University of British Columbia, Vancouver

Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: <u>Milestone 3</u>

Date: <u>July 24, 2024</u>

Group Number: <u>Project Group 3</u>

Name	**Student Number**	**CS Alias (userid)**	**Preferred E-mail Address**
Jeffrey Ho	26034066	g1k2b	jho1211@student.ubc.ca
Michelle Lei	11357167	j1r2b	michellejslei@gmail.com
Frederick Sunstrum	42266379	k0l3e	fr.sunstrum@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

University of British Columbia, Vancouver

Department of Computer Science

Timeline

Date	Task	Assigned
Wednesday, Jul 23	Create SQL scripts to create all tables and insert initial data into tables	Jeff
	Set up backend framework	Jeff
	Set up frontend framework	Jeff
	Configure connection to database	Michelle
	Create home page with Navbar and basic routing to required pages	Michelle
	Create gardens page with table to display the list of community gardens	Jeff
Friday, Jul 26	REST endpoint for INSERT ops (Donation/Receives, Plant (into plot) Handle INSERT cases where foreign key values do not exist	Michelle
Saturday, Jul 27	Create a table for Receives/Donations/GardenInfo	Michelle
Mon, Jul 29	REST endpoint for AGG w/ GROUP BY	Jeff
	Create tables to display garden plots and assigned gardeners on Community Garden page Implement forms to add/modify new plots. Implement dropdowns and buttons to assign (add) gardeners to plots.	Jeff
	Develop plants page with table showing list of plants and forms to add/remove plants	Fred
	Create user form to insert new donation into Donation/Receives table	Michelle
	REST endpoint for AGG w/ HAVING	Fred
Tuesday, Jul 30	REST endpoint for UPDATE ops (Plot, Gardener, GardenManager, Tool - availability attribute)	Michelle

University of British Columbia, Vancouver

Department of Computer Science

	REST endpoint for AGG for SELECTION + PROJECTION	Fred
	Implement filters for number of plots on Garden pages	Jeff
	Create Gardeners page with table to show information for all gardeners Implement forms to add new gardeners. Implement buttons to remove gardeners.	Jeff
Wed, Jul 31	REST endpoint for DELETE ops (delete GardenNumPlots -> triggers cascade in GardenerPlot).	Michelle
	Create a table to display tools stored at each garden. Implement buttons and text fields to add new tools. Implement buttons to toggle tool availability.	Jeff
Thu, Aug 1	Create a user interface to allow the user to execute a join query on Receives, Donations, and GardenInfo tables	Michelle
Friday, Aug 2	Create tables to display garden plots and assigned gardeners on Community Gardens pages Implement forms to add/modify new plots. Implement dropdowns and buttons to assign (add) gardeners to plots.	Jeff
	REST endpoint for DIVISION (finding plants grown in every plot/garden)	Fred
Saturday, Aug 3	Design the interface to allow selection of necessary filters on Donations page	Michelle
	Create tables to display garden plots and assigned gardeners. Implement forms to add/modify new plots.	Jeff

