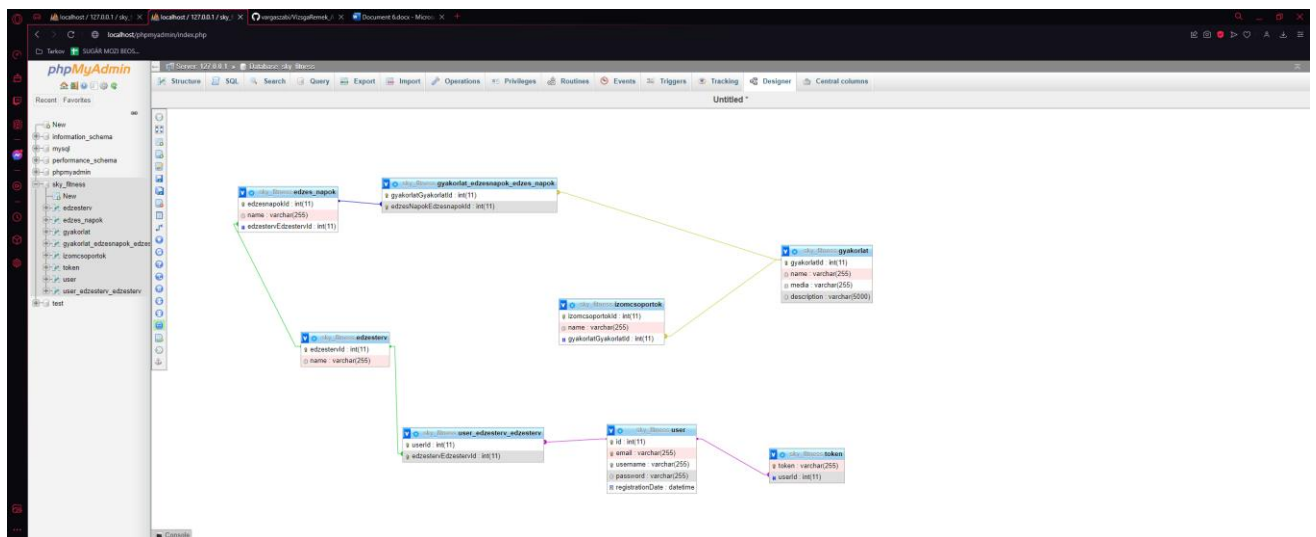


Android Vizsgaremek

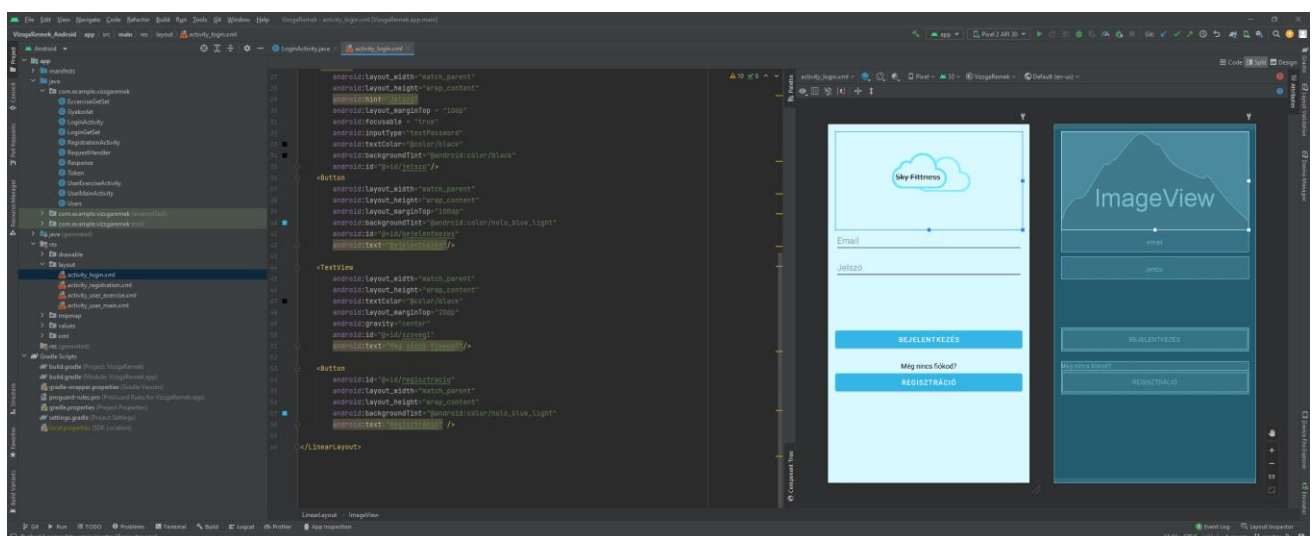
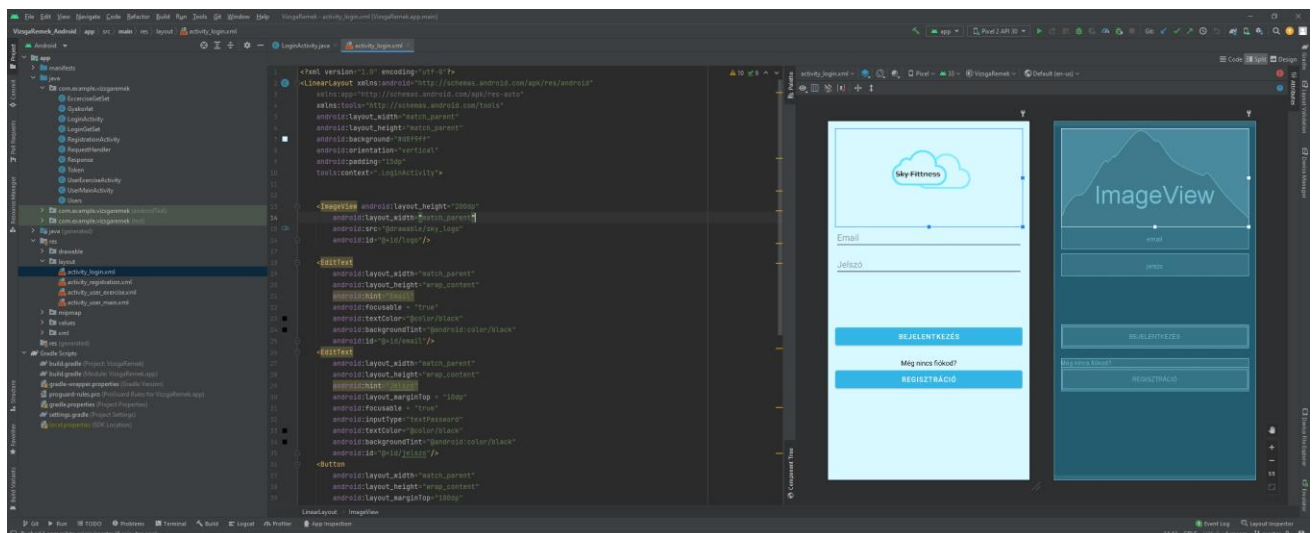
Fejlesztői Dokumentáció-

