ADDRESS BOOK SYSTEM

```
import java.util.*;

class Contact {
    String name;
    String address;
    String phoneNumber;
    String email;

    Contact(String name, String address, String phoneNumber, String email) {
        this.name = name;
        this.address = address;
        this.phoneNumber = phoneNumber;
        this.email = email;
    }

    @Override
    public String toString() {
        return name + ", " + address + ", " + phoneNumber + ", " + email;
    }
}

class AddressBook {
    List<Contact> contacts = new ArrayList<>();

    void addContact(Contact contact) {
        contacts.add(contact);
    }

    void removeContact(String name) {
        Contact contact = getContact(name);
        if (contact ≠ null) {
            contacts.remove(contact);
        }
    }

    Contact getContact(String name) {
        for (Contact contact : contacts) {
            if (contact.name.equalsIgnoreCase(name)) {
                return contact;
            }
        }
        return null;
    }

    void editContact(String name, String address, String phoneNumber, String email)
{
        Contact contact = getContact(name);
        if (contact ≠ null) {
            contact.address = address;
```

```
            contact.phoneNumber = phoneNumber;
            contact.email = email;
        }
    }

    void displayContacts() {
        for (Contact contact : contacts) {
            System.out.println(contact.toString());
        }
    }
}

public class Main {
    public static void main(String[] args) {
        AddressBook addressBook = new AddressBook();
        Scanner scanner = new Scanner(System.in);
        while (true) {
            System.out.println("1. Add contact\n2. Edit contact\n3. Remove
contact\n4. Search contact\n5. Display all contacts\n6. Exit");
            int choice = scanner.nextInt();
            scanner.nextLine();  // consume newline left-over
            switch (choice) {
                case 1:
                    System.out.println("Enter name, address, phone number and email
separated by comma: ");
                    String[] details = scanner.nextLine().split(",");
                    addressBook.addContact(new Contact(details[0].trim(),
details[1].trim(), details[2].trim(), details[3].trim()));
                    break;
                case 2:
                    System.out.println("Enter name of the contact to edit: ");
                    String name = scanner.nextLine();
                    System.out.println("Enter new address, phone number and email
separated by comma: ");
                    String[] newDetails = scanner.nextLine().split(",");
                    addressBook.editContact(name, newDetails[0].trim(),
newDetails[1].trim(), newDetails[2].trim());
                    break;
                case 3:
                    System.out.println("Enter name of the contact to remove: ");
                    String removeName = scanner.nextLine();
                    addressBook.removeContact(removeName);
                    break;
                case 4:
                    System.out.println("Enter name of the contact to search: ");
                    String searchName = scanner.nextLine();
                    Contact contact = addressBook.getContact(searchName);
                    if (contact ≠ null) {
                        System.out.println(contact);
```

```
                    } else {
                        System.out.println("Contact not found.");
                    }
                    break;
                case 5:
                    addressBook.displayContacts();
                    break;
                case 6:
                    System.exit(0);
                default:
                    System.out.println("Invalid option! Please choose a valid
operation.");
            }
        }
    }
}
```