CONFERENCE MANAGEMENT SYSTEM

A. INTRODUCTION

- Objective.

The Conference Management System manages the information concerning scientific conferences. From the point of a Program Committee deciding to organize a conference edition to the actual presentation of the authors' proposals, every step must be supported by the application.

- Users

Every person must create an account, by choosing a username, password and their personal information: full name and email address. Each user can create a conference, in which case he or she will be the chair of the conference. The user can also be invited by other chairs to participate in their conference, each conference having participants with different roles.

- Chair and Co-Chairs

  They have the permission to create a conference edition and to make the important decisions. They can invite other users, by requesting them to participate.

- Authors

  Authors can see all the conference editions on the platform and they can send their papers.

- Speakers

  If the authors' paper is accepted, then the author is going to give a live presentation, and become a speaker

- PC member

  Users who are invited to a conference are all initially Program Committee members. They will help the chair in managing the conference. After authors send their papers, PC members must "bid" their abstract and state if they want to review their full paper. PC members can also decide to send a paper, in which case they lose their other permissions.

- Reviewer

  The chair analyses the bids and assign each paper to 2-4 PC members, which become reviewers. Reviewers must read their assigned papers and rank each one from 1 to 5. The paper will be accepted/rejected based on the reviews.

- Listener

  Users who are present at a presentation but have no important role are named speakers.

- Session chair

  Each presentation is supervised by a session chair

