iGamer

Project ID:

Report

P.P.G.S.H.A.Guruge

B.Sc. Special (Hons.) Degree in Information Technology

Department of Software Engineering

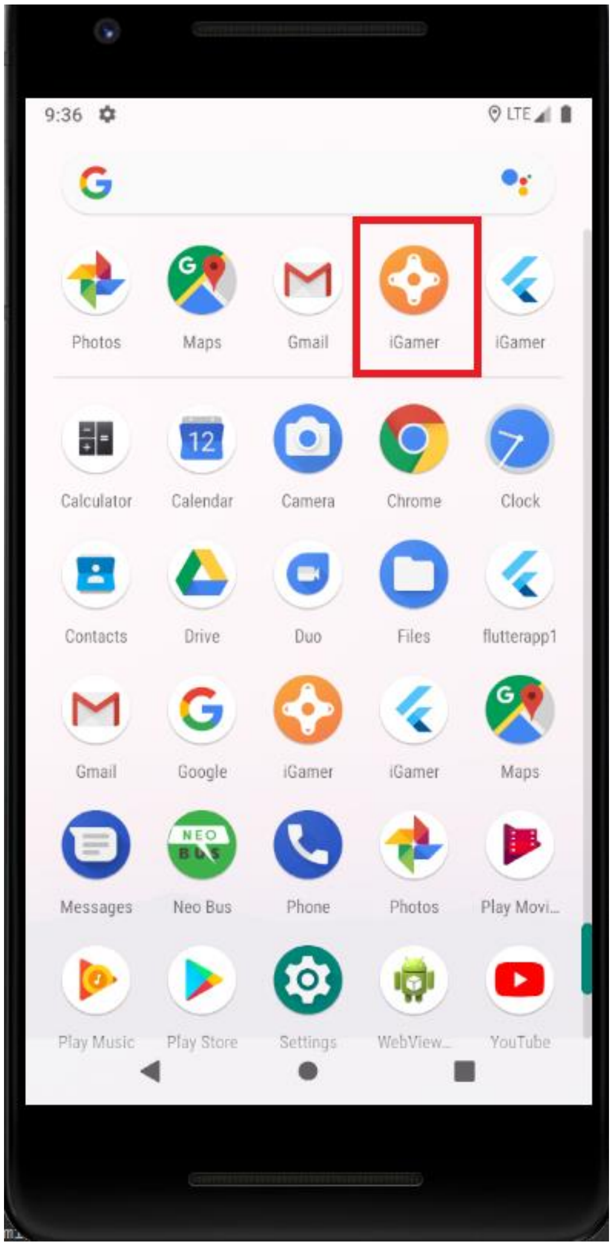
Sri Lanka Institute of Information Technology
Sri Lanka

March 2020

Table of Contents

No table of contents entries found.

