

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belagavi – 590 018.



A MINI PROJECT REPORT

On

“IPL MATCHES [All-Star] MANAGEMENT SYSTEM”

Submitted in partial fulfillment of the requirement for the curriculum of the 5th Semester

Bachelor of Engineering

In

Computer Science & Engineering

Submitted by

CHIRAG RADHAKRISHNA : **1VI19CS021**

CHIRANJEEVI PR : **1VI19CS022**

Under the supervision of

Prof. Naveen H S
Assistant Professor

Dr. Kantharaju H C
Associate Professor



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

VEMANA INSTITUTE OF TECHNOLOGY

Bengaluru – 560 034

2021-2022

Karnataka ReddyJana Sangha®
VEMANA INSTITUTE OF TECHNOLOGY.
Koramangala, Bengaluru-34.
(Affiliated to Visvesvaraya Technological University, Belagavi)



Department of Computer Science & Engineering

Certificate

This is certified that the mini-project work entitled “**IPL MATCHES [All-Star] MANAGEMENT SYSTEM**” carried out by **Chirag (1VI19CS021)**, **Chiranjeevi (1VI19CS022)** bonafide students of **Vemana Institute of Technology** in partial fulfillment for the completion of **Database Management System laboratory** of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belagavi during the year 2021-2022. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The mini-project report has been approved as it satisfies the academic requirements in respect of the mini-project work prescribed for the said degree.

Guide-I

Prof. Naveen H S

Guide-II

Dr. Kantharaju H C

HOD

Dr. M Ramakrishna

External Viva

Name of the Examiners

Signature with date

1. _____

2. _____

ACKNOWLEDGEMENT

We sincerely thank **Visvesvaraya Technological University** for providing a platform to do a mini-project.

Firstly, we would like to express our deep sense of gratitude to our institute “**Vemana Institute of Technology**” that provided us an opportunity to do a mini-project entitled “**IPL MATCHES [All-Star] MANAGEMENT SYSTEM**”.

We thank **Dr. Vijayasimha Reddy. B. G**, Principal, Vemana Institute of Technology, Bengaluru for providing the necessary support.

We would like to place on record our regards to **Dr. M. Ramakrishna**, Professor & Head of the Department, Computer Science and Engineering for his continued support.

We would like to thank our mini-project guides **Prof. Naveen H S** Assistant Professor and **Dr Kantharaju H C**, Associate Professor, Dept. of CSE for their continuous support and valuable guidance towards successful completion of the mini-project.

We would be failing in our duty if we do not thank the faculty members, batch mate, lab staff, technicians and family members for their constant support and guidance.

Date:

Chirag Radhakrishna(1VI19CS021)

Place: Bengaluru

Chiranjeevi PR (1VI19CS022)

ABSTRACT

The aim of developing the IPL-All Star Games Management System was to highlight the importance of a Database Management System.

Considering applications in the present world, we need to realize that to build a powerful and dynamic user-interactive application; we not only require a well-designed front-end component but also require a resilient, well-defined, structured back-end database management system.

Thus by implementing a new idea in the form of the IPL-All Star games, we have implemented an application that makes use of a front-end and back-end tool and helps users/fans create an account, view the teams, the matches and the players participating in the all-star game and provide them a provision to book tickets to the games.

TABLE OF CONTENTS

Sl.no	Topic/Chapter	Pg. No
1.	CHAPTER 1: INTRODUCTION	1-3
	1.1 E-Business	1
	1.2 The IPL in a nutshell	1-2
	1.3 Concept of All-Star Games	2
	1.4 Project Process Overview	2-3
2.	CHAPTER 2: SYSTEM REQUIREMENTS	4-5
	2.1 Functional Requirements	4
	2.2 Non-Functional Requirements	4-5
3.	CHAPTER 3: SYSTEM DESIGN	6-9
	3.1 System Architecture	6-7
	3.1.1 Modules comprising the system/application	7
	3.2 ER Diagram	7-8
	3.3 Schema Diagram	9
4.	CHAPTER 4: SYSTEM IMPLEMENTATION	10-18
	4.1 FRONT END	10-12
	4.1.1 Programming Language	10
	4.1.2 Tool	10-11
	4.2 BACK END	12
	4.2.1 Structured Query Language	12
	4.2.2 MySQL Database	12
	4.3 JDBC (Java DataBase Connectivity)	12-13
	4.4 SOURCE CODE	14-18

5.	CHAPTER 5: SNAPSHOTS	19-26
6.	CHAPTER 6: CONCLUSION	27
7.	BIBLIOGRAPHY	28

LIST OF FIGURES

Fig. No	Particulars	Pg. No
3.1	System Architecture	6
3.2	ER Diagram	8
3.3	Schema Diagram	9
4.1	Libraries Used	11
4.2	JDBC Architecture	13
5.1	Owner Sign-in Frame	19
5.2	Customer Log-in Frame	20
5.3	Customer Sign-up Frame	20
5.4	Customer log-in after creating account	21
5.5	Teams Frame	21
5.6	Matches Frame	22
5.7	Seating-Arena Frame	22
5.8	Ticket Confirmation Frame	23
5.9	Payment Frame	23
5.10	Final-Ticket Frame	24
5.11	Logout Frame	25
5.12	Verifying Owner Identity	25
5.13	Owner viewing match bookings	26

INTRODUCTION

The IPL matches management system (All-Star Games) is an application designed to help fans and users book match tickets to watch the IPL All-Star Games at the Sardar Vallabhai Patel Stadium aka Narendra Modi Stadium. The user can book a match ticket by creating an account and then viewing the teams, the players and the scheduled matches and available seats.

This application is robust, user-friendly, yet easy and simple to use. It is based on the concept of database connectivity i.e., connecting a suitable front-end tool to a DBMS.

1.1 E-Business:

As our application is a seat booking system which allows booking of match tickets online, it can be classified under the category of electronic business.

E-business refers to a broader definition of EC, not just the buying and selling of goods and services, but also servicing customers, collaborating with business partners, conducting e-learning, & conducting electronic transactions within an organization.

Hence, we can sum up E-BUSINESS as:

“E-business refers to any form of transaction (exchange) that uses an electronic medium to facilitate the transaction.”

1.2 The IPL in a nutshell:

The Indian Premier League, commonly known as the IPL, is the most popular T20 cricketing competition in the world. Since its inception in 2008, the IPL has grown heaps and bounds season after season.

Some statistics regarding the Indian Premier League:

- Highest run-getter: Virat Kohli (6283)

- Highest wicket-taker: Lasith Malinga (170)
- Most successful franchise: Mumbai Indians (5 titles)
- Most successful captain: Rohit Sharma (Mumbai Indians)
- Most successful player: Rohit Sharma (6 IPL titles)

1.3 Concept of All-Star Games:

The All-Star Games with respect to any sport, are a series of popular games contested between the world's best players in that particular sport. Some of the popular All-Star games that are conducted annually across the world are the NBA All-Star Game, the NFL Pro-Bowl and MLB All-Star Game to name a few.

The IPL being the most popular cricketing tournament in the world, recently revealed its intentions to stage a series of All-Star Games.

The IPL All-Star games are a series of matches that are contested by the 8 founding franchises of the IPL and comprise some of the best talented players to ever play the game and most importantly, providing fans the ultimate matchday experience!

1.4 Project Process Overview:

- There are 8 teams participating in the All-Star Games. The user can view the teams and its players and select a match accordingly.
- The matches are part of a round –robin format wherein the teams face off against each other. Details of the matches are stored in MySQL table.
- If the user is interested in booking a seat for an All-Star Game, he/she can do so by selecting the desired seat through the interactive seat booking frame model.
- Once the match and the seats have been selected, the user is taken to a payment form, where he/she can pay via credit card or debit card by giving valid details.
- Once the formalities are finished, a final ticket frame consisting of all the match and ticket details will be displayed automatically. The user can then logout after saving his/her details.