B. STAGES OF IMPLEMENTATION

➔ We started by understanding what and how we were going to realize the project.
➔ By studying "OBJECT-ORIENTED SOFTWARE ENGINEERING", we have designed a series of diagrams that helped us see different sides of the project, and by doing so, continuingly remodeling our plan.
➔ We then started the implementation. We created a simple, but efficient structure. The project is divided into 3 main parts: User, Conference and Permissions. Each user has certain permissions on a conference, these being checked each time he or she wants to make an action.
➔ For each component of the application (conference, review, section aso), we have structured a model, a repository, a service and a controller that enables us to tackle the complexity of the program.
➔ We have used JPA Repositories, as they provide a lot of code. Because of that, we did not have to make all the CRUD operations and we could work directly with our database easier.
➔ We have used DTOs (Data Transfer Object) for each entity, because the cost of each call is related to the round-trip

time between the client and the server, and the DTO aggregates the data that would have been transferred by the several calls and serves it in just one call.

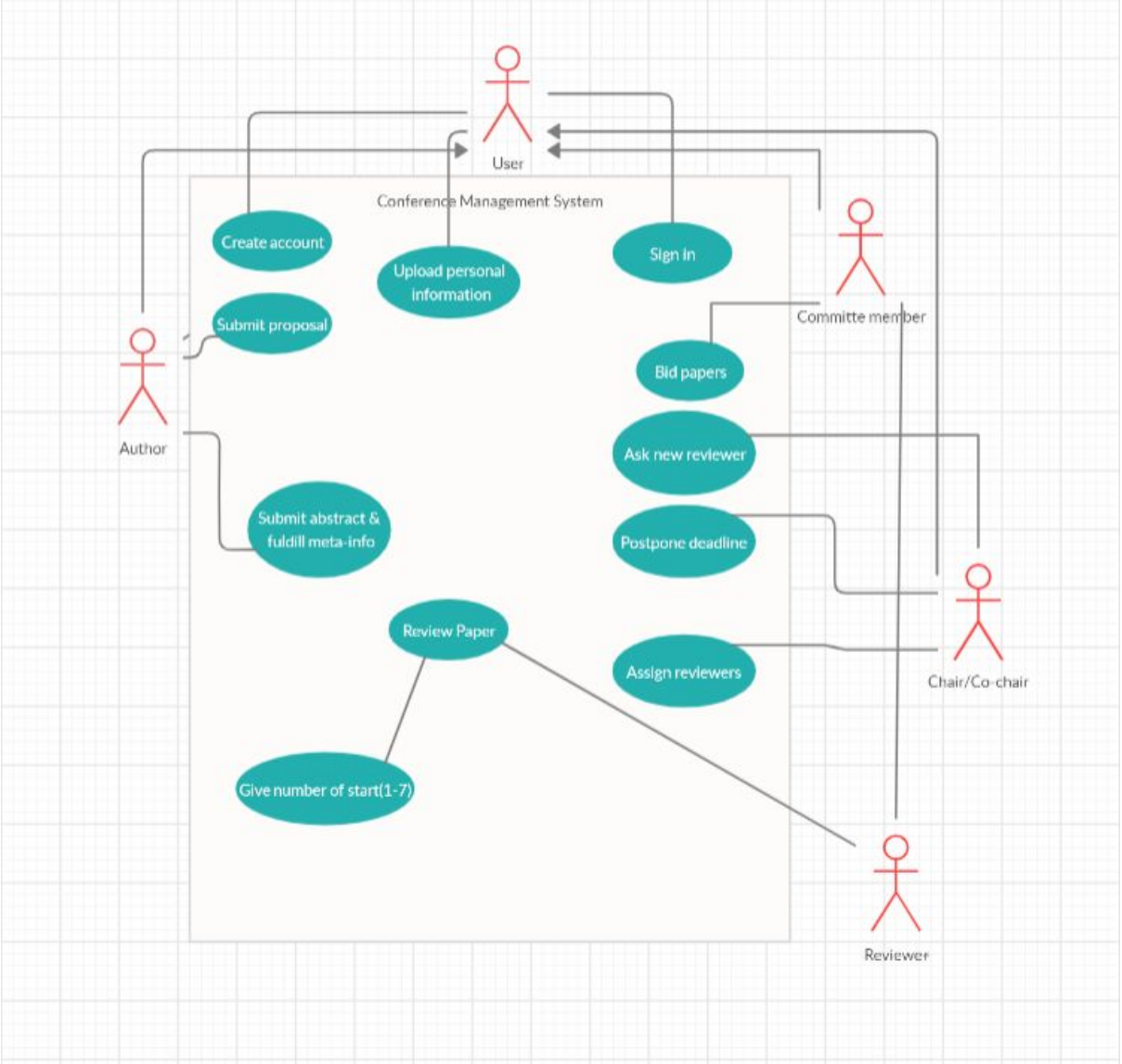
➔ For the communication between client and server, we used Spring MVC that dispatches requests to handlers. The default handler is based on the @Controller and @RequestMapping annotations, which is typically responsible for preparing a model Map with data and selecting a view name, but it can also write directly to the response stream and complete the request.
➔ As we were moving forward with the backend of the application, we started developing the design. We have used javaFx and tried to make the application as user-friendly as possible.
➔ We used MD5 hash generators to ensure the security of the users, through encoding the passwords.

