

Syrian Private University
Faculty of Engineering
Department of Software
and Information Systems



Events and Activities planning platform

A senior 1 project report- submitted to complete the requirements for obtaining a bachelor's degree in Informatics Engineering - Software Engineering and Information Systems

Prepared by:

Helpest Mohamad Mohamad Marwa Ghassan Aljaouni Hiba Ahmad Shawa

Supervised by:

Eng. Anas Abdulziz

2025



الجامعة السورية الخاصة
SYRIAN PRIVATE UNIVERSITY

الجامعة السورية الخاصة
كلية الهندسة

قسم البرمجيات ونظم المعلومات

منصة لتخطيط الفعاليات والأنشطة

مشروع تخرج 1 - قدم استكمالاً لمتطلبات الحصول على درجة البكالوريوس في هندسة المعلوماتية
هندسة البرمجيات ونظم المعلومات

إعداد:

هليست محمد محمد

مروة غسان الجاعوني

هبة احمد شاوہ

إشراف:

المهندس أنس عبد العزيز

2025

SUPERVISION CERTIFICATION

Supervisor Certification

I Certify that the preparation of this project entitled

[Events And Activities Planning Platform]

Prepared by [Hiba Shawa& Marwa Aljaouni & Helpest Mohamad] was made under my supervision at Faculty of Informatics Engineering in partial Fulfillment of the Requirements for the Degree of Bachelor of Software and Information system Engineering

Name: Eng.Anas Abdulaziz

Signature:.....

Date:

Abstract :

In a world where we increasingly rely on technology to simplify our daily lives, the need for integrated digital platforms to manage events and activities has become essential. Today, the events sector faces significant challenges due to fragmented operations among multiple parties. Attendees struggle to discover and book events across scattered platforms, organizers face complexities in managing bookings and coordinating with service providers, and suppliers are forced to invest heavily in marketing without guaranteed access to their target customers.

This project aims to develop a comprehensive online platform that brings together attendees, organizers, and suppliers in a unified digital marketplace. The platform provides a seamless experience for secure booking with automatic notifications, enables organizers to manage their events through a dashboard that offers real-time analytics, and allows suppliers to connect directly with customers through searchable profiles.

Using modern web technologies, the platform seeks to transform the traditional event booking model into a dynamic marketplace that enhances effective collaboration, reduces costs, and improves the overall experience

الملخص:

في عالم يتزايد فيه الاعتماد على التكنولوجيا لتسهيل حياتنا اليومية، أصبحت الحاجة إلى منصات رقمية متكاملة لإدارة الفعاليات والأنشطة أمراً ضرورياً. يواجه قطاع الفعاليات اليوم تحديات كبيرة تتمثل في تشتت العمليات بين أطراف متعددة، حيث يجد الحضور صعوبة في اكتشاف وحجز الفعاليات عبر منصات متفرقة، بينما يواجه المنظمون تعقيدات في إدارة الحجوزات والتنسيق مع مزودي الخدمات، ويضطر الموردون للاستثمار بكثافة في التسويق دون ضمان الوصول للعملاء المستهدفين.

يهدف هذا المشروع إلى تطوير منصة إلكترونية شاملة تجمع الحضور والمنظمين والموردين في سوق رقمي موحد. توفر المنصة تجربة سلسة للحجز الآمن مع إشعارات تلقائية، وتمكن المنظمين من إدارة فعالياتهم عبر لوحة تحكم توفر تحليلات فورية، كما تتيح للموردين التواصل المباشر مع العملاء من خلال ملفات تعريفية قابلة للبحث.

باستخدام تقنيات الوب الحديثة، تسعى المنصة لتحويل نموذج حجز الفعاليات التقليدي إلى سوق ديناميكي يعزز التعاون الفعال ويقلل التكاليف ويسهل التجربة العامة

List Of Contents:

Contents

SUPERVISION CERTIFICATION	3
الملخص:	5
List Of Contents:	6
List Of Tables:	9
List Of Figures:	10
Chapter 1: Introduction	14
1.Introduction	15
2-problem Definition	Error! Bookmark not defined.2
3.project Objectives	Error! Bookmark not defined.2
4.Report Organization	16
5-Summary	17
Chapter 2:Reference Study	18
1.Introduction:	19
2. Similar Applications:	19
1. SolarEdge:	19
2. Enphase Enlighten:	20
3. SMA Sunny Portal:	21
4. Tigo Energy:	Error! Bookmark not defined.2
5. KACO new energy:	Error! Bookmark not defined.2
3. Our System :	24
4.Comparison of Systems	25
5.Summary	Error! Bookmark not defined.2
Chapter 3: Project Management	27
1.Project Charter	28
2.Methodology:	30
3.Statement of work:	34
4.Project Timeline	36
5.Gantt Chart	38

6.Risk management:	39
7. Project Development Life Cycle Model:	41
8. Hybrid Agile Project Management:	41
9.Initial Requirement Trackability Matrix (RTM)(V.1)	42
10.Summary:	43
Chapter 4:System Analysis	Error! Bookmark not defined.
1.System Initial Analysis	Error! Bookmark not defined.
1. Introduction.....	Error! Bookmark not defined.
2.Overall Description	Error! Bookmark not defined.
3.High level requirements	Error! Bookmark not defined.
4.Use case diagram:	Error! Bookmark not defined.
4.Activity diagram:.....	Error! Bookmark not defined.
2.Detailed Analysis	Error! Bookmark not defined.
1.Introduction	Error! Bookmark not defined.
2.purposes	Error! Bookmark not defined.
3.System Requirements:	Error! Bookmark not defined.
3.initial test case:.....	Error! Bookmark not defined.
4. Requirement Trackability Matrix (RTM) (V.2)	Error! Bookmark not defined.
5.Summary	Error! Bookmark not defined.
Chapter 5: System Design	118
1.System Initial Design	119
1.Selected Architecture: Layered Architecture with Modularized Services.....	119
2.System Architecture:	121
3.Class diagram:	122
4.Sequence diagram.....	Error! Bookmark not defined.
2.Detailed Design	122
1.Introduction:	123
2.Purpose :	123
3.functional Requirements:	124
4.System Class Diagram	146
5.ERD - Entity-Relationship Diagram	Error! Bookmark not defined.
3.Requirement Trackability Matrix (RTM) (V.3)	154
7.Summary	161

Chapter6: Implementation	162
1.Introduction	163
2.Purpose:.....	163
3.User Interfaces.....	163
4. REST API Documentation – Alternative Energy Management System	179
4.1.Introduction	179
4.2.Purpose.....	179
4.3.API Structure Overview.....	179
4.4.Detailed Table of REST API Endpoints	187
5.Execution Test case.....	193
6. Requirement Trackability Matrix (RTM) (V.4)	196
7.Summary:	204

List Of Tables:

Table 1:Eventbrite	Error! Bookmark not defined.	2
Table 2:Cvent.....	Error! Bookmark not defined.	2
Table 3:Bizzabo	Error! Bookmark not defined.	2
Table 4:Splash	Error! Bookmark not defined.	2
Table 5: Faaliyeh App_Syria	Error! Bookmark not defined.	2
Table 6: comparison of systems	Error! Bookmark not defined.	
Table 7:project charter info		28
Table 8: roles and responsibilities.....		29
Table 9:schedules		36
Table 10:timeline		37
Table 11:risk management.....		40
Table 12:RTM(V.1).....		42
Table 13:high level requirements	Error! Bookmark not defined.	
Table 14:VM Functional Requirements.....	Error! Bookmark not defined.	
Table 15:use case specification 1	Error! Bookmark not defined.	
Table 16: use case specification 2	Error! Bookmark not defined.	
Table 17: use case specification 3	Error! Bookmark not defined.	
Table 18: UFR Functional Requirements	Error! Bookmark not defined.	
Table 19 :use case specification 4	Error! Bookmark not defined.	
Table 20: use case specification 5	Error! Bookmark not defined.	
Table 21:use case specification 6	Error! Bookmark not defined.	
Table 22:use case specification 7	Error! Bookmark not defined.	
Table 23:use case specification 8	Error! Bookmark not defined.	
Table 24:use case specification 9	Error! Bookmark not defined.	
Table 25: use case specification 10	Error! Bookmark not defined.	
Table 26: CPEP Functional Requirements	Error! Bookmark not defined.	
Table 27: use case specification 11	Error! Bookmark not defined.	
Table 28: use case specification 12	Error! Bookmark not defined.	
Table 29: use case specification 13	Error! Bookmark not defined.	
Table 30: use case specification 14		180
Table 31: use case specification 15	Error! Bookmark not defined.	2
Table 32: use case specification 16	Error! Bookmark not defined.	
Table 33: use case specification 17		203
Table 34: use case specification 18		203
Table 35:intial test cases		203

Table 36:RTM (V.2)	203
Table 37: VM Functional Requirements	203
Table 38: RAM Functional Requirements	203
Table 39: LM Functional Requirements	203
Table 40: FPM Functional Requirements	203
Table 41:UPM Functional Requirements	203
Table 42: EDB Functional Requirements	203
Table 43: ODBM Functional Requirements	203
Table 44: DEB Functional Requirements	203
Table 45: CNPEHP Functional Requirements	203
Table 46: EEP Functional Requirements	203
Table 47: DEP Functional Requirements	203
Table 48: VMP Functional Requirements	203
Table 49:EPI Functional Requirements.....	203
Table 50: EBI Functional Requirements	203
Table 51:RTM(V.3).....	203
Table 52: API Structure Overview.....	203
Table 53: API	203
Table 54: execution test cases.....	203

List Of Figures:

Figure 1 :gantt chart	38
Figure 2:project development life cycle model	41
Figure 3:use case diagram for system	Error! Bookmark not defined.
Figure 4:<<use case1(for visitor)>>	Error! Bookmark not defined.
Figure 5:activity diagram for visitor management.....	Error! Bookmark not defined.
Figure 6: <<use case2(for authentication system)>>	2Error! Bookmark not defined.
Figure 7: activity diagram for registration new account	Error! Bookmark not defined.
Figure 8:activity diagram for login	Error! Bookmark not defined.
Figure 9: activity diagram for forget password	Error! Bookmark not defined.
Figure 10: <<use case3(for user profile management)>>..	Error! Bookmark not defined.
Figure 11: activity diagram for user profile management...	Error! Bookmark not defined.
Figure 12: <<use case4(for events)>>	Error! Bookmark not defined.
Figure 13:<<use case5 (for package booking)>>	Error! Bookmark not defined.
Figure 14:<<use case 6(for Notification system) >>	Error! Bookmark not defined.

Figure 15:activity diagram for event discovery&booking	121
Figure 16: <<use case 7(for organizer) >>	122
Figure 17: activity diagram for organizer dashboard	Error! Bookmark not defined.
Figure 18: activity diagram for Disable/Enable Bookings.....	125
Figure 19: activity diagram for Create New Event	127
Figure 20: diagram for Edit Event	135
Figure 21: activity diagram for Drop Event.....	137
Figure 22: <<use case 8(for Provider) >>	139
Figure 23 : activity diagram for view My Profile	152
Figure 24: activity diagram for Edit Personal Information ..	Error! Bookmark not defined.
Figure 25: activity diagram for Edit Business Information..	Error! Bookmark not defined.
Figure 26:system architecture	168
Figure 27 :class diagram in high level	170
Figure 28:sequence diagram for visitor management	171
Figure 29: sequence diagram for registration account	173
Figure 30 : sequence diagram for Login	Error! Bookmark not defined.
Figure 31: sequence diagram for forget password Management.....	Error! Bookmark not defined.2
Figure 32 : sequence diagram for user profile management.....	Error! Bookmark not defined.
Figure 33: sequence diagram for event discovery&booking.....	Error! Bookmark not defined.
Figure 34: sequence diagram for Organizer Dashboard and Booking Management .	Error! Bookmark not defined.2
Figure 35: sequence diagram for Disable/Enable Bookings.....	Error! Bookmark not defined.2
Figure 36: sequence diagram for Create New Event or Host Package....	Error! Bookmark not defined.2
Figure 37: sequence diagram for Edit Event or Package ..	Error! Bookmark not defined.2
Figure 38: sequence diagram for Delete Event or Package	Error! Bookmark not defined.
Figure 39: sequence diagram for view my profile.....	Error! Bookmark not defined.2
Figure 40: sequence diagram for Edit Personal Information	Error! Bookmark not defined.2
Figure 41: sequence diagram for Edit Business Information	Error! Bookmark not defined.2
Figure 42: class diagram.....	Error! Bookmark not defined.2

Figure 43: Data Model Diagram.....	Error!
Bookmark not defined. 2	
Figure 44:user interface.....	2
Figure 45: user interface<<1>>.....	2
Figure 46: user interface<<2>>.....	2
Figure 47: user interface<<3>>.....	2
Figure 48: user interface<<4>>.....	2
Figure 49: user interface<<5>>.....	2
Figure 50: user interface<<6>>.....	2
Figure 51: user interface<<7>>.....	2
Figure 52: sign interface<<1>>.....	2
Figure 53: New Register For User<<1>>.....	2
Figure 54: New Register For User<<2>>.....	2
Figure 55: New Register For Supplier<<1>>.....	2
Figure 56: Dashboard For Organizer<<1>>.....	2
Figure 57: Dashboard For Organizer<<2>>.....	2
Figure 58: Dashboard For Supplier<<1>>.....	2

Figure 59: Manage events For Organizer<<2>>.....	2
Figure 60: Create New Event<<1>>.....	2
Figure 61: Create New Event<<2>>.....	2
Figure 62: Edit Event<<1>>.....	2
Figure 63: Edit Event <<2>>.....	2
Figure 64: Delete Event <<1>>.....	2
Figure 65: Disable/enable bookings<<1>>.....	2
Figure 66: Booking Management<<1>>.....	2
Figure 67: My Bookings<<1>>.....	2
Figure 68: User Profile for provider<<1>>.....	2

Chapter 1: Introduction

1. Introduction

This chapter provides an overview of the Events and Activities planning Platform project by outlining the problems we aim to address, the objectives we hope to achieve, and the structure of the report. It also serves as a guide to help you understand how the project was developed and the content of each section of the report.

1.1 Problem Definition

Today, the event industry faces a major challenge where everyone works separately, creating inefficiencies for all involved. Attendees struggle to find and book events across multiple websites with unclear pricing and availability information. Organizers waste time managing bookings manually, juggling different tools to communicate with suppliers, and lacking clear visibility into their event performance. Suppliers spend heavily on advertising just to get noticed by potential clients without a central platform to showcase their services. Our platform solves these problems by creating one unified space where attendees can discover and book events easily, organizers can manage everything through a simple dashboard with real-time insights, and suppliers can create profiles to connect directly with customers, making the entire event experience smoother and more efficient for everyone.

1.2 Project Objectives

Our platform is designed to serve three main user groups—attendees, organizers, and suppliers—each with features tailored to their needs. Attendees can securely manage their accounts, browse events using smart filters for date, location, price, and type, book tickets for existing events, or reserve ready-made packages to host their own, while receiving email confirmations and maintaining a full booking history. Organizers can create both public events and customizable host packages, add detailed event information including titles, descriptions, locations, dates, prices, and photos, track bookings in real time, view attendee lists, and control booking availability through a simple dashboard. Suppliers can create detailed profiles showcasing their services, locations, and contact information, making it easy for organizers and users to find them through searchable directories while building trust with clear information and future rating systems.

The platform also ensures smooth operation and organization by providing unique booking references, tracking the status of bookings, sending automated email notifications, and maintaining robust data security. Overall, our goal is to transform the platform from a basic booking system into a full-fledged event marketplace where attendees, organizers, and suppliers can collaborate efficiently and seamlessly.

4. Report Organization

The report is structured as follows:

This report is divided into the following chapters:

Chapter 1: Introduction

- What the project is about and why it matters

Chapter 2: Reference Study

- Looking at other similar platforms
- What's already out there and what we can learn from it

Chapter 3: Project Management

- How we planned and organized the work
- Timeline and responsibilities

Chapter 4: System Analysis and Design

- What the system needs to do
- How we designed the database and user interface
- Diagrams and mockups

Chapter 5: Implementation

- The technology we used
- How we built the main features
- User interfaces and APIs

5-Summary

This chapter introduces our Events and Activities planning platform. It outlines the challenges people face when planning or attending events, the goals we hope to achieve with the platform, and how this report is structured. The main idea is simple: bring attendees, organizers, and suppliers together on a single platform, making it easy for everyone to find what they need. In the upcoming chapters, we will go deeper into how we researched, planned, designed, and built the system.

Chapter 2:Reference Study

1. Introduction:

This chapter presents the reference studies we have conducted. The reference study is a process of reviewing, analysing, and summarizing previous sources and studies related to similar applications. This step has helped us identify areas that can be improved and developed.

2. Similar Applications:

1. Eventbrite:

Description: Eventbrite is a leading global platform for organizing and selling event tickets online. It allows users to create professional event pages, set ticket prices, manage registrations, and track sales through an easy-to-use interface. It is widely used for public events such as conferences, concerts, and workshops.

Advantages	Disadvantages
1. Simple and user-friendly interface for beginners and small organizers.	1. Charges commissions and fees on every ticket sold, reducing organizer profits.
2. Provides integrated tools for ticket booking and instant email confirmations.	2. Limited customization options for event page designs.
3. Allows integration with social media platforms to boost engagement.	3. Limited support for locally based or multi-service events (e.g., vendor collaboration).
4. Offers basic marketing tools and statistical reports on attendance and sales.	4. Analytical tools are basic compared to large enterprise platforms.

Table 1: Eventbrite

2. Cvent:

Description: Cvent is a comprehensive professional platform used for managing large-scale events, particularly conferences and exhibitions. It provides tools for registration management, venue coordination, vendor handling, badge printing, and advanced analytical reporting.

Advantages	Disadvantages
1. A fully integrated system that covers all stages of event management, from booking to evaluation.	1. The interface is relatively complex and requires training for new users.
2. Provides professional tools for managing vendors, budgets, and participant communication.	2. High cost makes it less suitable for small projects or individuals.
3. Offers precise analytics and interactive dashboards to support decision-making.	3. Not ideal for personal or simple events.
4. Reliable and trusted by large institutions, universities, and global corporations and Highly customizable to fit different event types (in-person or virtual).	4. Requires a longer setup and configuration time initially.

Table 2: Cvent

3. Bizzabo:

Description: Bizzabo focuses on enhancing attendee experience and engagement in professional events. It provides solutions for registration management and event marketing (virtual or in-person) while offering detailed reports to measure ROI and event impact.

Advantages	Disadvantages
1. Modern, flexible, and easily customizable user interface.	1. Subscription plans are relatively expensive compared to traditional platforms.
2. Strong reporting and analytics to measure event performance and business impact.	2. May not be ideal for small or local events.
3. Excellent support for virtual and hybrid events.	3. Customization of registration designs can be limited.
4. Provides marketing tools and personalized content for different attendee types and Emphasis on attendee networking and relationship building.	4. Requires some technical experience for efficient management.

Table 3: Bizzabo

4.Splash:

Description: Splash is a platform specialized in designing visually appealing event pages and promoting them as part of marketing campaigns. It is used by major companies to organize brand events and manage invitations and marketing analytics.

Advantages	Disadvantages
1. Professional design interface for creating visually striking event pages.	1. Not designed for ticket sales or logistical event management.
2. Strong focus on marketing aspects and audience data collection.	2. Full features require high-cost paid subscriptions.
3. Automated and well-coordinated tools for invitation and attendance management.	3. Limited suitability for non-marketing or non-commercial events.
4. Integrates with marketing tools like Mailchimp and CRM systems and Ideal for brand-focused corporate events emphasizing visual identity.	4. Analytical tools are less comprehensive than those of professional conference platforms like Cvent.

Table 4: Splash

5.Faaliyah App_ Syria:

Description: it is the official platform in Syria for organizing and managing exhibitions, projects, and events

It aims to support economic and cultural activities by bringing together government entities, organizers, and companies on a single digital platform.

It allows users to explore local events, register, and officially participate, while enabling organizers to manage requests and schedule exhibitions electronically, thus contributing to the digitization of the events sector and enhancing communication between participants and visitors.

Advantages	Disadvantages
1. A simple and user-friendly Arabic interface for local users.	1. Lacks ticketing tools or direct electronic payment.
2. It encompasses various types of events (economic, cultural, social, and educational).	2. Lacks interactive features such as event or supplier ratings.
3. It allows for application and participation in exhibitions through a unified electronic form.	3. Does not allow users to create custom events or hostable packages.
4. It helps promote local national projects.	4. Limited analytics and reporting, focusing on administrative rather than marketing aspects

Table 5: Faaliyah App_ Syria

3. Our System :

it offers a comprehensive solution connecting event organizers, attendees, and suppliers within a single, unified digital environment. Our vision is to provide a dynamic marketplace for event management, from discovery to execution. Users can register, edit their profiles, and search for and filter events by date, location, price, and type. For organizers, the platform offers comprehensive management features, including a shared dashboard, the ability to create and manage public events or hostable packages for their own events (such as wedding packages, birthday parties, or private gatherings), and track booking status (active, canceled, completed). We also facilitate collaboration by creating a searchable supplier directory by type and city. Our primary goal is to enhance usability. Each booking generates a unique, trackable reference, and the system is supported by instant automated email notifications for new registrations, ensuring high efficiency and a seamless user experience for both attendees and hosts.

4.Comparison of Systems

Feature	Splash	Bizzabo	Cvent	Eventbrite	Faaliyeh	Our system
User-Friendly Interface	✓	✓	✗	✓	✓	✓
Ticket sales & automated confirmations	✗	✓	✓	✓	✗	✓
Hostable/custom packages	✗	✓	✓	✗	✗	✓
Creating Special Events	✓	✓	✓	✓	✗	✓
Affordable for small/local events	✓	✓	✓	✓	✓	✓
Supporting Commercial Events	✓	✓	✓	✓	✗	✓
Collaborative Dashboard	✓	✓	✓	✓	✗	✓

Table 6: Comparison of Systems

5.Summary

We compared our system with popular event management platforms such as Eventbrite, Cvent, Bizzabo, Splash, and Faaliyeh App, focusing on key features like a user-friendly interface, ticket sales with automated confirmations, hostable or customizable event packages, the ability to create special events, suitability for small or local projects, support for commercial events, collaboration among users, and an event management dashboard. These platforms are effective in managing events and providing essential tools for users, but they face challenges such as high costs, limited customization, complex setup, and limited support for local events.

Chapter 3: Project Management

1. Project Charter

<u>Project title:</u> Alternative Energy Management System.
<u>Project start date:</u> October 1 2025
<u>Project finish date:</u> January 21 2026
<u>Project manager:</u> Eng. Anas Abdulaziz.

Table7:project charter info

Project objectives:

- ❖ Develop a unified event management platform that integrates attendees, organizers, and suppliers.
- ❖ Provide seamless event discovery with advanced filtering capabilities (date, location, price, event type).
- ❖ Enable dual booking functionality: booking tickets for public events and booking packages to host private events.
- ❖ Implement a comprehensive organizer dashboard with real-time analytics (bookings, revenue, available seats).
- ❖ Allow organizers to create and manage both public events and customizable host packages with full event details.
- ❖ Enable organizers to tag basic supplier notes for each event or package for internal reference.
- ❖ Create a supplier directory that connects service providers with organizers and users through searchable profiles.
- ❖ Manage bookings with unique reference numbers, status tracking (confirmed, canceled), and email notifications.
- ❖ Build a scalable foundation for future marketplace features including ratings, reviews, and direct supplier integrations.
- ❖ Create a supplier directory that connects service providers with organizers and users through searchable profiles.

Approach:

- ❖ Define the project scope and requirements.
- ❖ Conduct system analysis and design for all user roles (attendees, organizers, suppliers).
- ❖ Design system architecture including user authentication, event management, and booking systems.
- ❖ Develop user interface for event discovery, filtering, and dual booking options (public events and host packages).
- ❖ Implement a real-time notification system for booking confirmations and updates.
- ❖ Create a supplier directory with search and filter functionality.
- ❖ Deploy the platform and provide user documentation.

Roles and Responsibilities:

Name	Role	Responsibility
Eng. Anas Abdulaziz	Project Supervisor	Provides guidance and evaluates project progress.
Helpest Mohamad	Project Member	Responsibility for Project Management and overseeing all SDLC phases.
Hiba Shawa	Project Member	Responsibility for Project Management and overseeing all SDLC phases.
Marwa Aljaouni	Project Member	Responsibility for Project Management and overseeing all SDLC phases.

Table 8: roles and responsibilities

2.Methodology:

1.Introduction

This section aims to explain and justify the chosen methodology for developing the events and activities planning platform. Based on the project's nature and requirements, this document proposes adopting a hybrid Agile methodology combining the Scrum and Kanban frameworks to achieve the optimal balance between the necessary structure and flexibility.

2.Description of the Selected Methodology

Detailed description of the selected methodology: Hybrid Agile (Scrum and Kanban):
Core Principles and Values: This methodology is based on Agile principles, such as focusing on the customer and their needs, early and continuous delivery of functional value, welcoming changes in requirements, close collaboration between stakeholders and developers, motivating individuals and trusting them to accomplish their tasks, sustainable work that maintains a constant development pace, attention to technical details and quality, simplifying processes, encouraging team self-organization, and regular reflection on ways to continuously improve.

2.1 Framework and Key Processes:

Scrum: Scrum will be used to organize development cycles into short, fixed-duration time units called "Sprints." Scrum events will include Sprint Planning meetings, the Daily Scrum, the Sprint Review meeting, and the Sprint Retrospective meeting. System requirements will be documented and managed in the "Product Backlog" and prioritized by the "Product Owner."

Kanban: Kanban principles will be applied to visualize the workflow within each Sprint and across different development stages. A "Kanban board" will be used to track the status of different tasks and limit "Work in Progress" (WIP) to ensure a smooth flow of work and reduce potential bottlenecks.

2.2 Key Roles and Responsibilities:

- **Product Owner:** Responsible for defining the product vision and managing and prioritizing the Product Backlog: Eng. Anas Abdulaziz
- **Scrum Master:** Responsible for facilitating the Scrum process and ensuring the team understands and applies Scrum and Kanban principles and practices correctly: Hiba Shawa
- **Development Team:** A multidisciplinary team responsible for analyzing, designing, developing, testing, and delivering the solar energy management system: Marwa AlJaouni, Helpest Mohamad, Hiba Shawa

2.3 Expected Deliverables: There will be shippable software increments at the end of each Sprint, in addition to a visible Kanban board to track work progress, and a managed and continuously updated Product Backlog.

3. Justifications for Choosing the Methodology:

Adopting a hybrid Agile methodology combining Scrum and Kanban is considered the most suitable option for developing the solar energy management system for the following reasons:

3.1 Project Nature and Requirements:

Complexity and Flexibility: The solar energy management system may involve integration with various devices, analysis of complex data, and different user interfaces. These complexities require a high degree of flexibility in dealing with requirements that may become fully apparent during the development process.

Expected Changes: Changes in system requirements are likely to occur based on continuous developments in solar energy technology or based on user feedback.

Gradual Delivery of Value: Using Scrum allows for the frequent delivery of functional parts of the system at the end of each Sprint, allowing stakeholders to provide early feedback and obtain tangible value early in the project lifecycle.

Risk Mitigation: Through frequent delivery and regular reviews, potential problems and risks can be identified early and addressed effectively.

Team Characteristics: Given that the team size is only 3 people, Scrum promotes effective collaboration and communication within small and cross-functional teams.

3.2 Project Constraints:

Timeline and Early Delivery: Frequent delivery in Scrum allows for providing basic parts of the system early, which may meet some urgent stakeholder needs.

3.3 Stakeholder Requirements:

Active Participation: Scrum events such as Sprint Review meetings encourage stakeholders to participate and provide their feedback regularly, ensuring that the system is developed in accordance with their needs and expectations.

Clear Vision of Progress: Shippable software increments and the Kanban board provide a clear view of the progress made in developing the system.

4. Specific Application of the Methodology to This Project:

4.1. Methodology Customization:

The Scrum framework will be applied with the integration of a Kanban board to visualize the flow of Sprint tasks. The Sprint length will initially be set at two weeks, and this duration can be adjusted based on the team's experience and project needs. The focus will be on applying Work in Progress (WIP) limits to specific development stages within the Sprint to improve workflow and reduce task accumulation.

4.2 Tools and Technologies:

An Agile project management tool (such as Jira) will be used to support the management of the Product Backlog, Sprint Backlog, and visualization of the Kanban board. Online communication and collaboration tools will also be used to ensure effective communication between team members and stakeholders.

4.3 Communication Processes:

Communication processes will include the Daily Scrum for the development team, Sprint Planning, Review, and Retrospective meetings, as well as periodic meetings with stakeholders to present progress and gather feedback and guidance. Digital communication channels (such as email and instant messaging platforms) will be used for asynchronous communication.

5.Change Management Processes within the Selected Methodology:

Changes in requirements or unforeseen circumstances will be handled through the following mechanisms:

Daily Scrum: The team can use the Daily Scrum to discuss any emerging changes or new challenges and how they affect the current work, and to plan how to respond to these changes effectively.

Sprint Review: The Sprint Review will be an important opportunity for stakeholders to provide feedback on the work done and suggest changes or new priorities that can be added to the Product Backlog to be addressed in the next Sprint.

Product Backlog Refinement: Regular Product Backlog Refinement sessions will be held to review, clarify, and estimate requirements, allowing for the integration of new changes or modification of existing requirements based on new priorities set by the Product Owner.

6.Summary:

Based on the analysis of the nature of the solar energy management system project and its requirements, team characteristics, project constraints, and stakeholder requirements, we confirm that adopting a hybrid Agile methodology combining the Scrum and Kanban frameworks is the optimal choice for this project. This methodology provides the necessary flexibility to deal with complexities and potential changes, promotes effective collaboration and communication, allows for the gradual delivery of functional value, and helps mitigate potential risks.

3.Statement of work:

1.Project title: Events And Activities Planning platform.

2.project description and objectives:

This document specifies the software requirements and design for the Event Management Platform, a unified solution designed to connect event attendees, organizers, and suppliers in one seamless ecosystem. The system enables users to discover and book tickets for existing events or host their own events using pre-configured packages, organizers to create and manage events efficiently with real-time analytics, and suppliers to showcase their services, ultimately creating a dynamic event marketplace.

3.Project Scope

Scope: The Event Management Platform is a web-based application that brings together three key stakeholders in the event industry. The system allows attendees to browse and book events or host their own using pre-configured packages, provides organizers with comprehensive management tools and analytics dashboards, and offers suppliers a platform to connect directly with potential clients through searchable profiles.

4. project goals:

- Develop a user-friendly platform that simplifies event discovery, booking, and hosting for attendees.
- Create powerful management tools for organizers to handle multiple event types with real-time tracking and insights.
- Build a supplier directory that connects service providers with organizers and users transparently.
- Implement secure authentication and booking management systems.
- Provide users with:
 - Advanced event filtering and search capabilities
 - Automated booking confirmations and notifications

- Real-time dashboards showing bookings, revenue, and performance metrics
- Booking history and status tracking

5.Deliverables:

- Project plan and charter
- SRS (Software Requirements Specification) documentation
- System design documents (database schema, architecture diagrams)
- Final project report
- Backend and frontend system components – the complete platform

6. Project Requirements:

Technology and Tools:

Frontend: React.js – For user interfaces, dashboards, and interactive components

Backend: Spring Boot (java)– For API management and business logic

Database: PostgreSQL / Firebase Firestore – For storing users, events, bookings, and supplier data

Authentication: Firebase Auth – For secure user login and session management

Email Service: SendGrid or Nodemailer – For booking confirmations and notifications

Hosting: AWS, Heroku, or Vercel – For deployment

7. Assumptions:

Organizers will provide accurate event information (dates, locations, pricing).

Suppliers will maintain updated profile information.

Email addresses provided by users are valid for receiving notifications.

Continuous availability of project team members throughout development.

Regular supervision and feedback from the project supervisor.

8. Project Resources:

Human Resources:

Eng. Anas Abdulaziz : Project Manager

Hiba Shawa : SE Developer

Marwa Aljaouni : SE Developer

Helpest Mohamad : SE Developer

9. Schedules:

<u>Project Start Date:</u> 25 \10\2025
<u>First project Seminar:</u> 16\11\2025
<u>Second project Seminar:</u> 20\12\2025
<u>Project Finish Date:</u> 21\01\2026

Table 9: schedules

4. Project Timeline

1. Basic Project Information

- **Project Name:** Events And Activities Planning Platform
- **Project Manager:** Eng. Anas Abdulaziz
- **Start Date:** 25/10/2025
- **Expected End Date:** 21/01/2026
- **Total Duration :** Approximately 3 months

Main phase	sub-task	approximate start date	approximate end date	approximate
sprint 1:	1.1 Requirements Analysis	25/10/2025	31/10/2025	1 wk
	1.2 System Architecture Design	01/11/2025	07/11/2025	1 wk
	1.3 Database Schema Design	08/11/2025	15/11/2025	1 wk
sprint 2:	2.1 Actors Authentication	16/11/2025	29/11/2025	2 wks
	2.2 User Module (Event Browsing & Booking)	21/11/2025	05/12/2025	2 wks
sprint 3:	3.1 Organizer Module (Dashboard & Event Management)	06/12/2025	20/12/2025	2 wks
	3.2 Provider Module (Directory & Profile Management)	12/12/2025	24/12/2025	1.5 wks
sprint 4:	4.1 Dashboard Implementation	25/12/2025	08/01/2026	2 wks
	4.2 Booking Management	31/12/2025	13/01/2026	2 wks
	4.3 Notification System	25/12/2025	10/01/2026	3 wks
sprint 5:	5.2 Testing	03/01/2026	15/01/2026	2 wks
	5.3 Submission	10/01/2026	20/01/2026	1.5 wks
Project Completion			21/01/2026	

Table 10: timeline

5.Gantt Chart

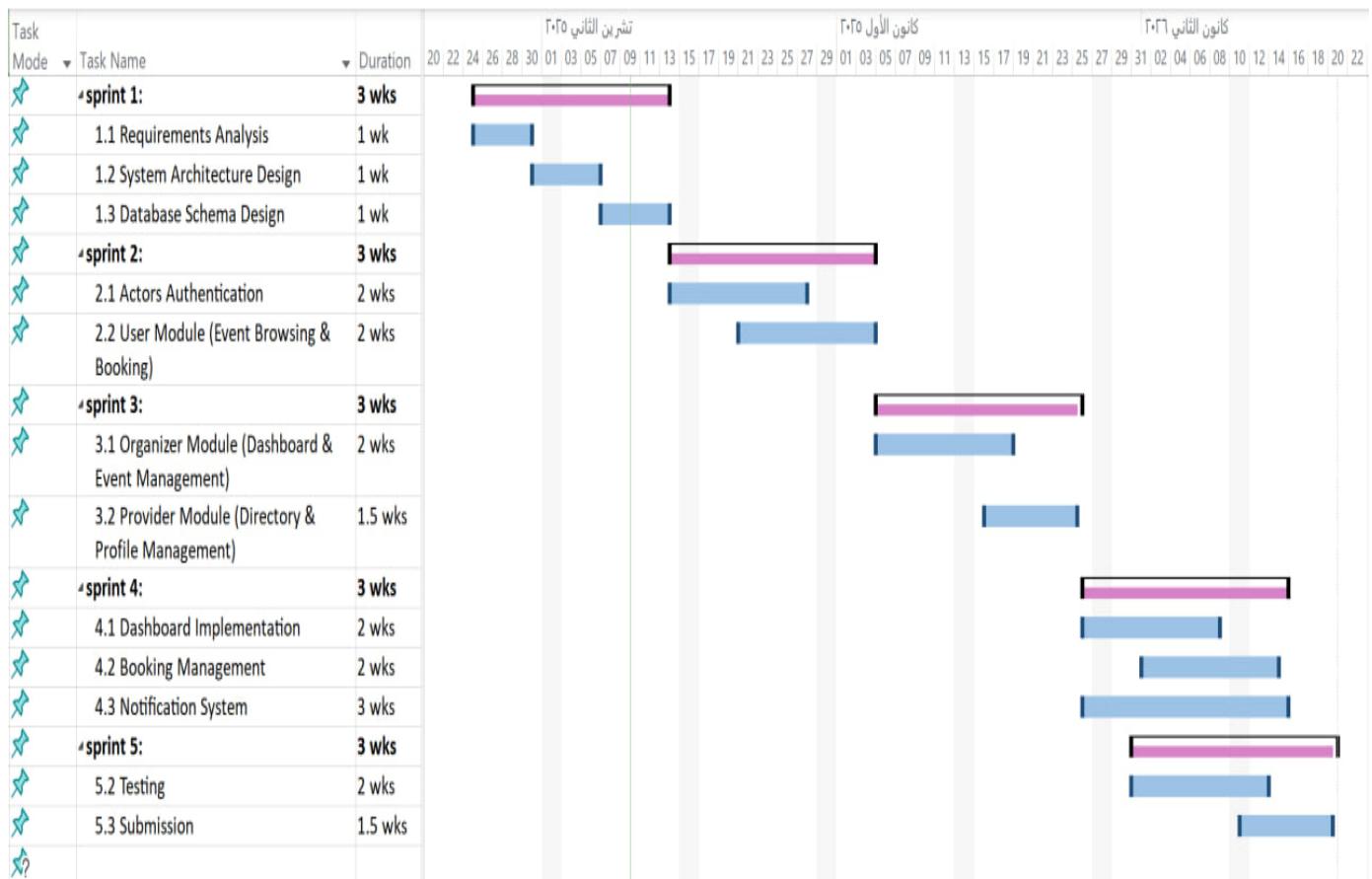


Figure 1 :gantt chart

6.Risk management:

Risk-ID	Risk Title	Risk Description	Raised Date	Tracking Frequency	State	Impact	Mitigation Plan
RK-01	Team limitation only 3 persons	Small team compared to the project size, so if one of the members stops working for any reason, it might cause the project to be delayed completely	01/04/2025	weekly	active	high	Collaboration between team members, defining tasks and dividing them clearly with the possibility of assistance when needed and working on completing them within the specified time.
RK-02	Repetition of the proposed project idea	It might become clear that the proposed idea resembles previous works of the department's students without the existence of additions or clear distinction that reduces the idea's value.	01/04/2025	weekly	closed	medium	Conducting a comprehensive research on the completed projects in the university and consulting the supervisor to evaluate the originality of the idea and its value and obtaining approval from him.
RK-03	Ambiguity in the project scope	The understanding of the project scope might be unclear at the beginning, which leads to committing big mistakes in implementation and unplanned expansion in the tasks required later.	01/04/2025	daily	closed	high	Conducting organized discussions and meetings for the team to clarify the objectives and ensure a shared understanding of the scope and documenting it clearly and obtaining the supervisor's approval.
RK-04	Inaccurate energy calculations	Inaccurate energy production/consumption calculations this could lead to errors in input data, affecting the final results	20/4/2025	weekly	under mitigation	high	Verifying the data sources used in the calculations and conducting intensive tests to verify the accuracy of the calculations in different scenarios.

RK-05	Data loss/corruption	concerns about database system failures Technical glitches or human errors might cause loss of critical information.	25\4\2025	daily	active	high	Implementing a data backup strategy and storing it in secure locations with the use of Firebase as an alternative and conducting checks to ensure the validity of the data.
RK-06	Incorrect user input	Users may enter incorrect or incomplete data this could result in processing and analysis errors	30\4\2025	weekly	under mitigation	medium	Designing easy-to-use data entry interfaces with the application of mechanisms to verify the validity of the entered data and providing clear error messages to guide users when entering incorrectly.
RK-07	Scalability issues	System may struggle with user/data volume increases this could lead to performance slowdowns	5\5\2025	weekly	active	medium	Designing the system while taking into account future scalability and conducting performance and endurance tests for the system to identify potential weaknesses when the load increases.
RK-08	Frequent requirement changes	frequent changes in project requirements Whether from the team or stakeholders, this may disrupt both the project timeline and scope	10/05/2025	daily	active	high	Encourage continuous and effective communication with stakeholders to understand their needs as early as possible, freeze the core requirements at an early project stage, and allocate some time to handle potential changes.

