Report of Software Engineering Project

“CRICKET DATA MANAGEMENT SYSTEM”

Submitted by

DISHANG VALOTIA (141070009)

B. Tech (Computer Engineering)

III Year (V Sem)

Under the guidance of

Prof. P. M. CHAWAN

Department of Computer Engineering and Information Technology

Veermata Jijabai Technological Institute

Mumbai-400019

(An Autonomous Institute affiliated to University of Mumbai)

2016-17

ACKNOWLEDGEMENT

I would like to thank all those people whose support and cooperation has been an invaluable asset during the development of this Project. I would also like to thank our Guide **Prof. P. M. Chawan** for guiding us throughout this project and giving it the present shape. It would have been impossible to complete the project without their support, valuable suggestions, criticism, encouragement and guidance.

I convey my gratitude also to **Dr. S. G. Bhirud**, Head of Department for his motivation and providing various facilities, which helped me greatly in the whole process of this stage of project.

I am also grateful for all other teaching and non-teaching staff members of the Computer Engineering and Information Technology Department for directly or indirectly helping me for the completion of this project and the resources provided.

Dishang Valotia

STATEMENT OF CANDIDATE

I state that work embodied in this Project titled "**Cricket Data Management System**" forms my own contribution of work under the guidance of Prof. P. M. Chawan at the Department of Computer Engineering and Information Technology, Veermata Jijabai Technological Institute. The report reflects the work done during the period of candidature but may include related preliminary material provided that it has not contributed to an award of previous degree. No part of this work has been used by me for the requirement of another degree except where explicitly stated in the body of the text and the attached statement.

Dishang Valotia

CERTIFICATE

This is to certify that Dishang Valotia (141070009), student of III Year (V Sem.) B. Tech. (Computer Engineering), Veermata Jijabai Technological Institute (VJTI), Mumbai has successfully completed the Project titled "**Cricket Data Management System**" under the guidance of Prof. P. M. Chawan.

Prof. P. M. Chawan

(Guide)

ABSTRACT

Cricket is a one of the most popular sport in India and is also popular around the world. Many records are made when cricket matches are played. People like to be aware and get knowledge of the various statistics of their favourite player or any other player in general. This system aims to provide these features to the people.

Currently there are many websites and softwares that provide these types of statistical information. But there can be issues with the statistics such as delayed updating of the various statistics, or particular statistics are not found, etc. This system aims to overcome these issues and provide a user friendly website so that people can view statistics quickly and provide feedback so that the more functionality can be added or incorrect can be rectified.

ARRANGEMENT OF CONTENTS OF PROJECT REPORT

1. Title sheet
2. Acknowledgement
3. Statement of the Candidate
4. Certificate Sheet
5. Abstract
6. Synopsis
7. Report Orientation
8. Table of Contents
9. List of Figures
10. List of Tables
11. Chapters
12. Reference

SYNOPSIS

INTRODUCTION

It is a software to computerize cricket management, to manage the database containing records of players of all countries who are members of ICC and also country records against each other and update these records whenever an ODI takes place. The database would contain various stats such as name, runs scored, wickets taken, strike rate, high score and so on.

Presently, multiple websites and softwares exist that provide statistical information of the cricketers. These face the problem of delayed statistic updating. There will be a delay between the match being played i.e. the statistics being changed (or updated) to the administrator updating it in the system.

This system aims to remove that delay by having multiple people (administrators) that can update the statistics. This helps in reducing the workload faced by each administrator in this system as compared to those systems in which only one or a very small amount of people update the statistics.

Another feature this system provides is to allow users to provide feedback to the administrators requesting a certain feature to be implemented or to provide information on incorrect statistics.

PROBLEM STATEMENT

“Cricket Data Management System”

The software aims to alleviate the cumbersome updating of player records in multiple records as compared to the existing system. Easy access of player stats is the motive of this software and to add to that the user will not have to calculate new stats and then update to the software, instead the software will save the manual calculation and queries and will provide easy method to simply update records on match to match basis.

REPORT ORIENTATION

The report is comprised of fourteen chapters with different content and scenarios providing the complete details about the project. The report is completed in such a way that it first provides the background knowledge about the project and then gives the thorough details about it. It also has conclusion. The different chapters of the report are as follows.

Chapter 1: Problem Statement

This chapter will provide introduction to the project and motivation for performing it. This chapter describes current scenario, previous attempts, and Problem statement and system requirements.

Chapter 2: Project Scope

This chapter will identify the scope of this project. This chapter describes the project scope in terms of objectives, goals, working of the system, facilities it provides to the user, and summary of the future scope.

Chapter 3: Resource Estimation

This chapter will analyze the resources that will be required so as to achieve project goals. This chapter calculates resources in terms of project resources, human resources, software requirements, and hardware resources.

Chapter 4: Effort Estimation

This chapter will analyze the efforts that will be required so as to achieve project goals. This chapter calculates resources in terms of lines of code, Function points, and COCOMO model.

Chapter 5: Risk Identification and Management

This chapter will first identify the various risks that threaten the integrity of the project and after comfortable and agreeable identification it deploys various methods so as to manage the specific risks.

Chapter 6: Project Scheduling and Tracking

This chapter defines the task set, task network, schedules the timeline chart, and finally tacks and schedules on a continuous basis.

Chapter 7: Project Planning

This chapter describes a specific model to plan the project development flow.

Chapter 8: Software Quality Assurance Plan

This chapter reviews the concept of quality concept, quality control, cost of quality, software quality assurance (SQA), and describes a specific SQA.

Chapter 9: Software Configuration Management

This chapter reviews the concept of quality concept, quality control, cost of quality, software quality assurance (SQA), and describes a specific SQA.

Chapter 10: Requirement Analysis Model

This chapter contains system architecture and Data flow diagrams. System Architecture shows structural view of system and data flow diagrams shows actual flow of data in the system.

Chapter 11: Design Modeling

This chapter contains UML diagrams with their significance. Use case, class diagram, sequence diagram, state chart diagram, activity diagram and component diagram is drawn to show the actual working of system.

Chapter 12: Software Testing

This chapter describes the method to test different modules of the project.

Chapter 13: Conclusion and Future Scope

This chapter gives the conclusion and the future scope of Cricket Data Management System.

Chapter 14: References

This chapter gives the details of reference books

Table of Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1. Introduction

It is a software to computerize cricket management, to manage the database containing records of players of all countries who are members of ICC and also country records against each other and update these records whenever an ODI takes place. The database would contain various stats such as name, age, runs scored, wickets taken, strike rate, high score and so on.

1.1 Problem Statement:

Cricket Database Management System

1.2 Background:

- **Previous Attempts:**

  Previously they used standalone softwares which were installed directly with the database and such a software faced a problem of updation versus the current system where the database is directly present in the server which can be accessed using an internet connection. Therefore the previous software could not be updated and also making changes in the database was difficult with the upcoming matches.

- **Current Scenario:**

  The existing system contains a database that stores all the details of players of major nations i.e. attributes like name, age, avg. runs scored, number of centuries/half centuries scored (for batsman) , avg. runs per ball, number of wickets (for bowler), highest scores and so on. Updating of database would require the user to enter the match-scorecard with batting-bowling details of both sides and also other details like venue, D/N etc. The software updates the attributes of all the tables based on the information provided by the user.

A limitation of this existing system is that every time a match is played, it has to be watched by the user and give input to the software after every ball is played. This is time consuming and requires manual work to be done.

1.3 Elaboration:

The software aims to alleviate the cumbersome updating of player records in multiple records as compared to the existing system. Easy access of player stats is the motive of this software and to add to that the user will not have to calculate new stats and then update to the software, instead the software will save the manual calculation and queries and will provide easy method to simply update records on match to match basis.

The main objectives of the proposed system can be enumerated as follows:

1. Easy access of player records.

2. Convenient updating methods without cumbersome calculations.

3. Updates will be made through the software and no need to directly access the database.

CHAPTER 2
PROJECT SCOPE

2. Project Scope

2.1 Features Identified:

The homepage will consist of the login for admin and users need not create an account to view the statistics and scores. The admin after logging in can update the score via the website itself and need not directly work on the database.

The user can access the database without logging in. On the homepage the user can view the various statistics and scores. Direct reference is made to the database from the homepage for the user to view the database.

The admin, after logging in, has the privileges to update the database. The admin will be taken to a web-page that will be inaccessible to the users where he/she can update the database.

The web-page will be developed using HTML, Bootstrap, CSS, Python and MySQL.

What work does the system do?

The software takes the name of the cricketer and identifies the player in the database and consequently returns the data corresponding to the player. The software allows the user to demand specific data if the user doesn't wish to view the rest of the data.

The software also provides easy updating of the database based on the matches played in the future. On entering the stats of a particular match the database uses that data and calculates the new stats of the player and also updates the database.

2.2 Benefits:

1. User friendliness of the software enables easy interaction between user and database.
2. Lesser time required as the update is made to a single database which is to the user's disposal.
3. The database is convenient for the user to expand on his/her own terms.
4. Statistics of all formats of all cricketers are present in a single location.

This project would be beneficial to the following people:-

1) Owner: The owners can upload the details of the cricketers like the name, runs, wickets, etc.

2) Users: The people who wish for analyse the cricket data can just visit this website, cricket fans.

2.3 Constraints:-

The system will be having following constraints after considering the time limits, complexity of the project and budget limitations as well.

- The system must be up and updated regularly.
- System will allow only the admins to update.
- The system will not be able to detect incorrect information being fed by the admins.

2.4 Performance Requirements:-

- **Response Time:-**

  It is measured in terms of time taken for web page to load from the server completely on the browser and is completely visible to the user. Estimated response time should be 2 seconds in 95% of the cases.

  The response time should not vary across the systems too much considering the hardware and software variations. The pages displaying, fetching or processing comparatively more amount of data than other pages are expected to respond within 7 to 9 seconds.

- **Workload:-**

Scenario	Daily Total	Pages	Think Time
Finding Cricketer	500	Login, Search, 50% Exit	20 secs
Registration	5	Login, User Register, 30% Exit	60 secs
Updating details	20	Login Admin, Update, 30% Exit	40 secs
Registering as a admin	10	Registration, Verification, 70%	90 secs
View Averages	700	Login, Search, Access DB, 60% Exit	30 secs

- **Scalability:-**

  As we know that the system may deal with huge amount of data in the future as it grows. Thus, we have set response time as low as possible for a database intensive website. The database design will be in such a way that it will be upgradable in the future and the data will be quickly accessed so the system will work in case the number of users increase surprisingly.

  Increasing number of users will not be an issue as the database does not give all the users the privilege to update database therefore major changes will not be associated with the users in the database making it a straightforward database only for users.

- **Platform:-**

  Since it is a website, the user may not need any kind of additional hardware. Though using a computer which works normally without any kind of lagging or hardware crashes is advisable. Obviously an internet connection of 256 kb/s is advisable for accessing the website with ease.

  On the software side, only a web browser is needed here. It is advised to have the latest updated web browser whichever you are using because some of the GUI features may not work on the older versions of that web browser.

CHAPTER 3
RESOURCE ESTIMATION

3. Resource Estimation

3.1 Project Resources:-

These include labour equipment (e.g. workstations)
a) Materials
b) Space
c) Services
d) Elapsed time can often be reduced by adding more staff

Resource Allocation:-

- Identify the resources needed for each activity.
- Identify resource types / individuals are interchangeable within the group.
- Allocate resource types to activities and examine the resource histogram.

3.2 Human Resources:-

Requirements other than these are:-

- The four members of group which are making this project are the human resources for this Software Development Life Cycle.
- One of us acts as a manager which commands the other three people to do the work and participates in doing the same.
- Each of us aim to dedicate 8-10 hours weekly for this project and the time durations for working for this projects will be adjusted according to changes in the deadlines or the requirements.

3.3 Hardware & Software Requirements:-

3.3.1 Hardware requirement:

Processor	:	Intel i3 or better
Cache Memory	:	2MB L2 cache
RAM	:	4 GB
Hard Disk Drive	:	Min. 20 GB
Monitor	:	14”SVGA

3.3.2 Software requirement:

OS	:	Windows 7/8/8.1/10
Front-end	:	Sublime Text 2/3, Python
Back-end	:	MySQL

CHAPTER 4
EFFORT ESTIMATION

4. Effort Estimation

4.1 LOC Based Estimation:

Function	Optimistic	Most Likely	Pessimistic	Estimated LOC
UI & Control	500	600	700	720
Registration of users	300	400	500	480
Database management	500	600	700	720
Searching, Updating Details	1000	1100	1200	1320
Estimated Lines of Code				3240

Estimated LOC = (Optimistic + (4 * Most Likely) + Pessimistic) / 5

- UI & Control:- (500+2400+700)/5=720
- Registration of users, projects and builders:- (300+1600+500)/5=480
- Database Management:- (500+2400+700)/5=720
- Searching and Updating details:- (1000+4400+1200)/5=1320

A historical data indicates that organizational average productivity for system of this kind is 800 LOC/pm labour rate $8000 per month based on this cost per line of code $10.
Total estimated project cost is 3240*$10=$32400
Months: 3240/800=4.05 months

4.2 FP-Based Estimation:

- Step 1:-
  Use an empirical relationship
  FP=count total x [0.65+0.01 x (∑Fi)]
  Cricket Database Management System FP based Calculations

Measurement Parameter	Count	Simple	Average	Complex	Function Points
No. of User Inputs	15	3	4	5	45
No. of User Outputs	15	4	5	7	60
No. of User inquiries	5	3	4	6	15
No. of files	7	7	10	15	49
No. of External interfaces	3	5	7	10	15
Total Count					**184**

- Step 2:-

Then we must consider 14 “Complexity Adjustment Values” rated on a scale of 0 – 5:

• No influence	0
• Incidental	1
• Moderate	2
• Average	3
• Significant	4
• Essential	5

Factor	**Value**
Backup and recovery	5
Data communications	3
Distributed processing	3
Performance critical	4
Existing operating environment	3
On-line data entry	4
Input transaction over multiple screens	1
Master files updated online	3
Information domain values complex	5
Internal processing complex	3
Code designed for reuse	3
Conversion/installation in design	3
Multiple installations	5
Application designed for change	5
TOTAL	**50**

Complexity adjustment factor = 0.65 + 0.01 x TOTAL
= **1.15**

- Total of FPs = 184
- Sum of all Fi’s = 50
- FP = 184 x [0.65+0.01 x 51]=211.6

System of this type is 12.3 FP/PM
Labour rate is $8000 per person per month
Cost per FP is approximately $650
Project Cost is $32500

4.3 COCOMO Based Estimation:

Software Projects	a_b	b_b	c_b	d_b
Organic	2.4	1.05	2.5	0.38
Semi-Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

This project is Organic type because it is a website. Effort is calculated as follows:-

E $=a_b\,(\text{KLOC})^{b_b}$
$=2.4\,(3.2)^{1.05}$
=8.1 person-months
D $=c_b\,(E)^{d_b}$
$=2.5\,(8.1)^{0.38}$
=5.53 Months
P =E/D
=1.46 Persons

CHAPTER 5

RISK ANALYSIS

&

MANAGEMENT

5. Risk Analysis & Management

5.1 Risk Identification:-

- **Generic Risks:**
    - The Database which has all the data about the cricketers becomes corrupted or non-responsive.
    - Work force may fall short due to the unavailability of team members.
    - Reduced time due to preponing deadlines.
    - Wrong updating might affect the calculation of further genuine data.
    - Updating database.
- **Product Specific Risks:**
    - The database might not be available for access if it goes down and will be unavailable for users to access until the admin restores the database.
    - Too many updates by the admins at the same time might possibly cause inconsistent database.
    - Due to manual updation by admins the software runs the possibility of human error.

