



People's Democratic Republic of Algeria  
Ministry of Higher Education and Scientific Research

**LAARBI TEBESSI UNIVERSITY**

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

---

**Build a web site for representing  
international conferences and  
submission on them**

---

*Realized by*

FATHI ABDELMALEK  
CHOUKRI MAAMAR

*Framed by*

TAREK NOUIOUA

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Problematic . . . . .	1
1.2	Hypotheses . . . . .	1
1.3	Memory structure . . . . .	1
<b>2</b>	<b>Presentation of the project frameworks</b>	<b>2</b>
2.1	What we have to do . . . . .	2
2.2	Why we want to do that . . . . .	2
2.3	How to do that . . . . .	2
<b>3</b>	<b>Analysis and design (UML)</b>	<b>3</b>
3.1	Class Diagram . . . . .	3
3.2	Use case Diagram . . . . .	3
3.3	Sequence Diagrams . . . . .	4
3.3.1	For login . . . . .	4
3.3.2	For register . . . . .	5
3.3.3	For create user . . . . .	6
3.3.4	For update user . . . . .	6
3.3.5	For delete user . . . . .	6
3.3.6	For create conference . . . . .	7
3.3.7	For update conference . . . . .	7
3.3.8	For delete conference . . . . .	8
3.3.9	For create submission . . . . .	8
3.3.10	For update submission . . . . .	9
3.3.11	For delete submission . . . . .	9
<b>4</b>	<b>The implementation</b>	<b>10</b>
4.1	Implementation Technologies . . . . .	10
4.1.1	For front-end . . . . .	10
4.1.2	For back-end . . . . .	10
4.2	Interfaces of the site . . . . .	11
4.2.1	Index Page . . . . .	11
4.2.2	Conference Details page . . . . .	11
4.2.3	Notifications page . . . . .	12
4.2.4	Conference Creation page . . . . .	12
4.2.5	Submission Creation page . . . . .	13
4.2.6	Admin Panel . . . . .	13
<b>5</b>	<b>Conclusion</b>	<b>15</b>

## List of Figures

1	class diagram . . . . .	3
2	use case diagram . . . . .	4
3	login sequence diagram . . . . .	5
4	register sequence diagram . . . . .	5
5	create user sequence diagram . . . . .	6
6	update user sequence diagram . . . . .	6
7	delete sequence diagram . . . . .	7
8	create user sequence diagram . . . . .	7
9	update user sequence diagram . . . . .	8
10	delete sequence diagram . . . . .	8
11	create user sequence diagram . . . . .	8
12	update user sequence diagram . . . . .	9
13	delete sequence diagram . . . . .	9
14	Home page for guests . . . . .	11
15	Home page for registered users . . . . .	11
16	Conference Details page for its organizer . . . . .	12
17	Conference Details page for its organizer . . . . .	12
18	Noifications page . . . . .	13
19	Conference Creation page . . . . .	13
20	Submission Creation page . . . . .	14
21	Administration panel look . . . . .	14

# 1 Introduction

National and international scientific conferences are an important event for universities and researchers from different parts of the world, so it is necessary to facilitate the process of publishing and accessing these conferences.

## 1.1 Problematic

Most or all universities of the world are publishing their conferences in custom web pages for each one, or they collect them in one website like 10times.com and ieee.org. Meanwhile any scholar how want to publish his paper or submit in a conference, he utilize another different website like easychair.org. Our work is related to finding a solution to the problem raised by answering the following questions:

- Why not they all publish everything from conferences to papers in one place ?
- How to let scholars and universities contact with each other from one place ?
- What is the best solution for this problem and how to achieve that ?

## 1.2 Hypotheses

Among the proposed solutions, we find that one of them relies on creating a website for publishing conferences and requesting registration in them, where the person in charge of the conference (university or organization) publishes the necessary information about the conference such as its name, date, and participation price... while any researcher or student can request to participate in it, as he sends his research to the officials in charge of the conference and is waiting for it to be accepted by them.

## 1.3 Memory structure

This article contains the following chapters:

- **Presentation of the project frameworks :** Chapter to figure out the problem and its solution in details.
- **Analysis and design (*UML*) :** Chapter to introduce the *UML* diagrams that we used to analysis the project and figure out his functions.
- **The implementation :** Chapter to view the technologies that we used in making the site, and the implementation of our site (pictures from the website itself).

## **2 Presentation of the project frameworks**

### **2.1 What we have to do**

Our project is based on a website of representing conferences and submission to them, when the organizer of the conference put information about his conference like title, description... And any one can see this conference in the web site, but only registered scholars and researchers can submit to it via sending their information to this organizer and wait to his acceptance and confirmation.

### **2.2 Why we want to do that**

The main reason is that there is no website to represent a conference and let scholars submit in it at the same time, so the conference is in a separated website like ieee.org or in a specific domain like icrami.faox.dk. Meanwhile the scholar publish his paper in another website like easychair.org, so there is no platform in the network that let them all publish everything in the same place.

### **2.3 How to do that**

We want to create a website that let any organizer of a conference to publish his information like the title, important dates (conference date, submission deadline, confirmation deadline, payment deadline), place, price... etc. Meanwhile any scholar or researcher can submit to it directly from the same web site, by sending to the organizer his information (name, email to contact, abstract of his work, and maybe the work itself, the authors... etc). but first they should be registered in this site.

There is some rules to submit, the scholar should demand a submission in a conference before the end of his submission deadline, once the its end, he couldn't submit, update or delete his submissions for that conference. The organizer can't accept it immediately, he should wait until the beginning of confirmation deadline, but he can see and read it at any time he want, when he accept a scholar, this scholar will get a notification and an email to paid for the submission before end of payment deadline, or he will pay a larger price, he take a screenshot of the payment ticket and send it to the organizer and wait his confirmation.

To publish a conference or submit to it, the user should be registered in the site, he register with the email, user-name, first and last name at the beginning, but he can complete the other information later via access his profile page and press (edit), he get a form to add his (birth date, sex, country, work place, degree, specialty, website if any and an profile image), all this information are optionally, but the organizer can either accept or refuse the submission depends on his rules. When a user submit to a conference he put the information below (his first and last name, his email, name of the paper, an abstract of his work, the paper itself and the authors of it), all this information are required expect the article itself, it depends on the conference rules, once the organizer read it and the confirmation deadline begin, he can accept or refuse it at any time, in case of acceptance, the submitter receive a notification in the site and an email of acceptance, and he should pay for the conference and send the payment ticket to the organizer who will confirm his submission after receiving it.

The conference organizer should put many information in the description text area including his payment method, schedule of the conference and many other information.

### 3 Analysis and design (UML)

We have utilized the following graphs: class, use cases and sequence diagram of each process in our website.

#### 3.1 Class Diagram

In Figure 1 we diagram to introduce classes of the project and their associations with each others.

There is tow types of user (simple user and superuser), only the superuser can access the admin panel like the shown in Figure 15, any user can create conference or submit to it, and can accept, confirm or refuse the submissions, when a submission is created or modified (updated, accepted, confirmed, refused) a notification will created and sent to the organizer or the submitter.

