Sri Lanka Institute of Information Technology

“Wordly”: Word Guessing App

Report

IT19121352

Kudarachchi K.A.N.D.

2022_REG_08

Current Trends in Software Engineering – SE4010

B.Sc. (Hons’) in Information Technology

Specializing in Software Engineering

Table of Contents

List of Figures

1. Project Description

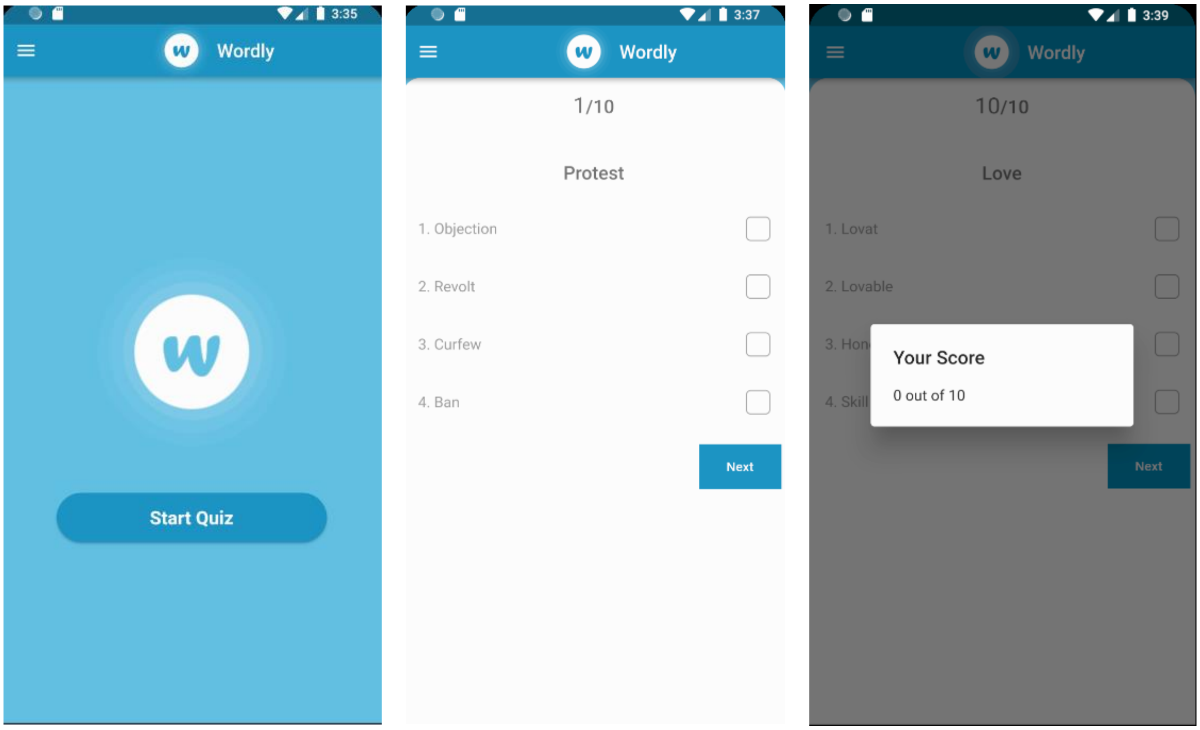
“Wordly” is a quiz-based word guessing game developed as a mobile app. The app was developed using Flutter and Firebase.

2. Implemented Functionalities

All the code related to the main functionalities implemented by Kudarachchi K.A.N.D (IT19121352) is available in the Appendices section of the report, or the complete project is available at https://github.com/rukshan99/wordly .

2.1 Display definitions to the user and calculate score

After successfully logged in as a user, user redirects to Home screen. When clicking start quiz user navigate to the quiz screen. 10 random definition questions are getting from “definitions” collection and display them to the user as in *figure 2.1*. The answers can be have multiple answers too. The points are calculated according to the checked correct answers. When calculating score, the correct answers according to a definition also getting from the “definitions” collection. The score is calculating as, if an answer is correct 1 point for a definition.



