# Приложение В.Текст программы

# АННОТАЦИЯ

В данном программном документе, в разделе «Текст программы» указан текст программы и его описание.

СОДЕРЖАНИЕ

[Приложение В.Текст программы 1](#_Toc73289910)

[АННОТАЦИЯ 1](#_Toc73289911)

[1. ТЕКСТ ПРОГРАММЫ 3](#_Toc73289912)

[1.1. Наименование программы 3](#_Toc73289913)

[1.2. Область применения программы 3](#_Toc73289914)

[1.3. Модули 3](#_Toc73289915)

[1.4. Код программы 4](#_Toc73289916)

# ТЕКСТ ПРОГРАММЫ

## Наименование программы

Наименование - «Мобильное приложения по продаже товаров и услуг DP Stuff Provider».

## Область применения программы

Данная программа предназначена для продажи товаров и услуг.

## Модули

Таблица 1 - Модули программы

| № | Модуль | Описание | Кол-во строк | Размер в КБ |
| --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 |
| 1 | AndroidManifest.xml | Нужен для хранения архитектуры Андроид-приложения | 40 |  |
| 2 | CustomRecyclerAdapter.kt | Логика RecyclerView | 56 | 3 |
| 3 | APInterface.kt | Интерфейс для запросов API | 19 | 1 |
| 4 | UrlModel.kt | Модель для получения данных с API | 27 | 1 |
| 5 | AutorizationMainActivity.kt | Класс для работы с авторизацией пользователя | 73 | 3 |
| 6 | ForFragmentMainActivity.kt | Класс для работы с фрагментами карты и аккаунта пользователя | 67 | 3 |
| 7 | MainActivityDetailEvent.kt | Класс для работы с подробным просмотром информации о мероприятий | 157 | 7 |
| 8 | MainActivityRec.kt | Класс для заполнения списка мероприятий | 133 | 5 |
| 9 | RegistrationMainActivity.kt | Класс для регистрации пользователя | 59 | 3 |
| 10 | MapBlankFragment.kt | Фрагмент на котором запускается карта | 305 | 12 |
| 11 | PessonalAccountBlankFragment.kt | Фрагмент с личными данными о пользователе | 108 | 4 |
| 12 | activity\_autorization\_main.xml | Дизайн для окна авторизации | 66 | 2 |
| 13 | activity\_for\_fragment\_main.xml | Дизайн для окна на котором будут запускаться фрагменты | 25 | 1 |
| 14 | activity\_main\_detail\_event.xml | Дизайн для окна подробного просмотра мероприятия | 132 | 5 |
| 15 | activity\_main\_rec.xml | Дизайн для окна со списком мероприятий | 40 | 3 |
| 16 | activity\_registration\_main.xml | Дизайн для окна с регистрацией | 56 | 3 |
| 17 | fragment\_map\_blank.xml | Дизайн для фрагмента с отображением карты | 52 | 2 |
| 18 | fragment\_pessonal\_account\_blank.xml | Дизайн для фрагмента персонального аккаунта | 84 | 5 |
| 19 | recyclerview\_item.xml | Дизайн для одного элемента списка мероприятий | 40 | 2 |
| 20 | menu.xml | Дизайн для кнопки навигации | 24 | 1 |
| 21 | google\_maps\_api.xml | Xml файл для подключения к GoogleMapAPI | 12 | 1 |
| 22 | strings.xml | Xml файл который хранит стрококвые значения | 26 | 1 |
| 23 | style.xml | Xml файл который хранит стили для разных строк в приложении | 62 | 4 |
| 24 | EventController.cs | Файл отвечающий за формирование запросов API | 133 | 5 |

## Код программы

1) AndroidManifest.xml

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.placeofrest">  
  
 <uses-permission android:name="android.permission.INTERNET" />  
 <uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />  
 <uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/qwe"  
 android:label="KudaToDay"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.PlaceOfRest">  
 <activity android:name=".MainActivityDetailEvent" android:screenOrientation="portrait"></activity>  
 <activity android:name=".RegistrationMainActivity" android:screenOrientation="portrait" />  
 <activity android:name=".MainActivityRec" android:screenOrientation="portrait" />  
 <activity android:name=".AutorizationMainActivity" android:screenOrientation="portrait" />  
 <activity  
 android:name=".ForFragmentMainActivity"  
 android:screenOrientation="portrait">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
  
 <meta-data  
 android:name="com.google.android.geo.API\_KEY"  
 android:value="@string/google\_maps\_key" />  
  
 <activity  
 android:name=".MapsActivity"  
 android:label="@string/title\_activity\_maps"  
 android:screenOrientation="portrait" />  
 </application>  
  
</manifest>

2) CustomRecyclerAdapter.kt

package com.example.placeofrest.classForRecyclerViewAndCardView  
  
import android.content.Intent  
import android.os.Bundle  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.ImageView  
import android.widget.TextView  
import android.widget.Toast  
import androidx.cardview.widget.CardView  
import androidx.core.content.ContextCompat.startActivity  
import androidx.recyclerview.widget.RecyclerView  
import com.example.placeofrest.MainActivityDetailEvent  
import com.example.placeofrest.R  
import com.squareup.picasso.Picasso  
  
class CustomRecyclerAdapter(  
 //Листы, с которыми будет рабобать адаптер  
 private val titles: List<String>, private val imageLink:List<String>, private val id:List<String>) :  
 RecyclerView.Adapter<CustomRecyclerAdapter.MyViewHolder>() {  
  
 class MyViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {  
 var imageView: ImageView? = null  
 var titleTextVeiw: TextView? = null  
 var cardView: CardView? = null  
  
 init {  
 imageView = itemView.findViewById(R.id.*imageRec*)  
 titleTextVeiw = itemView.findViewById(R.id.*textViewTitle*)  
 cardView = itemView.findViewById(R.id.*cardViewRec*)  
 }  
 }  
  
 override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): MyViewHolder {  
 val itemView =  
 LayoutInflater.from(parent.*context*)  
 .inflate(R.layout.*recyclerview\_item*, parent, false)  
 return MyViewHolder(itemView)  
 }  
  
 //Основной функционал для работы с элементами списка в ресайклерВью  
 override fun onBindViewHolder(holder: MyViewHolder, position: Int) {  
 Picasso.get().load(imageLink[position]).into(holder.imageView)  
 holder.titleTextVeiw?.*text* = titles[position]  
  
  
 holder.cardView?.setOnClickListener**{** val intent : Intent = Intent(holder.itemView.*context*, MainActivityDetailEvent::class.*java*)  
  
 intent.putExtra("idPEX", id[position])  
  
 startActivity(holder.itemView.*context*, intent, Bundle.*EMPTY*)  
 **}** }  
  
 override fun getItemCount() = titles.size  
}

3) APInterface.kt

package com.example.placeofrest.response  
  
import retrofit2.http.GET  
import retrofit2.http.Query  
import retrofit2.Call  
  
const val *key1* = "AIzaSyB\_1syNZPbX04aOhLkCtWIOV81j7zKW2tU"  
//интерфейс, созданный для подключения к апи, для получения данных о времени на путь  
interface APInterface {  
  
 @GET("/maps/api/directions/json")  
 fun getRoute(  
 @Query ("origin") startlocation: String,  
 @Query ("destination") endlocation: String,  
 @Query ("sensor") sensor: Boolean,  
 @Query ("mode") mode: String,  
 @Query ("key") key: String = *key1* ): Call<UrlModel>  
}

