Software Requirements Specification

March 26 2024

Cricket Tournament Management System (CTMS)

Gaurang Gupta(22CS30025)

Vaasu Singla(22CS10078)

Arudeep Kumar(22CS10011)

Submitted in partial fulfilment of the requirements of CS29202

Software Engineering Laboratory

Table of Contents

Introduction

Purpose:

The purpose of the Cricket Tournament Management System (CTMS) project is to develop a comprehensive software solution for managing and storing data related to cricket tournaments, such as the Indian Premier League (IPL) or the Cricket World Cup. This Software Requirements Specification (SRS) document serves as a comprehensive guide for the development team, outlining both functional and non-functional requirements, along with external interface specifications, necessary for the successful implementation of the system.

Document Conventions:

This Software Requirement Specification Document has been written using Free writing tools such as MS WORD typed in Calibri font. The font size used is 11 for text . All headings are highlighted appropriately in bold. The document is prepared using the UK English convention.

Intended Audience and Reading Suggestions:

This document is intended for the development team, project stakeholders, and anyone involved in the design, development, or testing phases of the Cricket Tournament Management System. It is recommended to read the document sequentially to gain a comprehensive understanding of the system requirements.

Product Scope

This software aims to automate and streamline the management of tournament-related information, including details of players, teams, and match outcomes. By using a database to store vast amounts of data, the CTMS will facilitate the generation of various statistics and summaries, enhancing the understanding and analysis of tournament dynamics. The software will support functionalities such as the random generation of player and team details, the creation of a random match schedule adhering to a round-robin format, the generation of random match details, and the viewing of detailed statistics for players, teams, and matches. Additionally, the system also includes optional features for viewing overall tournament statistics, such as the top wicket-takers and run-scorers.

References:

This document is based on the IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specification given by the IEEE Computer Society in 1998.

Structure of the Document

The document is organized into several key sections, each focusing on different aspects of the system:

1. **Introduction:** Provides an overview of the system, including its purpose and intended users.
2. **Overall Description:** Describes the general factors that affect the system and its requirements, offering a background for the requirements specified later.
3. **System Features:** Details the specific functionalities of the system, including user interactions and data management processes.
4. **External Interface Requirements:** Outlines the interfaces between the system and its users, other systems, and hardware, detailing the requirements for user interfaces, software interfaces, and communication interfaces.

5. **Functional Requirements:** Specifies the functional aspects of the system, defining what the system should do to meet the needs of its users.

2. Overall Description

2.1 Product Perspective

The Cricket Tournament Management System (CTMS) is envisioned as a comprehensive tool designed to automate and streamline the management of cricket tournaments, reminiscent of globally recognized formats like the Indian Premier League (IPL) or the ICC Cricket World Cup. By integrating detailed management of players, teams, and match statistics into a single platform, the CTMS provides a holistic view of tournament operations, significantly reducing the manual effort involved in such endeavors.

2.2 Product Functions

The key functionalities provided by the Cricket Tournament Management System include:

- Random Team generation
- Random assignment of players to the teams
- A round-robin based random match scheduling
- Player details such as right-hand or left-hand batsman/bowler, captaincy, etc.
- Generation of match details with randomized outcomes
- Calculation and display of comprehensive statistics for players, teams, and matches
- Display of tournament-wide statistics, such as top run-scorers and wicket-takers

2.3 Operating Environment

The Cricket Tournament Management System is a web-based application compatible with modern web browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge. It requires a stable internet connection for access.

2.4 Assumptions and Dependencies

The Cricket Tournament Management System assumes users possess basic knowledge of web browsing and form interaction. Dependencies include a stable internet connection and modern web browsers compatible with HTML5 and CSS3. This ensures a wide range of users can effectively interact with the system without the need for specific hardware or software, beyond a general-purpose computing device capable of running up-to-date web browsers.

3. System Features

3.1 Design and Implementation Constraints

3.1.1 Technology Stack Limitations

- The CTMS is built using the Flask framework for Python, which imposes constraints on the choice of programming languages and frameworks for development.
- Dependencies and libraries compatible with Flask and Python need to be carefully selected to avoid conflicts and ensure system stability.

3.1.2 Database Compatibility

- CTMS utilizes a SQLite database for data storage and management. This imposes constraints on scalability compared to more robust database systems like MySQL or PostgreSQL.
- Complex queries or high concurrency may pose challenges in SQLite, requiring optimization strategies and careful database design.

3.1.3 Web-Based Application Constraints

- As a web-based application, CTMS relies on modern web technologies and browsers such as Google Chrome, Mozilla Firefox, and Microsoft Edge.
- Compatibility issues with older browsers or non-standard web configurations may limit the system's accessibility and functionality.

