**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

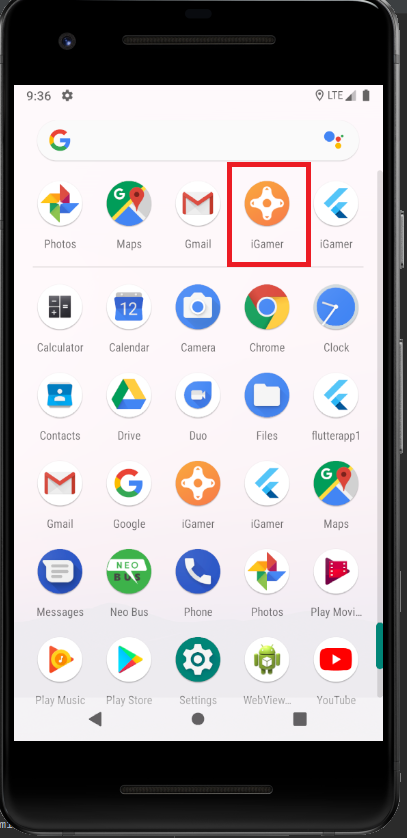
March 2020

**Table of Contents**

**No table of contents entries found.**

**App Icon**

The name “iGamer” is chosen for the app name and orange is the theme colour chosen for the iGamer app. The launcher icon is a picture of a game console.



AndroidManifest.xml

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|smallestScreenSize|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>

**Splash Screen**

The app consists of a Splash Screen showing the app icon and app name before starting the app.

