CRICKET DATA MANAGEMENT SYSTEM

Under the guidance of
Prof. P. M. Chawan

Submitted by
Mr. Aayush Shah
Mr. Meet Parekh
Mr. Nirmit Deliwala
Mr. Mann Doshi
Mr. Donovan Crasta
B. Tech IT (Sem V)

Veermata Jijabai Technological Institute, Matunga, Mumbai - 400019.

Certificate

Statement By Candidates

We wish to state that the work embodied in this specification titled ‘Cricket Data Management System’ forms our own contribution to the work carried out under the guidance of Prof. P. M. Chawan at Veermata Jijabai Technological Institute, Matunga, Mumbai - 19. This work has not been submitted for any other Degree or Diploma of any Institute/University. Wherever references have been made to previous works of others, they have been clearly stated.
GitHub: https://github.com/Aayushjshah/cricit

TABLE OF CONTENTS

PROBLEM STATEMENT

1.1 Problem Statement

Design a software that implements a cricket data management system.

1.2 Problem Statement Description

Match officials input the match details like teams playing, squads, venue. The software will take live ball to ball commentary as input and the software will generate a scoresheet of each match.

The name of the tournament could be entered along with the full schedule, venue and team details. While entering the details of the match,the software will update the match details table for the individual players.

The software will also manage the database containing profiles and records of players of each ICC cricketing nation. A user can view all the statistics of a player in each format. The data will then be segregated and stored under the respective players' column in the database (as his own record). Thus the player's scorecard is updated after each match. The scorecard will contain the tournament wise summary of runs scored, wickets taken, catches caught (for that tournament).

Processing the different stats of data, providing valuable insights in different domains of the game which will help enhance the performance of the player and will also provide recommendations of players for different formats.

SCOPE OF PROJECT

2.1 Deliverables

Ball-to-Ball Data Entry
Ball-to-ball scores could be entered manually into the software.

Scorecard generation and update database
Ball-to-ball commentary gets converted into a scorecard which goes under the player's stats at the end of the match.

Tournament/Match Specific Stats
Details of specific matches can be fetched. A leaderboard can be generated for the tournament.

Generate statistics of a particular player
Users can view any player's stats and his achievements. (Match-wise or tournament-wise)

Head-to-Head Statistics Comparison
Considering that we have the final scorecard with us, we could generate stats for a player versus another player (for eg. a batsman versus a bowler).

Notes Section for Coaches
Coaches could make their own notes player wise or match wise which could help enhance a player's performance.

2.2 Features

Add and manage tournaments
Users could enter details of a tournament such as the number of matches to be played, the details of each match such as the teams playing it and the venue, the playing eleven (and the subs) for each of those matches.

Enter ball-to-ball details for a match
Users could keep on updating the scorecard after every ball and at the end of each match, a scorecard is generated. This scorecard could then be linked to the respective players' profiles and their details will be updated.

Player vs player

Users will be treated with head-to-head statistics for better comparisons. This will be a combination of different types of head-to-head comparisons. For example : Batsman A to Batsman B comparison vs particular teams,
Or Batsman A vs Bowler B stats i.e. runs scored, average vs that bowler, number of times bowled out, etc.

Generate statistics of a particular player

The ball-by-ball score entry updated by the user will be stored and processed to filter out stats of players individually as well as for the team. Also, these statistics of each player will be used to compare the best players in each tournament.

Different type of access for different logins

There will be different types of accounts for different types of people:- Tournament organisation committee (such as BCCI), coaches, scorekeepers and general users. Tournament organisation committee login will be able to start a tournament, add teams in it and host matches, assign coaches, assign scorekeepers etc. Coaches will be able to add players in the team, decide the playing XI, etc. Scorekeepers' login will be able to add ball-by-ball match details. General users will only be able to view the data. Abstraction of data will also be done.

2.3 Scope

This software could primarily be used by individuals who want to conduct and manage their own cricketing tournaments (say, BCCI for instance) for managing their data. It could range all the way from managing a small inter-society cricket tournament to a king-sized one, like the IPL.

The player's head-to-head statistics could help team managers pick the right players for their team. This helps them manage their expenses in the event of an auction.
Not directly, but this software would also help team managers pick the right XI and substitutes for a tour. (We have the player's head to head stats against all the bowlers he has faced). This, however, has to be done manually.

A custom tournament could be made by a tournament organizer which could only be accessed by them. This is abstracted from national level match officials. The vice-versa, however (district level organizer accessing national level tournaments' stats) is permitted.

2.4 Limitations

Field placements and dynamic field changes during the match.

Player's fitness and health analysis. Hence, workload is not effectively managed.

Although the software has the head-to-head stats, a playing XI cannot be recommended for a match. This is because, for that we'd have to look for the head-to-head stats for each of the players from the team against the other. This would require a lot of searching in the database which will require lots of complex queries. The picking has to be done manually.

Distance of sixes, timing, accuracy, ball speed are not kept track of. Hence, for a tournament, we don't get the longest six hitter or fastest ball bowled.

Live win probability cannot be generated. This is because our software requires manual insertion of data and for calculating the live win, we require a complex model that predicts the future outcomes by running the current status in a simulator. This requires a lot of data, a complex model and a huge computational ability.

2.5 Performance Measure

Comparing our software with other cricketing softwares such as cricbuzz, ESPN live, etc. we could say that our software has all the features plus some additional features that are exclusive to our application. Features such as organizing custom tournaments/matches, coaches' notes, head-to-head comparisons that help tournament organizers effectively manage their data.

RESOURCE ESTIMATION

3.1 Human Resources

Their work includes researching, designing, developing, and testing software. A software developer may take part in design, computer programming or software project Management. This project consists of a group of 5 members (Nirmit Deliwala, Mann Doshi, Meet Parekh, Aayush J Shah, Donovan Crasta).

3.2 Estimated Timeframe

The project includes making a desktop application for managing and analysis of cricket data. It includes creating a frontend and a backend. Some skills required for this project need to be learnt which will take some time which will take 3-4 weeks of time.
After having learnt the languages/softwares 4-5 weeks of time will be needed to complete the frontend, backend, creating/designing databases, designing appropriate SQL queries for tasks like displaying player stats, detailed analysis, etc.
After having built the first version of the application, it needs to be checked for any bugs/shortcomings. The application needs to be tested on dummy datasets. This time is to be dedicated to finding errors/bugs and addressing them.

3.3 System Requirements

Development Environment: The success of the project also depends upon the quality and availability of the tools which are being used to develop the software. The development requirements include:

(a) Hardware Components

- Laptop
- Internet Connection

(b) Software Components

- Operating System
  - ➔ Windows
  - ➔ Ubuntu
- MySQL Database
- IDE / Text Editors such as VSCode/Sublime Text
- PDF Viewers

(c) Reusable Software

- Front-end layouts used for login/signup
- Some of JavaAPIs such as:
  - ➔ Java.awt
  - ➔ Java's Socket API
  - ➔ Java's Networking API

EFFORT ESTIMATION

4.1 Problem Based Estimation

Lines of code and function points were described as measures from which productivity metrics can be computed. LOC and FP data are used in two ways during software project estimation:

(1) as an estimation variable to "size" each element of the software and

(2) As baseline metrics collected from past projects and used in conjunction with estimation variables to develop cost and effort projections.

LOC and FP estimation are distinct estimation techniques. Yet both have a number of characteristics in common. 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. Alternatively, the planner may choose another component for sizing such as classes or objects, changes, or business EMPIRICAL MODEL A typical Empirical model is derived using regression analysis on data collected from past software projects.

4.1.1 Lines of Code(LOC) Estimation

LOC / Lines of Code, usually referring to non/commentary lines, meaning pure whitespace and lines containing only comments are not included in the metric.

The number of lines of program code is a wonderful metric. It's so easy to measure and almost impossible to interpret.

There are many different ways to count lines (e.g., with or without comments, counting statements rather than lines, or counting lines in an automatically formatted code)

Function	Optimistic	Most Likely	Pessimistic	Estimate LOC
User Interface	300	500	800	517
Input Commentary	400	600	800	600
Scorecard Development	400	600	800	600
Tournament and Player Statistics	600	800	1200	833
Database management	300	500	700	500
Estimated Lines of Code				**3050**

LOC = (Optimistic + 4*Most Likely + Pessimistic)/6

LOC of User Interface = (300 + 4*500 + 800)/6 = 517

LOC of Input Commentary = (400 + 4*600 + 800)/6 = 600

LOC of Scorecard Development = (400 + 4*600 + 800)/6 = 600

LOC of Tournament and Player Stats = (600 + 4*800 + 1200)/6 = 833

LOC of Database Management = (300 + 4*500 + 700)/6 = 500

Total LOC = 3500

Considering 1000 lines of code per month, the project requires approximately 3 months.

Considering cost per LOC to be Rs.10

Cost of Project = 3500*10 = **Rs. 35000**

4.1.2 Estimation technique FP base (function point based estimation)

FP = count total × [0.65 + 0.01 × (ΣFi)]

Event Management System FP base Calculations :

Measurement Parameter	Count	Simple	Average	Complex	Function Points
Number of user input	10	2	4	12	40
Number of user output	8	4	5	7	56
Number of user inquiries	0	0	4	6	0
Number of files	3	7	10	15	30
Number of external interfaces	1	5	7	10	5
Count total					**131**

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	Ratings
Backup and recovery	4
Data communications	3
Distributed processing	2
Performance critical	5
Existing operating environment	5
On-line data entry	5
Input transaction over multiple screens	2
Master files updated on-line	5
Information domain values complex	2
Internal processing complex	4
Code designed for reuse	3
Conversion/installation in design	1
Multiple installations	1
Application designed for change	3

Total Complexity Adjustment factors(Fi's) = 45

Total of FP's = 131

FP = Count Total * (0.65 + 0.01*Fi's)

= 131 * (0.65 + 0.01*45)

= 144

Systems of this type is 50 FP/PM

Considering Labour rate Rs.10000/PM

Cost per FP is approximately Rs200

Project Cost is 131 * 200 = **Rs. 26200**

RISK ANALYSIS AND MANAGEMENT

5.1 Risk Identification

Product Size Risks

- **Estimated size in lines of code (LOC)**
  CricInfo will have an estimated 3,050 lines of code.

- **Degree of confidence in estimated size**
  We are highly confident in our estimated size.

- **Estimated size in number of programs, files, and transactions**
  1. We estimate 10 large files for the engine, 5 large files for the user interface.
  2. We estimate 40 or more transactions for the engine, and 20 transactions for the user-interface.

- **Percentage deviation in size from average for previous products** We allow for a 20% deviation from average.

- **Size of database created or used**
  The size of the database that we will use will be an estimated 5 tables. The number of fields will vary per table and will have an overall average of 10 fields per table. The number of records in each table will vary with the number of matches/players that the user adds to the database, and the number of instances of each tournament that the user creates.

- **Number of users**
  The number of users will be fairly low for now. There will be multiple users per instance of the software running, as the software is intended for multi-user use.

- **Number of projected changes to the requirements**
  We estimate 3 possible projected changes to the requirements. These will be as a result of our realization of what is required and not required as we get further into implementation, as well as a result of interaction with the mentor and verification of the mentor's requirements.

- **Amount of reuse of software**
  Reuse will be very important to get the project started. Java's Swing API, Socket API will be among the many other software tools that'd be used.

Process Risks

- **Does the mentor support a written policy statement that emphasizes the importance of a standard process for software development?**
  N/A. PA Software does not have a mentor. It should be noted that the structured method has been adopted for the CricInfo project. At the completion of the project, it will be determined if the software method is acceptable as a standard process, or if changes need to be implemented.

- **Has your team developed a written description of the software process to be used on this project?**
  Yes. CricInfo is under development using the structured method as described in part two of Roger S. Pressman's Software Engineering, A Practitioner's Approach.

- **Are team members willing to use the software process?**
  Yes. The software process was agreed upon before development work began.

- **Is the software process used for other products?**
  N/A. PA Software has no other projects currently.

- **Have documented outlines and examples been developed for all deliverables defined as part of the software process?**
  Yes. The course instructor has supplied outlines for all deliverables.

- **Is there some mechanism for ensuring that work conducted on a project conforms with software engineering standards?** No.
  There has been no planned method to ensure software-engineering standards will be met.

- **Is configuration management used to maintain consistency among system/software requirements, design, code and test cases?** Yes. The accompanying Software Configuration Management document outlines the plan for maintaining consistency among all technical documents in the CricInfo project.

- **Is a mechanism used for controlling changes to mentor requirements that impact software?**
  No. The mentor requirements for the CricInfo project are fairly flexible. Mentors have allowed a great deal of freedom to the project developers. This could become a problem if the mentor does change the requirements, by the likelihood of that is extremely low.

- **Is there a documented statement of work, a software requirements specification, and a software development plan for each subcontract?**
  N/A. All work is done by a single development team. No subcontracting will take place on the CricInfo project.

- **Is there a procedure followed for tracking and reviewing the performance of subcontractors?**
  N/A. All work is done by a single development team. No subcontracting will take place on the CricInfo project.

Technical Issues

- **Are facilitated application specification techniques used to aid in communication between the mentor and the developer?**
  The development team will hold frequent meetings directly with the mentor. No formal meetings are held (all informal). During these meetings the software is discussed and notes are taken for future review.

- **Are specific methods used for software analysis?**
  Special methods will be used to analyze the software's progress and quality. These are a series of tests and reviews to ensure the software is up to speed. For more information, see the Software Quality Assurance and Software Configuration Management documents.

- **Do you use a specific method for data and architectural design?**

Data and architectural design will be mostly object oriented. This allows for a higher degree of data encapsulation and modularity of code.

- **Is more than 90 percent of your code written in a high-order language?**
  Yes. Code will be written in a combination of Visual Basic, Java, and a whole lot of SQL.

- **Are specific conventions for code documentation defined and used?**
  No. Specific conventions have not been established, but all design members have agreed to comment code as completely as possible.

- **Are software tools used to support planning and tracking activities?**
  No. No software tools are to be used. Due to the existing deadline, the development team felt it would be more productive to begin

implementing the project than trying to learn new software tools. After the completion of the project software tools may be implemented for future projects.

- **Are configuration management software tools used to control and track change activity throughout the software process?** No. No software tools are to be used. Due to the existing deadline, the development team felt it would be more productive to begin implementing the project than trying to learn new software tools. After the completion of the project software tools may be implemented for future projects.

- **Are software tools used to support the software analysis and design process?**
  No. No software tools are to be used. Due to the existing deadline, the development team felt it would be more productive to begin implementing the project than trying to learn new software tools. After the completion of the project software tools may be implemented for future projects.

- **Are tools used to create software prototypes?**
  Yes. Prototypes are created using pencil and paper, as well as interface mock-ups using Microsoft Visual Basic.

- **Are software tools used to support the testing process?** No. No software tools are to be used. Due to the existing deadline, the development team felt it would be more productive to begin implementing the project than trying to learn new software tools. After the completion of the project software tools may be implemented for future projects.

- **Are software tools used to support the production and management of documentation?**
  Yes. Microsoft Word will be used to support the production and management of all technical documentation.

- **Are quality metrics collected for all software projects?** No. No plans have been made to collect quality metrics at this time.

- **Are productivity metrics collected for all software projects?** No. No plans have been made to collect productivity metrics at this time.

Technology Risks

- **Is the technology to be built new to your team?**
  CricInfo is a software tool to help manage cricketing data. Development team members are familiar with development, as well as the necessary database implementation.

- **Do the mentor's requirements demand the creation of new algorithms or input or output technology?**
  No. CricInfo will be implemented using existing algorithms. Input and output are handled in a traditional manner.

- **Does the software interface with new or unproven hardware?**
  Since CricInfo is taking advantage of Java's Swing and Socket APIs, the software will allow for a multitude of new or future hardware products.

- **Does the software to be built interface with vendor supplied software products that are unproven?**
  No. CricInfo interfaces with VSCode and MySQL, all of which are licensed, open-source, free-to-use software products.

- **Is a specialized user interface demanded by the product requirements?**
  Yes. The interface is completely specialized. It is not based on anything other than every other Microsoft Windows application out. The GUI is completely our design and no other application out (to our knowledge) contains exactly what is expected of our software.

- **Do requirements for the product demand the creation of program components that are unlike any previously developed by your team?**
  Yes. The entire GUI is composed of subsystems that our software engineer has never had experience with.

- **Do requirements demand the use of new analysis, design, or testing methods?**
  No. The development team will implement existing analysis, design, and testing methods for the project.

- **Do requirements demand the use of unconventional software development methods?**
  No. CricInfo uses Java code in the form of APIs, which is not unconventional. It also integrates with Visual Basic, which is not unconventional.

- **Do requirements put excessive performance constraints on the product?**
  Yes, since there will be multiple users accessing and querying at the same time the database has to be structured in an efficient manner.

The data i.e. the match scores need to be updated as fast as possible.

- **Is the mentor uncertain that the functionality required is "doable"?**
  No. The mentor has full confidence in the project as described in the System Specification Document and the Software Specification Document.

Development Environment Risks

- **Is a software project management tool available?**
  No. No software tools are to be used. Due to the existing deadline, the development team felt it would be more productive to begin implementing the project than trying to learn new software tools. After the completion of the project software tools may be implemented for future projects.

- **Are tools for analysis and design available?**
  No. No software tools are to be used. Due to the existing deadline, the development team felt it would be more productive to begin implementing the project than trying to learn new software tools. After the completion of the project software tools may be implemented for future projects.

- **Do analysis and design tools deliver methods that are appropriate for the product to be built?**
  N/A. No analysis or design tools are to be used.

- **Are compilers or code generators available and appropriate for the product to be built?**
  Yes. Microsoft Visual Studio Code will be used to build CricInfo.

- **Are testing tools available and appropriate for the product to be built?**
  No. No software tools are to be used.

- **Are software configuration management tools available?** No. No software tools are to be used.

- **Have members of the project team received training in each of the tools?**
  N/A. No software tools are to be used. Due to the existing deadline, the development team felt it would be more productive to begin

implementing the project than trying to learn new software tools. After the completion of the project software tools may be implemented for future projects.

- **Are local experts available to answer questions about the tools?** N/A. No software tools are to be used. Local experts will not be necessary.

- **Is on-line help and documentation for the tools adequate?** Yes, online help and documentation for the tools will be adequate.

5.2 Risk Projection

Risk Table

Risks	Category	Probability	Impact
Computer Crash	TI	70%	1
Late Delivery	BU	30%	1
Technology will not meet Expectations	TE	25%	1
End Users Resist System	BU	20%	1
Changes in Requirements	PS	20%	2
Lack of Development Experience	TI	20%	2
Lack of Database Stability	TI	40%	2
Poor Quality Documentation	BU	35%	2
Deviation from Software Engineering Standards	PI	10%	3
Poor Comments in Code	TI	20%	4

Impact Values:

1 – Catastrophic
2 – Critical
3 – Marginal
4 – Negligible

Risk Refinement

At various points in the checklist, lack of software tools is identified as a potential risk. Due to time constraints, the members of the design team felt that searching for and learning to use additional software tools could be detrimental to the project, as it would take time away from project development. For this reason, we have decided to forgo the use of software tools. It will not be explored as a potential risk because all planning will be done without considering their use.

5.3 Risk Mitigation, Monitoring and Management

Risk: Computer Crash

- **Mitigation**

The cost associated with a computer crash resulting in a loss of data is crucial. A computer crash itself is not crucial, but rather the loss of data. A loss of data will result in not being able to deliver the product to the mentor. This will result in not receiving a letter of acceptance from the mentor. Without the letter of acceptance, the group will receive a failing grade for the course. As a result the team is taking steps to make multiple backup copies of the software in development and all documentation associated with it, in multiple locations.

- **Monitoring**

When working on the product or documentation, the project member should always be aware of the stability of the computing environment they're working in. Any changes in the stability of the environment should be recognized and taken seriously.

- **Management**

The lack of a stable-computing environment is extremely hazardous to a software development team. In the event that the computing environment is found unstable, the development team should cease work on that system until the environment is made stable again, or should move to a system that is stable and continue working there.

Risk: Late Delivery

- **Mitigation**

The cost associated with a late delivery is critical. A late delivery will result in a late delivery of a letter of acceptance from the mentor. Without the letter of acceptance, the group will receive a failing grade for the course. Steps have been taken to ensure a timely delivery by gauging the scope of the project based on the delivery deadline.

- **Monitoring**

  A schedule has been established to monitor project status. Falling behind schedule would indicate a potential for late delivery. The schedule will be followed closely during all development stages.

- **Management**

  Late delivery would be a catastrophic failure in the project development. If the project cannot be delivered on time the development team will not pass the course. If it becomes apparent that the project will not be completed on time, the only course of action available would be to request an extension to the deadline form the mentor.

Risk: Technology Does Not Meet Specifications

- **Mitigation**

  In order to prevent this from happening, meetings (formal and informal) will be held with the mentor on a routine business. This ensures that the product we are producing, and the specifications of the mentor are equivalent.

- **Monitoring**

  The meetings with the mentor should ensure that the mentor and our team understand each other and the requirements for the product.

- **Management**

  Should the development team come to the realization that their idea of the product specifications differs from those of the mentor, the mentor should be immediately notified and whatever steps necessary to rectify this problem should be done. Preferably a meeting should be held between the development team and the mentor to discuss at length this issue.

Risk: End Users Resist System

- **Mitigation**

  In order to prevent this from happening, the software will be developed with the end user in mind. The user-interface will be designed in a way to make use of the program convenient and pleasurable.

- **Monitoring**

  The software will be developed with the end user in mind. The development team will ask the opinion of various outside sources throughout the development phases. Specifically the user-interface developer will be sure to get a thorough opinion from others.

- **Management**

  Should the program be resisted by the end user, the program will be thoroughly examined to find the reasons that this is so. Specifically the user interface will be investigated and if necessary, revamped into a solution.

Risk: Changes in Requirements

- **Mitigation**

  In order to prevent this from happening, meetings (formal and informal) will be held with the mentor on a routine business. This insures that the product we are producing, and the requirements of the mentor are equivalent.

- **Monitoring**

  The meetings with the mentor should ensure that the mentor and our team understand each other and the requirements for the product.

- **Management**

  Should the development team come to the realization that their idea of the product requirements differs from those of the mentor, the mentor should be immediately notified and whatever steps necessary to rectify this problem should be taken. Preferably a meeting should be held between the development team and the mentor to discuss at length this issue.

Risk: Lack of Development Experience

- **Mitigation**

  In order to prevent this from happening, the development team will be required to learn the languages and techniques necessary to develop this software. The member of the team that is the most experienced in a particular facet of the development tools will need to instruct those who are not as well versed.

- **Monitoring**

  Each member of the team should watch and see areas where another team member may be weak. Also if one of the members is weak in a particular area it should be brought to the attention by that member, to the other members.

- **Management**

  The members who have the most experience in a particular area will be required to help those who don't, should it come to the attention of the team that a particular member needs help.

Risk: Database is not Stable

- **Mitigation**

  In order to prevent this from happening, developers who are in contact with the database, and/or use functions that interact with the database, should keep in mind the possible errors that could be caused due to poor programming/error checking. These issues should be brought to the attention of each of the other members that are also in contact with the database.

- **Monitoring**

  Each user should be sure that the database is left in the condition it was before it was touched, to identify possible problems. The first notice of database errors should be brought to the attention of the other team members.

- **Management**

  Should this occur, the team would call a meeting and discuss the causes of the database instability, along with possible solutions.

Risk: Poor Quality Documentation

- **Mitigation**

  In order to prevent this from happening, members who are in charge of developing the documentation will keep in contact with each developer on the team. Meetings will be held routinely to offer documentation suggestions and topics. Any topic deemed missing by a particular developer will be discussed and it will be decided whether or not to add that particular topic to the documentation. In addition, beta testers will be questioned about their opinion of the documentation.

- **Monitoring**

  Throughout development or normal in and out of house testing, the development team and or beta testers will need to keep their eyes open for any possible documentation topics that have not been included.

- **Management**

  Should this occur, the team would call a meeting and discuss the addition of new topics, or removal of unnecessary topics into the documentation.

Risk: Deviation from Software Engineering Standards

- **Mitigation**

  While it is possible to deviate from software engineering standards, it is unlikely to occur. All team members have a full understanding of the software process, and how we plan to implement them in the process.

- **Monitoring**

  Technical reviews involving comparison between documentation and the actual project will help to determine if deviation will occur. All relevant documents must be as complete and accurate as possible to ensure that work will conform to expressed software engineering standards.

- **Management**

  Should deviation occur, steps must be taken to guide the project back within the standards expressed in accompanying documents. Technical reviews help to determine what must be done to keep the project in line with established software engineering standards.

Risk: Poor Comments in Code

• **Mitigation**

Poor code commenting can be minimized if commenting standards are better expressed. While standards have been discussed informally, no formal standard yet exists. A formal written standard must be established to ensure quality of comments in all code.

• **Monitoring**

Reviews of code, with special attention given to comments will determine if they are up to standard. This must be done frequently enough to control comment quality. If they are not done, comment quality could drop, resulting in code that is difficult to maintain and update.