5.2 Risk Projection or Risk Estimation:-

Steps to be followed are as mentioned below:-

(1) Establish a scale that reflects the perceived likelihood of a risk,
(2) Delineate the consequences of the risk,
(3) Estimate the impact of the risk on the project and the product, and
(4) Note the overall accuracy of the risk projection so that there will be no misunderstandings.

Impact values:-

1- catastrophic
2- critical
3- marginal
4- negligible

Risk ID	Risk	Risk Type	Probability	Impact
1	Project Delay	Project	40%	3
2	System Failure	Technical	20%	1
3	User Authentication (Misuse of System)	Technical	10%	2
4	Concurrent Access	Technical	30%	3
5	Manpower Unavailability	Project	20%	2
6	Crossing the stated budget	Project	50%	2
7	Internet Failure	Technical	10%	2
8	Database Failure	Technical	10%	2

5.3 Risk Mitigation, Monitoring & Management:

All of the risk analysis activities presented to this point have a single goal—to assist the project team in developing a strategy for dealing with risk. An effective strategy must consider three issues:-

(1) Risk avoidance
(2) Risk monitoring
(3) Risk management and contingency planning

Risk ID:-1 (Project Delay) Probability: 40% Impact:2
Description:- The time taken by the project to get completed exceeds the estimated time.
Refinement/Context:- This delay can cause a big loss the financial position and reputation of the organization. Client faces the inconvenience and it may require adding more people to the development team to meet the deadlines which is not advisable or feasible.
Mitigation, Monitoring and Management of Contingency Plan:- The whole project can be divided into comparatively very small tasks and then these tasks can be divided among developers. Developers will be ordered to accomplish such tasks as per the schedule and time limit given to them. These tasks can be divided on daily, weekly or on monthly bases as suited for the project. Developers should communicate with the supervisor or manager about any limitations or difficulties that they are facing so the solution can be found by some other person or the higher authority of the team. This way we can avoid the occurrence of the delay in the development process.

Risk ID:-2 (System Failure) Probability: 20% Impact:1
Description:- The system on which the website runs or is getting hosted i.e. a personal computer or a server or if the system gets compromised and some data is lost i.e. data from the database or some web page files etc. then the system will not work as intended and crash frequently.
Refinement/Context:- It directly hampers on the overall customer satisfaction and profit made by the organization. The resultant state of system may not be recoverable and fair amount of data can be lost due to such kind of failure.
Mitigation, Monitoring and Management of Contingency Plan:- We can run periodical backup processes which store the whole data of the whole system at some safe place where it cannot be harmed or compromised. Rectifying malicious code is also very important in this case. Any suspicious employee or outsider should not have access for modifying the contents of the system. Changes made in the system should be noted in a log so the recovery can be done easily if an inappropriate and undesirable change is made in the system.

Risk ID:-3 (Crossing the stated budget) Probability: 50% Impact:2
Description:- The budget for developing the system crossed the estimated limit.
Refinement/Context:- This situation can occur due to multiple reasons. The employees may not work as fast as they are supposed to, due to frequent change in the requirements the project may not finish in time and thus the organization has to pay more money to adjust the timings, due to complications in the project testing, handling and debugging the system may get difficult.
Mitigation, Monitoring and Management of Contingency Plan:- The system should be built in such a way that it is easy to make frequent changes and updates to it. Schedule must be strictly followed by all the members of team so they product is delivered well within the time after appropriate testing of the product. Some debugging and project management tools may also be used by the team members so the complications are reduced and bugs are easy to reproduce and fix. The budget requirements of project should be checked at regular intervals so the adjustments can be made accordingly and before a complete failure in managing the budget.

CHAPTER 6

PROJECT SCHEDULING

&

TRACKING

6. Project Scheduling And Tracking

For the scheduling of project, we take following steps:
Split the project into tasks and estimate time and resources required to complete each task.

1. Organize tasks concurrently to make optimal use of workforce and time.
2. Minimize task dependencies to avoid delays caused by one task waiting for the other to complete.
3. Depends on the manager's intuition and experience.

The process model, framework activities and task set are included here.

6.1 Task set for the project:

A task set is the collection of work tasks, milestones, and deliverables. Different set of task proves to be appropriate to different projects. An effective software process is the collection of task sets, each designed to meet the needs of different types of project Task sets are designed to accommodate different types of project and different degrees of rigor.

Typical project types:

- Concept development projects
- New application development project
- Application enhancement project
- Application maintenance project
- Reengineering project

Degree of rigour:

- Casual
- Structured
- Strict
- Quick reaction

The grades are allotted as
0-nonincedental
1-minimal
2-low
3-moderate
4-substantial

Task Selector Value Computation:

Adaptation Criteria	Grade	Weight	Product
Size of project	3	1.20	3.6
Number of potential users	4	1.10	4.4
Mission criticality	4	1.10	4.4
Application longevity	3	1.20	3.6
Stability of requirements	3	1.20	3.6
Ease of communication	3	0.90	2.7
Maturity of applicable technology	3	0.90	2.7
Performance constraints	3	0.80	2.4
Embedded/non embedded characteristics	1	1.20	1.2
Project staffing	3	1.00	3.0
Interoperability	4	1.10	4.4
Re/engineering factors	3	1.20	3.6

Task Selector Value=3.08

Task set selector value	Degree of rigor
TSS<1.2	Casual
1.0 < TSS <3.0	Structured
TSS > 2.4	Strict

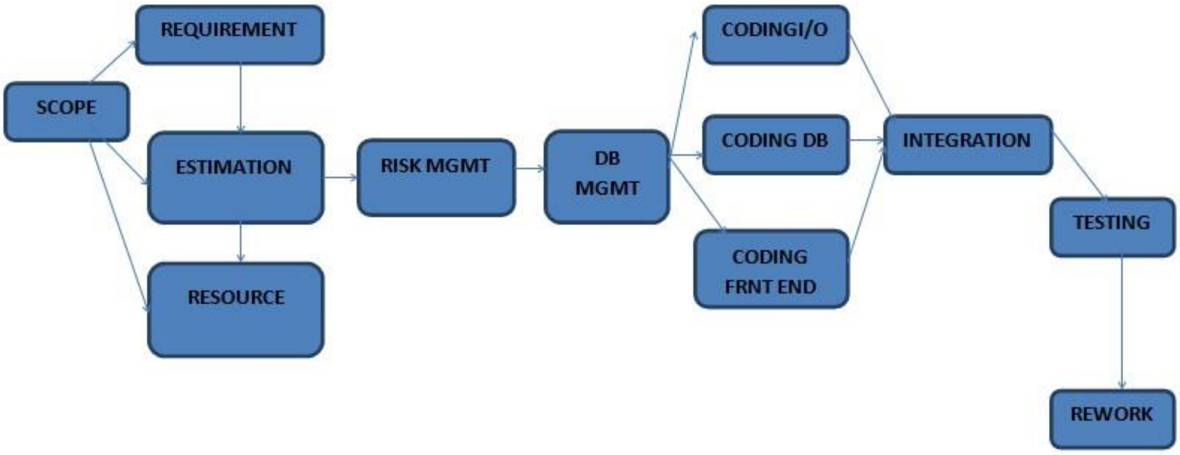
The overlap in ranges illustrates the fact that there exist no distinct boundaries between the three degrees of rigor. If a value that falls within the overlap is received we have to make a judgment call concerning the degree of rigor the project deserves.

6.2 Task Network:

Project tasks and dependencies are noted diagrammatically in task network. Functional decomposition is as shown.

Task	Description
T1	System Design
T2	Detailed Design
T3	Database Implementation
T4	Webpage Design
T5	Coding Input Module
T6	Coding database related module
T7	Coding Output Module
T8	Test Planning
T9	Integration
T10	Testing
T11	Rework
T12	Final Check

Activity Diagram:



6.3 SCHEDULING AND TIMELINE CHART

Tracking the Schedule Flow of Project Schedule:

```
Establish the project constraint
Make initial assessments of project parameters
Define project milestones and deliverables.
While project has not been completed or cancelled
    Draw up project schedule
    Initiate activities according to schedule
    Wait(for a while)
    Review project Progress
    Revise estimates of project parameters
    Update project schedule
    Re/negotiate project constraints and deliverables
    If(problems arise)then
        Initialize technical review and revision
    End if
End loop
```

Planning Table:

Task	Planned Start	Actual Start	Planned End	Actual End
Scope and objective, requirement gathering, planning	30/7/2016	30/7/2016	2/8/2016	31/7/2016
Estimation	2/8/2016	2/8/2016	12/8/2016	4/8/2016
Resources and arrange necessary tools and techniques, softwares,etc	2/8/2016	5/8/2016	12/8/2016	12/8/2016
Risk analysis &management	12/8/2016	15/8/2016	19/8/2016	19/8/2016
Scheduling tasks	19/8/2016	20/8/2016	26/8/2016	21/8/2016
Preparing database and coding it	26/8/2016	1/9/2016	2/9/2016	2/9/2016
Coding for all Modules and front end	2/9/2016	8/9/2016	9/9/2016	15/9/2016
Integrating the modules	9/9/2016	15/9/2016	16/9/2016	18/9/2016
Testing	16/9/2016	19/9/2016	23/9/2016	20/9/2016
Reworking	23/9/2016	23/9/2016	30/9/2016	24/9/2016
Finalizing the project	15/10/2016	15/10/2016	18/10/2016	19/10/2016

6.4 Tracking and Scheduling:

When you are managing a project, one of the most important responsibilities is effectively scheduling tasks and tracking their completion. This helps ensure that your project is completed on time and within budgetary constraints. There are different ways of tracking project.

1) Conducting different periodic project status meetings where team members report progress and problems.

2) Evaluating results of all reviews throughout the process.

3) Determining whether the formal milestones are accomplished within time as depicted by Gantt chart.

CHAPTER 7
PROJECT PLAN

7. PROJECT PLAN

The software project plan is produced at the culmination of the planning task. It provides baseline cost and scheduling information that will be used throughout the engineering process. The software project plan is relatively brief document that is addressed to diverse audience. An outline of the plan is presented below.

Project planning process

1. A statement of work (SOW) that describes all work produce that will be produced and a list of all resources people who will perform that work.
2. A resource list that contains a list of all resources that will be needed for the product and their availability.
3. A work breakdown structure and set of estimates.
4. We framed a project schedule and task network to plan the work considering the given deadlines.
5. A risk plan that identifies any risks that might be encountered and indicates how those risks would be handled should they occur.

The outline of the project is as presented below:

1. Definition and Introduction of Problem Statement :

A problem statement is a clear concise description of the issue(s) that need(s) to be addressed by a problem solving team.
Problem statement for this project is Cricket Database Management System.

2. Project Scope and objective:

Software scope describes the data and control to be processed, functions, performance requirements, constraints, interfaces, and reliability requirements.
Scope is defined by answering the following questions:

1) Context
2) Information Objectives
3) Function and Performance.

We have to design a software to computerize cricket management, to manage the database containing records of players of all countries who are members of ICC and also country records against each other and update these records whenever an ODI takes place. The database would contain various stats such as name, age, runs scored, wickets taken, strike rate, high score and so on.

The web-page will be developed using HTML, Bootstrap, CSS, Python and MySQL.

3. Project Estimation:

Software project estimation is a form of problem solving where the problem to be solved is too complex. For this reason, we decompose the problem.

Estimation Techniques: Could be direct or indirect.

- **Problem Based Estimation:** The project planner begins with a bounded statement of software scope and from this statement attempts to decompose software into problem functions that can each be estimated individually. LOC or FP (the estimation variable) is then estimated for each function.
- **LOC Based Estimation:** A method to measure size by counting number of lines of Code. When LOC is used as the estimation variable, decomposition is absolutely essential. The greater the degree of partitioning, the more likely reasonably accurate estimates of LOC can be developed. Estimated LOC is 3240 LOC
- **FP Based Estimation:** Function Point Analysis (FPA) is method to measure the functional size of an information system. The functional size reflects the amount of functionality that is relevant to and recognized by the user in the business. FP calculated is 211.6
- **Process Based Estimation:** The process is decomposed into a relatively small set of tasks and the effort required to accomplish each task is estimated. Estimated cost is $32400.

4. Project Resources:

Resources are commonly thought of as sources of supply or support, such as money, people, materials, technology, and space.

- **Human**: It is personnel pool like developers, managers, planners, testers etc. available to an organization
- **Reusable**: Reusable resources are the sources already available while doing some other projects or creating the new component to reuse it in some other project.
- **System Requirements**: It consists of planning, managing and making the necessary software/hardware components available.

People-

Four team members which are making this project are the human resources for this Software Development Life Cycle.