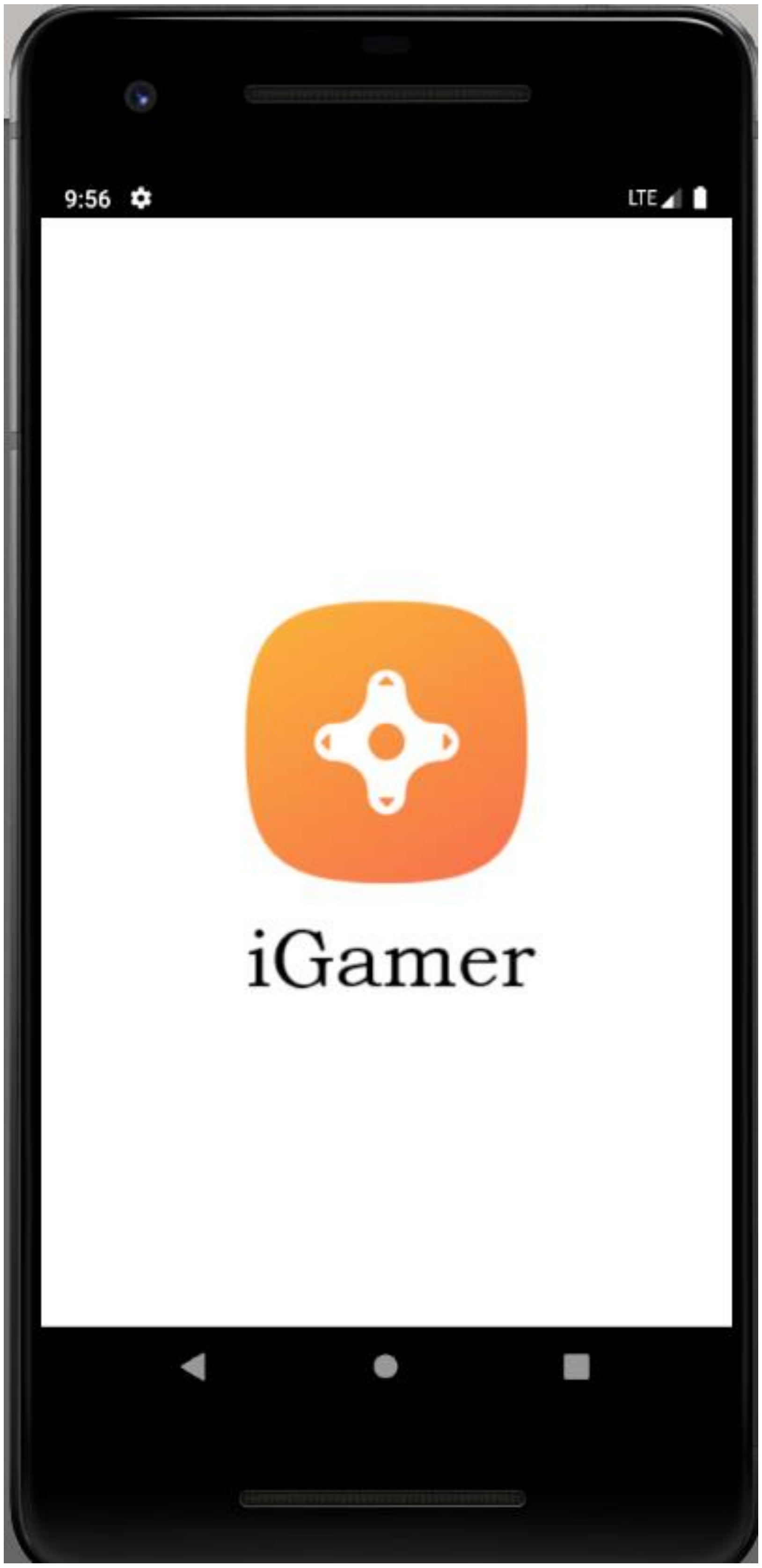
App Icon


9:36
LTE
Photos
Maps
Gmail
iGamer
iGamer
Calculator
Calendar
Camera
Chrome
Clock
Contacts
Drive
Duo
Files
flutterapp1
Gmail
Google
iGamer
iGamer
Maps
Messages
Neo Bus
Phone
Photos
Play Movi...
Play Music
Play Store
Settings
WebView...
YouTube

In the following AndroidManifest.xml file, the app name is set in **android:name** and app launcher icon is set in **android:icon** values.

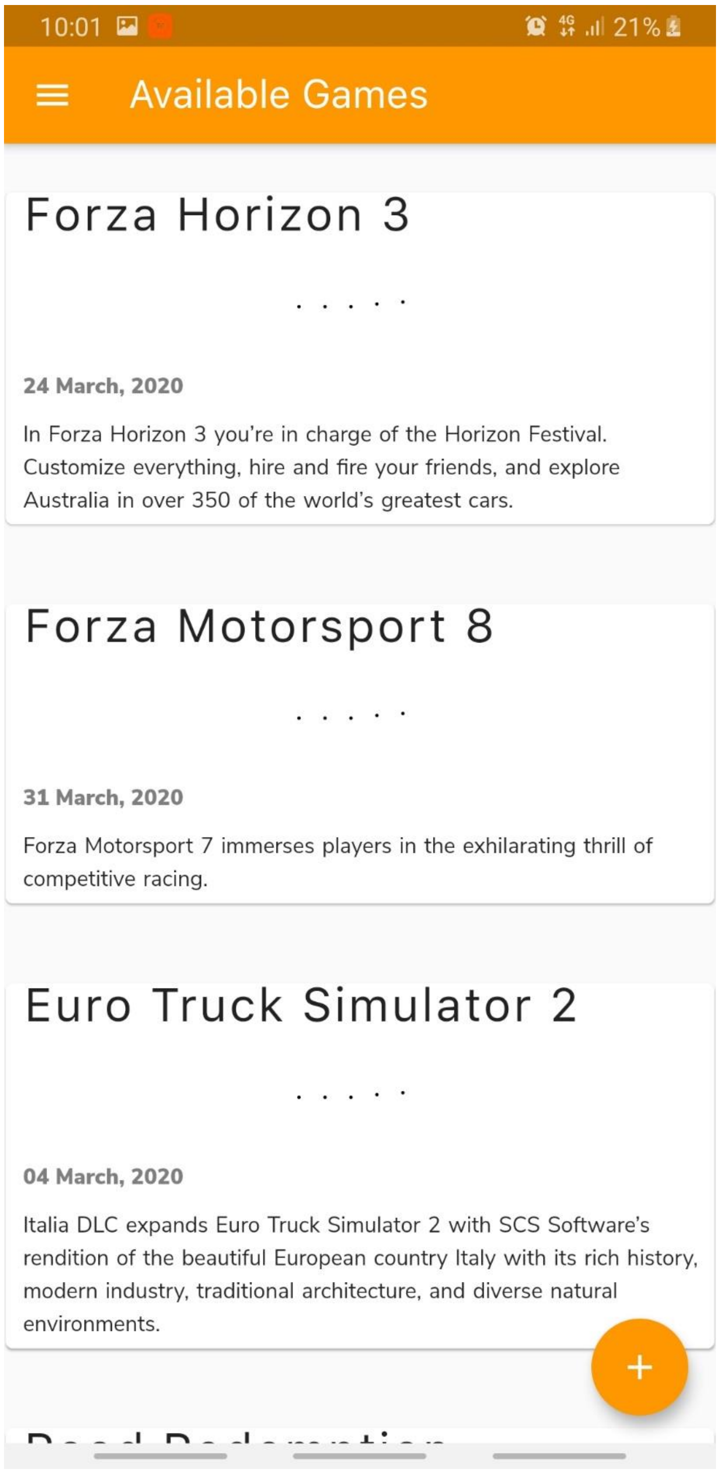
```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.igamer">
    <!-- io.flutter.app.FlutterApplication is an android.app.Application that
         calls FlutterMain.startInitialization(this); in its onCreate method.
         In most cases you can leave this as-is, but you if you want to provide
         additional functionality it is fine to subclass or reimplement
         FlutterApplication and put your custom class here. -->
    <application
        android:name="io.flutter.app.FlutterApplication"
        android:label="iGamer"
        android:icon="@mipmap/ic_launcher">
        <activity
            android:name=".MainActivity"
            android:launchMode="singleTop"
            android:theme="@style/LaunchTheme"

android:configChanges="orientation|keyboardHidden|keyboard|screenSize|smallestScr
eenSize|locale|layoutDirection|fontScale|screenLayout|density|uiMode"
            android:hardwareAccelerated="true"
            android:windowSoftInputMode="adjustResize">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <!-- Don't delete the meta-data below.
             This is used by the Flutter tool to generate
GeneratedPluginRegistrant.java -->
        <meta-data
            android:name="flutterEmbedding"
            android:value="2" />
    </application>
</manifest>
```

02. Splash Screen



```xml
<?xml version="1.0" encoding="utf-8"?>
<!-- Modify this file to customize your launch splash screen -->
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@android:color/white" />
    <item android:bottom="12dp">
        <bitmap
            android:gravity="center"
            android:src="@drawable/icon"/>
    </item>
    <item android:top="250dp">
        <bitmap
            android:gravity="center"
            android:src="@drawable/imagetext"/>
    </item>
</layer-list>
```

Main page



10:01 21%

Available Games

Forza Horizon 3

. . . . .

