CRICSCORE – LIVE CRICKET SCORE CARD

A Project

Presented to the

Faculty of

California State Polytechnic University, Pomona

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

In

Computer Science

By

Jayavardhan S Patil

2021
SIGNATURE PAGE

PROJECT: CRICSCORE – LIVE CRICKET SCORE CARD

AUTHOR: Jayavardhan S Patil

DATE SUBMITTED: Spring 2021

Department of Computer Science

Yu Sun, Ph. D
Project committee chair
Professor of Computer Science

Gilbert Young, Ph. D
Professor of Computer Science
ABSTRACT

Cricket is the second most popular game in the world. There are many countries in the world who organizes the tournament or matches between the schools or between the organizations, And. Currently there are no applications which lets you host these games online. CricScore mobile application lets you host the match online and update the score in real-time, and people around the world can follow the scores. People wants to recall their old game performances and want to have a glance at individual match details. CricScore app will track each individual game score. One can track the complete details about the game, like individual player performance, who played well and who took more wickets, and result of the match.
**TABLE OF CONTENT**

SIGNATURE PAGE ........................................................................................................ ii

ABSTRACT ....................................................................................................................... iii

LIST OF TABLES .............................................................................................................. v

LIST OF FIGURES ......................................................................................................... vi

CHAPTER 1: INTRODUCTION ....................................................................................... 1

CHAPTER 2: PROBLEM STATEMENT .......................................................................... 2

CHAPTER 3: FUNCTIONALITY ..................................................................................... 3

CHAPTER 4: SYSTEM ..................................................................................................... 4

CHAPTER 5: APPLICATION WALK THROUGH ......................................................... 5

CHAPTER 6: REQUIREMENTS ...................................................................................... 7

CHAPTER 7: TECHNOLOGY .......................................................................................... 9

CHAPTER 8: MOBILE USER INTERFACE .................................................................. 12

REFERENCES ................................................................................................................. 25
LIST OF TABLES

Table 1: Functional Requirements ...................................................................................... 7
LIST OF FIGURES

Figure 1: Manual Score card (source: Wikipedia) [1] ..................................................... 2
Figure 2: CricScore. system design ............................................................................. 4
Figure 3: JMeter Performance test report summary .................................................... 10
Figure 4: JMeter Performance Latency vs Request per sec ........................................ 10
Figure 5: Git Action CI-CD Pipeline .......................................................................... 11
Figure 6: Git action pipeline ..................................................................................... 11
Figure 7: Sign in page ............................................................................................... 12
Figure 8: Profile page ............................................................................................... 13
Figure 9: Home page ................................................................................................. 14
Figure 10: Select teams .............................................................................................. 15
Figure 11: Select team players .................................................................................. 16
Figure 12: Start match ............................................................................................... 17
Figure 13: Toss page ................................................................................................. 18
Figure 14: Select openers ........................................................................................ 19
Figure 15: Update score ............................................................................................ 20
Figure 16: Innings over popup box ......................................................................... 21
Figure 17: Match result ............................................................................................ 22
Figure 18: Game score card ..................................................................................... 23
Figure 19: Menu option ............................................................................................ 24
CHAPTER 1: INTRODUCTION

Everyone plays cricket almost on every weekend. And there are no applications exist to keep track of these match details, When I was a kid, I always wanted to recall my old game performances. This interests me to build an application which lets you host matches and keep track of individual game performances. Today when I meet my old friends, we always talk about the matches we had played during school time and recall all that memory. All this problem statement and to build healthy social relationship with friends I’m building this CricScore app where one can keep track of all history games. And recall in future,

The objective of this project is to build the mobile application using new technologies, which would be easy and user-friendly. Mobile front-end application is built on the natively compiled flutter platform for android, and for the backend I have used latest technologies like Spring boot for Rest APIs, and it is hosted on the Google cloud in Kubernetes cluster. For scalability and zero down time releases. I have used google authenticator for the login and creating player account. I have used Redis for the caching and PostgreSQL for the persistent storage. Used Spring data JPA along with hibernate for ORM (object relational mapping). The back-end system is compiled and deployed using Git Hub Actions CI-CD pipeline, and built image is stored in Docker Hub. Apache JMeter and Grafana is used for REST API performance analysis.
CHAPTER 2: PROBLEM STATEMENT

2.1 MANUAL SCORE CARD

Today all the games hosted by colleges, schools and local organizations are recorded manually in score card chart book as in Figure 1. This is tedious job to update manually and its error prone, addition to this it is difficult to understand the chart and there is no way audience can know the updated live game score.

