CSC 315 Spring 2020

Collaborative Project
Grade: 300 points

Objectives:
The primary objectives of the collaborative project are for students to:
- Gain experience with, and a clear understanding of, the database system development process.
- Apply concepts learned in this course to collaboratively design and implement a large database system for a real user.
- Strengthen problem solving and critical thinking abilities, integrating knowledge gained in this and other courses, through the design and development process.
- Develop and strengthen communication abilities, project management skills, and teamwork by working collaboratively in teams.
- Develop a deeper understanding of the significance of social issues, and the ability to analyze and address social issues using computational thinking.

Project Grade and Schedule:
The project will be completed in stages as described in this document.

Stage		Due Date	Due Time	Grade
I.	Team Formation	02/03/20	In Class	10
II.	Project Proposal Pitch and Discussions	02/06/20	In Class	40
	Project Proposal and Specifications	02/24/20	By 11:59 p.m.	
III.	Database Model	03/09/20	By 11:59 p.m.	50
	Mid-Semester Collaborative Project Report	03/09/20	By 11:59 p.m.	10
IV.	Design	03/26/20	By 11:59 p.m.	50
V.	Implementation – Tables and Queries	04/09/20	By 11:59 p.m.	50
	Implementation – User Interface	04/27/20	Before Class	
VI.	Handover – Testing and Presentation	05/04/20	In Class	50
	Final Product		Before Class	
VII.	Final Project Report and Reflections	05/04/20	By 11:59 p.m.	40
	Total			**300**

The dates given above are "planned" dates. Any changes will be updated on Canvas.

Each stage will be assigned an intermediate grade. You may modify deliverables to reflect feedback, and changes in the project specifications or design. Appropriate revisions may result in an improved final grade for that stage. Revisions should be submitted along with the deliverables for the next stage. **Grade revisions will be applied at the end of the semester.**

Each stage has a 10% penalty per day for late submissions. 10% of the grade for each stage will be based on active participation in group meetings, individual contributions to project work, and effective use of collaborative mechanisms such as `GitHub`, the wiki, and group meetings.

Documents must be submitted to Canvas in `.pdf` format. All submissions must include:
- the project and team names,
- names of all team members,
- link to the team's project repository on `GitHub`.

All project documents must be uploaded to `GitHub` project repository in the "`docs`" subfolder and displayed on the `GitHub` wiki page. All source code must be maintained in the "`code`" subfolder of the project repository.

Project Background and Requirements:
Based on discussions in class, each team will identify some aspect(s) of the CivicStory project that can be addressed through the design and implementation of an efficient database in PostgreSQL and associated queries, that can be later integrated into a web-based application. The goal is to hand over a complete database with real data, and a set of useful, complex, and efficient queries.

Each group will include students from CSC 315 Database Systems and JPW 321 Race, Gender, and the News. The JPW students will focus on the user interface and major functionality, while the CSC students will focus on the backend database and queries.

Responsibilities:
CSC 315 students will self-select into 8 groups of 4, and 2 groups of 5, who will collaboratively design and implement the system. You must add yourself to the appropriate Canvas group in the group set `CollabProject`. Each group will additionally include 2 JPW students.

Each group will be assigned to a team within an "organization" on `GitHub` with a private repository for the project. `GitHub` will be used to manage and share the project, including the source code and documentation. The project must be set up and maintained in a private repository but follow the principles of FOSS so that it can be made available publicly in the future, after the semester is over.

Each **team as a whole** is responsible for creating and managing "issues" on `GitHub` to aid in managing the project tasks and schedule. The initial milestones should list the stages for the project with deadlines as specified in this document. The team is responsible for creating subtasks as "issues" to track intermediate deadlines, tickets, resource requirements, etc., and keeping these updated.

The **team as a whole** is responsible for making one submission with the complete set of deliverables in the appropriate format and locations, as per the schedule and requirements detailed in this document.

Each team member is responsible for actively participating in all project meetings, responding to team communications, and collaborating with other team members to complete tasks on schedule. Every team member is expected to contribute substantially and will be graded accordingly.

A student who does not collaborate appropriately in any stage will receive a penalty of 10% for that stage. A student who does not actively participate or make a significant contribution for any stage will receive a grade of 0 for that stage.

Stage I – Project Team Formation
Students gain an understanding of CivicStory through class discussions, brainstorm ideas, identify the problem they will address, and form teams. Students will give their team and project module suitable names that must be approved by me. Each team member should have an account on `GitHub` and sign up at `education.github.com` for the free student developer pack that comes with private repositories.

> **Deliverables:**
> The Canvas wiki "Collaborative Projects" updated with the team name, project name, and details for each member including name, `GitHub` id, and preferred communication mode.

Stage II – Inception: Project Proposal and Specifications
Team members will research the problem domain, actively participate in meetings, brainstorm ideas, and collaboratively define an innovative module that will provide data and queries to meet the identified need. A suitable project module:
- Is large enough to require several persons' effort for it to be completed on time.

