

1. INTRODUCTION

1.1 PURPOSE

The purpose of this document is to build an online Conference Management System to support the automatic management of information related to scientific conferences.

1.2 DOCUMENT CONVENTIONS

CMS = Conference Management System

PC = Program committee

1.3 INTENDED AUDIENCE AND READING SUGGESTIONS

This project is a prototype for the Conference Management System and it is restricted within the university premises. This has been implemented under the guidance of university professors. This project is useful to anyone who wants to implement such a system.

1.4 PROJECT SCOPE

The purpose of the Conference Management System is to help conference committees in organizing conferences, as well as paper authors in submitting their papers for review and scheduling speeches at the conferences in the appropriate sections.

2. OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

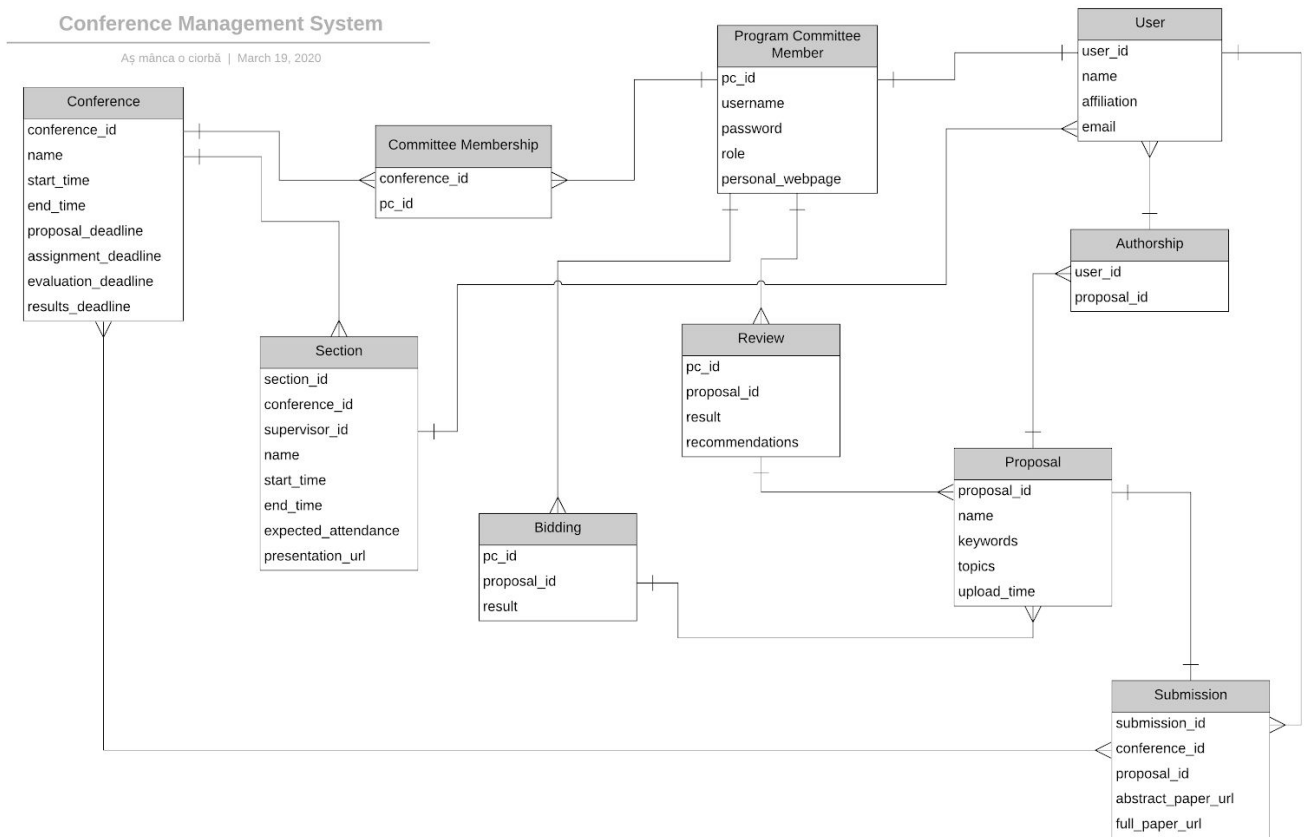
A relational database stores the following information:

- **Conferences:** the conference's name, the time interval in which it will take place, the deadlines, the members of the Program Committee, the conference sections etc.
- **Program Committee Members:** name, affiliation, e-mail address, personal web-page, username for the CMS, password for accessing the information (hashed)
- **Authors:** name, affiliation, e-mail address, submissions
- **Submission:** abstract paper, full paper(optional), proposal
- **Proposal (meta-informations):** name, keywords, topics, possible list of authors and their meta-information
- **Biddings:** the proposal and the result of the bidding (accepted/refused)
- **Review:** the reviewer, the result of the review, the reviewer's recommendations

- **Sections:** name, time interval, supervisor, nr of expected attendants, presentation(optional)

2.2 PRODUCT FEATURES

The major features of the conference management system are shown in the below entity-relationship model (ER model)



2.3 USER CLASSES AND CHARACTERISTICS

The system will support 1 type of user. The user will be able to create conferences and add other users to the committee and also submit proposals to his or other conferences.

