ITERATION-3

MOVEOUT SPORTS

By : Aarya Reddy
Shobhit Jain
ShreyVakharia
Vinyas Raju
Yeshwant Ranka

Table of Contents

1. Project Description

In this 21st century, people are running in a rat race where they are forgetting the importance of health and fitness. Sports events should be promoted as they help people keep a better health, physique and develop healthy team spirit. Hence, comes the need of managing sports events for proper scheduling of events and organizing them in an efficient manner. Sporting events also play a huge part in generating income not only for the sports organization, but also for local restaurants, hotels, taverns, and airports near the event. Attending a sporting event can be extremely fun, but there's plenty of hard work behind the scenes that must be done first. That's where sporting event planners and coordinators come into the picture. Olympic summer and winter games, FIFA World Cup, ICC, UEFA, all these sport events require investing of money and expertise and involving large numbers of people. Also, to organize such events in planned and efficient manner requires a lot of hard work and expert guidance. Apart from these large sporting events there are many mid-level events which require proper planning and execution. There comes the sporting event coordinators who work under the pressure and responsibility of arranging every detail involved in preparing for an event. Event planner works with different teams to organize, plan, integrate every aspect and look for every possible problem which can occur while organizing and managing an event. Teams are involved in venue implementation, planning lodging of the teams, purchasing transportation, creating emergency facilities, managing cleaning department, maintaining the security, marketing ticket sales, managing sponsors, seating arrangement for the guests and viewers, setting general amenities for the crowd and delegating the responsibilities to the assigned workers. Sporting event coordinators must plan carefully to keep venues operating efficiently and minimize wasted expenses. Lastly, event organizers must carefully assess every part of the event and check the feasibility. Also, organizers must look in what departments they can cut costs and improve the productivity. The coordinators must create alternative plans in case of disruption of a particular department and check whether all the requirements are fulfilled up to the expectations and develop mitigation plans accordingly.

2. Scope

Sports help people to achieve teamwork and make one disciplined. It helps in socializing with people from different communities and places. Moreover, participating in sports contributes to academic success. It imparts qualities likes working in cohesion and building team spirit. Participating in sports competitions help students to overcome challenges and achieve their goal. It assists in demonstrating determination and perseverance which are required for academic success. Hence, it is important for the students to participate in sports competitions for overall development. However, without proper planning and organization, it would become difficult to conduct a big sports event. Also, it would have lot of ambiguity, wastage of resources and chaotic situation for all. Sports event Management system helps in achieving this goal. It organizes various sports to gather players from various places, which helps the players to participate and showcase their talent. Our sports management system helps in proper and smooth management when sports events are being held.

3. Requirements

R1-The system must provide registration to new users and login to existing users.

R2- System must display a full exhaustive list of all events being hosted.

R3- System must maintain records and display all event related details.

R4- System must help user to search results after applying selection filters based on event type.

R5- System must display search results based on selected criteria

R6- System must allow administrator to get full access of the system to add an event.

R7- System must allow the users to register for a sports event.

R8- System must allow the administrator to get full access of the system to delete an event.

R9- System should display confirmation to the player.

4. Plan for each iteration

Iteration 1 (Already Implemented)

- Requirements analysis.
- UML DESIGNING
- Use Case diagram Modelling
- Use Case Diagrams
- Front end Designing– Login Page , Home Page, Event Page, Search Page
- Connecting Eclipse with MYSql WorkBench
- Populating database with test data
- Giving 2 types of Login- User and Admin; Verifying Credentials and Authenticating users
- Testing

Iteration 2 (Already Implemented)

- Iteration1
- Re-Scoping of Functionalities based on time.
- UML Re-designing.
- Revised Use Models.
- Revised Extended Use Cases.
- Front End Designing-Improved (Login Page, Event Page, Search Page)
- Signup/Registration For New Users
- Registration for events and simultaneously updating in database.
- Retrieving details of the events for which the user has registered.

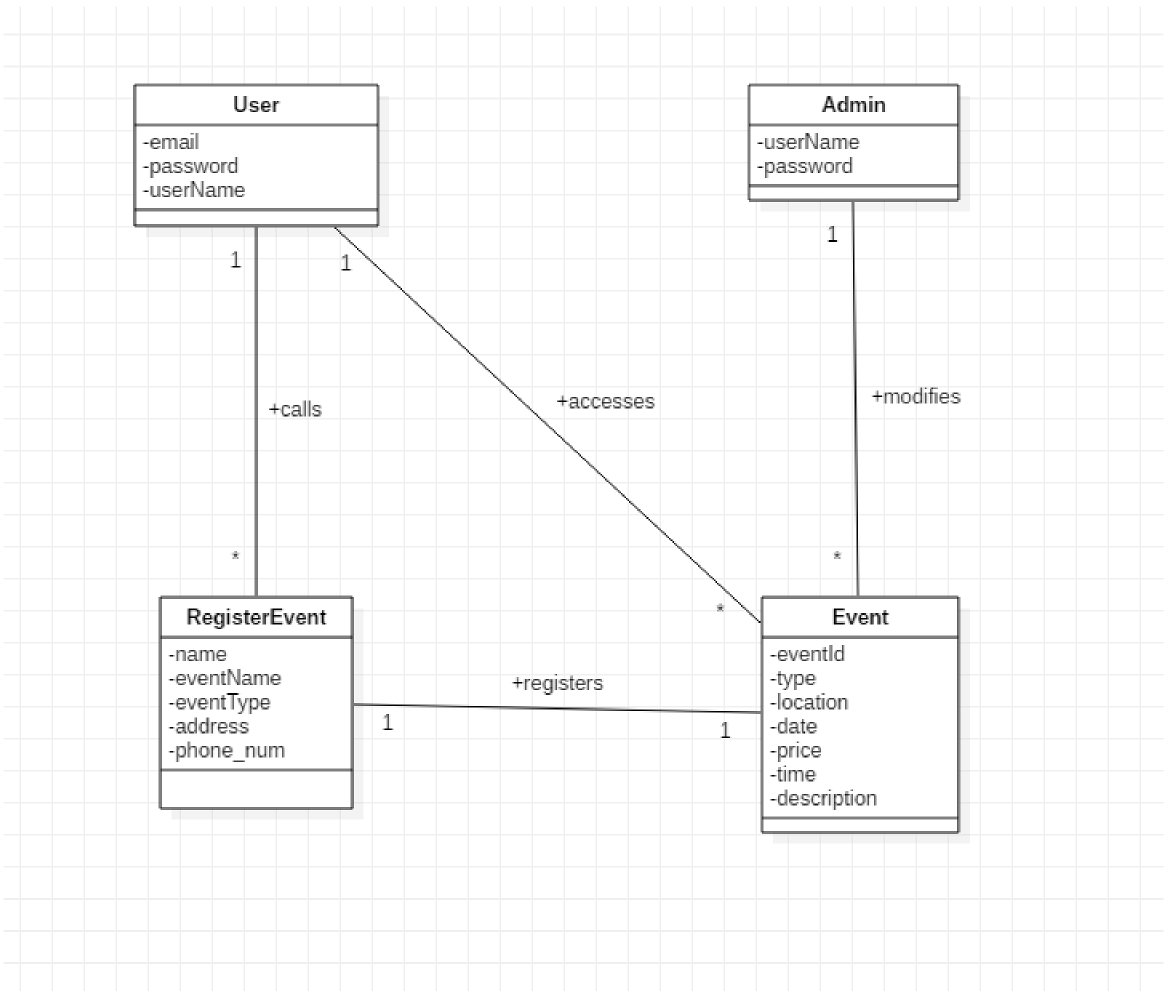
Iteration 3 (Current)

- **Iteration 1 + Iteration 2 Integrating new code into the existing system.**
- **UML Designing, Use Case Diagram Modelling.**
- **Database of the clients and the current events are modified according to the changed specifications.**
- **Front End Designing-Improved (Login Page, Event Page, Search Page)**
- **Performing "Search Operation" on database when user wants to search an event**
- **Unit testing and integration testing is done on the final product.**

