University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 3

Date: March 11th, 2025

Group Number: 6

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Edward Kim	97114250	c1p2i	eddiekim203@gmail.com
Nazif Ishrak	46844429	j9v7k	nzfishrak60@gmail.com
Daniel Owen Santosa	90451568	z1p5e	daniel.o.santosa@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

This project is an application that enables efficient management of automotive racing teams and their engineering aspects. It does so by enabling systematic tracking of the implementation of various projects and upgrades in different vehicles, ensuring that the user will always know which team is on which project for which car. The application also allows teams to seamlessly manage these aspects across different categories of racing by being a dynamic database system.

Timeline & Task Breakdown + Assignment

Week 1: Planning, Setup, and Basic CRUD

Edward:

1. Complete and submit project Cover Page
2. Implement SQL INSERT statements with comprehensive test data
3. Implement the INSERT query feature (handle foreign keys, error cases)

Daniel:

1. Draft initial SQL script structure (DROP TABLE, CREATE TABLE statements) with primary keys, foreign keys, and constraints based on the ER diagram
2. Write code for the UPDATE query, including handling unique and foreign key constraints

Nazif:

1. Set up the basic GUI structure (layout, buttons, dropdowns, input fields).
2. Design the GUI interactions and page navigations
3. Implement DELETE operation, including cascading deletes where appropriate
4. Create user-friendly error handling and verification notifications for CRUD operations

All Team Members:

- Regularly commit incremental changes
- Peer-review code
- Proactive communication

Week 2: Complex Queries and GUI Refinement

Edward:

1. Implement Selection Query with multiple AND/OR conditions via GUI dropdowns
2. Implement Projection operation with dynamic attribute selection via GUI
3. Implement SQL Division query, with clear documentation and GUI trigger

Daniel:

1. Implement JOIN query (at least two relations) with user input
2. Implement Aggregation with GROUP BY (COUNT, AVG, MIN, MAX) and GUI interface

Nazif:

1. Implement Aggregation with HAVING clause and GUI interface
2. Implement Nested Aggregation with GROUP BY
3. GUI improvements: enhance user-friendliness, simplify input interactions, and finalize layout

All Team Members:

- Regularly commit incremental changes
- Peer-review code
- Proactive communication
- Finalize the SQL script, ensuring reusability and drop/recreate functionality
- Ensure sufficient data for complex query demonstrations

Week 3: Documentation, Finalization, and Demo Preparation

Edward:

1. Write detailed project description PDF (project goals and functionality)
2. Document differences between initial and final schema with explanations
3. Prepare and manage presentation/demo scripts for INSERT, UPDATE, and DELETE queries

Daniel:

1. Prepare README file
2. Verify all queries are listed in documentation with file/line references
3. Present SELECT, PROJECTION, and JOIN queries during the TA meeting

Nazif:

1. Organize repository with clear commit history
2. Perform final sanitization and error-handling checks
3. Present Aggregation, HAVING, Nested Aggregation, and DIVISION queries during the TA meeting

All Team Members:

- Coordinate demo rehearsals
- Ensure all members are familiar with all code and queries
- Peer and self-evaluation (Milestone 6)
- Regularly update documentation based on code changes

Ongoing Responsibilities (All Team Members):

1. Regularly commit incremental changes
2. Peer-review each other's code
3. Address issues proactively through communication
4. Maintain up-to-date documentation

Milestone 5 - Demo Preparation (All Team Members):

1. Coordinate demo rehearsals

Milestone 6 - Final Task:

1. Peer and self-evaluation