Software Requirements and Design Document

for

Grand Soiree (event management system)

**Prepared by**

**Muhammad Nehal 21i-2998  
Zain Ijaz 21i-0753**

**SCOPE**

**User interactions ie:Customer and Organizer**

**DATE**

**13*th* May 2024**

**Table of Contents**

**Table of Contents ii**

**1.** **Introduction 1**

1.1 Purpose 1

1.2 Product Scope 1

1.3 Title 1

1.4 Objectives 1

1.5 Problem Statement 1

**2.** **Overall Description 1**

2.1 Product Perspective 1

2.2 Product Functions 2

2.3 List of Use Cases 2

2.4 Extended Use Cases 2

2.5 Use Case Diagram 2

**3.** **Other Nonfunctional Requirements 2**

3.1 Performance Requirements 2

3.2 Safety Requirements 2

3.3 Security Requirements 2

3.4 Software Quality Attributes 2

3.5 Business Rules 3

3.6 Operating Environment 3

3.7 User Interfaces 3

**4.** **Domain Model 3**

**5.** **System Sequence Diagram 3**

**6.** **Sequence Diagram 3**

**7.** **Class Diagram 4**

**8.** **Package Diagram 4**

**9.** **Deployment Diagram 4**

# Introduction

## Purpose

The purpose of this project is to develop an efficient and user-friendly event management system that caters to the needs of both organizers and customers. By creating a platform where organizers can effortlessly create and manage events while customers can easily discover and purchase tickets, the system aims to streamline the event planning and attendance process. Additionally, by incorporating features such as event feedback mechanisms, the system seeks to enhance the overall experience for both organizers and attendees, ultimately fostering greater engagement and satisfaction within the event management ecosystem.

## Product Scope

The event management system encompasses a user-friendly interface for both organizers and customers, facilitating event creation, ticket purchasing, and feedback submission. Organizers can seamlessly manage event attributes such as venue, tickets, and themes, while customers can easily browse events, purchase tickets, and provide feedback. The system ensures efficient communication and interaction between organizers and attendees, enhancing the overall event experience.

## Title :

Grand Soiree: Revolutionizing Event Management with Seamless Interaction

## Objectives

1. Develop a user-friendly event management system that caters to the needs of both organizers and customers.

2. Enable organizers to efficiently create and manage events, including setting event attributes such as venue, tickets, time, and theme.

3. Provide customers with an intuitive platform to browse events, purchase tickets, and view event details.

4. Implement features for organizers to monitor and manage events they have created, including viewing customer feedback.

5. Incorporate mechanisms for customers to leave feedback for organizers, contributing to the improvement of future events.

6. Ensure seamless interaction between organizers and customers throughout the event management process.

7. Enhance overall user satisfaction by delivering a reliable and responsive event management solution.

8. Foster engagement within the event management ecosystem by facilitating efficient communication and interaction between all stakeholders.

## Problem Statement

The current landscape of event management often lacks a streamlined and user-friendly solution that caters to the diverse needs of both organizers and customers. Existing platforms may suffer from cumbersome event creation processes, limited functionality for attendees, and inadequate feedback mechanisms for organizers. This fragmentation within the event management ecosystem leads to inefficiencies, hindering the seamless organization and participation in events. Therefore, there is a pressing need for an integrated event management system that bridges the gap between organizers and attendees, offering robust features for event creation, ticket purchasing, and feedback collection to enhance the overall event experience.

# Overall Description

## Product Perspective

"Grand Soiree" is a standalone event management system, distinct from existing solutions. It aims to streamline event organization and attendance for both organizers and customers. While it's self-contained, it may integrate with external systems for functionalities like payment processing or social media. A diagram will illustrate its components, subsystem connections, and external interfaces.

## Product Functions

1. **Event Organizer Functions:**

* - Create events with customizable attributes (venue, tickets, time, theme).
* - Manage created events (view, edit, delete).
* - View customer feedback for events.

**2. Customer Functions:**

* - Browse events created by organizers.
* - Select and purchase event tickets.
* - View event details and purchased tickets.
* - Provide feedback for attended events.

## List of Use Cases

The implemented use cases in the project are:

1. Create Event

2. Manage Event

3. Buy Tickets

4. Manage Tickets

5. Browse Events

6. Give Feedback

7. Browse Venues

## Extended Use Cases

**1. Create Event:**

**- Scope:** Grand Soiree

**- Level:** User Level

**- Primary Actor**: Organizer

**- Stakeholders and Interests:** Organizer, Customers, Logistics Staff, System

**- Pre-Conditions:**

- Organizer has successfully logged in.

- All Organizer information is available for authenticity.

**- Post-Conditions:**

- The new event has been added to the organizer’s event tab.

- Customers can see the event to buy tickets.

**- Main Success Scenario:**

- Organizer clicks on create event button.

- Organizer provides all the information related to the event including name, type, date, time, ticket price, capacity, and other special information related to the event.

- Organizer then browses and selects a venue for the event.

- Organizer pays the cost for the event.

- Clicks on submit.

- System checks all the details and authenticates everything.

- System adds the event on the Grand Soiree.

**- Extensions:**

- Payment Failed due to which event not successfully added: User changes payment method or tries again.

- Organizer couldn’t find a venue hence has to change the date.

- Information not complete: User has to provide information again.

**2. Manage Event:**

**- Scope:** Grand Soiree

**- Level:** User Level

**- Primary Actor:** Organizer

**- Stakeholders and Interests:** Organizer, Customers, Logistics Staff, System

**- Pre-Conditions:**

- Organizer has successfully logged in.

- All Organizer information is available for authenticity.

- Organizer already has events on Grand Soiree that can be managed.

**- Post-Conditions:**

- The changes that the organizer wants to implement on the event are successfully made.

- The Database has successfully been updated.

**- Main Success Scenario:**

- Organizer clicks on manage event button.

- System shows the organizer all the events under the name of organizer with all the updates and information related to the event.

- Organizer then browses and clicks on the event they want to edit.

- Organizer edits the event like update, delete, and postpone.

- Organizer Clicks on update.

- System updates the event on the Grand Soiree.

**- Extensions:**

- System does not update due to poor internet connection; the user has to try again later.

- System makes an update that is not possible, the system warns the user.

**3. Buy Tickets:**

**- Scope:** Grand Soiree

**- Level:** User Level

**- Primary Actor:** Customer

**- Stakeholders and Interests:** Organizer, Customer, Logistics Staff, System

**- Pre-Conditions:**

- User has successfully logged in.

- User has a running internet connection.

**- Post-Conditions:**

- User has successfully bought the ticket and the system is updated.

**- Main Success Scenario:**

- Customer clicks on the browse events button.

- System displays all the events taking place to the user with dates, ticket prizes, and venue.

- Customer clicks on the event they want to buy ticket for.

- Customer clicks on buy ticket.

- Customer provides the payment information and clicks on buy ticket.

- System checks all the details and authenticates everything.

- System adds the ticket to the user account and updates the event information for organizer.

**- Extensions:**

- Payment Failed due to which ticket is not successfully bought: User changes payment method or tries again.

- The event is sold out: User has to look for another event.

**4. Manage Tickets:**

**- Scope:** Grand Soiree

**- Level:** User Level

**- Primary Actor:** Customer

**- Stakeholders and Interests:** Organizer, Customers, Logistics Staff, System

**- Pre-Conditions:**

- Customer has successfully logged into the system.

- Customer has bought one or more tickets that he wants to manage.

**- Post-Conditions:**

- Customer has successfully used or returned their ticket depending on the requirement.

**- Main Success Scenario:**

- Customer clicked on manage tickets button.

- System displays all the tickets that the user has bought using the account.

- Customer either uses the ticket to get access to the event or to confirm the date, time, and venue. Customer can also ask for a refund till 5 days before the event.

- If asked for refund system checks for validity.

- System refunds the payment and updates the event information for organizer.

**- Extensions:**

- Refund fails as the date for refund has passed.

