

# Záródolgozat

---

**Témavezető:** Szabó Dániel

**Készítette:** Kovács László, Szoftverfejlesztő

**Év:** 2020-2021

**Osztály:** 13.F

**Iskola:** Weiss Manfréd Szakgimnázium, Szakközépiskola és Kollégium

## Tartalom

1. **Összefoglaló**
  - 1.1 **Záródolgozat téma kiválasztása**
  - 1.2 **Alkalmazás tervezése**
  - 1.3 **Programnyelv kiválasztása, előkészületek**
  - 1.4 **Adatbázis**
  - 1.5 **Köszönetnyilvánítás**
2. **Felhasználói dokumentáció**
  - 2.1 **Rendszer Követelmény**
  - 2.2 **Alkalmazás telepítése**
  - 2.3 **Alkalmazás használata**
  - 2.4 **Gyakran ismételt kérdések**
  - 2.5 **A program készítőjének elérhetősége**
3. **Fejlesztői dokumentáció**
  - 3.1 **Adatbázis**
    - 3.1.1 **Account tábla**
    - 3.1.2 **Hozzászólás tábla**
    - 3.1.3 **Mentetttopic tábla**
    - 3.1.4 **Topic tábla**
  - 3.2 **Forrás kód**
  - 3.3 **Java – PHP kapcsolat**
  - 3.4 **PHP kódok**
  - 3.5 **Főbb változók**
  - 3.6 **Admin felület**
  - 3.7 **Fejlesztési lehetőségek**
  - 3.8 **Milyen nehézségekkel találkoztam?**
  - 3.9 **Teszt dokumentáció**
4. **Irodalomjegyzék**

## 1. Összefoglaló

---

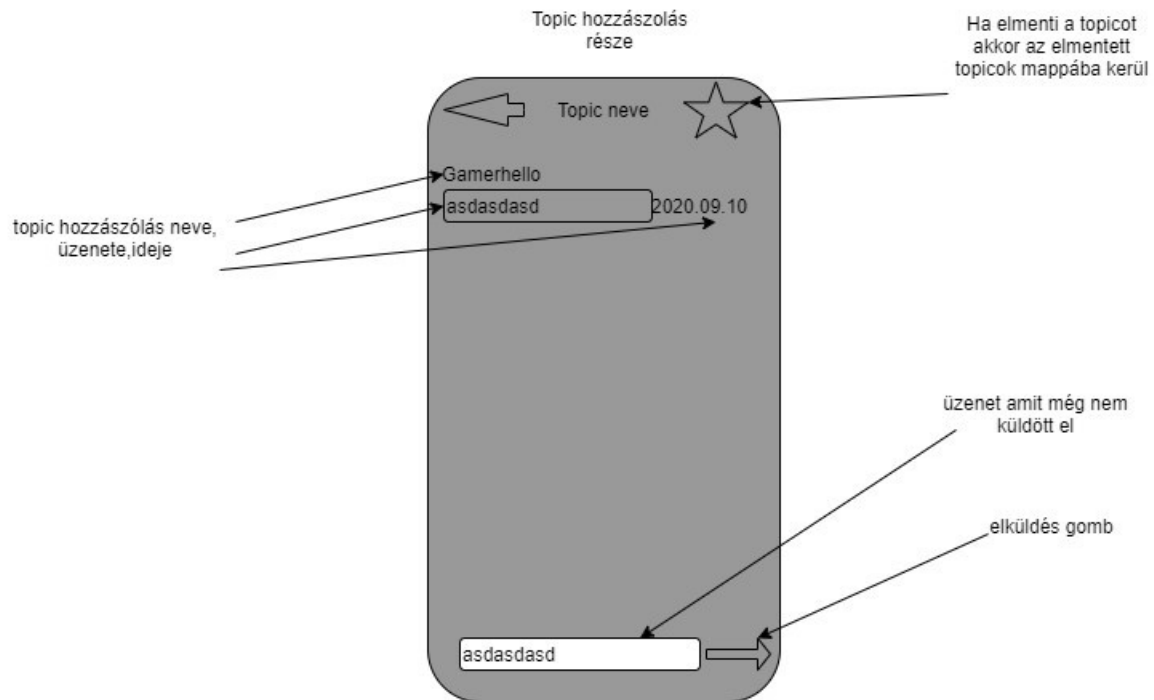
### 1.1 Záródolgozat téma kiválasztása

Először biztos voltam benne, hogy egy olyan felületet szeretnék létrehozni, ahol az általam létrehozott