- There are also provisions for the owner of the application, who can view the customer match data and get an idea of how many seats are booked for the All-Star games respectively.

SYSTEM REQUIREMENTS

System Requirements Specification (SRS) gives the information regarding analysis done for the application.

An SRS minimizes the time and effort required by developers to achieve desired goals and also minimizes the development cost. A good SRS defines how an application will interact with system hardware, other programs and human users in a wide variety of real-world situations.

System Analysis is done to capture the requirements of the user of the application.

The requirement specifications of the application are mentioned below-

2.1 Functional Requirements:

The functional requirements for the IPL Matches Management System (All-Star Games):

- a) Authentication of user whenever he/she logs into the application.
- b) The application must retrieve the match data related to the specific users.
- c) The application must display appropriate message while processing user sign-up and sign-in attempts, selecting a match and seats and validating and saving the match details.
- d) Ensure that there is no change in static data stored in the back-end database.
- e) Verify owner identity when owner wants to check number of tickets that have been booked.

2.2 Non-Functional Requirements:

The non-functional requirements for the IPL Matches Management System (All-Star Games):

During the window where users/fans are allowed to purchase All-Star tickets:

- a) The application should not be down for more than 5 minutes.
- b) The recovery time of the application should not exceed 3 minutes.

- c) The application should be able to track the user's actions when he/she is viewing the application to purchase All-Star tickets.

Availability requirements:

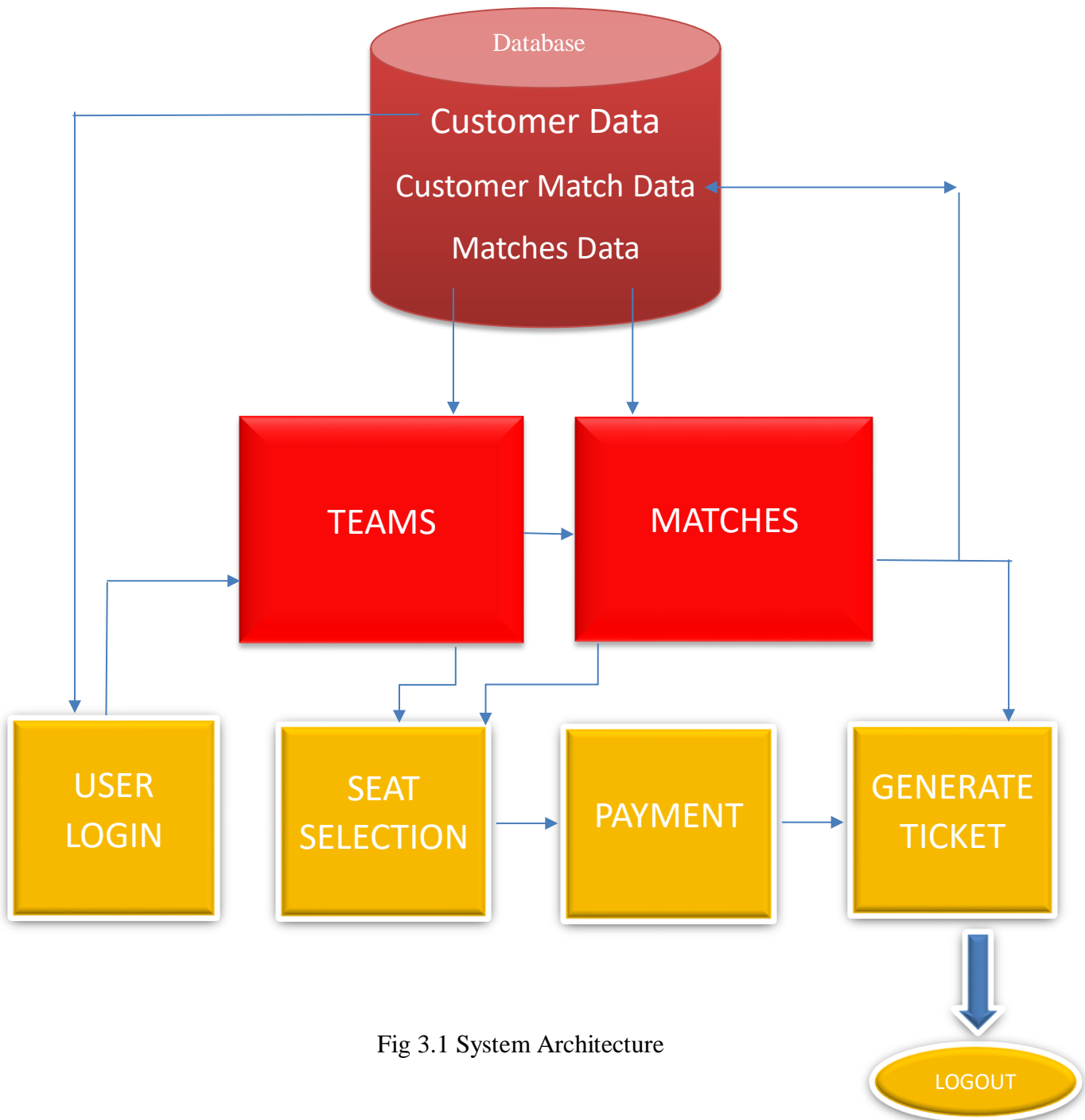
- a) When the customer presses any button of the application, the display should be updated in fractions of a second.
- b) Card validation should not exceed 10 seconds.

Security requirements:

- a) The application must ensure security of user details and his/her respective match details.

SYSTEM DESIGN

3.1 System Architecture



3.1.1 Modules comprising the system/application-

1. Database –

comprises of tables containing static data (teams, matches) and dynamic data which is manipulated at run-time by the customer/user. It is an integral component as it contains the necessary data that can be accessed by a user and also stores relevant data related to the user.

2. Teams-

comprises of the teams and their respective players. It houses a total of 8 teams and 88 players, that is 11 players per team.

3. Matches-

comprises of the matches that are to be played during the All-Star Games. Each team plays a total of 7 matches, that is a team plays each of the remaining teams once.

4. User login-

User creates an account that will be stored in the database. Once the account is set-up, the user will be able to login provided the entered login credentials matches the respective data stored in the database.

5. Seat Selection-

User selects the seats using interactive seat selection module and can calculate the price or change his/her selection before confirmation.

6. Payment & Generate ticket-

Once payment has been confirmed, final ticket can be generated.

7. Logout-

After obtaining the final All-Star ticket, the user can logout.