**5. Browse Events:**

**- Scope:** Grand Soiree

**- Level:** User Level

**- Primary Actor:** Customer

**- Stakeholders and Interests:** Organizer, Customers, Logistics Staff, System

**- Pre-Conditions:**

- Customer has logged in.

**- Post-Conditions:**

- The customer was shown all upcoming events in his area and he has closed the browsing window.

**- Main Success Scenario:**

- Customer clicks on the browse button.

- System opens a window with a list of all upcoming public events in the database.

- Customer clicks on one of the shown events.

- System shows more details about the chosen event like ticket price, activities, performers, available tickets, and a purchase tickets button.

- Customers either chooses to buy tickets or goes back to browsing.

- Customer goes back to the home screen.

**- Extensions:**

- There are no upcoming events to show.

**6. Give Feedback:**

**- Scope**: Grand Soiree

**- Level:** User Level

**- Primary Actor:** Customer

**- Stakeholders and Interests:** Organizer, Customers, Logistics Staff, System

**- Pre-Conditions:**

- User should’ve attended the Grand Soiree.

- User is logged in.

**- Post-Conditions:**

- The Feedback is successfully submitted.

- Both the organizer and Grand Soiree get the feedback.

**- Main Success Scenario:**

- After the event, the customer clicks on give feedback button.

- System displays all the events that the customer participated in.

- Customer gives written feedback and also gives a specific number of stars out of 5 after the event.

- System updates and sends the feedback to the organizer; this feedback can also be seen by other customers helping them to make a decision before buying a ticket for the organizer.

**- Extensions:**

- Customer has not attended any event hence has no event option to give feedback to.

**7. Browse Venues:**

**- Scope:** Grand Soiree

**- Level:** User Level

**- Primary Actor:** Customer

**- Stakeholders and Interests:** Organizer, Customers, Logistics Staff, System

**- Pre-Conditions:**

- Organizer has logged in.

- Organizer wants to arrange an event.

**- Post-Conditions:**

- Organizer has successfully found a venue that he/she wants to book for their event.

**- Main Success Scenario:**

- Organizer clicks on the browse venue button after creating an event.

- System opens a window with a list of all venues in the database.

- Organizer clicks on one of the shown venues that is suitable for them.

- System shows more details about the chosen venue like capacity, venue type, location, and dates and time on which the venue is available.

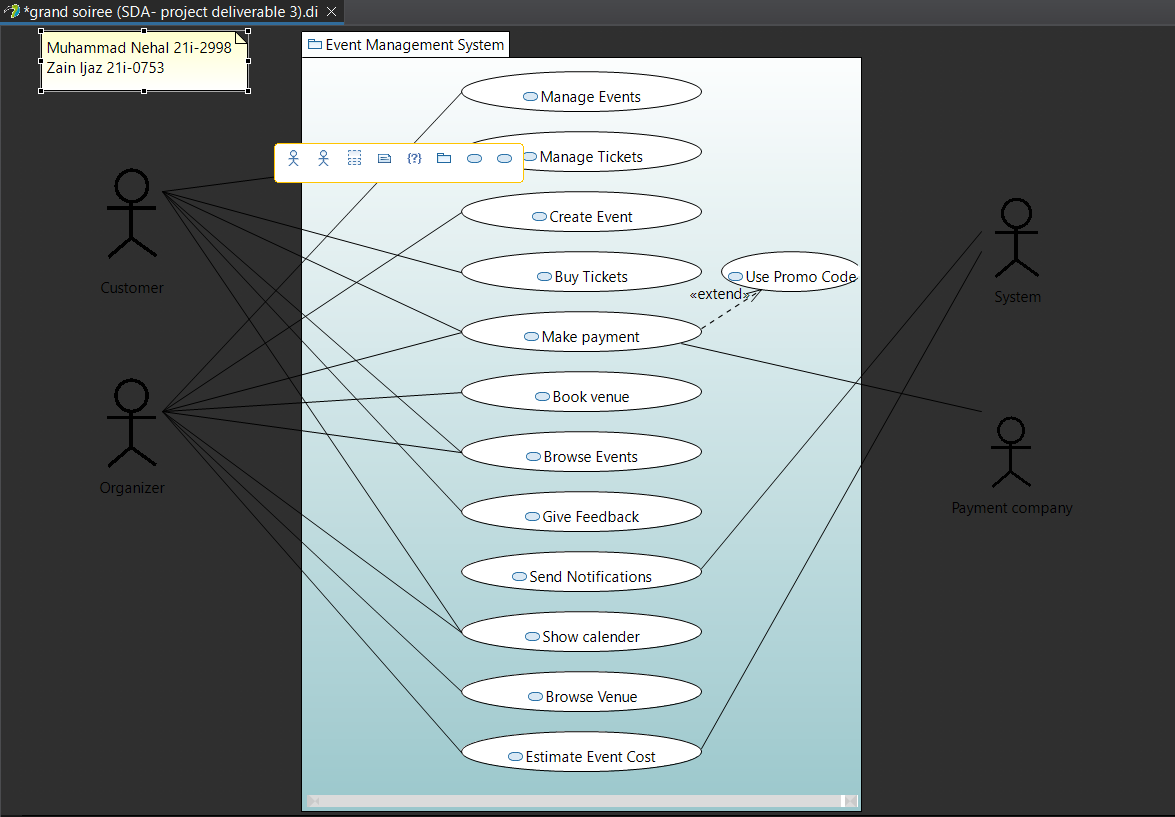
- Organizer either chooses to book a venue or goes back to browsing.

- Organizer goes back to the home screen.

**- Extensions:**

- Organizer is unable to find any suitable venue for them.

## Use Case Diagram



# Other Nonfunctional Requirements

## Performance Requirements

The performance requirements for the system are crucial for ensuring a seamless user experience. The system should respond to user actions within 2 seconds under normal load conditions to maintain smooth interaction. Key actions such as creating a new event, purchasing tickets, submitting feedback, and booking venues should be completed within specific time limits: event creation within 10 seconds, ticket purchase within 5 seconds, feedback submission within 10 seconds, and venue booking within 15 seconds. These requirements aim to optimize system responsiveness and efficiency, enhancing user satisfaction and encouraging continued usage of the platform.

## Safety Requirements

The safety of users and their data is paramount in the design of the system. To ensure this, the system implements stringent measures. It encrypts user data, including personal information and payment details, to prevent unauthorized access and safeguard against data breaches. Robust error handling mechanisms are in place to detect and address issues promptly, minimizing the risk of data loss or corruption. User authentication and authorization processes restrict access to sensitive features, enhancing overall security. Compliance with relevant regulations, such as GDPR and PCI DSS, further ensures data protection and user rights. Additionally, regular system backups are conducted to mitigate the impact of potential hardware failures or malicious attacks, guaranteeing the integrity and availability of critical data. These measures collectively safeguard users and their data, fostering trust and confidence in the system's safety and reliability.

## Security Requirements

The system prioritizes robust security measures to protect user data and ensure privacy. It implements secure user authentication, encrypted sensitive data, and enforces access control to prevent unauthorized access. Comprehensive audit trails monitor user activities, while regular security audits identify and address vulnerabilities. These measures collectively fortify the system's defenses, mitigating security risks and fostering user confidence in data protection.

## Software Quality Attributes

The success of the system hinges on its ability to deliver high-quality performance across various dimensions. Reliability is paramount, with a target uptime of at least 99%, ensuring users can consistently access the system without disruptions. Usability is equally crucial, aiming for an intuitive interface that scores at least 80% on usability testing scales, promoting user satisfaction and reducing training needs. Maintainability ensures the system remains adaptable and easy to enhance over time, with code adhering to standards and achieving a maintainability index of at least 80. Performance efficiency guarantees that the system can handle a minimum of 1000 concurrent users seamlessly. Security is a top priority, requiring regular assessments and achieving a security rating of at least 90% to protect user data. Testability ensures thorough validation, with at least 80% code coverage by automated tests and a testability score of at least 3.5, mitigating the risk of undiscovered defects. By prioritizing these quality attributes, the system aims to deliver a robust, user-friendly, and secure experience that meets both user expectations and development standards.