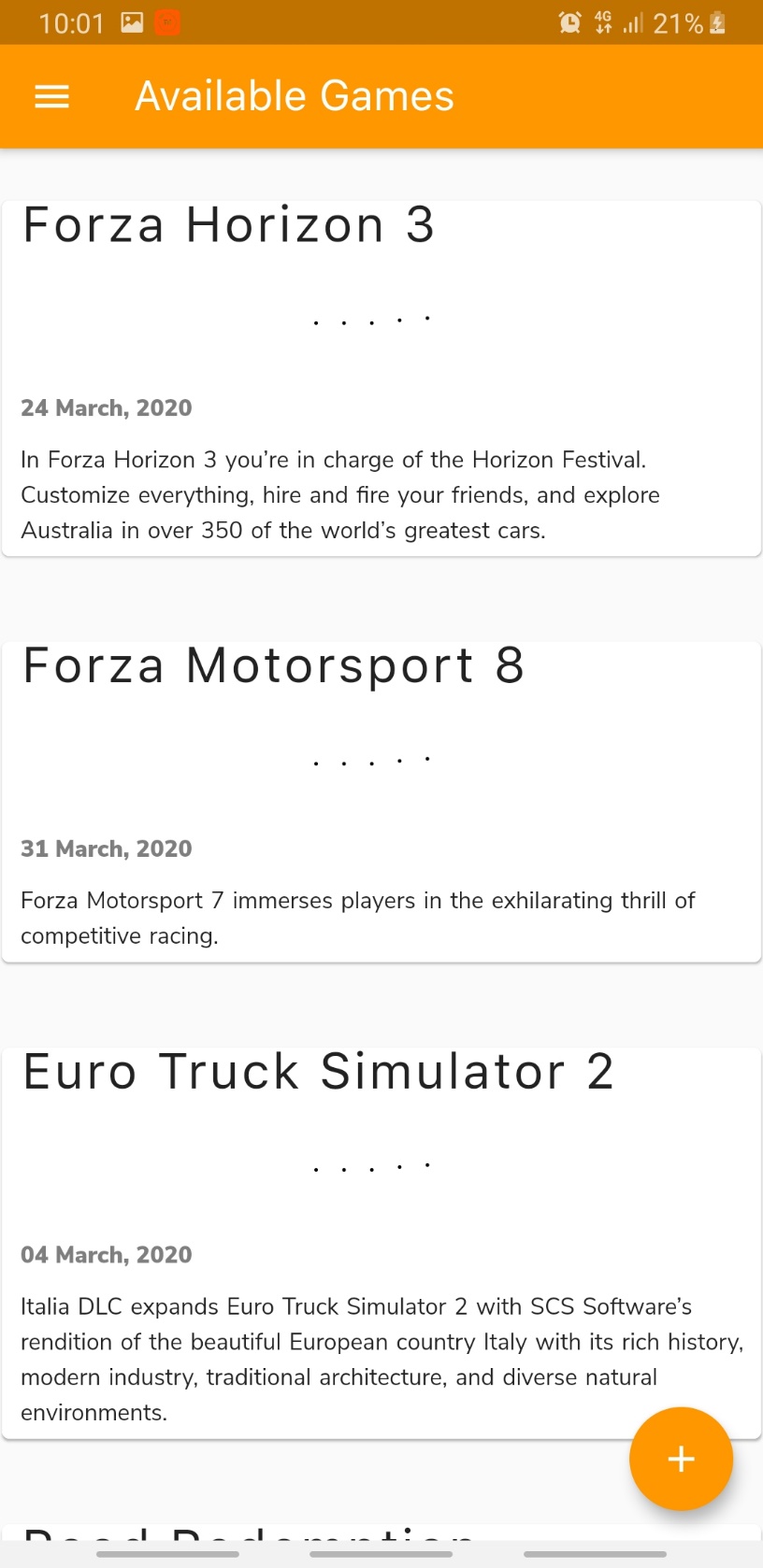
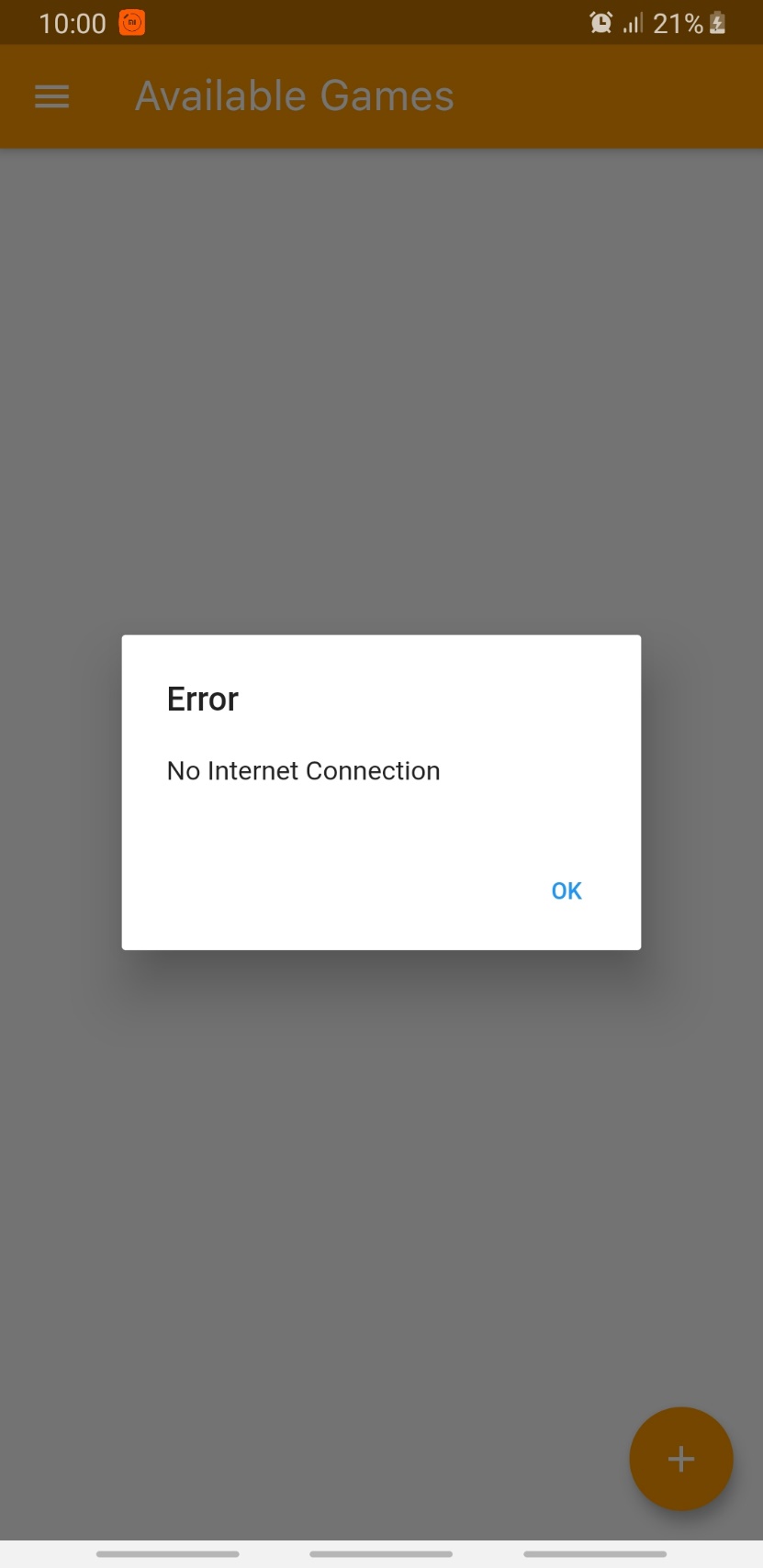


