**CRICKET FEVER**

**(CRICKET MANAGEMENT SYSTEM)**

**SOFTWARE ENGINEERING PROJECT REPORT**

B.Sc (H) Computer Science



SUBMITTED BY:-

|  |  |  |  |
| --- | --- | --- | --- |
| Riya Goel | Vrinda Anand | Priya Aggarwal | Sonakshi Garg |
| 16035570071 | 16035570094 | 16035570060 | 16035570080 |

Department of Computer Science

Keshav Mahavidyalaya

University of Delhi

**ACKNOWLEDGEMENT**

On the successful completion of our project CRICKET FEVER, we would like to express our sincere gratitude to everyone who helped us in the completion of the project.

We are sincerely thankful to our principal ma’am **Dr. Madhu Pruthi** and our software engineering teachers **Mr. Arun Agarwal, Dr. Sumit Aggarwal, Mr. Ravi Kumar** and **Ms. Vinita Jindal** for their interest, guidance and suggestions throughout the course of the project. We feel honored and privileged to work under them. They shared their vast pool of knowledge with us that helped us steer through all the difficulties with ease. This project would not have been possible without their guidance.

**CERTIFICATE**

This is to certify that the software engineering project report entitled “Cricket Fever (Cricket Management System) is the work carried out by Riya Goel, Vrinda Anand, Priya Aggarwal and Sonakshi Garg, students of B.Sc(H) Computer Science, IV semester, Keshav Mahavidyalaya, University of Delhi under the supervision of Mr. Arun Agarwal.

This report has not been submitted to any other organisation/institution for the award of any other degree/diploma.

SUPERVISOR TEACHER IN CHARGE

MR. ARUN AGARWAL MRS. VINITA JINDAL

PRINCIPAL

DR. MADHU PRUTHI

**TABLE OF CONTENT**

Table of Contents

1. SOFTWARE REQUIREMENT SPECIFICATION……….................................1

1.1 INTRODUCTION: .............................................................................................1

1.2 FEATURES…….................................................................................................1

1.3 PURPOSE: ..........................................................................................................1

1.4 SCOPE:................................................................................................................2

1.5 OVERALL DESCRIPTION: .............................................................................2

1.6 PRODUCT PERSPECTIVE: .............................................................................2

1.7 CONSTRAINTS, ASSUMPTIONS AND DEPENDENCIES:..........................3

1.8 SPECIFIC REQUIREMENTS: ..........................................................................3

1.8.1 INTERFACE REQUIREMENTS……..................................................3

1.8.2 FUNCTIONAL REQUIREMENTS……………………………..........3

1.8.3 NON-FUNCTIONAL REQUIREMENTS………………………........4

1.8.4 DESIGN CONSTRAINTS…………………………………………….4

1.9 DATA FLOW DIAGRAMS: ............................................................................ 5

1.9.1 DFD LEVEL 0………………………………………………………...5

1.9.1 DFD LEVEL 1………………………………………………………...6

1.9.1 DFD LEVEL 2………………………………………………………...7

1.10 DATA DICTIONARY......................................................................................8

2. SIZE ESTIMATION AND SCHEDULING ....................................................... 9

2.1 FUNCTION PIONTS ESTIMATION ..............................................................10

2.2 SIZE ESTIMATION ........................................................................................11

2.3 COST ESTIMATION .......................................................................................11

2.4 SCHEDULING .................................................................................................12

3. ARCHITECTURAL DESIGN............................................................................13

3.1 VIEW…………………………………………………………………..14

3.2 RETRIEVE……………………………………………………………..14

4. RISK ANALYSIS ..............................................................................................15

4.1 NEW UNPROVEN TECHNOLOGY.............................................................. 15

4.2 ORGANISATIONAL RISK............................................................................. 15

4.3 PROJECT DELAY……………………………………………………………16

4.4 RISK OF STORAGE AND DATABASE SPACE…………………………...16

5. IMPLEMENTATION OF A MODULE............................................................ 19

5.1 PSEUDO CODE............................................................................................... 19

5.2 SCREENSHOT OF THE SOFTWARE .......................................................... 24

6. TESTING ............................................................................................................29

7. USERS MANUAL .............................................................................................34

7.1 LOGIN ..............................................................................................................34

7.2 RETRIEVAL.....................................................................................................35

8. CONCLUSION ..................................................................................................37

9. REFERENCES ...................................................................................................38

**LIST OF FIGURES**

Figure 1: DATA FLOW DIAGRAM LEVEL-0………………………………………………….5

Figure 2: DATA FLOW DIAGRAM LEVEL-1………………………………………………….6

Figure 3: DATA FLOW DIAGRAM LEVEL-2………………………………………………….7

Figure 4: GANTT CHART………………………………………………………………………12

Figure 5: ARCHITECTURAL DESIGN………………………………………………………...13

**LIST OF TABLES**

Table 1- DATA DICTIONARY………………………………………………………………….8

Table 2- PROJECT ESTIMATORS………………………………………………………………9

Table 3 - FUNCTIONAL POINT ANALYSIS……………………………………………….…10

Table 4 – RISK ASSESSMENT…………………………………………………………………16

Table 5 – RMMM1………………………………………………………………………………17

Table 6 – RMMM2………………………………………………………………………………18

1. **Software Requirement Specification**
   1. Introduction

The aim of this project is to provide the complete information of the National and International statistics. The information is available country wise and player wise. By entering the data of each match, we can get all type of reports instantly, which will be useful to call back history of each player. Also the team performance in each match can be obtained. We can get a report on number of matches, wins and lost.

