

SZAKVIZSGA 2021

LemonApp

BABOS DÁVID
BOLYAI FARKAS ELMÉLETI LÍCEUM
XII.G

FELKÉSZÍTŐ TANÁR
JAKAB IRMA-TÜNDE

Tartalomjegyzék

Tartalomjegyzék

Témaindoklás

Felhasználói kézikönyv

Regisztráció

Bejelentkezés

Idővonal

Értesítések

Felhasználói profil

Posztok

Posztokról általánosan

Hozzászólások

Új poszt létrehozása

Más felhasználókkal való interakció

Rendszer követelmények

Hardver igény

Szoftver igény

Programozói kézikönyv

Általános információk

Felépítés

Adatbázis

Kommunikáció a szerverrel

Kép feltöltés

Fejlesztési lehetőségek

Profilkép és jelszó módosítása

Videó feltöltés a posztokhoz

Dark mode

Közvetlen üzenetküldés

“Story”-k megosztása

Szakirodalom

Témaindoklás

Témául a közösségi médiát választottam, több okból kifolyólag is: a közösségi média szerepe a minden nap életünkben, technikai lehetőségek és továbbfejlesztési lehetőségek.

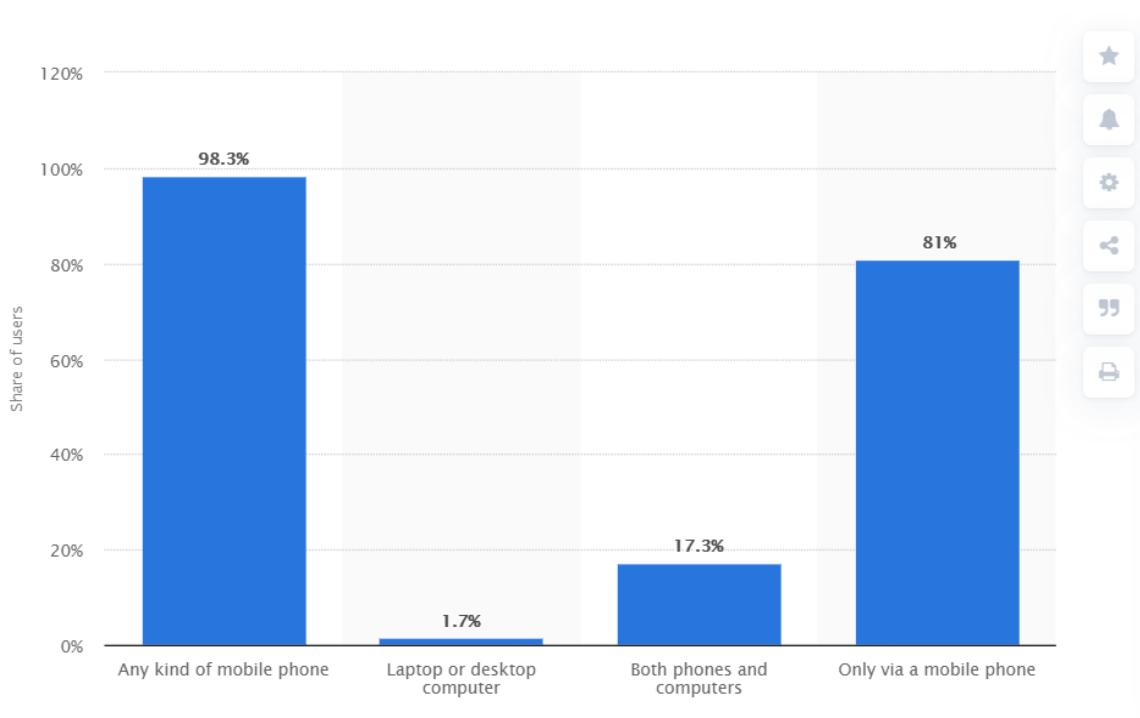
A közösségi médiának nagy szerepe van a minden napjainkban, manapság minden korosztály rendszeres felhasználó a különböző közösségi médiákon. Ebből kifolyólag lehetőségem volt különböző funkcióinálítások kifejlesztésére és ennek következményeként különböző kihívások elé állítottam saját magam.

A döntésemet az is befolyásolta, hogy a közösségi média jó lehetőség lehet az emberek számára a kapcsolatteremtésre, főként ebben a nehéz időszakban, amikor személyes találkozás nem mindig lehetséges.

A téma választásában nagy szerepet játszott az egyéni fejlődés lehetősége, fontos volt számomra, hogy kilépjek a komfort zónámból és olyan projekten dolgozzak amely biztosítja a technikai fejlődés lehetőségét.

A közösségi médiák több platformon is megjelennek, mint például Androidos és iOS-es telefonokon és webes felületen is. A Statista 2021-es felmérése¹ szerint a felhasználók 98,3%-a használ közösségi médiát a telefonján, így úgy döntöttem, hogy Android eszközökre fogok fejleszteni. Ehhez a döntéshez az is hozzájárult, hogy már van tapasztalatom Android applikációk fejlesztésében.

¹ <https://www.statista.com/statistics/377808/distribution-of-facebook-users-by-device/>



© Statista 2021

A téma továbbfejlesztési lehetőségek szempontjából is kedvező, a közösségi médiát több irányba is lehet fejleszteni és optimalizálni. Lehetőség nyílik különböző technológiák, adatbázisok és programozási nyelvek használatára is, valamint business, marketing, vagy szociológiai szempontból vizsgálva is több lehetőséget nyújt mint a legtöbb szoftver.

