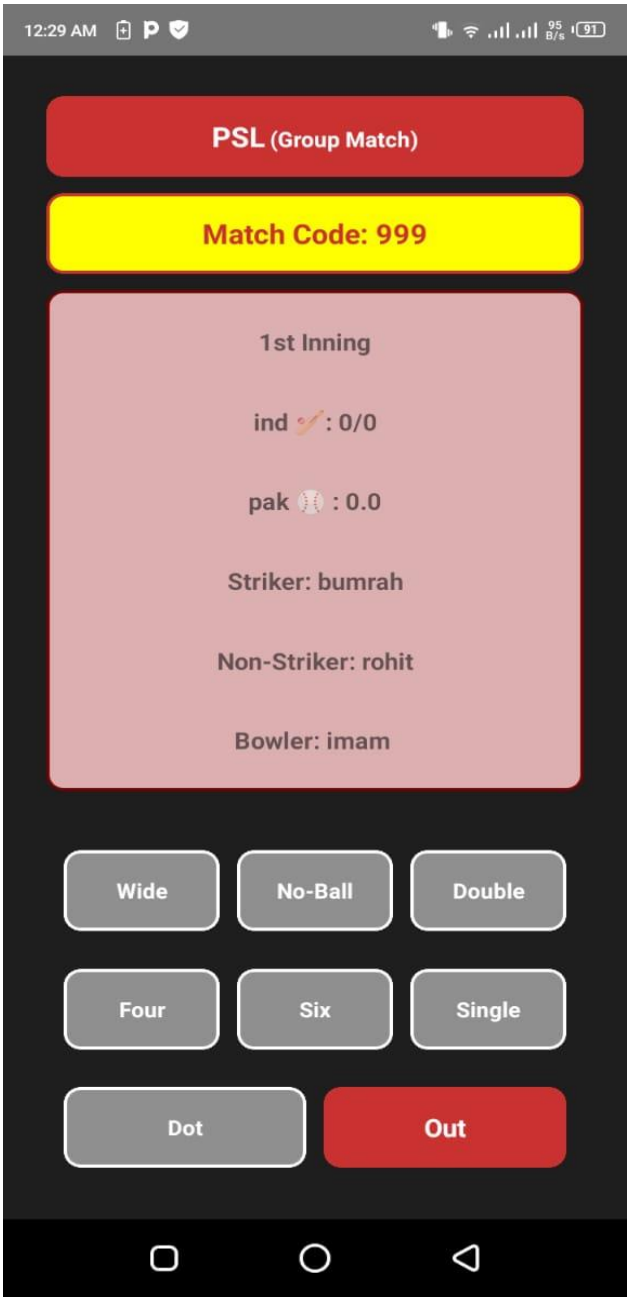

Figure 4.13 Enter score page

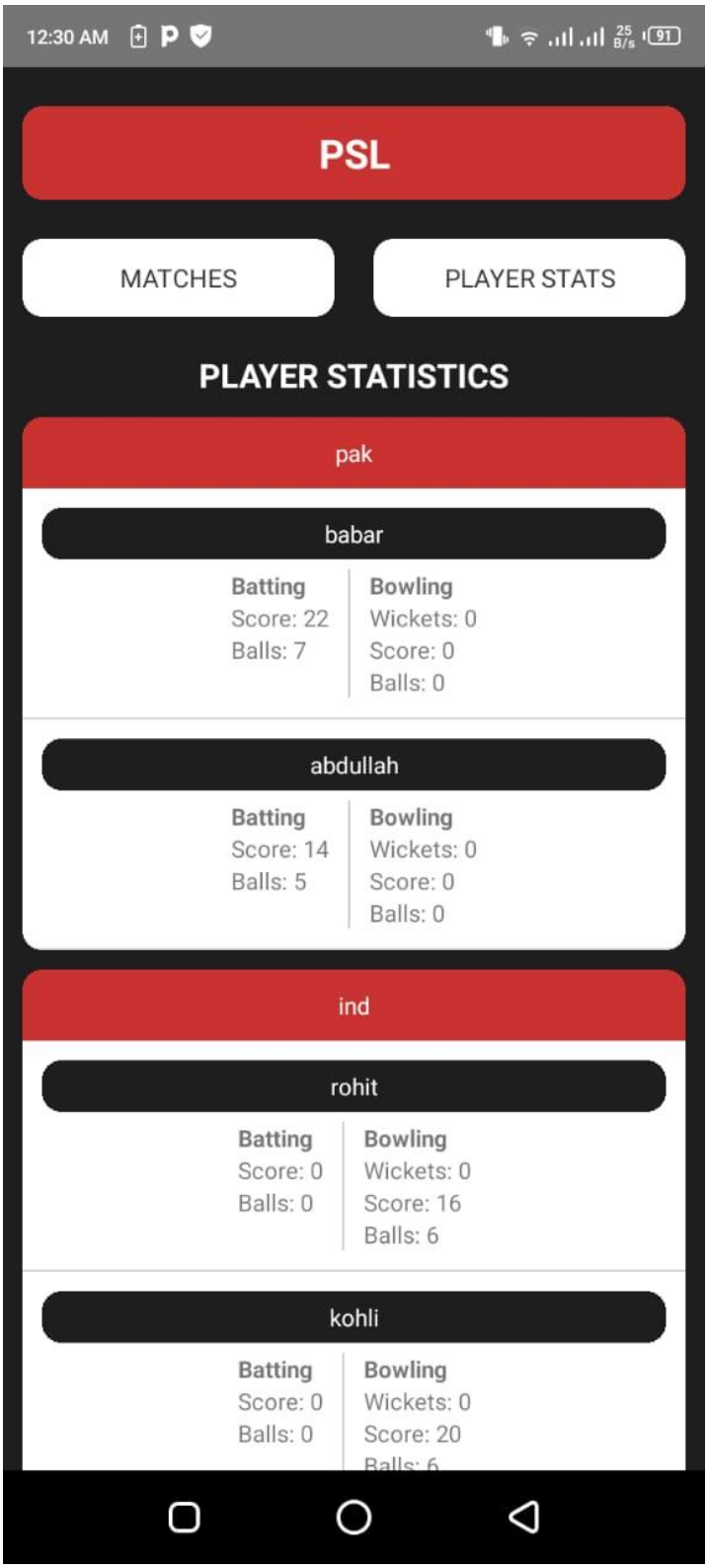

Figure 4.14 Player Stats Page

Chapter 5

System Implementation

In this chapter, the technical implementation of the Scoreboard Automation Application is delved into. The implementation process involves various tools, technologies, and strategic approaches to ensure the successful development and deployment of the application.

5.1 Tools and Technology

For the development of the Scoreboard Automation Application, the following tools and technologies have been utilized:

5.1.1 IDE (Integrated Development Environment)

A robust IDE tailored for web and mobile applications was employed, ensuring efficiency and productivity during the development process. The choice of IDE allowed for seamless coding, debugging, and testing of the application's components.

5.1.2 Database Management System

Firebase Realtime Database was chosen as the backend database for the application. Fir ebase provides a scalable NoSQL database over the cloud, enabling real-time data synchronization and efficient data storage.

5.1.3 Programming Languages

JavaScript was predominantly used for the frontend development of the application. Its versatility and compatibility across various platforms made it an ideal choice. Additionally, server-side scripting was implemented using Node.js, ensuring a seamless connection between the frontend and the backend components of the application.

5.2 Implementation Strategy

The implementation strategy of the project followed a systematic and milestone-based approach. Key milestones were established to achieve the desired functionality and user experience. These milestones included:

- **Design User Interface**

Develop an aesthetically pleasing and intuitive graphical user interface to ensure user engagement and accessibility.

- **Implement Core Features**

  Develop modules for Live Score, Match Stats, and Ranking, focusing on real-time data updates and accurate statistics representation.

- **Database Integration**

  Integrate the application with the Firebase Realtime Database, enabling seamless data storage, retrieval, and synchronization.

- **User Management**

  Implement user authentication, allowing users to create profiles and customize their experience within the application.

