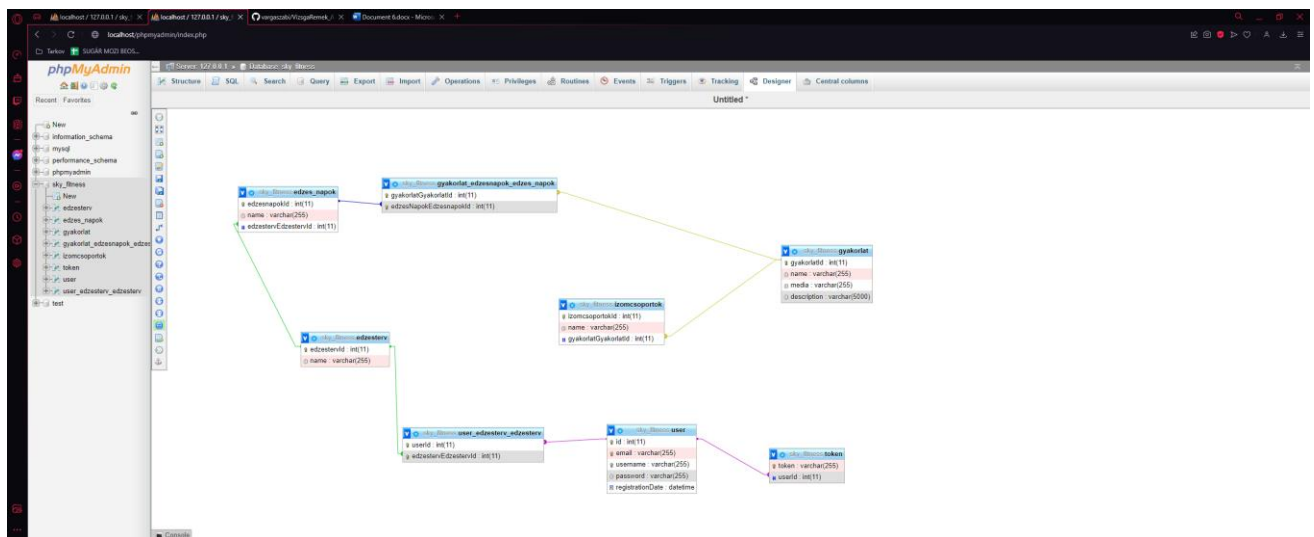
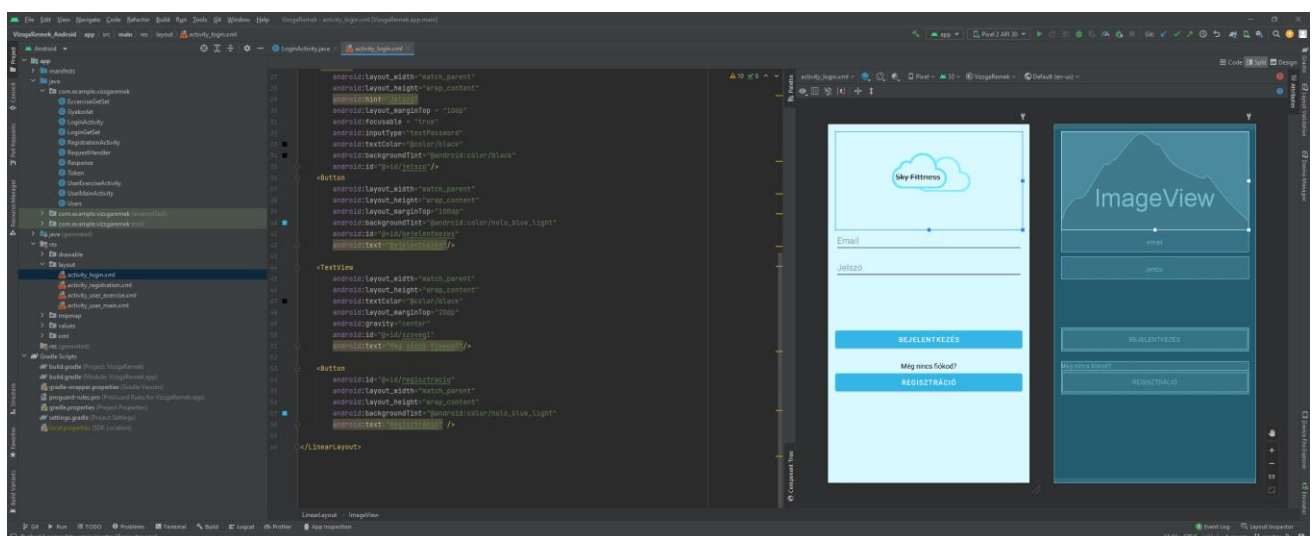
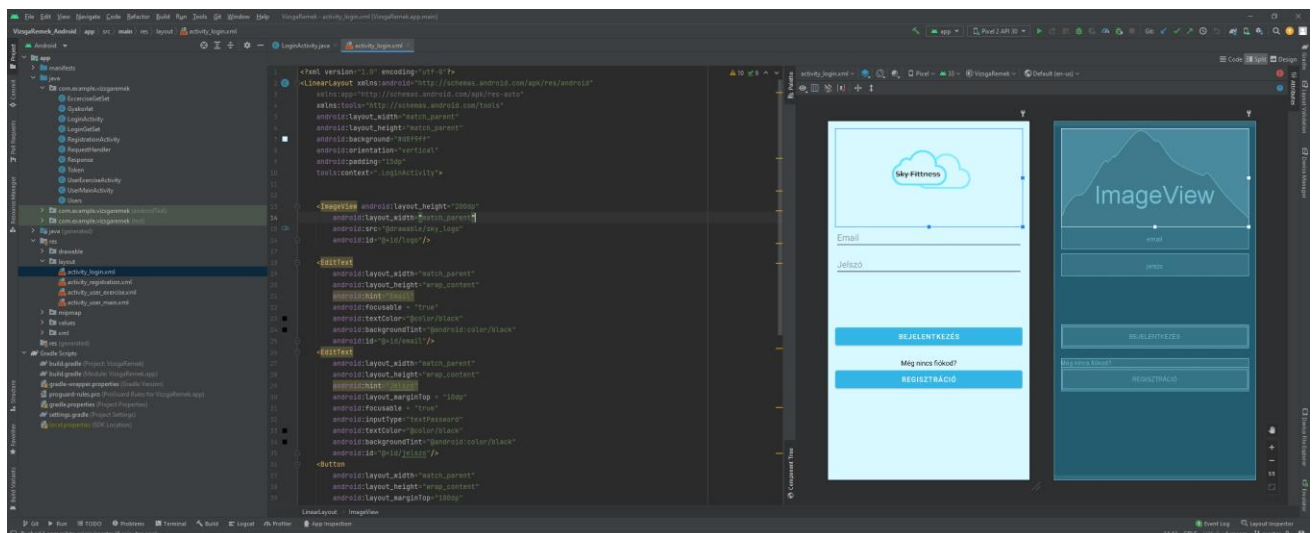