Hardware requirement:

Processor	:	Intel i3 or better
Cache Memory	:	2MB L2 cache
RAM	:	4 GB
Hard Disk Drive	:	Min. 20 GB
Monitor	:	14"SVGA

Software requirement:

OS	:	Windows 7/8/8.1/10
Front-end	:	Sublime Text 2/3, Python
Back-end	:	MySQL

5. Risk Analysis and Management:

Risk analysis and management are a series of steps that help a software team to understand and manage uncertainty. It consists of the following steps:

- **Identification**: Risk identification is a systematic attempt to specify threats to the project plan. By identifying known and predictable risks, the project manager takes a first step toward avoiding them when possible and controlling them when necessary.
- **Projection**: Risk projection, also called risk estimation, attempts to rate each risk in two ways—the likelihood or probability that the risk is real and the consequences of the problems associated with the risk, should it occur.
- **Risk Mitigation, Monitoring and Management**: All of the risk analysis activities presented to this point has a single goal—to assist the project team in developing a strategy for dealing with risk.

6. Project Scheduling and Planning

- **Defining Task Set for Software Project**: A task set is the collection of work tasks, milestones, and deliverables.
- **Task Network**: Project tasks and dependencies are noted diagrammatically in task network according to its functional dependencies.
- **Time Line chart**: Project milestones can be shown in a simple time line chart .While the chart doesn’t look complicated, it provides good amount of information on project progress in a simple and understandable chart.
- **Tracking the Schedule:** Tracking helps ensure that your project is completed on time and within budgetary constraints.

7. Project Plan:

Goal is to establish a pragmatic strategy for controlling, tracking, and monitoring a complex technical project. The purpose of project planning is to ensure that the end result is completed on time, within budget, and exhibits quality. The steps involved are: Project Scope Estimates, Risks, Schedule & Control strategy.

8. Software Quality Assurance:

Software quality assurance (SQA) is an umbrella activity that is applied throughout the software process.

SQA encompasses:

(1) A quality management approach,
(2) Effective software engineering technology (methods and tools),
(3) Formal technical reviews that are applied throughout the software process,
(4) A multitier testing strategy,
(5) Control of software documentation and the changes made to it,
(6) A procedure to ensure compliance with software development standards
(7) Measurement and reporting mechanisms.

- **Quality Concept**: Quality is a characteristic or attribute of something. In our software project it is of 2 types: quality of design which encompasses requirements and quality of conformance that focuses on implementation.
- **Quality Control**: Quality control involves the series of inspections, reviews, and tests used throughout the software process to ensure each work product meets the requirements placed upon it.
- **Cost of Quality**: The cost of quality includes all costs incurred in the pursuit of quality or in performing quality/related activities.
- **SQA**: Stands for Software Quality Assurance. Quality assurance consists of the auditing and reporting functions of management.
- **SQA Plan**: The plan is developed during project planning and is reviewed by all interested parties. Quality assurance activities performed by the software engineering team and the SQA group are governed by the plan.

9. Software Configuration Management:

Software configuration management (SCM) is an umbrella activity that is applied throughout the software process. Because change can occur at any time, SCM activities are developed to

(1) Identify change
(2) Control change
(3) Ensure that change is being properly implemented
(4) Report changes to others who may have an interest.

- **SCM**: Software configuration management (SCM) is a set of activities designed to control change by identifying the work products that are likely to change, establishing relationships among them, defining mechanisms for managing different versions of these work products, controlling the changes imposed, and auditing and reporting on the changes made.
- **SCM Process**: The software configuration and management process consists of the following five SCM tasks: identification, version control, and change control, configuration auditing, and reporting.

10. Requirement Analysis Modelling:

- **Data Modelling**: ER Diagram: An entity relationship diagram is an excellent tool for planning and designing a database. The entity relationship model starts with the entities, data normalization starts with the attributes and the two tools tend to verify each other. The entity relationship model’s entities, attributes and relationships map smoothly to a physical database.

- **Functional Modelling**: Data Flow Diagram: DFDs are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing. A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored.

- **Data Dictionary**: A data dictionary is a collection of descriptions of the data objects or items in a data model for the benefit of programmers and others who need to refer to them.

11. Design:

Designing is a constructive task. Software should be designed in a manner that uses anti bugging techniques. That is, software should be capable of diagnosing certain classes of errors. In addition, the design should accommodate automated testing and regression testing.

12. Project maintenance and Reusability:

Software maintenance in software engineering is the modification of a software product after delivery to correct faults, to improve performance or other attributes. The key software maintenance issues are both managerial and technical.

Reusability is the likelihood a segment of source code that can be used again to add new functionalities with slight or no modification. Reusable modules and classes reduce implementation time, increase the likelihood that prior testing and use has eliminated bugs and localizes code modifications when a change in implementation is required.

CHAPTER 8

SOFTWARE QUALITY ASSUARANCE

8. Software Quality Assurance

SQA is the process of evaluating the quality of a product and enforcing adherence to software product standards and procedures. It is an umbrella activity that ensures conformance to standards and procedures throughout the SDLC of software product. There are a large number of tasks involved in SQA activities.

The software quality control process for this project will consist of the following:

SQA Process	Description
SRS Review	The SRS will be reviewed by project team.
Design Review	Design document will be reviewed by project team.
Unit Testing	Programmer is responsible for Unit Testing of each module.
System Testing	It will be done according to system test plan, which will be then reviewed.

8.1 Quality Concepts:

Software quality assurance (SQA) is an umbrella activity that is applied throughout the software process. SQA encompasses

- Formulating a quality management plan.
- Applying software engineering techniques
- Conducting formal technical reviews
- Applying a multi-tiered testing strategy
- Enforcing process adherence
- Controlling change
- Measuring impact of change
- Performing SQA audits
- Keeping records and reporting

Quality refers to measurable characteristics of software. These items can be compared based on the given standards

Quality of Design:
Quality of design refers to the characteristics that designers specify for an item. The grade of materials, tolerances, and performance specifications all contribute to the quality of design. As higher/grade materials are used, tighter tolerances and greater levels of performance are specified, the design quality of a product increases, if the product is manufactured according to specifications.

Quality of Conformance:
Quality of conformance is the degree to which the design specifications are followed during manufacturing. Again, the greater the degree of conformance, the higher is the level of quality of conformance.

Crucial is customer satisfaction (quality is only a part of it):
User Satisfaction= Compliant product +Quality +Delivery within budget and Schedule

8.2 Quality Control:

Quality control involves the series of inspections, reviews, and tests used throughout the software process to ensure each work product meets the requirements placed upon it. Quality control includes a feedback loop to the process that created the work product. A key concept of quality control is that all work products have defined, measurable specifications to which we may compare the output of each process. The feedback loop is essential to minimize the defects produced. Quality assurance consists of the auditing and reporting functions of management. The goal of quality assurance is to provide management with the data necessary to be informed about product quality, thereby gaining insight and confidence that product quality is meeting its goals.

8.3 Cost of Quality:

Cost of quality includes all costs incurred in the pursuit of quality or perform quality related work.
Quality cost includes:

Prevention costs include

- Quality planning
- Formal technical reviews
- Test equipment
- Training

Appraisal costs include activities to gain insight into product condition the "first time through" each process.
Appraisal costs include

- In-process and inter-process inspection
- Equipment calibration and maintenance
- Testing

Failure costs are those that would disappear if no defects appeared before shipping a product to customers. Failure costs may be subdivided into

1. Internal failure costs
2. External failure costs.

Internal failure costs are incurred when we detect a defect in our product prior to shipment. Internal failure costs include:

- Rework
- Repair
- Failure mode analysis

External failure costs are associated with defects found after the product has been shipped to the customer. Examples of external failure costs are

- Complaint resolution
- Product return and replacement
- Help line support

- Warranty work

8.4 Software Quality Assurance:

Software quality is defined as

Conformance to explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software.

The definition serves to emphasize three important points:

- Use requirements as the foundation
- Use specified standards as the criteria
- Considering implicit characteristics

8.5 Software Quality Assurance Plan:

1. **Introduction**:

The SQA plan provides a road map for instituting software quality assurance.

- **Purpose**:

This document outlines the actions of our team in order to make our system “CRICKET DATABASE Management System” and other related artefacts conform to the requirements of the customers and the qualitative standards within the specified project resources and constraints following IEEE standards.

- **Scope**:

The scope of this document is to outline all procedures, techniques and tools to be used for quality assurance of this project.

This plan:
1. Identifies the SQA responsibilities of the project developer and the SQA consultant
2. Lists the activities, processes, and work products that the SQA consultant will review and audit.
3. Identifies the SQA work products

2. Management

A description of each major element of the organization and a description of the SQA tasks and their relationships. It consists of the following topics:

- Organization: This project is a team work where strength of each team is 4.
- Tasks and Responsibilities: The responsibilities of the developers are as follows:
  - ➢ Develop the requirement specification and cost estimation for the project
  - ➢ Develop the design plan and test plan for testing the tool
  - ➢ Implement and test the application and deliver the application along with the necessary documentation.

- Give a formal presentation to the committee on completion of the analysis, design and testing phases. The committee reviews the developer's work and provides feedback/suggestions.
- Planning, coordinating, testing and assessing all aspects of quality issues.

The responsibilities of the committee members are to:

1. Review the work performed by the developer
2. Provide feedback and advice

3. Documentation

In addition to this document, the essential documentation will include:

The Software Requirements Specification (SRS), which prescribes each of the essential requirements (functions, performances, design constraints and attributes) of the software and external interfaces.

The Formal Specification Document, which gives the formal description of the product design specified in Object Constraint Language (OCL).

The Software Design Description (SDD) which depicts how the software will be structured.

Software Test Plan: Describes the test cases that will be employed to test the product.

Software User Manual (SUM) which will identify the required data and control inputs, input sequences, options, program limitations or other actions.

4. Standards, practices, conventions, and metrics:

Metrics: LOC / lines of code and Function Points (FP) are used to measure the size of the software.

5. Reviews and audits:

The Committee will perform reviews at various stages of the project. This review will determine whether the requirements have been met for the deliverable, check that the product meets the requirements, ensure that the SQA plan has been adhered to, verify the performance of the software and ensure that acceptance testing is carried out.

In addition the developer will conduct a Formal Technical Review after the design phase. A design checklist will be used and the developer will check to see whether his/her design meets the checklist criteria.

6. Test:

Testing will be carried out in accordance with the Software Testing Plan (STP). Testing documentation will be sufficient to demonstrate that testing objectives and software requirements have been met. Test results will be documented and discussed in the final phase of the project.

7. Problem reporting and corrective action:

The corrective action process describes the steps for
1. Problem identification and correction occurring during software development to verify early detection of actual or potential problems
2. Reporting of the problem to the proper authority,
3. Analysis of the problem to propose corrective measures,
4. Timely and complete corrective action
5. The recording and follow/up of each problem's status.

8. Tools, techniques, and methodologies

The tools are evaluated for adequacy by assessing whether they perform the desired functions and for applicability by assessing whether the tool capabilities are needed for the software development or support. Planned tools are evaluated for feasibility.

9. Code control:

Code control includes the items listed below:

a) Identifying, labelling, and cataloging the software to be controlled
b) Identifying the physical location of the software under control
c) Identifying the location, maintenance, and use of backup copies
d) Distributing copies of the code
e) Identifying the documentation that is affected by a change
f) Establishing a new version
g) Regulating user access to the code.

SQA will conduct ongoing evaluations of the code control process to verify that the process of controlling the code is effective and in compliance with reference (f)

10. Records collection, maintenance, and retention:

SQA activities are documented by records and reports that provide a history of product quality throughout the software life cycle. Measurement data collected will be reviewed for trends and process improvement. All SQA records will be collected and maintained in the SDL or archival storage for the life cycle of the product or a minimum of few years.

11. Risk management:

Risk Management is done according to software documentation audit of project.

CHAPTER 9

SOFTWARE CONFIGURATION MANAGEMENT

9. Software Configuration Management

Software configuration management is the discipline of managing the evolution of complex software systems. It is also defined as 'the process of controlling and monitoring change to work products'.

9.1 Configuration Management:

The items that comprise all information produced as part of the software process are collectively called a software configuration for e.g. computer programs, documents and data.

Changes are inevitable and in most of the cases, justified. Customers may have modified requirements. Developers may want to modify the technical approach. Managers want to modify the project strategy.

Changes should be

1. Analysed in advance
2. Recorded before implementation
3. Reported by the need/to know basis
4. Controlled to improve quality and reduce errors.

The sources of changes can be stated as:

1. New business or market conditions.
2. New customer needs demand modification of data produced by information systems, functionality delivered by products, or services delivered by a computer based system.
3. Reorganization or business growth/downsizing causes.
4. Budgetary or scheduling constraints.

9.2 Software Configuration Management (SCM) Process:

SCM Umbrella Activity has following subtasks

- Identifies
- Controls
- Audits
- Report modifications

that invariably occur while software is being developed and after it has been released to a customer. All information produced as part of software engineering becomes part of a software configuration. The configuration is organized in a manner that enables orderly control of change.

SCI: Software Configuration Items:

There are growing number of artefacts for manage sharing the SE process. They are

1. **Programs**: source level and executable forms
2. **Documents**: technical practitioners and users
3. **Data**: internal and external

Types of SCI:

1. Requirements specification
2. Project plan
3. Preliminary user manual
4. Design specification
5. Source code listings
6. Test specifications
7. Installations/operations
8. Executable programs
9. Database description
10. As/built user manual
11. Maintenance documents
12. Standards and procedures

SCM TASKS:

1. Identifying change:

The identification scheme for software objects must recognize that objects evolve throughout the software process .Evolution graphs for each SCIs are used .For example in EMS, the changes can be in terms of number of people accessing site simultaneously may increase.

2. Version control:

Version control combines procedures and tools to manage different versions of configuration objects that are created during the software process.

3. Control change:

An engineering change order (ECO) is generated for each approved change. The ECO describes the change to be made, the constraints that must be respected, and the criteria for review and audit

E.g.: For the above change, an extra server may be arranged as control measure.

4. Configurational Auditing:

1) Ensure proper implementation:

A software configuration audit complements the formal technical review by assessing a configuration object.

