

# **Socialno omrežje – *BandWith***

Strokovno poročilo – poklicna matura

# Povzetek

V poročilu je predstavljeno socialno omrežje BandWith ter glavne komponente, iz katerih je sestavljeno. Izdelek je rezultat dela 4-ih let Vegove in zanimanja/radovednosti o razvijanju mobilnih aplikacij. Izdelek je narejen s tehnologijami Flutter, Dart, PHP, SQL, HTML, CSS, JavaScript. Namenjen je glasbenikom, ki želijo spoznati druge glasbenike in z njimi ustvarjati nova prijateljstva.

## Ključne besede:

Uporabnik, vmesnik, strežnik, aplikacija, podatek, podatkovna baza

# Abstract

In this report, a social media BandWith and its main components are examined. The project is a result of 4 years of work on Vegova and curiosity/interest in developing mobile applications. The app is made with the following technologies: Flutter, Dart, PHP, SQL, HTML/CSS/JavaScript. It is intended for musicians looking to meet and make connections with other musicians.

## Keywords:

User, interface, server, application, data, database

# Kazalo vsebine

## UVOD1

Koncept1.1

## ORODJA IN TEHNOLOGIJE2

Flutter2.1

PHP2.2

SQL2.3

HTML/CSS/JavaScript2.4

Visual Studio Code2.5

XAMPP2.6

## IZDELAVA IZDELKA3

Podatkovna baza3.1

API3.2

Mobilna aplikacija3.3

Administratorska konzola3.4

## ZAKLJUČEK4

## VIRI5

# Uvod

Zaradi zanimanja o mobilnih aplikacijah in želje po izzivu sem si za maturitetni izdelek izbral mobilno socialno omrežje. Ker se že nekaj časa ukvarjam tudi z glasbo, sem se odločil, da naredim socialno omrežje za spoznavanje glasbenikov. Aplikacija deluje na podoben princip, kot večina ostalih socialnih omrežij – uporabnik nanjo naloži osnovne informacije o sebi kot so ime, priimek, slika itd. Posebnost te aplikacije je, da uporabnik pove, katere instrumente igra. V aplikacijo lahko naloži tudi zvočni posnetek svoje glasbe. Na glavni strani aplikacije se prikažejo kartice drugih glasbenikov, na katerih so prikazane glavne informacije uporabnika – slika, ime in priimek ter zvočni posnetek. Uporabnik lahko druge glasbenike všečka, do česar lahko kasneje dostopa na ekranu s priljubljenimi. Aplikacija vsebuje tudi funkcijo pogovorov. Uporabniki lahko med seboj komunicirajo preko privatnih direktnih sporočil.

# Orodja in tehnologije

Pri izdelavi izdelka sem si pomagal z naprednimi okolji za programiranje in visokonivojskimi tehnologijami za ustvarjanje aplikacij. Ta orodja so mi pomagala do zelenega rezultata z visoko kvaliteto ter v najkrajšem možnem času.

## Flutter

Flutter je Googlovo odprtokodno orodje za izdelavo uporabniških vmesnikov za platforme Android, iOS in splet. Uporablja programski jezik Dart, ki je po zgradbi podoben JavaScriptu z nekaterimi elementi Jave.

Flutter koda je zgrajena iz vizualnih gradnikov (angl. widget-ov). To so vnaprej ustvarjeni vmesniški elementi. S tem se postavljanje uporabniškega vmesnika pospeši, saj ustvarjanje pogostih, a kompleksnih elementov ni potrebno.

Glavna posebnost Flutterja in tudi razlog, da sem ga izbral za ta projekt je, da lahko brez večih programskih jezikov vzporedno razvijamo za več platform hkrati. Aplikacija tako podpira obe mobilni platformi brez potrebe po pisanju kode v Swiftu za iOS ali Kotlinu za Android.

Flutter je primarno napisan v jeziku C++, kar mu omogoča neposredno nizkonivojsko upodabljanje slike na zaslon naprave. Posledica tega je izjemno hitro risanje elementov in brežhibno izvajanje animacij.

Flutter med drugim podpira tudi funkcijo 'hot reload', ki uporabniku omogoča sprotno shranjevanje in testiranje kode v parih sekundah. Ta funkcija izloči ponavljajoče shranjevanje in nalaganje kode na mobilno napravo, kar mi je pospešilo razvoj.

```

2   class MyApp extends StatelessWidget {
3     @override
4     Widget build(BuildContext context) {
5       return MaterialApp(
6         home: Scaffold(
7           body: Center(
8             child: Text(
9               'Hello world!',
10            ), // Text
11          ), // Center
12        ), // Scaffold
13      ); // MaterialApp
14    }
15  }

```

*Slika 1: Primer Flutter kode*

## PHP

PHP je odprtokodni programski jezik, ki ga lahko uporabljamo za razvijanje strežniških aplikacij ali dinamičnih spletnih vsebin. Za njegovo delovanje potrebujemo strežnik z vgrajenim tolmačem, ki PHP prevede in ga brskalniku pošlje v obliki HTML kode.

PHP sem na svoji spletni strani uporabil za strežniški del. Z njim sem dostopal do podatkovne baze, podatke obdelal in jih posredoval uporabniku. Zanj sem se odločil zaradi preprostosti uporabe in njegove uveljavljenosti. Z njim smo delali tudi pri pouku, zato sem imel manj dela z učenjem, kar mi je omogočilo hiter začetek.

```
1  <?php
2  |      echo 'Hello world!';
3  ?>
```

*Slika 2: Primer PHP kode*

## SQL

SQL ali structured query language je najbolj razširjen jezik za delo s podatkovnimi bazami na trgu. Sestoji iz poizvedb različnih tipov, kot so poizvedbe za izbiranje in izvršne poizvedbe.

Sam sem ga uporabil za kreacijo podatkovne baze in beleženje ter posodabljanje podatkov uporabnikov. Uporabljal sem ga v kombinaciji z orodjem MySQL, enim izmed najbolj popularnih sistemov za upravljanje SQL podatkovnih baz.

```
1  SELECT * FROM user WHERE Name = 'Janez'
```

*Slika 3: Primer SQL poizvedbe*

## HTML, CSS in JavaScript

HTML, CSS in JavaScript so tehnologije, ki se skupaj uporabljajo za ustvarjanje dinamičnih spletnih strani. HTML služi kot označevalni jezik, s katerim napravi sporočimo osnovno hierarhijo strani. S CSSom nato to vsebino postavimo in jo stiliziramo. JavaScript pa je programski jezik, ki podpira objektno-orientirano programiranje. Z njim lahko strani dodamo dinamične vsebine in interaktivnost.

