Subject:

Video Conference Chatting

Realized by:

FARABY Youssef

Supervised by:

MADA Badr

ELMARAI Oussama

TALEB Tarik

Abstract

This document is the summary of a work all along for Erasmus exchange program of the Aalto university.

The project consists of implementing a web application for Video Conference Chatting. The application will offer a communication and collaboration platform between all the members of the group.

In order to carry this project out, we started by analyzing and designing, then we moved to the implementation phase with the suitable technologies.

Table Of Content

table of figures

1-Introduction

Generally, people work with synchronous solutions for visual and distant interactions, the communication is mandatory, and a tool to exchange the information is surely needed, this collaborative tool will make communication easy and accessible.

In this project, we will realize an application for a Video Conference Chatting, and thought this report, we will summarize the entire work done in the last 4 weeks.

In the first chapter, we will put a major conception and analysis and describe the tools used for this work, in the second chapter, we will describe the realization of the application.

2-Analyze and Conception

The first step of the conception is to analyze the situation to take into account the requirements, this chapter allows us to identify all the features of our application for each type of users by identifying the functional needs. These functional are modeled by appropriate UML diagrams.

An actor represents a role played by an external entity which interacts directly with the system, He can consult and / or directly modify the state of the system, In the case of our system, we have identified the following actors:

- User: a person that gets access to the application and chooses either to be an admin or to be just a member.

- Admin: creates a room and become its admin he can invite/ban members and gives admin privileges to members, also can broadcast/stop their webcams.

- Member: users that are waiting the invitation from an admin to join his room.

A functional need is an action that a system must be able to perform, in this part, we will list the different needs that the system must provide. In our context the system must allow the following tasks:

- Create a room: the admin can create his room with all the privileges (send invitation, ban members…...).
- Go to salon: When a member wants to assist a conference, he will wait for the invitation from an admin in the salon.
- Add members: the admin has the privilege to add members in his room
- Ban members: the admin has also privilege to ban members.
- Send call: the admin can start the conference.
- Grant privilege: the admin affects admin privileges to members of the group.
- Broadcast and stop webcam: The admin has the privilege to broadcast and stop webcam.
- Accept an invitation: the members accept invitation for joining to the room.

We collect all identified use cases, in a general diagram as in the figure.

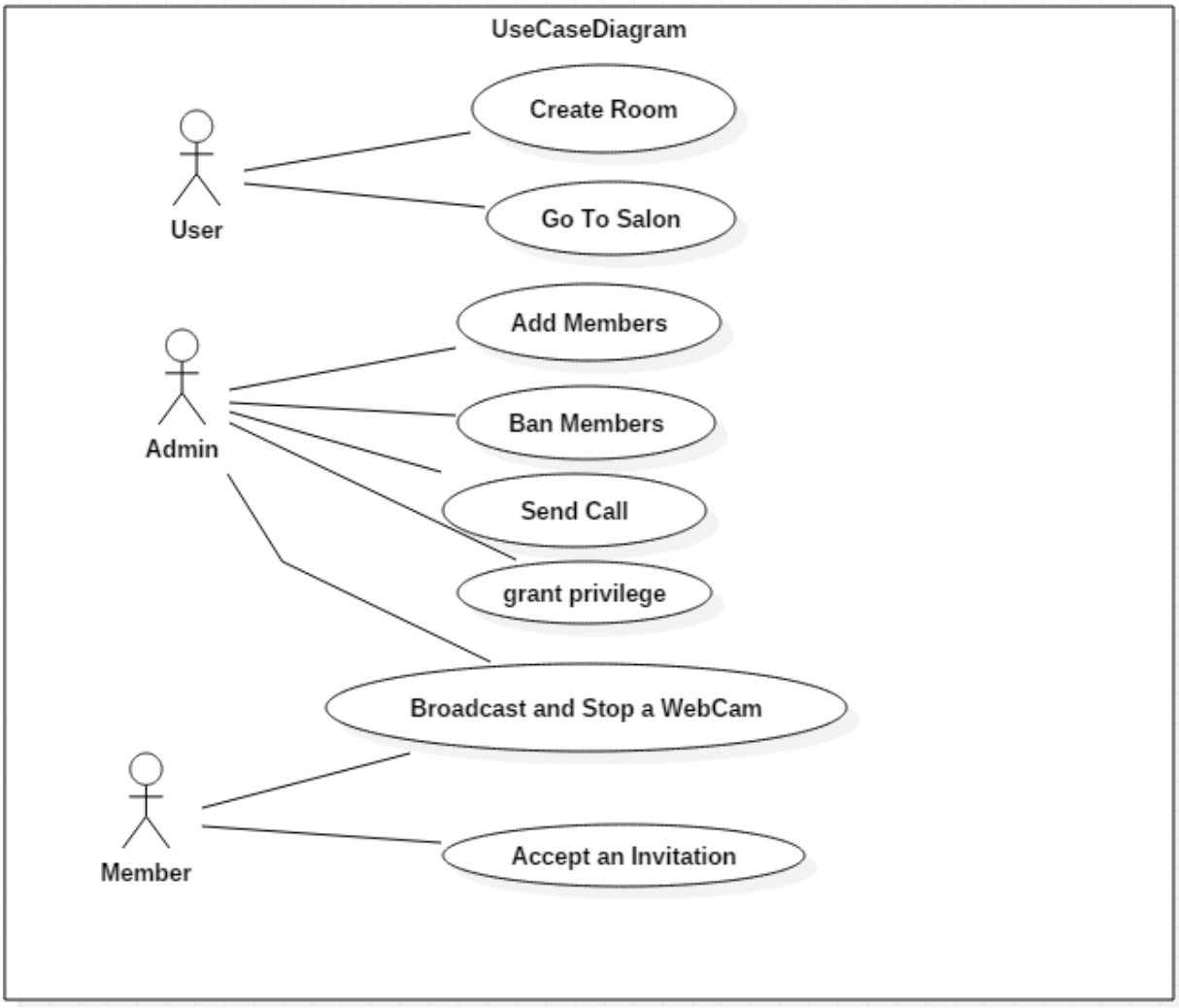

Figure 1 Use Case Diagram

-Sequence diagrams

With the help of the following sequence diagrams, we will illustrate the flow of using the application.

