# CS374 – Intro to Database Management

# Application Development Project

# Rubric for Second Deliverable

## Group Member #1: Brian Ault\_

## Group Member #2: Samuel Wynsma

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Requirements | Points | Awarded |
| Description of Application | * An overview of your application * System requirements (e.g. hardware, DBMS, other software) * A detailed description of your application * Are there features that will not be implemented? What are they, and why won’t you fulfill them? | 10 | 7 |
| Project Management -Schedule | * Detailed schedule of who will do what part of project, by when | 5 | 5 |
| Logical Diagram | * Logical diagram in UML or E-R * Discussion of how your data model will satisfy the needs of your application * Discussion of alternative designs that you did not do (and why) | 15 | 10 |
| Queries Required | * Required queries in English (not SQL) * What entities and/or relationships are required for each query? * How will each query satisfy the needs of your application | 15 | 15 |
| Grammar, punctuation, syntax, and references | * Follow rules from the Penguin handbook on writing * References as appropriate (e.g. if you are modeling your application after an existing application, make note of that) | 5 | 5 |

Overview

Create a mock version of the Whitworth class registration system in a C# application utilizing SQL.

System requirements

We need a database that can be accessed and updated from the multiple computers at the same time. For this reason, we can rule out Access as a database to use. We could use a no SQL database but currently we are more comfortable designing both tables and queries for an SQL database. For that reason, we will be using an SQL database.

Additionally, we need to have some web access to the SQL database that way things can be altered from multiple computers.

Description

Currently we have a C# application that allows students to create accounts and register for classes. The issue with this current app is that it is fileIO based instead of DBMS based. We intend to update the app to utilize a SQL server database to allow usability to multiple computers.

There are three main things that we want to update. First is the information stored on classes. Currently we only store the class department, class number, class name, number of credits, and professor. We want to add class time, class days, class description, and registration size limit along with waitlists once the limit is reached. When it comes time to register we want to make sure we enforce that a student cannot register for multiple classes offered at the same time and that students will get sent to the waitlist if the class is full.

Secondly, we want to update the student. Mainly we want to collect a little more information such as phone number, class standing, advisor, and we would also like to generate an email address for them.

See “Queries Required” for needed queries.

Project Management Schedule

Nov 19 – Have database created with all pages implemented (Sam/Brian), finish plans on how to store information in memory (Sam/Brian)

Nov 27 – All UIs built (Brian), Web page up and running (Sam)

Nov 30 – All queries implemented (Sam), Queries are working with the application (Brian)

Dec 3 – Working prototype, essentially all functionality has been implemented and we are debugging and refining from here on out.

Dec 10 – Final version of project complete, Begin presentation

Logical Diagram

\\CS1\students\2018\bault18\CS374-1\Final Projet\Database UML V.1.png

We feel that this is the best way to store information while limiting redundancies. This relation allows us to collect all the required information for each person in the database and connect it in a meaningful way. Also, this is made to limit access to certain information by each entity. For instance, professors and students should not be able to access each other’s passwords or usernames.

Currently we are still in debate about how to model “Department” in our Schema. Classes need to have a department for their department initials (ex CS) and Professor needs a department so we know where he works. We just have not decided if we should make an entire separate Department entity so that if there are no professors or classes for it, it does not cease to exist.

Other than that we have not been able to see a much different way to implement this database.

Queries Required

For Students

Currently registered classes

When students log in the first thing that they see is a list of classes that they are currently registered for. We will need a query that pulls the courses they are registered for, using their ID number, from the “Registration” table and display them in the app window.

Searching for classes

Students, when looking up classes to register for have a few options to fill. Things like class department, class number, and class title need to be implemented into a query that will scan through the “Classes” table in the database and pull all courses that match the students search parameters. If no classes match the parameters that the student is looking up, then we will display an error message and let them know to refine their search.

Edit account info

This query will pull the student information, items such as major, phone number and address, into memory where the student can edit them. Once the student is done updating their information in memory it is pushed back to the database and saved.

For Staff

List of advisees

Advisors should be allowed to view advisees (students). They will connect through the “Advising” table. This query will populate a page in our app that allows the professor to choose a specific student and pull up more information about them.

View Student Information

Once the advisor selects the advisee to view all the student information except for password and username. That means they will be able to see all the classes the student is registered for along with all their personal information.

Courses Teaching

Professors will have a tab on their page that allows them to see which courses they will be teaching. This page will display all information about the course such as class time, class name, and number of students registered for the course. Note, as of right now we do not have plans to allow professors to access who is on the waitlist. Should we finish everything else we will explore allowing the professors to register beyond the class limit from the waitlist.

Edit account info

This will be the exact same as it is with students except there is different information stored for professors so we will populate this page appropriately from what is saved.

For Both

Login information

This query will test the user entered information against the accounts stored in the database to see if one, the user exists, and two, if their passwords match that in the database. If it works correctly we can use this query to deny access to the user and provide them information as to why their login failed. As for entities involved in this query, we plan on making sure that students and professors have separate login options so that their tables will be accessed separately.

References

We will be loosely basing our system off the Whitworth University class registration system that exists on their WhitNet page that can be accessed through Pirate Port.

Additionally, we will be using some of the Whitworth University logos in our application. The guide to using the logos, and the logos themselves, can be found at: https://www.whitworth.edu/cms/administration/university-communications/brand-and-identity/