For the first two iterations, the tasks were divided as follows –

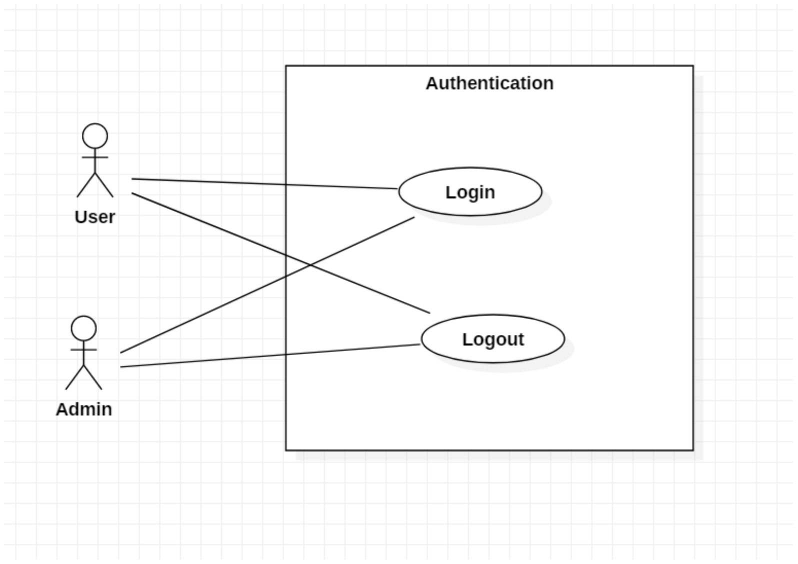
	TASKS	**ASSIGNED TO**	**STATUS**
Iteration 1			
1	Designing the database schema and initial database	Aarya Reddy Shobhit Jain	Completed
2	UML Designing, Use Case Diagram Modelling, Domain Model	Vinyas Raju Yeshwant Ranka	Completed
3	Front End Designing- Login Page, Home Page, Event Page, Search Page	Vinyas Raju	Completed
4	Connecting Eclipse with MYSQL WorkBench	Aarya Reddy Yeshwant Ranka	Completed
5	Populating Database with Test Data	ShreyVakharia	Completed
6	Sign In Feature to Users	Shobhit Jain	Completed
Iteration 2			
1	Re-scoping of functionalities based on time	Shobhit Jain ShreyVakharia	Completed
2	UML Re-Designing- Domain Modelling. Sequence Diagram	Yeshwant Ranka Aarya Reddy	Completed
3	Class Diagram	Vinyas Raju	Completed
4	Revised Use Case Diagrams and Extended Use Cases	Vinyas Raju Shobhit Jain	Completed
5	Website Front End Designing – Improved UI (Login Page, Event Page, Search Page)	Shobhit Jain Vinyas Raju	Completed
6	Sign Up, Registration for new Users	Yeshwant Ranka	Completed
7	Registration of Events and Simultaneously updating on DB	Vinyas Raju Aarya Reddy ShreyVakharia	Completed
8	Retrieving details of teams who registered for an Event , Testing (JUnit)	Yeshwant Ranka ShreyVakharia	Completed
Iteration 3			
1	UML Re-Designing- Domain Modelling. Sequence Diagram	Vinyas Raju Shobhit Jain	Completed
2	Class Diagram-Redesigning	Aarya Reddy Yeshwant Ranka	Completed
3	Extending Database	Shobhit Jain Yeshwant Ranka	Completed
5	Front End Designing(Improved)	Vinyas Raju Aarya Reddy	Completed
6	Performing search Operation	ShreyVakharia Vinyas Raju Aarya Reddy	Completed
7	Testing on Use Cases (Junit)	ShreyVakharia Shobhit Jain	Completed

5. Domain Model

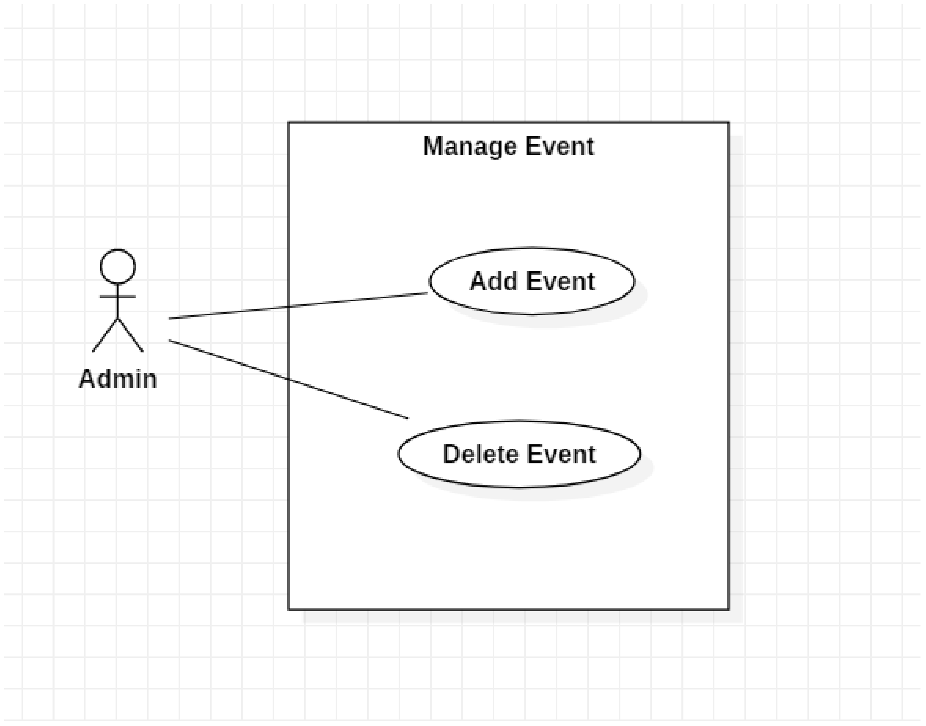


6.1 Use Case Model

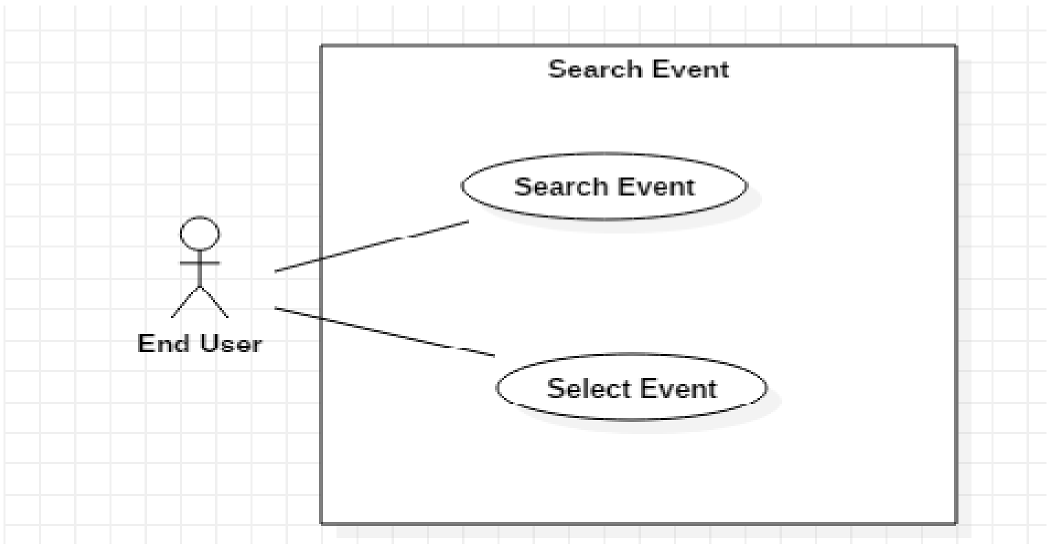
6.1.1 UC1: User Session:-



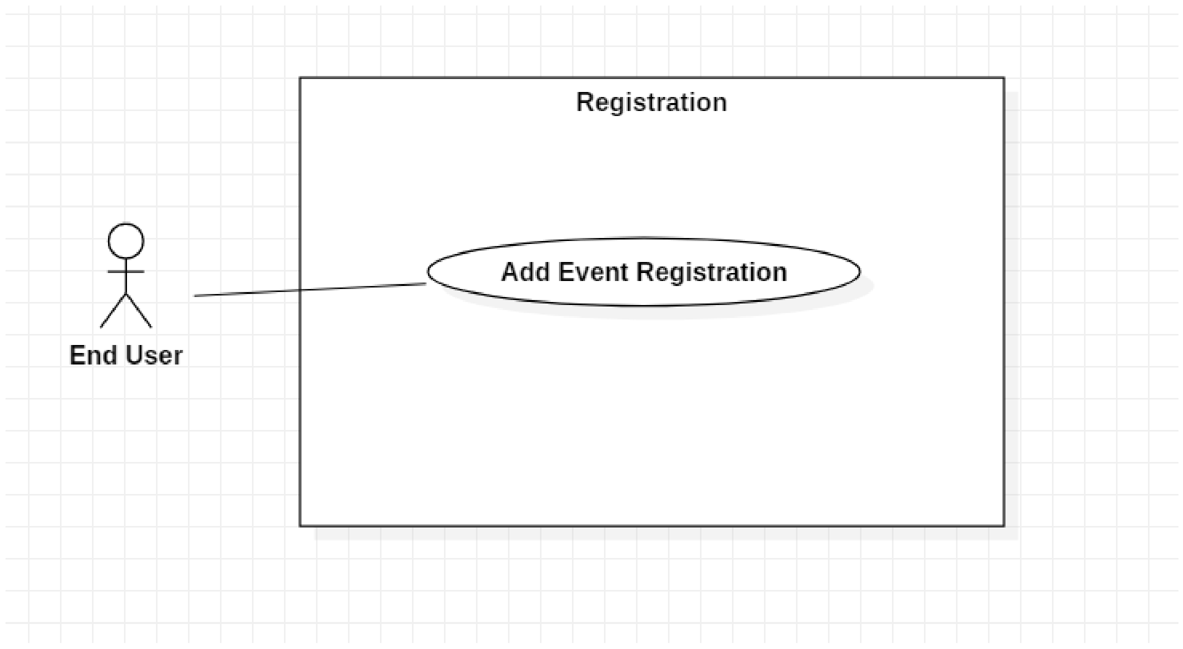
6.1.2 UC2: Manage Event:-



6.1.3 UC3: Search Event:-



6.1.4 UC4: Registration:-



6.2 Expanded Use Case

UC1: Login

Precondition: User must have already registered.	
Actor: User/ Admin	System: Sports Event Management System
1. **TUCBW** The actor clicks on "Login" button.	2. System display the login page.
3 The Actor enters credential and clicks on "Log in" button.	4(a) If credentials are correct system displays the homepage. 4(b)If credentials are wrong system displays error message
5. **(a) TUCEW** User/Admin sees the homepage. 5 **(b) TUCEW** User/Admin sees an error message.	
Post-Conditions: N/A	

UC2: Add Event

Precondition: Admin should be logged in.	
Actor: Admin	System: Sports Event Management System
1. **TUCBW** Admin will click on Add event button to add a new sport Event.	2. System will display a form to add event.
3. Admin fills the required details in the add event form.	4. System will check form details and successfully adds event.
5. **TUCEW** Admin receives success message.	
Post Condition: System is available for next use by admin.	

