**National University of Computer and Emerging Sciences**


# Event Organization System

**FYP Team**

Ammar Ahmad……...19L-2271
Ali Aamir……19L-0926
Muneeb-ur-Rehman……17L-4009

**Supervised by**

**Muhammad Ishaq Raza**

**FAST School of Computing**

**National University of Computer and Emerging Sciences**

**Lahore, Pakistan**

**October 2022**

# Abstract

The general population does not have extensive background checks to get an event suitable to their budget and regional requirements. Knowing the market gap, users would often go through all the trouble of reaching out to all the available event vendors to make an informed decision before arranging all sort of event [3]. project aims to connect the general population with all the available event vendors which allows the users to have all the information at their disposal to have the optimal event accommodating their budget and regional needs. The unsung vendors will get a chance to get traction through delivering arrangement of quality events resulting in their review and ratings to go up, making them pop-up in many users' feeds. Moreover, event organization system will allow the users to bid the event cost with the event vendors and vice versa, to attain a mutually agreeable price for the event. In short event organization system facilitates both the user and event vendor by improving connectivity among them. It is a one stop platform for booking of all events.

# Executive Summary

In Pakistan, booking events is a tenuous process. The general population does not have time, resources, and a vast outreach to all available event vendors in their region. Thus, limiting their options for getting a perfect event arranged and forces them out of their budget because of lack of available options. On the other hand, majority of the vendors have a small-scale business that is overshadowed by huge conglomerates. Their business suffers because of the lack of outreach and resources available to a small-scale event vendor.

The problem both the general population and the event vendors face is the lack of outreach and ease of information available at their available. Quantifiably, what lacks is a unified platform that facilitated both the user and the vendors by providing all the required information and set up in order to ensure smooth sailing of event booking. Aware of the market gap, the project is directed towards connecting the general population and the event vendors on a single platform to carry out all bookings related to any sort of event through an online medium.

The goal of Event organization system is to revolutionize the procedure of event arrangements and booking. Instead of making the general population go through all the trouble of reaching out to every vendor they have head of to get the venue listings, available date, available venues, prices for booking venues and availability of special arrangements within the event to be arranged, they are provided will all the information one click away. Not only that, but the users also get a list of all the event vendors available in their locality. For an average user having such a huge amount of information gathered via traditional means is practically impossible. For vendors, event organization system provides an easy means to get their business model upgraded from hosting, maintaining, and affirming bookings and user registrations through physical mean, to a completely online web-based platform. The unsung event vendors would get the opportunity to get their business out into the internet without having to spend fortunes on creating a website of their own. Furthermore, the vendors would get the chance to get their history records store to gain credibility and traction among the users to get their business trajectory uphill. In short Event Organization a user-friendly platform that facilitates both the general population and the event vendors, resulting in ease for both the users and the general population.

The project will be executed through a MERN-Stack web-based application [2]. The frontend of the web application will heavily rely on ReactJS, and the backend will be based on NodeJS. ExpressJS will be used to integrate the frontend with the backend. Data of the object instances and the users will be maintained using MongoDB. Moreover, a user will be able to use the web application on any browser that supports HTML 5.0 and above.

# Contents

# List of Tables

# List of Figures

# Chapter 1: Introduction

This section covers the introduction of the project Event Organization System, which includes the purpose, the intended audience, and the definitions, acronyms, and abbreviation of different words in this document.

## 1.1 Purpose of this Document

The purpose of this document is to completely establish the project vision by identifying the problem domain, statement, and elaborating on it. It also specifies the project's scope, restrictions, and the commercial prospects it will give. Moving on, it offers a full explanation of what the Event Organization System application will achieve by defining and describing the system's functional and non-functional requirements, as well as exhaustively explaining the use cases that the Event Organization System application's primary actors will fulfil. Furthermore, the goal of this article is to provide its readers a clear knowledge and concept of what Event Organization System does for its users, as well as who the system's core users will be and how this application will help them. Aside from that, the general architecture of the web application will be illustrated, and ultimately, the primary system requirements that must be satisfied in order for the Event Organization System to work, as well as any risks involved, will be emphasized.

## 1.2 Intended Audience

Although this document is directed towards different kinds of audiences, but the primary audience is the university’s project evaluation committee. Other intended audiences of this document is the future academic people and the develops who wants to conduct a research in organizing events through a web application. Moreover, it can also entertain the people having interest in understanding the goals of Event Organization System.

## 1.3 Definitions, Acronyms, and Abbreviations

List all important definitions, the acronyms and abbreviations used in this document.

**MERN:** MongoDB, ExpressJS, ReactJS, NodeJS

**SDG:** Sustainable Development Goal

# Chapter 2: Project Vision

This section provides the complete vision we have for Event Organization System. It includes the problem that exists, the solution our project provides, its goals and objectives, the project scope, and its constraints.

## 2.1 Problem Domain Overview

Booking events is the gruesome task for the general population given their lack of resources and connections, they tend to often overcompensate either on their budget or give up on ideal arrangements to get the event hosted at the desired time. The aim of the Event Organization System is to bridge the gap between the general population and the event vendors by gathering all the information at one platform. Event Organization System when developed, will give the event vendors an opportunity to showcase their arrangements and a client to get an event booked in their respective budget. Moreover, it will allow both the client and vendor to negotiate the budget for organizing through a bidding system.

## 2.2 Problem Statement

Living in 2022 time and resources are of the essence. Booking an event is a headache for the general population because they have outreach to a limited number of vendors, furthermore they must reach out to all the event vendors manually either through cell or direct contact with the representative. On the other hand, the unsung vendors are overshadowed by the huge business conglomerates.

## 2.3 Problem Elaboration

The process of event organization is a laborious one. The vast majority of people do not have enough connections with the companies that are providing the event's services. Because of this gap in knowledge and connectivity, it is difficult to organize any kind of event. The public is typically required to go through the hoops of getting on board with all of the event vendors by visiting them and conducting in-depth research in order to obtain information regarding the location of all of the available vendors, the level of quality of the events that are hosted, and the amount of money that is required for the arrangement of an event. Still many undiscovered vendors remain out there, forcing customers to choose from among the only few options that have been investigated.

Many small private event vendors lack the means to advertise their products and are overshadowed by large event organizing corporations. Recognizing the market need, our web-based application intends to assist the public by offering all event vendor information in a single click. Every user has a list of vendors, a budget, an evaluation of the events, and their availability.

## 2.4 Goals and Objectives

The primary goal of this project is to improve the way events are being booked by connecting the general population and the vendors on a single platform. Furthermore, it aims to provide all the information on a single platform for people to get a place booked for an event by negotiating the budget. Some other objectives include:

- To provide the user with a list of all the event vendors along with their extensive information

- To make event organization easier and accessible
- To eliminate discrimination between huge market prospects and newly started ones
- To provide a platform for reviewing the vendor's service and arrangements for future references
- To help the newly launched event organizations to get their name out into the public without having to spend enormous amount on marketing

## 2.5 Project Scope

The scope of this project is to provide a MERN-Stack based web application to connect the users with vendors and vice versa through a single platform, It will provide the user the list of all the vendors in a certain region, venues associated with vendors, their quotations, ranked by the user reviews, which is adjustable to a user's personal criteria. On the other hand, event vendors register their business and get a platform to get it out in the open. Moreover, vendors can get their venues through an online platform.

## 2.6 Sustainable Development Goal (SDG)

### Industry, Innovation and Infrastructure

Event organization system aims to revolutionize industry infrastructure by providing an online medium that will allow the users to book events of any sort through an online medium instead of resorting to the conventional physical medium of booking events.


**Figure 1: Sustainable Development Goals**

*This figure represents all the SDG's that can be target of a FYP*

## 2.7 Constraints

Following are the constraints that our project ca

### 2.7.1 Validation Constraint

To use this web application, an event vendor is required to register by providing sufficient information as a prove of his business. Furthermore, a user will need to register an account

accompanied by a payment method to book an event. However, a viewer can perform the action of window shopping without credentials.

### 2.7.2 End-User Constraint

The user will be required to have up-to-date browsers to support this MERN-Stack application. Moreover, a user requires a reliable internet connection to use our web application.

### 2.7.3 Data Repository Constraint

The backend of web-based application is supported by a cloud-based storage, i.e., MongoDB. The rights to alter the data residing in the maintained database, to ensure the correct alteration, insertion, and deletion of the data, is restricted only to the admin.

### 2.7.4 Interoperability Constraint

Client and server both have to agree upon a unified methodology of storing and sharing the data.

### 2.7.5 Network Communication Constraint

HTTP web protocol is used to send and receive data between client and server.

### 2.7.6 Performance Constraint

A user will be required to have a high-speed internet connection to navigate throughout the pages seamlessly performing various functionalities.

### 2.7.7 Language Constraint

The user is required to have a proper understanding of English language to use our web application.

## 2.8 Business Opportunity

In Pakistan, there are existing web applications that provide the list of event vendors along with their information. However, none of them allows the user to book an event. Knowing this market gap, the business in this scenario is to provide a web application where users can search, bid, and book a venue, all through the same platform.

## 2.9 Stakeholders Description/ User Characteristics

Identify and briefly describe who the users of this system will be, and their roles.

### 2.9.1 Stakeholders Summary

#### 2.9.1.1 Admin

The Event Organization System administrator will be responsible of overseeing and monitoring the application, as well as registering, deleting, and viewing event vendors.

#### 2.9.1.2 Client

The client will be the client who will utilize the service station's services. The customer will be able to search for, browse, and book events from event suppliers.

#### 2.9.1.3 Event Vendor

The event vendor is the owner of the business that maps out their services onto the web application.

## 2.9.2 Key High-Level Goals and Problems of Stakeholders

Key High-level goals for our project are:

- Business Opportunities for small vendors
- More custom oriented approach
- Ease of business
- Variety of options
- Availability to high-end venues at more reasonable price
- Economic growth

Problems that stakeholder can face are:

- Stakeholders Legitimacy
- Generational differences
- Organizational legitimacy
- Competing priority
- Risk Planning
- Fallacious Management.

# Chapter 3: Related Work

In today's world, booking event is quite hectic. People have to manually go to the event vendors to book an event, which can restrict them to book from the few options available. What is required is an online platform, which allows the user to book an event. This chapter covers the related work done in this department and what they lack.

## 3.1 Definitions, Acronyms, and Abbreviations

**Client:** General population
**Vendor**: Event organizations and vendors

## 3.2 Related Work

### 3.2.1 pakVenues.com

pakVenues.com is one of the event management platforms available for public in Pakistan. They claim to be Pakistan's largest venue search engine. They are currently providing services of photographers, decors, salons, rent cars and invitations. Interestingly they are providing services of event mangers as well for a hassle-free experience with a peace of mind. They are currently operating in Islamabad, Rawalpindi, Lahore & Karachi.

Firstly, this platform is only suitable for someone who is looking for a wedding event. Their targeted niche is all sort of weddings from lower class to all the way up to the elite class. Secondly, this platform does not allow new struggling vendors to publicize their businesses. They select vendors themselves and promote only those vendors with whom they reach at a mutual agreement. This situation also limits customers' option when using this platform for event booking. Lastly, customers are not allowed to bid their quotation to all the vendors directly, rather they get a "starting from" option which allows them to select a vendor on the bases of their budget and numbers of participants. This way customers are once again limited only to several options.

### 3.2.2 venueHub.pk

venueHub.pk is another web platform available for online event management in Pakistan. Unlike most of the event management websites which only deals in weddings, this is one of the few platforms which deal in corporate event management as well. As corporate event services, they are offering celebrity conference and seminars, management service, business dinners, product launch and incentive trips. They are currently dealing in Lahore, Faisalabad, Haripur, and Islamabad. They have newsletter service available as well which we could opt to subscribe if we are interested in latest updates regarding venues and services. They allow vendors to list their businesses for their potential customers and to increase their reach as well.

Firstly, this kind of web platform is only suitable for the elite or upper middle class. Their targeted niche is elite or at the very least upper middle-class vendors as well as customers. Secondly, they work as a bridge between vendors and customers to maximize their profits which does not allow full transparency to both parties. Lastly, whenever customers book any sort of events on this platform, they get a predefined quotation from the platform they are not allowed to bargain upon. On the other hand, vendors must agree upon the quotation provided by the platform for any sort of business coming from the website which does not allow vendors to be in fully control and it decreases their chances of maximum profit as well.

### 3.2.3 a2zEventSolutions.com

a2zEventSolution.com is an online event management service available in Pakistan. They claim to deal in all sorts of event including family events like weddings, birthday parties, get-togethers, and family parties as well as corporate events, gala nights, conferences, exhibitions, seminars etc. They are based in Lahore and offer three packages for customers based on their budget called Alpha package, Bravo package and Charlie package.

Firstly, this platform is only suitable for those customers who are looking forward to arranging an event with minimum 300 guests as described in all three of their packages. Someone who wants to arrange an event with 20 people is unable to benefit from this services or platform. Secondly, vendors are not allowed to publicize their business on this platform to maximize their reach. No information related to vendors or business listing is available on the website. Lastly, customers are not allowed to bargain or bid their quotations. If fortunately, one of their packages fall under customer's budget, they have a green signal. Otherwise, they must struggle on their own.

### 3.2.4 theEventBook.uk

theEventBook.uk is a London based web platform which deals in event consultation, event design, event management, venue finding, virtual events, hybrid events, summer parties, Christmas parties and press events etc. Their customers include large and small corporate organizations like Shell, Google, Ferrari, Paragon, and Oliver Wyman etc. They promote declutter, planning, best deals, trustworthiness, and ideas to life as their core values.

Firstly, they focus more on event experience rather than budget because the kind of organizations they handle have no issue related to capital. They work with an array of international unique event suppliers to create the perfect event to exceed your expectations and brief. Secondly, they are more of an event supervision company as compared to an event management company. Thirdly, there are no opportunities for any sort of vendors as they focus entirely on customers and then choose their vendors accordingly. Therefore, vendors cannot publicize their profiles to attract maximum customers on this platform. Last but not the least, you are promised a guaranteed best price if you are using their services but not necessarily the price of your choice.

## 3.3 Conclusion

We researched how the current platforms managed their operations before we began our work on this project. Since each of them is unique, we needed to develop a strategy that would enable us to improve the project. The project is based on creating an online platform for the clients to bid and book an event, which is not done in this area yet.

# Chapter 4: Software Requirement Specification

This section covers the features Event Organization System will have along with the functional and non-functional requirements, quality attributes of the system, assumptions, hardware, and software requirements that the web application will have, use cases for each actor, risk analysis, GUI of the web application and its database design.

## 4.1 List of Features

1. Book an event
2. Bid for an event
3. List venues for event
4. Give feedback
5. Rate a vendor
6. Sign In
7. Login/Logout
8. Register a vendor
9. Suspend a vendor
10. Delete a vendor
11. Edit Profile
12. Edit payment method
13. See vendor information
14. Search for an event venue
15. Bid on a client's budget for event
16. Budgeting filters
17. Location filters
18. Rating filters
19. Event category filters
20. Add venue
21. Delete venue
22. Alter venue's specifications
23. Map location
24. Cancel booking
25. Check availability
26. Page navigation
27. Add to favorites
28. Delete from favorites
29. Alter booking details
30. Add special arrangements for an event

## 4.2 Functional Requirements

### 4.2.1 Functional Requirements for Clients

1. System shall allow the client to sign up. Clients shall provide their name, email, passwords, phone number, address, date of birth, CNIC, and payment method. System shall save their information to its database and redirect the user to its homepage.
2. System shall allow the client to login when the/she provide the right email and password.
3. System shall allow the client to logout when he/she clicks the logout button.
4. System shall allow the client to alter his/her payment method.

5. System shall allow the client to make a payment.
6. System shall allow the client to alter his profile, email, and passwords.
7. System shall allow the client to search for an event.
8. System shall allow the client to apply the location, event category, rating, and budgeting filter while searching.
9. System shall allow the client to bid for an event venue.
10. System shall allow the client to alter the booking details.
11. System shall allow the client to add special arrangements for an event.
12. System shall allow the client to book an event.
13. System shall allow the client to add event venue/event vendor to his/her favorite list.
14. System shall allow the client to remove the event venue/event vendor from his/her favorite list.
15. System shall allow the client to give a rating to the event vendor.
16. System shall allow the client to give feedback to the event vendor.
17. System shall allow the client to cancel the booking.
18. System shall allow the client to look for the event venue's availability.

### 4.2.2 Functional Requirements for Event Vendors

- System shall allow the vendor to Sign up, by receiving their email address, password, name of business and service provided by the business through the Sing Up page.
- System shall allow the vendor to get authenticated as a certified vendor by providing proof of existing business and bank information.
- System shall allow the vendor to login based on vendor's credentials.
- System shall allow the vendor to receive payments from the clients.
- System shall allow the vendor to edit their profile.
- System shall allow the vendor to add/remove/update venue list.
- System shall allow the vendor to modify venue prices.
- System shall allow the vendor to modify venue specifications.
- System shall allow the vendor to add/remove/modify location to each venue provided by the vendor.
- System shall allow the vendor to add/delete/modify the list of special arrangements provided by the vendor.
- System shall allow the vendor to view booking requests from the users.
- System shall allow the vendor to accept/reject booking request.
- System shall allow the vendor to bid on the user's provided booking request.
- System shall allow the vendor to contact with the client on a messaging platform
- System shall allow the vendor to mark the service as complete.
- System shall allow the vendor to view the service logs.
- System shall allow the vendor to sign Out.

### 4.2.3 Functional Requirements for Admin

1. System shall allow the admin to login when the right credentials are entered.
2. System shall allow the admin to view the list of event venues and event vendors.
3. System shall allow the admin to edit his/her profile.
4. System shall allow the admin to change his/her credentials.
5. System shall allow the admin to view the list of all the event bookings.
6. System shall allow the admin to add an event vendor.
7. System shall allow the admin to delete an event vendor.

8. System shall allow the admin to suspend an event vendor.

## 4.3 Quality Attributes

#### 4.3.1.1 Maintainability

It must be simple to update and support various system versions. Easy upgradability for new features and technologies as well as code addition will be prioritized throughout development. The maintenance will be simple and cheap. It takes little effort to keep the system running well and fix bugs or update the software.

#### 4.3.1.2 Correctness

The system's computations and internal workings, as well as its navigation, will be accurate. This implies that the system will meet certain functional requirements.

#### 4.3.1.3 Efficiency

It is a crucial quality attribute of the system. If all available resources are being used by the system, then the user experience will suffer, and the system would be deemed inefficient. Without enough efficiency, the system will be useless for any kind of real-time function. So, the web application will ensure that the user will get the maximum efficiency.

#### 4.3.1.4 Testability

It will be straightforward to test the system and find bugs. It will be simple to break down into smaller pieces for testing if necessary.

#### 4.3.1.5 Flexibility

The system will be adaptable to different situations, capable of interacting with several other items. The system's integration with other common third-party tools will be straightforward.

## 4.4 Non-Functional Requirements

#### 4.4.1.1 Privacy and Security

The system will ensure that the data of both client and event vendor is in compliance with the three foundational concepts of information security, which include confidentiality, integrity, and authenticity. Making it so that the data is secured from any third-party attack. The passwords will be encrypted to protect the user data.

#### 4.4.1.2 Performance

The system efficiency will be optimal having a well-designed structure to provide an efficient navigation between the screen of the web application.

#### 4.4.1.3 Supportability

A browser having HTML 5 on a smartphone, tablet, or a PC will be able to access the system.

#### 4.4.1.4 Usability

The system will have a user-friendly intuitive interface. The contents of each page would be unambiguous along with easy navigation throughout the pages.

#### 4.4.1.5 Capacity

The system will have enough storage capacity to ensure that the users setting preferences and favorite tabs are maintained to provide a more personal experience for the user.

#### 4.4.1.6 Localization

The system will have the list of all the available event vendors localized to their geographical location.

#### 4.4.1.7 Reliability

The system will be reliable. Its reliability will be quantified based on the following factors:

##### *4.4.1.7.1 Percentage of the Probability of Failure*

Ideally, the system will have a negligible percentage of probability of failure measured by calculating the number of failures in the built divided by the overall functionalities performed.

##### *4.4.1.7.2 Number of Critical Failures*

Ideally, the system will have no critical failures quantified by tracking the number of critical failures/crashes over a predefined time interval.

## 4.5 Assumptions

- Users are assumed to have connected to a reliable internet connection.
- Each actor must have enough understanding of how to use a web application
- Users must have enough RAM and hard disk memory to use the web application.
- Users are assumed to have a basic understanding of English language.

## 4.6 Hardware and Software Requirements

### 4.6.1 Hardware Requirements

#### 4.6.1.1 Hardware Requirements During Development Phase

- Required 2 GB of RAM at the very least while 4 GB is recommended for better performance.
- Required a processor of 2.0 GHz or more
- Required a minimum of Core i3 processor, a better one is recommended for good performance.
- Required a high-speed internet connection with a minimum speed of 4 MB or more.

#### 4.6.1.2 Hardware Requirements After Deployment (For Users)

- Around 500 MB or more of a bowser space
- Required to have 1 GB of RAM at the least to run a browser
- Required a high-speed internet connection with a minimum speed of 4 MB or more.
- Required a processor of 2.0 GHz or more

### 4.6.2 Software Requirements

#### 4.6.2.1 Software Requirements During Development Phase

- Visual Studio Code

- MongoDB (v4.4.1 and above)
- NodeJS and its required libraries
- ExpressJS and its required libraries
- ReactJS and its required libraries
- Windows, Linux or macOS.
- Google Chrome, Mozilla Firefox or Microsoft Edge as web browsers.