• **Management**

Should code comment quality begin to drop, time must be made available to bring comments up to standard. Careful monitoring will minimize the impact of poor commenting. Any problems are resolved by adding and refining comments as necessary.

Risk Table

Risks	Category	Probability	Impact
Equipment failure	TI	70%	1
Late delivery	BU	30%	1
Technology will not meet expectations	TE	25%	1
End users resist system	BU	20%	1
Changes in requirements	PS	20%	2
Deviation from software engineering standards	PI	10%	3
Less reuse than planned	PS	60%	3
Poor comments in code	TI	20%	4

PROJECT SCHEDULING AND TRACKING

To schedule this project, we split the project into various subtasks that are organized efficiently among team members to make optimal usage of time and resources. Also, we ensure that task interdependencies are minimal, so that there is no situation in which there is infinite waiting for completion of a task.

6.1 Task Set

Task set is the collection of work tasks, targets and deliverables. Task set is project-specific; task set differs for different project types. Task sets are selected using measuring parameters such as type of project and degree of rigor.

6.2 Types of Project

Concept Development Projects
Initiated to explore new business concepts or application of new technologies.
New Application Development Projects
Undertaken as a consequence of specific customer requirements.
Application Enhancement Project
Major modifications to existing softwares in terms of functionality, performance or interfaces that are observable to the end-user.
Application Maintenance Project
Correction, adaptation or extension of existing softwares to include new functionalities that are not immediately observable to the end-user.
Re-engineering Project
Undertaken with the intent to rebuild an existing system in whole or in part.

6.3 Degree of Rigor

This is a function of many software characteristics. Four different degrees of rigor can be listed:

1. Casual: All process framework activities are applied, but only a minimal task set is required. Umbrella tasks and documentation are minimized.
2. Structured: Process framework and related tasks appropriate to the project type are applied. SQA, SQM, documentation and measurement tasks conducted in a streamlined manner.
3. Strict: Full process is applied with discipline ensuring high quality. Robust work products from umbrella tasks are virtually guaranteed.
4. Quick Reaction: Process framework is applied, but because of emergency situations, only quality assurance tasks will be maintained. Documentation and reviews are done after the product is delivered to the customer (back-filling).

The grades are allotted as:

0 - Non-incidental
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
No. of potential users	4	1.10	4.4
Mission Criticality	4	1.10	4.4
Application endurance	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

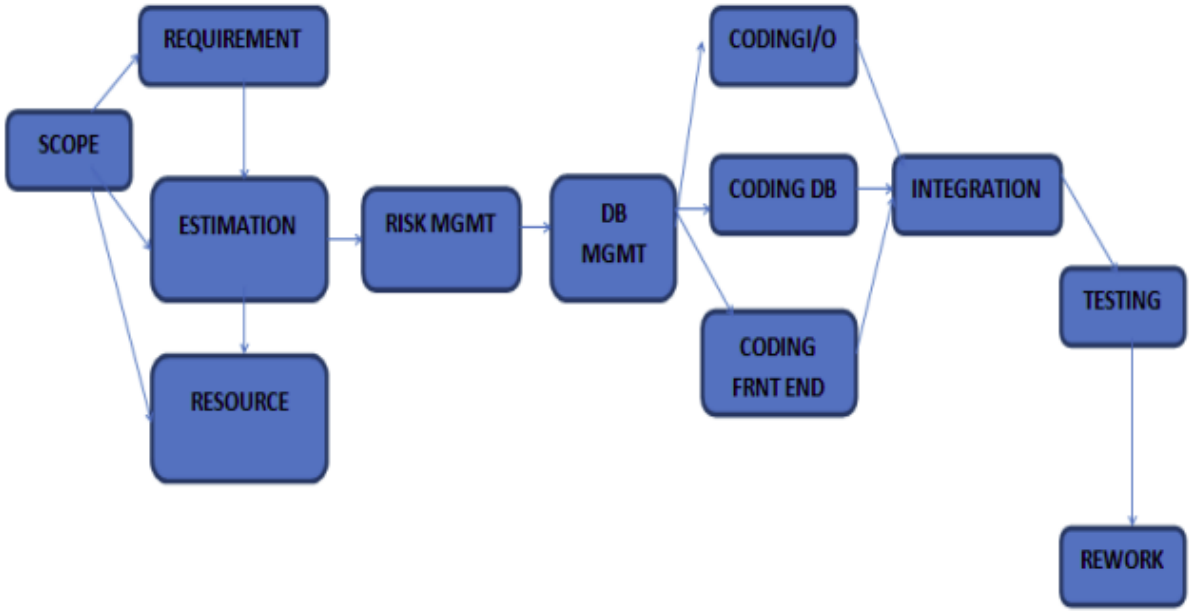
The overlap in ranges illustrates the fact that there exist no distinct boundaries between the three
degrees of rigor. If a value that falls within the overlap is received we have to make a judgment call
concerning the degree of rigor the project deserves.

6.4 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	Web Page 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:



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
  - ➢ Renegotiate 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	15/9/21	15/9/21	22/9/21	22/9/21
Estimation	5/10/21	5/10/21	10/10/21	10/10/21
Resources and arrange necessary tools and techniques, softwares, etc.	18/10/21	19/10/21	20/10/21	20/10/21
Risk analysis & management	20/10/21	20/10/21	26/10/21	26/10/21
Scheduling tasks	26/10/21	29/10/21	29/10/21	29/10/21
Preparing database and coding it	1/11/21	1/11/21	7/11/21	7/11/21
Coding for all Modules and front end	10/11/21	10/11/21	25/11/21	25/11/21
Integrating the modules	30/11/21	30/11/21	5/12/21	5/12/21
Testing	5/12/21	6/12/21	10/12/21	10/12/21
Reworking	11/12/21	10/12/21	16/12/21	16/12/21
Finalizing the project	17/12/21	17/12/21	20/12/21	20/12/21

6.5 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 projects.

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 the Gantt chart.
4) Using earned value analysis to assess progress quantitatively.

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 a relatively brief document that is addressed to a diverse audience. An outline of the plan is presented below.

Project planning process

1. A statement of work (SOW) that describes all work 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:

7.1 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.

In project management, the problem statement is part of the project charter and defines what the problem is so that the project team and stakeholder can focus their attention on solving the problem. It is important to have a good problem statement before starting eliciting requirements for a solution. Problem statement for this project is :

7.2 Design a software that implements a cricket data management system.

Match officials input the match details like teams playing, squads, venue. The software will take live ball to ball commentary as input and the software will generate a scoresheet of each match.

The name of the tournament could be entered along with the full schedule, venue and team details. While entering the details of the match,the software will update the match details table for the individual players.

The software will also manage the database containing profiles and records of players of each ICC cricketing nation. A user can view all the statistics of a player in each format. The data will then be segregated and stored under the respective players' column in the database (as his own record). Thus the player's scorecard is updated after each match. The scorecard will contain the tournament wise summary of runs scored, wickets taken, catches caught (for that tournament).

Processing the different stats of data, providing valuable insights in different domains of the game which will help enhance the performance of the player and will also provide recommendations of players for different formats.

7.3 Project Estimation

The project estimates provide cost, effort and time for the project Cricket Data Management.

Project planning involves estimation to determine how much money, how much effort, how many

resources, and how much time it will take to build a specific software-based system or product.

7.3.1 Problem Based Estimation

Lines of Code and Function Points are used as measures from which productivity metrics can be computed.LOC and FP data are used in two ways during software project estimation:

a. as an estimation variable to "size" each element of the software and
b. As baseline metrics collected from past projects and used in conjunction with estimation variables to develop cost and effort projections.

7.3.2 LOC Based Estimation

A formal method to measure size by counting the 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.

Function	Optimistic	Most Likely	Pessimistic	Estimate LOC
User Interface	300	500	800	517
Input Commentary	400	600	800	600
Scorecard Development	400	600	800	600
Tournament and Player Statistics	600	800	1200	833
Database management	300	500	700	500
Estimated Lines of Code				3050

7.3.3 FP Based Estimation

Function Point Analysis (FPA) is a 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. It is independent of the technology used to implement the system.

Measurement Parameter	Count	Simple	Average	Complex	Function Points
Number of user input	10	2	4	12	40
Number of user output	8	4	5	7	56
Number of user inquiries	0	0	4	6	0
Number of files	3	7	10	15	30
Number of external interfaces	1	5	7	10	5
Count total					131

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	Ratings
Backup and recovery	4
Data communications	3
Distributed processing	2
Performance critical	5
Existing operating environment	5
On-line data entry	5
Input transaction over multiple screens	2
Master files updated on-line	5
Information domain values complex	2
Internal processing complex	4
Code designed for reuse	3
Conversion/installation in design	1
Multiple installations	1
Application designed for change	3

FP = Count Total * (0.65 + 0.01*Fi's)

= 131 * (0.65 + 0.01*45)

= 144

7.3.4 Process Based Estimation

The process is decomposed into a relatively small set of tasks and the effort required to accomplish each task is estimated.

7.4 Project Resources

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

7.4.1 Human

It is a personnel pool like developers, managers, planners, testers etc available to an organization. This project consists of a group of 5 members (Nirmit Deliwala, Mann Doshi, Meet Parekh, Aayush J Shah, Donovan Crasta).

7.4.2 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. Reusable resources include:

- Front-end layout for signup/login.
- Java APIs:
    - Java.awt
    - Java Socket API
    - Java Networking API

7.4.3 System Requirements

It consists of planning, managing and making the necessary software/hardware components available. Hardware tools like our laptops and respective internet connections, along with softwares like Operating Systems (Windows/Ubuntu), MySQL database, IDE/Text Editors (VSCode/Sublime).

7.5 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:

7.5.1 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.

7.5.2 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.

7.5.3 Risk Mitigation, Monitoring and 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.

Risk Table

Risks	Category	Probability	Impact
Computer Crash	TI	70%	1
Late Delivery	BU	30%	1
Technology will not meet Expectations	TE	25%	1
End Users Resist System	BU	20%	1
Changes in Requirements	PS	20%	2
Lack of Development Experience	TI	20%	2
Lack of Database Stability	TI	40%	2
Poor Quality Documentation	BU	35%	2
Deviation from Software Engineering Standards	PI	10%	3
Poor Comments in Code	TI	20%	4

Impact Values:

1 – Catastrophic
2 – Critical
3 – Marginal
4 – Negligible

7.6 Project Scheduling and Planning

7.6.1 Defining Task Set for Software Project

A task set is the collection of work tasks, milestones, and deliverables. The grades are allotted as:

0 - Non-incidental

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
No. of potential users 4	1.10	4.4
Mission Criticality 4	1.10	4.4
Application endurance 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

7.6.2 Task Network

Project tasks and dependencies are noted diagrammatically in task network according to its functional dependencies.

Task Description
T1 System Design
T2 Detailed Design
T3 Database Implementation
T4 Web Page Design
T5 Coding Input Module
T6 Coding database related module
T7 Coding Output Module
T8 Test Planning T9 Integration

7.6.3 Timeline chart

Project milestones can be shown in a simple timeline chart. While the chart

doesn't look complicated, it provides good amount of information on project progress in a simple and understandable chart.

Task	Planned Start	Actual Start	Planned End	Actual End
Scope and objective, requirement gathering, planning	15/9/21	15/9/21	22/9/21	22/9/21
Estimation	5/10/21	5/10/21	10/10/21	10/10/21
Resources and arrange necessary tools and techniques, softwares, etc.	18/10/21	19/10/21	20/10/21	20/10/21
Risk analysis & management	20/10/21	20/10/21	26/10/21	26/10/21
Scheduling tasks	26/10/21	29/10/21	29/10/21	29/10/21
Preparing database and coding it	1/11/21	1/11/21	7/11/21	7/11/21
Coding for all Modules and front end	10/11/21	10/11/21	25/11/21	25/11/21
Integrating the modules	30/11/21	30/11/21	5/12/21	5/12/21
Testing	5/12/21	6/12/21	10/12/21	10/12/21
Reworking	11/12/21	10/12/21	16/12/21	16/12/21
Finalizing the project	17/12/21	17/12/21	20/12/21	20/12/21

7.6.4 Tracking the Schedule

Tracking helps ensure that your project is completed on time and within budgetary constraints.
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 projects.

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 the Gantt chart.

SOFTWARE QUALITY ASSURANCE PLAN

8.1 Introduction

8.1.1 Purpose

This document outlines the actions of our team in order to make our system “Cricket Data Management” and other related artifacts conform to the requirements of the customers and the Qualitative standards within the specified project resources.

8.1.2 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:

i. Identifies the SQA responsibilities of the project developer and the SQA consultant

ii. Lists the activities, processes, and work products that the SQA consultant will review and
audit

iii. Identifies the SQA work products

8.2 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:

8.2.1 Organization

This project is a team work where the strength of our team is 5 members.
The tasks are divided among the members by mutual co/ordination.

8.2.2 Roles

The roles of the members are planning, implementation, testing, error handling, design and research.

8.2.3 Tasks and Responsibilities

The responsibilities of the developers are as follows:

i. Develop the requirement specification and cost estimation for the project

ii. Develop the design plan and test plan for testing the tool

iii. Implement and test the application and deliver the application along with the

necessary documentation.
iv. 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.
v. Planning, coordinating, testing and assessing all aspects of quality issues.

The responsibilities of the committee members are to:
i. Review the work performed by the developer
ii. Provide feedback and advice

8.3 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.

8.4 Standards, practices, conventions, and metrics

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

8.5 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.

8.6 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.

8.7 Problem reporting and corrective action

The corrective action process describes the steps for
i. problem identification and correction occurring during software development to verify early detection of actual or potential problems
ii. reporting of the problem to the proper authority,
iii. analysis of the problem to propose corrective measures,
iv. timely and complete corrective action
v. The recording and follow/up of each problem's status.

8.8 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.

8.9 Code control

Code control includes the items listed below:
a. Identifying, labeling, 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

8.10 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 a few years.

8.11 Risk management

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

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

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.
There are four fundamental sources of change:
• New business or market conditions dictate changes in product requirements or business rules.
• New customers demand modification of data produced by information systems, functionality delivered by products, or services delivered by a computer-based system
• Reorganization or business growth/downsizing causes changes in project priorities or software engineering team structure.
• Budgetary or scheduling constraints cause a redefinition of the system or product.

Because many work products are produced when software is built, each must be uniquely identified. Once this is accomplished, mechanisms for version and change control can be established. To ensure that quality is maintained as changes are made, the process is audited; and to ensure that those with a need to know are informed about changes, reporting is conducted.

A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures.

9.1 Configuration Management

The items that comprise all information provided 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. For this matter, 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 customers 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
- Reports 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.

9.2.1 SCI: Software Configuration Items

There are a growing number of artefacts for managing sharing the SE process. They are
1. Programs: source level and executable forms
2. Documents:Technical practitioners and users
3. Data: internal and external

9.2.2 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

9.2.3 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 the Cricket Data Management System, the changes in terms of the number of people accessing the 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 a control measure.

4) Configurational Auditing: Ensure proper implementation: A software configuration audit complements the formal technical review by assessing a configuration object.

5) Reporting changes to others.

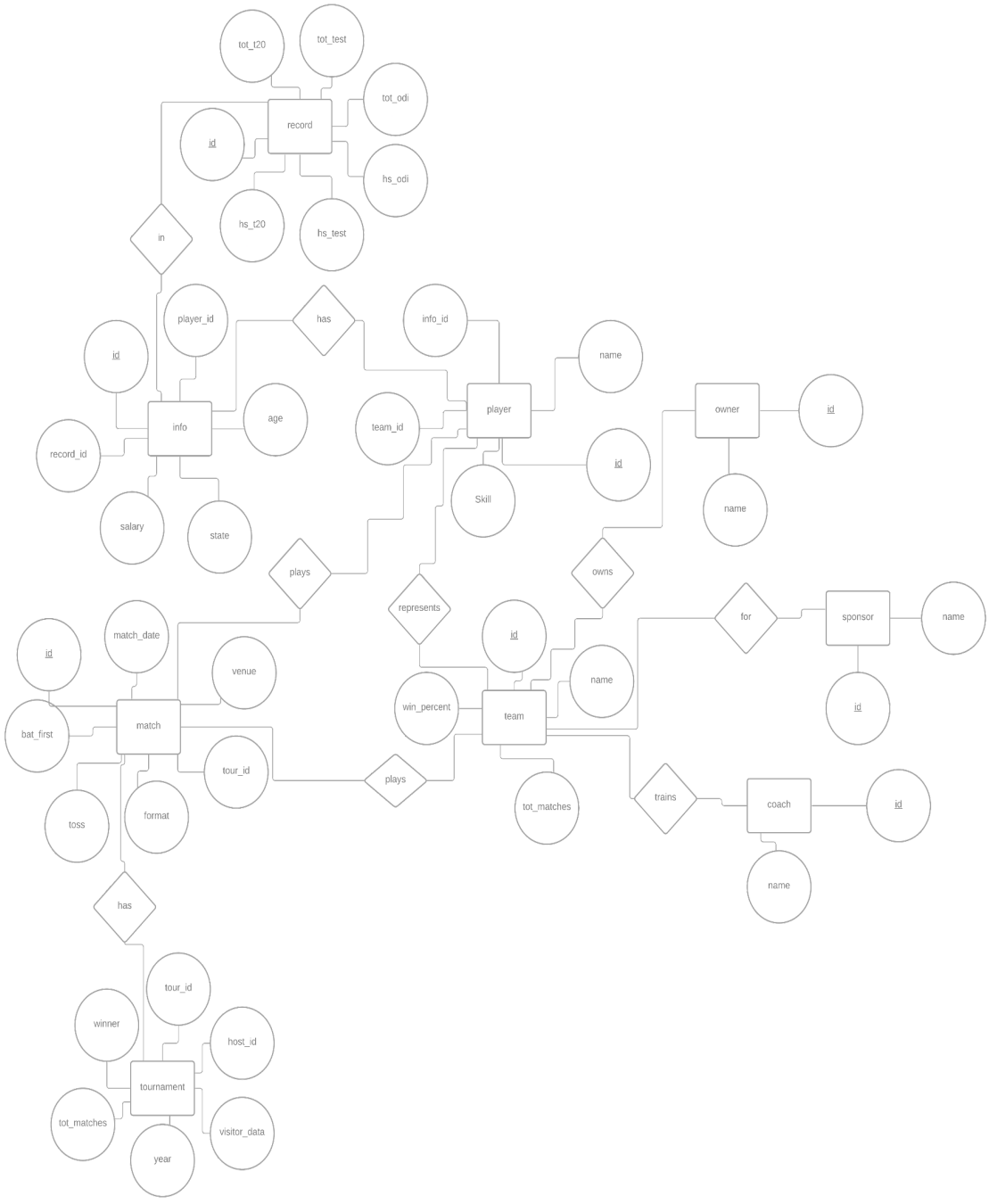
REQUIREMENT ANALYSIS MODELING

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

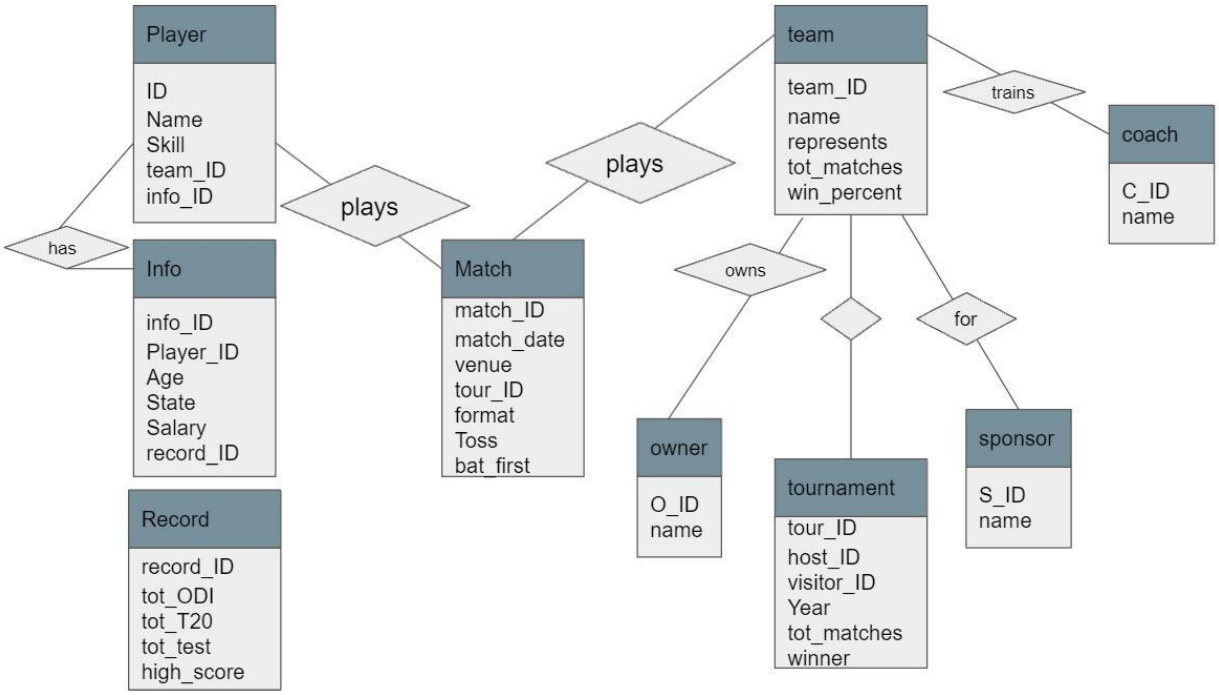
10.1 Data Modelling: Entity Relationship Diagram

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 object noted in the ERD can be described using data object description.

ER Diagram


tot_t20
tot_test
tot_odi
record
id
hs_odi
hs_t20
hs_test
in
player_id
has
info_id
id
name
info
age
team_id
player
owner
id
record_id
id
Skill
name
salary
state
plays
owns
represents
for
sponsor
name
id
match_date
id
name
venue
win_percent
team
id
bat_first
match
tour_id
plays
trains
coach
id
toss
format
tot_matches
name
has
tour_id
winner
host_id
tournament
tot_matches
visitor_data
year

Relational Schema

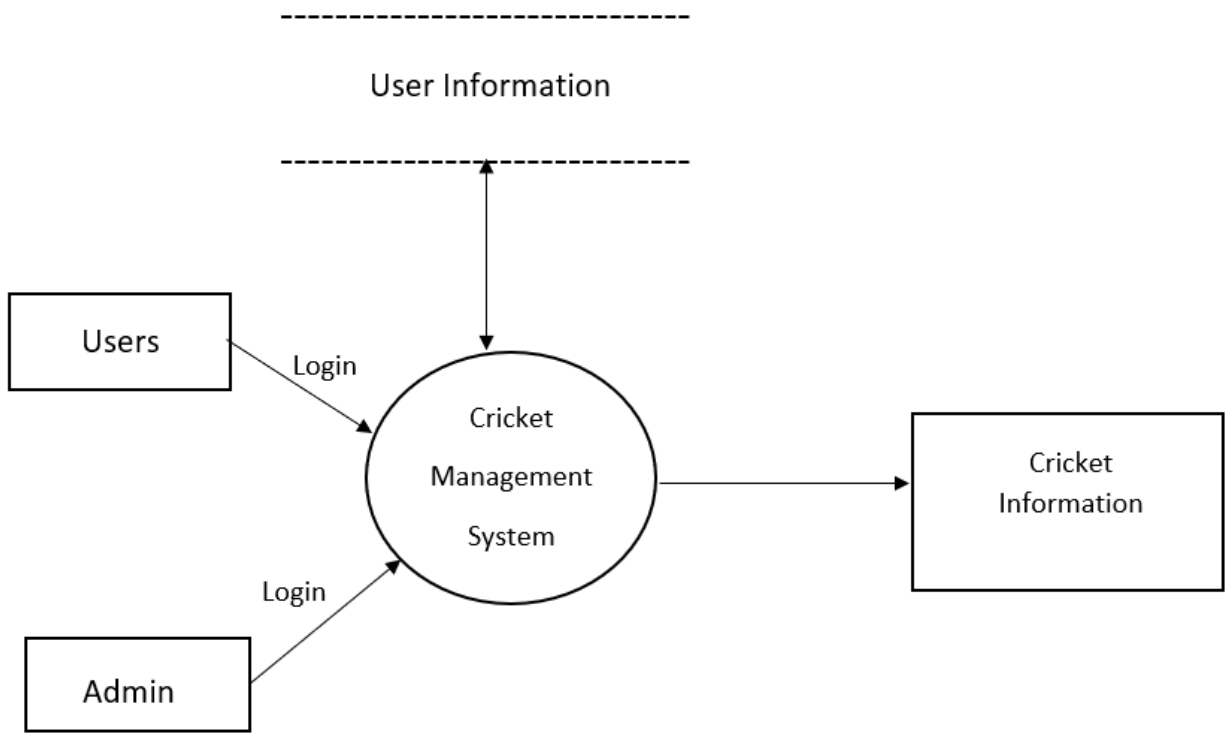


10.2 Functional Modelling: Data Flow Diagram

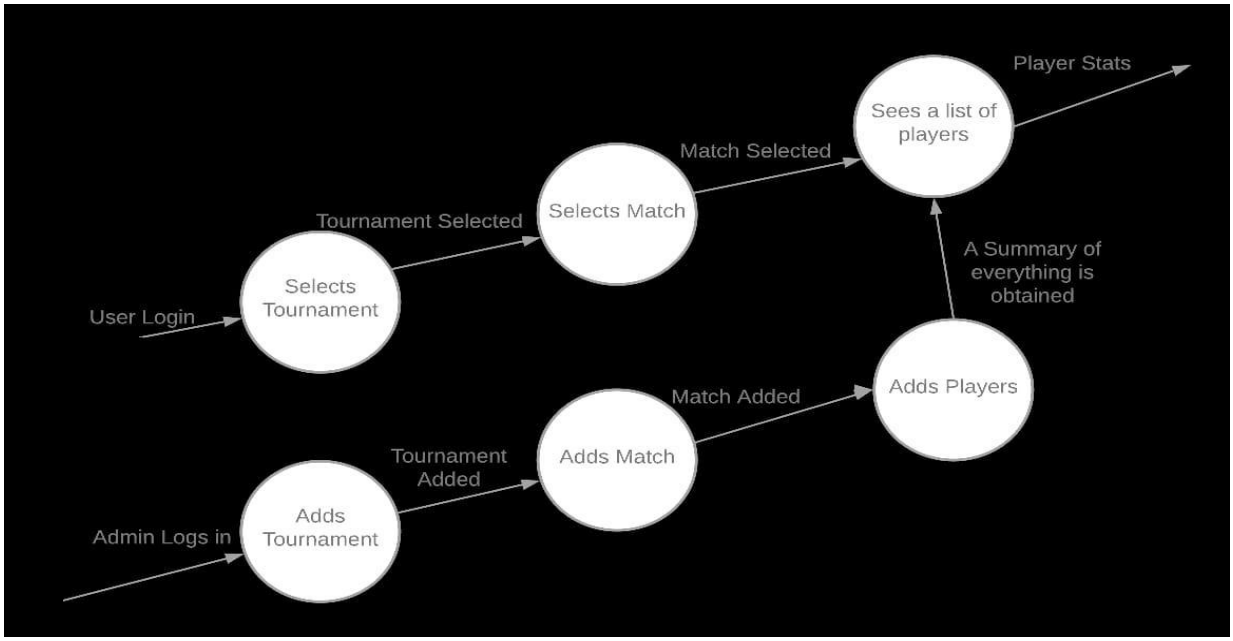
An activity that translates the data model developed during analysis into implementable data structures. Data models support data and computer systems by providing the definition and format of data. If this is done consistently across systems then compatibility of data can be achieved. If

the same data structures are used to store and access data then different applications can share data. The results of this are indicated below. However, the system and interface often cost more than they should, to build, operate and maintain. They may also constrain the business rather than support it. A major cause is that the quality of the data models implemented in the system and the interface is poor.

