```
import java.io.*;
import java.util.ArrayList;
import java.util.Scanner;

class Contact implements Serializable {
    private String name;
    private String phoneNumber;
    private String email;

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

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getPhoneNumber() {
        return phoneNumber;
    }

    public void setPhoneNumber(String phoneNumber) {
        this.phoneNumber = phoneNumber;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    @Override
    public String toString() {
        return "Name: " + name + ", Phone: " + phoneNumber + ", Email: " + email;
    }
}

public class ContactManager {
    private ArrayList<Contact> contacts;
    private Scanner scanner;
    private final String fileName = "contacts.ser";
```

```
    public ContactManager() {
        scanner = new Scanner(System.in);
        contacts = loadContacts();
    }

    private ArrayList<Contact> loadContacts() {
        try (ObjectInputStream ois = new ObjectInputStream(new
FileInputStream(fileName))) {
            return (ArrayList<Contact>) ois.readObject();
        } catch (FileNotFoundException e) {
            System.out.println("No existing contact list found. Starting a new
one.");
        } catch (IOException | ClassNotFoundException e) {
            System.out.println("Error loading contacts. Starting a new list.");
        }
        return new ArrayList<>();
    }

    private void saveContacts() {
        try (ObjectOutputStream oos = new ObjectOutputStream(new
FileOutputStream(fileName))) {
            oos.writeObject(contacts);
            System.out.println("Contacts saved successfully!");
        } catch (IOException e) {
            System.out.println("Error saving contacts.");
        }
    }

    public void addContact() {
        System.out.print("Enter name: ");
        String name = scanner.nextLine();
        System.out.print("Enter phone number: ");
        String phoneNumber = scanner.nextLine();
        System.out.print("Enter email address: ");
        String email = scanner.nextLine();

        Contact newContact = new Contact(name, phoneNumber, email);
        contacts.add(newContact);
        System.out.println("Contact added successfully!");
        saveContacts();
    }

    public void viewContacts() {
        if (contacts.isEmpty()) {
            System.out.println("No contacts available.");
            return;
        }
        System.out.println("Contact List:");
        for (int i = 0; i < contacts.size(); i++) {
            System.out.println((i + 1) + ". " + contacts.get(i));
```

```
        }
    }

    public void editContact() {
        viewContacts();
        if (contacts.isEmpty()) return;

        System.out.print("Enter the contact number to edit: ");
        int index = scanner.nextInt() - 1;
        scanner.nextLine();

        if (index >= 0 && index < contacts.size()) {
            Contact contact = contacts.get(index);
            System.out.print("Enter new name (" + contact.getName() + "): ");
            String name = scanner.nextLine();
            System.out.print("Enter new phone number (" + contact.getPhoneNumber()
+ "): ");
            String phoneNumber = scanner.nextLine();
            System.out.print("Enter new email address (" + contact.getEmail() + "):
");
            String email = scanner.nextLine();

            contact.setName(name.isEmpty() ? contact.getName() : name);
            contact.setPhoneNumber(phoneNumber.isEmpty() ? contact.getPhoneNumber()
: phoneNumber);
            contact.setEmail(email.isEmpty() ? contact.getEmail() : email);

            System.out.println("Contact updated successfully!");
            saveContacts();
        } else {
            System.out.println("Invalid contact number.");
        }
    }

    public void deleteContact() {
        viewContacts();
        if (contacts.isEmpty()) return;

        System.out.print("Enter the contact number to delete: ");
        int index = scanner.nextInt() - 1;
        scanner.nextLine();

        if (index >= 0 && index < contacts.size()) {
            contacts.remove(index);
            System.out.println("Contact deleted successfully!");
            saveContacts();
        } else {
            System.out.println("Invalid contact number.");
        }
    }
```

```
    public void showMenu() {
        int choice;
        do {
            System.out.println("\nContact Manager Menu:");
            System.out.println("1. Add Contact");
            System.out.println("2. View Contacts");
            System.out.println("3. Edit Contact");
            System.out.println("4. Delete Contact");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();
            scanner.nextLine(); // Consume newline

            switch (choice) {
                case 1:
                    addContact();
                    break;
                case 2:
                    viewContacts();
                    break;
                case 3:
                    editContact();
                    break;
                case 4:
                    deleteContact();
                    break;
                case 5:
                    System.out.println("Exiting...");
                    saveContacts();
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        } while (choice != 5);
    }

    public static void main(String[] args) {
        ContactManager manager = new ContactManager();
        manager.showMenu();
    }
}
```