The user logs in with a login form, the system verifies the given information, and send the user to a page where he can create a room and display its page.

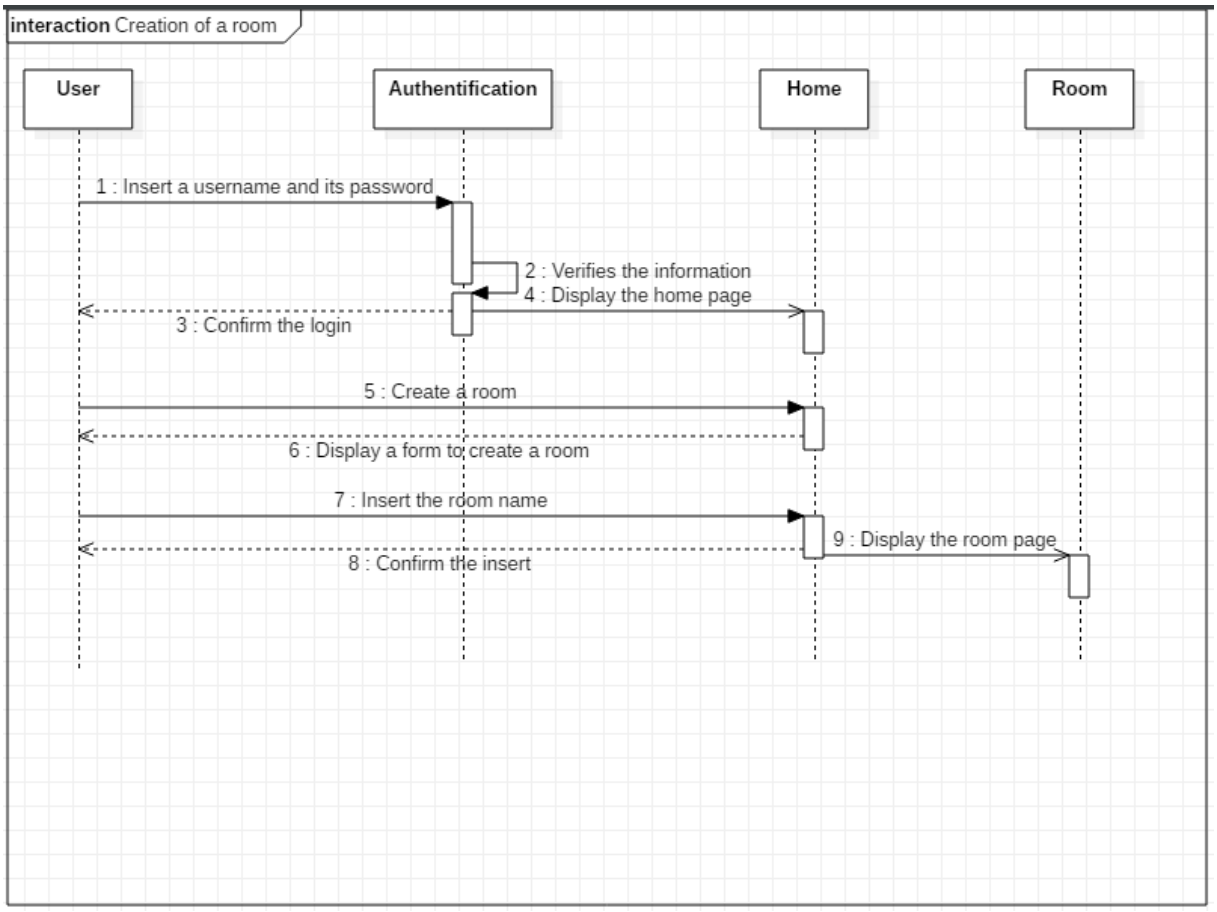

Figure 2 Sequence Diagram create a new room

After authenticating, the user can visualize all the rooms he has created, choose one and add users to his room.