24 March, 2020

In Forza Horizon 3 you're in charge of the Horizon Festival. Customize everything, hire and fire your friends, and explore Australia in over 350 of the world's greatest cars.

Forza Motorsport 8

. . . . .

31 March, 2020

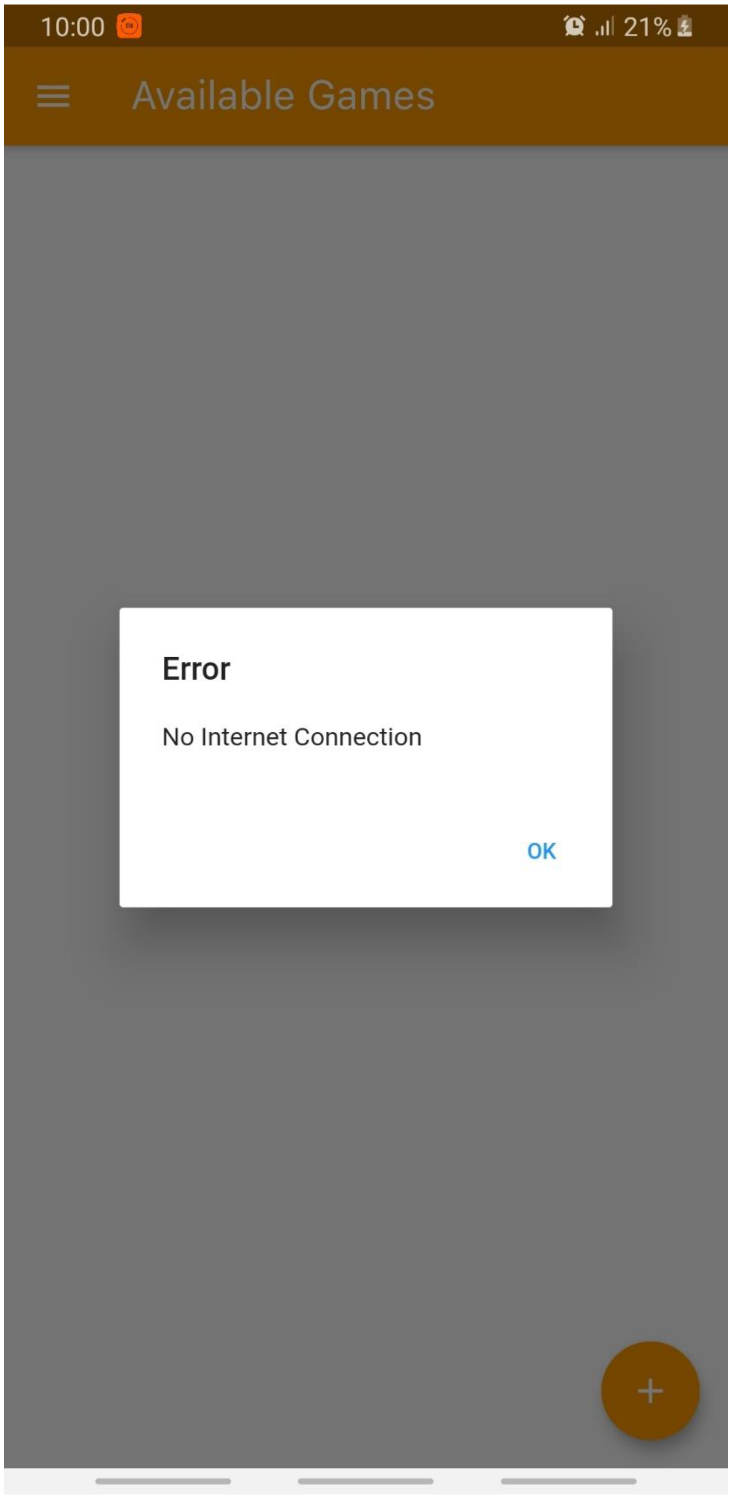
Forza Motorsport 7 immerses players in the exhilarating thrill of competitive racing.

Euro Truck Simulator 2

. . . . .

04 March, 2020

Italia DLC expands Euro Truck Simulator 2 with SCS Software's rendition of the beautiful European country Italy with its rich history, modern industry, traditional architecture, and diverse natural environments.

