Contact Book
(Milestone)

John Maalouf
Marist College

The piece of software I have chosen to develop over the past few weeks is a contact book. This is a standard form contact book that takes in data from the user and creates a contact. The unique aspect of this project is the ability to import and export single contacts or entity contact books. Contacts can be fed into the program that was generated from the program previously or can take information from the user from a text file.

I. Introduction

The motivation for this program was to create an application that would allow users to create and share large amounts of contacts with one another quickly. This product will allow communities to stay more connected as everyone would have each other contact information.

The program allows the users to enter in information to contact pertaining to data fields. Including but not limited to the first and last names, their phone number, address information, and job specifications. The program's import and export feature is used by writing and reading specific text files.

When a user wants to export a contact it is a straightforward process; the first and only requirement is that they have must have one contact already in the contact book. This contact could have been created in that instance of the program or could have been imported. All the user has to do is select the export contact[s] option then type the name of the file they would like to save it to. To import contacts, the process is even more straightforward. All the user has to do it select the import contact[s] button and enter the name of the text file they would like to import the contacts from.

II. Detailed System Description

The Interface(Driver) class would contain the main method and declare an ArrayList of 3 objects (Number, Address, Occupation). It would also include a method called menu housing a switch case for the menu and method call options.

The Number Class would be the first part of the contact creation and would get information about the user's firstName, lastName, and phone number. The first and last name would be String variables, and the phone number would be a long value. This method would contain getters, setters and a toString method.

The Address class would be the second part of the contact creation and would get the

users country, state, city, street, zip, and house number. The country state city and street would all be string, and the zip and house numbers would be integers.

The Occupation class would be the third part of the contact creation and would get information about the user's job, salaries, and hours per week. The job variable would be a string and the pay and hours would be ints. This method would contain getters, setters and a toString method. (This might be changed to Job Location, Job Email, and Job Title)

The FileExport Class would allow the user to export all the contacts into a text file. This would take all the contacts stored by the current program and write them to a text file. This would allow the user to import the contacts later.

The FileImport Method would allow the users to import contacts by reading the number of contacts within the file (and Integer in the first line) and then looping through the parameters of the programming fill and creating a new contact with each loop.

The Final program will implement a GUI using java swing and will incorporate all of the classes mentioned above. It will be a standard java form application.

III. Requirements

The specific problem that this program is addressing is transferable contacts. Many contact book applications share many of the same functionality as mine such as allowing the user's to create new contacts, but many do not allow the user to export multiple contacts at once. This program has little to no hardware requirements other than a standard computer

IV. Literature Survey

In recent updates with ios, Apple has allowed the user to share a single contact through the use of airdrop or instant messenger. This is useful for single contact sharing, but this does not solve the issue of multiple contacts being shared to a user.

V. User Manual

The first option would prompt the user to enter in all the information of a contact (FirstName, LastName, Number, Country, State, City, Street, Zip-Code, HouseNumber, Job, yearly Salary, and HoursperWeek)

The second option would prompt the user to enter the name of the contact they would like to remove from their contact book.

The third option would print all the contacts in the contact book.

The fourth option would remove all contacts.

The fifth option would prompt the user to enter in the name of the contact they would

like to select and then print the name and number of that contact

The sixth option would prompt the user to enter the name of the contact they would like to select and then print the address of that contact.

The seventh option would prompt the user to enter the name of the contact they would like to select and then print the occupation.

VI. Conclusion

Ultimately the piece of software produced should allow users to efficiently quickly create large amounts of contacts with all the parameters given, and then allow the user to share them among friends resulting in a growing community. A digital contact book that allows users to import contacts can be used in a variety of areas such as startups, internships, large corporations, or personal use.

VII. What's Left.

The backend of the project is completed, all that is left to do is to implement the front end Graphic User Interface. The GUI Will be made using Java Swing and will not require any use of the terminal. It will be independent of the command line and will allow the user easy access to options such as pre-fill drop-down menus. The last thing that needs to be changed is the occupation class. The fields of that class might change depending on what makes more sense, see section II.

VIII References/Bibliography.

"Package Javax.swing." Javax.swing (Java Platform SE 7), Oracle, 6 Oct. 2018, docs.oracle.com/javase/7/docs/api/javax/swing/package-summary.html.

Y. D. Liang, *Introduction to Java Programming*, Comprehensive Ver., 11th Ed.

M. A. Weiss, *Data Structures and Algorithm Analysis in Java*, 3rd Ed.