![Figure 1: Manual Score card (source: Wikipedia)](source: Wikipedia) [1]

This chart looks so messy no one would want to refer this in future. In today’s world everything is online, everyone wants to track the match scores live on phone. Like international game score-card application. CricScore mobile app address this problem and tries to solve by using latest technologies. this application is connected over network and serves live score update and keep all history match records.
CHAPTER 3: FUNCTIONALITY

The core functionality of the CricScore application is to convert above manual chart into Online Mobile application.

3.1 SCOPE

1) Create Account – Sign in / out using Google account.
2) Create Player profile – Profile page for update
3) Create Team
4) Start New Match
5) Select players from City for match
6) Select match venue and configure match meta data
7) Toss and option to select inning type
8) Select open Batting and Bowling players
9) Live update score page – for live score card
10) Match summary list on home page with automated score update – stream functionality
11) Match score card details along with player statistics
CHAPTER 4: SYSTEM

4.1 SYSTEM DESIGN

The back-end REST APIs built using Spring boot framework in java [2]. Used PostgreSQL for persistence storage and Redis for the caching. Enabled hibernate second level caching to avoid DB hits. CI-CD pipeline is built using GitHub Actions and image is stored in Docker Hub. The back-end system is hosted on google cloud platform in Kubernetes [3] with 2 clusters and with minimum 2 pods.

Figure 2: CricScore. system design

Used google Authenticator API [4] for user authentication using firebase Authentication API. [5].
CHAPTER 5: APPLICATION WALK THROUGH

5.1 CREATE PROFILE

Users are asked to sign in using google account as in figure 7. authenticated using google account. Used Flutter provider API and Firebase to check the login status. User would be asked to update profile such as City, Mobile number and Date of Birth as in figure 8, City field is mandatory to show the list of matches in his city. Json Payload to update player profile.

5.2 MATCH SUMMARY - LIVE SCORE CARD

Home page will display list of match summary happened in the user’s city as shown in figure 9, each match summary shows the details like who is current batsman bowlers and what is current score, target and if match is over then it has match result as well, live match data is served from Firebase API, historical data is served from service backend service http://{hostIP}:5050/matches/cities/{cityId}. Tapping on the match summary card would take to the match score card page as in figure 18, This page has all the details about match and every players statistic like runs scored, and balls faced, along with number of fours and sixes and the stick rate of the batsman, and for bowler’s statistics how many overs bowled, runs given and wickets taken and overs economy details of both innings.

5.3 START MATCH

To host match, user would need to start match and select two teams, used trie data structure for quick autosuggestion based on team name prefix. users have flexibility to select players from already for this team or players from his/her city as in figure 11. Once two teams selected It will ask for the match venue and total overs for this match as in figure
12. Later toss page and innings type as in figure 13. once match starts should select the open batsman and bowlers as in figure.

5.4 UPDATE SCORE

Update score page as in figure 15, shows complete details current score, target, current Stricker and bowler, along with their stats. It has score buttons 0 – 1 – 2 – 3 – 4 – 6, addition to these there are Wide and NB to account for extra run. Each bowl has different color for each event, if any batsman gets out it will pop-up dialog window to select next batsman. And if over is finished then it would ask for the next bowler. The current Stricker is highlighted with orange color and it get updated based on the runs scored and over finished this app would render every time there is an event everything set in setState() to update the current score. Back-end this data is updated both in firebase DB, Redis and PostgreSQL DB. Once first innings is over it would show up the dialogue window with target score and would ask to start second innings. Once match is over congratulation window will pop up with match result as in figure 17.
CHAPTER 6: REQUIREMENTS

6.1 FUNCTIONAL REQUIREMENTS

Table 1: Functional Requirements

<table>
<thead>
<tr>
<th>Sr No</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FR1</td>
<td>User needs create account to be listed as a player</td>
</tr>
<tr>
<td>FR2</td>
<td>User needs to authenticate using Google account</td>
</tr>
<tr>
<td>FR3</td>
<td>Update profile details</td>
</tr>
<tr>
<td>FR4</td>
<td>Should be able to view live match summary, without refreshing the screen</td>
</tr>
<tr>
<td>FR5</td>
<td>Should be able to create and select Team</td>
</tr>
<tr>
<td>FR6</td>
<td>Users should be able to select players from current City</td>
</tr>
<tr>
<td>FR7</td>
<td>Configure match details like Over and Match Venue</td>
</tr>
<tr>
<td>FR8</td>
<td>Based on Toss won, team should be given an option to select Innings type</td>
</tr>
<tr>
<td>FR9</td>
<td>Select Opener Players</td>
</tr>
<tr>
<td>FR10</td>
<td>Update score card, 1,2,3,4,6, Out, Wd buttons should be available for the score update</td>
</tr>
<tr>
<td>FR11</td>
<td>Select new Player on wicket</td>
</tr>
<tr>
<td>FR12</td>
<td>Striker should be changed and highlighted automatically</td>
</tr>
<tr>
<td>FR13</td>
<td>On Over complete select new bowler</td>
</tr>
<tr>
<td>FR14</td>
<td>On innings over, pop up window should appear with first innings summary and the target score</td>
</tr>
<tr>
<td>FR15</td>
<td>Should be able to start the second innings and select openers</td>
</tr>
<tr>
<td>FR16</td>
<td>On Match complete Match result should be displayed and ask to finish match</td>
</tr>
<tr>
<td>FR17</td>
<td>Game Player statistics should be available on match score card page</td>
</tr>
<tr>
<td>FR17</td>
<td>Game Player statistics should be available on match score card page</td>
</tr>
</tbody>
</table>

