

# IT314: Software Engineering

## Course Project Lab 6



## Group: 15 Conference Management System

| IDs       | Name                        |
|-----------|-----------------------------|
| 202001117 | PARVA BHARATKUMAR DHAMI     |
| 202001122 | MONPARA SUMIT RAJESHBHAI    |
| 202001128 | MEGHABEN RATHWA             |
| 202001132 | SONAGARA SARTHAK MOHANBHAI  |
| 202001154 | METKEL HARSHKUMAR ANANDBHAI |
| 202001162 | PADHIYAR KARAN HARESHBHAI   |
| 202001172 | PATEL VEDANT HARSHADBHAI    |
| 202001176 | MANGUKIYA JENISH MAHESHBHAI |
| 202001177 | PRAJAPATI PARTH SURESHBHAI  |

# Domain Analysis Modelling

## Boundary Objects:

### 1. Details Interface:

Display recent, current and upcoming conferences mentioned on our website. These details are visible even without authentication.

### 2. Authentication / Registration Interface:

#### 2.1. User Authentication:

2.1.1. The interface where the user can login and SingUp to the site to access all the functionalities. Without login user can only see the details

2.1.2. After authentication, the user can register to the particular Conference(s) (Only when the conferences are open for registration).

#### 2.2. Organization Authentication:

2.2.1. The Interface where the organizations can login and singUp to the site and manage their conferences.

2.2.2. After Authenticating, the organization can create or edit their conferences.

## Entity Objects:

### **1. Administrator:**

- Administrators can accept/reject the request by organizations to conduct their conferences on our site.

### **2. Organizer:**

- It contains coordinators names, their address, contact number, email address and other organizational information.

### **3. Attendee:**

- It contains username name, their address, contact information (number, email address), and other personal information.

### **4. User:**

- It contains information about roles and responsibilities of the user.

### **5. Publish:**

- Publishers can publish their papers on successful verification by the organizations into the conferences.

### **6. Conference:**

- It contains information about conferences' venues, timelines and fees.

### **7. Registration:**

- It contains information about conferences's registration by particular attendee.

## Control objects:

- 1. Registration as user:**

It controls the flow of registration data to the database.

- 2. Login as user:**

It controls the flow of validation of user information from the database.

- 3. Make payments:**

It controls the flow of payment while registering to the conference.

