| **<<CRICKET ROCKS>>** | **Software Requirement Specification** |
| --- | --- |
| Version ID Date: 16/09/2014  Document ID: SWD/…….. ??  Version ID: 1.0 | |
|  | |
|  | |

**Revision History**

| **Doc. Ver.** | **Date** | **Author** | **Reviewer** | **Description of Revision** |
| --- | --- | --- | --- | --- |
| 1.0 |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Contents**

[**1.**](#_heading=h.1ci93xb) **Introduction 4**

[1.1](#_heading=h.3whwml4) Purpose 4

[1.2](#_heading=h.2bn6wsx) Scope 4

[1.3](#_heading=h.qsh70q) Intended Stakeholder 4

[1.4](#_heading=h.3as4poj) References 4

[1.5](#_heading=h.1pxezwc) Definitions, Acronyms, and Abbreviations 4

[**2.**](#_heading=h.49x2ik5) **Overall Description 5**

[2.1](#_heading=h.2p2csry) Overview 5

[2.2](#_heading=h.147n2zr) Technical platform 5

[**3.**](#_heading=h.2s8eyo1) **Functional Requirements 5**

[3.1](#_heading=h.3o7alnk) Overview 5

[3.1.1.](#_heading=h.23ckvvd) Feature/Function 1 6

[3.1.2.](#_heading=h.lnxbz9) Feature/Function 2 6

[**4.**](#_heading=h.35nkun2) **User Interface 7**

[**5.**](#_heading=h.ihv636) **Non-Functional Requirements 7**

[5.1](#_heading=h.32hioqz) Performance Requirements 7

[5.2](#_heading=h.1hmsyys) Safety Requirements 8

[5.3](#_heading=h.41mghml) Security Requirements 8

[**6.**](#_heading=h.2grqrue) **Design Constraints 8**

[**7.**](#_heading=h.vx1227) **Software Quality Attributes 9**

[**8.**](#_heading=h.4i7ojhp) **User Interface 9**

[**9.**](#_heading=h.2xcytpi) **Other Requirements 9**

1. **Introduction**

From this section, we can get an overview of everything which has been implemented in this SRS document. Moreover, this section gives a scope description. Besides, in this section the description for the purpose of this document is shown as well as the list of definitions and abbreviations is provided.

* 1. **Purpose**

The purpose of this document is to give a description in a very detailed way for the requirements as well as functionality, performance etc. of “CRICKET ROCKS” which is a mobile application. Moreover, the functional requirements like - user interfaces, data storage, networking capabilities etc. is outlined by the SRS document. Besides, the SRS document also includes the non-functional requirements like - performance, scalability, compatibility etc. of the application. At last, we can say that, the main purpose of the SRS document is to give service as a guide for the stakeholders, development team and users so that the final product meets their expectations.

* 1. **Scope**

The scopes of this mobile application are given below -

* Functionality : Information about cricket matches, scores, statistics, news etc. would be provided by the mobile application.
* User Interface : A user-friendly interface would be present here which will give advantage to users to easily navigate and access information.
* Data Sources : Reliable sources would be used so that the app can provide up-to-date and accurate information on cricket matches.
* Platform Support : The mobile application would support on iOS platform.
* Analytics : Analytics on user behavior and app usage would be provided by the app so that the user experience would be improved.
* Performance : Good performance would be provided by the app, with minimal downtime and fast load times.
* Maintenance : The app would be updated on a regular basis so that the bugs and security vulnerabilities would be addressed as well as new features would be added.
  1. **Intended Stakeholder**

The intended stakeholders of the mobile application are given below -

* End Users : Basically, the end users are those who would use the app to access match schedules, match updates and other information.
* Advertiser : Advertisers are those who would pay for in-app advertisements which will help to reach the app user base.
* Development Team : They are directly related in building the app by using the SRS document as a blueprint.
* Mobile App Stores : A platform would be provided by this for the app for download which will help to earn a potential commision for app purchasing or advertising revenue.
* Cricket Organization : The data for the app would be provided by the cricket organizations and benefit would also be provided from increased engagement with cricket fans.
  1. **References**

| **Reference** | **Location** |
| --- | --- |
| Requirement Specification |  |
|  |  |
|  |  |