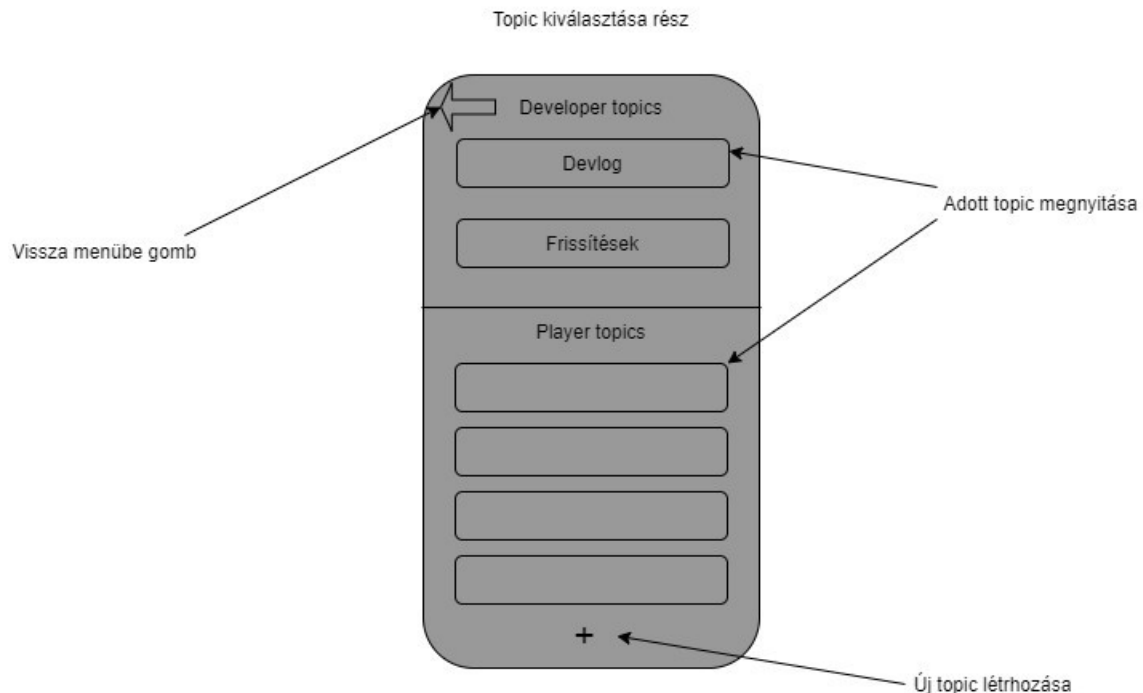
## Feladat

alkalmazásokról tudok kommunikálni a felhasználókkal. Azonban ki kellett találni, hogy milyen platformon szeretném ezt létrehozni (webes vagy rendes alkalmazás?). Elkezdtém tervezni egy weboldal alapú fórumot, ami számomra nem volt megfelelő, ezért eldöntöttem, hogy egy telefonos alkalmazást fogok létrehozni.

## 1.2 Alkalmazás tervezése



Az alkalmazás kiválasztása után kezdetleges látványterveket hoztam létre, amelyeket (egy-két kivétellel) meg is valósítottam.



Elsőnek úgy terveztem, hogy belépés után a felhasználót egy kezdő felület köszöntse, amely két opciót tartalmaz: Belépés vagy Regisztráció. Ezt végül elvetettem, ugyanis számomra jelentéktelenné vált.

## Feladat

A következő tervezési lépés a topic kiválasztó rész volt.

Végül a topichoz való hozzászólást terveztem meg.