- Will implement a database with many entities and innovative queries.
- Has different types of users with different levels of access.
- Will provide a simple and easy to use web-based user interface for the module.
- Will capitalize on each team member's skills, interests, background, and experience.

Each team will pitch their project idea in class, and gain approval from me for the project idea and scope.

Each team will modify the idea based on feedback received and prepare a 'Specifications' document (2-3 pages) that describes the problem or need, and possible approaches to solve it. The document should include at least the following:

- Problem statement.
- Objective of the module.
- Description of the desired end product, and the part you will develop for this class.
- Description of the importance and need for the module, and how it addresses the problem.
- Plan for how you will research the problem domain and obtain the data needed.
- Other similar systems / approaches that exist, and how your module is different or will add to the existing system.
- Possible other applications of the system (how it could be modified and reused.
- Performance – specify how and to what extent you will address this.
- Security – specify how and to what extent you will provide security features.
- Backup and recovery – specify how and to what extent you will implement this.
- Technologies and database concepts the team will need to learn, and a plan for learning these.
- A diagrammatic representation of the system boundary that specifies what data you will model and which queries you will implement.

Each team will identify one student to be the integration lead. That student's virtual machine (VM) will be used to host the complete project source code and database with live data and be used to demonstrate the project. The integration lead will prepare their VM for database implementation.

Each team will create and maintain a private project repository on `GitHub` that will include a wiki to provide clear documentation of the design and development for the project, and milestones for the stages for the project with deadlines as specified in this document. Subtasks for Stages II and III should be identified and set up as "issues" to assign specific tasks to each team member; tasks should include research needed, installation of tools, etc. Each repository should be organized, with a "`code`" subfolder for source code, and "`docs`" subfolder for project documents. Specify the `GNU GPL` for this project.

Deliverables:

Submit to Canvas as one `.pdf` file

- Project Proposal and Specifications document, including link to `GitHub` project wiki.

Submit on `GitHub`

- The team project wiki created and updated with name and description of the project, and sections for documents that will be created later in the process.
- Project Proposal and Specifications document in the "`docs`" subfolder and displayed on the wiki.
- Open source license in the appropriate format and location.
- Project milestones and issues created and/or updated.

Stage III – Elaboration: Database Model

Each team will review the specifications with stakeholders to discuss the requirements and scope of the project. All members of the team must actively participate in this review.

Based on the discussions, the team will <u>develop a complete entity-relationship (ER) diagram</u> to model the database. If you are adding to an existing system, show how and where the additional data fits into the original schema. Indicate what changes must be made to the original schema to integrate the new data.

- Show all entities and the relationships between them. Be sure to clearly specify aggregation, composition, specialization / generalization, and multiplicities.
- Show attributes for each entity and relationship (where applicable).
- For each entity, indicate which attribute(s) form the primary key.

<u>Map the ER diagram to a relational schema</u>, i.e. show the relations that evolve out of the ER diagram. Specify the keys and relationships between the relations.

Additionally, based on your research and understanding of the social justice issue, <u>estimate</u> the following:

- Initial database size (approximate number of records)
- Types and average number of searches
- The different views. For each view list the data and transaction requirements. Give a few examples of queries, in English, to illustrate.

Use an appropriate ERD diagramming tool, preferably `Lucidchart` https://www.lucidchart.com/pages/how-to-draw-ERD to draw diagrams in the correct notation.

Create new issues on `GitHub` to assign tasks to each team member and update existing ones. Every team member must participate actively on `GitHub`.

Deliverables:
<u>Submit to Canvas as one `.pdf` file</u>

- Database Model Document.
- Project Proposal and Specifications document, revised if applicable.

<u>Submit on `GitHub`</u>

- Database Model document in the "`docs`" folder and displayed on the wiki.
- Revised Project Proposal and Specifications document, if applicable.
- Project milestones and issues created and / or updated.

Stage IV – Elaboration: Design

Each team will review the Database Model document with stakeholders, and update the model as needed.

Demonstrate that all the relations in the relational schema are normalized to BCNF.

- For each table, specify whether it is in BCNF or not, and explain why.
- For each table that is not in BCNF, show the complete process that normalizes it to BCNF.

Design a complete set of queries to satisfy the transaction requirements identified in the previous stages.

Create new issues on `GitHub` to assign tasks to each team member and update existing ones. Issues should include details about the subtasks and resource allocation.

Deliverables:
<u>Submit to Canvas as one `.pdf` file</u>

- Design document.
- Documents for previous stages, revised if applicable.

<u>Submit on `GitHub`</u>

- Design document in the "`docs`" subfolder and displayed on the wiki.

- Documents for previous stages updated if applicable.
- Project milestones and issues created and / or updated.

Stage V – Construction: Tables in PostgreSQL, Queries in SQL, User Interface, and Testing

a. Write and execute <u>SQL data definition queries</u> to create the tables and views in PostgreSQL. Ensure that all constraints are specified in the queries. Write <u>scripts / programs that may be required to obtain and format the data</u>. <u>Populate the tables with valid data</u>, with all constraints being enforced.