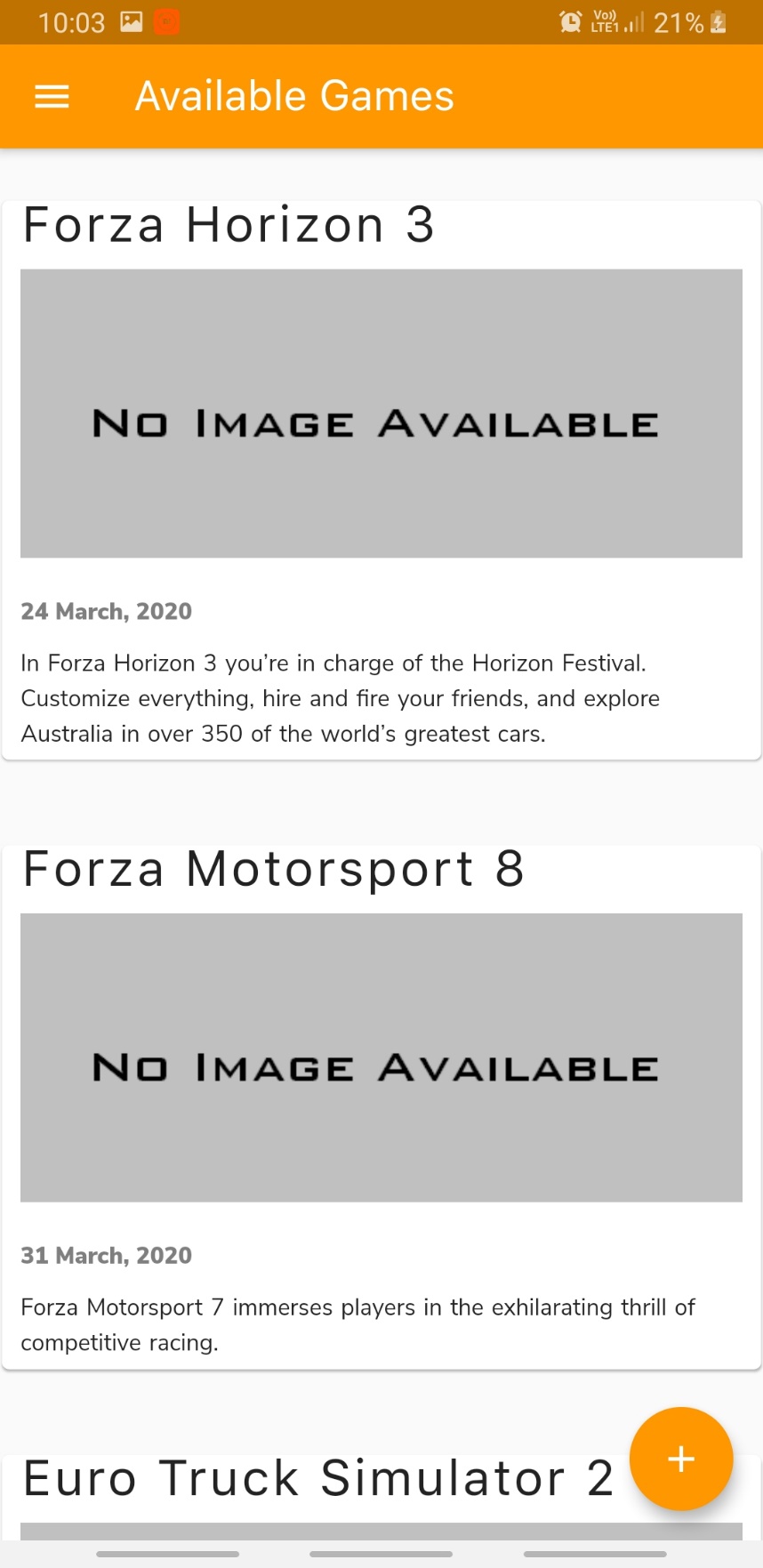
launch\_background.xml

The splash screen is implemented as follow

<?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>

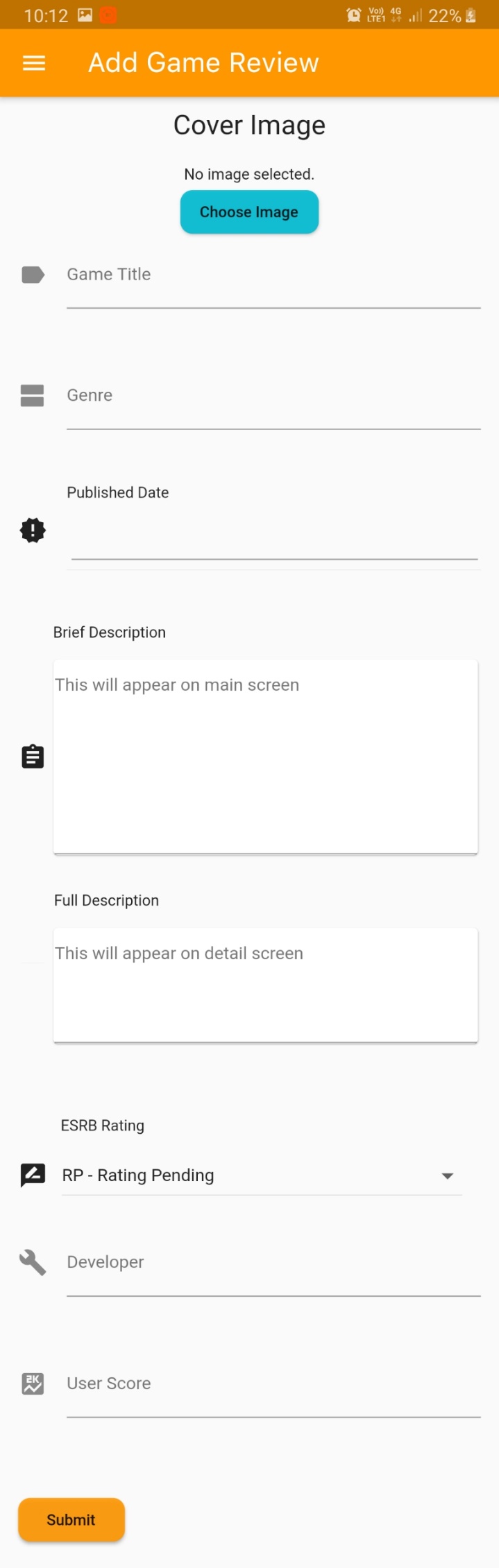
**Main page**





main.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));  
 }  
}

**Add Game**

**addGame.dart**

import 'dart:async';  
import 'dart:math';  
  
import 'package:flutter/material.dart';  
import 'package:igamer/common\_ui\_widgets/drawer.dart';  
import 'package:igamer/common\_ui\_widgets/inputWidgets.dart';  
import 'package:igamer/database/crud.dart';  
import 'package:igamer/database/gameRecord.dart';  
import 'package:igamer/database/imageUploader.dart';  
import '../common\_ui\_widgets/appBar.dart';  
import 'dart:io';  
import 'package:image\_picker/image\_picker.dart';  
import 'main.dart';  
  
// Name of the page  
final title = 'Add Game Review';  
  
// Main method  
void main() => runApp(AddGame());  
  
class AddGame extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 title: title,  
 home: Scaffold(  
 appBar: new CustomizedAppBar(title).getAppBar(),  
 //getting custom built app bar  
 body: AddGameForm(title: title),  
 drawer: new CustomizedDrawer(context)  
 .getDrawer(), //getting custom built app drawer  
 ),  
 );  
 }  
}  
  
// Create a Form widget.  
class AddGameForm extends StatefulWidget {  
 final String title;  
  