Varga Szabolcs

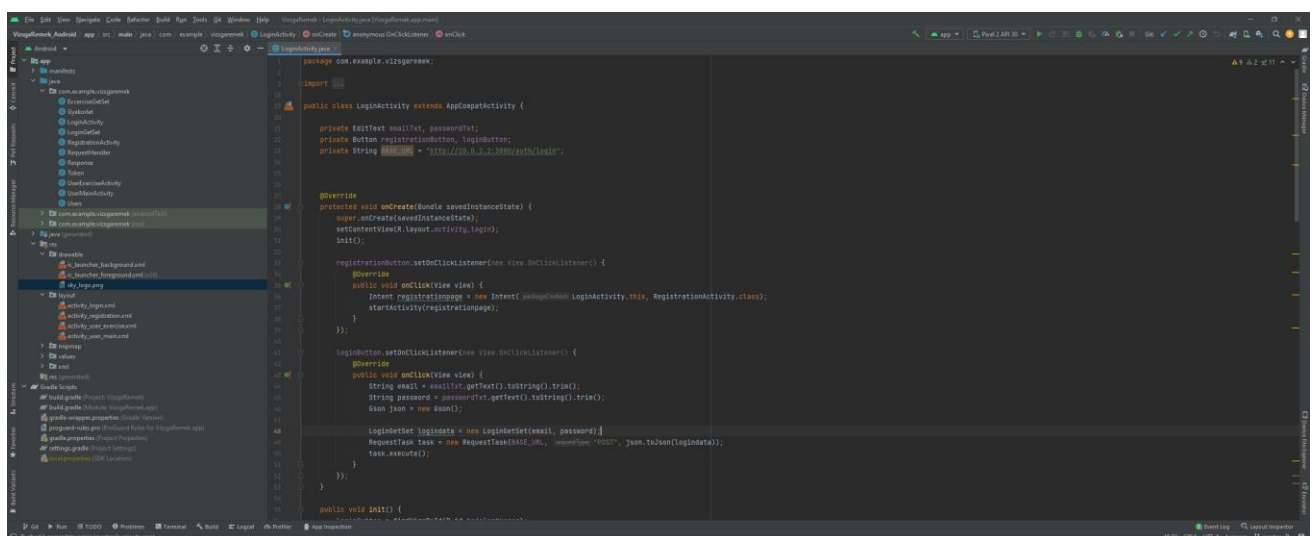


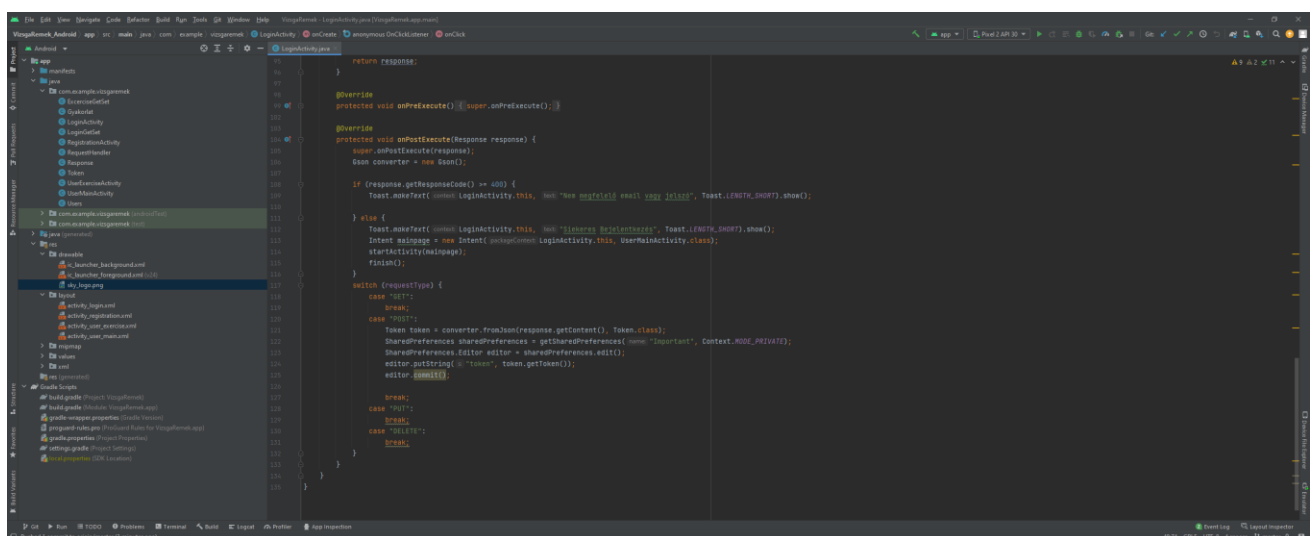
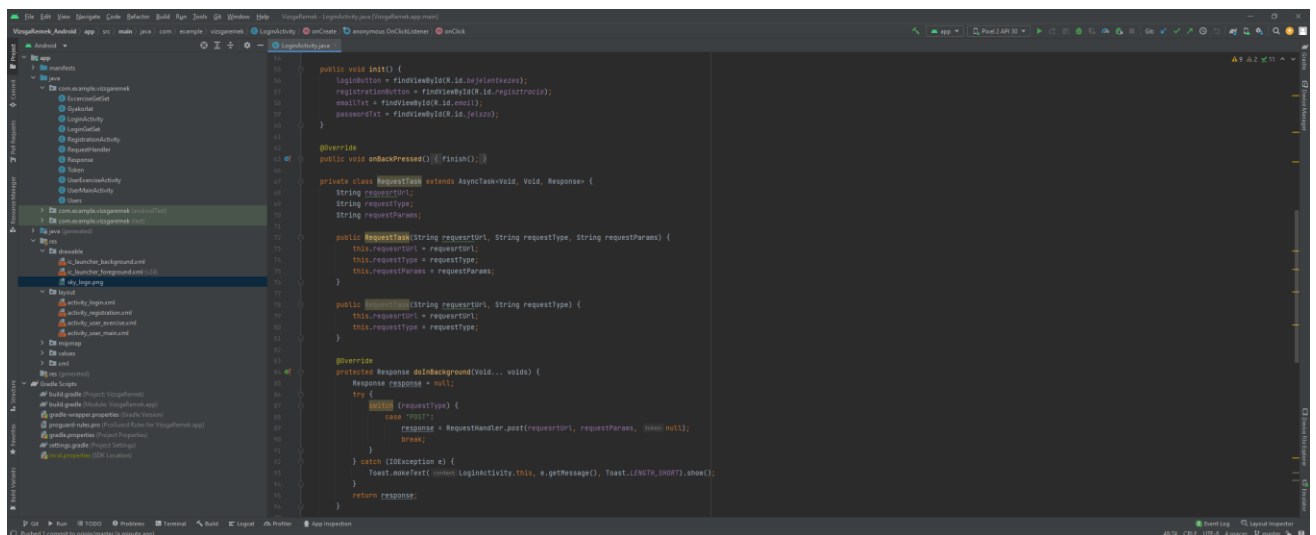
Ez az adatbázismodellje az alkalmazásnak. Ennek a pontos leírása a Backend Dokumentációba megtalálható.

