Software Engineering
Software Requirements Specification (SRS) Document

[EventCenter]

[09/25/2023]

[Version 1.0]

By: [Muhammad Khan]
[Jasani Smallwood]
[Cindy Sandoval]

[Honor Code]

Table of Contents

1. Introduction

1.1. Purpose

The purpose of the Event Center application is to provide a comprehensive and user-friendly platform for event enthusiasts, vendors, and attendees. Event Center aims to simplify the event discovery process, streamline ticket purchasing, and enhance the overall event experience. It seeks to foster a vibrant event community by connecting users with a diverse range of events in their area and facilitating efficient booth management for vendors. By doing so, Event Center not only promotes convenience and efficiency but also creates a thriving ecosystem that encourages event participation, interaction, and enjoyment.

1.2. Document Conventions

The purpose of this Software Requirements Document (SRD) is to define both client-oriented and developer-oriented requirements for the Event Center application. Client-oriented requirements focus on the system's functionality and user experience, catering to the needs of event enthusiasts, vendors, and attendees. Developer-oriented requirements provide technical insights for software developers, outlining functional, data, performance, and other essential aspects required for the successful implementation of Event Center.

1.3. Definitions, Acronyms, and Abbreviations

Java	A programming language originally developed by James Gosling at Sun Microsystems. We will be using this language to build the Restaurant Manager.
PostgreSQL	Open-source relational database management system.
.HTML	ReactJS, often simply referred to as React, is an open-source JavaScript library for building user interfaces (UIs) or user interface components.
SpringBoot	An open-source Java-based framework used to create a micro Service. This will be used to create and run our application.
MVC	Model-View-Controller. This is the architectural pattern that will be used to implement our system.
Spring Web	Will be used to build our web application by using Spring MVC. This is one of the dependencies of our system.
Tailwind CSS	It is a popular open-source CSS framework that simplifies and streamlines the process of building modern, responsive, and customizable web interfaces.
IntelliJ	An integrated development environment (IDE) for Java. This is where our system will be created.
API	Application Programming Interface. This will be used to implement a function within the software where the current date and time is displayed on the homepage.

1.4. Intended Audience

Stakeholders: This section is intended for all stakeholders involved in the project, including project managers and individuals who have a vested interest in the project's success. It provides a high-level overview of the project's goals, scope, and objectives.

Developers: Sections related to technical specifications, system architecture, and implementation details are primarily intended for software developers and engineers responsible for building the Event Center application. These sections provide detailed insights into the technical aspects of the project.

Project Managers: The project manager should focus on sections related to project scope, timelines, and resource requirements. These sections help in understanding project constraints and managing project schedules effectively.

Users: Event enthusiasts, vendors, and attendees are the primary users of the Event Center application. User-oriented sections, such as functional requirements, user interfaces, and user experience, are intended to provide a clear understanding of how the application will meet their needs and expectations.

1.5. Project Scope

The goal of the Event Center software is to provide an intuitive and centralized platform for event enthusiasts, organizers, vendors, and attendees. This aligns with the overall business goals of promoting a thriving event community and enhancing the event experience.

The benefits of the project to the business include:

Efficiency: Streamlining event discovery and ticket purchase processes leads to efficient event management, reducing operational overhead.
Community Building: Fostering a sense of community and engagement among event enthusiasts and vendors contributes to a vibrant event ecosystem.
Enhanced User Experience: Providing users with a convenient platform to discover, participate in, and manage events enhances their overall satisfaction and engagement.
Increased Event Participation: The simplified process encourages more event participation, benefiting event organizers and vendors alike.

1.6. Technology Challenges

Third-Party API Integration: Integrating with third-party event data sources may pose challenges related to data consistency, format compatibility, and API stability.

Database Management: Efficiently managing a database to handle event listings, user profiles, and real-time data updates while maintaining data integrity and performance.

1.7. References

- React Documentation. (n.d.). React - A JavaScript library for building user interfaces. [Website]. https://reactjs.org/

- Tailwind CSS. (n.d.). Tailwind CSS - A utility-first CSS framework for rapidly building custom user interfaces. [Website]. https://tailwindcss.com/
- Spring Boot. (n.d.). Spring Boot - Build fast, run fast, and more. [Website]. https://spring.io/projects/spring-boot

2. General Description

2.1. Product Perspective

Event Center finds its origins in the shared desire of event enthusiasts, organizers, and vendors for a streamlined and user-friendly platform to connect and enhance the event experience. The idea was conceived by individuals with a passion for events, intending to create a valuable solution for the event community by aligning with the needs and expectations of event-goers and organizers. This project is driven by the community, for the community.

2.2. Product Features

The Event Center application offers a range of features catering to different user roles:

User Authentication: Users, including event attendees and vendors, can log in securely. Administrators have the ability to manage user accounts.

Event Discovery: Users can search for events. Event listings provide details such as event name, date, venue, and ticket price.

Ticket Purchase: Attendees can purchase event tickets.

Vendor Booth Management: Vendors and exhibitors can manage their booths within specific events, including booth booking, inventory management.

Event Creation: The application fetches events from third party API.

Administrator Functions: Administrators have the ability to view and manage user accounts and events.

2.3. User Class and Characteristics

Event Attendees

Characteristics: Event attendees are individuals interested in discovering and participating in various events. They may range from casual event-goers to frequent attendees.
Skills/Knowledge: Users in this class are expected to have basic knowledge of using a web or mobile application for event discovery and ticket purchase.

Vendors/Exhibitors

Characteristics: Vendors and exhibitors are businesses or individuals who wish to promote their products or services by booking booths at events.
Skills/Knowledge: Users in this class should understand booth management and be familiar with digital platforms for booth booking.

Administrators

Characteristics: Administrators are responsible for managing the Event Center platform. They ensure user accounts, events, and vendor activities run smoothly.
Skills/Knowledge: Users in this class should possess administrative skills and familiarity with platform management, including user account management.

