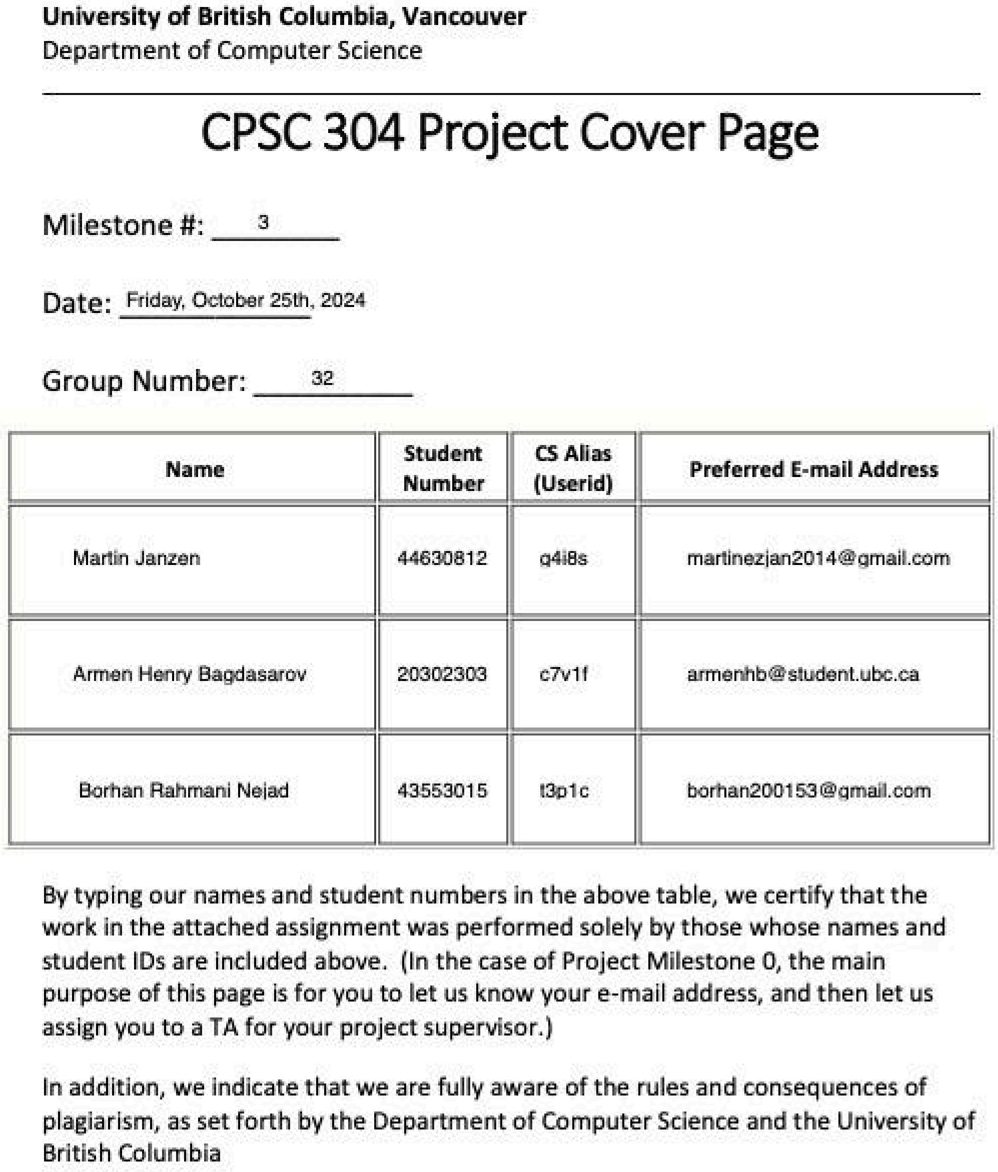
1.



University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 3

Date: Friday, October 25th, 2024

Group Number: 32

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Martin Janzen	44630812	g4i8s	martinezjan2014@gmail.com
Armen Henry Bagdasarov	20302303	c7v1f	armenhb@student.ubc.ca
Borhan Rahmani Nejad	43553015	t3p1c	borhan200153@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

2. A brief (~2-3 sentences) summary of your project.
Many of your TAs are managing multiple projects so this will help them remember details about your project.

Our project is a port management software. This system will help port managers handle the logistics of incoming ships and cargo, creating an efficient supply chain with easy retrieval, customs processes, and oversight. Our database will contain many port-related entities such as trade agreements, cargo, warehouses, and more.

3. Timeline and task breakdown/assignment.
The breakdown should be at a level of detail that demonstrates that the group has spent time meaningfully considering what there is left to do. Note that we are not asking you to predict every single possible task that you will need to do. We want to see that the group understands the scope of what is left to do and is prepared to accomplish the remaining tasks in a reasonable manner. We warmly recommend reading the milestone descriptions and associated rubrics on Canvas now so you have an idea of what we expect and what your TAs will be looking for during grading. Each task should be specifically assigned to a particular group member. Unless otherwise stated, it is assumed that all group members will work equally on the project. This does not mean that everyone needs to work on each task together. This means that the overall division of the work is equal. If this is not the case, state the work percentage breakdown for each member. This will serve as a written acknowledgement between all group members that there will be an uneven distribution of work. The member who does not do their fair share of work will have a penalty applied to their final project grade. While each member is not expected to know about every single line of code in the project, it is expected that all members can talk about the overall architecture of the project. The timeline should contain enough detail for your project mentor to determine that you understand that you need to produce a GUI for your full project. We strongly recommend reading through the description documents for milestones 4 and 5 along with the associated rubrics on Canvas so that you have a clear understanding of what is expected from you for the term project.

