

Contact Book

Software Development II Final Project

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. This program features a graphic user interface done in java swing.

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, otherwise the program would be exporting nothing and creating an empty text file. 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 from the main menu, 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 from the main menu and enter the name of the text file they would like to import the

contacts from. This will import all current contacts in the contact book.

II. Detailed System Description

The AddressBookGui(Driver) class would contain the main method and declare an ArrayList of 3 objects (Number, Address, Occupation).

This class holds are the major elements of the graphic user interface such as the main menus, submenus for import and export contacts, and the ability to switch between different pages.

The main page of the graphic user interface has eight primary buttons and a menu with two options. The eight options for the main page are (Add Contact, Remove Contacts, Print Specific Contacts, Basic Information, Address Information, Job Information, Import contacts, Export Contacts). The Menu holds two options, (Print all, Delete all).

The “Add Contact” option has 12 input fields and two buttons. The input fields directly relating to the Number, Address and Occupations Classes (Discussed

Later) The 12 fields are “First Name, Last Name, Phone Number, Country, State, City, Road, Zip-Code, House Number, Job Name, Job Email, Job Location”.

The two buttons the Add Contact book has are “Add Contact and Go Back” Once the user fills all fields and clicks “Add Contract” the contact will be added to the contact book. If the user fails to fill in all the fields, the contact simply will be rejected. The “Go Back” Button returns to the previous page.

The Remove Contact page has a combo box that is filled with the names of the currently existing contacts, and two buttons “Remove Contact” and ‘Go Back” The user can select a contact from the drop-down menu and select the “Remove Contact” button to delete that contact.

The Print Specific Contact page has a Combo Box with a list of all existing contracts, text fields to display all contact information, and two buttons “Enter” and go back”. The options “Basic Information”, “Address Information”, and Job

Information” are reflections of the “print specific contact page “ that spring their respective information.

The last two options Import and Export Contacts follow the same principles. They both request a string from the user for a file name. The Import Contacts option will look for a text file with the matching name, and then read all the contacts from the file. The Export contacts option will write to a text file with the name specified by the user.

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 Name, Email Address, and Location. This method would contain getters, setters and a toString method.

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. The program has been written in java meaning that is it cross-compatible with many different operating systems and has the possibility to be converted into an android application.

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. As of right now, sending contacts books is a rare feature.

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 a specific contact in the contact book

The fourth option would print the basic information of a specific contact in the contact book

The fifth option would print the address information of a specific contact in the contact book

The sixth option would print the job information of a specific contact in the contact book

The seventh option would prompt the user to enter a text file and import contacts from that file.

The eighth option would allow the user to Export all contacts to a text file.

The submenu (Top right) has two options, Print all which is only meant to be used for troubleshooting purposes as it prints to a console, and remove all which removes all of the contacts in the contact book.

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 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.

“Java Documentation - Get Started.” *Oracle Help Center*, 8 Oct. 2018, docs.oracle.com/en/java/.