Te tehnologije sem uporabil pri izdelavi administratorske konzole.

## Visual Studio Code

Kot glavno okolje za pisanje kode sem si izbral Microsoftov VS Code. Izšel je šele leta 2015 vendar je hitro prevzel več kot polovico tržnega deleža, število uporabnikov pa še vedno raste.

Izbor urejevalnika je bil eden izmed ključnih za hitrost razvijanja. Funkcije kot so uporaba različnih barv, avtomatska poravnava in avtomatski predlogi so znatno pospešile tok dela.

VS Code je bil prava izbira tudi zaradi možnosti dodajanja razširitev. Tako sem lahko izkoristil prednosti vgrajenih funkcij za Flutter in Dart.

## XAMPP

XAMPP je odprtokoden paket za razvijanje PHP aplikacij. Sam sem uporabljal funkcijo Apache serverja ter že omenjeni sistem MySQL.

Za XAMPP sem se odločil predvsem zato, ker smo ga uporabljali pri pouku. Je zelo podoben večjim spletnim strežnikom, kar naredi prehod na drug strežnik toliko lažji.



# Izdelava izdelka

Za potek razvijanja sem si izbral iterativni model. Ker osnovna ideja aplikacije ni bila popolnoma razdelana, je bil to najboljši način. Tako sem sproti videl, katere funkcije so bile uporabniku prijazne in katere odveč. Sproti sem tudi testiral hitrost in učinkovitost aplikacije ter probleme reševal sproti.

## Podatkovna baza

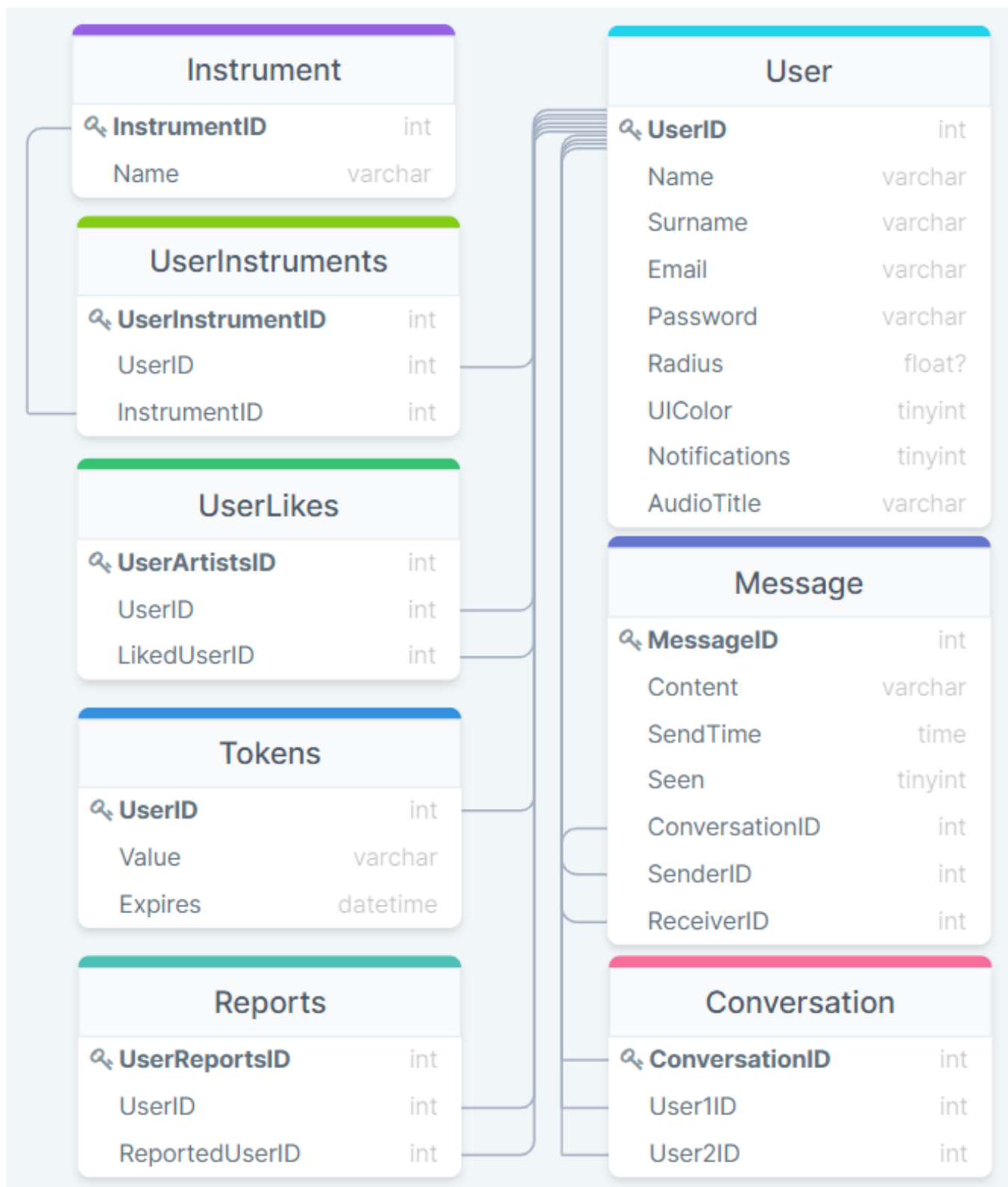
Podatkovna baza je eden izmed ključnih elementov aplikacije, saj funkcionira kot shramba podatkov vseh uporabnikov in povezav med njimi. Pri načrtovanju baze sem pazil, da je struktura čim bolj vitka, vendar še vedno podpira vse zelene funkcije aplikacije. Pozoren sem bil tudi na ACID strukturo, ki smo se je učili v 3. letniku pri predmetu PPB. Ta zagotavlja atomarnost, konsistenco, izolacijo, in vzdržljivost podatkov.

Sprva sem osnutek baze narisal na papir v obliki tabel s črtami, ki označujejo podatkovne relacije. Nato sem SQL kodo za kreacijo tabel napisal v beležnico in sproti dodajal oz. odstranjeval attribute posameznih tabel.