LoginActivity

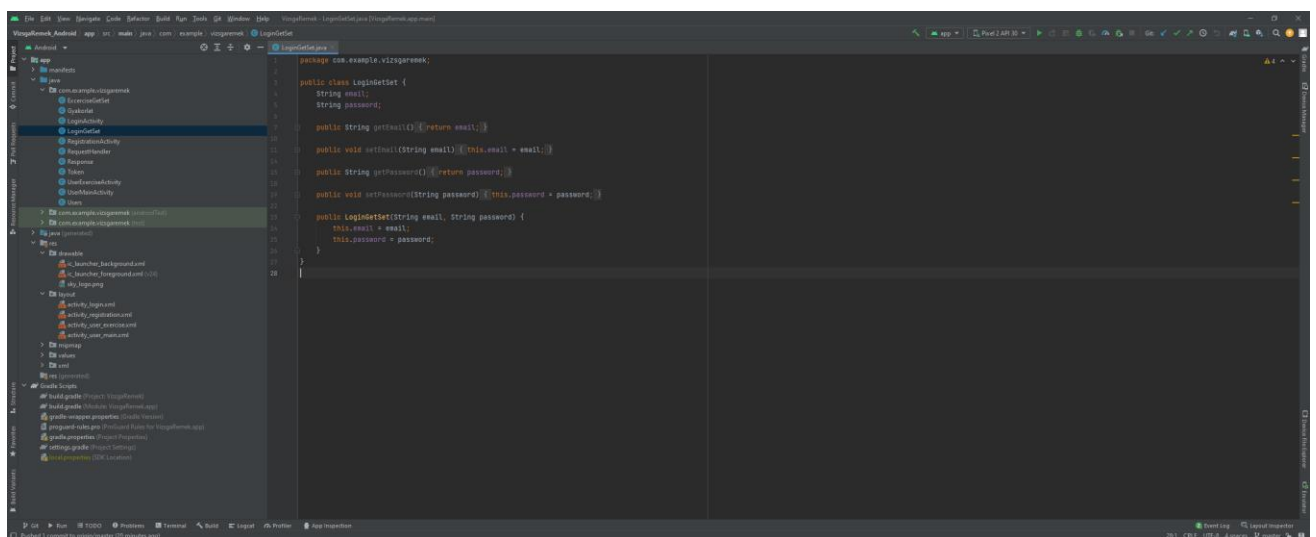


A login oldal XML fájlja. Ez a nyitóoldal az alkalmazásban innen lehet bejelentkezni vagy a regisztrációs oldalra átmenni a gombok segítségével.