C. TECHNOLOGIES

1. Postgres Database
2. Java
3. JavaFx
4. SpringBoot
5. Spring (JPA, Hibernate, REST, MVC framework)
6. MD5 hash generator for encoding passwords
7. AdobeXD to create the prototype

D. DESCRIPTIVE DIAGRAMS

1. Use case Diagram

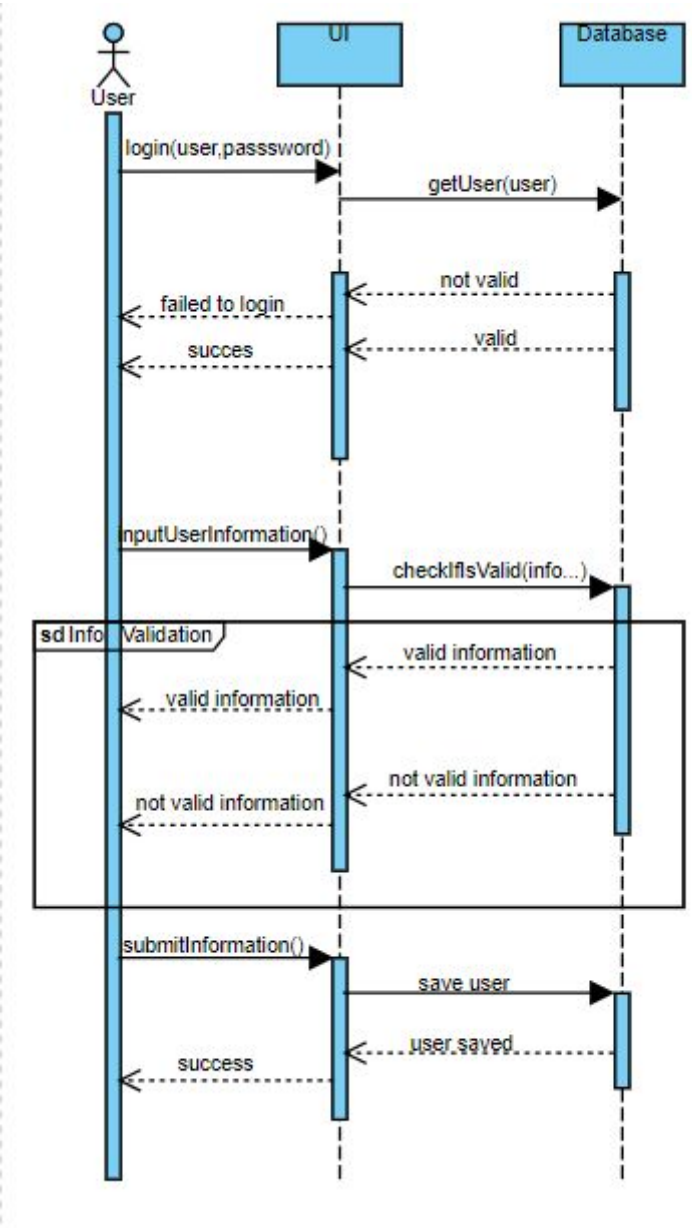


The Use Case Diagram focuses on the visible results of the behaviour of the system from an external point of view. Actors initiate a use case, which can then initiate other use cases. The use case

diagram is intuitive, such that it can be used as a medium for communication with the user.

2. Sequence Diagram

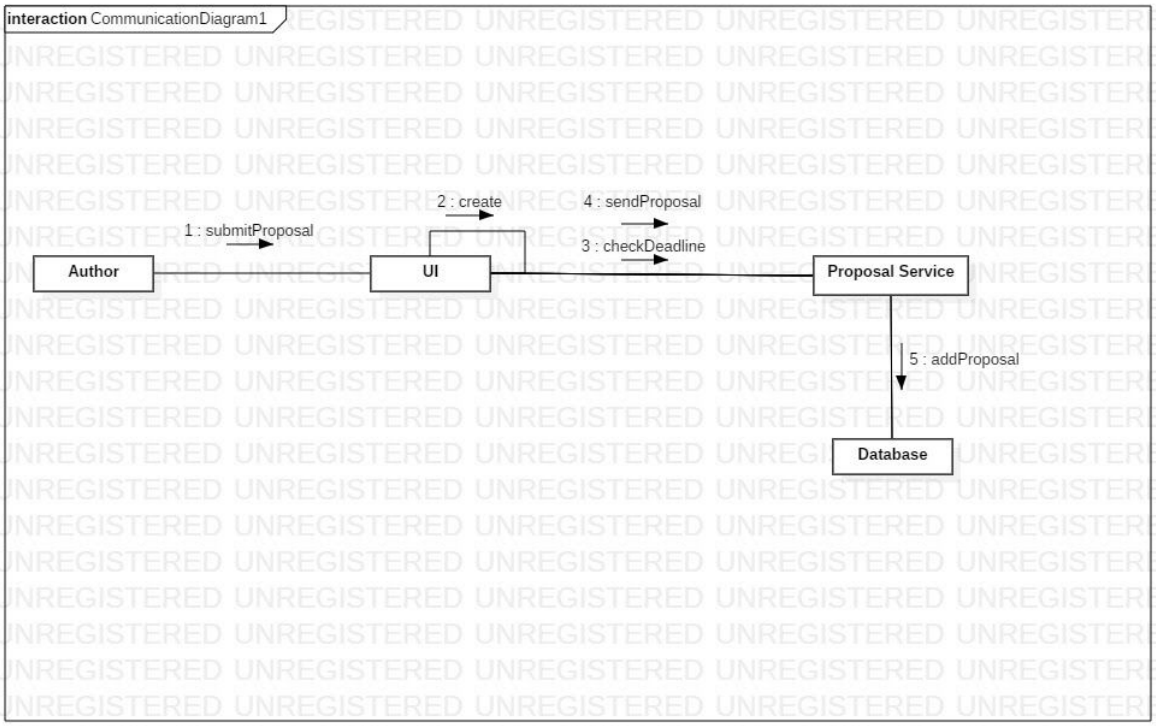
Sequence Diagrams ties use cases with objects. They require more technical background, but are more precise. By working on these diagrams, we expanded our perspective on the project and we were able to anticipate and fix some problems. The sequence diagram below depicts use cases for the general user. It not only shows the use cases initiated by the user, but also the interaction between the user interface and our database.



Note: You can see all the other Sequence diagrams here.