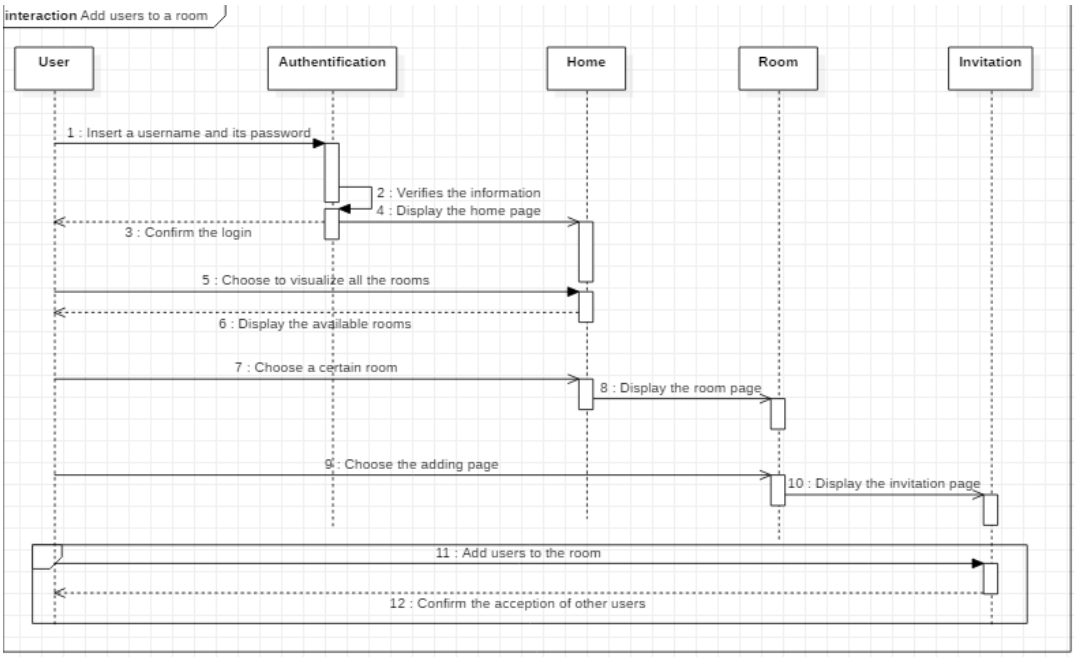

Figure 3 Sequence Diagram Add users to a room

3-Tools:

This chapter is dedicated to the practical part of the realization of our web application. It comprises a description of the development tools used, the choice was mainly based on the free and open source following tools:

Front-End

- AngularJs

  AngularJS is a JavaScript-based open-source front-end web application framework mainly maintained by Google and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications.

- Webix

  Webix is a flexible and extremely fast JavaScript UI framework with HTML5 widgets for developing cross-platform HTML5 and CSS3 compatible web and mobile apps. It's an ideal tool for building rich and powerful apps that can operate on all popular platforms.

- EasyRTC OpenSource is:

  - A browser client library written in JavaScript. This client handles the signaling and to a large extent insulates applications from ongoing changes in the WebRTC api.
  - A signaling server based on Node.js. Node.js runs on platforms as small as a single core Raspberry Pi (first edition) to servers in the cloud.

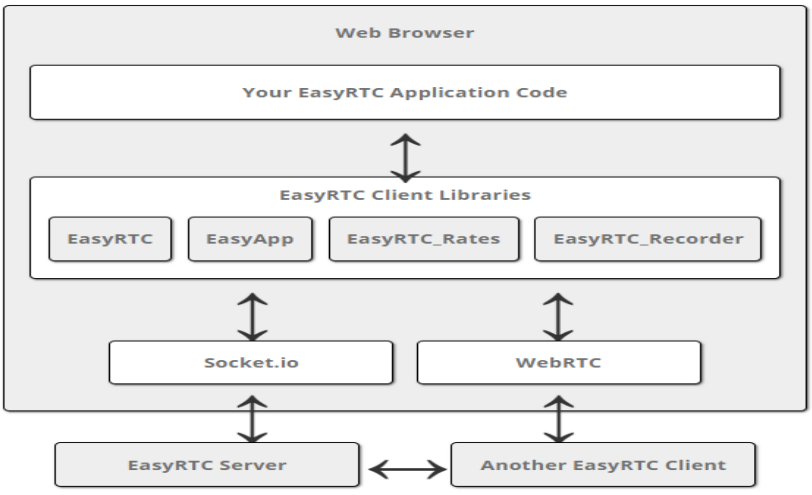

Figure 4 EasyRTC Architecture

EasyRTC is a full-stack open source WebRTC toolkit suitable for building highly secure, WebRTC applications. It is a bundle of web applications, code snippets, client libraries and server components meticulously written and documented to work right out of the box.WebRTC is a technology that allows two peers to communicate peer-to-peer, that is to say, to exchange data that doesn't go through a server.

Back-End

Django Rest Framework (DRF) is a library which works with standard Django models to build a flexible and powerful API for your project.

Basic Architecture
A DRF API is composed of 3 layers: the serializer, the view set, and the router.

Serializer: converts the information stored in the database and defined by the Django models into a format which is more easily transmitted via an API
View set: defines the functions (read, create, update, delete) which will be available via the API
Router: defines the URLs which will provide access to each view set

Figure 5 Django REST framework

It is mandatory now to build project on 3 scales, a backend rest end point application front end application and an EasyRtc application.

Architecture

The above diagram shows the architecture of the application. As it follows the model-view-controller the front end is developed using bootstrap and AngularJS and in the backend,, we have Django which is a python framework. In the front end we access data with REST API. To build the web API we need Django-rest-framework.

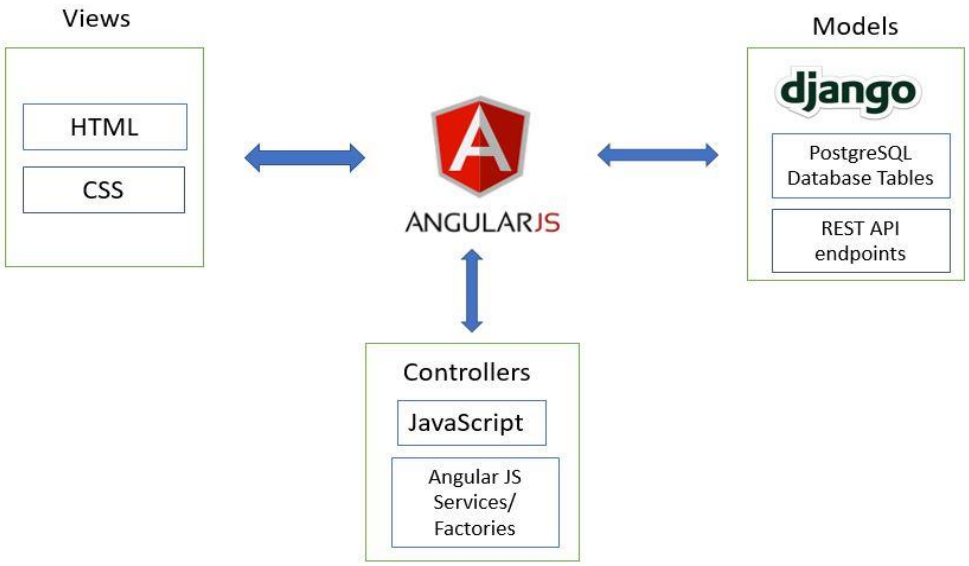

Figure 6 Architecture of the application

4-Realization:

After defining the user model, Django has a built-in authentication system that coordinate with the user model. With the built-in system we can generate the signup form.

