

ŠOLSKI CENTER VELENJE
ELEKTRO IN RAČUNALNIŠKA ŠOLA
Trg mladosti 3, 3320 Velenje

MLADI RAZISKOVALCI ZA RAZVOJ ŠALEŠKE DOLINE

RAZISKOVALNA NALOGA

ŠOLSKA APLIKACIJA ZA DIJAKE

Tematsko področje: Računalništvo

Avtorja:

Blaž Osredkar, 2. letnik
Urban Krepel, 2. letnik

Mentorja:

Islam Mušić, prof.
Iztok Osredkar, inž. elek.

Velenje, 2022

Raziskovalna naloga je bila opravljena na ŠC Velenje, Elektro in računalniški šoli, 2022.

Mentorja: Islam Mušić, prof., Iztok Osredkar, inž. elek.

Datum predstavitve: marec 2022

KLJUČNA DOKUMENTACIJSKA INFORMACIJA

ŠD Elektro in računalniška šola Velenje, šolsko leto 2021/2022

KG mobilna aplikacija/programiranje

AV OSREDKAR, Blaž / KREPEL, Urban

SA MUŠIČ, Islam / OSREDKAR, Iztok

KZ 3320 Velenje, Trg mladosti 3

ZA ŠC Velenje, Elektro in računalniška šola, 2022

LI 2022

IN **ŠOLSKA APLIKACIJA ZA DIJAKE**

TD Raziskovalna naloga

OP IX, 58 str., 66 sl., 14 vir.

IJ SL

JI sl / en

AI Dandanes uporabljamo za pridobivanje informacij različna spletna orodja. Pri tem ima veliko vlogo tudi šolanje, saj preko spletnih portalov izvemo, kaj imamo na urniku, kdaj se dogaja kakšen dogodek, ki je v povezavi s šolo ... Za iskanje teh orodij porabimo precej dragocenega časa, ki bi ga lahko porabili za kaj bolj koristnega. Platforma bo podpirala mobilne telefone (z operacijskim sistemom iOS ali Android) ter spletne brskalnike. Cilj raziskovalne naloge je bil za dijake ustvariti platformo, ki bi združevala spletna informacijska orodja različnih podjetij, ter s tem skrajšati čas iskanja le-teh. Ime platforme je ŠCVApp.

KEY WORDS DOCUMENTATION

ND Elektro in računalniška šola Velenje, šolsko leto 2021/2022

CX mobilna aplikacija/programiranje

AU OSREDKAR, Blaž / KREPEL, Urban

AA MUŠIČ, Islam / OSREDKAR, Iztok

PP 3320 Velenje, Trg mladosti 3

PB ŠC Velenje, Elektro in računalniška šola, 2022

PY 2022

TI SCHOOL APPLICATION FOR STUDENTS

DT Raziskovalna naloga

NO IX, 60 p., 66 fig., 14 ref.

LA SL

AL sl / en

AB The obligatory "equipment" of a contemporary student is the mobile phone. We use it several hours a day for communication, entertainment, and school. Due to the fact that the world is becoming more and more digital, all the information can be acquired online, the advantage being that this makes it easier to inform people, whereby school is no exception. The idea of our research project is to create and upgrade a mobile application for students of the School Centre Velenje, where everything needed for school is in one place. We named it ŠCVApp. The application contains all the information necessary for schooling at the School Centre Velenje. By using the application, the student reviews the schedule and news from the website, determines lunch preferences... We wanted to create an application that provides assistance for students in search of relevant information about schooling.

KAZALO VSEBINE

1	UVOD.....	1
1.1	HIPOTEZE	1
2	PREGLED OBJAV	2
2.1	Programska orodja	2
2.1.1	Flutter	2
2.1.2	React	3
2.1.3	NestJS	3
2.1.4	Microsoft Azure.....	4
2.1.5	cPanel	4
2.1.6	Figma	5
2.1.7	Android Studio	5
2.1.8	Xcode.....	5
2.1.9	Heroku	5
3	MATERIAL IN METODE DELO	6
3.1	Iskanje podobnih aplikacij na tržišču	6
3.2	Microsoft OAuth 2.0	6
3.2.1	Registracija aplikacije na Microsoft Azure	7
3.2.2	Uporaba Microsoft OAuth2 v naši aplikaciji	8
3.3	Izgled aplikacije.....	10
3.3.1	Izgled aplikacije v Figma	11
3.4	Prevajanje gumba »Login with Microsoft« v slovenščino	13
3.5	Platforme aplikacije	14
3.5.1	Čelni del spletne aplikacije.....	14
3.6	Gostovanje in domena aplikacije.....	19
4	REZULTATI	20
4.1	Prijava s šolskim Microsoftovim računom.....	20
4.2	Platforme aplikacije	20

4.2.1	Spletna platforma.....	20
4.2.2	Mobilna platforma	25
4.2.3	Posodobljena mobilna verzija aplikacije	28
4.3	Uvajanje preizkušne različice programa za mobilno aplikacijo	35
4.3.1	Razvoj preizkušne različice za iOS naprave.....	35
4.3.2	Razvojna različica za Android naprave	37
4.4	Težave.....	40
4.4.1	Težave čelnega dela aplikacije	40
4.4.2	Težave zalednega dela aplikacije	40
4.4.3	Težave z objavo aplikacije za naprave Android.....	40
5	RAZPRAVA.....	41
6	ZAKLJUČEK	43
7	POVZETEK	44
8	ZAHVALA.....	45
9	PRILOGE	46
10	VIRI IN LITERATURA.....	48