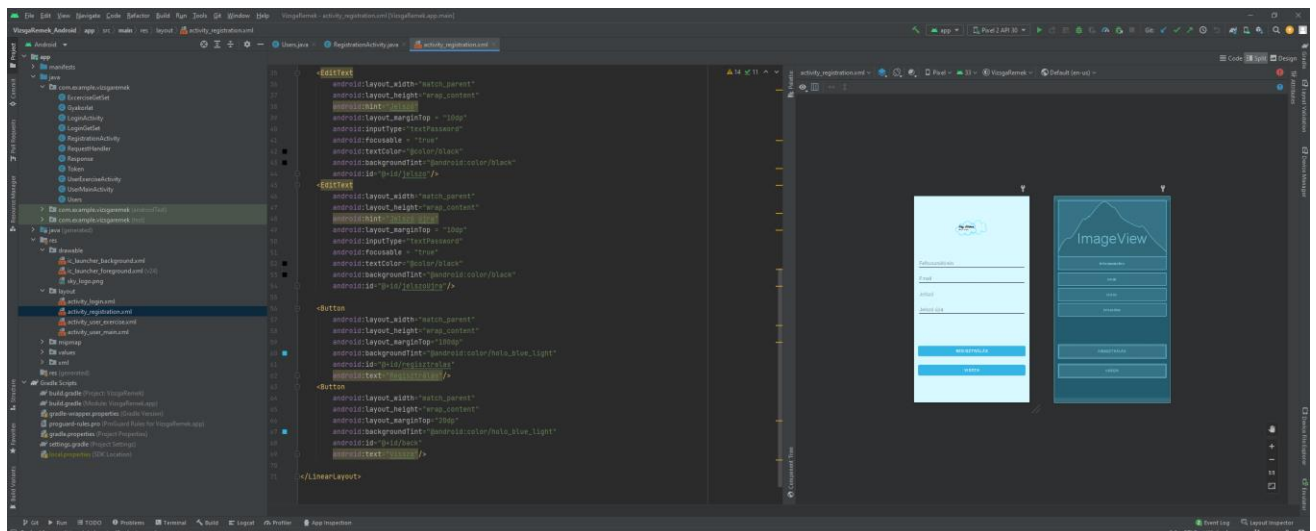
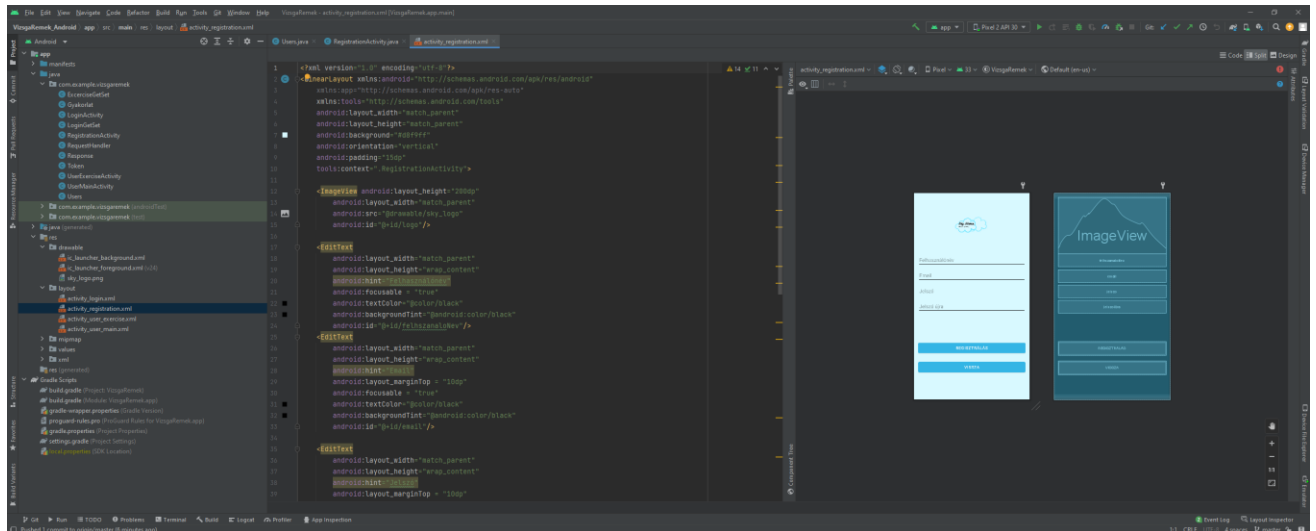


Ez maga a LoginActivity-nek a Java nyelven megírt kódja. Itt az androidon belül átalakítom az adatokat egy Json formátumba és így küldi el a Backendnek az adatot. Hibakezelés történik Frontend és Backend részen is.



A Login osztálynak a Gettere Settere ami segít átalakítani az adatokat Json formátumra.

RegistrationActivity



Ez a registration oldalnak az xml fájlja. Itt is a gombok segítségével lehet navigálni.

```

1 package com.example.v123456789;
2
3 import android.content.Intent;
4 import android.os.AsyncTask;
5 import android.os.Bundle;
6 import android.view.View;
7 import android.widget.Button;
8 import android.widget.EditText;
9 import android.widget.Toast;
10
11 import androidx.appcompat.app.AppCompatActivity;
12
13 import com.google.gson.Gson;
14
15 import java.io.IOException;
16
17 public class RegistrationActivity extends AppCompatActivity {
18     private EditText usernameTxt, emailTxt, passwordTxt, passwordGainTxt;
19     private Button registrationButton, backButton;
20     private String baseUrl = "http://192.168.1.100:8080/register";
21
22     public void init() {
23         usernameTxt = findViewById(R.id.usernameTxt);
24         emailTxt = findViewById(R.id.email);
25         passwordTxt = findViewById(R.id.password);
26         passwordGainTxt = findViewById(R.id.passwordGain);
27         registrationButton = findViewById(R.id.registration);
28         backButton = findViewById(R.id.back);
29     }
30
31     @Override
32     protected void onCreate(Bundle savedInstanceState) {
33         super.onCreate(savedInstanceState);
34         setContentView(R.layout.activity_registration);
35         init();
36
37         registrationButton.setOnClickListener(new View.OnClickListener() {
38             @Override
39             public void onClick(View view) {
40                 String username = usernameTxt.getText().toString().trim();
41                 String email = emailTxt.getText().toString().trim();
42                 String password = passwordTxt.getText().toString().trim();
43                 String passwordGain = passwordGainTxt.getText().toString().trim();
44                 if (username.isEmpty()) {
45                     usernameTxt.setError("New login here");
46                     return;
47                 } else if (email.isEmpty()) {
48                     emailTxt.setError("New login here");
49                     return;
50                 } else if (password.isEmpty()) {
51                     passwordTxt.setError("New login here");
52                     return;
53                 } else if (passwordGain.isEmpty()) {
54                     passwordGainTxt.setError("New login here");
55                     return;
56                 } else if (!password.equals(passwordGain)) {
57                     passwordGainTxt.setError("New login here");
58                     return;
59                 } else {
60                     passwordGainTxt.setError(null);
61                 }
62             }
63         });
64     }
65 }

