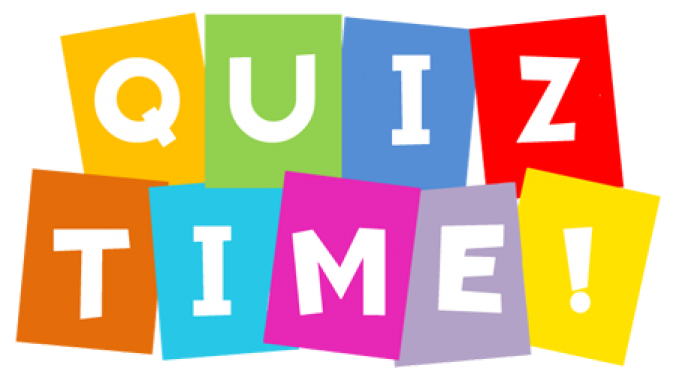
SAPIENTIA EMTE

MAROSVÁSÁRHELY



**ANDROID QUIZ APP**

**Készítette: Tanár:**

Izsák Andrea Szántó Zoltán

SZOFTVERRENDSZEREK TERVEZÉSE

2020

**TARTALOMJEGYZÉK**

[1. BEVEZETÉS 3](#_Toc58234104)

[1.1. Célkitűzések 3](#_Toc58234105)

[2. KÖVETELMÉNYEK 3](#_Toc58234106)

[2.1. Felhasználói követelmények 3](#_Toc58234107)

[2.2. Nem funkcionális követelmények 4](#_Toc58234108)

[2.3.Funkcionális követelmények 4](#_Toc58234109)

[3. IMPLEMENTÁCIÓ 10](#_Toc58234110)

[3.1. A projekt elkészítésének lépései 10](#_Toc58234111)

[4. OSZTÁLYDIAGRAMOK (UML) 12](#_Toc58234112)

[5. ÖSSZEFOGLALÓ 13](#_Toc58234113)

[5.1. JÖVŐBELI TERVEK 13](#_Toc58234114)

[6. FELHASZNÁLT IRODALOM 14](#_Toc58234115)

# BEVEZETÉS

Az új technológiák egyre intenzívebben jelennek meg a mindennapi életünkben, ami természetes folyamat, ahogyan az is, hogy főleg a gyerekek egyre többször keresik a szórakozási lehetőségeket az online térben. De vajon lecserélhetjük azokat a hagyományos logikai, fejlesztő és oktató játékokat, amelyeken gyerekek nemzedékei nőttek fel, ezek online változatára?

A válasz igen, helyettesíthetjük őket, de a hatások rendkívűl eltérőek lesznek. A logikai és fejlesztő játékok online nem ugyanazt a hatást váltják ki, amihez az offline verzióikban hozzászokhattak az emberek. A gyerekek nem élik át a lelkesedést, amelyet személyes játék keretében megismerhetnének, hiszen ők nem barátaikkal, osztálytársaikkal fedezik fel a játékot, hanem online, úgy, hogy közben fizikailag egyedül vannak.

Tekintettel a kialakult helyzetre, hogy a gyerekek nem járhatnak iskolába, nem mehetnek ki a játszóterekre játszani, meg kell adni a lehetőséget a gyerekeknek különböző játékok elérésére.

A dolgozat témája egy olyan Android-on futó alkalmazás, amely lehetőséget ad bármelyik korosztálynak egy ingyen letölthető Quiz játék használatára. Az applikációnak felhasználóbarát grafikus felülete van, amiben könnyen eligazolódik bármely Android felhasználó.

Az Android operációs rendszer egy ingyenesen elérhető nyílt forráskódú operációs rendszer, mely mögött a Google és számos más nagyvállalat összefogása áll. Ennek ismeretében nem meglepő, hogy napjainkra az Android operációs rendszer vált a legnépszerűbb platformmá az okostelefonok piacán. Ezt bizonyítja az is, hogy ma már több, mint 900 millió eszközön aktiválták az Android operációs rendszert, több, mint 190 országban, és naponta több, mint 1 millió új Androidos készüléket aktiválnak világszerte.

# 1.1. Célkitűzések

A projekt célja létrehozni egy ingyenesen letölthető Quiz alkalmazást, amely könnyen kezelhető, átlátható felhasználói felülettel rendelkezik, valamint az alkalmazás lehetőséget adjon minden korosztálynak a játékra, és különböző beállítottságú/érdeklődési körű felhasználó találja meg a számára ideális kérdéskategóriákat.

# KÖVETELMÉNYEK

# 2.1. Felhasználói követelmények , lásd Figure 1

* Lehessen elindítani a játékot
* Legyen lehetőség regisztrálni a felhasználónak
* Legyen lehetőség bejelentkezésre
* Lehessen kérdéskategóriák közül választani
* Legyen lehetőség az eredmények megtekintésére