Table11: risk management

7. Project Development Life Cycle Model:

5.1 Determine development model based on system requirements and structure: This section explains the process of selecting the appropriate software development model for the project. After conducting a thorough analysis and extracting the system requirements, as well as breaking down the system into its core components, the next step is to choose the most suitable software development process for our system. Given the nature of the system and its requirements, which necessitate flexibility and continuous adaptation, a Hybrid Agile methodology that combines Agile principles with specific frameworks, namely Scrum and Kanban, has been adopted as the best approach for development



Figure 2:project development life cycle model

8. Hybrid Agile Project Management:

This section adopts a flexible approach to project management that differs from traditional models requiring comprehensive and detailed upfront planning. Hybrid Agile project management relies on adaptive and iterative planning and execution, integrating elements from Scrum and Kanban to achieve maximum efficiency and effectiveness. This approach focuses on delivering business value and providing benefits to clients through: Iterative and Incremental Planning ,Visualize Workflow and Limit Work in Progress ,Regular Meetings and Continuous Communication ,Regular and Transparent Reporting and Sending alerts.

9. Initial Requirement Trackability Matrix (RTM)(V.1)

Requirement id	Title	Use case	System Design	Analysis	Detailed Design	Coding	Test Cases
PM-01	Assign roles and responsibilities to team members						
PM-02	Prepare and monitor a detailed project schedule						
PM-03	Hold weekly meetings to review project progress						
PM-04	Conduct risk assessments and develop mitigation plans						
PM-05	Keep track of resources (team time, skills, tools) to use them well and finish the project on time						
PM-06	Ensure quality assurance checkpoints are in place						
PM-07	Perform final project closure and evaluation						
PM-08	Document and approve the chosen project methodology						
PM-09	Periodically review the application of the project methodology						
PM-10	Document all project changes and approvals						

Table12: RTM(V.1)

10.Summary:

This chapter outlined the project management plan for the "**Events and activities planning platform**" It began with the project charter, which defined the main objectives, team roles, and responsibilities. The chosen methodology (**Hybrid Agile, combining Scrum and Kanban**) was selected for its flexibility and suitability to the project's evolving requirements. The development process was broken down into clearly defined stages, from requirements analysis to final delivery, all organized within a detailed project timeline. The chapter also covered the project scope, required resources, and selected development tools, along with the appropriate software development life cycle model. Overall, the plan emphasizes that flexible management and effective organization are key to delivering a high-quality system on time.

Chapter 4: System Analysis

1. System Initial Analysis

This section presents the initial analysis and key features identified during the early stages of project planning and development.

1. Introduction

1.1 Purpose

This document specifies the software requirements and design for the Event Management Platform, a comprehensive solution designed to facilitate event discovery, booking, and management. The platform connects three key stakeholders: attendees seeking events to join or host, organizers managing events and packages, and suppliers offering event-related services. The system provides an intuitive marketplace that simplifies event participation, creation, and collaboration.

1.2 Project Scope

This system is an effective and user-friendly platform for managing the complete event lifecycle—from discovery and booking to hosting and supplier collaboration. It relies on organizer-created events and supplier services to provide a seamless experience for all stakeholders, enabling efficient event operations and enhanced user satisfaction.

Key Features:

- Unified platform for attending and hosting events
- Comprehensive organizer dashboard with analytics
- Integrated supplier directory for service discovery
- Real-time booking management and notifications
- Scalable architecture supporting multiple user roles

1.2.1 High-Level Requirements:

1. User Management & Authentication

- User registration and login
- Profile management
- Password recovery
- Account deletion

2. Event Discovery & Browsing

- Browse available events
- Search and filter events
- View event details
- Display event availability

3. Booking Management

- Book tickets for public events (Attend)
- Book host packages (Host)
- View booking history
- Cancel bookings
- Receive booking confirmations

4. Organizer Management

- Create and manage public events
- Create and manage host packages
- Enable/disable bookings
- Track event status

5. Organizer Dashboard & Analytics

- View booking statistics
- Monitor revenue
- Track event performance
- Access recent bookings

6. providers Directory

- provider registration
- provider categorization
- Display provider profiles

7. Notification System

- Registration notifications
- Password recovery notifications
- Booking confirmations and cancellations
- Event update notifications
- Event reminders
- Organizer alerts

8. Visitor Access

- Browse events without registration
- Search and filter functionality
- View event and provider information
- Access registration when attempting to book

9. Security & Validation

- Data validation
- Secure authentication
- Password encryption
- Input sanitization

1.2.2 Actors:

1. Visitor (Non-registered User)

Responsibilities:

- Browse available events
- Search and filter events
- View event details
- View supplier directory
- Explore platform features before registration

Privileges:

- Read-only access to public content
- No booking capabilities
- No profile management

2. User / Attendee (Registered User)

Responsibilities:

- Register and maintain profile
- Browse and search events
- Book tickets for public events (Attend)
- Book host packages for personalized events (Host)
- Manage personal bookings
- Cancel bookings when needed
- Receive notifications

Privileges:

- Access to personal dashboard
- View booking history
- Modify personal profile data
- Receive booking confirmations
- Access full event details
- Cancel own bookings

- **3. Organizer**

Responsibilities:

- Create and manage public events
- Create and manage host packages
- Define event details and pricing
- Enable/disable bookings per event
- Monitor event performance
- Tag suppliers to events
- Respond to booking notifications
- Manage event capacity

Privileges:

- Access to organizer dashboard
- Create/edit/delete events
- View detailed analytics and revenue reports
- Access to all booking information
- Filter and search own events
- Receive organizer-specific notifications
- Tag providers to events

4. provider

Responsibilities:

- Register and maintain business profile
- Provide service information
- Update contact details
- Maintain service quality for visibility

Privileges:

- Create public supplier profile
- Display business information
- Be discoverable by users and organizers

- Static presence in directory

2. Overall Description

2.1 Product Perspective

Product Perspective

The Event Management Platform is a standalone web-based product designed to revolutionize event discovery, participation, and management. It is not part of an existing product family nor a replacement for any current system. The platform serves as a comprehensive marketplace connecting three distinct user groups: attendees seeking engaging events, organizers creating memorable experiences, and suppliers providing essential event services.

The system operates as a centralized hub where:

- Users discover events that match their interests and location
- Organizers manage event logistics and track performance
- Suppliers gain visibility and connect with potential clients

The platform is designed with scalability in mind, allowing for future enhancements such as payment gateway integration and advanced supplier interactions.

2.2 Product Features

1. Event Management

- Create and manage public events and host packages
- Enable/disable bookings
- Track event status and performance

2. Booking System

- Book tickets for public events
- Book host packages
- Manage bookings and cancellations

3. Organizer Dashboard

- View bookings and revenue analytics
- Monitor event performance

- View booking statistics

4. Provider Directory

- Browse service providers
- Search by category and location
- Tag providers to events

5. Notification System

- Automated email notifications
- Booking confirmations and reminders
- Event updates and alerts

6. User Management

- Registration and authentication
- Profile management
- Role-based access

2.3 User Categories and Their Characteristics

2.3.1 Visitor (Non-registered User)

Responsibilities:

- Browse events
- Search and filter events
- View event and supplier details

Privileges:

- Read-only access to public content
- Cannot make bookings

2.3.2 User/Attendee (Registered User)

Responsibilities:

- Manage profile

- Book events (Attend or Host)
- Manage bookings
- Cancel bookings

Privileges:

- Access personal dashboard
- View booking history
- Modify profile
- Receive email notifications

2.3.3 Organizer

Responsibilities:

- Create and manage events
- Create and manage host packages
- Enable/disable bookings
- Monitor event performance
- Tag suppliers to events

Privileges:

- Access organizer dashboard
- Create/edit/delete events
- View booking details and analytics
- Filter and search events
- Receive organizer notifications

2.3.4 Provider

Responsibilities:

- Register and maintain business profile
- Provide service information
- Update contact details

Privileges:

- Create and edit business profile
- Public directory listing
- Display contact information

3.High-Level Requirements Table:

HLR ID	High-Level Requirement	Description	Priority
HLR-1	User Management	System shall provide complete user account management including registration, authentication, profile management, and session handling	High
HLR-2	Event Discovery & Browsing	System shall allow users and visitors to discover, search, filter, and view event details	High
HLR-3	Event Booking System	System shall enable users to book events and host packages with capacity management and booking confirmation	High
HLR-4	Booking Management	System shall allow users to view booking history and cancel bookings with automatic capacity restoration	High
HLR-5	Event Management (Organizer)	System shall allow organizers to create, edit, delete, and manage public events and host packages	High
HLR-6	Organizer Dashboard & Analytics	System shall provide organizers with dashboard displaying event management tools, statistics, and revenue tracking	High
HLR-7	Supplier Directory	System shall provide supplier registration, directory browsing, and search functionality	Medium
HLR-8	Notification System	System shall send email notifications for registration, booking actions, event updates, reminders, and organizer alerts	High
HLR-9	Visitor Access	System shall allow non-registered visitors to browse events and suppliers with restricted booking access	High
HLR-10	Security & Validation	System shall enforce data validation, password security, session management, and access control	High

Table13: High-Level Requirements

4.Use case diagram:

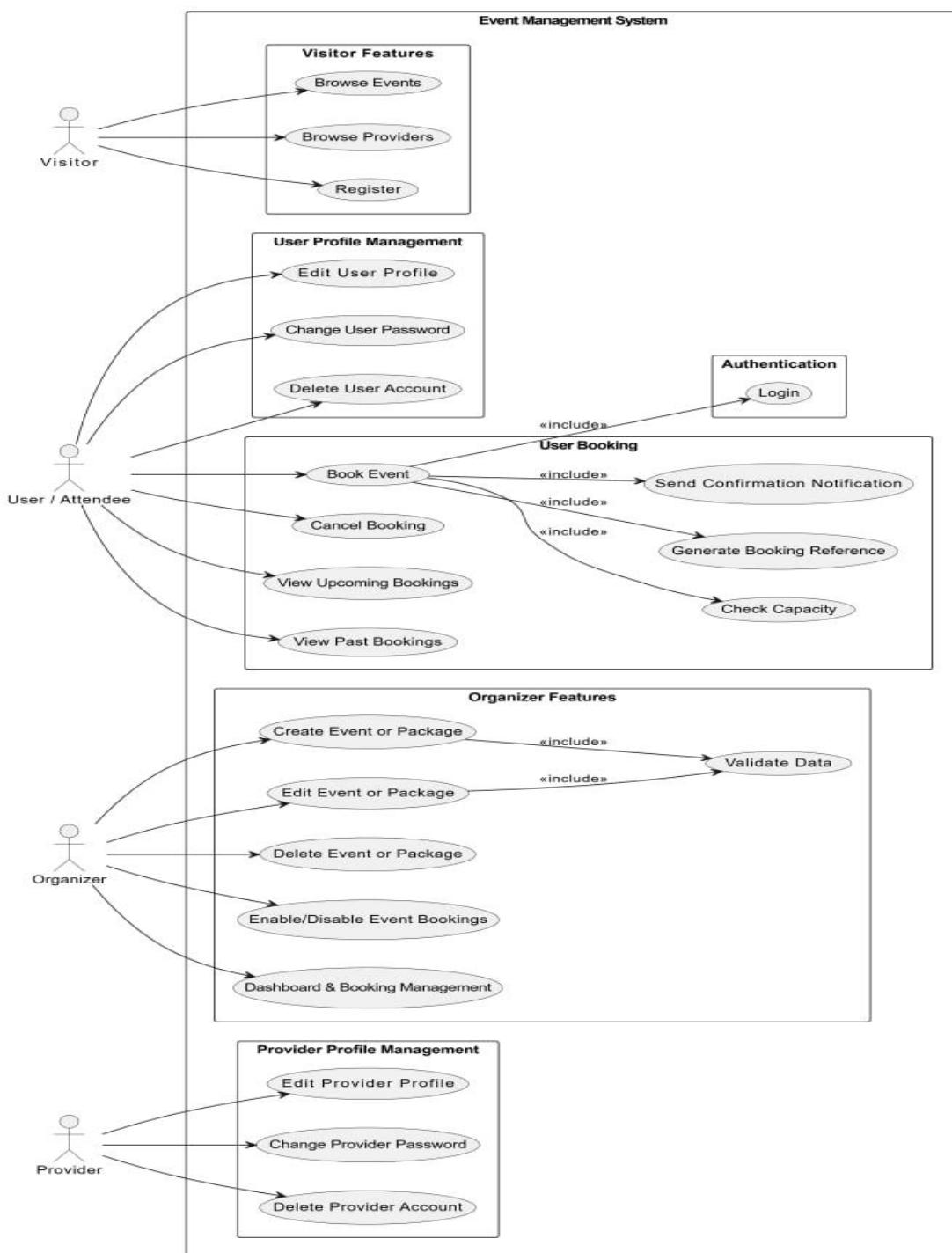


Figure 3: use case diagram

2.Detailed Analysis

1.Introduction

This chapter aims to transform the functional requirements identified during the preliminary analysis phase into precise engineering designs. Utilizing UML modeling to illustrate system interactions, this section will establish the technical framework to be adopted by the development team in subsequent stages

2.purposes

we will:

- Formalize system behavior through state diagrams capturing entity lifecycles
- Map operational workflows via activity diagrams
- Define interaction patterns using sequence diagrams
- Establish validation protocols for data integrity
- Model the static structure of the system using class diagrams, defining key entities (classes), their attributes, operations, and the relationships between them. This will provide a blueprint for the system's architecture and data organization.

3.system requirements:

1. Visitor Management Functional Requirements:

Requirements ID	Description	Module	Priority	Detailed Specification
REQ-FR-VM-01	Visitor browse events	Visitor Management	High	<ul style="list-style-type: none"> Allows visitors to view all available events without logging in
REQ-FR-VM-02	Visitor search & filter	Visitor Management	High	<ul style="list-style-type: none"> Allows visitors to search events by title or keywords Allows filtering events by date, location, price, and type
REQ-FR-VM-03	Visitor view event details	Visitor Management	High	<ul style="list-style-type: none"> Displays full event details including description, images, date, time, location, and availability
REQ-FR-VM-04	Prevent visitor booking	Visitor Management	High	<ul style="list-style-type: none"> Restricts booking functionality to logged-in users only
REQ-FR-VM-05	Redirect to login for booking	Visitor Management	High	<ul style="list-style-type: none"> Redirects visitors to registration/login page when attempting to book
REQ-FR-VM-06	Visitor browse suppliers	Visitor Management	Medium	<ul style="list-style-type: none"> Allows visitors to view provider directory Displays provider name, service type, city, and description

Table 14: VM Functional Requirements

3.1 Requirements Modeling:

3.3.1.1 use case diagram for visitor :

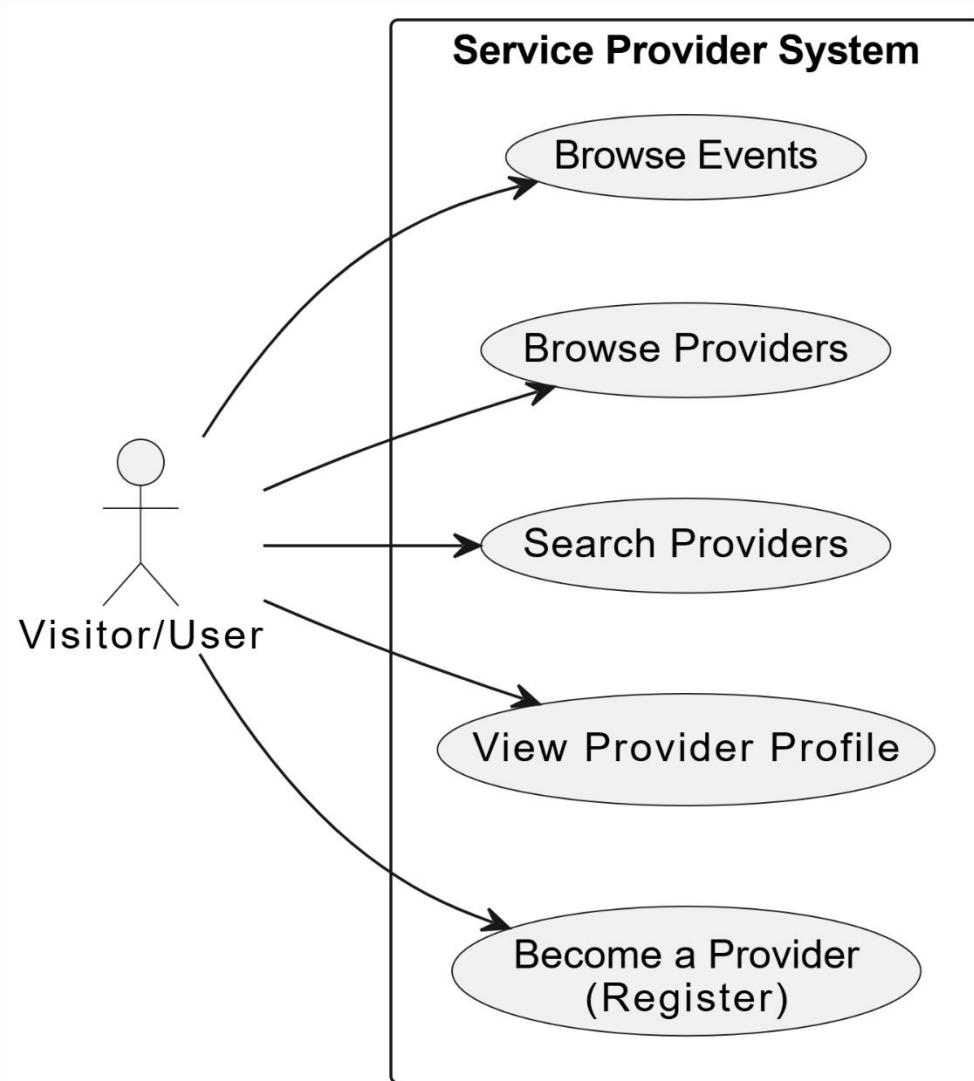


Figure 4 : <<use case 1 (for visitor) >>

3.3.1.2 activity diagram for visitor management:

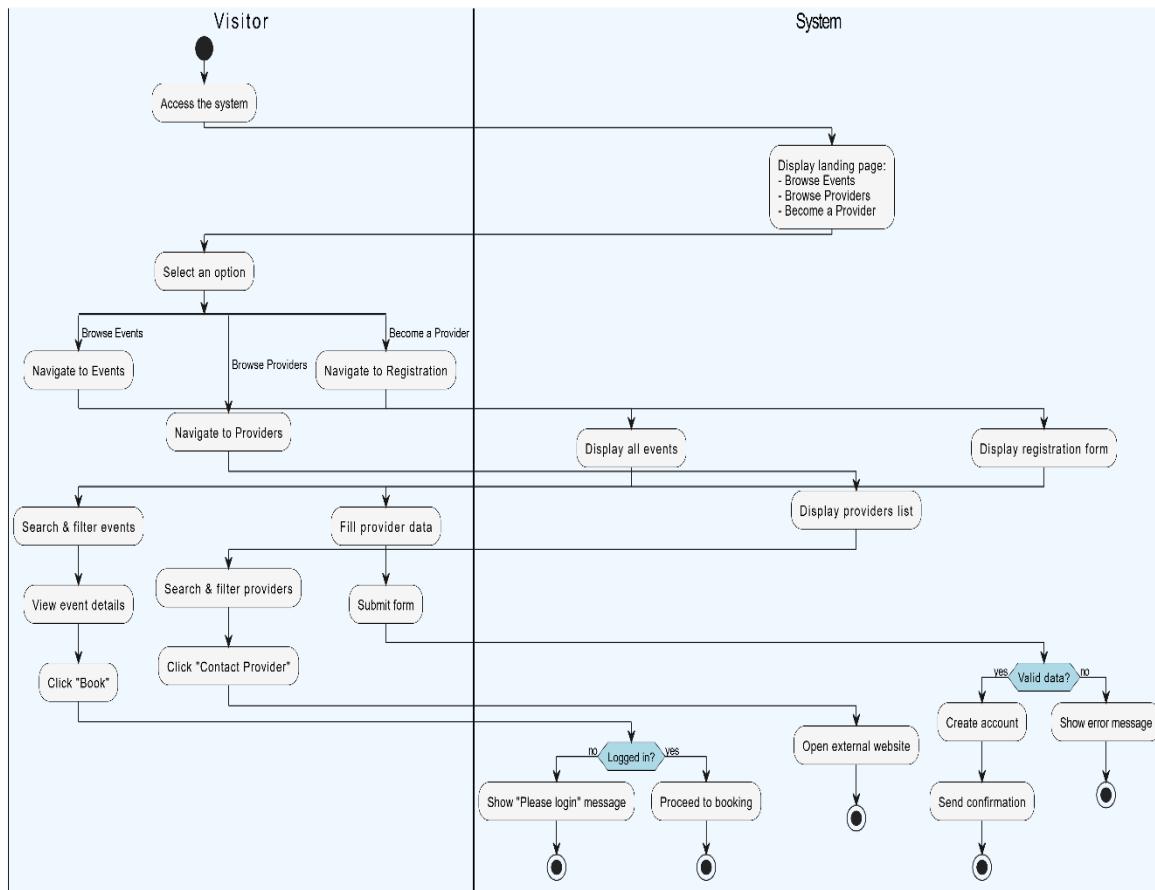


Figure 5:activity diagram for visitor management

3.3.1.3 use case specification:

Use Case 1: Browse Events as Visitor

Use case name:	Browse and View Events Without Registration
Participating Actors:	Visitor (Guest User)
The flow of events:	<ol style="list-style-type: none"> 1. Visitor accesses the system 2. System displays landing page with three options: <ul style="list-style-type: none"> o Browse Events o Browse Providers o Become a Provider 3. Visitor selects "Browse Events" 4. System displays all available events without requiring login 5. Visitor uses search & filter functionality (date, location, price, type) 6. System displays filtered results 7. Visitor clicks on "View Event Details" 8. System displays full event details (description, images, capacity, price, supplier info) 9. Visitor clicks on "Book Event" 10. System checks if user is logged in 11. System displays message: "You don't have an account ID. Please login or register to book" 12. System redirects to login/register page
Alternative flows:	<p>A1. Visitor doesn't attempt to book:</p> <ul style="list-style-type: none"> • Visitor continues browsing events • No login required <p>A2. Visitor is already logged in:</p> <ul style="list-style-type: none"> • System proceeds to booking flow • No login prompt displayed

	A3. No events match filter: <ul style="list-style-type: none"> System displays "No events found" message
Entry condition	<ol style="list-style-type: none"> System is operational Visitor has internet access
Exit conditions	<p>Success:</p> <ul style="list-style-type: none"> Visitor can browse all events freely Full event details displayed Search and filter work correctly Login prompt shown when attempting to book <p>Failure:</p> <ul style="list-style-type: none"> No events available System unable to load event data

Table 15: use case specification 1

Use Case 2: Browse Providers as Visitor

Use case name:	Browse Supplier Directory Without Registration
Participating Actors:	Visitor (Guest User)
The flow of events:	<ol style="list-style-type: none"> 1. Visitor accesses the system 2. System displays landing page with three options 3. Visitor selects "Browse Providers" 4. System displays supplier directory without requiring login 5. Visitor browses and filters supplier listings 6. Visitor clicks on "Contact Provider" 7. System navigates to provider's external website
Alternative flows:	<p>A1. No providers match filter:</p> <ul style="list-style-type: none"> • System displays "No providers found" message <p>A2. External website unavailable:</p> <ul style="list-style-type: none"> • System shows error message
Entry condition	<ol style="list-style-type: none"> 1. System is operational 2. Providers are registered in system
Exit conditions	<p>Success:</p> <ul style="list-style-type: none"> • Visitor can browse all providers freely • Filter functionality works • External website opens successfully <p>Failure:</p> <ul style="list-style-type: none"> • No providers available • External link broken

Table 16: use case specification 2

Use Case 3: Become a Provider (Register as Provider)

Use case name:	Provider Registration by Visitor
Participating Actors:	Visitor (Guest User becoming Provider)
The flow of events:	<ol style="list-style-type: none"> 1. Visitor accesses the system 2. System displays landing page with three options 3. Visitor selects "Become a Provider" 4. System displays Provider Registration Form 5. Visitor fills registration data: <ul style="list-style-type: none"> o Business Name o Contact Information (email, phone) o Services Offered o Business Description o Other required details 6. Visitor clicks "Submit" 7. System validates registration data 8. If valid: <ul style="list-style-type: none"> o System creates provider account o System sends confirmation email o System displays success message
Alternative flows:	<p>A1. Invalid or missing data:</p> <ul style="list-style-type: none"> • System displays validation error message • Visitor must correct errors • Visitor remains on registration form <p>A2. Email already registered:</p> <ul style="list-style-type: none"> • System displays "Email already exists" error • Visitor must use different email <p>A3. Visitor cancels registration:</p> <ul style="list-style-type: none"> • Visitor clicks "Cancel"

	<ul style="list-style-type: none"> • System returns to landing page • No account created
Entry condition	<p>1. System is operational</p> <p>2. Visitor wants to become a provider</p>
Exit conditions	<p>Success:</p> <ul style="list-style-type: none"> • Provider account created successfully • Confirmation email sent • Visitor can now login as provider <p>Failure:</p> <ul style="list-style-type: none"> • Validation errors displayed • Missing required fields • Email already exists • No account created

Table 17: use case specification 3

2. User Functional Requirements:

Requirements ID	Description	Module	Priority	Detailed Specification
REQ-FR-UFR-01	Create Account	Account Creation	High	<ul style="list-style-type: none"> • Collect user name, email, phone, and password • Validate email format • Enforce strong password policy (min 8 chars, 1 uppercase, 1 number) • Show registration success/failure notifications
REQ-FR- UFR - 02	User Login	Authentication	High	<ul style="list-style-type: none"> • Verify email and password • Authenticate against stored data • Maintain session securely • Show error message if credentials invalid
REQ-FR- UFR - 03	User Logout	Authentication	Medium	<ul style="list-style-type: none"> • Terminate user session • Redirect to home page or login page after logout
REQ-FR- UFR - 04	Profile Management	Profile Management	Medium	<ul style="list-style-type: none"> • View profile information (name, email, phone, profile picture) • Edit profile details • Change password • Delete account with password confirmation • Prevent unauthorized access to other users' profiles
REQ-FR- UFR - 05	Browse & Search Events	Event Browsing / User	High	<ul style="list-style-type: none"> • View all available events in list or card format • Filter by date, location, price, type • Search by title or keywords
REQ-FR- UFR - 06	View Event Details	Event Details / User	High	<ul style="list-style-type: none"> • Display full event details (description, images, date/time, location, capacity, ticket price, total bookings, display provider info if linked) • Show booking availability • Display provider info if linked
REQ-FR- UFR - 07	Book Event	Booking / User	High	<ul style="list-style-type: none"> • Book an event (requires login) • Generate unique booking reference • Update capacity immediately • Prevent overbooking • Send confirmation email

REQ-FR- UFR - 08	Manage Bookings	Booking / User	High	<ul style="list-style-type: none"> • Display upcoming and past bookings • Cancel bookings and restore capacity • Send cancellation notification • Filter/sort bookings
------------------	-----------------	----------------	------	--

Table 18: UFR User Functional Requirements

3.3.2.1 Use case diagram for Authentication system :

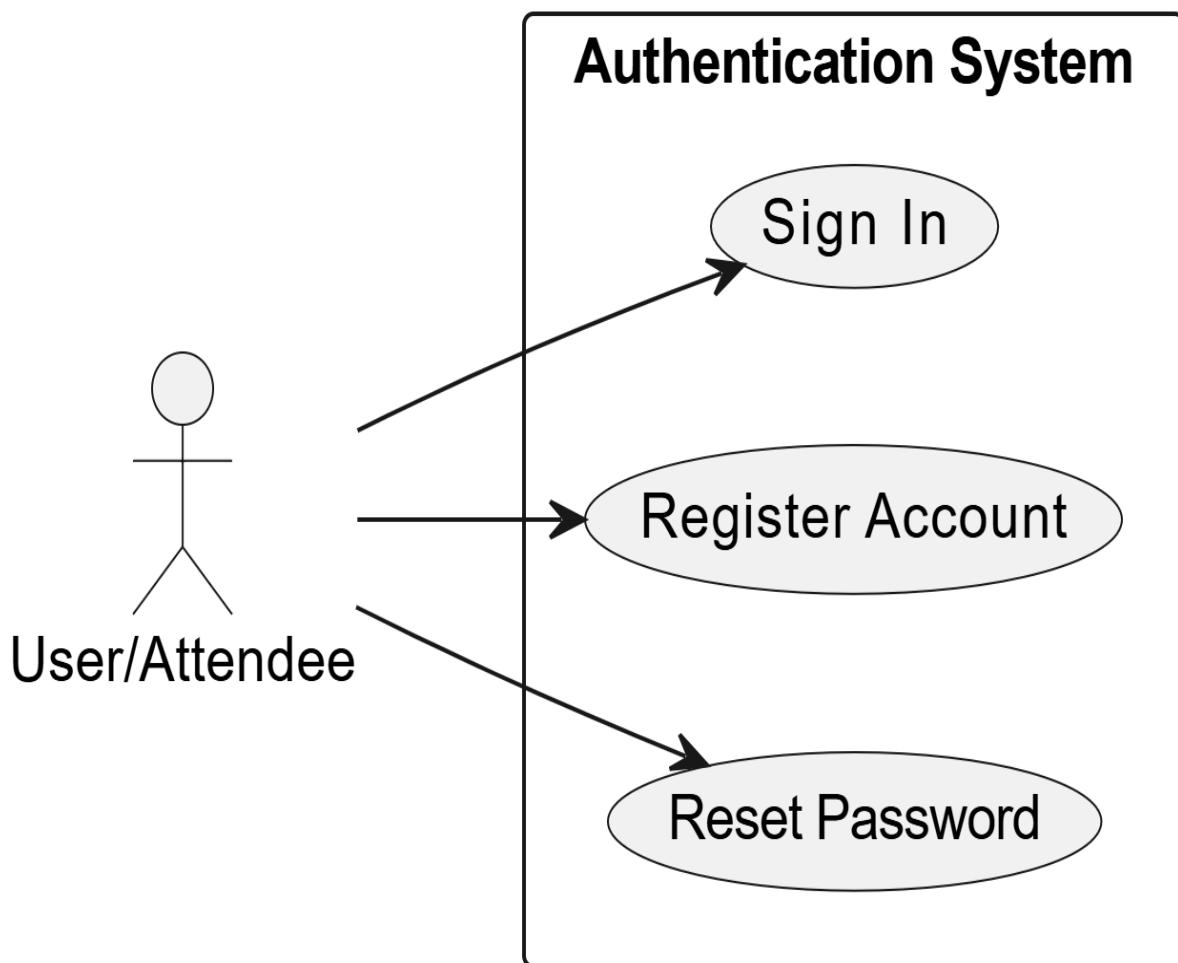


Figure 6: <<Use case 2 (for Authentication system)>>

3.3.2.2 Activity diagram for registration account :

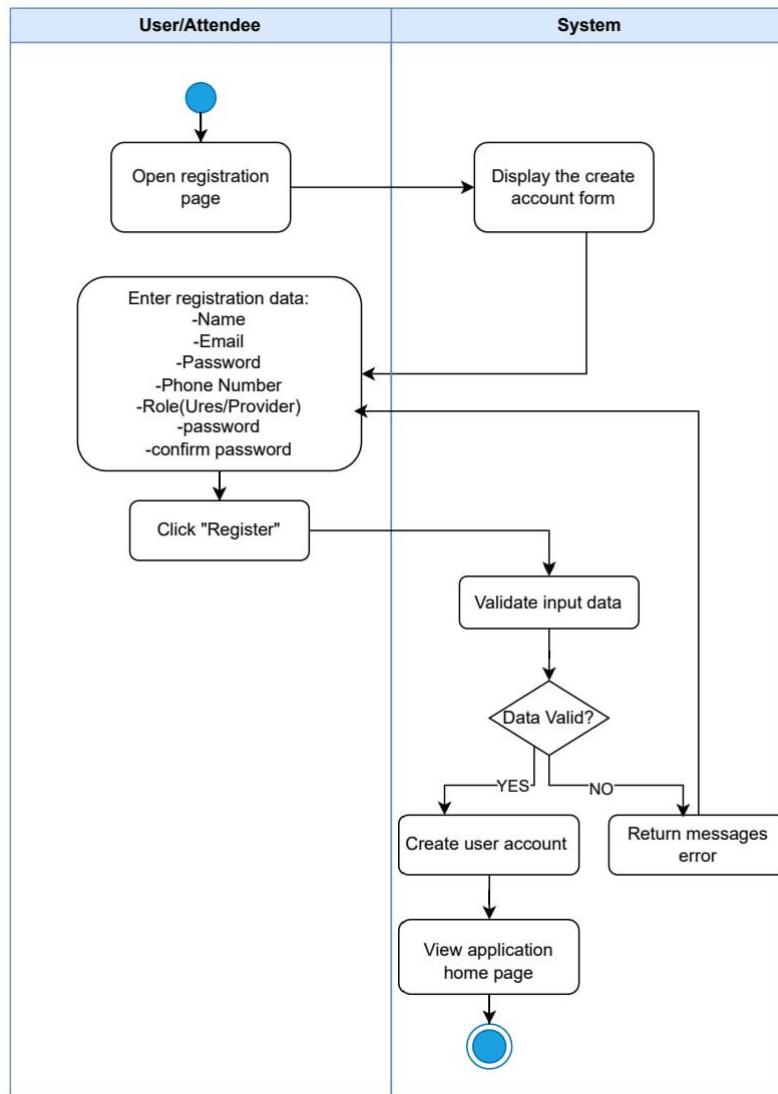


Figure7: activity diagram for registration new account

3.3.2.3 Use case specification:

Use case name:	Register New Account
Participating Actors:	User/Attendee
The flow of events:	<ol style="list-style-type: none"> 1. User/Attendee opens the registration page 2. System displays the create account form 3. User/Attendee enters registration data: <ul style="list-style-type: none"> o Name o Email o Password o Phone Number o Role (User/Provider) o Confirm password 4. User/Attendee clicks "Register" 5. System validates input data 6. System checks if data is valid <ul style="list-style-type: none"> o If valid: <ul style="list-style-type: none"> ▪ System creates user account ▪ System displays the application home page ▪ Use case ends successfully o If invalid: <ul style="list-style-type: none"> ▪ System returns error messages ▪ System redisplays the registration form ▪ User/Attendee can re-enter data
Alternative flows:	<p>A1. Invalid data format:</p> <ul style="list-style-type: none"> • System displays specific validation errors (e.g., "Invalid email format") • User/Attendee remains on registration form

	<ul style="list-style-type: none"> • User/Attendee corrects the data <p>A2. Password mismatch:</p> <ul style="list-style-type: none"> • System displays "Passwords do not match" error • User/Attendee re-enters passwords <p>A3. Email already exists:</p> <ul style="list-style-type: none"> • System displays "Email already registered" error • User/Attendee redirected to sign-in page or uses different email <p>A4. Weak password:</p> <ul style="list-style-type: none"> • System displays password strength requirements • User/Attendee enters stronger password <p>Entry condition:</p> <ol style="list-style-type: none"> 1. User/Attendee does not have an existing account 2. System is operational 3. Registration page is accessible
Entry condition	<ol style="list-style-type: none"> 1. User/Attendee does not have an existing account 2. System is operational 3. Registration page is accessible
Exit conditions	<p>Success:</p> <ul style="list-style-type: none"> • User account created successfully • User/Attendee authenticated automatically • Application home page displayed • Welcome notification sent <p>Failure:</p> <ul style="list-style-type: none"> • Validation errors displayed • User/Attendee remains on registration form • No account created in database • Error messages guide user to correct inputs

Table 19: use case specification 4

3.3.3.2 Activity diagram for login:

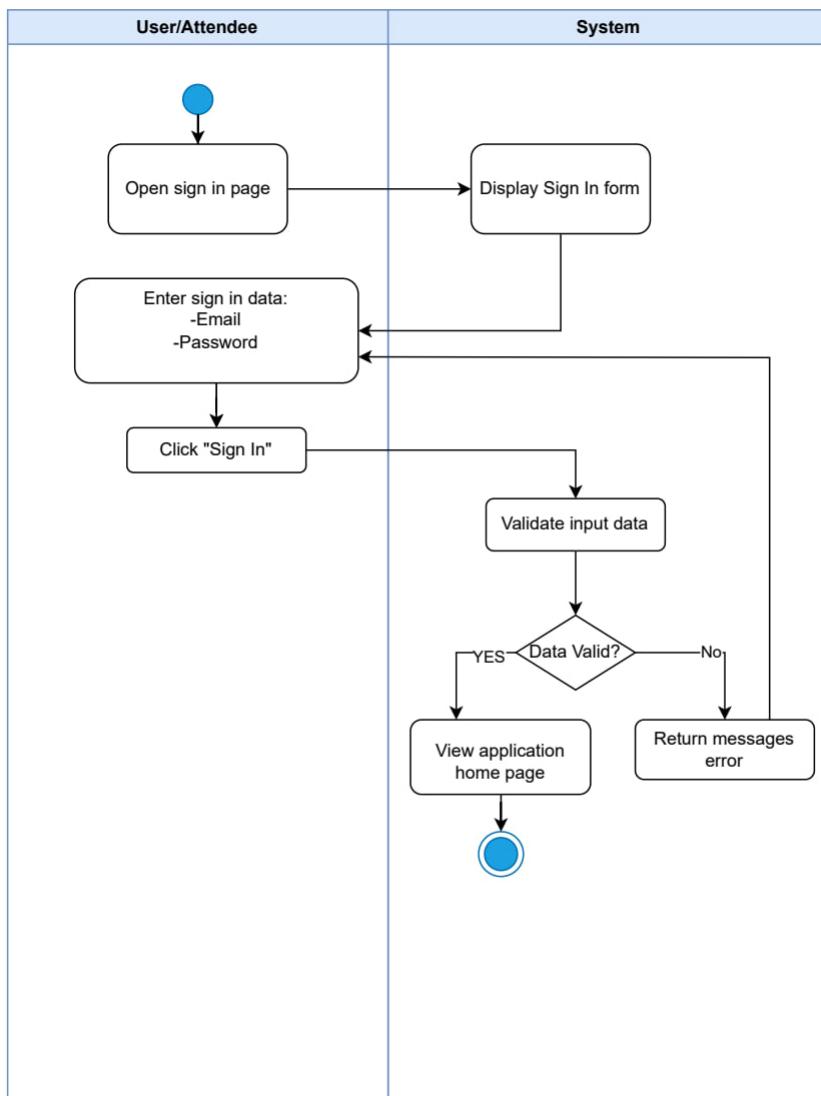


Figure8: activity diagram for Login

3.3.3.3 Use case specification:

Use case name:	Sign In to System
Participating Actors:	User/Attendee
The flow of events:	<ol style="list-style-type: none"> 1. User/Attendee opens the sign-in page 2. System displays the Sign In form 3. User/Attendee enters sign-in data: <ul style="list-style-type: none"> o Email o Password 4. User/Attendee clicks "Sign In" 5. System validates input data 6. System checks if data is valid <ul style="list-style-type: none"> o If valid: <ul style="list-style-type: none"> ▪ System authenticates the user ▪ System displays the application home page ▪ Use case ends successfully o If invalid: <ul style="list-style-type: none"> ▪ System returns error messages ▪ System redisplays the Sign In form ▪ User/Attendee can re-enter credentials
Alternative flows:	<p>A1. Invalid credentials:</p> <ul style="list-style-type: none"> • System displays "Invalid email or password" error • User/Attendee remains on sign-in page • User/Attendee can retry or reset password <p>A2. Empty fields:</p> <ul style="list-style-type: none"> • System displays "Required fields missing" error • User/Attendee prompted to fill all fields