- **Tournament Management**

  Develop a module for tournament management, enabling organizers to create, update, and manage tournaments and fixtures.

- **Graphs and Charts**

  Implement visual representation modules, incorporating graphs and charts for match statistics, enhancing user understanding of the game dynamics.

- **Notification System**

  Integrate a robust notification system to provide users with real-time updates on match scores, player performance, and other relevant events.

5.3 Processing Logic/Algorithm

The processing logic of the Scoreboard Automation Application involves several essential steps to manage live scores, player stats, and match information:

- **Data Collection**

  Collect real-time data from various sources, including official sports databases and APIs, ensuring accuracy and timeliness.

- **Data Processing**

  Process the collected data to extract relevant information, such as live scores, player stats, and match updates.

- **Data Presentation**

  Present the processed data to users in a visually appealing and comprehensible manner, incorporating graphs and charts where applicable.

- **User Interactions**

  Allow users to interact with the presented data, such as selecting specific matches, players, or tournaments for detailed information.

- **Database Management**

  Implement secure data storage and maintenance procedures, ensuring data integrity and accessibility for future reference.

- **Notification System**

  Integrate a notification system to alert users about live match events, score updates, and other relevant information.

- **Multi-Sport Support**

  Implement support for multiple sports, allowing users to switch between different sports seamlessly.

- **User Feedback and Rating**

  Include features for user feedback and rating, enabling users to provide their opinions and enhance the application's overall user experience.

- **Admin Panel Management**

  Develop an admin panel for platform administrators to manage user accounts, match data, and resolve disputes, ensuring smooth operation of the application.

- **Ongoing Maintenance**

  Plan for continuous maintenance, including bug fixes, updates, and support for new features, ensuring the application remains reliable and up to date.

Chapter 6

System Testing and Evaluation

In this chapter, the Scoreboard Automation Application is meticulously evaluated through a series of systematic tests and methodologies to ensure its functionality, usability, and performance meet the highest standards. The testing techniques employed are designed to identify any defects, ensuring a seamless and user-friendly experience.

6.1 Testing Techniques

6.1.1 Graphical User Interface (GUI) Testing

GUI testing ensures the user interface is intuitive and user-friendly. Elements like buttons, layouts, and color schemes are evaluated for consistency and responsiveness. The interface is tested to guarantee that all functions operate as intended, and users can easily interpret texts, graphics, and controls.

Table 6.1: Test Case General GUI

Test Case ID	**Test case 1**	
Description	Tests the graphical user interface	
Compile SDK version	(minSdkVersion 15, targetSdkVersion 29)	
Initial Condition	Equipment is set up as per requirements.	
Steps	Task	**Result**
1.	Open the application.	**PASS**
2.	Verify that all of the tabs are working properly.	**PASS**
3.	Check for the internal links are working properly.	**PASS**
4.	Verify that all the categories are working properly.	**PASS**
5.	Verify that all the categories are working properly.	**PASS**

Table 6.2: GUI Test Case

Test Case ID	**Test case 2**	
Description	Tests the graphical user interface	
Compile SDK version	(minSdkVersion 15, targetSdkVersion 29)	
Initial Condition	Equipment is set up as per requirements.	
Steps	Task	**Result**
1.	Open the categories section.	**PASS**
2.	Verify that all of the buttons are working properly.	**PASS**
3.	Verify that each text field is expecting data properly.	**PASS**
4.	Verify that each selection in every manner is working properly.	**PASS**
5.	Verify that without selection the result would not be generated.	**PASS**

Table 6.2: GUI Test Case

6.1.2 Software Performance Testing

Performance testing assesses the responsiveness, speed, and stability of the application under various workloads. The main objective is to identify system breaking points and ensure the application performs optimally even under heavy usage.