#### 4.6.2.2 Software Requirements After Deployment (For Users)

- A smartphone, tablet, or PC with the up-to-date version of a web browser

## 4.7 Use Cases

## 4.7.1 Use Cases for Client

#### 4.7.1.1 Client Login

| | | | |
|---|---|---|---|
| **Name** | Client Login | | |
| **Actors** | Client | | |
| **Summary** | The client shall provide their email and password on the login form and after successful verification, redirect the client to the home page. | | |
| **Pre-Conditions** | The client must be in the database records either added by any of the authorized clients or added manually by a developer.<br>The client must not already be logged in. | | |
| **Post-Conditions** | The client's session is successfully established and shall be redirected to the home page. | | |
| **Special Requirements** | None | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The client opens the login page. | 2 | The login page is displayed asking for email and password. |
| 3 | The client enters valid email and password. | 4 | The system verifies the email and password, establishes a session for the user and redirects the user to the home page. |
| **Alternative Flow** | | | |
| 3 | The client enters invalid email or password. | 4-A | The system responds with an error message: *Incorrect email or password entered.* |

#### 4.7.1.2 Client Sign Up

| | |
|---|---|
| **Name** | Client Sign Up |
| **Actors** | Client |
| **Summary** | The client shall provide their name, email, date of birth, CNIC, age and password on the sign in form and after successful verification, redirect the user to the payment method page. |

| | | | |
|---|---|---|---|
| **Pre-Conditions** | The client must not be in the database records. | | |
| **Post-Conditions** | The client's session is successfully established and shall be redirected to the payment method page. | | |
| **Special Requirements** | None | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The client opens the sign in page. | 2 | The sign in page is displayed asking for email and password. |
| 3 | The client enters valid name, email, date of birth, CNIC, age and password.. | 4 | The system verifies that the name, email, date of birth, CNIC, age and password does not already exist in the database, establishes a session for the client and redirects the client to the payment method page. |
| **Alternative Flow** | | | |
| 3 | The client enters the name, email, date of birth, CNIC, age and password that already exists in the database | 4-A | The system responds with an error message: *Already a user*. |

### 4.7.1.3 Client Log Out

| | | | |
|---|---|---|---|
| **Name** | Client Log out | | |
| **Actors** | Client | | |
| **Summary** | The client shall click on the log out button and after successful verification, redirect the user to the payment method page. | | |
| **Pre-Conditions** | The client must be logged in the system. | | |
| **Post-Conditions** | The client's session is successfully deleted and shall be redirected to the login page. | | |
| **Special Requirements** | None | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The client clicks on the log out button. | 2 | The system logs out the client and redirect to the login page. |

### 4.7.1.4 Alter Payment Method

| | |
|---|---|
| **Name** | Alter Payment Method |
| **Actors** | Client |
| **Summary** | The client shall go to Edit profile page, and then click on Change Payment Method. Then, enter the card details and after the successful verification, the system will alter his/her payment method in the |

| | | | |
|---|---|---|---|
| | database and will be redirected to the Edit Profile Page. | | |
| **Pre-Conditions** | The client must be logged in. | | |
| **Post-Conditions** | The client's session is successfully altered in the database and shall be redirected to the edit profile page. | | |
| **Special Requirements** | Client shall provide the valid card details. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The client opens the Edit Profile page. | 2 | The Edit Profile page is displayed. |
| 3 | The client clicks on Change Payment Method button. | 4 | The Payment Method page is displayed asking for the card details. |
| 5 | The client enters the valid card details. | 6 | The system verifies the card details, alter his/her payment method in the database and redirect to the Edit Page with the message: *Payment Method Altered.* |
| **Alternative Flow** | | | |
| 5 | The client enters the valid card details. | 6-A | The system responds with an error message: *Card not valid. Add another payment method.* |

### 4.7.1.5 Make a Payment

| | | | |
|---|---|---|---|
| **Name** | Make a Payment | | |
| **Actors** | Client | | |
| **Summary** | The client shall click on Make a Payment, and after successful verification, the system will add the booking in the database and redirect the user to the Booking page. | | |
| **Pre-Conditions** | The client must be logged in.<br>The client must book the event first. | | |
| **Post-Conditions** | The event shall be successfully booked and stored in the database and shall be redirected to the payment method page. | | |
| **Special Requirements** | The client must have enough credit in his/her bank account | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The client opens the Bookings page. | 2 | The Bookings page is displayed showing the clients booking. |
| 3 | The client clicks on the Booking he/she wants to make a payment for... | 4 | The Event page is displayed showing the event details. |
| 5 | The client clicks on Make a Payment button | 6 | The system verifies, alters the Booking status in the database, and responds with a message: *Payment Successful* |
| **Alternative Flow** | | | |

| | | | |
|---|---|---|---|
| 5 | The client clicks on Make a Payment button | 4-A | The system responds with an error message: *Payment Unsuccessful. Recharge Your Card or Change the Payment Method.* |

### 4.7.1.6 Alter the Profile

| | | | |
|---|---|---|---|
| **Name** | Alter the Profile | | |
| **Actors** | Client | | |
| **Summary** | The client shall go to the Edit Profile page, enter the name/password/email/CNIC/date of birth and after successful verification, redirect the user to the Edit Profile page. | | |
| **Pre-Conditions** | The client must be logged in. | | |
| **Post-Conditions** | The client's profile is successfully altered and shall be redirected to the Edit Profile page. | | |
| **Special Requirements** | The entered information must be different from the previous ones. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The client opens the Edit Profile page. | 2 | The Edit Profile page is displayed showing the client's profile. |
| 3 | The client enters valid name/email/date of birth/CNIC/password. | 4 | The system verifies the information, alter it into the database, and redirects to the Edit Profile page with a message: *Successfully Changed* |
| **First Alternative Flow** | | | |
| 3 | The client enters the same name/email/date of birth/CNIC/password | 4-A | The system responds with an error message: *Unchanged Information Entered.* |
| **Second Alternative Flow** | | | |
| 3 | The client enters invalid name/email/date of birth/CNIC/password | 4-B | The system responds with an error message: *Invalid Information Entered.* |

### 4.7.1.7 Search for an Event

| | |
|---|---|
| **Name** | Search for an Event |
| **Actors** | Client |
| **Summary** | The client shall provide the event vendor/event venue name and after successful verification, the system will provide the list of event venues and event vendors related to the search information. |
| **Pre-Conditions** | The client must be logged in. |
| **Post-Conditions** | The system will provide the list of event venues and event vendors related to the search information. |

| **Special Requirements** | None | | |
|---|---|---|---|
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The client enters the event venue or event vendor name. | 2 | The system will look for the related names and provide the suggestions. |
| 3 | The client clicks on the Search button. | 4 | The system will get the information from the database and provide the list of the event vendors and event venues related to the client's search. |
| **Alternative Flow** | | | |
| | | 4-A | If the system finds nothing related to the search, it will respond with the message: *Oops! Nothing matched your search.* |

### 4.7.1.8 Apply Location Filter

| **Name** | Apply Location Filter | | |
|---|---|---|---|
| **Actors** | Client | | |
| **Summary** | The client shall apply the location filter and after successful verification, the system will provide the list of event venues and event vendors related to the search information. | | |
| **Pre-Conditions** | The client must be logged in. | | |
| **Post-Conditions** | The system will provide the list of event venues and event vendors in the selected area. | | |
| **Special Requirements** | None. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The client select the location under the location tab. | 2 | The system will look for the area name and display it in the tab. |
| 3 | The client clicks on the Search button. | 4 | The system will get the information from the database and provide the list of the event vendors and event venues in the selected area. |
| **Alternative Flow** | | | |
| | | 4-A | If the system finds nothing related to the search, it will respond with the message: *Oops! Nothing matched your search.* |

### 4.7.1.9 Apply Event Category Filter

| **Name** | Apply Event Category Filter |
|---|---|
| **Actors** | Client |

| | |
|---|---|
| **Summary** | The client shall apply the event category filter and after successful verification, the system will provide the list of event venues and event vendors related to the search information. |
| **Pre-Conditions** | The client must be logged in. |
| **Post-Conditions** | The system will provide the list of event venues and event vendors in the selected area. |
| **Special Requirements** | None. |

| **Basic Flow** | | | |
|---|---|---|---|
| **Actor Action** | | **System Response** | |
| 1 | The client select the event category under the event category tab. | 2 | The system will look for the event categories it covers, and display them on the screen. |
| 3 | The client clicks on the Search button. | 4 | The system will get the information from the database and provide the list of the event vendors and event venues of the that specific event category. |
| **Alternative Flow** | | | |
| | | 4-A | If the system finds nothing related to the search, it will respond with the message: *Oops! Nothing matched your search*. |

### 4.7.1.10 Apply Rating Filter

| | |
|---|---|
| **Name** | Apply Rating Filter |
| **Actors** | Client |
| **Summary** | The client shall apply the rating filter and after successful verification, the system will provide the list of event venues and event vendors related to the search information. |
| **Pre-Conditions** | The client must be logged in. |
| **Post-Conditions** | The system will provide the list of event venues and event vendors in the selected area. |
| **Special Requirements** | None. |

| **Basic Flow** | | | |
|---|---|---|---|
| **Actor Action** | | **System Response** | |
| 1 | The client select the rating under the rating tab. | 2 | The system will look for the event vendors and venues with that particular rating |
| 3 | The client clicks on the Search button. | 4 | The system will get the information from the database and provide the list of the event vendors and event venues of the that specific rating. |
| **Alternative Flow** | | | |

| | | | |
|---|---|---|---|
| | | 4-A | If the system finds nothing related to the search, it will respond with the message: *Oops! Nothing matched your search*. |

### 4.7.1.11 Apply Budgeting Filter

| **Name** | Apply Budgeting Filter | | |
|---|---|---|---|
| **Actors** | Client | | |
| **Summary** | The client shall apply the budgeting filter and after successful verification, the system will provide the list of event venues and event vendors related to the search information. | | |
| **Pre-Conditions** | The client must be logged in. | | |
| **Post-Conditions** | The system will provide the list of event venues and event vendors in the selected area. | | |
| **Special Requirements** | None. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The client specifies his/her under the budget tab. | 2 | The system will look for the budgeting range and display it on the screen. |
| 3 | The client clicks on the Search button. | 4 | The system will get the information from the database and provide the list of the event vendors and event venues of the that specific budget. |
| **Alternative Flow** | | | |
| | | 4-A | If the system finds nothing related to the search, it will respond with the message: *Oops! Nothing matched your search*. |

### 4.7.1.12 Bid for an Event

| **Name** | Bid for an Event | | |
|---|---|---|---|
| **Actors** | Client | | |
| **Summary** | The client shall click on the Make a Bid button, specify the budget and after successful verification, the system will sent his bid request to the event vendor. | | |
| **Pre-Conditions** | The client must be logged in.<br>The client must specify the bidding budget. | | |
| **Post-Conditions** | The system will send the client's bid request to the event vendor. | | |
| **Special Requirements** | The client shall not choose the budget less than the recommended one. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |

| | | | |
|---|---|---|---|
| 1 | The client click on the event venue. | 2 | The system will display the event venue details. |
| 3 | The client clicks on the Make a Bid button. | 4 | The system will display the recommended price asking for him/her to specify the budget. |
| 5 | The client specifies a valid budget and clicks on Send Bid Request. | 6 | The system will send the bid request to the event vendor, redirects the client to the event details page with a message: *Bid Successful* |
| **Alternative Flow** | | | |
| 5 | The client specifies an invalid budget | 6-A | The system will show the error message: *You cannot go less than the recommended budget.* |

### 4.7.1.13 Alter the Booking Details

| | | | |
|---|---|---|---|
| **Name** | Alter the Booking Details | | |
| **Actors** | Client | | |
| **Summary** | The client shall alter the booking details, such as time and special arrangements, and after successful verification, the system shall notify the event vendor with the altered booking details. | | |
| **Pre-Conditions** | The client must be logged in.<br>The client must have an event booked. | | |
| **Post-Conditions** | The system shall notify the event vendor with the new booking details. | | |
| **Special Requirements** | The client can only change the booking when there are 12 or more hours left in the booking time. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The client go to the Booking page | 2 | The system will display his bookings. |
| 3 | The client on the event he/she wish to modify. | 4 | The system will display the booking details. |
| 5 | The client clicks on the Alter Booking Details button. | 6 | The system will display the Alter Booking Details page asking for the new details. |
| 7 | The client enters the valid details and click on OK. | 8 | The system will alter the details in the database, send a notification to the event vendor, and redirects client to the Booking Details page with a message: *Altered Successfully*. |
| **First Alternative Flow** | | | |
| | | 6-A | The system will show the error message: *You cannot change the details as less than 12 hours are left.* |
| **Second Alternative Flow** | | | |
| 7 | The client enters invalid booking | 8-A | The system will show the error message: |

| | | | |
|---|---|---|---|
| | details. | | *Enter valid details* |

### 4.7.1.14 Add Special Arrangement for an Event

| | | | |
|---|---|---|---|
| **Name** | Add Special Arrangement | | |
| **Actors** | Client | | |
| **Summary** | The client shall add special arrangements, and after successful verification, the system shall notify the event vendor with the altered booking details. | | |
| **Pre-Conditions** | The client must be logged in.<br>The client must have an event booked. | | |
| **Post-Conditions** | The system shall notify the event vendor with the new booking details. | | |
| **Special Requirements** | The client can only add special arrangements when there are at least 12 hours left for the event.<br>The client have not added the special arrangements for that event before. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The client go to the Booking page | 2 | The system will display his bookings. |
| 3 | The client on the event he/she wish to modify. | 4 | The system will display the booking details. |
| 5 | The client clicks on the Add Special Arrangement. | 6 | The system will display the Add Special Arrangement page asking for the arrangements. |
| 7 | The client enters the arrangements. | 8 | The system will alter the details in the database, send a notification to the event vendor, and redirects client to the Booking Details page with a message: *Arrangements Added Successfully..* |
| **First Alternative Flow** | | | |
| 7 | The client left the arrangement empty | 6-A | The system will show the error message: *Please write something.* |

### 4.7.1.15 Book an Event

| | |
|---|---|
| **Name** | Book an Event |
| **Actors** | Client |
| **Summary** | The client shall book an event, and after successful verification, the system shall notify the event vendor, and remove the event from the bidding. |
| **Pre-Conditions** | The client must be logged in. |
| **Post-Conditions** | The system shall notify the event vendor and remove the event from the bidding page. |
| **Special** | None |

| **Requirements** | | | |
|---|---|---|---|
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The client shall go to the Bidding page | 2 | The system will display his biddings. |
| 3 | The client shall go to the Completed section. | 4 | The system will display the completed biddings. |
| 5 | The client clicks on the event he wants to book. | 6 | The system will display the boking details page. |
| 7 | The client clicks on Book an Event. | 8 | The system will add the details in the database, send a notification to the event vendor, and redirects client to the Booking Details page with a message: *Event Booked Successfully*. |

### 4.7.1.16 Add Event Venue to the Favorite List

| **Name** | Add Event Venue to the Favorite List | | |
|---|---|---|---|
| **Actors** | Client | | |
| **Summary** | The client shall add the event venue to the favorite list and after successful verification, the system will redirect the client to the Event details page. | | |
| **Pre-Conditions** | The client must be logged in.<br>The event venue must not already exist in the favorite list. | | |
| **Post-Conditions** | The system will redirect the client to the Event Details page. | | |
| **Special Requirements** | None. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The client shall click on the event venue. | 2 | The system will display the Event Details page. |
| 3 | The client clicks on the Add to Favorites button. | 4 | The system will make the changes in database and redirects the client to the Event Details page with a message: *Added Successfully*. |
| **Alternative Flow** | | | |
| | | 4-A | If it exists in the client's favorite list, the system will respond with the message: *Already exist in favorite list.* |

### 4.7.1.17 Remove Event Venue to the Favorite List

| **Name** | Add Event Venue/Event Vendor to the Favorite List |
|---|---|
| **Actors** | Client |

| | | | |
|---|---|---|---|
| **Summary** | The client shall remove the event venue to the favorite list and after successful verification, the system will redirect the client to the Event details page. | | |
| **Pre-Conditions** | The client must be logged in. | | |
| **Post-Conditions** | The system will redirect the client to the Favorite List page. | | |
| **Special Requirements** | None. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The client shall go to Favorite List page | 2 | The system will provide the list of event venues |
| 3 | The client clicks on the Remove button besides the event venue he/she wishes to delete. | 4 | The system will make the changes in database and responds with a message: *Removed Successfully.* |

### 4.7.1.18 Give a Rating to the Event Vendor

| | | | |
|---|---|---|---|
| **Name** | Give a Rating to the Event Vendor | | |
| **Actors** | Client | | |
| **Summary** | The client shall give the rating to the event vendor, the system will make the changes in the database, and redirect the client to the Completed Bookings page. | | |
| **Pre-Conditions** | The client must be logged in.<br>The client must have the event booked and completed. | | |
| **Post-Conditions** | The system will make the changes in the database and redirects to the Completed Bookings page. | | |
| **Special Requirements** | None | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The client shall go to the Bookings page. | 2 | The system will provide the list of his bookings. |
| 3 | The client shall go to the Completed Booking page. | 4 | The system will provide the list of his completed bookings |
| 5 | The client shall give a rating | 6 | The system will make the changes in database and redirects to the Completed Bookings page. |
| **Alternative Flow** | | | |
| | | 4-A | If the system finds nothing related to the search, it will respond with the message: *No Event Completed yet..* |

### 4.7.1.19 Give a Feedback to the Event Vendor

| | | | |
|---|---|---|---|
| **Name** | Give a Feedback to the Event Vendor | | |
| **Actors** | Client | | |
| **Summary** | The client shall give the feedback to the event vendor, the system will make the changes in the database, and redirect the client to the Completed Bookings page. | | |
| **Pre-Conditions** | The client must be logged in.<br>The client must have the event booked and completed. | | |
| **Post-Conditions** | The system will make the changes in the database and redirects to the Completed Bookings page. | | |
| **Special Requirements** | None | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The client shall go to the Bookings page. | 2 | The system will provide the list of his bookings. |
| 3 | The client shall go to the Completed Booking page. | 4 | The system will provide the list of his completed bookings |
| 5 | The client shall give a feedback | 6 | The system will make the changes in database and redirects to the Completed Bookings page. |
| **Alternative Flow** | | | |
| | | 4-A | If the system finds no completed bookings, it will respond with the message: *No Event Completed yet.* |

### 4.7.1.20 Cancel the Booking

| | | | |
|---|---|---|---|
| **Name** | Give a Feedback to the Event Vendor | | |
| **Actors** | Client | | |
| **Summary** | The client shall cancel the booking of the event, the system will make the changes in the database, and redirect the client to the Ongoing Bookings page. | | |
| **Pre-Conditions** | The client must be logged in.<br>The client must have the event booked. | | |
| **Post-Conditions** | The system will make the changes in the database and redirects to the Ongoing Bookings page. | | |
| **Special Requirements** | The client can cancel the booking with at least 12 hours before the event time. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The client shall go to the Bookings page. | 2 | The system will provide the list of his bookings. |

| | | | |
|---|---|---|---|
| 3 | The client shall go to the Ongoing Booking page. | 4 | The system will provide the list of his ongoing bookings |
| 5 | The client cancels the booking | 6 | The system will make the changes in database and redirects to the Ongoing Bookings page. |
| **Alternative Flow** | | | |
| | | 4-A | If the system finds no ongoing bookings, it will respond with the message: *No Event Completed yet.*. |
| **Second Alternative Flow** | | | |
| | | 6-A | The system will show the error message: *Not enough time left to cancel the booking.* |

### 4.7.1.21 Check for the Event Venue's Availability

| | | | |
|---|---|---|---|
| **Name** | Give a Feedback to the Event Vendor | | |
| **Actors** | Client | | |
| **Summary** | The client shall provide the date and system redirect the system will provide the schedule of the event venue on that date. | | |
| **Pre-Conditions** | The client must be logged in. | | |
| **Post-Conditions** | The system shall provide the schedule of the event venue on the specified date. | | |
| **Special Requirements** | The client can look for a date with at least one day remaining. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The client clicks the event venue. | 2 | The system will provide the event venue details. |
| 3 | The client provides a valid date. | 4 | The system will look in the database and provides the schedule of the event venue on that date. |
| **Alternative Flow** | | | |
| 3 | The client provides an invalid date. | 4-A | The system will show the error message: *Invalid Date Entered.* |

## 4.7.2 Use Cases for Event Vendor

### 4.7.2.1 Vendor Login

| | |
|---|---|
| **Name** | Vendor Login |
| **Actors** | Vendor |
| **Summary** | The vendor shall provide their email and password on the login form and after successful verification, redirect the vendor to the home page. |

| | |
|---|---|
| **Pre-Conditions** | The vendor must be in the database records either added by any of the authorized clients or added manually by a developer.<br>The client must not already be logged in. |
| **Post-Conditions** | The vendor's session is successfully established and shall be redirected to the home page. |
| **Special Requirements** | None |

| **Basic Flow** | | | |
|---|---|---|---|
| **Actor Action** | | **System Response** | |
| 1 | The vendor opens the login page. | 2 | The login page is displayed asking for email and password. |
| 3 | The vendor enters valid email and password. | 4 | The system verifies the email and password, establishes a session for the vendor and redirects the vendor to the home page. |
| **Alternative Flow** | | | |
| 3 | The vendor enters invalid email or password. | 4-A | The system responds with an error message: *Incorrect email or password entered.* |

### 4.7.2.2 Vendor Sign Up

| | |
|---|---|
| **Name** | Vendor Sign Up |
| **Actors** | Vendor |
| **Summary** | The vendor t shall provide their name, email, date of birth, CNIC, age, password, and the business the vendor owns, on the sign in form and after successful verification, redirect the vendor to the payment method page. |
| **Pre-Conditions** | The vendor must not be in the database records. |
| **Post-Conditions** | The vendor's session is successfully established and shall be redirected to the payment method page. |
| **Special Requirements** | None |

| **Basic Flow** | | | |
|---|---|---|---|
| **Actor Action** | | **System Response** | |
| 1 | The vendor opens the sign-up page. | 2 | The sign-up page is displayed asking for email and password. date of birth, CNIC, age and business owned by the vendor. |
| 3 | The vendor t enters valid credentials and proceeds to create the account by clicking on the sign-up button. | 4 | The system verifies that the name, email, date of birth, CNIC, age and password does not already exist in the database, establishes a session for the vendor and redirects the client to the payment method page. |

| First Alternative Flow | | | |
|---|---|---|---|
| 3 | The vendor enters the name, email, date of birth, CNIC, age and password that already exists in the database | 4-A | The system responds with an error message: *Already a user*. |
| **Second Alternative Flow** | | | |
| 3 | The vendor enters invalid the name, email, date of birth, CNIC, age and password. | 4-B | The system responds with an error message: *Enter valid credentials*. |

### 4.7.2.3 Vendor Log Out

| **Name** | Vendor Log out | | |
|---|---|---|---|
| **Actors** | Vendor | | |
| **Summary** | The vendor shall click on the log out button and after successful verification, redirect the vendor to the payment method page. | | |
| **Pre-Conditions** | The vendor must be logged in the system. | | |
| **Post-Conditions** | The vendor's session is successfully deleted and shall be redirected to the login page. | | |
| **Special Requirements** | None | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The vendor clicks on the log out button. | 2 | The system logs out the vendor and redirect to the login page. |

### 4.7.2.4 Alter Bank Information

| **Name** | Alter Bank Information | | |
|---|---|---|---|
| **Actors** | Vendor | | |
| **Summary** | The vendor shall go to Edit profile page, and then click on Change Banking Information. Then, enter the Bank account details and after the successful verification, the system will alter their payment method in the database and will be redirected to the Edit Profile Page. | | |
| **Pre-Conditions** | The vendor must be logged in. The client will need to be on the edit information page and have a bank account set up as their account to make/receive payments from. | | |
| **Post-Conditions** | The client's session is successfully altered in the database and shall be redirected to the edit profile page. | | |
| **Special Requirements** | Vendor must provide the valid account details. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The vendor opens the Edit Profile | 2 | The Edit Profile page is displayed. |

| | | | |
|---|---|---|---|
| | page. | | |
| 3 | The vendor clicks on Change Payment Method button. | 4 | The Payment Method page is displayed asking for the card details. |
| 5 | The vendor enters the card details. | 6 | The system verifies the card details, alter their payment method in the database and redirect to the Edit Page with the message: *Payment Method Altered.* |
| **Alternative Flow** | | | |
| 5 | The vendor enters valid card details. | 6-A | The system responds with an error message: *Card not valid. Payment method not altered* |

### 4.7.2.5 Receive Payment

| | | | |
|---|---|---|---|
| **Name** | Receive Payment | | |
| **Actors** | Vendor, Client | | |
| **Summary** | The vendor shall be able to receive a Payment, and after successful verification of vendor's bank account, the system will mark the status of the booking and store it in the database. | | |
| **Pre-Conditions** | The vendor must have an event listed to be booked. | | |
| **Post-Conditions** | The event shall be successfully booked and stored in the database and shall be redirected to the payment method page. | | |
| **Special Requirements** | The vendor must be verified as a vendor and have information of their bank account stored in the database. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The vendor verifies the account and provide bank information. | 2 | The system responds with a pop-up informing. Information updated successfully. The system also updates the status of vendor along with their banking information. |
| 3 | The client sends vendor the transaction for booking on an event. | 4 | The client is notified about the successful transaction along with a receipt as proof of transaction.<br>The vendor is notified amount the deposit made to their bank account. |
| 5 | The client clicks on Make a Payment button | 6 | The system verifies, alters the Booking status in the database, and responds with a message: *Payment Successful* |
| **Alternative Flow** | | | |
| 5 | The client clicks on Make a Payment button | 6-A | The system responds with an error message: *Payment Unsuccessful. Recharge Your Card or Change the Payment Method.* |

### 4.7.2.6 Alter the Profile

| | |
|---|---|
| **Name** | Alter the Profile |
| **Actors** | Vendor |
| **Summary** | The vendor shall go to the Edit Profile page, enter the name/password/email/CNIC/date of birth and after successful verification, redirect the user to the Edit Profile page. |
| **Pre-Conditions** | The vendor must be logged in and visit the Edit profile page. |
| **Post-Conditions** | The vendor's profile is successfully altered and shall be redirected to the Edit Profile page. |
| **Special Requirements** | The entered information must be valid and different from the previous ones. |

| **Basic Flow** | | | |
|---|---|---|---|
| **Actor Action** | | **System Response** | |
| 1 | The vendor opens the Edit Profile page. | 2 | The Edit Profile page is displayed showing the client's profile. |
| 3 | The vendor enters valid name/email/date of birth/CNIC/password. | 4 | The system verifies the information, alter it into the database, and redirects to the Edit Profile page with a message: *Successfully Changed* |
| **First Alternative Flow** | | | |
| 3 | The vendor enters the same name/email/date of birth/CNIC/password | 4-A | The system responds with an error message: *Unchanged Information Entered.* |
| **Second Alternative Flow** | | | |
| 3 | The vendor enters invalid name/email/date of birth/CNIC/password | 4-B | The system responds with an error message: *Invalid Information Entered.* |

