

# **Ticketing System Application**

## **Participating Students:**

Petrachioiu Radu Nicolae

Ispas Dinu-Ioan

Constantin Alin

**2022**

## Document description:

An application designed to buy tickets for different events. The users can buy a wide range of tickets for multiple use cases such as: concerts, private parties and cultural occasions.

## Project goal:

The goal of the project is to create a platform where people can easily purchase tickets for events. This should be done in a modern minimalistic way. The users should be able to seamlessly manage their tickets on the platform, moreover an email confirmation system is going to be implemented.

## Definitions and abbreviations:

The technologies that the project is going to implement are: Spring Framework for backend development, this is going to be used in order to create the register, login, email confirmation system and buying tickets. For the frontend, we will be using React Javascript library for the purpose of designing how the application looks and to interact with the created endpoints.

## Functional Requirements:

The application offers tickets for different events and the users decide to buy a ticket for a certain event. The features of the application consist in sorting the events after their type of activity, the user will have several types of events to choose from. The tickets which are bought will be generated with an unique QR code.

The user has requirements such as: introducing his credentials in order to create an account, they also have to choose a payment method e.g. Credit card. The admin of the web application will be able to create, edit and delete events and tickets accordingly. Event organizers will be able to request to create an event.

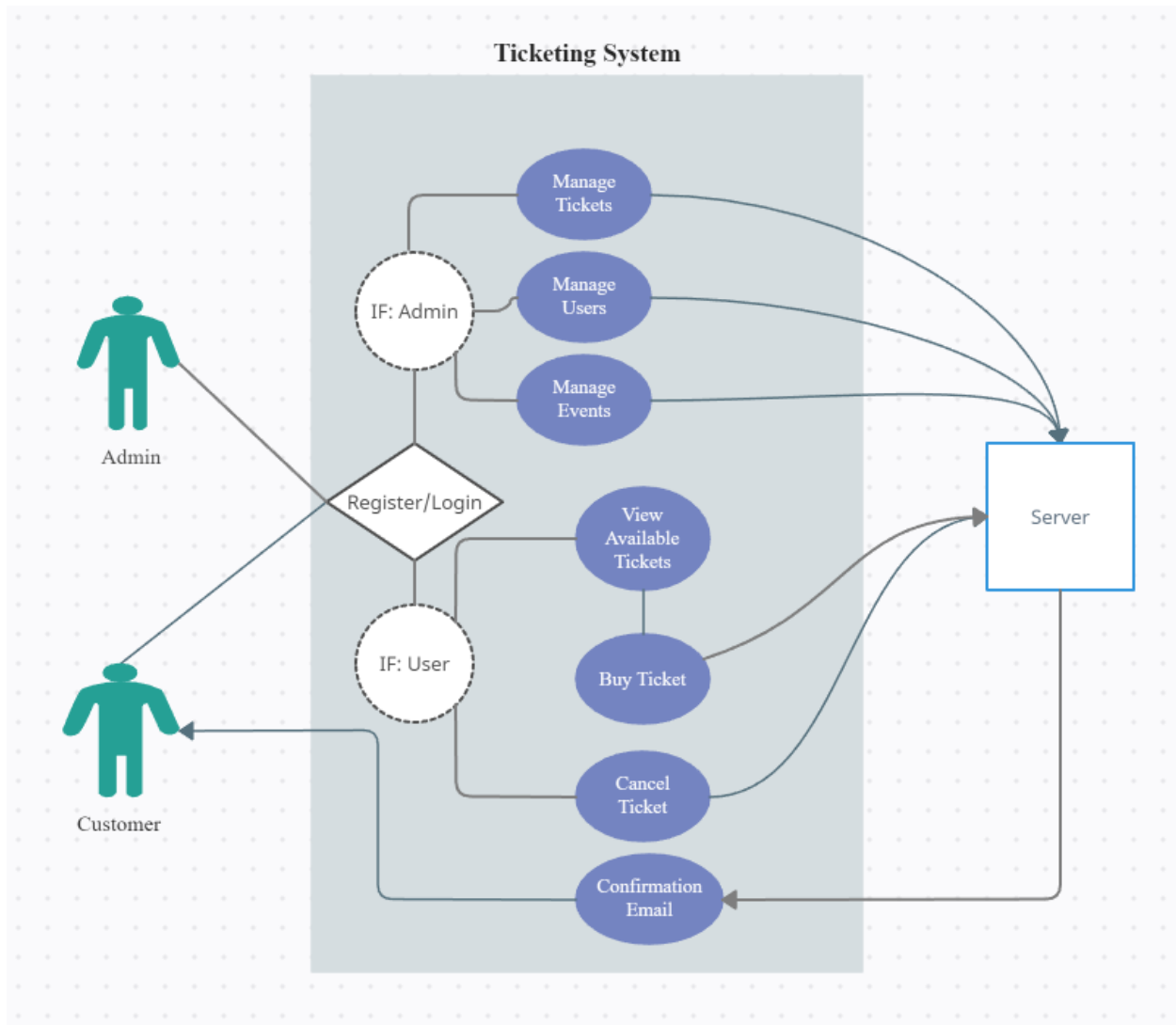
## Nonfunctional Requirements:

- The application will confirm that a ticket has been bought and then after a maximum of one minute, an email should be sent to the user with the ticket and confirmation.
- After pressing the login button, it should take maximum 10 seconds until the user is logged in.
- After creating an account, the user should receive an email to verify his account.
- The password will be encrypted with the hash method.
- The system should be available 24/7.
- The web application should be displayed properly regardless of the device used to access it.

## Stakeholders:

The stakeholders are users that are looking to attend concert, parties etc. Some of our competitors are: iabilet.ro, eventim.ro, etc. The only people which have the authority to influence the project or its outcome are the founders.

## Case diagram:



Case table:

- **Register case**

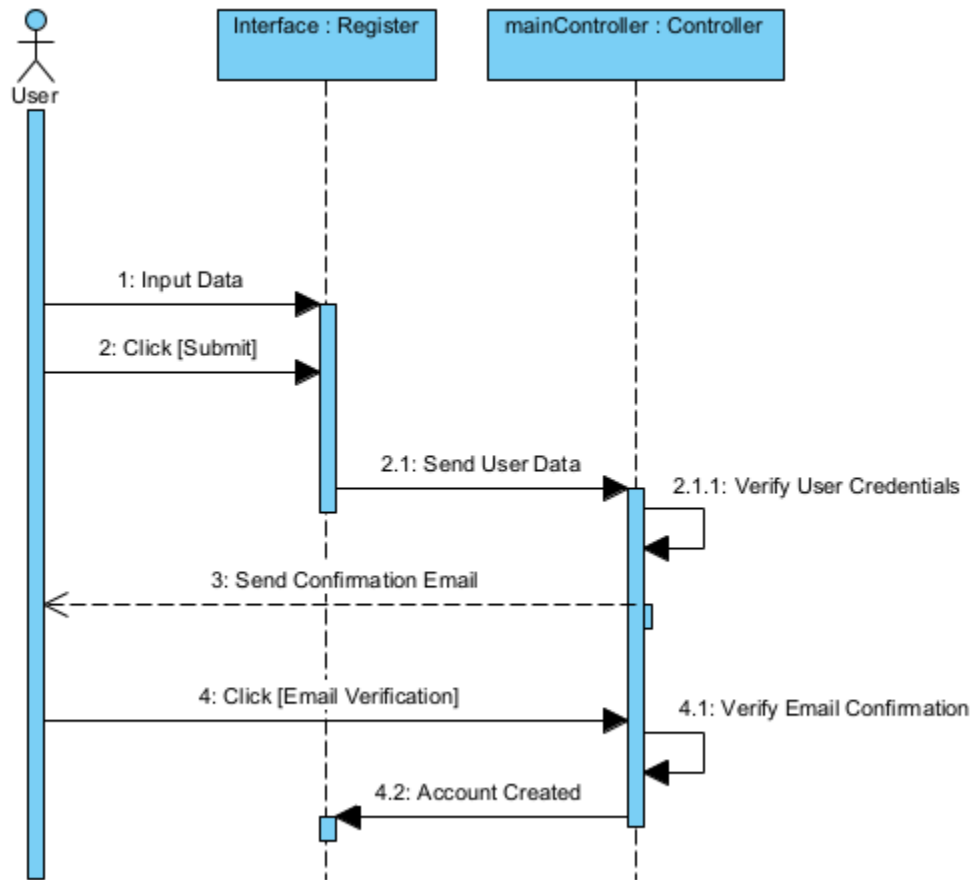
Client	User
Description	This case describes how an user can create an account to have access to the web application.
Pre-Conditions	The user needs to fill in the credentials in order to create an account, specially the mandatory fields, such as name, email etc.
Post-Conditions	After the user completes the register process, he will gain access to the core features of the application.
Flow of the action	1 <sup>st</sup> the user needs to fill every blank space. 2 <sup>nd</sup> the user needs to press the submit button to send the data further. 3 <sup>rd</sup> the user will receive a confirmation link on the email that needs to be clicked. 4 <sup>th</sup> after the account is verified, the register process is complete.
Alternate Flow	If the user will leave empty spaces, an alert will be raised. If the user doesn't click on the confirmation link in the email, the account won't be verified.

