CPSC 304 Project Cover Page

Milestone #: 3

Date: March 12, 2023

Group Number: 93

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Makafui Amouzouvi	86537412	d0q3f	hortenseamouzouvi@gmail.com
Nilay Mankala	78389624	g3h2k	nilaym171002@gmail.com
Linh Pham	13150578	j8z2b	phamhlinh01@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

CPSC 304: Project Description

Milestone 3: Project Check-In

Deliverables

Your deliverables should be committed to the CPSC 304 provided repository at least two business days prior to the meeting with your TA. For a specific definition of what two business days ahead of schedule means, see the syllabus.

1. 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. You can reuse the summary from milestone 2.

The domain is food product supply management. So, it involves the logistics behind gathering and storing food products and supplying them to customers. The application is from the perspective of a supplier company managing different deliveries and orders from farmers to different customers (who could be either individuals or businesses). We're trying to address the different components of logistics behind delivering food products to customers.

2. **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.

Each task should be specifically assigned to a group member (or combination of group members). It is in your best interest to be as explicit as possible about who will work on what. In the event that there is a dispute between group members, this is one of the first things the course staff will look at when evaluating the situation. If it is clear to us what has been agreed on, it will speed up the process of conflict resolution.

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.

1. Set up databases
    a. Set up oracle database (https://www.students.cs.ubc.ca/~cs-304/resources/sql-plus-resources/sql-plus-setup.html) **Linh**
    b. Create the tables using statements from milestone 2 – **Makafui**
    c. Populate database tables using INSERT statements from milestone 2. - **Nilay**
2. Set up backend-
    a. Get a starter Laravel app running, with maybe one API route that just echos a message **Linh**
    b. Implement API routes:
        i. Insert one / multiple row(s) into the “Delivery” table - **Linh**
        ii. Delete one / multiple row(s) from the “Delivery” table - **Nilay**
        iii. Update one row in the “Delivery” table with specific values – **Makafui**
        iv. Select rows from the “Delivery” table given some condition - **Makafui**
        v. Display only selected columns for any given table – **Makafui**
        vi. Find all deliveries from the “Delivery” table that have scheduled dates within 2 days of the current day, and grouped by destination address (Aggregation with HAVING) - **Nilay**
        vii. Find the deliveries that contain all user-specified items (Division) - **Linh**

viii. Find counts of a certain item in the “Items” table, and group them by name (Aggregation with GROUP BY) - **Linh**

ix. Find the name of all customers whose average number of deliveries in a week is less or equal to the overall minimum average number of deliveries in a week (Nested Aggregation with GROUP BY)-- **Makafui**

x. Join the Delivery Table and Customer table to find the names and addresses of all customers who have scheduled delivery on the current day (Join query)--- **Nilay**

3. Set up frontend
    a. Get starter frontend page running using Vue.js framework **Linh**
    b. Set up basic skeleton layout for the frontend - **Linh**
    c. Create homepage with four navigation options (Delivery, Customer, and Warehouse, Payments page) - **Nilay**
    d. Create generic buttons/menus that allow manipulation of the table data (e.g. insertion, deletion, update)
        i. Insertion - **Linh**
        ii. Deletion **- Nilay**
        iii. Update – **Makafui**
    e. Create generic buttons/menus that allow display options (e.g. filtering, sorting, aggregating values)
        i. Filtering– **Makafui**
        ii. Sorting **- Nilay**
        iii. Aggregating with HAVING - **Nilay**
        iv. Aggregating with GROUP BY - **Linh**
        v. Nested Aggregation with GROUP BY - **Makafui**

vi. Joining tables - **Nilay**

vii. Performing a division query - **Linh**

f. Create the Delivery page, which displays the main “Delivery” table, and allows the display of the Items, Drivers, Transport Vehicles tables– **Makafui**

g. Create the Customer page, which displays the main “Customer” page, with all the generic table buttons and allows display of the Business, Individual, and Contract tables **- Nilay**

h. Create the Warehouse page with the main “Warehouse” table, with all the generic table buttons and allows display of the Farms and Farm_Warehouse_Supplies table – **Makafui**

i. Create the Payments page with the main “Bills” table with all the generic table buttons **- Nilay**

3. The deliverables from milestones 1 and 2 have been added to the repository.

Linh - milestone 1

Makafui– Milestone 2

Nilay - Milestone 3

4. Each group member has made a commit to the repository. The commits do not have to be code related. For example, one group member can commit the milestone 1 deliverables, another the milestone 2 deliverables, and the third member the milestone 3 timeline.