

**Syrian Arab Republic
Syrian Private University
Faculty of Engineering
Department of Software &
Information Systems Engineering.**



Event Management and Organization

Project(senior-2)- Completed the requirements for obtaining a bachelor's degree in
Informatics Engineering - Software Engineering and Information Systems

Prepared by:

Mohammed Bakr Safi Ibrahim Ahmed Al-Ali

Supervised by:

Eng. Maher Sarem

2026/Jan

شكر وتقدير

يُمثل هذا البحث تنويجاً لمرحلة من التحصيل الأكاديمي والجهد المتواصل في رحاب كلية الهندسة المعلوماتية، حيث سعينا من خلاله إلى تقديم مساهمة علمية جادة ترقد ميدان هندسة البرمجيات ونظم المعلومات، وتواكب التطورات التقنية المتتسارعة في هذا المجال الحيوي.

وإيماناً بأن الإنجاز العلمي لا يكتمل إلا بتقدير الجهد التي رافقته، أتقدم بواфер الشكر وعميق الامتنان إلى إدارة كلية الهندسة المعلوماتية وكوادرها، وأخص بالذكر الأستاذ الدكتور المهندس مهيب النقري، عميد الكلية، لما قدمه من دعم مؤسسي وتوجيهات علمية رصينة كان لها أثر بالغ في بلورة الرؤية البحثية لهذا المشروع.

كما أتوجه ببالغ التقدير والامتنان إلى المهندس ماهر صارم، الذي تفضل بالإشراف على هذا البحث، حيث كانت لملحوظاته المنهجية الدقيقة ومتابعته الحثيثة لمراحل العمل كافة الدور الأساسي في الارتقاء بمستوى البحث وضمان جودته العلمية، فجزيل الشكر له على ما قدمه من وقت وجهد وخبرة.

كما لا يفوتي أن أعرب عن خالص تقديرني لجميع الأساتذة الذين نهلت من علمهم وخبراتهم طوال المسيرة التعليمية، والذين كان لإسهاماتهم المعرفية أثر ملموس في بناء خلفيتي الأكاديمية، ونخص بالذكر:

د. فادي ابراهيم، د. بسمير برهوم، د. وسيم جندي، د. حاتم (د. أكرم مسوح، د. وسام الخطيب، النجدي، د. عدنان عمورة، د. ثراء أصلان، د. كادان جمعة، د. هاني عماشة، د. كريستين زينيه، د. روز محمد، د. منى العسكري، د. ماجدة بكور، د. زيد شحيدة، د. بدر أحمد، م. محمد عثمان، م. رنا أبو حдан، م. أنس عبد العزيز، م. روان الكردي، م. شادي بلدي، م. عبد الرحمن عبد الرحمن، م. خرامي اليوسف، م. مايا الأحمد). رانيا رجب، م. رانيا صندوق،

ختاماً، نضع هذا العمل بين أيدي المختصين والباحثين، آملين أن يشكل لبنة إضافية في صرح البحث العلمي، وأن يسهم في تقديم حلول تقنية ذات قيمة مضافة للمجتمع العلمي والتقي.

Supervisor Certification

I certify that the preparation of the project entitled Healthcare Institution Management Prepared by Ibrahim Ahmed Al-Ali and Mohammed Bakr Safi, was made under my supervision at Department of Software & Information Systems Engineering – Faculty of Computer & information Engineering in partial fulfillment of the Requirements for the degree pf Bachelors of Software and Information Systems Engineering.

Name: Eng. Maher Sarem

Date:

Signature:

Abstract

An electronic system for organizing various events and activities, providing an easy-to-use and customizable online platform for users. This system aims to facilitate the process of finding the appropriate venue for parties, weddings, and various occasions, as well as finding service providers specialized in organizing and coordinating them.

Current systems suffer from several aspects that need improvement. These include difficulties in connecting clients with event management and organization providers efficiently, providing accurate information to customers, and failing to meet user needs. Furthermore, existing platforms lack the ability to allow users to select the services they desire or request services from multiple locations. This project offers a comprehensive solution to address these challenges and achieve the desired goals. This project aims to improve the user experience when booking venues for various events and facilitate access to the various services user's desire. It also seeks to improve the efficiency of offerings and increase competition, allowing users to access a wide range of offers that suit their needs and financial capabilities. The project also provides a platform for service providers and venue owners to showcase their services and publications, enabling them to reach a wider segment of users and increase their chances of acquiring customers.

This project aims to facilitate quick and easy booking of event venues and provide a variety of services to meet users' needs. The project includes various tools that facilitate communication between users, service providers, and venue owners. Additionally, users participate in the process of managing and organizing events to meet their needs and can express their opinions and share their ratings of the venues and services they engage with.

Table of Contents

LIST OF TABLES.....	VII
LIST OF FIGURES.....	VIII
LIST OF ABBREVIATIONS.....	IX

Chapter One - Introduction

1. Introduction.....	17
2. The problem	17
3. goal of the project.....	18
4. The proposed system.....	19
5. Organizing the report.....	19
6. Conclusion	20

Chapter Two - Basic Concepts and Reference Study

1. Introduction	22
2. Basic Concepts	22
3. Reference Study	23
4. Importance of the project	25
5. The proposed system	26
6. Conclusion	27

Chapter Three - Analytical Study of the Proposed System

1. Introduction	29
2. Project Charter.....	29
3. Statement of Work (SOW).....	31
4. Project Plan - Gantt Char.....	38
5. Risk Management.....	39
6. Version Control and Configuration Management.....	42

7. Conclusion.....	42
---------------------------	-----------

Chapter Four - Analytical Study of the Proposed System

1. Introduction	44
2. Feasibility study.....	44
3. System Requirements Specification (SRS).....	46
4. Requirements Modeling.....	53
5. Unit Testing.....	159
6. Requirements Traceability Matrix.....	164
7. System Architecture.....	167
8. Conclusion.....	168

Chapter Five - System Design Study

1. Introduction	171
2. Design Models (Logical Level).....	171
3. Design Models (Physical Level).....	173
4. Diagrams	174
5. Algorithms Used	175
6. Updated Unit Testing	176
7. Updated Requirements Traceability Matrix	183
8. Conclusion	189

Chapter Six - Practical Application

1. data acquisition strategy	191
2. data persistence layer	191
3. Data Preprocessing & Transformation.....	191
4. Model Training & Algorithm.....	192
5. Backend Integration	192

6. Data Delivery & API	193
---	------------

Chapter seven - Practical Application

1. Introduction	195
2. Tools used	195
3. Miniature Models	196
4. Test Execution.....	200
5. Updating the Requirements Traceability Matrix.....	206
6. Conclusion.....	213

Chapter eight - Conclusions and Future Prospects

1. Introduction	215
2. Conclusions	217
3. Future Prospects	218
4. The reviewer.....	219

LIST OF TABLES

Table 2.1 Comparative table of the reference study.....	57
Table 3.1 Modified Requirements.....	57
Table 3.2 Milestones.....	58
Table 3.3 Risk Register.....	58
Table 4.1 Recommendation & immediate next steps.....	58
Table 4.2 Scenario Comparison.....	59
Table 4.4 login description.....	59
Table 4.5 register (User Registration) description.....	60
Table 4.6 Register Logout description.....	60
Table 4.7 view venues description.....	60
Table 4.8 view venues details description.....	61
Table 4.9 view bookings description.....	61
Table 4.10 add booking description.....	61
Table 4.11 edit booking description.....	62
Table 4.12 delete booking description.....	62
Table 4.13 view events description.....	63
Table 4.14 view events details description.....	63
Table 4.15 unified search for venues description.....	63
Table 4.16 unified search for events description.....	64
Table 4.18 filter venues by specific criteria description.....	64
Table 4.18 filter events by specific criteria description.....	64
Table 4.19 view registration description.....	65
Table 4.20 add registration description.....	65
Table 4.21 edit registration description.....	65
Table 4.22 delete registration description.....	66
Table 4.23 view venue ratings description.....	66

Table 4.24 add venue rating description.....	67
Table 4.25 edit venue rating description.....	67
Table 4.26 delete venue rating description.....	67
Table 4.27 view event ratings description.....	68
Table 4.28 add event rating description.....	68
Table 4.29 edit event rating description.....	68
Table 4.30 delete event rating description.....	69
Table 4.31 view organizers description.....	69
Table 4.32 view providers description.....	70
Table 4.34 filter providers by specific criteria description.....	70
Table 4.33 filter Organizers by specific criteria description.....	71
Table 4.35 upvote description.....	71
Table 4.36 downvote description.....	71
Table 4.37 view recent activities description.....	72
Table 4.38 display map of event and venue locations description.....	72
Table 4.39 browse venues (provider) description.....	72
Table 4.40 brows archived venues description.....	73
Table 4.41 add venue description.....	73
Table 4.42 edit venue description.....	73
Table 4.43 delete venue description.....	74
Table 4.44 archive venue description.....	74
Table 4.45 unarchive venue description.....	74
Table 4.47 accept booking description.....	75
Table 4.48 reject booking description.....	75
Table 4.49 cancel booking description.....	76
Table 4.50 browse events (organizer) description.....	76
Table 4.51 browse archived events description.....	76
Table 4.52 create event description.....	77
Table 4.53 edit event description.....	77

Table 4.54 delete event description.....	77
Table 4.55 archive event description.....	78
Table 4.56 unarchive event description.....	78
Table 4.57 browse registrations description.....	79
Table 4.58 accept registration description.....	79
Table 4.59 reject registration description.....	80
Table 4.60 cancel registration description.....	80
Table 4.61 explore registration as excel file description.....	81
Table 4.62 view booking (organizer) description.....	81
Table 4.63 add booking (organizer) description.....	82
Table 4.64 edit booking (organizer) description.....	82
Table 4.65 delete booking (organizer) description.....	83

LIST OF FIGURES

Figure 4.1 login activity diagram.....	84
Figure 4.2 register (User Registration) activity diagram.....	85
Figure 4.3 Register Logout activity diagram.....	85
Figure 4.4 view venues activity diagram.....	86
Figure 4.5 view venues details activity diagram.....	86
Figure 4.6 view bookings activity diagram.....	87
Figure 4.7 add booking activity diagram.....	87
Figure 4.8 edit booking activity diagram.....	88
Figure 4.9 delete booking activity diagram.....	88
Figure 4.10 view events activity diagram.....	89
Figure 4.11 view events details activity diagram.....	89
Figure 4.12 unified search for venues activity diagram.....	90
Figure 4.13 unified search for events activity diagram.....	90
Figure 4.14 filter venues by specific criteria activity diagram.....	91
Figure 4.15 filter events by specific criteria activity diagram.....	91
Figure 4.16 view registration activity diagram.....	92
Figure 4.17 add registration activity diagram.....	92
Figure 4.18 edit registration activity diagram.....	93
Figure 4.19 delete registration activity diagram.....	94
Figure 4.20 view venue ratings activity diagram.....	94
Figure 4.21 add venue rating activity diagram.....	95
Figure 4.22 edit venue rating activity diagram.....	95
Figure 4.23 delete venue rating activity diagram.....	96
Figure 4.24 view event ratings activity diagram.....	96
Figure 4.25 add event rating activity diagram.....	97
Figure 4.26 edit event rating activity diagram.....	97
Figure 4.27 delete event rating activity diagram.....	98
Figure 4.28 view organizers activity diagram.....	98
Figure 4.29 view providers activity diagram.....	99
Figure 4.30 filter providers by specific criteria activity diagram.....	99
Figure 4.31 filter organizers by specific criteria activity diagram.....	100
Figure 4.32 upvote activity diagram.....	100
Figure 4.33 downvote activity diagram.....	101
Figure 4.34 view recent activities activity diagram.....	101
Figure 4.35 display map of event and venue locations activity diagram.....	102
Figure 4.36 browse venues (provider) activity diagram.....	102
Figure 4.37 brows archived venues activity diagram.....	103
Figure 4.38 add venue activity diagram.....	103
Figure 4.39 edit venue activity diagram.....	104
Figure 4.40 delete venue activity diagram.....	104
Figure 4.41 archive venue activity diagram.....	105

Figure 4.42 unarchive venue activity diagram.....	105
Figure 4.43 browse bookings activity diagram.....	106
Figure 4.44 accept booking activity diagram.....	106
Figure 4.45 reject booking activity diagram.....	107
Figure 4.46 cancel booking activity diagram.....	108
Figure 4.47 browse events (organizer) activity diagram.....	108
Figure 4.48 browse archived events activity diagram.....	109
Figure 4.49 create event activity diagram.....	109
Figure 4.50 edit event activity diagram.....	110
Figure 4.51 delete event activity diagram.....	110
Figure 4.52 archive event activity diagram.....	111
Figure 4.53 unarchive event activity diagram.....	111
Figure 4.54 browse registrations activity diagram.....	112
Figure 4.55 accept registration activity diagram.....	112
Figure 4.56 reject registration activity diagram.....	113
Figure 4.57 cancel registration activity diagram.....	114
Figure 4.58 explore registration as excel file activity diagram.....	115
Figure 4.59 view booking (organizer) activity diagram.....	115
Figure 4.60 add booking (organizer) activity diagram.....	116
Figure 4.61 edit booking (organizer) activity diagram.....	116
Figure 4.62 delete booking (organizer) activity diagram.....	117
Figure 4.62 export registration as excel file (organizer) activity diagram.....	117
Figure 4.62 view e-ticket activity diagram.....	118
Figure 4.62 create e-ticket activity diagram.....	118
Figure 4.62 export list of attendees as a pdf activity diagram.....	119
Figure 4.62 view calendar activity diagram.....	119
Figure 4.63 login sequence diagram.....	120
Figure 4.64 register (User Registration) sequence diagram.....	121
Figure 4.65 Register Logout sequence diagram.....	122
Figure 4.66 view venues sequence diagram.....	122
Figure 4.67 view venues details sequence diagram.....	124
Figure 4.68 view bookings sequence diagram.....	125
Figure 4.69 add booking sequence diagram.....	126
Figure 4.70 edit booking sequence diagram.....	127
Figure 4.71 delete booking sequence diagram.....	127
Figure 4.72 view events sequence diagram.....	128
Figure 4.73 view events details sequence diagram.....	128

Figure 4.74 unified search for venues sequence diagram.....	129
Figure 4.75 unified search for events sequence diagram.....	129
Figure 4.76 filter venues by specific criteria sequence diagram.....	130
Figure 4.77 filter events by specific criteria sequence diagram.....	130
Figure 4.78 view registration sequence diagram.....	131
Figure 4.79 add registration sequence diagram.....	131
Figure 4.80 edit registration sequence diagram.....	132
Figure 4.81 delete registration sequence diagram.....	132
Figure 4.82 view venue ratings sequence diagram.....	133
Figure 4.83 add venue rating sequence diagram.....	134
Figure 4.84 edit venue rating sequence diagram.....	134
Figure 4.85 delete venue rating sequence diagram.....	135
Figure 4.86 view event ratings sequence diagram.....	135
Figure 4.87 add event rating sequence diagram.....	136
Figure 4.88 edit event rating sequence diagram.....	136
Figure 4.89 delete event rating sequence diagram.....	137
Figure 4.90 view organizers sequence diagram.....	137
Figure 4.91 view providers sequence diagram.....	138
Figure 4.92 filter providers by specific criteria sequence diagram.....	138
Figure 4.93 filter organizers by specific criteria sequence diagram	139
Figure 4.94 upvote sequence diagram.....	140
Figure 4.95 downvote sequence diagram.....	140
Figure 4.96 view recent activities sequence diagram.....	141
Figure 4.97 display map of event and venue locations sequence diagram.....	141
Figure 4.98 browse venues (provider) sequence diagram.....	142
Figure 4.99 brows archived venues sequence diagram.....	143
Figure 4.100 add venue sequence diagram.....	143
Figure 4.101 edit venue sequence diagram.....	144
Figure 4.102 delete venue sequence diagram.....	144
Figure 4.103 archive venue sequence diagram.....	145
Figure 4.104 unarchive venue sequence diagram.....	146
Figure 4.105 browse bookings sequence diagram.....	146

Figure 4.106 accept booking sequence diagram.....	147
Figure 4.107 reject booking sequence diagram.....	147
Figure 4.108 cancel booking sequence diagram.....	148
Figure 4.109 browse events (organizer) sequence diagram.....	148
Figure 4.110 browse archived events sequence diagram.....	149
Figure 4.111 create event sequence diagram.....	149
Figure 4.112 edit event sequence diagram.....	150
Figure 4.113 delete event sequence diagram.....	150
Figure 4.114 archive event sequence diagram.....	151
Figure 4.115 unarchive event sequence diagram.....	151
Figure 4.116 browse registrations sequence diagram.....	152
Figure 4.117 accept registration sequence diagram.....	152
Figure 4.118 reject registration sequence diagram.....	153
Figure 4.119 cancel registration sequence diagram.....	153
Figure 4.120 explore registration as excel file sequence diagram.....	154
Figure 4.121 view booking (organizer) sequence diagram.....	154
Figure 4.122 add booking (organizer) sequence diagram.....	155
Figure 4.123 edit booking (organizer) sequence diagram.....	155
Figure 4.124 delete booking (organizer) sequence diagram.....	156
Figure 4.62 export registration as excel file (organizer) sequence diagram.....	156
Figure 4.62 view e-ticket sequence diagram.....	157
Figure 4.62 create e-ticket sequence diagram.....	157
Figure 4.62 export list of attendees as a pdf sequence diagram.....	158
Figure 4.62 view calendar sequence diagram.....	158

LIST OF ABBREVIATIONS

Abbreviation	Definition
UI	User Interface
UX	User Experience
API	Application Programming Interface
DB	Database
CRUD	Create Read Update Delete
RSVP	Répondez s'il vous plaît (event response)
CSV	Comma Separated Values
XLSX	Microsoft Excel Open XML Spreadsheet
JWT	JSON Web Token
SSO	Single Sign-On
OAuth	Open Authorization (authorization framework)
SSL	Secure Sockets Layer
TLS	Transport Layer Security
KPI	Key Performance Indicator
SLA	Service Level Agreement
QA	Quality Assurance
UAT	User Acceptance Testing
CI/CD	Continuous Integration / Continuous Deployment
REST	Representational State Transfer
JSON	JavaScript Object Notation
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
SQL	Structured Query Language
CDN	Content Delivery Network
GDPR	General Data Protection Regulation

Chapter One

introduction

1.introduction

In recent years, the event and activity organization industry has witnessed a radical shift toward relying on technology and online platforms. Given the importance of this sector and its growing demand, the idea of providing an electronic system that helps facilitate event organization has begun to attract attention. Through this project, the company aims to provide an online platform that provides an effective and convenient user experience for users to organize events with ease and convenience.

In this project, the company focuses on providing reliable services to meet users' needs when organizing various events and occasions. The electronic system is an effective solution to many of the challenges that direct users in this field face challenges such as finding a venue that fits their needs and budget, dealing with multiple service providers and waiting for responses, and determining the exact dates and details of events.

Through this project, the company aims to provide a convenient and efficient experience for users, helping them organize their events with ease and convenience, in addition to providing various services that help them find the right solutions to meet their needs. With the increasing reliance on technology in all aspects of life.

2.The problem

An event management platform addresses many of the problems and challenges faced by individuals and organizations when planning events. When organizing an event, whether it's a wedding, conference, or business meeting, there are many details that need to be coordinated. It can be difficult to keep track of all the necessary elements and ensure they run smoothly. The platform provides a central interface to manage all aspects of the event from one place, reducing complexity and minimizing errors.

Additionally, planners must communicate with a variety of vendors and suppliers, such as caterers, set designers, and sound and lighting teams. The online platform provides an internal communication system through which messages and notes can be exchanged effectively, ensuring everyone is on the same page.

Budget management can be one of the most challenging aspects of event planning. The platform provides tools for tracking expenses, setting budgets, and comparing offers from different vendors, helping users stay within their budget and avoid unexpected expenses.

Every occasion has its own unique character and specific requirements. The platform provides a comprehensive database of suppliers and services, along with ratings and reviews from previous users, making it easy for users to choose the best one that best suits their needs.

Managing invitations and attendees is another thing that can become cumbersome. The platform enables users to send electronic invitations, track responses, and manage the

attendee list with ease. It is also possible to send Souvenirs and alerts to attendees to ensure actual attendance at the event.

After the event concludes, it's important to evaluate whether the goals were achieved and what can be improved in the future. The platform provides tools for analysis and data collection, helping provide a comprehensive assessment of the event's success and suggestions for future improvement.

Finally, the platform seeks to enhance the user experience when organizing events, through an easy-to-use interface that facilitates navigation between various options and the customization of its services in a simple and efficient manner. In this way, the platform contributes to making the event planning process easier and more efficient, ensuring greater success and providing an enjoyable and stress-free experience for users.

3.the goal

The goal of an event management platform is to provide a comprehensive and integrated solution to facilitate the process of organizing events efficiently and effectively. Organizing events is often a complex and stressful process that requires coordination between multiple parties, managing budgets, and paying attention to both small and large details. Hence, the need for a specialized platform that contributes to facilitating this process.

The platform aims to provide a central interface that enables users to manage all aspects of an event from one place. Users can easily plan events with advanced tools that allow them to track every element of the event, from venue selection and guest list creation to coordination with vendors and suppliers such as caterers, interior designers, and sound and lighting teams. All of this helps reduce complexity and increase efficiency.

One of the main challenges in organizing events is budget management. The platform provides accurate tools for tracking expenses, setting budgets, and comparing offers from different vendors, helping users control costs and stay within budget. The platform also helps avoid unexpected expenses by providing detailed financial reports and analyses.

Additionally, the platform aims to improve communication and coordination among all stakeholders. It provides an internal communication system through which messages and feedback can be exchanged effectively, ensuring everyone is on the same page and reducing the chances of errors or misunderstandings.

The platform also focuses on providing a personalized and unique experience for every occasion. Through a comprehensive database of suppliers and services, users can choose the most appropriate and suitable for their specific needs, while leveraging the ratings and reviews of previous users to ensure quality service.

The invitation and attendance management are another element the platform seeks to improve. By sending electronic invitations, tracking responses, and easily managing the

attendee list, users can ensure that all invitees are informed of event details and receive reminders to ensure physical attendance.

Finally, the platform aims to provide advanced analytical tools to measure event success and offer suggestions for future improvement. By collecting and analyzing data, users can assess whether goals were achieved and what can be improved upon in future events, contributing to greater success and a more enjoyable and stress-free experience for users.

4. Proposed system

This report provides a comprehensive overview of an integrated electronic system designed to efficiently and professionally manage and organize events. The system aims to simplify the planning and organizing process by offering advanced tools and features that help users coordinate all aspects of the event with ease. Whether planning large events such as weddings and conferences or special occasions, the system provides a centralized solution that includes budget management, coordination with suppliers, and sending electronic invitations. By offering an enhanced user experience and analytical tools to measure success, the system contributes to greater efficiency and smoothness in event organization, ensuring that goals are met and providing a pleasant, stress-free experience for users.

The integrated electronic event management system simplifies and enhances the event planning process through a central interface and advanced tools for tracking expenses, coordinating vendors, and managing invitations. The system aims to reduce complexity and increase efficiency, allowing users to organize successful, hassle-free events. Problems in addition, the system provides analytical tools to evaluate success and suggest future improvements, enhancing the user experience and ensuring the achievement of desired goals.

5. Organizing the report

The report consists of six chapters covering the project's work stages:

1. Chapter One: Introduction
2. Chapter Two: Basic Concepts and Reference Study
3. Chapter Three: Analytical Study of the Proposed System
4. Chapter Four: Design Study of the Proposed System
5. Chapter Five: Practical Application
6. Chapter Six: Response and Future Prospects

6. Conclusion

The Event Management and Organization System is an integrated electronic system that aims to facilitate and simplify the process of planning and organizing various events. The system provides advanced analytical tools for tracking expenses, evaluating event success, and suggesting future improvements. This helps achieve event goals and provides a pleasant, stress-free experience for users. Through this system, users have access to a comprehensive database of suppliers and services, complete with ratings and reviews from previous users, making it easier to choose the best fit for their specific needs.

Chapter Two

Basic concepts and reference study

1.introduction

This chapter addresses the basic concepts related to the event management and organization system, explaining the theoretical aspects upon which the project is based. At the beginning of this chapter, we review the basic concepts that form the foundation of this system, including the importance of good planning and organization in the success of events, and advanced event management tools and techniques that contribute to simplifying the process and achieving the best results. Additionally, this chapter includes a detailed benchmark study of a website similar to our project, called "Mnasabati" This study examines the similarities and differences between the "Mnasabati" website and the system we are developing. We evaluate the strengths and weaknesses of each project, highlighting the competitive advantages of our system. Through this benchmark study, we seek to understand the lessons learned from the "Mnasabati" website and how to apply them to better develop our system. The focus is on elements that could be improved or enhanced in our system, based on the successes and challenges faced by the "Mnasabati" website. This in-depth analysis aims to provide a comprehensive and practical vision for developing a more efficient and successful event management and organization system.

2.Basic concepts

- 1. Event planning and organization:** Planning is setting the goals and initial details of an event, such as determining the location, time, and budget. Organization comes next, where the plan is implemented and all practical aspects of the event are coordinated to ensure it runs smoothly.
- 2. Event management tools and techniques:** Software and applications used to coordinate and organize events include event planning software, tracking tools, and communication apps that help keep different teams updated with real-time details.
- 3. Internal communication system and coordination between suppliers:** This system enables instant communication between event organizers and various suppliers, such as catering, décor, and sound providers. It helps quickly exchange information and resolve urgent issues efficiently.
- 4. Budget Management and Expense Tracking:** This involves monitoring and recording all event-related expenses and ensuring adherence to the allocated budget. Budget management tools provide financial analysis and reports to help control costs.
- 5. Service allocation and supplier evaluation:** This allows us to identify and select the most appropriate services and suppliers for the event based on specific requirements and previous user reviews. This ensures quality service and optimally meets the event's needs.

- 6. Invitation and attendance management:** These tools include sending out electronic invitations, tracking responses, and registering attendees. These tools help ensure all invitees are aware of event details and remind them of the event date to ensure their attendance.
- 7. Analytical tools to evaluate the success of events:** Provide analytics and data on event performance, such as attendee satisfaction and success. These tools help assess whether specific goals were met and what can be improved upon in future events
- 8. Strategies to improve user experience:** Focusing on improving every aspect of the user experience during event planning, we use user-friendly interfaces, customize services based on individual needs, and ensure efficient and seamless communication between all parties. These strategies help provide a pleasant and stress-free experience for users.

3. Reference study

Study my occasions application:

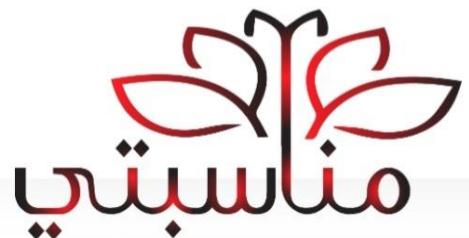
1. Application Information:

- **Application name:** My occasions
- **Application Description:** Application To manage and implement events
- **Application Type: Mobile** (Designed to work on tablets and mobile devices)
- **Target audience:** Customers and service providers (owners of halls and event venues, people who wish to book these venues)
- **App Features/Requirements:**

2. Feature/Requirement Analysis:

- Browse halls and shows
- reservation
- Modify reservation
- Halls and services evaluation
- Contact the hall
- Electronic payment
- Show the hall and its services
- Additional features
 - View hall prices and specifications before booking
 - Choose the type of event
 - Share the location of the hall with others
 - Show lounge availability times
 - Setting a budget

3. User Interface Analysis:



- User Interface Design: Easy to use and simple.
- User Interface Functions: Easy to navigate, efficient.
- User Interface Notes:
 - Improvements: Redesigned user interface to be more modern.
 - New ideas:
 - Integrate the maps feature with the booking feature
 - Allowing requests for additional services from outside the hall
 - Send electronic invitations
 - Allowing owners of restaurants, cafes, farms, and other venues suitable for holding events to display their locations on the website.
 - Allowing logistics service providers to offer their services on the site.
 - Expand the event type options to be more inclusive.
 - Allow visitors to browse the site before subscribing.

4. Performance evaluation:

- **Application speed:** Fast, short page load time.
- **Application stability:** stable.
- **Performance Notes:**
 - **Improvements:** Improve image loading speed.
 - **Problem solver:** Solve the problem of limited payment methods.

5. Strengths and weaknesses analysis:

Table 2.1 Comparison table of the reference study

Attribute	Munasabati (مناسباتي)	Azzam (عزم)	EventApp / Eventap
Primary focus	Digital invitations + marketplace for event services	Professional event management & ticketing	Full-featured event app for conferences and attendee engagement
Core features	Invitation templates; WhatsApp invitations; guest tracking	Event creation; ticket sales; attendance tracking; certificates	Agenda, QR check-in, live Q&A, networking, analytics
Target users	Individuals & small businesses organizing social events	Organizations, conferences, festivals, training providers	Conferences, expos, trade shows, corporate events
Organizer tools	Simple dashboard; packages for guest counts	Advanced reports; multi-attendee tracking; retargeting	Organizer backend; speaker management; sponsor tools
Attendee experience	WhatsApp-based RSVP; mobile-friendly	Ticketing + multiple attendance options	Mobile app with QR check-in, session selection, live interaction
Monetization / Pricing	Tiered packages per guest count (flexible plans)	Subscription / per-event pricing for enterprises	Custom pricing; white-label and add-ons for large events
Notable strengths	Ease of use; fast setup; strong WhatsApp integration	Robust event lifecycle features; trusted by many entities	Rich engagement features (networking, live polls, AR options)

Typical use case	Weddings, small gatherings, digital invitations	Large conferences, ticketed events, training courses	Professional conferences, expos, networking-heavy events
-------------------------	---	--	--

1. Customer needs:

- Easy to book a lounge.
- Electronic payment.
- Contact the hall.
- View lounge and service prices before ordering.
- Know the location of the hall.
- Lounge and service evaluation.
- Choose the type of event.
- Choose the time and date of the reservation.

2. Gym owners' needs:

- Receiving reservation requests.
- Communicate with customers.
- Knowing the type of occasion.
- Knowing the time and date of the reservation and the number of attendees.
- Knowing the required services.

4. Importance of the project

An event management and organization system are of great importance for several reasons, related to facilitating the planning and organization process, achieving efficiency, and ensuring a pleasant experience. Happy and successful for users. Here are some points: Which illustrates the importance of this project:

1. Simplify and improve efficiency: This system makes it easy for users to manage all aspects of the event from one place, reducing complexity and contributing to more efficient organization. Users can track expenses, coordinate with suppliers, and send invitations easily and efficiently.

2. Improving user experience: With an easy-to-use interface and advanced tools, the system provides a seamless and enjoyable experience for users. Services can be customized to suit the needs of each event, increasing customer satisfaction and ensuring the achievement of set goals.

3. Effectively manage the budget: The system provides tools for tracking expenses and setting budgets, helping you control costs and avoid unexpected expenses. You can compare offers from different vendors and choose the best one to ensure you stay within budget.

4. Improve communication and coordination: The system provides an internal communication system that enables users to exchange messages

and notes with suppliers and vendors effectively. This ensures consistency and coordination. efforts, and reduces the chances of errors or misunderstandings.

5. Analytical tools to evaluate success: After the event concludes, analytical tools can be used to assess the extent to which objectives were achieved and provide detailed performance reports. This helps identify strengths and weaknesses and suggest improvements for the future.

6. Increase the chances of success: By facilitating the planning and organizing process and providing advanced tools, the system increases the chances of event success. The system can be relied upon to provide a stress-free and enjoyable experience for users, enhancing the reputation of the events and occasions it organizes.

5. Proposed system

The proposed event management and organization system could include a variety of tools and technologies to improve the planning, organization, and communication process.

1-Components of the proposed system:

- **User-friendly interface:** Intuitive interface design allows users to access all tools easily.
- **Planning and organizing tools:** Includes tools for setting location, time, and budget, as well as schedules and reminders.
- **Invitation and attendance management:** Includes sending electronic invitations, tracking responses, and registering attendance. It may also include options for customizing invitations and sending reminders.
- **Intercom system:** It enables instant communication between event organizers and various suppliers, such as caterers, decorators, and sound.
- **Budget tracking tools:** Monitor and record all event-related expenses and ensure adherence to the allocated budget. It also provides detailed financial reports.
- **Supplier Management:** Includes a system for identifying and selecting suppliers based on specific requirements and previous user reviews.
- **Performance Analytics:** Tools to evaluate event success by collecting and analyzing data to provide detailed reports on event performance.
- **Improved user experience:** It includes user-friendly interfaces, service customization options, and ensures efficient and seamless communication between all parties.

2- Additional features:

- **Mobile applications:** Provides access to the system through smartphones and tablets.
- **Electronic payment:** Allows users to pay fees online in safe and convenient ways.
- **Interactive maps:** Allows users to easily find event locations and share them with others.
- **Multilingual support:** To meet the needs of users from different linguistic backgrounds.
- **Advanced Reports:** Provides advanced analytics and detailed reporting on event performance, helping to improve future operations.

3- Benefits of the proposed system:

- **Increase efficiency:** Thanks to advanced organization and planning, users can save time and effort and achieve efficient event planning.
- **Improving user experience:** With user-friendly interfaces and advanced tools, users can enjoy a smooth and enjoyable experience.
- **Better budget management:** Budget tracking and financial analytics tools allow you to control costs and avoid unexpected expenses.
- **Enhancing communication:** Thanks to the intercom system, users can coordinate effectively with suppliers and vendors.

The proposed system aims to provide a comprehensive and integrated solution for managing and organizing events, ensuring the achievement of specific objectives and increasing the chances of success.

6.Conclusion

The proposed event management and organization system offers a comprehensive and integrated solution to improve the planning and organization process. It features an easy-to-use user interface and advanced tools for managing invitations and attendees, tracking budgets, and communicating instantly with suppliers. The system also provides analytical tools to evaluate event success and detailed financial reports. By providing a smooth and enjoyable user experience, the proposed system helps improve efficiency, better manage budgets, and enhance communication and coordination among all parties. In summary, the proposed system aims to provide an effective and comprehensive solution that ensures the success of events and the achievement of specific objectives.

Chapter Three

Project Management

1. Introduction

This chapter addresses the administrative and organizational aspects of the **AI Student Assistant (Moein)** project and aims to provide a clear framework for managing project execution from initiation through delivery. It focuses on documenting the project's formal authorization, defining the scope and objectives, and allocating roles and responsibilities among team members and supervisors, while highlighting the tools and methodologies used to ensure effective workflow.

The chapter includes key elements such as the Project Charter, the Statement of Work (SOW), the project plan with a Gantt chart for tracking deadlines, stakeholder analysis, and risk management with mitigation plans. It also presents version and configuration control policies, branching and merging strategies for Git repositories, and release procedures to ensure code stability and traceability.

Through these documents and plans, the chapter provides mechanisms for performance monitoring and quality assurance and establishes an organized working environment for developing the backend, frontend, and AI services. It serves as an administrative and methodological foundation that links design requirements to practical implementation, paving the way for the following chapters on detailed design, implementation, and testing.

2. Project Charter

1. General Information

- **Project Title:** Event Management and Organization
- **Planned Start Date:** 2025-10-16
- **Planned End Date:** 2026-01-07
- **Project Manager:** Eng. Maher Sarem
- **Sponsoring / Supervising Entity:** Eng. Maher Sarem

2. Executive Summary and Objective

Executive Summary: The project aims to deliver an integrated online platform for event management and organization, enabling clients, venue providers, and event organizers to interact through dedicated interfaces with tools for venue booking, schedule management, and basic reporting.

Objective: Digitize and streamline event organization processes to reduce manual errors and improve the user experience for all stakeholders.

3. Project Scope

- **In Scope:** Development of three main user interfaces and their integrated functions for the three actor types (client, venue provider, event organizer); user registration and

- account management; event management features; preliminary booking workflow; administrative dashboard; basic reporting.
- **Out of Scope:** Physical logistical services (e.g., transportation, on-site equipment, physical supplier management).

4. Deliverables and Acceptance Criteria

- **Expected Deliverables:** Web application with three user interfaces; functional database; core APIs; brief user guide; final project report.
- **Acceptance Criteria:**

5. Core Team and Roles

- **Supervisor / Project Manager:** Eng. Maher Sarem
- **Lead Developer (Coding):** Mohammad Bakr Safi
- **Report, Analysis, and Design Lead:** Ibrahim Ahmed Nouman Al-Ali

6. Schedule and Major Milestones

- **Final Delivery Date:** After 2026-01-07 (final submission and approval).

7. Budget Estimate (Preliminary)

- **Hosting and Cloud Services (development + initial deployment):** \$150 – \$400
- **AI / Third-party API Costs:** \$100 – \$500
- **Tools and Project Management:** \$50 – \$200
- **Contingency:** Add ~10% of the total as a reserve.

Note: Estimates are indicative.

8. Initial Risks and Mitigation Plans

- **Human Resource Risk:** Loss or unavailability of a key team member → *Mitigation:* task redistribution, incremental documentation, backup assignment plan.
- **Scope Creep:** Additional feature requests delaying delivery → *Mitigation:* change control process and supervisor approval for scope changes.
- **Third-party Dependency Risk:** API outages or unexpected costs → *Mitigation:* evaluate alternatives, set usage limits, and test fallback options.
- **Security and Privacy Risk:** Data breach or unauthorized access → *Mitigation:* basic encryption, access control policies, and security testing.

- **Schedule Risk:** Development or review delays → *Mitigation:* weekly progress monitoring and sprint reprioritization.

9. Key Stakeholders

- **Supervisor:** Eng. Maher Sarem
- **Development Team:** Mohammad Bakr Safi; Ibrahim Ahmed Nouman Al-Ali
- **Intended End Users:** Clients, venue providers, event organizers (represented during development and testing).

10. Assumptions and Constraints

- **Assumptions:** Team members remain available throughout the project; continuous internet access and availability of third-party services; regular supervisor feedback and reviews.
- **Constraints:** Limited budget; fixed final delivery date; exclusion of physical logistics from project scope.

11. Signatures and Approval (Proposed Format)

- **Project Manager / Supervisor:** Eng. Maher Sarem
- **Lead Developer:** Mohammad Bakr Safi — Signature
- **Report and Design Lead:** Ibrahim Ahmed Nouman Al-Ali — Signature

3. The SOW document

Project: Event Management and Organization

1. Introduction

This document defines the detailed scope of work for the project, expected deliverables, acceptance criteria, assumptions, and constraints. It serves as an official reference to clarify what will be delivered and how project success will be measured among the development team, supervisor, and stakeholders.

2. Detailed Description of Work

Core Scope:

- Design and develop a comprehensive web platform for event management and organization, supporting three types of users: **Client (Attendee)**, **Venue Provider**, and **Event Organizer**.
- Dedicated user interfaces for each actor, including registration/login, dashboards, and account management.
- Event creation and management system: create events, set schedules, capacity, descriptions, and upload media (images/files).
- Venue booking and preliminary payment system: check availability, reserve venues, simulate or integrate basic payment gateway.
- Venue provider dashboard to manage bookings, approvals, and scheduling.
- Event organizer dashboard to manage attendees, send notifications, and export basic reports.
- Basic notification system (email or in-app alerts).
- Core APIs (CRUD operations, search, filtering).
- Functional database with secure and simplified schema design.
- Basic user documentation and deployment guide.
- Unit testing, integration testing, and acceptance testing before delivery.

Out of Scope:

- Physical logistics (transportation, on-site equipment, supplier management).
- Advanced integrations with multiple payment gateways unless explicitly agreed.
- Multi-language support or advanced customization features beyond the agreed requirements.

3. Deliverables

- Functional web application with three user interfaces (Client, Venue Provider, Event Organizer).
- Structured database and ER diagram.
- RESTful APIs for core operations.
- Basic user guide and deployment documentation.
- Final project report including design, testing, and validation results.
- Test package: unit test cases, integration tests, and test results log.
- Deployable version in a test/temporary hosting environment.

4. Acceptance Criteria

- **Functional:** All core features operate as specified (login/registration, event creation, venue booking, booking management, notifications).
- **Usability:** Interfaces are navigable and user-friendly.
- **Performance:** Page load times within acceptable limits (e.g., dashboard loads \leq 3 seconds under typical test conditions).
- **Security:** Authentication and authorization implemented; passwords stored securely (encrypted).
- **Testing:** $\geq 90\%$ of defined test cases pass successfully.
- **Documentation:** User guide and deployment documentation included with delivery.

Note: Acceptance criteria can be further detailed with functional and non-functional requirements upon supervisor approval.

5. Assumptions

- Team members remain available throughout the development cycle.
- Stable internet connectivity and access to hosting services for testing.
- Availability of representative test data for booking and event scenarios.
- Supervisor provides periodic reviews and feedback within agreed timeframes.
- Payment simulation or basic gateway integration is feasible during development.

6. Constraints

- Limited budget and resources may restrict third-party integrations.
- Fixed final delivery date (2026-01-07) unless formally extended.
- Physical logistics excluded from project scope.
- Dependence on third-party services (APIs, cloud hosting) may impose limitations or costs.

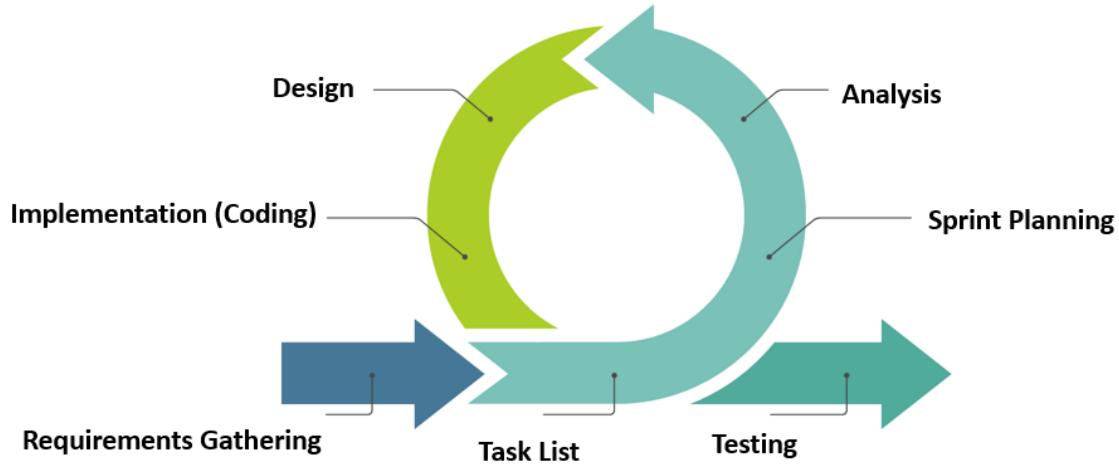
7. Success Measurement

- Delivery of all defined functional outputs within project scope and meeting acceptance criteria.
- Successful completion of unit and integration testing, with supervisor approval of final delivery.
- Stakeholder satisfaction (supervisor and development team) with documentation quality and functional outcomes.

4. Project plan - Gantt Chart

Work is organized according to the Scrum methodology across five short sprints, with regular ceremonies including Sprint Planning, daily standups, Sprint Reviews, and Retrospectives. Results from each Sprint Review are used to update priorities in the Product Backlog and to adjust the Gantt chart as necessary, ensuring execution flexibility and gradual improvement of product quality.

SCRUM METHODOLOGY



Technical reasons

- **Stack fit:** Django + DRF enables fast, secure API development, and React provides a reusable interactive frontend, accelerating MVP delivery.
- **Easy integrations:** Relying on Stripe Sandbox, Leaflet/OSM, and SMTP simplifies connecting external services without early architectural complexity.
- **Migration capability:** Starting with SQLite for development and migrating to PostgreSQL for production reduces initial complexity while preserving future scalability.
- **Early testing of critical features:** Implementing webhooks, QR, and payment flows in early sprints uncovers technical issues sooner and lowers later fix costs.

Business and methodological reasons

- **Fast demand validation:** A sprint-organized MVP allows a quick pilot (10–20 providers, 30–50 bookings) to measure demand before major investment.
- **Revenue model flexibility:** The MVP structure supports experimenting with transaction commissions, premium subscriptions, and paid listings without major structural changes.
- **Market risk reduction:** Local pilots and differentiators (multi-vendor booking, simplified booking flow) help position the product against established competitors.

User experience reasons

- **Improve conversion rate:** A multi-step booking flow with QR generation and smooth payment reduces abandonment and increases conversions.
- **Value for providers and organizers:** Dashboards, reports, and CSV export give operational tools that encourage onboarding and retention.
- **User trust:** Notifications, email confirmations, and QR tickets increase credibility and attendee confidence.

Operational and managerial reasons

- **Simple experimental operations:** Dev/staging environments and lightweight CI enable frequent, safe releases and simple A/B testing.
- **Risk management:** Sprint-based delivery with DoD and integration tests reduces payment and compliance risks and ensures webhook logs for auditing.
- **Automation readiness:** Adding Celery for background tasks (notifications, scheduled reports) later is possible without major redesign.

Academic and graduation report reasons

- **Rich technical documentation:** The stack, ER diagram, API docs, tests, and Gantt/Sprint Backlogs provide comprehensive material for the design and implementation chapters.
- **Clear evaluation criteria:** Measurable acceptance criteria and KPIs (conversion rate, average booking value, API latency) make academic assessment straightforward.

Modified Requirements Table

The following table shows the distribution of core requirements across the five sprints adopted in the Scrum methodology. Each row indicates the targeted user group and the main requirements to be implemented during that sprint. This breakdown is used to ensure the team focuses on a clear scope in each development cycle and to facilitate progress measurement and priority review at the end of each sprint.

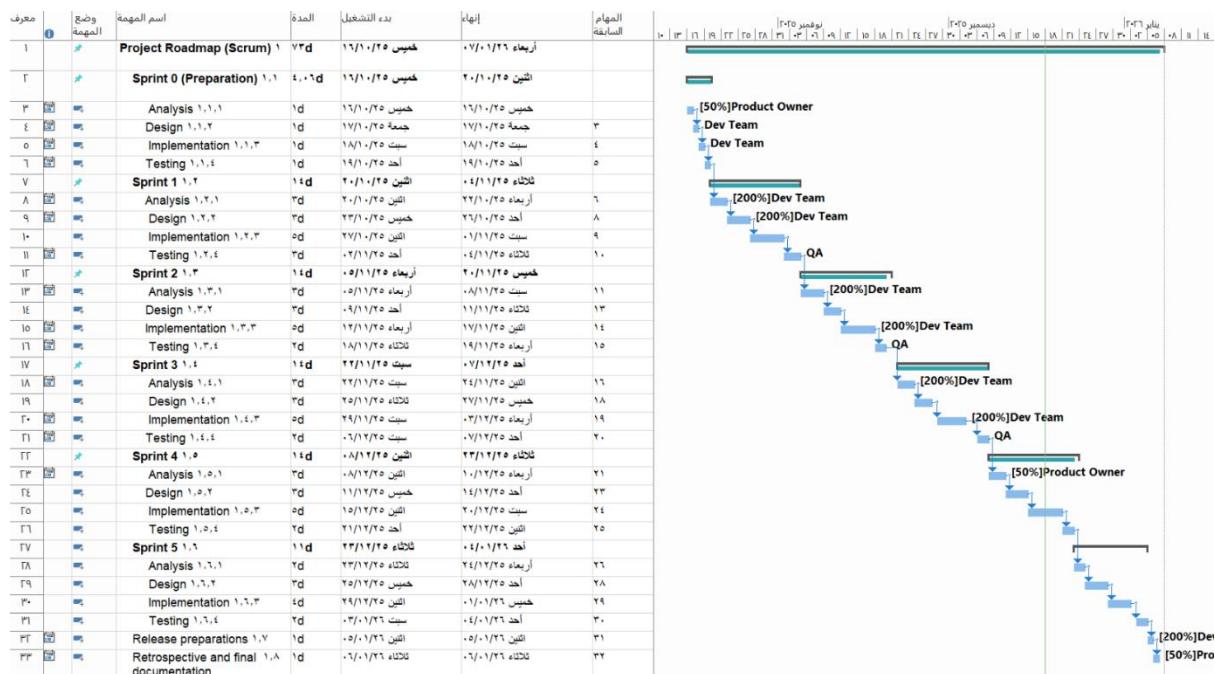
Table 3.1 Modified Requirements

Sprint	Group	Main Requirements (features)
Sprint 1	General basics & account management	Login, Logout, Register (User Registration), Password recovery, Manage personal account, Change password, Delete account
Sprint 2	Content discovery & search	Unified search for venues and events, Filter venues and events by specific criteria, Display map of event and venue locations, View venue and event ratings, Browse venues events
Sprint 3	Venue Provider	Browse venues (Provider), Manage venue(add-edit-delete-archive-unarchive), Browse bookings (Provider), Manage

		booking (Provider), Reject-Cancel-Accept, Manage files and images
Sprint 4	Event Organizer	Browse events (Organizer), Manage event(add-edit-delete-archive-unarchive), Manage event tickets (Create/Edit/Delete), Manage registrations, Create/manage attendee evaluation forms, View statistics and reports, Export attendee list, Manage speakers, Send attendee invitations (QR / e-ticket)
Sprint 5	Attendee / Client interaction	View bookings, Manage booking(add-edit-delete), View registrations, Manage registration(add-edit-delete), Manage venue rating, Manage event rating, Upvote, Downvote, View recent activities, Manage notifications

The table defines the content of each sprint in terms of target groups and main requirements, while the Gantt chart places this content in a clear temporal context that shows the sequence of sprints, task dependencies, and delivery milestones. In other words, the table is used to specify *what* will be implemented in each development cycle, and the Gantt chart is used to specify *when*, who will do it, and how activities overlap; by updating the detailed sprint tables in the appendix and linking them to the timeline in the Gantt chart, the team gains an integrated planning and tracking tool, where progress is measured through Sprint Reviews and dependencies and durations in the Gantt chart are adjusted as needed.

Gantt Chart



The timeline above shows the distribution of work across an initial preparation phase and five consecutive sprints, followed by release preparations and a final review. Each sprint follows an **Analyze → Design → Implement → Test** cycle, and the chart displays temporal

dependencies between tasks and delivery milestones. The text below translates the chart into a practical description suitable for inclusion in the Project Plan chapter.

Sprint summaries and main tasks

- **Sprint 0 — Preparation (12 days)**

Tasks: requirements analysis, preliminary design, basic infrastructure implementation, and preparatory testing.

Deliverables: repository and dev/staging environments ready, ER diagram, simple CI setup, basic authentication.

- **Sprint 1 (14 days)**

Tasks: detailed analysis, UI design, implement CRUD for venues, local media upload, initial tests.

Deliverables: API endpoints and venue UI, basic search engine, initial unit tests.

- **Sprint 2 (14 days)**

Tasks: analysis, design, and implementation of advanced search and filtering, map display, and ratings display.

Deliverables: unified search for venues and events with filters and location map, improved detail pages.

- **Sprint 3 (14 days)**

Tasks: analysis, design, and implementation of venue provider features: venue management, bookings calendar, media management.

Deliverables: working provider dashboard, bookings calendar, accept/reject booking actions, basic statistics.

- **Sprint 4 (14 days)**

Tasks: analysis, design, and implementation of event organizer tools: create events, manage ticket types, export attendees.

Deliverables: multi-step event creation, ticket management, CSV export for attendee lists.

- **Sprint 5 (11 days)**

Tasks: improve booking UX, attendee interfaces, reviews system, and final testing.

Deliverables: final booking flow, attendee interfaces for managing bookings and reviews, notifications.

- **Release preparations (2 days) and final review/documentation (2 days)**

Tasks: final reviews, staging deployment, operational guide and final documentation, conduct Retrospective.

Deliverables: pilot release ready, final test report, improvement notes for the next phase.

Milestones

Table 3.2 Milestones

Milestone	Description
-----------	-------------

End of Sprint 0	Environment ready + ER diagram + basic authentication
End of Sprint 2	Search, filtering, and location map working
End of Sprint 3	Provider dashboard and bookings calendar working
End of Sprint 4	Event creation and ticketing flow + CSV export
End of Sprint 5	Full booking flow + reviews system + pilot release

Suggested acceptance criteria

- **Core functions work:** registration/login, create venue/event, and create booking up to pending state.
- **Payment sandbox integration:** payment flow in Sandbox receives webhook and updates booking to confirmed.
- **QR generation:** QR ticket is generated after booking confirmation and can be verified via an endpoint.
- **Testing:** basic unit tests pass for each model and critical API endpoint.
- **Usability:** multi-step booking UI is clear and validated via a simple user test.

5. Risk Management:

This section outlines the methodology for identifying, assessing, and addressing risks that may affect the success of the **Event Management and Organization** project. The objective is to reduce the likelihood and impact of risks through clear mitigation plans and regular monitoring.

Risk Management Methodology

- **Identify risks:** Gather potential risks from the team, supervisor, and stakeholders during planning sessions.
- **Assess risks:** Classify each risk by **likelihood** (Low/Medium/High) and **impact** (Low/Medium/High) to prioritize responses.
- **Plan mitigation:** Define preventive and responsive actions for prioritized risks.
- **Monitor and follow up:** Review risk status weekly and update mitigation actions and owners.
- **Document:** Maintain a central Risk Register with descriptions, owners, mitigation plans, and status.

Risk Register (proposed)

Table 3.3 Risk Register

Risk ID	Risk Title	Likelihood	Impact	Mitigation / Owner / Status
RK-01	Key team member unavailability	High	High	Redistribute tasks; document work; Owner: Lead Dev; Status: Active
RK-02	Scope creep / feature requests	Medium	Medium	Change control; supervisor approval for scope changes; Owner: PM; Status: Active
RK-03	Third-party API or service outage	Medium	High	Identify alternatives; set usage limits; Owner: Lead Dev; Status: Under Mitigation
RK-04	Security or data privacy incident	Low	High	Implement access controls; encrypt sensitive data; Owner: PM; Status: Active
RK-05	Schedule delays in development	Medium	High	Weekly progress reviews; re-prioritize sprints; Owner: PM; Status: Active
RK-06	Budget overrun	Low	Medium	Monitor expenses; maintain contingency reserve (~10%); Owner: PM; Status: Active

Detailed Mitigation Plans (examples)

- **Key team member unavailability:** Maintain incremental documentation, cross-train team members, and assign temporary backups to reduce single-person dependency.
- **Schedule slippage:** Break work into shorter sprints, defer non-critical features, and allocate buffer days for testing before milestones.
- **Third-party integration issues:** Early testing of alternatives, use mock interfaces during development, and set consumption limits to avoid unexpected costs.
- **Security risks:** Implement basic authentication and authorization, store passwords securely (hashed), and perform a simple security check before delivery.

Monitoring and Reporting

- **Follow-up frequency:** Weekly risk review meetings to update the Risk Register and mitigation statuses.
- **Key indicators:** Number of active risks, percentage of mitigated risks, and schedule impact measured in days delayed.
- **Escalation:** Any high-impact risk that cannot be mitigated is escalated immediately to the supervisor for administrative decisions or additional resource allocation.

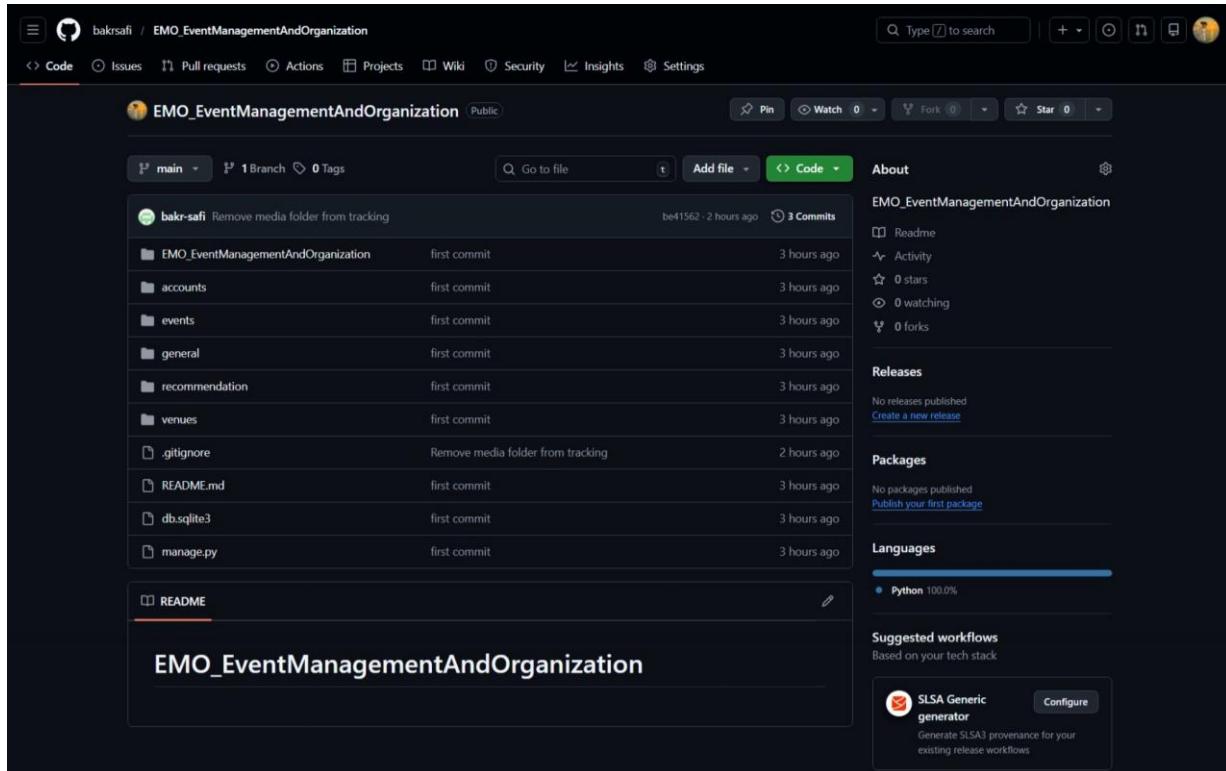
6. Version Control and Configuration Management

6.1 Project description

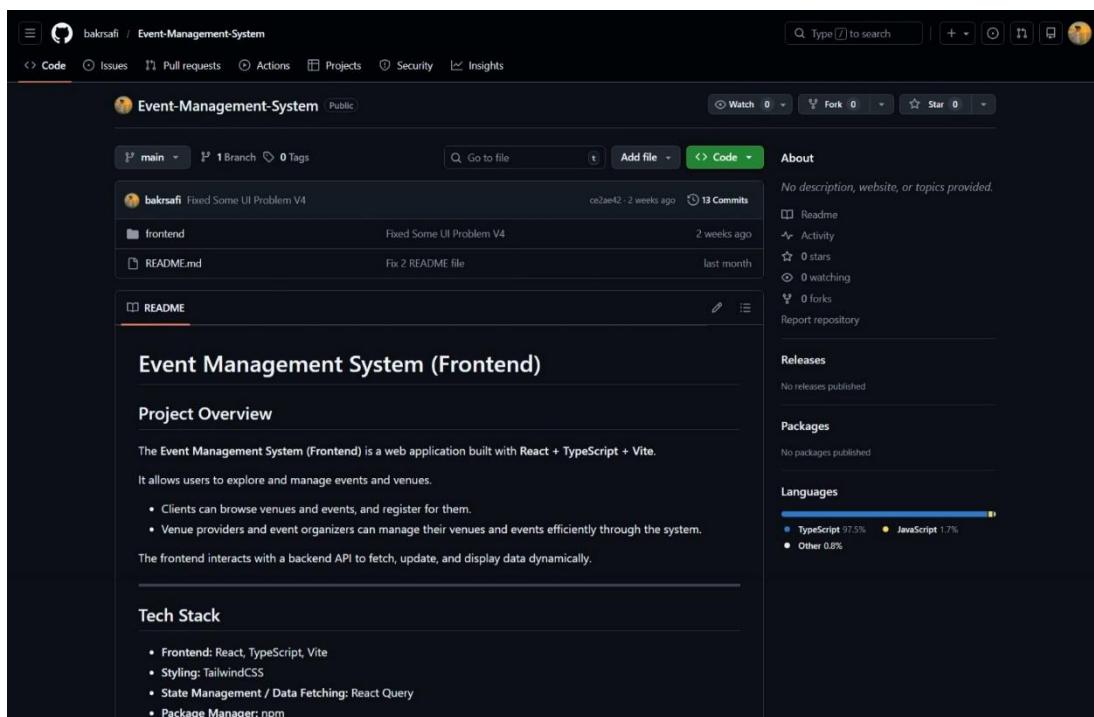
EventM uses a three-tier architecture with a React frontend, Django REST backend, and a development SQLite database. The recommendation model is trained offline and served via a backend endpoint. Repositories are organized to separate frontend and backend concerns for independent development and deployment.

6.2 Git repository structure

- **Backend repository:** junior-project-eventm-backend — Django application, API, model serving code.



- **Frontend repository:** junior-project-eventm-frontend — React application, UI components and client integration.



Each repository contains logical directories for features, tests, configuration files and deployment scripts (where applicable).

6.3 Branching and merging strategy

Adopt a simple, consistent flow across repositories:

- production stable release branch.
- Development integration branch for active development.
- Feature short-lived branches for discrete features or fixes.

Workflow:

1. Create feature/* from development.
2. Implement and test locally.
3. Open a Pull Request for code review.
4. Merge into development after approval and successful checks.
5. Merge development into production for stable release and tag.

6.4 Team responsibilities

- Backend team: implement REST APIs, integrate the NMF model, manage data schema and migrations.
- Frontend team: implement React UI, integrate with backend APIs, build dashboards.
- AI contributor: train and evaluate the NMF model, provide serialized model artifacts for serving.

All merges should use pull requests for code review and traceability.

6.5 Development workflow

1. Identify task and create feature/* branch from development.
2. Implement feature and run local tests.
3. Open Pull Request and request a peer review.
4. After approval, merge to development.
5. When the development branch reaches a stable milestone, merge to production and tag a release.

6.6 Tags and versioning

Planned release tags:

- v1 — Authentication and role system.
- v2 — Client functionality (search, booking, registrations).
- v3 — Provider functionality (venue management, booking approvals).
- v4 — Organizer functionality and recommender integration.

6.7 Releases management

Releases will be created on GitHub with accompanying release notes describing included features and a Postman collection demonstrating tested endpoints. Releases serve as the official submission artifacts for academic evaluation.

7. Conclusion

The Project Management chapter establishes the governance, planning, and control mechanisms necessary to deliver the **Event Management and Organization** platform successfully. By defining the project charter, a detailed Statement of Work, a structured project plan with milestones, a risk register with mitigation measures, and a clear version-control and configuration strategy, the team gains a shared roadmap and accountability framework that reduces ambiguity and supports coordinated execution.

Key outcomes of this chapter include a clarified project scope that separates digital deliverables from physical logistics, a timeline that sequences analysis, design, development, testing, and delivery, and a set of operational practices (branching strategy, CI/CD pipeline, and backup/rollback procedures) that promote code stability and repeatable releases. The documented risk-management approach ensures that likely threats—such as resource unavailability, scope creep, third-party dependencies, and schedule slippage—are monitored and mitigated proactively.

With these management artifacts in place, the project is positioned to move into detailed design and implementation with controlled change processes, measurable acceptance criteria, and regular supervisory checkpoints. Continued adherence to the plans and timely communication among stakeholders will be essential to meet the final delivery date and achieve the project's objective of providing a reliable, user-centered event management platform.

Chapter four

Analytical study

1.introduction

This chapter presents a rigorous analytical foundation for the event management system by translating high-level goals into actionable plans and verifiable specifications. It frames the project's feasibility across technical, economic, and operational dimensions, defines the scope and deliverables, and establishes measurable criteria for success. The chapter details the System Requirements Specification (SRS), models the functional and non-functional requirements, and maps use cases to data and process flows. It also describes the project plan and timeline, outlines unit testing strategies, and provides a Requirements Traceability Matrix to ensure every requirement is implemented and validated. Finally, the chapter concludes with a clear system architecture that aligns design choices with identified risks and mitigation strategies, setting the stage for the subsequent design and implementation phases.

2.Feasibility study

Quick decision guide (key considerations & questions)

- **Target first:** Organizers and venue providers or end attendees? Prioritize the group that will pay (organizers/providers).
- **Payment gateway:** Stripe (international) or a local gateway—choose based on target market support.
- **Hosting plan:** Start with low-cost cloud (Heroku/Vercel/AWS free tier) and plan for S3 + PostgreSQL at scale.
Decision points: MVP scope (search, venue pages, booking/tickets, organizer dashboard); go/no-go metrics (MVP ready in 8 weeks; 30 pilot bookings).

Market & opportunity

- **Growing market:** The event-management platform market is expanding rapidly with **double-digit CAGR** and strong demand for cloud and hybrid event tools.
- **Addressable niche:** Focus on **multi-vendor booking** (ability to add services from multiple providers in one booking) and **transparent price comparison** to differentiate from incumbents.

Technical feasibility

- **Stack:** Django + Django REST Framework (backend), React (frontend), SQLite for development, **PostgreSQL for production**.
- **Core integrations:** Payment (Stripe sandbox), maps (Leaflet/OpenStreetMap), email service (SMTP/SendGrid), optional Celery+Redis for background tasks.

- **Security:** JWT or HTTP Only cookies, HTTPS, input sanitization, webhook signature verification.

Financial snapshot (estimates)

- **Initial development:** **\$3,000–\$10,000** (developer time, domain, basic hosting, Stripe fees) for a functional MVP.
- **Monthly ops:** **\$50–\$300** (hosting, storage, email/SMS).
- **Revenue model:** commission per booking (5–10%), premium organizer subscriptions, promoted listings.

Risks, limitations, and mitigations

- **Technical risk — SQLite in production:** SQLite limits concurrency; **mitigation:** plan and test migration to PostgreSQL before public launch.
- **Market risk — competition & adoption:** incumbents and SaaS suites dominate; **mitigation:** launch focused MVP with unique multi-vendor booking and aggressive pilot outreach.
- **Payment & legal risk:** compliance with gateway rules and refund policies; **mitigation:** use sandbox testing, document clear T&Cs, and log transactions for audits.
- **Operational risk — vendor onboarding:** low supply of quality venues; **mitigation:** offer onboarding incentives and simple provider dashboards.

Recommendation & immediate next steps

1. **Build MVP (8–12 weeks):** search/venue pages, booking/tickets, organizer/provider dashboards, QR e-ticket generation.
2. **Pilot:** onboard **10–20 venues** and run **30–50 pilot bookings** to validate UX and pricing.
3. **Prepare production migration:** configure PostgreSQL, S3 storage, and HTTPS; finalize payment integration.
4. **Measure:** track conversion rate, average booking value, and churn; iterate based on pilot feedback.

Table 4.1 Recommendation & immediate next steps

Scenario	Initial cost	Time to MVP	Scalability
Low budget	\$1k–3k	6–8 weeks	Limited
Medium (recommended)	\$3k–10k	8–12 weeks	Good (with migration)
High scale	\$10k+	12+ weeks	High

Implementation Plan and Timeline

- **Phase 0 (Week 0):** Define MVP scope; set up repositories and development environments.
- **Phase 1 (Weeks 1–3):** Data models, authentication, CRUD for venues and events, basic search UI.
- **Phase 2 (Weeks 4–6):** Ticketing and booking flows, organizer/provider dashboards, QR ticket generation, payment integration in sandbox.
- **Phase 3 (Weeks 7–8):** Functional and security testing, pilot launch with 30–50 bookings, collect feedback and iterate.
- **Deliverables:** Working MVP, ER diagram, API documentation, test report, and production migration plan.

Scenario Comparison

Table 4.2 Scenario Comparison

Scenario	Scope	Time to MVP	Scalability	Priority focus
Proof of Concept	Basic booking + QR	6–8 weeks	Low	Attendee UX and booking flow
Local Adoption (recommended)	Search, dashboards, payments sandbox	8–10 weeks	Medium	Balanced across attendees/providers/organizers
Scale Ready	Reports, PostgreSQL, S3, marketing	10–14 weeks	High	Scalability, security, growth

3. Software Requirements Specification (SRS)

Project Title

"Event Management and Organization"

Introduction

This section provides an introductory overview of the Software Requirements Specification (SRS) document for the **Venue and Events Booking System**. The document describes the system's functional and non-functional requirements and expected behavior in a structured,

testable way to serve as a single reference for technical teams and stakeholders throughout the project lifecycle. Its goal is to align expectations and facilitate design, development, and testing.

Problem Definition

the complexity and fragmentation of event and booking management processes: organizers, providers, and attendees struggle to track availability and schedules and to avoid conflicts; duplicate or conflicting bookings occur; verifying ticket validity and preventing fraud is difficult; activity and audit logs lack transparency; aggregating registration and ticket data for reporting and accounting is hard; notifications and confirmations may fail or be delayed; handling large events or peak demand is challenging; and search and filtering interfaces are often complex, making it tiring for users to find suitable events and venues.

Purpose of the Document

The purpose of this document is to record all requirements that the Venue and Events Booking System must satisfy, including use cases, acceptance criteria, and performance and security constraints. The document will be used as a reference by engineers, designers, testers, project managers, and business stakeholders to define scope, estimate effort, and verify the final product against agreed requirements.

System Scope

The system supports management of venues and events for the main user roles: **attendee/client, venue providers, event organizers**. Scope includes: venue and event listing and display, search and filtering, booking and registration management, content management (add/edit/archive venues and events), ratings and reviews, payment interfaces, and integrations with map providers, payment gateways, and notification services.

Functional Requirements

REQ-01 Login — User authentication flow: credential validation, session/token issuance, redirect to appropriate dashboard, and login event logging.

REQ-02 Logout — End user session or revoke token, redirect to login, and log the logout event.

REQ-03 Register (User Registration) — User sign-up flow with field validation, duplicate email check, account creation, confirmation email, and activation granting basic permissions.

REQ-04 View Venues — Display a list of available venues as cards with basic info (name, location, capacity, price, thumbnail, rating) with a default sort and pagination support.

REQ-05 View Venue Details — Show a venue detail page including image gallery, full description, pricing, capacity, availability schedule, and action buttons (book, share, reviews); handle venue-not-found.

REQ-06 View Bookings — Show the user's current and past bookings with booking number, venue, date/time, status, and sorting/filtering options.

REQ-07 Add Booking — Create a new booking via a form with immediate availability checks, conflict prevention, atomic transaction, and confirmation notification.

REQ-08 Edit Booking — Modify allowed fields of an existing booking with revalidation of availability and audit logging of changes.

- REQ-09 Delete Booking** — Cancel or delete a booking after confirmation, applying cancellation policies and logging the reason.
- REQ-10 View Events** — Display a list of events with name, date, location, thumbnail and pagination for large result sets.
- REQ-11 View Event Details** — Show event detail page with time/date, venue, description, ticket options, organizers, registration/purchase links, and map if available.
- REQ-12 Unified Search for Venues** — Unified keyword search across venues, returning categorized results (Venues), supporting partial matches and pagination.
- REQ-13 Unified Search for Events** — Unified keyword search across events, returning categorized results (Events), supporting partial matches and pagination.
- REQ-14 Filter Venues by Specific Criteria** — Multi-criteria filtering (location, price, capacity, type, date) with immediate or applied updates and easy reset.
- REQ-15 Filter Events by Specific Criteria** — Multi-criteria filtering (location, price, capacity, type, date) with immediate or applied updates and easy reset.
- REQ-16 View Registrations** — Show the user's event registrations with attendance status and event details; support pagination.
- REQ-17 Add Registration** — Register a user for an event with availability check, payment processing if required, and ticket/confirmation delivery.
- REQ-18 Edit Registration** — Edit an existing registration within organizer policy limits, with availability checks and confirmation notification.
- REQ-19 Delete Registration** — Cancel a registration after confirmation, showing cancellation implications and updating ticket availability.
- REQ-20 View Venue Ratings** — Display venue reviews and ratings: average, total count, individual reviews (reviewer or anonymous), with sorting/filtering.
- REQ-21 Add Venue Rating** — Submit a numeric rating (1–5) and optional comment, validate eligibility, prevent unauthorized duplicates, and update averages.
- REQ-22 Edit Venue Rating** — Allow the reviewer to edit their rating within an allowed time window and record the edit in the audit trail.
- REQ-23 Delete Venue Rating** — Delete the user's rating after confirmation, update the venue average, and log the deletion.
- REQ-24 View Event Ratings** — Display event ratings and comments, average rating and total count; encourage first rating if none exist.
- REQ-25 Add Event Rating** — Submit an event rating subject to attendance policy, prevent unauthorized duplicates, and optionally notify the organizer.
- REQ-26 Edit Event Rating** — Edit an existing event rating within the allowed timeframe, update averages and record the change.
- REQ-27 Delete Event Rating** — Delete an event rating after confirmation, update averages and log the action.
- REQ-28 View Organizers** — List organizers with name, short bio, number of past events, overall rating, and link to organizer details.
- REQ-29 View Providers** — List venue providers with name, number of venues, overall rating, and links to contact or view their venues.
- REQ-30 Filter Organizers by Specific Criteria** — Filter organizers by city, rating, experience, combine filters and reset them.
- REQ-31 Filter Providers by Specific Criteria** — Filter providers by city, rating, experience, combine filters and reset them.
- REQ-32 Upvote** — Record an upvote on content or a review, increment the UI count immediately, prevent duplicate votes, and log voter identity and timestamp.
- REQ-33 Downvote** — Record a downvote on content or a review, update the UI count immediately, prevent duplicate votes, and log voter identity and timestamp.

REQ-34 View Recent Activities — Show the user's recent activity timeline (bookings, registrations, ratings, votes) with timestamps, links, and filters by type/date.

REQ-35 Display Map of Event and Venue Locations — Interactive map showing venue/event pins with info windows and links; support zoom, clustering, and filtering.

REQ-36 Browse Venues (Provider) — Provider dashboard listing the provider's venues as cards with search, filters, and links to edit each venue.

REQ-37 Browse Archived Venues — List archived venues with archive date/reason and actions to restore or permanently delete per policy.

REQ-38 Add Venue — Provider creates a venue with required fields (name, description, location, capacity, pricing, amenities, images, coordinates), validation, and media handling.

REQ-39 Edit Venue — Provider edits venue data with validation (e.g., schedule conflicts), saves changes, updates public listing, and records audit trail.

REQ-40 Delete Venue — Delete or soft-delete a venue after checking for active bookings/obligations; block deletion if constraints exist and log the action.

REQ-41 Archive Venue — Archive a venue (change status to Archived), hide it from public listings, and record archive metadata.

REQ-42 Unarchive Venue — Restore an archived venue after verifying conditions, reindex it, and log the restoration.

REQ-43 Browse Bookings (Provider) — Provider view of bookings for their venues with filters and action buttons to manage each booking.

REQ-44 Accept Booking (Provider) — Provider accepts a pending booking after rechecking availability, updates status to Accepted, reserves resources, and notifies the client.

REQ-45 Reject Booking (Provider) — Provider rejects a pending booking with optional reason, releases resources, notifies the client, and logs the action.

REQ-46 Cancel Booking (Provider) — Provider cancels an accepted/confirmed booking after confirmation, handles refunds if applicable, notifies the client, and logs the cancellation.

REQ-47 Browse Events (Organizer) — Organizer dashboard listing their events with status (Active/Draft/Archived) and management actions.

REQ-48 Browse Archived Events — List organizer's archived events with archive metadata and options to restore or delete per policy.

REQ-49 Create Event — Organizer creates an event (title, description, dates, sessions, ticket types, speakers, images, coordinates), with venue availability checks and option to save as draft or publish.

REQ-50 Edit Event — Organizer edits event details, system validates impacts (tickets, schedules), saves updates, and notifies affected attendees if needed.

REQ-51 Delete Event — Delete or archive an event after checking for refunds/disputes; block deletion if unresolved obligations exist and notify registrants.

REQ-52 Archive Event — Archive an event, hide it from public listings, and record archive metadata.

REQ-53 Unarchive Event — Restore an archived event after verifying conditions and reindex it.

REQ-54 Browse Registrations (Organizer) — Organizer view of registrations with filters and quick actions to manage each registration.

REQ-55 Accept Registration — Organizer accepts a pending registration after verifying ticket availability, updates status, notifies the registrant, and updates ticket counts.

REQ-56 Reject Registration — Organizer rejects a registration with optional reason, releases tickets, notifies the registrant, and logs the action.

REQ-57 Cancel Registration — Organizer or attendee cancels a registration with confirmation, handles refunds/fees per policy, updates ticket availability, and logs the action.

REQ-58 Export Registrations as Excel file — Generate an XLSX export of registrations

for a selected range/criteria and provide it for download while logging the export.

REQ-59 View Booking (Organizer) — Organizer view of bookings for their events with details, filters, and management actions.

REQ-60 Add Booking (Organizer) — Organizer creates a booking on behalf of a client, validates availability, processes payment if needed, and sends confirmations.

REQ-61 Edit Booking (Organizer) — Organizer edits a booking, system validates conflicts, saves changes, notifies the client, and records the audit trail.

REQ-62 Delete Booking (Organizer) — Organizer deletes or cancels a booking with confirmation, handles refunds if applicable, notifies the client, and logs the operation.

REQ-63 Export Registrations as Excel file — Generate an XLSX export of registrations for a selected range/criteria, provide it for download or queue it, and log the export (user, criteria, timestamp).

REQ-64 View e-ticket — Attendee requests to view an e ticket; system verifies auth and booking, retrieves or generates the ticket (QR/PDF), displays/offers download, notifies on errors, and logs the action.

REQ-65 Create e-ticket — Triggered by booking confirmation or organizer request; system gathers booking data, generates a unique e ticket (ID, QR/barcode, validity, seat), attaches it to the booking, delivers to attendee, and records the generation event.

REQ-66 Export e-tickets list as PDF file — Organizer selects criteria; system validates permissions, compiles matching e tickets, generates a PDF report (or queues large jobs), provides download/notification, and logs the export.

REQ-67 Display an interactive calendar — User opens calendar; system loads events for the chosen scope, supports navigation (day/week/month), filters and search, shows event details and actions (book/cancel/view e ticket), handles empty/offline/restricted states, and logs interactions.

Nonfunctional Requirements

Performance

- **Response time targets**
 - **Search results:** median response $\leq 1 \text{ second}$ for common queries under typical load.
 - **Filter application and list updates:** response $\leq 2 \text{ seconds}$ for medium result sets.
 - **Page load:** First Meaningful Paint for main pages $\leq 1.5\text{--}2.5 \text{ seconds}$ on a typical broadband connection.
- **Concurrent user load**
 - Support **2,000** concurrent active users as a baseline; design to scale horizontally to **10,000+** concurrent users with autoscaling.
- **Pagination and throughput**
 - All endpoints must support pagination and return pages of configurable size (default **20**).
- **Payload size constraints**
 - Respect HTTP payload guidance per use case; detail and map pages may be larger. Lightweight actions (voting, rating, registration) should target small payloads (e.g., $< 1.5 \text{ KB}$ for ticketing actions; $< 15 \text{ KB}$ for rating/registration lists; search/list pages **15–250 KB** depending on media).
- **Performance testing**
 - Provide load and stress test reports demonstrating that targets are met for baseline and scaled scenarios.