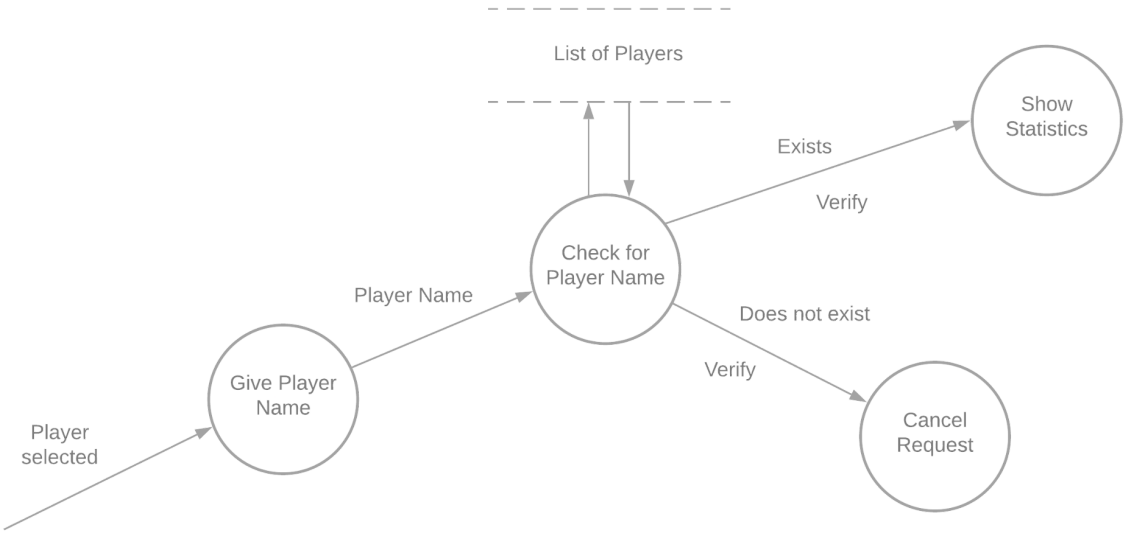
DFD Level 0:



DFD Level 1:



DFD Level 2:



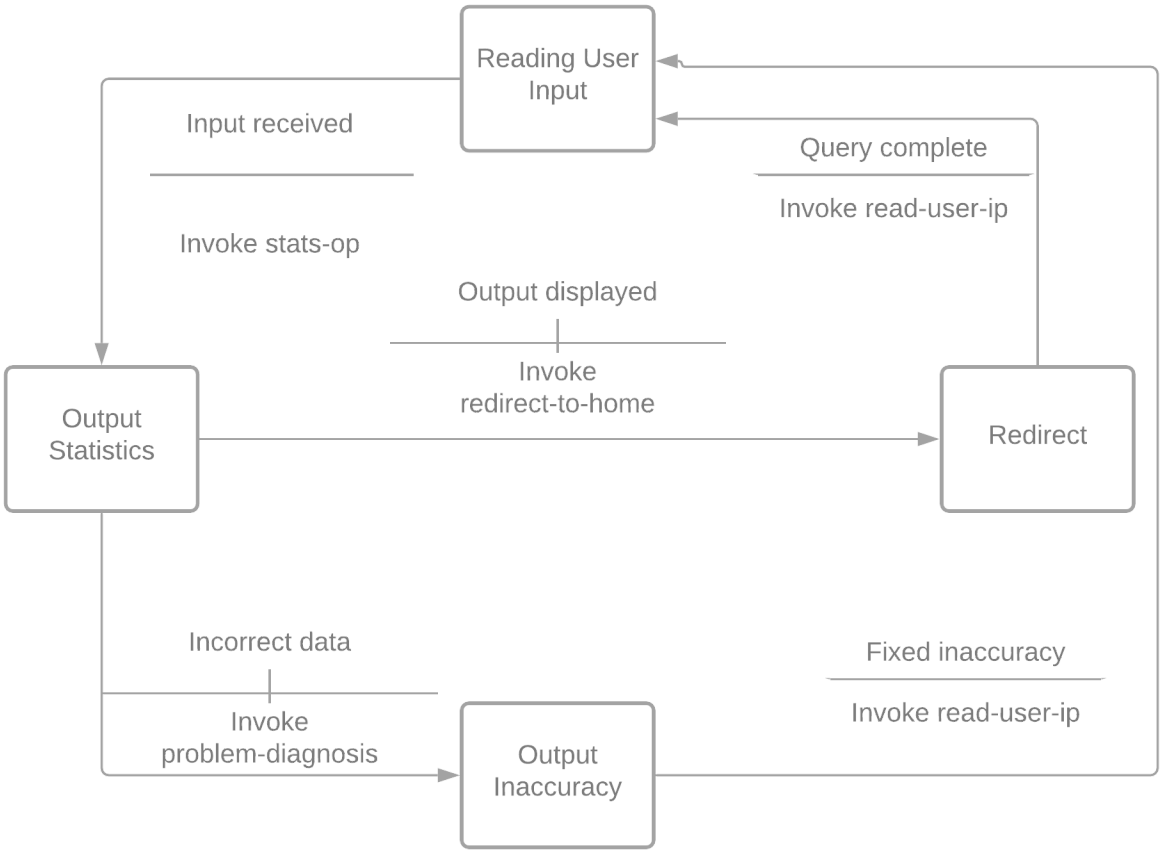
10.3 Data Dictionary

The analysis model encompasses representation of data objects, functions and control. In each representation data objects and/or control items play a role. Therefore, it is necessary to provide an organized approach for representing the characteristics of each data object and control item. This is accomplished with a data dictionary.

Title of Object Class	Summary/ Description
User	The person that logs into the system and provides a score for the match.
Administrator	The person who manages and monitors and also provides the scorecard and info.
Players	This contains the list of all the players per team in our database.
Matches	This contains the list of matches per tournament once the tournament is input.
Tournaments	This contains the list of all tournaments.

10.4 Behavioral Model

Behavioral modeling is an operational principle for all requirements analysis methods. The state transition diagram represents the behavior of a system by depicting its states and the events that cause the system to change state. In addition, the STD indicates what actions (e.g., process activation) are taken as a consequence of a particular event. A state is any observable mode of behavior.



10.5 Software Resource Specification

10.5.1 Introduction

Purpose

This SRS document presents a detailed description of the cricket management system. It helps manage cricket data effectively. Taking the scorecard as an input, various stats for players are generated that help provide valuable insights to the hiring officials and management. This in turn helps improve a player's performance.

Product Scope

Cricinfo is a general purpose application designed for cricket lovers. While it displays stats player-wise and team-wise like any other application, it has a much wider scope. Match officials could add in match details and create custom tournaments, add matches and the players that would be going on the tour. Naive users could view the data. Data abstraction is done at all levels so that international match officials could not see local custom matches.

10.5.2 Overall Description

Product Functions

We can subdivide the project into 6 main features. A gist of all the functions is mentioned below.

Login

Description This function allows a registered user to login his account using his/her email address and password. If a user is not registered, the system shall ask the user to register first. The system will check both the email address and password, when a user attempts to login.

Rationale: This provides security to the system by authenticating each member and provides confidence to the user that his/her personal information is secure.

Registration

Description This function allows unregistered users to register and to create a new account in the system. In order to create a new account, the user has to provide required information such as first name, last

name, email address and password. Other optional information, such as phone number and mailing address, can be provided later while editing the profile. The system checks if all required data are provided and then will prompt the user to enter additional information, if required.

Rationale: A user who wishes to create surveys and take non-anonymous surveys must be logged in. However, without registration, a user can never be a member. This section offers all users a chance to become a member.

View Tournaments

Description The user can use the View tournaments function to view the existing tournaments in the system. This will depend on the level of abstraction done. Once a user selects the tournament, they could view a list of all the matches, the players playing in each of the teams and a leaderboard.

Create Tournaments

Description This function could be used to create your own custom tournament. Users could add in matches, players and all the relevant details required. The application takes the data, generates leaderboards, stats and all kinds of information related to it.

My Account

Description This section gives the user the power to view, save, edit, or delete the information stored in his/her account. The user can check, edit and update the created tournaments/matches. This feature is not available for unregistered users.

Rationale: A customer's information changes time to time. Giving the users a way to modify their account information allows them to update their profile.

Logout

Description The Logout section provides a way for the user to securely log out of the system. This process will save all user operations when he/she exits the system. If a user wishes to continue accessing the system, he/she must login again.

Rationale: Customers often use shared computers. Providing a way to clear state and logout gives customers confidence that nobody else will use their data, keeping them confidential.

User Classes and Characteristics

The main actors in the system are (1) the user, (2) an organization and (3) the tours. The user will create a survey and view various tournaments. Brief descriptions of these classes are as follows:

- User
  - Has properties like Name, Address, Email ID.
  - Associated with created and participated tournaments.
- Organization
  - Has rights to prepare as well as to contribute to tournaments.

Our user may be associated with multiple tournaments, and many users may be associated with any tournament. Thus, a many-to-many relationship exists within the system.

Operating Environment

OE-1: The system shall use Java and MySQL

User Documentation

UD-1: All end-user documentation shall be accessible and viewable in the system.

UD-2: Documentation shall be written for end users.

UD-3: Documentation shall be written for developers who wish to modify or extend the system.

Assumptions and Dependencies

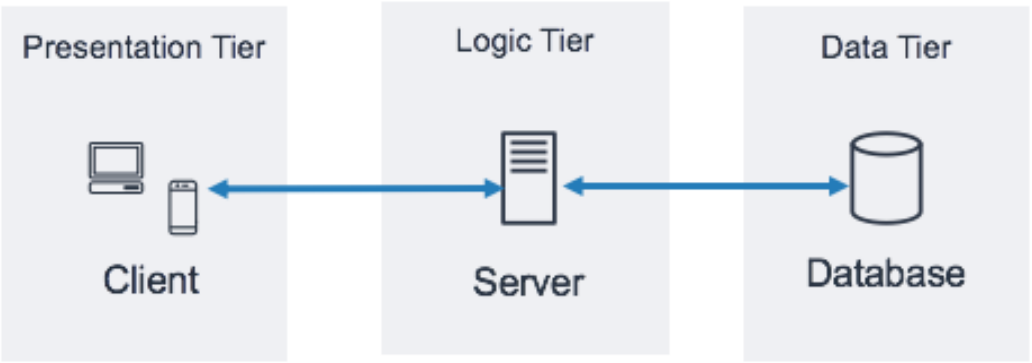
AD-1: The system shall provide information on surveys and allow users to satisfy their

requirements for creating tours, either for personal use or by organizations.

10.5.3 System Architecture

Overview

The system architecture of the online survey system is based on a 3-tier architecture. This generally includes three phases viz. Client side, Server side and Database management.



3 Tier Architecture

Client Side

This includes what the user sees, such as text, images, and the rest of the UI, along with any actions that an application performs. The end user application including GUI and integration with the server using API is done in this tier. All the data rendered is retrieved from the server.

Server Side

The client side application requests data from the server. The server processes the request and tries to generate a response. If needed, the server fetches data from the database and constructs an organised object to return the requested query from the client.

Database Management

Database includes the persistent storing of the user generated data. It contains some structure to store the data. When the server requests specific data related to a specific query, the database management system searches in the pool of data. Depending on the type of application, data is classified as structured, semi-structured or unstructured. For our use case, the data generated is unstructured and hence a NoSQL database can be used.

10.5.4 Other Nonfunctional Requirements

Performance Requirements

- The system shall update the statistics of the respective player/ match immediately so that data consistency is maintained.

- The system shall update the edited details for all the users who are viewing the data at a given time.

Security Requirements

- Passwords must be a minimum of eight characters with at least one uppercase alphabet, one lowercase alphabet, one special symbol and one numeric digit.

- Email addresses should be verified before the system grants user access. This verification shall be exercised by sending the prospective user a confirmation email after enrollment. This email must contain information specific to completing the enrollment process.

- All exchanges from client to server involving private data shall occur using the highest available level of secure connection (e.g., https).

Software Quality Attributes

Usability:

The survey system design shall allow deployment on both Windows and UNIX (Linux) servers. The design should support Windows Server 2003, Linux 2.6.x, V10 UNIX and later.

10.5.5 Other Requirements

This project requires financial support for database and hosting expenditures. It also needs to be legally registered for commercial use. The system requires a validation for satisfying legal conditions of various countries for deployment of the system internationally.

DESIGN

11.1 Data Design

11.1.1 Description of each table

1. **matches**(tour_id,match_id, team1_id, team2_id, venue, start_date, end_date,toss_win, bat_first,match_win,match_win_quantity, man_of_the_match, team1_runs, team2_runs, team1_wickets, team2_wickets);

```
Database changed
mysql> desc matches;
+---------------------+-------------+------+-----+---------+----------------+
| Field               | Type        | Null | Key | Default | Extra          |
+---------------------+-------------+------+-----+---------+----------------+
| match_id            | int         | NO   | PRI | NULL    | auto_increment |
| tour_id             | int         | YES  | MUL | NULL    |                |
| team1_id            | int         | YES  | MUL | NULL    |                |
| team2_id            | int         | YES  | MUL | NULL    |                |
| venue               | varchar(20) | YES  |     | NULL    |                |
| start_date          | date        | YES  |     | NULL    |                |
| end_date            | date        | YES  |     | NULL    |                |
| toss                | int         | YES  | MUL | NULL    |                |
| bat_first           | int         | YES  | MUL | NULL    |                |
| match_win           | int         | YES  | MUL | NULL    |                |
| win_quantity        | int         | YES  |     | NULL    |                |
| man_of_the_match    | int         | YES  | MUL | NULL    |                |
| team1_runs          | int         | YES  |     | NULL    |                |
| team2_runs          | int         | YES  |     | NULL    |                |
| team1_wickets       | int         | YES  |     | NULL    |                |
| team2_wickets       | int         | YES  |     | NULL    |                |
| team1_name          | varchar(20) | YES  |     | NULL    |                |
| team2_name          | varchar(20) | YES  |     | NULL    |                |
| match_type          | varchar(20) | YES  |     | NULL    |                |
| status              | varchar(20) | YES  |     | NULL    |                |
| on_strike_batsmen   | int         | YES  | MUL | NULL    |                |
| non_strike_batsmen  | int         | YES  | MUL | NULL    |                |
| curr_bowler         | int         | YES  | MUL | NULL    |                |
| on_strike_bat_name  | varchar(40) | YES  |     | NULL    |                |
| non_strike_bat_name | varchar(40) | YES  |     | NULL    |                |
| curr_bowl_name      | varchar(40) | YES  |     | NULL    |                |
| overs_team1         | float(3,1)  | YES  |     | NULL    |                |
| overs_team2         | float(3,1)  | YES  |     | NULL    |                |
+---------------------+-------------+------+-----+---------+----------------+
28 rows in set (0.05 sec)

mysql>
```

2. **scorecard**(match_id, player_id, runs, balls_played,fours,sixes,out_style,wicket_by,wickets,overs,maiden,runs_given,extras)

```
mysql> desc scorecardBatting;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| match_id     | int         | NO   | PRI | NULL    |       |
| player_id    | int         | NO   | PRI | NULL    |       |
| runs         | int         | YES  |     | NULL    |       |
| balls_played | int         | YES  |     | NULL    |       |
| fours        | int         | YES  |     | NULL    |       |
| sixes        | int         | YES  |     | NULL    |       |
| out_style    | varchar(20) | YES  |     | NULL    |       |
| bowled_by    | varchar(20) | YES  |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
8 rows in set (0.00 sec)

mysql> desc scorecardBowling;
+------------+------------+------+-----+---------+-------+
| Field      | Type       | Null | Key | Default | Extra |
+------------+------------+------+-----+---------+-------+
| match_id   | int        | NO   | PRI | NULL    |       |
| player_id  | int        | NO   | PRI | NULL    |       |
| wickets    | int        | YES  |     | NULL    |       |
| maiden     | int        | YES  |     | NULL    |       |
| overs      | float(4,1) | YES  |     | NULL    |       |
| runs_given | int        | YES  |     | NULL    |       |
+------------+------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql>
```

```
mysql> desc extras;
+-------------+------+------+-----+---------+-------+
| Field       | Type | Null | Key | Default | Extra |
+-------------+------+------+-----+---------+-------+
| match_id    | int  | YES  | MUL | NULL    |       |
| bat_team_id | int  | YES  | MUL | NULL    |       |
| no_ball     | int  | YES  |     | NULL    |       |
| wide        | int  | YES  |     | NULL    |       |
| byes        | int  | YES  |     | NULL    |       |
+-------------+------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql>
```

3. **team**(team_id, player_id, player_name, designation, tour_id);

```
mysql> desc team;
+------------+-------------+------+-----+---------+----------------+
| Field      | Type        | Null | Key | Default | Extra          |
+------------+-------------+------+-----+---------+----------------+
| team_id    | int         | NO   | PRI | NULL    | auto_increment |
| tour_id    | int         | YES  | MUL | NULL    |                |
| team_name  | varchar(40) | YES  |     | NULL    |                |
| team_alias | varchar(7)  | YES  |     | NULL    |                |
+------------+-------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)

mysql>
```

4. **tournament**(id, name, format, tot_teams, year, team_won, user_id);

```
mysql> desc tournament;
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| tour_id     | int         | NO   | PRI | NULL    | auto_increment |
| name        | varchar(20) | YES  |     | NULL    |                |
| format      | varchar(30) | YES  |     | NULL    |                |
| tot_teams   | int         | YES  |     | NULL    |                |
| year        | int         | YES  |     | NULL    |                |
| team_won_id | int         | YES  |     | NULL    |                |
| user_id     | varchar(20) | YES  |     | NULL    |                |
| overs       | int         | YES  |     | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)

mysql>
```

5. **player**(player_id, name, skill, hand, salary, team_id);

```
mysql> desc players;
+-----------+-------------+------+-----+---------+----------------+
| Field     | Type        | Null | Key | Default | Extra          |
+-----------+-------------+------+-----+---------+----------------+
| player_id | int         | NO   | PRI | NULL    | auto_increment |
| name      | varchar(20) | YES  |     | NULL    |                |
| skill     | varchar(30) | YES  |     | NULL    |                |
| hand      | varchar(8)  | YES  |     | NULL    |                |
| team_id   | int         | YES  | MUL | NULL    |                |
| age       | int         | YES  |     | NULL    |                |
| role      | varchar(30) | YES  |     | NULL    |                |
+-----------+-------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)

mysql>
```

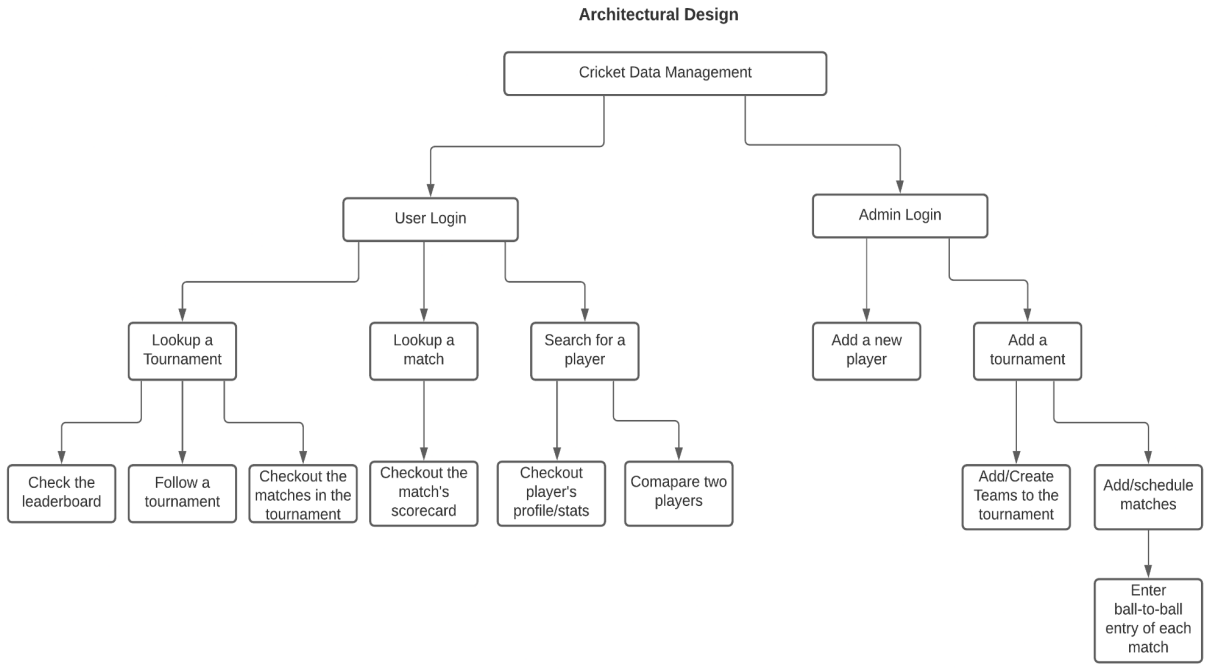
6. **stats**(id, format, runs, fours, sixes, centuries, half_centuries, avg, strike_rate, wickets, runs_given, economy, fifers, hattricks);

```
mysql> desc stats;
+--------------+------------+------+-----+---------+-------+
| Field        | Type       | Null | Key | Default | Extra |
+--------------+------------+------+-----+---------+-------+
| player_id    | int        | YES  | MUL | NULL    |       |
| runs         | int        | YES  |     | NULL    |       |
| balls_played | int        | YES  |     | NULL    |       |
| fours        | int        | YES  |     | NULL    |       |
| sixes        | int        | YES  |     | NULL    |       |
| avg          | float(4,2) | YES  |     | NULL    |       |
| strike_rate  | float(5,2) | YES  |     | NULL    |       |
| half_cent    | int        | YES  |     | NULL    |       |
| cent         | int        | YES  |     | NULL    |       |
| wickets      | int        | YES  |     | NULL    |       |
| runs_given   | int        | YES  |     | NULL    |       |
| hatricks     | int        | YES  |     | NULL    |       |
| fifers       | int        | YES  |     | NULL    |       |
| economy      | float(3,2) | YES  |     | NULL    |       |
+--------------+------------+------+-----+---------+-------+
14 rows in set (0.00 sec)

mysql>
```