Ko je bil napisan prvi osnutek kode, sem ga prenesel v phpMyAdmin konzolo in tam izvedel. Koda ob prvem poskusu ni delovala, tako sem jo prilagajal še nekaj časa, dokler nisem dobil prve oblike podatkovne baze.

Od prve oblike podatkovne baze v resnici ni ostalo veliko, saj sem jo skozi proces grajenja aplikacije stalno spreminjal in brisal oz. dodajal tabele in attribute. Bazo sem skozi proces postopoma zmanjšal in tako prihranil na prostoru ter nepotrebnih relacijah.

Končna baza sestoji iz 8 tabel.



Slika 4: ER model podatkovne baze

## API

Za programiranje API-ja sem si izbral jezik PHP, saj smo z njim delali tudi pri pouku. Ta izbira je preprečila zapravljanje časa z učenjem sintakse, obenem pa je prednost PHP-ja tudi ta, da ima ogromno podpore na internetu.

Grajenja API-ja sem se lotil zgledujoč se na uporabniški tok aplikacije. Začel sem s prijavo in registracijo, nato sem sprogramiral nastavitve in dodajanje instrumentov, potem sem dodal stran za odkrivanje in všečkanje drugih glasbenikov, na koncu pa sem dodal še stran s pogovori in všečkanimi uporabniki.

### Assets

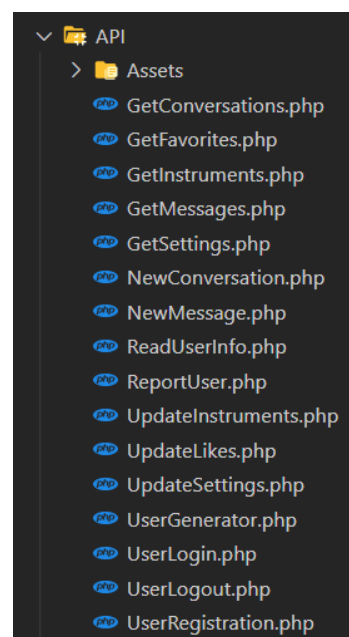
Mapa Assets vsebuje dve podmapi in sicer Images in Audio. V nji se shranjujejo profilne slike in zvočni posnetki uporabnikov. Za to metodo sem se odločil, ker bi bilo neposredno shranjevanje v bazo počasnejše in težje.

### UserLogin in UserRegistration

To sta bili prvi datoteki, ki sem jih ustvaril za API. Datoteka UserRegistration sprejme podatke o uporabniku, preveri, ali je ta email v bazo že vpisan. Če je temu tako, vrne sporočilo o napaki, drugače izvede vpis v tabelo User in tako ustvari novega uporabnika. UserLogin prebere poslane podatke in jih primerja s podatki shranjenimi v bazi. Če sta geslo ali e-naslov neveljavna, vrne primerno sporočilo, drugače pa ustvari nov žeton in ga zapiše v tabelo tokens. Uporabniku vrne žeton, da lahko ta kasneje z njim dokazuje svojo identiteto.

### GetSettings in UserLogout

Nato sem ustvaril datoteko, ki ob klicu sprejme žeton, uporabnika overi in mu nazaj vrne nastavitve njegovega računa. Vrne mu informacije o imenu in



Slika 5: Datotečna hierarhija API-ja

priimku, e-naslovu in naslovu zvočnega posnetka, shranjenega v bazi. Za tem sem ustvaril tudi UserLogout, ki ob klicu uporabnika prav tako overi in odjavi iz aplikacije, saj je gumb za odjavo postavljen na strani z nastavitvami.

## **UpdateSettings**

Zatem, ko sem v aplikaciji zgradil vmesnik za nastavitve na podlagi informacij pridobljenih iz GetSettings, sem sprogramiral datoteko UpdateSettings, ki sprejme nastavitve uporabnika in tiste, ki so se spremenile, posodobi v tabeli user. Preveriti mora tudi, če sta bila posodobljena slika ali avdio posnetek. Če je temu tako, se prejeta datoteka shrani na strežnik v mapo Assets v eno izmed dveh podmap Images ali Audio. Kot ime datoteke se nastavi ID uporabnika, kateremu slika oz. posnetek pripada.

## **GetInstruments in UpdateInstruments**

Te dve datoteki delujeta na isti princip kot GetSettings in UpdateSettings. GetSettings uporabnika overi in vrne podatke o inštrumentih. Uporabnik lahko v aplikaciji inštrumente dodaja ali odstranjuje. Ko s spreminjanjem konča, se pokliče datoteka UpdateInstruments, ki uporabnika prav tako overi in instrumente uporabnika posodobi v tabeli UserInstruments.

## **ReadUserInfo**

Datoteka ReadUserInfo na glavni ekran aplikacije vrne informacije o drugih glasbenikih. Deluje tako, da sprejme avtorizacijski žeton in števec. Uporabnika overi in mu vrne podatke o drugih glasbenikih, potrebne za prikaz na zaslonu, vključno z informacijo o všečkanju posameznih uporabnikov. Števec je potreben, saj API vrne le določeno število uporabnikov hkrati. Glede na števec datoteka ve, katere uporabnike iz vrste aplikacija želi. To izloči dolge čakalne čase v aplikaciji in omogoči nalaganje novih uporabnikov v ozadju. Število vrnjenih uporabnikov je določeno s spremenljivko, kot privzeto sem jo nastavil na 5.

## **UpdateLikes**

Ta datoteka se kliče na dveh mestih v aplikaciji in sicer na glavnem ekranu in na strani z všečkanimi uporabniki. Datoteka sprejme informacije o všečkanem uporabniku, uporabnika overi in nato posodobi tabelo UserLikes, glede na prejete informacije. Pomembna je spremenljivka status, ki označuje ali je bil uporabnik všečkan ali odvšečkan.

## **GetConversations in GetFavorites**

Datoteki sta potrebni za prikaz strani z všečkanimi uporabniki in pogovori. GetFavorites overi uporabnika in vrne uporabnike, ki jih je uporabnik predhodno všečkal. GetConversations deluje skoraj identično, uporabnika overi in vrne vse uporabnike, s katerimi ima uporabnik pogovor. Vrneta se tudi vsebina in čas zadnjega sporočila.

