

Governors State University

OPUS Open Portal to University Scholarship

All Capstone Projects

Student Capstone Projects

Fall 2023

Event Management System

Pramidha Kaleru

Governors State University

Follow this and additional works at: <https://opus.govst.edu/capstones>

Recommended Citation

Kaleru, Pramidha, "Event Management System" (2023). *All Capstone Projects*. 674.

<https://opus.govst.edu/capstones/674>

For more information about the academic degree, extended learning, and certificate programs of Governors State University, go to http://www.govst.edu/Academics/Degree_Programs_and_Certifications/

Visit the [Governors State Computer Science Department](#)

This Capstone Project is brought to you for free and open access by the Student Capstone Projects at OPUS Open Portal to University Scholarship. It has been accepted for inclusion in All Capstone Projects by an authorized administrator of OPUS Open Portal to University Scholarship. For more information, please contact opus@govst.edu.

EVENT MANAGEMENT SYSTEM

By

PRAMIDHA KALERU

B.TECH, JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, 2021

GRADUATE CAPSTONE SEMINAR PROJECT

Submitted in partial fulfillment of the requirements

For the Degree of Master of Science,

With a Major in Computer Science



Governors State University
University Park, IL 60484

2023

ABSTRACT

The main aim of the event management system is to provide a platform for the users to view information about the events that took place in the past and the ones that are about to take place shortly. The application enables the user to view and book an ongoing or upcoming event in a particular locality. Users should register themselves to log into the account for online booking of tickets. The users can be administrators, event organizers, and customers. They can first log into the website and see through the information such as details about the events.

The administrator can log in and update the information, delete any unwanted data, and arrange the information accordingly. The event organizer is the one who organizes events; they will upload event details. Customers can view events and they can book events. There is a secured mechanism is there for payments. This web application also gives some suggestions to the customers based on their previous bookings. Customers can view or send feedback on particular events. A Centralized database is used to store all the details of the admin, event organizer and customer details.

Event Management System is developed using HTML, CSS, and Bootstrap as the front end, MySQL database as the backend and Java as a programming language. We will develop this web application using the Spring MVC framework. Tomcat server is used for deployment.

Table of Content

1	<i>Project Description</i>	1
1.1	Competitive Information	1
1.2	Relationship to Other Applications/Projects	1
1.3	Assumptions and Dependencies	1
1.4	Future Enhancements	2
1.5	Definitions and Acronyms	2
2	<i>Technical Description</i>	2
2.1	Project/Application Architecture	2
2.2	Project/Application Information flows	4
2.3	Interactions with other Projects (if Any)	5
2.4	Interactions with other Applications	5
2.5	Capabilities	5
2.6	Risk Assessment and Management	6
2.7	User Based Recommendation System using Apache Mahout	6
3	<i>Project Requirements</i>	8
3.1	Identification of Requirements	8
3.2	Operations, Administration, Maintenance and Provisioning (OAM&P)	8
3.3	Security and Fraud Prevention	9
3.4	Release and Transition	9
4	<i>Project Design Description</i>	9
5	<i>Project Internal/external Interface Impacts and Specification</i>	11
6	<i>Project Design Units Impacts</i>	14
6.1	Functional Area/Design Unit A	15
6.1.1	<i>Functional Overview</i>	15
6.1.2	<i>Impacts</i>	15
6.1.3	<i>Requirements</i>	15
6.2	Functional Area/Design Unit B	15
6.2.1	<i>Functional Overview</i>	15
6.2.2	<i>Impacts</i>	15
6.2.3	<i>Requirements</i>	15
7	<i>Open Issues</i>	15
8	<i>Acknowledgements</i>	16
9	<i>References</i>	16
10	<i>Appendices</i>	16

1 Project Description

An Event Management System is a web application using Java as a programming language to display events that take place in a particular location. We have many websites available on the internet to display event details. But Our Event Management System not only displays event details but also user can book events and displays recommendations for the users; also a centralized database is maintained to store all the operations.

Event Management system is a non-profitable organization that helps to establish better communication between event organizers and users. This system is useful for both organizers and users. In the proposed web application once the user logs into the system, he will be welcomed with recommended events for this we have used Apache Mahout API so, the user can view different events, book events and payments is also done. After successful login user can send feedback. A rich user interface is designed to perform this operation. Event organizers are registered with the system and their details are verified by the administrator. Being an authorized user he can publish event details and he can view booking details also. The main advantage of the proposed system is every transaction will be recorded into the database using a secured mechanism. Users and event organizers can view previous event booking details. It is a web application; multiple users can access simultaneously using the internet.

1.1 Competitive Information

Event Management System is a Java-based web application to displays various event details that take place in different locations. There are different websites available over the internet to perform this operation. Our website can accept bookings from different locations and a centralized database is maintained to store all the operations. This system's user interface is developed with Bootstrap styles so users can navigate from one location to another location easily. UI is simple so no need to try people to use this website.

1.2 Relationship to Other Applications/Projects

The main objective of our application is to display events in different locations. There are many sites available in the market that also performs similar operations but they provide different options to users. So, some of the functionalities are inherited from existing websites.

1.3 Assumptions and Dependencies

- Entire application works under administrator. So all the events uploaded by event organizer are authorized by the admin. After getting approval from admin, event details are displayed to users.