5. Reporting changes to others

CHAPTER 10
REQUIREMENT ANALYSIS MODELLING

10. Requirement Analysis Modelling

Requirements analysis in systems engineering and software engineering, encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, analyzing and managing software or system requirements.

Requirement analysis modelling uses combination of text and diagrammatic form to depict requirements for data, function and behaviour in a way that is relatively easy to understand and to review for correctness, completeness and convenience.

10.1 Data Modelling:

Entity Relationship Diagram:

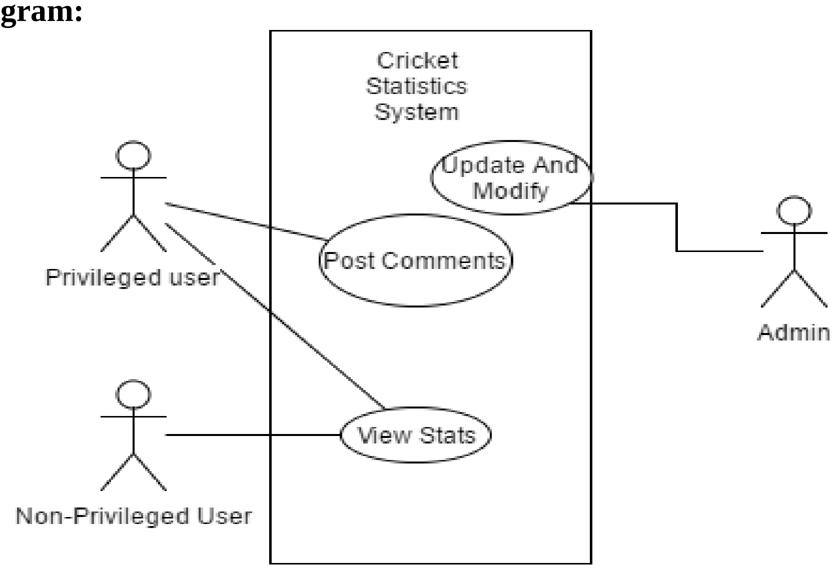
ERD depicts relationships between data objects. The ERD is the notation that is used to conduct the data modelling activity. The attributes of each data objects noted in the ERD can be described using data object description.

Use Case Analysis:

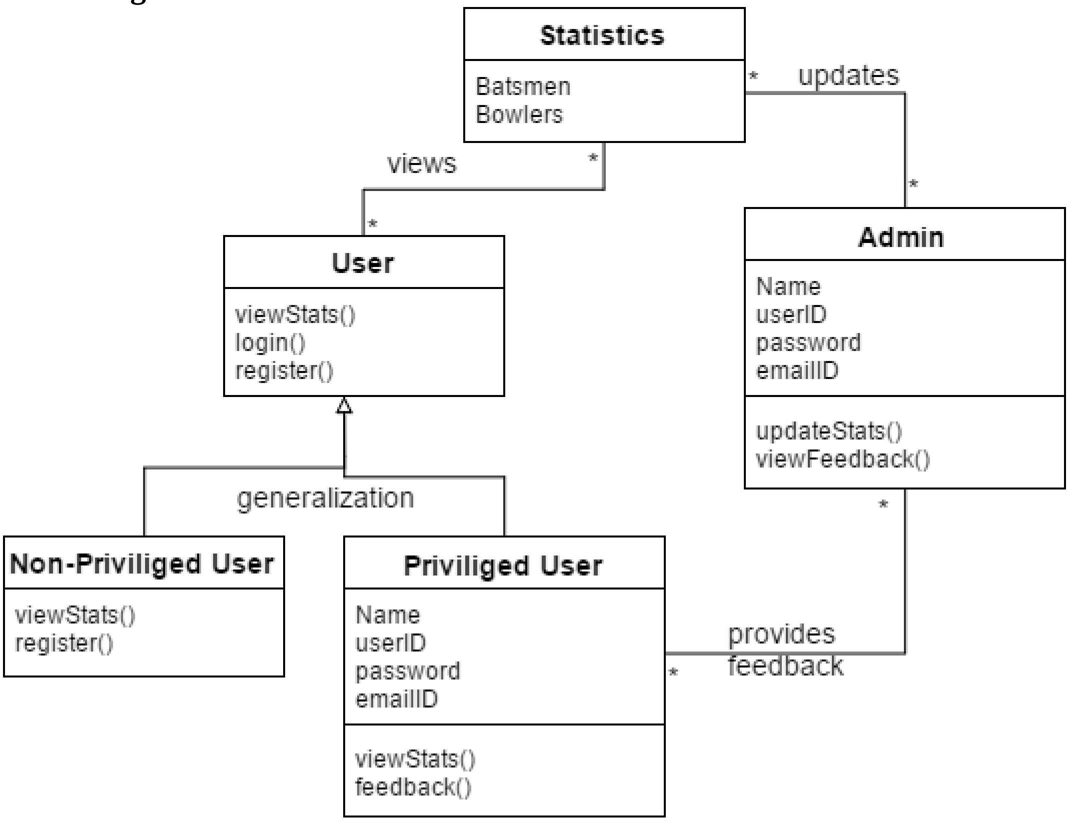
Use Case for cricket database management System

Actor List	Role
Administrator	Add, delete and modify the information.
Privileged Users	Can view statistics and post comments.
Non-Privileged Users	Can view only statistics.

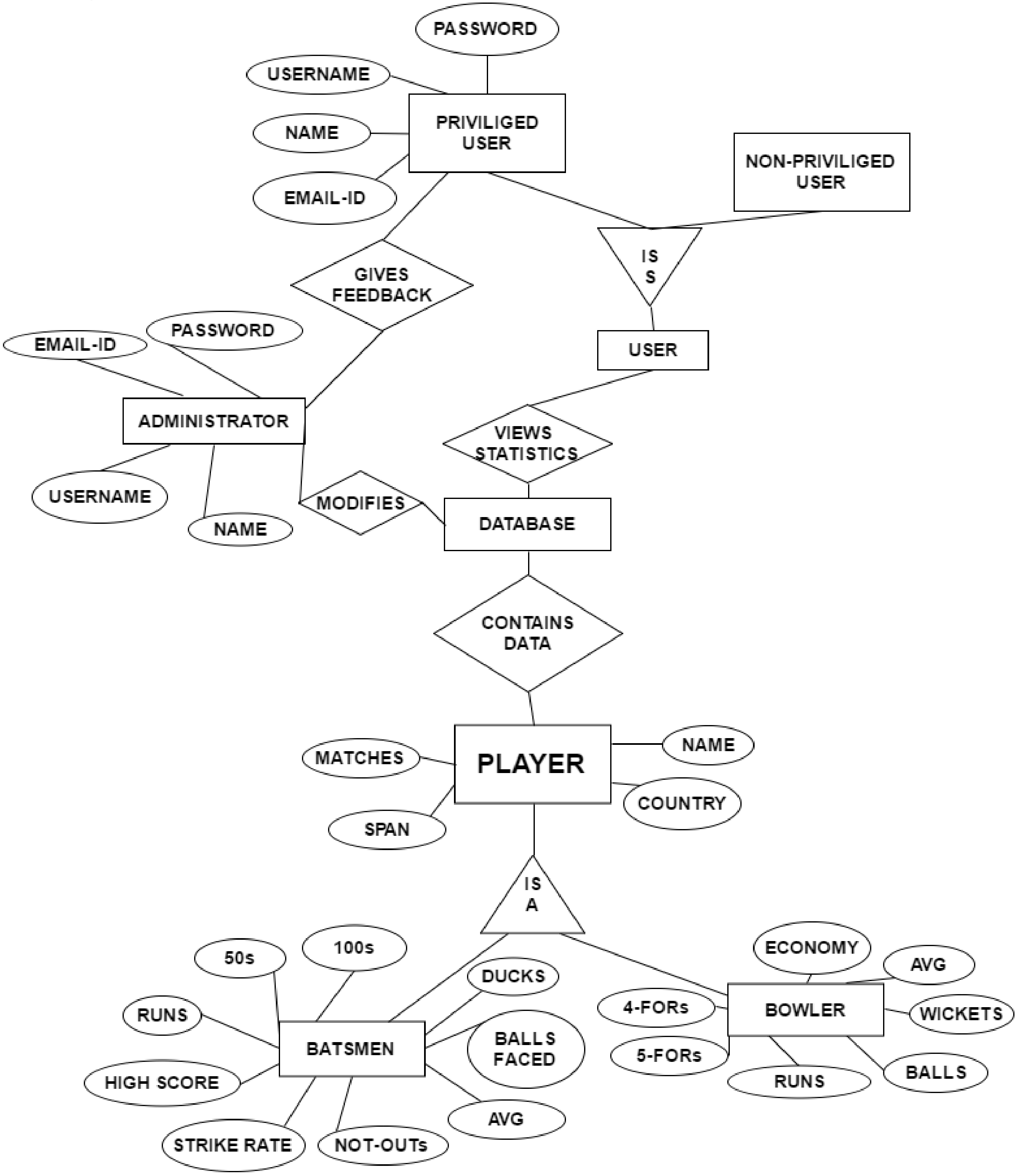
Use Case Diagram:



Class Diagram:



ER Diagram:

