

**CS 3354 Software Engineering**  
**Final Project Deliverable 1**

**FestGuide**

Fatima Khalid, Adrian Martin, Kamal Al Shawa, Jasmine Mathers, Rida Zameer,  
Mauricio Rodriguez Rios, Krishna

## 1. Final project draft description with instructor feedback:

### **Title of our project**

FestGuide

### **Group Members' first and last names**

Fatima Khalid

Adrian Martin

Kamal Al Shawa

Jasmine Mathers

Rida Zameer

Mauricio Rodriguez Rios

Krishna Chilukuri

### **What you'll be doing**

An application that helps organizations communicate information about festivals and other events. It would include important points of interest, and the ability to handle transactions among other things helpful for the people attending.

### **Detailed description of motivation and expected use cases**

Festivals and events are a crucial part of culture and society today. Attending such events is a detailed task that requires organization, communication, and solid navigation. Our motivation in creating this app is to streamline the process of hosting and attending events. The app would allow organizations to create their events while also allowing other users to access the information about these events. Meaning, the app would host event creation while also hosting event viewing on the same platform.

### **List of tasks to be delegated to each person (1 sentence per person)**

Kamal - UML Design

Mauricio- Use Case diagram

Adrian- Functional requirements, task 1.5

Rida - Class diagram, task 1.3 (add all team members & TA to Github)

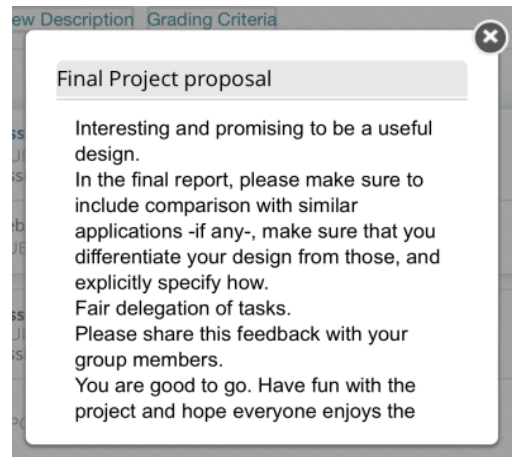
Jasmine - Sequence Diagram

Fatima Khalid - Overall project Management, task 1.4

### **Scholar paper?**

Not interested

### Instructor Feedback:



Based on instructor feedback, we are confident that our project will be a functional and practical one. To improve the quality of our design based on the feedback, we will compare FestGuide with competitors. It is important to clearly specify what makes FestGuide different from other similar applications, because there is no need for redundancy in the software market. We will be including these detailed descriptions when presenting our final project to the audience. We have also **slightly revised** the delegation of tasks since the original document plan since we felt different areas of the project required more attention and we are aiming for a sound design.

## 2. Set up GitHub repository

<https://github.com/RidaZameer07/CS3354-FestGuide>

- 1.1 Everyone has made a GitHub
- 1.2 Repo created and named
- 1.3 Added team members and TA
- 1.4 Did first commit (ReadME)
- 1.5 Added Project scope file
- 1.6 Added all other project related files

## 3. Delegation of Tasks

Fatima Khalid:

- Revising final project draft proposal based on Instructor feedback

- Setting up basic GitHub repo

Adrian Martin:

- Software requirements: Functional and non-functional
- Initial GitHub commit

Kamal Al Shawa:

- Architectural design

Jasmine Mathers:

- Sequence diagram

Rida Zameer:

- Adding GitHub collaborators
- Class diagram

Mauricio Rodriguez Rios:

- Use case diagram

Krishna:

- Software Model Process

#### 4. Which software model process is employed and why ?

- 1) Iterative Development: Given the goal of our app we have the iterative approach of scrum allows the team to deliver increasing value to users with each sprint.
- 2) Flexibility: As the user and market trends evolve with the times, scrum allows the team to act on feedback and make changes accordingly.
- 3) Collaboration: The success of an app depends on the collaboration with the event organizers and attendees to understand what is needed and should be included.
- 4) Risk Mitigation: By breaking the project into smaller parts, scrum can help prevent risks especially with a complex project like this.
- 5) Transparency: Being transparent helps stakeholders understand the progress made and the failures that have occurred, so that informed decisions can be made.