* 1. Features

The project will enable the user to view:

* Every Cricketer’s Information
* Every Cricketer’s Personal Achievements
* International Travel News
* Upcoming Series
* Every Team’s Information
* ODI and test ranking
* Increment of each ball ,over, run is done automatically
* Commentary of each ball

* 1. Purpose

This document aims to give a brief description about the Cricket Management System Project. This project is very use for Cricket match broadcasters to get information quickly. Also for Cricket lovers who are very much interested in Cricket Statistics. In other words this document will provide a basis for validation and verification.

* 1. Scope

The project is designed very user friendly such that even people who know only the basic operation of the computer can use this software. So no technical expertise is required to use this software.

This software is also functional to find out the application of cricket management system and to provide information and manage the system regarding cricket matches and team training. It also helps cricket teams to register new members.

1.5 Overall Description

The overall description of our project can be stated as creating and managing the database, developing a friendly user interface to manipulate the database, provide an authentication mechanism to safely accomplish tasks mentioned above.

1.6 Product Perspective

With the CMS (Cricket Management System) we will provide them with capabilities and properties organized neatly. CMS which is an online intranet System will be used by anyone who wishes to find information for the players and teams.

Admin will basically query and edit the database via CMS. It will also calculate the rank of each player depending on their Strike Rate and no. of matches played.

1.7 Constraints, Assumptions and Dependencies

**Regularity Policies:**

Each user must login first to have access to information of players. In other words, each user has an account created by the Admin.

**Hardware Limitations:**

There is no limitation in the operating system in which CMS will work. However, the CMS system and the database will work on a server that needs to be always online.

1.8 Specific Requirements

1.8.1 Interface Requirements

All the users will see the same page when they enter CMS(Cricket Management System).There will be 2 options: One for the user and one for the admin. The admin is required to login through an ID and a password so as to make changes in the database or to add new information to it. On the other hand users can retrieve the information about the different players, along with their personal details and team details as well after getting logged in to their respective accounts.

1.8.2 Functional Requirements

Requirements which are related to functional aspects of software fall into this category. According to the CMS we have:

* View option is given to the user to view the details of their favorite players along with their personal details.
* Retrieve function lets the user to offline download the details of the players and the teams.

1.8.3 Non-Functional Requirements

Requirements which are not related to functional aspects of software fall into this category. They are implicit or expected characteristics of the software, which users make assumption of. The Non-Functional Requirements of CMS are:

* **Logging**- Both the Admin and the User need to log in to access data from CMS.
* **Storage**- The information regarding the players and their teams is stored on a database. The information includes the players’ personal details along with the achievements in their cricketing career.
* **Performance**- The CMS satisfies the user requirements and can handle multiple users at a time, resulting in better performance.
* **Cost**- The overall cost of the project is minimal which is easily affordable by one and all.
* **Flexibility**- The CMS is highly flexible as it can be accessed by any person or the organization without any difficulty.
  + 1. Design Constraints
* **Hardware Requirements**

The CMS application will be storing 500 players’ personal data which will roughly require 500mb of database storage space.

* **Software Requirements**

1. Since the CMS is an android based application so the user will require mobile handsets for the deployment of the software.
2. To store and fetch data from the database online i.e. cricdb, the user also requires an active internet connection for smooth working of the project.
3. The CMS will work properly on any mobile set with android version 4.4 or higher.
4. Application will run on 256mb or higher of RAM.
5. It requires a space of 2.5mb to deploy onto the machine.

1.9 Data Flow Diagram

Data flow diagrams (also called data flow graphs) are commonly used during problem analysis A DFD shows the flow of data through a system. It views a system as a function that transforms the inputs into desired outputs. The agent that performs the transformation of data from one state to another is called a process (or a bubble). Thus, a DFD shows the movement of data through the different transformations or processes in the system.

1.9.1 DFD Level 0

ADD

VIEW

**USER**

**ADMIN(ICC)**

STATISTICS

LOGIN/ REGISTER UPDATE

Figure 1: Data Flow Diagram Level 0

1.9.2DFD Level 1

PLAYER INFO

VIEW

PLAYER

PLAYER RECORD

VIEW

TEAM

TEAM INFO

TEAM

**USER**

RECORD

RECORD

**ADMIN**

VIEW STATS

PLAYER STATISTICS

STATS

RETRIEVE STATS

ACHIEVEMENTS

TEAM ACHIEVEMENTS

RETRIEVE

ACHIEVE

PLAYER PERSONAL

DETAILS

RETRIEVE

RANK

PLAYER RANKS

Figure 2: Data Flow Diagram Level 1

1.9.3 DFD Level 2

PLAYER RECORD

PLAYER INFO

RECORD

NAMES OF BATSMEN

NAMES OF BOWLERS

**USER**

TEAM INFO

TEAM

RECORD

STATS

**ADMIN**

Player ID

PLAYER STATISTICS

PLAYER NAME

Team ID

TEAM

ACHIEVEMENTS

ACHIEVEMENTS

TEAM ID

PLAYER NAME

PLAYER ID

PLAYER ID

PLAYER PERSONAL

DETAILS

PLAYER RANKS

Figure 3: Data Flow Diagram Level 2

* 1. Data Dictionary

|  |  |
| --- | --- |
| **DATA** | **DESCRIPTION** |
| Record | Pid + Pname + DOB + Team + Bat\_Style + Bowl\_Style + Strike\_Rate + No\_Of\_Centuries + Tid + Tname + Captain + No\_Of\_Batsmen + No\_Of\_Bowlers + Team\_Colour + Coach\_Name +… |
| Player Record | Pid+ Pname+ DOB+ Team+ Bat\_Style + Bowl\_Style+ Strike\_Rate+ No\_Of\_Centuries +… |
| Team Record | Tid + Tname+ Captain + No\_Of\_Batsmen+ No\_Of\_Bowlers + Team\_Colour + Coach\_Name+… |
| Stats | Man\_Match + Man\_Series + Highest\_Centuries |
| Achievements | Won\_matches + Lost\_matches + Tie\_matches |
| Player Personal Details | Pid + Pname + DOB |