- Only registered users can view event details.
- After successful login only event organizers can post event details
- Admin can view all the booking details
- User can send feedback only after booking events

1.4 Future Enhancements

In the future, we can have some more user-friendly functionality with advanced filters and UX improvements [1]. We add email and mobile SMS notifications. Whenever an event organizer adds any new events all the users are intimated by email or SMS. We can implement a real-time payment mechanism using PayPal for bookings.

1.5 Definitions and Acronyms

UI - User Interface

VX – User Experience

JSP – Java Server Page

ER – Entity Relation.

User – is the one can view event details and he can book events.

Event Organizer – is the one who uploads event details and he can view event bookings also.

2 Project Technical Description

The proposed system is developed with 3 modules Admin, Event Organizer and User. The system is designed with rich UI, so users can switch from one page to another easily.

2.1 Application Architecture

The system architecture acts as a template for a system. The system architecture describes number of modules, operations and relationship between components. Figure1 shows system architecture of Event Management System web site.

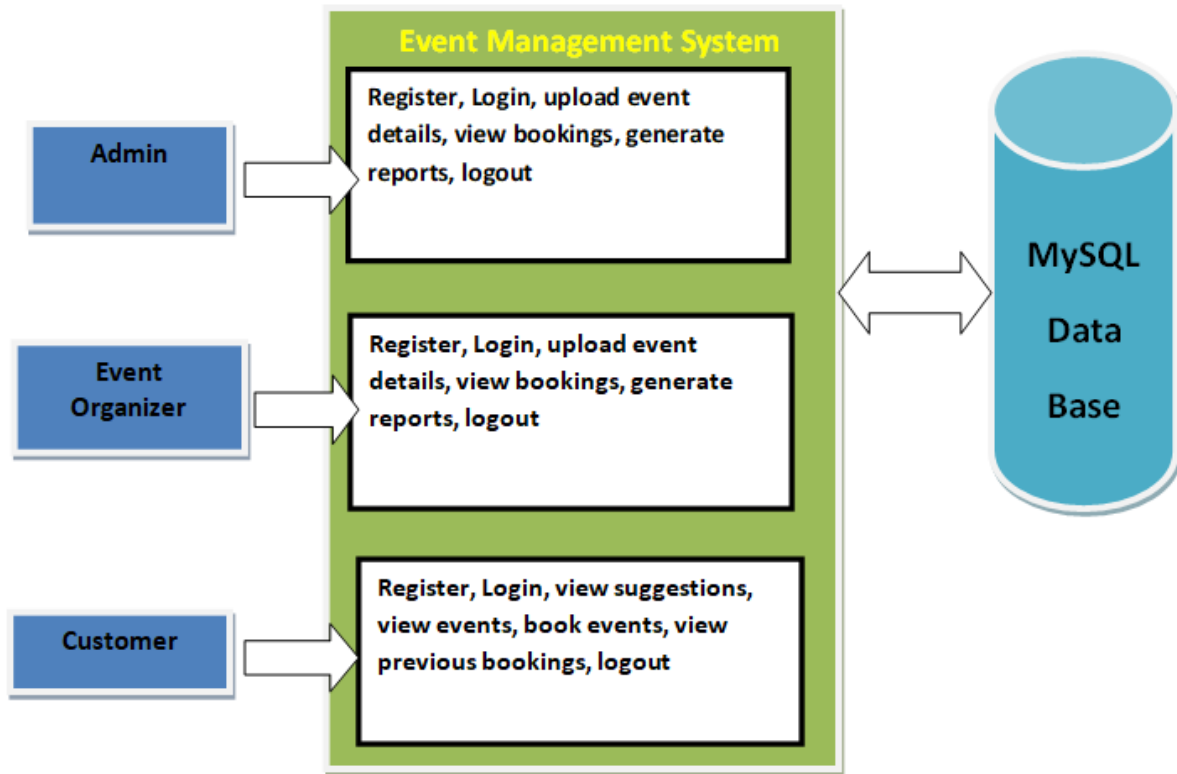


Figure1. System Architecture of Event Management System

This application is developed based on Spring MVC Framework. Figure2 shows technical architecture of Event Management System.

