CPSC 304 Project Cover Page

Project Name: Seed Gem - A gardening management tool

Milestone #: __ 3_____

Date: ___24th_Oct 2024_________

Group Number: ____14_____

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Shu (Charlie) Chen	61628137	i8r6v	schen622@student.ubc.ca
David Tianyi Yin	49385537	k1l2s	tianyiy.ubc@gmail.com
Lewis Li	39058169	b9d8f	weitianl@student.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.

Project description

Seed Germ is a comprehensive system for managing and tracking plants within a garden, focusing on plant growth data analytics. The system accommodates various plant types and stages, tracks scheduled and ad-hoc events and provides traceability from seed purchase through to harvest and distribution. Its functionality includes maintaining records of plant event such as watering, weeding, etc., observations such as bud breaking, fruiting, etc. and generating timelines and analytics for efficient garden management.

Timeline and Task Distribution

Date	Goal	Responsible
Oct. 25, 2024	Be familiar with Oracle Database: Explore Oracle database tools, basic commands, and data management features to understand the environment and prepare for database integration.	Charlie, Lewis, David
Nov.1, 2024	Frontend skeleton design: Each User story translated to one index.html, establishing the basic layout, structure, and design elements. This phase focuses solely on the visual representation of the UI, without implementing any functionality. Be familiar with the backend: Understand and review key backend files: server.js, AppController.js, and appService.js to prepare for integration.	Frontend: Lewis See below for examples of front-end elements. Backend: Charlie & David Everyone should be familiar with the structure of both the front-end and the back end
Nov.8, 2024	Frontend-Backend Connection: Implement the necessary connections between the frontend and backend	Frontend: Lewis Backend: Charlie & David

	components, enabling data flow and basic interactions.	
Nov. 15, 2024	Queries Implemented: Develop and test database queries to ensure proper data retrieval, insertion, updates, and deletion.	Evenly split among Everyone
Nov.22, 2024	Backend Functions Implemented: Add core business logic and backend functions required to support the application's user stories. This includes processing requests, handling data, and managing errors.	Charlie and David will be responsible for CRUD functions. See below for some examples of the CRUD functions.
Nov.29, 2024	Front-End Styling: Apply styles to the frontend to improve the visual aesthetics and user experience.	Lewis

Views (Each view is one html file):

- All Plants
- All Suppliers
- All Orders
- Event logs
- Inventory overview
- Batch summary: current stage and past events.

User Forms (maybe stand-alone page or part of a view):

- Garden visit log
- Order form
- New Supplier form
- New location form
- new plant form
- New user form
- new batch form
- Other forms that are used to update or add information.

Backend Functions:

- addOrder: add rows of seed, supplier, quantity and price. (SQL: add)
- deletePlant: delete a plant (SQL: delete)
- tag filtering: find a plant that has ALL of the following tags: cold resistant, drought resistant, high yield. (SQL: selection)
- popular plant: find a plant that is grown in all years (SQL: division, join)
- expectedYield: update expect yield of a plant when new yield data is entered (SQL: Group with average)
- highestYield: find a cultivar that has the highest yield among all cultivars. (SQL: Nested aggregation with GROUP BY)
- survival_rate: find the cultivar with the highest survival rate with yield > 100g. (SQL: groupby with having)
- Other functions that emerge during the development.

Potential Challenges and Contingency Plan:

Frontend Skeleton Design & User Stories

Translating user stories directly into static HTML files might be straightforward for some user interfaces, but more complex interactions might be harder to capture.

Backend Functions Implementation

Developing backend logic can lead to unexpected issues related to handling complex business rules, edge cases, and error handling. We give a one-week buffer for unexpected emergencies.

Unexpected changes:

Despite our best predictions, unexpected changes are bound to happen. We planned some queries with the aim of satisfying course requirements. However, there may be better ways to meet them as the development progresses. In that case, some backend functionality will be updated.

Compressed Timeline

The workload varies across weeks. It could become intense as tasks overlap or require integration efforts. To manage this variability, we will adopt a flexible scheduling approach. When team members complete their tasks early, they will be encouraged to move on to the next task and leave more time to handle unexpected challenges in the future.