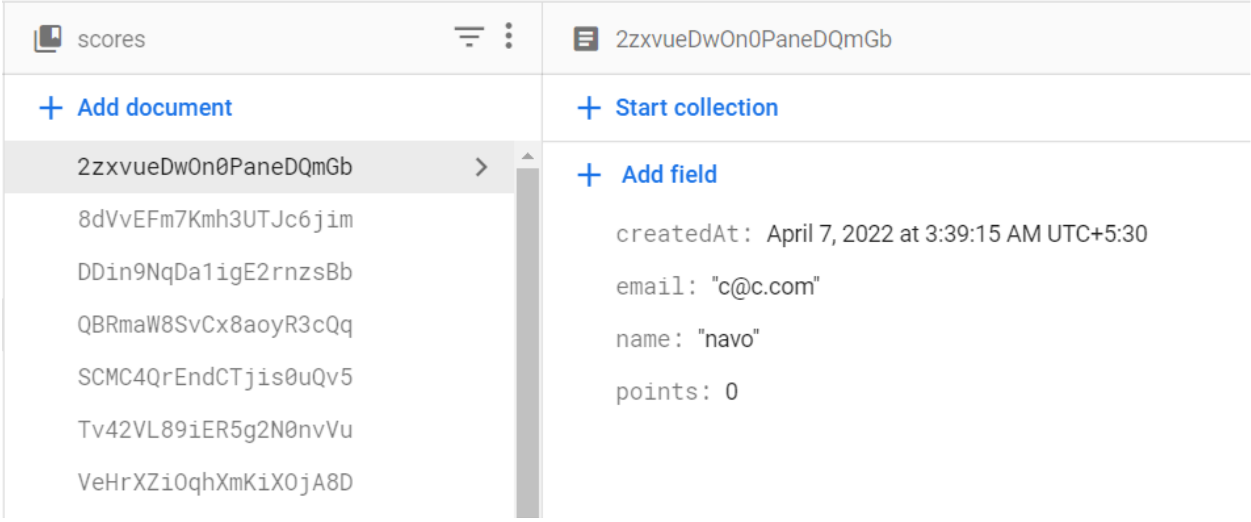
Your Score

0 out of 10

Figure 2.1 Home screen, Quiz screen and Score Pop Up

2.2 Add score details to the “scores” collection in Cloud Firestore

This is a “create” operation. The data is saved to this collection when a user is successfully faced to a quiz. As illustrated in *figure 2.2* a user document consists of email (String), name (String), created At (Date Time), and points (Number) fields. The field “points” was used to store the number of points earned by a user when attempting the quizzes. The default or initial value of this field is set to be 0. The field “email” has a unique value for each user, which is ensured by the Firebase Authentication service in the user registration process.