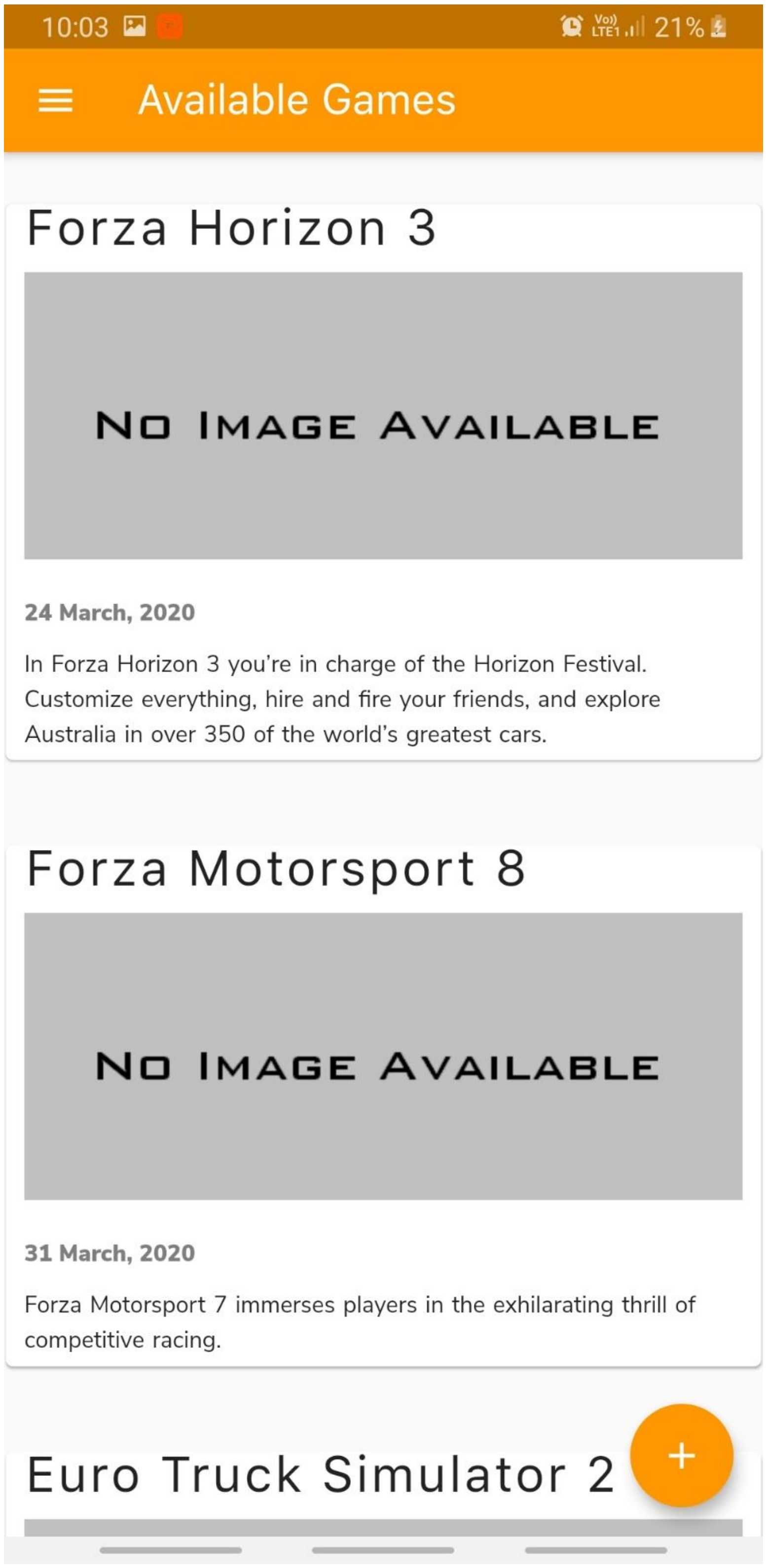


10:00 21%

Available Games

Error

No Internet Connection



10:03 21%

Available Games

Forza Horizon 3

NO IMAGE AVAILABLE

24 March, 2020

In Forza Horizon 3 you're in charge of the Horizon Festival. Customize everything, hire and fire your friends, and explore Australia in over 350 of the world's greatest cars.

