

Nejc Podvratnik

**RAZVOJ PLATFORME ZA PODPORO SPLETNIM SKUPNOSTIM**

Diplomsko delo

Maribor, avgust 2022



Nejc Podvratnik

**RAZVOJ PLATFORME ZA PODPORO SPLETNIM SKUPNOSTIM**

Diplomsko delo

Maribor, avgust 2022

**RAZVOJ PLATFORME ZA PODPORO SPLETNIM SKUPNOSTIM**

Diplomsko delo

|  |  |
| --- | --- |
| Študent: | Nejc Podvratnik |
| Študijski program: | Univerzitetni študijski program  Računalništvo in informacijske tehnologije |
| Mentor: | izr. prof. dr. Matej Črepinšek |
| Lektorica: | Manja Belina, prof. mag. slov. in prof. mag. špan. |
|  |  |
|  | About CC Licenses - Creative Commons |

**Zahvala**

Zahvaljujem se izr. prof. dr. Mateju Črepinšku za pomoč

in usmerjanje pri pisanju diplomskega dela.

Posebna zahvala tudi družini, ki mi je

omogočila študij in mi stala ob strani.

**Razvoj platforme za podporo spletnim skupnostim**

**Ključne besede:** platforma, hibridna aplikacija, igrifikacija

**UDK:** 004.777(043.2)

**Povzetek**

*Namen diplomskega dela je izdelava prototipa platforme, ki omogoča odgovarjanje in postavljanje vprašanj prijavljenim uporabnikom. Opisan je postopek, kako motiviramo uporabnika s pomočjo igrifikacije. Našteti so že obstoječi sistemi za podporo spletnim skupnostim. Vsebuje kratek opis tehnologij in orodij, s katerimi je platforma izdelana. Predstavljena in razložena sta izgled in funkcionalnost čelnega dela kot tudi zalednega dela in podatkovne baze. V zadnjem delu je prikazan postopek testiranja aplikacije.*

**Development of a platform to support online communities**

**Ključne besede:** platform, hybrid application, gamification

**UDK:** 004.777(043.2)

**Abstract**

*The aim of the thesis is to create a prototype of the platform, which enables registered users participation in asking and answering questions. The process of how we motivate users with the help of gamification is described. Already existing systems for supporting online communities are listed. It contains a brief description of the technologies and tools with which the platform is built. The appearance and functionality of the frontend part as well as the backend part and the database are presented and explained. The last part shows the application testing process.*

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Kazalo vsebine

[Kazalo slik X](#_Toc112072022)

[Uporabljeni simboli in kratice XI](#_Toc112072023)

[1 Uvod 1](#_Toc112072024)

[2 Sistemi za podporo spletnim skupnostim 2](#_Toc112072025)

[2.1 Igrifikacija 2](#_Toc112072026)

[2.2 Obstoječi sistemi 4](#_Toc112072027)

[2.2.1 Stack Overflow 4](#_Toc112072028)

[2.2.2 Healthtap 5](#_Toc112072029)

[2.2.3 Quora 6](#_Toc112072030)

[2.2.4 Reddit 7](#_Toc112072031)

[3 Uporabljene tehnologije 8](#_Toc112072032)

[3.1 Čelni del 8](#_Toc112072033)

[3.2 Zaledni del 9](#_Toc112072034)

[4 Predstavitev PRototipne Platforme 10](#_Toc112072035)

[4.1 Elementi 10](#_Toc112072036)

[4.1.1 Uporabnik 10](#_Toc112072037)

[4.1.2 Vprašanje 10](#_Toc112072038)

[4.1.3 Odgovor 11](#_Toc112072039)

[4.1.4 Glas 12](#_Toc112072040)

[4.2 Aplikacija 12](#_Toc112072041)

[5 Implementacija 19](#_Toc112072042)

[5.1 Zaledni del 19](#_Toc112072043)

[5.1.1 Overjanje 20](#_Toc112072044)

[5.2 Čelni del 21](#_Toc112072045)

[5.2.1 Ovoj gumba 22](#_Toc112072046)

[5.2.2 Iskalna vrstica 23](#_Toc112072047)

[5.2.3 Samodejna prijava 24](#_Toc112072048)

[6 Testiranje 25](#_Toc112072049)

[7 SKLEP 27](#_Toc112072050)

[Viri in literatura 29](#_Toc112072051)

Kazalo slik

[Slika 2.1: Elementi igrifikacije v spletni strani Stack Overflow 3](#_Toc112072052)

[Slika 2.2: Primer vprašanja v Stack overflowu 5](#_Toc112072053)

[Slika 2.3: Primer vprašanja v Healthtapu 6](#_Toc112072054)

[Slika 3.1: Demonstracijska Flutter aplikacija 8](#_Toc112072055)

[Slika 4.1: Primer vprašanja 11](#_Toc112072056)

[Slika 4.2: Začetna stran 12](#_Toc112072057)

[Slika 4.3: Prijavna stran 13](#_Toc112072058)

[Slika 4.4: Registracijska stran 14](#_Toc112072059)

[Slika 4.5: Glavna stran 15](#_Toc112072060)

[Slika 4.6: Primer iskalne vrstice 15](#_Toc112072061)

[Slika 4.7: Profilna stran 16](#_Toc112072062)

[Slika 4.8: Stran za ustvarjanje vprašanja 16](#_Toc112072063)

[Slika 4.9: Stran uporabnikovih vprašanj 17](#_Toc112072064)

[Slika 4.10: Stran vprašanja 18](#_Toc112072065)

[Slika 4.11: Besedilno polje za odgovor 18](#_Toc112072066)

[Slika 4.12: Pojavno okno za spreminjanje odgovora 18](#_Toc112072067)

[Slika 5.1: Mape zalednega dela 20](#_Toc112072068)

[Slika 5.2: Ustvarjanje žetona 20](#_Toc112072069)

[Slika 5.3: Primer uporabe strani Quicktype 21](#_Toc112072070)

[Slika 5.4: Mape čelnega dela 22](#_Toc112072071)

[Slika 5.5: Možni postavitvi gumbov 23](#_Toc112072072)

[Slika 5.6: Koda za gradnik ButtonWrapper 23](#_Toc112072073)

[Slika 5.7: Koda za prikaz oznak v iskalni vrstici 24](#_Toc112072074)

[Slika 6.1: Primer uporabe postmana 25](#_Toc112072075)

[Slika 6.2: Primer testov za zahtevo za profil 26](#_Toc112072076)

Uporabljeni simboli in kratice

AES – Napredni standard šifriranja (angl. Advanced Encryption Standard)

BSON – Binarna javascript objektna notacija (angl. Binary javascript object notation)

HTTP – Protokol za prenašanje hiperteksta (angl. Hypertext transfer protocol)

ID – Identifikacijska številka (angl. Identification card)

JWT – Spletni žetoni za javascript objektno notacijo (angl. Javascript object notation web tokens)

MERN – MongoDB, express, react in node.js

MVC – Model-pogled-kontroler (angl. Model–view–controller)

noSQL – Ne samo strukturirani povpraševalni jezik (angl. Not only structured query language)

Q&A – Vprašanja in odgovori (angl. Question and answers)

REST – Predstavitveni prenos stanja (angl. Representational state transfer)

UI – Uporabniški vmesnik (angl. user interface)

# Uvod

Spletne platforme so v zadnjih letih dosegle velik uspeh na področju komunikacije, nakupovanja, storitev, marketinga, razvoja in zabave. V tem diplomskem delu se bomo osredotočili na razvoj spletne platforme, ki nudi podporo, sodelovanje in pomoč uporabnikom s strani drugih uporabnikov. Pravimo jim platforme Q&A (angl. Question and answers). Na vprašanja uporabnikov odgovarjajo drugi uporabniki. Teme vprašanj se navezujejo na znanost (Briliant.org), zdravje (HealthTap in ShareCare), trgovanje (LinkedIn Answers), programiranje (Stack Overflow) in drugo (Reddit, Quora in Yahoo! Answers) [1]. Platforme Q&A motivirajo uporabnike k odgovarjanju z igrifikacijo, ki tehnike motiviranja iz tradicionalnih in družabnih iger aplicira v okolje, ki tradicionalno ni namenjeno igri [2].