Security

- **Authentication**
 - Support secure authentication (email/password) with passwords stored hashed (PBKDF2 with SHA-256) and optional SSO/OAuth2 for providers/organizers.
 - Enforce account lockout and rate limiting on failed attempts.
- **Authorization**
 - Role-based access control (RBAC) ensuring users can access only authorized resources (attendee, provider, organizer, admin).
- **Encryption**
 - All communications over TLS 1.2+ (HTTPS).
 - Encrypt sensitive data at rest (e.g., payment tokens, contact details) using strong algorithms (AES-256).
- **Logging and audit**
 - Maintain immutable audit logs for critical actions (create/edit/delete bookings, payments, rating changes, exports) including actor, timestamp, and reason.
 - Protect logs from tampering and retain them according to the retention policy.
- **Input validation and protection**
 - Server-side validation for all inputs; protect against injection (SQL/NoSQL), XSS, CSRF.
- **Privacy**
 - Mask or obfuscate personal contact data in public views; comply with applicable data protection laws and document retention policies.

Reliability Backup and Recovery

- **Availability**
 - Target **99.9%** availability for core services (excluding scheduled maintenance).
- **Backups and retention**
 - Daily backups for transactional data; weekly full backups for large datasets.
 - Define retention windows (e.g., **30–90 days**) and store backups in a secure offsite location.
- **Recovery objectives**
 - **RTO (Recovery Time Objective): ≤ 2 hours** for critical services.
 - **RPO (Recovery Point Objective): ≤ 15 minutes** for transactional data (bookings/registrations).
- **Fault tolerance**
 - Design for multi-AZ or multi-region deployment where feasible; automatic failover for databases and stateless services.
- **Monitoring and alerting**
 - Implement health checks, metrics (latency, error rates), and alerts for SLA breaches and security incidents.

Maintainability and Extensibility

- **Modular architecture**
 - Separate services (search, bookings, notifications) to enable independent deployment and scaling.
- **Scalability**
 - Stateless application tiers; horizontal scaling for web/API layers; database scaling strategy (read replicas, sharding plan if needed).

- **Code quality and documentation**
 - Provide API documentation (OpenAPI/Swagger) for all public endpoints.
 - Developer documentation: architecture overview, deployment steps, environment variables, and runbooks for common operations.

Usability and Accessibility

- **User interface requirements**
 - Responsive design supporting desktop and mobile; consistent UI patterns for listings, details, and forms.
 - Fast, clear feedback for user actions (success, validation errors, loading states).
- **Accessibility**
 - Conform to **WCAG 2.1 AA**: keyboard navigation, semantic markup, sufficient color contrast, and screen-reader labels.
- **Internationalization**
 - Format dates, times, numbers, and currencies per user locale; display event times clearly with timezone awareness.

Technical Constraints

- **Supported platforms and clients**
 - Web: modern browsers (last two major versions of Chrome, Firefox, Edge, Safari).
 - Mobile: responsive web interface.
- **Datastores and storage**
 - Primary relational database (e.g., **SQLite** or other RDBMS) for transactional data.
 - Search index (e.g., Elasticsearch or managed search service) for unified search and filtering.
 - Object storage (e.g., S3) for media assets with a CDN for delivery.
- **External integrations**
 - **Map provider**: Google Maps or alternative; secure API keys and monitored usage quotas.
- **Rate limits and quotas**
 - Define per-endpoint rate limits (e.g., higher throughput for search; stricter limits for voting endpoints to prevent abuse).
 - Return meaningful HTTP status codes and clear error payloads to clients.
- **HTTP payload size limits**
 - Enforce upload limits for media (e.g., images \leq **5–10 MB** per file); compress responses where appropriate; adhere to the lightweight payload targets in the Performance section.

4. Requirements Modeling

Use Case Diagrams

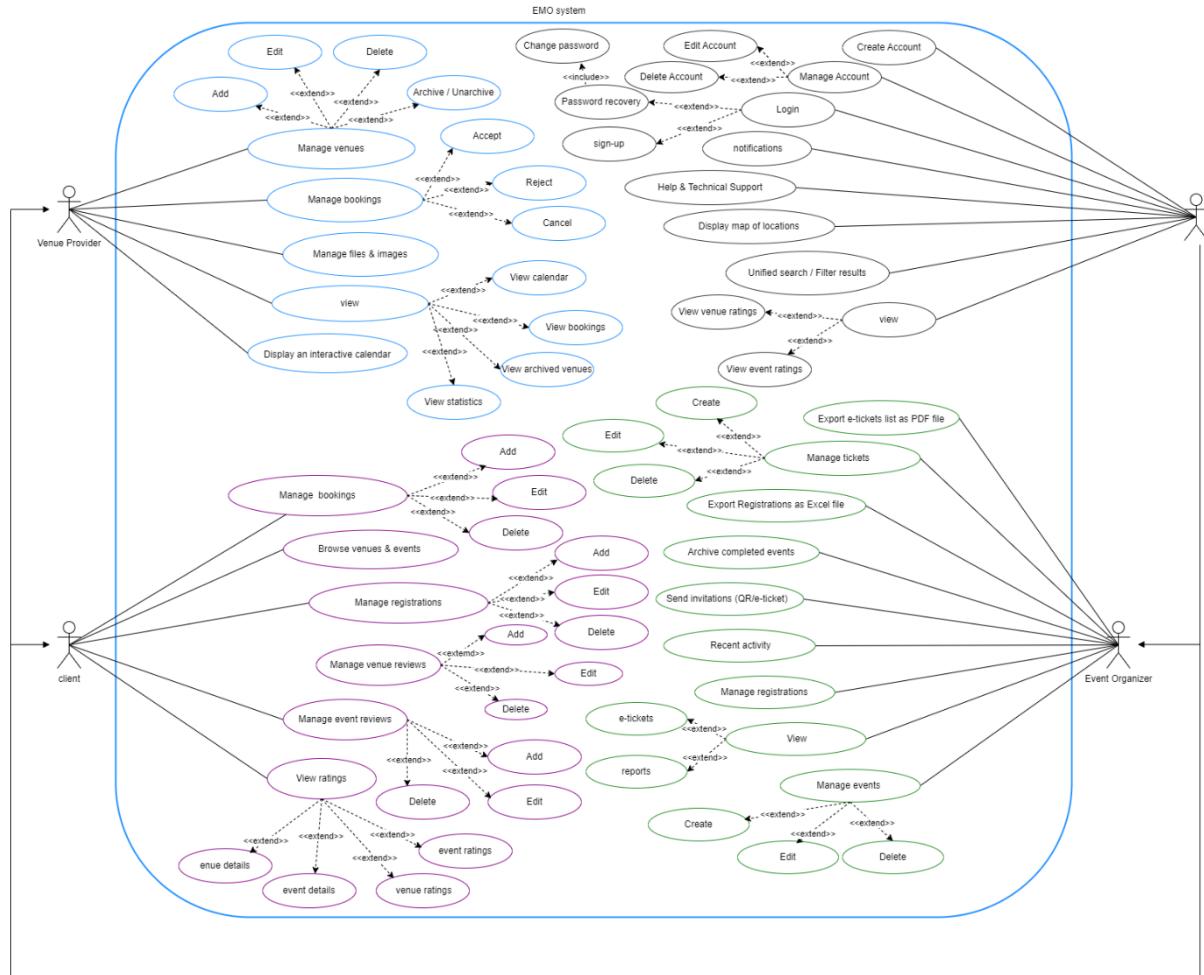


Figure 4.4 general use case diagram

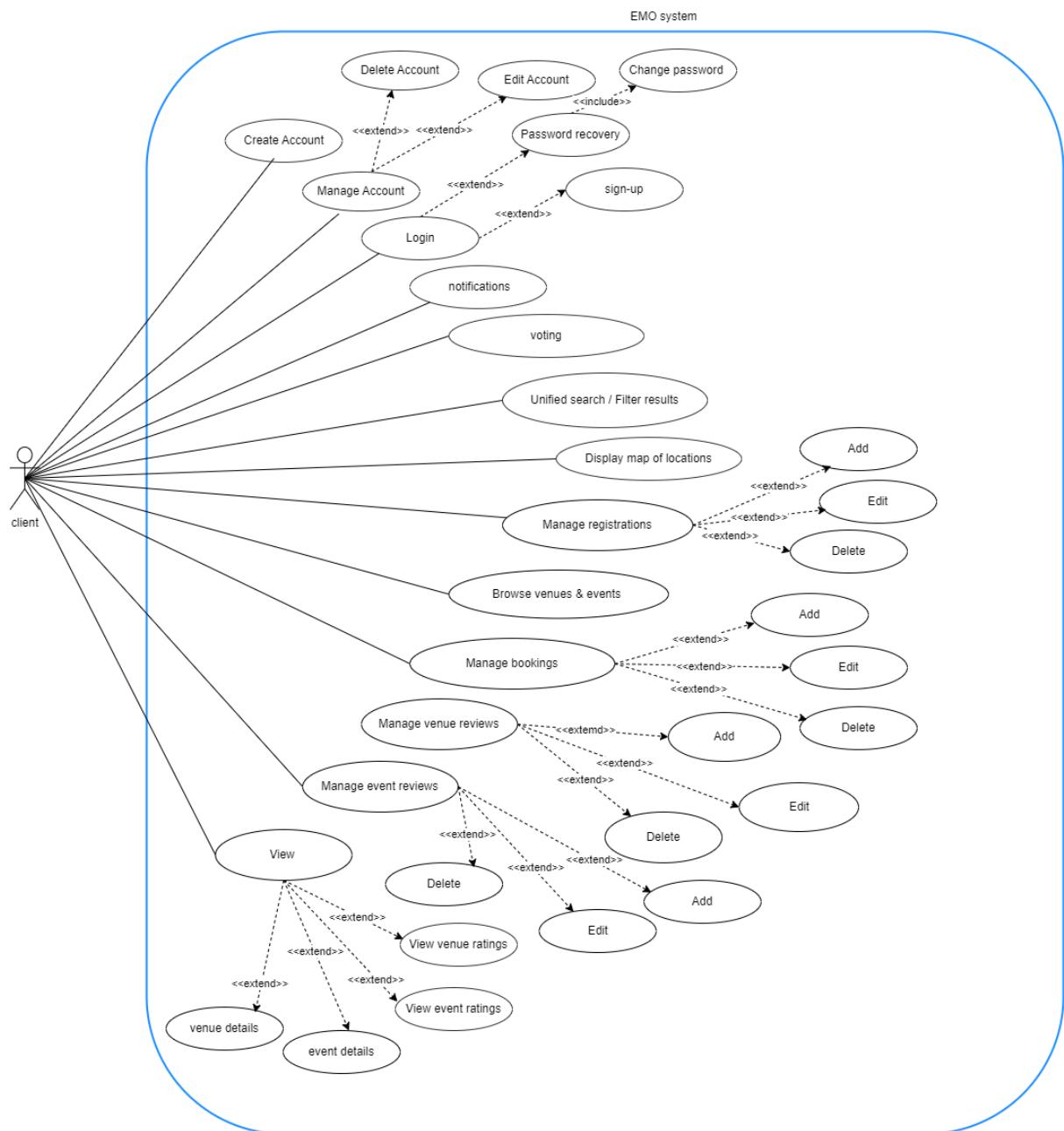


Figure 4.4 client/attendee use case diagram

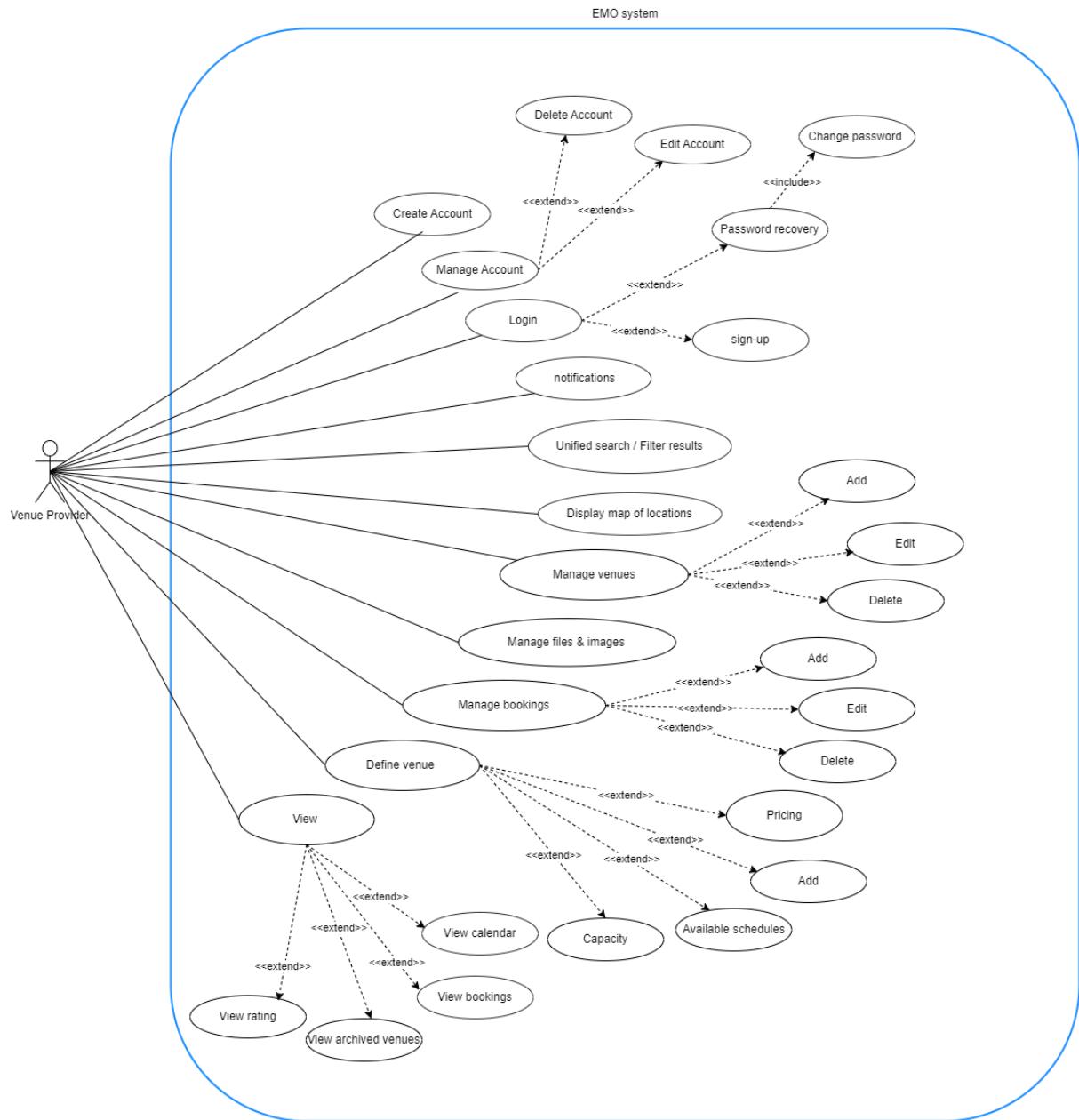


Figure 4.4 venue provider use case diagram

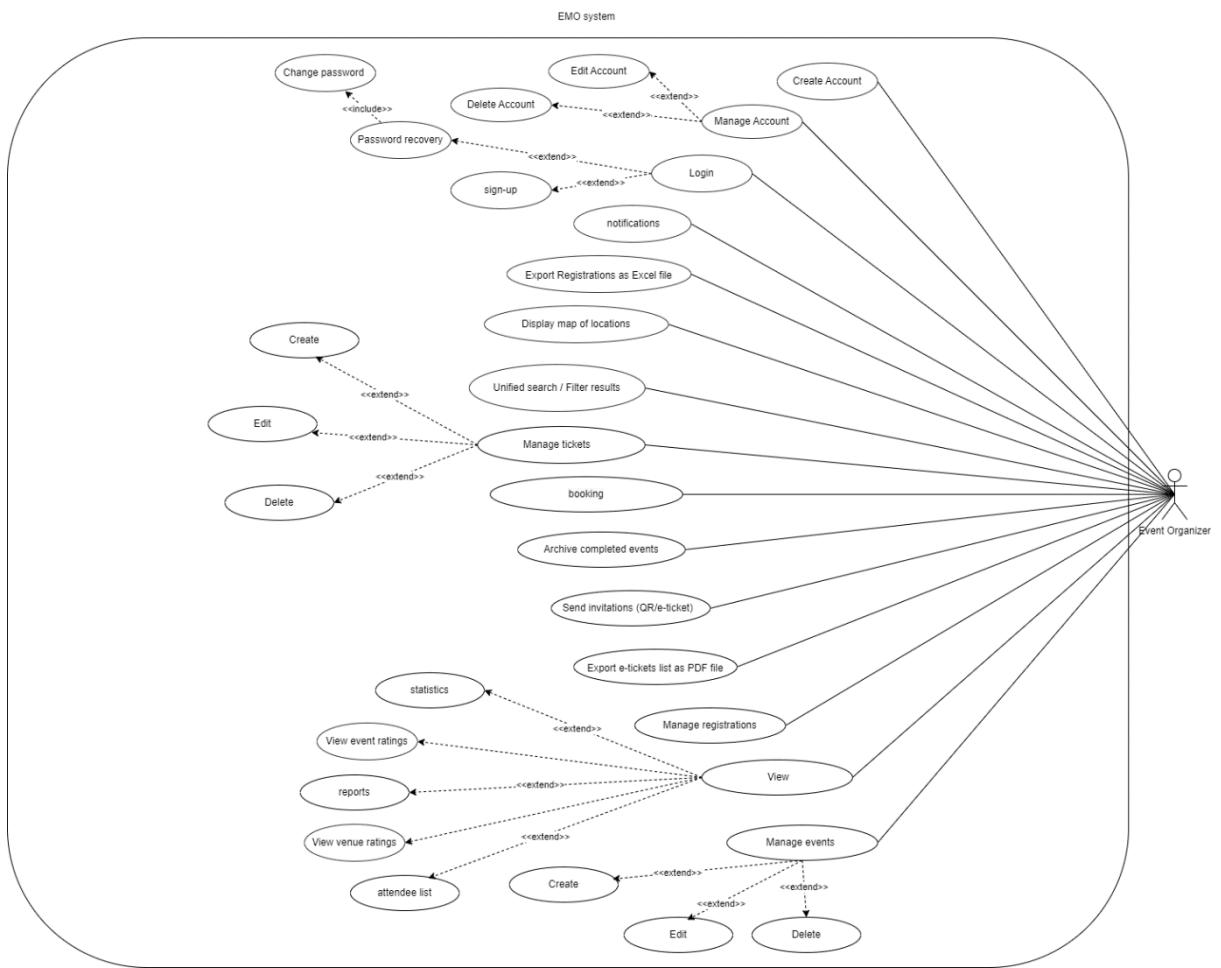


Figure 4.4 event organizer use case diagram

Use Case Specification

Table 4.4 login description

Field	Details
ID	UC-01
Use Case Name	Login
Actors	User (Attendee, Provider, Organizer, Admin)
Preconditions	User account exists; authentication service available
Main Flow	<ol style="list-style-type: none"> 1. User opens the login page and enters credentials (email/username and password). 2. System validates input format and authenticates credentials. 3. On success, system creates an issue a token and redirects user to the appropriate dashboard. 4. System logs the login event.
Alternate Flows	Invalid credentials → system shows error; locked account → system shows recovery instructions.
Postconditions	User authenticated; session active.

Table 4.5 register (User Registration) description

Field	Details
ID	UC-02
Use Case Name	Register (User Registration)
Actors	user
Preconditions	Registration forms available; email service for confirmation available
Main Flow	<ol style="list-style-type: none"> 1. Visitor opens home page and selects the type of account (client, venue provider, organizer) he wants to create. 2. Visitor fills required fields (name, username, email, password, confirm password). 3. System validates field formats and password strength. 4. System checks for duplicate email. 5. On success, system creates the account and sends a confirmation email. 6. system redirect the user to login page. 7. After activation, system grants basic permissions to the user.
Alternate Flows	Duplicate email → system shows error; invalid input → system requests correction; weak password → system shows error
Postconditions	User account created and activated after verification.

Table 4.6 Register Logout description

Field	Details
ID	UC-03
Use Case Name	Logout
Actors	Authenticated User; System
Preconditions	Active user session
Main Flow	<ol style="list-style-type: none"> 1. User clicks logout. 2. System invalidates the token. 3. System redirects user to the login page and displays confirmation. 4. System logs the logout event.
Alternate Flows	Session already expired → system redirects to login immediately.
Postconditions	Session terminated; user unauthenticated.

Table 4.7 view venues description

Field	Details
ID	UC-04
Use Case Name	View Venues
Actors	Attendee
Preconditions	Venues database prepared; listing UI available
Main Flow	<ol style="list-style-type: none"> 1. User opens the venues page. 2. System retrieves venues ordered by default. 3. System displays venues as cards. 4. For each item the system shows: name, location (city/short address), capacity, price, thumbnail image, rating, number of reviewers.
Alternate Flows	No venues → system displays message " No venues available ".
Postconditions	Venues list displayed or appropriate status message shown.

Table 4.8 view venues details description

Field	Details
ID	UC-05
Use Case Name	View Venue Details
Actors	Attendee
Preconditions	Complete venue record exists (images; specifications; schedules)
Main Flow	<ol style="list-style-type: none"> 1. User selects a venue. 2. System retrieves the full venue record. 3. System displays image gallery, full description, pricing, capacity, availability schedule. 4. System shows action buttons: Book, View reviews, Share location.
Alternate Flows	Venue not found or deleted → system displays " Venue not available ".
Postconditions	Venue details page displayed or an appropriate error message shown.

Table 4.9 view bookings description

Field	Details
ID	UC-06
Use Case Name	View Bookings
Actors	Attendee
Preconditions	User is logged in as Attendee; booking records associated with the user exist
Main Flow	<ol style="list-style-type: none"> 1. User opens the bookings page. 2. System retrieves current and past bookings. 3. System displays for each booking: booking number, venue, date & time, status (Confirmed / Pending / Cancelled). 4. System provides sorting/filtering options.
Alternate Flows	No bookings → system displays message " No bookings ".
Postconditions	Bookings list displayed or appropriate status message shown.

Table 4.10 add booking description

Field	Details
ID	UC-07
Use Case Name	Add Booking
Actors	Attendee
Preconditions	Logged in as attendee; venue available for requested date; valid booking form
Main Flow	<ol style="list-style-type: none"> 1. User fills booking form (venue; start date; end/date-time; number of attendees; contact details). 2. System immediately checks availability, capacity, and field validity. 3. System checks for duplicate bookings and time conflicts with other bookings. 4. If available, system creates the booking and sends confirmation notification.
Alternate Flows	Time conflict → system shows error explaining the reason.
Postconditions	Booking recorded with number and details and confirmation sent, or booking in pending state.

Table 4.11 edit booking description

Field	Details
ID	UC-08
Use Case Name	Edit Booking
Actors	Attendee (Owner of Booking)
Preconditions	Logged in as attendee; Editable booking exists; user has permission
Main Flow	<ol style="list-style-type: none"> User opens booking details and edits allowed fields (dates; number of attendees; notes). System rechecks availability and applies validation rules. If valid, system saves changes and displays success message.
Alternate Flows	Edit causes conflict → system shows error and prevents save.
Postconditions	Changes saved and audit trail updated.

Table 4.12 delete booking description

Field	Details
ID	UC-09
Use Case Name	Delete Booking
Actors	Attendee (Owner)
Preconditions	Booking exists; user has deletion permission
Main Flow	<ol style="list-style-type: none"> User clicks delete. System displays confirmation dialog explaining cancellation implications (fees/lead time). After confirmation, system marks booking as cancelled or deletes record per policy. System logs deletion reason and sends notification.
Alternate Flows	Deletion blocked due to cancellation policy or financial links → system shows reason and provides instructions.
Postconditions	Booking cancelled or deleted and operation logged.

Table 4.13 view events description

Field	Details
ID	UC-10
Use Case Name	View Events
Actors	Attendee
Preconditions	Events are registered in the system
Main Flow	<ol style="list-style-type: none"> User opens the events page. System retrieves events list. System displays for each event: name, date, location, thumbnail image. System supports pagination when results are large.
Alternate Flows	No events → system displays message " No events ".
Postconditions	Events list displayed or appropriate status message shown.

Table 4.14 view events details description

Field	Details
ID	UC-11
Use Case Name	View Event Details
Actors	Attendee
Preconditions	Complete event record exists (description, schedule, tickets, organizers, coordinates)
Main Flow	<ol style="list-style-type: none"> 1. User selects an event from the list. 2. System retrieves the full event record. 3. System displays time & date, venue, event description, ticket options, organizers list, and registration/purchase links. 4. System displays a map of the location if available.
Alternate Flows	Event not found → system shows a clear alert in the notifications area: " Event not available ".
Postconditions	Event details page displayed or an alert shown.

Table 4.15 unified search for venues description

Field	Details
ID	UC-12
Use Case Name	Unified Search for Venues
Actors	User (Attendee, Venue Provider, Event Organizer)
Preconditions	Venues and events indexed and searchable
Main Flow	<ol style="list-style-type: none"> 1. User enters keywords or phrase in the search field. 2. System executes the query against the index and returns results grouped by category (Venues). 3. System supports partial matches and is case-insensitive. 4. System supports pagination for large result sets.
Alternate Flows	No matches → system displays " No matching results " and suggests broadening the search.
Postconditions	Search results displayed and categorized.

Table 4.16 unified search for events description

Field	Details
ID	UC-13
Use Case Name	Unified Search for Events
Actors	User (Attendee, Event Organizer)
Preconditions	Venues and events indexed and searchable
Main Flow	<ol style="list-style-type: none"> 1. User enters keywords or phrase in the search field. 2. System executes the query against the index and returns results grouped by category (Events). 3. System supports partial matches and is case-insensitive. 4. System supports pagination for large result sets.
Alternate Flows	No matches → system displays " No matching results " and suggests broadening the search.
Postconditions	Search results displayed and categorized.

Table 4.18 filter venues by specific criteria description

Field	Details
ID	UC-14
Use Case Name	Filter Venues by Specific Criteria
Actors	User (Attendee, Venue Provider, Event Organizer)
Preconditions	Data contains filterable attributes (location, price, capacity, type, date)
Main Flow	<ol style="list-style-type: none"> 1. User selects filter criteria via the UI. 2. Client sends filter criteria to the server. 3. System executes optimized queries or uses the search engine to return matching items. 4. System displays filtered results and shows counts per filter. 5. User can reset filters easily.
Alternate Flows	Invalid filter values → system ignores invalid values or shows a warning; No results → system displays an appropriate message.
Postconditions	Filtered list displayed; filters can be reset.

Table 4.18 filter events by specific criteria description

Field	Details
ID	UC-15
Use Case Name	Filter Events by Specific Criteria
Actors	User (Attendee, Event Organizer)
Preconditions	Data contains filterable attributes (location, price, capacity, type, date)
Main Flow	<ol style="list-style-type: none"> 1. User selects filter criteria via the UI. 2. Client sends filter criteria to the server. 3. System executes optimized queries or uses the search engine to return matching items. 4. System displays filtered results and shows counts per filter. 5. User can reset filters easily.
Alternate Flows	Invalid filter values → system ignores invalid values or shows a warning; No results → system displays an appropriate message.
Postconditions	Filtered list displayed; filters can be reset.

Table 4.19 view registration description

Field	Details
ID	UC-16
Use Case Name	View Registrations
Actors	Attendee
Preconditions	User is logged in; registration records associated with the user exist
Main Flow	<ol style="list-style-type: none"> 1. User opens the registrations page. 2. System retrieves the user's registrations. 3. System displays each registration with attendance status and event details. 4. System supports pagination when results are large.
Alternate Flows	No registrations → system displays " No registrations ".
Postconditions	Registrations list displayed or appropriate status message shown.

Table 4.20 add registration description

Field	Details
ID	UC-17
Use Case Name	Add Registration
Actors	Attendee
Preconditions	Seats/tickets available; user is logged in
Main Flow	<ol style="list-style-type: none"> 1. User opens the event page and fills the registration form (name, email, number of tickets, additional data). 2. System checks availability. 3. On success, system creates the registration, displays confirmation, and sends notification with ticket details.
Alternate Flows	Tickets sold out → system shows error and offers waitlist option if available.
Postconditions	Registration created or pending; confirmation sent.

Table 4.21 edit registration description

Field	Details
ID	UC-18
Use Case Name	Edit Registration
Actors	Attendee (Owner of Registration)
Preconditions	Logged in; Editable registration exists; organizer policy allows edits
Main Flow	<ol style="list-style-type: none"> 1. User opens their registration and selects edit. 2. User updates allowed fields (ticket quantity, contact details). 3. System validates changes and checks availability. 4. If valid, system saves changes and sends confirmation notification.
Alternate Flows	Edit not allowed → system displays reason (organizer policy or sold-out status).
Postconditions	Registration updated and audit trail recorded.

Table 4.22 delete registration description

Field	Details
ID	UC-19
Use Case Name	Delete Registration
Actors	Attendee (Owner); System
Preconditions	Logged in; Registration exists; cancellation policy defined
Main Flow	1. User initiates cancellation. 2. System displays confirmation with cancellation implications (refunds/fees/lead time). 3. After confirmation, system cancels the registration and updates ticket availability. 4. System sends confirmation to the user.
Alternate Flows	Cancellation not permitted → system shows reason and alternative steps (contact support or partial refund).
Postconditions	Registration cancelled and ticket availability updated.

Table 4.23 view venue ratings description

Field	Details
ID	UC-20
Use Case Name	View Venue Ratings
Actors	User (Attendee, Venue Provider, Event Organizer)
Preconditions	Ratings and reviews exist for the venue
Main Flow	1. User opens the venue page. 2. System retrieves average rating, total reviews, and review details (reviewer name or anonymous, numeric rating, comment, publish date). 3. System displays star distribution and provides sort/filter options for reviews.
Alternate Flows	No reviews → system displays "No reviews" and prompts user to add the first review.
Postconditions	Ratings and reviews displayed; average rating calculated.

Table 4.24 add venue rating description

Field	Details
ID	UC-21
Use Case Name	Add Venue Rating
Actors	Attendee
Preconditions	User is logged in; system policy defines who may rate (e.g., attendees who visited)
Main Flow	1. User opens the rating form and selects a numeric value (1–5) and optional text. 2. User submits the rating. 3. System validates eligibility and content. 4. System saves the rating, updates the venue average, and displays success message.
Alternate Flows	Duplicate rating attempt → system enforces policy (reject or allow update per policy) and informs user.
Postconditions	Rating stored and venue average updated.

Table 4.25 edit venue rating description

Field	Details
ID	UC-22
Use Case Name	Edit Venue Rating
Actors	Attendee (Owner of Rating); System
Preconditions	User has an existing rating; edit window is within allowed timeframe
Main Flow	<ol style="list-style-type: none"> 1. User opens their venue rating. 2. User edits allowed fields (numeric value, comment). 3. User submits the changes. 4. System validates eligibility and content. 5. System updates the rating and recalculates the venue average. 6. System records the edit in the audit trail and displays a success message.
Alternate Flows	Edit window expired → system prevents edit and shows reason.
Postconditions	Rating updated and audit trail recorded.

Table 4.26 delete venue rating description

Field	Details
ID	UC-23
Use Case Name	Delete Venue Rating
Actors	Attendee (Owner) or Authorized Venue Provider
Preconditions	Logged in; User has an existing rating; deletion permitted by policy
Main Flow	<ol style="list-style-type: none"> 1. User initiates delete on the rating. 2. System shows confirmation explaining effect on average. 3. After confirmation, system deletes or marks the rating as removed and updates the venue average. 4. System logs the deletion.
Alternate Flows	Attempt to delete another user's rating → system blocks action and shows error.
Postconditions	Rating removed or flagged; venue average updated; deletion logged.

Table 4.27 view event ratings description

Field	Details
ID	UC-24
Use Case Name	View Event Ratings
Actors	User (Attendee, Venue Provider, Event Organizer)
Preconditions	Ratings exist for the event
Main Flow	<ol style="list-style-type: none"> 1. User opens the event page. 2. System retrieves ratings and comments. 3. System displays reviewer details (or anonymous), numeric rating, comment, average rating, and total count.
Alternate Flows	No ratings → system displays an appropriate message and encourages adding the first rating.
Postconditions	Event ratings displayed and average calculated.

Table 4.28 add event rating description

Field	Details
ID	UC-25
Use Case Name	Add Event Rating
Actors	Attendee
Preconditions	User is logged in; eligibility to rate determined by attendance policy
Main Flow	<ol style="list-style-type: none"> 1. User fills rating form (numeric value and optional comment). 2. User submits the rating. 3. System validates eligibility and content. 4. System saves the rating, updates event average, and optionally notifies the organizer.
Alternate Flows	Duplicate rating attempt → system enforces policy and informs user.
Postconditions	Rating saved and event average updated.

Table 4.29 edit event rating description

Field	Details
ID	UC-26
Use Case Name	Edit Event Rating
Actors	Attendee (Owner of Rating)
Preconditions	Logged in; User has an existing event rating; edit window is within allowed timeframe
Main Flow	<ol style="list-style-type: none"> 1. User opens their event rating. 2. User edits allowed fields (numeric value, comment). 3. User submits the changes. 4. System validates eligibility and content. 5. System updates the rating and recalculates the event average. 6. System records the edit in the audit trail and displays a success message.
Alternate Flows	Edit window expired → system blocks the edit and shows the reason.
Postconditions	Rating updated; audit trail recorded.
Performance & Security	Protect against manipulation; log edits for auditing.

Table 4.30 delete event rating description

Field	Details
ID	UC-27
Use Case Name	Delete Event Rating
Actors	Attendee (Owner) or Event Organizer
Preconditions	Logged in; User has an existing event rating; deletion permitted by policy
Main Flow	<ol style="list-style-type: none"> User initiates delete on their event rating. System shows confirmation explaining the effect on averages. After confirmation, system deletes or flags the rating as removed. System recalculates the event average. System logs the deletion and notifies the user.
Alternate Flows	Attempt to delete another user's rating → system blocks the action and shows an error.
Postconditions	Rating removed or flagged; averages updated; deletion logged.

Table 4.31 view organizers description

Field	Details
ID	UC-28
Use Case Name	View Organizers
Actors	User (Attendee, Venue Provider, Event Organizer)
Preconditions	Organizer records exist and are complete
Main Flow	<ol style="list-style-type: none"> User opens the organizers page. System retrieves organizer list. System displays for each organizer: name, short bio, number of past events, overall rating. System provides link to each organizer's detail page.
Alternate Flows	No organizers → system displays an appropriate message.
Postconditions	Organizers list displayed; links to details available.
Performance & Security	Show only public information; protect contact details.

Table 4.32 view providers description

Field	Details
ID	UC-29
Use Case Name	View Providers
Actors	User (Attendee, Venue Provider, Event Organizer)
Preconditions	Provider records exist and are complete
Main Flow	<ol style="list-style-type: none"> User opens the provider's page. System retrieves provider list. System displays for each provider: name, number of venues, overall rating, and links to contact or view venues.
Alternate Flows	No providers → system displays an appropriate message.
Postconditions	Providers list displayed; action links available.

Table 4.33 filter providers by specific criteria description

Field	Details
ID	UC-30
Use Case Name	Filter Providers by Specific Criteria
Actors	User (Attendee, Venue Provider, Event Organizer)
Preconditions	Filterable attributes exist (city, rating, experience)
Main Flow	<ol style="list-style-type: none"> 1. User opens provider page and selects filter criteria. 2. Client sends criteria to server. 3. System executes queries and returns matching items. 4. System displays results and shows counts per filter. 5. User can combine filters or reset them.
Alternate Flows	No matches → system displays a message and suggests broadening criteria.
Postconditions	Filtered list displayed; filters resettable.

Table 4.34 filter providers by specific criteria description

Field	Details
ID	UC-31
Use Case Name	Filter Organizers by Specific Criteria
Actors	User (Attendee, Venue Provider, Event Organizer)
Preconditions	Filterable attributes exist (city, rating, experience)
Main Flow	<ol style="list-style-type: none"> 1. User opens Organizers page. 2. User selects filter criteria. 3. Client sends criteria to server. 4. System executes queries and returns matching items. 5. System displays results and shows counts per filter. 6. User can combine filters or reset them.
Alternate Flows	No matches → system displays a message and suggests broadening criteria.
Postconditions	Filtered list displayed; filters resettable.

Table 4.35 upvote description

Field	Details
ID	UC-32
Use Case Name	Upvote
Actors	Attendee
Preconditions	User is logged in; target item is votable and exists
Main Flow	<ol style="list-style-type: none"> 1. User clicks the Upvote button on an item. 2. System records the vote and increments the displayed count immediately. 3. System stores voter identity and timestamp in the audit log. 4. System prevents duplicate votes per policy (or toggles vote if policy allows).
Alternate Flows	Duplicate vote attempt → system blocks or toggles per policy and informs the user.
Postconditions	Vote recorded; UI updated.

Table 4.36 downvote description

Field	Details
ID	UC-33
Use Case Name	Downvote
Actors	Attendee
Preconditions	User is logged in; target item is votable and exists
Main Flow	<ol style="list-style-type: none"> 1. User clicks the Downvote button on an item. 2. System records the vote and updates the displayed count immediately. 3. System stores voter identity and timestamp in the audit log. 4. System prevents duplicate votes per policy (or toggles vote if policy allows).
Alternate Flows	Duplicate vote attempt → system blocks or toggles per policy and informs the user.
Postconditions	Vote recorded; UI updated.

Table 4.37 view recent activities description

Field	Details
ID	UC-34
Use Case Name	View Recent Activities
Actors	Attendee
Preconditions	Logged in; Activity log exists for the user
Main Flow	<ol style="list-style-type: none"> 1. User opens the Recent Activities page. 2. System retrieves chronological activity entries (bookings, registrations, ratings, votes). 3. System displays each entry with timestamp and link to the related item. 4. System provides filters by activity type and date range.
Alternate Flows	No activities → system displays an appropriate message.
Postconditions	Activity timeline displayed; filters functional.

Table 4.38 display map of event and venue locations description

Field	Details
ID	UC-35
Use Case Name	Display Map of Event and Venue Locations
Actors	User (Attendee, Venue Provider, Event Organizer)
Preconditions	Accurate geo-coordinates for venues/events; map provider integration available
Main Flow	<ol style="list-style-type: none"> 1. User opens the map page or an item detail page. 2. System requests map tiles from the map provider. 3. System places pins for each location. 4. User clicks a pin; system shows a brief info window with name and link to details. 5. User can zoom, pan, and apply filters to displayed locations.
Alternate Flows	Map provider unavailable → system shows a fallback list of locations and an error message.
Postconditions	Interactive map displayed or fallback list shown.

Table 4.39 browse venues (provider) description

Field	Details
ID	UC-36
Use Case Name	Browse Venues (Provider)
Actors	Venue Provider
Preconditions	Provider is logged in; provider has venues in the system
Main Flow	<ol style="list-style-type: none"> 1. Provider opens their venues dashboard. 2. System retrieves the provider's venues ordered by default. 3. System displays venues as cards. 4. For each item the system shows: name, location (city/short address), capacity, price, thumbnail image, rating, number of reviewers. 5. Provider can search, filter, and navigate to each venue's detail or edit page.
Alternate Flows	No venues → system displays " No venues available " and a link to add a new venue.
Postconditions	Provider's venues listed and manageable.

Table 4.40 brows archived venues description

Field	Details
ID	UC-37
Use Case Name	Browse Archived Venues
Actors	Venue Provider
Preconditions	Provider is logged in; archived venues exist for the provider
Main Flow	<ol style="list-style-type: none"> Provider opens the “Archived Venues” section in the dashboard. System retrieves archived venues with archive date and reason. System displays archived venue with actions (Restore, Delete).
Alternate Flows	No archived venues → system displays an appropriate message.
Postconditions	Archived venues listed; restore/delete actions available per policy.

Table 4.41 add venue description

Field	Details
ID	UC-38
Use Case Name	Add Venue
Actors	Venue Provider
Preconditions	Provider is logged in; add-venue form available; provider has creation permission
Main Flow	<ol style="list-style-type: none"> Provider opens the Add Venue form. Provider fills required fields (name, description, location, capacity, pricing, amenities, images, coordinates). System validates fields and uploaded files. System stores the venue record and media, shows preview and public/private link, and notifies the provider.
Alternate Flows	Missing or invalid data → system shows validation errors and requests correction.
Postconditions	Venue created and visible in provider dashboard (and public listings if published).

Table 4.42 edit venue description

Field	Details
ID	UC-39
Use Case Name	Edit Venue
Actors	Venue Provider (Owner)
Preconditions	Provider is logged in; Venue exists and belongs to provider; provider has edit permission
Main Flow	<ol style="list-style-type: none"> Provider opens the venue edit page. System loads current fields and change history. Provider updates fields and submits changes. System validates updates (e.g., schedule conflicts) and saves changes. System updates public listing and records audit trail.
Alternate Flows	Concurrent edits or validation failures → system shows conflict/error and prevents save until resolved.
Postconditions	Venue record updated; audit trail recorded.

Table 4.43 delete venue description

Field	Details
ID	UC-40
Use Case Name	Delete Venue
Actors	Venue Provider (Owner)
Preconditions	Venue exists; provider has deletion permission; Provider is logged in;
Main Flow	<ol style="list-style-type: none"> 1. Provider requests venue deletion. 2. System displays warning and consequences (bookings, data retention). 3. System checks for active bookings or obligations. 4. If allowed, system performs soft-delete or permanent delete per policy, updates indexes, and notifies relevant parties. 5. System logs the deletion action.
Alternate Flows	Active bookings or financial obligations → system blocks deletion and provides instructions.
Postconditions	Venue deleted or archived per policy; logs updated.

Table 4.44 archive venue description

Field	Details
ID	UC-41
Use Case Name	Archive Venue
Actors	Venue Provider (Owner)
Preconditions	Provider is logged in; Venue exists; no blocking active bookings
Main Flow	<ol style="list-style-type: none"> 1. Provider clicks “Archive” for a venue. 2. System shows confirmation explaining effects. 3. Provider confirms; system sets venue status to Archived, updates search index, and hides it from public listings. 4. System records archive date and reason.
Alternate Flows	Archive blocked due to active bookings → system shows reason and next steps.
Postconditions	Venue status set to Archived and reflected in UI and search.

Table 4.45 unarchive venue description

Field	Details
ID	UC-42
Use Case Name	Unarchive Venue
Actors	Venue Provider (Owner)
Preconditions	Venue is archived; provider has restored permission; Provider is logged in;
Main Flow	<ol style="list-style-type: none"> 1. Provider requests to unarchive a venue. 2. System verifies conditions (no conflicts). 3. System sets venue status to Active, reindexes it, and confirms restoration. 4. System logs the restore action.
Alternate Flows	Restore blocked due to conflicts → system shows reason
Postconditions	Venue active and visible in listings per settings.

Table 4.46 browse bookings description

Field	Details
ID	UC-43
Use Case Name	Browse Bookings (Provider)
Actors	Venue Provider
Preconditions	Provider is logged in; bookings exist for provider's venues
Main Flow	<ol style="list-style-type: none"> Provider opens the bookings page in the dashboard. System retrieves bookings with filters (date, status, venue). System displays bookings table with action buttons. Provider selects a booking to view details and take actions (Accept/Reject/Cancel).
Alternate Flows	No bookings → system displays empty state and suggestions.
Postconditions	Bookings displayed and manageable; actions logged.

Table 4.47 accept booking description

Field	Details
ID	UC-44
Use Case Name	Accept Booking (Provider)
Actors	Venue Provider
Preconditions	Provider is logged in; Pending booking exists; provider authorized to accept
Main Flow	<ol style="list-style-type: none"> Provider opens pending booking details. Provider clicks "Accept." System rechecks availability and conflicts. If available, system updates booking status to Accepted, reserves resources, and notifies the client. System logs the status change.
Alternate Flows	Conflict detected on recheck → system shows error and blocks acceptance or suggests alternatives.
Postconditions	Booking status set to Accepted; client notified; logs updated.

Table 4.48 reject booking description

Field	Details
ID	UC-45
Use Case Name	Reject Booking (Provider)
Actors	Venue Provider
Preconditions	Pending booking exists; Provider is logged in;
Main Flow	<ol style="list-style-type: none"> Provider reviews pending booking and selects "Reject" with optional reason. System updates booking status to Rejected, releases reserved resources, and notifies the client with the reason. System logs the rejection.
Alternate Flows	Error updating status → system shows error and requests retry.
Postconditions	Booking marked Rejected; resources freed; client notified; action logged.

Table 4.49 cancel booking description

Field	Details
ID	UC-46
Use Case Name	Cancel Booking (Provider)
Actors	Venue Provider
Preconditions	Booking exists (Accepted/Confirmed); provider authorized to cancel
Main Flow	<ol style="list-style-type: none"> Provider opens booking and selects “Cancel.” System displays cancellation implications (fees/refunds). Provider confirms; system updates booking status to Cancelled, processes refunds if applicable, and notifies the client. System logs the cancellation.
Alternate Flows	Cancellation blocked due to policy or ongoing transaction → system shows reason and next steps.
Postconditions	Booking cancelled; refunds processed as needed; seats/resources freed; logs updated.

Table 4.50 browse events (organizer) description

Field	Details
ID	UC-47
Use Case Name	Browse Events (Organizer)
Actors	Event Organizer
Preconditions	Organizer is logged in; organizer has events in the system
Main Flow	<ol style="list-style-type: none"> Organizer opens the events dashboard. System retrieves the organizer’s events with status (Active / Draft / Archived). System displays events as a list or cards with key fields (title, date, venue, status). Organizer can filter, sort, and navigate to event details or edit pages.
Alternate Flows	No events → system displays a message and a link to create a new event.
Postconditions	Organizer’s events listed and manageable.

Table 4.51 browse archived events description

Field	Details
ID	UC-48
Use Case Name	Browse Archived Events
Actors	Event Organizer
Preconditions	Organizer is logged in; archived events exist for the organizer
Main Flow	<ol style="list-style-type: none"> Organizer opens the “Archived Events” section. System retrieves archived events with archive date. System displays archived entries with actions (Restore, Delete).
Alternate Flows	No archived events → system displays an appropriate message.
Postconditions	Archived events listed; restore/delete actions available per policy.

Table 4.52 create event description

Field	Details
ID	UC-49
Use Case Name	Create Event
Actors	Event Organizer
Preconditions	Organizer is logged in; venue selected or venue details provided
Main Flow	<ol style="list-style-type: none"> 1. Organizer opens the Create Event form. 2. Organizer fills required fields (title, description, dates, sessions, ticket types, speakers, images, coordinates). 3. System validates inputs and checks venue availability and schedule conflicts. 4. Organizer chooses Draft or Publish; system saves the event accordingly and returns confirmation.
Alternate Flows	Venue conflict → system shows conflict details and suggests alternatives; missing required fields → system highlights errors.
Postconditions	Event created as Draft or Published; audit log updated.

Table 4.53 edit event description

Field	Details
ID	UC-50
Use Case Name	Edit Event
Actors	Event Organizer
Preconditions	Event exists and belongs to organizer; organizer has edit permission
Main Flow	<ol style="list-style-type: none"> 1. Organizer opens the event edit page. 2. System loads current event data and change history. 3. Organizer updates fields and submits changes. 4. System validates changes (ticket impacts, schedule conflicts) and saves updates. 5. System notifies affected attendees if required.
Alternate Flows	Changes conflict with sold tickets or schedules → system warns and requires confirmation or admin intervention.
Postconditions	Event updated; notifications and audit trail recorded.

Table 4.54 delete event description

Field	Details
ID	UC-51
Use Case Name	Delete Event
Actors	Event Organizer
Preconditions	Event exists; check for unresolved obligations (refunds, disputes); Organizer is logged in;
Main Flow	<ol style="list-style-type: none"> 1. Organizer requests event deletion. 2. System displays consequences (registrations, refunds). 3. System checks for pending transactions or disputes. 4. If allowed, system deletes or archives the event per policy and notifies registrants. 5. System logs the deletion.
Alternate Flows	Pending refunds/disputes → system blocks deletion and provides next steps or escalates to admin.
Postconditions	Event deleted or archived; records retained per retention policy.

Table 4.55 archive event description

Field	Details
ID	UC-52
Use Case Name	Archive Event
Actors	Event Organizer
Preconditions	Event exists and is eligible for archiving; no blocking obligations
Main Flow	<ol style="list-style-type: none"> 1. Organizer selects “Archive” for an event. 2. System shows confirmation and effects. 3. Organizer confirms; system sets event status to Archived, hides it from public listings, and records archive metadata.
Alternate Flows	Archive blocked due to obligations → system shows reason and remediation steps.
Postconditions	Event archived and excluded from public search/listings.

Table 4.56 unarchive event description

Field	Details
ID	UC-53
Use Case Name	Unarchive Event
Actors	Event Organizer
Preconditions	Event is archived; organizer has restore permission
Main Flow	<ol style="list-style-type: none"> 1. Organizer requests to unarchive the event. 2. System verifies conditions (no conflicts). 3. System sets event status to Active or Draft, reindexes it, and confirms restoration.
Alternate Flows	Restore blocked due to conflicts → system shows reason and suggested fixes.
Postconditions	Event active or draft and visible per settings.

Table 4.57 browse registrations description

Field	Details
ID	UC-54
Use Case Name	Browse Registrations
Actors	Event Organizer
Preconditions	Organizer is logged in; registrations exist for organizer's events
Main Flow	<ol style="list-style-type: none"> 1. Organizer opens the registrations page. 2. System retrieves registrations with filters (status, date, ticket type). 3. System displays registrations list/table with quick actions. 4. Organizer selects a registration to view details or take action (Accept/Reject/Cancel).
Alternate Flows	No registrations → system displays an appropriate message.
Postconditions	Registrations displayed and manageable; actions logged.

Table 4.58 accept registration description

Field	Details
ID	UC-55
Use Case Name	Accept Registration
Actors	Event Organizer
Preconditions	Pending registration exists; organizer authorized
Main Flow	<ol style="list-style-type: none"> 1. Organizer reviews the pending registration. 2. Organizer clicks "Accept." 3. System verifies ticket availability and updates registration status to Accepted. 4. System notifies the registrant and updates ticket counts. 5. System logs the action.
Alternate Flows	Insufficient tickets → system blocks acceptance and suggests alternatives.
Postconditions	Registration accepted; registrant notified; ticket counts updated.

Table 4.59 reject registration description

Field	Details
ID	UC-56
Use Case Name	Reject Registration
Actors	Event Organizer (owner)
Preconditions	Pending registration exists; organizer authorized
Main Flow	<ol style="list-style-type: none"> 1. Organizer selects "Reject" and optionally provides a reason. 2. System updates registration status to Rejected and releases any reserved tickets. 3. System notifies the registrant with the reason. 4. System logs the rejection.
Alternate Flows	Error updating status → system shows error and requests retry.
Postconditions	Registration rejected; tickets released; registrant notified; action logged.

Table 4.60 cancel registration description

Field	Details
ID	UC-57
Use Case Name	Cancel Registration
Actors	Event Organizer or Attendee
Preconditions	Registration exists; cancellation policy defined
Main Flow	<ol style="list-style-type: none"> 1. Organizer or attendee initiates cancellation. 2. System displays confirmation with cancellation implications (refunds/fees/lead time). 3. User confirms cancellation. 4. System updates registration status to Cancelled and updates ticket availability. 5. System notifies affected parties (organizer/attendee) and logs the action.
Alternate Flows	Cancellation blocked by policy or ongoing transaction → system shows reason and next steps.
Postconditions	Registration cancelled; ticket availability updated; action logged.
Performance & Security	Process refunds if applicable; record audit trail; enforce authorization checks.

Table 4.61 explore registration as excel file description

Field	Details
ID	UC-58
Use Case Name	Export Registrations as Excel file
Actors	Event Organizer
Preconditions	Registrations exist; user has export permission
Main Flow	<ol style="list-style-type: none"> 1. Organizer selects export criteria (date range, status, event). 2. System gathers matching registration data. 3. System generates an Excel (XLSX) file with the requested columns. 4. System provides the file for download and logs the export action.
Alternate Flows	Export fails due to large dataset → system suggests narrowing the range or paginated export and shows an error.
Postconditions	Excel file generated and available for download; export logged.

Table 4.62 view booking (organizer) description

Field	Details
ID	UC-59
Use Case Name	View Booking (Organizer)
Actors	Event Organizer
Preconditions	Organizer is logged in; bookings exist for organizer's events
Main Flow	<ol style="list-style-type: none"> 1. Organizer opens the bookings page. 2. System retrieves current and past bookings. 3. System displays for each booking: booking number, venue, date & time, status (Confirmed / Pending / Cancelled). 4. System provides sorting/filtering options.
Alternate Flows	No bookings → system displays an appropriate message.
Postconditions	Booking details accessible; organizer can take management actions.

Table 4.63 add booking (organizer) description

Field	Details
ID	UC-60
Use Case Name	Add Booking (Organizer)
Actors	Event Organizer
Preconditions	Organizer authorized to create bookings; venue available
Main Flow	<ol style="list-style-type: none"> 1. Organizer fills booking form (venue; start date; end/date-time; number of attendees; contact details). 2. System immediately checks availability, capacity, and field validity. 3. System checks for duplicate bookings and time conflicts with other bookings. 4. If available, system creates the booking and sends confirmation notification.
Alternate Flows	Insufficient availability → system shows error and suggests alternatives.
Postconditions	Booking created and confirmations sent.

Table 4.64 edit booking (organizer) description

Field	Details
ID	UC-61
Use Case Name	Edit Booking (Organizer)
Actors	Event Organizer
Preconditions	Booking exists; organizer has permission to modify
Main Flow	<ol style="list-style-type: none"> User opens booking details and edits allowed fields (dates; number of attendees; notes). System rechecks availability and applies validation rules. If valid, system saves changes and displays success message.
Alternate Flows	Change causes conflict or violates policy → system blocks save and shows reason.
Postconditions	Booking updated; notifications sent; audit trail recorded.

Table 4.65 delete booking (organizer) description

Field	Details
ID	UC-62
Use Case Name	Delete Booking (Organizer)
Actors	Event Organizer
Preconditions	Booking exists; deletion policy defined; Organizer authorized
Main Flow	<ol style="list-style-type: none"> Organizer initiates booking deletion/cancellation. System displays consequences (refunds/fees). Organizer confirms action. System cancels or deletes the booking per policy, processes refunds if needed, and notifies the client. System logs the deletion.
Alternate Flows	Deletion blocked due to disputes or financial constraints → system shows reason and next steps.
Postconditions	Booking cancelled/deleted; refunds processed as applicable; logs updated.

Table 4.66 Export Registrations as Excel file (organizer) description

Field	Details
ID	UC-63
Use Case Name	Export Registrations as Excel file
Actors	Event Organizer; Event Manager; System
Preconditions	User authorized to export; registrations exist within the selected range; XLSX generation service available
Main Flow	<ol style="list-style-type: none"> 1. Organizer opens the event registrations management page. 2. Organizer selects export criteria (date range, registration status, fields). 3. Organizer clicks "Export" or "Download as Excel". 4. System validates permissions and criteria, retrieves registrations, and generates an XLSX file. 5. System starts the download or provides a download link. 6. System logs the export action (user, criteria, timestamp).
Alternate Flows	- No matching registrations: System shows "No registrations found" and cancels export.
Postconditions	XLSX file is available for download or queued for delivery; export action recorded in audit logs with criteria and timestamp.

Table 4.67 View e-ticket description

Field	Details
ID	UC-64
Use Case Name	View e-ticket
Actors	Attendee; Event Organizer
Preconditions	User authenticated; valid booking/registration exists; e-ticket generation service available; e-ticket linked to booking.
Main Flow	<ol style="list-style-type: none"> 1. User opens "My Bookings" or receives a link/email for the e-ticket. 2. User selects the booking and chooses "View e-ticket". 3. System verifies booking and authorization. 4. System retrieves the e-ticket (QR code / e-ticket details) and displays it with event info, seat/entry details, and validity. 5. User can download, print, or show the e-ticket on device.
Alternate Flows	<ul style="list-style-type: none"> - Invalid/expired booking: System shows an error and suggests contacting organizer. - E-ticket not generated yet: System offers to regenerate or notifies user when ready.
Postconditions	E-ticket displayed or appropriate error shown; user can present/download the e-ticket; access attempt can be logged.

Table 4.68 View e-ticket description

Field	Details
ID	UC-65
Use Case Name	Create e-ticket
Actors	Event Organizer;
Preconditions	Valid booking/registration exists; e-ticket generation service available; organizer/system has required ticket template and metadata
Main Flow	<ol style="list-style-type: none"> 1. Trigger: booking confirmed or organizer requests ticket generation. 2. System gathers booking and attendee details. 3. System generates e-ticket (unique ID, QR/barcode, validity, seat info). 4. System attaches e-ticket to booking record and stores securely. 5. System delivers e-ticket to attendee (email, in-app, or SMS) and marks status as "issued". 6. System logs generation event (who/what triggered, timestamp).
Alternate Flows	<ul style="list-style-type: none"> - Generation error: retry generation or queue job and notify organizer/attendee. - Duplicate booking: detect and prevent duplicate tickets; notify organizer. - Delivery failure: retry sending or provide in-app access.
Postconditions	E-ticket created, linked to booking, and delivered or queued for delivery; generation recorded in audit logs; ticket status set to issued.

Table 4.69 : Export e-tickets list as PDF file description

Field	Details
ID	UC-66
Use Case Name	Export e-tickets list as PDF file
Actors	Event Organizer
Preconditions	User authorized to export; e-tickets exist for selected event/range; PDF generation service/template available
Main Flow	<ol style="list-style-type: none"> 1. Actor opens e-tickets management page. 2. Actor selects export criteria (date range, status, fields, layout). 3. Actor clicks "Export as PDF". 4. System validates permissions and criteria, queries e-tickets and compiles data. 5. System generates a PDF (list or formatted report) and provides download link or starts download. 6. System logs the export action (user, criteria, timestamp).
Alternate Flows	<ul style="list-style-type: none"> - No matching e-tickets: show "No e-tickets found" and abort.

	<ul style="list-style-type: none"> - Permission denied: show authorization error and abort. - Generation error: show error, log failure, offer retry or queue job. - Large export: queue generation and notify user when ready.
Postconditions	PDF file available for download or queued for delivery; export recorded in audit logs with criteria and timestamp; exported snapshot matches queried e-tickets.

Table 4.69 : display an interactive calendar description

Field	Details
ID	UC-67
Use Case Name	Display an interactive calendar
Actors	Venue Provider
Preconditions	User authenticated if required; bookings exist; calendar service/library available
Main Flow	<ol style="list-style-type: none"> 1. User opens calendar view; 2. System loads events for selected scope; 3. User navigates (day/week/month), filters, or searches; 4. System updates view and details pane; 5. User selects an event to view details or take actions (book, cancel, view e-ticket).
Alternate Flows	No events for range → show empty state and suggestion to change filters; Offline → show cached events with notice; Permission restricted events → hide details and show request access option.
Postconditions	Calendar displays current event data; user can navigate, filter, and open event details; interactions are logged as configured.