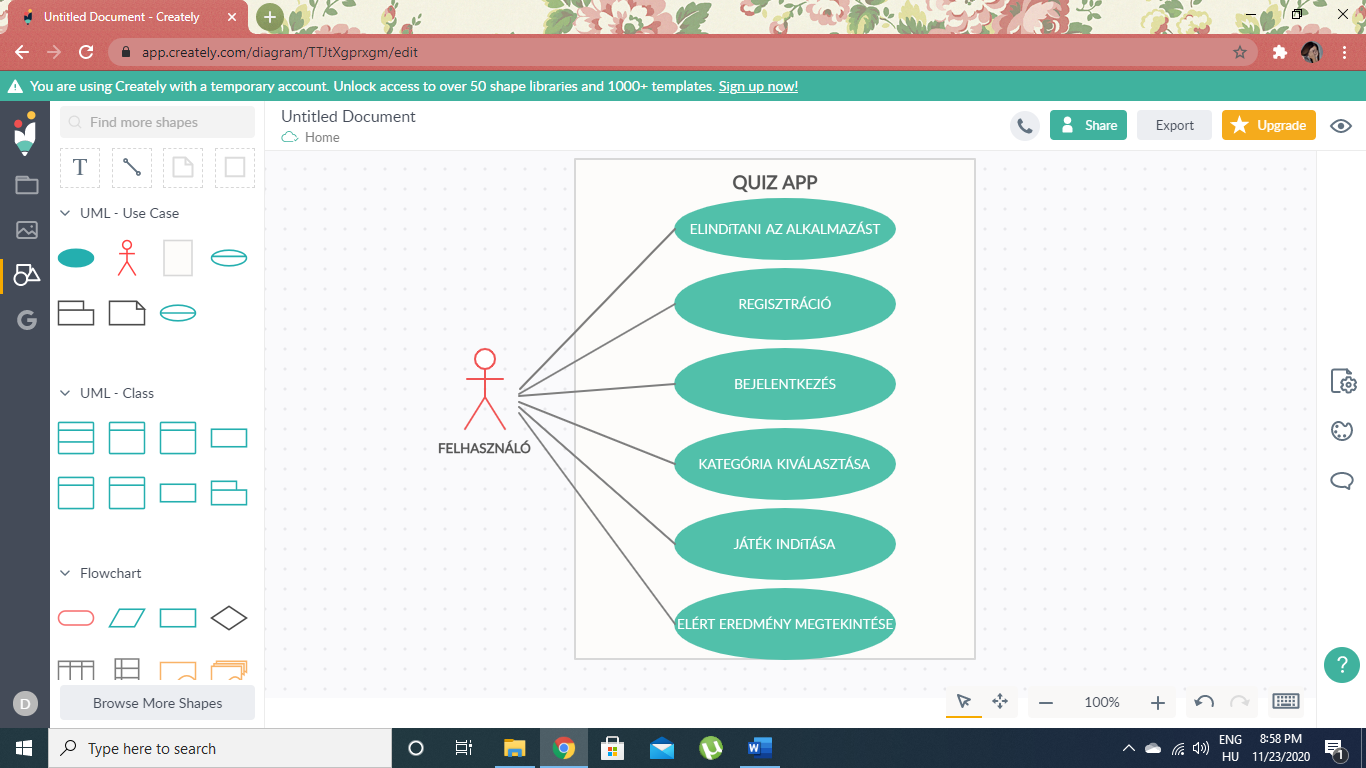
****

Figure 1 Felhasználói eset diagram

# 2.2. Nem funkcionális követelmények

* Android Studio 4.0 +
* Nexus S API R Emulátor vagy Android készülék (Android 4.4+)
* Adatbázis kezelő rendszer: Firebase RealTime Database
* Java programozási nyelv
* GitHub verziókövetés

# 2.3.Funkcionális követelmények

* **Kezdőoldal** (Home Page), lásd Figure 2: az alkalmazás ikonjára kattintva megjelenik a kezdőoldal, ahol a felhasználó számára két gomb jelenik meg, a SIGN UP (Regisztráció), valamint a SIGN IN (Bejelentkezés)

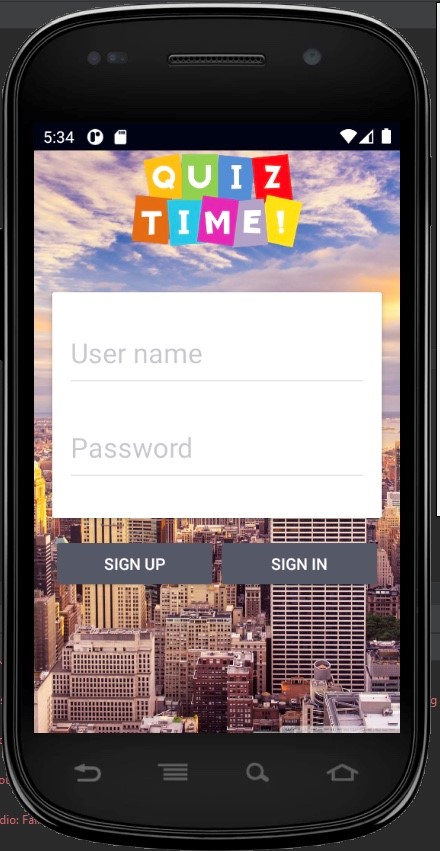


Figure 2 Kezdőoldal

* **Regisztrációs felület** (Sign Up), lásd Figure 3 : a Kezdőlap (Home Page) bal oldalán található SIGN UP gomb, erre rákattintva a felhasználó bekerül a regisztrációs felületre. Itt megjelenik egy “Please fill full information” üzenet, amely arra kéri a felhasználót, hogy töltse ki az összes mezőt.
  + “User name”-hez kell felhasználónevet megadni, amely maximum 15 karakter hosszú lehet,
  + “Password” mezőbe a felhasználó által választott jelszót kell beírni
  + Az “E-mail” mezőbe kerül a felhasználó e-mail címe

No és Yes gombokra kattinthat ezután a felhasználó. Yes-re kattintva az alkalmazás létre fogja hozni az új felhasználót.

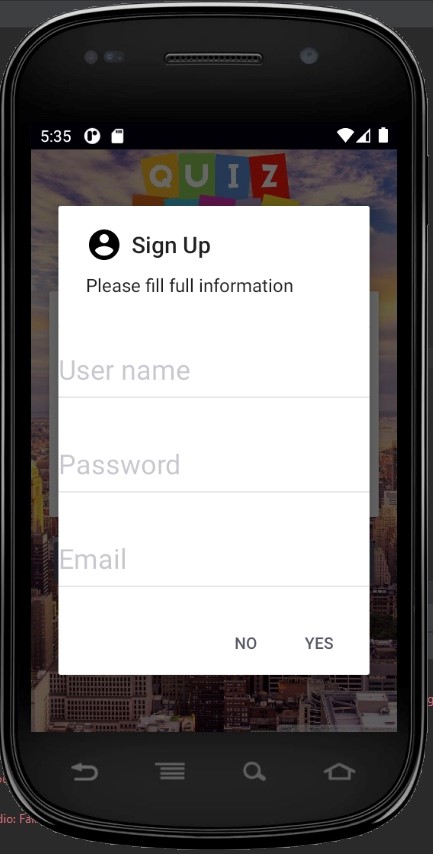


Figure Regisztrációs oldal

* **Bejelentkező felület** (Sign In): a Kezdőlap (Home Page) jobb oldalán látható SIGN IN gombra kattintva, bejutunk a bejelentkező felületre.

