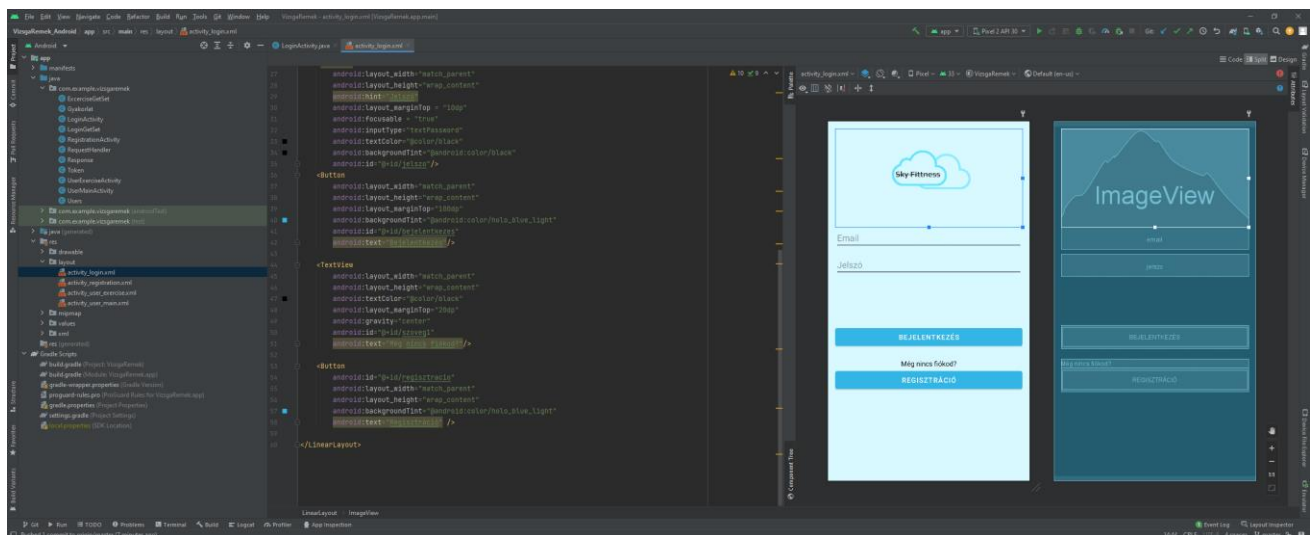
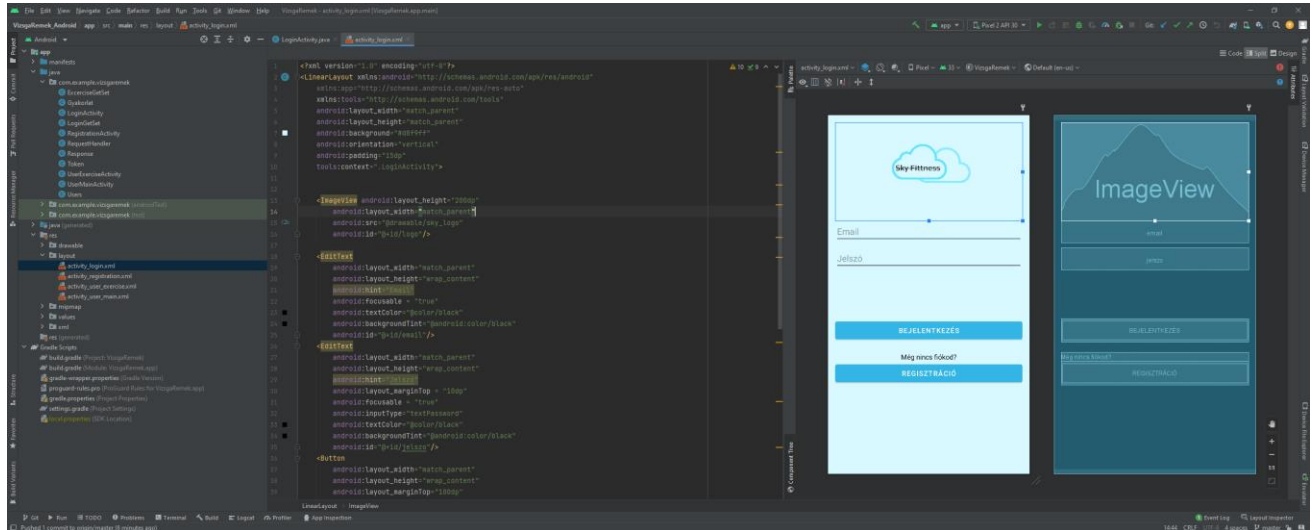