### 1.3 Programnyelv kiválasztása, előkészületek

Miután eldöntöttem, hogy alkalmazást szeretnék létrehozni, szembesültem azzal a problémával, hogy rengeteg platform van egy alkalmazás létrehozására. Időbe telt, mire átnéztem párat ezek közül, de végül az **Android Studio** nyerte el a tetszésemet, amelyben Java-t használtam. Miután kiválasztottam az alapokat, megkezdődtek az előkészületek, hiszen eddig nem programoztam Javában. Az első hetekben egy egyszerű számológépet készítettem, amelyen nagyon sok mindent kitapasztaltam, és később ez mind a hasznomra vált.

### 1.4 Adatbázis

Az adatbázis létrehozásához és szerkesztéséhez az XAMPP programot használtam, ami teljesen megfelelt mindenre, amire szükségem lehetett. Természetesen az adatbázissal kapcsolatban is rendelkeztem egy tervvel.

### 1.5 Köszönetnyilvánítás

Annak ellenére, hogy ebben az évben igencsak keveset tartózkodtunk az iskolában, rengeteg segítséget és támogatást kaptam Kovács László tanár úrtól, gyakorlati és elméleti szinten is.

---

## 2. Felhasználói dokumentáció

---

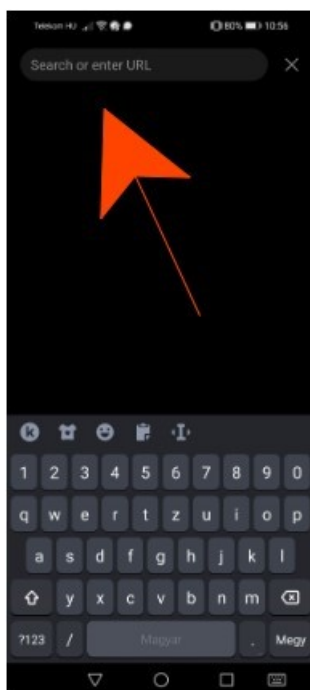
### 2.1 Rendszer Követelmény

- Android rendszer
- Internet kapcsolat
- Minimum 5 MB szabad tárhely
- Minimum 3.00 -inch kijelző

### 2.2 Alkalmazás telepítése



Miután kiválasztottuk, hogy melyik eszközre szeretnénk telepíteni, utána indítsunk el bármilyen böngészőt rajta. Huawei készülékeken egy kék bolygó ikon jelzi ezt.



Megnyitás után a képernyő tetején található keresőbe írjuk be ezt a címet: `szdaniel.hu`.

Amennyiben jól írtuk be a címet, abban az esetben ez az oldal lesz látható számunkra. Itt a 'Letöltés' feliratú gombra kattintva elindíthatjuk a letöltést.

Böngészőtől függően megkérdezi, hogy biztosan le szeretnénk-e tölteni ezt a fájlt. A 'download' gombra kattintva ezt engedélyezzük és le is tölti nekünk.

## Feladat

Miután letöltöttük, a rendszer megkérdezi, hogy biztosan szeretnénk telepíteni? Az 'allow' (vagy magyar nyelvű készülékek esetében 'engedélyezés') feliratú gombra kattintva engedélyezzük a telepítést.

Engedélyezés után nincs más dolgunk, mint telepíteni az alkalmazást. Kattintsunk a jobb alsó sarokban található 'Install' (magyar rendszer esetén 'telepítés') gombra, és a rendszer utána telepíti nekünk a programot.

Amennyiben mindent jól csináltunk, és a telepítés sikeres, abban az esetben egy zöld pipával jelzi a program, hogy minden rendben van. Ez után kiléphetünk a telepítési felületről.

## 2.3 Alkalmazás használata

Az alkalmazás megnyitását követően a belépési felület fogad minket. Első belépésnél ez számunkra nem lesz megfelelő, hiszen nincs regisztrált fiókunk a fórumra. A 'bejelentkezés' gomb alatt található 'regisztráció' felíratra kattintva regisztrálhatunk, annak érdekében, hogy később be tudjunk jelentkezni a fiókunkba.