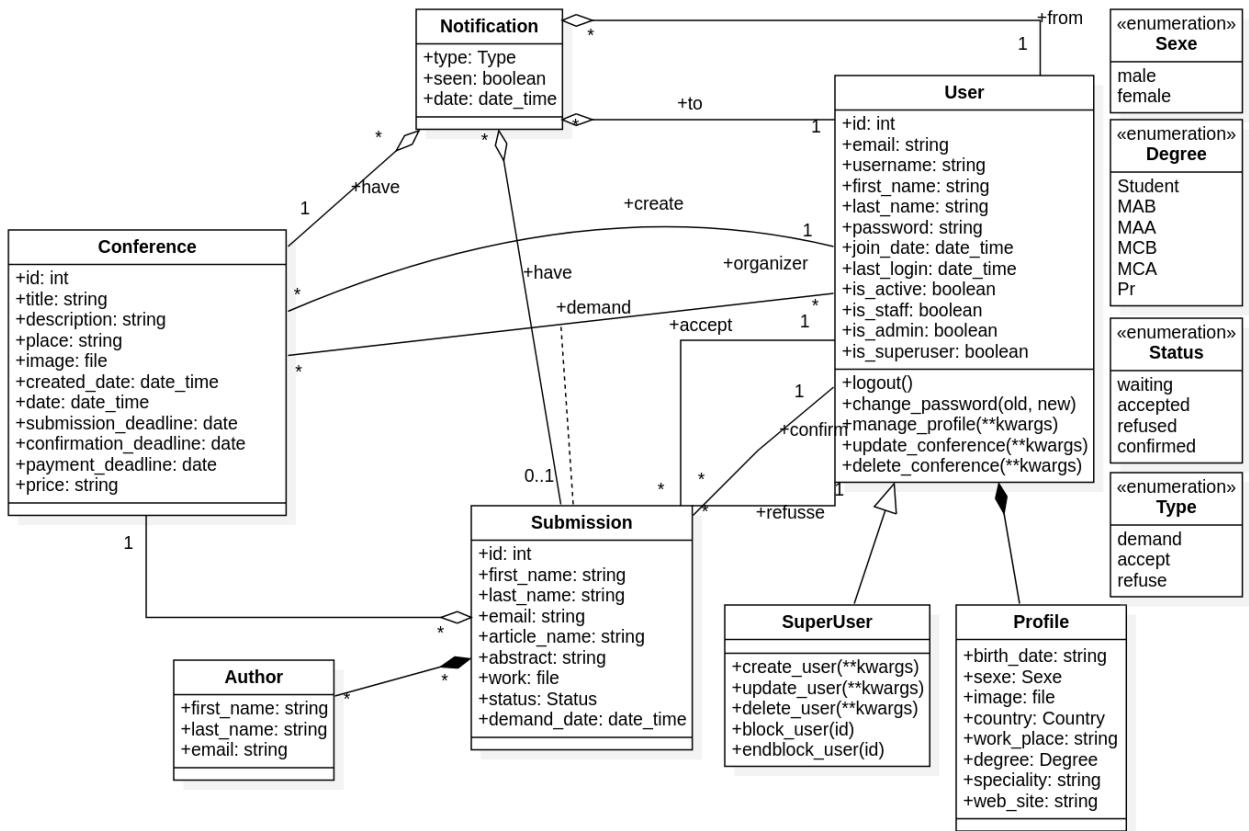


Figure 1: class diagram

#### 3.2 Use case Diagram

In Figure 2 we diagram to introduce what can every actor do in this project.

there is a guest who can register as new user or login, and a user (simple user) who can create, update, delete conferences and submission, or accept, confirm, refuse others submissions, and there is the superuser who can do everything the user can do additionally to manage the database of the site (manage users and conferences and everything).

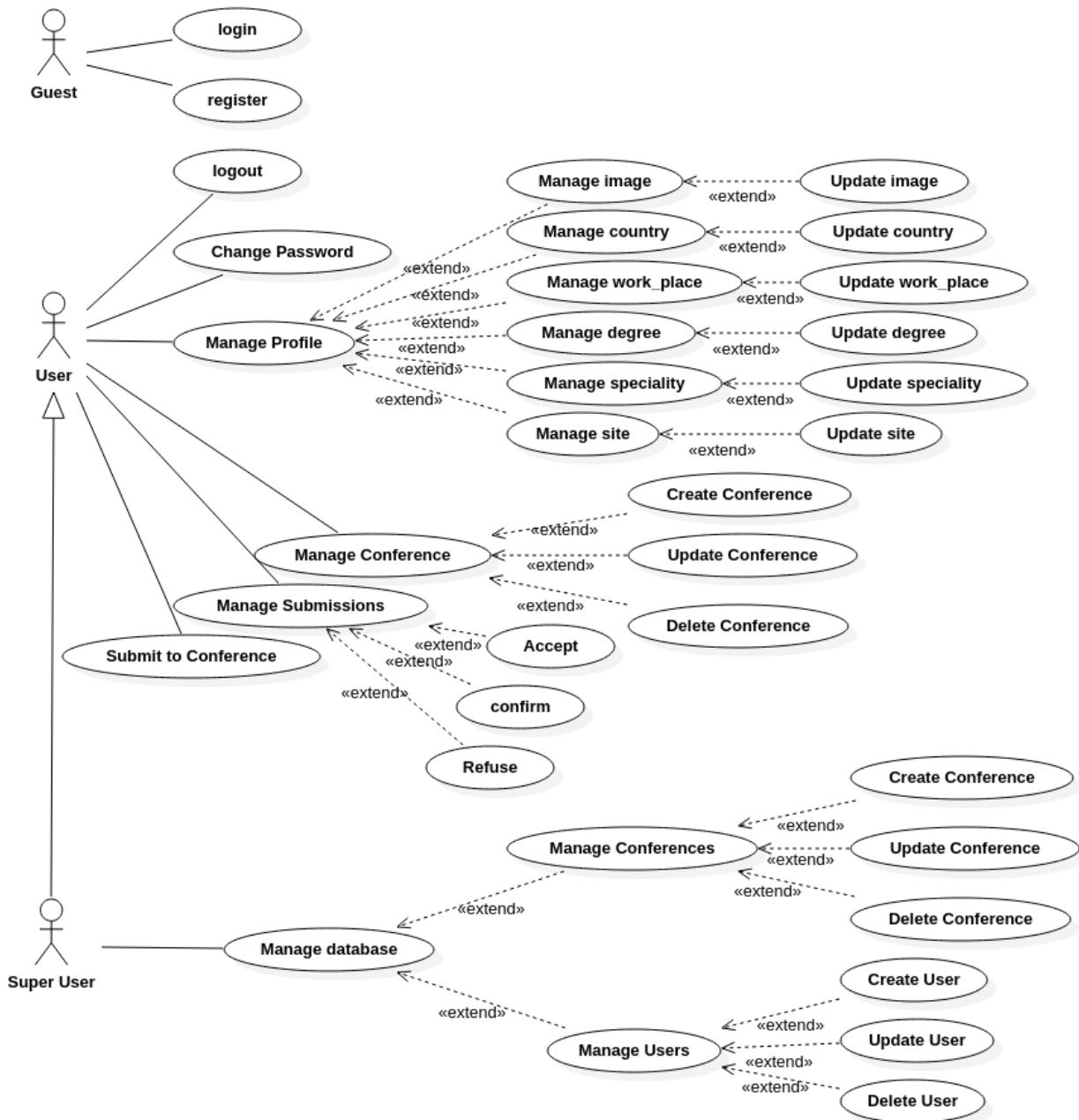


Figure 2: use case diagram

### 3.3 Sequence Diagrams

Diagrams to figure out how each process is working behind the scene.