UC2: Delete Event

Precondition: Admin should be logged in.	
Actor: Admin	System: Sports Event Management System
1. **TUCBW** Admin will click on delete events button.	2. System will ask the Admin to enter the Event-Id to be deleted. 3.
4. Admin will enter the Event-ID of the event to be deleted and clicks on “Delete” button.	5. (a) If Event Id is correct System displays successful deletion message 4 (b) If Event-Id is wrong System displays an error message
6. **TUCEW** Admin receives a message.	
Post Condition: System is available for next use by admin.	

UC3: Search Events

Precondition: End User should be logged in.	
Actor: End User	System: Sports Event Management System
1. **TUCBW** End User search events by selecting one of the category.	2. System will display all sports events according to search criteria.
3. **TUCEW** End User will select one of the searched event.	
Post Condition: The system successfully displays the details of the selected event.	

UC4: Add Event Registration

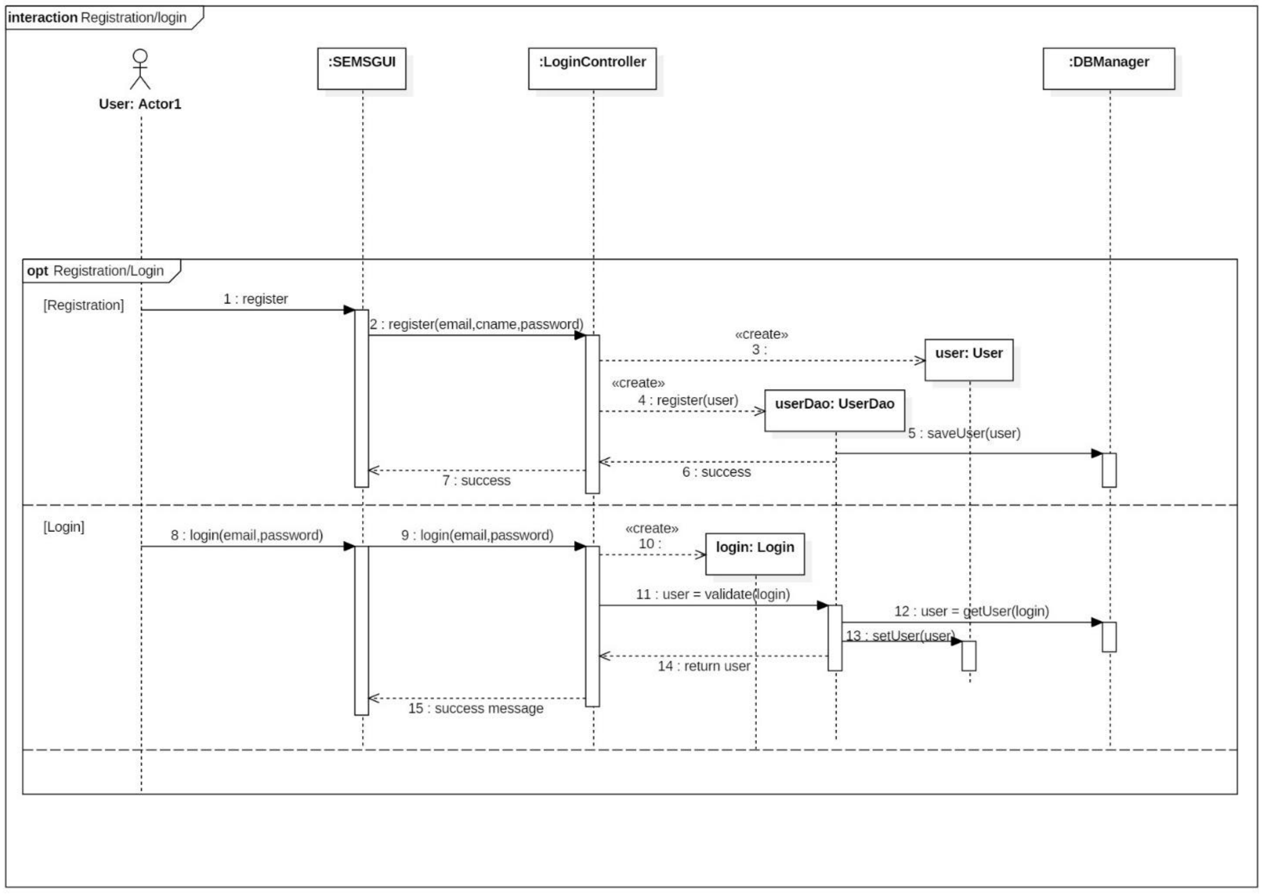
Precondition: The event should be available for the end User to register.	
Actor: End User	System: Sports Event Management System
1. **TUCBW** End User will click on the register event button.	2. System will display the event registration form.
3. User fills the required details and clicks the "Submit" button.	4. System sends successful registration message.
5. **TUCEW** End User receives success message.	
Post Condition: System is available for next use.	

6.3 Traceability Matrix

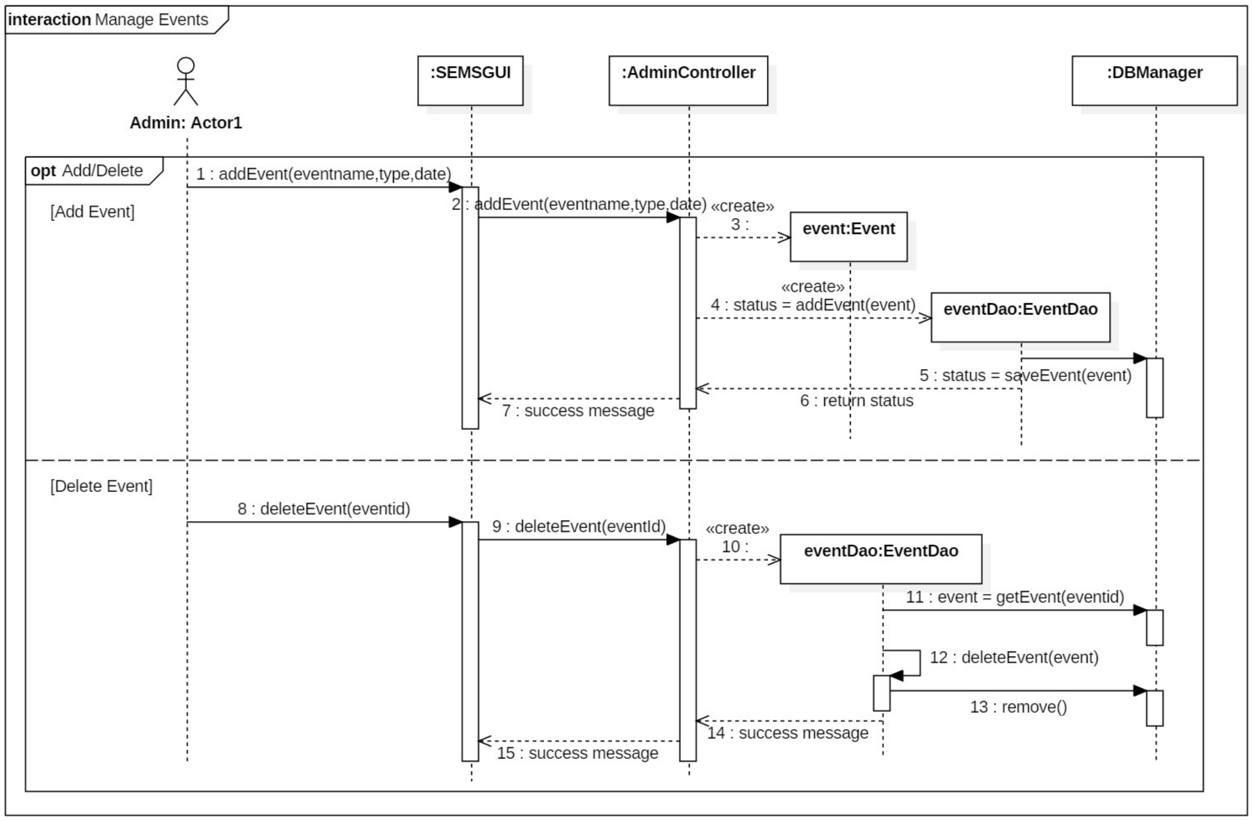
	Priority Weight	UC1	UC2	UC3	UC4
R1	6	X			
R2	5				X
R3	4		X		
R4	4			X	
R5	3			X	
R6	3		X		
R7	3				X
R8	2		X		
R9	2				X
Score		6	9	7	10