Entity Splits:
Each team member will be responsible for the back and front-end of their own subset of entities as follows:
Armen: Ship, Shipping Route, Port
Borhan: Shipment Container, Warehouse, Company
Martin: Foreign Country, Home Country, Country, Tariff

Should we find this distribution of work to be unequal, we will adjust accordingly. Additionally, should two entities from different group members relate to each other (e.g. Warehouse and Port), implementation can be done by both members in partnership.

WEEK 1: Oct 27 - Nov 2 : Setup and Backend - (Deadline Nov 2)

General Description: Set up the environment with necessary IntelliJ extensions and tools. Begin implementing SQL tables to represent our assigned regions of the ER diagram.

Task: Environment

Description: Set up coding environment, integrate oracle to each local environment.

All Team Members:

- Install necessary extensions and ensure database linkage

Task: Beginner Schemas

Description: Create and populate tables within the database for initial testing purposes.

Armen:

- INSERT Ship, Shipping Route, and Port tables into database with sufficient user data.

Borhan:

- INSERT Shipment Container, Warehouse, and Company tables into database with sufficient user data.

Martin:

- INSERT Foreign Country, Home Country, Country, and Tariff tables into database with sufficient user data.

WEEK 2: Nov 3 - Nov 9 : Individual Backend to Frontend Development - (Deadline Nov 9)

General Description: Implement backend tables. As we go, implement skeleton frontend functions as needed. Finalize any necessary changes as encountered.

Task: Backend Table Functions

Description: Each group member is responsible for the functional relations between their assigned entities.

Armen:

- Connect ship arrivals from certain routes to certain ports with JOIN
- Collaborate with Borhan to ensure port and warehouse are sufficiently connected.
- Implement ability to DELETE ships, ports, etc.

Martin:

- Collaborate with Borhan to ensure Tariff is sufficiently connected with Shipping Container through JOIN function
- Implement ability to DELETE countries, tariffs, etc.

Borhan:

- Collaborate with Martin to ensure Shipping Container is sufficiently connected with Tariff

- Collaborate with Armen to ensure Warehouse is sufficiently connected with Port
- Ensure proper integration of Shipping Container with Warehouse
- Implement ability to DELETE containers, warehouses, companies, etc.

Task: Skeleton Frontend Functionality
Description: For all functions created by the team, members must create skeleton frontend integration. That is, they must create buttons/input fields on a blank background that have proper functional

All Team Members:
- Implement UPDATE functionality (SQL backend integrated with skeleton frontend) for all assigned tables and relations
- Add the appropriate sample buttons to skeleton frontend for DELETION functions

WEEK 3: Nov 10- Nov 16: Individual Backend to Frontend Development - (Deadline Nov 16)

General Description: Continuation of Week 2 tasks.

Task: Implement backend query searches
Description: Create functions which will allow for the filtering of entities by attribute. Implement these filters through the frontend “skeleton”.
Armen:
- Use SELECTION, PROJECTION, GROUP BY, HAVING to filter and search for ships, ports, etc.

Martin:
- Use SELECTION, PROJECTION, GROUP BY, HAVING to filter and search for countries, tariffs, etc.

Borhan:
- Use SELECTION, PROJECTION, GROUP BY, HAVING to filter and search for warehouses, containers, etc.

Task: Finalize Initializing SQL query
Description: Ensure that we have a single script that can initialize the database and begin the app.
All Team Members:
- Concatenate all INSERTION queries into a single SQL script used to initialize the database.
- Each member adds their appropriate tables.

WEEK 4: Nov 17 - Nov 23: Frontend Focus - (Deadline Nov 23)

General Description: Finish any significant changes to backend implementation (ensure necessary INSERT, UPDATE, etc. functions are implemented in all desired places). Move to more frontend focused design, make layout user friendly and ensure no SQL has to be written by the user.

Task: Frontend Design
Description: Make a beta design of what the frontend should look like with Figma

Team Member: Armen :)

Task: Frontend Implementation
Description: Take the skeleton front end we have and make it look nice. Ensure user-friendliness such that someone with no CS knowledge can use the app. Implement user-friendly query results with notifications as to the success/failure of INSERT, UPDATE, and DELETE. Implement a way to verify these actions' effect on the DB.
Team Member: All

Task: Error handling
Description: Make sure that invalid inputs or duplicate values are appropriately handled.
Team Member: All (each member is to harden the functions that they implemented in weeks 3-4)

WEEK 5: Nov 24 - Nov 29: Finalize and Ensure desired behavior - (Deadline Nov 29)
Description: Finish any left over tasks from any of the previous weeks. Perform tests and practice DB interaction to prepare for the demo.

Task: Sanitization
Description: Implement security measures and clean code practices
All members:

- Ensure values from the user are not directly used in DB
- Research and implement how to prevent injection and rainbow attacks on assigned tables
- Implement cover cases (yet to be discussed in class) for assigned tables

Task: Prepare for demo
Description: Practice and test starting up the system, adding, removing, updating values.
All members:

- Practice together and ensure their individual schemas are behaving properly.
- Implement basic tests ensuring desired behavior.

Challenges

- We are all in different cities for reading break – need to efficiently plan how we will collaborate.
- We do not have experience with full stack applications. A learning curve is expected.
- We do not yet know the scope of how entities will interact with each other, or the functions we will need to implement as we go along.