Object Oriented Programming

PROJECT REPORT

By

Priya Sharma (08301012019)
Sonia Verma (10701012019)
Astha (13201012019) Nidhi Yadav (15101012019)

Guided by

MS. VIBHA PRATAP
Assistant Professor CSE Department

INDIRA GANDHI DELHI TECHNICAL UNIVERSITY FOR WOMEN
NEW DELHI – 110006

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our Object Oriented Programming teacher, **MS. VIBHA PRATAP** for her immense support and guidance in the making of this project.

We would also like to thank the whole team for working so passionately since the ideation phase of this project.

DECLARATION

We solemnly declare that the project report is based on our own work carried out during the course of our study under the supervision of **MS. VIBHA PRATAP.** We assert the statements made and conclusions drawn are an outcome of our research work. We further certify that:

1. The work contained in the report is original and has been done by us under the supervision of our supervisor.
2. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other University of India or abroad.
3. We have followed the guidelines provided by the university in writing the report.
4. Whenever we have used materials (text, data, theoretical analysis/equations, codes/program, figures, tables, pictures, text etc.) from other sources, we have given due credit to them in the report and have also given their details in the references.

Priya Sharma (08301012019)

Sonia Verma (10701012019)

Astha (13201012019)

Nidhi Yadav (15101012019)

TABLE OF FIGURES

INDEX

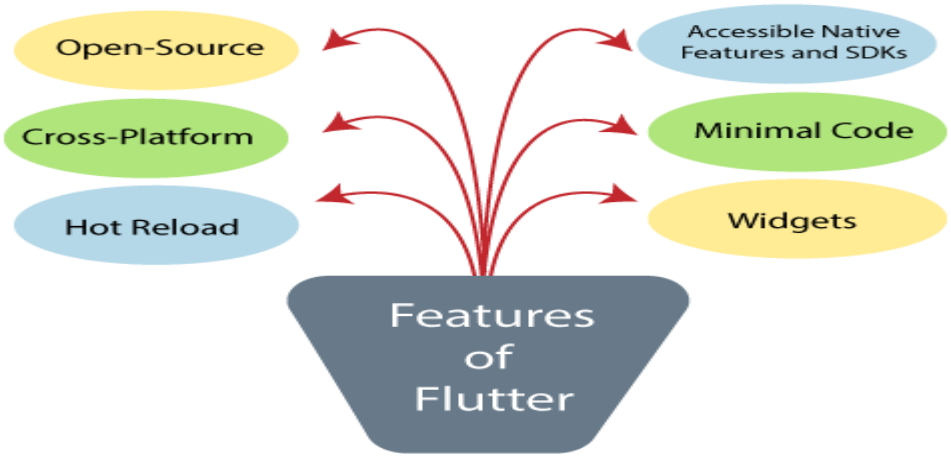
WHAT IS DART?

- Dart is an open-source, general-purpose, object-oriented programming language with C-style syntax.
- The purpose of Dart programming is to create a frontend user interfaces for the web and mobile apps. Since Dart is a compiled language so you cannot execute your code directly; instead, the compiler parses it and transfer it into machine code.
- It supports most of the common concepts of programming languages like classes, interfaces, functions, unlike other programming languages. Dart language does not support arrays directly. It supports collection, which is used to replicate the data structure such as arrays, generics, and optional typing.
- The following example shows simple Dart programming.

```
void main() {
   for (int i = 0; i < 5; i++) {
       print('hello ${i + 1}');
          }
       }
```

WHAT IS FLUTTER?

- Flutter is a UI toolkit for creating fast, beautiful, natively compiled applications for mobile, web, and desktop with one programing language and single codebase. It is free and open-source.
- Flutter apps use Dart programming language for creating an app.
- Flutter is mainly optimized for 2D mobile apps that can run on both Android and iOS platforms. We can also use it to build full-featured apps, including camera, storage, geolocation, network, third-party SDKs, and more.
- Flutter is different from other frameworks because it neither uses **WebView** nor the **OEM** widgets that shipped with the device. Instead, it uses its own high-performance rendering engine to draw widgets.
- It also implements most of its systems such as animation, gesture, and widgets in Dart programing language that allows developers to read, change, replace, or remove things easily. It gives excellent control to the developers over the system.



HOW OOPS CONCEPTS ARE IMPLEMENTED IN THE PROJECT

Use of class

Dart is an object-oriented language. It supports object-oriented programming features like classes, interfaces, etc. A **class** in terms of OOP is a blueprint for creating objects. A **class** encapsulates data for the object. Dart gives built-in support for this concept called **class**.