- 4. Registration into conference:**

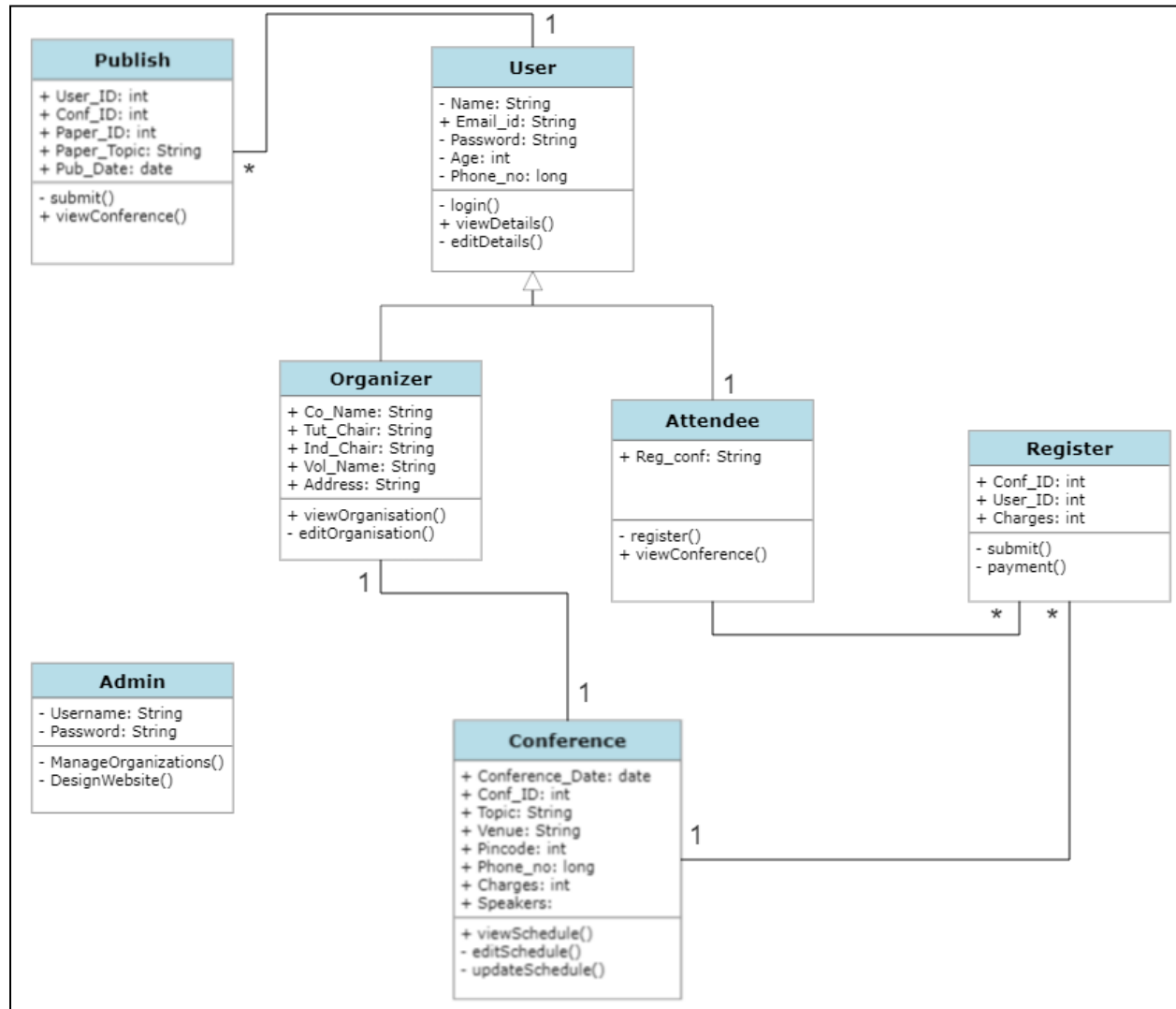
It controls the flow of user information to the database.