### 4.7.2.7 Bid for an Event

| | |
|---|---|
| **Name** | Bid for an Event |
| **Actors** | Vendor |
| **Summary** | The vendor shall click on the Make a Bid button, specify the budget and after successful verification, the system will send the bid request to the client. |
| **Pre-Conditions** | The Vendor must be logged in. The vendor must be verified. The vendor must land on the client's bid request to place the bid. |
| **Post-Conditions** | The system will send the client's bid request to the event vendor. |
| **Special Requirements** | The client must not choose the budget less than the recommended one. |

| **Basic Flow** | |
|---|---|
| **Actor Action** | **System Response** |

| | | | |
|---|---|---|---|
| 1 | The vendor goes the page showings bids by client. | 2 | The system will display the lists of bids along with the current highest placed by a vendor. |
| 3 | The vendor clicks on the Make a Bid button. | 4 | The system will display the recommended price and ask to specify the budget. |
| 5 | The Vendor specifies a valid budget and clicks on Send Bid Request. | 6 | The system will send the bid request to the event vendor, redirects the client to the event details page with a message: *Bid Successful* |
| **First Alternative Flow** | | | |
| 5 | The client specifies an invalid budget | 6-A | The system will show the error message: *You cannot go less than the recommended budget.* |
| **Second Alternative Flow** | | | |
| 7 | The client enters invalid booking details. | 8-A | The system will show the error message: *Enter valid details* |

### 4.7.2.8 Add Venue to Venue List

| | |
|---|---|
| **Name** | Add Venue to Venue List |
| **Actors** | Vendor |
| **Summary** | The vendor will be able to add a venue to their existing venue list. On adding a venue, the user will be redirect to the venue list page. |
| **Pre-Conditions** | The Vendor must be logged in. The vendor must be verified. The vendor must land on the venue specifications page by clicking on the Add venue button within the venue list page. |
| **Post-Conditions** | The system will update the venue list and redirect the vendor to the venue list page. |
| **Special Requirements** | The venue to be added must not be the same as any other venue present in the list.<br>The vendor must specify all the information before clicking on the adding button to add the venue. |

| **Basic Flow** | | | |
|---|---|---|---|
| **Actor Action** | | **System Response** | |
| 1 | The vendor goes to the venue list page | 2 | The system will display the venue list page. |
| 3 | The vendor clicks on the Add a venue button. | 4 | The system will redirect the vendor to venue specifications page to add information regarding the venue to be added. |
| 5 | The Vendor enters the information for the new venue and clinks on the Add button. | 6 | The system will redirect the user to the venue list page with a new venue added into the list. |
| **First Alternative Flow** | | | |
| 5 | The Vendor enters the information of a | 6-A | The system will show the error message: |

| | | | |
|---|---|---|---|
| | venue that already exists within the list, and clinks on the Add button. | | *Venue already added to the list*. |
| **Second Alternative Flow** | | | |
| 5 | The Vendor enters incomplete information of a venue, and clinks on the Add button | 6-B | The system will show the error message: *Enter all the details for the venue.* |

### 4.7.2.9 Remove Venue to Venue List

| | | | |
|---|---|---|---|
| **Name** | Remove Venue to Venue List | | |
| **Actors** | Vendor | | |
| **Summary** | The vendor will be able to remove a venue to their existing venue list. On removing a venue, the user will be redirect to the venue list page. | | |
| **Pre-Conditions** | The Vendor must be logged in. The vendor must be verified. The vendor must be on the venue list page in-order to remove a venue from the list. | | |
| **Post-Conditions** | The system will update the venue list and keep the vendor to the venue list page. | | |
| **Special Requirements** | The venue should be added to list in order to remove the venue from the list. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The vendor goes to the venue list page | 2 | The system will display the venue list page. |
| 3 | The vendor clicks on the Remove a venue button. | 4 | The system will keep the vendor to venue list page and update the venue list. |

### 4.7.2.10 Alter Venue to Venue List

| | | | |
|---|---|---|---|
| **Name** | Alter Venue to Venue List | | |
| **Actors** | Vendor | | |
| **Summary** | The vendor will be able to alter venue from the venue list. On altering a venue, the user will be redirect to the venue list page. | | |
| **Pre-Conditions** | The Vendor must be logged in. The vendor must be verified. The vendor must land on the venue specifications page by clicking on the Alter venue button within the venue list page. | | |
| **Post-Conditions** | The system will update the venue list and redirect the vendor to the venue list page. | | |
| **Special Requirements** | There should be change in at least one of the specifications to successfully alter the venue. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The vendor goes to the venue list page | 2 | The system will display the venue list |

| | | | |
|---|---|---|---|
| | | | page. |
| 3 | The vendor clicks on the alter a venue button. | 4 | The system will redirect the user to venue specifications page to alter information regarding the venue. |
| 5 | The Vendor alters the information for the new venue and clinks on the alter button. | 6 | The system will redirect the user to the venue list page with venue list updated. |
| **Alternative Flow** | | | |
| 5 | The Vendor enters the information of a venue that already exists within the list, and clinks on the Add button. | 6-A | The system will show the error message: *Venue already added to the list*. |
| **Alternative Flow** | | | |
| 5 | The Vendor enters incomplete information of a venue, and clinks on the Add button | 6-B | The system will show the error message: *Enter all the details for the venue*. |

### 4.7.2.11 Vendor authentication.

| | |
|---|---|
| **Name** | Vendor authentication. |
| **Actors** | Vendor |
| **Summary** | the vendor will be able to get their business authenticated by providing the required information to the website. On successful Authentication The vendor will be notified. And have their status of the business updated. As authenticated. |
| **Pre-Conditions** | The Vendor must be logged in. The vendor must be verified. the vendor must land on the authenticate business page to provide the details. Regarding their business Before submitting the request. The request for authentication. |
| **Post-Conditions** | The vendor will be redirected to their home page. The system will. Update. The vendor's status as in progress. |
| **Special Requirements** | the vendor needs to have their bank account information stored in the data base. |

| **Basic Flow** | | | |
|---|---|---|---|
| **Actor Action** | | **System Response** | |
| 1 | The vendor goes to the authenticate account page. | 2 | The system will redirect the vendor to the authenticate account page. |
| 3 | The vendor adds all the information required to authenticate their business. | 4 | The system will redirect the user to their home page. And update the status of the verification as in progress. |
| **Alternative Flow** | | | |
| 5 | the vendor does not provide all the credentials needed to get the business verified. | 6-A | The system responds with an error message. Enter all the credentials required. |

#### 4.7.2.12 View a List of Bookings.

| | | | |
|---|---|---|---|
| **Name** | View the list of bookings. | | |
| **Actors** | Vendor | | |
| **Summary** | The vendor will be able to view the list of all the bookings made by the clients. For the event venues. | | |
| **Pre-Conditions** | The vendor must be logged into their account. The vendor must be verified as a business owner. The vendor must have a list of existing venues available to be able to view the bookings. | | |
| **Post-Conditions** | The system will keep the users on the view cooking page. | | |
| **Special Requirements** | None. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The window opens list of bookings page. | 2 | The system will redirect the vendor to the List of bookings page. |

#### 4.7.2.13 Contact the User

| | | | |
|---|---|---|---|
| **Name** | Contact the user. | | |
| **Actors** | Vendor | | |
| **Summary** | The winner will be able to contact the client through a messaging platform to respond to any queries of the of the client. | | |
| **Pre-Conditions** | the window must be signed in. The window must visit the message page. | | |
| **Post-Conditions** | The vendor will send a message to the user. The system will post the message on the message page. And keep the user on the same page. | | |
| **Special Requirements** | The winter needs to be a verified user. The vendor needs to have a message in the inbox from a client to respond. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The vendor goes to the messaging page. | 2 | The System will redirect the vendor to the messaging page. |

## 4.7.3 Use Cases for Admin

#### 4.7.3.1 Admin Login

| | |
|---|---|
| **Name** | Admin Login |
| **Actors** | Admin |
| **Summary** | The admin shall provide their email and password on the login form and after successful verification, redirect the admin to the home page. |

| | |
|---|---|
| **Pre-Conditions** | The admin must be in the database records added manually by a developer.<br>The admin must not already be logged in. |
| **Post-Conditions** | The admin's session is successfully established and shall be redirected to the home page. |
| **Special Requirements** | None |

| **Basic Flow** | | | |
|---|---|---|---|
| **Actor Action** | | **System Response** | |
| 1 | The admin opens the login page. | 2 | The login page is displayed asking for email and password. |
| 3 | The admin enters valid email and password. | 4 | The system verifies the email and password, establishes a session for the user and redirects the user to the home page. |
| **Alternative Flow** | | | |
| 3 | The admin enters invalid email or password. | 4-A | The system responds with an error message: *Incorrect email or password entered.* |

### 4.7.3.2 View the List of Event Vendors

| | |
|---|---|
| **Name** | View the List of Event Vendors |
| **Actors** | Admin |
| **Summary** | The admin shall access the list of event vendors and after successful verification, the admin shall get the list of event vendors. |
| **Pre-Conditions** | The admin must already be logged in. |
| **Post-Conditions** | The admin shall get the list of event vendors. |
| **Special Requirements** | None |

| **Basic Flow** | | | |
|---|---|---|---|
| **Actor Action** | | **System Response** | |
| 1 | The admin opens the list of event vendors page | 2 | The system will look for the event vendors in the database and will display them on the list of event vendors page. |

### 4.7.3.3 View the List of Event Venues

| | |
|---|---|
| **Name** | View the List of Event Venues |
| **Actors** | Admin |
| **Summary** | The admin shall access the list of event venues and after successful verification, the admin shall get the list of event venues. |
| **Pre-Conditions** | The admin must already be logged in. |

| | | | |
|---|---|---|---|
| **Post-Conditions** | The admin shall get the list of event venues. | | |
| **Special Requirements** | None | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The admin opens the list of event venues page | 2 | The system will look for the event venues in the database and will display them on the list of event venues page. |

### 4.7.3.4 Edit Profile

| | | | |
|---|---|---|---|
| **Name** | Edit Profile | | |
| **Actors** | Admin | | |
| **Summary** | The client shall go to the Edit Profile page, enter the name/password/email/CNIC/date of birth and after successful verification, redirect the admin to the Profile page. | | |
| **Pre-Conditions** | The admin must be logged in. | | |
| **Post-Conditions** | The admin's profile is successfully altered and shall be redirected to the Profile page. | | |
| **Special Requirements** | The entered information must be different from the previous ones. | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The admin opens the Edit Profile page. | 2 | The Edit Profile page is displayed showing the admin's profile. |
| 3 | The admin enters valid name/email/date of birth/CNIC/password. | 4 | The system verifies the information, alter it into the database, and redirects to the Profile page with a message: *Successfully Changed* |
| **First Alternative Flow** | | | |
| 3 | The admin enters the same name/email/date of birth/CNIC/password | 4-A | The system responds with an error message: *Unchanged Information Entered.* |
| **Second Alternative Flow** | | | |
| 3 | The admin enters invalid name/email/date of birth/CNIC/password | 4-B | The system responds with an error message: *Invalid Information Entered.* |

### 4.7.3.5 Change Credentials

| | |
|---|---|
| **Name** | Change Credentials |
| **Actors** | Admin |

| | |
|---|---|
| **Summary** | The admin shall go to the Edit Profile page, enter the password and email and after successful verification, redirect the user to the Profile page. |
| **Pre-Conditions** | The admin must be logged in. |
| **Post-Conditions** | The admin's credentials are successfully altered and shall be redirected to the Edit Profile page. |
| **Special Requirements** | The entered information must be different from the previous ones. |

| **Basic Flow** | | | |
|---|---|---|---|
| **Actor Action** | | **System Response** | |
| 1 | The admin opens the Edit Profile page. | 2 | The Edit Profile page is displayed showing the admin's profile. |
| 3 | The admin changes the password/email or both | 4 | The system verifies the information, alter it into the database, and redirects to the Profile page with a message: *Successfully Changed* |
| **First Alternative Flow** | | | |
| 3 | The admin enters the same password/email | 4-A | The system responds with an error message: *Unchanged Information Entered.* |
| **Second Alternative Flow** | | | |
| 3 | The admin enters invalid email/password or both | 4-B | The system responds with an error message: *Invalid Information Entered.* |

### 4.7.3.6 View the List of Event Bookings

| | |
|---|---|
| **Name** | View the List of Event Bookings |
| **Actors** | Admin |
| **Summary** | The admin shall access the list of event bookings and after successful verification, the admin shall get the list of event bookings. |
| **Pre-Conditions** | The admin must already be logged in. |
| **Post-Conditions** | The admin shall get the list of event bookings. |
| **Special Requirements** | None |

| **Basic Flow** | | | |
|---|---|---|---|
| **Actor Action** | | **System Response** | |
| 1 | The admin opens the list of event bookings page | 2 | The system will look for the event bookings in the database and will display them on the list of event bookings page. |

### 4.7.3.7 Add an Event Vendor

| | |
|---|---|
| **Name** | Add an Event Vendor |

| | | | |
|---|---|---|---|
| **Actors** | Admin | | |
| **Summary** | The admin shall add an event vendor in the add vendor form and after successful verification, redirect the admin to the home page. | | |
| **Pre-Conditions** | The admin must not be in the database records.<br>The admin must not already be logged in. | | |
| **Post-Conditions** | The admin shall be redirected to the home page. | | |
| **Special Requirements** | None | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The admin opens the Add an Event Vendor. | 2 | The Add an Event vendor is displayed asking for event vendor's information. |
| 3 | The admin enters valid event vendor information. | 4 | The system make the changes in the database and redirects the admin to the home page with a message: *Successfully Added.* |
| **First Alternative Flow** | | | |
| 3 | The admin enters the event vendor information that already exist. | 4-A | The system responds with an error message: *Already an Event Vendor* |
| **Second Alternative Flow** | | | |
| 3 | The admin enters invalid event vendor information. | 4-B | The system responds with an error message: *Invalid Information.* |

### 4.7.3.8 Delete an Event Vendor

| | | | |
|---|---|---|---|
| **Name** | Delete an Event Vendor | | |
| **Actors** | Admin | | |
| **Summary** | The admin shall delete an event vendor and after successful verification, the event vendor shall be deleted, and the admin shall be redirected to the List of Event Vendors page. | | |
| **Pre-Conditions** | The admin must already be logged in. | | |
| **Post-Conditions** | The event vendor shall be deleted, and admin shall be redirected to the List of Event Vendors page. | | |
| **Special Requirements** | None | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The admin opens the list of event vendors page | 2 | The system will look for the event vendors in the database and will display them on the list of event vendors page. |
| 3 | The admin deletes an event vendor. | 4 | The system will make the changes in the database and redirects to the list of events page with a message: *Successfully Deleted* |

#### 4.7.3.9 Suspend an Event Vendor

| | | | |
|---|---|---|---|
| **Name** | Suspend an Event Vendor | | |
| **Actors** | Admin | | |
| **Summary** | The admin shall suspend an event vendor and after successful verification, the event vendor shall be suspended, and the admin shall be redirected to the List of Event Vendors page. | | |
| **Pre-Conditions** | The admin must already be logged in. | | |
| **Post-Conditions** | The event vendor shall be suspended, and admin shall be redirected to the List of Event Vendors page. | | |
| **Special Requirements** | None | | |
| **Basic Flow** | | | |
| **Actor Action** | | **System Response** | |
| 1 | The admin opens the list of event vendors page | 2 | The system will look for the event vendors in the database and will display them on the list of event vendors page. |
| 3 | The admin suspends an event vendor. | 4 | The system will make the changes in the database and redirects to the list of events page with a message: *Successfully Suspended* |

## 4.8 Graphical User Interface

This section should give the GUI dumps of each screen, with reference to the users. The navigation flow of each user is also required, and each GUI should mark the functionality/use case that it covers.

## 4.8.1 Login and Sign Up

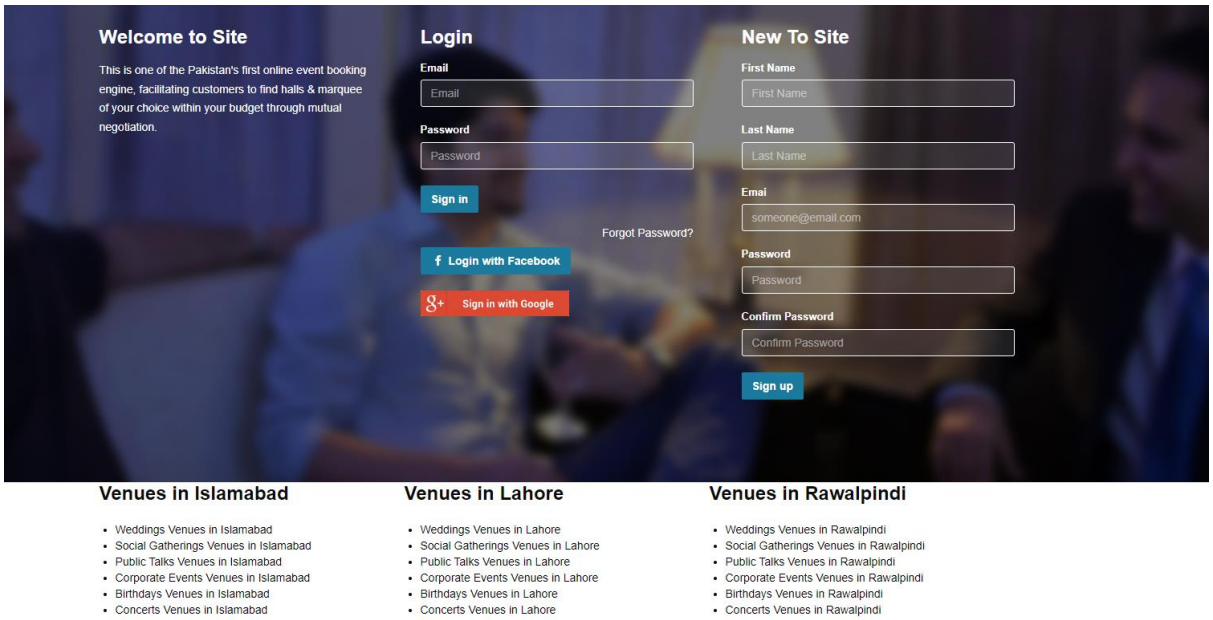


**Figure 2: Login and Sign Up screen**

*The user can Login and Sign Up using the right credentials and clicking the respective button.*

## 4.8.2 Search Events

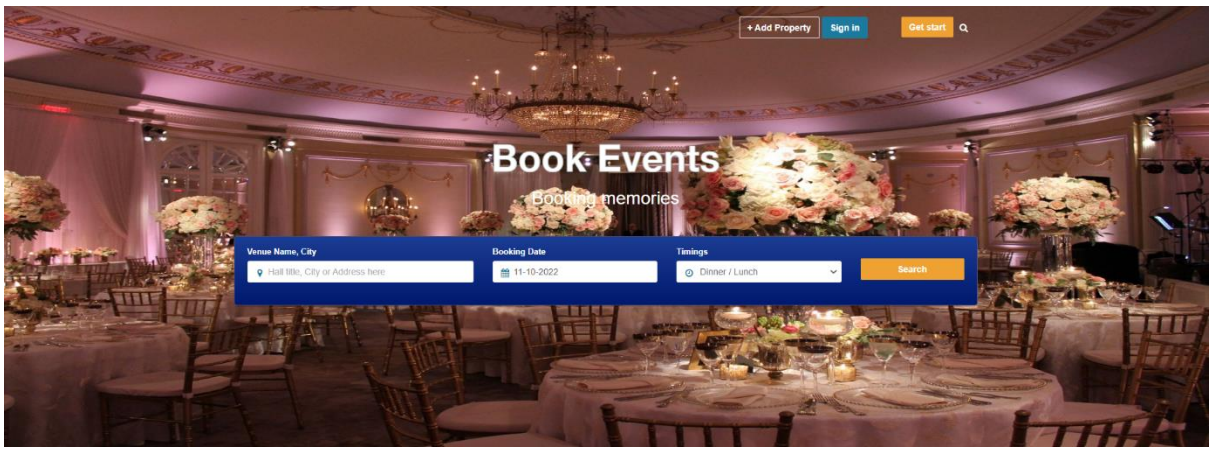


**Figure 3: Search Event screen**

*The user can look for events by providing the city, time, and date, and clicking the Search button.*

## 4.8.3 Sign Up for New Vendor

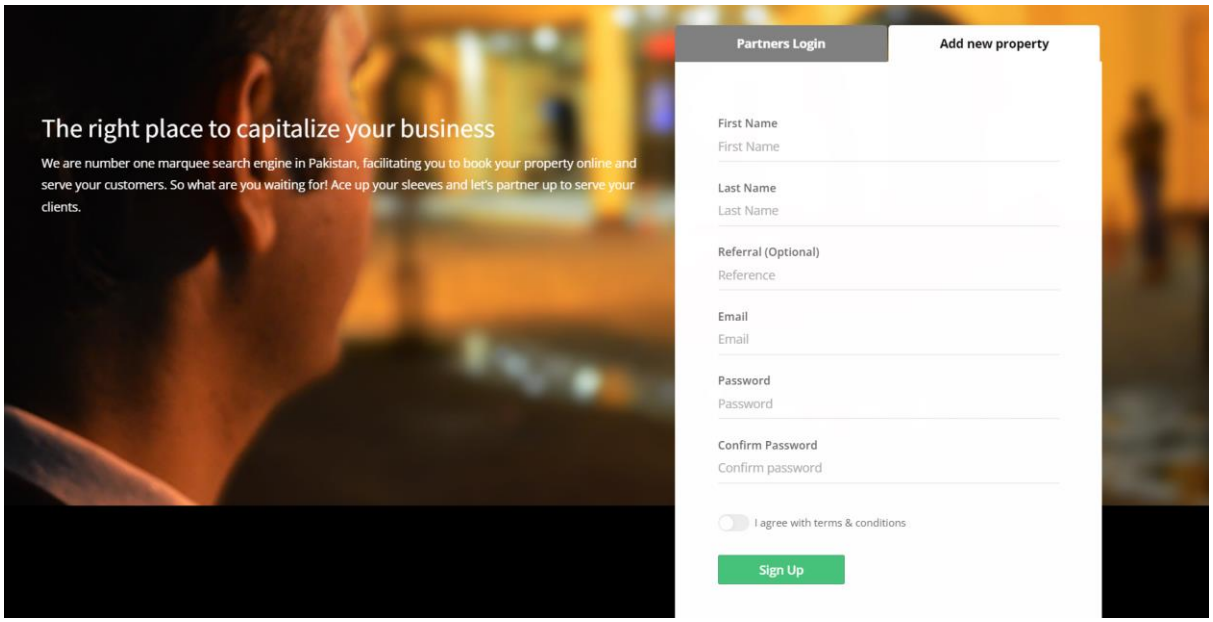


**Figure 4: Sign Up for new vendor screen**

*The new vendor can sign up by providing the right information and click on the Sign Up button.*

## 4.8.4 How the Website Works

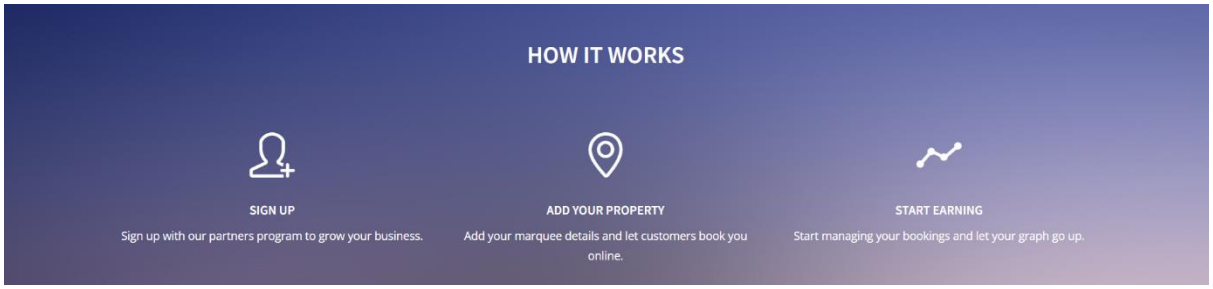


**Figure 5: How the Website Works screen**

*The user can sign up as a client, vendor, and start earning by clicking the respective buttons.*

