**ACKNOWLEDGEMENT**

I want to thank everyone who has helped and supported me throughout my college project, **Crick Informer.** First and foremost, a big thanks to my guide, **Mr. Sachin Subnis,** for giving me valuable guidance and pointing me in the right direction. Who helped me learn a lot about this project. **His/Her** ideas and comments aided in the completion of this project.

I also take the opportunity to acknowledge the contribution of Dr. Kavita T. Rangari, Head of Department of BBA(CA), for her full support and assistance during the development of the project.

I am also grateful to our wonderful principal, Dr. M.S.Wazal Sir, who has created a nurturing and creative environment in our college. I would also like to acknowledge my friends who contributed their ideas and perspectives, which greatly enriched the project. Thank you all!

Signature of the Student

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr.No.** | **Particular Name** | **Page No.** |
| **1.** | **Abstract Introduction:** |  |
|  | 1.1 Motivation |  |
|  | 1.2 Problem Statement |  |
|  | 1.3 Purpose/Objectives and Goals |  |
|  | 1.4 Literature Survey |  |
|  | 1.5 Project Scope and Limitations |  |
| **2.** | **System Analysis:** |  |
|  | 2.1 Existing Systems |  |
|  | 2.2 Scope and Limitation of Existing  System |  |
|  | 2.3 Project Perspective, Features |  |
|  | 2.4 Stakeholder |  |
|  | 2.5 Requirements Analysis: -   * Functional Requirements * Performance Requirements * Security Requirements |  |
| **3.** | **System Design:** |  |
|  | 3.1 Design Constraints |  |
|  | 3.2 System Model: DFD |  |
|  | 3.3 Data Model |  |
|  | 3.4 User Interface |  |
| **4.** | **Implementation Details:** |  |
|  | 4.1 Hardware Specifications |  |
|  | 4.2 Software Specifications |  |
| **5.** | **Output and Report Testing** |  |
|  | 5.1 Test Plan |  |
|  | 5.2 Black Box Testing |  |
|  | 5.3 White Box Testing |  |
|  | 5.4 Validation Test Cases and Result |  |
| **6.** | **Conclusion & Recommendations** |  |
| **7.** | **Future Scope** |  |
| **8.** | **Bibliography And References** |  |

1.**INTRODUCTION**

**Motivation:**

* + Cricket is a globally beloved sport, and enthusiasts often find it challenging to access comprehensive and real-time information about matches.
  + The motivation behind Crick Informer is to bridge this gap, providing a centralized platform for cricket enthusiasts to access live updates and historical match data effortlessly.

**Problem Statement:**

* + Limited accessibility to real-time cricket match updates.
  + Absence of a consolidated source for historical match data

**Purpose/Objective & Goal**:

* **Purpose:**
  + To create a centralized platform for cricket enthusiasts to access live and historical match information seamlessly.
* **Objectives:**
  + Provide real-time updates for ongoing cricket matches.
  + Offer an extensive database for exploring historical match data.

**Goal**

**Project Goal:**

* + Establish Crick Informer as a go-to platform for cricket information.
  + Enhance the cricket-watching experience by offering comprehensive and easily accessible data.

**Literature Survey:**

* + Examined existing cricket-related platforms and applications.
  + Investigated technologies commonly used in similar projects.
  + Reviewed literature on user preferences and challenges in accessing cricket information.

**Project Scope**

1. **1.Live Score Updates:** The project will provide users with live scores of ongoing matches, keeping them updated with the latest match information in real-time.
2. **Match Cards for Past Matches:** Users will have access to match cards containing data from past matches, allowing them to review details such as scores, and other relevant information.
3. **Point Table for Recent CWW:** The project will provide users with a point table for the recent Cricket World Cup (CWW), showcasing the standings of teams based on their performance in the tournament

**Limitations:**

* + Real-time updates might be subject to delays in data sources.
  + Historical match data might not cover every cricket event due to data availability constraints

**2. System Analysis**

* 1. **Existing Systems**: Traditional cricket coverage primarily relies on television broadcasts, radio commentary, and sports news websites

**2.2 Scope and Limitation of Existing Systems:**

* **Scope:** Current systems provide basic match updates and limited historical data.
* **Limitations:** Lack of real-time updates, limited historical data access, potential for misinformation. Existing systems offer match scores but may not provide in-depth historical match information or real- time updates

**2.3 Project Perspective and Features:**

* **Project Perspective:** Crick Informer" is a web application that aims to provide cricket enthusiasts with a comprehensive platform for accessing live match updates and historical match information. Leveraging Spring Boot for the backend and Angular for the frontend, this project combines robust backend services with a dynamic and user-friendly interface.

