Ziad Ashraf Ahmed Ahmed

19p7095

MileStone2

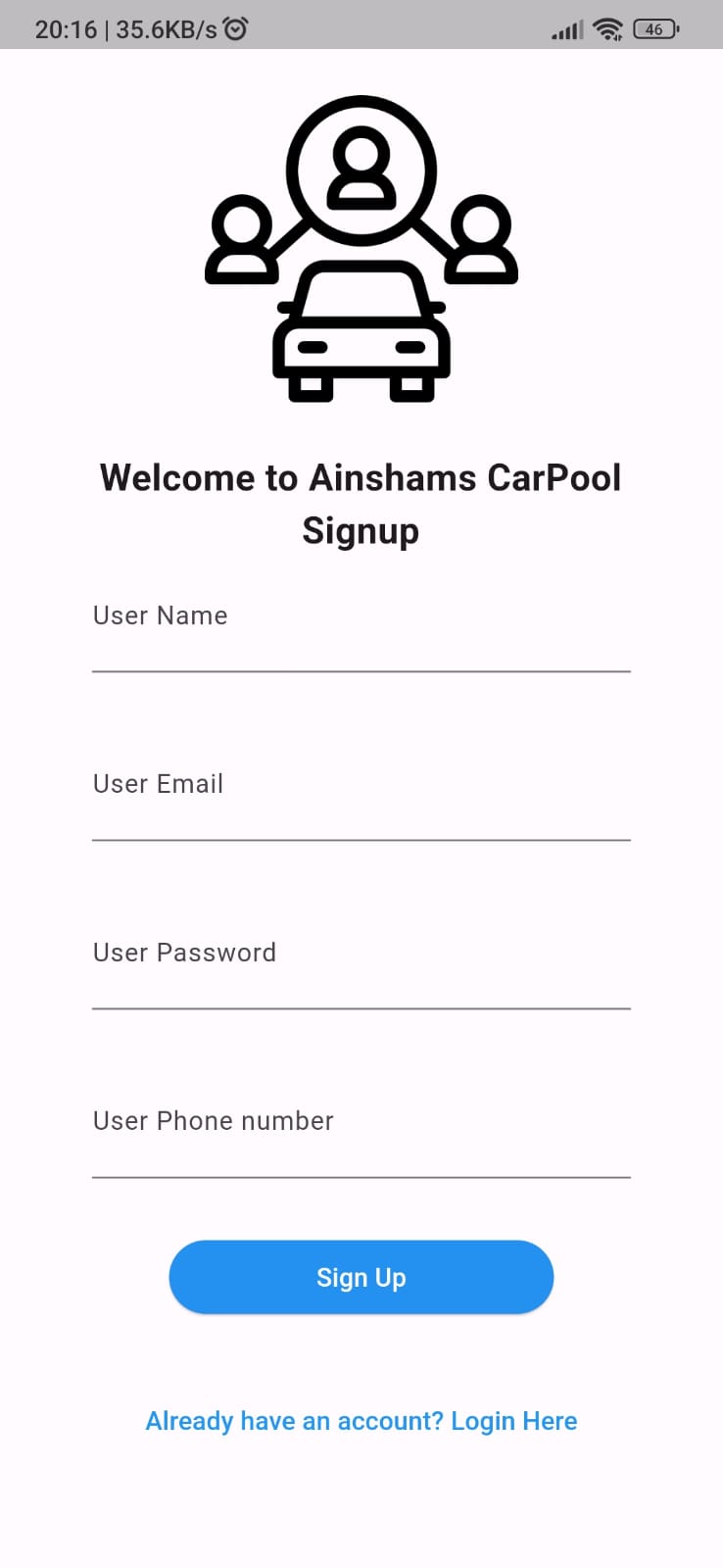
Milestone2

The second milestone is the basic function of the application of sign-up/login page, available routes. The Java/Kotlin/Dart code should be copied in a Word file, add screenshots of the layout, and list the testcase scenarios. Following that, convert the file into PDF and submit it on LMS.

sign-up

import 'package:firebase\_auth/firebase\_auth.dart';  
import 'package:firebase\_core/firebase\_core.dart';  
import 'package:firebase\_database/firebase\_database.dart';  
import 'package:flutter/material.dart';  
import 'package:project/home\_screen.dart';  
import 'package:project/login\_screen.dart';  
import 'package:project/methods/reusable\_methods.dart';  
  
class SignUpScreen extends StatefulWidget {  
 const SignUpScreen({super.key});  
  
