|  |  |
| --- | --- |
| Cricketoon | Software Requirement Specification |
| Version ID Date: 07/02/2023  Document ID: SWD 01  Version ID: 1.0 | |
|  | |
|  | |

Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Doc. Ver. | Date | Author | Reviewer | Description of  Revision |
| 1.0 | 07-02-2023 | Mustafizur Rahman | Nani Gopal Barai,  SarwarMiral |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Contents

[1. Introduction 4](#_Toc2070)

[1.1 Purpose 4](#_Toc8660)

[1.2 Scope 4](#_Toc15947)

[1.3 Intended Stakeholder 5](#_Toc18945)

[1.4 References 5](#_Toc26880)

[1.5 Definitions, Acronyms, and Abbreviations 5](#_Toc21843)

[2. Overall Description 6](#_Toc17754)

[2.1 Overview 6](#_Toc25455)

[2.2 Technical platform 6](#_Toc22302)

[3. Functional Requirements 7](#_Toc2295)

[3.1 Overview 7](#_Toc10410)

[3.1.1. Live Score 7](#_Toc22025)

[3.1.2. Team and player profiles 9](#_Toc25589)

[3.1.3. Search Players 10](#_Toc1482)

[3.1.4. Match schedules 11](#_Toc21614)

[4. Non-Functional Requirements 13](#_Toc8432)

[5.1 Performance Requirements 14](#_Toc21013)

[5.2 Safety Requirements 15](#_Toc17950)

[5.3 Security Requirements 15](#_Toc19911)

[5. Design Constraints 16](#_Toc23788)

[6. Software Quality Attributes 17](#_Toc2256)

[7. User Interface 18](#_Toc22045)

1. Introduction

This section gives a scope description and overview of everything included in this SRS document. Also,the purpose for this document is described and a list of abbreviations and definitions is provided.

* 1. Purpose

The purpose of this document is to give a detailed description of the requirements for the “Cricketoon” software. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. Main objective of Cricketoon is to create a complete mobile application for cricket fans that will offer real-time results, performance reviews. The program will be created to satisfy the requirements of cricket fans

* 1. Scope

The “Cricketoon” is a mobile application which will help cricket enthusiasts to stay updated about recent cricket news and matches. The application will provide users with live scores of cricket matches, performance analysis of player. The application should be free to download from either a mobile phone application store or similar services. User can use the app without creating any account.

Moreover, the software need Internet to fetch and display live result. Some information is maintained in database. When there is no internet connection it will display previous save data. After internet is connected it will update data. The notifies user about upcoming matched 15min earlier.

* 1. Intended Stakeholder

BJIT Academy is the main stakeholder of the project.

* 1. References

|  |  |
| --- | --- |
| Reference | Location |
|  |  |
|  |  |
|  |  |

* 1. Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| Term/Acronym | Definition |
| APP | Application |
| SRS | Software Requirement Specification |
| User | Someone who interacts with the mobile phone application |

1. Overall Description
2. Overview

Cricketoon is a cross-platform mobile application designed for cricket enthusiasts, which will provide live scores, performance analysis, and news updates for cricket matches. The application will be designed with a user-friendly interface and will provide users with a personalized dashboard to track their favourite teams and players.

1. Technical platform

The application will be built with the Sportmonks API and will operate on Android smartphones. The application will be optimized for performance, stability, and dependability and tested to verify that it satisfies the stated quality requirements.

* **Android OS:** This is the open-source mobile operating system used to operate the application on mobile devices.
* **Kotlin:** Kotlin is a cross-platform programming language that runs on the Java Virtual Machine. It is meant to be expressive, concise, and secure.
* **XML:** XML is a markup language used to define the structure of user interfaces. Android defines the layout of its user interfaces with XML, making it simple to develop and adjust the app's appearance.
* **Navigation Component:** The Navigation Component is part of the Android Jet-pack package and is used to manage navigation between app fragments. It facilitates the creation and management of the navigation flow inside an application.
* **Room Database: Room-db** is a popular lightweight database based on SQLite is used for data storage and retrieval. The application will utilize Room-db to store data such as details on cricket matches, players, and teams.
* **Material Design”** The app's UI/UX design will be based on Material Design. These technological platforms provide a robust and adaptable framework for the construction of the cricket software, allowing you to design a high-quality and user-friendly application.
* **Retrofit Library:** Retrofit is a popular Android library for managing HTTP requests and API calls. It makes it easy to obtain data from API and handle network requests within an application

These technological platforms provide a robust and adaptable framework for creating “Cricketoon”, allowing you to design a high-quality and user-friendly application.