Forza Motorsport 8

NO IMAGE AVAILABLE

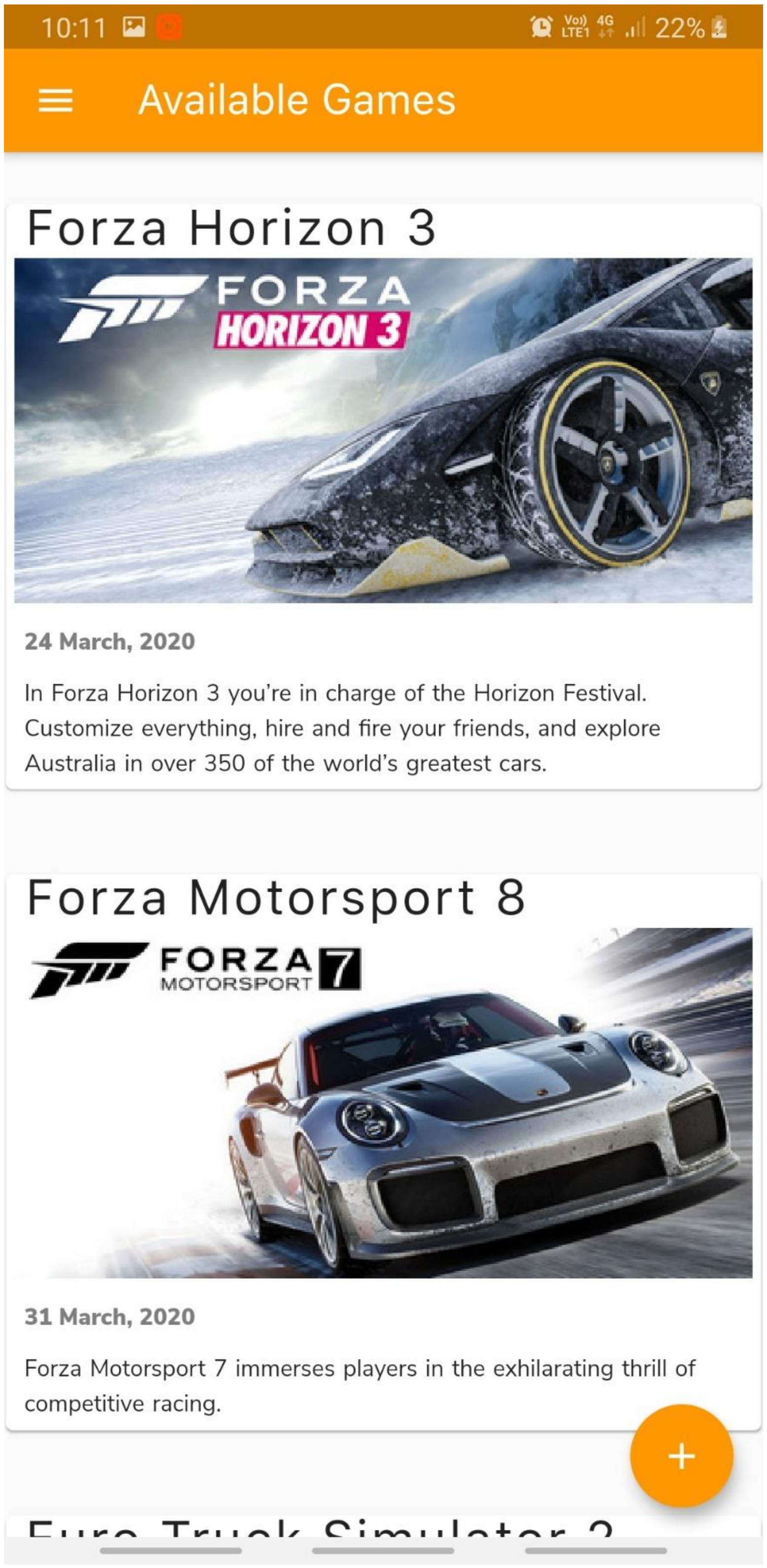
31 March, 2020

Forza Motorsport 7 immerses players in the exhilarating thrill of competitive racing.

Euro Truck Simulator 2

main.dart

```dart
import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:igamer/database/crud.dart';
import '../database/gameRecord.dart';
import '../common_ui_widgets/appBar.dart';
import '../common_ui_widgets/drawer.dart';
import '../common_ui_widgets/gameCard.dart';
import '../common_ui_widgets/alertBox.dart';
import 'addGame.dart';
import 'dart:io';

// Name of the page
final pageTitle = "Available Games";

// Main method
void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: pageTitle,
      home: MyHomePage(title: pageTitle),
    );
  }
}

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

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

class _MyHomePageState extends State<MyHomePage> {

  // this function checks active internet connection
  // if not, it will popup an Alert Box
  _checkInternetConnection(BuildContext context) async {
    try {
      await InternetAddress.lookup('google.com');
    } on SocketException catch (_) {
      new AppAlertBox(context, "Error", "No Internet Connection", "OK")
          .showAlertDialog();
    }
  }

  @override
  Widget build(BuildContext context) {
    _checkInternetConnection(context);
    return Scaffold(
      appBar: new CustomizedAppBar(pageTitle).getAppBar(), // Calling Custom
build app bar
      body: _buildBody(context),
      floatingActionButton: FloatingActionButton(
        onPressed: () {
          Navigator.push(
              context, MaterialPageRoute(builder: (context) => AddGame())); //
Navigates to Add Game screen
        },
        tooltip: 'Increment',
        child: Icon(Icons.add),
        backgroundColor: Colors.orange,
      ),
      drawer: new CustomizedDrawer(context).getDrawer(),
    );
  }

  Widget _buildBody(BuildContext context) {
    return StreamBuilder<QuerySnapshot>(
      stream: new CRUD().getGames(), // getting a list of games
      builder: (context, snapshot) {
        // checking if data exists
        if (!snapshot.hasData)
          // if no data a Circular Progress Indicator shows up in the middle of
the screen
          return Center(
              child: new Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: <Widget>[
              Container(
                height: 50,
                width: 50,
                child: CircularProgressIndicator(),
              ),
              Container(
                margin: const EdgeInsets.only(top: 10),
                child: Text(
                  "Loading",
                  style: TextStyle(fontSize: 18),
                ),
              )
            ],
          ));
        // if data exist build a list
        return _buildList(context, snapshot.data.documents);
      },
    );
  }
  // this function returns a ListView based on snapShot data
  Widget _buildList(BuildContext context, List<DocumentSnapshot> snapShot) {
    return ListView(
      padding: const EdgeInsets.only(top: 20),
      children: snapShot.map((data) => _buildListItem(context, data)).toList(),
    );
  }

  // this function returns a Card embedded with Padding
  Widget _buildListItem(BuildContext context, DocumentSnapshot data) {
    final gameRecord = GameRecord.fromSnapshot(data);
    return Padding(
        key: ValueKey(gameRecord.title),
        padding: const EdgeInsets.symmetric(horizontal: 1, vertical: 8),
        child: GameCard(game: gameRecord));
  }
}
```


10:11 22%

Available Games

Forza Horizon 3

24 March, 2020

In Forza Horizon 3 you're in charge of the Horizon Festival. Customize everything, hire and fire your friends, and explore Australia in over 350 of the world's greatest cars.