**Project Features:**  
  
 Live match updates.

* Historical match information.
* Backend using Spring Boot for robust services.
* Frontend using Angular for a dynamic and user-friendly interface.

**2.4 Stakeholders**

* Cricket enthusiast.
* Cricket Players.
* Team Coach.
* Admin

**2.5 Requirement Analysis:**

-Functional Requirement  
 **1. Live Score Updates**

* 1. The system provides real-time updates for ongoing cricket matches.
  2. Users are able to view live scores, match events (e.g., wickets, boundaries).

**2. Match Cards for Past Matches**

1. The system display’s match cards containing data from past cricket matches.

2. Users are able to access information such as match results, scores, teams.

1. **Point Table for Recent Cricket World Cup (CWW)**
   1. The system generates and display a point table for the recent Cricket World Cup (CWW).
   2. The point table shows the standings of teams based on their performance in the tournament.
   3. Users are able to view the points earned, matches played, wins, losses, and net run rate for each team.

-Performance Requirements  
 **1. Response Time**

* 1. The system shall respond to user interactions, such as clicking on a match card or updating favorite teams, within 1 second.
  2. This includes the time taken to fetch and display data from the backend server.

1. **Data Retrieval**
   1. The system shall retrieve historical match information, including match cards and point tables, within 2 seconds.
   2. This includes querying the database and processing the data for display on the frontend.