3.2 ER DIAGRAM-

Refer Figure 3.2 for ER Diagram

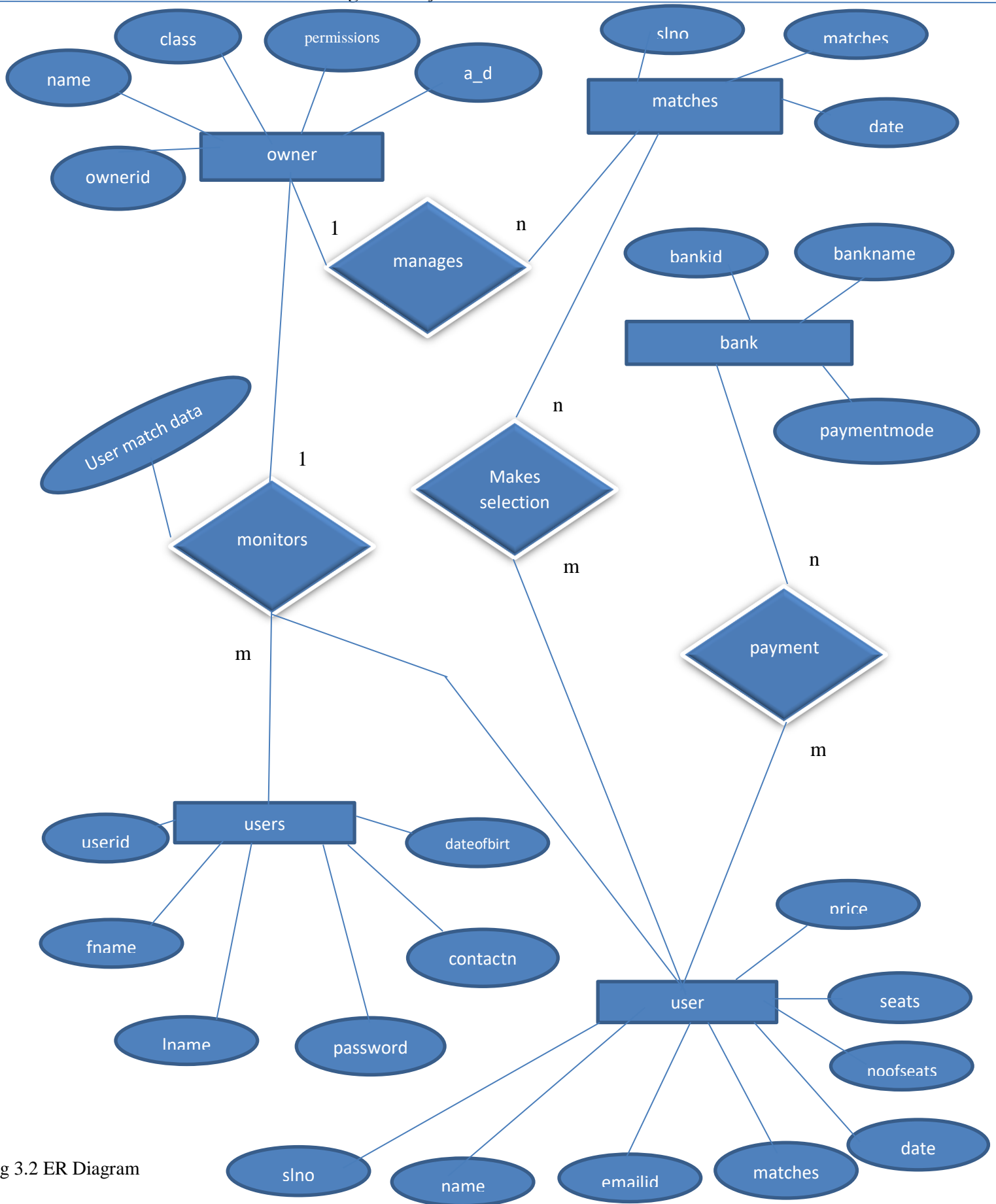


Fig 3.2 ER Diagram

3.3 Schema Diagram

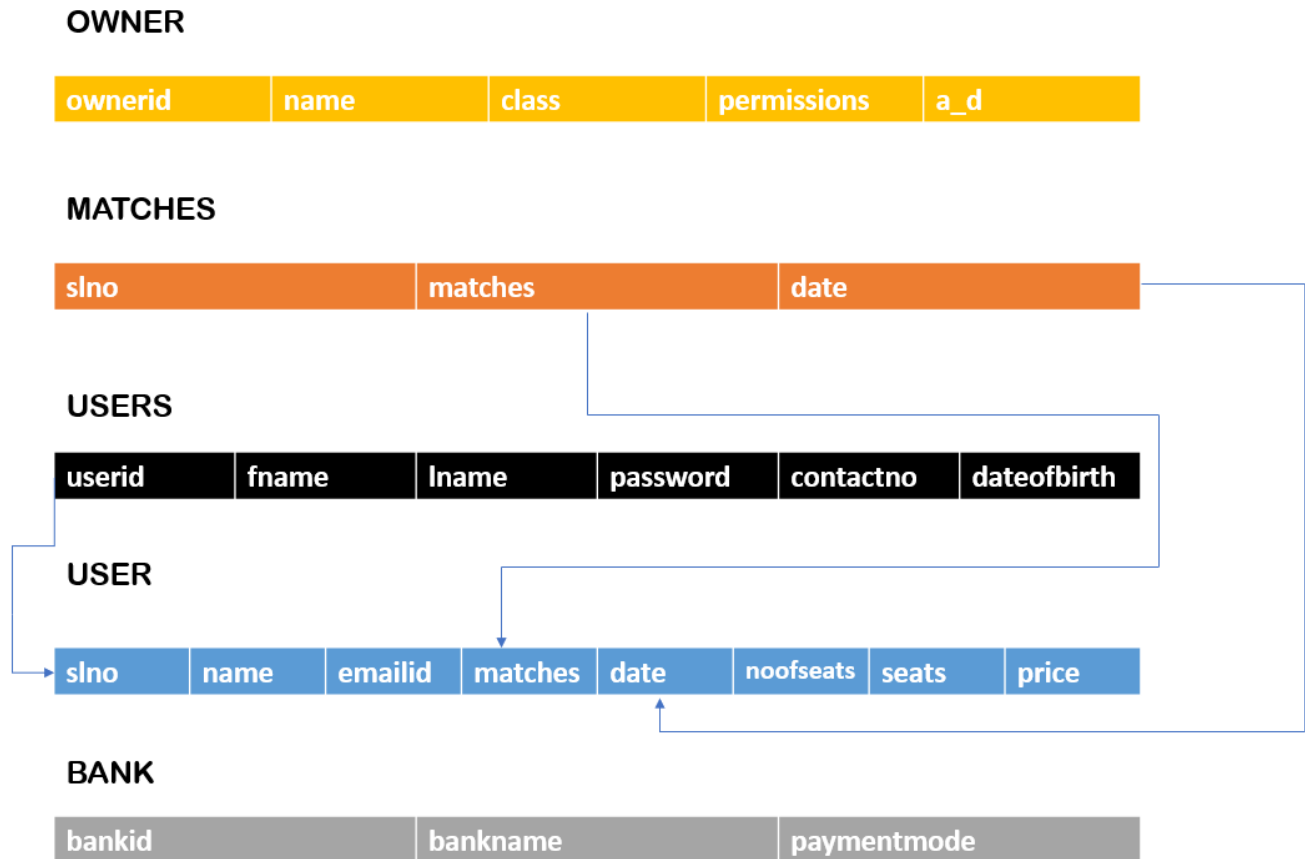


Fig 3.3 Schema Diagram

SYSTEM IMPLEMENTATION

4.1 Front-End

4.1.1 Programming language

Front-end programming language used is Java programming language.

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible.

4.1.2 Tool

The tool used for developing the front-end application is NetBeans IDE. It is an integrated development environment.

An integrated development environment is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of at least a source code editor, build automation tools and a debugger.

NetBeans IDE-

NetBeans is an open-source project dedicated to providing rock solid software development products (the NetBeans IDE and the NetBeans Platform) that address the needs of developers, users and the businesses who rely on NetBeans as a basis for their products; particularly, to enable them to develop these products quickly, efficiently and easily by leveraging the strengths of the Java platform and other relevant industry standards. In June 2000, NetBeans was made open source by Sun Microsystems, which remained the project sponsor until January 2010 when Sun Microsystems became a subsidiary of Oracle.

