

# **CPSC 304 Project Cover Page**

Milestone #: 3

Date: 20/10/2024

Group Number: 46

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Andrew Chen	42072777	x6b9i	<a href="mailto:iphoneotto3@gmail.com">iphoneotto3@gmail.com</a>
Helen Ma	44776268	n0j9u	<a href="mailto:helenjym@gmail.com">helenjym@gmail.com</a>
Stephanie Gao	38000618	y2m7b	<a href="mailto:stephmimi27@gmail.com">stephmimi27@gmail.com</a>

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

## Deliverables

All deliverables must be committed to the CPSC 304 provided repository by the deadline to allow TA to look at your milestone before you meet with them. You must submit:

1. A completed cover page (template on Canvas)
2. A brief (~2-3 sentences) summary of your project. Many of your TAs are managing multiple projects, so this will help them remember the details of your project.
3. **Timeline and task breakdown/assignment.**
4. The deliverables from milestones 1 and 2 have been added to the repository.
5. Each group member has made a commit to the repository. Use your UBC provided account to make the commits. The commits do not have to be code-related. For example, one group member can commit to the milestone 1 deliverables, another the milestone 2 deliverables, and the third member the milestone 3 timeline
6. In the milestone 3 assignment on Canvas, submit the URL to your group's repository.

## **Project Description (Brief)**

This project is designed to create a database system for managing **warehouse operations** and **supply chain logistics**. The system focuses on efficiently tracking inventory levels, maintaining supplier relationships, and managing the process of restocking products when stock levels fall below a certain threshold. The goal is, therefore, to ensure that products are available without overstocking, enabling warehouse managers or business owners to make informed decisions about when and how much to restock.

## **Database Specifications**

The database will allow users (like business owners or warehouse managers) to:

- Track inventory levels and determine when to restock.
- Manage relationships with suppliers, including placing and tracking orders.
- View product details and restock history.

The database will also store details about shipments, payments, and inventory availability in multiple warehouses, providing a comprehensive solution for supply chain management.

## **Description of the application platform**

We will use the department-provided **Oracle** for our database and **HTML/CSS & JavaScript** to implement the application in our technology stack.

# Timeline & Task Breakdown

## Task 0: Brainstorm, Discussion and Database Setup

- **Details:** Brainstorm the overall functionality of the project. Identify key features such as inventory tracking, supplier management, restocking orders, and how the user will interact with the database. Discuss the queries needed to achieve these functionalities. Finally, set up the Oracle database for the project.
- **Person(s) in Charge:** All team members
- **Due:** October 28
- **Deliverables:**
  - Finalized project requirements.
  - List of queries to be implemented.
  - Oracle database connection setup.

## Task 1: Database Schema Creation (Table Definitions)

- **Details:** Create all the tables in the database using SQL DDL based on the ER diagram. Define primary keys, foreign keys, constraints, and relationships between tables. Finally, compress everything into a single SQL script.
- **Person(s) in Charge:** Stephanie Gao
- **Due:** October 30
- **Deliverables:**
  - SQL script that creates all the necessary tables with constraints.
  - Table creation tested on Oracle.

## **Task 2: Populate the Database (Data Insertion)**

- **Details:** Write INSERT statements to populate each table with a reasonable number of tuples for testing and querying.
- **Person(s) in Charge:** Stephanie Gao
- **Due:** October 30
- **Deliverables:**
  - SQL script to insert data into tables.
  - Verified successful data insertion.

**NOTE:** Data are randomly generated but should mimic/reflect realistic values.

## **Task 3: Frontend Development (UI)**

- **Details:** Develop the project's front end using HTML/CSS/JavaScript. The interface should be intuitive, allowing users (warehouse managers or business owners) to interact with the system.
- **Person(s) in Charge:** Andrew Chen
- **Due:** November 7
- **Deliverables:**
  - SQL script that creates all the necessary tables with constraints.
  - Table creation tested on Oracle.

## **Task 4: Backend API Development (Node.js + Express)**

- **Details:** Implement the backend API using Node.js and Express. This includes setting up a connection to the Oracle database and defining API endpoints to handle CRUD operations like **checking inventory**, **creating restock orders**, etc.
- **Person(s) in Charge:** Helen Ma
- **Due:** November 14
- **Deliverables:**
  - RESTful API with endpoints for each core function.
  - Tested Oracle database connections and API functionality.

**NOTE:** The language used might still be subject to changes.

## **Task 5: Integration of Frontend and Backend**

- **Details:** Link the frontend and backend to ensure proper function mapping. Verify that user actions in the UI trigger appropriate API calls and database updates.
- **Person(s) in Charge:** Helen Ma, Andrew Chen
- **Due:** November 21
- **Deliverables:**
  - Fully integrated system with frontend actions properly mapped to backend API calls.

## **Task 6: Testing and Debugging**

- **Details:** Comprehensive testing of the application's features, focusing on error handling, preventing SQL injection, and correct operation of each query.
- **Person(s) in Charge:** All team members
- **Due:** November 24
- **Deliverables:**
  - Fully tested application with bug fixes and functional data validation.

## **Task 7: Documentation and Project Packaging**

- **Details:** Prepare a cover page, README, and project schema documentation, including detailed instructions on running the application. Document the Node.js and Oracle configuration, data structure, and core functionalities.
- **Person(s) in Charge:** All team members
- **Due:** November 26
- **Deliverables:**
  - Complete project documentation, including README, schema PDF, and cover page.
  - Explanation of application architecture and user functionalities.

## **Task 8: Final Submission and Demo**

- **Details:** Submit the final project and prepare for the demo. Record the demo to showcase all functionalities, including adding and updating records, querying data, and handling typical use cases.
- **Person(s) in Charge:** All team members
- **Due:** November 28
- **Deliverables:**
  - Final project submission.
  - Prepared and recorded video demos covering the entire system's functionality.

## Revised Timeline Summary

Task	Description	Person in Charge	Due Date
Task 0	Project Initialization	All Members	October 28
Task 1	Database Schema Creation	Stephanie Gao	October 30
Task 2	Data Insertion	Stephanie Gao	October 30
Task 3	Frontend Development	Andrew Chen	November 7
Task 4	Backend API (Node.js)	Helen Ma	November 14
Task 5	Frontend-Backend Integration	Helen Ma, Andrew Chen	November 21
Task 6	Testing and Debugging	All Members	November 24
Task 7	Documentation	All Members	November 26
Task 8	Final Submission & Demo	All Members	November 28