CS348 – Project – Stage 1

You will develop a database-backed web or mobile application. The goal of this project is to use course concepts in a real application. The project will allow you to practice most of the following concepts:

- Database design (Entity-Relationship Diagram (ERD) and Normalization).
    - Indexing.
- A database query language (most likely SQL).
    - Stored procedures in databases.
- Using a database query language in code (e.g., using SQL in Python or Java).
- Transactions and concurrency.
- Cloud databases

The project will follow a self-learning approach. You will need to choose and learn about a programming language and a web framework to develop your application. The project evaluation will focus on the back-end of the application (especially code and SQL used to access the database). Therefore, **any graphical user interface will be accepted** as long as a regular user can utilize the features of your application.

Sample Application

This is a sample application that presents example database and features. This application serves student clubs management and regular students. Student clubs organize different types of meetings. Meeting organizers will be able to create a meeting and to invite students to the meeting. Students will be able to RSVP (yes, no, maybe). The system provides reports, such as a list of attendees for a specific meeting and statistics regarding the meetings in a specific period of time.

Note: the following project description is not complete and may miss important information, such as a detailed explanation of project features and database design.

Database Design (sample data organized in tables):

Students(student_id, name, email)
Meetings(id, date, time, duration, description, club_id, room_id, invitedCount, acceptedCount)
MeetingOrganizers(meeting_id, student_id)
Clubs(id, name, address, description)
Rooms (id, building, number, maxCapicity)
RSVP(meeting_id, student_id, invitation_date, reply_date, replyType, comment)

Main Features:

- Add, edit, delete features for most tables (e.g., add students or rooms to the database).
    - All features (especially delete) should maintain the consistency of the database (e.g., should not delete a student that attended a meeting before).
- Create a meeting (a page that allows you to insert a new row in the meeting table). The page allows the user to choose students to be the meeting organizers, which results in adding new rows in the MeetingOrganizers table.

- Add students to a meeting. A page (or multiple pages) that allows the user to select a meeting from the meeting table (i.e., the page allows the user to search for a meeting by id, date, or club). Then, the user can search for students (by name, department, … etc.). The application should allow the user to send email invitations to some of the displayed students (e.g., each student has a send-invitation button next to their name). Sending an invitation updates the invitedCount in the meetings table.
- A page that allows a student to respond to an invitation (yes, no, maybe) and write a comment.
- Example reports:
    - Meeting stats: A page that allows you to filter meetings by date (From a start date to an end date), club, and room. The application then generates a report of returned meetings, including average duration time, average number of invited students, average number of accepted invitations, and average attendance rate (attended/invited).
    - One student report: allows the user to search for a student and get a report that includes the number of meetings invitations, the number of acceptances, a list of clubs with number of meetings the student attended for each club, a list of rooms and the number of meeting in each room.

Project Size and Complexity.

Your database should contain between five and ten tables. Your application needs to support add, edit, and delete operations for your tables. Also, you should have at least a few features that touch multiple tables (e.g., the RSVP feature updates the acceptedCount in the meetings table in addition to inserting a new row in the RSVP table). Your application should include at least four reports that require complex queries (join queries, aggregations, and sub-queries). These reports allow users to filter the data included in the report. For example, in an e-commerce website, you can enter a price range, product category like electronics, and rating range (e.g., 4 to 5) to generate an aggregate report of the products that match the specified criteria.

How many members does each group include? How can I choose my group members?

Each group should include three to four students. You can choose your team members (members are not assigned by the instructor). If you would like to look for team members who share similar interests to yours (e.g., interest in an application domain or a database system), you can use campuswire chat rooms to advertise your project idea or development framework. For example, you can create a room with the title “club-meeting application, PostgreSQL database, and Node.js.” If you cannot find teammates, the instructor will assign you to a team.

Please visit the following shared Excel sheet to **write your group members**. If you do not have a group, **write your name**. The instructor will group individuals in groups. Please **write your information by the end of 9/15/2021.**

https://purdue0-my.sharepoint.com/:x:/g/personal/hbenotma_purdue_edu/EdZV2aoO35pMjzzeYjNGADQBQiIqKiR5Gi2mo-EDrwBRLw?e=CQgBPO

Can I use any database system?