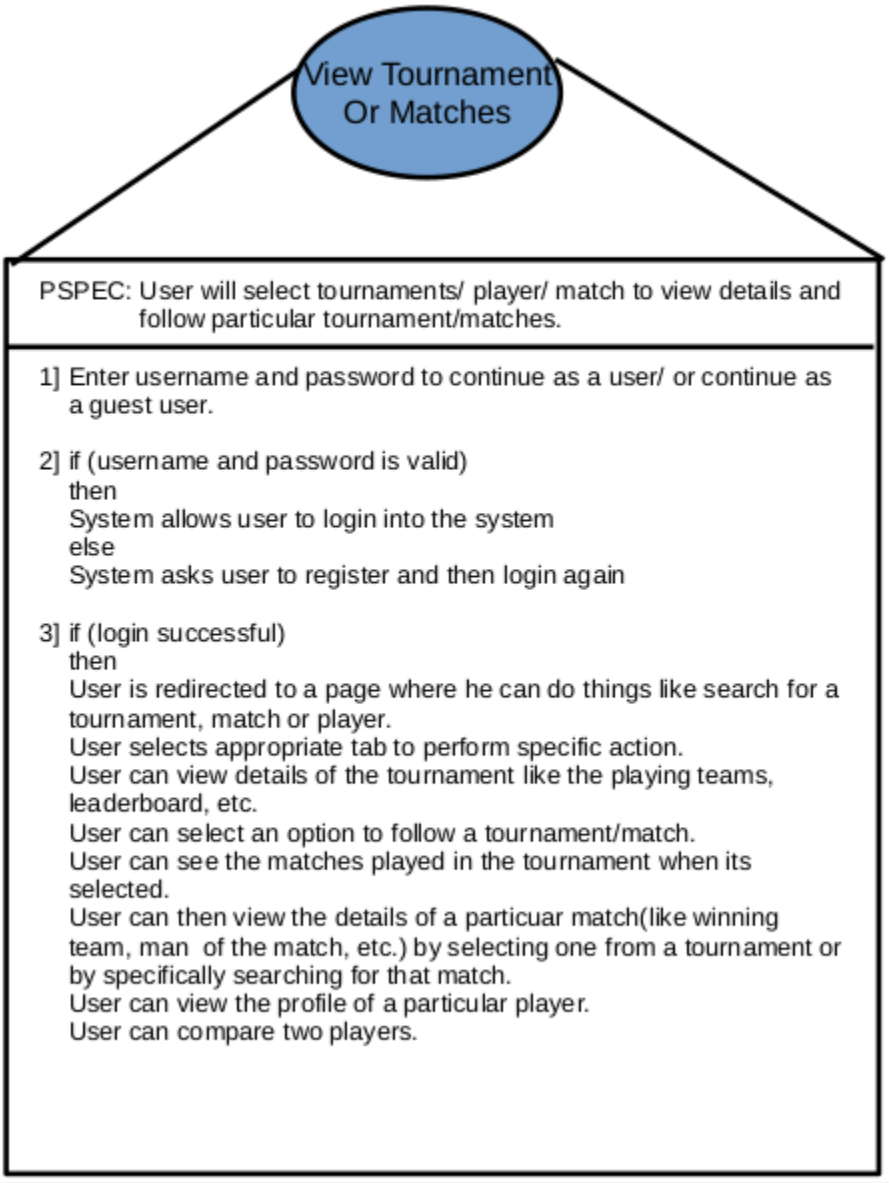
7. **user**(id, username, password);

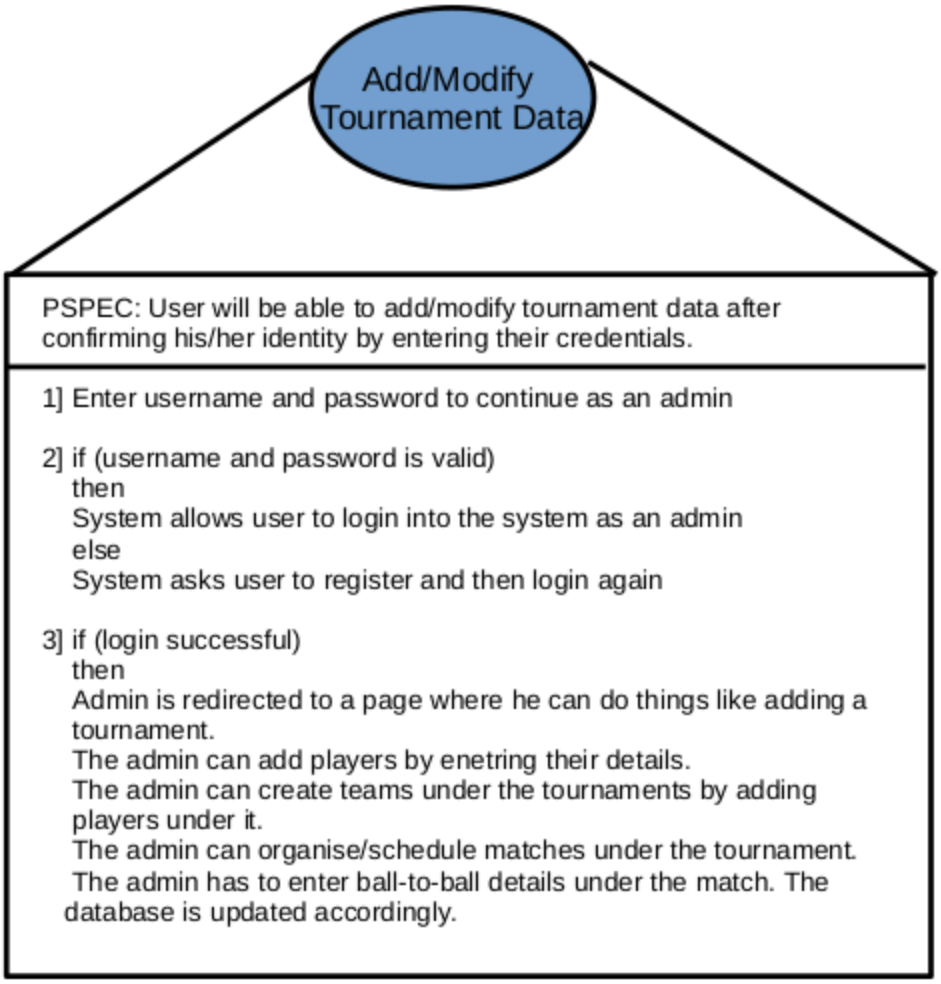
```
mysql> desc user;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| username | varchar(15) | NO   | PRI | NULL    |       |
| password | varchar(50) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.01 sec)

mysql>
```

11.2 Architectural Design



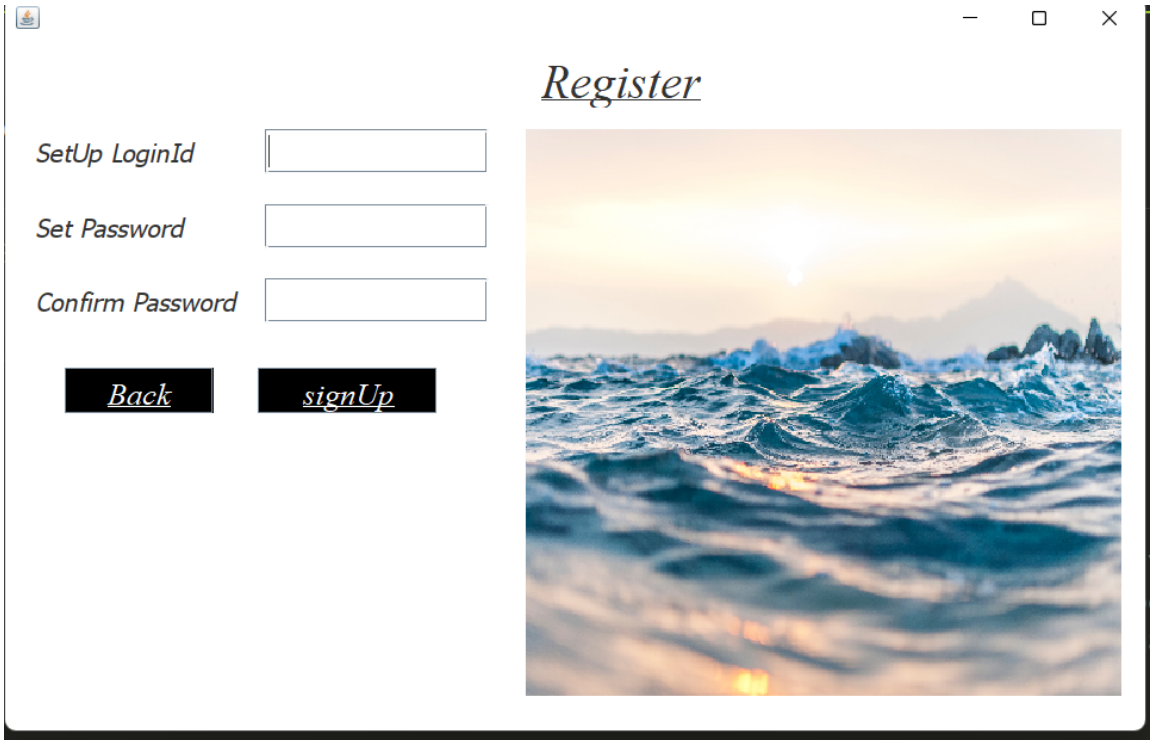
11.3 PSPEC (Project Specification)



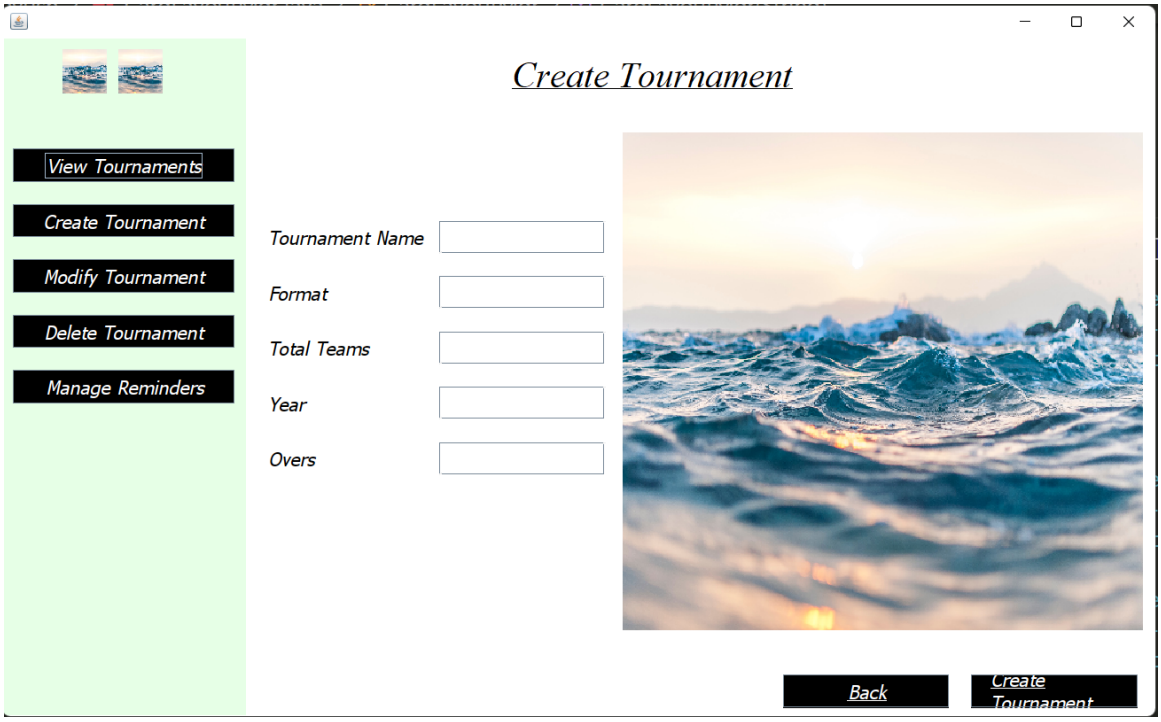

Add/Modify Tournament Data
PSPEC: User will be able to add/modify tournament data after confirming his/her identity by entering their credentials.
1] Enter username and password to continue as an admin
2] if (username and password is valid)
then
System allows user to login into the system as an admin
else
System asks user to register and then login again
3] if (login successful)
then
Admin is redirected to a page where he can do things like adding a tournament.
The admin can add players by enetring their details.
The admin can create teams under the tournaments by adding players under it.
The admin can organise/schedule matches under the tournament.
The admin has to enter ball-to-ball details under the match. The database is updated accordingly.

11.4 User Interface

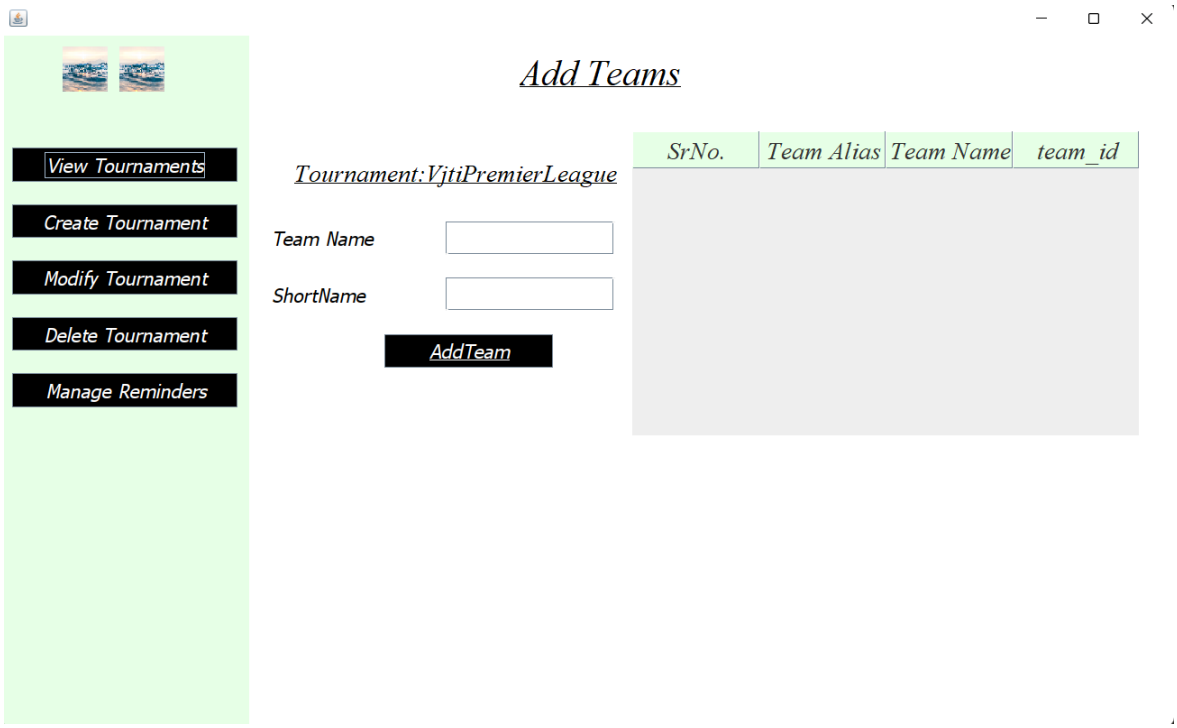
11.4.1 User Register



11.4.2 Create Tournament

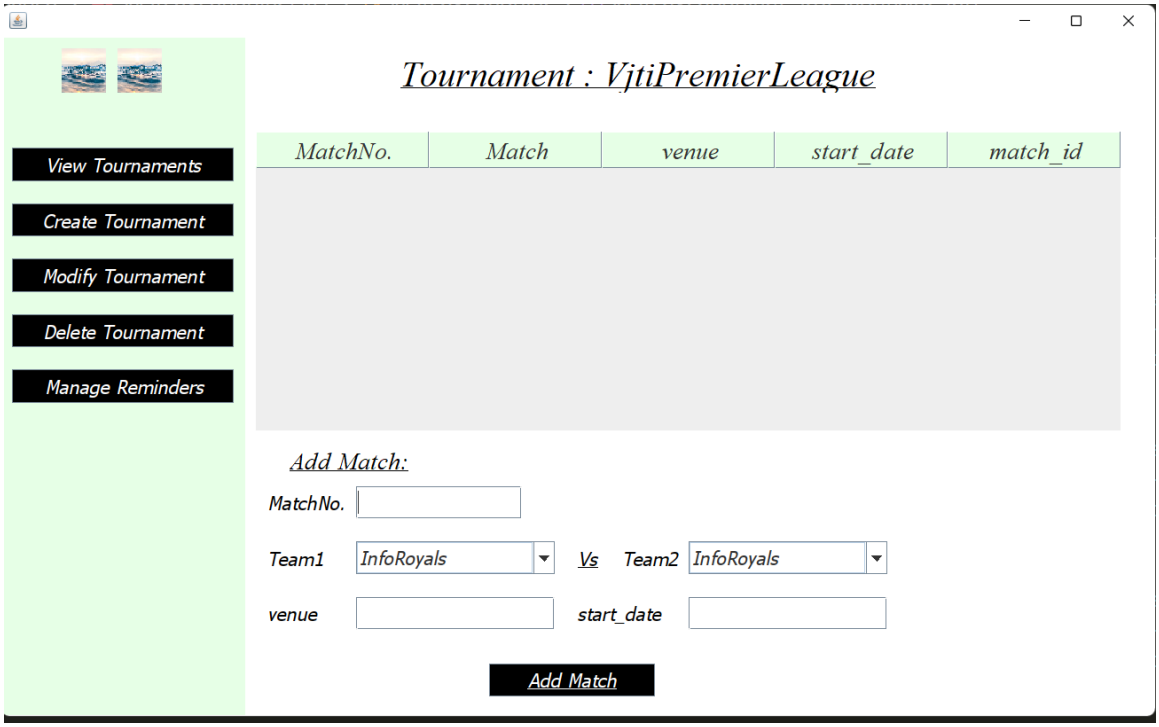


11.4.3 Add Teams



11.4.4 Add Players

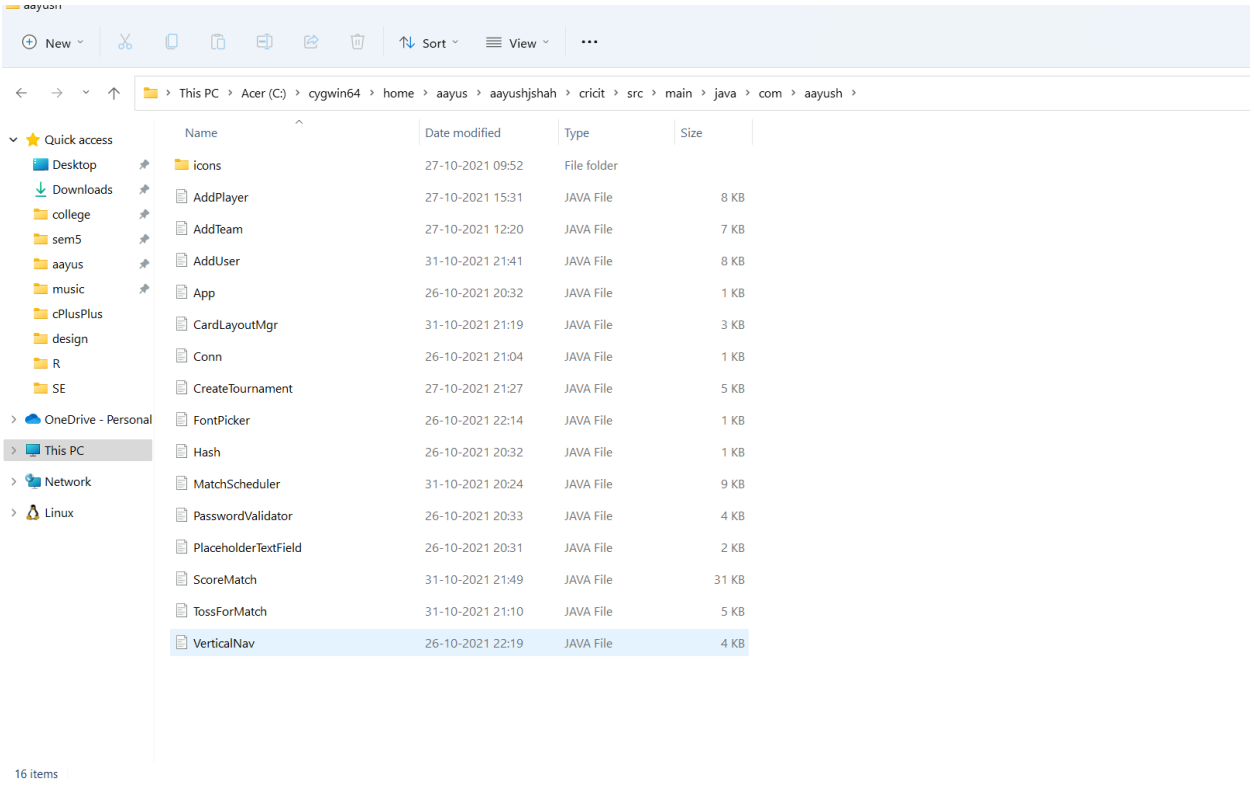
11.4.5 Schedule Matches

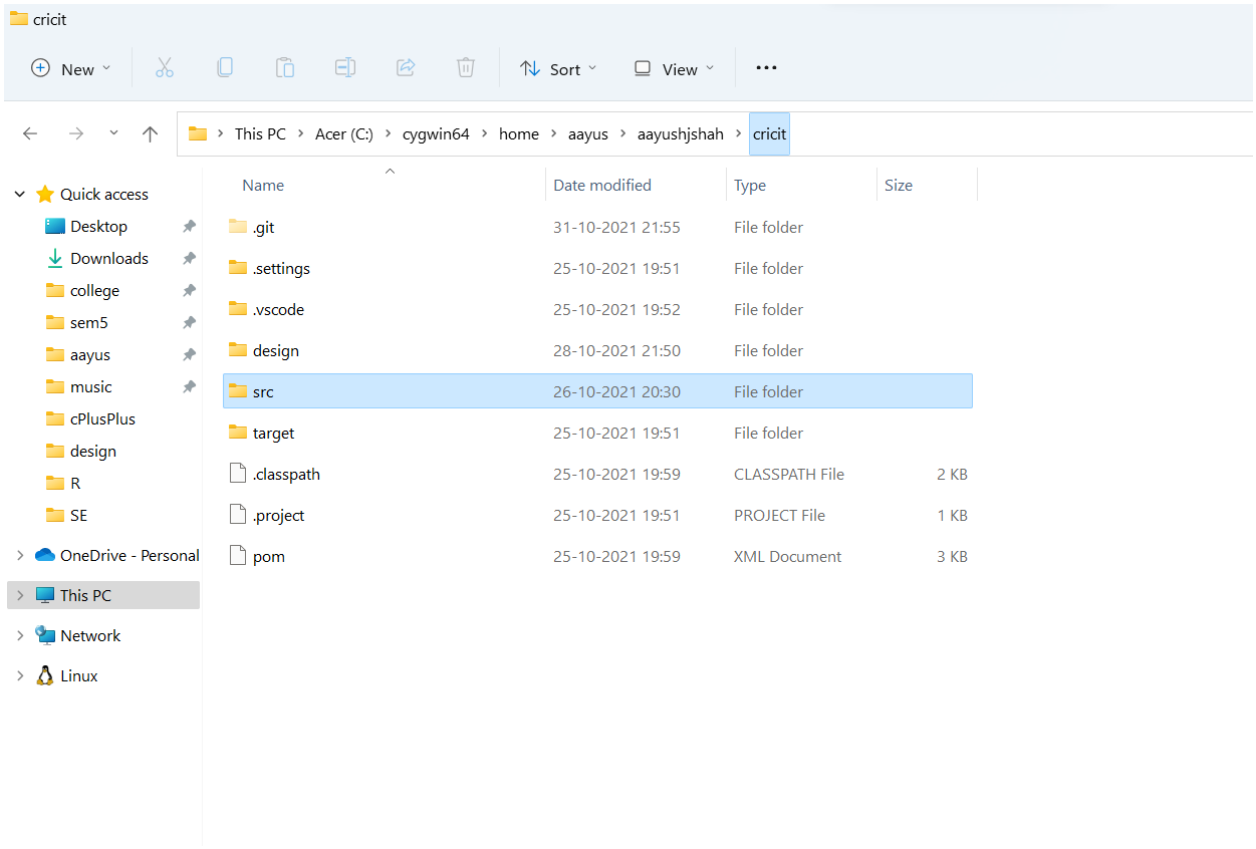


CODE SNIPPETS

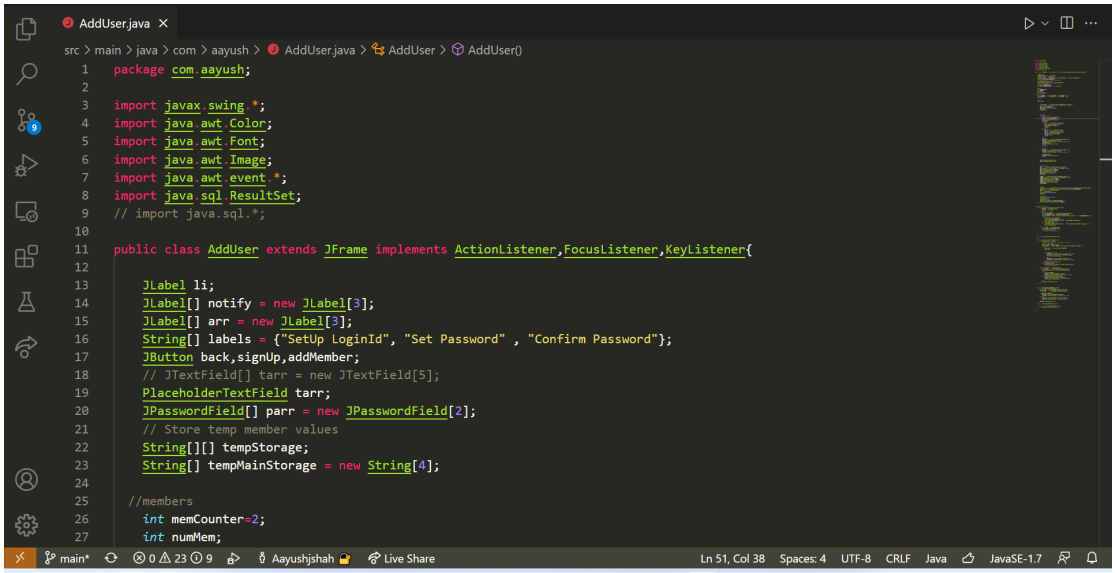
Following are the screenshots of the different files and their approximate Lines of Code (LOC).