Mivel a közösségi média egy nagyon tág fogalom, tudatosan kellett választanom a lehetséges funkcionálitások között. Első sorban a tartalom írás (mint szöveg, vagy komment formájában) és digitális fotómegosztás fejlesztésével foglalkoztam, mivel a Wikipédia szerint a Facebook felhasználók ezt a két funkcionálitást veszik igénybe leggyakrabban². Létrehoztam az idővonalaat is amely az applikáció fő funkcionálitásává vált.

² https://hu.wikipedia.org/wiki/K%C3%B3z%C3%A1ss%C3%A1g_m%C3%A1dia

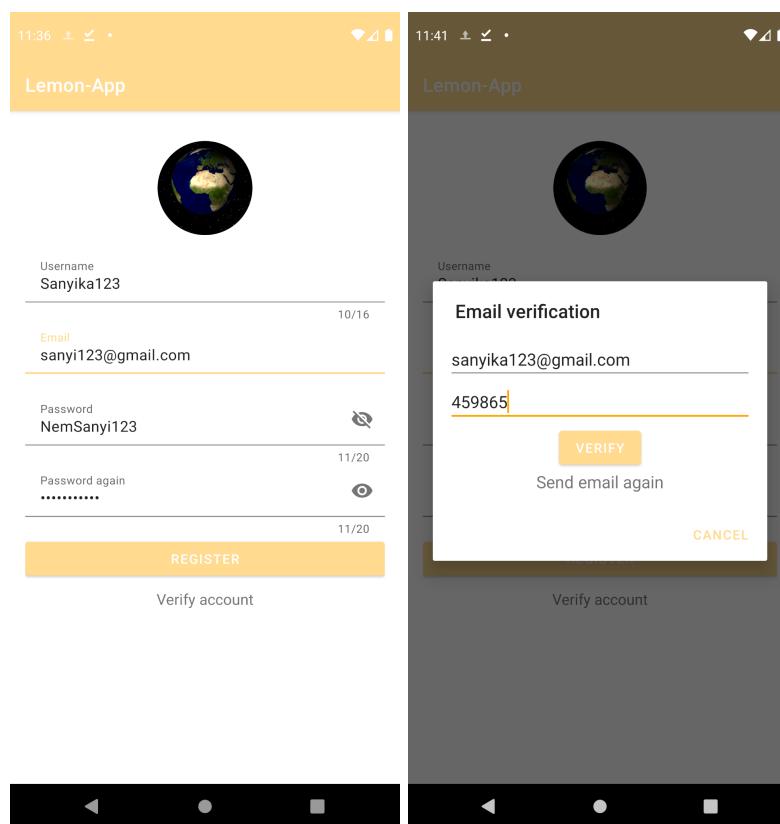
Felhasználói kézikönyv

Regisztráció

Regisztráció az első lépés ahhoz, hogy a felhasználó használni tudja az appot. Regisztrációhoz szükséges egy egyedi felhasználónév, egy létező email cím, és jelszó (a jelszót meg kell erősíteni). Opcionálisan fel lehet tölteni egy profilképet. A regisztráció során érkezik egy email, amelyben egy token segítségével meg kell erősíteni a felhasználót.

A sikeres regisztrációhoz szükséges:

- Egyedi felhasználónév, amely tartalmazhat kis- és nagybetűket a magyar ABC-ből és számokat, maximum 16 karakterből állhat.
- Egyedi email cím, egy email címhez csak egy felhasználó tartozhat.
- Jelszó amely tartalmazhat kis- és nagybetűket a magyar ABC-ből, számokat és minimum 6 és maximum 20 karakterből kell álljon.
- Profilkép (opcionális) amelyet a saját eszkösről kell kiválasztani, PNG, JPG vagy GIF formátumban.

