TEAM WILDCATS

Morea Răzvan Mihai

Munteanu Ana-Diana

Naiman Alexandru-Nicolae

Nariţa Gheorghe-Marcel

Nistor Alexandra-Maria

TEAM LEADER

Naiman Alexandru-Nicolae

COORDINATOR

Borodi Flavia

TABLE OF CONTENTS

REQUIREMENTS

The objective of a Conference Management System is to support the automatic management of information related to scientific conferences. This information concern: the authors submitting proposals, the members of the Program Committee, the submissions' abstract and full papers proposed, meta-information about these, the deadlines for different phases of sending proposals, assigning paper to reviewers, evaluation deadline and announcing the results of paper valuation. Once the accepted proposals are known, the different sections of the conference are decided; each accepted paper is planned to be presented in a section. Each conference participant have to pay the registration. The conference participants which can be: session chairs, speakers or listeners.

Once a steering committee decides to organize a conference edition, the information about the event is posted on the conference site. This information includes the name of the conference, the interval in which the event will took place, the "Call for papers", the deadlines, the Program Committee, the conference sections and program a. s. o.

Mainly there are 3 distinct phases regarding the management of the information preceded by a preliminary phase, the PC members (including the conference chair or the conference co-chairs) uploads information about their name, affiliation, email address, their personal web-page, their username for the CMS and the password for accessing the information about the conference. The chair or one of co-chairs have the permission to change deadlines by postponing them to later data.

In the first phase, each person interested to submit a proposal create its own user account fulfilling the information required (the name, affiliation, email address). Once the account is validated, the submitter logins, submit the abstract and fulfills the meta-information required: the name of the proposal, the keywords, the topics, the possible list of authors and their meta-information). Depending on the conference it is possible to upload also the full paper (usually a file in .pdf or .msword format). Uploading proposals can be done before the deadline mentioned on the conference site and on the "Call for papers". It is possible to have two different deadlines: one for the abstract and another for the proposal. Until each deadline, users may upload new versions both for the abstract and the paper.

In the second phase, PC members are required to bid the proposals. Each PC member has to do a brief analyze of abstracts or papers in order say if they are pleased to review some papers, or if they are could evaluate some papers, or if they refuse to evaluate other papers. Once the bidding process closed (there is a deadline for

bidding), the conference chair or co-chairs assign to each reviewer the papers required to be evaluated. By default any reviewer do not receive for evaluation a paper refused in the bidding phase. At least two different reviewers have to evaluate each paper. Usually there are three reviewers (evaluators) for each paper but depending on the conference level there can be also 4 reviewers. The result of each reviewer is one of the followings qualifiers: strong accept, accept, weak accept, borderline paper, weak reject, reject and strong reject. The papers whose evaluation do not contains any level of reject are accepted by default. Also the papers do not containing any accept will be rejected. In case of papers having contradictory evaluations the chair/co-chairs request reviewers to discuss in order to get closer evaluations. If this will not be possible, is up to the chair to decide if another new evaluation from a different reviewer will be required or if the paper will be rejected or accepted. The evaluation qualifier is justified by means of a set of recommendations that each reviewer attach to evaluated proposals. Once a reviewer upload the results of its evaluation, she/he will see the other evaluations of the same paper delivered by other reviewers. After deciding about the acceptance or not for all submissions, the authors are announced by email about their results. Only papers accepted and presented on the conference will be published. After receiving the acceptance decision, authors are invited to improve their accepted papers taking into account reviewers recommendations. PC members excepting the chair/co-chairs can submit proposals by logging as authors. In this case, they have not the right to see neither which are the reviewers of their paper nor the comments between reviewers.