4) UrlModel.kt

package com.example.placeofrest.response  
  
import com.google.gson.annotations.Expose  
import com.google.gson.annotations.SerializedName  
//обработка данных и вынесение определенных данных из запроса к API  
class UrlModel {  
 @SerializedName("routes")  
 @Expose  
 var routes = ArrayList<Routes>()  
 @SerializedName("status")  
 @Expose var status : String? = null  
}  
  
class Routes {  
 @SerializedName("legs")  
 @Expose var legs = ArrayList<Legs>()  
}  
  
class Legs {  
 @SerializedName("duration")  
 @Expose var duration : Duration? = null  
}  
  
class Duration {  
 @SerializedName("text")  
 @Expose var text: String? = null  
}

5) AutorizationMainActivity.kt

package com.example.placeofrest  
  
import android.content.Context  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.preference.PreferenceManager  
import android.util.Log  
import android.widget.Toast  
import androidx.core.content.ContentProviderCompat.requireContext  
import com.google.firebase.auth.FirebaseAuth  
import com.google.firebase.auth.FirebaseUser  
import kotlinx.android.synthetic.main.activity\_autorization\_main.\*  
  
  
class AutorizationMainActivity : AppCompatActivity() {  
  
 private lateinit var auth: FirebaseAuth  
  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_autorization\_main*)  
  
 auth = FirebaseAuth.getInstance()  
  
 buttonAutorization.setOnClickListener **{** saveData()  
 signInUser()  
 //loadData()  
 **}** buttonReg.setOnClickListener **{** startActivity(Intent(this,RegistrationMainActivity::class.*java*))  
  
 **}** }  
 //проверка, заполнены ли поля для авторизации  
 private fun signInUser() {  
 if (editTextLogin.*text*.toString().*isNullOrEmpty*() || editTextPassword.*text*.toString().*isNullOrEmpty*()) {  
 Toast.makeText(*baseContext*, "Одно из полей не заполнено", Toast.*LENGTH\_LONG*).show()  
 }  
 else {  
 auth.signInWithEmailAndPassword(editTextLogin.*text*.toString(), editTextLogin.*text*.toString())  
 .addOnCompleteListener(this) **{** task **->** if (task.*isSuccessful*) {  
 val user = auth.*currentUser* updateUI(user, editTextLogin.*text*.toString())  
 } else Toast.makeText(*baseContext*, "Ошибка", Toast.*LENGTH\_LONG*).show()  
 **}** }  
 }  
 //проверка того, подтвержден ли адресс электонной почты  
 private fun updateUI(currentUser: FirebaseUser?, emailAdd: String) {  
 if (currentUser != null){  
 if (currentUser.*isEmailVerified*){  
 Toast.makeText(*baseContext*, "Пользователь авторизован", Toast.*LENGTH\_LONG*).show()  
 startActivity(Intent(this,ForFragmentMainActivity::class.*java*))  
 }  
 else Toast.makeText(*baseContext*, "Адрес электронной почты не подтверждён", Toast.*LENGTH\_LONG*).show()  
 }  
 }  
 //сохраннение данных о пользователе, если он авторизирован  
 private fun saveData() {  
 val mainActivity = this as AutorizationMainActivity  
 val insertedText = editTextLogin.*text*.toString()  
 val sharedPreferences = mainActivity.getSharedPreferences("sharedPrefs", Context.*MODE\_PRIVATE*)  
 val editor = sharedPreferences.edit()  
 editor.*apply* **{** putString("string\_KEY", insertedText.toString())  
 **}**.apply()  
 }  
}

6) ForFragmentMainActivity.kt

package com.example.placeofrest  
  
import android.content.pm.PackageManager  
import android.location.Location  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.view.View  
import android.widget.Toast  
import androidx.core.app.ActivityCompat  
import androidx.fragment.app.Fragment  
import androidx.navigation.Navigation  
import androidx.navigation.ui.NavigationUI  
import com.example.placeofrest.fragments.MapBlankFragment  
import com.example.placeofrest.fragments.PessonalAccountBlankFragment  
import com.example.placeofrest.fragments.EventFeedBlankFragment  
import com.google.android.gms.location.FusedLocationProviderClient  
import com.google.android.gms.location.LocationServices  
import com.google.android.gms.maps.CameraUpdateFactory  
import com.google.android.gms.maps.GoogleMap  
import com.google.android.gms.maps.OnMapReadyCallback  
import com.google.android.gms.maps.SupportMapFragment  
import com.google.android.gms.maps.model.LatLng  
import com.google.android.gms.maps.model.MarkerOptions  
import kotlinx.android.synthetic.main.activity\_for\_fragment\_main.\*  
import kotlinx.coroutines.CoroutineScope  
import kotlinx.coroutines.Dispatchers  
import kotlinx.coroutines.launch  
  
//Окно, которое является для родителем ля фрагментов, так же на нам размещен кнопка навигации между фрагментами  
class ForFragmentMainActivity : AppCompatActivity(), OnMapReadyCallback {  
  
 private lateinit var currentLocation: Location  
 private lateinit var fusedLocationProviderClient: FusedLocationProviderClient  
 private var permissionCode = 101  
  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_for\_fragment\_main*)  
  
 val mapBlank = MapBlankFragment()  
 val eventFeedBlank = EventFeedBlankFragment()  
 val pessonalAccountBlank = PessonalAccountBlankFragment()  
  
 //загрузка фрагментов по нажатию на определенные елементы на кнопке навигации  
 makeCurrentFragment(mapBlank)  
 bottomNavigationView.setOnNavigationItemSelectedListener **{** when (**it**.*itemId*) {  
 R.id.*ic\_map* -> makeCurrentFragment(mapBlank)  
 R.id.*ic\_home* -> makeCurrentFragment(pessonalAccountBlank)  
 }  
 true  
 **}** }  
  
 //функция погрузки и фрагментов  
 private fun makeCurrentFragment(fragment: Fragment) =  
 *supportFragmentManager*.beginTransaction().*apply* **{** replace(R.id.*fl\_wrapper*, fragment)  
 commit()  
 **}** override fun onMapReady(p0: GoogleMap?) {  
 *TODO*("Not yet implemented")  
 }  
}

7) MainActivityDetailEvent.kt

package com.example.placeofrest  
  
import android.content.Context  
import android.content.Intent  
import android.net.Uri  
import android.os.Bundle  
import android.util.Log  
import androidx.appcompat.app.AppCompatActivity  
import com.example.placeofrest.response.APInterface  
import com.example.placeofrest.response.UrlModel  
import com.squareup.picasso.Picasso  
import kotlinx.android.synthetic.main.activity\_main\_detail\_event.\*  
import okhttp3.\*  
import org.json.JSONArray  
import retrofit2.Retrofit  
import retrofit2.Call  
import retrofit2.Callback  
import retrofit2.Response  
import retrofit2.converter.gson.GsonConverterFactory  
import java.io.IOException  
  
  
import java.text.SimpleDateFormat  
  
//окно на котором открывается подробное описание мероприятия  
class MainActivityDetailEvent : AppCompatActivity() {  
  
 var imageView = ""  
  
 var timepath = ""  
  
 var coordinatelatitude: Double = 0.0  
 var coordinatelongitude: Double = 0.0  
  
 var events: String = "http://www.google.com"  
  
 var buildFull: String = ""  
  
 public var id = ""  
 var URL = "https://apiplaceofrestproject.azurewebsites.net/api/event/CatchIDEvent/id/"  
 var URL2 = ""  
 var okHttpClient: OkHttpClient = OkHttpClient()  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.activity\_main\_detail\_event)  
  
 id = intent.getStringExtra("idPEX")!!  
  
