

CPSC 304 Project Cover Page

Milestone #: 3

Date: October 30, 2023

Group Number: 65

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Sharon Marfatia	79155529	c8c4d	ssmubc@students.cs.ubc.ca
Chris Jung	14231609	b4i0j	eunhocj@students.cs.ubc.ca
Ayan Qadir	90759622	z7e0k	aqadir01@students.cs.ubc.ca

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 Milestone 3 - Project Check In

Project Description

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 database is designed to comprehensively model the many aspects of an aquarium exhibition, capturing both its internal and external components. Our database models core entities of an aquarium which are water tanks, devices within the tanks/maintenance equipment, marine animals, aquatic plants, the supply chain of curated animal diet, and the individuals responsible for operations and maintenance. Our project aims to address and represent the intricacies of the aquarium domain, and as an example of a real-life situation, the database can be applied to efficiently manage large-scale.

Timeline: The project TA has been provided with a list of tasks that the team has left to do before the project is completed. Each task has been given a reasonable deadline.

We are planning to use: Java/CS department provided Oracle

Project Repository

We updated our repository(Link below) with deliverables we have created so far through each member's commit.

Repo link: https://github.students.cs.ubc.ca/CPSC304-2023W-T1/project_b4i0j_c8c4d_z7e0k

Project Timeline

	(internal purpose)	*LIDU: LIST, INSERT, DELETE, and UPDATE			
Milestone	Sub-Milestone	Task	Assignee	Deadline	Note
M4	M4-1	Frontend(JSwing) Setup	Chris	Nov 3th	
		Create Tables	Sharon	Nov 3th	
		Add sample data	Sharon	Nov 6th	
		Connect frontend with DB	Ayan	Nov 6th	
	Setting up for project is done and we should be ready to add features			Nov 6th	
	M4-2	LIDU functions for Inventory	Ayan	Nov 12th	Including constraints check
		LIDU functions for Item	Ayan	Nov 12th	Including constraints check
		LIDU functions for Vendor	Sharon	Nov 12th	Including constraints check
		LIDU functions for Tank	Ayan	Nov 12th	Including constraints check
		LIDU functions for Animal	Ayan	Nov 12th	Including constraints check
		LIDU functions for Plant	Sharon	Nov 12th	Including constraints check
		LIDU functions for Exhibit	Ayan	Nov 12th	Including constraints check
		LIDU functions for Staff	Sharon	Nov 12th	Including constraints check
		LIDU functions for Custodian	Sharon	Nov 12th	Including constraints check
		LIDU functions for Aquarist	Sharon	Nov 12th	Including constraints check
		LIDU functions for Veterinarian	Sharon	Nov 12th	Including constraints check
		Design and implement layout of each page	Chris	Nov 10th	Including constraints check
		Show list of data in each page	Chris	Nov 12th	
		Create a button for each basic feature	Chris	Nov 12th	
		Connect buttons with backend	Chris	Nov 12th	
	Frontend	Basic pages are completed without detailed components		Nov 12th	
	Backend	LIDU features for each entity are completed			
	M4-3	Success and fail notification feature	Chris	Nov 18th	
		Add/Delete Non-primary key attributes(Backend)	Chris	Nov 18th	
		Add/Delete Non-primary key attributes(Frontend)	Chris	Nov 18th	
		Filter by attributes(Backend)	Chris	Nov 18th	
		Filter by attributes(Frontend)	Chris	Nov 18th	
		Order Item feature	Sharon	Nov 18th	
		Clean Exhibition feature	Ayan	Nov 18th	
		Maintain tank feature	Ayan	Nov 18th	
		Go to Page feature	Sharon	Nov 18th	
		Aggregation with HAVING	Ayan	Nov 24th	Details to be decided
		Nested Aggregation with GROUP BY	Sharon	Nov 24th	Details to be decided
		Division feature	Ayan	Nov 24th	Details to be decided
		Create deliverables for Milestone4	Chris, Sharon, Ayan	Nov 24th	
	Frontend	All components of frontend are complete		Nov 25th	
	Backend	Complicated features(Aggregation, Join, and etc) are implemented			
M5	M5-1	Create list for testings to ensure all features work	Chris, Sharon, Ayan	Dec 1st	
		End-To-End tests for all features	Chris, Sharon, Ayan	Dec 1st	
		Final code update to fix any error caught during the test	Chris, Sharon, Ayan	Dec 1st	
	The implementation of the code is completed and all features work perfectly			Dec 2nd	
	M5-2	PDF of screenshots demonstrating the functionality	Chris, Sharon, Ayan	Dec 6th	
		Create Readme containing information missing in the pdf	Chris, Sharon, Ayan	Dec 6th	
	Create deliverables for Milestone5			Dec 6th	
	Ready for demo			Dec 7th	

Challenges / Things Left to Do

As seen in the timeline above, we have three features left to be specified. The details of them are below.

1. Aggregation with HAVING:
 - What user-friendly feature can be implemented by this task?
 - What specific conditions or thresholds will trigger the HAVING clause?
 - How will these conditions be dynamically constructed in the query based on user input?
 - What backup plan will be in place if no rows meet the HAVING criteria?
2. Nested Aggregation with GROUP BY:
 - What user-friendly feature can be implemented by this task?
 - Which columns/attributes are we allowing for primary and secondary groupings?
 - How will we handle situations where the nested aggregation returns no results?
 - How will the system adapt when attributes or tables change, ensuring nested groupings remain valid?
3. Division feature:
 - What user-friendly feature can be implemented by this task?
 - Which tables/relations will be the main focus for the division operation?
 - How will the user specify the numerator and denominator relations?