## Business Rules

- Only organizers can create and manage events.

- Customers can only purchase tickets for events created by organizers.

- Organizers can view and manage the events they have created.

- Customers can leave feedback for events they have attended.

- Payment must be successfully processed before tickets are issued.

- Events cannot be deleted once tickets have been sold.

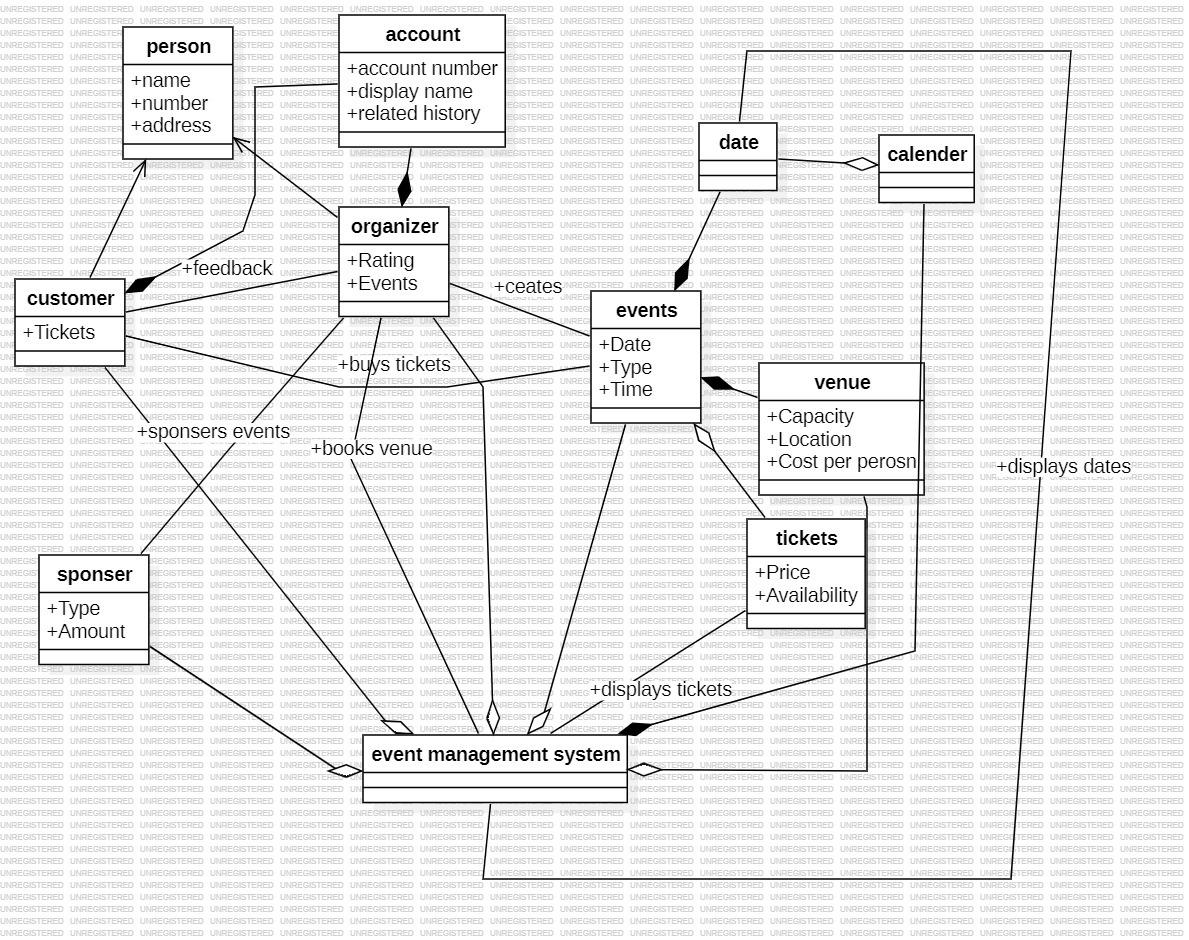
## Operating Environment

The software will operate on Java-compatible hardware platforms, supporting Windows, macOS, and Linux operating systems. It requires Java runtime environment (JRE) version 8 or higher for execution. Additionally, it interacts with the MySQL database management system for data storage and retrieval, ensuring compatibility with MySQL versions 5.7 and above.