A bejelentkezésnél a felhasználónak be kell írnia a felhasználónevét a “User name” mezőbe, valamint a jelszavát a “Password” mezőbe, lásd Figure 4 .

* + Helytelen felhasználónév beírása esetén az alkalmazás “User is not exist” , vagyis “ilyen felhasználó nem létezik” üzenetet fog kiírni a képernyőre.
  + Helytelen jelszó beírása esetén “Wrong Password”, vagyis “helytelen jelszó” üzenetet fogunk kapni.

Csak abban az esetben tud továbblépni, ha hiteles adatokat visz be a felhasználó.

A SIGN IN gombra kattintva, a felhasználó bekerül a főmenübe

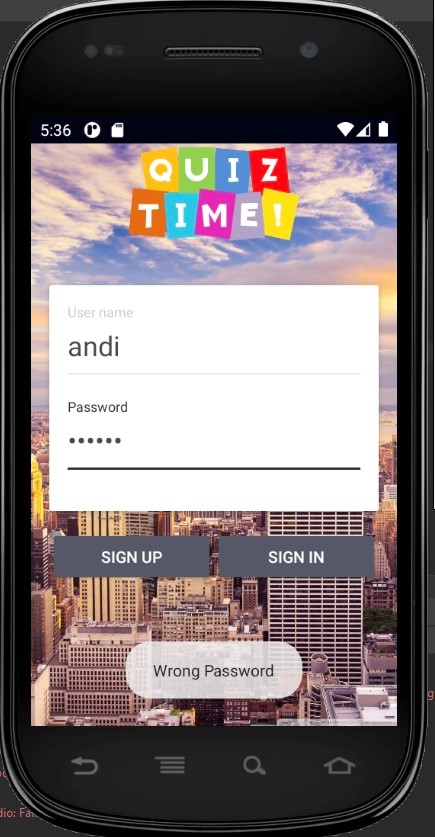
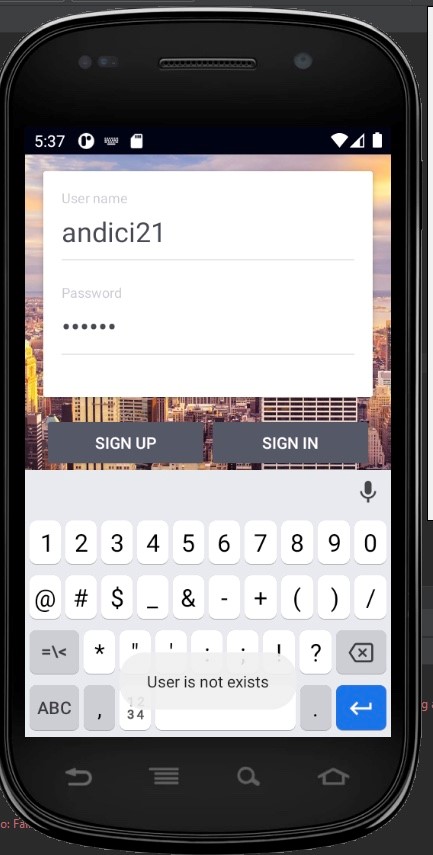
 

Figure Rossz jelszó Figure 4 Nincs ilyen felhasználó

* **Kategoriák kiválasztása**: ha sikeresen megtörtént a regisztráció, majd a felhasználó helyesen vitte be az adataid a bejentkezésnél, a SIGN IN gombra kattintva bekerül a főmenübe, ahol a kérdés kategoriák vannak kilistázva, egy-egy illusztrációval társítva.

** **

Figure Kategóriák kiválasztása

* **Játék,** lásd Figure 6 : amikor a játékosnak sikerült eldöntenie, hogy melyik kategóriában szeretné kipróbálni magát, elég a képre kattintania és megjelenik középen egy PLAY gomb, amire kattintva elkezdődnek a kérdések.

Minden kérdésnél a játékosnak 10 másodperc gondolkodási ideje van, amit a képernyő alján tud követni, amennyiben nem válaszolt azidő alatt a kérdésre, a pontot elveszítette.



Figure Játék

Két típusú kérdéssel találkozhat a játékos: képes és szöveges kérdéssel, lásd Figure 7 .

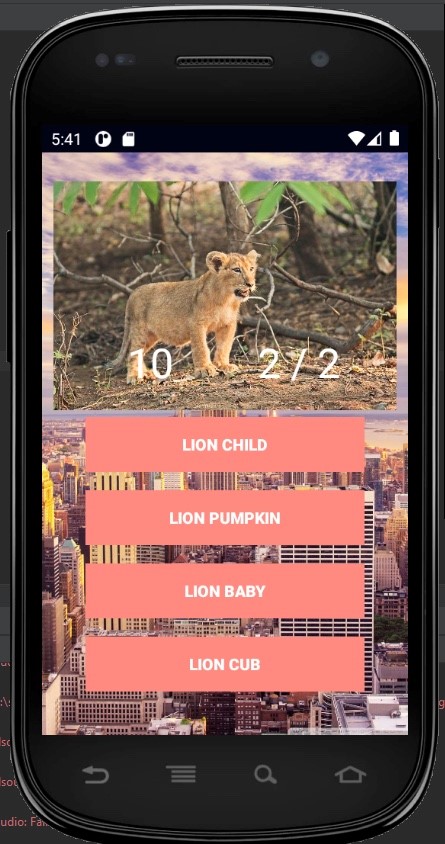
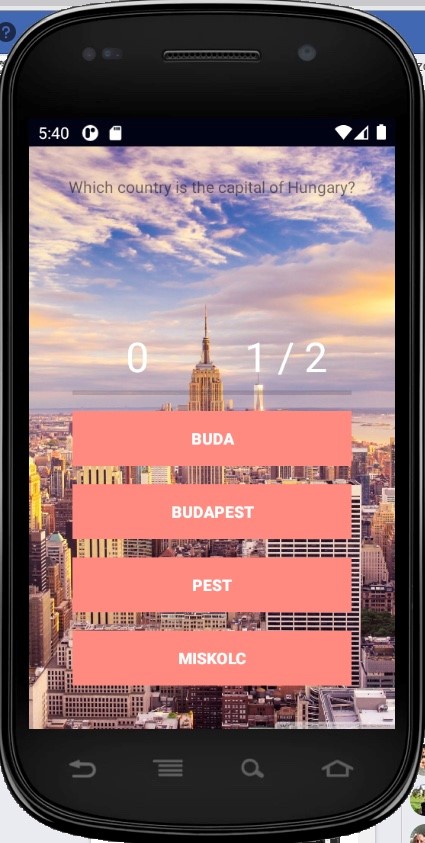
 

Figure Kérdéstípusok

Ha a játékos jól válaszolta meg az összes kérdést vagy valamelyiknél elbukott, az alkalmazás kiírja az elért pontszámot (SCORE-ként van feltüntetve) és hogy az összes addigi válasza közül hány volt helyes (PASSED).