KAZALO SLIK

Slika 1: Prikaz logotipa programskega jezika Flutter.....	2
Slika 2: React logo (3).....	3
Slika 3: Nest logo (5).....	3
Slika 4: Microsoft Azure logo (7).....	4
Slika 5: cPanel logotip (9).....	4
Slika 6: Logotip Figma.....	5
Slika 7: Diagram protokola za prijavo v Microsoftov račun (14).....	7
Slika 8: Diagram našega zalednega dela (Vir: lasten).....	8
Slika 9: .env datoteka (Vir: lasten).....	8
Slika 10: Diagram za preusmeritev uporabnika na prijavno stran (Vir: lasten).....	9
Slika 11: Diagram za ureditev Microsoftove preusmeritve (Vir: lasten).....	9
Slika 12: Diagram za pridobitev osnovnih uporabnikovih podatkov (Vir: lasten).....	10
Slika 13: Diagram za odjavo uporabnika (Vir: lasten).....	10
Slika 14: Osnovna predloga za izgled v aplikaciji Figma (Vir: lasten).....	11
Slika 15: Izgled prijavne strani izdelan v aplikaciji Figma (Vir: lasten).....	11
Slika 16: Slika izgleda stranskega menija, ustvarjenega v Figma (Vir: lasten).....	12
Slika 17: Izgled odseka nastavitve, ustvarjen v Figma (Vir: lasten).....	13
Slika 18: Prikaz, kako uporabljati gumb "Login with Microsoft" (Vir: Microsoft).....	13
Slika 19: Urejen gumb "Vpis z Microsoft računom". (Vir: lasten).....	14
Slika 20: React komponenta za urejanje spletnih poti (Vir: lasten).....	15
Slika 21: Diagram za zahtavo uporabnikovih podatkov (Vir: lasten).....	16
Slika 22: Primer vnosa uporabnikovih podatkov na spletno stran (Vir: lasten).....	16
Slika 23: Prikaz šolske spletne strani (Vir: lasten).....	16
Slika 24: Komponenta »Link« (Vir: lasten).....	17
Slika 25: Prikaz določene komponente glede na spletno pot (Vir: lasten).....	18
Slika 26: Spletna stran zalednega dela (Vir: lasten).....	19
Slika 27: Podatek, da si lahko uporabnik zmeraj izbere račun za prijavo.....	20
Slika 28: Prijavna stran ŠCVApp-a (Vir: lasten).....	20
Slika 29: Spletna stran šole (Vir: lasten).....	21
Slika 30: Spletni portal malice na spletnem mestu ŠCVApp (Vir: lasten).....	21
Slika 31: Spletni portal eAsistent na spletnem mestu ŠCVApp (Vir: lasten).....	22

Slika 32: Zavihek nastavitve na spletnem mestu ŠCVApp (Vir: lasten).....	22
Slika 33: Spletni portal SIO.MDM na spletnem mestu ŠCVApp (Vir: lasten).....	23
Slika 34: Spletni portal Arnes učilnice na spletnem mestu ŠCVApp (Vir: lasten)	23
Slika 35: Orodja Office (Word, Excel ...), (Vir: lasten)	24
Slika 36: Prikaz ur na spletnem mestu ŠCVApp (Vir: lasten)	24
Slika 37: Prikaz profila na spletnem mestu ŠCVApp (Vir: lasten)	24
Slika 38: Spletna stran šole (Vir: lasten)	25
Slika 39: Urnik (Vir: lasten)	26
Slika 40: Spletni portal malice (Vir: lasten)	26
Slika 41: Spletni portal eAsistent (Vir: lasten).....	27
Slika 42: Nastavitve v aplikaciji (Vir: lasten)	27
Slika 43: Izbor šole (Vir: lasten)	28
Slika 44: Prva stran uvodnega dela aplikacije v aplikaciji ŠCVApp (Vir: lasten)	29
Slika 45: Druga stran uvodnega dela aplikacije v aplikaciji ŠCVApp (Vir: lasten)	29
Slika 46: Tretja stran uvodnega dela aplikacije v aplikaciji ŠCVApp (Vir: lasten).....	30
Slika 47: Prijava z šolskim Microsoft računom v aplikaciji ŠCVApp (Vir: lasten).....	30
Slika 48: Domača stran šole v aplikaciji v aplikaciji ŠCVApp (Vir: lasten)	31
Slika 49: Spletni portal malic v aplikaciji ŠCVApp (Vir: lasten)	32
Slika 50: Spletni portal eAsistent v aplikaciji ŠCVApp (Vir: lasten)	32
Slika 51: Urnik razreda, ki ga obiskujemo v aplikaciji ŠCVApp (Vir: lasten).....	33
Slika 52: Zavihek nastavitve v aplikaciji ŠCVApp (Vir: lasten).....	33
Slika 53: Menjava statusa v aplikaciji ŠCVApp (Vir: lasten).....	34
Slika 54: Vkllop temnega načina (Vir: lasten)	34
Slika 55: Nastavitev biometričnega odklepanja v aplikaciji ŠCVApp (Vir: lasten)	35
Slika 56: Prikaz TestFlight zavihka v App Store Connect-u (Vir: lasten)	36
Slika 57: Aplikacija TestFlight (Vir: lasten)	37
Slika 58: Povratne informacije preizkuševalcev (Vir: lasten).....	37
Slika 59: Uporabnik, ki je pridružen preizkušnji na Android platformi (Vir: lasten)	38
Slika 60: Spletna stran Play Console, kjer uvajamo zaprto preizkušnjo (Vir: lasten)....	39
Slika 61: Funkcija za preusmeritev uporabnika na prijavno stran (Vir: lasten)	46
Slika 62: Funkcija za generiranje URL-ja za preusmeritev uporabnika na prijavno stran (Vir: lasten).....	46