## 4.8.5 View Event Venues

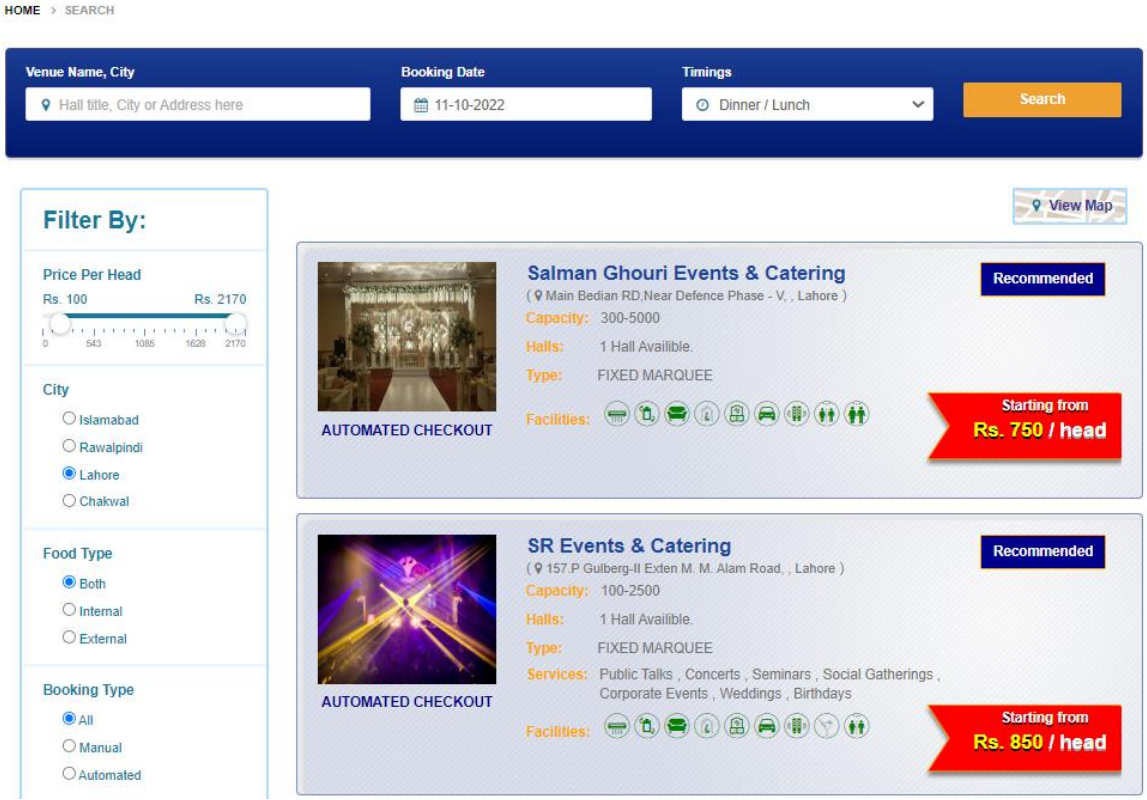


**Figure 6: View Event Venues screen**

*The user can view the event venues on this screen.*

## 4.8.6 Check Availability

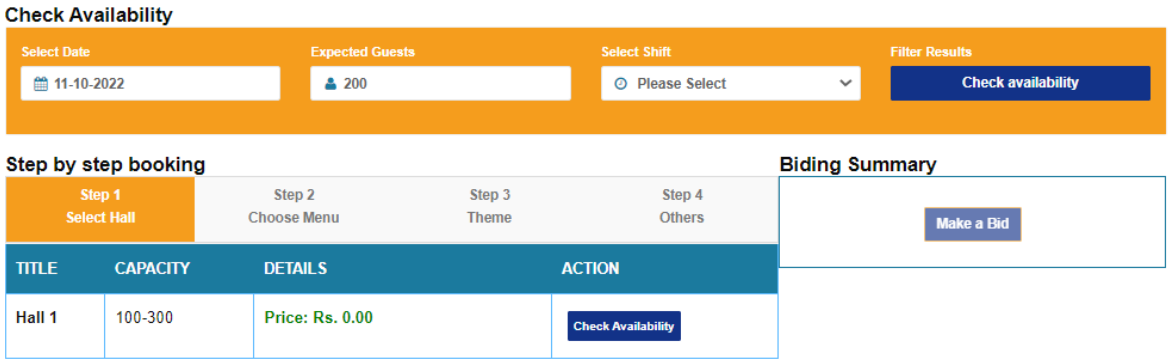


**Figure 7: Check Availability screen**

*Client can check for the availability of an event venue by providing date, expected guests, and shift, and then click on Check Availability button.*

# 4.9 Database Design

## 4.9.1 ER Diagram

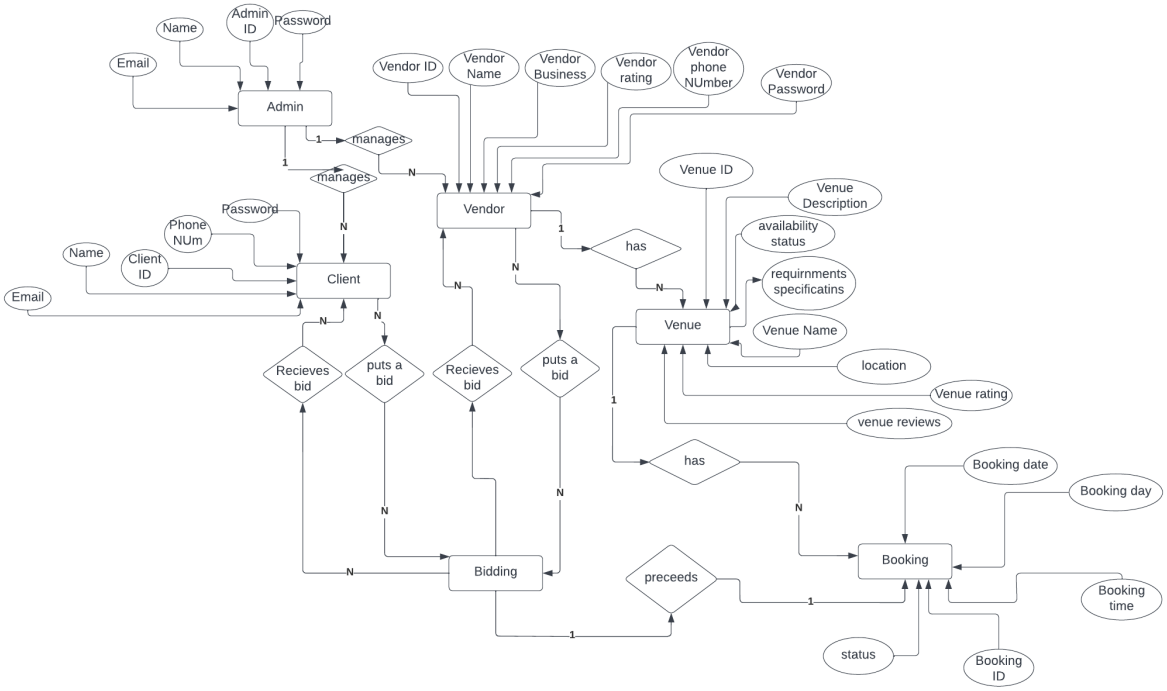


**Figure 8: ER Diagram**

*The figure shows the Entity Relationship diagram for Event Organization System consisting of the system's major entities along with their attributes.*

## 4.9.2 Data Dictionary

| Entity | Attribute | Data Type | Nullable | Relation Type | Description |
|---|---|---|---|---|---|
| Admin | Admin_ID | Int | No | 1 to M | Primary key |
| | Name | String | No | | Name of Admin |
| | Email | String | No | | Email of Admin |
| | Password | String | No | | Password of Admin |

| | | | | | |
|---|---|---|---|---|---|
| Vendor | Vendor_ID | Int | No | M to M | Primary Key |
| | Name | String | No | | Name of Vendor |
| | Business | String | No | | Business of Vendor |
| | Rating | Float | No | | Rating of Vendor |
| | Phone_Number | Int | No | | Phone Number of Vendor |
| | Password | String | No | | Password of Vendor |
| Client | Client_ID | Int | No | M to M | Primary Key |
| | Name | String | No | | Name of Client |
| | Phone_Number | Int | No | | Phone Number of Client |
| | Email | String | No | | Email of Client |
| | Password | String | No | | Client's Password |
| Venue | Venue_ID | Int | No | 1 to M | Primary key |
| | Description | String | No | | Venue's Profile |
| | Availability Status | Boolean | Yes | | Venue Vacuity |
| | RequirnmentSpecifications | String | No | | Extra Features |
| | Name | String | No | | Venue's Name |
| | Location | String | No | | Exact Location |
| | Rating | Float | No | | Venue's Rating |
| | Reviews | String | No | | User Reviews |

| | | | | | |
|---|---|---|---|---|---|
| Booking | Booking_ID | Int | No | 1 to M | Primary key |
| | Day | String | No | | Booking Day |
| | Time | String | No | | Booking Time |
| | Date | String | No | | Booking date |
| | Status | Boolean | Yes | | Booking Status |

**Table 1: Data Dictionary**

*The table consists of the variables that the entities in database will along with their information.*

## 4.10 Risk Analysis

- All team members will need to devote a considerable amount of time and effort to the project. There are additional duties and projects that each participant must do in order to see this project through to its conclusion. This might make it tough to fulfil any upcoming deadlines.
- To guarantee that the web app can fulfil the demands of the current market and the desired audience, extensive research must be undertaken.
- To safeguard the web application against potential attacks, enhanced security check methods must be used.
- To keep the web application quick and responsive, utilize the suggested optimization tactics and processes.
- Many more people will utilize the web app if its user interface is well designed. Users aren't given ease of use, and the product suffers as a result, when the user interface is ugly and difficult to use.

# Chapter 5: High-Level and Low-Level Design

This chapter covers the high and low-level design of the Event Organization System, which includes system overview, assumptions, dependencies, general constraints, architecture of the system, class and sequence diagrams of the project's workflow, and the policy tactics.

## 5.1 System Overview

Event Organization System is a web application that aims to ease the event booking system for general population as well as the event vendors by proving an online platform. The framework used to implement such a web-based application is MERN Stack. The backend will be in NodeJS, the frontend will be in ReactJS, and ExpressJS will be used to integrate both front and backend. Moreover, the system data will be stored in a cloud-based database, such as MongoDB.

The architecture technique used to implement the MERN Stack is the MVC (Model View Controller). It's an architectural style that divides apps into the model, view, and controller layers. These parts were purpose-built to deal with various aspects of web-based development. When it comes to developing scalable and flexible web applications, MVC is one of the most popular frameworks employed.

The users of the web application are divided into three sub-groups, i.e., client, vendor, and admin. Each of them will have different functionalities according to their roles. Clients are the generation which will book the events based on the consumer needs. The events booked by the clients are provided by the second sub-group called the event vendors. To manage the client-vendor interaction and the authenticity of the event vendors, the system will have a third sub-group, named admin that will have the authority to manage the system.

### 5.1.1 Client

To utilize the web application, the clients will have to sign up by providing their details, such as name, email, CNIC, phone number, payment method, etc. Once the initial sign up has been dealt with, clients can view the list of all event vendors and venues available along with additional filter, which can be applied to narrow down the list suitable to the client's needs. The client can add/remove the event to their favorite list. Furthermore, they can place bids or book an event. They will also have a right to cancel the booking within the first 24 hours of placing the booking. Clients can give feedback and rating to the event venue they have successfully had arranged to maintain the record of the vendor, so their profile is maintained accordingly.

### 5.1.2 Vendor

To get their business out into the online platform using Event Organization System, they can register themselves as vendor of the website by providing the necessary details. Unlike the clients, Vendors need to be verified by the admin to take their account deemed as usable and their venues bookable by the clients. Once the verification process is completed the vendors can place new venues, remove existing venues, and update the special arrangements provided by each venue. The vendors can bid on the client's budget or their specifications, they can accept, discard, or raise the bid placed by the clients. Moreover, a vendor can cancel the booking within the specified time. A vendor can also directly contact the clients through an integrating messaging platform.

### 5.1.3 Admin

The admin will have to login into the system to utilize it. The admin will be able to alter his profile and credentials. The admin can manage the complaints against vendor and clients. The admin can suspend the vendor temporarily. Also, the admin can delete the vendor in case of some serious complaints. Besides that, the admin can see the list of event vendors, event venues, and the clients registered on the system. The admin will have the authority to view the list of on-going, completed, or canceled bookings and biddings made through the system.

## 5.2 Design Considerations

This section describes many of the issues which need to be addressed or resolved before attempting to devise a complete design solution.

### 5.2.1 Assumptions and Dependencies

#### 5.2.1.1 Hardware

- 2.0 GHz processor at minimum
- 4GB recommended RAM size with a minimum threshold of 2GB
- Core i3 Processor and above

#### 5.2.1.2 Software

- VS Code Editor
- MongoDB
- NodeJS
- ExpressJS
- ReactJS
- Google Chrome, Mozilla Firefox, Microsoft Edge, and other web browsers

#### 5.2.1.3 End-User Characteristics

- End-users are not IT experts
- End-users are people who use the system frequently
- End-users are assumed to be able to work on an intuitive interface
- End-users have a reliable internet connection
- End-users have the updated version of their preferred web browser

### 5.2.2 General Constraints

#### 5.2.2.1 Validation Constraint

Event vendors are needed to register their businesses and provide relevant information to utilize this online application. This serves as proof that the vendors are legitimate businesses. In addition, to book an event, a user will be required to sign up for an account and provide a payment method. A viewer, on the other hand, does not need credentials to engage in the activity of window shopping.

#### 5.2.2.2 End-User Constraint

For this MERN-Stack application to work properly, the user will need to have the most recent version of their browser installed. In addition, to utilize our web application, a user is required to have a stable internet connection.

#### 5.2.2.3 Data Repository Constraint

MongoDB is a kind of cloud-based storage that serves as the support system for the backend of a web-based application. Only the administrator has permission to make changes to the information that is stored in the database that is being maintained. This is done to guarantee that accurate insertions, modifications, and deletions are made to the data.

#### 5.2.2.4 Interoperability Constraint

Client and server must reach a consensus on a single protocol for storing and distributing the data to proceed.

#### 5.2.2.5 Network Communication Constraint

The hypertext transfer protocol, or HTTP, is used, which allows clients and servers to communicate and receive data with one another.

#### 5.2.2.6 Performance Constraint

To travel around the pages and carry out the numerous operations in an easy manner, a user will be expected to have a connection to the internet that is of a high speed.

#### 5.2.2.7 Language Constraint

For the user to be permitted to use our web application, they will need to have a sufficient command of the English language.

### 5.2.3 Goals and Guidelines

#### 5.2.3.1 The KISS Principle

It emphasizes on the principle that a project can't reach its full potential if just a fraction of the target audience can't utilize it effectively, and that everyone should be able to comprehend it.

#### 5.2.3.2 Let Function Inform Design

Instead of incorporating a design the population is not familiar with, causing understandability issue making it difficult for the new user, the design provided against all the functions should be accustomed with the general understanding of the population

### 5.2.4 Development Methods

The development method chosen for event organization system is Agile. The primary benefit of approaching agile development method is that development is done in increments, improving the quality of the developed application in each iteration to work towards the optimized application.

The technique used to carry out agile development is Scrum. In Scrum, a backlog is maintained ordered based on the precedence of the functionalities implemented by the product owner being Ali Aamir, Ammar Ahmad, and Muneeb-Ur-Rehman. The Scrum team includes the product owners, evaluators, and the project advisor. The requirements are specified in the Sprints to which the deadline must be met delivering the required specifications.

In each Sprint, a potentially shippable product increment is delivered to which a sprint review is given by the project advisor and the evaluators classified as the Scrum Master. After the Sprint review has been conducted, Sprint retrospective is carried out to get the crux of the flaws and the acceptable functionalities in the executed Sprint, alongside implementing the targeted

goals for the next Sprint keeping in view the positives and negatives extracted from the current Sprint.

## 5.3 System Architecture

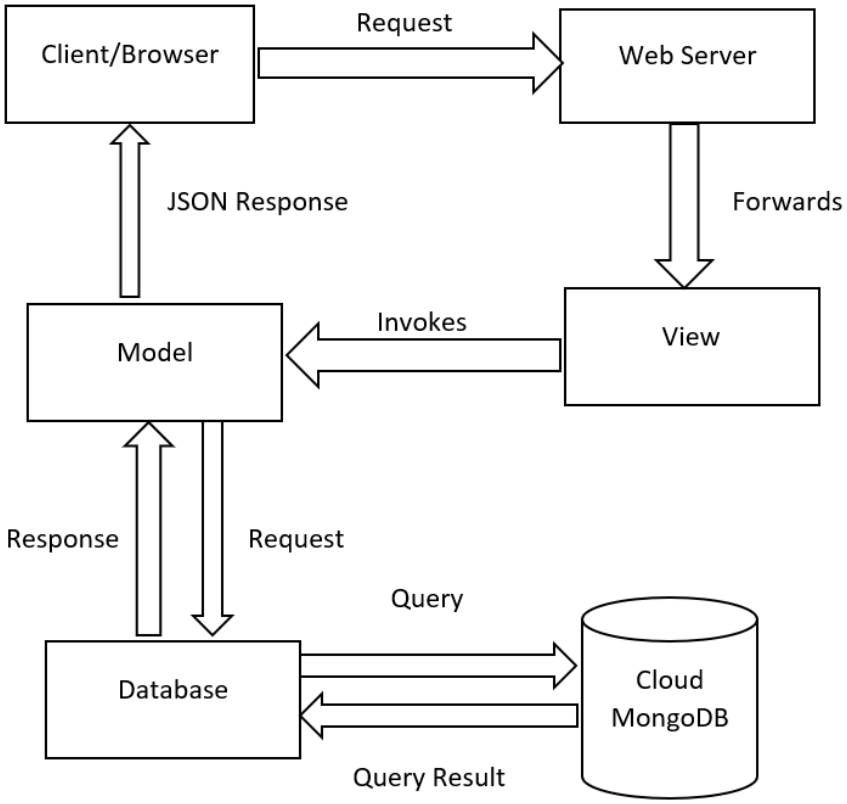


**Figure 9: System Architecture**

*This figure represents the workflow of the Event Organization System, including how the requests and responses are handled.*

The system architecture technique used in the project is MVC. MVC stands for Model View Controller. MVC is an architectural pattern that guides the design of whole web application. While this is commonly referred to as a design pattern, we might be missing the mark if we only refer to it in that capacity. In this architecture, the client/browser on which the web application is running sends a request to the web server, which then forwards the request to View. View invokes the Model component, which is connected to the database. Model sends a request to the database, which is Cloud MongoDB in this scenario. It sends the data/response back to the Model component, which converts the response in JSON format, and deliver it to the client/browser.

## 5.3.1 Subsystem Architecture

### 5.3.1.1 Admin Component

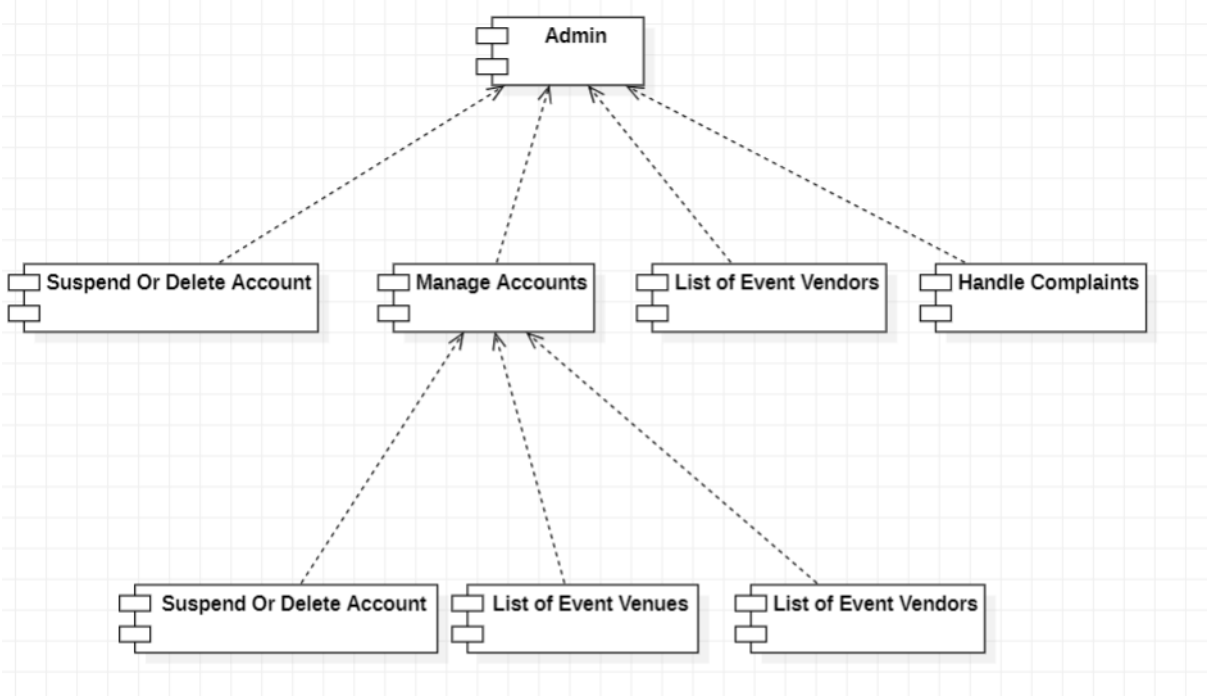


**Figure 10: Admin Component**
*This figure represents the component diagram for Admin.*

After being verified by the server's database, the admin will be granted access. The admin will manually verify the event vendor after receiving the sign-up request and create an account if the verification is successful. Using the manage account component, an admin can query the database for a list of available Event vendors and venues. Admin will be able to remove or temporarily suspend vendor accounts as needed. With the help of the handle complaints component, the admin can address user concerns.

### 5.3.1.2 Client Component

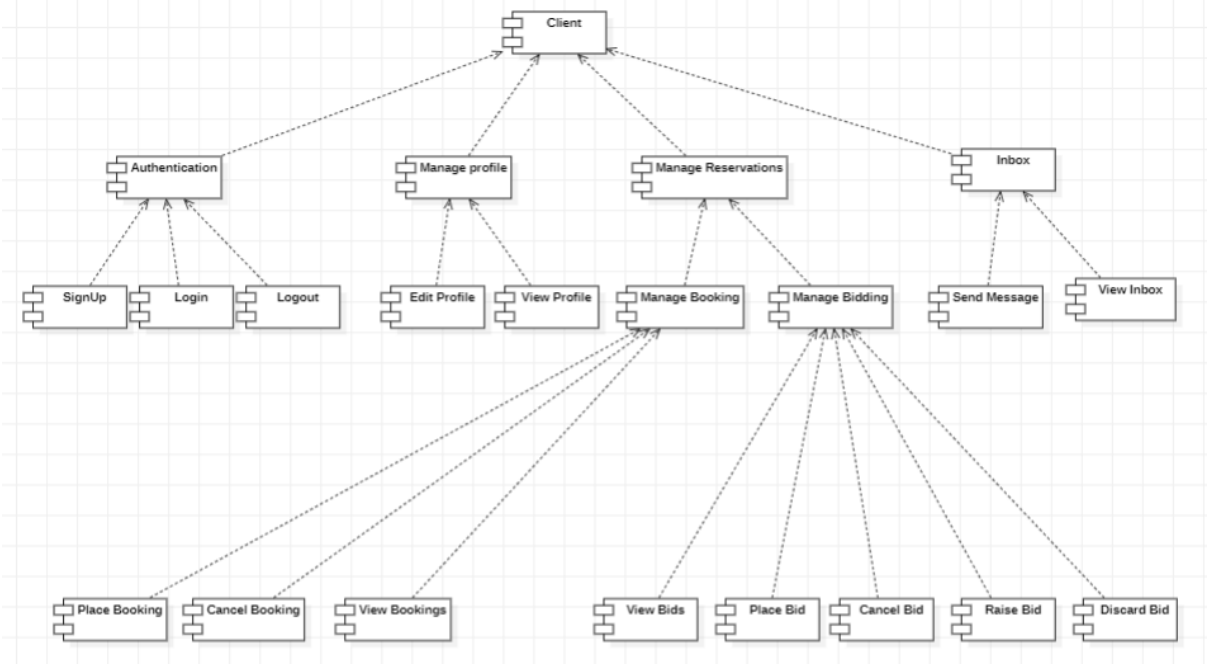


**Figure 11: Client Component**

*This figure represents the component diagram of Client.*

The Authentication component will allow the client to register, login, and logout. The client will be able to modify the profile details via Manage Profile component. Using the Manage Reservations, the client will be able to bid and book an event. From the Bidding component, the client can place, view, cancel, discard, and raise a bid. With the help of Booking component, the client can view their bookings, cancel, and book an event. The Inbox component will allow the client to send a new message to a vendor and view their messages.