A képernyőn még megjelenik a TRY AGAIN gomb, amire kattintva a játékost újra bedobja a főoldalra, ahol a kategóriák közül tud választani, lásd Figure 8 .

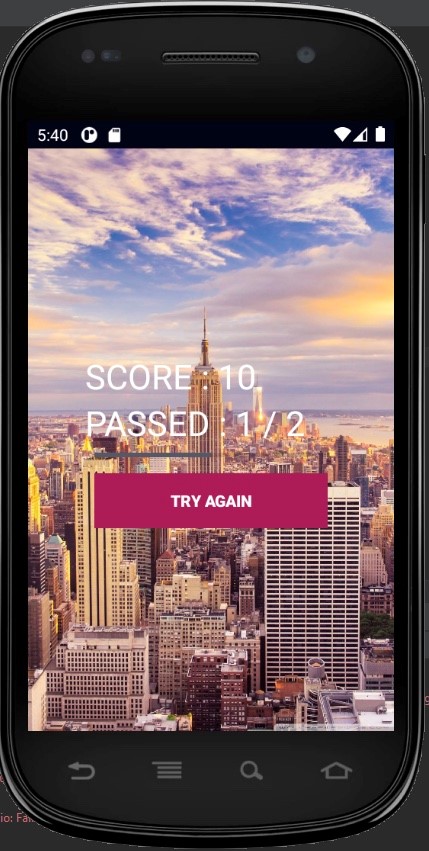


Figure Újrapróbálkozás

* **Kiértékelő felület** (Ranking Page), lásd Figure 9 : a főoldalon, bal alsó sarokban a csillaggal jelölt RANKING gombra kattintva a felhasználó megtekintheti az összes eddigi felhasználók által elért eredményeket, valamint azok kategoriánkénti teljesítményét.

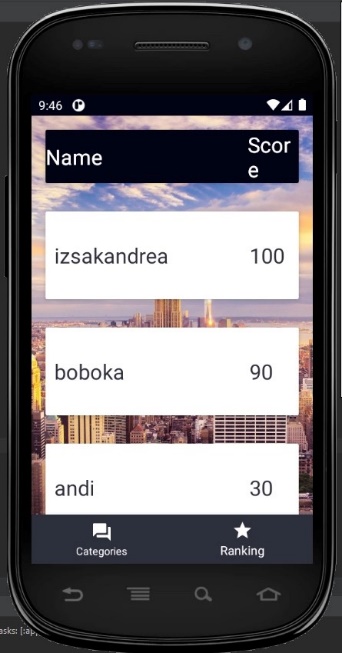
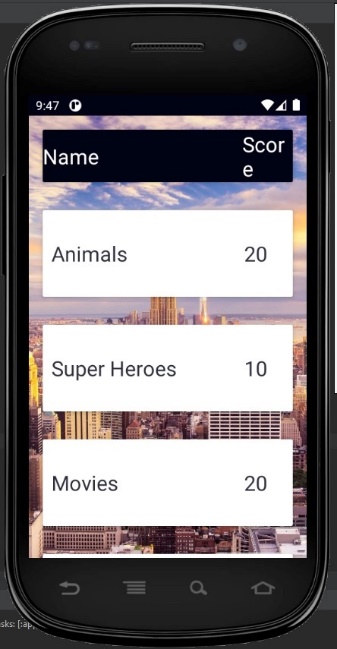
** **

Figure Kiértékelő felület

# IMPLEMENTÁCIÓ

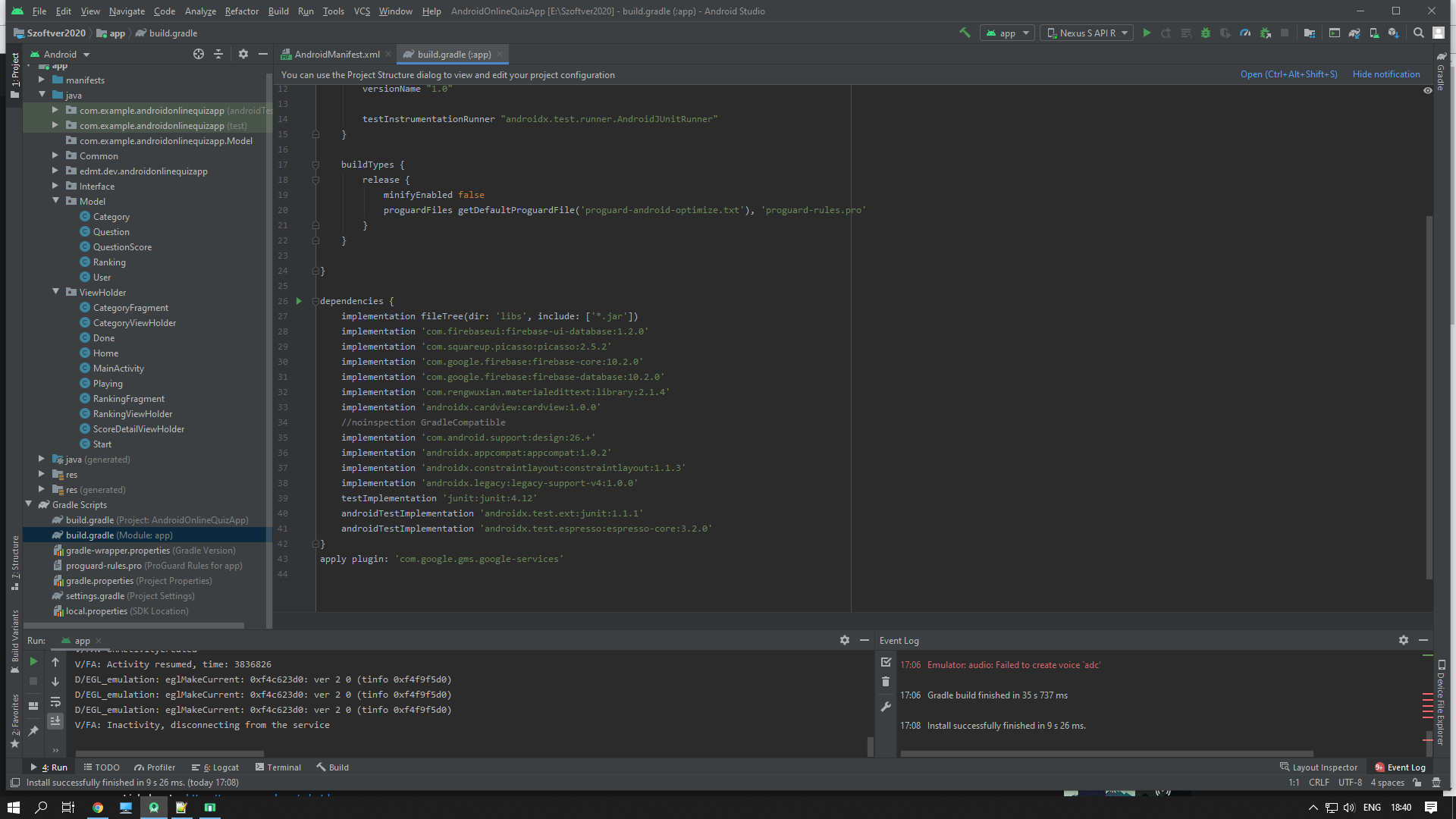
Az alkalmazást Android Studio-ban írtam, mivel egy android alkalmazásról van szó. A játékot Java nyelvben, a felületet, megjelenítést viszont .xml fájlokba.