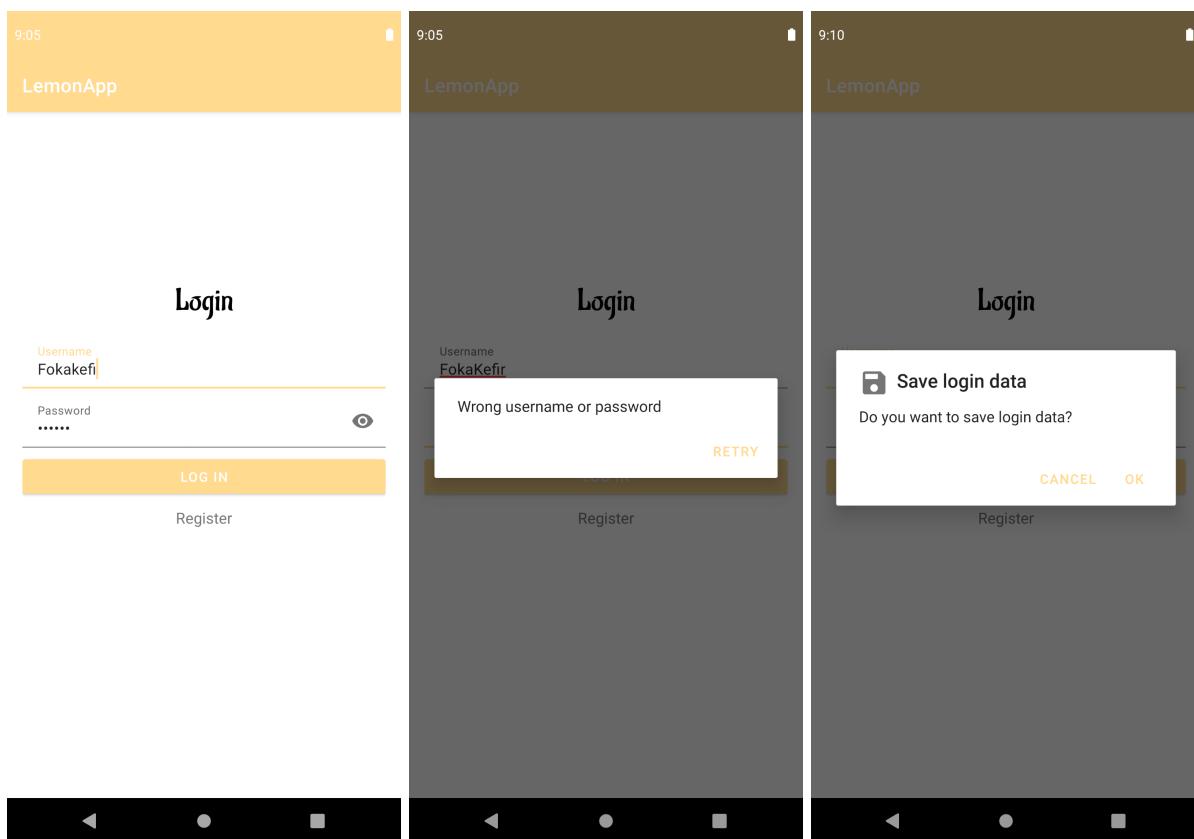


Bejelentkezés

A sikeres bejelentkezéshez szükséges:

- Egyedi felhasználónév, amelyet a felhasználó a regisztráció során választott.
- Jelszó, amelyet a felhasználó a regisztráció során választott.
- A felhasználó meg kell legyen erősítve az emailben érkezett token segítségével

Helytelen felhasználónév vagy jelszó esetén az applikáció hibaüzenetet ír ki. Sikeres bejelentkezés esetén választható, hogy az applikáció megjegyezze a bejelentkezéshez szükséges adatokat. A bejelentkezés után a felhasználó az idővonalat fogja látni.

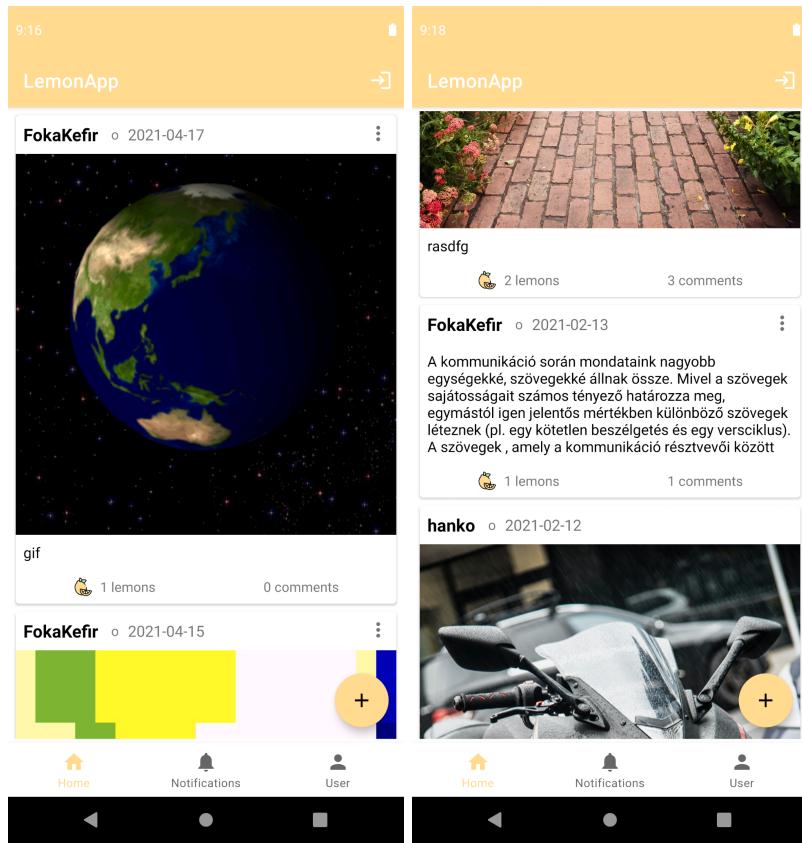


Idővonal

Az idővonal a "Home" fülön jelenik meg.

Az idővonalon különböző posztok jelennek meg, első sorban írott tartalom, sok esetben képpel párosítva. Az itt megjelenő posztok tartozhatnak a felhasználóhoz vagy a felhasználó által követett felhasználókhöz. A posztok kronológiai sorrendben jelennek meg, legújabbtól a legrégebbiig, vízszintesen követve egymást.

Az idővonalat frissíteni lehet, hogy ha a "Home" fülre kattint a felhasználó vagy ha az idővonal tetején a felhasználó felfelé próbál görgetni.