12.1 File Structure

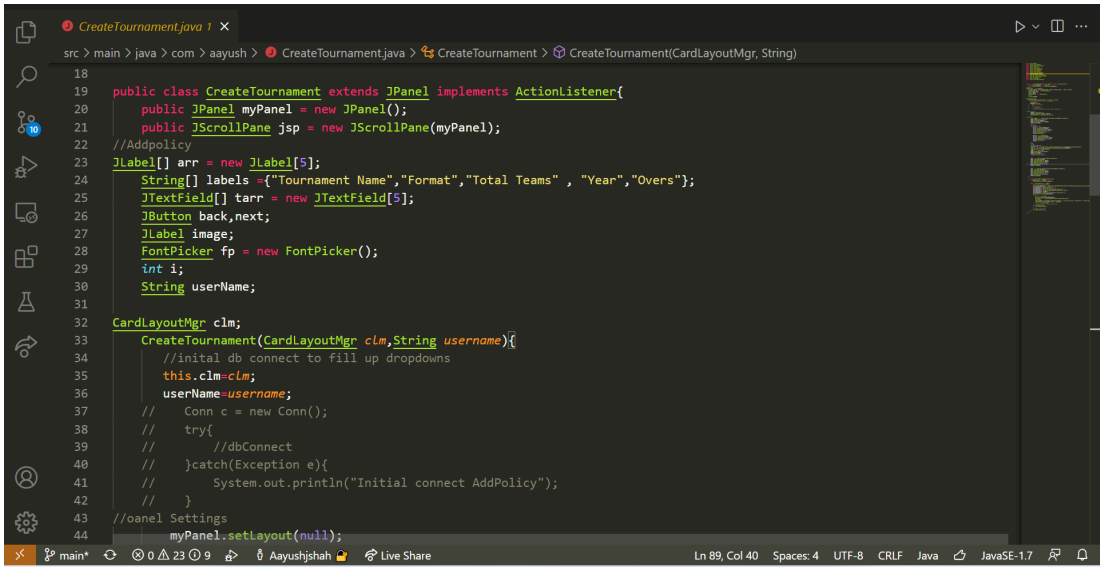




12.2 AddUser.java (LOC~216)

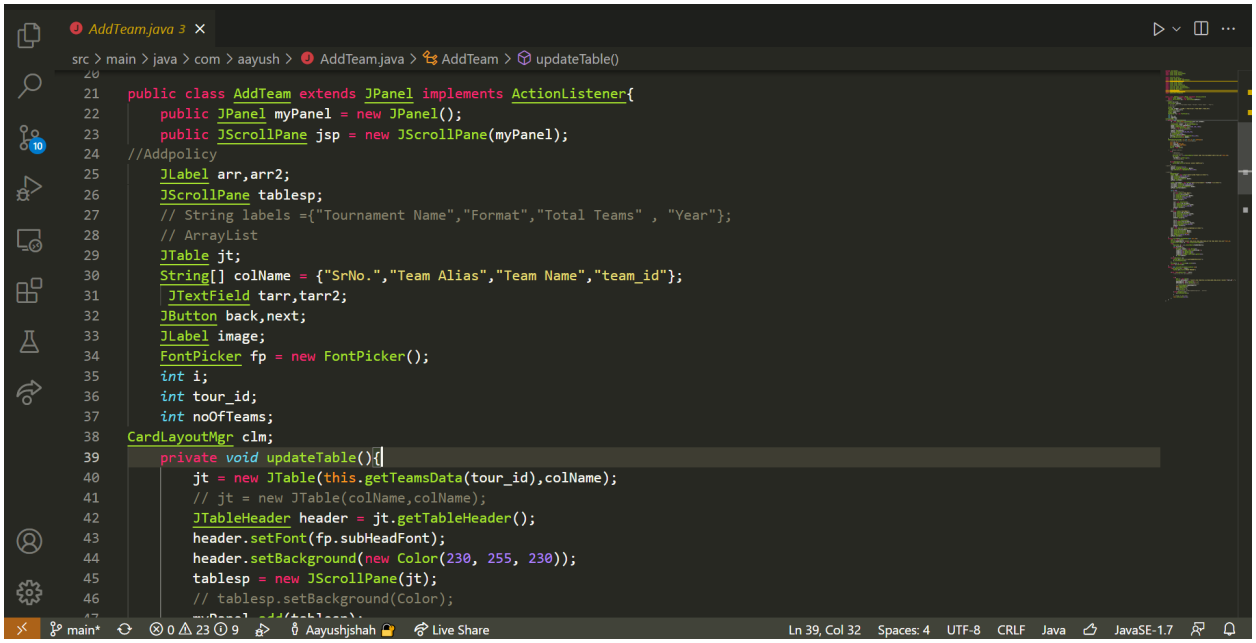


12.3 CreateTournament.java (LOC~133)



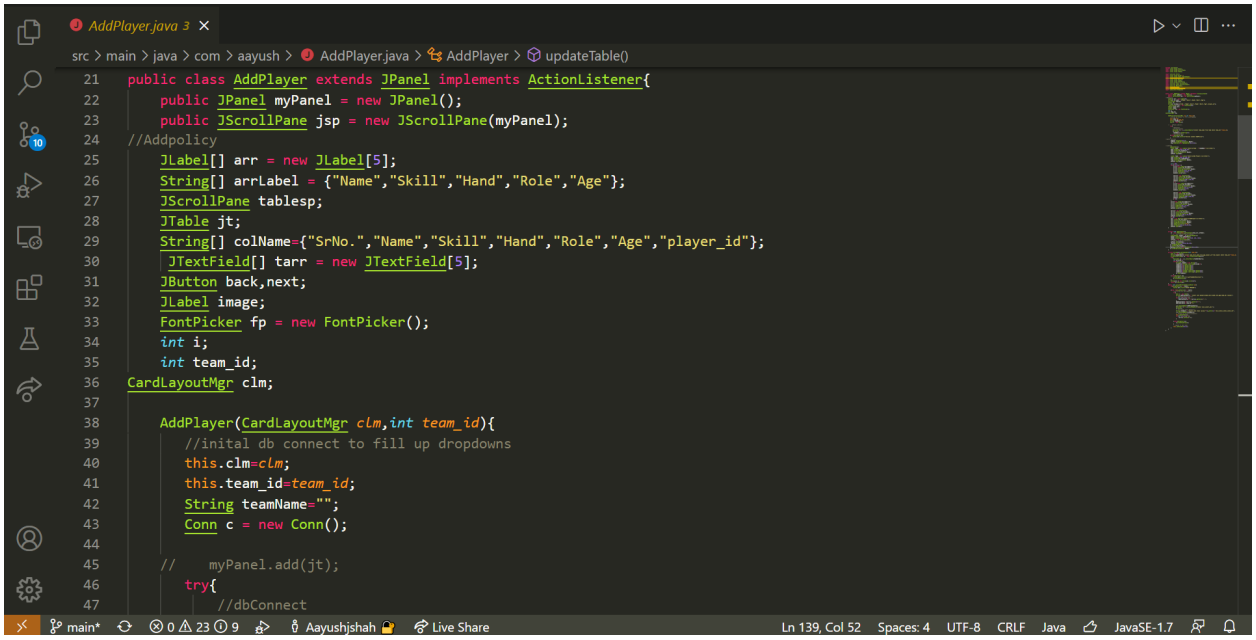
```
CreateTournament.java 1
src > main > java > com > aayush > CreateTournament.java > CreateTournament > CreateTournament(CardLayoutMgr, String)
18
19  public class CreateTournament extends JPanel implements ActionListener{
20      public JPanel myPanel = new JPanel();
21      public JScrollPane jsp = new JScrollPane(myPanel);
22  //Addpolicy
23  JLabel[] arr = new JLabel[5];
24      String[] labels ={"Tournament Name","Format","Total Teams" , "Year","Overs"};
25      JTextField[] tarr = new JTextField[5];
26      JButton back,next;
27      JLabel image;
28      FontPicker fp = new FontPicker();
29      int i;
30      String userName;
31
32  CardLayoutMgr clm;
33      CreateTournament(CardLayoutMgr clm,String username){
34          //inital db connect to fill up dropdowns
35          this.clm=clm;
36          userName=username;
37      //      Conn c = new Conn();
38      //      try{
39      //          //dbConnect
40      //      }catch(Exception e){
41      //          System.out.println("Initial connect AddPolicy");
42      //      }
43  //oanel Settings
44          myPanel.setLayout(null);
main*  0 23 9  Aayushjshah  Live Share   Ln 89, Col 40  Spaces: 4  UTF-8  CRLF  Java  JavaSE-1.7
```

12.4 AddTeam.java (LOC~179)



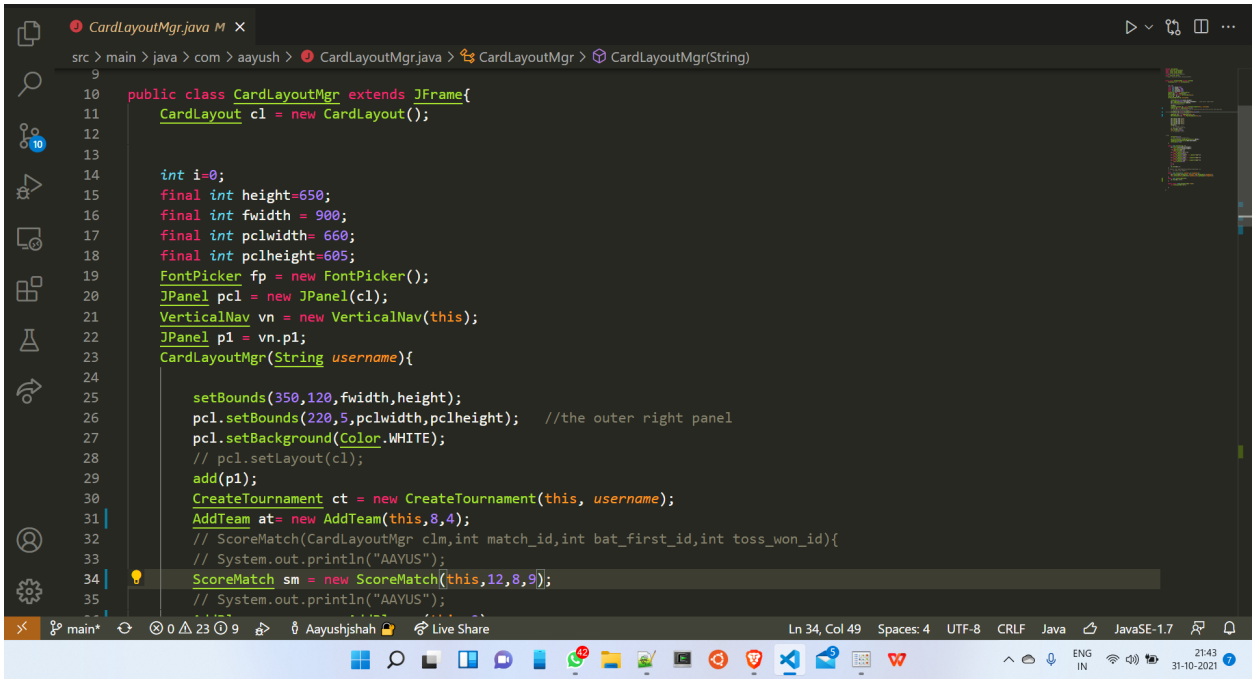
```
AddTeam.java 3
src > main > java > com > aayush > AddTeam.java > AddTeam > updateTable()
21  public class AddTeam extends JPanel implements ActionListener{
22      public JPanel myPanel = new JPanel();
23      public JScrollPane jsp = new JScrollPane(myPanel);
24  //Addpolicy
25      JLabel arr,arr2;
26      JScrollPane tablesp;
27      // String labels ={"Tournament Name","Format","Total Teams" , "Year"};
28      // ArrayList
29      JTable jt;
30      String[] colName = {"SrNo.","Team Alias","Team Name","team_id"};
31       JTextField tarr,tarr2;
32      JButton back,next;
33      JLabel image;
34      FontPicker fp = new FontPicker();
35      int i;
36      int tour_id;
37      int noOfTeams;
38  CardLayoutMgr clm;
39      private void updateTable(){
40          jt = new JTable(this.getTeamsData(tour_id),colName);
41          // jt = new JTable(colName,colName);
42          JTableHeader header = jt.getTableHeader();
43          header.setFont(fp.subHeadFont);
44          header.setBackground(new Color(230, 255, 230));
45          tablesp = new JScrollPane(jt);
46          // tablesp.setBackground(Color);
main*  0 23 9  Aayushjshah  Live Share   Ln 39, Col 32  Spaces: 4  UTF-8  CRLF  Java  JavaSE-1.7
```

12.5 AddPlayer.java (LOC~204)



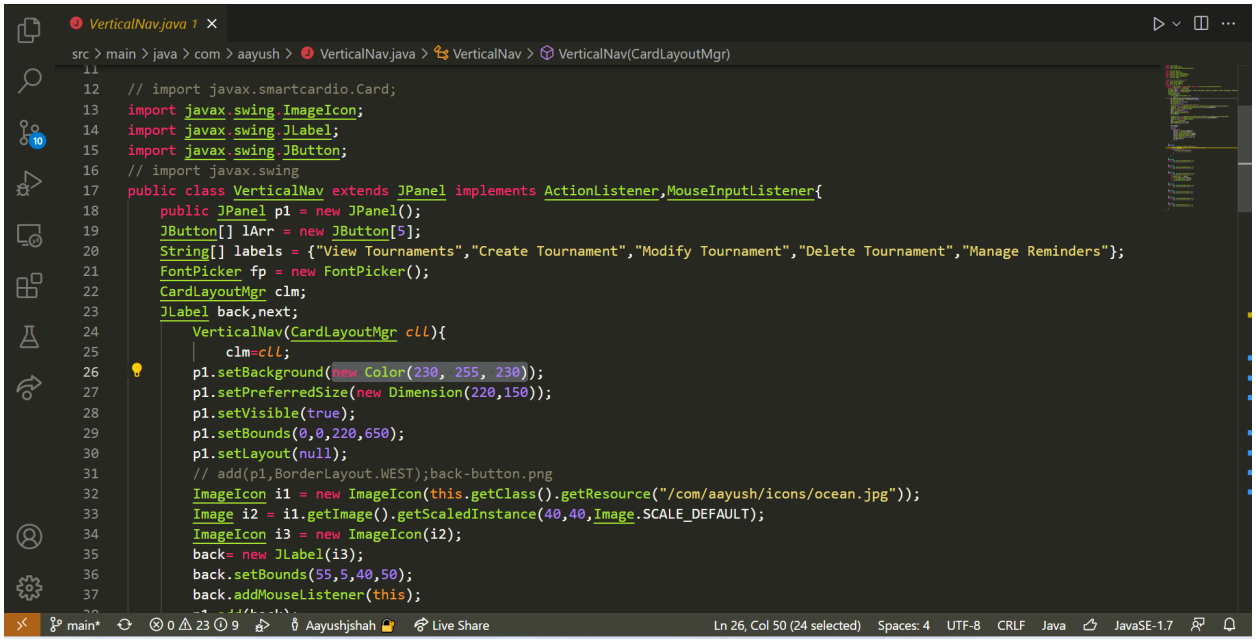
```
AddPlayer.java 3
src > main > java > com > aayush > AddPlayer.java > AddPlayer > updateTable()
21  public class AddPlayer extends JPanel implements ActionListener{
22      public JPanel myPanel = new JPanel();
23      public JScrollPane jsp = new JScrollPane(myPanel);
24  //Addpolicy
25      JLabel[] arr = new JLabel[5];
26      String[] arrLabel = {"Name","Skill","Hand","Role","Age"};
27      JScrollPane tablesp;
28      JTable jt;
29      String[] colName={"SrNo.","Name","Skill","Hand","Role","Age","player_id"};
30       JTextField[] tarr = new JTextField[5];
31      JButton back,next;
32      JLabel image;
33      FontPicker fp = new FontPicker();
34      int i;
35      int team_id;
36  CardLayoutMgr clm;
37
38      AddPlayer(CardLayoutMgr clm,int team_id){
39        //inital db connect to fill up dropdowns
40        this.clm=clm;
41        this.team_id=team_id;
42        String teamName="";
43        Conn c = new Conn();
44
45  //    myPanel.add(jt);
46        try{
47          //dbConnect
main*  0 23 9  Aayushjshah  Live Share   Ln 139, Col 52  Spaces: 4  UTF-8  CRLF  Java  JavaSE-1.7
```

12.6 CardLayoutMgr.java (LOC~95)



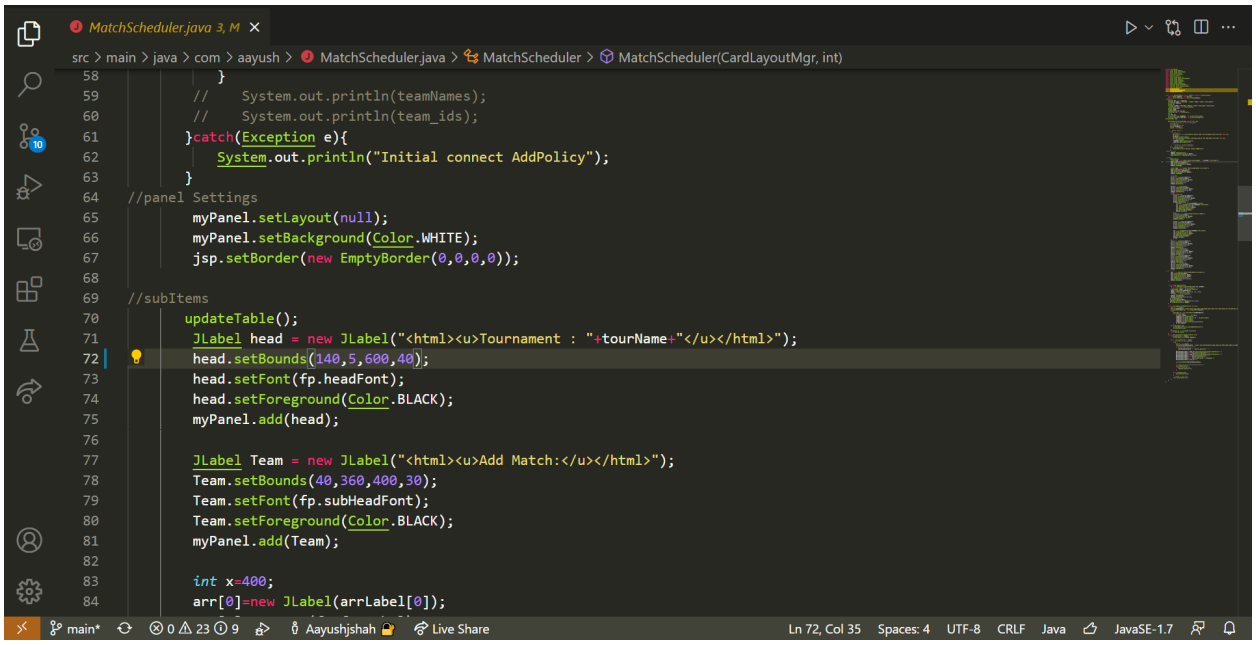
```
CardLayoutMgr.java M
src > main > java > com > aayush > CardLayoutMgr.java > CardLayoutMgr > CardLayoutMgr(String)
10  public class CardLayoutMgr extends JFrame{
11      CardLayout cl = new CardLayout();
12
13
14      int i=0;
15      final int height=650;
16      final int fwidth = 900;
17      final int pclwidth= 660;
18      final int pclheight=605;
19      FontPicker fp = new FontPicker();
20      JPanel pcl = new JPanel(cl);
21      VerticalNav vn = new VerticalNav(this);
22      JPanel p1 = vn.p1;
23      CardLayoutMgr(String username){
24
25          setBounds(350,120,fwidth,height);
26          pcl.setBounds(220,5,pclwidth,pclheight);   //the outer right panel
27          pcl.setBackground(Color.WHITE);
28          // pcl.setLayout(cl);
29          add(p1);
30          CreateTournament ct = new CreateTournament(this, username);
31          AddTeam at= new AddTeam(this,8,4);
32          // ScoreMatch(CardLayoutMgr clm,int match_id,int bat_first_id,int toss_won_id){
33          // System.out.println("AAYUS");
34          ScoreMatch sm = new ScoreMatch(this,12,8,9);
35          // System.out.println("AAYUS");
main*  0 23 9  Aayushjshah  Live Share   Ln 34, Col 49  Spaces: 4  UTF-8  CRLF  Java  JavaSE-1.7
```

12.7 VerticalNav.java (LOC~113)



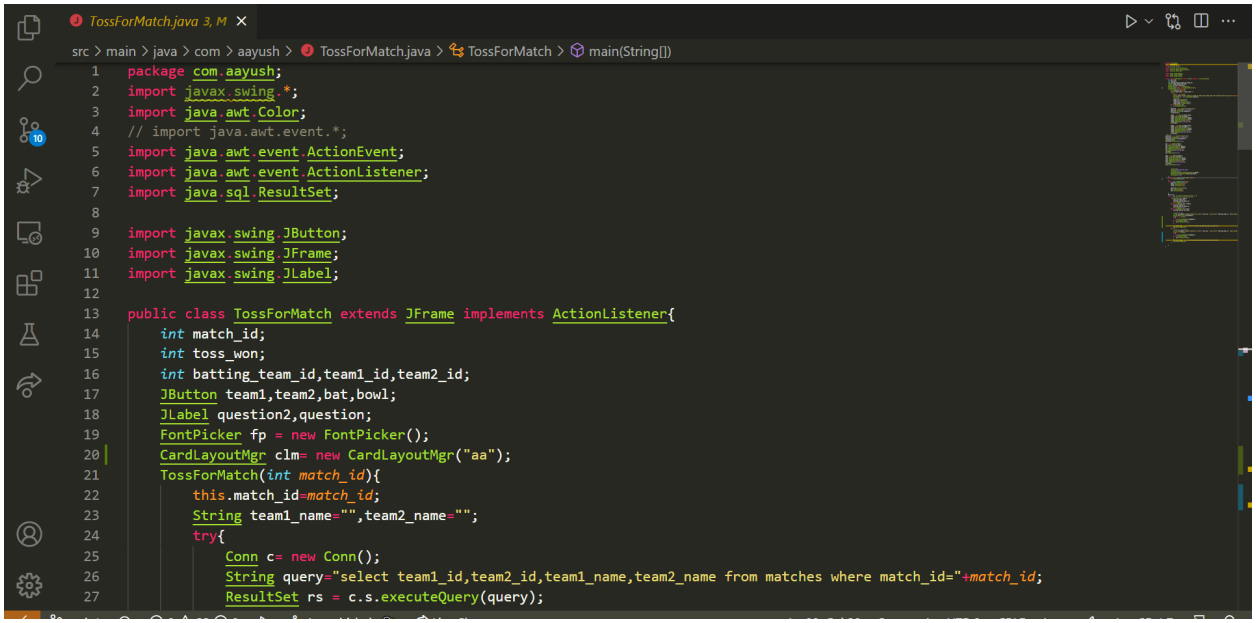
```
// import javax.smartcardio.Card;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JButton;
// import javax.swing
public class VerticalNav extends JPanel implements ActionListener,MouseInputListener{
    public JPanel p1 = new JPanel();
    JButton[] lArr = new JButton[5];
    String[] labels = {"View Tournaments","Create Tournament","Modify Tournament","Delete Tournament","Manage Reminders"};
    FontPicker fp = new FontPicker();
    CardLayoutMgr clm;
    JLabel back,next;
        VerticalNav(CardLayoutMgr cll){
            clm=cll;
        p1.setBackground(new Color(230, 255, 230));
        p1.setPreferredSize(new Dimension(220,150));
        p1.setVisible(true);
        p1.setBounds(0,0,220,650);
        p1.setLayout(null);
        // add(p1,BorderLayout.WEST);back-button.png
        ImageIcon i1 = new ImageIcon(this.getClass().getResource("/com/aayush/icons/ocean.jpg"));
        Image i2 = i1.getImage().getScaledInstance(40,40,Image.SCALE_DEFAULT);
        ImageIcon i3 = new ImageIcon(i2);
        back= new JLabel(i3);
        back.setBounds(55,5,40,50);
        back.addMouseListener(this);
```