Egy realtime adatbázisra volt szükség, erre a FireBaset használtam. Az adatbázisban tárolva vannak a felhasználók, a kérdéskategóriák és a kérdések.

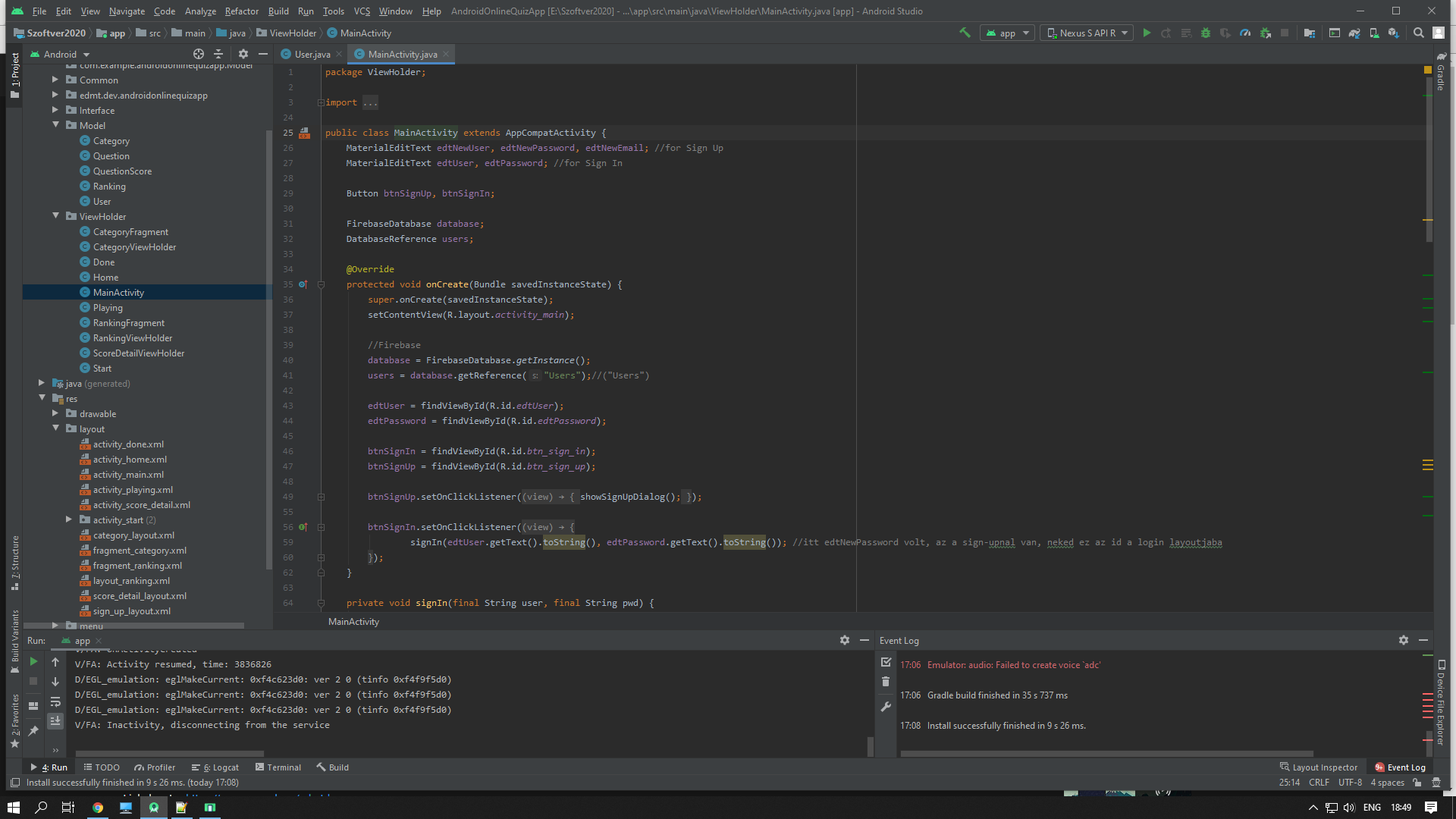
Ennek megvalosításához létre ellett hoznom egy új projektet FireBaseben és össze kellett kapcsolni az Android Studioban levő projekttel.