 @override  
 State<SignUpScreen> createState() => \_SignUpScreenState();  
}  
  
class \_SignUpScreenState extends State<SignUpScreen> {  
 TextEditingController emailTextEditingController = TextEditingController();  
 TextEditingController nameTextEditingController = TextEditingController();  
 TextEditingController passwordTextEditingController = TextEditingController();  
 TextEditingController phoneTextEditingController = TextEditingController();  
 ReusableMethods rMethods = ReusableMethods();  
  
 checkIfNetworkIsAvailabe(){  
 rMethods.checkConnectivity(context);  
 signUpFormValidation();  
 }  
  
 signUpFormValidation(){  
 if(nameTextEditingController.text.trim().length<3){  
 rMethods.displaySnakBar("Name Must Be Atleast 4 charachters", context);  
 }  
 else if(!emailTextEditingController.text.endsWith("@eng.asu.edu.eg")){// try to find method to check last few digits  
 rMethods.displaySnakBar("Please SignUp with ASU Domain Email", context);  
 }  
 else if(passwordTextEditingController.text.trim().length<6){  
 rMethods.displaySnakBar("Password Must Be Atleast 6 Charachters", context);  
 }  
 else if(phoneTextEditingController.text.trim().length<11){  
 rMethods.displaySnakBar("Phone Number Must Be Atleast 11 Digets", context);  
 }  
 else{  
 registerNewUser();  
 }  
  
 }  
  
 registerNewUser()async{  
 final User? userFirebase = (  
 await FirebaseAuth.*instance*.createUserWithEmailAndPassword(  
 email: emailTextEditingController.text.trim(),  
 password: passwordTextEditingController.text.trim(),  
 ).catchError((errorMsg){  
 rMethods.displaySnakBar(errorMsg.toString(), context);  
 })  
 ).user;  
 if(!context.mounted)return;  
 DatabaseReference usersRef = FirebaseDatabase.*instance*.ref().child("users").child(userFirebase!.uid);  
 Map userDataMap = {  
 "name":nameTextEditingController.text.trim(),  
 "email":emailTextEditingController.text.trim(),  
 "phone":phoneTextEditingController.text.trim(),  
 "id":userFirebase.uid,  
 "blockStatus":"no",  
 };  
 usersRef.set(userDataMap);  
  
 Navigator.*pushReplacement*(context,MaterialPageRoute(builder: (c)=>MyScreen()));  
  
 }  
   
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 body: SingleChildScrollView(  
 child: Padding(  
 padding: const EdgeInsets.all(50),  
 child: Center(  
 child: Column(  
 children: [  
 Image.asset(  
 "assets/images/car-sharing.png",  
 width: 170,  
 ),  
 SizedBox(height: 25,),  
  
  
 Text(  
 "Welcome to Ainshams CarPool",  
 style: TextStyle(  
 fontSize: 20,  
 fontWeight: FontWeight.*bold*,  
 ),  
 ),  
 Text(  
 "Signup",  
 style: TextStyle(  
 fontSize: 20,  
 fontWeight: FontWeight.*bold*,  
 ),  
 ),  
  
 //NameTextField  
 TextField(  
 controller: nameTextEditingController,  
 keyboardType: TextInputType.*name*,  
 decoration: const InputDecoration(  
 labelText: "User Name",  
 labelStyle: TextStyle(  
 fontSize: 14,  
 )  
 ),  
 style: const TextStyle(  
 color: Colors.*blue*,  
 fontSize: 15,  
 ),  
 ),  
 SizedBox(height: 30,),  
 //emailTextField  
 TextField(  
 controller: emailTextEditingController,  
 keyboardType: TextInputType.*emailAddress*,  
 decoration: const InputDecoration(  
 labelText: "User Email",  
 labelStyle: TextStyle(  
 fontSize: 14,  
 )  
 ),  
 style: const TextStyle(  
 color: Colors.*blue*,  
 fontSize: 15,  
 ),  
 ),  
 SizedBox(height: 30,),  
 //passTextField  
 TextField (  
 controller: passwordTextEditingController,  
 keyboardType: TextInputType.*text*,  
 obscureText: true, // to hide password  
 decoration: const InputDecoration(  
 labelText: "User Password",  
 labelStyle: TextStyle(  
 fontSize: 14,  
 )  
 ),  
 style: const TextStyle(  
 color: Colors.*blue*,  
 fontSize: 15,  
 ),  
 ),  
 SizedBox(height: 30,),  
 //phone number  
 TextField (  
 controller: phoneTextEditingController,  
 keyboardType: TextInputType.*phone*,  
 obscureText: true, // to hide password  
 decoration: const InputDecoration(  
 labelText: "User Phone number",  
 labelStyle: TextStyle(  
 fontSize: 14,  
 )  
 ),  
 style: const TextStyle(  
 color: Colors.*blue*,  
 fontSize: 15,  
 ),  
 ),  
 SizedBox(height: 30,),  
 //signup button  
  