- Activity Diagrams

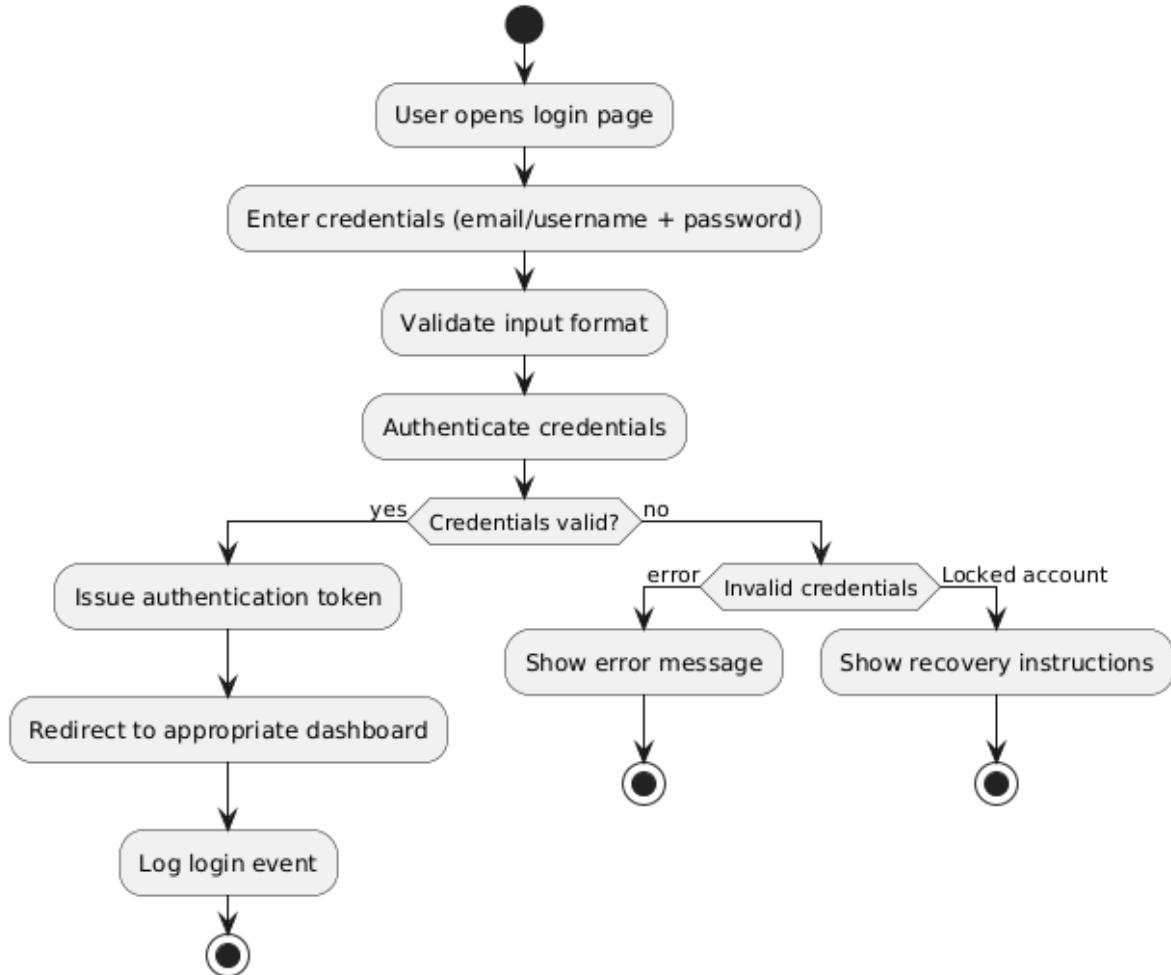


Figure 4.1 login activity diagram

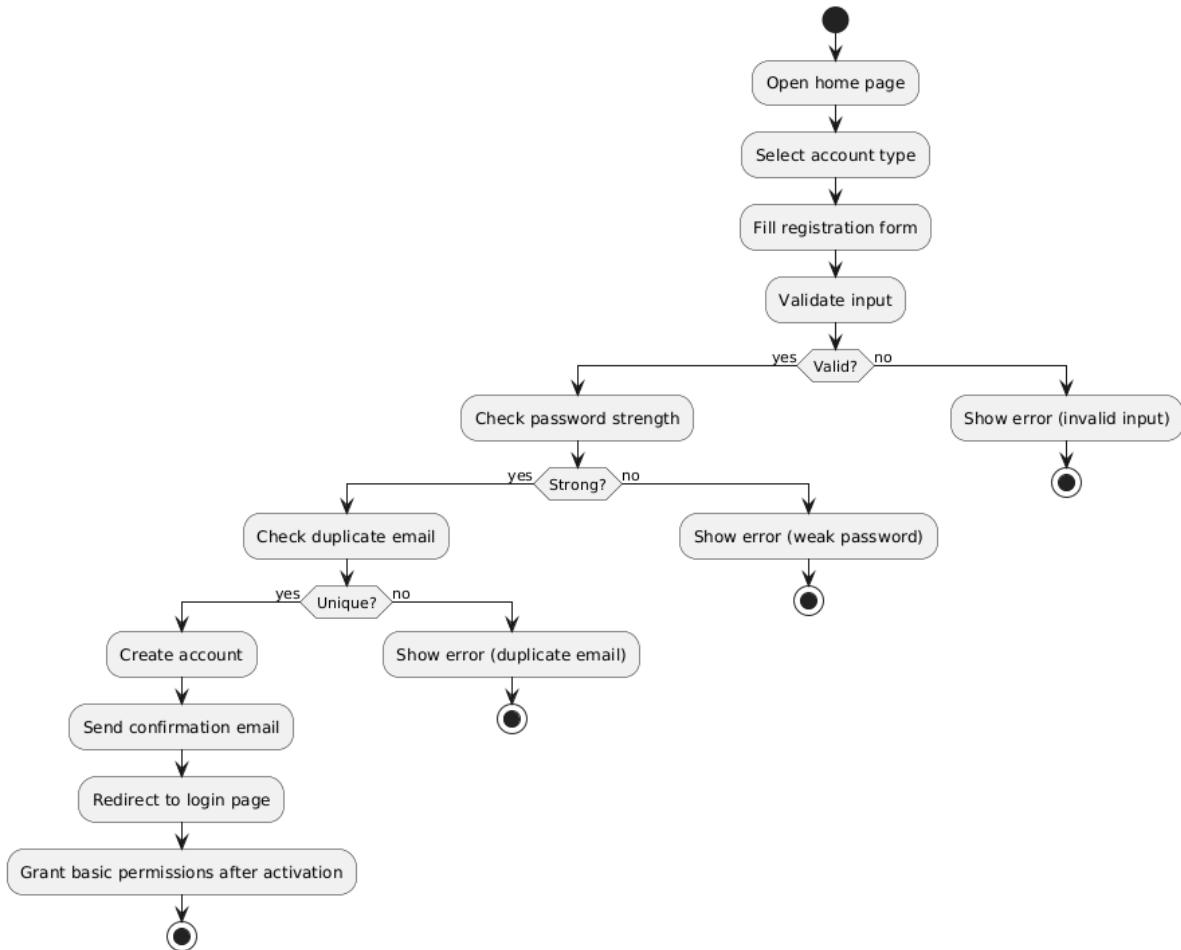


Figure 4.2 register (User Registration) activity diagram

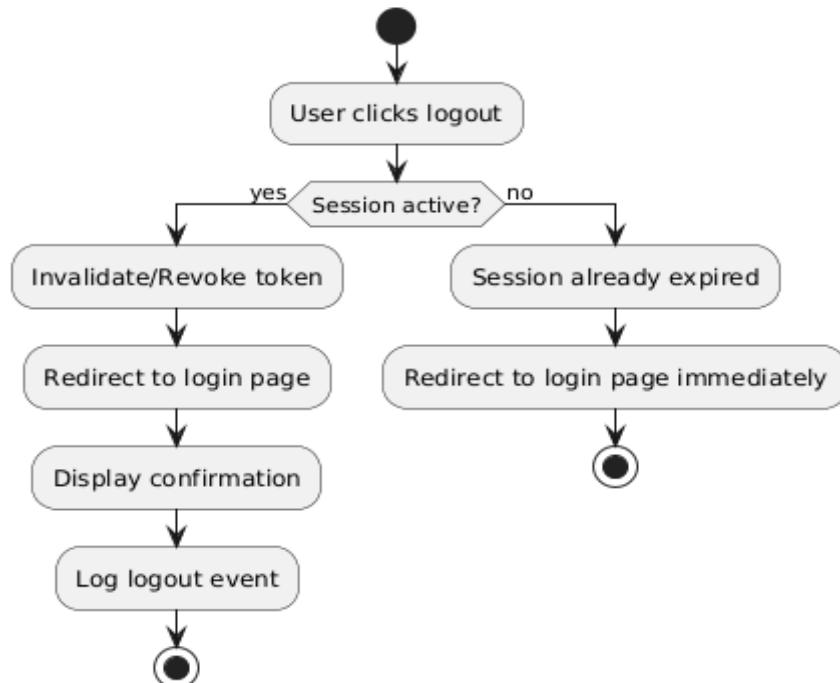


Figure 4.3 Register Logout activity diagram

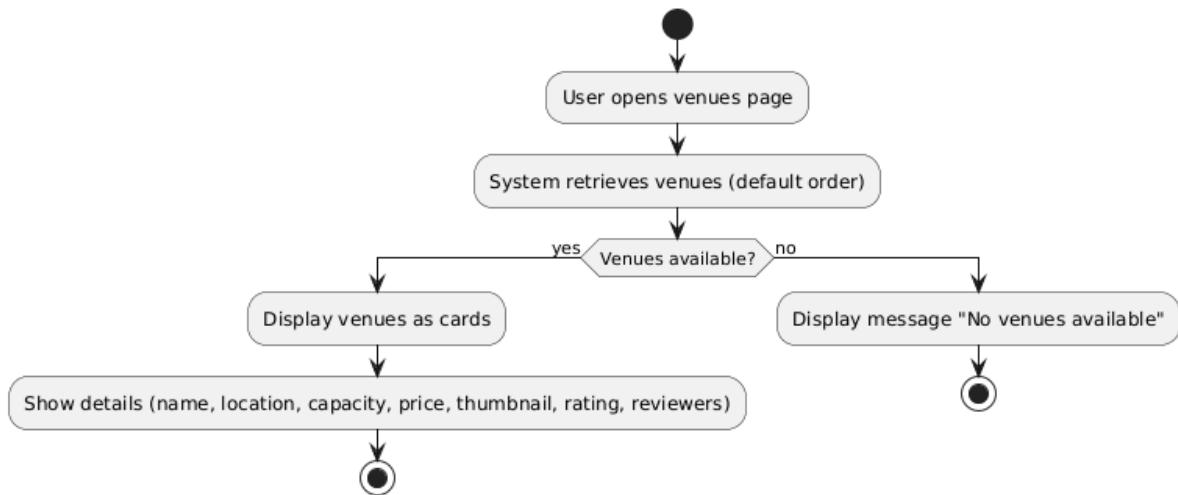


Figure 4.4 view venues activity diagram

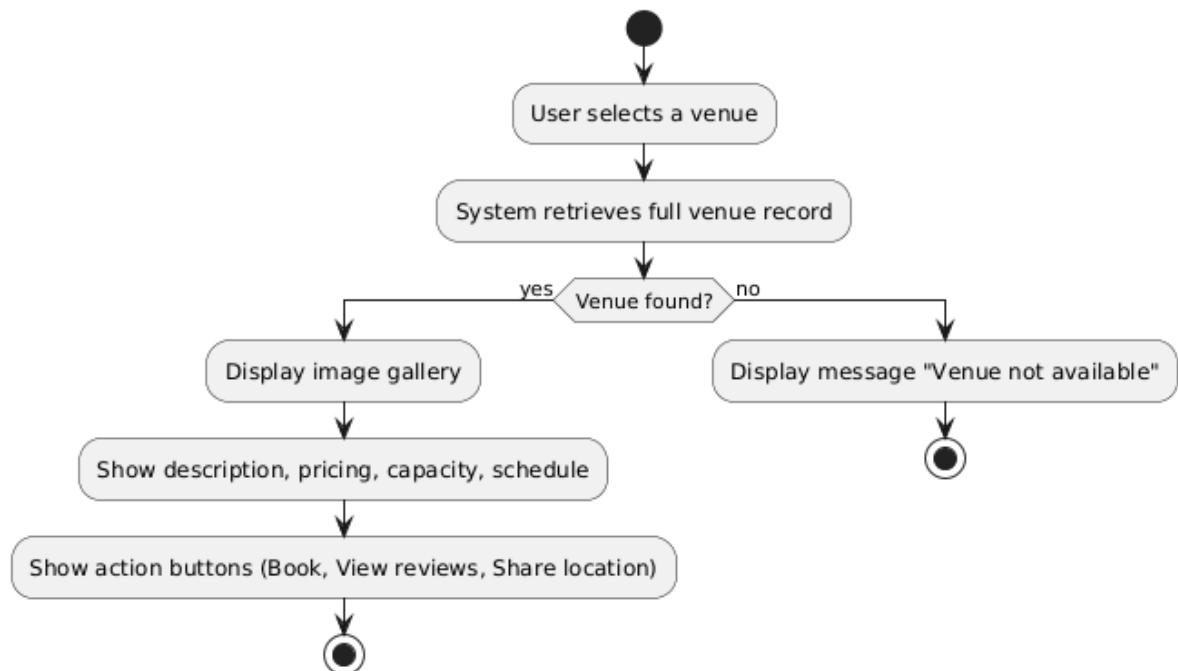


Figure 4.5 view venues details activity diagram

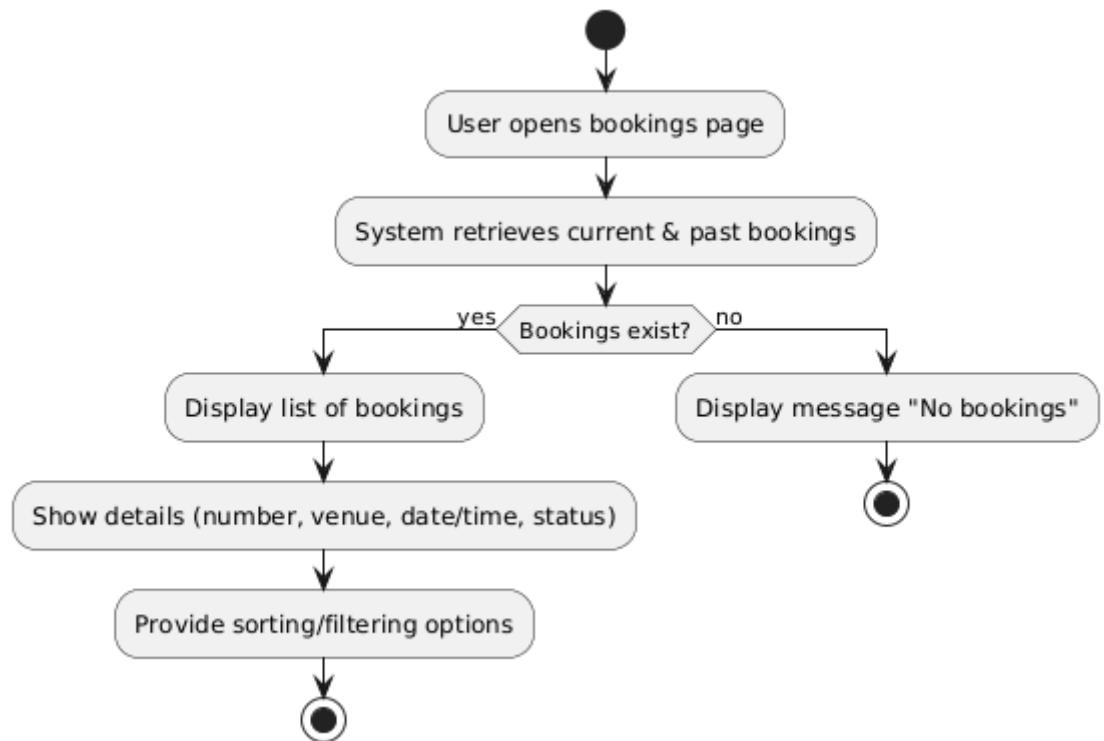


Figure 4.6 view bookings activity diagram

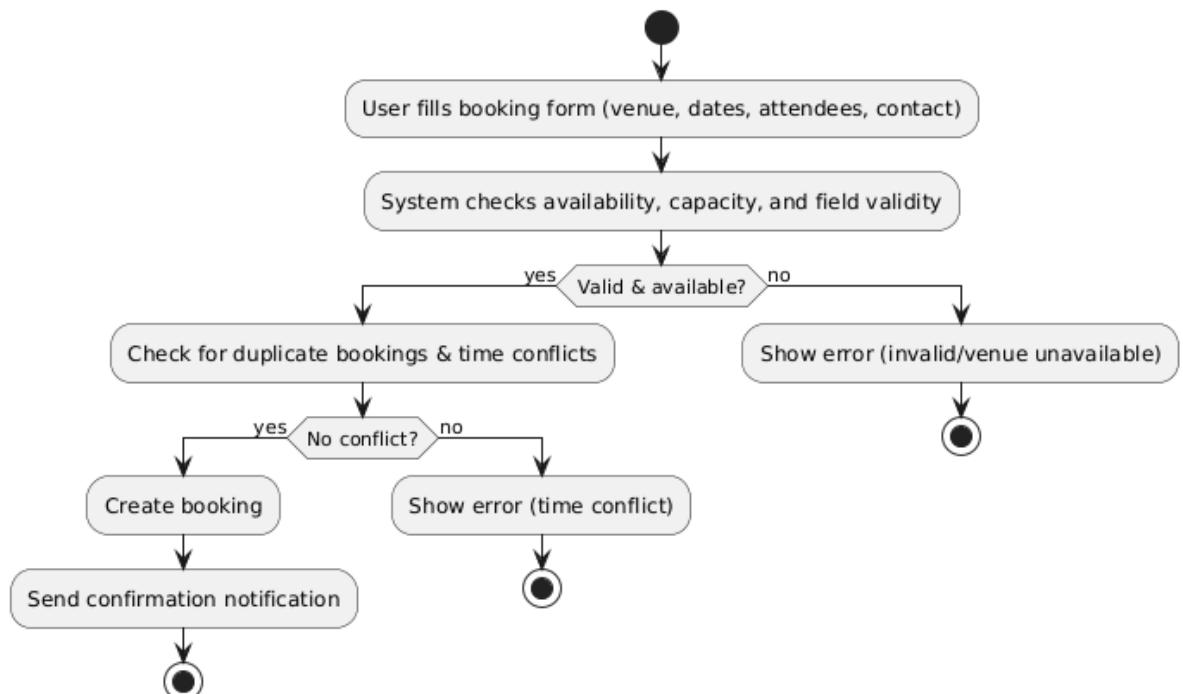


Figure 4.7 add booking activity diagram

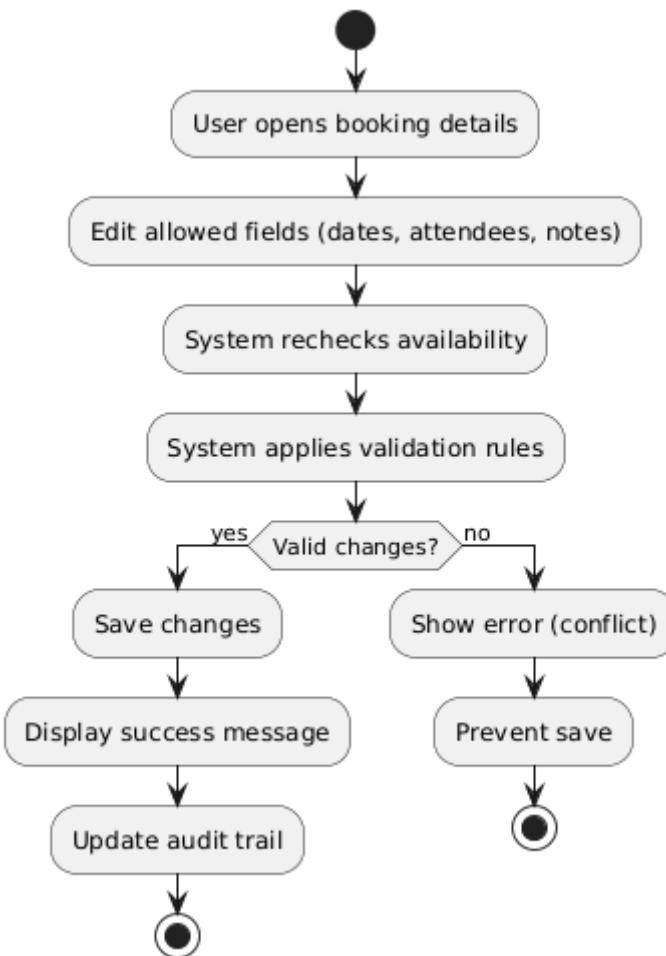


Figure 4.8 edit booking activity diagram

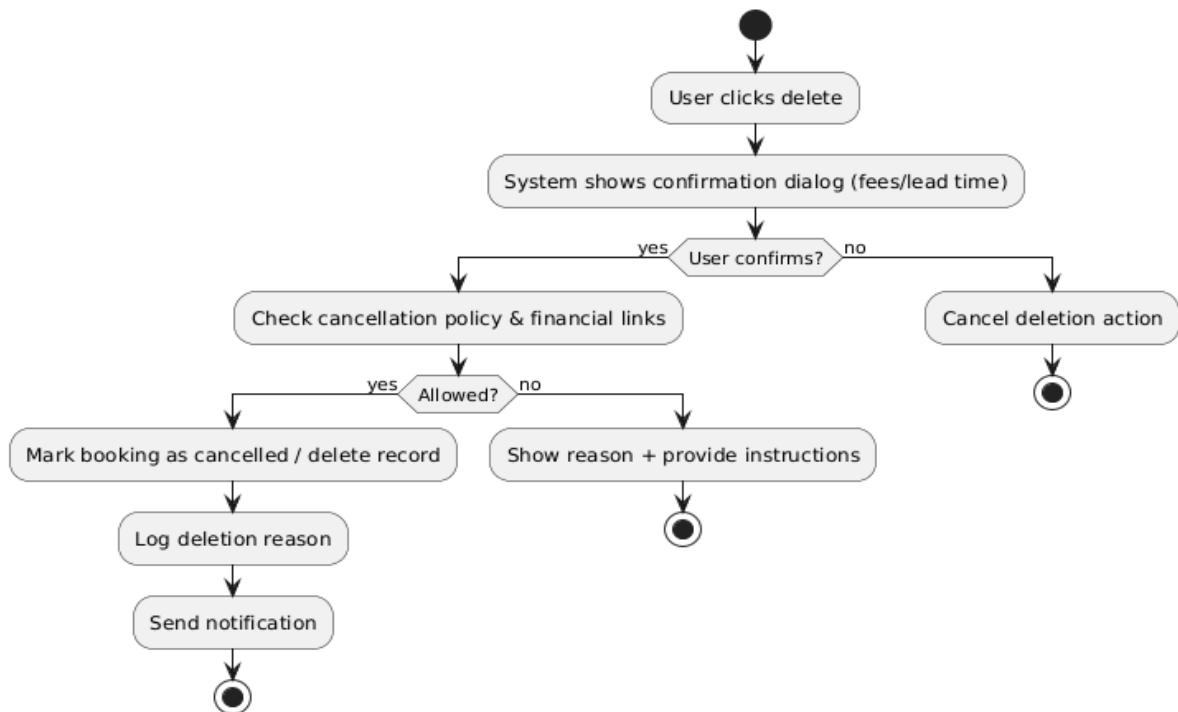


Figure 4.9 delete booking activity diagram

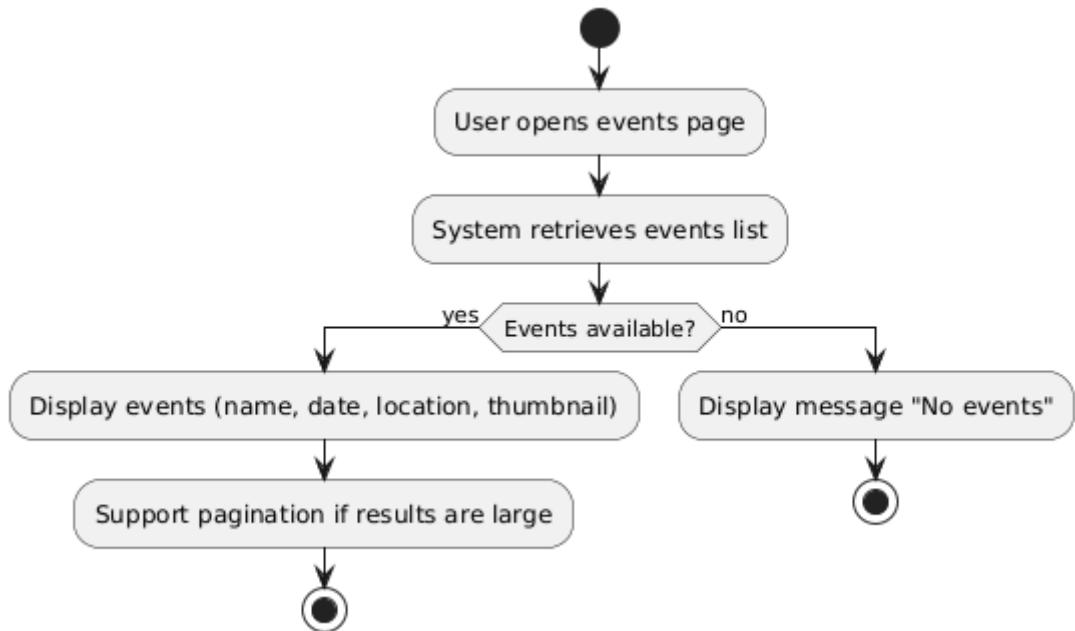


Figure 4.10 view events activity diagram

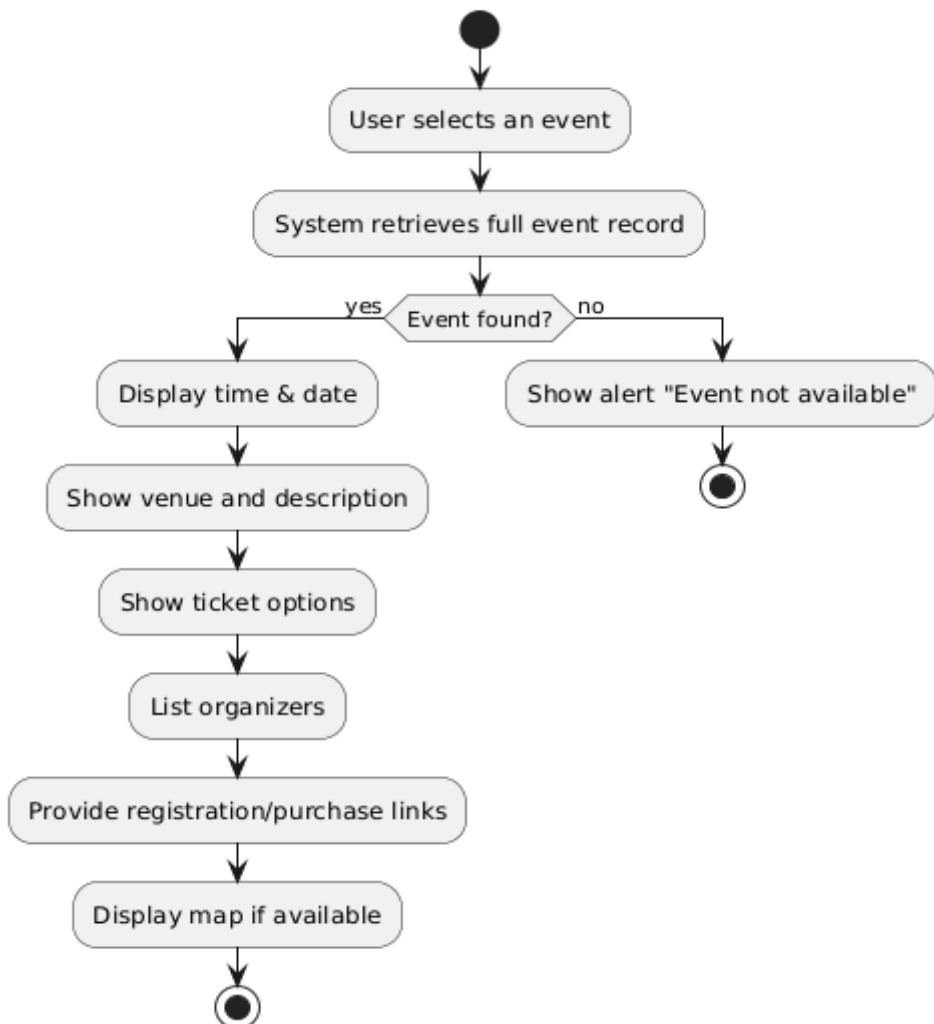


Figure 4.11 view events details activity diagram

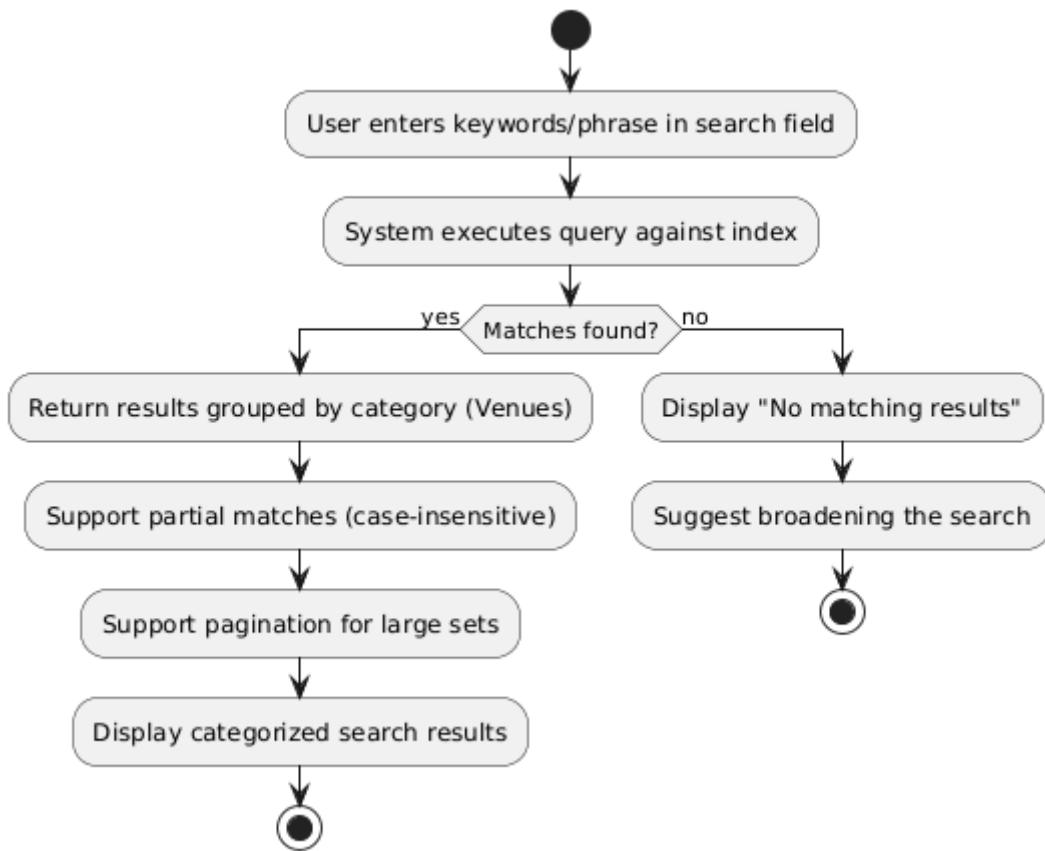


Figure 4.12 unified search for venues activity diagram

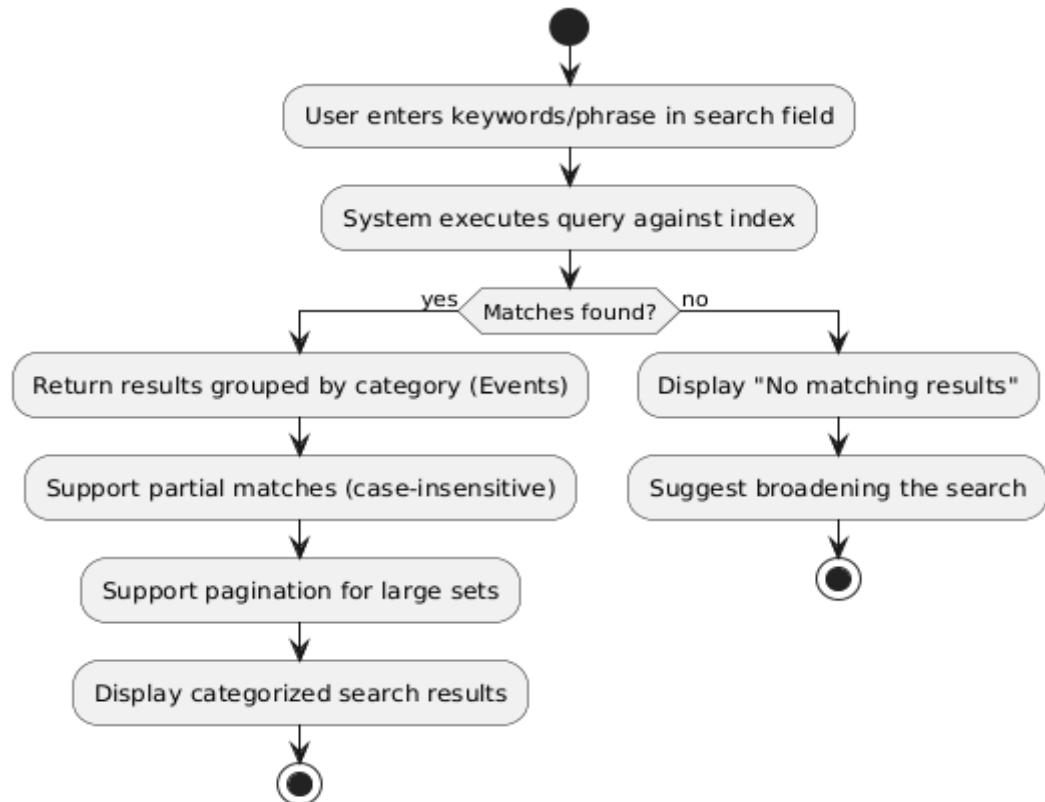


Figure 4.13 unified search for events activity diagram

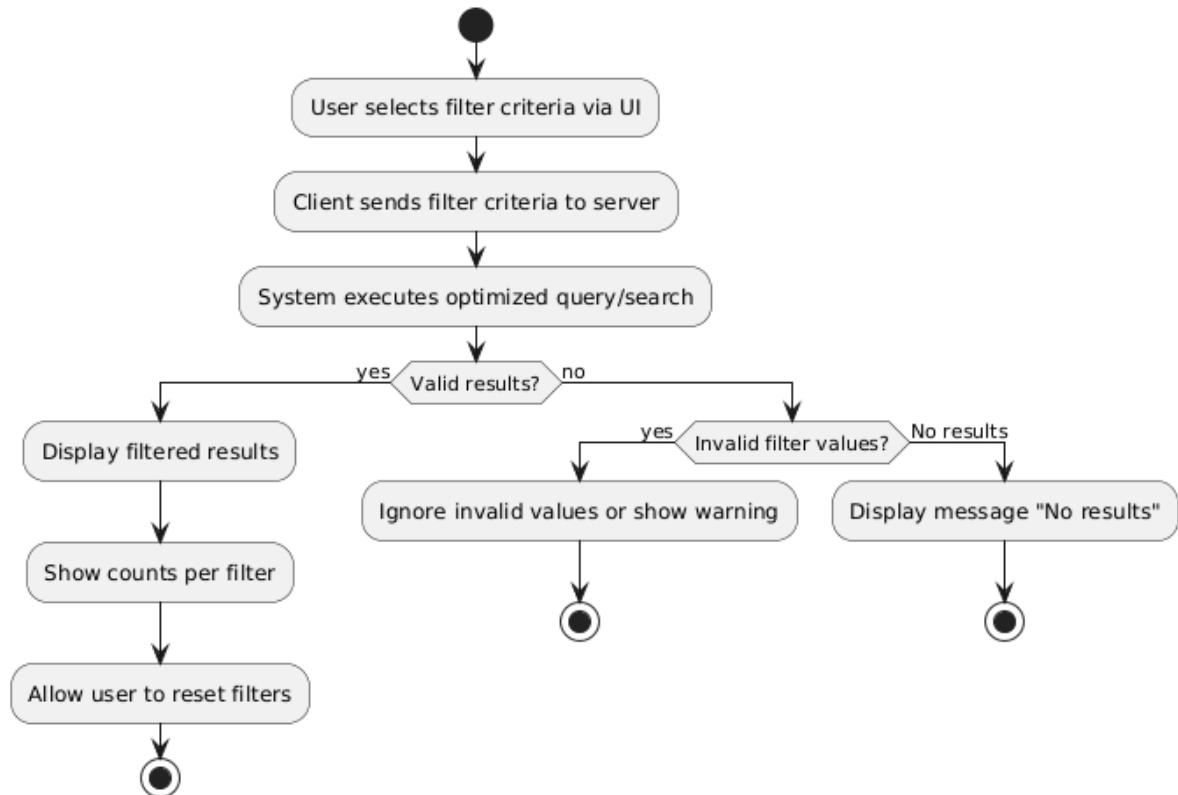


Figure 4.14 filter venues by specific criteria activity diagram

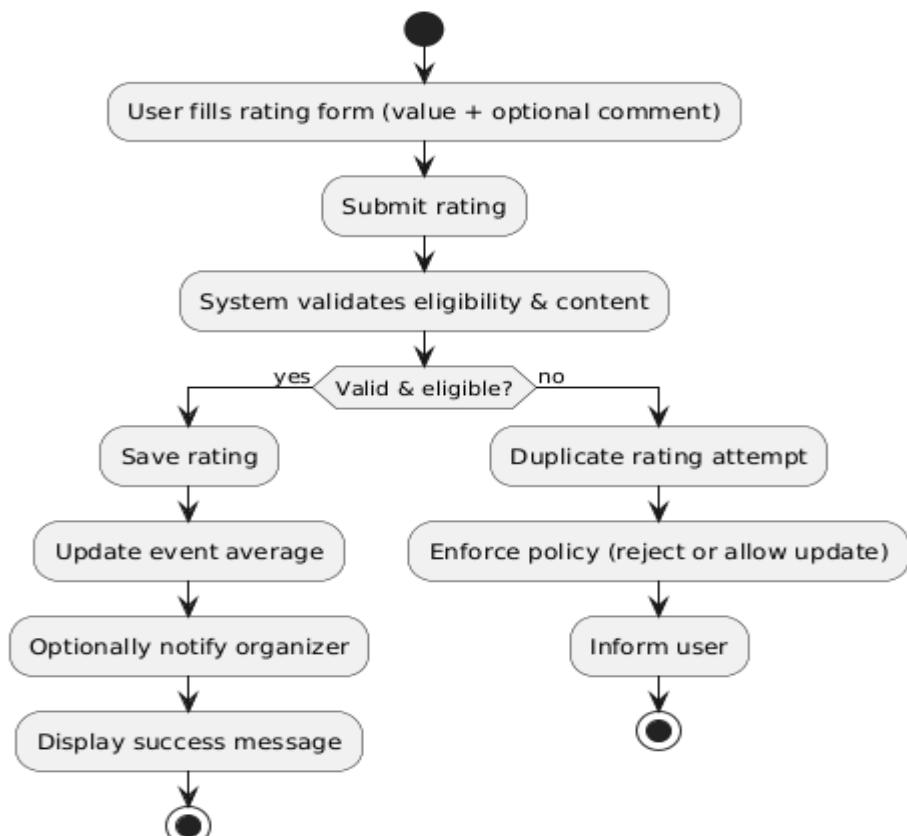


Figure 4.15 filter events by specific criteria activity diagram

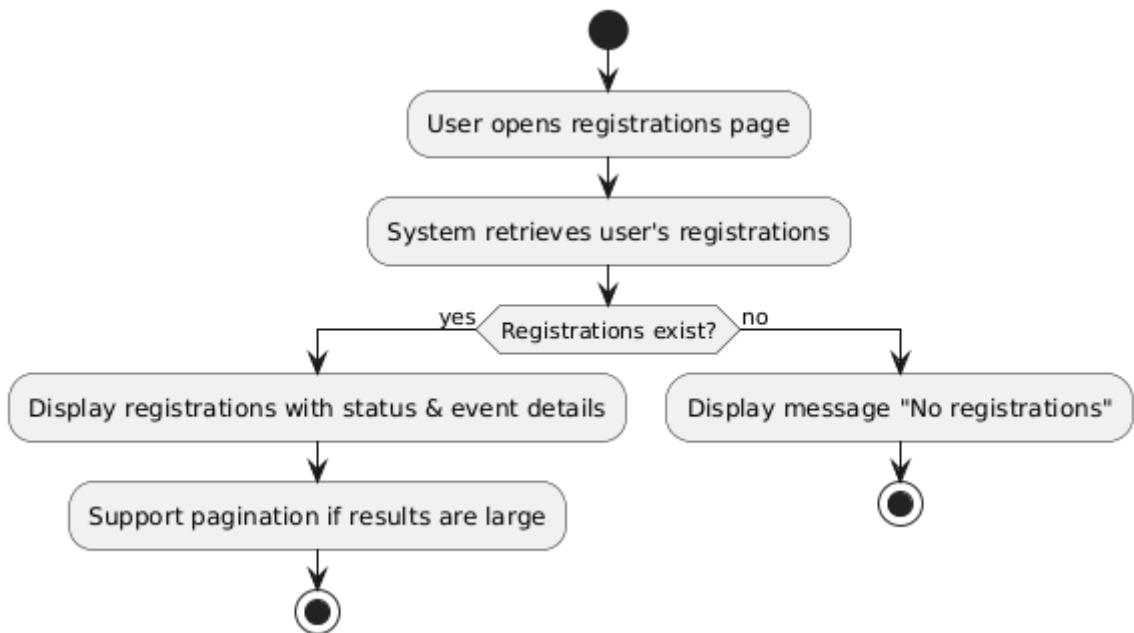


Figure 4.16 view registration activity diagram

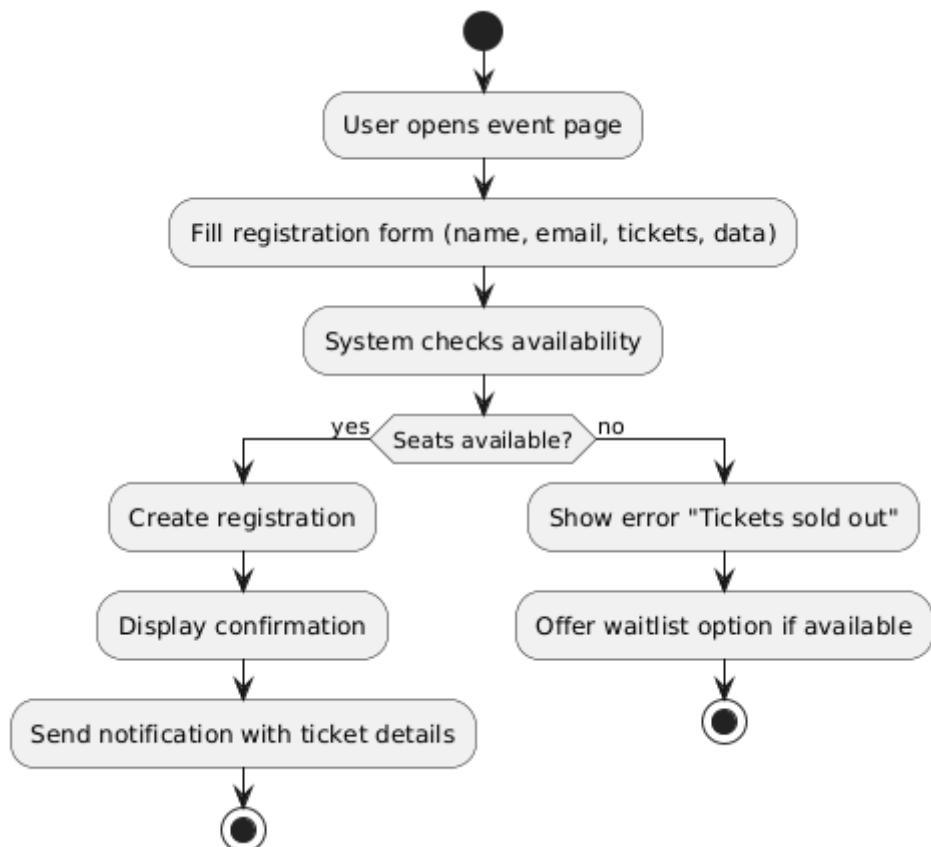


Figure 4.17 add registration activity diagram

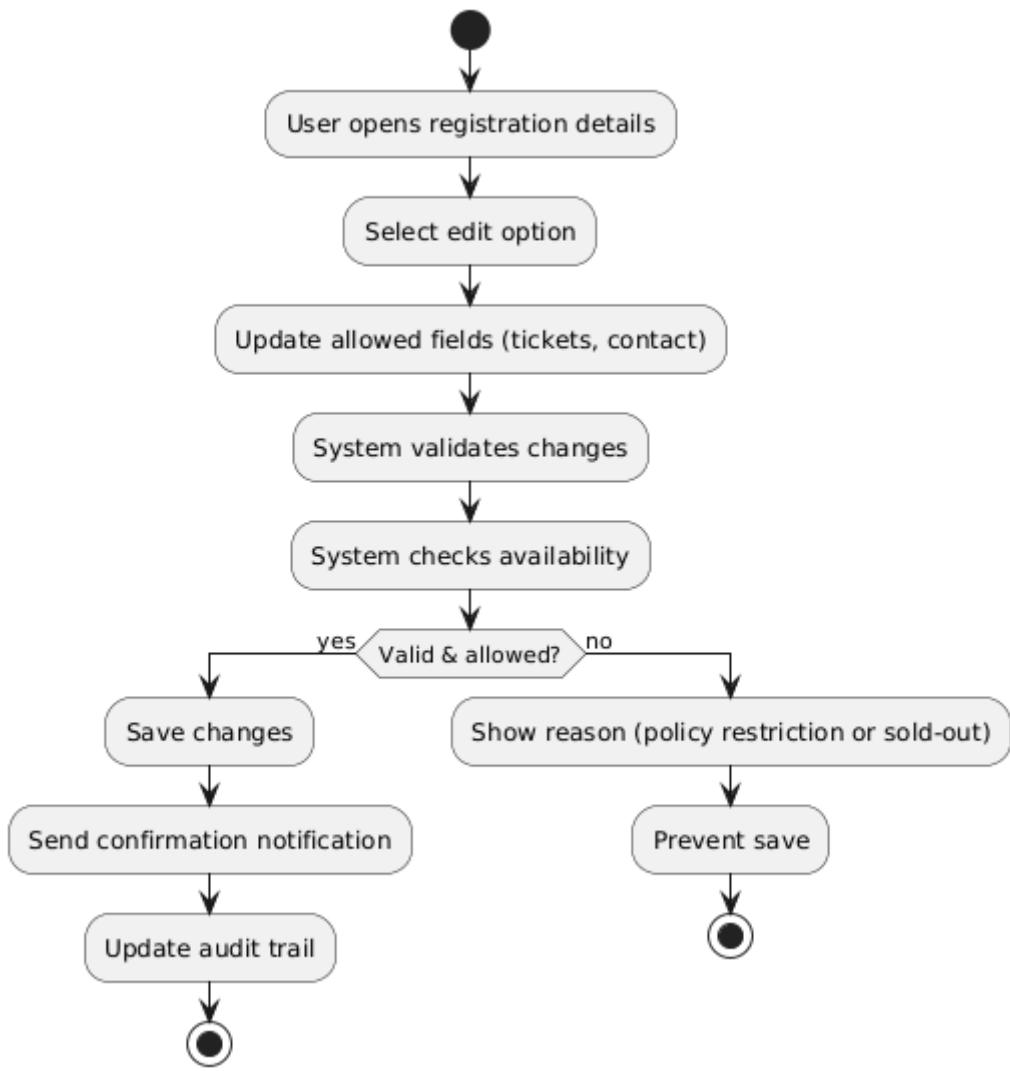


Figure 4.18 edit registration activity diagram

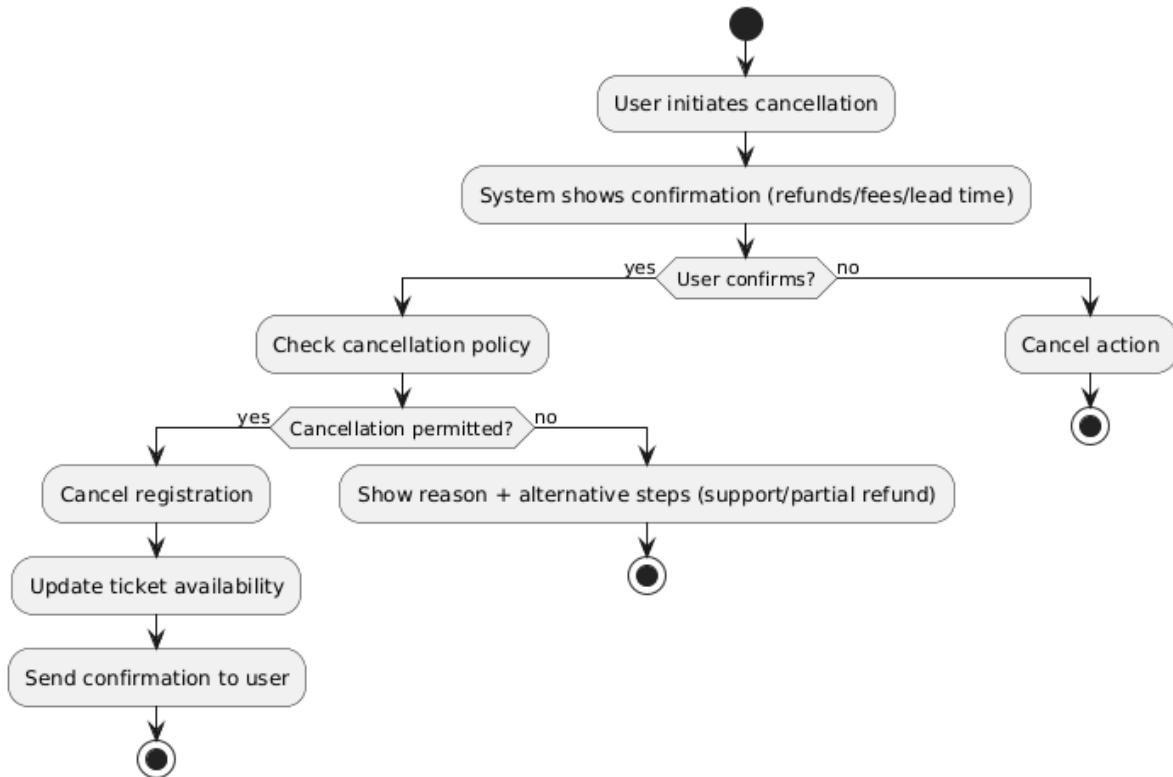


Figure 4.19 delete registration activity diagram

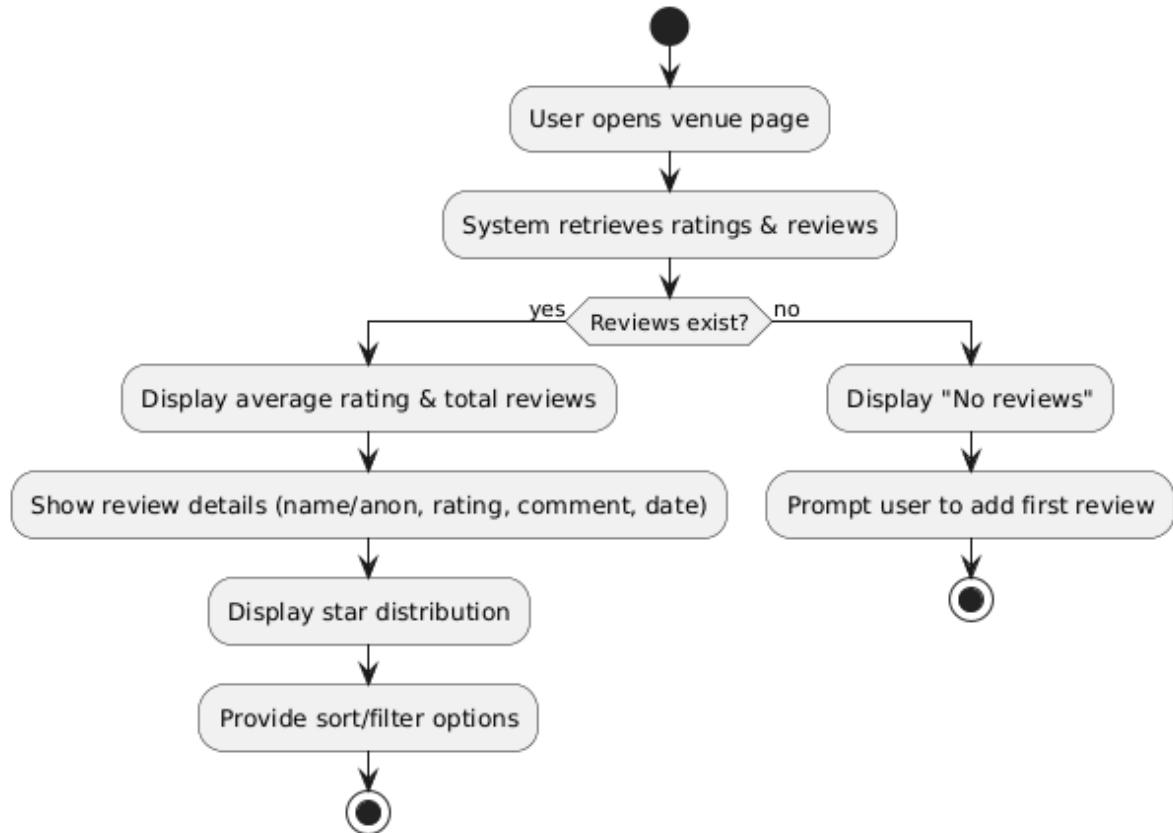


Figure 4.20 view venue ratings activity diagram

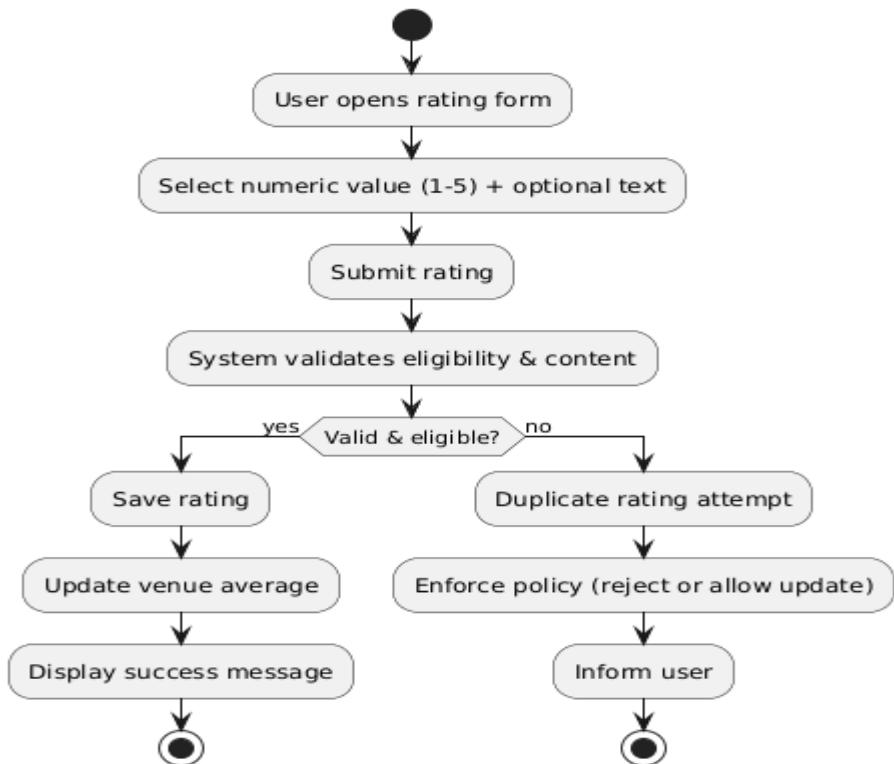


Figure 4.21 add venue rating activity diagram

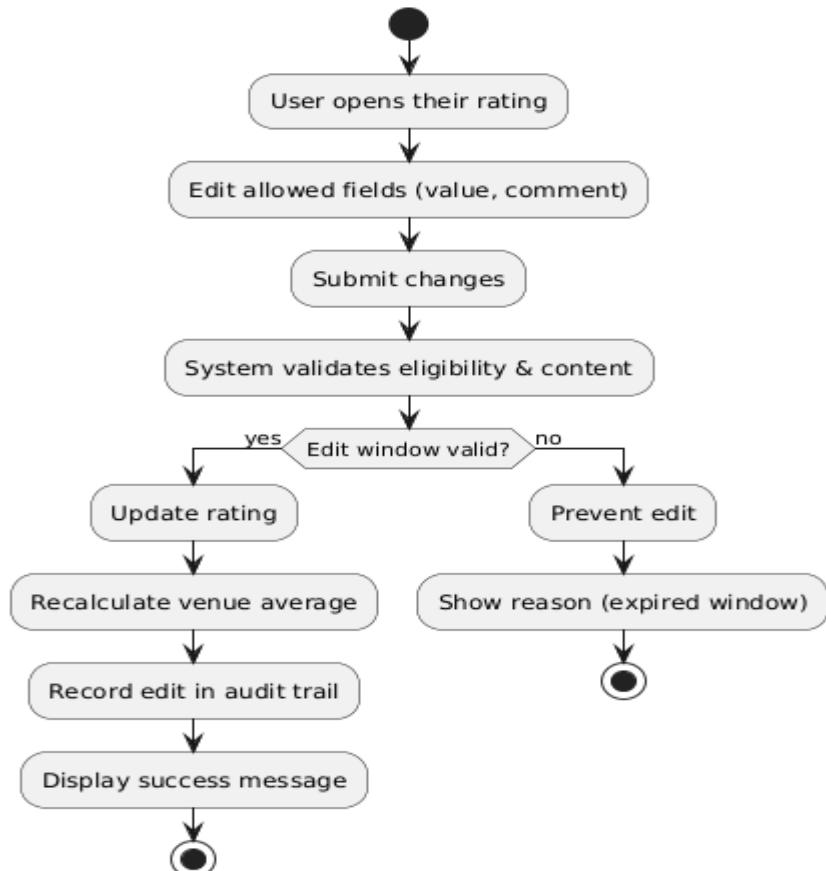


Figure 4.22 edit venue rating activity diagram

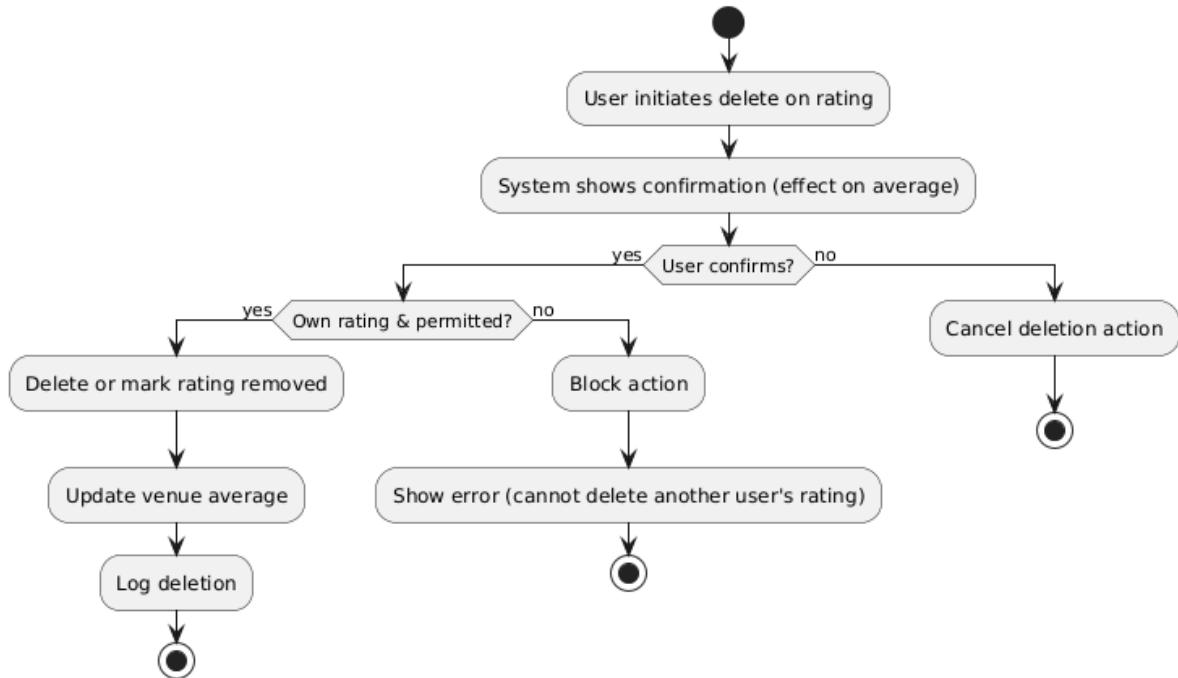


Figure 4.23 delete venue rating activity diagram

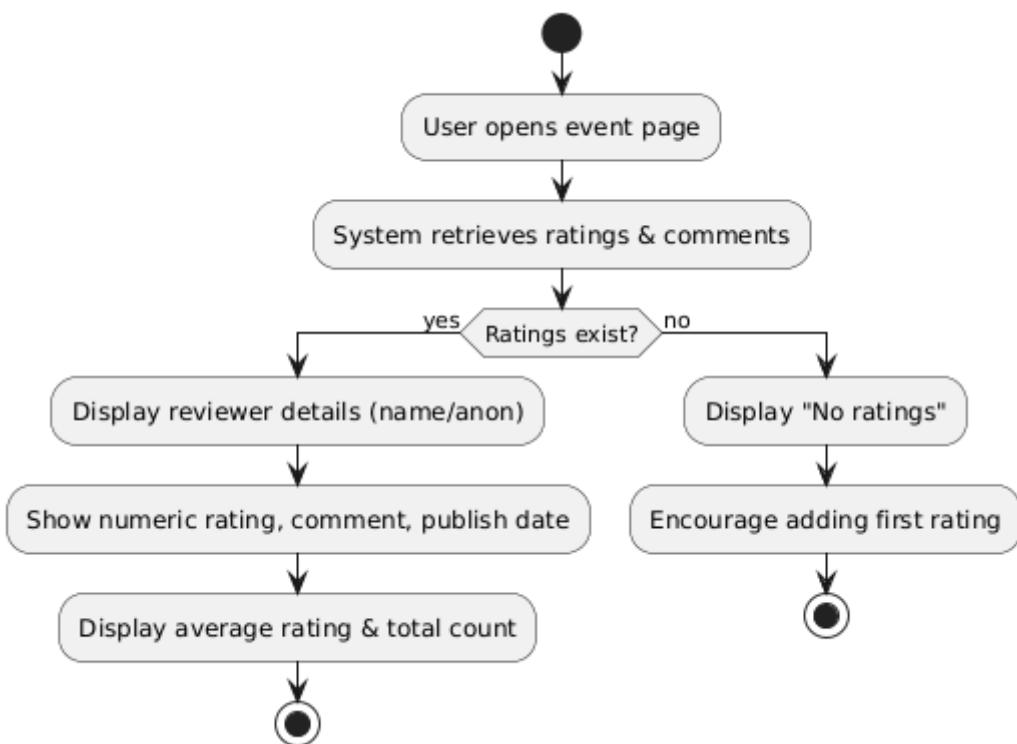


Figure 4.24 view event ratings activity diagram

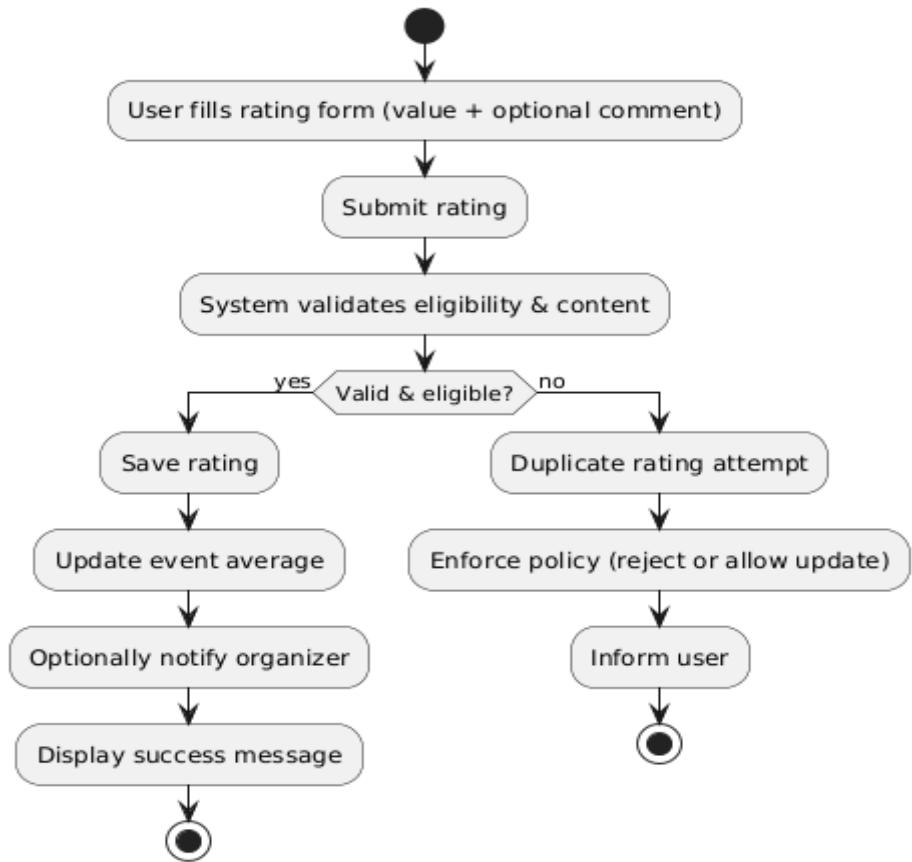


Figure 4.25 add event rating activity diagram

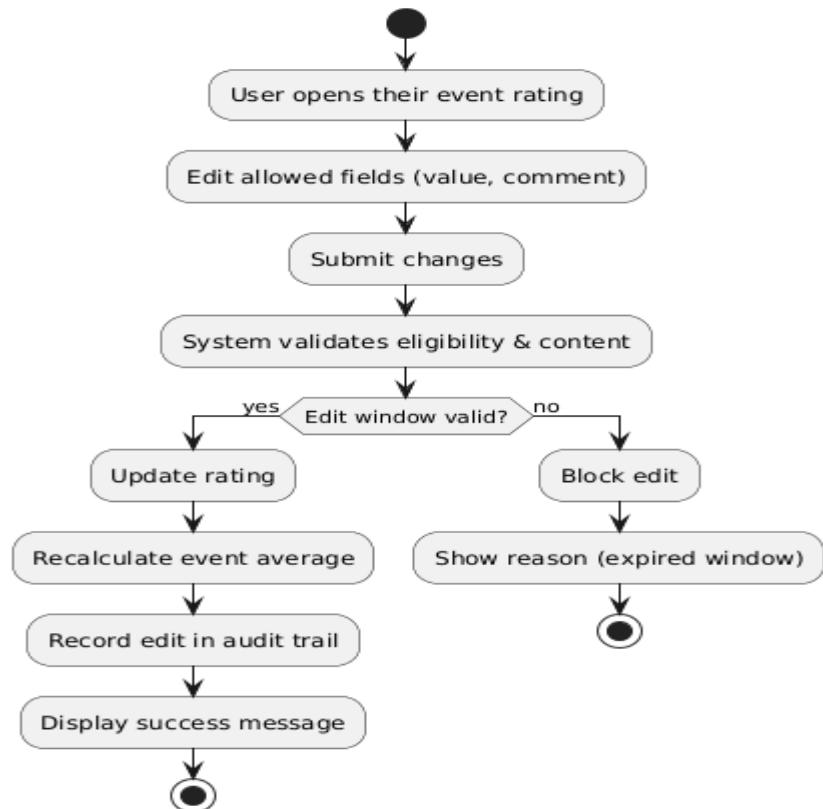


Figure 4.26 edit event rating activity diagram

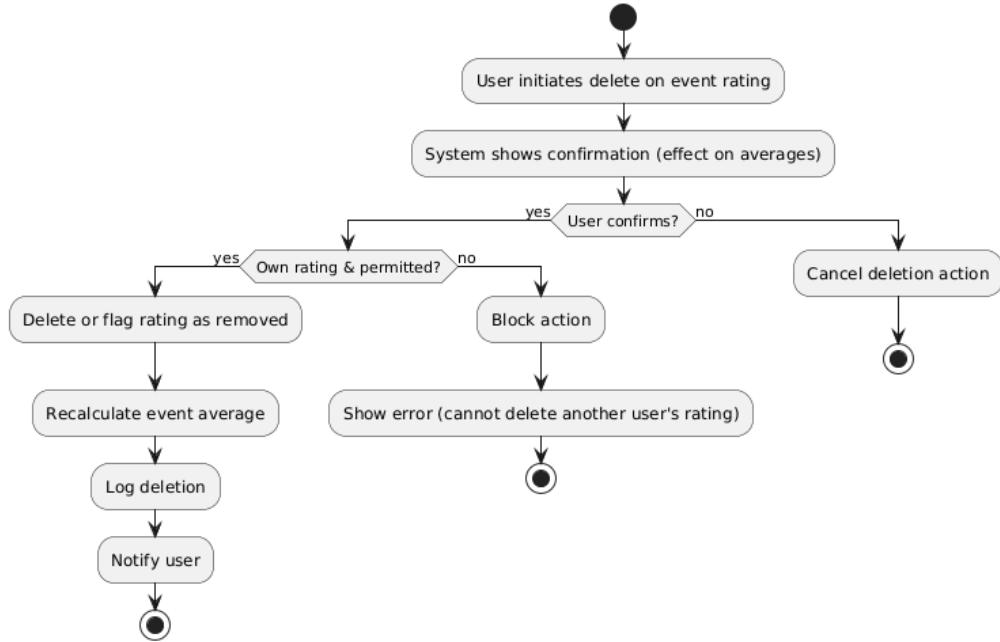


Figure 4.27 delete event rating activity diagram

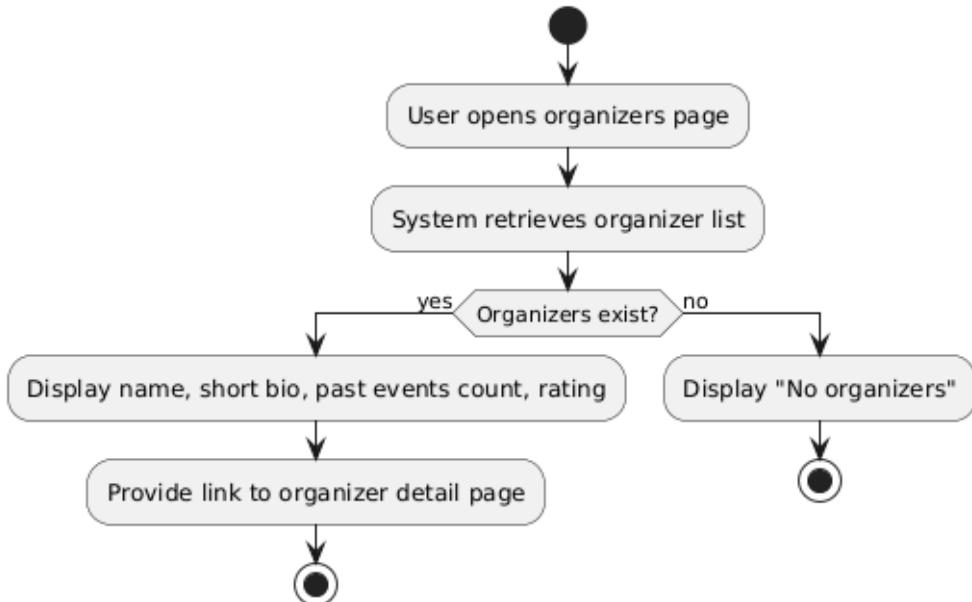


Figure 4.28 view organizers activity diagram

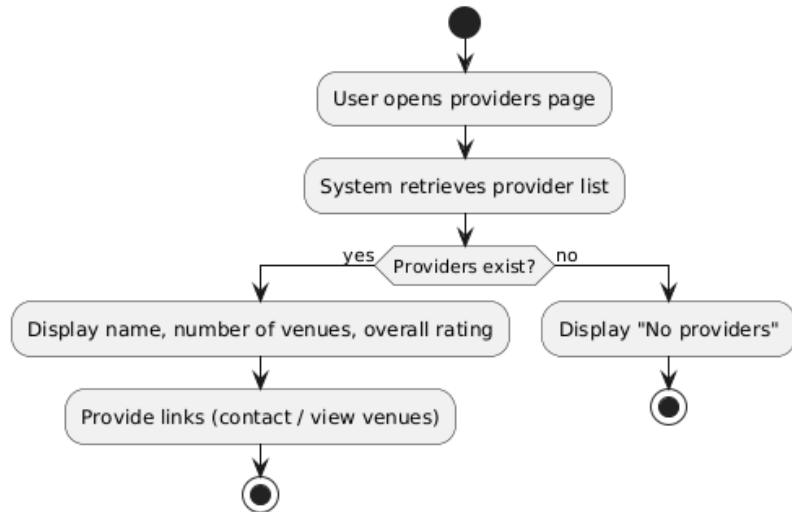


Figure 4.29 view providers activity diagram

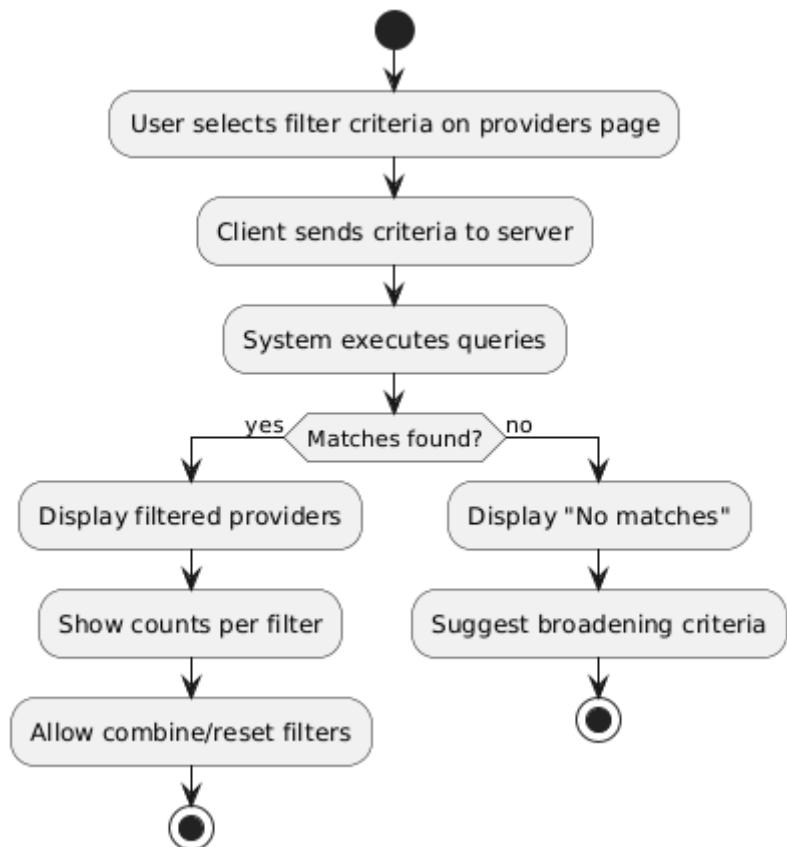


Figure 4.30 filter providers by specific criteria activity diagram

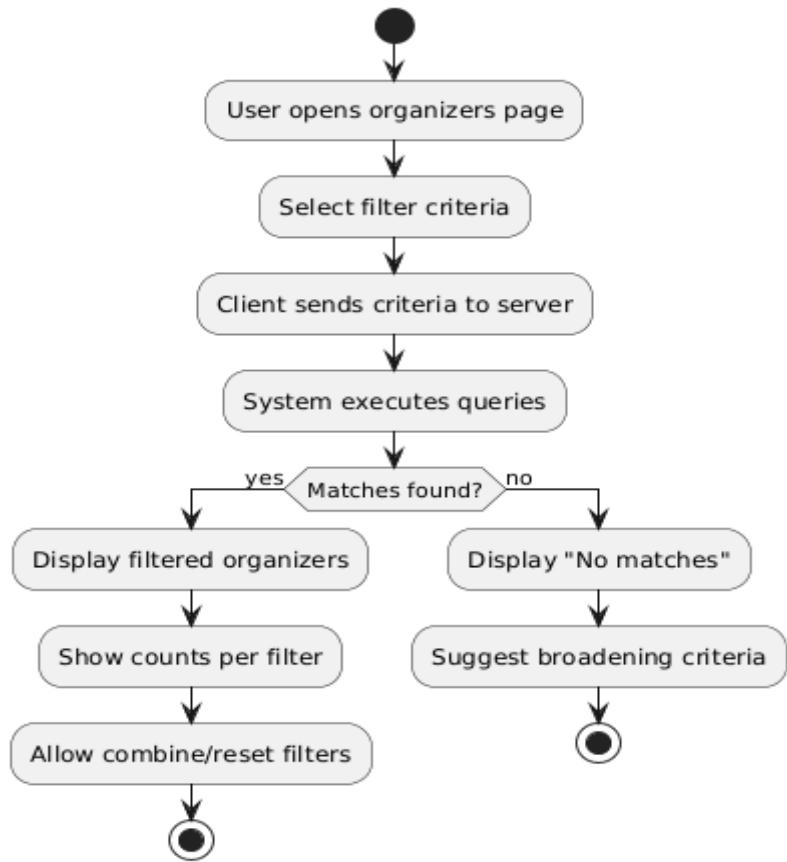


Figure 4.31 filter Organizers by specific criteria activity diagram

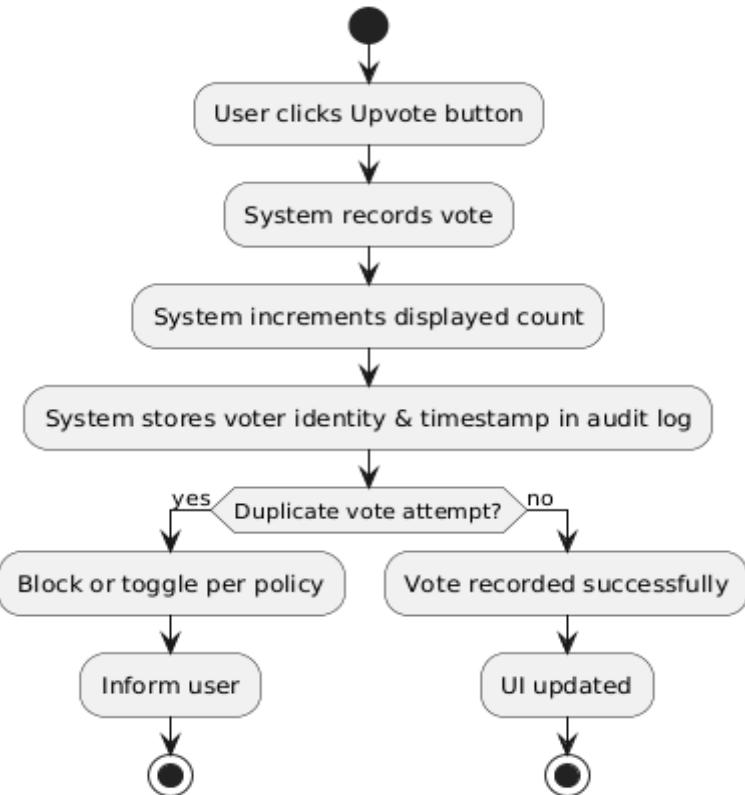


Figure 4.32 upvote activity diagram

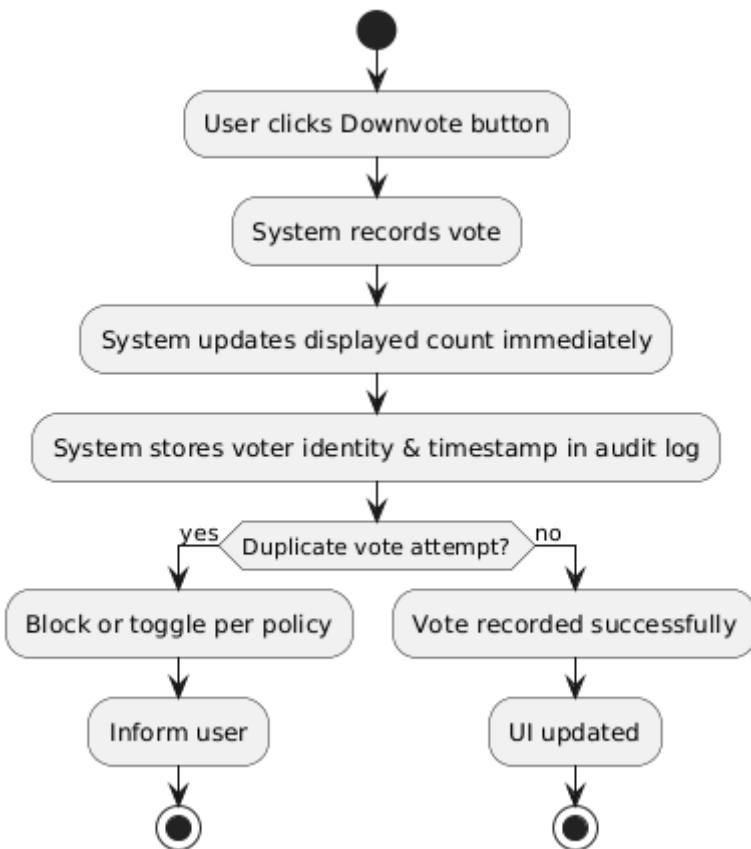


Figure 4.33 downvote activity diagram

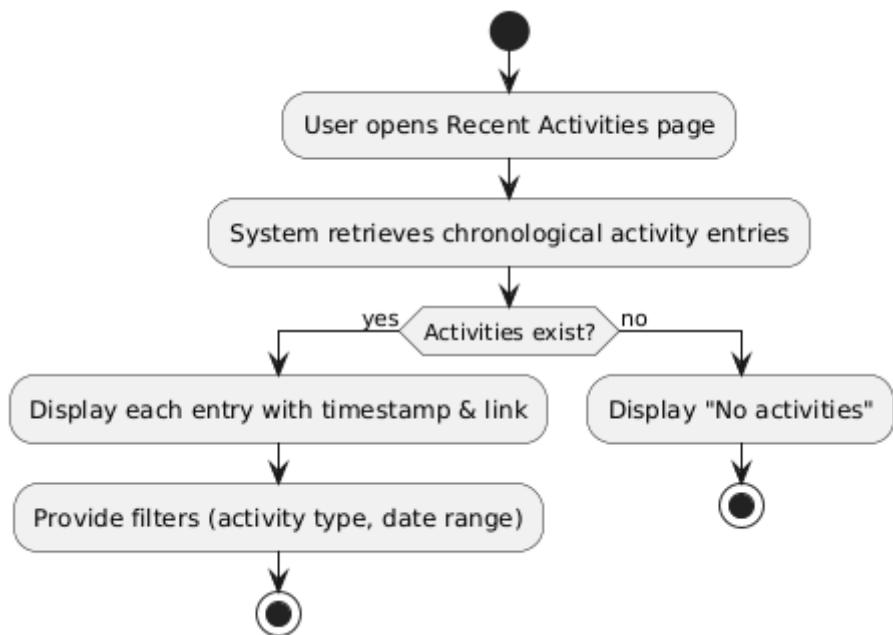


Figure 4.34 view recent activities activity diagram

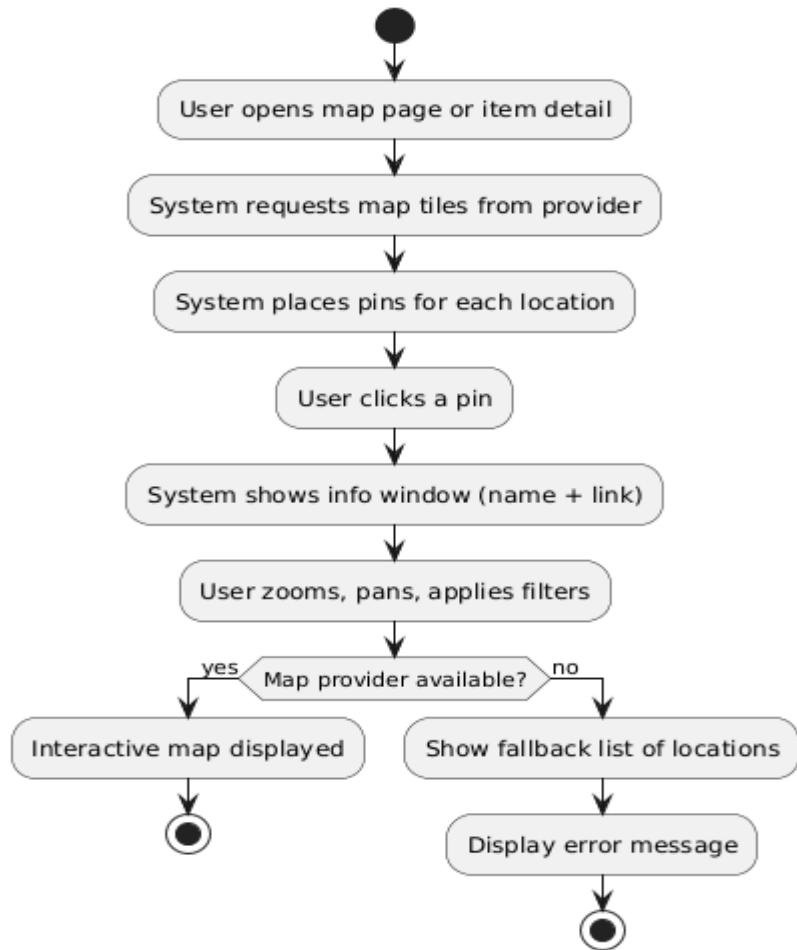


Figure 4.35 display map of event and venue locations activity diagram

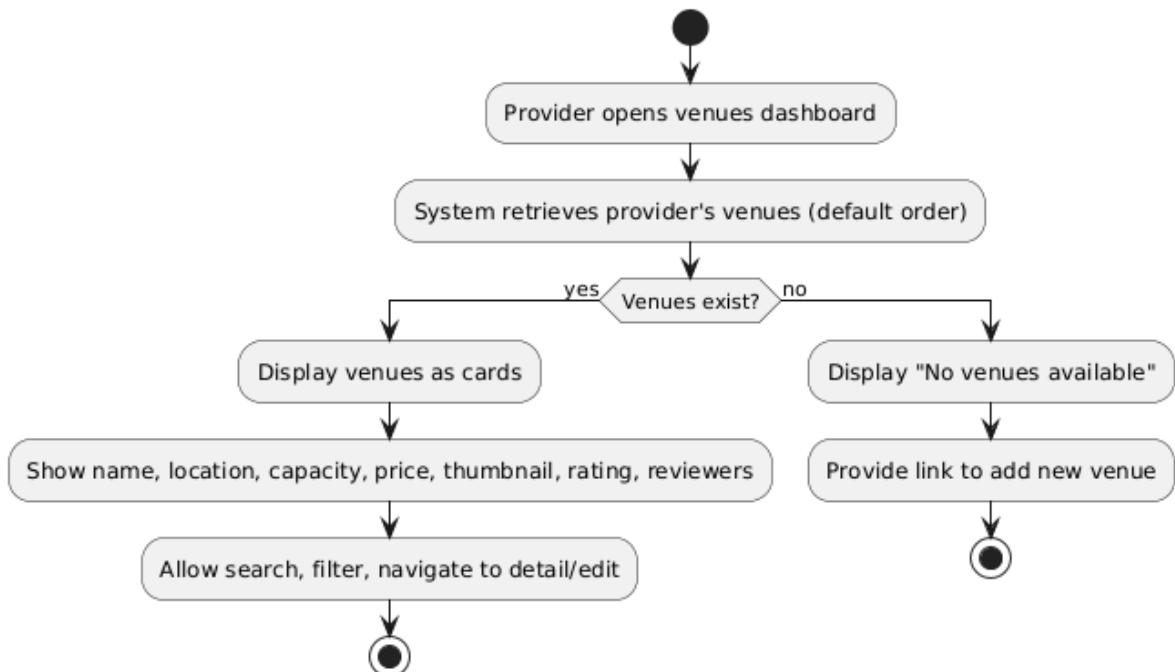


Figure 4.36 browse venues (provider) activity diagram

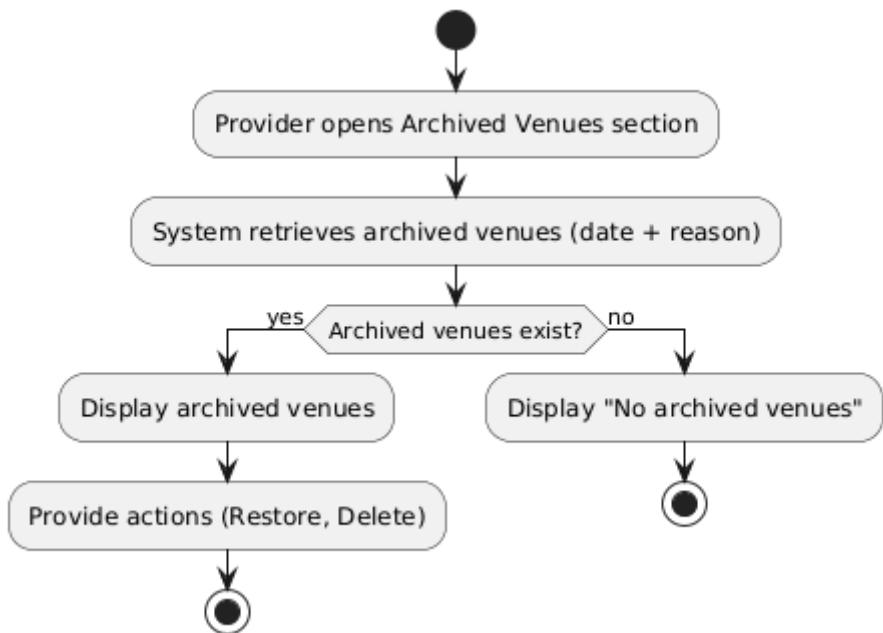


Figure 4.37 brows archived venues activity diagram

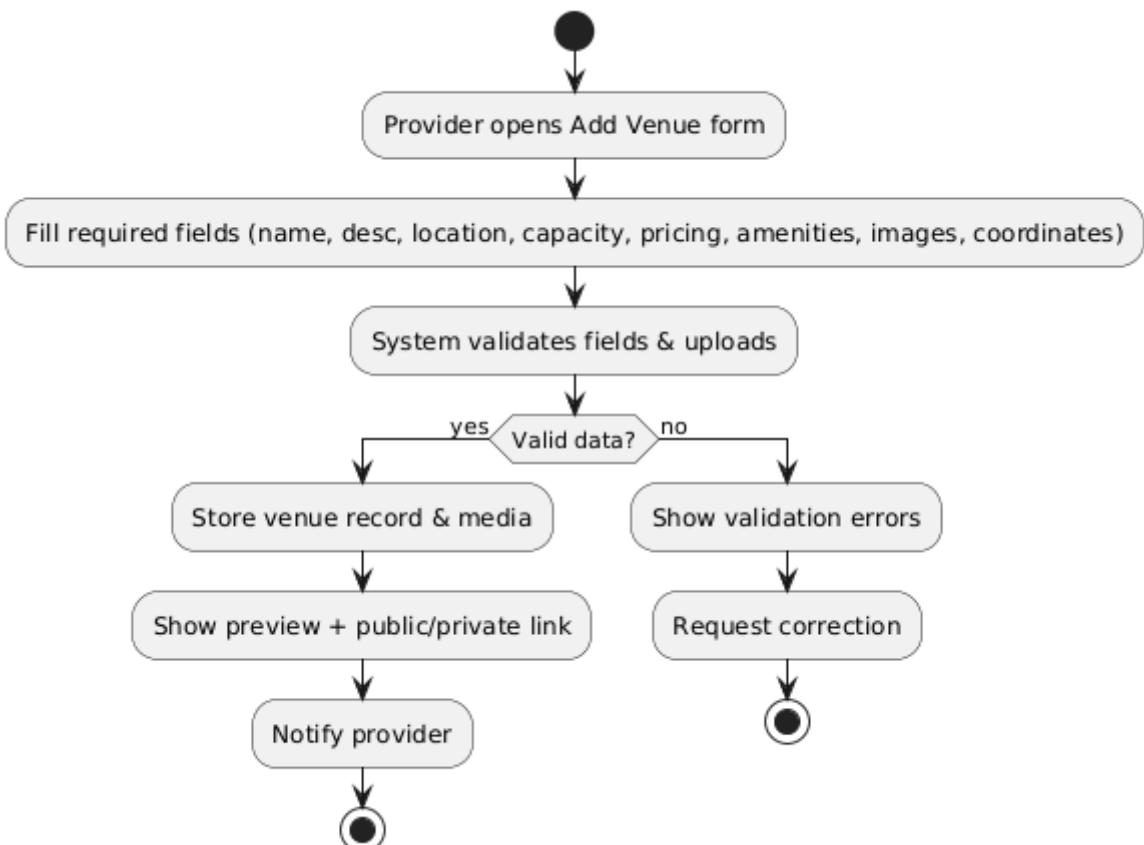


Figure 4.38 add venue activity diagram.