* 1. **Definitions, Acronyms, and Abbreviations**

| **Term/Acronym** | **Definition** |
| --- | --- |
| Cricket | A very popular game all over the world played between two teams of 11 players each |
| Match | A contest between two teams |
| Live Score | Up-to-date score of a live cricket match played between two teams |
| UI | User Interface |
| API | Application Programming Interface |
| Developer | People who are directly connected in developing the application by coding, testing and other activities |
| Stakeholder | People who are directly or indirectly involved in the project |
| Feature | Functionalities of the app |
| iOS | Operating System related to iPhone |

1. **Overall Description**
2. **Overview**

The goal, features, and intended audience of the app are briefly described in this part of the Software Requirements Specification (SRS) document. This part should give readers a thorough knowledge of the app's goals, target audience, and what the app will have to offer. A typical overview section may include the following information :

* Purpose : A clear explanation of the app's goal, such as "The goal of this app is to give cricket fans real-time information on live matches, player statistics, news articles, and other pertinent information," would serve as a good example.
* Scope : A description of the features and services the app will offer, such as push notifications, news articles, live scores, match schedules, player data, and live scores.
* Technical specifications : A breakdown of the app's technical specifications, including the hardware it will run on, the operating systems it will support, and any required software elements.
* Key features : A summary of the app's most important features and capabilities, including push alerts, live scores, match schedules, player data, and news articles.
* User interface : A description of the screen layouts, navigation, feedback, and accessibility of the app's user interface.
* Development Team : Project managers, developers, designers, and testers are included in the development team; their roles are also described.

The SRS document's Overview section offers a succinct synopsis of the app and lays the groundwork for the next, more in-depth requirements. It helps to make sure that everyone involved is aware of what the app is meant to do, who it is meant for, and what it will provide consumers.

1. **Technical platform**

In a Software Requirements Specification (SRS) document for a cricket app, the Technical Platform section outlines the app's technical specifications, including the hardware it will run on, the operating systems it will support, and any required software components. These details could be found in a typical Technical Platform section:

* Devices : A list of the mobile phones, tablets, and PCs that the app will be compatible with.
* Operating systems : The platforms that the software will work with, including Android, iOS, and Windows.
* Hardware specifications : The hardware necessities for the app, including the needed processing speed, memory, and storage capacity.
* Software components : A description of any software elements, such as databases, APIs, or libraries, that are necessary for the app to operate.
* Integration : A description of any platforms, services, or systems that the app will be integrated with, such as cloud services, payment gateways, or social media sites.
* Security : Information on the app's security requirements, including data encryption, user authentication, and access control.

As it lays the groundwork for the creation and testing of the app, the Technical Platform portion of the SRS document is a crucial component of the requirements specification. It aids in ensuring that the required software components and integrations are in place, the app will operate as expected on the target devices and operating systems.

1. **Functional Requirements**
2. **Overview**

<This section sums up in the below table the main functionalities or services provided by the sub-system, which will be detailed in the following subsections. A use case diagram could be also used to list the main functionalities.>

| **Serial No** | **Main Features** | **Description** |
| --- | --- | --- |
| 1 | Live score | * Show scores of live matches * Show team squad list * Show match stats like ScoreBoard, Venue, Man of the Match, Match Result etc |
| 2 | Recent matches with Scoreboard | * Show summary of recent matches * Show match stats like scoreboard, squad list, venue, date, match results, MoM etc |
| 3 | Fixtures | * Show whole fixture of the tournament * Show country wise/team-wise players * Show date-wise fixture |
| 4 | Search Players | * Search a player by name * After searching 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 |
| 5 | Winning Percentage | * Calculate winning percentage based on current data * Set the logic to produce winning percentage and change it when data is coming from API |
| 6 | Get Statistic | * Show data like Top performer of the tournament, most runs for a batsman, team-wise ranking etc * Show some advanced level data according to your understanding |

* + 1. **Live Score**

A crucial aspect of a cricket app that offers real-time updates on active matches is the Live Score feature. The information shown below is what the Software Requirements Specification (SRS) document for this feature can contain : a description of the sources, such as official sources, APIs, or live score providers, from which the live score data will be collected, the rate of updating for the live score information, such as every ball, every over, or every wicket, An explanation of the data that will be shown for each encounter, including the teams involved, the location, the score, the number of completed overs, the number of wickets lost, the run rate, a description of the push notifications that users will receive in response to events like a wicket, a century, or the beginning of a game, a description of the layout, navigation, and any necessary images or animations for the Live Score feature's user interface etc. An integral part of a cricket app that gives users up-to-the-minute information on active matches is the Live Score feature. The SRS document's specifications for this feature ought to guarantee that the app delivers correct and current information in a user-friendly and effective manner.