  
 URL2 = "${URL}"+"${id}"  
  
 val request: Request = Request.Builder().url(URL2).build()  
 okHttpClient.newCall(request).enqueue(object : okhttp3.Callback {  
  
 override fun onFailure(call: okhttp3.Call, e: IOException) {  
 Log.e("Error", e.toString())  
 }  
  
 override fun onResponse(call: okhttp3.Call?, response: okhttp3.Response?) {  
 val jo = JSONArray(response!!.body()!!.string())  
 if(jo!=null){  
 var i = 0  
 var tr = true  
 //получение данных из запроса к API, заполнение листов определенных данных  
 while(tr){  
 try {  
 val image = jo.getJSONObject(i).get("picture").toString()  
 val title = jo.getJSONObject(i).get("name").toString()  
 val date = jo.getJSONObject(i).get("date").toString()  
 val discription = jo.getJSONObject(i).get("discription").toString()  
 val street = jo.getJSONObject(i).get("street").toString()  
 val build = jo.getJSONObject(i).get("build").toString()  
 val structur = jo.getJSONObject(i).get("structur").toString()  
 val metro = jo.getJSONObject(i).get("metro").toString()  
 val latitude = jo.getJSONObject(i).get("coordinatee").toString()  
 val longitude = jo.getJSONObject(i).get("longitude").toString()  
 val event = jo.getJSONObject(i).get("eventev").toString()  
  
 i++  
  
 runOnUiThread()**{** imageView = image  
 Picasso.get().load(image).into(imageViewDetail)  
 textViewTitleDetail.text = title  
 textViewDescriptionDetail.text = discription  
 if(structur!="0"){  
 buildFull = "ул. "+"${street}"+" "+" Дом "+"${build}"+" строение "+"${structur}"  
 }else{  
 buildFull = "ул. "+"${street}"+" "+" Дом "+"${build}"  
 }  
 textViewCoordinateBuildAndMetroDetail.text = buildFull  
 textViewMetro.text = "Метро: "+"${metro}"  
 coordinatelatitude = latitude.toDouble()  
 coordinatelongitude = longitude.toDouble()  
  
 events = event  
  
 var dateDate = ""  
 var dateTime = ""  
 var dateInText = ""  
 dateDate = date.substringBefore("T")  
  
 dateInText = "${dateDate.substring(8,10)}"+"-"+"${dateDate.substring(5,7)}"+"-"+"${dateDate.substring(0,4)}"  
  
  
 dateTime = date.substringAfter("T")  
 dateTime = dateTime.substring(0, 5)  
 if(dateTime.substring(0, 1).toString() == "0"){  
 textViewDescriptionTime.text = "Дата: "+"${dateInText.toString()}"+" Время: круглые сутки"  
 }  
 else{  
 textViewDescriptionTime.text = "Дата: "+"${dateInText}"+" Время: "+"${dateTime.toString()}"  
 }  
 **}** }catch(e: Exception){  
 tr = false  
 }  
 //response()  
 }  
 }  
 }  
 })  
  
 buttonForMap.setOnClickListener **{** saveData()  
 startActivity(Intent(this,ForFragmentMainActivity::class.java))  
 **}** textViewLink.setOnClickListener**{** val browserIntent = Intent(Intent.ACTION\_VIEW, Uri.parse("${events}")); startActivity(browserIntent)  
 startActivity(browserIntent)  
 **}** }  
 //сохранение данных о координатах мероприятия  
 private fun saveData() {  
 val mainActivity = this as MainActivityDetailEvent  
 val insertedText = coordinatelatitude.toString()  
 val insertedText2 = coordinatelongitude.toString()  
  
 val sharedPreferences = mainActivity.getSharedPreferences("sharedPrefsCor", Context.MODE\_PRIVATE)  
 val editor = sharedPreferences.edit()  
 editor.apply **{** putString("string\_KEYCor1", insertedText.toString())  
 putString("string\_KEYCor2", insertedText2.toString())  
 **}**.apply()  
  
 Log.i("myLog", insertedText)  
 Log.i("myLog", insertedText2)  
 }  
}

8) MainActivityRec.kt

package com.example.placeofrest  
  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.util.Log  
import android.widget.Toast  
import androidx.recyclerview.widget.LinearLayoutManager  
import androidx.recyclerview.widget.RecyclerView  
import com.example.placeofrest.classForRecyclerViewAndCardView.CustomRecyclerAdapter  
import kotlinx.android.synthetic.main.activity\_main\_rec.\*  
import okhttp3.\*  
import org.json.JSONArray  
import java.io.IOException  
import java.lang.Exception  
  
//окно на котором заполняется лента с мероприятиями  
class MainActivityRec : AppCompatActivity() {  
  
 var nameList = mutableListOf<String>()  
 var themeList = mutableListOf<String>()  
 var imageList = mutableListOf<String>()  
 var idList = mutableListOf<String>()  
  
 val URL = "https://apiplaceofrestproject.azurewebsites.net/api/event/getevents"  
 var URLCatch = "https://apiplaceofrestproject.azurewebsites.net/api/event/SearchByEvents/theme/"  
 var okHttpClient: OkHttpClient = OkHttpClient()  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.activity\_main\_rec)  
  
 val recyclerView: RecyclerView = findViewById(R.id.recyclerView)  
 recyclerView.layoutManager = LinearLayoutManager(this)  
  
 val request: Request = Request.Builder().url(URL).build()  
 okHttpClient.newCall(request).enqueue(object : Callback {  
  
 override fun onFailure(call: Call, e: IOException) {  
 Log.e("Error", e.toString())  
 }  
 //заполнение листа с любыми тематиками  
 override fun onResponse(call: Call?, response: Response?) {  
 val jo = JSONArray(response!!.body()!!.string())  
 if(jo!=null){  
 var i = 0  
 var tr = true  
  
 while(tr){  
 try {  
 val name = jo.getJSONObject(i).get("name").toString()  
 val image = jo.getJSONObject(i).get("picture").toString()  
 val id = jo.getJSONObject(i).get("id").toString()  
 val theme = jo.getJSONObject(i).get("theme").toString()  
  
  
 nameList.add(name)  
 imageList.add(image)  
 themeList.add(theme)  
 idList.add(id)  
  
 i++  
  
 runOnUiThread()**{** recyclerView.adapter = CustomRecyclerAdapter(nameList, imageList, idList)  
 **}** }catch(e: Exception){  
 tr = false  
 }  
 }  
 }  
 }  
 })  
  
 textInputLayout.setEndIconOnClickListener**{** nameList.clear()  
 imageList.clear()  
 var URLCatch2 = ""  
  
 if(textInputEditText.text.toString() ==""){  
 URLCatch2 = "${URL}"  
 }else{  
 URLCatch2 = "${URLCatch}"+"${textInputEditText.text.toString()}"  
 }  
  
 var request: Request  
 request = Request.Builder().url(URLCatch2).build()  
  
 okHttpClient.newCall(request).enqueue(object : Callback {  
  
 override fun onFailure(call: Call, e: IOException) {  
 Log.e("Error", e.toString())  
 }  
 //заполнение листа с определенной тематикой  
 override fun onResponse(call: Call?, response: Response?) {  
 val jo = JSONArray(response!!.body()!!.string())  
 if(jo!=null){  
 var i = 0  
 var tr = true  
  
 while(tr){  
 try {  
 val name = jo.getJSONObject(i).get("name").toString()  
 val image = jo.getJSONObject(i).get("picture").toString()  
  
 nameList.add(name)  
 imageList.add(image)  
  
 i++  
  
 runOnUiThread()**{** val theme = textInputEditText.text  
 recyclerView.adapter = CustomRecyclerAdapter(nameList, imageList, idList)  
 **}** }catch(e: Exception){  
 tr = false  
 }  
 }  
 }  
 }  
 })  
 URLCatch2 = URLCatch  
 **}** }  
  
 private fun fillList(): List<String> {  
 val data = mutableListOf<String>()  
 (0..30).forEach **{** i **->** data.add("\$i element") **}** return data  
 }  
}