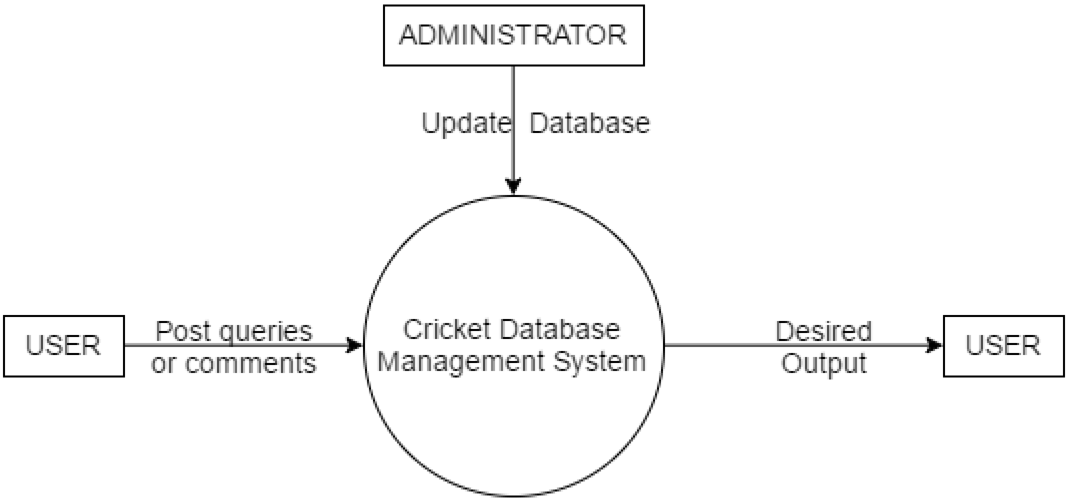


10.2 Functional Modelling:

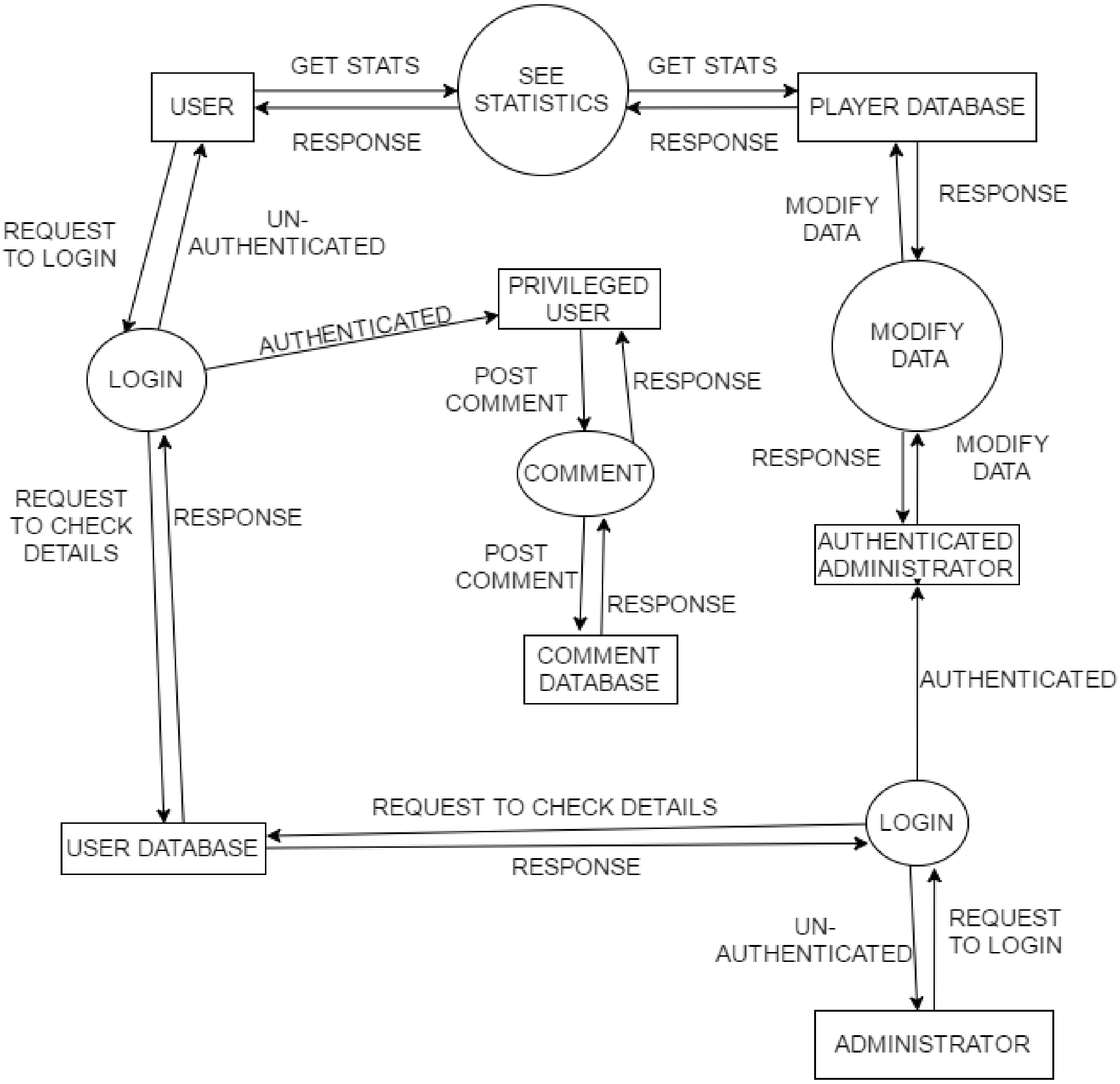
Data Flow diagram

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

DFD Level 0

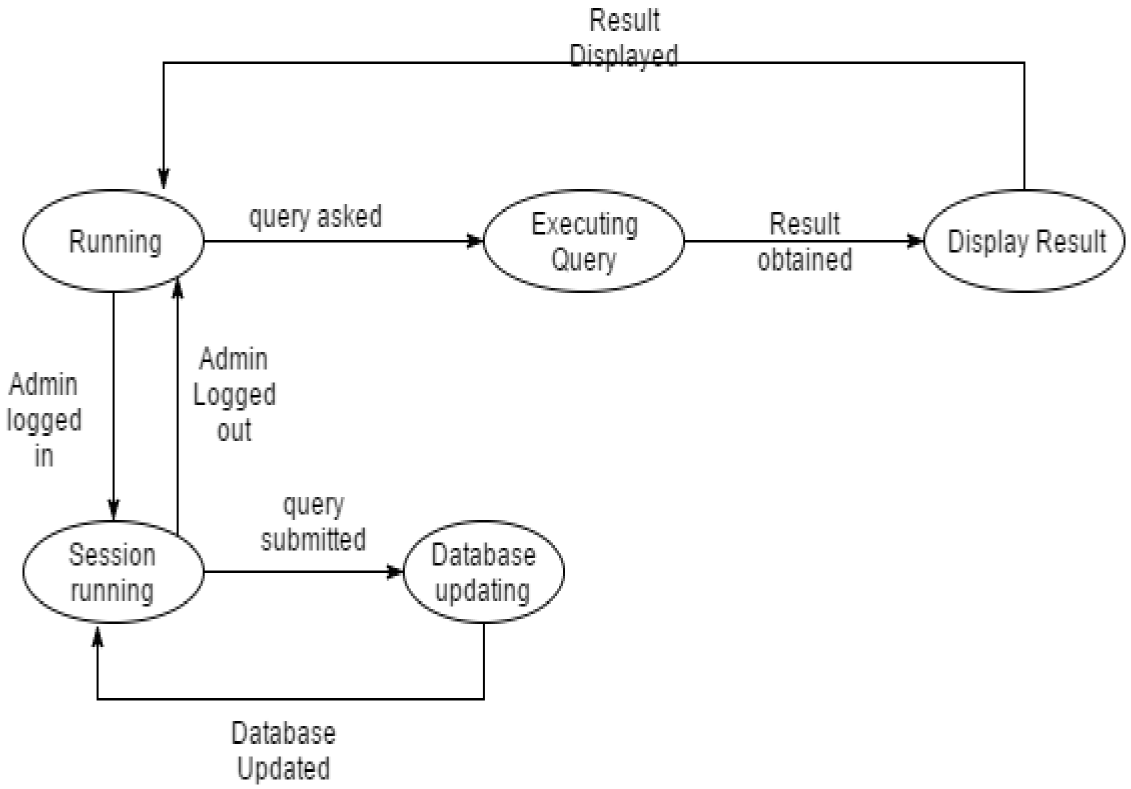


DFD Level 1

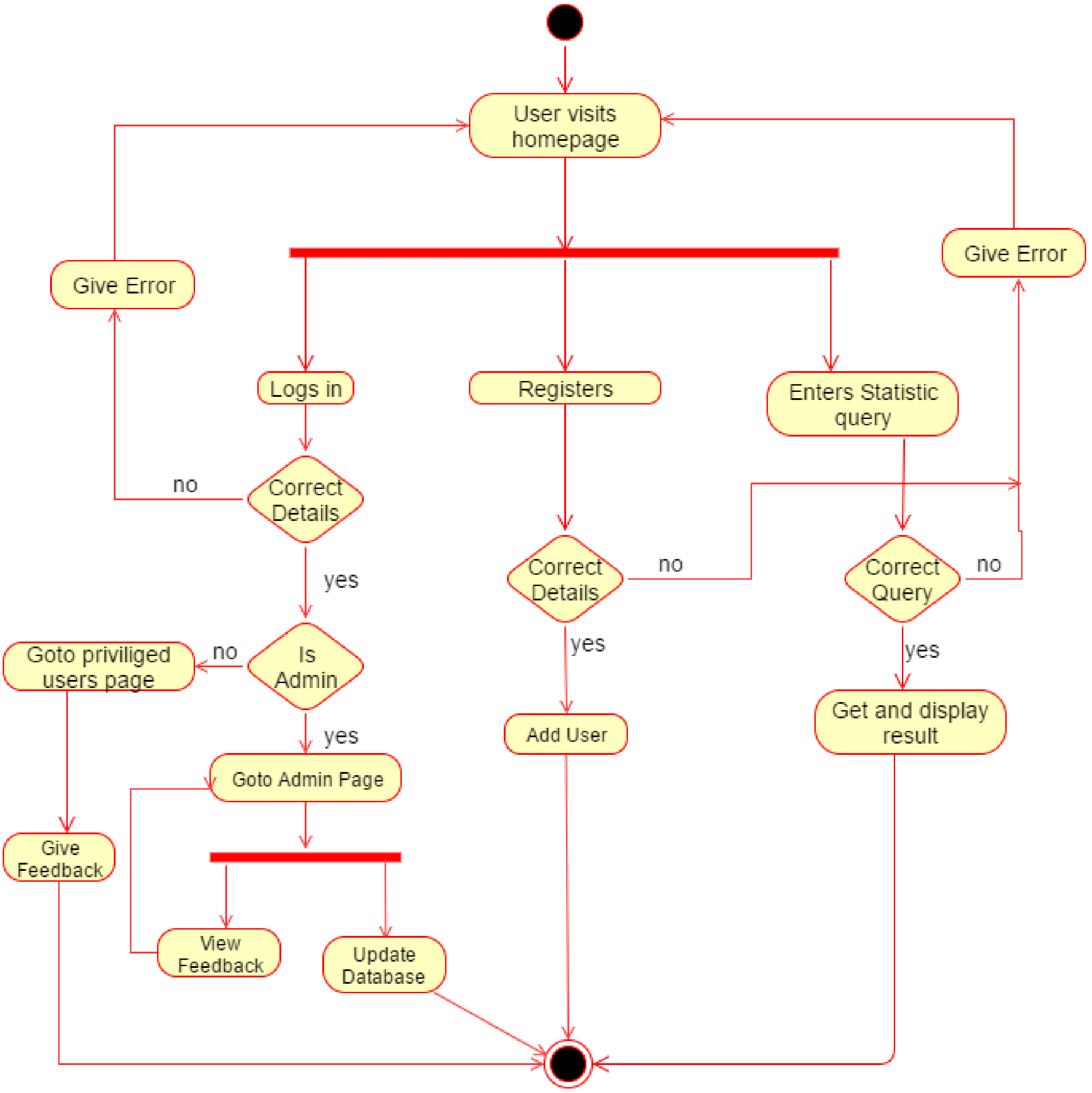


10.3 Behavioural Model:

State Transition Diagram:



Activity Diagram:



10.4 Data Dictionary

A data dictionary, or data repository, is a central storehouse of information about the system's data. The main purpose of a data dictionary is to describe, document and organize facts about the:

- **Data flows,**
- **Data stores,**
- **Processes,**
- **External entities.**

Title of Object Class	Description
Users	People who want to view statistics
Privileged users	Contains details of users that want to provide feedback or post comments
Non-privileged users	Users that just view statistics
Players	Contains general details of cricketers
Batsmen	Contains batting related statistics of cricketers
Bowlers	Contains bowling related statistics of cricketers
Administrators	Contains details of administrator that can modify database
Database	Contains details of all players

Algorithms

To view statistics:

```
1. Enter fields to view statistics
2. Submit the query
3. The query is submitted to database and result is returned
4. If result is not empty
   a. Then display result to user
   b. Else tell the user that no such statistics are present
```

To modify database:

```
1. Enter username and password for admin
2. If both are valid
   a. Then redirect admin to page for modifying database
   b. Else print incorrect login details and go to step 1
3. Admin posts data to be modified on the page for modifying database
4. Query is submitted to database
5. If query is executed successfully
   a. Then print Database updated
   b. Else print Error has occurred
6. If logout is pressed return to home page
```

To make an account:

1. Click create account button
2. Fill in required account details
3. Check correctness of details entered
4. Click submit
5. If account can be created
    a. Then print Account has been created
    b. Else print Account could not be created
6. Return to home page

To post comments:

1. Enter username and password
2. If both are valid
    a. Then continue
    b. Else print incorrect login details and go to step 1
3. Enter what you want to comment about in comment box
4. Press submit
5. If comment is added
    a. Then print comment added
    b. Else print comment could not be added

10.5 Software Requirements Specification

Revision History:

Name	**Date**	**Reason For Changes**	**Version**
Cricket Statistics	04/10/16	Initial Version	1.0

10.5.1 Introduction

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations, references and overview of the SRS. The aim of this document is to gather and analyze and give an in-depth insight of the complete Cricket Database System by defining the problem statement in detail. Nevertheless, it also concentrates on the capabilities required by stakeholders and their needs while defining high-level product features. The detailed requirements of the Cricket Database System are provided in this document.

Purpose

The purpose of the document is to collect and analyze all assorted ideas that have come up to define the system, its requirements with respect to consumers. Also, we shall predict and sort out how we hope this product will be used in order to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the product develops.

10.5.1.2 Document Conventions

TERM	DEFINITION
Admin/Administrator	System administrator who is given specific permissions for managing and controlling the system.
SRS	Software Requirements Specification –This document
User	Someone who interacts with the application.
Cricketer/Players	People who play cricket and whose stats are provided by this software.

Intended Audience and Reading Suggestions

This Software Requirements document is intended for:

Developers who view project capabilities and more easily understand where their efforts should be targeted to improve or add more features to it both design and code wise as it sets the guidelines for future development.

Project Testers can use this document as a base for their testing strategy as some bugs are easier to find using a requirements documents. This way testing becomes more methodologically organised.

This document can be viewed by users who wish to provide extended functionalities for the existing software or just wish to implement it.

Product Scope

We have to design a software to computerize cricket management, to manage the database containing records of players of all countries who are members of ICC and also country records against each other and update these records whenever an ODI takes place. The database would contain various stats such as name, age, runs scored, wickets taken, strike rate, high score and so on.

The web-page will be developed using HTML, Bootstrap, CSS, Php and MySQL.

This SRS is also aimed at specifying requirements of software to be developed but it can also be applied to assist in the selection of in-house and commercial software products. The standard can be used to create software requirements specifications directly or can be used as a model for defining an organization or project specific standard. It does not identify any specific method, nomenclature or tool for preparing an SRS.

References

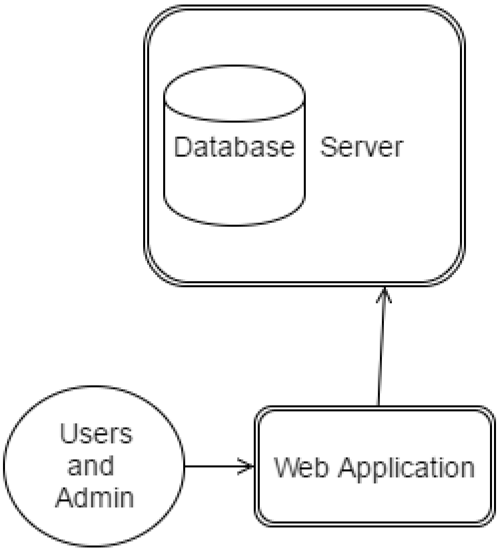
[1] IEEE Software Engineering Standards Committee, “IEEE Std 830 - 1999, IEEE Recommended
Practice for Software Requirements Specifications by Karl E. Wiegers”.

10.5.2 Overall Description

This document contains the problem statement that the current system is facing which is hampering the growth opportunities of the company. It further contains a list of the stakeholders and users of the proposed solution. It also illustrates the needs and wants of the stakeholders that were identified in the brainstorming exercise as part of the requirements workshop. It further lists and briefly describes the major features and a brief description of each of the proposed system. It provides the detail product functions of cricket management system with user characteristics permitted constraints, assumptions and dependencies and requirements subsets.

Product Perspective

The system consists of two parts: a web application and a server. The web app will be used by users and administrators who wish to access/view/modify the statistics of cricketers stored in the database on the server.



Block Diagram

Product Functions

1. Configuring Statistics using Log in.

- The system shall display all the cricket statistics.
- The system shall allow user to find statistics.
- The system shall enable admin to add one or more statistics to the database.
- The system shall allow user to provide feedback to resolve conflict in the current statistics.
- The system shall allow admin to confirm the completion of current statistics.

2. Provide comprehensive cricketer details without Log in.

- The system shall display detailed statistics of the demanded cricketer.
- The system shall provide browsing options to see statistics but can't make changes to it.

3. Provide Search facility.

- The system shall enable user to enter the search statistics on the screen.
- The system shall enable user to select multiple options on the screen to find statistics.
- The system shall display all the matching statistics based on the search
- The system shall notify the user when no matching product is found on the search.

4. Maintain customer profile.

- The system shall allow user and admin to create profile and set his credential.
- The system shall authenticate user credentials to view the profile.
- The admin will be a different account type which will be approved by the existing admins.

5. Provide personalized profile

- The system shall allow user to register for admin accounts and make changes in the profile.
- Every profile will have comments associated with it.

6. Allow updation methods.

- The system shall display updation method for statistics.
- The system shall allow admins update stats for changes.

User Classes and Characteristics

Three types of user can access this software: Privileged users, Non-Privileged users and Administrators.
Their characteristics are defined as follows:

User Class	Characteristics
Administrator	Add, delete and modify the information.
Privileged Users	Can view statistics and post comments.
Non-Privileged Users	Can view only statistics.

Operating Environment

The software is a website so it can be viewed on any computer that has an internet browser and network connectivity.