Table 1: Data Dictionary

1. **Size Estimation And Scheduling**

|  |  |  |
| --- | --- | --- |
|  |  | **GRADE**  **VALUE** |
| **1.** | Does the system require reliable backup and recovery? | 5 |
| **2.** | Are specialized data communications required to transfer  Information to or from the application? | 3 |
| **3.** | Are there distributed processing functions? | 4 |
| **4.** | Is performance critical? | 0 |
| **5.** | Will the system run in an existing, heavily utilized  Operational environment? | 5 |
| **6.** | Does the system require on-line data entry? | 5 |
| **7.** | Does the on-line data entry require the input transaction to be  built over multiple screens or operations? | 3 |
| **8.** | Are the Master files updated online? | 5 |
| **9.** | Are the inputs, outputs, files, or inquiries complex? | 2 |
| **10.** | Is the internal processing complex? | 1 |
| **11.** | Is the code designed to be reusable? | 1 |
| **12.** | Are conversions and installations included in the design? | 0 |
| **13.** | Is the system designed for multiple installations in different  Organizations? | 5 |
| **14.** | Is the application design to facilitate change and for ease of use by the user? | 5 |

Table 2: Project Estimators

* 1. Function Points Estimation

Considering the following assumptions:

**3-4 INPUTS AS LOW**

**4-7 INPUTS AS AVERAGE**

**>7 INPUTS AS HIGH**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | COUNT | | | WEIGHING FACTOR | | | TOTAL COUNT |
| Simple | average | Complex | Simple | Average | Complex |  |
| Number of  internal inputs | 2 | 2 | 1 | 3 | 4 | 6 | 20 |
| Number of  external outputs | 3 | 2 | 1 | 4 | 5 | 7 | 29 |
| Number of  inquiries | 2 | 2 | 2 | 3 | 4 | 6 | 26 |
| Number of  internal logical  files | 0 | 0 | 3 | 7 | 10 | 15 | 45 |
| Number of  external  interfaces | 1 | 1 | 1 | 5 | 7 | 10 | 22 |
|  |  |  | **Total FPs** |  |  |  | 142 |

Table 3: Function Point Analysis

Thus, the UNADJUSTED FUNCTIONAL POINTS (UFP) ARE:

**UFP** = Total I/P + Total O/P + Total ILF + Total EIF + Total Enquiries

= 20 + 29 + 45 + 22 + 26

= **142**

Now, COMPLEXITY ADJUSTMENT FACTOR (CAF) IS:

**CAF**= 0.65 + (0.01 \*∑ fi)

= 0.65 + (0.01 \*3\*14)

= **1.07**

So the Functional Points (FP) is:

**FP** = UFP\*CAF

= **151.94**

2.2 Size Estimation

Assuming that average productivity for such a system is 6.5 FP/pm i.e. 6.5 functional points per person month, then the estimates are:

Size=Total FP’s / Average productivity

= 151.94/6.5

= 23.37 pm i.e. 24 person month.

* 1. Cost Estimation

Assuming that the cost is Rs.5000/pm, i.e. 5000 per person month.

Cost=cost per person month \* effort

= Rs. (5000\*14) = Rs.70,000

2.4 Scheduling

Scheduling is the culmination of a planning activity that is a primary component of Software project management. When combined with estimation methods and risk analysis, scheduling establishes a road map for the project manager. The characteristics of the project are used to adapt an appropriate task set for the work to be done. A task network depicts each engineering task, its dependency on other tasks, and its projected duration.

GANTT CHART

Figure 4: Gantt Chart

1. **ARCHITECTURAL DESIGN**

DISPLAY ACHIEVEMENTS

DISPLAY TEAM INFO

DISPLAY PLAYER PERSONAL

DISPLAY RANKS

DISPLAY TIED\_MATCHES

DISPLAY LOST\_MATCHES

DISPLAY WON\_MATCHES

ENTER

PERSONAL

DETAILS

GET

STATS

ENTER

PLAYER RECORD

CALCULATE

GET STRIKE RATE

GET

RANKS

GET

TEAM RECORD

VIEW\_ACHIEVE

VIEW\_TEAM

VIEW\_PLAYER

GET\_PLAYER

GET\_TEAM

VIEW

RETRIEVE

**CRICKET FEVER**

Figure 5: Architectural design

The architectural design of the software defines every module of the application in detail in a hierarchical way. The CMS application is mainly divided into two modules:

* 1. VIEW

The VIEW MODULE can view information about the PLAYER RECORD , TEAM RECORD and ACHIEVEMENTS of the Team.

It will display the personal information about the player and the ranks.

Also , It will also display the won, lost and tied matches.

3.2 RETRIEVE

The RETRIEVE MODULE will retrieve details of the player and the team.