Libraries used in NetBeans IDE for implementing the front-end frames (refer Figure 4.1):

1. Absolute Layout
2. MySQL JDBC Driver
3. JCalendar

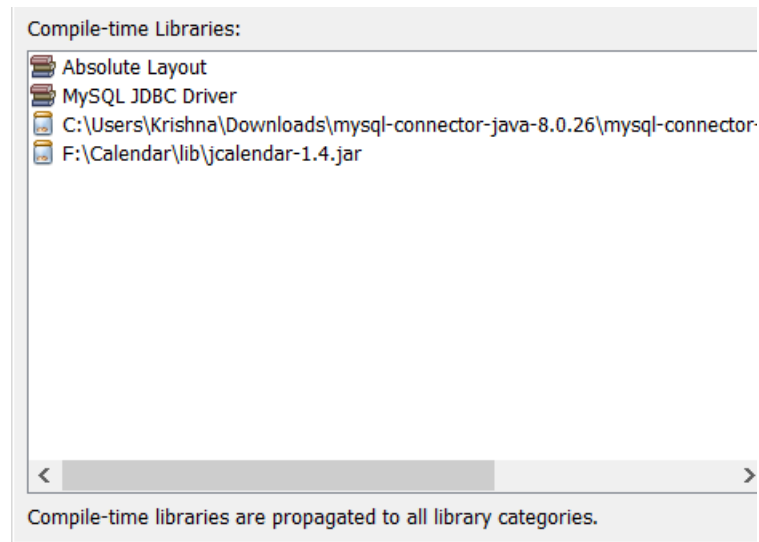


Fig 4.1 Libraries Used

Containers used:

Top-Level Containers: are used for presenting the frames.

- ✓ JFrame
- ✓ JDialog

Medium-Level Containers: are used for designing frame layouts. They are placed in the top-level containers.

- ✓ JPanel
- ✓ JTable

Low-Level Containers: are used for designing user-interactive components and are placed in medium-level containers.

- ✓ JButton
- ✓ JCheckBox
- ✓ JComboBox
- ✓ JRadioButton
- ✓ JPasswordField
- ✓ JLabel
- ✓ JOptionPane

- ✓ JTextArea
- ✓ JButtonGroup
- ✓ JDateChooser
- ✓ JPasswordField

4.2 Back-End

4.2.1 Structured Query Language

SQL is a domain-specific language used in programming and designed for managing data held in a relational database management system, or for stream processing in a relational data stream management system.

4.2.2 MySQL Database

MySQL is a freely available open-source *Relational Database Management System* (RDBMS) that uses Structured Query Language (SQL). A single MySQL database can contain many tables at once and store thousands of individual records.

MySQL provides you with a rich set of features that support a secure environment for storing, maintaining, and accessing data. MySQL is a fast, reliable, scalable alternative to many of the commercial RDBMS available today.

4.3 JDBC (Java Database Connectivity):

Helps to connect to a relational database (Ex: MySQL) from within a Java Application. We have done it using the JDBC API of Java. (Refer Figure 4.2)

It is part of the Java Standard Edition platform, from Oracle Corporation.

It mainly does the following tasks from within a Java application code:

- Establish connection with the database.
- Send SQL statements to the database server.
- Process the results obtained.

There are four main classes in the JDBC API hierarchy that are generally used for database connectivity. These are:

- ✓ **DriverManager Class:** The JDBC DriverManager class loads the JDBC driver needed to access a particular data source, locates and logs on to the database and returns a *Connection Object*.
- ✓ **Connection Class:** The JDBC Connection class manages the communication between a Java client application and a specific database, including passing SQL statements to the DBMS and managing transactions.
- ✓ **Statement Class:** The JDBC Statement Class contains SQL strings that are submitted to the DBMS. An SQL *Select statement* returns a *ResultSet object* that contains the data retrieved as the result of an SQL statement.
- ✓ **ResultSet Class:** The JDBC ResultSet class provides predefined methods to access analyze and convert data values returned by SQL *Select Statement*.

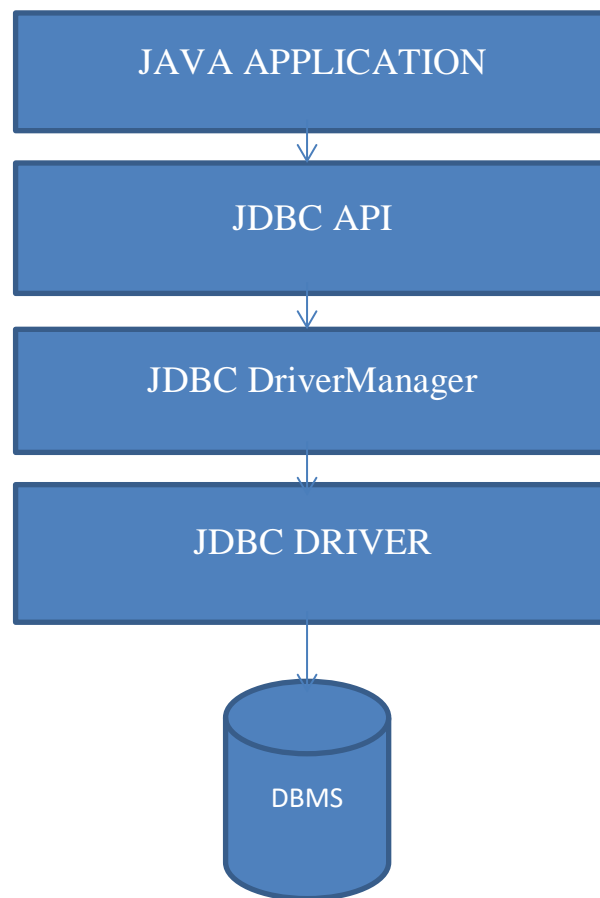


Fig 4.2 JDBC Architecture

4.4 Source Code

CREATING A USER ACCOUNT

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    int x = Integer.parseInt(jTextField1.getText());  
  
    String name = jTextField2.getText();  
  
    String lname = jTextField3.getText();  
  
    String pass = jPasswordField1.getText();  
  
    String phno = jTextField4.getText();  
  
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");  
  
    String date = sdf.format(jDateChooser1.getDate());  
  
    try  
    {  
  
        Class.forName("java.sql.Driver");  
  
        Connection con =  
        DriverManager.getConnection("jdbc:mysql://localhost/project","root","student");  
  
        Statement stmt = con.createStatement();  
  
        String query = "INSERT INTO users  
        VALUES("+x+", '"+name+"', '"+lname+"', '"+pass+"', '"+phno+"', '"+date+"')";  
  
        int rs = stmt.executeUpdate(query);  
  
        stmt.close();  
  
        con.close();  
  
        JOptionPane.showMessageDialog(this, "YOU HAVE OFFICIALLY SIGNED UP!!!!!!");  
  
        logInFrame xyz = new logInFrame();  
  
        xyz.setVisible(true);  
    }  
}
```

```
dispose();  
  
}  
  
catch(Exception e) {  
  
    JOptionPane.showMessageDialog(this,e);  
  
}}
```

SELECTING A MATCH AFTER VIEWING AVAILABLE MATCHES-

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
  
try  
  
{  
  
String n1=jTextField1.getText();  
  
String n2=jTextField2.getText();  
  
String n3=jLabel10.getText();  
  
Class.forName("java.sql.Driver");  
  
Connection  
con=DriverManager.getConnection("jdbc:mysql://localhost/project","root","student");  
  
Statement stmt=con.createStatement();  
  
String query="Insert into user(slno,matches,date)values('"+n3+"','"+n1+"','"+n2+"');";  
  
stmt.executeUpdate(query);  
  
con.close();  
  
stmt.close();  
  
new seatingFrame(n3).setVisible(true);  
  
dispose();  
  
}  
  
catch(Exception e)
```

```
{  
  
    JOptionPane.showMessageDialog(null,e);  
  
} }
```