Table 6.3 Performance test case

Test case ID	**Test case 4**	
Description	Tests the performance of the application.	
Compile SDK version	(minSdkVersion 15, targetSdkVersion 29)	
Initial Condition	Equipment is set up as per requirements.	
Steps	Task	**Result**
1.	Open the application.	**PASS**
2.	Verify that the load time of the application is minimum.	**PASS**
3.	Verify that the response time of application to user input is very small.	**PASS**
4.	Verify that the application can handle expected number of users.	**PASS**
5.	Verify that the application works well under slow connections.	**FAIL**

6.1.3 Usability Testing

Usability testing identifies potential issues before coding, allowing for preemptive fixes. It assesses the ease with which users can perform tasks, ensuring the system is user-friendly. This testing method helps in creating an environment to evaluate and design measures during usability testing.

Test Case ID	**Test case 3**	
Description	Tests the usability of system	
Compile SDK version	(minSdkVersion 15, targetSdkVersion 29)	
Initial Condition	Equipment is set up as per requirements.	
Steps	Task	**Result**
1.	Open the application.	**PASS**
2.	Verify that the button icons/texts are relatable by the user.	**PASS**
3.	Verify that user can easily navigate through different pages from the home page.	**PASS**

4.	Verify whether the graphics put emphasis on the most important features users should benoticing right away.	**PASS**
5.	Verify that the user can easily complete all tasks.	**PASS**

6.2 Test Cases for Scoreboard Automation App

6.2.1 Test Case: User Registration

To ensure users can register successfully and gain access to the application.

Table 6.5 Test Case TC-01: User Registration

Test Case ID	TC-01
Description of the Test Case	Users register for the application.
Testing Functionality	User registration process.
Testing Setup	• Internet connectivity available • All required credentials provided. • Server is responsive.
Testing Evaluation	• Enter Email • Enter Password • Enter Confirm Password
Expected Result	User registration successful.
Actual Result	User registered successfully.
Status	Pass

6.2.2 Test Case: User Login

To ensure users can log into the application securely.

Table 6.6 Test Case TC-02: User Login

Test Case ID	TC-02
Description of the Test Case	Users log in to the application.
Testing Functionality	User login process.
Testing Setup	• Internet connectivity available • Correct user email and password provided. • Server is responsive.
Testing Evaluation	• Enter user email • Password • press "Login" button.
Expected Result	User login successful.
Actual Result	User logged in successfully.
Status	Pass

6.2.3 Test Case: Password Recovery

To ensure users can recover their password in case of a forgotten password.

Table 6.7 Test Case TC-03: Password Recovery

Test Case ID	TC-03
Description of the Test Case	Users recover their password.
Testing Functionality	Password recovery process.

Testing Setup	• Internet connectivity available • Email account active
Testing Evaluation	• Click on "Forget Password" button. • Enter user email and click "Send Link." • Open the link from the email and enter a new password.
Expected Result	Password recovery successful.
Actual Result	Password recovered successfully.
Status	Pass

6.2.4 Test Case: Add Project

To ensure users can successfully add a new project to the application.

Table 6.8 Test Case TC-04: Add project

Test Case ID	TC-04
Description of the Test Case	Users add a new project.
Testing Functionality	Project addition process.
Testing Setup	• Internet connectivity available • All project details provided. • Server is responsive.

Testing Evaluation	• Enter project title • description • choose category • claim funds.
Expected Result	: Project added successfully.
Actual Result	Project added successfully.
Status	Pass

Chapter 7

Conclusion

In the development journey of the Scoreboard Automation App, several significant insights have emerged, offering valuable lessons about the intersection of sports, technology, and user experience. Through meticulous planning and execution, the project has not only showcased its strengths but also highlighted areas for improvement and innovation.

7.1 Key Learnings

1. Real-time Engagement: The integration of live score updates and comprehensive match statistics proved to be the heart of user engagement. Real-time data delivery emerged as a fundamental aspect, immersing users in the unfolding events and enhancing their overall experience.

2. User-Centric Design: The emphasis on creating an intuitive and user-friendly interface demonstrated its importance. Navigability and simplicity are key, ensuring that the app is accessible to a wider audience, even those less tech-savvy. A user-centric approach is not just a feature; it's a fundamental philosophy shaping positive user interactions.

3. Data Management Excellence: Efficient data management, including player stats, match records, and rankings, emerged as a cornerstone for sports analysis. Accurate insights are only possible with meticulous data processing. The app demonstrated the need for robust algorithms for ranking and statistical computation, providing reliable and detailed information for users.