Slika 63: Funkcija za ureditev preusmeritve iz Microsofta po prijavi (Vir: lasten)	46
Slika 64: Funkcija za pridobitev osnovnih uporabnikovih podatkov (Vir: lasten).....	47
Slika 65: Funkciji za odjavo uporabnika (Vir: lasten).....	47
Slika 66: Funkcija, ki pridobi uporabnikove podatke (Vir: lasten).....	47

SEZNAM OKRAJŠAV IN SIMBOLOV

API – Application programming interface

ID – identifier

URL – Uniform Resource Locator

Zaledni del – del aplikacije, do katerega uporabnik ne dostopa neposredno, običajno je odgovoren za shranjevanje in obdelavo podatkov.

Čelni del – del aplikacije, s katerim uporabnik neposredno komunicira; pomemben je predvsem dizajn aplikacije.

ERŠ – Elektro in računalniška šola

1 UVOD

Za šolanje v tako imenovani dobi tehnologije potrebujemo, oz. uporabljamo veliko spletnih aplikacijah različnih podjetij (eAsistent, Arnes, Office 365 ...). Z iskanjem teh aplikacij vsako posebej porabimo kar nekaj časa, ki bi ga lahko zapravili za druge, nam ljubše stvari. Najin cilj je bil povezati vse te aplikacije v nekakšno enotno aplikacijo ter omogočiti prijavo, da lahko aplikacije varno uporabljamo. Želela sva, da bi aplikacija bila dostopna povsod, zato sva ustvarila mobilno in spletno aplikacijo.

1.1 HIPOTEZE

1. hipoteza: S pomočjo šolskega Microsoftovega računa lahko ustvarimo prijavo na našo aplikacijo.
2. hipoteza: V najino aplikacijo bova lahko vključila še 10 ostalih orodij, ki so potrebna za šolanje.
3. hipoteza: Spletna verzija ŠCVApp-a bo imela podporo na vseh brskalnikih.
4. hipoteza: Z uporabo programskega ogrodja Flutter bova pridobila podporo na iOS in Android platformo.

2 PREGLED OBJAV

2.1 Programska orodja

2.1.1 Flutter

Flutter je odprtokoden komplet za razvoj programske opreme uporabniškega vmesnika, ki ga je ustvaril Google. Uporablja se za razvoj aplikacij za več platform za Android, iOS, Linux, macOS, Windows, Google Fuchsia in splet iz ene kodne baze. Flutter je bil prvič opisan leta 2015, izšel je maja 2017.

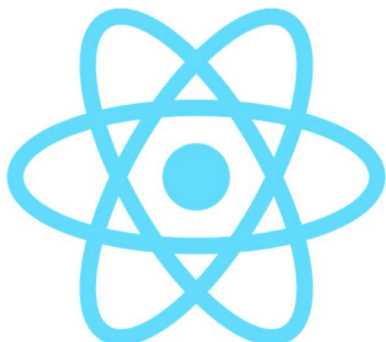
Aplikacije Flutter so napisane v skriptnem programskem jeziku Dart in uporabljajo številne naprednejše funkcije tega jezika. (1)



Slika 1: Prikaz logotipa programskega jezika Flutter

2.1.2 React

React je odprtokodna, učinkovita in prilagodljiva knjižnica JavaScript za gradnjo uporabniških vmesnikov. Razvilo jo je podjetje Meta. Omogoča sestavljanje zapletenih uporabniških vmesnikov iz majhnih izoliranih kosov imenovanih komponente. (2)



Slika 2: React logo (3)

2.1.3 NestJS

Nest (NestJS) je okvir ogrodja za gradnjo učinkovitih, razširljivih aplikacij na strani strežnika Node.js. Uporablja skriptni programski jezik JavaScript, ki je zgrajen in v celoti podpira programski jezik TypeScript. Pod pokrovom Nest uporablja robustne okvire strežnika HTTP, kot je Express. (4)



Slika 3: Nest logo (5)

2.1.4 Microsoft Azure

Microsoft Azure je oblachna racunalniska storitev, ki jo upravlja Microsoft. Zagotavlja programsko opremo kot storitev (angl. SaaS), platformo kot storitev (angl. PaaS) in infrastrukturo kot storitev (angl. IaaS) ter podpira številne različne programske jezike, orodja in ogrodja, vključno s programsko opremo in sistemi, specifičnimi za Microsoft, in sistemi drugih proizvajalcev. (6)



Slika 4: Microsoft Azure logo (7)