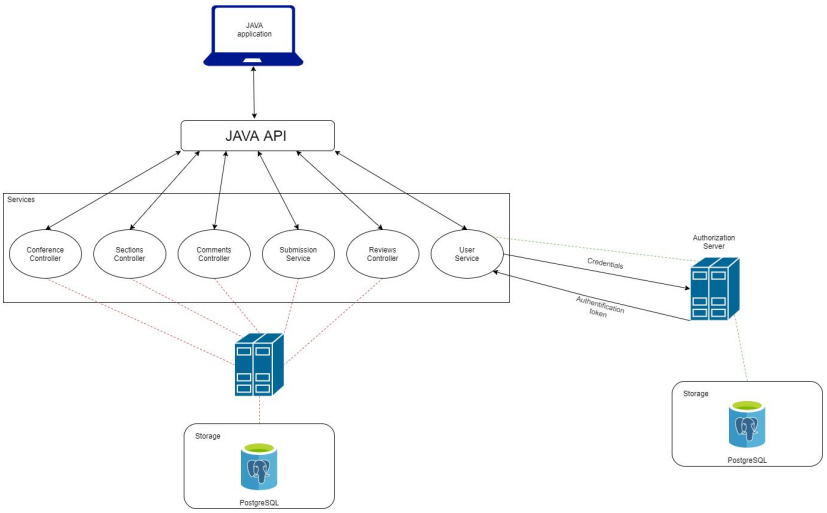
3. Collaboration Diagram

In the collaboration diagrams, we depict the same information as in the sequence diagram. However, by the numbering of the interactions we can be more precise about each action. We splitted the diagrams into independent, smaller, easier to read ones.



Note: You can see all the other Sequence diagrams here.

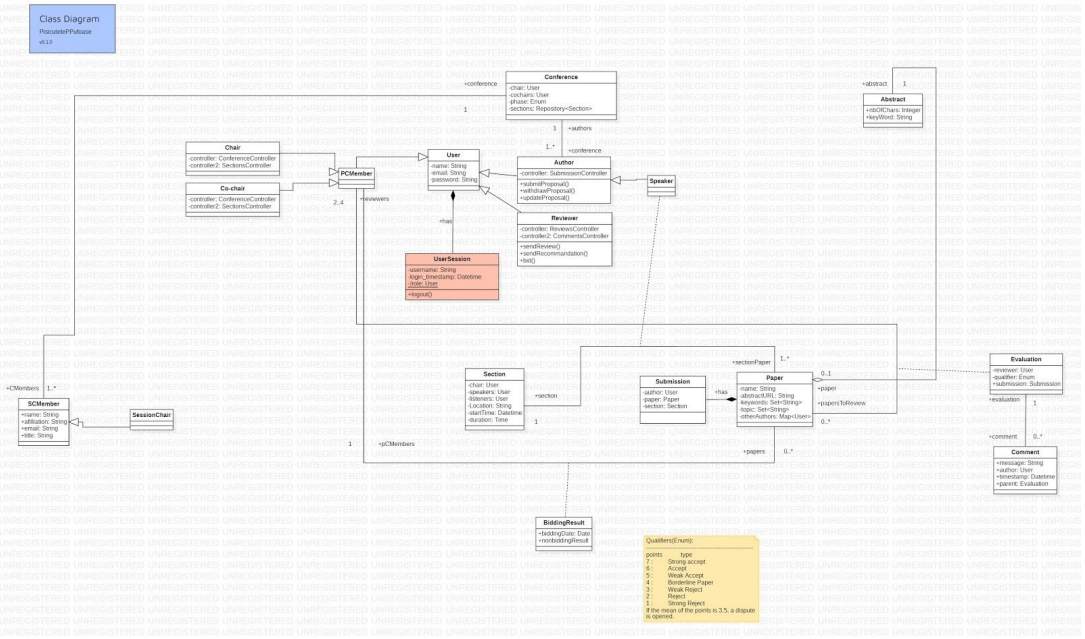
4. Architecture Diagram



5. Class Diagram

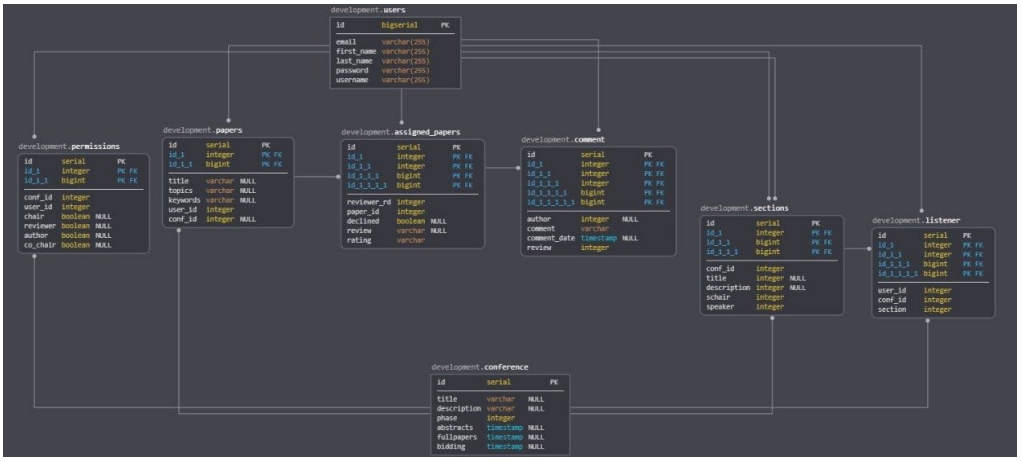
The following diagram represents the static structure of a system in terms of objects. Classes are described by their attributes and methods, but also by the relationships between them. By working