USER SELECTING SEATS AND CALCULATING THE PRICE

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    String n = res.getText();  
  
    int a =0,b=0,c=0;  
  
    for(int i=0;i<n.length();i++)  
  
    {  
  
        if((n.charAt(i)=='T')||(n.charAt(i)=='B'))  
  
        {  
  
            a++;  
  
        }  
  
        else if((n.charAt(i)=='L')||(n.charAt(i)=='R'))  
  
        {  
  
            b++;  
  
        }  
  
    }  
  
    jTextField1.setText((a*20000+b*30000)+"");  
  
    int x=0;  
  
    for(int i=0;i<n.length();i++)  
  
    {  
  
        if((n.charAt(i)=='T')||(n.charAt(i)=='B')||(n.charAt(i)=='R')||(n.charAt(i)=='L'))  
  
        {
```

```
x++;  
  
}  
  
jTextField2.setText(""+x);  
  
}  
  
}
```

GENERATING FINAL TICKET

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
  
    String id = jLabel8.getText();  
  
    try  
    {  
  
        Class.forName("java.sql.Driver");  
  
        Connection  
        con=DriverManager.getConnection("jdbc:mysql://localhost/project","root","student");  
  
        Statement stmt=con.createStatement();  
  
        String quer="select name,emailid,seats,matches from user where slno= '"+id+"'";  
  
        System.out.println(quer);  
  
        ResultSet rs=stmt.executeQuery(quer);  
  
        if(rs.next())  
        {  
  
            String n1=rs.getString("name");  
  
            String n2=rs.getString("emailid");  
  
            String n3=rs.getString("seats");  
  
            String n4 = rs.getString("matches");  
  
            jTextField1.setText(""+n1);
```



```
textField2.setText(""+n2);

textField3.setText(""+n3);

textField4.setText(""+n4);

}

rs.close();

con.close();

stmt.close();

}

catch(Exception e)

{

    JOptionPane.showMessageDialog(null,"Error");

}

}

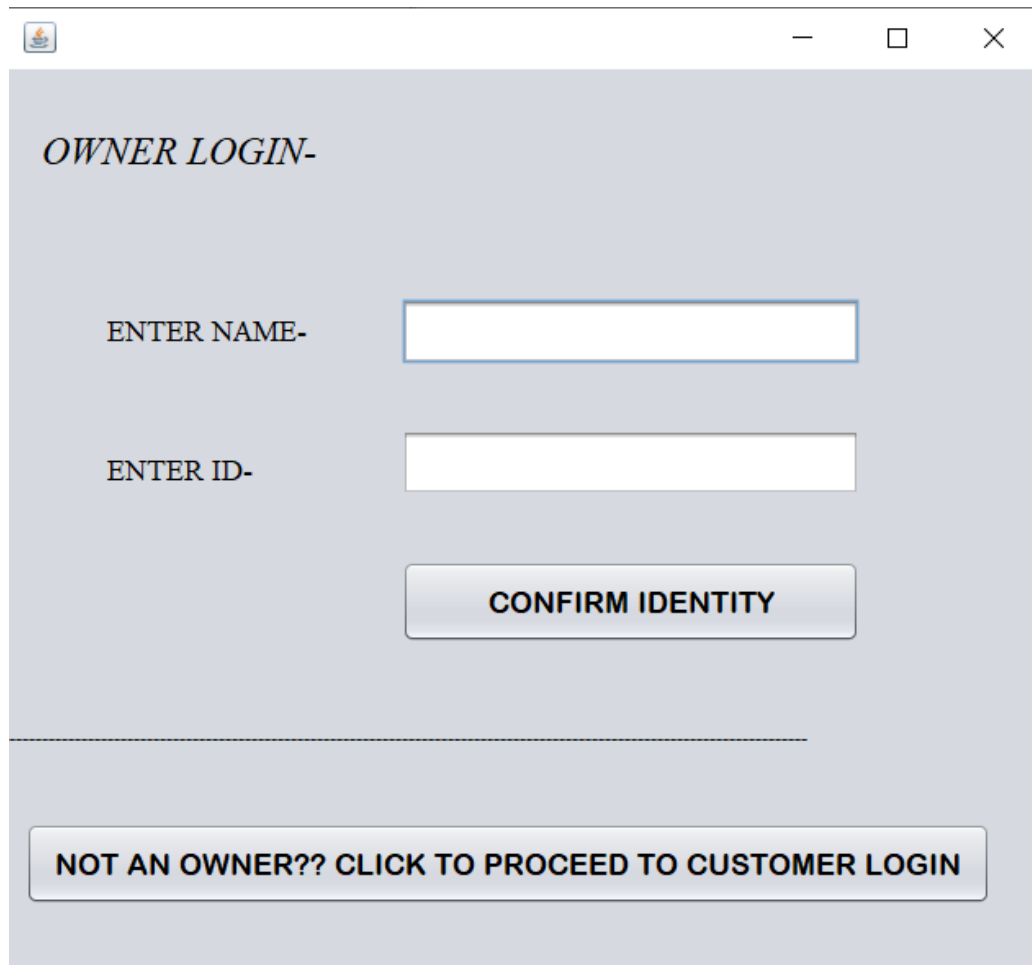
// saving the details for future reference

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

JOptionPane.showMessageDialog(this,"DETAILS SAVED");

}
```

SNAPSHOTS



A screenshot of a web application window titled "OWNER LOGIN-". The window has a standard title bar with a minimize button, a maximize button, and a close button. The main content area is light gray and contains the following elements:

- The text **OWNER LOGIN-** in a bold, black, serif font.
- A label **ENTER NAME-** followed by a white rectangular text input field with a blue border.
- A label **ENTER ID-** followed by a white rectangular text input field with a gray border.
- A button labeled **CONFIRM IDENTITY** with a gray gradient and a black border.
- A horizontal dashed line separating the login fields from the bottom section.
- A button labeled **NOT AN OWNER?? CLICK TO PROCEED TO CUSTOMER LOGIN** with a gray gradient and a black border.