###### **Requirements**

| **REQUIREMNT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| <CRICKET ROCKS \_001> | Show scores of live matches | The requirement to display live match scores in a cricket app has a set of requirements that must be met in order to be deemed satisfied. These requirements are known as the Acceptability/Completion Criteria. Some examples of Acceptability/Completion Criteria are - The real-time score information must come from a dependable source and be correct, 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. | The variables or factors that will affect or hinder the implementation of the need to display live match scores in a cricket app are known as limitations or constraints. Some examples are - It's possible that not all games will have real-time live score information accessible, Depending on the source, the live score data's accuracy could not be entirely accurate, A reliable and quick network connection is necessary for the Live Score feature to perform properly etc. |  |
| <CRICKET ROCKS \_002> | Show team squad list | The requirement to display team squad lists in a cricket app has a set of requirements that must be accomplished in order to be deemed complete. These requirements are known as the Acceptability/Completion Criteria. Some examples may be - The squad lists for each team must come from a trustworthy source and be accurate, The team squad lists must be presented in a way that is easy to read and navigate, like a table or list, The team squad listings must allow for player, position, and other search parameters etc. | The conditions or elements that will affect or restrict the implementation of the requirement to display team squad lists in a cricket app are known as limitations or constraints. Some examples may be - 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. |  |
| <CRICKET ROCKS \_003> | Show match stats like ScoreBoard, Venue, Man of the Match, Match Result etc | The Completion/Acceptability There are a number of requirements that must be accomplished in order for the requirement to display match stats in a cricket app to be deemed satisfied, including Scoreboard, Venue, Man of the Match, Match Result, and others. Some examples are - The match statistics must be accurate and come from a reputable source, A readable and user-friendly structure, such as tables, graphs, or charts, must be used to present the match statistics etc. | The conditions or elements that will affect or restrict the implementation of the requirement to display match stats like Scoreboard, Venue, Man of the Match, Match Result, etc. in a cricket app are referred to as limitations or constraints. Some examples are - The match stats might not be current or they might not be available for all matches, The match statistic' accuracy varies depending on the source and could not be exact etc. |  |

* + 1. **Recent Matches With Scoreboard**

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** |
| --- | --- | --- | --- | --- |
| <CRICKET ROCKS \_004> | Show summary of recent matches | The requirement to display a summary of recent matches in a cricket app has a set of requirements that must be completed in order to be deemed complete. These requirements are known as the Acceptability/Completion Criteria. Some examples are - 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. | The conditions or elements that will affect or restrict the implementation of the need to present a summary of recent matches in a cricket app are known as limitations or constraints. Some examples are - It's possible that not all matches will have a summary available, or that the summary may not be current, The accuracy of the summary of recent matches is depending on the source and may not be 100% accurate etc. |  |
| <CRICKET ROCKS \_005> | Show match stats like scoreboard, squad list, venue, date, match results, MoM etc | The Completion/Acceptability There are a number of requirements that must be followed in order for match stats like ScoreBoard, Squad List, Venue, Date, Match Results, Man of the Match (MoM), etc. to be displayed in a cricket app. Some examples are - The match statistics must be accurate and come from a reputable source, The teams participating, the location and date of the encounter, the ScoreBoard, the Squad List, and the Match Results must all be included in the match stats etc. | The conditions or elements that would affect or restrict the implementation of the requirement are referred to as the Limitations/Constraints of the demand to present match stats like ScoreBoard, Squad List, Venue, Date, Match Results, Man of the Match (MoM), etc. in a cricket app. Some examples are - The match stats might not be current or they might not be available for all matches etc. |  |

* + 1. **Fixtures**

A cricket app's fixtures function gives users a list of forthcoming matches for a specific tournament or series. Some key details are - The schedule data has to come from dependable and current sources, including official cricket boards or tournament organizers, The teams competing, the location, the date and time of the event, and any additional information, such as the contest's format, should all be listed in a clear and simple manner in the fixtures information (e.g., T20, ODI, Test), With a user-friendly interface that enables users to search and filter the fixtures based on several criteria, such as tournament, teams, and date, the fixtures information should be simple to obtain and browse, To guarantee that users have access to the most recent and correct information, the fixtures data should be updated in real-time etc. The development team has a clear knowledge of the requirements and expectations for the fixtures feature thanks to the inclusion of these crucial details in the SRS document. This data enables the development team to create a feature that satisfies the demands of the stakeholders while taking into account any potential roadblocks or restrictions.

