CS 353
Database Management Systems Project

Project Proposal
Scientific Papers Data Management System

Section 3 / Group 1
Oguz Kaan Agac 21503895
Berke Deniz Basaran 21400996
Mert Özerdem 21300835
Ceren Uysal 215211972

Instructor
Cagri Toraman

INTRODUCTION

This report explains our proposal for establishing a Scientific Papers Data Management System. In this report we will discuss the aim of our project, its basic functionionality, the constraints and limitations, how it will be designed and how it is going to be implemented.

In the Project Description chapter we will introduce the properties of the Scientific Papers Data Management System and discuss why we need a database and how we can use it for our for this specific project.

In the requirements chapter we will discuss the functional, non-functional, pseudo requirements are explained. Functional requirements are requirements that specifies what system should do while non-functional requirements are for what needs to be done to satisfy those functions. Functional requirements are found by analyzing the user input. Non-functional requirements are mostly back-end requirements like an authentication system, performance, reliability etc. Pseudo requirements include the technologies that we are intending to use

In the limitations chapter we will discuss the constraints and the boundaries of the system.

Finally, we will provide a E-R diagram to concretize the system we are planning to implement.

PROJECT DESCRIPTION

Named Scientific Papers Data Management System, the focus is to keep the data available in a proper manner with fine query efficiency.

This database project aims to store, manipulate, and retrieve the information of people who serve as authors in academic environments, and the data of academic papers with certain attributes. The project will be implemented with SQL standard language.

Database will be available for several user types and these are described as follows: Author, editor, reviewer, subscriber. Author can add new academic papers to the system. They can be uniquely identified with ID issued by the system. Also, information such as proficciency, submission, position, and name are available for reliability reasons. Another user type is called reviewer and this user type can create reviews about existing journals. They have unique ID just like authors. Last user type editor also and assign reviewers to certain papers. ID is created also

for editors. Subscribers are not as active as the other user types and can join the notification group when new posts become available and existing ones are updated in the database. Each of the users are required to login to the system in order to access the functionalities described above.

Journal specifications are another important element of this database system. Each journal is unique due to the composition of title, date, and volume additional to increase the efficiency we also keep a unique id for each journal. The journals can be subscribed by the subscribers so that in case of any update, notifications may be received. Information about the editor of journal is also available for purposes of contact. Additionally, journals are assembled of multiple publications that have different identification numbers. Just like journals, the publications have title and date, yet these attributes are not composed to uniquely define a publication. Information of citation count and scientific area will also be available to stick with academic regulations and better querying. The publication review is allowed to be done by reviewers.

In the database, there is also information related to the conferences and institutions. Conferences are identified with specific date, name, and location. Each conference is concerned with a scientific area and each of them can be subscribed by the users. Papers and their authors are published in conferences, and each paper that is used in the conference is registered into database alıngaide their authors.

WHY DO WE USE DATABASE

Scientific Papers Data Management system is solely dedicated to collect and manage the data that's relevant to the academic environment. The main reason to choose this database is that there is no other data kept beside the authors, articles and other entities that are strictly related to the scientific area. In this case, the database at hand will give more efficient querry results when compared to the other database systems which are more extensive and do contain more unrelated information that is useless. Both time and space complexity will become less of an issue since all other data is excluded for the purpose of managing the data of academic entities. At the end, this system will be more preferable for the social networking websites/applications for several academic people to share papers and find collaborators.

HOW DO WE USE A DATABASE IN OUR PROJECT

Our database model is going to provide a way of managing the general and the specific data for the scientific papers. The general data includes such as the name of the authors (or co-authors) of a paper or the journal/conference that the paper is published on. More specific data includes the citation count for a paper, and subscription count for a journal or a conference. Authors or institutions can add their papers to the system after going through required authentication process. Users can search papers by the author, scientific area (such as CS,

Biology etc) and the journal/conference. We will also provide option for sorting papers by the citation count.

REQUIREMENTS

FUNCTIONAL REQUIREMENTS

Users will be separated into five different categories; authors, reviewers, editors, institutions, conferences. Users need to go through the authentication process to change the records of their papers.

User

- Users can be subscribers, reviewers, editors, authors or institutions.
- Users must have a unique email address.
- Users must have a password.
- Users will have an authentication level assigned to them.

Subscriber

- Subscribers can subscribe to conferences or journals.
- They can receive notification related with their subscriptions
- Subscribers have the lowest authentication level (cannot edit/remove/change)

Author

- Authors can add their paper to the database.
- Authors can edit their papers.
- Authors can submit their paper to as many conferences or journals they wished to.
- Authors can search for other publications.
- Authors can see the upcoming submit date of journals and conferences.

Editors

- Editors have the highest authentication level.
- Editors can add/remove papers from conference/journal lists.
- Editors can assign reviewers to papers

Institution

- Institutions can register multiple authors under their name.
- Institutions can publish journals.
- Institutions can hold conferences.
- Institutions must have a valid address.

Conference

- Conferences must hold a list of submitted papers.
- Conferences must have a valid address.
- Conferences must have an editor.

Journal

- Journals must hold a list of submitted papers.
- Journals must have an editor.

Paper

- Papers can be submitted to
- as many conferences or journals as user wishes to.
- Papers can be published only once.
- Papers can be published by only a single journal / conference.

NON-FUNCTIONAL REQUIREMENTS

Reliability

Data in the system must be stored with taking high security regulations to account. For the academic regards, system must be sufficiently preserved from any malicious acts. This requirement can be called as robustness

Performance

There must be a strict balance between the performance and reliability. Query efficiency must be considered so that system will be more preferable among the others. To ensure better query results, tables must be created with sufficient planning. Responsiveness is another way to

define this requirement.