```

```

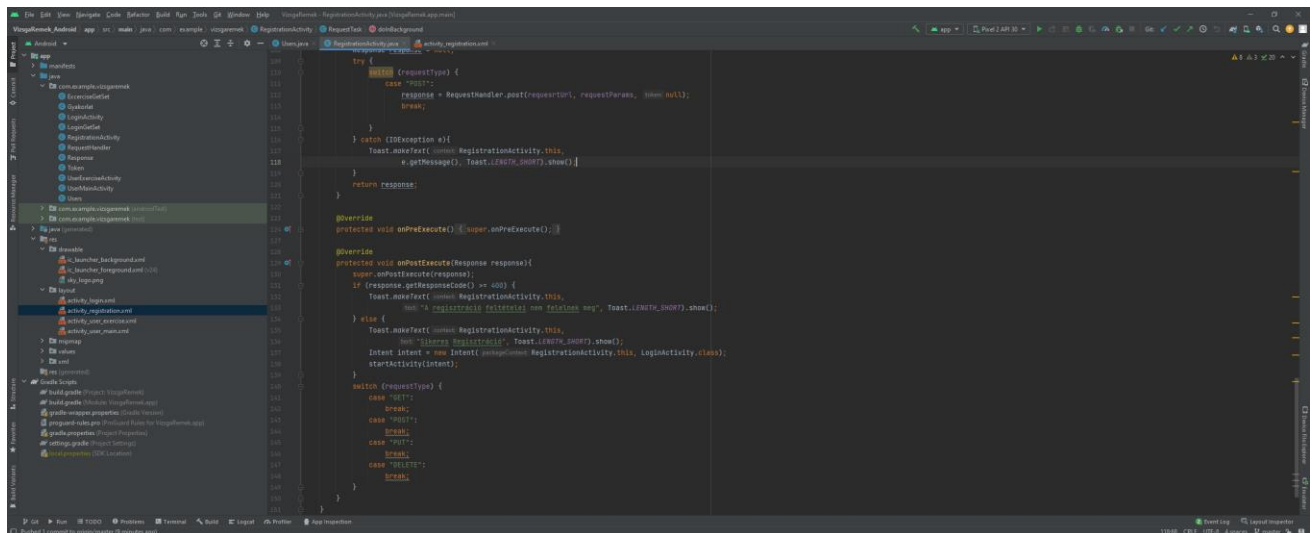
66     @Override
67     protected void onCreate(Bundle savedInstanceState) {
68         super.onCreate(savedInstanceState);
69         setContentView(R.layout.activity_registration);
70         init();
71
72         registrationButton.setOnClickListener(new View.OnClickListener() {
73             @Override
74             public void onClick(View view) {
75                 String username = usernameTxt.getText().toString().trim();
76                 String email = emailTxt.getText().toString().trim();
77                 String password = passwordTxt.getText().toString().trim();
78                 String passwordGain = passwordGainTxt.getText().toString().trim();
79                 if (username.isEmpty()) {
80                     usernameTxt.setError("New login here");
81                     return;
82                 } else if (email.isEmpty()) {
83                     emailTxt.setError("New login here");
84                     return;
85                 } else if (password.isEmpty()) {
86                     passwordTxt.setError("New login here");
87                     return;
88                 } else if (passwordGain.isEmpty()) {
89                     passwordGainTxt.setError("New login here");
90                     return;
91                 } else if (!password.equals(passwordGain)) {
92                     passwordGainTxt.setError("New login here");
93                     return;
94                 } else {
95                     passwordGainTxt.setError(null);
96                 }
97             }
98         });
99     }
100 }

```

```

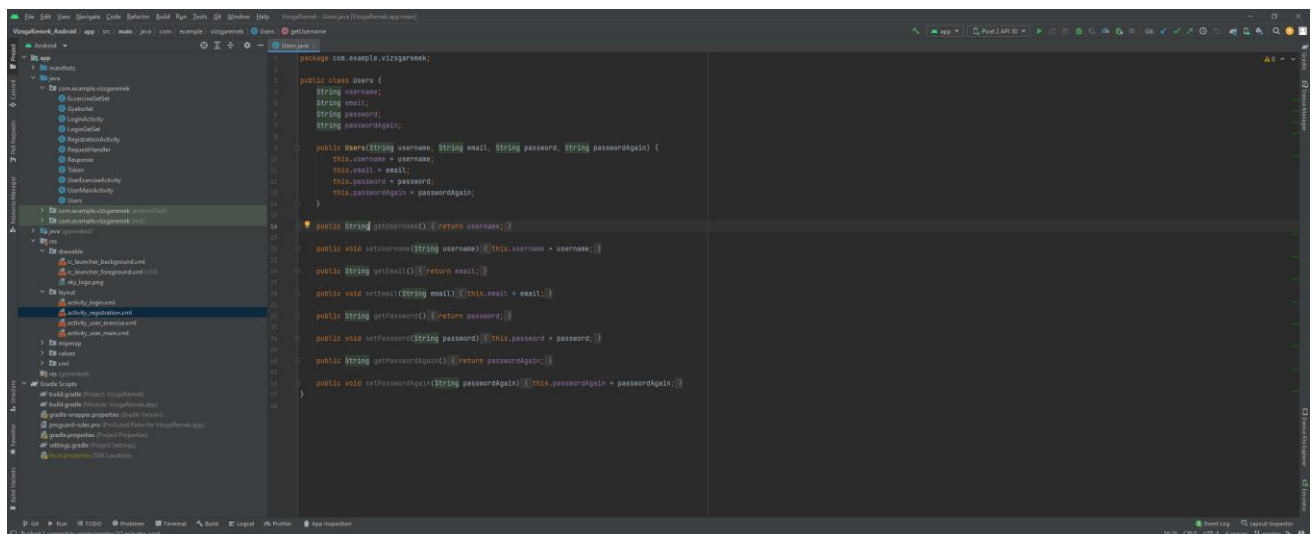
101     @Override
102     protected void onCreate(Bundle savedInstanceState) {
103         super.onCreate(savedInstanceState);
104         setContentView(R.layout.activity_registration);
105         init();
106
107         registrationButton.setOnClickListener(new View.OnClickListener() {
108             @Override
109             public void onClick(View view) {
110                 String username = usernameTxt.getText().toString().trim();
111                 String email = emailTxt.getText().toString().trim();
112                 String password = passwordTxt.getText().toString().trim();
113                 String passwordGain = passwordGainTxt.getText().toString().trim();
114                 if (username.isEmpty()) {
115                     usernameTxt.setError("New login here");
116                     return;
117                 } else if (email.isEmpty()) {
118                     emailTxt.setError("New login here");
119                     return;
120                 } else if (password.isEmpty()) {
121                     passwordTxt.setError("New login here");
122                     return;
123                 } else if (passwordGain.isEmpty()) {
124                     passwordGainTxt.setError("New login here");
125                     return;
126                 } else if (!password.equals(passwordGain)) {
127                     passwordGainTxt.setError("New login here");
128                     return;
129                 } else {
130                     passwordGainTxt.setError(null);
131                 }
132             }
133         });
134     }
135 }

