Software Requirements Specification

for

Cricket Tournament Management System (CTMS)

Version 1.0 approved

Prepared by Yashwant Krishna (20CS30036)
Sarthak Nikumbh (20CS30035)
Anuj Kakde (20CS30005)

IIT Kharagpur

16-03-2022

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1. Purpose

The main objective of this document is to illustrate the requirements of the project Cricket Tournament Management System (CTMS). The document gives the detailed description of the both functional and non-functional requirements. The purpose of this project is to provide a simple and basic field to store and maintain the details of the tournament matches, teams and players. The main purpose of this project is to maintain an easy system to provide various details of the stored entities*, calculated statistics* etc.

1.2. Document Conventions

- To denote sub-headers, dot separated numbers are used, For example: 1.2.1 is a sub-header of 1.2 which is a sub-header of 1.

- An asterisk after a word indicates that its information is in the glossary

- Convention for Main title
  - Font face: Times New Roman
  - Font style: Bold
  - Font size: 18 pt
- Convention for Sub title
  - Font face: Times New Roman
  - Font style: Bold
  - Font size: 14 pt
- Convention for body
  - Font face: Arial
  - Font size: 11 pt

1.3. Intended Audience and Reading Suggestions

CTMS* is intended for the developers and cricket enthusiasts.

User generates the necessary data which includes the statistics of the players, teams and matches.

Users can get all the statistics of the entities that is uploaded to the database. Users can also view the statistics of the entities.

This is a suggested sequence to go through the SRS*:

- Product Scope
- Product Perspective
- Product Functions
- User Classes and Characteristics
- System Features
- Appendix B
- Others Nonfunctional Requirements

1.4. Product Scope

CTMS* provides with the statistics of the players, teams and matches, which allows the users to view the statistics of the entities mentioned above.

The project can be easily implemented under various situations. New features can be added when required, making re-usability possible as there is flexibility in all the modules. The language used for developing the project is Java as it is quite advantageous than other languages in terms of performance, tools available, cross platform compatibility, libraries, cost (free of cost), and development process.

The future releases of the project's versions can contain

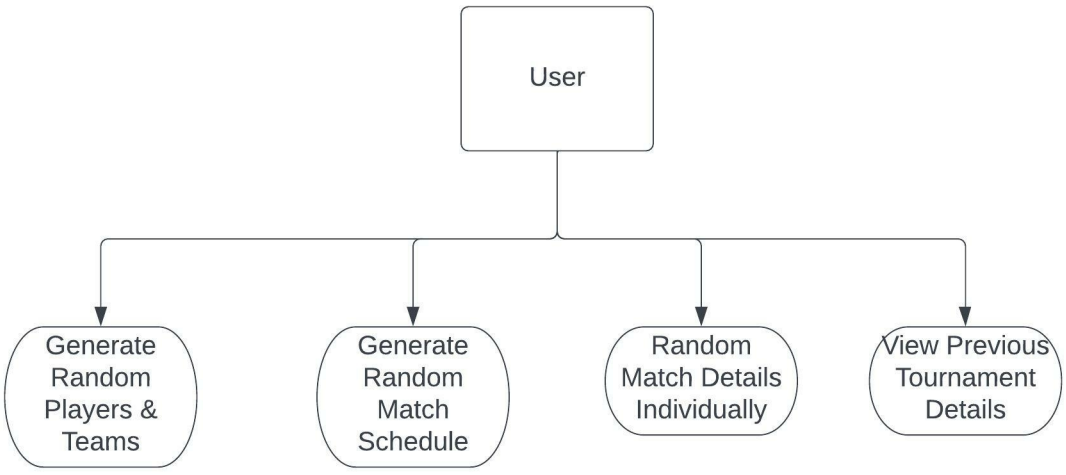
- an online storage of data
- live score updates that can be accessed from any device provided a good internet connection
- two types of users: viewers and tournament manager (administrator)
- tournament manager has administrator rights, who only can upload the statistics of the entities, match scores etc.