The third phase concerns the conference activities - mainly papers' presentation. Depending on the number of participants and of accepted papers, the conference is structured on different sections, some of them organized in parallel. Each section is supervised by a session chair which in the most cases is a member of the Program Committee (PC member) including the chair/co-chairs. Authors presenting their accepted papers are named speakers. Participants to each section are: chair, speakers and listeners. In order to assign the most appropriate rooms for each section, in case of important conferences (having hundreds of participants) these are invited to specify which are the sections they intend to participate in. A speaker cannot be the chair of the section in which she/he present his paper. Usually, before the presentation, speakers upload on the conference site the content of the presentation by means of a .pdf or .ppt(x) file.

TECHNOLOGIES USED

1. **Programming languages**:
    - front-end in JavaScript with React Mobx, HTML5, CSS3
    - back-end in Java Spring Boot
2. **ORM**: Hibernate
3. **Database**: PostgreSQL
4. **Versioning System for Database**: Liquibase
5. **Client-Server communication**: Axios
6. **Tools used for diagram creation**: Visual Paradigm, Star UML
7. **Version Control**: GitHub
8. **Task Management**: Trello
9. **Manual testing with bugs reporting on Trello**

INSTALATION MANUAL

Create new folder for project --> open GitBash there:

git clone https://github.com/alexnaiman/Conference-Management-System.git

In order to use the application you need internet connection.

You have to install PostgreSQL, which will require setting a password. Be careful when choosing the password, since you will need it later.

Once PostgreSQL is installed, open the pgAdmin app by looking it up on the default search bar of windows. It will ask you for the username and password chosen beforehand.

Open Servers->PostgreSQL->Databases. Right click on Databases and on CREATE, choose a name for your database. You do not have to create tables inside the database!

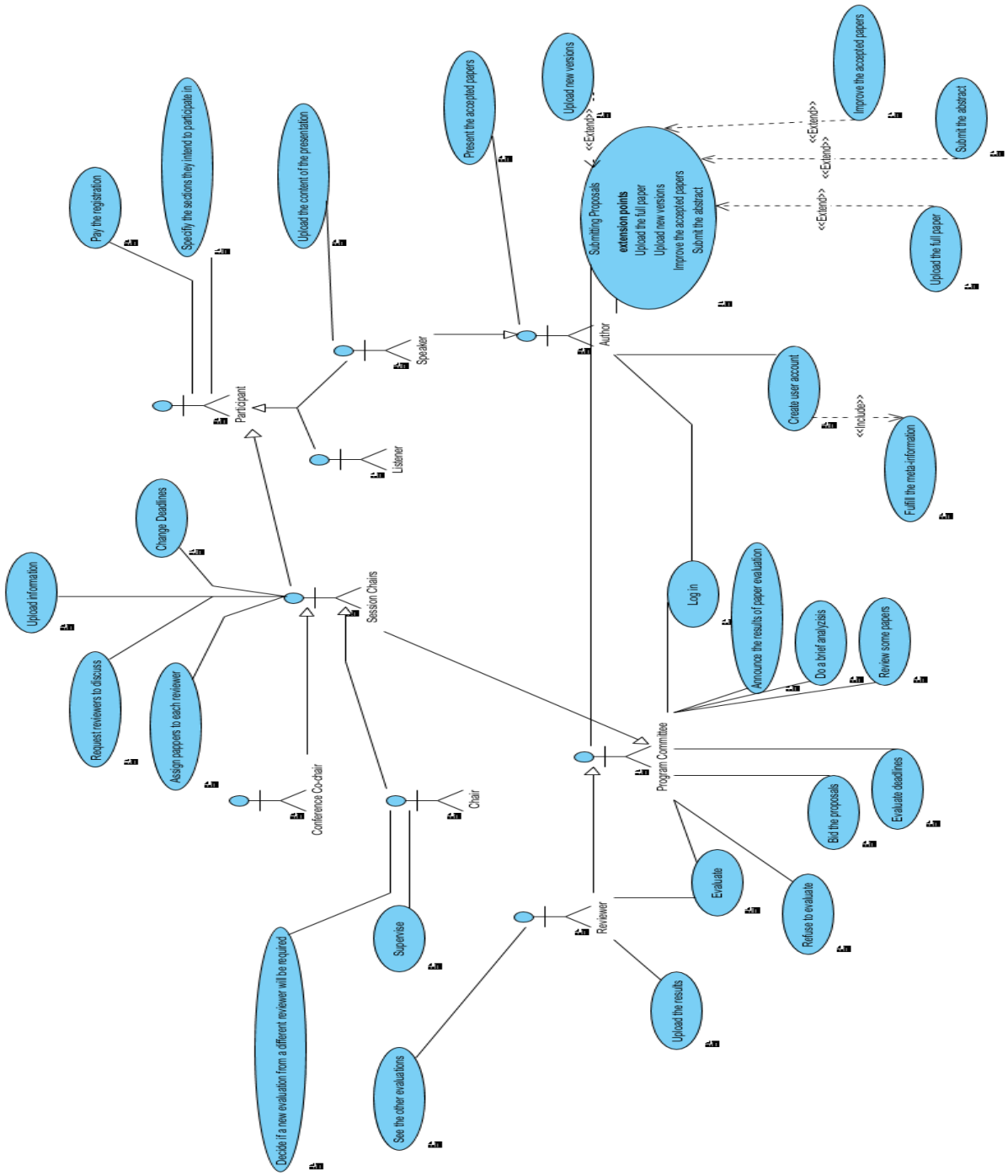
Modify the application.properties file by changing the username, password and name of the database.

