Assignment 2
Shotmaniacs Group 3

Ana Stoica
Tudor Matei
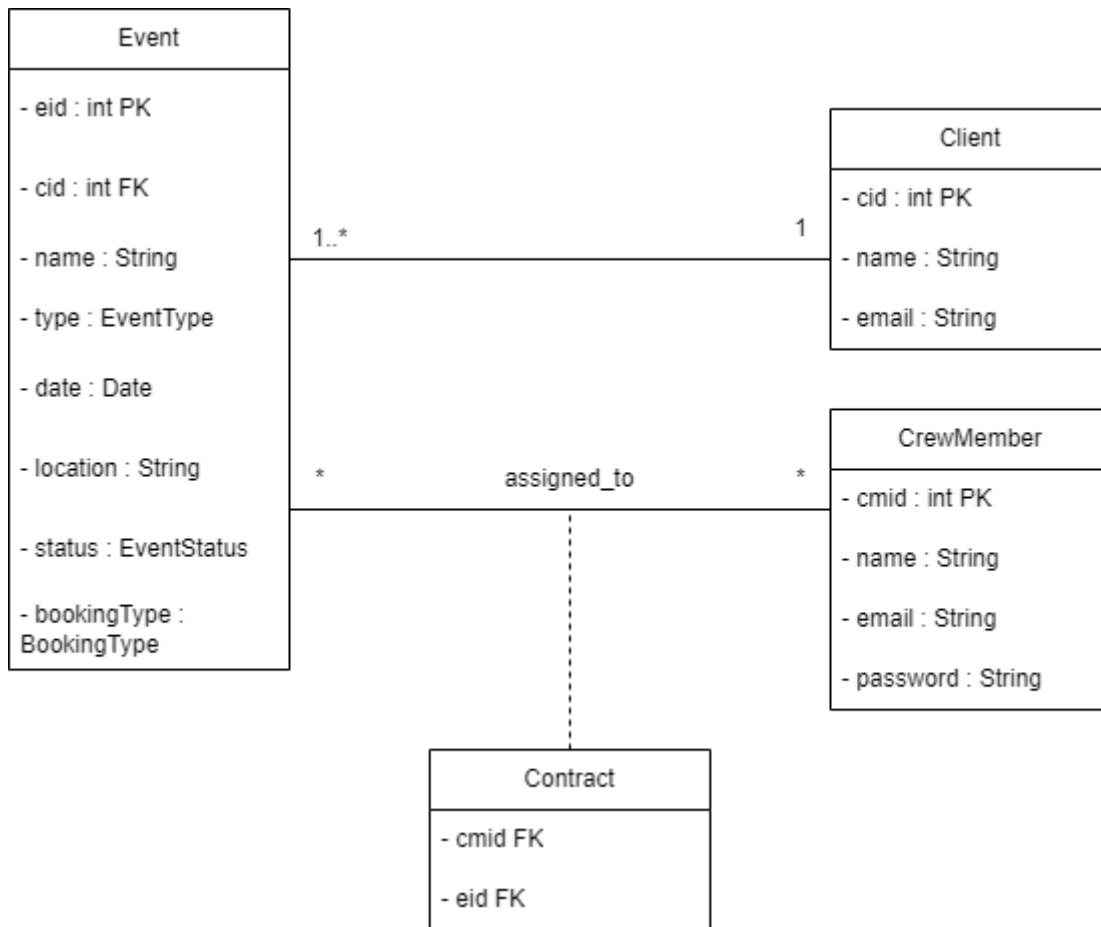Andrei Dorneanu
Roelof van der Hooft
Harveer Singh
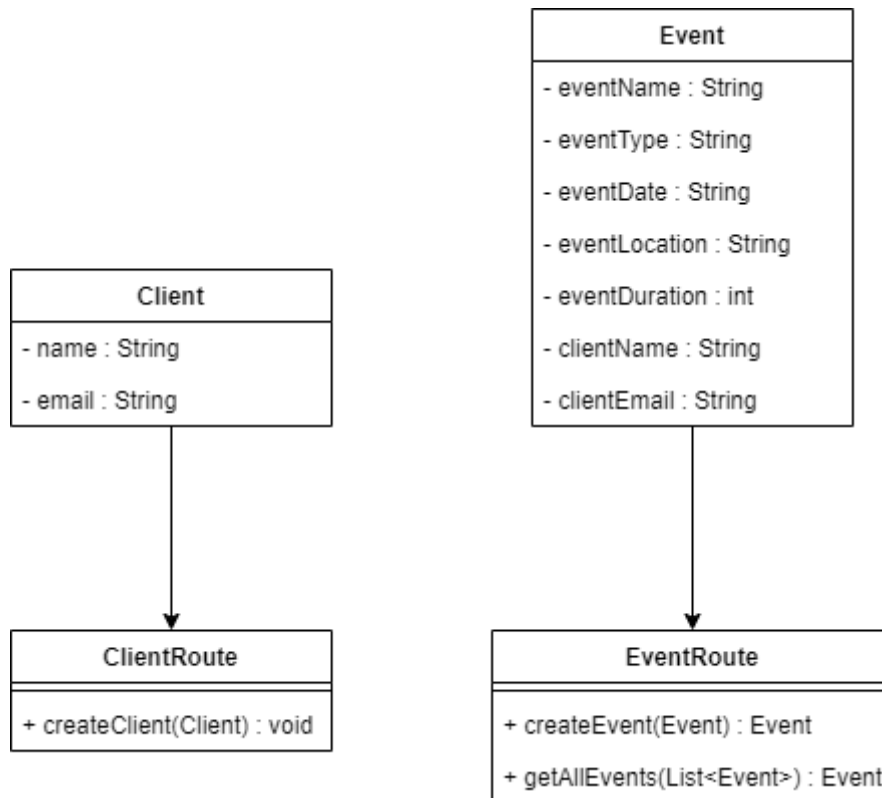Aron Eekma