9) RegistrationMainActivity.kt

package com.example.placeofrest  
  
import android.content.Context  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.util.Log  
import android.widget.Toast  
import com.google.firebase.auth.FirebaseAuth  
import com.google.firebase.auth.FirebaseUser  
import kotlinx.android.synthetic.main.activity\_autorization\_main.\*  
import kotlinx.android.synthetic.main.activity\_registration\_main.\*  
  
class RegistrationMainActivity : AppCompatActivity() {  
  
 private lateinit var auth: FirebaseAuth  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_registration\_main*)  
  
 auth = FirebaseAuth.getInstance()  
  
 buttonRegis.setOnClickListener **{** signUpUser()  
 **}** }  
 //проверка на заполнение полей для регистрации  
 private fun signUpUser() {  
 if (false) {  
 Toast.makeText(*baseContext*, "Одно из полей не заполнено", Toast.*LENGTH\_LONG*).show()  
 }  
 else {  
 auth.createUserWithEmailAndPassword(editTextLoginReg.*text*.toString(), editTextPasswordReg.*text*.toString())  
 .addOnCompleteListener(this) **{** task **->** if (task.*isSuccessful*) {  
 auth.*currentUser*?.sendEmailVerification()  
 ?.addOnCompleteListener **{** task **->** if (task.*isSuccessful*) {  
 Toast.makeText(*baseContext*, "Подтвердите адрес электронной почты", Toast.*LENGTH\_LONG*).show()  
 startActivity(Intent(this,ForFragmentMainActivity::class.*java*))  
 } else Toast.makeText(*baseContext*, "Ошибка", Toast.*LENGTH\_LONG*).show()  
 **}** }  
 **}** }  
 }  
 private fun loadData() {  
 val sharedPreferences = getSharedPreferences("sharedPrefs", Context.*MODE\_PRIVATE*)  
 val savedString = sharedPreferences.getString("string\_KEY", null)  
 val savedString1 = sharedPreferences.getString("string1\_KEY", null)  
 var q = savedString!!  
 Log.i("myLog", q)  
 }  
}

10) MapBlankFragment.kt

package com.example.placeofrest.fragments  
  
import android.app.Activity  
import android.content.Context  
import android.content.Intent  
import android.content.pm.PackageManager  
import android.location.Location  
import android.os.Bundle  
import android.util.Log  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.Toast  
import androidx.core.app.ActivityCompat  
import androidx.core.os.bundleOf  
import androidx.core.view.isVisible  
import androidx.fragment.app.Fragment  
import androidx.fragment.app.FragmentActivity  
import com.example.placeofrest.\*  
import com.example.placeofrest.response.APInterface  
import com.example.placeofrest.response.UrlModel  
import com.google.android.gms.location.FusedLocationProviderClient  
import com.google.android.gms.location.LocationServices  
import com.google.android.gms.maps.CameraUpdateFactory  
import com.google.android.gms.maps.GoogleMap  
import com.google.android.gms.maps.OnMapReadyCallback  
import com.google.android.gms.maps.SupportMapFragment  
import com.google.android.gms.maps.model.BitmapDescriptor  
import com.google.android.gms.maps.model.BitmapDescriptorFactory  
import com.google.android.gms.maps.model.LatLng  
import com.google.android.gms.maps.model.MarkerOptions  
import kotlinx.android.synthetic.main.fragment\_map\_blank.\*  
import kotlinx.android.synthetic.main.recyclerview\_item.\*  
import kotlinx.coroutines.GlobalScope  
import kotlinx.coroutines.launch  
import okhttp3.\*  
import org.json.JSONArray  
import retrofit2.Retrofit  
import retrofit2.converter.gson.GsonConverterFactory  
import java.io.IOException  
import retrofit2.Call  
import retrofit2.Callback  
import retrofit2.Response  
  
  
  
class MapBlankFragment : Fragment(), OnMapReadyCallback {  
  
 private lateinit var currentLocation: Location  
 private lateinit var fusedLocationProviderClient: FusedLocationProviderClient  
 private var permissionCode = 101  
  
 private lateinit var intent: Intent  
  
 var timeWay = "q1"  
 var timeWaySettings = ""  
  
 private var myContext: FragmentActivity? = null  
  
 var tf = false  
  
 var coordinatelatitude: String = ""  
 var coordinatelongitude: String = ""  
 var nameCor:String= ""  
  
 private lateinit var map1: GoogleMap  
  
 var nameList = mutableListOf<String>()  
 var latitudeList = mutableListOf<String>()  
 var longitudeList = mutableListOf<String>()  
  
 var cr1=""  
 var cr2=""  
  
 val URL = "https://apiplaceofrestproject.azurewebsites.net/api/event/getmapmarker"  
 var okHttpClient: OkHttpClient = OkHttpClient()  
  
 public var eventlatitude = 0.0  
 public var eventlongitude = 0.0  
  
  
 override fun onCreateView(  
 inflater: LayoutInflater, container: ViewGroup?,  
 savedInstanceState: Bundle?  
 ): View? {  
 cr1 = bundleOf().getString("cr1").toString()  
 cr2 = bundleOf().getString("cr2").toString()  
  
 return inflater.inflate(R.layout.fragment\_map\_blank, container, false)  
 }  
  
 //функция в которой прописывается основной функционал на фрагменте  
 override fun onViewCreated(view: View, savedInstanceState: Bundle?) {  
 super.onViewCreated(view, savedInstanceState)  
  
 intent = requireActivity().intent!!  
  
 textViewTrans.isVisible = false  
  
 fusedLocationProviderClient = LocationServices.getFusedLocationProviderClient(requireActivity())  
 fetchLocation()  
  
 //нажатие на кнопку для открытия активити, на котором будет лента с мероприятиями  
 buttonevent.setOnClickListener **{** startActivity(  
 android.content.Intent(  
 activity,  
 MainActivityRec::class.java  
 )  
 )  
 **}** //okhht запрос для обращения к API  
 val request: Request = Request.Builder().url(URL).build()  
 okHttpClient.newCall(request).enqueue(object : okhttp3.Callback{  
 //метод вызываемый при ошибке к подключению  
 override fun onFailure(call: okhttp3.Call, e: IOException) {  
 Log.e("Error", e.toString())  
 }  
 //метод положительного запроса к обращению запроса  
 override fun onResponse(call: okhttp3.Call?, response: okhttp3.Response?) {  
 val jo = JSONArray(response!!.body()!!.string())  
 if(jo!=null){  
 var i = 0  
 var tr = true  
 //цикл заполнения листа мерприятиями  
 while(tr){  
 try {  
 val name = jo.getJSONObject(i).get("name").toString()  
 val latitude = jo.getJSONObject(i).get("coordinatee").toString()  
 val longitude = jo.getJSONObject(i).get("longitude").toString()  
  
 nameList.add(name)  
 latitudeList.add(latitude)  
 longitudeList.add(longitude)  
  
 i++  
  
 requireActivity().runOnUiThread()**{** //Toast.makeText(requireActivity(), "${nameList[0].toString()}", Toast.LENGTH\_SHORT).show()  
  