Design and Implementation Constraints

- The user's computer is constrained by the server's capabilities. It is a website so it solely depends on network connectivity for data transfer.
- The website will be constrained by the capacity of the database. Multiple accesses to the database simultaneously should not hinder the performance.

Assumptions and Dependencies

- The computers must be equipped with web browsers such as Internet explorer.
- A general knowledge of basic computer skills is required to use the product

10.5.3 External Interface Requirements

User Interfaces

- The user interface for the software shall be compatible to any browser such as Internet Explorer, Mozilla or Chrome by which user can access to the system.
- The user interface shall be implemented using any tool or software package like Bootstrap, php, html.

Hardware Interfaces

- Since the application must run over the internet, all the hardware shall require to connect internet will be hardware interface for the system. As for e.g. Modem, WAN – LAN, Ethernet Cross-Cable.

Software Interfaces

- The cricket management system shall communicate with the Configurator to identify all the available components to configure the product.
- The cricket management system shall communicate with the content manager to get the product specifications, statistics.
- The cricket management system shall communicate with system to identify available statistics.
- The cricket management system shall communicate with admin for tracking changes of Updation methods.

Communication Interfaces

- The cricket management system shall use the HTTP protocol for communication over the internet and for the intranet communication will be through TCP/IP protocol suite.

10.5.4 System Features

This sections describes each application's feature in detail in terms of stimulus/ response sequence and their functional requirements.

1. View Statistics

1.1 Description and Priority

The users should be able to view statistics of players and this is the main function of the website and hence is of the highest priority.

1.2 Stimulus/Response Sequences

Stimulus: User enters the required details of the players.
Response: The requested details of the players.

1.3 Functional Requirements

- Getdetails – Retreive the details entered by the user
- Getdata – Send the query to the database to get data of requested players
- Displaydata – Display the requested data in an easy to read form.

2. User authentication

2.1 Description and Priority

The admins will be required to login to their account before making updates. The users who wish to give feedback should be logged in before doing so.

2.2 Stimulus/Response Sequences

Stimulus: Admin visits the website.
Response: Admin is validated and redirected to a page that has functions which that particular user has access to.

Stimulus: User wants to provide feedback.
Response: User is validated and redirected to a page that has functions which that particular user has access to.

2.3 Functional Requirements

- Admin/user enters_credentials – The admin or the user provide their account details for authentication.
- Admin/user credentials_authenticate – The credentials provided is authenticated against those stored on the server.

Admin/user credentials_redirect – The admin or the user will be redirected to a page that has functions which that particular user has access to after successful login.

3. Statistics Updation

3.1 Description and Priority

The admin will be provided with function to update statistics, which the admin performs on a regular basis to maintain correctness of statistics.

3.2 Stimulus/Response Sequences

Stimulus: Admin enters the updated statistics.

Response: Success or failure reflecting the updation of statistics.

3.3 Functional Requirements

- Getdetails – Retreive the details entered by the admin
- Updatedata – Send the query to the database to update the data of requested players
- Is_data_updated– Display whether the data has been updated or not.

10.5.5 Other Non-functional Requirements

1. Performance Requirements

● The product shall be based on web and has to be run from a web server.

● The product shall take initial load time depending on internet connection strength which also depends on the media from which the product is run.

● The performance shall depend upon hardware components of the client/customer.

2. Safety Requirements

This is a software product, a website. There are no safety requirements as such. There can be no physical harm caused by the usage of the application.

3. Security Requirements

3.1 Data Transfer:

● The system shall use secure sockets in all transactions that include any confidential customer information.

● The system shall automatically log out all users after a period of inactivity.

● The system shall confirm all transactions with the customer's web browser.

● The system shall not leave any cookies on the customer's computer containing any of the user's confidential information.

3.2 Data Storage:

● The user's web browser shall never display a user's password. It shall always be echoed with special characters representing typed characters.

● The system's back-end servers shall never display a customer's password. The customer's password may be reset but never shown.

● The system's back-end servers shall only be accessible to authenticated administrators.

● The system's back-end databases shall be encrypted.

4. Software Quality Attributes

1. Adaptability: The website is adaptable to various devices. It can be viewed in a user-friendly manner on PCs, laptops, tablets and mobile.

2. Correctness: The data displayed in the application is correct to within a certain delay required to update the data when data is changed (delay would be the time between match being played and admin updating data).
3. Flexibility: The software has immense potential, and other features can be incorporated easily into the existing one.
4. Maintainability: The application is accurately documented, so maintaining it is not a problem.
5. Reliability: The data shown is accurate, so it is reliable to a great extent.
6. Testability: Unit testing is simple with this application.
7. Usability: The statistics of the cricketers are simple enough to interpret.

5. Business Rules

5.1 Administrator: The administrator has various elevated rights such as accessing the database with read as well as write access. The administrator can manually add the users who wish to become an admin.

5.2 User: The user can login into the system and view various statistics, provide feedback.

The user need not login just for viewing statistics.

10.5.6 Other Requirements

Server Requirements

A server and a domain will be required where we will host this website and a server is required to view statistics.

CHAPTER 11
DESIGN MODELLING

11. Design Modelling

The software development phase consists of design, coding and testing. The development phase takes 75% or more of the total software development cost. The aim of the design activity is to transform the requirements specified in the SRS (Software Requirements Specification) document in to a format that can be easily implemented using a suitable programming language or any implementation tool.

11.1 Data Design

Data design is the first and most important design activity. Here the main issue is to select the appropriate data structure. That is the Data design focuses on the definition of data structures.

The data design describes structures that reside within the software. Attributes and relationships between data objects dictate the choice of data structures.

PRIMARY DATA: For this project, it is regular cricket matches

SECONDARY DATA: It is obtained using different websites, blogs and handbooks.

User Table:

Attribute	Data type	Constraint	Description
Name	Varchar	Not null	User name
Email_id	Varchar	Not null	Email Id of user
User_id	Varchar	Primary Key	Id of user
Password	Varchar	Not null	Users password
is_admin	tinyint	Not null	User is admin or not

Bowling Table:

Attribute	Data type	Constraint	Description
Player	Varchar	Primary Key	Player name
Span	Varchar	Not null	Carrier time
Mat	int	Not null	Total matches
Inns	Int	Not null	Total innings played
Balls	Int	NULL	Total balls bowled
Runs	Int	NULL	Total runs conceeded
Wkts	Int	NULL	Total wickets
Ave	Decimal	NULL	Average
Econ	Decimal	NULL	Economy
SR	Decimal	NULL	Strike rate
4_For	Int	Not Null	No of times 4 wickets taken in match
5_For	int	Not Null	No of times 5 wickets taken in match

Batting Table:

Attribute	Data type	Constraint	Description
Player	Varchar	Primary Key	Player name
Span	Varchar	Not Null	Carrier
Mat	Int	Not Null	Total matches
Inns	Int	Not Null	Total innings played
No	Int	Not Null	No. of times not out
Runs	Int	NULL	Total runs in carrier
HS	Varchar	NULL	Highest score
Ave	Decimal	NULL	Average
BF	Int	NULL	Ball Faced
SR	Decimal	NULL	Strike rate
100s	Int	Not Null	No. of 100s
50s	Int	Not Null	No. of 50s
0s	int	Not Null	No. of ducks

11.2 Architectural Design

The architecture design uses information flowing characteristics, and maps them into the program structure.

Transform Mapping:

Transform mapping is a set of design steps that allows a DFD with transform flow characteristics to be mapped into a specific architectural style.

STEPS:

1. Review the fundamental system model.
2. Review and refine data flow diagrams for the software.
3. Determine whether the DFD has transform or transaction flow characteristics.
4. Isolate the transform center by specifying incoming and outgoing flow boundaries.
5. Perform "first-level factoring."
6. Perform "second-level factoring."
7. Refine the first-iteration architecture using design heuristics for improved software quality.

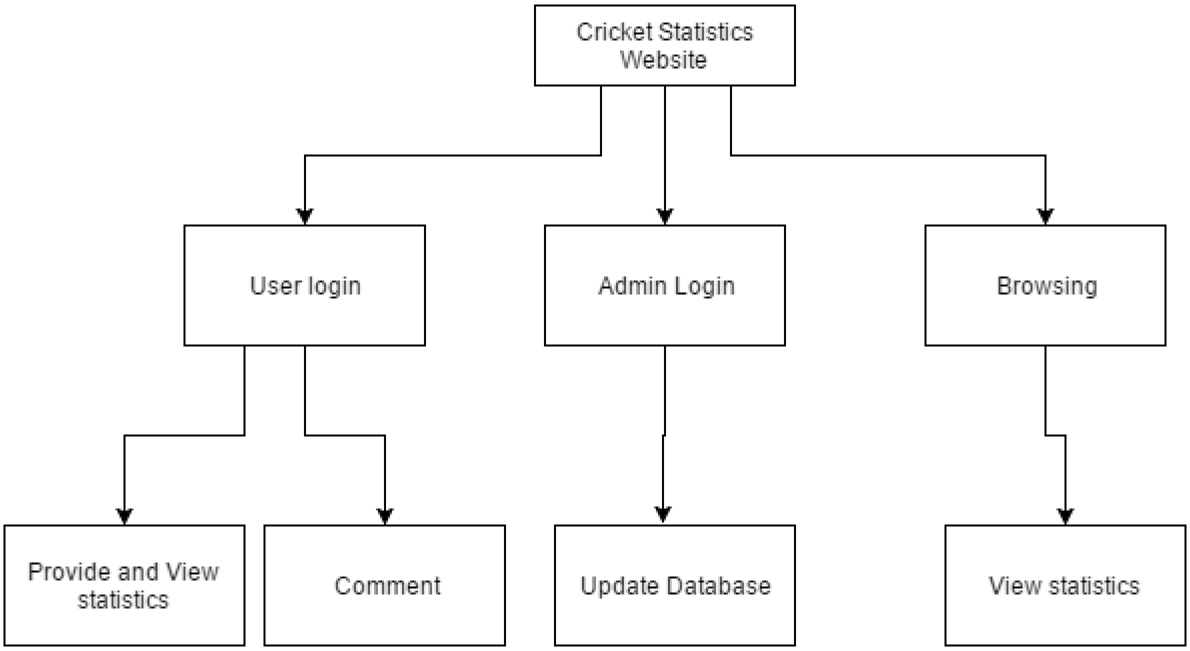
Transaction Mapping:

In many software applications, a single data item triggers one or a number of information flows that effect a function implied by the triggering data item. The data item is called a transaction.

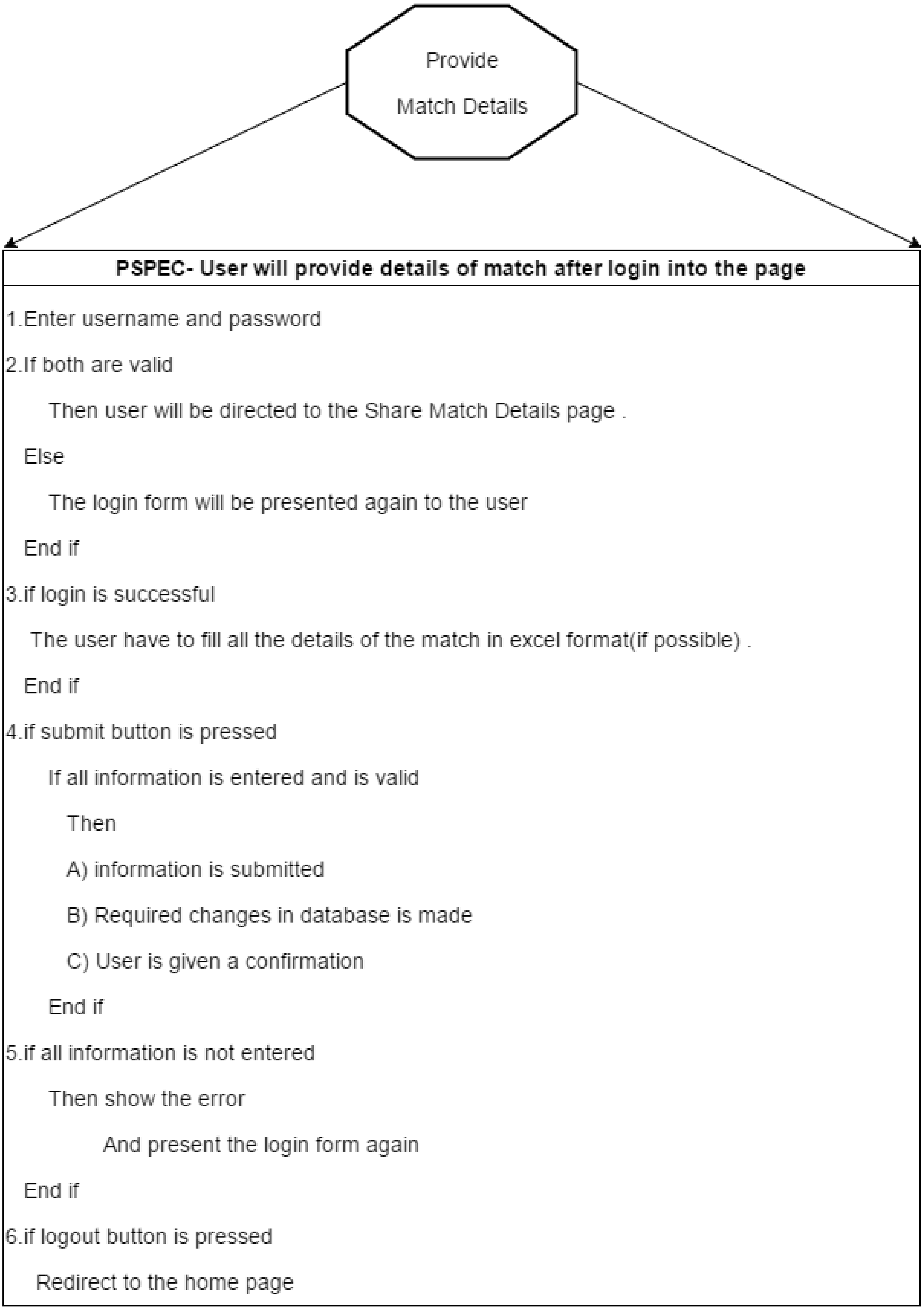
STEPS:

1. Review the fundamental system model.
2. Review and refine data flow diagrams for the software.
3. Determine whether the DFD has transform or transaction flow characteristics.
4. Identify the transaction center and the flow characteristics along each of the action paths.
5. Map the DFD in a program structure amenable to transaction processing.
6. Factor and refine the transaction structure and the structure of each action path.
7. Refine the first-iteration architecture using design heuristics for improved software

quality.