 AddGameForm({Key key, this.title}) : super(key: key);  
  
 @override  
 AddGameFormState createState() {  
 return AddGameFormState();  
 }  
}  
  
class AddGameFormState extends State<AddGameForm> {  
 final \_formKey = GlobalKey<FormState>();  
 List<DropdownMenuItem<String>> \_dropDownMenuItems;  
 String \_selectedESRBRating;  
 File \_image;  
  
 //Initializing text editing controllers  
 TextEditingController \_titleController = new TextEditingController();  
 TextEditingController \_genreController = new TextEditingController();  
 TextEditingController \_developerController = new TextEditingController();  
 TextEditingController \_noOfUsersController = new TextEditingController();  
 TextEditingController \_userScoreController = new TextEditingController();  
 TextEditingController \_briefDescController = new TextEditingController();  
 TextEditingController \_fullDescController = new TextEditingController();  
 TextEditingController \_pubDateController = new TextEditingController();  
 TextEditingController \_relDateController = new TextEditingController();  
 CommonInputWidgets \_commonInputWidgets = new CommonInputWidgets();  
  
 //Initializing drop down values for ESRB Ratings  
 List \_ratings = [  
 "RP - Rating Pending",  
 "EC - Early Childhood",  
 "E - Everyone",  
 "E10+ - Everyone 10+",  
 "T - Teen",  
 "M - Mature",  
 "AO - Adults Only"  
 ];  
  
