****

# **Semester 2019/2020 (A191)**

**School of Computing, CAS, UUM**

**STIW2044: Mobile Programming**

**Lab 2: First Deliverable**

Name: Nur Aini Syamimi Binti Md Rozilan

Matric No.: 245480

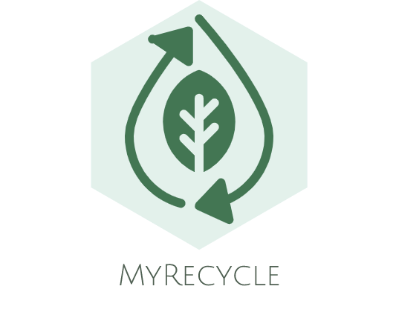
Lecturer: Sir Ahmad Hanis Mohd Shabli

Due Date: 25th October 2019

# Application Name

MyRecycle (for Agent only)

# Application Logo



# Application Description

MyRecycle is an application that involve both parties from users and company. It is about people who sell or buy unused things but still can be reused or recycled such as paper, drink can, box, plastic (bottle/egg carton), book, wire, metal, and cloth. The logo itself shows two-way arrow symbol as it reminds us on recycle and leaf remind us on nature. Assume that MyRecycle is a company that have branches over Malaysia and open agent who may sign up under their company. Those agents might available on certain places only. So, agent from each place is needed as to collect recycle item.

# Features

## Agent/Runner

1. Agent profile – agent should provide details such as picture, username, email, phone, and city and state as their address. They don’t need to provide full address as they only pickup recycle item.
2. Manage item category – not all company will buy all type of recycle item as mentioned. Some agent might only accept cloth while there is agent might not buy cloth as recycle item. Agent should able to provide price for each recycle category per kilogram (kg). However, agent able to update price, add or delete the recycle category.
3. Manage customer – agent can view customer details such as customer name, address, slot, items’ category, and pickup status. However, agent might add description for their references before change the pickup status.
4. View job history – agent can view their job history by select month and year.

# Darts File

1. splashscreen.dart

import 'package:flutter/material.dart';

import 'loginscreen.dart';

void main() => runApp(MySplash());

class MySplash extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      color: Color.fromRGBO(161, 219, 123,1),

      debugShowCheckedModeBanner: false,

      home: Scaffold(

        backgroundColor: Color.fromRGBO(8, 3, 20, 1),

        body: Center(

          child: Column(

            mainAxisAlignment: MainAxisAlignment.center,

            children: <Widget>[

              Image.asset(

                'assets/images/logo.png',

                scale: 2,

              ),

              SizedBox(

                height: 20,

              ),

              new ProgressIndicator(),

            ],

          ),

        ),

      ),

    );

  }

}

class ProgressIndicator extends StatefulWidget {

  @override

  \_ProgressIndicatorState createState() =>

      \_ProgressIndicatorState();

}

class \_ProgressIndicatorState

    extends State<ProgressIndicator>

    with SingleTickerProviderStateMixin {

  AnimationController controller;

  Animation<double> animation;

  @override

  void initState() {

    super.initState();

    controller = AnimationController(

        duration: const Duration(milliseconds: 2000), vsync: this);

    animation = Tween(begin: 0.0, end: 1.0).animate(controller)

      ..addListener(() {

        setState(() {

          if (animation.value > 0.99) {

            //print('Sucess Login');

            Navigator.pushReplacement(

                context,

                MaterialPageRoute(

                    builder: (BuildContext context) => LoginPage()));

          }

        });

      });

    controller.repeat();

  }

  @override

  void dispose() {

    controller.stop();

    super.dispose();

  }

  @override

  Widget build(BuildContext context) {

    return new Center(

        child: new Container(

      width: 20,

      height: 20,

      color: Colors.transparent,

      child: CircularProgressIndicator(

        value: animation.value,

        backgroundColor: Colors.transparent,

        valueColor:

            new AlwaysStoppedAnimation<Color>(Color.fromRGBO(161, 219, 123,1)),

      ),

    ));