Because we have MVC design pattern in Django, all the sequences diagrams will contain a view, controller and a model for the process.

#### 3.3.1 For login

In Figure 3 we have how the user can login.

The guest fill login form and submit it, the view send his data to the controller who valid it first, and then verify the database, in case of valid data it will redirect him to profile page, or he return errors messages to the view.

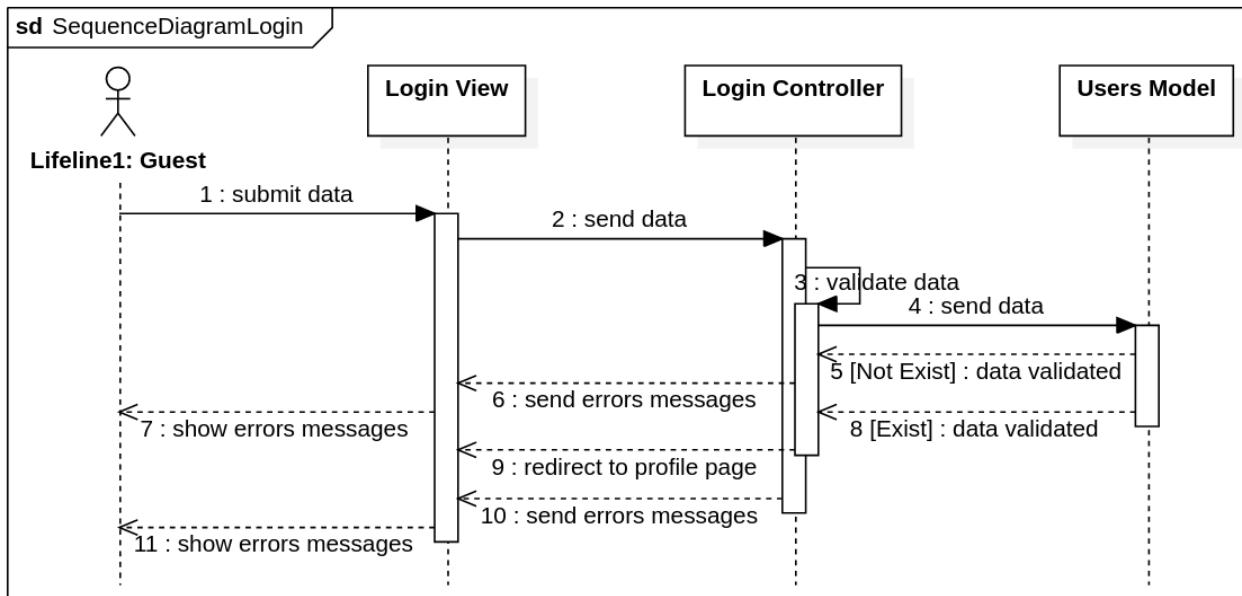


Figure 3: login sequence diagram

### 3.3.2 For register

In Figure 4 we have how to register as a new user.

The guest fill registration form and submit it, the view send his information to the controller who validate it and verify if exist in database, if valid he will create new user and redirect him to profile page, if not he will return errors messages to the view.

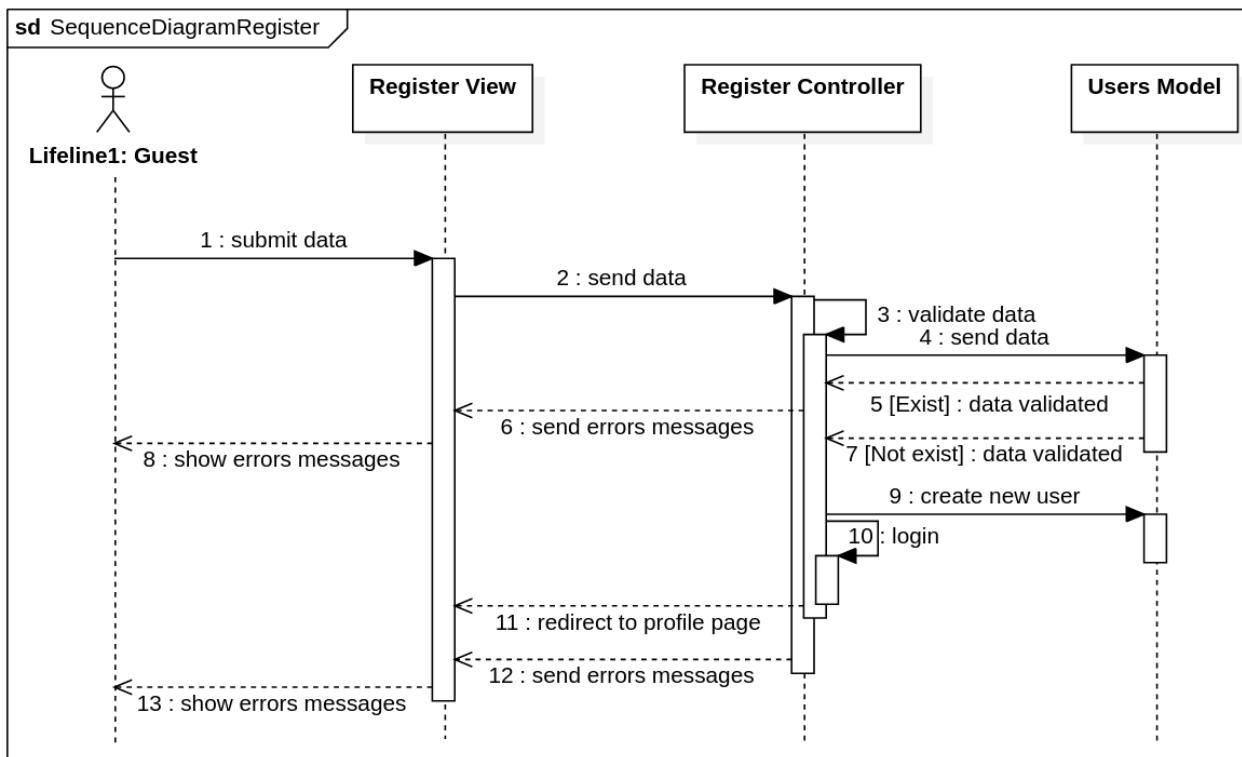


Figure 4: register sequence diagram

### 3.3.3 For create user

In Figure 5 we have how to create new user.