 // initializing local variables at the beginning of the screen  
 @override  
 void initState() {  
 super.initState();  
 \_dropDownMenuItems = \_buildAndGetDropDownMenuItems(\_ratings);  
 \_selectedESRBRating = \_dropDownMenuItems[0].value;  
 \_image = null;  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 return Form(  
 key: \_formKey,  
 child: SingleChildScrollView(  
 padding: const EdgeInsets.only(left: 15, right: 15, top: 10),  
 child: Column(  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: <Widget>[  
 \_getImagePicker(),  
 \_commonInputWidgets.getTextField(  
 "Game Title", "Forza Horizon", Icons.*label*, \_titleController),  
 \_commonInputWidgets.getTextField(  
 "Genre",  
 "Racing, Simulation, Automobile",  
 Icons.*view\_agenda*,  
 \_genreController),  
 \_commonInputWidgets.getDatePicker(  
 "Released Date", Icons.*calendar\_today*, \_relDateController),  
 \_commonInputWidgets.getDatePicker(  
 "Published Date", Icons.*new\_releases*, \_pubDateController),  
 \_commonInputWidgets.getNumberTextField(  
 "No Of Users", "2", Icons.*person*, true, \_noOfUsersController),  
 \_commonInputWidgets.getTextArea(  
 "Brief Description",  
 "This will appear on main screen",  
 Icons.*assignment*,  
 \_briefDescController),  
 \_commonInputWidgets.getTextArea(  
 "Full Description",  
 "This will appear on detail screen",  
 Icons.*videogame\_asset*,  
 \_fullDescController),  
 \_getDropDown("ESRB Rating", Icons.*rate\_review*),  
 \_commonInputWidgets.getTextField("Developer", "Playground Games",  
 Icons.*build*, \_developerController),  
 \_commonInputWidgets.getNumberTextField(  
 "User Score", "7.8", Icons.*score*, false, \_userScoreController),  
 Padding(  
 padding: const EdgeInsets.symmetric(vertical: 16.0),  
 child: RaisedButton(  
 onPressed: () async {  
 if (\_formKey.currentState.validate()) {  
 Scaffold.*of*(context).showSnackBar(SnackBar(  
 content: Text('Adding New Game Review'),  
 duration: const Duration(seconds: 1),  
 ));  
 ImageUploader uploader = new ImageUploader(  
 \_titleController.text +  
 "-" +  
 \_generateID().toString(),  
 \_image);  
 var imageURL = await uploader.uploadFile();  
 GameRecord game = new GameRecord(  
 \_generateID(),  
 \_titleController.text,  
 \_pubDateController.text,  
 \_briefDescController.text,  
 imageURL,  
 \_genreController.text,  
 \_developerController.text,  
 \_relDateController.text,  
 \_fullDescController.text,  
 \_selectedESRBRating,  
 \_userScoreController.text,  
 \_noOfUsersController.text,  
 null);  
 await CRUD().addGame(game);  
 Navigator.*push*(  
 context,  
 MaterialPageRoute(  
 builder: (context) => MyHomePage()));  
 }  
 },  
 child: Text('Submit'),  
 color: Colors.*orange*.withOpacity(0.9),  
 shape: RoundedRectangleBorder(  
 borderRadius: new BorderRadius.circular(10.0))),  
 ),  
 ],  
 ),  
 ),  
 );  
 }  
  
 // this function returns a random integer between 0 and 10000  
 int \_generateID() {  
 var random = Random();  
 return random.nextInt(10000);  
 }  
  
 // this function picks an image from the gallery and set to \_image  
 Future \_getImage() async {  
 return ImagePicker.*pickImage*(source: ImageSource.gallery).then((file) {  
 setState(() {  
 \_image = file;  
 });  
 });  
 }  
  
 // the function removes the selected image from the gallery  
 Future \_clearImage() async {  
 setState(() {  
 \_image = null;  
 });  
 }  
  
 // this function returns a Column having an Image Picker  
 Column \_getImagePicker() {  
 return (new Column(  
 children: <Widget>[  
 Container(  
 margin: const EdgeInsets.only(bottom: 20),  
 child: Text(  
 'Cover Image',  
 style: TextStyle(fontSize: 25),  
 ),  
 ),  
 \_image == null ? new Text('No image selected.') : Image.file(\_image), // if no image is selected show Text else show the image  
 \_image == null // if no image is selected, show Choose Image button  
 ? new RaisedButton(  
 child: Text('Choose Image'),  
 onPressed: () {  
 \_getImage();  
 },  
 color: Colors.*cyan*.withOpacity(0.9),  
 shape: RoundedRectangleBorder(  
 borderRadius: new BorderRadius.circular(10.0)))  
 : Container(),  
 \_image != null // if image is selected, show Remove Button  
 ? new Container(  
 margin: const EdgeInsets.only(top: 10, bottom: 20),  
 child: RaisedButton(  
 child: Container(  
 width: 85,  
 height: 40,  
 child: Row(  
 children: <Widget>[Icon(Icons.*delete*), Text('Remove')],  
 mainAxisAlignment: MainAxisAlignment.center,  
 ),  
 ),  
 onPressed: () {  
 setState(() {  
 \_clearImage();  
 });  
 },  
 color: Colors.*red*.withOpacity(0.9),  
 shape: RoundedRectangleBorder(  
 borderRadius: new BorderRadius.circular(10.0))),  
 )  
 : Container()  
 ],  
 ));  
 }  
  
 // this function adds items in the ratings list to the Drop down menu item  
 List<DropdownMenuItem<String>> \_buildAndGetDropDownMenuItems(List ratings) {  
 List<DropdownMenuItem<String>> items = List();  
 for (String rating in ratings) {  
 items.add(DropdownMenuItem(value: rating, child: Text(rating)));  
 }  
 return items;  
 }  
  
 // this function returns a dropdown list  
 Container \_getDropDown(String label, IconData icon) {  
 return (Container(  
 margin: const EdgeInsets.only(bottom: 20),  
 child: Column(  
 children: <Widget>[  
 Container(  
 alignment: Alignment(-1, -1),  
 margin: const EdgeInsets.only(top: 30, bottom: 10, left: 35),  
 child: Text(label),  
 ),  
 Row(  
 children: <Widget>[  
 Container(  
 child: Icon(icon),  
 ),  
 Container(  
 width: 330,  
 margin: const EdgeInsets.only(left: 12),  
 child: DropdownButton(  
 value: \_selectedESRBRating,  
 items: \_dropDownMenuItems,  
 onChanged: (pickedValue) {  
 setState(() {  
 \_selectedESRBRating = pickedValue;  
 });  
 },  
 isExpanded: true,  
 ),  
 )  
 ],  
 )  
 ],  
 ),  
 ));  
 }  
}

