VISVESVARAYA TECHNOLOGICAL UNIVERSITY BELGAUM-590014

A DBMS Mini-Project Report

on

"NBA SEAT AND DESTINATION BOOKING SYSTEM"

A Mini-project report submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering in* Computer Science and Engineering *of Visvesvaraya Technological University, Belgaum.

Submitted by:

JEEVAN RAJU (1DT19CS061)

Under the Guidance of:

Prof. K. Deepa Shree
(Asst. Prof. Dept of CSE)

Department of Computer Science and Engineering

DAYANANDA SAGAR ACADEMY OF TECHNOLOGY & MANAGEMENT

Opp. Art of Living, Udayapura, Kanakapura Road, Bangalore-560082
(Affiliated to Visvesvaraya Technological University, Belagavi and Approved by AICTE, New Delhi)
CE, CSE, ECE, EEE, ISE, ME Courses Accredited by NBA, New Delhi, NAAC A+

2021-2022

DAYANANDA SAGAR ACADEMY OF TECHNOLOGY & MANAGEMENT

Opp. Art of Living, Udayapura, Kanakapura Road, Bangalore-560082
(Affiliated to Visvesvaraya Technological University, Belagavi and Approved by AICTE, New Delhi)
CE, CSE, ECE, EEE, ISE, ME Courses Accredited by NBA, New Delhi, NAAC A+

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Mini-Project on Database Management System (DBMS) entitled **"NBA SEAT AND DESTINATION BOOKING SYSTEM"** has been successfully carried out by **JEEVAN RAJU (1DT19CS061)** a bonafide student of **Dayananda Sagar Academy of Technology and Management** in partial fulfillment of the requirements for the award of degree in **Bachelor of Engineering in Computer Science and Engineering** of **Visvesvaraya Technological University, Belgaum** during academic year 2021-2022. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of project work for the said degree.

Mini Project Coordinator :
Dr. Prashanth C M
Professor, Dept. of CSE

Dr. M Ravishankar
Principal, DSATM

GUIDES :
Prof. K. Deepa Shree
(Asst. Prof. Dept. of CSE)

Dr. C Nandini
Vice Principal & HOD, Dept. of CSE

Examiners:
1:

Signature with Date

2:

ACKNOWLEDGEMENT

It gives me immense pleasure to present before you my project titled **"NBA SEAT AND DESTINATION BOOKING SYSTEM".** The joy and satisfaction that accompany the successful completion of any task would be incomplete without the mention of those who made it possible. I am glad to express my gratitude towards our prestigious institution **DAYANANDA SAGAR ACADEMY OF TECHNOLOGY AND MANAGEMENT** for providing us with utmost knowledge, encouragement and the maximum facilities in undertaking this project.

We wish to express a sincere thanks to our respected principal **Dr. M Ravishankar** for all the support.

We express our deepest gratitude and special thanks to **Dr. C Nandini, Vice Principal & H.O.D, Dept. Of Computer Science Engineering**, for all her guidance and encouragement.

We sincerely acknowledge the guidance and constant encouragement of our mini- project coordinator, **Dr. Prashanth C M, Professor, Dept. of CSE** and our mini – project guide,
Prof. K. Deepa Shree, Asst. Prof. Dept of CSE

JEEVAN RAJU
(1DT19CS061)

ABSTRACT

My project **NBA Seat and Destination Booking System** is designed to help fans and users book match tickets and destination tickets to watch the NBA All Star Weekend Charlotte 2022. The user can book a match ticket and is provided an option of booking a flight to the stadium destination-**Charlotte** in an efficient and an organized way.

It includes maintenance of user details, ticket (seat) details, transaction details. The system allows the user to check the seats available for the selected match and its price. The applications guides the user to book and confirm his tickets. The system asks the user to enter his details such as name, seat number, card details, e-mail id and contact number to book tickets.

This software has the facility to add new record, update existing record and delete an existing record. The main purpose of this software is to make it convenient for the users to select a NBA match, book tickets for it and confirm their flight tickets.

This project is a robust, user-friendly application, yet easy and simple to use. It is based on the concept of database connectivity i.e., connecting the java application to MySQL.

TABLE OF CONTENTS

LIST OF FIGURES IN OUTPUT SCREENSHOTS

CHAPTER 1

INTRODUCTION

1.1 Background

Considering the volumes of data that needs to be tracked and accessed, it would be very difficult to manage the accuracy and quality of data manually and deliver them accordingly. It would be almost impossible to get the details required in case of manual maintenance of data. The NBA Seat and Destination Booking System keeps a track of all the tickets booked by the user and stores them in an efficient manner.

1.2 Problem Definition

This project is aimed to reduce the manual work involved in data maintenance in the ticket details, card details and automates the NBA Seat and Destination Booking System. This project is developed mainly to simplify the manual work and allows smooth administration of the operations of ticket booking. The purpose of the project is to build a software which is user friendly, simple, fast, and cost – effective. It deals with the collection of user information, tickets etc. .

1.3 Motivation

Manual System: The system is very time consuming and lazy. This system is more prone to errors and sometimes the approaches to various problems are unstructured.

Technical System: With the help of the NBA Seat and Destination Booking System we are able to efficiently book match and flight tickets in an organized manner.

1.4 Objective

Main goal of this project is to simplify the manual operation of the customer and company services with the following advantages:

1. Faster System
2. Accuracy
3. Reliability
4. Cost Effective
5. User Friendly
6. Immediate access to the data

1.5 Scope of the project

•The user can choose any of the 4 teams and can learn about the players and the matches.

•The matches are part of a round – robin format wherein the teams face off against each other. Details of the matches are stored in the MySQL table.

•If the user is interested in booking a seat for the All-Star Weekend, 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 entering valid details.

•Once the formalities are finished, a final seat frame consisting of all the match and ticket details will be displayed automatically. The user then can either logout or book an NBA destination flight to the All Star destination through our "BOOK YOUR DESTINATION FLIGHT" option.

•If the user is interested he/she can choose a suitable flight and book the ticket and seats. After these procedures, details of the user are stored in SQL tables and the final frame consists of offers that the user is entitled to.

•All the information and details are stored in MySQL table and details from the java application are erased for the next user.

•Our MySQL database has 4 tables. The 'matches' table displays details of the matches whereas the 'flight' table displays the list of flights available and journey details. Personal information of the user along with the selected match and seats are stored in the 'user' table whereas flight details are stored in the 'air' table.

CHAPTER 2

REQUIREMENTS

The requirements can be broken down into 2 major categories namely hardware and software requirements. The former specifies the minimal hardware facilities expected in a system in which the project has to be run. The latter specifies the essential software needed to build and run the project.

2.1 Hardware Requirements

The Hardware requirements are very minimal and the program can be run on most of the machines.

- Processor - Intel 486/Pentium processor or better
- Processor Speed - 500 MHz or above
- Hard Disk - 20GB(approx)
- RAM - 64MB or above
- Storage Space - Approx. 2MB

2.2 Software Requirements

- Technology Implemented : MySQL Server
- Language Used : JAVA
- Database : My SQL
- User Interface Design : JAVA, JS
- Web Browser : Google Chrome
- IDE : NETBEANS