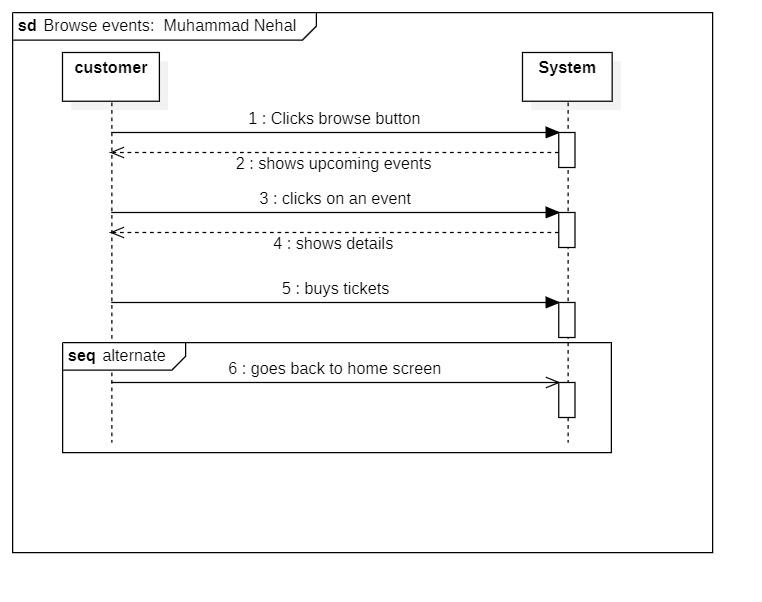
## User Interfaces

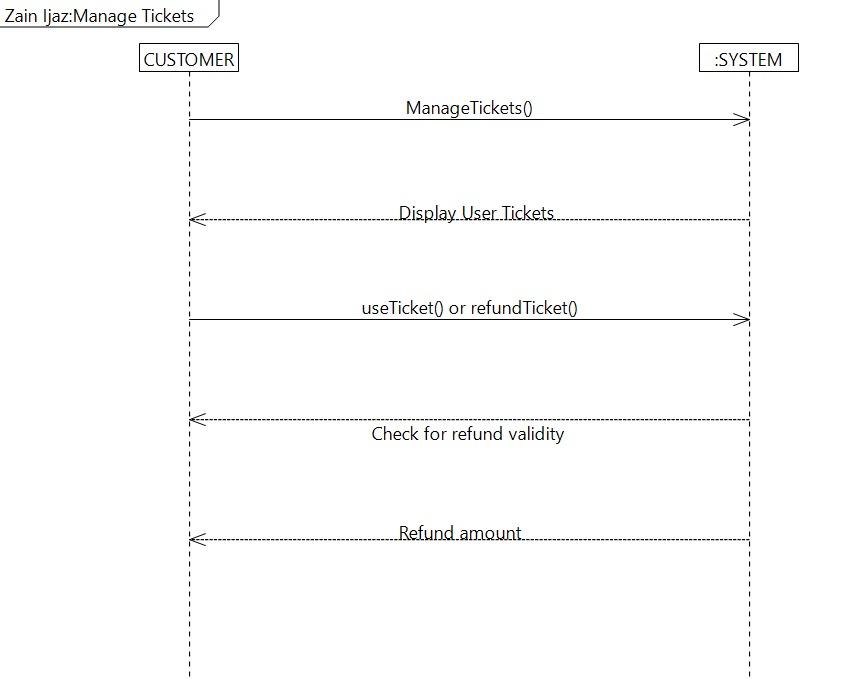
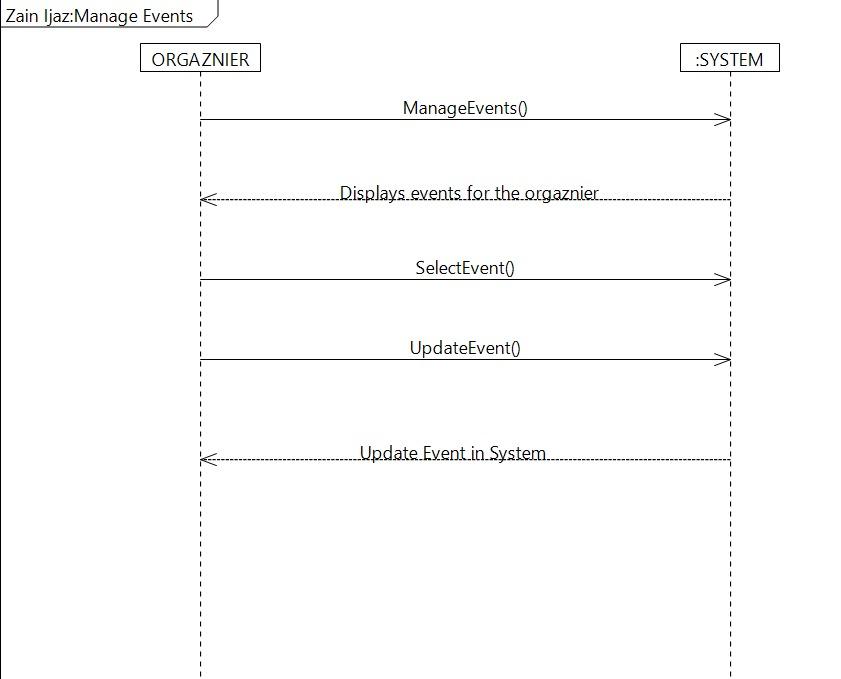
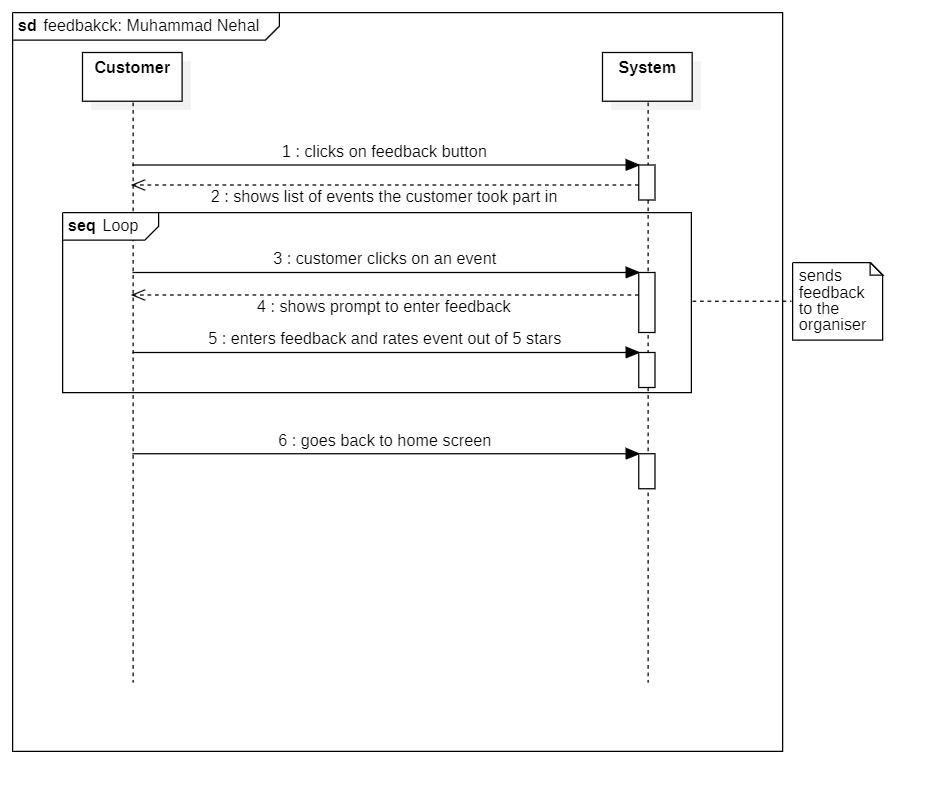
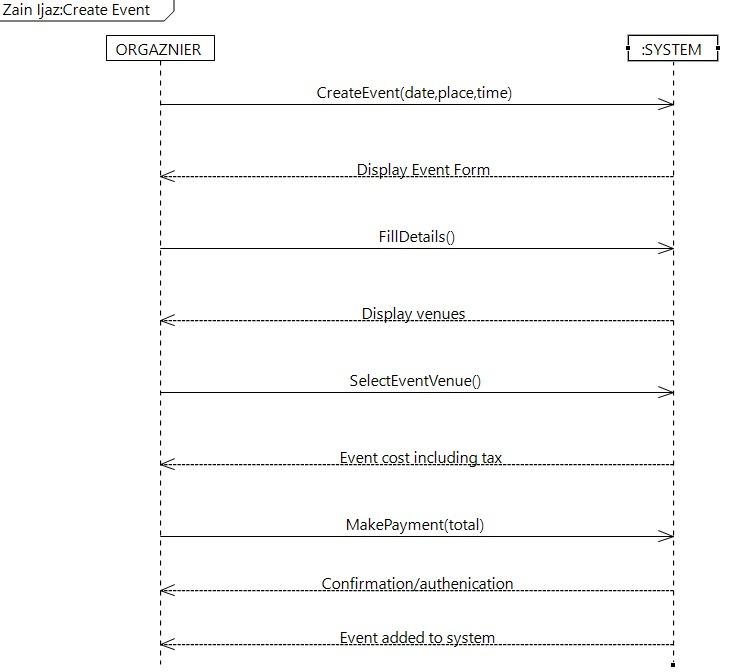
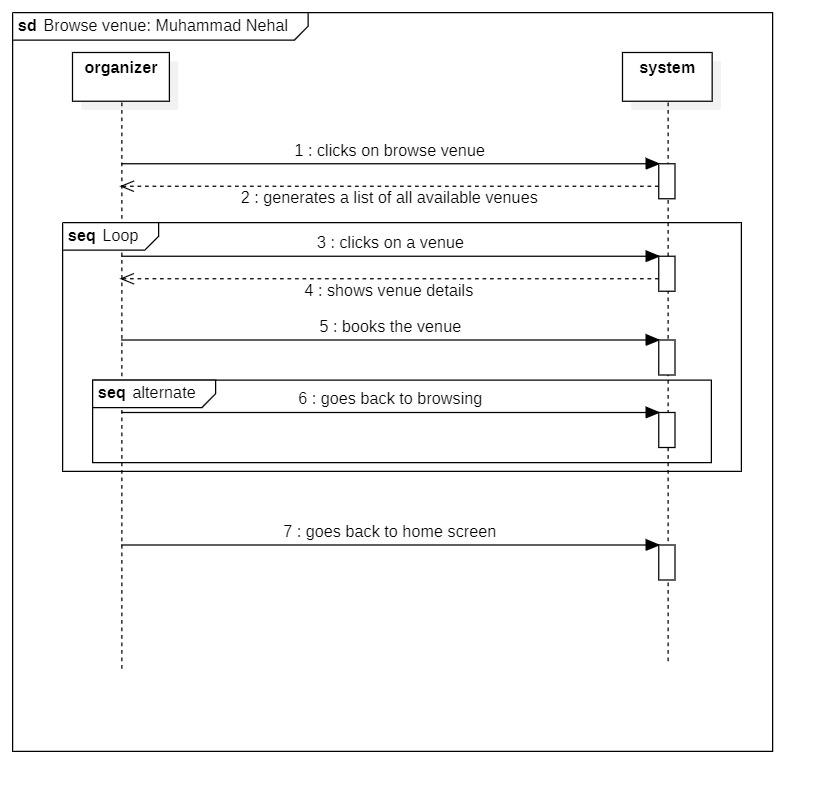
The user interface will provide an intuitive and user-friendly interaction experience for both organizers and customers. It will feature standard buttons and functions for navigation, such as "Create Event," "Manage Events," "Browse Events," "Buy Tickets," and "Leave Feedback." The layout will prioritize ease of use, with clear navigation menus and informative tooltips to guide users through the system. Error messages will be displayed prominently and provide helpful guidance for resolving issues. The interface will adhere to established GUI standards and ensure consistency across all screens. Specific details of the user interface design will be outlined in a separate user interface specification document.