3.1.4 Network and Internet Dependency

- CTMS requires a stable internet connection for access, which can be a constraint in areas with limited or unreliable connectivity.
- Network latency and bandwidth limitations may impact the responsiveness and performance of the web application, especially during peak usage periods.

4. External Interface Requirements

4.1 User Interfaces

4.1.1 Reset Page

- **Purpose:** Enables users to reset the database, reinitializing the system for new data.
- **Functionality:** Users are prompted for confirmation to reset the database. Upon agreement, the system deletes the existing database and initializes a new one with the necessary schema, excluding match details.
- **Navigation:** Post-reset, users are directed to the Front Page, the central navigation hub.

Step by Step Description:

1. User opts to reset the database.
2. System drops the table with teams.
3. System drops the table with match details.
4. System drops the table with player details.
5. System creates new tables for team, match schedule, and player details.
6. System randomly allocates players to teams.
7. System fills the match schedule table with a randomly generated schedule.

4.1.2 Front Page

- **Purpose:** Acts as the application's main navigation hub.
- **Functionality:** Contains tabs for Teams, Matches, Tournament, and Players, directing users to detailed pages for each category.
- **Navigation:** Users select tabs to access specific sections.

4.1.3 Teams Page

- **Purpose:** Displays an overview of all participating teams.
- **Functionality:** Lists teams with logos and outcomes of their last five matches. Team selection leads to the Team Details Page.
- **Navigation:** Selecting a team provides detailed information about it.

4.1.4 Team Details Page

- **Purpose:** Offers in-depth information on a selected team.
- **Functionality:** Shows team composition and match history, with player details interactive, leading to the Player Details Page.
- **Navigation:** Player selection offers detailed player statistics.

4.1.5 Player Details Page

- **Purpose:** Provides detailed statistics for a selected player.
- **Functionality:** Includes comprehensive player information and performance metrics.
- **Navigation:** Accessed through Team Details Page or Players Page.

4.1.6 Players Page

- **Purpose:** Showcases all tournament players.
- **Functionality:** A grid of player photos, with selection leading to detailed player information.
- **Navigation:** Selecting a photo takes the user to the Player Details Page.

4.1.7 Matches Page

- **Purpose:** Lists the tournament's match schedule.
- **Functionality:** Displays a round-robin schedule, with match selection leading to detailed outcomes.
- **Navigation:** Users view specific match details by selecting a match.

4.1.8 Match Details Page

- **Purpose:** Details outcomes and statistics of a chosen match.
- **Functionality:** Shows winning team, scores, and individual player statistics.
- **Navigation:** Accessed by choosing a match from the Matches Page.

4.1.9 Tournament Page

- **Purpose:** Highlights tournament key statistics and achievements.
- **Functionality:** Displays top scorers and wicket-takers with performance summaries.
- **Navigation:** Directly accessible from the Front Page for an overview of tournament metrics.

4.2 Software Interfaces

The CTMS utilizes the Flask framework and interacts with a SQLite database, leveraging SQL for data management.

4.3 Hardware Interfaces

As a web application, CTMS requires standard web server and database hardware, without the need for specialized interfaces.

4.4 Communication Interfaces

CTMS operates over HTTP/HTTPS, ensuring access from anywhere with an internet connection, utilizing standard web protocols for client-server communication.

5. Functional Requirements for the Cricket Tournament Management System (CTMS)

FR1: Database Initialization and Reset

- **FR1.1:** The system must offer an option to reset the database upon startup or through a designated user interface.
- **FR1.2:** On opting to reset the database, the system must delete the existing database file and create a new, empty database.
- **FR1.3:** The system must automatically initialize the newly created database with the required schema for storing player details, team compositions, and match schedules.

FR2: Player and Team Data Generation

- **FR2.1:** Upon database initialization, the system must generate random player details, including name, age, role (e.g., bowler, batsman), and role-specific attributes (e.g., bowling type).
- **FR2.2:** The system must automatically create teams, assigning the generated players to these teams in a balanced manner based on their roles.
- **FR2.3:** Each team must consist of exactly 11 players, without considering substitutes or extras.

FR3: Match Schedule Generation

- **FR3.1:** The system must generate a round-robin match schedule ensuring each team plays against every other team exactly once.
- **FR3.2:** The generated match schedule should not include match details such as outcomes and player performances.

