University of Calgary
Department of Computer Science
CPSC 471 : Database Management Systems

Group Project Specifications

In practice, database management is rarely accomplished by a single person. For this reason, a major component of the assessment process involves a group project component.

Group

You are expected to work in a **group of three students**, two students or 4 students are acceptable options but under some conditions. Group members can be from different tutorials and are not necessary to have the **same TA**. You are highly recommended to work with students who you never met before and to mix gender in the group. This will help developing your communication skills as the market never hires you with your friends. Select your group members and upload a document to related D2L section that **includes the name, student ID, tutorial number for each group member by, 20 January (11:59 PM)**. Your TA can help you form groups if needed. Projects will be assigned to the TA's to handle them thru the semester. Each project group will be assigned to one of the Tas (the list with group members, an assigned group number and the assigned TA information will be published after your group submissions). It is not necessary to have your tutorial TA responsible from your project.

Contributions

We cannot closely mentor the contribution of every student to the project. It is important to avoid breaking the group during the semester. We had less than a handful of such cases over the past 20 years at the University of Calgary. To be fair and to give credit where it belongs, at the end of the semester, **each student will be asked to confidentially evaluate the performance of other team members**. If certain members of a group do not contribute as much as they should have, their marks will be adjusted according to their contribution to the project. You will submit one completed evaluation form for each of your group members (i.e. for a 3 member group, each member will submit 2 forms, one for each of the other 2 members). The form submission will be confidential in an individual dropbox (by **12 April (11:59 PM)**) so that your submitted forms are not visible for your group members. The form is **'CPSC471_Project Group Evaluation Form.pdf'** and includes criteria to evaluate the contribution of each member.

The Project

Each group has to develop a mobile friendly **Web-based interface** and an **API** (preferably using the ASP.NET core platform) to access a **Database**. Groups are allowed to use other platforms (i.e. Python, Django, PHP, Node.js, etc.), but they **MUST** receive approval from the TA in charge of the project. The project is comprised of four distinct components and every component is mandatory. If a group fails to complete one of the components, the group members may each be assigned a mark of 'Zero' for all components. **You have to make sure that your implemented API can prevent SQL injections.**

1. Project Proposal (Due Date: 30 January (11:59 PM))

The first component is a formal project proposal. The proposal should be for a real-life problem or scenario for which a database application would be appropriate. Possible problems include car rental, pharmacy, airlines reservation, real estate, multimedia store, medical clinic, etc. It is highly recommended to find your own real-life problem based on your own external contacts. This could involve making a real database API for someone running a business. You have to make sure that the workload of your proposed project is reasonable compared to the size of your team.

Your project proposal should include:

1. Project Title
2. Introduction.
3. Problem Definition.
4. Proposed Solution.
5. Motivation.
6. Conclusion.
7. References (Preferred).

The proposal must describe some of the functionality you are expected to implement in your system. You are not required to make specific design decisions at this point, but you should provide a general overview of your proposed system. **The detail guidelines are in 'Project Proposal Guidelines CPSC 471.pdf'.**

The proposal should be submitted via D2L.

After your proposal is accepted, the topic may not be changed without permission of the TA in charge. The TA will review your proposals and by **10 February** you will get the evaluation result, either accepted as is or your group may be asked to "scale" this the proposed project up or down if it is deemed too simple or too difficult. For the changes you will discuss with the TA on ZOOM how to adjust your project to have it acceptable.

2. Intermediate progress reports

Intermediate progress reports are to be submitted according to the following schedule. The TA assigned for your project group will provide you feedbacks after each submission, and you need to incorporate the modifications in your design according to those feedbacks before the next component submission.

1. A detailed **extended/expanded ERD** and all the related assumptions **due to 17 February (11:59 PM)**.
   The basic requirements for the system you are designing are as follows;
   you are expected to have
   a. At least eight (8) unique entity types. The subclasses are not counted as unique entity types.
   b. At least one (1) of your entity types must be weak.
   c. At least ten (10) relationship types.
   d. Entities for at least two different types of end-users such as (admins and clients).

2. The initial (logical) relational model generated from the EERD **due to 01 March (11:59 PM).**

3. The initial draft design of the functional (programming) part of the project and the Web design. Students have to demonstrate their API endpoints along with the parameters and expected returned json object. Be aware your API should not have plain queries; instead your API should deal with stored procedures. This part **due to 27 March (11:59 PM)**.

   The basic requirements of this submission are as follows :

   You need to submit one of the following 2 sets (**either set (i) OR (ii)**) -

   (i) Hierarchical Input Process Output (HIPO) diagram, Data Flow Diagram (DFD), Dummy Website, API Blueprints

   (ii) UML Diagram, Sequence Diagrams, Dummy Website, API Blueprints.