#### 5.3.1.3 Vendor Component

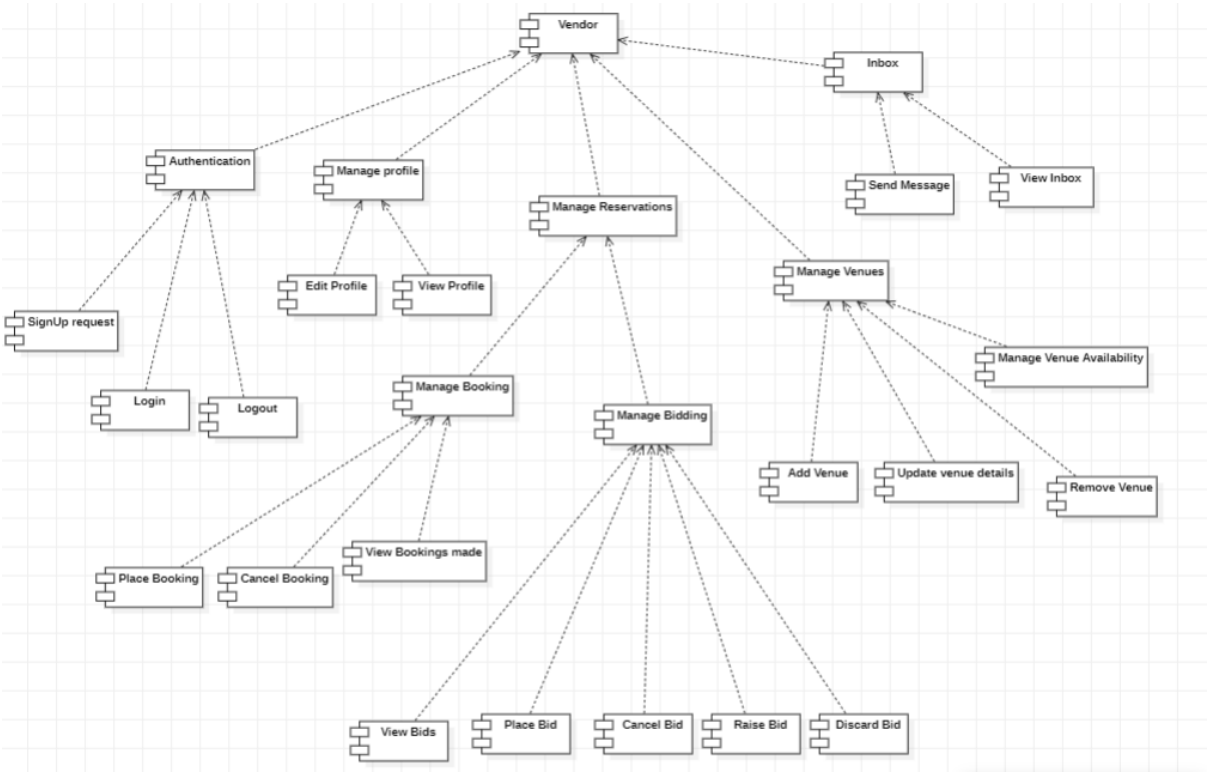


**Figure 12: Vendor Component**

*This figure represents the component diagram of Vendor.*

The Authentication component will enable the vendor to make a sign-up request, login, and logout. The Manage Profile component will allow the vendor to alter the profile. The vendor will be able to bid and monitor the events booked on his registered venues by using the Manage Reservations feature. The vendor may place, view, cancel, discard, and increase a bid via the Bidding component. The vendor may see and cancel reservations for venues using the Booking component. Moreover, the Manage Venue component will allow the vendor to add, modify, remove, and update the availability of the venue. The vendor may use the Inbox component to send a new message to a client and examine their previous messages.

## 5.4 Architectural Strategies

### 5.4.1 Tech Stack

The MERN stack will be used since it is the most widely used stack for developing JavaScript applications. In MERN stands for MongoDB, ExpressJS, ReactJS, and NodeJS. If you want to create a full stack JavaScript application, you should use NodeJS, which is a runtime for JavaScript. React is a front-end node package that makes it simple and scalable to render UI components. We won't be trapped on a difficulty using this stack for too long because of the large MERN stack community and the fact that it is the most widely used JavaScript framework.

### 5.4.2 Error Logging

Pino, a logging module, will be used to log any errors that occur in our web application. Pino, one of the most well-known Node.js logging libraries, advertises itself as a very low overhead solution. Its rapid speed is due to asynchronous logging.

### 5.4.3 Database Management

Since we are using the MERN stack, we have decided to go with the NoSQL database MongoDB. To be NoSQL indicates that it does not use the standard SQL language to construct connections between data entities. Our time spent on database and modelling setup will be reduced, and it is very adaptable and scalable.

### 5.4.4 User Interface Paradigms

The interface we create will be intuitive, straightforward, and basic. Our goal is to make our application accessible to as wide an audience as possible, thus we'll be using a design approach that emphasizes ease of use.

### 5.4.5 External Dependencies

The Event organization system is dependent on external services for its complete working functionalities. The platform chosen to provide the existing external dependencies is most likely to be AWS or Azure.

## 5.5 Domain Model/Class Diagram

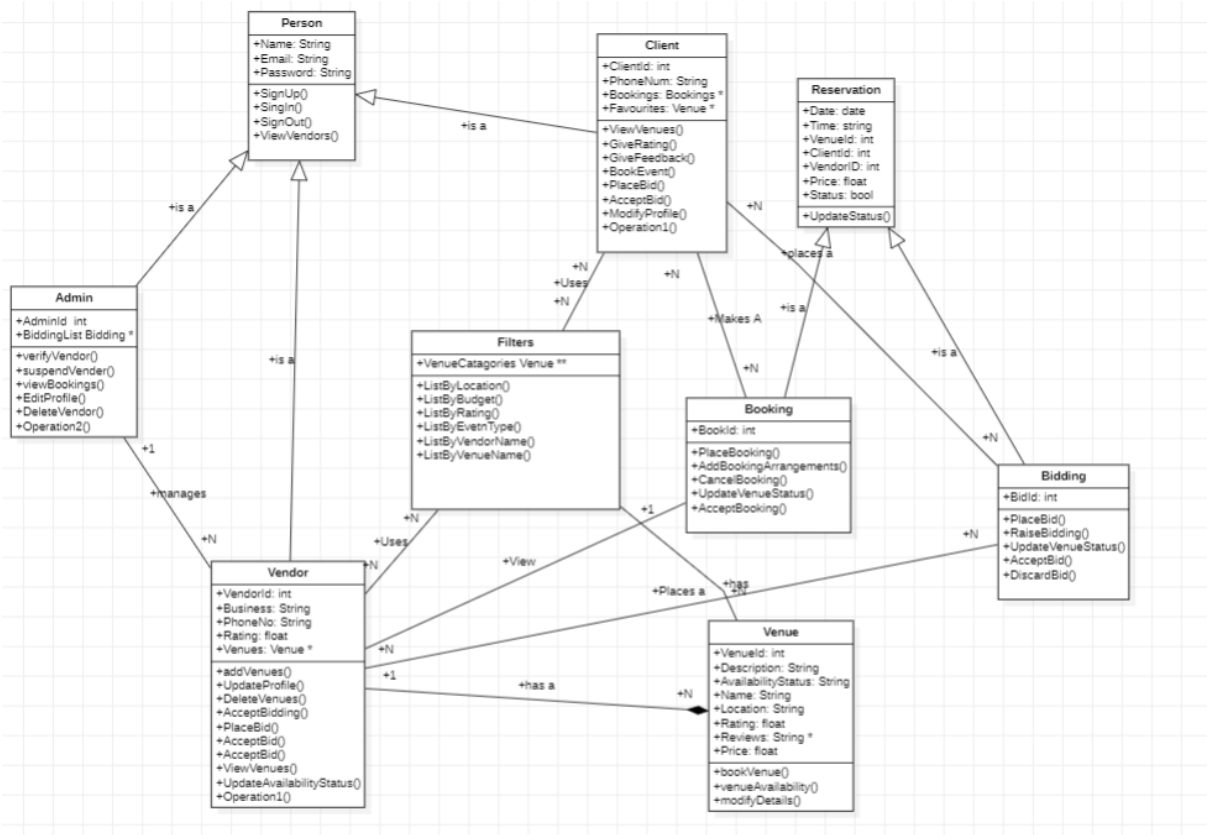


**Figure 13: Class Diagram**

*The figure represents the class diagram of the Event Organization System, including the classes and their relationships.*