 ElevatedButton(  
  
 child: const Text(  
 "Sign Up",  
 style: TextStyle(color: Colors.*white*,),  
 ),  
 onPressed:(){  
 checkIfNetworkIsAvailabe();  
  
 },  
 style: ElevatedButton.*styleFrom*(  
 backgroundColor: Colors.*blue*,  
 padding: EdgeInsets.symmetric(horizontal: 80,)  
 ),  
 ),  
 //having an account navigate to login Screen  
 SizedBox(height: 30,),  
 TextButton(  
 onPressed: (){  
 Navigator.*pushReplacement*(context,MaterialPageRoute(builder: (c)=>LoginScreen()));  
 },  
 child: Text(  
 "Already have an account? Login Here",  
 style: TextStyle(  
 color: Colors.*blue*,  
 ),  
 ),  
 ),  
 ],  
 ),  
 ),  
 ),  
 ),  
   
 );  
 }  
}

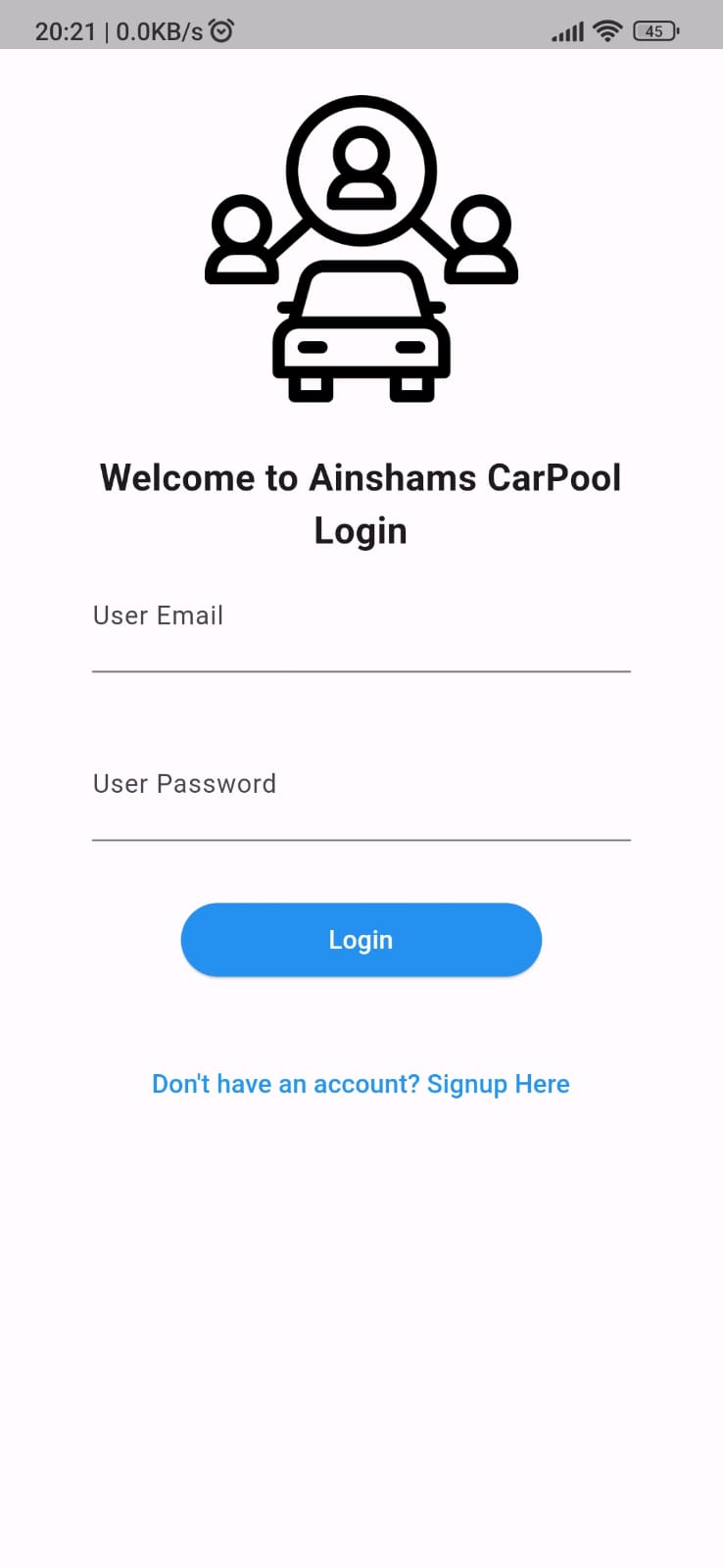
Screenshot



Login

import 'package:firebase\_auth/firebase\_auth.dart';  
import 'package:firebase\_database/firebase\_database.dart';  
import 'package:flutter/material.dart';  
import 'package:project/home\_screen.dart';  
import 'package:project/signup\_screen.dart';  
  
import 'methods/reusable\_methods.dart';  
class LoginScreen extends StatefulWidget {  
 const LoginScreen({super.key});  
  