 **}** }catch(e: Exception){  
 tr = false  
 }  
 }  
 }  
 }  
 })  
 }  
  
 //открытие и прогрузка карты  
 private fun fetchLocation(){  
 if(ActivityCompat.checkSelfPermission(requireContext(), android.Manifest.permission.ACCESS\_FINE\_LOCATION)!= PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission(requireContext(), android.Manifest.permission.ACCESS\_COARSE\_LOCATION)!= PackageManager.PERMISSION\_GRANTED){  
 ActivityCompat.requestPermissions(requireActivity(), arrayOf(android.Manifest.permission.ACCESS\_FINE\_LOCATION), permissionCode)  
 return  
 }  
 val task = fusedLocationProviderClient.lastLocation  
 task.addOnSuccessListener **{** location **->** if (location != null){  
 currentLocation = location  
 //Toast.makeText(requireActivity(), currentLocation.latitude.toString() + " " + currentLocation.longitude, Toast.LENGTH\_SHORT).show()  
 val mapFragment = childFragmentManager.findFragmentById(R.id.map) as SupportMapFragment?  
 mapFragment?.getMapAsync(this)  
 }  
 **}** }  
  
 //метод для работы с картой, заполнением ее элементами и выставлением маркеров  
 override fun onMapReady(googleMap: GoogleMap) {  
 var latLng:LatLng = LatLng(currentLocation.latitude, currentLocation.longitude)  
 var markerOptions = MarkerOptions().position(latLng)  
 .title("Я")  
 .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE\_AZURE))  
 //.icon(BitmapDescriptorFactory.fromResource(R.drawable.q))  
 googleMap?.animateCamera(CameraUpdateFactory.newLatLng(latLng))  
 googleMap?.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng, 10f))  
 googleMap?.addMarker(markerOptions)  
  
 loadData()  
 if(coordinatelatitude.toString() != "") {  
 var latLngLL: LatLng = LatLng(coordinatelatitude.toString().toDouble(), coordinatelongitude.toString().toDouble())  
 var markerOptionsLL = MarkerOptions().position(latLngLL).title("${nameCor}")  
 googleMap?.addMarker(markerOptionsLL)  
 saveData()  
  
 var cr1 = latLng.toString()  
 var cr2 = latLngLL.toString()  
 cr1= cr1.substring(10, cr1.length -1)  
 cr2= cr2.substring(10, cr2.length -1)  
 Log.i("myLog", cr1)  
 Log.i("myLog", cr2)  
  
 GlobalScope.launch **{** response(cr1, cr2)  
 **}** }  
  
 var j = 0  
 while (j < latitudeList.size) {  
 var m = LatLng(latitudeList[j].toDouble(), longitudeList[j].toDouble());  
 var marker = MarkerOptions().position(m).title("${nameList[j]}")  
 googleMap?.addMarker(marker)  
 //Toast.makeText(context, "qwe", Toast.LENGTH\_SHORT).show()  
 j++  
 }  
 j=0  
 }  
  
 override fun onRequestPermissionsResult(  
 requestCode: Int,  
 permissions: Array<String?>,  
 grantResults: IntArray  
 ) {  
 when(requestCode){  
 permissionCode -> if(grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERMISSION\_GRANTED){  
 fetchLocation()  
 }  
 }  
 }  
  
 override fun onAttach(activity: Activity) {  
 myContext = activity as FragmentActivity  
 super.onAttach(activity)  
 }  
  
 //функция прогрузки данных о метоположении вибранного мероприятя для выставления марке на карте  
 private fun loadData() {  
 val mainActivity = this.activity as ForFragmentMainActivity  
 val sharedPreferences= mainActivity.getSharedPreferences("sharedPrefsCor", Context.MODE\_PRIVATE)  
 val savedString1 = sharedPreferences.getString("string\_KEYCor1", "")  
 val savedString2 = sharedPreferences.getString("string\_KEYCor2", "")  
 coordinatelatitude = savedString1.toString()  
 coordinatelongitude = savedString2.toString()  
 }  
  
 //обнуление данных  
 private fun saveData() {  
 val mainActivity = this.activity as ForFragmentMainActivity  
 val insertedText = ""  
 val insertedText2 = ""  
  
 val sharedPreferences = mainActivity.getSharedPreferences("sharedPrefsCor", Context.MODE\_PRIVATE)  
 val editor = sharedPreferences.edit()  
 editor.apply **{** putString("string\_KEYCor1", insertedText.toString())  
 putString("string\_KEYCor2", insertedText2.toString())  
 **}**.apply()  
 }  
 //функция для обращения к API для получения времени затраченного на путь  
 private fun response(q1:String, q2:String){  
  
 loadDataTrans2()  
  
 var interpreter ="transit"  
 if (timeWaySettings == "Свой автомобиль"){  
 interpreter = "driving"  
 }  
 else{  
 if(timeWaySettings == "Общественный транспорт"){  
 interpreter = "transit"  
 }  
 else{timeWaySettings == "Пешком"  
 interpreter = "walking"}  
 }  
  
 val builder = Retrofit.Builder()  
 .baseUrl("https://maps.googleapis.com")  
 .addConverterFactory(GsonConverterFactory.create())  
 val retrofit = builder.build()  
 val apiInterface : APInterface = retrofit.create<APInterface>(APInterface::class.java)  
 val call : retrofit2.Call<UrlModel> = apiInterface.getRoute(q1,q2,true, interpreter)  
 call.enqueue(object : Callback<UrlModel> {  
 override fun onFailure(call: retrofit2.Call<UrlModel>, t: Throwable) {  
 Log.i("myLog", t.message.toString())  
 }  
 override fun onResponse(call: Call<UrlModel>, response: Response<UrlModel>) {  
 val statusResponse = response.body()!!  
 if(response.code()==200){  
 timeWay = statusResponse.routes[0]!!.legs[0]!!.duration!!.text!!.toString()  
 //Toast.makeText(requireActivity(), "${timeWay}", Toast.LENGTH\_SHORT).show()  
 textViewTrans.text = "Вид транспорта: "+"${timeWaySettings.toString()}\n" +  
 "Время на путь: ${timeWay.toString()}"  
 if (timeWaySettings !="."){}  
 textViewTrans.isVisible = true  
 }  
 else  
 {  
 Toast.makeText(requireActivity(), "404", Toast.LENGTH\_SHORT).show()  
 }  
 }  
 })  
 }  
 //функция передачи времени на другое активити  
 private fun loadDataTrans2() {  
 val mainActivity = this.activity as ForFragmentMainActivity  
 val sharedPreferences= mainActivity.getSharedPreferences("sharedPrefsTrans", Context.MODE\_PRIVATE)  
 val savedString1 = sharedPreferences.getString("string\_KEYTrans", ".")  
 timeWaySettings = savedString1.toString()  
 Log.i("myLog", timeWaySettings.toString())  
 }  
  
