Dur Snails of Cal Poly Pomona - Timothy Jo, Olive Stam, Diego Meija, Casey Wong, Matthew Yeung

Professor Dong

CS2400

30 April 2024

SRC Technical Paper

The Socially Responsible Computing project aims to help solve problems in a local farm, the Lopez Urban Farm, located near Cal Poly Pomona using students' knowledge of computer science. In order to figure out what to do for the project, our team visited the Lopez Urban Farm and talked to the workers. We discovered that the way of logging donations and visitors was outdated as they used an old school way of pen and paper to write down their donations. Although this was not directly correlated to farming practices, by improving their donation logs it would greatly improve the quality of recording donations and help the government see their needs, which in turn helps the farm overall.

Our project is a terminal based frontend Java application. It asks the user for a filename and multiple options such as create, edit, remove, and display entries, or get a .csv file, or quit the client. In this file, the user can create or remove the different entries, and these entries consist of donations, number of visitors, and date. By being able to create and remove entries at ease without having to go through the hassle of erasing and writing new columns and entries each time, as you would on a pen and paper log, the application proves to be more useful as it makes logging simpler and requires less work for the user. Also as these entries add up in the file, it will be similar to the pen and paper log the farm had but more neat and organized. This is because when searching for certain dates or time frames, the user can use the display function which will display all the entries with that certain date or range between two dates, making it easier to find and overall more organized. Another option the user has is creating a .csv file. This is useful as well because a comma separated file can provide all the data in a neat and organized fashion so the user can easily read all the necessary entries. Through these user options, this application is able to act like a pen and paper donation log but more useful and organized, with certain features like the display function that only this application can do.

The two data structures used in this application were the bag and list. A bag was used in order to store all donations because a donation could range from clothes to food etc. Since the donations could be anything and there was no need for order as well as there being a possibility of being duplicates, we decided on using a bag to store all donations for each entry. We implemented the bag using a resizable array bag and used it in the Entry class as one of the instance variables. The other data structure used was the list. In order to store our entries, we used a list since there would be multiple entries and the entries could be manipulated easier than data structures like bags. Since we would be modifying these entries, like adding and removing, we needed a data structure that could be manipulated and located using an index. By using a linked list implementation, the list is used in the EntryList class to store all the entries that are added, removed, or edited. By utilizing these two data structures, the bag and the list, we were able to organize our entries and donations more efficiently.

This project can be scalable as when the problem size gets bigger, the results stay the same. As more entries are added, there is no difference as there is no limit to how big the donation list is since it is just a file. In addition, when using the data structures, the worst case of most functions is O(n) time complexity which is mediocre but this shows that even with the increase in entries, the time complexity will stay rather consistent although it may be a little slower as the problem size gets bigger. If the number of users accessing this application increases, then the functionality will also remain the same. Although this project is meant for one user to log all the donations, it is still possible for multiple users to access it as it depends on the file they are using to log their data. Even if multiple users are using the same file, it would not change as they would just add or remove entries at their own pace. This application can be scalable because the features of it do not change based on problem size or number of users, although as the numbers of entries get larger, the application may get a little slower due to the somewhat inefficient time complexity of the functions.

If we were to do the project over again and improve it, we would implement three main things, using a different data structure for the entry list, using a better spreadsheet such as excel or google spreadsheets, and implementing a method for searching for entries based on donations. For our EntryList class, a better data structure could be used to navigate the different entries, such as a dictionary or using hashing. If we used a dictionary with keys and values, then we could make a key an entry number and the value as an array or list data structure of the elements of date, number of visitors, and donations. This could organize our entries a little better. This would make retrieval a little faster with the time complexity of O(log(n)). The user would display all the entries and if they wanted to edit an entry they would see the entry number key and then retrieve the entry in quicker time compared to a list. Similar to using a dictionary, we could use hashing as well. By using hashing, we could make all retrieval O(1), making retrieval even faster and more efficient. Also, using the search key to be an entry number and then getting the values of date, number of visitors, and donations as a list or array, we can make the EntryList class more organized. Another change we would make if we started over was changing the file to an actual spreadsheet. In our application now, we use a .txt and .csv file. Although these files do present the data and donations accurately, we could implement a way to transport this data to an excel or google spreadsheet. This would be a good change because it is easier to access these types of spreadsheets and look at all the data in one go, separated in boxes instead of commas. The issue with the .csv file is that the comma separated items are not labeled which could cause confusion to new users but having the box separated in columns would change that. Finally, the last change we would add is an option to search entries based on donations. For example, if the user wanted to see all the donations for clothes, then the user could search for all the clothes and see which entries had clothes. This could elevate the search option in our application a little further. In addition to all these changes, although the project specifications did not require this, adding an alternative frontend user interface such as a website or mobile application would be a beneficial change. This is because having a terminal frontend can be confusing to those who are not comfortable with computers and terminals. Even though our current application can execute our needs without error, implementing these new changes could improve our application even further.

Although our application does not directly contribute to sustainable farming practices, programmers can contribute to farming practices by creating algorithms and applications to optimize and automate certain practices. For example in the sample ideas on the project instructions, there was crop rotation optimization. This can be automated by programmers to make the farmer's job a little easier without having to plan, only simply having to input the crops. By using these types of algorithms and applications, the farmer will have an easier time with managing the farm.

Computing brings a big impact on sectors like farming. Although farming may seem like a sector that has no use for computers, these applications made by programmers can be very useful as they can help manage farms in an efficient manner. As mentioned before with the crop rotation optimization, simple but time consuming things can be automated for the benefit of the users. Other sectors can also be helped just like the farms with computing being able to optimize and manage small time consuming things which allows the users to focus their attention on other important things. Even if the sector cannot be directly helped with computing, it can be helped as with examples like our current application. Our application does not help the farm directly but helps manage the farm with a donation log. By optimizing the things around the farm, it gives way to the workers to focus on the farm, rather than the donations it gets. Through computing, sectors that may seemingly have no use for computers, can be optimized and managed helping the workers focus their time on other more important things.