General Users

Characteristics: General users may include anyone who visits the platform without registering or logging in. They can browse events but have limited access to certain features.
Skills/Knowledge: No specific skills or knowledge are required for this class as they have restricted access.

2.4. Operating Environment

The Event Center application is designed to operate in a web-based environment, ensuring accessibility and compatibility across various devices and platforms.

2.5. Constraints

Data Accuracy: To minimize user errors when entering location information, Event Center implements a drop-down selection for country, state, and city. This constraint ensures accurate location data but may require continuous updates to maintain data accuracy.

2.6. Assumptions and Dependencies

The development and operation of the Event Center application rely on certain assumptions and external dependencies:

Spring Boot Framework: It is assumed that the Spring Boot framework will be used to develop the backend of the application, facilitating the creation of RESTful APIs, security features, and database interactions.

React.js Library: The frontend of the application is developed using the React.js library, allowing for the creation of dynamic and interactive user interfaces.

PostgreSQL Database: The application is dependent on a PostgreSQL database for storing user data, event information, and other relevant data.

Internet Connectivity: Users are assumed to have a stable internet connection to access and use the Event Center application effectively.

Browser Compatibility: The application is designed to be compatible with modern web browsers, and it is assumed that users will access the platform using one of these browsers.

Third-Party APIs: The application may depend on third-party APIs for event data retrieval and location services, assuming continued API availability and compatibility.

3. Functional Requirements

3.1. Primary

FR0: Event Lookup

- The system shall allow users to search for events based on criteria such as location, date, category, and user interests.
- The system shall provide detailed event information, including event dates, venues, ticket prices, and event descriptions.

FR1: Event Creation

- The system shall facilitate event creation by allowing event organizers to input event details such as title, description, location, date, and category through a user-friendly interface.
- Event creation shall involve a third-party API call to retrieve and persist event data from the JSON response.

FR2: Ticket Issuance

- The system shall enable authorized users, such as event organizers or administrators, to generate event tickets electronically.
- Generated tickets shall include essential event information, such as event name, date, venue, pricing**,** and unique ticket identifiers

Secondary

Password Protection and Authorization

- User accounts shall be password-protected to ensure access only to authorized individuals.
- An authorization scheme shall be implemented to restrict access to specific data.

4.1. Operating System and Compatibility

The application will be compatible with any operating system that is able to view and to interact with traditional web pages.

4.2. Interface Requirements

4.2.1. User Interfaces

The Event Center application will feature a user-friendly and intuitive user interface (UI) designed to provide a seamless and engaging user experience. The following elements and guidelines will be incorporated into the UI:

Homepage:

- The homepage will serve as the central hub for users to discover and explore events.
- It will feature a search bar for event lookup based on location, date, category, and user interests.
- A "Browse Events" button will direct users to a comprehensive event listing page.

Event Listing Page:

- Event listings will include event titles, dates, venues, and ticket prices.
- Each event card will provide an option to view event details.
- Users can filter and sort event listings based on various criteria.

Event Details Page:

- Event details pages will display comprehensive information about individual events, including descriptions, start and end dates, location and price.
- Users can purchase event tickets directly from this page.

Booth Management (Vendors):

- Vendors and exhibitors will have access to a booth management interface.
- This interface will enable booth booking.

Authentication and Authorization:

Role-based authorization will ensure that users see only the features and data relevant to their roles.

4.2.2. Hardware Interfaces

The web application will run on any hardware device that has access to the internet, the ability to display webpages, and the ability to interact with web pages. This includes, but is not limited to, smartphones, tablets, desktop computers, and laptops.

4.2.3. Communications Interfaces

It must be able to connect to the internet as well as the local database on pgAdmin.
The communication protocol, HTTP, must be able to connect to the Third party API and return the correct response.

4.2.4. Software Interfaces

We will use React and Spring Boot to help build the frontend, as well as JPA for the backend database functionality. We will also use Spring Boot with Java to connect the frontend to the backend.

5. Non-Functional Requirements

5.1. Performance Requirements

- NFR0(R): The local copy of the event center database will consume less than 500 MB of memory
- NFR1(R): The system (including the local copy of the event center database) will consume less than 1GB of memory
- NFR2(R): The novice user will be able to purchase a ticket in less than 5 minutes.
- NFR3(R): The expert user will be able to purchase a ticket in less than 2 minute.

5.2. Safety Requirements

Access Control:

- Role-based access control will be implemented to restrict unauthorized access to sensitive data and functionalities.
- Users will only have access to the features and data relevant to their roles.

Secure Third-Party Integration:

- Third-party APIs and integrations will be selected based on their security measures and compliance with industry standards.

5.3. Security Requirements

- The vendor and admin dashboards will only be usable by authorized users.

5.4. Software Quality Attributes

The Event Center application aims to meet various software quality attributes to ensure a positive user experience and efficient system operation.

5.4.1. Availability

Details: The system shall strive for high availability, aiming for minimal downtime and service interruptions.

5.4.2. Correctness

Details: The system shall produce accurate and reliable results in event listings, ticketing, and user interactions.

5.4.3. Maintainability

Details: The system's codebase shall be organized, well-documented, and modular to facilitate future updates and maintenance.

5.4.4. Reusability

Details: Code components and modules shall be designed for reusability to minimize redundancy and enhance development efficiency.

5.4.5. Portability

Details: The application shall be designed to be portable across different environments and devices.

5.5. Process Requirements

5.5.1. Development Process Used

The Event Center application will follow the Agile Scrum development methodology. Key characteristics include:

Sprints: Time-bound iterations for incremental development.
Cross-Functional Teams: Collaborative teams responsible for sprint goals.
Product Backlog: Prioritized list of requirements.
Sprint Planning: Selection of backlog items and goal definition.
Daily Standups: Daily progress and planning meetings.
Sprint Review: Showcase completed features and gather feedback.
Sprint Retrospective: Reflect on the sprint and make process improvements.
Continuous Integration and Testing: Regular integration and automated testing.
User Involvement: Encouragement of user feedback and involvement.
Adaptability: Flexibility to accommodate changing requirements.