on this diagram, we created the main structure for the project. We carefully analysed the hierarchy between all classes and remodelled it until it was both sufficiently effective and straightforward.



6. Database Diagram

The database diagram shows the logical structure of the data. Information about all the entities in our projects are stored, therefore a well-designed database is important.



E. TESTING

We made use of one of the advantages of working with SpringBoot, which are the automatic tests for our code.

F. USER MANUAL

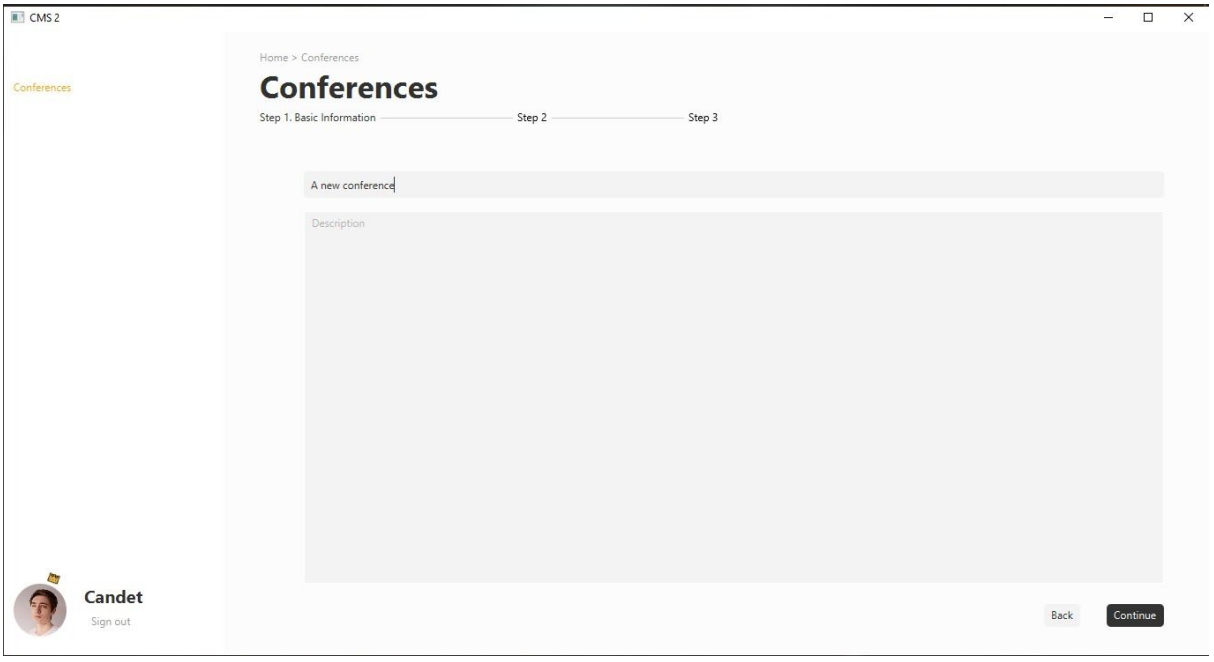
1. Registration

The user must must fill in his or her full name and email, then choose a username and a password.