# Varga Szabolcs

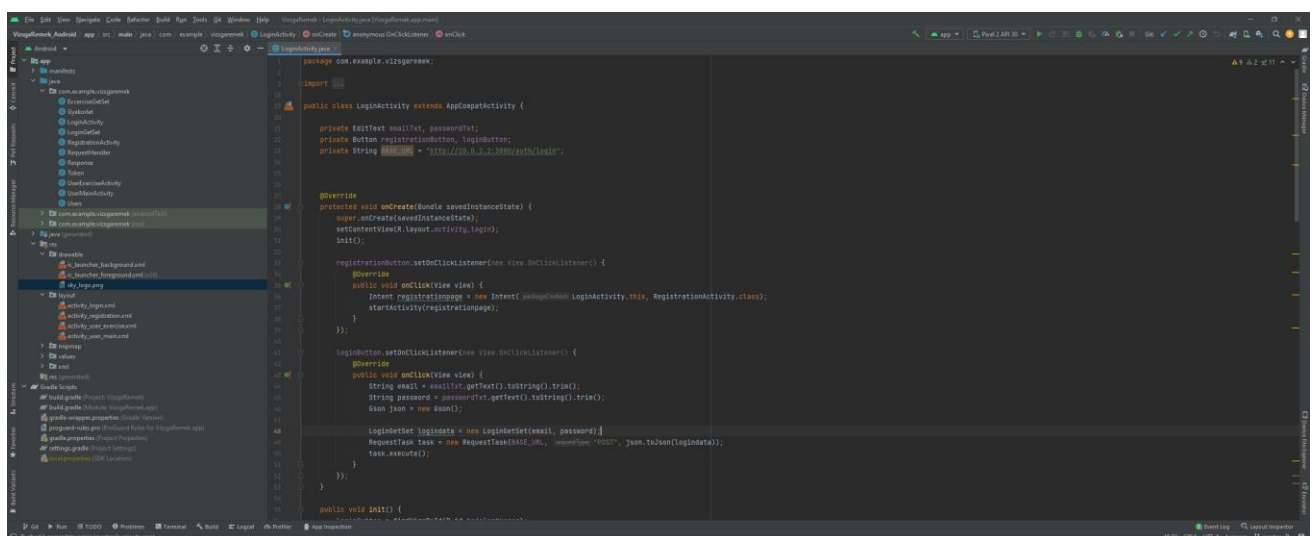


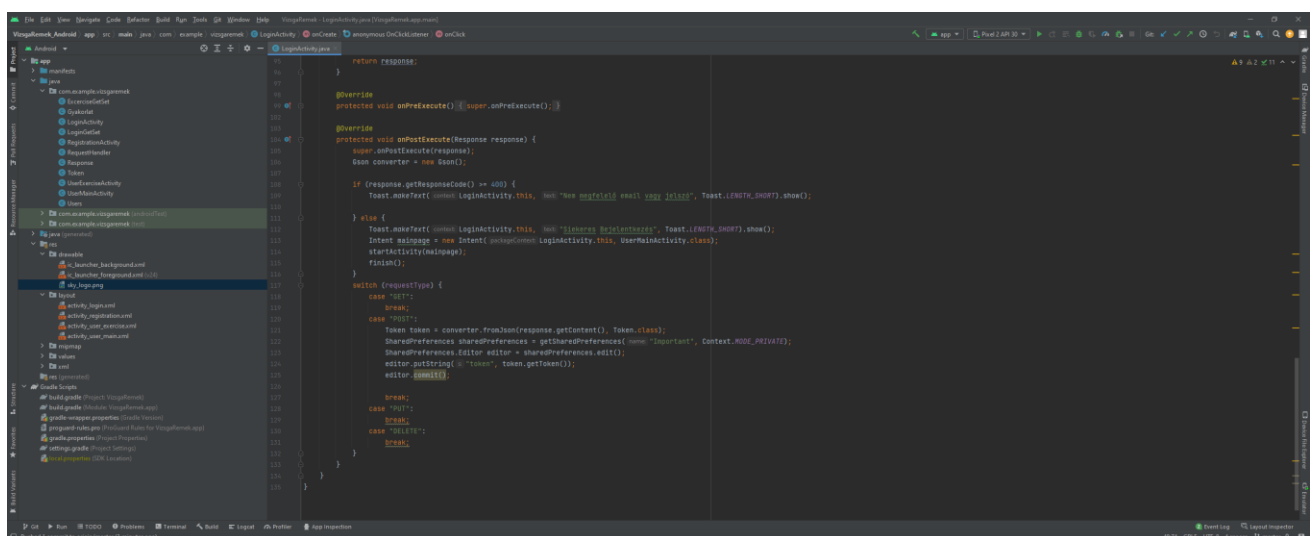
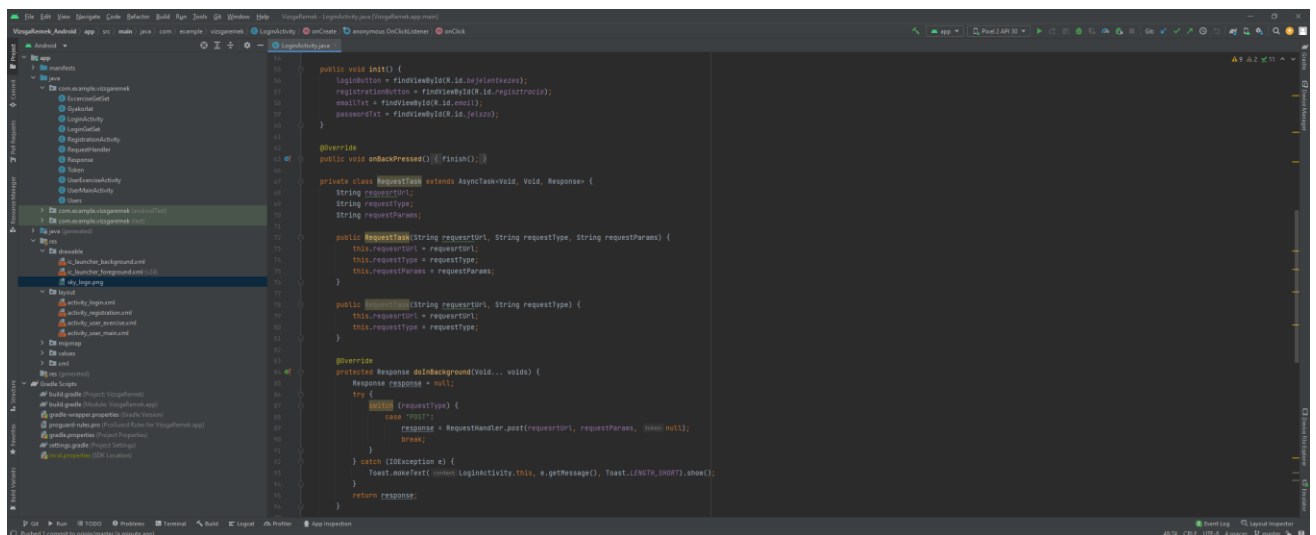
Ez az adatbázismodellje az alkalmazásnak. Ennek a pontos leírása a Backend Dokumentációba megtalálható.