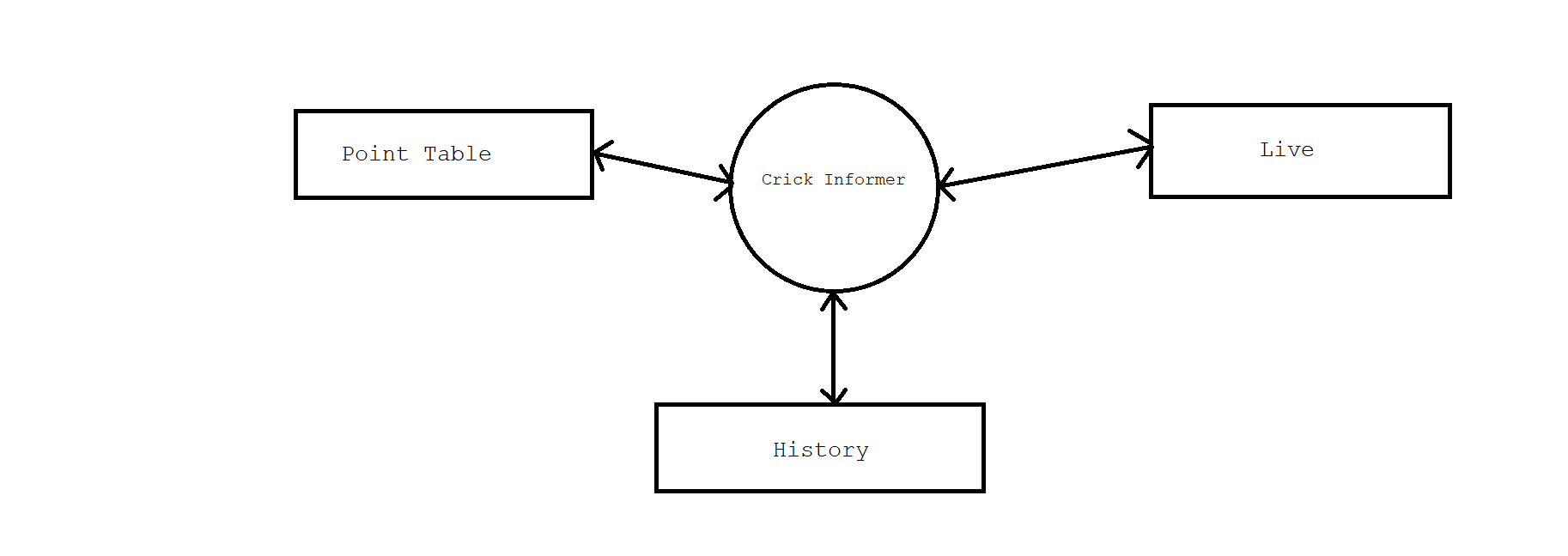
**3.User Experience**

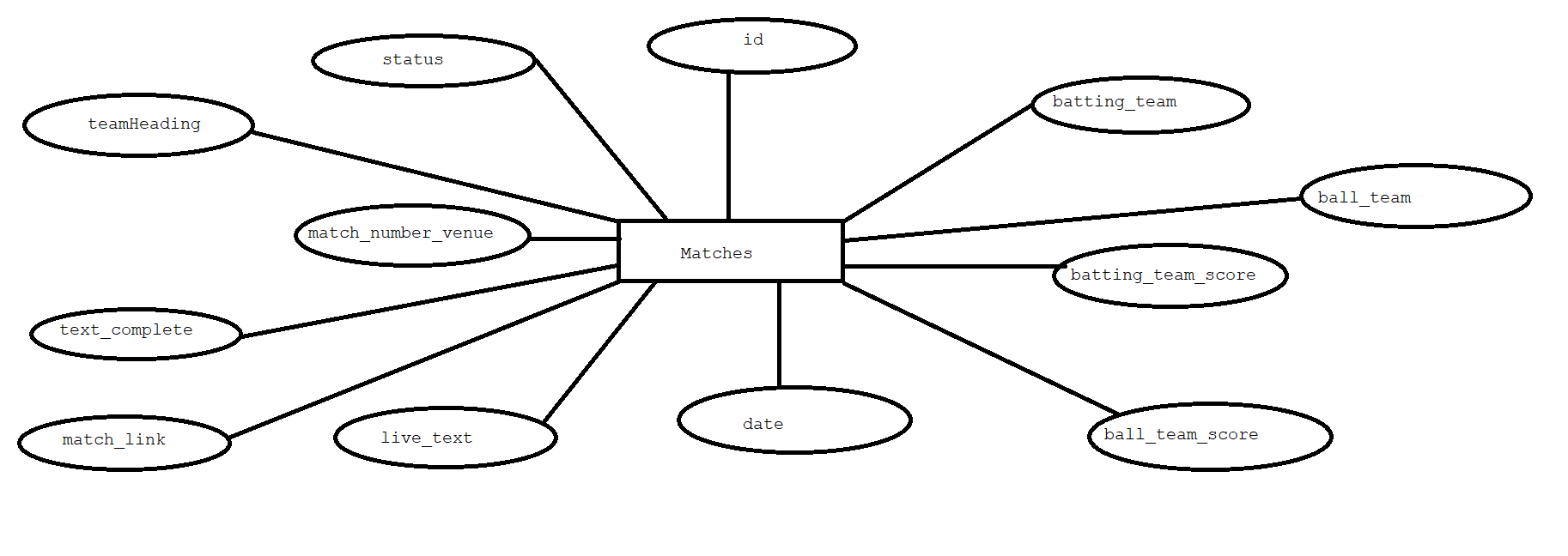
* + The system shall prioritize user experience by ensuring smooth and seamless navigation.
  + It should minimize loading times for pages and data, providing a responsive and engaging user interface.