Fig 5.1 Owner Sign-in Frame



Fig 5.2 Customer log-in Frame

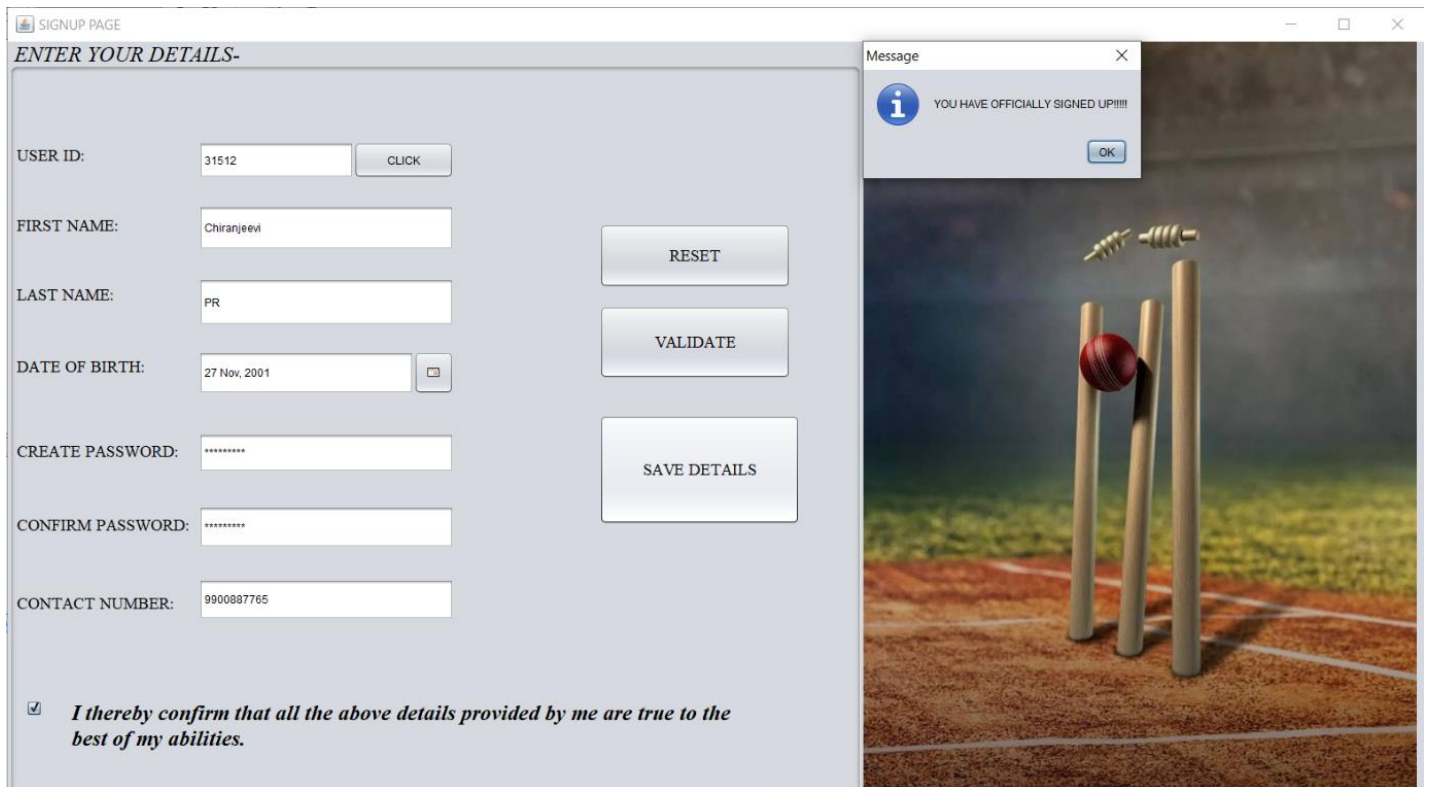


Fig 5.3 Customer sign-up Frame: Customer creates an account.