5.5.2. Time Constraints

The development of the Event Center application is subject to time constraints to ensure timely delivery and alignment with project milestones. A detailed project timeline will be established during project planning, including sprint durations and release schedules. The time constraints will be communicated to the development team to maintain a steady development pace and meet project deadlines.

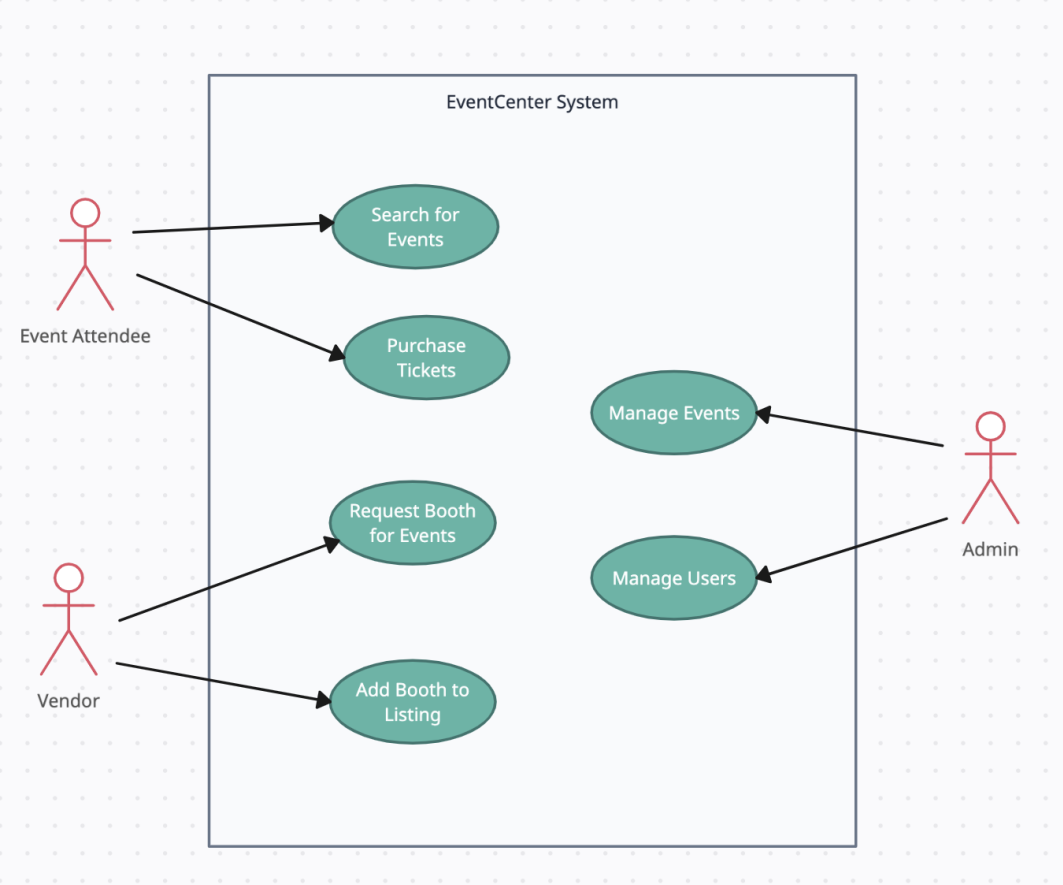
5.5.3. Cost and Delivery Date

The budget and cost constraints for the Event Center project will be outlined in the project plan. The project's financial resources will be allocated to cover development, infrastructure, third-party services, and other associated costs. A delivery date will also be established, taking into account the project scope, resource availability, and project priorities. Meeting cost and delivery date expectations will be a key project objective, and the project management team will closely monitor progress to ensure adherence to these constraints.

5.6. Other Requirements

TBD

5.7. Use-Case Model Diagram



5.8. Use-Case Model Descriptions

5.8.1. Actor: Event Attendee (Jasani Smallwood)

- **Search for Events**: Search for Events going on around the Campus
- **Purchase Tickets**: Buying tickets for events.

5.8.2. Actor: Vendor (Cindy Sandoval)

- **Request Booth for Events**: For vendors to request a booth at the event.
- **Add Booth to Listing**: Adding a vendor booth to a Event.

5.8.3. Actor: Admin (Muhammad Khan)

- **Manage Events**: Manage what events show up on the EventCenter app
- **Manage Users**: Manage the User that register with a login and password.

5.9. Use-Case Model Scenarios

5.9.1. Actor: Event Attendee (Jasani Smallwood)

- **Use-Case Name**: Search for Events
    - **Initial Assumption**:
    - **Normal**:
    - **What Can Go Wrong**
    - **Other Activities**:
    - **System State on Completion**:

- **Use-Case Name**: Purchase Tickets
    - **Initial Assumption**:
    - **Normal**:
    - **What Can Go Wrong**:
    - **Other Activities**:
    - **System State on Completion**:

5.9.2. Actor: Vendor (Cindy Sandoval)

- **Use-Case Name**: Request Booth for Events
    - **Initial Assumption**:
    - **Normal**:
    - **What Can Go Wrong**:
    - **Other Activities**:
    - **System State on Completion**:
- **Use-Case Name**: Add Booth to Listing
    - **Initial Assumption**:
    - **Normal**:
    - **What Can Go Wrong**:
    - **Other Activities**:
    - **System State on Completion**:

5.9.3. Actor: Admin (Muhammad Khan)

- **Use-Case Name**: Manage Events
    - **Initial Assumption**:
    - **Normal**:
    - **What Can Go Wrong**:
    - **Other Activities**:
    - **System State on Completion**:
- **Use-Case Name**: Manage Users
    - **Initial Assumption**:
    - **Normal**:
    - **What Can Go Wrong**:
    - **Other Activities**:
    - **System State on Completion**:

6. Design Documents

6.1. Software Architecture

The Event Center application is designed to simplify the event discovery process, streamline ticket purchasing, and enhance the overall event experience. It fosters a vibrant event community by connecting users with a diverse range of events and facilitating efficient booth management for vendors.