2.1.5 cPanel

cPanel je programska oprema, ki vsebuje nadzorno ploščo, s katero nadzoruješ spletno gostovanje, razvilo ga je podjetje cPanel. Zagotavlja grafični vmesnik (GUI) in orodja za avtomatizacijo, zasnovana za poenostavitev postopka gostovanja spletnega mesta lastniku spletnega mesta ali "končnemu uporabniku". (8)



Slika 5: cPanel logotip (9)

2.1.6 Figma

Figma je spletni urejevalnik vektorske grafike in orodje za izdelavo prototipov. Mobilna aplikacija Figma za Android in iOS omogoča ogled in interakcijo s prototipi Figma v realnem času na mobilnih napravah. Nabor funkcij Figma se osredotoča na uporabo v uporabniškem vmesniku in oblikovanju uporabniške izkušnje, s poudarkom na sodelovanju v realnem času. (10)



Slika 6: Logotip Figue

2.1.7 Android Studio

Android Studio je razvojno okolje za operacijski sistem Android. Android Studio je uradno integrirano razvojno okolje za Googlov operacijski sistem Android, zgrajeno na programski opremi IntelliJ IDEA podjetja JetBrains in zasnovano posebej za razvoj Androida. (11)

2.1.8 Xcode

Xcode je IDE – integrirano razvojno okolje, ki ga je ustvaril Apple za razvoj programske opreme za macOS, iOS, watchOS in tvOS. Je edino uradno podprto orodje za ustvarjanje in objavljanje aplikacij v Applovi trgovini z aplikacijami in je zasnovano za uporabo začetnikom in izkušenim razvijalcem. (12)

2.1.9 Heroku

Heroku je platforma v oblaku kot storitev, ki podpira več programskih jezikov. Heroku je ena prvih platform v oblaku, ki se razvija od junija 2007, ko je podpiral samo programski jezik Ruby, zdaj pa podpira Java, Node.js, Scala, Clojure, Python, PHP in Go. (13)

3 MATERIAL IN METODE DELA

3.1 Iskanje podobnih aplikacij na tržišču

Pred začetkom razvoja sva se lotila iskanja podobnih aplikacij. Teh na trgu ni veliko. Med iskanjem sva našla samo aplikacijo Moja Šola, katere razvijalec je eAsistent.

Prednosti aplikacije Moja Šola:

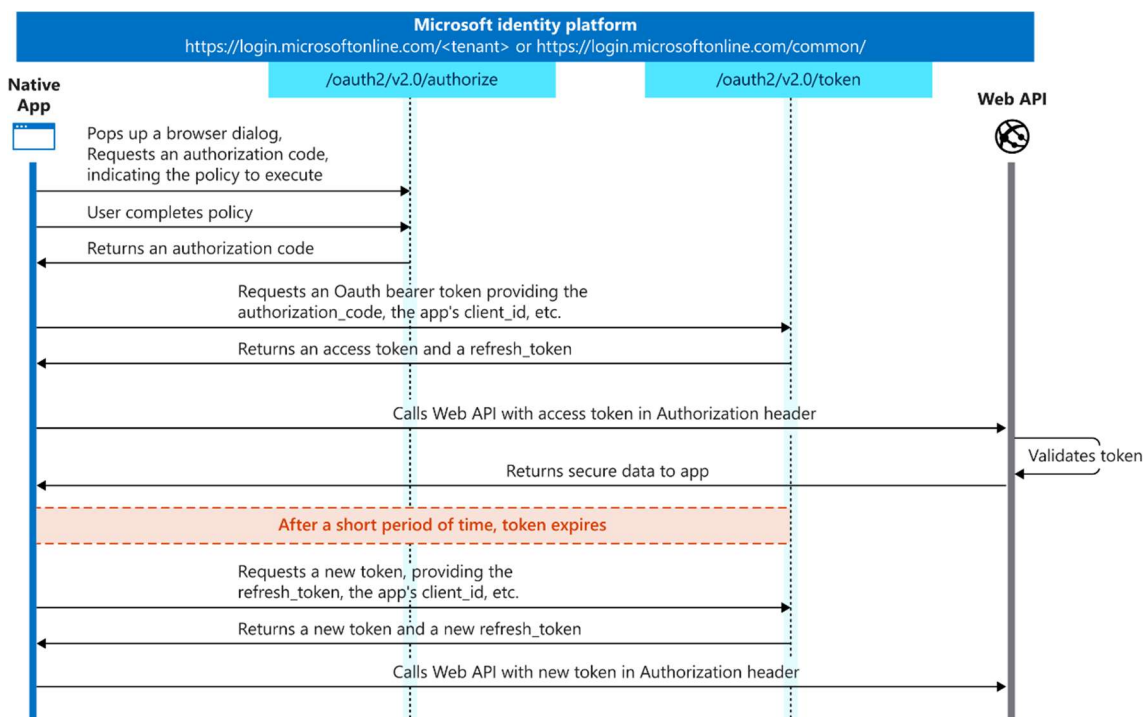
- razvijalec aplikacije je eAsistent, kar pomeni boljšo podporo pri urnikih ter funkcijah eAsistenta;
- redovalnica,
- koledar z možnostjo napovedovanja ocenjevanja znanja.

Slabosti aplikacije Moja Šola:

- ne združuje vseh orodij, vendar uporablja samo svoja;
- ne moremo se prijaviti z Microsoftovim šolskim računom;
- je plačljiva;
- je univerzalna za vse šole po Sloveniji in ni prilagojena za šolo, ki jo obiskujemo.