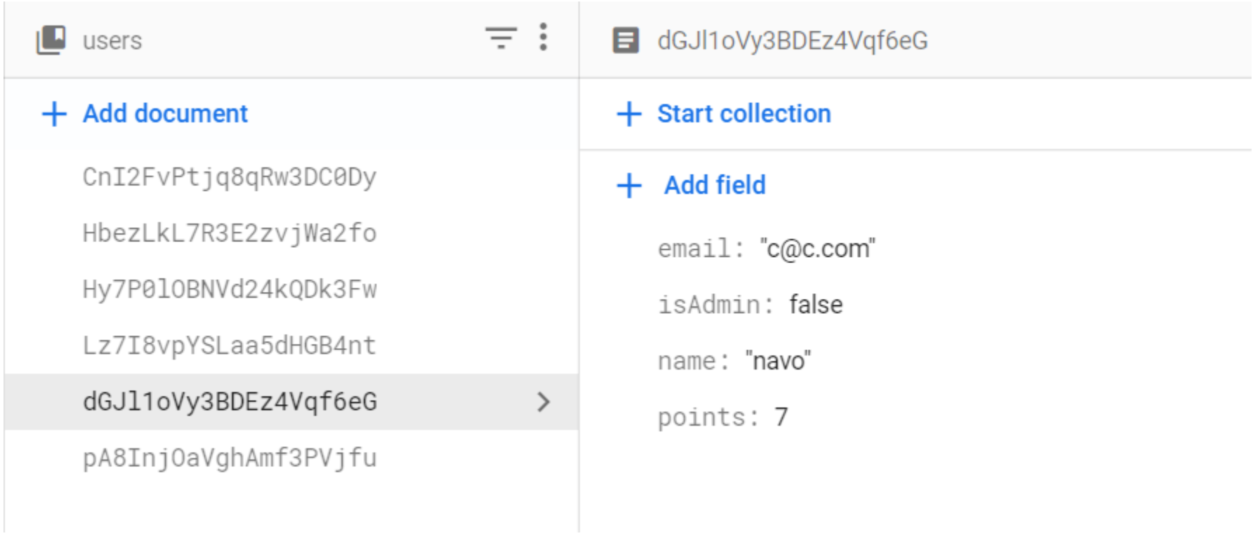
2zxvueDwOn0PaneDQmGb
8dVvEFm7Kmh3UTJc6jim
DDin9NqDa1igE2rnzsBb
QBRmaW8SvCx8aoyR3cQq
SCMC4QrEndCTjis0uQv5
Tv42VL89iER5g2N0nvVu
VeHrXZiOqhXmKiX0jA8D

createdAt: April 7, 2022 at 3:39:15 AM UTC+5:30
email: "c@c.com"
name: "navo"
points: 0

Figure 2.2 Score document structure

2.3 Update points earned by users according to quizzes in “users” collection

This is a “update” operation. When users attempting to quizzes, the number of points they have earned are updated in this “users” collection. When updating points, new points are added to the previous points. When a user doing more quizzes, they can earn more points. The field “points” was used to handle the rankings of users which is discussed later in this report and the initial value is 0.



dGJl1oVy3BDEz4Vqf6eG

CnI2FvPtjq8qRw3DC0Dy
HbezLkL7R3E2zvjWa2fo
Hy7P0lOBNVd24kQDk3Fw
Lz7I8vpYSLaa5dHGB4nt
dGJl1oVy3BDEz4Vqf6eG
pA8InjOaVghAmf3PVjfu

email: "c@c.com"
isAdmin: false
name: "navo"
points: 7

Figure 2.3 User data

3. References

- Flutter documentation: https://docs.flutter.dev/
- Firebase documentation: https://firebase.flutter.dev/
- Background animations: https://felixblaschke.medium.com/fancy-background-animations-in-flutter-4163d50f5c37

4. Appendices

5.1 Models/score.dart

```
// @dart=2.9
import 'package:cloud_firestore/cloud_firestore.dart';

class Score {
  final String name;
  final String email;
  final int points;
  final DateTime createdAt;

  Score({
    this.name,
    this.email,
    this.points,
    this.createdAt,
  });

  factory Score.fromJson(Map<String, dynamic> json, DocumentReference docRef) {
    // ignore: avoid_print
    print(json['score']);
    return Score(
      name: json['name'] as String,
      email: json['email'] as String,
      points: json['points'] as int,
      createdAt: json['createdAt'] as DateTime,
    );
  }

  Map<String, dynamic> toMap() {
    return {
      "name": name,
      "email": email,
      "points": points,
      "createdAt": createdAt,

    };
  }
}
```

5.2 Controllers/score_controller.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:wordly/models/score.dart';
import 'package:wordly/models/definition.dart';

class ScoreController {
  final FirebaseFirestore _db = FirebaseFirestore.instance;

  // Create a CollectionReference that references the firestore collection
  CollectionReference collectionRef =
      FirebaseFirestore.instance.collection('scores');

  CollectionReference usersRef = FirebaseFirestore.instance.collection('users');