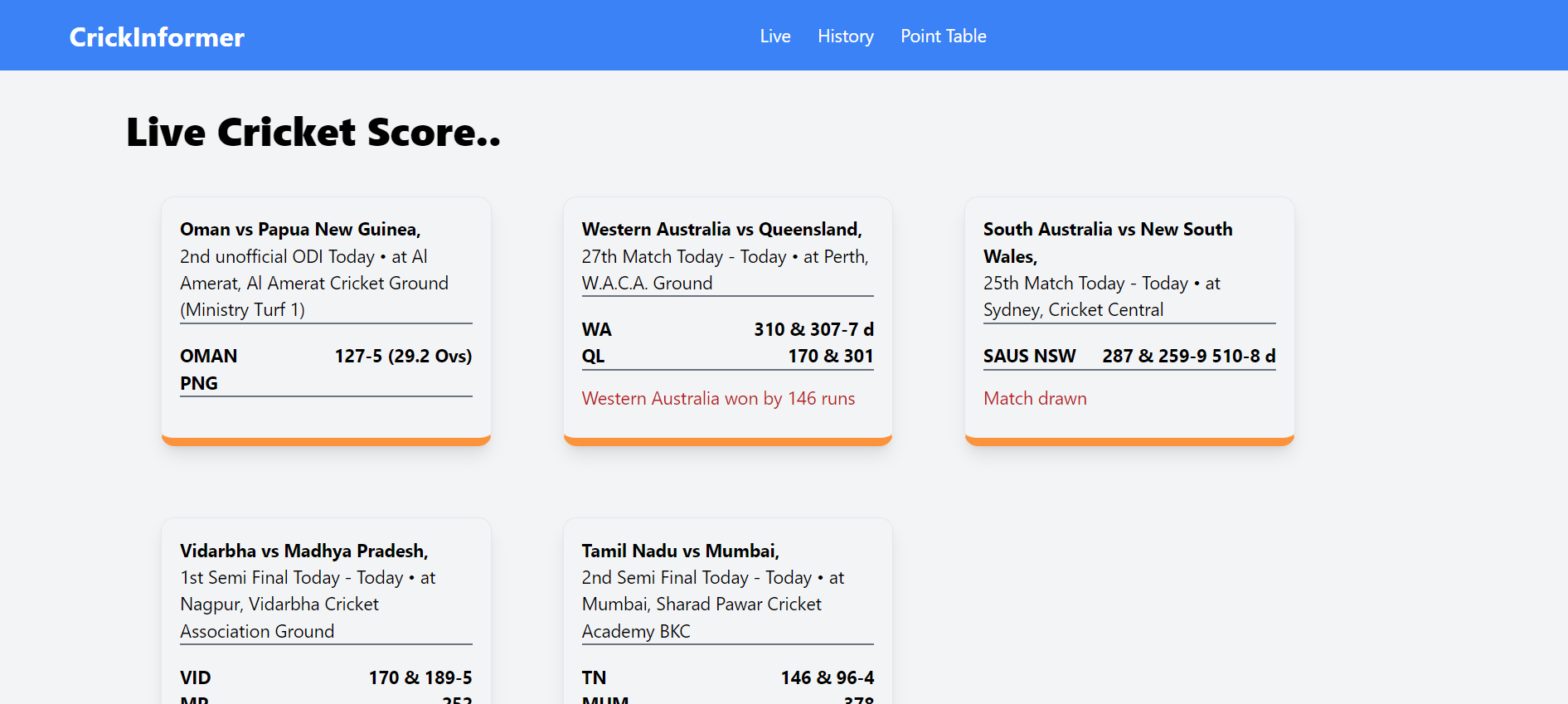
**3. SYSTEM DESIGN**

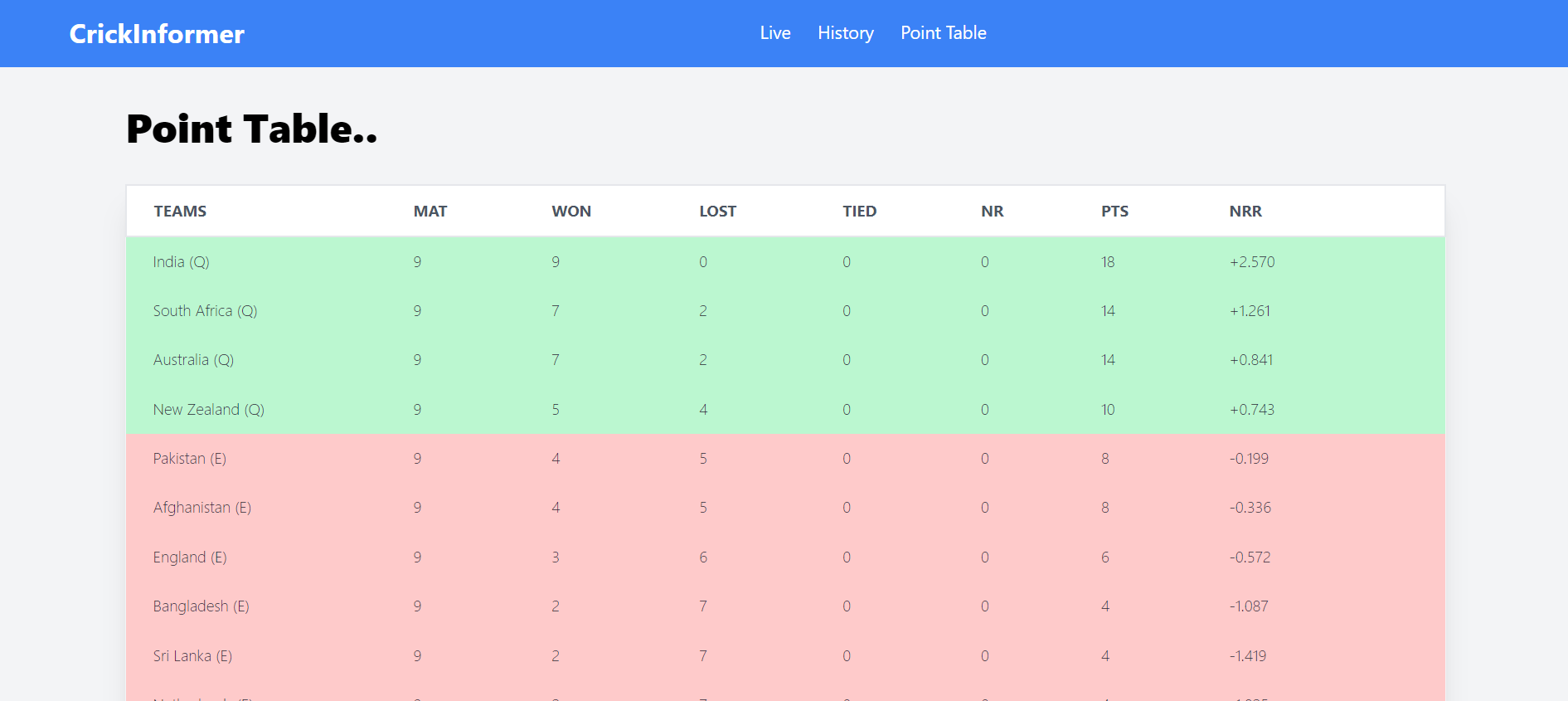