LoginActivity



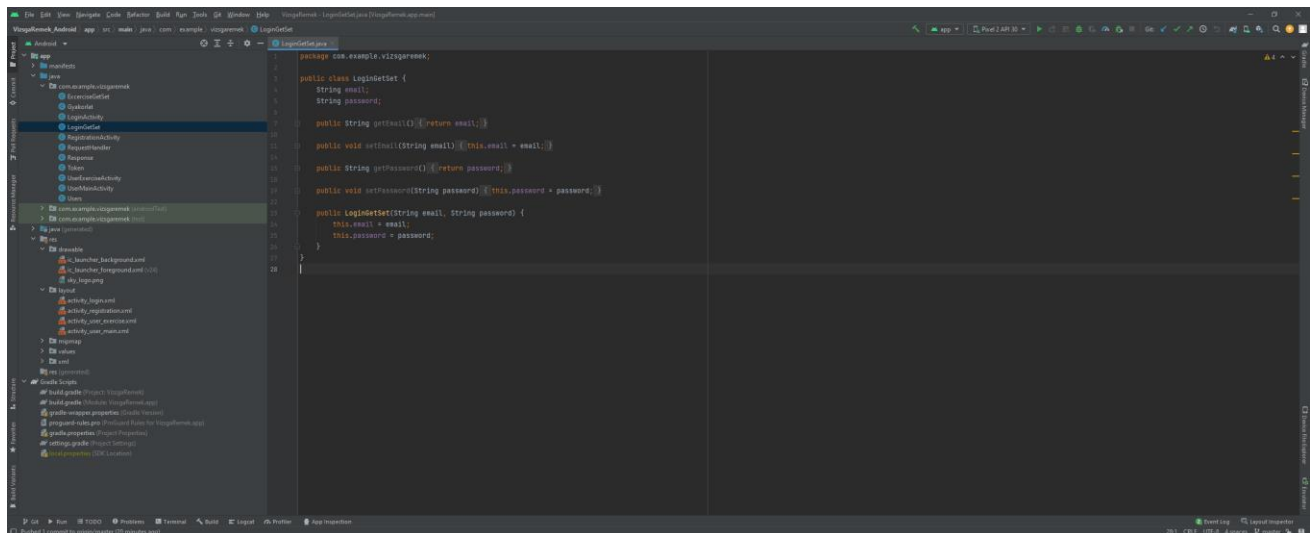
A login oldal XML fájlja. Ez a nyitóoldal az alkalmazásban innen lehet bejelentkezni vagy a regisztrációs oldalra átmenni a gombok segítségével.

```
1 package com.example.vizsgarenen;
2 import androidx.appcompat.app.AppCompatActivity;
3
4 public class LoginActivity extends AppCompatActivity {
5
6     private EditText emailEt, passwordEt;
7     private Button registrationButton, loginButton;
8     private String baseUrl = "http://192.168.1.1000:8080/login";
9
10    @Override
11    protected void onCreate(Bundle savedInstanceState) {
12        super.onCreate(savedInstanceState);
13        setContentView(R.layout.activity_login);
14        init();
15
16        registrationButton.setOnClickListener(new View.OnClickListener() {
17            @Override
18            public void onClick(View view) {
19                Intent registrationPage = new Intent(LoginActivity.this, RegistrationActivity.class);
20                startActivity(registrationPage);
21            }
22        });
23
24        loginButton.setOnClickListener(new View.OnClickListener() {
25            @Override
26            public void onClick(View view) {
27                String email = emailEt.getText().toString().trim();
28                String password = passwordEt.getText().toString().trim();
29                Gson gson = new Gson();
30
31                LoginRequest loginData = new LoginRequest(email, password);
32                RequestTask task = new RequestTask(baseUrl, "POST", gson.toJson(loginData));
33                task.execute();
34            }
35        });
36
37    public void init() {
38        loginButton = findViewById(R.id.loginButton);
39        registrationButton = findViewById(R.id.registrationButton);
40        emailEt = findViewById(R.id.email);
41        passwordEt = findViewById(R.id.password);
42    }
43}
```

```
44 public void init() {
45     loginButton = findViewById(R.id.loginButton);
46     registrationButton = findViewById(R.id.registrationButton);
47     emailEt = findViewById(R.id.email);
48     passwordEt = findViewById(R.id.password);
49 }
50
51 @Override
52 public void onBackPressed() { finish(); }
53
54 private class RequestTask extends AsyncTask<Void, Void, Response> {
55     String requestURL;
56     String requestType;
57     String requestParams;
58
59     public RequestTask(String requestURL, String requestType, String requestParams) {
60         this.requestURL = requestURL;
61         this.requestType = requestType;
62         this.requestParams = requestParams;
63     }
64
65     public Response execute(String requestURL, String requestType) {
66         this.requestURL = requestURL;
67         this.requestType = requestType;
68     }
69
70     @Override
71     protected Response doInBackground(Void... voids) {
72         Response response = null;
73         try {
74             response = new RequestTask(requestURL, requestType, requestParams).execute();
75             break;
76         } catch (IOException e) {
77             Toast.makeText(LoginActivity.this, e.getMessage(), Toast.LENGTH_SHORT).show();
78         }
79         return response;
80     }
81 }
```