12.8 MatchScheduler.java (LOC~242)

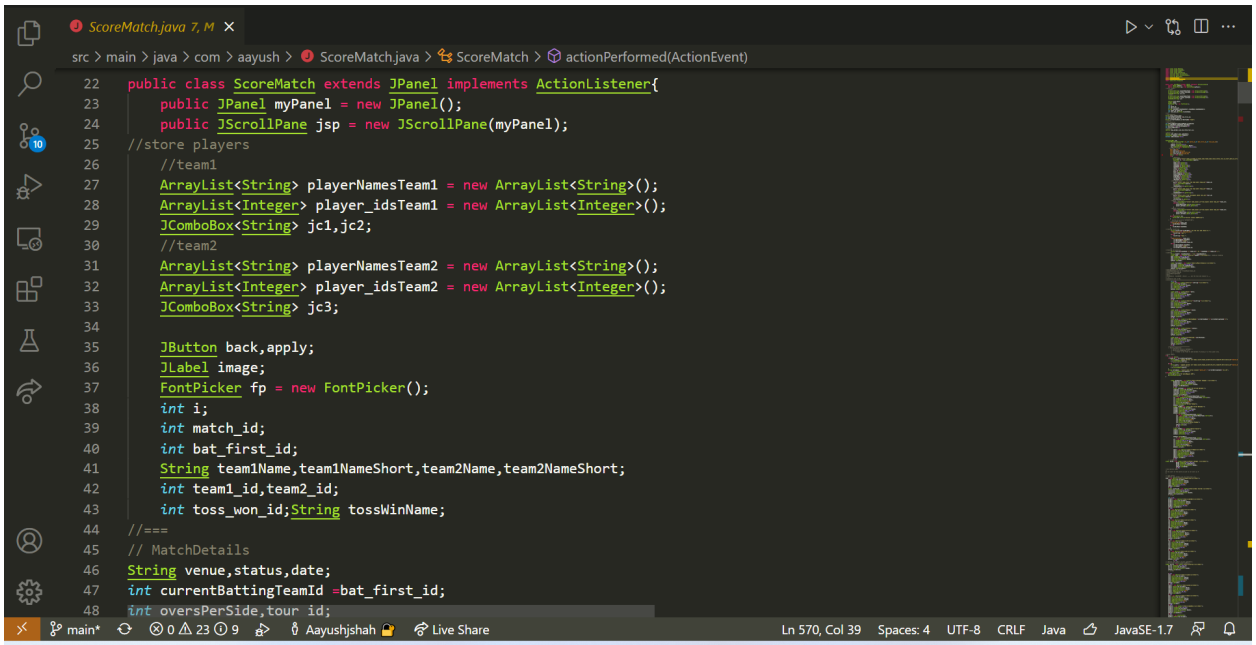


```
            }
          //    System.out.println(teamNames);
          //    System.out.println(team_ids);
        }catch(Exception e){
            System.out.println("Initial connect AddPolicy");
        }
//panel Settings
        myPanel.setLayout(null);
        myPanel.setBackground(Color.WHITE);
        jsp.setBorder(new EmptyBorder(0,0,0,0));

//subItems
       updateTable();
        JLabel head = new JLabel("<html><u>Tournament : "+tourName+"</u></html>");
        head.setBounds(140,5,600,40);
        head.setFont(fp.headFont);
        head.setForeground(Color.BLACK);
        myPanel.add(head);

        JLabel Team = new JLabel("<html><u>Add Match:</u></html>");
        Team.setBounds(40,360,400,30);
        Team.setFont(fp.subHeadFont);
        Team.setForeground(Color.BLACK);
        myPanel.add(Team);

        int x=400;
        arr[0]=new JLabel(arrLabel[0]);
```

12.9 TossForMatch.java (LOC~143)

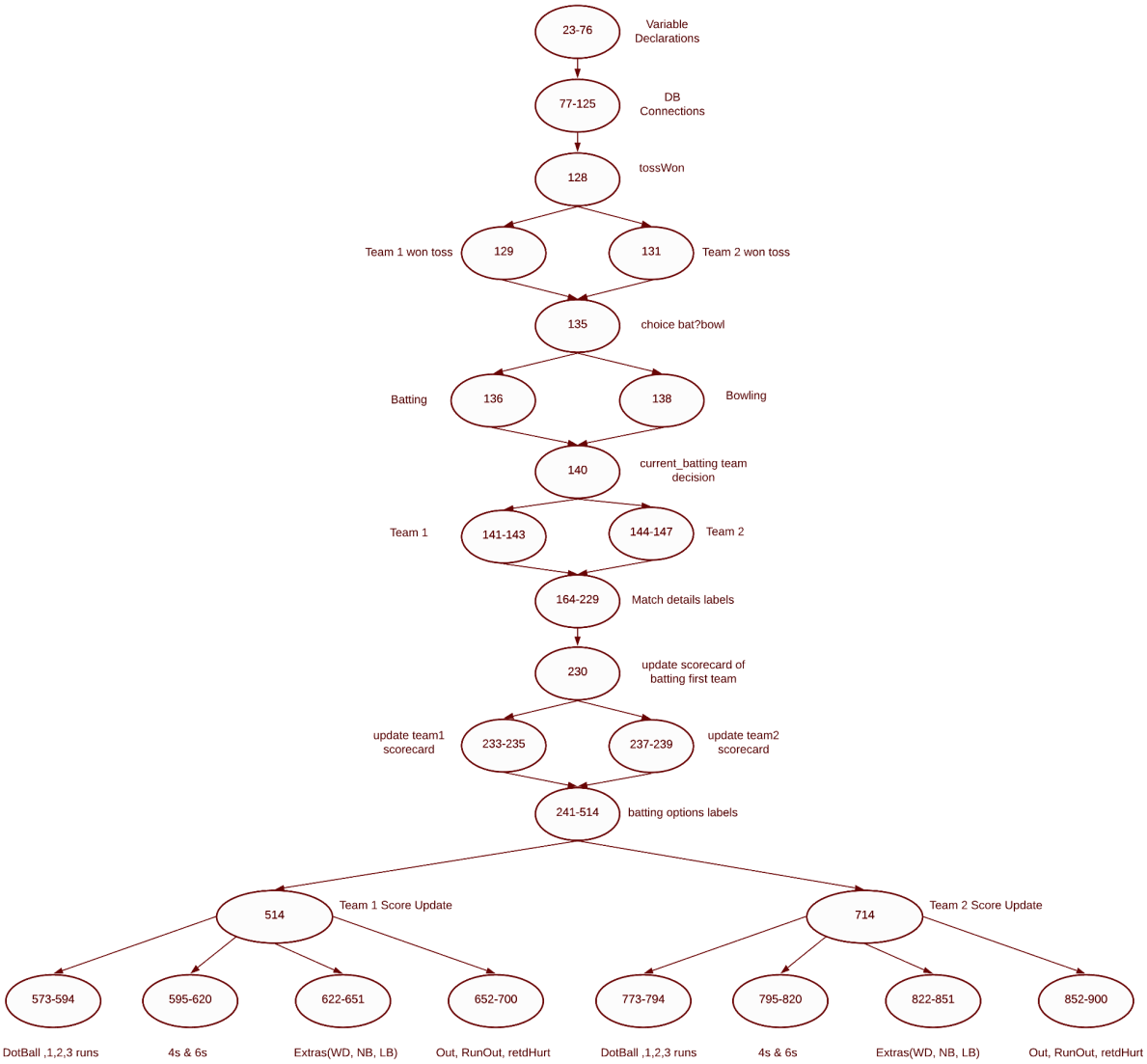


12.10 ScoreMarch.java (LOC ~ 782)



SOFTWARE TESTING

13.1 Flowchart



==========

Similarly, we have built flowchart for adding and modifying tournaments from the respective code files.

13.2 Calculation of CycloMatic Complexity

V(G)= No. of edges - No. of nodes +2^2
No. of Edges = 29
No. of Nodes = 26

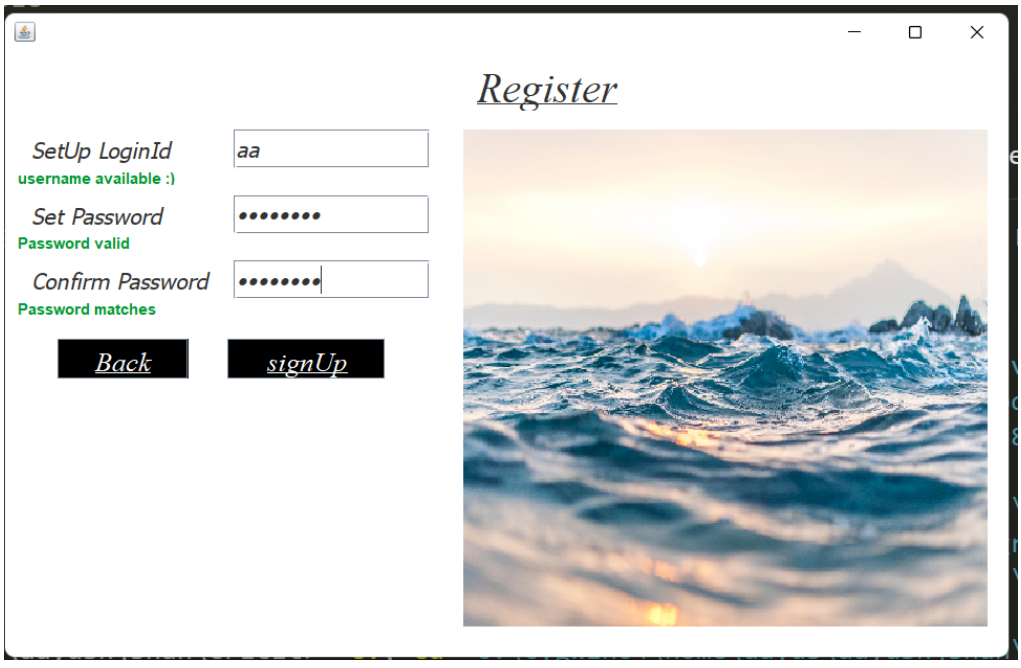
Therefore V(G) = 29-26+4
= 7
CycloMatic Complexity :: V(G) = 7

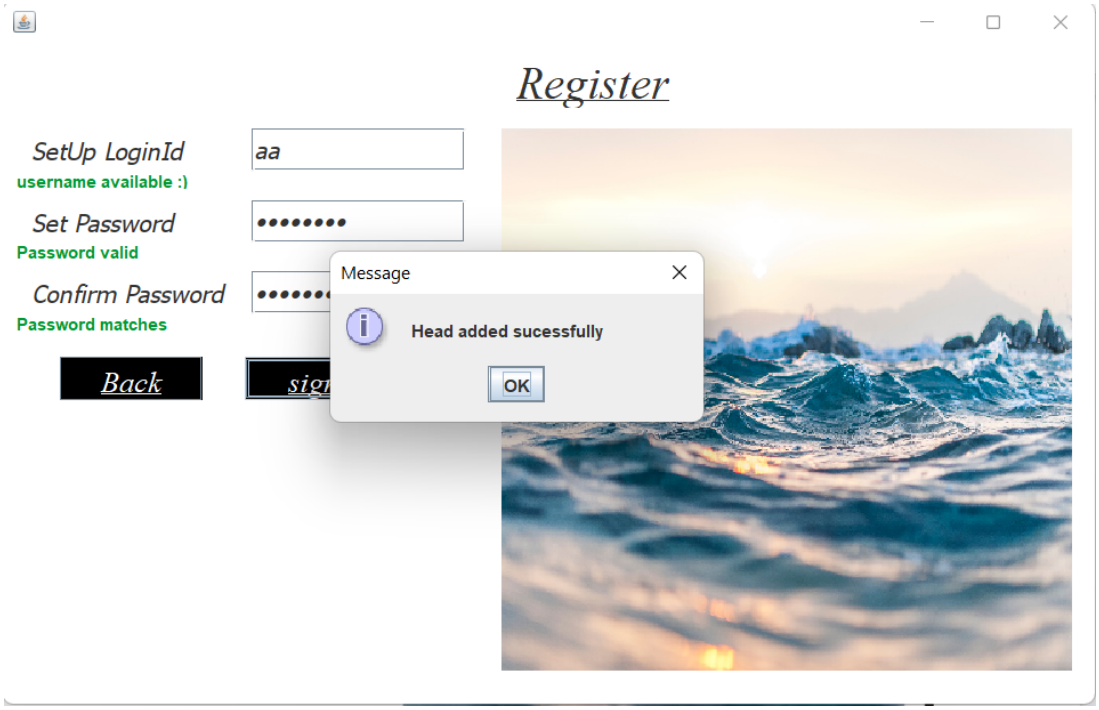
13.3 Test Cases

Test cases for different features implemented in our desktop application using Java Panels are listed below.

13.3.1 User Registration

- Adding a new user





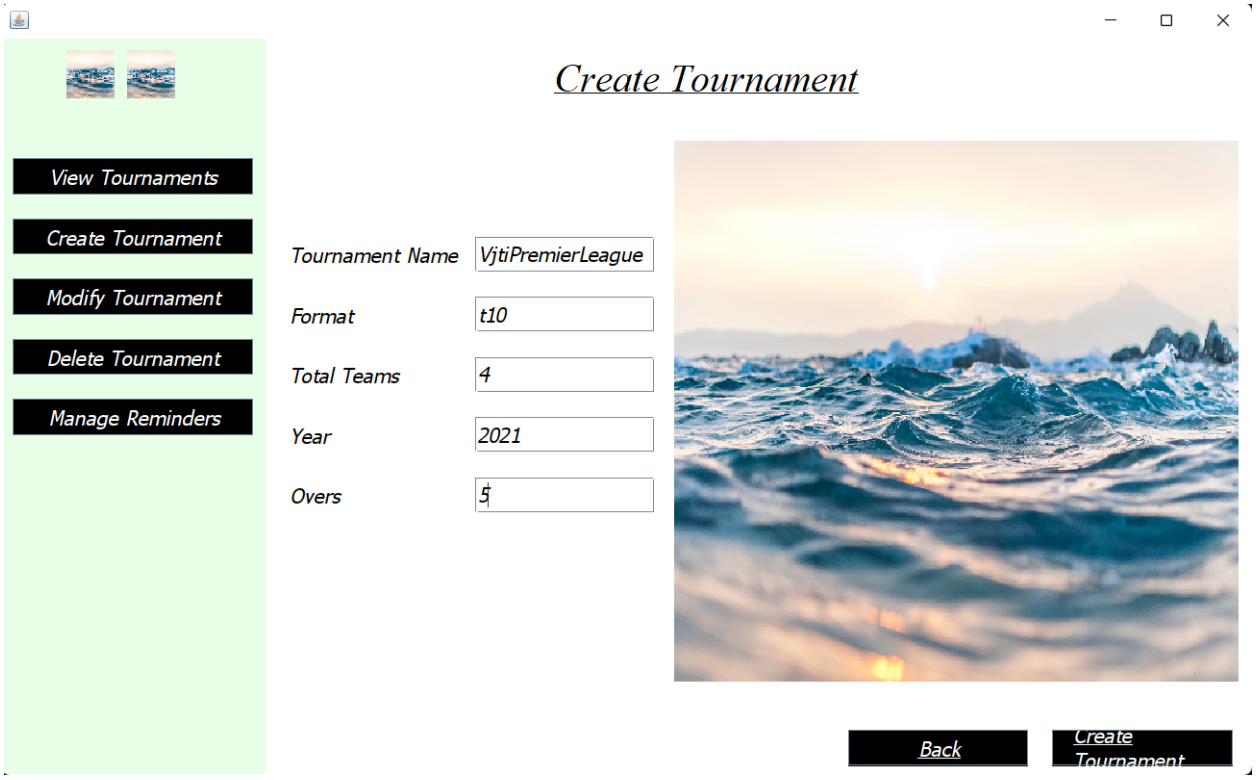
- User table before and after adding a new user

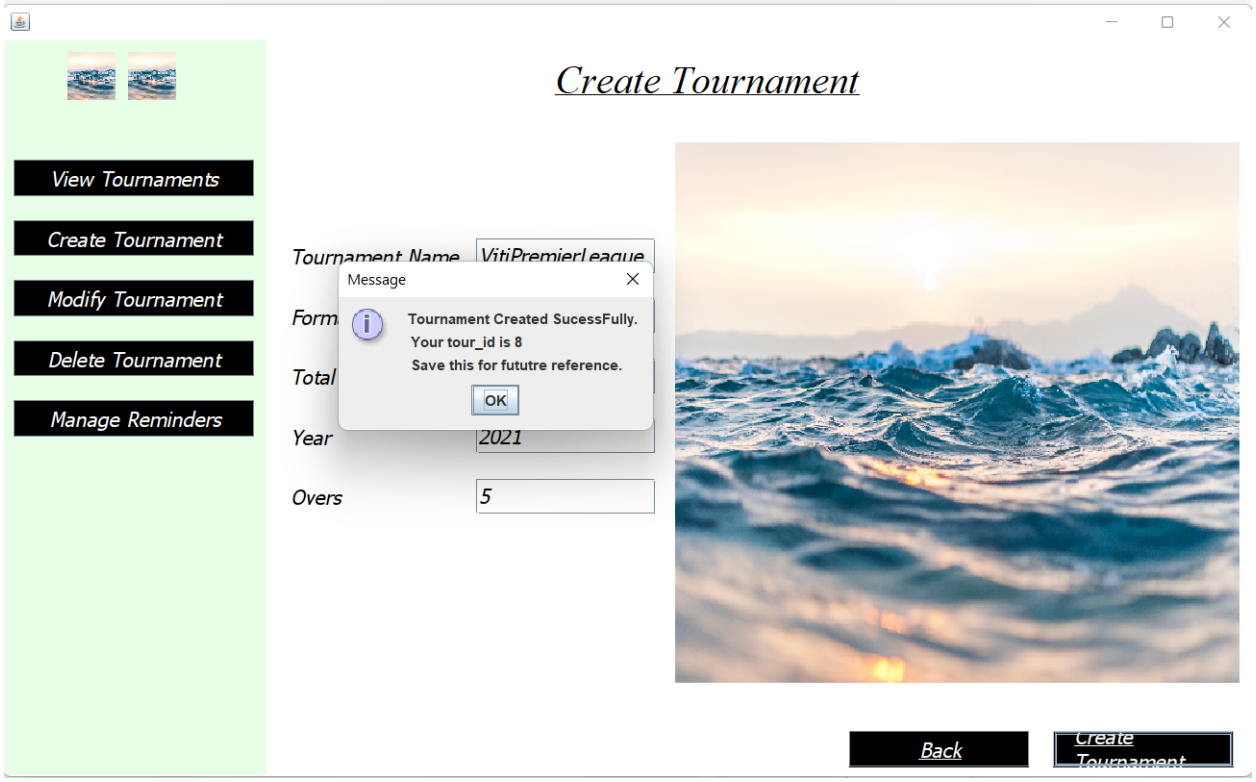
```
mysql> select * from user;
+----------+----------------------------------+
| username | password                         |
+----------+----------------------------------+
| A        | b81e8818bfcaa728e20869d97e1905cc |
+----------+----------------------------------+
1 row in set (0.00 sec)

mysql> select * from user;
+----------+----------------------------------+
| username | password                         |
+----------+----------------------------------+
| A        | b81e8818bfcaa728e20869d97e1905cc |
| aa       | dc069fbb0779ea17832a64878c738bb4 |
+----------+----------------------------------+
2 rows in set (0.00 sec)

mysql>
```

- Expected output: The details of the user added should be reflected in the user table.
- Result: Test Case Passed

13.3.2 Create Tournament

- Adding a new tournament



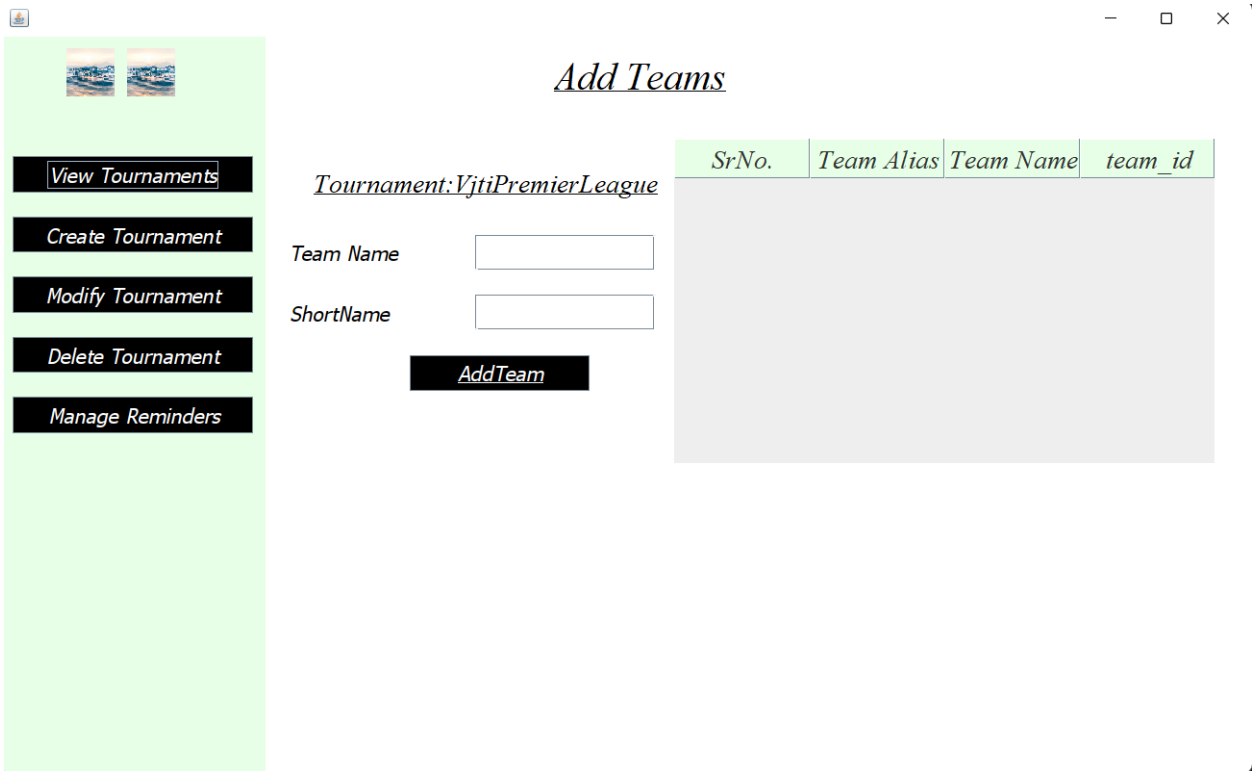


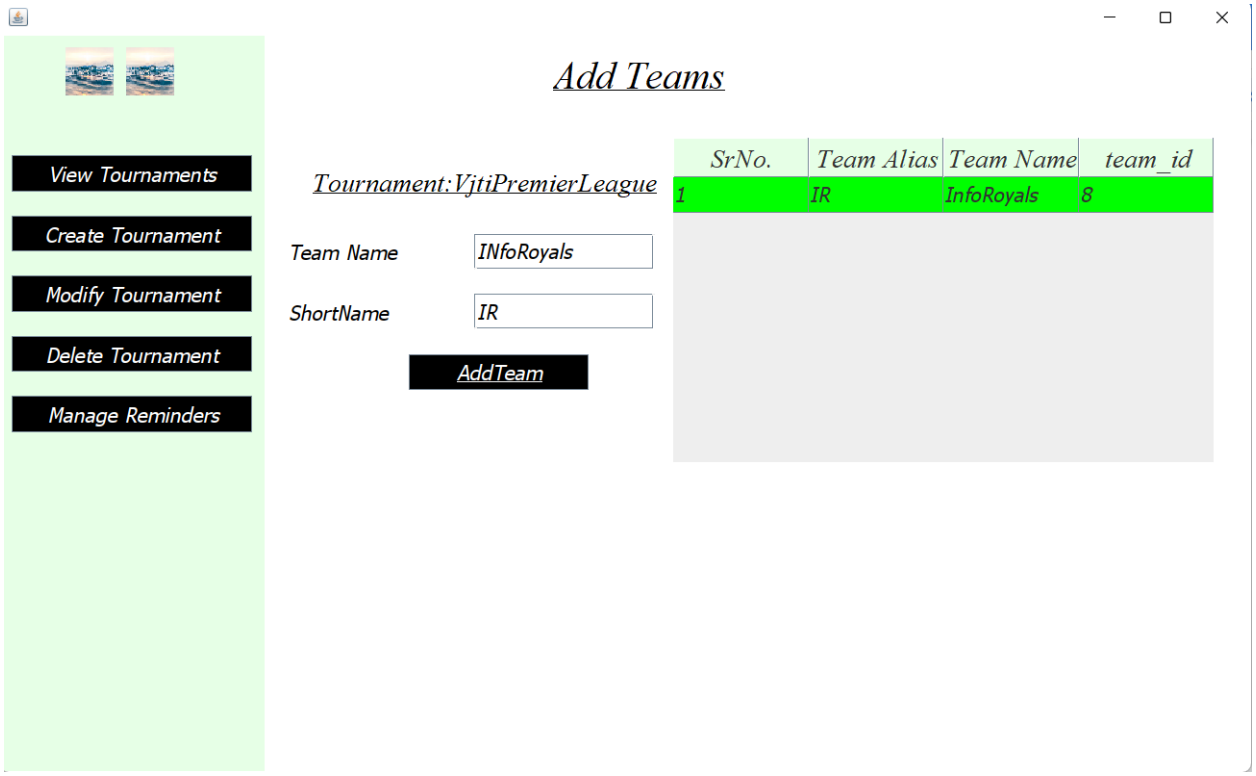
- Tournament table before and after adding a new tournament.

