Tunisian Republic Ministry of Higher Education and Scientific Research

University of Carthage

Higher Institute of Information Technologies and Communication

# Report
# Of
# Requirements specification

## Subject:

**Development of a Social Network**

## Prepared by:

**Aziz Badis, Mansali Omar, Moussa lejmi,Baha Eddine Gassar**

## Academic year: 2023- 2024

# 1.Introduction:

## 1.1. Description:

“Evently” is an innovative app designed to simplify the process of making friends and planning gatherings based on shared interests. It provides a platform for discovering and connecting with like-minded individuals, allowing users to join existing events or create their own. By focusing on genuine connections and meaningful experiences, “Evently” aims to redefine social interaction and foster lasting friendships.

## 1.2. Objective:

The objective of Evently is to simplify the process of making friends and planning gatherings by connecting individuals who share common interests. It aims to facilitate meaningful social interactions and foster genuine connections by providing a platform where users can discover events related to their passions and connect with like-minded people. Overall, Evently seeks to redefine the way people connect and socialize in the digital age, with a focus on creating lasting friendships and memorable experiences.

# 2.Functional Requirements Specification:

## 2.1. Requirements:

The project aims to create a study website that meets the following needs:

- Publish Post
- Register account
- Authenticate

- Organize Event
- React

## 2.2. Actors Identification:

An actor is an entity, whether it be a natural person or a legal entity, that directly participates in a specific action or project. Therefore, it is essential to clearly specify which action or series of actions we are seeking to determine the involved actors. In our platform, we identify four types of actors who interact directly with the system: User: This category includes anyone accessing the platform and having the ability to create an account, manage it, and receive notifications regarding the funds they have marked as favorites.

# 3.Non-Functional Requirements Specification:

Non-functional requirements are quality criteria that indirectly influence the outcome and performance of the user, making them essential and not to be neglected. To meet these requirements, our social media project must satisfy the following criteria:

- **Reliability**: The application must operate consistently, without errors, and provide a satisfactory user experience.
- **Error Handling**: Ambiguities should be signaled by clear and organized error messages to effectively guide the user and familiarize them with our platform.

• **Usability and User-Friendly Interface**: The application should be intuitive for the user, with smooth navigation between different sections, clear and easy usage, as well as an attractive visual presentation.

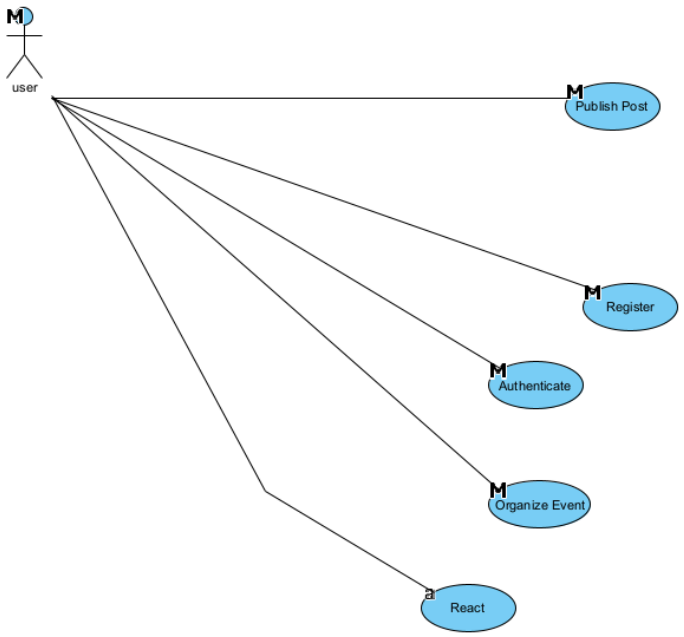
• **Security**: Our solution must ensure the confidentiality of users' personal data, as privacy protection remains a crucial priority in online applications

. • **Maintainability and Reusability**: The system must conform to a standard an