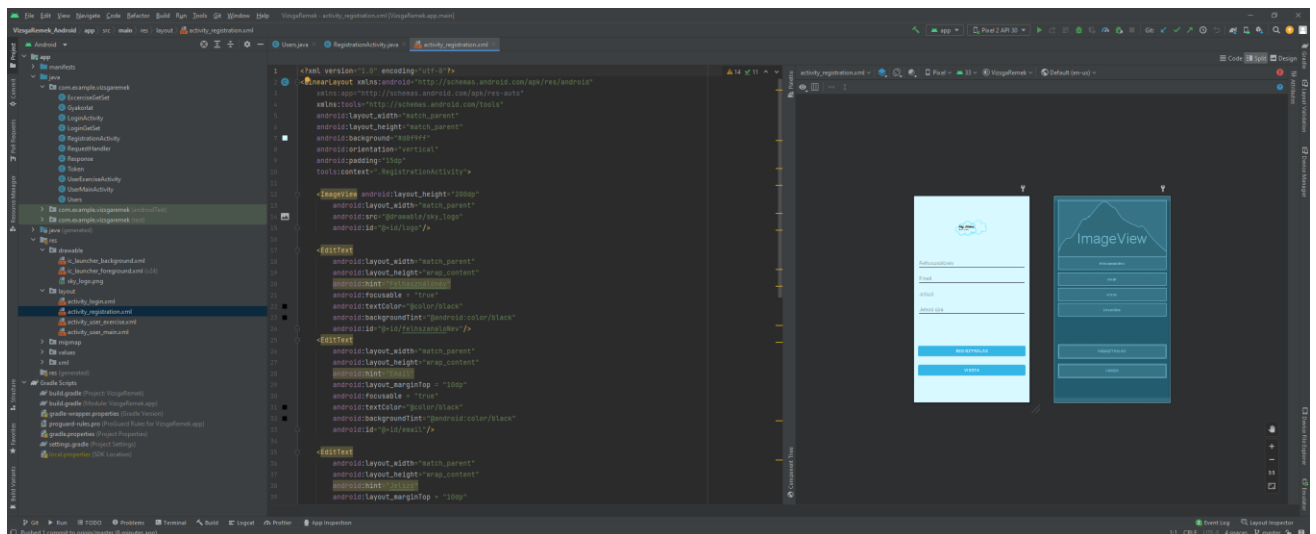
```
82 return response;
83
84 @Override
85 protected void onPostExecute() { super.onPostExecute(); }
86
87 @Override
88 protected void onPostExecute(Response response) {
89     super.onPostExecute(response);
90     Gson gson = new Gson();
91
92     if (response.getStatusCode() == 400) {
93         Toast.makeText(LoginActivity.this, "Hiba! Nem megfelelő email vagy jelszó", Toast.LENGTH_SHORT).show();
94     } else {
95         Toast.makeText(LoginActivity.this, "Sikeres bejelentkezés", Toast.LENGTH_SHORT).show();
96         Intent mainPage = new Intent(LoginActivity.this, MainActivity.class);
97         startActivity(mainPage);
98         finish();
99     }
100
101     switch (requestType) {
102         case "GET":
103             break;
104         case "POST":
105             Token token = converter.fromJson(response.getContent(), Token.class);
106             SharedPreferences sharedPreferences = getSharedPreferences("login", Context.MODE_PRIVATE);
107             SharedPreferences.Editor editor = sharedPreferences.edit();
108             editor.putString("token", token.getToken());
109             editor.commit();
110             break;
111         case "PUT":
112             break;
113         case "DELETE":
114             break;
115     }
116 }
```

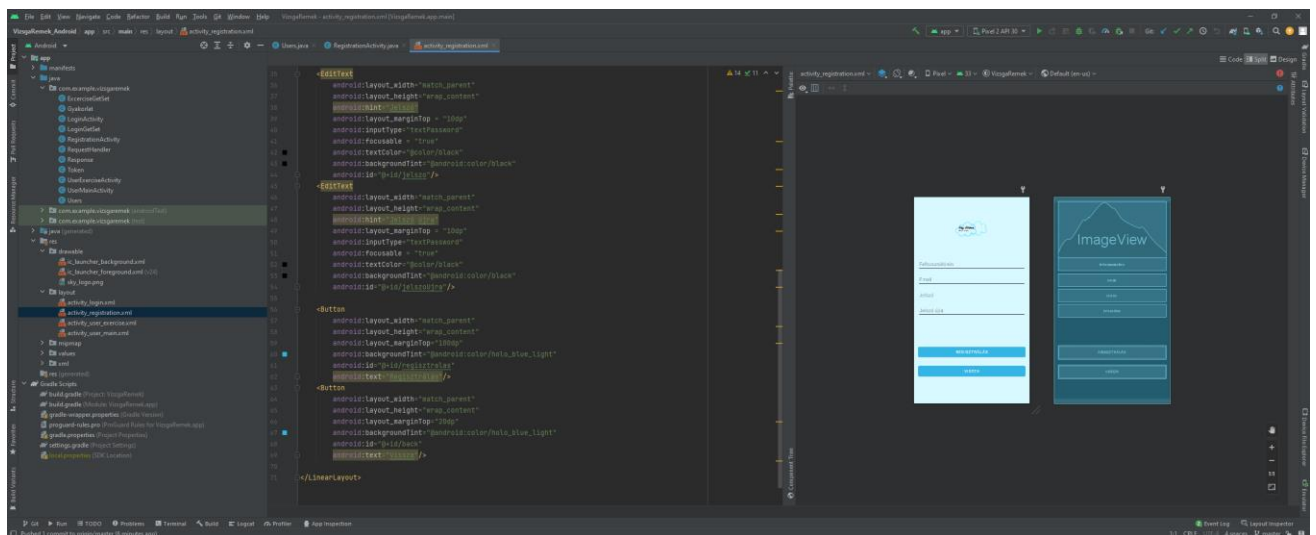
Ez maga a LoginActivity-nek a Java nyelven megírt kódja. Itt az androidon belül átalakítom az adatokat egy Json formátumba és így küldi el a Backendnek az adatot. Hibakezelés történik Frontend és Backend részen is.