## LoginActivity

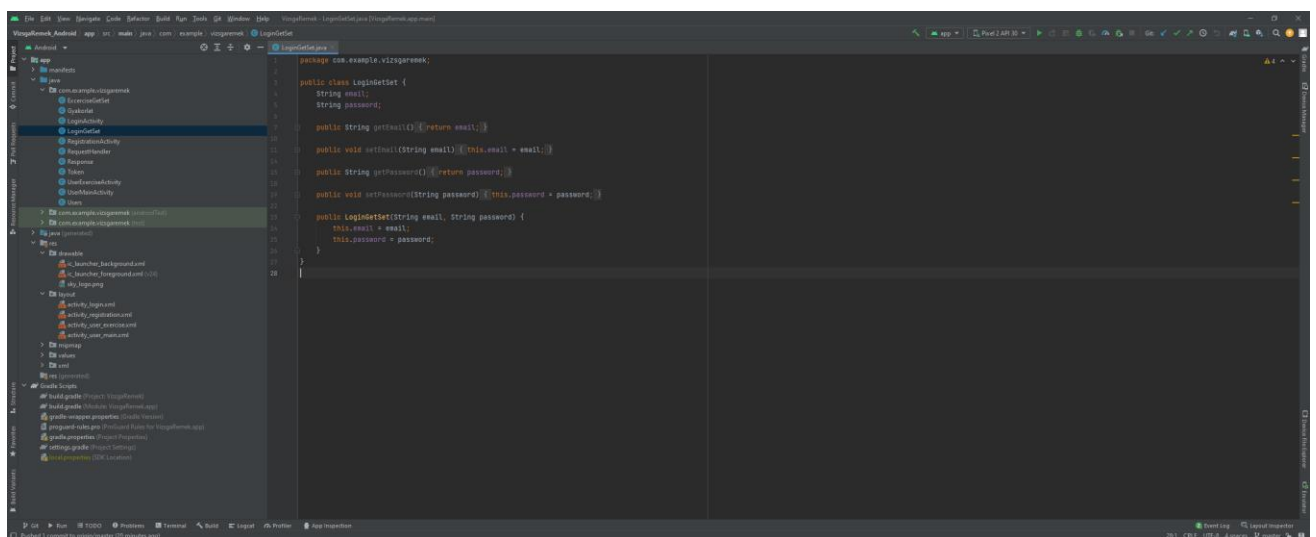


A login oldal XML fájlja. Ez a nyitóoldal az alkalmazásban innen lehet bejelentkezni vagy a regisztrációs oldalra átmenni a gombok segítségével.



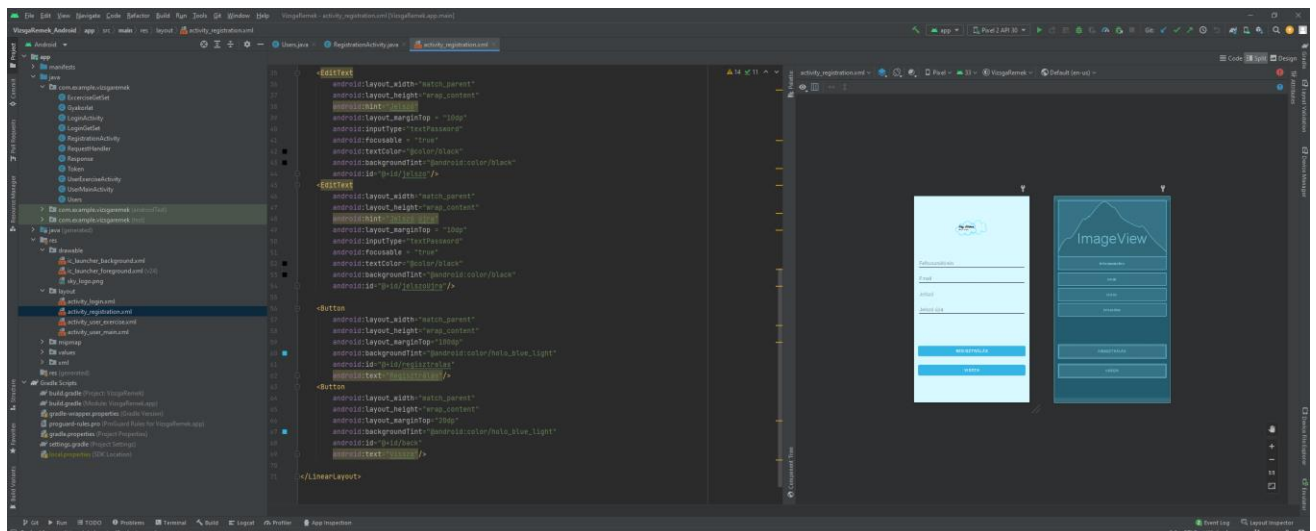
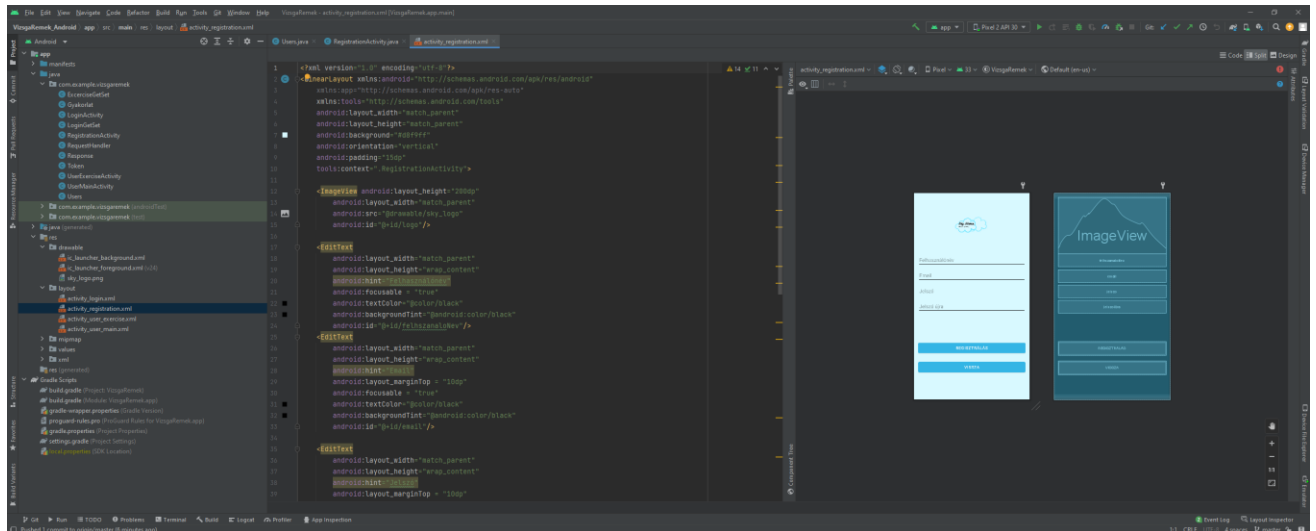


Ez maga a LoginActivity-nek a Java nyelven megírt kódja. Itt az androidon belül átalakítom az adatokat egy Json formátumba és így küldi el a Backendnek az adatot. Hibakezelés történik Frontend és Backend részen is.