Here the user will enter personal details and statistics of the player by the help of which STRIKE RATE will be calculated .

The approach used is a top to bottom approach.

1. **RISK ANALYSIS**

While deploying a project, several risks are always there with different probability of happening and different consequences. Here we are giving such risks for our project.

* 1. New Unproven Technology

The Cricket Management System is developed on Android Studio. One of the biggest risks involved with our project is the advancement in technology. As soon as new tools and better technology will come into play in the near future, the software might become outdated and hence

would not be able to meet the user expectations. Thus, the software will not be able to compete with other better software available in the market and henceforth will ultimately fail.

* 1. Organizational Risk

Another risk involved with the Cricket Management System is that of the organization of player info, team info into various tables in the database. Even if there is the slightest possibility that there exists some ambiguity and inconsistency in the stored information about different

players and teams, this would make it difficult for the users to fetch the information of their favorite players and would display wrong results too. Thus, the Software would lose its essence of correctness at the first place. In the longer run, this situation may lead to the complete

removal of the Software from the user market.

* 1. Project Delay

One of the major risks involved with our project is in the project delay that if the project is not completed within the given deadline, then we have to increase the person hours on our project which will ultimately lead to an increase in the overall cost of the project. Furthermore, if the

quality software is not delivered on time, it would be of no use to the costumers as well. Thus, The whole project will not serve any fruitful purpose to the users.

* 1. Risk of storage of database space

What if the company faces the shortage of database space for storing the application database?

Then the CMS application will not work properly. To resolve this, the management has to redesign the database and some changes to the CMS application which will add cost of the application to the management. The probability of this to the management is less in short term and has a little higher risk in the long term.

|  |  |  |  |
| --- | --- | --- | --- |
| **RISK** | **CATEGORY** | **PROBABILITY**  **(OUT OF 100%)** | **IMPACT**  **(1-4)** |
| 1.Size estimate is  significantly low | Product Size | 40% | 2 |
| 2.Large no of users | Product Size | 30% | 3 |
| 3.Less Reuse | Product Size | 20% | 2 |
| 4.Delivery Deadline Not Met | Business  Impact | 20% | 4 |
| 5.Cost Exceeds | Cost Risk | 50% | 4 |
| 6.Custom Requirement Change | Product Size | 45% | 3 |
| 7.Freezing Cost | Business  Impact | 35% | 2 |
| 8.Lack Of Planning | Technology | 10% | 3 |
| 9.Staff In Experience | Staff Size  And  Experience | 10% | 2 |

Table 4: Risk Assessment

**RISK INFORMATION SHEET (RMMM1)**

|  |  |  |  |
| --- | --- | --- | --- |
| **RISK ID:**  R01 | **DATE :**  10-02-2018 | **PROBABILITY:**  60% | **IMPACT:**  High |

**DESCRIPTION:**

Organizational Risk (Inconsistent data into tables)

Table 6: RMMM Table 1

**REFINEMENT / CONTEXT:**

1. Inconsistency of data will lead to the formation of spurious tuples in the database.
2. Incorrect information will be passed on to the users about the players and teams.

**MITIGATION / MONITORING:**

1. Correct data must be inserted into the database tables by the Admin so that no spurious tuples are formed.
2. Cricket fans and other users can get accurate and precise information about their favorite players and teams along with the live scores.

**MANAGEMENT / CONTINGENCY PLAN / TRIGGER:**

1. Risk Estimation computed to be 5000. Allocate this amount within project contingency cost.
2. Develop, revise and schedule assuming that high capable server is to be adopted: allocate staff accordingly.