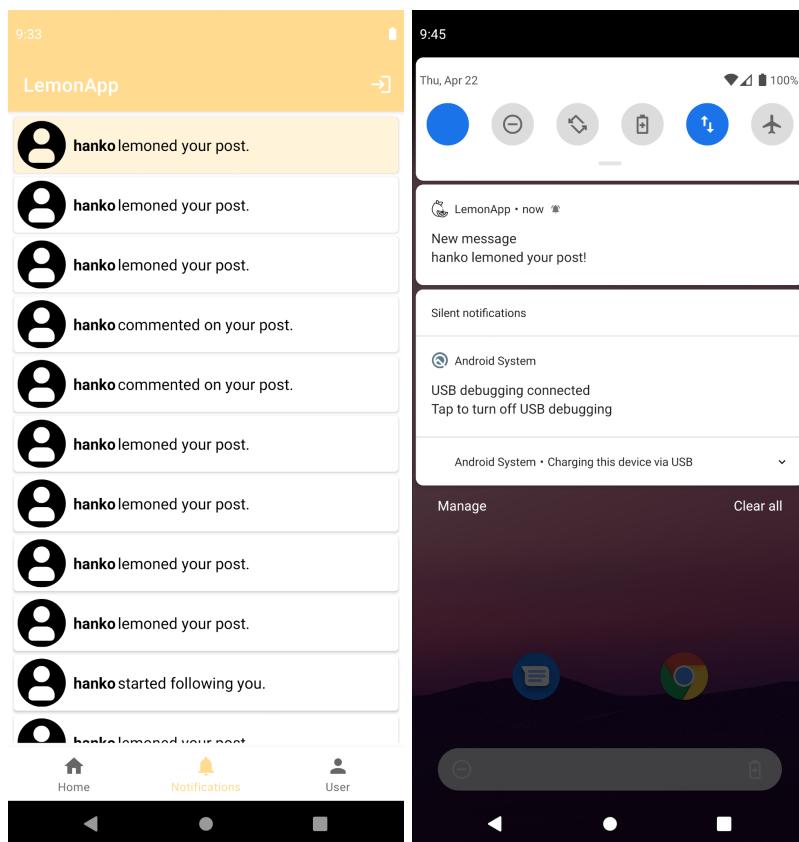
Értesítések

Az értesítések a "Notifications" fülön jelennek meg, az idővonalhoz hasonló módon, vízszintesen, kronológiai sorrendben. Egy felhasználó értesítést kaphat ha reagálnak vagy hozzászólnak egy posztjához vagy valaki elkezdi követni. A még meg nem nézett értesítések háttérszíne színes, ami akkor változik fehérre ha a felhasználó megnézi az értesítést. Ha egy interakciót visszavonnak és még nincs megnézve, akkor az értesítés automatikusan törlődik a listából.

Minden értesítés az applikáción kívül is megjelenik, ezekre kattintva megnyílik a "Notifications" fül.

Egy értesítés tartalmazza a feladó felhasználónevét és profilképét és az interakció típusát. A felhasználónévre vagy profilképre kattintva megnyitható az adott felhasználó profilja és az értesítés többi részére kattintva, értesítéstől függően, az adott poszthoz vagy a felhasználó oldalához irányít.

Az értesítések frissülnek, ha a "Notifications" fülre kattint a felhasználó vagy ha az értesítések tetején a felhasználó felfelé próbál görgetni.



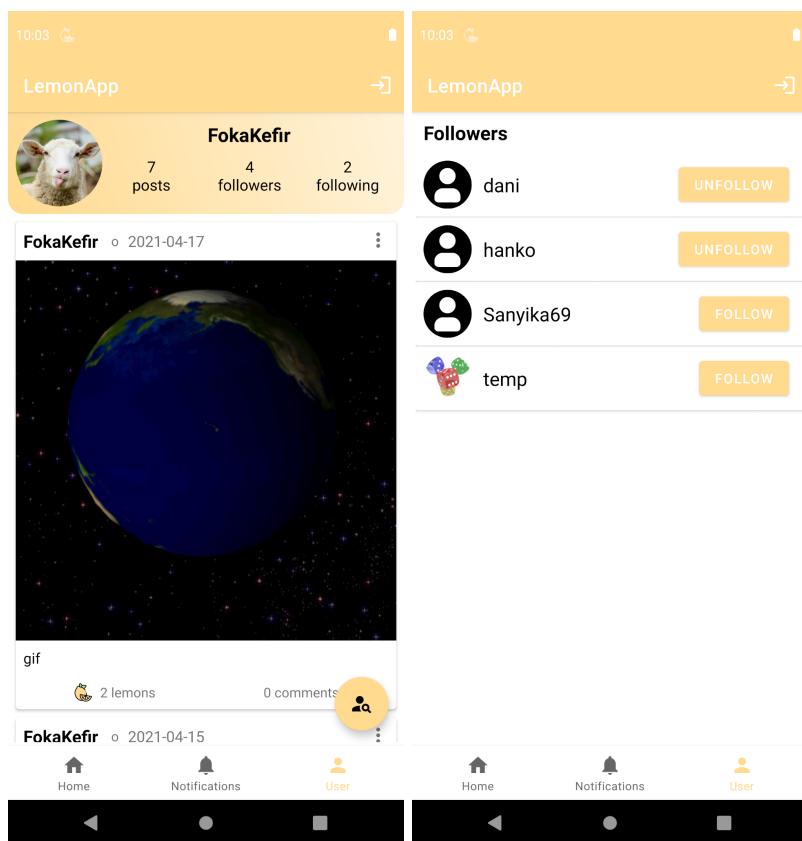
Felhasználói profil

A felhasználói profil a “User” fülön jelenik meg.