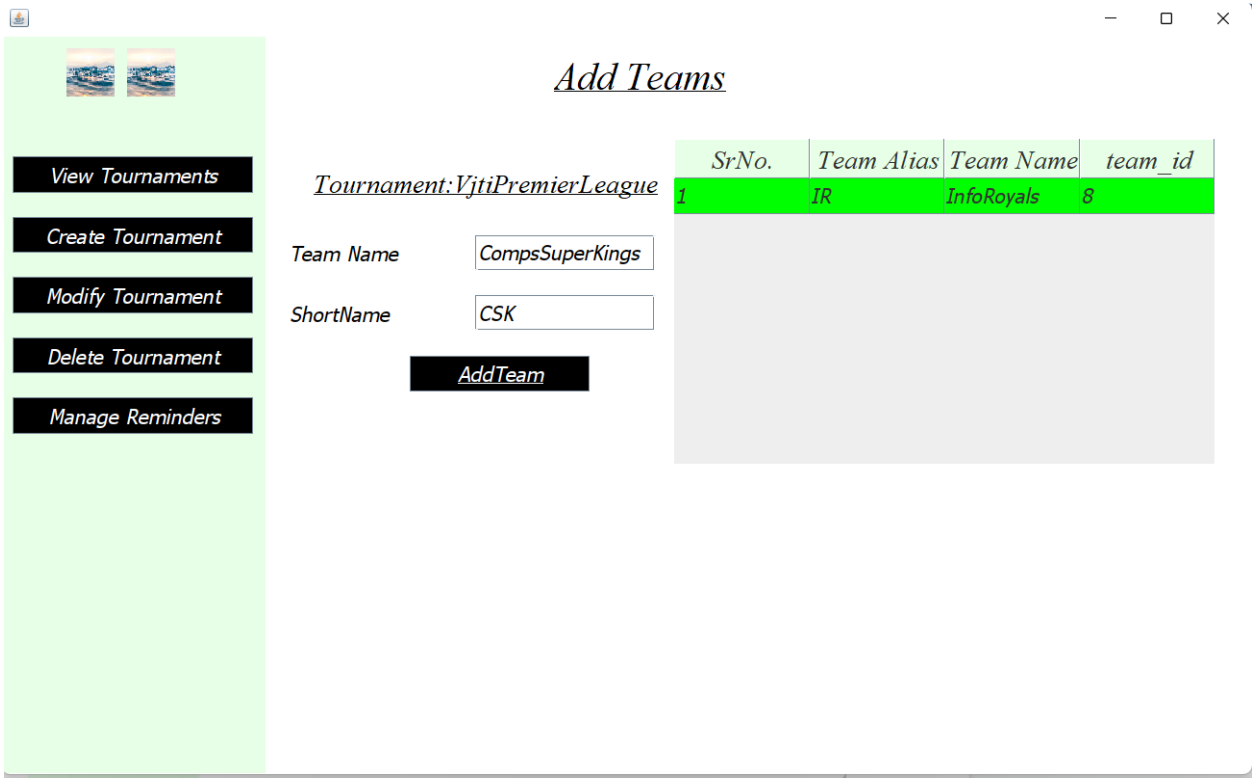
```
mysql> select * from tournament;
+---------+-------------+--------+-----------+------+-------------+---------+-------+
| tour_id | name        | format | tot_teams | year | team_won_id | user_id | overs |
+---------+-------------+--------+-----------+------+-------------+---------+-------+
|       1 | mohanPalace | t20    |         8 | 2021 |           1 | 2       |    20 |
|       2 | Ipl         | t20    |         8 | 2021 |        NULL | aa      |    20 |
|       6 | cpl         | t10    |         4 | 2022 |        NULL | aa      |     5 |
|       7 | dpl         | t4     |         3 | 2021 |        NULL | aa      |     5 |
+---------+-------------+--------+-----------+------+-------------+---------+-------+
4 rows in set (0.01 sec)

mysql> select * from tournament;
+---------+-------------------+--------+-----------+------+-------------+---------+-------+
| tour_id | name              | format | tot_teams | year | team_won_id | user_id | overs |
+---------+-------------------+--------+-----------+------+-------------+---------+-------+
|       1 | mohanPalace       | t20    |         8 | 2021 |           1 | 2       |    20 |
|       2 | Ipl               | t20    |         8 | 2021 |        NULL | aa      |    20 |
|       6 | cpl               | t10    |         4 | 2022 |        NULL | aa      |     5 |
|       7 | dpl               | t4     |         3 | 2021 |        NULL | aa      |     5 |
|       8 | VjtiPremierLeague | t10    |         4 | 2021 |        NULL | aa      |     5 |
+---------+-------------------+--------+-----------+------+-------------+---------+-------+
5 rows in set (0.00 sec)

mysql>
```

- Expected output: The details of the tournament added must be reflected in the tournament table.
- Result: Test Case Passed

13.3.3 Add Teams




Add Teams
View Tournaments
Create Tournament
Modify Tournament
Delete Tournament
Manage Reminders
Tournament:VjtiPremierLeague
Team Name
INfoRoyals
ShortName
IR
AddTeam
SrNo.
Team Alias
Team Name
team_id
1
IR
InfoRoyals
8


Add Teams
View Tournaments
Create Tournament
Modify Tournament
Delete Tournament
Manage Reminders
Tournament:VjtiPremierLeague
Team Name
CompsSuperKings
ShortName
CSK
AddTeam
SrNo.
Team Alias
Team Name
team_id
1
IR
InfoRoyals
8


Add Teams
View Tournaments
Create Tournament
Modify Tournament
Delete Tournament
Manage Reminders
Tournament:VjtiPremierLeague
Team Name
ShortName
AddTeam
SrNo.
Team Alias
Team Name
team_id
1
IR
InfoRoyals
8
2
CSK
CompsSuper...
9
3
ML
MechLions
10
4
TT
TextileTuskers
11

```
mysql> select * from team;
+---------+---------+---------------------+------------+
| team_id | tour_id | team_name           | team_alias |
+---------+---------+---------------------+------------+
|       3 |       2 | Chennai Super Kings | CSK        |
|       4 |       2 | Mumbai Indians      | MI         |
|       5 |       2 | Kolkata             | KKR        |
|       6 |       7 | Tamil Rockers       | TR         |
|       7 |       7 | Maha R              | MR         |
+---------+---------+---------------------+------------+
5 rows in set (0.00 sec)

mysql> select * from team where tour_id=8;
Empty set (0.00 sec)

mysql> select * from team where tour_id=8;
+---------+---------+-----------------+------------+
| team_id | tour_id | team_name       | team_alias |
+---------+---------+-----------------+------------+
|       8 |       8 | InfoRoyals      | IR         |
|       9 |       8 | CompsSuperKings | CSK        |
|      10 |       8 | MechLions       | ML         |
|      11 |       8 | TextileTuskers  | TT         |
+---------+---------+-----------------+------------+
4 rows in set (0.00 sec)

mysql>
```

(Before and After : table teams)

13.3.4 Add Players

- Adding Players to team InfoRoyals

- players table before and after adding players to their respective teams.

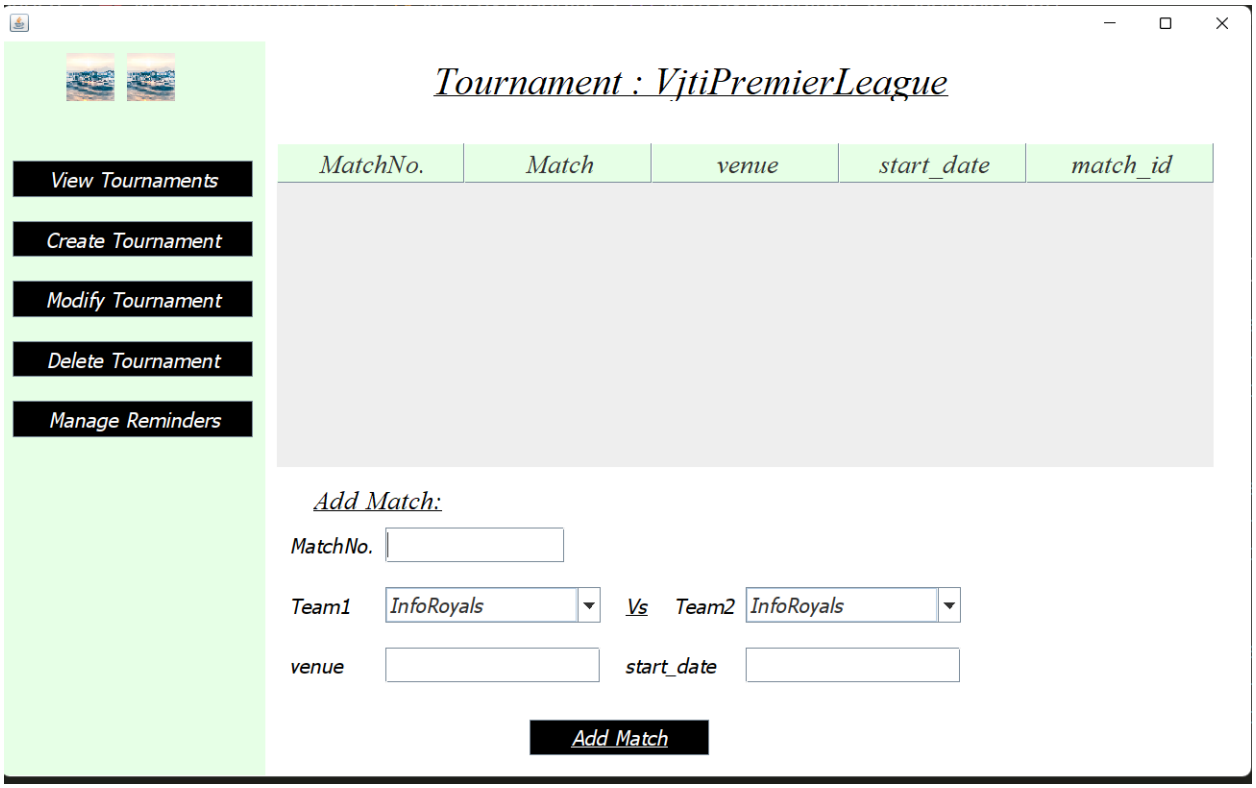
```
mysql> select * from players;
+-----------+-----------------+---------------+-------+---------+------+--------------+
| player_id | name            | skill         | hand  | team_id | age  | role         |
+-----------+-----------------+---------------+-------+---------+------+--------------+
|         1 | Dhoni           | Batsmen       | Right |       3 |   37 | Captain      |
|         2 | Jadeja          | All-rounder   | Left  |       3 |   30 | Player       |
|         3 | Raina           | Batsmen       | Left  |       3 |   33 | Vice-Captain |
|         4 | Ruturaj Gaikwad | Batsmen       | Right |       3 |   24 | Player       |
|         5 | Faf DU plesis   | Batsmen       | Right |       3 |   35 | Batsmen      |
|         6 | Rohit Sharma    | Batsmen       | Right |       4 |   33 | Captain      |
|         7 | Hardik Pandya   | All-rounder   | Right |       4 |   27 | Player       |
|         8 | Ishan Kishan    | Batsmen-keeper| left  |       4 |   25 | Player       |
|         9 | Surya           | Batsmen       | Right |       4 |   33 | Player       |
|        10 | Bumrah          | Bowler        | Right |       4 |   29 | ViceCaptain  |
|        11 | Boult           | Bowler        | Right |       4 |   32 | Player       |
+-----------+-----------------+---------------+-------+---------+------+--------------+
11 rows in set (0.00 sec)

mysql> select * from players where team_id=8;
Empty set (0.00 sec)

mysql> select * from players where team_id=8;
+-----------+-----------+-------------+-------+---------+------+--------------+
| player_id | name      | skill       | hand  | team_id | age  | role         |
+-----------+-----------+-------------+-------+---------+------+--------------+
|        12 | Aayush    | Batsmen     | Left  |       8 |   20 | Player       |
|        13 | Meet      | Bowler      | Left  |       8 |   20 | Captain      |
|        14 | Mann Doshi| All rounder | Right |       8 |   20 | Player       |
|        15 | Parth     | WicketKeeper| Right |       8 |   20 | Vice-Captain |
+-----------+-----------+-------------+-------+---------+------+--------------+
4 rows in set (0.00 sec)

mysql>
```

```
mysql> select * from players where team_id=9;
+-----------+----------------+-------------+-------+---------+------+-------------+
| player_id | name           | skill       | hand  | team_id | age  | role        |
+-----------+----------------+-------------+-------+---------+------+-------------+
|        16 | Bhavya         | Bowler      | Right |       9 |   20 | Captain     |
|        17 | Nirmit         | Batsmen     | Right |       9 |   20 | Vice-Captain |
|        18 | Donovan Crasta | All-rounder | Right |       9 |   20 | Player      |
|        19 | Kushal         | Bowler      | Right |       9 |   20 | player      |
+-----------+----------------+-------------+-------+---------+------+-------------+
4 rows in set (0.00 sec)

mysql>
```

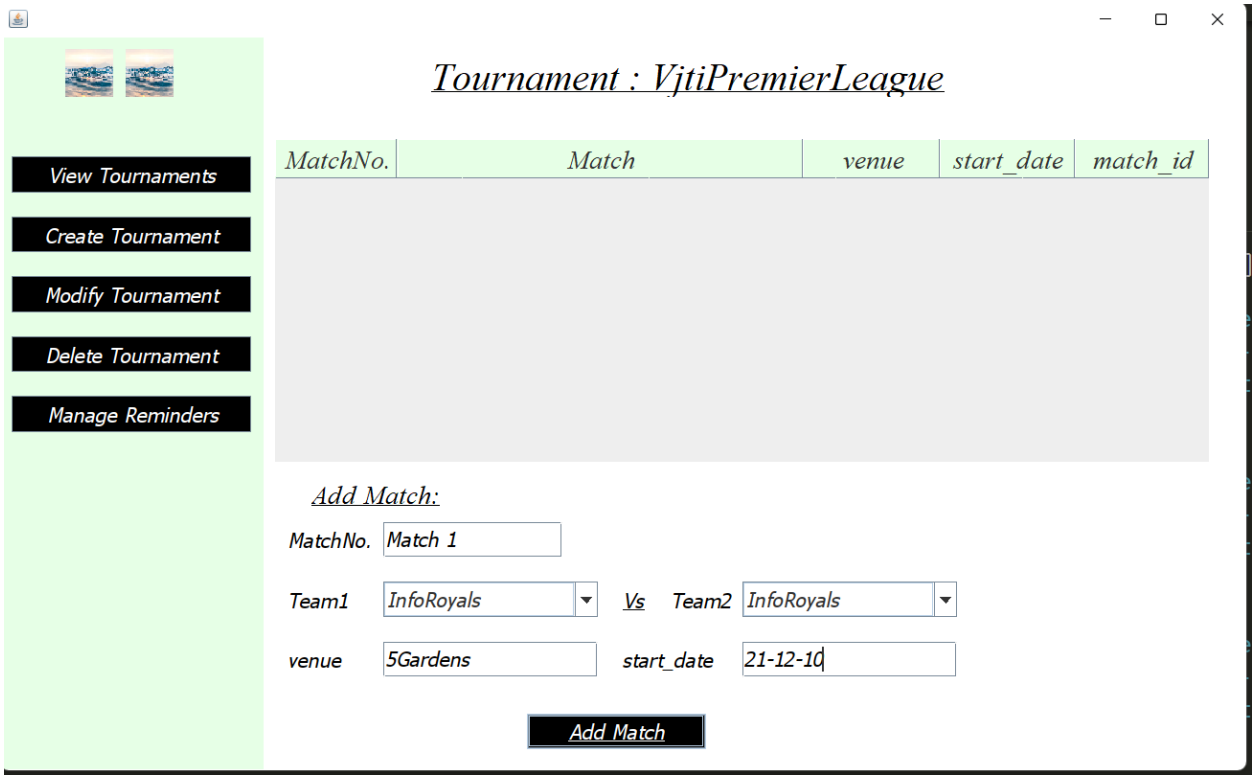
- Expected Output: The details of all the players added must be reflected in the player table.
- Result: Test Case Passed

13.3.5 Schedule Matches

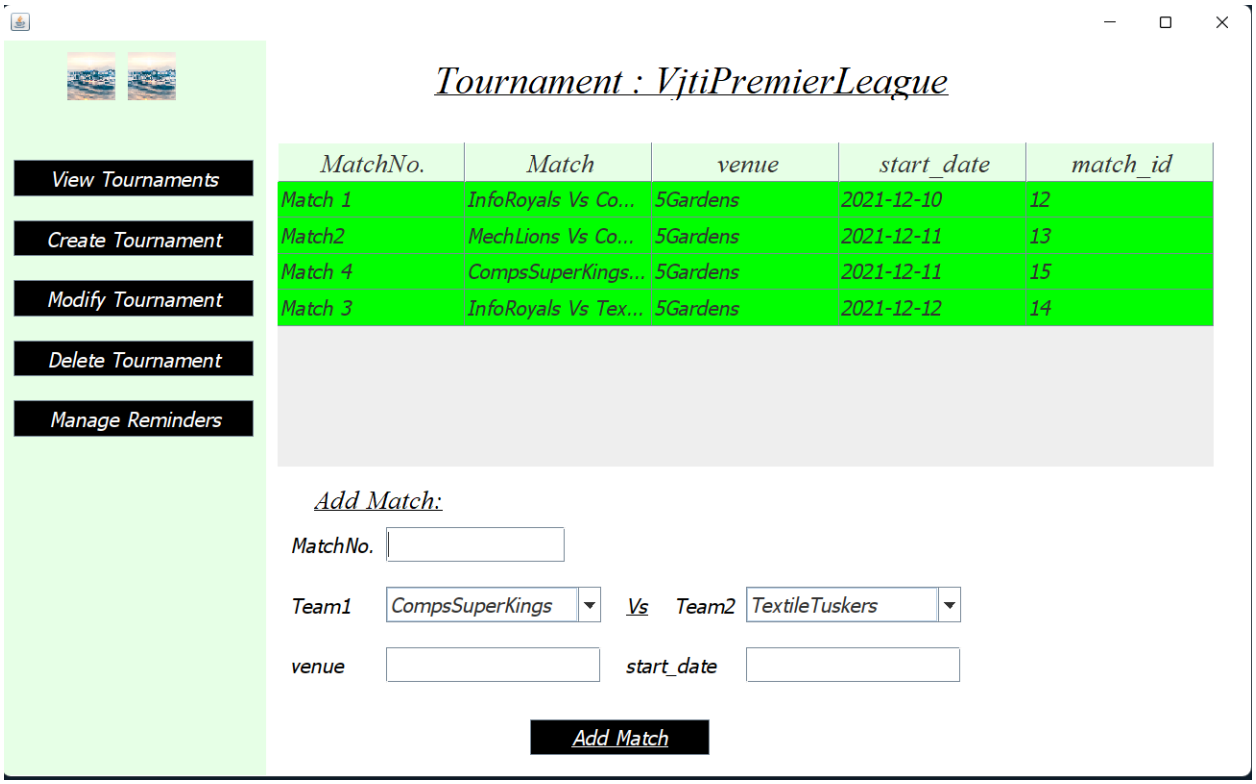
- Adding matches to tournament Vjti Premier League



- Putting in the venue and the start date for the tournament



- A List of added matches is displayed



- Changes regarding the same are reflected in the database

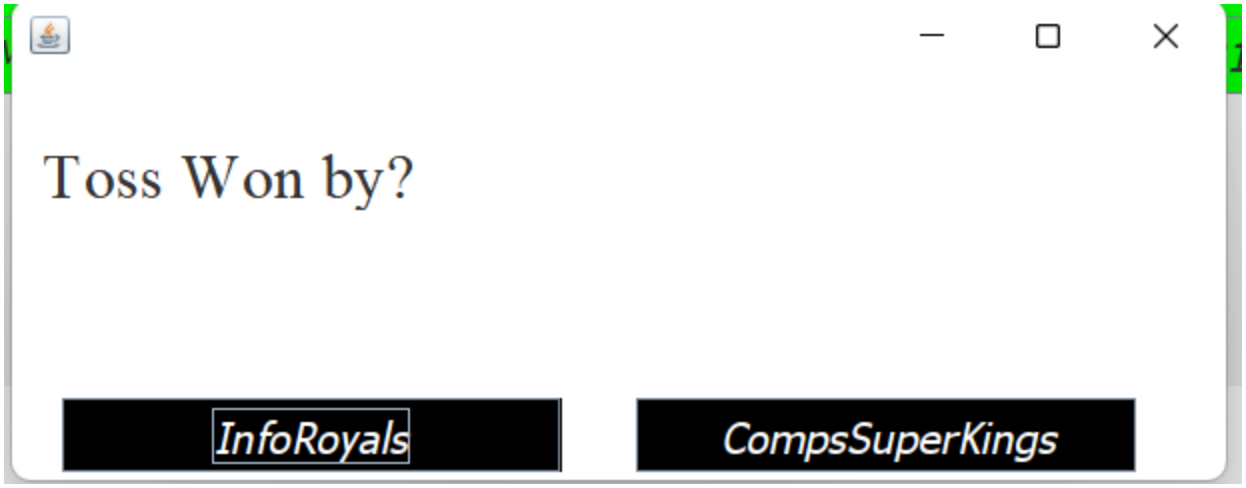
```
mysql> select match_id,team1_name,team2_name,tour_id from matches;
+----------+-----------------+---------------------+---------+
| match_id | team1_name      | team2_name          | tour_id |
+----------+-----------------+---------------------+---------+
|        7 | Mumbai Indians  | Chennai Super Kings |       2 |
|       12 | InfoRoyals      | CompsSuperKings     |       8 |
|       13 | MechLions       | CompsSuperKings     |       8 |
|       14 | InfoRoyals      | TextileTuskers      |       8 |
|       15 | CompsSuperKings | TextileTuskers      |       8 |
+----------+-----------------+---------------------+---------+
5 rows in set (0.00 sec)

mysql> select match_id,team1_name,team2_name,tour_id from matches where tour_id=8;
+----------+-----------------+-----------------+---------+
| match_id | team1_name      | team2_name      | tour_id |
+----------+-----------------+-----------------+---------+
|       12 | InfoRoyals      | CompsSuperKings |       8 |
|       13 | MechLions       | CompsSuperKings |       8 |
|       14 | InfoRoyals      | TextileTuskers  |       8 |
|       15 | CompsSuperKings | TextileTuskers  |       8 |
+----------+-----------------+-----------------+---------+
4 rows in set (0.00 sec)

mysql>
```

(Database changes above.Addition of matches.)

- Expected output: The details of the matches added should be reflected in the matches table.
- Result: Test Case Passed

13.3.6 Toss Winning Team

- Choosing which team wins the toss. Here, CompsSuperKings won the toss and chose to bowl. Hence, bat_first is InfoRoyal.