```

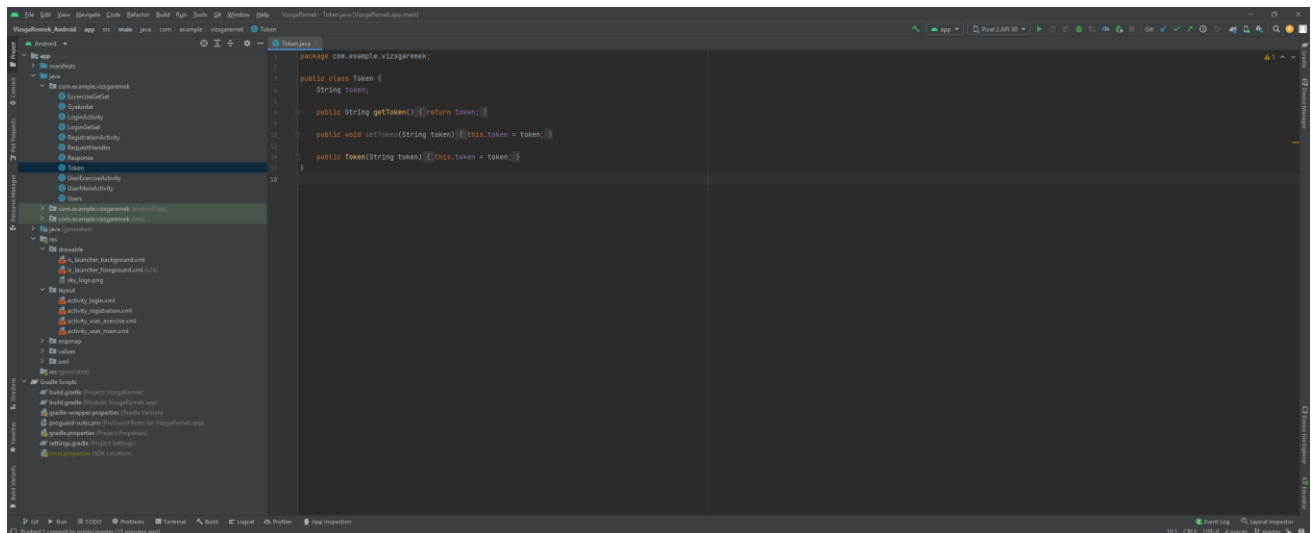


Ez a RegistrationActivity-nek a Java nyelven megírt kódja. Itt is az adatok hibakezelve vannak Frontend és Backend részen is és Json formátumban küldi tovább a Backend részre.

User Class és Token Class

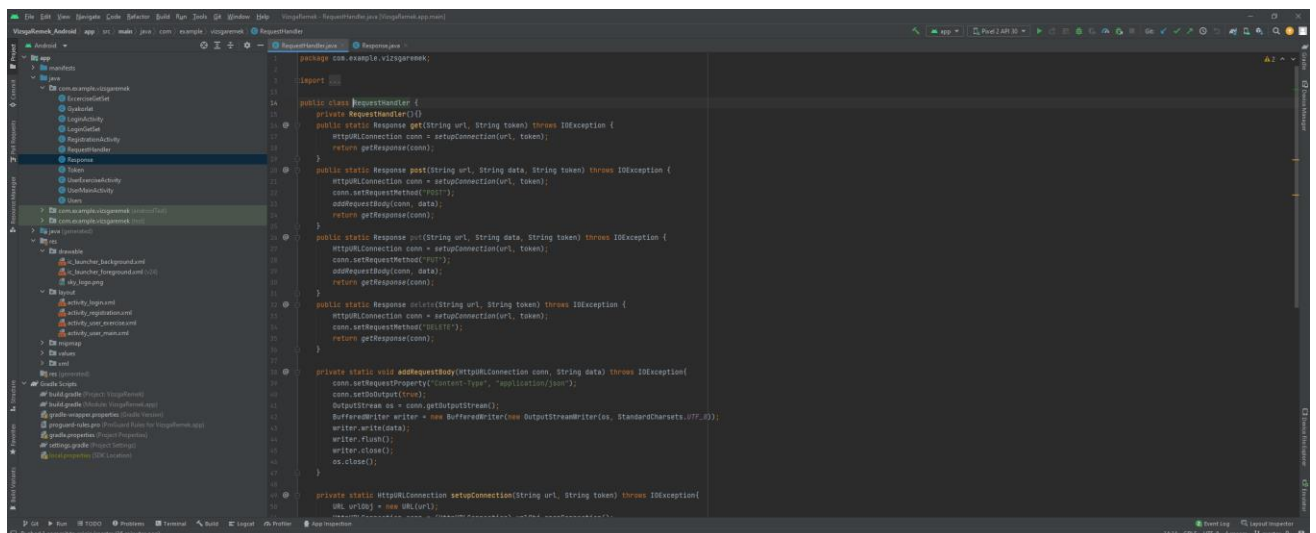


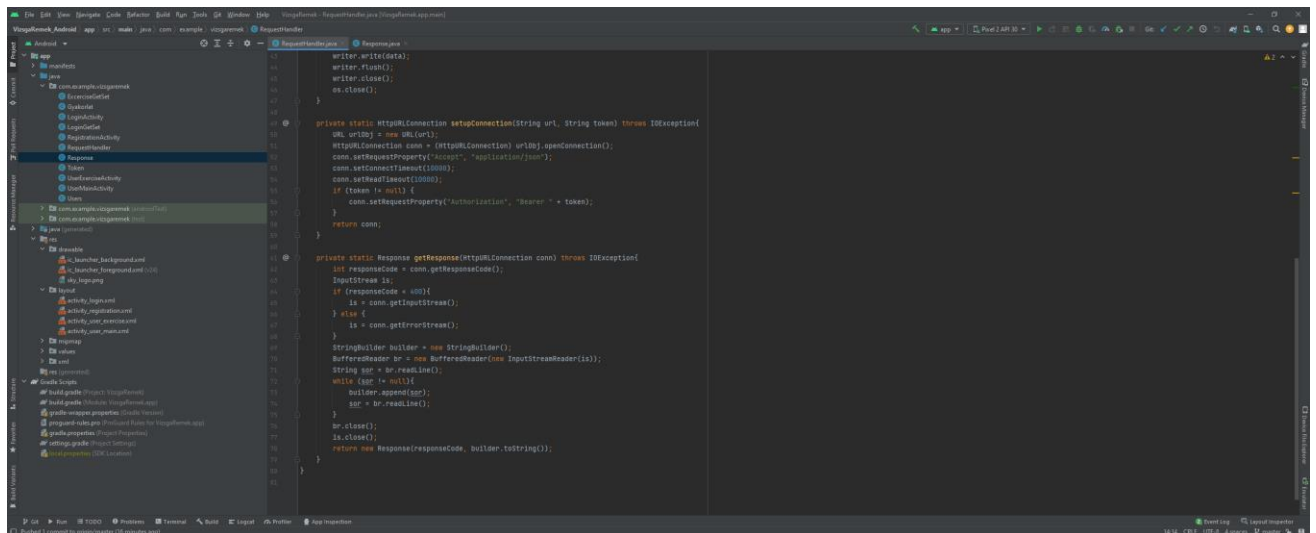
Ez a User osztály aminek a segítségével Json formátumá van alakítva a User-nek az adata. Konstruktót, Getter és Setter van benne.