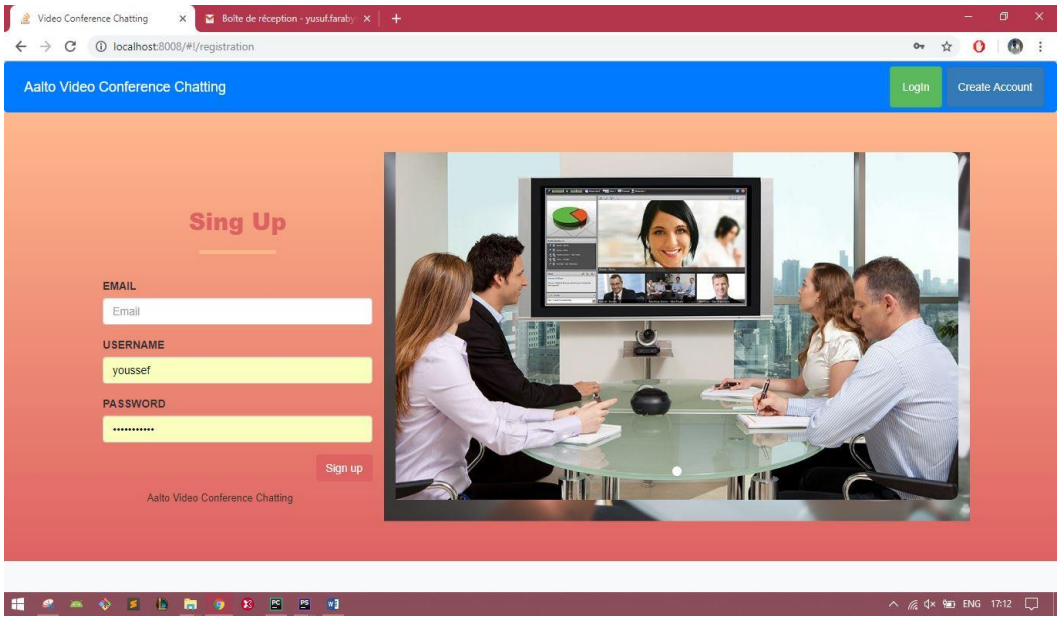

Figure 7 Sign Up Page

After the registration the user is redirected to the login page which is again handled by the Django authentication system

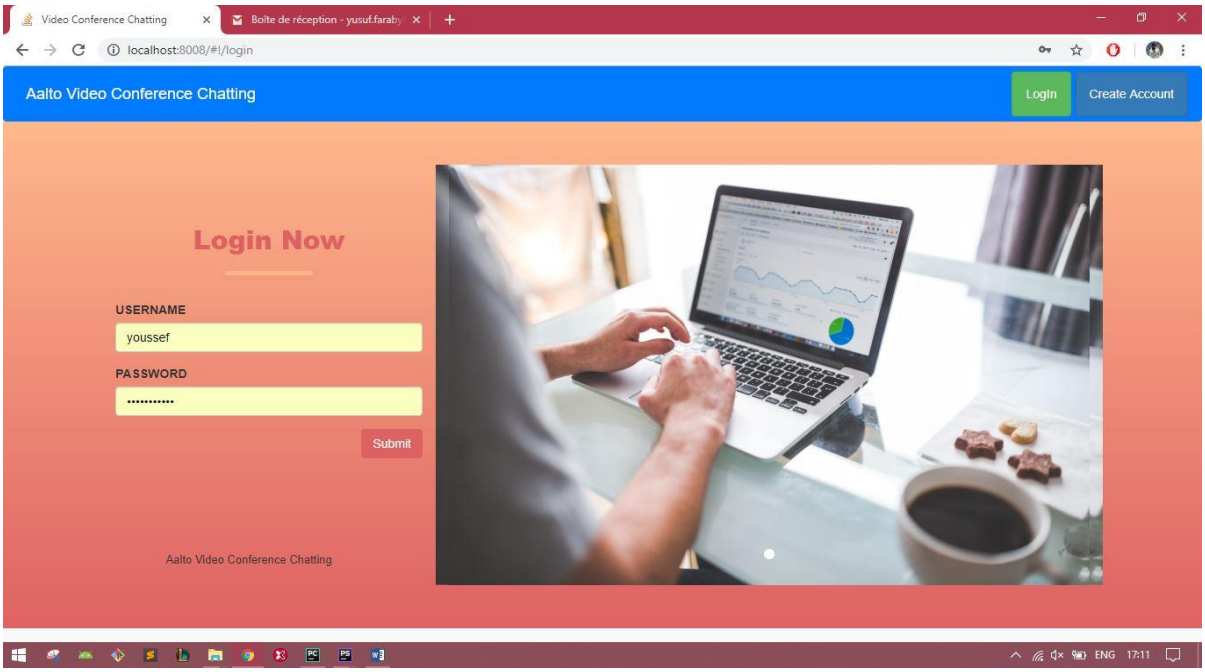

Figure 8 login page

When the user is successfully logged in the app redirects the user to the home. The user has 3 choices, display his rooms, create a new room, join the salon to wait for an invitation from another user.