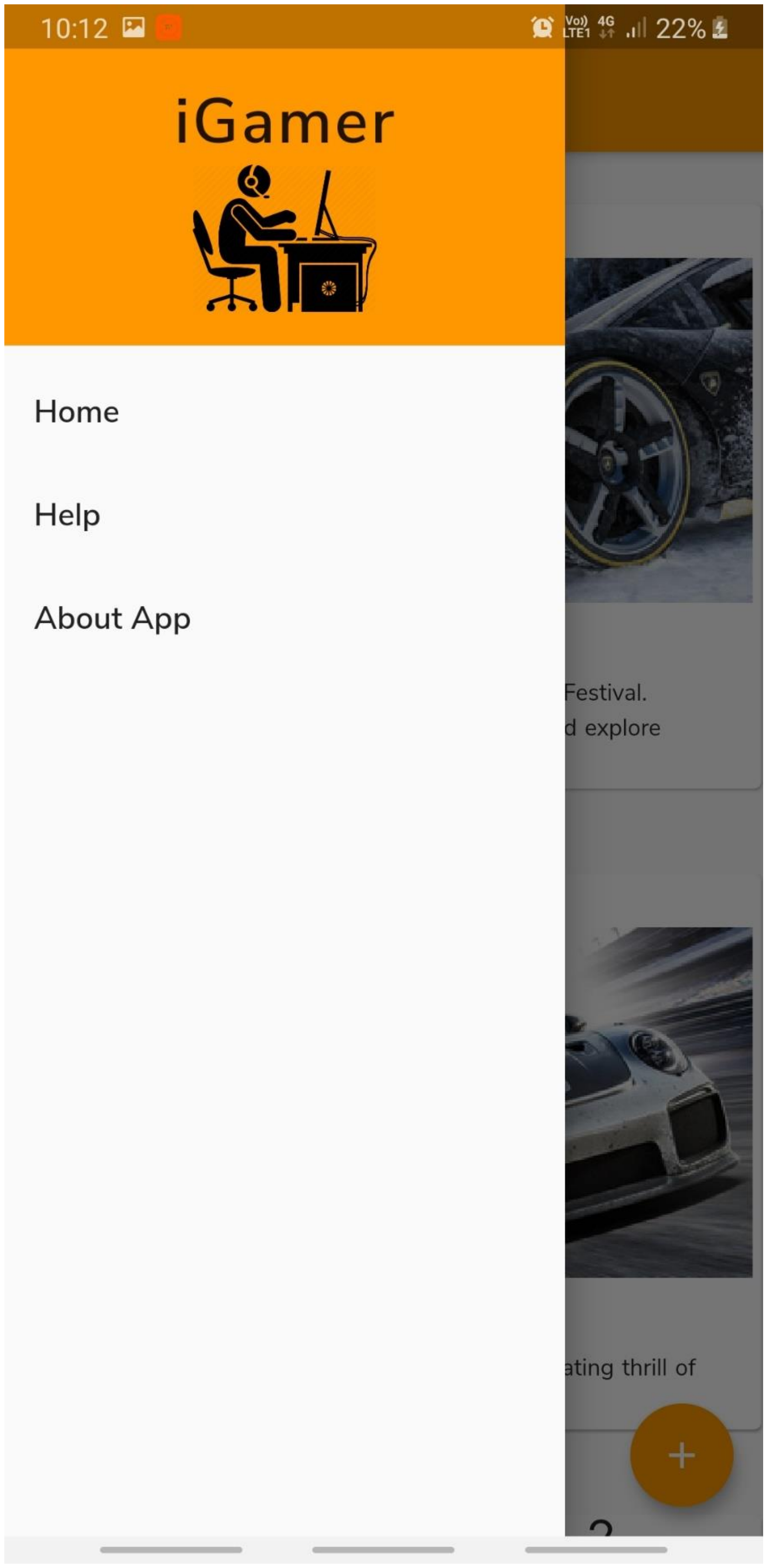
Forza Motorsport 8

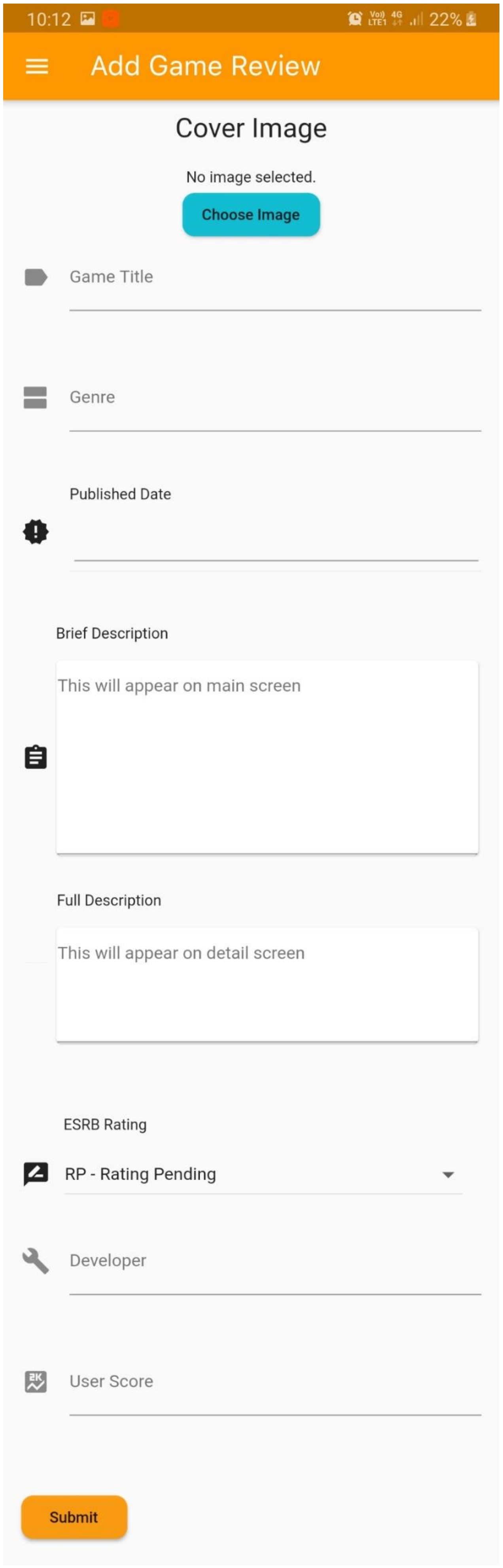
31 March, 2020

Forza Motorsport 7 immerses players in the exhilarating thrill of competitive racing.

+

Euro Truck Simulator 2


10:12
22%
iGamer
Home
Help
About App
Festival.
d explore
ating thrill of



10:12 22%

Add Game Review

Cover Image

No image selected.