- Vendors: Users offering event booths.
- Attendees: Users purchasing event tickets.
- Admins: Managing the Events & Users.

The System employs Three-Tier Software Architecture, consisting of:

- Client Interface: Web and mobile interfaces for event discovery and management.
- Application Server: Business logic, user management, and event management.
- Database Server: Data storage for events, user profiles, and vendor information.

Key Components

- User Interface: Provides the user-facing web applications.
- Event Management: Manages event listings.
- Ticket Purchasing: Facilitates ticket purchasing.
- Vendor Management: Facilitates booth registration and management.
- Database: Stores event data, user information, and vendor details.

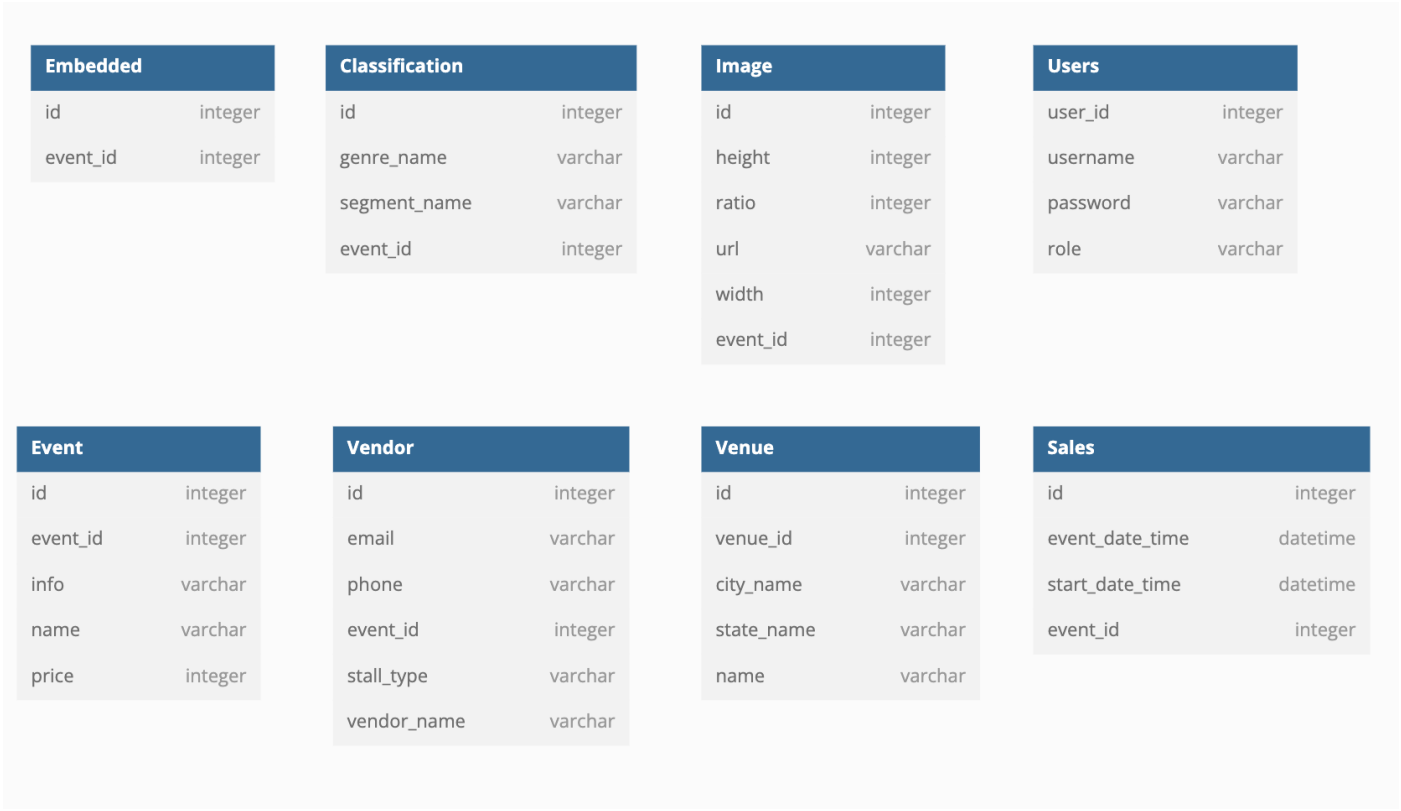
Communication and Interfaces

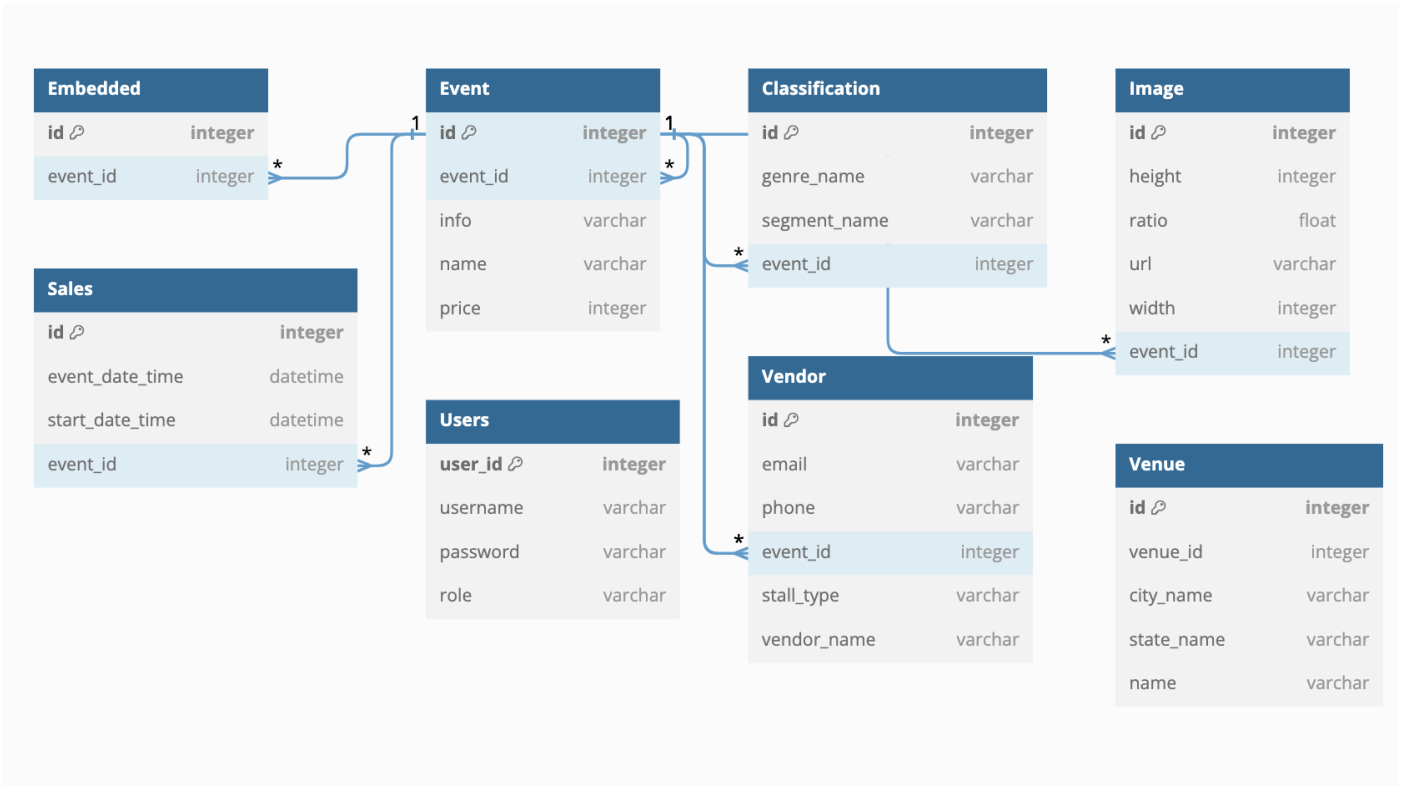
- The User Interface communicates with the Application Server through RESTful APIs. Application Server interfaces with the Database Server via JPA and communicates with third-party integrations.

Data Storage and Management

- Event data, user profiles, and vendor information are stored in a relational database. User data is securely managed with hashing and encryption.