The functions available for the users are defined in the use case diagram, which was uploaded to the repository.

2.4 OPERATING ENVIRONMENT

- client/server system
- Operating system: Windows/Linux
- database: MySQL database
- platform: Java

- frontend tech: Angular

3. FUNCTIONAL REQUIREMENTS

3.1 incipial phase

-Steering committee decides to organize a conference edition => information about the event is posted on the conference site (name, interval of taking place, "Call for papers", deadlines, the Program Committee, the conference sections and program, etc a.s.o.)

3.2 preliminary phase

-the PC members (including the conference chair and co-chairs) upload information (name, affiliation, email address, personal web page, username for the CMS and password for accessing information about the conference)

-the chair or one of the co-chairs have the permission to postpone deadlines

3.3 phase 1

-potential proposal submitters create a user account (name, affiliation, email address);

-account is validated

-submitter logs in, uploads the abstract and fulfills the meta required (name of proposal, keywords, topics, possible list of authors and their meta)

-depending on the conference, it is also possible to submit the full paper(.pdf or .ms)

-uploading proposals can be done before the deadline mentioned on the conference site on the "Call for papers"; deadlines can be different for the abstract and the proposal; until each deadline, users may upload new versions both for the abstract and the paper;

3.4 phase 2

-PC members bid proposals

-each PC member has to do a brief analysis of abstracts or papers in order to decide (review, evaluate, refuse to evaluate) a paper;

-bidding deadline is reached => conference chair or co-chairs assign to each reviewer(member of the PC) the papers required for evaluation (by default, you cannot assign to a reviewer a paper he refused in the bidding phase)

-each paper must be reviewed at least by 2 reviewers (usually 3 or 4, depending on conference level); result of review may be : strong accept, accept, weak accept, borderline paper, weak reject, reject, strong reject

-papers with no kind of reject are accepted

-papers with no kind of accept are rejected

-papers with mixed reviews, chair/co-chairs request reviewers to discuss in order to get closer evaluations; if not possible, it is up to the chair to decide : new evaluation from different reviewer, accepted or rejected

- the evaluation qualifier is justified by means of a set of recommendations attached to the evaluated proposal
- once an evaluation is uploaded, the reviewer will see the other evaluations on the same paper
- after deciding about the acceptance for all submissions, the authors are announced by email about their results
- papers accepted and presented on the conference will be published
- accepted authors are invited to improve their accepted papers according to the reviewers' recommendations
- PC members excepting the chair/co-chairs can submit proposals by logging in as authors; they cannot see the reviewers of their paper nor the comments between the reviewers

3.5 phase 3

- the conference is structured on different sections (some organised in parallel)
- each section is supervised by a session chair which in most cases is a PC member (incl the chair/co-chair); a speaker cannot be the chair of its own section!
- authors presenting their accepted papers are named speakers
- participants to each section are: chair, speakers, listeners
- listeners may be invited to specify the sections they intend to participate in in order to allocate rooms that are appropriate in size
- before the presentation speakers may upload on the conference site the content of the presentation (.pdf or .ppt(x) file)

4. NONFUNCTIONAL REQUIREMENTS

4.1 Accuracy

1. Every user has to provide valid and verifiable data when creating an account
2. Every user has to have a unique username
3. Every file (.pdf/.pptx) must have an unique identification.

4.2 Adaptability

1. The finished software must accept any type of new users without needing to be rewritten or recompiled .
2. The finished software must accept any type of new file without needing to be rewritten or recompiled .

4.3 Aesthetics

1. The user must be pleased with the GUI appearance.

4.4 Compatibility

1. All web client programs must interact with the server without any changes to the server code.

4.5 Configurability

1. The system must adapt to any changes in any input record format without the need to recompile the code.

4.6 Consistency

1. A consistent GUI is required.

4.7 Documentation

1. The source code shall be self-documented by placing the design description in a Javadoc-readable method header.

4.8 Frequency of failure

1. There can be no unhandled exceptions from user inputs.

4.9 Human Factors

1. The UI must be user-friendly.
2. The UI must be intuitive.
3. All menus must have a consistent format.

4.10 Maintainability

1. The maintenance group must be able to maintain the software.
2. Mean Time to Change (MTTC) for defects < 7 days.

4.11 Portability

1. The system will run, without modification, on any Web target platform.
2. Any Java-enabled web browser must be able to access the system data.
3. The software must be written character set-neutral.

4.12 Recoverability

1. During a system restart, the system will return to a functioning state.

4.13 Reliability

1. The system must never crash.

4.14 Resource Consumption

1. The system must consume a maximum of _____ Mbytes of memory.
2. The system will use no more than _____ of system resources.

4.15 Response Time

1. Query response time must be fast.
2. All queries must return a response in < _____ seconds.

4.16 Security

1. Every user must enter a password to use the system.

4.17 Understandability

1. 95% of novice users can learn to operate major use cases without outside assistance.