1.5. References

- https://realpython.com/token-based-authentication-with-flask/
- https://developer.android.com/training/basics/firstapp
- https://www.javatpoint.com/android-tutorial
- https://developer.android.com/training/data-storage/sqlite

2. Overall Description

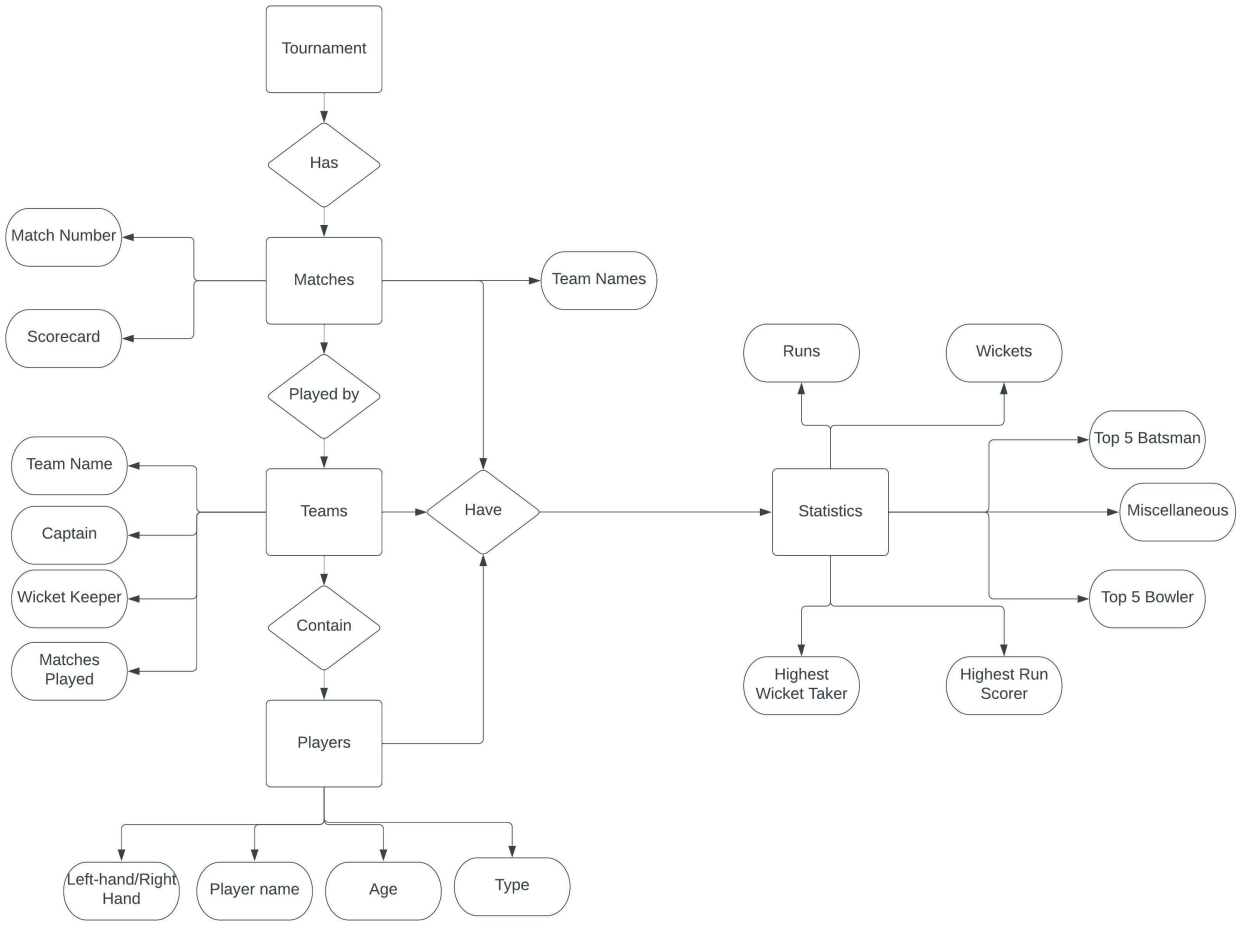
2.1. Product Perspective



The product is a new self-contained product which aims to automate the Cricket Tournament Management System, such as

- calculating the statistics of the entities in the tournament
- generating match schedule
- qualify the winners of the tournament to the higher levels such as playoffs, semifinals and finals.