Namen diplomskega dela je izdelava prototipa platforme, ki bo vsebovala zaledni in čelni del. Omogočala bo funkcionalnost za odgovarjanje in postavljanje vprašanj prijavljenim uporabnikom. Raziskovali bomo, kako motivirati uporabnike z elementi igrifikacije, kar bomo izvedli v naši platformi. Seveda bomo tudi preučili tehnologije, potrebne za implementacijo takšne platforme.

V drugem poglavju so na kratko predstavljeni že obstoječi sistemi za podporo spletnim skupnostim in igrifikacija. Temu sledi poglavje, v katerem so opisane uporabljene tehnologije, razdeljene na dva dela (čelni in zaledni del). Nato je prikazana prijava, registracija, postavljanje vprašanj, iskanje in filtriranje vprašanj, pisanje odgovorov in ocenjevanje v izdelanem prototipu. V naslednjem poglavju je podrobneje razložena implementacija teh značilnosti. Pred sklepom je še poglavje, namenjeno testiranju prototipa.

# Sistemi za podporo spletnim skupnostim

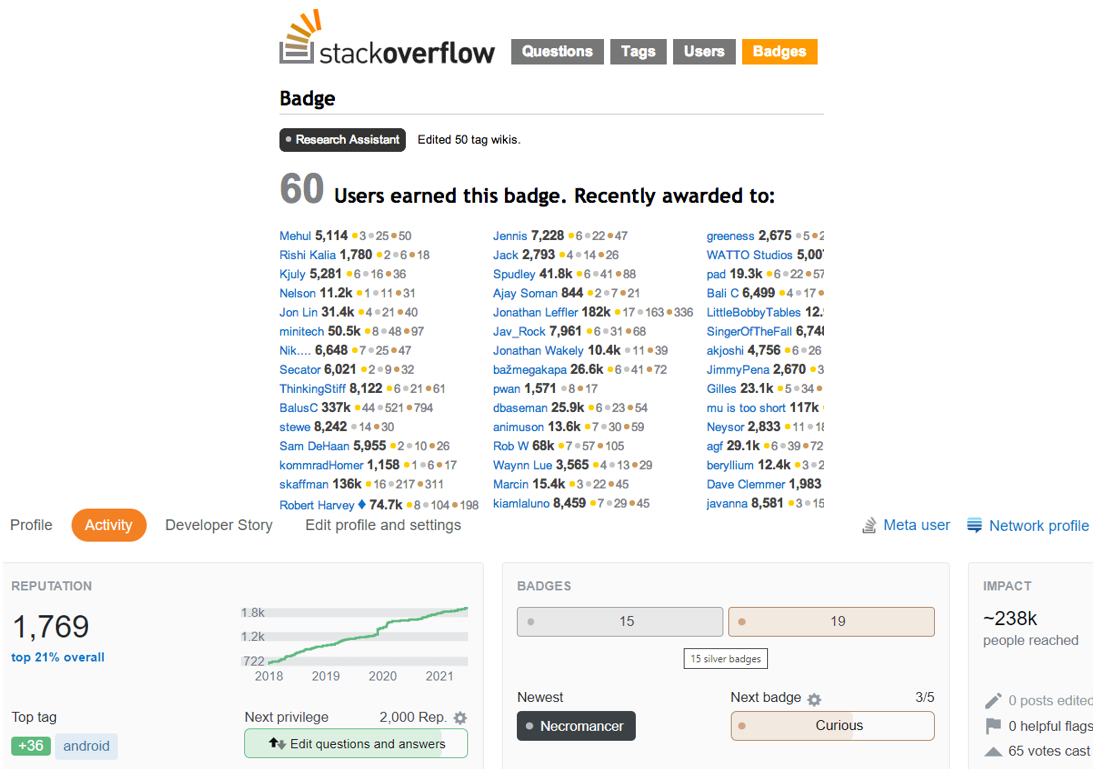
Skupnostne platforme so virtualni prostor, preko katerega se lahko družijo ljudje, ki si delijo iste cilje in interese. Najboljše sodobne skupnostne platforme ponujajo članom različne načine povezovanja med seboj. To so neposredni ali skupinski pogovori, ankete, dogodki in spletni tečaji. V tem diplomskem delu se bomo osredotočili na sisteme za podporo spletnim skupnostim. V njem uporabniki pridobivajo neznane informacije in posredujejo znano informacijo s sistemom vprašanj in odgovorov.

Kako motivirati uporabnika, da sodeluje v tem sistemu? Priljubljen odgovor je uporaba igrifikacije. Forbes je leta 2013 ocenil, da je več kot 70% podjetij na lestvici Global 2000 v anketi izjavilo, da nameravajo uporabljati igrifikacijo za namene trženja in ohranjanja strank. [3]

## Igrifikacija

Igrifikacija je uporaba elementov in principov računalniških iger v ne-igralskem kontekstu. Opredelimo ga tudi kot niz dejavnosti in procesov za reševanje problemov z uporabo karakteristik igralskih elementov. Klasični elementi iger so: točke, značke in lestvice. Točke predstavljajo napredek v določeni aktivnosti v numerični obliki. Značke prikazujejo dosežek za opravljeno delo. Lestvice pa se uporabljajo za predstavitev igralcev od najboljšega do najslabšega v določeni dejavnosti, kar spodbuja konkurenčnost [3].

Platforma Stack overflow uporablja točke in značke (slika 2.1). Točke dobiš kadar uporabniki pozitivno glasujejo na tvoja vprašanja in odgovore, značke pa kadar si dosegel nek dosežek. Obstajajo trije nivoji značk (bronze, silver in gold), ki se razlikujejo po težavnosti dosežka. Svoje značke lahko tudi primerjaš z drugimi.



Slika 2.1: Elementi igrifikacije v spletni strani Stack Overflow

Obstaja več razlogov zakaj oseba želi doseči čim več točk, pridobiti značke ali biti čim višje na lestvici. Morda si s tem želi izboljšati svojo samopodobo ali strokovni ali socialni status. Pogosto storitev omogoča koriščenje pridobljenih točk in ponuja ugodnosti ljudem z določeno značko ali nazivom. [6]

Čeprav je digitalne storitve lažje igrifikirat, je igrifikacija tudi prisotna v resničnem življenju. Trgovine uporabljajo kartice zvestobe (na primer spar plus kartica), s katerimi lahko uporabiš dobroimetje za izkoriščanje ugodnosti. Skavti pridobijo značke za določene aktivnosti. Uporabljajo jo v delovnem mestu za sledenje napredka, postavljanjem ciljev in spodbujanju prijateljskega tekmovanja [4] [7].

