EVENT MANAGEMENT SYSTEM

Alfina I(21Z207)

Santhoshi R(21Z251)

19Z512 – Software Package Development Laboratory

report submitted in partial fulfilment of the requirement for the award of degree of

BACHELOR OF ENGINEERING

Branch: COMPUTER SCIENCE AND ENGINEERING

Of Anna University

October 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)

COIMBATORE – 641 004

PSG COLLEGE OF TECHNOLOGY

TABLE OF CONTENTS:

Chapter	Title	Page number
1	Problem Statement	2
2	Software requirements specification	3
3	Use case model	5
4	Domain class diagram	6
5	Graphical User Interface	7
6	Entity Relationship diagram	10
7	Activity diagram	11
8	State Diagram	12
9	Sequence diagram	13
10	Design class diagram	14
11	Testing	15
12	Conclusion	17
13	Appendix	18

CHAPTER 1

PROBLEM STATEMENT

1.1 INTRODUCTION

This project is designed to automate the process of booking a venue for the event, managing the whole process online. It will help in managing events on both small and large scales like conferences, weddings, festivals, formal parties, gatherings, get-togethers, etc.

Event management is a process of organizing a professional and focused event, for a particular target audience. It involves visualizing concepts, planning, budgeting, organizing and executing events such as fashion shows, musical concerts, corporate seminars, exhibitions, wedding celebrations, theme parties, product launching etc.

1.2 PROBLEM DEFINITION

These days the world has become a digital world where everything is available in a single click or touch.

The definition of our problem lies in manual system and a fully automated system of Event Management System.

1.3 MANUAL SYSTEM

In Manual system is more prone to errors and sometimes it encounters various problems which are unstructured. Because things are managing by the human on the paper there might high chances to get mistakes as well as its time consuming and high money consuming.

1.4 TECHNICAL SYSTEM

The technical system comes with the advent of latest technology and user can access the application over the browser and manage the things in just few clicks.

CHAPTER 2

SOFTWARE REQUIREMENTS SPECIFICATION

2.1 ABSTRACT

This is the requirements document for the project 'Event management system' that will be used throughout the project. This project is designed to automate the process of booking a venue for the event, managing the whole process online. It will help in managing events on both small and large scales like conferences, weddings, festivals, formal parties, gatherings, get-togethers, etc.

2.2 INTRODUCTION

2.2.1 PURPOSE

The purpose of this document is to describe the external requirements for an event scheduling system. It also describes the interfaces for the system.

2.2.2 SCOPE

This document is the only one that describes the requirements of the system. It is meant for use by the end users and the admin and will be the basis for validating the final delivered system. Any changes made to the requirements in the future will have to go through a formal change approval process. The developer is responsible for asking for clarifications, where necessary, and will not make any alterations without the permission of the client.

2.2.3 DEFINITIONS, ACRONYMS, ABBREVIATIONS

Not applicable

2.2.4 REFERENCES

Not applicable

2.2.5 DEVELOPER'S RESPONSIBILITIES

The developer is responsible for (a) developing the system, (b) installing the software on the client's hardware, (c) conducting any user training that might be needed for using the system and (d) maintaining the system for a period of one year after installation.

2.3 GENERAL DESCRIPTION

2.3.1 PRODUT FUNCTIONS OVERVIEW

In a city there are a set of venues in which various kinds of events take place. Each year, the city witnesses various events and the citizens celebrate a variety of events like wedding, anniversary, social meeting and get together. For each event, the event manager gives the list of available venues for the given date and time depending on the nature of the event.

The type of the event is mentioned by the end user. No two events should be scheduled in the same venue at the same time and date.

2.3.2 USER CHARACTERISTICS

The main users of this system will be the people who wants to conduct the event (i. e the end user) and the admin who are somewhat literate with computers and can use programs such as editors and text processors.

2.3.3 GENERAL CONSTRAINTS

The system should run on Windows 10 or more.

2.4 SPECIFIC REQUIREMENTS

We have a wide range of options of programming languages. From these options we can choose appropriate platform tools, technologies and languages for development of the airline reservation project.

Some of these are as following

Programming Languages: Java.

Relational Database: MYSQL.

SOFTWARE REQUIREMENTS:

Operating system	:	Windows Family
Front End	:	CSS, JSP, HTML
Back End	:	Servlet, JDBC.
Server	:	Tomcat Server

HARDWARE SPECIFICATIONS

Processor	:	(i3) Intel Pentium or more
Ram	:	4 GB
Hard disk	:	16 GB hard disk recommended

2.5 FUNCTIONAL REQUIREMENTS

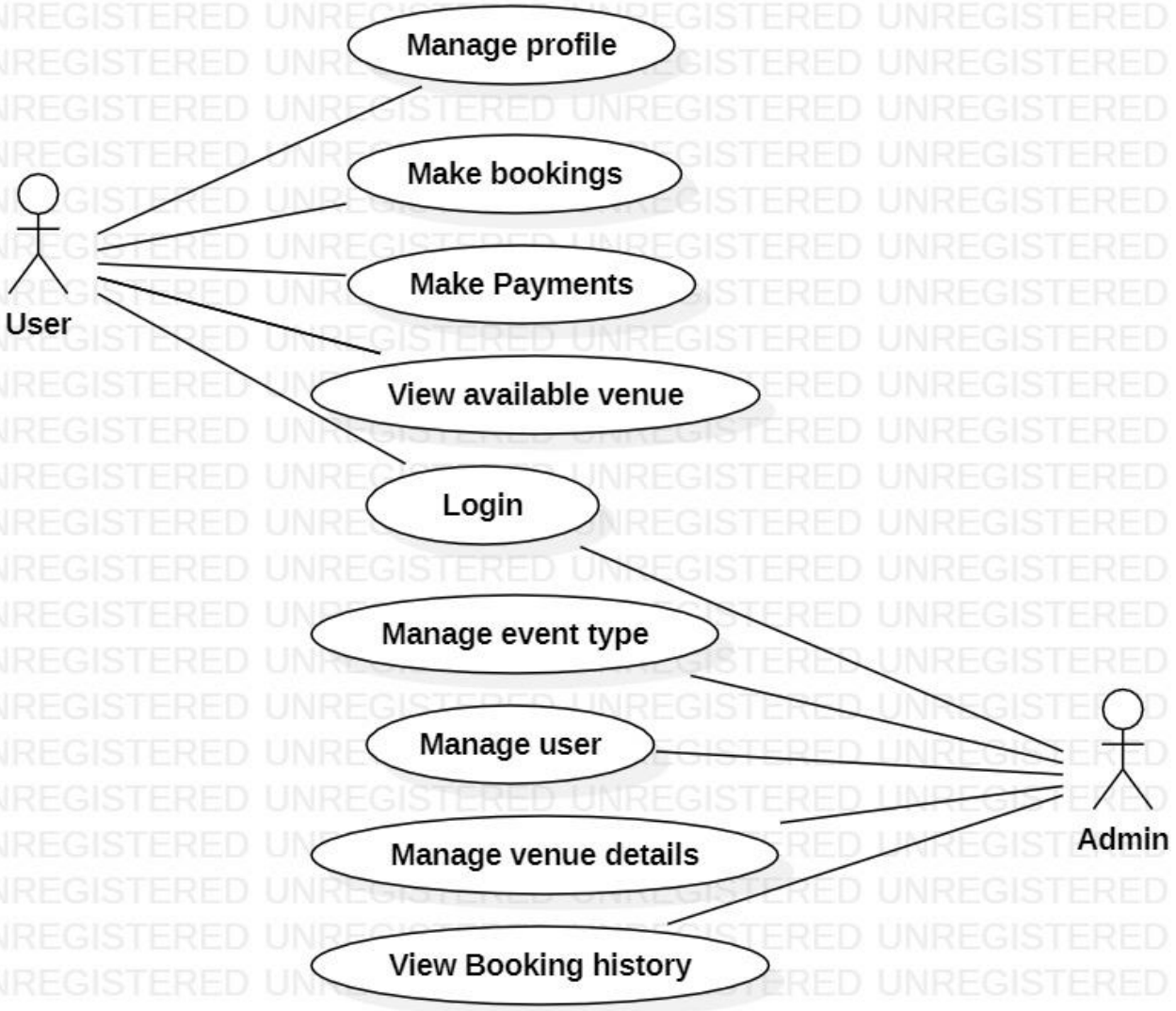
- To develop a system to Event Management, this will perform all the Event Management operation on an online platform.
- To develop a system that has good management of data along with integrity and minimizing redundancy.
- To develop a system that will be user friendly in all possible ways.
- To provide better customer support for User.