- 5. Send Certification:**

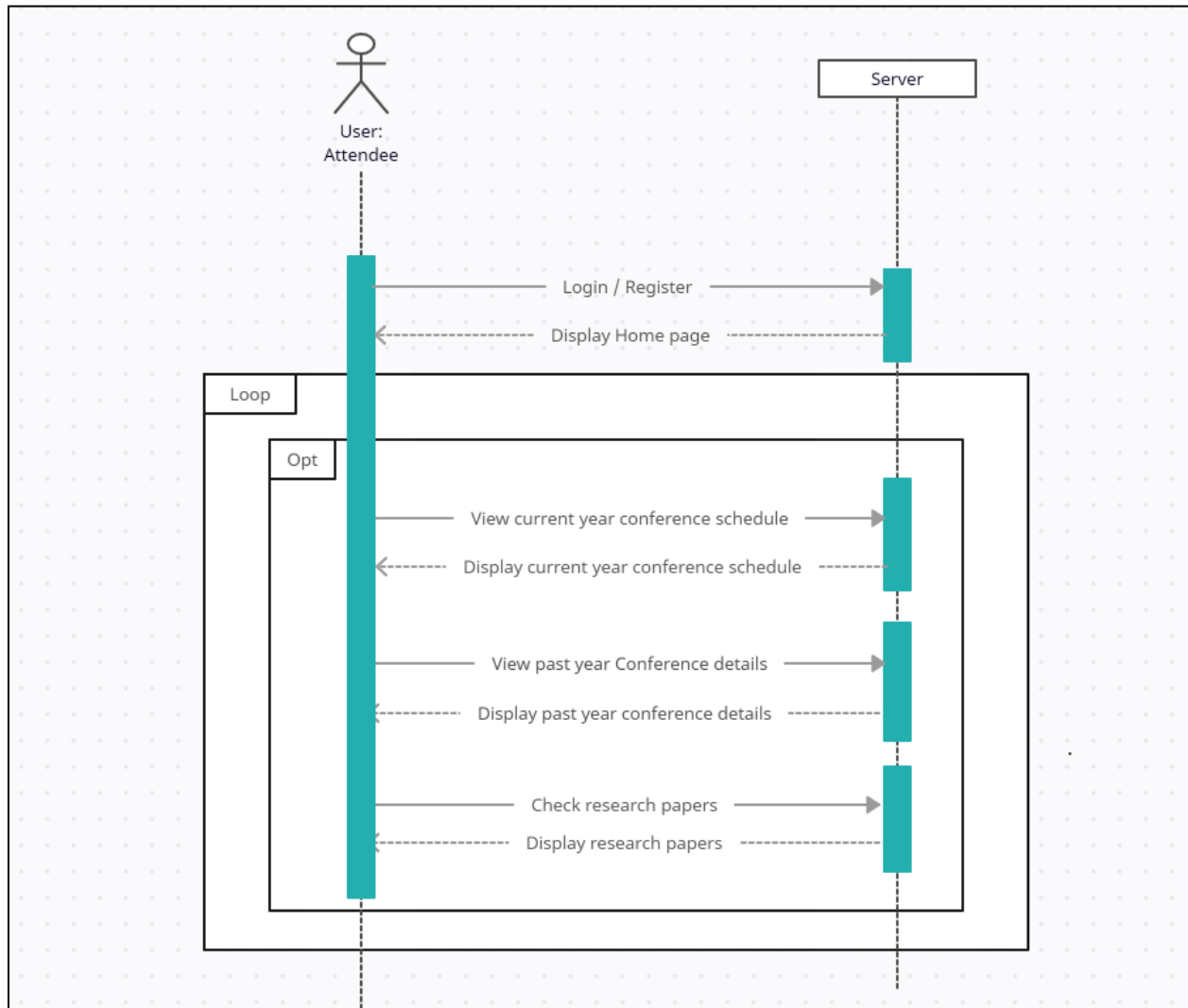
It controls the flow of sending certificates to attendees.