Event

- eid : int PK

- cid : int FK

- name : String

- type : EventType

- date : Date

- location : String

- status : EventStatus

- bookingType : BookingType

Client

- cid : int PK

- name : String

- email : String

CrewMember

- cmid : int PK

- name : String

- email : String

- password : String

1..* ——— 1

* assigned_to *

**Contract**

- cmid FK

- eid FK

The diagram represents our basic database schema that we will use for storing events. Moreover we will store the client that reserved the booking and link it to the event. The crew member is linked to the event through a contract. This architecture allows us to correctly link the same client to multiple events and assign multiple crewmembers to the same event and also remove them.

Route Class Diagram:


```
┌─────────────────────────┐
│          Event          │
├─────────────────────────┤
│ - eventName : String    │
│ - eventType : String    │
│ - eventDate : String    │
│ - eventLocation : String│
│ - eventDuration : int   │
│ - clientName : String   │
│ - clientEmail : String  │
└─────────────────────────┘

┌─────────────────────────┐
│          Client         │
├─────────────────────────┤
│ - name : String         │
│ - email : String        │
└─────────────────────────┘

┌─────────────────────────┐         ┌─────────────────────────────────┐
│       ClientRoute       │         │           EventRoute            │
├─────────────────────────┤         ├─────────────────────────────────┤
│ + createClient(Client) : void     │ + createEvent(Event) : Event    │
└─────────────────────────┘         │ + getAllEvents(List<Event>) : Event │
                                     └─────────────────────────────────┘
```

The class diagram represents the current routes for our landing page of the website. On the landing page the potential client is able to register himself and an event by providing the correct event details. The frontend then calls the EventRoute class path and creates a new event. This event is linked to the database and is added accordingly. Moreover, the EventRoute class contains the getAllEvents GET request that allows us to populate the admin dashboard with the current events. The ClientRoute is used to create the client inside the database.

