CSC 4402 Project Management Plan

Sorority Register Database Project

April 25, 2017

Hardika Patel

Javier Rodriguez

Chris Schayer

Will Raziano

**Table of Contents:**

**Introduction……………………………………………………………………………………....3**

**Project Overview and Organization.……………...……………………….…………………...3**

**Requirements…………………………………………………………………………………….4**

**Tools and Techniques……………………………………………………………………………5**

**Queries…………………………………………………………………………………………....6**

**Commentary/Conclusion………………………………………………………………………..6**

**Introduction**

The current member cataloging system that is in place for many sororities is based on a file system, which is maintained by members of the sorority on a voluntary basis. This system is very archaic and difficult to organize, maintain, and use. The methodologies we would like to implement into a cataloging system are based on fundamental database design and integration. This modernized design will allow for useful querying of the data, ease of maintenance, and future encryption of the entire system. This change will allow the use of a website to apply an application layer to the database enabling naive users full access to information as needed. Overall, the implementation of a database design to this already existing data set will be a vast improvement over the system currently in place. Our project, “Hermes” is a database implementation that we hope will be adopted in lieu of the current, archaic system.

**Project Overview**

We used a holistic approach to the problem set, creating a sample set of fictitious names to populate the database. This can be accomplished be several means, but we employed a Java program to pull the names from random locations on the Internet, and then used those names to fill the tables with fictitious names automatically by mixing first and last names from those websites randomly. We used a logical approach to set up the data in the tables, with meaningful information readily available for querying. Entity Relationship diagrams were used for the initial design of the tables. The primary key we used for the tables is the individual’s student identification number. This number would be kept for all tables indefinitely, so it is ideal for querying. In addition, the student ID is a super key and a candidate key. Our database contains four tables, one for active members of the sorority, one for new members pledging to the sorority, and one for alumni members of the sorority. Lastly, we created a fourth table based upon the active and pledge members to create “buddy” pairs. This fourth table is needed because it is common practice for the sorority to pair new members with existing members, and creating the table as a separate permanent table would increase the efficiency of querying that information substantially. We used this pairing information in order to know which existing members needed a protégé, and which new members needed a mentor.

Each group member contributed by making suggestions for the different aspects of the project, keeping the overall software design team structure an egoless team approach. There was no team leader, and all responsibility was divided up equally among the team members. While the project was divided up among all members of the group in an egoless team approach, each member was charged with an aspect of the project for which he or she could enlist the help of other team members as needed. Hardika handled the implementation of the website for the naive users, and implementation of the database in a server environment, Javier handled the creation and design of queries for the database administration, Chris handled the population of the tables and requirements evaluation, and Will created the documentation and organization of the presentation materials.

**Requirements**

Based on the given data that is currently collected, and the desired database design, the following requirements were generated for the project:

* A website for simple querying of information for naive users
* A location for the website and database
* A table to hold new member tuples:
* Student ID, first name, last name, major, year, high school, mentor name, graduation year
* A table to hold existing member tuples:
  + Student ID, first name, last name, major, year, high school, protégé name, graduation year
* A table to hold alumni member tuples:
* Student ID, first name, last name, major, high school, graduation year
* A table to hold mentor-protégé tuples:
* Mentor student ID, protégé student ID

**Tools and Techniques**

We used an Amazon EC2 instance to hold the web server and a MySQL server. The tools used for this project included IntelliJ, Eclipse EE, MySQL, Apache, Tomcat, Boot Strap, Sublime Text, GitHub, MS Office, and the application GroupMe for coordination. Our GitHub repository page can be accessed at: <https://github.com/wilraz/CSC4402> . Our webpage is currently private, but will be made public after appropriate security measures have been implemented, which is outside of the scope of this project. In addition, the implementation of this project at a business level requires approaching the targeted organization with a completed project, which includes creating a cost analysis of the project for purposes of maintenance plans, initial creation of the assets, and compensation of the team members. The business aspects aforementioned are also not within the scope of this project. The use of the Entity Relation Modeling technique was employed to create a design for the structure of the database. The ER Diagram we used is as follows:

*DIAGRAM HERE*

**Queries**

The queries that we implemented in this project included the following:

1. Create a query that will find available mentors for a new student.
2. Create a query that pairs a mentor with a protege.
3. Create a query that will move a student to the alumni table, altering the appropriate fields.
4. Create queries that will create lists for inviting members to functions.
5. Create a table listing mentor protege pairs, should be a standing table, not an intermediate table

5a. Weak relational table, mentor\_id references Members & protege\_id references Applicants

5b. No Primary Key in mentors\_proteges table

1. Create a query which lists all the members which last name starts with a C and contain a M in the first name.
2. Create a query which lists all mentors and proteges which belong to the same university department.
3. Create a query which lists all members with ID greater than 89...150 and lists all alumni with ID greater than 50.
4. Create a query which counts all applicants in the Computer Science department.
5. Create a query which lists all members which are in either 1srt or 2nd year of college.
6. Create a query which lists all members which are in their 4th year of college and belong to the Computer Engineer department.
7. Create a query which lists mentors id in the 4th year of college with proteges in their 1st year of college.
8. Create a query which lists mentors in their 3rd year of college and have no protege.
9. Create a query which creates a view which lists all applicants having rank less than 5.
10. Create a query which creates a view which lists all applicants with a partner id.

**Commentary/Conclusion**

Through the creation and implementation of this project, our team learned the value of utilizing the database model for the organization of data into a more accessible form. The improvements include the ability to query the data in order to create web views for the naïve user, the ability to manipulate data quickly and easily, and the ability to organize the data into a more structured format. The use of the techniques we learned in class aided us tremendously in the design of this project. Through the use of ER modeling, the database design presented itself with very little effort and logical thought, while its implementation remained fundamentally sound and logical in design. The querying of the stored data into valuable, meaningful information is greatly facilitated through the use of SQL. This factor alone makes the learning of SQL a great tool for anyone pursuing a career in Computer Science.