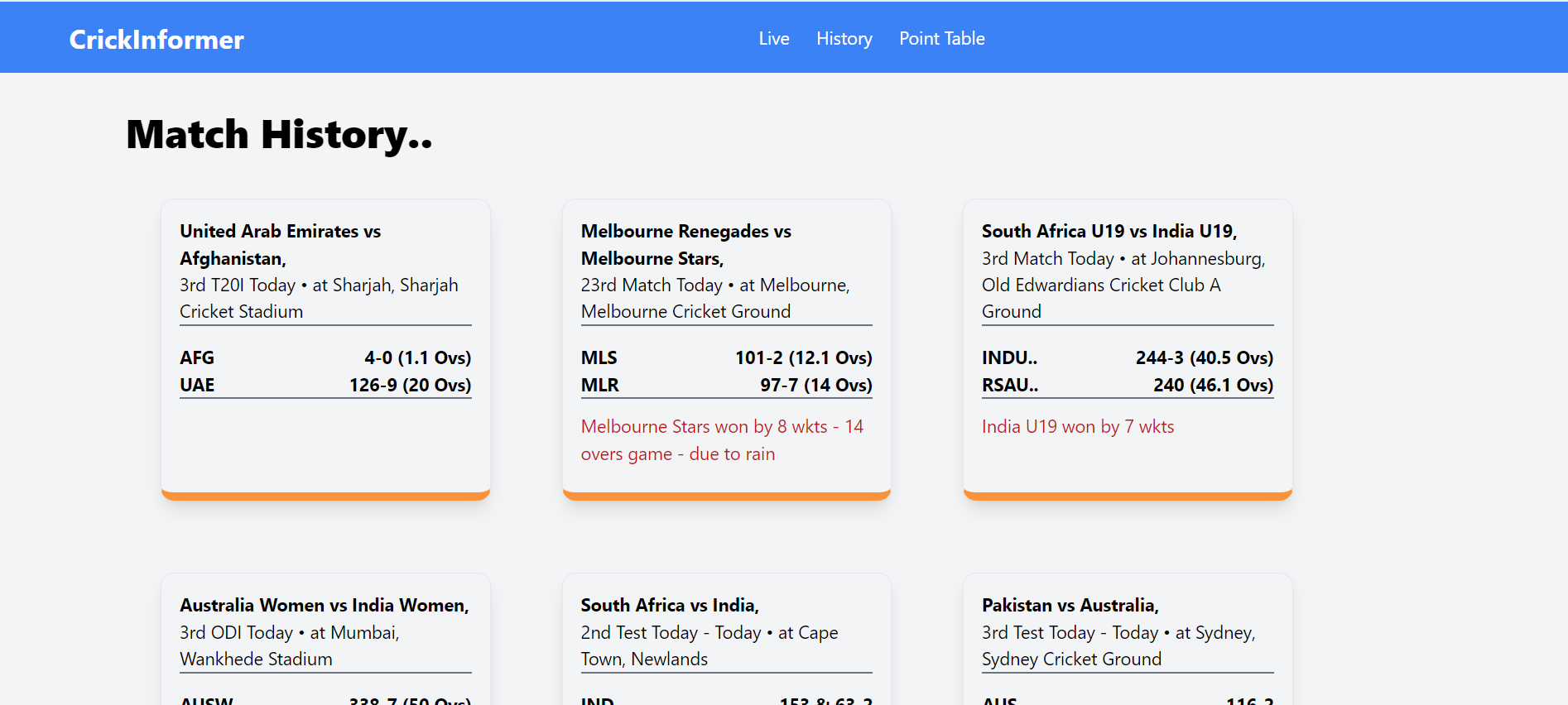
**3.1 Design Constraints:**