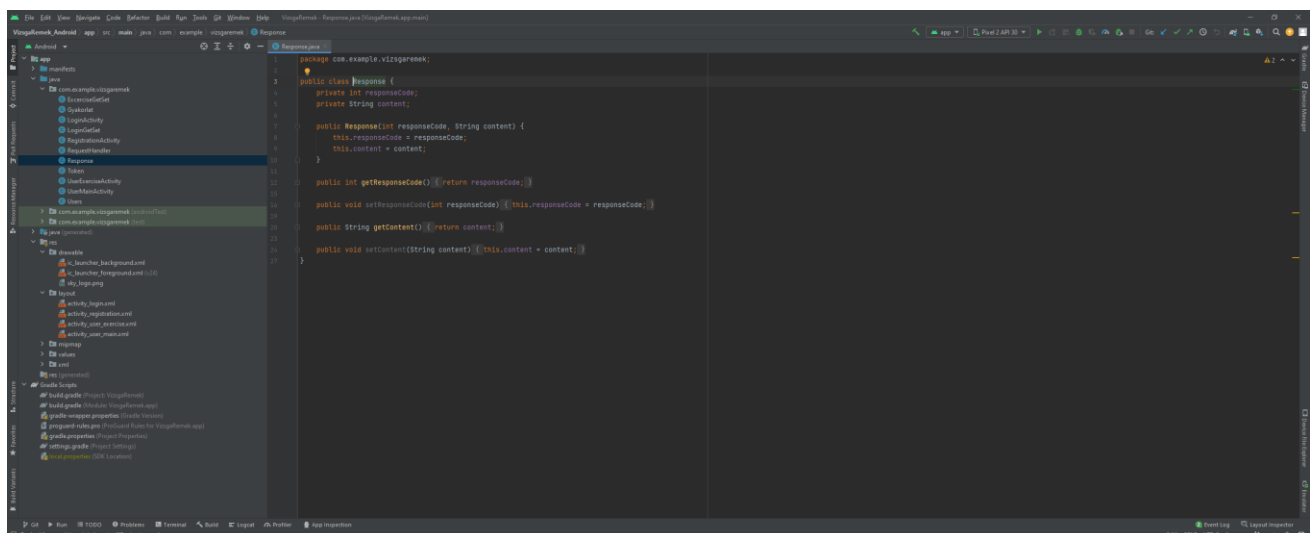
Ez a Token osztály ami a bejelentkezési a Tokent segít Json formátumúvá tenni.

RequestHandler és Response



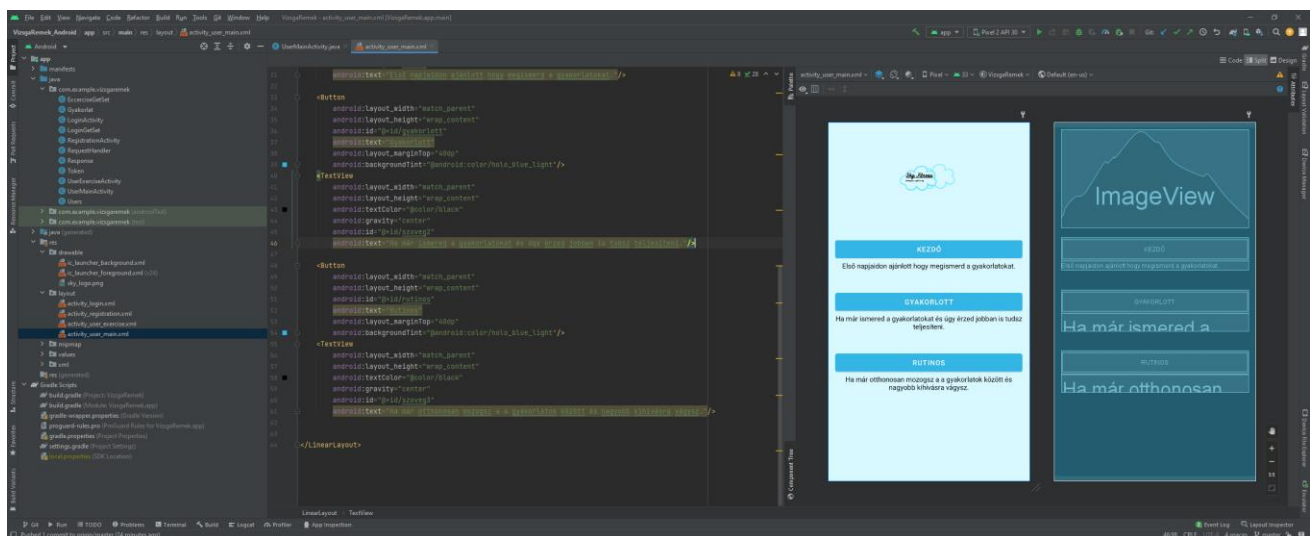


A RequestHandler a Backend felé elvégzett kéréseket kezeli.



A Response a vissza érkező választ dolgozza fel a Backendtől.