- **Buy ticket case**

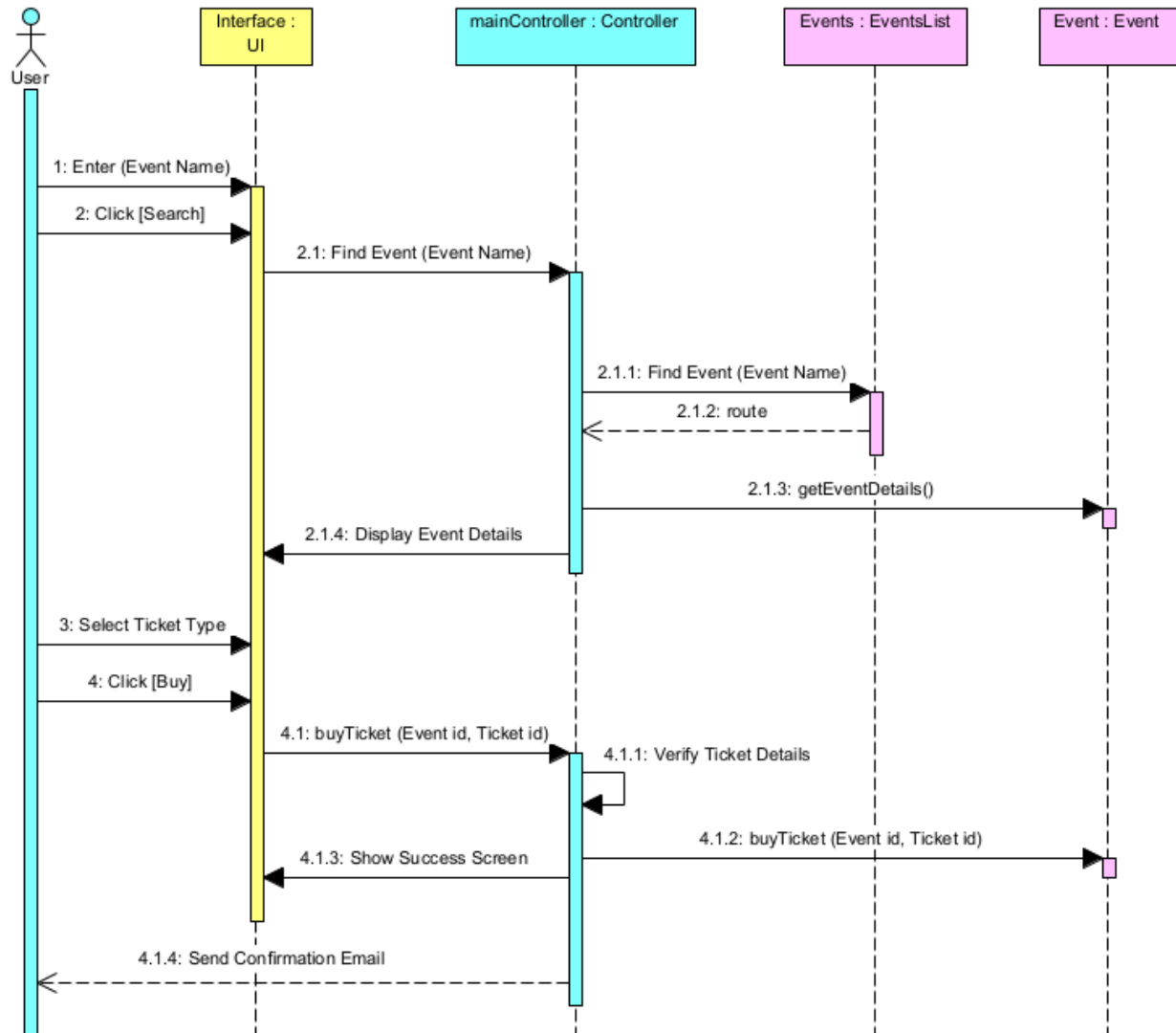
Client	User
Description	The user has to choose a ticket to buy for an event that he/she wants to attend. They need to choose a payment method.
Pre-Conditions	The user needs to choose the event he wants to attend, order a ticket for it, choose his payment method and then pay for it.
Post-Conditions	After the user completes those steps, he will receive an email confirmation in which he has the ticket.
Flow of the action	1 <sup>st</sup> the user has to select an event. 2 <sup>nd</sup> the user has to select how many tickets he/she wants to buy. 3 <sup>rd</sup> after the number of the tickets has been selected, the user will have to pay for the tickets. 4 <sup>th</sup> after the user has paid for the tickets, he/she will receive an email to confirm the transaction.
Alternate Flow	If the user doesn't have the funds required for the transaction, an email regarding the failure of the process will be sent.