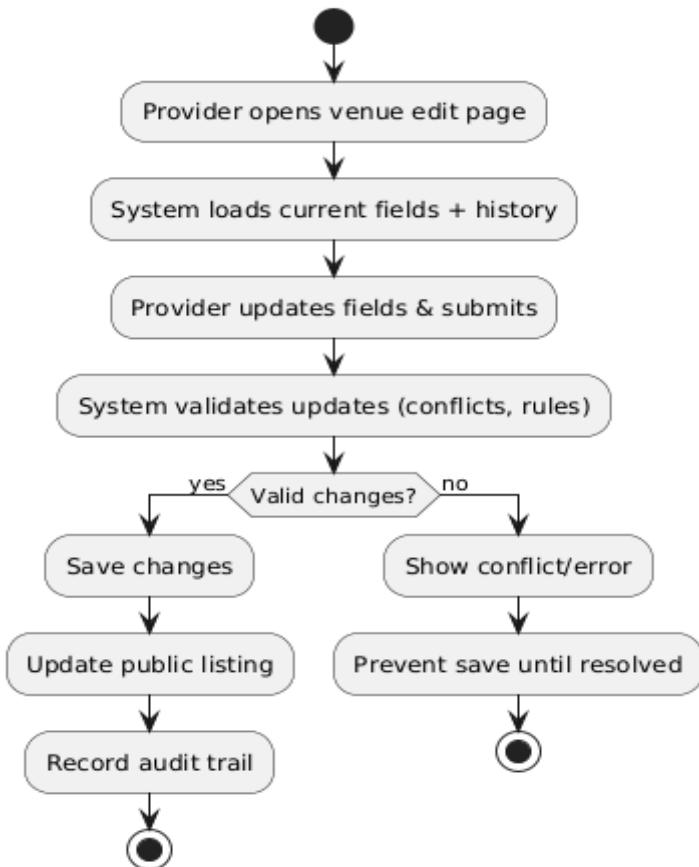


Figure 4.39 edit venue activity diagram

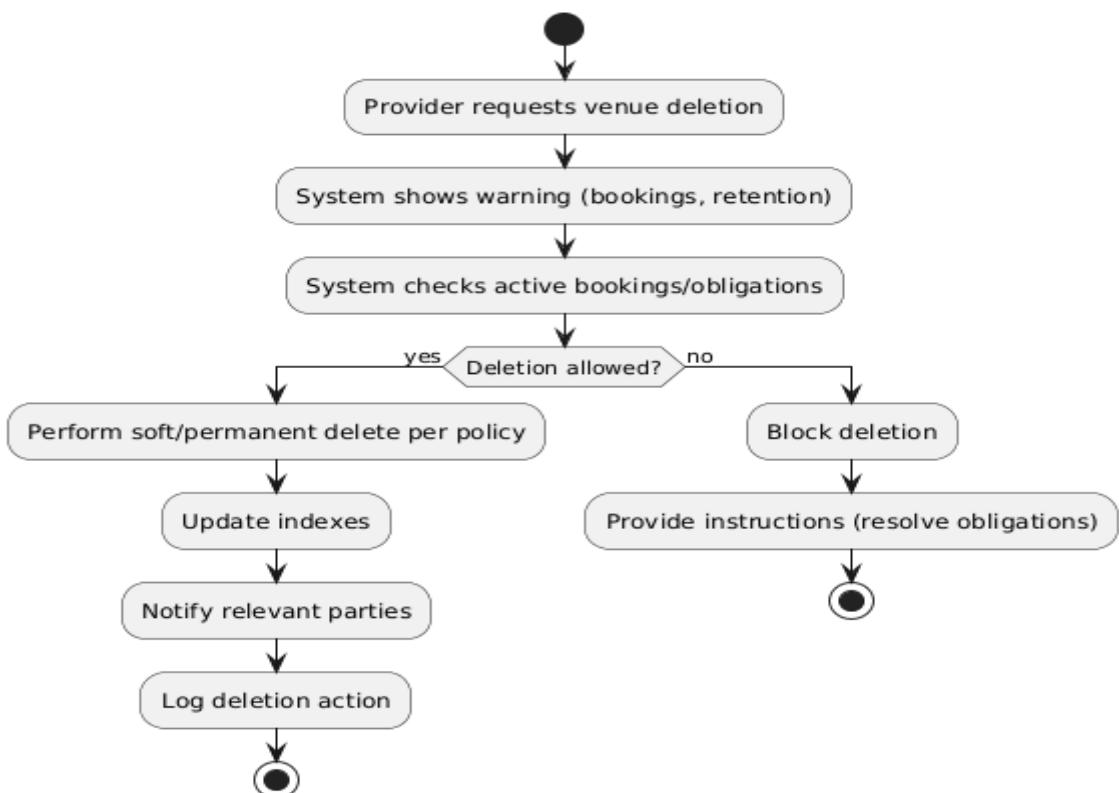


Figure 4.40 delete venue activity diagram

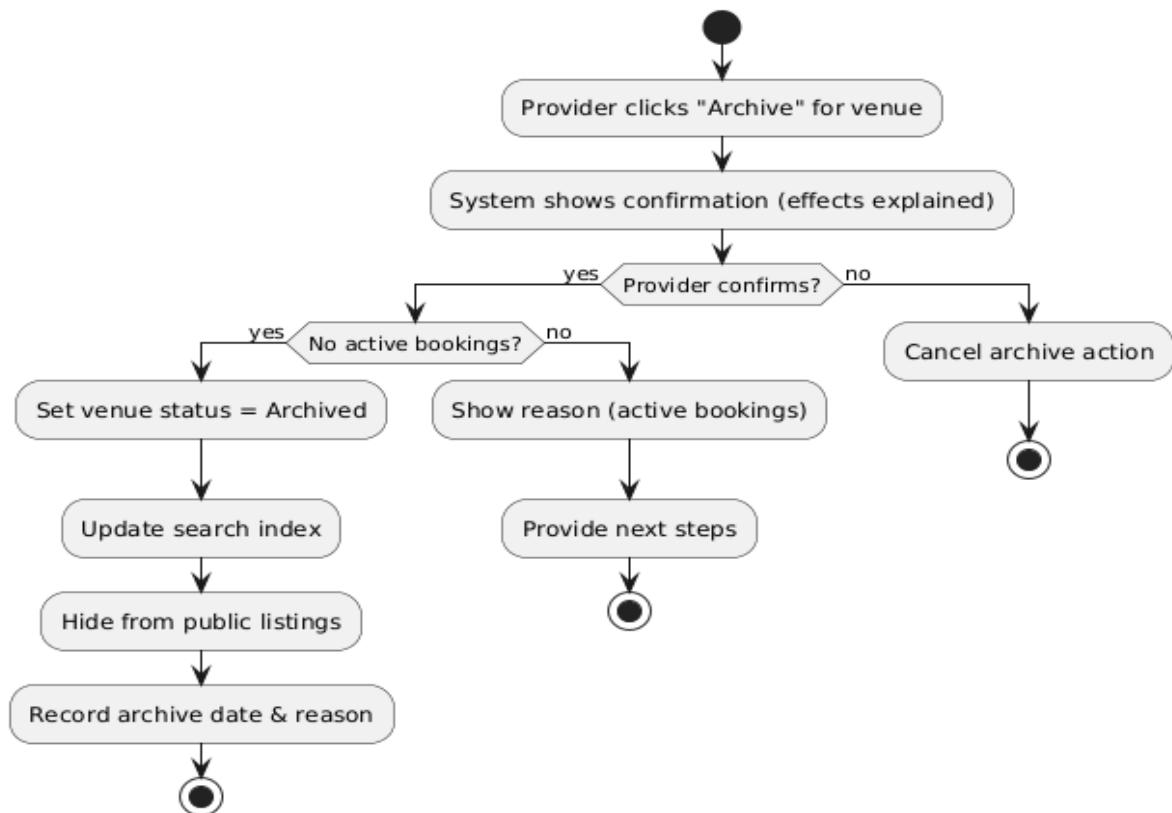


Figure 4.41 archive venue activity diagram

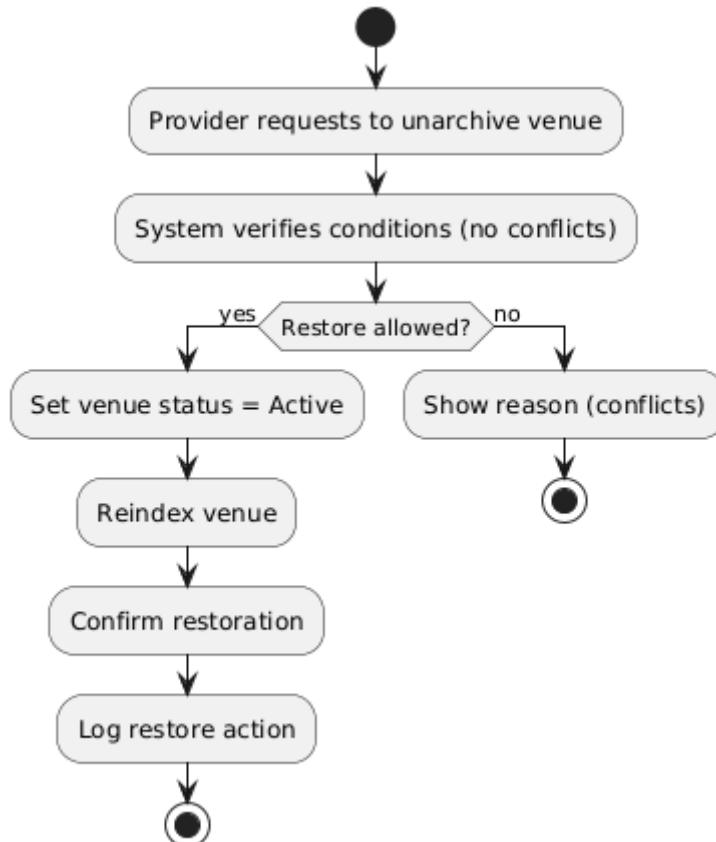


Figure 4.42 unarchive venue activity diagram

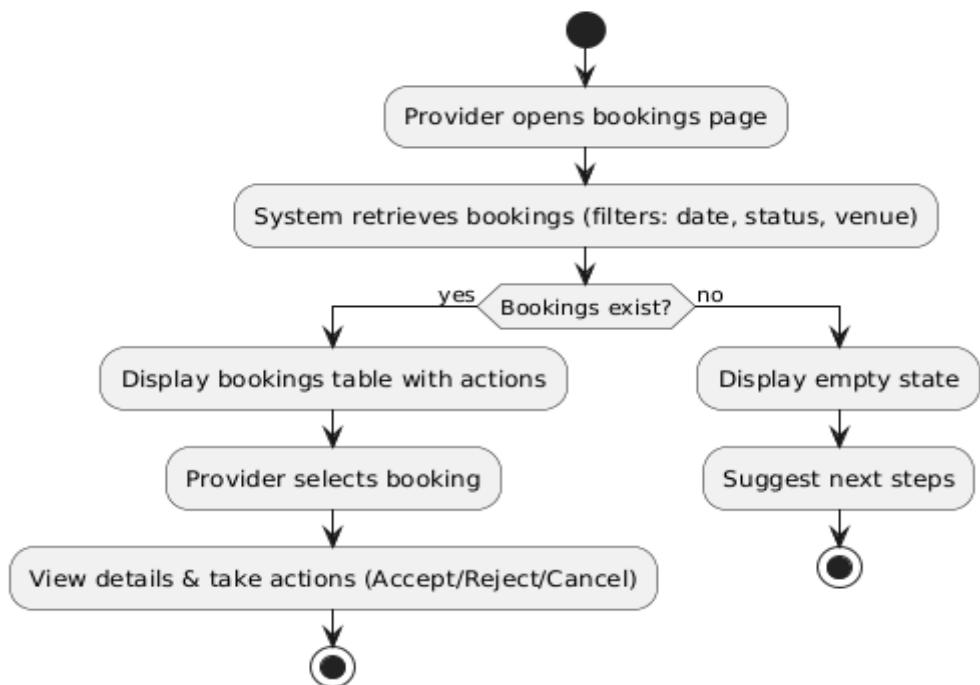


Figure 4.43 browse bookings activity diagram

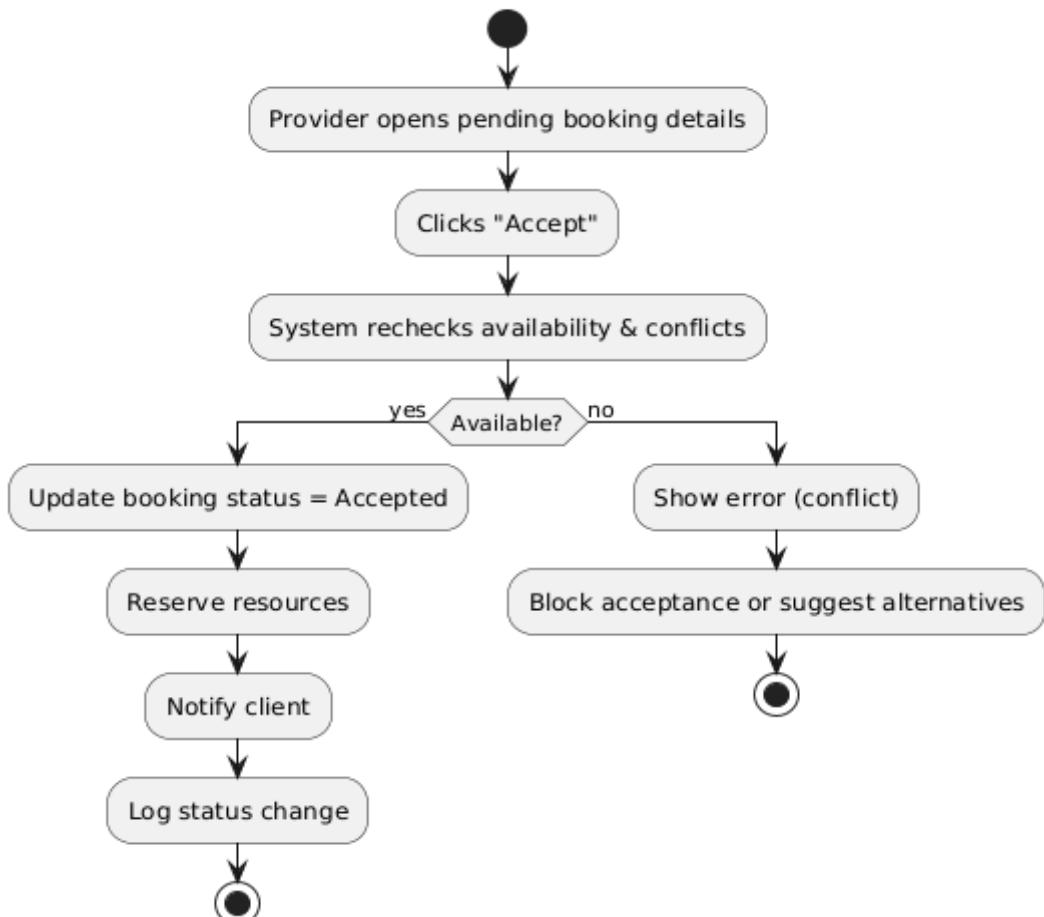


Figure 4.44 accept booking activity diagram

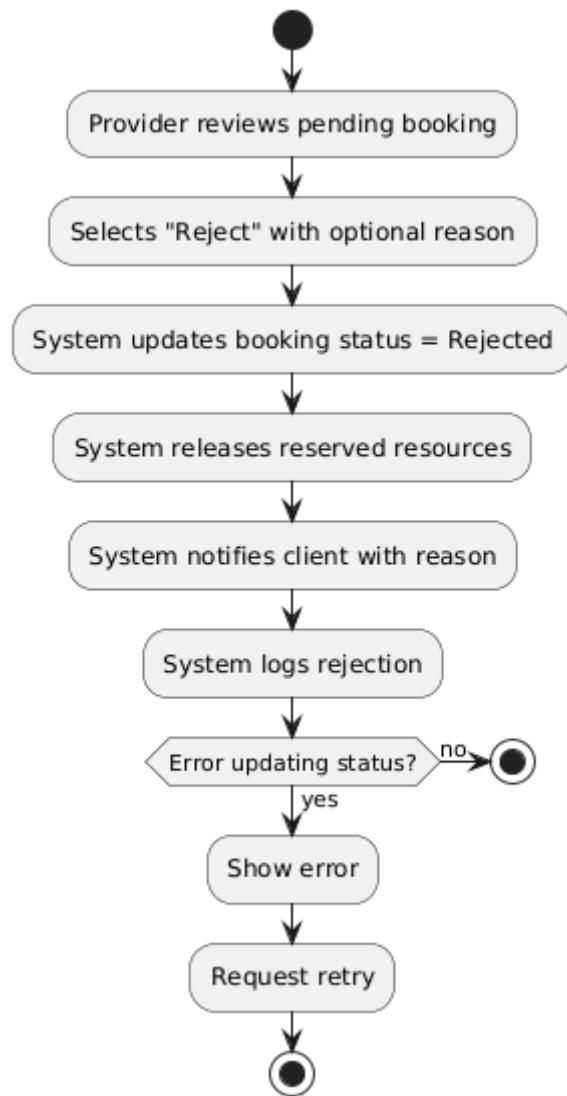


Figure 4.45 reject booking activity diagram

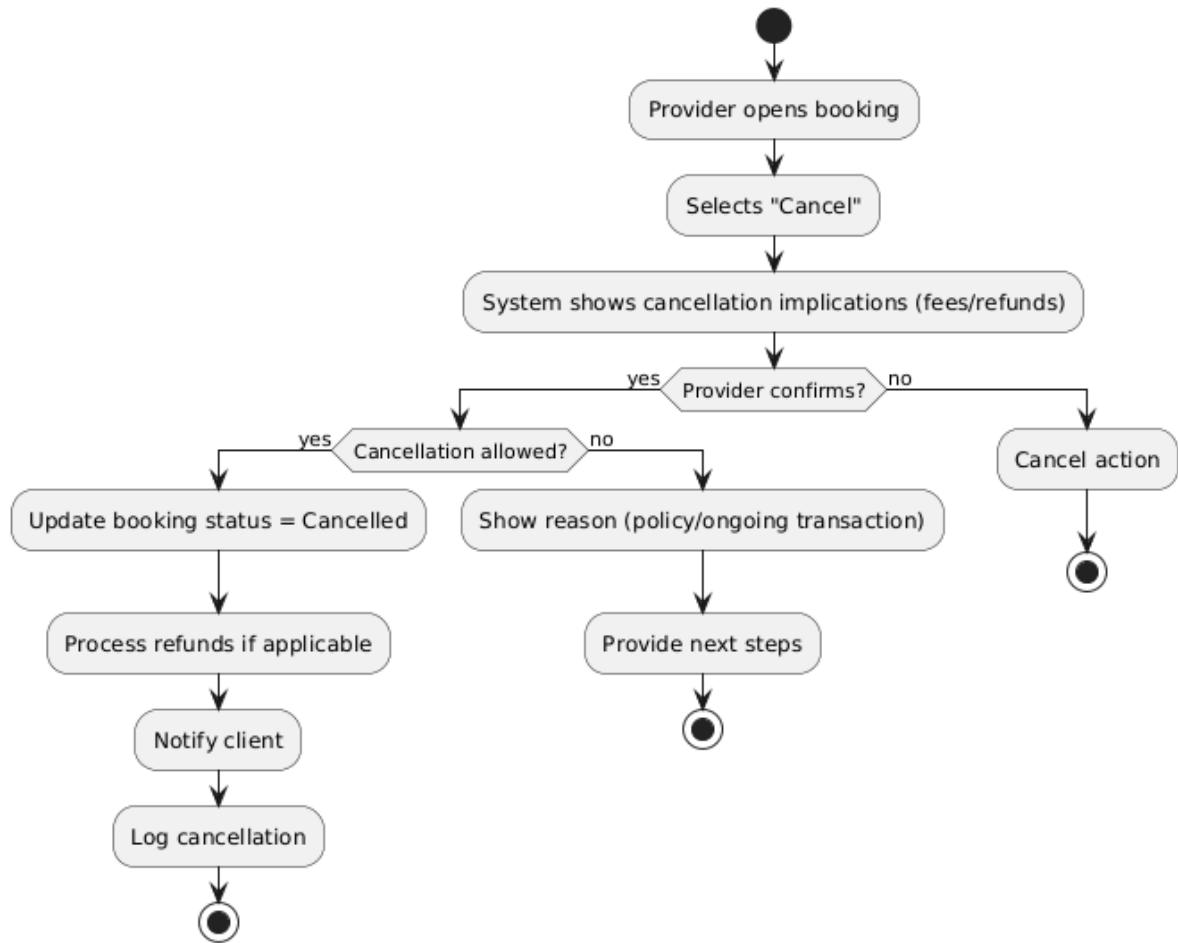


Figure 4.46 cancel booking activity diagram

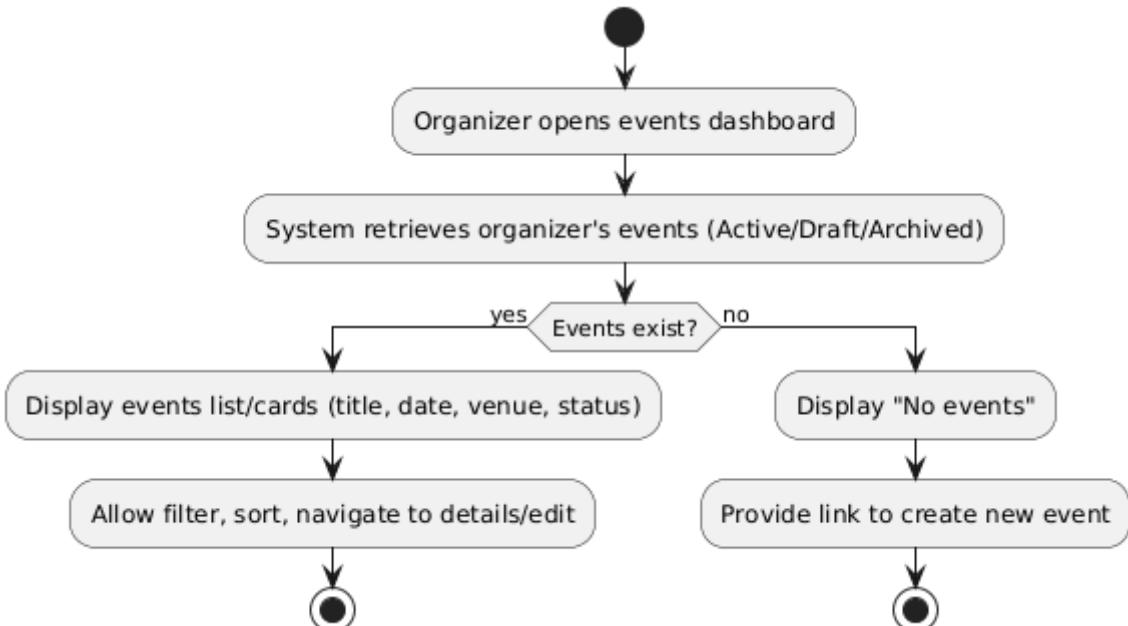


Figure 4.47 browse events (organizer) activity diagram

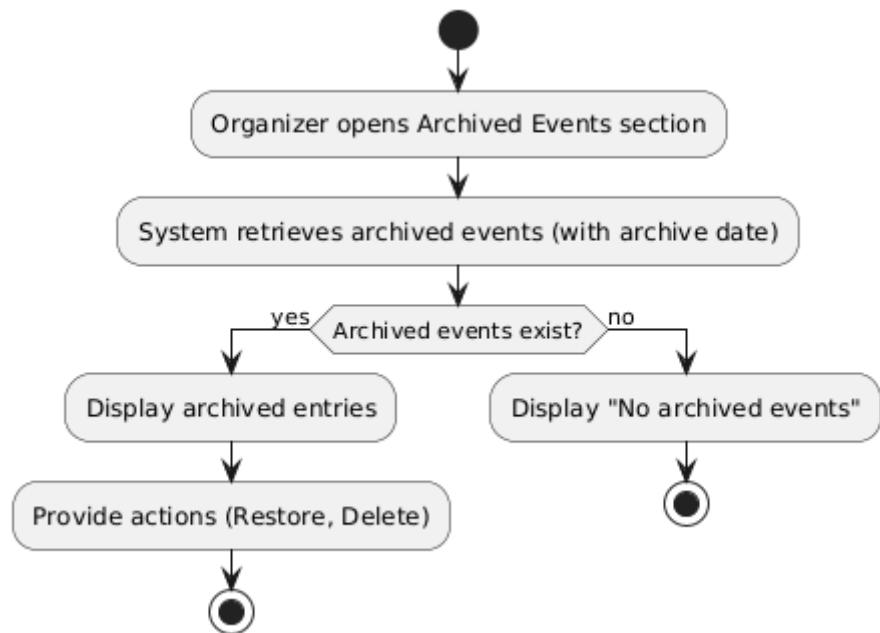


Figure 4.48 browse archived events activity diagram

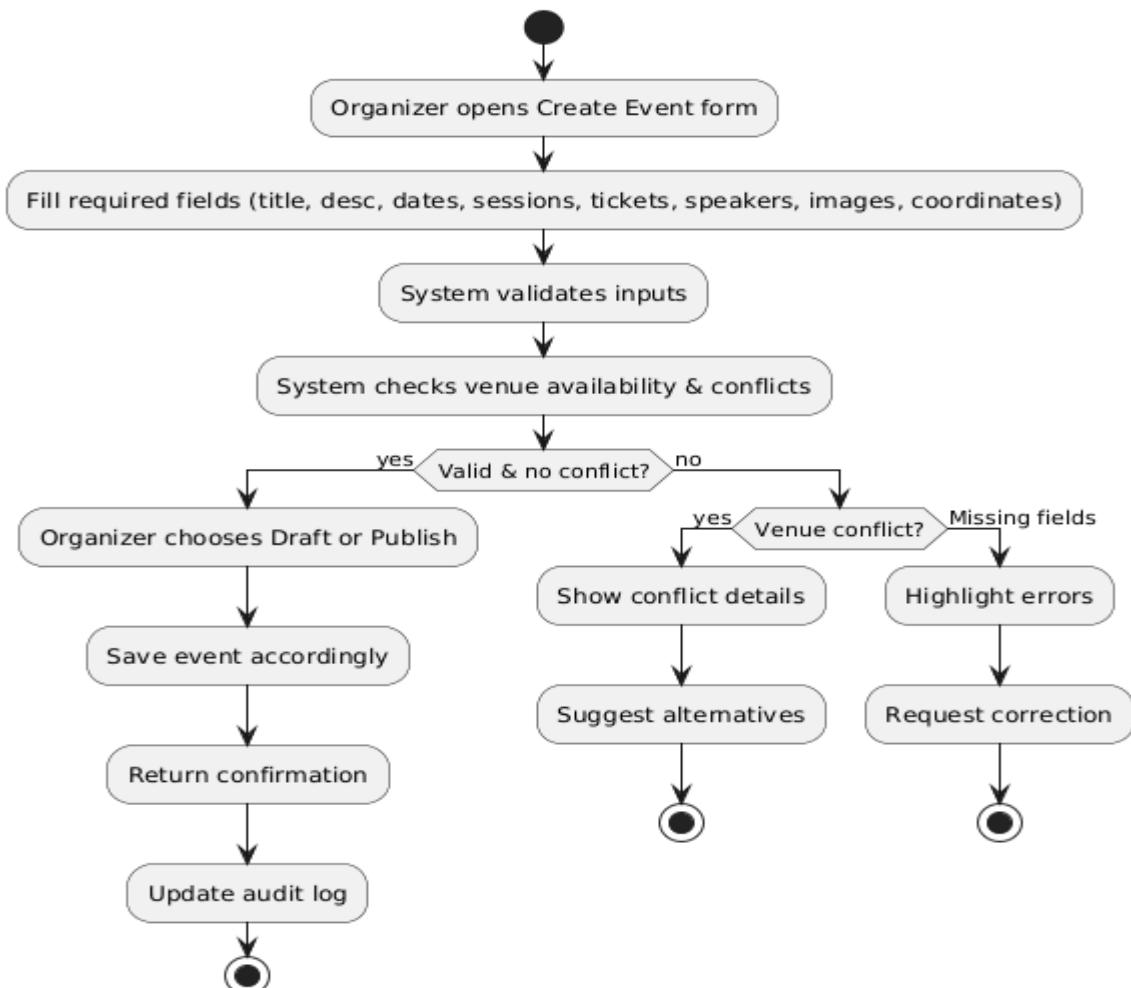


Figure 4.49 create event activity diagram

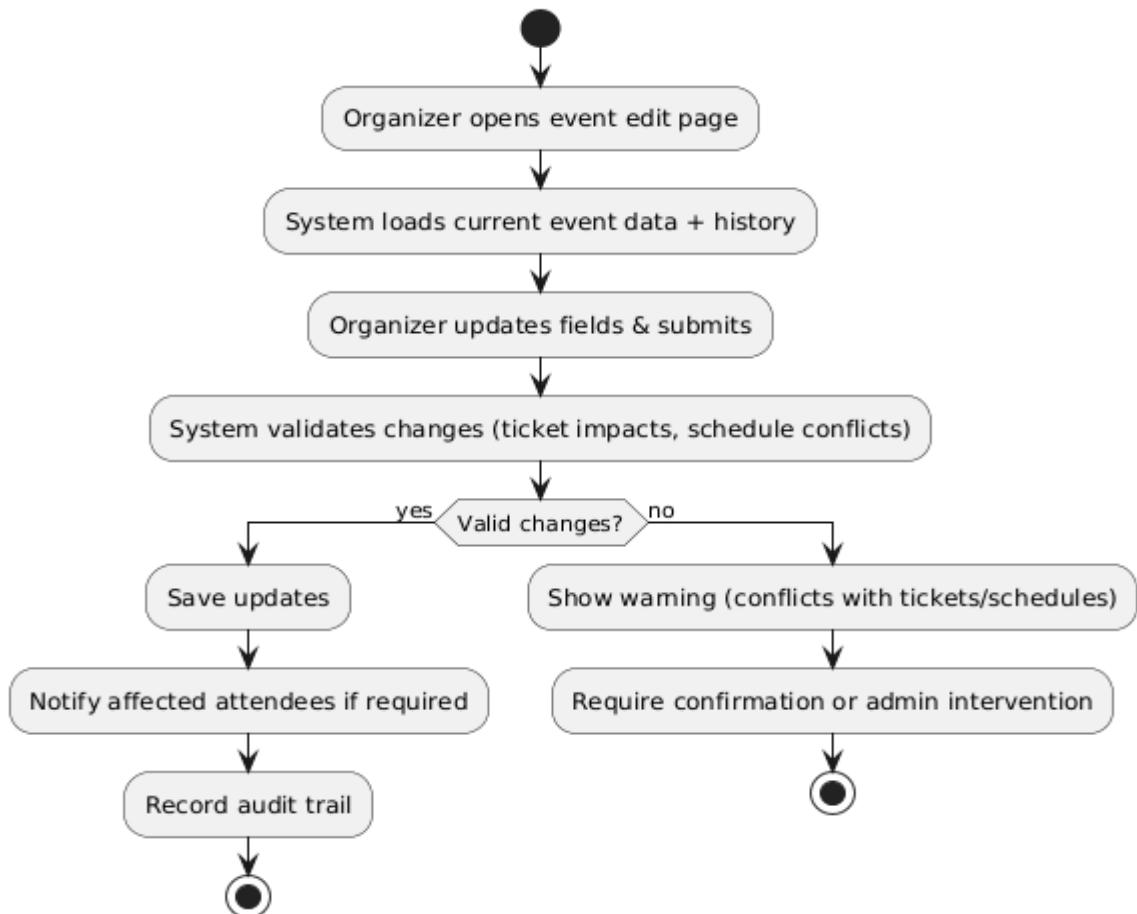


Figure 4.50 edit event activity diagram

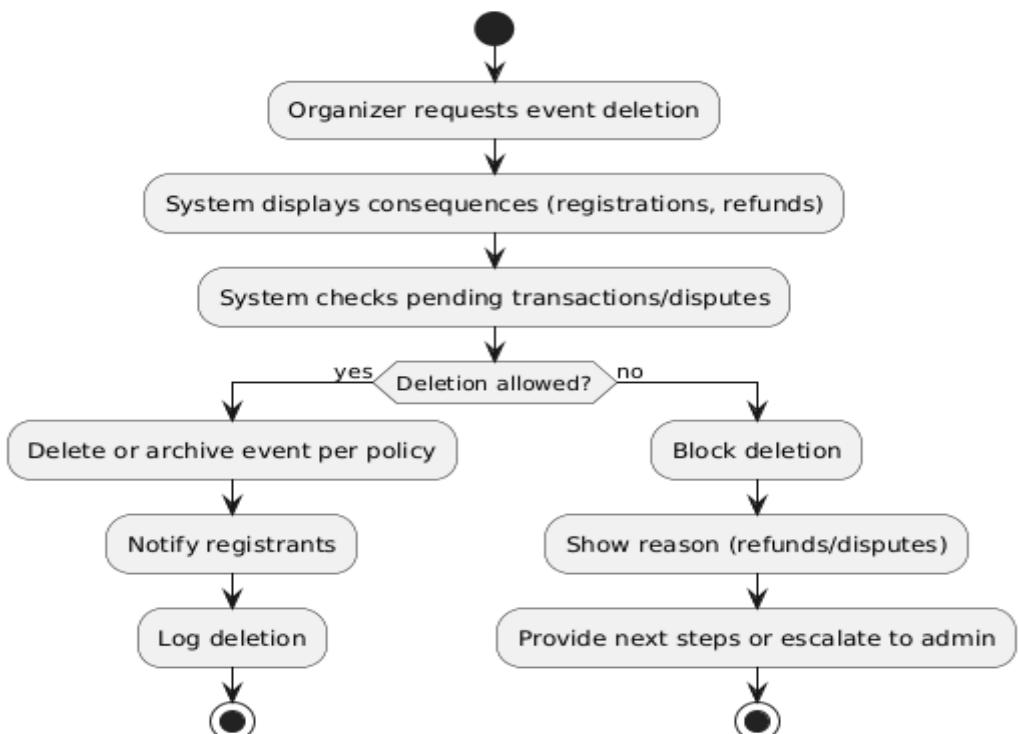


Figure 4.51 delete event activity diagram

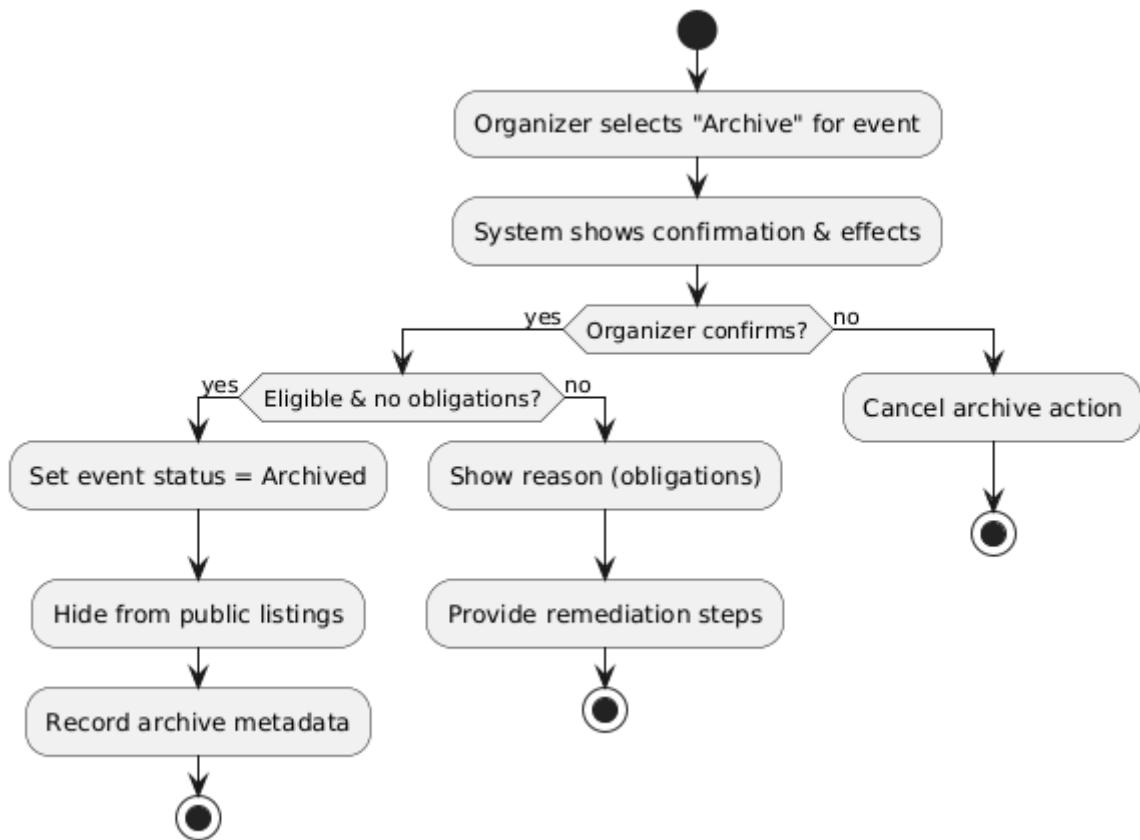


Figure 4.52 archive event activity diagram

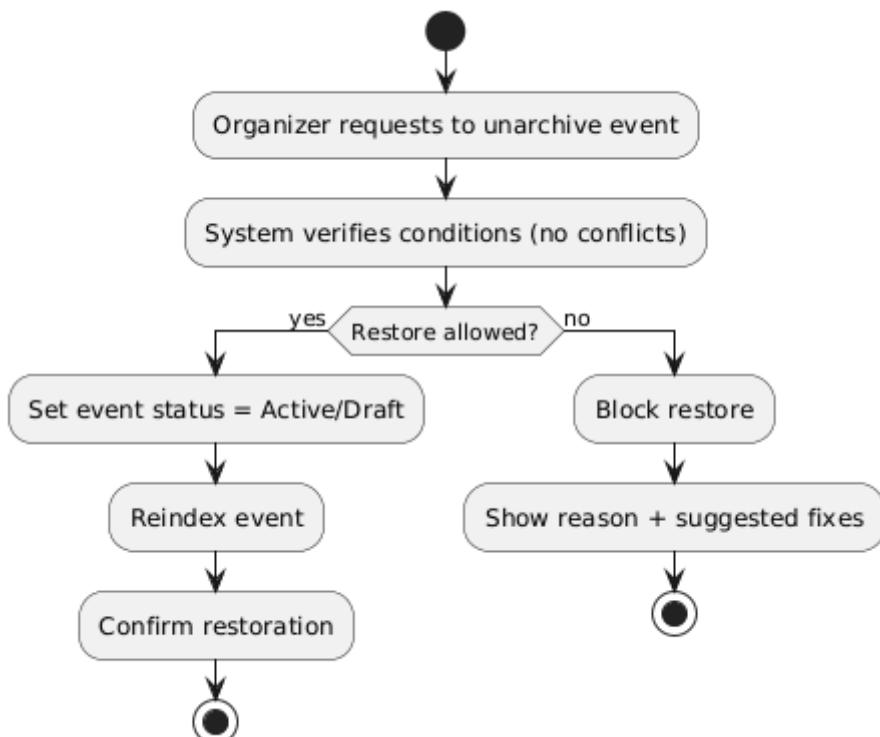


Figure 4.53 unarchive event activity diagram

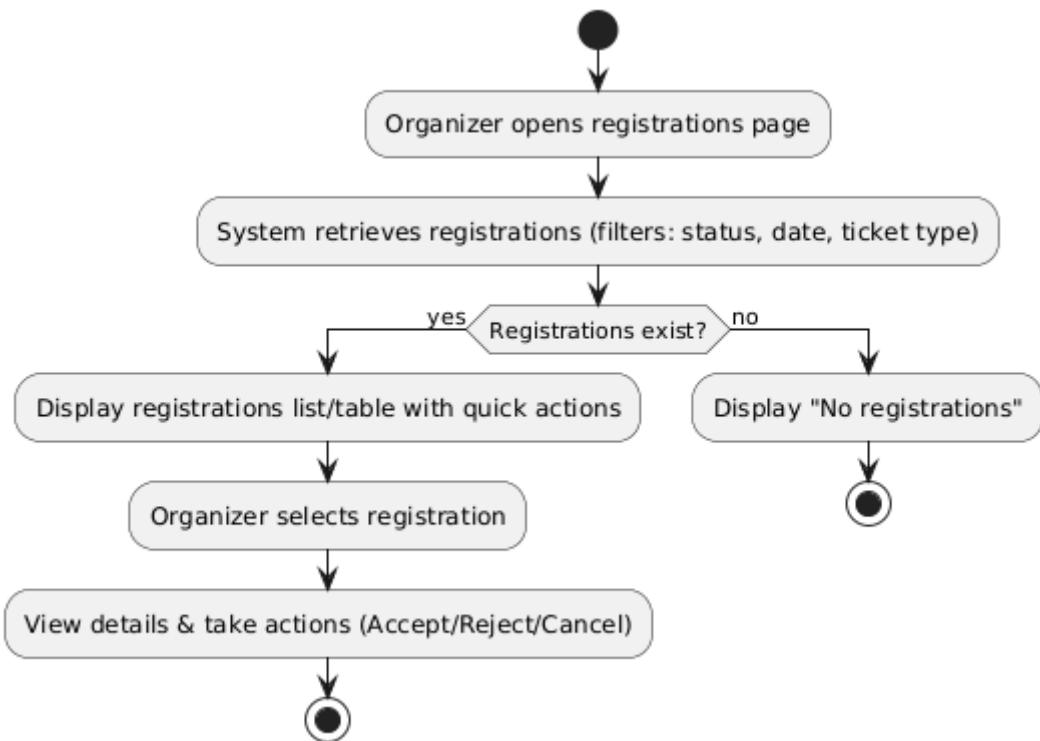


Figure 4.54 browse registrations activity diagram

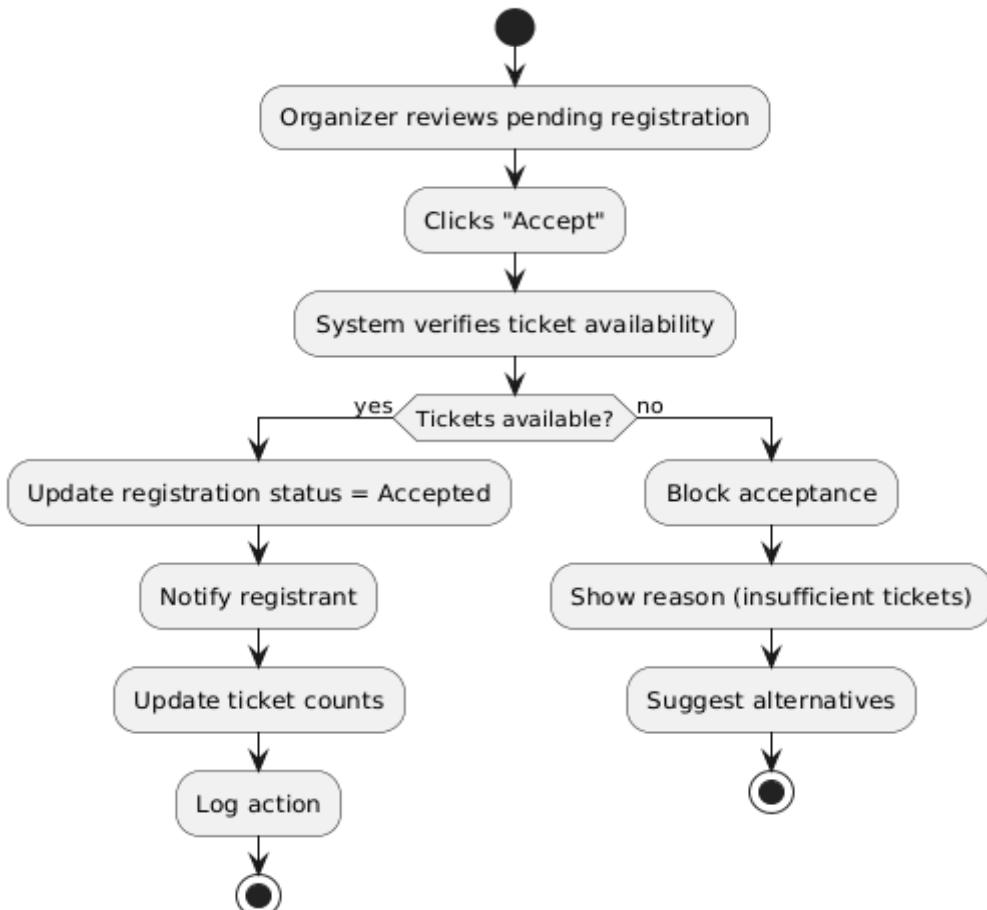


Figure 4.55 accept registration activity diagram

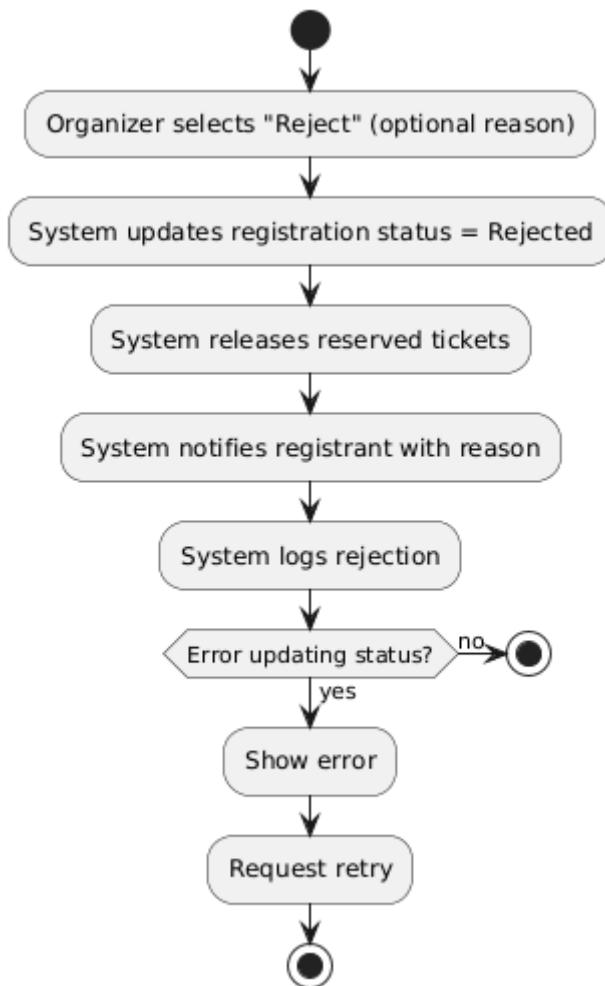


Figure 4.56 reject registration activity diagram

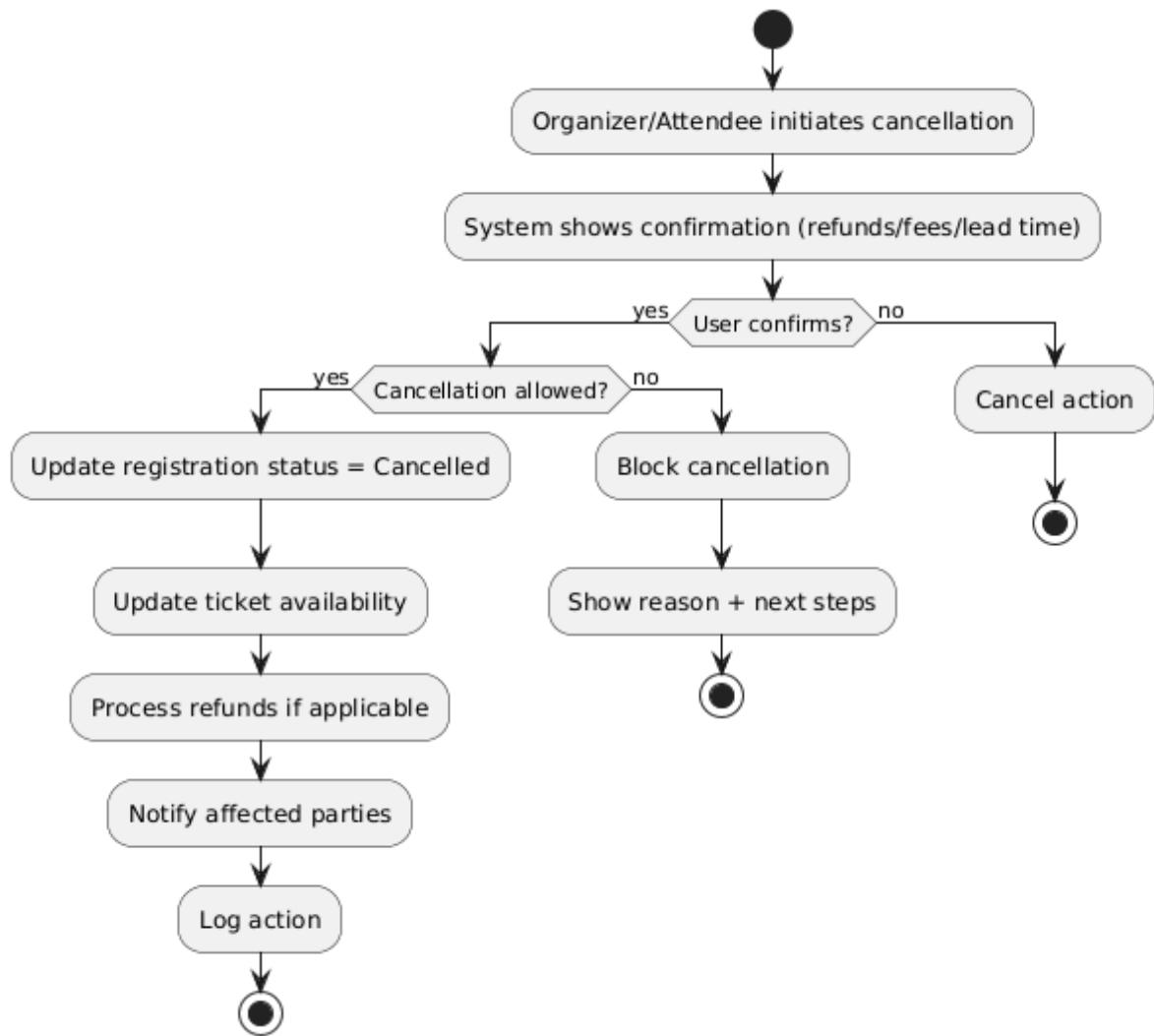


Figure 4.57 cancel registration activity diagram

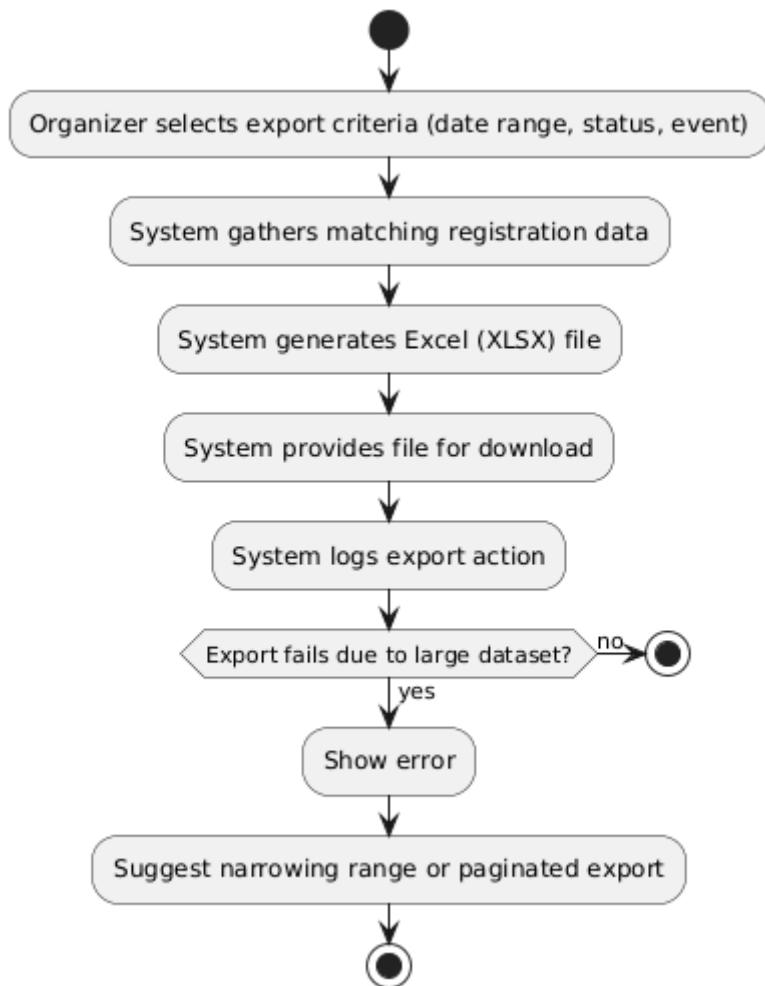


Figure 4.58 explore registration as excel file activity diagram

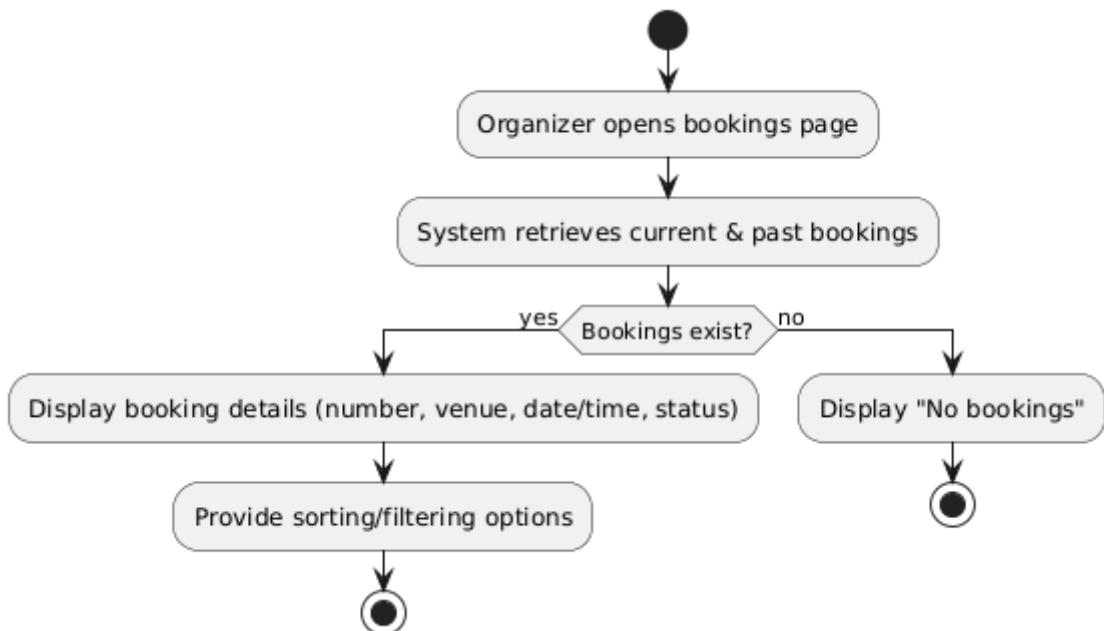


Figure 4.59 view booking (organizer) activity diagram

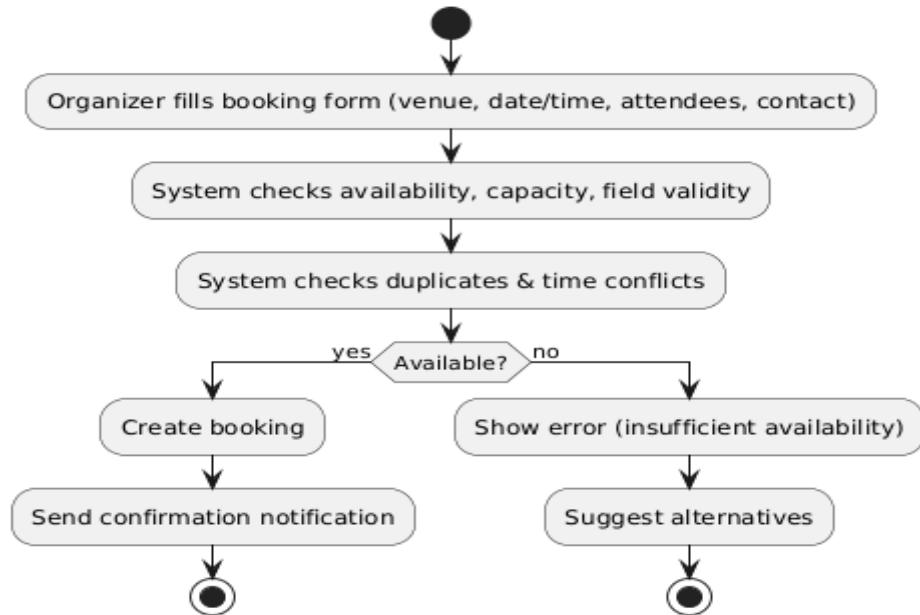


Figure 4.60 add booking (organizer) activity diagram

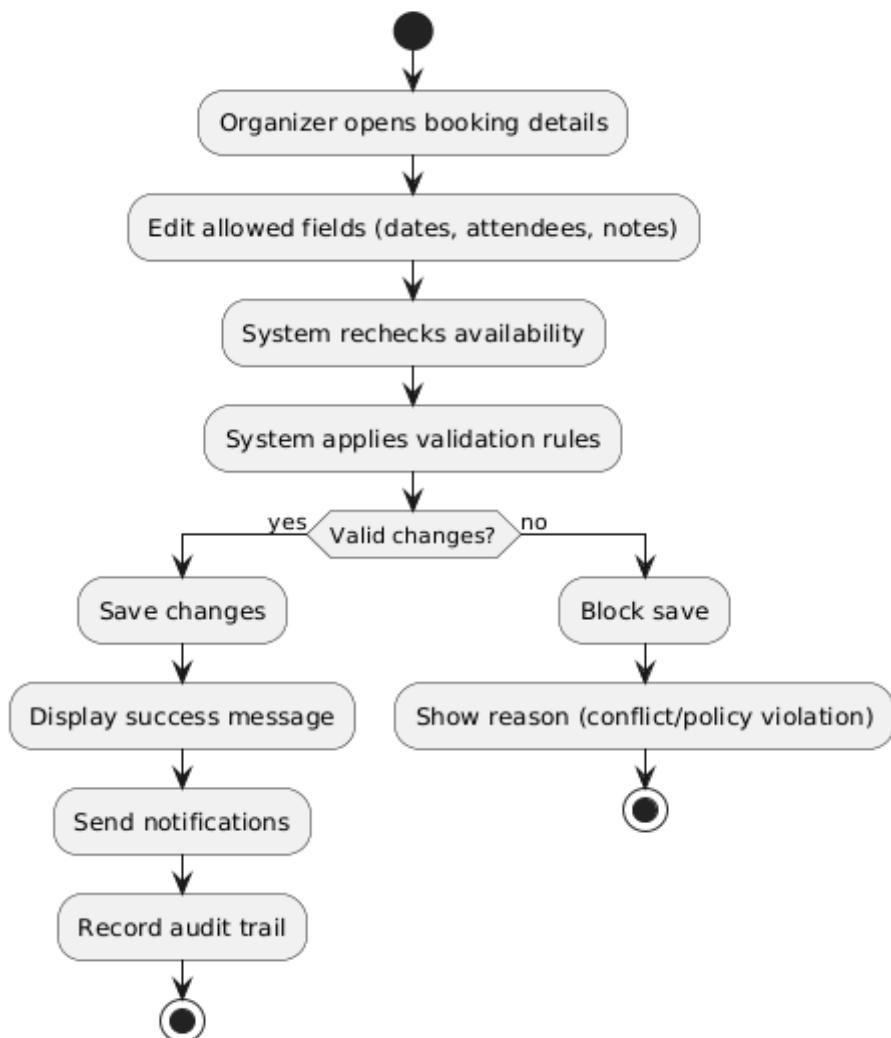


Figure 4.61 edit booking (organizer) activity diagram

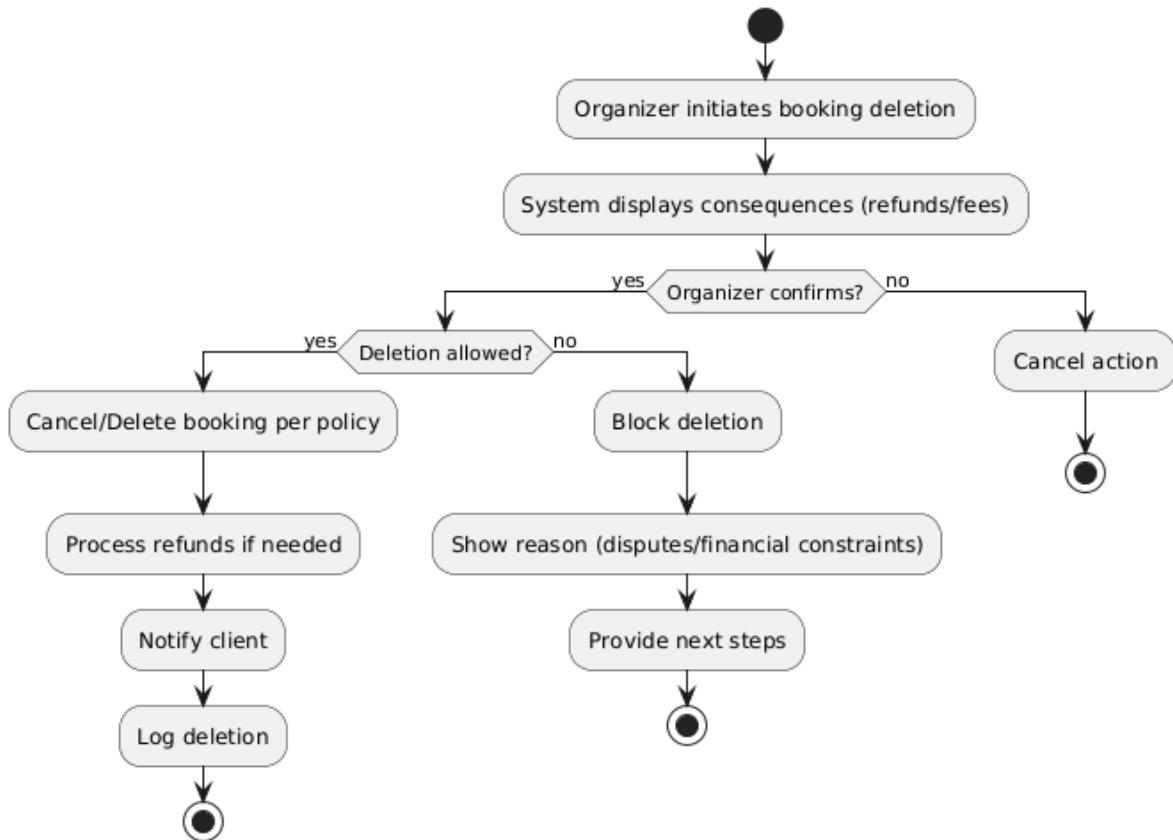


Figure 4.62 delete booking (organizer) activity diagram

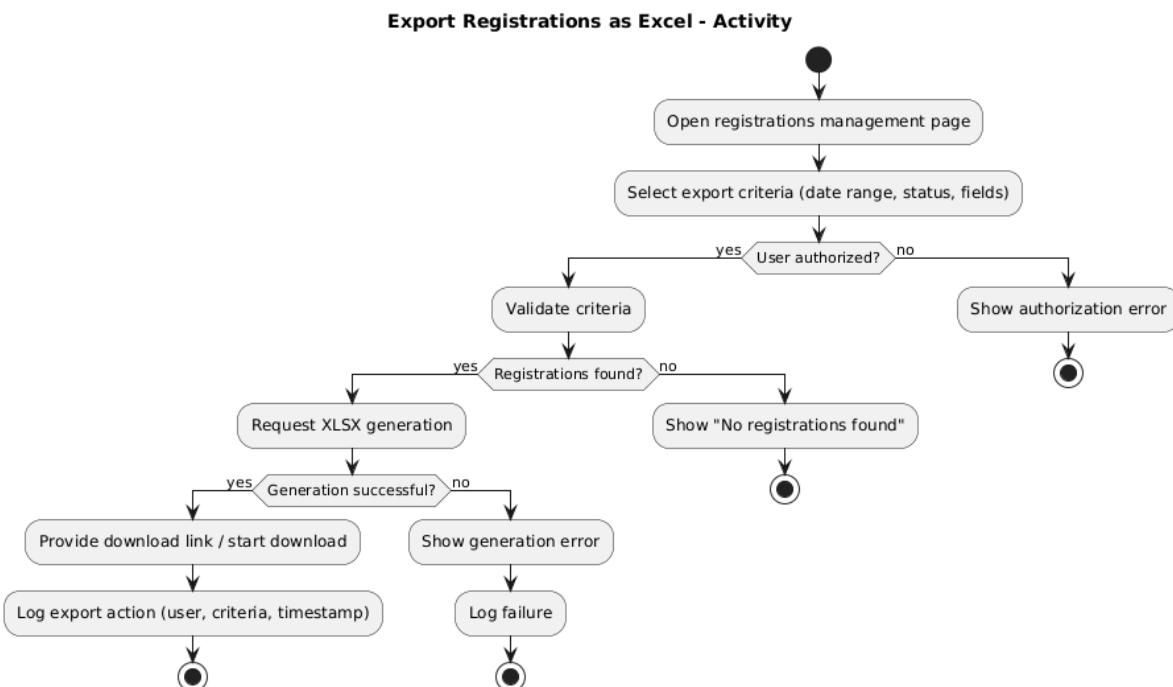


Figure 4.63 export registration as excel file activity diagram

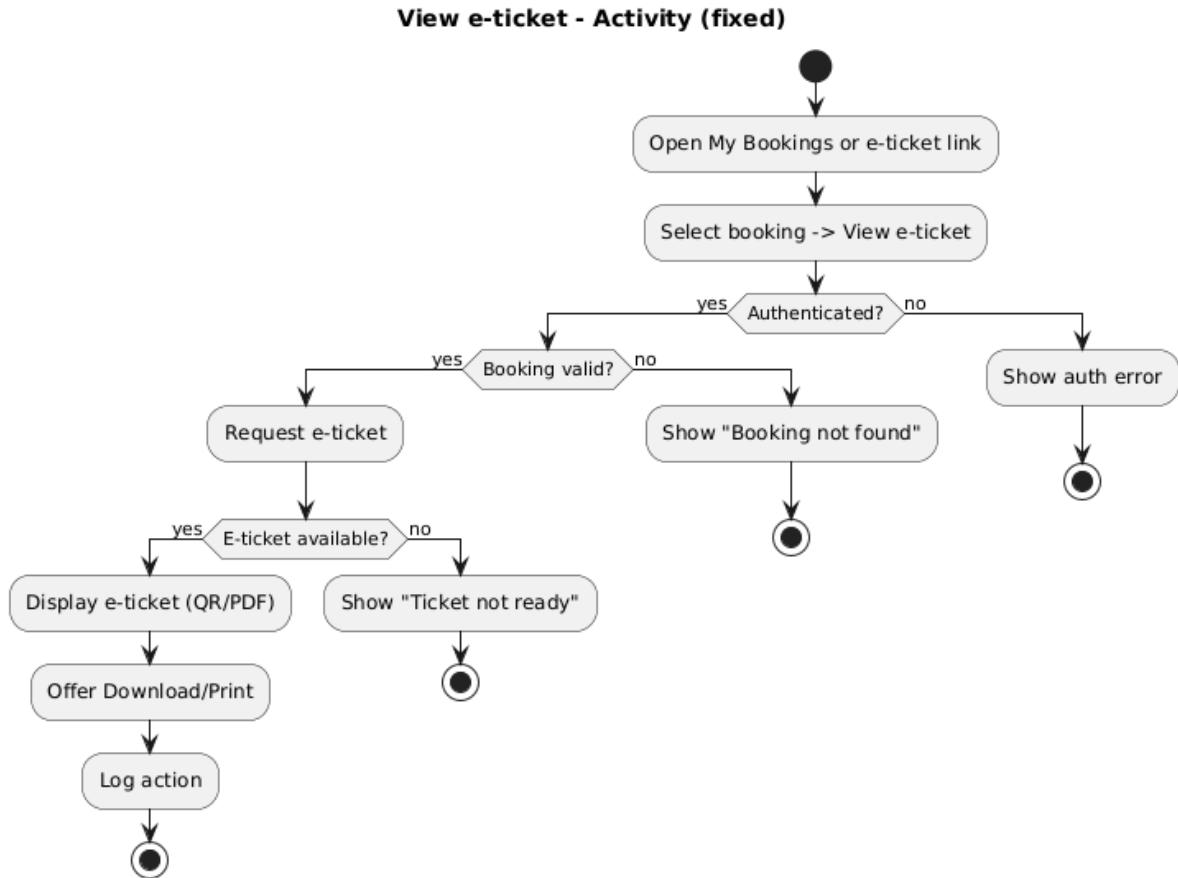


Figure 4.64 view e-ticket activity diagram

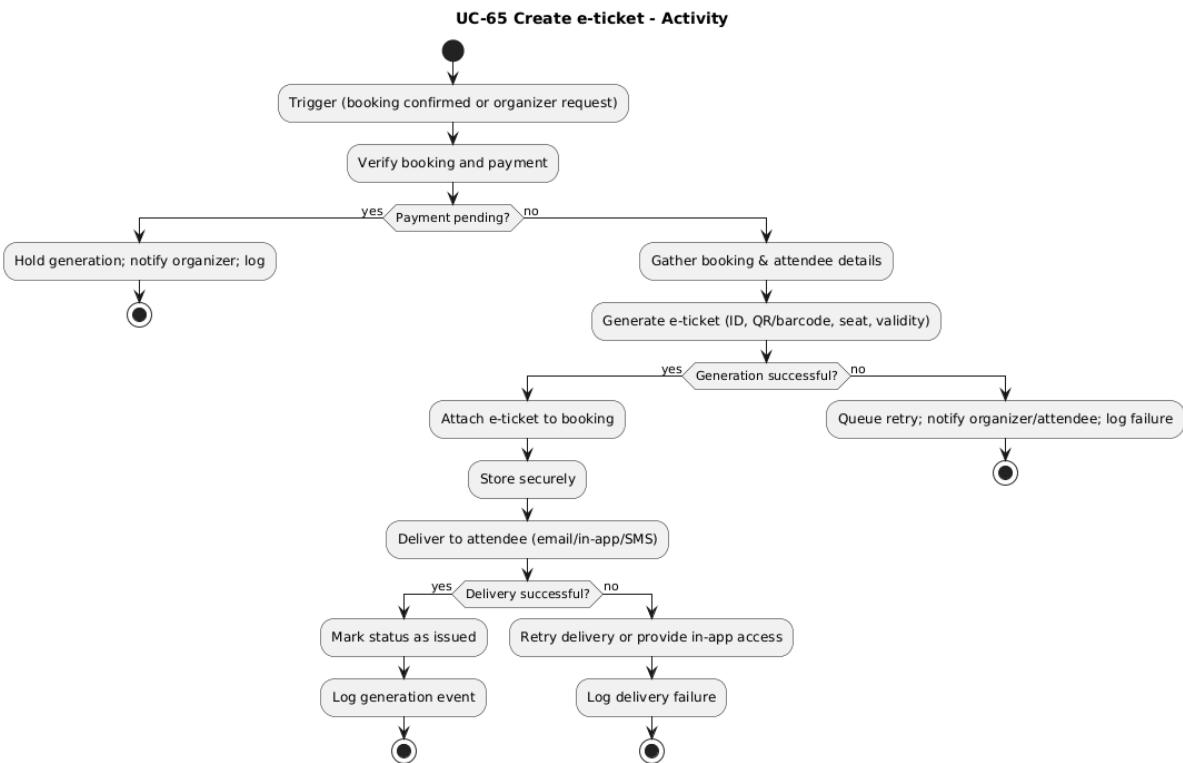


Figure 4.65 create e-ticket activity diagram

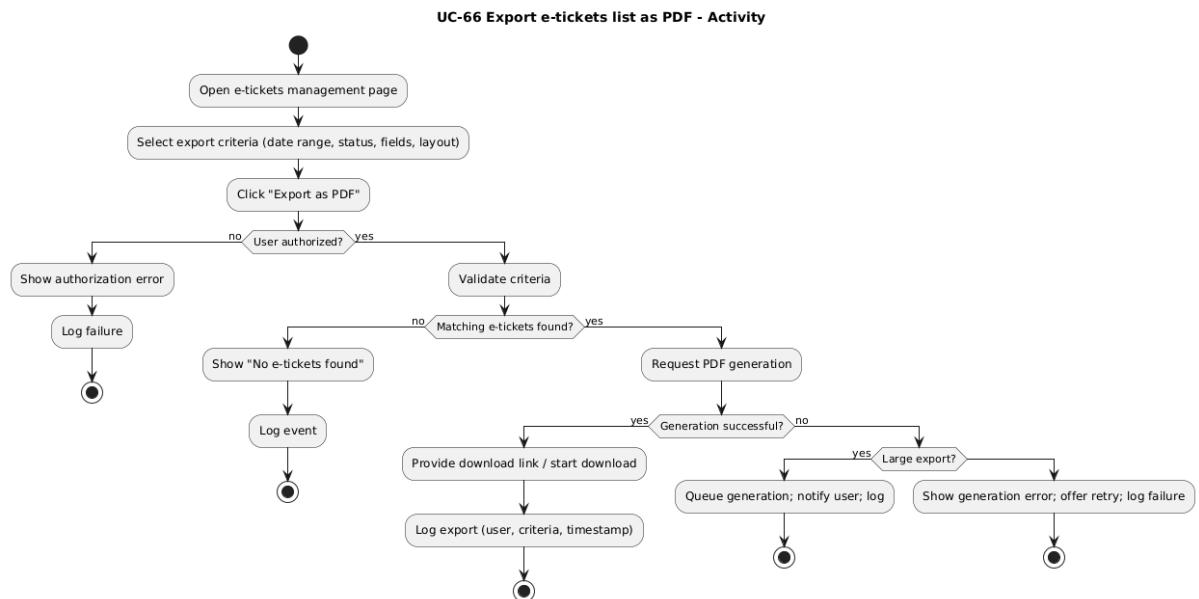


Figure 4.66 export e-tickets list as pdf activity diagram

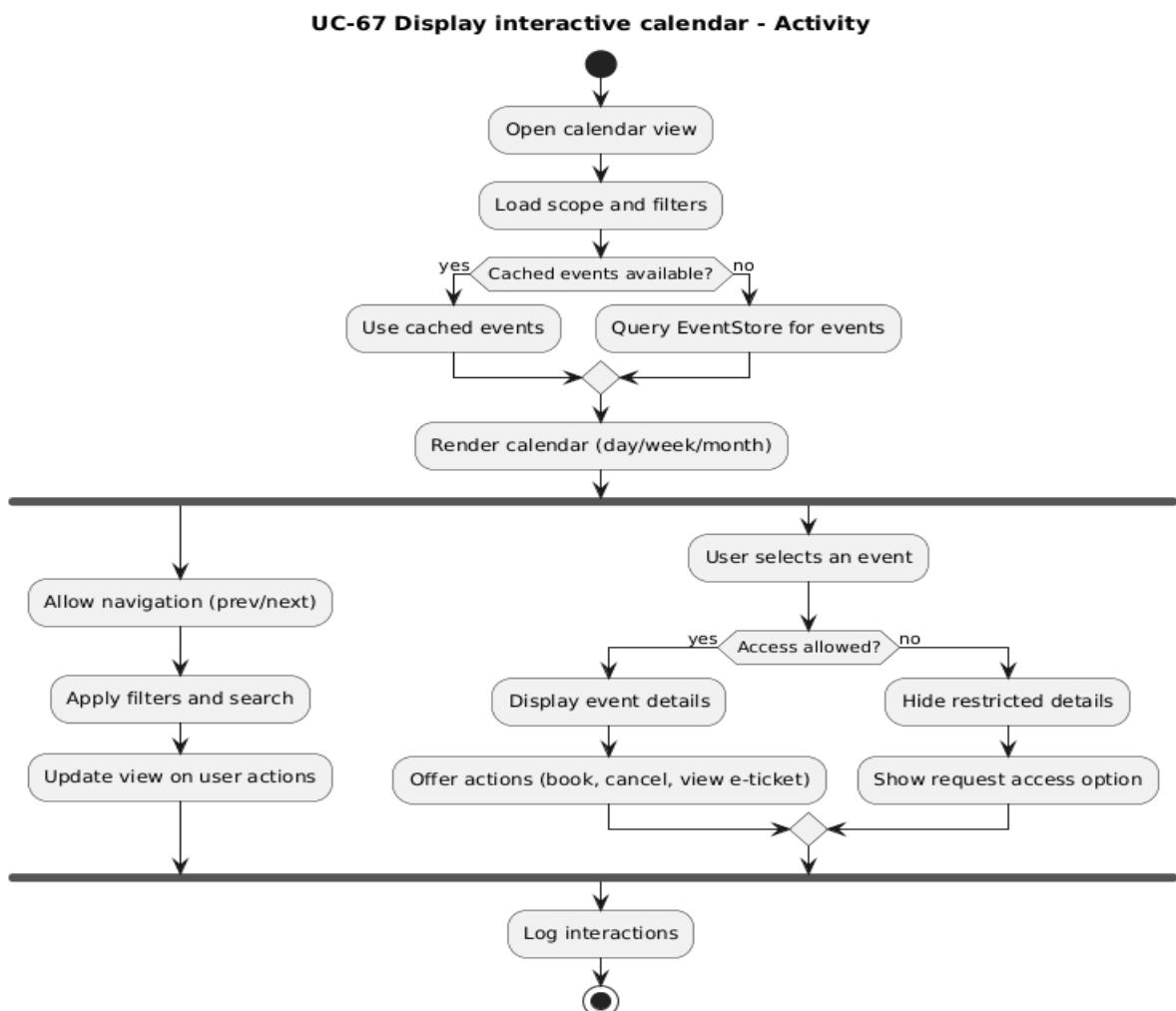


Figure 4.67 display interactive calendar activity diagram

- sequence Diagram

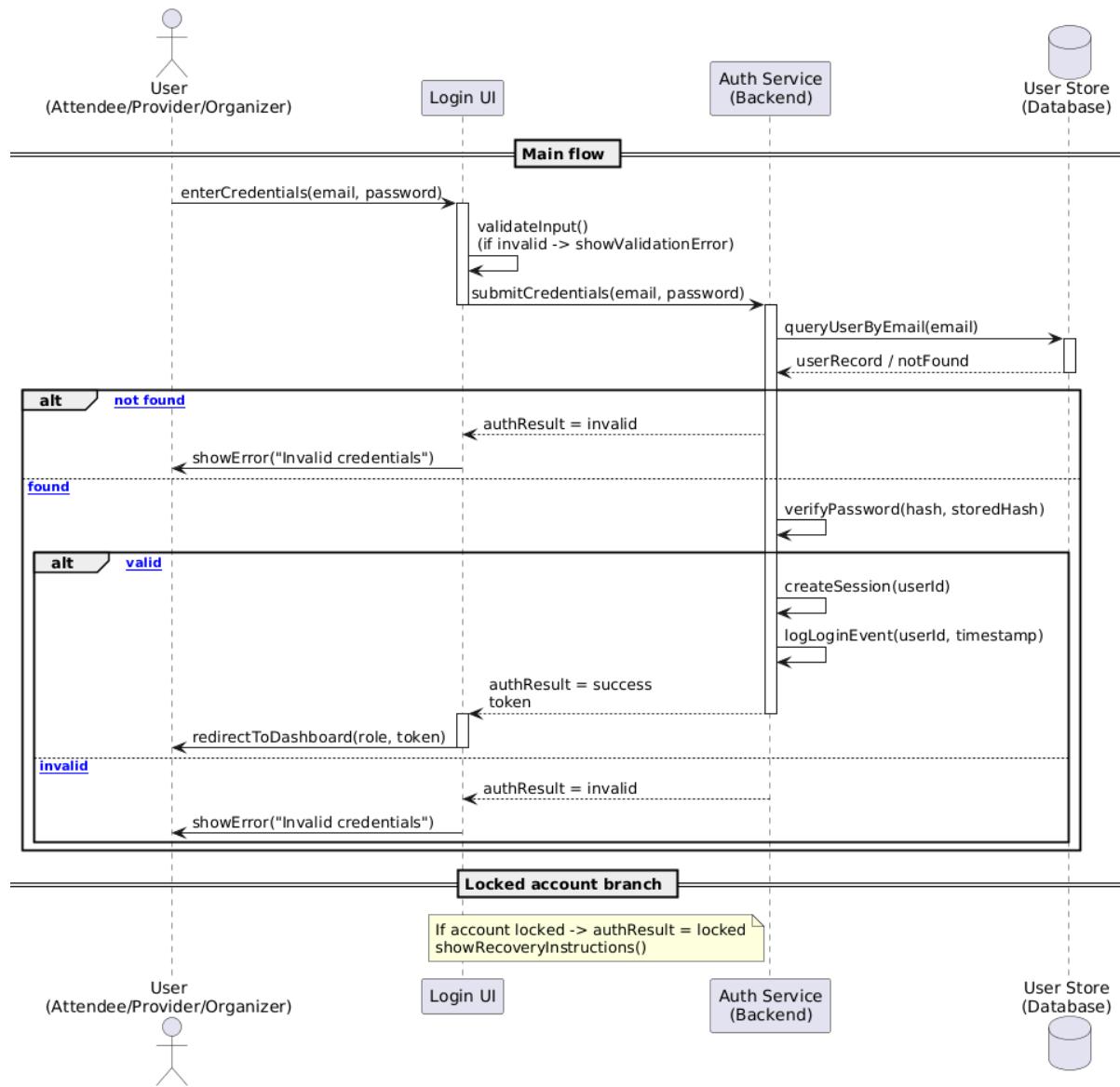


Figure 4.68 login sequence diagram

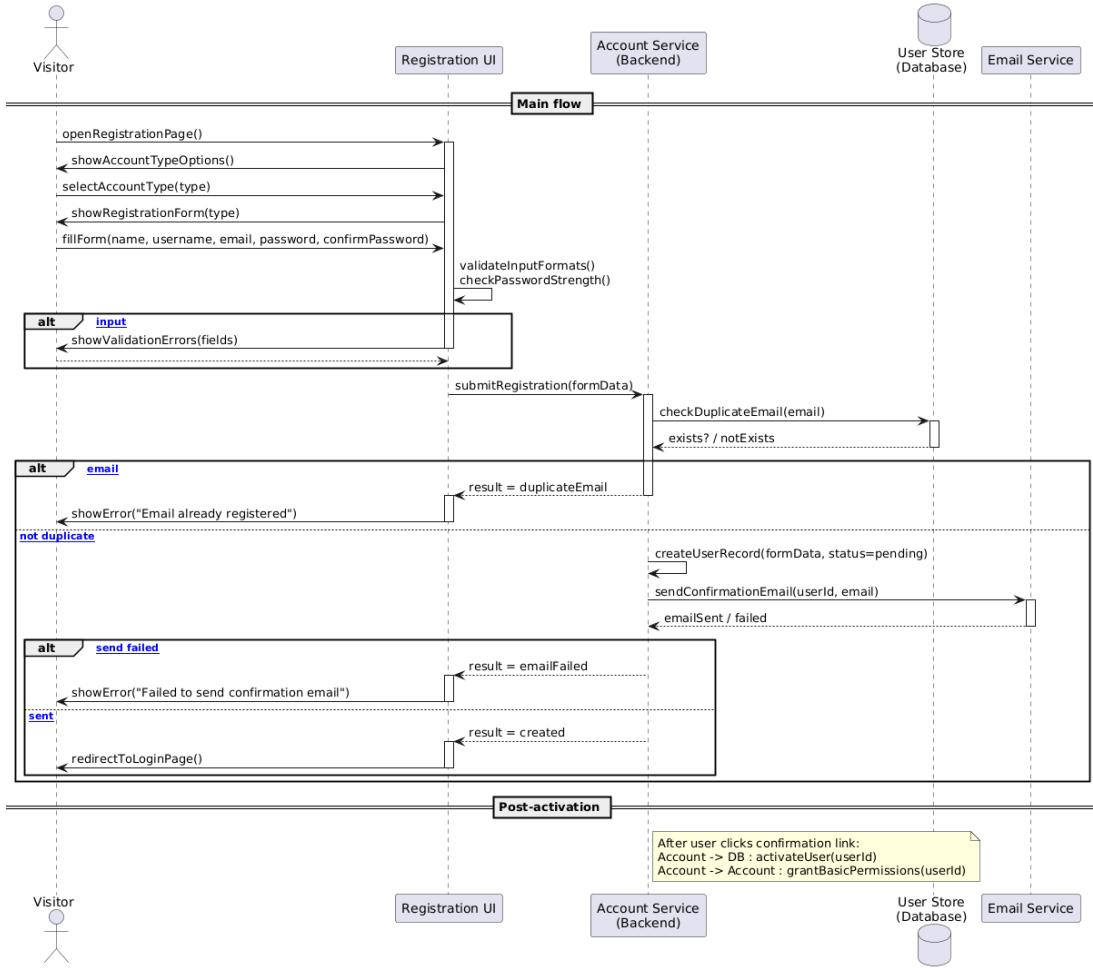


Figure 4.69 register (User Registration) sequence diagram

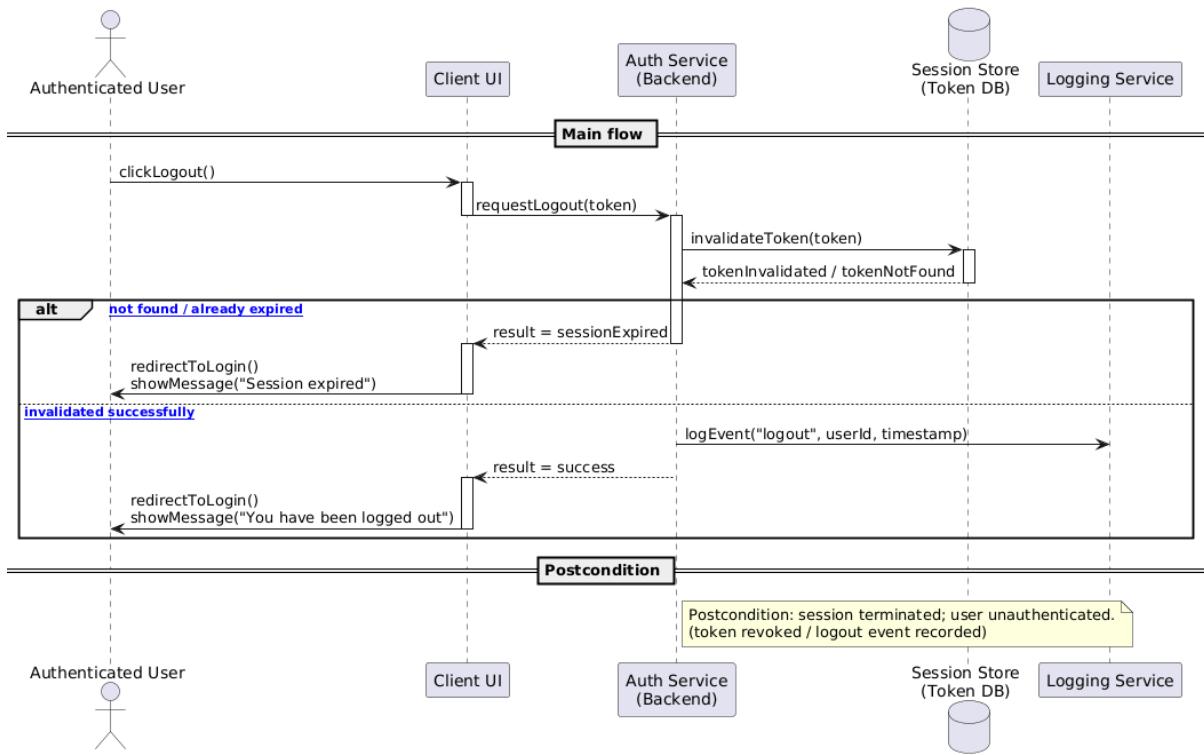


Figure 4.70 Register Logout sequence diagram

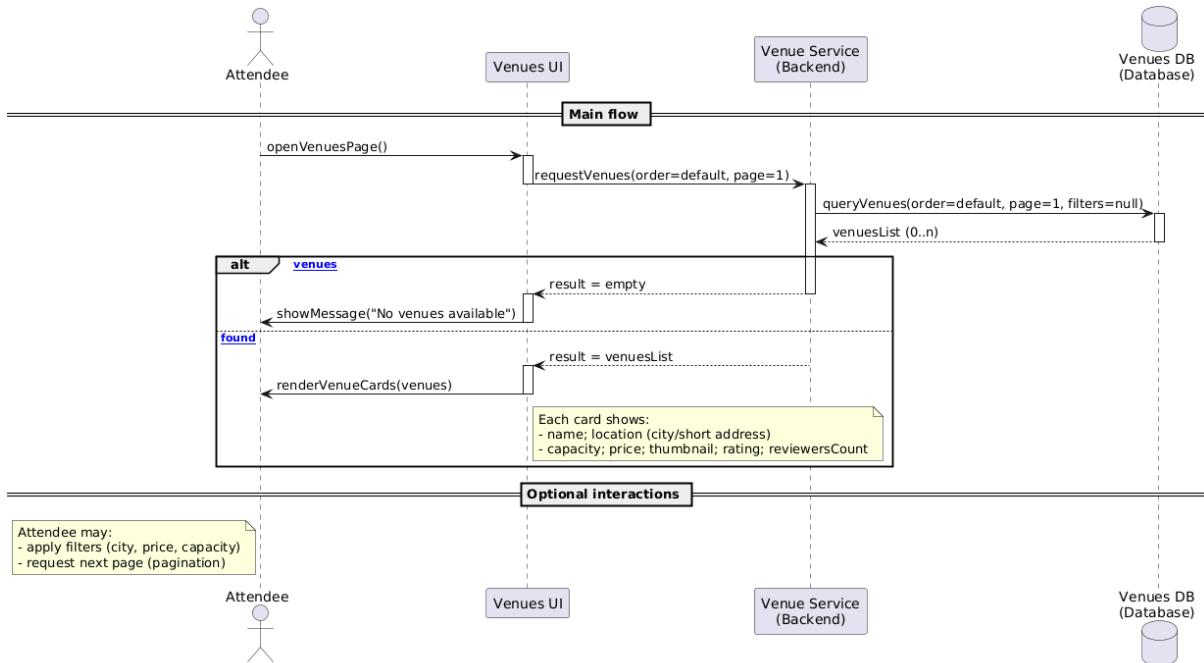


Figure 4.71 view venues sequence diagram

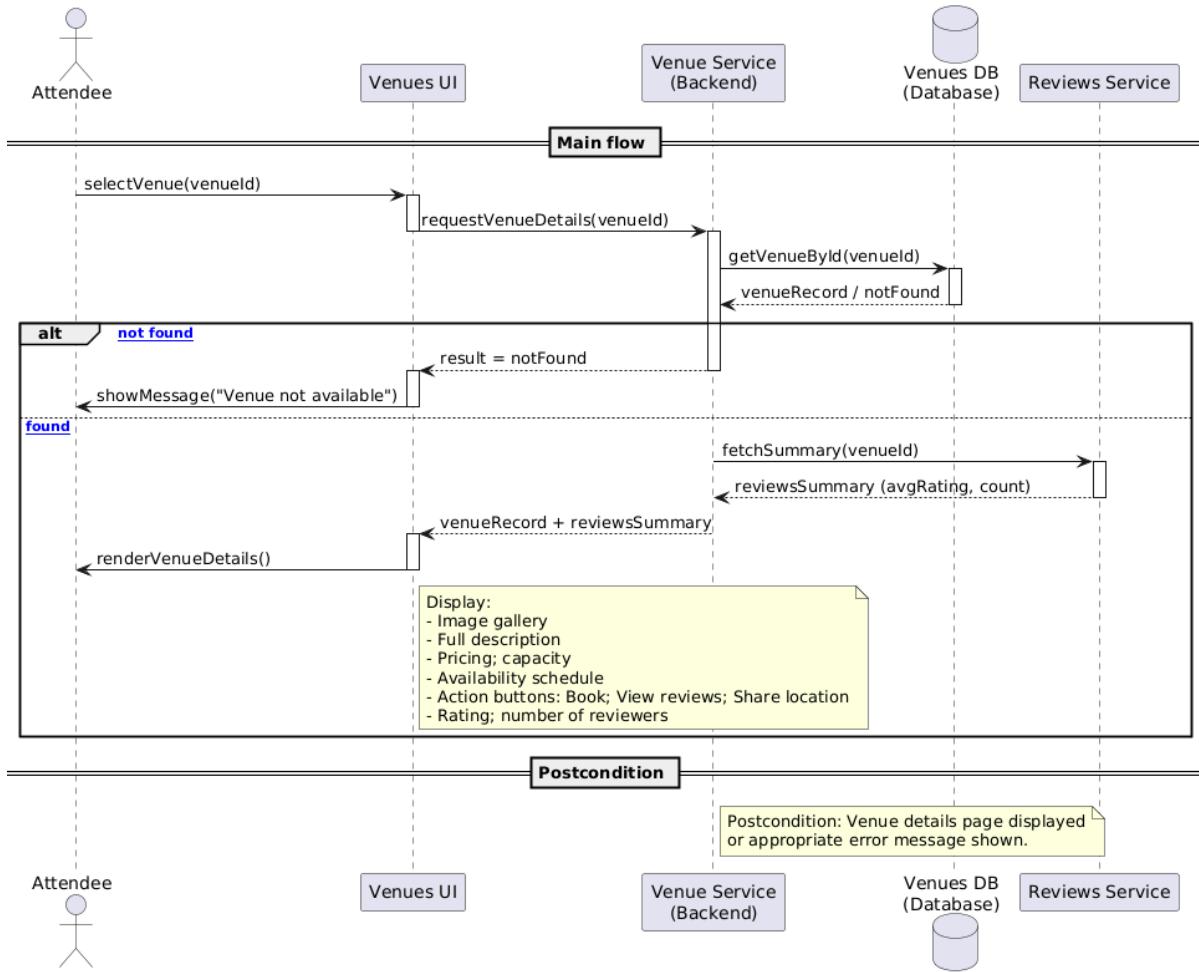


Figure 4.72 view venues details sequence diagram

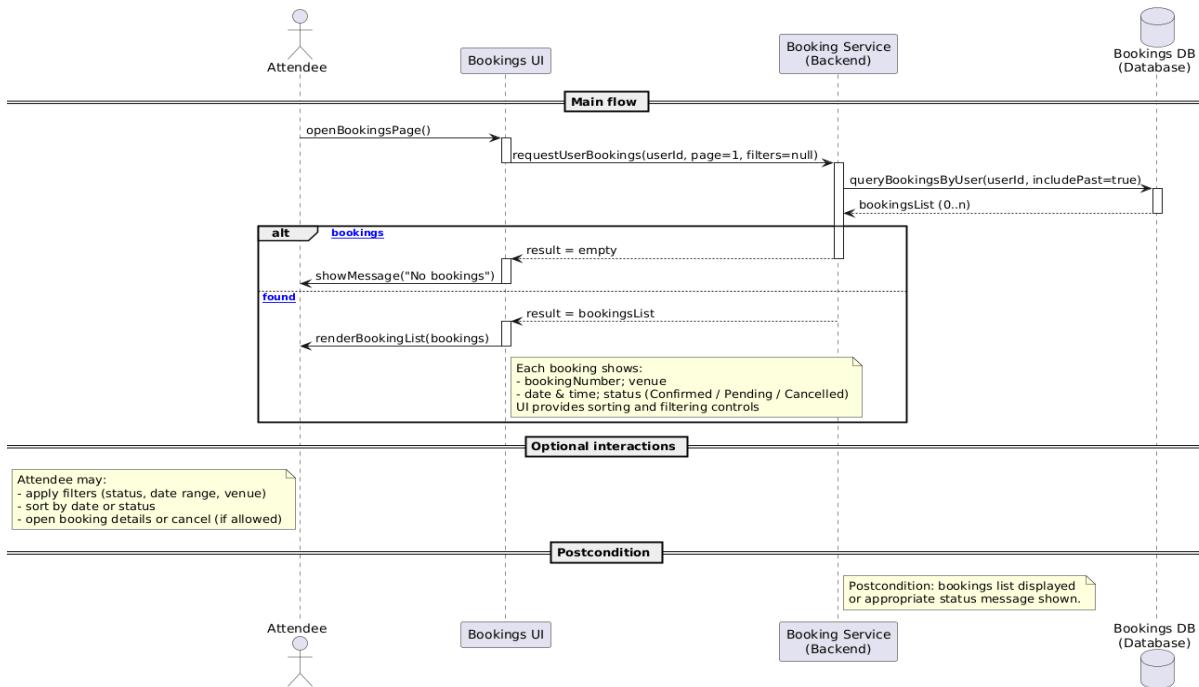


Figure 4.73 view bookings sequence diagram

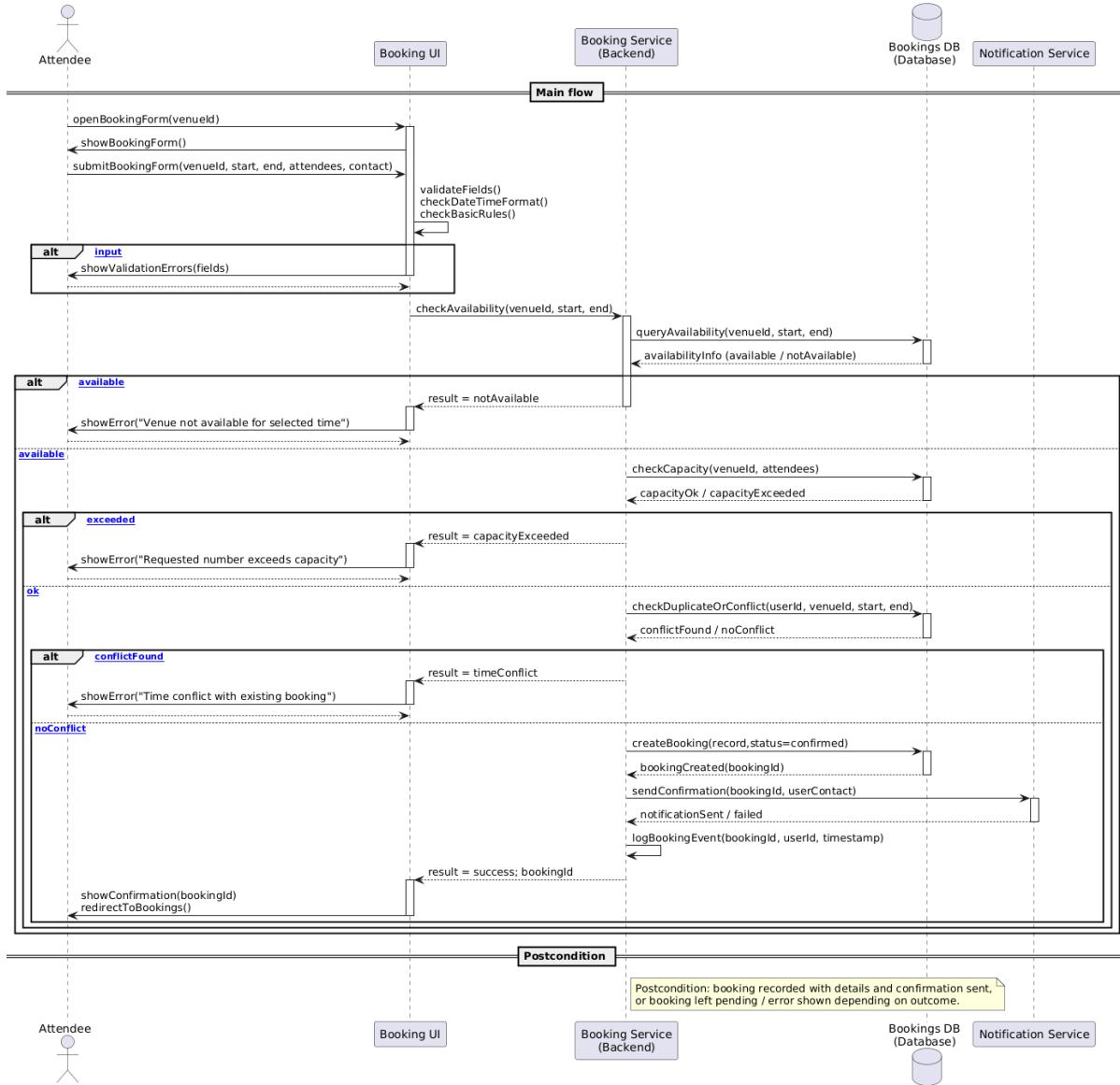


Figure 4.74 add booking sequence diagram

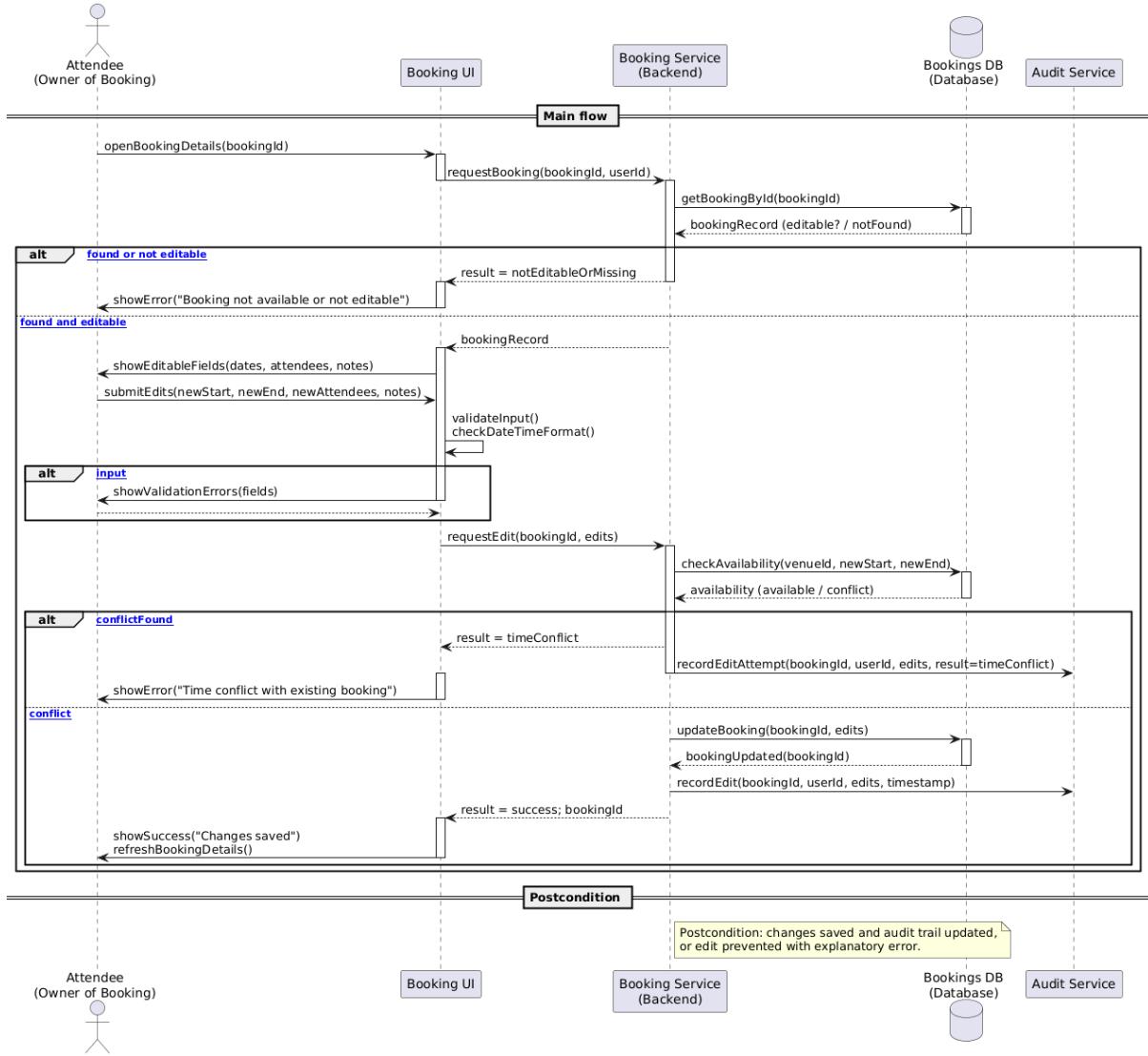


Figure 4.75 edit booking sequence diagram

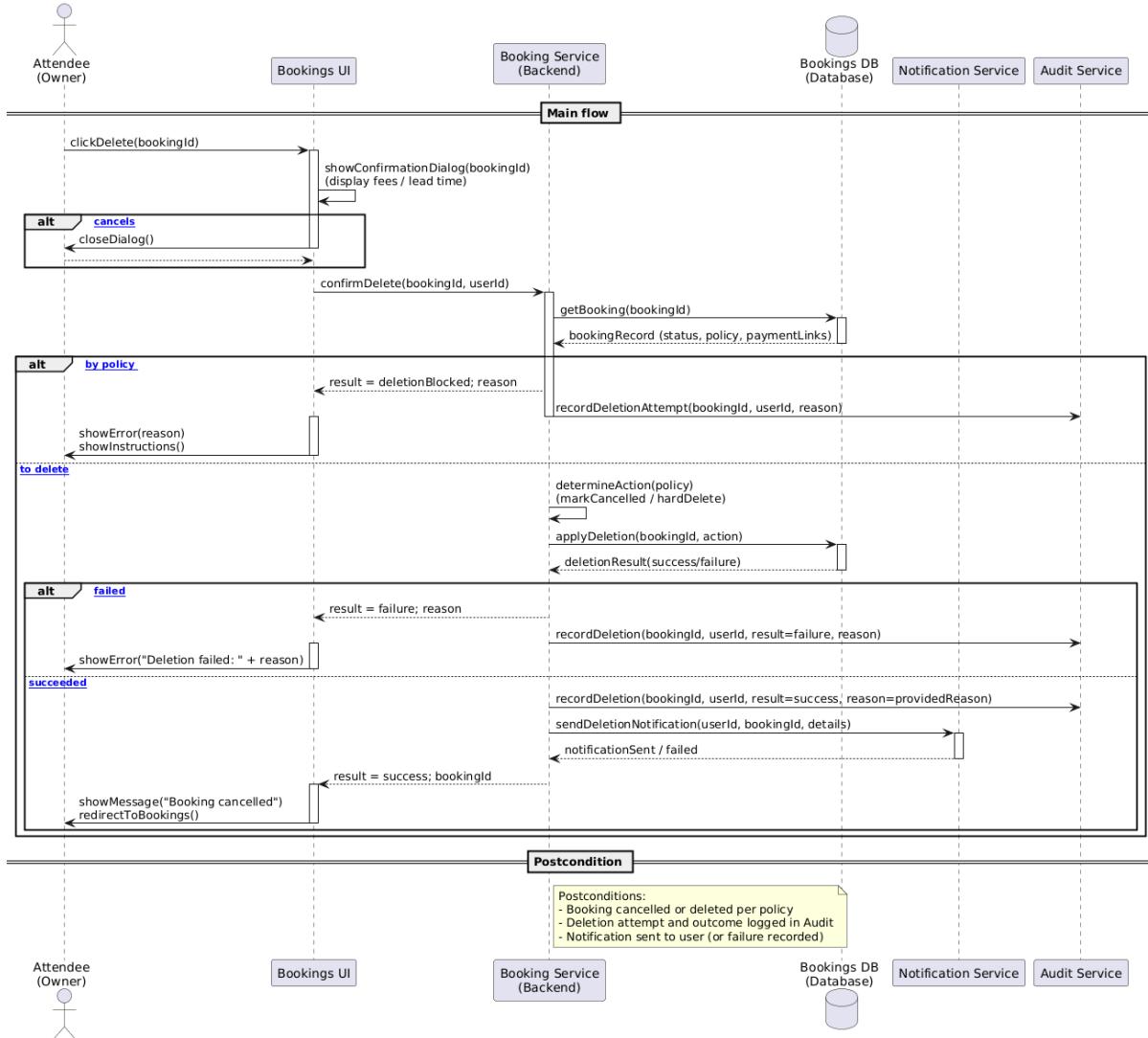


Figure 4.76 delete booking sequence diagram

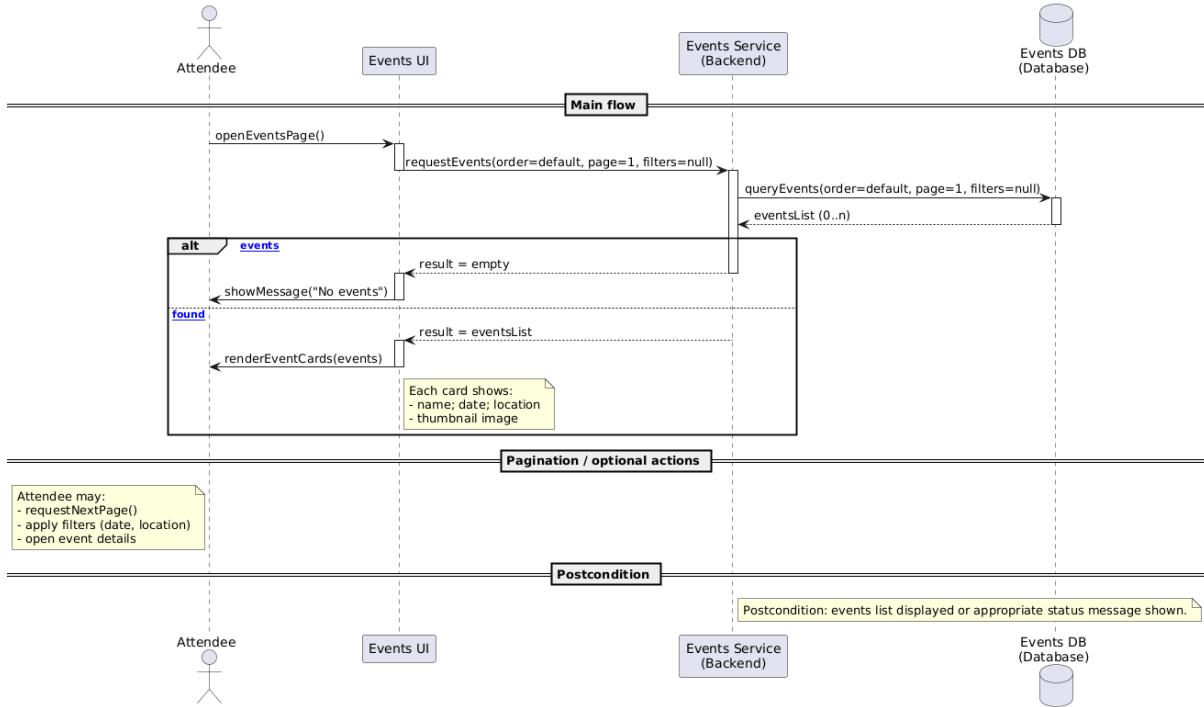


Figure 4.77 view events sequence diagram

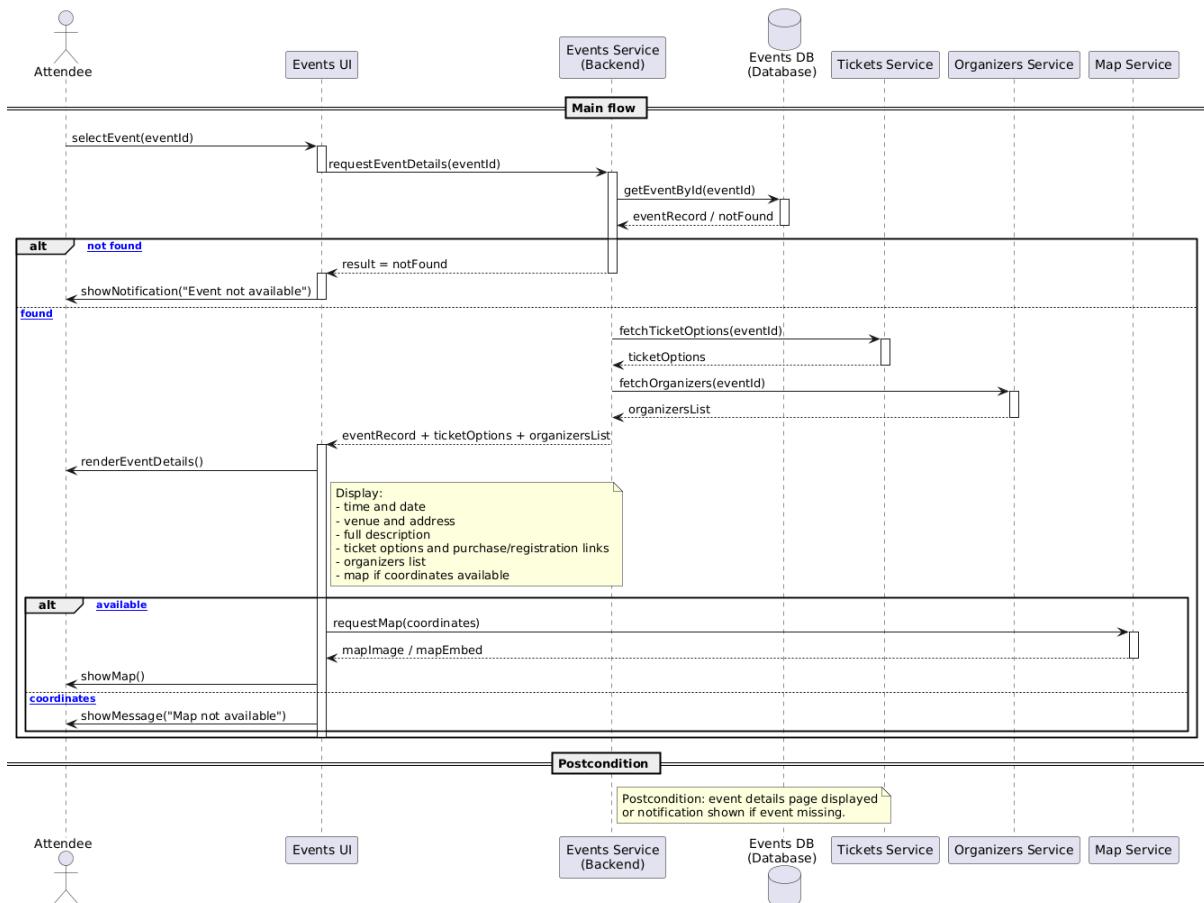


Figure 4.78 view events details sequence diagram

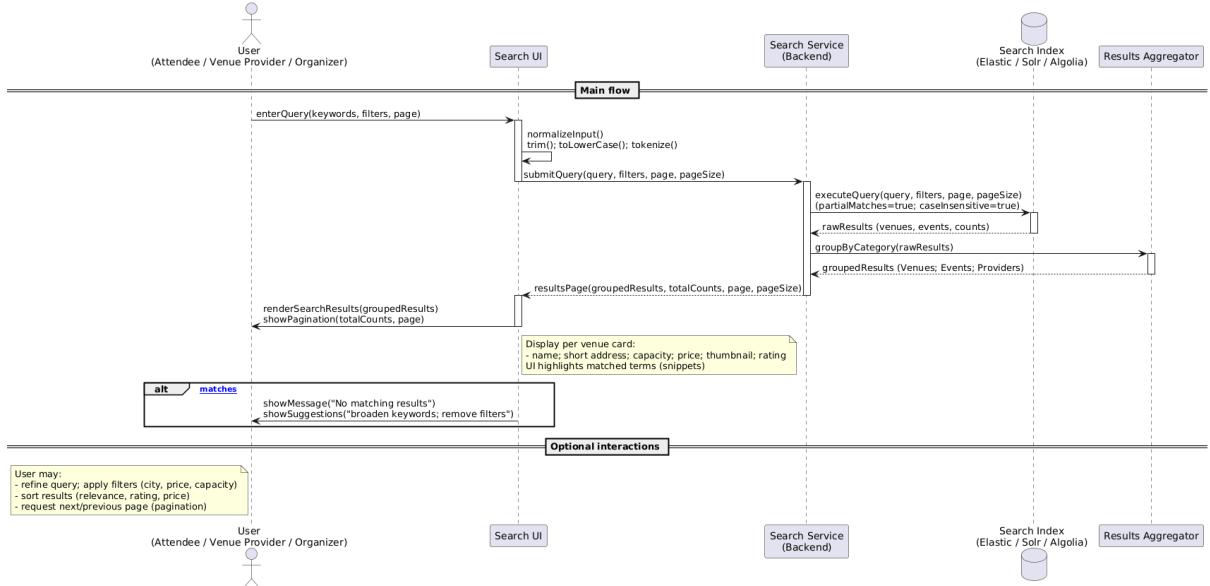


Figure 4.79 unified search for venues sequence diagram

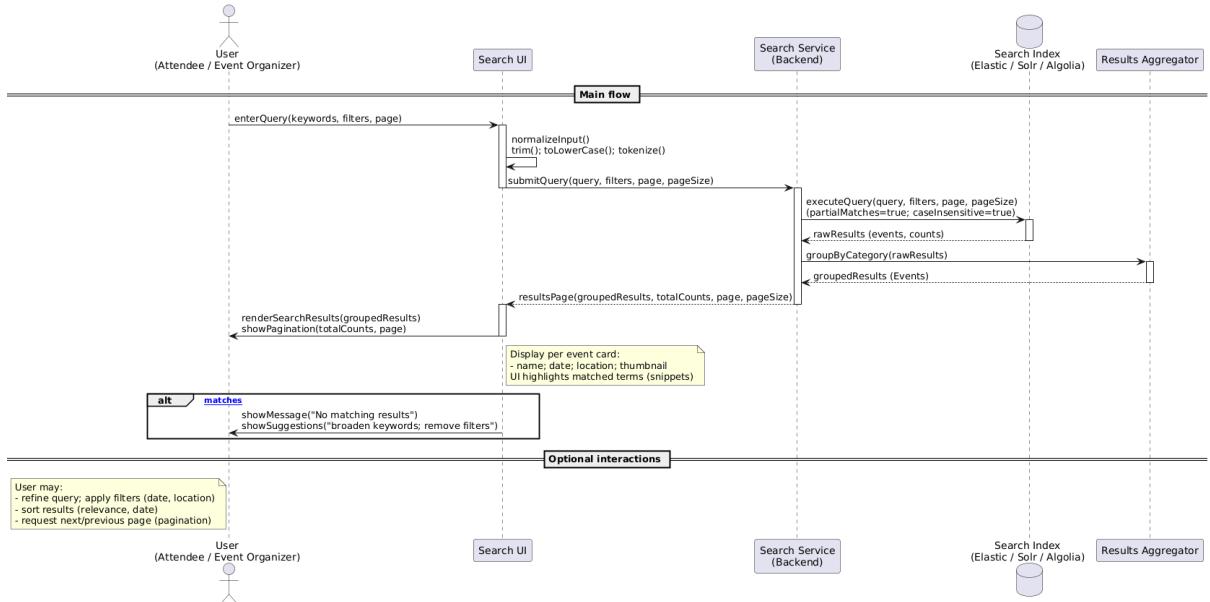


Figure 4.80 unified search for events sequence diagram

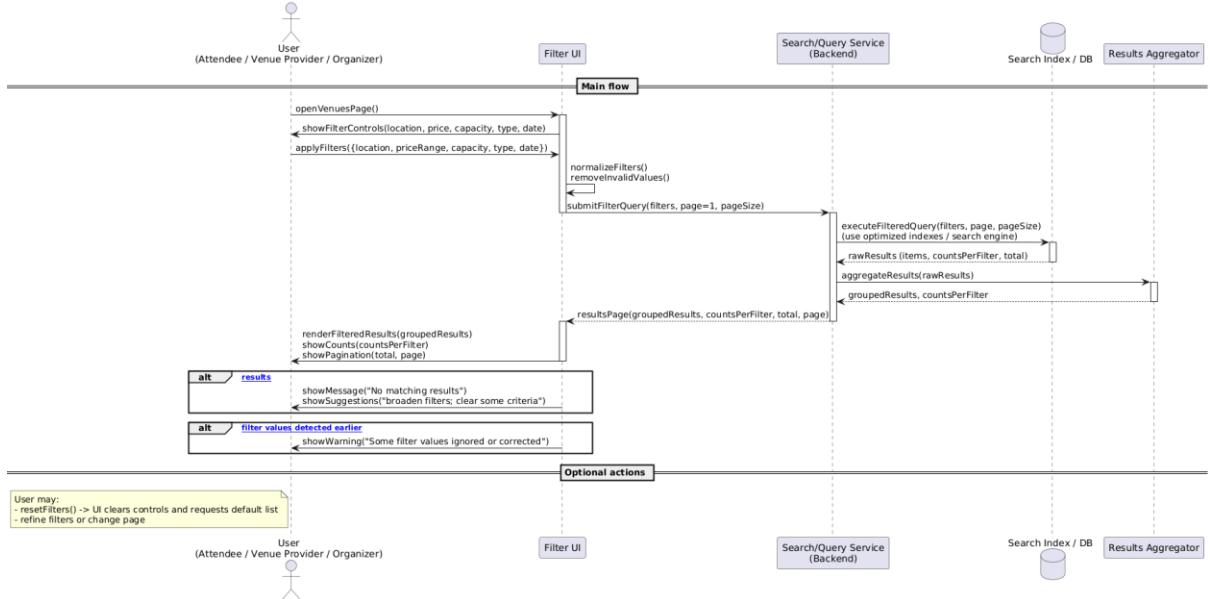


Figure 4.81 filter venues by specific criteria sequence diagram

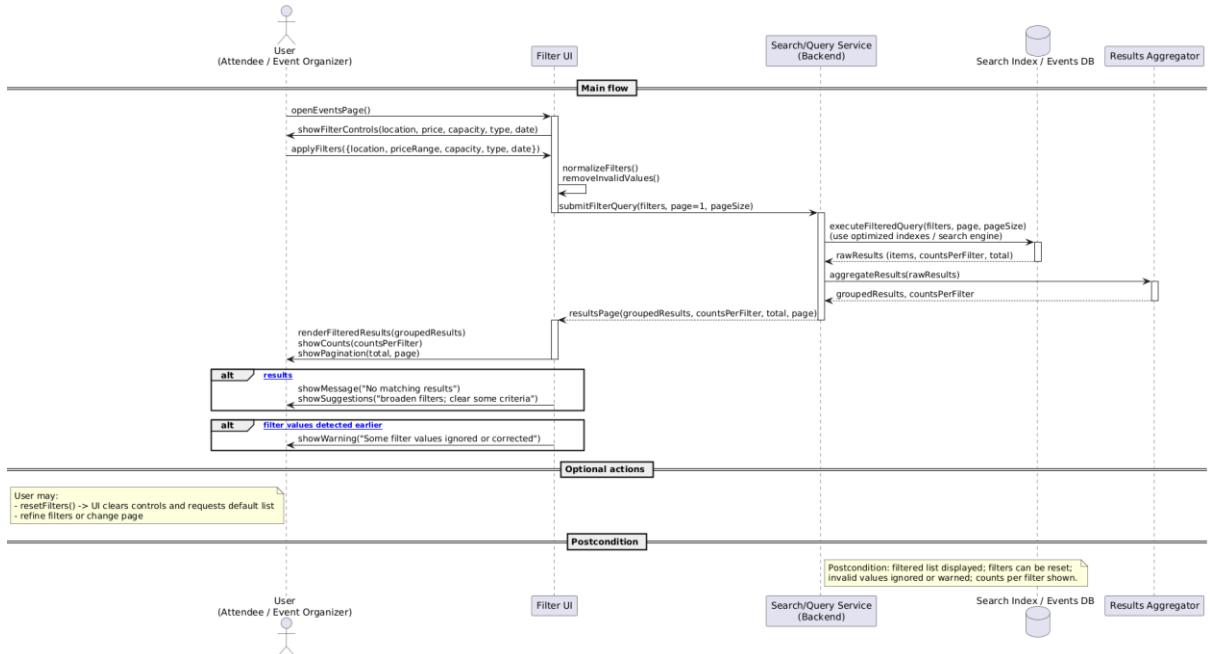


Figure 4.82 filter events by specific criteria sequence diagram

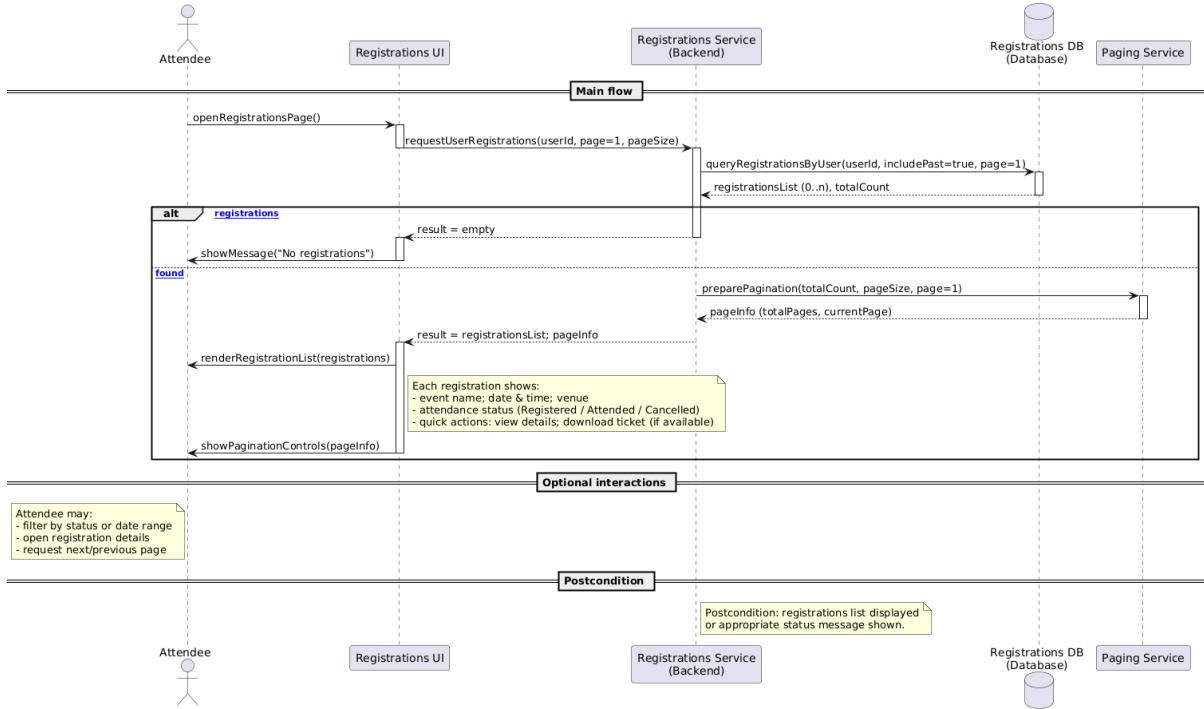


Figure 4.83 view registration sequence diagram

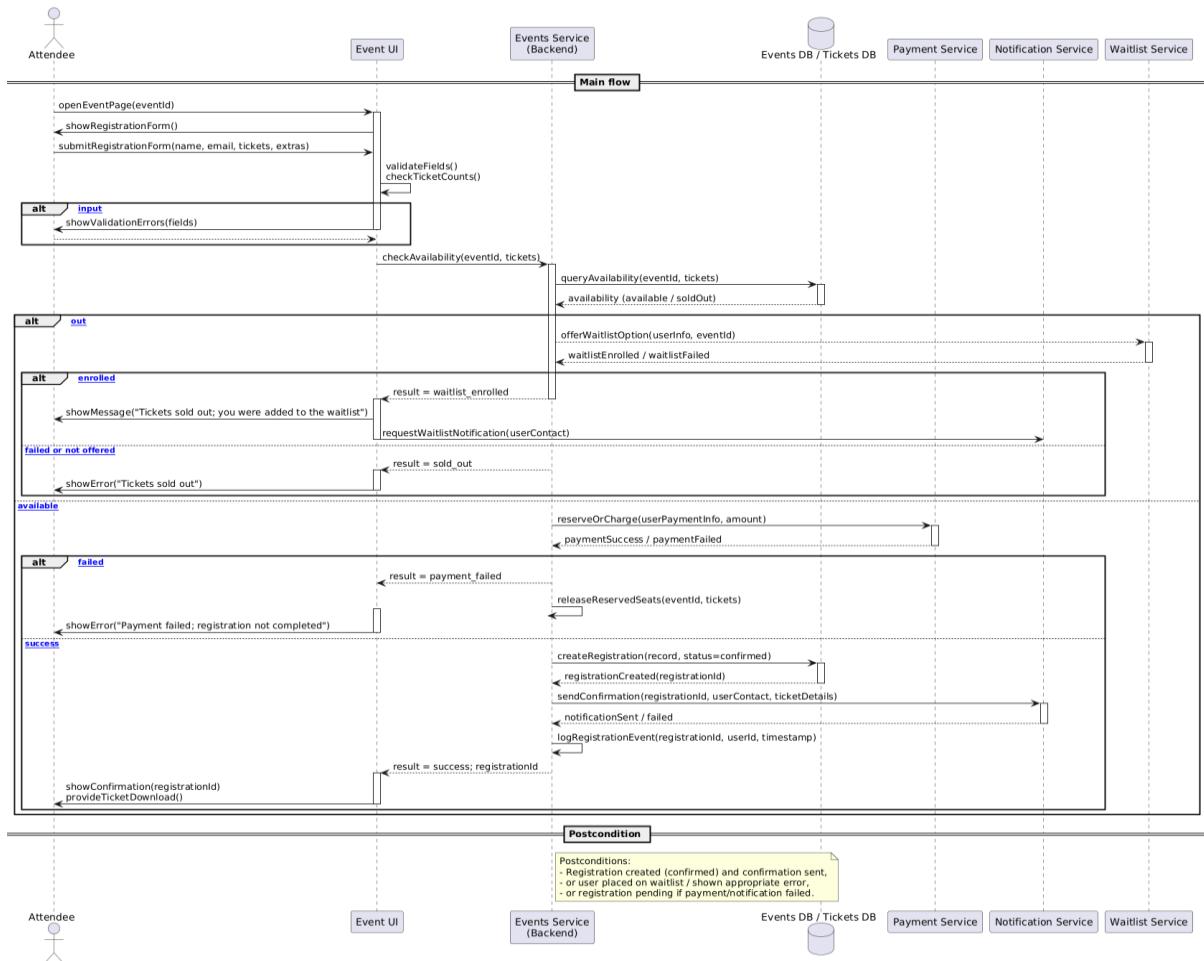


Figure 4.84 add registration sequence diagram

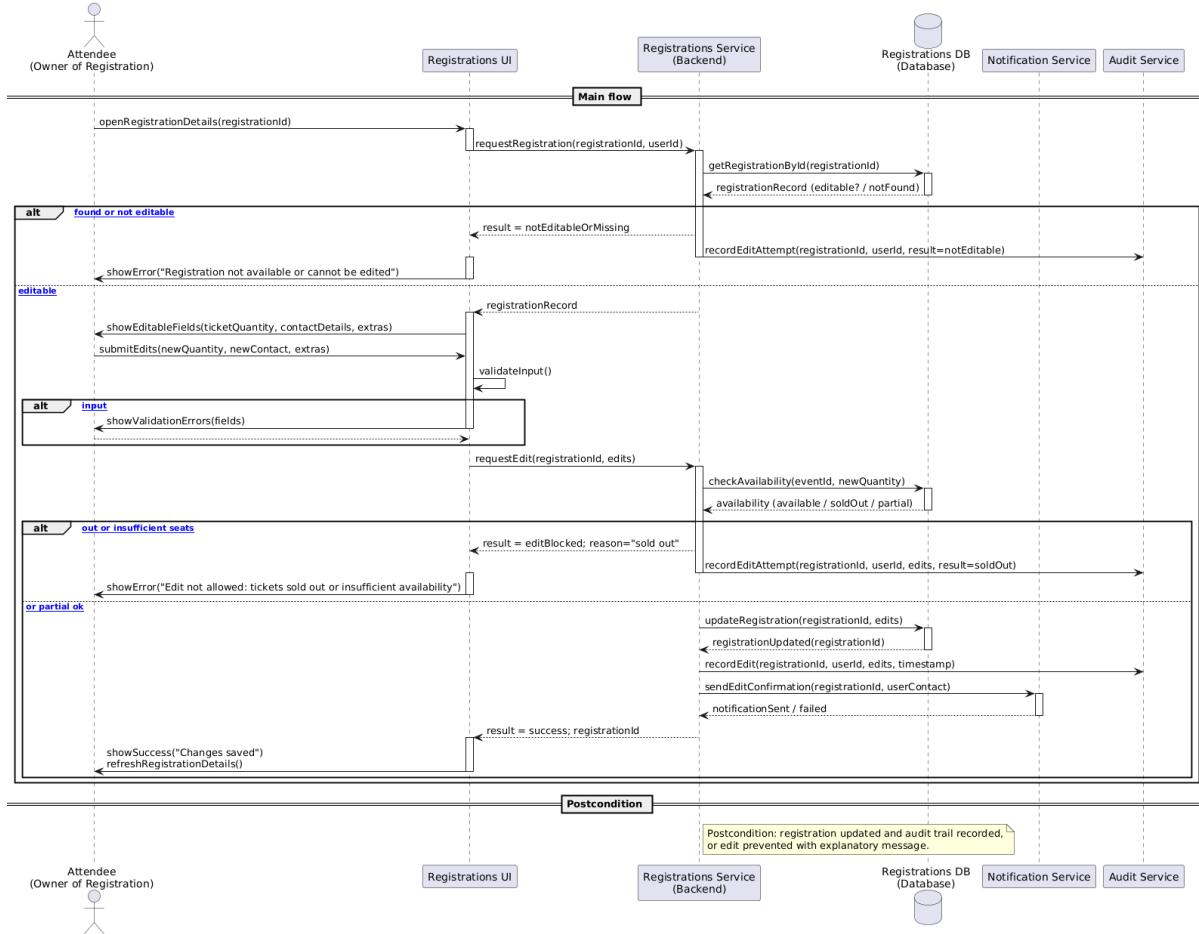


Figure 4.85 edit registration sequence diagram

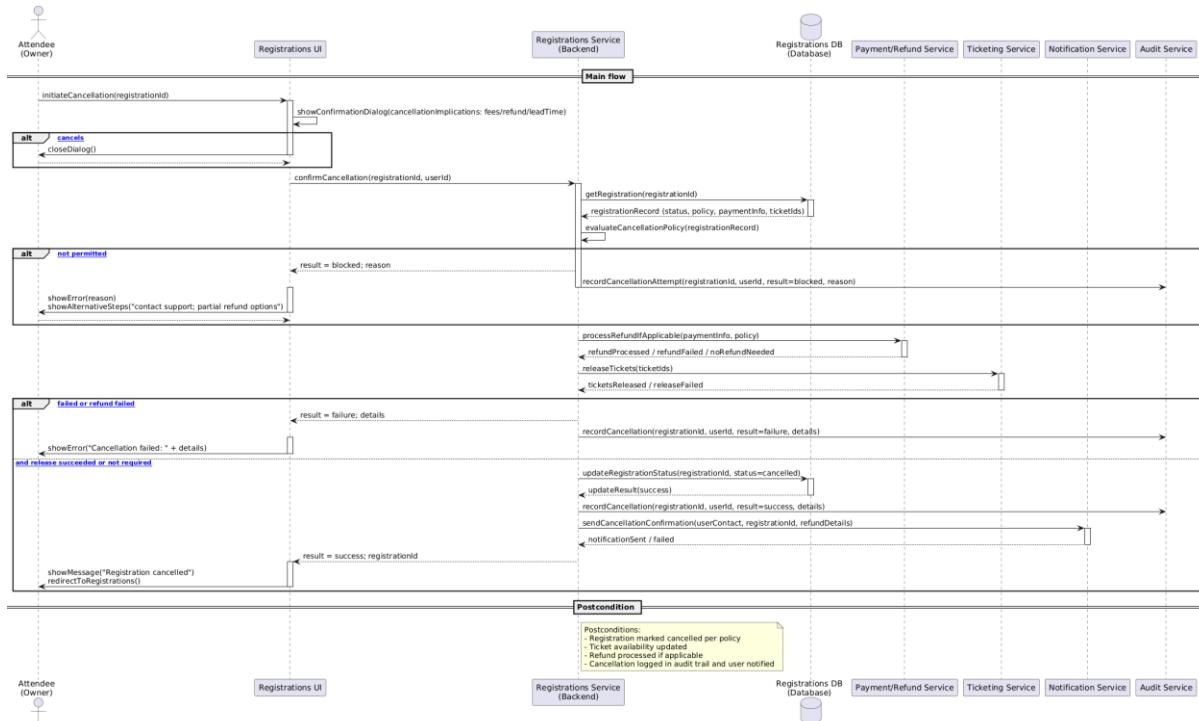


Figure 4.86 delete registration sequence diagram

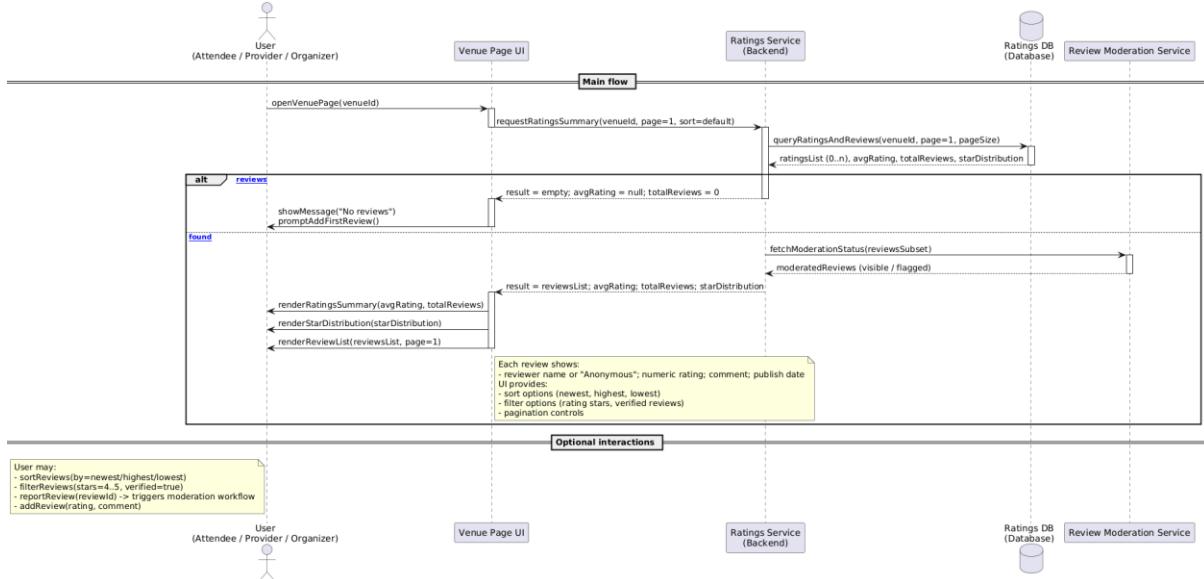


Figure 4.87 view venue ratings sequence diagram

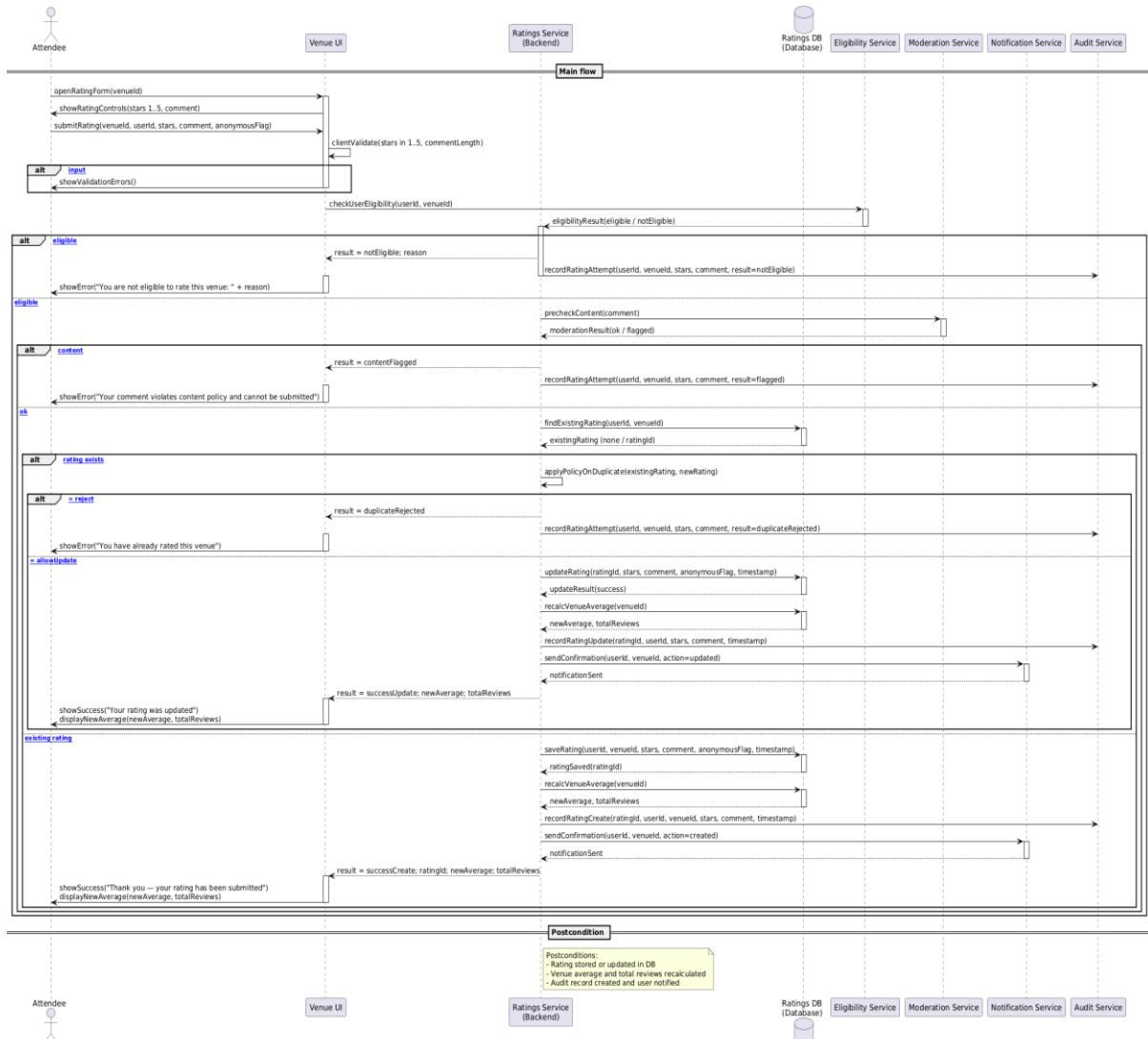


Figure 4.88 add venue rating sequence diagram

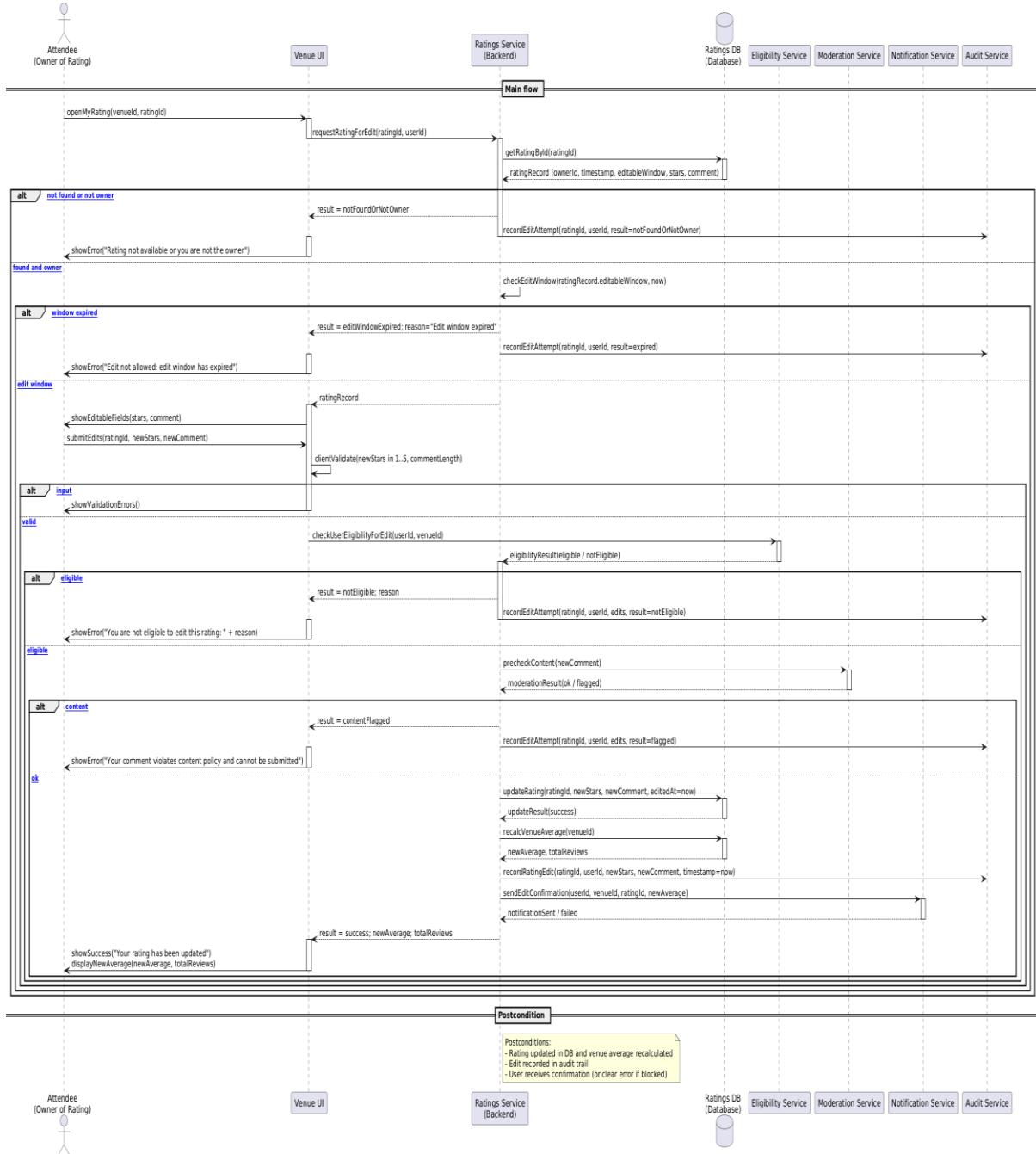


Figure 4.89 edit venue rating sequence diagram

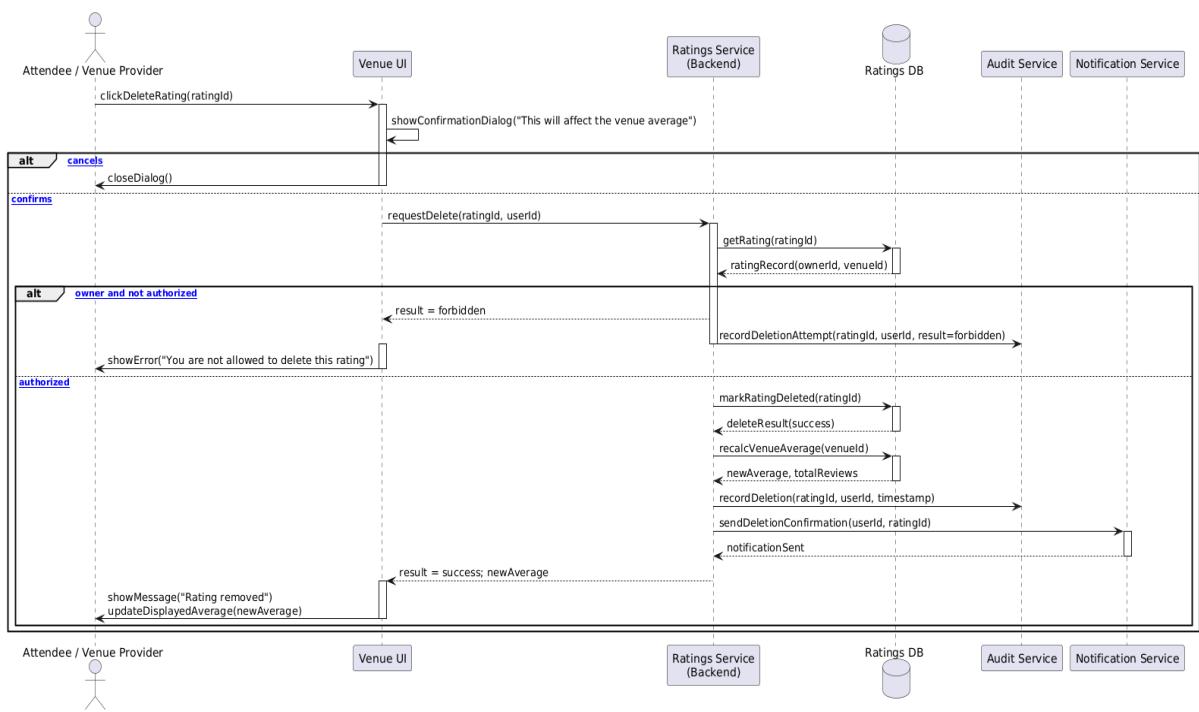


Figure 4.90 delete venue rating sequence diagram

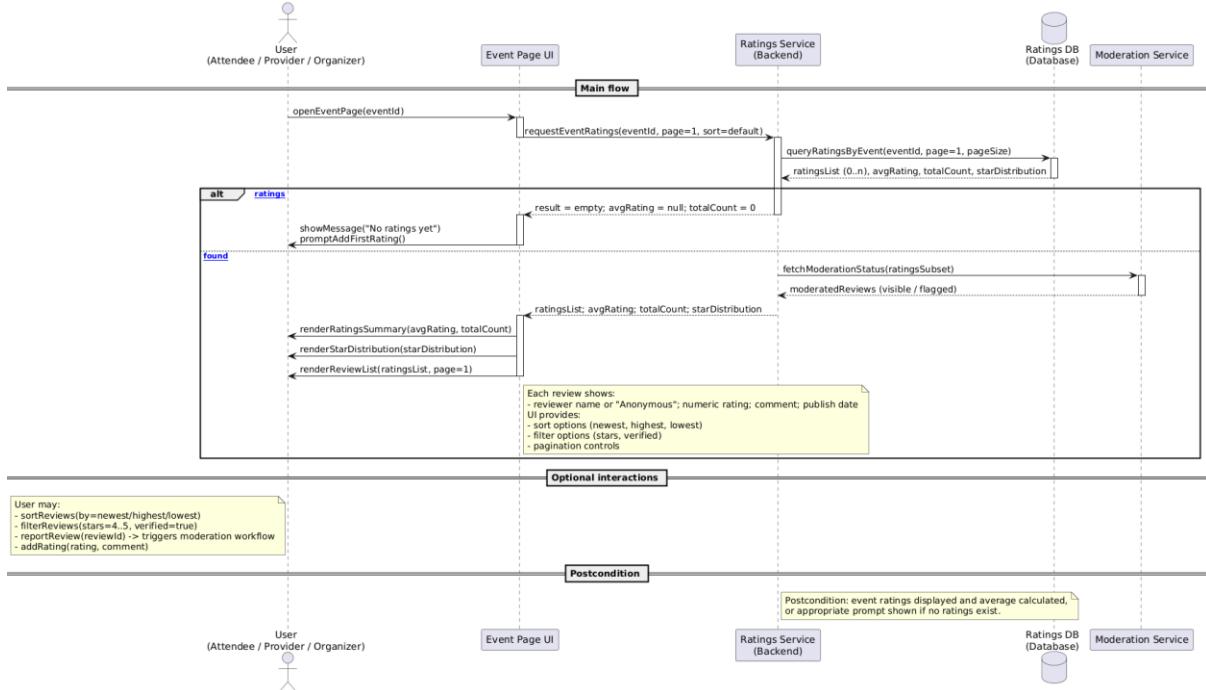


Figure 4.91 view event ratings sequence diagram

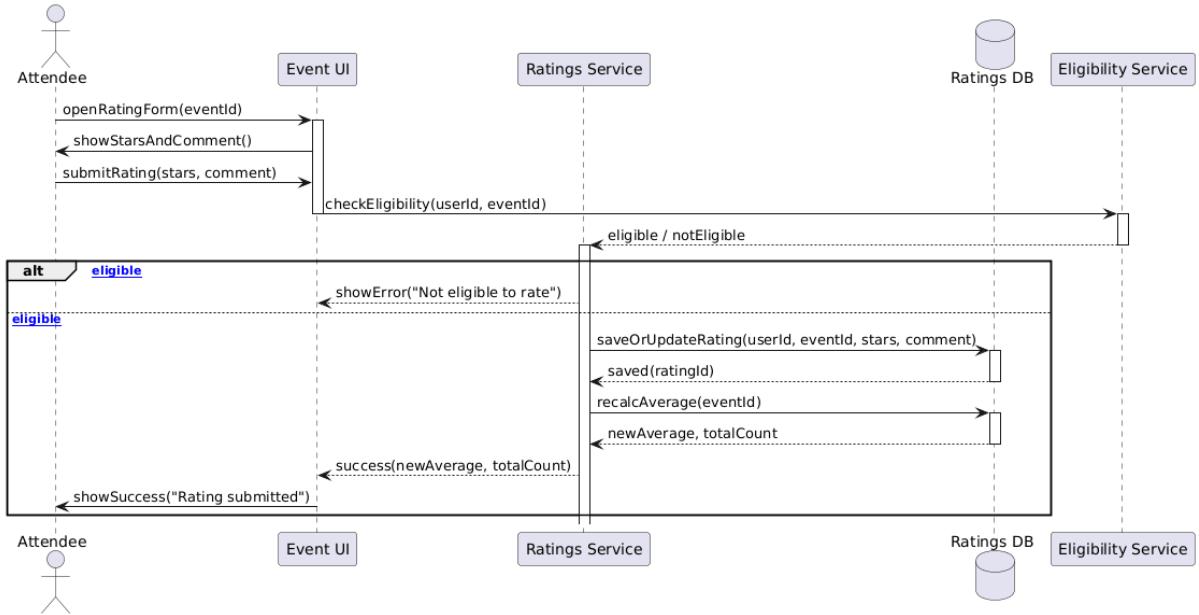


Figure 4.92 add event rating sequence diagram

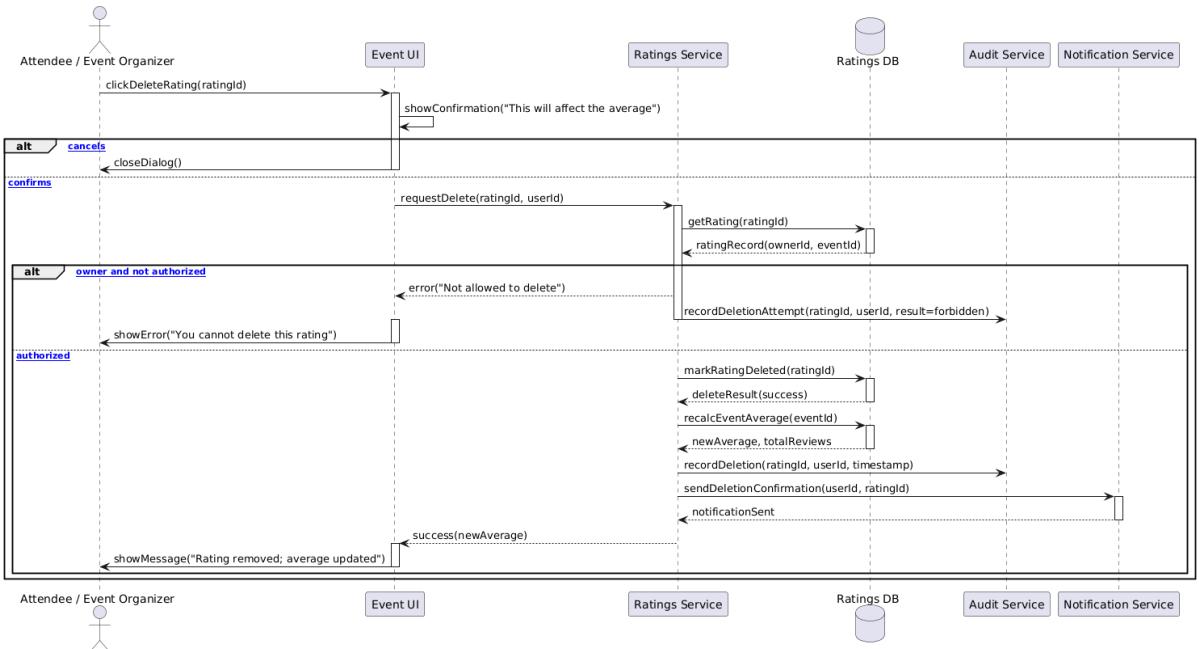


Figure 4.93 edit event rating sequence diagram

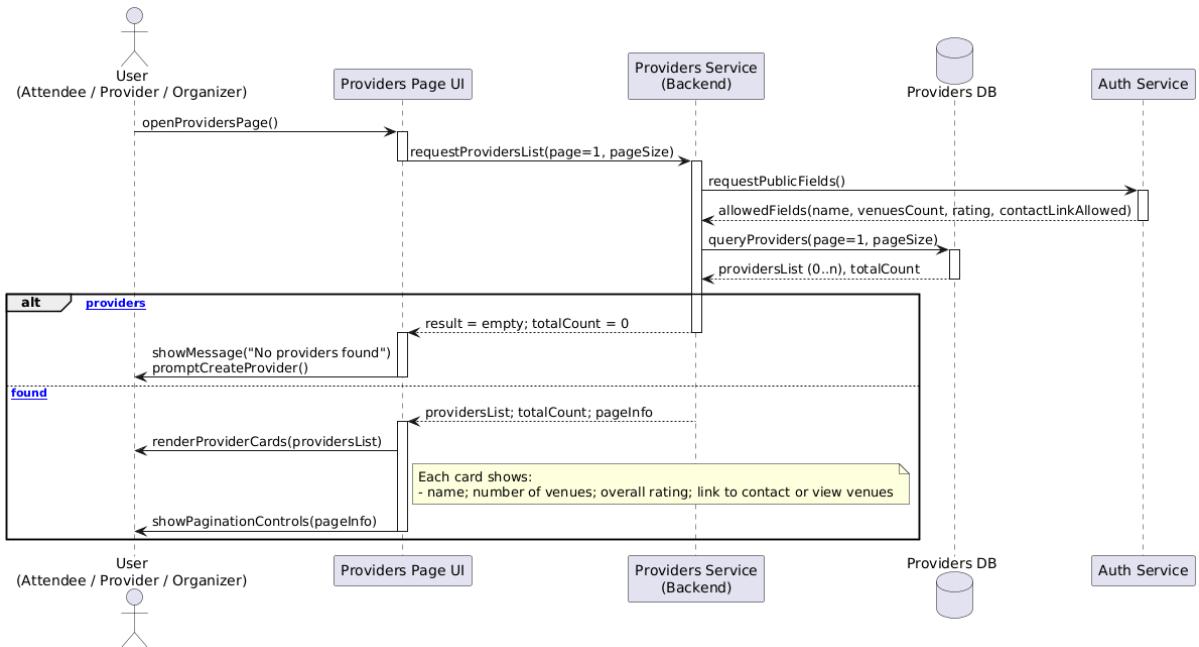


Figure 4.94 delete event rating sequence diagram

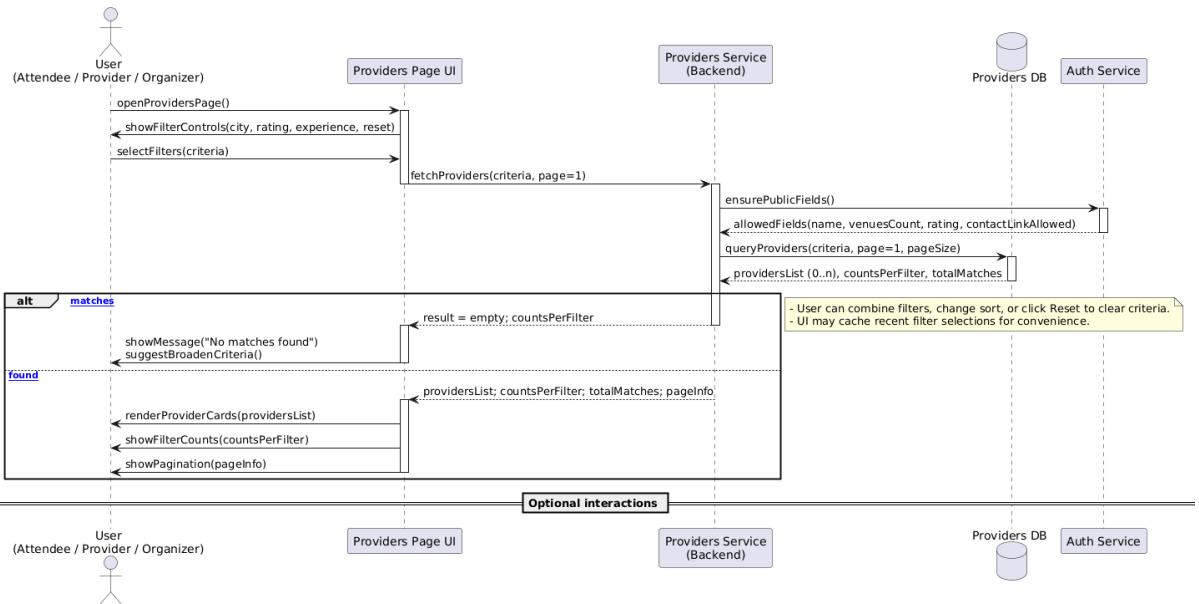


Figure 4.95 view organizers sequence diagram

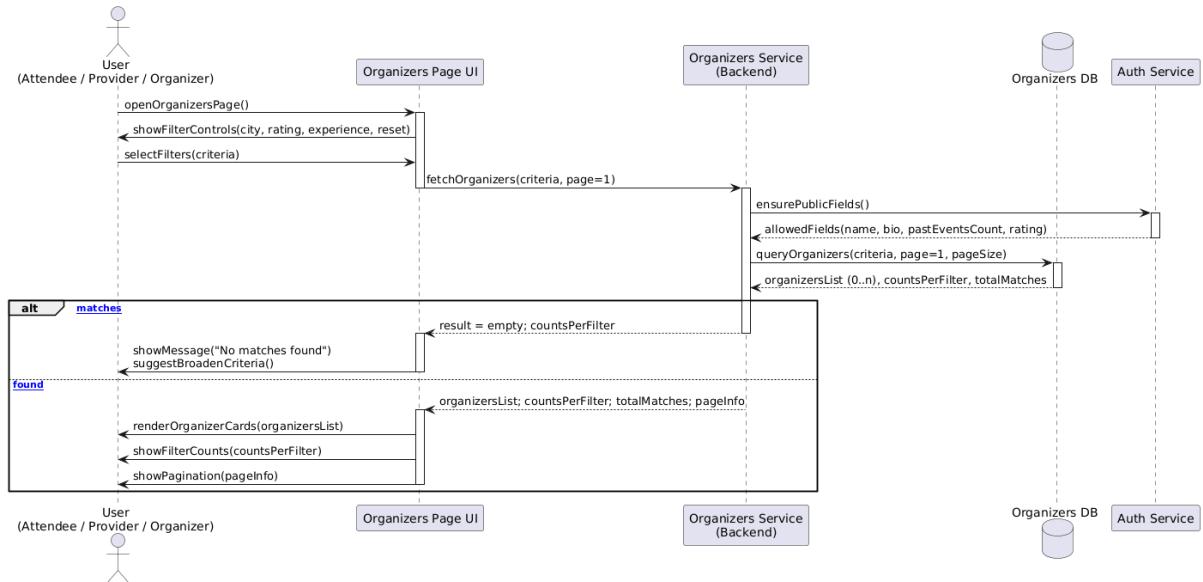


Figure 4.96 view providers sequence diagram

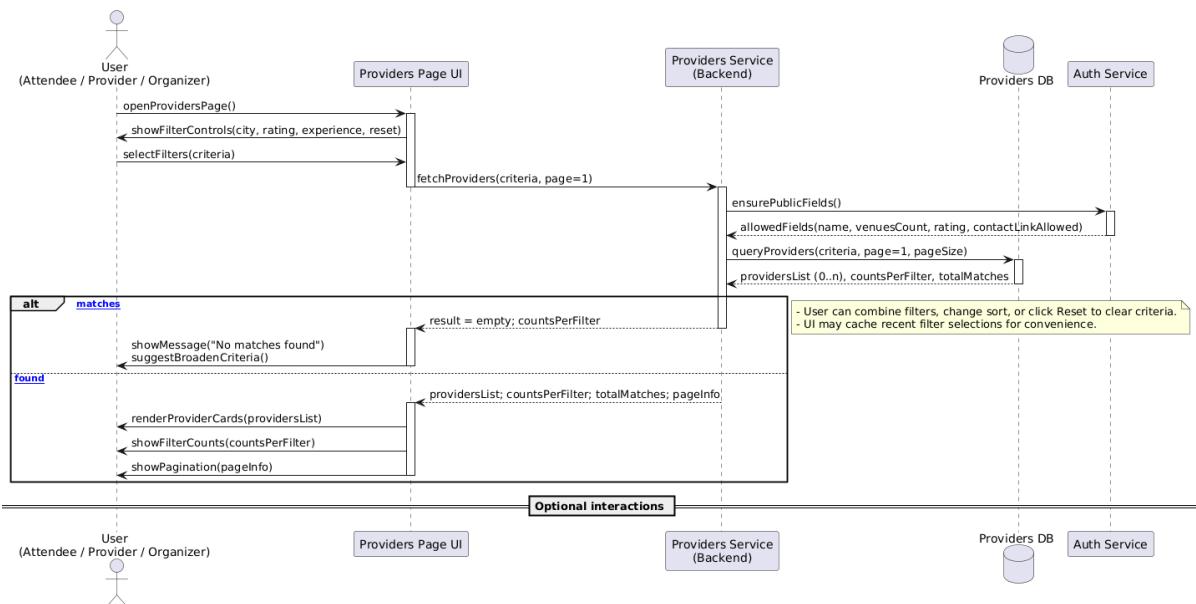


Figure 4.97 filter providers by specific criteria sequence diagram

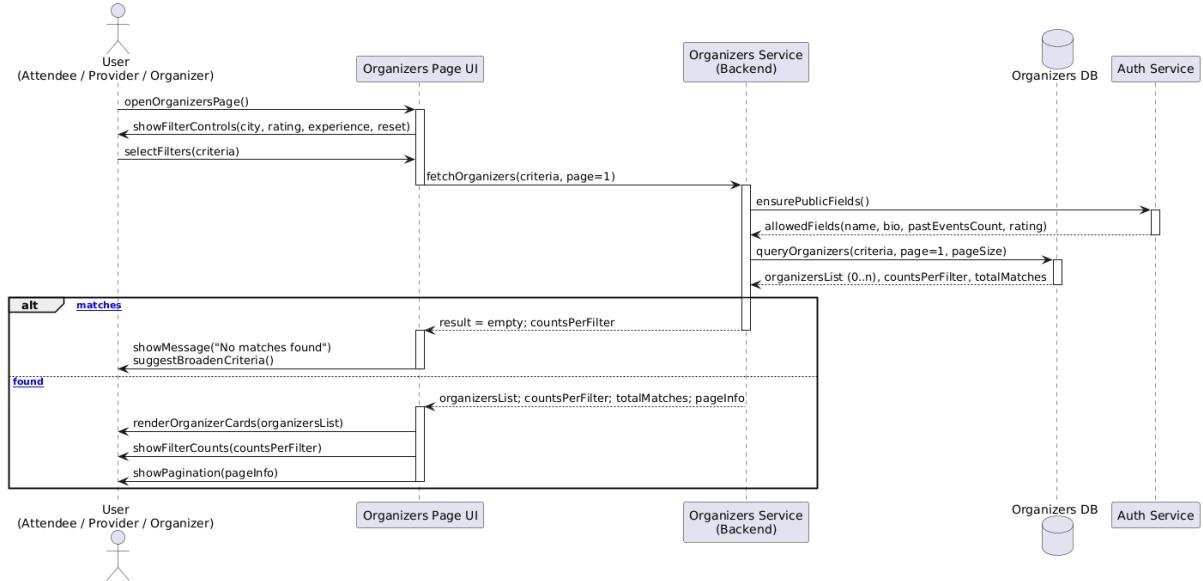


Figure 4.98 filter Organizers by specific criteria sequence diagram

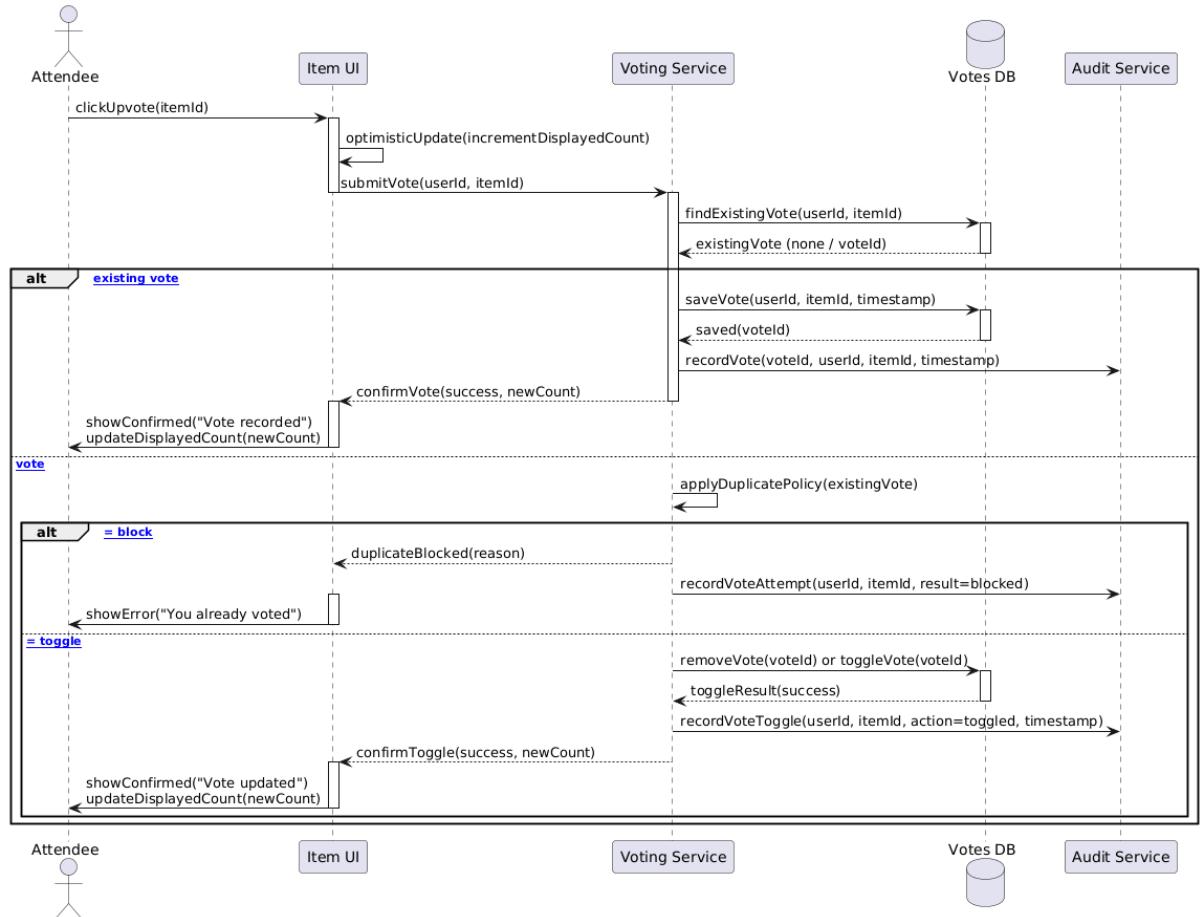


Figure 4.99 upvote sequence diagram

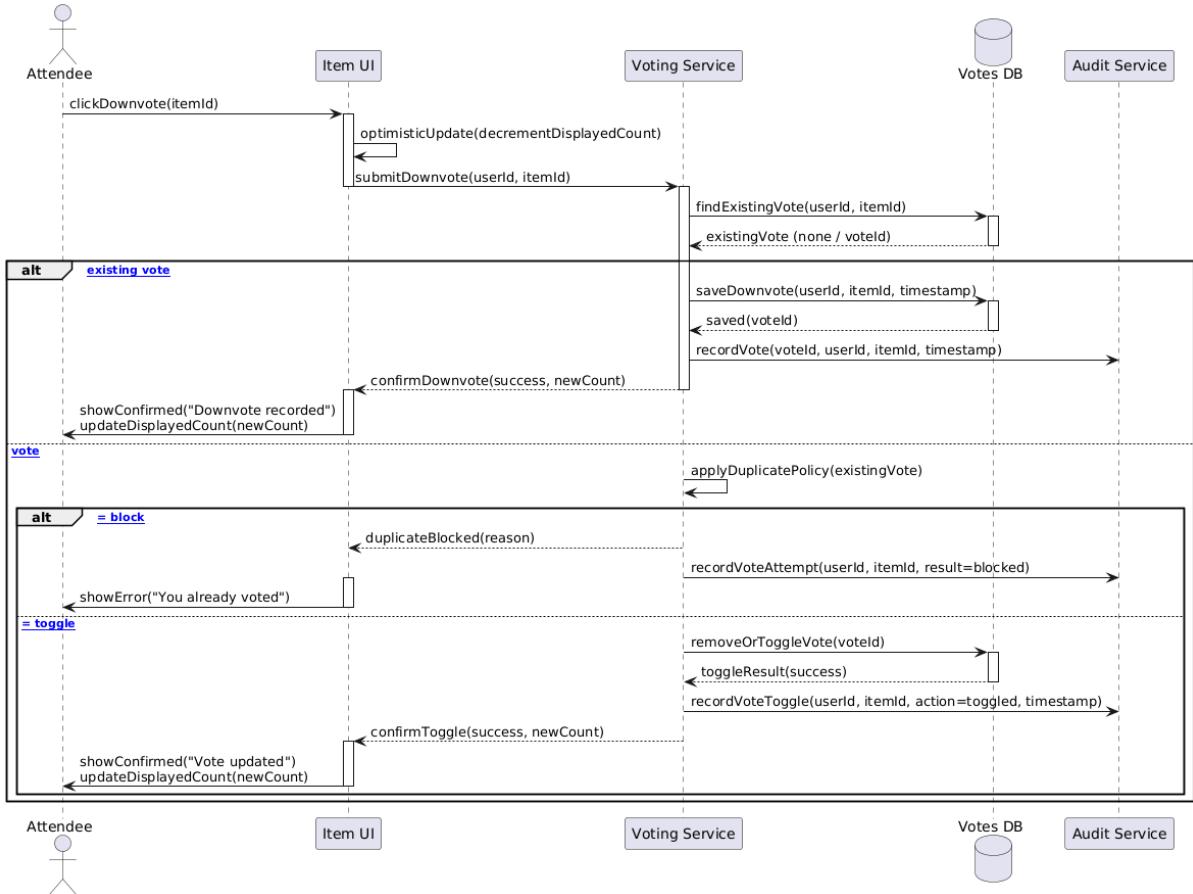


Figure 4.100 downvote sequence diagram

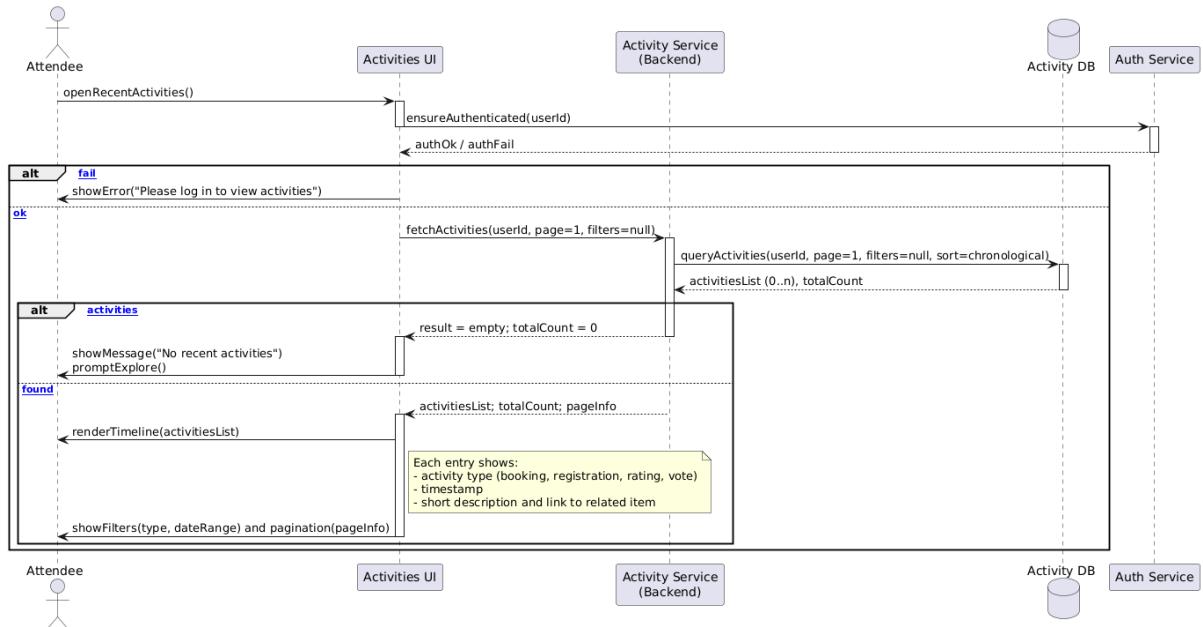


Figure 4.101 view recent activities sequence diagram

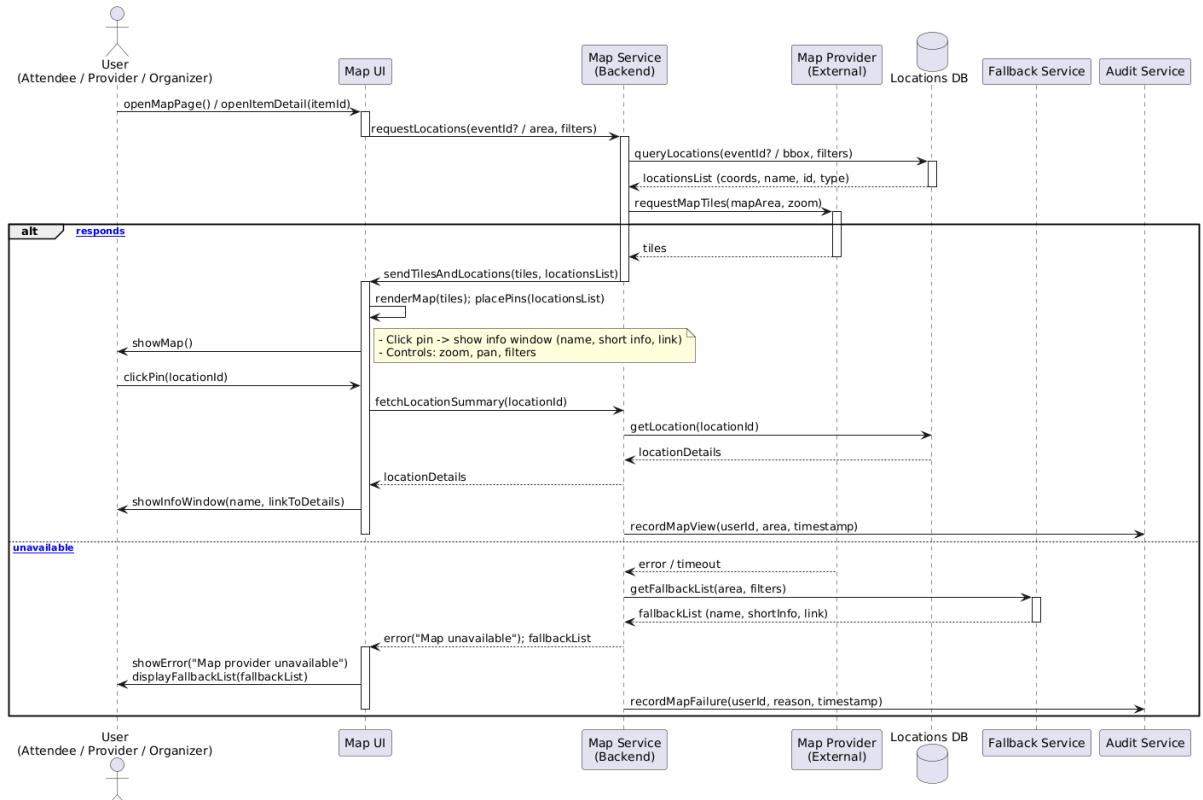


Figure 4.102 display map of event and venue locations sequence diagram

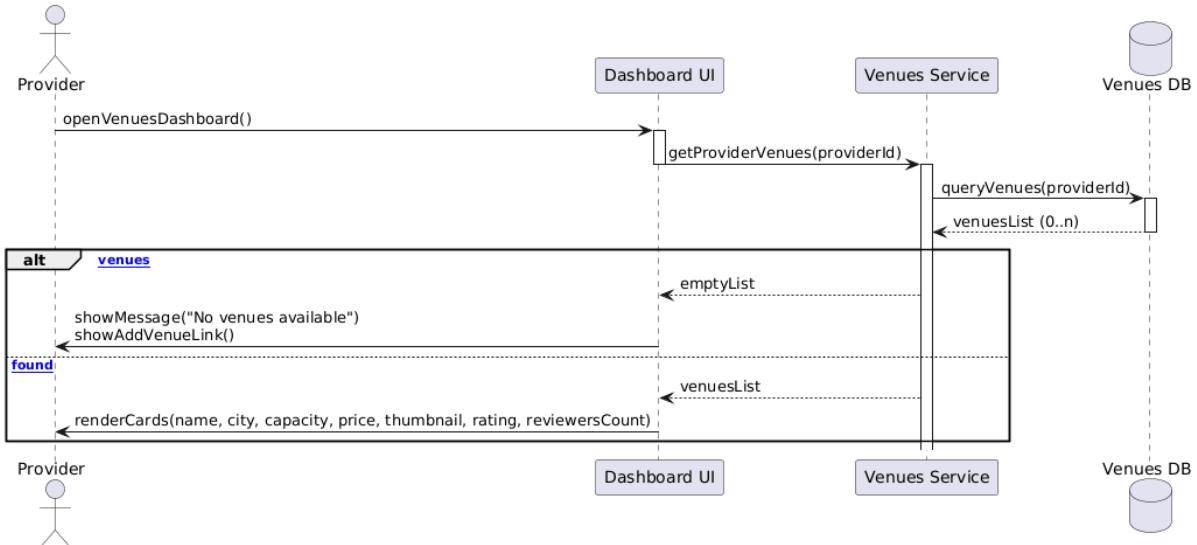


Figure 4.103 browse venues (provider) sequence diagram

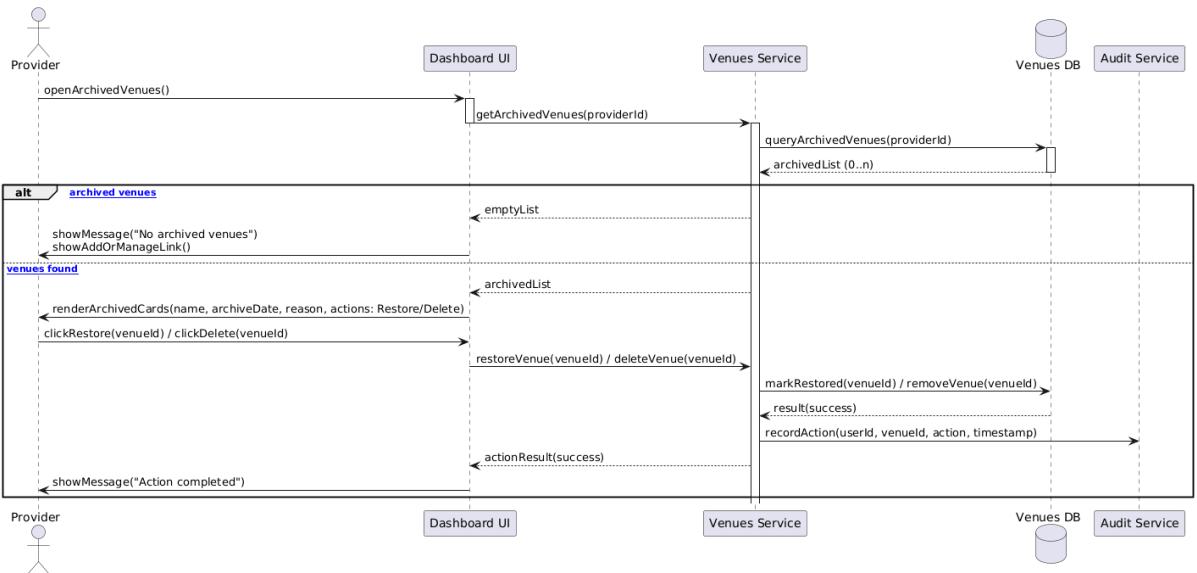


Figure 4.104 brows archivd venues sequence diagram

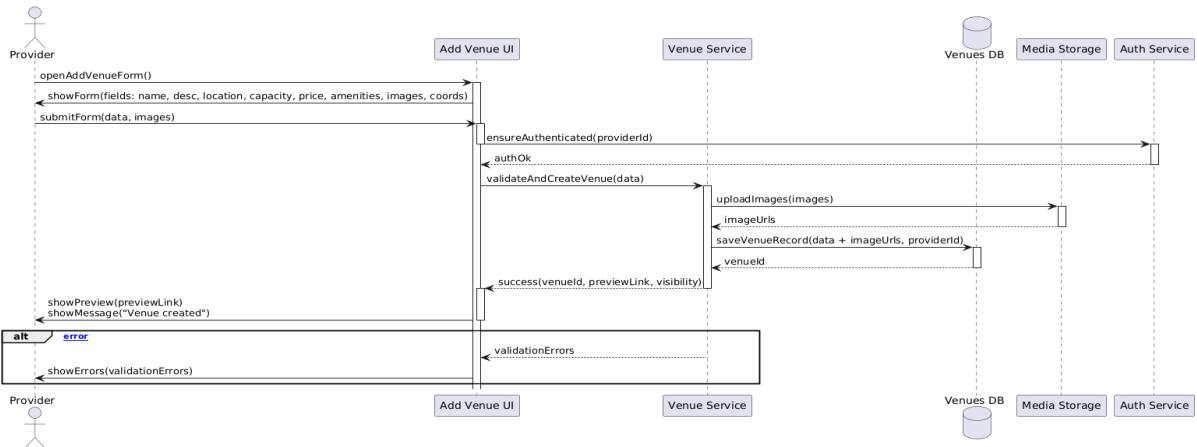


Figure 4.105 add venue sequence diagram

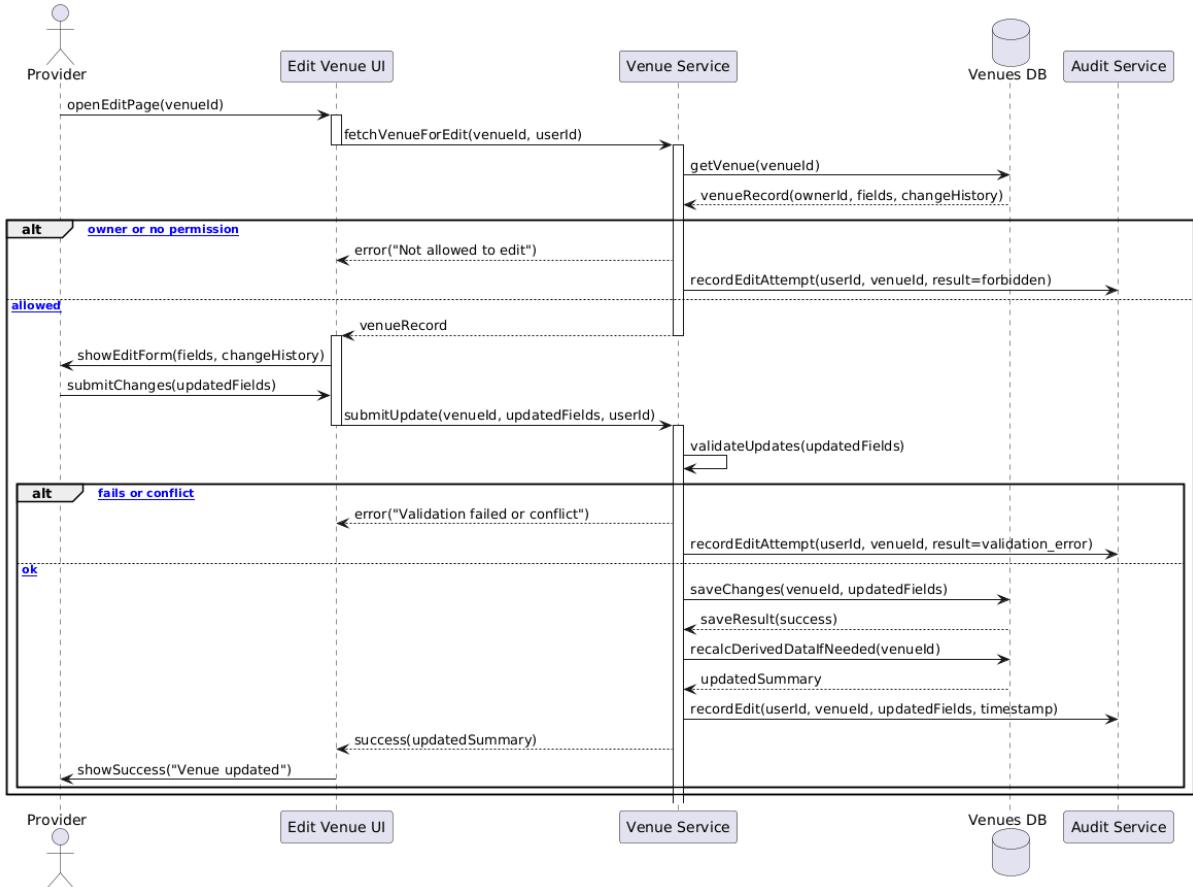


Figure 4.106 edit venue sequence diagram

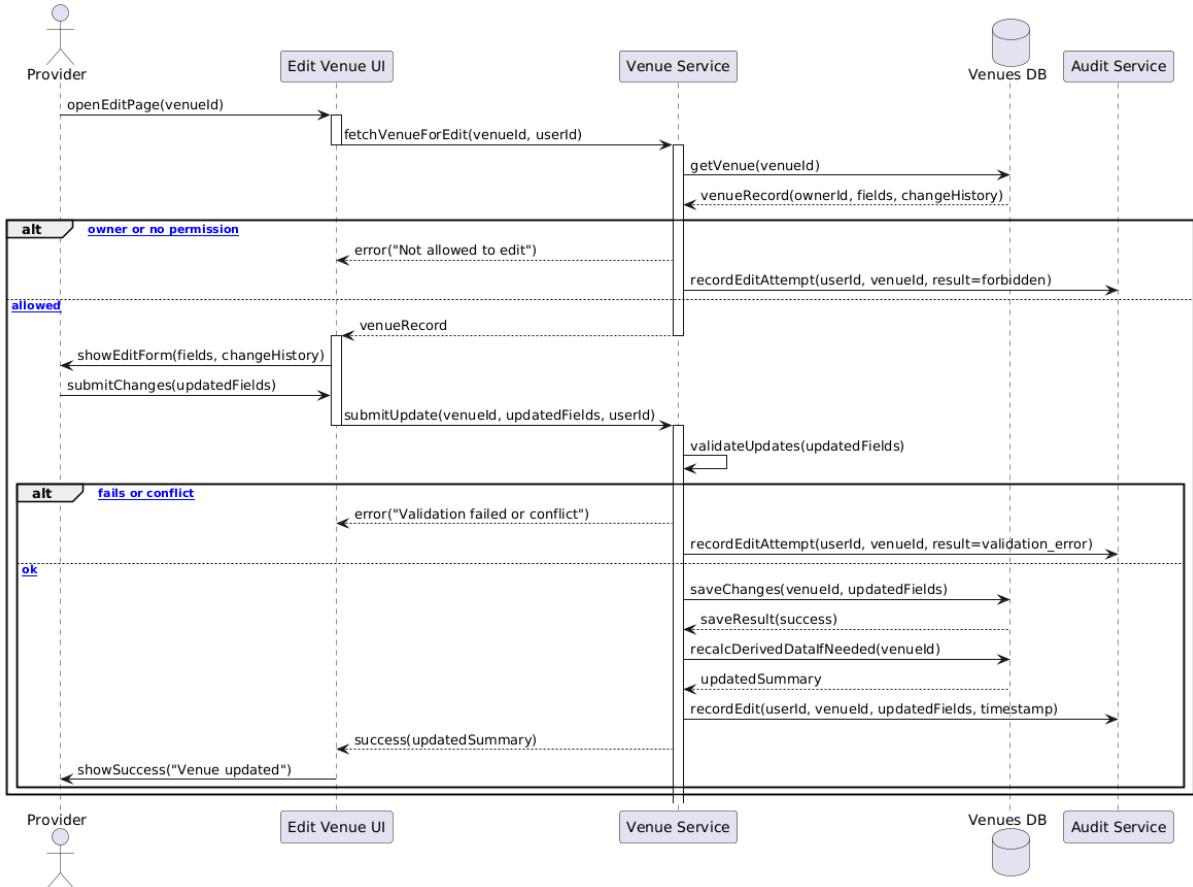


Figure 4.107 delete venue sequence diagram

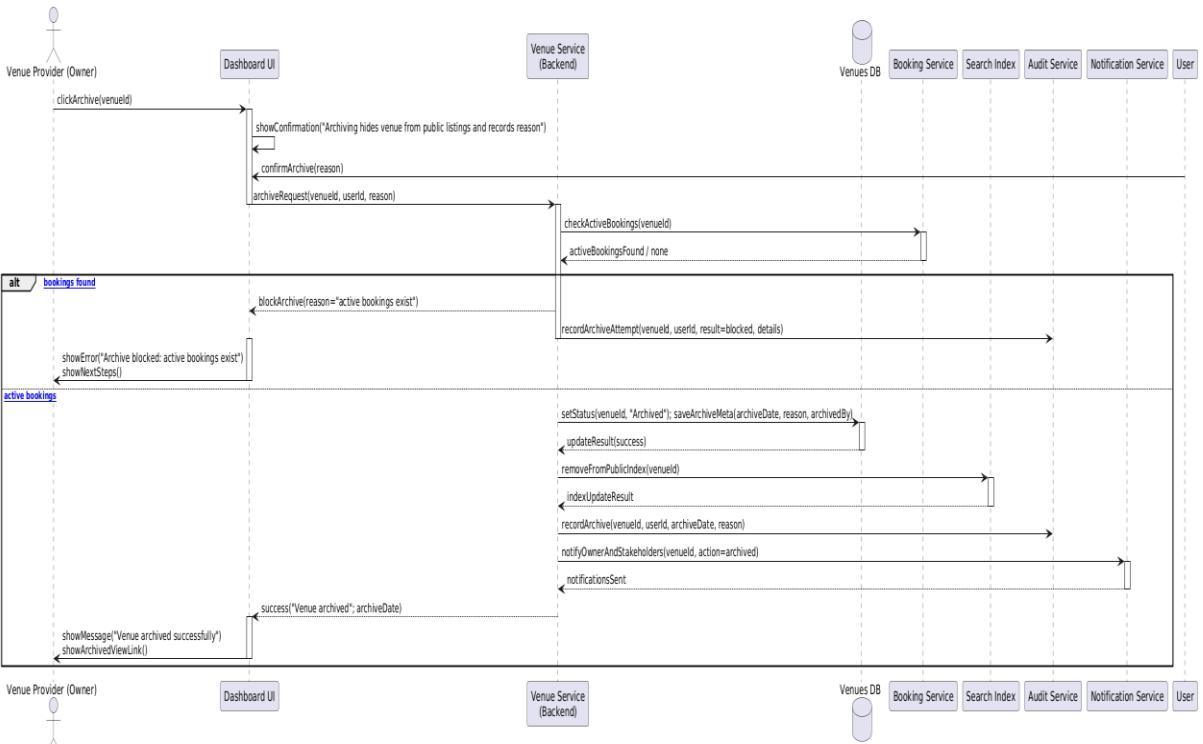


Figure 4.108 archive venue sequence diagram

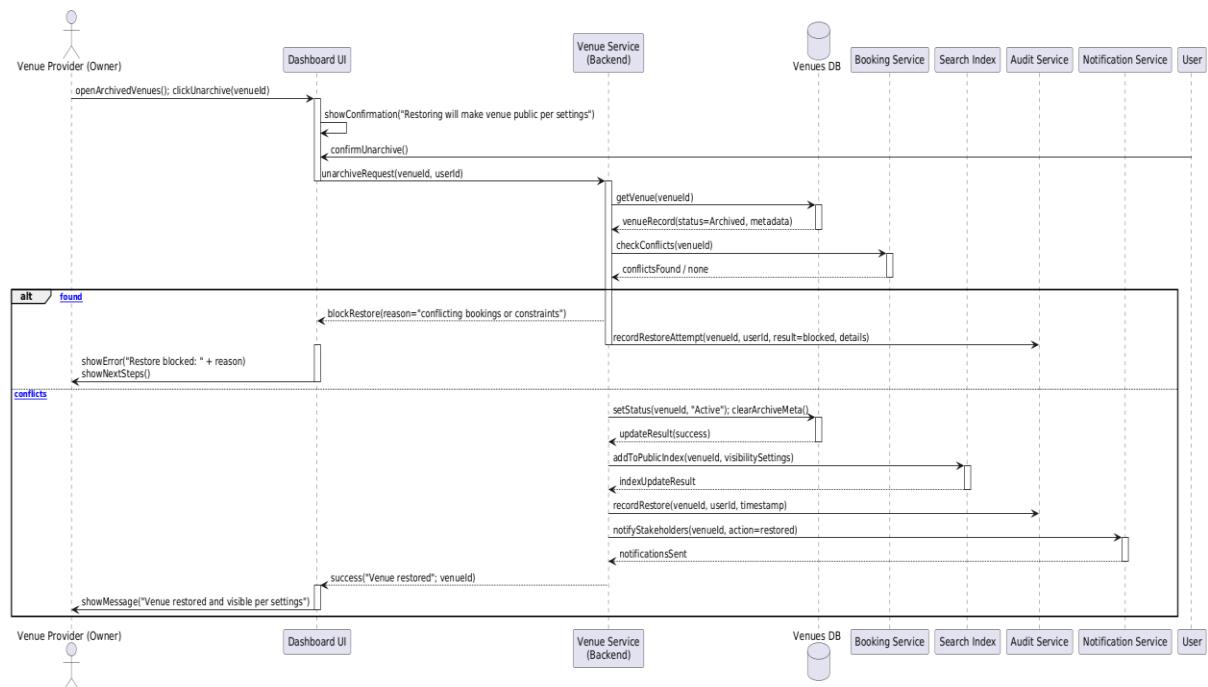


Figure 4.109 unarchive venue sequence diagram

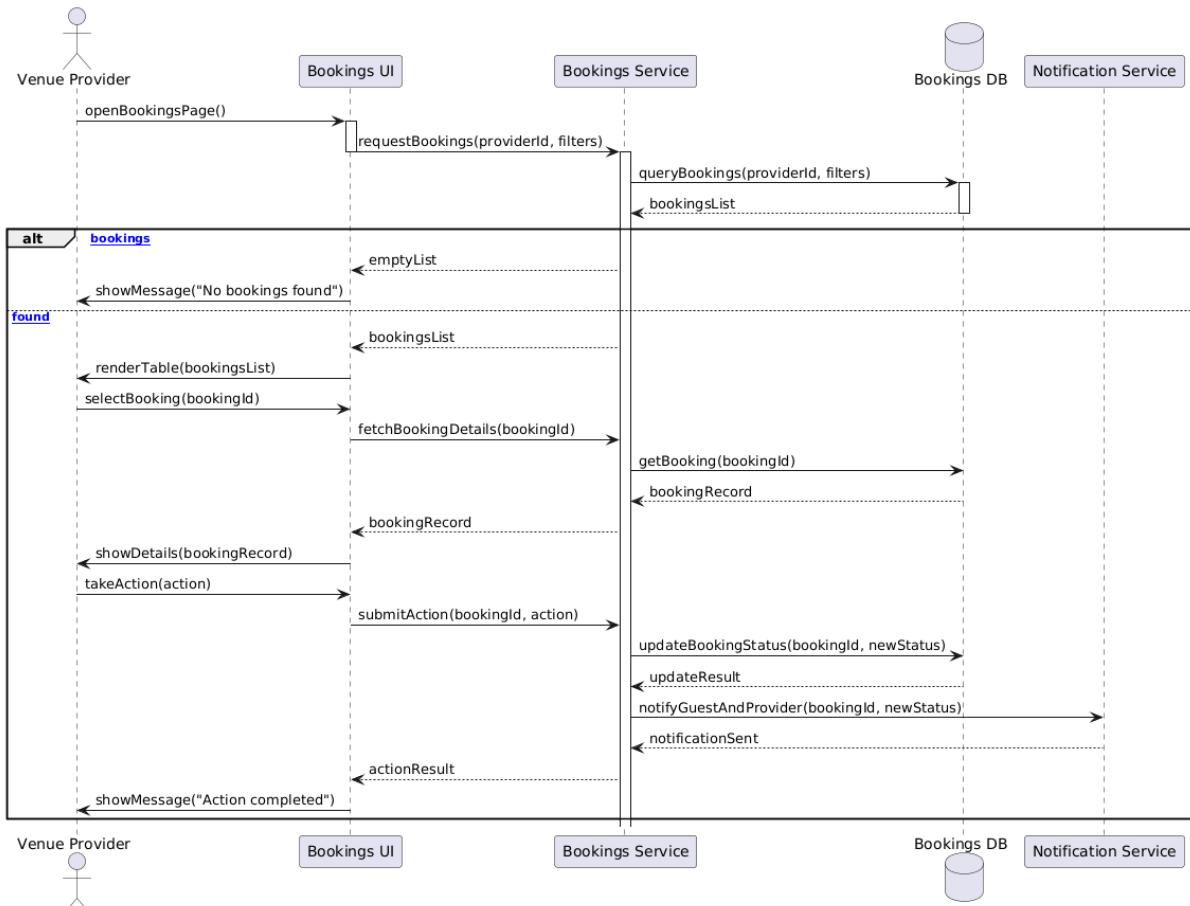


Figure 4.110 browse bookings sequence diagram

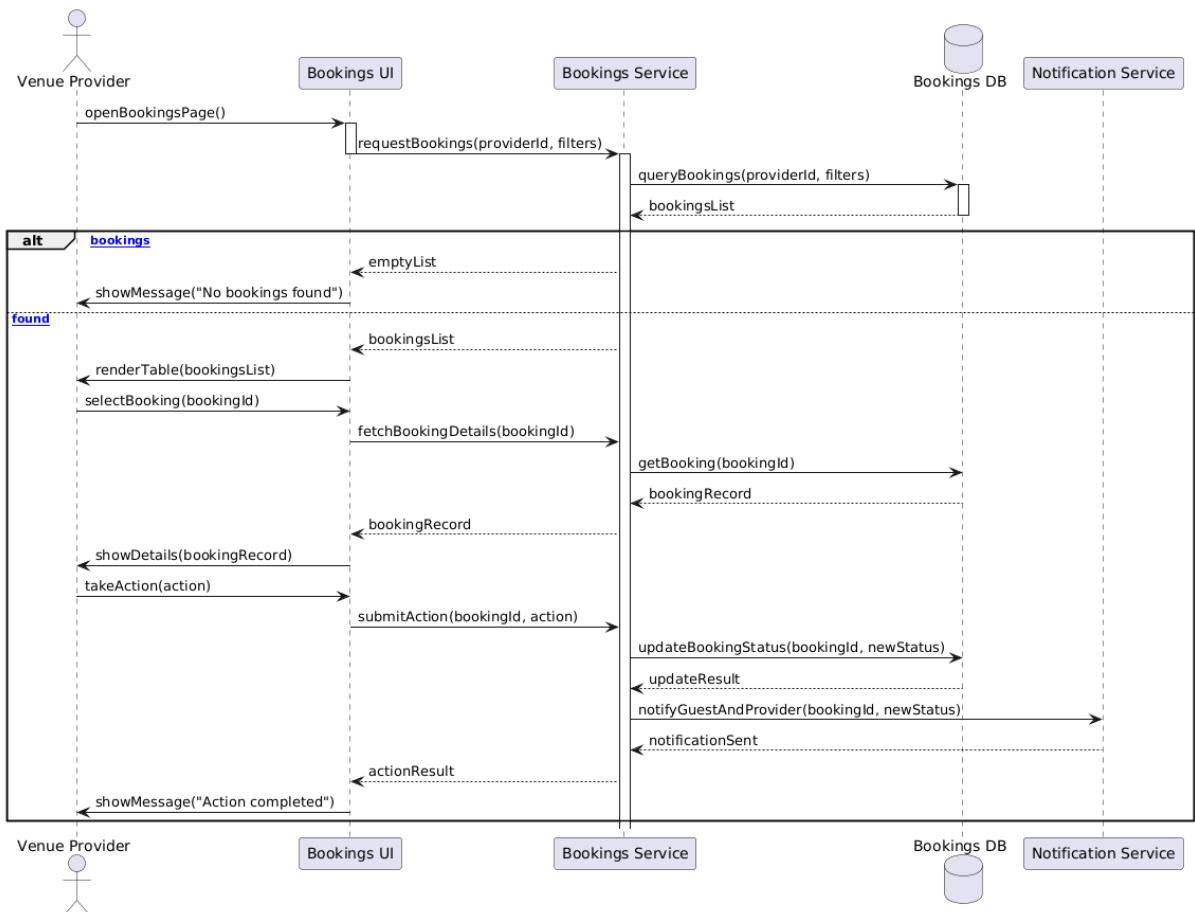


Figure 4.111 accept booking sequence diagram

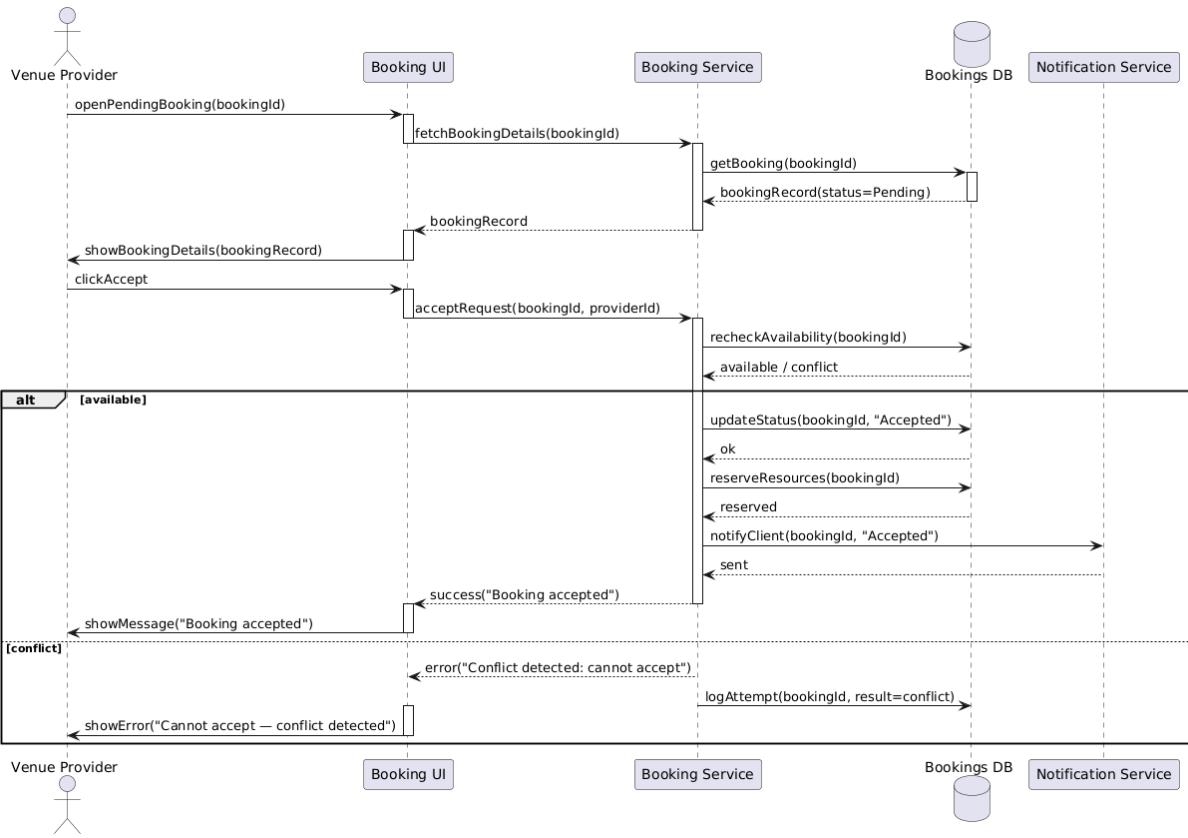


Figure 4.112 reject booking sequence diagram

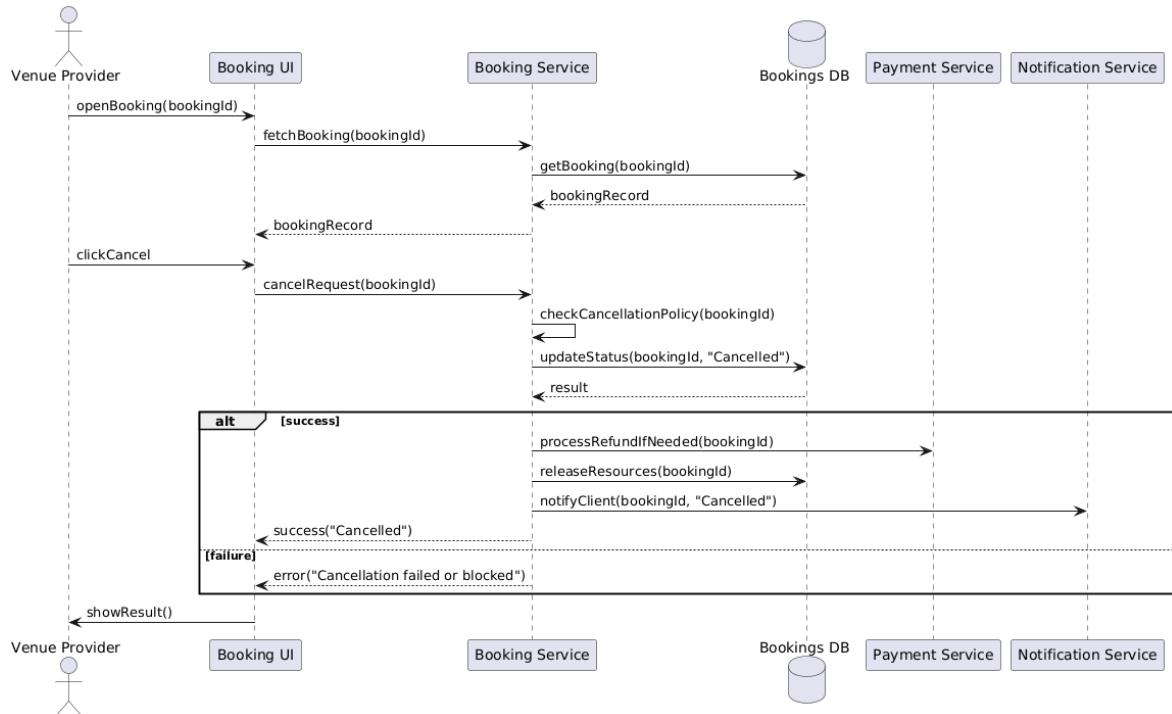


Figure 4.113 cancel booking sequence diagram

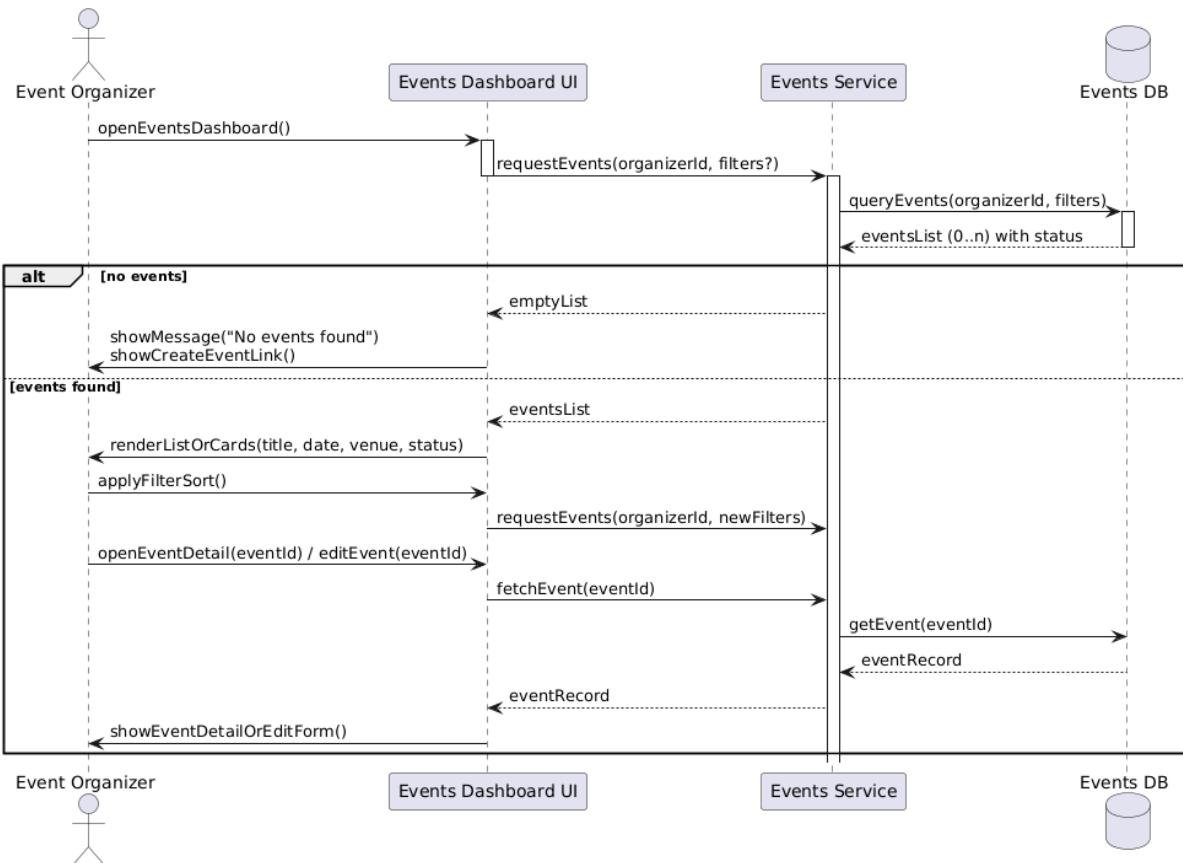


Figure 4.114 browse events (organizer) sequence diagram

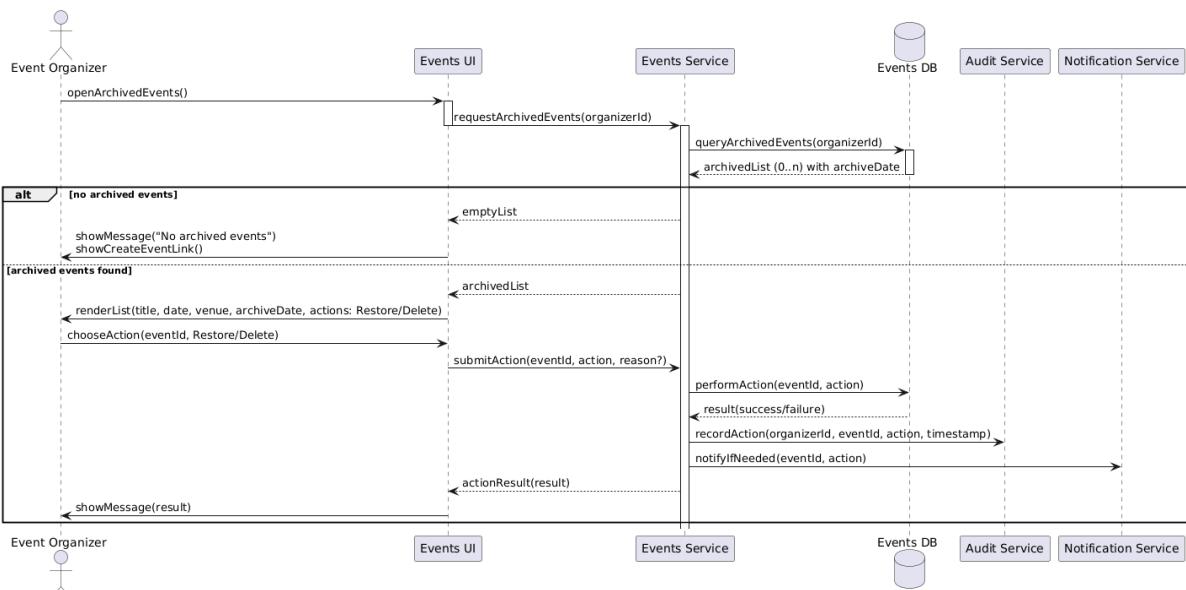


Figure 4.115 browse archived events sequence diagram

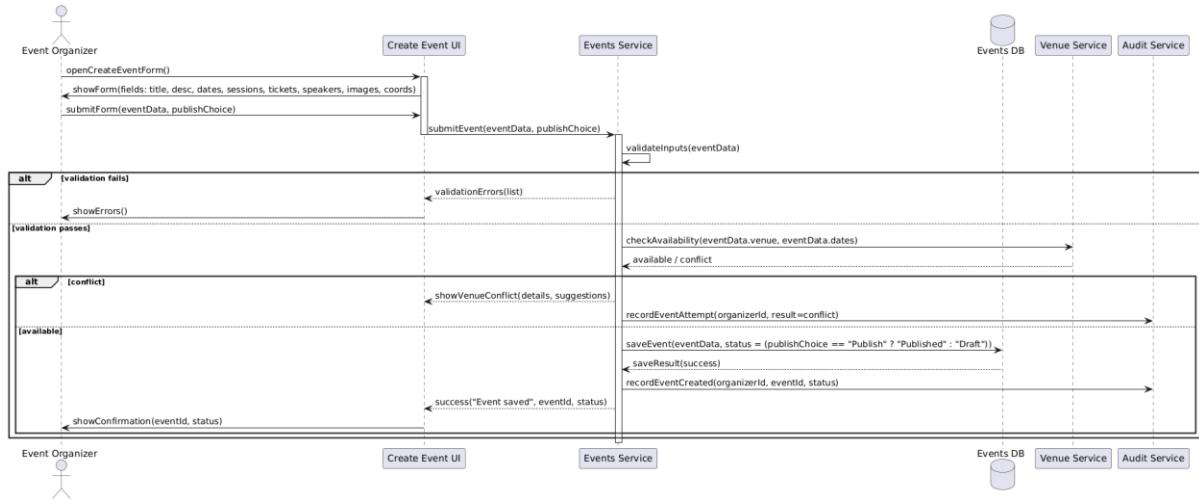


Figure 4.116 create event sequence diagram

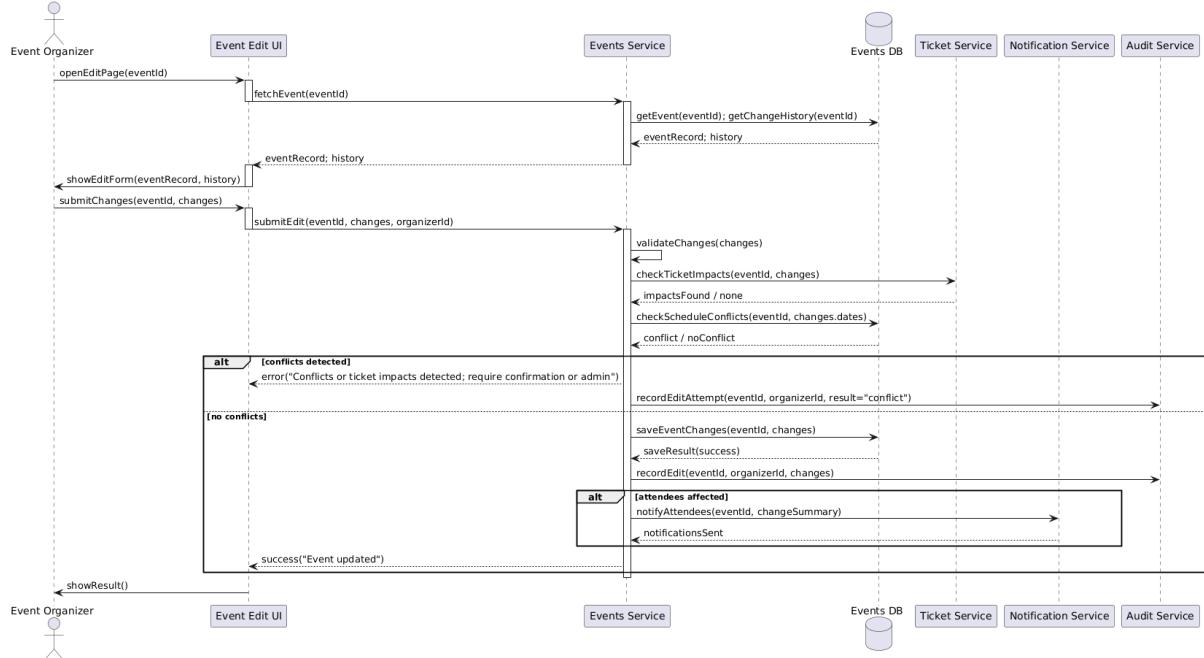


Figure 4.117 edit event sequence diagram

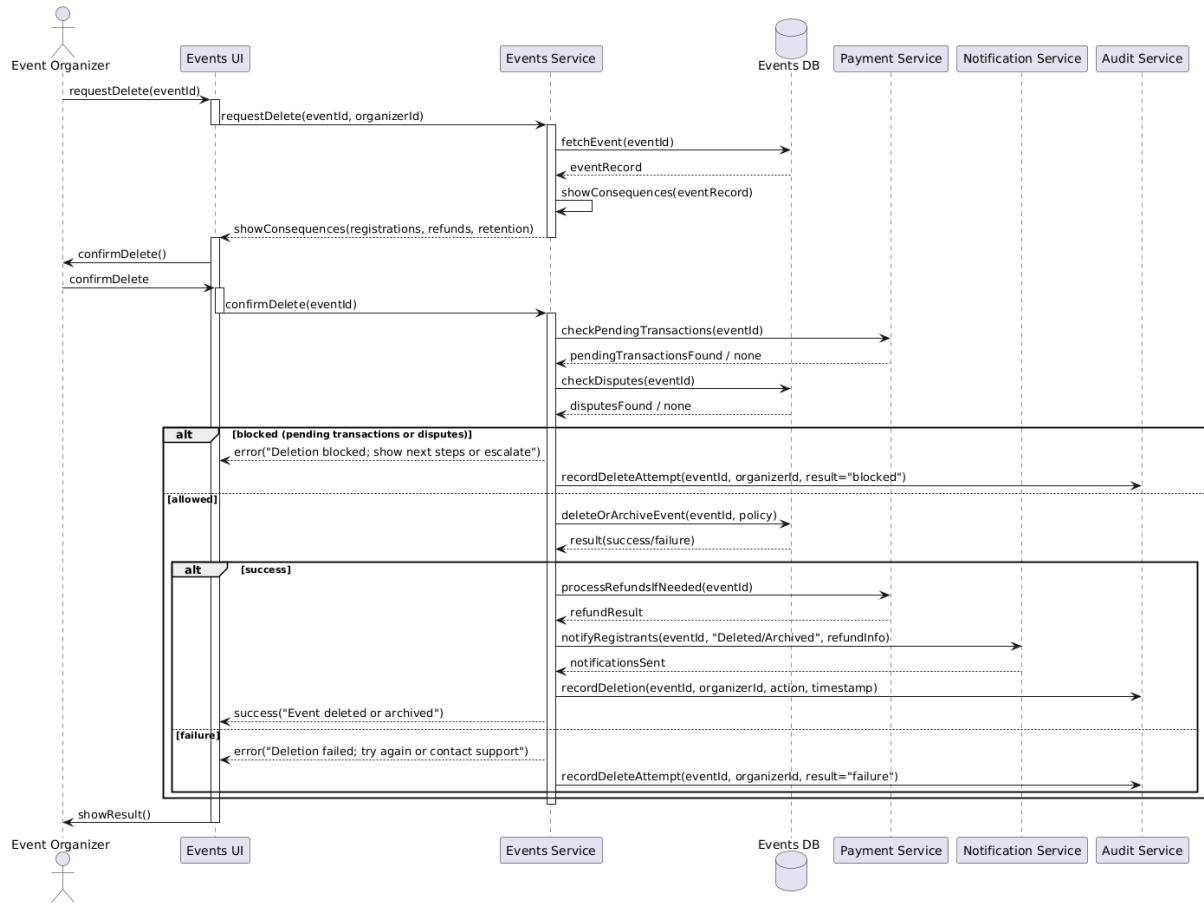


Figure 4.118 delete event sequence diagram

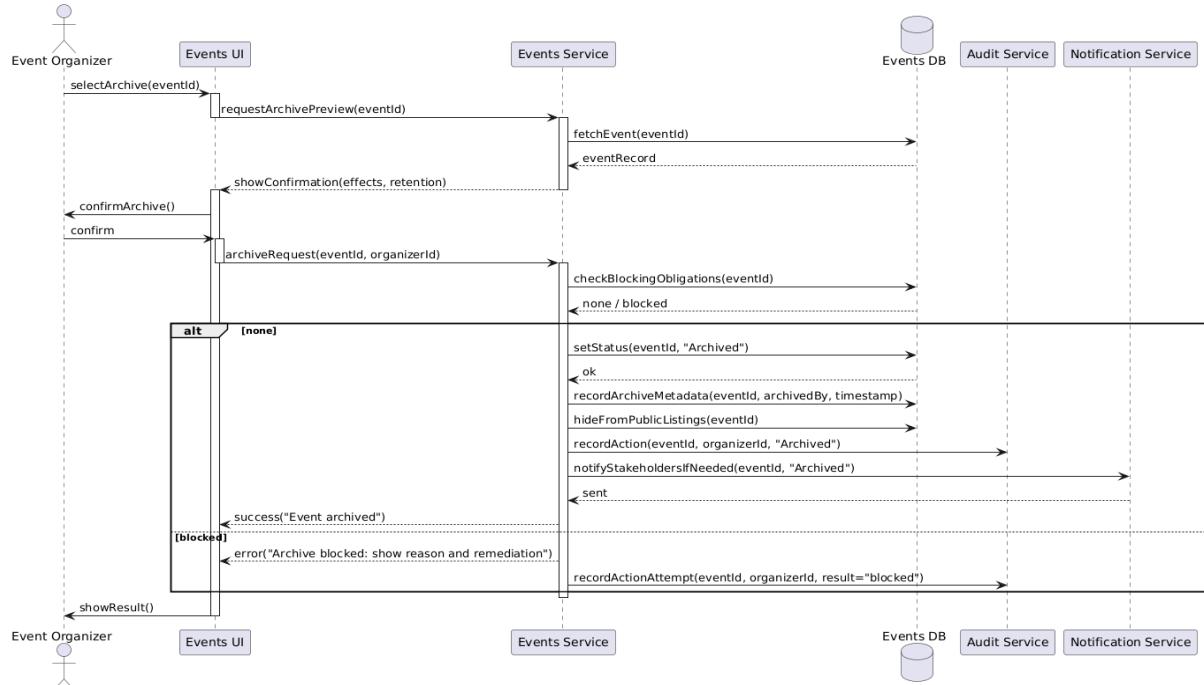


Figure 4.119 archive event sequence diagram

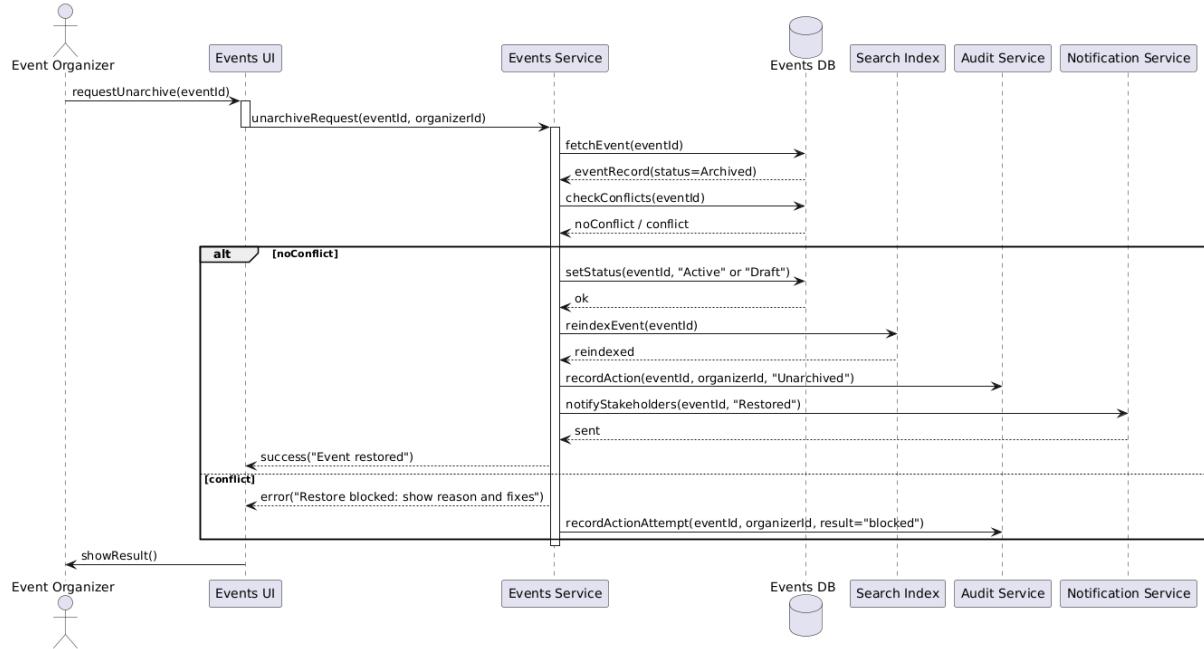


Figure 4.120 unarchive event sequence diagram

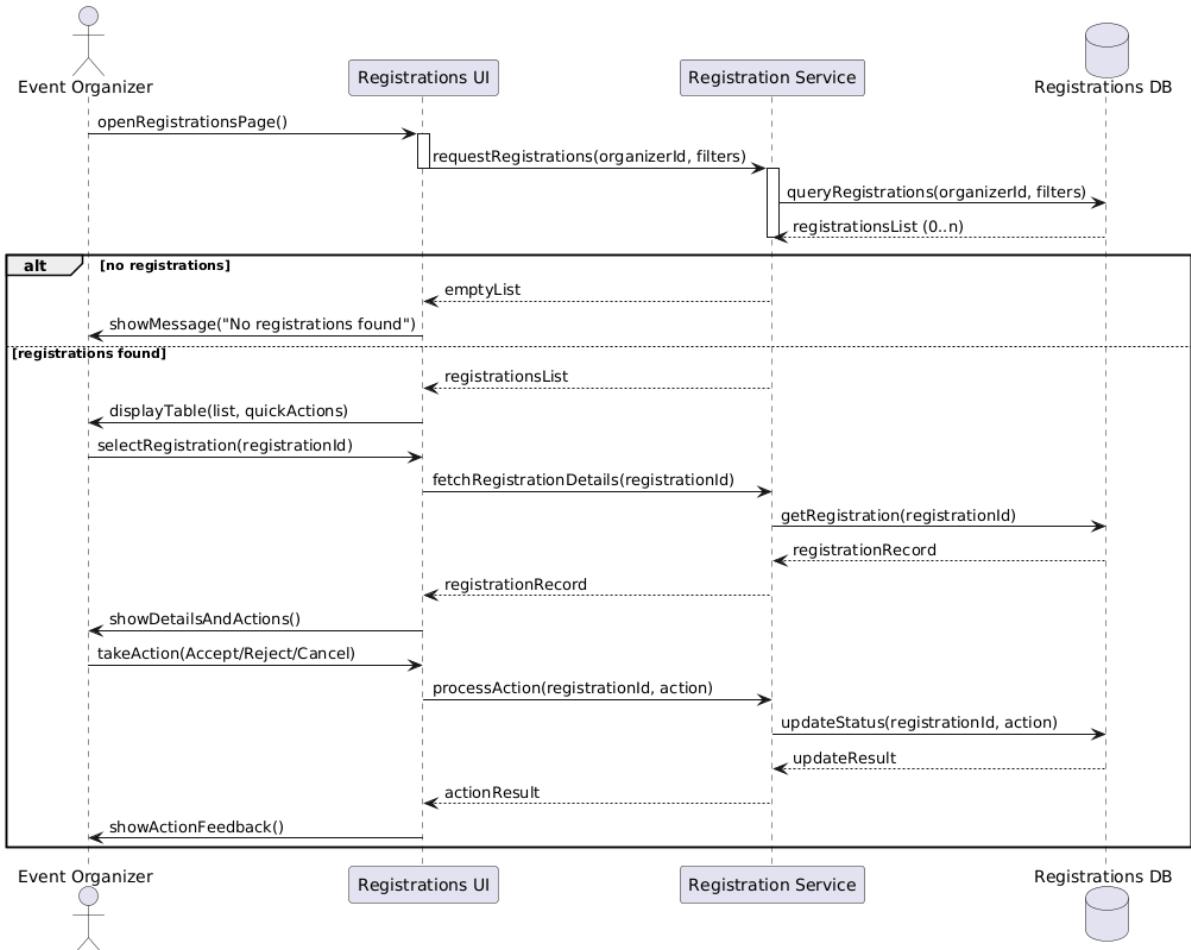


Figure 4.121 browse registrations sequence diagram

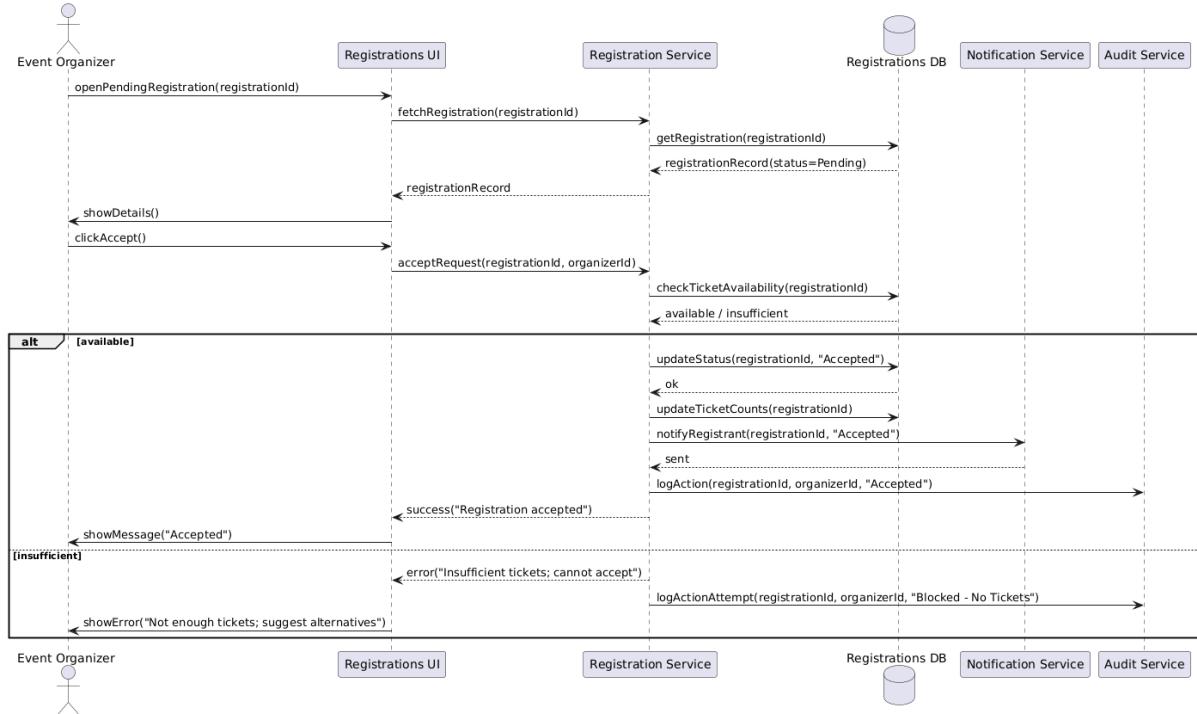


Figure 4.122 accept registration sequence diagram

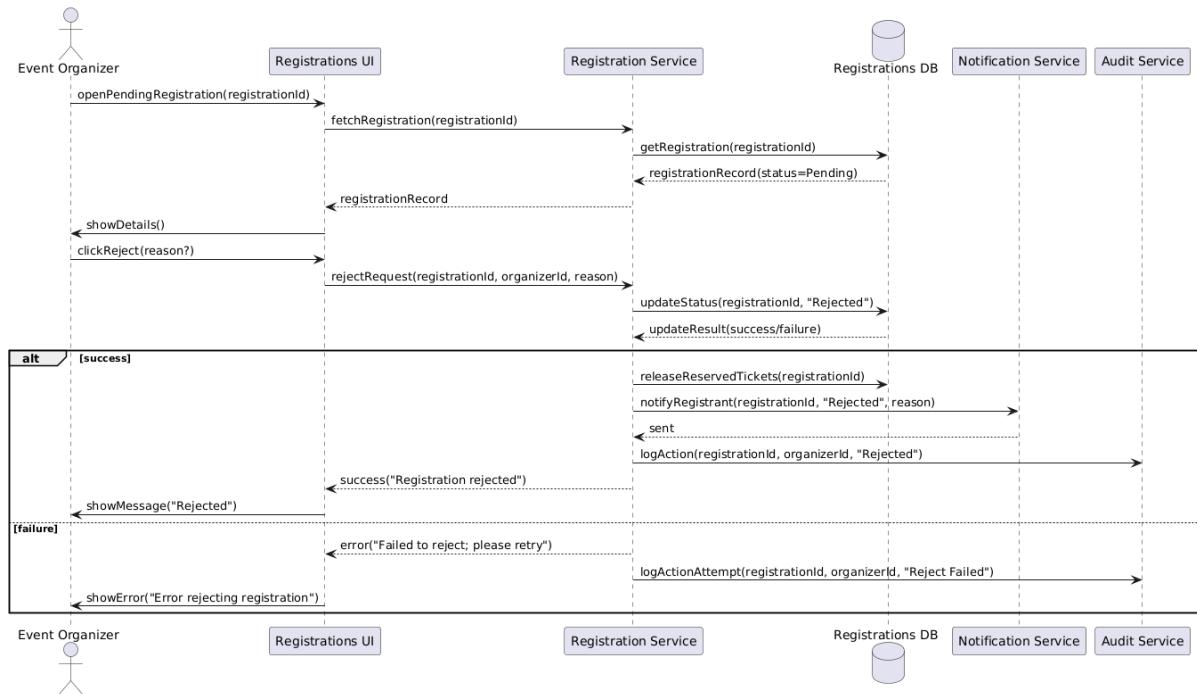


Figure 4.123 reject registration sequence diagram

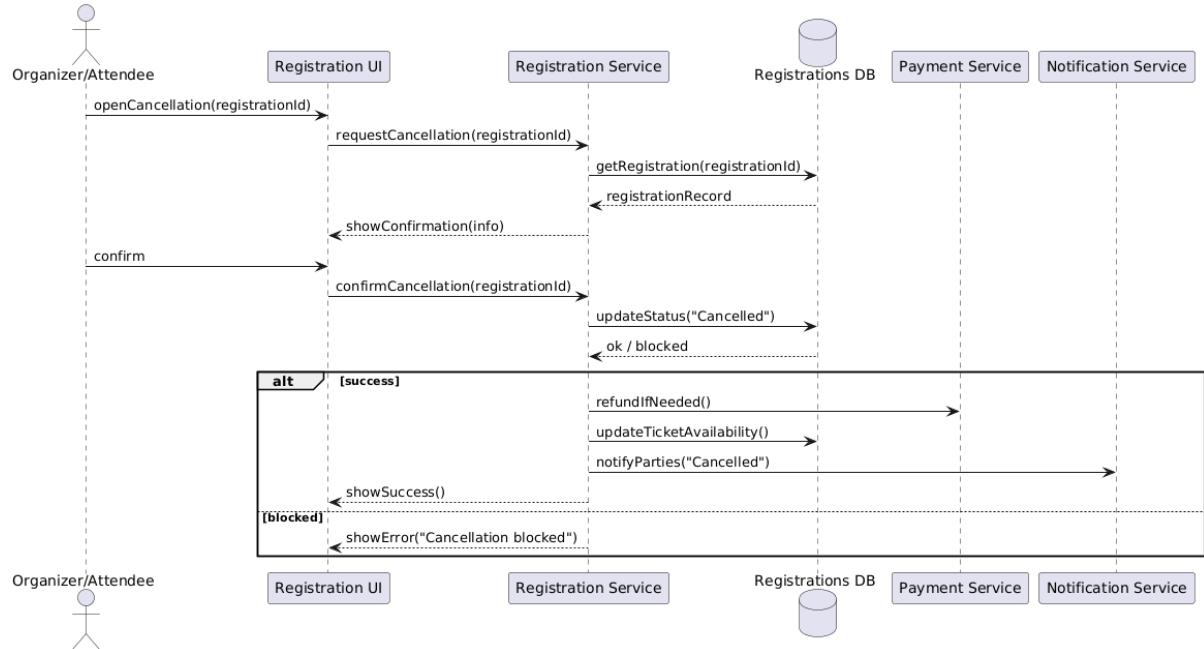


Figure 4.124 cancel registration sequence diagram

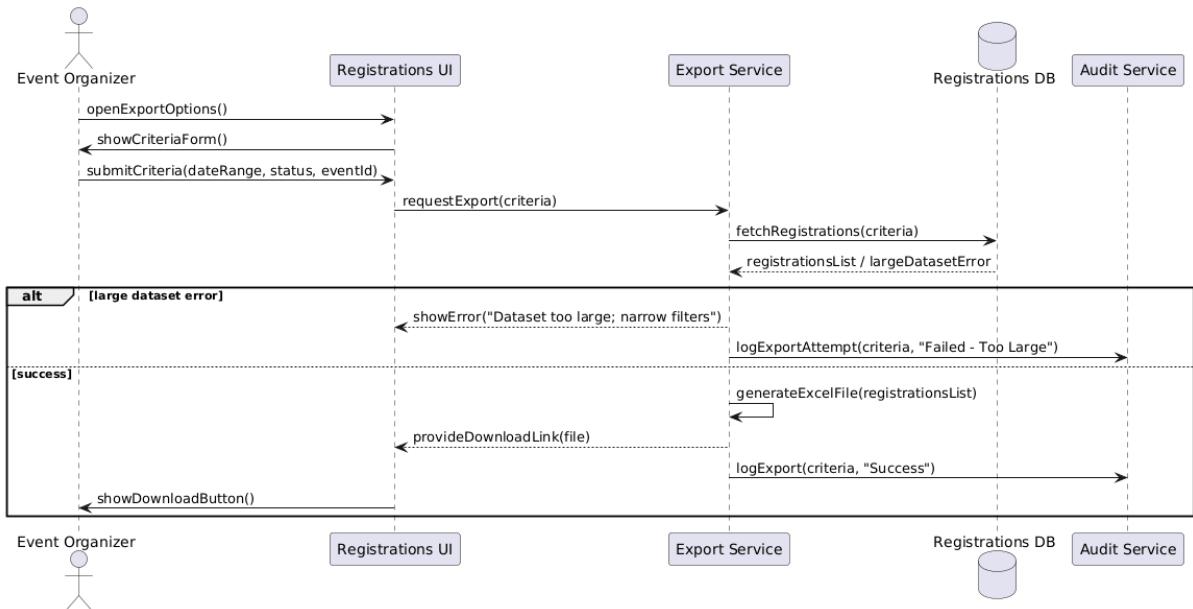


Figure 4.125 explore registration as excel file sequence diagram

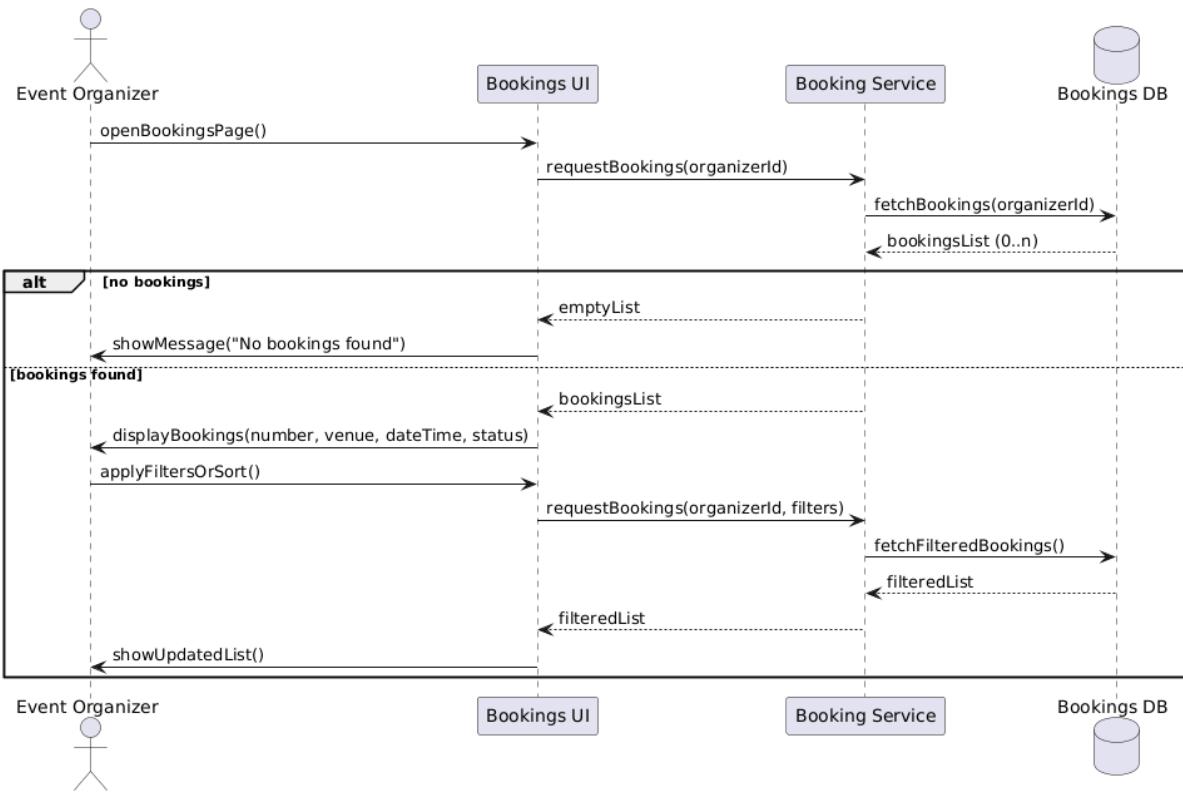


Figure 4.126 view booking (organizer) sequence diagram

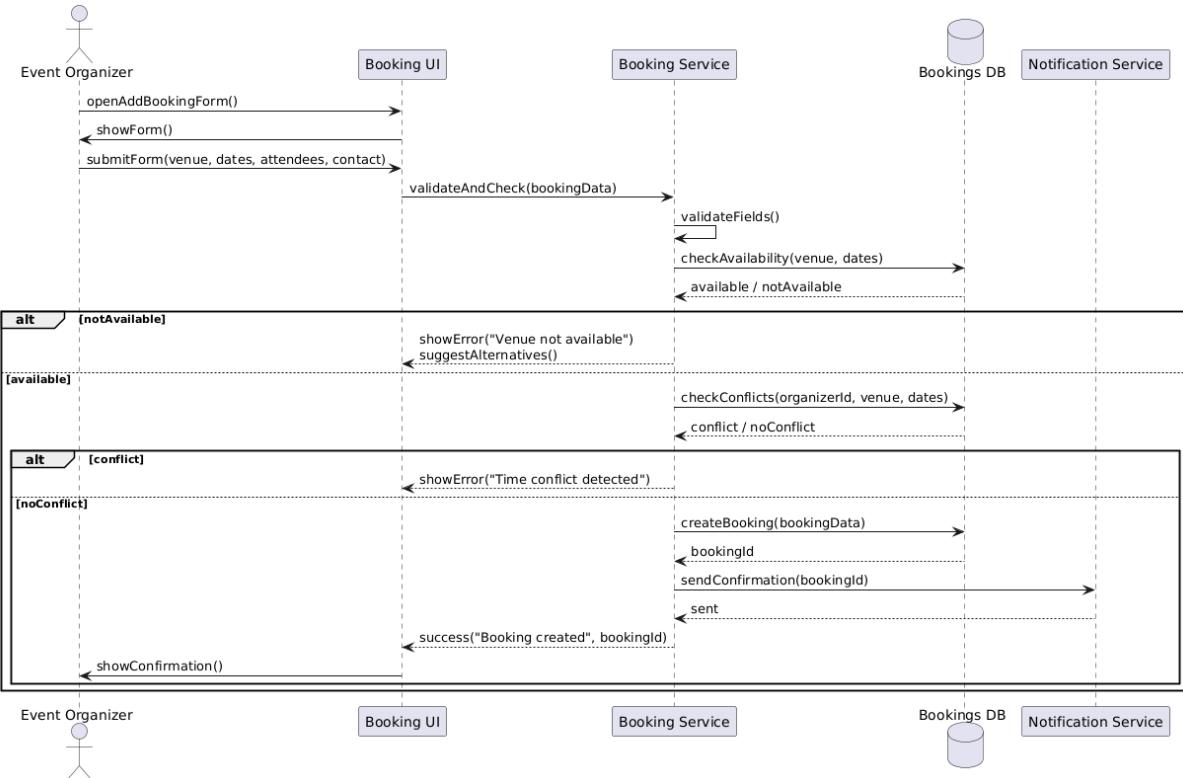


Figure 4.127 add booking (organizer) sequence diagram

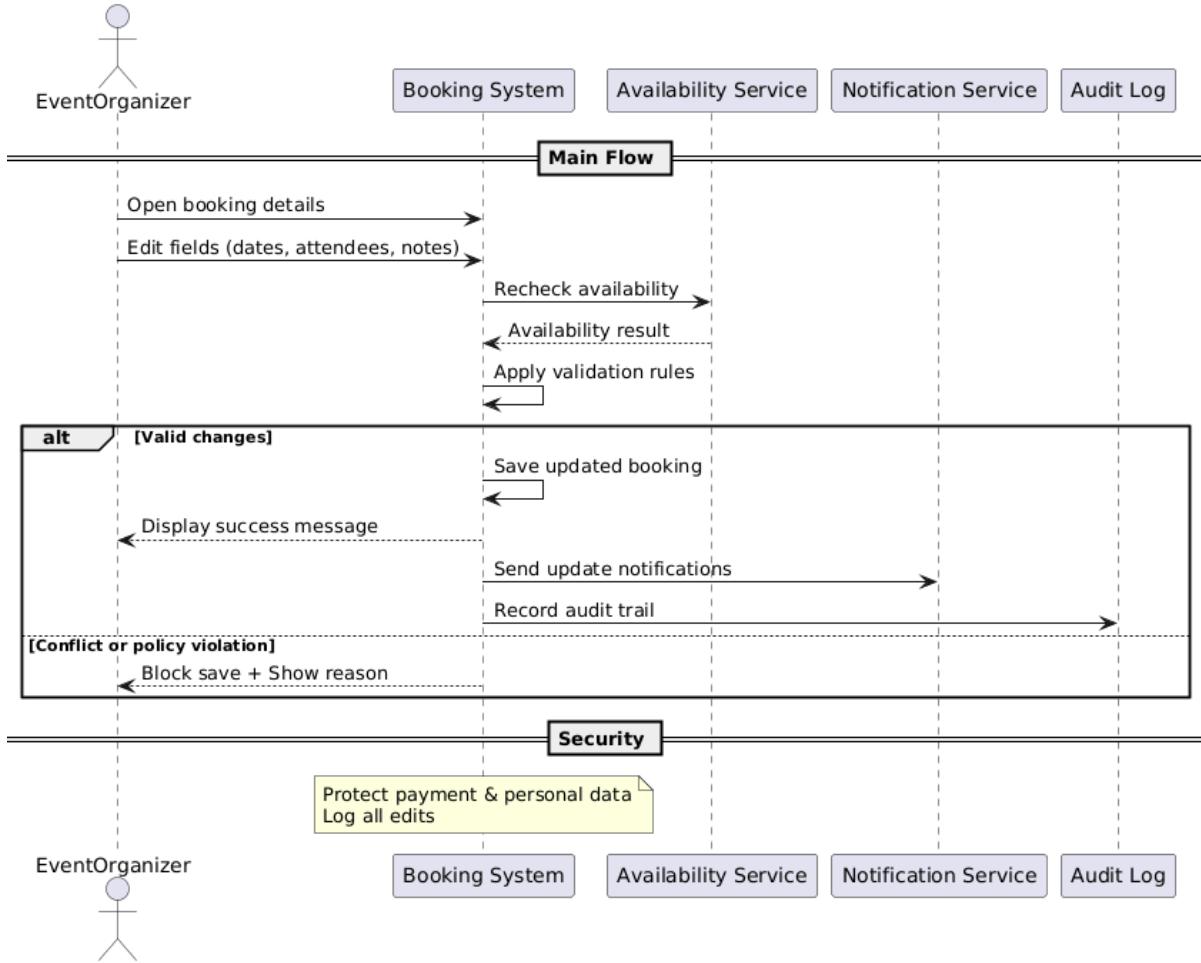


Figure 4.128 edit booking (organizer) sequence diagram

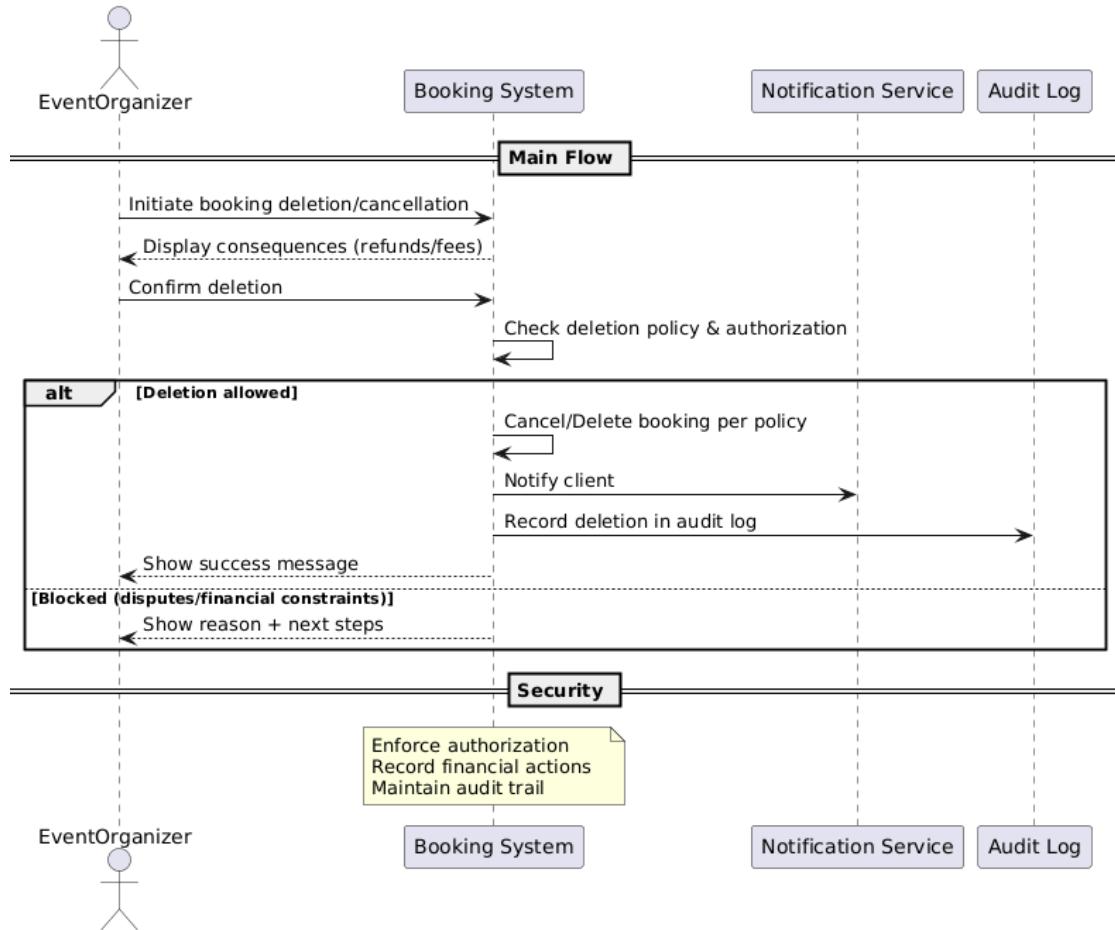


Figure 4.129 delete booking (organizer) sequence diagram

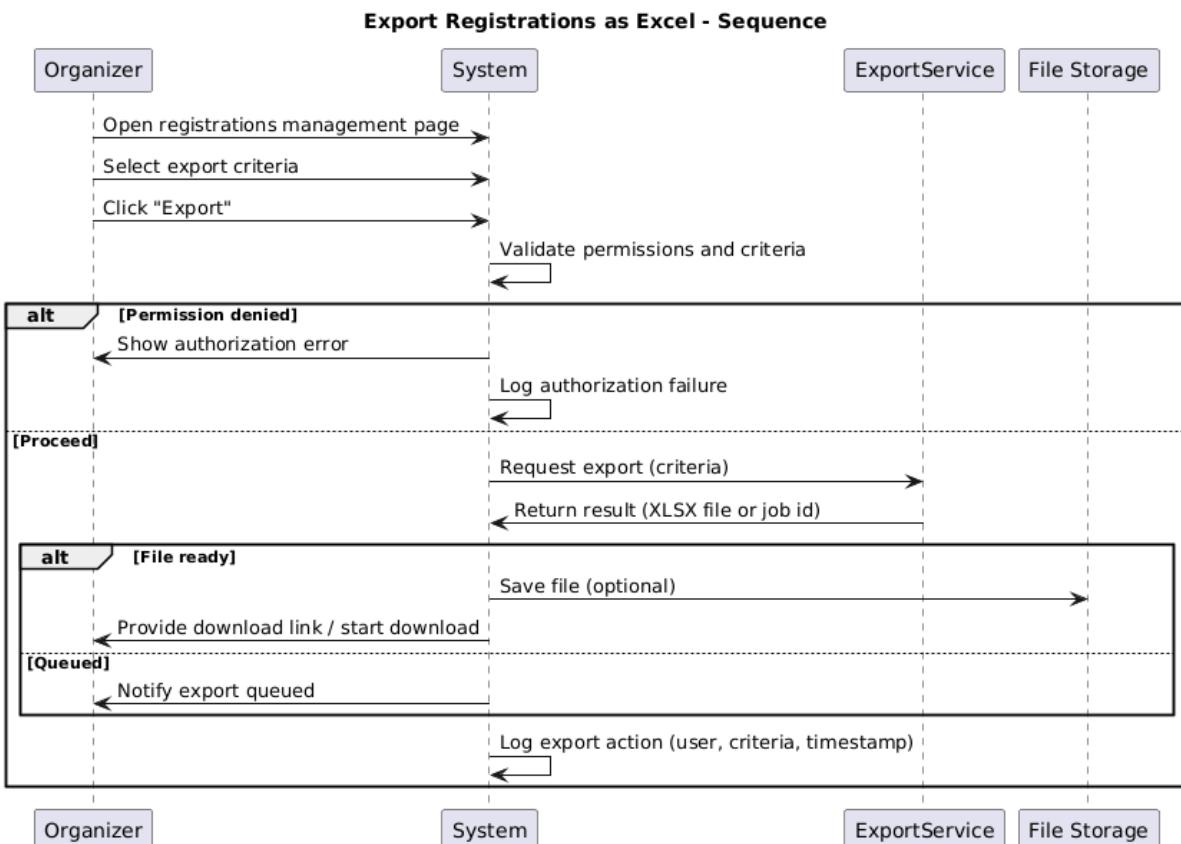


Figure 4.130 export registration as excel file sequence diagram

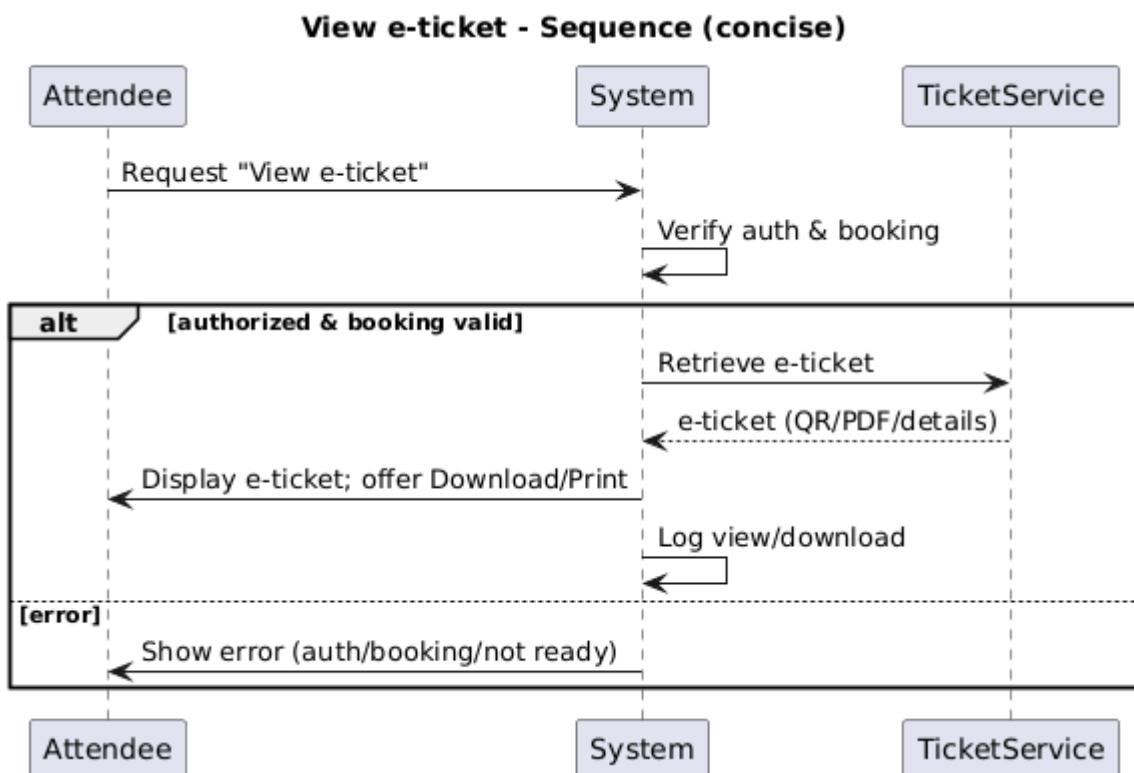


Figure 4.131 view e-ticket sequence diagram

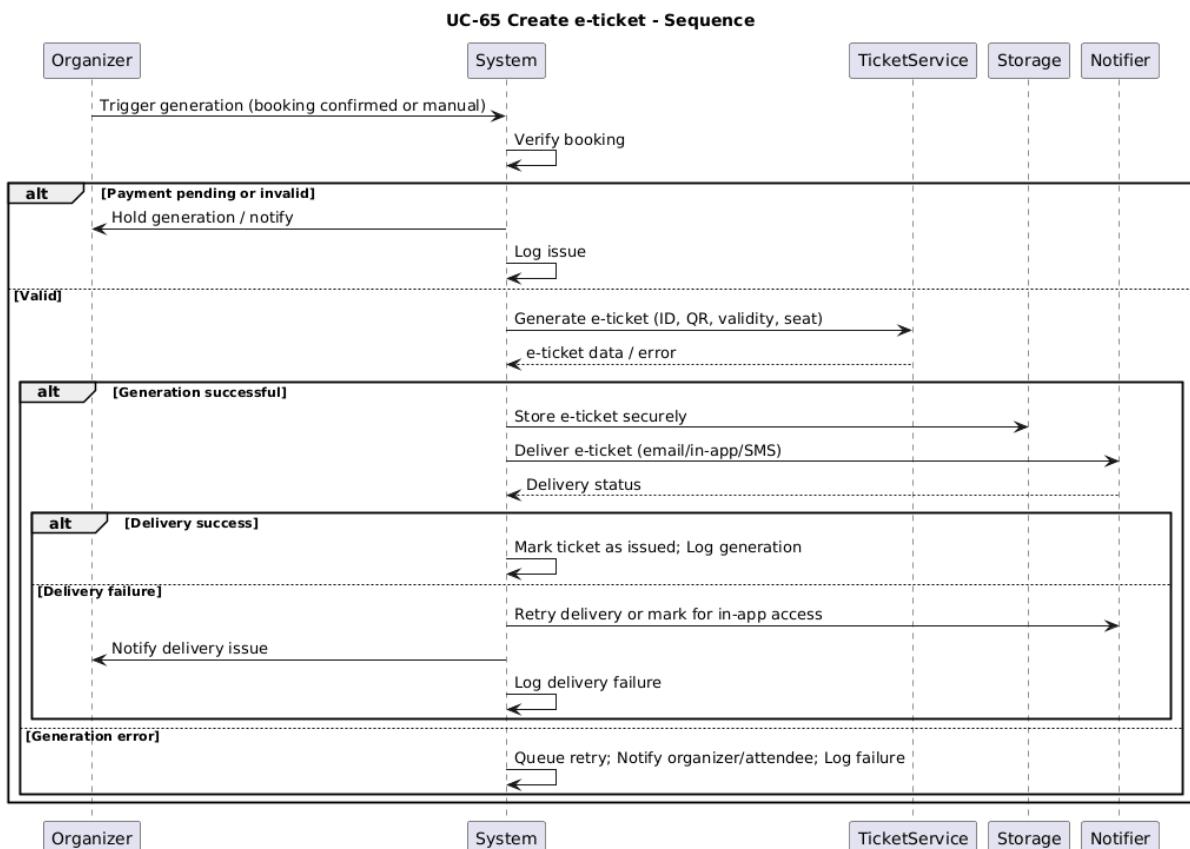


Figure 4.132 create e-ticket sequence diagram

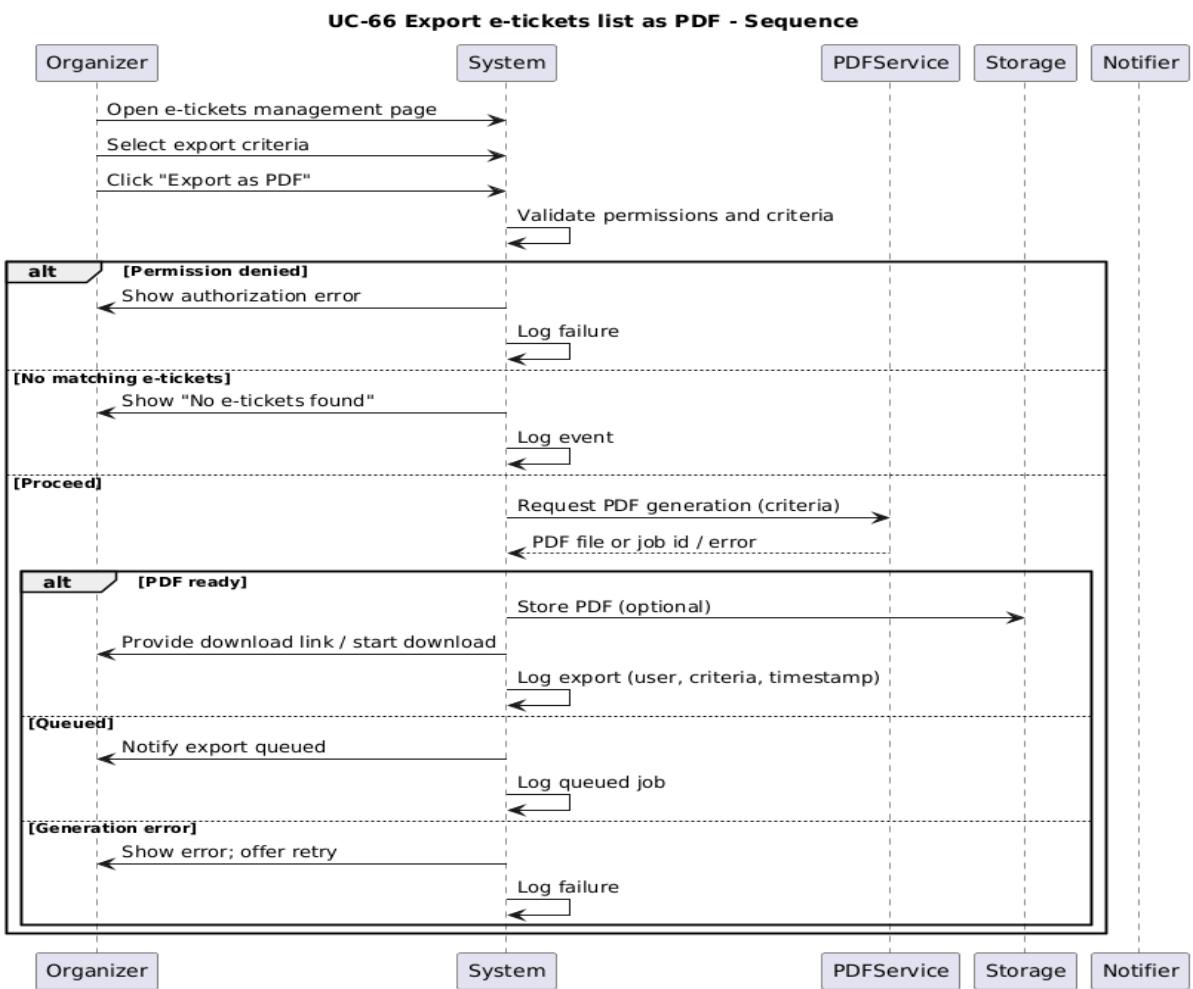


Figure 4.133 export e-ticket list as pdf sequence diagram

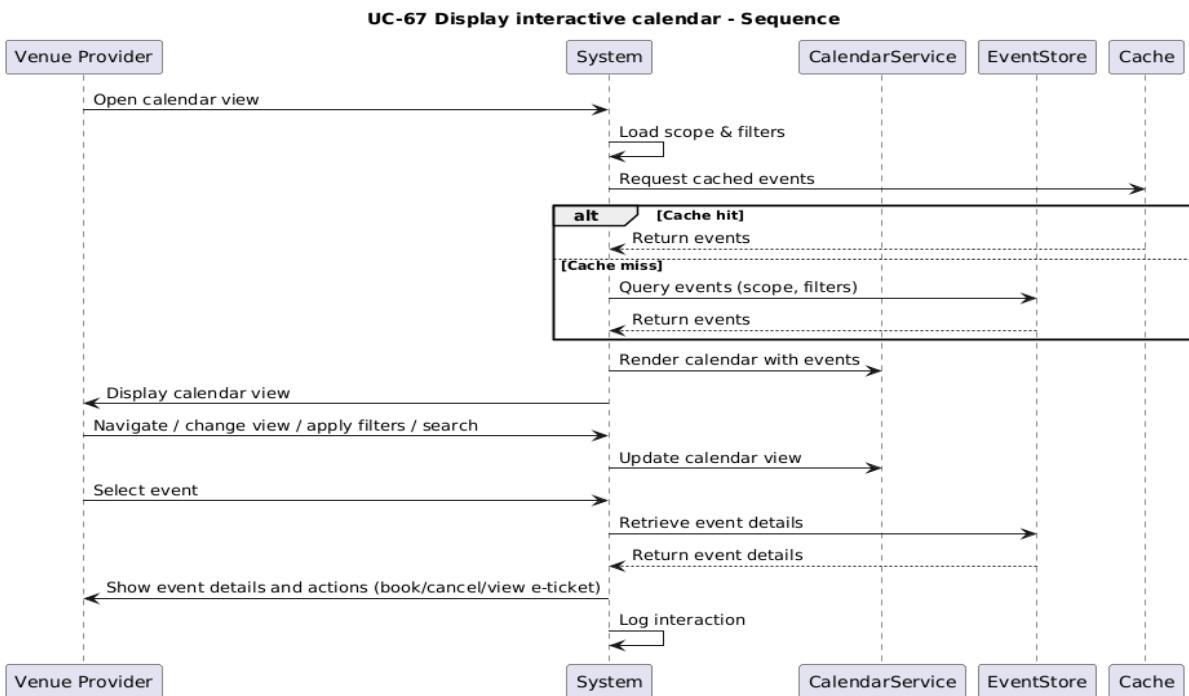


Figure 4.134 display interactive calendar sequence diagram

6.Initial test cases

Table 4.888 Initial test cases

Test Case ID	Test Case Name	Objective	Preconditions	Test Steps	Input Data	Expected Result
TC-UC01-01	Successful login with valid credentials	Verify login, session/token issuance, redirect, and login logging	Login page open; valid account exists	Enter valid email and password; click Login	email:user@example.com; password:CorrectPass123	Token/session issued; redirect to dashboard; login event logged