 @override  
 State<LoginScreen> createState() => \_LoginScreenState();  
}  
  
class \_LoginScreenState extends State<LoginScreen> {  
 TextEditingController emailTextEditingController = TextEditingController();  
 TextEditingController passwordTextEditingController = TextEditingController();  
 ReusableMethods rMethods = ReusableMethods();  
  
 checkIfNetworkIsAvailabe(){  
 rMethods.checkConnectivity(context);  
 logInFormValidation();  
 }  
  
 logInFormValidation(){  
 if(!emailTextEditingController.text.endsWith("@eng.asu.edu.eg")){// try to find method to check last few digits  
 rMethods.displaySnakBar("Please SignUp with ASU Domain Email", context);  
 }  
 else if(passwordTextEditingController.text.trim().length<6){  
 rMethods.displaySnakBar("Password Must Be Atleast 6 Charachters", context);  
 }  
 else{  
 if(emailTextEditingController.text == "test@eng.asu.edu.eg"){// used to bypass the authentication --for the aid of testing  
 Navigator.*pushReplacementNamed*(context,'/home\_screen');  
 }  
 else  
 LogInUser();  
 }  
  
 }  
  
 LogInUser()async{  
  
 final User? userFirebase = (  
 await FirebaseAuth.*instance*.signInWithEmailAndPassword(  
 email: emailTextEditingController.text.trim(),  
 password: passwordTextEditingController.text.trim(),  
 ).catchError((errorMsg){  
 rMethods.displaySnakBar(errorMsg.toString(), context);  
 })  
 ).user;  
  
 if(!context.mounted)return;  
  
 if(userFirebase != null){  
 DatabaseReference usersRef = FirebaseDatabase.*instance*.ref().child("users").child(userFirebase!.uid);  
 usersRef.once().then((snap){  
 if(snap.snapshot.value != null){  
  
 if((snap.snapshot.value as Map)["blockStatus"] == "no"){  
 Navigator.*pushReplacementNamed*(context,'/home\_screen');  
 }  
 else{  
 FirebaseAuth.*instance*.signOut();  
 rMethods.displaySnakBar("This Account Is Blocked", context);  
 }  
  
 }else{  
 FirebaseAuth.*instance*.signOut();  
 rMethods.displaySnakBar("The Account Not Found As User", context);  
  
 }  
  
  
 });  
 }  
  
 }  
  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 body: SingleChildScrollView(  
 child: Padding(  
 padding: const EdgeInsets.all(50),  
 child: Center(  
 child: Column(  
 children: [  
 Image.asset(  
 "assets/images/car-sharing.png",  
 width: 170,  
 ),  
 SizedBox(height: 25,),  
  
  
 Text(  
 "Welcome to Ainshams CarPool",  
 style: TextStyle(  
 fontSize: 20,  
 fontWeight: FontWeight.*bold*,  
 ),  
 ),  
 Text(  
 "Login",  
 style: TextStyle(  
 fontSize: 20,  
 fontWeight: FontWeight.*bold*,  
 ),  
 ),  
  
  
 //emailTextField  
 TextField(  
 controller: emailTextEditingController,  
 keyboardType: TextInputType.*emailAddress*,  
 decoration: const InputDecoration(  
 labelText: "User Email",  
 labelStyle: TextStyle(  
 fontSize: 14,  
 )  
 ),  
 style: const TextStyle(  
 color: Colors.*blue*,  
 fontSize: 15,  
 ),  
 ),  
 SizedBox(height: 30,),  
 //passTextField  
 TextField (  
 controller: passwordTextEditingController,  
 keyboardType: TextInputType.*text*,  
 obscureText: true, // to hide password  
 decoration: const InputDecoration(  
 labelText: "User Password",  
 labelStyle: TextStyle(  
 fontSize: 14,  
 )  
 ),  
 style: const TextStyle(  
 color: Colors.*blue*,  
 fontSize: 15,  
 ),  
 ),  
 SizedBox(height: 30,),  
 //Login button  
 ElevatedButton(  
 child: const Text(  
 "Login",  
 style: TextStyle(color: Colors.*white*,),  
 ),  
 onPressed:(){  
 checkIfNetworkIsAvailabe();  
 },  
 style: ElevatedButton.*styleFrom*(  
 backgroundColor: Colors.*blue*,  
 padding: EdgeInsets.symmetric(horizontal: 80,)  
 ),  
 ),  
 //having an account navigate to login Screen  
 SizedBox(height: 30,),  
 TextButton(  
 onPressed: (){  
 Navigator.*pushReplacement*(context,MaterialPageRoute(builder: (c)=>SignUpScreen()));  
 },  
 child: Text(  
 "Don\'t have an account? Signup Here",  
 style: TextStyle(  
 color: Colors.*blue*,  
 ),  
 ),  
 ),  
 ],  
 ),  
 ),  
 ),  
 ),  
  