1. Functional Requirements
2. Overview

|  |  |  |
| --- | --- | --- |
| Serial No | Main Features | Description |
| 1 | Live Score Updates | The application will provide users with real-time updates of live cricket matches, including scoreboards, match details, and commentary. |
| 2 | Match schedules | The app must display the schedules for upcoming cricket matches, including the date, time, and teams playing. |
| 3 | Team and player profiles | The app must provide detailed information about teams and players, including their current rankings, stats, and career history. |
| 4 | Search Players | Search a player by name and show the player data in a box with picture. Players information will be displayed like Name, Country, Team/Club, Position, Total Runs, Total Wickets, Achievements etc. |

* + 1. Live Score

An important aspect of any cricket news app that offers real time update of ongoing matches. An explanation of the data that will be displayed for each match, including the teams involved, the location, the score, the number of completed overs, the number of wickets lost, and the run rate, a description of the push notifications that users will receive 15minute before the game. The Live Score feature is a vital aspect of a cricket app that delivers users up-to-the-minute information on ongoing matches. The standards for this feature in the SRS document should ensure that the application provides accurate and up-to-date information in a user-friendly and efficient way.

###### Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| REQUIREMNT ID | Requirement Description | Acceptability/  Completion Criteria | Limitations/  Constraints | Test case Identifier |
| Cricketoon \_001 | Show scores of live matches | Each match's information display must include the teams involved, the location, the score, the number of overs completed, the number of wickets taken, the run rate etc | It's possible that not all games will have real-time live score information accessible, the live score data accuracy could not be entirely accurate, A quick network connection is necessary for the Live Score feature to perform properly etc. |  |
| Cricketoon \_002 | Show Match stats(Scoreboard, Venue,Match Result,Man of the Match) | The match stats (Scoreboard,Venue,Man of the match)must be accurate, readable and user-friendly structure. Such as tables, graphs. | The match stats might not be available for all matches, The match statistic' accuracy varies depending on the source and could not be exact etc. |  |
| Cricketoon \_003 | Show Playing Squad | The squad lists for each team must be accurate, The team squad lists must be presented in a way that is easy to read and navigate.The team squad listings must allow for player, position, and other search parameters etc | It's possible that not all teams have access to or that the team squad listings are out-of-date, Depending on the source, the team squad lists may not be entirely accurate etc. |  |

* + 1. Team and player profiles

###### Team and player profiles are an essential aspect of a cricket news app, providing in-depth information about the teams and players involved in the sport. This feature should give users access to a wealth of information about their favorite teams and players, including their statistics, performance history, and current form. With this information at their fingertips, users can stay up-to-date on the latest update in the world of cricket and get a better understanding of the teams and players they follow. The requirements for this feature in the SRS document should ensure that the profiles are comprehensive, easy to access, and regularly updated to reflect the latest information.

###### Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| REQUIREMNT ID | Requirement Description | Acceptability/  Completion Criteria | Limitations/  Constraints | Test case Identifier |
| Cricketoon \_004 | Show country-wise players | The Team and Player Profiles feature should allow users to view players sorted by their country of origin. Accurate display of player information sorted by country in a user-friendly and efficient manner. | Inaccuracies in player information and the requirement for a quick network connection for proper performance. |  |
| Cricketoon\_005 | Show Position-Wise Players | The Team and Player Profiles feature should allow users to view players sorted by their positions (e.g. batsman, bowler, etc.). Accurate display of player information sorted by position in a user-friendly and efficient manner | Potential inaccuracies in player information and the requirement for a quick network connection for proper performance |  |
| Cricketoon\_006 | Show Ranking Wise Players | The Team and Player Profiles feature should allow users to view players sorted by their current rankings (e.g. top batsman, top bowler, etc.). Accurate display of player information sorted by ranking in a user-friendly and efficient manner | Inaccuracies in player information and rankings, as well as the requirement for a quick network connection for proper performance |  |