7. Sequence Diagram

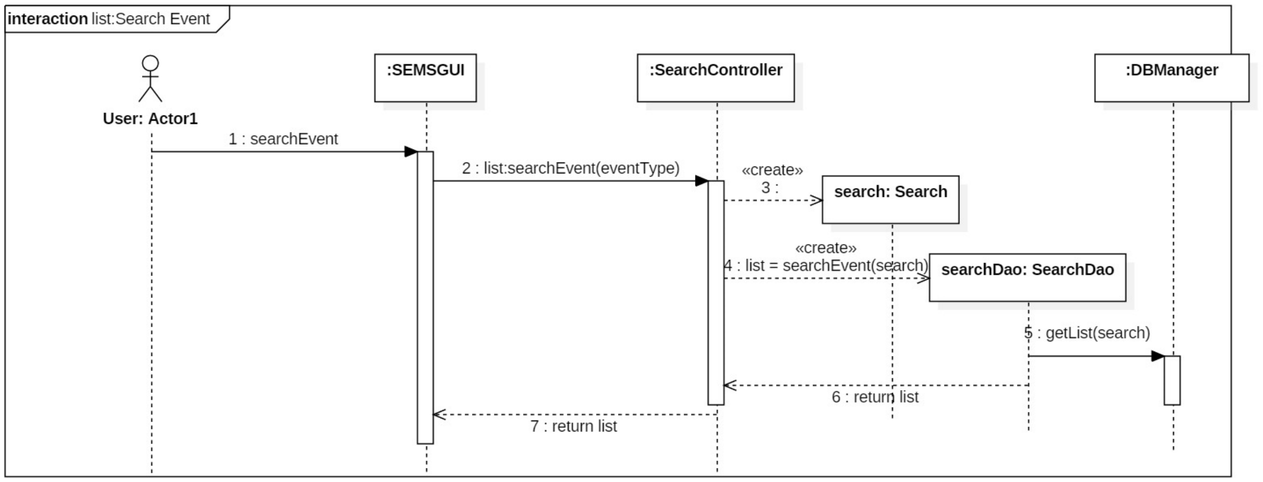
7.1 UC1 Sequence Diagram



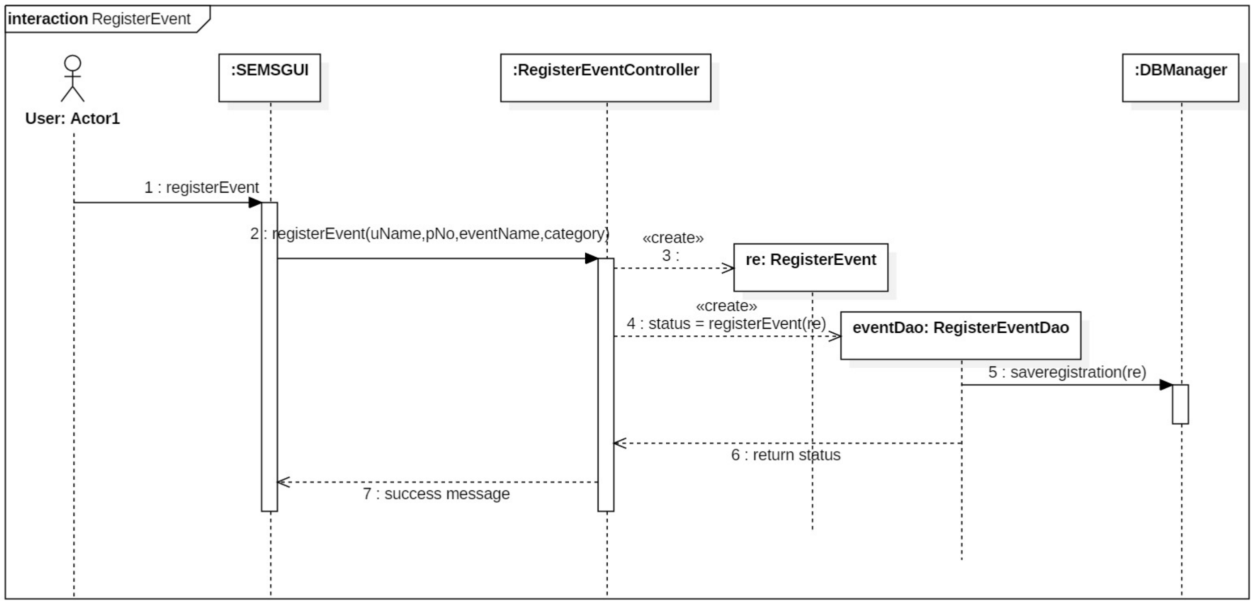
7.2 UC2 Sequence Diagram



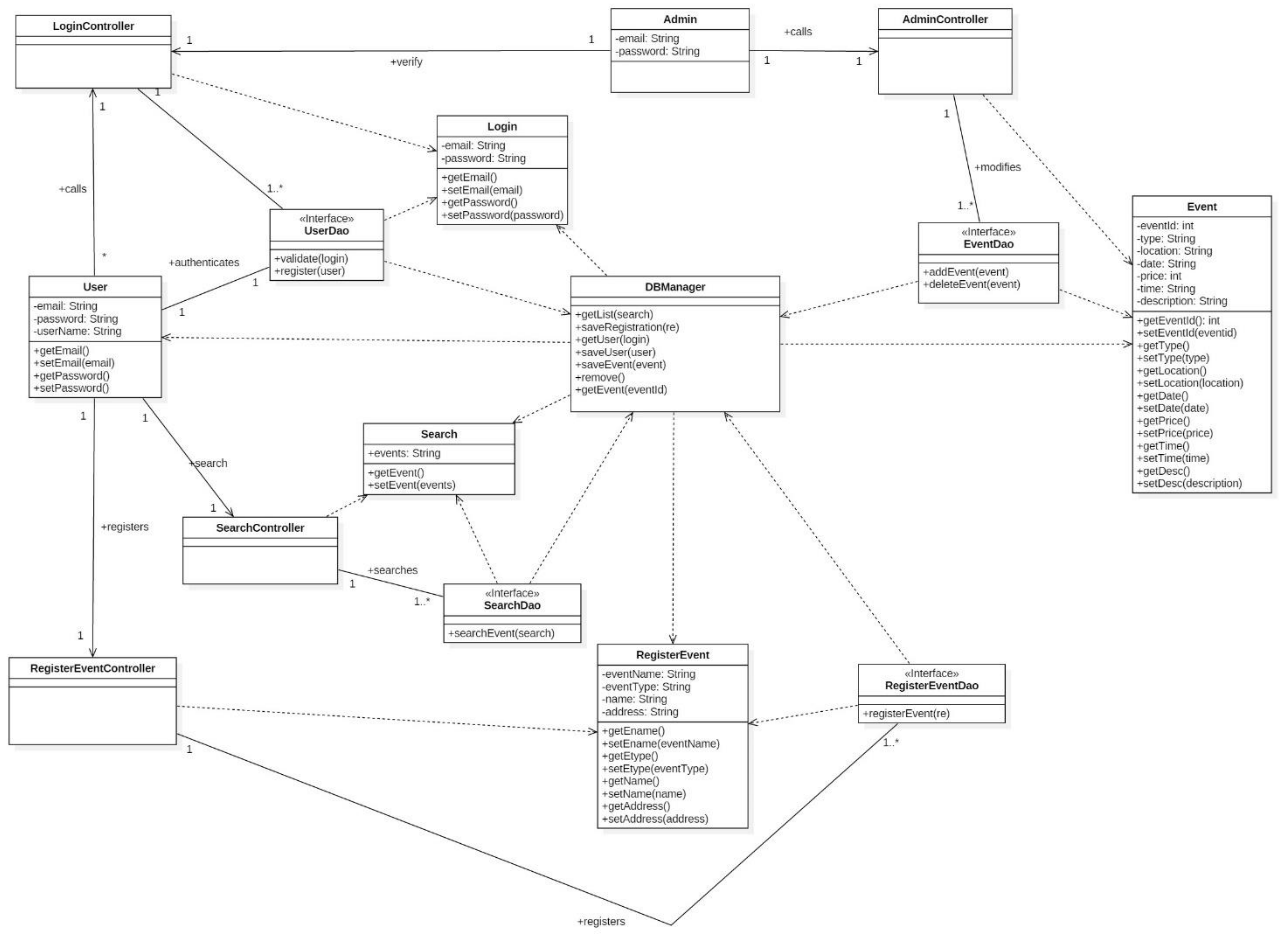
7.3 UC3 sequence Diagram



7.4 UC4 Sequence Diagram



8. Class Diagram



9. Testing

Testing is done to check whether the system is satisfying all the requirements mentioned in class diagram.

9.1 Testing plan

The following activities and test cases are used to check the quality of implementation with the requirements of the system:

No.	Test Case Description	Expected Output	Observed Output
UC1 Login:	1. User enters username and password to login into website. 2. User data is extracted from the database.	A success message should appear mentioning that the User was able to register successfully	A success message appeared and the user can see the events page and see all events available.
UC2 Manage Events: Admin	1. Admin can add/delete the current list of events.	Event details of the added/deleted event should be reverted.	Event details of the added/deleted event is reverted.
UC3 Search Event: User	1. User can search event according to categories. 2. User select event from the listed search events result.	1.Search results should be displayed according to selected category. 2. Event details should be displayed according to selected event.	1. Results are shown according to selected category. 2. Event details is displayed on screen.
UC4 Registration: End User	1. User can add registration.	1. Changes should be made in database accordingly.	1. Changes are done in database accordingly.