FR4: Dynamic Match Detail Generation

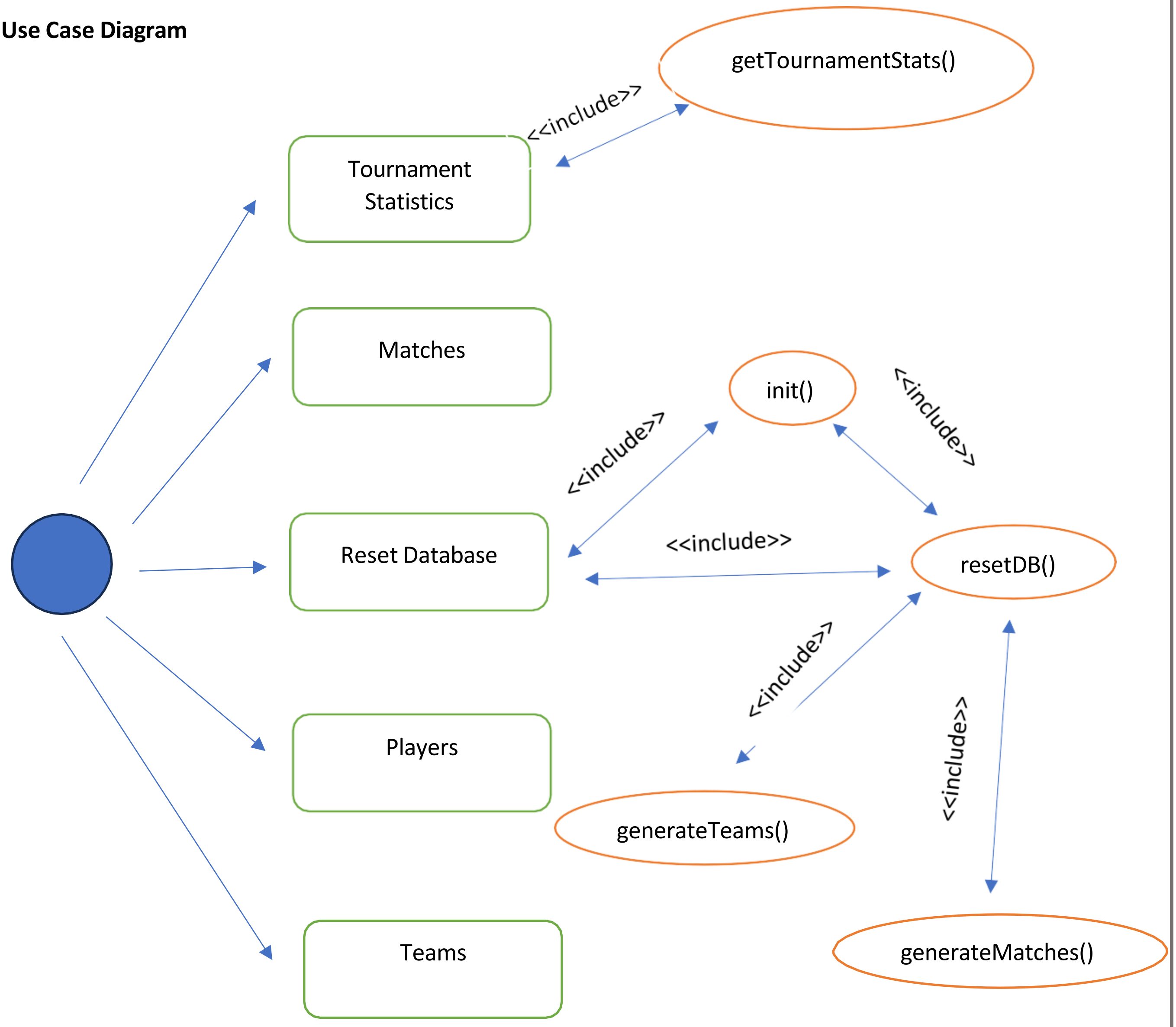
- **FR4.1:** The system should not generate match details at the time of database initialization or reset.
- **FR4.2:** Match details should be generated dynamically when the user selects a particular match from the schedule.
- **FR4.3:** Upon generating match details, the system must update the database with the outcomes, including runs scored, wickets taken, and individual player performances.

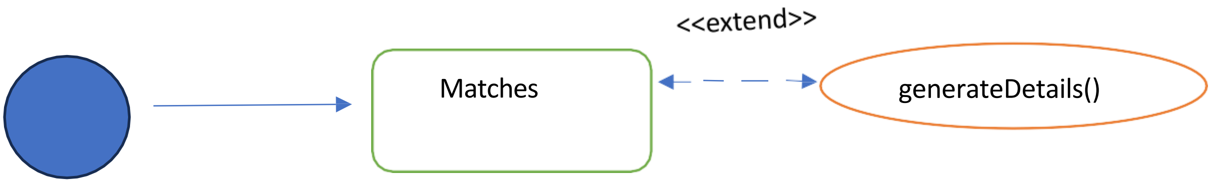
FR5: User Interaction and Interface Requirements

- **FR5.1:** The system must provide an interactive web interface allowing users to choose to reset the database.
- **FR5.2:** Users must be able to view the match schedule and select any match to generate and view its details.