A profil tartalmazza a felhasználó nevét, profilképét, posztjainak a számát, a követőinek a számát, az általa követett személyek számát, és a posztjait, az idővonalhoz hasonló formátumban. Ezen az oldalon található egy keresés gomb is, amely segítségével más felhasználókra lehet rákeresni.

A “followers” -re és a “following”-ra kattintva meg lehet tekinteni a követők vagy a követett felhasználók listáját.

A felhasználói profil akkor frissül, ha a “User” fölre kattint a felhasználó vagy ha a felhasználói idővonal tetején a felhasználó felfelé próbál görgetni.



Posztok

Posztokról általánosan

A posztokat az idővonalon, a felhasználói profilon vagy az értesítésekben megnyitva lehet megtekinteni.

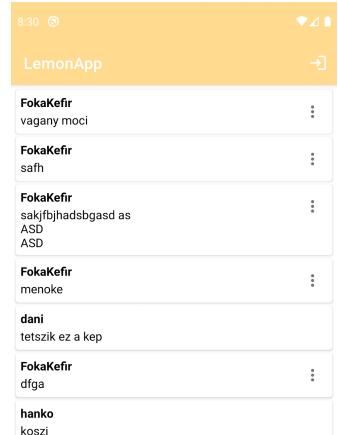
Minden poszt tartalmazza a szerző felhasználónevét, a poszt létrejöttének dátumát, egy képet (opciónális), az írott tartalmat, a reakciók és kommentek számát, valamint saját posztnál egy opció gombot, amely segítségével a posztot törölni lehet.

A citrom ikonra kattintva a posztokra lehet reagálni vagy visszavonni róluk reakciót; a reagált posztok alatt a citrom színes változata látható.



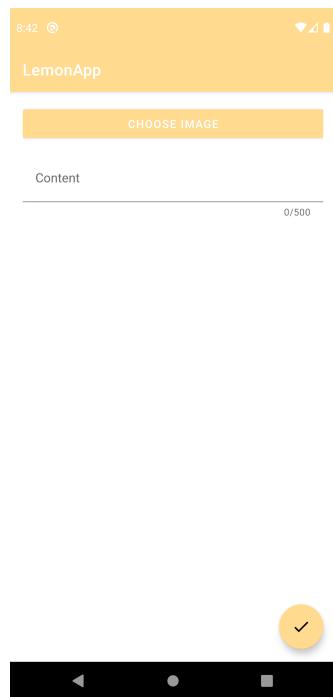
Hozzászólások

A "comments"-re kattintva megnyithatók a poszt hozzászólásai. A kommentek az idővonalhoz hasonló formátumban jelennek meg, csak ebben az esetben a legrégebbitől kezdve a legújabbig. Egy hozzászólás tartalmazza a hozzászóló felhasználónevét és az üzenetet. Egy felhasználó saját hozzászólása valamint saját posztnál bárki hozzászólása törölhető az opció gomb segítségével. Új hozzászólást lehet írni a beviteli mező segítségével, majd az elküldés gombbal lehet azt elmenteni.



Új poszt létrehozása

Az idővonalon a “+” gombra kattintva nyílik meg a poszt szerkesztő amelynek segítségével új poszt hozható létre. A szerkesztő lehetőséget biztosít képfeltöltésre és tartalomírásra. Amit az elküldés gombbal lehet véglegesíteni és posztolni, ami után az applikáció az idővonalra navigál.

