# CricStats – Cricket Statistics Analysis System

SCD Lab Semester Project

Muhammad Adil   
F2022065268  
Section (W2)  
  
Submitted to Ma’am: Shifa Naveed

Table of Contents

Project Description 2

Scope 2

Functional Requirements (FR) 3

Non-Functional Requirements (NFR) 4

Actors 4

Use Case Diagram 5

System Features 6

System Architecture 7

User Interface Mockups 8

Assumptions and Dependencies 9

Future Enhancements 10

# Project Description

CricStats is a desktop application designed for analyzing and visualizing cricket player statistics over the past 10 years. Users can compare player performance across formats, generate charts, and export reports.

# Scope

The app covers data management and visualization for international cricket stats across Test, ODI, and T20 formats. It will provide filtering, comparison, offline access, and reporting features.

# Functional Requirements (FR)

FR01 Store and retrieve 10-year player data.

FR02 Search/filter players by name, format, team, performance.

FR03 Compare multiple players side by side.

FR04 Display visual insights using graphs.

FR05 Export reports in PDF and Excel.

FR06 Login/authentication for analysts/admins.

FR07 CRUD operations for admins.

FR08 Highlight top performances.

FR09 Support format-specific analysis.

FR10 Work offline post installation.

# Non-Functional Requirements (NFR)

NFR01 Load and compare within 2 seconds.

NFR02 99.9% availability.

NFR03 Stable on large datasets.

NFR04 Role-based access control.

NFR05 Secure data storage.

NFR06 Compliance with data laws.

NFR07 Scheduled/manual backups.

NFR08 Runs on Windows/macOS.

NFR09 API integration support.

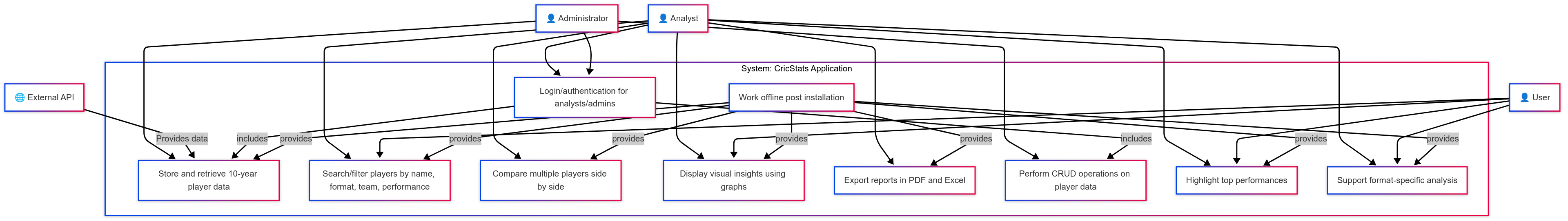
NFR10 Accessible UI design.

# Actors

Analyst: Explores trends, compares players, and exports data.  
Administrator: Manages player data and system settings.  
User: Views stats and visualizations.  
External API: Provides real-time cricket data.

# Use Case Diagram

(To be added manually)



# System Features

1. Store and manage player stats from last 10 years

2. Advanced search and filtering system

3. Multi-player comparison dashboard

4. Interactive data visualizations

5. Data export options

6. User login and access control

7. Top performer highlights

8. Offline data access

9. Admin panel for stat management

10. Integration with external APIs

# System Architecture

Layered architecture:  
1. Presentation Layer (UI)  
2. Logic Layer (analysis & comparison)  
3. Data Layer (local storage/API sources)

# User Interface Mockups

UI mockups include:  
- Dashboard  
- Player search  
- Comparison view  
- Graph export screen  
(To be attached manually)

# Assumptions and Dependencies

- Desktop OS required  
- Python or Electron framework used  
- Matplotlib/Chart.js for graphs  
- SQLite/JSON for storage  
- Initial dataset bundled

# Future Enhancements

- Real-time match integration  
- AI predictions  
- Mobile version  
- Shareable charts  
- Heatmap visualizations