During the development period, the tables for the database are generated at runtime with the help of Spring, meaning that every time you run the application the database is created again and data is not persisted.

When in production, if the database is to be modified changelocks will be made and liquibase will read them and execute them accordingly.

Liquibase is an open-source database-independent library for tracking, managing and applying database schema changes.

USE-CASE DIAGRAM



FLOW OF EVENTS

Use case name: Create user account

Participating actor: A person interested to participate at the conference

Flow of events:

1. The person enters the website of the conference

2. The person clicks the “register” button

3. The person is required to provide some personal information: name, email, password, entity type, web page and affiliation

4. If any information is invalid, the person is required to enter it again Entry condition: The person accesses the system from an internet connected device

Exit condition: The account was successfully created

Quality requirements: The email must be valid and the password must be at least 6 characters long

Use case name: Submit the abstract

Participating actor: Author of a paper

Flow of events:

1. The author is logged in into the conference system and wants to submit an abstract

2. The author enters the page of the conference for which he wants to submit the abstract

3. The author clicks the “upload abstract” button and adds the title and the content of the abstract

Entry condition: The author has a valid user account and is logged in into the system

Exit condition: The abstract was successfully uploaded

Quality requirements: The required fields cannot be empty

Use case name: Upload new versions

Participating actor: Author of a paper

Flow of events:

1. The author of a paper has already submitted an abstract/paper

2. The author wants to upload a new version of the abstract and/or of the paper

3. The author logs in into the system and enters the conference page

4. The author clicks the “update” button

5. The author can modify the content of the abstract or the paper

Entry condition: The author has a valid user account and has already submitted an abstract/paper

Exit condition: The new version of the abstract and/or of the paper is successfully saved into the system. The PC Members can see only the last version of these

Quality requirements: The content can not be modified to be empty.

Use case name: Change deadlines

Participating actor: Chair/Co-chair

Flow of events:

1. The Chair/Co-chair logs in into the conference system

2. The Chair/Co-chair clicks the “change deadlines” button

3. The Chair/Co-chair selects the event (e.g submission of proposals) and postpones it for later

Entry condition: The Chair/Co-chair has a valid account and is logged in into the system

Exit condition: The change of the deadline was successfully saved into the system

Quality requirements: The time of the deadline can only be changed for a later time

Use case name: Assign papers to each reviewer

Participating actor: Chair/Co-chair

Flow of events:

1. After the period for bidding proposals ends, the Chair/Co-chair takes into consideration the options of the PC Members

