Software Requirements Specification Document

Table of Contents

**1.Introduction**

1.1 Purpose

1.2 Document Conventions

1.3 Intended Audience and Reading Suggestions

1.4 Project Scope

1.5 References

**2. Overall Description**

2.1 Product Perspective

2.2 Product Features

2.3 User Classes and Characteristics

2.4 Operating Environment

2.5 Design and Implementation Constraints

2.6 Assumptions and Dependencies

**3. System Features**

3.1 Functional Requirements

**4. External Interface Requirements**

4.1 User Interfaces

4.2 Hardware Interfaces

4.3 Software Interfaces

4.4. Communications Interfaces

**5. Nonfunctional Requirements**

5.1 Performance Requirements

5.2 Safety Requirements

5.3 Security Requirements

5.4 Software Quality Attributes

## 1. INTRODUCTION

* 1. **PURPOSE**

The purpose of this document is to build an app that supports the automatic management of information related to scientific conferences.

* 1. **DOCUMENT CONVENTIONS**

This document uses the following conventions:

|  |  |
| --- | --- |
| DB | Database |
| DDB | Distributed Database |
| ER | Entity Relationship |

* 1. **INTENDED AUDIENCE AND READING SUGGESTIONS**

This project is a prototype for the conference management system and it is allowed within college premises. This has been implemented under the guidance of college professors. This project is useful for the scientific community and people who want to share information in conferences.

* 1. **PROJECT SCOPE**

The purpose of the conference management system is to ease the sharing of information and to create a convenient and easy-to-use application for scientists or other intellectual people, trying to publish papers. The system is based on a relational database with its conference management and submissions, reviewing functions. We will have a database server supporting different types of app users (chair, co-chair, authors, etc.), uploading and editing of proposals composed of abstract/full paper and meta-information, reviewing and adding conferences with sections..

**1.5 REFERENCES**

[to be completed with sites that helped us on our way]

<https://spring.io/quickstart>

<https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database>

**2. OVERALL DESCRIPTION**

**2.1 PRODUCT PERSPECTIVE**

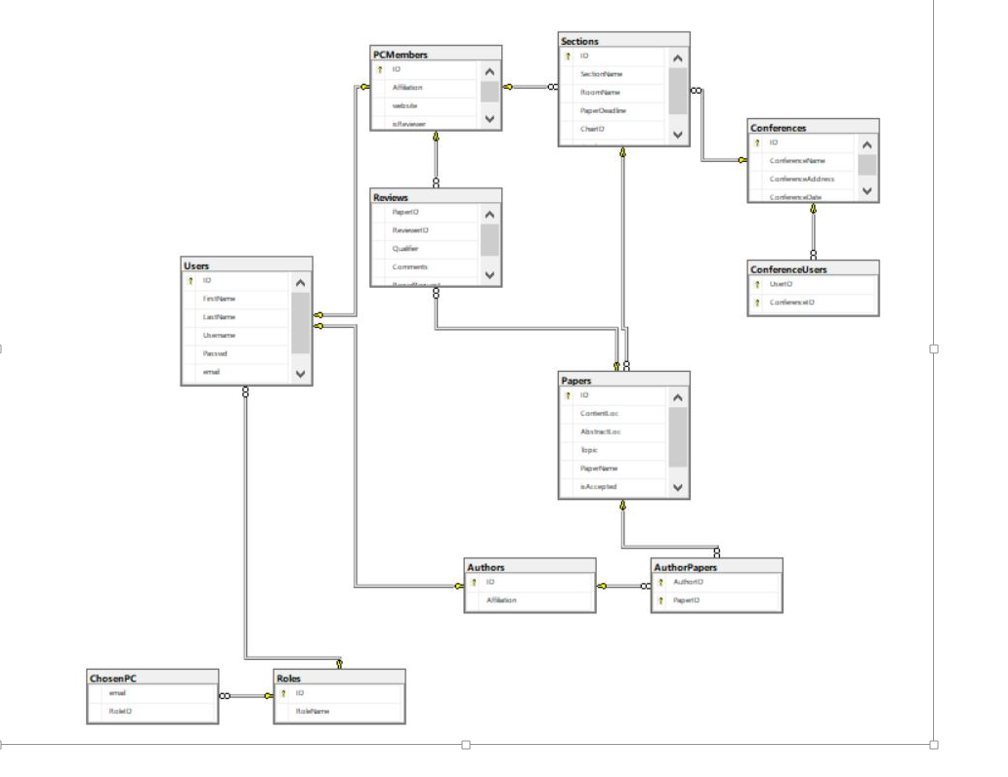
A conference management database system stores the following information.