University of British Columbia, Vancouver

Department of Computer Science

Tasks

Fred
Jeff
Michelle
- Create SQL scripts (DONE)
    - Write SQL scripts to create all tables with necessary constraints (primary keys, foreign keys, unique constraints).
    - Write SQL scripts to insert initial data into the tables.
    - Test the scripts in the database.
- Set Up Backend Framework (DONE)
    - Set up a backend framework (Express.js with Node.js).
    - Configure connection to the database.
    - Create the basic folder structure for the backend project.
- Set Up Frontend Framework (DONE)
    - Set up a frontend framework (NextJS)
    - Configure the basic routing for the application.
    - Create the basic folder structure for the frontend project.
- Implement REST Endpoints for INSERT Operations (Friday, Jul 26)
    - Create endpoints for inserting data into tables (Donation/Receives, Gardener, Plant(into plot), Tool).
    - Handle cases where foreign key values do not exist (e.g., insert referred values or return error messages).
    - Test endpoints with Postman
- Implement REST Endpoints for DELETE Operations (Wednesday, Jul 31)
    - Create endpoints for deleting data from tables (delete GardenNumPlots -> triggers cascade in GardenerPlot).
    - Implement ON DELETE CASCADE
    - Ensure data validation and error handling.
    - Test endpoints with Postman.
- Implement REST Endpoints for UPDATE Operations (Monday, Jul 29)
    - Create endpoints for updating data in tables (Plot, Gardener, GardenManager, Tool - availability attribute).
    - Ensure data validation and error handling.
    - Present available tuples to the user for selection.
    - Test endpoints with Postman.
- Implement REST Endpoints for Selection and Projection
    - Create endpoints for selecting data with filtering conditions (GardenerPlot).
    - Allow for AND/OR clauses in filtering conditions.
    - Ensure proper query optimization and security.
    - Test endpoints with Postman

University of British Columbia, Vancouver

Department of Computer Science

- Implement REST endpoint for aggregation with group by (Fri, Jul 26)
    - Endpoint for grouping the Grows table by garden address and counting the number of plots in use for each garden address
- Implement REST endpoint for aggregation with having (Monday Jul 29)
    - Endpoint for grouping the Grows table by garden address and count the number of plots that have no gardeners assigned (to determine the number of available plots)
- Implement REST endpoint for nested aggregation with group by (Fri, Jul 26)
    - Endpoint for grouping the Grows table by garden address and summing the total quantity of plants and finding the garden with the most plants planted
- Implement REST endpoint for division (Fri, Aug 1)
    - Endpoint for finding plants that are grown in every plot/garden
- Develop Home Page (DONE)
    - Add navbar with links to list of community gardens and donations
- Develop page listing community gardens (Mon, Jul 29)
    - Create a table to display the list of community gardens.
    - Add links to relevant community garden pages.
    - Implement filters for the number of plots.
    - Add buttons and forms for CRUD operations.
- Develop Donations Page
    - Create a table for Receives/Donations/GardenInfo (Friday, Jul 26)
    - Create user form to insert new donation into Donation/Receives table (Tuesday, Jul 30)
    - Create a user interface to allow the user to execute a join query on Receives, Donations, and GardenInfo tables.
    - Ensure the user can provide at least one value to qualify in the WHERE clause.
    - Design the interface to allow selection of necessary filters. (Thursday, August 1)
- Develop Community Garden Pages (Wed, Jul 31)
    - Create a table to display tools stored at each garden.
    - Implement buttons and text fields to add new tools.
    - Implement buttons to toggle tool availability.
    - Create tables to display garden plots and assigned gardeners.
    - Implement forms to add/modify new plots.
    - Implement dropdowns and buttons to assign (add) gardeners to plots.
    - Add buttons and forms for CRUD operations.
- Develop Gardeners Page (Wed, Jul 31)
    - Create a table to show information for all gardeners.
    - Implement forms to add new gardeners.
    - Implement buttons to remove gardeners.
    - Add buttons and forms for CRUD operations.
- Develop Plants Page (Monday Jul 29)

University of British Columbia, Vancouver

Department of Computer Science

- Create a table to show the list of plants available for planting or that have already been planted.
- Implement forms to add new plants.
- Implement buttons to remove plants.
- Add buttons and forms for CRUD operations.

Description of Challenges

- Unfamiliar with how to maintain security over queries (e.g. preventing DROP TABLE being used in the query)
  - Research on Google about how to tackle these issues (e.g. StackOverflow)
  - Review the later lecture material and the tutorial 2 project
- Dealing with connection issues between remote CS server and local development environment
  - Review the tutorial 2 guide on how to setup local environment
  - Consult TA for help if they are familiar with it
- Managing the short timeline
  - Communicate with the team and discuss issues as they arise
  - Help each other with tasks or provide guidance if needed
  - Ask TA early if we have questions about how to approach a problem