## **ReportUser**

Ta datoteka sprejme avtentikacijski žeton in ime osebe, ki je uporabnik prijavil. Uporabnik se overi in posodobi se tabela ReportedUser.

## **GetMessages**

Ta datoteka se pokliče ob odprtju ekrana za direktna sporočila. Tako kot drugje, se prvo overi uporabnika in iz tabele Messages vrne vsa sporočila pripadajočemu pogovoru.

## **NewConversation in NewMessage**

Te dve datoteki sta v aplikaciji dostopni iz ekrana direktnega pogovora. Funkcija NewConversation se pokliče v trenutku, ko se pošlje sporočilo uporabniku, s katerim pošiljatelj še nima pogovora. V tem primeru se posodobi tabela UserConversations in ustvari se nov pogovor. NewMessage se kliče vsakič, ko uporabnik pošlje sporočilo.

## Mobilna aplikacija

Pri načrtovanju uporabniškega vmesnika sem bil pozoren na osnovne principe heuristike. Osredotočil sem se na funkcionalnost in minimalističen design. Na začetku sem sicer ustvaril osnoven prototip vmesnika, vendar je izdelek končal zelo drugače, kot sem si ga sprva zamislil. Veliko sem kopiral tudi od že obstoječih aplikacij, predvsem aplikacije Tinder z zaslonom za odkrivanje.


Aplikacija sestoji iz petih različnih strani. Postavitve na posameznih straneh so ustvarjene kar se da intuitivno in preprosto za uporabo. Za prikaz vsebine sem poskušal izkoristiti celoten zaslon in zmanjšati količino neuporabnega prostora v aplikaciji. Osnovni tok sem sestavil tako, da sta dve glavni strani dostopni iz osnovnega zaslona. S tem sem dosegel manjše število potrebnih klikov za želen rezultat uporabnika.

Za barvno paleto sem si izbral preprosto črno-belo z odtenki sive za sence gumbov in podobnih elementov. Eksperimentiral sem tudi z drugimi barvami, vendar sem ostal pri tej kombinaciji, saj nase ne veže preveč pozornosti in tako poskrbi, da uporabnik večjo pozornost nameni drugim, bolj pomembnim delom aplikacije.

Pisava izbire je bila na koncu pisava Montserrat iz spletne strani Google Fonts. Je minimalistična Sans Serif pisava, katero sem uporabil čez obseg celotne aplikacije. Izbral sem si jo, ker mi je bil všeč njen design, pisava pa podpira tudi obseg različnih debelin in vse potrebne znake ASCII tabele.

### **Prijava in registracija**

Stran za prijavo in registracijo je bila tista, s katero sem začel gradnjo vmesnika. Je stran, ki se pojavi uporabniku, ko prvič odpre aplikacijo, ali ko se odjavi in se želi ponovno prijaviti. Sestavljajo jo logotip na vrhu, polja za vnos podatkov in na dnu gumb za prijavo oz. registracijo. Preklapljanje med opcijo prijave ali registracije je omogočeno s klikom na tekstovni gumb med polji za vnos in glavnim gumbom. Stran sem opremil tudi z avtomatskim preverjanjem vnosov teksta in javljanem statusa o napačnem geslu ali e-poštnem naslovu.

— —  — —  
**BandWith**


Email

Password

New to BandWith? [Sign Up!](#)

**LOG IN**

*Slika 6: Ekran za prijavo*

— —  — —  
**BandWith**

Name

Surname

Email

Password

Already have an account? [Log In!](#)

**REGISTER**

*Slika 7: Ekran za registracijo*

**Password**

Invalid email or password.

*Slika 8: Odziv na napačno geslo/email*

## Nastavitve

Naslednji korak pri izdelavi je bil oddelek z nastavitvami uporabnika. Ta sestoji iz strani za osnovne nastavitve in strani za nastavitve inštrumentov, ki je dostopna z gumbom "Instruments".

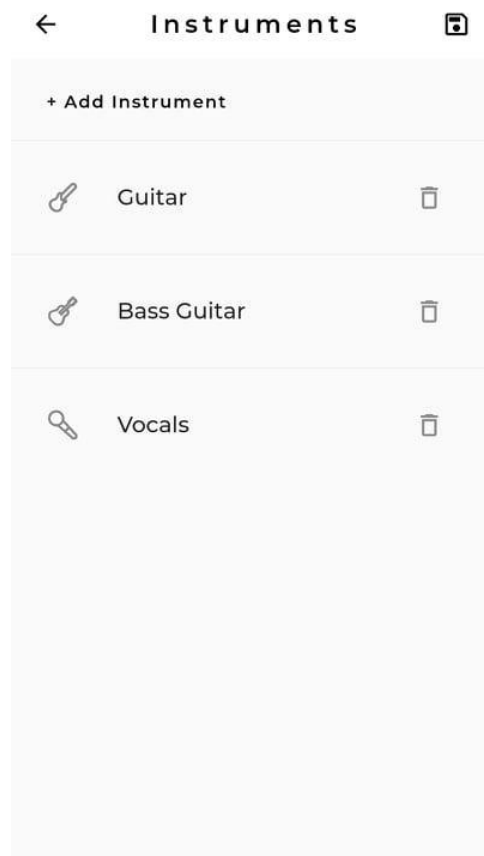
Osnovna stran pokliče datoteko `GetSettings` na API in iz nje pridobi podatke o že obstoječih nastavitvah uporabnika. Dokler se ta klic izvaja in v aplikaciji še ni nastavitvev uporabnika, se v telesu strani prikazuje ikona za nalaganje. Na vrhu osnovne strani je navigacijska vrstica, ki vsebuje gumb "shrani", za shranjevanje sprememb nastavitvev. Telo strani vsebuje osnovne podatke o uporabniku kot so slika profila, ime in e-poštni naslov uporabnika. Pod sliko profila je tudi gumb, s katerim lahko določimo novo sliko. Klik na ta gumb nas popelje v Androidov raziskovalec datotek, kjer izberemo novo sliko. Pod osnovnimi informacijami se nahaja ploščica "SoundID", na kateri lahko ob kliku na gumb "Play" poslušamo naš zvočni posnetek ali pa ga zamenjamo. Ob kliku na ploščico se nam prav tako odpre raziskovalec datotek, kjer izberemo nov zvočni posnetek. Sledi ploščica "Instruments". Klik na njo nas popelje na stran za urejanje inštrumentov. Na dnu pa je še ploščica za odjavo, ki nas odjavi iz aplikacije in vrne na zaslon za prijavo.