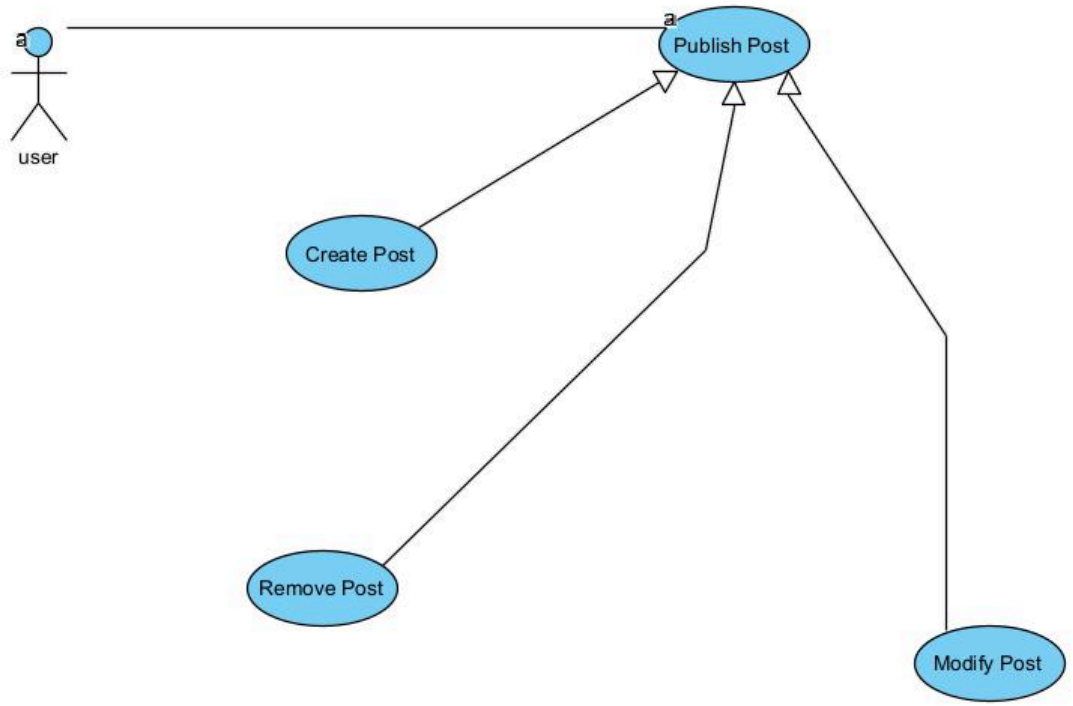
# 4.Conception UML:

## 4.1User case diagrams:

1.Global use case diagram:



2.Publish Post Diagram:




The following table elaborates on this user story with textual description:

| | |
|---|---|
| **Use case scenario** | as a user, i can publish a post |
| **Actors** | User |
| **Pre-Condition** | The user must be<br>connected |
| **Post-Conditions** | Post published |

| | |
|---|---|
| **Main scenario** | The user will accesses the post creation feature, and composes the content they wish to share. After reviewing and confirming, the user clicks”publish”, making the post visible to their network |
| **Extensions** | ● create post<br>● remove post<br>● modify post |

3.Manage account diagram:


The table below provides a detailed description of this user story in text.:

| | |
|---|---|
| **Use case scenario** | as a user, i can be register |
| **Actors** | User |

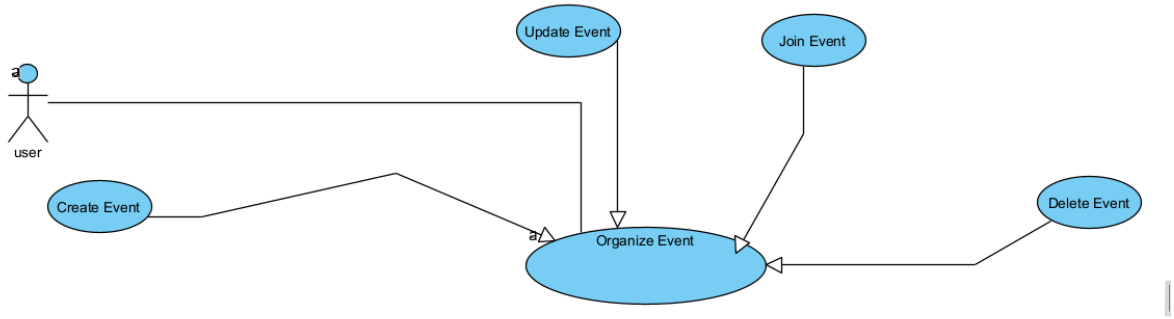
| | |
|---|---|
| **Pre-Condition** | User must be connected to the internet |
| **Post-Conditions** | User registred |
| **Main scenario** | The user will put all his information in the form singup then he will be registred as a user in this platform |