  }

}

1. loginscreen.dart

import 'package:flutter/material.dart';

import 'package:shared\_preferences/shared\_preferences.dart';

import 'package:toast/toast.dart';

import 'package:flutter/services.dart';

bool isChecked = true;

final TextEditingController emcontroller = TextEditingController();

final TextEditingController passcontroller = TextEditingController();

String email = '';

String password = '';

FocusNode myFocusNode = new FocusNode();

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      home: LoginPage(),

    );

  }

}

class LoginPage extends StatefulWidget {

  @override

  \_LoginPageState createState() => \_LoginPageState();

}

class \_LoginPageState extends State<LoginPage> {

  @override

  Widget build(BuildContext context) {

    return new Scaffold(

      backgroundColor: Color.fromRGBO(8, 3, 20, 1),

      resizeToAvoidBottomPadding: false,

      body: new Container(

        padding: EdgeInsets.all(30),

        child: Column(

          mainAxisAlignment: MainAxisAlignment.center,

          children: <Widget>[

            Image.asset(

              'assets/images/logo.png',

              scale: 1.7,

            ),

            TextField(

                controller: emcontroller,

                style: TextStyle(color: Colors.white),

                keyboardType: TextInputType.emailAddress,

                focusNode: myFocusNode,

                decoration: InputDecoration(

                    labelText: 'Email',

                    icon: Icon(Icons.email),

                    labelStyle: TextStyle(

                        color: myFocusNode.hasFocus

                            ? Colors.blue

                            : Color.fromRGBO(162, 199, 131, 0.5)))),

            TextField(

              controller: passcontroller,

              style: TextStyle(color: Colors.white),

              //focusNode: myFocusNode,

              decoration: InputDecoration(

                  labelText: 'Password',

                  icon: Icon(Icons.lock),

                  labelStyle: TextStyle(

                      color: myFocusNode.hasFocus

                          ? Colors.blue

                          : Color.fromRGBO(162, 199, 131, 0.5))),

              obscureText: true,

            ),

            Row(

              children: <Widget>[

                Checkbox(

                  value: isChecked,

                  onChanged: (bool value) {

                    onChange(value);

                  },

                ),

                Text('Remember Me',

                    style: TextStyle(

                        fontSize: 16,

                        color: Color.fromRGBO(162, 199, 131, 0.5)))

              ],

            ),

            SizedBox(height: 10),

            MaterialButton(

              shape: RoundedRectangleBorder(

                  borderRadius: BorderRadius.circular(25)),

              minWidth: 200,

              height: 50,

              child: Text('LOGIN', style: TextStyle(fontSize: 14)),

              color: Color.fromRGBO(162, 199, 131, 0.5),

              textColor: Color.fromRGBO(41, 74, 13, 1),

              elevation: 15,

              onPressed: onPress,

            ),

            SizedBox(

              height: 20,

            ),

            GestureDetector(

                onTap: \_onRegister,

                child: Text('Register New Account',

                    style: TextStyle(fontSize: 16))),

            SizedBox(

              height: 10,

            ),

            GestureDetector(

                onTap: \_onForgot,

                child: Text('Forgot Account', style: TextStyle(fontSize: 16))),

          ],

        ),

      ),

    );

  }

  void onPress() {

    print(emcontroller.text);

    print(passcontroller.text);

  }

  void onChange(bool value) {

    setState(() {

      isChecked = value;

      savepref(value);

    });

  }

  void \_onRegister() {

    print('onRegister');

  }

  void \_onForgot() {

    print('Forgot');

  }

  Future savepref(bool value) async {

    print('Inside savepref');

    email = emcontroller.text;

    password = passcontroller.text;

    SharedPreferences prefs = await SharedPreferences.getInstance();

    if (value) {

      //true save pref

      if (\_isEmailValid(email) && (password.length > 5)) {

        await prefs.setString('email', email);

        await prefs.setString('pass', password);

        print('Save pref $email');

        print('Save pref $password');

        Toast.show("Preferences have been saved", context,

            duration: Toast.LENGTH\_SHORT, gravity: Toast.BOTTOM);