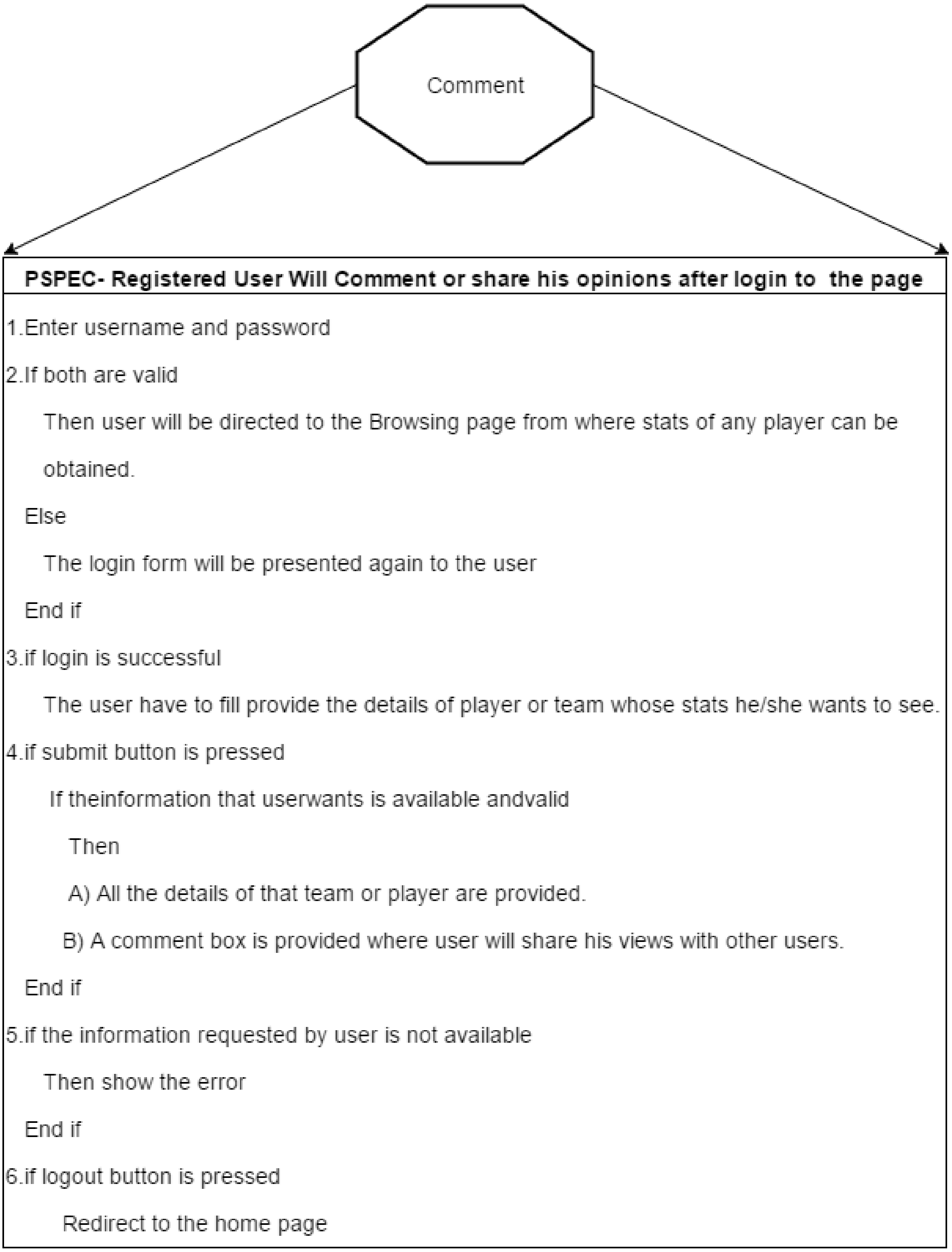

Cricket Statistics Website
User login
Admin Login
Browsing
Provide and View statistics
Comment
Update Database
View statistics

Process Specifications:



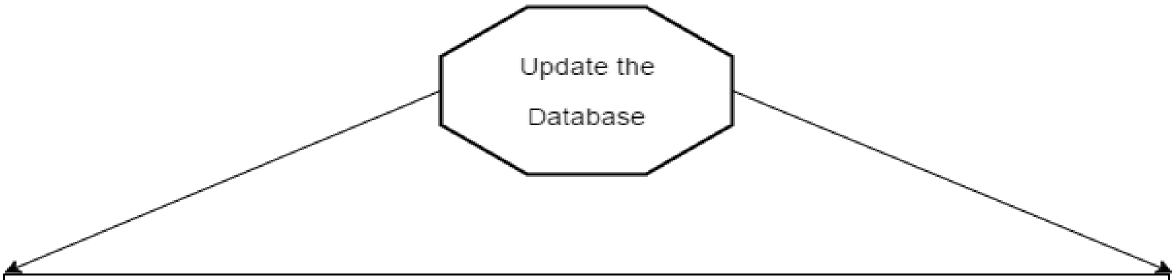
PSPEC- User will provide details of match after login into the page

```
1.Enter username and password
2.If both are valid
     Then user will be directed to the Share Match Details page .
  Else
     The login form will be presented again to the user
  End if
3.if login is successful
   The user have to fill all the details of the match in excel format(if possible) .
  End if
4.if submit button is pressed
     If all information is entered and is valid
       Then
       A) information is submitted
       B) Required changes in database is made
       C) User is given a confirmation
     End if
5.if all information is not entered
     Then show the error
            And present the login form again
  End if
6.if logout button is pressed
    Redirect to the home page
```



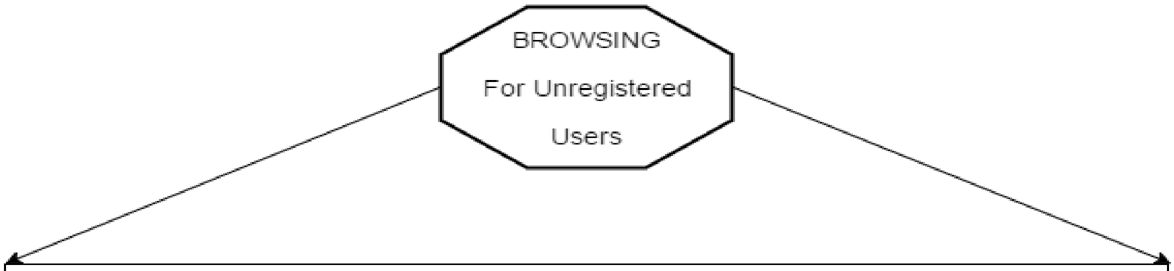
PSPEC- Registered User Will Comment or share his opinions after login to the page

```
1.Enter username and password
2.If both are valid
    Then user will be directed to the Browsing page from where stats of any player can be
    obtained.
  Else
    The login form will be presented again to the user
  End if
3.if login is successful
    The user have to fill provide the details of player or team whose stats he/she wants to see.
4.if submit button is pressed
     If theinformation that userwants is available andvalid
       Then
       A) All the details of that team or player are provided.
      B) A comment box is provided where user will share his views with other users.
  End if
5.if the information requested by user is not available
    Then show the error
  End if
6.if logout button is pressed
      Redirect to the home page
```



PSPEC-Admin will have to login to update the database.

```
1. Enter username and password of admin
2. If both are valid
      Then admin will be directed to the page that has all Cricket Stats details
   Else
      The login form will be presented again to the user
   End if else
3. If login is successful
      The admin updates the required information (stats)in the page with all the details.
   End if
4. If logout button is pressed
       Redirect admin to the home page.
```



PSPEC-Unregistered user will obtain the information byclicking on links provided.

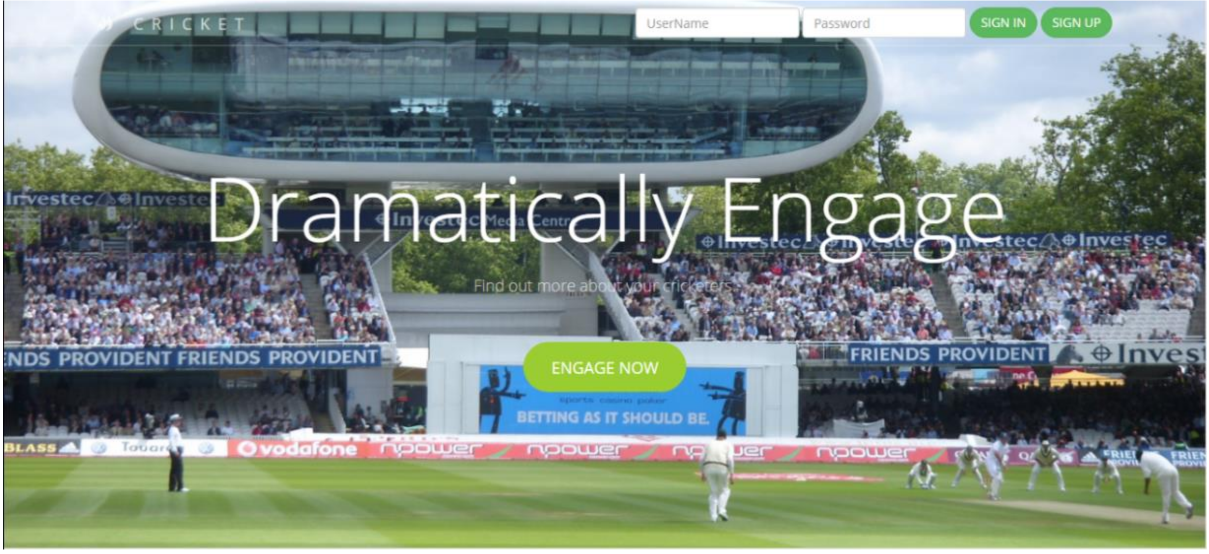
```
1. User will have to click on Find Button which links them to browsing page.
2.The user have to fill provide the details of player or team whose stats he/she wants to see.
3.If submit button is pressed
     If the information that user wants is available and valid
        Then
        All the details of that team or player are provided.
     End if
4.If the information requested by user is not available
     Then show the error
      Redirect to the browsing page
  End if
```

11. 3 User Interface Design

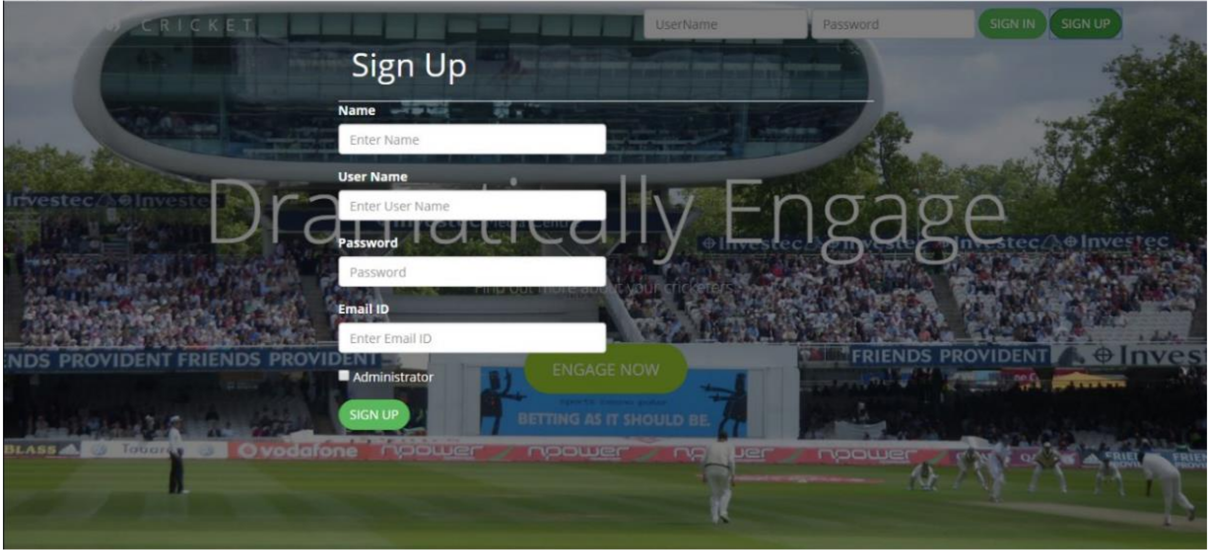
User interface design (UID) or user interface engineering is the design of websites, computers, appliances, machines, mobile communication devices, and software applications with the focus on the user's experience and interaction. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals—what is often called user-centred design.

Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to itself. Graphic design may be utilized to support its usability, influencing how the user performs certain interactions and improving the aesthetic appeal of the design; design aesthetics may enhance or detract from the ability of users to use the functions of the interface.

