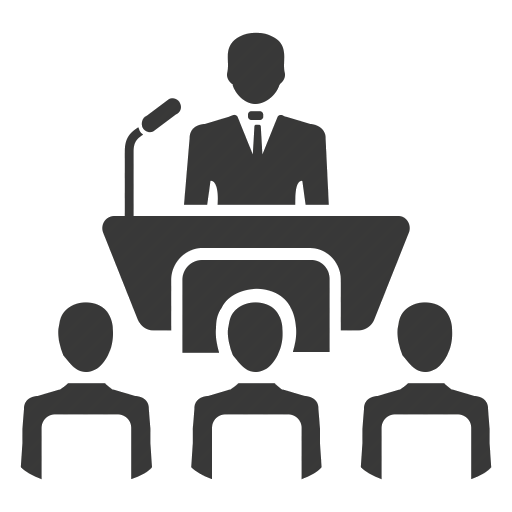
conference management system



# TEAM COVID-19

GROUP 925/2

**TEAM LEADER**

**Socaciu Mihai (931)**

**TEAMMATES**

**Olah Istvan**

**Vlad Opris**

**Ciprian Muresan**

**Daniel Oprea**

**Eugen Ajechiloae**

**Razvan Neta**

**Tudor Paun**

**Sebastian Chirodea (931)**

# **ABOUT/HELP**

# In this application you can do things based on your role. If you’re the chair, you can create conferences, if not, then you can have assigned roles to a conference by the chair. You can create submissions for a conference, which later can be reviewed by your colleagues, in the same manner as you can review theirs if you like their submissions. Later these can be presented at meetings called sections, where you or other people can pitch the submissions. Due to current situations, we recommend using any online platform for these meetings.

# **PROJECT REQUIREMENTS**

# **CMS** (Conference Management System) is the solution to the problem that the scientific community has: a lack of an easy to use tool to manage scientific conferences. From the beginning, our task was to build an application that helps ease the tedious task of organizing a scientific conference. What the application does is: it streamlines the process of creating the conference, registering, uploading papers and abstracts and of reviewing submissions.

# The process takes place in several steps. In the beginning, the chair that wants to organize the conference must first publish it on the website with all the information. This includes: name, deadlines, call for papers, reviewing committee, program etc. After the details about the conference and about the program committee members are settled, the users can start submitting abstracts by submitting the abstract and some other meta-information about it, but this must be done before all the deadlines. Some conferences allow uploading full papers.

# When the deadline for the abstracts passes, the next stage is the reviewing stage. Here each PC member bids on submissions in order to be assigned for review in latter phases. This is the final stage for the submissions.

# The paper’s presentation is the last step of the conference. Here are different roles such as speakers (participants with accepted papers), session chairs (supervising sessions) and listeners (attendees to the session).

# **STAGES**

1. **Analysis** **phase**

We started by analyzing the project requirements and having a clear image about the objects in our application.

1. **Design** **phase**

We moved further creating the use case diagrams in order to recognize the authorized roles of the application. Then we had to discuss about the implementation of the database. We created the database diagram and then the class diagram.

1. **Implementation**

After the creation of the database, we started implementing the application. First, we implemented the model classes and some basic frontend components. Then we connected the database and started adding the functionalities along with their associated UI forms.

1. **Testing**

The app has been tested manually by every member of the team. Each of us has tested the functionalities he/she has implemented on the feature branch, and only after that, the functionality has been merged to the development branch.

Inside the code, data has been validated using try/catch blocks. For every invalid input from the user, the app will show a Snackbar, explaining why that operation cannot be performed.

1. **Delivery**

We deployed the server on a Raspberry Pi running Rasbian. It connects to a MySQL database. We had constant check-ups with the requirements document to check on our project. By doing this, we assured ourselves that we are on the right path and time was not being wasted by implementing unnecessary features and correcting faulty ones.

# **TECHNOLOGIES USED**

**Web Framework:** Angular, Spring Boot

**Database server:** MySQL with JPA as database manager for the backend

**Programming languages:** Javascript(with Typescript), MySQL, Java

**User Interface**: Angular Material

**Version Control:** Git

**Software for Diagram Creation:** StarUML

**Task management:** Trello

**TypeScript:** Angular is a web application framework led by the Angular Team at Google. It has a massive community that contributes tons in the way of functionality and assistance. Angular is written using the TypeScript language, which is basically a superset for JavaScript. It fully compiles to JavaScript, but also helps in spotting and eliminating common mistakes when actually typing the code

**Git:** This open source version control system helps developers to easily track code changes among non-linear workflows. In Git, a core assumption is that a change will be merged more often than it is written, as it is passed around to various reviewers

**Angular Material:** The Angular team has been updating their framework with material design components. Material Design is a visual language that synthesizes the classic principles of good design with the innovation of technology and science. The great thing about this is that it’s a consistent and deeply justified system that considers how people interact with digital products and strives to make their experience as smooth as possible.