9.2 Test Cases:

UC1:– User Login

Case No	Expected	Observed
Test 0	Invalid Username	Invalid Username
Test 1	Invalid Password	Invalid Password
Test 2	Valid for user	Valid for user
Test 3	Valid for admin	Valid for admin

UC2:- Manage Events

Case No	Expected	Observed
Test 0	Logs out if Inactivity for more than 30 seconds.	Logs out
Test 1	Add event failed for empty fields.	Add event failed
Test 2	Add event failed for invalid event type.	Add event failed
Test 3	Add event failed for invalid date.	Add event failed
Test 4	Add event failed for invalid price.	Add event failed
Test 5	Add event failed for invalid time.	Add event failed
Test 6	Add event success for type single	Add event success
Test 7	Add event success for type multilayer.	Add event success
Test 8	Delete event Success	Delete event Success

UC3:- Search Events

Case No	Expected	Observed
Test 0	Search success for single player.	Search success
Test 1	Search success for multiplayer player.	Search success

UC4 :- User Registration

Case No	Expected	Observed
Test 0	Invalid login for non existing user	Invalid login
Test 1	Invalid registration for empty data fields	Invalid registration
Test 2	Invalid registration for password mismatch	Invalid registration
Test 3	Valid registration for non existing user	Valid registration
Test 4	Valid login for existing user.	Valid login

10. Project Code

10.1 Sample register event code

```
package domain.login;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

import db.DbManager;



public class RegisterEventDaoImpl implements RegisterEventDao {

	static Connection conn;
	static PreparedStatement ps;
	DbManager db = new DbManager();

	@Override
	public int addEvent(RegisterEvent registerevent) {
		int status = 0;
		try{
			conn = db.getConnection();
			ps =conn.prepareStatement("insert into reservation values(?,?,?,?,?)");
			ps.setString(1, registerevent.getUsername());
			ps.setString(2, registerevent.getPhoneNumber());
			ps.setString(3, registerevent.getEventName());
			ps.setString(4, registerevent.getCategory());
			ps.setString(5, registerevent.getAddress());

			status = ps.executeUpdate();
			conn.close();
		}
		catch(Exception e){
			System.out.println(e);
		}
		return status;
	}
}
```

10.2 SQL Code

```
create database project;

show databases;

use project;

drop table events;
create table events(

EventId INT NOT NULL,

Topic VARCHAR(30) NOT NULL,

EventType VARCHAR(30) NOT NULL,

EventDate DATE NOT NULL,

Location VARCHAR(200) NOT NULL,

Price FLOAT NOT NULL,

EventTime VARCHAR(10) NOT NULL,

Description  VARCHAR(400) ,

PRIMARY KEY (EventId)

);

insert into events values(1, "cricket", "multiplayer","2019-10-05", "dallas", 2.0, 2000,"Cricket is
a bat-and-ball game played between two teams of eleven players on a field");
insert into events values(2, "soccer", "multiplayer","2019-11-05", "boston", 3.0, 1900,"soccer, is
a team sport played with a spherical ball between two teams of eleven players.");
insert into events values(3, "tennis", "single","2020-10-05", "chicago", 4.0, 1000,"Each player
uses a tennis racket that is strung with cord to strike a hollow rubber ball covered with felt
over or around a net and into the opponent's court.");
insert into events values(4, "table tennis", "single","2019-07-05", "seattle", 1.0, 1500,"The game
takes place on a hard table divided by a net");

select * from events;
```

```
drop table User;
create table User(

Email VARCHAR(100) NOT NULL,

UserPassword VARCHAR(1000),

PRIMARY KEY(Email)

);

insert into User values("shrey@gmail.com","shrey");
insert into User values("aarya@gmail.com","aarya");
insert into User values("vinyas@gmail.com","vinyas");

select * from User;

drop table reservation;
create table reservation(
username varchar(20),
 phonenumber varchar(20),
 eventname varchar(20),
 category varchar(20),
 address varchar(50)
 );
```

10.3 RegisterEvent.jsp

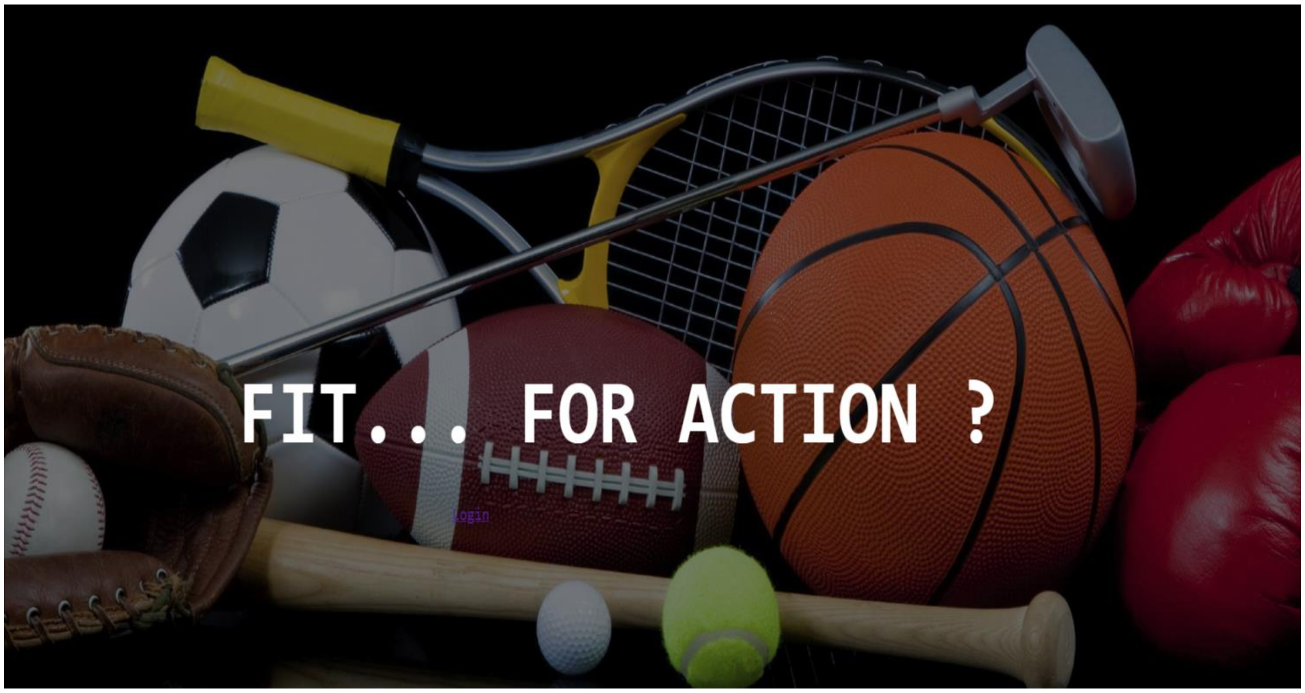
```
<%@page import="db.DbManager"%>
<%@page import="java.sql.Connection"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<!-- Start Changed -->
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<!-- The above 3 meta tags *must* come first in the head; any other head content must come
*after* these tags -->
```

```
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">

    <!-- bootstrap general -->
            <!-- <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet" type="text/css" /> -->
            <script type="text/javascript" src="bootstrap/js/bootstrap.min.js"></script>
            <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<meta name="description" content="">
<meta name="author" content="">
<link rel="icon" href="../../favicon.ico">
<!-- End Changed -->
<script type="text/javascript" src="script.js"></script>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Login Page</title>
<!-- Bootstrap core CSS -->
<link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
<link href="bootstrap/css/signin.css" rel="stylesheet">
<!-- Custom styles for this template -->
<link href="bootstrap/css/shop-homepage.css" rel="stylesheet">
<%
                    session.invalidate();
                    response.setHeader("Cache-Control", "no-cache, no-store, must-revalidate"); //
HTTP 1.1.
                    response.setHeader("Pragma", "no-cache"); // HTTP 1.0.
                    response.setHeader("Expires", "0");
                    response.setHeader("Vary", "*");
%>
<!-- Start Change -->
    <!-- Bootstrap core CSS -->
    <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">

    <!-- IE10 viewport hack for Surface/desktop Windows 8 bug -->
    <link href="bootstrap/css/ie10-viewport-bug-workaround.css" rel="stylesheet">

    <!-- Custom styles for this template -->
    <link href="bootstrap/css/signin.css" rel="stylesheet">

    <!-- Just for debugging purposes. Don't actually copy these 2 lines! -->
    <!--[if lt IE 9]><script src="../../assets/js/ie8-responsive-file-warning.js"></script><![endif]-->
    <script src="bootstrap/js/ie-emulation-modes-warning.js"></script>
    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
   <!--  End Change -->
</head>
<body>
    <!-- Navigation -->




    </br>

            <%out.print(" "); %>

            <%
   response.addHeader("Cache-Control", "no-cache,no-store,private,must-revalidate,max-
stale=0,post-check=0,pre-check=0");
   response.addHeader("Pragma", "no-cache");
   response.addDateHeader ("Expires", 0);
            %>
            <%--
            <%! int number1, number2; %>

            <%
                        DbManager db = new DbManager();
                        Connection conn = (Connection) db.getConnection();
                        if(conn == null)
                                    out.print("failed");
                        else
                                    out.print("");
            %>--%>
              <center><h3>Welcome, Please Fill The Form</h3></center>
            <br/>




    <!-- IE10 viewport hack for Surface/desktop Windows 8 bug -->
    <script src="bootstrap/js/ie10-viewport-bug-workaround.js"></script>
```