}

11)PessonalAccounBlank.kt

package com.example.placeofrest.fragments  
  
import android.content.Context  
import android.os.Bundle  
import android.preference.PreferenceManager  
import android.util.Log  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.AdapterView  
import android.widget.Toast  
import androidx.fragment.app.Fragment  
import com.example.placeofrest.AutorizationMainActivity  
import com.example.placeofrest.ForFragmentMainActivity  
import com.example.placeofrest.MainActivityRec  
import com.example.placeofrest.R  
import kotlinx.android.synthetic.main.activity\_autorization\_main.\*  
import kotlinx.android.synthetic.main.fragment\_pessonal\_account\_blank.\*  
  
  
class PessonalAccountBlankFragment : Fragment() {  
  
 var let =""  
  
 override fun onCreateView(  
 inflater: LayoutInflater, container: ViewGroup?,  
 savedInstanceState: Bundle?  
 ): View? {  
 return inflater.inflate(R.layout.fragment\_pessonal\_account\_blank, container, false)  
 }  
 //основная функция, которая содежжит  
 override fun onViewCreated(view: View, savedInstanceState: Bundle?) {  
 super.onViewCreated(view, savedInstanceState)  
  
 spinnerTrans.isEnabled = false  
 loadData()  
 textViewAkk.text = "Почта: "+let  
  
 if(textViewAkk.text == "Нет данных о почте"){  
 spinnerTrans.isEnabled = false  
 textViewTrans.text = "Функция определения времени поездки будет досьупна после авторизации"  
 }else{  
 spinnerTrans.isEnabled = true  
 textViewTrans.text = "Вам доступна функция определения времени поездки"  
 }  
  
 spinnerTrans.onItemSelectedListener = object : AdapterView.OnItemSelectedListener{  
 override fun onItemSelected(//метод если список бует заполен  
 parent: AdapterView<\*>?,  
 view: View?,  
 position: Int,  
 id: Long  
 ) {  
 saveDataTrans()  
 }  
 //метод если в списке нчего не выбрано  
 override fun onNothingSelected(parent: AdapterView<\*>?) {  
 }  
 }  
  
 //переход на другое активити  
 buttonIn.setOnClickListener **{** startActivity(  
 android.content.Intent(  
 activity,  
 AutorizationMainActivity::class.java  
 )  
 )  
 **}** textViewExit.setOnClickListener**{** saveData()  
 Toast.makeText(requireContext(), "Вы вышли из аккаунта", Toast.LENGTH\_SHORT).show()  
 textViewTrans.text = "Функция определения времени поездки будет досьупна после авторизации"  
 **}** }  
 //прогрузка данных о почте  
 private fun loadData() {  
 val mainActivity = this.activity as ForFragmentMainActivity  
 val sharedPreferences= mainActivity.getSharedPreferences("sharedPrefs", Context.MODE\_PRIVATE)  
 val savedString1 = sharedPreferences.getString("string\_KEY", "Нет данных о почте")  
 let = savedString1!!  
  
 Log.i("myLog", let)  
 }  
  
 //проверка того, авторизирован ли пользователь или нет  
 private fun saveData() {  
 val mainActivity = this as AutorizationMainActivity  
 val insertedText = "Аккаунт не авторизован"  
 val sharedPreferences = mainActivity.getSharedPreferences("sharedPrefs", Context.MODE\_PRIVATE)  
 val editor = sharedPreferences.edit()  
 editor.apply **{** putString("string\_KEY", insertedText.toString())  
 **}**.apply()  
 }  
 //сохранение данных о времени затраеннго на путь  
 private fun saveDataTrans() {  
 val mainActivity = this.activity as ForFragmentMainActivity  
 val insertedText = spinnerTrans.selectedItem.toString()  
  
 val sharedPreferences = mainActivity.getSharedPreferences("sharedPrefsTrans", Context.MODE\_PRIVATE)  
 val editor = sharedPreferences.edit()  
 editor.apply **{** putString("string\_KEYTrans", insertedText.toString())  
 **}**.apply()  
 Log.i("myLog", insertedText)  
 }  
}

12) activity\_autorization\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".AutorizationMainActivity">  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/autorization"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintBottom\_toTopOf="@id/editTextLogin"  
 android:layout\_marginBottom="30dp"  
 style="@style/TextTitlePage"  
 />  
  
 <EditText  
 android:id="@+id/editTextLogin"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="qwe"  
 android:hint="Mail"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 android:layout\_marginBottom="70dp"  
 style="@style/TextOnPageEditText"/>  
  
 <EditText  
 android:id="@+id/editTextPassword"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="qwe"  
 android:hint="Password"  
 android:layout\_marginTop="30dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/editTextLogin"  
 style="@style/TextOnPageEditText"/>  
  
 <Button  
 android:id="@+id/buttonAutorization"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintTop\_toBottomOf="@+id/editTextPassword"  
 android:text="@string/autorizationButton"  
 style="TextOnPageAutoOrReg"  
 android:layout\_marginStart="70dp"  
 android:layout\_marginEnd="70dp"/>  
  
 <Button  
 android:id="@+id/buttonReg"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintTop\_toBottomOf="@+id/buttonAutorization"  
 android:text="@string/registrationButton"  
 style="TextOnPageAutoOrReg"  
 android:layout\_marginStart="70dp"  
 android:layout\_marginEnd="70dp"/>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

13) activity\_for\_fragment\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".ForFragmentMainActivity">  
  
   
 <FrameLayout  
 android:id="@+id/fl\_wrapper"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_above="@id/bottomNavigationView"/>  
  
 <com.google.android.material.bottomnavigation.BottomNavigationView  
 android:id="@+id/bottomNavigationView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentBottom="true"  
 android:foreground="?attr/selectableItemBackground"  
 android:background="?android:attr/windowBackground"  
 app:menu="@menu/menu"/>  
  
</RelativeLayout>

14) activity\_main\_detail\_event.xml

<?xml version="1.0" encoding="utf-8"?>  
  
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivityDetailEvent"  
 android:background="@color/grey1"  
 >  
  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:background="@color/grey1">  
 <androidx.cardview.widget.CardView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:cardCornerRadius="25dp"  
 >  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
  
 <ImageView  
 android:id="@+id/imageViewDetail"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:src="@drawable/ic\_launcher\_background"  
 android:scaleType="centerCrop"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 />  
  
  
 <TextView  
 android:id="@+id/textViewTitleDetail"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 android:gravity="center"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginEnd="10dp"  
 android:textColor="@color/black"  
 app:layout\_constraintTop\_toBottomOf="@id/imageViewDetail"  
 style="@style/DetailTitle"  
 android:text="Загрузка" />  
  
 <TextView  
 android:id="@+id/textViewDescriptionDetail"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:justificationMode="inter\_word"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 android:textColor="@color/black"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 app:layout\_constraintTop\_toBottomOf="@id/textViewTitleDetail"  
 style="@style/DetailTextOnPage"  
 android:text="Загрузка" />  
  
 <TextView  
 android:id="@+id/textViewDescriptionTime"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:justificationMode="inter\_word"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 android:textColor="@color/black"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 app:layout\_constraintTop\_toBottomOf="@id/textViewDescriptionDetail"  
 style="@style/DetailTextOnPage"  
 android:text="Загрузка" />  
  
 <TextView  
 android:id="@+id/textViewCoordinateBuildAndMetroDetail"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 android:textColor="@color/black"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 app:layout\_constraintTop\_toBottomOf="@id/textViewDescriptionTime"  
 style="@style/DetailTextOnPage"  
 android:text="Загрузка" />  
  