TC-UC01-02	Login with invalid password	Verify authentication rejection for wrong password	Login page open	Enter valid email and wrong password; click Login	email:user@example.com; password:WrongPass	Authentication error shown; no token issued; no session
TC-UC01-03	Login with inactive account	Prevent login for inactive account	Account exists but inactive	Enter inactive account credentials; click Login	email:inactive@example.com ; password:Pass12345	Account activation required message; no token issued
TC-UC02-01	Successful logout	Verify session termination, redirect to login, and logout logging	User is logged in	Click Logout	—	Session invalidated/revoked; redirect to login; logout event logged
TC-UC03-01	Successful registration	Verify account creation, confirmation email, activation and basic permissions	Registration page open	Fill valid fields; click Register	name:Sara; email:sara@example.com; password:StrongPass1	Account created; confirmation email sent; account activated or pending activation
TC-UC03-02	Register with duplicate email	Reject duplicate email registration	Registration page open; email already exists	Fill form with existing email; click Register	email:user@example.com; password:AnyPass	Error: email already in use; no account created
TC-UC04-01	View venues list default sort & pagination	Verify venues displayed as cards with default sort and pagination	User on venues page	Open venues page; navigate pages	—	Cards show name; location; capacity; price; thumbnail; rating; pagination present
TC-UC05-01	View venue details existing venue	Verify full venue detail page content	Valid venue exists	Open venue detail link	venue_id:12345	Image gallery; full description; pricing; capacity; availability; action buttons present
TC-UC05-02	View venue details not found	Handle venue-not-found gracefully	Invalid venue ID	Open detail link with invalid ID	venue_id:99999	"Venue not found" message or redirect; no server error
TC-UC06-01	View user bookings list	Show current and past bookings with details and filters	User logged in with bookings	Open bookings page; apply filters	—	List shows booking number; venue; date/time; status; sorting/filtering available
TC-UC07-01	Add booking successful	Create booking with availability check and confirmation	User logged in; venue available	Fill booking form; submit	user_id:10; venue_id:123; date:2026-03-10; time:18:00	Availability checked; booking created atomically; confirmation sent
TC-UC08-01	Edit booking with revalidation	Modify booking fields with availability revalidation and audit log	Existing editable booking	Open booking; change allowed fields; save	booking_id:200; new_time:19:00	Availability revalidated; changes saved; audit log recorded

TC-UC09-01	Delete booking with confirmation	Cancel/delete booking applying policies and logging reason	Existing booking	Click Delete; confirm	booking_id:200; reason:"Client request"	Booking cancelled/deleted; cancellation policy applied; action logged
TC-UC10-01	View events list with pagination	Display events list with basic info and pagination	Events page open	Open events page; navigate pages	—	Events show name; date; location; thumbnail; pagination supported
TC-UC11-01	View event details	Show event page with full details and ticket links	Event exists	Open event detail	event_id:200	Time/date; venue; description; ticket options; organizers; map if available
TC-UC12-01	Unified search venues partial match	Unified keyword search across venues with partial matches and pagination	Search page open	Enter partial keyword; search	query:"hall"	Results categorized under Venues; partial matches returned; pagination
TC-UC13-01	Unified search events exact match	Unified keyword search across events with pagination	Search page open	Enter event name; search	query:"Tech Summit 2026"	Results categorized under Events; matching results returned; pagination
TC-UC14-01	Filter venues by multiple criteria	Multi-criteria filtering with reset option	Venues page open	Apply location; price; capacity filters	location:Damascus; price_min:100; price_max:500	Filtered venues displayed; reset filters works
TC-UC15-01	Filter events by multiple criteria	Multi-criteria filtering with reset option	Events page open	Apply date; type filters	date:2026-05-01; type:Conference	Filtered events displayed; reset filters works
TC-UC16-01	View user registrations	Show user's event registrations with status and pagination	User logged in with registrations	Open registrations page	—	Registrations list with attendance status and event details; pagination
TC-UC17-01	Register for paid event	Register with availability check, payment, and ticket delivery	User logged in; tickets available	Select ticket; complete payment; confirm	event_id:200; ticket_type:VIP; payment:card	Availability checked; payment processed; ticket and confirmation delivered
TC-UC18-01	Edit registration within policy	Edit registration subject to policy and availability checks	Existing editable registration	Open registration; modify allowed fields; save	registration_id:555; new_ticket:Standard	Availability revalidated; changes saved; confirmation sent
TC-UC19-01	Cancel registration with confirmation	Cancel registration and update ticket availability	Existing registration	Click Cancel; confirm	registration_id:555	Registration cancelled; ticket availability updated; action logged
TC-UC20-01	View venue ratings list	Display reviews, average, count, and individual reviews with filters	Venue has reviews	Open reviews tab	venue_id:123	Average rating; total count; list of reviews; sorting/filtering available
TC-UC21-01	Submit venue rating eligible user	Submit rating and update averages; prevent duplicates	User eligible (past booking)	Submit rating and comment; save	venue_id:123; rating:5; comment:"Great"	Rating accepted; duplicates prevented; averages updated
TC-UC22-01	Edit venue rating within window	Allow reviewer to edit rating within	Rating exists and owned by user	Edit rating; save	rating_id:789; new_rating:4	Rating updated; average recalculated; edit recorded in audit trail

		allowed time and audit				
TC-UC23-01	Delete venue rating with confirmation	Delete rating, update average, and log deletion	Rating owned by user	Delete rating; confirm	rating_id:789	Rating removed; average updated; deletion logged
TC-UC24-01	View event ratings when none exist	Encourage first rating if none exist	Event has no ratings	Open ratings tab	event_id:200	Message encouraging first rating; average = N/A
TC-UC25-01	Add event rating after attendance	Submit rating subject to attendance policy and prevent duplicates	User attended event	Submit rating; save	event_id:200; rating:4	Rating accepted; duplicates prevented; optional organizer notification
TC-UC26-01	Edit event rating within timeframe	Edit rating within allowed timeframe and update averages	Rating exists and owned by user	Edit rating; save	rating_id:900; new_rating:5	Rating updated; average recalculated; change recorded
TC-UC27-01	Delete event rating and update avg	Delete rating and log action	Rating exists	Delete rating; confirm	rating_id:900	Rating removed; average updated; deletion logged
TC-UC28-01	View organizers list	List organizers with summary info and links	Organizers exist	Open organizers page	—	List shows name; short bio; past events count; overall rating; details link
TC-UC29-01	View providers list	List providers with venue counts and contact links	Providers exist	Open providers page	—	List shows provider name; number of venues; rating; contact/view links
TC-UC30-01	Filter organizers by criteria	Filter by city; rating; experience with reset	Organizers page open	Apply city and rating filters	city:Aleppo; rating_min:4	Filtered organizers displayed; reset works
TC-UC31-01	Filter providers by criteria	Filter by city; rating; experience with reset	Providers page open	Apply experience filter	experience_min:3	Filtered providers displayed; reset works
TC-UC32-01	Upvote content once per user	Record upvote; prevent duplicates; log voter and timestamp	User logged in	Click Upvote on content	content_id:321	UI count increments immediately; duplicate prevented; vote logged
TC-UC33-01	Downvote content once per user	Record downvote; prevent duplicates; log voter and timestamp	User logged in	Click Downvote on content	content_id:321	UI count updates immediately; duplicate prevented; vote logged
TC-UC34-01	View recent activities timeline	Show user's activity timeline with filters and links	User has recent activities	Open activities page; apply filter	filter:bookings	Timeline shows activities with timestamps and links; filters work
TC-UC35-01	Display interactive map with pins	Show event/venue pins with info windows and filtering	Location data available	Open map; zoom; click pin	—	Pins displayed; info windows show details; clustering and filters supported
TC-UC36-01	Browse provider venues list	Provider dashboard lists venues with search and filters	Provider account with venues	Login as provider; open dashboard	provider_id:77	Provider sees venue cards with search; filters; edit links
TC-UC37-01	View archived	List archived venues with metadata and	Provider has archived venues	Open archived	venue_id:400	Archive date/reason shown; restore works; action logged