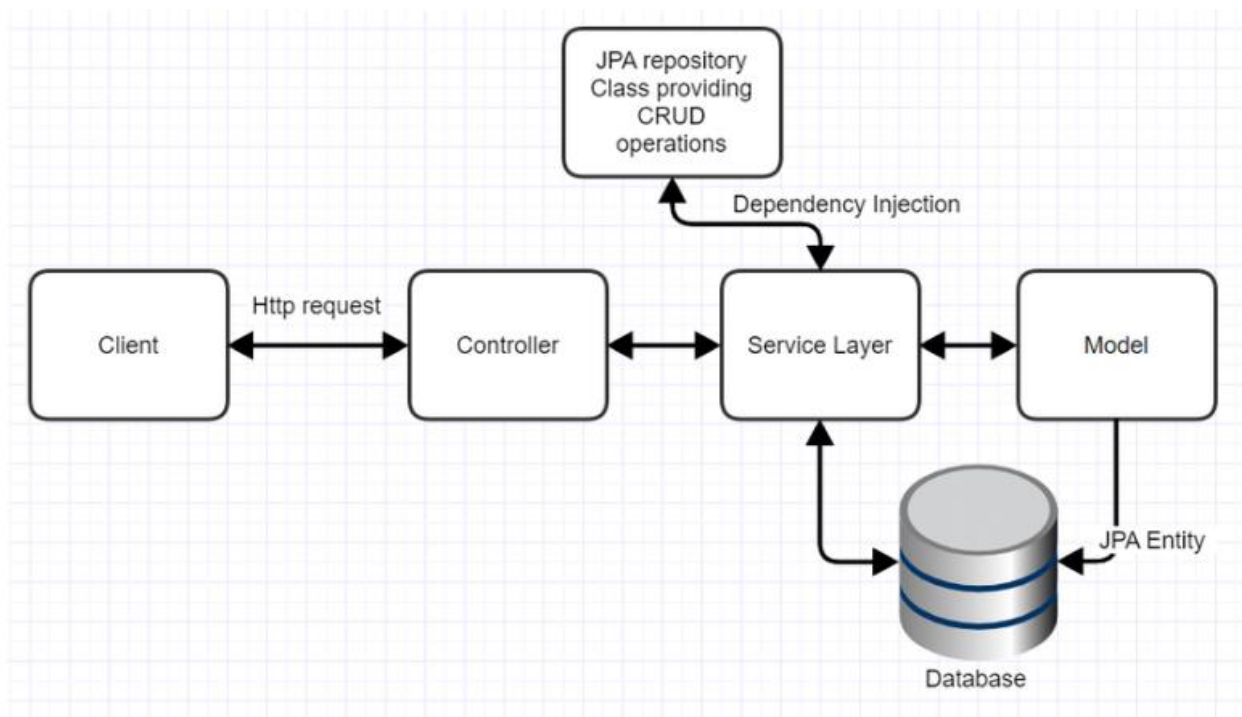


Figure2. Technical Architecture of Event Management System

In Spring MVC, Servlet acts as a controller, spring framework acts as model and JSP acts as view component. Whenever the client sends a request server, the server forwards the request to the Controller i.e. Servlet. The controller handles the request. The controller forwards the request to the model. A model is a Java object which is responsible for generating the business logic of the application. The Model can communicate with the database to store and retrieve data. The Model sends a generated response to the controller. The controller will send the data to the view component. The view is used to display the data present in the Model.

2.2 Application Information flows

DFD (Data Flow Diagram) diagram gives the flow of information for any system. The figure 3 and 4 shows DFD diagrams of admin.