# Domain Model

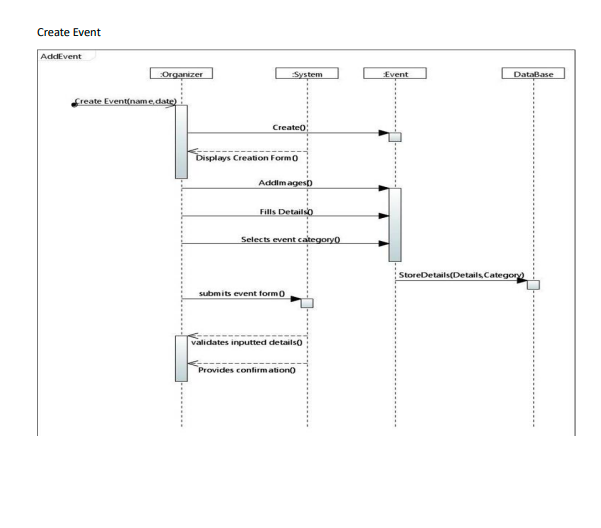


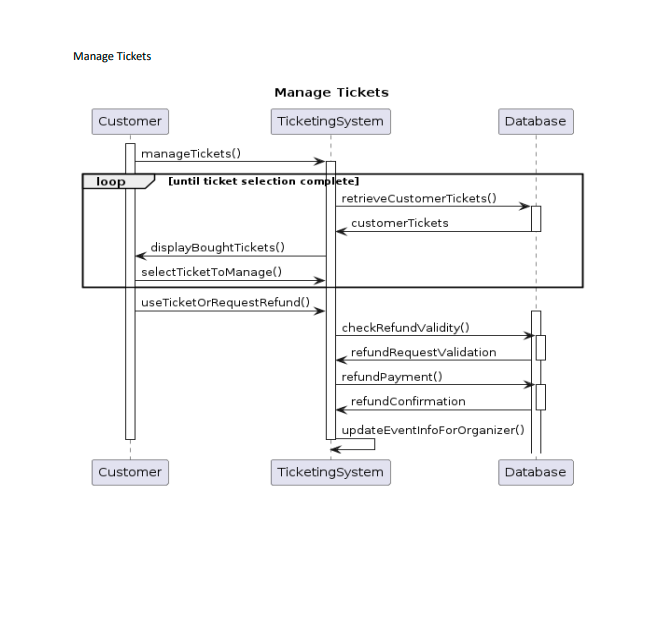
# System Sequence Diagram

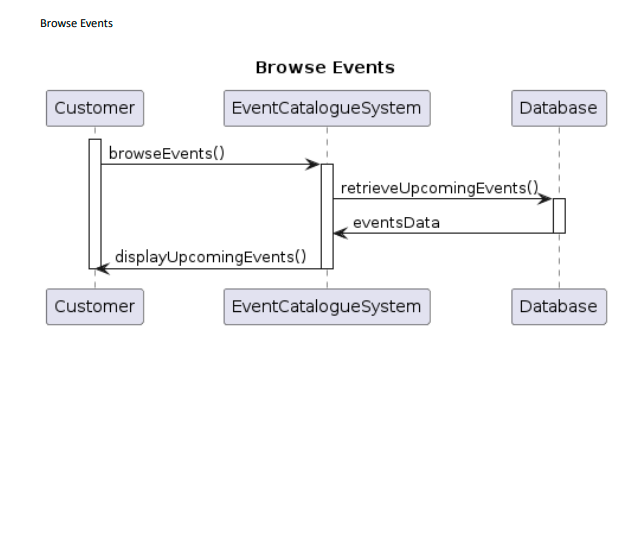