2.2. Product Functions



The major functions of CTMS* are:

- Random generation of player and team statistics:
  - user randomly generates team statistics
- Generation of match schedule:
  - user generates the match schedule in a round robin system*
- Random generation of match statistics:
  - user randomly generates match statistics
- View statistics of the entities:
  - user can navigate the corresponding entities' page to view its statistics
- View match scoreboard:
  - user can view the scoreboard of any match in the tournament

2.3. User Classes and Characteristics

The system provides different types of services based on the type of users. The User can be a cricket enthusiast who can view the statistics of the players as well as teams and every match's scoreboard, or a developer that generates random test cases to test the correctness of the project.

The features that are available to the users are:-

- Randomly generating the teams of the tournament
- Randomly generating the players and their statistics for each team
- Generating the match schedule for the tournament
- Can view a player's statistics
- Can view a team's statistics
- Can view a match's statistics

2.4. Operating Environment

The software operates on all android devices with version 5.1 (lollipop) and above. CTMS* is an offline android app that opens on specific android devices and all the features will be compatible with devices with version 5.1 and above. No internet connection is needed to run this application.

The hardware configuration includes an android device with version 5.1 and above, any screen layout as the application is made to fit all screen devices.

2.5. Design and Implementation Constraints

Design constraints applicable to mobile app development:

- **Screen size:** Multiple device size compatibility, Android comes with multiple screen sizes and multiple device resolutions
- **No hover state:** The user can't hover over a GUI* to know its functional
- **Error-Prone typing:** Typing on a touch keyboard is far slower compared to that of a desktop or a laptop.
- **Inaccurate clicks:** Users use their fingers to click on the links of buttons which might be of small size might click on the wrong button or GUI.
- **Running out of memory:** When working with high resolution images developers need to create cache controller.
- **Software fragmentation:** There are multiple Android OS* versions which developers find hard to keep up with when it comes to app development. It is impractical to focus only on the most recent version as not all users may have upgraded to the most recent OS.

Implementation constraints of the project:

- **Offline, not online:** The project is set on a mobile platform which is not online, that is, it doesn't sync the data of a particular tournament across the apps on various devices. The tournament manager and user can only operate on only *one* device.

2.6. User Documentation

- Click on the "Create Tournament" button to start a new tournament
- Click on the "Generate Random Teams" button to randomly generate the teams for the tournament
- Click on the "Generate Random Players" button to randomly generate the players for every team

- Click on the “Generate Random Statistics” button to randomly generate the statistics of every match, team and players.
- The tournament has been completed with the given statistics of the entities and the user can view the results of the tournaments.
- On the home page, there are the list of tournaments ordered from newest to oldest. Click on the tournament to view the statistics and results.
- On the tournament page, there are list of matches held according to the match schedule. Click on the match to view the statistics.
- On the mach page, there are two teams whose match is held, and the details of the match, this page also has the buttons of the two teams, when clicked shows the statistics of the players of that team

2.7. Assumptions and Dependencies

The assumptions are:

- The coding should be error free
- The system should be user-friendly so that it is easy to use for the users
- The statistics of all the entities should be correctly stored in a database that is accessible by the application.
- The system should have enough RAM* to perform quick operations
- The system should have enough storage capacity and provide fast access to the database.
- The system is offline which means there is no need for an internet connection.
- Users can access any information regarding the statistics of the players, teams and matches held in the tournament.

The dependencies are:

- The specific hardware and software due to which the product will be run
- On the basis of listing requirements and specification the project will be developed and run
- The end users should have proper understanding of the product
- The statistics of all the entities must be stored in a database that is accessible by the CTMS*.

3. External Interface Requirements

3.1. User Interfaces

The software provides a minimalist good looking graphical interface* for the user and the administrator (Tournament manager in this case). The administrator can operate on the system, perform the required tasks such as uploading the statistics of the entities such as players, teams and matches.

It allows the user to:

- view list of tournaments
- view the list of matches in each tournament and their statistics
- view the teams that played in every match and their statistics
- view the players of each team and their statistics
- randomly generate the statistics of the players, teams and the matches for the tournament

All the modules provided with the software must fit into this graphical user interface and accomplish to the standard defined. The design is minimalist, simple and user friendly and interacts with the user.