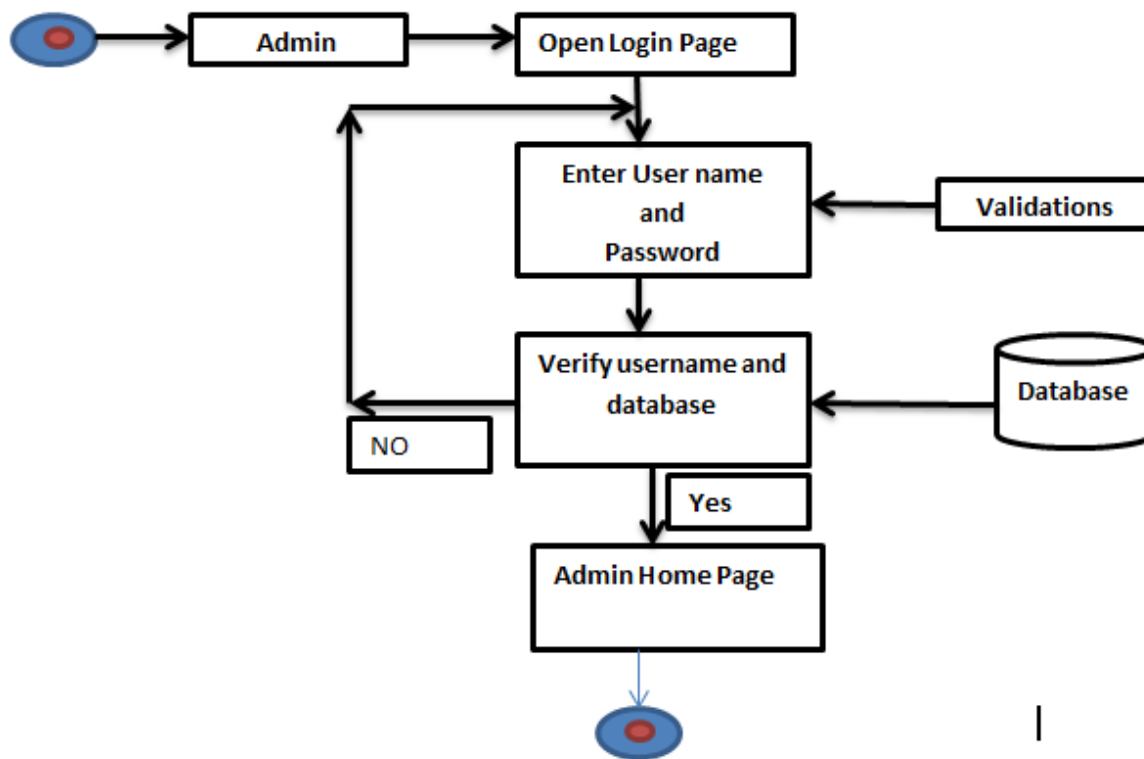


Figure3. Level-0 DFD for Admin

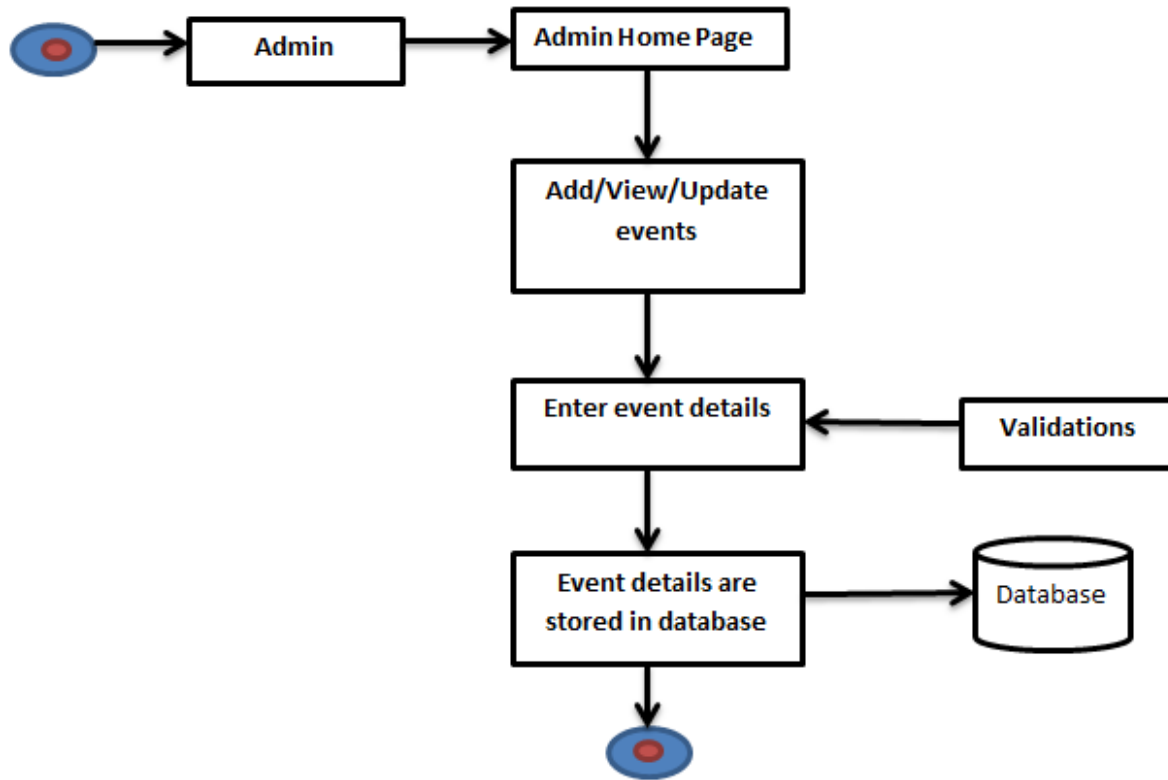


Figure4. Level-1 DFD for Admin

2.3 Interactions with other Projects (if Any)

Event Management System is an independent application. This application is designed by taking <https://www.patentpendingnyc.com> as a reference. Some of the functionalities are the same like displaying event details and bookings etc.

2.4 Interactions with other Applications

The proposed system is designed to interact with other online event management websites like <https://www.patentpendingnyc.com>, and [cevent.com](https://www.cevent.com). Some of the functionalities are similar to those of web applications. We also designed to communicate with different payment gateways for online payments.

2.5 Capabilities

This web application is implemented using different free source software. For the View component, we used HTML, CSS and Bootstrap. MySQL is used as a database. Using model classes our application performs insert, delete, update and select operations on the database. Whenever a new user registers with

the system model component perform an insert operation on the database. To display event details, the model component performs select operations on the database.

2.6 Risk Assessment and Management

The term risk management refers to finding, understanding and prioritizing risks. In any project development risk management plays a pivotal role because if any risk arises, the risk management team will identify the risk and mitigate it early to reduce impact on the system and they also update the risk management log for future reference. Risk management operates on the highest priority first principle i.e. higher priority risks are handled first.

2.7 Used Based Recommendation System using Apache Mahout

In this project we have implemented User Based Recommendation System using Apache Mahout [2]. Apache Mahout is an open source project which is widely used to build recommendation Engines. Apache Mahout implements popular machine learning techniques such as Recommendation, Classification and Clustering. Companies like Adobe, Facebook, LinkedIn, Foursquare etc. are using Apache Mahout internally. Mahout lets applications to analyze large sets of data effectively and in quick time. Using Mahout we provide recommendations to users.