CHAPTER 3

SYSTEM DESIGN

3.1 ER Diagram

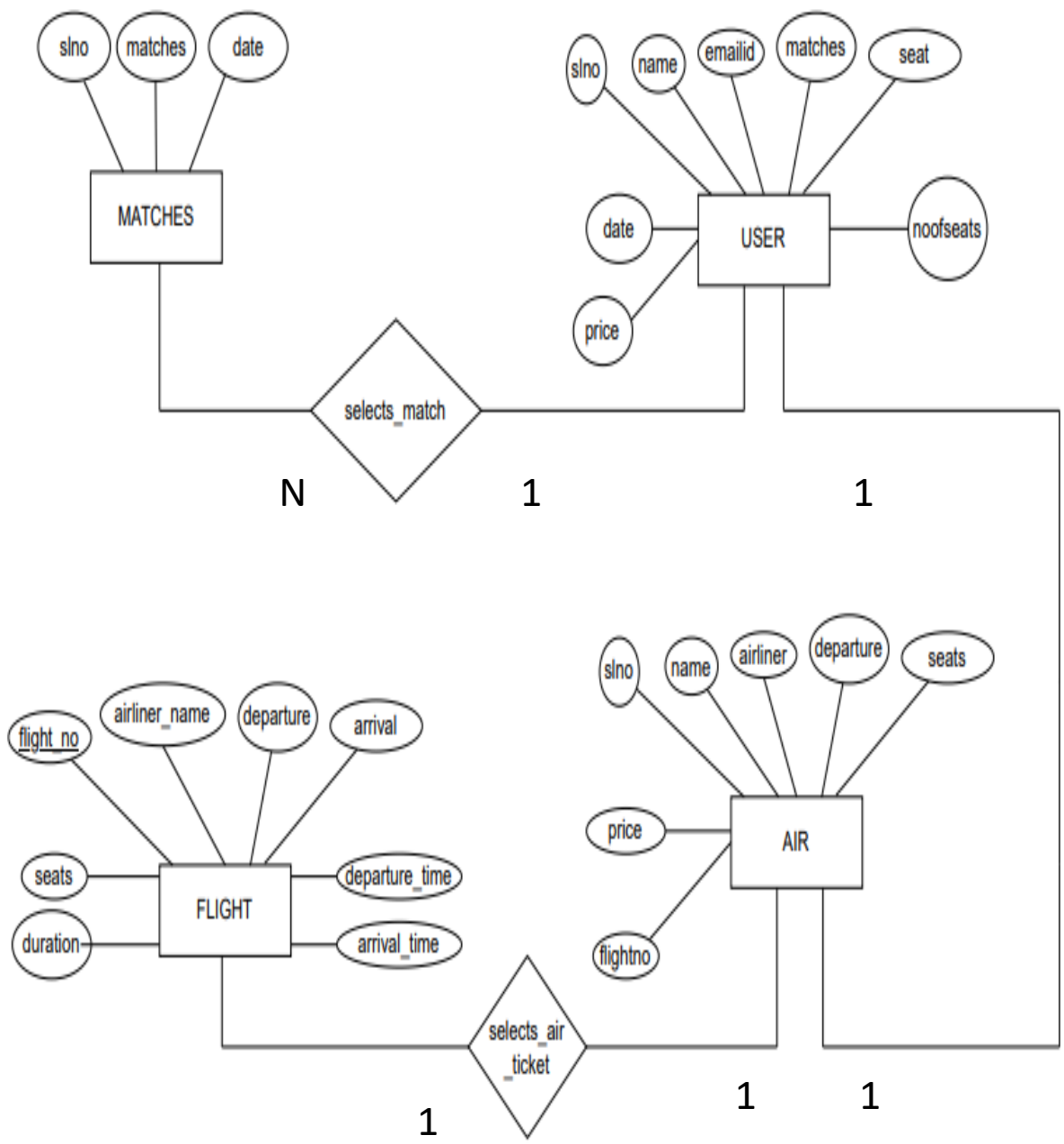


Fig:3.1

3.2 Relational Schema

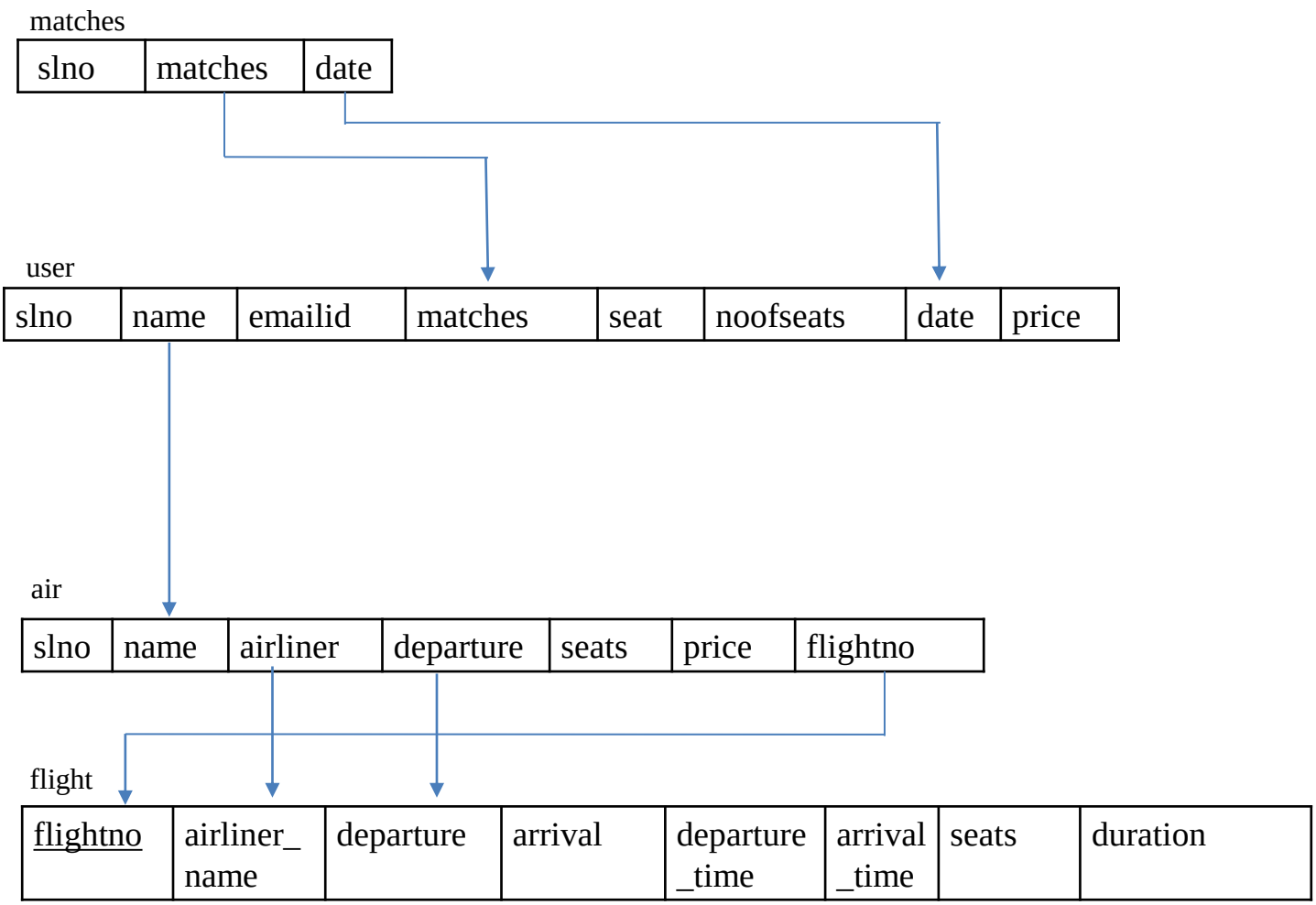


Fig:3.2

3.3 Description Of Database and Tables

Database : airline

```
mysql> use airline;
Database changed
mysql> show tables;
+------------------+
| Tables_in_airline |
+------------------+
| air              |
| flight           |
| matches          |
| user             |
+------------------+
4 rows in set (0.00 sec)
```

3.3.1 Database : airline

Table : air

```
mysql> desc air;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| slno      | int         | YES  |     | NULL    |       |
| name      | varchar(30) | YES  |     | NULL    |       |
| airliner  | varchar(30) | YES  |     | NULL    |       |
| departure | varchar(30) | YES  |     | NULL    |       |
| seats     | int         | YES  |     | NULL    |       |
| flightno  | varchar(30) | YES  |     | NULL    |       |
| price     | varchar(30) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
7 rows in set (0.03 sec)
```

Table 3.3.2 : air

Table contents : air

```
mysql> select * from air;
+------+------+----------+-----------+-------+----------+-------+
| slno | name | airliner | departure | seats | flightno | price |
+------+------+----------+-----------+-------+----------+-------+
|  100 | NULL | NULL     | NULL      |  NULL | NULL     | NULL  |
+------+------+----------+-----------+-------+----------+-------+
1 row in set (0.01 sec)
```

Table *:* flight

```
mysql> desc flight;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| flight_no      | varchar(30) | NO   | PRI | NULL    |       |
| airliner_name  | varchar(30) | NO   |     | NULL    |       |
| departure      | varchar(30) | YES  |     | NULL    |       |
| arrival        | varchar(30) | YES  |     | NULL    |       |
| departure_time | time        | YES  |     | NULL    |       |
| arrival_time   | time        | YES  |     | NULL    |       |
| duration       | time        | YES  |     | NULL    |       |
| seats          | int         | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

Table 3.3.3 *:* flight

Table contents : flight

```
mysql> select * from flight;
+-----------+---------------+-------------+-----------+----------------+--------------+----------+-------+
| flight_no | airliner_name | departure   | arrival   | departure_time | arrival_time | duration | seats |
+-----------+---------------+-------------+-----------+----------------+--------------+----------+-------+
| AC999     | Emirates      | Auckland    | Charlotte | 05:45:00       | 07:15:00     | 01:30:00 |    18 |
| BC345     | Etihad        | Bangalore   | Charlotte | 08:30:00       | 08:45:00     | 12:15:00 |    23 |
| JC567     | Jet Airways   | Johannesburg| Charlotte | 12:15:00       | 03:45:00     | 03:30:00 |    24 |
| LC234     | Indigo        | London      | Charlotte | 06:30:00       | 09:45:00     | 03:15:00 |    30 |
| SC789     | Indigo        | Sydney      | Charlotte | 09:45:00       | 04:00:00     | 06:15:00 |    35 |
+-----------+---------------+-------------+-----------+----------------+--------------+----------+-------+
5 rows in set (0.02 sec)
```

Table : matches

```
mysql> desc matches;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| slno    | int         | YES  |     | NULL    |       |
| matches | varchar(23) | YES  |     | NULL    |       |
| date    | date        | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

Table 3.3.4 : matches

Table contents : matches

```
mysql> select * from matches;
+------+-----------------------+------------+
| slno | matches               | date       |
+------+-----------------------+------------+
|  100 | Warriors vs Lakers    | 2022-01-20 |
|  101 | Celtics vs Lakers     | 2022-01-24 |
|  102 | Celtics vs Mavericks  | 2022-01-27 |
|  103 | Mavericks vs Lakers   | 2022-01-20 |
|  104 | Warriors vs Celtics   | 2022-01-20 |
|  105 | Warriors vs Mavericks | 2022-01-20 |
+------+-----------------------+------------+
6 rows in set (0.00 sec)
```

Table : user

```
mysql> desc user;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| slno      | int         | YES  |     | NULL    |       |
| name      | varchar(30) | YES  |     | NULL    |       |
| emailid   | varchar(30) | YES  |     | NULL    |       |
| matches   | varchar(30) | YES  |     | NULL    |       |
| date      | date        | YES  |     | NULL    |       |
| seat      | varchar(30) | YES  |     | NULL    |       |
| noofseats | int         | YES  |     | NULL    |       |
| seats     | varchar(30) | YES  |     | NULL    |       |
| price     | int         | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
9 rows in set (0.00 sec)
```

Table 3.3.5 : user

Table contents : user

```
mysql> select * from user;
+------+------+---------+---------+------+------+-----------+-------+-------+
| slno | name | emailid | matches | date | seat | noofseats | seats | price |
+------+------+---------+---------+------+------+-----------+-------+-------+
|  100 | NULL | NULL    | NULL    | NULL | NULL |      NULL | NULL  |  NULL |
+------+------+---------+---------+------+------+-----------+-------+-------+
1 row in set (0.00 sec)
```

CHAPTER 4

IMPLEMENTATION

4.1 Modules and Description

4.1.1 Team Selection

Process Name	**: User selects team**
Process Number	: 1.1.1
Input	: Select desired team
Output	: Starting 5 of team

Process Name	**: User selects match**
Process Number	: 1.1.2
Input	: Select desired match
Output	: Status Message
Error Condition	: Selection is Required

-

4.1.2 Stadium Seat Selection Module

Process Name	**: Seat Selection**
Process Number	: 1.2.1
Input	: Click on required seats
Output	: Status Message
Error Condition	: Limited to 5 seats per booking

4.1.3 Seat Ticket Confirmation Module

Process Name	**: User Confirms Seats**
Process Number	: 1.3.1
Input	: First name : Last name : E-mail ID : Mobile number : Credit / Debit Card
Output	: Status Message
Error Condition	: Only alphabets for names : Only digits for contact number

Process Name	**: Payment through Credit Card**
Process Number	: 1.3.2
Input	: Name on card : Card number : Expiry date
Output	: Payment confirmed
Error Condition	: Only alphabets for name : 16 digit card number : Card expired in January 2022

Process Name	**: Payment through Debit Card**
Process Number	: 1.3.3
Output	: Name on card : Card number : Expiry date
Output	: Payment confirmed
Error Condition	: Only alphabets for name : 16 digit card number : Card expired in January 2022 : 3 digit CVV

4.1.4 Air Ticket Reservation Module

Process Name	**: Flight Ticket Booking**
Process Number	:1.4.1
Input	: NBA credits : Departure airport
Output	: Status Message
Error Condition	: Invalid departure airport

Process Name	**: Airplane Booking**
Process Number	: 1.5.1
Input	: Select Airline and preferred flight
Output	: Status Message
Error Condition	: Maximum 4 seats per booking

4.2 Triggers and Stored Procedures

4.2.1 Stored Procedure :

The Stored procedure implemented in the project for :

(i) helps user to view seat and available flight details.