  
  
  
 );;  
 }  
}

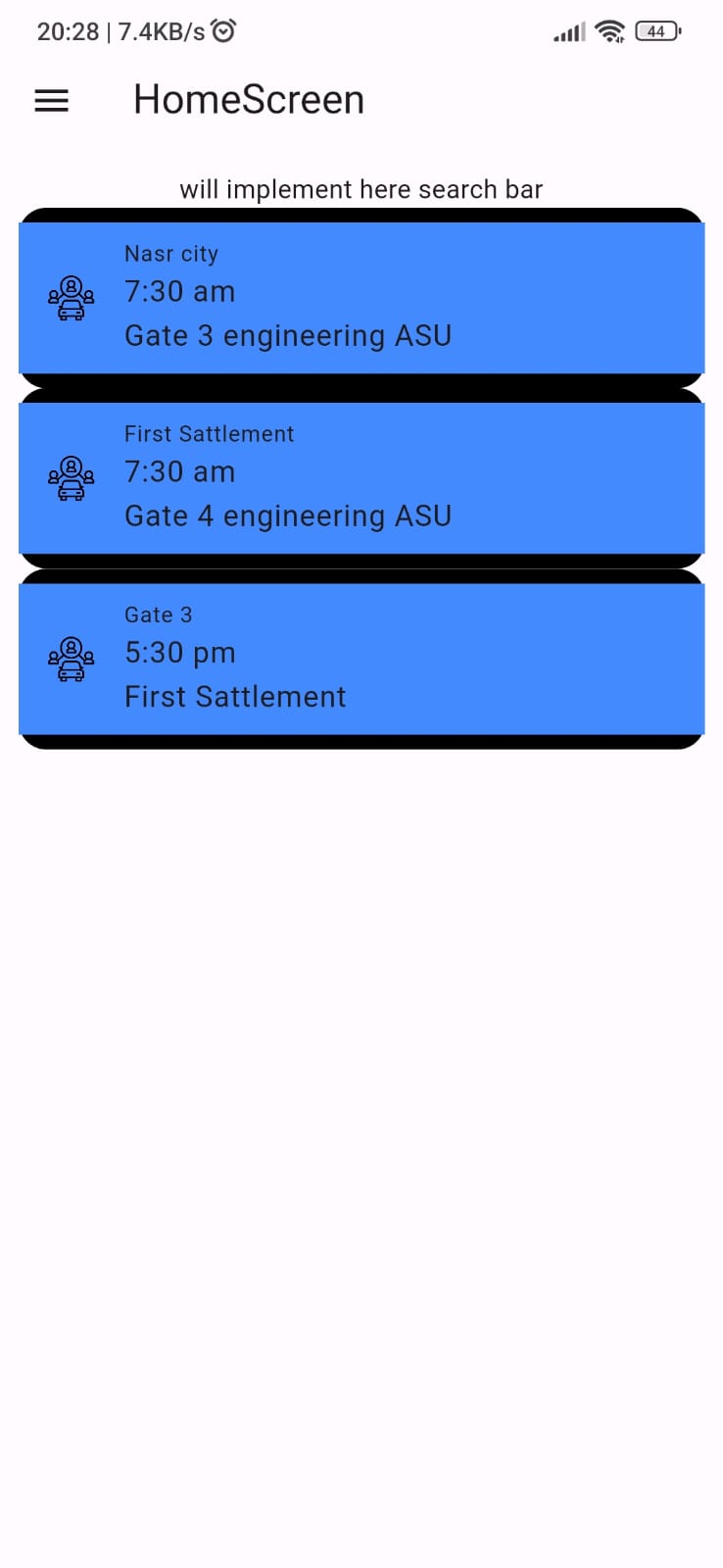
Screenshot



Home\_screen / User App

import 'package:firebase\_auth/firebase\_auth.dart';  
import 'package:flutter/material.dart';  
import 'package:firebase\_database/firebase\_database.dart';  
void main() {  
 runApp(MyScreen());  
}  
  
class MyScreen extends StatefulWidget {  
 @override  
 State<MyScreen> createState() => \_MyScreenState();  
}  
  
class \_MyScreenState extends State<MyScreen> {  
 late DatabaseReference \_databaseReference;  
 List<Map<String, String>> mapRoutes = [];  
  