Amennyiben rendelkezünk regisztrált fiókkal, abban az esetben a regisztrációkor megadott emailt és jelszót beírva, majd a 'bejelentkezés' gombot megnyomva be tudunk jelentkezni.

Ha nem adjuk meg valamelyik információt, vagy hibásan adjuk meg, akkor a program ezt jelzi nekünk.

Regisztrációkor 3 adatot kell megadnunk:

- Egy felhasználónevet, ezt a 'Nick név' felirat alá írjuk (alatta lévő vízszintes vonalra kattintva írhatunk oda).
- Egy email címet, amit az 'email cím' felirat alá írhatunk.
- Egy jelszót, amit a 'Jelszó' felirat alá írunk.

Mind a három adat szükséges a regisztrációhoz! Amennyiben valamelyiket nem adjuk meg, abban az esetben a program jelzi nekünk.

Adatok kitöltése után a 'Regisztráció' gombra kattintva regisztrálhatunk is. Amennyiben mégis van regisztrált fiókunk, abban az esetben a 'Belépés' felíratra kattintva visszatérhetünk a belépés menühöz.

Belépés vagy regisztráció után, a menüben találjuk magunkat. Itt három opció közül választhatunk. Kiválasztás után kattintsunk a kiválasztott opcióra.

- Kijelentkezés
- Topicok
- Profil

A Kijelentkezés gomb, ahogy a neve is mutatja, kijelentkezteti a felhasználót. A Topic gomb kiválasztása esetén más felhasználók által létrehozott topikokat olvashatunk, illetve akár létre is hozhatunk egy saját topikot. A Profil fül kiválasztása esetén a saját profilunkat tekinthetjük meg.

A topic fület kiválasztva ez a kép tárul elénk. Ha úgy döntünk, hogy inkább visszamennénk, abban az esetben a 'vissza' gomb segítségével megtehetjük ezt.

Ha új topikot szeretnénk létrehozni, akkor azt a '+' gomb megnyomásával elkezdhetjük.

A 'devlog' gomb Dániel által létrehozott játékok státuszához visz. A frissítések gomb az alkalmazásokhoz és játékokhoz készült frissítéseket írja le részletesen. Amennyiben a képernyő közepétől kezdődő listából látunk olyan rövid leírást, ami felkelti a figyelmünket, abban az esetben kattintsunk rá, és a program annak a topiknak a

hozzászólásaihoz visz minket.

### Új topic létrehozása

Új topic létrehozásánál ez az ablak tárul elénk. Itt is van lehetőség visszalépni a topic kiválasztó részhez. Amennyiben létre szeretnénk hozni egy topicot, abban az esetben két dologra lesz szükségünk:

- Egy topic névre.
- A probléma hosszas leírására.

Egy rövid, ámde sokat eláruló névre lesz szükség, amely röviden összefoglalja a problémát. Érdemes kulcsszavakat beírni. A probléma hosszas leírásánál ajánlott minél több dolgot leírni a problémáról (pl. mikor történt, mi történt pontosan), hogy utána sokkal könnyebben tudjanak segíteni a felhasználók. Amennyiben valamelyik részt nem töltjük ki, abban az esetben a program hibát jelez.

### Topicok kezelése

Itt is megtalálható az eddig sokszor használt vissza gomb, amellyel visszaléphetünk a topic kiválasztó menühöz. A képernyő fenti részénél, középen található a topic neve. Emellett pedig a topic elmentésére szolgáló gomb. Ha erre rákattintunk, akkor a profil menüben a 'mentett topicok' listában megtalálható lesz ez a topic.

Alattuk egy lista lesz található, ahol az eddigi hozzászólások vannak. Ezek közül kiválaszthatunk egyet, amelyet követően a program részletesen kiírja az információkat a hozzászólásról. A képernyő alján található a hozzászólás írására szolgáló vízszintes vonal, amire kattintva beleírhatunk. Ha írtunk egy hasznosnak vélt hozzászólást, akkor a mellette lévő 'Send' gombbal elküldhetjük azt, hogy a többi felhasználó is olvashassa. Amennyiben nem töltjük ki a hozzászólás részt, abban az esetben a program nem küld el semmit.