Stran za urejanje inštrumentov se prav tako začne z navigacijsko vrstico, podobno tisti na osnovni strani nastavitvev. Dokler se prenašajo podatki o inštrumentih, se v telesu strani prav tako prikazuje ikona za nalaganje. Ko se podatki naložijo, se prikaže prava stran, ki se začne z gumbom "Add Instruments". Klik nanj odpre seznam na dnu ekrana, na katerem so prikazani vsi inštrumenti, ki jih še nimamo dodanih. Klik na enega od inštrumentov ga doda na seznam. Sledi seznam naših inštrumentov, ki je sestavljen iz seznamskih ploščic. Vsaka ploščica se začne z ikono inštrumenta, sledi ime inštrumenta, na koncu pa je še gumb s katerim lahko inštrument odstranimo iz seznama. Če uporabnik nima izbranega nobenega inštrumenta, se na mestu seznama prikaže tekst s primerno vsebino.

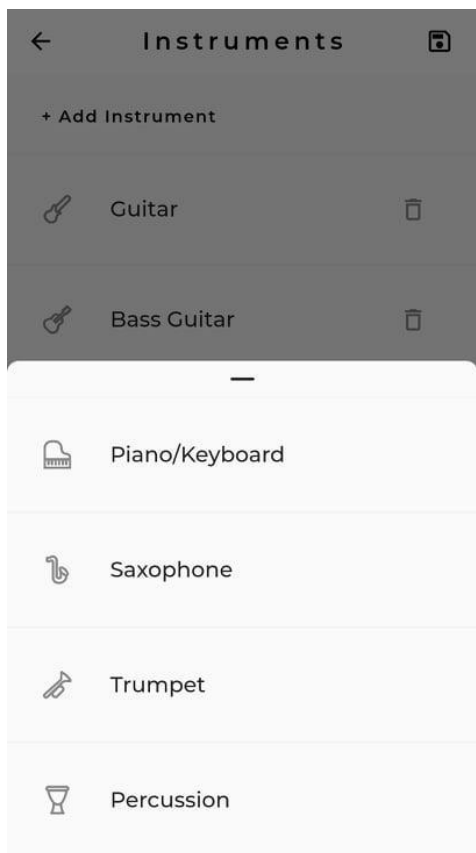




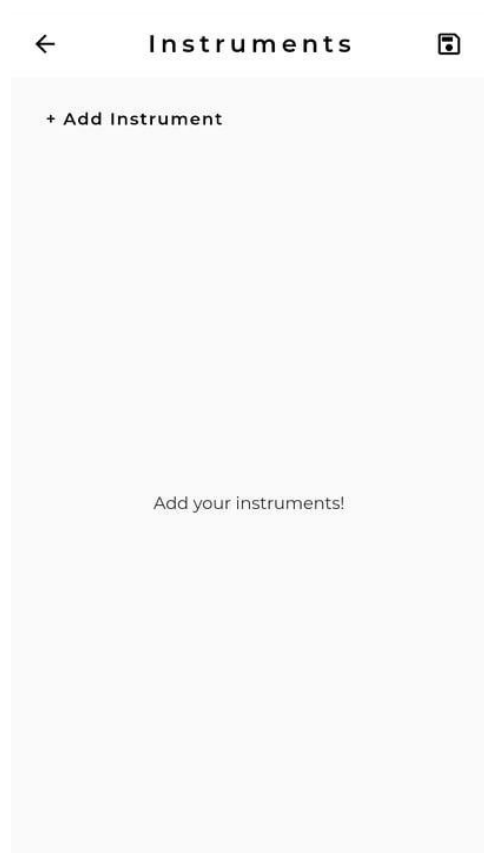
Slika 9: Osnovne nastavitve



Slika 10: Nastavitve inštrumentov



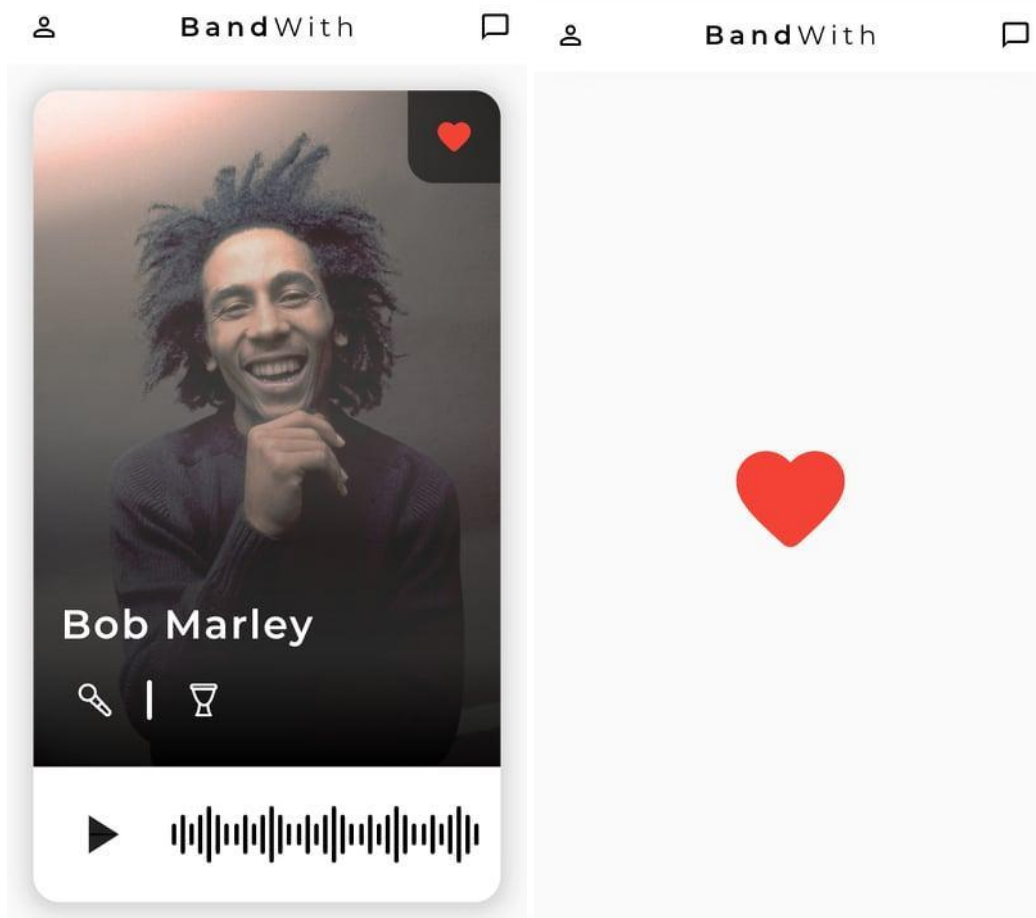
Slika 11: Seznam za dodajanje



Slika 12: Seznam brez inštrumentov

## Stran za raziskovanje

Stran za raziskovanje je osnovna stran, ki se odpre, ko se uporabnik prijavi v aplikacijo. Na vrhu zaslona je ponovno navigacijska vrstica. Na levi strani vsebuje ikono "Profile", ki nas popelje na nastavitve profila, na desni pa ikono "Chat", ki nas postavi na ekran s pogovori in všečkanimi uporabniki. Dokler se iz strežnika prenašajo podatki o uporabnikih, je v telesu zaslona prikazana ikona nalaganja, nato pa se z uporabniki napolni seznam, po katerem se premikamo s potezami vlečenja levo in desno. Seznam je sestavljen iz kartic z informacijami o uporabniku. V ozadju je prikazana slika, na njej pa se izpiše ime uporabnika in z ikonami se pod imenom prikažejo njegovi inštrumenti. Desno zgoraj je gumb za všečkanje uporabnika, na dnu kartice pa je tudi gumb za predvajanje zvočnega posnetka trenutnega uporabnika. Všeček lahko izvedemo tudi tako, da kartico povlečemo navzgor, nakar se čez cel zaslon prikaže ikona, ki naznani, da smo uporabnika všečkali. Seznam uporabnikov se v ozadju samodejno nalaga. Ko se približamo koncu seznama, se na API pokliče datoteka ReadUserInfo, ki nam vrne nove uporabnike in z njimi napolni