3.2 Microsoft oAuth 2.0

Da lahko v naši aplikaciji uporabiva prijavo z Microsoftovim računom, morava najprej pogledati, kako to deluje. Za lažjo razlago bova uporabila Slika 7.



Slika 7: Diagram protokola za prijavo v Microsoftov račun (14)

Primer postopka prijave

1. Odpre se nam okno, na katerem je Microsoftova prijavna stran.
2. V stran vnesemo svoje vpisne podatke (uporabniško ime/e-pošto, geslo), in če zahteva potrditev zaradi kakšnih podatkov, potrdimo še to.
3. Microsoftov API vrne kodo, s katero bomo lahko potem dobili dostopni žeton, žeton za osvežitev ...
4. Potem naredimo »GET« prošnjo, v kateri vključimo to kodo.
5. V prošnji Microsoftov API vrne dostopni žeton, žeton za osvežitev, kdaj dostopni žeton poteče ...
6. Potem lahko z dostopnim žetonom dobimo uporabnikove podatke (do katerih imamo dostop).

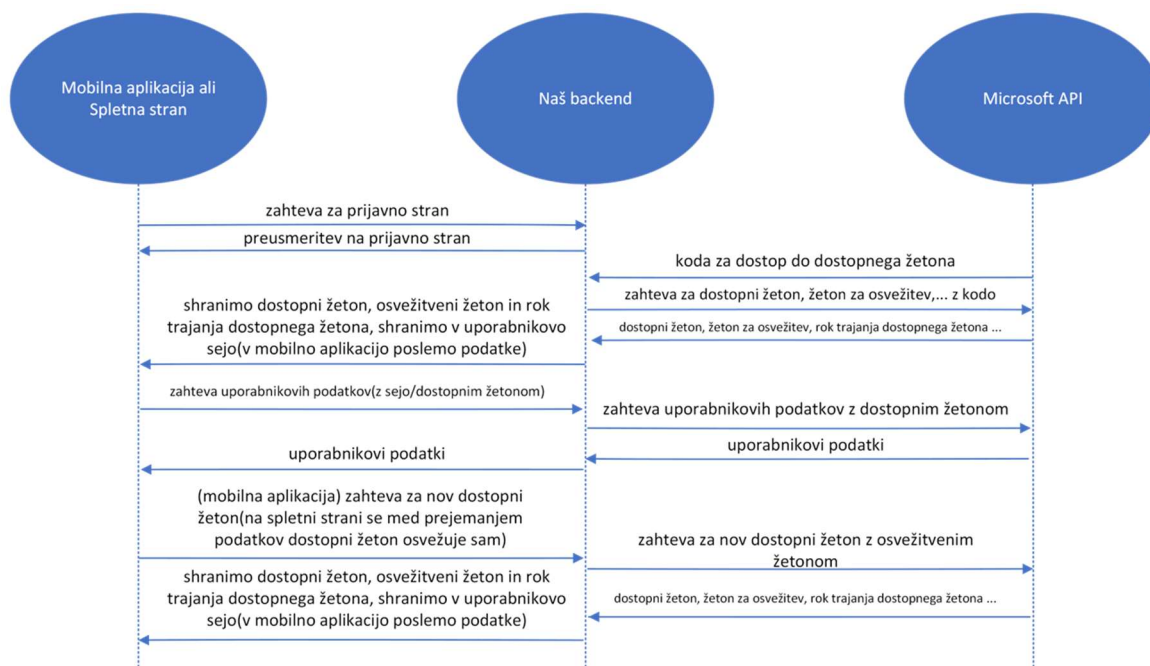
Zdaj lahko uporabnik uporablja šolski Microsoftov račun za prijavo v najino aplikacijo.

3.2.1 Registracija aplikacije na Microsoft Azure

Za implementacijo prijave z Microsoftovim računom morava najprej registrirati aplikacijo na portalu Microsoft Azure.

3.2.2 Uporaba Microsoft oauth2 v naši aplikaciji

Z registrirano aplikacijo pridobiva ID aplikacije in varnostni žeton, ki ju potrebujemo za delujočo prijavo z Microsoftovim računom. Z povezavo z Microsoftovim API-jem bova naredila zaledni del, ki bo povezal našo spletno stran in mobilno aplikacijo. Za zaledni del bova uporabila NestJS. Potem sva za zaledni del naredila diagram in kako bo vse skupaj delovalo.(Slika 8)