## 5.6 Sequence Diagrams

### 5.6.1 Client Login

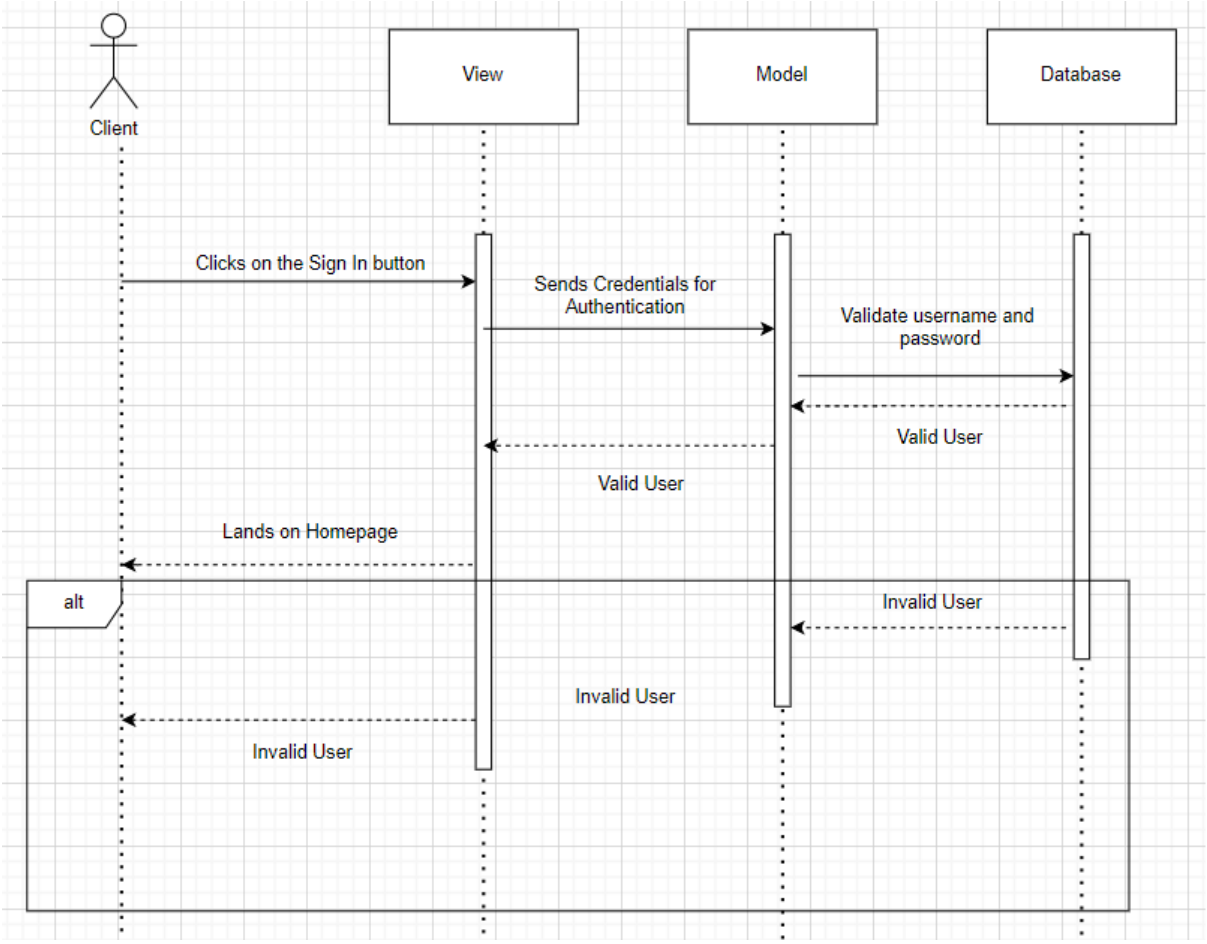


**Figure 14: Client Login Sequence Diagram**

*This figure represents the sequence diagram of client sign up process.*

## 5.6.2 Vendor Login

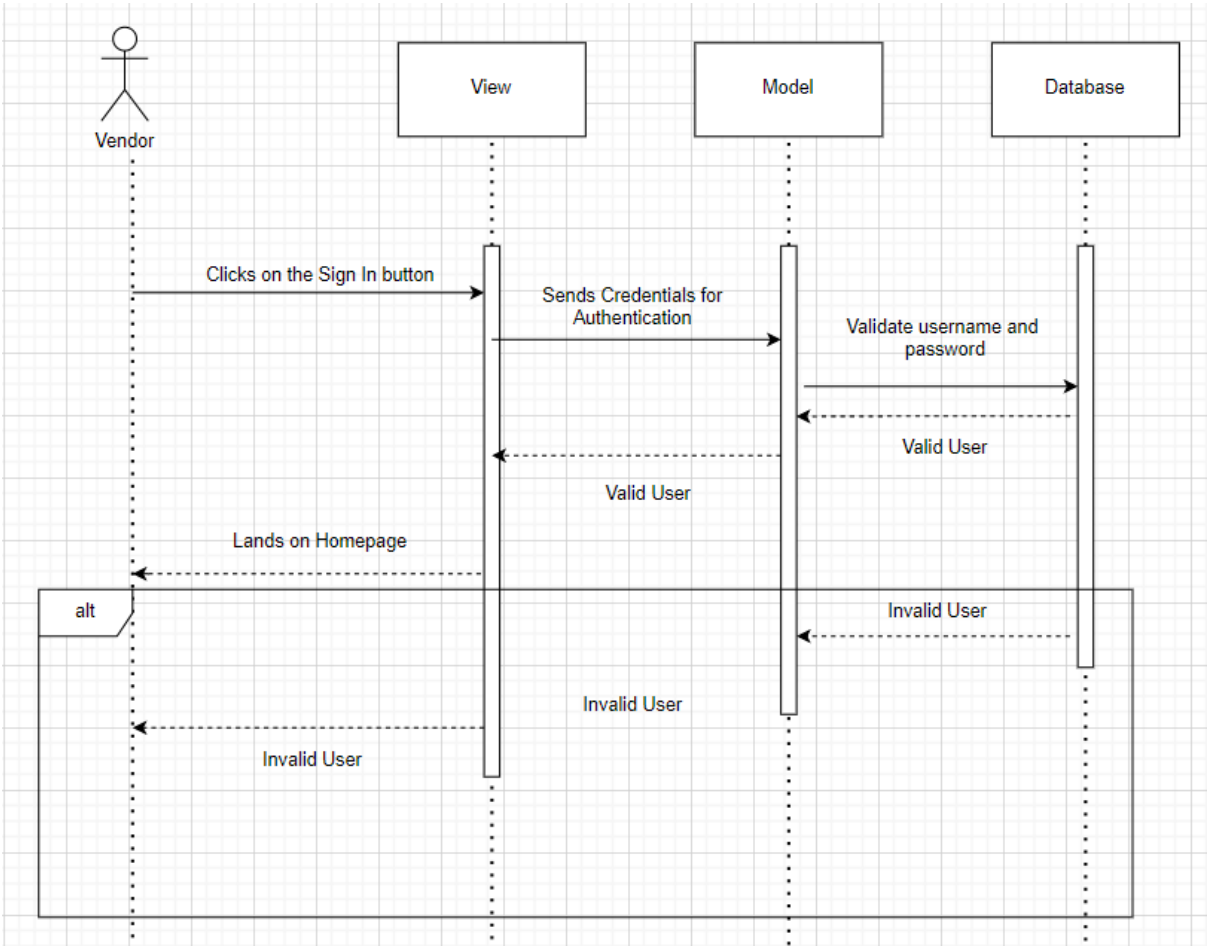


**Figure 15: Vendor Login Sequence Diagram**

*This figure represents the sequence diagram of vendor sign up process.*

### 5.6.3 Admin Login

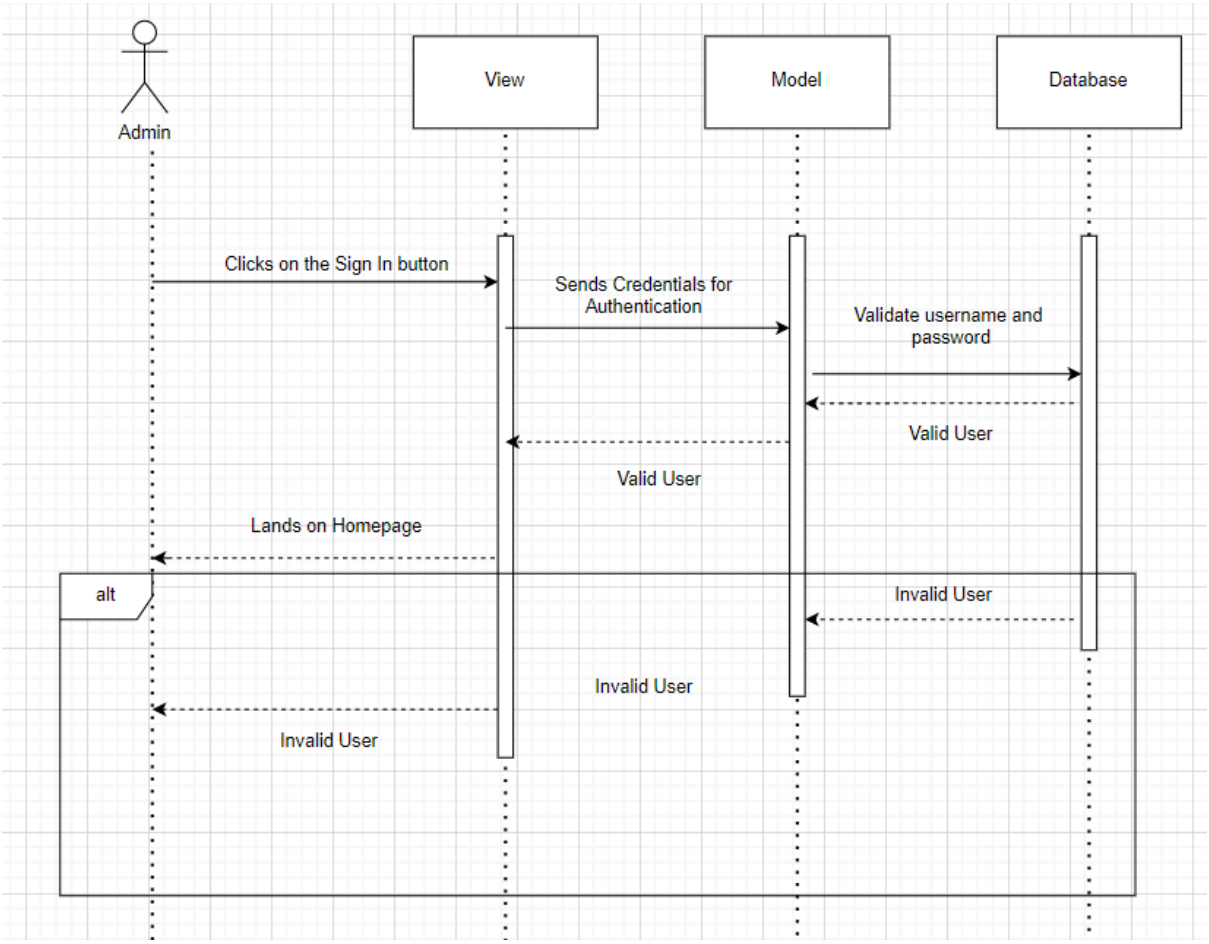


**Figure 16: Admin Login Sequence Diagram**

*This figure represents the sequence diagram of admin sign up process.*

## 5.6.4 Book an Event

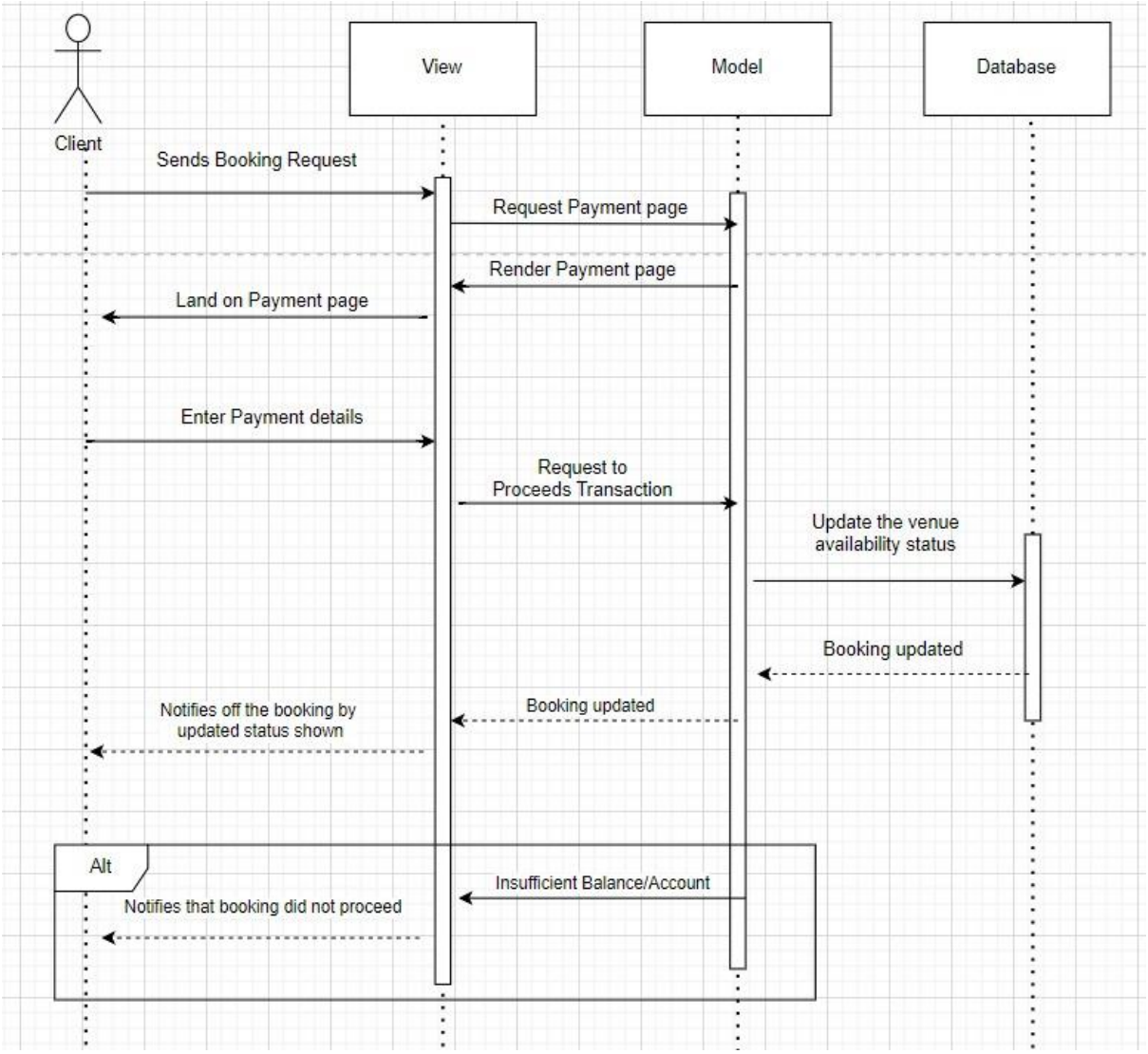


**Figure 17: Book an Event Sequence Diagram**

*This figure represents a sequence diagram of booking an event.*

## 5.6.5 Bid for Event

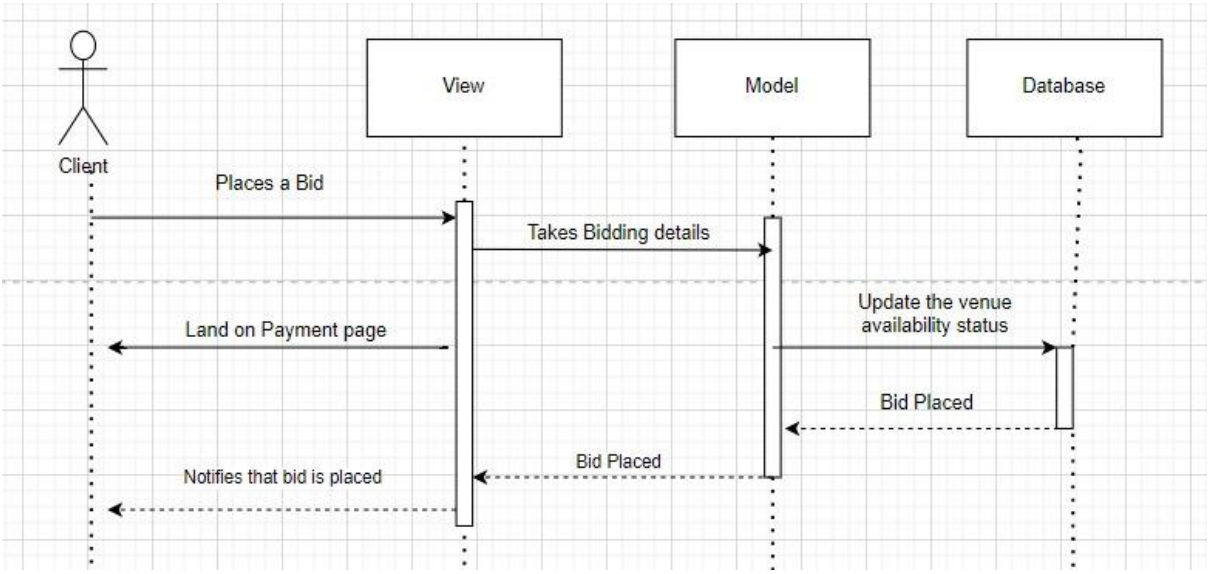


**Figure 18: Bid for Event Sequence Diagram**

*This figure represents the sequence diagram for bidding on an event.*

## 5.6.6 Client Sign Up

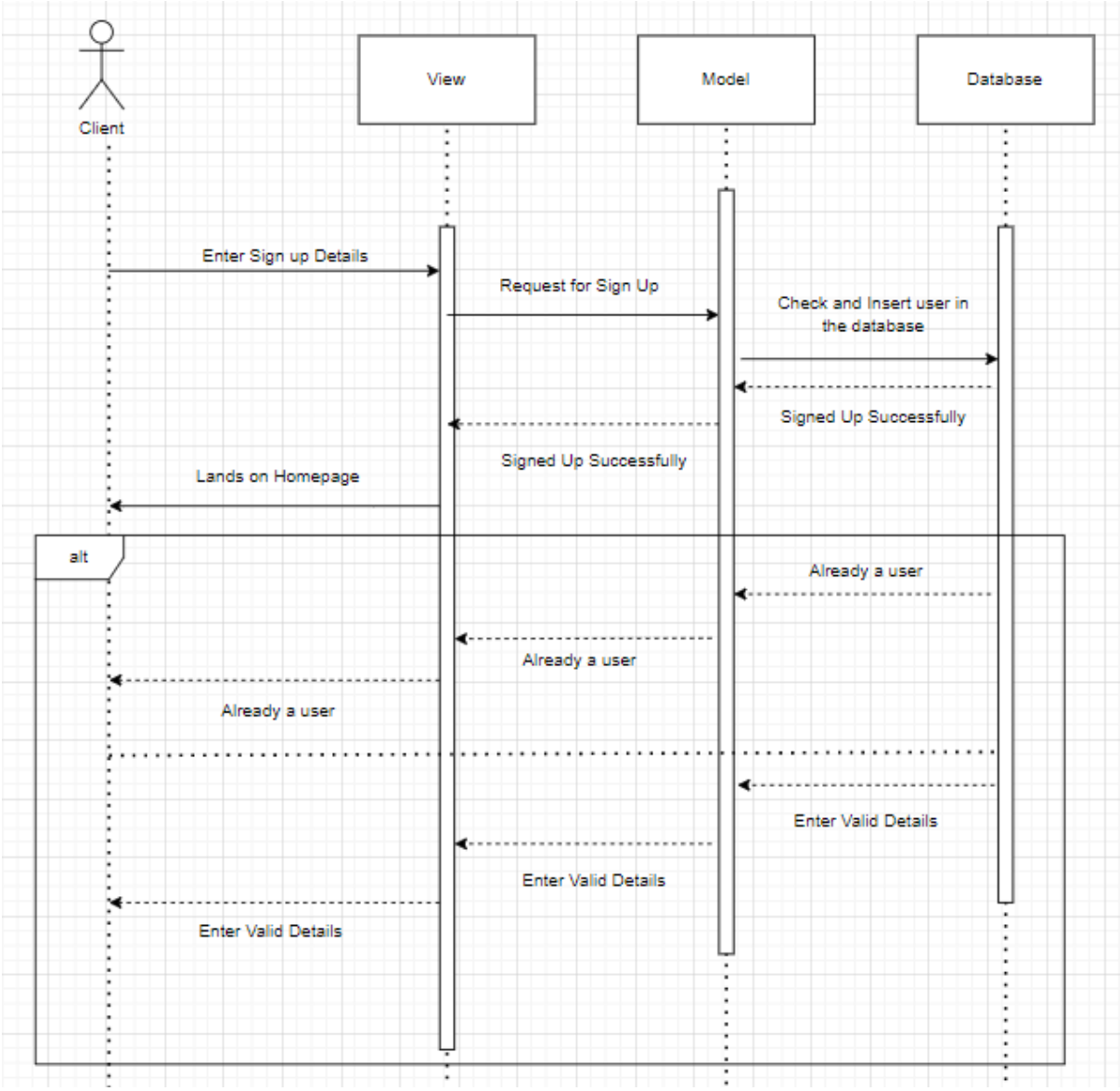


**Figure 19: Client Sign Up Sequence Diagram**

*This figure represents the sequence diagram for client sign up.*

## 5.6.7 Vendor Sign Up

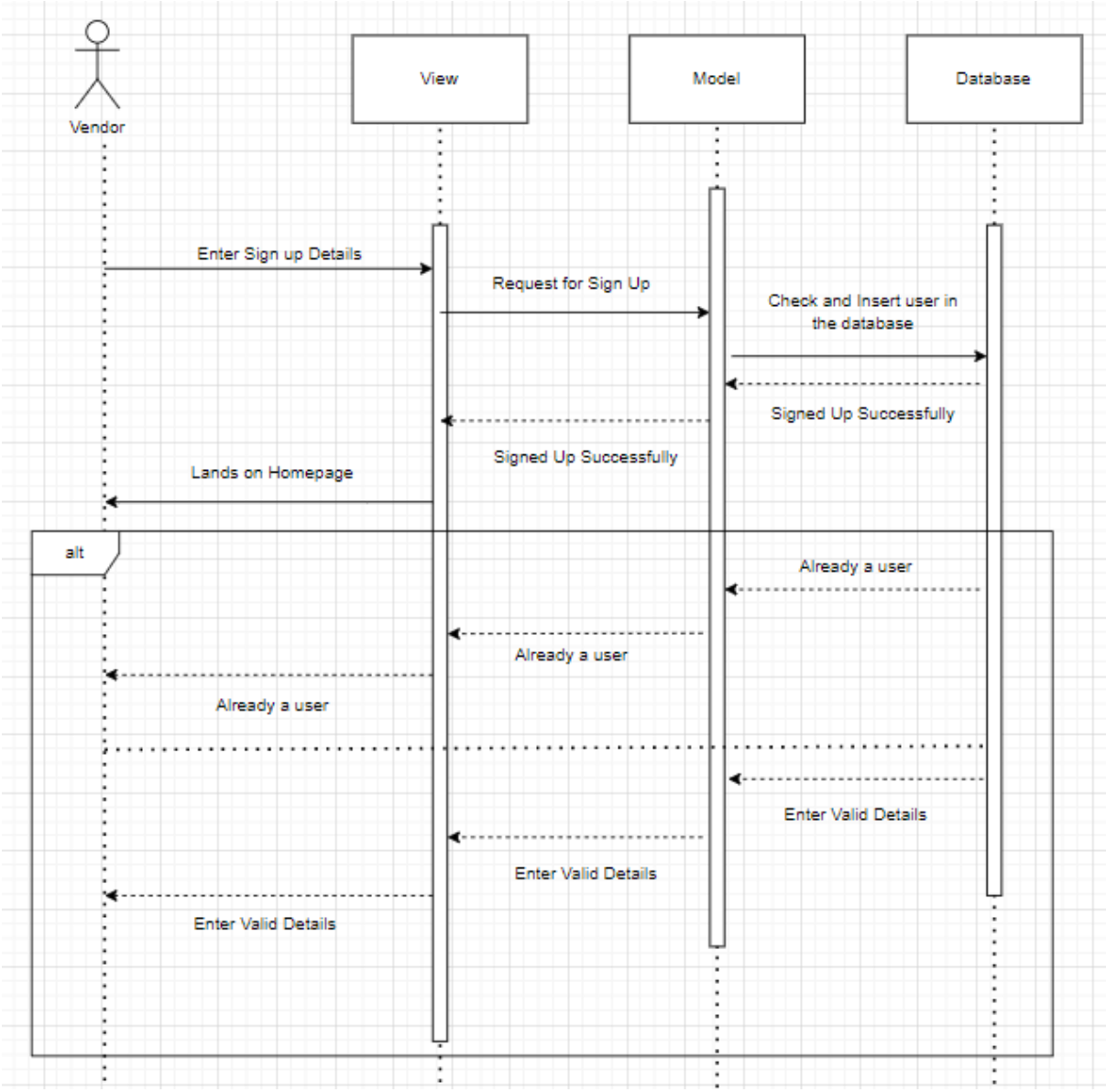


**Figure 20: Vendor Sign Up Sequence Diagram**

*This figure represents the sequence diagram of vendor sign up.*

## 5.6.8 Vendor Edit Profile

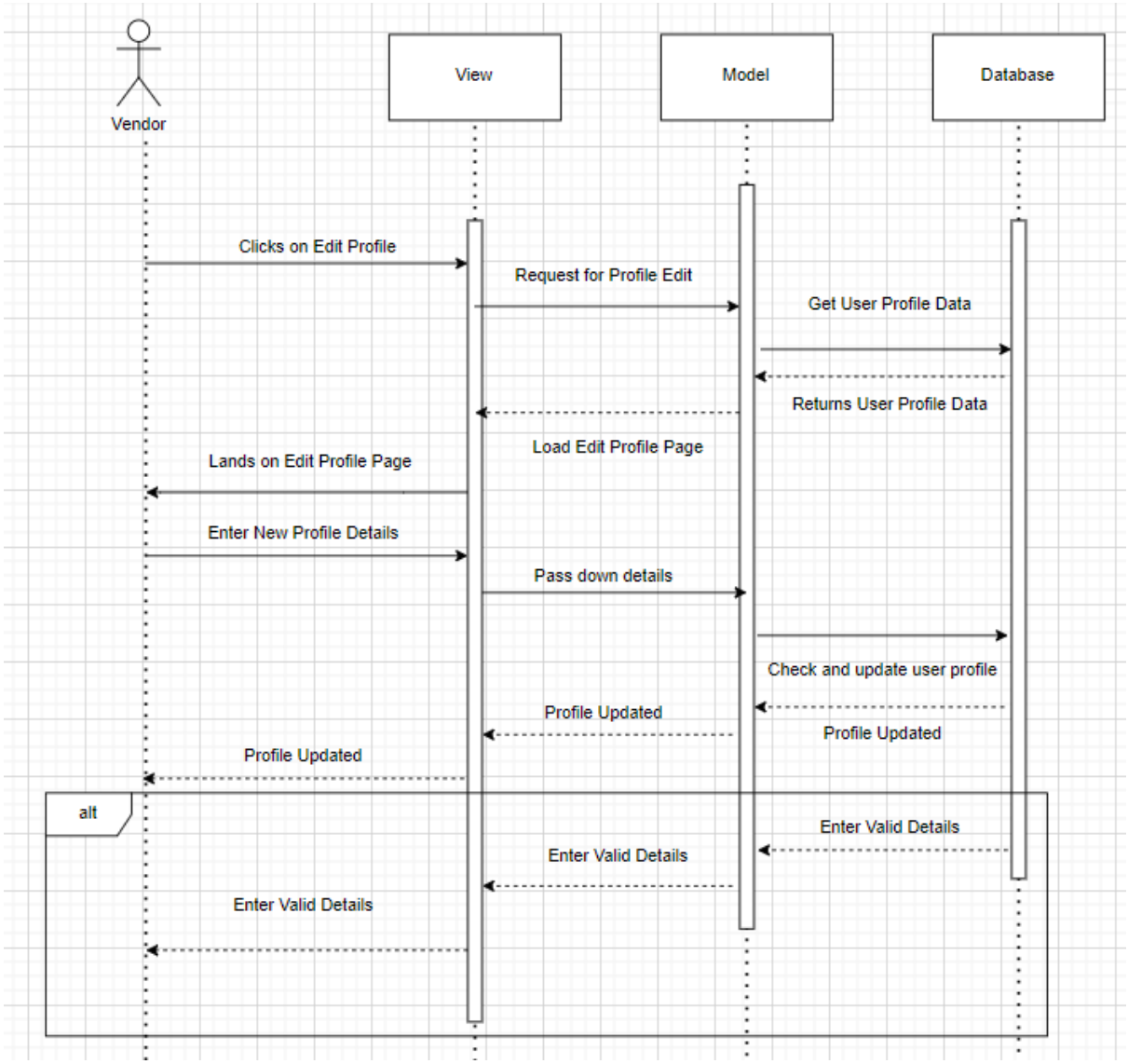


**Figure 21: Vendor Edit Profile Sequence Diagram**

*This figure represents the sequence diagram for vendor edit profile.*

## 5.6.9 Client Edit Profile

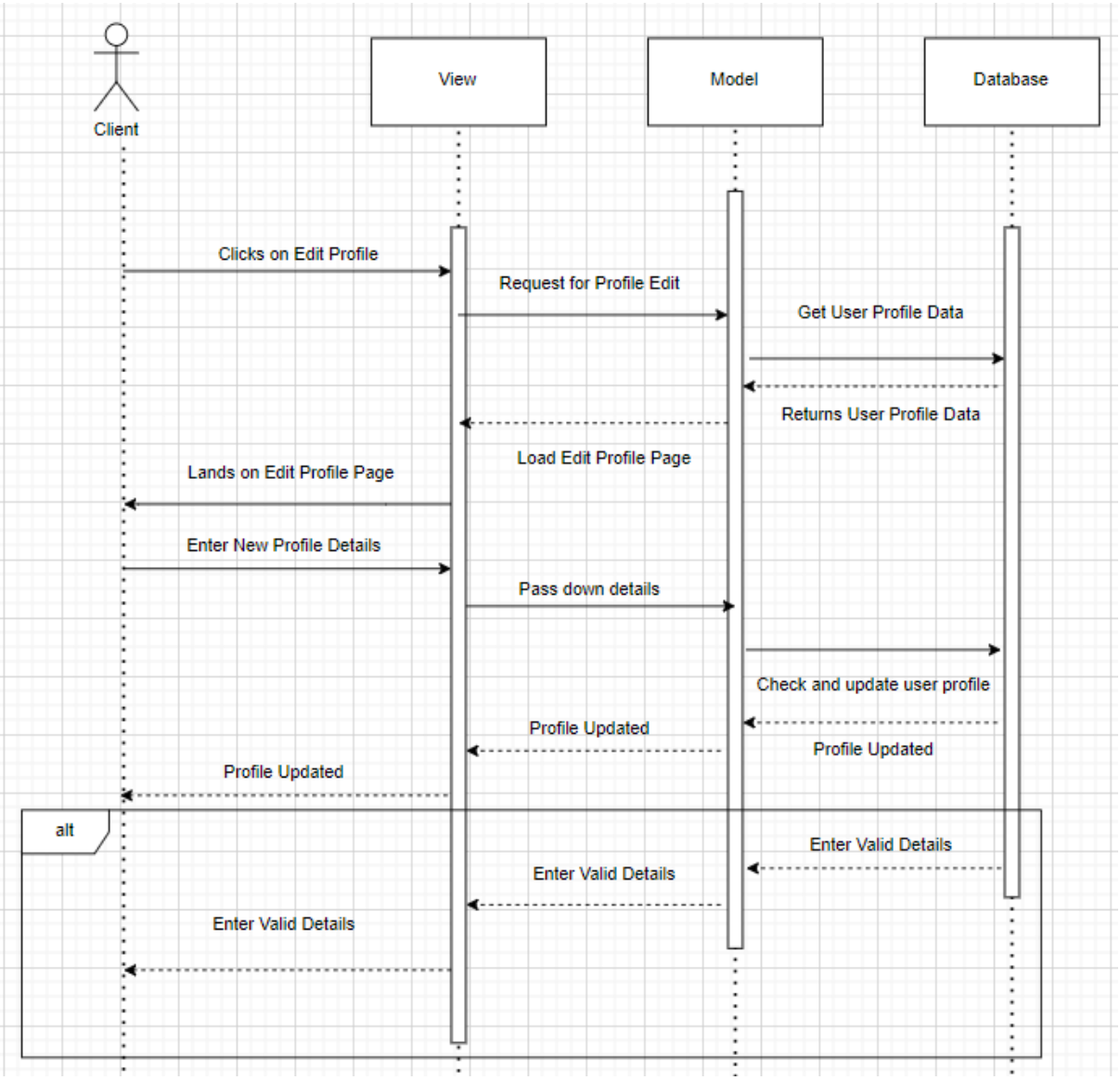


**Figure 22: Client Edit Profile Sequence Diagram**

*This figure represents the sequence diagram for vendor edit profile.*

## 5.6.10 Admin Edit Profile

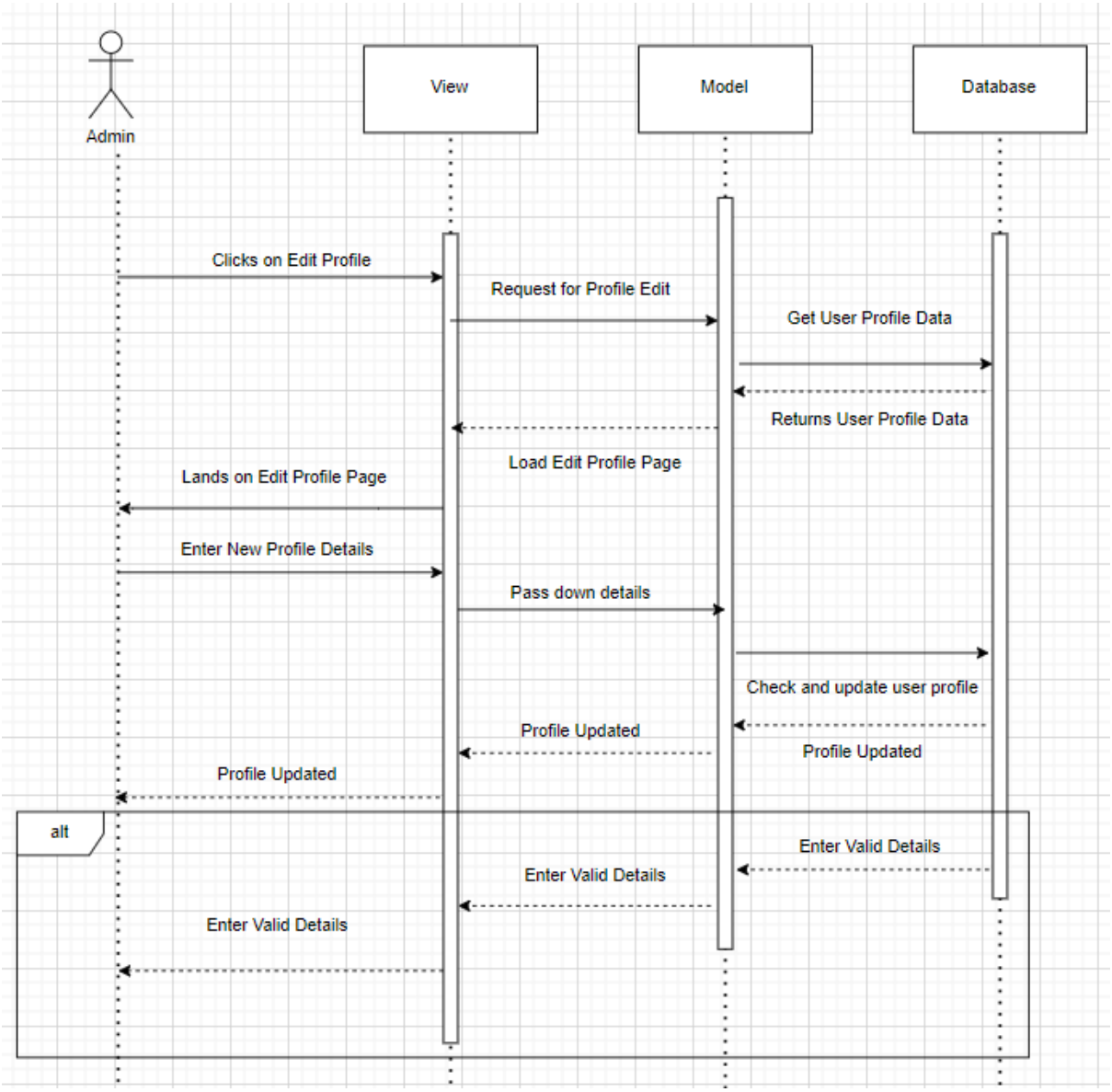


**Figure 23: Admin Edit Profile Sequence Diagram**

*This figure represents the sequence diagram of the admin edit profile.*

## 5.6.11 Suspend Vendor

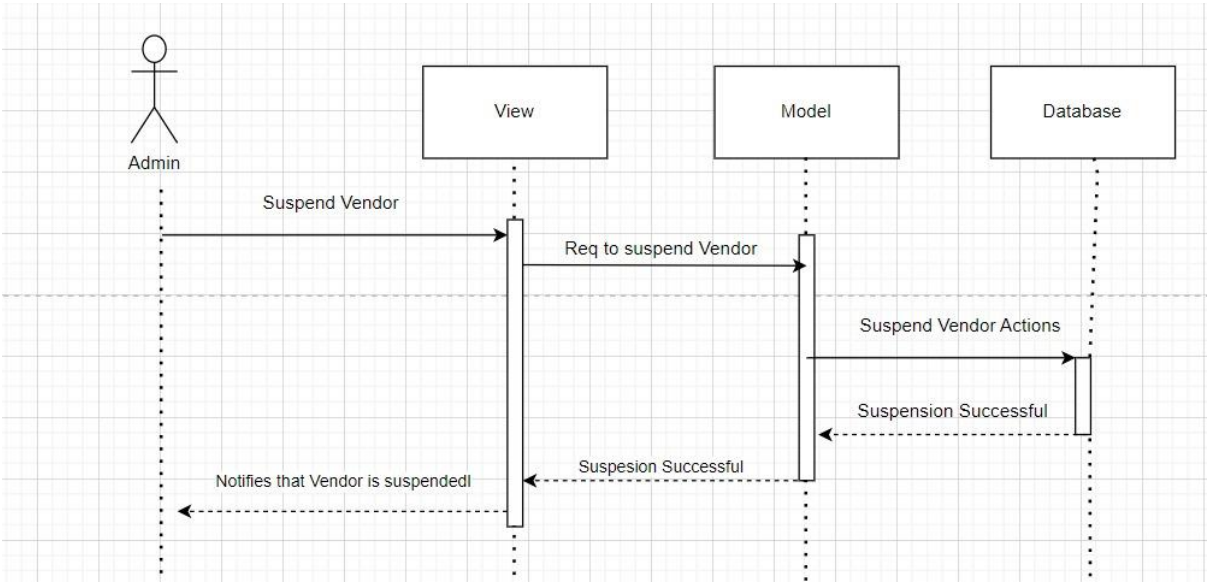


**Figure 24: Suspend Vendor Sequence Diagram**

*This figure represents the sequence diagram of the suspend vendor.*

## 5.6.12 Delete Vendor

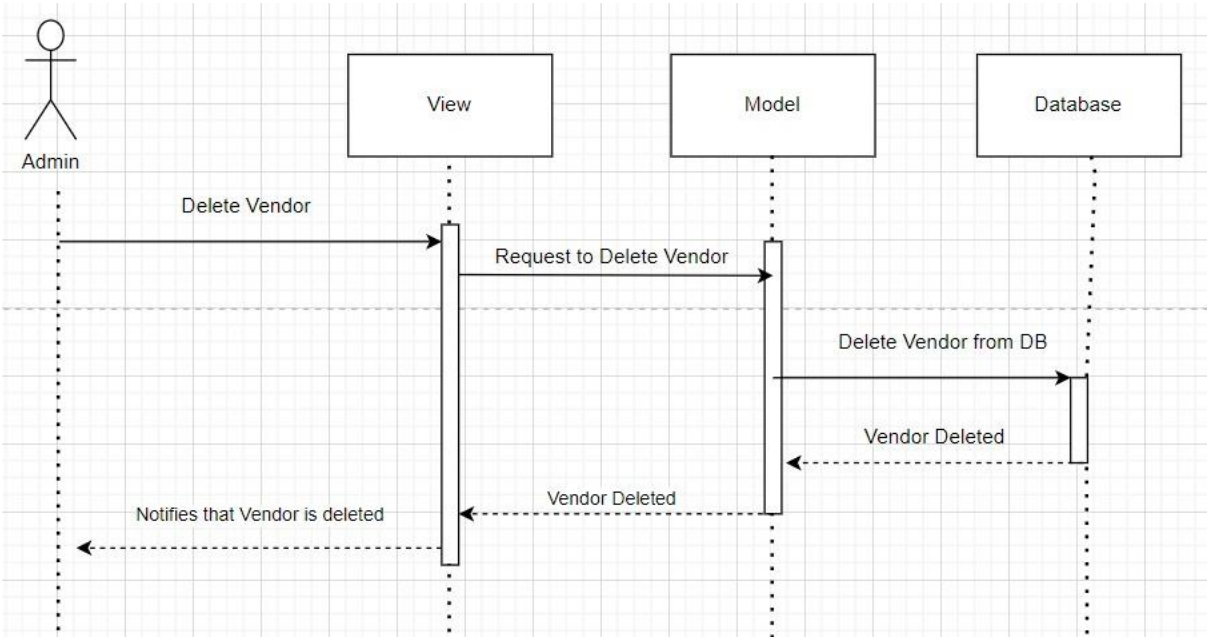


**Figure 25: Delete Vendor Sequence Diagram**

*This figure represents the sequence diagram of the delete vendor.*

## 5.6.13 Client Log Out

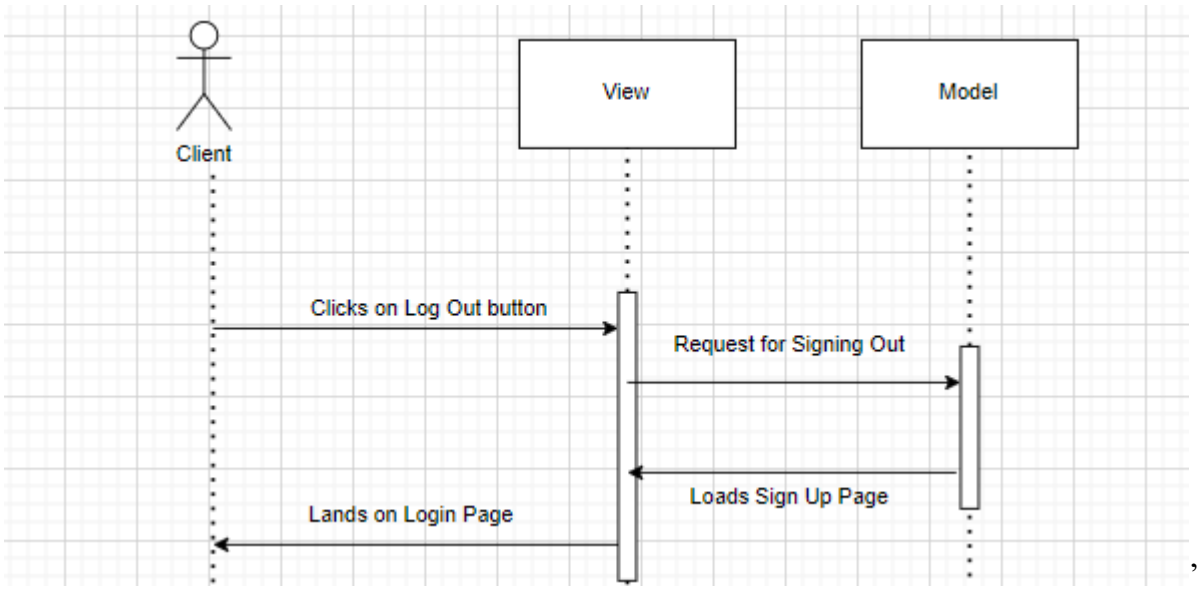