Amennyiben kiválasztottunk egy hozzászólást, abban az esetben ez a kép tárul elénk. Itt megtalálható a hozzászóló neve, illetve maga a hozzászólás. Amennyiben vissza szeretnénk menni a többi hozzászóláshoz, akkor a vissza gomb segítségével megtehetjük ezt.

### Profil megtekintése

A menüben található profil opció kiválasztása esetén ide kerülünk. Itt is megtalálható a vissza gomb természetesen. Ezen felül megtalálható a felhasználó által regisztrációnál megadott felhasználónév és email cím. Alul megtalálható két opció:

- Hozzászolt topics: elvisz az általunk hozzászolt topicokhoz.
- Mentett topics: elvisz az általunk elmentett topicok listájához.

Amennyiben a hozzászolt topic opciót vagy a mentett topic opciót választottuk, ez a kép tárul elénk. Itt a program egy listába összeszedi az összes felhasználó által mentett vagy hozzászolt topicot. Ha a felhasználó meg akar tekinteni egy topicot innen, akkor egyszerűen csak ki kell választania. Egy kattintással átviszi a felhasználót ahhoz a topichoz.

Természetesen itt sem ragad meg a felhasználó, hiszen az eddig megszokott vissza gomb itt is megtalálható.

## 2.4 Gyakran ismételt kérdések

- **Elérhető lesz különféle webáruházakban?**  
*Igen, amint a szükséges dokumentumokat kitöltöttem.*

- **Mennyi különféle felhasználót hozhatok létre?**  
*Bármennyit. Nincs megkötve, viszont nem árt figyelembe venni, hogy 6 hónap inaktivitás után töröljük a nem használt felhasználókat!*
- **Abban az esetben, ha problémám van a programmal, hol jelezhetem ezt?**  
*Amennyiben a fórumon ezt nem teheti meg, abban az esetben a 'elérhetőség' részénél feltüntetett email címen lehet jelezni.*

## 2.5 A program készítőjének elérhetősége

- E-mail cím: danikaszab@gmail.com
- Telefon szám: 06-30-812-8489

## 3. Fejlesztői dokumentáció

### 3.1 Adatbázis

#### 3.1.1 Account tábla

Ebben a táblában található a felhasználó összes információja, amire szükség lehet.

oszlop neve	Típusa
aid®	int(10)
anév	varchar(20)
ajelszo	varchar(20)
aemail	varchar(30)
jogosultsag	int(1)

Itt az `aid` kapta az elsődleges kulcsot, hiszen ez alapján lehet a legkönnyebben beazonosítani egy felhasználót. Ez az érték automatikusan növekszik, így külső beavatkozás nem szükséges (új felhasználó esetén növekszik eggyel). Az `anév` oszlopban a felhasználó felhasználónevét adhatja meg, amely maximum 20 betűt vagy számot tartalmazhat. Ez lesz az a név, amit az alkalmazásban hozzászóláskor és topic létrehozáskor ki fog írni az alkalmazás. Az `ajelszo` hasonló az `anév` oszlophoz, viszont ezt az értéket sehol nem mutatjuk meg, mivel a felhasználó ezzel, illetve az `aemail`-el tud belépni. Az `aemail` oszlop szolgál a felhasználó email címének elmentésére. Ezt a négy értéket a felhasználó adja meg a regisztrációkor. A `jogosultsag` oszlop tartalmazza a felhasználó jogosultságát. Itt három érték szerepelhet:

- 0: admin felhasználó, ez azt jelenti, hogy későbbiekben látni fogja a csak adminoknak szóló gombot.
- 1: sima felhasználó, ilyenkor csak az alap minden felhasználónak megengedett funkciókat használhatja.
- 2: bannolt fiók, ez az az eset, mikor a felhasználó megsértette valamelyik szabályt. Ebben az esetben ezt a fiókot nem lehet tovább használni, ugyanis belépéskor nem engedi tovább a felhasználót.

Összességében ezt tartom a legfontosabb táblának, hiszen e-nélkül belépni sem lehet az alkalmazásba.