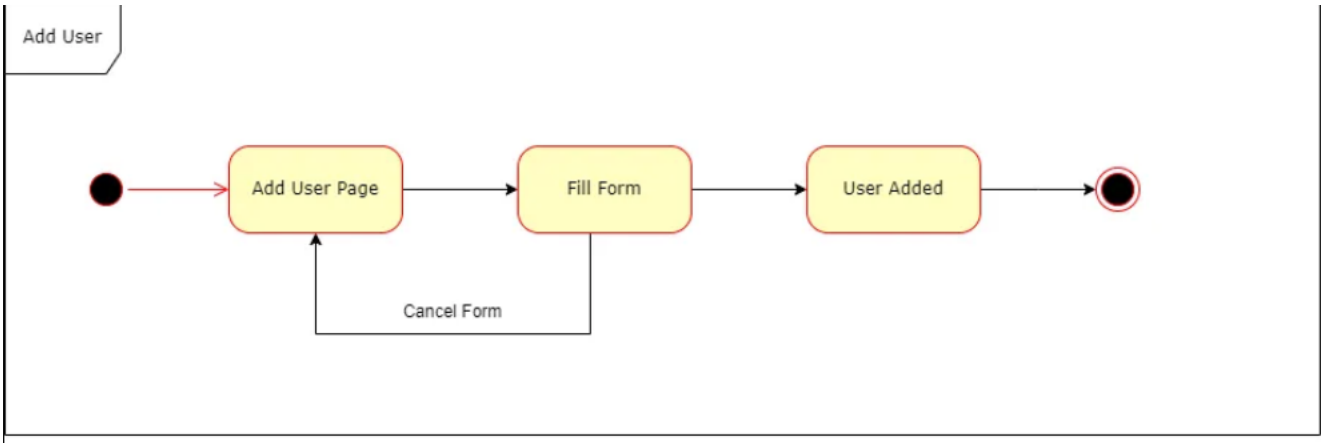
6.2. High-Level Database Schema

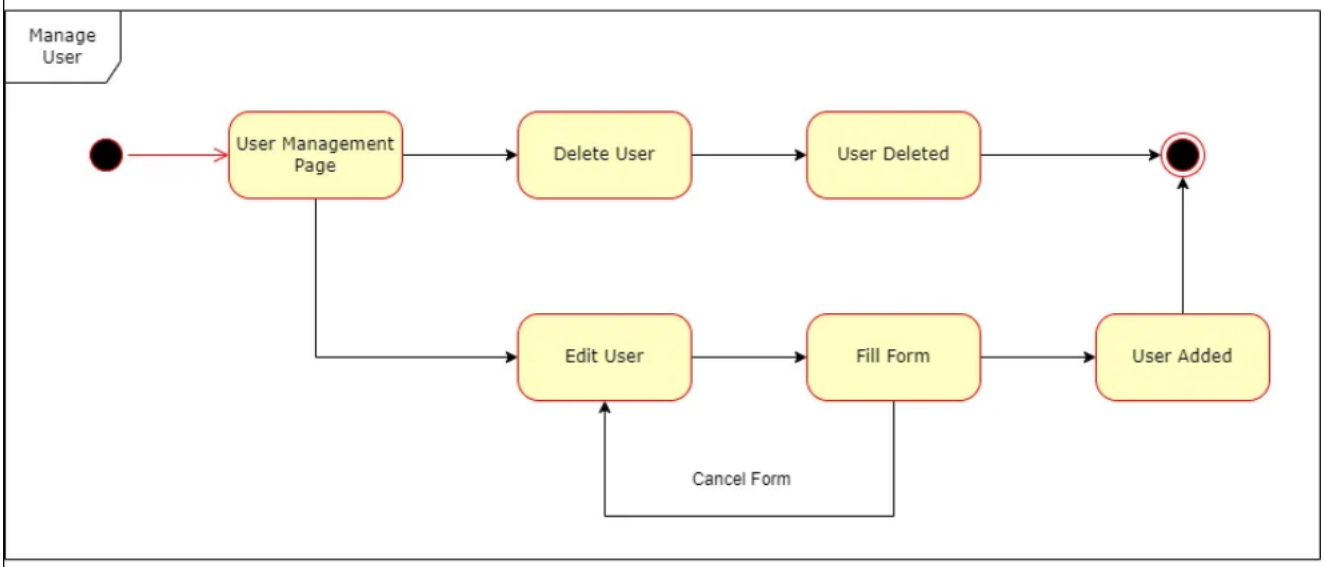


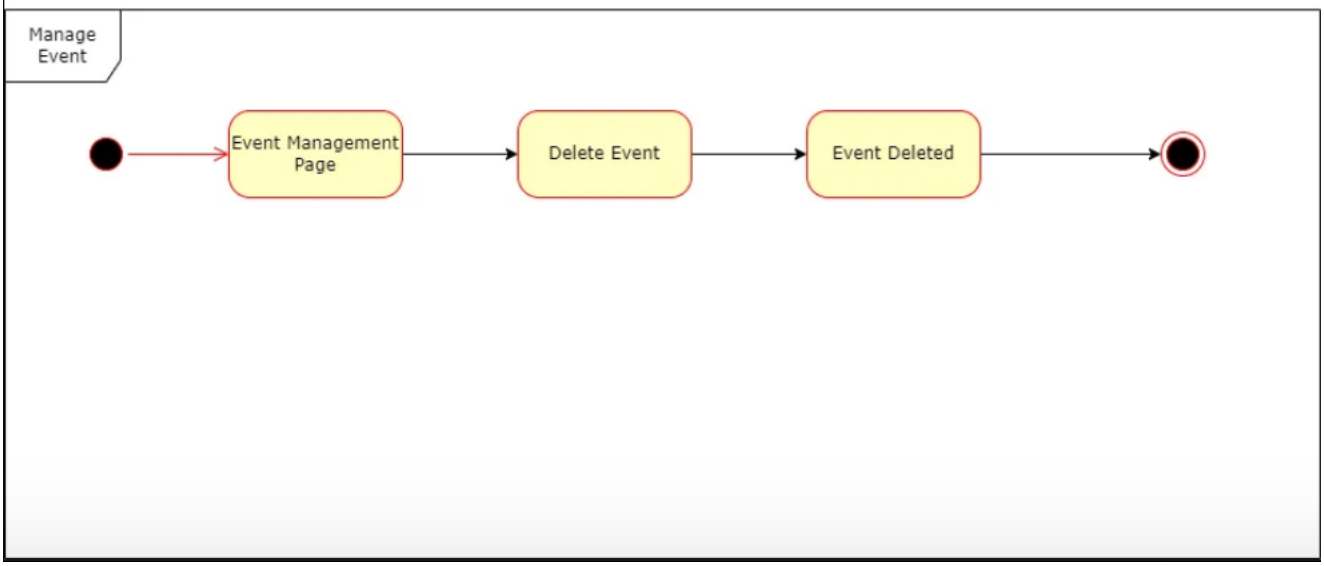