**Figure 26: Client Log Out Sequence Diagram**

*This figure represents the sequence diagram of client log out.*

## 5.6.14 Vendor Log Out

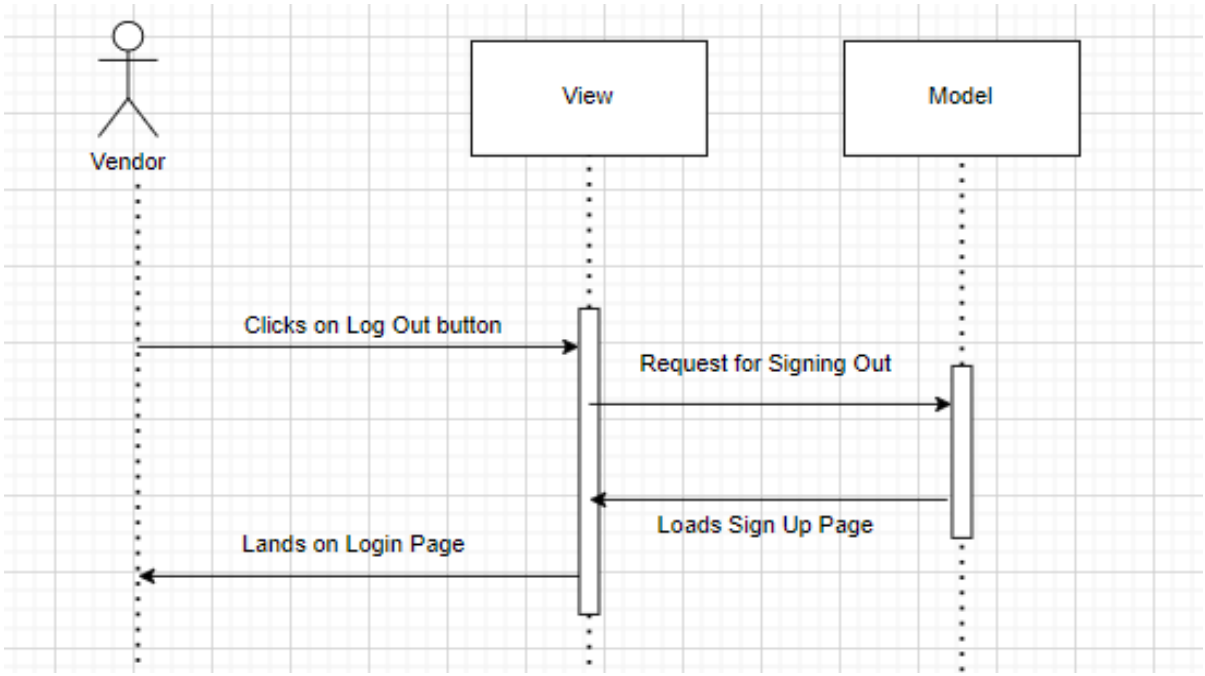


**Figure 27: Vendor Log Out Sequence Diagram**

*This figure represents the sequence diagram of vendor log out.*

## 5.6.15 Admin Log Out

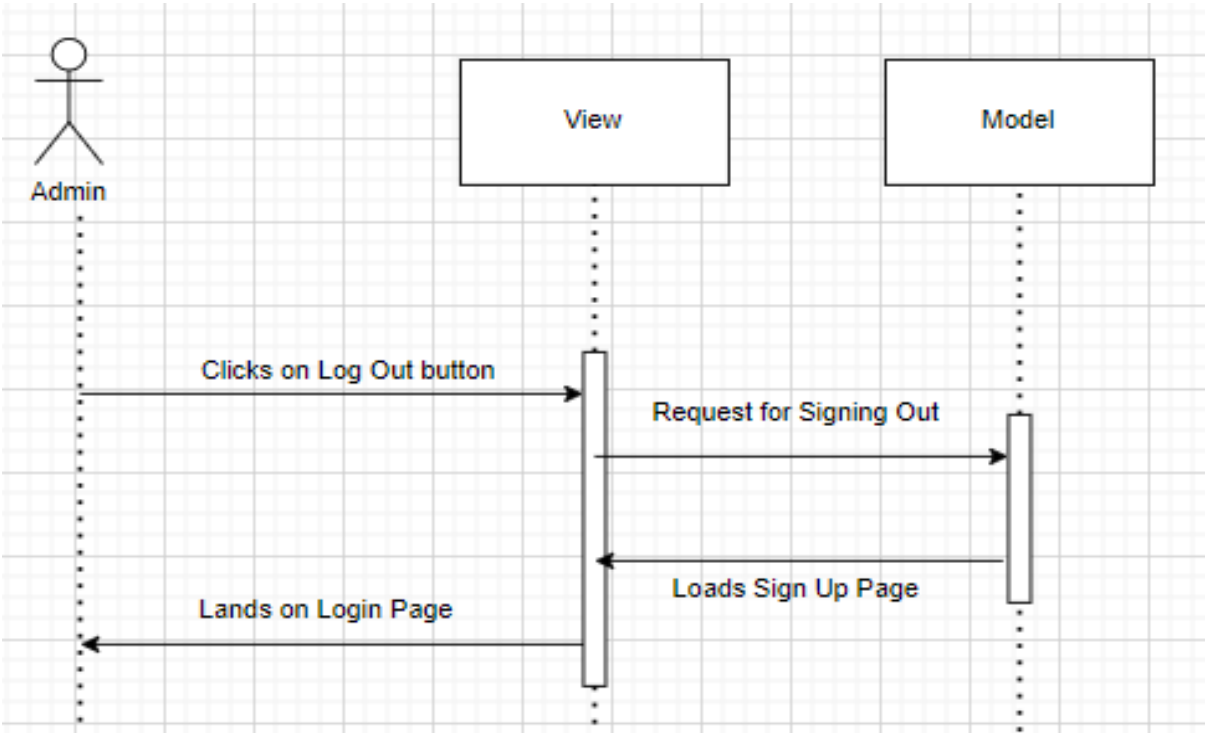


**Figure 28: Admin Log Out Sequence Diagram**

*This figure represents the sequence diagram of admin log out.*

## 5.6.16 Client Apply Filter(s)

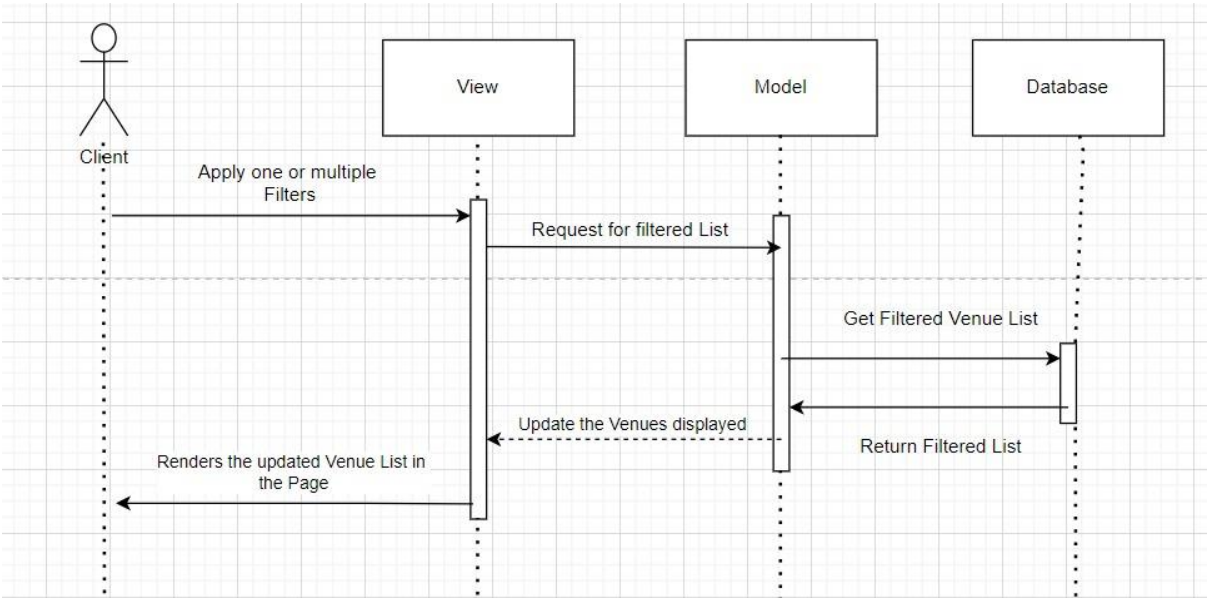


**Figure 29: Client Apply Filter(s) Sequence Diagram**

*This figure represents the sequence diagram for client apply filter(s).*

## 5.6.17 Vendor Apply Filter(s)

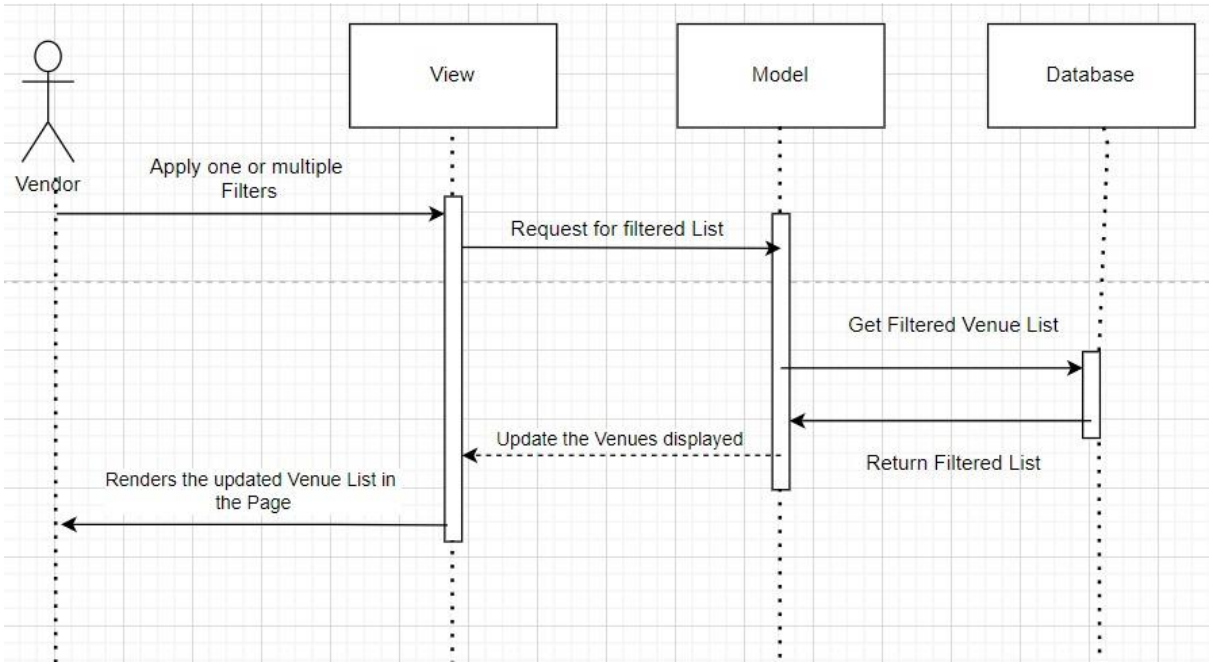


**Figure 30: Vendor Apply Filter(s) Sequence Diagram**

*This figure represents the sequence diagram of vendor apply filter(s).*

## 5.6.18 Give Rating to An Event

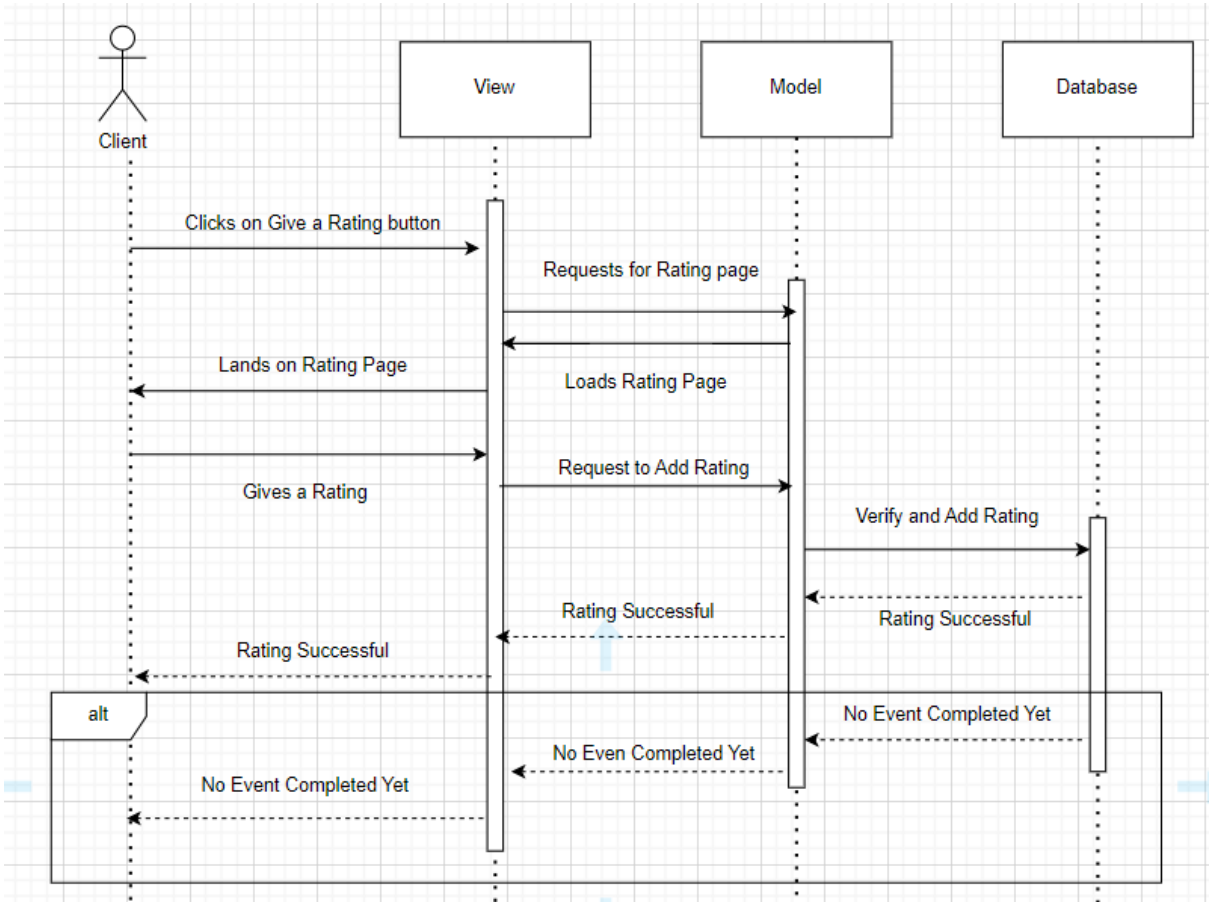


**Figure 31: Give Rating to An Event Sequence Diagram**

*This figure represents the sequence diagram of give rating to an event.*

## 5.6.19 Give Feedback to An Event

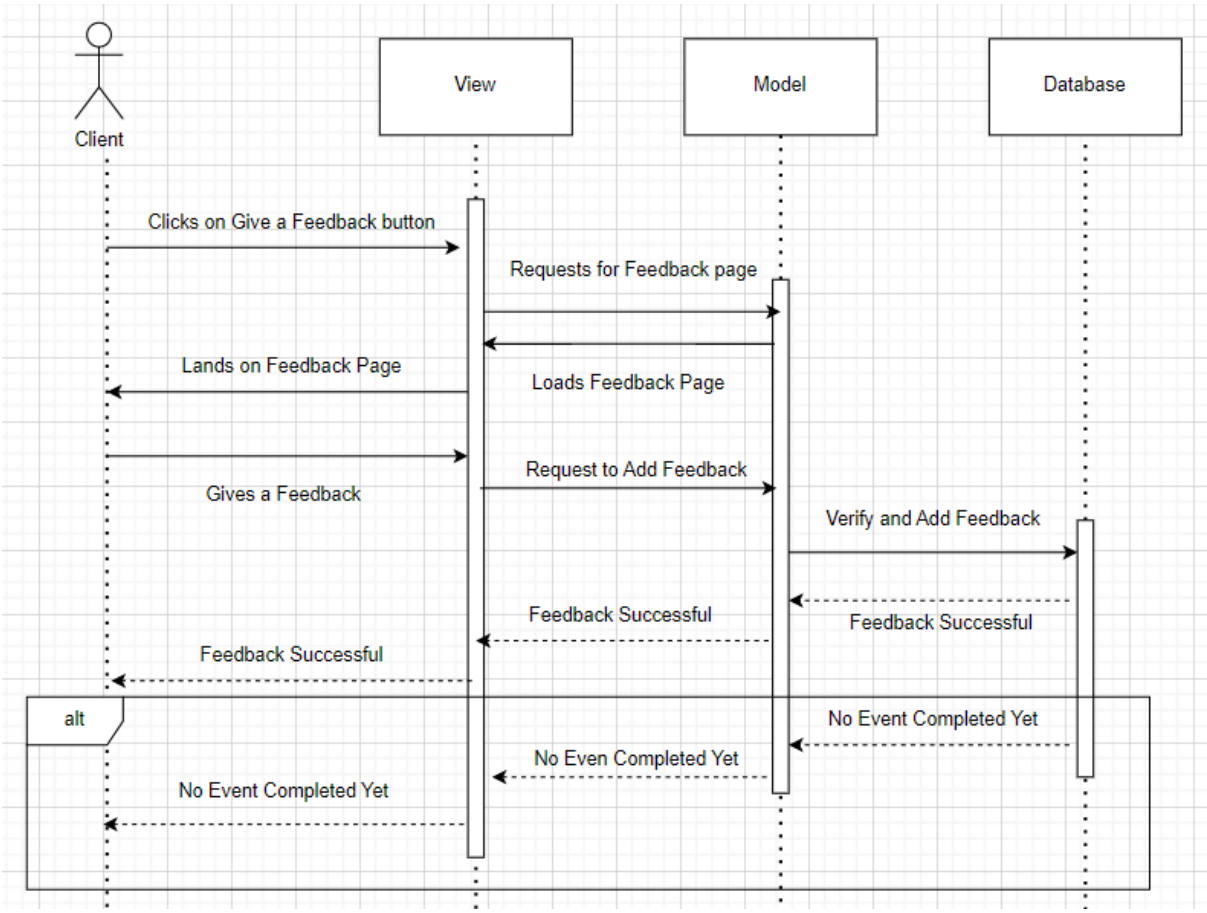


**Figure 32: Give a Feedback to An Event**

*This figure represents the sequence diagram of give feedback to an event*

## 5.6.20 Verify Vendor

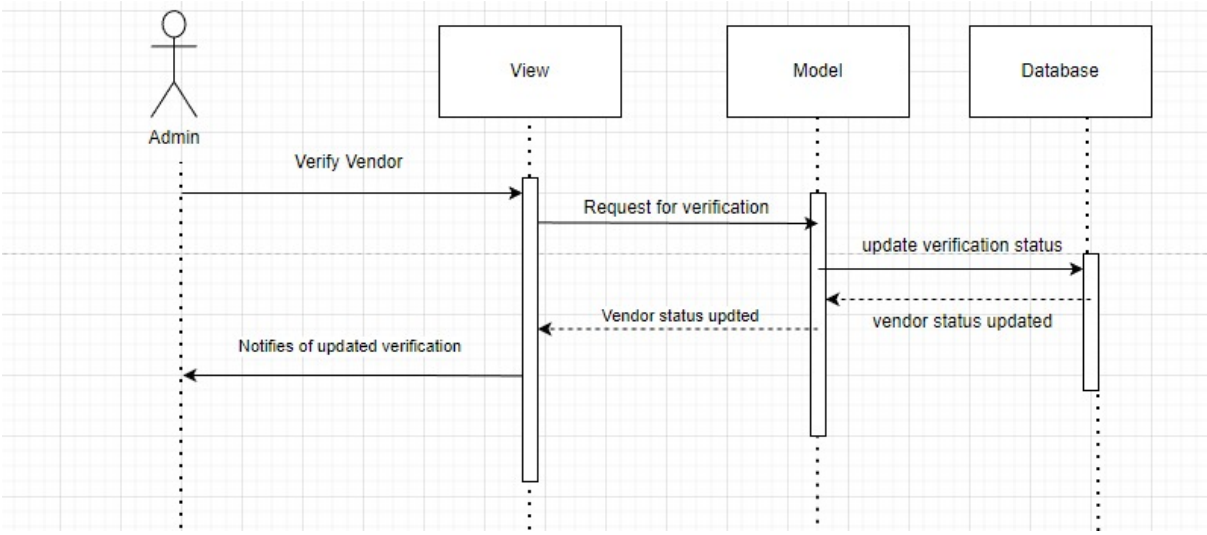


**Figure 33: Verify Vendor Sequence Diagram**

*This figure represents the sequence diagram of verify vendor.*

## 5.6.21 Vendor Cancel Booking

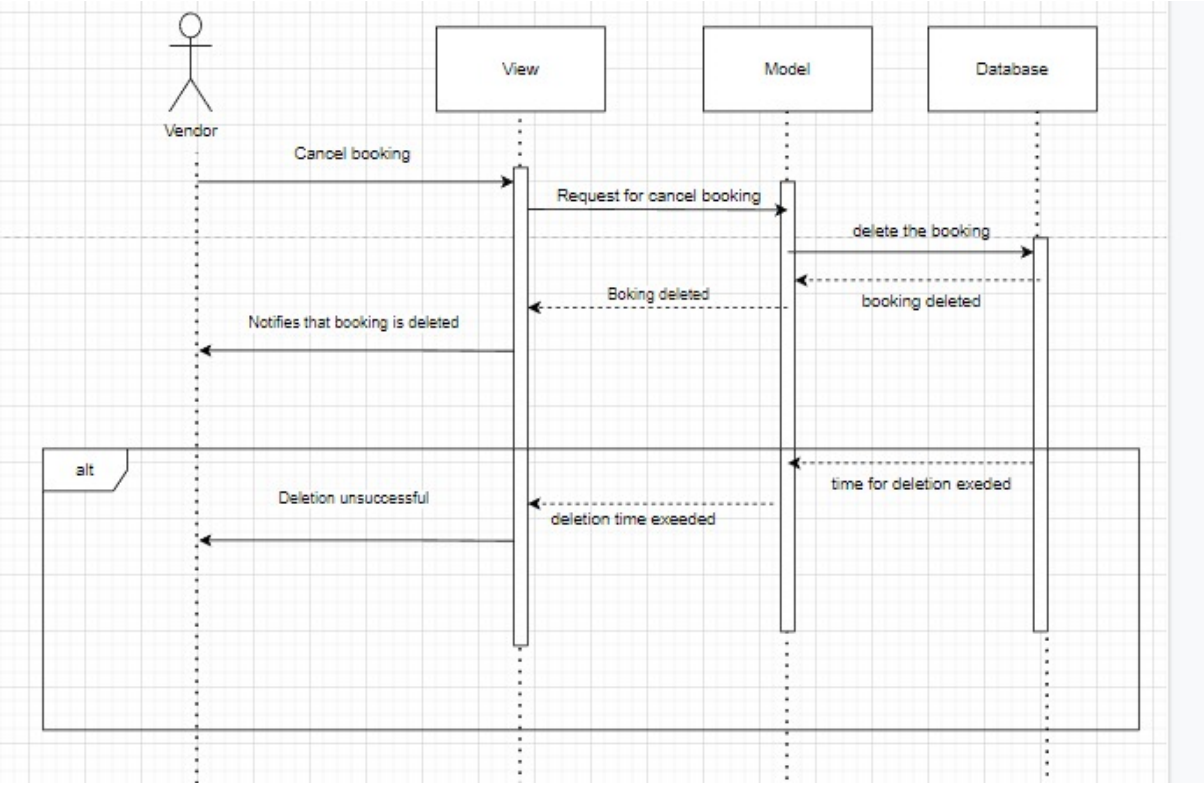


**Figure 34: Vendor Cancel Booking Sequence Diagram**

*This figure represents the sequence diagram of client cancel booking.*

## 5.6.22 Client Cancel Booking

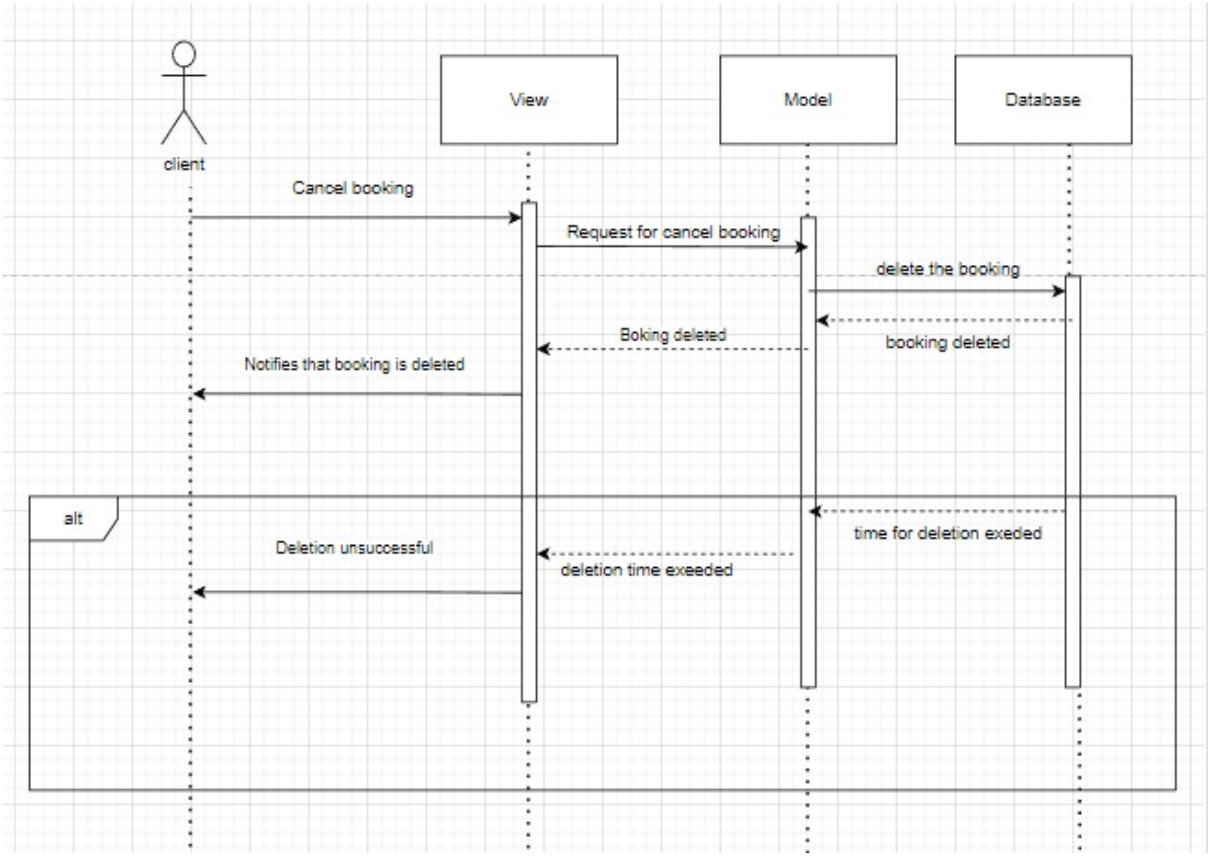


**Figure 35: Client Cancel Booking Sequence Diagram**

*This figure represents the sequence diagram of client cancel booking.*

## 5.6.23 Search for An Event

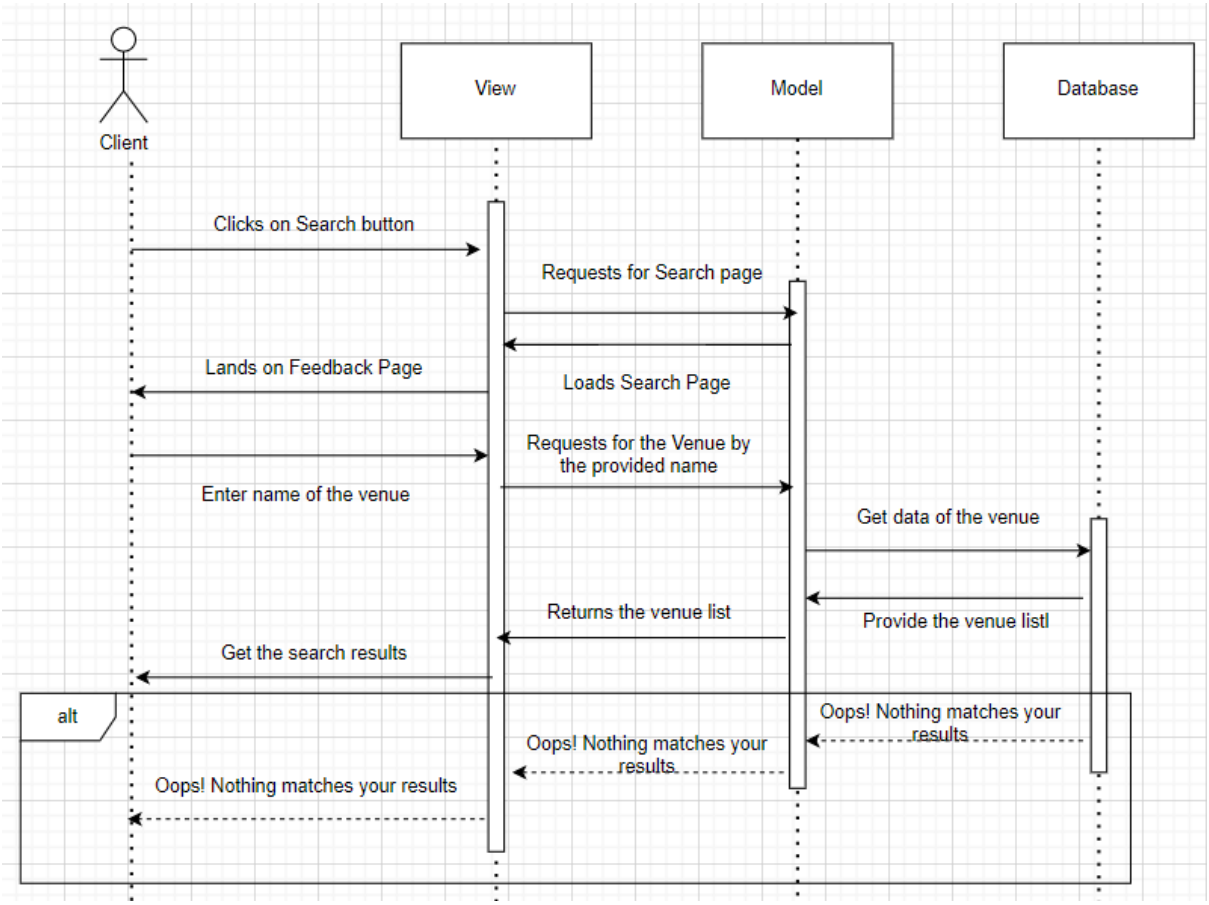


**Figure 36: Search for An Event Sequence Diagram**

*This figure represents the sequence diagram of search for an event.*

## 5.6.24 Client Alter Payment Method

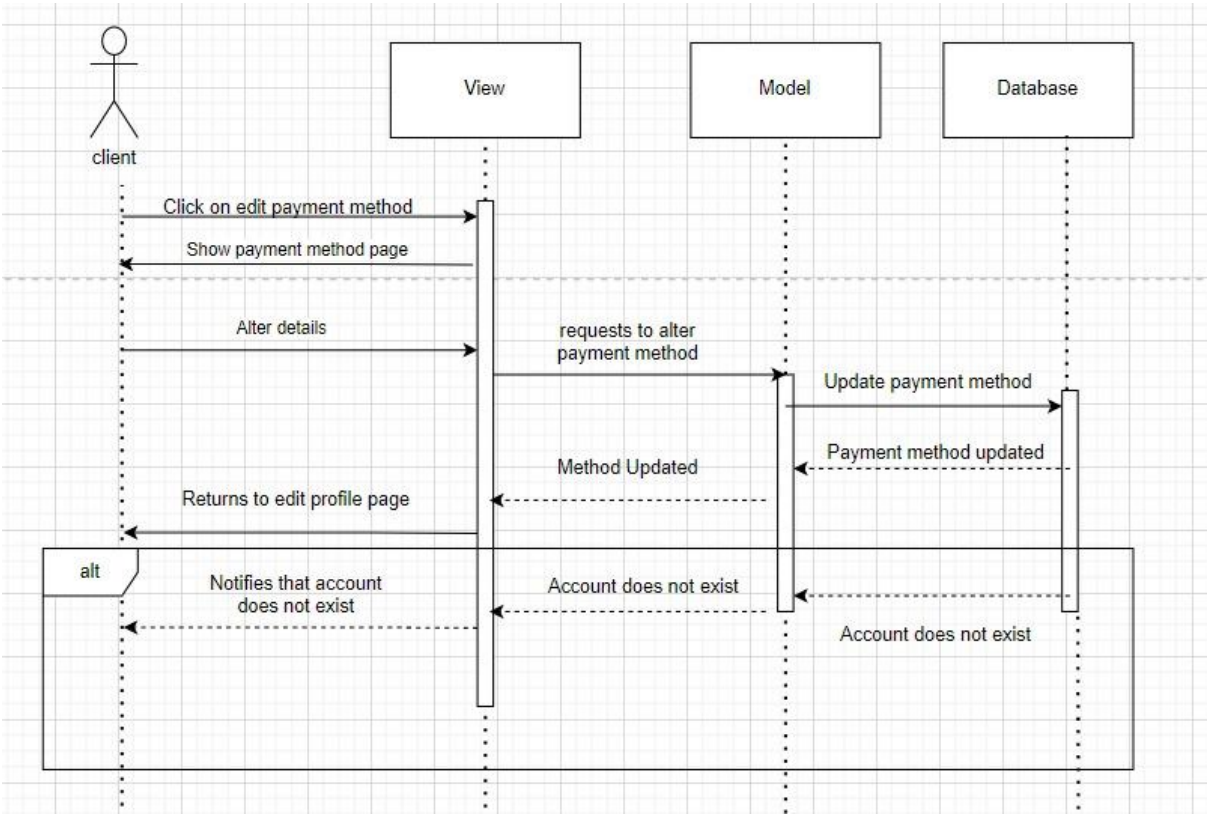


**Figure 37: Client Alter Payment Method Sequence Diagram**

*This figure represents the sequence diagram of the client alter payment method.*

## 5.6.25 Vendor Alter Payment Method

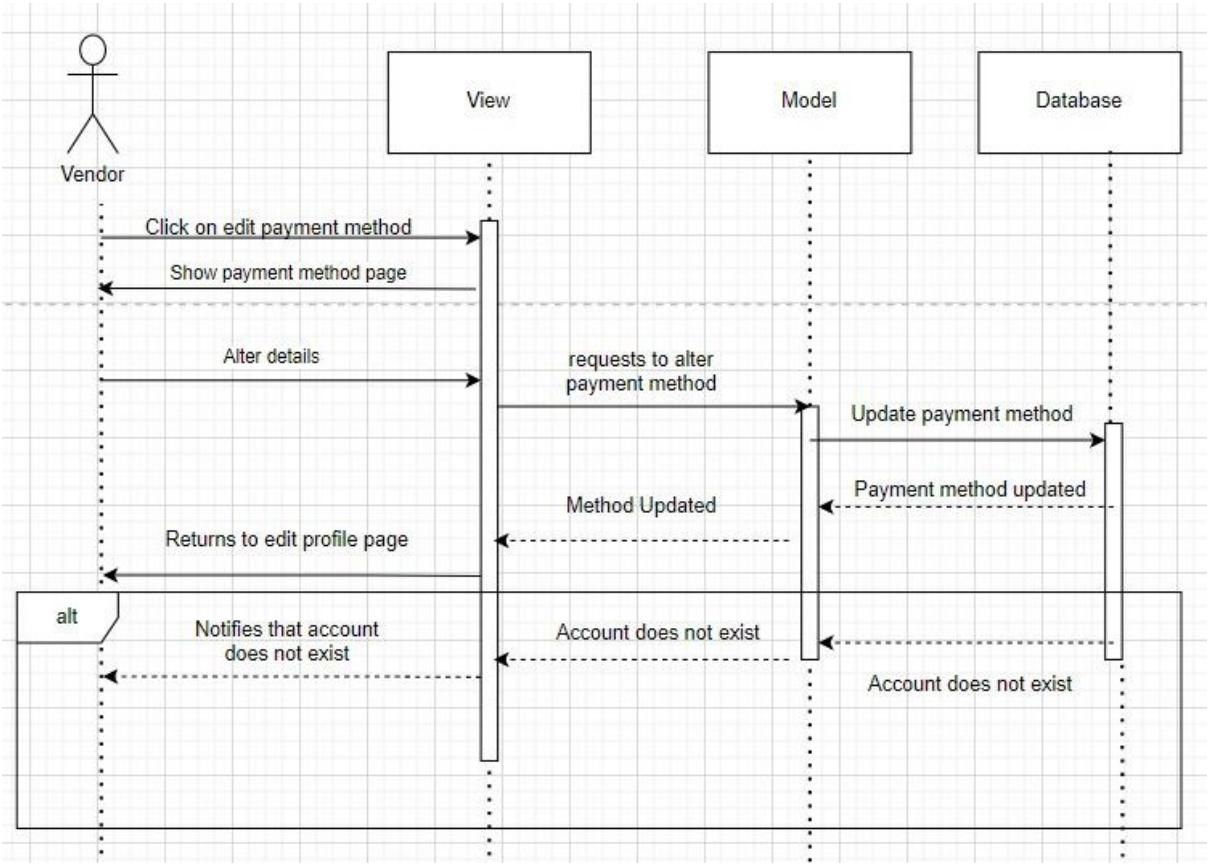