Polymorphism (Method Overriding)

Polymorphism is exemplified in Dart by the @override metatag. With it, a subclass's implementation of an inherited behavior can be specialized to be appropriate to its more specific subtype. When a class has properties that are themselves instances of other classes, it's using composition to add to its abilities.

Method overriding occurs in dart when a child class tries to override the parent class's method. When a child class extends a parent class, it gets full access to the methods of the parent class and thus it overrides the methods of the parent class. It is achieved by re-defining the same method present in the parent class.

Inheritance in dart

In Dart, the extends keyword is typically used to alter the behavior of a class using Inheritance. The capability of a class to derive properties and characteristics from another class is called **Inheritance**. It is ability of a program to create new class from an existing class. In simpler words, we can say that we use extends to create a subclass, and super to refer to the superclass. The class whose properties are inherited by

child class is called **Parent Class**. Parent class is also known as *base class* or **super class**.The class that inherits properties from another class is called **child class**.

Encapsulation in Dart

Encapsulation in Dart happens at library level instead of class level unlike other object oriented programming languages.
In Dart, (_) underscores in Dart are private. This wraps up the series on object oriented programming.

ABOUT

Co.De.code is a QR/Barcode scanner and generator app. It is extremely easy to use; with quick scan built in simply point it will automatically scan the code. There is no need to press any buttons, take photos or adjust zoom as Co.De.code reader works automatically.

The app can scan and read all QR codes / barcode types including text, url, product, contact, calendar, email, location, Wi-Fi and many more formats. After scan and automatic decoding user is provided with only the relevant options for individual QR or Barcode type and can take appropriate action. You can even use the app to scan coupons/ coupon codes to receive discounts and save some money.

Co.De.code can also be used to scan product barcodes with bar code reader in shops and compare prices with online prices to save money. Not only scanning the app can generate QR code too. It is a complete app in itself for scanning and generating all types of QR codes and can be used anywhere.

IMPLEMENTATION

Dependencies:
(in pubspsec.yaml)

- cupertino_icons: ^1.0.2
- flutter_barcode_scanner: ^2.0.0
- barcode_widget: ^2.0.1

main.dart
This is the main driver file of the app, in this the primary structure of the app is build. This file has three classes:-

- MyApp – in this class the theme and the title of the app is coded, this class also calls another class MyHomePage.
- MyHomePage – the title of the bar is passed.
- _MyHomePageState – extends to class MyHomePage, calls ScanScreen() and CreateScreen() classes, in this class the navigation functionality, buttons, widget alignment and size are implemented.

```
import 'package:mycode/screens/scanqr.dart';
import 'package:mycode/screens/createqr.dart';
import 'package:flutter/material.dart';

void main() {
  runApp(MyApp());

}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(

      debugShowCheckedModeBanner: false,
      title: 'B C R',
      theme: ThemeData(
        primarySwatch: Colors.pink,
        scaffoldBackgroundColor: Colors.black,
      ),
      home: MyHomePage(title: 'Co.De.code'),
    );
  }
}
```

```
class MyHomePage extends StatefulWidget {
  MyHomePage({Key key, this.title}) : super(key: key);

  final String title;

  @override
  _MyHomePageState createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.pink,
        title: Text(widget.title),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          crossAxisAlignment: CrossAxisAlignment.center,
          children: <Widget>[
            ElevatedButton(
              onPressed: () {
                print("tapped on create QR button.");
                Navigator.of(context).push(
                  MaterialPageRoute(
                    builder: (ctx) => CreateScreen(),
                  ),
                );
              },
              child: Text("Create QR"),
            ),
            ElevatedButton(
              onPressed: () {
                print("tapped on scan QR button.");
                Navigator.of(context).push(
                  MaterialPageRoute(
                    builder: (ctx) => ScanScreen(),
                  ),
                );
              },
              child: Text("Scan QR"),
            ),
          ],
        ),
      ),
    );
  }
}
```