Danes je igrifikacija še posebej prisotna zaradi naraščanja igralniške industrije. Ocenjeno je bilo, da je leta 2020 2.69 bilijon ljudi igralo video igre [5]. Zato uporabljajo seznanjenost do igralskih elementov v njihovo prid.

Izbira pravih elementov je lahko izziv. Pomembno je, da elementi igre spodbujajo želeno vedenje, ker se bodo na to udeleženci osredotočili. Če je možnost napredovanja pomanjkljiva in če ne moreš slediti svojega napredka in napredka drugih, bo uporabnik manj motiviran [6] [7].

Igrifikacija se uporablja v več panogah. V marketingu je popularno oglaševanje v videoigrah. Nedavni primer je bilo oglaševanje Tesla avtomobilov v popularni igri Player Unknown's Battlegrounds, kjer so lahko igralci vozili tesla avtomobile v virtualnem svetu. En izmed prvih primerov se je zgodil leta 2006, kjer je popularna igra Counter-Strike omogočila oglaševalcem promoviranje filmskih plakatov na stenah igre. V zdravju obstajajo aplikacije, ki sledijo tvojemu napredku pri telovadbi (kot je število porabljenih kalorij, število sklec, razdalja teka, povprečna hitrost, teža uteži itd.) in glede na tvojo uspešnost podarijo značke in nagrade. Omogoča primerjanje starih rezultatov z novimi in primerjanje z drugimi uporabniki, kar spodbuja tekmovalnost. Aplikacija Nike+Run je ena izmed takšnih aplikacij, ki se osredotoča na tek. V izobraževanju uporabljajo video igre kot pripomoček za učenje. Podjetje Mojang je ustvarilo različico igre Minecraft, ki je zelo popularna med otroki. Imenuje se Minecraft: Education edition, ki uči otroke teme iz naravoslovja in kemije. Aplikacija Kahoot spodbuja tekmovalnost med učenci s pomočjo kvizov. Pri spletnih straneh uporabijo igrifikacijo, da spodbudijo želeno obnašanje obiskovalcev. Na straneh kot je Reddit je željeno druženje in vključenost. Zato uporabniki dobijo točke in značke, ki so prikazane na njihovem profilu in lestvici [3].

## Obstoječi sistemi

Preden smo se lotili izdelave lastne platforme nas je zanimalo kako delujejo že obstoječi sistemi in kako so implementirali igrifikacijo. Ugotovili smo da so igrifikacijo implementirali na podoben način, čeprav se je tema vprašanj in odgovorov med njimi razlikuje. V zgled nam je bila platforma Stack Overflow, ker se ukvarja z isto temo vprašanj s katero se bo ukvarjal naš prototip.

### Stack Overflow

Stack Overflow je ena izmed najpopularnejših spletnih mest za vprašanja in odgovore, ki se nanašajo na programiranje. Zato so obiskovalci spletne strani večinoma programerji, navdušenci in študentje. Ustvarila sta jo Jeff Atwood in Joel Spolsky leta 2008 [8]. Ker je programiranje zelo široka panoga, omogoča Stack overflow kategoriziranje vprašanj s pomočjo oznak.

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 2.2: Primer vprašanja v Stack overflowu

Na sliki 2.2 ima vprašanje 5 oznak (c++, c, operators, code-formatting in standards-compliance), s katerimi lahko drugi lažje najdejo vprašanje. Bila je glasovana 9787-krat in dobila skoraj en milijon ogledov. Ima 28 odgovorov. Zelena kljukica nam pove, da je vsaj en odgovor bil sprejet. Uporabniki lahko iščejo sporočilo s pomočjo besedila, oznak in imena avtorja. Neprijavljen uporabnik ima dostop do vprašanj in odgovorov ampak jih ne more objaviti. Prijavljeni uporabniki pridobijo točke ugleda (angl. reputation), če so njihove objave bile pozitivno ocenjene [8]. Zraven statusa imaš z visokim ugledom privilegije, ki jih uporabniki z manj ugleda nimajo. Privilegije so lahko majhne, kot je možnost pozitivnega glasovanja pri 15 točkah ugleda. Lahko so pa zelo močne, kot je spreminjanje tujih objav pri 2000 točkah in dostop do analitik spletne strani pri 25000 točkah [9].

### Healthtap

HealthTap je podjetje, ki zagotavlja zdravstvo prek spleta. Ustanovljeno je bilo leta 2000. Posebnost foruma je, da lahko na vprašanja (slika 2.3) odgovarjajo samo zdravniki. Zraven imena je še podana vrsta doktorja in leta izkušnje. Neprijavljeni uporabniki lahko samo berejo vprašanja, za odgovore pa je potrebna prijava. Zdravniki lahko označijo da se z odgovorom strinjajo, navadni uporabniki pa se lahko odgovoru zahvalijo [10].

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 2.3: Primer vprašanja v Healthtapu

### Quora

Quora je spletna stran z vprašanji in odgovori. Ustanovljena je bila leta 2009. Soustanovila sta jo nekdanja Facebook uslužbenca Adam D'Angelo in Charlie Cheever. Uporabniki lahko sodelujejo z urejanjem vprašanj in komentiranjem odgovorov, ki so jih objavili drugi uporabniki [11]. Teme se ne nanašajo na specifično panogo kot Stack overflow in HealthTap. Vprašanja se lahko nanašajo na karkoli, kar da uporabnikom več svobode. Ima podobne elemente kot stack overflow ampak so manj zapleteni in preprostejši. Za kategoriziranje vprašanj uporablja zgodbe (angl. stories). Vsako vprašanje more spadati v točno eno zgodbo. Quora je imela v preteklosti sistem, kjer si pridobil kredite, če je tvoja objava bila pozitivno glasovana. S krediti si lahko promoviral svoje vprašanje. Veš plačanih kreditov je pomenilo več doseženih ljudi [12].

### Reddit

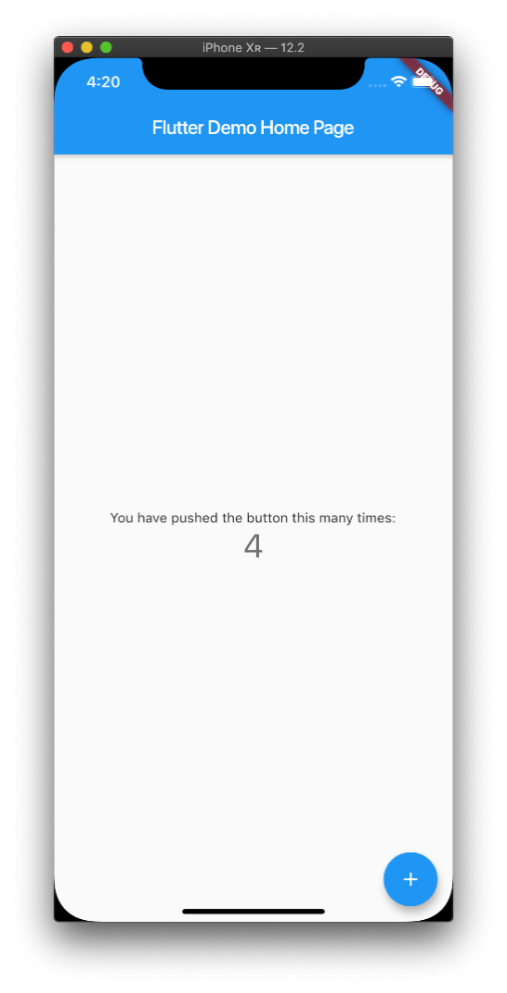
Reddit je zelo podobna Quori. Ampak Reddit se ne uporablja samo za vprašanja in odgovore. V bistvu se lahko uporablja za vse. Za diskusije, pripovedovanje zgodb, šale in deljenje mnenja. Ustanovljena je bila leta 2005. Za kategoriziranje uporabljajo »subreddite« ali podstrani, kjer se objave nanašajo samo na določeno temo. Prijavljeni uporabniki lahko objavljajo in glasujejo. Ima dve valuti. Reddit karma je valuta, ki jo oseba dobi, kadar so njegove objave pozitivno glasovane. Uporabnikom pove koliko je oseba prispevala na strani, nima pa nobene funkcionalnosti [13]. Druga valuta so Reddit Coinsi, katere more uporabnik kupiti. Uporabljajo se za nakup značk, ki jih uporabnik lahko nagradi določeni objavi. Obstajajo tri značke (silver, gold in platinum), ki se razlikujejo po ceni in ugodnosti, ki jih avtor dobi [14].

