University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: ___3_____

Date: ___October 24th, 2024_______

Group Number: ____99______

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Alex Kwok	70386099		a.kwok0191@gmail.com
Byeori Kim	74612821		bk.byeori.kim@gmail.com
Jun Lee	21913603		Junemessi040714@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Summary of Project:

This project models the operation of food charities to enable food donation and redistribution management. The application operates within the intersection of food logistics, food donor management, food recipient tracking, and food inventory level optimization. By connecting these four pillars to our database, we enable an informative information system to track the operation success of food charities.

Milestone 4 Tasks (Due Date: Nov 29th, 2024)

Project Summary and Documentation

All Members:
Make changes according to feedback given on milestone 2 to ensure correct relations are represented and referred to for subsequent tasks.
Decide and finalize the query functionality to implement according to project specification

Task Allocation and Due Dates - members will all note what GUI is needed to make the GUI prototype smoother to create

Alex: INSERT and DELETE Operations
Tasks:
2.1.1 INSERT
Choose a relation with a foreign key for the INSERT operation.
Note the user-friendly interface to accept values for the INSERT operation.
Handle cases where the foreign key does not exist in the referenced relation (e.g., reject or insert).
Provide appropriate error messages.
2.1.3 DELETE
Choose a relation for the DELETE operation.
Implement cascade-on-delete behavior.
Note the user interface to allow users to specify which values to delete.

Jun: UPDATE and Projection Operations
Tasks:
2.1.2 UPDATE
Choose a relation that has at least two non-primary key attributes, including one with a UNIQUE constraint or foreign key.
Note the interface to display tuples available for update.

Implement functionality to update multiple non-primary key attributes based on user selection.
2.1.5 Projection
Choose a relation for the Projection operation.
Note the interface allowing users to select attributes they want to view.
Ensure that only the selected attributes appear in the result.

Byeori: Selection and Join Operations
Tasks:
2.1.4 Selection
Choose a relation for the Selection operation.
Implement a dynamic dropdown for AND/OR conditions or a string input for user queries.
Ensure proper parsing and validation of user input.
2.1.6 Join
Choose at least two relations for the Join operation.
Note the interface to allow users to provide values for qualifying conditions.
Ensure the results display the required joined attributes based on user input.

Jun & Alex: Aggregation Operations and Division
Tasks:
2.1.7 Aggregation with GROUP BY
Choose relation to implement the aggregation query.
Note the interface (button/dropdown) for user execution.
Ensure the query uses an aggregate functions
2.1.8 Aggregation with HAVING
Choose relation for the HAVING clause aggregation.
Note the interface for user execution.
Implement the query including a HAVING clause to filter results.
2.1.9 Nested Aggregation with GROUP BY
Choose relation for nested aggregation.
Note the interface for execution.
Implement the query to find aggregated values for each group.
2.1.10 Division
Choose relation to implement the division query.
Note the interface for user execution.
Implement the division logic to retrieve the desired results.

DUE DATE ON SQL PRACTICE AND KNOWLEDGE TO IMPLEMENT THESE: NOVEMBER 12th

GUI PROTOTYPE BY NOVEMBER 15th

HAVING SQL QUERIES FOR THESE OPERATIONS: NOVEMBER 19th

GUI IMPLEMENTATION BASED ON WORK DONE SO FAR NOVEMBER 23rd

IMPLEMENTATION DONE BY ALL MEMBERS BY NOV 27th

PDF FILE & FIXING UP LAST ISSUES: by November 29th (due date)

Milestone 5 Tasks (Due Date: Dec 2, 2024)

Mock Demo: Schedule the mock demo to happen two days before the actual demo date (NOVEMBER 30th). Ensure all members can attend and provide constructive feedback to one another, and improve the quality of the presentation
Final Run-Through: Conduct a final run-through of the demo one day before the actual demo (DECEMBER 1st) to ensure everyone knows their part and the flow is smooth.
Communication: Use a group chat or project management tool to keep everyone updated on progress and any changes to tasks or due dates.

POTENTIAL CHALLENGES

1. Finalizing SQL Queries

Challenge:

- Ensuring all SQL queries return accurate results as we desire

Review Queries (Alex):

- Go through each SQL query in the codebase to ensure they meet project requirements.

Testing (Jun & Byeori):

- Create test cases for each query (or run through each query, ensuring they return the expected results with various inputs such as peer review and a third point of view
- Document any edge cases and how they are handled in the SQL, and any confusion or uncertainty
- Use Postman to create and test API endpoints that interface with the SQL database.
- Documenting the expected results for each query and comparing them with actual results, ensuring accurate functionality, or queries that need to change

2. Ensuring smooth GUI, working together with SQL as needed

Challenge:

- Ensuring the GUI is intuitive and easy to use for non-technical users.
- Fix any usability issues that arise during testing.

UI Design Improvement Testing & feedback(Alex):

- Review the current GUI layout and identify areas for improvement (e.g., button labels, navigation structure), and also ensure SQL queries are well implemented as it is more important than the GUI

Implementation of Feedback (Byeori):

- Implement changes based on team feedback and ensure all elements are functional.
- Test the GUI on different devices/browsers to ensure compatibility.

3. Preparing for the Demo

Challenge:

- Ensuring all team members are prepared to present and answer questions.
- Coordinate the demo presentation effectively to cover all features.

Practice Runs (All Members):

- Schedule a series of practice demos where each member presents their part.
- Time each demo to ensure it fits within the allotted half-hour.

Q&A Preparation (All Members):

- Prepare a list of potential questions based on the project specs and previous milestones.

- Assign each member specific questions to focus on during the Q&A session.

4. Documenting Project Deliverables

Challenge:

- Keeping documentation clear and up-to-date.
- Ensuring all project aspects are covered in the documentation.

Documentation Structure (Byeori):

- Outline the main sections of the documentation (e.g., Introduction, Features, Technical Architecture, SQL Queries).
- Write an overview section summarizing the project and its objectives.

Compilation and Review (All Members):

- Each member contributes their respective parts to the documentation.
- Schedule a meeting to review the complete documentation for clarity and accuracy before final submission