**Trigger:** Mitigation steps unproductive as of 20-02-2018

**CURRENT STATUS:**

05/03/2018: Mitigation steps initiated

|  |  |
| --- | --- |
| **Originator:** Vrinda Anand | **Assigned:** Riya Goel |

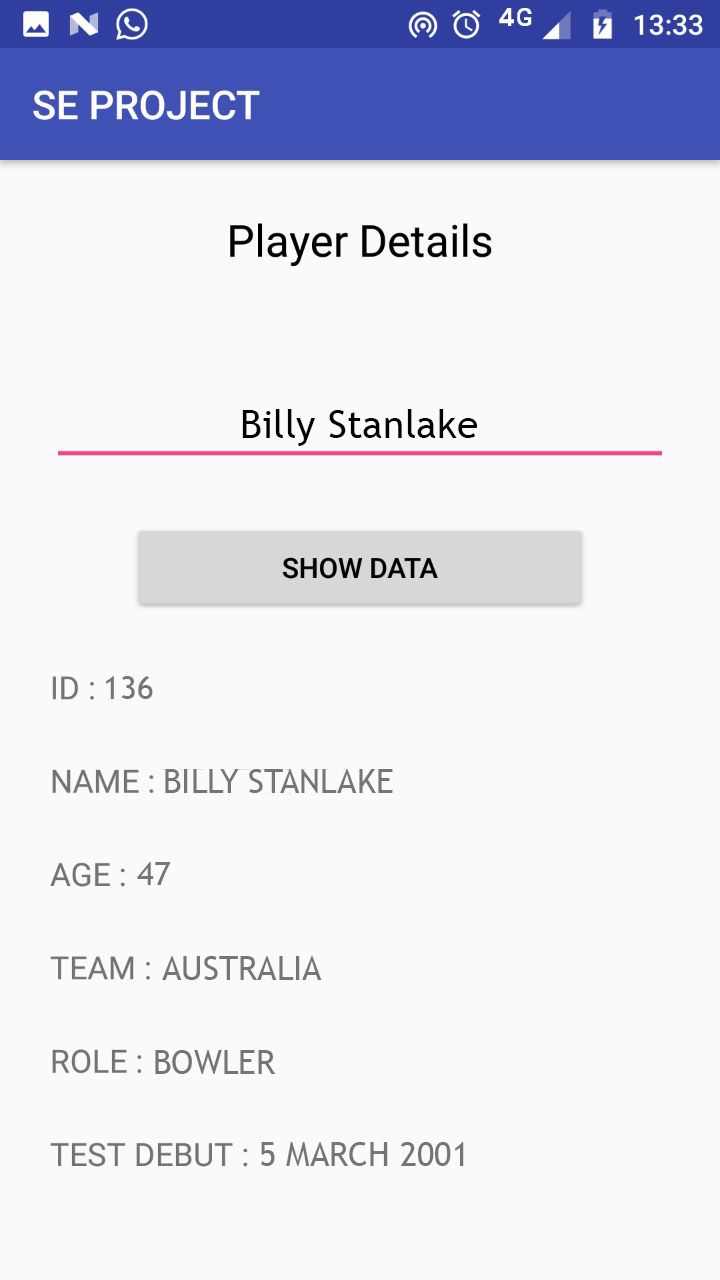
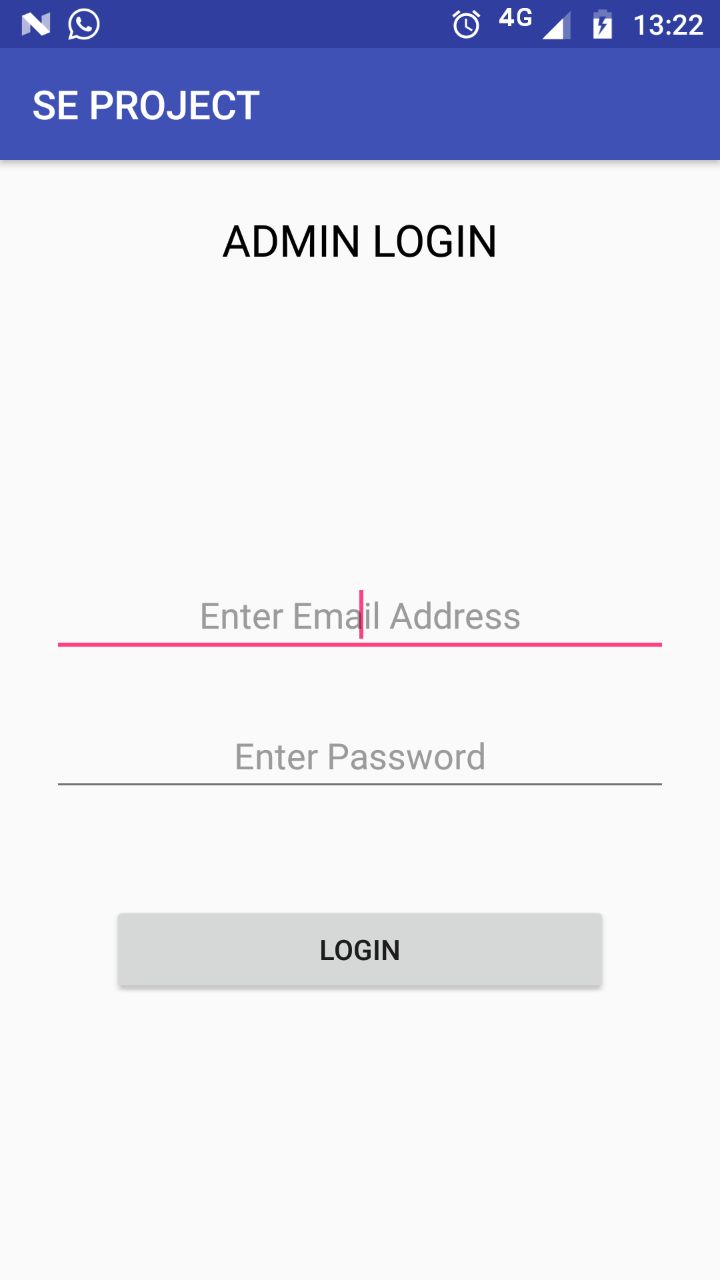
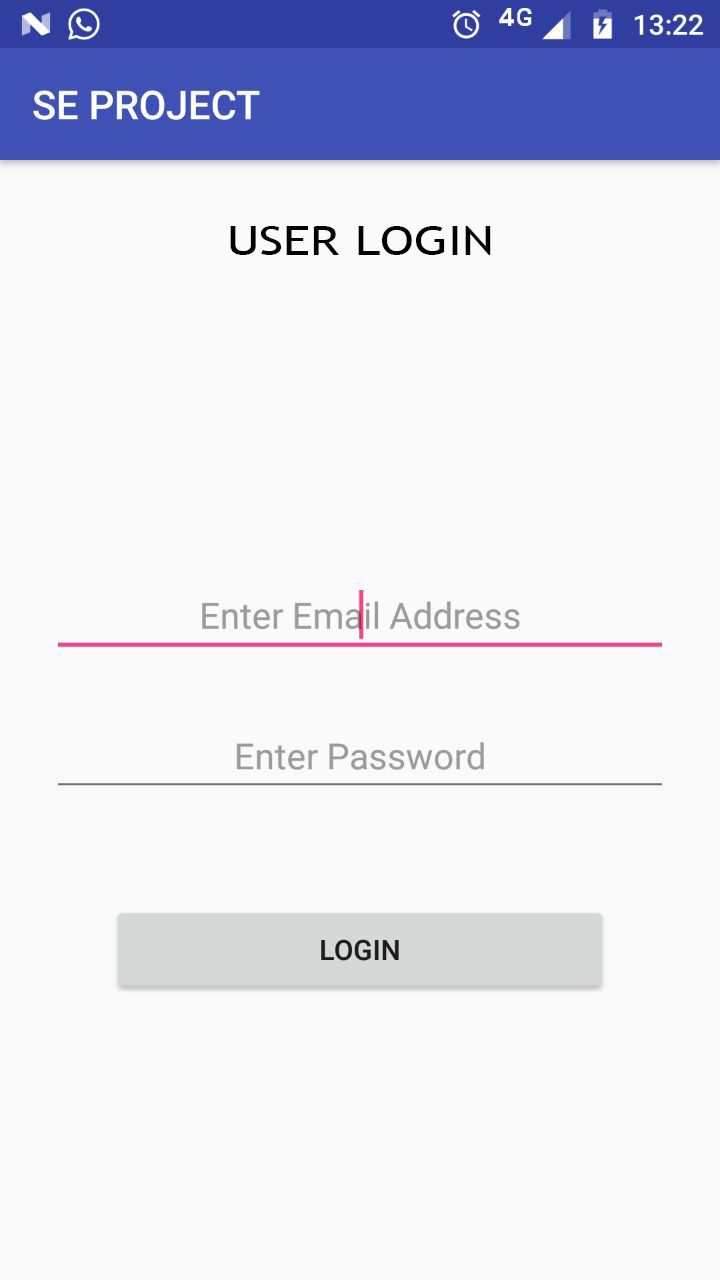
|  |  |  |  |
| --- | --- | --- | --- |
| **RISK INFORMATION SHEET (RMMM2)** | | | |
| **Risk ID:**  R02 | **DATE:**  12-02-2018 | **PROBABILITY:**  40% | **IMPACT:**  AVERAGE |
| **DESCRIPTION:**  New Unproven technology (Obsolete version of Android Studio) | | | |
| **REFINEMENT / CONTEXT:**   1. The software will fail if Android becomes outdated and is replaced by some other Operating System. 2. The software will fail to provide information to the cricket fans about their favourite players. | | | |
| **MITIGATION/MONITORING:**  The software will continue to rule the market if the developers continue to release newer versions of Android. | | | |
| **MANAGEMENT / CONTINGENCY PLAN/ TRIGGER:**   1. Risk Estimation computed to be 40,000. Allocate this amount within project contingency cost. 2. Develop, revised schedule assuming that high capable server is to be adopted: allocate staff accordingly.   **Trigger:** Mitigation steps unproductive as of 22-02-2018 | | | |
| **CURRENT STATUS:**  07/03/2018: Mitigation steps initiated | | | |
| **Originator**: Priya Aggarwal | | **Assigned:** Sonakshi Garg | |