createqr.dart – for creating the QR code
The class _CreateScreenState extends to CreateScreen
Makes another screen to generate QR code in which the user can input data and the QR will be generated.

```
import 'package:flutter/material.dart';
import 'package:barcode_widget/barcode_widget.dart';
import 'package:flutter/services.dart';

class CreateScreen extends StatefulWidget {
  @override
  _CreateScreenState createState() => _CreateScreenState();
}

class _CreateScreenState extends State<CreateScreen> {
  String qrString = "Add Data";
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Create QR Code"),
      ),
      body: Column(
        crossAxisAlignment: CrossAxisAlignment.center,
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        children: [
          // qr code
          BarcodeWidget(
            color: Colors.grey,
            data: qrString,
            height: 250,
            width: 250,
            barcode: Barcode.qrCode(),
          ),
          // link
          Container(
            width: MediaQuery.of(context).size.width * .8,
            child: TextField(
              style: TextStyle(color: Colors.pinkAccent),
              autofocus: false,
              cursorColor: Colors.pink,
              onChanged: (val) {
                setState(() {
                  qrString = val;
                });
              },
              decoration: InputDecoration(
                fillColor: Colors.white,
                hintText: "Enter you data here",
                hintStyle: TextStyle(fontSize: 20.0, color: Colors.pink),
                border: OutlineInputBorder(
```

```
                  borderSide: BorderSide(
                    color: Colors.pink,
                    width: 2,
                  ),
                ),
              ),
            ),
          ),
          SizedBox(
            width: MediaQuery.of(context).size.width,
          ),
        ],
      ),
    );
```

scanqr.dart – For scanning the QR code

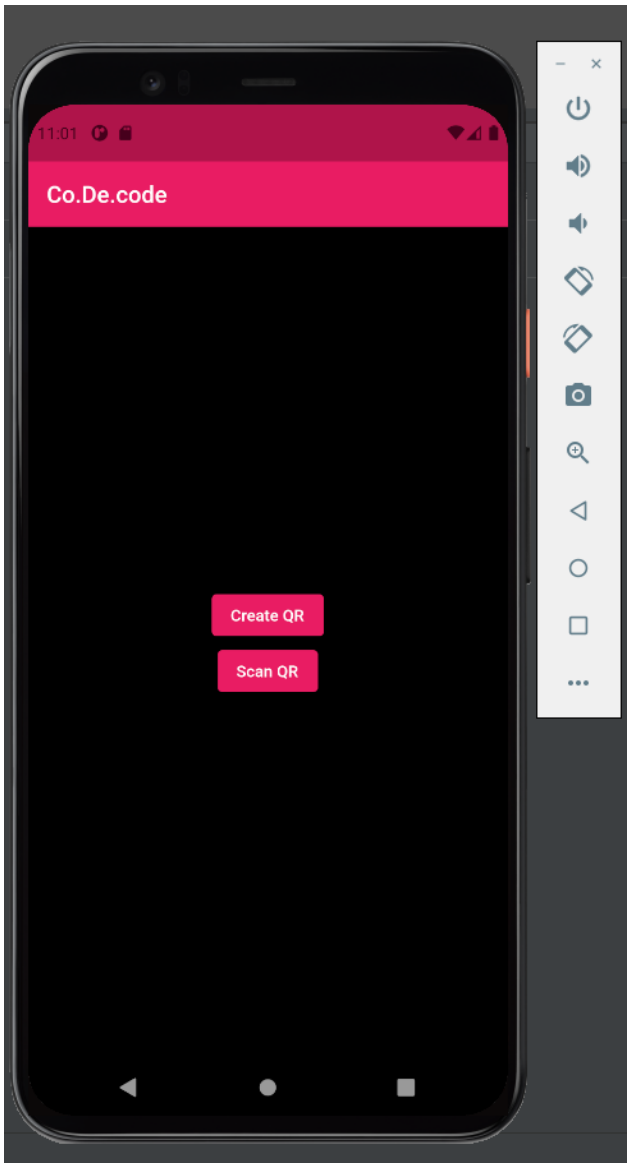
The class _ScanScreenState extends to ScanScreen