      } else {

        print('No email');

        setState(() {

          isChecked = false;

        });

        Toast.show("Check your credentials", context,

            duration: Toast.LENGTH\_SHORT, gravity: Toast.BOTTOM);

      }

    } else {

      await prefs.setString('email', '');

      await prefs.setString('pass', '');

      setState(() {

        emcontroller.text = '';

        passcontroller.text = '';

        isChecked = false;

      });

      print('Remove pref');

      Toast.show("Preferences have been removed", context,

          duration: Toast.LENGTH\_SHORT, gravity: Toast.BOTTOM);

    }

  }

  bool \_isEmailValid(String email) {

    return RegExp(r"^[a-zA-Z0-9.]+@[a-zA-Z0-9]+\.[a-zA-Z]+").hasMatch(email);

  }

}

1. widget\_test.dart

import 'package:flutter/material.dart';

import 'package:flutter\_test/flutter\_test.dart';

import 'package:my\_recycle\_agent/splashscreen.dart';

void main() {

  testWidgets('Counter increments smoke test', (WidgetTester tester) async {

    // Build our app and trigger a frame.

    await tester.pumpWidget(MySplash());

    // Verify that our counter starts at 0.

    expect(find.text('0'), findsOneWidget);

    expect(find.text('1'), findsNothing);

    // Tap the '+' icon and trigger a frame.

    await tester.tap(find.byIcon(Icons.add));

    await tester.pump();

    // Verify that our counter has incremented.

    expect(find.text('0'), findsNothing);

    expect(find.text('1'), findsOneWidget);

  });

}

1. pubspec.yaml

name: my\_recycle\_agent

description: A new Flutter project.

# The following defines the version and build number for your application.

# A version number is three numbers separated by dots, like 1.2.43

# followed by an optional build number separated by a +.

# Both the version and the builder number may be overridden in flutter

# build by specifying --build-name and --build-number, respectively.

# In Android, build-name is used as versionName while build-number used as versionCode.

# Read more about Android versioning at https://developer.android.com/studio/publish/versioning

# In iOS, build-name is used as CFBundleShortVersionString while build-number used as CFBundleVersion.

# Read more about iOS versioning at

# https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Articles/CoreFoundationKeys.html

version: 1.0.0+1

environment:

  sdk: ">=2.1.0 <3.0.0"

dependencies:

  flutter:

    sdk: flutter

  # The following adds the Cupertino Icons font to your application.

  # Use with the CupertinoIcons class for iOS style icons.

  cupertino\_icons: ^0.1.2

dev\_dependencies:

  flutter\_test:

    sdk: flutter

  shared\_preferences: ^0.5.3+4

  toast: ^0.1.5

  image\_picker:

# For information on the generic Dart part of this file, see the

# following page: https://dart.dev/tools/pub/pubspec

# The following section is specific to Flutter.

flutter:

  # The following line ensures that the Material Icons font is

  # included with your application, so that you can use the icons in

  # the material Icons class.

  uses-material-design: true

  # To add assets to your application, add an assets section, like this:

  assets:

    - assets/images/

  #  - images/a\_dot\_ham.jpeg

  # An image asset can refer to one or more resolution-specific "variants", see

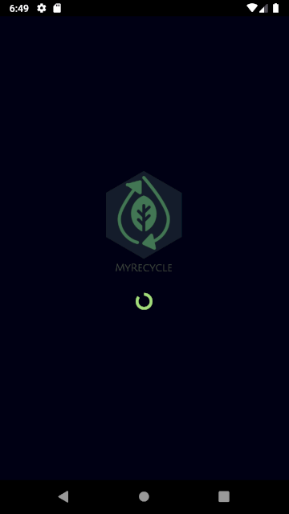
  # https://flutter.dev/assets-and-images/#resolution-aware.