You are encouraged to use a relational database system (because we study those in more detail). However, you can use any database system that supports a query language, transactions, and stored procedures (or provides equivalent features). It is important to list all systems, languages, and tools you will use in the shared spreadsheet (described below) so the staff can approve your choices. We will contact you in case we have comments.

Project stages and deliverables:

Stage 1:

In this stage, you will need to form a team and select a project domain/idea. Please do the following:

1. Write your information in the Excel sheet (link is above). In the spreadsheet (in a new row), write your group name, team member names, project title, database system you will likely use, programming language, and a short description of the project (3 to 4 sentences). Please note that all project information including the team members can be changed until the due date of stage 2. This is totally expected as we progress in the semester and you learn about new concepts and tools. List your group even if you do not have other members. The instructor and the TAs will merge small groups together. We may be able to merge groups that share similar interests.

2. A TA will be assigned to your group. The TA name and email will be listed in the **project-teams** Excel sheet described in the previous point.

3. **Create a shared folder** for your team (e.g., in OneDrive or Google Drive). In your folder, create the "Stage1" document. The stage1 document should not exceed two pages and should include the following information:
    a. Group name.
    b. Group member names and Purdue email addresses.
    c. Project title.
    d. Project description, including the main features that your application will provide.
        i. List data items you think you will include in your database. You can organize your data in tables. However, a detailed database design is a requirement of stage 2.
    e. List tools you think you will use in the project (DB system, programming language, frameworks and libraries). You can change those in the coming weeks (until the due date of stage 2).

4. Start exploring database systems, programming languages, and web frameworks.

Notes about the programming language and tools used for development.

1. The instructor and the TAs may use Python and/or Java for SQL-in-code examples.
2. You will be required to use different approaches to access a database from code. The approaches are raw SQL, stored procedures, and Object Relation Mapping (ORM) (if ORM is supported in your language/framework). Each approach has to account for at least 25% of your database access. It will be useful to check whether your language/framework supports these approaches (e.g., even

though Django main approach is ORM, I think you can also use raw SQL). More details about this requirement will be covered in week 6 (SQL in code).

3. You may contact your TA to ask for help. However, the instructor and TAs will not be able to assist you in coding. You can ask for help by posting a question on Campuswire. Multiple teams or members using the same tools are welcome to help one another by answering questions on Campuswire.

Stage 2:

In stage 2, you will design your database and user interface. You will also start development by setting up your database system and implement a few pages to get familiar with the programming language/framework. The main deliverables for stage 2 are:

1. Create a document in your shared folder to describe your ERD and database design (you will use concepts covered in weeks six, seven, and eight).
2. Create a document in your shared folder to describe your user interface design. Provide sketches of the user interface of the main features and reports of your application (e.g., you can use PowerPoint to draw how a page or report will look like when you finish development).
3. A proof that you have a database installed and data entered and a proof that you have some basic features in your application (e.g., a page for inserting new students). In your shared folder, include a url to your application or a 5-minute recorded demo (e.g., show that you can enter data in your application and then you can retrieve it).

Grading:

Stage 2 counts for 30% of the project grade. Your assigned TA will grade your ERD, database design, and user interface design. The TA will also make sure you have set up a database and you have a few pages (basic features) working.

More details regarding stage 2 will be posted when the stage is assigned.

Stage 3:

The main deliverable of Stage 3 is a final demo of your project. You will record a 10- to 15-minute demo of your project features. You will describe parts of your code where you implemented course concepts (e.g., different types of SQL queries, stored procedures, using indexes, transactions, isolation levels, … etc.). You will also discuss the lessons learned during the project phases. The other deliverables are a url to your application (if available) and your application's code (e.g., a GitHub link).

Grading:

Stage 2 counts for 70% of the project grade. Your assigned TA will grade your demo and code.

More details regarding stage 3 will be posted when the stage is assigned.