	venues and restore	restore actions		tab; select Restore		
TC-UC38-01	Provider adds new venue successfully	Create venue with required fields and media handling	Add venue page open	Fill required fields; upload images; save	name:"Hall A"; location:Homs; images:3	Validation passes; venue saved; media processed; appears in provider dashboard
TC-UC39-01	Edit venue with schedule validation	Edit venue data with conflict checks and audit trail	Venue exists and editable	Edit fields; save	venue_id:123; new_schedule:...	Conflicts validated; changes saved; public listing updated; audit recorded
TC-UC40-01	Prevent deletion if active bookings exist	Block deletion when constraints exist and log attempt	Venue has active bookings	Attempt to delete venue	venue_id:123	Deletion blocked with explanatory message; attempt logged
TC-UC41-01	Archive venue hides from public	Archive venue and record metadata	Venue exists	Archive venue; confirm	venue_id:123	Status set to Archived; hidden from public; archive metadata logged
TC-UC42-01	Unarchive venue restores listing	Restore archived venue and reindex	Venue is archived	Unarchive venue; confirm	venue_id:123	Venue restored; reindexed; restoration logged
TC-UC43-01	Provider views bookings with actions	Provider sees bookings with management actions	Provider has bookings	Open provider bookings page; apply filters	provider_id:77	Bookings listed with accept/reject/detail actions
TC-UC44-01	Provider accepts pending booking	Accept booking after availability recheck and notify client	Booking status Pending	Click Accept; confirm	booking_id:888	Availability rechecked; status set to Accepted; resources reserved; client notified
TC-UC45-01	Provider rejects booking with reason	Reject booking, release resources, notify client and log	Booking status Pending	Click Reject; enter reason; confirm	booking_id:889; reason:"Unavailable"	Status set to Rejected; resources released; client notified; reason logged
TC-UC46-01	Provider cancels accepted booking	Cancel accepted booking, handle refunds, notify client	Booking Accepted/Confirmed	Click Cancel; confirm; process refunds if applicable	booking_id:890	Status set to Cancelled; refunds processed if applicable; client notified; action logged
TC-UC47-01	Browse events dashboard (organizer)	Organizer sees events with statuses and management actions	Organizer account with events	Login as organizer; open events dashboard	organizer_id:55	Events listed with Active/Draft/Archived statuses and management actions
TC-UC48-01	View archived events and restore	List archived events with metadata and restore options	Organizer has archived events	Open archived tab; select Restore	event_id:300	Archive metadata shown; restore works; action logged
TC-UC49-01	Create event with venue availability check	Create event with venue checks and draft/publish options	Create event page open; venue available	Fill event details; check availability; save or publish	title:"Dev Meetup"; venue_id:123; dates:2026-06-10	Availability confirmed; event created; can save as draft or publish; organizer notified
TC-UC50-01	Edit event and notify attendees if needed	Edit event and validate impacts on tickets/schedules	Published event with registrants	Edit dates/sessions; save	event_id:300; new_date:2026-06-12	System validates impacts; changes saved; affected attendees notified
TC-UC51-01	Prevent deletion with unresolved refunds	Block deletion if refunds/disputes exist and	Event has pending refunds/disputes	Attempt to delete event	event_id:300	Deletion blocked; reason logged; organizer notified

		notify organizer				
TC-UC52-01	Archive event hides from public	Archive event and record metadata	Event exists	Archive event; confirm	event_id:300	Status set to Archived; hidden from public; archive metadata logged
TC-UC53-01	Unarchive event restores listing	Restore archived event and reindex	Event is archived	Unarchive event; confirm	event_id:300	Event restored; reindexed; restoration logged
TC-UC54-01	Organizer views registrations with actions	View registrations with filters and quick actions	Organizer has registrations	Open registrations page; apply filters	organizer_id:55	Registrations listed with accept/reject actions and stats
TC-UC55-01	Organizer accepts pending registration	Accept registration, update ticket counts, notify registrant	Registration Pending	Click Accept; confirm	registration_id:700	Status set to Accepted; ticket counts updated; registrant notified
TC-UC56-01	Organizer rejects registration with reason	Reject registration, release tickets, notify and log	Registration Pending	Click Reject; enter reason; confirm	registration_id:701; reason:"Incomplete"	Status set to Rejected; tickets released; registrant notified; reason logged
TC-UC57-01	Cancel registration and handle refund	Cancel registration with refund handling per policy	Registration Accepted/Confirmed	Click Cancel; confirm; apply refund policy	registration_id:702	Status set to Cancelled; refunds processed if applicable; availability updated; action logged
TC-UC58-01	Export registrations as XLSX	Generate XLSX export for selected criteria and log export	Registrations exist in range	Select date range; click Export	date_from:2026-01-01; date_to:2026-01-31	XLSX file generated and available for download; export logged
TC-UC59-01	Organizer views booking details	Organizer views booking details with management actions	Organizer linked to booking	Open booking detail	booking_id:888	Booking details displayed with management actions
TC-UC60-01	Organizer creates booking on behalf of client	Create booking for client with availability validation and confirmation	Organizer logged in; venue available	Fill booking form for client; submit	client_name:"Ali"; venue_id:123; date:2026-04-05	Availability validated; booking created; payment processed if needed; confirmation sent to client
TC-UC61-01	Organizer edits booking with conflict check	Edit booking and validate conflicts; notify client and audit	Booking exists	Edit allowed fields; save	booking_id:888; new_time:19:00	Conflicts checked; changes saved; client notified; audit recorded
TC-UC62-01	Organizer deletes booking with refund handling	Delete/cancel booking with refund processing and notifications	Booking exists and cancellable	Click Delete; confirm; process refunds if applicable	booking_id:888	Booking deleted or cancelled; refunds processed if applicable; client notified; action logged

7. Requirements Traceability Matrix:

Table 4.888 Requirements Traceability Matrix

Requirement ID	Analysis	Design	Application Interface	Code	Test Case ID
REQ-A01	Login				initial test cases
REQ-A02	Logout				initial test cases
REQ-A03	Register (User Registration)				initial test cases
REQ-A04	View Venues				initial test cases
REQ-A05	View Venue Details				initial test cases
REQ-A06	View Bookings				initial test cases
REQ-A07	Add Booking				initial test cases
REQ-A08	Edit Booking				initial test cases
REQ-A09	Delete Booking				initial test cases
REQ-A10	View Events				initial test cases
REQ-A11	View Event Details				initial test cases
REQ-A12	Unified Search for Venues				initial test cases
REQ-A13	Unified Search for Events				initial test cases
REQ-A14	Filter Venues by Specific Criteria				initial test cases
REQ-A15	Filter Events by Specific Criteria				initial test cases
REQ-A16	View Registrations				initial test cases
REQ-A17	Add Registration				initial test cases
REQ-A18	Edit Registration				initial test cases
REQ-A19	Delete Registration				initial test cases
REQ-A20	View Venue Ratings				initial test cases

REQ-A21	Add Venue Rating				initial test cases
REQ-A22	Edit Venue Rating				initial test cases
REQ-A23	Delete Venue Rating				initial test cases
REQ-A24	View Event Ratings				initial test cases
REQ-A25	Add Event Rating				initial test cases
REQ-A26	Edit Event Rating				initial test cases
REQ-A27	Delete Event Rating				initial test cases
REQ-A28	View Organizers				initial test cases
REQ-A29	View Providers				initial test cases
REQ-A30	Filter Organizers by Specific Criteria				initial test cases
REQ-A31	Filter Providers by Specific Criteria				initial test cases
REQ-A32	Upvote				initial test cases
REQ-A33	Downvote				initial test cases
REQ-A34	View Recent Activities				initial test cases
REQ-A35	Display Map of Event and Venue Locations				initial test cases
REQ-A36	Browse Venues (Provider)				initial test cases
REQ-A37	Browse Archived Venues				initial test cases
REQ-A38	Add Venue				initial test cases
REQ-A39	Edit Venue				initial test cases
REQ-A40	Delete Venue				initial test cases
REQ-A41	Archive Venue				initial test cases

REQ-A42	Unarchive Venue				initial test cases
REQ-A43	Browse Bookings (Provider)				initial test cases
REQ-A44	Accept Booking (Provider)				initial test cases
REQ-A45	Reject Booking (Provider)				initial test cases
REQ-A46	Cancel Booking (Provider)				initial test cases
REQ-A47	Browse Events (Organizer)				initial test cases
REQ-A48	Browse Archived Events				initial test cases
REQ-A49	Create Event				initial test cases
REQ-A50	Edit Event				initial test cases
REQ-A51	Delete Event				initial test cases
REQ-A52	Archive Event				initial test cases
REQ-A53	Unarchive Event				initial test cases
REQ-A54	Browse Registrations (Organizer)				initial test cases
REQ-A55	Accept Registration				initial test cases