2.6 PERFORMANCE CONSTRAINTS

The output should be displayed in less than 1 minute in the worst case.

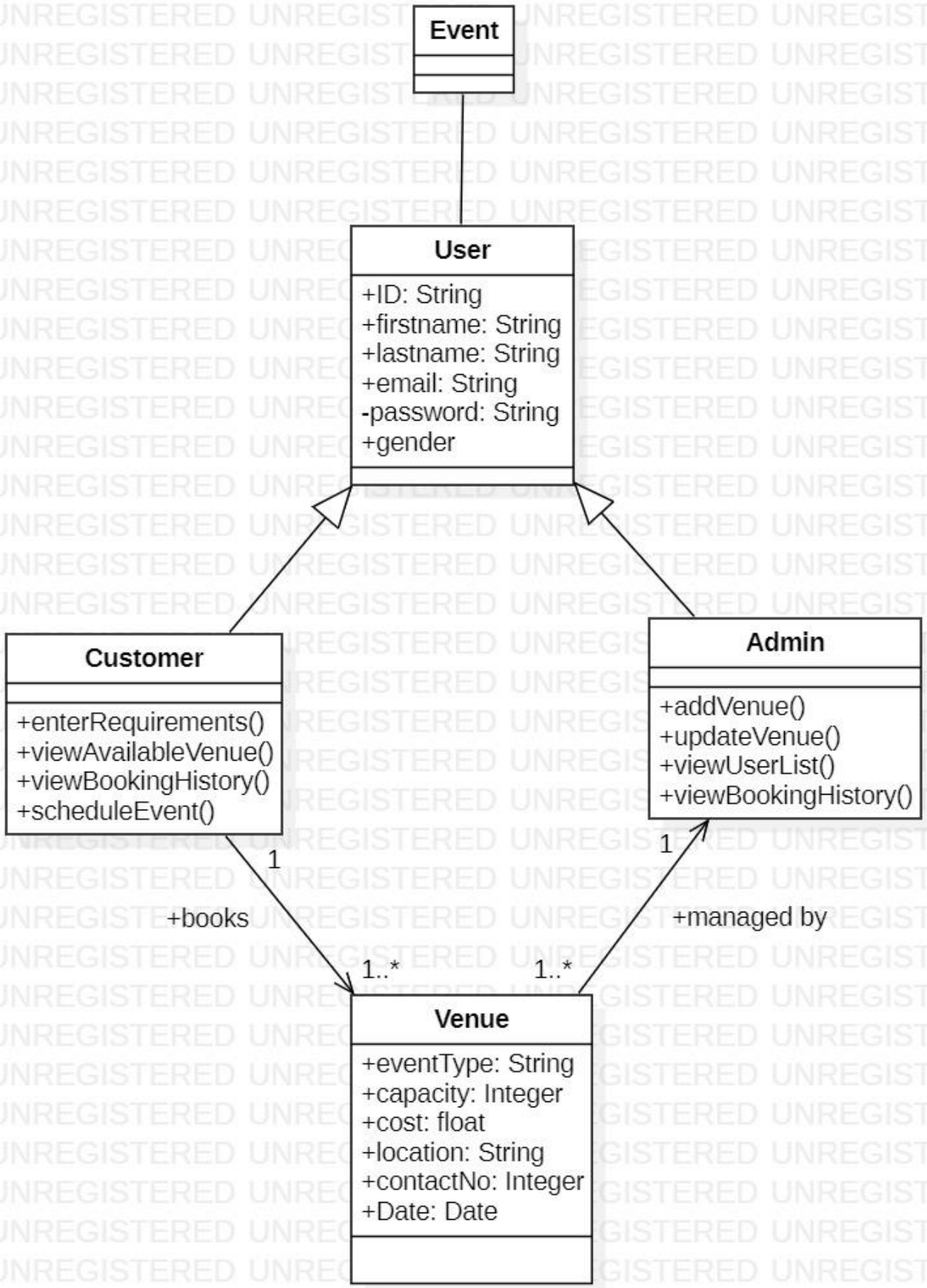
CHAPTER 3

USE CASE MODEL



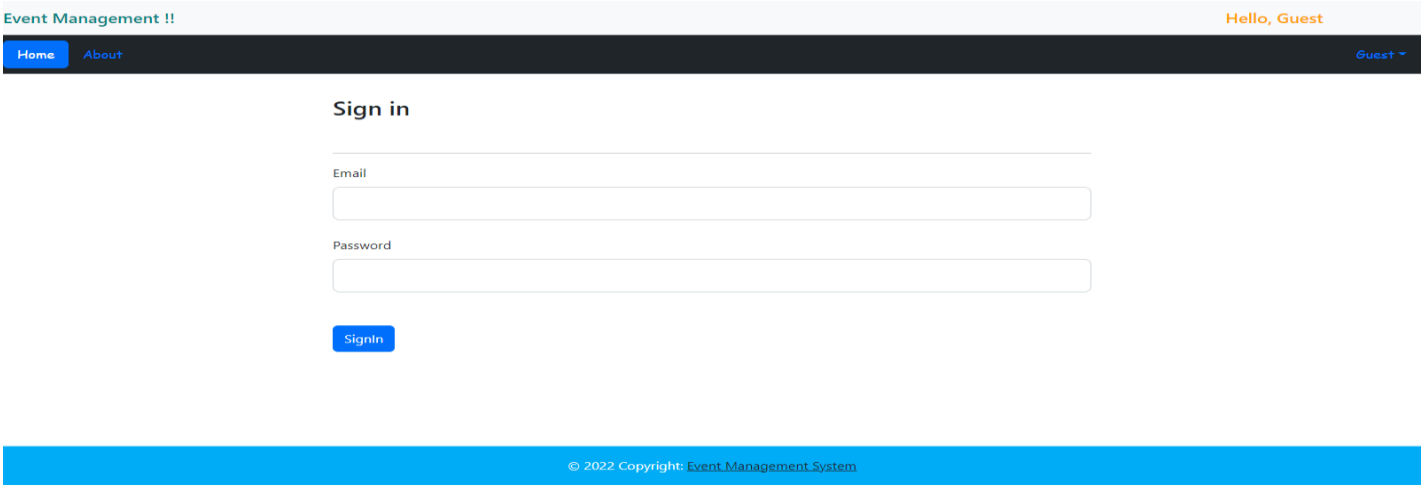
CHAPTER 4

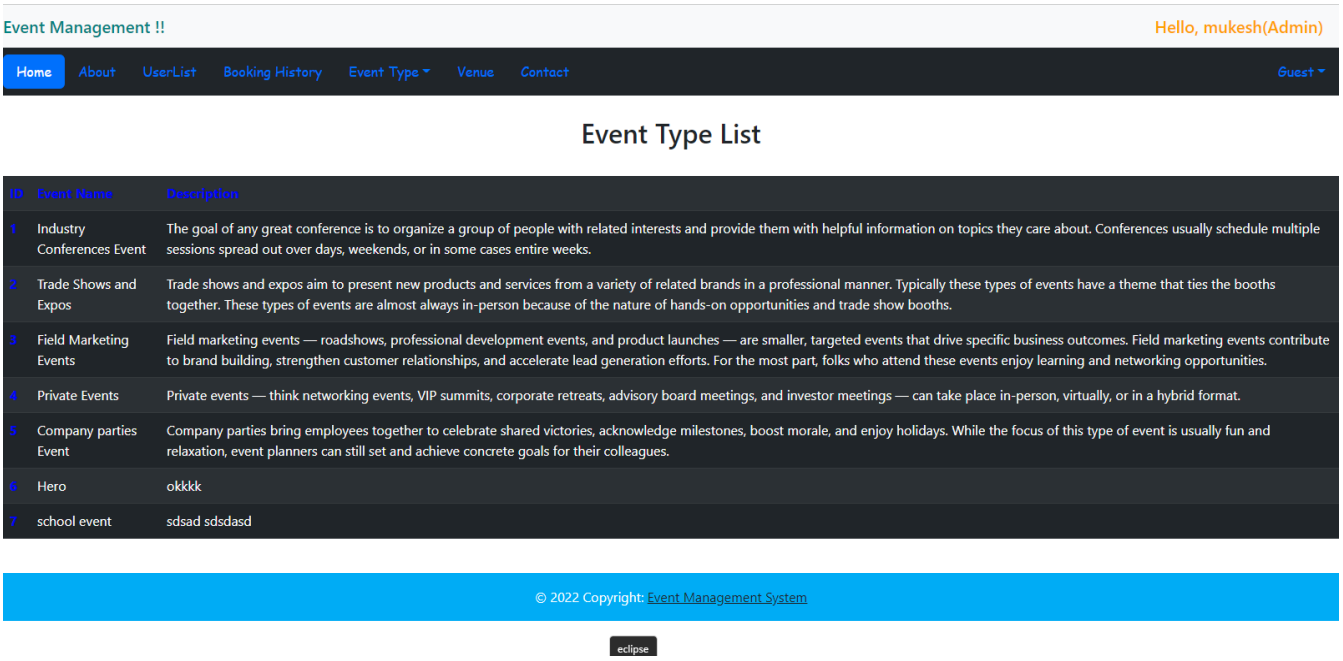
DOMAIN CLASS DIAGRAM



CHAPTER 5

GRAPHICAL USER INTERFACE





Event Management !! Hello, mukesh(Admin)

Home About UserList Booking History Event Type Venue Contact Guest

VENUE

Event Type:

Capacity:

Enter here...

Cost:

Enter here...

Location

Enter here...

Contact:

Enter here.....

Date:

date Enter Here

Event Photo:

Choose File No file chosen

Submit

Event Management !! Hello, mukesh(Admin)

Home About UserList Booking History Event Type Venue Contact Guest

History

ID	PersonName	CardNumber	Expire	CVV	
1	ishwar	878726475745	01/2030	980	Payment Success
2	govind	352525266211	09/2031	465	Payment Success
3	mukesh	535523525	01/2043	463	Payment Success
4	manish	r345435	43252	e4243	Payment Success
5	Harper Wyatt	675	Illum amet maiores	Pa$$w0rd!	Payment Success

© 2022 Copyright: Event Management System

Event Management !! Hello, manish(User)

Home About Book Booking Histroy Contact Guest

Venue List

ID	Image	Location	Capacity	Cost	Contact	Date	EventType	Action
1		Mumbai	1000	50000	9875536564	2022-07-21	Industry Conferences Event	Book& Pay
2		Mandsaur	500	50000	9630935633	1955-01-02	Industry Conferences Event	Book& Pay
3		indore	500	500	9832867765	2022-07-27	Trade Shows and Expos	Book& Pay

Event Management !! Hello, manish(User)