#### 5. Software requirements

##### a. 5-7 functional requirements

Users can add and edit event information such as name, date, time, and location to ensure accuracy and relevance.

Users can search for events based on categories, dates, or locations to find events easily.

Users can purchase event tickets within the app, providing convenience and an easy transaction experience.

The main page displays upcoming events, enhancing user engagement and encouraging participation.

Users can access event details which include date, time, location, and description for informed decision-making.

Users can share events on external platforms which helps facilitate event promotion and increases visibility among potential attendees.

The app includes a FAQ section addressing common event attendance questions, and providing support and guidance to users

b. One nonfunctional requirement for each type

Security Requirements: The use of user authentication and authorization mechanisms to protect user data. Also the Encryption of sensitive data such as payment information and user credentials.

Regulatory Requirements: Compliance with data protection regulations such as GDPR or CPRA to ensure user privacy.

Ethical Requirements: Being transparent in data usage and handling by ensuring user consent for data processing.

Legislative Requirements: Compliance with the local laws and regulations related to online ticket sales and event organization.

Operational Requirements: Scalability to accommodate an increasing number of users and events over time. Also, regular maintenance and updates to keep the system running smoothly.

Development Requirements: Adherence to coding standards and best practices to ensure code quality and maintainability.

Environmental Requirements: Minimization of resource consumption to reduce the environmental impact of the software.

Safety/Security Requirements: Measures to protect user safety during physical events, such as emergency contact information and venue safety guidelines. Also implementing security measures to prevent unwanted access to event-related data.

Accounting Requirements: Integration with accounting systems for accurate tracking of financial transactions and revenue generation from ticket sales.

Product Requirements: Compatibility with various devices to reach a wide user base and a well-made user interface design for easy navigation and interaction.

Organizational Requirements: App users must have a verified account to purchase tickets or create an event.

External Requirements: Following the regulations regarding the sale of tickets online and the management of events, ensuring that the platform is legally compliant and approved for use by regulatory bodies.

Efficiency Requirements: Optimized code for efficient use of system resources and minimization of key functionalities.