### 3.1.2 Hozzászólás tábla

Ebben a táblában a hozzászólásokról található információ.

oszlop neve	Típusa
hid <sup>®</sup>	int(10)
aid	int(10)
tid	int(10)
htext	varchar(6000)

Itt a `hid` az elsődleges kulcs. Ebben az oszlopban tároljuk a hozzászólás azonosítóját. Ez az érték automatikusan növekszik, így külső beavatkozás nem szükséges (új hozzászólás esetén növekszik eggyel). Az `aid` oszlopban a hozzászólást létrehozó felhasználó azonosítóját tároljuk. Ez azért fontos, mert ha valaki megvizsgálja a hozzászólást, akkor a hozzászóló nevét kiírja, emellett amennyiben a hozzászólás megsértett egy szabályt, abban az esetben bannolhatjuk a felhasználót. A `tid` oszlopban a hozzászólás topic azonosítóját tároljuk, így tudjuk összekötni a hozzászólást a topic-al. A `htext` oszlopban magát a hozzászólást tároljuk.

### 3.1.3 Mentetttopic tábla

Ebben a táblában tárolom el a felhasználó által mentett topicokat.

oszlop neve	Típusa
tid	int(10)
aid	int(10)
valid	tinyint(1)

Itt a `tid` az elsődleges kulcs. Ebben az oszlopban tároljuk a topic azonosítóját. Ez az érték automatikusan növekszik, így külső beavatkozás nem szükséges (új mentett topic esetén növekszik eggyel). Az `aid` oszlopban a mentő felhasználó azonosítóját tároljuk. Ezzel tudjuk összekapcsolni a mentett topicot a felhasználóval. A `valid` oszlop tárolja, hogy a felhasználó visszavonta-e a mentést. Erre azért van szükség, mert ha a felhasználó meggondolja magát, és nem akarja mentve hagyni, akkor egyszerűen csak 1-re állítom (ez mutatja, hogy nem akarja látni), viszont ha utána mégis menteni akarja, akkor nem hozom létre újra, hanem ezt az értéket 0-ra állítom.

- 0: látni akarja
- 1: nem akarja látni

### 3.1.4 Topic tábla

Ebben a táblában találhatóak a topichoz fűződő információk.

oszlop neve	Típusa
tid	int(3)
anev	varchar(20)



oszlop neve	Típusa
tvalid	tinyint(1)
ttartalom	text
tnev	varchar(40)

Itt a `tid` az elsődleges kulcs. Ez a topic azonosítója, hogy később könnyebben lehessen megtalálni. Ez az érték automatikusan növekszik, így külső beavatkozás nem szükséges (új topic esetén növekszik eggyel). Az `anev` oszlopban tároljuk a létrehozó nevét, hogy könnyebben tudjuk kiírni. A `tvalid` jelzi, hogy az adott topic látható-e. Amennyiben túl sok negatív hozzászólás van, vagy akár maga a topic értelmetlen, abban az esetben láthatatlanná lehet tenni. Ennek két értéke lehet:

- 0: Látható minden felhasználónak
- 1: Nem látható senkinek.

A `ttartalom` a létrehozó által megadott tartalom helye (amit létrehozásnál megad, pl. kérdések és információk). A `tnev` pedig a létrehozó által megadott topic név. Ezt fogja mindenki először meglátni.

## 3.2 Forrás kód

Mivel magát az alkalmazást az `Android Studio` programban csináltam, így a scripteket külön tárolja el, ezért fontosnak gondoltam, hogy külön is mellékeljem a kódsoraimat. Ezt a `forráskódok` mappában lehet megtalálni.

Illetve az adatbázis kezelését és minden más internetet igénylő feladatot PHP-ban oldottam meg. Ezt a `php kódok` mappában lehet megtalálni. A `forráskódok` mappában található egy `gyakranhasznalt` Java class. Ez 2 nagyon fontos változót tartalmaz.

Mivel külön fájlokban van a PHP-kód, illetve ezek egy webszerveren találhatóak, ezért elérési utat kell megadni. Annak érdekében, hogy ne folyton ugyanazt kelljen megadni mindenhol, létrehoztam ezt a két változót, hogy csak beilleszteni kelljen. Ha itt átirom, akkor nem lesz olyan gond, hogy valahol máshol nem írtam át, és kifagy az alkalmazás.