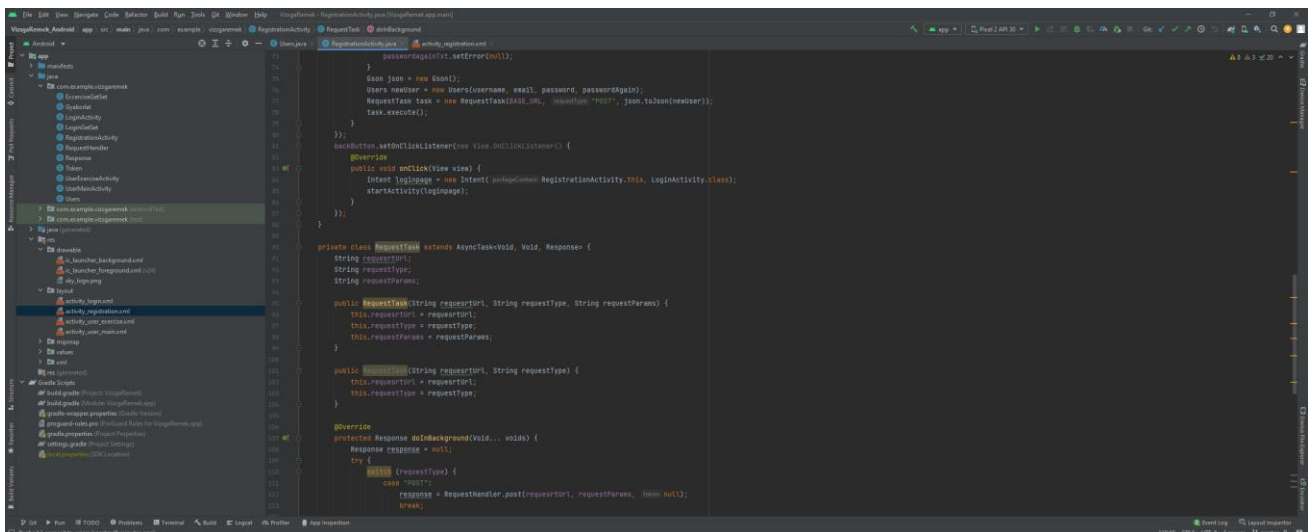
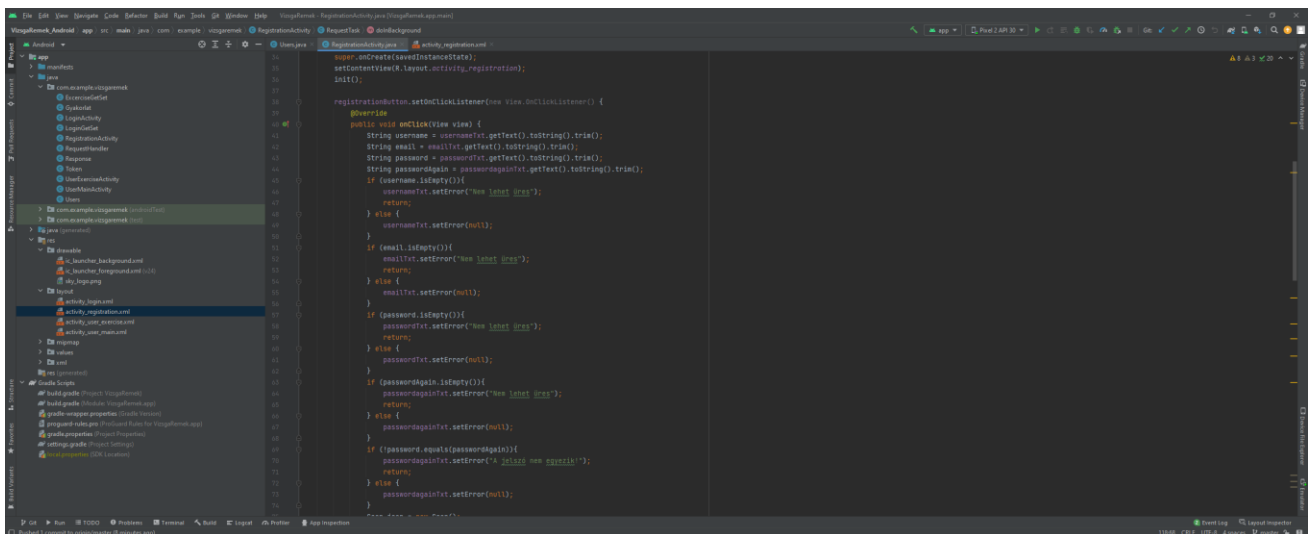
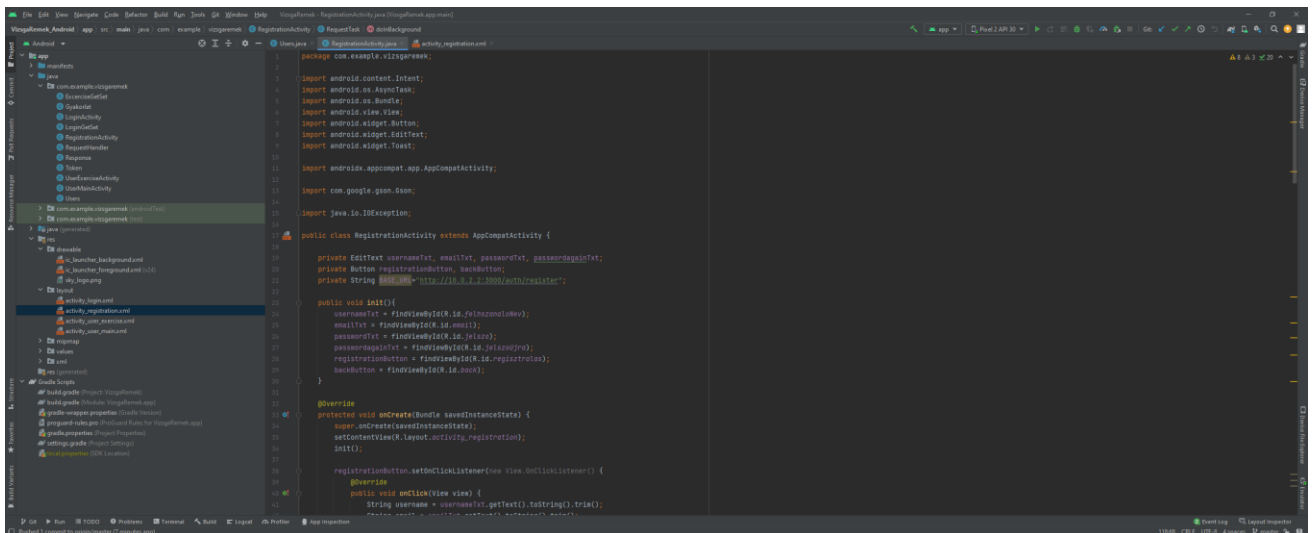