* + 1. **Search Players**

A cricket app's search players feature enables users to look up details about certain cricket players. It makes player lookup easier.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **REQUIREMNT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| Cricketoon\_007 | Find player by name | Users must be able to utilize the tool to look up players by typing their names into any order in search field etc. For example Babar Azam if user type Azam it will show Babar Azam’s name along with other players. | All players, especially for those from lesser-known clubs or leagues, The player information in the app might come from outside sources, so there might be data flaws or mistakes that the app is powerless to fix etc. |  |
| Cricketoon \_08 | Show the player data in a box with picture | Upon conducting a search, player information including profile photo should be displayed in a clear and well-organized layout. Accurate display of player information in response to user searches in a user-friendly and efficient manner | This may include - The accessibility of player data can place a cap on the feature. The feature might not work as intended if correct player data is not available or is difficult to collect etc. |  |

* + 1. Match schedules

Users of a cricket app can view a summary of recently finished cricket matches as well as the Scoreboard for each match by using the "Recent Matches with Scoreboard" option. Some key details are - the most recent games and Scoreboard information must come from a trustworthy source and be correct, The data from the Scoreboard and recent games must be presented in a way that is easy to read and use, like tables or lists, The teams engaged, the match's location, the date, and the match's outcome must all be included in the information about recent matches. The number of runs, wickets, and overs for each team must be listed on the Scoreboard, The Scoreboard user interface for recent matches must be simple to use, pleasing to the eye, and straightforward etc. Users now have a quick and easy way to check the results of recent cricket matches and compare the performances of other teams thanks to the Scoreboard feature that displays recent matches. This function can improve the cricket app's user experience and increase adoption and popularity

###### Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| REQUIREMNT ID | Requirement Description | Acceptability/  Completion Criteria | Limitations/  Constraints | Test case Identifier |
| Cricketoon \_009 | Show summary of recent matches | A trustworthy source must be used to compile an accurate report of recent matches, The teams engaged, the match's location, the date, and the outcome must all be listed in the summary of recent matches etc. | It's possible that not all games will have real-time live score information accessible, the live score data accuracy could not be entirely accurate, A quick network connection is necessary for the Live Score feature to perform properly etc. |  |
| Cricketoon \_010 | Show match stats like scoreboard, squad list, venue, date, match results, MoM etc | The match stats (Scoreboard,Venue,Man of the match)must be accurate, readable and user-friendly structure. Such as tables, graphs. | The match stats might not be available for all matches, The match statistic' accuracy varies depending on the source and could not be exact etc. |  |

1. Non-Functional Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| REQUIREMNT ID | Requirement Description | Acceptability/  Completion Criteria | Limitations/  Constraints | Test case Identifier |
| Cricketoon\_NFR\_001 | User-Friendliness | Provision of a user-friendly interface that is easy to navigate and access features. | Need for user testing to determine actual user-friendliness and the possibility of design limitations. |  |
| Cricketoon\_NFR\_002 | Accuracy | Accurate information on all relevant aspects of the app | Potential for inaccuracies in the information provided, such as changes in schedules or player statistics. |  |
| Cricketoon\_NFR\_003 | Efficiency | Real-time updates and information in a timely and efficient manner. | The apps need for a quick network connection for proper performance, as well as the possibility of delays in information updates. |  |
| Cricketoon\_NFR\_004 | Performance | Optimal performance and a smooth user experience, even under heavy usage and with a large number of concurrent users. | The apps need for ongoing performance testing and optimization, as well as the possibility of performance degradation with heavy usage. |  |
| Cricketoon\_NFR\_005 | Responsiveness | Responsive UI that adapts to different devices and screen sizes, providing a consistent and optimal user experience. | Need for ongoing testing on different devices and screen sizes, as well as the possibility of design limitations. |  |

* 1. Performance Requirements

In the cricket app, performance is a critical aspect as it directly impacts the user experience. In this section, the performance characteristics of the system are outlined to ensure that the app performs optimally and provides a smooth user experience.