Usability

User interface of the system must be less challenging for the database users. There must be enough documentation and GUI elements provided with the mainframe so that users can navigate in the system better.

Supportability

System can be easily maintainable and adaptable to the necessary change. With enough modularity, modifying the system will be a less challenging task and different faculties/subsystems in the system will be managed easily.

Concurrency

Concurrency is a real problem that must be dealt with. Especially publishing and submitting operations must be atomic meaning that they must be completed entirely or discarded. Cascading abort and cascading removal must be handled with care especially when dealing with paper status'. Once a paper is published it should be removed from other journals' or conferences' submitted list.

Availability

System will be available for at any time of the day and will be maintained regularly. With proper time scheduling, the system will be serving to any kind of users for the full portion of a day. In case of any maintenance, there will be notifications that warn the users about unavailable system time period.

Backup/Recovery

System files shall be saved according to the time schedule. At each checkpoint, system files will be recorded so that the harma causes by the malicious programs will be system will be resisted and no permanent damage will be occured with the system files. The files will be stored in a cloud environment directed by the database administors.

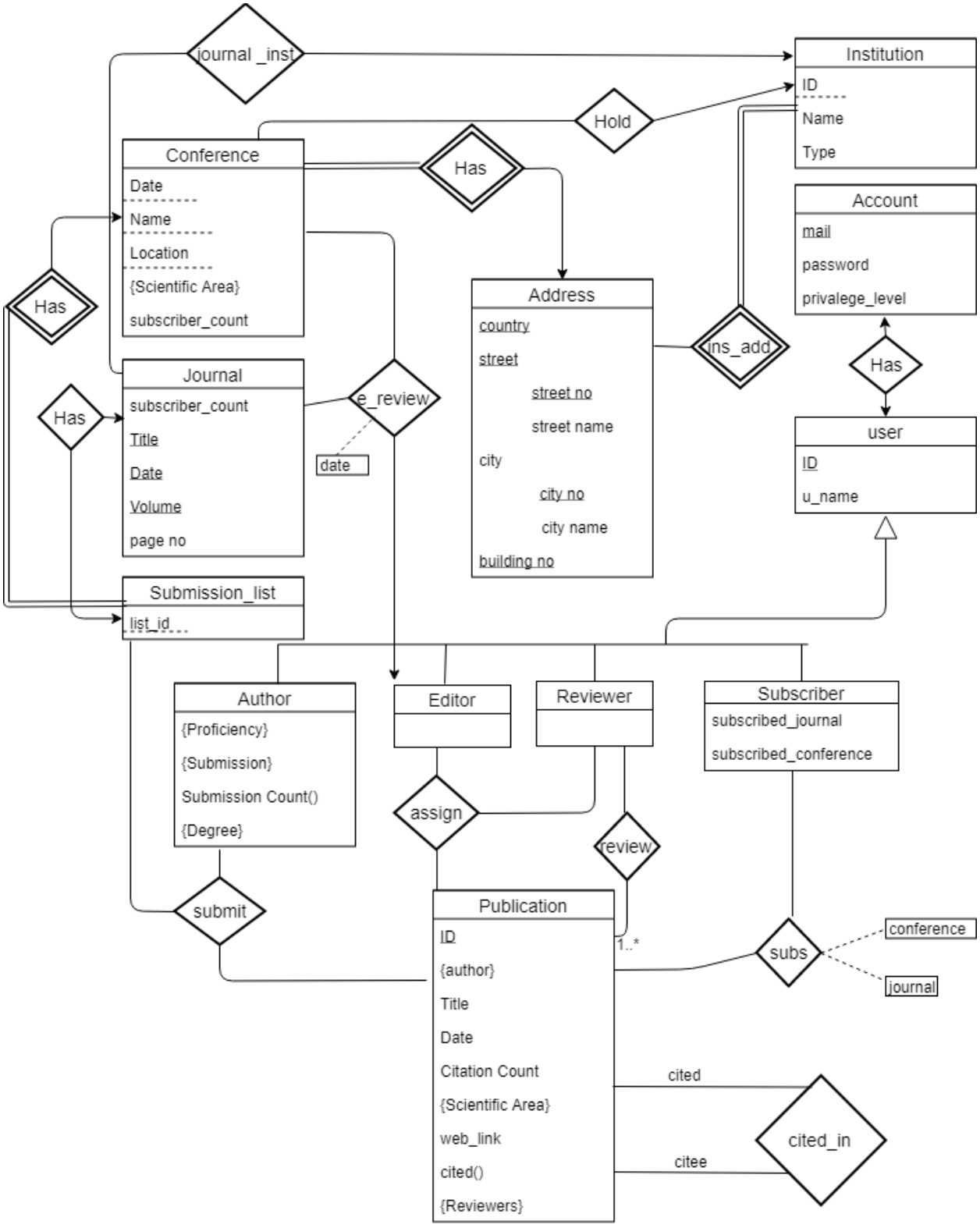
PSEUDO REQUIREMENTS

- MySQL will be used as the query language.
- HTML, Javascript, Django Python, Bootstrap will be used to implement the web service.
- PHP will be used for the back-end.

LIMITATIONS

- Papers can be published only once.
- Every conference and journal must at least one editor.
- For the papers, authors and date cannot be null.
- Conferences must have valid address.
- Journals must have a valid institution.
- Institutions must have valid addresses.
- User emails must be unique and valid.
- To subscribe, one must create an account.

ENTITY RELATIONSHIP DIAGRAM



WEBSITE

The link of GitHub page of the project is the following:
https://github.com/cerenuysal/CS-353