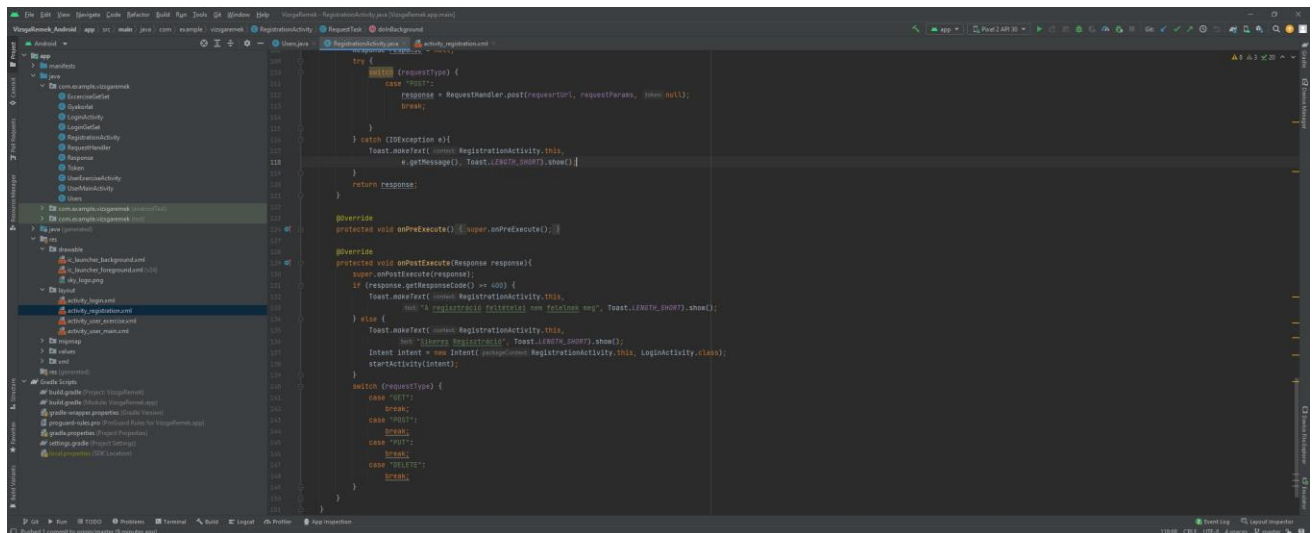
A Login osztálynak a Gettere Settere ami segít átalakítani az adatokat Json formátumra.

## RegistrationActivity



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

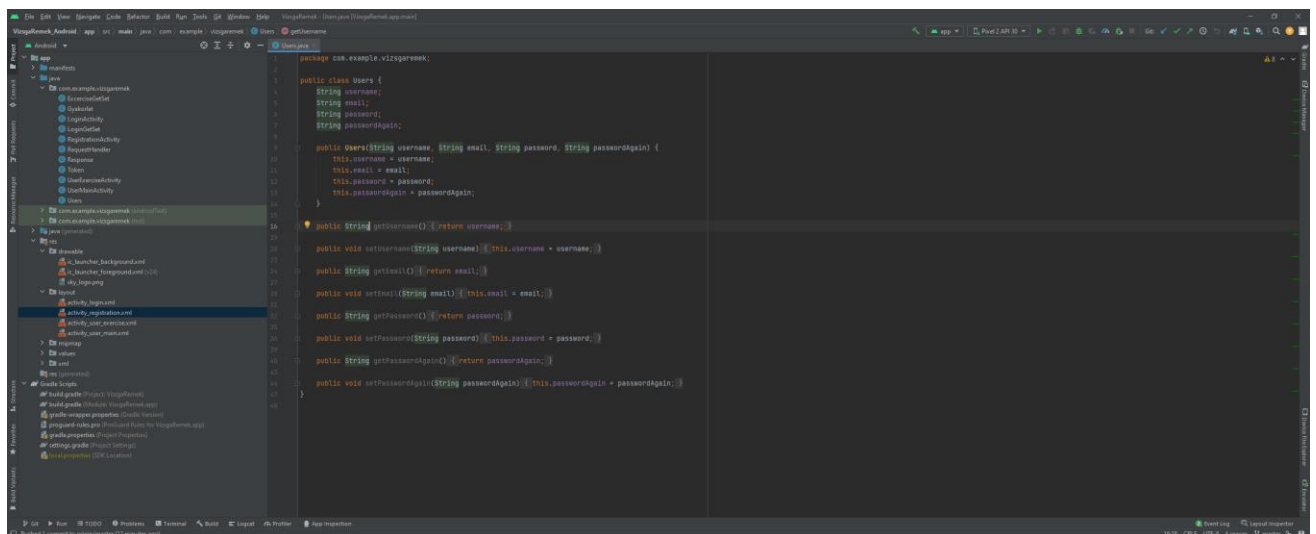




```
109 try {
110     switch (requestType) {
111         case "POST":
112             response = RequestHandler.post(requestUrl, requestParams, null);
113             break;
114     }
115     catch (IOException e) {
116         Toast.makeText(context, RegistrationActivity.this,
117             e.getMessage(), Toast.LENGTH_SHORT).show();
118     }
119     return response;
120 }
121
122 @Override
123 protected void onStart() {
124     super.onStart();
125 }
126
127 @Override
128 protected void onPostExecute(Response response) {
129     super.onPostExecute(response);
130     if (response.getResponseCode() == 400) {
131         Toast.makeText(context, RegistrationActivity.this,
132             new "A regisztráció sikertelen meg", Toast.LENGTH_SHORT).show();
133     } else {
134         Toast.makeText(context, RegistrationActivity.this,
135             new "Sikeres regisztráció", Toast.LENGTH_SHORT).show();
136         Intent intent = new Intent(context, LoginActivity.class);
137         startActivity(intent);
138     }
139     switch (requestType) {
140         case "POST":
141             String username = response.body().get("username");
142             String email = response.body().get("email");
143             String password = response.body().get("password");
144             String passwordAgain = response.body().get("passwordAgain");
145             break;
146         case "GET":
147             break;
148     }
149 }
```