  // get all quesions
  Stream<List<Definition>> getQuesions() {
    return _db.collection('definitions').snapshots().map((snapshot) => snapshot
        .docs
        .map((doc) => Definition.fromJson(doc.data(), doc.reference))
        .toList());
  }

  // Get all scores
  Stream<List<Score>> getScores() {
    return _db
        .collection('scores')
        .orderBy('points', descending: true)
        .snapshots()
        .map((snapshot) => snapshot.docs
            .map((doc) => Score.fromJson(doc.data(), doc.reference))
            .toList());
  }

  // Add a Score
  addScore(Score userObj) async {
    try {
      _db.runTransaction((Transaction transaction) async {
        await _db.collection('scores').doc().set(userObj.toMap());
```

```
   });
  } catch (e) {
   // ignore: avoid_print
   print(e.toString());
  }
 }

 //update score
 updateScore(Score userObj) async {
  try {
   _db.runTransaction((Transaction transaction) async {
    var snapshots =
      await usersRef.where('email', isEqualTo: userObj.email).get();

    snapshots.docs.forEach((DocumentSnapshot doc) {
     doc.reference.update({'points': FieldValue.increment(userObj.points)});
    });

    // await _db
    //     .collection('users')
    //     .doc(doc)
    //     .update({'points': userObj.points});
   });
  } catch (e) {
   // ignore: avoid_print
   print(e.toString());
  }
 }
}
```

5.3 Widgets/user_drawer.dart

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:wordly/models/review.dart';
import 'package:wordly/screens/admin_users.dart';
import 'package:wordly/screens/home.dart';
import 'package:wordly/screens/userLeaderboard.dart';
import 'package:wordly/screens/user_profile.dart';
import 'package:wordly/utils/color.dart';
import 'package:wordly/screens/definitions.dart';
import 'package:wordly/screens/definition_welcomesplash.dart';

import '../screens/leaderboard.dart';
import '../screens/review.dart';

class UserDrawer extends StatelessWidget {
  const UserDrawer({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    final FirebaseAuth _auth = FirebaseAuth.instance;
    final user = _auth.currentUser;
    navigateLogin() async {
      Navigator.pushReplacementNamed(context, "login");
    }

    return Drawer(
      child: Column(
        children: [
          Container(
            width: double.infinity,
            padding: const EdgeInsets.all(20),
            color: blueLightColors,
            child: Center(
              child: Column(
                children: [
                  Container(
                    width: 100,
                    height: 100,
                    margin: const EdgeInsets.only(top: 30, bottom: 30),
                    decoration: const BoxDecoration(
                      shape: BoxShape.circle,
                      image: DecorationImage(
                        image: AssetImage('assets/img/user.png'),
                      ),
                    ),
                  ),
                  user != null
                      ? Text(
                          user.email!,
                          style: TextStyle(
                              color: white,
                              fontSize: 18,
                              fontFamily: 'Righteous'),
                        )
                      : const Text(''),
   ],
  ),
 ),
),
     ListTile(
 leading: const Icon(Icons.home_outlined),
 title: const Text('Home'),
 onTap: () {
  Navigator.push(context,
    MaterialPageRoute(builder: (context) => const HomeScreen()));
 },
),

 ListTile(
 leading: const Icon(Icons.person_outline_outlined),
 title: const Text('Profile'),
 onTap: () {
  Navigator.push(context,
    MaterialPageRoute(builder: (context) => const UserProfileScreen()));
 },
),
 ListTile(
 leading: const Icon(Icons.star_half_rounded),
 title: const Text('Add Review'),
 onTap: () {
  Navigator.push(context,
    MaterialPageRoute(builder: (context) => const ReviewScreen()));
 },
),
ListTile(
 leading: const Icon(Icons.leaderboard_outlined),
 title: const Text('Leaderboard'),
 onTap: () {
  Navigator.push(context,
    MaterialPageRoute(builder: (context) => const ULeaderBoard()));
 },
),

ListTile(
 leading: const Icon(Icons.exit_to_app),
 title: const Text('Sign out'),
 onTap: () async {
  await _auth.signOut().then((value) => navigateLogin());
```

```
            },
          ),
        ],
      ),
    );
  }
}
```