# Uporabljene tehnologije

Ker smo razvili čelni in zaledni del prototipa smo uporabili celostno arhitekturo (angl. full stack) . Je sklad rešitev oziroma tehnologij, s katerim lahko razvijemo spletno stran ali aplikacijo v celoti. Pri izboru smo dali prednost tistim tehnologijam, ki so primerne za naš izdelek, so nam dobro znane, so dokaj preproste in so trenutno v uporabi. Na začetku smo želeli uporabiti popularni sklad MERN (mongoDB, express, react in node.js) v katerem vsaka črka predstavlja eno izmed tehnologij. Odločili smo se, da bomo namesto react izbrali ogrodje Flutter.

## Čelni del

Flutter je odprtokodno ogrodje za razvoj uporabniških vmesnikov. Prva različica je izšla leta 2018. Izdelal jo je Google [15]. Glavni gradnik Flutter aplikacij je gradnik (angl. widget). V bistvu je vsak element pri izgledu aplikacije gradnik. Gradniki so lahko vizualni kot sta gradnika za tekst in gumb. Z gradniki nadziramo postavitev elementov na strani. Še sama stran in aplikacija sta gradnika.



Slika 3.1: Demonstracijska Flutter aplikacija

Flutter omogoča izdelavo hibridnih aplikacij. Hibridne aplikacije združijo lastnosti nativnih in spletnih aplikacij. Njihova popularnost izhaja iz tega, da omogočajo podporo več platformam z uporabo ene kodne zbirke (angl. codebase). Programski jezik Flutterja je dart, ki je specializiran za izdelavo uporabniških vmesnikov [15].

## Zaledni del

Za spletni strežnik smo izbrali tehnologije, ki temeljijo na okolju node.js. Je odprtokodno izvajalno okolje za javascript kodo na spletnem strežniku. Zasnovana je na google chrome V8 pogonu. Uporablja asinhroni dogodkovno orientiran model, zaradi česar je lahek in učinkovit, kot nalašč za podatkovno intenzivne aplikacije v realnem času, ki se izvajajo v porazdeljenih napravah. Uporabili smo dva pomembna paketa. Express omogoča hitrejši in lažji razvoj spletnega strežnika. Mongoose pa omogoča lažjo komunikacijo s podatkovno bazo MongoDB [17] [18].

Za podatkovno bazo smo izbrali mongoDB. Je NoSQL (angl. not only structured query language) podatkovna baza, kar pomeni da ni nobenih povezav oziroma relacij med podatki. Podatki se ne shranjujejo v tabelah ampak v enem dokumentu, ki je v BSON (angl. binary javascript object notation) formatu. Ampak še vedno podpira veliko operacij, ki jih nudi tradicionalna relacijska podatkovna baza [19].

Za opazovanje stanja podatkovne baze smo izbrali orodje MongoDB Compass. Za komunikacijo s strežnikom in testiranje smo uporabljali platformo Postman.

# Predstavitev PRototipne Platforme

V okviru diplomskega dela razvita platforma omogoča prijavljenim uporabnikom postavljanje in odgovarjanje vprašanj. Igrifikacijo smo dodali s sistemom točk, ki jih uporabnik dobi s pozitivnimi glasovi. Uporabnik s pridobivanjem točk dosega različne range. Ogled točk in ranga je možna za vse uporabnike. Zaradi tega, ker je osrednja tema vprašanj na prototipu programerske narave, smo se zgledovali s sistemom Stack overflow.

## Elementi

Med uporabo aplikacije se bo uporabnik srečal z raznimi elementi. Imajo svoja pravila in omejitve, ki jih more uporabnik poznati za uspešno uporabo aplikacije.

### Uporabnik

Če želi oseba uporabljati storitev, se bo morala registrirati. Vsak uporabnik more zaradi overjanja vstaviti elektronski naslov in geslo. Elektronski naslov ne sme biti že v uporabi in more obstajati. S tem omilimo ustvarjanje več računov za eno osebo. Pri registraciji še more vstaviti uporabniško ime, ki bo prikazano v aplikaciji. Dosežki uporabnika so vidni na strani za profil (slika 4.7). Zraven števila postavljenih vprašanj, odgovorov in koristnih odgovorov sta še prikazani seštevki ocen za vsa vprašanja in odgovore. Vsota teh dveh ocen pa je uporabljena za določitev ranga. Uporabnik z manj kot 100 točk ima železni rang, med 100 in 200 točk ima bronasti rang, med 200 in 400 ima srebrni rang, med 400 in 800 ima zlati rang in z več kot 800 točk ima platinast rang. Ko doseže 1600 točk ga lahko administrator ročno promovira na diamantni rang. Diamantni rang omogoča osebi spreminjanje odgovorov drugim uporabnikom.

### Vprašanje

Kadar ima uporabnik kakšno vprašanje ali problem ga lahko objavi. Uporabnik lahko objavi več vprašanj naenkrat. Vsako vprašanje (slika 4.1) je sestavljeno iz naslova, kjer je zapisano vprašanje oziroma tema problema. Vsebovati more 10 do 60 znakov. Opis more vsebovati 10 do 280 znakov in se poglobiti v vprašanje oziroma problem v naslovu. Ima oceno, ki je odvisna od tega koliko pozitivnih in negativnih ocen je dobilo vprašanje. Višja ocena pomeni, da je vprašanje zanimivo in pomembno za odgovoriti. Vprašanje lahko uporabnik označi za »priljubljenega« tako da klikne na ikono s srčkom. S tem bo lahko naslednjič vprašanje lažje našel. Avtor doda vprašanju vsaj tri oznake. Oznaka nam pove v katero panogo programiranja spada vprašanje. Sproti se zapisuje število ogledov vprašanja, kot tudi število odgovorov. Ob ustvarjanju se samodejno shranita datum objave in avtor vprašanja. Naslov, oznake, datum objave, ocena, število ogledov, avtor in priljubljenost se uporabljata za lažje iskanje in filtriranje vprašanj. Avtor ima možnost odstranitve vprašanja. To dejanje ni spodbujeno, ker odgovori ne bodo na voljo drugim uporabnikom, ki imajo podoben problem oziroma vprašanje. Z izbrisom izgubiš tudi oceno za rang, ki si jo pridobil s tem vprašanjem.