(ii)When the user confirms on his/her seat selection, they can complete their payment.

4.2.2 Trigger :

The trigger is implemented for :

The trigger is used to display a pop-up message when an error/invalid condition arises, these happen when constraints are violated.

4.3 Database Connectivity

A database connection is a facility in computer science that allows client software to talk to database server software, whether on the same machine or not. A connection is required to send commands and receive answers, usually in the form of a result set. Connections are a key concept in data-centric programming.

DB Connectivity

```
import java.sql.*;
class MysqlCon{
public static void main(String args[]){
try{
Class.forName("com.mysql.jdbc.Driver");
Connection con=DriverManager.getConnection(
"jdbc:mysql://localhost:3306/airline","root","root");
//here airline is database name, root is username and password
Statement stmt=con.createStatement();
ResultSet rs=stmt.executeQuery("select * from emp");
while(rs.next())
System.out.println(rs.getInt(1)+"  "+rs.getString(2)+"  "+rs.getString(3));
con.close();
}catch(Exception e){ System.out.println(e);}  }
}
```

To connect a Java application with the MySQL database, we need to follow the following steps.

In this example we are using MySql as the database.

1.Driver class : The driver class for the mysql database is **com.mysql.jdbc.Driver**.

2.Connection URL : The connection URL for is **jdbc:mysql://localhost:3306/airline** where jdbc is the API, MySQL is the database, localhost is the server name on which MySQL is running, we may also use IP address, 3306 is the port number and airline is the database name.

3.Username : The default username for the MySQL database is **root**.

4.Password : It is the password given by the user at the time of installing the MySQL database. In this example, we are going to use root as the password.

4.3 SOURCE CODE

Sample Code: ticketFrame

```
import java.sql.*;
import javax.swing.JOptionPane;/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author student
 */
public class ticketFrame extends javax.swing.JFrame {

    /**
     * Creates new form ticketFrame
     */
    public ticketFrame() {
        initComponents();
    }
```

```
 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
new seatingFrame().setVisible(true);
dispose();
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
if(jRadioButton1.isSelected()==true)
{
    new creditcardFrame().setVisible(true);
    dispose();
}
if(jRadioButton2.isSelected()==true)
{
    new debitcardFrame().setVisible(true);
    dispose();
}
try {
String n1 = jTextField1.getText();
String n2 = jTextField3.getText();
Class.forName("java.sql.Driver");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost/airline","root","student");
Statement stmt = con.createStatement();
String query="update user set name='"+n1+"', emailid='"+n2+"' where slno=100;";
stmt.executeUpdate(query);
con.close();
stmt.close();
}
catch(Exception e)
{
    JOptionPane.showMessageDialog(null,"Error");
}
    }
```

```
 private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        jTextField1.setText(" ");
        jTextField2.setText(" ");
        jTextField3.setText(" ");
        jTextField4.setText(" ");
    }

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    try
{
Class.forName("java.sql.Driver");
Connection con=DriverManager.getConnection("jdbc:mysql://localhost/airline","root","student");
Statement stmt=con.createStatement();
String query="select matches,date,seats,price,noofseats from user where slno=100";

ResultSet rs=stmt.executeQuery(query);

if(rs.next())
{

String n1=rs.getString("matches");
String n2=rs.getString("date");
String n3=rs.getString("seats");
String n4=rs.getString("price");
String n5=rs.getString("noofseats");
jTextField5.setText(""+n5);
jTextField8.setText(""+n1);
jTextField7.setText(""+n2);
jTextField6.setText(""+n3);
jTextField10.setText(""+n4);

}

rs.close();
con.close();
stmt.close();
}
            catch(Exception e)
           {
           JOptionPane.showMessageDialog(null,"Error");

           }
    }
```

```
private void jTextField1KeyTyped(java.awt.event.KeyEvent evt) {
char c = evt.getKeyChar();
if(Character.isDigit(c)){
   evt.consume();
}
   }

   private void jTextField2KeyTyped(java.awt.event.KeyEvent evt) {
char c = evt.getKeyChar();
if(Character.isDigit(c)){
   evt.consume();
}
   }

   private void jTextField4KeyTyped(java.awt.event.KeyEvent evt) {
char c = evt.getKeyChar();
if(!Character.isDigit(c)){
   evt.consume();
}
   }
```

CHAPTER 5

TESTING

5.1 Testing

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

TESTING OBJECTIVES:

1. Testing is process of executing a program with the intent of finding an error.
2. A good test case design is one that has a probability of finding an as yet undiscovered error.
3. A successful test is one that uncovers an as yet undiscovered error.

These above objectives imply a dramatic change in view port.

Testing cannot show the absence of defects, it can only show that software errors are present. There are three types of testing strategies

1.Unit test

2.Integration test

3.Performance test

5.2 Unit Testing:

Unit testing focuses verification efforts on the smallest unit of software design module. The unit test is always white box oriented. The tests that occur as part of unit testing are testing the module interface, examining the local data structures, testing the boundary conditions, execution all the independent paths and testing error-handling paths.

5.3 Integration Testing:

Testing is a systematic technique or construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. Scope of testing summarizes the specific functional, performance, and internal design characteristics that are to be tested. It employs top-down testing and bottom-up testing methods for this case.

5.4 Performance Testing:

Timing for both read and update transactions should be gathered to determine whether system functions are being performed in an acceptable timeframe.

5.5 Module Testing

Module testing is defined as a software testing type, which checks individual subprograms, subroutines, classes, or procedures in a program. Instead of testing whole software program at once, module testing recommends testing the smaller building blocks of the program. Module testing is largely a white box oriented. The objective of doing Module, testing is not to demonstrate proper functioning of the module but to demonstrate the presence of an error in the module.

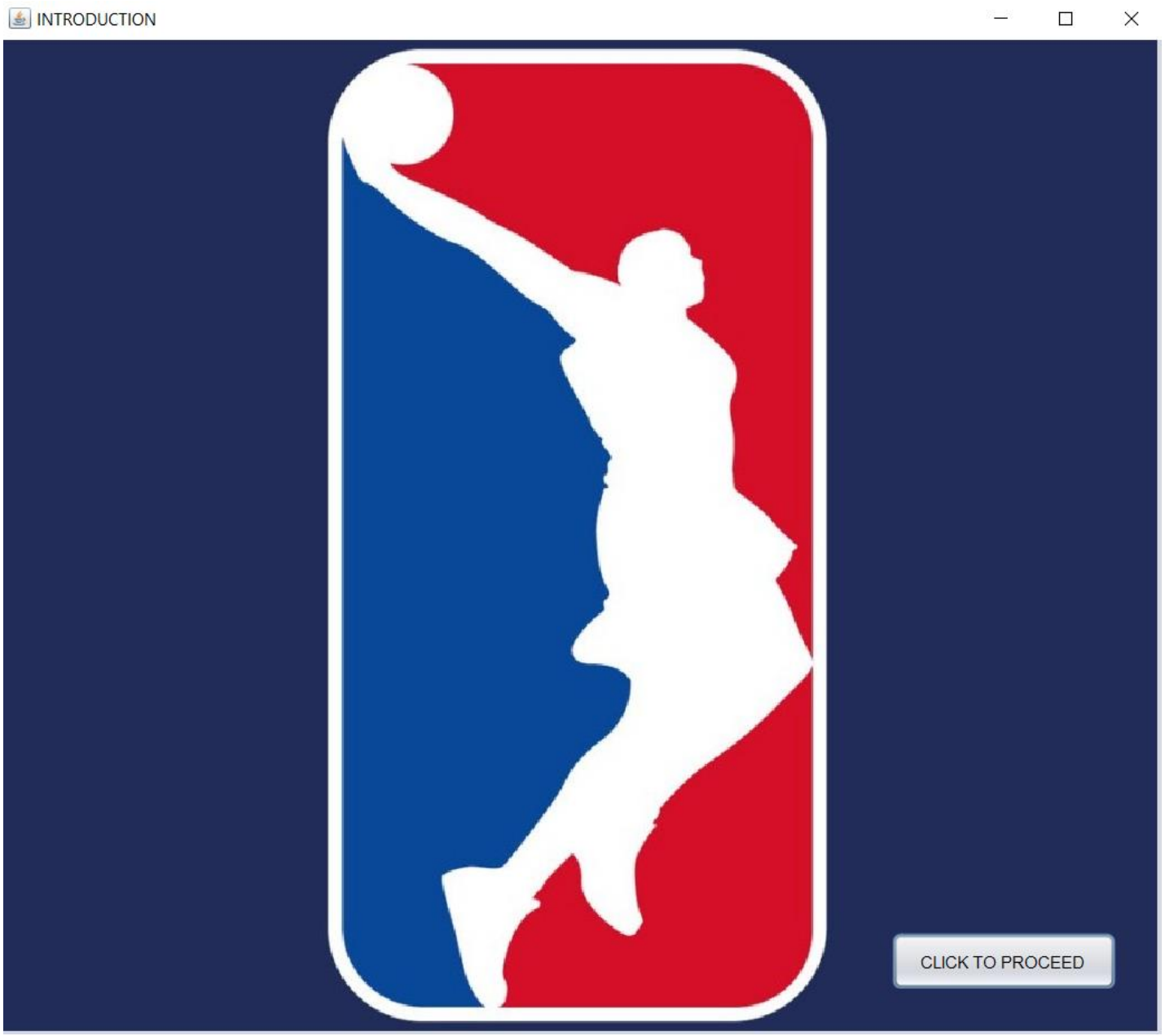
5.6 System Testing

System testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems. System testing is actually a series of different tests whose sole purpose is to exercise the full computer-based system.