###### 

###### Requirements

| **REQUIREMNT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| <CRICKET ROCKS \_006> | Show whole fixture of the tournament | The Completion/Acceptability The requirements for the requirement to display the entirety of a tournament's fixture on a cricket app must be satisfied in order for the requirement to be deemed complete and acceptable. Some examples are - The fixture data needs to be correct and current, The teams participating, the match's location, the date, and the time must all be included in the fixture data etc. | Limitations or constraints imposed by the requirement to display every game in a tournament in a cricket app could affect the design and implementation of the functionality. Some examples are - It's possible that the fixture data won't be available for every competition, especially for lesser-known or smaller competitions, The fixture information could change because of unforeseen circumstances, bad weather, or other considerations etc. |  |
| <CRICKET ROCKS \_007> | Show country wise/team-wise players | What makes a successful implementation of a feature is specified by the Acceptability/Completion Criteria of the requirement to display country- or team-specific players in a cricket app. Some examples are - The player information must be correct, current, and reflect the most recent alterations to the player's status, including transfers, retirements, etc, The player data must have all pertinent details, including the player's name, position, country, and other pertinent information etc. | The limits or limitations that must be taken into account when implementing the feature are described in the Limitations/Constraints of the requirement to present country- or team-based players in a cricket app. Some examples are - The player information must come from dependable and trustworthy sources, including the cricket boards' official websites or recognized sports news organizations, The player information needs to be kept in a safe database that can handle high user traffic and data volumes etc. |  |
| <CRICKET ROCKS \_008> | Show date-wise fixture | The Completion/Acceptability The parameters that must be completed in order for the feature to be deemed finished and prepared for distribution are described in the criteria of the requirement to present date-wise fixtures in a cricket app. Some examples are - The date-wise fixture feature must have an easy-to-use interface that makes it possible for users to traverse and comprehend the data, The date-wise fixture information must be correct, current, and derived from reputable sources etc. | The restrictions or difficulties that might affect the creation, testing, or implementation of the functionality are described in the limitations/constraints of the demand to show date-wise fixtures in a cricket app. Some examples are - Depending on the dependability of the data sources and the promptness of changes, the availability of precise and up-to-date fixture information may be constrained, The user's network connection's quality and dependability may have an impact on how well the date-wise fixture feature performs, which could lead to lengthy loading times and a bad user experience etc. |  |

* + 1. **Search Players**

A cricket app's search players feature enables users to look up details about certain cricket players.

###### Requirements

| **REQUIREMNT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| <CRICKET ROCKS \_009> | Search a player by name | The "Search a player by name" requirement's acceptability/completion criteria for a cricket app would specify the requirements that must be satisfied for the feature to be deemed successfully implemented. Some examples are - Users must be able to utilize the tool to look up players by typing their names into a search field etc. | The limitations may include - The feature might be constrained by the accessibility of player data, and the app might not be able to provide comprehensive information for 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. |  |
| <CRICKET ROCKS \_010> | After searching show the player data in a box with picture | This may include - Users must be able to search for players by name, and the functionality must show player information, including a profile photo, in a box with a clear layout on the screen etc. | 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. |  |
| <CRICKET ROCKS \_011> | Players' information will be displayed like Name, Country, Team/Club, Position, Total Runs, Total Wickets, Achievements etc | This may include - The presentation should be clear and succinct and include all the needed information (Name, Country, Team/Club, Position, Total Runs, Total Wickets, Achievements, etc.) etc. | This may include - The information sources utilized to feed the app could place restrictions on the player data that is accessible, Some data, such as player statistics or accomplishments, may be arbitrary or subject to different interpretations etc. |  |

* + 1. **Winning Percentage**

Users of a cricket app could be able to view the win-loss record of a certain team or player thanks to the Winning Percentage feature. This may include - a summary of all the games, along with the number of victories and defeats, an estimated winning % depending on the overall quantity of games played, the capacity to look up a team's or player's win-loss record, the capacity to filter the data depending on a particular time frame or contest (e.g., test matches, one day internationals, T20s, etc.), a chart or graph that shows the win-loss record graphically etc. This feature aims to give users a quick and simple way to assess a team's or player's performance over a given time frame.