Game Title

Genre

Published Date

Brief Description

This will appear on main screen

Full Description

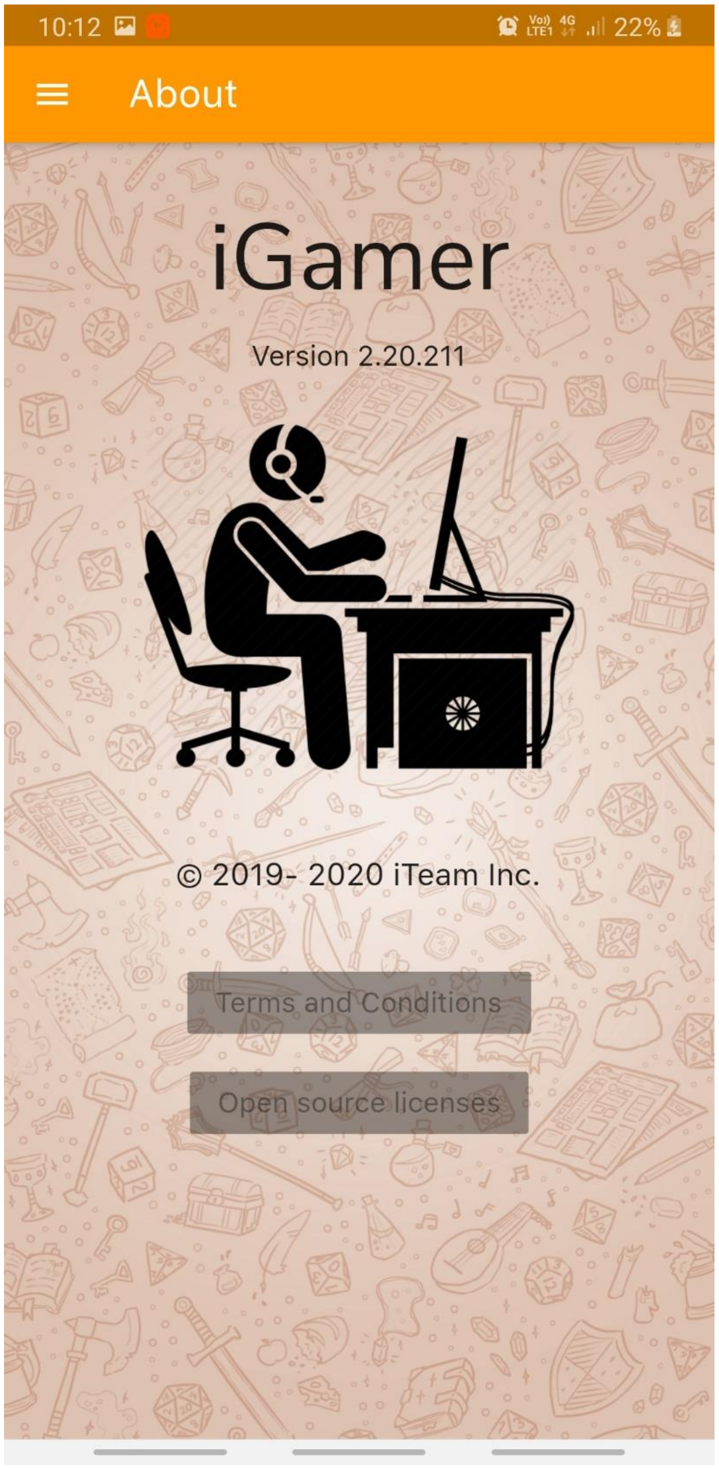
This will appear on detail screen

ESRB Rating

RP - Rating Pending

Developer

User Score


10:12
22%

About

iGamer

Version 2.20.211

© 2019- 2020 iTeam Inc.

<u>Common UI Widgets</u>

appBar.dart

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

// This class contains attributes and methods for a Customized App Bar
class CustomizedAppBar {
  final Color backgroundColor = Colors.orange;
  String title;

  // Constructor
  CustomizedAppBar(this.title);

  // this function returns the Customized App Bar
  Widget getAppBar() {
    return new AppBar(
      title: Text(title, style: TextStyle(fontSize: 25, fontFamily:
'SanFrancisco'),),
      backgroundColor: backgroundColor,
    );
  }
}
```

drawer.dart

```
import 'package:flutter/material.dart';
import 'package:igamer/screens/about.dart';
import 'package:igamer/screens/help.dart';
import 'package:igamer/screens/main.dart';

// this class contains attributes and methods for App Drawer
class CustomizedDrawer {
  final Color backgroundColor = Colors.orange;
  BuildContext context;

  CustomizedDrawer(this.context);

  // this function returns a Customized App Drawer
  Widget getDrawer() {
    return new Drawer(
      child: ListView(
        padding: EdgeInsets.zero,
        children: <Widget>[
          DrawerHeader(
            child: new Column(
              children: <Widget>[
                new Container(
                  child: Text('iGamer',
                    style: TextStyle(fontSize: 38, letterSpacing: 1.5 ,
fontFamily: 'NunitoSansSemiBold'),),
                ),
                new Container(
                  child: new Image.asset('assets/images/gamer.png'),
                  height: 80,
                  width: 100,
                )
              ],
            ),
            decoration: BoxDecoration(color: Colors.orange),
          ),
          ListTile(
            title: Text('Home', style: TextStyle(fontFamily:
'NunitoSansSemiBold', fontSize: 19),),
            onTap: () => {
              Navigator.push(context, MaterialPageRoute(builder: (context) => new
MyHomePage()))},
          ),
          ListTile(
            title: Text('Help', style: TextStyle(fontFamily:
'NunitoSansSemiBold', fontSize: 19),),
            onTap: () => {
              Navigator.push(context, MaterialPageRoute(builder: (context) => new
HelpScreenPage()))},
          ),
          ListTile(
            title: Text('About App', style: TextStyle(fontFamily:
'NunitoSansSemiBold', fontSize: 19),),
            onTap: () => {Navigator.push(context, MaterialPageRoute(builder:
(context) => new AboutScreenPage()))},
          )
        ],
      ),
    );
  }
}
```

inputWidgets.dart

```dart
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:flutter/services.dart';
import 'package:datetime_picker_formfield/datetime_picker_formfield.dart';
import 'package:intl/intl.dart';

// this class contains all the common input widgets used in the app
class CommonInputWidgets {

  // this function returns a TextField
  Container getTextField(String labelText, String hintText, IconData icon,
      TextEditingController controller) {
    return (Container(
      height: 100,
      child: TextField(
        decoration: InputDecoration(
            labelText: labelText, hintText: hintText, icon: Icon(icon)),
        controller: controller,
      ),
    ));
  }