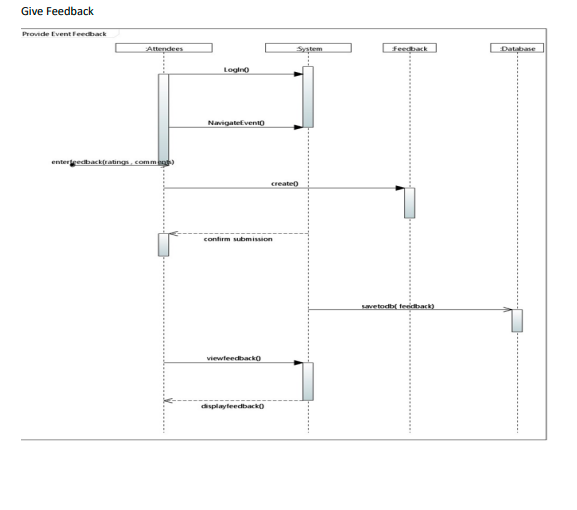


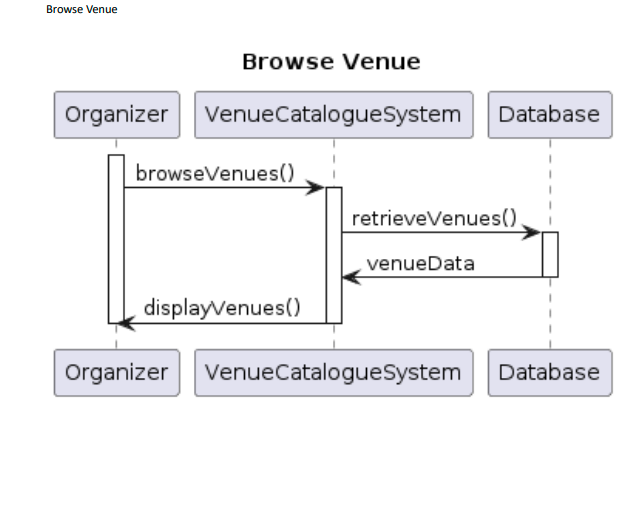
# Sequence Diagram





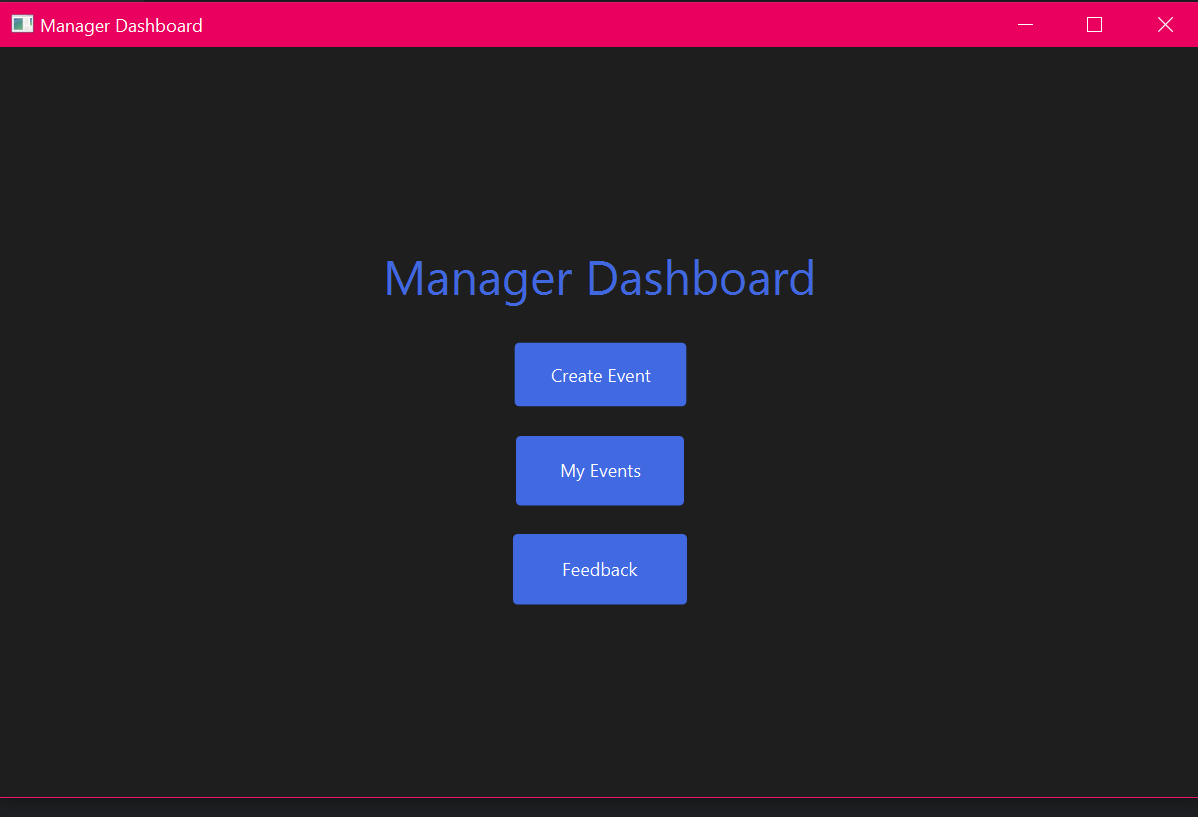