1. **IMPLEMENTATION OF MODULE**

5.1 Pseudo Code

1. #include<iostream.h>
2. #include<conio.h>
3. Using namespace std;
4. void add\_player()
5. {
6. int pid, dob, no\_of\_centuries;
7. char pname, bat\_style, bowl\_style;
8. cout<<”enter player id”;
9. cin>>pid;
10. cout<<”enter player name”;
11. cin>>pname;
12. cout<<”enter dob of player”;
13. cin>>dob;
14. cout<<”enter bat style”;
15. cin>>bat\_style;
16. cout<<”enter bowl tyle”;
17. cin>>bowl\_style;
18. cout<<”enter no. of centuries”;
19. cin>>no\_of\_centuries;
20. }
21. void del\_player(int y)
22. {
23. if(pid==y)
24. cout<<”player details deleted successfully”;
25. }
26. void add\_team()
27. {
28. int tid, bats, bowls;
29. char tname, captain, tcol, cname;
30. cout<<”enter team id”;
31. cin>>tid;
32. cout<<”enter team name”;
33. cin>>tname;
34. cout<<”enter name of captain”;
35. cin>>captain;
36. cout<<”enter no. of batsmen”;
37. cin>>bats;
38. cout<<”enter no. of bowlers”;
39. cin>>bowls;
40. cout<<”enter team color”;
41. cin>>tcol;
42. cout<<”enter coach name”;
43. cin>>cname;
44. }
45. void del\_team(int x)
46. {
47. if(tid==x)
48. cout<<”team details deleted successfully”;
49. }
50. void fetch\_player()
51. {
52. int pid, dob, no\_of\_centuries;
53. char pname, bat\_style, bowl\_style;
54. cout<<” player id is:”;
55. cout<<pid;
56. cout<<”player name is:”;
57. cout<<pname;
58. cout<<” dob of player is:”;
59. cout<<dob;
60. cout<<” bat style is:”;
61. cout<<bat\_style;
62. cout<<”bowl style is:”;
63. cout<<bowl\_style;
64. cout<<”no. of centuries are:”;
65. cout<<no\_of\_centuries;
66. }
67. void fetch\_team()
68. {
69. int tid, bats, bowls;
70. char tname, captain, tcol, cname;
71. cout<<”team id is:”;
72. cout<<tid;
73. cout<<” team name is:”;
74. cout<<tname;
75. cout<<”name of captain is:”;
76. cout<<captain;
77. cout<<”no. of batsmen are:”;
78. cout<<bats;
79. cout<<”no. of bowlers is:”;
80. cout<<bowls;
81. cout<<”team color is:”;
82. cout<<tcol;
83. cout<<”coach name is:”;
84. cout<<cname;
85. }
86. int main()
87. {
88. int choice, ch, ch1;
89. Cout<<”1. LOGIN AS USER”;
90. Cout<<”2. LOGIN AS ADMIN”;
91. Cout<<”enter your choice”;
92. Cin>>choice;
93. switch(choice)
94. {
95. case 1: **cout<<”entering as user”;**
96. cout<<”1. fetch player details”;
97. Cout<<”2. fetch team details”;
98. Cout<<”enter your choice”;
99. Cin>>ch;
100. switch(ch)
101. {
102. case 1: cout<<”\*\*\*\*\*Displaying player details\*\*\*\*\*”;
103. fetch\_player();
104. break;
105. case 2: cout<<”\*\*\*\*\*Displaying team details\*\*\*\*\*”;
106. fetch\_team();
107. break;
108. default : cout<<”wrong choice”;
109. break;
110. }
111. case 2: **cout<<”entering as admin”;**
112. cout<<”1. Add player details”;
113. cout<<”2. Delete player details”;
114. cout<<”3. Add team details”;
115. cout<<”4. Delete team details”;
116. cout<<”enter your choice”;
117. cin>>ch1;
118. switch(ch1)
119. {
120. case 1: cout<<”\*\*\*\*\*Adding player details\*\*\*\*\*”;
121. add\_player();
122. break;
123. case 2: cout<<”\*\*\*\*\*Deleting player details\*\*\*\*\*”;
124. del\_player();
125. break;
126. case 3: cout<<”\*\*\*\*\*Adding team details\*\*\*\*\*”;
127. add\_team();
128. break;
129. case 4: cout<<”\*\*\*\*\*Deleting team details\*\*\*\*\*”;
130. del\_team();
131. break;
132. default : cout<<”wrong choice”;
133. break;
134. }
135. }
136. }