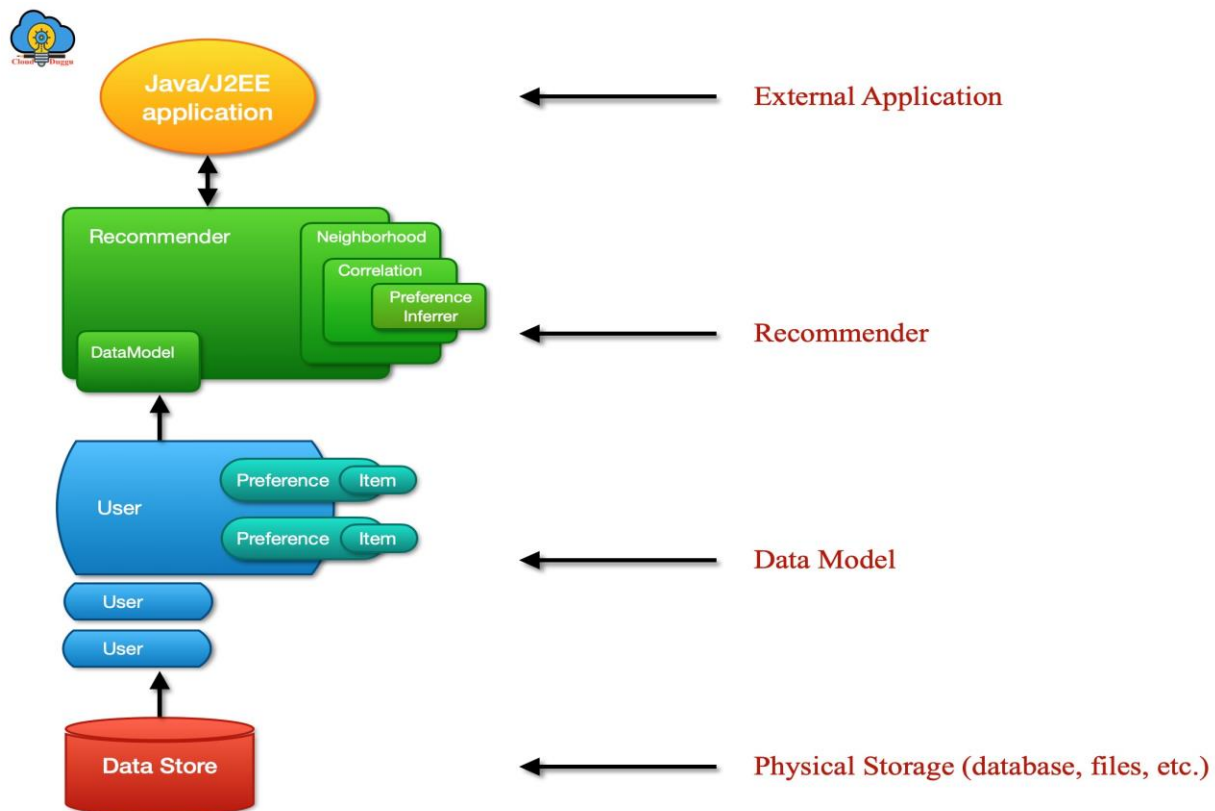


Figure5. Apace Mahout Architecture

Mahout implementation

1) We created a data set. Data set which is in the format of UserID, Event ID and Rating and saved data set as MDist1.csv file, anywhere on our computer system.

2) Updated pom.xml with mahout dependencies.

ex:

```
<dependency>
    <groupId>org.apache.mahout</groupId>
    <artifactId>mahout-mr</artifactId>
    <version>0.10.0</version>
</dependency>
```

3) Implementing recommendation System.

Sample Code:

```
try {
    DataModel model = new FileDataModel (new File("C:\\MDist1.csv"));
    CityBlockSimilarity similarity = new CityBlockSimilarity (model);
    UserNeighborhood neighborhood = new ThresholdUserNeighborhood (0.1, similarity, model);
    UserBasedRecommender recommender = new GenericUserBasedRecommender (model,
neighborhood, similarity);

    // The First argument is the userID and the Second parameter is 'HOW MANY recommendations'
    List<RecommendedItem> recommendations = recommender.recommend (userid, 5);
    for (RecommendedItem recommendation : recommendations){
        String result =recommendation;
    }
}

Catch (Exception e) {
    e.printStackTrace ();
}
```

Procedure:

The similarity threshold which is a lower limit for the similarity of two data records that belong to the same cluster. For example, if you set the similarity threshold to 0.25, data records with field values that

are 25% similar are likely to be assigned to the same cluster. If you specify a similarity threshold of 1.0, then you are insisting that, for customers to appear in the same group, their characteristics must be identical. Finally we are interested in calculating n recommendations (Here 5) for UserId.

3 Project Requirements

For developing this web application we have used Agile methodology [3]. Agile is one type of incremental model. In this method entire project is divided into small incremental builds and these builds are developed in an iteration model. The first step in agile is to identify the requirements listed in the project proposal document.

3.1 Identification of Requirements

<GSU-eventmanagementsystem_001 Login-Functionality 000001>

Create login functionality for admin, event organizer and user. All the actors i.e. users are authenticated with username and password. If username and password are correct page redirects to home page otherwise application will generate alert message to user.

<GSU-eventmanagementsystem_001 Admin-Functionality 000001>

Create an interface page for admin home, after successful login admin can accept or reject events uploaded by the event organizer and also can view event details. Admin can also view event booking details also.

<GSU-eventmanagementsystem_001 User-Functionality 000001>

Create an interface page for user home, after successful login user can view event details, he can also book events and he can view previous bookings also.

<GSU-eventmanagementsystem_001 Eventorganizer-Functionality 000001>

Create an interface page for event organizer home, after successful login he can post event details and he can view previous bookings also.

