Table 1: Revision History

Date	**Developer(s)**	**Change**
2022-09-25	Juan Moncada, Aaron Billones, Steven Ramundi, Gillian Ford	Initial release

The Nursery Project
Development Plan

Aaron Billones
Gillian Ford
Juan Moncada
Steven Ramundi

September 26, 2022

1 Stakeholders

Sheridan Nurseries Norval Manager

2 Project Description

The current method of preparing pots and trays to be filled with soil and populated with seeds at Sheridan Nurseries is a process with little to no automation, requiring many manual labour hours. Recently, the owners of these farms have been finding it increasingly more difficult to fill these roles with enough workers to run the operation smoothly, and meet production demands. We aim to aid in this process by designing and implementing a machine that is able to fill trays with pots and prepare them for populating with soil and seeds. This will alleviate the reliance on manual labour and improve the overall efficiency of the farm.

3 Team Meeting Plan

4 Team Communication Plan

5 Team Member Roles

6 Workflow Plan

All team members will use the designated public repository on GitHub for inquiries relating to the project. The entirety of the projects' documentation, task distribution, software/hardware design can be found in The Nursery Project.

6.1 Git Branches

The repository presented above consists of three branches: main, develop, and prev-milestone. The workflow within these branches will ensure organized and secure development.

- *main*: This branch is the default branch in the repository. It will contain all working code, all parts/assemblies, and complete documentation. It should always be behind or up-to-date with the *develop* branch.
- *branch*: This branch is where all the development will take place. The team will review changes and perform testing in this branch. A pull request will be submitted and approved by a minimum of two team members prior to being merged into the *main* branch where the working code should be.
- *prev-milestone*: This branch will contain all the code, parts, and documentation of the project's most recently completed milestone. It will pose as a backup branch that can be referred to at any time for ease of communication and development throughout the team.

Additional branches may be added when a new feature is being implemented if a team member believes that it will help the workflow of the project.

6.2 Git Issues

A Git Project Board will be used to organize issues and tasks for each team member. The issues may be moved within the project board to the appropriate sections (ToDo, In Progress, Done). Issues will be created upon task assignment and closed when the commit has been merged into the main branch. With the creation of each issue, the appropriate assignee(s), label, project, and milestone is chosen to provide the team with useful information about the issue.

The assignees are the team members that are responsible for completing the respective task.

The following labels will be the most prominent in the project workflow to accurately describe and classify each issue:

- Bug – Something isn't working
- Documentation – Additions or improvements to project documentation
- Enhancement – Additions or improvements of hardware/software design features
- Help wanted – Extra attention is needed from another team member
- Question – Further information is requested

The Git project will be selected for the issue to appear on the project board.

The milestone will be selected to improve organization and timeline awareness.

7 Proof of Concept Demonstration Plan

The main risk of this project is the size and robustness that the project will have to have. The project deals with the manupulation of tangible objects and thus must be scaled to match this. The risk of this project can be minimized by splitting the project to four main sections (conveyor, tray allocator, dropper, and verification) which will should be able to work independantly with standardized inputs and outputs, this way the final project will be the implementations of four systems working together. Proof of concept demonstation will consist of a Cad model of the system as a whole aswell as a physical solution and proof for the main hurdle of each indivdual section.

8 Technology

- Specific programming language
- Specific linter tool (if appropriate)
- Specific unit testing framework
- Investigation of code coverage measuring tools
- Specific plans for Continuous Integration (CI), or an explanation that CI is not being done
- Specific performance measuring tools (like Valgrind), if appropriate
- Libraries you will likely be using?
- Tools you will likely be using?

9 Coding Standard

The project will follow the object-oriented programming paradigm. This will ensure flexibility and control over the complexity of the program while maintaining a level of abstraction.

Another standard that will be followed is the four-eyes principle. The four-eyes principle will improve team awareness and reduce bugs. This will be achieved in the GitHub repository by working mostly in the *develop* branch and having a team member approve a pull request to be merged into *main*.

For naming conventions, Pascal Case will be used for naming classes/structures, while camelCase will be used for naming methods/functions.

10 Project Scheduling

The project will primarily follow the course deliverables found in the Google Calendar. The progress of the project will be tracked on through the relevant milestones using the GitHub Project Board. Ordering parts will also be kept track of to ensure that the project timelines are met.

Task responsibilities will be discussed and organized during weekly meetings to keep up with the deliverables. An expected timeframe will be strongly encouraged to adhere to for the tasks assigned to each team member. The project will ideally be split equally among all members of the team for accomplishing each deliverable. Throughout the lifecycle of the project, members may be subject to a slightly heavier workload than others at specific periods in time, based on the strengths of each member depending on the hardware and software deliverable needs.