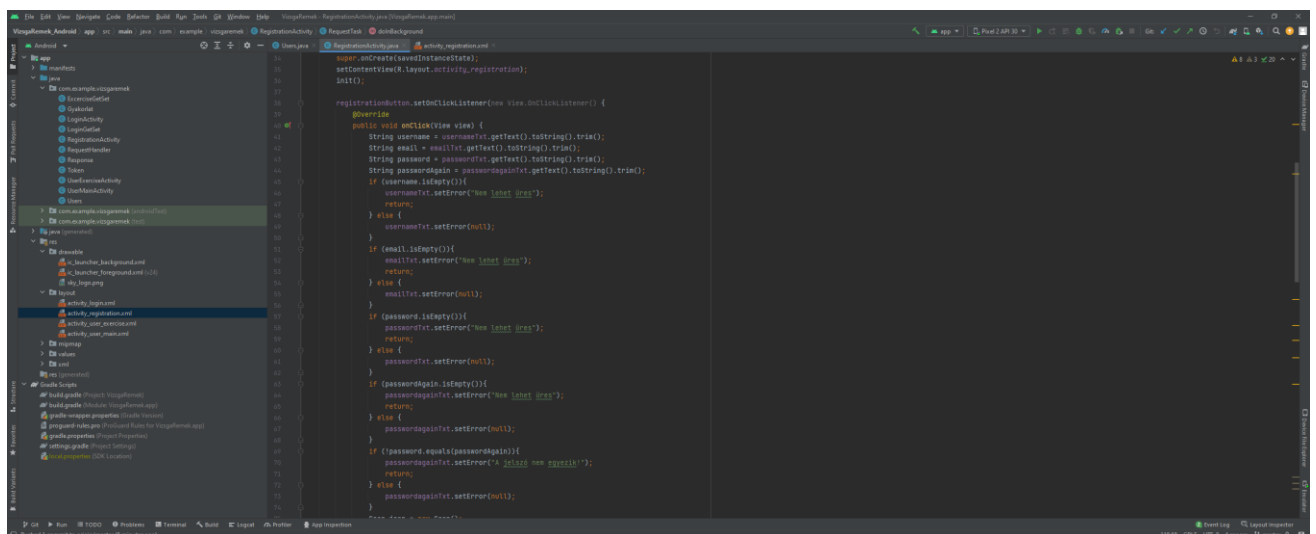
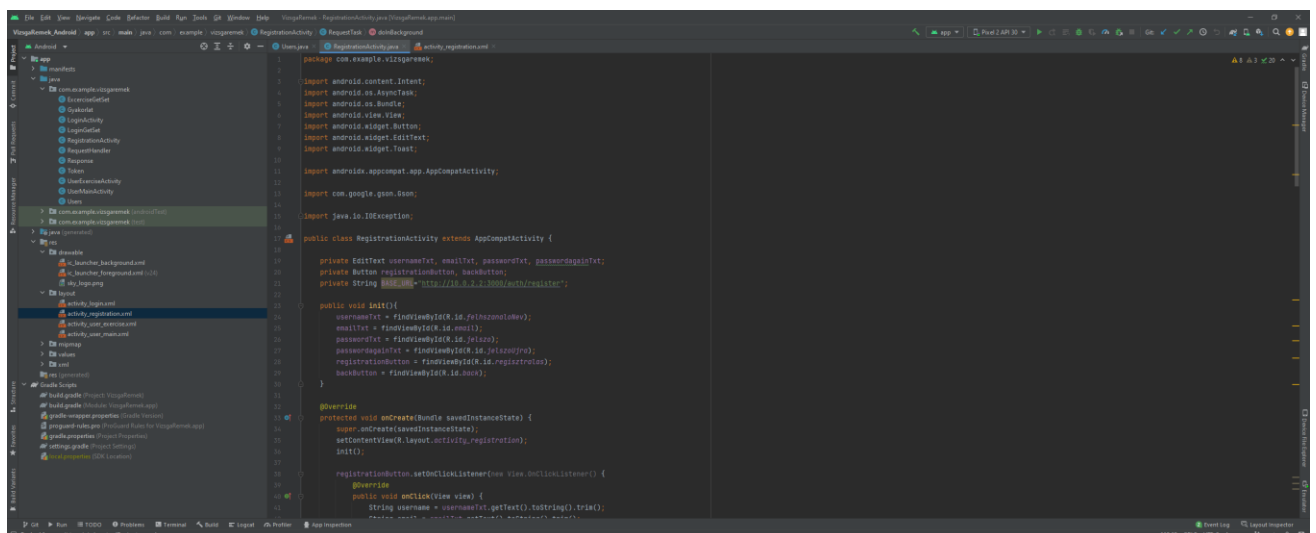
A Login osztálynak a Gettere Settere ami segít átalakítani az adatokat Json formátumra.