6.3. Software Design

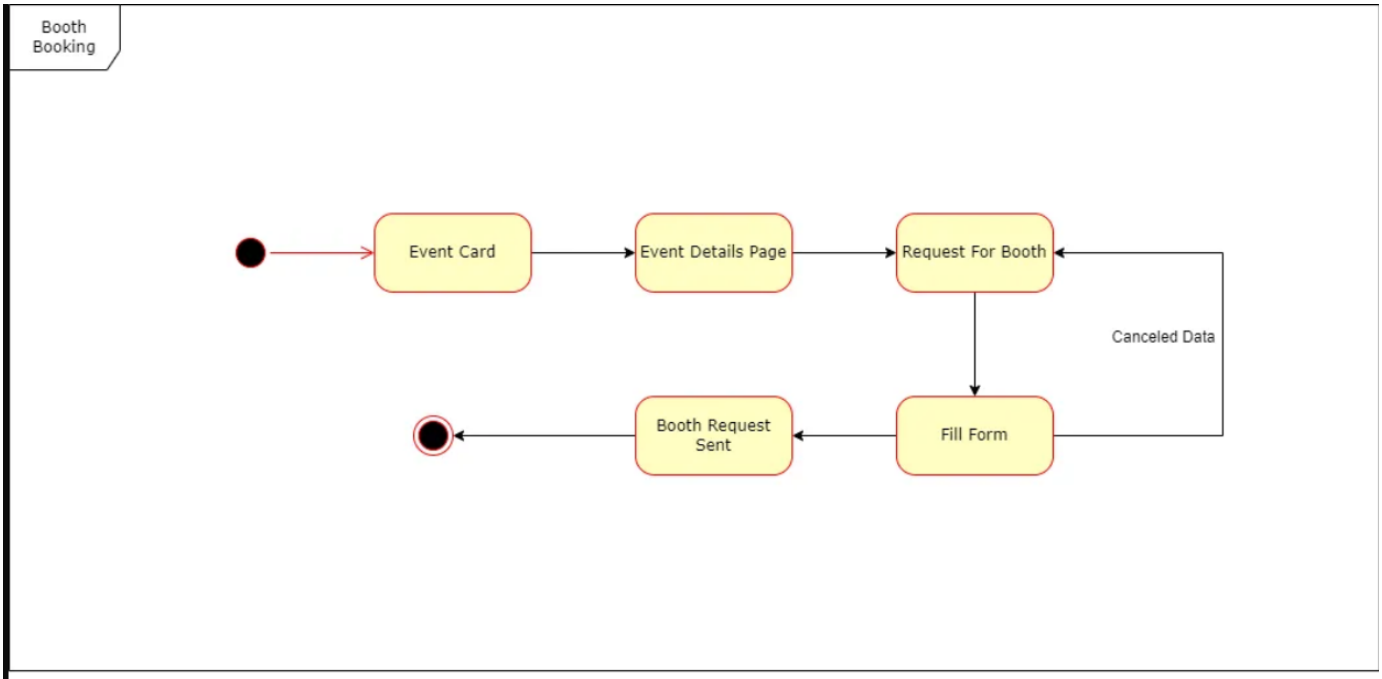
6.3.1. State Machine Diagram: Actor Name (Responsible Team Member)