 TextEditingController fromController = TextEditingController();  
 TextEditingController toController = TextEditingController();  
 TextEditingController timeController = TextEditingController();  
  
 void initState() {  
 super.initState();  
 \_databaseReference = FirebaseDatabase.*instance*.reference().child("routes");  
 \_setupDataListener();  
 }  
  
 void \_setupDataListener() {  
 \_databaseReference.onValue.listen((event) {  
 if (event.snapshot.value != null) {  
 print("Retrieved Data: ${event.snapshot.value}");  
  
 // The retrieved data is a map, convert entries to a list  
 var data = (event.snapshot.value as Map<dynamic, dynamic>).entries;  
  
 if (data != null) {  
 setState(() {  
 mapRoutes = data  
 .map((entry) => Map<String, String>.from({  
 'From': '${entry.value['From']}',  
 'To': '${entry.value['To']}',  
 'Time': entry.value['Time'],  
 }))  
 .toList();  
 });  
 }  
 } else {  
 print("Snapshot value is null");  
 }  
 });  
 }  
  
  
  
  
  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('HomeScreen'),  
 ),  
 drawer: \_buildSideDrawer(context),  
 body: Container(  
 padding: EdgeInsets.all(10),  
 child: Column(  
 children: [  
 Text("will implement here search bar"),  
 Expanded(  
 child: ListView.builder(  
 itemCount: mapRoutes.length,  
 itemBuilder: (context, index) {  
 return Container(  
 decoration: BoxDecoration(  
 color: Colors.*black*,  
 borderRadius: BorderRadius.circular(15),  
 ),  
 padding: const EdgeInsets.symmetric(vertical: 8.0),  
 child: Container(  
 color: Colors.*blueAccent*,  
 child: ListTile(  
 title: Column(  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: [  
 Text("${mapRoutes[index]['From']}",style: TextStyle(fontSize: 12),),  
 Text("${mapRoutes[index]['Time']}"),  
 Text("${mapRoutes[index]['To']}"),  
 ],  
 ),  
 leading: Image.asset(  
 "assets/images/car-sharing.png",  
 height: 25,  
 ),  
 onTap: () {  
 Navigator.*pushNamed*(context, "/Cart\_screen");  
 },  
 ),  
 ),  
 );  
 },  
 ),  
 ),  
 ],  
 ),  
 ),  
 );  
 }  
  
 Drawer \_buildSideDrawer(BuildContext context) {  
 return Drawer(  
 child: ListView(  
 padding: EdgeInsets.*zero*,  
 children: [  
 DrawerHeader(  
 decoration: BoxDecoration(  
 color: Colors.*blue*,  
 ),  
 child: Text(  
 'Menu',  
 style: TextStyle(  
 color: Colors.*white*,  
 fontSize: 24,  
 ),  
 ),  
 ),  
 \_buildDrawerItem(context, 'Profile Page', '/Profile\_screen'),  
 \_buildDrawerItem(context, 'Orders History', '/OrderHistory\_screen'),  
 \_buildDrawerItem(context, 'Upcoming Rides', '/UpcommingRides\_screen'),  
 \_buildDrawerItem(context, 'Logout', '/Login\_screen'),  
 ],  
 ),  
 );  
 }  
  
 ListTile \_buildDrawerItem(BuildContext context, String title, String route) {  
 return ListTile(  
 title: Text(title),  
 onTap: () {  
 if (title != 'Logout')  
 {  
 Navigator.*pop*(context); // Close the drawer  
 Navigator.*pushNamed*(context, route);  
 }  
 else{  
 Navigator.*pop*(context); // Close the drawer  
 FirebaseAuth.*instance*.signOut();  
 Navigator.*pushReplacementNamed*(context, route);  
 }  
  
