Logan Gravitt, Jayden Mai, Ryan Miller, Brayan Quevedo Ramos

Qichao Dong

CS 2400

28 April 2024

**Socially Responsible Computing: Crafting a Spreadsheet for Crops at Lopez Urban Farm**

Computer science is one of the most rapidly growing fields on the planet. As such, it is also vital that computer science is used ethically and responsibly. This project aims to utilize computer science to improve the local community, specifically by improving the efficiency of the Lopez Urban Farm in Pomona, California. The Lopez Urban Farm is a nonprofit agricultural center in which volunteers grow a variety of crops for distribution to those who need the support, such as less fortunate locals and homeless shelters. At the farm, anyone is welcome to give what they can and take what they need, thus supporting the local people of the community and combating food insecurity. The farm is a fantastic and necessary location for the community, so this project aims to use computer science to improve the farm further by increasing efficiency through the use of a user-friendly spreadsheet that can keep track of exactly how many crops are grown and given. This data will help the farm focus on the specific crops that are in the highest demand, improve sustainability by having exact data on how much food a farm of this size can produce, and may even be useful for securing more grants from the California Department of Agriculture.

The project outlined in this paper uses the Google Sheets database to hold information about the farm name and location. We are able to save crop information for each site, which includes, the crop ID (unique to each type of crop), crop name, quantity available, crop changes (no change, created, modified, or deleted), harvest date, and whether the crop is in season. This data is linked to particular sheets, identified by their ID and sheet name, to which specific volunteers may have access to certain sheets. Our project makes it less difficult for farm volunteers to add new crops, change the details of current crops, and remove crops from the database without the added stress of having to manually input all the numbers themselves.

In this project, we implemented two main data structures, Queues and Lists. Queues are a data structure in which new elements are added to the end and removed from the front. Additionally, we used dynamic queues, so it can scale indefinitely without having to specify a rigid length. This data structure was used to stage changes from the user to push onto the Google Sheets database. In order to preserve data integrity, we have decided to employ this data structure to make sure that user modifications are handled in the order that they were received. We also decided to employ queues because of their role in storing api calls. This has to do with how, when using a queue, we can use one api call but still be able to hold multiple elements.

Throughout our project, we also used Lists to hold the contents of a particular sheet within the Google Sheets database. Lists are used to hold collections of data arranged linearly in memory allowing for random access of any item in the list. We choose to use a list since they can store a variety of data kinds, including strings (ex: crop name), numbers (ex: as crop ID), and booleans (ex: whether the crop is in season). Also, since we cannot predict which crops will be altered and when, using a list with random access is advantageous as well because it makes handling necessary modifications easier.

This project stores all the collected and inputted data in a spreadsheet that can be accessed by a variety of people. This method of data storage provides a variety of benefits to the managers of the farm. Firstly, having exact numbers for each crop provides more accurate and valuable data than simple estimations written down physically. This data will always be accessible and legible, regardless of physical conditions or location, and the exact time of each data input can be easily found. However, the most beneficial way to make use of this data is likely to find trends. Knowing when people tend to input data can provide a more clear schedule for when volunteers are active, and knowing how many crops get used and how many are simply left alone can help the farm improve its efficiency. In addition, this data can be used to show the crops that tend to be in demand at certain times of year, and which ones tend to be avoided, thus allowing volunteers to know what they need more of and what they can afford to have less of. One could even compare and contrast the farm’s output and utilized space with other, similar urban farms to optimize efficiency state-wide.

By using google sheets, this project becomes easily scalable. If the farm wants to expand, each cell can easily store the number of crops being produced, and if new crops are added, it will be similarly effortless to include them in the spreadsheet. Even new locations can be easily added and analyzed with very minor changes. For the individual volunteers, implementing this system will result in a few changes. Firstly, the data will have to be regularly gathered and inputted, and while the data is extremely valuable, this will take some time away from other forms of farm maintenance. In addition, some form of internet access will also be necessary, both to read the spreadsheet and to input the information. While this can be easily done with a smartphone, arrangements may need to be made for volunteers without access to their own phone or data. The simplest way to fix this issue would be to either get a laptop for communal use, or for volunteers with smartphones to be prioritized for data collection.

Ultimately, this project served not only as a wonderful method to aid the community, but also as a way to improve the team’s computer science skills. One of the largest issues that arose from this project is following the strict API protocols and requirements that Google Services enforces at runtime. Because security and data privacy are of utmost importance, pushing changes to Google Sheets had to be in a specific format as required by APIv4. Furthermore, the planning of this project was also deemed difficult as for many of the group members, it was their first time being exposed to the practice of real software engineering through the use of industry-specific tools and services. More specifically, everyone had to write unit tests (using JUnit5) for their own code, use authentication tokens to access the database, use a Gradle build engine, and have extremely clean implementation/documentation of the code.

While data collection is certainly vital and useful to the farm, this specific solution is not the only one. Instead of utilizing the provided code, the farm could simply have a google or excel spreadsheet available to all volunteers. This would essentially accomplish the same thing but would be much less user-friendly. In addition, doing so would make it harder to input the data, thus slowing down the data collection process and reducing efficiency. If this project were to expand, our code could be used as the basis for an API, allowing for a plethora of advanced features to benefit the community even more. For example, a mobile application can be built using our API to create a marketplace where users can use farm materials to make food, jams, and more! Furthermore, the API can be integrated into lopezurbanfarm.com, allowing the community to view what’s available for pickup, saving the community valuable commute time if a specific crop is out-of-stock.