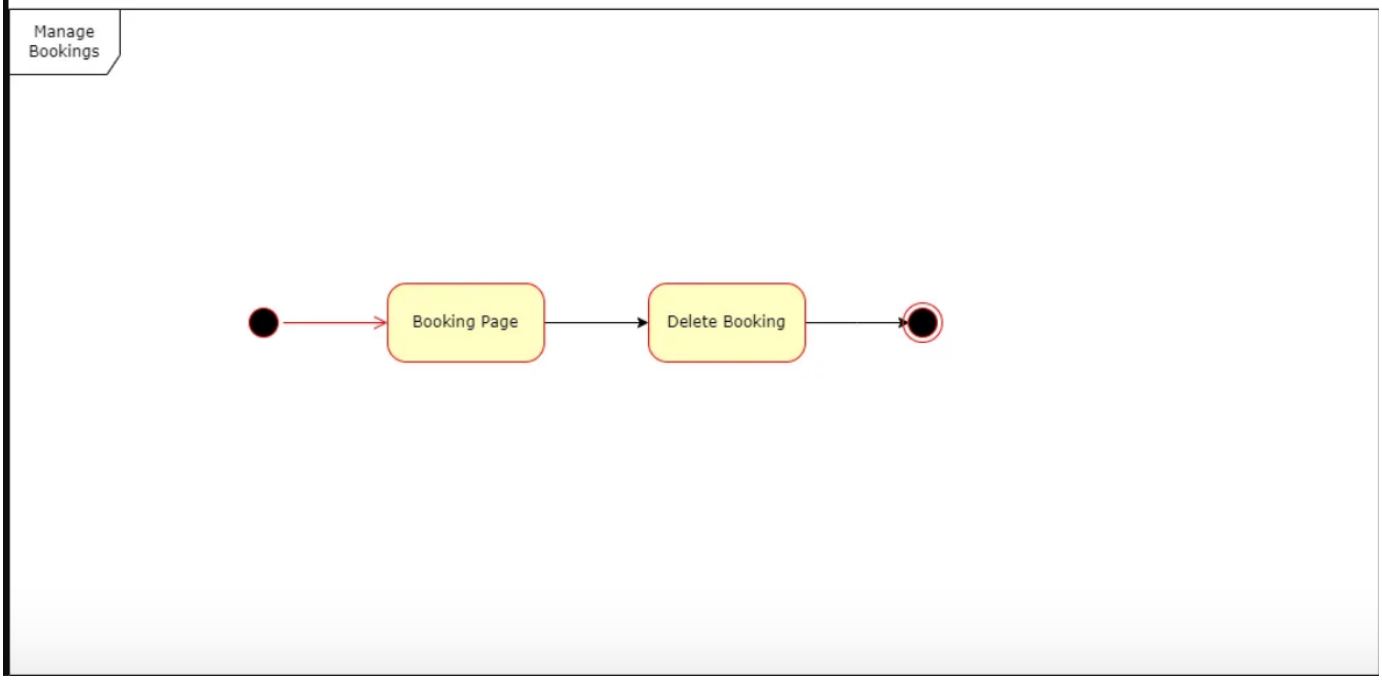




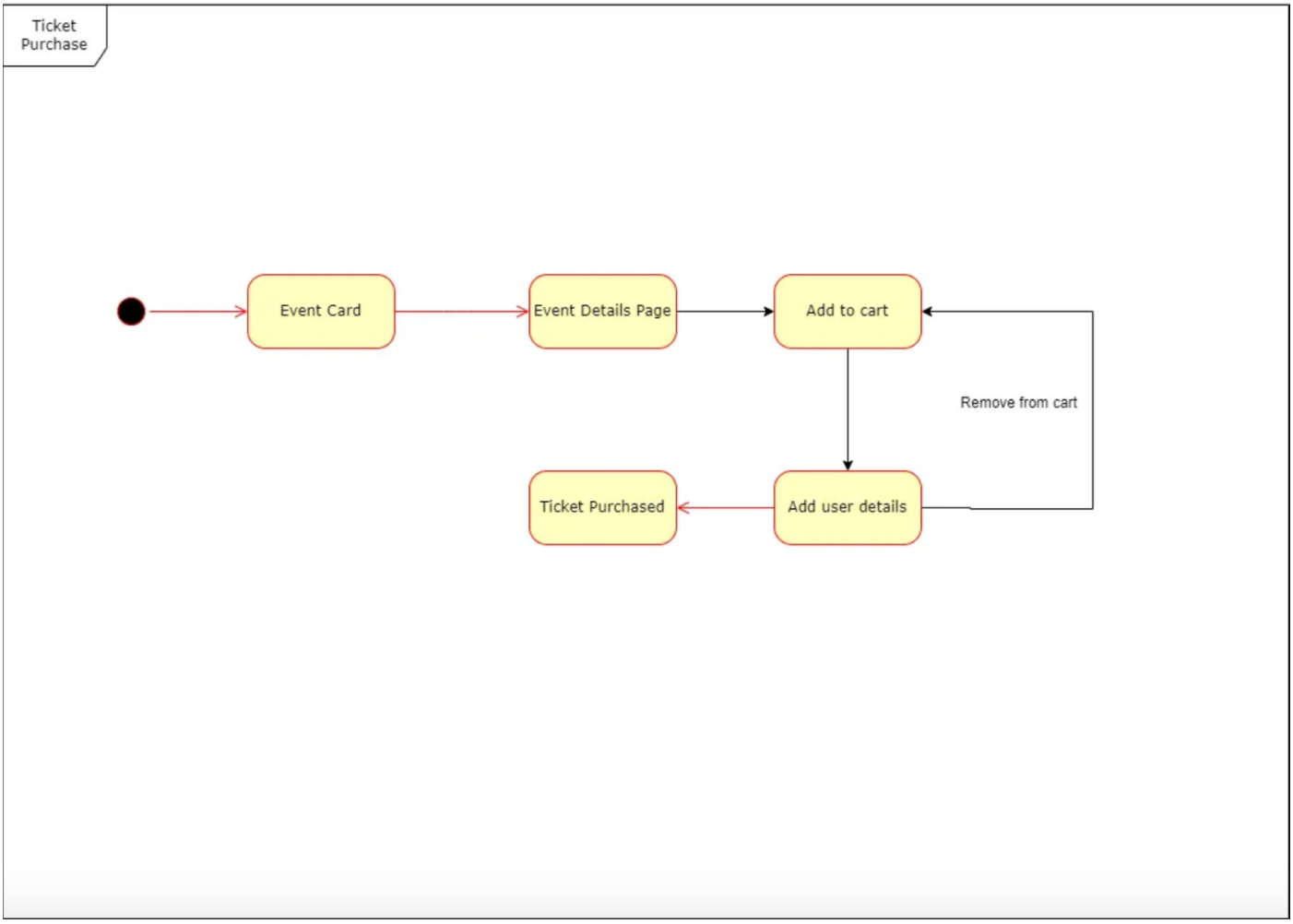


6.3.2. State Machine Diagram: Actor Name (Responsible Team Member)