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 4.1: Primer vprašanja

### Odgovor

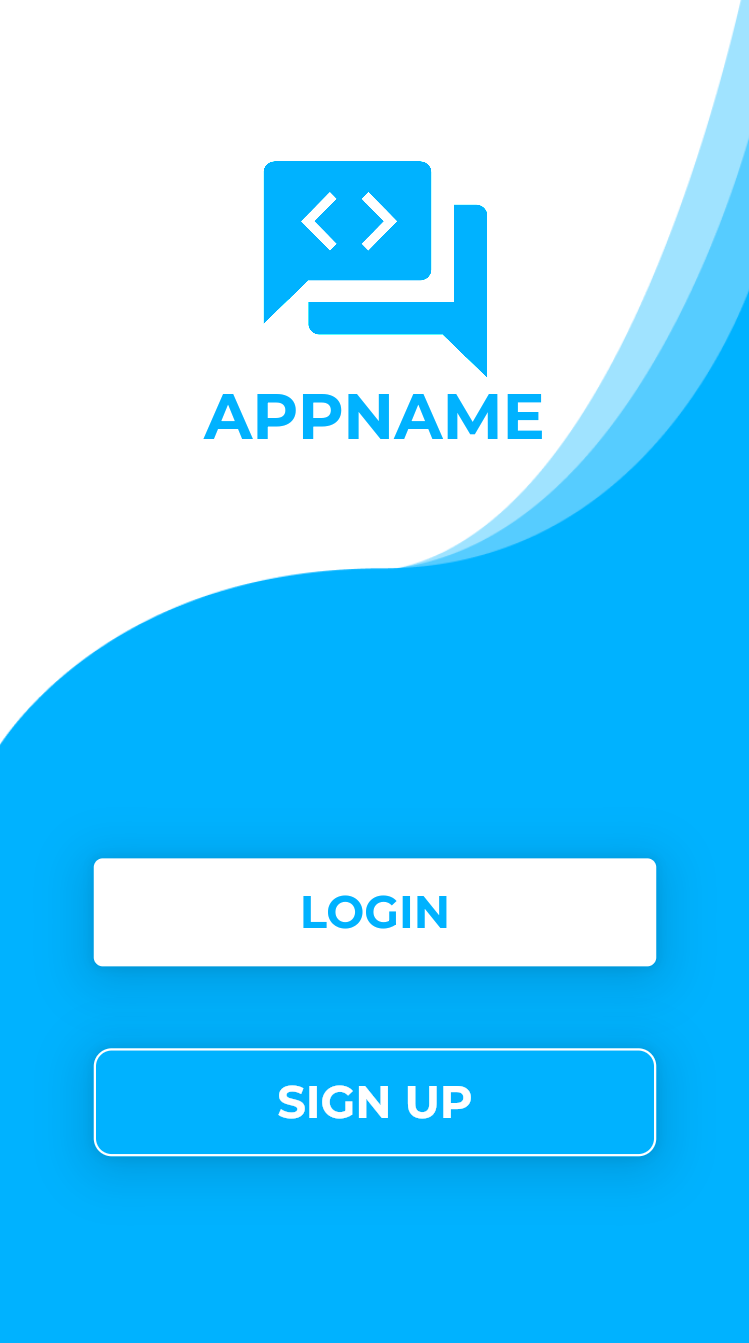
Kadar uporabnik pozna odgovor na vprašanje ali rešitev na problem lahko objavi odgovor. Odgovori lahko na več vprašanj ampak na vprašanje lahko odgovori samo enkrat. Ima možnost spreminjanja svojega odgovora. Če ima diamanten rang lahko tudi spreminja tuje odgovore. Sestavljen je iz teksta, datuma objave, imena avtorja, ocene in oznake za koristen odgovor. Tekst lahko vsebuje 10 do 500 znakov. Avtor vprašanja lahko označi odgovore kot »koristne«. S tem avtor pove ostalim uporabnikom, ki imajo podoben problem ali vprašanje kot on, da mu je ta odgovor pomagal. To, kot tudi datum objave in ocena, se uporabljajo za razvrščanje in filtriranje odgovorov.

### Glas

Ocena vprašanja in odgovora je odvisna od njihovih glasov (angl. vote). Vsak uporabnik lahko glasuje na določeno vprašanje ali odgovor samo enkrat. Ne sme glasovati lastnih objav. Možno je glasovati pozitivno ali negativno. Pozitivni glas bo povečal oceno za 10, negativni pa zmanjšal za 5. Možno je da ima objava negativno oceno. Uporabnik lahko svoj glas vedno spremeni ali odstrani.

## Aplikacija

Uporabniki, ki nimajo samodejne prijave imajo na začetni strani (slika 4.2) možnost, da se prijavijo ali registrirajo.



Slika 4.2: Začetna stran

Ob kliku na gumb »LOGIN« se bo odprla stran za prijavo (slika 4.3). Uporabnik more vnesti pravilni elektronski naslov in geslo, če se želi uspešno prijaviti. Pred postopkom overjanja se preveri, če je naslov podan in v pravilnem formatu in če je geslo podano in dolgo šest do petdeset znakov. Pri uspešni prijavi bo uporabnik preusmerjen na glavno stran (slika 4.5), drugače mu bo aplikacija z besedilom povedala da se je zmotil. Pred prijavo lahko uporabnik obkljuka možnost da ostane prijavljen. Njegovi podatki za prijavo bodo lokalno shranjeni in se uporabili za samodejno prijavo. Zaradi tega bo prva stran uporabnika, ki je izbral to opcijo glavna stran (slika 4.5).

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 4.3: Prijavna stran

Ob kliku na gumb »SIGN UP« se bo odprla stran za registracijo (slika 4.4). Tukaj bo uporabnik ustvaril svoj novi račun. Zraven istih omejitev vnosnih polj pri prijavi more uporabnik še dodatno paziti da njegovo uporabniško ime ne bo daljše od 16 znakov. Ne sme uporabiti uporabniškega imena ali naslova, ki sta že v rabi. Preveri se še tudi, če elektronski naslov dejansko obstaja. Ob uspešni registraciji se na podatkovni bazi zapišejo podatki za novega uporabnika, ki je preusmerjen na glavno stran (slika 4.5).

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 4.4: Registracijska stran

Na glavni strani (slika 4.5) so prikazana vprašanja razvrščena od najnovejšega do najstarejšega. Uporabnik lahko vprašanja oceni in jih označi za priljubljena. Če klikne na ime avtorja bo preusmerjen na njegovo profilno stran (slika 4.7). Klik na naslov vprašanja pa ga preusmeri na stran vprašanja (slika 4.10). Lahko razvršča vprašanja glede na njihovo starost (privzeto), število ogledov in ocene ter filtrira glede na priljubljenost. V iskalno vrstico (slika 4.6) lahko vpiše niz in oznake, ki so zapisane v oglatih vprašanjih. Aplikacija bo pokazala samo tista vprašanja, ki vsebujejo vse zapisane oznake in naslovu vsebujejo niz. Z klikom na zgornji desni gumb se odpre predal (angl. drawer), ki vsebuje povezave do drugih strani.