 <TextView  
 android:id="@+id/textViewMetro"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 android:textColor="@color/black"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 style="@style/DetailTextOnPage"  
 android:text="Загрузка"  
 app:layout\_constraintTop\_toBottomOf="@id/textViewCoordinateBuildAndMetroDetail"/>  
  
  
 <Button  
 android:id="@+id/buttonForMap"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 android:layout\_marginEnd="35dp"  
 android:layout\_marginStart="35dp"  
 android:text="Показать на карте"/>  
  
 <TextView  
 android:id="@+id/textViewLink"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="match\_parent"  
 android:layout\_marginStart="40dp"  
 android:layout\_marginEnd="40dp"  
 android:text="Открыть в браузере"  
 android:layout\_gravity="center"  
 android:textColor="@color/link"/>  
 </LinearLayout>  
 </androidx.cardview.widget.CardView>  
 </LinearLayout>  
</ScrollView>

15) activity\_main\_rec.kt

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivityRec"  
 android:background="@color/grey1">  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/textInputLayout"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Тема"  
 app:endIconMode="custom"  
 app:endIconDrawable="@drawable/ic\_baseline\_zoom\_in\_24"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/textInputEditText"  
 android:layout\_width="match\_parent"  
 android:background="@color/white"  
 android:layout\_height="wrap\_content"  
 />  
  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/recyclerView"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 tools:listitem="@layout/recyclerview\_item"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/textInputLayout"/>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

16) activity\_registration\_main.kt

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".RegistrationMainActivity">  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/registration"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintBottom\_toTopOf="@id/editTextLoginReg"  
 android:layout\_marginBottom="30dp"  
 style="@style/TextTitlePage"  
 />  
  
 <EditText  
 android:id="@+id/editTextLoginReg"  
 android:layout\_marginBottom="70dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="qwe"  
 android:hint="Mail"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 style="@style/TextOnPageEditText"/>  
  
 <EditText  
 android:id="@+id/editTextPasswordReg"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="qwe"  
 android:hint="Password"  
 android:layout\_marginTop="30dp"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/editTextLoginReg"  
 style="@style/TextOnPageEditText"/>  
  
 <Button  
 android:id="@+id/buttonRegis"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintTop\_toBottomOf="@+id/editTextPasswordReg"  
 android:text="@string/registrationButton"  
 style="TextOnPageAutoOrReg"  
 android:layout\_marginStart="70dp"  
 android:layout\_marginEnd="70dp"/>  
  
</androidx.constraintlayout.widget.ConstraintLayout>

17) fragment\_map\_blank.kt

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 tools:context=".fragments.EventFeedBlankFragment">  
  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/recyclerView1"  
 android:layout\_width="0dp"  
 android:layout\_height="0dp"  
 android:layout\_marginBottom="8dp"  
 android:layout\_marginEnd="8dp"  
 android:layout\_marginStart="8dp"  
 android:layout\_marginTop="8dp"  
 tools:listitem="@layout/item\_list"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

18) fragment\_pessonal\_accout\_blank.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 tools:context=".fragments.PessonalAccountBlankFragment">  
  
  
 <TextView  
 android:id="@+id/title"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Информация об аккаунте"  
 style="@style/TextTitlePage"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 android:textColor="@color/black"  
 app:layout\_constraintTop\_toTopOf="parent"/>  
  
 <TextView  
 android:id="@+id/textViewAkk"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/title"  
 android:hint="почта"  
 android:textColor="@color/black"  
 style="@style/TextOnPageTitleAccount"  
 />  
  
 <TextView  
 android:id="@+id/textViewTrans"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textViewAkk"  
 android:hint="Функция определения времени поездки будет досьупна после авторизации"  
 android:layout\_marginTop="40dp"  
 android:textSize="15dp"  
 android:textColor="@color/black"  
 style="@style/TextOnPageTitleAccount"  
 />  
  
 <Spinner  
 android:id="@+id/spinnerTrans"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/textViewTrans"  
 android:layout\_marginTop="30dp"  
  
 android:entries="@array/TransForSpinner"  
 />  
  
 <Button  
 android:id="@+id/buttonIn"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="@string/autorizationButton"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintBottom\_toTopOf="@id/textViewExit"  
 android:layout\_marginStart="35dp"  
 android:layout\_marginEnd="35dp"  
 android:layout\_marginBottom="10dp"  
 />  
  
 <TextView  
 android:id="@+id/textViewExit"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Выйти из аккаунта"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 android:layout\_marginBottom="20dp"  
 android:gravity="center"  
 />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

19) recyclerview\_item.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.cardview.widget.CardView android:id="@+id/cardViewRec"  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 app:cardBackgroundColor="@color/grey1"  
 app:cardCornerRadius="35dp"  
 android:layout\_marginBottom="10dp"  
 xmlns:app="http://schemas.android.com/apk/res-auto">  
  
 <RelativeLayout  
 android:background="@color/white"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <ImageView  
 android:id="@+id/imageRec"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:scaleType="centerCrop"  
 android:src="@drawable/ic\_launcher\_background"/>  
  
 <TextView  
 android:id="@+id/textViewTitle"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@id/imageRec"  
 android:text="TextView"  
 android:textColor="@color/black"  
 android:layout\_marginStart="16dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginEnd="16dp"  
 android:layout\_marginBottom="10dp"  
 android:textAlignment="center"  
 android:textSize="18sp"  
 android:textStyle="bold"  
 android:textAppearance="?android:attr/textAppearanceSmall" />  
 </RelativeLayout>  
 </androidx.cardview.widget.CardView>

20) menu.xml

<?xml version="1.0" encoding="utf-8"?>  
<menu xmlns:android="http://schemas.android.com/apk/res/android">  
 <item  
 android:id="@+id/ic\_map"  
 android:icon="@drawable/ic\_map"  
 android:title="Карта" />  
  
 <item  
 android:id="@+id/ic\_home"  
 android:icon="@drawable/ic\_home"  
 android:title="Аккаунт" />  
</menu>

21) google\_maps\_api.xml

<resources>  
 <!--  
 *TODO: Before you run your application, you need a Google Maps API key.*  
  
 To get one, follow this link, follow the directions and press "Create" at the end:  
  
 https://console.developers.google.com/flows/enableapi?apiid=maps\_android\_backend&keyType=CLIENT\_SIDE\_ANDROID&r=5B:57:32:FE:C6:9B:4B:77:08:64:E3:EB:F9:F7:27:08:D8:2F:07:7B%3Bcom.example.placeofrest  
  
 You can also add your credentials to an existing key, using these values:  
  
 Package name:  
 com.example.placeofrest  
  
 SHA-1 certificate fingerprint:  
 5B:57:32:FE:C6:9B:4B:77:08:64:E3:EB:F9:F7:27:08:D8:2F:07:7B  
  
 Alternatively, follow the directions here:  
 https://developers.google.com/maps/documentation/android/start#get-key  
  
 Once you have your key (it starts with "AIza"), replace the "google\_maps\_key"  
 string in this file.  
 -->  
 <string name="google\_maps\_key" templateMergeStrategy="preserve" translatable="false">AIzaSyB\_1syNZPbX04aOhLkCtWIOV81j7zKW2tU</string>  
</resources>

22) strings.xml

<resources>  
 <string name="app\_name">КудаСегодня</string>  
 <string name="title\_activity\_maps">Map</string>  
 <!-- *TODO: Remove or change this placeholder text* -->  
 <string name="hello\_blank\_fragment">Hello blank fragment</string>  
  