   Some instructions for them are –

   - The HIPO diagram is needed. Specific detailed algorithms for each task are not required for this, but still, you may develop them, it is up to you.
   - For DFD, not only the context diagram, but a complete DFD showing all components of the system properly describing the system design in detail is needed. Some examples are given in **'HIPO & DFD.pdf'**.
   - For the dummy website, we need the design of your site like "under construction", but do not display "under construction". Rather have all links inactive (i.e. using "#" to refer to the same site). In other words, submit a general view of your website, how it will look like as a draft. Only the dummy site is needed without any active link. Links are to be activated in the final submission. You do not need to submit a fully functional website, just the blueprint of the website is sufficient for this submission. The fully functional website that is connected to the database is required for the final project demonstration. You need to submit the screenshot of each page. The screenshots should be taken from the web pages that are generated by your code. You are not supposed to submit images that are drawn or created by Web UI design tools. The fully functional website being showed on demo will need to be built based on the dummy version shown in screenshot.
   - The API blueprint component of the project is not expected to be the full working API, but the design of the API and the relevant components. The final API will be submitted with the final project. Submitting the API specifications is enough, that is the blueprint of the API. An API specification is an API planning. In the API specifications you are expected to tell how your API will perform, how to specify requests for your API, how your API will react to requests, and how the expected outcome will look. It is a general blueprint. You do not need to include your database or any codes at this stage, only the general sketch of the API is acceptable as the full running project is expected at the project demonstration. There are multiple ways to include the API basics, but you must include information about all of your endpoints and, for each of them, the endpoint's URL, inputs, outputs, and functionality.

3. Demonstration (Last week of classes or during first week of Exams if TA's and students mutually agree)

The third component is a demonstration. The demonstration will not be a presentation so don't prepare slides. This is more of zoom session with your TA near the end of the semester where you share your screen and show all functionalities of your project live. Every group member must attend and participate in the demonstration. **YOU SHOULD HAVE A COMPLETE IMPLEMENTATION OF YOUR PROJECT AT THE DEMONSTRATION.** Your TA will be calling different API endpoints using the Postman tool installed in your computer.

Your demonstration will be assessed according to the following criteria:

- Were the scenario and system details fully described?
- Was the system well-designed and well-presented?
- Were the basic requirements (noted above) met?
- Were the API endpoints functional and covered all functioanlities?
- Is the Web interface user friendly with minimum typed-input (mostly selection and click based)?
- Were the structure and results of several well-designed stored procedures presented?
- Did every group member participate and respond to questions?

The demo evaluation weight distribution is as follows –

- The final outcome matches the initial proposal (25%)
- The web-based interface is completely working (20%)
- User friendly interface with minimum keyboard input. (10%)
- The APIs are completely working (20%)
- Correctness of the project functionality. (25%)

4. Final Report (Due Date: 12 April (11:59 PM))

The fourth component is a final report. This report should contain a comprehensive summary of the real- world scenario you selected as a topic (as though you conducted some form of use case analysis) and a complete description of the database and endpoints constructed. The final report has to be submitted via D2L. Submission should include a single PDF file and the project's source code and database.

The report should include the following sections:

- An abstract of no more than 300 words.
- An introduction where in you:
    - Describe the problem or task your database was designed to address.
- Describe (briefly) the system you have created to address the problem or task.
- A project design section where you discuss the different users of your system. Your discussion in this section should be considerably more detailed than what you described for the presentation – this section should describe a complete transaction collection and, consequently, provide a complete picture of the functionality offered by your API.

- The project design section must also include a thorough EER diagram. Every component of this diagram must be present, visible, and legible and any changes that were made since the presentation should be clearly indicated.
- An implementation section should begin with a complete relational model diagram, indicating that you followed the algorithm for converting an entity relationship diagram to a relational schema diagram. Discuss any significant or unusual decisions made during this process.
- The implementation section should then describe the DBMS you selected for the implementation of the project and must also include the SQL statements for each of the transactions implemented. It is not necessary to discuss these transactions in relational algebra or calculus.
- A **professional API documentation** with using the postman tool. Also, you have to use the built-in API documentation feature provide by **Postman** (https://www.postman.com/). The generated API documentation should show a description of each API endpoint along with type and expected parameters. Moreover, this documentation should show a sample request and response for each endpoint.
- A proper **user guide** for people other than you and your TA to be able to smoothly use your project outcome, you must show all functionalities of your Web interface (i.e. with screenshots of every option of every page with descriptions).

There is no minimum page requirement, but you must thoroughly address every section requested above. Insufficient coverage in any of the areas detailed above may warrant penalties.

Do not copy anything from other sources without a reference and credit to the source. Your group is expected to reference all materials (including the course textbook and any personal communications) used to complete the components described above. Failure to do so may be considered academic misconduct.

We will stick to this document as the protocol for the project of this course. However, in rare cases, it may become necessary to amend this specification at some point to the benefit of the students. If that is the case, you will be notified via D2L.

Marks Distribution:

The project is worth 30 marks of the course assessment. The breakdown of the weight of each component is as follows:

- Project Proposal: 02
- Extended ERD: 03
- Relational Model: 02
- Functional Design: 03
- Project Demo: 12 (fully working outcome as per the project proposal)
- Final Report: 08 (describing whole project content and a user-manual)