CPSC 304 Project Cover Page

Milestone #: 3

Date: Oct 24 2024

Group Number: 61

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Gregory Liu	94330628	k5o7g	gregoryliu123@gmail.com
Tony Gao	23782675	o7d0t	tonygao742@gmail.com
Richard Zhou	48759534	h5x5l	richardzhou1688@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

Your deliverables should be committed to the CPSC 304 provided repository by the milestone deadline listed on Canvas in order to allow your TA to look at your milestone before you meet with them.

2. A brief (~2-3 sentences) summary of your project.

Our project is a tourist attraction application, focusing on helping users discover, review, and book experiences in major cities, starting with Vancouver. The application will provide information on local tourist locations, acting as a central hub for travellers to plan and organize their trips.

3. **Timeline and task breakdown/assignment:**

Milestone 4: Implementation

1. Setup SQL Database and Schema
    a. Create the SQL schema(tables, relationships, keys, and constraints).
    b. We have 14 entities in total(including ISAs). Each **group member will each be responsible for translating 4-5 entities into SQL Schema, and putting it into the initialization script**
    c. Approx. time (1-1 ½ weeks)
2. Setup SQL initialization script
    a. Script to initialize the database and provide some data to work with.
    b. Provides a standardized environment for every member to test against.
    c. Should be runnable
    d. Provides enough data so that queries won't be trivial
        i. Can add more data if needed throughout the project
3. Merge to a single SQL script
    a. Merge scripts to create tables and scripts to initialize data to one script

Queries:
Task Division: each group member will be responsible for writing and implementing 3-4 of these queries, both in SQL and adding it into the GUI
Approx time: (1 week)

4. INSERT Queries (Richard)
    a. Allows users to insert data into the database
    b. Occur in relation with foreign key
    c. Invalid insertions are appropriately reject
    d. Relations: Rating, Booking, User data
5. UPDATE Queries (Greg)

    a. Allow users to update non-primary keys (ex. Users editing their comments, ratings, changing their booking, etc.)
    b. At least one non-primary key attribute must have either a UNIQUE constraint or a foreign key that references another relation.
    c. Should display tuples that are available so users can select which tuple they want to update (based on the key).
    d. Relations: Rating, Booking, User data
6. DELETE Queries (Tony)
    a. Implement a cascade-on-delete situation for this relation (ex. Allow users to cancel reservations, delete ratings, etc.)
    b. Relations: Rating, Booking, User data
7. Selection Queries (Richard)
    a. Allow users to select tuples satisfying conditions (ex. Allow user to filter location based on their interest) using any number of AND/OR clauses and combinations of attributes w/ any type of equality operations (equal, not equal)
    b. Using a dynamically generated dropdown of AND/OR options.
    c. Relations: Location, Event, City
8. Projection Queries (Greg)
    a. Allow users to select specific attributes to view the data. (ex. Allow user to view only location name on default for simplicity, users can choose which other information on location they can view more of)
    b. Relations: Location
9. Join Queries (Tony)
    a. Allow users to see other users' profile, and what ratings they have done in the past
10. Aggregation with GROUP BY (Greg)
    a. Let users Find the average rating score for each user, like whats the average score they have given in all their ratings, implement this using a dropdown menu
11. Aggregation with HAVING (tony)
    a. Let users find all locations with a rating score of x or higher, implement this using a dropdown menu/with something where the user can specify what x is
12. Nested aggregation with GROUP BY (Richard)
    a. Let users see highest rating score of users with at least X amount of reviews, so they can look for people who have done a lot of reviews and see which location they rated the highest
    b. Implement this using a dropdown menu
13. Division (Greg)

    a. Allow users to search for and get all locations filtered by certain conditions, like by city and/or by categories using a dropdown menu
    b. Relations: Location, city, category

Task Division for GUI:

- We have divided the GUI tasks into 6 steps, with 6 main key features. Each of our group members will be responsible for **2 of these key features**, implementing all the steps for each feature.
- **Approx time to complete(2-2.5 weeks)**

14. GUI Prototyping and Setup
    a. Create a basic prototype of the GUI, and design user interfaces for data entry and query execution.
    b. Create a basic prototype of the GUI using Javascript(Team)
    c. **Key features needed**:
        i. login/user authentication (Greg)
        ii. searching for locations with and without filters like city or category (Tony)
        iii. Displaying info about locations (Richard)
        iv. Displaying info about users (Greg)
        v. Users make bookings (Tony)
        vi. Users make ratings (Richard)
15. Integrate SQL Queries with GUI
    a. Link GUI buttons and input fields in our features with the SQL queries and make sure that the data gets updated in our database, and make sure all our info is displayed correctly
16. Sanitization - Security Practices
17. Error Handling and User Notification
    a. For our key features, Implement error handling for user inputs, invalid actions, and displaying error/success notifications
18. Test and Refine GUI for User Friendliness
    a. Test and refine GUI to ensure all features are operational and usable
19. Comprehensive Testing of SQL Queries and Data
    a. Test each query carefully to ensure non-trivial results for all queries

Milestone 5: Demo

Approx time(3-4 days), Team will prepare as a group

1. Milestone 5 Demo Preparation
    a. Prepare the demo, ensuring that all tasks are demonstrable, with scripts and GUI interactions ready.
2. Final Documentation and Code Freeze

a. Document project functionalities, describe schema and queries, finalize the SQL script, and submit code to GitHub.

Milestone 6: Peer and self-evaluation **(individual task)**

6. In the milestone 3 assignment on Canvas, submit the URL to your group's repository

https://github.students.cs.ubc.ca/CPSC304-2024W-T1/project_h5x5l_k5o7g_o7d0t