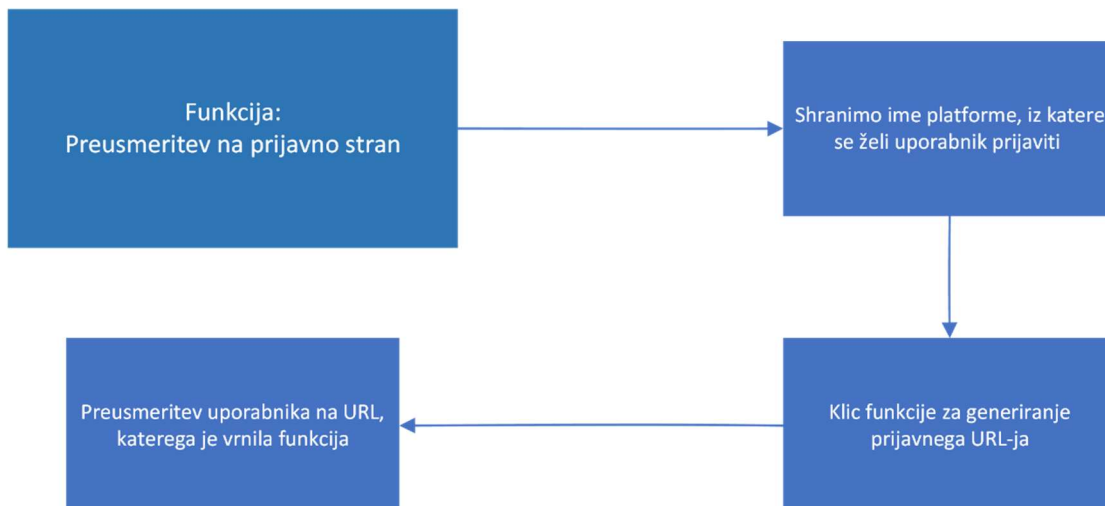
Slika 8: Diagram našega zalednega dela (Vir: lasten)

Po narejenem diagramu sva morala rešitev implementirati še v kodi. Vse pomembne konstante sva zapisala v »env« datoteko Slika 9.

```
1 OAUTH_APP_ID=927a3958-cac7-4c91-832c-81b128bf2b8f
2 OAUTH_APP_CLIENT_SECRET=hwY
3 OAUTH_REDIRECT_URI=http://localhost:5050/auth/redirect/
4 OAUTH_SCOPES='email offline_access openid profile user.read groupMember.read.all presence.readwrite user.readbasic.all people.read people.read.all '
5 OAUTH_AUTHORITY=https://login.microsoftonline.com/organizations/
6 OAUTH_ID_METADATA=/v2.0/.well-known/openid-configuration
7 OAUTH_AUTHORIZE_ENDPOINT=/oauth2/v2.0/authorize
8 OAUTH_TOKEN_ENDPOINT=/oauth2/v2.0/token
9 SESSION_SECRET=a
```

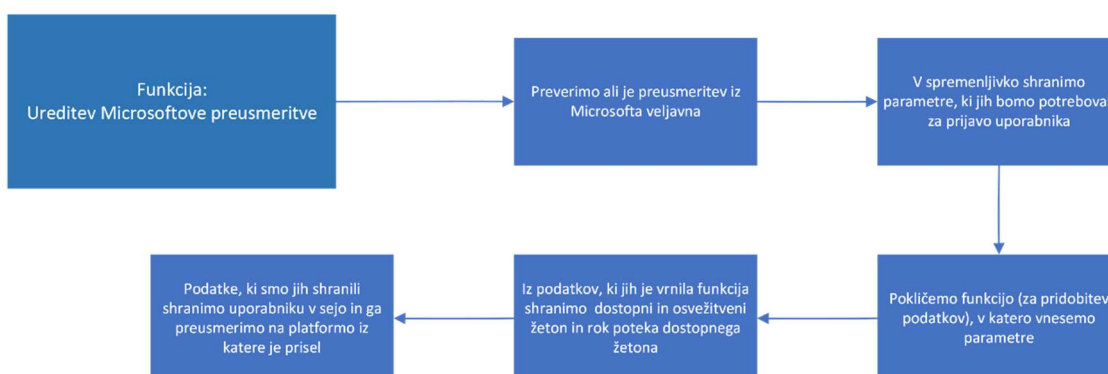
Slika 9: .env datoteka (Vir: lasten)

Potem sva morala urediti preusmeritev uporabnika na Microsoftovo prijavno stran. Za lažje razumevanje delovanja sva pripravila diagram na Slika 10.



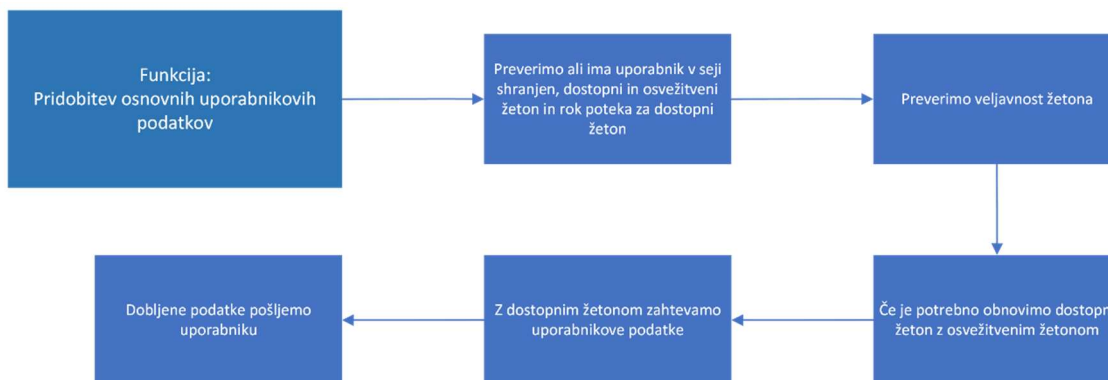
Slika 10: Diagram za preusmeritev uporabnika na prijavno stran (Vir: lasten)

Ko se uporabnik prijavi v Microsoftov račun, morava urediti Microsoftovo preusmeritev na zaledni del in to sva opisala s pomočjo diagram na Slika 11.