  // this function return a date picker
  Container getDatePicker(
      String label, IconData icon, TextEditingController controller) {
    final format = DateFormat("dd MMMM, yyyy");

    return Container(
      margin: const EdgeInsets.only(bottom: 30),
      child: Column(children: <Widget>[
        Container(
          alignment: Alignment(-1, -1),
          child: Text(label),
          margin: const EdgeInsets.only(left: 40),
        ),
        Row(
          children: <Widget>[
            Container(
              child: new Icon(icon),
            ),
            Container(
              width: 335,
              margin: const EdgeInsets.only(left: 20),
              child: DateTimeField(
                format: format,
                controller: controller,
                onShowPicker: (context, currentValue) {
                  return showDatePicker(
                      context: context,
                      firstDate: DateTime(1900),
                      initialDate: currentValue ?? DateTime.now(),
                      lastDate: DateTime(2100));
                },
              ),
            )
          ],
        )
      ]),
    );
  }

  // this function returns a Number Text Field
  // if the parameter onlyDigits is true , only digits can be entered (not point
values)
  Container getNumberTextField(String labelText, String hintText, IconData icon,
      bool onlyDigits, TextEditingController controller) {
    return (Container(
      height: 100,
      child: TextField(
        decoration: InputDecoration(
            labelText: labelText, hintText: hintText, icon: Icon(icon)),
        controller: controller,
        keyboardType: TextInputType.number,
        inputFormatters: <TextInputFormatter>[
          if (onlyDigits) WhitelistingTextInputFormatter.digitsOnly
        ],
      ),
    ));
  }

  // this function returns a Text Area
  Container getTextArea(String labelText, String hintText, IconData icon,
      TextEditingController controller) {
    return (Container(
        margin: const EdgeInsets.only(bottom: 30),
        child: Column(
          children: <Widget>[
            Container(
              alignment: Alignment(-.8, -1),
              child: Text(labelText),
              margin: const EdgeInsets.only(bottom: 15),
            ),
            Row(
              children: <Widget>[
                Container(
                  child: new Icon(icon),
                ),
                Container(
                  width: 355,
                  child: Card(
                      color: Colors.white,
                      margin: const EdgeInsets.only(left: 5),
                      child: Padding(
                        padding: EdgeInsets.all(1.0),
                        child: TextField(
                          maxLines: 8,
                          decoration: InputDecoration(hintText: hintText),
                          controller: controller,
                        ),
                      )),
                )
              ],
            )
          ],
        )));
  }

}
```

Database

crud.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:igamer/database/gameRecord.dart';

//
gameID|this.title|publishedDate|gameDescription|imageLink|genre|developer|release
Date|fullDescription|esrbRating|userScore|noOfUsers

// This class contains the necessary CRUD actions and attributes for the games
used in the app
class CRUD {

  // Collection name
  final String _collection = "games";

  // Add a new game
  Future<void> addGame(GameRecord gameRecord) async {
    final db = Firestore.instance;
    await db.collection("games").add({
      'gameID': gameRecord.gameID,
      'title': gameRecord.title,
      'publishedDate': gameRecord.publishedDate,
      'gameDescription': gameRecord.gameDescription,
      'imageLink': gameRecord.imageLink,
      'genre': gameRecord.genre,
      'developer': gameRecord.developer,
      'releaseDate': gameRecord.releaseDate,
      'fullDescription': gameRecord.fullDescription,
      'esrbRating': gameRecord.esrbRating,
      'userScore': gameRecord.userScore,
      'noOfUsers': gameRecord.noOfUsers,
    }).then((documentReference) {
      print(documentReference.documentID);
    }).catchError((e) {
      print(e);
    });
  }

  // Get all games
  Stream<QuerySnapshot> getGames(){
    return Firestore.instance.collection(_collection).snapshots();
  }
}
```

gameRecord.dart

```
import 'package:cloud_firestore/cloud_firestore.dart';

// This class contains attributes and relevant methods for the Game entity
class GameRecord {
  final int gameID;
  final String title;
  final String publishedDate;
  final String gameDescription;
  final String imageLink;
  final String genre;
  final String developer;
  final String releaseDate;
  final String fullDescription;
  final String esrbRating;
  final String userScore;
  final String noOfUsers;
  final DocumentReference reference;

  // Constructor
  GameRecord(this.gameID, this.title, this.publishedDate, this.gameDescription,
      this.imageLink, this.genre, this.developer, this.releaseDate,
      this.fullDescription, this.esrbRating, this.userScore, this.noOfUsers,
this.reference);

  // this function maps the attributes received from map to GameRecord class
  // meanwhile this function also asserts if the all the mapping attributes are
null
  GameRecord.fromMap(Map<String, dynamic> map, {this.reference})
      : assert(map['gameID'] != null),
        assert(map['title'] != null),
        assert(map['publishedDate'] != null),
        assert(map['gameDescription'] != null),
        assert(map['imageLink'] != null),
        assert(map['genre'] != null),
        assert(map['developer'] != null),
        assert(map['releaseDate'] != null),
        assert(map['fullDescription'] != null),
        assert(map['esrbRating'] != null),
        assert(map['userScore'] != null),
        assert(map['noOfUsers'] != null),
        gameID = map['gameID'],
        title = map['title'],
        publishedDate = map['publishedDate'],
        gameDescription = map['gameDescription'],
        imageLink = map['imageLink'],
        genre = map['genre'],
        developer = map['developer'],
        releaseDate = map['releaseDate'],
        fullDescription = map['fullDescription'],
        esrbRating = map['esrbRating'],
        userScore = map['userScore'],
        noOfUsers = map['noOfUsers'];

  GameRecord.fromSnapshot(DocumentSnapshot snapshot)
      : this.fromMap(snapshot.data, reference: snapshot.reference);

  @override
  String toString() => "Record<$title:$title>";
}
```