# 3.1. A projekt elkészítésének lépései

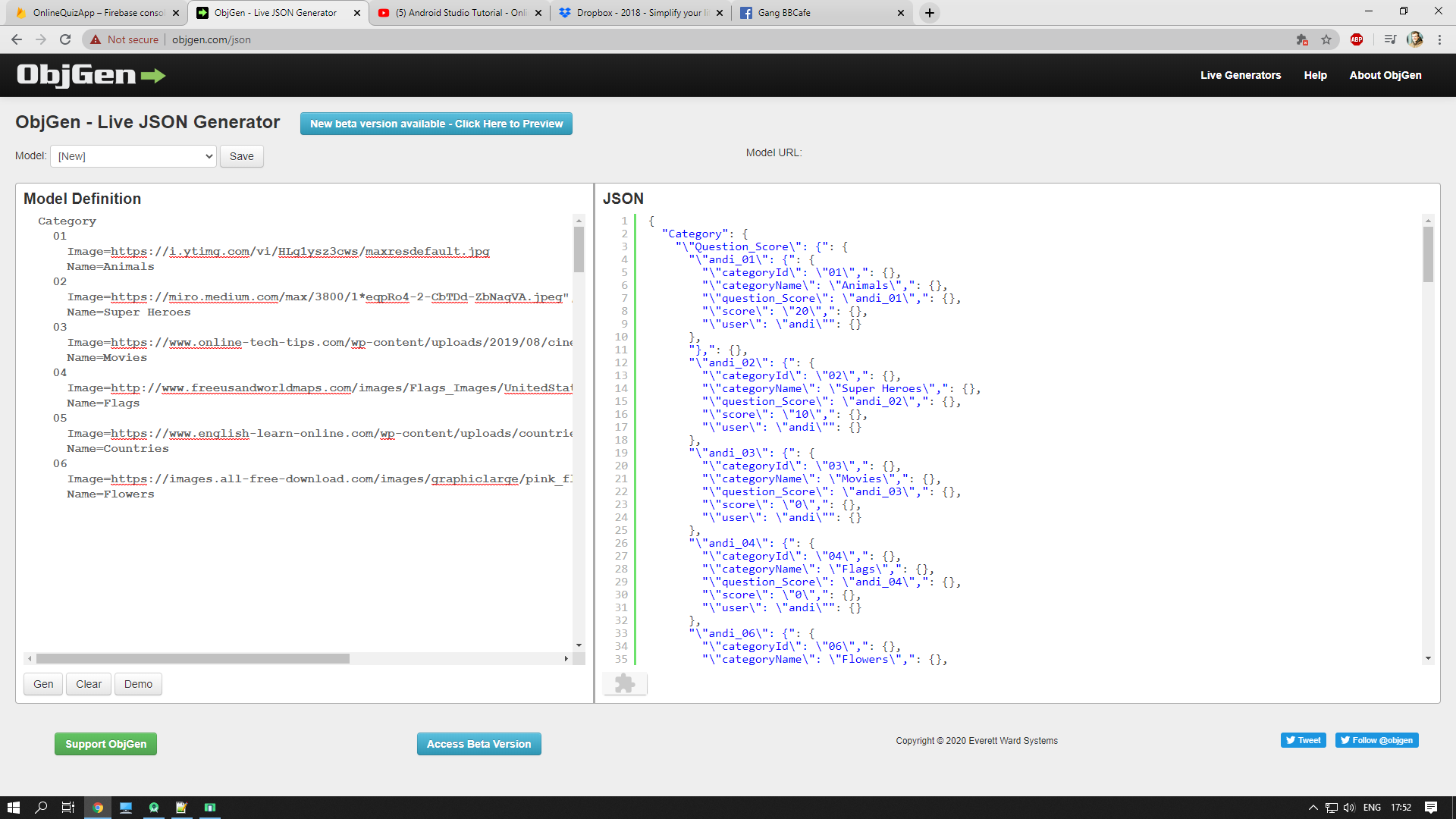
* új projekt létrehozása Android Studio-ban
* új projekt létrehozása FireBase-ben
* összekötés után az alábbi könyvtárak hozzáadása: (FireBase Core, FireBase DataBase, Material Edit Text, Card View, Support Design)



* “USER” modell létrehozása (UserName, Password, Email adattagokkal), ezt követően layout a SignUp-hoz
* Main Activity, majd az activity\_main.xml (a MainActivityben van implementlva az adatbázishoz való kapcsolás)