	<p>A3. Account locked/suspended:</p> <ul style="list-style-type: none"> • System displays appropriate error message • User/Attendee cannot proceed to home page
Entry condition	<p>4. User/Attendee has internet connection</p> <p>5. System is operational</p> <p>6. Sign-in page is accessible</p>
Exit conditions	<p>Success:</p> <ul style="list-style-type: none"> • User/Attendee authenticated successfully • Application home page displayed • User session initiated <p>Failure:</p> <ul style="list-style-type: none"> • Validation errors displayed • User/Attendee remains on sign-in page • No session created • Error messages shown to user

Table20: use case specification 5

3.3.4.2 Activity diagram for forget password :

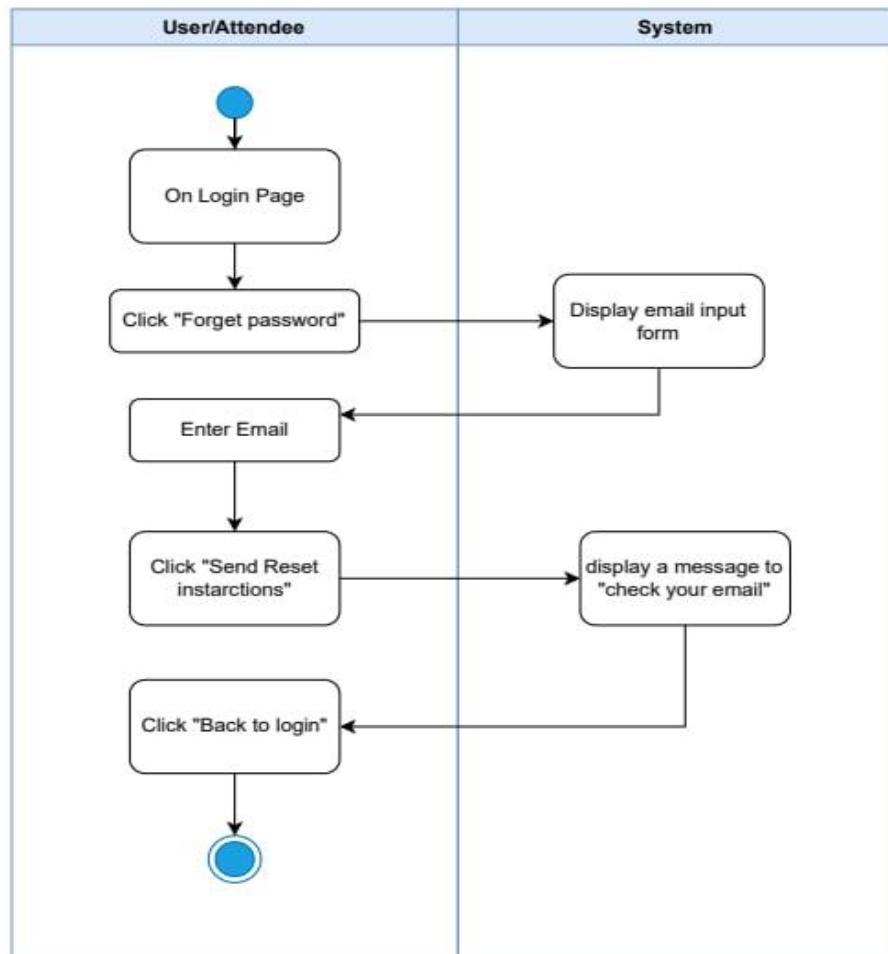


Figure 9: activity diagram for forget password

3.3.4.3 use case specification:

Use case name:	Reset Forgotten Password
Participating Actors:	User/Attendee
The flow of events:	<p>1. User/Attendee is on the login page</p> <p>2. User/Attendee clicks "Forget password"</p> <p>3. System displays email input form</p> <p>4. User/Attendee enters email address</p> <p>5. User/Attendee clicks "Send Reset Instructions"</p> <p>6. System displays message "check your email"</p> <p>7. System sends password reset email to the user</p> <p>8. User/Attendee clicks "Back to login"</p> <p>9. Use case ends</p> <p>▪</p>
Alternative flows:	<p>A1. Email not found:</p> <ul style="list-style-type: none"> System displays "Email not registered" error User/Attendee can retry with different email or register new account <p>A2. Invalid email format:</p> <ul style="list-style-type: none"> System displays "Invalid email format" error User/Attendee corrects email address <p>A3. Email delivery failure:</p> <ul style="list-style-type: none"> System logs the error User/Attendee may not receive email User/Attendee can request resend <p>A4. User cancels operation:</p> <ul style="list-style-type: none"> User/Attendee clicks "Back to login" before submitting No email sent

	<ul style="list-style-type: none"> • Returns to login page
Entry condition	<ol style="list-style-type: none"> 1. User/Attendee has forgotten their password 3. User/Attendee is on the login page 4. System is operational 5. Email service is available
Exit conditions	<p>Success:</p> <ul style="list-style-type: none"> • Password reset email sent successfully • User/Attendee receives instructions in email • User/Attendee returns to login page • System logs the reset request <p>Failure:</p> <ul style="list-style-type: none"> • Email not sent due to invalid address • Error message displayed to user • User/Attendee remains on password reset form • No reset email delivered

Table21: use case specification 6

3.3.3.1 Use case diagram for user profile management

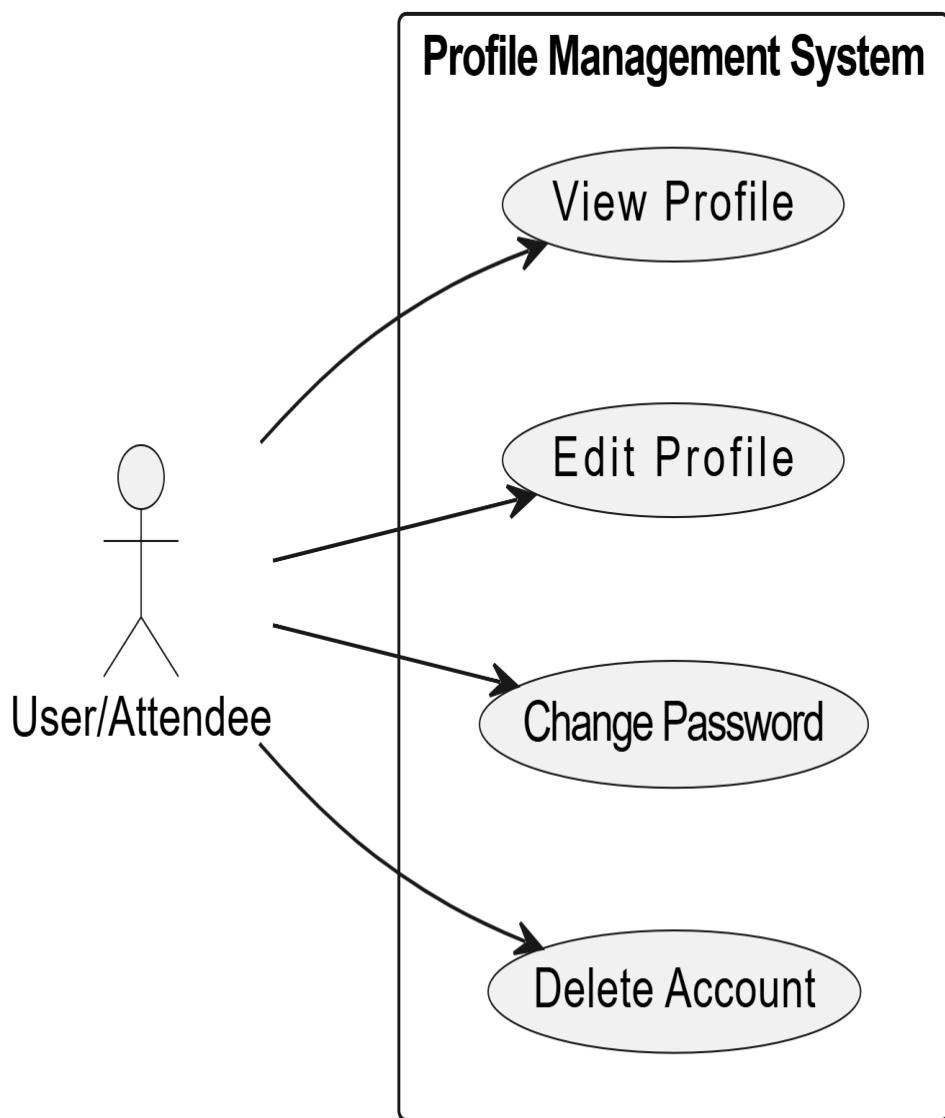


Figure 10: <<use case3(for user profile management)>>

3.3.5.2 activity diagram for user profile management

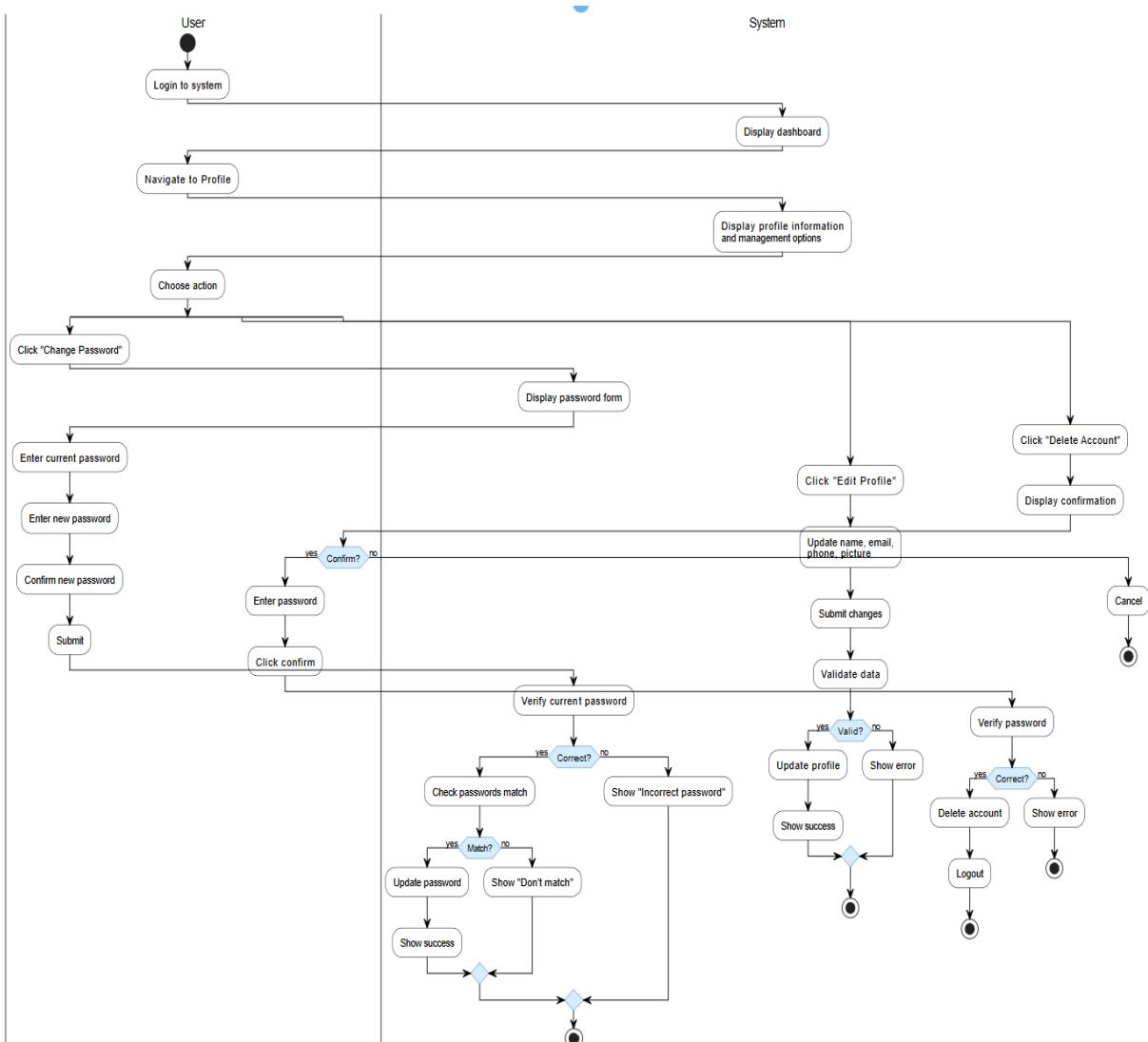


Figure 11: activity diagram for user profile management

3.3.5.3 use case specification:

Use Case Specification for Profile Management

Use Case 1: Edit Profile

Use case name:	Edit User Profile Information
Participating Actors:	User/Attendee
The flow of events:	<ol style="list-style-type: none">1. User logs into the system2. System displays dashboard3. User navigates to Profile4. System displays profile information and management options5. User clicks "Edit Profile"6. System displays edit form with current information7. User updates information:<ul style="list-style-type: none">o Nameo Emailo Phone numbero Profile picture8. User submits changes9. System validates data10. System updates profile information11. System displays success message
Alternative flows:	<p>A1. Invalid data entered:</p> <ul style="list-style-type: none">• System displays validation error• User must correct errors• User remains on edit form <p>A2. Email already exists:</p> <ul style="list-style-type: none">• System displays "Email already in use" error• User must use different email

	<p>A3. User cancels edit:</p> <ul style="list-style-type: none"> • System returns to profile page • No changes saved
Entry condition	<ol style="list-style-type: none"> 1. User is authenticated and logged in 2. System is operational
Exit conditions	<p>Success:</p> <ul style="list-style-type: none"> • Profile information updated successfully • Success message displayed • Updated information visible in profile <p>Failure:</p> <ul style="list-style-type: none"> • Validation errors displayed • Invalid data entered • No changes saved to database

Table22: use case specification 7

Use Case 2: Change Password

Use case name:	Edit User Profile Information
Participating Actors:	User/Attendee
The flow of events:	<ol style="list-style-type: none"> 1. User logs into the system 2. System displays dashboard 3. User navigates to Profile 4. System displays profile information and management options 5. User clicks "Change Password" 6. System displays password change form 7. User enters current password 8. User enters new password 9. User confirms new password 10. User clicks "Submit" 11. System verifies current password is correct 12. System checks if new passwords match 13. System updates password in database 14. System displays success message
Alternative flows:	<p>A1. Current password incorrect:</p> <ul style="list-style-type: none"> • System displays error: "Incorrect current password" • User must enter correct current password • User remains on password form <p>A2. New passwords don't match:</p> <ul style="list-style-type: none"> • System displays error: "Passwords don't match" • User must re-enter matching passwords • User remains on password form <p>A3. New password doesn't meet requirements:</p> <ul style="list-style-type: none"> • System displays validation error

	<ul style="list-style-type: none"> • User must enter valid password <p>A4. User cancels:</p> <ul style="list-style-type: none"> • System returns to profile page • Password remains unchanged •
Entry condition	<p>3. User is authenticated and logged in</p> <p>4. System is operational</p>
Exit conditions	<p>Success:</p> <ul style="list-style-type: none"> • Password updated successfully • Success message displayed • User can login with new password <p>Failure:</p> <ul style="list-style-type: none"> • Current password incorrect • New passwords don't match • Validation errors displayed • Password remains unchanged

Table23: use case specification 8

Use Case 3: Delete user account

Use case name:	Delete User Account
Participating Actors:	User/Attendee
The flow of events:	<p>15. User logs into the system</p> <p>16. System displays dashboard</p> <p>17. User navigates to Profile</p> <p>18. System displays profile information and management options</p> <p>19. User clicks "Delete Account"</p> <p>20. System displays confirmation dialog with warning message</p> <p>21. User confirms deletion</p> <p>22. System prompts user to enter password for confirmation</p> <p>23. User enters password</p> <p>24. User clicks "Confirm Delete"</p> <p>25. System verifies password is correct</p> <p>26. System deletes user account</p> <p>27. System deletes all associated user data</p> <p>28. System logs out user</p> <p>29. System displays success message or redirects to homepage</p>
Alternative flows:	<p>A1. User cancels deletion:</p> <ul style="list-style-type: none"> • User clicks "Cancel" on confirmation dialog • System returns to profile page • No account deletion occurs <p>A2. Incorrect password entered:</p> <ul style="list-style-type: none"> • System displays error: "Incorrect password" • User must enter correct password • Account not deleted

	<p>A3. User has active bookings:</p> <ul style="list-style-type: none"> • System displays warning about active bookings • User can still proceed or cancel (business rule dependent)
Entry condition	<ol style="list-style-type: none"> 1. User is authenticated and logged in 2. System is operational 1. User has valid account
Exit conditions	<p>Success:</p> <ul style="list-style-type: none"> • User account deleted successfully • All user data removed from database • User logged out automatically • Success message displayed <p>Failure:</p> <ul style="list-style-type: none"> • Incorrect password entered • User cancels deletion • Account remains active • User remains logged in

Table24: use case specification 9

3.3.4.1 Use case diagram for event:

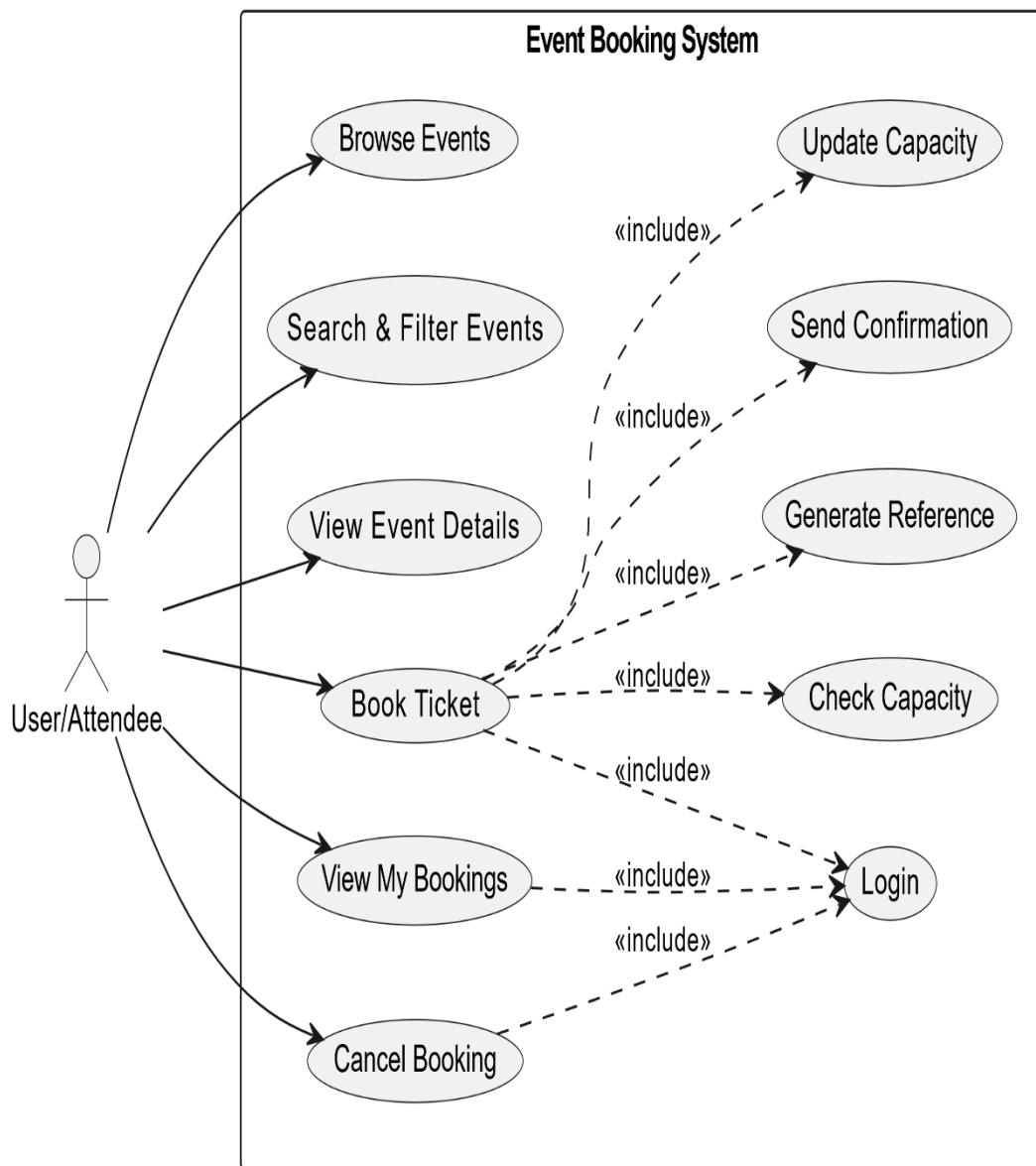


Figure 12: <<use case 4 (for events)>>

3.3.5.1 Use case diagram for package booking:

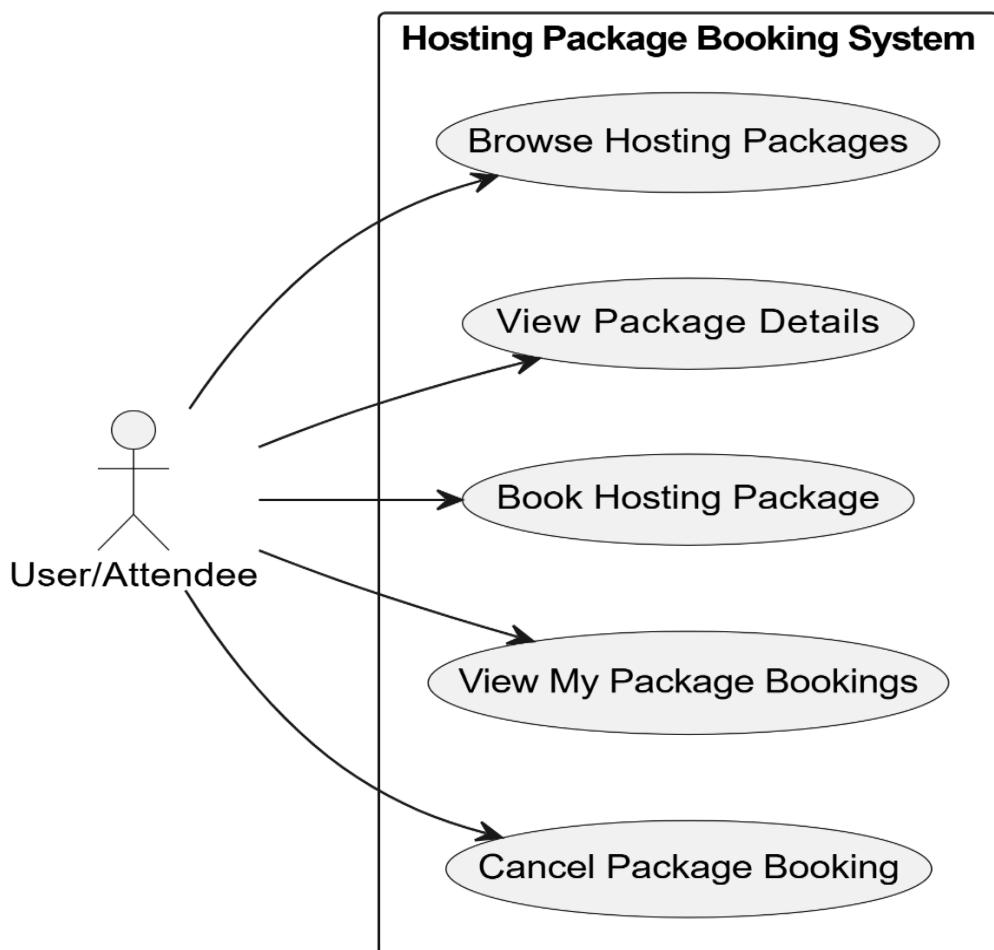


Figure 13: use case 4 for package booking

3.3.6.1 Use case diagram for Notification System:

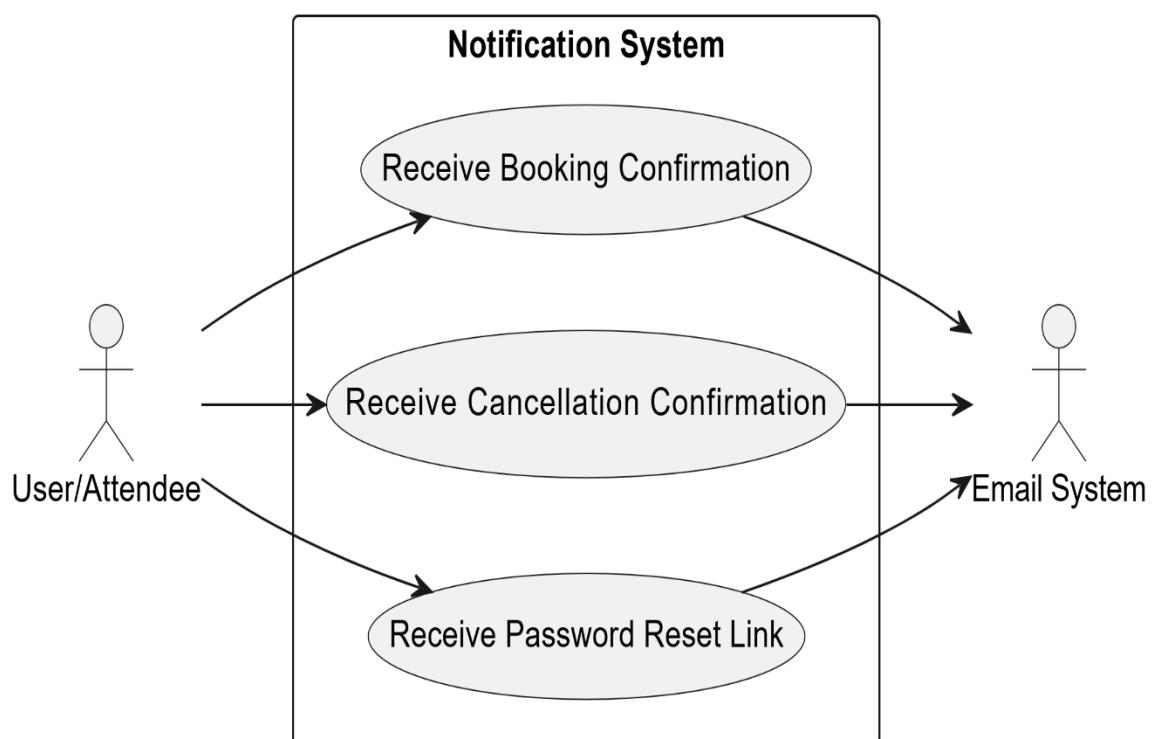


Figure 14: <<use case 6 (for Notification System)>>

3.3.6.2 Activity diagram for event discovery & booking activity diagram

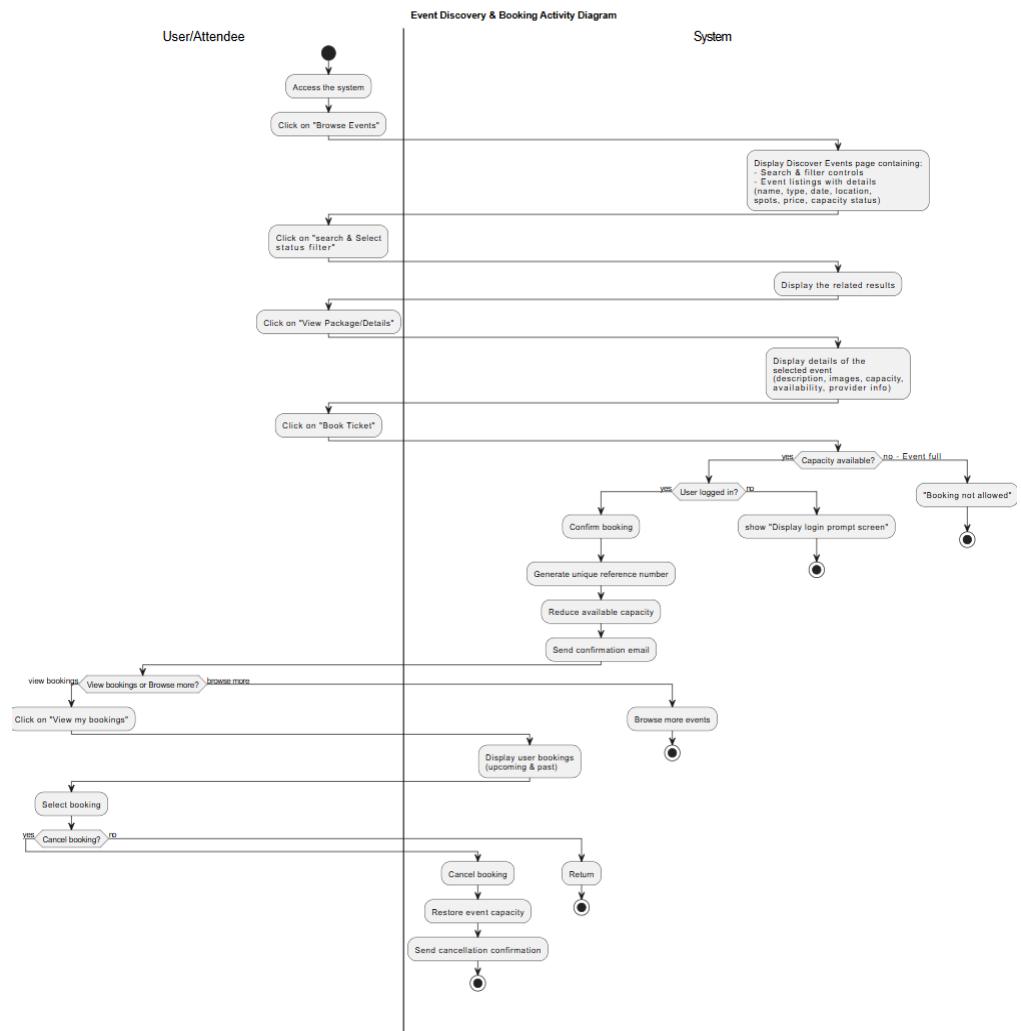


Figure15: Activity diagram for event discovery & booking activity diagram

3.3.6.3 use case specification:

3.3.2.2 Use Case specification

Use case name:	Event Discovery and Booking
Participating Actors:	End User
The flow of events:	<p>Main Flow:</p> <ol style="list-style-type: none"> 1. User accesses the system 2. User clicks on "Browse Events" 3. System displays Discover Events page containing: <ul style="list-style-type: none"> o Search & filter controls (search field, city, type, date, price) o Event listings (event name, type, date, location, spots, price, capacity status) 4. User clicks on "search & Select status filter" 5. System displays the related results 6. User clicks on "View Package/Event" 7. System displays details of the selected event: <ul style="list-style-type: none"> o Description, images, capacity, availability o Ticket price, supplier information (if tagged) 8. User clicks on "Book package/event" 9. System checks if capacity is available 10. System checks if user is logged in 11. System confirms booking 12. System generates unique booking reference number 13. System reduces available capacity immediately 14. System sends booking confirmation email with reference number 15. User chooses to view bookings or browse more events <p>Alternative Flow A (View My Bookings):</p> <ol style="list-style-type: none"> 1. User clicks on "View my bookings" 2. System displays user bookings with details (upcoming & past bookings) 3. User selects a booking 4. User decides to cancel or return 5. If cancel: <ul style="list-style-type: none"> o System cancels booking o System restores event capacity o System sends cancellation confirmation message <p>Alternative Flow B (Browse More Events):</p> <ol style="list-style-type: none"> 1. System returns to browse more events page
Alternative flows:	<p>A1. Event capacity exceeded:</p> <ul style="list-style-type: none"> • System shows error message: "Event capacity exceeded" • User cannot proceed with booking <p>A2. User not logged in:</p> <ul style="list-style-type: none"> • System shows error message: "Please login first" • User redirected to login page <p>A3. Empty search/filter:</p> <ul style="list-style-type: none"> • System displays all available events

	<p>A4. No events match filter criteria:</p> <ul style="list-style-type: none"> • System displays "No events found" message <p>A5. Booking cancellation:</p> <ul style="list-style-type: none"> • User cancels booking from "My Bookings" • System restores event capacity • System sends cancellation confirmation
Entry condition	<p>1- System is operational 2-Events are available in the system 3- User has access to the system</p>
Exit conditions	<p>Success:</p> <ul style="list-style-type: none"> • Event booking confirmed successfully • Unique reference number generated • Confirmation email sent • Event capacity updated • User can view booking in "My Bookings" <p>Success (Cancel Booking):</p> <ul style="list-style-type: none"> • Booking canceled successfully • Event capacity restored • Cancellation confirmation sent <p>Failure:</p> <ul style="list-style-type: none"> • Error message displayed (capacity exceeded / not logged in) • User remains on current page • No changes to event capacity or booking status

Table25: use case specification 10

3. Create Public Event & Package Functional Requirements:

Requirements ID	Description	Module	Priority	Detailed Specification
REQ-FR-CPEP-01	Create Public Event	Event Management	High	<ul style="list-style-type: none"> Allows organizer to create a new public event Mandatory fields: title, description, location, future date, time, capacity, ticket price Validate all inputs before saving
REQ-FR-CPEP-02	Edit Event	Event Management	High	<ul style="list-style-type: none"> Allows organizer to edit existing events Update title, description, location, date/time, capacity, price
REQ-FR-CPEP-03	Delete Event	Event Management	Medium	<ul style="list-style-type: none"> Allows organizer to delete events Requires confirmation before deletion Notify participants via email if deleted
REQ-FR-CPEP-04	Create Host Package	Event Management	Medium	<ul style="list-style-type: none"> Allows organizer to open or close event booking (Open/Close)
REQ-FR-CPEP-05	Edit Host Package	Host Package Management	Medium	<ul style="list-style-type: none"> Allows organizer to create a new host package Mandatory fields: title, description, base setup details, location options, package price
REQ-FR-CPEP-06	Delete Host Package	Host Package Management	Medium	<ul style="list-style-type: none"> Allows organizer to edit host packages after creation
REQ-FR-CPEP-07	Enable/Disable Host Package Booking	Host Package Management	Medium	<ul style="list-style-type: none"> Allows organizer to delete host packages Requires confirmation
REQ-FR-CPEP-08	Manage Bookings	Host Package Management	Medium	<ul style="list-style-type: none"> Allows organizer to open or close booking for host packages
REQ-FR-CPEP-09	Display Event Dashboard	Event Dashboard	High	<ul style="list-style-type: none"> Show all events created by the organizer Display type, city, date, status, bookings, statistics, revenue, available actions
REQ-FR-CPEP-10	Display Per-Event Statistics	Analytics	High	<ul style="list-style-type: none"> Show statistics per event: total bookings, remaining seats, total revenue
REQ-FR-CPEP-11	Display General Statistics	Analytics	High	<ul style="list-style-type: none"> Show general statistics: total events created, active events, total overall revenue
REQ-FR-CPEP-12	Booking Notifications	Notifications	High	<ul style="list-style-type: none"> Notify organizer for new bookings Notify when event capacity reaches 80% or 100%
REQ-FR-CPEP-13	Event Update Notifications	Notifications	High	<ul style="list-style-type: none"> Notify attendees if event is canceled, date changed, or location changed
REQ-FR-CPEP-14	Filter Events by Status	Event Dashboard	Medium	<ul style="list-style-type: none"> Filter events by status: All, Active, Canceled, Completed
REQ-FR-CPEP-15	Search Events	Event Dashboard	Medium	<ul style="list-style-type: none"> Search events by title or keywords