seznam.

Slika 13: Kartica uporabnika

Slika 14: Prikaz po všečkanju  
s swipe gesto

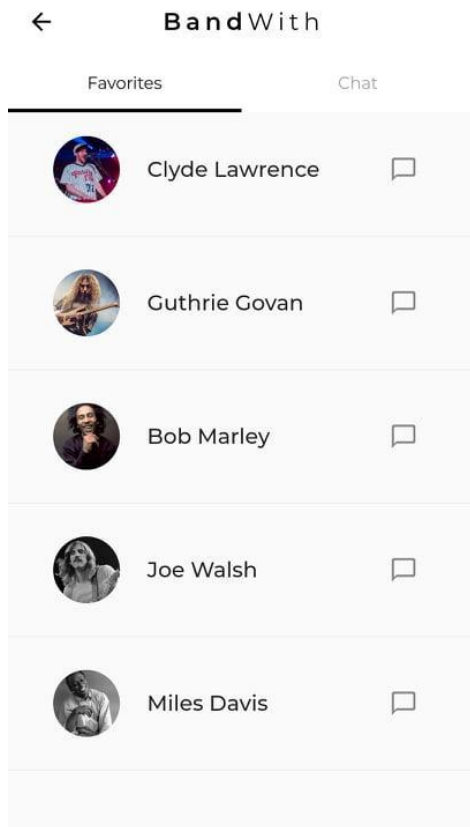
## Stran z všečki in pogovori

Do strani z všečki in pogovori dostopamo iz strani za raziskovanje z ikono "Chat". Na vrhu strani je navigacijska vrstica, tokrat le z gumbom "Back" za vrnitev na domači zaslon aplikacije. Pod njim imamo vrstico z dvema zavihkoma. To sta zavihka "Favorites" in "Chat", med katerima lahko navigiramo tudi s potegi v levo ali desno. V telesih obeh zavihkov se vsebina ponovno izrisuje glede na stanje pridobivanja podatkov iz strežnika.

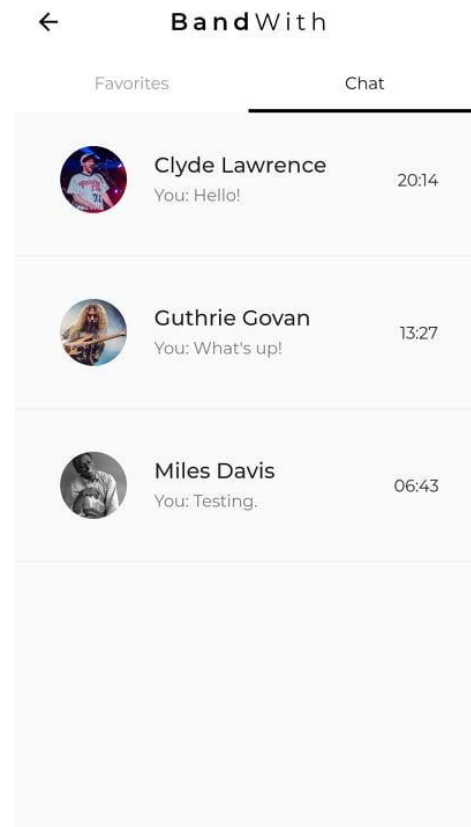
Ko so podatki preneseni, se pod zavihkom "Favorites" prikaže seznam priljubljenih glasbenikov. Seznam je sestavljen iz ploščic. Na levi strani najdemo sliko uporabnika, sledi ime in na desni imamo gumb "Chat" za ustvarjanje novega pogovora z uporabnikom, ali za vstop v že obstoječ pogovor. Ob dolgem pritisku na ploščico se nam na dnu zaslona prikaže seznam. Ta seznam je sestavljen iz imena uporabnika, ki smo ga izbrali z dolgim pritiskom, ponudijo pa se še opcije "Unlike" in "Report". Gumb "Unlike" je samoumeven, uporabnika odstrani iz seznama priljubljenih, to se posodobi tudi na zaslonu za raziskovanje, kjer uporabnik po tem ni več oznanjen kot priljubljen. Gumb "Report" je namenjen prijavljanju uporabnikov, ki objavljajo neprimerno vsebino. Ob pritisku na gumb se na zaslonu pokaže potrditveni dialog in ob potrditvi prijave zlorabe se na API pokliče datoteka ReportUser, ki prijavo zabeleži. Do vseh prijav lahko dostopamo preko administratorske konzole, kjer se izpišejo prijavljeni uporabniki sortirani po številu prijav.