The controller validate the data and verify the database(users model), in case of valid data he will create a new user successfully, or return errors messages to the view.

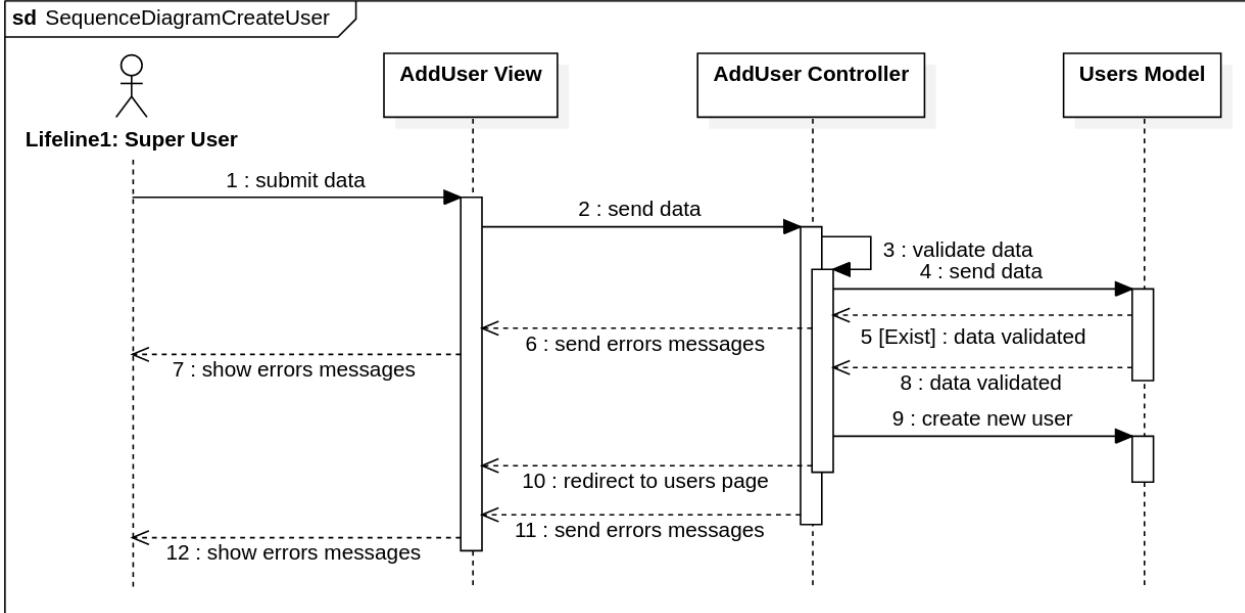


Figure 5: create user sequence diagram

### 3.3.4 For update user

In Figure 6 we have how to update user information.

The controller validate the data and update the user in case of valid data, or return errors messages to the view.

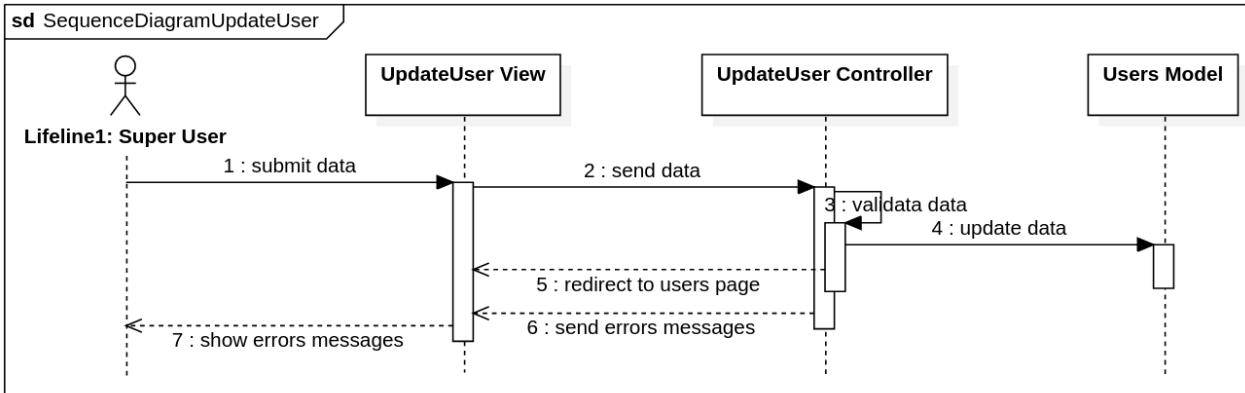


Figure 6: update user sequence diagram

### 3.3.5 For delete user

In Figure 7 we have how to delete a user from the data base.

The controller delete the user directly from the database and refresh the page.

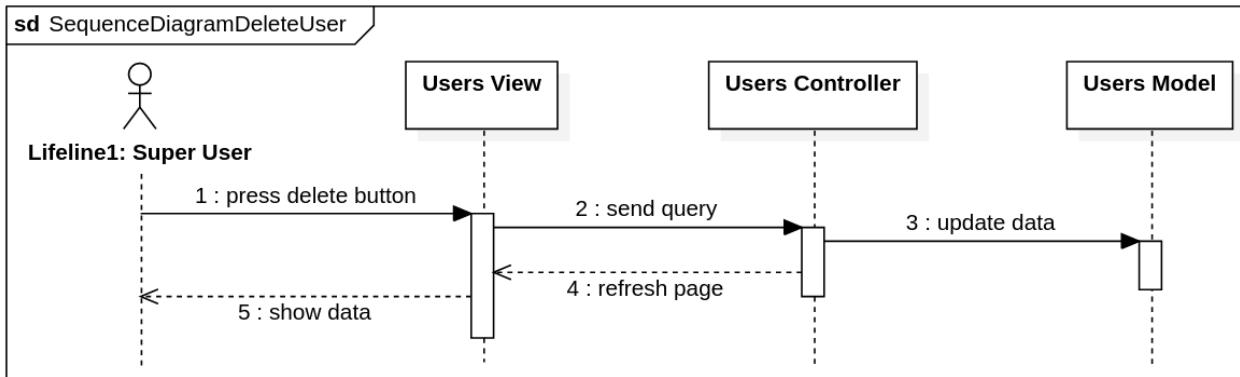


Figure 7: delete sequence diagram

### 3.3.6 For create conference

In Figure 8 we have how to create new user.

The controller validate the data of the conference that sent from the user to the view, and verify the database for duplicated data, then he will create new conference (with INSERT query in SQL Language) and redirect the user to the previous page or return errors messages to him.