Table26: CPEP Functional Requirements

3.3.7.1 Use case diagram for organizer:

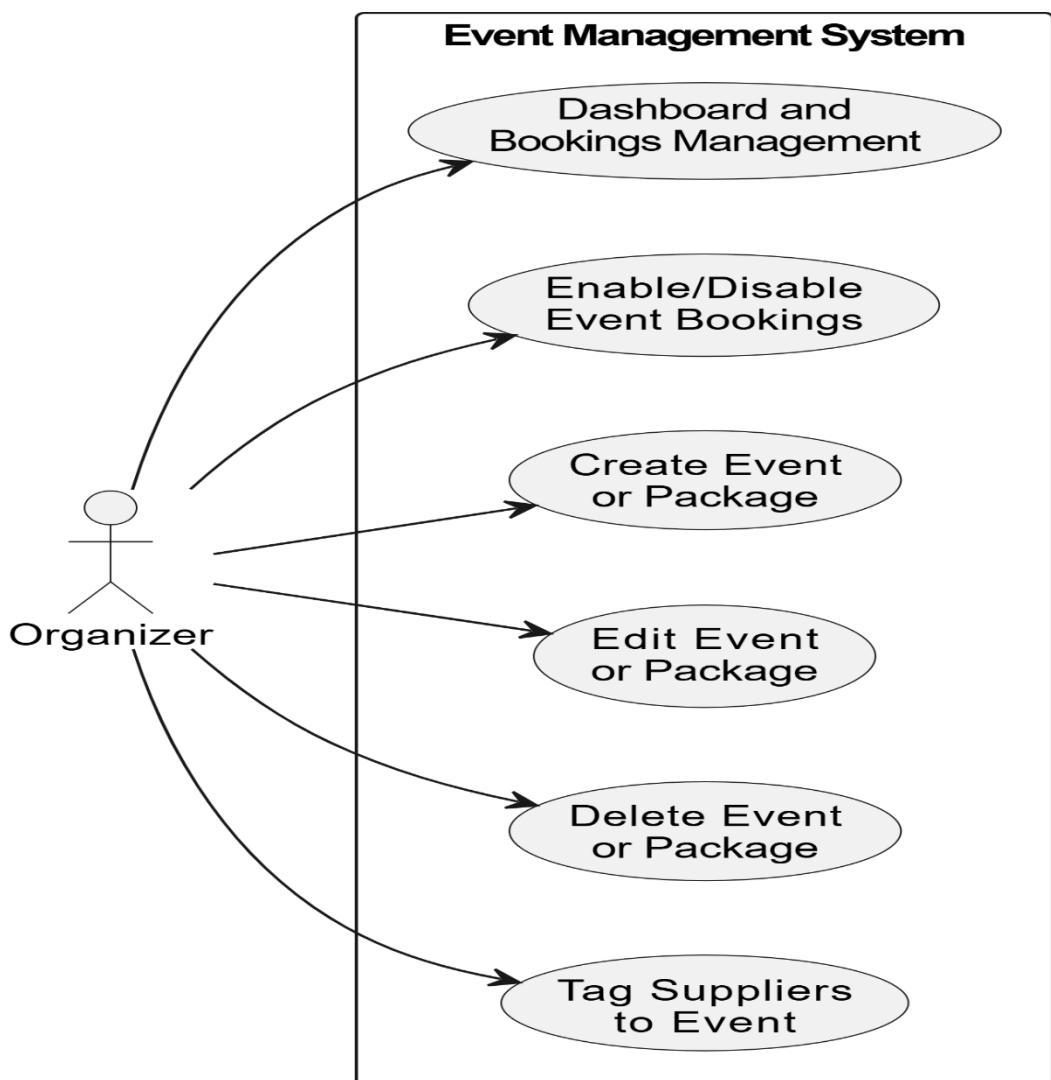


Figure 16:use case 7 for organizer

3.3.7.2 Activity diagram for organizer dashboard:

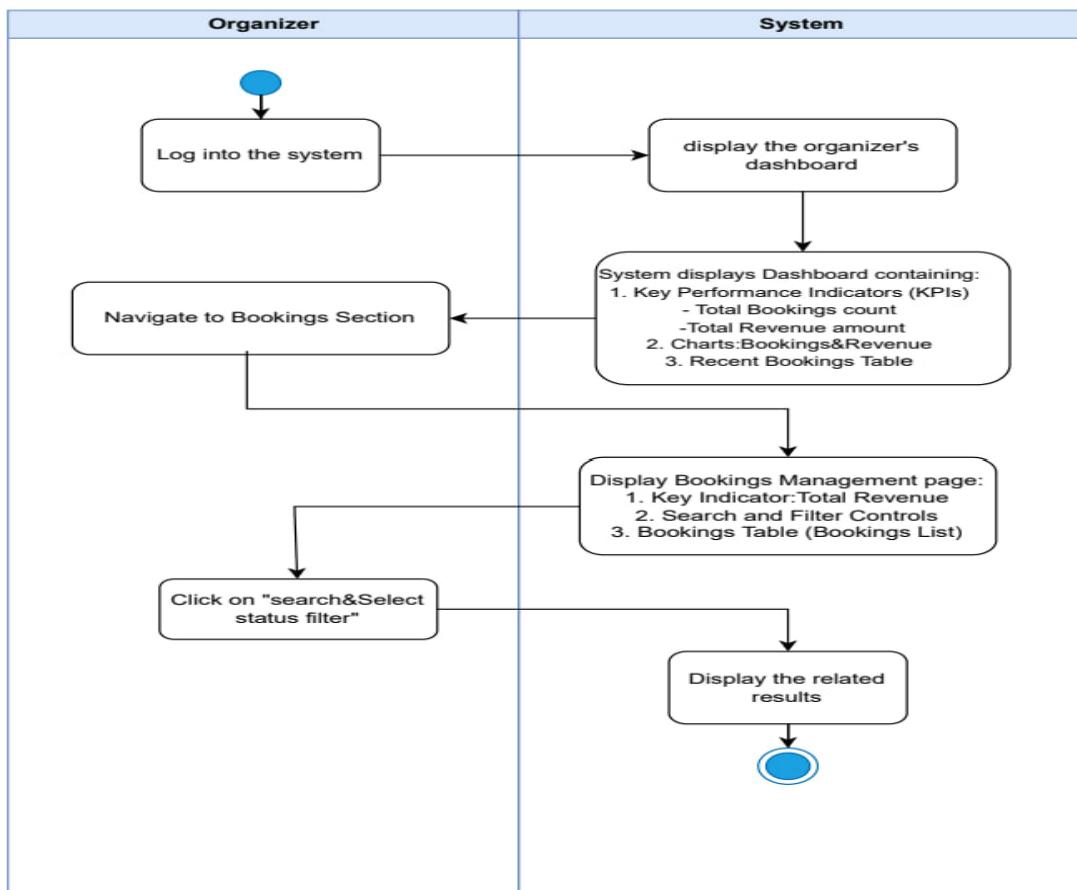


Figure 17:activity diagram for organizer dashboard

3.3.7.3 use case specification:

Use case name:	Organizer Dashboard and Bookings Management
Participating Actors:	Organizer
The flow of events:	<ol style="list-style-type: none"> 1. Organizer logs into the system 2. System displays the organizer's dashboard containing: <ul style="list-style-type: none"> o Key Performance Indicators (KPIs): <ul style="list-style-type: none"> ▪ Total Bookings count ▪ Total Revenue amount o Charts: Bookings & Revenue o Recent Bookings Table 3. Organizer navigates to Bookings Section 4. System displays Bookings Management page containing: <ul style="list-style-type: none"> o Key Indicator: Total Revenue o Search and Filter Controls o Bookings Table (Bookings List) 5. Organizer clicks on "search & Select status filter" 6. System displays the related results
Alternative flows:	<p>A1. No bookings available:</p> <ul style="list-style-type: none"> • System displays "No bookings found" message <p>A2. Empty search/filter:</p> <ul style="list-style-type: none"> • System displays all bookings
Entry condition	<ol style="list-style-type: none"> 1. Organizer is authenticated 2. System is operational
Exit conditions	<p>Success:</p> <ul style="list-style-type: none"> • Dashboard data displayed successfully • Bookings list filtered and displayed • Statistics and charts visible <p>Failure:</p> <ul style="list-style-type: none"> • Error loading dashboard data • No results match filter criteria

Table27: use case specification 11

3.3.8.2 Activity diagram for Disable/Enable Bookings:

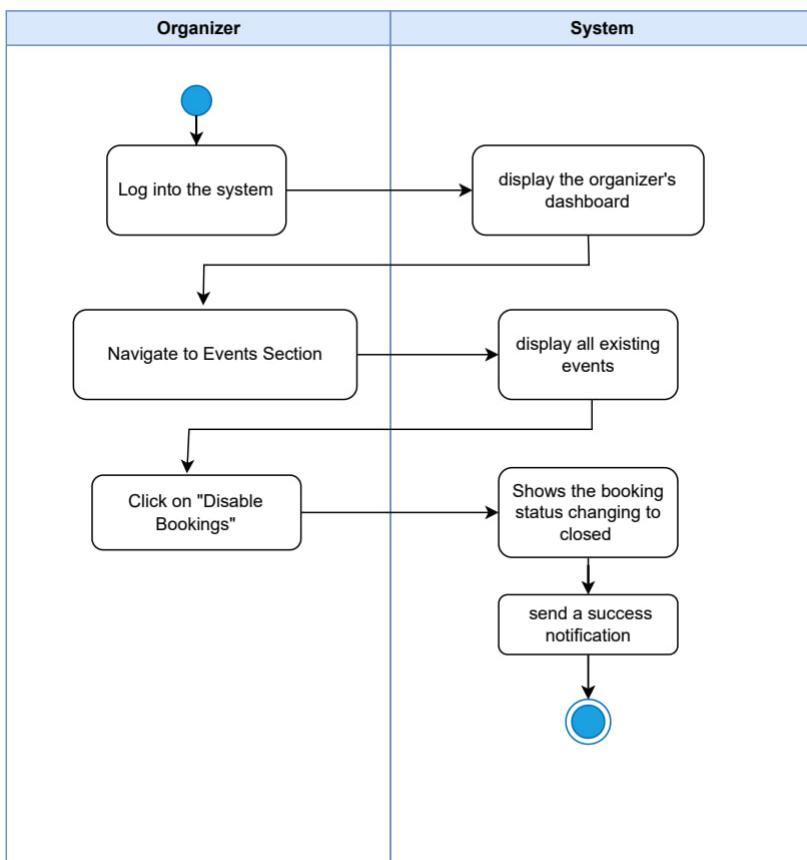


Figure 18: activity diagram for Disable/Enable Bookings

3.3.8.3 use case specification:

Use case name:	Enable or Disable Event Bookings
Participating Actors:	Organizer
The flow of events:	<ol style="list-style-type: none"> 1. Organizer logs into the system 2. System displays the organizer's dashboard 3. Organizer navigates to Events Section 4. System displays all existing events 5. Organizer clicks on "Disable Bookings" button 6. System shows the booking status changing to "closed" 7. System sends a success notification 8. Event booking status updated
Alternative flows:	<p>A1. Enable Bookings:</p> <ul style="list-style-type: none"> • Organizer clicks "Enable Bookings" • System changes status to "open" • Success notification sent <p>A2. No events available:</p> <ul style="list-style-type: none"> • System displays "No events found" message
Entry condition	<ol style="list-style-type: none"> 1. Organizer is authenticated 2. Event exists in the system 3. System is operational
Exit conditions	<p>Success:</p> <ul style="list-style-type: none"> • Booking status changed successfully (Open/Closed) • Success notification displayed • Event updated in database <p>Failure:</p> <ul style="list-style-type: none"> • Error updating booking status • System displays error message

Table28: use case specification 12

3.3.9.2 Activity diagram for Create New Event:

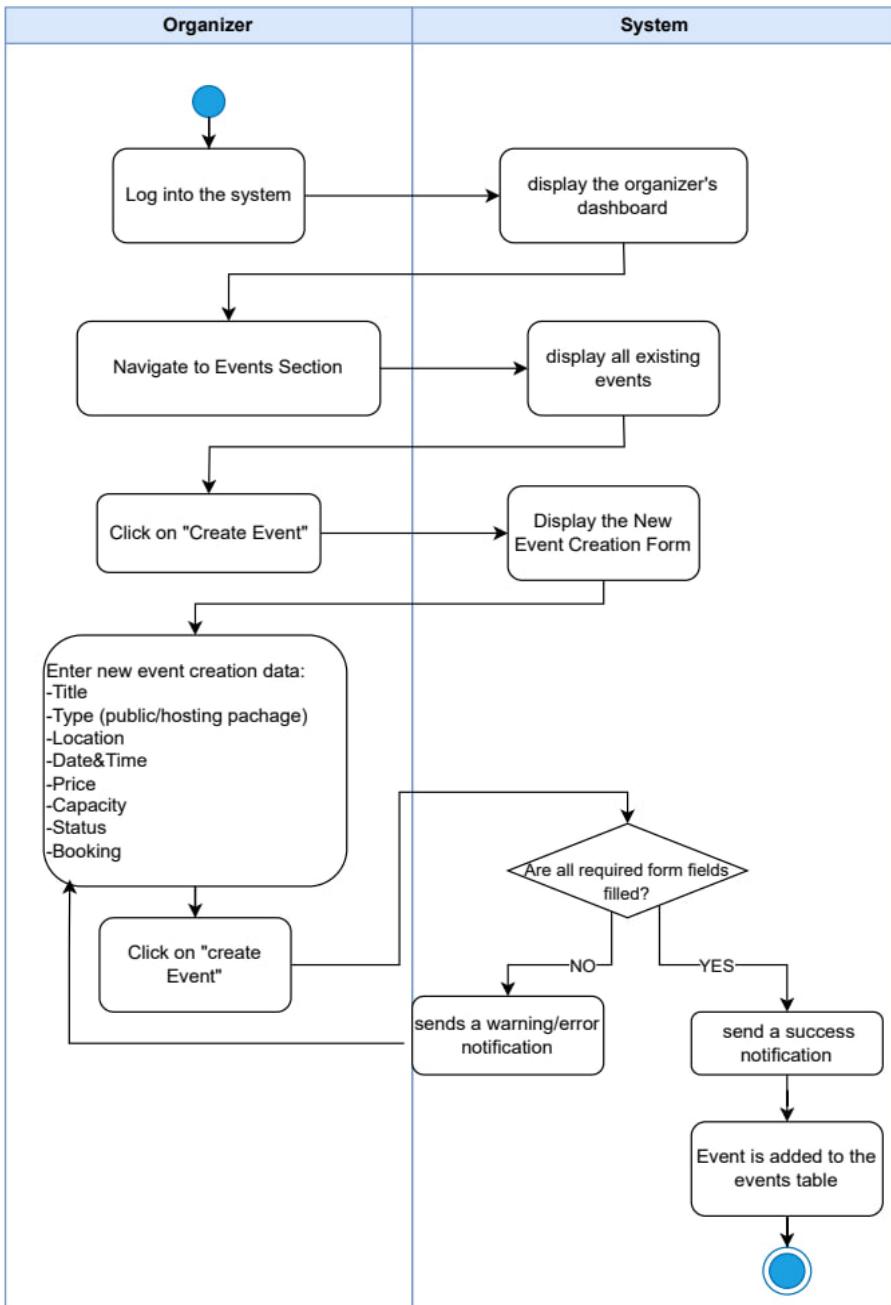


Figure 19: activity diagram Create New Event

3.3.9.3 use case specification:

Use case name:	Create New Public Event or Host Package
Participating Actors:	Organizer
The flow of events:	<ol style="list-style-type: none"> 1. Organizer logs into the system 2. System displays the organizer's dashboard 3. Organizer navigates to Events Section 4. System displays all existing events 5. Organizer clicks on "Create New Event" 6. System displays Create Event Form 7. Organizer selects Event Type (Public Event or Host Package) 8. If Public Event: <ul style="list-style-type: none"> o Organizer enters: Title, Description, Location, Date (future), Time, Capacity, Ticket Price o Optional: Image, Supplier information 9. If Host Package: <ul style="list-style-type: none"> o Organizer enters: Title, Description, Base setup details, Location options, Package Price, Customizable elements o Optional: Image, Supplier information 10. Organizer clicks "Create Event" or "Cancel" 11. If submitted: <ul style="list-style-type: none"> o System validates all required fields are filled o System checks if date is in the future o System saves event to database o System sends success notification o System displays event in events list
Alternative flows:	<p>A1. Missing required fields:</p> <ul style="list-style-type: none"> • System displays error: "Please fill all required fields" • Organizer remains on create form <p>A2. Past date entered:</p> <ul style="list-style-type: none"> • System displays error: "Date must be in the future" • Organizer must correct the date <p>A3. Cancel creation:</p> <ul style="list-style-type: none"> • Organizer clicks "Cancel" • System returns to events list • No event created
Entry condition	<ol style="list-style-type: none"> 6. Organizer is authenticated 7. System is operational 8.
Exit conditions	Success:

	<ul style="list-style-type: none"> • Event/Package created successfully • Success notification displayed • New event visible in events list • Event available for users to browse and book <p>Failure:</p> <ul style="list-style-type: none"> • Validation errors displayed • Required fields missing • Invalid date (past date) • Organizer remains on create form • No event created in database
--	--

Table29: use case specification 13

3.3.10.2 Activity diagram for Edit Event:

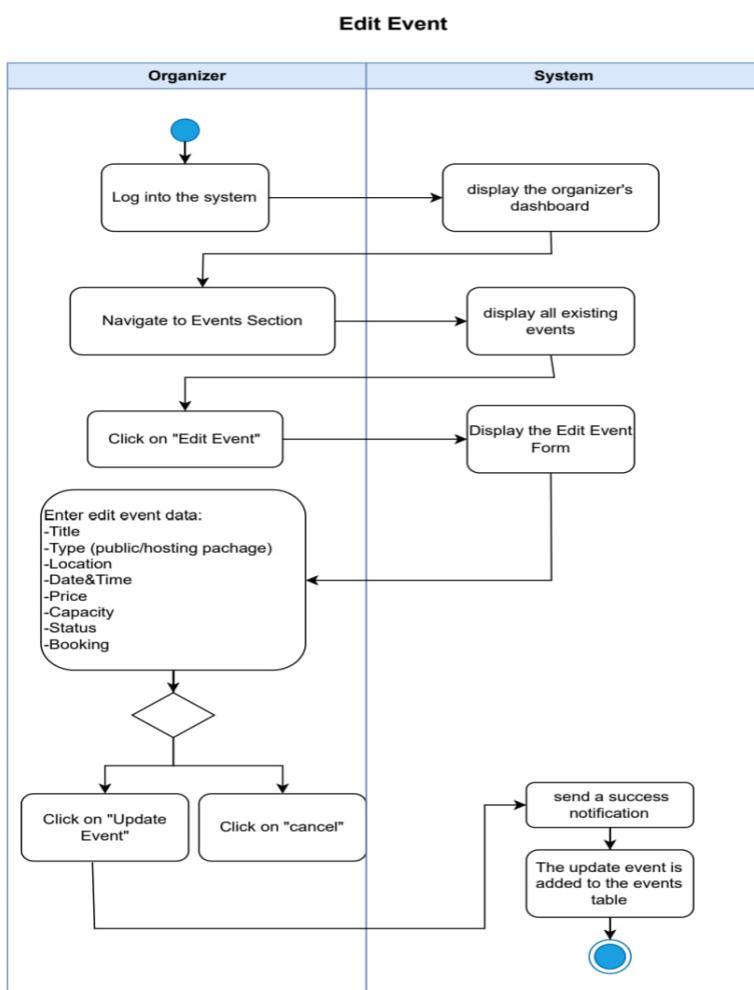


Figure 20: activity diagram Edit Event

3.3.10.3 use case specification:

Use case name:	Edit Existing Event or Package
Participating Actors:	Organizer
The flow of events:	<p>The flow of events:</p> <ol style="list-style-type: none"> 1. Organizer logs into the system 2. System displays the organizer's dashboard 3. Organizer navigates to Events Section 4. System displays all existing events 5. Organizer clicks on "Edit Event" 6. System displays the Edit Event Form with current data 7. Organizer enters edit event data: <ol style="list-style-type: none"> 1. Title 2. Type (public/hosting package) 3. Location 4. Date & Time 5. Price 6. Capacity 7. Status 8. Booking settings 8. Organizer makes a decision to update or cancel 9. If update: <ol style="list-style-type: none"> 1. Organizer clicks "Update Event" 2. System validates inputs 3. System sends a success notification 4. The updated event is reflected in the events table
Alternative flows:	A1. Cancel edit:

	<ul style="list-style-type: none"> • Organizer clicks "cancel" • System returns to events list • No changes saved <p>A2. Invalid data:</p> <ul style="list-style-type: none"> • System displays validation error • Organizer remains on edit form <p>A3. Missing required fields:</p> <ul style="list-style-type: none"> • System prompts to fill required fields •
Entry condition	1.Organizer is authenticated 2.Event exists in the system 3.System is operational
Exit conditions	<p>Success:</p> <ul style="list-style-type: none"> • Event updated successfully • Success notification displayed • Changes reflected in events table <p>Failure:</p> <ul style="list-style-type: none"> • Validation errors displayed • Organizer remains on edit form • No changes saved to database • System displays error message

Table30: use case specification 14

3.3.11.2 Activity diagram for Drop Event:

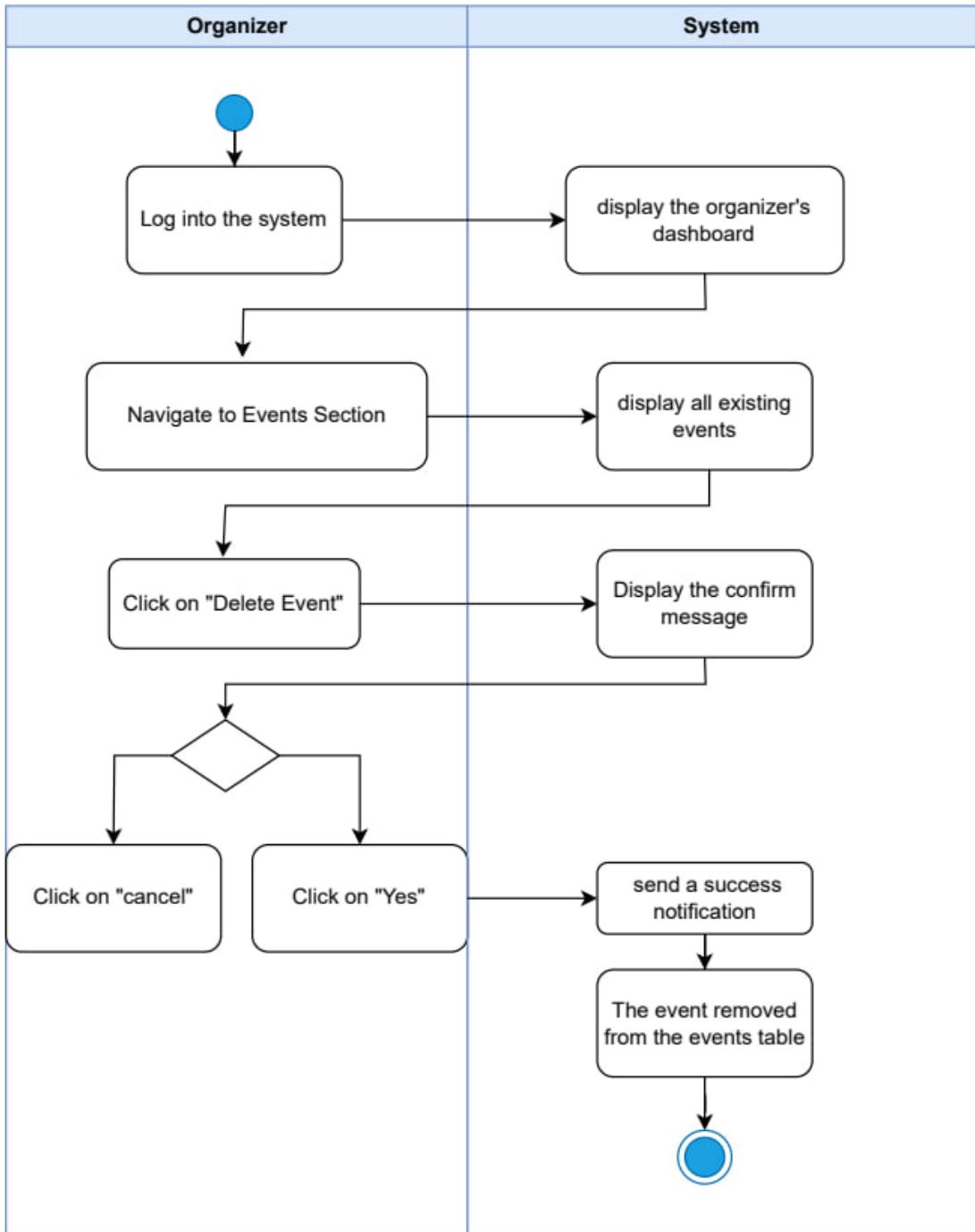


Figure 21: activity diagram Drop Event

3.3.11.3 use case specification:

Use case name:	Delete Event or Package
Participating Actors:	Organizer
The flow of events:	<ol style="list-style-type: none"> 1. Organizer logs into the system 2. System displays the organizer's dashboard 3. Organizer navigates to Events Section 4. System displays all existing events 5. Organizer clicks on "Delete Event" 6. System displays the confirm message: "Are you sure you want to delete this event?" 7. Organizer makes a decision: <ul style="list-style-type: none"> o Click "cancel" to abort o Click "Yes" to confirm deletion 8. If confirmed: <ul style="list-style-type: none"> o System sends a success notification o The event is removed from the events table
Alternative flows:	<p>A1. Cancel deletion:</p> <ul style="list-style-type: none"> • Organizer clicks "cancel" • System returns to events list • Event remains unchanged <p>A2. Event has active bookings:</p> <ul style="list-style-type: none"> • System displays warning message • Organizer can still proceed with deletion (optional business rule)
Entry condition	<ol style="list-style-type: none"> 9. Organizer is authenticated 10. Event exists in the system 11. System is operational

Exit conditions	<p>Success:</p> <ul style="list-style-type: none"> • Event deleted successfully • Success notification displayed • Event removed from database and events table <p>Failure:</p> <ul style="list-style-type: none"> • Deletion canceled by organizer • Event remains in system •
------------------------	--

Table31: use case specification 15

3.3.8.1 Use case diagram for provider:

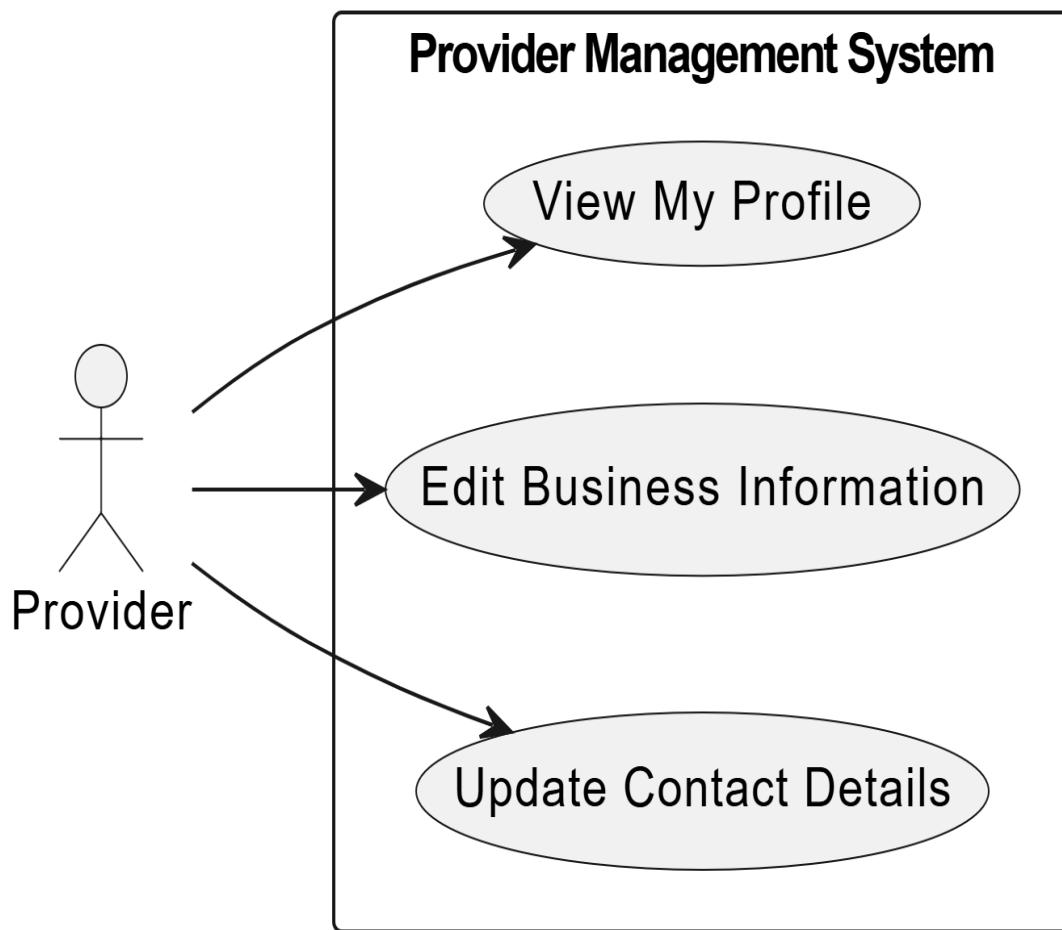


Figure 22.use case 8 for organizer

3.3.12.2Activity diagram for View My Profile :

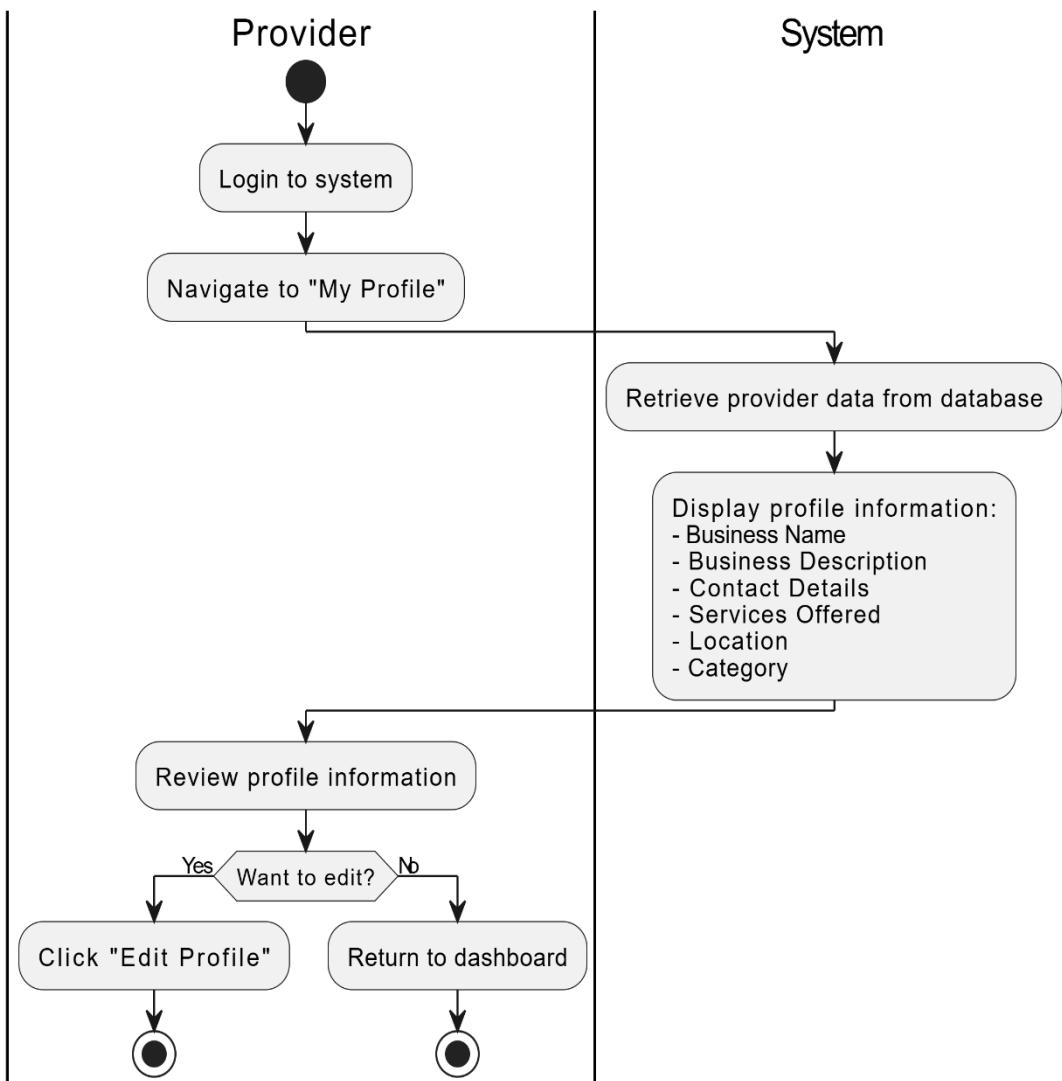


Figure 23 : Activity diagram for View My Profile

3.3.12.3 use case specification:

Use Case 1: View My Profile

Use case name:	View My Profile
Participating Actors:	Provider
The flow of events:	<ol style="list-style-type: none"> 1. Provider logs in to the system. 2. Provider navigates to the "My Profile" section. 3. The system retrieves the provider's data from the database. 4. The system displays the profile information, including: <ul style="list-style-type: none"> - Business Name - Business Description - Contact Details - Services Offered - Location - Category 5. Provider reviews the displayed information. 6. Provider chooses not to edit the profile. 7. The system returns the provider to the main dashboard.
Alternative flows:	<p>A1. Provider wants to edit:</p> <ul style="list-style-type: none"> • In step 6, if the provider chooses to edit, they click "Edit Profile". • The use case transitions to the "Edit Personal Information" or "Edit Business Information" use case.
Entry condition	<ol style="list-style-type: none"> 1. Provider has a valid account and is logged into the system. 2. Provider's profile data exists in the database. 3. The system is operational.
Exit conditions	<p>Success:</p> <ul style="list-style-type: none"> • Profile information is displayed correctly. • Provider returns to the dashboard. <p>Failure:</p> <ul style="list-style-type: none"> • Profile data cannot be retrieved or displayed. • Provider remains on the profile page or sees an error.

Table32: use case specification 16

3.3.13.2 Activity diagram for Edit Personal Information:

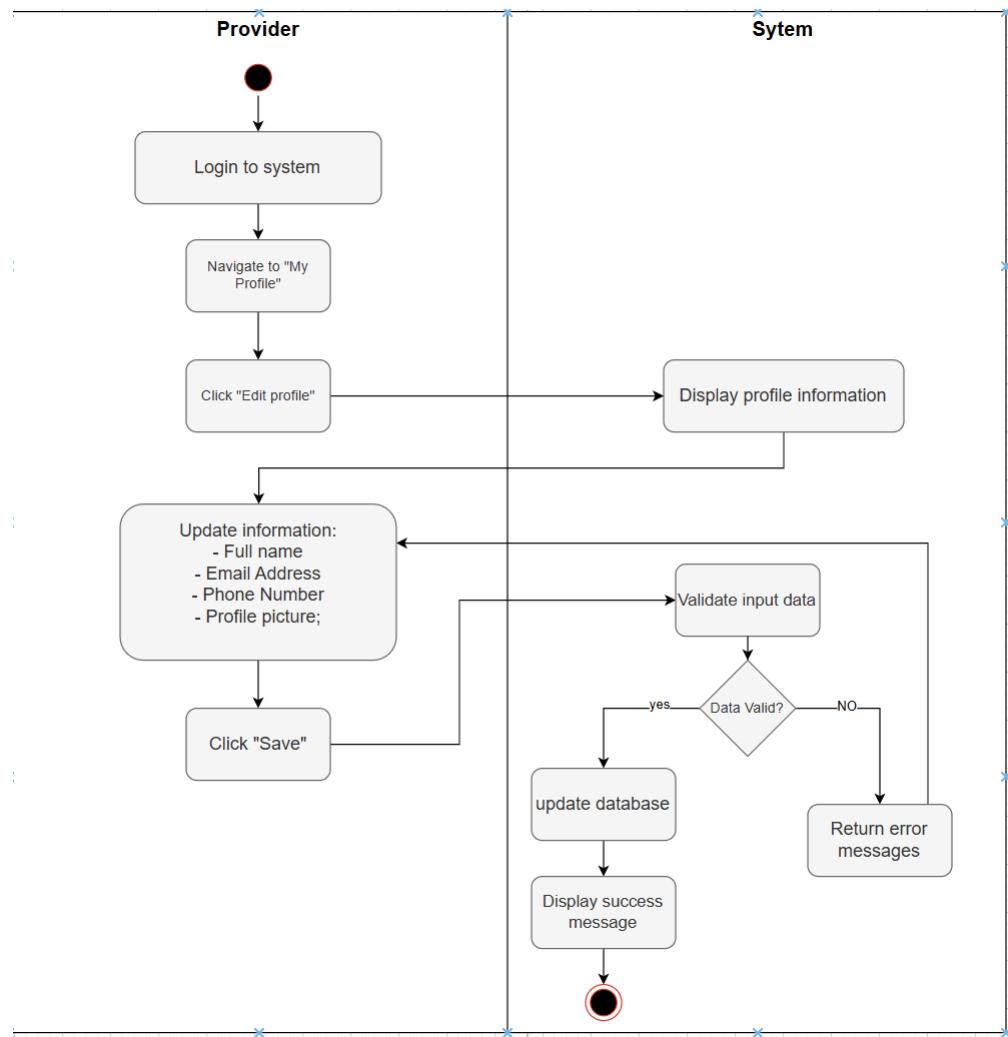


Figure 24 :Activity diagram for Edit Personal Information

Use Case 2: Edit Personal Information

Use case name:	Edit Personal Information
Participating Actors:	Provider
The flow of events:	<ol style="list-style-type: none"> 1. Provider logs in to the system. 2. Provider navigates to "My Profile". 3. Provider clicks on the "Edit Profile" button (for personal info). 4. The system displays an edit form pre-filled with current data. 5. Provider updates their personal information: <ul style="list-style-type: none"> - Full Name - Email Address - Phone Number - Profile Picture 6. Provider clicks "Save". 7. The system validates the entered data. 8. If data is valid, the system updates the database and displays a success message.
Alternative flows:	<p>A1. Invalid Data:</p> <ul style="list-style-type: none"> • In step 7, if the system finds invalid data (e.g., wrong email format), • The system does not update the database and instead displays relevant error messages. • The provider corrects the errors and resubmits.
Entry condition	<ol style="list-style-type: none"> 1. Provider is logged into the system and viewing their profile. 2. The "Edit Profile" functionality is accessible
Exit conditions	<p>Success:</p> <ul style="list-style-type: none"> • Personal information is successfully updated in the database. • Success message is shown to the provider. <p>Failure:</p> <ul style="list-style-type: none"> • Data validation fails. • Error messages are displayed, and no changes are saved to the database.

Table33: use case specification 17

3.3.14.2 Activity diagram for Edit Business Information:

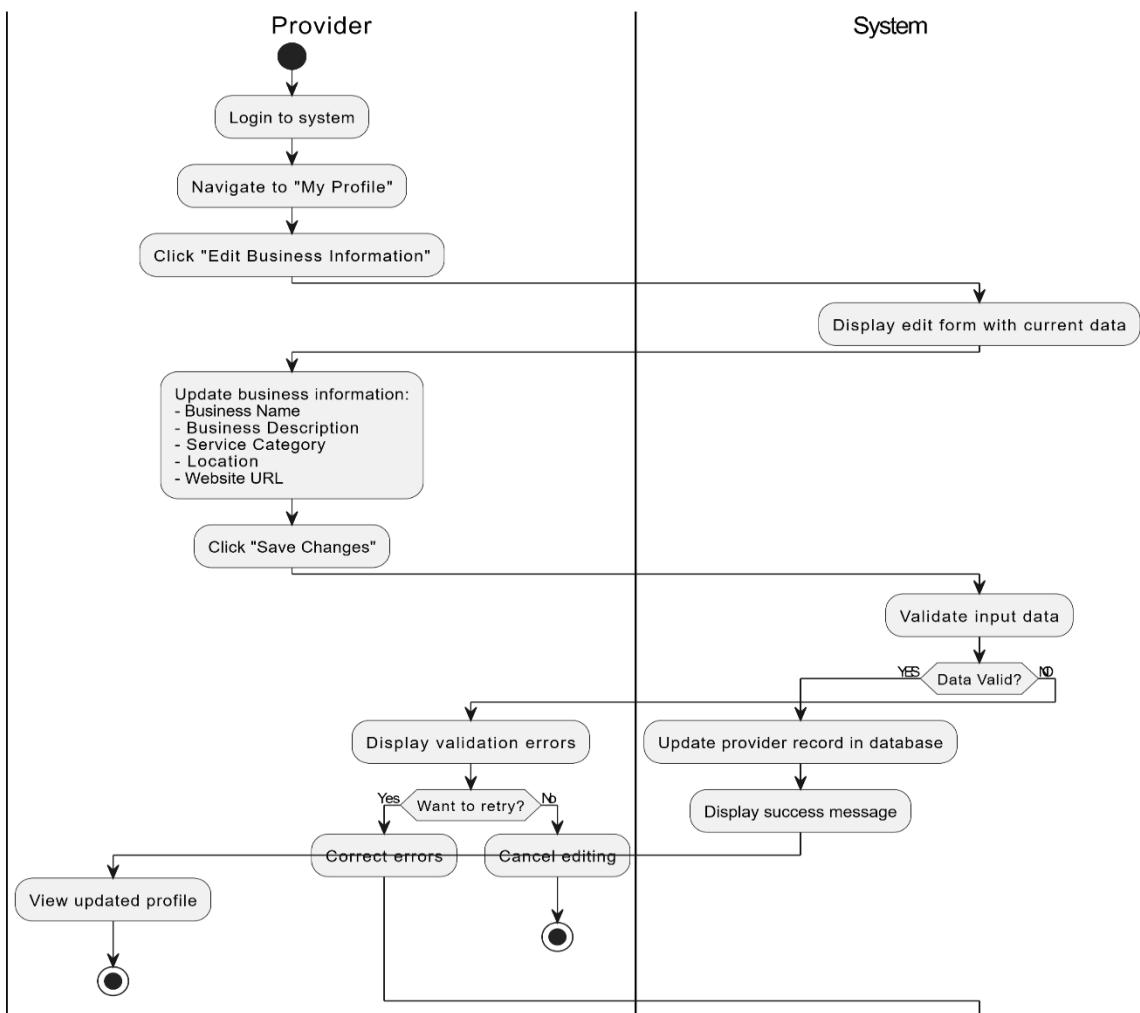


Figure 25 :Activity diagram for Edit Business Information

Use Case 3: Edit Business Information

Use case name:	Edit Business Information
Participating Actors:	Provider
The flow of events:	<ol style="list-style-type: none"> 1. Provider logs in to the system. 2. Provider navigates to "My Profile". 3. Provider clicks on "Edit Business Information". 4. The system displays an edit form pre-filled with current business data. 5. Provider updates the business information: <ul style="list-style-type: none"> - Business Name - Business Description - Service Category - Location - Website URL 6. Provider clicks "Save Changes". 7. The system validates the input data. 8. If data is valid, the system updates the database, displays a success message, and shows the updated profile.
Alternative flows:	<p>A1. Invalid Data:</p> <ul style="list-style-type: none"> • In step 7, if the system finds invalid data, it displays validation error messages. • The system prompts: "Want to retry?" • If Provider chooses to retry: Returns to step 4 with errors highlighted so the provider can correct them. • If Provider chooses to cancel: The editing process is cancelled, no changes are saved, and the provider returns to the profile view.
Entry condition	<ol style="list-style-type: none"> 1. Provider is logged into the system. 2. Provider is viewing "My Profile" page. 3. Business information data exists for this provider.
Exit conditions	<p>Success:</p> <ul style="list-style-type: none"> • Business information is successfully validated and updated in the database. • Success message is displayed. • Provider views their updated profile. <p>Failure:</p> <ul style="list-style-type: none"> • Data validation fails and the provider chooses to cancel editing. • No changes are made to the business information in the database. • Provider returns to the profile view without updates.