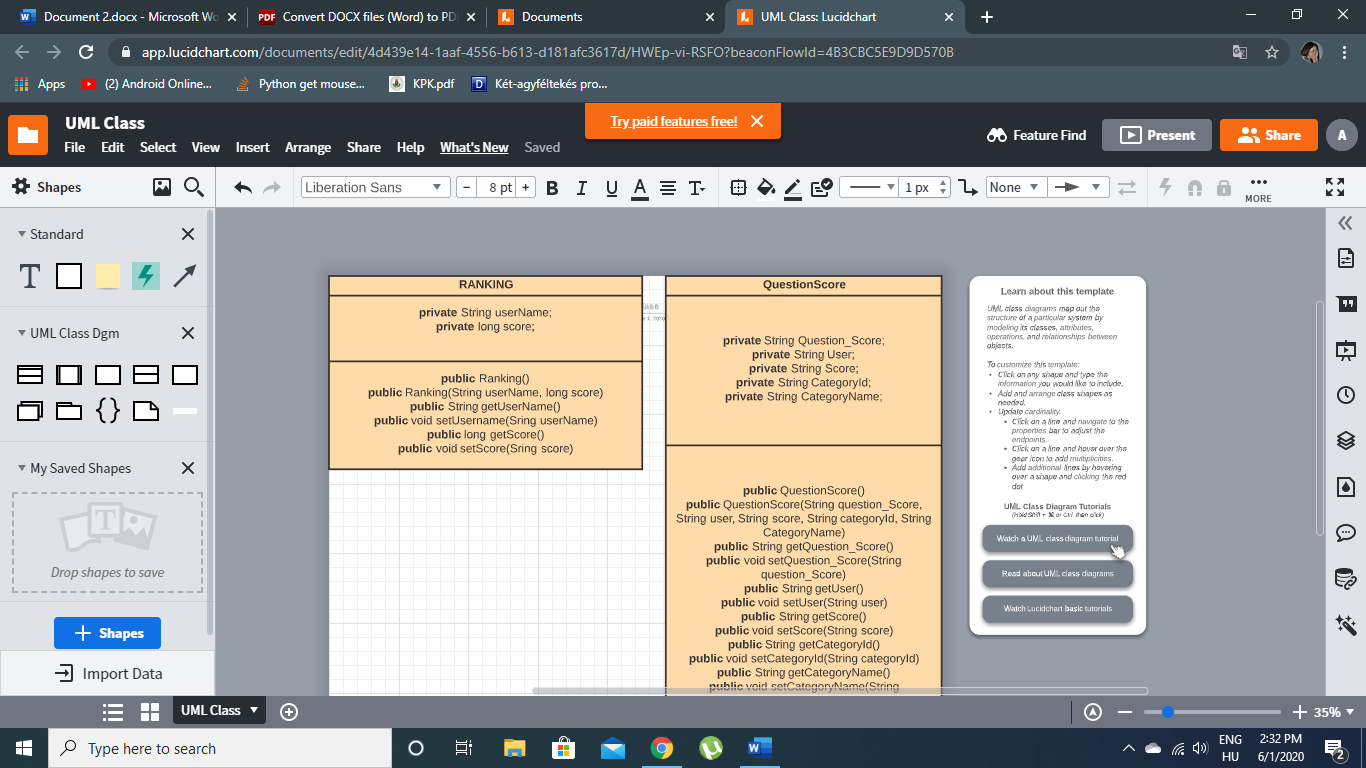
* Home Screen – A kategóriák és a pontozás betöltése (Load Category/ Load Ranking)
* “Home” Activity létrehozása, majd activity\_home.layout
* Fragment mindkettőhöz, majd mindkettőhöz megjelenítés (fragment\_category.layout, fragment\_ranking\_layout)
* A kategóriák feltöltése adatbázisba (ObjGen oldalon a Live JSON generator használatával)