4.Authentification:


The table below provides a detailed description of this user story in text.:

| | |
|---|---|
| **Use case scenario** | as a user, I can be authticated |
| **Actors** | User |
| **Pre-Condition** | User must be registered |
| **Post-Conditions** | User authticated |
| **Main scenario** | The system will allow the user to access and use the platform. |

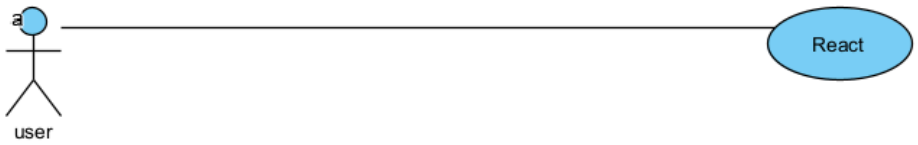
5.Organize Event:




The table below provides a detailed description of this user story in text.:

| | |
|---|---|
| **Use case scenario** | Like a user, i can orginaze event |
| **Actors** | user |
| **Pre-Condition** | The user must be connected |
| **Post-Conditions** | Event organized |
| **Main scenario** | In the event interface, users can explore upcoming events and choose to join ones that interest them or create their own events. When a user creates an event, a group chat specific to that event is automatically generated to facilitate communication among attendees. |
| **Extensions** | • Join event<br>• Create event<br>• Delete event<br>• Update event |

6.React:

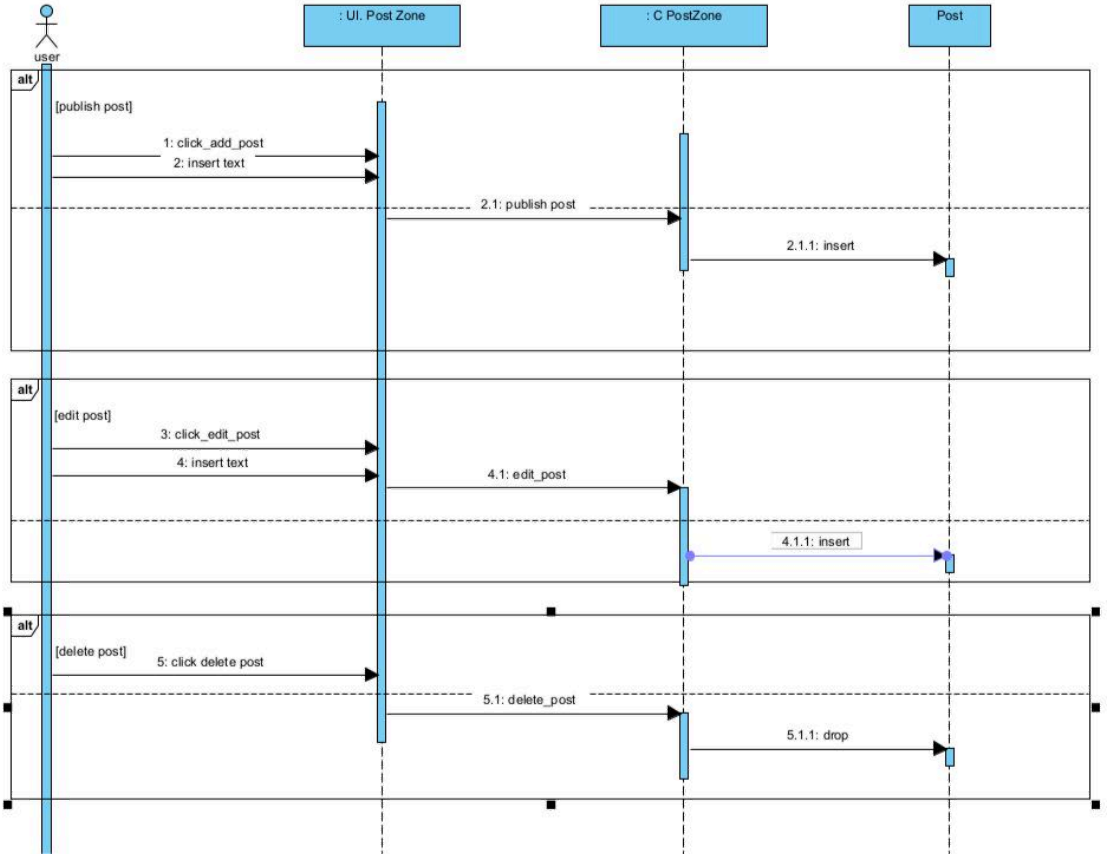



The table below provides a detailed description of this user story in text.:

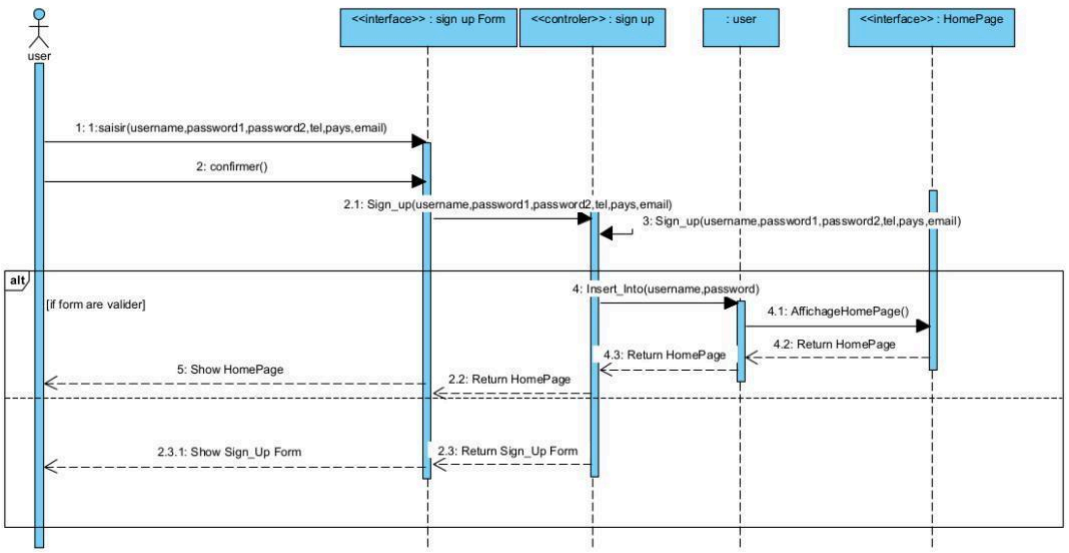
| | |
|---|---|
| **Use case scenario** | As a user, I vote by like or dislike |
| **Actors** | User |
| **Pre-Condition** | -The user must be inscripted |
| **Post-Conditions** | Vote sent |
| **Main scenario** | The system will add a like or dislike on a post |
| **Extensions** | • Like<br>• Dislike |