Table34: use case specification 18

3.initial test case:

TC ID	Title	Preconditions	Test Steps	Expected Result
TC-01	User Registration	User is on the registration page	Enter valid data and submit the form	Account is created and confirmation email is received
TC-02	Registration Validation	User is on the registration page	Enter invalid email, phone, or password and submit	Validation error messages are displayed
TC-03	Login & Logout	User account exists	Log in with valid credentials then log out	User logs in and logs out successfully
TC-04	Password Recovery	Email is registered in the system	Click "Forgot Password" and enter email	Password reset email is sent
TC-05	Change Password	User is logged in	Enter old password and new password	Password is changed successfully
TC-06	View & Edit Profile	User is logged in	Open profile, edit information, and save	Profile information is updated
TC-07	Delete User Account	User is logged in	Confirm account deletion	User account is deleted
TC-08	View Events	Published events exist	Open events page as guest or user	Events list is displayed
TC-09	Search & Filter Events	Published events exist	Search events or apply filters	Correct events are displayed

TC-10	Event Details & Capacity	Event is published	Open event details page	Event details, capacity, and booking status are shown
TC-11	View Vendor Information	Event is linked to a vendor	Open event details page	Vendor information is displayed
TC-12	Book Event	User is logged in, seats available	Complete booking process	Booking is confirmed and capacity is updated
TC-13	Booking Restrictions	Event is full or user not logged in	Attempt to book event	Booking is blocked or user redirected to login
TC-14	Cancel Booking	User has a confirmed booking	Cancel the booking	Booking is canceled and seat is restored
TC-15	Booking Notifications	Booking action occurs	Create or cancel booking	Notification email is received
TC-16	View Bookings	User has bookings	Open bookings page	Upcoming and past bookings are displayed
TC-17	View & Book Packages	Packages are available	Open packages and complete booking	Packages are displayed and booking is successful
TC-18	Event Management	Organizer is logged in	Create, edit, or delete an event	Event is saved or removed correctly
TC-19	Package Management	Organizer is logged in	Create, edit, or delete a package	Package is saved or removed correctly

TC-20	Dashboard & Statistics	Organizer has events	Open dashboard	Dashboard data and statistics are displayed
TC-21	Vendor Registration	Vendor not registered	Complete vendor registration	Vendor account is created
TC-22	System Basic Quality	System is running	Use system on different browsers	System works correctly with acceptable performance

Table 35: initial test cases

4. Requirement Trackability Matrix (RTM) (V.2)

Requirements Id	Title	Use case	System Design	Analysis	Detailed Design	Coding	Test Cases
REQ-FR-01	The system should allow users and service providers to register with basic information	Uc-01					TC-01
REQ-FR-02	The system should validate email, phone, and password	UC-01, UC-02					TC-02
REQ-FR-03	The system should allow users, service providers, and admins to log in and log out	UC-03					TC-03
REQ-FR-04	The system should allow users to reset their password via an email link	UC-03					TC-04
REQ-FR-05	The system should display user profile information	US-03					TC-06
REQ-FR-06	The system should allow users to edit personal details	US-03					TC-06
REQ-FR-07	The system should allow users to change their password	UC-03					TC-05
REQ-FR-08	The system should allow users to delete their account	US-03					TC-07
REQ-FR-09	The system should display all available events on the homepage	US-04					TC-08
REQ-FR-10	The system should allow users to filter events by date, location, price, and event type	UC-04					TC-09
REQ-FR-11	The system should display event cards with key information	UC-04					TC-08, TC-09

REQ-FR-12	The system should allow users to search events by title or keywords	UC-04					TC-09, TC-12
REQ-FR-13	The system should display available seats for each event	UC-04					TC-10
REQ-FR-14	The system should display full event details, including description, images, capacity, and ticket price	UC-04					TC-10
REQ-FR-15	The system should display booking availability status for each event	UC-04					TC-10
REQ-FR-16	The system should display vendor information associated with the event	UC-04					TC-11
REQ-FR-17	The system should prevent booking when event capacity is full	UC-04					TC-13
REQ-FR-18	The system should require users to log in before completing a booking	UC-04					TC-12, TC-13
REQ-FR-19	The system should generate a unique booking reference number upon confirmation	UC-05					TC-12
REQ-FR-20	The system should immediately reduce available capacity after booking	UC-05, UC-07					TC-12
REQ-FR-21	The system should send a booking confirmation email with the reference number	UC-06					TC-15
REQ-FR-22	The system should display hosting packages with setup details	UC-05					TC-17

REQ-FR-23	The system should allow users to book a hosting package for a specific date or location	UC-05						TC-17
REQ-FR-24	The system should display upcoming bookings for the user	UC-04						TC-16
REQ-FR-25	The system should display completed bookings	UC-04						TC-16
REQ-FR-26	The system should allow cancellation of confirmed bookings	UC-04						TC-14
REQ-FR-27	The system should restore event capacity when a booking is canceled	UC-04						TC-14
REQ-FR-28	The system should allow organizers to create new public events	UC-07						TC-18
REQ-FR-29	The system should require mandatory fields for events	UC-07						-
REQ-FR-30	The system should allow organizers to edit existing event details	UC-07						TC-18
REQ-FR-31	The system should allow organizers to delete events	US-07						TC-18
REQ-FR-32	The system should allow organizers to enable or disable booking for an event	UC-07						-
REQ-FR-33	The system should allow organizers to create new hosting packages	UC-07						TC-19
REQ-FR-34	The system should require mandatory fields for hosting packages	UC-05						-
REQ-FR-35	The system should allow defining customizable	UC-05						-

	elements within packages							
REQ-FR-36	The system should allow organizers to edit hosting package details	UC-07						TC-19
REQ-FR-37	The system should allow organizers to delete hosting packages	UC-07						TC-19
REQ-FR-38	The system should allow organizers to filter events by status	UC-07						—
REQ-FR-39	The system should allow organizers to search events within the dashboard	UC-07						—
REQ-FR-40	The system should display event type, city, date, status, bookings, statistics, and revenue in the dashboard	UC-04						TC-20
REQ-NFR-01	The system should be easy to use							TC-20
REQ-NFR-02	The system should load main pages in <3 seconds							TC-20
REQ-NFR-03	The system should protect user data							—
REQ-NFR-04	The system should work on modern browsers							—
REQ-NFR-05	Notifications should be delivered on time							—

Table 36: RTM (V.2)

5. Summary

In this chapter, we thoroughly analyzed the Events and activities planning platform , starting from the initial phase of understanding the general requirements to identifying the main functions the system needs to fulfill. We focused on the key points that matter to the end user,

We clarified the most important use cases where users interact with the system, including login, device management, displaying data on the dashboard, and updating user profiles. Activity diagrams were also created to visualize how processes flow within each part of the system.

Next, we specified the functional requirements for each system module, detailing the operations involved. We also prepared initial test cases to ensure these requirements are properly met.

Finally, we developed a Requirements Traceability Matrix (RTM) to link each requirement with the corresponding analysis, design, and implementation stages. This helps ensure the work covers all aspects and nothing important is overlooked in the upcoming development phases.

Chapter 5: System Design

1. System Initial Design

In order to design the backend architecture of a robust and scalable web application like the Event and activities planning platform , developed using Java Spring Boot, Firebase, and ReactJS, a properly designed architectural pattern needs to be applied. The applied architecture for the backend is the Layered (N-Tier) Architecture with Modularized Microservice Principles.

1. Selected Architecture: Layered Architecture with Modularized Services

Definition: The Layered Architecture, or N-Tier Architecture, structures the backend into tiers where each tier is intended to address a given concern. Shared tiers are:

1. **Presentation Layer (Controller)** – Takes care of processing HTTP requests and forwarding them to service layer actions.
2. **Service Layer (Business Logic)** – Stores intrinsic logic, verifications, and rules that rule the application.
3. **Data Access Layer (Repository)** – Manages database interaction (Firebase in our case here).
4. **Model Layer (Entities/DTOs)** – Made up of data structures that are passed between layers.

Modularization in spirit of microservice architecture is performed in this pattern to maintain the core domains such as EventService, BookingService, UserService, etc., decoupled, even when there is a monolithic Spring Boot application.

Why This Architecture?

1. Separation of Concerns

- Clean decoupling between web processing (controllers), business logic (services), and database operations (repositories) results in better maintainability and testability.
- Each module (e.g., user, Event, Booking) can evolve independently with minimal side effects.

2. Scalability

- Even though the system begins as a monolith, services are designed to be modular and can then scale out into individual microservices if needed (e.g., running BookingService independently).
- Firebase enables easy real-time data updates and scalability for some data types (alerts, user preferences, etc.).

3. Spring Boot Compatibility

- Spring Boot inherently supports layer-based architecture and dependency injection and is therefore an ideal choice.
- Developers like Spring Security, Bean Validation, and integration with Firebase using Spring Cloud GCP libraries are supported easily.

4. Security and Validation

- Input validation (e.g., email address, number of seats, price, event date, booking status) is implemented at two levels:
 - DTO Layer using annotations such as @Valid, @Data@NoArgsConstructor,@AllArgsConstructor
 - Service Layer to validate business logic such as:
 - Ensuring seat availability before booking
 - Verifying event validity (not canceled or expired)
- Authentication and role-based access control are managed using:
 - Spring Security to protect APIs
 - Firebase Authentication to verify user identities
 - Clear permissions are defined, such as:
 - Event Organizer (create/edit/delete events)
 - Regular User (view events/book/pay)
 - Provider (edit your profile,service)

5. Maintainability and Extensibility

6. Firebase Integration

- Firebase Firestore (NoSQL) integrates nicely with Spring Boot using Firebase Admin SDK.
- Services can have custom DAOs or Firebase utility classes to encapsulate Firebase's query and update data logic.
- Firebase's capabilities of real-time can be utilized to refresh dashboards in real time.

2. System Architecture:

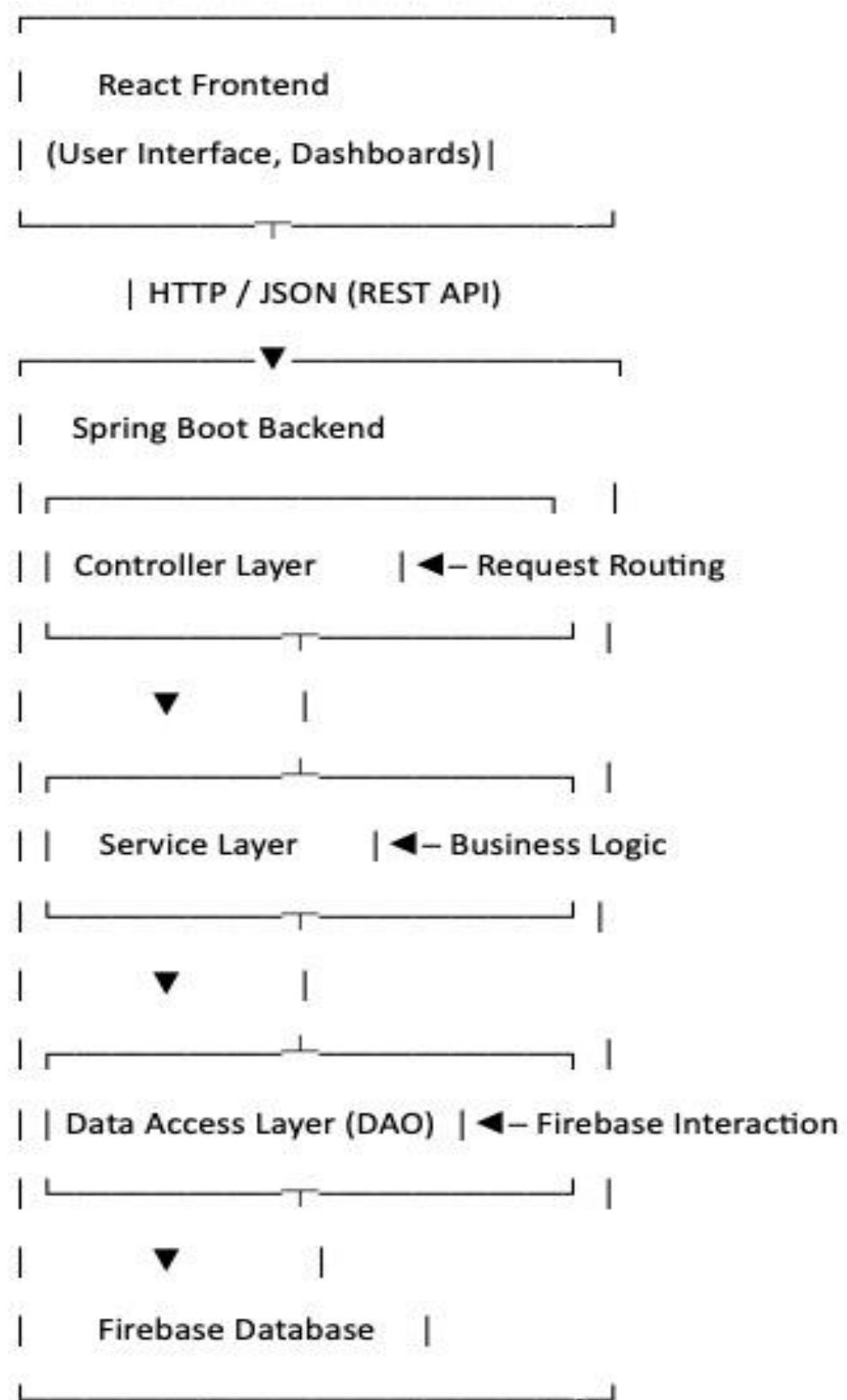


Figure 26:system architecture

3. Class diagram:

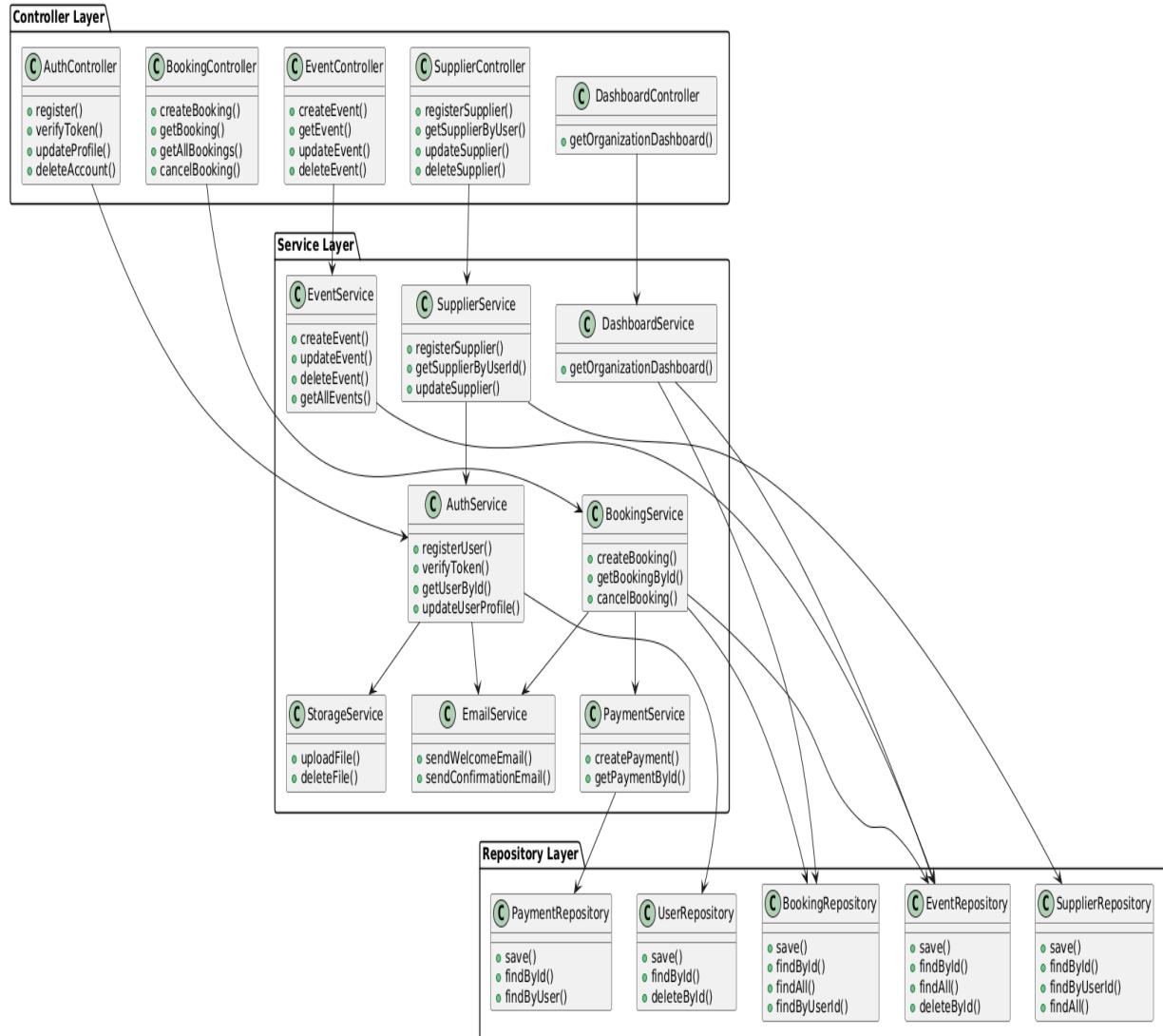


Figure 27:class diagram in high level

2.Detailed Design

1.Introduction:

This section presents the detailed design of the system by providing an in-depth representation of its internal components. It focuses on outlining the software structure and database architecture using diagrams that describe the relationships and interactions among different elements of the system.

2.Purpose :

The purpose of the Detailed Design section is to translate the system's functional requirements into clear, technical structures that guide the development process. This section defines how each part of the system will be implemented and how different components interact with each other. The included diagrams help organize the codebase. This level of design is essential for building a robust, scalable, and maintainable application.

3.functional Requirements:

1. visitor management Functional Requirements:

Requirements ID	Description	Module	Priority	Detailed Specification
REQ-FR-VM-01	System Navigation	landing page	High	system shall display the main landing page and allow the visitor to select between: Browse Events, Browse Providers, or Become a Provider.
RQ-FR-VM-02	Event Discovery & Booking	Event Management	High	The system shall allow users to browse, search, and filter events. Upon clicking 'Book', the system must verify login status and either proceed to booking or prompt a login message.
RQ-FR-VM-03	Provider Exploration	Provider Directory	Medium	The system shall display a list of providers with search/filter capabilities and provide a link to the provider's external website upon clicking 'Contact Provider'.
RQ-FR-VM-04	Provider Onboarding	Registration	High	The system shall provide a registration form for new providers, validate the submitted data, and upon success, create an account and send a confirmation to the user.
RQ-FR-VM-05	Error & Validation Handling	System Logic	High	The system shall handle validation logic for both user authentication (during booking) and data integrity (during registration), displaying appropriate error or success messages.

Table 37: VM Functional Requirements

3.3.1.4 Sequence Diagram for visitor management:

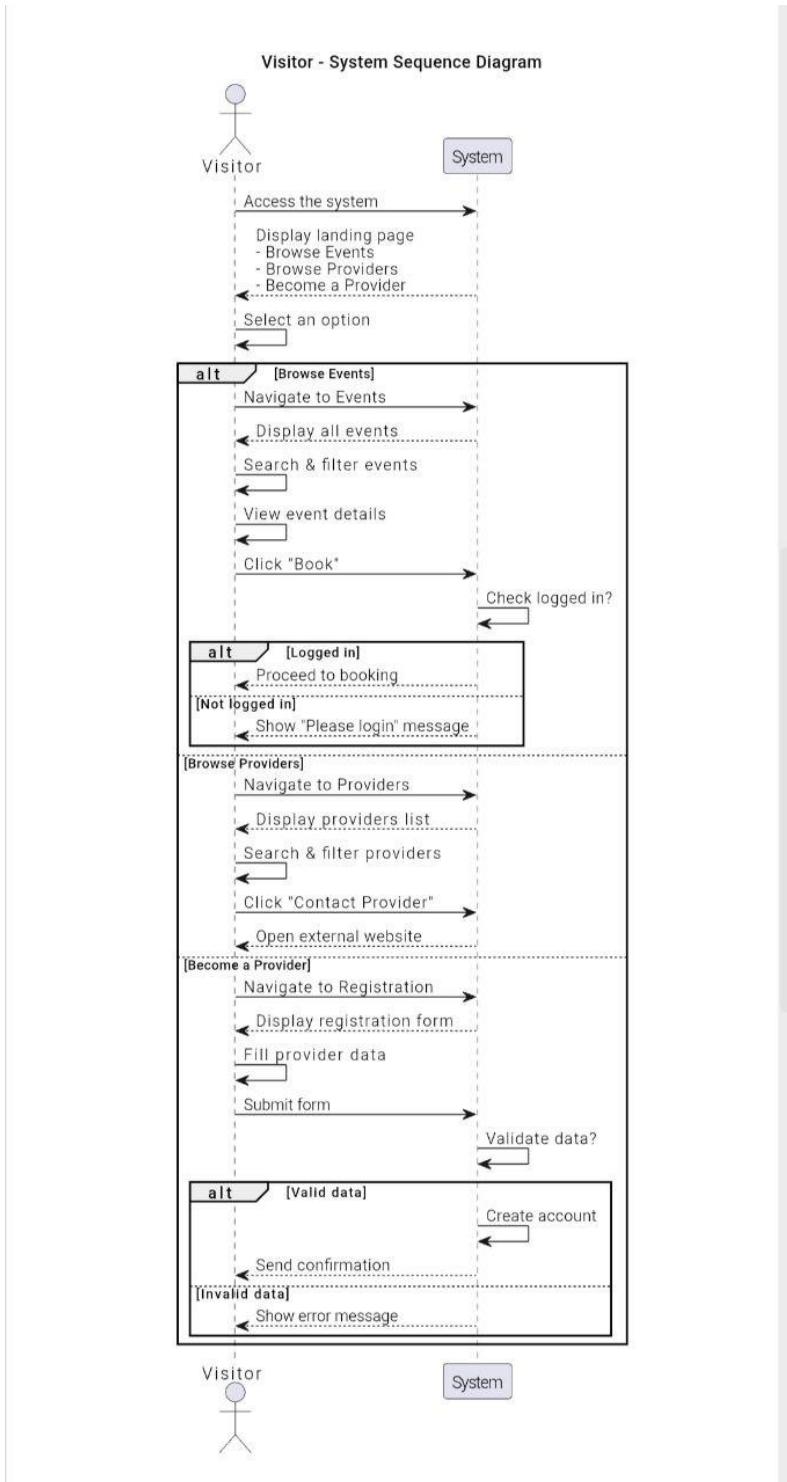


Figure 28:sequence diagram for visitor management

2. registration account management Functional Requirements:

Requirements ID	Description	Module	Priority	Detailed Specification
REQ-FR-RAM-01	Registration Page	Registration	High	The system shall display the "Create Account" form when the user navigates to the registration page.
REQ-FR-RAM-02	Data Entry	Registration	High	The system shall allow users to input: Name, Email, Password, Phone Number, Role (User/Provider), and Password Confirmation.
REQ-FR-RAM-03	Input Validation	Validation	High	The system shall validate all mandatory fields and ensure password matches the confirmation before processing.
REQ-FR-RAM-04	Account Creation	Account Mgmt	High	Upon successful validation, the system shall create a new user account in the database.
REQ-FR-RAM-05	Post-Registration Redirect	Navigation	Medium	The system shall redirect the user to the application home page after a successful registration.
REQ-FR-RAM-06	Error Notification	Error Handling	High	The system shall return and display an error message if the input data is invalid or registration fails.

Table 38: RAM Functional Requirements

3.3.2.4 Sequence Diagram for registration account :

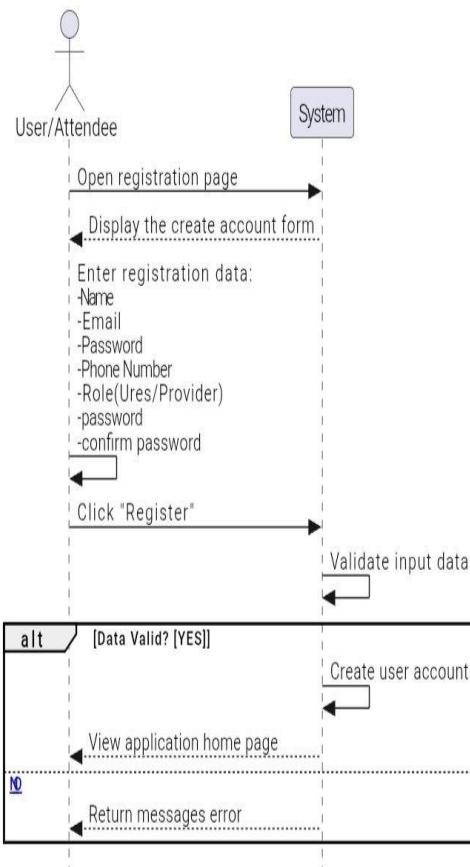


Figure 29: sequence diagram for registration account

3. Login management Functional Requirements:

Requirements ID	Description	Module	Priority	Detailed Specification
REQ-FR-LM-01	Access Sign-In Form	Authentication	High	The system shall display the sign-in form upon user request.
REQ-FR-LM-02	Data Entry	Authentication	High	The system shall allow users to input their Email and Password.
REQ-FR-LM-03	Input Validation	Validation	High	The system must validate the entered credentials against the database.
REQ-FR-LM-04	Login Success	Authentication	High	If data is valid, the system shall redirect the user to the application home page.
REQ-FR-LM-05	Login Failure	Error Handling	Medium	If data is invalid, the system shall return an error message and allow re-entry (Loop).

Table 39: LM Functional Requirements

3.3.3.4 Sequence diagram for Login:

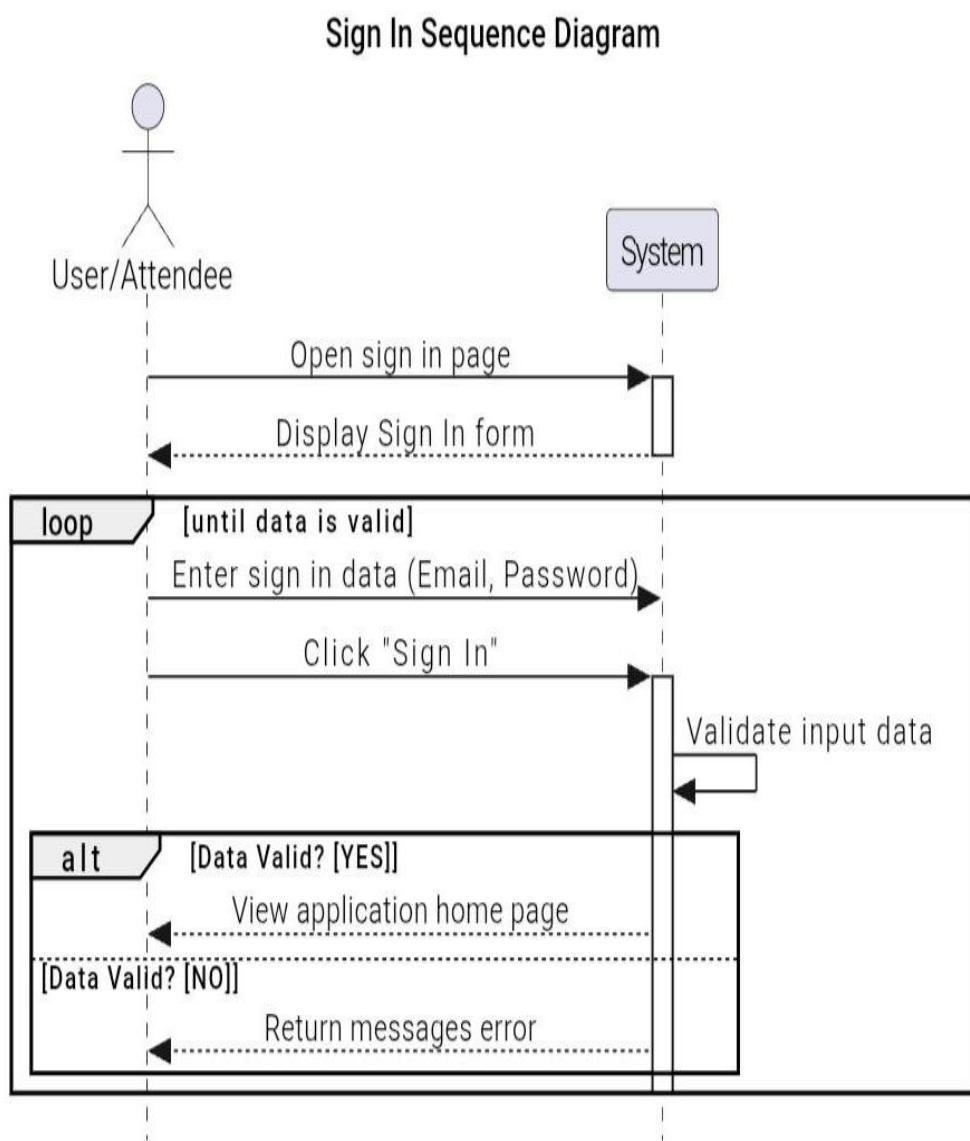


Figure 30: sequence diagram for Login

4. forget password Management Functional Requirement:

Requirements ID	Description	Module	Priority	Detailed Specification
REQ-FR-FPM-01	Forgot Password Initiation	User Auth	High	The system shall provide a "Forgot password" option on the login page to initiate the recovery process.
REQ-FR-FPM-02	Email Input Form	User Auth	High	The system shall display a dedicated form for users to enter their registered email address.
REQ-FR-FPM-04	Reset Instructions	Notification	Medium	Upon clicking "Send Reset instructions", the system shall process the request and trigger a reset email.
REQ-FR-FPM-05	Confirmation Message	User UI	Low	The system shall display a "check your email" notification to the user after successful submission.
REQ-FR-FPM-06	Return to Login	Navigation	Medium	The system shall provide a "Back to login" navigation option to return the user to the main authentication page.

Table 40: FPM Functional Requirements

3.3.4.4 Sequence diagram for forget password Management:

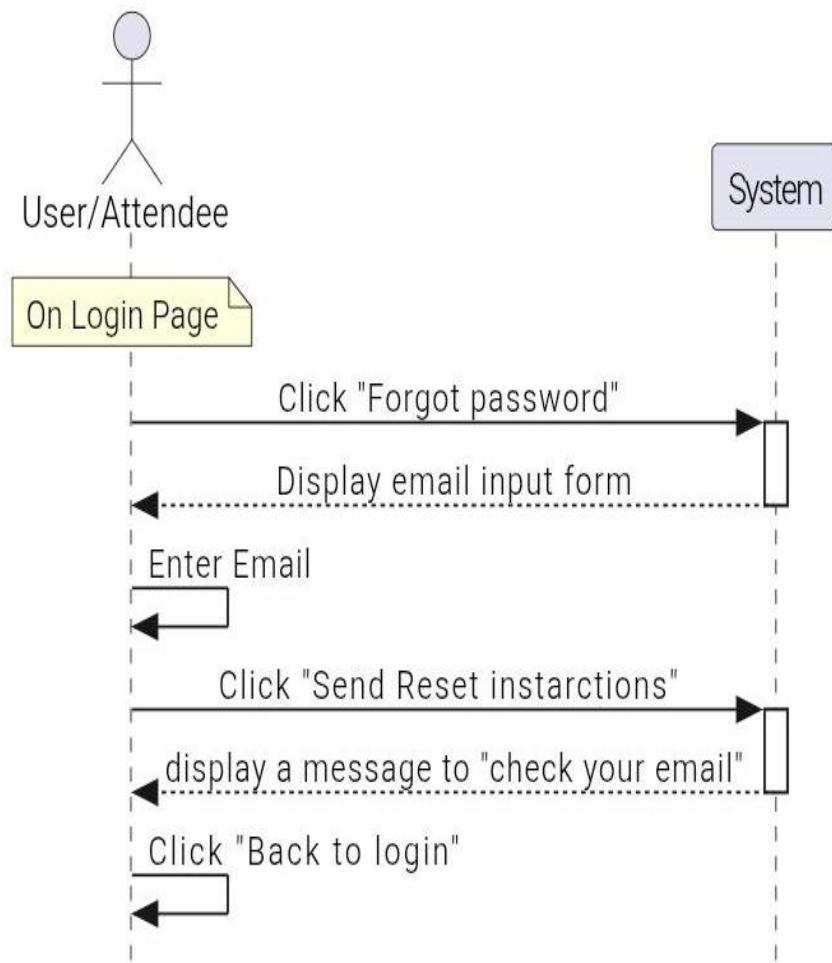


Figure 31: sequence diagram for forget password Management

5. user profile management Functional Requirements:

Requirements ID	Description	Module	Priority	Detailed Specification
REQ-FR-UPM-01	User Login	Authentication	High	The system shall allow users to log in and, upon successful authentication, display the dashboard.
REQ-FR-UPM-02	Access Profile	Navigation	Medium	The system shall provide navigation to the Profile section and display user information and management options.
REQ-FR-UPM-03	Change Password	Profile Management	High	The system shall allow users to change their password by verifying the current password and ensuring the new passwords match.
REQ-FR-UPM-04	Edit Profile	Profile Management	Medium	The system shall allow users to update their name, email, phone, and profile picture after validating the provided data.
REQ-FR-UPM-05	Delete Account	Profile Management	Low	The system shall allow users to permanently delete their account after a confirmation step and password verification.
REQ-FR-UPM-06	Error Handling	Profile Management	Medium	The system must provide specific feedback/error messages for "Incorrect password", "Passwords don't match", and "Invalid data".

Table 41: UPM Functional Requirements

3.3.5.4 Sequence Diagram for user profile management:

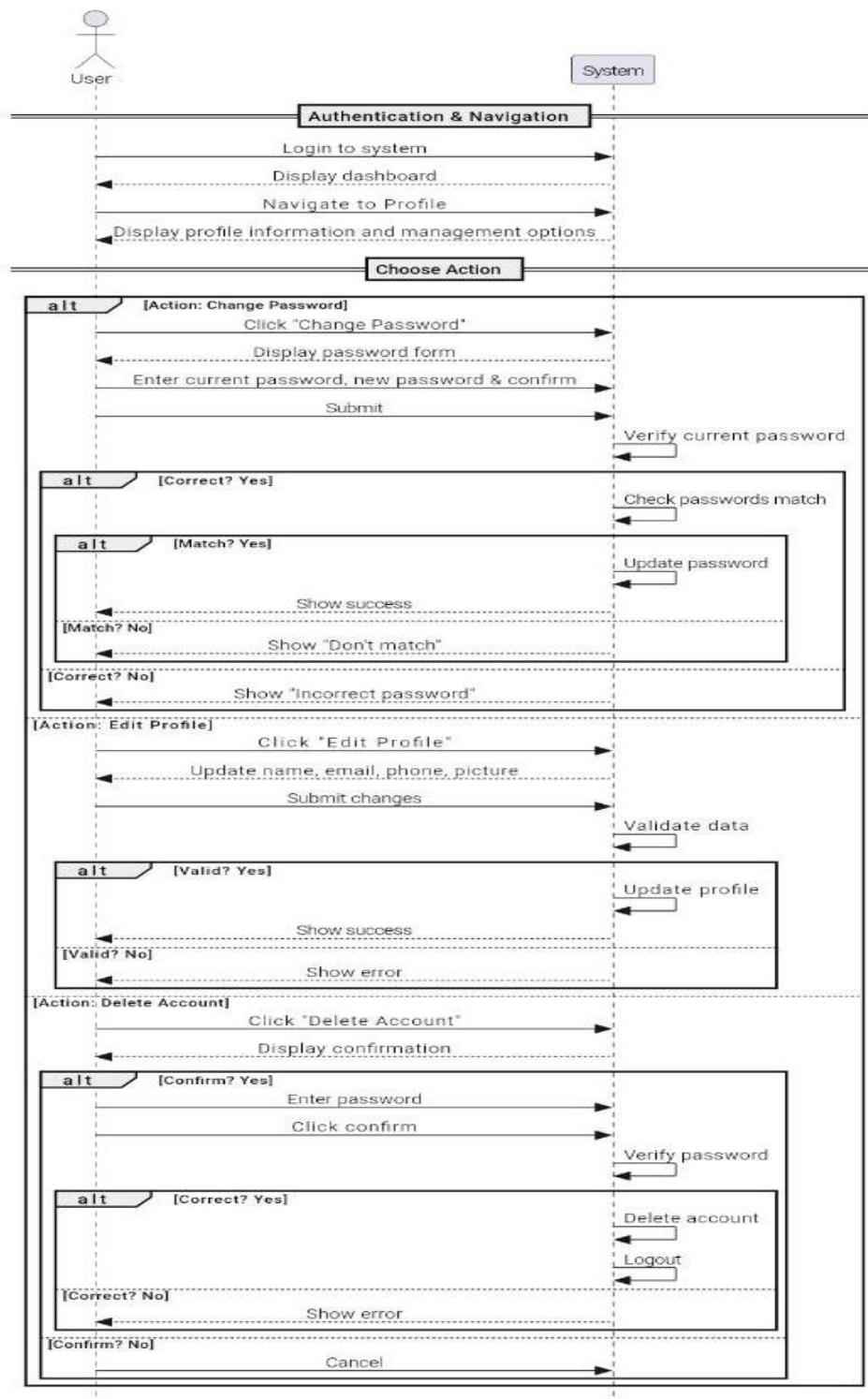


Figure 32:sequence diagram for user profile management

6. Event discovery & booking Functional Requirements:

Requirements ID	Description	Module	Priority	Detailed Specification
REQ-FR-EDB-01	Search & Browse Events	Event Discovery	High	The system shall allow users to browse event listings and apply filters (like status) to view specific results.
REQ-FR- EDB -02	View Event Details	Event Discovery	High	The system shall display comprehensive details for a selected event, including description, images, and capacity.
REQ-FR- EDB -03	Capacity Validation	Booking	High	The system must check seat availability; if the event is full, it must prevent booking and notify the user.
REQ-FR- EDB -04	User Authentication Check	Booking	High	The system shall verify if the user is logged in before proceeding with the booking; if not, a login prompt is displayed.
REQ-FR- EDB -05	Process Booking	Booking	Critical	Upon successful validation, the system shall confirm the booking and generate a unique reference number.
REQ-FR- EDB -06	Inventory Management	Booking	High	The system shall automatically reduce the available capacity of the event once a booking is confirmed.
REQ-FR- EDB -07	Booking Confirmation	Booking	Medium	The system shall send an automated confirmation email to the attendee and display a success message on the screen.
REQ-FR- EDB -08	View My Bookings	Post-Booking	Medium	The system shall allow users to view a list of their upcoming and past event bookings.
REQ-FR- EDB -09	Cancel Booking	Post-Booking	Medium	The system shall allow users to request a cancellation, update the booking status, and restore the event capacity.
REQ-FR- EDB -10	Cancellation Notify	Post-Booking	Low	The system shall send a cancellation confirmation to the user and confirm completion on the interface.