CHAPTER 6

SCREENSHOTS

Introduction Frame

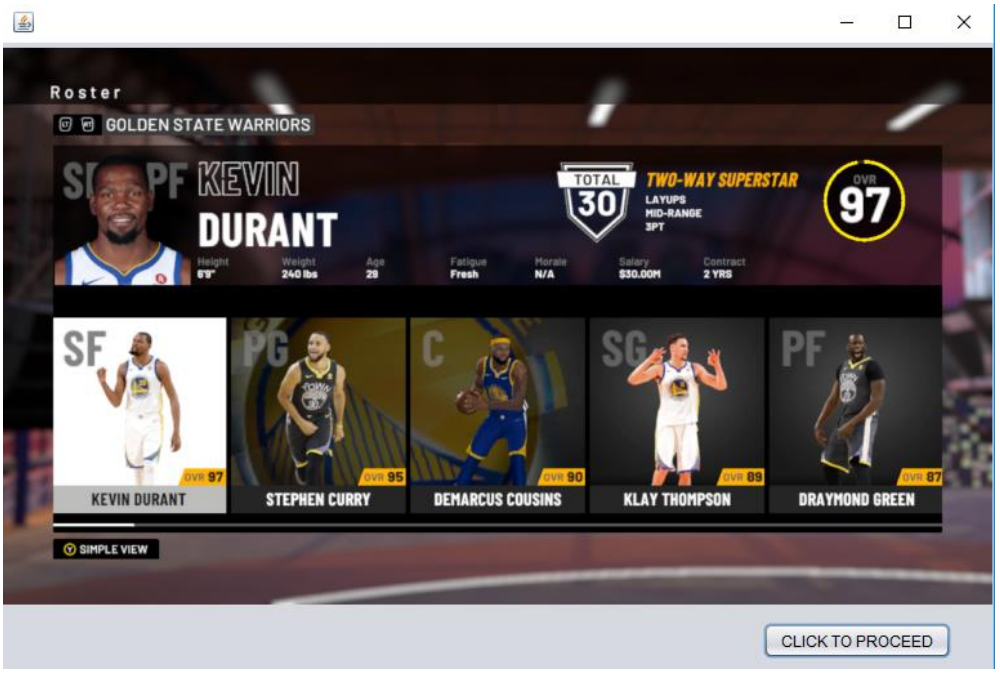


Screenshot 6.1 : Introduction Frame

NBA Teams Frame

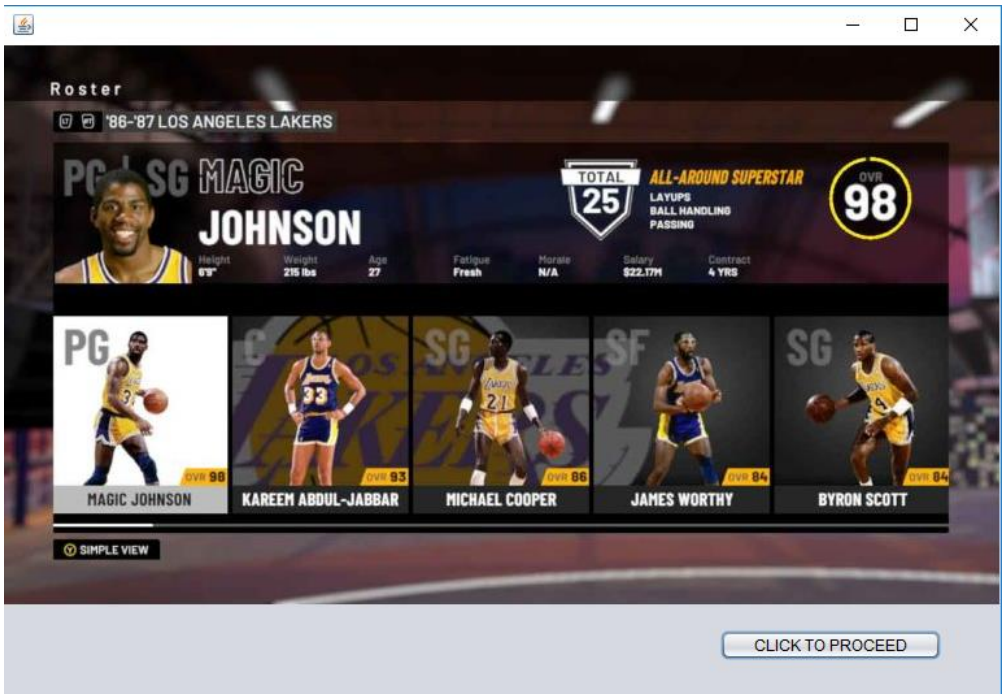
Screenshot 6.2 : NBA Teams Frame

GOLDEN STATE WARRIORS FRAME



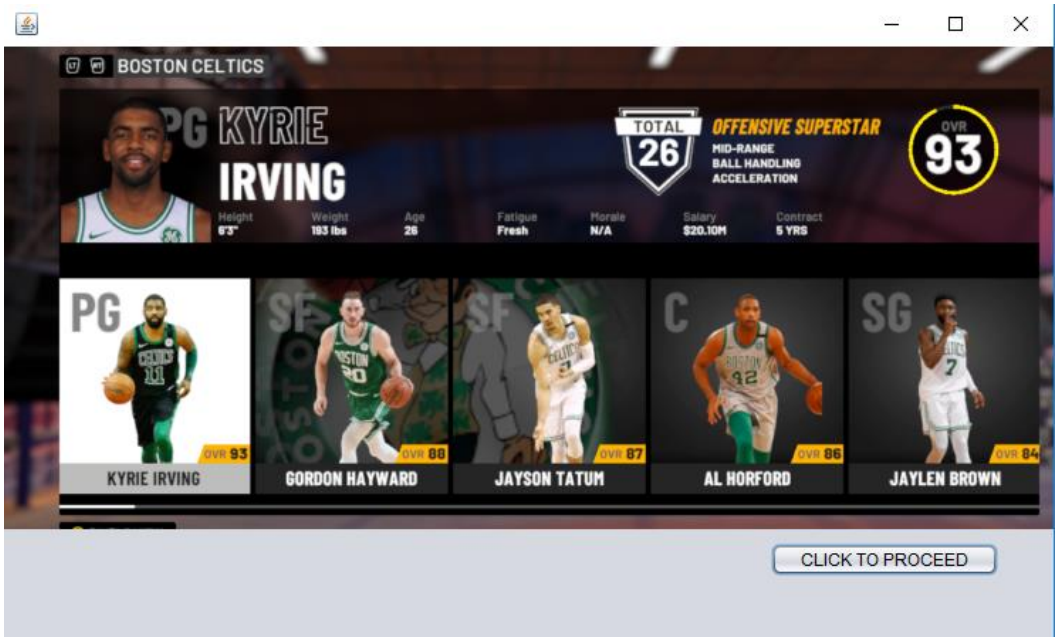
Screenshot 6.3 : warrFrame

LOS ANGELES LAKERS FRAME



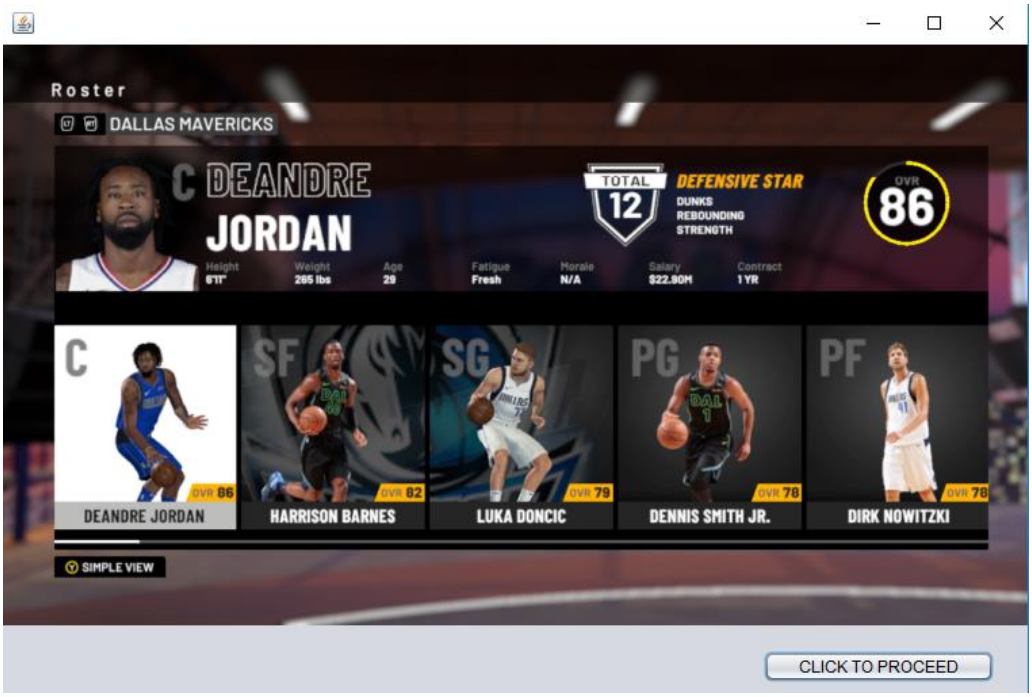
Screenshot 6.4 : LakersFrame

BOSTON CELTICS FRAME



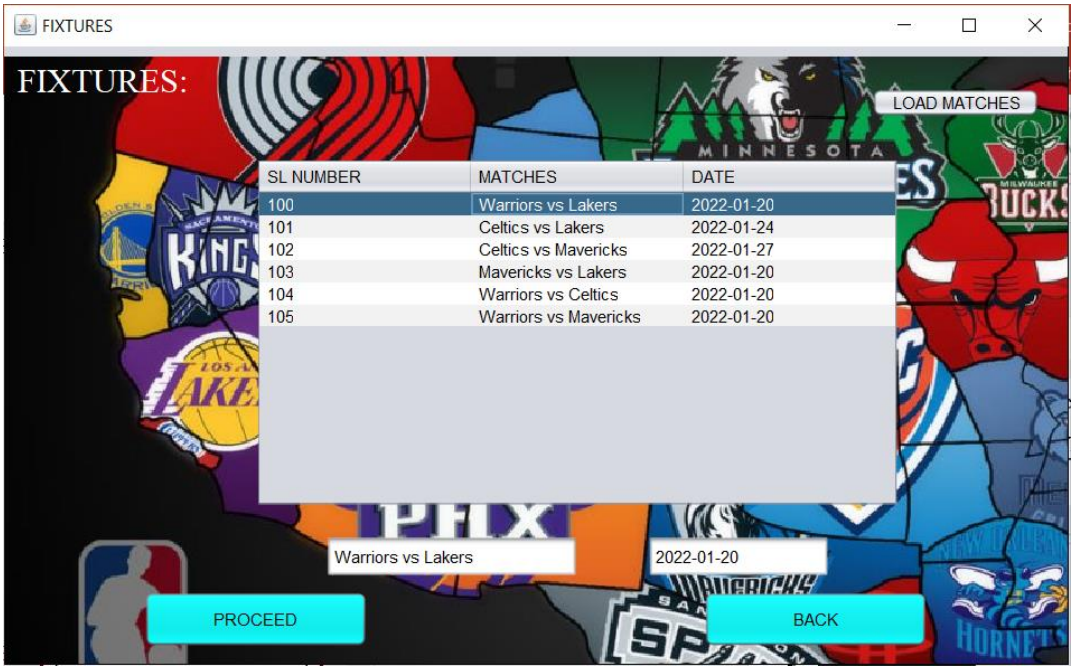
Screenshot 6.5: celticsFrame

DALLAS MAVERICKS FRAME



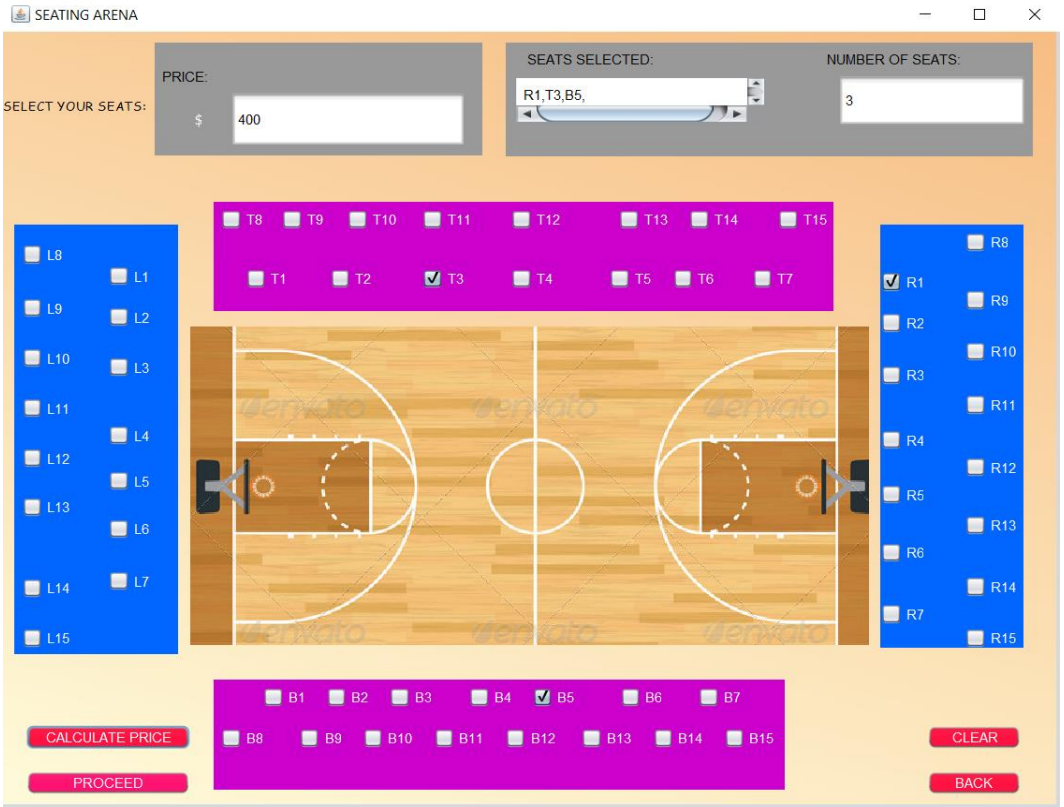
Screenshot 6.6 : maveriksFrame

FIXTURES FRAME



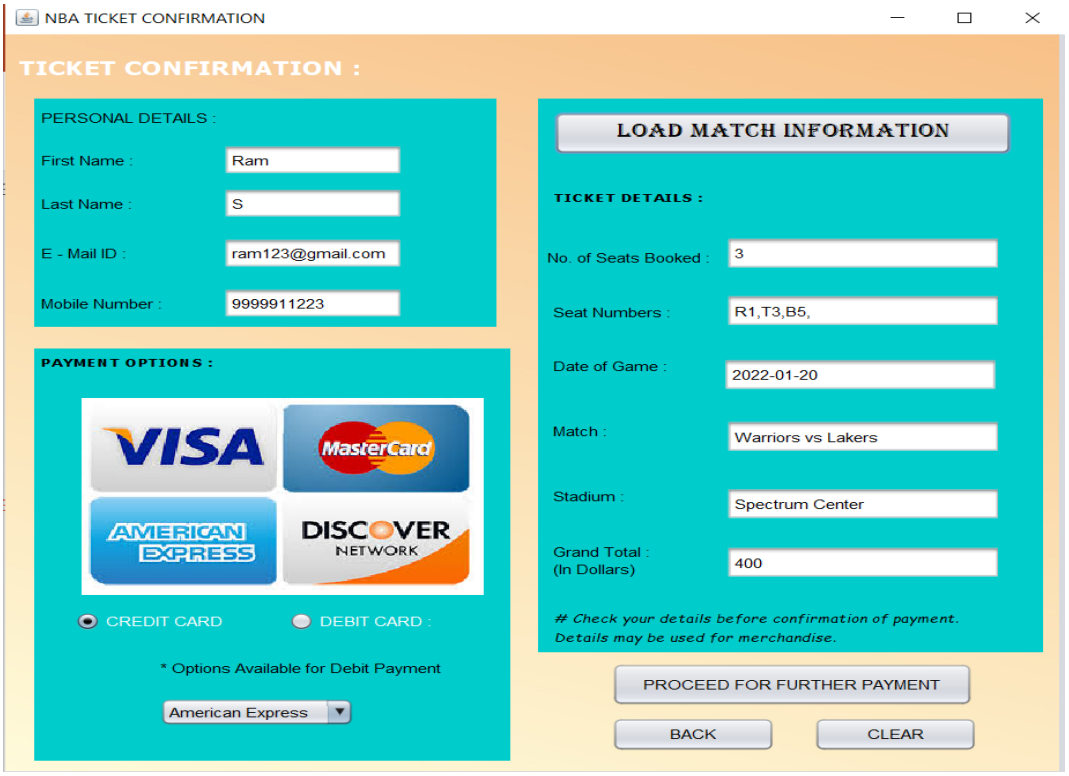
Screenshot 6.7 : fixturesFrame

SEATING ARENA FRAME



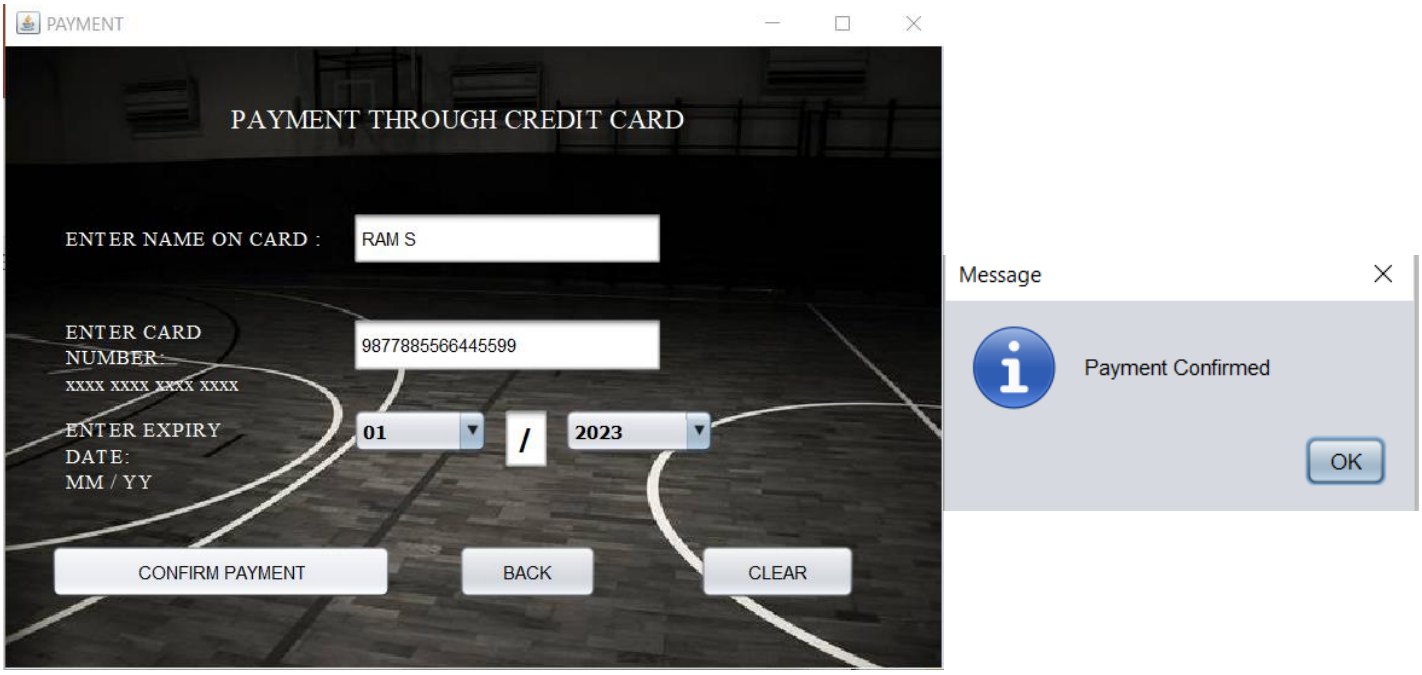
Screenshot 6.8 : seatingFrame