## 3.3 Java – PHP kapcsolat

Ahhoz, hogy sikerüljön adatot átrakni PHP-ba, 2 db tömbre van szükség. Az első tömbnek a változó nevet kell tartalmaznia, pl. `email` vagy `felhasznalonev`, a másodiknak pedig magát a változót kell tartalmaznia, pl. `galuska@galuska.com` vagy `gali`. Ez után a `putData` paranccsal el tudjuk indítani a PHP-t.

Ezek után amennyiben jó elérési utat adtunk meg, és a tömbök is rendben vannak, akkor a PHP le is fut. Meg kell néznünk, hogy a visszatérő adatok megfelelőek-e.

Ebben az esetben a felhasználó jogosultságát néztem meg. Először megnéztem, hogy sikerült-e felrakni, utána pedig azt, hogy sikerült-e hiba nélkül befejezni a PHP-t. Amennyiben ezek sikeresen teljesülnek, abban az esetben a `putData.getResult()` segítségével megszerezhetjük a PHP-ban `echo`-val kiírt eredményeket.

PHP oldalon az áthozott tömb változókat `POST` methoddal tudjuk használni.

## 3.4 PHP kódok

□

Mivel PHP programozási nyelvvel már volt dolgom, így megpróbáltam a legátláthatóbban megoldani a feladatokat. Minden PHP kódot három részre bontottam: `Áthozott infó`, `Adatbázis info` és `Lekérdezés`. Erre azért volt szükség, hogy sokkal jobban lehessen tudni, melyik változót hol hozom létre. Mikor teszteltem a PHP-kódokat, akkor egy új csoportot hoztam létre `Próba ként`.

Az `Áthozott infó` csoportban azokat a változókat tárolom, amelyeket a Java programból hoztam át `POST` methoddal. Ezek általában olyan információk, amelyeket később lekérdezésnél használok, pl. email, felhasználónév, id.

Az `Adatbázis info` csoportba tartozik minden, ami szükséges ahhoz, hogy elérjem az adatbázist: webszerverhez való csatlakozási név, jelszó, illetve a tábla neve. Ezt utána fel is használom, hogy csatlakozzak az adatbázishoz. Mivel webszerveren van az adatbázis és maga a PHP fájl is, így localhost névvel is pontosan tudja, hogy hova csatlakozzon.

A `Lekérdezés` csoportban minden olyan van, ami maga a program. Itt dolgozik a PHP, mivel itt kommunikálok az adatbázissal (pl. feltöltés, lekérdezés), illetve itt írom ki az adatokat, hogy utána a Java programban lekérdezzem.

## 3.5 Főbb változók

□

Nagyon fontos kiemelnem a `profil` class-t, hiszen ebben nagyon sok fontos változót tároltam el annak érdekében, hogy ne kelljen újra és újra lekérdeznem őket a szervertől.

- A `mail` változó a felhasználó email címét tárolja el.
- A `nev` a felhasználónév.
- A `jogosultsag` és az `id` a nevükből adódóan elmondják, mit tartalmaznak.

Felmerülhet a kérdés, hogy miért volt szükség elmenteni a felhasználó jogosultságát, ha már az elején megnézzük, hogy bannolva van-e vagy sem? Ez egy teljesen jogos kérdés lenne, ha nem lenne admin felület az alkalmazásban. Az alkalmazás tartalmaz egy admin felület részt, amit csak az adminok érhetnek el, ezt viszont csak akkor tudom megvizsgálni, ha elmentem a felhasználó jogosultságát.

Ebben a kódsorban vizsgálom meg, hogy admin-e a felhasználó. Amennyiben admin, abban az esetben megjelenik az admin felülethez vezető gomb.

## 3.6 Admin felület

Ez az adminoknak ad egy felületet a topicok és az emberek viselkedésének szabályozására. Amennyiben egy admin lép be az alkalmazásba, akkor így néz ki a menü:

Természetesen nem nagy varázslat egy gombot láthatatlanná tenni egy átlag felhasználó szeme előtt. Azonban a fontossága annál inkább érdekesebb.