2. The Chair/Co-chair evenly assigns the papers to the reviewers

Entry condition: The Chair/Co-Chair has a valid account and is logged in into the system

Exit condition: Each reviewer receives an email from the Chair/Co-Chair having attached the papers to be reviewed

Quality requirements: The Chair/Co-Chair assigns a paper to at most 2 reviewers

Use case name: Bid proposals

Participating actor: Reviewer

Flow of events:

1. The Reviewer enters the conference page and sees all the submitted abstracts

2. The Reviewer selects certain abstracts to review

Entry condition: The Reviewer has a valid account and is logged in into the system, the abstracts are stored into the system in a list that can be accessed by all the Reviewers and Chair/Co-Chair

Exit condition: The changes are successfully saved into the system

Quality requirements: A reviewer can only choose if he wants to review an abstract or not.

Use case name: Evaluate

Participating actor: Reviewer

Flow of events:

1. The Reviewer enters the conference page and sees all the papers he has to review

2. The Reviewer gives to each paper a qualifier (strong accept, accept, weak accept, borderline paper, weak reject, reject, strong reject)

Entry condition: The Reviewer has a valid account and is logged in into the system

Exit condition: The qualifier for each paper is successfully saved into the system

Quality requirements: Each paper must receive a qualifier

Use case name: Present the accepted papers

Participating actor: Author of a paper

Flow of events:

1. The author submitted a paper that was accepted for the conference

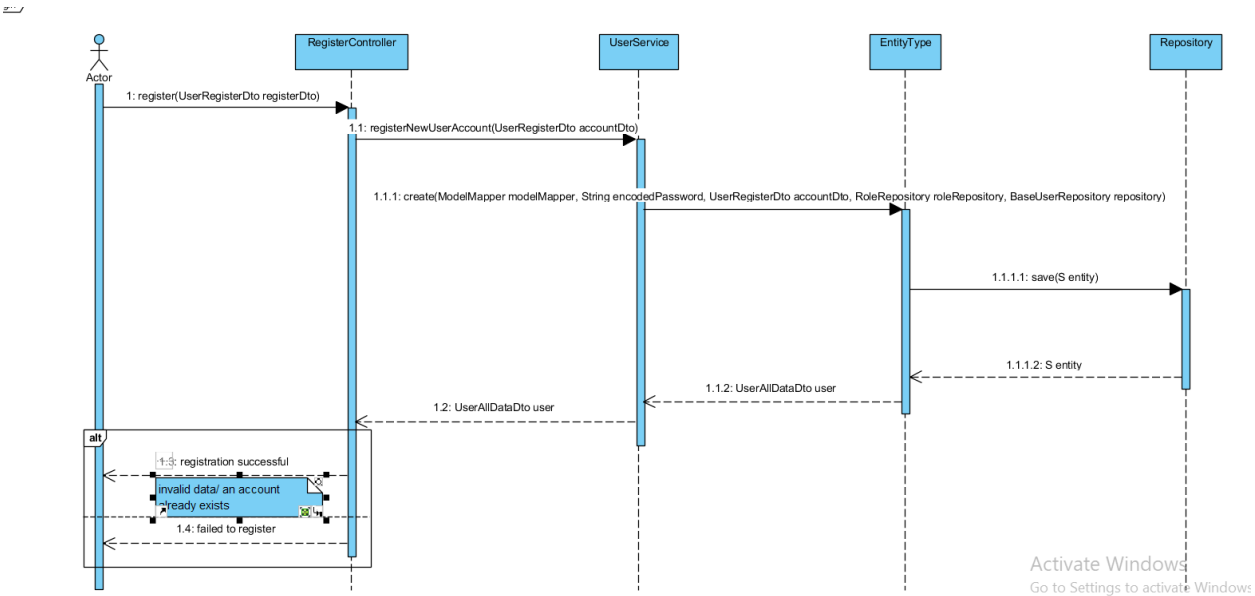
2. The author presents the paper at the conference