NBA TICKET CONFIRMATION FRAME



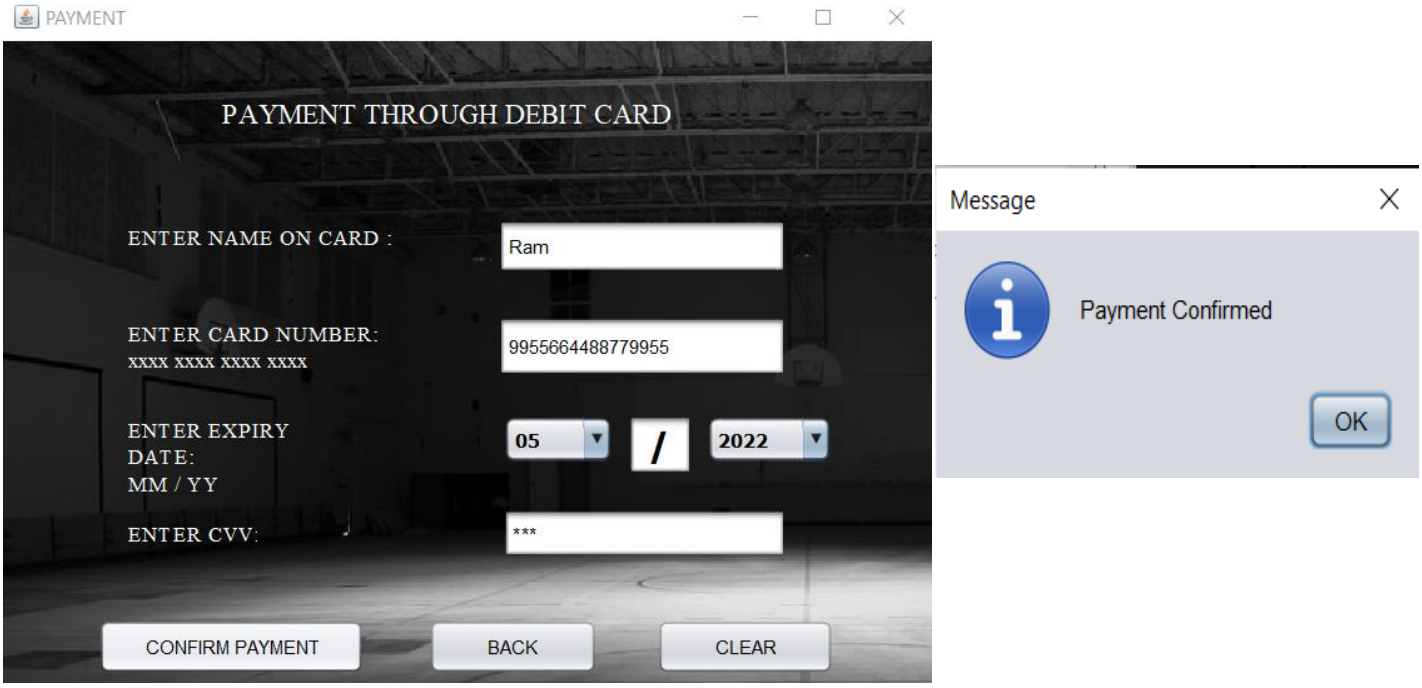
Screenshot 6.9 : ticketFrame

CREDIT CARD PAYMENT FRAME



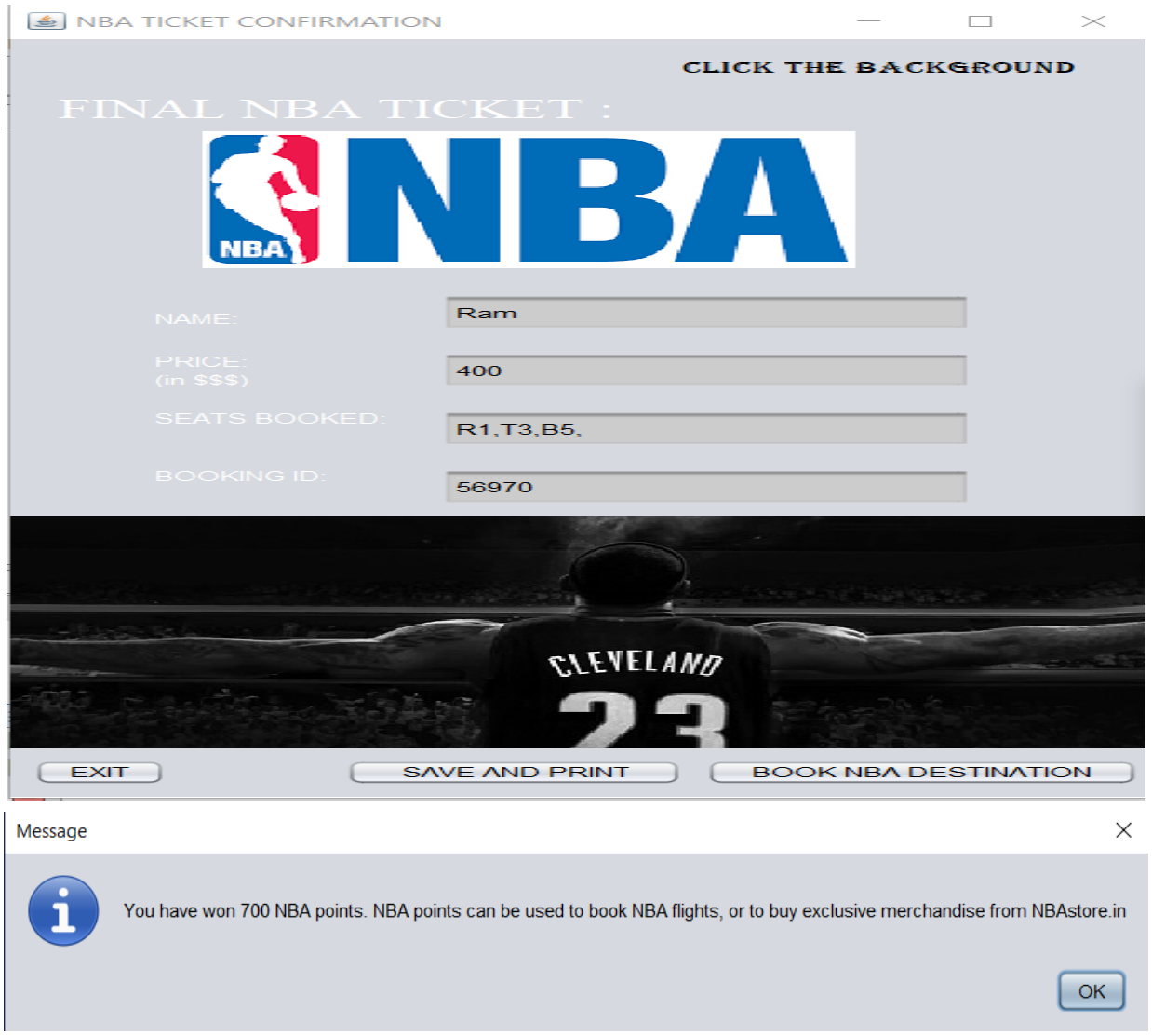
Screenshot 6.10 : creditcardFrame

DEBIT CARD PAYMENT FRAME



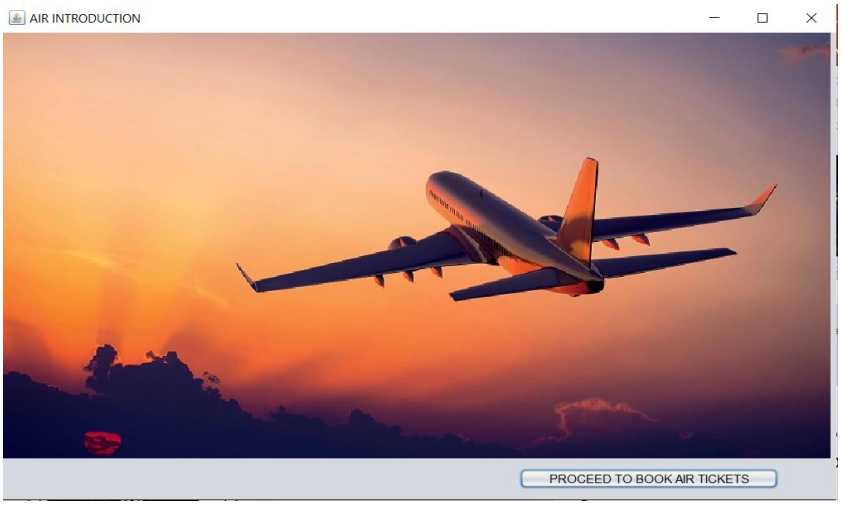
Screenshot 6.11 : debitcardFrame

NBA CONFIRMATION FRAME



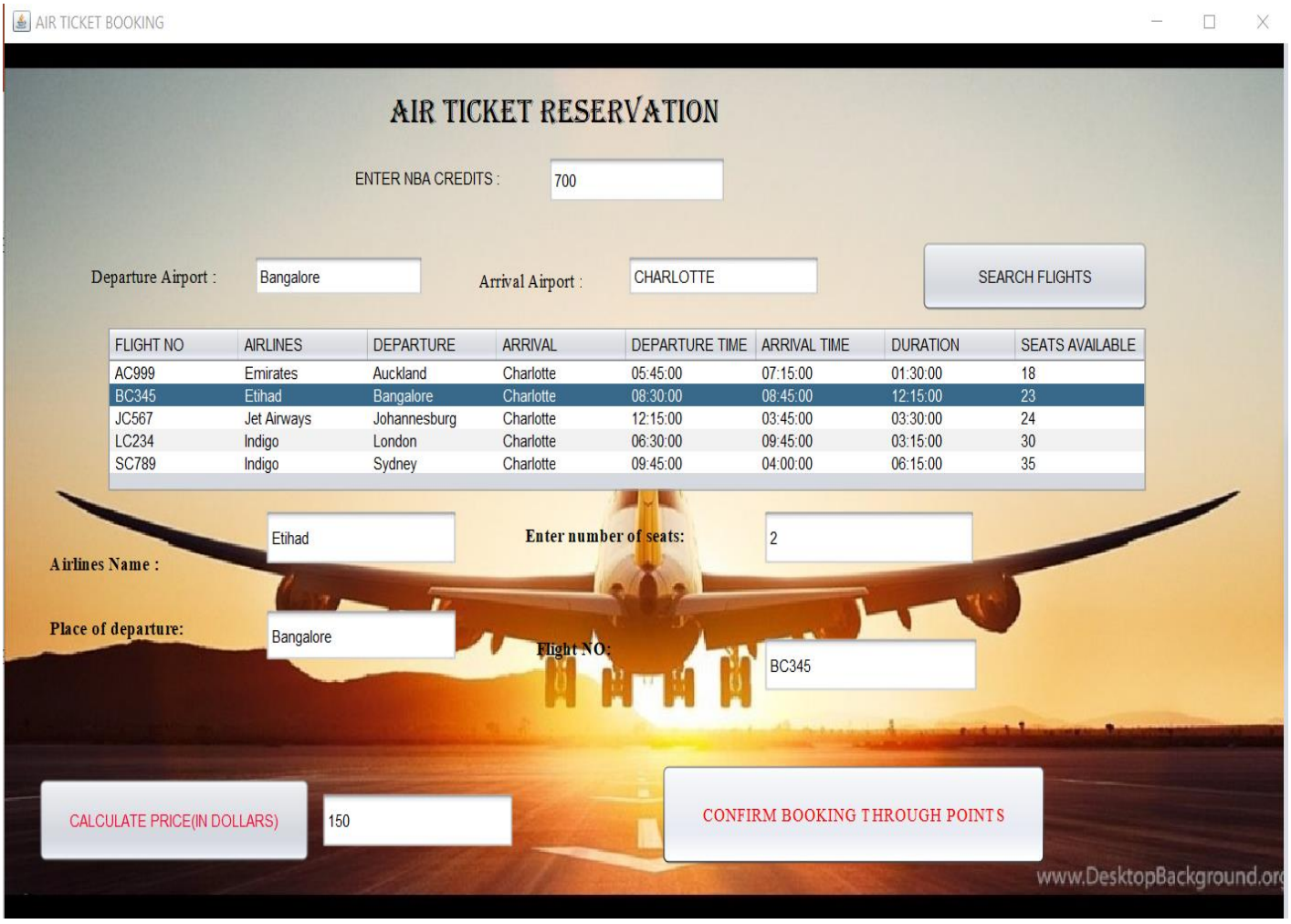
Screenshot 6.12 : confirmationFrame

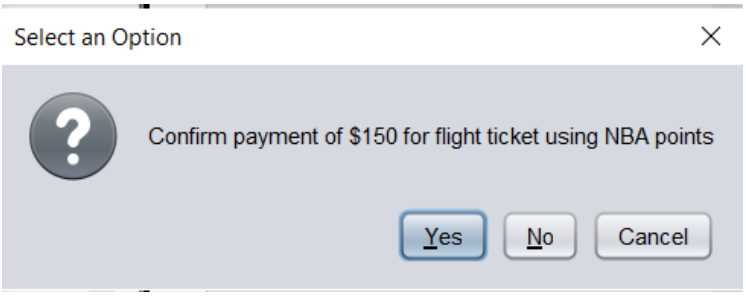
AIR INTRODUCTION FRAME



Screenshot 6.13 : airplaneIntroFrame

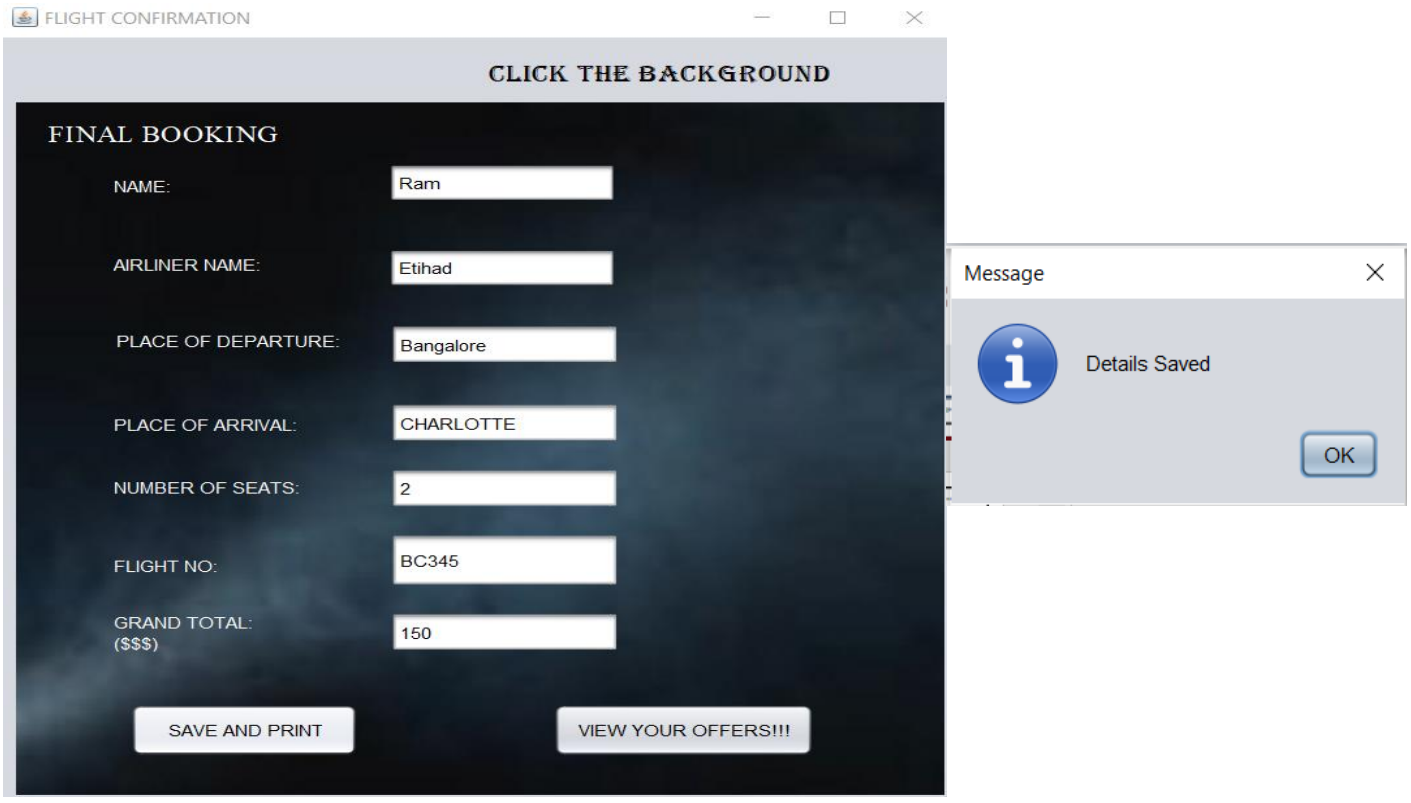
AIR TICKET BOOKING FRAME





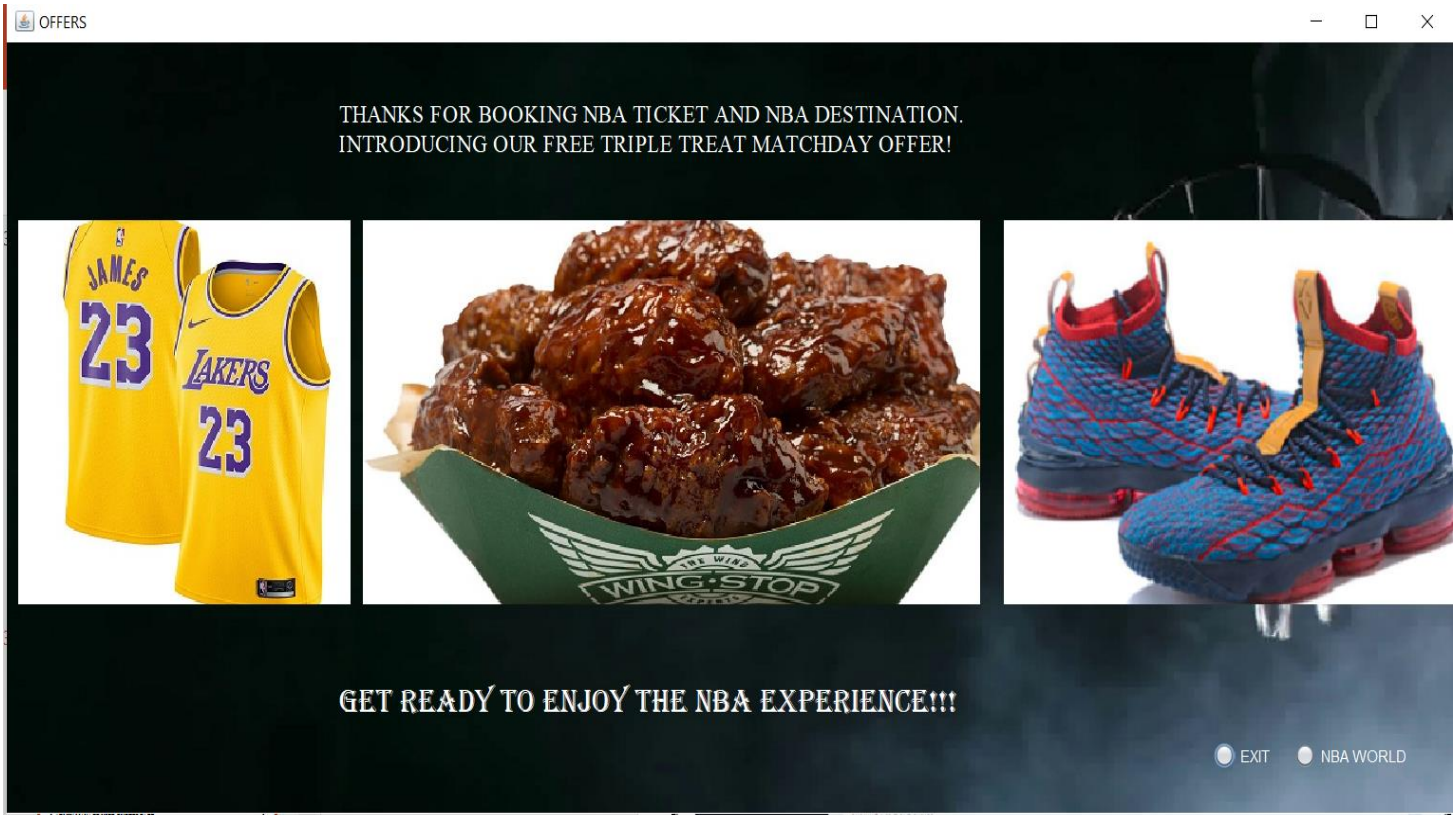
Screenshot 6.14 : flightFrame

FLIGHT TICKET CONFIRMATION FRAME



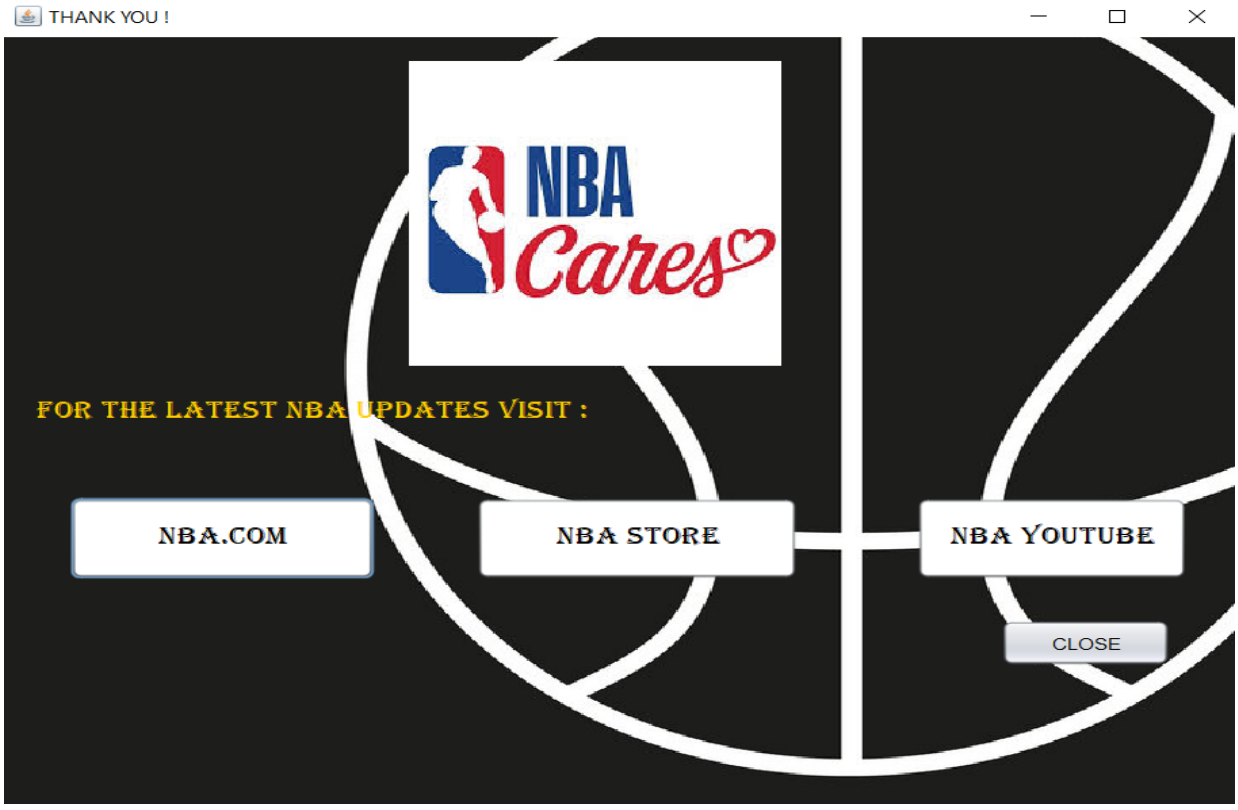
Screenshot 6.15 : flightConfirmationFrame

MATCHDAY OFFERS FRAME