5.4 Screens/home.dart

```
import 'package:flutter/material.dart';
import 'package:wordly/utils/color.dart';
import 'package:wordly/widgets/user_drawer.dart';

class HomeScreen extends StatefulWidget {
  const HomeScreen({Key? key}) : super(key: key);

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

class _HomeScreenState extends State<HomeScreen> {
  navigateQuiz() async {
    Navigator.pushReplacementNamed(context, "quiz");
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: blueLightColors,
              appBar: AppBar(
          backgroundColor: blueColors,
          title: Row(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Image.asset('assets/img/Logo.png', fit: BoxFit.cover, height: 60.0,),
              Container(
                padding: const EdgeInsets.all(8.0),
                child: Text(
                  'Wordly',
                  style: TextStyle(
                    fontFamily: 'Righteous',
          fontSize: 20.0
         ),
        ),
       )
      ],
     ),
    ),
    drawer: UserDrawer(),
   body: Column(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
     Container(
      child: Center(
       child: Image.asset('assets/img/Logo.png'),
      ),
     ),
     SizedBox(
      height: 50,
     ),
     ElevatedButton(
      style: ElevatedButton.styleFrom(
       primary: blueColors, // background
       onPrimary: Colors.white,
       padding: EdgeInsets.fromLTRB(100, 15, 100, 15),
       shape: RoundedRectangleBorder(
         borderRadius: BorderRadius.circular(100.0)), // foreground
      ),
      onPressed: () {
       navigateQuiz();
      },
      child: Text(
       'Start Quiz',
       style: TextStyle(
               fontSize: 20.0,
               fontWeight: FontWeight.bold,
               color: Colors.white),
            ),
     ),
    ],
   ),
  );
 }
}
```

5.5 Widgets/question_card.dart

```
// @dart=2.9
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:wordly/models/definition.dart';
import 'package:wordly/models/score.dart';
import 'package:wordly/controllers/score_controller.dart';
import 'package:wordly/providers/definition_provider.dart';
import 'package:wordly/screens/quiz.dart';
import 'package:wordly/widgets/option.dart';
import 'package:provider/provider.dart';
import 'package:wordly/utils/color.dart';
import 'package:firebase_auth/firebase_auth.dart';

class DefinitionCard extends StatelessWidget {
  final int length;
  final Definition definition;

  const DefinitionCard({Key key, @required this.length, @required this.definition})
      : super(key: key);

  @override
  Widget build(BuildContext context) {
    final FirebaseAuth _auth = FirebaseAuth.instance;
    final user = _auth.currentUser;
    final scoreController = ScoreController();
    final createdAt = DateTime.now().toLocal();

    scoreSave(int length) async {
      Score scoreObj = Score(
        name: user.displayName,
        email: user.email,
        points: length,
        createdAt: createdAt,
      );
      await scoreController.addScore(scoreObj);
      await scoreController.updateScore(scoreObj);
    }

    return Container(
child: Column(
  children: [
    Padding(
      padding: EdgeInsets.symmetric(horizontal: 12),
      child: Text(
        definition.definition,
        style: Theme.of(context)
            .textTheme
            .headline6
            .copyWith(color: Colors.black54),
      ),
    ),
    SizedBox(
      height: 12,
    ),
    ...List.generate(
      definition.options.length,
      (index) => Option(index: index, text: definition.options[index]),
    ),
    SizedBox(
      height: 20,
    ),
    Row(
      mainAxisAlignment: MainAxisAlignment.end,
      children: [
        Container(
          margin: EdgeInsets.only(right: 5),
          child: FlatButton(
            onPressed: () {
              Provider.of<DefinitionProvider>(context, listen: false)
                  .checkAnswer(definition);
              if (length !=
                  Provider.of<DefinitionProvider>(context, listen: false)
                      .definitionNumber
                      .value) {
                Future.delayed(Duration(seconds: 1), () {
                  Provider.of<DefinitionProvider>(context, listen: false)
                      .nextDefinition();
                });
              } else {
                int correct =
                    Provider.of<DefinitionProvider>(context, listen: false)
                        .correctAnswers;
```

```
           showDialog(
             context: context,
             builder: (BuildContext context) {
              Future.delayed(Duration(seconds: 3), () {
               Navigator.pushReplacementNamed(context, "home");
              });
              return AlertDialog(
               title: Text("Your Score"),
               content: Text('$correct out of $length'),
              );
             });
           scoreSave(correct);
          }
         },
         child: Text(
          "Next",
          style: TextStyle(color: Colors.white),
         ),
        ),
        color: blueColors,
       ),
      ],
     ),
    ],
   ),
  );
 }
}
```