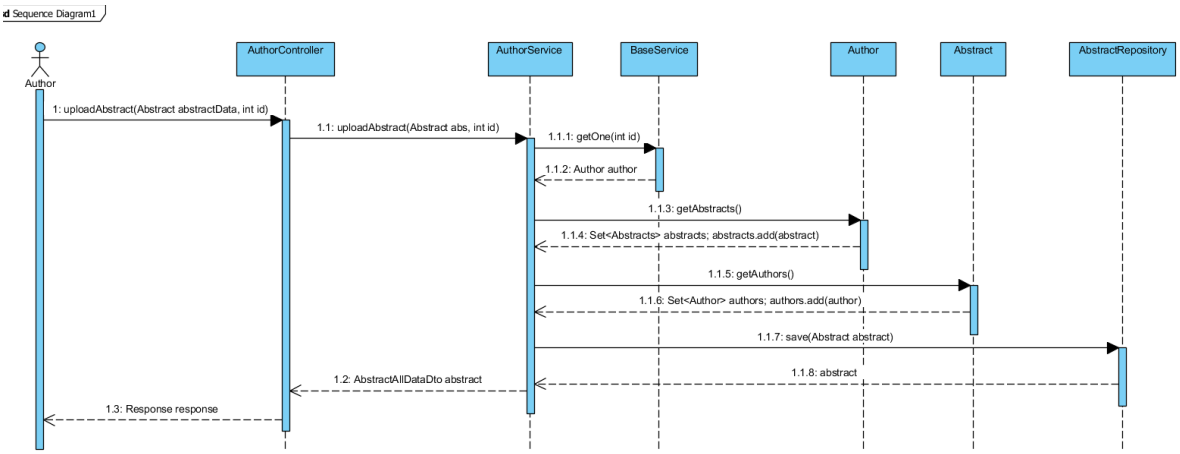
Entry condition: The author submitted a paper before the deadline

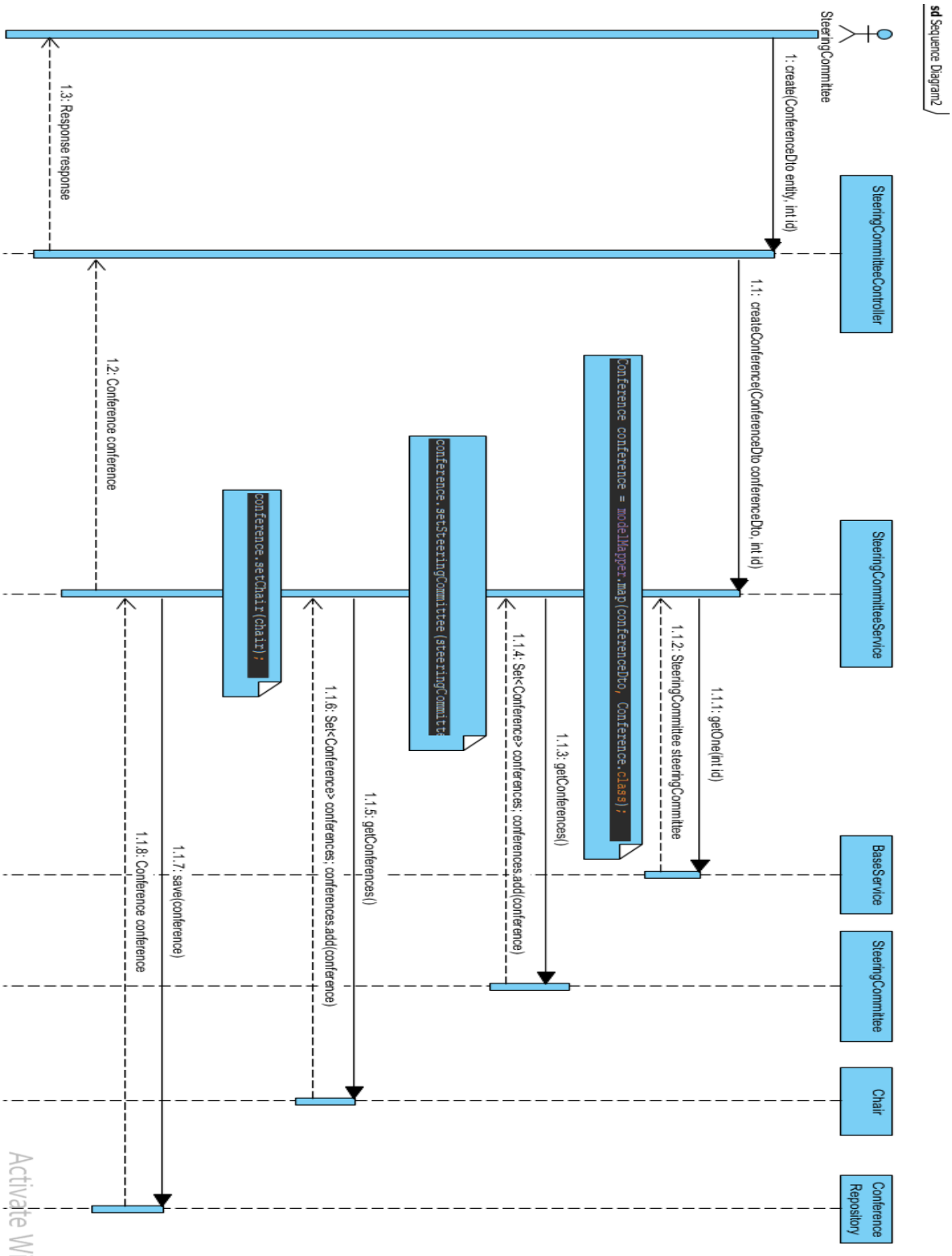
Exit condition: The author presents his paper in front of the participants at the conference