4. Timely Communication: The implementation of a notification system proved pivotal. Timely updates keep users engaged, creating a sense of connection to the events. Push notifications ensure that users are informed about match progress, scores, and relevant news, enhancing their active participation.

7.2 Future Directions

Looking forward, the Scoreboard Automation App can further evolve to meet the dynamic demands of sports enthusiasts and the ever-changing landscape of technology. Future enhancements could include:

1. Advanced Graphical Representations: Enhancing graph and chart features can provide users with deeper insights into player and match performance. Visual representations offer a quick and clear understanding, enriching the overall user experience.

2. Enhanced Player Performance Analytics: Investing in sophisticated algorithms for player performance analysis can offer coaches, players, and fans unparalleled insights. Analyzing player movements, strengths, and weaknesses in real-time can revolutionize strategic decisions and training programs.

3. Multi-Sport Expansion: Broadening the app's support to encompass a variety of sports can attract a diverse user base. Each sport has unique aspects, and tailoring the app's features to cater to different games ensures a comprehensive and inclusive sports platform.

4. Social Interaction Features: Incorporating social elements such as chat rooms, forums, and community discussions can foster a sense of belonging among sports enthusiasts. Direct interactions between fans, players, and analysts create vibrant communities within the app, enhancing user engagement and loyalty.

5. Personalized User Experiences: Creating personalized user profiles that allow individuals to track their favorite teams, players, and tournaments can deepen user engagement. Tailored recommendations, customized notifications, and unique user journeys enhance the overall app experience.

6. Integration with Wearable Technology: Exploring integration with wearable devices opens doors to real-time health and performance data. Monitoring players' physical states and vital signs during matches provides holistic insights for players, coaches, and fans alike.

References

[1] D,Linman,2022. [Online] Available at: https://mymanagementguide.com/how-to- write-a-project-description-key-criteria-and-steps/. Last Accessed On 19 October 2022.

[2] Houston,K.2022. *What is a literature review?*. [Online]. Available at: https://guides.library.bloomu.edu/litreview. Last Accessed at 18 October 2022.

[3] play.google.com. *What is scoreboard* [Online] Available at: https://play.google.com/store/apps/details?id=br.com.fiereck.placar. Last Accessed on 25 October 2022

[4] Play.google.com.2022. *What is Cricbuzz* [Online]. Available at: https://play.google.com/store/apps/details?id=com.cricbuzz.android. Last Accessed on 25 October 2022.

[5] Play.google.com.2022. *What is Virtual Scoreboard* [Online]. Available at: https://play.google.com/store/apps/details?id=com.leonardobortolotti.virtualscore board. Last Accessed on 25 October 2022.

[6] discoverphds.com.2022. *Project Scope and objectives* [Online]. Available at: https://www.discoverphds.com/advice/doing/research-aims-and-objectives. Last Ac-cessed on 25 October 2022.

[7] Wardynski,D.J.2022. *scope of a project and its importance* [Online]. Available at: https://www.brainspire.com/blog/how-to-define-the-scope-of-a-project-and-its-importance. Last Accessed on 25 October 2022.

[8] altexsoft.com.2022. *project development Methodology* [Online]. Available at: https://www.altexsoft.com/whitepapers/agile-project-management-best-practices- and-methodologies/. Last Accessed on 25 October 2022

[9] indeed.com.2022 *what is work plan* [Online]. Available at: https://www.indeed.com/career-advice/career-development/work-plan-template . Last Accessed on 25 October 2022

[10] appvizer.com.2022 *what is Team Structure* [Online]. Available at: https://www.appvizer.com/magazine/operations/project-management/project- team-structure. Last Accessed on 25 October 2022

[11] igi-global.com.2022 *What is Work Distribution* — I [Online]. Available at: https://www.igi-global.com/dictionary/work-distribution/32729 . Last Accessed on 25 October 2022

[12]Investopedia.com.2022. *what is gantt chart* [Online] Available at: https://www.investopedia.com/terms/g/gantt-chart.asp. Last Accessed on 25 October 2022