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjenSlika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

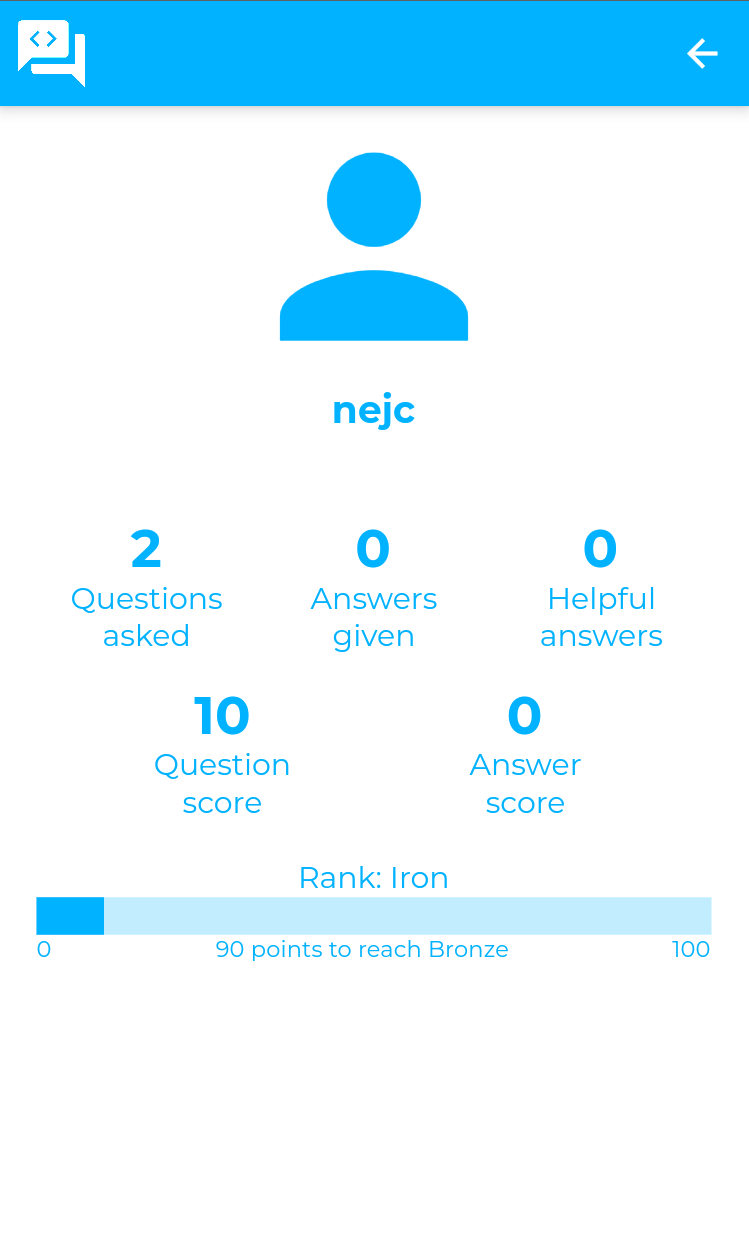
Slika 4.5: Glavna stran

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 4.6: Primer iskalne vrstice

Na profilni strani (slika 4.7) lahko ogledaš statistike in dosežke sebe ali drugih uporabnikov.



Slika 4.7: Profilna stran

Na strani za ustvarjanje vprašanj (slika 4.8) vpišeš naslov, oznake in opis vprašanja. Vprašanje se bo ustvarilo, če bosta besedili pravilne dolžine. Drugače aplikacija opozori

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 4.8: Stran za ustvarjanje vprašanja

Na strani uporabnikovih vprašanj so samo vprašanja, ki jih je ustvaril uporabnik. Lahko jih filtriraš in razvrščaš kot na glavni strani. Svoja vprašanja lahko tudi izbrišeš. Pred izbrisom te bo pojavno okno vprašalo, če to res želiš.

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 4.9: Stran uporabnikovih vprašanj

Ob kliku na naslov vprašanja na glavni strani (slika 4.5) in strani uporabnikovih vprašanj (slika 4.9) boš preusmerjen na stran vprašanja (slika 4.10), ki vsebuje vse podatke o tem vprašanju. Klik na ikono pogovornega okna s plusom bo odprl/zaprl besedilno polje (slika 4.11), v katerega lahko napišeš svoj odgovor. V primeru da si odgovor že podal, bo polje vsebovalo besedilo odgovora, ki ga lahko spreminjaš. Administratorjem, avtorjem odgovora in uporabnikom z diamantnim rangom je možno spreminjanje besedila na drugi način. Ob kliku ikone pisala, ki je zraven odgovora, se bo odprlo pojavno okno (slika 4.12) v katerem lahko besedilo odgovora spremeniš. Tako kot vprašanja lahko tudi odgovore na podoben način razvrščamo in filtriramo. Namesto priljubljenosti filtriramo odgovore po koristnosti. Avtor vprašanja lahko označi katerikoli odgovor za koristnega. To je vidno kot ikona pogovornega okna z zvezdico.

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 4.10: Stran vprašanja

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 4.11: Besedilno polje za odgovor

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 4.12: Pojavno okno za spreminjanje odgovora

# Implementacija

V naslednjem poglavju so opisane osrednje teme implementacije prototipa. To so uporabljeni arhitekturni in načrtovalski vzorci ter hierarhija map in datotek. Poglobljeno smo opisali par delov aplikacije, ki se nam zdijo najbolj zanimivi.

## Zaledni del

Kot temelj našega strežnika smo uporabili že implementirano rešitev storitve, ki imitira zaledni del Stack overflowa [20]. To storitev smo predelali na tak način, da podpira naša pravila in način delovanja, ki smo jih postavili. Strežnik je zasnovan po arhitekturnem vzorcu REST (angl. representational state transfer). To pomeni, da odjemalci pošljejo zahteve strežniku, kot so pridobivanje ali spreminjanje virov, in strežnik jim vrne odzive na njihovo zahtevo. Komunikacija poteka preko HTTP (angl. Hypertext Transfer Protocol) in je brez stanja, kar pomeni, da si strežnik ne shranjuje podatkov o njegovih odjemalcih. Podatki, ki so vrnjeni odjemalcem, so najpogosteje v formatu JSON [21]. Čeprav REST podpira vse metode HTTP, so najpogostejše uporabljene:

* GET za pridobitev vira z določenim ID-jem (angl. Identification card),
* POST za ustvarjanje novega vira,
* PUT za posodobitev vira z določenim ID-jem,
* DELETE za odstranjevanje vira z določenim ID-jem.

Vsaka zahteva more vsebovati pot, ki bo strežniku povedala, kakšno operacijo mora izvesti. Dobra praksa je, da so poti ustvarjene na takšen način, da so berljive za uporabnika [21]. Primer ene izmed poti je »/api/questions« z metodo POST, ki ustvari vprašanje, če je uporabnik prijavljen in je vprašanje v dovoljeni obliki.

Strežnik (slika 5.1) je oblikovan po načrtovalskem vzorcu MVC (angl. Model–view–controller), ki razdeli odgovornost. Usmernik (angl. router) preusmeri zahteve v pravilni kontroler. V njem se upravlja vsa logika zahteve. Kliče funkcije modela, kadar želi pridobiti ali spreminjati podatke v bazi. Nato pošlje odgovor odjemalcu, ki bo glede na podatke v odzivu oblikoval spletno stran. Torej je v tem vzorcu čelni del aplikacije v vlogi pogleda (angl. view).