**Figure 38: Vendor Client Payment Method Sequence Diagram**

*This figure represents the sequence diagram of the vendor alter payment method.*

## 5.6.26 Add Event Venue to Favorite List

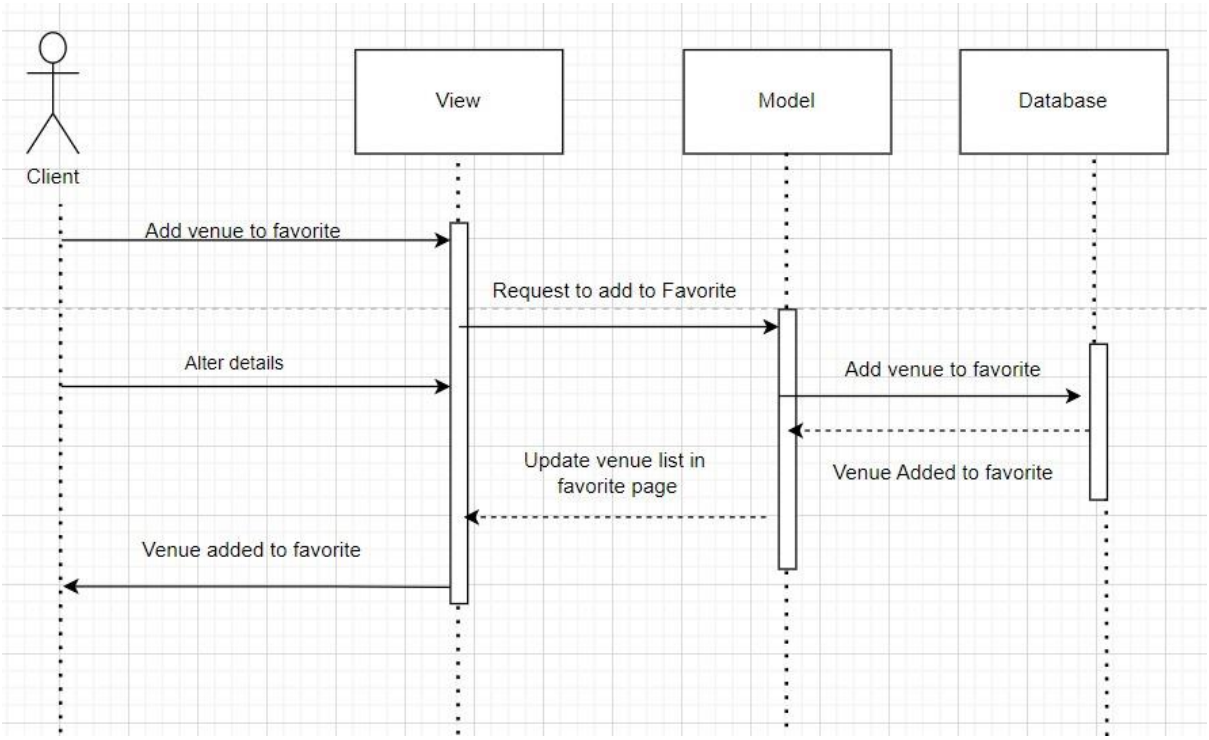

**Figure 39: Add Event Venue to Favorite List Sequence Diagram**

*This figure represents the sequence diagram of add event venue to favorite list.*

### 5.6.27 Remove Event Venue to Favorite List

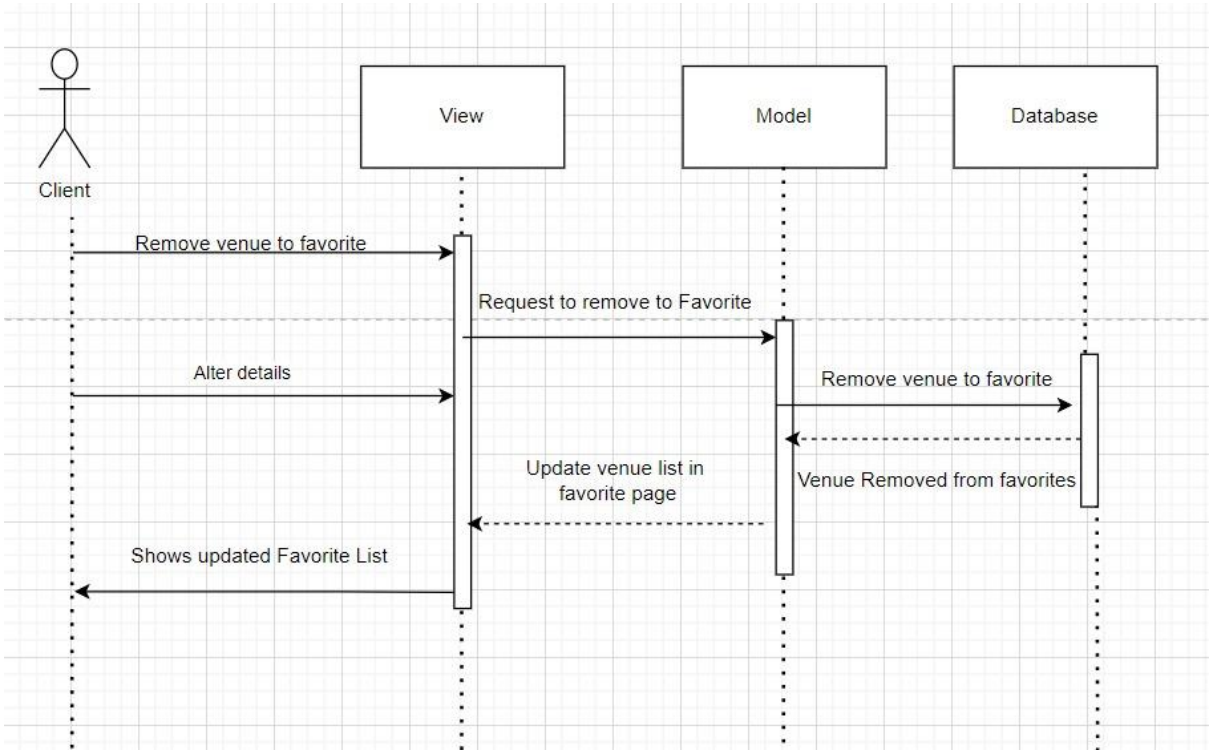


**Figure 40: Remove Event Venue to Favorite List Sequence Diagram**

*This figure represents the sequence diagram of remove event venue to favorite list.*

## 5.7 Policies and Tactics

### 5.7.1 Conventions

When developing the web application, it will be made sure abide by all standard JS practices. Each member of the team will use Prettier for linting and style to ensure that a unified font is presented. Visual Studio Code will be the IDE of choice since it is compact, has a built-in terminal, and supports JS out of the box.

### 5.7.2 Testing

White Box Testing technique will be used to test the web application. Since it is the most suitable testing technique that a developer can use to test and improve the design and usability, the application will be tested using this technique.

### 5.7.3 Data Collection

The data will be collected from clients, vendors, and admins whenever they register or perform a functionality requiring a use of any of the CRUD operations. Moreover, cookies will be maintained to keep track of the user session to provide stored settings local to the user.

### 5.7.4 Control Vulnerabilities

It is becoming more difficult to keep up with the constant stream of exploited vulnerabilities. Npm provides with the “npm audit” function used for a thorough scan of security flaws in the existing dependencies and alerts if any are present in the current release. The 'fix' command brings the version up to the most recently patched one.

### 5.7.5 Protocols Used

Event Organization System will have HTTP, JSON and FTP protocols used.

# References

[1] Schwaber K. (1997) SCRUM Development Process. In: Sutherland J., Casanave C., Miller J., Patel P., Hollowell G. (eds) Business Object Design and Implementation. Springer, London. https://doi.org/10.1007/978-1-4471-0947-1_11
[2] “MERN Stack,” https://www.geeksforgeeks.org/mern-stack, Oct 07, 2022.
[3] “Real Time Event Management,” http://www.douban.com, Oct 9, 2022.