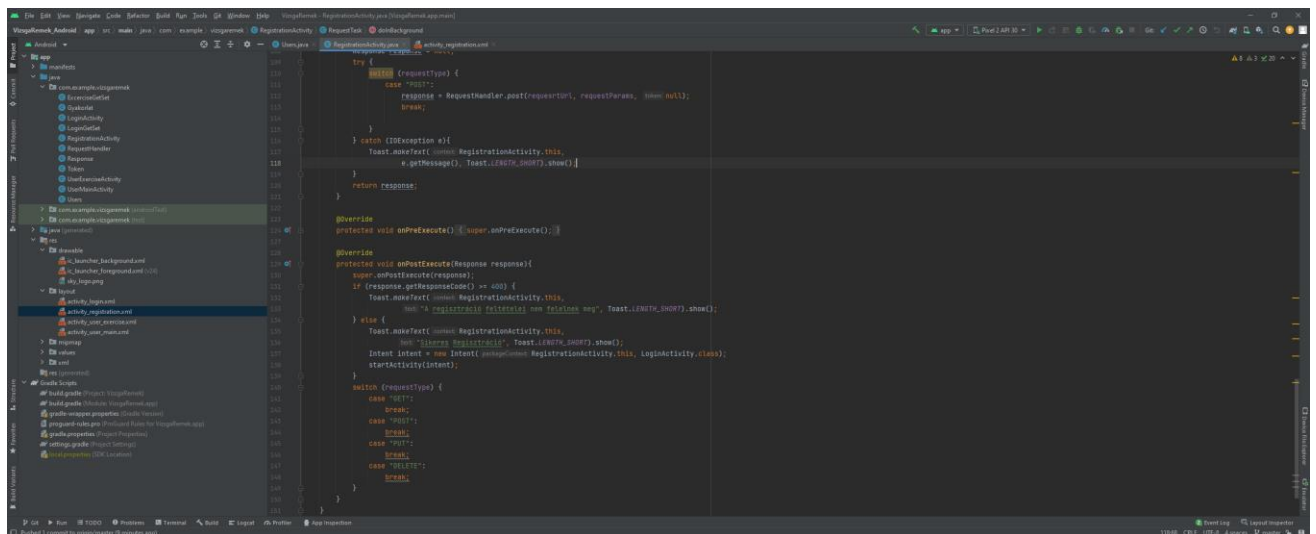
RegistrationActivity





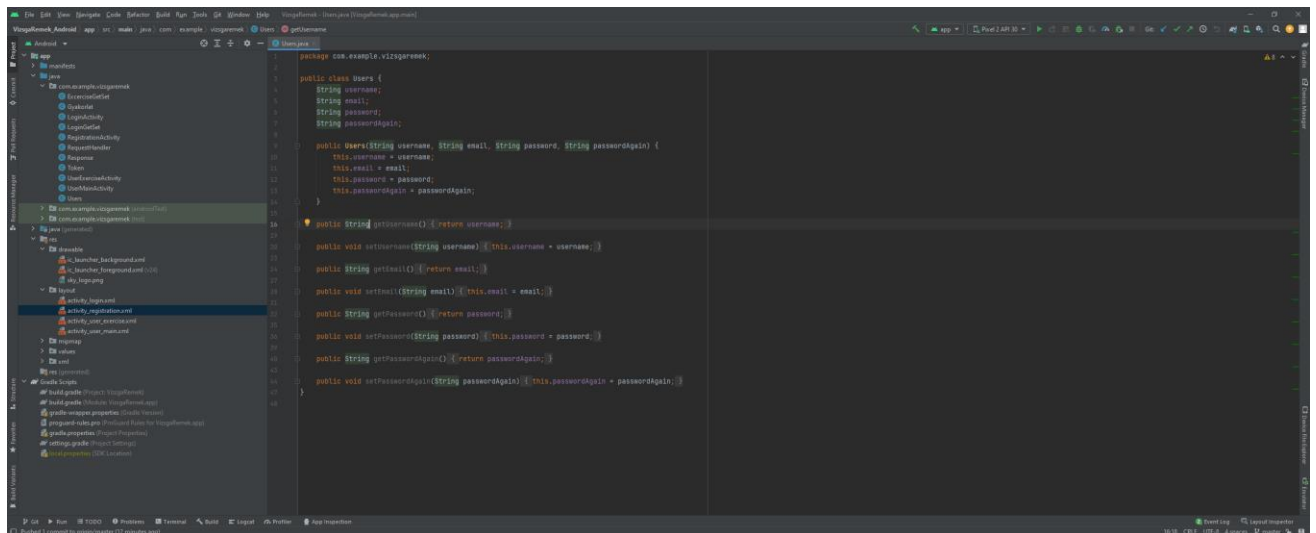
Ez a regisztráció oldalnak az xml fájlja. Itt is a gombok segítségével lehet navigálni.