Use Case Class Diagram:
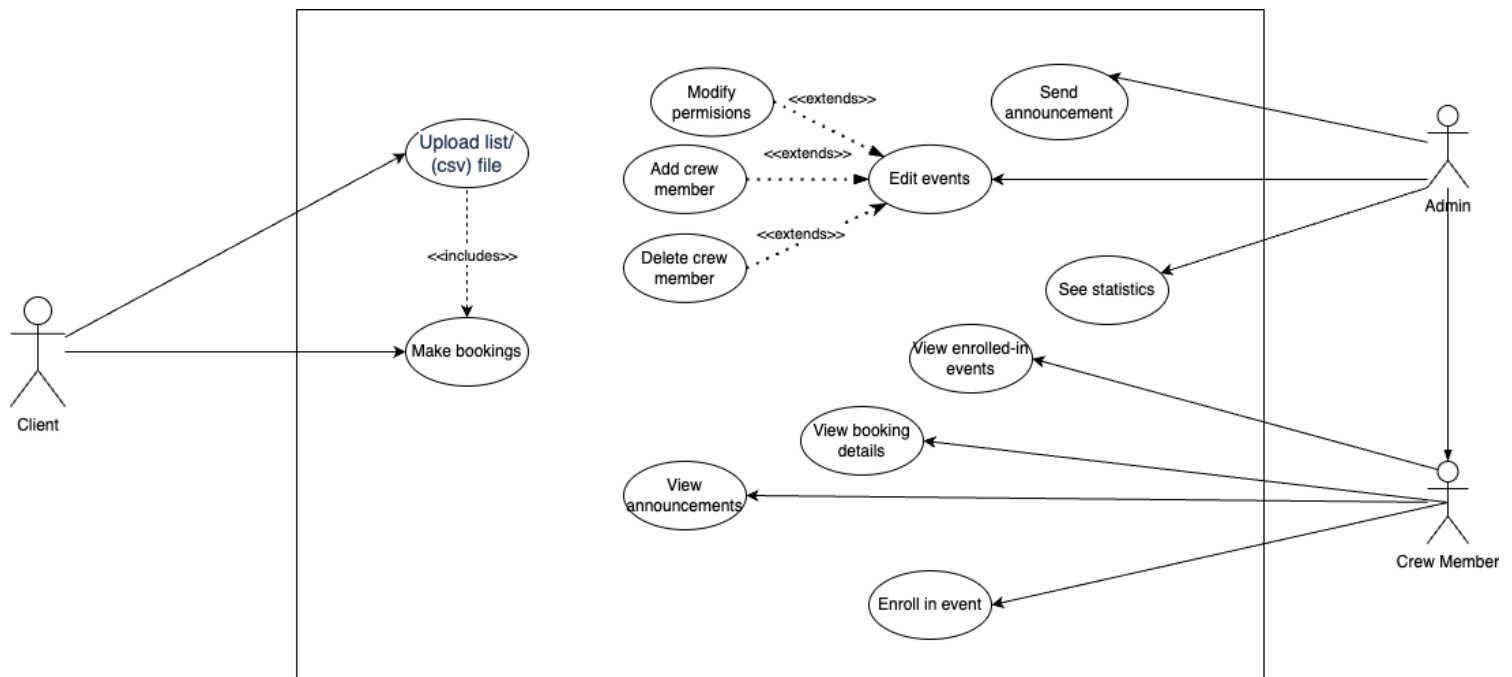
The use case diagram represents the simple view of the user stories we created on the Trello board. It includes the client, crew member and the admin. The admin is also a crew member, so it is implied that they can do whatever a crewmember can ("View booking details", "View enrolled-in"). A client can make bookings and also upload related files. However, uploading a CSV file also includes making a booking, because it would be pointless for someone to upload a file without booking.

This diagram is used to guide us through the basics of the functionality of the web app.