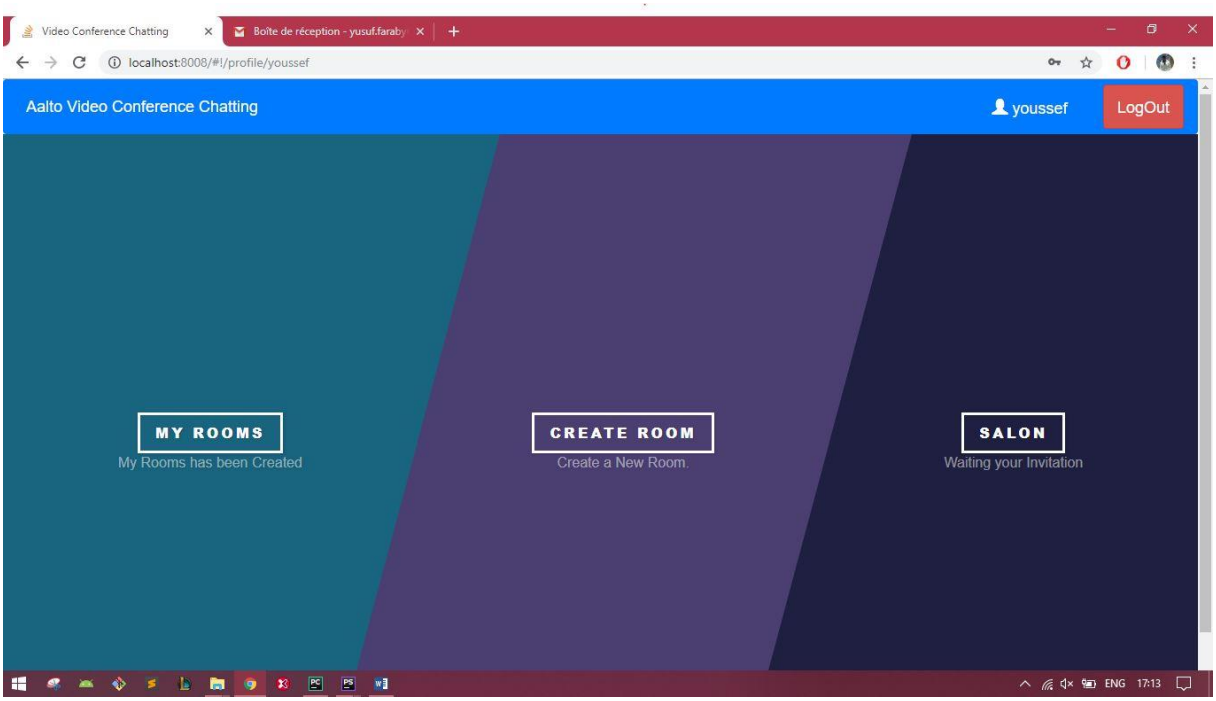

Figure 9 Home page

This is the form to create a new room and consult the list of his rooms.

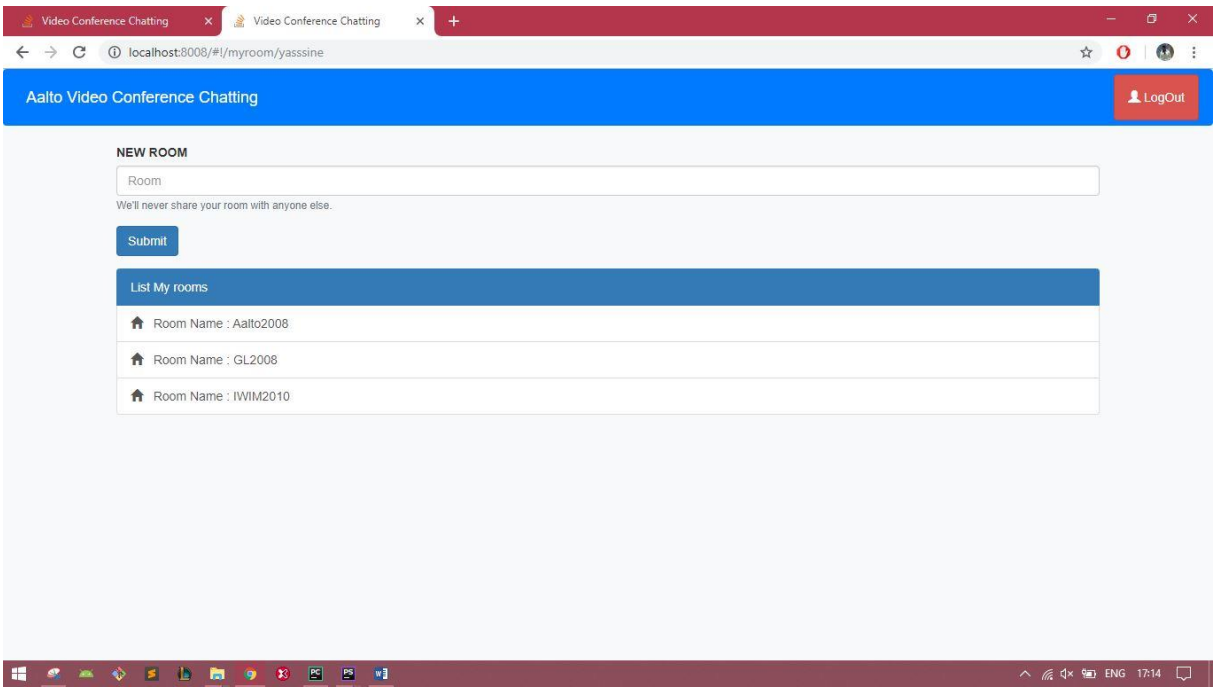

Figure 10 Create a new Room

When a user chooses to be member, he waits for an invitation from an admin.