Home About Book Booking Histroy Contact Guest

Payment Details

Person Name

Name

Card Number

1234 5678 435678

Expiry

MM/YYYY

CVV/CVC

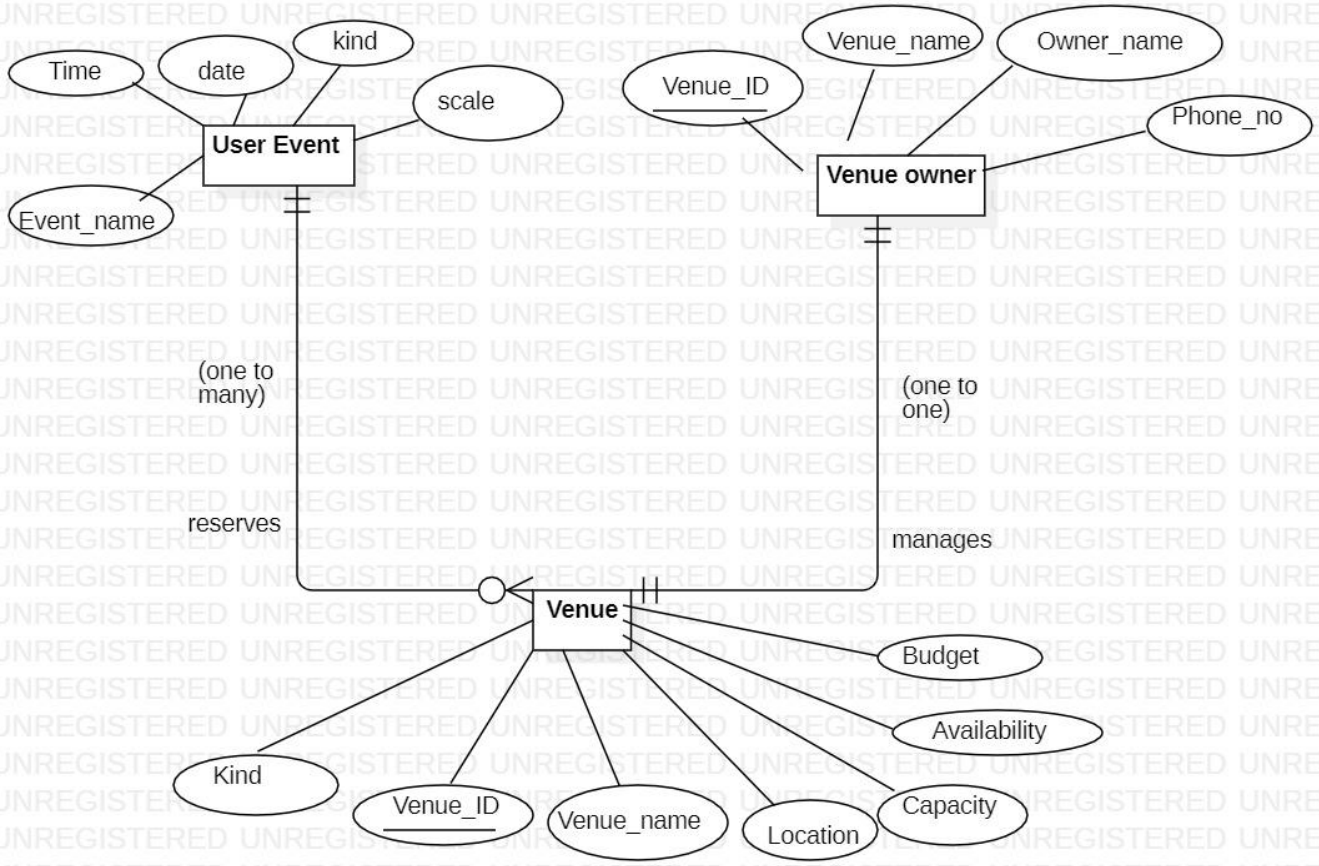
***

Pay

© 2022 Copyright: Event Management System

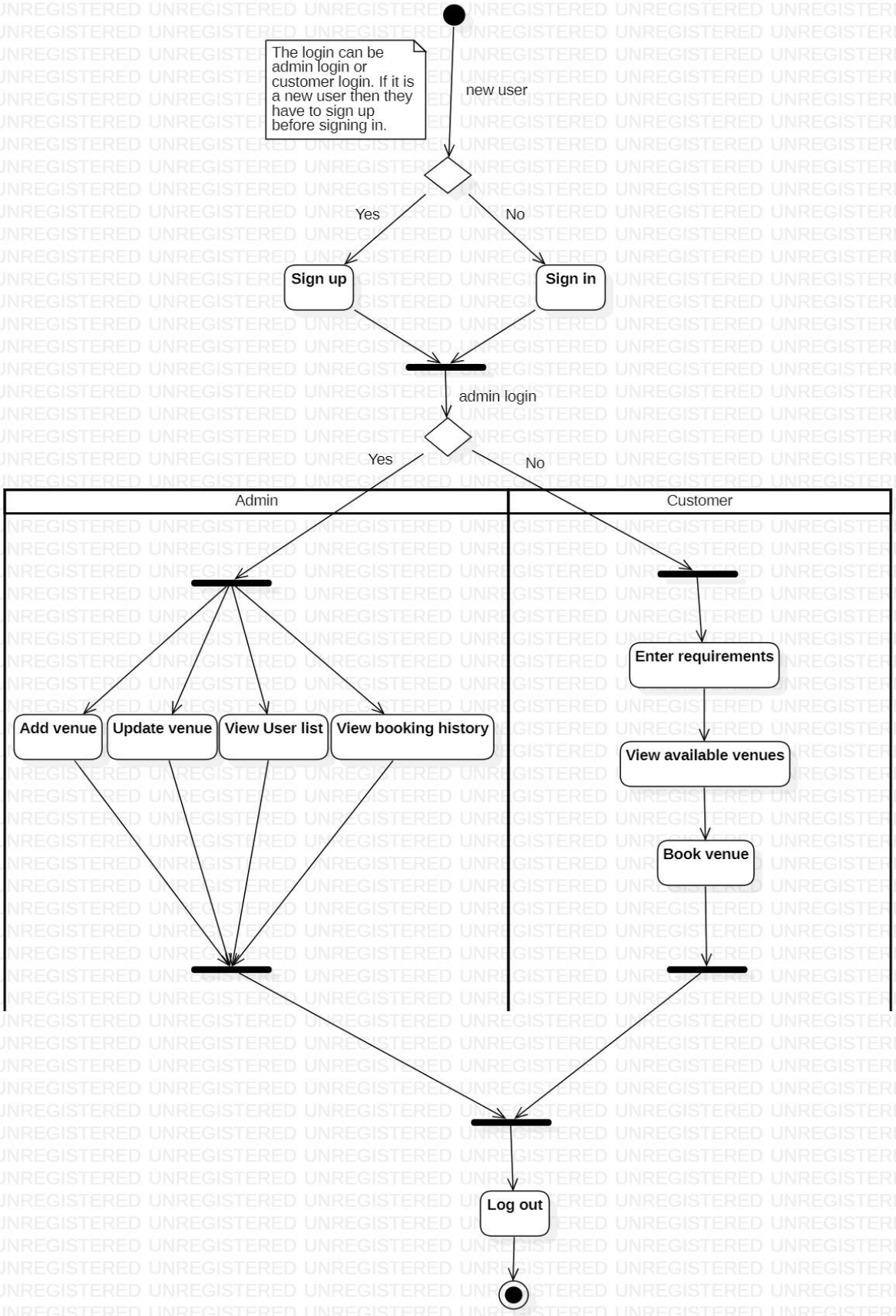
CHAPTER 6

ENTITY RELATIONSHIP DIAGRAM



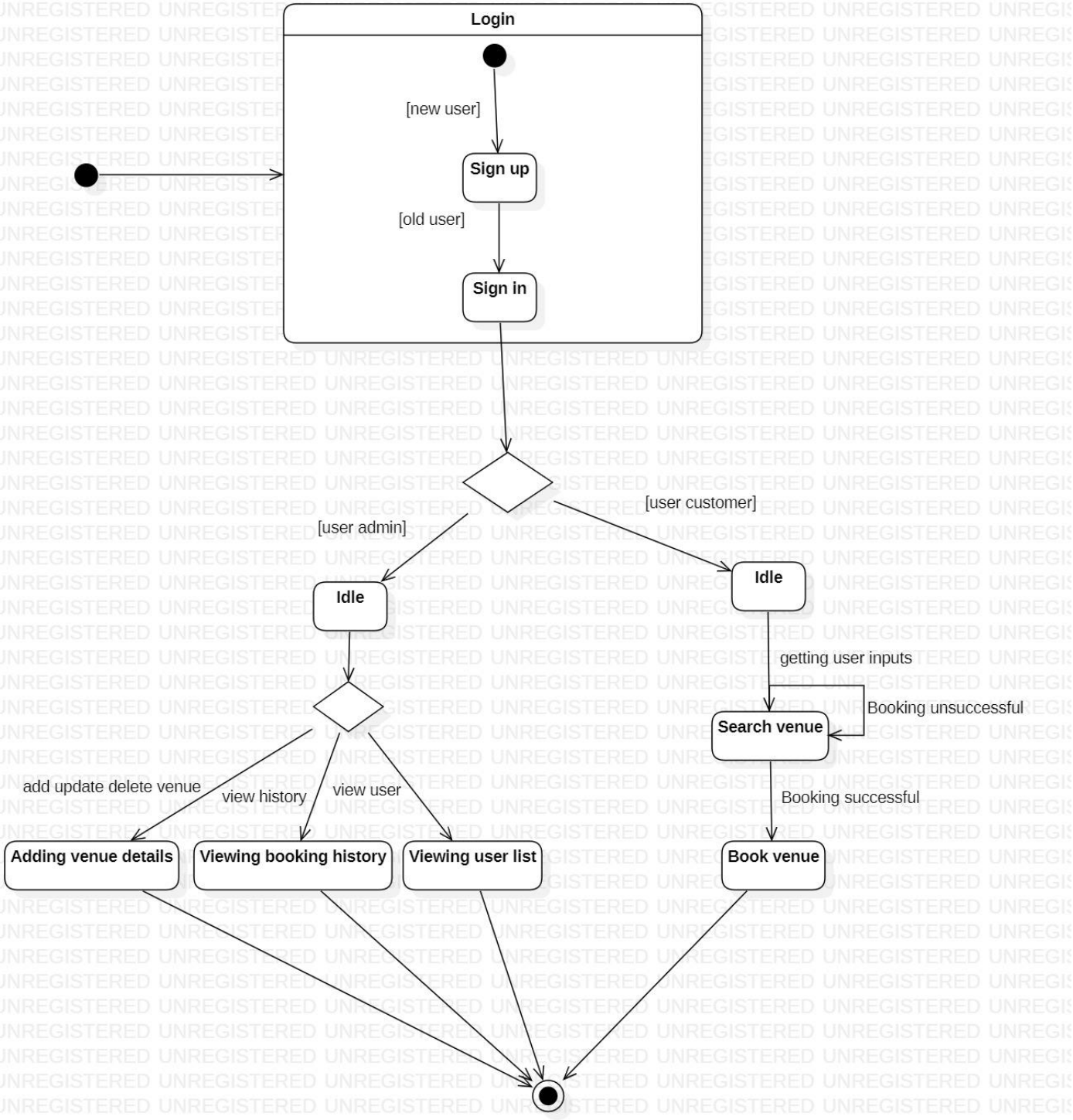
CHAPTER 7

ACTIVITY DIAGRAM



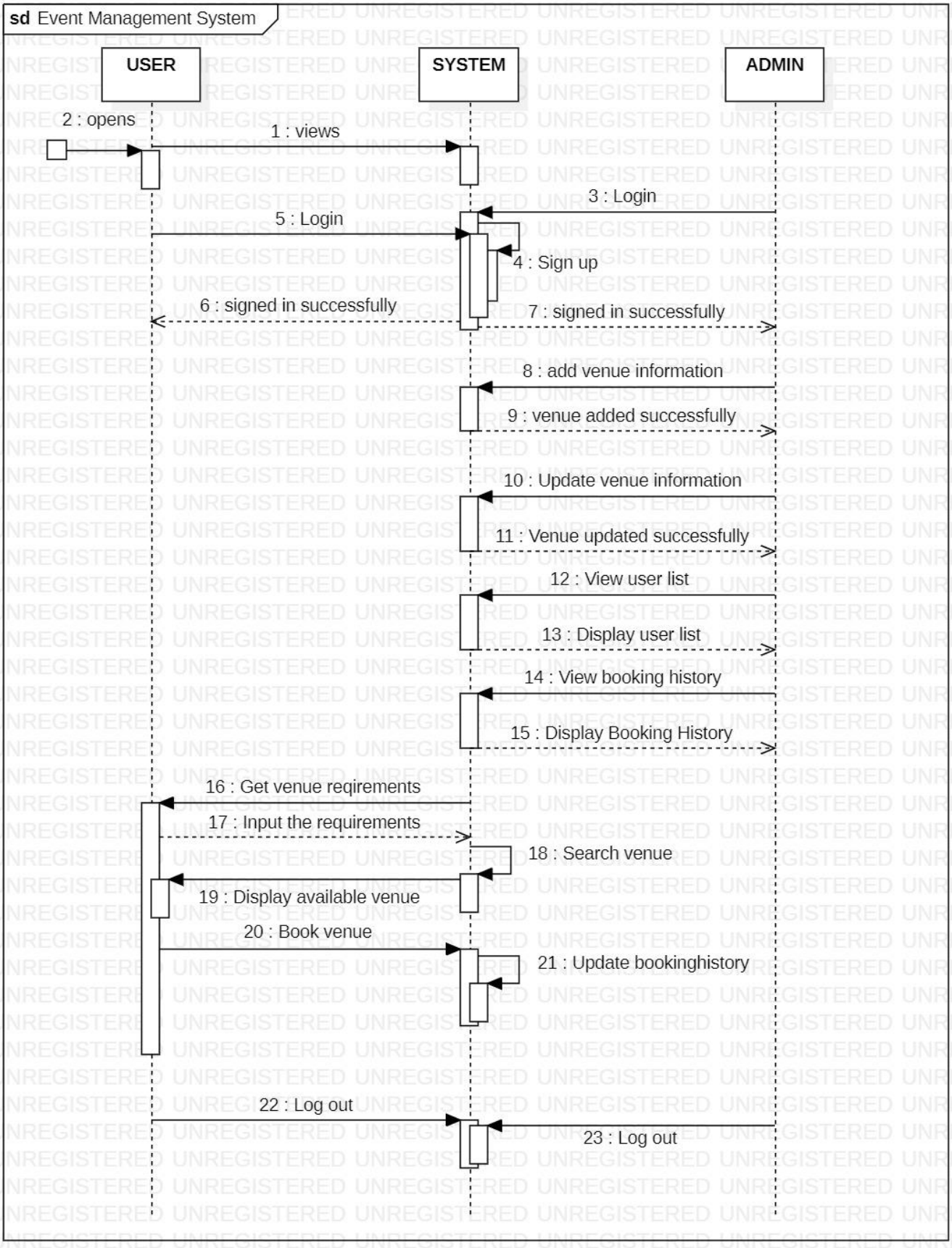
CHAPTER 8

STATE DIAGRAM



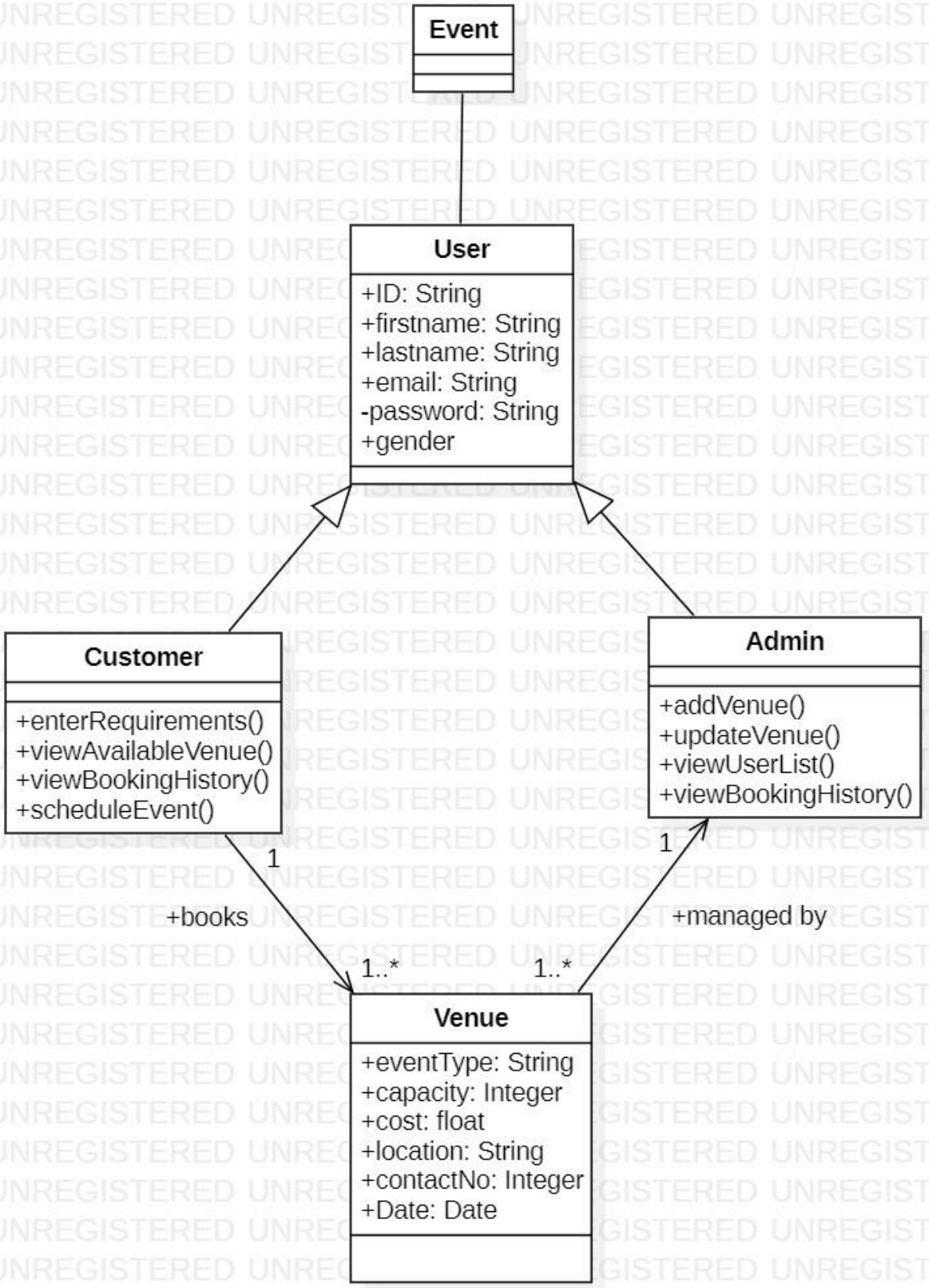
CHAPTER 9

SEQUENCE DIAGRAM



CHAPTER 10

DESIGN CLASS DIAGRAM



CHAPTER 11

TESTING

Software testing is a process of verifying and validating that a software application or program.

Meets the business and technical requirements that guided its design and development, and Works as expected.

TESTCASES FOR LOGIN:

Admin:

Username(Mail id)	Password
Admin123@gmail.com	Admin@1234
shroo@gmail.com	#txtfan#@1234
sheela@gmail.com	sheela@1234
123mahadevan	pwd

User:

Username(mail id)	Password
User123@gmail.com	User@123
rosa@gmail.com	rosa@1234
alfina@gmail.com	alfina@1234
User24.gmail.com	453rfgh78

TESTCASES FOR ADDING EVENT TYPE:

Event Type	Description
College festival	It's a platform for students to showcase their talents, interact with peers from different departments, and celebrate the vibrant spirit of campus life. College fests often span multiple days, offering an exciting lineup of events that cater to a variety of interests.
Industry Conferences Event	The goal of any great conference is to organize a group of people with related interests and provide them with helpful information on topics they care about. Conferences usually schedule multiple sessions spread out over days, weekends, or in some cases entire weeks.
Trade Shows and Expos	Trade shows and expos aim to present new products and services from a variety of related brands in a professional manner. Typically these types of events have a theme that ties the booths together. These types of events are almost always in-person because of the nature of hands-on opportunities and trade show booths.

TESTCASES FOR ADDING VENUE:

Event type	Capacity	Cost	Location	Contact	Date
Industry conference	500	5000	Coimbatore	9150045461	2023-07-09
School event	745	18000	Salem	9500643885	2023-08-07
null	100	2000	Tirunelveli	9645934777	2023-06-02
Wedding	1000	100000	Chennai	7654388999	Null
Trade fair	null	67000	Madurai	null	2023-11-23

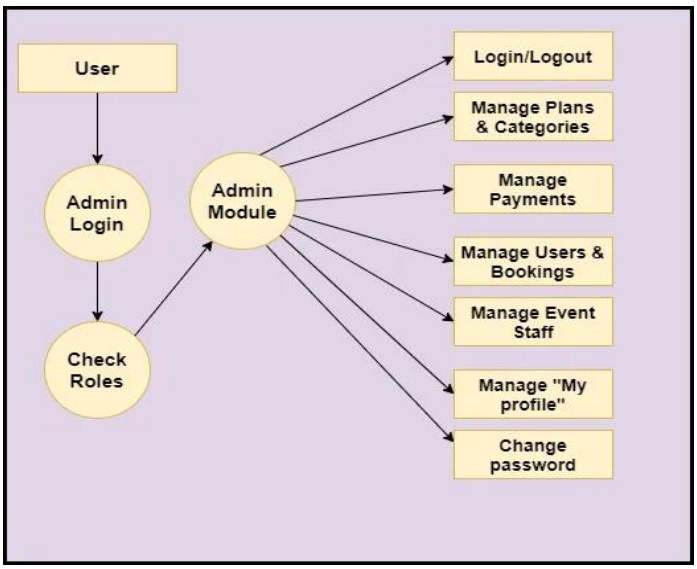
TESTCASES FOR PAYMENT PAGE:

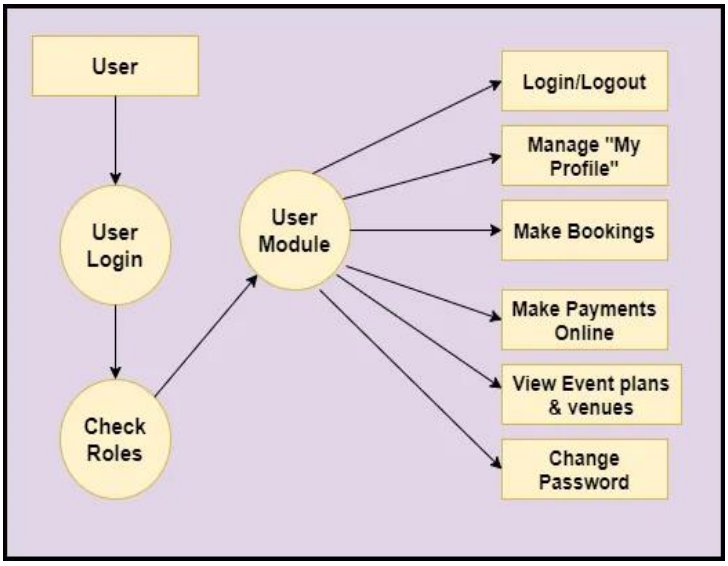
Person name	Expiry	Card number	CVV
Rajesh	09/23	1234 5678 123456	528
Shinee	10th July, 2026	5436 7898 234578	889
Priya	10/24	54367898234578	624
Roshini	08/25	5100 8745 239564	*89
Alfina	04/25	5123 8895 939164	abc

CHAPTER 12

CONCLUSION

Event Management System is user friendly and cost effective system, it is customized with activities related to event management life-cycle. In this project, we made attempt to effectively introduce the concept of event management systems already existing in the society. We then explain the concept of online event management systems which are already present. We describe the proposed system and explain the features implemented by our proposed system. We also give a brief overview of the technologies used during the development of our proposed system. This project can be further refined and extended by introducing new and more innovative features.





