**Software Requirements Specification**

1. **Introduction**

1.1 *Purpose*

The purpose of this documents is to build an online system to manage scientific conferences and papers submitted by participants.

1.2 *Document Conventions*

DB - Database

ER - Entity Relationship

CMS - Conference Management System

PC - Program Committee

1.3 *Intended Audience and Reading Suggestions*

This project is a prototype for the CMS and it is restricted within the university premises. This project will prove to be useful for the committee organizing the conferences as well as for the paper applicants.

1.4 *Project Scope*

The scope of this project is to ease the scientific conference management and to create a convenient application for speakers who wish to submit papers to the conference. The system is based on a relational database with its paper management and conference functions. We will have a database server supporting dozens of conference papers as well as user information whether they are part of a committee, a speaker, or a listener.

1.5 *References*

* <http://webcache.googleusercontent.com/search?q=cache:https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database>
* <https://www.visual-paradigm.com/VPGallery/index.html>
* <https://openclassrooms.com/en/courses/4191736-design-a-database-with-uml/4191743-learn-about-class-diagrams>
* <https://www.infoq.com/articles/why-architectural-diagrams/>

1. **Overall Description**

2.1 *Product Perspective*

A management system for conferences which enables chairs, speakers and listeners to connect. Each speaker provides papers that will later on be discussed and evaluated by a committee formed by chair members. Papers can either be accepted or rejected. The rejected papers will receive feedback for improvement. The accepted papers will be further published and available to listeners.

* 1. *Product Features*

See the attached file.

2.3 *User Classes and Characteristics*

The system will support 3 main types of users with different privileges: chairs, speakers and listeners.

Speakers should be able to do the following functions:

* Submit papers
* Full (files via upload)
* Abstract (text)
* Meta information
* Edit papers
* Present papers
* Publish papers once they are accepted

Chairs should be able to do the following functions:

* Be part of a committee (program committee, steering committee)
* Evaluate papers
* Reject a paper and offer feedback
* Accept a paper

Listeners should be able to do the following functions:

* Enlist to a conference

2.4 *Operating environment*

Operating environment for the conference management system is as listed below.

//TODO

2.5 *Design and implementations constraints*

2.6 *Assumption dependencies*

1. **System Features**

* *Description and Priority*

The papers management system maintains information on user information, committees, conferences, conference sessions and authors, and of course, papers. This project has a medium priority since it can help organizing those scientific conferences and information on its participants and papers.

* *Response Sequences*

//TODO

* *Functional Requirements*

//TODO

1. **External Interface Requirements**

4.1 *User Interfaces*

* Front-end software: HTML, CSS, JavaScript + some JavaScript framework (TODO)
* Back-end software: SQLite

4.2 *Hardware Interfaces*

* Windows

4.3 *Software Interfaces*

* Operating system - we have chosen Windows operating system for its best support and user-friendliness
* Database - To save the scientific conference papers

4.4 *Communication Interfaces*

* This project will support all types of web browsers.

1. **Non-functional Requirements**

5.1 *Performance requirements*

The steps involved to perform the implementation of a Conference Management System are as listed below.

* 1. *E-R DIAGRAM*

The E-R Diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is then used to organize data as a relation, normalizing relation and finally obtaining a relation database.

* **ENTITIES:** Which specify distinct real-world items in an application.
* **PROPERITES/ATTRIBUTES:** Which specify properties of an entity and relationships.
* **REALTIONSHIPS:** Which connect entities and represent meaningful dependencies between them.

//TO DO: a diagram which shows the ER diagram of conference management system

* 1. *NORMALIZATION*

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

If a database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.

Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme. There are three different kinds of modifications of anomalies and formulated the first, second and third normal forms (3NF) is considered sufficient for most practical purposes. It should be considered only after a thorough analysis and complete understanding of its implications.

5.2 *Safety Requirements*

//TODO

5.3 *Security Requirements*

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

5.4 *Software Quality Attributes*

//TO DO