COMSATS UNIVERSITY ISLAMABAD

OBJECT ORIENTED PROGRAMMING

(CPP)

CEP REPORT

NAME:

1. MUNEEB AHMAD
2. JAMAL KHAN

REGISTRATION NO:

1. FA21-BEE-145
2. FA20-BEE-078

SUBMITTED TO:

Maam Nayab Gogosh

Contact Book Application in Flutter

Project Overview

This project is a Contact Book Application developed using Flutter, a modern UI toolkit for building natively compiled applications for mobile, web, and desktop from a single codebase. The application demonstrates core concepts of Object-Oriented Programming (OOP) such as Encapsulation and Aggregation, and implements a simple yet effective UI to manage contacts grouped by categories.

Objectives

Develop a cross-platform mobile/web application using Flutter.
Implement OOP principles by modeling contacts and groups.
Provide an interactive UI to manage contact groups.
Demonstrate the use of list management and aggregation relationships.

System Design

Class Design

1. Contact Class (Encapsulation)

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

  Contact({required this.name, required this.phoneNumber, required this.email});
}
```

2. Group Class (Aggregation)

```
class Group {
  String name;
  List<Contact> contacts;

  Group({required this.name}) : contacts = [];

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

Application UI

The main screen shows a list of groups (e.g., Family, Friends, Work). Each group can be expanded to view or add contacts. A search bar allows filtering contacts.

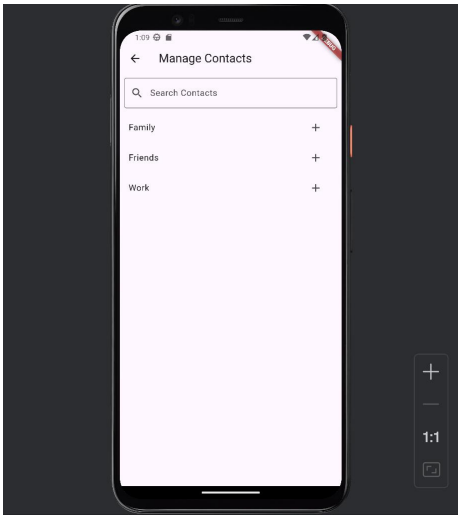

Figure 1: User Interface

Development & Testing

Developed using Flutter and run on Chrome (web) for testing. Also tested on a virtual Android device (Pixel 4 XL API). Logs show the app successfully launched and connected to the debug service.

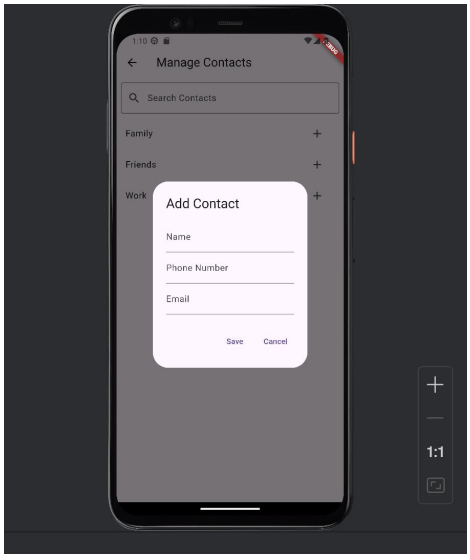

Figure 2: Add Contacts

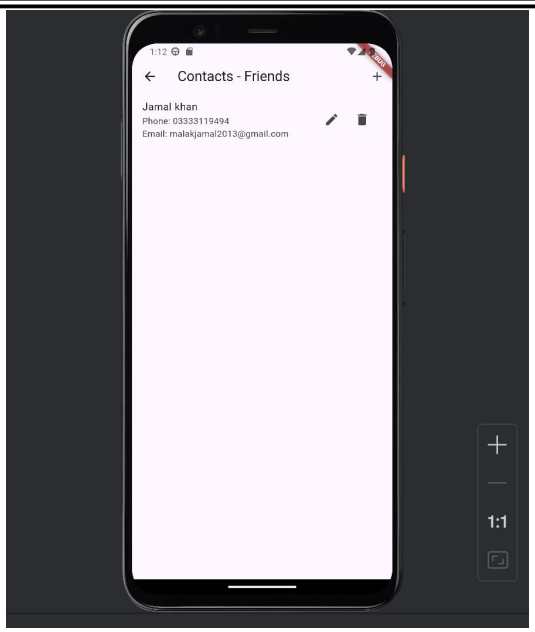

Figure 3: Saved Contacts

Key Features

Contact management with encapsulated data.
Grouping functionality using aggregation.
Expandable list UI for groups.
Cross-platform support (Android, Web).
Clean and minimal user interface.

Conclusion

This project efficiently demonstrates OOP principles in Dart with Flutter, focusing on encapsulation and aggregation. The UI is simple yet practical for managing categorized contact lists. This project can be extended by integrating local storage (SQLite, Hive) or cloud databases (Firebase) for persistent data management.