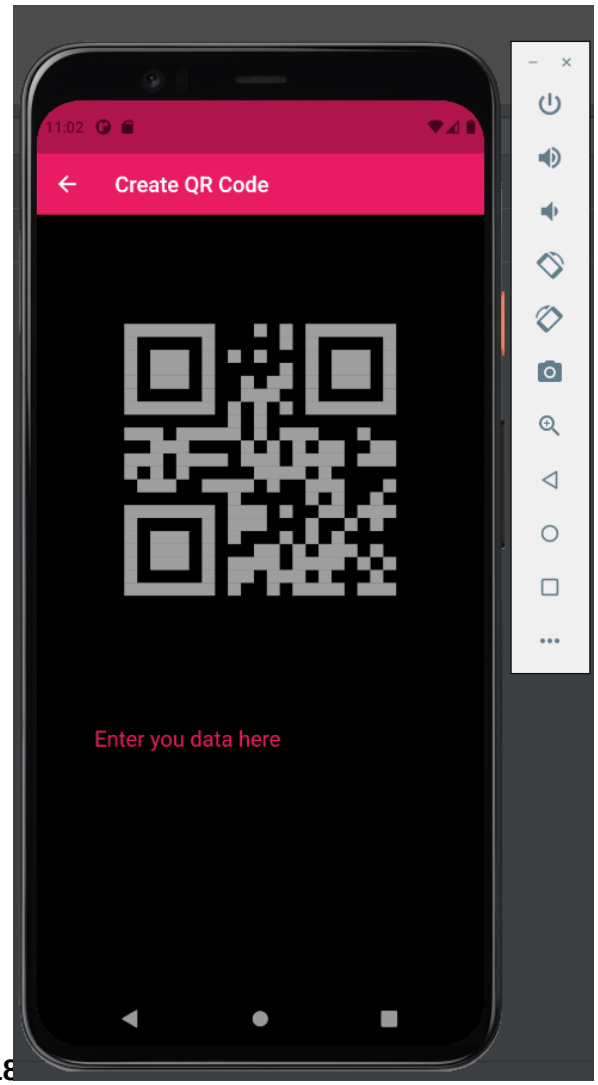
Makes another screen for scanning the QR code. The app asks for the camera permission of the device, in case the app doesn't scan the code due to camera issues it returns output as -1

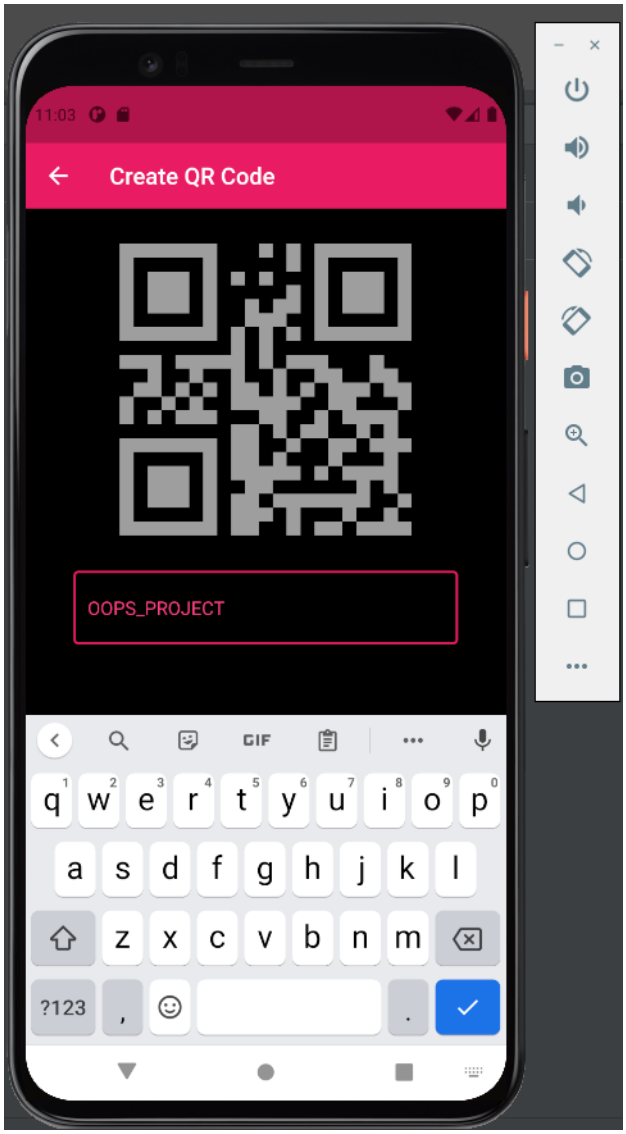
```
import 'package:flutter/material.dart';
import 'package:flutter_barcode_scanner/flutter_barcode_scanner.dart';

class ScanScreen extends StatefulWidget {
  @override
  _ScanScreenState createState() => _ScanScreenState();
}

class _ScanScreenState extends State<ScanScreen> {
  double height, width;
  String qrString = "Not Scanned";
  @override
  Widget build(BuildContext context) {
    height = MediaQuery.of(context).size.height;
    width = MediaQuery.of(context).size.width;
    return Scaffold(
      appBar: AppBar(
        title: Text("Scan QR Code"),
      ),
      body: Column(
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        children: [
          Text(
            qrString,
            style: TextStyle(color: Colors.pink, fontSize: 30),
          ),
          ElevatedButton(
            onPressed: scanQR,
            child: Text("Scan QR Code"),
          ),
```