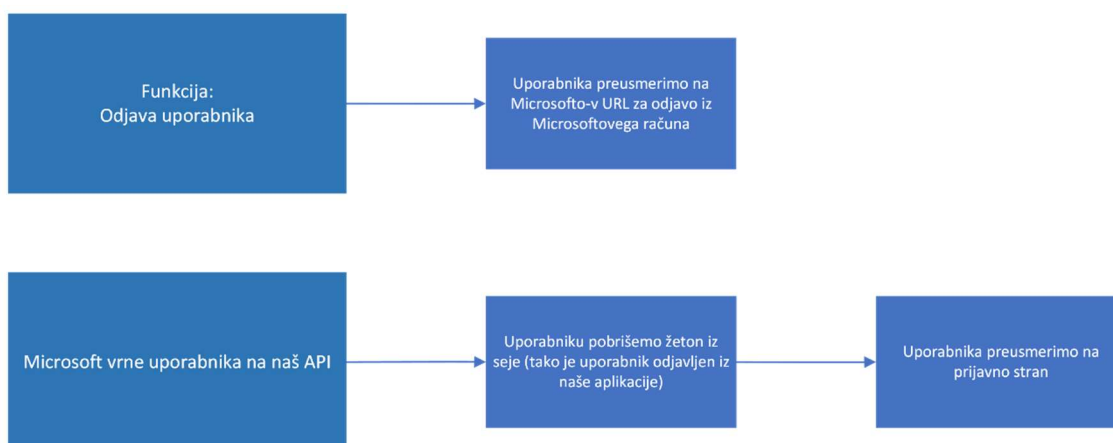
Slika 11: Diagram za ureditev Microsoftove preusmeritve (Vir: lasten)

Takoj je uporabnik prijavljen. Zdaj pa samo zahtevamo uporabnikove podatke, kot so ime, priimek, e-pošta, razred, šola, profilna slika, status na teamsih, urnik ... In to naredimo z več funkcijami. Primer diagrama za pridobivanje osnovnih uporabnikovih podatkov.



Slika 12: Diagram za pridobitev osnovnih uporabnikovih podatkov (Vir: lasten)

Ko se uporabnik želi odjaviti, imamo dve funkciji, najprej ga odjavimo iz Microsoftovega računa in potem nato izbrisemo sejo; za to sta zadolženi dve funkciji, ki ju bova prikazala z diagramom.



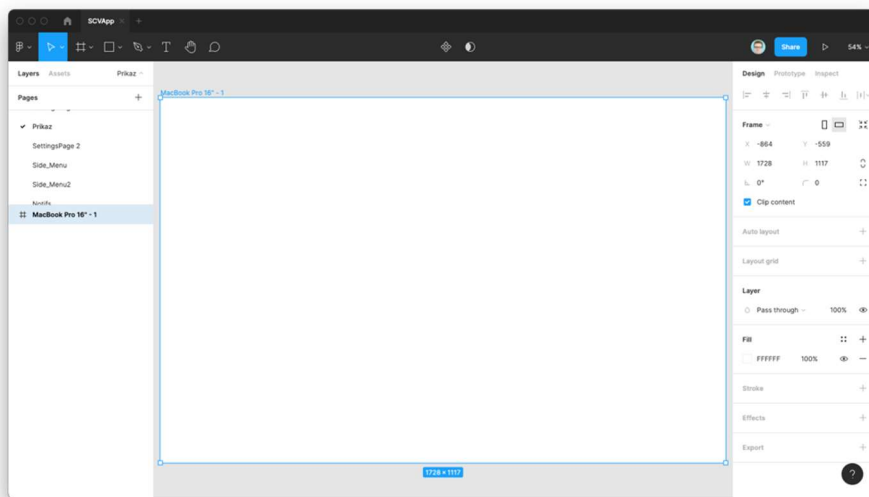
Slika 13: Diagram za odjavo uporabnika (Vir: lasten)

3.3 Izgled aplikacije

Ker želimo, da je najina aplikacija čim bolj dostopna in prijazna uporabnikom, je pomembno narediti tudi sprednji del spletne strani in mobilno aplikacijo. Za to morava najprej vedeti, kaj bodo prikazovale in kaj je najbolj ključno, kar dijak potrebuje; zato sva se na podlagi najinih izkušenj odločila, kaj bo aplikacija prikazovala, npr. kaj je sedaj na urniku, kje in kaj je naslednja ura, spletna stran šole, eAsistent, malice, Arnes učilnice ...