About



about.dart

import 'package:flutter/cupertino.dart';  
import 'package:flutter/material.dart';  
import 'package:igamer/common\_ui\_widgets/appBar.dart';  
import 'package:igamer/common\_ui\_widgets/drawer.dart';  
  
// Page Title  
final String pageTitle = "About";  
  
// Main method  
void main() {  
 runApp(MaterialApp(  
 home: AboutScreenPage(),  
 theme: ThemeData(fontFamily: 'SanFrancisco'),  
 ));  
}  
  
class AboutScreenPage extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return new Scaffold(  
 body: new Stack(  
 children: <Widget>[  
 // background image  
 new Container(  
 decoration: new BoxDecoration(  
 image: new DecorationImage(  
 image: new AssetImage("assets/images/aboutus.png"),  
 fit: BoxFit.cover,  
 colorFilter: new ColorFilter.mode(  
 Colors.*black*.withOpacity(0.1), BlendMode.softLight)),  
 ),  
 ),  
 Column(  
 children: <Widget>[  
 Container(  
 margin: const EdgeInsets.only(top: 30),  
 alignment: Alignment.*center*,  
 child: Text(  
 "iGamer",  
 style: TextStyle(fontSize: 60, fontFamily: 'NunitoSans'),  
 ),  
 ),  
 Container(  
 margin: const EdgeInsets.only(top: 10),  
 alignment: Alignment.*center*,  
 child: Text(  
 "Version 2.20.211",  
 style: TextStyle(fontSize: 18, fontFamily: 'SanFrancisco'),  
 ),  
 ),  
 Container(  
 alignment: Alignment.*center*,  
 margin: const EdgeInsets.only(top: 30, bottom: 20),  
 height: 200,  
 child: Image.asset('assets/images/gamer.png'),  
 ),  
 Container(  
 margin: const EdgeInsets.only(top: 30, bottom: 20),  
 child: Text(  
 '© 2019- 2020 iTeam Inc.',  
 style: TextStyle(fontSize: 20, fontFamily: 'SanFrancisco'),  
 ),  
 ),  
 Container(  
 margin: const EdgeInsets.only(top: 20),  
 child: RaisedButton(  
 child: Text(  
 'Terms and Conditions',  
 style: TextStyle(fontSize: 18, fontFamily: 'SanFrancisco'),  
 ),  
 ),  
 ),  
 Container(  
 margin: const EdgeInsets.only(top: 10),  
 child: RaisedButton(  
 child: Text('Open source licenses',  
 style: TextStyle(fontSize: 18, fontFamily: 'SanFrancisco')),  
 ),  
 ),  
 ],  
 )  
 ],  
 ),  
 drawer: new CustomizedDrawer(context).getDrawer(),  
 appBar: new CustomizedAppBar(pageTitle).getAppBar(),  
 );  
 }  
}