# **FUNCTIONALITIES**

**Views**:

- see all conferences with details

- see submitted papers

- see given reviews and others of an abstract

- see received and given reviews

- see sections for presentations

**Actions**:

- register and login

- create conference (invite co-chair, pc or sc members)

- submit abstract and fullpaper

- change deadlines of the conference

- assign submission to PC Member for review

- review = decide verdict for submission

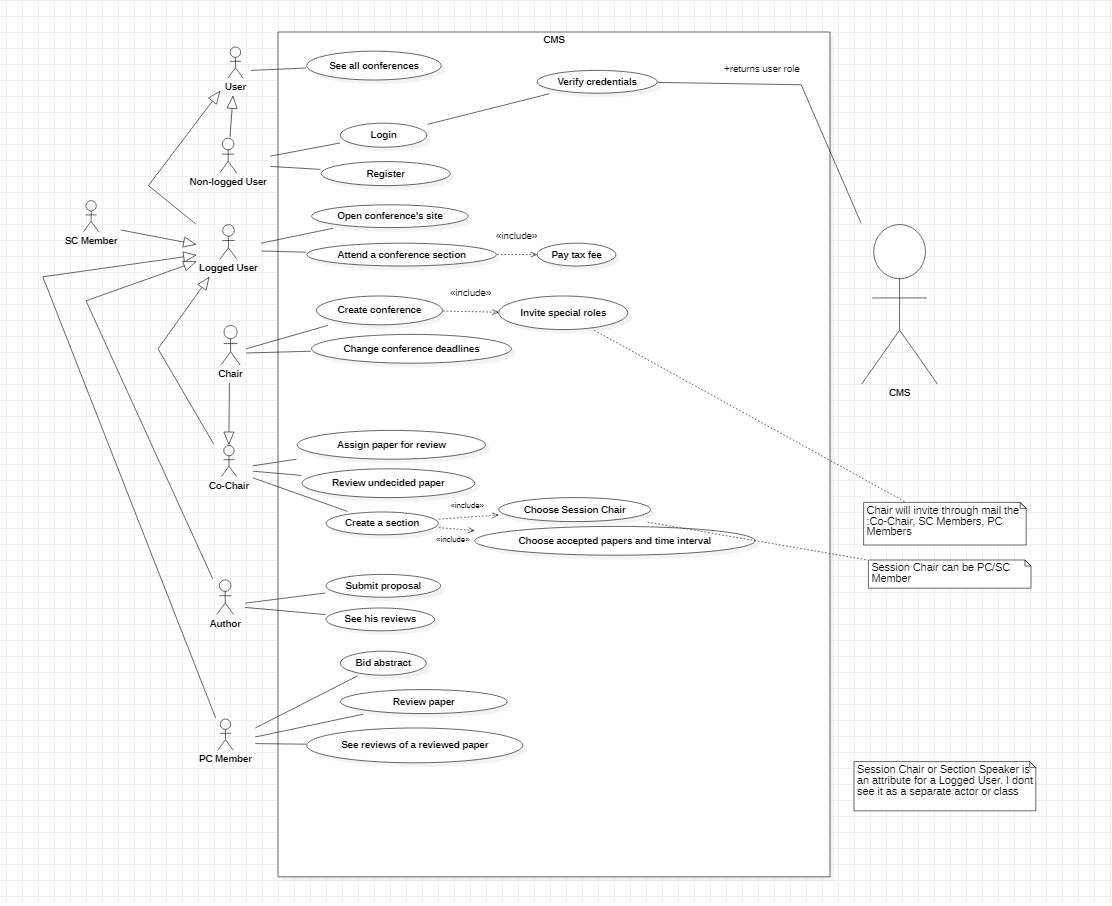
- bid abstract if wants to review

- participate in a conference section (pay tax) [mandatory for some]

- create section for the conference

# **DIAGRAMS**

# USE CASE DIAGRAM



# DATABASE DIAGRAM

A close up of a map

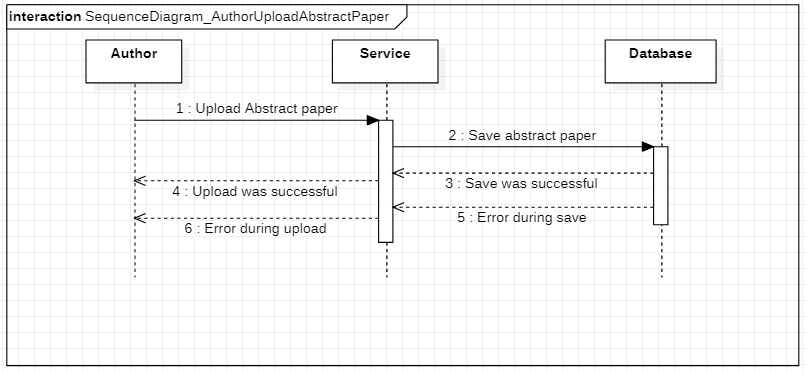
Description automatically generated

# A close up of text on a white background Description automatically generatedCLASS DIAGRAM

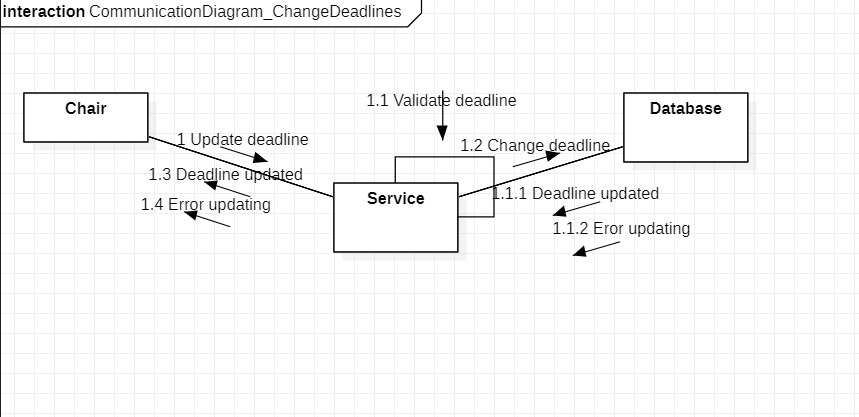
ARHITECTURE DIAGRAM

A close up of a map

Description automatically generated

SEQUENCE DIAGRAM – UPLOAD SUBMISSION

COMMUNICATION DIAGRAM – CHANGE DEADLINES



## STATE MACHINE - LOGIN

A close up of a piece of paper

Description automatically generated

# STATE MACHINE - REVIEW

A close up of text on a white background

Description automatically generated