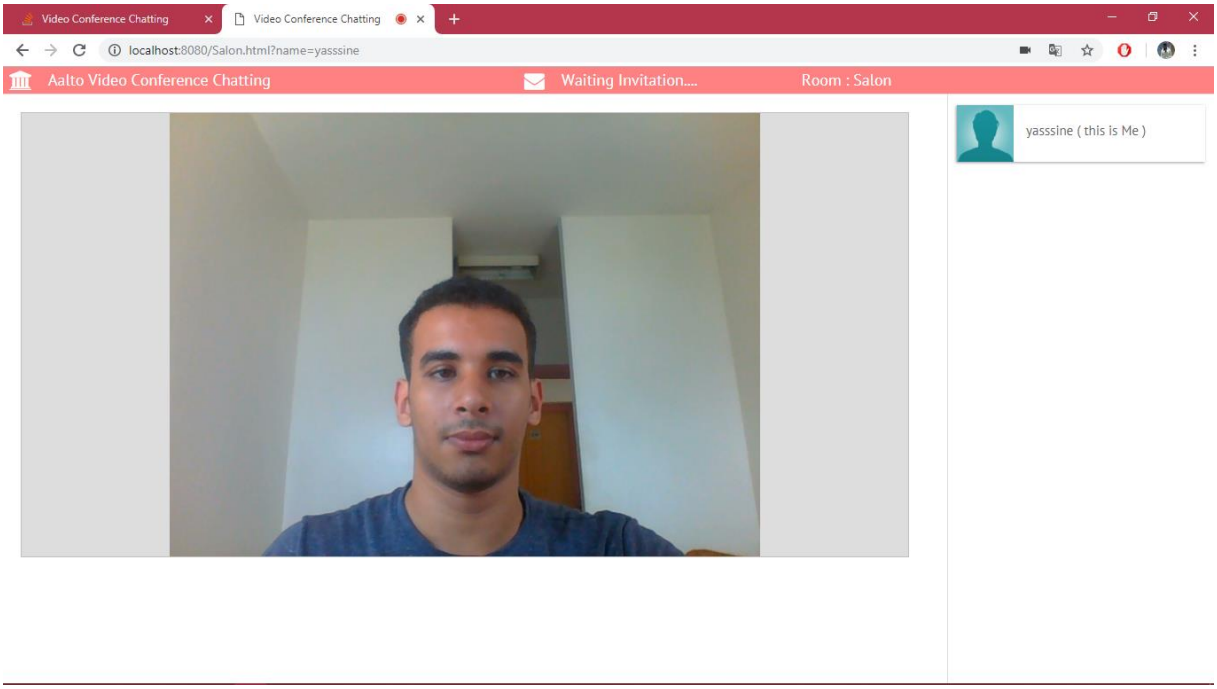

Figure 11 Salon Room

In this page the admin can make the following operations :
-invite other users the are waiting in the salon to join the conference
-allow other users to become admins
-Mute/Unmute the conversation

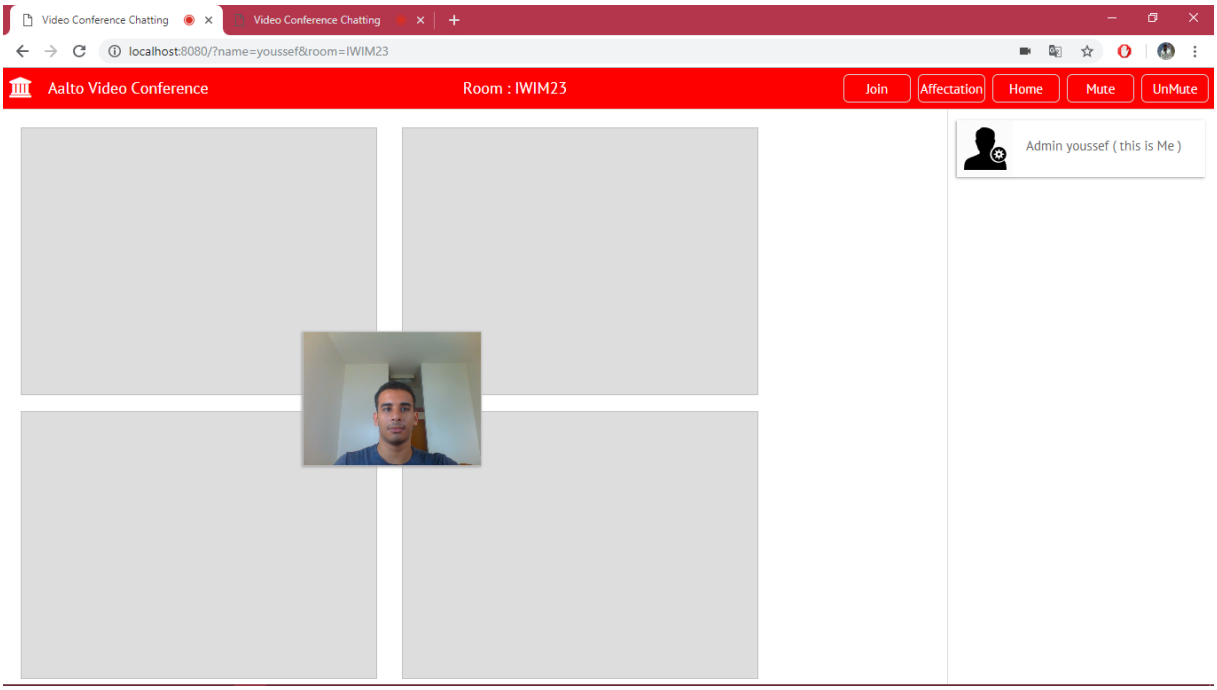

Figure 12 Admin page

This page displays the users whom the admin can operate on, that being said, he can add them to the room and mute/unmute them