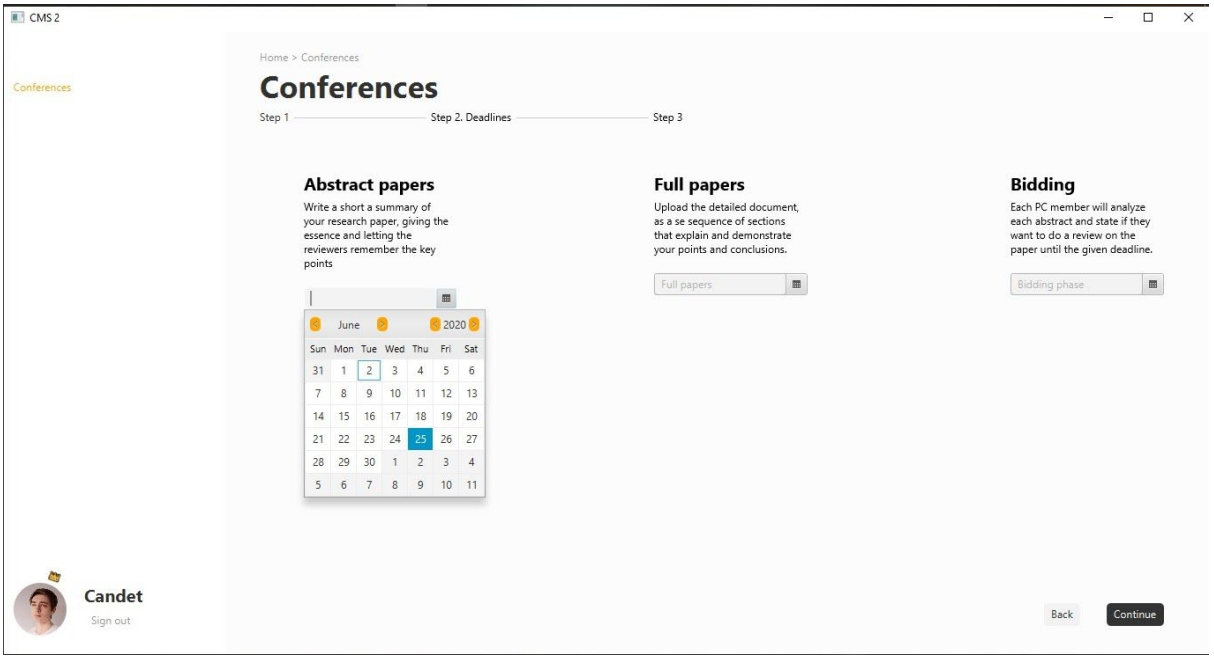
2. Creating a conference

Any user can create a conference by clicking on "New conference". He or she will have the rank of "chair".

i) First step: choosing a name and a description.

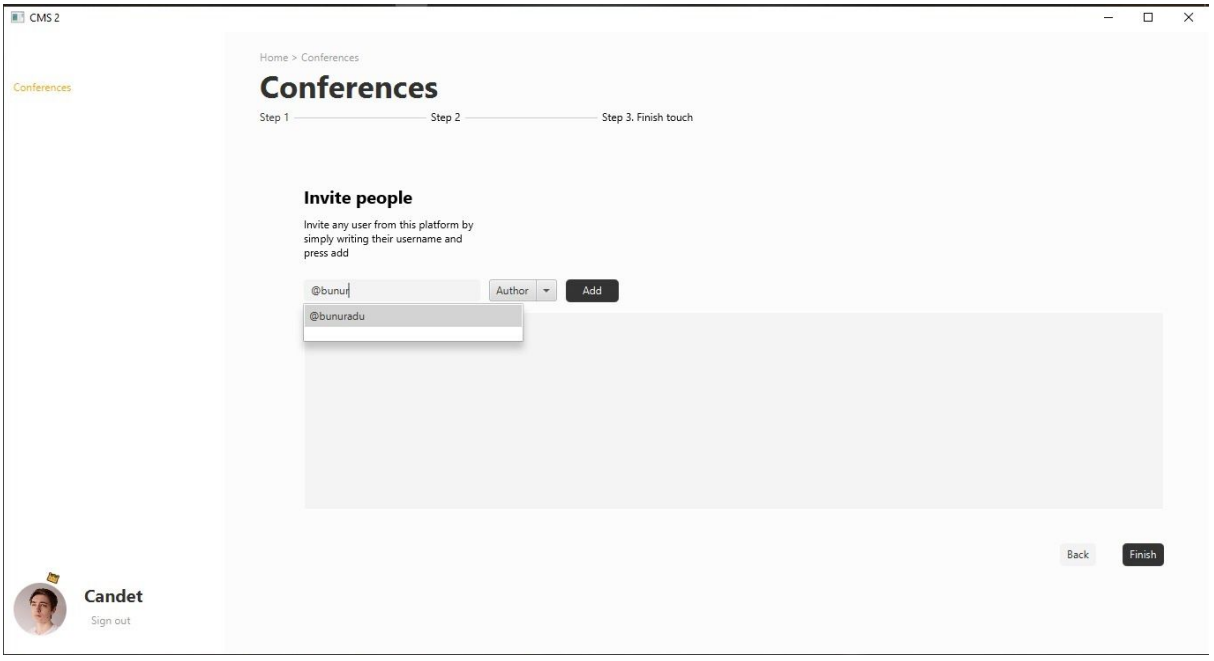


ii) Second step:setting deadlines: one for the abstract uploading, one for the full paper uploading and one for the bidding process. The first two concern the authors that want to submit their proposals and the later concerns the PC members.