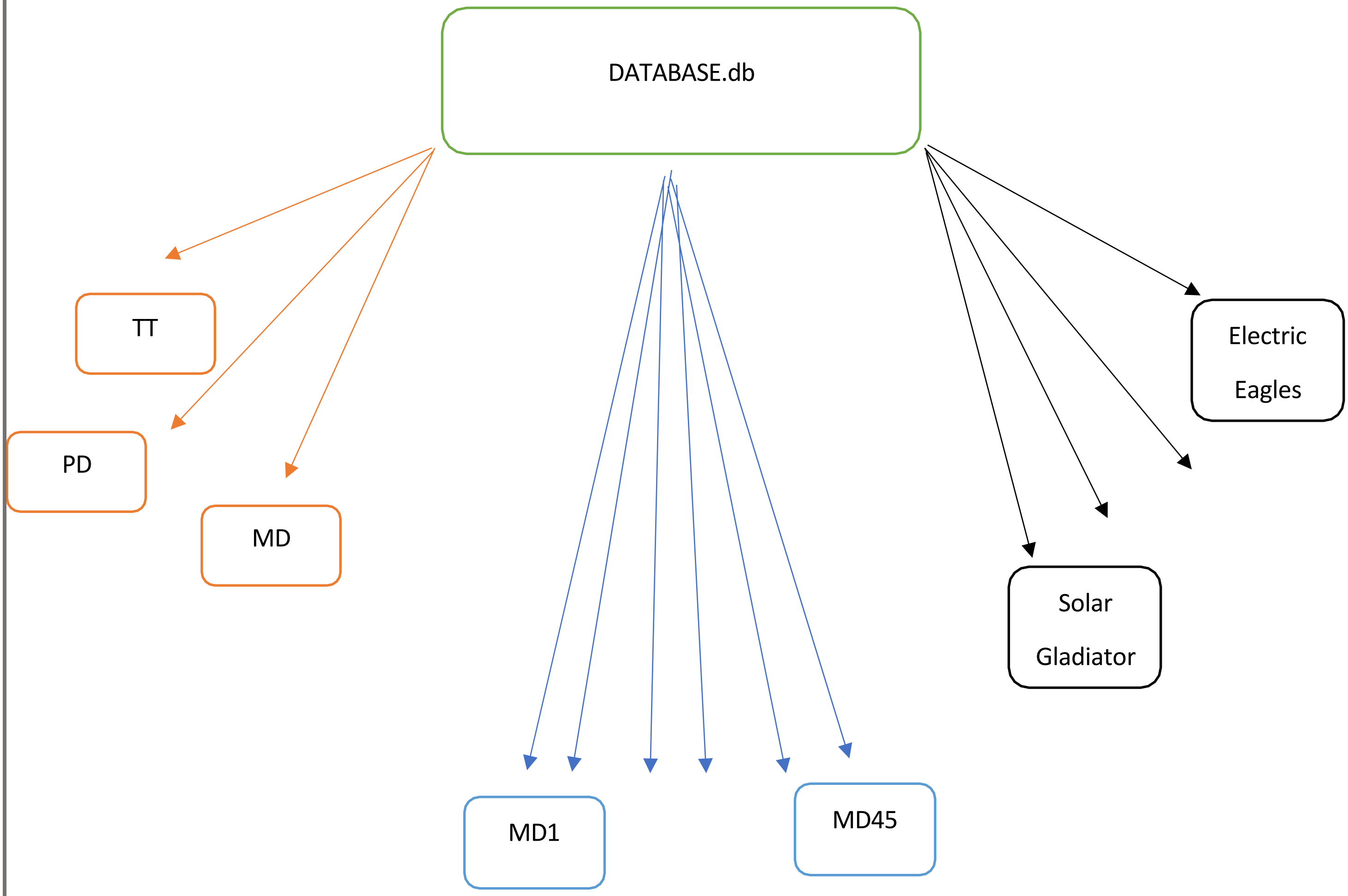
- **FR5.3:** The system must redirect users to a detailed view of the match outcomes and player performances upon selecting a match.

Use Case Diagram





CLASS DIAGRAM


DATABASE.db
TT
PD
MD
MD1
MD45
Electric
Eagles
Solar
Gladiator