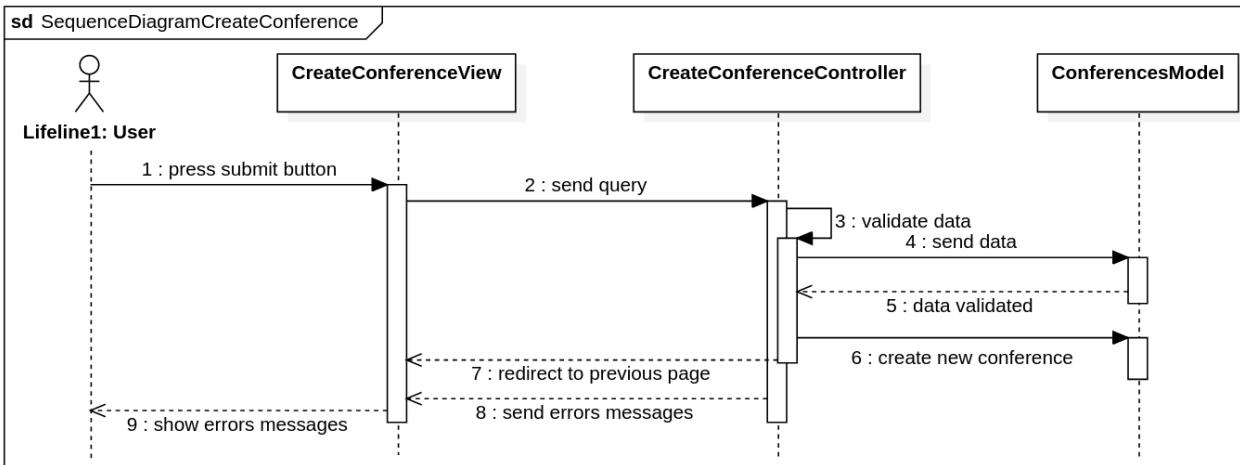


Figure 8: create user sequence diagram

### 3.3.7 For update conference

In Figure 9 we have how to update user information.

The controller validate the data of the conference that sent from the user to the view, and verify the database for duplicated data, then he will update this conference with the new data (with ALTER TABLE query in SQL Language) and redirect the user to the previous page or return errors messages to him.

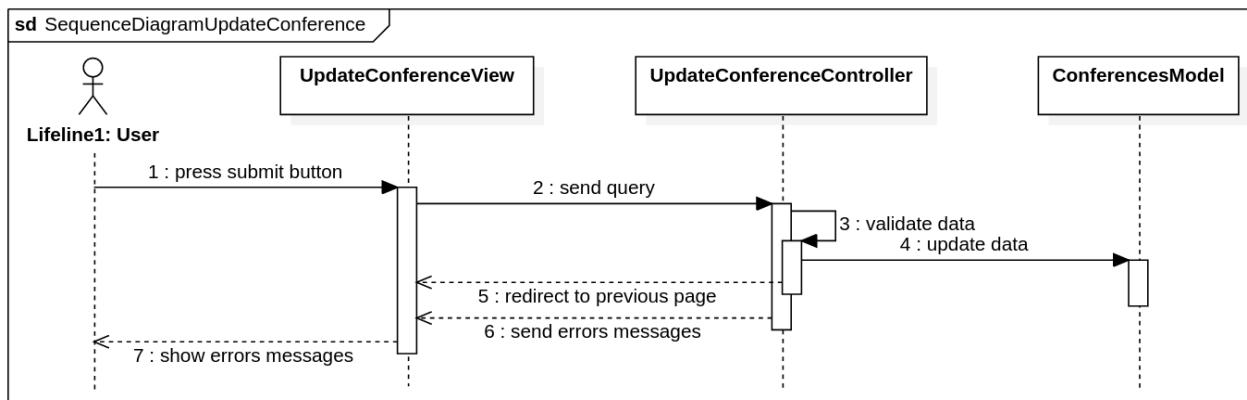


Figure 9: update user sequence diagram

### 3.3.8 For delete conference

In Figure 10 we have how to delete a user from the data base.

Same as user delete process.

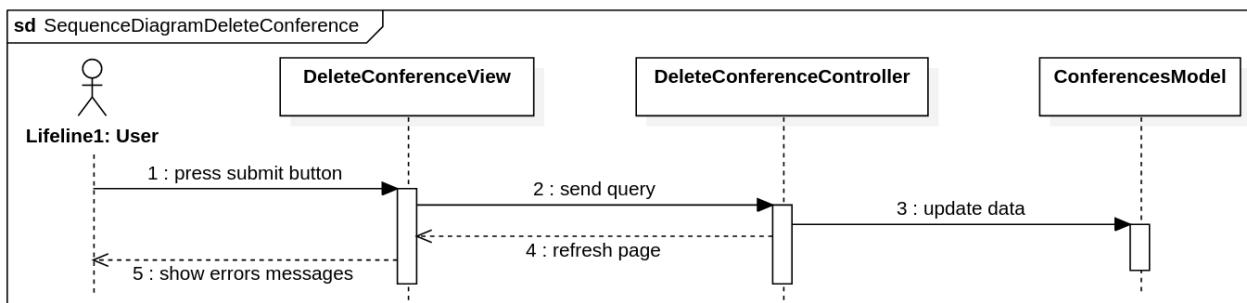


Figure 10: delete sequence diagram

### 3.3.9 For create submission

In Figure 11 we have how to create new user.

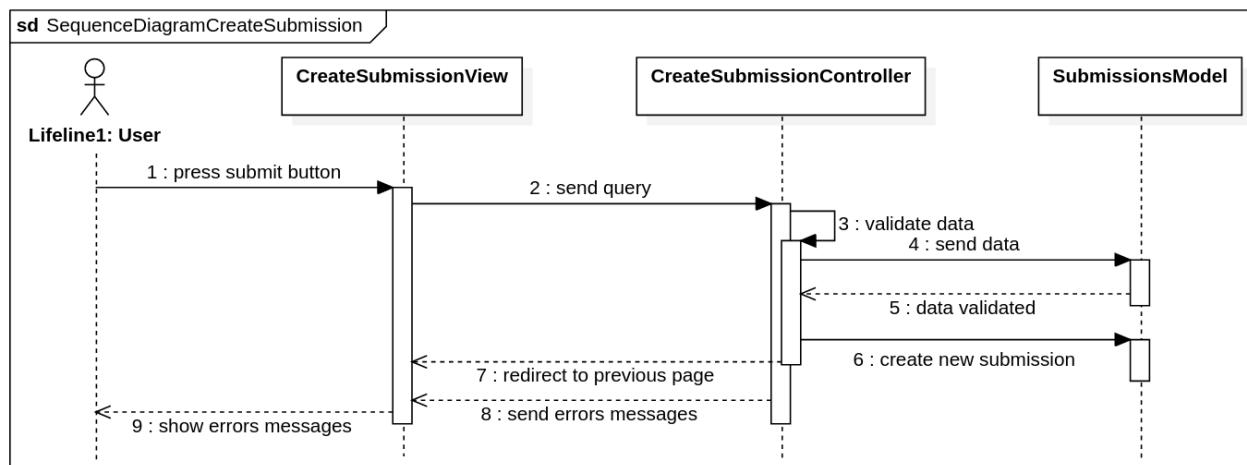


Figure 11: create user sequence diagram

### 3.3.10 For update submission

In Figure 12 we have how to update user information.