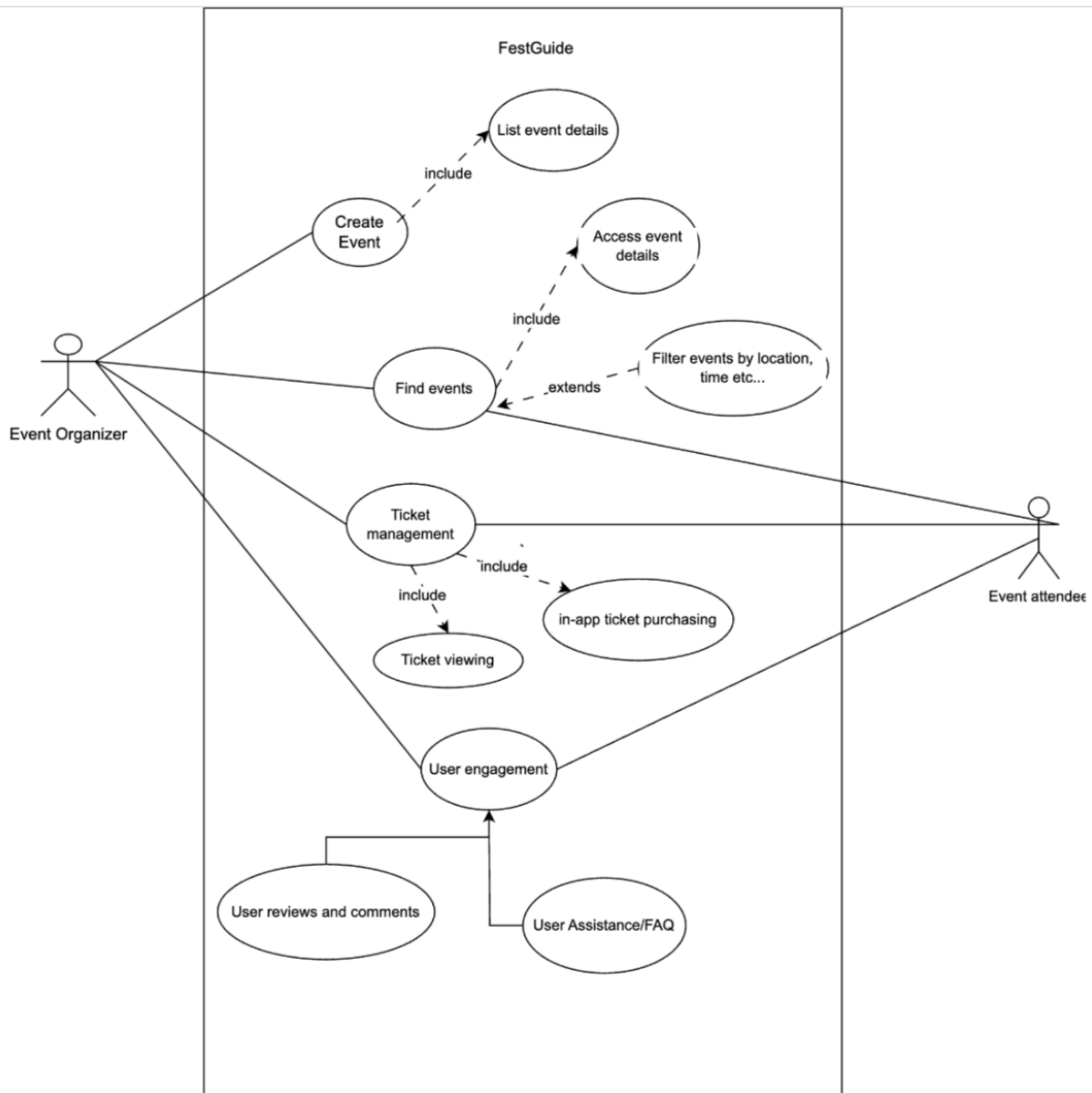
Dependability Requirements: Reliable error handling and logging to facilitate troubleshooting and debugging.

Performance Requirements: Fast response times for user interactions such as event searches, ticket purchases, and page loading.

Space Requirements: At least 500.555MB daily storage requirement for event data, user profiles, and multimedia content.

Usability Requirements: A user-friendly interface to enhance user satisfaction and engagement