3.2. Hardware Interfaces

Any android device running version 5.1 and above can run the software. The application requires access to device storage. The database which stores the statistics of the entities is stored in the device itself.

Software constantly stands by to await user input, at which point it then performs the necessary action by sending a signal to the hardware.

Hardware performs the required action by accessing memory stored as bits on memory chips. There is a chain of command the information goes through to finally return back to the software to then do what the user intended.

3.3. Software Interfaces

Language in which the project is coded: **Java**
Operating system that supports the project: **Android devices with version 5.1 and above**
IDE used to code the project: **Android Studio**
Database: **SQLite***

3.4. Communications Interfaces

No communication interfaces are required for the working of the software.

4. System Features

4.1. Randomly generate player, team and match details

4.1.1 Description and Priority

The user can randomly generate player statistics such as the player's name, age, roles, bowling type etc., team statistics such as name of the team, list of players and captain.

Priority: high

Random generation of statistics are used to test the software and there is no user input for every test case which saves a lot of time for the developers.

4.1.2 Stimulus/Response Sequences

Stimulus: User asks to create a new tournament

Response: System creates a new tournament

Stimulus: User randomly generates the list of teams in the tournament
Response: System asks for the user for the number of teams playing in the tournament generates the teams and random generation of its data.

Stimulus: User randomly generates the list of players for each team

Response: System generates the random statistics of each player and a list of players for every team

Stimulus: User generates the match schedule

Response: System generate a match schedule based on the number of teams in a round robin system.

Stimulus: User generates random statistics of each match

Response: System generates random statistics of each player of both the teams playing in a match and decided the winner of qualifier according to the randomly generated statistics

4.1.3 Functional Requirements

Generate teams: User enters the number of teams in the tournament and randomly generate them

Generate players: Random generation of players' statistics of each team

Generate match schedule: generation of match schedule given the list of teams in a round robin manner

Generate statistics: generation of each player's statistics of every match

4.2. View statistics of entities:

4.2.1 Description and Priority

The user can view the statistics of the entities.

Priority: high

4.2.2 Stimulus/Response Sequences

Stimulus: User selects a tournament out of the total tournaments held

Response: System opens a window of that tournament and shows a list of all the matches are held in the tournament sorted from newer to older and the statistics of the tournament.

Stimulus: User selects a match out of the list of matches of the tournament

Response: System opens a window of that match and shows the teams between which the match is held, and statistics of the match.

Stimulus: User selects a team out of the list of teams that played in the tournament

Response: System opens a window of that team and shows its statistics.

4.2.3 Functional Requirements

View tournament statistics: User can select the tournament to view its statistics and the list of matches

View match statistics: User can select the match to view its statistics and the teams between which the match was held

View player statistics: User can select any team from the list of teams to view the list of players and their data and statistics

5. Other Nonfunctional Requirements

5.1. Performance Requirements

- The performance of the system should be smooth, fast and accurate
- CTMS* should handle expected and non-expected errors in ways that prevent loss in information and long downtime period. Thus it should have inbuilt error testing to identify invalid information or data of the entity.
- The system should be able to handle large amount of data as there can be multiple tournaments with each having multiple matches. Thus it should accommodate high number of tournaments, matches, teams and players.

5.2. Safety Requirements

The database may get crashed at any certain time due to bugs in operating system, virus or operating system failure. Therefore, it is required to take the database backup so that the database is not lost.

5.3. Security Requirements

- System will use secured database
- After the generation of players, teams and match schedule, the user is not allowed to modify or change any of the entity values of that particular tournament, the user can only view the tournament results, the statistics of the entities.

5.4. Software Quality Attributes

- The project should be open source
- The quality of the database is maintained in such a way so that it can be very user friendly to all the users of the database
- The user will be able to download and install the system easily on their device.

5.5. Business Rules

A rule can enforce business policy, make a decision, or infer new data from existing data. This includes the rules and regulations that the system users should abide by. This includes the cost of the project and the discount offers provided. The users should avoid illegal rules and protocols and should not cross the rules and regulations.

6. Other Requirements

For management of slots, every day exactly 1 match is assumed to be played.