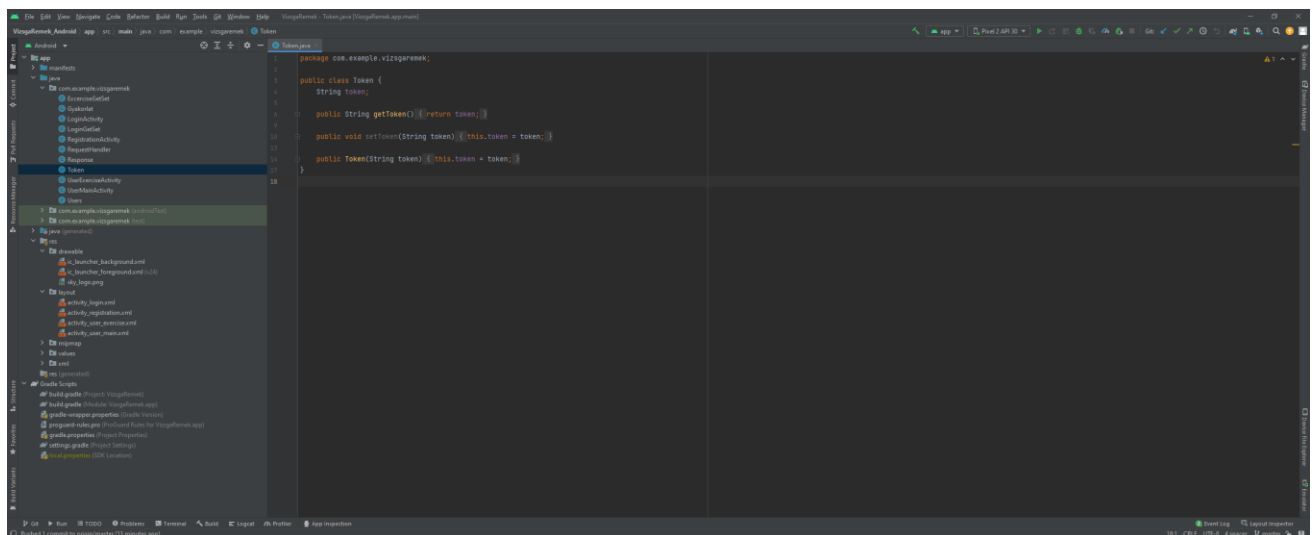


User Class és Token Class

User Class és Token Class

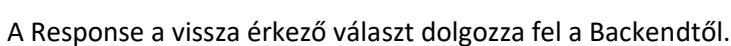
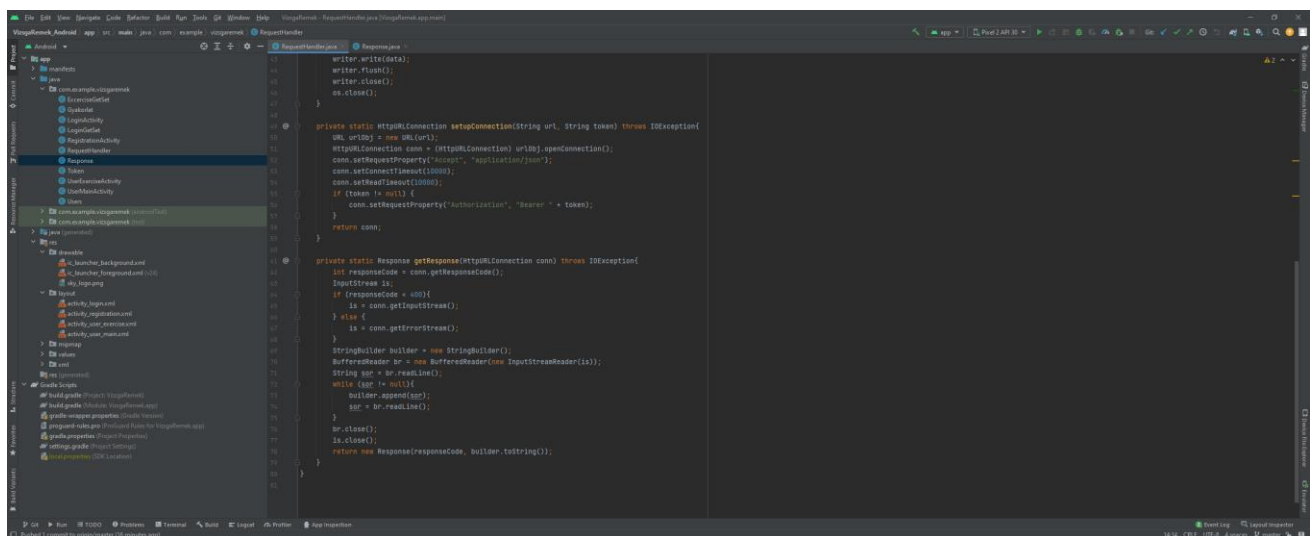


Ez a User osztály aminek a segítségével Json formátumá van alakítva a User-nek az adata.
Konstruktót, Getter és Setter van benne.