1. **Performance Constraints:** Handle high traffic and provide timely updates
2. **Compatibility Constraints:** The system should be compatible with a wide range of devices and browsers to ensure accessibility for all users.
3. **Scalability Constraints**: Easily scale to accommodate increased user demand.
4. **Reliability Constraints**: Minimize downtime and ensure high availability  
     
   **3.2 -System Model:**



**3.3 Data Model**  






**4.Implementation Details**  
  
  
**Software Specifications:**

1. **Frontend Development:**
   1. Angular: Version 17.0.8
   2. TypeScript: Version 4.0
   3. HTML5, CSS3 for frontend design
2. **Backend Development:**
   1. Spring Boot: Version 3.2.3
   2. Java: JDK 17
   3. RESTful API development
3. **Database:**
   1. MySQL: Version 8.0
4. **Development Tools:**
   1. Integrated Development Environment (IDE): IntelliJ IDEA, Visual Studio Code.
   2. Package Management: npm (Node Package Manager)

**Hardware Specifications:**

1. **Development Machine:**
   1. Processor: Intel Core i3
   2. RAM: 525MB
   3. Storage: SSD

**2.Networking:**

* 1. Internet connectivity required for real-time updates.
  2. Firewall for security and scalability.

**5.CONCLUSION & Recommendations:**

The "Crick Informer" project is a comprehensive web application designed to provide cricket enthusiasts with live match updates and historical match information. Leveraging Spring Boot for the backend and Angular for the frontend, the project aims to deliver a dynamic and user-friendly experience for cricket fans. Through the implementation of real-time updates, a well-structured database for historical match information, and a user-friendly frontend interface, the "Crick Informer" project has the potential to become a valuable resource for cricket enthusiasts worldwide.  
  
**Recommendations:**

1. **Enhanced User Engagement:** Consider adding features such as user comments, match predictions, and social media integration to enhance user engagement and interaction with the platform.
2. **Performance Optimization:** Continuously monitor and optimize the performance of the application to ensure fast and reliable access to live match updates and historical match information.

Future Scope

1. **Match Prediction:** Adding a match prediction feature where users can predict match outcomes and compare their predictions with actual results can enhance user engagement.
2. **Data Analytics:** Incorporating data analytics to provide insights into player performance.
3. **Fantasy Cricket:** Introducing fantasy cricket leagues where users can create their teams and compete with others based on player performances in real matches can increase user engagement.
4. **Localization:** Adding support for multiple languages and regional content can help in expanding the user base to different regions.

8.Biblography & Reference

1. Web Development Books:

- "Web Development with Node and Express: Leveraging the JavaScript Stack" by Ethan Brown

- "Learning React: A Hands-On Guide to Building Web Applications Using React and Redux" by Kirupa Chinnathambi

2. Online Documentation and Tutorials:

- Documentation from official sources such as React, Node.js, and Express.js websites.

- Tutorials and guides from platforms like MDN Web Docs, W3Schools, and freeCodeCamp.

3. Articles and Blog Posts:

- Technical articles on topics like real-time web applications, microservices architecture, and RESTful API design.

- Blog posts from experienced developers sharing insights and best practices in web development.

4. Research Papers or Journals:

- Academic papers related to web technologies, user interface design, and sports-related websites.