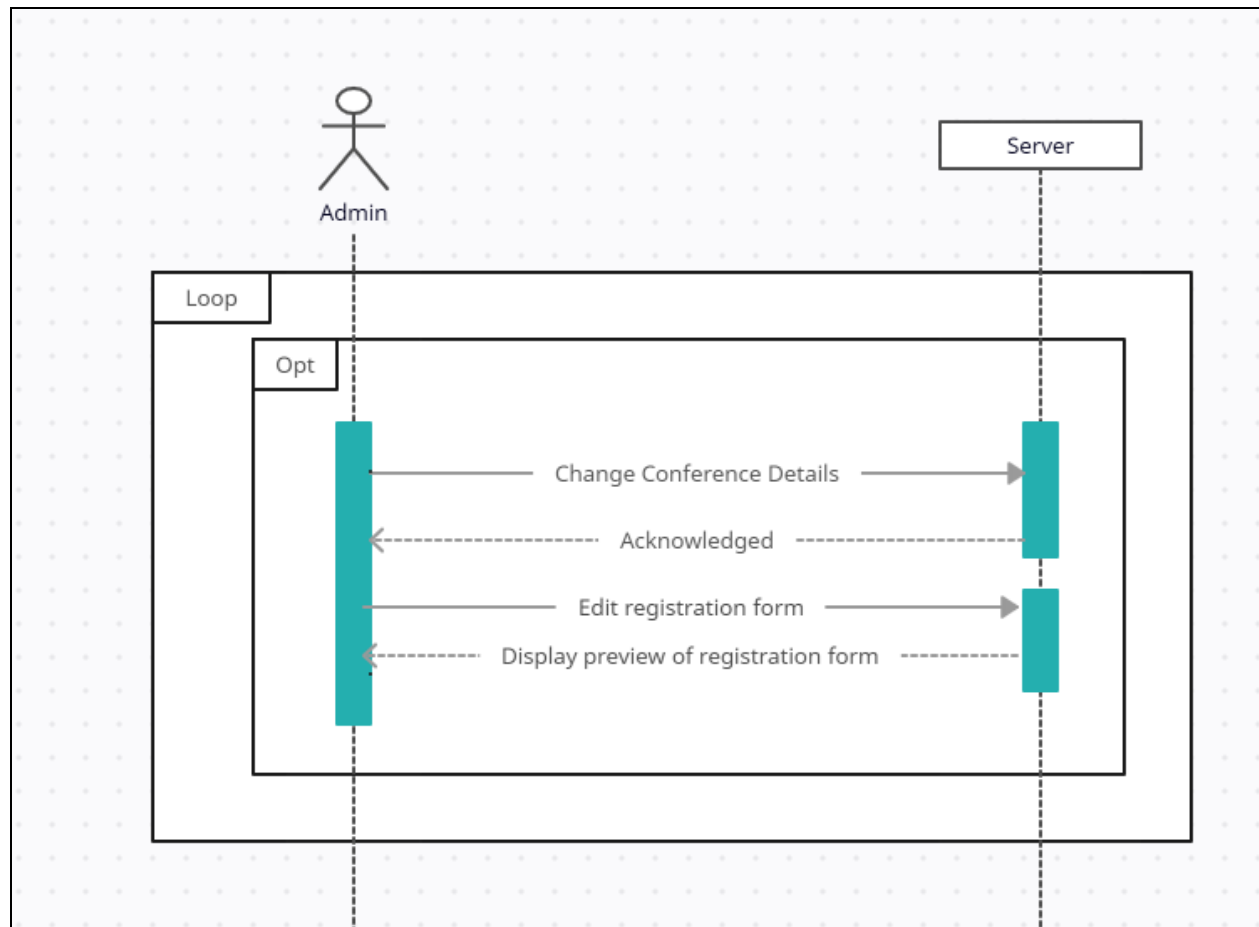
---

# Class diagram



## Sequence Diagram





# Design Goals:



## 1. Reliability:

A layout that is still functional in a variety of circumstances. It is a major concern for most of the developers as low reliability leads to a higher amount of risk .

## 2. Maintenance:

The purpose of maintenance is to increase the system's functionality, performance, and dependability as well as its capacity to adapt to shifting conditions and settings.

In order to keep the software meeting the demands of the users throughout time, software maintenance is an essential component of the software development life cycle.



### **3. Efficient:**

There should not be a delay in retrieving the user to get information as if we fail to give information to the user, the user was not able to attend the conference . Which might lead to poor experience to users and which leads to bad design.

### **4. User-Friendly:**

Some users are not aware of the system. so there must be good interface design and by seeing system users will be able to get an idea of how it works.

### **5. Flexibility:**

Flexibility refers to the system's capacity to react to uncertainty while maintaining regular operation.

Software flexibility, which indicates a program's capacity to change, is an important consideration when designing and creating software.

System needs to be flexible in order to update any functionality in the system that will not cause errors in working of the system.

### **6. Practicality:**

System that should be aimed at the group of users (audiences) for whom they are building.

---

# HIGH LEVEL SYSTEM DESIGN:

## ARCHITECTURE:

This application will use a client-server architecture. We will use a 3 tier application architecture that consists of a user interface , Business logic layer and Data layer. The data layer stores information, the business logic layer handles logic, and the user interface consists of the front end.

- **User Interface:**

This layer is distributed to a computing device using a web browser or a web-based application and is constructed with **HTML5** and **CSS3**. Application programme interface (API) calls are the primary communication between the user interface and other layers of the application.

- **Business logic layer:**

This layer is implemented using **Django** and contains the business logic that supports the application's core functions. This choice is because it allows the app to be used by many users and provides flexibility for future development.

- **Data layer:**

The data layer, sometimes called database layer, data access layer or back-end, is where the information processed by the application is stored and managed. We will be using the **NoSQL Database(MongoDb)** for this layer.

