**Requirements**

**Overview**

The Phonebook will allow a user navigate all their desired contact information through this application. The user will have the ability to store this information on a txt file for later use. The application will have an array of options for the different types of contact information they can add, along with different ways they can use that information within the application.

**Functional Requirements**

**Minimum Requirements**

1. **Implement Hashtable** to sort stored information
2. **Implement ArrayLists** in order to sort the contact information
3. **Implement a Main class** for the program to run
4. **Implement a Phonebook class** that contains the methods for the program to function
5. **Adding Contact Information** will allow the user to input contact information such as name, phone number, email, category for such contact, ringtone
6. **Saving Contact Information** to File will allow the user to be able to save all the necessary contact information to a txt file for later use
7. **Loading Contact Information** will allow the user to load saved information from the text file they have previously worked on
8. **Help** will allow the user to view the list of commands that will state all the possible options that the user is able to do
9. **Listing Contact Information** will allow the user to list contact information for the entire phonebook or possibly one category
10. **Removing Contact Information** will allow the user to edit their phonebook by removing certain contacts from the file
11. **Showing Contact Information** will allow the user to see a specific contact entry based on one of the subsets in the contact entry
12. **Searching Contact Information** will allow the user to be able to look up a certain contact based on either name or phone number or email, etc
13. **Sorting the Contacts** will be a feature that will be sorting the contacts as they are entered based on specific categories that the user has designated.
14. **Groups** will be allowing three categories for the user to store contact information in (friends, family, colleagues)
15. **Create new ArrayLists** for groups
16. **Allow for sorting** of arraylists for groups

**Preferred Requirements**

1. **Alphabetizing** will show the user the contact information in a specific category alphabetized by name
2. **Category Choice** will allow the user to decide the names of the categories they want to implement in their phonebook
3. **Ringtone** would allow the user to set one of the ringtone options given to certain users and/or categories

**Non-functional Requirements**

* Language: Java
* Design Pattern: MVC pattern, a main class that runs the program and a phonebook class that contains all the methods: adding, sorting, loading, etc.
* Scalability: data structures will be scalable in order for the user to add new contacts and store them in an arraylist
* Maintainability: a clear and clean structure that will allow the implementation of new code and the ability to quickly resolve issues that may arise during further development
* Usability: UI enables intuitive and easy interactions to add contacts and sort
* Write to a txt file the contact information that’s being added
* Read from a txt file the contact information that’s added