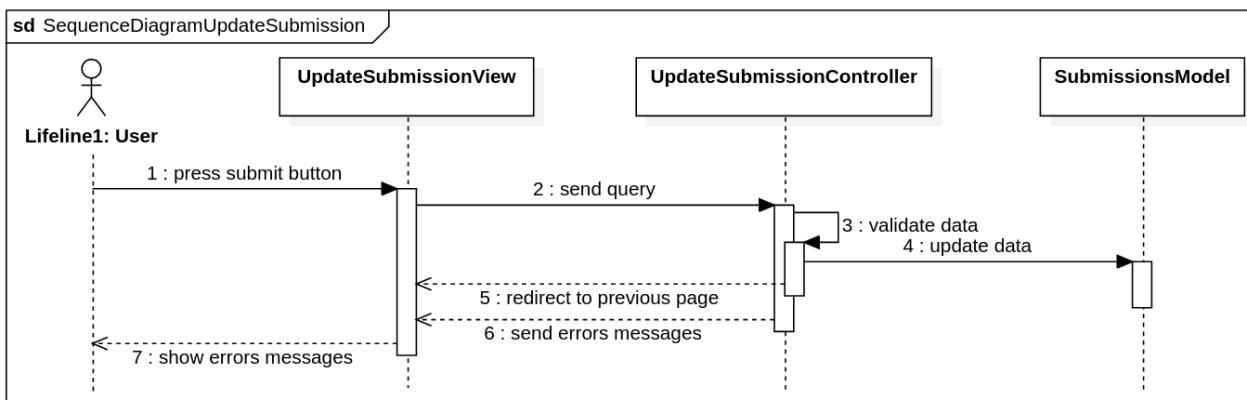


Figure 12: update user sequence diagram

### 3.3.11 For delete submission

In Figure 13 we have how to delete a user from the data base.

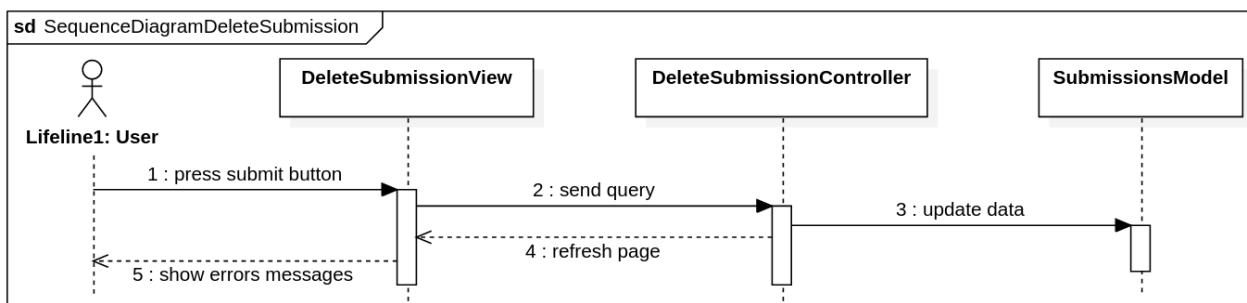


Figure 13: delete sequence diagram

## 4 The implementation

There are many technologies that allow building web applications, and they are divided into two main parts, the front-end part such as HTML, CSS, JS ..., and the back-end part such as PHP, NodeJS ..., in addition to databases, but we will depend in our project on the following:

### 4.1 Implementation Technologies

#### 4.1.1 For front-end

**HTML** (Hyper-Text Markup Language) is the standard mark-up language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.[1]

**CSS** (Cascading Style Sheets) is a simple mechanism for adding style (e.g., fonts, colors, spacing) to Web documents.[2]

**Bootstrap** The world's most popular front-end open source toolkit, featuring Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful JavaScript plugins.[3]

#### 4.1.2 For back-end

**Django Web Framework** Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.[4]

#### why django ?

- It's a Python Language.
- Easy to learn.
- Fast and secured.
- Built-in administration interface.
- Framework able to customization.
- large community.
- MVC (model view controller) design pattern.
- Built-in ORM (Object Relational Mapping) for databases.
- and much more ...

**SQLite** SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine. SQLite is the most used database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day..[5]

## 4.2 Interfaces of the site

### 4.2.1 Index Page

In Figure 14 we see home page of the site for the user who haven't logged in.

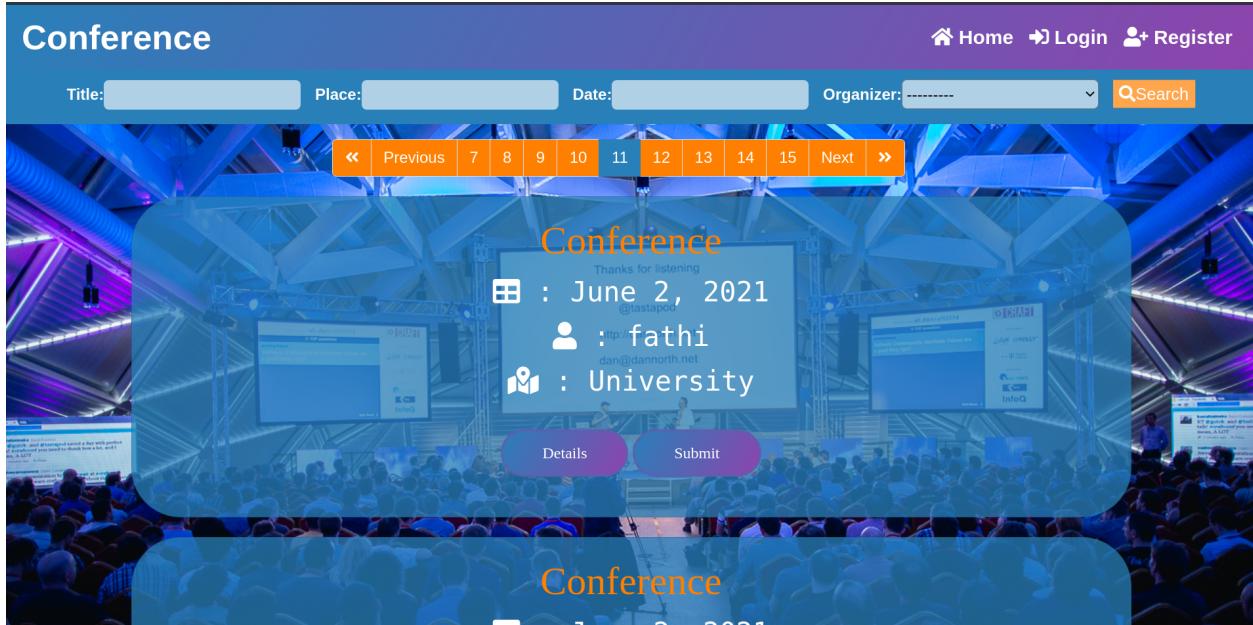


Figure 14: Home page for guests

In Figure 15 we see home page of the site for the user who have logged in.

