University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 3

Date: July 21st, 2024

Group Number: 31

Name	Student Number	CS Alias (Userid)	Preferred Email Address
Leyang Pan	93460962	yang0526	yangplypan@gmail.com
Binjie Ye	29415965	yebinjie	yebinjie2021@163.com
Cheryl Chen	44983922	cchen70	chen.q.cheryl@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 email 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

Brief Summary:

This project models the Belcarra Beachkeepers summer program, in which volunteers partake in a research study on the crabs at the park, as well as educate park visitors about conservation. Entities include volunteers, supervisors, park visitors and crabbers, their crabs and traps, and logistics like roles and shifts. This project is intended to help staff schedule shifts and manage scientific data about the crabs, as well as insights into their public interactions.

1 Timeline and Task Breakdown:

Tasks

Frontend Tasks:

- Project GitHub repository setup
- Login page, need a username (key) and password (likely phone number, will need to add that as an attribute for Volunteer), option to login as Volunteer or Supervisor
- **Volunteer:**
    - Home page (dashboard):
        - Display dynamic statistics, e.g, the number of crabs studied by this volunteer, how many hours they've worked, how many park visitors they have interacted with, the topic they teach most often about
        - Widget with their next shift (including tides, roles, etc.)
        - Small display of their current supervisor's name and contact info
    - Calendar page:
        - Actual calendar view of their full work schedule with icons on the dates when they have a shift
        - Registration system to sign up for their next shift
            - Select shift and role from two drop-down menus
    - Data upload page:
        - Upload data on crabs studied
            - Dropdown menus for: species, sex, shell condition, takeHome
            - Textbox for: crab #, size, injury, legalSize
        - Log new crabbers and their traps
            - Dropdown menus for: bait type, trap type, location, age group
            - Textbox for: trapID, success rate, phone, first name, license #
        - Log interactions with park visitors (non-crabbers or crabbers)
            - Textbox for: phone, first name, topic
            - Dropdown menus for: age group
- **Supervisor:**
    - Homepage (dashboard):
        - Display statistics (volunteers working under them, # crabs studied, total park visitors interacted with, # of crabbers worked with)
    - Calendar page:

        - Calendar view of all shift offerings and icons for each shift which, when pressed, shows a pop-up of all volunteers signed up that day and their roles
        - Release new shifts by selecting shift date on calendar and create as many roles needed
- **Public (no login necessary):**
    - About page:
        - Blurb about the Belcarra Beachkeepers program with a carousel of photos of the volunteers working (and of the crabs)
        - Buttons that link to 1) the supervisor's contact information for questions, and 2) the sign-up page for becoming a volunteer
    - Insights page:
        - A search bar at the top to display mass data (i.e, all crabs studied so far) but also with many filtering features (e.g, filter by bait type, location, find the max/min crabs studied, average size, etc.)

Backend tasks:

- Backend project setup
    - Create a script(s) that can setup tables for each entity and relation
    - Create a script(s) that can insert many tuples of data upon initialization
    - Merge the script(s) into 1 script to be run for demo and debugging purposes
    - Create a Node.js server that can run Oracle as a DBMS
- Backend query setup: (INSERT, DELETE, UPDATE, Selection, Projection, Join, Aggregation with Group By, Aggregation with Having, Nested Aggregation with Group By, Division)
- **INSERT**
    - Insert new volunteers, supervisors, and park visitors (designate them as crabber or tourist or both upon creation using a multiple choice buttons)
    - Insert new traps and link them to existing crabbers
    - Insert new crab data and link them to existing traps
    - Crabbers and trappers are inserted with info on the volunteer or crabber that recruited/owns them (total participation)
    - Insert new shifts and insert roles for each shift
    - Insert new tuples for the Interacts, Studies, and Registers relationships
- **DELETE**
    - Create corresponding DELETE statements for the above INSERT statements
    - Most foreign keys should be the default ON DELETE (i.e, DELETE fails) since even after crabbers leave and remove their traps, the study still logs them
    - However, the foreign key Crabber has that reference the Volunteer that recruited them should be ON DELETE CASCADE
- **UPDATE**
    - Volunteers and supervisors should both be able to update data on the crabs, traps, and park visitors
    - Only volunteers should be able to update their own shift schedule
    - Only supervisors should be able to update shift roles
- **Selection**
    - For all tables, include a default 'select all' for that table

---

    - Select crabs for any combination of attributes (size > 100 mm, female & Dungeness, etc.)
    - Select traps for any combination of attributes (south ^ turkey, clamshell trap ^ successRate > 0.5, etc.)
- **Projection (any combination of:)**
    - Public user is able to choose any number of attributes to view from any relation in the database
- **Join**
    - Join volunteers with park visitors to get which volunteers spoke with which visitors, so we can reward top volunteers
    - Natural join of crabs and traps to get full information about a crab (crab data, trap data, crabber data)

Aggregation...

- **... with Group By**
    - Public users can query for the max, min, and average size of crabs using a dropdown menu
- **... with Having**
    - Find illegal crab traps (only two traps allowed per license), i.e, something like SELECT COUNT(TrapID), Trap FROM Trap GROUP BY Crabber HAVING COUNT(trapID) > 2;
- **... Nested with Group By**
    - Find number of crabs with average size greater than 165 mm, i.e, COUNT() + AVG() aggregate functions
    - Find the worst volunteer (sum of shifts is the minimum of all volunteers), i.e, SUM() + MIN() aggregate functions
- **Division**
    - Divide traps to get the traps that have been able to catch all three types of crabs

- Backend endpoint
    - Create the backend endpoints in tandem with creating these queries

University of British Columbia, Vancouver
Department of Computer Science

Timeline: 07-28 ~ 08-04

Date	FE	Assigned	BE	Assigned
07-28	Setup - Node.js project - Routes for all the required pages	Leyang Binjie	Setup - Node.js server - Connect to oracle	Leyang Cheryl
07-29	- Login page - Volunteer and Supervisor dashboard basic layout	Leyang Binjie	- Script for tables - Script to insert tuples Queries: - INSERT	Leyang Cheryl
07-30	- Able to login with correct queries - Able to request data with the select queries, and display on the dashboard	Leyang Binjie	Queries: - Select - Projection - Join	Cheryl Binjie
07-31	- Calendar Page layout for volunteers and supervisor - Volunteer report page layout	Binjie Leyang	Queries: - UPDATE - Division	Cheryl
08-01 ~ 08-02	- Supervisor able to create a shift on calendar - Volunteers able to see the updated shifts, and sign up for the shift	Cheryl Binjie	- Aggregation - DELETE	Cheryl Leyang
08-02	Merge and debug	Leyang Binjie	Merge and debug	Cheryl
08-03	Project description Queries screenshots	Cheryl		
08-04	Finalization	Binjie Leyang		