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

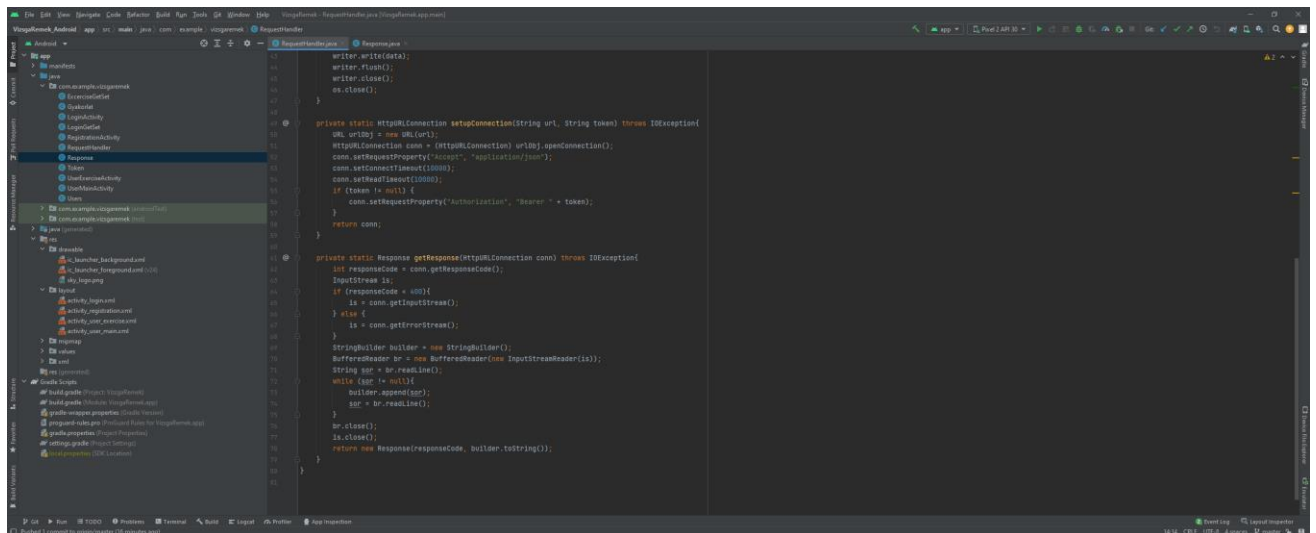


```
1 package com.example.vicsgarenek;
2
3 public class Users {
4     String username;
5     String email;
6     String password;
7     String passwordAgain;
8
9     public Users(String username, String email, String password, String passwordAgain) {
10         this.username = username;
11         this.email = email;
12         this.password = password;
13         this.passwordAgain = passwordAgain;
14     }
15
16     public String getUsername() {
17         return username;
18     }
19
20     public void setUsername(String username) {
21         this.username = username;
22     }
23
24     public String getEmail() {
25         return email;
26     }
27
28     public void setEmail(String email) {
29         this.email = email;
30     }
31
32     public String getPassword() {
33         return password;
34     }
35
36     public void setPassword(String password) {
37         this.password = password;
38     }
39
40     public String getPasswordAgain() {
41         return passwordAgain;
42     }
43
44     public void setPasswordAgain(String passwordAgain) {
45         this.passwordAgain = passwordAgain;
46     }
47 }
```

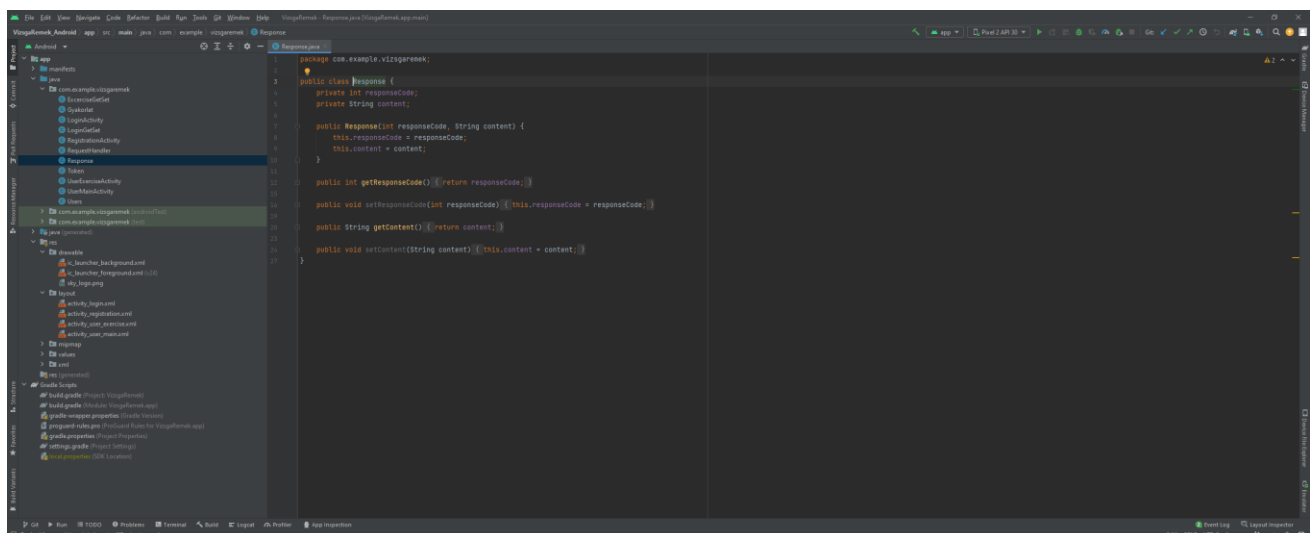
Ez a User osztály aminek a segítségével Json formátumá van alakítva a User-nek az adata. Konstruktót, Getter és Setter van benne.