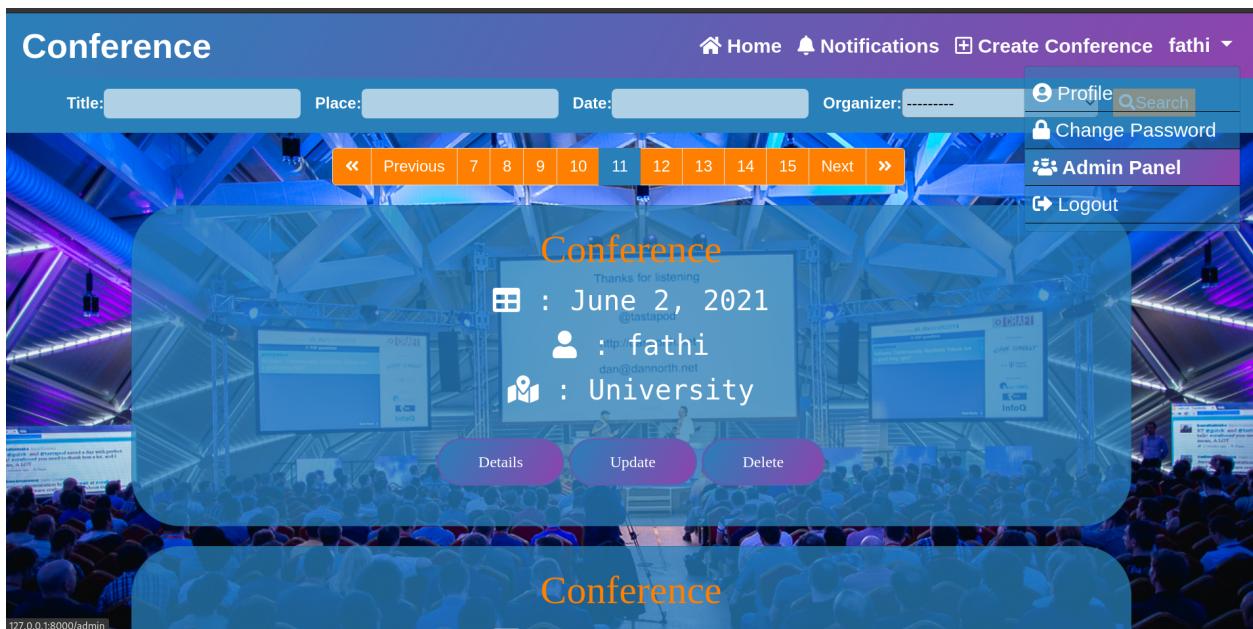


Figure 15: Home page for registered users

### 4.2.2 Conference Details page

In Figure 16 we see a conference details page when the logged in user is the organizer.

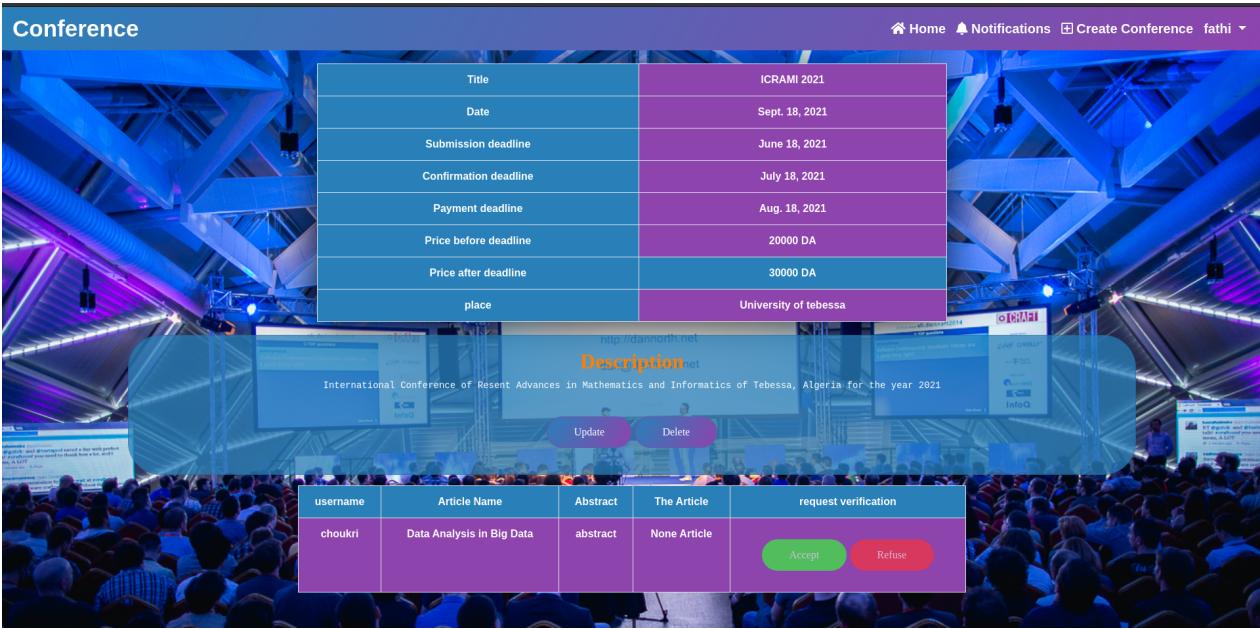


Figure 16: Conference Details page for its organizer

In Figure 17 we see a conference details page when the logged in user isn't the organizer.