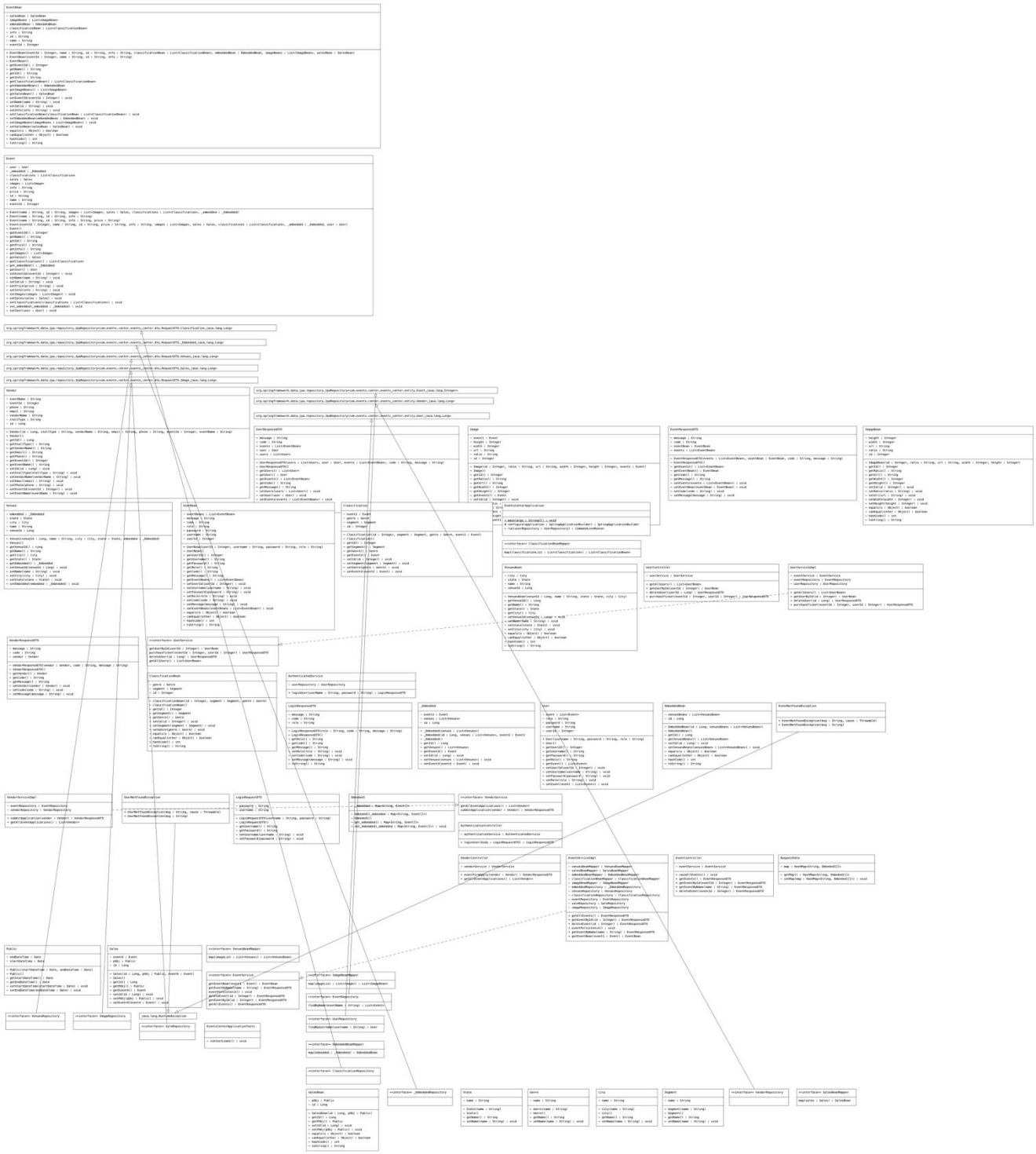




6.3.3. State Machine Diagram: Actor Name (Responsible Team Member)



6.4. UML Class Diagram



7. Scenario

7.1. Brief Written Scenario with Screenshots