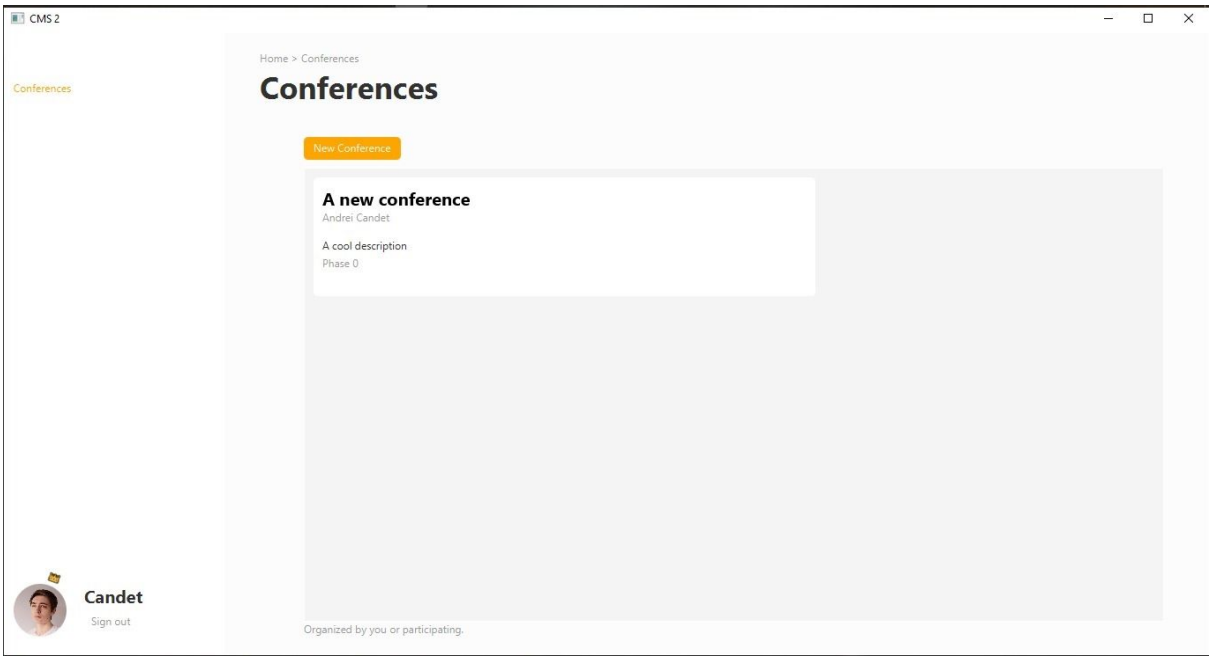


iii) Step three: Inviting people to be part of the conference's activity.

The chair can invite people by typing their username. Suggestions will appear while typing, so the username is easy to find. Before sending the invite, the rank of the new participant must be specified.



By clicking "finish", the new conference will be created.