Table 42: EDB Functional Requirements

3.3.6.4 Sequence diagram for event discovery & booking :

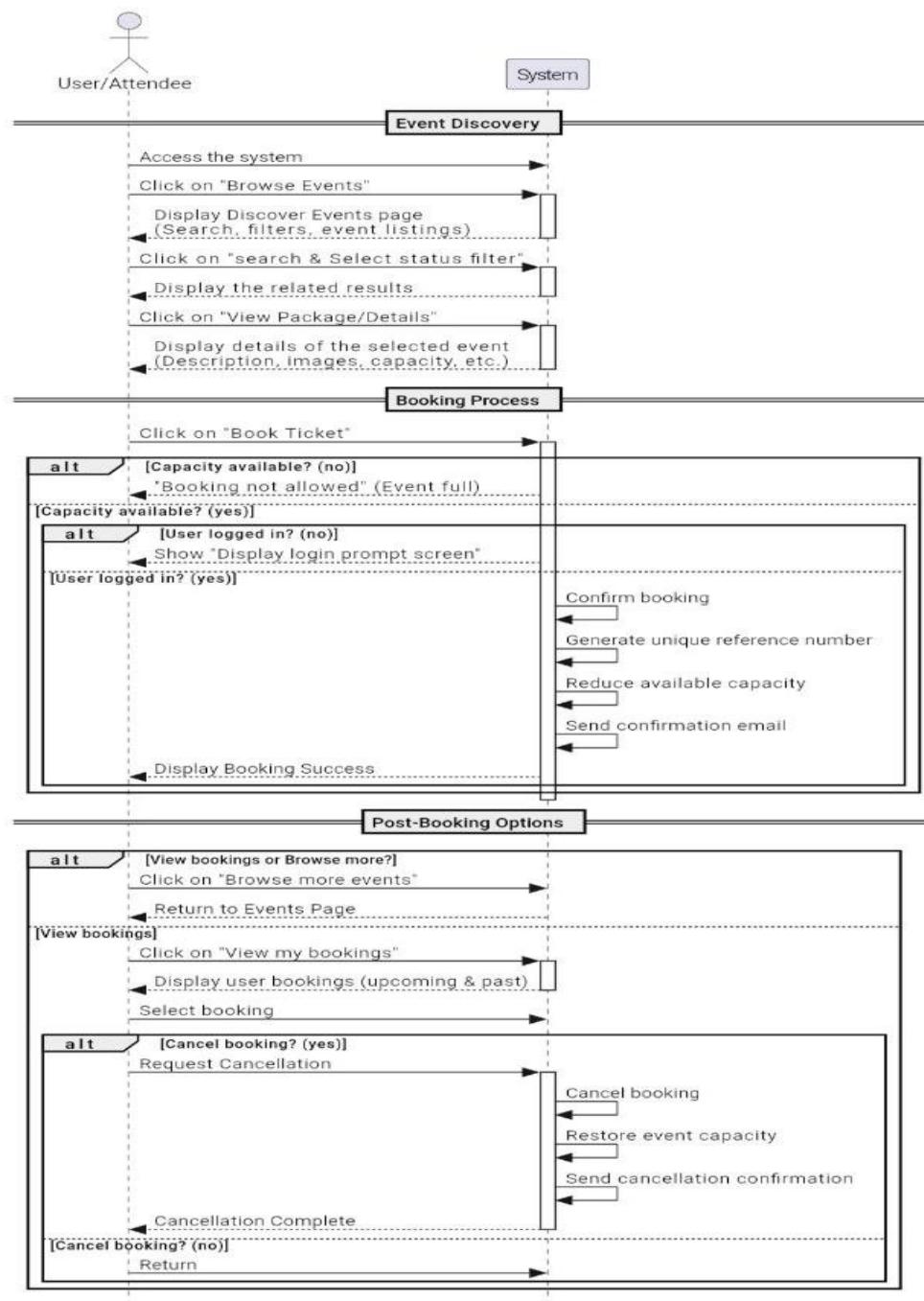


Figure 33:sequence diagram for event discovery & booking

7. Organizer Dashboard and Bookings Management- Functional Requirement:

Requirements ID	Description	Module	Priority	Detailed Specification
REQ-FR-ODBM-01	User Login	Authentication	High	The system shall allow the Organizer to log in and access their specific dashboard.
REQ-FR-ODBM-02	Dashboard Visualization	Dashboard	High	The system must display KPIs (Total Bookings, Total Revenue), performance charts, and a table of recent bookings.
REQ-FR-ODBM-03	Navigation & Management	Bookings	Medium	The system shall provide a dedicated section to manage bookings, displaying total revenue and a comprehensive bookings list.
REQ-FR-ODBM-04	Search & Filtering	Bookings	Medium	The system shall allow the user to filter and search bookings based on specific status criteria and display related results.

Table 43: ODBM Functional Requirements

3.3.7.4 Sequence diagram for Organizer Dashboard and Bookings Management:

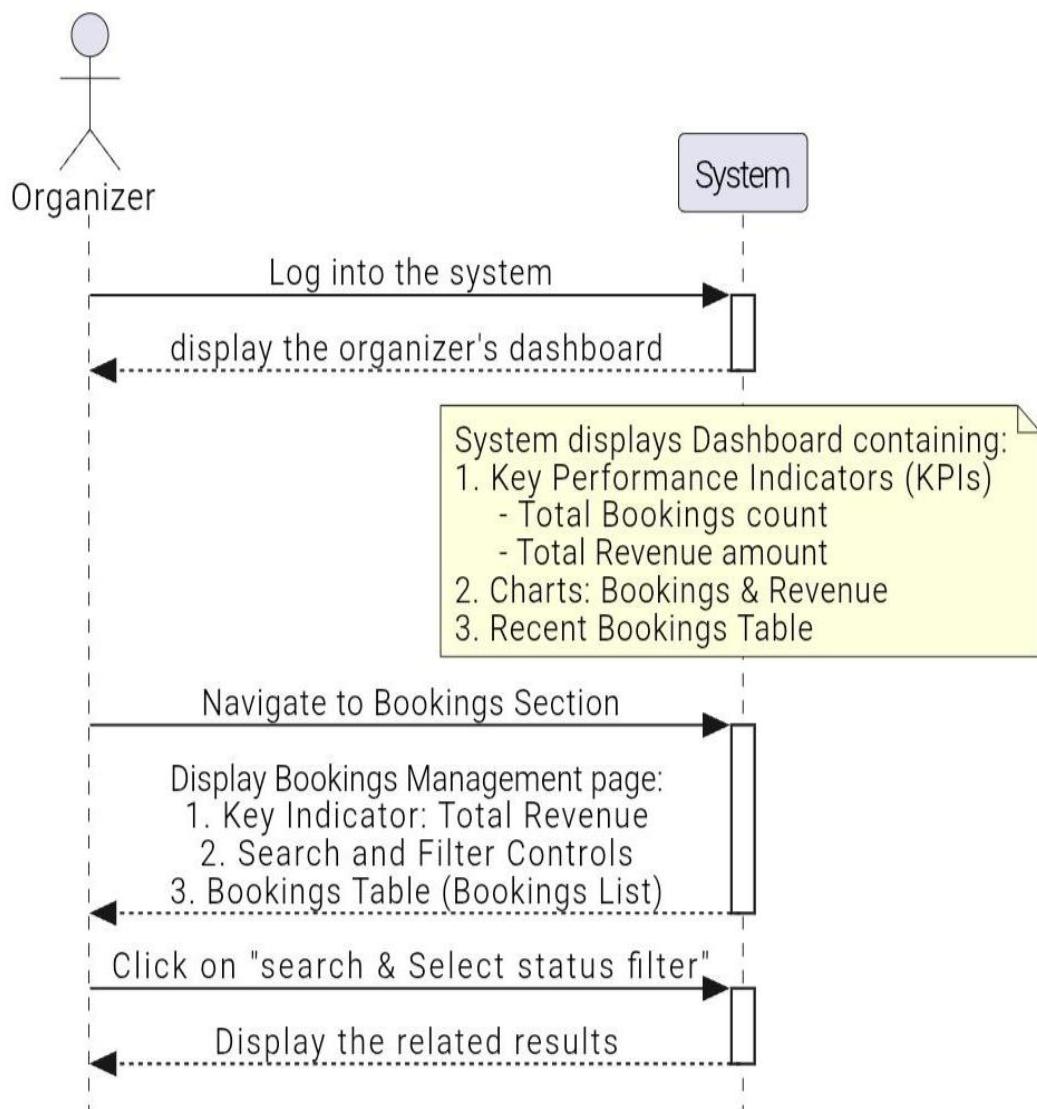


Figure 34:sequence diagram for Organizer Dashboard and Bookings Management

8. Disable/Enable Bookings Functional Requirement:

Requirements ID	Description	Module	Priority	Detailed Specification
REQ-FR-DEB-01	User Login	Authentication	High	The system shall allow the Organizer to log in and access their personal dashboard.
REQ-FR-DEB-02	View Events	Event Management	High	The system shall provide a navigation path to the Events Section and display a list of all existing events.
REQ-FR-DEB-03	Disable Bookings	Booking Control	High	The system shall allow the Organizer to disable bookings for a specific event.
REQ-FR-DEB-04	Update Status	Status Management	Medium	The system shall automatically update and display the booking status as "Closed" upon disabling.
REQ-FR-DEB-05	Success Feedback	Notification	Medium	The system shall send a success notification to the Organizer confirming the change in booking status.

Table 44: DEB Functional Requirements

3.3.8.4 Sequence diagram for Disable/Enable Bookings:

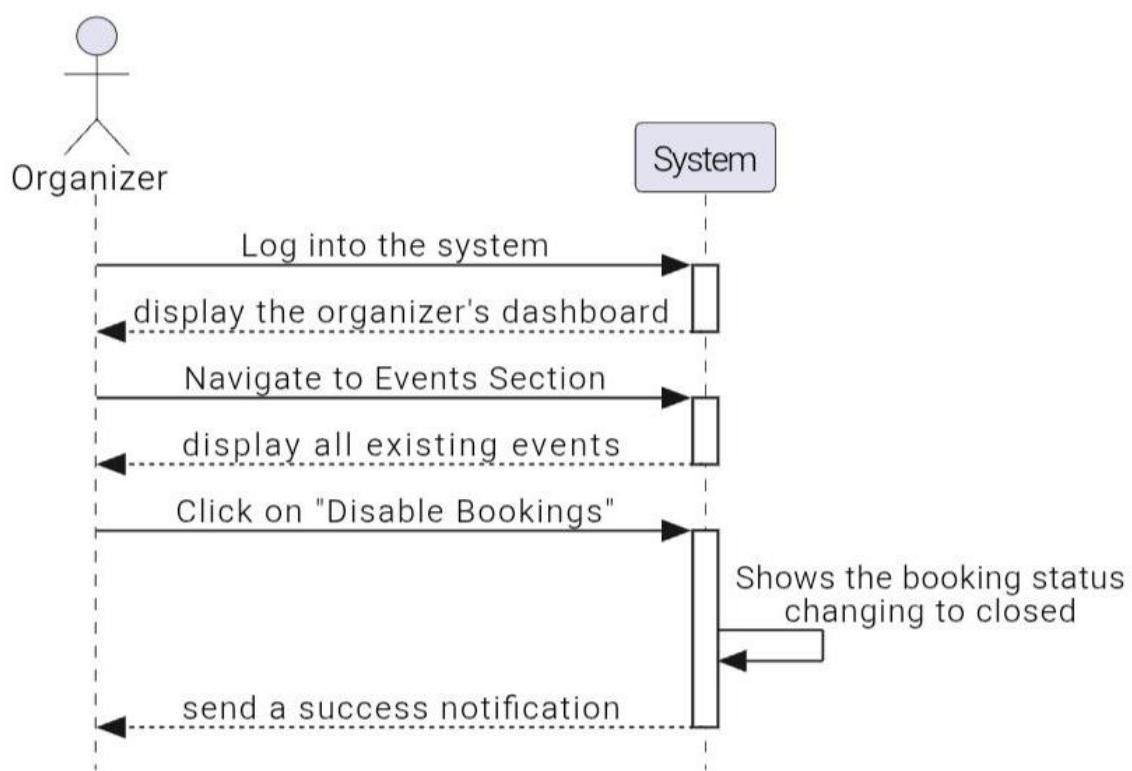


Figure 35:sequence diagram for Disable/Enable Bookings

9. Create New Public Event or Host Package Functional Requirements:

Requirements ID	Description	Module	Priority	Detailed Specification
REQ-FR-CNPEHP-01	User Login	Authentication	High	The system shall allow the Organizer to log in and access the dashboard.
REQ-FR-CNPEHP-02	View Events	Event Management	Medium	The system shall display a list of all existing events upon navigating to the Events Section.
REQ-FR-CNPEHP-03	Form Access	Event Management	High	The system shall provide a "Create Event" form to the Organizer.
REQ-FR-CNPEHP-04	Data Entry	Event Management	High	The system shall allow inputting Title, Type, Location, DateTime, Price, Capacity, Status, and Booking.
REQ-FR-CNPEHP-05	Form Validation	Validation	High	The system shall validate that all required fields are filled before processing the creation.
REQ-FR-CNPEHP-06	Error Handling	Notification	Medium	The system shall display a warning/error notification if any required fields are missing.
REQ-FR-CNPEHP-07	Event Storage	Database / Storage	High	The system shall add the new event to the events table upon successful validation.
REQ-FR-CNPEHP-08	Success Confirmation	Notification	Medium	The system shall send a success notification to the Organizer after the event is successfully created.

Table 45: CNPEHP Functional Requirements

3.3.9.4 Sequence diagram for Create New Public Event or Host Package:

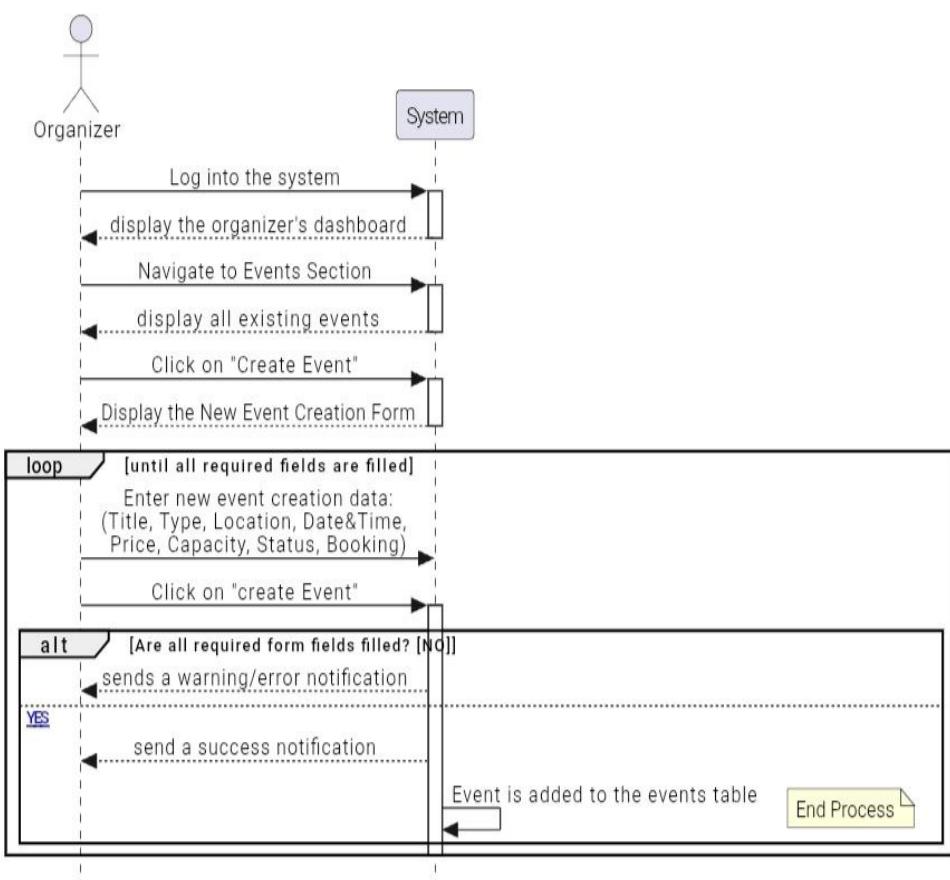


Figure 36: sequence diagram for Create New Public Event or Host Package

10. Edit Existing Event or Package Functional Requirements:

Requirements ID	Description	Module	Priority	Detailed Specification
REQ-FR-EFP-01	User Login	Authentication	High	The system shall allow the Organizer to log in and access the dashboard.
REQ-FR-EFP-02	View Events	Event Management	High	The system shall display a list of all existing events created by the organizer.
REQ-FR-EFP-03	Access Edit Form	Event Management	Medium	The system shall provide an "Edit" option and display a form populated with current event data.
REQ-FR-EFP-04	Update Event	Event Management	High	The system shall allow editing of: Title, Type, Location, Date & Time, Price, Capacity, Status, and Booking.
REQ-FR-EFP-05	Save Changes	Event Management	High	Upon clicking "Update", the system shall save the new data to the database and show a success notification.
REQ-FR-EFP-06	Cancel Operation	Event Management	Low	The system shall allow the user to cancel the edit process and return to the events list without saving.

Table 46: EEP Functional Requirements

3.3.10.4 Sequence diagram for Edit Existing Event or Package:

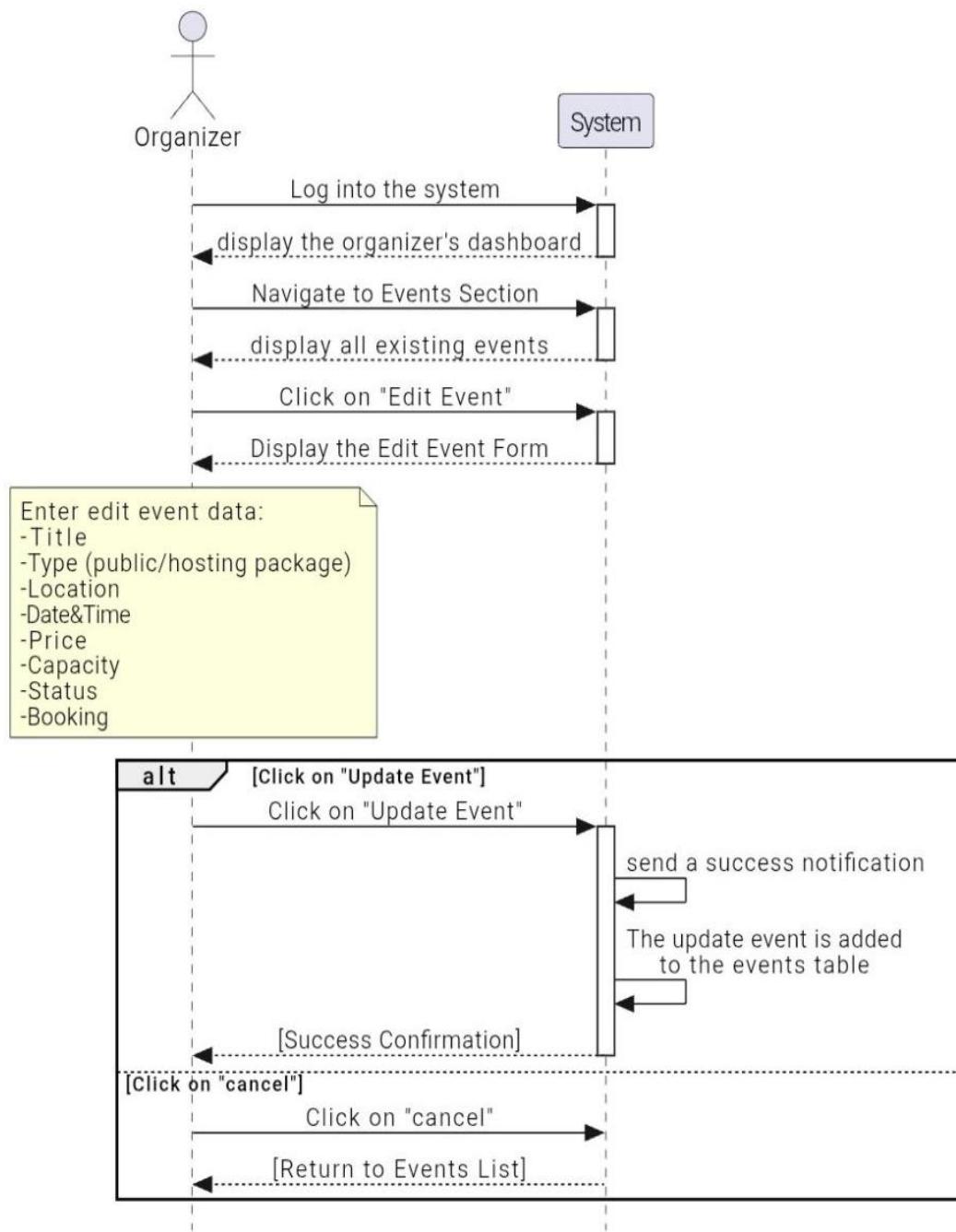


Figure 37: sequence diagram for Edit Existing Event or Package

11. Delete Event or Package Functional Requirements:

Requirements ID	Description	Module	Priority	Detailed Specification
REQ-FR-DEP-01	User Login	Authentication	High	The system shall allow the Organizer to log in and access the dashboard.
REQ-FR-DEP-02	View Events	Event Management	High	The system shall display a list of all existing events in the Events Section.
REQ-FR-DEP-03	Delete Event Initiation	Event Management	Medium	The system shall provide a "Delete" option and prompt for confirmation before removal.
REQ-FR-DEP-04	Confirm Deletion	Event Management	High	Upon clicking "Yes", the system shall permanently remove the event from the database.
REQ-FR-DEP-05	Success Feedback	Notification	Medium	The system shall send a success notification to the Organizer after a successful deletion.
REQ-FR-DEP-06	Cancel Deletion	Event Management	Low	If the user clicks "Cancel", the system shall stop the process and return to the previous state.

Table 47: DEP Functional Requirements

3.3.11.4 Sequence diagram for Delete Event or Package:

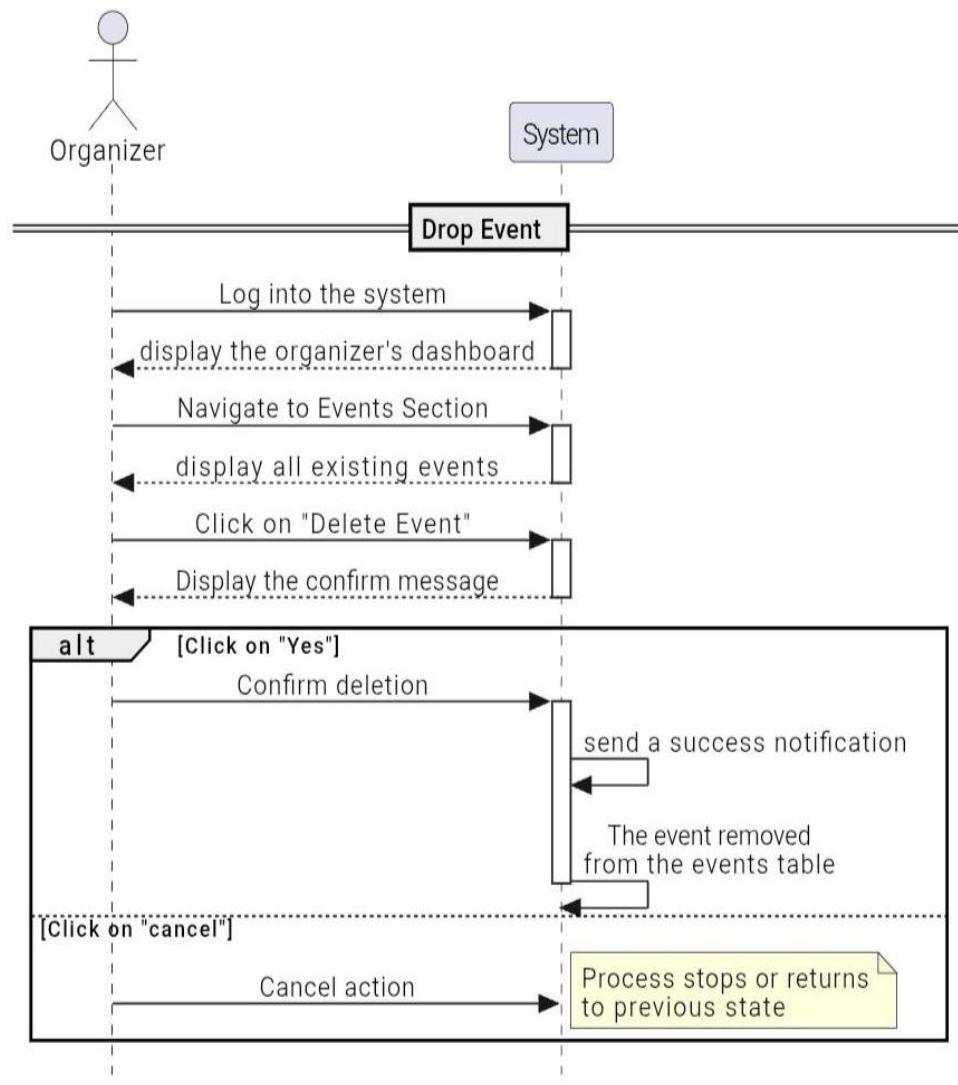


Figure 38: sequence diagram for Delete Event or Package

12. View My Profile Functional Requirements:

Requirements ID	Description	Module	Priority	Detailed Specification
REQ-FR-VMP-01	System Login	Authentication	High	The system shall allow the Provider to log in to access their account.
REQ-FR-VMP-02	Navigate to Profile	Profile Management	Medium	The system shall provide a "My Profile" navigation option for the Provider.
REQ-FR-VMP-03	Retrieve Profile Data	Profile Management	High	The system must fetch provider data (Business Name, Description, Contacts, Services, Location, Category) from the database.
REQ-FR-VMP-04	Display Profile View	Profile Management	High	The system shall display the retrieved profile information to the Provider for review.
REQ-FR-VMP-05	Edit Profile Trigger	Profile Management	Medium	The system shall enable the "Edit Profile" action if the user chooses to modify their information.
REQ-FR-VMP-06	Return to Dashboard	Dashboard	Low	The system shall allow the user to exit the profile view and return to the main dashboard.

Table 48: VMP Functional Requirements

3.3.12.4 Sequence diagram for view my profile:

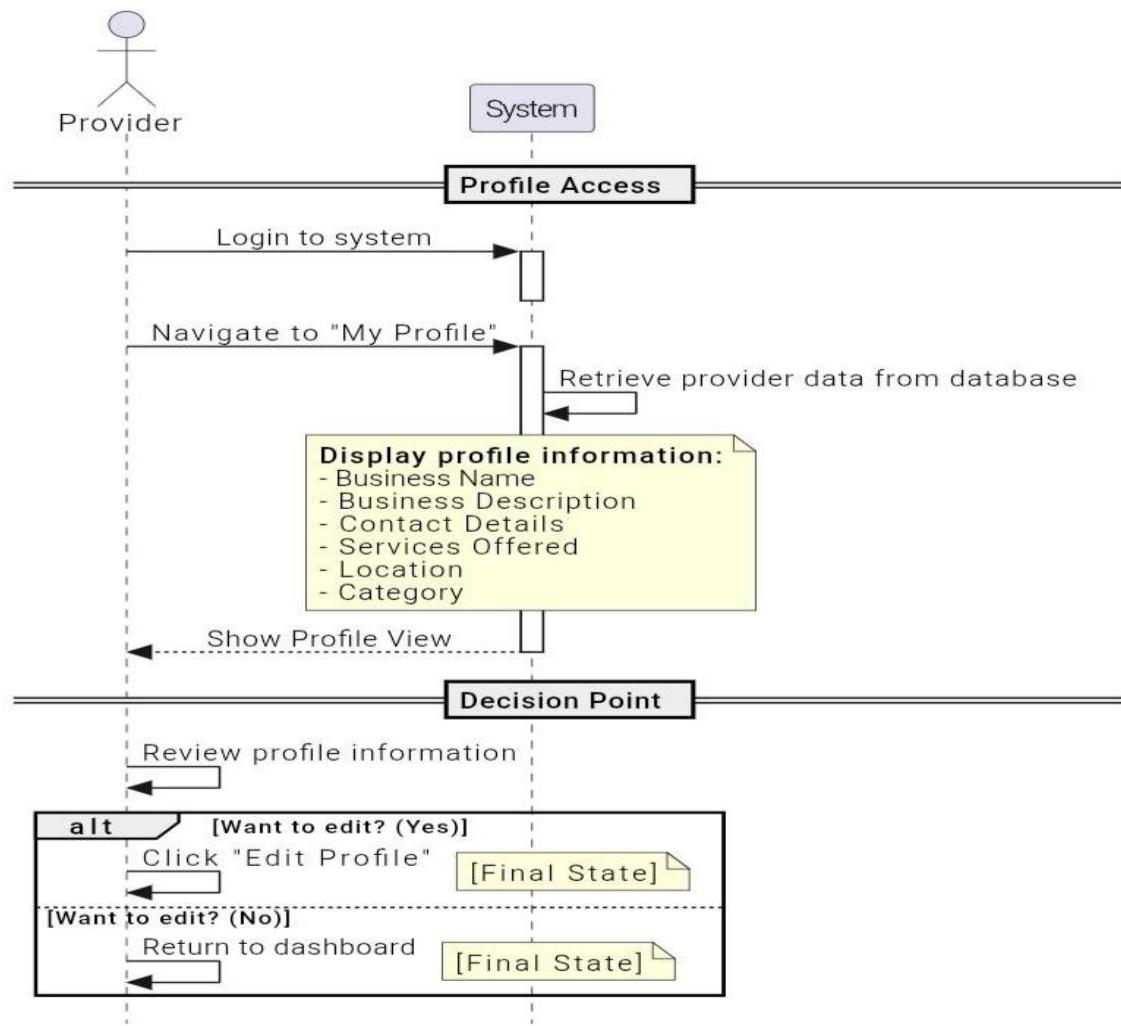


Figure 39: sequence diagram for view my profile

13. Edit Personal Information Functional Requirements:

Requirements ID	Description	Module	Priority	Detailed Specification
REQ-FR-EPI-01	User Login	Authentication	High	The system shall allow the Provider to authenticate and access the system.
REQ-FR- EPI -02	Profile Navigation	Profile Management	Medium	The system shall enable the user to navigate to the "My Profile" section.
REQ-FR- EPI -03	Edit Profile Mode	Profile Management	High	The system shall display existing profile data (Name, Email, Phone, Picture) when "Edit Profile" is clicked.
REQ-FR- EPI -04	Data Entry	Profile Management	High	The system shall allow the user to input and modify full name, email, phone number, and profile picture.
REQ-FR- EPI -05	Logic Input Validation	Validation Logic	High	The system must validate all entered data for correctness and format after clicking "Save".
REQ-FR- EPI -06	Database Update	Data Persistence	High	Upon successful validation, the system shall permanently update the provider's information in the database.
REQ-FR- EPI -07	Success Feedback	Notifications	Medium	The system shall display a success message to the provider after a successful database update.
REQ-FR- EPI -08	Error Notification	Error Handling	High	If data is invalid, the system shall return error messages and prompt the user to correct the information.

Table 49: EPI Functional Requirements

3.3.13.4 Sequence diagram for Edit Personal Information:

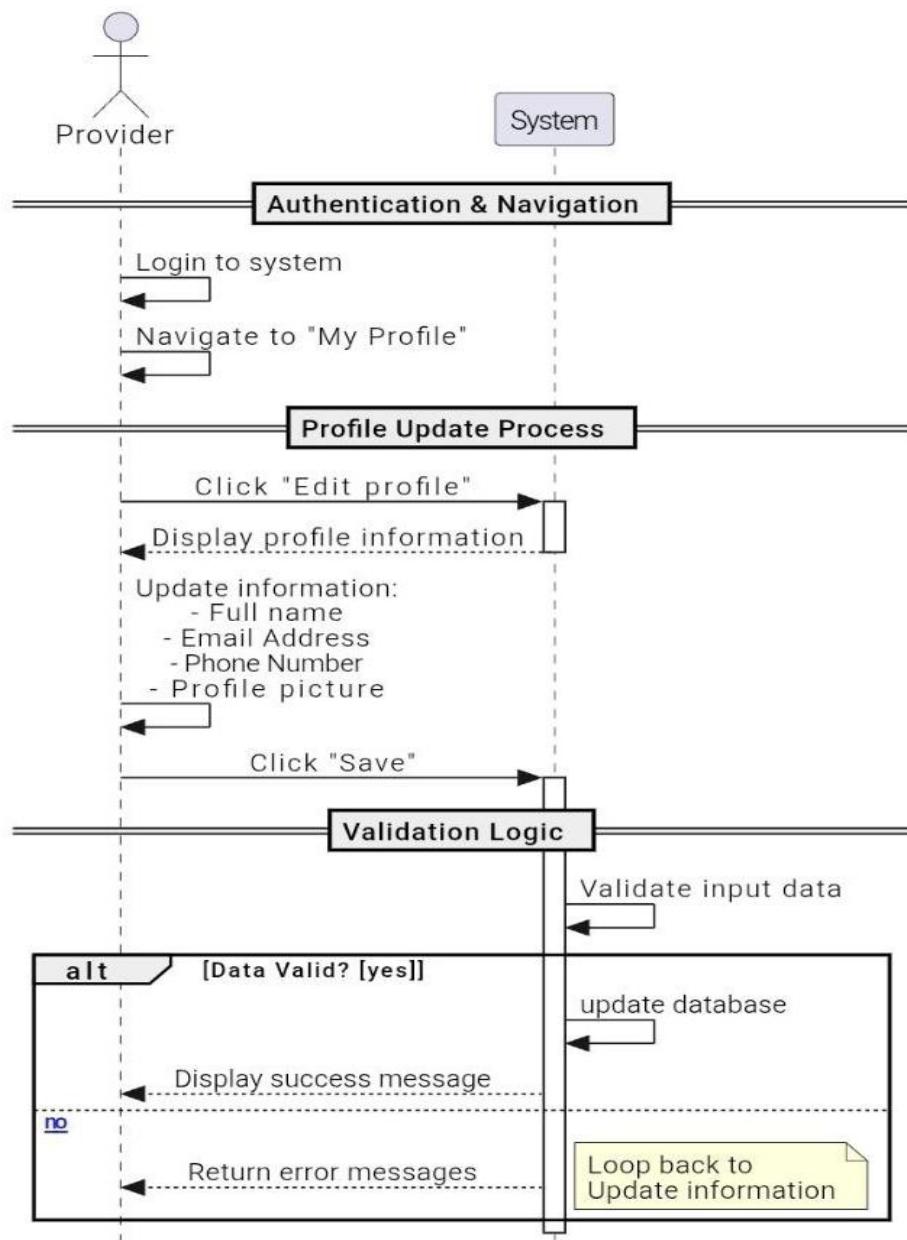


Figure 40: sequence diagram for Edit Personal Information

14. Edit Business information Functional Requirements:

Requirements ID	Description	Module	Priority	Detailed Specification
REQ-FR-EBI-01	User Login	Authentication	High	The system shall allow the Provider to log in to access the profile management features.
REQ-FR- EBI-02	Access Business Info	Profile Management	High	The system shall allow the Provider to navigate to "My Profile" and initiate editing of business information.
REQ-FR- EBI-03	Display Edit Form	Profile Management	Medium	The system must retrieve and display a form populated with current data (Name, Description, Category, Location, URL).
REQ-FR- EBI-04	Data Validation	Profile Management	High	The system shall validate all input fields upon the user clicking "Save Changes" to ensure data integrity.
REQ-FR- EBI-05	View Updates	Profile Management	High	The system shall allow the Provider to view the updated profile immediately after a successful save.
REQ-FR- EBI-06	Update Record	Database	High	Upon successful validation, the system shall update the provider's record in the database and display a success message.
REQ-FR- EBI--07	Validation Errors	Error Handling	Medium	If data is invalid, the system shall display specific validation errors and allow the user to correct them or cancel.