5.6 Screens/quiz.dart

```
// @dart=2.9
import 'package:flutter/material.dart';
import 'package:wordly/controllers/score_controller.dart';
import 'package:wordly/providers/definition_provider.dart';
import 'package:wordly/widgets/quiz_drawer.dart';
import 'package:wordly/widgets/quiz_body.dart';
import 'package:provider/provider.dart';
import 'package:wordly/utils/color.dart';

class QuizScreen extends StatelessWidget {

  @override
  Widget build(BuildContext context) {
    final controller = ScoreController();
    return MultiProvider(
      providers: [
        ChangeNotifierProvider(create: (context) => DefinitionProvider()),
        StreamProvider(create: (context) => controller.getQuesions()),
      ],
      child: Scaffold(
        backgroundColor: blueLightColors,
        appBar: AppBar(
          backgroundColor: blueColors,
          title: Row(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Image.asset('assets/img/Logo.png', fit: BoxFit.cover, height: 60.0,),
              Container(
                padding: const EdgeInsets.all(8.0),
                child: Text(
                  'Wordly',
                  style: TextStyle(
                    fontFamily: 'Righteous',
                    fontSize: 20.0
                  ),
                ),
              )
            ],
          ),
        ),
        drawer: QuizDrawer(),
        body: QuizBody(),
      ),
    );
  }
}
```

5.7 Widgets/quiz_drawer.dart

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:wordly/screens/admin_users.dart';
import 'package:wordly/screens/home.dart';
import 'package:wordly/utils/color.dart';
import 'package:wordly/screens/definitions.dart';
import 'package:wordly/screens/welcomesplash.dart';

import '../screens/leaderboard.dart';

class QuizDrawer extends StatelessWidget {
  const QuizDrawer({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    final FirebaseAuth _auth = FirebaseAuth.instance;
    final user = _auth.currentUser;
    navigateLogin() async {
      Navigator.pushReplacementNamed(context, "login");
    }

    return Drawer(
      child: Column(
        children: [
          Container(
            width: double.infinity,
            padding: const EdgeInsets.all(20),
            color: blueLightColors,
            child: Center(
              child: Column(
                children: [
                  Container(
                    width: 100,
                    height: 100,
                    margin: const EdgeInsets.only(top: 30, bottom: 30),
                    decoration: const BoxDecoration(
           shape: BoxShape.circle,
           image: DecorationImage(
            image: AssetImage('assets/img/user.png'),
           ),
          ),
         ),
         user != null
           ? Text(
             user.email!,
             style: TextStyle(
               color: white,
               fontSize: 18,
               fontFamily: 'Righteous'),
            )
           : const Text(''),
        ],
       ),
      ),
     ),


     ListTile(
      leading: const Icon(Icons.home_outlined),
      title: const Text('Home'),
      onTap: () {
       Navigator.push(context,
         MaterialPageRoute(builder: (context) => const HomeScreen()));
      },
     ),
     ListTile(
      leading: const Icon(Icons.exit_to_app),
      title: const Text('Sign out'),
      onTap: () async {
       await _auth.signOut().then((value) => navigateLogin());
      },
     ),
    ],
   ),
  );
 }
}
```