Azért tartottam fontosnak, hogy az admin felülethez vezető gomb egy helyen legyen a topic kiválasztó résszel,

## Feladat

mert így egy admin is élvezheti azokat a funkciókat, amiket egy átlag felhasználó, és így nem kell átjelentkeznie egy normál fiókba.

Az admin felületre kattintva ezt fogjuk látni. Itt a nem admin jogosultsággal rendelkező felhasználók neve található. Amennyiben rákattintunk egy névre, annak az embernek az információit találjuk meg.

Ha rákattintunk egy névre, ez a kép tárul elénk:

Az eddigi információk mellett alul kiírja a felhasználó által írt hozzászólásokat a könnyebb moderálás érdekében. Amennyiben nincs bannolva a felhasználó, abban az esetben **Ban** gomb jelenik meg, ellenkező esetben pedig az **Unban** feliratú gomb jelenik meg. Bármelyik feliratú gomb jelenik meg, rákattintva bannolhatjuk vagy unbannolhatjuk a felhasználót.

## 3.7 Fejlesztési lehetőségek

- Felhasználói adatok megváltoztatása.
- Felhasználó törlése.
- Automatikus bejelentkezés.
- Hozzászólás like-dislike lehetőség.
- Egy adott személy ignorálása: Egy felhasználó kiválaszt egy másikat, akinek a hozzászólásait nem szeretné látni, és a továbbiakban azok a hozzászólások nem jelennek meg.
- Hozzászólás, illetve egy adott topic keresési lehetőségei.
- Admin felkeresése opció: Lehetővé teszi, hogy egy adminnak írjunk egy adott topic-kal kapcsolatban (Report funkció).

## 3.8 Milyen nehézségekkel találkoztam?

Igazából rengeteggel. Elsősorban iskola mellett nehéz volt belekezdeni egy új programozási nyelvbe, ez a kódsoraimon is látszik, hiszen sok helyen ott a megjegyzés, hogy mi mit csinál. A belépési menü ezért lett ilyen kesze-kusza.

Miután kiismerem magam a lehetőségeimmel, utána jött egy újabb akadály: nem tudtam adatbázishoz kapcsolni a programot. Természetesen erre rengeteg megoldás volt az interneten, viszont mindegyik más és más volt, mivel a probléma nem mindenkinél ugyanaz. Az elején nem is gondoltam arra, hogy PHP-val kötöm össze az alkalmazást. Viszont a kutatómunkám eredménye az lett, hogy ez a legkényelmesebb megoldás számomra.

## 3.9 Tesztdokumentáció

- **Belépési teszt:** Amennyiben belépéskor a felhasználó nem ad meg, vagy rosszul ad meg bármilyen adatot, abban az esetben a program jelzi.
- **Regisztrációs teszt:** Abban az esetben, ha a felhasználó regisztrációnál nem ad meg bármilyen adatot, abban az esetben a program ezt jelzi.
- **Foglaltság teszt:** Ha a felhasználó által beírt név vagy email cím foglalt, abban az esetben a program ezt jelzi.
- **Internet kapcsolat teszt:** Abban az esetben, ha használat közben az internetkapcsolat megszakad, a

## Feladat

program nem reagál semmilyen inputra.

Admin felület teszteléséhez szükséges belépési adatok:

- Felhasználó név: elso
  - Jelszó: elso
- 

## 4. Irodalomjegyzék

---

- Színválasztás:  
[https://www.w3schools.com/colors/colors\\_picker.asp](https://www.w3schools.com/colors/colors_picker.asp)
- PHP parancsok:  
<https://www.php.net/manual/en/index.php>
- Java segítség:  
<https://www.w3schools.com/java/>
- Java olvasmány:  
*Alkalmazásfejlesztés Android Studio rendszerben*  
[https://www.w3schools.com/colors/colors\\_picker.asp](https://www.w3schools.com/colors/colors_picker.asp)  
<https://www.w3schools.com/java/>