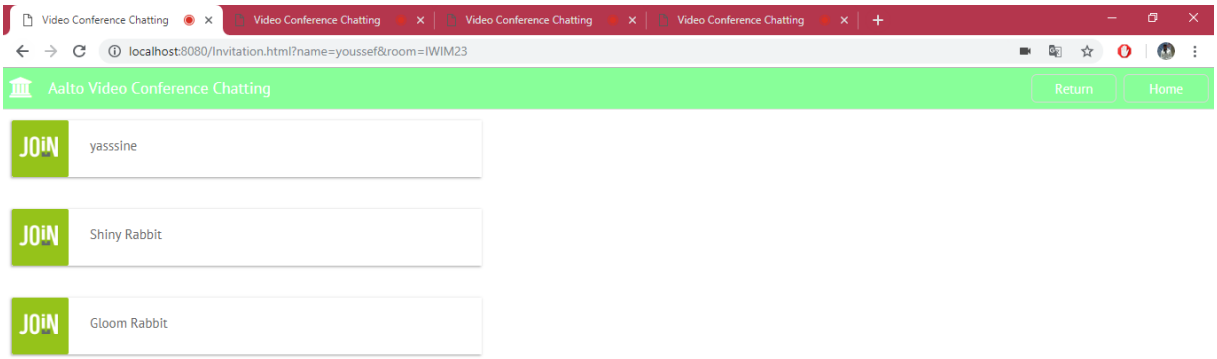

Figure 13 list of members

A member of the salon receives an invitation from the admin to join the room

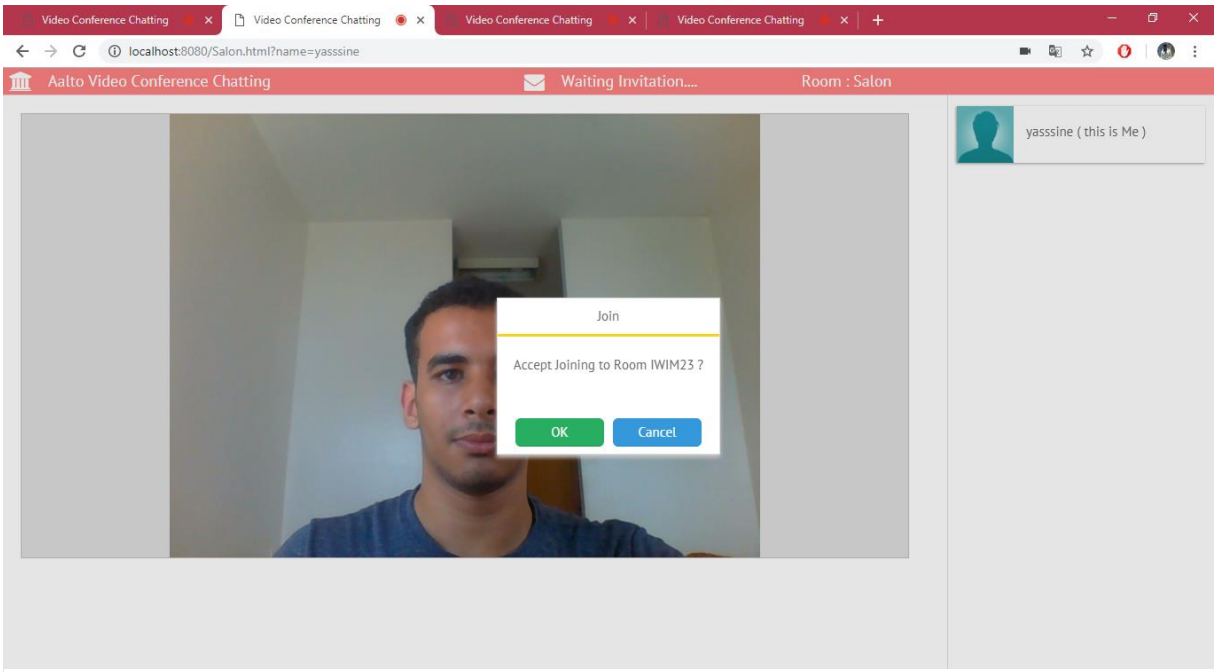

Figure 14 Accepting an invitation

This figure below illustrates the list of the users that the admin can allow them to become admins too.

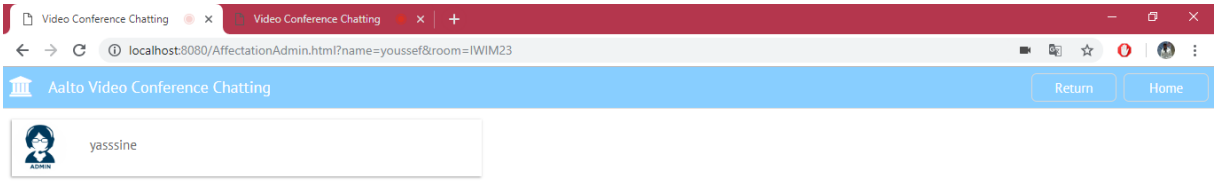

Figure 15 list of members

The members who were waiting in the salon are now in the room created by the admin