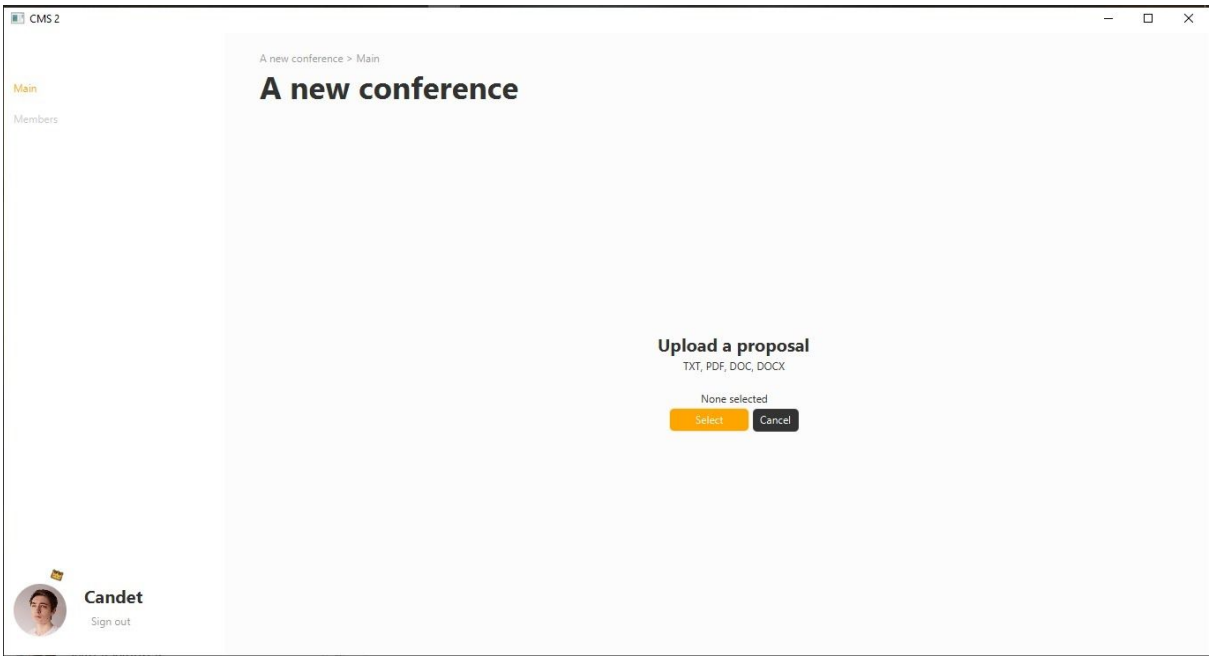


3. Managing your conferences

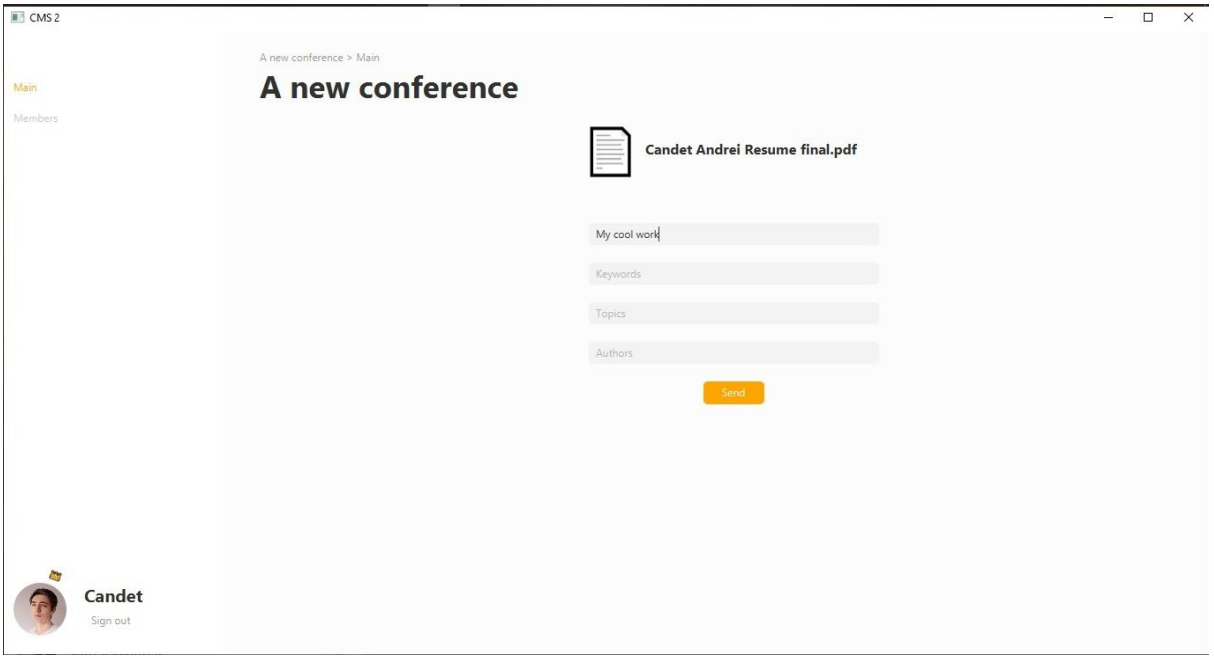
The user can browse through his conferences and can manage all of them independently. By selecting a conference, the user can see his details and the tasks he or she has. When clicking on any conference, the user must specify the rank he or she wants to enter. The user can't, however, choose a rank he did not receive a permission for.

4. Sending a proposal

By entering with the role "author", users can send their proposals. First, the author has to select a file (TXT, PDF, DOC, DOCX) containing his or her work.



By filling in the required meta-information (title, keywords, topics and other authors) and clicking send, the user's proposal is received by the PC committee.



G. TEAM

MEMBER	ROLE
BRANEA ALEXANDRU	
BONDOR DAN	
BUNU ANDREI	
CANDET ANDREI	
CEACA RADU	
CALARASU VICTOR	