University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 3

Date: July 26, 2024

Group Number: 42

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Dominic Leo	66545906	t6b8l	dpaleo22@gmail.com
Robert Burns	27053263	j4w1e	rdburns96@gmail.com
Shane Mander	38316287	s1u2u	shane.mander31@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.

University of British Columbia, Vancouver
Department of Computer Science

Project Description

A real estate management system that would allow a real estate company to store and query its listed properties, and the parties related to said listed properties. This database will allow employees of a real estate company to store and query all property listings, both past and present, that the company has been involved in. It will enable an agent to view information about a property, which agent listed said property, the contact information of the seller (or the seller's lawyer), the building manager's contact details (if the property is an apartment), and any disclosures that may have been made regarding the property (e.g., history or asbestos, nearby construction, etc.).

General Tasks:

1. Example Data: to be done by **Robert** by **July 29**
   Come up with dummy data sets which can be used to populate tables for the purpose of testing.
2. Query Specification: to be done by **Shane**, **Robert**, and **Dominic** by **July 29**
   Come up with queries that are of the types specified in Milestone 5 (i.e, Selection, Projection, Join, etc.).
3. Test Script: to be done by **Shane** and **Dominic** by **August 3**
   Finalize a test script that demonstrates the functionality of our application.
4. Update Schema: to be done by **Dominic** by **August 7**
   If any changes were made to the originally proposed schema during implementation, this must be documented with a writeup as well as screenshots.
5. Final Description: to be done by **Robert** by **August 7**
   Write up a description of our otherwise finished project.
6. Practice for Demo: to be done by **Shane**, **Robert**, and **Dominic** by **August 8**
   Carry out a dry run of our demonstration to ensure we are prepared to display our applications functionality in a timely manner and answer any questions about the various parts of our project.

Front End Tasks:

7. Implement GUI: to be done by **Shane** by **August 2**
   This rather broad task encompasses implementing all of the GUI. This means creating all of the necessary windows, prompts, and message boxes, all with functioning buttons, as well as ensuring transitions between these are smooth. In addition to what is seen by the user, this task will also require implementing the code that takes any user instructions or data and hands it to the back end.
8. Test GUI: to be done by **Shane** by **August 2**
   Once the GUI has been implemented, we will need to rigorously test it to ensure its robustness. This includes evaluating user friendliness, carefully looking for any visual bugs, and testing that it can withstand unusual user behavior (e.g., a strange sequence of instructions). All of this must be documented in screenshots.

Back End Tasks:

9. Implement Data Structures: to be done by **Robert** by **August 1**
   This includes the implementation of all tables as well as the code that allows for their management, updating, and deletion.
10. Implement Queries: to be done by **Dominic** by **August 1**
    Implement all the different queries our application must support as well the code that, if a query is successful, returns the answer to the front end or, if a query is unsuccessful, informs the front end.
11. Test Back End: to be done by **Robert** and **Dominic** by **August 1**
    Once the data structures and queries have been implemented, we must come up with tests (in addition to the tests created during implementation) that ensure the back end is functioning as intended before we begin integrating it with the front end.
12. Implement Error Handling: to be done by **Robert** and **Dominic** by **August 2**
    Implement error handling so that the application can recover from simple errors (e.g., invalid requests, invalid sequence of commands, etc.).