Dodali smo tudi par funkcij vmesnega sloja (angl. middlewares). To so povratni klici (angl. callbacks), ki prestrežejo zahteve http, preden pridejo do kontrolerja. Zmožni so dostopa in spreminjanja parametrov zahteve. Najpogosteje smo jih uporabljali pri overjanju uporabnika in preverjanju pravic.

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 5.1: Mape zalednega dela

### Overjanje

Za overjanje uporabnika se uporabljajo žetoni JWT (angl. Javascript object notation web tokens). Z njimi lahko predstavimo uporabnikovo identiteto brez, da bi za vsako zahtevo poslali njegove osebne podatke. Kadar uporabnik pošlje pravilni elektronski naslov in geslo, se bo ustvaril žeton (slika 5.2), ki je podpisan z ID, imenom, vlogo in promoviranostjo uporabnika. Ta žeton bo nato poslan uporabniku, ki ga bo lahko dodal v zahtevo, ko bo potrebno overjanje. Žetoni imajo dobo 7 dni in so zgoščeni z algoritmom HS256.

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 5.2: Ustvarjanje žetona

## Čelni del

Tako kot zaledni del smo tudi kodo aplikacije razdelili na več enot oziroma map. Storitve (angl. services) vsebujejo kodo, ki s klici HTTP komunicira s strežnikom. Torej kadar želimo dostopati do storitve strežnika, se sklicujemo na storitve. Ker so odzivi v formatu JSON, moramo odziv deserializirati, preden ga lahko uporabimo. To naredimo s pomočjo paketa dart.convert in razredov, ki imajo enake atribute kot JSON.   
Namesto da ročno ustvarimo razrede za vsak različni odziv, lahko pospešimo naše delo s spletno stranjo Quicktype (slika 5.3). Omogoča samodejno kreiranje razreda kot tudi funkcije za serializacijo in deseralizacijo za podan niz JSON. Podpira ogromno jezikov, kot so C++, Java, Kotlin, Python, Rust, Ruby in Dart. Te razrede shranjujemo v enoti modeli.

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 5.3: Primer uporabe strani Quicktype

Pripomočki (angl. utilities) vsebujejo kodo za skupne nastavitve (angl. shared preferences), ki omogoča lokalno shranjevanje podatkov tipa ključ-vrednost. To so osebni podatki, ki so potrebni za delovanje aplikacije in samodejno prijavo. Ti podatki se shranijo pri vsaki prijavi in odstranijo pri vsaki odjavi.

Največja enota je UI (angl. user interface), ki skrbi za izgled aplikacije. Razdeljena je po straneh, pri kateri ima vsaka mapa kodo, ki je specifična za določeno stran. Mapa »widgets« vsebuje gradnike, ki se pojavijo na več straneh, mapa »theme« pa vsebuje osrednjo temo in barve aplikacije.

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 5.4: Mape čelnega dela

### Ovoj gumba

Kadar smo preverjali, kako izgleda stran pri manjših zaslonih, smo ugotovili, da imata gumba na prijavni in registracijski strani (slika 4.3 in 4.4) premalo prostora. Zato smo ovili gumba z gradnikom (slika 5.6), ki bo deloval kot ovojnica (angl. wrapper) za gumba. Uporabili smo gradnik zaboj (angl. container), ki je eden izmed najpogostejših gradnikov. Uporablja se za določanje velikosti, pozicije in roba gradnika, ki ga obdaja. V našem primeru gradnik obdaja vrstico ali stolpec (odvisno od višine zaslona) in ne sme presegati 250 enot dolžine. Kadar bo višina zaslona manjša od določenega praga, se bo spremenila postavitev gumbov. Pri dovolj velikem prostoru bosta postavljena v enem stolpcu, če ne pa v eni vrstici (slika 5.5).

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 5.5: Možni postavitvi gumbov

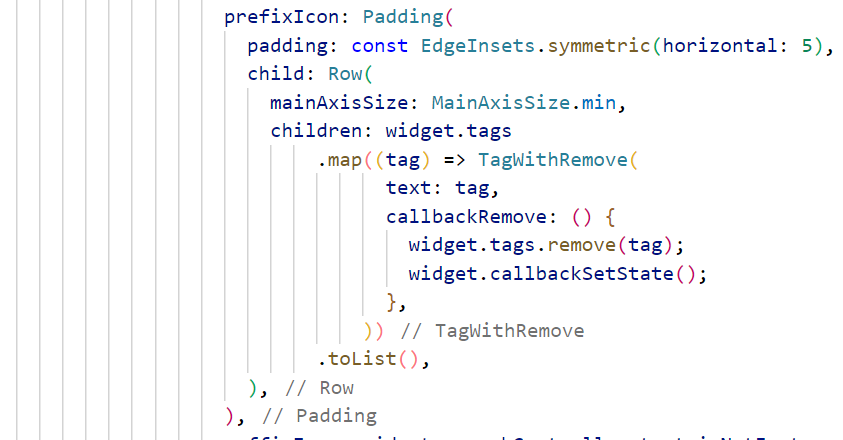
Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 5.6: Koda za gradnik ButtonWrapper

### Iskalna vrstica

Iskalna vrstica na glavni strani (slika 4.5) je bila zapletena za implementacijo. Navdih smo dobili s stack overflowa, kjer lahko iščemo z oznakami in iskalnim nizom tako, da oznake ovijemo z oglatimi oklepaji. Želeli smo nadgraditi izgled oznak tako (slika 4.6), da izgledajo enako kot oznake pri vprašanjih. Morali smo rešiti več problemov. Prvi problem je bil sam prikaz oznake. Iskalna vrstica je gradnik TextFormField, ki ima parameter »suffixIcon«. Običajno se uporablja za prikaz ikon na začetku iskalne vrstice, ker pa je tip parametra gradnik, lahko v njega vstavimo karkoli. Zato smo vstavili vrstico (angl. Row), ki vsebuje vse oznake (slika 5.7).



Slika 5.7: Koda za prikaz oznak v iskalni vrstici

Drugi problem je ločevanje oznak in iskalnega niza. To smo rešili s pomočjo regularnih izrazov (angl. regular expressions), ki so ignorirali vso vsebino v oglatih oklepajih in s tem izluščili iskalni niz. Tretji problem pa je, kdaj spremenimo oznako iz niza v gradnik. Odločili smo se, da bo uporabnik za vsako končano oznako kliknil gumb za presledek, ki bo niz v oglatih oklepajih dodal v listo nizov »tags«. Ta niz bo pa izpisan v gradniku.

### Samodejna prijava

Osebni podatki o uporabniku so tudi po zaprtju lokalno shranjeni v napravi. Torej če uporabnik na prijavni strani (slika 4.3) označi, da želi ostati vpisan, lahko uporabimo njegove podatke za vpis brez njegovega vnosa. Uporabili smo spremenljivko, ki se shrani tudi v skupnih nastavitvah in pove, ali se je uporabnik odločil za samodejno prijavo. Preden aplikacija zgradi izgled za začetno stran (slika 4.2), preveri, ali je vklopljena samodejna prijava in ali sta shranjena uporabniško ime in geslo. Po uspešni prijavi bo uporabnik samodejno preusmerjen na glavno stran (slika 4.5). Uporabnik lahko prekliče samodejno prijavo tako, da se odjavi.