  # For details regarding adding assets from package dependencies, see

  # <https://flutter.dev/assets-and-images/#from-packages>

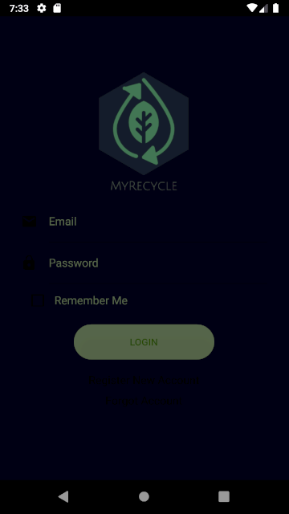
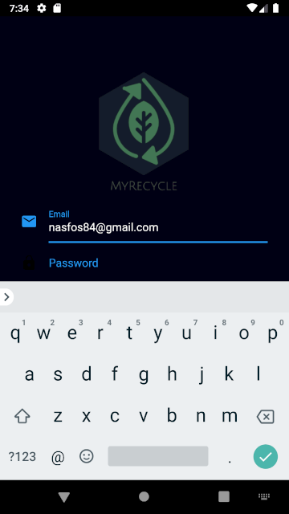
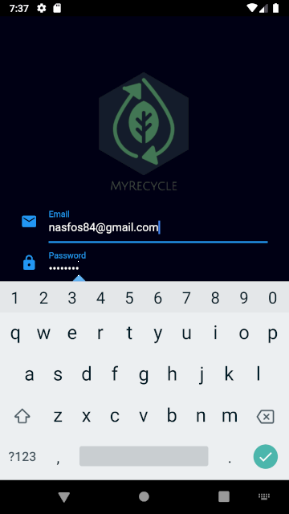
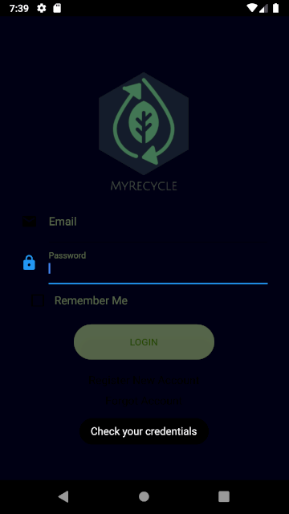
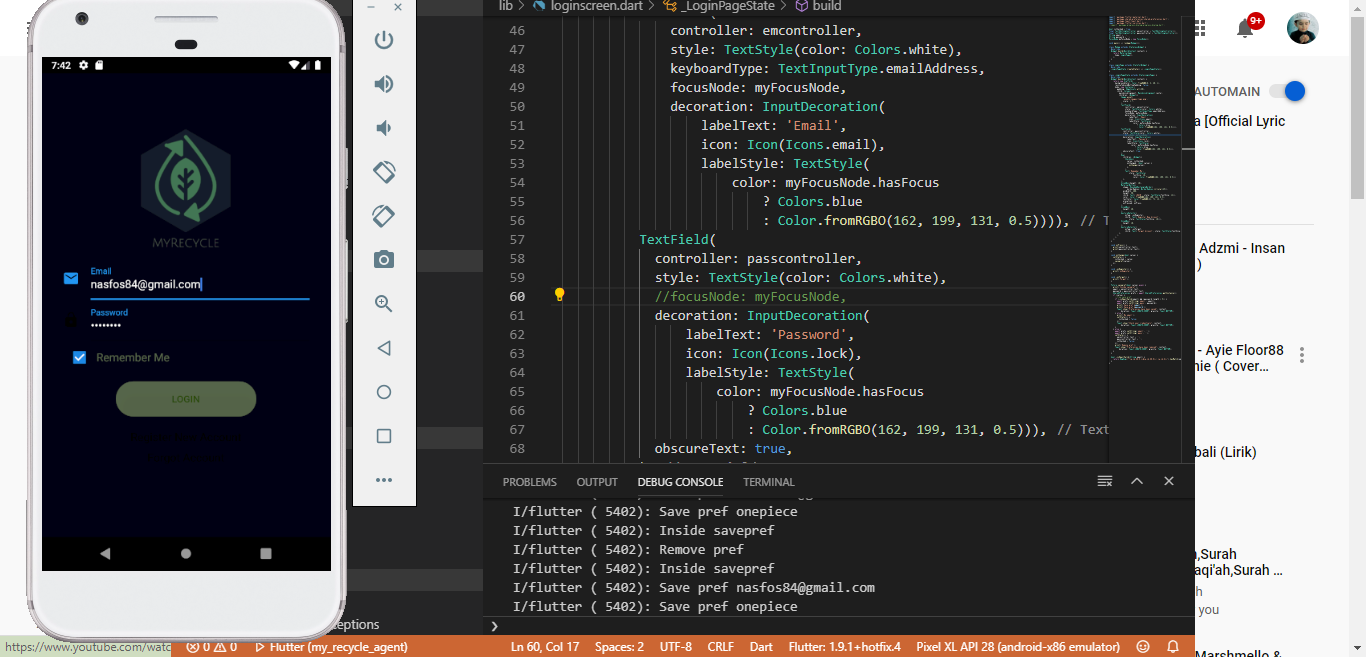
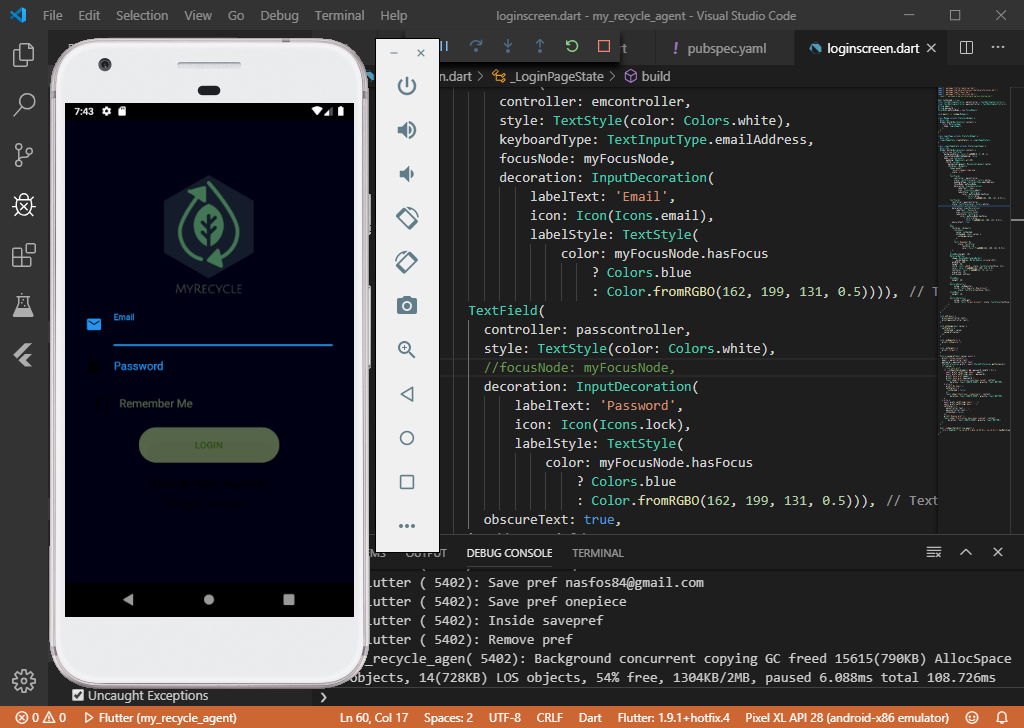
# Interface

1. Splash Screen



Splash screen display logo and circular progress indicator. Since this app is for recycling, the theme chosen are blue and green. Splash screen does not take a longer time to loading for the next page or screen. In this case, it will only take for 2 seconds for loading.

1. Login Screen

Login screen displays application’s logo, and require users to insert email and password. Users able to decide either to remain their email and password for each time they want to login. This is because not everyone remembered their password for application that rarely used. When users click a checkbox, it will display a message on a screen as they want the apps to remember their email and password. However, the screen still displays a message as users unclick a checkbox which means remove preferences.