Quality requirements: The paper is presented in a relevant section of the conference

SEQUENCE DIAGRAMS

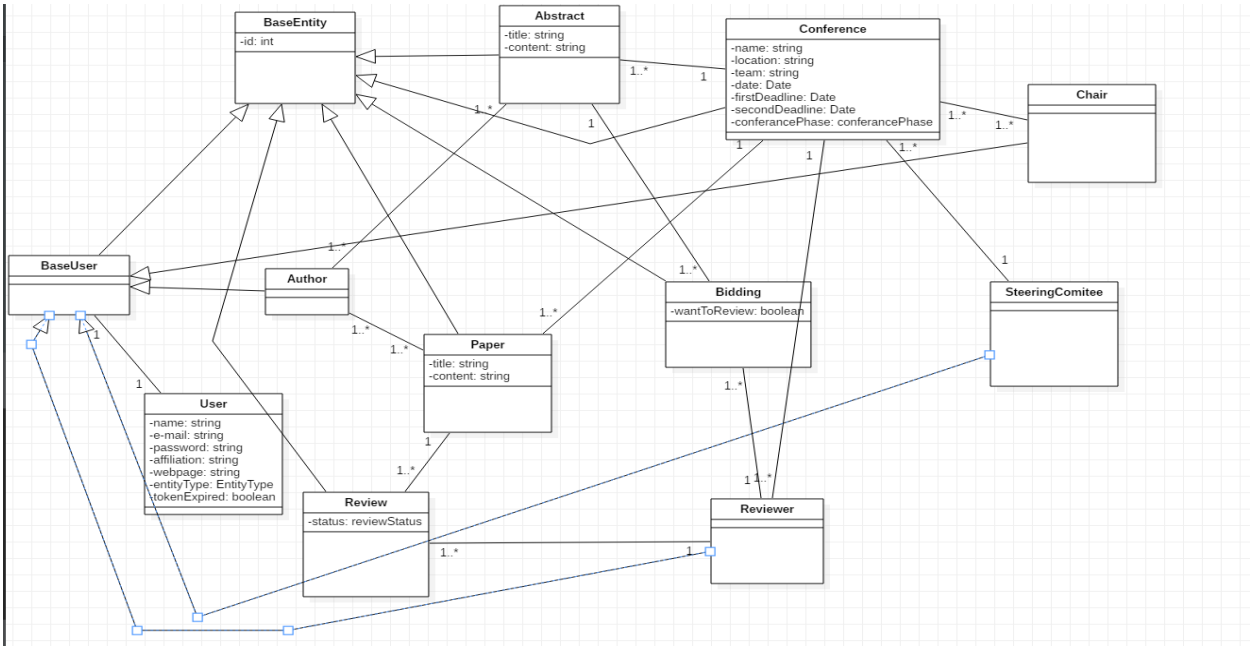




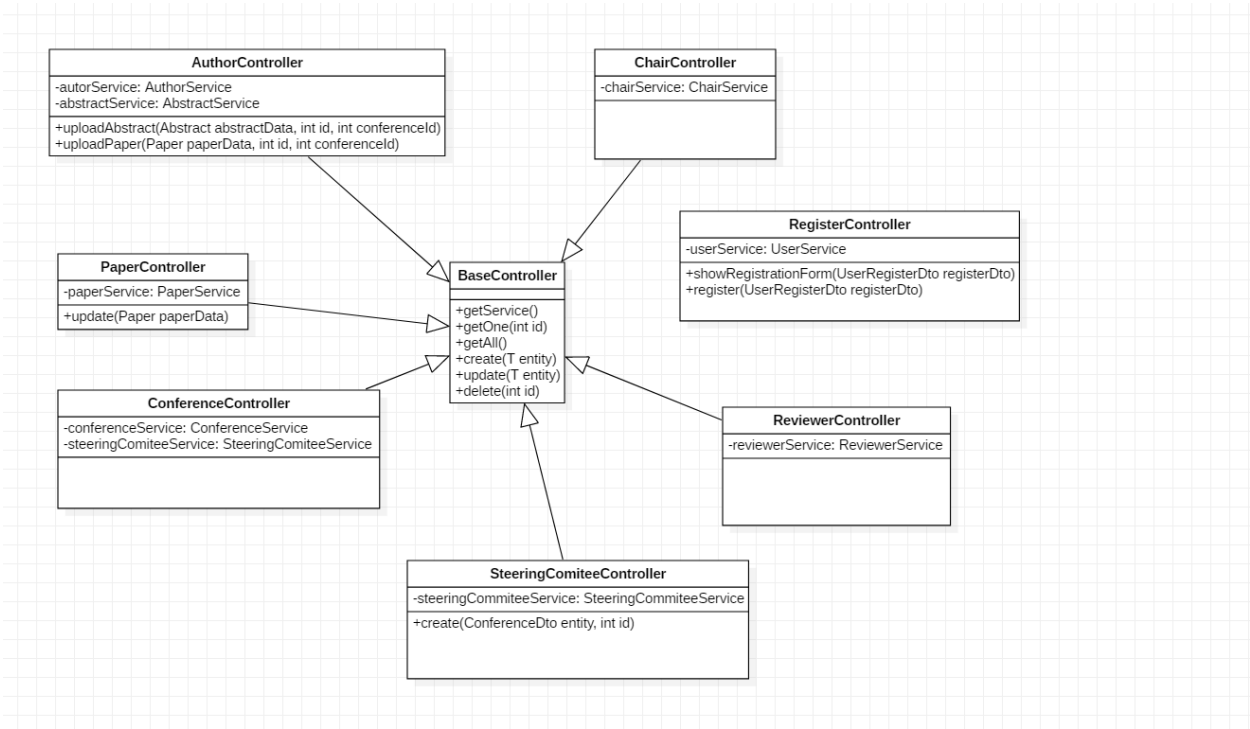


CLASS DIAGRAM

MODEL



CONTROLLER



DATABASE DIAGRAM