* **Conference details:**  
  It includes the details of where is it held, when and its participants.
* **Sections description:**  
  A conference is made of more sections. The section incudes information about how many people take part in each section and what papers will be presented.
* **Papers description:**  
  It includes the abstract, full paper, meta information, the author submitting it, reviewers and other details about its content.
* **Users**

It includes the clients of the application which can be authors, reviewers, conference organizers, chairs.

**2.2 PRODUCT FEATURES**

The major features of conference database system as shown in below [**entity–relationship model**](https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model) (**ER model**)

****

**2.3 USER CLASS and CHARACTERISTICS**

Users are of different types.

The Author should be able to do the following functions:

* Create an account
* Log in
* Submit paper
* Upload full paper

The Reviewer should be able to do the following functions:

* Review papers
* Attach recommendations
* Upload evaluations
* See evaluation

The Chair should be able to do the following functions:

* Change deadlines
* Assign papers to reviewers
* Request reviewers to re-evaluate
* Request evaluation from reviewer
* Decide acceptance

The CMS should be able to do the following functions:

* Announce authors about evaluation results
* Generate accounts
* Assign session chairs
* Assign rooms for sections

**2.4 OPERATING ENVIRONMENT**

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

* client/server system
* Operating system: Windows
* database: MS SQLServer + database
* platform: Angular/Java
* relational database

**2.5 DESIGN and IMPLEMENTATION CONSTRAINTS**

1. Connecting the objects which are in one/many to many relationship. Managing deletion and insertion of new elements.
2. The database with a stable schema.

**2.6 ASSUMPTION DEPENDENCIES**

## 3. SYSTEM FEATURES

* **DESCRIPTION and PRIORITY**

The conference management system maintains information on proposals, reviews, submissions deadline, conferences and their participants. Of course, this project has a high priority because it is very difficult to present new scientific ideas face-to-face and be reviewed by multiple persons fast.

* **STIMULUS/RESPONSE SEQUENCES**
  + Login and Register pages show at first
  + Authors will be presented with pages that allow adding a Proposal with Abstract/Full papers and other content
  + Chair will be presented with proposals submitted by authors and will be able to connect a Reviewer and a Proposal. It can also change deadlines after which conferences with accepted proposals will start.
  + Reviewers will be presented with the assigned proposals, check their content and give an evaluation (rejected, accepted)
* **FUNCTIONAL REQUIREMENTS**

Functional system features include:

**CLIENT/SERVER SYSTEM**

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end).

A client/server system is a distributed system in which,

* Some sites are client sites and others are server sites.
* All the data resides at the server sites.
* All applications execute at the client sites.

## 4. EXTERNAL INTERFACE REQUIREMENTS

**4.1 USER INTERFACES**

* Front-end software: Angular
* Back-end software: Java

**4.2 HARDWARE INTERFACES**

* Windows.
* A browser which supports CGI, HTML & JavaScript.

**4.3 SOFTWARE INTERFACES**

Following are the software used for the conference management system.

|  |  |
| --- | --- |
| **Software used** | **Description** |
| Operating system | We have chosen Windows operating system for its best support and user-friendliness. |
| Database | To save client data and data about all the papers, reviews and conferences we have chosen SQL Server database. |
| Backend | To implement the project we have chosen Java for its wider use in real world. Spring Boot will help us with dependency injection, REST API endpoints and other logic problems in our program. We will connect to the DB using Hibernate which implements the JPA interface. |
| Frontend | We have chosen Angular framework which uses the Typescript language, CSS and HTML to help create a beautiful and dynamic design. Angular Material will also be used to improve the aspect. |
|  |  |

**4.4 COMMUNICATION INTERFACES**

This project supports all types of web browsers. We are using classic forms to add data and list views to show it.

## 5. NONFUNCTIONAL REQUIREMENTS

**5.1 PERFORMANCE REQUIREMENTS**

The steps involved to perform the implementation of conference database are as listed below.

**A) 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.
* **PROPERTIES/ATTRIBUTES:** Which specify properties of an entity and relationships.
* **RELATIONSHIPS:** Which connect entities and represent meaningful dependencies between them.

**B) 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**

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

**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.

The application must show some parts of the web site only to specific users. A login system with username and password should use JWT and other techniques to ensure that information is kept secure.

**5.4 SOFTWARE QUALITY ATTRIBUTES**

* **AVAILABILITY:** The submissions, reviews, logins, biddings should be available not matter how many clients are using the application.
* **CORRECTNESS:** The reviewers should not intersect and review more proposals than they have been assigned.
* **MAINTAINABILITY:** The conference managers should maintain correct deadlines for the submissions.
* **USABILITY:** The proposals submitting should satisfy a maximum number of authors needs.