## Sequence Diagrams:

- **Register Case**

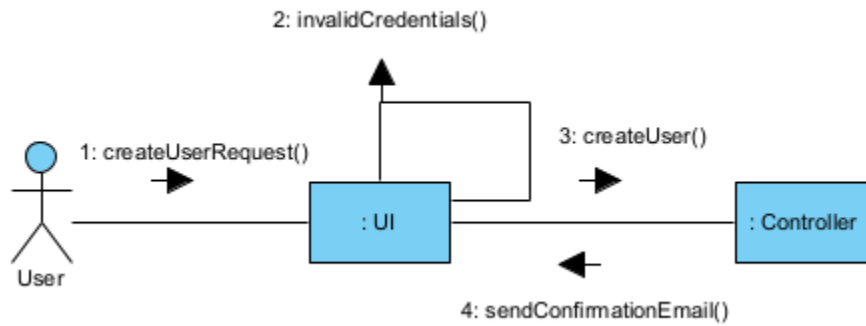


- Buy Ticket Case

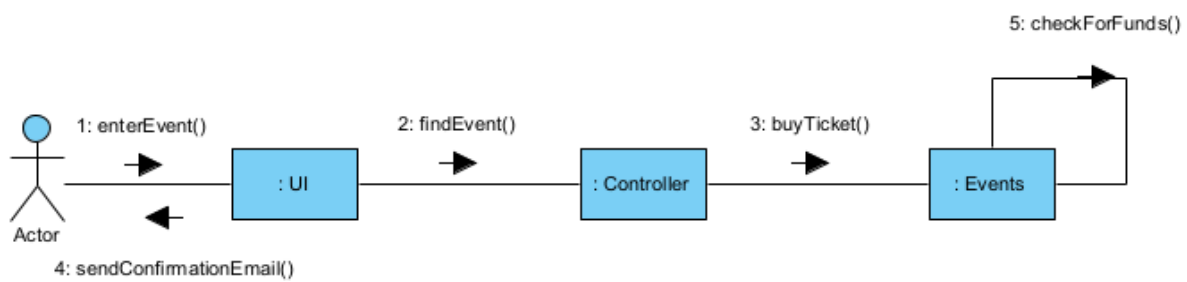


## Communication Diagrams:

- **Register Case**

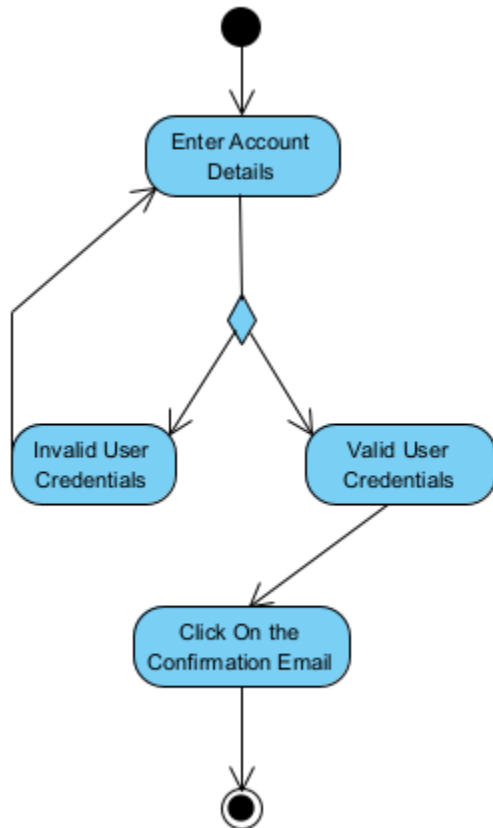


- **Buy Ticket Case**



## Activity Diagrams:

**Register Case**



**Buy Ticket Case**