###### Requirements

| **REQUIREMNT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| <CRICKET ROCKS \_012> | Calculate winning percentage based on current data | This may include - Based on the most recent statistics, an accurate assessment of the win-loss record is made, the capacity to look up a team's or player's win-loss record, an estimated winning % depending on the overall quantity of games played etc. | This may include - The availability and dependability of match data will have an impact on how accurately the calculation is performed. The proportion could not be exact if some pieces of data are missing or obsolete etc. |  |
| <CRICKET ROCKS \_013> | Set the logic to produce winning percentage and change it when data is coming from API | This may include - It is important to put the rationale behind determining a team's winning percentage into practice, the logic should update the winning % in accordance with the most recent information obtained from the API. | This may include - Data from the API availability and accuracy: The calculation of the winning percentage is reliant on the accuracy and dependability of the data coming from the API. The calculation of the winning percentage might not be correct if the data is not current or full etc. |  |

* + 1. **Get Statistic**

A cricket app's "Get Statistics" feature is used to show statistical information about cricket matches and players. This feature may contain various statistical information about players and teams, such as averages, strike rates, rankings, etc. For better presentation, the data might be presented in graphical or tabular formats. The data is often retrieved from a database or API that is regularly updated with the most recent data. The Acceptability/Completion Criteria for this feature may include demands for the precise representation of the most recent statistical data, an intuitive user interface, and quick data retrieval. This feature's restrictions and limits may include data availability and accuracy, restrictions on the API, and compatibility with various devices.

###### 

###### 

###### Requirements

| **REQUIREMNT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| <CRICKET ROCKS \_014> | Show data like Top performer of the tournament, most runs for a batsman, team-wise ranking etc | This may include - The information is presented in an accurate, easily readable manner, When fresh information is received from the API, the data is updated instantly, Based on the most recent information available, the ranking and performance figures are calculated. | This may include - The information displayed must be correct and current, The data source needs to be trustworthy and able to deliver real-time data etc. |  |
| <CRICKET ROCKS \_015> | Show some advanced level data according to your understanding | This may include - The information shown should be pertinent to the cricket competition and offer users useful insights, The information should be presented in an orderly and understandable way etc. | Thai may include - The availability of information from the sources consulted to compile advanced level data, The sources used may have a limit on the degree of accuracy of the data obtained and shown for the advanced level data etc. |  |

1. **User Interface**

| **UI No.** | **UI Name** | **Related Function Req ID** | **Description** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| <Application name \_001> |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. **Non-Functional Requirements**

| **REQUIREMNT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| <Application name \_001> |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

* 1. **Performance Requirements**

<The performance characteristics of the system should be outlined in this section. Include specific response times. Where applicable, reference related Use Cases by name.

Response time for a transaction (average, maximum)

Throughput (e.g., transactions per second)

Capacity (e.g., the number of customers or transactions the system can accommodate)

Degradation modes (what is the acceptable mode of operation when the system has been degraded in some manner)

Resource utilization: memory, disk, communications, etc.>

* 1. **Safety Requirements**

| **REQUIREMNT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| <Application name \_001> |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

* 1. **Security Requirements**

<Specify the factors that protect the software from accidental or malicious access, use, modification, destruction or disclosure. Specific requirements may include the need to

* utilize certain cryptographic techniques
* Keep specific log or history data sets.
* assign certain functions to different modules
* restrict communications between some areas of the program
* check data integrity for critical variables>

| **REQUIREMNT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| <Application name \_001> |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. **Design Constraints**

<This section should indicate any design constraints on the system being built. Design constraints represent design decisions that have been mandated and must be adhered to. Examples include software languages, prescribed use of developmental tools, architectural and design constraints, purchased components, class libraries, etc.>

1. **Software Quality Attributes**

| **REQUIREMNT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| <Application name \_001> |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. **User Interface**

| **UI No.** | **UI Name** | **Related Info No.** | **Description** | **Notes** | **Test case Identifier** |
| --- | --- | --- | --- | --- | --- |
| <Application name \_001> |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1. **Other Requirements**

| **REQUIREMNT ID** | **Requirement Description** | **Acceptability/**  **Completion Criteria** | **Limitations/**  **Constraints** | **Test case Identifier** |
| --- | --- | --- | --- | --- |
| <Application name \_001> |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |