Contact Book Application

Technology: JSE (Core Java)

IDE: Eclipse / Netbeans

Focus: OOPs concepts, Exception Handling, Collections Framework

----

Create a **Contact Book** application with the following functionalities:

1. Add a new contact
2. Display all existing contacts
3. Search contacts based on given search string
4. Update an existing contact
5. Delete an existing contact

This application should be developed using Command Line Interface (CLI) as there is no Graphical User Interface (GUI).

A **Contact** must have the following information:

1. uid: User ID (Mandatory)
   This is an internally generated id that uniquely identifies a contact within the application.
2. firstName: First name (Mandatory)
   This field cannot be empty; it must be between 3 – 15 characters long. If the entered first name is not of this length, an error message should be displayed "**First name should be between 3 to 15 characters in length.**" The system should then prompt for **firstName** to be re-entered.
3. lastName: Last name (Optional)
   This is an optional field with no validation.
4. mobile: mobile phone number (Mandatory)
   This field cannot be empty. It should be 10 characters long, should contain only numbers, and no alphabets or special characters are permitted. If the entered data violates this rule, an error message should be displayed "**Mobile number must have 10 digits with no special characters.**" The system should prompt for **mobile** to be re-entered.
5. email: email address (Mandatory)
   This field cannot be empty. It must be a valid email in the correct format. If it is not valid, an error message should be displayed "**Enter valid email. Example: abc123@xyz.com**" The system should prompt for **email** to be re-entered.
6. workplace: Name of city where the contact resides/works (Optional)
   This is an optional field with no validation.

Generic error messages:

1. If a mandatory field is left blank or has only spaces, display the error message "**This field cannot be left blank.**"
2. If a non-alphabetic character is entered in a character only field (like first name or last name), display the error message **"Please enter only alphabets.".**

3. If an alphabet is entered in a number only field (like mobile number), display the error message **"Please enter only numerals".**

Points that need to be addressed:

1. Validations
2. Exceptions handling
3. Using the most efficient data structure for storing data (Preferably Collections Framework)
4. Following naming conventions and writing clean code
5. Project directory structure

Detailed Specifications

This "Contact Book" application should follow a menu driven approach per the details provided below. When the application is run, the following **main-menu** should be displayed, with a prompt to enter the user's choice.

```
************* Contact Book*********************
    1.  Add Contact
    2.  View Contacts
    3.  Search Contact
    4.  Update Contact
    5.  Delete Contact
    6.  Exit

Enter your choice: _
*********************************************
```

If the value of the choice entered by the user is other than 1 through 6, the following message should be displayed "**Invalid choice. Please enter 1 through 6 only.**" The prompt "**Enter your choice:** "should be displayed again.

When the entered choice is "1"

On entering the choice as "1", the user should be asked to enter the following details, one after the other, but only after each prompt has received valid input data. Displayed below is a valid set of input data provided by the user.

```
First Name: Ramesh
Last Name: Kumar
Mobile: 9998887774
Email: ramesh.kumar@gmail.com
Workplace: Bangalore
```

- If a contact already exists with the entered mobile number, then an error message "**A contact already exists with the entered mobile number.**" should be displayed. The user should be prompted to re-enter the mobile number.
- Whenever there is any input data error on any field, an appropriate error message as described on page 1 should be displayed and the user should be prompted to re-enter the data by displaying the field's prompt.

- If all the validations are successful and there is no error in the entered data, the contact details should get stored with the display of a success message "**Contact has been successfully added with unique id: <<ID_VALUE>>**," where the ID_VALUE is a system generated number. The main menu should then be displayed again.

```
************* Contact Book*******************
   1.  Add Contact
   2.  View Contacts
   3.  Search Contact
   4.  Update Contact
   5.  Delete Contact
   6.  Exit

Enter your choice: _
******************************************
```

When the entered choice is "2"

If there are no contacts in the Contact Book, display the message "**No Contacts created**", and then display the main menu for the user to make a different selection. If Contact details are available, then the same should be displayed as shown below. Following the last record, the main menu should be displayed again.

```
Uid          Firstname      Lastname      Mobile        Email                     WorkPlace

-----------------------------------------------------------------------------------------------

 C001        Lakshman        A             9999998888   lakshman.a@spaneos.com    Bangalore

C002         Krish          T              9999998887   krish.t@spaneos.com       Bangalore

C003         Suresh         S              9999998889   suresh.s@spaneos.com      Bangalore
```

Note: Do not worry about pagination of this data. Let the data (including the column headings) scroll by. The main menu should be displayed after the last record has been displayed.

When the entered choice is "3"

A contact can be searched based on any one of the following: First Name, Last Name, Mobile number, or Email. Accept the user input after the prompt "**Enter text to search:** ", and perform a string search based on these 4 fields. If the user input matches any one of these four fields, that contact information should be displayed. After displaying the search results, the main menu should be displayed.

Examples:

```
Enter text to search: man
```

Search Output:

```
Uid          Firstname      Lastname      Mobile        Email                     Workplace
-----------------------------------------------------------------------------------------------
 C001        Lakshman        A             9999998888   lakshman.a@spaneos.com    Bangalore
```

Enter text to search: 777

Search Output:

No date found

When the entered choice is "4"

Before updating any contact, the system should first display the details of all the contacts.

- The system should then prompt the user to enter the **unique id** of the contact that needs to be updated. "**Enter the unique id to be updated: C002"**
- If there is no contact with the entered unique id, display the message "**No contact exists with the entered unique id**", and display the main menu again for the user to make a selection.
- If a contact exists with the entered unique id, the user should be prompted to enter data for each field (First Name, Last Name, Mobile, Email, Workplace) similar to the "Add Contact" mode. Note that data for each field MUST be re-typed, even if there is no change needed, because the entire record is replaced. Perform the field level validations as described earlier.
- If all validations are successful, store the updated contact details. Display the success message "**Contact information successfully updated**". Finally, display the main menu again.

IMPORTANT: If the updated mobile number belongs to a different contact (other than the one whose record is being updated), display the error message "**Updated mobile number belongs to a different contact. Update cancelled.**". Abort the changes, and display the main menu.

When the entered choice is "5"

Before deleting any contact, the system should first display the details of all the contacts.

- The system should then prompt the user to enter the **unique id** of the contact that needs to be deleted. "**Enter the unique id to be deleted: C002"**
- If there is no contact with the entered unique id, display the message "**No contact exists with the entered unique id**", and display the main menu again for the user to make a selection.
- If a contact exists with the entered unique id, the user should be prompted for a confirmation with the message "**Selected contact will be deleted. Proceed (Y/N)?**".
- If the user enters "**N**", the delete activity is aborted. The system should display a message "**Deletion aborted**", and the main menu should be displayed again.
- If the user enters "**Y**", the selected record should be deleted. The system should display a message "**Selected record deleted successfully.**", and the main menu should be displayed again.

When the entered choice is "6"

- The system should display the message "**Thank you. Please hit any key to exit the system.**".
- Once any key is pressed, the application should terminate.