Table 50 : EBI Functional Requirements

3.3.14.4 Sequence diagram for Edit Business information:

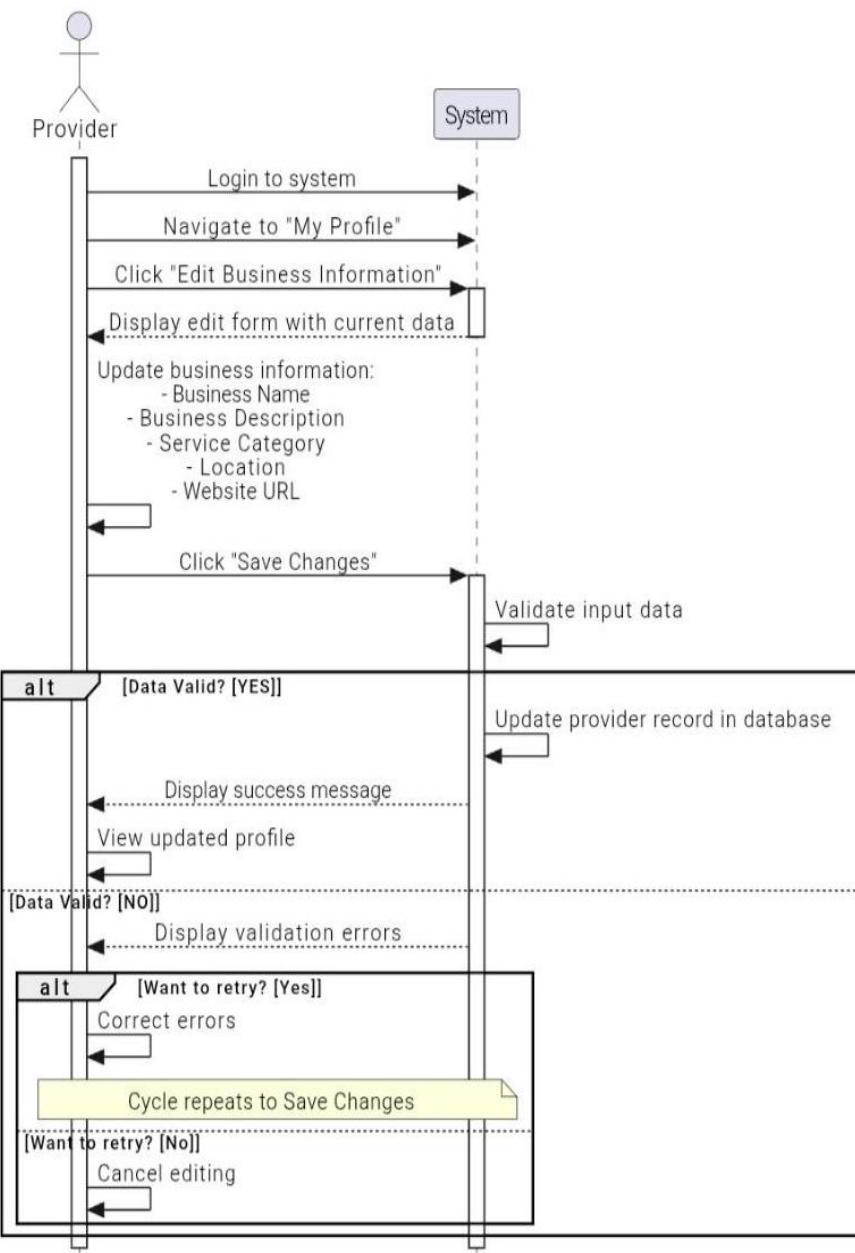


Figure 41: sequence diagram for Edit Business Information

4. System Class Diagram:

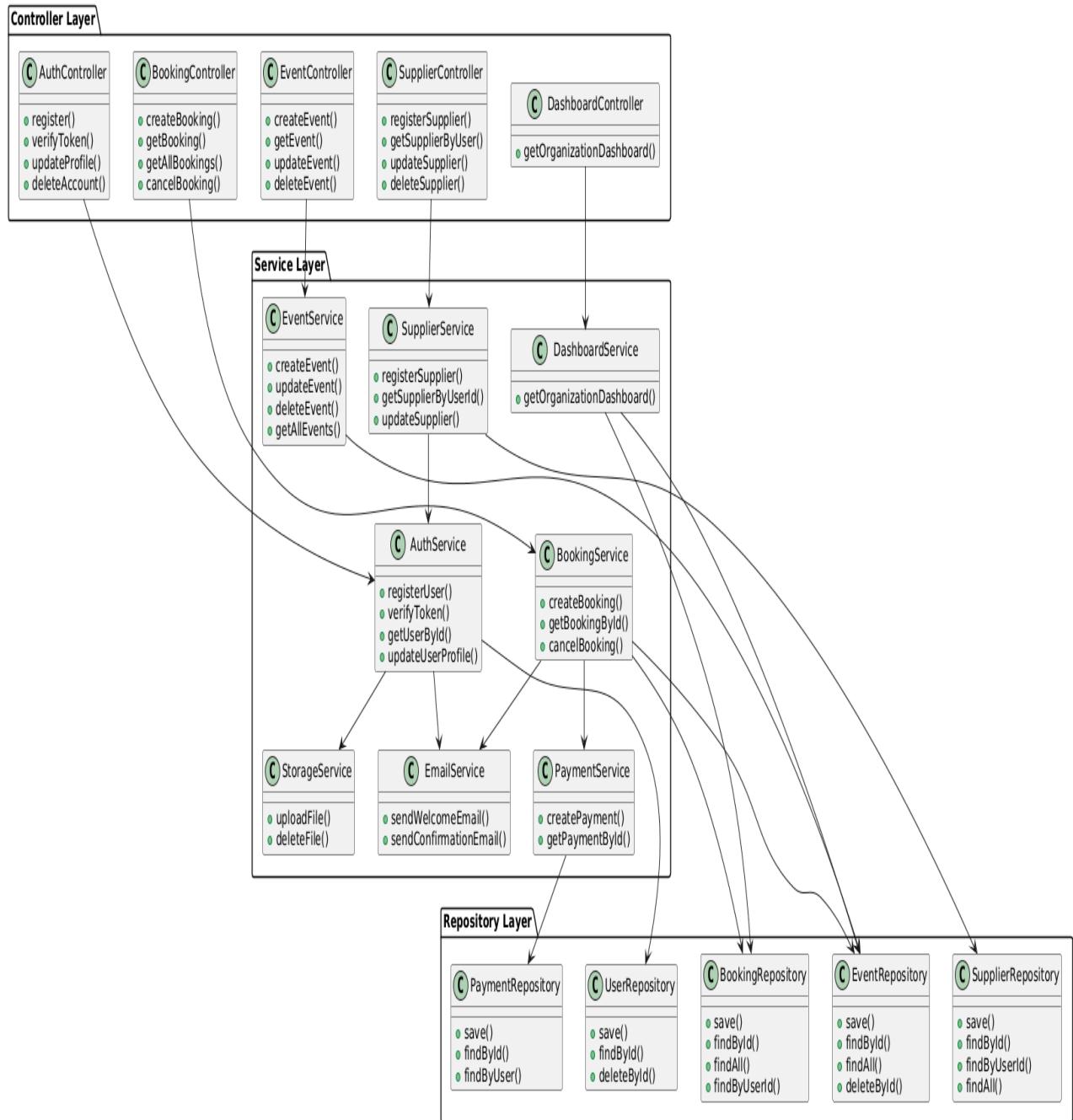


Figure 42 : class diagram

5.Data Model Diagram:

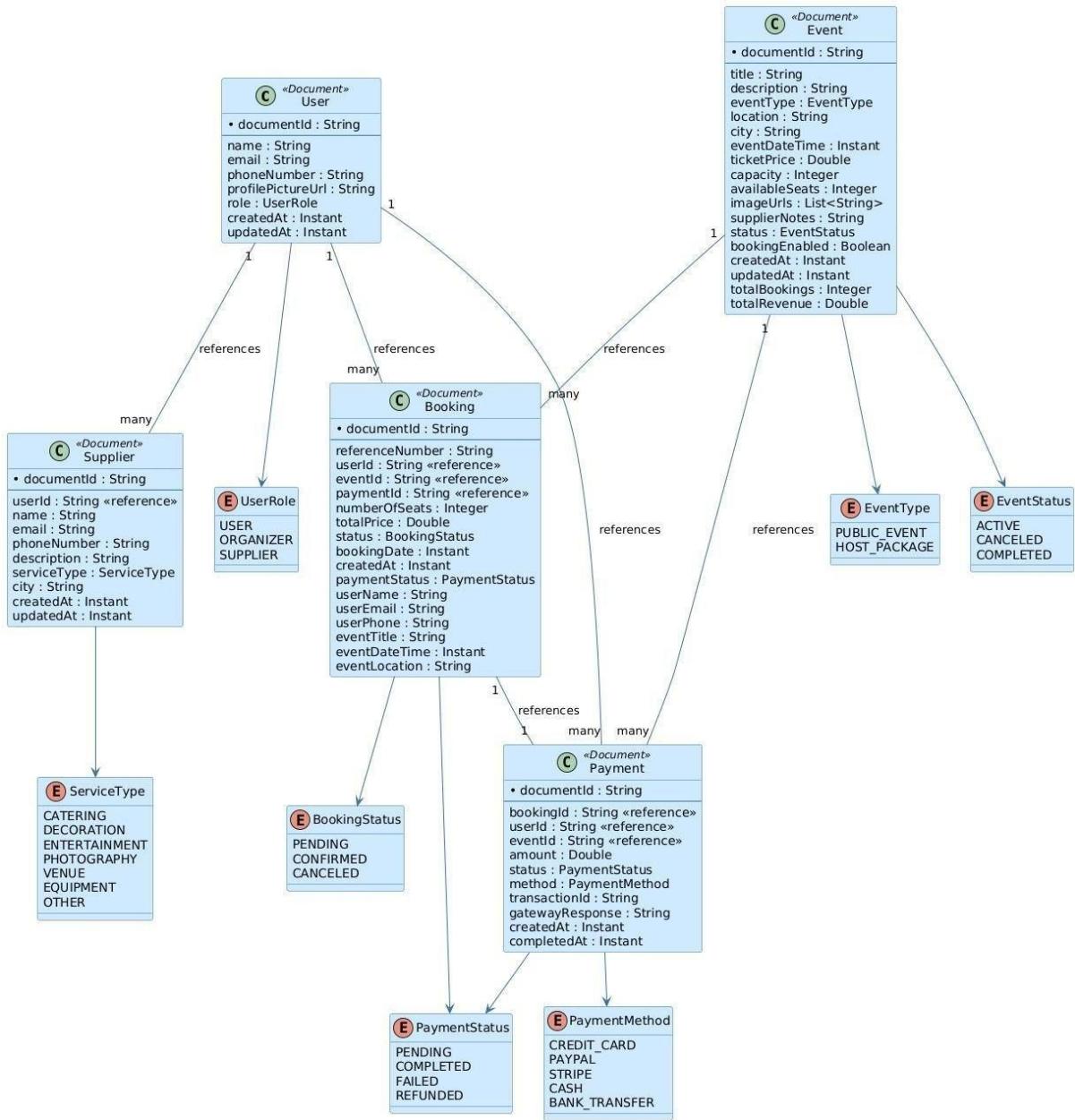


Figure 43:Data Model Diagram

6.Requirement Trackability Matrix (RTM) (V.3):

Requirements Id	Title	Use case	System Design	Analysis	Detailed Design	Coding	Test Cases	change log
REQ-FR-01	The system should allow users and service providers to register with basic information.	UC-02						
REQ-FR-02	The system should validate email format, phone number, and password strength.	UC-02						
REQ-FR-03	The system should allow users, service providers, and admins to log in and log out.	UC-02						
REQ-FR-04	The system should allow users to reset their password via an email link.	UC-02						
REQ-FR-05	The system should display the user profile information.	UC-03						
REQ-FR-06	The system should allow users to edit their name, email, phone number, and profile picture.	UC-03						
REQ-FR-07	The system should allow users to change their password.	UC-03						
REQ-FR-08	The system should allow users to delete their account after password confirmation.	UC-03						
REQ-FR-09	The system should display all available events on the homepage.	UC-04						

REQ-FR-10	The system should allow users to filter events by date, location, price, and event type.	UC-04					
REQ-FR-11	The system should display event cards containing title, description, image, date, time, location, and availability.	UC-04					
REQ-FR-12	The system should allow users to search events by title or keywords.	UC-04					
REQ-FR-13	The system should display available seats for each event.	UC-04					
REQ-FR-14	The system should display full event details, including description, images, capacity, and ticket price.	UC-04					
REQ-FR-15	The system should display booking availability status for each event.	UC-04					
REQ-FR-16	The system should display vendor information associated with the event.	UC-04					
REQ-FR-17	The system should prevent booking when event capacity is full.	UC-04					
REQ-FR-18	The system should require users to log in before completing a booking.	UC-04					
REQ-FR-19	The system should generate a unique booking reference number upon confirmation.	UC-04					This requirement has been added

REQ-FR-20	The system should immediately reduce available capacity after booking.	UC-04						This requirement has been added
REQ-FR-21	The system should send a booking confirmation email with the reference number.	UC-06						
REQ-FR-22	The system should display hosting packages with setup details.	UC-05						
REQ-FR-23	The system should allow users to book a hosting package for a specific date or location.	UC-05						
REQ-FR-24	The system should display upcoming bookings for the user.	UC-04						
REQ-FR-25	The system should display completed bookings.	UC-04						
REQ-FR-26	The system should allow cancellation of confirmed bookings.	UC-04						
REQ-FR-27	The system should restore event capacity when a booking is canceled.	UC-04						
REQ-FR-28	The system should allow organizers to create new public events.	UC-07						
REQ-FR-29	The system should require mandatory fields for events: title, description, location, date, time, capacity, and ticket price.	UC-04						This requirement has been added
REQ-FR-30	The system should allow organizers to edit existing event details.	UC-07						

REQ-FR-31	The system should allow organizers to delete events.	UC-07						This requirement has been added
REQ-FR-32	The system should allow organizers to enable or disable booking for an event.	UC-07						This requirement has been added
REQ-FR-33	The system should allow organizers to create new hosting packages.	UC-07						
REQ-FR-34	The system should require mandatory fields for hosting packages: title, description, setup details, location options, and price.	UC-05						This requirement has been added
REQ-FR-35	The system should allow defining customizable elements within packages.	UC-05						This requirement has been added
REQ-FR-36	The system should allow organizers to edit hosting package details.	UC-07						
REQ-FR-37	The system should allow organizers to delete hosting packages.	UC-07						
REQ-FR-38	The system should allow organizers to filter events by status: all, active, canceled, or completed.	UC-07						This requirement has been added
REQ-FR-39	The system should allow organizers to search events within the dashboard.	UC-07						This requirement has been added

REQ-FR-40	The system should display event type, city, date, status, bookings, statistics, and revenue in the dashboard.	UC-07						This requirement has been modified
REQ-FR-41	The system should display total bookings, remaining seats, and total revenue per event.	UC-07						This requirement has been added
REQ-FR-42	The system should display total created events, active events, and total revenue.	UC-07						This requirement has been added
REQ-FR-43	The system should allow registering vendors with business name, service type, city, description, and contact information.	UC-01						
REQ-FR-44	The system should allow searching vendors by service type or city.	UC-01						
REQ-FR-45	The system should provider vendor service categories such as catering, photography, venues, entertainment, decoration, audio/visual, and transportation.	UC-08						
REQ-FR-46	The system should send email notifications upon registration (success or failure).	UC-06						
REQ-FR-47	The system should send password reset links and confirmation emails.	UC-06						

REQ-FR-48	The system should send booking notifications for confirmation and cancellation.	UC-06						This requirement has been added
REQ-FR-49	The system should notify attendees of event cancellations or changes in date or location.	UC-01						
REQ-FR-50	The system should allow guests to browse all events without registration.	UC-01						
REQ-FR-51	The system should allow guests to search and filter events.	UC-01						This requirement has been added
REQ-FR-52	The system should display full event details to guests.	UC-01						
REQ-FR-53	The system should prevent guests from booking without logging in.	UC-01						This requirement has been added
REQ-FR-54	The system should redirect guests to the registration page when attempting to book.	UC-01						
REQ-FR-55	The system should allow guests to browse the vendor directory.	UC-01						This requirement has been added
REQ-FR-56	The system shall send reminder email 24 hours before event							This requirement has been removed
REQ-NFR-01	The system should be easy to use and clear, even for non-technical users.							

REQ-NFR-02	The system should load main pages (Home, Events, Booking) within less than 3 seconds.						
REQ-NFR-03	The system should be available at least 99% of the time.						
REQ-NFR-04	The system should protect user data and not share it with any external party.						
REQ-NFR-05	The system should work correctly on modern browsers (Chrome, Firefox, Edge).						
REQ-NFR-06	The system should handle an increasing number of users and events without issues.						This requirement has been modified
REQ-NFR-07	Any successful booking cannot be lost or duplicated.						
REQ-NFR-08	Notifications should be delivered on time without noticeable delay.						This requirement has been modified

Table 51 : RTM V.3

7. Summary:

In this chapter, we focused on designing the system in a clear and organized way to ensure high performance and reliability. We adopted a Layered Architecture, where each layer—such as Controllers, Services, and Repositories—handles specific tasks like request processing, core business logic, and database interaction. This separation of concerns makes the system significantly easier to maintain, debug, and extend as the application grows.

We divided the system into specialized, modular services designed to handle the complexities of event planning, including:

EventService: To manage event creation and scheduling.

PackageService: To handle the customization of specialized service bundles.

BookingService: To streamline the reservation process and payment status.

UserService & Analysis: To manage user authentication and provide data-driven insights.

To build a secure, responsive, and scalable platform, we utilized a modern technology stack featuring Java Spring Boot for the backend, integrated with Firebase for real-time data management and authentication, and React for a dynamic user interface.

We also developed detailed diagrams illustrating how these components interact, providing a clear roadmap for efficient development. Finally, we defined the requirements for each module and linked them directly to the design, implementation, and testing phases, ensuring the system's quality, completeness, and alignment with the needs of event organizers and clients.

Chapter6:

Implementation

1.Introduction

This section provides a visual overview of the system's core interfaces through a series of detailed screenshots. These screenshots help illustrate the overall design of key interactive elements and explain how different user groups, including regular users, event organizers, and service providers, interact with the system. This contributes to enhanced usability and a seamless, efficient user experience.

2.Purpose:

This section aims to visually review the core interfaces of the Events and Activities Planning Platform System, explaining the principles upon which their design is based. By presenting screenshots with brief descriptions of each interface, the expected user experience is clarified, the most important interactive elements are highlighted, and how these interfaces support interaction between users, organizers, and service providers with the system's various functions are explained. This also allows stakeholders to understand the system's visual design and provide constructive feedback for its improvement.

3.User Interfaces

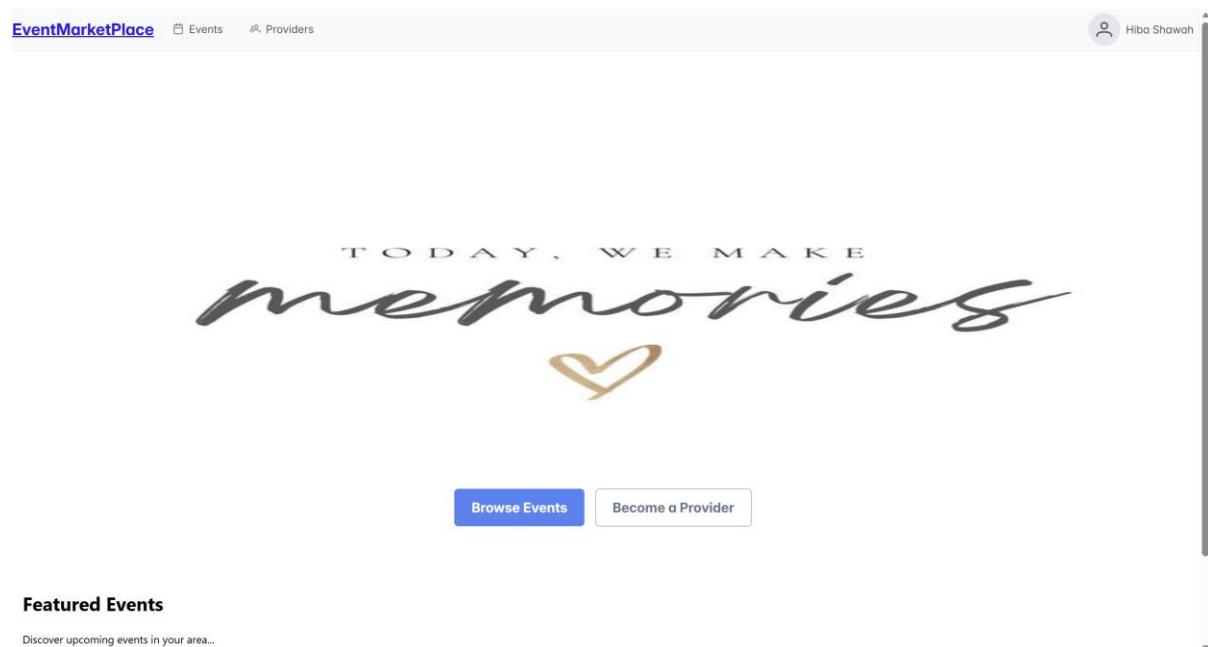


figure 44:user interface

Discover Events

Find and book amazing events or host packages

Search

City

Event Type

Start Date

End Date

Min Price

Max Price

Reset
Search

4 events found



Test Event

Test Event...

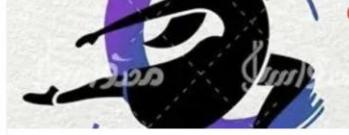
Public

12/10/2025 - 12/10/2025 | Damascus

39 spots left

\$50.00

[View Details](#)



Mosaic of a Nation

A grand celebration of Syria's rich cultural diversity at the historic Damascus Citadel...

Public

12/10/2025 - 12/10/2025 | Damascus

Sold Out

\$0.00

[View Details](#)



Chords of Damascus

Enjoy a variety of periodic film screenings and musical concerts. A recent highlight was the screen...

Public

12/10/2025 - 12/10/2025 | Damascus

147 spots left

\$5.00

[View Details](#)

figure 45:userinterface<<2>>



Hosting a

...

12/10/2025 - 12/10/2025 | Damascus

50 spots left

\$350.00

[View Package](#)



Mosaic of a Nation

A grand celebration of Syria's rich cultural diversity at the historic Damascus Citadel...

Public

12/10/2025 - 12/10/2025 | Damascus

Sold Out

\$0.00

[View Details](#)



Test4

Test4...

12/10/2025 - 12/10/2025 | Test4

1 spots left

\$0.00

[View Details](#)



A Night Of Tarab

the evening aims to revive refined musical taste...

Public

12/10/2025 - 12/10/2025 | Damascus

194 spots left

\$75.00

[View Details](#)



Test disabled

Test disabled...

Public

12/10/2025 - 12/10/2025 | Damascus

59 spots left

\$60.00

[View Details](#)



Hosting a Birthday

A birthday celebration filled with fun and laughter...

Host

12/10/2025 - 12/10/2025 | Damascus

60 spots left

\$500.00

[View Package](#)

figure 46:user interface<<3>>



Chords of Damascus

Public Event ACTIVE

Booking Open

About This Event

Enjoy a variety of periodic film screenings and musical concerts. A recent highlight was the screening of "The Bad Son" by director Ghatafan Ghannoum.

Event Details

Date Monday, December 22, 2025

Time 8:15 PM

Location Umayyad Square, Damascus

Availability 148 / 150 spots available

Total Bookings 2 bookings

Ticket Price
\$5.00

[Book Ticket](#)

[Contact Organizer](#)

Free cancellation up to 24 hours before the event

figure 47:userinterface<<4>>

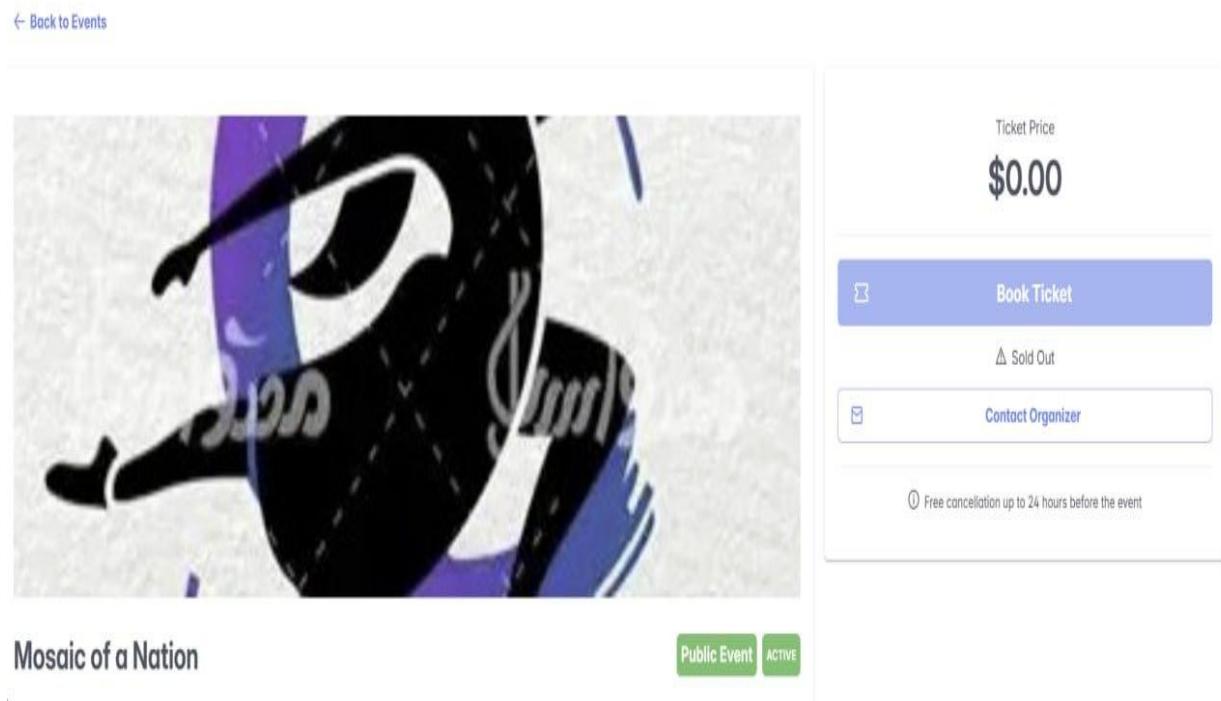


figure 48:user interface<<5>>

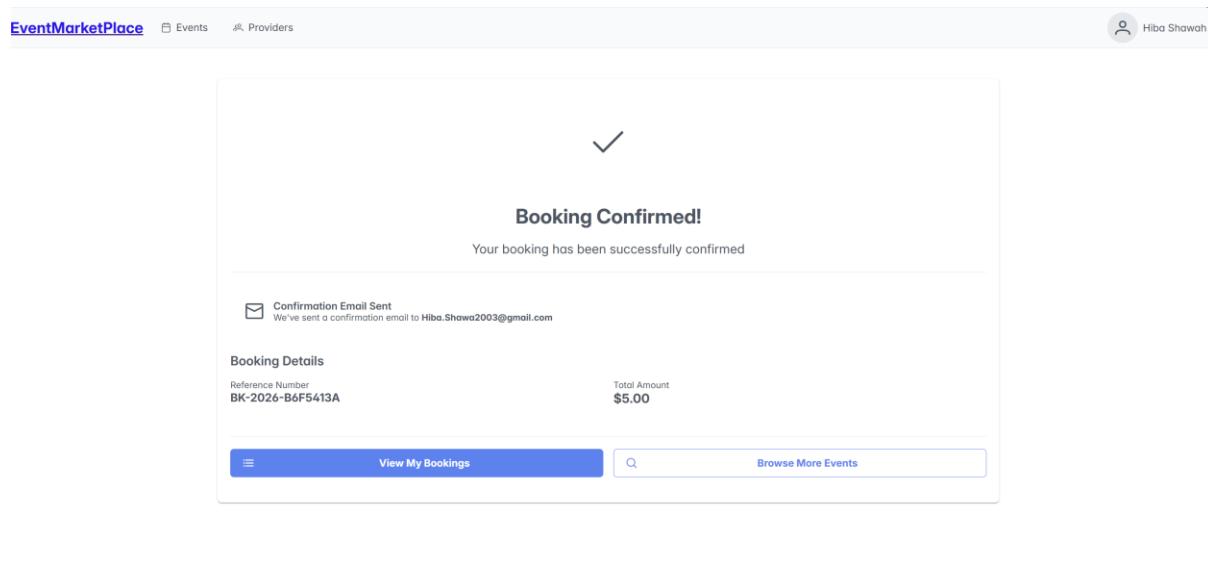


figure 49:user interface<<6>>



Hosting a Birthday

Booking Open

About This Event

A birthday celebration filled with fun and laughter

Host Package ACTIVE

Event Details

Date Tuesday, December 30, 2025

Time 1:15 PM

Location uptown, Damascus

Availability 60 / 60 spots available

Free cancellation up to 24 hours before the event

Book Package

Contact Organizer

figure 50:user interface<<8>>

Provider Directory

Find the perfect Providers for your event

Search suppliers...

All Services

All Cities

Showing 7 Providers

Lolo
 Decoration
Damascus

Modern deco!
 su@su.com
 +963552145854

Contact Provider

Yaman
 Entertainment
Damascus

The Entertainment the best thing
 Yaman2020@gmail.com

Contact Provider

Supplier Fast
 Catering
Damascus

Best Food Ever!
 sun@sun.ae
 +963992154785

Contact Provider

Sama
 Decoration
Damascus

create detailed drawings,plans and 3D models
 SamaSleib2020@gmail.com

Contact Provider

Lama
 Photography

Sara
 Decoration

Ahmad
 Catering

figure 51:user interface<<9>>

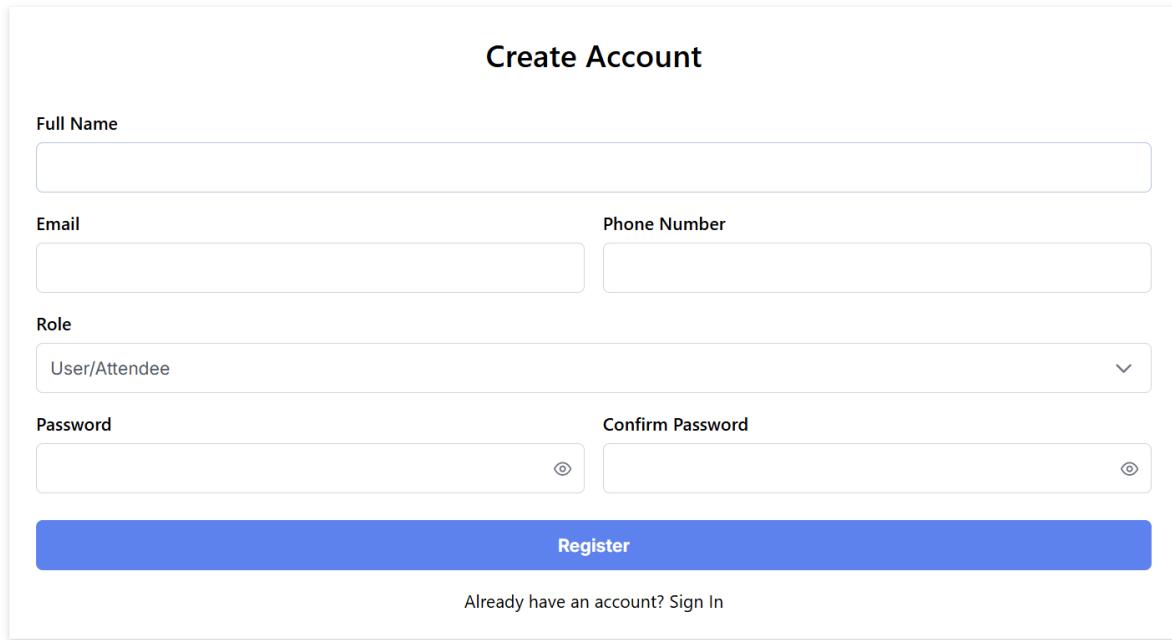
1.Sign in



The sign-in interface features a central title "Sign In". Below it are two input fields: "Email" and "Password". A "Forgot password?" link is positioned between them. A large blue "Sign In" button is centered at the bottom, with a cursor icon indicating it is active. Below the button is a link "Don't have an account? Register".

figure 52: sign interface <<1>>

2. New Register For User



The registration interface has a title "Create Account". It includes fields for "Full Name" (a single input field), "Email" and "Phone Number" (two separate input fields side-by-side), "Role" (a dropdown menu showing "User/Attendee"), "Password" and "Confirm Password" (two input fields with eye icon password helpers), and a large blue "Register" button at the bottom. Below the button is a link "Already have an account? Sign In".

figure 53: New Register For User <<1>>

Check Your Email



We've sent password reset instructions to **hibashawah2003@gmail.com**

Please check your inbox and follow the link to reset your password.

[Back to Login](#)

figure 54: New Register For User <<2>>

3. New Register For provider

Create Account

Full Name

Email Phone Number

Role

Password Confirm Password

Provider Details
Tell us more about your service so clients can find you.

City Service Type

Description

Register

Already have an account? [Sign In](#)

figure 55 : New Register For Supplier<<1>>

4. Dashboard For Organizer

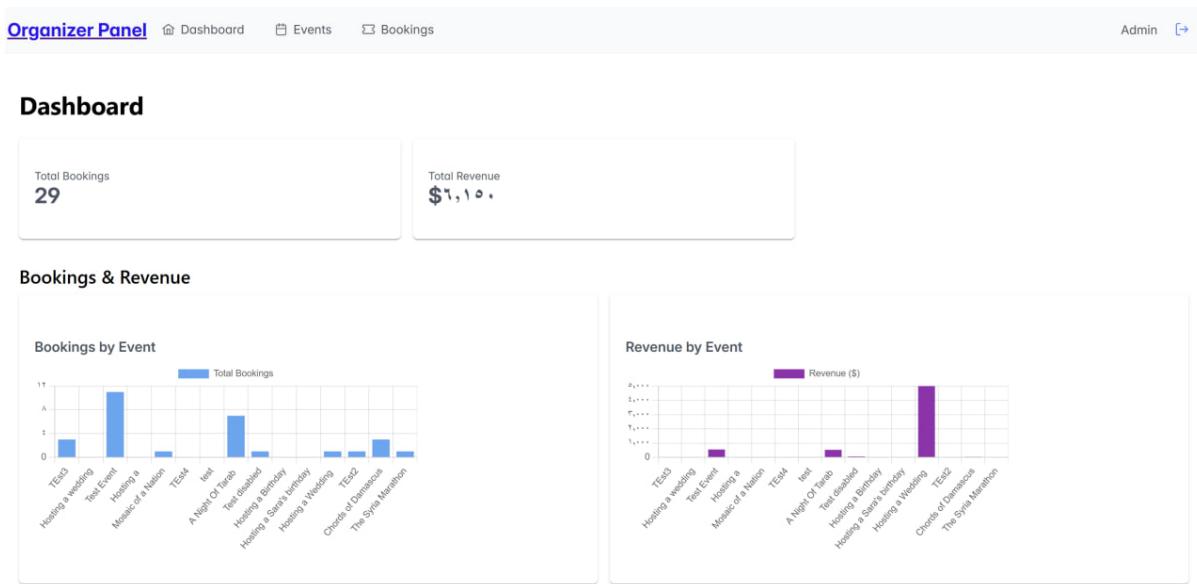


figure 56: Dashboard For Organizer <<1>>

Recent Bookings

Event	Customer	Tickets	Amount	Status	Date
TEst2	Admin		\$0.00	CONFIRMED	٢٠٢٣/١/٨
TEst3	Hiba Shawah		\$0.00	CONFIRMED	٢٠٢٣/١/٩
TEst3	Hiba Shawah		\$0.00	CONFIRMED	٢٠٢٣/١/٩
TEst3	Hiba Shawah		\$0.00	CONFIRMED	٢٠٢٣/١/٩
TEst3	Hiba Shawah		\$0.00	CONFIRMED	٢٠٢٣/١/٩
TEst3	Hiba Shawah		\$0.00	CONFIRMED	٢٠٢٣/١/٩
TEst3	Hiba Shawah		\$0.00	CONFIRMED	٢٠٢٣/١/٩
TEst3	Hiba Shawah		\$0.00	CONFIRMED	٢٠٢٣/١/٩
A Night Of Tarab	Hiba Shawah		\$75.00	CONFIRMED	٢٠٢٣/١/٩
Chords of Damascus	Hiba Shawah		\$5.00	CONFIRMED	٢٠٢٣/١/٨

figure 57: Dashboard For Organizer <<2>>

5. Dashboard For Supplier

The screenshot shows the 'Supplier Directory' section of the EventMarketPlace platform. At the top, there are navigation links for 'Events' and 'Suppliers'. A user profile for 'Hiba Shawah' is visible on the right. Below the header, the title 'Supplier Directory' is displayed with the subtitle 'Find the perfect suppliers for your event'. There is a search bar labeled 'Search suppliers...' and two dropdown filters: 'All Services' and 'All Cities'. A message at the bottom states 'Showing 0 suppliers' with a small envelope icon. A note below says 'No suppliers found matching your criteria' and 'Try adjusting your filters'.

figure 58: Dashboard For Supplier<1>>

6. Manage events :

The screenshot shows the 'Manage Events' section of the Organizer Panel. At the top, there are navigation links for 'Dashboard', 'Events', and 'Bookings'. An 'Admin' link is on the right. The main area is titled 'Manage Events' and features a table with columns: 'Event Title', 'Type', 'City', 'Date', 'Status', 'Bookings', 'Stats', 'Revenue', and 'Actions'. The table lists several events, each with a status indicator (e.g., ACTIVE, COMPLETED, CANCELED) and booking details. A 'Creates Event' button is located at the top right of the table. At the bottom, there is a pagination control showing page 1 of 10.

Event Title	Type	City	Date	Status	Bookings	Stats	Revenue	Actions
TEst3	Public	TEst3	٢٠٢٣/١٢/٢٥	ACTIVE	Open	3 bookings 101 / 104 available	\$0.00	
Hosting a wedding	Host	Damascus	٢٠٢٣/١٢/٢٦	COMPLETED	Open	0 bookings 200 / 200 available	\$0.00	
Test Event	Public	Damascus	٢٠٢٣/١٢/٢٧	ACTIVE	Closed	11 bookings 39 / 50 available	\$550.00	
Hosting a	Host	Damascus	٢٠٢٣/١٢/٢٨	ACTIVE	Open	0 bookings 50 / 50 available	\$0.00	
Mosaic of a Nation	Public	Damascus	٢٠٢٣/١٢/٢٩	ACTIVE	Open	1 bookings 0 / 1 available	\$0.00	
TEst4	Public	TEst4	٢٠٢٣/١٢/٣١	ACTIVE	Closed	0 bookings 1 / 1 available	\$0.00	
test	Public	damascus	٢٠٢٣/١٢/٣١	ACTIVE	Open	0 bookings 100 / 100 available	\$0.00	
A Night Of Tarab	Public	Damascus	٢٠٢٣/١٢/٣١	ACTIVE	Open	7 bookings 193 / 200 available	\$525.00	
Test disabled	Public	Damascus	٢٠٢٣/١٢/٣١	CANCELED	Open	1 bookings 59 / 60 available	\$60.00	
Hosting a Birthday	Host	Damascus	٢٠٢٣/١٢/٣١	ACTIVE	Open	0 bookings 60 / 60 available	\$0.00	

figure 59: Manage events <2>>

6.Create New Event

The screenshot shows the 'Create New Event' interface. At the top, there's a navigation bar with 'Organizer Panel', 'Dashboard', 'Events', 'Bookings', 'Admin', and a user icon. Below the navigation is a title 'Create New Event' with a 'Cancel' button. The main area is divided into sections: 'Basic Information', 'Location Details', and 'Date & Time'. In 'Basic Information', there are fields for 'Event Title' (containing 'I'), 'Description' (empty), 'Event Type' (set to 'Public Event'), and 'Status' (set to 'Active'). To the right, an 'Event Preview' panel displays summary information: Title: Untitled Event, Type: Public Event, Location: Not set, Not set, Date & Time: Not set, Price: \$0.00, Capacity: 1 spots, Status: ACTIVE, and Bookings: Enabled.

figure60: Create New Event <1>>

Pricing & Capacity

Ticket Price (\$): \$0.00

Capacity *: 1

Enable Bookings

Media

Event Images

+ Select Images

Additional Information

Supplier Notes

X Cancel

Event Preview

Title: A Night Of Tarab

Type: Public Event

Location: Four Seasons Hotel, Damascus

Date & Time: ٢٠٢٣-١٠-٢٨ ١٠:٥٠ AM

Price: \$0.00

Capacity: 1 spots

Status: ACTIVE

Bookings: Enabled

figure 61 : Create New Event <<2>>

7.Edit Events

Organizer Panel Admin

Edit Event

Basic Information

Event Title *: Mosaic of a Nation

Description *: A grand celebration of Syria's rich cultural diversity at the historic Damascus Citadel.