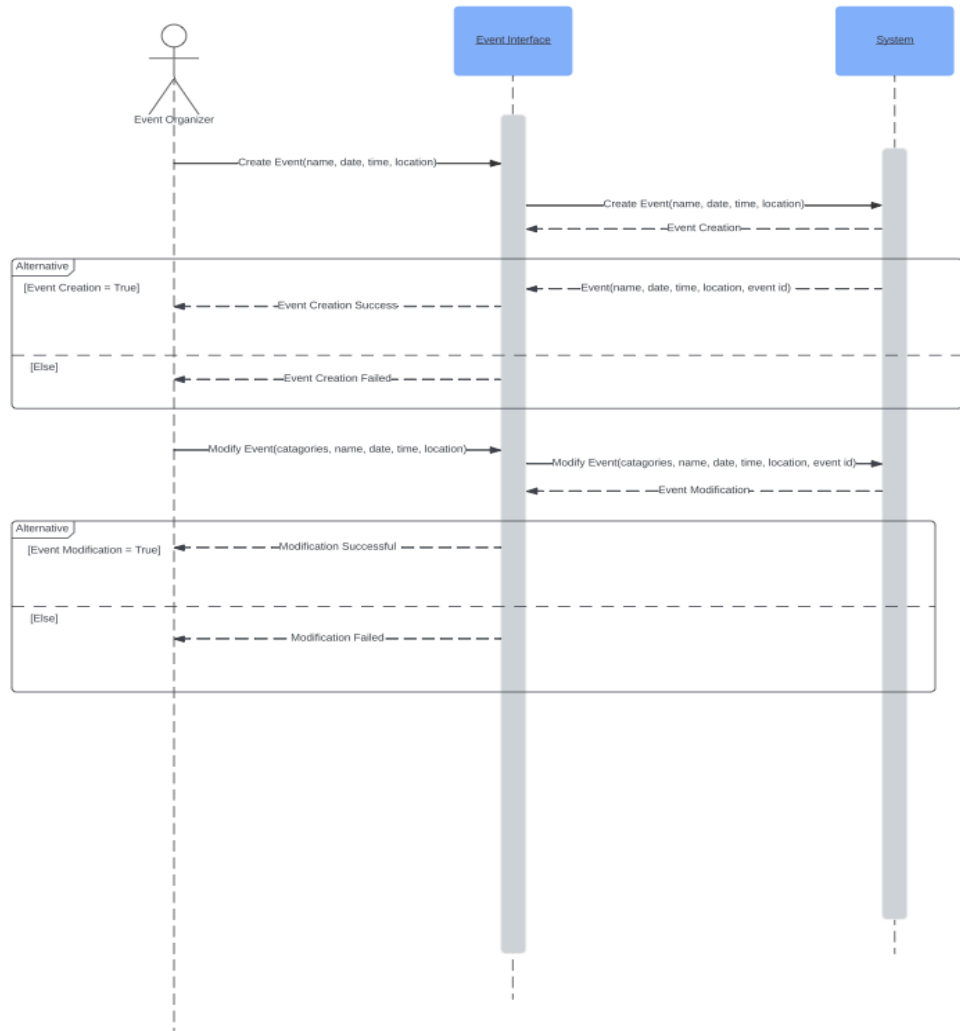
## 6. Use Case diagram



## 7. Sequence diagram



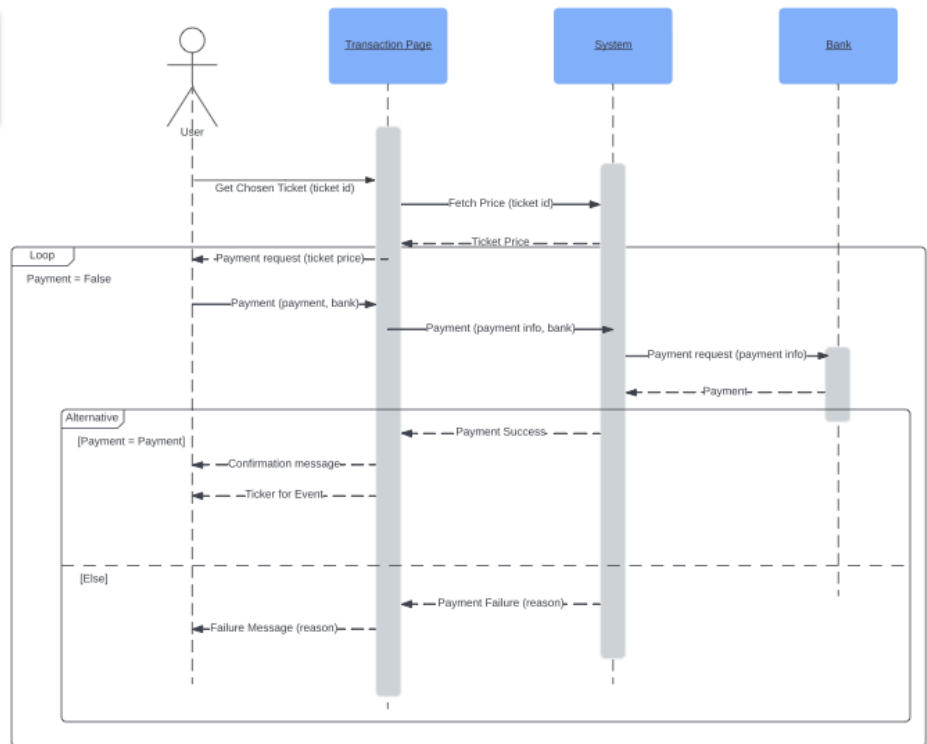
## Sequence Diagram 1.1



## Sequence Diagram 1.2 and 3.1

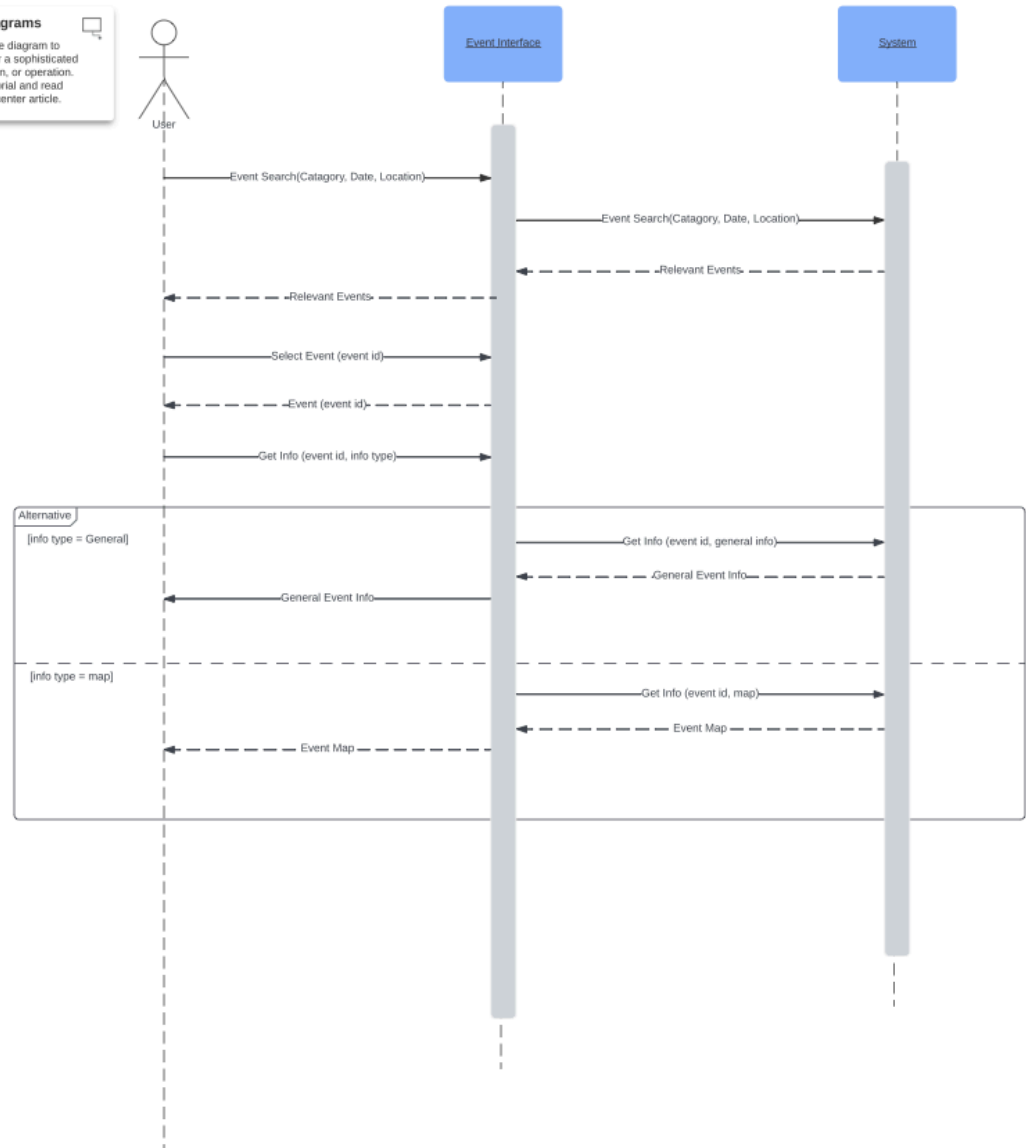
### Sequence diagrams

Create a sequence diagram to model the logic for a sophisticated procedure, function, or operation. Watch a basic tutorial and read more in this help center article.