Write <u>SQL data manipulation queries</u> that were designed in the previous stages. The queries must be elegant and make effective use of complex query constructs such as subqueries. Execute and test these queries to ensure that they work correctly. Examine the outputs carefully to verify that the queries do not return spurious tuples.

b. <u>Implement a simple user-friendly web-based interface</u> to enable the user to interact with the database and run the queries you have written. Integrate your queries into the UI implemented by your team. Test the implementation to ensure that the results are as expected.

All source code must follow good programming guidelines, be user-friendly, implement elegant and efficient algorithms, and handle errors gracefully. Every source code file must have identification (maintenance) information, including the name(s) of person(s) primarily responsible for that code. Code must be well documented.

Create issues to assign tasks to each team member and update existing ones. Create new issues for tasks that need to be completed or can be added in a future version of the application.

Deliverables:

Submit to Canvas

- Text files with the data definition and population queries (submitted with .txt extension).
- Data files.
- Scripts used to obtain and format data and populate tables.
- Text files with the data manipulation queries (submitted with .txt extension).
- SQL script files that demonstrate successful execution of the queries.
- Documents for previous stages, revised if applicable.

Submit on the VM

- The schema created in PostgreSQL, and the tables populated with data.
- Text files with the data definition and population queries (submitted with .txt extension).
- Data files.
- Scripts used to obtain and format data and populate tables.
- Text files with the data manipulation queries (submitted with .txt extension).
- SQL script files that demonstrate successful execution of the queries.
- Well-documented UI source code files.

Submit on `GitHub`.

- Text files with the data definition and population queries (submitted with .txt extension).
- Scripts used to obtain and format data and populate tables.
- Text files with the data manipulation queries (submitted with .txt extension).
- Well-documented source code files.
- Documents for previous stages updated if applicable.
- Project milestones and issues created and / or updated.

Stage VI – Transition: Testing and Final Project Presentation
Each team will present a summary of their project and demonstrate the working of their module to the class. Each team will be allotted 15 minutes for the presentation. Every member of the team is expected to participate equally in the presentation.

The presentation must include a brief overview of the module, the final design, any interesting aspects, etc. The focus must be on demonstrating the database implementation and execution of queries, the challenges that were faced and how they were overcome, and insights gained from the project experience.

Use a presentation tool like PowerPoint to create and present the slides in a professional manner. Ensure that the presentation and demonstration are in suitable formats for the instructor's machine in Forcina Hall 403. You may use your own laptop, but you should test that it works before the class starts.

Plan the presentation and practice together as a team so you have sufficient time for an organized and detailed demonstration that highlights the features of your module and the system.

Deliverables:
<u>Submit to Canvas as one `.pdf` file.</u>
- Slides and other presentation materials.

Oral presentation and product demonstration to the class.

Stage VII – Transition: Final Project Report
As a team, you will assemble a well-organized and complete Project Report that is anonymized so it can be submitted to the stakeholders. The report organization and presentation will affect the final grade.

The final report should be submitted as a single `.pdf` file. It is comprised primarily of the documents submitted for all previous stages with the information below, in the order listed. The report should include the original documents as well as clearly marked, updated documents, where applicable. Clearly identify where changes have been made to any of the pages or documents.

Inception: Team Identification
Team name, project name, and names and contact information of team members. This page will not be submitted to the stakeholders.

Inception: Objectives and Overview
In this section, describe the project goals and aspirations – why you chose this project, what you hoped to learn, the original plan for the project, etc. This will be about one or two pages long.

Elaboration: Project Proposal and Specifications
This consists of the materials submitted for Stage II, with revisions clearly identified.

Elaboration: Design
This consists of the materials submitted for Stages III and IV, with revisions clearly identified.

Construction: Tables, Queries, and User Interface
This consists of the submissions for Stage V, with revisions clearly identified.

Transition: Maintenance
This is satisfied if the source code is appropriately documented and uploaded on `GitHub`, and the final report is appropriately labeled and organized.

Transition: Product Hand Over
This consists of successful testing, the slides used in the presentation, and a brief write-up or 'manual' with instructions on accessing / installing and using the system. Include a listing of queries that were not completed or can be added in a future version of the application. All documents should be uploaded to `GitHub` wikis.

CSC 315 Spring 2020

In addition to the team project report, each team member will individually complete a survey that will have questions related to the project experience. This will include:

1. A description of the project management structure and how the team was organized.
2. Reflections on the experience of working on a socially relevant project, particularly as part of a collaborative team.
3. Reflections on database concepts you learned through the experience of working on the project.
4. Difficulties and challenges faced, particularly with the collaboration aspect and how these were overcome.

The link for the survey will be posted on Canvas towards the end of the semester.

Deliverables:
- Complete Project Report as one `.pdf` file organized as specified above.
- Reflections submitted through the survey.
- All required documents and source code on `GitHub`.
- Project milestones and issues updated and closed where appropriate.
- The raw files (e.g. `jpg`, `doc`, `txt`, etc.) that make up the project report. The name of each file should begin with a number that indicates the order in which the file appears in the final report. For example, the title page would be named "1. Team Identification".