5.2 Screenshot of the Software





1. **TESTING**

SOFTWARE TESTING

Testing is done with an objective of finding most errors with minimum amount of time and effort.

**WHITE BOX testing** sometimes called *glass-box testing,* is a test-case design philosophy that uses the control structure described as part of component-level design to derive test cases. Using white-box testing methods, you can derive test cases that :

(1) Guarantee that all independent paths within a module have been exercised at

least once.

(2) Exercise all logical decisions on their true and false sides.

(3) Execute all loops at their boundaries and within their operational bounds.

(4) Exercise internal data structures to ensure their validity.

***Basic Path Testing*** is a White Box testing technique that enables to derive logical complexity and defines basic test of execution paths. The test cases are prepared so that each execution path will occur at least once.

* **To Design Program Graph**

1. int main(intballs\_played , inttot\_runs)
2. {
3. float strike\_rate ;
4. char rank;
5. strike\_rate=(tot\_runs/balls\_played)\*100;
6. cout<<”the strike rate of the player is:”<<strike\_rate;
7. if((strike\_rate<=300 &&strike\_rate>=200)&&tot\_runs>3000)
8. {
9. rank=’H’;
10. cout<<”the player has the highest rank”<<rank;
11. }
12. else if((strike\_rate<=200 &&strike\_rate>=100)&&tot\_runs>2000)
13. {
14. rank=’A’;
15. cout<<”the player has an average rank”<<rank;
16. }
17. else if((strike\_rate<=100 &&strike\_rate>=50)&&tot\_runs>1000)
18. {
19. rank=’B’;
20. cout<<”the player has a below average rank”<<rank;
21. }
22. Else
23. {
24. rank=’L’;
25. cout<<”the player has the least rank”<<rank;
26. }
27. }

**T**

**F**

**F**

**T**

**F**

**T**

* **Cyclomatic Complexity**

Cyclomatic complexity is a software metric that provides a quantitative measure of the

logical complexity of a program. When used in the context of the basis path testing method,

the value computed for cyclomatic complexity defines the number of independent paths in

the basis set of a program and provides you with an upper bound for the number of tests

that must be conducted to ensure that all statements have been executed at least once.

P = 3

E = 11

N = 9

R=4

WHERE P = NO OF PREDICATE NODES

E = NO. OF EDGES

N = NO. OF NODES

R = NO. OF REGIONS

CYCLOMATIC COMPLEXITY = E-N+2

= 11-9+2

= 4

OR

CYCLOMATIC COMPLEXITY = P+1

= 3+1

= 4

OR

CYCLOMATIC COMPLEXITY = R

=4

* **To Find No. Of Independent Paths**

No. of independent paths=Cyclomatic Complexity

=4

* **To Determine Independent Paths**

1-6,7,8-11,27

1-6,7,12,13-16,27

1-6,7,12,17,18-21,27

1-6,7,12,17,22-26,27

* **Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| Independent Paths | Inputs | Expected O/P | Received O/P |
| 1-6, 7, 8-11, 27 | S.R=250,TOT\_RUNS=3200 | rank=’H’; | rank=’H’; |
| 1-6, 7, 12, 13-16, 27 | S.R=198,TOT\_RUNS=2500 | rank=’A’; | rank=’A’; |
| 1-6, 7, 12, 17, 18-21, 27 | S.R=90,TOT\_RUNS=1014 | rank=’B’; | rank=’B’; |
| 1-6, 7, 12, 17, 22-26, 27 | S.R=20,TOT\_RUNS=500 | rank=’L’; | rank=’L’; |

1. **USER MANUAL**

The Cricket Management System is an android application. Following are the requirement for the proper deployment of the software:

1. An android mobile phone is required.

2. The mobile phones must run on android 4.4 or higher version.

3. It has at least 250 MB of RAM and a 20 Mb of the hard disk space.

4. It must have an active internet connection for the smooth functioning

of the software.