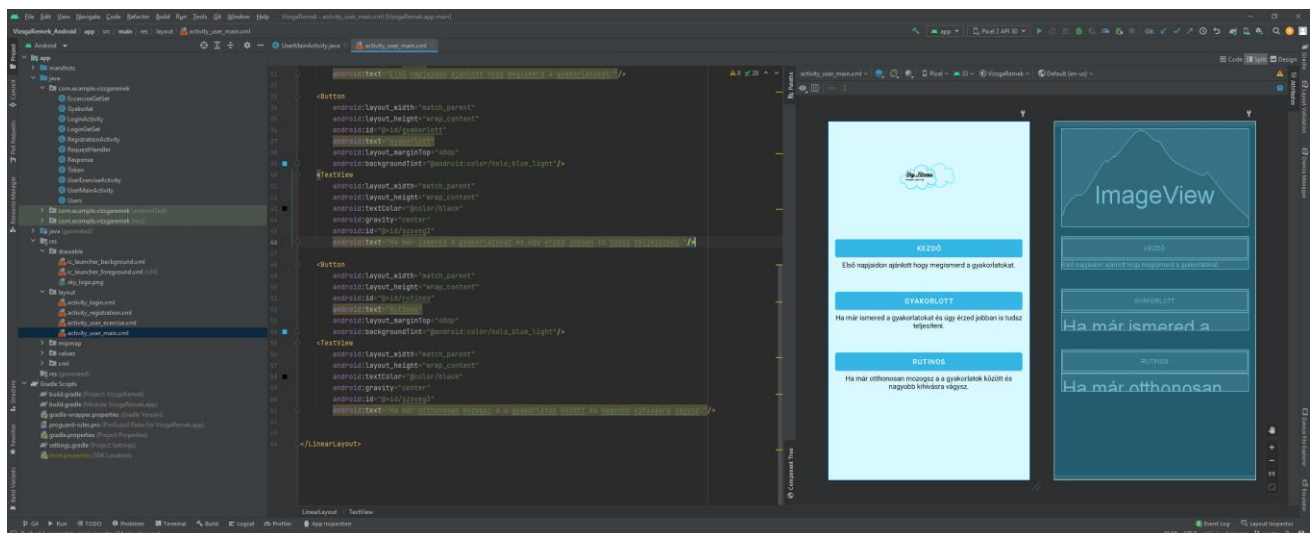
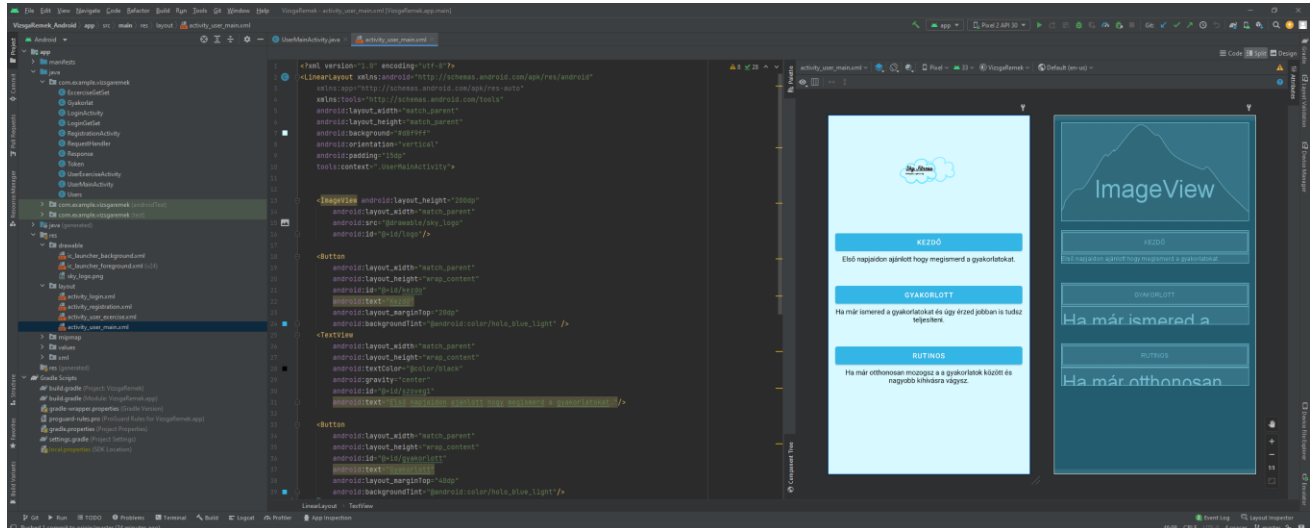


Ez a Token osztály ami a bejelentkezési a Tokent segít Json formátumúvá tenni.