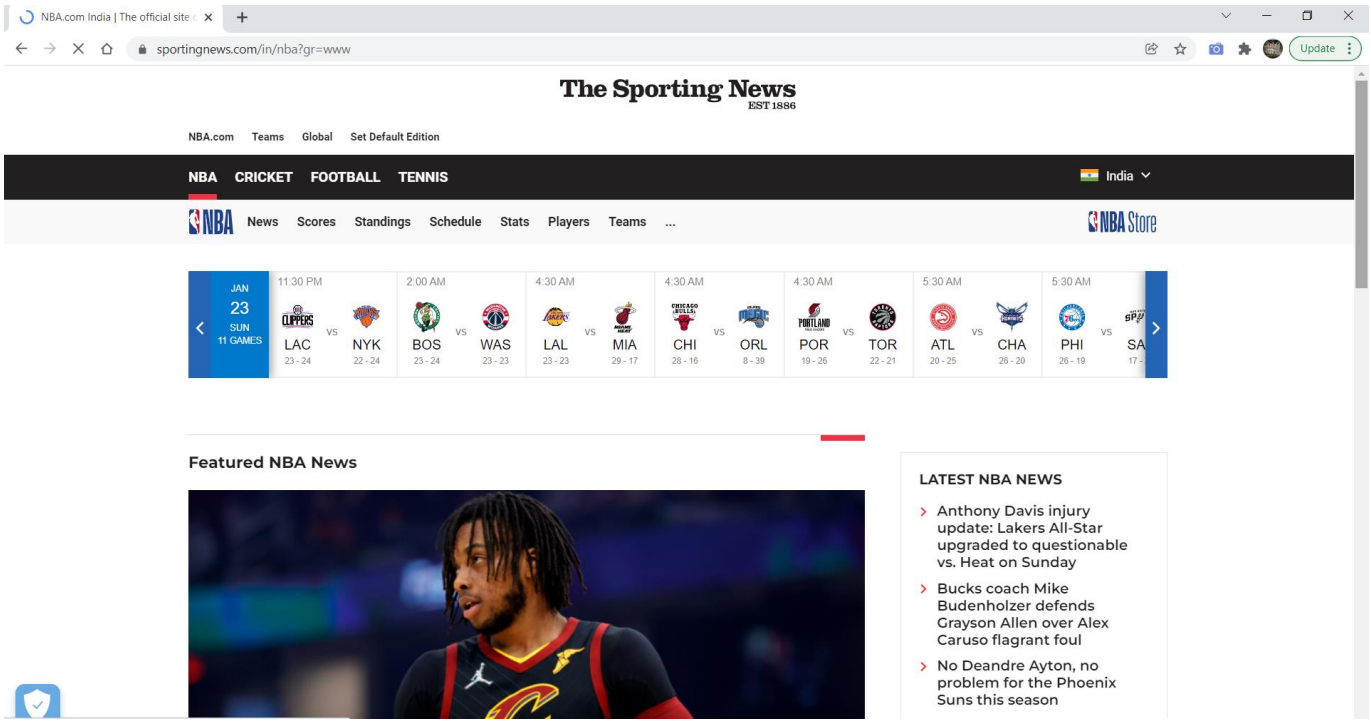
Screenshot 6.16 : offersFarme

FINAL FRAME



Screenshot 6.17 : thankYouFrame

NBA OFFICIAL WEBSITE



Screenshot 6.18 : Official NBA Page

CHAPTER 7

CONCLUSION AND FUTURE WORK

The NBA Seat and Destination Booking System is a great improvement over the manual system which uses lots of manual work. The computerization of the system helps speed up the process. This system was thoroughly checked and tested with multiple entries of data and found to be very reliable way to handle seat and air ticket booking efficiently.

Advantages

- The NBA Seat and Destination Booking System is fast, efficient and reliable.
- Avoids data redundancy and inconsistency.
- Web-based.
- Any number of users can use it.
- Provides more security and integrity to data.
- Increases the efficiency of the company.