APPENDIX

```
package in.co.online.Bean;


import java.sql.Blob;
import java.util.Date;


public class VenueBean extends BaseBean {


	private  long eventtypeid;
	private String location;
	private String capacity;
	private String cost;
	private Date date;
	private String contact;
	private Blob image;
	private String eventtype;


	public String getEventtype() {
		return eventtype;
	}
	public void setEventtype(String eventtype) {
		this.eventtype = eventtype;
	}
	public long getEventtypeid() {
		return eventtypeid;
	}
	public void setEventtypeid(long eventtypeid) {
		this.eventtypeid = eventtypeid;
	}
	public String getLocation() {
		return location;
	}
	public void setLocation(String location) {
		this.location = location;
```

```
    }
    public String getCapacity() {
        return capacity;
    }
    public void setCapacity(String capacity) {
        this.capacity = capacity;
    }
    public String getCost() {
        return cost;
    }
    public void setCost(String cost) {
        this.cost = cost;
    }
    public Date getDate() {
        return date;
    }
    public void setDate(Date date) {
        this.date = date;
    }
    public String getContact() {
        return contact;
    }
    public void setContact(String contact) {
        this.contact = contact;
    }
    public Blob getImage() {
        return image;
    }
    public void setImage(Blob image) {
        this.image = image;
    }

}
```

```
package in.co.online.Controller;

import java.io.IOException;
import java.io.InputStream;
import java.sql.Blob;
import java.util.List;

import javax.servlet.ServletException;
import javax.servlet.annotation.MultipartConfig;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.Part;
import javax.sql.rowset.serial.SerialBlob;

import in.co.online.Bean.BaseBean;
import in.co.online.Bean.EventTypeBean;
import in.co.online.Bean.UserBean;
import in.co.online.Bean.VenueBean;
import in.co.online.Model.EventTypeModel;
import in.co.online.Model.UserModel;
import in.co.online.Model.VenueModel;
import in.co.online.Utility.DataUtility;
import in.co.online.Utility.DataValidater;
import in.co.online.Utility.PropertyReader;
import in.co.online.Utility.ServletUtility;

/**
 * Servlet implementation class VenueCtl
 */
@WebServlet(name = "VenueCtl", urlPatterns = "/venue")
```

```
@MultipartConfig(maxFileSize = 16177215)
public class VenueCtl extends BaseCtl {
	private static final long serialVersionUID = 1L;
	public static final String OP_SUBMIT = "Submit";



	protected boolean validate(HttpServletRequest request) {
		boolean pass = true;
		if (DataValidater.isNull(request.getParameter("eventtypeid"))) {
			request.setAttribute("eventtypeid", PropertyReader.getvalue("error.require", "EventTypeid"));
			pass = false;
		}
		if (DataValidater.isNull(request.getParameter("location"))) {
			request.setAttribute("location", PropertyReader.getvalue("error.require", "Location"));
			pass = false;
		}
		if (DataValidater.isNull(request.getParameter("capacity"))) {
			request.setAttribute("capacity", PropertyReader.getvalue("error.require", "Capacity"));
			pass = false;
		}
		if (DataValidater.isNull(request.getParameter("cost"))) {
			request.setAttribute("cost", PropertyReader.getvalue("error.require", "Cost"));
			pass = false;
		}
		if (DataValidater.isNull(request.getParameter("date"))) {
			request.setAttribute("date", PropertyReader.getvalue("error.require", "Date"));
			pass = false;
		}
		if (DataValidater.isNull(request.getParameter("contact"))) {
			request.setAttribute("contact", PropertyReader.getvalue("error.require", "Contact"));
			pass = false;
		}
		return pass;
	}
```

```
/**
 * @see HttpServlet#HttpServlet()
 */
public VenueCtl() {
	super();
	// TODO Auto-generated constructor stub
}

/**
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
 *     response)
 */
protected BaseBean populateBean(HttpServletRequest request){
	VenueBean bean = new VenueBean();
	bean.setId(DataUtility.getlong(request.getParameter("id")));
	bean.setEventtypeid(DataUtility.getlong(request.getParameter("eventtypeid")));
	bean.setLocation(DataUtility.getString(request.getParameter("location")));
	bean.setCapacity(DataUtility.getString(request.getParameter("capacity")));
	bean.setCost(DataUtility.getString(request.getParameter("cost")));
	bean.setDate(DataUtility.getDate(request.getParameter("date")));
	bean.setContact(DataUtility.getString(request.getParameter("contact")));
	System.out.println("cost:"+bean.getCost());
	System.out.println("cost1:"+bean.getDate());
	System.out.println("cost2:"+bean.getCapacity());
	System.out.println("cost3:"+bean.getContact());
	System.out.println("cost4:"+bean.getEventtypeid());
	Blob blob = null;
	Part filepart;
	try {
		filepart =  request.getPart("image");
		blob = medicinePacketUpload(filepart);
	} catch (Exception e) {
```

```
		}
		bean.setImage(blob);
		System.out.println("cost6:"+bean.getImage());
		populateDto(bean, request);
		return bean;
	}

	public Blob medicinePacketUpload(Part part) throws IOException {
		System.out.println("this si part :" + part);
		InputStream inputStream = null;
		Blob blob = null;
		inputStream =  part.getInputStream();
		byte[] b = new byte[inputStream.available()];
		inputStream.read(b);
		try {
			blob = new SerialBlob(b);

		} catch (Exception e) {
		}

		return blob;
	}

	protected void doGet(HttpServletRequest request, HttpServletResponse response)
			throws ServletException, IOException {

		ServletUtility.forward(getView(), request, response);
	}

	/**
	 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
	 *     response)
	 */
	protected void doPost(HttpServletRequest request, HttpServletResponse response)
```

```
                throws ServletException, IOException {
        System.out.println("in do post");
        long id = DataUtility.getlong(request.getParameter("id"));
        String op = DataUtility.getString(request.getParameter("operation"));
        VenueModel model = new VenueModel();
         VenueBean bean = new VenueBean();
        if (OP_SUBMIT.equalsIgnoreCase(op)) {
        bean =   (VenueBean) populateBean(request);
        try {
                        long pk =  model.add(bean);
                        ServletUtility.setbean(bean, request);
                        ServletUtility.setSuccessMessage("Venue ADD Successfully", request);
                        ServletUtility.forward(getView(), request, response);
                        return;
                } catch (Exception e) {
                        e.printStackTrace();
                }

        }
        System.out.println("forword");
    ServletUtility.forward(getView(), request, response);
    }

    @Override
    protected String getView(){
        return EM_View.VENUE_VIEW;
    }

}
```

