Talen Fisher

Professor Michelle Ruse

Data Structures (CIS 152)

16 July 2017

Project Report

My project is a mailing list. It allows a user to enter a number of different email addresses and tie them to a person object, which holds a first and last name, as well as how many clicks they have, that is how many times they've opened an email. In my project the clicks is pseudo-randomly incremented each time an email is sent out. To add an email, a user would enter the letter a, and then enter information accordingly (email, firstname, lastname). To remove one, a user would enter the letter r and then enter the email to remove. To show the complete list, a user would enter the letter s. To send an email to the entire list, a user would enter the letter q and then enter the message. Finally, to exit the program the user would press e.

On 7/4 I created the prototype version that only allowed the storage of email addresses and correlating person's information. On 7/5 I submitted the first retro online. On 7/9 I added the ability to send emails out. On 7/11 I added a check to make sure emails entered are in the format something@something. I spent the rest of the time until 7/16 testing and making sure everything worked correctly and not giving any errors.

I initially had a lot more features planned, like viewing the mailing list in different orders, sending out actual emails, etc. However I realized that these features were not necessary to the scope of the project, and I did not include them in the end. Also, sending out actual emails would require an SMTP server, which I have but people running my code may not. Adding such a feature would require more testing and more time, which I have only so much of. Time was the only blocker for me. A good initial version of a project also does not need these additional features, just that it does what it is supposed to and demonstrates use of the required elements (a map and a queue in my case).

The code for my project can be downloaded from [github here.](https://github.com/TalenFisher/MailingList) I have also included a zip of the java files.

Professor Ruse's principles of good programming include modularity, efficiency, robustness, usability, readability, and elegance. Modularity is the separation of different components of programs. Efficiency is efficient use of memory and time. Robustness is (mostly) error free code, and handling errors correctly. Usability is the ease in which a program can be used by end users. Readability is where code should be made clutter free and readable. Elegancy is where code is elegant in terms of readability and design.

My project in summary is a mailing list. It allows users to enter emails and information about a person into a list and send emails to them. It uses a Map to map emails to a Person object, which contains information about the person (firstname, lastname, clicks). For more specifics, see the first paragraph. The real world problem that this project solves is keeping track of people who want to sign up for newsletters and updates by email.

Future versions of the project could include sending out real emails, and keeping track of how many clicked on a link in the actual email. It could also include a graphical user interface. More ideas include showing the entire list in different orders, and removing inactive subscribers. Instead of a desktop application this could be managed via the web, so that you could add people from anywhere, and allow users to sign up online as well.