Event Type *: Public Event

Status *: Active

Location Details

City *: Damascus

Venue/Location *: Damascus Castle

Date & Time

Event Date & Time *: 12/28/2025 05:30 PM

Event Preview

Title: Mosaic of a Nation

Type: Public Event

Location: Damascus Castle, Damascus

Date & Time: ٢٠٢٥-١٢-٢٨ ٠٥:٣٠ PM

Price: \$0.00

Capacity: 1 spots

Status: ACTIVE

Bookings: Enabled

figure 62 : Edit Event <<1>>

Pricing & Capacity

Ticket Price (\$) Capacity * Enable Bookings

Media

Event Images

Additional Information

Provider Notes

figure 63 : Edit Event <<2>>

8.Delete event:

Organizer Panel Admin

Manage Events

Event Title ↑	Type ↑	City ↑	Date ↑	Status ↑	Bookings ↑	Stats	Revenue ↑	Actions
TEst3	Public	TEst3	1-12/11/11	ACTIVE	Open	3 bookings 101 / 104 available	\$0.00	<input type="button"/> <input type="button"/> <input type="button"/>
Hosting a wedding	Host	Damascus	1-12/11/11	COMPLETED	Open	0 bookings 200 / 200 available	\$0.00	<input type="button"/> <input type="button"/> <input type="button"/>
Test Event	Public	Damascus	1-12/11/11	ACTIVE	Open	11 bookings 39 / 50 available	\$550.00	<input type="button"/> <input type="button"/> <input type="button"/>
Hosting a	Host	Damascus	1-12/11/11	ACTIVE	Open	0 bookings 50 / 50 available	\$0.00	<input type="button"/> <input type="button"/> <input type="button"/>
Mosaic of a Nation	Public	Damascus	1-12/11/11	ACTIVE	Open	1 bookings 0 / 1 available	\$0.00	<input type="button"/> <input type="button"/> <input type="button"/>
TEst4	Public	TEst4	1-12/11/11	ACTIVE	Closed	0 bookings 1 / 1 available	\$0.00	<input type="button"/> <input type="button"/> <input type="button"/>

Delete Event

⚠ Are you sure you want to delete "Test Event"?

figure 64 : Delete Event <<1>>

9. Disable / enable bookings:

Event Title	Type	City	Date	Status	Bookings	Stats	Revenue	Actions
TEst3	Public	TEst3	٢٠٢١/١٢/٢٣	ACTIVE	Open	3 bookings 101 / 104 available	\$0.00	Edit Delete Disable Bookings
Hosting a wedding	Host	Damascus	٢٠٢١/١٢/٢٤	COMPLETED	Open	0 bookings 200 / 200 available	\$0.00	Edit Delete
Test Event	Public	Damascus	٢٠٢١/١٢/٢٥	ACTIVE	Closed	11 bookings 39 / 50 available	\$550.00	Edit Delete
Hosting a	Host	Damascus	٢٠٢١/١٢/٢٦	ACTIVE	Open	0 bookings 50 / 50 available	\$0.00	Edit Delete
Mosaic of a Nation	Public	Damascus	٢٠٢١/١٢/٢٧	ACTIVE	Open	1 bookings 1 / 1 available	\$0.00	Edit Delete

figure 65 : Disable / enable bookings <<1>>

9. Booking Management

Reference	Event	Customer	Amount	Status	Booked On	Event Date	Actions
BK-2025-B0331A22	A Night Of Tarab	Hiba Shawah	\$75.00	CONFIRMED	٢٠٢١/١٢/٢٣	٢٠٢١/١٢/٢٣	@
BK-2025-F2270149	A Night Of Tarab	Hiba Shawah	\$75.00	CONFIRMED	٢٠٢١/١٢/٢٣	٢٠٢١/١٢/٢٣	@
BK-2025-2A7C4BC4	Host a wedding	Hiba	\$500.00	CONFIRMED	٢٠٢١/١٢/٢٣	٢٠٢١/١٢/٢٣	@
BK-2025-D38E8E69	A Night Of Tarab	Hiba Shawah	\$75.00	CONFIRMED	٢٠٢١/١٢/٢٣	٢٠٢١/١٢/٢٣	@
BK-2025-68F69C39	A Night Of Tarab	Hiba Shawah	\$75.00	CONFIRMED	٢٠٢١/١٢/٢٣	٢٠٢١/١٢/٢٣	@
BK-2025-BF2CF4C6	A Night Of Tarab	Hiba Shawah	\$75.00	CONFIRMED	٢٠٢١/١٢/٢٣	٢٠٢١/١٢/٢٣	@
BK-2025-9ADE07B7	A Night Of Tarab	Hiba Shawah	\$75.00	CONFIRMED	٢٠٢١/١٢/٢٣	٢٠٢١/١٢/٢٣	@
BK-2026-44079F6B	TEst3	Hiba Shawah	\$0.00	CONFIRMED	٢٠٢١/١٢/٢٤	٢٠٢١/١٢/٢٤	@

figure 66: Booking Management <<1>>

10. My Bookings

The screenshot shows the 'My Bookings' section of the EventMarketPlace website. At the top, there are navigation links for 'EventMarketPlace', 'Events', 'Suppliers', and a user profile for 'Hiba Shawah'. Below this, the title 'My Bookings' is displayed, followed by the sub-instruction 'View and manage your event bookings'. A filter bar at the top indicates 'Upcoming (4)' and 'Past (0)'. The bookings are listed in a grid format:

- The Syria Marathon** (Confirmed)
Fri, Dec 26, 2025 | Departure from Al-Fayha Stadium | 1 ticket | # BK-2025-C6CD53E3
\$0.00 Booked: 1-10/11/21
View Details Cancel
- The Syria Marathon** (Confirmed)
Fri, Dec 26, 2025 | Departure from Al-Fayha Stadium | 1 ticket | # BK-2025-CAB22576
\$0.00 Booked: 1-10/11/21
View Details Cancel
- Chords of Damascus** (Confirmed)
Mon, Dec 22, 2025 | Umayyad Square | 1 ticket | # BK-2025-56C579A4
\$5.00 Booked: 1-10/11/21
View Details Cancel
- Chords of Damascus** (Confirmed)
Mon, Dec 22, 2025 | Umayyad Square | 1 ticket | # BK-2025-56C579A4
\$5.00 Booked: 1-10/11/21
View Details Cancel

figure 67: My Bookings <<1>>

11. User Profile for provider

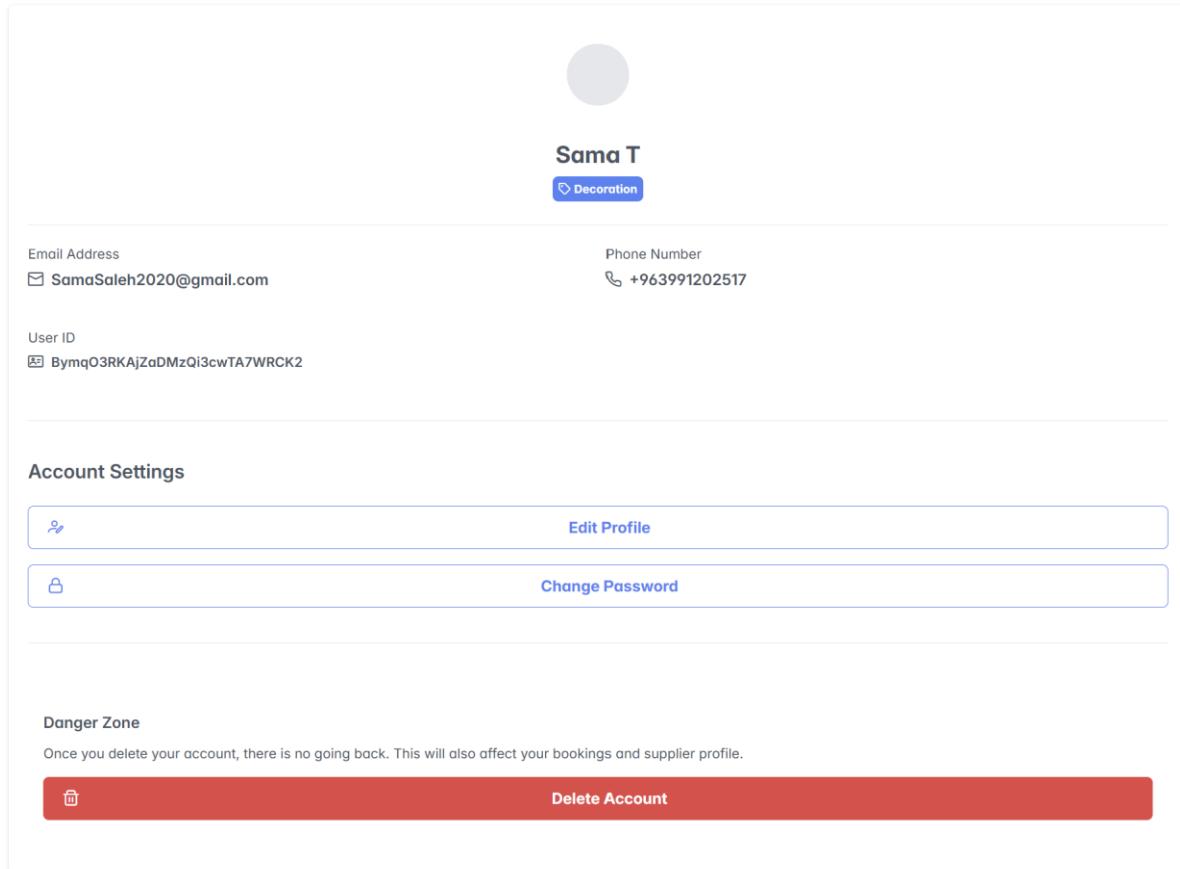


figure 68: User Profile for provider <<1>>

4. REST API Documentation – Events and Activities Planning Platform System

4.1. Introduction

4.1. Introduction

This document provides a comprehensive reference for the RESTful APIs of the Events and Activities Planning Platform. This platform is designed to support efficient event management and organization by enabling event organizers to create public invitations, organize public and private events, and manage associated bookings. Service providers, on the other hand, are limited to creating a profile to showcase their information and services within the platform, without managing events or other content. These APIs allow third-party applications, such as web or mobile apps, to interact with the system to perform a range of essential tasks, including registering users, organizers, and providers; creating and managing invitations and events of various types; and processing bookings. The platform also provides visual graphs illustrating booking and revenue metrics for each event, supporting an understanding of event performance without relying on detailed analytical reports.

4.2. Purpose

The purpose of this documentation is to explain how to use each API endpoint in the system, including their paths, HTTP methods, required parameters, and the structure of both requests and responses. This aims to make integration easier and to ensure that the system is used properly and efficiently by developers and technical teams.

4.3. API Structure Overview

The APIs in this system are systematically categorized into four main sections. Each section is responsible for a specific component of Events and Activities Planning Platform System. This modular organization makes it easier for developers to navigate and utilize the APIs based on their intended functionality:

Section	Primary Function
SupplierController	Manages supplier profiles linked to existing users, including operations to register a new supplier, retrieve supplier information by user ID, update supplier details, and delete supplier profiles.
EventController	Manages event operations, including creation, retrieval, updating, deletion, and booking management, providing full event lifecycle control.
BookingController	Handles event bookings by allowing users to create, view, and cancel bookings, while managing booking and payment information.

DashboardController	Aggregates event and booking data to provide dashboard statistics, including bookings, capacity, revenue, and recent booking activity.
AuthenticationController	Manages user authentication and profile operations, including user registration, identity verification via Firebase, profile updates with optional profile picture upload, and permanent account deletion.

Table 52 : API Structure Overview

1. SupplierController API Documentation:

1.1. Register New Supplier

- **Endpoint:** /api/Supplier

- **Method:** POST

- **Description:** Registers a new supplier profile. The supplier information is linked to an existing user account.

- **Parameters:** None

- **Request Body:**{

```
"userId": "string",
"description": "string",
"serviceType": "enum",
"city": "string",}
```

In addition: The name, email, and phoneNumber fields are automatically populated from the associated user profile.

-Response:

200 OK with the created Supplier object.

400 Bad Request with error message.

1.2. Get Supplier by User ID

- **Endpoint:** /api/suppliers/user/{userId}

- **Method:** GET

- **Description:** Retrieves a supplier profile associated with a specific user ID.

- Parameters:

userId (path, string, required): The ID of the user linked to the supplier profile.

- Request Body: None

- Response:

200 OK with the Supplier object.

404 Not Found if no supplier exists for the user.

400 Bad Request with error message.

1.3. Get All Suppliers

- Endpoint: /api/suppliers

- Method: GET

- Description: Retrieves a list of all registered suppliers.

- Parameters: None

- Request Body: None

- Response:

200 OK with a list of Supplier objects.

400 Bad Request with error message.

1.4. Update Suppliers

- Endpoint: /api/suppliers/{id}

- Method: PUT

- Description: Updates an existing supplier profile. Preserves the original createdAt timestamp.

- Parameters:

id (path, string, required): The ID of the supplier to update.

- Request Body:

"userId": "string",

"name": "string",

"email": "string",

```
"phonenumer": "string",  
"description": "string",  
"serviceType": "enum",  
"city": " string ",  
"createdAt": " Instant ",  
"updateAt": "Instant",}
```

- Response:

200 OK with the updated Supplier object.

400 Bad Request if the update fails.

500 Internal Server Error.

1.5. Delete Supplier

- Endpoint: /api/suppliers/{id}

- Method: DELETE

- Description: Deletes a supplier profile by ID.

- Parameters:

id (path, string, required): The ID of the supplier to delete.

- Request Body: None

- Response:

200 OK with confirmation message "Supplier deleted".

400 Bad Request with error message.

2. EventController API Documentation:

2.1. Create Event

- Endpoint: /api/events

- Method: POST

- Description: Creates a new event in the system.

- **Parameters:** None

- **Request Body:** Event object containing the event details (title, description, type, location, date/time, capacity, ticket price, etc.)

- **Response:**

200 OK with the created Event object

400 Bad Request with error message if creation fails

2.2. Get Event by ID

- **Endpoint:** /api/events/{id}

- **Method:** GET

- **Description:** Retrieves the details of a specific event by its ID.

- **Parameters:**

id (path, string, required): The ID of the event

- **Request Body:** None

- **Response:**

200 OK with the Event object

404 Not Found if event does not exist

400 Bad Request if there is an error

2.3. Get All Events

- **Endpoint:** /api/events

- **Method:** GET

- **Description:** Retrieves a list of all events.

- **Parameters:** None

- **Request Body:** None

- **Response:**

200 OK with a list of Event objects

400 Bad Request if there is an error

2.4. Update Event

- **Endpoint:** /api/events/{id}

- **Method:** PUT

- **Description:** Updates the details of an existing event.

- **Parameters:**

id (path, string, required): The ID of the event to update

- **Request Body:** Event object containing updated event details

- **Response:**

200 OK with the updated Event object

400 Bad Request with error message if update fails

2.5. Toggle Event Booking

- **Endpoint:** /api/events/{id}/toggle-booking

- **Method:** PUT

- **Description:** Toggles the booking availability of an event (enabled/disabled).

- **Parameters:**

id (path, string, required): The ID of the event

- **Request Body:** None

- **Response:**

200 OK with the updated Event object showing new booking status

400 Bad Request with error message if toggle fails

2.6. Delete Event

- **Endpoint:** /api/events/{id}

- **Method:** DELETE

- **Description:** Deletes an event by its ID.

- **Parameters:**

id (path, string, required): The ID of the event to delete

- **Request Body:** None

- **Response:**

200 OK with confirmation message "Event deleted"

400 Bad Request with error message if deletion fails

3. BookingController API Documentation

3.1. Create Booking

- **Endpoint:** /api/bookings
- **Method:** POST
- **Description:** Creates a new booking for a specific event.
- **Parameters:** None
- **Request Body:**

```
{  
  "eventId": "string",  
  "numberOfSeats": "integer",  
  "paymentMethod": "string"  
}
```

- **Response:**

201 Created with the created Booking object and associated Payment object, or 400 Bad Request / 500 Internal Server Error.

3.2. Get Booking by ID

- **Endpoint:** /api/bookings/{id}
- **Method:** GET
- **Description:** Retrieves a booking by its ID.
- **Parameters:**

id (path, string, required): The ID of the booking.

- **Request Body:** None

- **Response:**

200 OK with the Booking object, or 404 Not Found, or 500 Internal Server Error.

3.3. Get All Bookings

- **Endpoint:** /api/bookings
- **Method:** GET
- **Description:** Retrieves all bookings in the system.

- **Parameters:** None

- **Request Body:** None

- **Response:**

200 OK with a list of Booking objects, or 500 Internal Server Error.

3.4. Get Bookings by User ID

- **Endpoint:** /api/bookings/user/{userId}

- **Method:** GET

- **Description:** Retrieves all bookings for a specific user.

- **Parameters:**

userId (path, string, required): The ID of the user.

- **Request Body:** None

- **Response:**

200 OK with a list of Booking objects, or 500 Internal Server Error.

3.5. Get Upcoming Bookings for User

- **Endpoint:** /api/bookings/user/{userId}/upcoming

- **Method:** GET

- **Description:** Retrieves upcoming bookings for a specific user (events not yet occurred and not canceled).

- **Parameters:**

userId (path, string, required): The ID of the user.

- **Request Body:** None

- **Response:**

200 OK with a list of upcoming Booking objects, or 500 Internal Server Error.

3.6. Get Past Bookings for User

- **Endpoint:** /api/bookings/user/{userId}/past

- **Method:** GET

- **Description:** Retrieves past bookings for a specific user (events already occurred and not canceled).

- **Parameters:**

userId (path, string, required): The ID of the user.

- **Request Body:** None

- **Response:**

200 OK with a list of past Booking objects, or 500 Internal Server Error.

3.7. Cancel Booking

- **Endpoint:** /api/bookings/{id}/cancel

- **Method:** PUT

- **Description:** Cancels a booking by ID, only allowed by the booking owner. Payment status is set to refunded.

- **Parameters:**

id (path, string, required): The ID of the booking.

- **Request Body:** None

- **Response:**

200 OK with the updated Booking object, or 403 Forbidden if user not allowed, or 404 Not Found, or 500 Internal Server Error.

4. AuthController API Documentation

4.1 Register New User

- **Endpoint:** /api/auth/register

- **Method:** POST

- **Description:** Registers a new user account using Firebase Authentication and stores the user profile data in Firestore. Upon successful registration, a welcome email is sent to the user.

- **Parameters:** None

- **Request Body:**{

 "name": "string",

 "email": "string",

 "password": "string",

```
"phoneNumber": "string (optional)", }
```

- **Response:**

200 OK with the created User object

400 Bad Request if registration fails

4.2 Verify User Token (Login Verification)

- **Endpoint:** /api/auth/verify

- **Method:** POST

- **Description:** Verifies a Firebase authentication token and retrieves the corresponding user profile from the database.

- **Parameters:**

Authorization (header, string, required): Bearer Firebase ID Token

- **Request Body:** None

- **Response:**

200 OK with a LoginResponse object containing the Firebase token and User data

401 Unauthorized if the token is invalid

4.3 Update User Profile

- **Endpoint:** /api/auth/user/{uid}

- **Method:** PUT

- **Description:** Updates user profile information, including optional profile picture upload. If a new email is provided, it is also updated in Firebase Authentication.

- **Parameters:**

uid (path, string, required): The unique ID of the user

- **Request Body (Multipart):**

user (User object, required)

file (MultipartFile, optional): Profile picture

- **Response:**

200 OK with the updated User object

400 Bad Request if update fails

4.4 Delete User Profile Picture

- **Endpoint:** /api/auth/user/{uid}/profile-picture
- **Method:** DELETE
- **Description:** Deletes the user's profile picture from Firebase Storage and updates the user profile accordingly.

- **Parameters:**

- uid (path, string, required): The unique ID of the user

- **Request Body:** None

- **Response:**

- 200 OK with the updated User object

- 400 Bad Request if no profile picture exists or deletion fails

4.5 Delete User Account

- **Endpoint:** /api/auth/user/{uid}
- **Method:** DELETE
- **Description:** Permanently deletes a user account from Firebase Authentication and removes the corresponding user record from Firestore.

- **Parameters:**

- uid (path, string, required): The unique ID of the user

- **Request Body:** None

- **Response:**

- 200 OK with confirmation message.

- 400 Bad Request if deletion fails.

5. DashboardController API Documentation

5.1. Get Organization Dashboard Statistics

- **Endpoint:** /api/dashboard/organization
- **Method:** GET
- **Description:** Retrieves aggregated dashboard statistics for the organization. This endpoint provides an overview of all events, including booking statistics, capacity information, revenue details, and a list of the most recent bookings.

- **Parameters:** None

- **Request Body:** None

- Response:

200 OK Returns a DashboardStats object containing:

A list of event-level statistics (EventStatsDTO) including total bookings, total capacity, available seats, and total revenue per event.

The overall number of bookings across all events.

The overall revenue generated from all events.

A list of the most recent bookings, sorted by booking date in descending order (limited to the latest 10 bookings).

500 Internal Server Error: If an unexpected server-side error occurs during data retrieval or processing.

4.4.Detailed Table of REST API Endpoints

This table presents all the REST API endpoints used in the System, organized into four main categories based on the system's core functionalities:

SupplierController, EventController, BookingController, DashboardController, AuthController. Each entry includes the HTTP method, full endpoint path, and a brief description to facilitate understanding of how each API function operates and connects to system tasks.

Controller	Endpoint	Method	Description
Supplier	/api/Supplier	POST	Registers a new supplier profile. The supplier information is linked to an existing user account.
	/api/suppliers/user/{userId}	GET	Retrieves a supplier profile associated with a specific user ID.
	/api/suppliers	GET	Retrieves a list of all registered suppliers.
	/api/suppliers/{id}	PUT	Updates an existing supplier profile. Preserves the original createdAt timestamp.
	/api/suppliers/{id}	DELETE	Deletes a supplier profile by ID.
EventController	/api/events	POST	Creates a new event in the system.
	/api/events/{id}	GET	Retrieves the details of a specific event by its ID.
	/api/events	GET	Retrieves a list of all events.
	/api/events/{id}	PUT	Updates the details of an existing event.
	/api/events/{id}/toggle-booking	PUT	Toggles the booking availability of an event (enabled/disabled).
BookingController	/api/bookings	POST	Creates a new booking for a specific event.
	/api/bookings/{id}	GET	Retrieves a booking by its ID.
	/api/bookings	GET	Retrieves all bookings in the system.
	/api/bookings/user/{userId}	GET	Retrieves all bookings for a specific user.
	/api/bookings/user/{userId}/upcoming	GET	Retrieves upcoming bookings for a specific user (events not yet occurred and not canceled).

	/api/bookings/user/{userId}/past	GET	Retrieves past bookings for a specific user (events already occurred and not canceled).
	/api/bookings/{id}/cancel	PUT	Cancels a booking by ID, only allowed by the booking owner. Payment status is set to refunded.
AuthController	/api/auth/register	POST	Registers a new user account using Firebase Authentication and stores the user profile data in Firestore. Upon successful registration, a welcome email is sent to the user.
	/api/auth/verify	POST	Verifies a Firebase authentication token and retrieves the corresponding user profile from the database.
	/api/auth/user/{uid}	PUT	Updates user profile information, including optional profile picture upload. If a new email is provided, it is also updated in Firebase Authentication.
	/api/auth/user/{uid}/profile-picture	DELETE	Deletes the user's profile picture from Firebase Storage and updates the user profile accordingly.
	/api/auth/user/{uid}	DELETE	Permanently deletes a user account from Firebase Authentication and removes the corresponding user record from Firestore.
DashboardController	/api/dashboard/organization	GET	Retrieves aggregated dashboard statistics for the organization. This endpoint provides an overview of all events, including booking statistics, capacity information, revenue details, and a list of the most recent bookings.

Table 53: API

5.Execution Test case

TC ID	Req ID	Title	Preconditions	Test Steps	Expected Result
TC-01	FR-01 ,FR-46	User Registration & Notification	User on registration page	1. Open registration page 2. Enter valid data 3. Submit form	Account is created and success email is received
TC-02	FR-02	Registration Validation	User on registration page	1. Enter invalid email/phone/password 2. Submit form	Validation error message is displayed
TC-03	FR-03	Login & Logout	User account exists	1. Login with valid credentials 2. Logout	User logs in and out successfully
TC-04	FR-04 ,FR-47	Password Recovery & Notification	Email registered in system	1. Click "Forgot Password" 2. Enter email	Password reset link sent and reset email received
TC-05	FR-07	Change Password	User logged in	1. Enter old password 2. Enter new password	Password changed successfully
TC-06	FR-05	View User Profile	User logged in	Open profile page	User profile information displayed
TC-07	FR-06	Edit User Profile	User logged in	Edit profile data and save	Profile updated successfully
TC-08	FR-08	Delete User Account	User logged in	Enter password and confirm deletion	Account deleted
TC-09	FR09, FR-50	View Events (User & Guest)	Published events exist	Open events page	Events list displayed
TC-10	FR10 ,FR-51	Search & Filter Events	Published events exist	Apply filters or search keyword	Correct results displayed
TC-11	FR11, FR-12	Event Cards & Details	Published events exist	Open event cards and details	Correct info displayed
TC-12	FR-13 ,FR-14 , FR-15	Event Capacity & Booking Status	Event published	Open event page	Available seats and booking status shown
TC-13	FR-16	View Vendor Info	Event linked to resource	View event details	Vendor info displayed

TC-14	FR-17 ,FR-18 , FR-53 , FR-54	Booking Flow & Restrictions	Event fully booked / user not logged in	Attempt to book event	Booking blocked if full or guest redirected
TC-15	FR-19 ,FR-20, FR-27	Booking Processing & Capacity Update	User logged in, event available	Complete booking, cancel, or repeat	Booking reference generated, capacity updated/restored
TC-16	FR-21, FR-48, FR-49	Booking & Event Notifications	Booking or event updated	Trigger booking/event changes	Notifications sent on time
TC-17	FR-22, FR-23	View & Book Hosting Packages	Packages available	Open packages, select & confirm booking	Packages displayed and booking successful
TC-18	FR-24 , FR-25	View Bookings (Upcoming & Past)	User has bookings	Open bookings page/history	Bookings displayed
TC-19	FR-26	Cancel Booking	User has confirmed booking	Cancel booking	Booking canceled successfully
TC-20	FR-33 , FR-36	Create & Edit Hosting Package	Organizer logged in	Enter/edit package details and save	Package created/updated
TC-21	FR-34	Validate Mandatory Package Fields	Organizer on package creation page	Submit incomplete package form	Validation error displayed
TC-22	FR-35	Customize Package Elements	Package supports customization	Select customization options	Custom options applied
TC-23	FR-37	Delete Hosting Package	Package exists	Delete package	Package deleted
TC-24	FR-28 ,FR-30, FR-31 , FR-32	Event Management (Create/Edit/Delete/Enable Booking)	Organizer logged in	Create, edit, delete, toggle booking	Events updated correctly
TC-25	FR-38, FR-39, FR-40	Dashboard & Event Search	Organizer has events	Open dashboard and search/filter events	Correct events & dashboard data displayed
TC-26	FR-41 ,FR-42	Event & General Statistics	Events have bookings	View analytics	Event/general statistics displayed
TC-27	FR-43	Vendor Registration	Vendor not registered	Register vendor	Vendor account created

TC-28	FR-44, FR-45, FR-55	Search & Filter Vendors	Providers exist	Search/filter vendors page	Correct vendors displayed
TC-29	FR-46, FR-47	Registration & Password Notifications	Vendor registered or requesting reset	Complete action	Notification email received
TC-30	NFR-01	Usability	User logged in	Navigate through main pages	User can use all features easily
TC-31	NFR-02	Performance	System under normal load	Load Home, Events, Booking pages	Pages load <3 sec
TC-32	NFR-03	Availability	System deployed	Monitor uptime	≥99% uptime
TC-33	NFR-04	Data Protection	User account exists	Attempt to access data	Data protected
TC-34	NFR-05	Compatibility	Modern browsers installed	Open system in Chrome/Firefox/Edge	Works correctly
TC-35	NFR-06	Scalability	System under load	Simulate many users/events	System handles load
TC-36	NFR-07	Reliability	Existing bookings present	Make booking/repeat	Booking not lost/duplicated
TC-37	NFR-08	Notifications	Notifications enabled	Trigger system notifications	Notifications delivered on time

Table 54: execution test cases

5. Requirement Trackability Matrix (RTM) (V.4)

Requirement Id	Title	Use Case	System Design	Analysis	Detailed Design	Coding		Test Cases
						Frontend	backend	
REQ-FR-01	The system should allow users and service providers to register with basic information.	UC-02	interface-08	seq-02	Class diagram	RegisterPage.jsx	AuthService.java	manual test cases.docx
REQ-FR-02	The system should validate email format, phone number, and password strength.	UC-02	interface-08	seq-02	Class diagram	RegisterPage.jsx	AuthService.java	manual test cases.docx
REQ-FR-03	The system should allow users, service providers, and admins to log in and log out.	UC-02	interface-07	seq-03	Class diagram	LoginPage.jsx	AuthController.java	manual test cases.docx
REQ-FR-04	The system should allow users to reset their password via an email link.	UC-02	interface-29	seq-04	Class diagram	AuthContext.jsx	EmailService.java	manual test cases.docx
REQ-FR-05	The system should display the user profile information.	UC-03	interface-20	seq-05	Class diagram	ProfilePage.jsx	UserProfile.java	manual test cases.docx

REQ-FR-06	The system should allow users to edit their name, email, phone number, and profile picture.	UC-03	interface-20	seq-05	Class diagram	ProfilePage.jsx	UserProfile.java	manual test cases.docx
REQ-FR-07	The system should allow users to change their password.	UC-03	interface-20	seq-05	Class diagram	ProfilePage.jsx	UserProfile.java	manual test cases.docx
REQ-FR-08	The system should allow users to delete their account after password confirmation.	UC-03	interface-20	seq-05	Class diagram	ProfilePage.jsx	UserProfile.java	manual test cases.docx
REQ-FR-09	The system should display all available events on the homepage.	UC-04	interface-01	seq-06	Class diagram	HomePage.jsx	EventController.java	manual test cases.docx
REQ-FR-10	The system should allow users to filter events by date, location, price, and event type.	UC-04	interface-01	seq-06	Class diagram	EventsListPage.jsx	EventController.java	manual test cases.docx

REQ-FR-11	The system should display event cards containing title, description, image, date, time, location, and availability.	UC-04	interface-01	seq-06	Class diagram	EventsListPage.jsx	EventController.java	manual test cases.docx
REQ-FR-12	The system should allow users to search events by title or keywords.	UC-04	interface-01	seq-06	Class diagram	EventsListPage.jsx	EventController.java	manual test cases.docx
REQ-FR-13	The system should display available seats for each event.	UC-04	interface-03	seq-06	Class diagram	EventsListPage.jsx	EventController.java	manual test cases.docx
REQ-FR-14	The system should display full event details, including description, images, capacity, and ticket price.	UC-04	interface-03	seq-06	Class diagram	EventsListPage.jsx	EventController.java	manual test cases.docx

REQ-FR-15	The system should display booking availability status for each event.	UC-04	interface-03	seq-06	Class diagram	EventsListPage.jsx	BookingController.java	manual test cases.docx
REQ-FR-16	The system should display vendor information associated with the event.	UC-04	interface-03	seq-06	Class diagram	EventsListPage.jsx	ProviderController.java	manual test cases.docx
REQ-FR-17	The system should prevent booking when event capacity is full.	UC-04	interface-25	seq-06	Class diagram	BookingFormPage.jsx	BookingController.java	manual test cases.doc
REQ-FR-18	The system should require users to log in before completing a booking.	UC-04	interface-24	seq-06	Class diagram	BookingFormPage.jsx	BookingController.java	manual test cases.docx
REQ-FR-19	The system should generate a unique booking reference number upon confirmation.	UC-04	interface-04	seq-06	Class diagram	BookingConfirmationPage.jsx	BookingController.java	manual test cases.docx
REQ-FR-20	The system should immediately reduce available capacity after booking.	UC-04	interface-19	seq-06	Class diagram	BookingConfirmationPage.jsx	BookingController.java	manual test cases.docx
REQ-FR-21	The system should send a booking confirmation email with the reference number.	UC-06	interface-04	seq-06	Class diagram	BookingConfirmationPage.jsx	EmailService.java	manual test cases.docx
REQ-FR-22	The system should display hosting packages with setup details.	UC-05	interface-05	seq-06	Class diagram	EventsListPage.jsx	EventController.java	manual test cases.docx
REQ-FR-23	The system should allow users to book a hosting package for a specific date or location.	UC-05	interface-05	seq-06	Class diagram	EventsListPage.jsx	BookingController.java	manual test cases.docx

REQ-FR-24	The system should display upcoming bookings for the user.	UC-04	interface-19	seq-06	Class diagram	MyBookingsPage.jsx	BookingController.java	manual test cases.doc
REQ-FR-25	The system should display completed bookings.	UC-04	interface-19	seq-06	Class diagram	MyBookingsPage.jsx	BookingController.java	manual test cases.doc
REQ-FR-26	The system should allow cancellation of confirmed bookings.	UC-04	interface-19	seq-06	Class diagram	MyBookingsPage.jsx	BookingController.java	manual test cases.doc
REQ-FR-27	The system should restore event capacity when a booking is canceled.	UC-04	interface-19	seq-06	Class diagram	MyBookingsPage.jsx	BookingController.java	manual test cases.doc
REQ-FR-28	The system should allow organizers to create new public events.	UC-07	interface-14	seq-09	Class diagram	CreateEventPage.jsx	Event2Controller.java	manual test cases.doc
REQ-FR-29	The system should require mandatory fields for events: title, description, location, date, time, capacity, and ticket price.	UC-04	interface-14	seq-09	Class diagram	CreateEventPage.jsx	Event2Controller.java	manual test cases.doc
REQ-FR-30	The system should allow organizers to edit existing event details.	UC-07	interface-15	seq-10	Class diagram	EditEventPage.jsx	Event2Controller.java	manual test cases.doc
REQ-FR-31	The system should allow organizers to delete events.	UC-07	interface-16	seq-11	Class diagram	OrganizerEventPage.js	EventService.java	manual test cases.doc
REQ-FR-32	The system should allow organizers to enable or disable booking for an event.	UC-07	interface-17	seq-08	Class diagram	OrganizerEventPage.jsx	EventService.java	manual test cases.doc
REQ-FR-33	The system should allow organizers to create new hosting packages.	UC-07	interface-14	seq-09	Class diagram	CreateEventPage.jsx	EventService.java	manual test cases.doc
REQ-FR-34	The system should require mandatory fields for hosting packages: title, description, setup details, location options, and price.	UC-05	interface-14	seq-09	Class diagram	CreateEventPage.jsx	EventService.java	manual test cases.doc

REQ-FR-35	The system should allow defining customizable elements within packages.	UC-05	interface-14	seq-09	Class diagram	CreateEventPage.jsx	Event2Controlleur.java	manual test cases.docx
REQ-FR-36	The system should allow organizers to edit hosting package details.	UC-07	interface-15	seq-10	Class diagram	EditEventPage.jsx	Event2Controlleur.java	manual test cases.docx
REQ-FR-37	The system should allow organizers to delete hosting packages.	UC-07	interface-16	seq-11	Class diagram	OrganizerEventPage.jsx	EventService.java	manual test cases.docx
REQ-FR-38	The system should allow organizers to filter events by status: all, active, canceled, or completed.	UC-07	interface-23	seq-08	Class diagram	OrganizerEventPage.jsx	EventService.java	manual test cases.docx
REQ-FR-39	The system should allow organizers to search events within the dashboard.	UC-07	interface-13	seq-07	Class diagram	OrganizerEventPage.jsx	EventService.java	manual test cases.docx
REQ-FR-40	The system should display event type, city, date, status, bookings, statistics, and revenue in the dashboard.	UC-07	interface-10	seq-07	Class diagram	OrganizerDashboard.jsx	DashboardService.java	manual test cases.docx
REQ-FR-41	The system should display total bookings, remaining seats, and total revenue per event.	UC-07	interface-10	seq-07	Class diagram	OrganizerDashboard.jsx	DashboardService.java	manual test cases.docx
REQ-FR-42	The system should display total created events, active events, and total revenue.	UC-07	interface-18	seq-07	Class diagram	OrganizerDashboard.jsx	DashboardService.java	manual test cases.docx
REQ-FR-43	The system should allow registering vendors with business name, service type, city, description, and contact information.	UC-01	interface-09	seq-14	Class diagram	RegisterPage.jsx	ProviderController.java	manual test cases.docx
REQ-FR-44	The system should allow searching vendors by service type or city.	UC-01	interface-12	seq-01	Class diagram	ProvidersDirectoryPage.jsx	ProviderController.java	manual test cases.docx
REQ-FR-45	The system should provider vendor service categories such as catering, photography, venues, entertainment, decoration, audio/visual, and transportation.	UC-08	interface-09	seq-13	Class diagram	EventsListPage.jsx	MarketplaceController.java	manual test cases.docx

REQ-FR-46	The system should send email notifications upon registration (success or failure).	UC-06	interface-29	seq-02	Class diagram	RegisterPage.jsx	EmailService.java	manual test cases.doc
REQ-FR-47	The system should send password reset links and confirmation emails.	UC-06	interface-29	seq-04	Class diagram	ForgotPasswordPage.jsx	EmailService.java	manual test cases.doc
REQ-FR-48	The system should send booking notifications for confirmation and cancellation.	UC-06	interface-04	seq-06	Class diagram	BookingConfirmationPage.jsx	EmailService.java	manual test cases.doc
REQ-FR-50	The system should allow guests to browse all events without registration.	UC-01		seq-01	Class diagram	EventsListPage.jsx	Controller	manual test cases.doc
REQ-FR-51	The system should allow guests to search and filter events.	UC-01		seq-01	Class diagram	ProvideRSVPController	Controller	manual test cases.doc
REQ-FR-52	The system should display full event details to guests.	UC-01		seq-01	Class diagram	EventsListPage.jsx	Controller	manual test cases.doc
REQ-FR-53	The system should prevent guests from booking without logging in.	UC-01		seq-01	Class diagram	EventsListPage.jsx	Booking2Controller.java	manual test cases.doc
REQ-FR-55	The system should allow guests to browse the vendor directory.	UC-01		seq-01	Class diagram	ProvideRSVPController.jsx	ProvideRSV.java	manual test cases.doc

REQ-NFR-56	The system should be easy to use and clear, even for non-technical users.							manual test cases.doc X
REQ-NFR-57	The system should load main pages (Home, Events, Booking) within less than 3 seconds.							manual test cases.doc X
REQ-NFR-58	The system should be available at least 99% of the time.							manual test cases.doc X
REQ-NFR-59	The system should protect user data and not share it with any external party.							manual test cases.doc X
REQ-NFR-60	The system should work correctly on modern browsers (Chrome, Firefox, Edge).							manual test cases.doc X
REQ-NFR-61	The system should handle an increasing number of users and events without issues.							manual test cases.doc X
REQ-NFR-62	Any successful booking cannot be lost or duplicated.							manual test cases.doc X
REQ-NFR-63	Notifications should be delivered on time without noticeable delay.							manual test cases.doc X

Table 55: RTM V.4

7.Summary:

In this chapter, we reviewed the user interfaces (UI) that allow users and organizers to interact with the system seamlessly, through screens such as login, event management, package selection, and booking overviews. These interfaces were designed to be intuitive and fluid, providing a comfortable and efficient user experience. We presented a comprehensive overview of the Event and Package Management System by thoroughly explaining four main components: APIs, event management, package customization, and user management, in addition to the user interfaces (UI) that facilitate these interactions.

We began by explaining how to retrieve booking reports and analytics to track event attendance and revenue over time. Then, we moved on to event and package management, where we detailed the processes of adding, updating, deleting, and retrieving data, with a specific focus on calculating costs and managing package bundles for different user tiers. We also provided a detailed table of all API endpoints, which clarifies how the system operates and how its backend services communicate to handle real-time bookings.

The chapter concluded with a series of test cases covering all functional requirements to ensure the system's reliability and stability under various scenarios. It also included a Requirements Traceability Matrix (RTM) that links business requirements, use cases, design architecture, code implementation, and test cases to ensure comprehensive coverage and continuous monitoring of the project's lifecycle.

Overall, this chapter illustrates how the system integrates to provide a complete environment for organizing events and managing service packages, offering secure and user-friendly interfaces that meet the diverse needs of both organizers and clients.