Pod zavihkom "Chat" se po uspešnem prenosu podatkov na zaslon prikaže seznam obstoječih pogovorov z drugimi uporabniki. Seznam je prav tako sestavljen iz ploščic. Na levi strani ploščice je profilna slika uporabnika, na sredini najdemo ime uporabnika, ter ime pošiljatelja in vsebino zadnjega sporočila pogovora, na desni strani pa je prikazan čas pošiljanja zadnjega sporočila. Sporočila so sortirana po času pošiljanja od najnovejšega do najstarejšega. Klik na ploščico pogovora nas popelje na stran za klepet z izbranim uporabnikom.

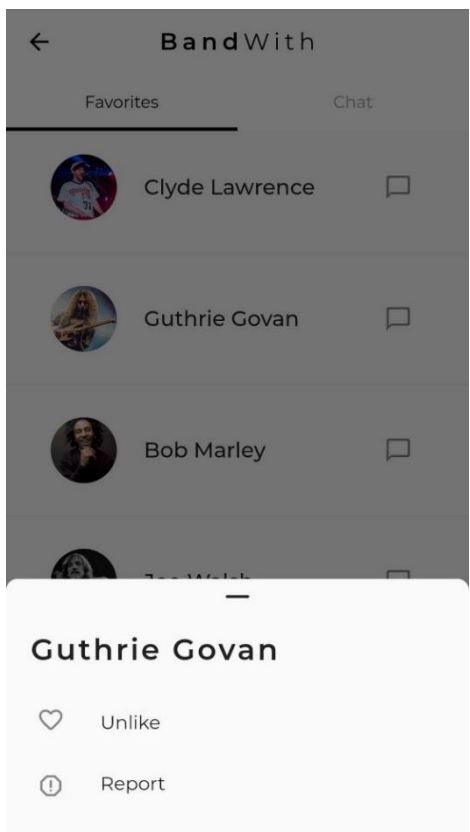
Obe strani imata vgrajeno tudi funkcijo potega navzdol za osvežitev seznama priljubljenih ali pogovorov.