<GSU-eventmanagementsystem_001 Add New Event-Functionality 000001>

Create an interface for adding new events. After successful login event organizer can post event details by entering required fields. Event details are stored in database once all the fields are validated.

<GSU-eventmanagementsystem_001 Book Events -Functionality 000001>

Create an interface page for displaying event details. After successful login users can view event details and he can book events by specifying credit/debit card details.

3.2 Operations, Administration, Maintenance and Provisioning (OAM&P)

The project involves different operations performed by the admin, event organizer and user. To perform any operation every user has to register and log in with the system. After successful login only they are allowed to perform various operations. Users can view recommendations, can view recent event details

and he can book events. The event organizer will post new event details and he can also view bookings of previous events. Admin will authorize events posted by the event organizer and he can also view event details and booking details. The application is maintained at regular intervals by giving security updates, reviewing code quality and updating the applications.

3.3 Security and Fraud Prevention

Sensitive information like user authentication details and payment details are posted in our web application. This could be exploited to steal an applicant's identity or commit financial fraud. We can detect any fraud operations with the following two steps.

- 1) Look out for the number of events added by the event organizer per day
- 2) How many events are booked by the user in a single day?

If an event organizer asks to pay an extra amount for an event then we consider it a fraud event. If a single user books more than the permissible limit per day then we consider it as a fraud.

3.4 Release and Transition Plan

The event management system is a web application independent of the operating system, technology and database. Our testing team has tested the application and satisfied all the requirements in the proposed document. After being tested by the QA [4] team, the application is ready to deploy on the server. To run the application all the required software's are installed and ready to test. Based on client requirements, upcoming releases are scheduled. After successful deployment, there will be maintenance and upgrades in periodical time intervals.

4 Project Design Description

The event Management System project is designed with three modules admin, event organizer and user. Different user interfaces are there for 3 modules. All the users can perform insert, delete, update and select operations on database.

Following are the operations performed by admin module

- 1) Login
- 2) View Event details
- 3) Accept or reject events
- 4) View event bookings
- 5) Logout

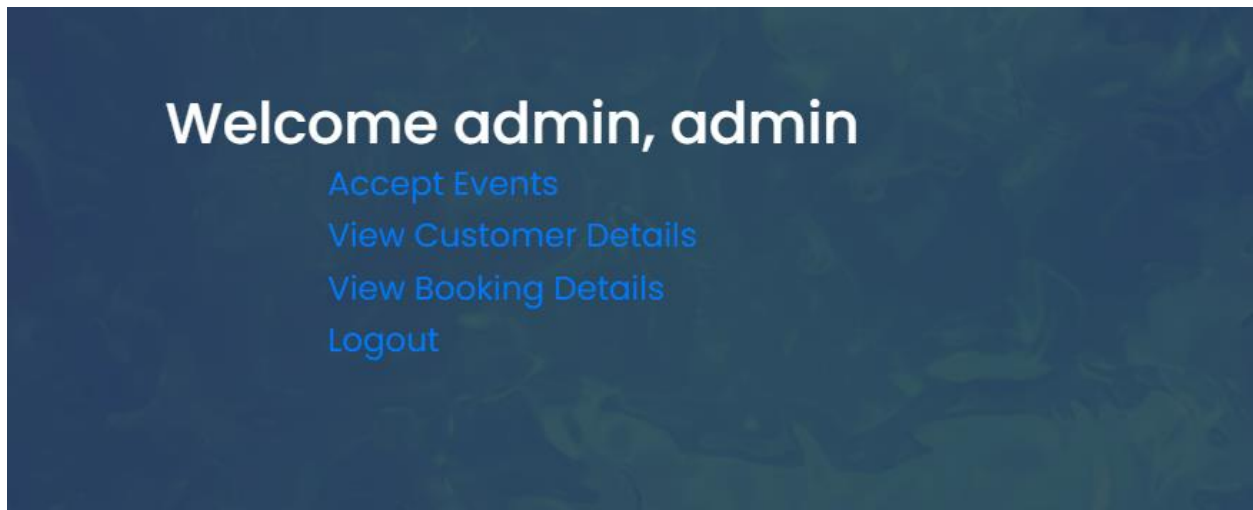


Figure6. Admin Home Page

Following are the operations performed by the event organizer module

- 1) Register1
- 2) Login11
- 3) Add new Events
- 4) View Bookings
- 5) Logout



Figure7. Event organizer Home page

Following are the operations performed by the user module

- 1) Register
- 2) Login
- 3) View recommendations
- 4) View Event Details
- 5) Book Events
- 6) Payment
- 7) View Previous Bookings