 },  
 );  
 }  
}

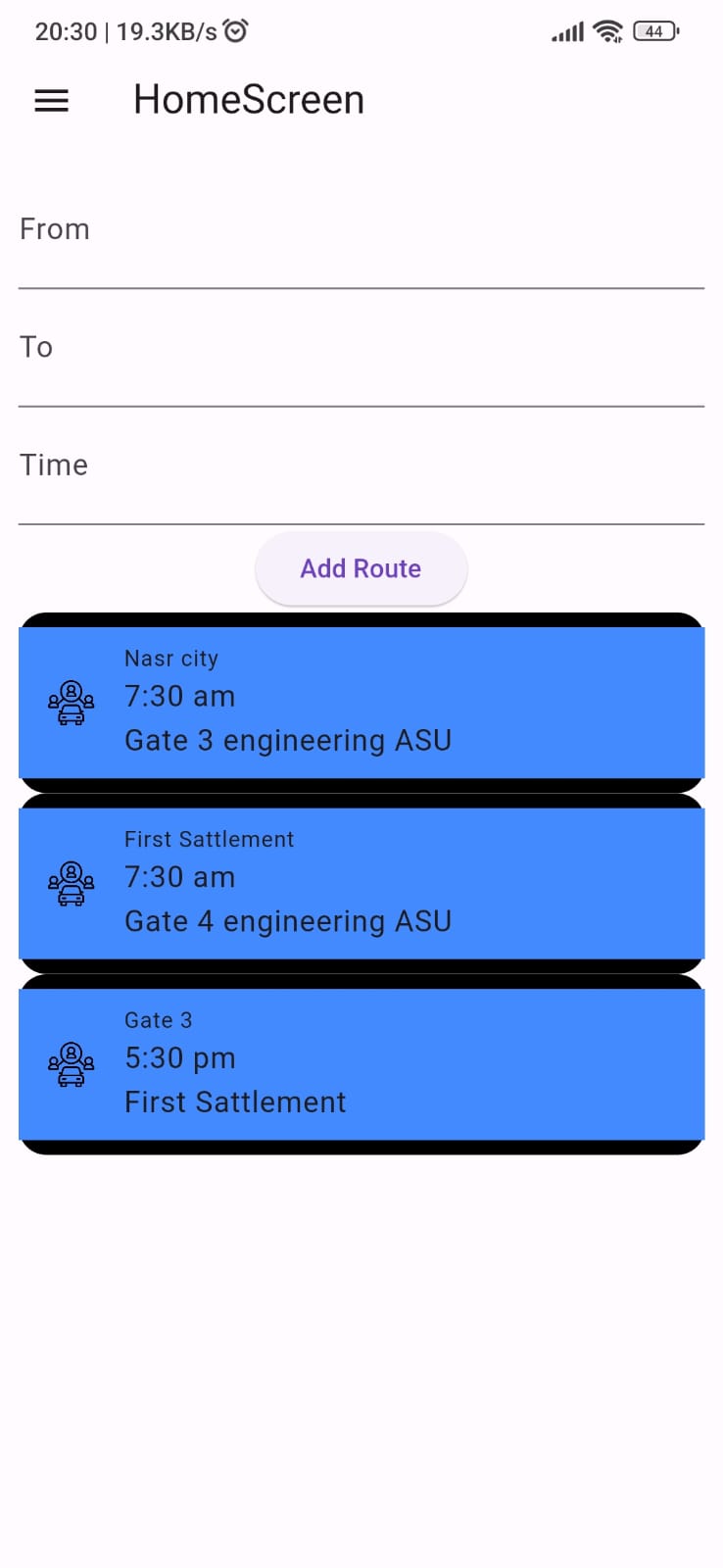
Screen Shot



Home\_screen / Driver

import 'package:flutter/material.dart';  
import 'package:firebase\_database/firebase\_database.dart';  
void main() {  
 runApp(MyScreen());  
}  
  
class MyScreen extends StatefulWidget {  
 @override  
 State<MyScreen> createState() => \_MyScreenState();  
}  
  
class \_MyScreenState extends State<MyScreen> {  
 late DatabaseReference \_databaseReference;  
 List<Map<String, String>> mapRoutes = [];  
  
 TextEditingController fromController = TextEditingController();  
 TextEditingController toController = TextEditingController();  
 TextEditingController timeController = TextEditingController();  
  
 void initState() {  
 super.initState();  
 \_databaseReference = FirebaseDatabase.instance.reference().child("routes");  
 \_setupDataListener();  
 }  
  
 void \_setupDataListener() {  
 \_databaseReference.onValue.listen((event) {  
 if (event.snapshot.value != null) {  
 print("Retrieved Data: ${event.snapshot.value}");  
  
 // The retrieved data is a map, convert entries to a list  
 var data = (event.snapshot.value as Map<dynamic, dynamic>)?.entries;  
  
 if (data != null) {  
 setState(() {  
 mapRoutes = data  
 .map((entry) => Map<String, String>.from({  
 'From': '${entry.value['From']}',  
 'To': '${entry.value['To']}',  
 'Time': entry.value['Time'],  
 }))  
 .toList();  
 });  
 }  
 } else {  
 print("Snapshot value is null");  
 }  
 });  
 }  
  
  
 void \_addRoute() {  
 String from = fromController.text.trim();  
 String to = toController.text.trim();  
 String time = timeController.text.trim();  
  
 if (from.isNotEmpty && to.isNotEmpty && time.isNotEmpty) {  
 \_databaseReference.push().set({  
 'From': '$from',  
 'To': '$to',  
 'Time': time,  
 });  
  
 // Clear the text controllers after adding a route  
 fromController.clear();  
 toController.clear();  
 timeController.clear();  
 } else {  
 print("One or more fields are empty");  
 }  
 }  
  