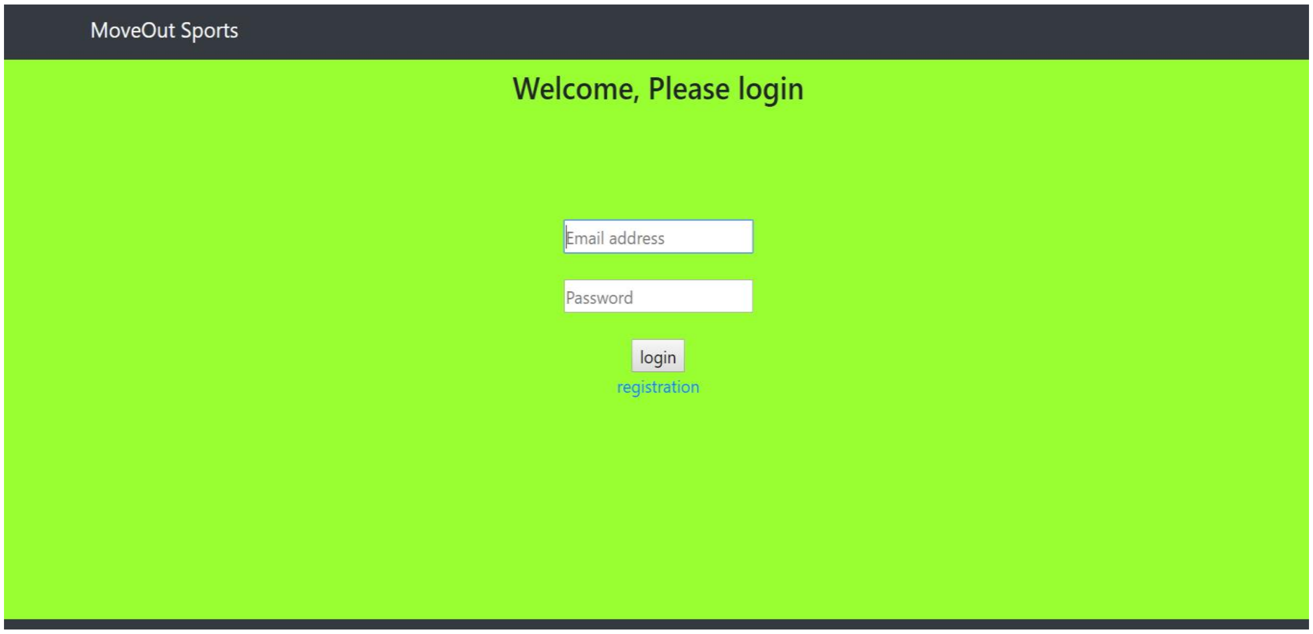
```
<form name="regform" action="RegisterEventController" method="post" onsubmit="">
	<br>${message}<br>

	<label for="inputName" class="sr-only">Name</label>
	<input type="text" name="username" id="username" placeholder="user name" required
autofocus><br>
	</br>
	<label for="inputPhonenumber" class="sr-only">Phone Number</label>
	<input type="text" name="phonenumber" id="phonenumber" placeholder="phone number"
required><br>
	</br>
	<label for="inputEventname" class="sr-only">Event Name</label>
	<input type="text" name="eventname" id="eventname" placeholder="event name"
required autofocus><br>
	</br>
	<label for="inputCategory" class="sr-only">Event Category</label>
	<input type="text" name="category" id="category" placeholder="event category"
required><br>
	</br>
	<label for="inputAddress" class="sr-only">Address</label>
	<input type="text" name="address" id="address" placeholder="address" required
autofocus><br>
	</br>

	<input type="submit" name="submit" value="Reserve Event" >
	<input type="reset" name="reset">

	</form>

	<footer class="py-2 bg-dark fixed-bottom">
     <div class="container">
       <p class="m-0 text-center text-white"></p>

     </div>
   </footer>

   <!-- Bootstrap core JavaScript -->
   <script src="bootstrap/js/jquery.min.js"></script>
   <script src="bootstrap/js/bootstrap.bundle.min.js"></script>

</body>
</html>
```