8) Send Feedback

9) Logout

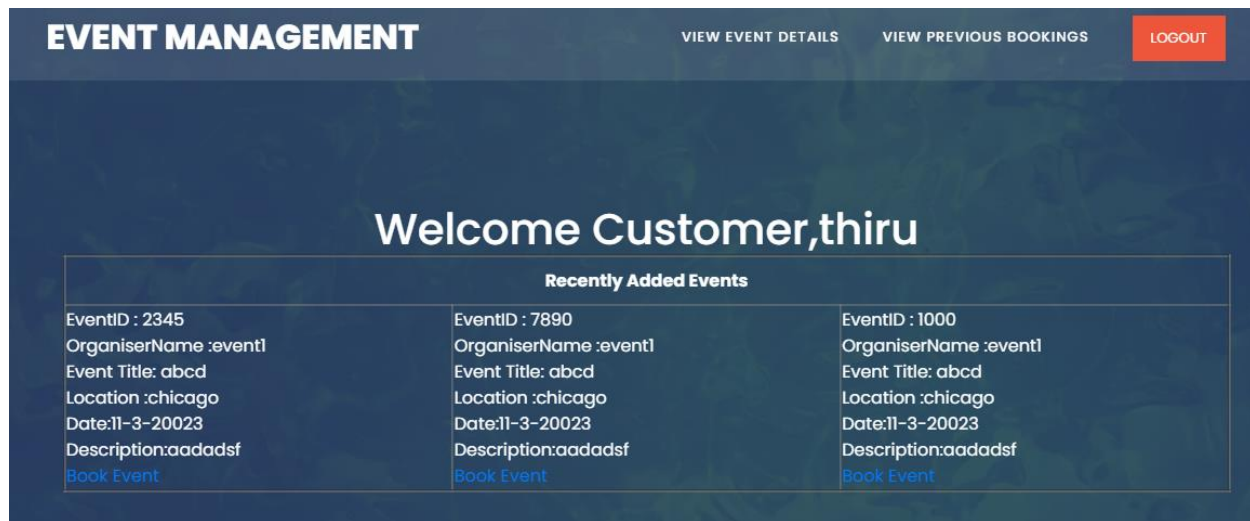


Figure8. User Home page

5 Internal/external Interface Impacts and Specification

Most of the web applications are 3-tier applications i.e. frontend, backend and middleware. For implementation, we have used different technologies at each level. For frontend implementation, we have used HTML, CSS and Bootstrap. MySQL database acts as the backend and spring framework acts as middleware. MySQL database is used to store all the transactions. MySQL is a simple and open-source relational database. We have configured the MySQL details in the Spring MVC configuration file.

```
<context:component-scan base-package="com.project.eventmanagement" />
<context:annotation-config />

<bean id="userService" class="com.project.eventmanagement.service.UserServiceImpl" />
<bean id="userDao" class="com.project.eventmanagement.dao.UserDaoImpl" />

<bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">
  <property name="dataSource" ref="datasource" />
</bean>

<bean id="datasource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
  <property name="driverClassName" value="com.mysql.jdbc.Driver" />
  <property name="url" value="jdbc:mysql://localhost:3306/eventmanagement" />
  <property name="username" value="root" />
  <property name="password" value="" />
</bean>
```

Figure9. MySQL configuration in spring MVC configuration file

Our web application Event Management System contains 5 database tables. Admin, Customer, Eventorganizer, Events and Bookings. Following figure 8 represents ER-Diagram of our web applications.

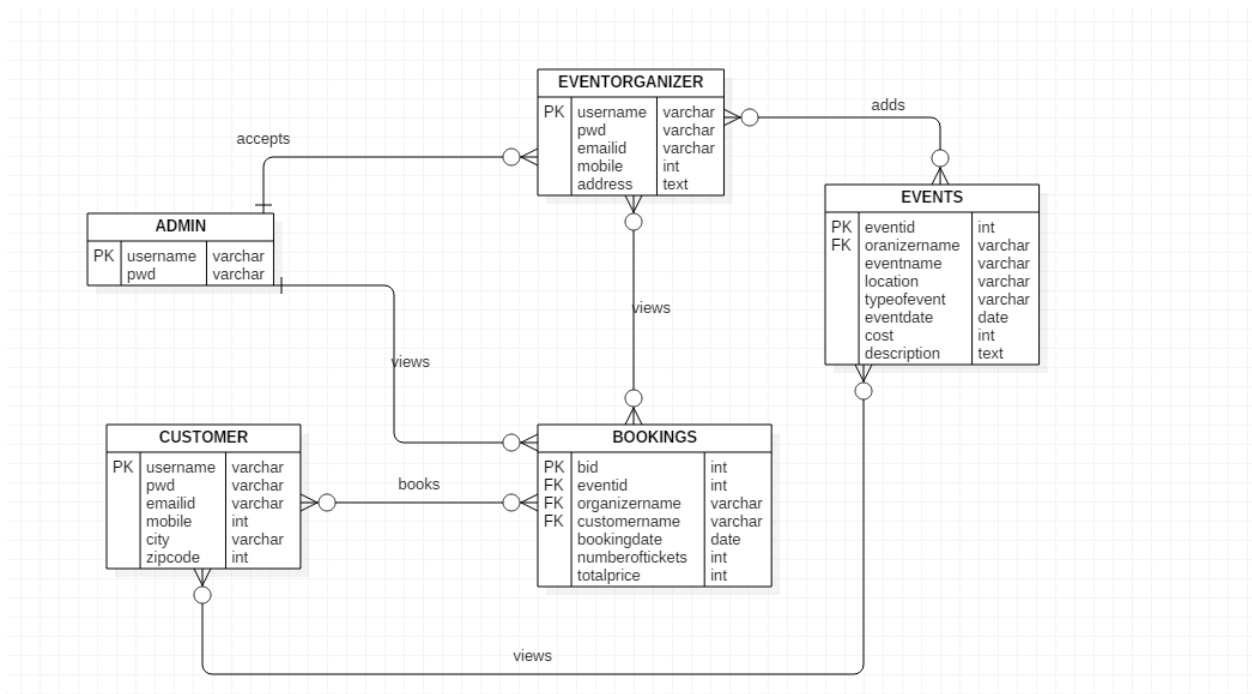


Figure10. ER-Diagram

Project Output Screen Shots

Add new events page

The screenshot shows a web form titled "Add New Event" on a dark blue background. The form contains the following fields and controls:

- Organiser Name:** Text input field with the value "event1".
- Event Title:** Text input field.
- Event Type:** Text input field.
- Location:** Text input field.
- Event Date:** Text input field with a date picker icon, showing the format "mm/dd/yyyy".
- Cost(\$):** Text input field.
- Description:** Text area.
- Submit** and **Reset** buttons at the bottom.

Figure11. Add New Events page

Admin approve event organizer events page

New Events								
EventId	Organiser Name	Event Title	Location	Type Of Event	Date	Cost	Description	Operation
332415	eventl	Mucisal Night	Texas	Entertainment	2023-11-30	50	Musical Night	Delete / Accept

Figure12. Admin approves event organizer events page

Admin view Bookings page

Event Booking Details			
EventId	Organiser Name	Date	Amount
993072	eventl	2023-10-13	100
993072	eventl	2023-10-13	100
13094	eventl	2023-10-24	10000

Figure13. Admin view bookings page

User home page

EVENT MANAGEMENT			VIEW EVENT DETAILS	VIEW PREVIOUS BOOKINGS	LOGOUT
Welcome Customer,thiru					
Recently Added Events					
EventID : 2345 OrganiserName :eventl Event Title: abcd Location :chicago Date:11-3-20023 Description:aadadsf Book Event	EventID : 7890 OrganiserName :eventl Event Title: abcd Location :chicago Date:11-3-20023 Description:aadadsf Book Event	EventID : 1000 OrganiserName :eventl Event Title: abcd Location :chicago Date:11-3-20023 Description:aadadsf Book Event			

Figure14. User Home page

User Book events page

Book Events

Event ID

Organiser Name

Customer Name

Event Date

Amount (\$)

Credit/Debit Card Details

Card Number

Expiry Date

CVV

Figure15. User book events

6 Design Units Impacts

Design units help find key areas that we need to focus on. Some of the design units implemented in this project are listed below.

- After successful login, every user different recommendations should be displayed
- Only after admin approval, events are displayed to the user
- Admin can accept or reject events
- User can book any number of events

6.1 Functional Area A/Design Unit A

6.1.1 Functional Overview

In this project development, we are focused on UI development and Model components. All data displayed by the view component are part of the front end. We use spring framework to write business logic like login page verification, registration, search event details etc.

6.1.2 Impacts

All the functional requirements specified are used for the successful flow of business functionalities and they do not impact on business logic. Our application is independent of operating system and technology.

6.1.3 Requirements

- Front-End: HTML5, CSS3, Bootstrap.
- Programming language: Java
- Operating System: Windows 10/11
- Framework: Spring MVC.
- Integrated Development Environment (IDE): Eclipse

6.2 Functional Area B/Design Unit B

6.2.1 Functional Overview

To store event details, login information of event organizer and user we have used MySQL database. To run application we have used tomcat server.

6.2.2 Impacts

To interact with the database we have used Java programming language. So our application is database-independent. Even though we replace it with other data base actual programming code does not affect and our application is server-independent, so we can deploy on any server.

6.2.3 Requirements

- Server : Apache Tomcat Web server
- Database : MySQL

7 Open Issues

To address risk management, a proper risk management team is maintained to reduce the risk impact. In the modern era, users can access applications from remote locations so they may face some risks. The main risk is to find events, fraud event organizers and payment frauds.

8 Acknowledgements

I would like to thank my professor Xchen. I also thank my mentors who helped to finish this project, guiding me to complete this project through all major decisions and believing in our capabilities.

9 References

- [1]. (2021), Bohdana Muzyka, 10 Tips on How to Improve UX/UI Design of a Web Application, <https://www.techmagic.co/blog/tips-on-how-to-improve-ui-ux/>
- [2]. (2018), Rahul Vaish, Creating a user based recommendation system in java, Apache Mahout, <https://medium.com/@rahulvaish/creating-a-user-based-recommendation-system-java-apache-mahout-6148f5e4d105>
- [3]. (2022), Alex Vilmur, What is agile web development? Everything you need to know, <https://www.marceldigital.com/blog/what-is-agile-web-development-everything-you-need-to-know>
- [4] (2022), Indeed Editorial Team, Quality Assurance in Project Management, <https://www.indeed.com/career-advice/career-development/what-is-quality-assurance-in-project-management#:~:text=What%20is%20quality%20assurance%20in,standards%20and%20produce%20quality%20deliverables.>
- [5] (2023), Common client-side web technologies, <https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-client-side-web-technologies>

10 Appendices

- 1. <https://www.baeldung.com/spring-mvc-tutorial>
- 2. <https://www.guru99.com/jsp-mvc.html>
- 3. <https://www.guru99.com/software-development-life-cycle-tutorial.html>
- 4. <https://www.mysqltutorial.org/>