package in.co.online.Model;

```
import java.sql.Connection;
import java.sql.Date;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;

import in.co.online.Bean.VenueBean;
import in.co.online.Exception.ApplicationException;
import in.co.online.Utility.JDBCDataSource;

public class VenueModel {

	public Integer nextpk() {
		Connection conn = null;
		int pk = 0;
		try {
			conn = JDBCDataSource.getconnection();
			PreparedStatement ps = conn.prepareStatement("SELECT MAX(ID) FROM venue");
			ResultSet rs = ps.executeQuery();
			while (rs.next()) {
				pk = rs.getInt(1);
			}

		} catch (Exception e) {

		}
		return pk + 1;
	}

	public long add(VenueBean bean) throws Exception {
```

```
System.out.println("in add method");
Connection conn = null;
int pk = 0;
try {
	conn = JDBCDataSource.getconnection();
	pk = nextpk();
	conn.setAutoCommit(false);
	PreparedStatement ps = conn.prepareStatement("INSERT INTO venue
VALUES(?,?,?,?,?,?,?,?,?,?,?,?)");
	ps.setLong(1, pk);
	ps.setLong(2, bean.getEventtypeid());
	ps.setString(3, bean.getLocation());
	ps.setString(4, bean.getCapacity());
	ps.setString(5, bean.getCost());
	ps.setDate(6, new Date(bean.getDate().getTime()));
	ps.setString(7, bean.getContact());
	ps.setBlob(8, bean.getImage());
	ps.setString(9, bean.getCreatedby());
	ps.setString(10, bean.getModifiedby());
	ps.setTimestamp(11, bean.getCreateddatetime());
	ps.setTimestamp(12, bean.getModifieddatetime());
	ps.executeUpdate();
	conn.commit();
	ps.close();

} catch (Exception e) {
	throw new ApplicationException("Exception : add rollback exception " +
e.getMessage());
}finally {
	JDBCDataSource.closeconnection(conn);
}
return pk;
```

```
        }

        public List list() throws Exception {
                ArrayList list = new ArrayList();
                try {
                        Connection conn = null;
                        conn = JDBCDataSource.getconnection();
                        PreparedStatement ps =
        conn.prepareStatement("SELECT
venue.id,eventtype.eventname,capacity,cost,image,date,contact,location FROM venue INNER JOIN
eventtype ON venue.eventtypeid=eventtype.id");
                        ResultSet rs = ps.executeQuery();
            while (rs.next()) {
                                VenueBean  bean = new VenueBean();
                                bean.setId(rs.getLong(1));
                                bean.setEventtype(rs.getString(2));
                                bean.setCapacity(rs.getString(3));
                                bean.setCost(rs.getString(4));
                                bean.setImage(rs.getBlob(5));
                                bean.setDate(rs.getDate(6));
                                bean.setContact(rs.getString(7));
                                bean.setLocation(rs.getString(8));
                                list.add(bean);
            }

                } catch (Exception e) {
                        e.printStackTrace();
                }
                return list;
}
}
```

```
<%@page import="in.co.online.Bean.VenueBean"%>
<%@page import="in.co.online.Controller.VenueCtl"%>
<%@page import="java.util.List"%>
<%@page import="java.sql.ResultSet"%>
<%@page import="java.sql.PreparedStatement"%>
<%@page import="in.co.online.Utility.JDBCDataSource"%>
<%@page import="java.sql.Connection"%>
<%@page import="in.co.online.Utility.ServletUtility"%>
<%@page import="in.co.online.Utility.DataUtility"%>
<%@page import="java.util.Iterator"%>

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
	pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet"
	href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css">
<link rel="stylesheet"
	href="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-datepicker/1.9.0/css/bootstrap-
datepicker.min.css">
<link rel="stylesheet"
	href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-
awesome.min.css">
<script
	src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script
	src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js"></scr
ipt>
<script
	src="https://cdnjs.cloudflare.com/ajax/libs/bootstrap-datepicker/1.9.0/js/bootstrap-
datepicker.min.js"></script>
<script type="text/javascript">
	$('.datepicker').datepicker();
</script>
<meta charset="ISO-8859-1">
<title>Venue</title>
```

```
</head>
<body>
        <%@include file="Header.jsp"%>
        <br>
        <div class="container">
                <div class="row">
                        <div class="col-2"></div>
                        <div class="col-8"   style="background-color: orange;">
                                <form action="<%=EM_View.VENUE_CTL%>" method="post"
                                        enctype="multipart/form-data">



                                        <h2>VENUE</h2>
                                        <hr class="border border-primary border-3 opacity-75">
                                        <h6 style="color:
red;"><%=ServletUtility.getErrorMessage(request)%></h6>
                                        <h6 style="color:
green;"><%=ServletUtility.getSuccessMessage(request)%></h6>
                                        <jsp:useBean id="bean" scope="request"
                                                class="in.co.online.Bean.VenueBean" />


                                        <input type="hidden" name="id" value="<%=bean.getId()%>">
<input
                                                type="hidden" name="createdby"
value="<%=bean.getCreatedby()%>">
                                        <input type="hidden" name="modifiedby"
                                                value="<%=bean.getModifiedby()%>"> <input type="hidden"
                                                name="createdDatetime"
value="<%=bean.getCreateddatetime()%>">
                                        <input type="hidden" name="modifiedDateTime"
                                                value="<%=bean.getModifieddatetime()%>">


                                        <div class="container">
                                                <div class="col-md-12">
                                                        <label for="exampleInputPassword1" style="font-
family: cursive;">Event
                                                                Type:</label>
                                                <div class="form-group">
```

```
                                                <select class="custom-select" name=eventtypeid>
                                                        <%
                                                                Connection con =
JDBCDataSource.getconnection();
                                                                String sql = "SELECT * FROM
eventtype";
                                                                PreparedStatement ps =
con.prepareStatement(sql);
                                                                ResultSet rs = ps.executeQuery();
                                                                while (rs.next()) {
                                                        %>


                                                        <option value="--------Select--------
"></option>
                                                        <option
value="<%=rs.getLong(1)%>"><%=rs.getString(2)%></option>


                                                        <%
                                                                }
                                                        %>
                                                <select>
                                                <div class="form-text" style="color:
red"><%=ServletUtility.getErrorMessage("eventtypeid", request)%></div>
                                        <div>
                                <div>


                                <div class="col-12">
                                        <label for="inputAddress" class="form-
label">Capacity:</label> <input
                                                type="text" class="form-control"
                                                name="capacity" placeholder="Enter here..."

        value="<%=DataUtility.getStringData(bean.getCapacity())%>">
                                <div>
                                <font
color="red"><%=ServletUtility.getErrorMessage("capacity", request)%></font>


                                <div class="col-12">
                                        <label for="inputAddress" class="form-
label">Cost:</label> <input
```

```
                                                type="text" class="form-control"  name="cost"
                                                placeholder="Enter here..."

        value="<%=DataUtility.getStringData(bean.getCost())%>">
                                    </div>
                                    <font color="red"><%=ServletUtility.getErrorMessage("cost",
request)%></font>



                                    <div class="col-12">
                                          <label for="inputAddress" class="form-
label">Location</label> <input
                                                type="text" class="form-control"
id="inputAddress"
                                                name="location" placeholder="Enter here..."

        value="<%=DataUtility.getStringData(bean.getLocation())%>">
                                    </div>
                                    <font
color="red"><%=ServletUtility.getErrorMessage("location", request)%></font>


<div class="col-md-12">
                                          <label for="form_message">Contact:</label> <input
                                                class="form-control" type="text" name="contact"
                                                placeholder="Enter here....."

        value="<%=DataUtility.getStringData(bean.getContact())%>">
                                    </div>
                                    <div class="form-text" style="color:
red"><%=ServletUtility.getErrorMessage("contact", request)%></div>


<div class="col-md-12">


                                    <label for="form_message">Date:</label>


                                    <div class="form-group">
                                          <input type="text" class="form-control"
id="exampleInputEmail1"
```

```
                                                aria-describedby="emailHelp" name="date"
id="datepicker"
                                                data-provide="datepicker"

        value="<%=DataUtility.getStringData(bean.getDate())%>"
                                                placeholder="date Enter Here"> <font
color="red"><%=ServletUtility.getErrorMessage("date", request)%></font>
                                    </div>
</div>




                            <div class="col-md-12">
                                        <label for="exampleFormControlInput1" class="form-
label">Event
                                                Photo:</label> <br><input type="file"
id="exampleFormControlInput1"
                                                name="image">
                                    </div>


                                    <br>
                                     <input type="submit" class="btn btn-primary"
                                          name="operation" style="margin-left: 130px;"
                                          value="<%=VenueCtl.OP_SUBMIT%>">
</div>
                            </form>
                        </div>
                    </div>
        </div>



        <div style="margin-top: 2%;">
                <%@include file="Footer.jsp"%>
        </div>
</body>
                        </html>
```