11. Screenshots

11.1 Home Page



11.2 LOGIN FORM


MoveOut Sports

Welcome, Please login

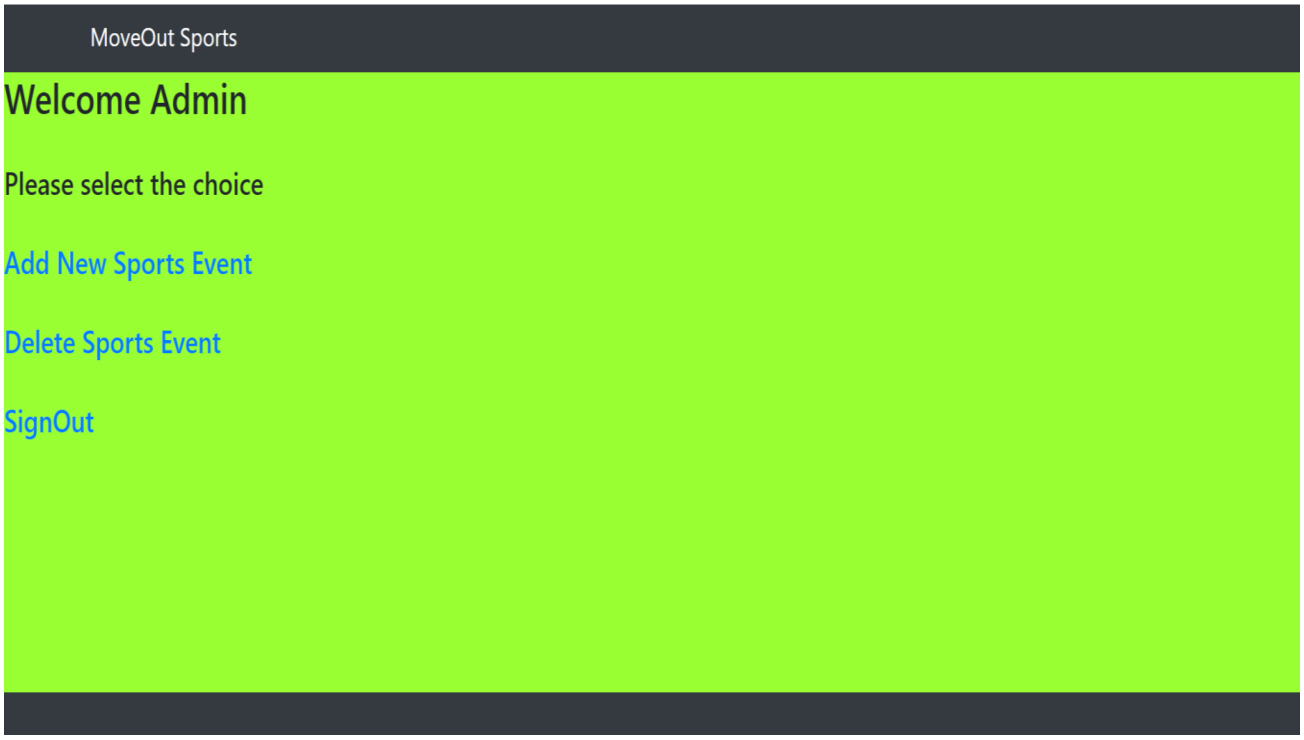
Email address

Password

login

registration

11.3 Admin Home Page


MoveOut Sports

Welcome Admin

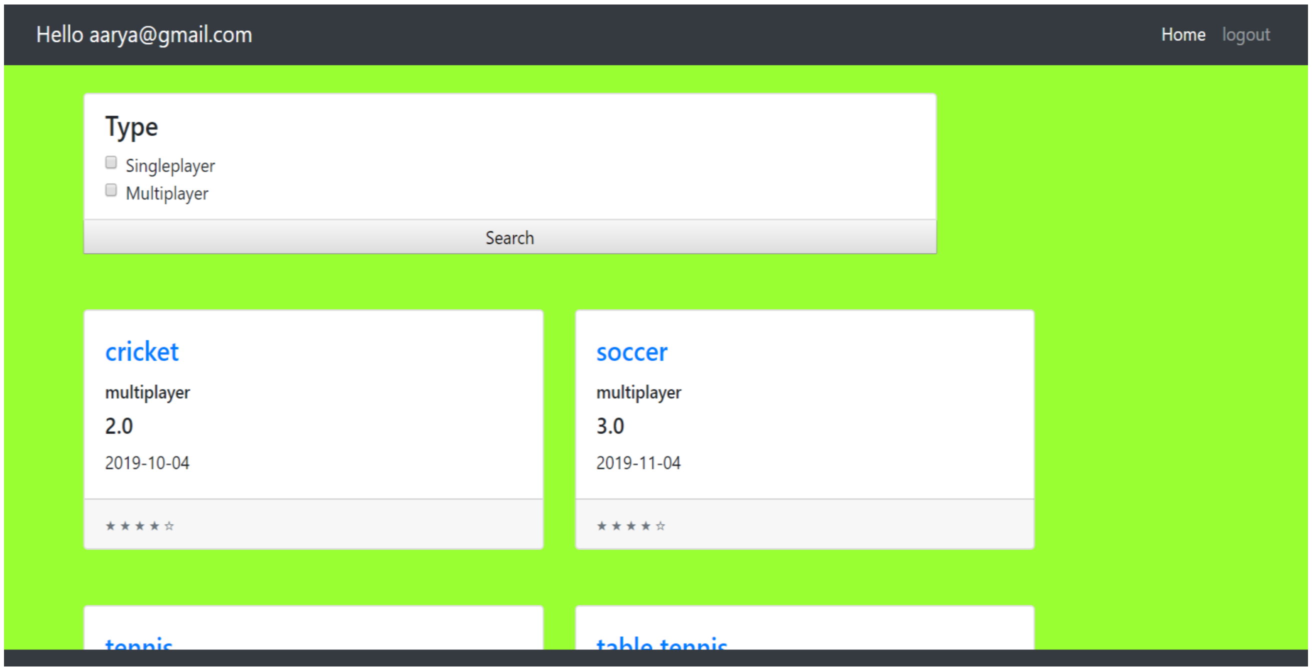
Please select the choice

Add New Sports Event

Delete Sports Event

SignOut

11.4 User SEARCH PAGE



Hello aarya@gmail.com

Home logout

Type

- [ ] Singleplayer
- [ ] Multiplayer

Search

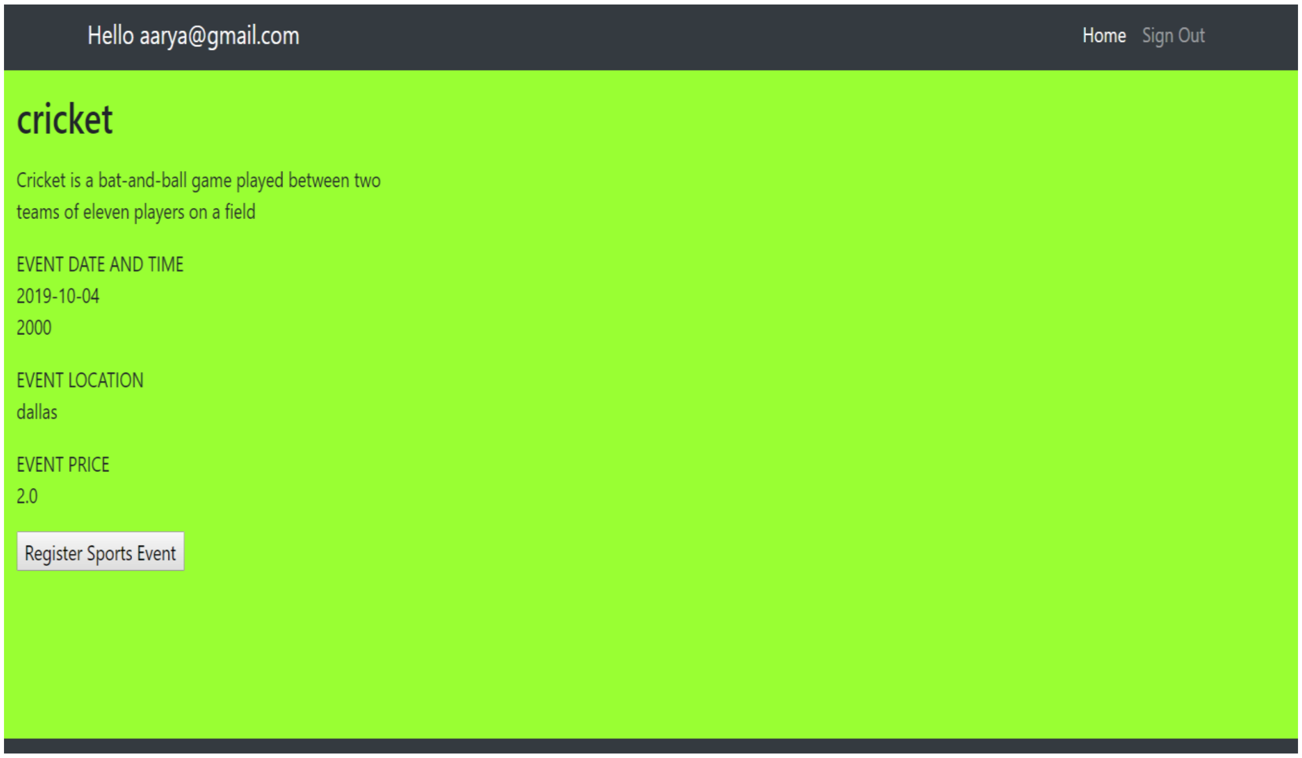
cricket
multiplayer
2.0
2019-10-04
★★★★☆

soccer
multiplayer
3.0
2019-11-04
★★★★☆

tennis

table tennis

11.5 Event Description



Hello aarya@gmail.com

Home Sign Out

cricket

Cricket is a bat-and-ball game played between two teams of eleven players on a field

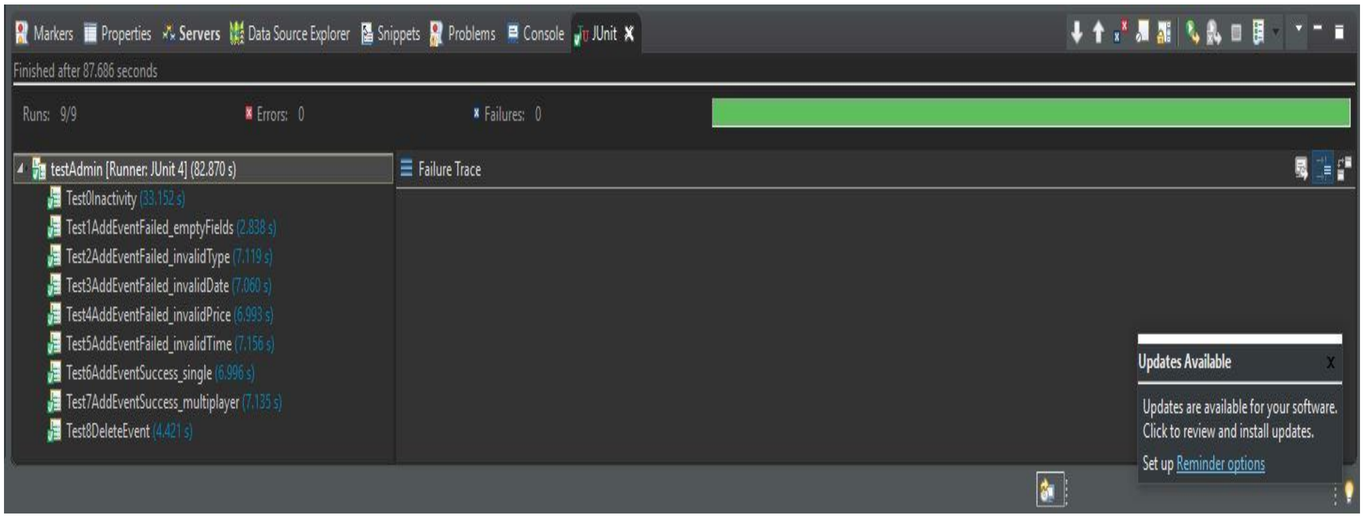
EVENT DATE AND TIME
2019-10-04
2000

EVENT LOCATION
dallas

EVENT PRICE
2.0

Register Sports Event

11.6 Testing Outcome Samples

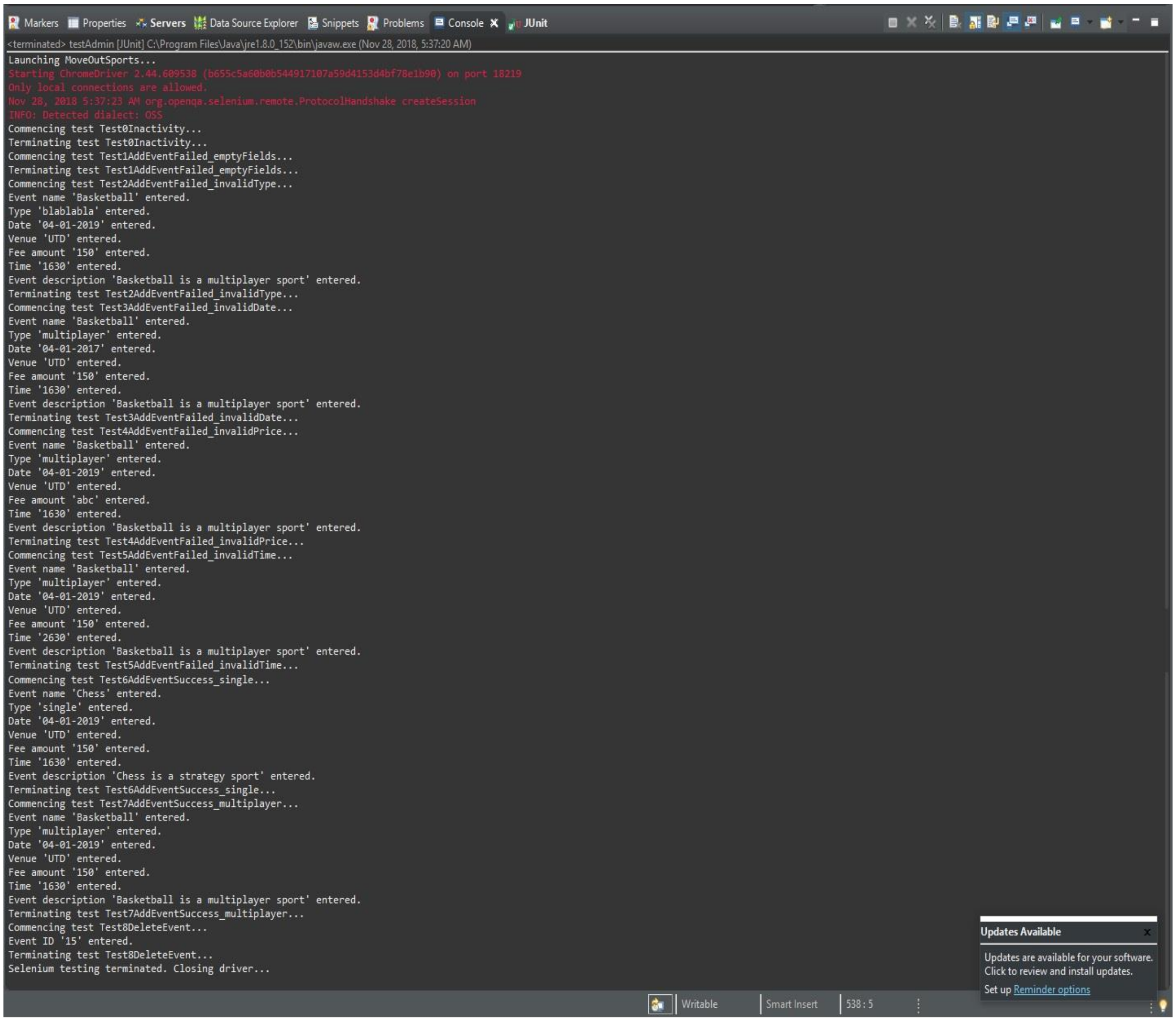


```
Finished after 87.686 seconds

Runs: 9/9        Errors: 0        Failures: 0

testAdmin [Runner: JUnit 4] (82.870 s)
    Test0Inactivity (33.152 s)
    Test1AddEventFailed_emptyFields (2.838 s)
    Test2AddEventFailed_invalidType (7.119 s)
    Test3AddEventFailed_invalidDate (7.060 s)
    Test4AddEventFailed_invalidPrice (6.993 s)
    Test5AddEventFailed_invalidTime (7.156 s)
    Test6AddEventSuccess_single (6.996 s)
    Test7AddEventSuccess_multiplayer (7.135 s)
    Test8DeleteEvent (4.421 s)
```

Updates Available

Updates are available for your software. Click to review and install updates.