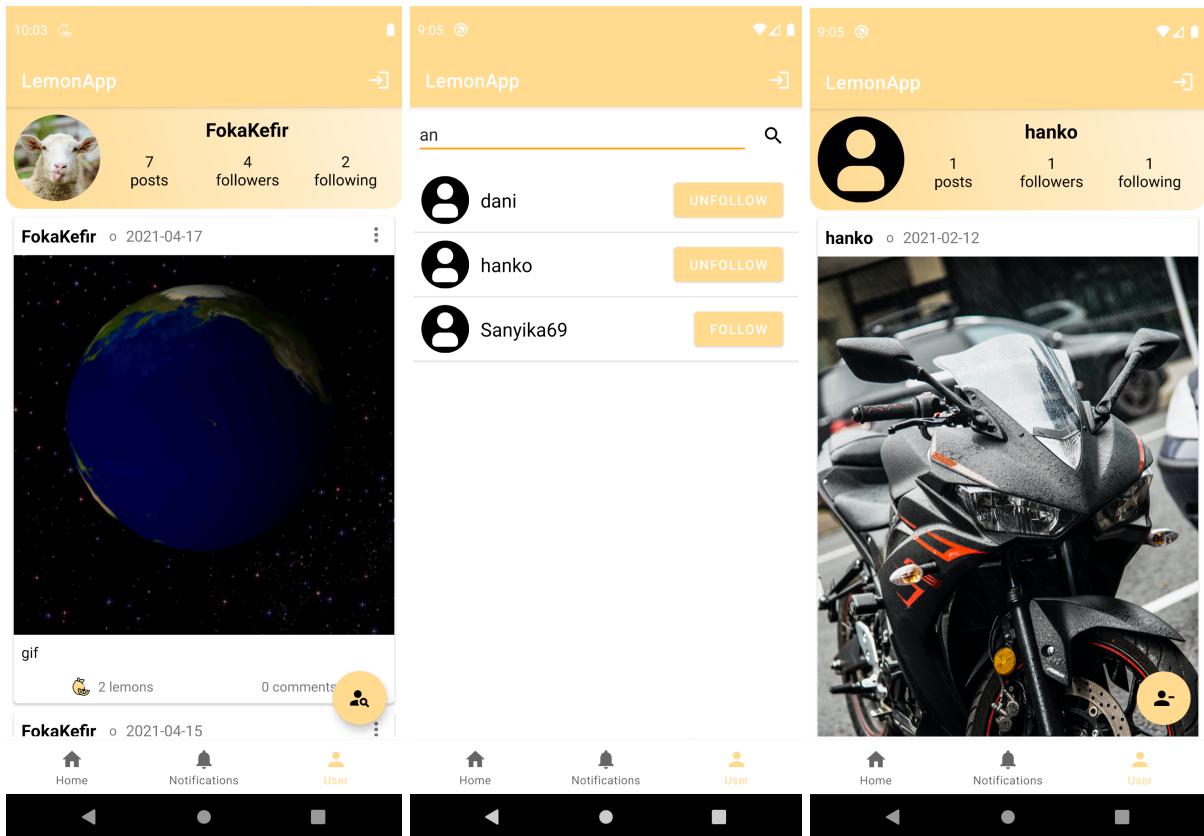


Más felhasználókkal való interakció

Minden felhasználónak lehetősége van más felhasználókat követni, és ezáltal lehetőség nyílik a felhasználók közötti interakcióra. A bekövetett felhasználó posztjai is megjelennek az idővonalon, így lehetőség nyílik reagálni és hozzászólni azokhoz a posztokhoz.

A felhasználó idővonalán megjelenő keresés gombbal megnyitható a felhasználó kereső. A keresőn más felhasználókra lehet rákeresni a felhasználónevük alapján. A felhasználók névsor szerint jelennek meg, először azok akiknek a nevük a keresett karakterekkel kezdődik, majd azok a felhasználók akiknek a nevükben szerepel az adott szó. Itt be és ki lehet követni az adott felhasználókat, vagy megtekinteni a profilukat.

Egy rákeresett felhasználó profilján megjelennek az adott felhasználó adatai és egy gomb a jobb alsó sarokban, amivel be vagy ki lehet követni a rákeresett felhasználót.



Rendszer követelmények

Hardver igény

Bármilyen android eszköz (legjobb élményért telefon) vagy emulátor, minimum 16 MB szabad tárhellyel.

Szoftver igény

Minimum Android 8-as eszköz, minél jobb internet kapcsolattal.

Programozói kézikönyv

Általános információk

Az applikáció az Android Studio segítségével lett létrehozva a Java és XML programozási nyelvet használva. A szerver PHP-ban íródott és az MySQL relációs adatbázis-kezelő szerver segítségével módosítja az adatbázisban lévő adatokat. Az értesítések küldéséhez pedig Firebase van használva.