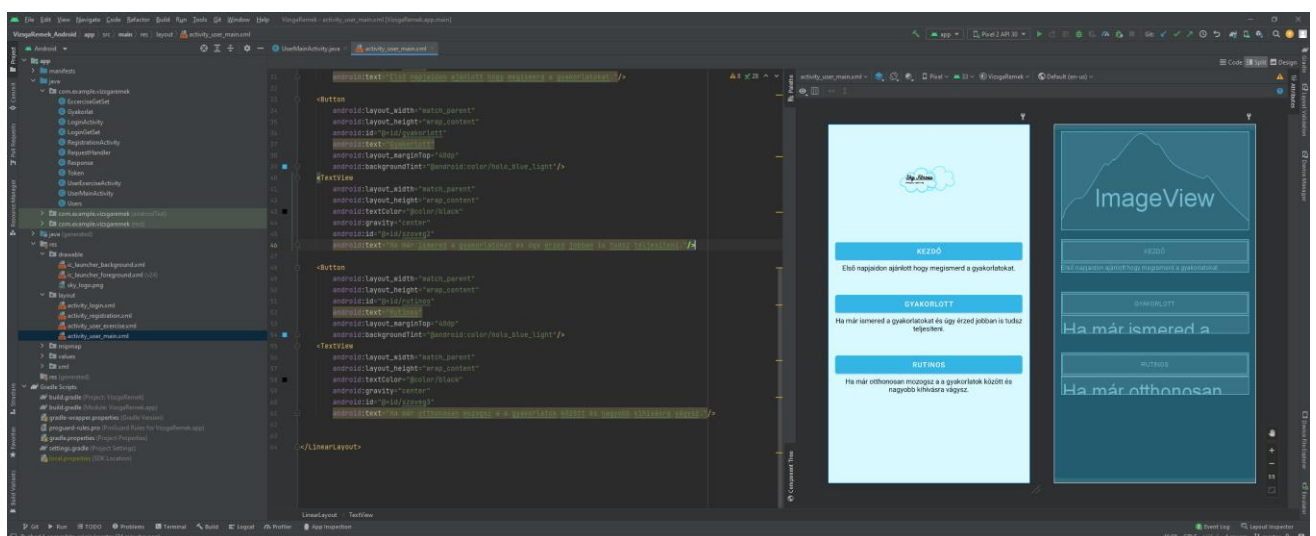
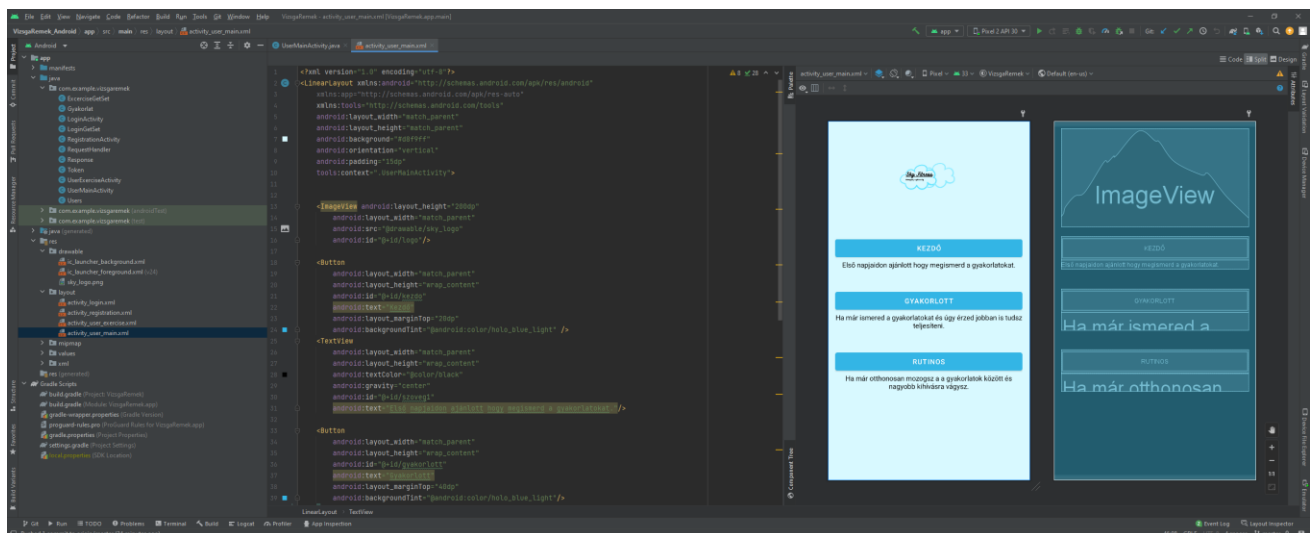
A RequestHandler a Backend felé elvégzett kéréseket kezeli.



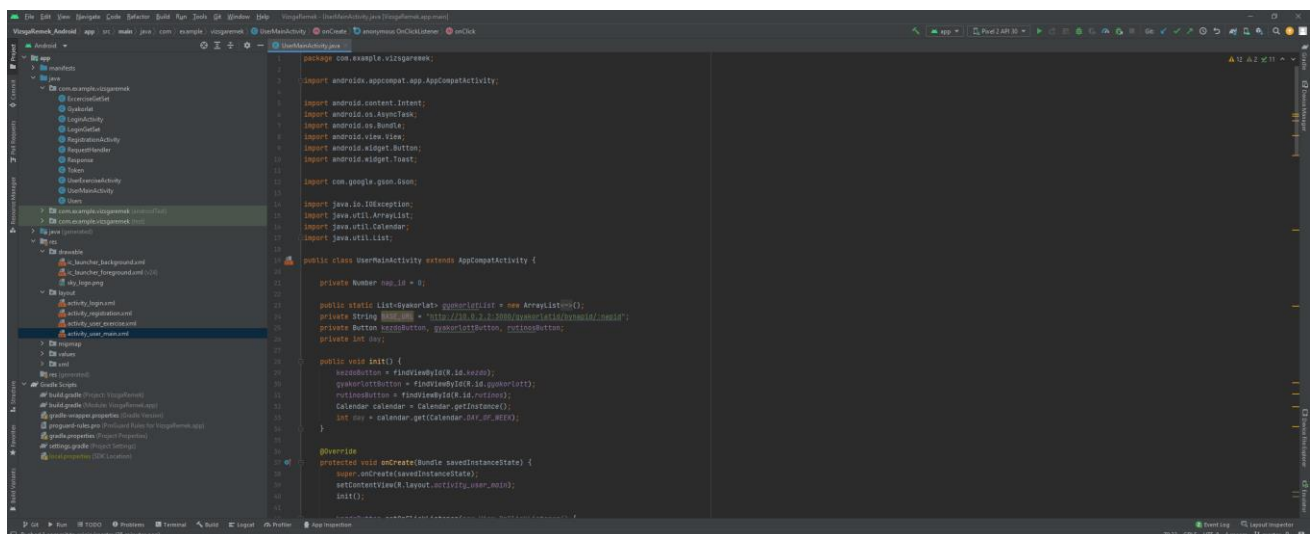
A Response a vissza érkező választ dolgozza fel a Backendtől.