# Testiranje

Preden smo začeli komunicirati s strežnikom preko aplikacije, smo pošiljali zahtevke preko spletne aplikacije Postman. Omogoča elegantno ustvarjanje zahtev in lepi izpis odziva. Ustvarili smo zahteve za vsako funkcionalnost strežnika in jih razdelili v mape (slika 6.1).

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 6.1: Primer uporabe postmana

Sproti lahko vsako zahtevo posebej še testiramo s pisanjem testov. Običajno smo vsakemu klicu napisali štiri tipe testov. Prvi in najpreprostejši test preverja, ali je statusna koda pravilna. Običajno mora biti 200, razen v primeru, kjer ustvarimo (koda 201) ali odstranimo (koda 204) element. Drugi test preverja, ali ima odziv vse potrebne lastnosti. Tretji test preveri, ali so vse lastnosti pravilnega tipa. Zadnji tip pa preveri, ali ima lastnost pravilno vrednost. Pri takih tipih preverimo, ali ima niz pravilno število znakov in ali je število v pravilnem intervalu. V primeru zahteve za profil (slika 6.2) se testira, ali ima niz id točno 24 znakov in ali je število postavljenih vprašanj, odgovorov in koristnih odgovorov večje ali enako 0.

Slika, ki vsebuje besede besedilo

Opis je samodejno ustvarjen

Slika 6.2: Primer testov za zahtevo za profil

Ker ima večina odzivov podatke v obliki uporabnika, vprašanja ali odgovora, smo zanje ustvarili teste v globalni datoteki. Na njih se sklicujemo v posameznih testih za odziv.

# SKLEP

Vsi cilje, ki smo jih podali na začetku projekta, so bili uspešno dokončani. Končni izdelek je delujoča aplikacija, ki vsebuje vso osnovno funkcionalnost za podporo spletnim skupnostim. Deluje na spletu in operacijskih sistemih Android in iOS.

Največja težava pri razvoju izdelka je bila odzivnost (angl. responsive design) čelnega dela. Probali smo ustvariti čim lepši izgled za vse velikosti, razmerja in postavitve zaslona. Pri velikih računalniških zaslonih je na robovih straneh kar veliko praznega prostora. Izgled ni dobro optimiziran za zaslone, ki niso popolnoma »pravokotni«, kot so na telefonih iPhone.

Problematično je bilo to, da nekatere funkcionalnosti v Flutterju in njegovih paketih niso podpirale določene platforme ali različice. Zato smo včasih morali nekatere dele aplikacije rešiti znova z drugimi metodami. Na srečo smo omejitev našli dokaj hitro.

Primer tega je bil s paketom flutter\_secure\_storage. Uporablja se za podobne namene kot skupne nastavitve (torej lokalno shranjevanje podatkov), ampak podatke tudi šifrira z AES (angl. Advanced Encryption Standard). To je v našem primeru zelo koristno, ker morajo biti shranjeni podatki, kot je geslo, zaščiteni. Paketa nismo mogli uporabiti, ker ni podprt na spletu.

Kadar osvežimo stran na spletu, se bodo vse lokalne spremenljivke ponastavile na začetno vrednost. To v nekaterih primerih pomaga, kot je to pri ponastavitvi vseh filtrov in iskalnega niza pri glavni strani. Škodi pa, kadar želimo prenesti podatke iz ene strani na drugo. Ko se spletna stran ponovno naloži, se ti podatki ponastavijo, zato ne moremo ustvariti spletne strani na želen način. Pri ustvarjanju profilne strani mora Flutter vedeti, od katerega uporabnika naj zahteva informacijo. Če tega podatka bo zahteval podatka iz prijavljenega uporabnika, kar v nekaterih primerih ni želeno. Pri osvežitvi se tudi izbriše vsa zgodovina obiska strani. Torej se s klikom na gumb »nazaj« vrnemo na začetno stran in ne na prejšnjo.

Za nadaljnje delo bi priporočali implementacijo lokalizacije. Trenutno je edini podprti jezik angleščina, zato bi bilo dobro dodati še slovenščino. Lahko bi dodali privilegije za vse range, ne pa samo diamanta. Ena izmed teh bi lahko bila možnost združevanja dveh oznak, ki imata enak ali zelo podoben pomen (na primer nodejs in node.js). Aplikacija bi lahko sledila naši aktivnosti in glede na to poskusila ugotoviti naše obnašanje. Na primer, če pogosto odgovarjamo na vprašanja, ki imajo določeno oznako, bomo dobili naziv, ki to omeni. Zraven tega bi lahko dodali še dosežke, pri katerih bi dobili značko za določeno aktivnost (na primer, da smo postavili 100 vprašanj).

Viri in literatura

[1] Wikipedia Primerjava Q&A strani, <https://en.wikipedia.org/wiki/Comparison_of_Q%26A_sites> [18.7.2022]

[2] Wikipedia Igrifikacija, <https://sl.wikipedia.org/wiki/Igrifikacija> [18.7.2022]

[3] Gamify Igrifikacija, <https://www.gamify.com/what-is-gamification> [19.7.2022]

[4] Growth Engineering Igrifikacija, <https://www.growthengineering.co.uk/definition-of-gamification> [19.7.2022]

[5] Finances Online, <https://financesonline.com/number-of-gamers-worldwide> [19.7.2022]

[6] Črepinšek M. *Teorija iger in tehnike usmerjanja vedenja uporabnikov z elementi iger.* Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko

[7] Investopedia Igrifikacija, <https://www.investopedia.com/terms/g/gamification.asp> [19.7.2022]

[8] Wikipedia Stack Overflow, <https://en.wikipedia.org/wiki/Stack_Overflow> [20.7.2022]

[9] Stack overflow Ugled, <https://stackoverflow.com/help/whats-reputation> [20.7.2022]

[9] Stack overflow Privilegiji, <https://stackoverflow.com/help/privileges> [20.7.2022]

[10] Healthtap Vprašanja in odgovori, <https://www.healthtap.com/q/ask-a-doctor-a-question-free-online/> [17.8.2022]

[11] Wikipedia Quora, <https://en.wikipedia.org/wiki/Quora> [20.7.2022]

[12] Tech Crunch Quora, <https://techcrunch.com/2012/02/21/now-you-need-quora-credits-to-ask-questions-but-can-also-use-them-to-promote-content/> [20.7.2022]

[13] Reddit Karma <https://reddit.zendesk.com/hc/en-us/articles/204511829-What-is-karma-> [20.7.2022]

[14] Reddit Coins <https://www.reddit.com/coins> [20.7.2022]

[15] Kosar T. *Hibridne mobilne aplikacije.* Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko

[16] Flutter by example, <https://flutterbyexample.com/lesson/thinking-in-widgets> [22.7.2022]

[17] Procoders Express in Node.js <https://procoders.tech/blog/express-js-vs-node-js/> [23.7.2022]

[18] Kosar T. *Node.js.* Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko

[19] Kosar T. *Express.js, Mongoose, MongoDB.* Maribor: Fakulteta za elektrotehniko, računalništvo in informatiko

[20] Github Stack Overflow clone <https://github.com/salihozdemir/stackoverflow-clone> [31.7.2022]

[21] Code Academy REST <https://www.codecademy.com/article/what-is-rest> [31.7.2022]