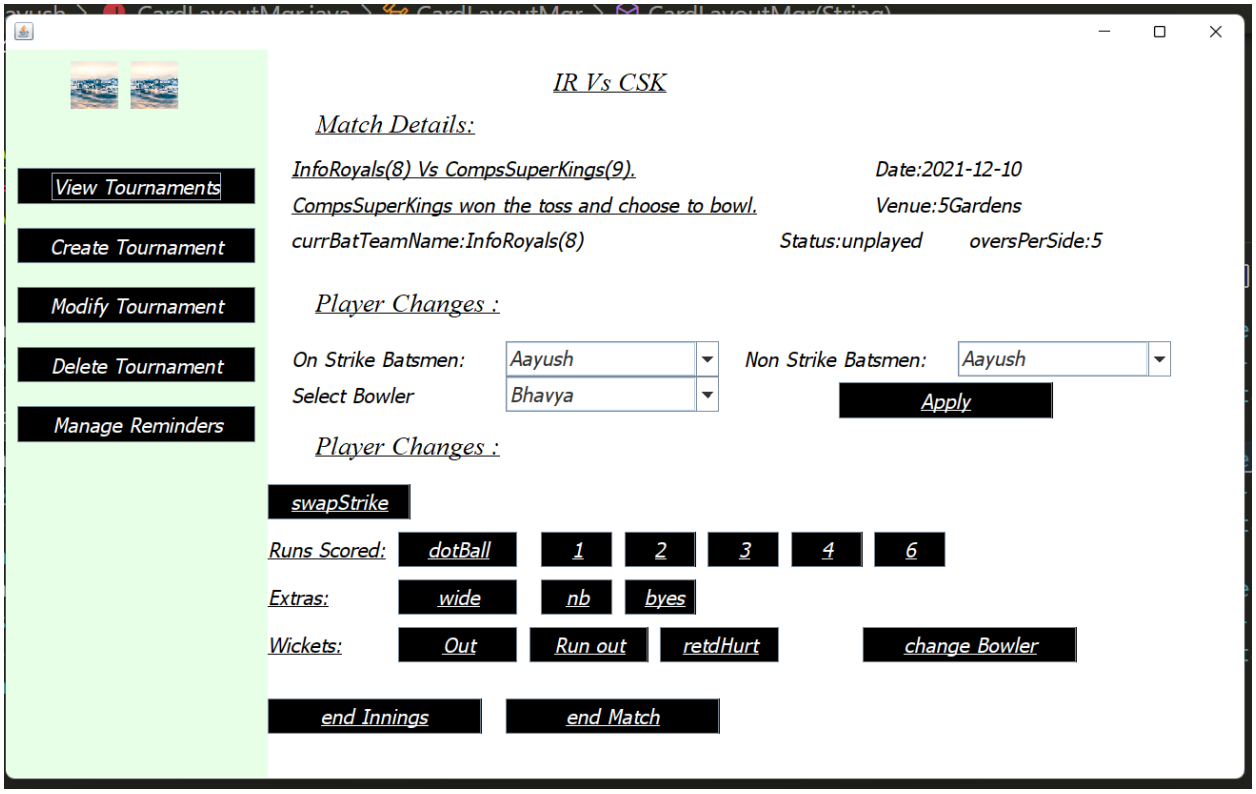
- Database after choosing who won the toss and entering what they chose.

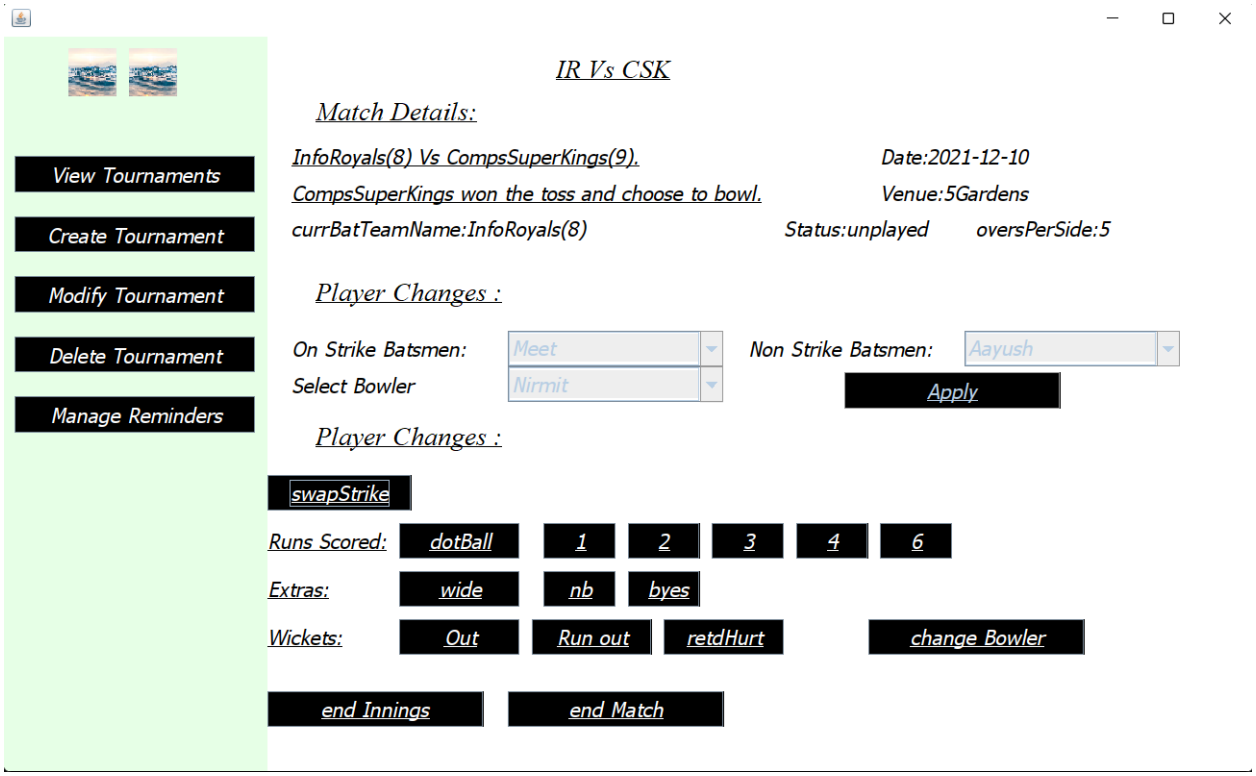
```
mysql> select match_id,team1_name,team2_name,toss,t.team_name,bat_first,t2.team_id from matches,team as t,team as t2 w
here t.team_id=toss and t2.team_id=bat_first;
+----------+---------------+--------------------+------+---------------+-----------+---------+
| match_id | team1_name    | team2_name         | toss | team_name     | bat_first | team_id |
+----------+---------------+--------------------+------+---------------+-----------+---------+
|        7 | Mumbai Indians | Chennai Super Kings |    4 | Mumbai Indians |         3 |       3 |
+----------+---------------+--------------------+------+---------------+-----------+---------+
1 row in set (0.00 sec)

mysql> select match_id,team1_name,team2_name,toss,t.team_name,bat_first,t2.team_id from matches,team as t,team as t2 w
here t.team_id=toss and t2.team_id=bat_first;
+----------+---------------+--------------------+------+---------------+-----------+---------+
| match_id | team1_name    | team2_name         | toss | team_name     | bat_first | team_id |
+----------+---------------+--------------------+------+---------------+-----------+---------+
|        7 | Mumbai Indians | Chennai Super Kings |    4 | Mumbai Indians |         3 |       3 |
|       12 | InfoRoyals    | CompsSuperKings    |    9 | CompsSuperKings |         8 |       8 |
+----------+---------------+--------------------+------+---------------+-----------+---------+
2 rows in set (0.00 sec)
```

- Expected Output: The team_id of who won the toss(as under column in the above screenshot) and the team_id of the team who will bat first(as under the bat_first column in the above screenshot) should be reflected in the database.
- Result: Test Case Passed

13.3.7 Scorer Panel

- An ongoing match between InfoRoyals and CompsSuperKings that enables user to input the ball event (runs/extras/wicket), change the strike for batters and change bowler.






```
mysql> select on_strike_bat_name,non_strike_bat_name,curr_bowl_name from matches where match_id=12;
+--------------------+--------------------+----------------+
| on_strike_bat_name | non_strike_bat_name | curr_bowl_name |
+--------------------+--------------------+----------------+
| NULL               | NULL               | NULL           |
+--------------------+--------------------+----------------+
1 row in set (0.00 sec)

mysql> select on_strike_bat_name,non_strike_bat_name,curr_bowl_name from matches where match_id=12;
+--------------------+--------------------+----------------+
| on_strike_bat_name | non_strike_bat_name | curr_bowl_name |
+--------------------+--------------------+----------------+
| Meet               | Aayush             | Nirmit         |
+--------------------+--------------------+----------------+
1 row in set (0.00 sec)

mysql>
```

- Expected Output:
  Database changes accordingly

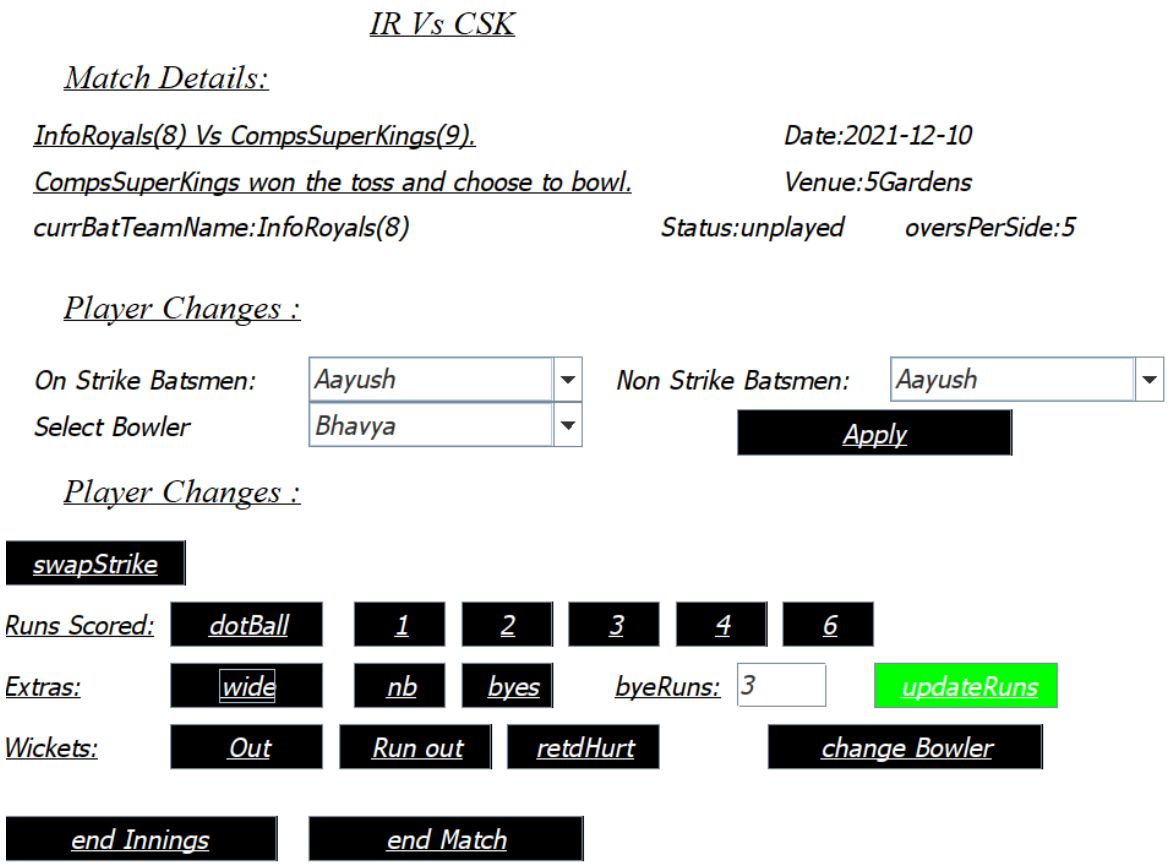
1 run scored

```
mysql> select * from scorecardBatting where player_id=13;
+----------+-----------+------+--------------+-------+-------+-----------+-----------+
| match_id | player_id | runs | balls_played | fours | sixes | out_style | bowled_by |
+----------+-----------+------+--------------+-------+-------+-----------+-----------+
|       12 |        13 |    0 |            0 |     0 |     0 | not_out   | NULL      |
+----------+-----------+------+--------------+-------+-------+-----------+-----------+
1 row in set (0.00 sec)

mysql> select * from scorecardBatting where player_id=13;
+----------+-----------+------+--------------+-------+-------+-----------+-----------+
| match_id | player_id | runs | balls_played | fours | sixes | out_style | bowled_by |
+----------+-----------+------+--------------+-------+-------+-----------+-----------+
|       12 |        13 |    1 |            1 |     0 |     0 | not_out   | NULL      |
+----------+-----------+------+--------------+-------+-------+-----------+-----------+
1 row in set (0.00 sec)

mysql> select * from scorecardBowling where player_id=17;
+----------+-----------+---------+--------+-------+------------+
| match_id | player_id | wickets | maiden | overs | runs_given |
+----------+-----------+---------+--------+-------+------------+
|       12 |        17 |       0 |      0 |   0.1 |          1 |
+----------+-----------+---------+--------+-------+------------+
1 row in set (0.00 sec)
```

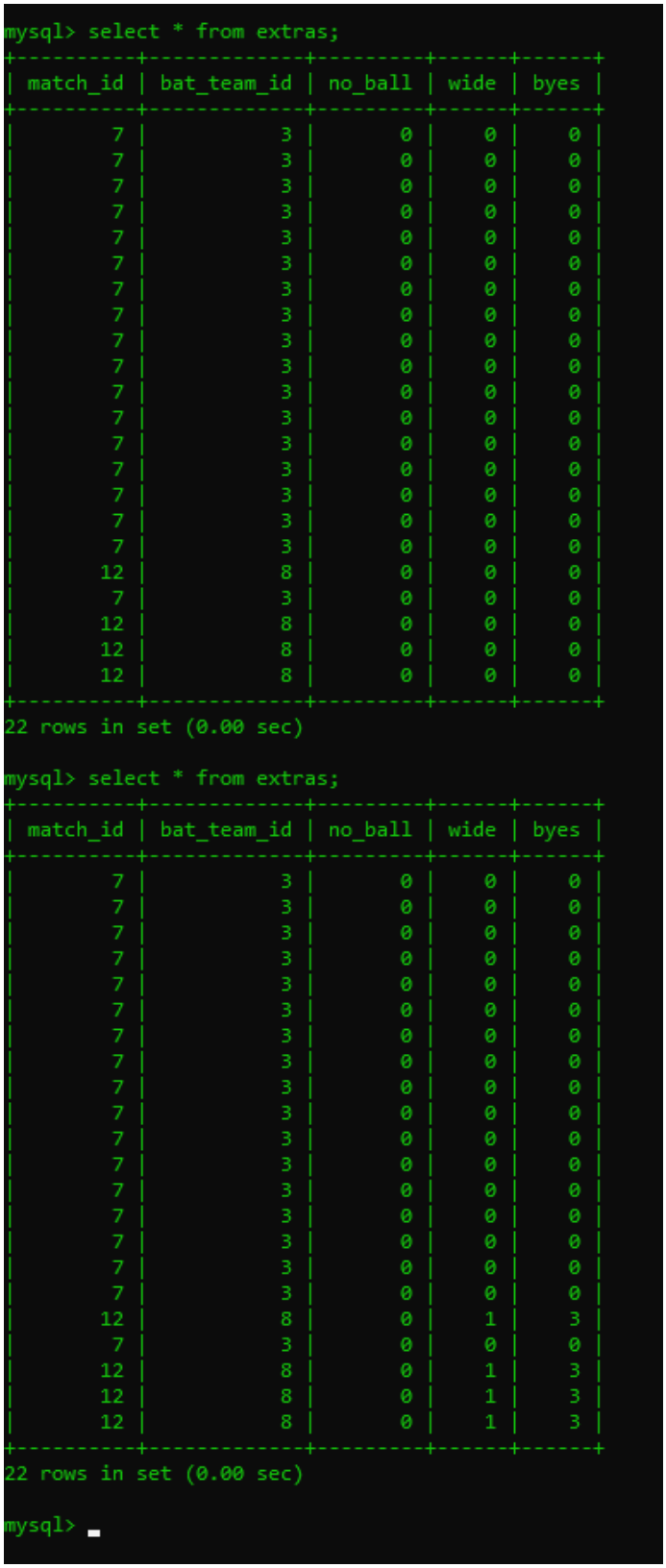
Database changes . Likewise for dotBall, 2, 3 runs.

4,6 runs scored

```
mysql> select * from scorecardBatting where player_id=13;
+----------+-----------+------+--------------+-------+-------+-----------+-----------+
| match_id | player_id | runs | balls_played | fours | sixes | out_style | bowled_by |
+----------+-----------+------+--------------+-------+-------+-----------+-----------+
|       12 |        13 |    6 |            3 |     1 |     0 | not_out   | NULL      |
+----------+-----------+------+--------------+-------+-------+-----------+-----------+
1 row in set (0.00 sec)

mysql> select * from scorecardBowling where player_id=17;
+----------+-----------+---------+--------+-------+------------+
| match_id | player_id | wickets | maiden | overs | runs_given |
+----------+-----------+---------+--------+-------+------------+
|       12 |        17 |       0 |      0 |   0.4 |          6 |
+----------+-----------+---------+--------+-------+------------+
1 row in set (0.00 sec)

mysql> select team1_runs from matches where match_id=12;
+------------+
| team1_runs |
+------------+
|          5 |
+------------+
1 row in set (0.00 sec)

mysql>
```



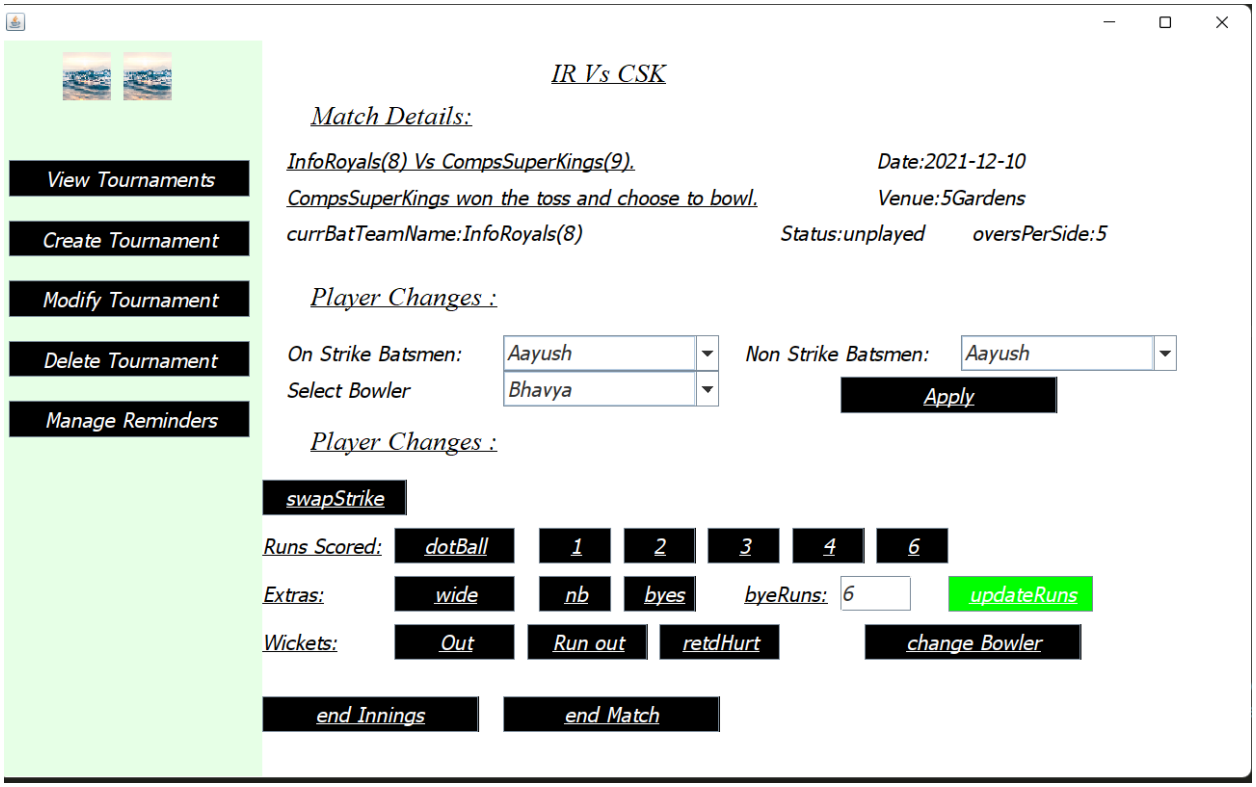
Wide and 3 bye runs



```
mysql> select * from extras;
+----------+-------------+---------+------+------+
| match_id | bat_team_id | no_ball | wide | byes |
+----------+-------------+---------+------+------+
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|       12 |           8 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|       12 |           8 |       0 |    0 |    0 |
|       12 |           8 |       0 |    0 |    0 |
|       12 |           8 |       0 |    0 |    0 |
+----------+-------------+---------+------+------+
22 rows in set (0.00 sec)

mysql> select * from extras;
+----------+-------------+---------+------+------+
| match_id | bat_team_id | no_ball | wide | byes |
+----------+-------------+---------+------+------+
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|       12 |           8 |       0 |    1 |    3 |
|        7 |           3 |       0 |    0 |    0 |
|       12 |           8 |       0 |    1 |    3 |
|       12 |           8 |       0 |    1 |    3 |
|       12 |           8 |       0 |    1 |    3 |
+----------+-------------+---------+------+------+
22 rows in set (0.00 sec)

mysql>
```

Addition of byes and wide in the database table extras.

No ball + six runs



```
mysql> select * from extras;
+----------+-------------+---------+------+------+
| match_id | bat_team_id | no_ball | wide | byes |
+----------+-------------+---------+------+------+
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|        7 |           3 |       0 |    0 |    0 |
|       12 |           8 |       1 |    2 |    3 |
|        7 |           3 |       0 |    0 |    0 |
|       12 |           8 |       1 |    2 |    3 |
|       12 |           8 |       1 |    2 |    3 |
|       12 |           8 |       1 |    2 |    3 |
|       12 |           8 |       1 |    0 |    0 |
+----------+-------------+---------+------+------+
23 rows in set (0.00 sec)

mysql>
```

```
mysql> select * from scorecardBatting where player_id=13;
+----------+-----------+------+--------------+-------+-------+-----------+-----------+
| match_id | player_id | runs | balls_played | fours | sixes | out_style | bowled_by |
+----------+-----------+------+--------------+-------+-------+-----------+-----------+
|       12 |        13 |   12 |            3 |     1 |     1 | not_out   | NULL      |
+----------+-----------+------+--------------+-------+-------+-----------+-----------+
1 row in set (0.00 sec)

mysql>
```

Also the bye runs are added to the on strike batsmen runs and six is incremented by 1.

CONCLUSION

Being avid cricket fans, this project was not only a great learning experience, but also highly interesting for all of us to work on. We also got to learn many new things and used new technologies which have added to our skill sets.

We endeavored to make this application as best as we could, but we also inevitably encountered quite a few technical difficulties. For instance, we spent considerable time trying to discuss and finalize the main functionalities which we would work upon in this project, and the scale of the system that we were making. Initially progress was slow but as time progressed, we learned on the move and the pace picked up and we were able to zero in on a main goal and worked towards it.

Given the resources at our disposal and the difficulty of not being able to work on this project as a group physically due to the pandemic, we managed to work on major aspects of the system together and tried to achieve as much perfection as we could. Before starting on the project, we researched about similar applications like CricBuzz and ESPN Live, and learnt about the working of those applications to help shape our project more efficiently. But that does not mean that our app is exactly identical to them; we have included more features for the end-user such as adding and modifying tournaments, which gives the user more freedom to input data and see unique results.

The last 2-3 weeks were spent in code debugging and fine-tuning important things like the ER diagram, the relational schema, the Software Requirement Specification, and the design of the system. The code was written as and how we added functionalities to our project, hence we were only left with debugging it. The last week was spent in writing the report and testing out the software with various test cases and testing methods.

In a nutshell, this project was a very interesting journey for us throughout this semester. It helped us learn more about different aspects of software development and improved our programming and database management skills.

FUTURE SCOPE

We wish to add in a couple of new features to make the application easier to use and more adaptive to the standard of existing cricket applications. The Head to head player comparison enables coaches to compare players based on past records and statistics. Machine learning algorithms could be used for recommending players to the selection committee. Coaches could select specific players and add notes to keep track of their performances. These notes would provide valuable insights to the selection committee. Speaking of the database, we wish to create several levels of abstraction based on the login type. Currently the application merely has Admin and user login. However, depending on the nature of the created tournament, whether it's local or international, we could hide the local tournaments from a user that operates at an international level. Lastly, we wish to put in a bit of work into the UI too.