UserMainActivity

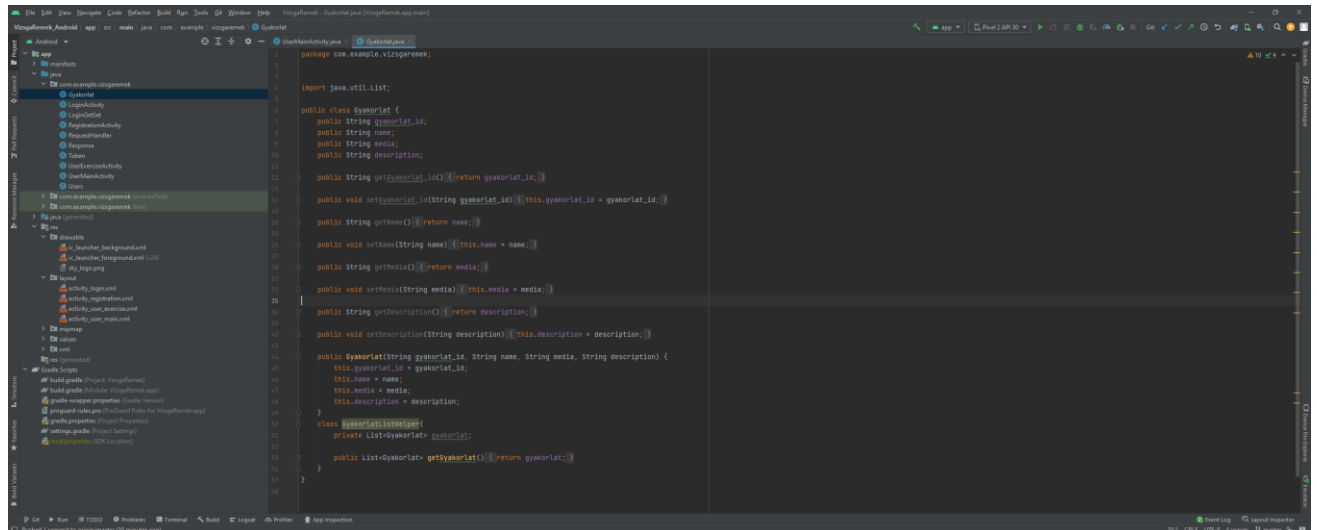


```

1 package com.example.vizgopher;
2
3 import android.appCompatActivity;
4
5 import android.content.Intent;
6 import android.os.AsyncTask;
7 import android.os.Bundle;
8 import android.view.View;
9 import android.widget.Button;
10 import android.widget.Toast;
11
12 import com.google.gson.Gson;
13
14 import java.io.IOException;
15 import java.io.InputStream;
16 import java.util.ArrayList;
17 import java.util.Calendar;
18 import java.util.List;
19
20 public class MainActivity extends AppCompatActivity {
21
22     private Number num_id = 0;
23
24     public static List<DynamicLabel> gsonList = new ArrayList<>();
25     private String BASE_URL = "http://192.168.0.1.2000/autocomplete/brand/";
26     private Button btnSearch, btnAddButton, btnDeleteButton;
27     private int day;
28
29     public void init() {
30         btnSearch = findViewById(R.id.btnSearch);
31         btnAddButton = findViewById(R.id.btnAdd);
32         btnDeleteButton = findViewById(R.id.btnDelete);
33         Calendar calendar = Calendar.getInstance();
34         int num = calendar.get(Calendar.DAY_OF_MONTH);
35
36         @Override
37         protected void onCreate(Bundle savedInstanceState) {
38             super.onCreate(savedInstanceState);
39             setContentView(R.layout.activity_main);
40             init();
41         }
42     }
43 }

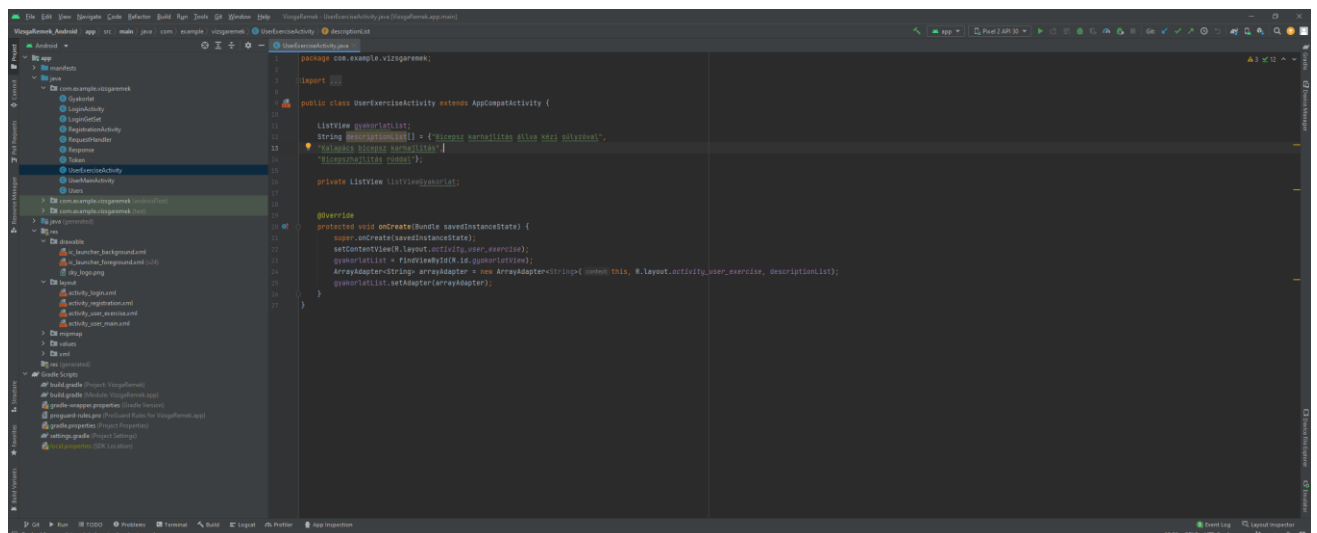
```

A UserMainActivity-nek a Java nyelven megírt fájlja. Itt miután a User kiválasztotta hogy milyen nehézségen szeretne edzeni ez egy ID-t add vissza a programnak ami a Backendbe egy JoinTableből, az adott nehézségre és napra tartozó edzéstervet adja vissza és egy listába tölti be.



Ez a Gyakorlat Osztály ami a napi edzéstervnek a megszerzését segíti meg.

UserExerciseAcvtivity



Ez az alkalmazás rész nem működik.