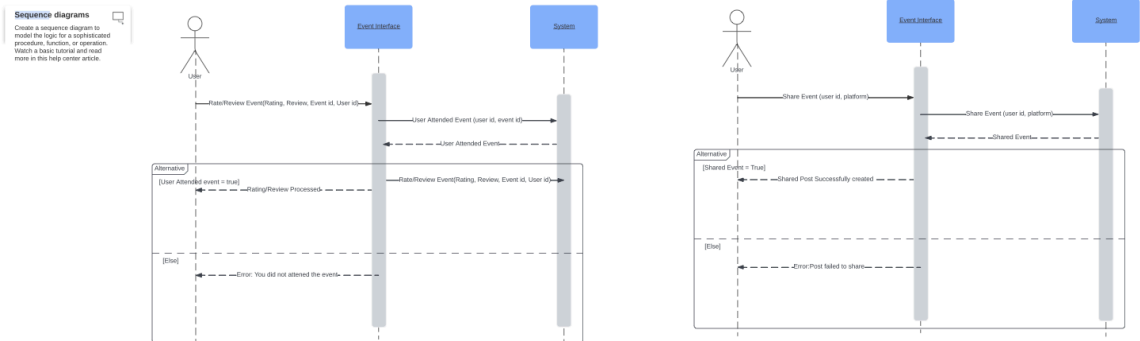


## Sequence Diagram 2.1 and 2.2

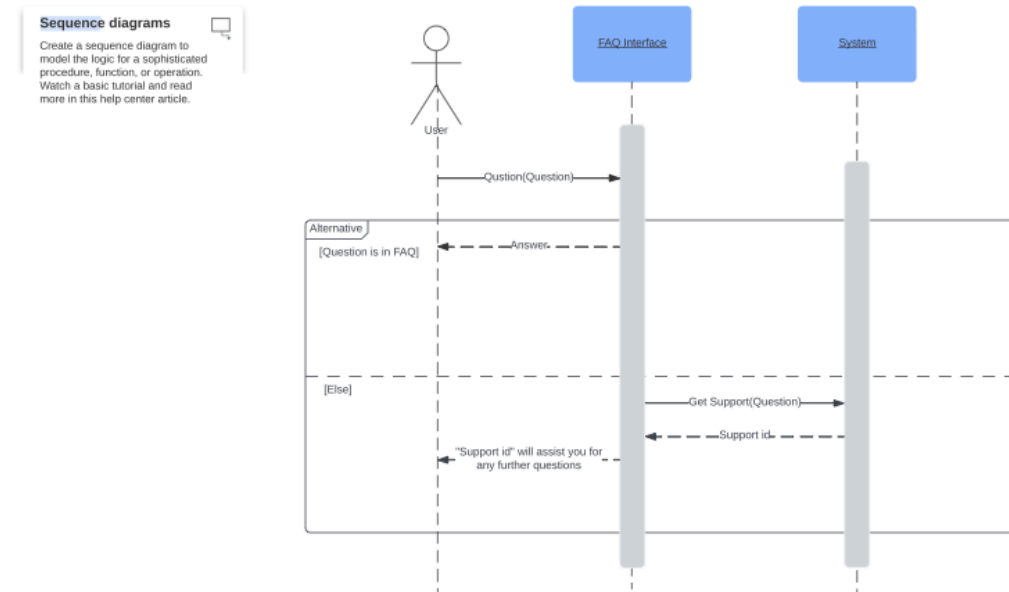
**Sequence diagrams**  
Create a sequence diagram to model the logic for a sophisticated procedure, function, or operation. Watch a basic tutorial and read more in this help center article.