REQ-A56	Reject Registration				initial test cases
REQ-A57	Cancel Registration				initial test cases
REQ-A58	Export Registrations as Excel file				initial test cases
REQ-A59	View Booking (Organizer)				initial test cases
REQ-A60	Add Booking (Organizer)				initial test cases
REQ-A61	Edit Booking (Organizer)				initial test cases

REQ-A62	Delete Booking (Organizer)				initial test cases
REQ-A63	Export Registrations as Excel file				initial test cases
REQ-A64	View e-ticket				initial test cases
REQ-A65	Create e-ticket				initial test cases
REQ-A66	Export e-tickets list as PDF file				initial test cases
REQ-A67	Display an interactive calendar				initial test cases

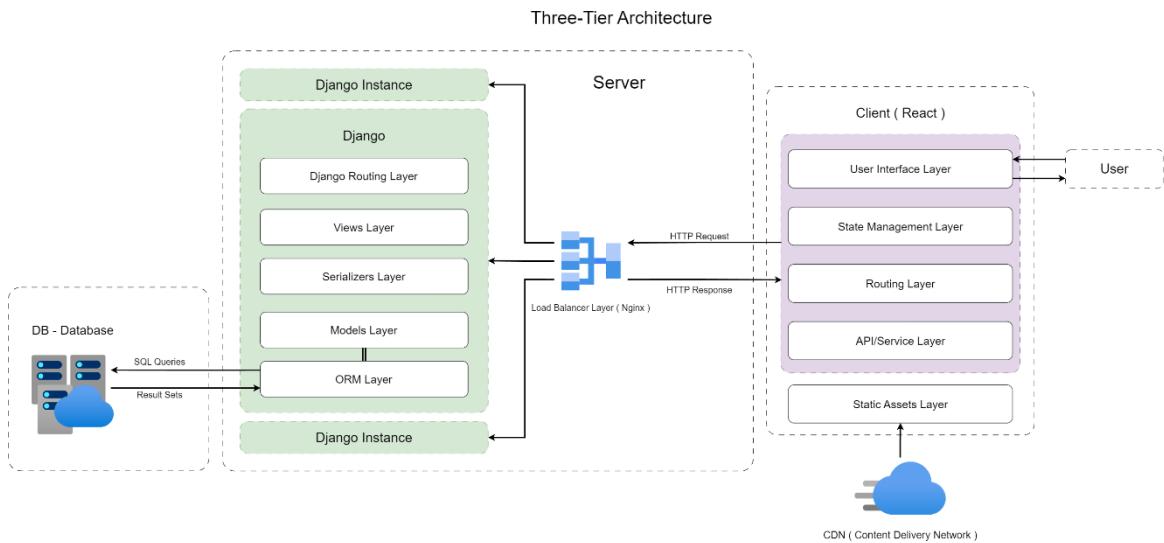
8.System Architecture

The system follows a **client-server** model where the frontend is a **React SPA** that communicates with the backend via a **Django REST API** using JSON over HTTPS. The design separates the UI from backend logic and external services to simplify maintenance and iterative development.

Main Components

- **Frontend React:** Reusable pages and components, authentication managed via `AuthContext`, and centralized request handling using `Axios` interceptors.
- **Backend Django and DRF:** Data models, serializers, `ViewSets/endpoints`, and authentication using `SimpleJWT`. The payment-related webhook handler has been removed from the current design.
- **Database:** `SQLite` is used locally and also as the simplified database for the experimental/early production MVP, noting its performance and concurrency limitations.
- **Media Storage:** Local storage during development; for deployment, S3 or an equivalent object store can be used for images and media.
- **External Services:** Maps (Leaflet/OpenStreetMap or Google Maps) and an email delivery service (SMTP or a provider). Payment gateway and payment services have been removed from the external services list.
- **Background Tasks (optional):** `Celery + Redis` for long-running or asynchronous jobs such as notifications or report generation; can be enabled later as needed.

Simplified Interaction Diagram



Security Considerations

- **Secure transport:** Enforce HTTPS for all communications.
- **Authentication and authorization:** Use JWT (SimpleJWT) for authentication; enforce permissions at the ViewSet level and perform ownership checks before any modification or deletion.
- **Media protection:** Control access to stored media using signed URLs or storage-level access rules to prevent unauthorized access.
- **Additional practices:** Password hashing, environment secret management via environment variables or a secrets service, and audit logging.

Implementation Notes

- **SQLite in production:** Using SQLite for an MVP or limited pilot is possible but carries performance and concurrency constraints; plan capacity accordingly.
- **Reintroducing payments later:** If a payment gateway is added later, reintroduce webhook handling in the backend and design the Booking flow to accommodate payment states (e.g., pending → confirmed) without major schema changes.
- **Documentation updates:** Update the Gantt chart, acceptance criteria, and Sprint Backlogs to remove payment tasks and related integration steps, and remove any payment-related tests.

9.Conclusion

Chapter Three represented a pivotal transition from abstract requirements into a concrete execution framework. We began by **evaluating feasibility**, ensuring that the proposed

solution is technically achievable, economically viable, and aligned with the academic and market context. This evaluation provided confidence that the project can progress without encountering insurmountable risks.

We then established a **comprehensive project timeline** supported by a Gantt chart and sprint planning. This timeline not only defines the duration of each sprint but also highlights dependencies, milestones, and deliverables, giving the team a clear roadmap for execution. Alongside the timeline, we documented **Software Requirements Specifications (SRS)**, which formalize functional and non-functional requirements, serving as a contractual baseline between stakeholders and the development team.

To bridge requirements with implementation, we modeled **use cases and requirement diagrams**, clarifying how different user groups—attendees, venue providers, and event organizers—interact with the system. These models ensure that user needs are directly translated into system behaviors and guide the prioritization of features across sprints.

Recognizing the importance of quality assurance, we defined a **testing strategy** that includes unit tests, integration tests, and usability evaluations. A **traceability matrix** was also developed to link each requirement to its corresponding test case, guaranteeing full coverage and reducing the risk of overlooked functionality.

Finally, we designed a **system architecture** that is both practical and extensible. The architecture separates frontend and backend responsibilities, leverages Django REST Framework and React SPA for modularity, and integrates external services such as maps and email delivery. Security considerations—such as HTTPS, JWT authentication, and controlled access to media—were embedded into the design to ensure robustness from the outset.

In summary, Chapter Three transformed theoretical requirements into a **clear, actionable plan**: feasibility validated, timeline defined, specifications formalized, models created, testing strategy established, traceability ensured, and architecture designed. This foundation prepares the project for the next stages. The upcoming chapters will move into **detailed design and practical implementation**, where the system will evolve from documented plans into a working prototype, tested iteratively and refined according to the roadmap.

Chapter Five

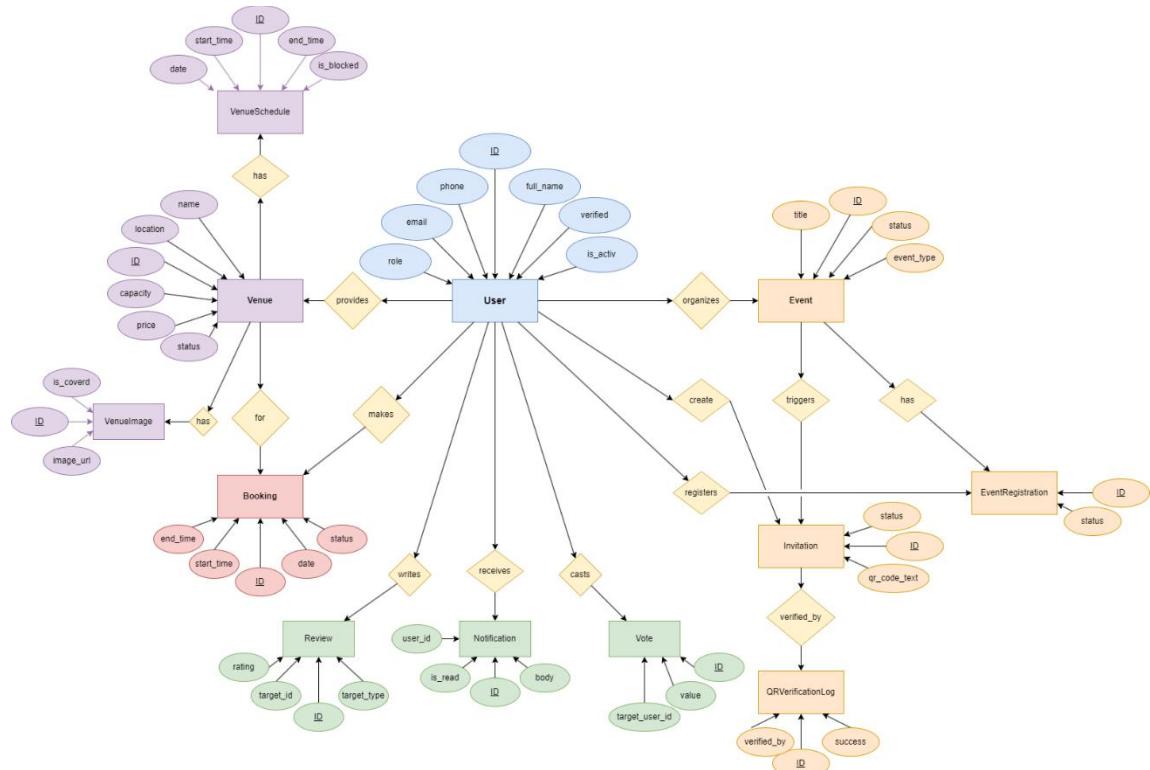
System Design Study

1.introduction

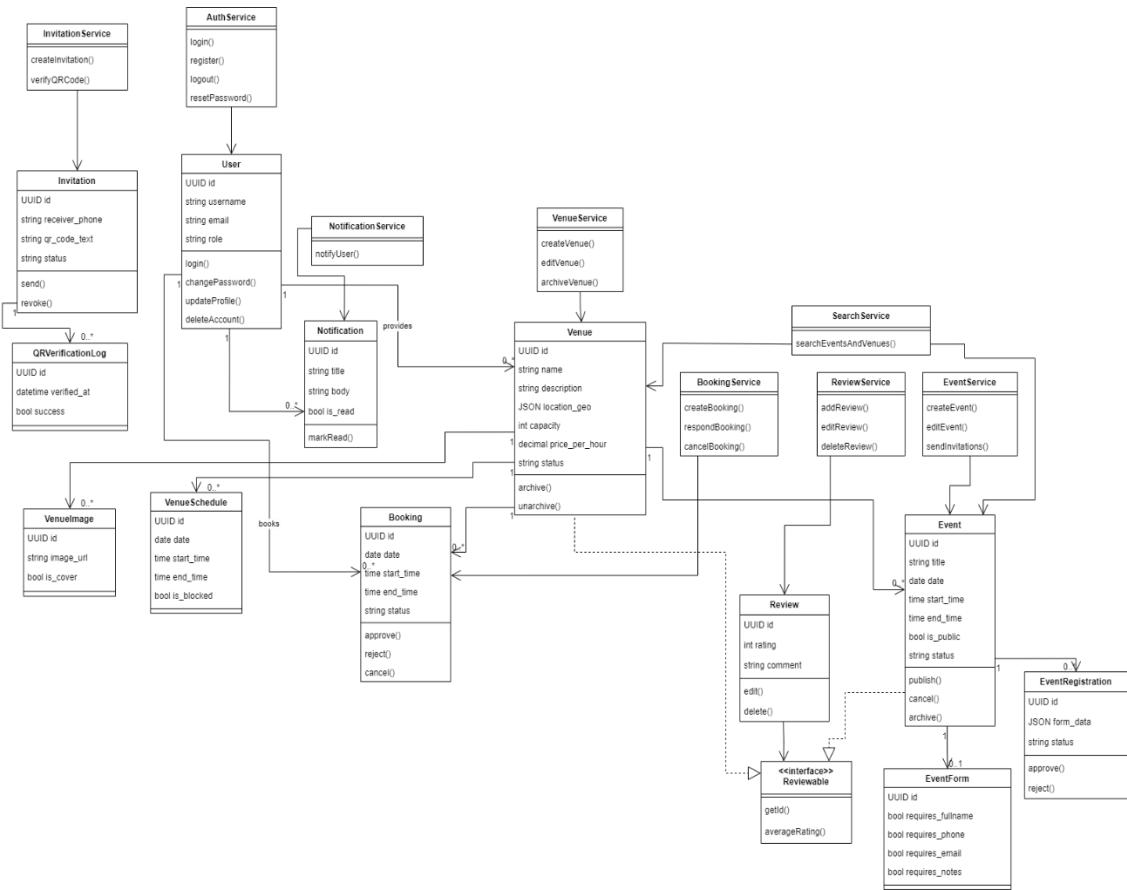
This chapter presents the **design study of the proposed system**, moving from the execution plan into a detailed technical blueprint. The aim is to translate the documented requirements into clear design models that guide implementation. We begin with logical design models such as the **database schema, class diagram, and component diagram**, which illustrate the structure of data and the relationships between system components. We then proceed to physical design models that show how the system will be deployed and operated on the chosen infrastructure. In addition, the chapter includes **interface diagrams**—such as the system interface tree and web page flows—to clarify the user experience and navigation paths. It also outlines the **algorithms used** to support core functionalities, ensuring efficiency and accuracy. To maintain quality assurance, the chapter updates the **test plan** with unit and integration tests, and revises the **requirements traceability matrix** to confirm that every requirement is covered by corresponding test cases. By the end of this chapter, the system design is fully specified at both logical and physical levels, providing a **clear, implementable architecture** that bridges requirements with execution. This foundation prepares the project for the next stage: **detailed design and practical implementation**.

2.Design Models (Logical Level)

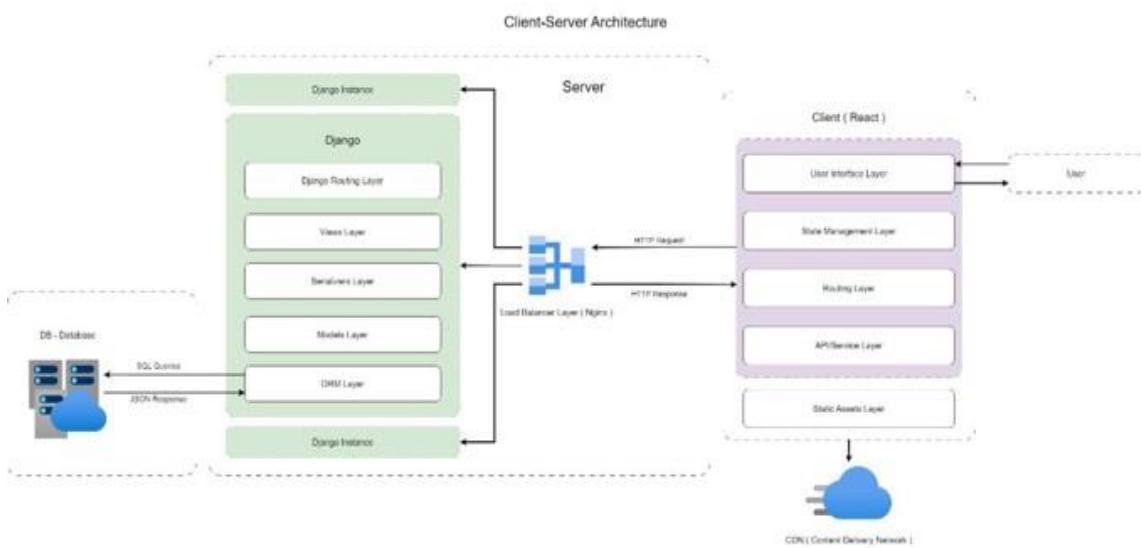
- ERD



• Class Diagram

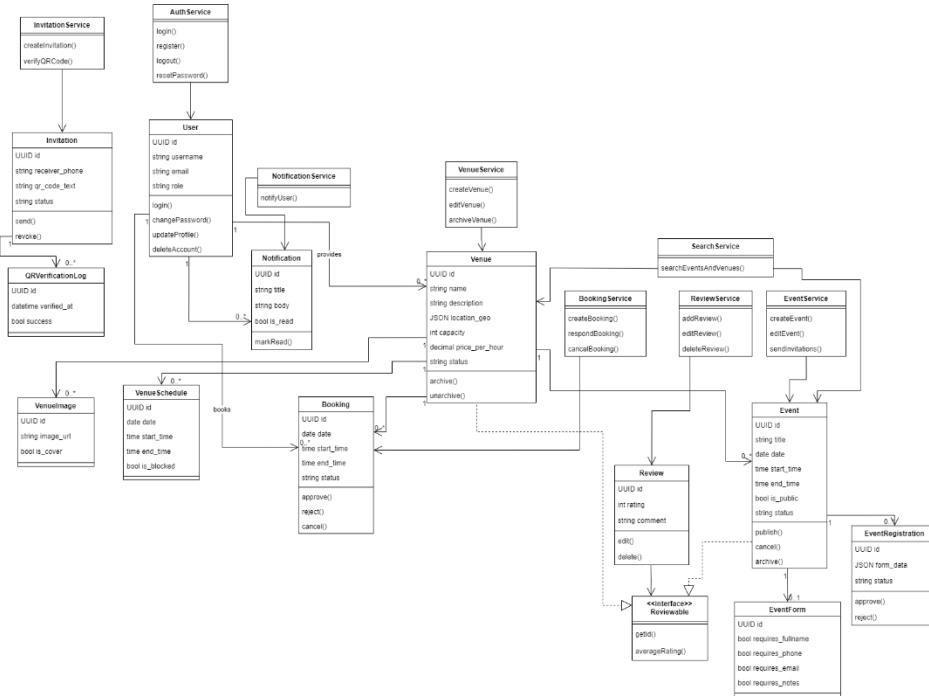


• Component Diagram

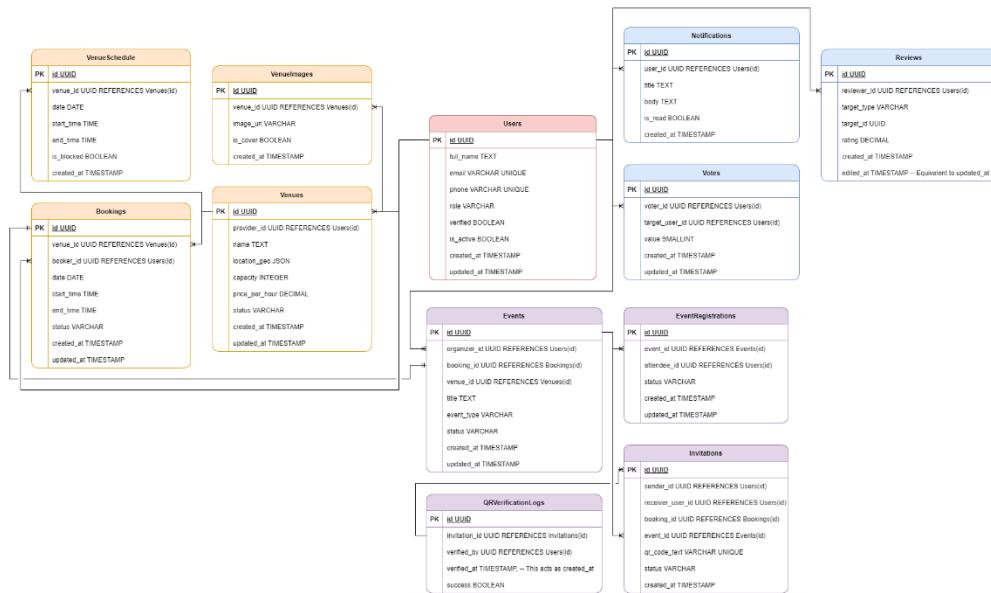


3. Design Models (Physical Level)

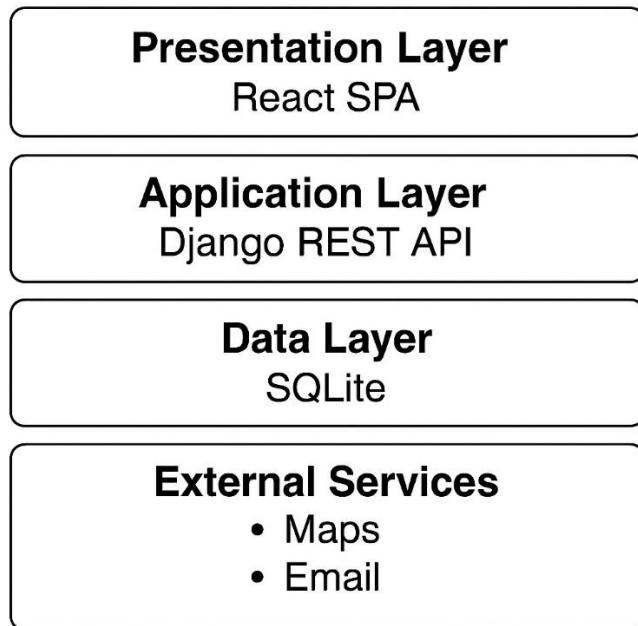
- class Diagram



- Infrastructure Considerations

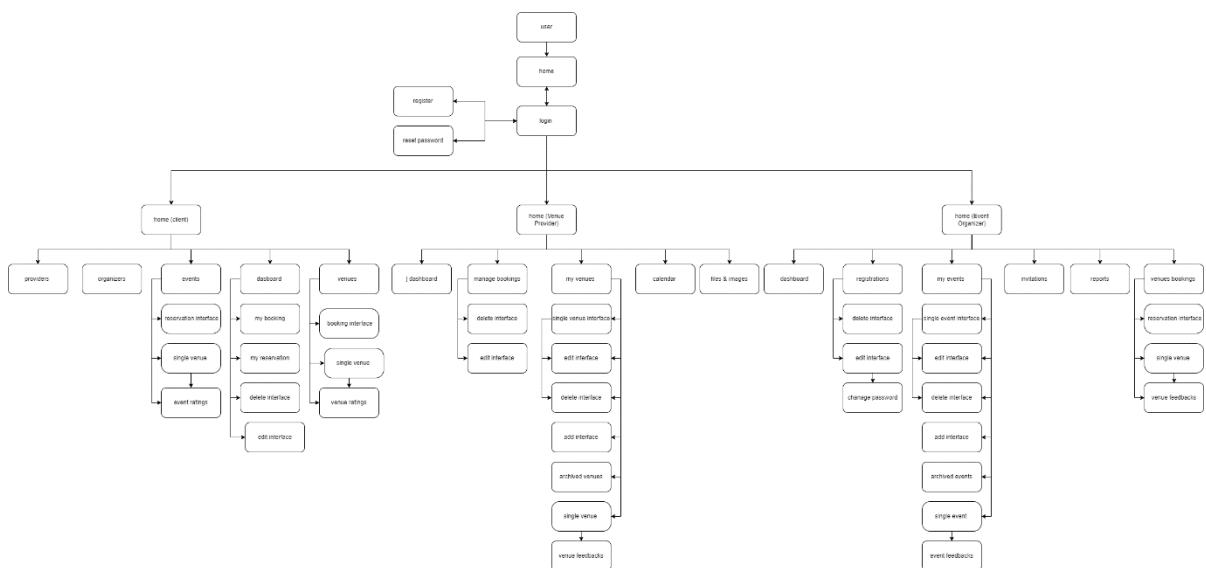


- Physical Architecture Diagram



4. Diagrams

sitemap



5. Algorithms Used

PBKDF2 (Password-Based Key Derivation Function 2)

- **Core Idea:** A cryptographic algorithm that derives secure keys from user passwords using iterative hashing and a salt.
- **Usage:** Protects stored passwords by making brute-force and dictionary attacks computationally expensive.
- **Role in Recommendation Systems:** Ensures user account security and safe handling of authentication data.

NMF (Non-negative Matrix Factorization)

- **Core Idea:** Decomposes a large user-item matrix into smaller non-negative matrices to reveal hidden patterns.
- **Usage:** Learns latent features that represent user preferences and item characteristics.
- **Role in Recommendation Systems:** Generates personalized recommendations by predicting missing values in the user-item matrix.

Pagination Algorithm (Page Number Pagination)

- **Core Idea:** Splits large sets of results into smaller, numbered pages.
- **Usage:** Improves usability and system performance when displaying many recommendations.
- **Role in Recommendation Systems:** Organizes recommended items into manageable sections for better user experience.

Numerical Scoring Algorithm (Data Engineering)

- **Core Idea:** Assigns numerical scores to items based on relevance, importance, or similarity to user preferences.
- **Usage:** Ranks items so the most suitable recommendations appear first.
- **Role in Recommendation Systems:** Provides the scoring backbone that determines which items are prioritized in the recommendation list.