Set up Reminder options

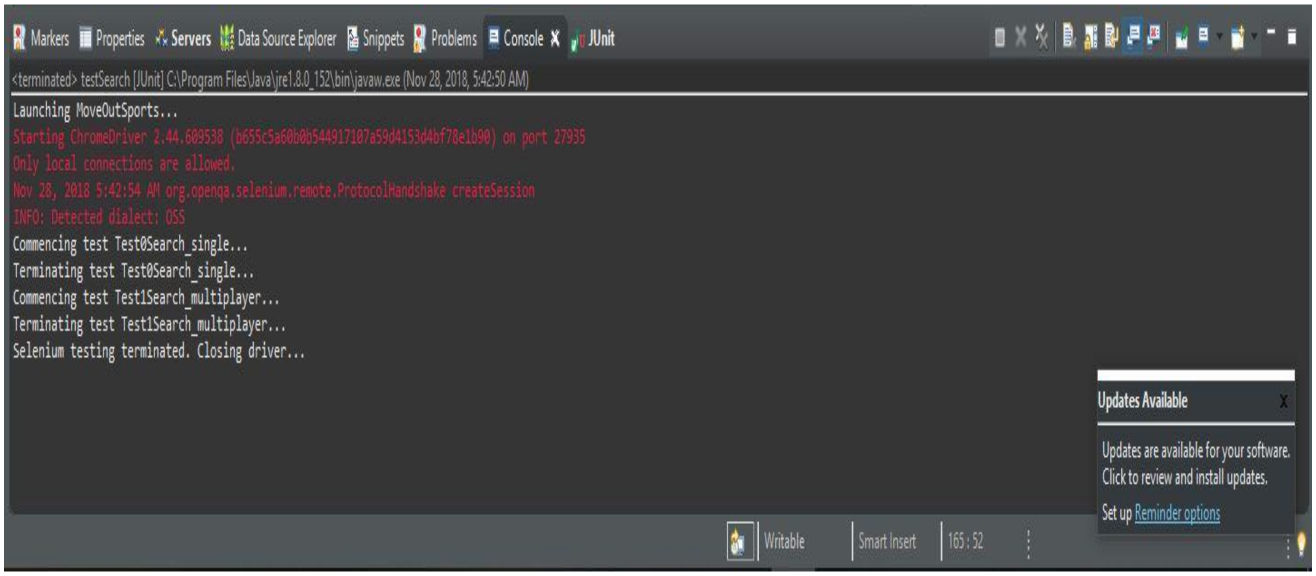


```
<terminated> testAdmin [JUnit] C:\Program Files\Java\jre1.8.0_152\bin\javaw.exe (Nov 28, 2018, 5:37:20 AM)
Launching MoveOutSports...
Starting ChromeDriver 2.44.609538 (b655c5a60b0b544917107a59d4153d4bf78e1b90) on port 18219
Only local connections are allowed.
Nov 28, 2018 5:37:23 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: OSS
Commencing test Test0Inactivity...
Terminating test Test0Inactivity...
Commencing test Test1AddEventFailed_emptyFields...
Terminating test Test1AddEventFailed_emptyFields...
Commencing test Test2AddEventFailed_invalidType...
Event name 'Basketball' entered.
Type 'blablabla' entered.
Date '04-01-2019' entered.
Venue 'UTD' entered.
Fee amount '150' entered.
Time '1630' entered.
Event description 'Basketball is a multiplayer sport' entered.
Terminating test Test2AddEventFailed_invalidType...
Commencing test Test3AddEventFailed_invalidDate...
Event name 'Basketball' entered.
Type 'multiplayer' entered.
Date '04-01-2017' entered.
Venue 'UTD' entered.
Fee amount '150' entered.
Time '1630' entered.
Event description 'Basketball is a multiplayer sport' entered.
Terminating test Test3AddEventFailed_invalidDate...
Commencing test Test4AddEventFailed_invalidPrice...
Event name 'Basketball' entered.
Type 'multiplayer' entered.
Date '04-01-2019' entered.
Venue 'UTD' entered.
Fee amount 'abc' entered.
Time '1630' entered.
Event description 'Basketball is a multiplayer sport' entered.
Terminating test Test4AddEventFailed_invalidPrice...
Commencing test Test5AddEventFailed_invalidTime...
Event name 'Basketball' entered.
Type 'multiplayer' entered.
Date '04-01-2019' entered.
Venue 'UTD' entered.
Fee amount '150' entered.
Time '2630' entered.
Event description 'Basketball is a multiplayer sport' entered.
Terminating test Test5AddEventFailed_invalidTime...
Commencing test Test6AddEventSuccess_single...
Event name 'Chess' entered.
Type 'single' entered.
Date '04-01-2019' entered.
Venue 'UTD' entered.
Fee amount '150' entered.
Time '1630' entered.
Event description 'Chess is a strategy sport' entered.
Terminating test Test6AddEventSuccess_single...
Commencing test Test7AddEventSuccess_multiplayer...
Event name 'Basketball' entered.
Type 'multiplayer' entered.
Date '04-01-2019' entered.
Venue 'UTD' entered.
Fee amount '150' entered.
Time '1630' entered.
Event description 'Basketball is a multiplayer sport' entered.
Terminating test Test7AddEventSuccess_multiplayer...
Commencing test Test8DeleteEvent...
Event ID '15' entered.
Terminating test Test8DeleteEvent...
Selenium testing terminated. Closing driver...
```

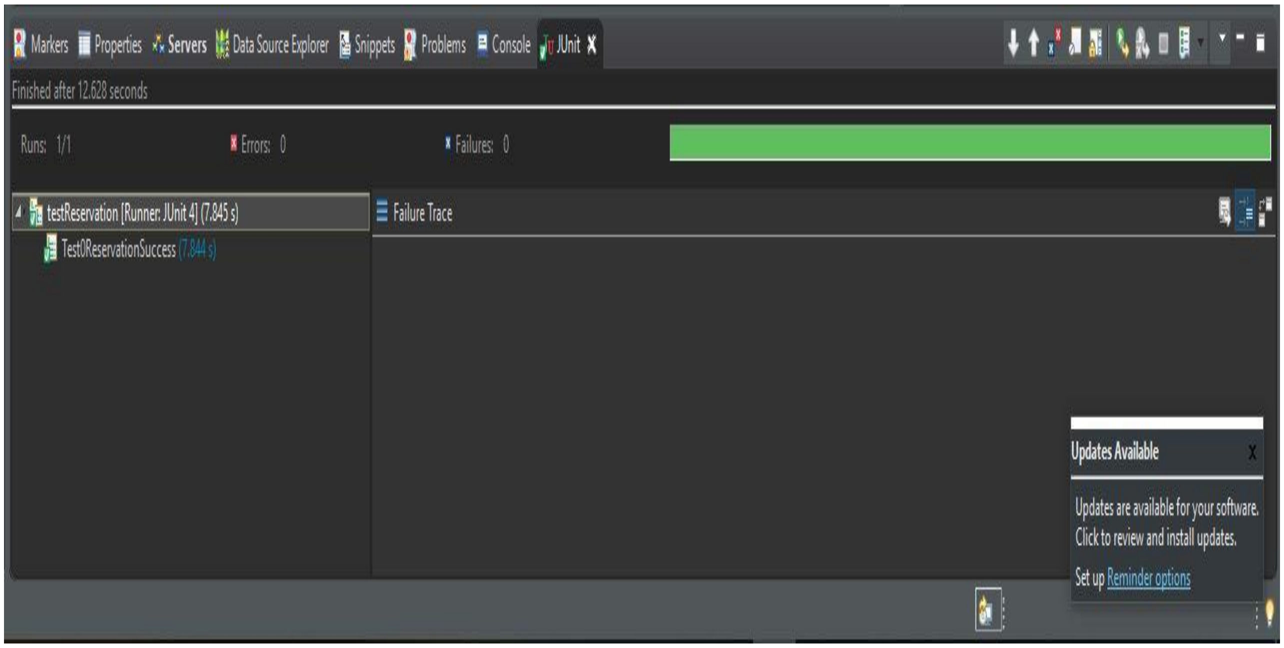
Updates Available

Updates are available for your software. Click to review and install updates.

Set up Reminder options



```
<terminated> testSearch [JUnit] C:\Program Files\Java\jre1.8.0_152\bin\javaw.exe (Nov 28, 2018, 5:42:50 AM)
Launching MoveOutSports...
Starting ChromeDriver 2.44.609538 (b655c5a60b0b544917107a59d4153d4bf78e1b90) on port 27935
Only local connections are allowed.
Nov 28, 2018 5:42:54 AM org.openqa.selenium.remote.ProtocolHandshake createSession
INFO: Detected dialect: OSS
Commencing test Test0Search_single...
Terminating test Test0Search_single...
Commencing test Test1Search_multiplayer...
Terminating test Test1Search_multiplayer...
Selenium testing terminated. Closing driver...
```



```
Finished after 12.628 seconds
Runs: 1/1        Errors: 0        Failures: 0
testReservation [Runner: JUnit 4] (7.845 s)
    Test0ReservationSuccess (7.844 s)
```