- Everything is recorded and organized.

Future Enhancements

The NBA Seat and Destination Booking System can be enhanced by including additional functionalities like:

Sales Management : Integration software with accounting software can be a big game-changer for this enterprise. The software will analyze past data of your business and generate sales forecasts automatically. Thus, you can easily have 30% better forecasts than before. As all the sales data are available in a centralized location, accessibility improves. You can then utilize this data and boost the overall productivity of your business.

Customer Interaction : Customer is always the important aspect irrespective of the size, type, and nature of business. Therefore, you can retain your business growth by attracting new customers and retaining the old ones. The Customer Database feature compiles all customer-related data in a central place. Therefore, you can facilitate better customer interaction by analyzing emails, seat preferences and payment methods of past transactions. It not only helps you to improve your customer relationship but also nurture it for the future.

REFERENCES

For JAVA

- https://www.javatpoint.com/java-jframe
- https://www.w3schools.com/java/default.asp
- https://www.tutorialspoint.com/java/index.htm

For MySQL

- https://www.mysql.com/
- http://www.mysqltutorial.org

For JDBC

- https://www.javatpoint.com/example-to-connect-to-the-mysql-database
- https://dev.mysql.com/doc/connector-j/5.1/en/connector-j-usagenotes-connect-drivermanager.html

For Images

- Google Images
- Shutterstock
- Getty Images