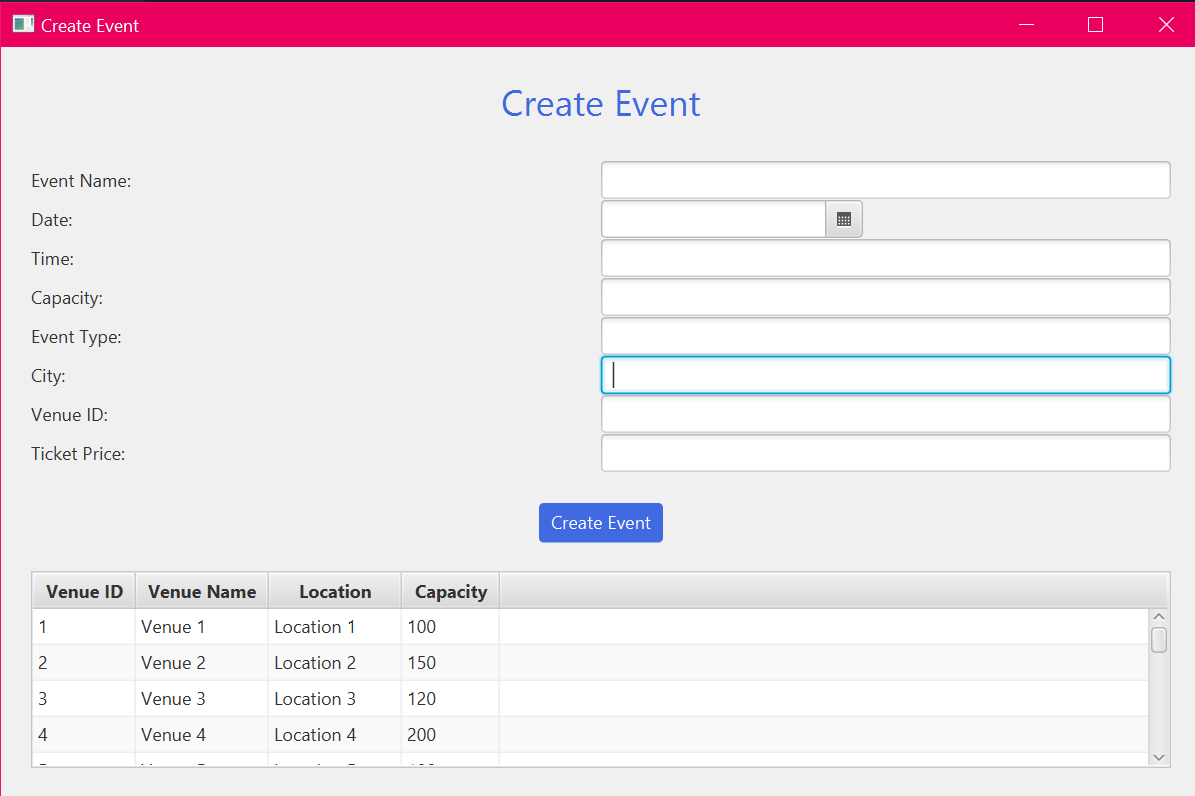


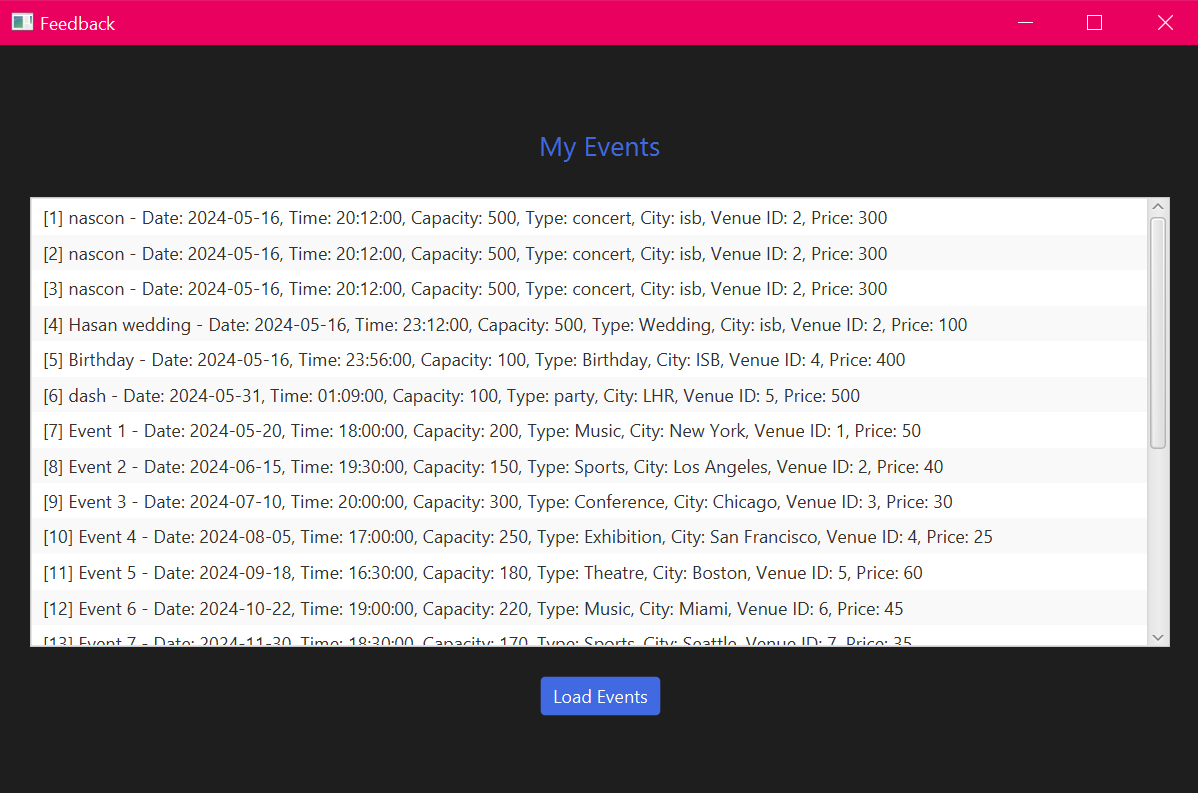


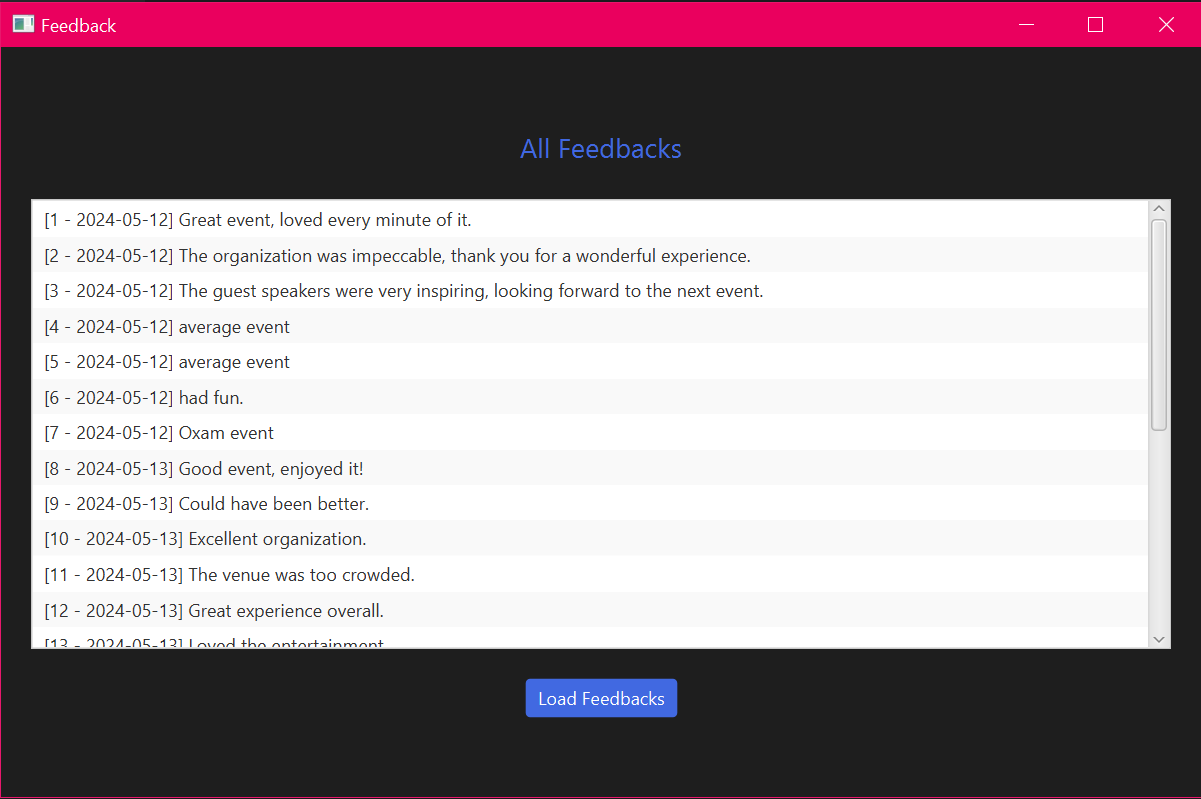
# User Interfaces

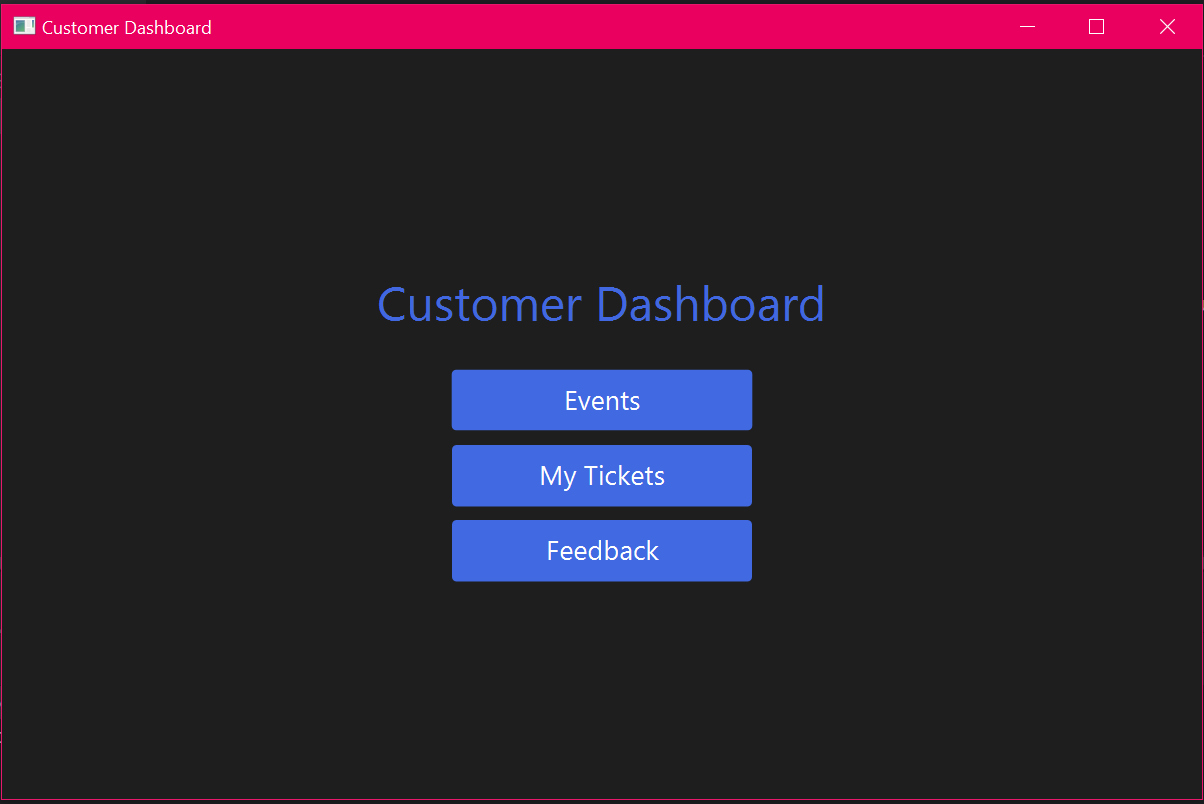