## 4.2. Sequence diagrams:

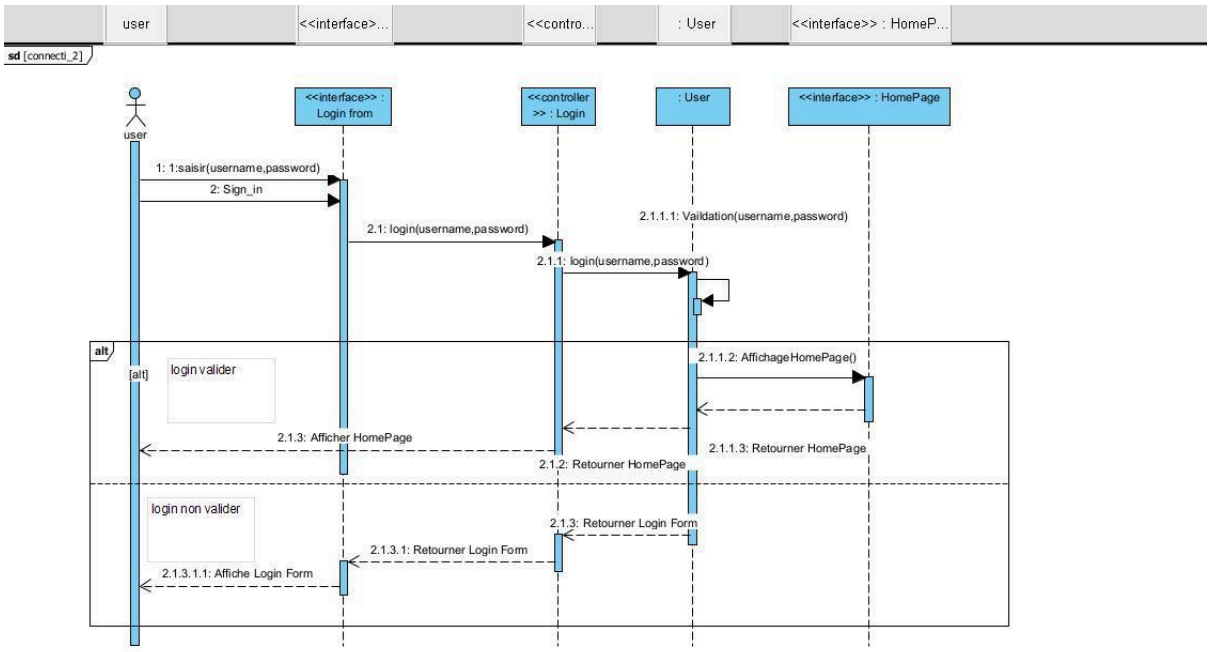
1.The sequence diagrams of “Publish Post”:



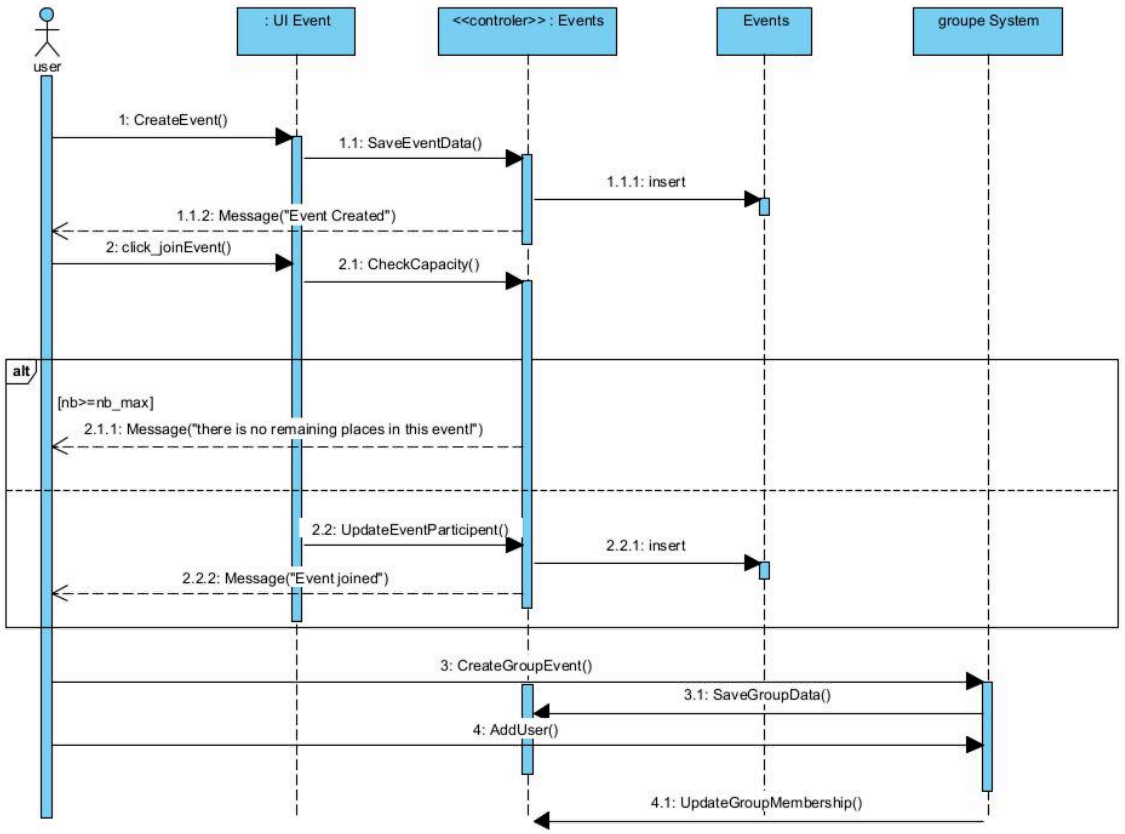

2.The sequence diagrams of “Register”:



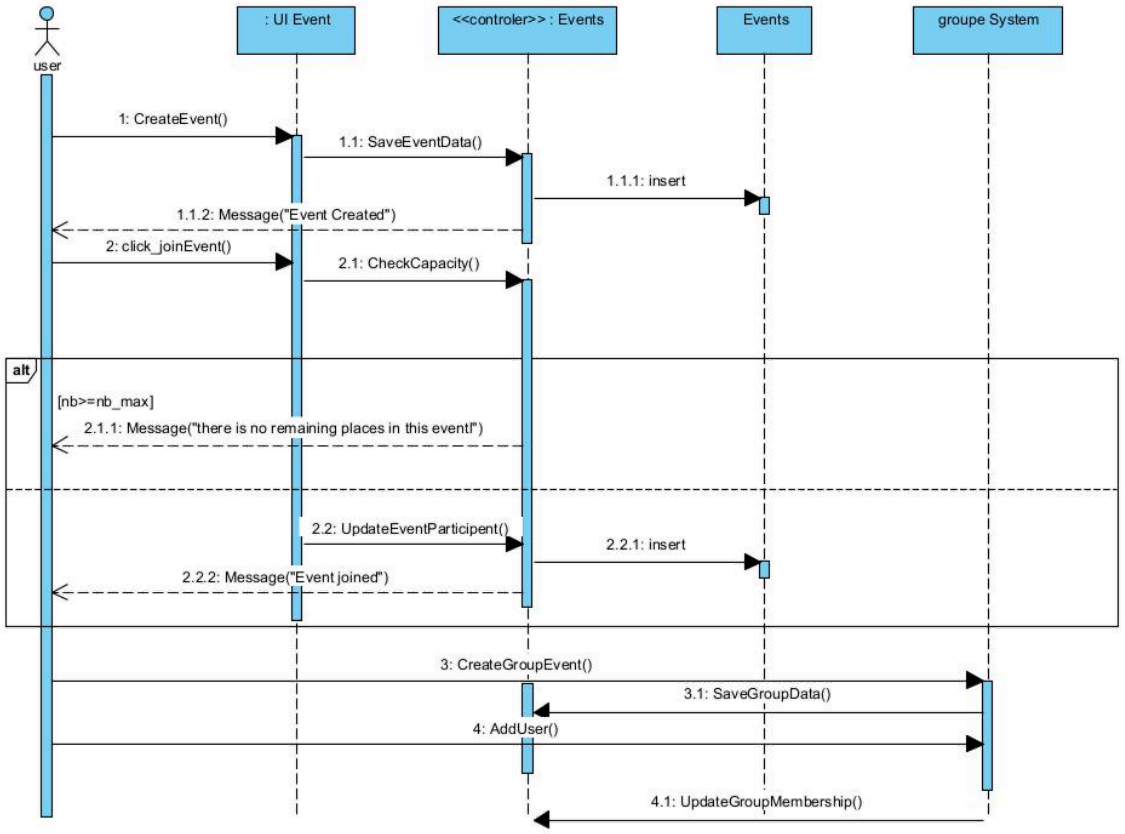
## 3.The sequence diagrams of “authenticate”:




## 4.The class and sequence diagrams of “Organize Event”:

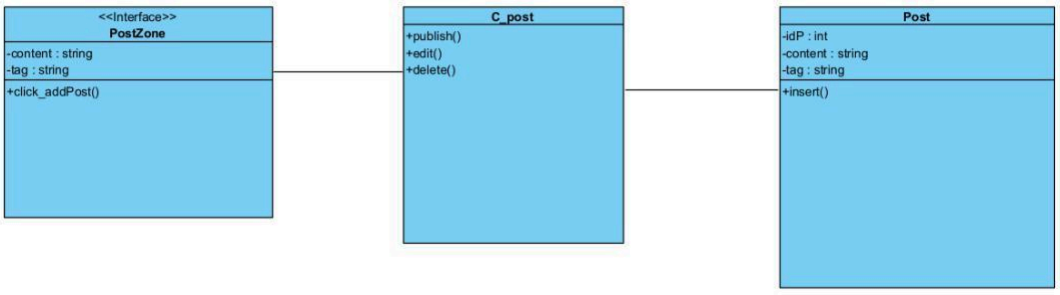


5.The sequence diagrams of “React”:

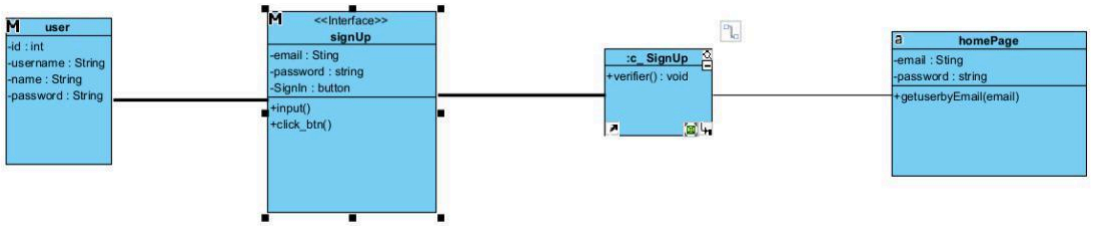



## 4.3. Classes diagrams:

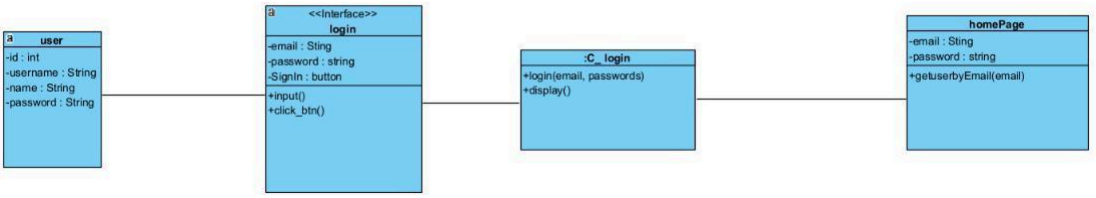
1.The class diagrams of “Publish Post”:



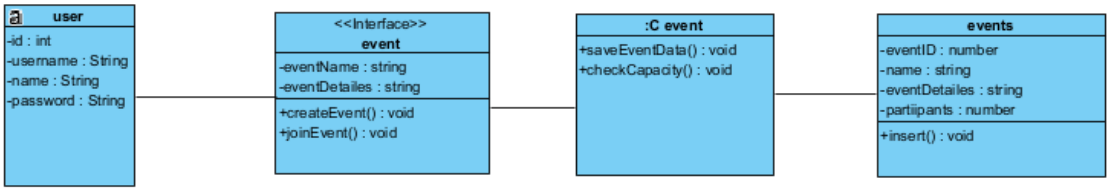
2. The class diagrams of “Register”:



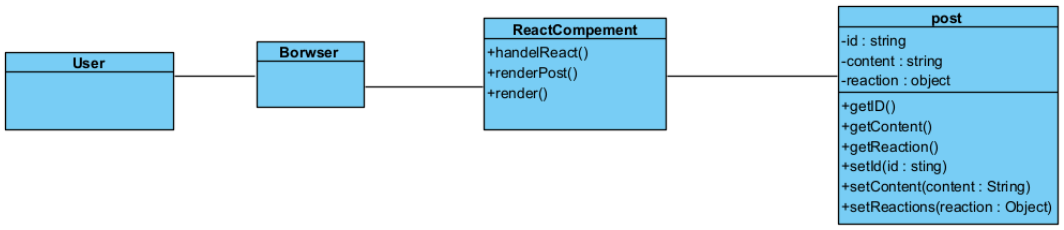

3. The class diagrams of “authenticate”:




4.The class diagrams of “Organize Event”:



5. The class diagrams of of “React”:




## 5.Conclusion:

In this chapter, we introduced our project “Evently” alongside of covering the importance of requirements specification for our software under the name of "Evently". We discussed identifying functional and non-functional requirements including reliability, error Handling, user-Friendly Design, security and maintenance and reusability, determining actors, and creating user case diagram in addition of sequence diagram and classes diagram.