Fig 5.4 Customer log-in after creating an account

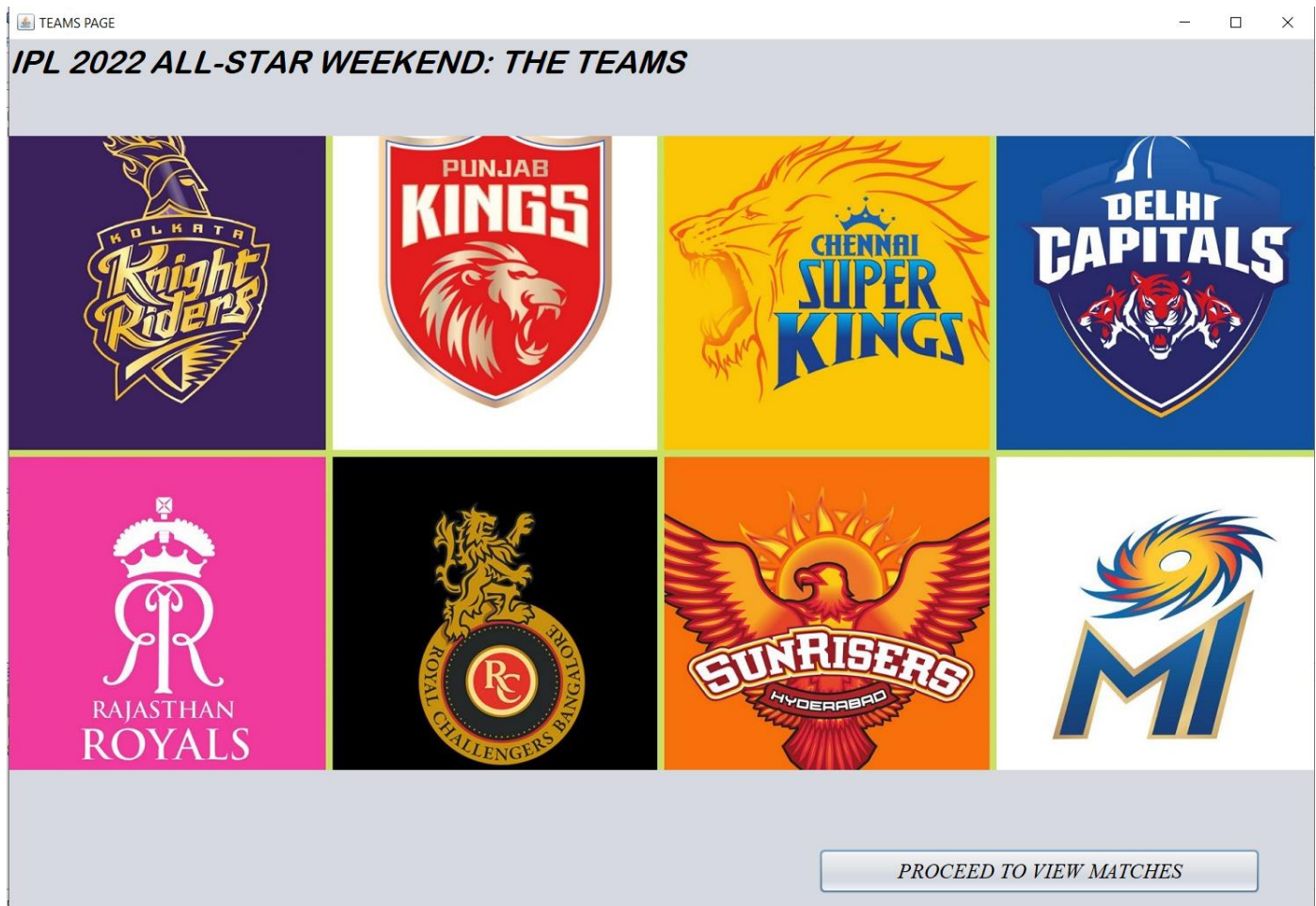


Fig 5.5 Teams Frame

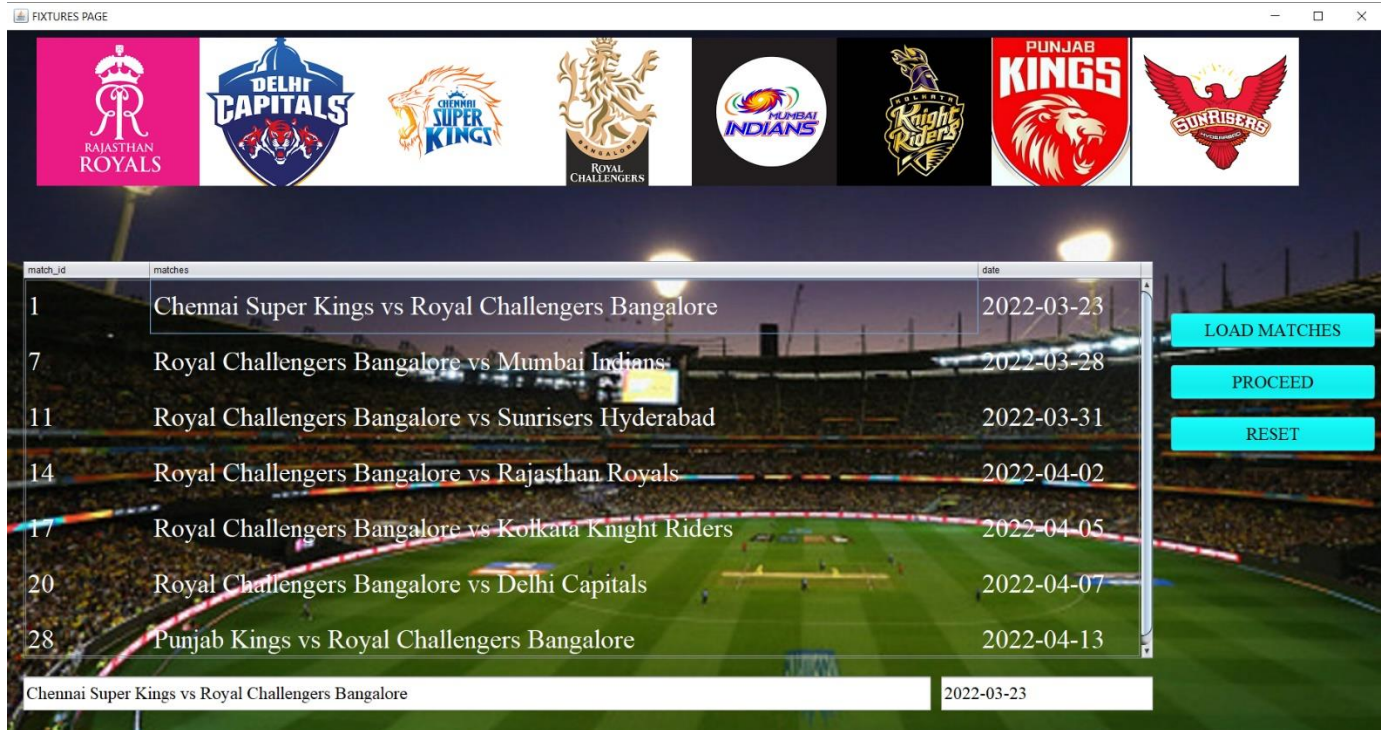


Fig 5.6 Matches Frame

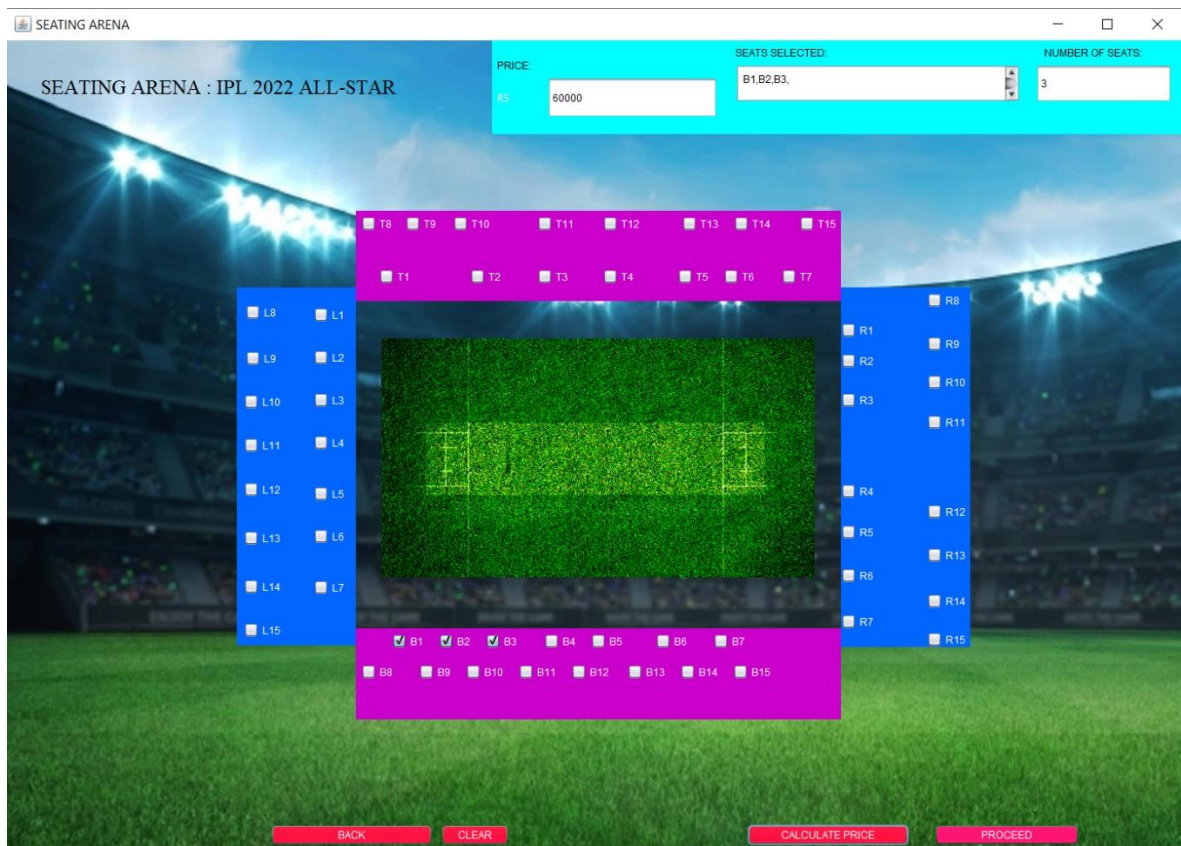


Fig 5.7 Seating-Arena Frame

TICKET CONFIRMATION PAGE

ENTER NAME AND EMAIL-ID: [Details will be used for final ticket]-

NAME OF TICKET HOLDER: CHIRANJEEVI PR

EMAIL-ID : chiranjeeviabd18@gmail.com

LOAD MATCH INFORMATION

YOUR TICKET DETAILS:

NO. OF SEATS BOOKED: 3

SEAT NUMBERS: B1,B2,B3

MATCH DATE!!!! 2022-03-23

MATCH: Chennai Super Kings vs Royal Challengers Bangalore

STADIUM: SARDAR VALLABHAI PATEL STADIUM

TOTAL PRICE [IN RS] 60000

PROCEED TO MAKE PAYMENT BACK

#CHECK YOUR DETAILS BEFORE MAKING PAYMENT. DETAILS MAY BE USED FOR MERCHANDISE.

Fig 5.8 Ticket Confirmation Frame

PAYMENT PAGE

OUR PARTNERED BANKS-

bankid	bankname	paymentmode
IINDAX88DB	AXIS BANK	DEBIT CARD
IINDCB11AA	CITI BANK	CREDIT CARD
IINDCCN777	CANARA BANK	CREDIT CARD
IINDHDFC30	HDFC BANK	CREDIT CARD
IINDICIC13	ICICI BANK	CREDIT CARD
IINDIDFCDB	IDFC BANK	DEBIT CARD

LOAD BANKS

FILL OUT YOUR CARD DETAILS TO COMPLETE PAYMENT-

ENTER NAME ON CARD: CHIRANJEEVI PR

ENTER CARD NO: 1232 3456 7890 3456

ENTER EXPIRY DATE: 12/23

ENTER CVV: ***

CONFIRM PAYMENT

GENERATE FINAL TICKET

PAYMENT OPTIONS

CREDIT CARD

DEBIT CARD

* Options available for Debit Payment

AMERICAN EXPRESS

Message

Payment Confirmed

OK


Fig 5.9 Payment Frame

FINAL TICKET GENERATION PAGE

GENERATE YOUR IPL 2022 ALL-STAR TICKET.....