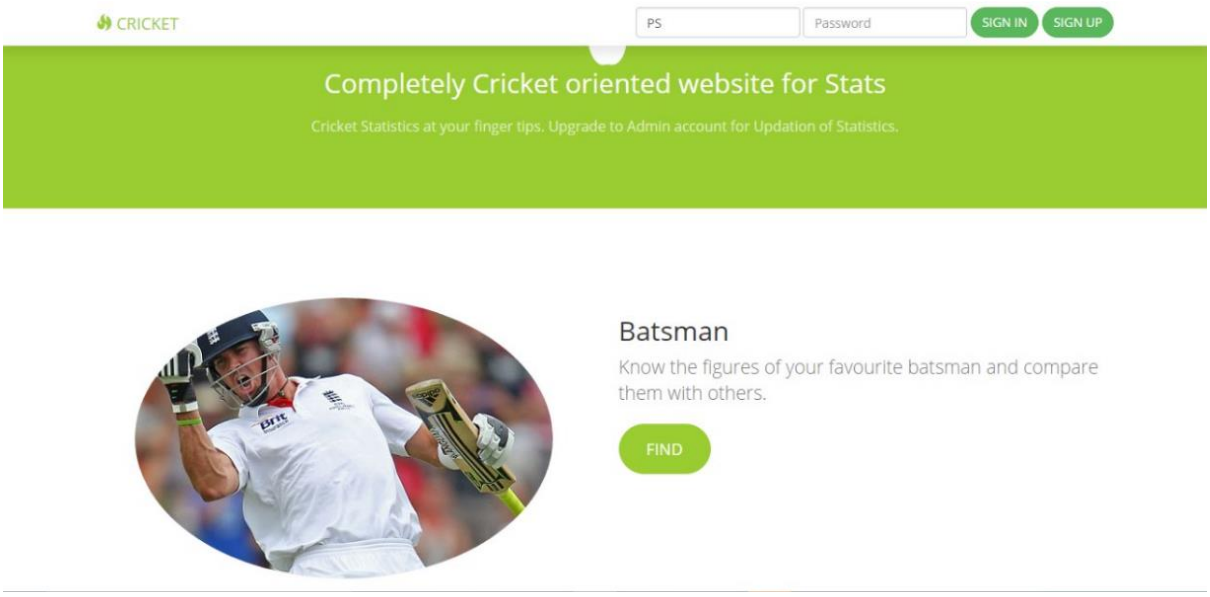
Home Page (1)



Signup Page



Batsman Page



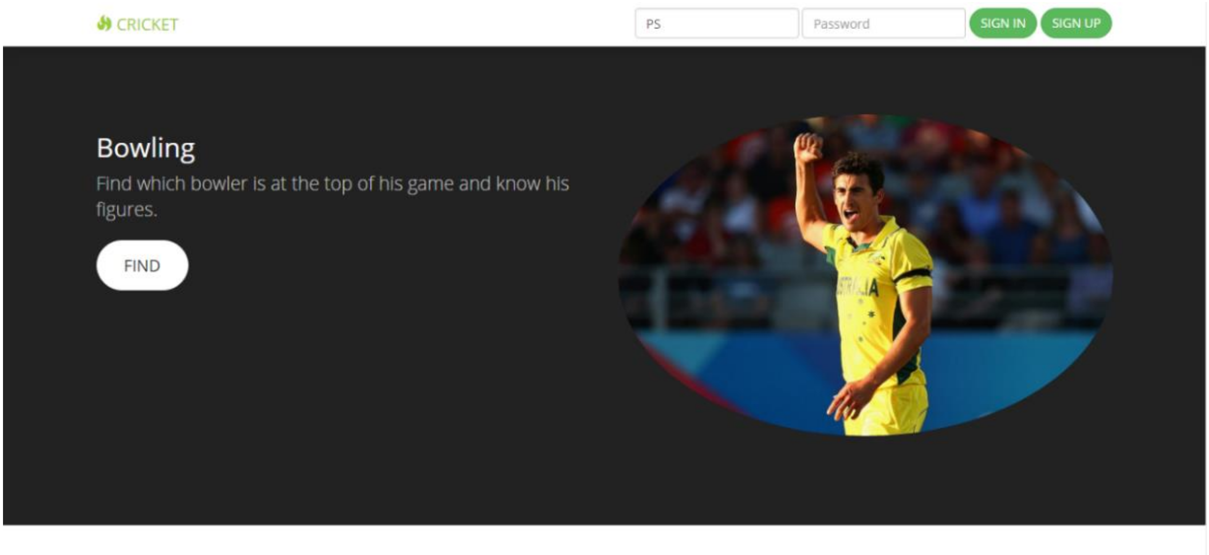
Completely Cricket oriented website for Stats

Cricket Statistics at your finger tips. Upgrade to Admin account for Updation of Statistics.

Batsman

Know the figures of your favourite batsman and compare them with others.

Bowler Page:

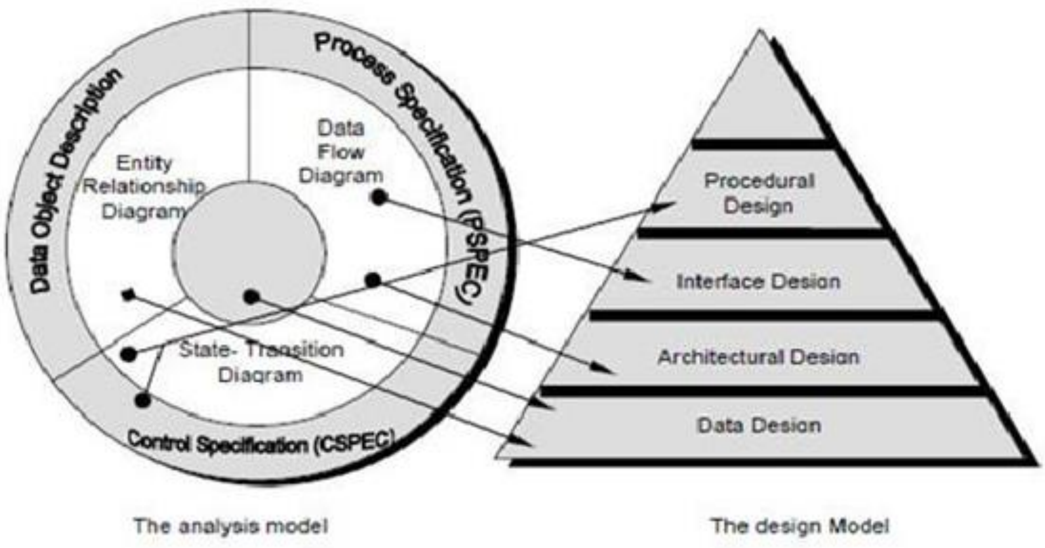


Bowling

Find which bowler is at the top of his game and know his figures.

11.4 Procedural Design

The procedural design describes structured programming concepts using graphical, tabular and textual notations. These design mediums enable the designer to represent tabular and textual notations. These design mediums enable the designer to represent forms the basis for all subsequent software engineering worked.



CHAPTER 12

SOFTWARE TESTING

12. SOFTWARE TESTING

Testing should be done through the implementation process. Even before and application is installed; it makes sense to verify that the basic platform is capable of achieving its design capabilities. System testing is a critical process. Testing is a process of executing a program with the explicit intention of finding errors that is making the program fail. This helps in finding the bottle neck in the system. Executing a program in a stimulated environment performs testing. The feedback from testing phase generally produces changes in the software to deal with errors and failures that are uncovered.

The objectives of testing are as follows:
1. Testing is a process of executing a program with the intent of finding an error.
2. A good test case is one that has a high probability of finding an as-yet-undiscovered error.
3. A successful test is one that uncovers an as-yet-undiscovered error.

12.1 Test Case Design:

A rich variety of test case design methods have evolved for software. These methods provide the developer with a systematic approach to testing. More important, methods provide a mechanism that can help to ensure the completeness of tests and provide the highest likelihood for uncovering errors in software.

Black box testing:

When computer software is considered, black-box testing alludes to tests that are conducted at the software interface. In black box testing or functional testing test cases are decided. Test cases are decided on the basis of requirements or specifications of the program or module. A black-box test examines some fundamental aspect of a system with little regard for the internal logical structure of the software.
Black box testing is done in the project to remove these errors:

- Incorrect or missing function
- Interface errors.
- Errors in data structure or external database access.
- Behavioural or performance error.
- Errors in initiation and termination.

White box testing:

The White box testing or structural testing performs close operation of procedural details. They test the software logical path by having test cases exercising specific sets of condition and loops. The "status of the program" may be examined at various points to determine if the expected or asserted status corresponds to the actual status.
White box testing is done in the project to remove the errors:

- All modules path have been exercised at least once.
- Exercised on logical decisions.
- Executed all loops at their boundaries and within their operational bounds.

- Exercised internal data structure to ensure their validity.

The attributes of both black- and white-box testing can be combined to provide an approach that validates the software interface and selectively ensures that the internal workings of the software are correct.

12.2 Unit Testing:

Unit testing focuses verification effort on the smallest unit of software design—the software component or module. Using the component-level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered errors is limited by the constrained scope established for unit testing. The unit test is white-box oriented, and the step can be conducted in parallel for multiple components.

Unit testing is normally considered as an adjunct to the coding step. After source level code has been developed, reviewed, and verified for correspondence to component-level design, unit test case design begins. A review of design information provides guidance for establishing test cases that are likely to uncover errors in each of the categories discussed earlier. Each test case should be coupled with a set of expected results.

Unit testing is simplified when a component with high cohesion is designed. When only one function is addressed by a component, the number of test cases is reduced and errors can be more easily predicted and uncovered.

Following test were performed during unit testing:

- **Module Interface Test**: Module interface was tested to ensure information flow in and out of the program unit.
- **Local Data Structure Testing**: Local data structures were tested to make surety that data store temporarily maintained their integrity during all steps in algorithm execution.
- **Boundary Condition Testing**: Boundary conditions were tested to make sure that the modules operate properly at boundaries.
- **Independent Path Testing**: All independent paths through control structure were checked to make sure that all statements in a module has been executed.
- **Error Handling Path Test**: This is performed to handle exception.

12.3 Integration testing:

This testing is done to tackle problems of interface that is putting all interfaces together. When the separate modules are put together in an integrated manner, this testing is performed. This testing is systematic technique. This testing is performed to check the data should not be lost across an interface. The objective is to take a unit tested module and build a program structure that has been dedicated by design.

Top-Down Integration:

Top-down integration testing is an incremental approach to construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main control module (main program). Modules subordinate (and ultimately subordinate) to the main control module are incorporated into the structure in either a depth-first or breadth-first manner.

Bottom-Up Integration:

Bottom-up integration testing, as its name implies, begins construction and testing with atomic modules (i.e., components at the lowest levels in the program structure). Because components are integrated from the bottom up, processing required for components subordinate to a given level is always available and the need for stubs is eliminated.

Regression Testing:

Regression was done to ensure proper working of each module with the whole system. Each module is embedded in the system and the whole tested for integrity.

Smoke Testing:

Smoke Testing is a testing technique that is inspired from hardware testing, which checks for the smoke from the hardware components once the hardware's power is switched on. Similarly in Software testing context, smoke testing refers to testing the basic functionality of the build.

If the Test fails, build is declared as unstable and it is NOT tested anymore until the smoke test of the build passes.

12.4 Validation Testing:

At the culmination of integration testing, software is completely assembled as a package, interfacing errors have been uncovered and corrected, and a final series of software tests—validation testing—may begin. Validation can be defined in many ways, but a simple definition is that validation succeeds when software functions in a manner that can be reasonably expected by the customer.

An important element of the validation process is a configuration review. The intent of the review is to ensure that all elements of the software configuration have been properly developed, are catalogued, and have the necessary detail to bolster the support phase of the software life cycle.

Alpha and Beta Testing:

The alpha test is conducted at the developer's site by a customer. The software is used in a natural setting with the developer "looking over the shoulder" of the user and recording errors and usage problems. Alpha tests are conducted in a controlled environment.

The beta test is conducted at one or more customer sites by the end-user of the software. Unlike alpha testing, the developer is generally not present. Therefore, the beta test is a "live" application of the software in an environment that cannot be controlled by the developer. The customer records all problems (real or imagined) that are encountered during beta testing and reports these to the developer at regular intervals. As a result of problems reported during beta tests, software engineers make modifications and then prepare for release of the software product to the entire customer base.

CHAPTER 13
CONCLUSION
AND
FUTURE SCOPE

13. Conclusion and Future Scope

Cricket Database Systems are meant for instant updates on cricketers so that the followers of the game can rely on statistics those are recent as well as accurate. Every cricket fan streams through statistics of different players and compares them and accuracy is something which is expected so that the user is not disappointed. A Cricket database system should allow a user to view the details, comment about them and if necessary apply updates to them. A good system makes up for the lack of update feature that needs to be present.

Cricket fans usually wish to see their favourite batsman's Runs, Strike Rate, Balls Faced and other such statistics and for their favourite bowler they check no of balls, wickets, economy, average etc. A user of the system wishes to see accurate data and if not can make changes himself, such an option is provided only to the administrator so that there are false changes.

In an existing system,

The Cricket databases are updated by a single owner of the website and can take more time if busy, the current system also has to enter every single field to update which increases the dependency of input of data. Existing systems do not allow users to update the data or sometimes there is no provision for them to even comment on the stats. The advantages are listed below:

Data Delay: Due to a single user update it takes long time to get updates which are rather expected in short time.

Inaccuracy: Due to updates by a single user, wrong data often goes unverified and there is no way to get it updated properly by any other user.

User Comments: Users do not get the chance to comment on the statistics and are often left unattended.

Data Dependency: Usually user has to update all fields and the dependent fields don't get updated automatically.

While in the proposed system,

Users get a variety of opportunities to use the website by commenting on stats. If users get access to matches then they can also update by subscribing to the admin account. Users can also make changes to the updates if they find inaccuracy.

Time Effective: Due to multiple admins, it is easy to get information on the website in short intervals of time.

Accuracy: In case inaccurate data is found it can be corrected by anybody who uses the admin account.

Data Dependency: Only few fields have to be filled, the rest can be inferred or calculated from the mandatory fields.

User Reviews: Users are given the opportunity to make comments on the statistics.

Future Scope:

- The proposed system can further use a ball by ball commentary which engages a user in a more innovative web service for cricket database management.
- The website could also allow a user to mark their matches as favourite or save commentary for particular matches.
- The user will be able to share the statistics with other users thereby providing a communication between the users of the website.
- Users should be able to contest polls in between the matches to compare the players of the match.
- More segregation of data by comparing performances of cricketers in particular stadium or conditions.
- Rating system to show the performance of player in the ongoing match.

CHAPTER 14
REFERENCES

14. References

1. Roger S. Pressman, “SOFTWARE ENGINEERING – A PRACTITIONERS APPROACH”, 5th Edition.
2. IEEE Software Engineering Standards Committee, “IEEE Std 830 - 1999, IEEE Recommended Practice for Software Requirements Specifications by Karl E. Wiegers”.
3. Website For making flowcharts and diagrams www.draw.io
4. Test Documentation www.chambers.com.au/glossary/**test_documentation**.php
5. Cricket Statistics stats.espncricinfo.com