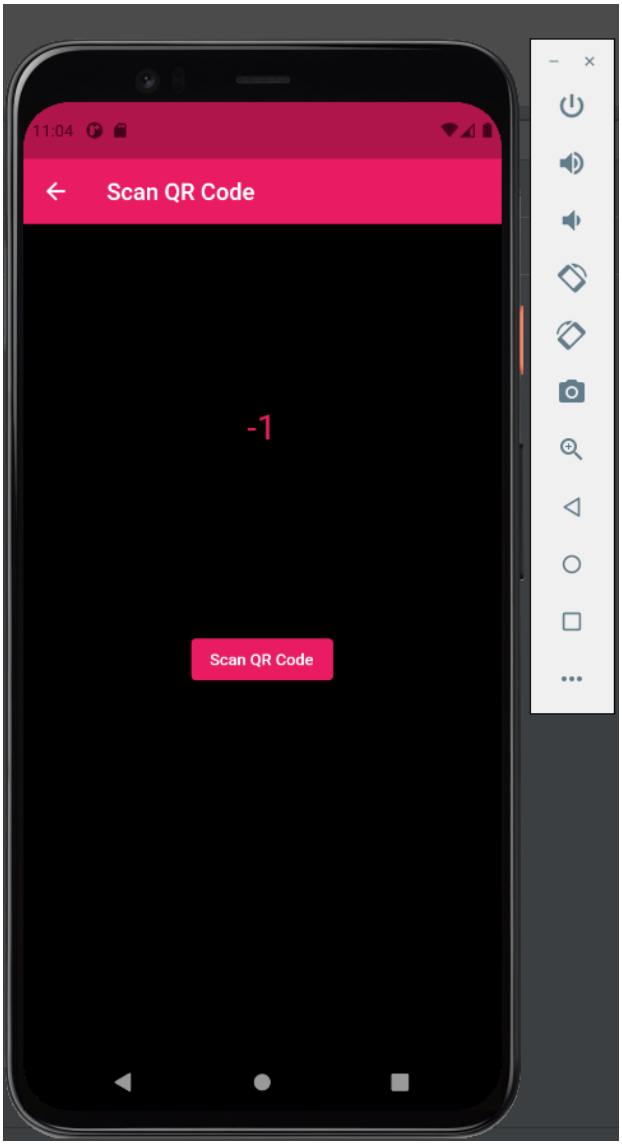
```
          SizedBox(width: width),
        ],
      ),
    );
  }

  Future<void> scanQR() async {
    try {
      FlutterBarcodeScanner.scanBarcode("#2A99CF", "Cancel", true, ScanMode.QR)
          .then((value) {
        setState(() {
          qrString = value;
        });
      });
    } catch (e) {
      setState(() {
        qrString = "unable to read the qr";
      });
    }
  }
```

Output


11:01
Co.De.code
Create QR
Scan QR


11:02
Create QR Code
Enter you data here


11:03
Create QR Code
OOPS_PROJECT


11:04
Scan QR Code
-1
Scan QR Code

BIBLIOGRAPHY

- https://dart.dev/
- https://www.javatpoint.com/flutter-dart-programming
- https://www.tutorialspoint.com/dart_programming/index.htm
- https://developers.google.com/learn/topics/dart
- https://dev.to/kcooperdev/encapsulation-in-dart-part-5-1oo4#:~:text=Encapsulation%20is%20use%20to%20hide,the%20scope%20of%20the%20function.&text=NOTE%3A%20Encapsulation%20in%20Dart%20happens,underscores%20in%20Dart%20are%20private.
- https://dart.academy/inheritance-polymorphism-and-composition-in-dart-and-flutter/#:~:text=Polymorphism%20is%20exemplified%20in%20Dart,to%20add%20to%20its%20abilities.
- https://www.geeksforgeeks.org/dart-extends-vs-with-vs-implements/#:~:text=In%20Dart%2C%20the%20extends%20keyword,of%20a%20class%20using%20Inheritance.&text=The%20class%20that%20inherits%20properties,the%20typical%20OOP%20class%20inheritance.

Project Link:

https://drive.google.com/drive/folders/1ZbNIRL7uaOxBlriYjT2emgnS8DpOYLA8?usp=sharing