6.2 TECHNICAL REQUIREMENTS

System should be always available, with low latency, since back end system is deployed in Kubernetes it should auto scale on increased traffic. Cache the static data for the low latency. CI – CD pipeline for the Continuous integration and delivery, deployment should happen only after pipeline is successful, Cover unit test cases.
CHAPTER 7: TECHNOLOGY

7.1 MOBILE APPLICATION

Mobile application developed using Flutter SDK. Application is available for Android devices and will be hosted on Play store. Using Google auth API for authentication, no sensitive information is collected from this application. Code can be found in the below link. jayavardhanpatil/Cricscore_mobile (github.com)

7.2 BACK-END REST API

Spring boot framework for the REST APIs using java as programming language. Used Maven build tool for building the jar. Used Spotify plugin to build and push image to Docker Hub. Code can be found in the below link. jayavardhanpatil/cricscore-backend: Masters project, Live cricket scorecard application (github.com)

7.3 DATA BASE

PostgreSQL used to store all the data. This is hosted on the Kubernetes cluster for auto scale. Used Spring data JPA for entity mapping. In Google cloud service.

7.4 TESTING

7.4.1 Unit test Case:

Followed TDD approach to develop backend system, have written unit testcases using Junit framework. Used Mockito framework to mock the DB services and data.

7.4.2 Performance Testing:

Used JMeter for REST API performance testing. Performance testing done with 50 concurrent users. Test reports are shown in Figure 3 & 4.
### 7.4 CI-CD

CI – CD pipeline is configured in GitHub actions. On commit git webhook would trigger the pipeline, there are 3 jobs in the pipeline as shown in figure 5, Build, Test case and Deploy in Docker Hub. CI-CD status shown in Figure 6. Image can be found in the below link [Docker Hub](#).
7.5 DEPLOYMENT

Once CI – CD pipeline success image is built and pushed to Docker Hub. We can deployment.yaml in Kubernetes. Configured 40 sec delay time in Kubernetes so there won’t be down time for release. We can release the new version by running below Kubernetes command: “kubectl set image deployment/cricscore-springboot cricscore-springboot=jayavardhanpatil/cricscore-springboot:v7”
CHAPTER 8: MOBILE USER INTERFACE

8.1 SIGN IN SCREEN

Figure 7: Sign in page
8.2 PROFILE PAGE

![Profile Page](image)

*Figure 8: Profile page*
8.3 HOME PAGE

Figure 9: Home page
8.4 SELECT TEAMS

Figure 10: Select teams
8.5 TEAM PLAYERS

![Select team players](image)

*Figure 11: Select team players*
8.6 SELECT MATCH VENUE

![Start match](image)

*Figure 12: Start match*
8.7 TOSS SCREEN

Figure 13: Toss page
8.8 SELECT INNINGS OPENERS

Figure 14: Select openers
8.9 UPDATE SCORE

Figure 15: Update score
8.10 INNINGS OVER DIALOGUE BOX

Figure 16: Innings over popup box
8.11 MATCH OVER DIALOGUE BOX

Figure 17: Match result
8.12 MATCH SCORE CARD

**Figure 18: Game score card**

<table>
<thead>
<tr>
<th>BATTLING</th>
<th>R</th>
<th>B</th>
<th>4's</th>
<th>6's</th>
<th>SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Willie</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>600.00</td>
</tr>
<tr>
<td>Jacques</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Tristan</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bordy</td>
<td>13</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>325.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BOWLING</th>
<th>O</th>
<th>R</th>
<th>W</th>
<th>EC</th>
<th>Extr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jerald</td>
<td>0.50</td>
<td>19</td>
<td>0</td>
<td>0</td>
<td>38.00</td>
</tr>
</tbody>
</table>
8.13 LOGOUT AND START MATCH MENU

Figure 19: Menu option
REFERENCES


