CS 321 - 02
Dr. Stacey Lukins
12.03.2020
Group 7
Colton Cappa
Michaela Dent
Amber Lai Hipp
Bre McNichols
Meghan Scott

Rooted Reminders Project Analysis

Ultimately, each requirement specified in the Rooted Reminders Requirements Document was satisfied. One change that was made between the requirements and implementation was the removal of the clock actor. Use Case 5 was satisfied in implementation using the current date as opposed to a timer. Therefore, requirement 10 was not met exactly as planned, but the functionality remains; instead of sending reminders at a given time on the clock, the application checks the date last watered and compares it to today's date, sending a reminder if the specified watering period has been reached or passed.

The project did change somewhat somewhat between the design and implementation phases. Most of our original class structure remained unchanged, but the biggest changes were made with respect to the project's GUI. We originally planned to use one class to implement all GUI components, but upon implementation we found that we needed to have several separate ones. These new GUI classes became PForm, PlantTest, AddPlant, and RemovePlant.

We also had to add and remove certain functions in many of our original classes. In the PlantList class, we removed the sortList function, as it was deemed unnecessary, and added the functions filterList and getRandom to aid in quiz functionality. Functions iterateList and saveList were also added to help with overall PlantList functionality.

The Quiz class was changed significantly from the original design; the original functions and members were removed and instead, a PlantList member was added to store the list of Plants that the quiz could choose from, and the function processAnswers was created to process the user's quiz answers and recommend a plant based upon those answers.

The team met once a week in a scheduled Zoom meeting throughout the semester; we also held supplementary meetings to work on assignments together. The group also used GroupMe to communicate between meetings and Github to upload files important to the project.

Each member of the group contributed to most parts of the project. The group came up with the idea for the project together on Zoom, and Amber wrote the proposal while Meghan and Bre created the schedule. In the requirements document, Michaela wrote use case 1, Amber use case 2, Meghan use cases 3 and 4, and Bre use case 5. Colton created the use case diagram and wrote the requirements summary with help from Amber.

Each member also wrote one class for the UML diagram and its relationship to the other classes. Amber and Bre put these pieces together and made a few changes to the overall diagram. Lastly, for the final implementation, Colton worked on the GUI components, Bre contributed to the PlantList, Plant, and Utility classes as well as Javadoc comments, and Michaela contributed to the Quiz class and Javadoc comments. Meghan worked on implementing reminders and watering dates in Utility and Plant, and Amber wrote the Plant and Driver classes and contributed to PlantList, Utility, Quiz, GUI components, as well as the reminders.