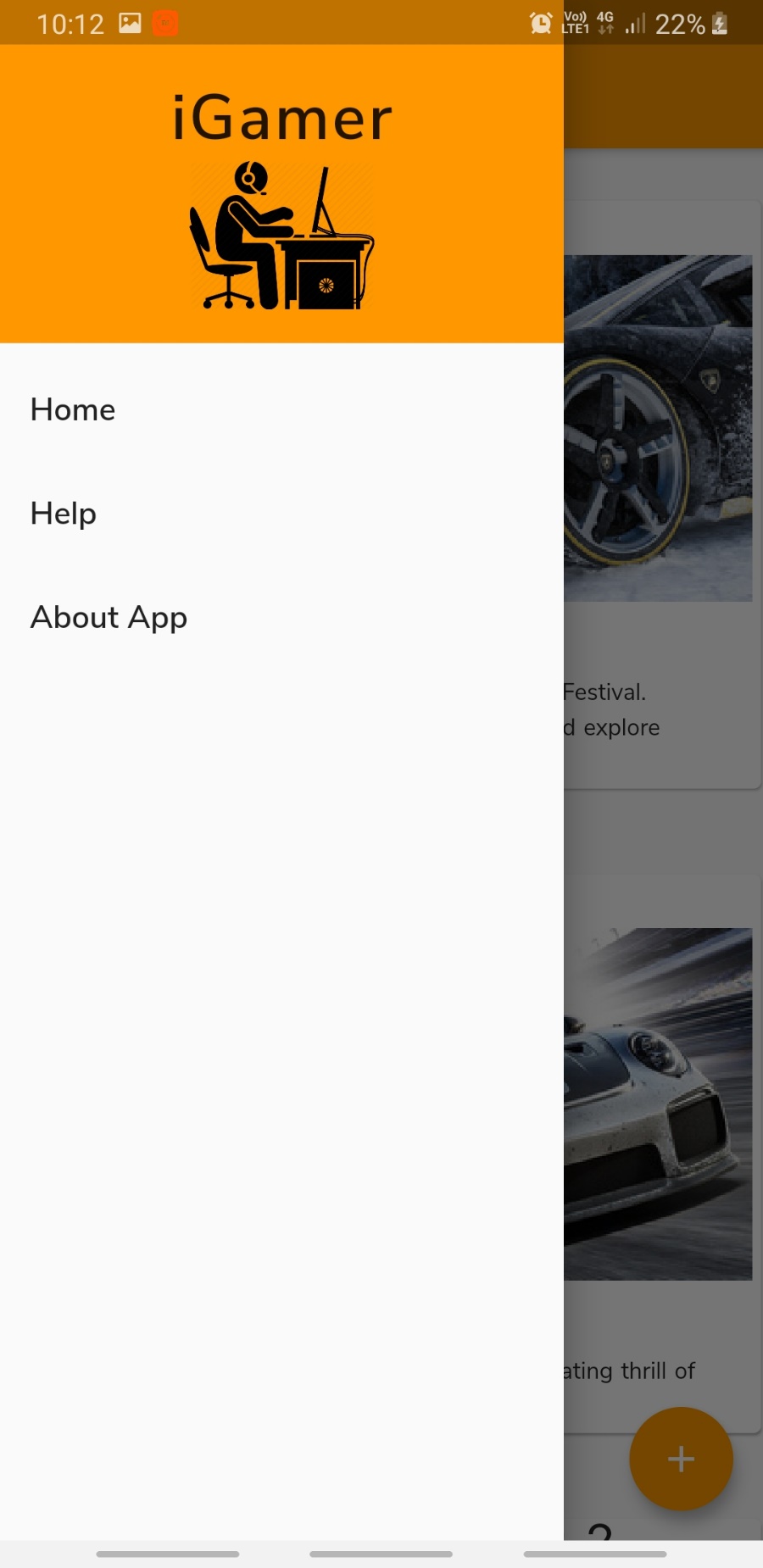
**Common UI Widgets**

**App Bar**

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



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

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

alertBox.dart

import 'package:flutter/material.dart';  
  
// this class contain attributes and method for Alert Box  
class AppAlertBox {  
 BuildContext context;  
 String title;  
 String message;  
 String buttonText;  
  
 // Constructor  
 AppAlertBox(this.context, this.title, this.message, this.buttonText);  
  
 // this function pops up Alert Box  
 showAlertDialog() {  
 Widget okButton = FlatButton(  
 child: Text(this.buttonText),  
 onPressed: () {  
 Navigator.*pop*(this.context);  
 },  
 );  
  
 // set up the AlertDialog  
 AlertDialog alert = AlertDialog(  
 title: Text(this.title),  
 content: Text(this.message),  
 actions: [  
 okButton,  
 ],  
 );  
  
 // show the AlertDialog  
 showDialog(  
 context: this.context,  
 builder: (BuildContext context) {  
 return alert;  
 },  
 );  
 }  
}

Card



gameCard.dart

import 'package:cached\_network\_image/cached\_network\_image.dart';  
import 'package:flutter/material.dart';  
import 'package:progress\_indicators/progress\_indicators.dart';  
import '../database/gameRecord.dart';  
import 'gameDetail.dart';  
  
// this class contains methods and attributes used for Card in the List in the app  
class GameCard extends StatelessWidget {  
 const GameCard({Key key, this.game, this.selected: false}) : super(key: key);  
  
 final GameRecord game;  
 final bool selected;  
  
 @override  
 Widget build(BuildContext context) {  
 TextStyle textStyle = Theme.*of*(context).textTheme.display1;  
 if (selected)  
 textStyle = textStyle.copyWith(color: Colors.*lightGreenAccent*[400]);  
 return Card(  
 margin: const EdgeInsets.only(bottom: 30),  
 color: Colors.*white*,  
 child: new InkWell(  
 onTap: () => {  
 Navigator.*push*(context,  
 MaterialPageRoute(builder: (context) => GameDetailPage(game))) // if the card is pressed navigate to Detailed Screen  
 },  
 child: Column(  
 children: <Widget>[  
 // Title  
 new Container(  
 margin: const EdgeInsets.only(left: 10),  
 child: new Text(game.title,  
 style: new TextStyle(  
 fontSize: 30, letterSpacing: 1.5, height: 1, fontFamily: 'SanFrancisco'),),  
 ),  
 // Image of the Game  
 new Container(  
// child: Image.network(game.imageLink),  
 child: CachedNetworkImage(  
 imageUrl: game.imageLink,  
 placeholder: (context, url) => Center(  
 child: Padding(  
 padding: const EdgeInsets.all(8.0),  
 child: Container(  
 margin: const EdgeInsets.only(top: 10, bottom: 10),  
 child: Column(  
 children: <Widget>[  
 Container(  
 child: JumpingDotsProgressIndicator(fontSize: 20, numberOfDots: 5, dotSpacing: 10, milliseconds: 250,),  
 ),  
 ],  
 ),  
 ),  
 ),  
 ),  
 errorWidget: (context, url, error) => Container(  
 margin: const EdgeInsets.only(top: 10, bottom: 10, left: 10, right: 10),  
 child: Center(  
 child: Image.asset("assets/images/no-image-available.png"),  
 ),  
 ),  
 ),  
 ),  
 //Published Date  
 new Container(  
 padding: const EdgeInsets.all(10.0),  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: <Widget>[  
 new Column(  
 children: <Widget>[  
 Text(game.publishedDate,  
 style: TextStyle(  
 color: Colors.*black*.withOpacity(0.5), fontFamily: 'NunitoSansBlack')),  
 ],  
 )  
 ],  
 ),  
 ),  
 // Game Description  
 Container(  
 margin: const EdgeInsets.only(left: 10, bottom: 5),  
 child: new Text(  
 game.gameDescription,  
 style: new TextStyle(wordSpacing: 1, height: 1.5, fontFamily: 'NunitoSans'),  
 ),  
 )  
 ],  
 crossAxisAlignment: CrossAxisAlignment.stretch,  
 ),  
 ));  
 }  
}