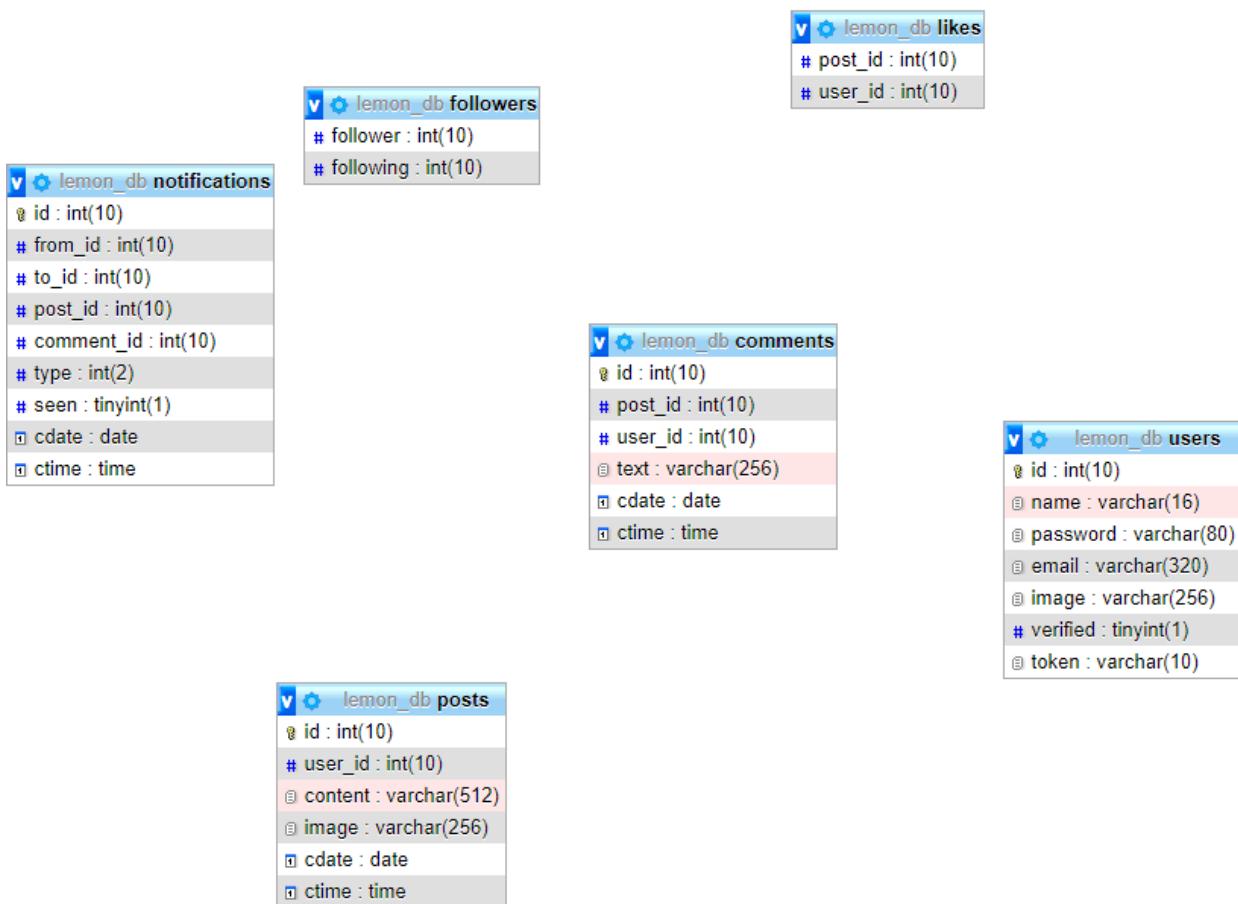
Felépítés

Az applikáció 4 db activity-ből áll:

1. LoginActivity, ez jelenik meg amikor az applikációt a felhasználó elindítja.
2. RegisterActivity, ezt a Login activity-in keresztül lehet elérni.
3. MainActivity az egyik legfontosabb activity-t ami összefoglal magában több fragmentet, köztük: PostsFragment, CommentsFragment, NotificationsFragment és UserFragment. A MainActivity bejelentkezés során válik elérhetővé és kijelentkezással visszalép a LoginActivity-be. Itt tölti el a felhasználó a legtöbb idejét az applikáció használata során.
4. CreatePostActivity, amelyet a PostsFragment-ről lehet elérni és amely ugyanarra a PostFragment-re visz vissza a bezárása után.

Adatbázis

Az applikáció a MySQL-t, egy SQL-alapú relációs adatbázis-kezelő szervert, használ az adatok módosítására az adatbázisban. Az SQL alapú relációs adatbázis használata kedvező ebben a projektben, mert az applikációnak fontos része a különböző adatok közötti kapcsolat. A relációs adatbázisok az adatok közötti kapcsolatot könnyen lehetővé teszik, amely a közösségi média applikációk egyik legfontosabb része.



Táblák és oszlopok az adatbázisban

Kommunikáció a szerverrel

A szerverrel való kommunikáció REST API használatával történik, amely egy olyan alkalmazásprogramozási felület (API vagy web API), amely megfelel a REST architektura stílus megszorításainak, és lehetővé teszi a RESTful web szolgáltatásokkal való interakciót. A kommunikáció ebben az esetben HTTP-n keresztül történik.

A front-end egy API hívás segítségével küld információt a szervernek, a következő példában látható a kód amely lehetővé teszi a regisztrációt.

```
public void register(String name, String password, String email, String image) {  
    Map<String, String> params = new HashMap<>();  
    params.put("name", name);  
    params.put("password", password);  
    params.put("email", email);  
    params.put("image", image);  
    DataRequest dataRequest = new DataRequest(params, REGISTER_REQUEST_URL, listener: this, errorListener: this);  
    Volley.newRequestQueue(this.context).add(dataRequest);  
}
```

Az API meghívása regisztráció esetén

A Request létrehozásakor meg kell adni az adott URL-t és az adatokat HashMap formában elmentve, ezeket pedig a szerver fogja fogadni.

A szerver ezeket az adatokat feldolgozza és helyes adatok esetén elvégzi a műveleteket az adatbázison, majd pedig JSON formátumban az adatokból létrejött választ a front-end-nek. A visszaküldött válasz tartalmazhat adatokat, vagy egy hibaüzenetet.

```
<?php  
  
include $_SERVER['DOCUMENT_ROOT'].'/lemon_app/php/config.php';  
  
// Connecting to database  
$conn = new mysqli(DB_HOST, DB_USER, DB_PASS, DB_NAME);  
  
// Checking error  
if (mysqli_connect_errno()) {  
    echo "Failed to connect to MySQL: ".mysqli_connect_error();  
    die();  
}  
  
// Getting the params  
$name = $_POST['name'];  
$password = $_POST['password'];  
$email = $_POST['email'];  
$image = $_POST['image'];  
  
// Creating response  
$response = array();  
  
// Checking the name  
if (checkingName($conn, $name) == false) {  
    $response['success'] = false;  
    $response['message'] = "Username is not available";  
}  
  
//Checking the email  
} else if (checkingEmail($conn, $email) == false){  
    $response['success'] = false;  
    $response['message'] = "Email is not available";  
} else {  
    // Generate hashed password  
    $hashed_password = hash('sha256', $password);  
  
    $token_number = rand(100000, 999999);  
}
```

```
@Override  
public void onResponse(String response) {  
    try {  
        JSONObject jsonResponse = new JSONObject(response);  
        boolean success = jsonResponse.getBoolean("name: "success");  
  
        if (success) {  
            String token = String.valueOf(jsonResponse.getInt("name: "token"));  
            this.onResponseListener.onSuccessfulRegisterResponse(token);  
        } else {  
            String errorMessage = jsonResponse.getString("name: "message");  
            this.onResponseListener.onFailedRegisterResponse(errorMessage);  
        }  
    } catch (JSONException e) {  
        e.printStackTrace();  
    }  
}
```