 <string name="autorization">Авторизация</string>  
 <string name="registration">Регитсрация</string>  
  
 <string name="autorizationButton">Авторизироваться</string>  
 <string name="registrationButton">Зарегестрироваться</string>  
  
 <string-array name="ArrayForSpinner">  
 <item>Аккаунт</item>  
 <item>Авторизация</item>  
 <item>Регистрация</item>  
 </string-array>  
  
 <string-array name="TransForSpinner">  
 <item>Свой автомобиль</item>  
 <item>Общественный транспорт</item>  
 <item>Пешком</item>  
 </string-array>  
  
  
</resources>

23) style.xml

<resources>  
  
 <style name="TextTitlePage">  
 <item name="android:textSize">25dp</item>  
 <item name="android:typeface">monospace</item>  
 <item name="android:textAlignment">center</item>  
 <item name="android:layout\_marginTop">5dp</item>  
 </style>  
  
 <style name="TextOnPageEditText">  
 <item name="android:textSize">17dp</item>  
 <item name="android:typeface">monospace</item>  
 <item name="android:textAlignment">textStart</item>  
 </style>  
  
 <style name="TextOnPageTextView">  
 <item name="android:textSize">17dp</item>  
 <item name="android:typeface">monospace</item>  
 <item name="android:textAlignment">center</item>  
 </style>  
  
 <style name="TextOnPageTitleAccount">  
 <item name="android:textSize">25dp</item>  
 <item name="android:typeface">monospace</item>  
 <item name="android:textAlignment">textStart</item>  
 </style>  
  
 <style name="TextOnPageText">  
 <item name="android:textSize">20dp</item>  
 <item name="android:typeface">monospace</item>  
 <item name="android:textAlignment">textStart</item>  
 </style>  
  
 <style name="TextOnPageInSpinner">  
 <item name="android:textSize">30dp</item>  
 <item name="android:typeface">monospace</item>  
 </style>  
  
 <style name="TextOnPageAutoOrReg">  
 <item name="android:textSize">20dp</item>  
 <item name="android:color">@color/white</item>  
 <item name="android:typeface">monospace</item>  
 <item name="android:textAlignment">center</item>  
 </style>  
  
 <style name="TextOnDeteil">  
 <item name="android:textSize">20dp</item>  
 <item name="android:typeface">monospace</item>  
 <item name="android:textAlignment">center</item>  
 </style>  
  
 <style name="DetailTitle">  
 <item name="android:textSize">20dp</item>  
 <item name="android:typeface">monospace</item>  
 <item name="android:textAlignment">center</item>  
 </style>  
  
 <style name="DetailTextOnPage">  
 <item name="android:textSize">17dp</item>  
 </style>  
  
</resources>

24) EventController.cs

using Microsoft.AspNetCore.Http;

using Microsoft.AspNetCore.Mvc;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace APIPlaceOfRestProject.Controllers

{

[Route("api/[controller]")]

[ApiController]

public class EventController : ControllerBase

{

APIPlaceOfRestContext \_context;

public EventController(APIPlaceOfRestContext context)

{

\_context = context;

}

[HttpGet]

[Route("[action]")]

public IEnumerable<object> GetEvents()

{

var result = from events in \_context.Events

join themeevents in \_context.Themes on events.Theme equals themeevents.Idtheme

join picturelick in \_context.PictureLinks on events.PictureLink equals picturelick.Idpicture

join eventlick in \_context.EventLinks on events.EventLink equals eventlick.Idevent

join building in \_context.Buildings on events.Building equals building.Idbuilding

join street in \_context.Streets on building.Idstreet equals street.Idstreet

join underground in \_context.Undergrounds on events.Underground equals underground.Idunderground

join coordinateLatitude in \_context.Coordinates on building.Idcoordinate equals coordinateLatitude.Idcoordinate

join coordinateLongitude in \_context.Coordinates on building.Idcoordinate equals coordinateLongitude.Idcoordinate

select new { id = events.Idevent,

name = events.Name,

theme = themeevents.Name,

date = events.Date,

discription = events.Discription,

picture = picturelick.Link,

eventev = eventlick.Link,

build = building.NumberBuilding,

structur = building.NumberStructure,

street = street.Name,

metro = underground.Name,

coordinatee = coordinateLatitude.Latitude,

coordinateLongitude.Longitude};

return result;

}

[HttpGet]

[Route("[action]")]

public IEnumerable<object> GetMapMarker()

{

var result = from events in \_context.Events

join building in \_context.Buildings on events.Building equals building.Idbuilding

join coordinateLatitude in \_context.Coordinates on building.Idcoordinate equals coordinateLatitude.Idcoordinate

join coordinateLongitude in \_context.Coordinates on building.Idcoordinate equals coordinateLongitude.Idcoordinate

select new { id = events.Idevent, name = events.Name, coordinatee = coordinateLatitude.Latitude, coordinateLongitude.Longitude };

return result;

}

[HttpGet]

[Route("[action]/theme/{theme}")]

public IEnumerable<Object> SearchByEvents(string theme)

{

var result = from events in \_context.Events

join themeevents in \_context.Themes on events.Theme equals themeevents.Idtheme

join picturelick in \_context.PictureLinks on events.PictureLink equals picturelick.Idpicture

join eventlick in \_context.EventLinks on events.EventLink equals eventlick.Idevent

join building in \_context.Buildings on events.Building equals building.Idbuilding

join street in \_context.Streets on building.Idstreet equals street.Idstreet

join underground in \_context.Undergrounds on events.Underground equals underground.Idunderground

join coordinateLatitude in \_context.Coordinates on building.Idcoordinate equals coordinateLatitude.Idcoordinate

join coordinateLongitude in \_context.Coordinates on building.Idcoordinate equals coordinateLongitude.Idcoordinate

where themeevents.Name == theme

select new

{

id = events.Idevent,

name = events.Name,

theme = themeevents.Name,

date = events.Date,

discription = events.Discription,

picture = picturelick.Link,

eventev = eventlick.Link,

build = building.NumberBuilding,

structur = building.NumberStructure,

street = street.Name,

metro = underground.Name,

coordinatee = coordinateLatitude.Latitude,

coordinateLongitude.Longitude

};

return result;

}

[HttpGet]

[Route("[action]/id/{id}")]

public IEnumerable<Object> CatchIDEvent(int id)

{

var result = from events in \_context.Events

join themeevents in \_context.Themes on events.Theme equals themeevents.Idtheme

join picturelick in \_context.PictureLinks on events.PictureLink equals picturelick.Idpicture

join eventlick in \_context.EventLinks on events.EventLink equals eventlick.Idevent

join building in \_context.Buildings on events.Building equals building.Idbuilding

join street in \_context.Streets on building.Idstreet equals street.Idstreet

join underground in \_context.Undergrounds on events.Underground equals underground.Idunderground

join coordinateLatitude in \_context.Coordinates on building.Idcoordinate equals coordinateLatitude.Idcoordinate

join coordinateLongitude in \_context.Coordinates on building.Idcoordinate equals coordinateLongitude.Idcoordinate

where events.Idevent == id

select new

{

id = events.Idevent,

name = events.Name,

theme = themeevents.Name,

date = events.Date,

discription = events.Discription,

picture = picturelick.Link,

eventev = eventlick.Link,

build = building.NumberBuilding,

structur = building.NumberStructure,

street = street.Name,

metro = underground.Name,

coordinatee = coordinateLatitude.Latitude,

coordinateLongitude.Longitude

};

return result;

}

}

}