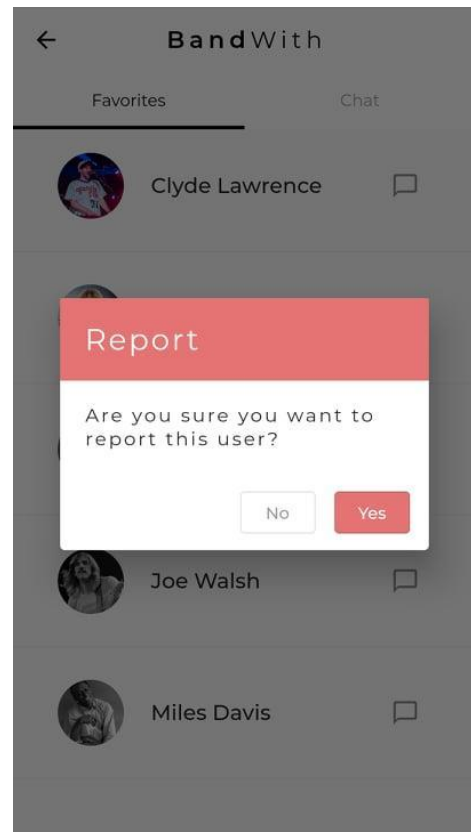
Slika 15: Stran s priljubljenimi



Slika 16: Stran s pogovori



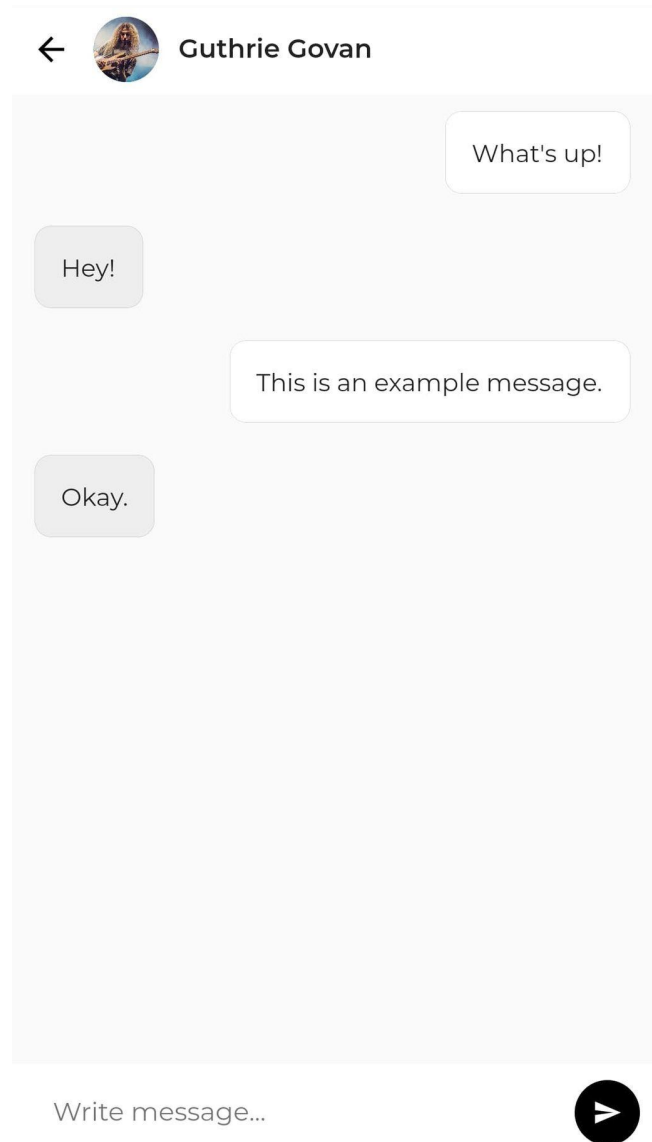
Slika 17: Dolg pritisk na ploščico



Slika 18: Potrditveni dialog

## Stran za klepet

Stran za klepet je dostopna s klikom na eno izmed ploščic na ekranu s pogovori. Na vrhu najdemo vrstico z gumbom "Back" in z osnovnimi informacijami uporabnika, pod katere spada slika ter ime in priimek. V telesu se ponovno izrisuje ikona nalaganja, dokler se iz strežnika ne preberejo informacije o pogovoru. Strežniška datoteka `GetMessages` nam vrne vsa sporočila z določeno osebo, ta se izrišejo v obliki seznama, ki ima elemente postavljene na levo ali desno stran ekrana, glede na to, kdo je pošiljatelj sporočila. Pošiljatelj je razviden tudi iz barve sporočila. Na dnu ekrana najdemo besedilno polje in gumb "Send". Ob kliku na besedilno polje se na zaslonu prikaže tipkovnica in vanj lahko vtipkamo sporočilo, ki ga želimo poslati. Ko pošljemo novo sporočilo, se seznam posodobi in doda novo sporočilo. Ob pritisku na gumb send se pokliče datoteka `NewMessage`, aplikacija pa hkrati preverja, ali je to prvo sporočilo v pogovoru s to osebo. Če je temu tako, pokliče tudi datoteko `NewConversation`, kar v podatkovni bazi ustvari nov pogovor.



*Slika 19: Stran za klepet*

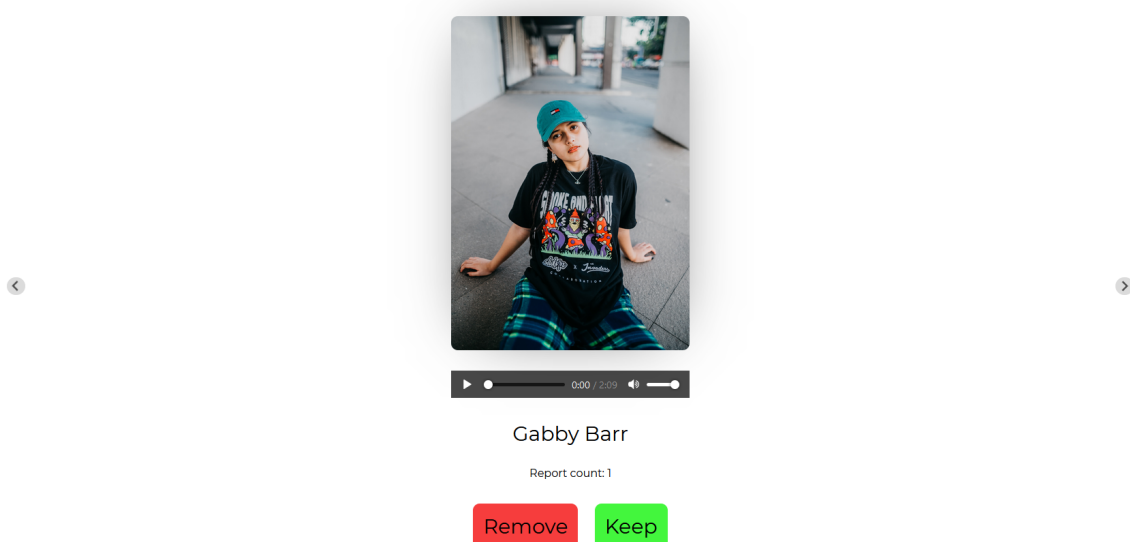
## Administratorska konzola

Administratorska konzola je ustvarjena za upravljanje in nadziranje uporabnikov, prijavljenih za objavljanje neprimerne vsebine in je namenjena moderatorjem platforme.

Dostopna je na posebnem url naslovu, stran pa je zaščitena z uporabniškim imenom in geslom, da lahko do nje dostopajo samo pooblašcene osebe.

Sestavljena je iz seznam vseh prijavljenih uporabnikov. Po seznamu se lahko premikamo z miško, s pritiskom na gumba levo ali desno, ali pa s smernimi tipkami na tipkovnici. Elementi seznama so sestavljeni iz slike uporabnika, pod njo najdemo zvočni posnetek, ime, priimek, ter število, kolikokrat je bil uporabnik prijavljen za neprimerno vsebino. Na dnu sta gumba "Remove" in "Keep". Z njima se moderator odloča, ali bo uporabnika obdržal na platformi ali ga je potrebno odstraniti. Seznam je sortiran po številu prijav, tako moderator najprej obravnava najbolj problematične uporabnike.

Administratorska konzola in funkcija prijavljanja sta ključni za delovanje platforme, saj bi drugače računalniški programi zlahka zrušili stran, ali pa bi prišlo do nenadzorovanega objavljanja neprimernih vsebin.



*Slika 20: Administratorska konzola*

# Zaključek

Izdelek je presegel moja pričakovanja, saj sem šel v projekt brez pravega znanja in brez konkretnega načrta. Ta projekt je brez dvoma moja najboljša izkušnja na Vegovi, saj sem se iz nje naučil ogromno. Spoznal sem se z načrtovanjem napredne podatkovne baze, naučil sem se sprogramirati strežnik in zgraditi znam odlično mobilno aplikacijo. Spoznal sem se tudi z osnovami designa in načrtovanja uporabniških vmesnikov, kar me je pritegnilo bolj, kot sem pričakoval.

Aplikacijo imam namen v prihodnjih mesecih še izboljšati in dodati funkcionalnost.

Prav tako želim v prihodnosti nadgrajevati svoje znanje. Ta projekt mi je dal motivacijo za delo in učenje na tem področju in prepričan sem, da se bom s tem ukvarjal še dolgo časa.

# Viri

- <https://www.udemy.com/course/learn-flutter-dart-to-build-ios-android-apps/learn> (1. 4. 2021)
- <https://stackoverflow.com> (1. 4. 2021)
- [https://en.wikipedia.org/wiki/Flutter\\_\(software\)](https://en.wikipedia.org/wiki/Flutter_(software)) (1. 4. 2021)
- [https://en.wikipedia.org/wiki/Dart\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Dart_(programming_language)) (1. 4. 2021)
- <https://sl.wikipedia.org/wiki/PHP> (1. 4. 2021)
- <https://sl.wikipedia.org/wiki/SQL> (1. 4. 2021)



## Izjava o avtorstvu

Izjavljam, da je seminarska naloga Socialno omrežje - BandWith v celoti moje avtorsko delo, ki sem ga izdelal samostojno s pomočjo navedene literature in pod vodstvom mentorja.

Logatec, 3. 4. 2021

Vane Skubic