A szerverről érkezett válasz feldolgozása a front-end-en

A szerverről érkezett választ JSON formátumba alakítja át String formátumból, majd ebből pedig kiolvassa a művelet sikerességét. A fenti példában, amennyiben sikeres volt a regisztráció a JSON válaszból kiolvasott tokent elmenti, amire később szükség van a további kommunikációhoz a szerverrel. Sikertelen regisztráció esetén az applikáció kiírja a visszatérített hibaüzenet.

Kép feltöltés

A kép feltöltés hasonlóan történik az adatok feltöltésével, csak ebben az esetben a front-end egy képfájlt is elküld a back-end-nek.

```
private void uploadImage() {
    String path = getPath(this.filePath);

    try {
        String uploadId = UUID.randomUUID().toString();

        //Creating a multi part request
        new MultipartUploadRequest( context: this, uploadId, UPLOAD_IMAGE_REQUEST_URL )
            .addFileToUpload(path, parameterName: "image")
            .addParameter( paramName: "name", this.strImage)
            .setNotificationConfig(new UploadNotificationConfig())
            .setMaxRetries(2)
            .startUpload();

    } catch (Exception exc) {
        Toast.makeText( context: this, exc.getMessage(), Toast.LENGTH_SHORT).show();
    }
}
```

A front-end csatolja a képfájlt az API híváshoz

A szerver megkapja a kép nevét és magát a képfájlt, amit beszúr az "images" mappába, ide vannak elmentve a profilképek és a különböző posztokhoz tartozó képek. A front-end a URL-n keresztül referál a képekre.

```
<?php
include $_SERVER['DOCUMENT_ROOT'].'/lemon_app/php/config.php';

$upload_path = "images/";
$server_ip = SERVER_IP;
$upload_url = 'http://'.$server_ip.'/lemon_app/'.$upload_path;

$response = array();
$response['error'] = true;

if($_SERVER['REQUEST_METHOD']=='POST'){
    if(isset($_POST['name']) and isset($_FILES['image']['name'])){
        $name = $_POST['name'];

        $fileinfo = pathinfo($_FILES['image']['name']);
        $extension = $fileinfo['extension'];

        $file url = $upload url . $name . '.jpg';
        $file_path = "C:/xampp/htdocs/lemon_app/images/" . $name . '.jpg';

        try{
            move_uploaded_file($_FILES['image']['tmp_name'], $file_path);

            $response['error'] = false;
            $response['url'] = $file_url;
            $response['name'] = $name;
        }catch(Exception $e){
            $response['error']=true;
            $response['message']=$e->getMessage();
        }
    }else{
        $response['error']=true;
        $response['message']='Please choose a file';
    }
}
echo json_encode($response);
```

Fejlesztési lehetőségek

Profilkép és jelszó módosítása

Regisztráció után a felhasználó adatait nem lehet módosítani. Gyakran előfordul a felhasználóknál, hogy az adott profilképüket megunják vagy a jelszavukat elfelejtik, ezért hasznos lehetőség lenne, egy olyan oldal, ahol az adatokat tudják módosítani.

Videó feltöltés a posztokhoz

Egyenlőre a posztokhoz csak képeket lehet feltölteni, egy hasznos opció lenne, ha a felhasználó képek mellett még videókat is fel tudna tölteni.

Dark mode

Legtöbb közösségi oldalon már elérhető a Dark mode, ami megváltoztatja az egész applikáció designját sötétebb színekre, ezzel a felhasználó szemét kímélve.

Közvetlen üzenetküldés

Legtöbb közösségi oldalhoz tartozik közvetlen üzenetküldő applikáció is. Ezzel a lehetőséggel a felhasználó az általa követett személyeknek személyes üzenetet tudna küldeni és egymással tudnának csevegni.

“Story”-k megosztása

A “Story”-k a felhasználónak lehetővé teszik, hogy olyan képet vagy videót tudjanak megosztani, ami csak 24 órán keresztül tekinthető meg.

Szakirodalom

Forrás: Statista

Látogatás dátuma: 2021. Március

Link: [statista.com](https://www.statista.com)

Forrás: Wikipédia

Látogatás dátuma: 2021. Március

Link: [wikipedia.org](https://en.wikipedia.org)

Forrás: Firebase

Látogatás dátuma: 2021. Április -

Link: firebase.google.com

Forrás: Android Developers

Látogatás dátuma: 2021. Április -

Link: developer.android.com

Forrás: Stack Overflow

Látogatás dátuma: 2021. Április -

Link: stackoverflow.com

Forrás: GitHub

Látogatás dátuma: 2021. Április -

Link: github.com

Forrás: Coding in Flow

Látogatás dátuma: 2021. Április -

Link: elopage.com/s/codinginflow

Forrás: Geeksforgeeks

Látogatás dátuma: 2021. Április

Link: geeksforgeeks.org