

University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 3

Date: October 29, 2025

Group Number: 68

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Kaitlin Gill	28374149	g0u9b	gkaitlin@student.ubc.ca
Natalia Blanco	16174898	a5z0k	talia22@student.ubc.ca
Kieran Freitag	19405778	t9j0l	kfreitag@kieran.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

University of British Columbia, Vancouver
Department of Computer Science

PROJECT SUMMARY

Our project is a smart-garden monitoring system that allows for a garden to be broken up into sections, each of which may be allocated resources, plants, and tools as required. Each section has a monitoring system which tracks various environmental endpoints and resource usage, as well as maintains a maintenance log. Every garden has an owner and others may be granted access to it.

PROJECT TIMELINE AND TASK BREAKDOWN

Overview:

Milestone 3 Meeting: Nov 7 (9 days) - Project check-in

Milestone 4 & Milestone 6 Due: Nov 28 (30 days) - Implementation & peer evaluation

Milestone 5 Demo: Dec 5 (37 days) - Live demonstration

Phase 1: Setup & Foundation (Oct 29 - Nov 10)

Kaitlin Gill

- Design and implement basic GUI framework/layout
- Initialization SQL setup script with sample data
- Implement Insert query with foreign key handling
- Implement Update query with constraint handling

Natalia Blanco

- Set up SQL initialization script with sample data
- Design user input format
- Implement Selection query with AND/OR clauses
- Implement Projection query with attribute selection

Kieran Freitag

- Normalize schema and create comprehensive test data
- Set up error handling and sanitization framework
- Implement Join query across multiple tables

University of British Columbia, Vancouver
Department of Computer Science

- Begin Aggregation with GROUP BY query

DELIVERABLES:

- 6 queries functional in GUI
- Database Setup and Initialization
- GUI and Interaction Layer
- Github repo structured

Phase 2: Core Development (Nov 11 - Nov 24)

Kaitlin Gill

- Implement Delete query with cascade functionality
- Enhance GUI usability and user notifications
- Integration testing for data management queries

Natalia Blanco

- Complete selection features and filtering
- Implement error handling across all queries
- Create user-friendly result display formats

Kieran Freitag

- Implement Aggregation with HAVING query
- Implement Nested aggregation with GROUP BY query
- Implement Division query

All:

- User experience improvements and bug fixes

DELIVERABLES:

- 10 queries fully functional
- User-friendly GUI and Interaction Layer
- Error Handling + Sanitization Completion

Phase 3: Integration & Final Implementation (Nov 25 - Nov 28)

All Members

University of British Columbia, Vancouver

Department of Computer Science

- Final integration testing and bug resolution
- Complete SQL script for drop/create/reload functionality
- Finalize PDF documentation with query descriptions and locations

DELIVERABLES:

- Milestone 4, Milestone 6 peer evaluations

Phase 4: Demo Preparation (Nov 29 - Dec 5)

All Members

- Demo script preparation and rehearsal
 - Database recreation, demo queries
- Final testing and troubleshooting

DELIVERABLES:

- Group Demonstration - Live demo flow
-

Specific Query Assignments

Kaitlin Gill

1. Insert: Garden/Section creation with location foreign key validation
2. Update: Plant status and maintenance information updates
3. Delete: Tool removal

Natalia Blanco

4. Selection: Search plants by type, status, location with AND/OR filters
5. Projection: Display customizable garden/section information views
6. Join: Find plant care requirements by joining Plant and PlantType tables

Kieran Freitag

7. Aggregation with GROUP BY: Resource usage summary by garden section
8. Aggregation with HAVING: Sections with above-average water consumption
9. Nested aggregation with GROUP BY: Gardens with highest plant diversity
10. Division: Find sections that have grown all available plant types

Shared Responsibilities

University of British Columbia, Vancouver

Department of Computer Science

- All members: Contribute equally to GUI design and user experience
- All members: Write comprehensive test data for assigned features
- All members: Document code and queries
- All members: Participate in integration testing and bug fixing

Challenges/Potential Pitfalls

- Buffer time built into each phase for unexpected challenges:

We've included extra time in each phase to handle unexpected issues that require additional work. We'll update each other weekly before or after class to stay accountable and keep the timeline on track. This also provides us with extra time to consult with the TA if we encounter bugs.

- Cross-training during the integration phase so all members understand the full system:

To ensure everyone understands the system, we will focus on writing clean code, documenting processes, and reviewing each other's commits. In preparation for the demo and during team meetings, we'll walk through our individual contributions so all members understand the database structure and queries. Based on team feedback, we'll update the code and query logic as needed.

- Check-ins during the final weeks to maintain timeline adherence:

We'll decide check-in dates before due deliverables and during the final weeks. This will allow us to share progress, identify issues in the code, and adjust task assignments as needed.

- Backup plans for complex queries if initial approaches fail:

For complex queries, we'll prepare alternative strategies in advance. If issues arise, we'll communicate with each other early-on, troubleshoot together, and make sure all base cases are covered through testing. All query changes and key decisions will be documented and shared with the team.