* **Response Time:** The average response time for transactions on the cricket app must be less than 3 seconds, with a maximum response time of 5 seconds.
* **Throughput:** The cricket app must be able to process an average of 50 transactions per second, with a maximum of 100 transactions per second.
* **Capacity:** The cricket app must be able to accommodate at least 10,000 concurrent users, with the potential for scaling to accommodate increased demand.
* **Degradation Modes:** In the event of system degradation, the cricket app must continue to provide essential features, such as live scores and player information, with acceptable performance degradation.
* **Resource Utilization:** The cricket app must utilize system resources efficiently, including memory, disk, and communications, to ensure optimal performance and prevent degradation. The app must be designed to minimize resource usage and allow for efficient scaling as needed.
  1. Safety Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| REQUIREMNT ID | Requirement Description | Acceptability/  Completion Criteria | Limitations/  Constraints | Test case Identifier |
| Cricketoon \_011 | Data privacy and security | The app must use encryption and secure data storage to protect sensitive information from unauthorized access. | Security measures should be periodically reviewed and updated to keep up with the evolving threat landscape. |  |
| Cricketoon\_012 | Data Backup | Regular backups of user data must be performed to prevent data loss in case of system failure or other issues | The backup process may take time, user data may still be lost in the event of a major system failure, backup data may not be easily accessible in the event of a system failure |  |

* 1. Security Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| REQUIREMNT ID | Requirement Description | Acceptability/  Completion Criteria | Limitations/  Constraints | Test case Identifier |
| Cricketoon\_013 | Cryptography | Implement encryption for sensitive data | Encryption may slow down the app performance and may increase storage usage. |  |
| Cricketoon\_014 | Logging | Maintain a log of all system activities including user search, modification to data and security events | The log data must be secure and tamper-proof |  |
| Cricketoon\_015 | Data Integrity | Verify the integrity of critical variables to prevent data tampering | The verification process must be efficient and reliable |  |

1. Design Constraints

Design constraints are limitations or restrictions that influence the design of a software system, in this example,

* **Platform Limitations:** The Kotlin programming language is mostly utilized for Android development, which might affect the design of the cricket app if certain functionalities are unavailable on Android.
* **Performance Requirements:** The design of the cricket application must consider for performance constraints like as reaction time and throughput, which may have an effect on the design if particular algorithms or data structures are too computationally costly for the hardware on which the application will operate.
* **Integration with Other Systems:** The cricket app must interface with other systems, such as databases and online API services, the design must take into account the unique constraints and limitations of such systems, such as data format and API compatibility.
* **Security Requirement:** The design of the cricket app must account for security requirements, such as data encryption and secure data transfer, which may have an effect on the design if certain encryption methods or security protocols are unavailable in Kotlin.
* **Development Environment:** The design of the cricket application must take into account the development environment, including the tools and libraries available in Kotlin as well as any constraints or compatibility difficulties associated with these tools.

1. Software Quality Attributes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| REQUIREMNT ID | Requirement Description | Acceptability/  Completion Criteria | Limitations/  Constraints | Test case Identifier |
| Cricketoon\_016 | Usability | The application must meet the expectations of its target users, allowing for a smooth user experience. | The design may need to be adapted for users with disabilities, the interface should work seamlessly on different platforms and screen sizes |  |
| Cricketoon\_017 | Reliability | The application should be able to perform its intended functions and be available at all times, with minimal errors or bugs. | The system's reliability may be impacted by factors such as network connectivity and server uptime. |  |
| Cricketoon\_018 | Performance | The application must meet the performance requirements outlined in the SRS document, such as response times and throughput | The performance of the system may be impacted by factors such as network connectivity, server capacity, and the number of users accessing the system at the same time. |  |
| Cricketoon\_019 | Security | The system must implement appropriate security measures to protect user data and information from cyber threats | The system's security may be impacted by factors such as network vulnerabilities, the use of insecure third-party components, and the risk of user account hijacking. |  |

1. User Interface

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **UI No.** | **UI Name** | **Related Info No.** | **Description** | Test case Identifier |
| Cricketoon-20 | Homepage |  | The app's home screen shows cricket results, and Live Score |  |
| Cricketoon-21 | Team and Player Info |  | This screen will display the detail information of player and team |  |
| Cricketoon-22 | Detail Screen |  | This screen will display the details info(Venue, Man of the Match, commentary Etc) of a specific match |  |

This section defines any other requirements not covered elsewhere in the SRS. This might include database requirements, budget requirements, schedule requirements internationalization requirements, legal requirements and so on. Feel free to add any new sections that are pertinent to the project.

This section specifies those requirements that are concerned with possible loss, damage, or harm that could result from the use of the sub-system. It defines any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the sub-system design or use. Define any safety certifications that must be satisfied.