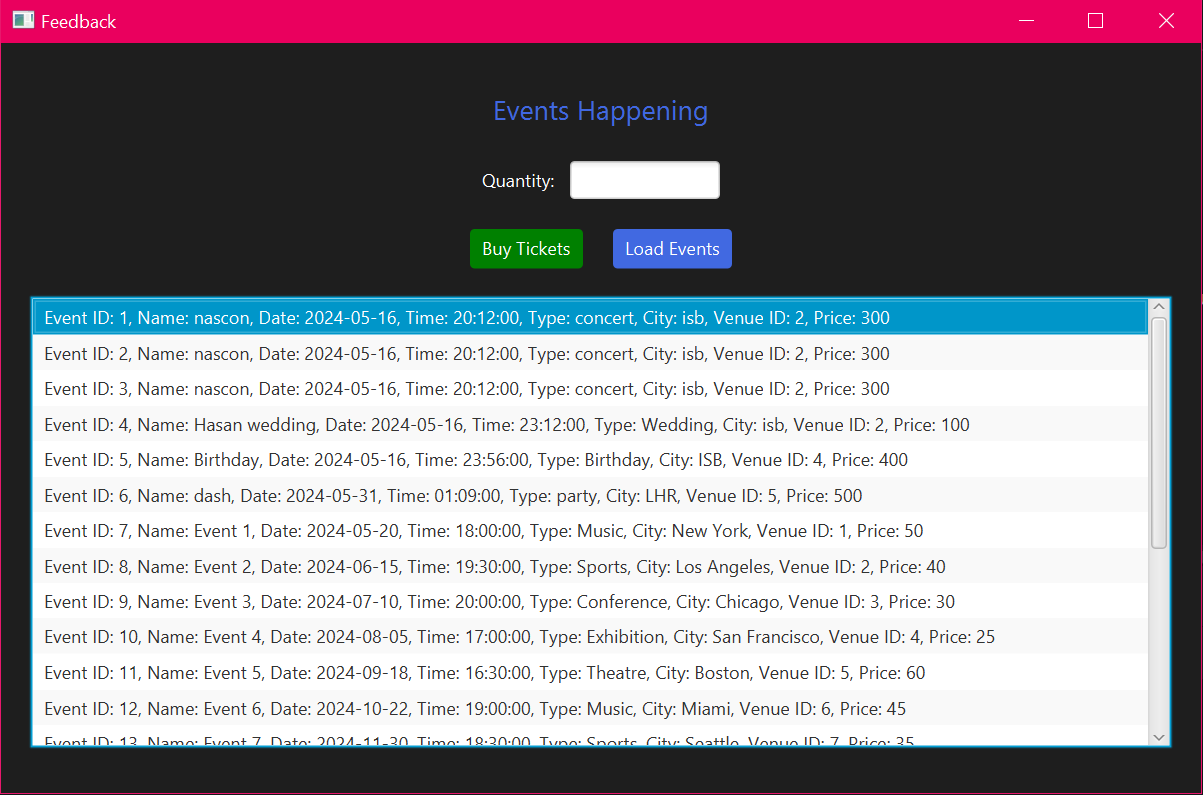


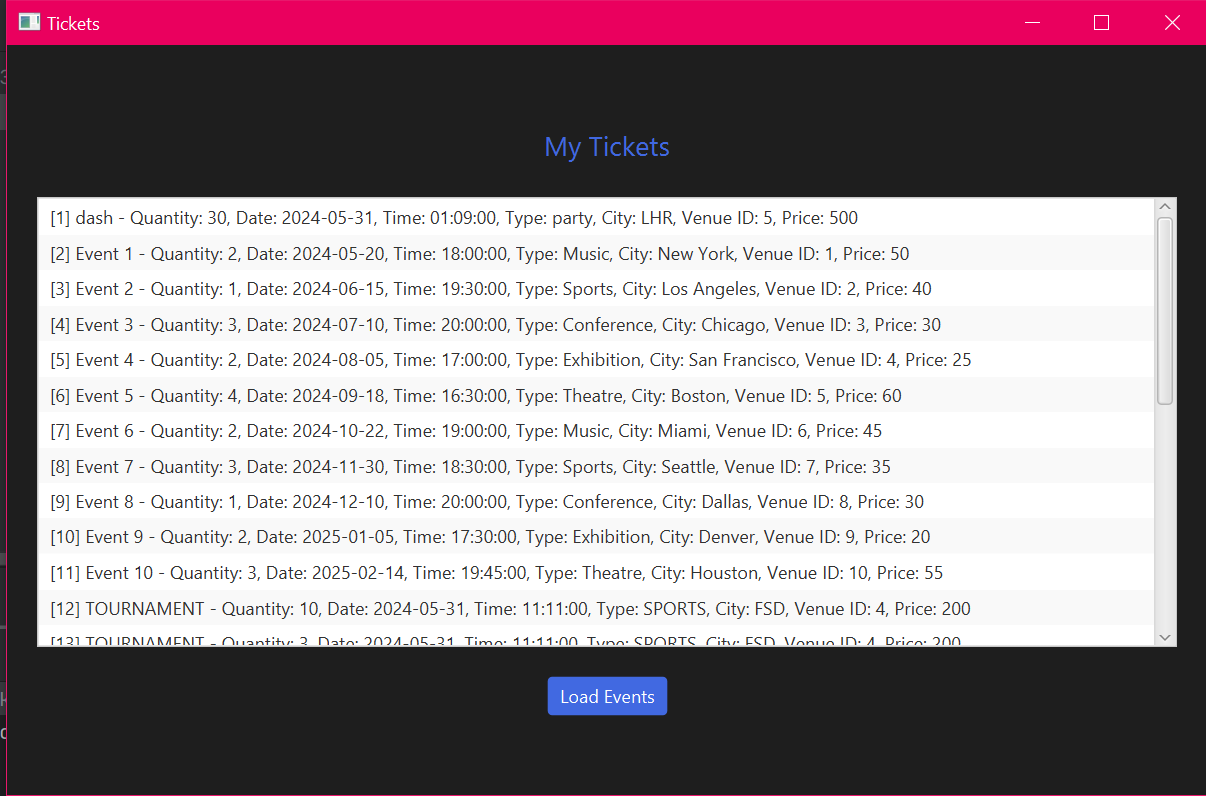


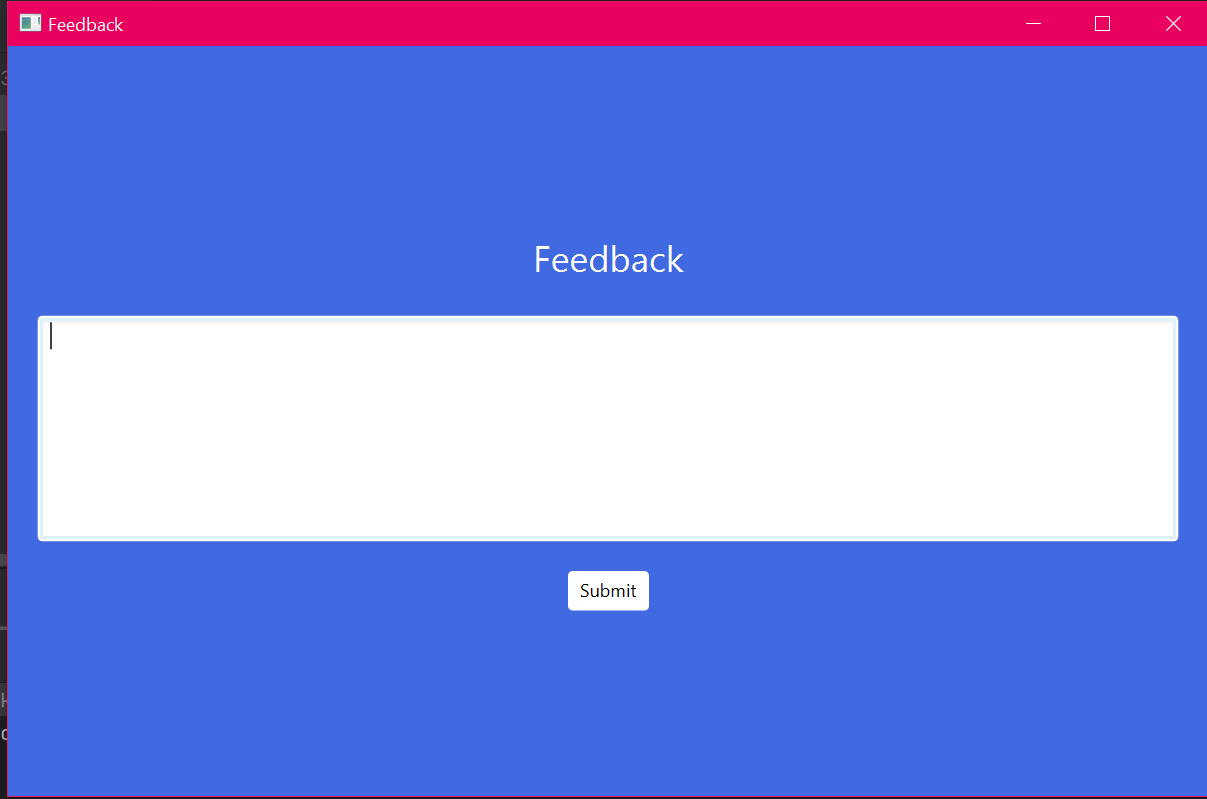












# Class Diagram