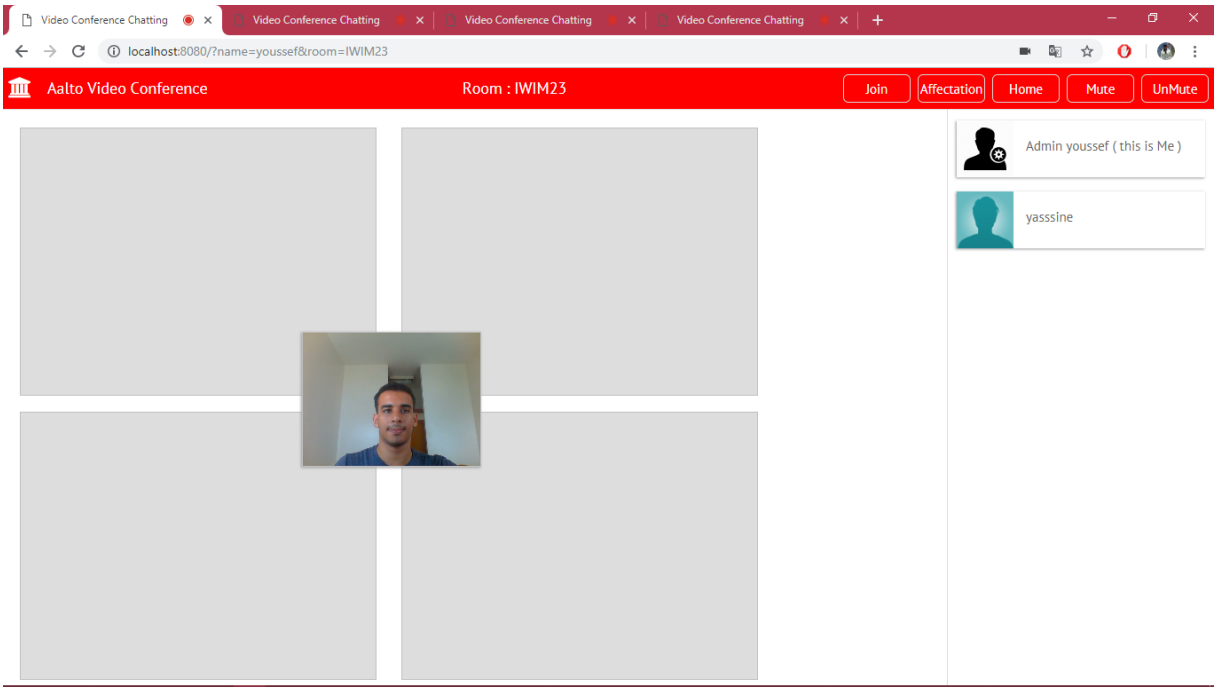

Figure 16 Accepted an invitation

Finally, the admin and the members are gathered in one single room to start the conference, organized by the admin.

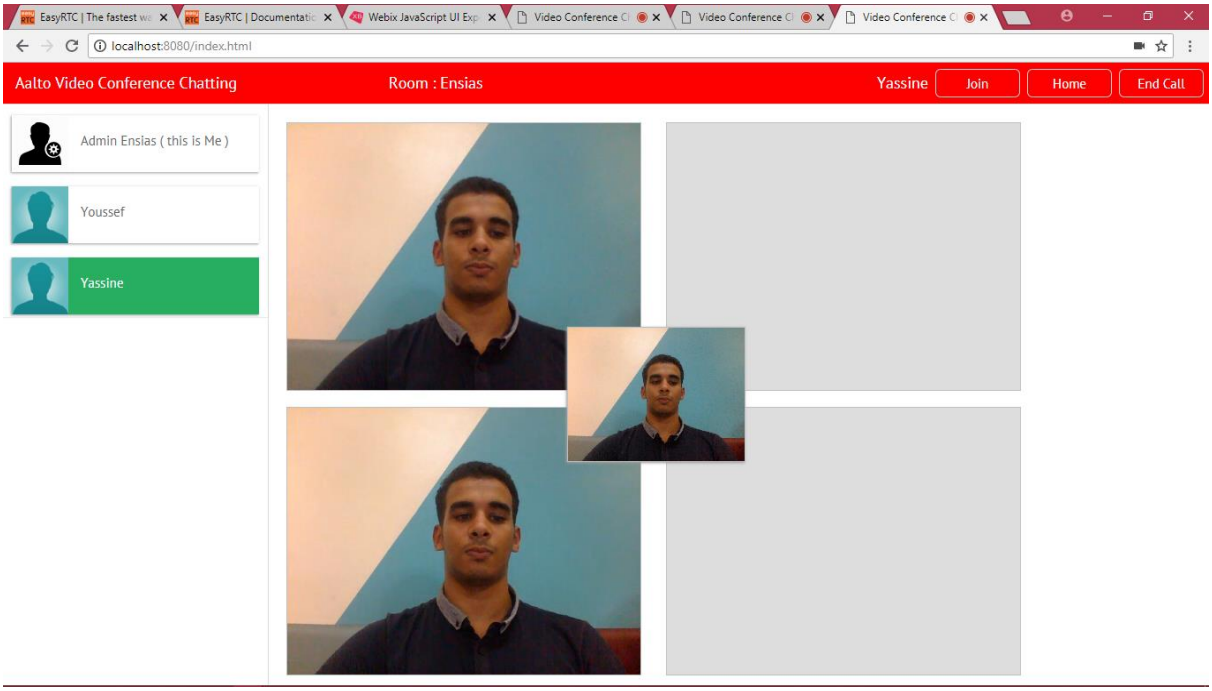

Figure 17 start the conference

5-Conclusion:

In essence, this project's goal was to develop a web application of Video Conference Chatting. We have presented throughout this report the approach we have taken to develop this application. and to improve it.

This project was a great opportunity to work with video streaming, to develop my knowledge and skills and especially to learn how to create a web application with Django Rest API.