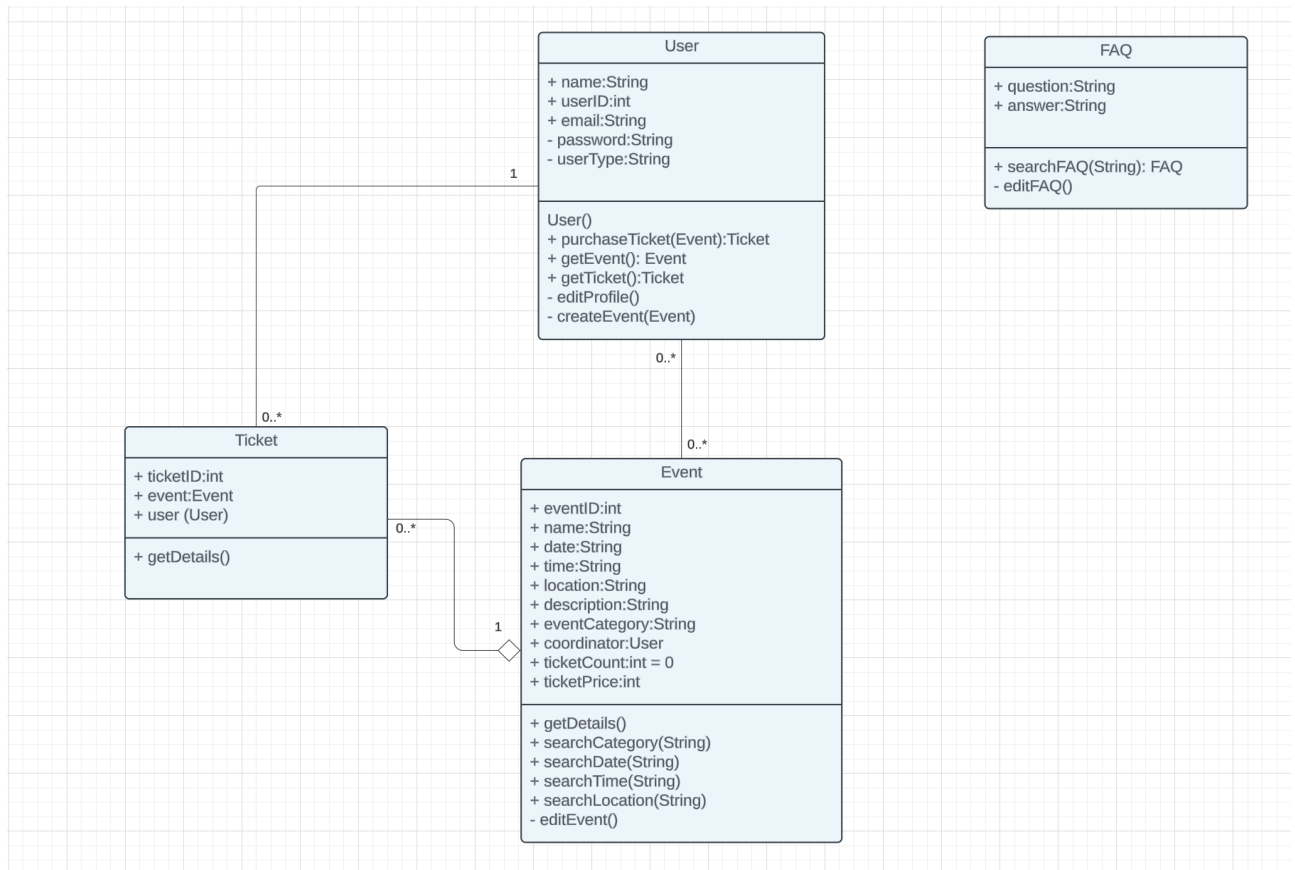
## Sequence Diagram 3.2



## Sequence Diagram 4.1



## 8. Class diagram



## 9. Architectural design - MVC pattern