## UserMainActivity

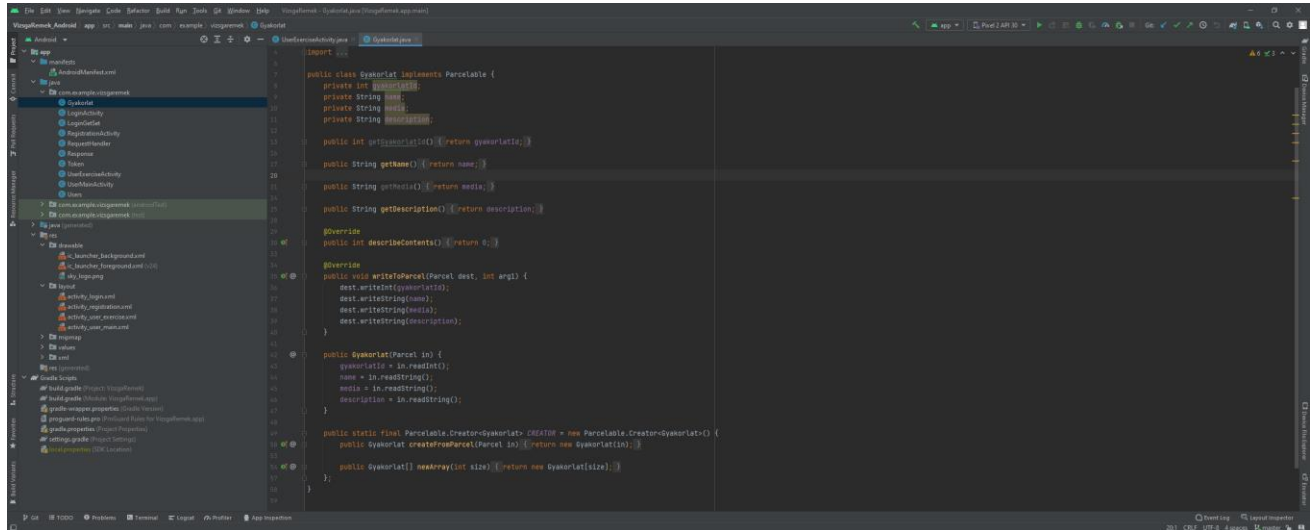




A usermainactivitynek az xml fájlja amin a felhasználó nehézséget tud választani.

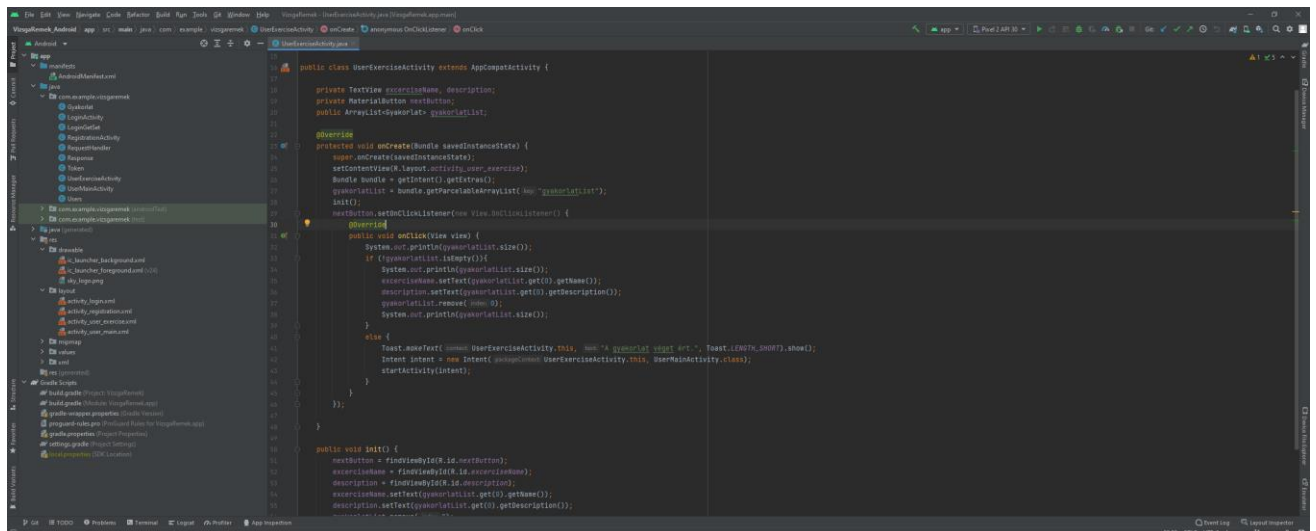


A UserMainActivity-nek a Java nyelven megírt fájlja. Itt miután a User kiválasztotta hogy milyen nehézségen szeretne edzeni ez egy ID-t ad vissza a programnak ami a Backendbe egy JoinTableből, az adott nehézségre és napra tartozó edzéstervet adja vissza és egy listába tölti be.

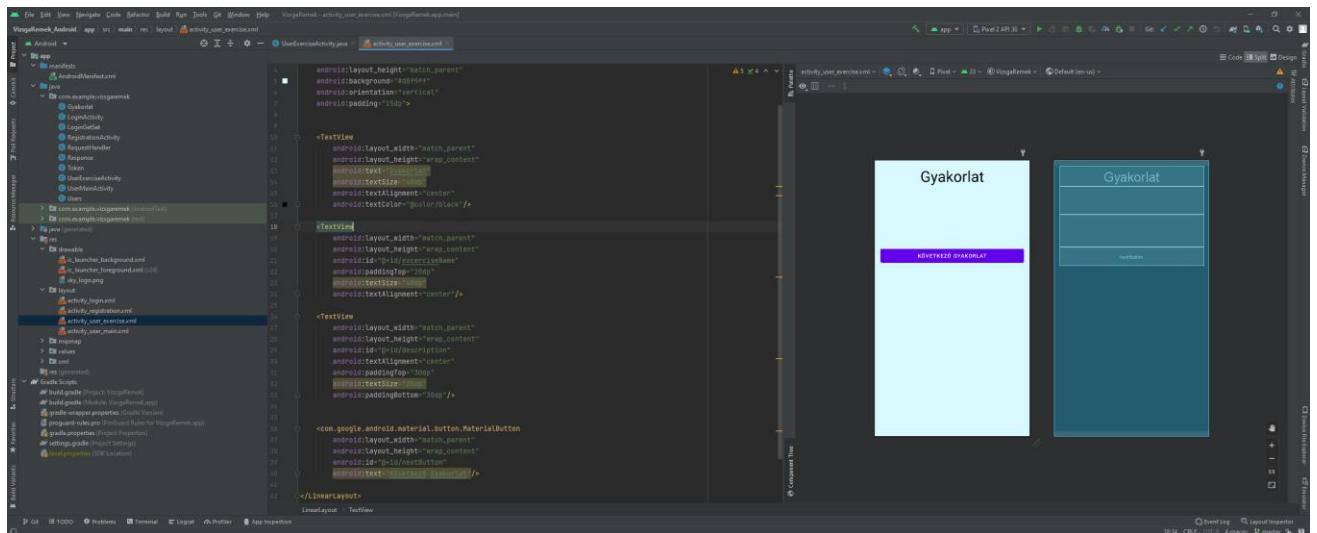


Ez a Gyakorlat Osztály ami a napi edzéstervnek a megszerzését segíti meg.

## UserExerciseAcvtivity



Ebben az activityben a UserMainActivity-ben létrehozott listát kapja meg ez az Activity és abból tölti be az éppen soron lévő gyakorlatokat egy EditText-be és a gomb segítségével lehet léptetni.



Ez itt az XML fájlja a `UserExerciseActivity`nek amibe az éppen soron következő gyakorlat jelenik meg és a gomb segítségével léptethető. Ha nincs több gyakorlat a `UserMainActivity` indul el.