Game Detail



gameDetail.dart

import 'package:flutter/material.dart';  
import 'appBar.dart';  
import '../database/gameRecord.dart';  
import 'drawer.dart';  
  
void main() {  
 runApp(GameDetailPage(null));  
}  
  
class GameDetailPage extends StatelessWidget {  
 GameRecord game;  
  
 GameDetailPage(this.game);  
  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 title: game.title,  
 home: Scaffold(  
 appBar: new CustomizedAppBar(game.title).getAppBar(),  
 // get customized app bar  
 drawer: new CustomizedDrawer(context).getDrawer(),  
 // get customized app drawer  
 body: SingleChildScrollView(  
 child: Column(  
 children: <Widget>[  
 // Image  
 new Container(  
 padding: const EdgeInsets.all(8.0),  
 child: Image.network(game.imageLink)),  
  
 // Released Date  
 getDetailRow("Released On", game.releaseDate),  
  
 // Full Description  
 new Container(  
 margin: const EdgeInsets.only(left: 10, bottom: 5, right: 10),  
 child: new Text(  
 game.fullDescription,  
 style: new TextStyle(  
 fontSize: 16, height: 1.5, fontFamily: 'SanFrancisco'),  
 textAlign: TextAlign.justify,  
 ),  
 ),  
  
 // Genre  
 getDetailRow("Genre", game.genre),  
  
 // Developer  
 getDetailRow("Developer", game.developer),  
  
 // ESRB Rating  
 getDetailRow("ESRB Rating", game.esrbRating),  
  
 // User Score  
 getDetailRowHorizontal("User Score", game.userScore),  
  
 // No of Users  
 getDetailRowHorizontal("No of Users", game.noOfUsers),  
 ],  
 crossAxisAlignment: CrossAxisAlignment.start,  
 ))),  
 );  
 }  
  
 // this function returns a Row in which the label is on top and the value is at bottom  
 Widget getDetailRow(String label, String value) {  
 return new Container(  
 padding: const EdgeInsets.all(10.0),  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
 crossAxisAlignment: CrossAxisAlignment.start,  
 children: <Widget>[  
 new Container(  
 child: new Text(  
 label,  
 style: new TextStyle(fontSize: 18, fontFamily: 'NunitoSans'),  
 ),  
 margin: const EdgeInsets.only(bottom: 5),  
 ),  
 new Column(  
 children: <Widget>[  
 Text(value,  
 style: TextStyle(  
 color: Colors.*black*.withOpacity(0.5),  
 fontSize: 18,  
 fontFamily: 'SanFrancisco'))  
 ],  
 )  
 ],  
 ),  
 );  
 }  
  
 // this function returns a Row in which the value is place next to the label  
 Widget getDetailRowHorizontal(String label, String value) {  
 return new Container(  
 padding: const EdgeInsets.all(10.0),  
 child: Row(  
 mainAxisAlignment: MainAxisAlignment.start,  
 crossAxisAlignment: CrossAxisAlignment.center,  
 children: <Widget>[  
 new Container(  
 child: new Text(  
 label,  
 style: new TextStyle(fontSize: 18, fontFamily: 'NunitoSans'),  
 ),  
 margin: const EdgeInsets.only(right: 8),  
 ),  
 new Column(  
 children: <Widget>[  
 Text(value,  
 style: TextStyle(  
 color: Colors.*black*.withOpacity(0.5),  
 fontSize: 38,  
 fontFamily: 'SanFrancisco'))  
 ],  
 )  
 ],  
 ),  
 );  
 }  
}

**Database**

crud.dart

import 'package:cloud\_firestore/cloud\_firestore.dart';  
import 'package:igamer/database/gameRecord.dart';  
  
// gameID|this.title|publishedDate|gameDescription|imageLink|genre|developer|releaseDate|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>";  
}