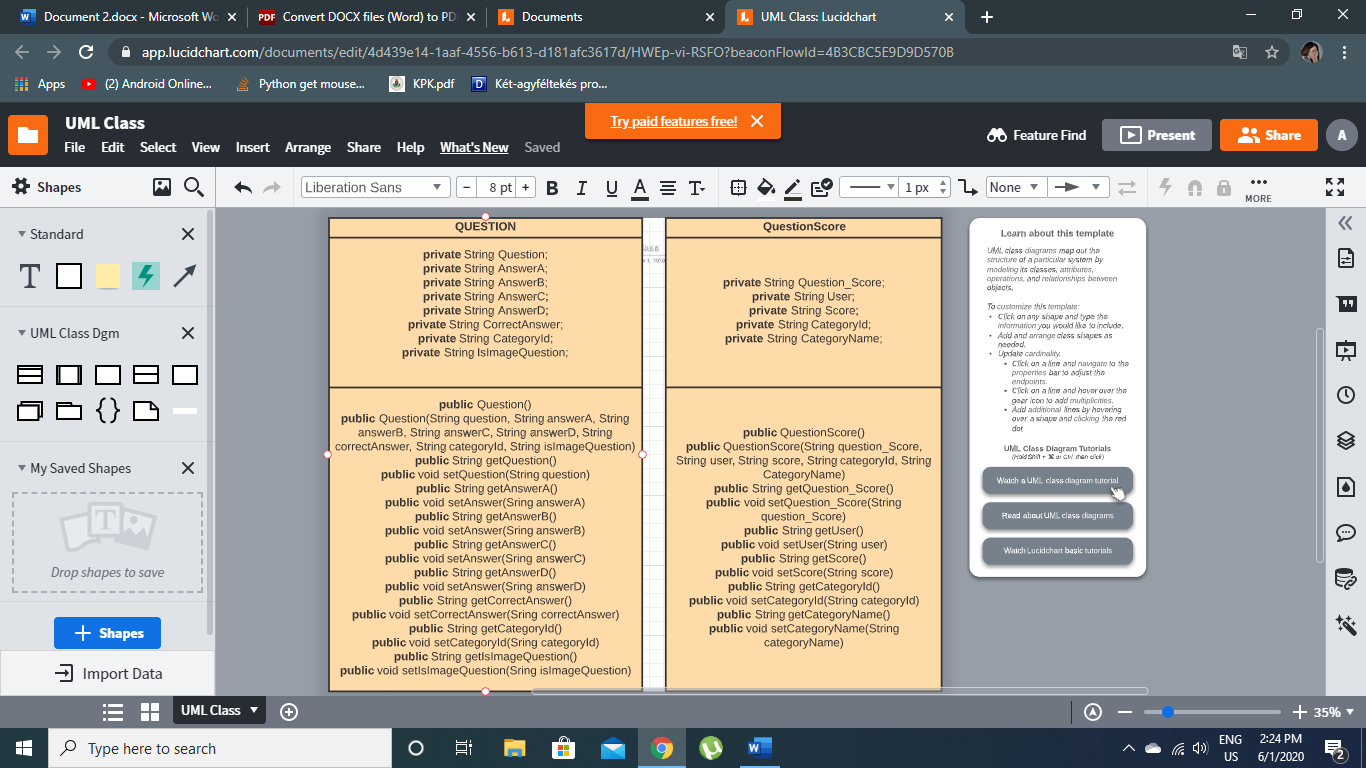


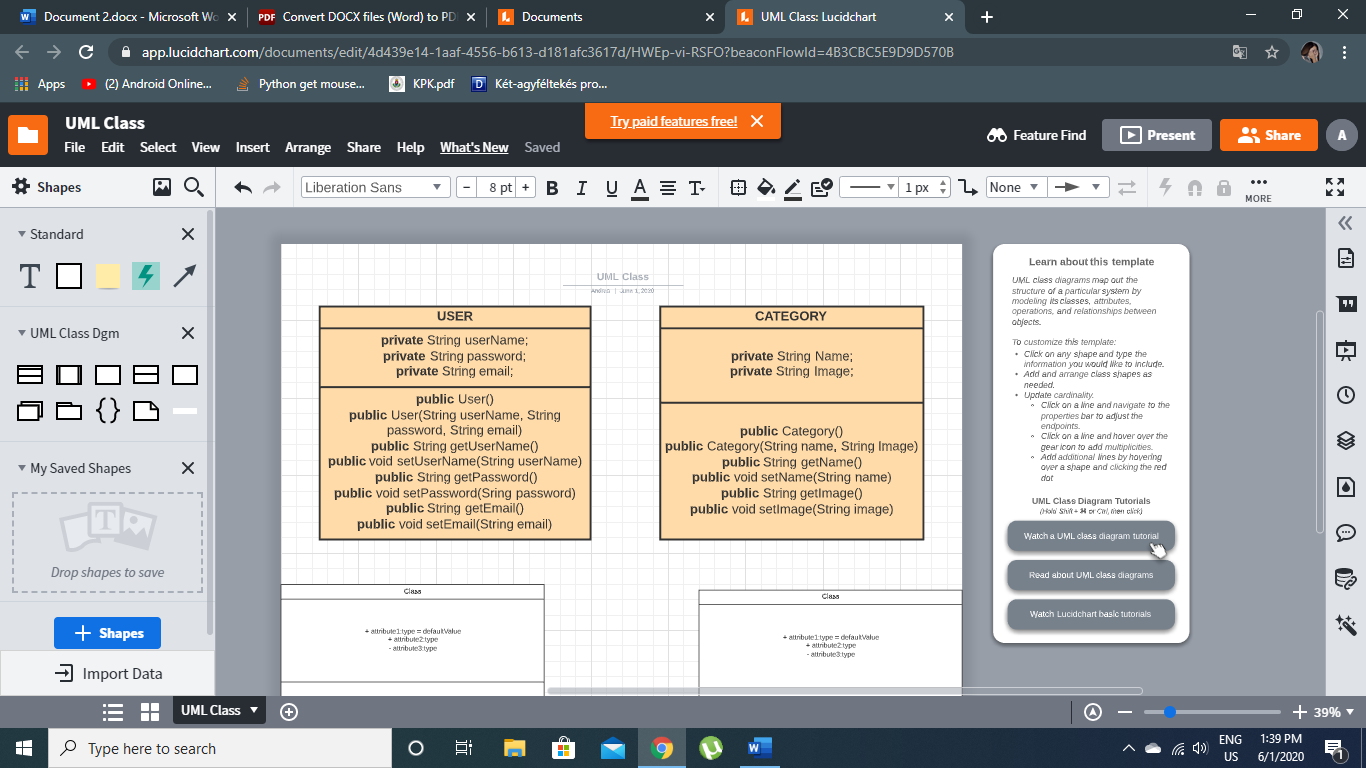
* A kapott json fájlt importálni kell az adatbázisba
* Category model létrehozása, itt elemezzük az adatokat, amiket kapunk az adabázisból, majd category\_layout
* Category View Holder – felfolgoz minden elemet
* ItemClickListener interfész létrehozása az onClick implementálásához
* Mivel a kérdéseknél nem csak szöveget, hanem képeket is fogunk betölteni az internetről, ezért importálni kell a FirebaseUI és Picasso könyvtárakat
* A játék folyamán addig jönnek a kérdések, amíg véget ér a lista FireBase-ben, vagy, ha rossz választ adunk, kilép
* Létrehozunk egy Common.java állományt a globális változók tárolásához
* Question model létrehozása a kérdésekhez
* Exportáljuk a régi FireBase adatbázis json fájlját, és hozzáadjuk az új json-t a kérdésekkel
* Activity\_start\_layout – a képerenyő a játékhoz
* Playing Activity és playing\_layout, itt alkalmazzuk az ImageView és TextView tulajdonságokat ugyanarra a pozicióra, annak függvényében, hogy a kérdés kép vagy szöveg formában van megadva és alkalmazzuk rájuk a visible/invisible tulajdonságot
* Start.java – az összes kérdést sorra kilistázza, ha jó választ adunk, tovább lép a következő kérdésre, ellenkező esetben kilép
* A kérdések betöltésére használjuk a Done activity, majd létrehozzuk az activity\_layout-ot
* Ranking model, összeszámolja az összes összegyűjtött pontot a QuestionScore-ból és feltölti a Ranking-be
* RankingFragment, ranking\_layout, RankingViewHolder
* ScoreDetailViewHolder, activity\_score\_detail\_layout, amikor az osztályozás menübe lépünk, és kíváncsiak vagyunk, hogy egy adott felhasználó melyik kategóriákból hogy teljesített, itt lehet megtekinteni

# OSZTÁLYDIAGRAMOK (UML)

Az osztálydiagramokról leolvasható, hogy az alkalmazást milyen osztályok alkotják. Minden osztályhoz tartoznak adattagok és metódusok.

****

****

****

# ÖSSZEFOGLALÓ

Több szemszögből is hasznosnak találtam ennek a játéknak a megvalosítását. Az egyik ezek közül a Java nyelv közelebbről való megismerése. A másik fontos szempont az, hogy betekintést nyerhettem abba, hogy milyennek kell lennie egy államvizsga dolgozatnak, valamint, hogy hogyan tudok különböző akadályokat, nehézségeket legyőzni önerőmből.

Összességében, a sok nehézség és elakadás ellenére, pozitívan zártam a projektet, hisz sikerült egy működő projektet megvalósítanom.

# 5.1. JÖVŐBELI TERVEK

Természetesen rengeteg ötlet, lehetőség van az alkalmazás bővítésére. Ezek közül a legfontosabbakat említeném még:

* Nehézségi szintek bevezetése (legyen kezdőknek – haladóknak és a nagyon jóknak egyaránt lehetősége játszani)
* A kérdéskategóriák kibővítése újakkal
* Úgynevezett “Challengek”, pl. időre játszani, két perc alatt ki válaszolja meg helyesen a legtöbb kérdést

# FELHASZNÁLT IRODALOM

<http://developer.android.com/studio/intro/>

<http://stackoverflow.com/questions/>

<http://youtube.com/>

<https://app.lucidchart.com/documents#/templates?folder_id=> (UML és Use- Case diagramok elkészítésére használt)