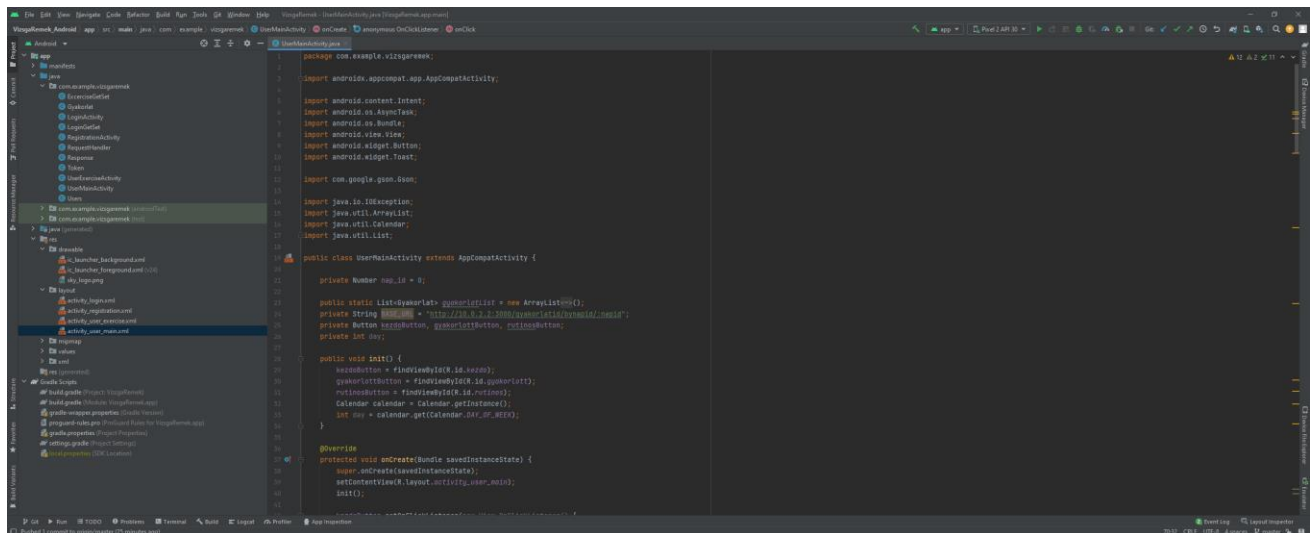
RequestHandler és Response



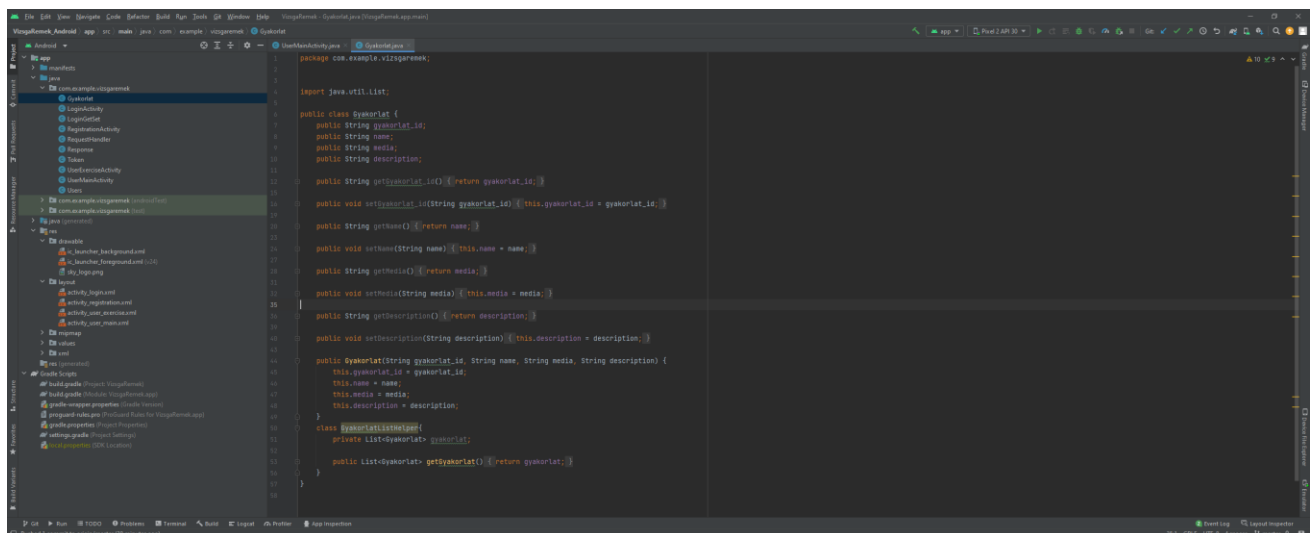
UserMainActivity



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

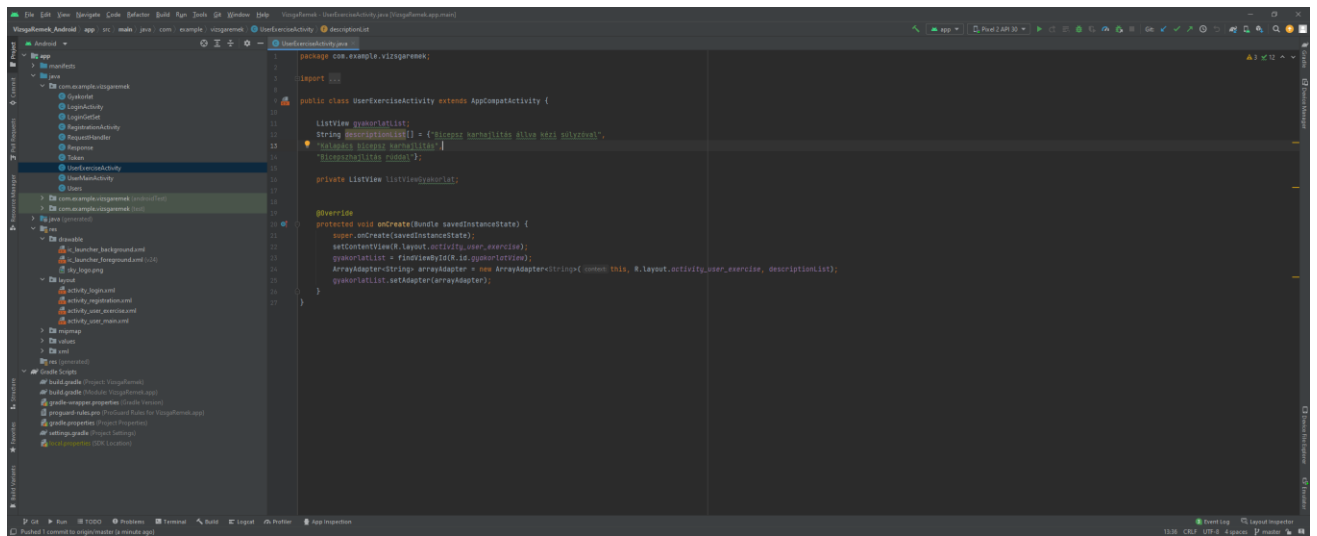


A UserMainActivity-nek a Java nyelven megírt fájlja. Itt miután a User kiválasztotta hogy milyen nehézségen szeretne edzeni ez egy ID-t add vissza a programnak ami a Backendbe egy JoinTableből, az adott nehézségre és napra tartozó edzéstervet adja vissza és egy listába tölti be.



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

UserExerciseAcvtivity



Ez az alkalmazás rész nem működik.