6. Updated Unit Testing

Test Case ID	Test Case Name	Objective	Preconditions	Test Steps	Input Data	Expected Result
TC-UC01-01	Successful login with valid credentials	Verify login, session/token issuance, redirect, and login logging	Login page open; valid account exists	Enter valid email and password; click Login	email:user@example.com; password:CorrectPass123	Token/session issued; redirect to dashboard; login event logged
TC-UC01-02	Login with invalid password	Verify authentication rejection for wrong password	Login page open	Enter valid email and wrong password; click Login	email:user@example.com; password:WrongPass	Authentication error shown; no token issued; no session
TC-UC01-03	Login with inactive account	Prevent login for inactive account	Account exists but inactive	Enter inactive account credentials; click Login	email:inactive@example.com ; password:Pass12345	Account activation required message; no token issued
TC-UC02-01	Successful logout	Verify session termination, redirect to login, and logout logging	User is logged in	Click Logout	—	Session invalidated/revoked; redirect to login; logout event logged
TC-UC03-01	Successful registration	Verify account creation, confirmation email, activation and basic permissions	Registration page open	Fill valid fields; click Register	name:Sara; email:sara@example.com; password:StrongPass1	Account created; confirmation email sent; account activated or pending activation
TC-UC03-02	Register with duplicate email	Reject duplicate email registration	Registration page open; email already exists	Fill form with existing email; click Register	email:user@example.com; password:AnyPass	Error: email already in use; no account created
TC-UC04-01	View venues list default sort & pagination	Verify venues displayed as cards with default sort and pagination	User on venues page	Open venues page; navigate pages	—	Cards show name; location; capacity; price; thumbnail; rating; pagination present
TC-UC05-01	View venue details existing venue	Verify full venue detail page content	Valid venue exists	Open venue detail link	venue_id:12345	Image gallery; full description; pricing; capacity; availability; action buttons present
TC-UC05-02	View venue details not found	Handle venue-not-found gracefully	Invalid venue ID	Open detail link with invalid ID	venue_id:99999	"Venue not found" message or redirect; no server error
TC-UC06-01	View user bookings list	Show current and past bookings with details and filters	User logged in with bookings	Open bookings page; apply filters	—	List shows booking number; venue; date/time; status; sorting/filtering available
TC-UC07-01	Add booking successful	Create booking with availability check and confirmation	User logged in; venue available	Fill booking form; submit	user_id:10; venue_id:123; date:2026-03-10; time:18:00	Availability checked; booking created atomically; confirmation sent
TC-UC08-01	Edit booking with revalidation	Modify booking fields with availability	Existing editable booking	Open booking; change allowed fields; save	booking_id:200; new_time:19:00	Availability revalidated; changes saved; audit log recorded

		revalidation and audit log				
TC-UC09-01	Delete booking with confirmation	Cancel/delete booking applying policies and logging reason	Existing booking	Click Delete; confirm	booking_id:200; reason:"Client request"	Booking cancelled/deleted; cancellation policy applied; action logged
TC-UC10-01	View events list with pagination	Display events list with basic info and pagination	Events page open	Open events page; navigate pages	—	Events show name; date; location; thumbnail; pagination supported
TC-UC11-01	View event details	Show event page with full details and ticket links	Event exists	Open event detail	event_id:200	Time/date; venue; description; ticket options; organizers; map if available
TC-UC12-01	Unified search venues partial match	Unified keyword search across venues with partial matches and pagination	Search page open	Enter partial keyword; search	query:"hall"	Results categorized under Venues; partial matches returned; pagination
TC-UC13-01	Unified search events exact match	Unified keyword search across events with pagination	Search page open	Enter event name; search	query:"Tech Summit 2026"	Results categorized under Events; matching results returned; pagination
TC-UC14-01	Filter venues by multiple criteria	Multi-criteria filtering with reset option	Venues page open	Apply location; price; capacity filters	location:Damascus; price_min:100; price_max:500	Filtered venues displayed; reset filters works
TC-UC15-01	Filter events by multiple criteria	Multi-criteria filtering with reset option	Events page open	Apply date; type filters	date:2026-05-01; type:Conference	Filtered events displayed; reset filters works
TC-UC16-01	View user registrations	Show user's event registrations with status and pagination	User logged in with registrations	Open registrations page	—	Registrations list with attendance status and event details; pagination
TC-UC17-01	Register for paid event	Register with availability check, payment, and ticket delivery	User logged in; tickets available	Select ticket; complete payment; confirm	event_id:200; ticket_type:VIP; payment:card	Availability checked; payment processed; ticket and confirmation delivered
TC-UC18-01	Edit registration within policy	Edit registration subject to policy and availability checks	Existing editable registration	Open registration; modify allowed fields; save	registration_id:555; new_ticket:Standard	Availability revalidated; changes saved; confirmation sent
TC-UC19-01	Cancel registration with confirmation	Cancel registration and update ticket availability	Existing registration	Click Cancel; confirm	registration_id:555	Registration cancelled; ticket availability updated; action logged
TC-UC20-01	View venue ratings list	Display reviews, average, count, and individual reviews with filters	Venue has reviews	Open reviews tab	venue_id:123	Average rating; total count; list of reviews; sorting/filtering available
TC-UC21-01	Submit venue rating eligible user	Submit rating and update averages; prevent duplicates	User eligible (past booking)	Submit rating and comment; save	venue_id:123; rating:5; comment:"Great"	Rating accepted; duplicates prevented; averages updated

TC-UC22-01	Edit venue rating within window	Allow reviewer to edit rating within allowed time and audit	Rating exists and owned by user	Edit rating; save	rating_id:789; new_rating:4	Rating updated; average recalculated; edit recorded in audit trail
TC-UC23-01	Delete venue rating with confirmation	Delete rating, update average, and log deletion	Rating owned by user	Delete rating; confirm	rating_id:789	Rating removed; average updated; deletion logged
TC-UC24-01	View event ratings when none exist	Encourage first rating if none exist	Event has no ratings	Open ratings tab	event_id:200	Message encouraging first rating; average = N/A
TC-UC25-01	Add event rating after attendance	Submit rating subject to attendance policy and prevent duplicates	User attended event	Submit rating; save	event_id:200; rating:4	Rating accepted; duplicates prevented; optional organizer notification
TC-UC26-01	Edit event rating within timeframe	Edit rating within allowed timeframe and update averages	Rating exists and owned by user	Edit rating; save	rating_id:900; new_rating:5	Rating updated; average recalculated; change recorded
TC-UC27-01	Delete event rating and update avg	Delete rating and log action	Rating exists	Delete rating; confirm	rating_id:900	Rating removed; average updated; deletion logged
TC-UC28-01	View organizers list	List organizers with summary info and links	Organizers exist	Open organizers page	—	List shows name; short bio; past events count; overall rating; details link
TC-UC29-01	View providers list	List providers with venue counts and contact links	Providers exist	Open providers page	—	List shows provider name; number of venues; rating; contact/view links
TC-UC30-01	Filter organizers by criteria	Filter by city; rating; experience with reset	Organizers page open	Apply city and rating filters	city:Aleppo; rating_min:4	Filtered organizers displayed; reset works
TC-UC31-01	Filter providers by criteria	Filter by city; rating; experience with reset	Providers page open	Apply experience filter	experience_min:3	Filtered providers displayed; reset works
TC-UC32-01	Upvote content once per user	Record upvote; prevent duplicates; log voter and timestamp	User logged in	Click Upvote on content	content_id:321	UI count increments immediately; duplicate prevented; vote logged
TC-UC33-01	Downvote content once per user	Record downvote; prevent duplicates; log voter and timestamp	User logged in	Click Downvote on content	content_id:321	UI count updates immediately; duplicate prevented; vote logged
TC-UC34-01	View recent activities timeline	Show user's activity timeline with filters and links	User has recent activities	Open activities page; apply filter	filter:bookings	Timeline shows activities with timestamps and links; filters work
TC-UC35-01	Display interactive map with pins	Show event/venue pins with info windows and filtering	Location data available	Open map; zoom; click pin	—	Pins displayed; info windows show details; clustering and filters supported
TC-UC36-01	Browse provider venues list	Provider dashboard lists venues	Provider account with venues	Login as provider; open dashboard	provider_id:77	Provider sees venue cards with search; filters; edit links

		with search and filters				
TC-UC37-01	View archived venues and restore	List archived venues with metadata and restore actions	Provider has archived venues	Open archived tab; select Restore	venue_id:400	Archive date/reason shown; restore works; action logged
TC-UC38-01	Provider adds new venue successfully	Create venue with required fields and media handling	Add venue page open	Fill required fields; upload images; save	name:"Hall A"; location:Homs; images:3	Validation passes; venue saved; media processed; appears in provider dashboard
TC-UC39-01	Edit venue with schedule validation	Edit venue data with conflict checks and audit trail	Venue exists and editable	Edit fields; save	venue_id:123; new_schedule:....	Conflicts validated; changes saved; public listing updated; audit recorded
TC-UC40-01	Prevent deletion if active bookings exist	Block deletion when constraints exist and log attempt	Venue has active bookings	Attempt to delete venue	venue_id:123	Deletion blocked with explanatory message; attempt logged
TC-UC41-01	Archive venue hides from public	Archive venue and record metadata	Venue exists	Archive venue; confirm	venue_id:123	Status set to Archived; hidden from public; archive metadata logged
TC-UC42-01	Unarchive venue restores listing	Restore archived venue and reindex	Venue is archived	Unarchive venue; confirm	venue_id:123	Venue restored; reindexed; restoration logged
TC-UC43-01	Provider views bookings with actions	Provider sees bookings with management actions	Provider has bookings	Open provider bookings page; apply filters	provider_id:77	Bookings listed with accept/reject/detail actions
TC-UC44-01	Provider accepts pending booking	Accept booking after availability recheck and notify client	Booking status Pending	Click Accept; confirm	booking_id:888	Availability rechecked; status set to Accepted; resources reserved; client notified
TC-UC45-01	Provider rejects booking with reason	Reject booking, release resources, notify client and log	Booking status Pending	Click Reject; enter reason; confirm	booking_id:889; reason:"Unavailable"	Status set to Rejected; resources released; client notified; reason logged
TC-UC46-01	Provider cancels accepted booking	Cancel accepted booking, handle refunds, notify client	Booking Accepted/Confirmed	Click Cancel; confirm; process refunds if applicable	booking_id:890	Status set to Cancelled; refunds processed if applicable; client notified; action logged
TC-UC47-01	Browse events dashboard (organizer)	Organizer sees events with statuses and management actions	Organizer account with events	Login as organizer; open events dashboard	organizer_id:55	Events listed with Active/Draft/Archived statuses and management actions
TC-UC48-01	View archived events and restore	List archived events with metadata and restore options	Organizer has archived events	Open archived tab; select Restore	event_id:300	Archive metadata shown; restore works; action logged
TC-UC49-01	Create event with venue availability check	Create event with venue checks and draft/publish options	Create event page open; venue available	Fill event details; check availability; save or publish	title:"Dev Meetup"; venue_id:123; dates:2026-06-10	Availability confirmed; event created; can save as draft or publish; organizer notified
TC-UC50-01	Edit event and notify attendees if needed	Edit event and validate impacts on tickets/schedules	Published event with registrants	Edit dates/sessions; save	event_id:300; new_date:2026-06-12	System validates impacts; changes saved; affected attendees notified

TC-UC51-01	Prevent deletion with unresolved refunds	Block deletion if refunds/disputes exist and notify organizer	Event has pending refunds/disputes	Attempt to delete event	event_id:300	Deletion blocked; reason logged; organizer notified
TC-UC52-01	Archive event hides from public	Archive event and record metadata	Event exists	Archive event; confirm	event_id:300	Status set to Archived; hidden from public; archive metadata logged
TC-UC53-01	Unarchive event restores listing	Restore archived event and reindex	Event is archived	Unarchive event; confirm	event_id:300	Event restored; reindexed; restoration logged
TC-UC54-01	Organizer views registration s with actions	View registrations with filters and quick actions	Organizer has registrations	Open registrations page; apply filters	organizer_id:55	Registrations listed with accept/reject actions and stats
TC-UC55-01	Organizer accepts pending registration	Accept registration, update ticket counts, notify registrant	Registration Pending	Click Accept; confirm	registration_id:700	Status set to Accepted; ticket counts updated; registrant notified
TC-UC56-01	Organizer rejects registration with reason	Reject registration, release tickets, notify and log	Registration Pending	Click Reject; enter reason; confirm	registration_id:701; reason:"Incomplete"	Status set to Rejected; tickets released; registrant notified; reason logged
TC-UC57-01	Cancel registration and handle refund	Cancel registration with refund handling per policy	Registration Accepted/Confirmed	Click Cancel; confirm; apply refund policy	registration_id:702	Status set to Cancelled; refunds processed if applicable; availability updated; action logged
TC-UC58-01	Export registration s as XLSX	Generate XLSX export for selected criteria and log export	Registrations exist in range	Select date range; click Export	date_from:2026-01-01; date_to:2026-01-31	XLSX file generated and available for download; export logged
TC-UC59-01	Organizer views booking details	Organizer views booking details with management actions	Organizer linked to booking	Open booking detail	booking_id:888	Booking details displayed with management actions
TC-UC60-01	Organizer creates booking on behalf of client	Create booking for client with availability validation and confirmation	Organizer logged in; venue available	Fill booking form for client; submit	client_name:"Ali"; venue_id:123; date:2026-04-05	Availability validated; booking created; payment processed if needed; confirmation sent to client
TC-UC61-01	Organizer edits booking with conflict check	Edit booking and validate conflicts; notify client and audit	Booking exists	Edit allowed fields; save	booking_id:888; new_time:19:00	Conflicts checked; changes saved; client notified; audit recorded
TC-UC62-01	Organizer deletes booking with refund handling	Delete/cancel booking with refund processing and notifications	Booking exists and cancellable	Click Delete; confirm; process refunds if applicable	booking_id:888	Booking deleted or cancelled; refunds processed if applicable; client notified; action logged
Test Case ID	Test Case Name		Objective		Preconditions	
TC-UC01-01	Successful login with valid credentials		Verify login, session/token issuance, redirect, and login logging		Login page open; valid account exists	
TC-UC01-02	Login with invalid password		Verify authentication rejection for wrong password		Login page open	
TC-UC01-03	Login with inactive account		Prevent login for inactive account		Account exists but inactive	
TC-UC02-01	Successful logout		Verify session termination, redirect to login, and logout logging		User is logged in	

TC-UC03-01	Successful registration	Verify account creation, confirmation email, activation and basic permissions	Registration page open
TC-UC03-02	Register with duplicate email	Reject duplicate email registration	Registration page open; email already exists
TC-UC04-01	View venues list default sort & pagination	Verify venues displayed as cards with default sort and pagination	User on venues page
TC-UC05-01	View venue details existing venue	Verify full venue detail page content	Valid venue exists
TC-UC05-02	View venue details not found	Handle venue-not-found gracefully	Invalid venue ID
TC-UC06-01	View user bookings list	Show current and past bookings with details and filters	User logged in with bookings
TC-UC07-01	Add booking successful	Create booking with availability check and confirmation	User logged in; venue available
TC-UC08-01	Edit booking with revalidation	Modify booking fields with availability revalidation and audit log	Existing editable booking
TC-UC09-01	Delete booking with confirmation	Cancel/delete booking applying policies and logging reason	Existing booking
TC-UC10-01	View events list with pagination	Display events list with basic info and pagination	Events page open
TC-UC11-01	View event details	Show event page with full details and ticket links	Event exists
TC-UC12-01	Unified search venues partial match	Unified keyword search across venues with partial matches and pagination	Search page open
TC-UC13-01	Unified search events exact match	Unified keyword search across events with pagination	Search page open
TC-UC14-01	Filter venues by multiple criteria	Multi-criteria filtering with reset option	Venues page open
TC-UC15-01	Filter events by multiple criteria	Multi-criteria filtering with reset option	Events page open
TC-UC16-01	View user registrations	Show user's event registrations with status and pagination	User logged in with registrations
TC-UC17-01	Register for paid event	Register with availability check, payment, and ticket delivery	User logged in; tickets available
TC-UC18-01	Edit registration within policy	Edit registration subject to policy and availability checks	Existing editable registration
TC-UC19-01	Cancel registration with confirmation	Cancel registration and update ticket availability	Existing registration
TC-UC20-01	View venue ratings list	Display reviews, average, count, and individual reviews with filters	Venue has reviews
TC-UC21-01	Submit venue rating eligible user	Submit rating and update averages; prevent duplicates	User eligible (past booking)
TC-UC22-01	Edit venue rating within window	Allow reviewer to edit rating within allowed time and audit	Rating exists and owned by user
TC-UC23-01	Delete venue rating with confirmation	Delete rating, update average, and log deletion	Rating owned by user
TC-UC24-01	View event ratings when none exist	Encourage first rating if none exist	Event has no ratings
TC-UC25-01	Add event rating after attendance	Submit rating subject to attendance policy and prevent duplicates	User attended event
TC-UC26-01	Edit event rating within timeframe	Edit rating within allowed timeframe and update averages	Rating exists and owned by user
TC-UC27-01	Delete event rating and update avg	Delete rating and log action	Rating exists
TC-UC28-01	View organizers list	List organizers with summary info and links	Organizers exist
TC-UC29-01	View providers list	List providers with venue counts and contact links	Providers exist
TC-UC30-01	Filter organizers by criteria	Filter by city; rating; experience with reset	Organizers page open
TC-UC31-01	Filter providers by criteria	Filter by city; rating; experience with reset	Providers page open
TC-UC32-01	Upvote content once per user	Record upvote; prevent duplicates; log voter and timestamp	User logged in
TC-UC33-01	Downvote content once per user	Record downvote; prevent duplicates; log voter and timestamp	User logged in
TC-UC34-01	View recent activities timeline	Show user's activity timeline with filters and links	User has recent activities
TC-UC35-01	Display interactive map with pins	Show event/venue pins with info windows and filtering	Location data available
TC-UC36-01	Browse provider venues list	Provider dashboard lists venues with search and filters	Provider account with venues

TC-UC37-01	View archived venues and restore	List archived venues with metadata and restore actions	Provider has archived venues
TC-UC38-01	Provider adds new venue successfully	Create venue with required fields and media handling	Add venue page open
TC-UC39-01	Edit venue with schedule validation	Edit venue data with conflict checks and audit trail	Venue exists and editable
TC-UC40-01	Prevent deletion if active bookings exist	Block deletion when constraints exist and log attempt	Venue has active bookings
TC-UC41-01	Archive venue hides from public	Archive venue and record metadata	Venue exists
TC-UC42-01	Unarchive venue restores listing	Restore archived venue and reindex	Venue is archived
TC-UC43-01	Provider views bookings with actions	Provider sees bookings with management actions	Provider has bookings
TC-UC44-01	Provider accepts pending booking	Accept booking after availability recheck and notify client	Booking status Pending
TC-UC45-01	Provider rejects booking with reason	Reject booking, release resources, notify client and log	Booking status Pending
TC-UC46-01	Provider cancels accepted booking	Cancel accepted booking, handle refunds, notify client	Booking Accepted/Confirmed
TC-UC47-01	Browse events dashboard (organizer)	Organizer sees events with statuses and management actions	Organizer account with events
TC-UC48-01	View archived events and restore	List archived events with metadata and restore options	Organizer has archived events
TC-UC49-01	Create event with venue availability check	Create event with venue checks and draft/publish options	Create event page open; venue available
TC-UC50-01	Edit event and notify attendees if needed	Edit event and validate impacts on tickets/schedules	Published event with registrants
TC-UC51-01	Prevent deletion with unresolved refunds	Block deletion if refunds/disputes exist and notify organizer	Event has pending refunds/disputes
TC-UC52-01	Archive event hides from public	Archive event and record metadata	Event exists
TC-UC53-01	Unarchive event restores listing	Restore archived event and reindex	Event is archived
TC-UC54-01	Organizer views registrations with actions	View registrations with filters and quick actions	Organizer has registrations
TC-UC55-01	Organizer accepts pending registration	Accept registration, update ticket counts, notify registrant	Registration Pending
TC-UC56-01	Organizer rejects registration with reason	Reject registration, release tickets, notify and log	Registration Pending
TC-UC57-01	Cancel registration and handle refund	Cancel registration with refund handling per policy	Registration Accepted/Confirmed
TC-UC58-01	Export registrations as XLSX	Generate XLSX export for selected criteria and log export	Registrations exist in range
TC-UC59-01	Organizer views booking details	Organizer views booking details with management actions	Organizer linked to booking
TC-UC60-01	Organizer creates booking on behalf of client	Create booking for client with availability validation and confirmation	Organizer logged in; venue available
TC-UC61-01	Organizer edits booking with conflict check	Edit booking and validate conflicts; notify client and audit	Booking exists
TC-UC62-01	Organizer deletes booking with refund handling	Delete/cancel booking with refund processing and notifications	Booking exists and cancellable

7. Updated Requirements Traceability Matrix

Table 4.888 Requirements Traceability Matrix

Requirement ID	Analysis	Design	Application Interface	Code	Test Case ID
REQ-A01	Login	Login - activity diagram ; Login - sequence diagram			initial test cases
REQ-A02	Logout	Logout - activity diagram ; Logout - sequence diagram			initial test cases
REQ-A03	Register (User Registration)	Register (User Registration) - activity diagram ; Register (User Registration) - sequence diagram			initial test cases
REQ-A04	View Venues	View Venues - activity diagram ; View Venues - sequence diagram			initial test cases
REQ-A05	View Venue Details	View Venue Details - activity diagram ; View Venue Details - sequence diagram			initial test cases
REQ-A06	View Bookings	View Bookings - activity diagram ; View Bookings - sequence diagram			initial test cases
REQ-A07	Add Booking	Add Booking - activity diagram ; Add Booking - sequence diagram			initial test cases
REQ-A08	Edit Booking	Edit Booking - activity diagram ; Edit Booking - sequence diagram			initial test cases
REQ-A09	Delete Booking	Delete Booking - activity diagram ; Delete Booking - sequence diagram			initial test cases
REQ-A10	View Events	View Events - activity diagram ; View Events - sequence diagram			initial test cases
REQ-A11	View Event Details	View Event Details - activity diagram ; View Event Details - sequence diagram			initial test cases

REQ-A12	Unified Search for Venues	Unified Search for Venues - activity diagram ; Unified Search for Venues - sequence diagram			initial test cases
REQ-A13	Unified Search for Events	Unified Search for Events - activity diagram ; Unified Search for Events - sequence diagram			initial test cases
REQ-A14	Filter Venues by Specific Criteria	Filter Venues by Specific Criteria - activity diagram ; Filter Venues by Specific Criteria - sequence diagram			initial test cases
REQ-A15	Filter Events by Specific Criteria	Filter Events by Specific Criteria - activity diagram ; Filter Events by Specific Criteria - sequence diagram			initial test cases
REQ-A16	View Registrations	View Registrations - activity diagram ; View Registrations - sequence diagram			initial test cases
REQ-A17	Add Registration	Add Registration - activity diagram ; Add Registration - sequence diagram			initial test cases
REQ-A18	Edit Registration	Edit Registration - activity diagram ; Edit Registration - sequence diagram			initial test cases
REQ-A19	Delete Registration	Delete Registration - activity diagram ; Delete Registration - sequence diagram			initial test cases
REQ-A20	View Venue Ratings	View Venue Ratings - activity diagram ; View Venue Ratings - sequence diagram			initial test cases
REQ-A21	Add Venue Rating	Add Venue Rating - activity diagram ; Add Venue Rating - sequence diagram			initial test cases
REQ-A22	Edit Venue Rating	Edit Venue Rating - activity diagram ; Edit Venue Rating - sequence diagram			initial test cases

REQ-A23	Delete Venue Rating	Delete Venue Rating - activity diagram ; Delete Venue Rating - sequence diagram			initial test cases
REQ-A24	View Event Ratings	View Event Ratings - activity diagram ; View Event Ratings - sequence diagram			initial test cases
REQ-A25	Add Event Rating	Add Event Rating - activity diagram ; Add Event Rating - sequence diagram			initial test cases
REQ-A26	Edit Event Rating	Edit Event Rating - activity diagram ; Edit Event Rating - sequence diagram			initial test cases
REQ-A27	Delete Event Rating	Delete Event Rating - activity diagram ; Delete Event Rating - sequence diagram			initial test cases
REQ-A28	View Organizers	View Organizers - activity diagram ; View Organizers - sequence diagram			initial test cases
REQ-A29	View Providers	View Providers - activity diagram ; View Providers - sequence diagram			initial test cases
REQ-A30	Filter Organizers by Specific Criteria	Filter Organizers by Specific Criteria - activity diagram ; Filter Organizers by Specific Criteria - sequence diagram			initial test cases
REQ-A31	Filter Providers by Specific Criteria	Filter Providers by Specific Criteria - activity diagram ; Filter Providers by Specific Criteria - sequence diagram			initial test cases
REQ-A32	Upvote	Upvote - activity diagram ; Upvote - sequence diagram			initial test cases
REQ-A33	Downvote	Downvote - activity diagram ; Downvote - sequence diagram			initial test cases
REQ-A34	View Recent Activities	View Recent Activities - activity diagram ; View Recent Activities - sequence diagram			initial test cases

REQ-A35	Display Map of Event and Venue Locations	Display Map of Event and Venue Locations - activity diagram ; Display Map of Event and Venue Locations - sequence diagram			initial test cases
REQ-A36	Browse Venues (Provider)	Browse Venues (Provider) - activity diagram ; Browse Venues (Provider) - sequence diagram			initial test cases
REQ-A37	Browse Archived Venues	Browse Archived Venues - activity diagram ; Browse Archived Venues - sequence diagram			initial test cases
REQ-A38	Add Venue	Add Venue - activity diagram ; Add Venue - sequence diagram			initial test cases
REQ-A39	Edit Venue	Edit Venue - activity diagram ; Edit Venue - sequence diagram			initial test cases
REQ-A40	Delete Venue	Delete Venue - activity diagram ; Delete Venue - sequence diagram			initial test cases
REQ-A41	Archive Venue	Archive Venue - activity diagram ; Archive Venue - sequence diagram			initial test cases
REQ-A42	Unarchive Venue	Unarchive Venue - activity diagram ; Unarchive Venue - sequence diagram			initial test cases
REQ-A43	Browse Bookings (Provider)	Browse Bookings (Provider) - activity diagram ; Browse Bookings (Provider) - sequence diagram			initial test cases
REQ-A44	Accept Booking (Provider)	Accept Booking (Provider) - activity diagram ; Accept Booking (Provider) - sequence diagram			initial test cases
REQ-A45	Reject Booking (Provider)	Reject Booking (Provider) - activity diagram ; Reject Booking (Provider) - sequence diagram			initial test cases

REQ-A46	Cancel Booking (Provider)	Cancel Booking (Provider) - activity diagram ; Cancel Booking (Provider) - sequence diagram			initial test cases
REQ-A47	Browse Events (Organizer)	Browse Events (Organizer) - activity diagram ; Browse Events (Organizer) - sequence diagram			initial test cases
REQ-A48	Browse Archived Events	Browse Archived Events - activity diagram ; Browse Archived Events - sequence diagram			initial test cases
REQ-A49	Create Event	Create Event - activity diagram ; Create Event - sequence diagram			initial test cases
REQ-A50	Edit Event	Edit Event - activity diagram ; Edit Event - sequence diagram			initial test cases
REQ-A51	Delete Event	Delete Event - activity diagram ; Delete Event - sequence diagram			initial test cases
REQ-A52	Archive Event	Archive Event - activity diagram ; Archive Event - sequence diagram			initial test cases
REQ-A53	Unarchive Event	Unarchive Event - activity diagram ; Unarchive Event - sequence diagram			initial test cases
REQ-A54	Browse Registrations (Organizer)	Browse Registrations (Organizer) - activity diagram ; Browse Registrations (Organizer) - sequence diagram			initial test cases
REQ-A55	Accept Registration	Accept Registration - activity diagram ; Accept Registration - sequence diagram			initial test cases
REQ-A56	Reject Registration	Reject Registration - activity diagram ; Reject Registration - sequence diagram			initial test cases
REQ-A57	Cancel Registration	Cancel Registration - activity diagram ;			initial test cases

		Cancel Registration - sequence diagram			
REQ-A58	Export Registrations as Excel file	Export Registrations as Excel file - activity diagram ; Export Registrations as Excel file - sequence diagram			initial test cases
REQ-A59	View Booking (Organizer)	View Booking (Organizer) - activity diagram ; View Booking (Organizer) - sequence diagram			initial test cases
REQ-A60	Add Booking (Organizer)	Add Booking (Organizer) - activity diagram ; Add Booking (Organizer) - sequence diagram			initial test cases
REQ-A61	Edit Booking (Organizer)	Edit Booking (Organizer) - activity diagram ; Edit Booking (Organizer) - sequence diagram			initial test cases
REQ-A62	Delete Booking (Organizer)	Delete Booking (Organizer) - activity diagram ; Delete Booking (Organizer) - sequence diagram			initial test cases
REQ-A63	Export Registrations as Excel file	Export Registrations as Excel file - sequence diagram ; Export Registrations as Excel file - activity diagram			initial test cases
REQ-A64	View e-ticket	View e-ticket - sequence diagram ; View e-ticket - activity diagram			initial test cases
REQ-A65	Create e-ticket	Create e-ticket - sequence diagram ; Create e-ticket - activity diagram			initial test cases
REQ-A66	Export e-tickets list as PDF file	Export e-tickets list as PDF file - sequence diagram ; Export e-tickets list			initial test cases

		as PDF file - activity diagram			
REQ-A67	Display an interactive calendar	Display an interactive calendar - sequence diagram; Display an interactive calendar - activity diagram			initial test cases

8. Conclusion

In this chapter, we moved from the general execution plan into the **detailed design phase**, where the documented requirements were translated into clear technical models. We began with an introduction outlining the chapter's objectives, followed by a presentation of **logical design models** such as the database schema, class diagram, and component diagram, which illustrate the relationships between entities and the system's core functionalities. We then explored the **physical design models**, which demonstrate how the system will be deployed and operated on the proposed infrastructure.

We also presented **interface diagrams**, including the page tree and user interaction flows, which help clarify the expected user experience. Next, we reviewed the **algorithms used** for data processing and critical operations, highlighting how efficiency and accuracy are ensured.

In addition, we updated the **test plan** to include both unit tests and integration tests, and revised the **requirements traceability matrix** to ensure that every required function is covered by appropriate testing.

This chapter serves as a bridge between theoretical requirements and the practical execution plan, providing a **technically implementable design** that aligns with the project timeline and testing strategy. The following chapters will move into **detailed design and practical implementation**, ensuring that these models are transformed into a working system that meets the defined objectives.

Chapter Six

Artificial Intelligence

Smart Recommendation System - Technical Architecture Report

1. Data Acquisition Strategy

This phase serves as the foundation, where random user behaviors are transformed into analytical data points.

Process Description:

A Real-time Monitoring mechanism was implemented using Django Signals. "Listeners" were attached to vital system tables (Bookings, Reviews, Votes). Once a user performs an action, the event is captured and passed to the data processor.

2. Data Persistence Layer

Data is aggregated into a dedicated repository, separate from operational tables, to ensure query efficiency.

Process Description:

The signal is received and stored in a specialized relational table named UserInteraction. This table acts as a historical "Log" for all system interactions. Data is unified here, treating both Venues and Events as standard "Target Items."

This layer produces:

UserInteraction	
PK	<u>id UUID</u>
	user_id UUID NOT NULL
	interaction_type VARCHAR(20) NOT NULL
	target_type VARCHAR(10) NOT NULL
	target_id UUID NOT NULL
	rating DECIMAL(2,1)
	vote_value
	timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL

3. Data Preprocessing & Transformation

Converting data from its "storage" format into a "mathematical" format compatible with AI models.

Process Description:

1. Numerical Scoring/Weighting: Converting qualitative interactions into quantitative values representing "Interest Intensity":
 - Booking → 10.0

- Registration → 8.0
- Review → 5.0 / 0.1
- Vote (Up/Down) → 1 / -1
- Visit → 1.0

2. Matrix Construction: Transforming tabular data into a Sparse Matrix (CSR Format). Rows represent Users, columns represent Items, and the values are the calculated weights.
-

4. Model Training & Algorithm

The "Brain of the System" where hidden patterns are discovered.

Process Description:

The system utilizes the Non-negative Matrix Factorization (NMF) algorithm from the Scikit-Learn library. The algorithm factorizes the massive interaction matrix into two smaller matrices representing Latent Features:

1. User Features Matrix (W)

2. Item Features Matrix (H)

- Hyperparameters: Set to n_components=15 and max_iter=500 to balance accuracy and performance.
 - Persistence: The Pickle library is used to save the trained model into a physical file.

 - Inputs: Sparse Interaction Matrix, Algorithm Hyperparameters.
 - Outputs: recommender_model.pkl file.
-

5. Backend Integration

Integrating the AI model into the Django web application.

- Process Description:

An engineering class, RecommenderEngine, was developed using the Singleton Pattern (to load the model into RAM only once). Upon request, the engine performs the following calculation:

$$\text{Prediction} = \text{UserVector} \cdot \text{ItemMatrix}$$

Results are then sorted in descending order to extract the Top-N recommendations.

- Inputs: user_id, Number of recommendations required (N).
- Outputs: List of Recommended Item UUIDs.

6. Data Delivery & API

Delivering organized and intelligent results to the end-user.

- Process Description:

APIViews were built using Django Rest Framework (DRF). This stage includes smart logic to handle different scenarios:

1. Verification: Does the user have an existing record in the model?
2. Personalization: If yes, recommendations are fetched from the engine.
3. Fallback Mechanism: If the user is new (Cold Start), the system fetches "Most Popular" items directly from the database.
4. Serialization: Converting IDs into full objects including images, prices, and geographic location.
5. Endpoints: Dedicated endpoints were created for Venues (/venues/) and Events (/events/)

This layer produces:

GET: /api/recommendation /venues/{user_id}/

GET: /api/recommendation /events/{user_id}/

7. Scalability & Maintenance Cycle

Ensuring the sustained accuracy of the system over time.

- Process Description:

Since user tastes and item availability change constantly, the system adopts a Periodic Retraining strategy.

- A custom Django Management Command was designed.
- The task is scheduled via Celery to run Weekly.
- The command pulls all new data from UserInteraction, rebuilds the matrix, and updates the .pkl file with Zero Downtime.

Chapter Seven

Practical application

1. introduction

This chapter presents a practical application of the development methodology for an event and booking management system, transforming theoretical requirements into measurable, actionable steps: it describes the tools and environments used, presents illustrative miniature models of the system components and workflows, explains the steps for executing specific tests and shows their results, and clarifies how to update the requirements traceability matrix based on test outputs, while drawing lessons learned and practical recommendations for improvement before moving to integration and deployment phases.

2. Tools Used

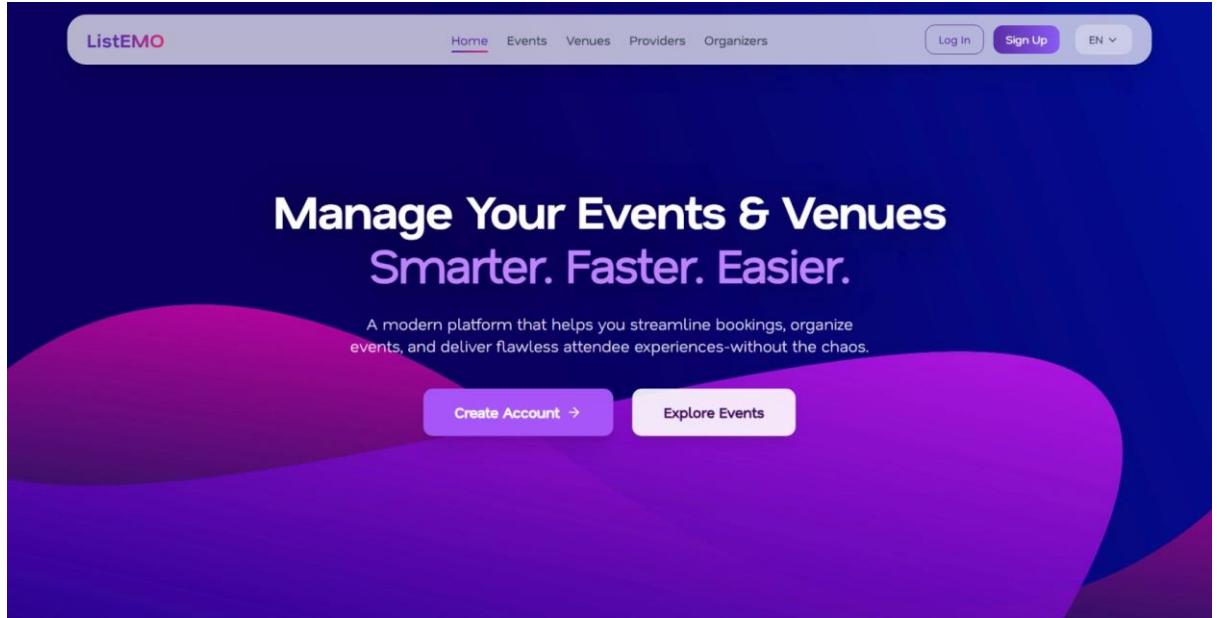
In this project, an integrated set of tools and technologies was selected to support user interface development, build robust APIs, manage data, and streamline the development and testing workflow. The selection focused on lightweight, open-source solutions that enable rapid delivery, easy maintenance, and future scalability, along with productivity and collaboration aids for the team.

- **Python** — the primary programming language for backend logic and processing, used to implement services and background tasks with clarity and testability.
- **Django REST Framework** — a framework for building secure, well-structured RESTful APIs, supporting authentication, permissions, and standardized serialization/deserialization.
- **HTML / CSS / JavaScript** — core web technologies for constructing front-end pages, designing the visual experience, and ensuring cross-browser compatibility.
- **React** — a UI library for building interactive, reusable components that accelerate front-end development and simplify client-side state management.
- **SQLite** — a lightweight database suitable for development and testing environments, offering fast local data storage and easy backup/transfer.
- **Visual Studio Code** — a lightweight integrated development environment that supports code editing, testing tool integration, and extensions for debugging and package management.
- **GitHub** — a version control and repository hosting platform for coordinating collaborative work, code reviews, release management, and deployment workflows.

Each tool was chosen to fulfill a specific role in the project lifecycle: Python and Django provide a reliable, testable API layer; React and web technologies deliver an interactive user experience; SQLite speeds up development and testing setups; and VS Code with GitHub supports collaboration, code review, and change tracking. This combination balances development velocity, code quality, and a smooth path to migrate later to more robust production databases and services as needed.

3. Miniatures:

- Home page



- Venue provider

The screenshot shows the "venue providers" section of the ListEMO Client Suite. On the left is a sidebar with a "MAIN MENU" containing "Dashboard", "VENUE REGISTRATIONS", "VENUE RATINGS", "EVENT RATINGS", and "HOSTS" sections. The "HOSTS" section has a "Venue Providers" item selected, which is highlighted with a purple background. Below the sidebar is a "client" section with a "client" icon and a "client" label. The main content area is titled "venue providers" and describes "View People who list and manage venues." It features a search bar, a sort by dropdown, and an advanced search link. Below this is a table with three rows, each representing a venue provider. Each row contains a profile picture, the provider's name, their email, their score (3.0, 1.0, or 1.0), the number of votes (3, 3, or 3), a rating scale, and a "Rate host" button. At the bottom of each row is a "View Venues" button.

Provider	Email	Score	Votes	Rate host	Action
provider	provider@gmail.com	3.0	3	Rate host: ⬆️ ⬇️ ✓ You upvoted	View Venues
provider1	provider1@gmail.com	1.0	3	Rate host: ⬆️ ⬇️ ✓ You upvoted	View Venues
provider2	provider2@gmail.com	1.0	3	Rate host: ⬆️ ⬇️ ✓ You upvoted	View Venues

• Venue ratings

The screenshot shows the 'Venue Ratings' section of the ListEMO Client Suite. The left sidebar has a 'Venue Ratings' item selected under 'MANAGEMENT'. The main area displays a list of venues with their names, star ratings, and review counts. Each venue entry includes a search bar, a 'Latest' dropdown, and edit/delete icons.

Venue	Rating	Reviews
Beit Jabri Restaurant	★★★★★ (3)	hal Jan 19, 2026
Al Khawali Restaurant	★★★★ (2)	I'll be there for sure Jan 13, 2026
Al Halabi Restaurant	★★★★★ (1)	

• Dashboard

The screenshot shows the 'Dashboard' section of the ListEMO Client Suite. The left sidebar has a 'Dashboard' item selected under 'MAIN MENU'. The main area displays various performance metrics in cards, such as Total Bookings (22), Registrations (9), and Event Ratings (5). Below these are sections for Quick Actions and a grid of smaller cards for New Registration, Rate Event, New Booking, Rate Venue, Venue Providers, and Event Organizers.

Category	Value
Total Bookings	22
Registrations	9
Venue Ratings	22
Event Ratings	5
Event Organizers	1
Venue Providers	4

• Venue bookings

Venue Bookings
Manage all your event bookings in one place.

Search venues...
Min capacity Max capacity Min price / hr Max price / hr Sort by

Al Khawali Restaurant
Al Khawali is a traditional Syrian res.
50 people \$ 2550.00 / hour
View Details → Rating: 3.5

Beit Jabri Restaurant
Beit Jabri is a traditional Damascene
10 people \$ 1200.00 / hour
View Details → Rating: 4.5

Leila's Restaurant
Leila's is a modern, elegant restaura.
15 people \$ 1000.00 / hour
View Details → Rating: 3.5

Al Halabi Restaurant
Al Halabi is a renowned restaurant c.
30 people \$ 1250.00 / hour
View Details → Rating: 5

• Manage bookings

Manage Bookings
Handle and track bookings for your venues.

Search bookings... All Bookings

BOOKING	VENUE	CLIENT	DATE & TIME	NOTES	STATUS	ACTIONS
#BK-0001 Booking request	Beit Jabri Restaurant	client client@gmail.com	2026-01-21 05:39:00 – 08:39:00	—	Pending	✓ ✎ ✖
#BK-0002 Booking request	Naranj Restaurant	client2 client2@gmail.com	2026-01-22 22:05:00 – 12:05:00	—	Rejected	
#BK-0003 Booking request	Dunes Boulevard	organizer organizer@gmail.com	2026-01-16 17:04:00 – 18:04:00	—	Approved	✖
#BK-0004 Booking request	Beit Jabri Restaurant	organizer organizer@gmail.com	2026-01-26 16:02:00 – 19:02:00	—	Approved	✖
#BK-0005	...	organizer	2026-01-29	...		

• Add venue

Add Venue

Venue Name:

Description:

Area: _____ City: _____

Capacity: _____ Price / hour: _____

Venue Location
Select governorate & drag the pin

Damascus

Drag the marker to adjust location

Belt Jabri Restaurant

Damascus, Old City

10 guests 41200.00

★★★★★

• My events

My Events
Manage your event listings.

Create Event
Event details

Select Booking
Choose venue

Event Summary

Title: asdfasdf

Type: roundedTable

142 attendees

Date: 2026-01-19

Time: 00:31 ~ 01:31

Description: asdfasdf

Venue & Booking

Dunes Boulevard
Dunes Boulevard, located in the Mezze district of Damascus, is part of the well-known Dunes Club complex. It offers a stylish blend of dining, leisure, and social experiences. The boulevard is designed as a contemporary avenue where visitors can enjoy a variety of restaurants, cafés, and lounges, each with its own unique concept and atmosphere.

Approved

Al-Mazeh, Damascus

4. Test Execution

Test Case ID	Test Case Name	Objective	Preconditions	Test Steps	Input Data	Expected Result	Actual Result	Status
TC-UC01-01	Successful login with valid credentials	Verify login, session/token issuance, redirect, and login logging	Login page open; valid account exists	Enter valid email and password; click Login	email:user@example.com; password:CorrectPass123	Token/session issued; redirect to dashboard; login event logged	Token/session issued; redirect to dashboard; login event logged	Pass
TC-UC01-02	Login with invalid password	Verify authentication rejection for wrong password	Login page open	Enter valid email and wrong password; click Login	email:user@example.com; password:WrongPass	Authentication error shown; no token issued; no session	Authentication error shown; no token issued; no session	Pass
TC-UC01-03	Login with inactive account	Prevent login for inactive account	Account exists but inactive	Enter inactive account credentials; click Login	email:inactive@example.com; password:Pass12345	Account activation required message; no token issued	Account activation required message; no token issued	Pass
TC-UC02-01	Successful logout	Verify session termination, redirect to login, and logout logging	User is logged in	Click Logout	—	Session invalidated/revoked; redirect to login; logout event logged	Session invalidated/revoked; redirect to login; logout event logged	Pass
TC-UC03-01	Successful registration	Verify account creation, confirmation email, activation and basic permissions	Registration page open	Fill valid fields; click Register	name:Sara; email:sara@example.com; password:StrongPass1	Account created; confirmation email sent; account activated or pending activation	Account created; confirmation email sent; account activated or pending activation	Pass
TC-UC03-02	Register with duplicate email	Reject duplicate email registration	Registration page open; email already exists	Fill form with existing email; click Register	email:user@example.com; password:AnyPass	Error: email already in use; no account created	Error: email already in use; no account created	Pass
TC-UC04-01	View venues list default sort & pagination	Verify venues displayed as cards with default sort and pagination	User on venues page	Open venues page; navigate pages	—	Cards show name; location; capacity; price; thumbnail; rating; pagination present	Cards show name; location; capacity; price; thumbnail; rating; pagination present	Pass
TC-UC05-01	View venue details existing venue	Verify full venue detail page content	Valid venue exists	Open venue detail link	venue_id:12345	Image gallery; full description; pricing; capacity; availability; action buttons present	Image gallery; full description; pricing; capacity; availability; action buttons present	Pass
TC-UC05-02	View venue details not found	Handle venue-not-found gracefully	Invalid venue ID	Open detail link with invalid ID	venue_id:99999	"Venue not found" message or redirect; no server error	"Venue not found" message or redirect; no server error	Pass
TC-UC06-01	View user bookings list	Show current and past bookings with	User logged in with bookings	Open bookings page; apply filters	—	List shows booking number; venue;	List shows booking number; venue;	Pass

		details and filters				date/time; status; sorting/filtering available	date/time; status; sorting/filtering available	
TC-UC07-01	Add booking successful	Create booking with availability check and confirmation	User logged in; venue available	Fill booking form; submit	user_id:10; venue_id:123; date:2026-03-10; time:18:00	Availability checked; booking created atomically; confirmation sent	Availability checked; booking created atomically; confirmation sent	Pass
TC-UC08-01	Edit booking with revalidation	Modify booking fields with availability revalidation and audit log	Existing editable booking	Open booking; change allowed fields; save	booking_id:200; new_time:19:00	Availability revalidated; changes saved; audit log recorded	Availability revalidated; changes saved; audit log recorded	Pass
TC-UC09-01	Delete booking with confirmation	Cancel/delete booking applying policies and logging reason	Existing booking	Click Delete; confirm	booking_id:200; reason:"Client request"	Booking cancelled/deleted; cancellation policy applied; action logged	Booking cancelled/deleted; cancellation policy applied; action logged	Pass
TC-UC10-01	View events list with pagination	Display events list with basic info and pagination	Events page open	Open events page; navigate pages	—	Events show name; date; location; thumbnail; pagination supported	Events show name; date; location; thumbnail; pagination supported	Pass
TC-UC11-01	View event details	Show event page with full details and ticket links	Event exists	Open event detail	event_id:200	Time/date; venue; description; ticket options; organizers; map if available	Time/date; venue; description; ticket options; organizers; map if available	pass
TC-UC12-01	Unified search venues partial match	Unified keyword search across venues with partial matches and pagination	Search page open	Enter partial keyword; search	query:"hall"	Results categorized under Venues; partial matches returned; pagination	Results categorized under Venues; partial matches returned; pagination	Pass
TC-UC13-01	Unified search events exact match	Unified keyword search across events with pagination	Search page open	Enter event name; search	query:"Tech Summit 2026"	Results categorized under Events; matching results returned; pagination	Results categorized under Events; matching results returned; pagination	Pass
TC-UC14-01	Filter venues by multiple criteria	Multi-criteria filtering with reset option	Venues page open	Apply location; price; capacity filters	location:Damascus; price_min:100; price_max:500	Filtered venues displayed; reset filters works	Filtered venues displayed; reset filters works	Pass
TC-UC15-01	Filter events by multiple criteria	Multi-criteria filtering with reset option	Events page open	Apply date; type filters	date:2026-05-01; type:Conference	Filtered events displayed; reset filters works	Filtered events displayed; reset filters works	Pass
TC-UC16-01	View user registrations	Show user's event registrations with status and pagination	User logged in with registrations	Open registrations page	—	Registrations list with attendance status and event details; pagination	Registrations list with attendance status and event details; pagination	Pass

TC-UC17-01	Register for paid event	Register with availability check, payment, and ticket delivery	User logged in; tickets available	Select ticket; complete payment; confirm	event_id:200; ticket_type:VIP; payment:card	Availability checked; payment processed; ticket and confirmation delivered	Availability checked; payment processed; ticket and confirmation delivered	Pass
TC-UC18-01	Edit registration within policy	Edit registration subject to policy and availability checks	Existing editable registration	Open registration; modify allowed fields; save	registration_id:555; new_ticket:Standard	Availability revalidated; changes saved; confirmation sent	Availability revalidated; changes saved; confirmation sent	Pass
TC-UC19-01	Cancel registration with confirmation	Cancel registration and update ticket availability	Existing registration	Click Cancel; confirm	registration_id:555	Registration cancelled; ticket availability updated; action logged	Registration cancelled; ticket availability updated; action logged	Pass
TC-UC20-01	View venue ratings list	Display reviews, average, count, and individual reviews with filters	Venue has reviews	Open reviews tab	venue_id:123	Average rating; total count; list of reviews; sorting/filtering available	Average rating; total count; list of reviews; sorting/filtering available	Pass
TC-UC21-01	Submit venue rating eligible user	Submit rating and update averages; prevent duplicates	User eligible (past booking)	Submit rating and comment; save	venue_id:123; rating:5; comment:"Great"	Rating accepted; duplicates prevented; averages updated	Rating accepted; duplicates prevented; averages updated	Pass
TC-UC22-01	Edit venue rating within window	Allow reviewer to edit rating within allowed time and audit	Rating exists and owned by user	Edit rating; save	rating_id:789; new_rating:4	Rating updated; average recalculated; edit recorded in audit trail	Rating updated; average recalculated ; edit recorded in audit trail	Pass
TC-UC23-01	Delete venue rating with confirmation	Delete rating, update average, and log deletion	Rating owned by user	Delete rating; confirm	rating_id:789	Rating removed; average updated; deletion logged	Rating removed; average updated; deletion logged	Pass
TC-UC24-01	View event ratings when none exist	Encourage first rating if none exist	Event has no ratings	Open ratings tab	event_id:200	Message encouraging first rating; average = N/A	Message encouraging first rating; average = N/A	Pass
TC-UC25-01	Add event rating after attendance	Submit rating subject to attendance policy and prevent duplicates	User attended event	Submit rating; save	event_id:200; rating:4	Rating accepted; duplicates prevented; optional organizer notification	Rating accepted; duplicates prevented; optional organizer notification	Pass
TC-UC26-01	Edit event rating within timeframe	Edit rating within allowed timeframe and update averages	Rating exists and owned by user	Edit rating; save	rating_id:900; new_rating:5	Rating updated; average recalculated; change recorded	Rating updated; average recalculated ; change recorded	Pass
TC-UC27-01	Delete event rating and update avg	Delete rating and log action	Rating exists	Delete rating; confirm	rating_id:900	Rating removed; average updated; deletion logged	Rating removed; average updated; deletion logged	Pass
TC-UC28-01	View organizers list	List organizers with	Organizers exist	Open organizers page	—	List shows name; short bio; past events count;	List shows name; short bio; past events	Pass

		summary info and links				overall rating; details link	count; overall rating; details link	
TC-UC29-01	View providers list	List providers with venue counts and contact links	Providers exist	Open providers page	—	List shows provider name; number of venues; rating; contact/view links	List shows provider name; number of venues; rating; contact/view links	Pass
TC-UC30-01	Filter organizers by criteria	Filter by city; rating; experience with reset	Organizers page open	Apply city and rating filters	city:Aleppo; rating_min:4	Filtered organizers displayed; reset works	Filtered organizers displayed; reset works	Pass
TC-UC31-01	Filter providers by criteria	Filter by city; rating; experience with reset	Providers page open	Apply experience filter	experience_min:3	Filtered providers displayed; reset works	Filtered providers displayed; reset works	Pass
TC-UC32-01	Upvote content once per user	Record upvote; prevent duplicates; log voter and timestamp	User logged in	Click Upvote on content	content_id:321	UI count increments immediately; duplicate prevented; vote logged	UI count increments immediately; duplicate prevented; vote logged	Pass
TC-UC33-01	Downvote content once per user	Record downvote; prevent duplicates; log voter and timestamp	User logged in	Click Downvote on content	content_id:321	UI count updates immediately; duplicate prevented; vote logged	UI count updates immediately; duplicate prevented; vote logged	pass
TC-UC34-01	View recent activities timeline	Show user's activity timeline with filters and links	User has recent activities	Open activities page; apply filter	filter:bookings	Timeline shows activities with timestamps and links; filters work	Timeline shows activities with timestamps and links; filters work	Pass
TC-UC35-01	Display interactive map with pins	Show event/venue pins with info windows and filtering	Location data available	Open map; zoom; click pin	—	Pins displayed; info windows show details; clustering and filters supported	Pins displayed; info windows show details; clustering and filters supported	Pass
TC-UC36-01	Browse provider venues list	Provider dashboard lists venues with search and filters	Provider account with venues	Login as provider; open dashboard	provider_id:77	Provider sees venue cards with search; filters; edit links	Provider sees venue cards with search; filters; edit links	Pass
TC-UC37-01	View archived venues and restore	List archived venues with metadata and restore actions	Provider has archived venues	Open archived tab; select Restore	venue_id:400	Archive date/reason shown; restore works; action logged	Archive date/reason shown; restore works; action logged	pass
TC-UC38-01	Provider adds new venue successfully	Create venue with required fields and media handling	Add venue page open	Fill required fields; upload images; save	name:"Hall A"; location:Homs; images:3	Validation passes; venue saved; media processed; appears in provider dashboard	Validation passes; venue saved; media processed; appears in provider dashboard	Pass
TC-UC39-01	Edit venue with schedule validation	Edit venue data with conflict checks and audit trail	Venue exists and editable	Edit fields; save	venue_id:123; new_schedule:...	Conflicts validated; changes saved; public listing	Conflicts validated; changes saved; public listing	Pass

						updated; audit recorded	listing updated; audit recorded	
TC-UC40-01	Prevent deletion if active bookings exist	Block deletion when constraints exist and log attempt	Venue has active bookings	Attempt to delete venue	venue_id:123	Deletion blocked with explanatory message; attempt logged	Deletion blocked with explanatory message; attempt logged	Pass
TC-UC41-01	Archive venue hides from public	Archive venue and record metadata	Venue exists	Archive venue; confirm	venue_id:123	Status set to Archived; hidden from public; archive metadata logged	Status set to Archived; hidden from public; archive metadata logged	Pass
TC-UC42-01	Unarchive venue restores listing	Restore archived venue and reindex	Venue is archived	Unarchive venue; confirm	venue_id:123	Venue restored; reindexed; restoration logged	Venue restored; reindexed; restoration logged	Pass
TC-UC43-01	Provider views bookings with actions	Provider sees bookings with management actions	Provider has bookings	Open provider bookings page; apply filters	provider_id:77	Bookings listed with accept/reject/detail actions	Bookings listed with accept/reject/detail actions	Pass
TC-UC44-01	Provider accepts pending booking	Accept booking after availability recheck and notify client	Booking status Pending	Click Accept; confirm	booking_id:888	Availability rechecked; status set to Accepted; resources reserved; client notified	Availability rechecked; status set to Accepted; resources reserved; client notified	Pass
TC-UC45-01	Provider rejects booking with reason	Reject booking, release resources, notify client and log	Booking status Pending	Click Reject; enter reason; confirm	booking_id:889; reason:"Unavailable"	Status set to Rejected; resources released; client notified; reason logged	Status set to Rejected; resources released; client notified; reason logged	pass
TC-UC46-01	Provider cancels accepted booking	Cancel accepted booking, handle refunds, notify client	Booking Accepted/Confirmed	Click Cancel; confirm; process refunds if applicable	booking_id:890	Status set to Cancelled; refunds processed if applicable; client notified; action logged	Status set to Cancelled; refunds processed if applicable; client notified; action logged	Pass
TC-UC47-01	Browse events dashboard (organizer)	Organizer sees events with statuses and management actions	Organizer account with events	Login as organizer; open events dashboard	organizer_id:55	Events listed with Active/Draft/Archived statuses and management actions	Events listed with Active/Draft/Archived statuses and management actions	Pass
TC-UC48-01	View archived events and restore	List archived events with metadata and restore options	Organizer has archived events	Open archived tab; select Restore	event_id:300	Archive metadata shown; restore works; action logged	Archive metadata shown; restore works; action logged	Pass
TC-UC49-01	Create event with venue availability check	Create event with venue checks and draft/publish options	Create event page open; venue available	Fill event details; check availability; save or publish	title:"Dev Meetup"; venue_id:123; dates:2026-06-10	Availability confirmed; event created; can save as draft or publish;	Availability confirmed; event created; can save as draft or publish;	Pass

						organizer notified	organizer notified	
TC-UC50-01	Edit event and notify attendees if needed	Edit event and validate impacts on tickets/schedules	Published event with registrants	Edit dates/sessions; save	event_id:300; new_date:2026-06-12	System validates impacts; changes saved; affected attendees notified	System validates impacts; changes saved; affected attendees notified	Pass
TC-UC51-01	Prevent deletion with unresolved refunds	Block deletion if refunds/disputes exist and notify organizer	Event has pending refunds/disputes	Attempt to delete event	event_id:300	Deletion blocked; reason logged; organizer notified	Deletion blocked; reason logged; organizer notified	Pass
TC-UC52-01	Archive event hides from public	Archive event and record metadata	Event exists	Archive event; confirm	event_id:300	Status set to Archived; hidden from public; archive metadata logged	Status set to Archived; hidden from public; archive metadata logged	Pass
TC-UC53-01	Unarchive event restores listing	Restore archived event and reindex	Event is archived	Unarchive event; confirm	event_id:300	Event restored; reindexed; restoration logged	Event restored; reindexed; restoration logged	Pass
TC-UC54-01	Organizer views registrations with actions	View registrations with filters and quick actions	Organizer has registrations	Open registrations page; apply filters	organizer_id:55	Registrations listed with accept/reject actions and stats	Registrations listed with accept/reject actions and stats	Pass
TC-UC55-01	Organizer accepts pending registration	Accept registration, update ticket counts, notify registrant	Registration Pending	Click Accept; confirm	registration_id:700	Status set to Accepted; ticket counts updated; registrant notified	Status set to Accepted; ticket counts updated; registrant notified	Pass
TC-UC56-01	Organizer rejects registration with reason	Reject registration, release tickets, notify and log	Registration Pending	Click Reject; enter reason; confirm	registration_id:701; reason:"Incomplete"	Status set to Rejected; tickets released; registrant notified; reason logged	Status set to Rejected; tickets released; registrant notified; reason logged	Pass
TC-UC57-01	Cancel registration and handle refund	Cancel registration with refund handling per policy	Registration Accepted/Confirmed	Click Cancel; confirm; apply refund policy	registration_id:702	Status set to Cancelled; refunds processed if applicable; availability updated; action logged	Status set to Cancelled; refunds processed if applicable; availability updated; action logged	Pass
TC-UC58-01	Export registrations as XLSX	Generate XLSX export for selected criteria and log export	Registrations exist in range	Select date range; click Export	date_from:2026-01-01; date_to:2026-01-31	XLSX file generated and available for download; export logged	XLSX file generated and available for download; export logged	Pass
TC-UC59-01	Organizer views booking details	Organizer views booking details with management actions	Organizer linked to booking	Open booking detail	booking_id:888	Booking details displayed with management actions	Booking details displayed with management actions	Pass

TC-UC60-01	Organizer creates booking on behalf of client	Create booking for client with availability validation and confirmation	Organizer logged in; venue available	Fill booking form for client; submit	client_name:"Ali"; venue_id:123; date:2026-04-05	Availability validated; booking created; payment processed if needed; confirmation sent to client	Availability validated; booking created; payment processed if needed; confirmation sent to client	pass
TC-UC61-01	Organizer edits booking with conflict check	Edit booking and validate conflicts; notify client and audit	Booking exists	Edit allowed fields; save	booking_id:888; new_time:19:00	Conflicts checked; changes saved; client notified; audit recorded	Conflicts checked; changes saved; client notified; audit recorded	Pass
TC-UC62-01	Organizer deletes booking with refund handling	Delete/cancel booking with refund processing and notifications	Booking exists and cancellable	Click Delete; confirm; process refunds if applicable	booking_id:888	Booking deleted or cancelled; refunds processed if applicable; client notified; action logged	Booking deleted or cancelled; refunds processed if applicable; client notified; action logged	Pass

5. Updating the Requirements Traceability Matrix

Requirement ID	Analysis	Design	Application Interface	Code	Test Case ID
REQ-A01	Login	Login - activity diagram ; Login - sequence diagram	user interfaces	Login	initial test cases
REQ-A02	Logout	Logout - activity diagram ; Logout - sequence diagram	user interfaces	Logout	initial test cases
REQ-A03	Register (User Registration)	Register (User Registration) - activity diagram ; Register (User Registration) - sequence diagram	user interfaces	Register (User Registration)	initial test cases
REQ-A04	View Venues	View Venues - activity diagram ; View Venues - sequence diagram	user interfaces	View Venues	initial test cases
REQ-A05	View Venue Details	View Venue Details - activity diagram ; View Venue Details - sequence diagram	user interfaces	View Venue Details	initial test cases
REQ-A06	View Bookings	View Bookings - activity diagram ; View Bookings - sequence diagram	user interfaces	View Bookings	initial test cases

REQ-A07	Add Booking	Add Booking - activity diagram ; Add Booking - sequence diagram	user interfaces	Add Booking	initial test cases
REQ-A08	Edit Booking	Edit Booking - activity diagram ; Edit Booking - sequence diagram	user interfaces	Edit Booking	initial test cases
REQ-A09	Delete Booking	Delete Booking - activity diagram ; Delete Booking - sequence diagram	user interfaces	Delete Booking	initial test cases
REQ-A10	View Events	View Events - activity diagram ; View Events - sequence diagram	user interfaces	View Events	initial test cases
REQ-A11	View Event Details	View Event Details - activity diagram ; View Event Details - sequence diagram	user interfaces	View Event Details	initial test cases
REQ-A12	Unified Search for Venues	Unified Search for Venues - activity diagram ; Unified Search for Venues - sequence diagram	user interfaces	Unified Search for Venues	initial test cases
REQ-A13	Unified Search for Events	Unified Search for Events - activity diagram ; Unified Search for Events - sequence diagram	user interfaces	Unified Search for Events	initial test cases
REQ-A14	Filter Venues by Specific Criteria	Filter Venues by Specific Criteria - activity diagram ; Filter Venues by Specific Criteria - sequence diagram	user interfaces	Filter Venues by Specific Criteria	initial test cases
REQ-A15	Filter Events by Specific Criteria	Filter Events by Specific Criteria - activity diagram ; Filter Events by Specific Criteria - sequence diagram	user interfaces	Filter Events by Specific Criteria	initial test cases
REQ-A16	View Registrations	View Registrations - activity diagram ; View Registrations - sequence diagram	user interfaces	View Registrations	initial test cases
REQ-A17	Add Registration	Add Registration - activity diagram ; Add Registration - sequence diagram	user interfaces	Add Registration	initial test cases

REQ-A18	Edit Registration	Edit Registration - activity diagram ; Edit Registration - sequence diagram	user interfaces	Edit Registration	initial test cases
REQ-A19	Delete Registration	Delete Registration - activity diagram ; Delete Registration - sequence diagram	user interfaces	Delete Registration	initial test cases
REQ-A20	View Venue Ratings	View Venue Ratings - activity diagram ; View Venue Ratings - sequence diagram	user interfaces	View Venue Ratings	initial test cases
REQ-A21	Add Venue Rating	Add Venue Rating - activity diagram ; Add Venue Rating - sequence diagram	user interfaces	Add Venue Rating	initial test cases
REQ-A22	Edit Venue Rating	Edit Venue Rating - activity diagram ; Edit Venue Rating - sequence diagram	user interfaces	Edit Venue Rating	initial test cases
REQ-A23	Delete Venue Rating	Delete Venue Rating - activity diagram ; Delete Venue Rating - sequence diagram	user interfaces	Delete Venue Rating	initial test cases
REQ-A24	View Event Ratings	View Event Ratings - activity diagram ; View Event Ratings - sequence diagram	user interfaces	View Event Ratings	initial test cases
REQ-A25	Add Event Rating	Add Event Rating - activity diagram ; Add Event Rating - sequence diagram	user interfaces	Add Event Rating	initial test cases
REQ-A26	Edit Event Rating	Edit Event Rating - activity diagram ; Edit Event Rating - sequence diagram	user interfaces	Edit Event Rating	initial test cases
REQ-A27	Delete Event Rating	Delete Event Rating - activity diagram ; Delete Event Rating - sequence diagram	user interfaces	Delete Event Rating	initial test cases
REQ-A28	View Organizers	View Organizers - activity diagram ; View Organizers - sequence diagram	user interfaces	View Organizers	initial test cases
REQ-A29	View Providers	View Providers - activity diagram ; View Providers - sequence diagram	user interfaces	View Providers	initial test cases
REQ-A30	Filter Organizers	Filter Organizers by Specific Criteria - activity diagram ;	user interfaces	Filter Organizers	initial test cases

	by Specific Criteria	Filter Organizers by Specific Criteria - sequence diagram		by Specific Criteria	
REQ-A31	Filter Providers by Specific Criteria	Filter Providers by Specific Criteria - activity diagram ; Filter Providers by Specific Criteria - sequence diagram	user interfaces	Filter Providers by Specific Criteria	initial test cases
REQ-A32	Upvote	Upvote - activity diagram ; Upvote - sequence diagram	user interfaces	Upvote	initial test cases
REQ-A33	Downvote	Downvote - activity diagram ; Downvote - sequence diagram	user interfaces	Downvote	initial test cases
REQ-A34	View Recent Activities	View Recent Activities - activity diagram ; View Recent Activities - sequence diagram	user interfaces	View Recent Activities	initial test cases
REQ-A35	Display Map of Event and Venue Locations	Display Map of Event and Venue Locations - activity diagram ; Display Map of Event and Venue Locations - sequence diagram	user interfaces	Display Map of Event and Venue Locations	initial test cases
REQ-A36	Browse Venues (Provider)	Browse Venues (Provider) - activity diagram ; Browse Venues (Provider) - sequence diagram	user interfaces	Browse Venues (Provider)	initial test cases
REQ-A37	Browse Archived Venues	Browse Archived Venues - activity diagram ; Browse Archived Venues - sequence diagram	user interfaces	Browse Archived Venues	initial test cases
REQ-A38	Add Venue	Add Venue - activity diagram ; Add Venue - sequence diagram	user interfaces	Add Venue	initial test cases
REQ-A39	Edit Venue	Edit Venue - activity diagram ; Edit Venue - sequence diagram	user interfaces	Edit Venue	initial test cases
REQ-A40	Delete Venue	Delete Venue - activity diagram ; Delete Venue - sequence diagram	user interfaces	Delete Venue	initial test cases
REQ-A41	Archive Venue	Archive Venue - activity diagram ; Archive Venue - sequence diagram	user interfaces	Archive Venue	initial test cases

REQ-A42	Unarchive Venue	Unarchive Venue - activity diagram ; Unarchive Venue - sequence diagram	user interfaces	Unarchive Venue	initial test cases
REQ-A43	Browse Bookings (Provider)	Browse Bookings (Provider) - activity diagram ; Browse Bookings (Provider) - sequence diagram	user interfaces	Browse Bookings (Provider)	initial test cases
REQ-A44	Accept Booking (Provider)	Accept Booking (Provider) - activity diagram ; Accept Booking (Provider) - sequence diagram	user interfaces	Accept Booking (Provider)	initial test cases
REQ-A45	Reject Booking (Provider)	Reject Booking (Provider) - activity diagram ; Reject Booking (Provider) - sequence diagram	user interfaces	Reject Booking (Provider)	initial test cases
REQ-A46	Cancel Booking (Provider)	Cancel Booking (Provider) - activity diagram ; Cancel Booking (Provider) - sequence diagram	user interfaces	Cancel Booking (Provider)	initial test cases
REQ-A47	Browse Events (Organizer)	Browse Events (Organizer) - activity diagram ; Browse Events (Organizer) - sequence diagram	user interfaces	Browse Events (Organizer)	initial test cases
REQ-A48	Browse Archived Events	Browse Archived Events - activity diagram ; Browse Archived Events - sequence diagram	user interfaces	Browse Archived Events	initial test cases
REQ-A49	Create Event	Create Event - activity diagram ; Create Event - sequence diagram	user interfaces	Create Event	initial test cases
REQ-A50	Edit Event	Edit Event - activity diagram ; Edit Event - sequence diagram	user interfaces	Edit Event	initial test cases
REQ-A51	Delete Event	Delete Event - activity diagram ; Delete Event - sequence diagram	user interfaces	Delete Event	initial test cases
REQ-A52	Archive Event	Archive Event - activity diagram ; Archive Event - sequence diagram	user interfaces	Archive Event	initial test cases
REQ-A53	Unarchive Event	Unarchive Event - activity diagram ;	user interfaces	Unarchive Event	initial test cases

		Unarchive Event - sequence diagram			
REQ-A54	Browse Registrations (Organizer)	Browse Registrations (Organizer) - activity diagram ; Browse Registrations (Organizer) - sequence diagram	user interfaces	Browse Registrations (Organizer)	initial test cases
REQ-A55	Accept Registration	Accept Registration - activity diagram ; Accept Registration - sequence diagram	user interfaces	Accept Registration	initial test cases
REQ-A56	Reject Registration	Reject Registration - activity diagram ; Reject Registration - sequence diagram	user interfaces	Reject Registration	initial test cases
REQ-A57	Cancel Registration	Cancel Registration - activity diagram ; Cancel Registration - sequence diagram	user interfaces	Cancel Registration	initial test cases
REQ-A58	Export Registrations as Excel file	Export Registrations as Excel file - activity diagram ; Export Registrations as Excel file - sequence diagram	user interfaces	Export Registrations as Excel file	initial test cases
REQ-A59	View Booking (Organizer)	View Booking (Organizer) - activity diagram ; View Booking (Organizer) - sequence diagram	user interfaces	View Booking (Organizer)	initial test cases
REQ-A60	Add Booking (Organizer)	Add Booking (Organizer) - activity diagram ; Add Booking (Organizer) - sequence diagram	user interfaces	Add Booking (Organizer)	initial test cases
REQ-A61	Edit Booking (Organizer)	Edit Booking (Organizer) - activity diagram ; Edit Booking (Organizer) - sequence diagram	user interfaces	Edit Booking (Organizer)	initial test cases
REQ-A62	Delete Booking (Organizer)	Delete Booking (Organizer) - activity diagram ; Delete Booking (Organizer) - sequence diagram	user interfaces	Delete Booking (Organizer)	initial test cases
REQ-A63	Export Registrations as Excel file	Export Registrations as Excel file - sequence diagram ; Export	user interfaces	Export Registrations as Excel file	initial test cases

		<u>Registrations as Excel file - activity diagram</u>			
REQ-A64	<u>View e-ticket</u>	<u>View e-ticket - sequence diagram</u> ; <u>View e-ticket - activity diagram</u>	<u>user interfaces</u>	<u>View e-ticket</u>	<u>initial test cases</u>
REQ-A65	<u>Create e-ticket</u>	<u>Create e-ticket - sequence diagram</u> ; <u>Create e-ticket - activity diagram</u>	<u>user interfaces</u>	<u>Create e-ticket</u>	<u>initial test cases</u>
REQ-A66	<u>Export e-tickets list as PDF file</u>	<u>Export e-tickets list as PDF file - sequence diagram</u> ; <u>Export e-tickets list as PDF file - activity diagram</u>	<u>user interfaces</u>	<u>Export e-tickets list as PDF file</u>	<u>initial test cases</u>
REQ-A67	<u>Display an interactive calendar</u>	<u>Display an interactive calendar - sequence diagram</u> ; <u>Display an interactive calendar - activity diagram</u>	<u>user interfaces</u>	<u>Display an interactive calendar</u>	<u>initial test cases</u>

6. Conclusion

In summary, testing the event and booking system documented tools and environments, used miniature models to simulate core scenarios, and revealed strengths and areas for improvement (load performance, batch processing, and UX for filters/calendar). Test results updated the requirements traceability matrix, underscoring the need for realistic test environments, robust logging/monitoring, and peak-load strategies. Recommended next steps are performance tuning, stronger notification/delivery mechanisms, and adding integration and acceptance tests before production to ensure a reliable deployment roadmap based on measurable data.

Chapter Seven

Conclusions and future prospects

1.introduction

The **Event Management and Organization** project aims to provide an integrated online platform that facilitates event organization through dedicated interfaces for clients, venue providers, and event organizers, with booking, management, and basic reporting features. During the project period, a functional version of the application was developed including the three user interfaces, a preliminary booking flow, an administrative dashboard, and core REST APIs. These outcomes meet the primary objective of digitizing event organization and improving stakeholder experience. The system is ready for final testing and delivery with basic documentation and an initial test

2.Key Achievements — System Framework Considered

- **Core interface development:** Implemented dedicated interfaces for the Client, Venue Provider, and Event Organizer as a **React SPA**, including authentication and account management.
- **Client-server model:** The system was designed using a **client-server** architecture where the React frontend communicates with the backend via a **Django REST API** using **JSON over HTTPS**, ensuring separation of the UI from backend logic.
- **Booking and management system:** Implemented a preliminary booking workflow with availability checks and booking management by venue providers, with integration performed through secure API calls.
- **Backend and APIs:** Designed a functional database schema and developed core REST APIs for CRUD operations and integration between interfaces, with a clear separation of external services to simplify maintenance and iterative development.
- **Documentation and testing:** Prepared a brief user guide, unit and integration test cases, and executed initial tests to verify core functionality and the integrity of communications between the SPA and the APIs.

3.Lessons Learned (Adapted to the Project)

- **Incremental documentation as an operational necessity:** Adopting continuous documentation for each design and development phase, technical decisions, and operational procedures made task handover easier during absences and reduced single-person dependency during the implementation of the event management platform.
- **Strict scope control:** Defining clear requirements and a formal change-management process enabled us to defer or reject non-essential requests, preserving the schedule and delivery quality.
- **Careful selection of third-party integrations:** Before relying on any external service, we evaluated alternatives and created mock interfaces for testing, which reduced the risk of service outages or unexpected operating costs.
- **Regular communication and effective coordination:** Weekly review meetings and early supervisor reviews improved early defect detection, accelerated problem resolution, and strengthened collaboration between frontend and backend developers.

4. Constraints and Issues Encountered (Adapted to the Project)

- **Scope and implementation limits:** Some items were excluded from the final delivery, such as on-site logistics and equipment management, to keep the scope focused on a digital solution and ensure the platform could be developed, tested, and deployed within available resources.
- **Technical and operational challenges:** We faced issues related to reliance on third-party services (e.g., payment gateways or test hosting providers), incomplete test coverage due to limited time and testing environments, and budget constraints that restricted broader testing and advanced integrations.
- **Impact on deliverables and schedule:** These constraints led to postponing certain secondary features to later phases, narrowing the initial scope of integrations, and allocating buffer time for integration testing. As a result, the team concentrated on delivering a stable, extensible core rather than a fully comprehensive solution at launch.

5. Future prospects and recommendations

- **Expand integrations and operational flexibility:** Integrate multiple payment gateways, connect the system with supplier management systems and external platforms via reliable APIs, and provide a flexible integration layer that allows adding new providers without major architectural changes.
- **Migrate to a robust server database:** Adopt **PostgreSQL** as the production database instead of SQLite to improve performance, concurrency, and reliability. This includes planning the migration, setting up staging environments for testing the migration, using appropriate migration tools, and configuring secure connections and credentials.
- **Enhance security and operational reliability:** Perform advanced security testing (penetration tests, vulnerability scans), apply end-to-end data encryption in transit and at rest, and implement a mature CI/CD pipeline with stable **staging** and **production** environments and monitored deployment processes.
- **Add intelligent and analytical features:** Incorporate usage analytics to track user behavior, develop a recommendation engine for venues based on preferences and booking history, and add rating and review mechanisms to improve user experience and enable data-driven decisions.
- **Sustainable business and operational plan:** Prepare a study of revenue models (subscriptions, commissions, premium services), build strategic partnerships with venue providers and service vendors, and establish a post-launch support and maintenance plan with clear service levels and a regular update schedule.

6. Conclusion

The project delivered a functional core of an **Event Management** platform that meets the primary objectives of digitizing booking and management processes and providing tailored interfaces for key users. The system was designed using a client-server model, where a **React SPA** communicates with a **Django REST API** over JSON and HTTPS, enabling separation of the UI from backend logic and facilitating maintainable, iterative development.

Despite time and budget constraints and the exclusion of some on-site aspects, the team succeeded in building a stable and extensible foundation, with incremental documentation, initial testing, and a basic development pipeline. The planned migration to **PostgreSQL**, together with the proposed security and CI/CD improvements, will enhance reliability and performance when moving to production.

In the next phase, it is recommended to prioritize executing the PostgreSQL migration plan, expanding core integrations such as payment gateways and supplier management systems, and strengthening security testing and operational monitoring. By focusing on these priorities, the platform can be transformed into a production-ready, commercially viable solution that serves event organizers, venue providers, and clients with greater efficiency and stability.

2.Conclusions

Through the analysis and implementation of the project, a set of main conclusions were reached that reflect the success of the system and the achievement of its objectives.:

1. Achieving Objectives: The electronic event management system has proven effective in facilitating the process of finding suitable venues and organizing events efficiently. The system has been able to meet the diverse needs of users by providing an easy-to-use interface and the ability to customize services to suit each user's requirements.
2. Improving the user experience: The user experience was significantly improved through the design of an intuitive and user-friendly user interface. User feedback was positive, noting the ease of accessing the information and services required quickly and accurately.
3. System Integration: The system demonstrated a high ability to integrate with other external systems, expanding its capabilities and enabling additional services. Users were able to benefit from a wide range of services and data provided by external systems, increasing the system's value to them.
4. Budget Management: The system provided powerful budget management and expense tracking tools, helping users control event costs and stay within budget. These tools were particularly useful for users who needed to organize large, complex events.
- .5 User Satisfaction: User feedback was generally positive, with users expressing satisfaction with the system and the services it provides. This satisfaction reflects the significant success achieved in developing the system and meeting user needs.

3.Future prospects

To keep up with the ever-changing events market and achieve user satisfaction, there are some future directions that can be considered to improve and develop the system.:

1. Add new features: Can Adding new features such as group booking support, managing large and complex events, and providing more detailed analytical reports will help attract more users and meet their diverse needs.

2. Improving security: Enhancing system security and protecting user data by implementing advanced security standards and using modern encryption technologies. Security is a vital component of ensuring user confidence in the system.

3. Multiple Language Support: Expanding the system to support multiple languages to meet the needs of users of different nationalities and cultures. This will expand the potential user base and promote the system's global reach.

4. Social media integration: Enable users to share event details and invitations via social media to increase exposure and reach a wider audience. This will help enhance communication between users and encourage them to use the system more.

5. Mobile Application Development: Develop mobile applications for iOS and Android systems to provide easy and seamless access to the system through smartphones and tablets. This will increase the system's usability and provide a convenient experience for users on the go.

6. Data Analysis and Performance Improvement: Using advanced analytics techniques to analyze data collected from system usage and detect patterns and trends. These analyses can help improve system performance and provide personalized services to each user based on their data and preferences.

7. Expanding the user base: Determine effective marketing strategies to increase the user base and encourage more individuals and organizations to use the system to organize their events.

4. Conclusion

This chapter reviews the key findings that reflect the system's success and achievement of its objectives. It also discusses the project's future prospects and ways to improve and develop it. By adhering to these future directions, the system can remain flexible and able to meet users' growing needs and provide an excellent and integrated user experience. Achieving these goals will contribute to the project's continued success and make it more attractive and effective for future users.

the reviewer

- **Python (official docs)** — <https://www.python.org/doc/>
- **Django (official docs)** — <https://docs.djangoproject.com/>
- **Django REST framework (official docs)** — <https://www.django-rest-framework.org/>
- **React (official docs)** — <https://react.dev/>
- **MDN Web Docs (HTML, CSS, JavaScript)** — <https://developer.mozilla.org/>
- **Can I use (browser compatibility)** — <https://caniuse.com/>
- **SQLite (official docs)** — <https://www.sqlite.org/docs.html>
- **Visual Studio Code (official docs & downloads)** — <https://code.visualstudio.com/docs>
- **VS Code Marketplace (extensions)** —
<https://marketplace.visualstudio.com/vscode> ([marketplace.visualstudio.com in Bing](#))
- **GitHub Docs** — <https://docs.github.com/>
- **Pro Git (free book)** — <https://git-scm.com/book/en/v2> ([git-scm.com in Bing](#))
- **OWASP (security guidance)** — <https://owasp.org/www-project-top-ten/> ([owasp.org in Bing](#))
- **Django + React tutorial (example guides and boilerplates)** —
<https://www.fullstackpython.com/django-react.html> ([fullstackpython.com in Bing](#))
- **Django REST + React example projects on GitHub (searchable starter templates)** —
<https://github.com/search?q=django+react+boilerplate> ([github.com in Bing](#))
- **Using SQLite with Django (configuration guide)** —
<https://docs.djangoproject.com/en/stable/ref/databases/#sqlite-notes> ([docs.djangoproject.com in Bing](#))