Appendix A: Glossary

CTMS: Cricket Tournament Management System

GUI: Graphical User Interface

UI: User Interface

OS: Operating System

RAM: Random Access Memory

SQLite: Structured Query Language; used to retrieve information from a database

Entity: The software's major components are the players, teams and matches. These are the entities of the software

Interface: Something used to communicate across different mediums

Round robin system: The matches are schedules such that each team gets to play with every other team once

SRS: System Requirement Specifications

TBD: To Be Determined

Statistics: Statistics of the entities such as-

Player statistics: Statistics of the player include:

- name
- age
- left-handed/right-handed
- roles – bowler, batsman, wicket-keeper, all-rounder
- details specific to their role: fast bowler/spin bowler

Team statistics: Statistics of the team include:

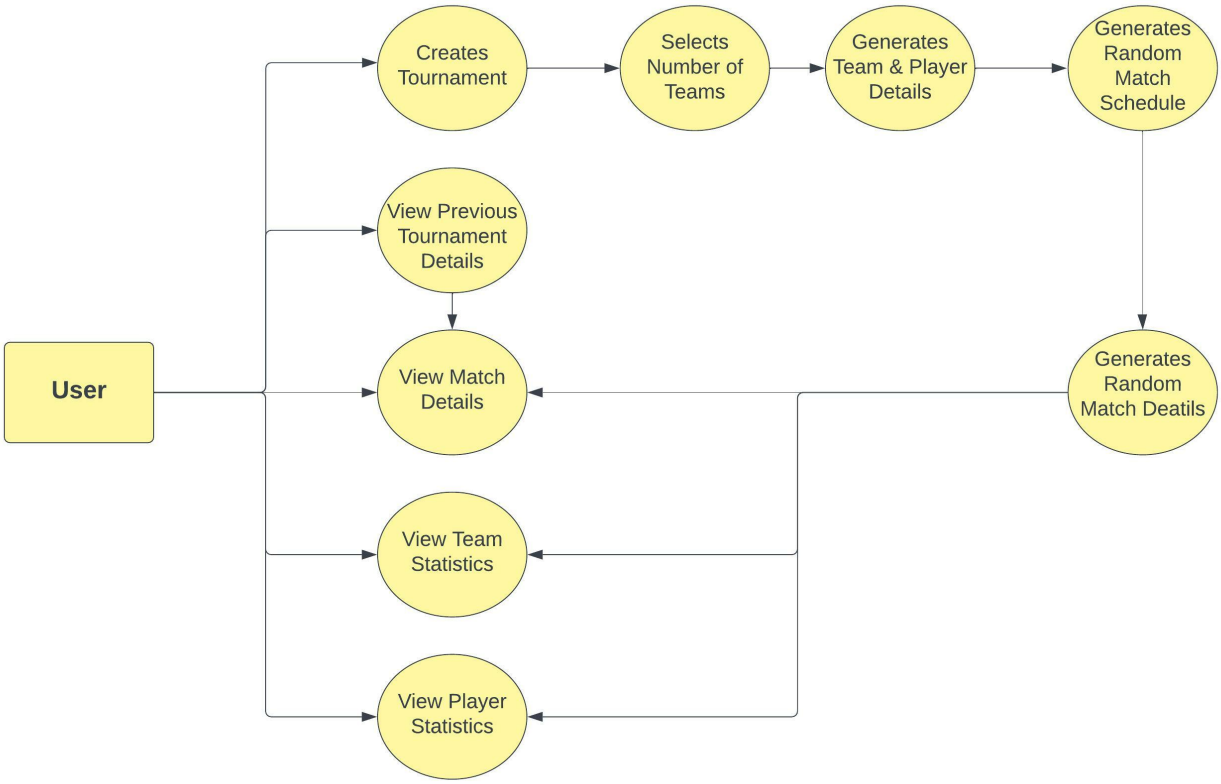
- name
- list of players
- captain

Match statistics: Statistics of the match include:

- two teams between which the match was held
- match result
- team batted first
- runs scored by each player
- balls faced by each player
- wickets taken by each player
- overs bowled by each player

Appendix B: Analysis Models

Use case diagram:



Class diagram: TBD*

Appendix C: To Be Determined List

- Class diagram is to be determined