  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('HomeScreen'),  
 ),  
 drawer: \_buildSideDrawer(context),  
 body: Container(  
 padding: EdgeInsets.all(10),  
 child: Column(  
 children: [  
 TextField(  
 controller: fromController,  
 decoration: InputDecoration(labelText: 'From'),  
 ),  
 TextField(  
 controller: toController,  
 decoration: InputDecoration(labelText: 'To'),  
 ),  
 TextField(  
 controller: timeController,  
 decoration: InputDecoration(labelText: 'Time'),  
 ),  
 ElevatedButton(  
 onPressed: () {  
 \_addRoute();  
 // Adding print statement to check mapRoutes after adding a route  
 print("Updated mapRoutes: $mapRoutes");  
 },  
 child: Text('Add Route'),  
 ),  
 Expanded(  
 child: ListView.builder(  
 itemCount: mapRoutes.length,  
 itemBuilder: (context, index) {  
 return Container(  
 decoration: BoxDecoration(  
 color: Colors.black,  
 borderRadius: BorderRadius.circular(15),  
 ),  
 padding: const EdgeInsets.symmetric(vertical: 8.0),  
 child: Container(  
 color: Colors.blueAccent,  
 child: ListTile(  
 title: Column(  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: [  
 Text("${mapRoutes[index]['From']}",style: TextStyle(fontSize: 12),),  
 Text("${mapRoutes[index]['Time']}"),  
 Text("${mapRoutes[index]['To']}"),  
 ],  
 ),  
 leading: Image.asset(  
 "assets/images/car-sharing.png",  
 height: 25,  
 ),  
 onTap: () {  
 Navigator.pushNamed(context, "/Cart\_screen");  
 },  
 ),  
 ),  
 );  
 },  
 ),  
 ),  
 ],  
 ),  
 ),  
 );  
 }  
  
 Drawer \_buildSideDrawer(BuildContext context) {  
 return Drawer(  
 child: ListView(  
 padding: EdgeInsets.zero,  
 children: [  
 DrawerHeader(  
 decoration: BoxDecoration(  
 color: Colors.blue,  
 ),  
 child: Text(  
 'Menu',  
 style: TextStyle(  
 color: Colors.white,  
 fontSize: 24,  
 ),  
 ),  
 ),  
 \_buildDrawerItem(context, 'Profile Page', '/Profile\_screen'),  
 \_buildDrawerItem(context, 'Orders History', '/OrderHistory\_screen'),  
 \_buildDrawerItem(context, 'Upcoming Rides', '/UpcommingRides\_screen'),  
 \_buildDrawerItem(context, 'Logout', '/Login\_screen'),  
 ],  
 ),  
 );  
 }  
  
 ListTile \_buildDrawerItem(BuildContext context, String title, String route) {  
 return ListTile(  
 title: Text(title),  
 onTap: () {  
 Navigator.pop(context); // Close the drawer  
 Navigator.pushNamed(context, route);  
 },  
 );  
 }  
}

Screen shot.



Firebase Screenshot

A screenshot of a computer

Description automatically generated

Testcases scenarios

**User Authentication (Firebase Authentication):**

1. **User Registration:**
   * Input valid registration details (email with collage domain, password, username, phonenumber).
   * Ensure the user account is created successfully.
   * Verify the user data in the Firebase Authentication console.
2. **User Successfully Login:**
   * Provide valid login credentials.
   * Verify that the user is logged in successfully.
   * Check for appropriate error handling when providing invalid credentials.
3. **Logout:**
   * Log out the user and ensure they are redirected to the login screen.
   * Verify that the user cannot access protected routes after logging out.

**User Home Screen:**

1. **Route Retrieval:**
   * Ensure that the available routes are fetched from the database and displayed on the home screen.

**Driver's Home Screen and Data Submission:**

1. **Driver Home Screen Trip Data Submission:**
   * Submit the data and ensure it is successfully stored in the Firebase real-time database.
   * Check for appropriate error handling for invalid or incomplete data submissions.
   * Check that the submitted data is fetched correctly from the user application

**Interaction Between User and Driver:**

1. **Real-time Data Update:**
   * After a driver submits a new trip, ensure that the user's home screen is automatically updated with the new route information without needing a manual refresh.

**Error Handling:**

1. **Invalid Input Handling:**
   * Test the application with invalid inputs during registration, login, and data submission.
   * Ensure appropriate error messages are displayed.
2. **Network Issues:**
   * Simulate network issues and check if the application handles them gracefully, providing informative error messages.

**Navigation:**

1. **Navigation Between Screens:**
   * Verify that users can navigate seamlessly between all screens

Note:

For the Aid of testing, we have email: [test@eng.asu.edu.eg](mailto:test@eng.asu.edu.eg) and password 123456789

This account don’t go through the firebase authentication process in order to decrease the test cases dependencies in case something went wrong.