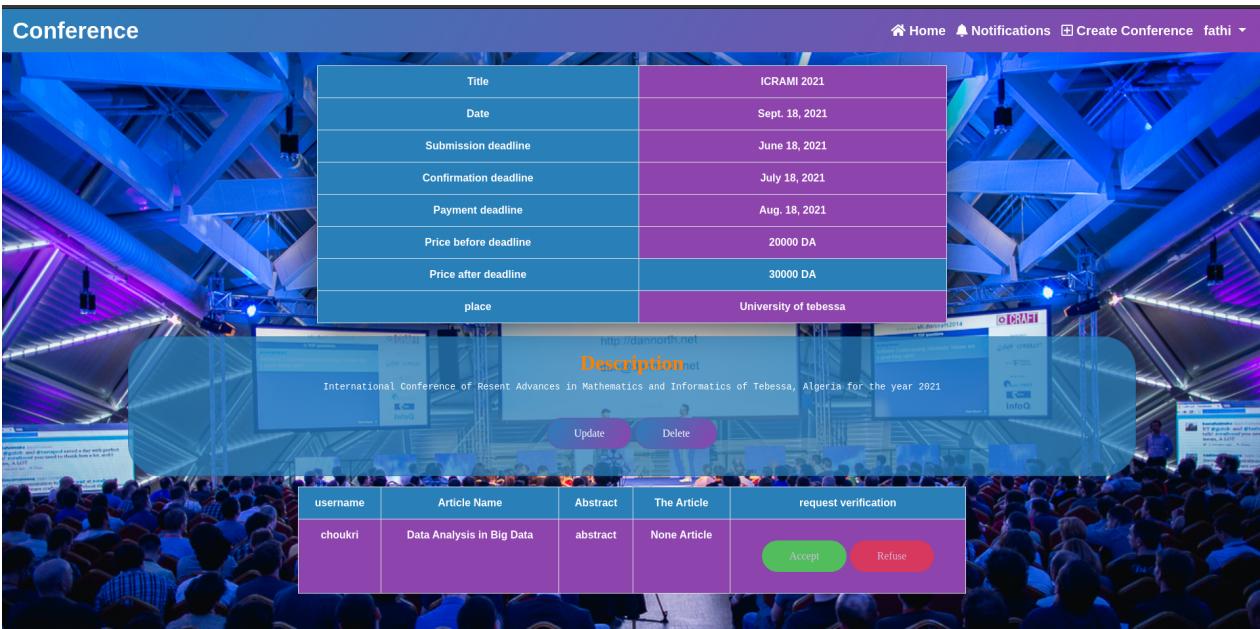


Figure 17: Conference Details page for its organizer

#### 4.2.3 Notifications page

In Figure 18 we see notifications page.

#### 4.2.4 Conference Creation page

In Figure 19 we see the page of conference creation.

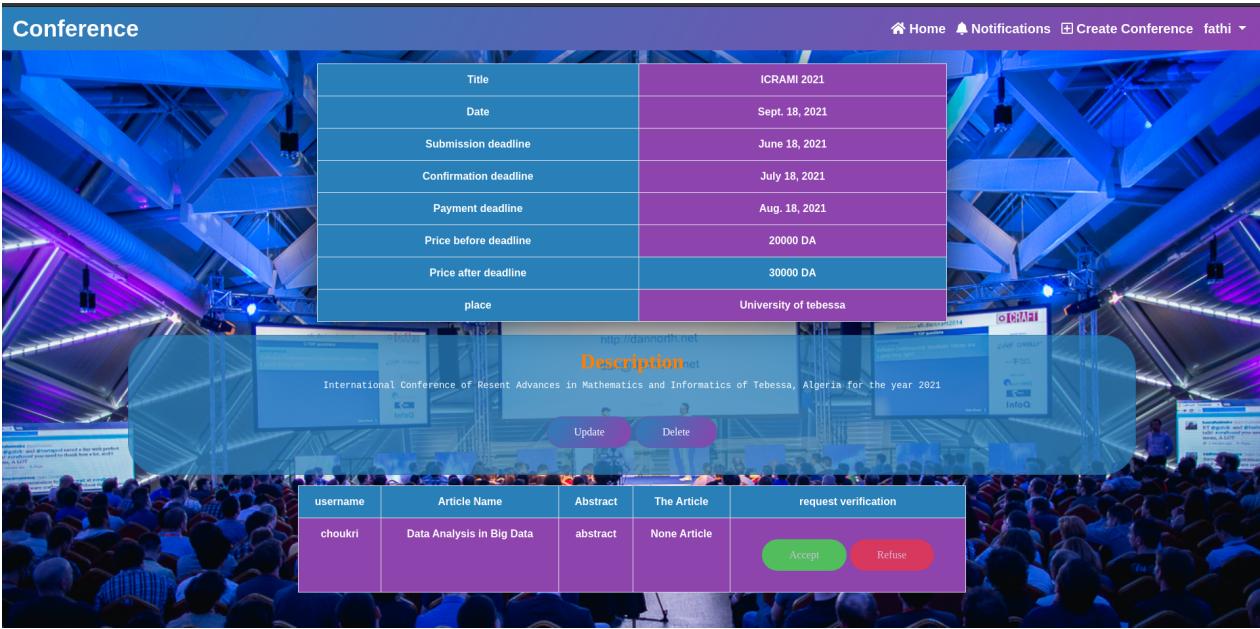


Figure 18: Notifications page

The screenshot shows a "Create new Conference" form. The fields include:

- Title: (empty)
- Description: (empty)
- Conference Date: (empty)
- Submission Deadline: (empty)
- Confirmation Deadline: (empty)
- Payment Deadline: (empty)
- Price before deadline: (empty)
- Price after deadline: (empty)
- Place: (empty)

Each field has a dropdown arrow icon to its right. The background of the form features a blue-toned image of a conference hall.

Figure 19: Conference Creation page

#### 4.2.5 Submission Creation page

In Figure 20 we see the page of submission creation.

#### 4.2.6 Admin Panel

In Figure 21 we see a page from administration panel.

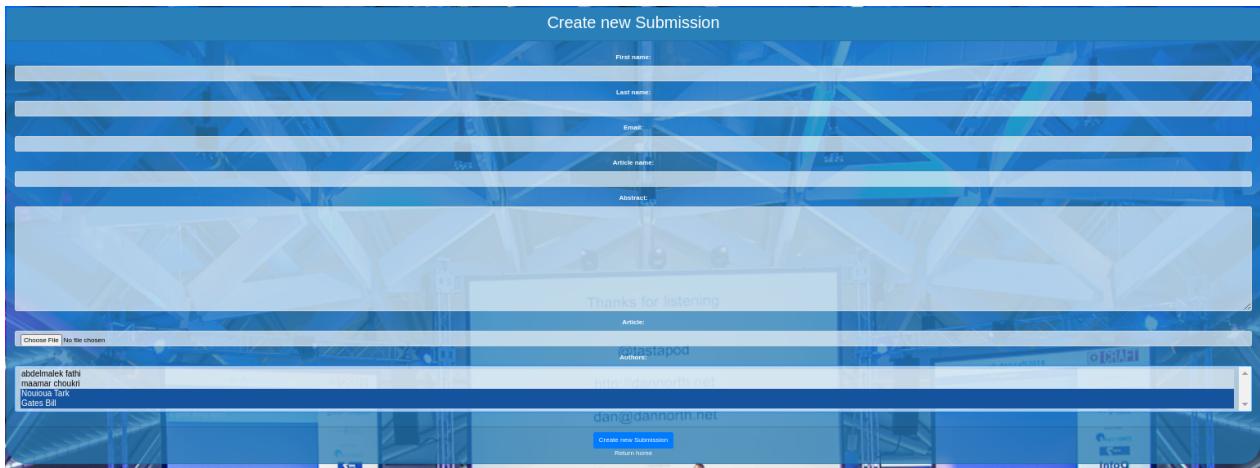


Figure 20: Submission Creation page

Figure 21: Administration panel look

## 5 Conclusion

## References

- [1] <https://en.wikipedia.org/wiki/HTML>
- [2] <https://www.w3.org/Style/CSS/Overview.en.html>
- [3] <https://getbootstrap.com/>
- [4] <https://www.djangoproject.com/>
- [5] <https://www.sqlite.org/index.html>