Text

Description automatically generated

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **Assignment No** | Assignment No 07 |

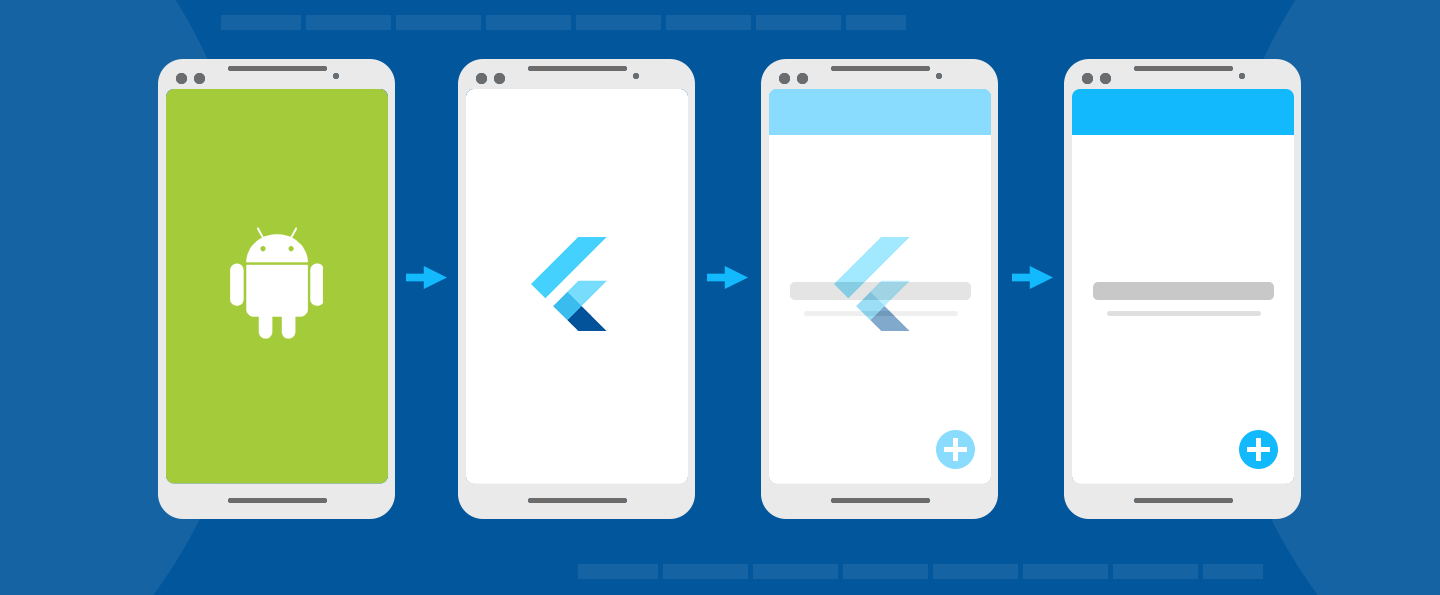
Assignment Number - 07

**Title :** Implementation of Splashscreen by using Timer Function

**Theory :**

**SplashScreen**

Splash screens (also known as launch screens) provide a simple initial experience while your Android app loads. They set the stage for your application, while allowing time for the app engine to load and your app to initialize.



In Android, there are two separate screens that you can control: a launch screen shown while your Android app initializes, and a splash screen that displays while the Flutter experience initializes.

The common elements of a splash screen contain a company name and logo or a title.

* Branding or identity recognition of the application
* loading progress indicator
* Connectivity with Home screen

**Implementation of SplashScreen**

1. **Timer Function:**

The timer function contains two arguments where the first is duration, and the second is action to be performed.

initState();

 Timer(Duration(seconds: 5),

()=>Navigator.pushReplacement(context,

MaterialPageRoute(builder:

  (context) => HomeScreen()

Timer class abstract interface A countdown timer that can be configured to fire once or repeatedly.

The timer counts down from the specified duration to 0. When the timer reaches 0, the timer invokes the specified callback function. Use a periodic timer to repeatedly count down the same interval.

1. **Splashscreen Package**

SplashScreen ({ Color loaderColor,

int seconds,

double photoSize,

Image image,

Text loadingText,

Color backgroundColor,

Text title,

TextStyle styleTextUnderTheLoader,

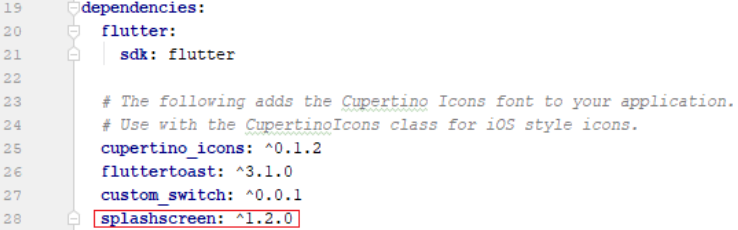
dynamic onClick,

dynamic navigateAfterSeconds,

ImageProvider<dynamic> imageBackground,

Gradient gradientBackground})

**While using spashscreen package add dependencies in pubspec.yaml**

****

**dart:async library**

It Support for asynchronous programming, with classes such as Future and Stream.

Futures and Streams are the fundamental building blocks of asynchronous programming in Dart. They are supported directly in the language through async and async\* functions, and are available to all libraries through the dart:core library.

This library provides further tools for creating, consuming and transforming futures and streams, as well as direct access to other asynchronous primitives like Timer, scheduleMicrotask and Zone.

**Source Code:-**

import 'dart:async';

import 'package:flutter/material.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Splash Screen',

theme: ThemeData(

primarySwatch: Colors.green,

),

home: MyHomePage(),

debugShowCheckedModeBanner: false,

);

}

}

class MyHomePage extends StatefulWidget {

@override

\_MyHomePageState createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

@override

void initState() {

super.initState();

Timer(Duration(seconds: 10),

()=>Navigator.pushReplacement(context,

MaterialPageRoute(builder:

(context) =>

SecondScreen()

)

)

);

}

@override

Widget build(BuildContext context) {

return Container(

color: Colors.white,

child:Image.network('https://media.tenor.com/Wg9fW\_XEft0AAAAM/pout-christian-bale.gif',

height: 100,

width: 100,

fit: BoxFit.fitWidth,

), //FlutterLogo(size:MediaQuery.of(context).size.height)

);

}

}

class SecondScreen extends StatelessWidget {

@override

Widget build(BuildContext context) {

return Scaffold(

appBar: AppBar(title:Text("Sigma Male Application")),

body: Center(

child:Text("Sigma Male",textScaleFactor: 2,)

),

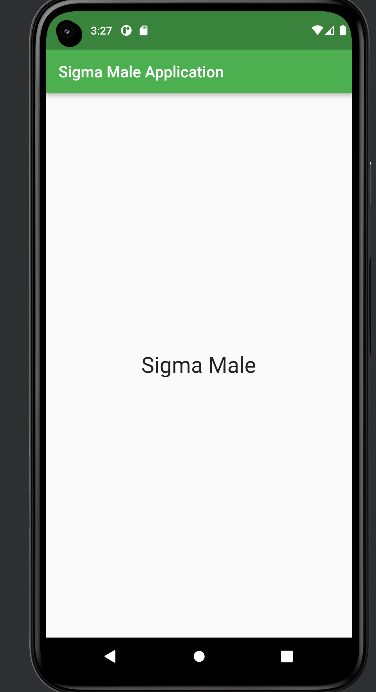
);

}

}

**Output:**





**Conclusion : In this assignment I have learn about Implementation of SplashScreen and its packages.**