

# Java MVC Address Book Application

## 1. Project Description

This Address Book application is developed using the Model-View-Controller (MVC) architecture in Java. The application allows users to store, view, and manage contact information such as name, phone number, and email address. It is a console-based application that demonstrates the separation of concerns using MVC.

## 2. MVC Architecture Explanation

Model: Represents the data and business logic. In this project, the Contact class is the model.

View: Responsible for displaying information to the user. The AddressBookView handles user interaction.

Controller: Connects the model and view. It processes user input and updates the model and view accordingly.

## 3. Project Structure

```
AddressBookApp
  |
  +-- model
    +-- Contact.java
  +-- view
    +-- AddressBookView.java
  +-- controller
    +-- AddressBookController.java
      +-- AddressBookService.java
  +-- Main.java
```

## 4. Source Code

### Contact.java (Model)

```
package model;

public class Contact {
    private int id;
    private String name;
    private String phone;
    private String email;

    public Contact(int id, String name, String phone, String email) {
        this.id = id;
```

```

        this.name = name;
        this.phone = phone;
        this.email = email;
    }

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public String getPhone() {
        return phone;
    }

    public String getEmail() {
        return email;
    }

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

```

#### [AddressBookService.java \(Controller Logic\)](#)

```

package controller;

import model.Contact;
import java.util.ArrayList;
import java.util.List;

public class AddressBookService {
    private List<Contact> contacts = new ArrayList<>();
    private int idCounter = 1;

    public void addContact(String name, String phone, String email) {

```

```

        Contact contact = new Contact(idCounter++, name, phone, email);
        contacts.add(contact);
    }

    public List<Contact> getAllContacts() {
        return contacts;
    }

    public void deleteContact(int id) {
        contacts.removeIf(contact -> contact.getId() == id);
    }
}

```

### [AddressBookView.java \(View\)](#)

```

package view;

import model.Contact;
import java.util.List;

public class AddressBookView {

    public void showMenu() {
        System.out.println("\n--- Address Book ---");
        System.out.println("1. Add Contact");
        System.out.println("2. View Contacts");
        System.out.println("3. Delete Contact");
        System.out.println("4. Exit");
        System.out.print("Choose an option: ");
    }

    public void displayContacts(List<Contact> contacts) {
        if (contacts.isEmpty()) {
            System.out.println("No contacts found.");
            return;
        }
        for (Contact contact : contacts) {
            System.out.println(contact);
        }
    }

    public void showMessage(String message) {
        System.out.println(message);
    }
}

```

```
    }  
}
```

### **AddressBookController.java (Controller)**

```
package controller;  
  
import view.AddressBookView;  
import java.util.Scanner;  
  
public class AddressBookController {  
    private AddressBookService service;  
    private AddressBookView view;  
  
    public AddressBookController(AddressBookService service, AddressBookView view) {  
        this.service = service;  
        this.view = view;  
    }  
  
    public void start() {  
        Scanner scanner = new Scanner(System.in);  
        int choice;  
  
        do {  
            view.showMenu();  
            choice = scanner.nextInt();  
            scanner.nextLine();  
  
            switch (choice) {  
                case 1:  
                    System.out.print("Enter name: ");  
                    String name = scanner.nextLine();  
  
                    System.out.print("Enter phone: ");  
                    String phone = scanner.nextLine();  
  
                    System.out.print("Enter email: ");  
                    String email = scanner.nextLine();  
  
                    service.addContact(name, phone, email);  
                    view.showMessage("Contact added successfully.");  
                    break;  
            }  
        } while (choice != 0);  
    }  
}
```

```

        case 2:
            view.displayContacts(service.getAllContacts());
            break;

        case 3:
            System.out.print("Enter contact ID to delete: ");
            int id = scanner.nextInt();
            service.deleteContact(id);
            view.showMessage("Contact deleted.");
            break;

        case 4:
            view.showMessage("Exiting application.");
            break;

        default:
            view.showMessage("Invalid choice.");
    }

} while (choice != 4);

scanner.close();
}
}

```

### Main.java (Entry Point)

```

import controller.AddressBookController;
import controller.AddressBookService;
import view.AddressBookView;

public class Main {
    public static void main(String[] args) {
        AddressBookService service = new AddressBookService();
        AddressBookView view = new AddressBookView();
        AddressBookController controller = new AddressBookController(service, view);
        controller.start();
    }
}

```

## 5. Sample Output

--- Address Book ---

1. Add Contact
2. View Contacts
3. Delete Contact
4. Exit

Choose an option: 1

Enter name: Vindya

Enter phone: 9988774455

Enter email: vindya@gmail.com

Contact added successfully.

--- Address Book ---

1. Add Contact
2. View Contacts
3. Delete Contact
4. Exit

Choose an option: 2

ID: 1, Name: Rahul Sharma, Phone: 9876543210, Email: rahul@gmail.com

```
--- Address Book ---
```

1. Add Contact
2. View Contacts
3. Search Contact
4. Update Contact
5. Delete Contact
0. Exit

Enter choice: 1

Name: *Vindya*

Phone: *9988774455*

Email: *vindya@gmail.com*

Contact added successfully.

```
--- Address Book ---
```

1. Add Contact
2. View Contacts
3. Search Contact
4. Update Contact
5. Delete Contact
0. Exit

Enter choice: 2

Name: *Vindya*, Phone: *9988774455*, Email: *vindya@gmail.com*