GENERATE TICKET

FINAL ALL-STAR TICKET-



NAME: CHIRANJEEVI PR

EMAIL-ID: chiranjeeviabd18@gmail.com

SEATS BOOKED- B1,B2,B3,

MATCH- Chennai Super Kings vs Royal Challengers Bangalore

Message

i DETAILS SAVED

OK

SAVE DETAILS **PROCEED**

Fig 5.10 Final-Ticket Frame

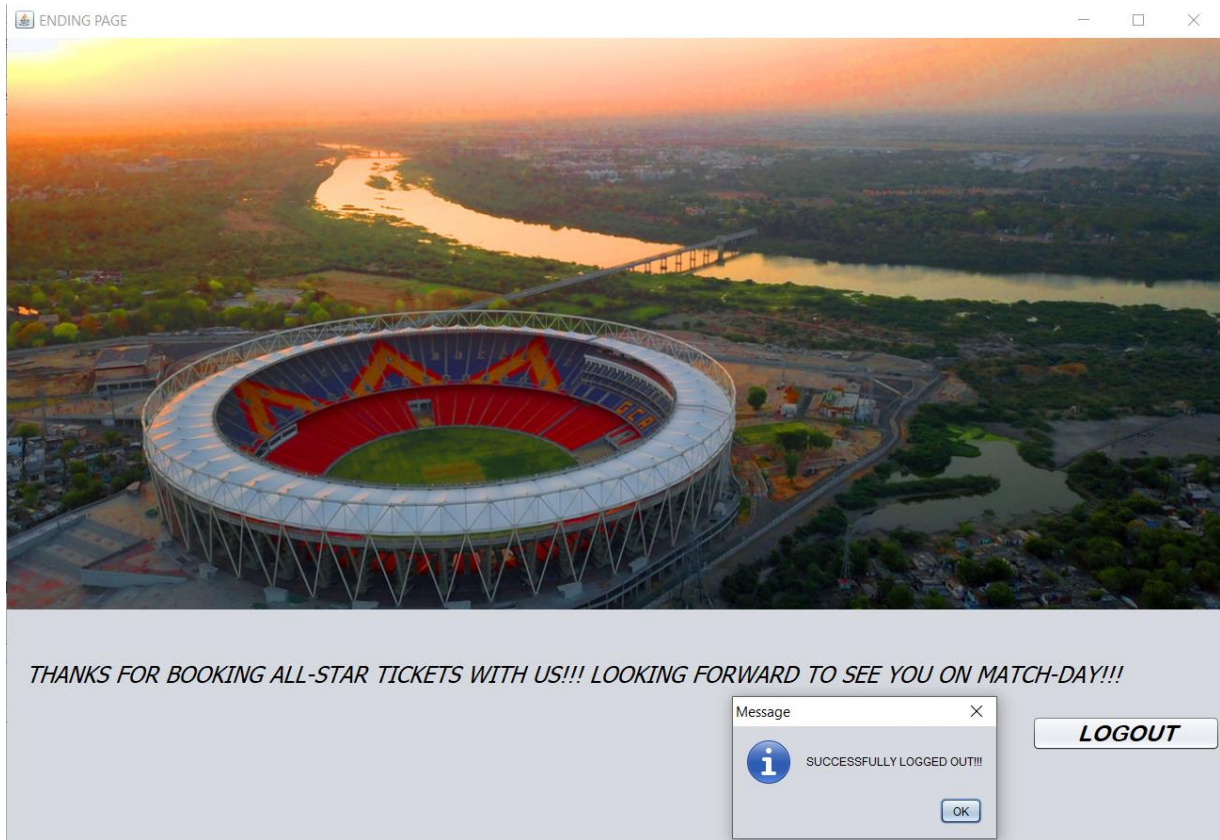


Fig 5.11 Logout Frame



Fig 5.12 Verifying Owner Identity

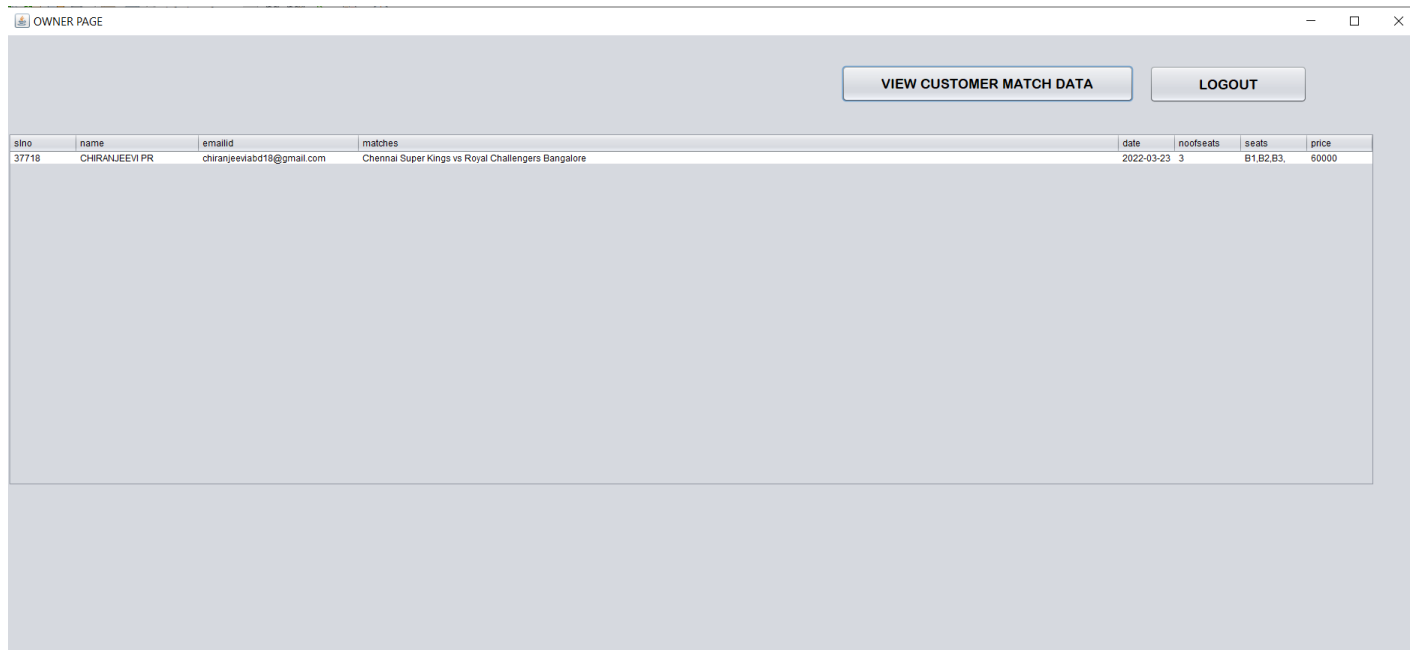


Fig 5.13 Owner viewing match bookings

CONCLUSION

In this mini-project we have successfully implemented IPL All-Star Games Management System that allows a user to purchase tickets for the IPL All-Star games. By implementing this system/application we have showcased various functionalities and uses of a database and an integrated development environment (IDE). Thus created a fully functional application for purchasing all-star tickets. Online seat booking is gradually becoming more and more popular day by day and has several advantages like reduction in commuting costs, avoid standing in long queues and also helps promote digital transactions. By using this system, the user can know which seats are available and can make the selections accordingly. This system is designed for commercial purpose, that is, to market All-Star Game tickets and provide users/fans with a dynamic yet user-friendly interface to purchase tickets to the games and support and cheer the sporting stars they idolize.

BIBLIOGRAPHY

- [1].<https://stackoverflow.com>
- [2].<https://www.javaguides.net>
- [3].<https://www.tutorialspoint.com/index.htm>
- [4]. Sumita Arora: informatics practices
- [5]. www.wikipedia.org