* 1. LOGIN

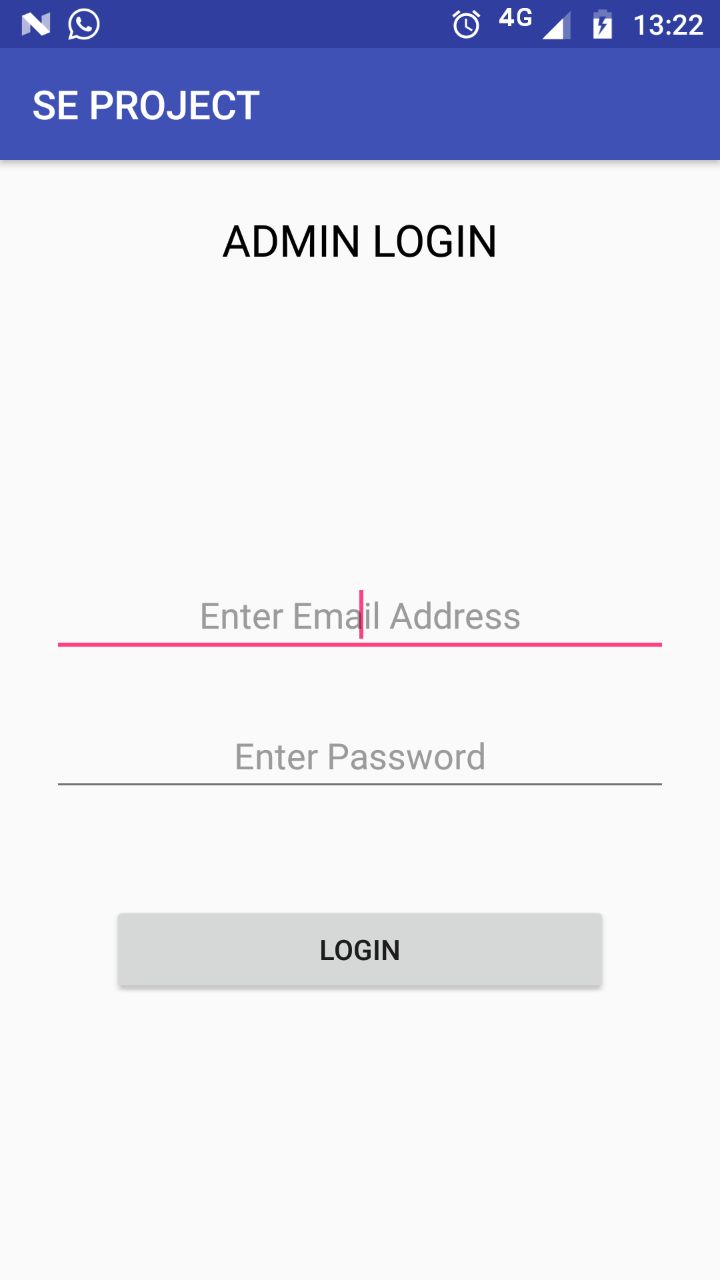
In this section of the project, the user has to login with the credentials of login ID and Password. Different application layout is there for the admin and other users.

For a new user, only the admin can create a new user using his password

provide the login details to him/her so that no other outsider will violate the security of the application.

Users can logout from his/her id at any time using the logout

home page.



PRESS HERE

ENTER ADMIN ID

ENTER PASSWORD HERE

7.2 RETRIEVAL

An interface will be displayed to the users so as to fetch the information regarding their favorite players or teams. Team achievements can also be retrieved by the user without any difficulty.

Given below is the snapshot by which the user can get the player’s information by entering the player’s name. The project is capable of calculating the rank of each player in ODI based upon their strike rate.

****

ENTER PLAYER NAME

PRESS HERE

1. **CONCLUSION**

We can conclude that the Cricket Management System can be used by people all across the globe to access the information of various players and teams to retrieve details along with the player personal details also. It can help in accessing new data and news about favourite players and teams. It is a modern approach to fetch data of players and teams, along with live scores.

It is based on advanced technology as it is compatible with all mobile sets having android version 4.4 or higher. Some of the advantages of CMS (Cricket Management System) are as follows:-

* Can ODI Tournament details on Internet.
* Players, Organizers, Selectors, Fan and Followers can access it from anywhere and anytime.
* The ODI Tournament can reach thousands of online users and gain major popularity.
* User-Friendly admin and easy to manage.
* Admin can add and edit information at any point of time and also from any location.
* Can share ODI Details through Social Media: Comments, posting, etc.
* Latest cricket news updates, ongoing ODI Events information is easy to retrieve.

1. **REFERENCES**

The successful completion of this project has been achieved by the assistance from various resources which include:

1. Software Engineering: A Practitioners Approach by Roger S. Pressman
2. An Integrated Approach to Software Engineering by P. Jalote
3. Software Engineering by K.K. Aggarwal
4. https://www.tutorialspoint.com/software\_engineering/index.html