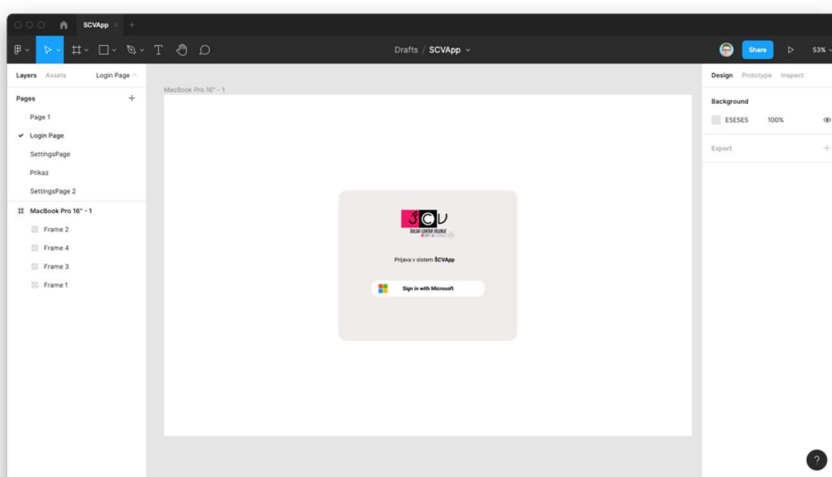
3.3.1 Izgled aplikacije v Figma

Ko sva vedela, kaj bo aplikacija prikazovala, sva morala narediti še izgled aplikacije. Za izgled aplikacije sva se odločila, da bova uporabila program Figma. V programu Figma ustvariva nov projekt in prazno predlogo, katere širina in dolžina je 1728 x 1117 px.



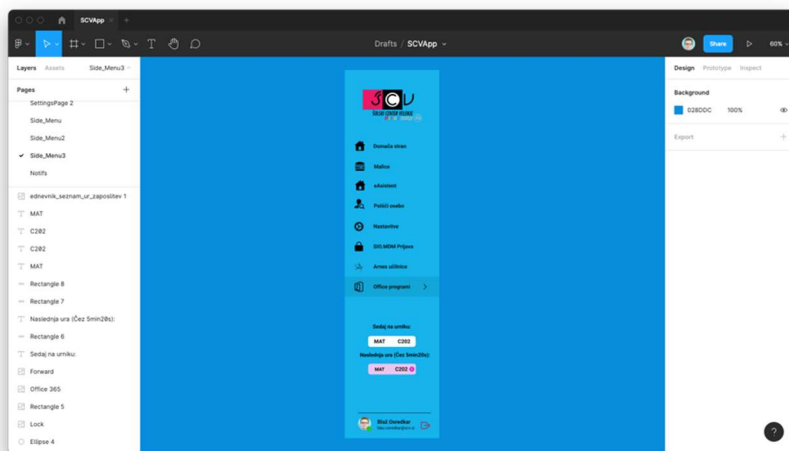
Slika 14: Osnovna predloga za izgled v aplikaciji Figma (Vir: lasten)

Začela sva z izgledom prijavnne strani, ki jo uporabnik, ko obiše najino spletno stran, vidi najprej.



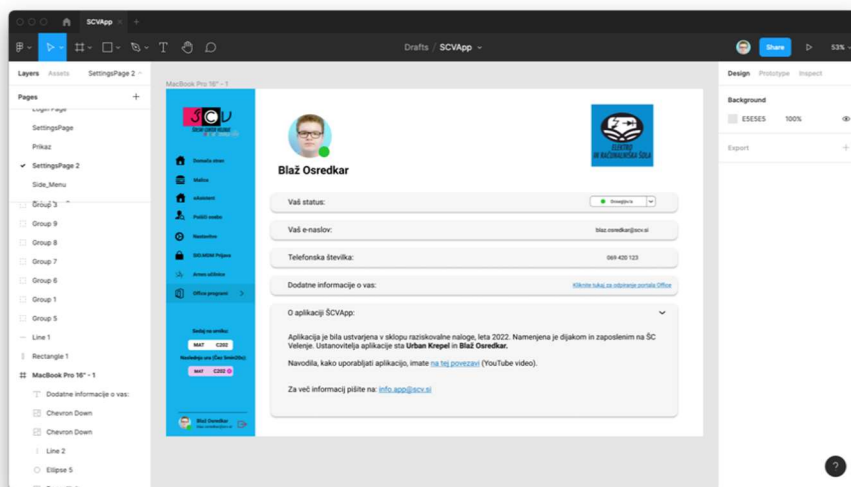
Slika 15: Izgled prijavnne strani izdelan v aplikaciji Figma (Vir: lasten)

Nato je sledil glavni meni najine aplikacije, ki se vedno nahaja na levi strani spletne aplikacije.



Slika 16: Slika izgleda stranskega menija, ustvarjenega v Figma (Vir: lasten)

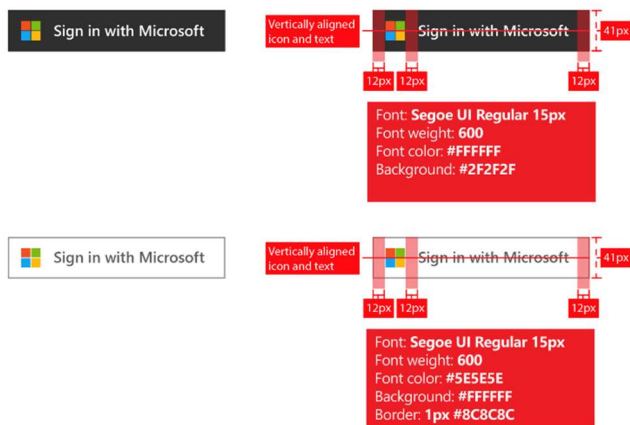
Temu je sledil izgled odseka nastavitve.



Slika 17: Izgled odseka nastavitve, ustvarjen v Figma (Vir: lasten)

3.4 Prevajanje gumba »Login with Microsoft« v slovenščino

Za najino spletno stran sva želela, da bi za lažjo orientacijo bil gumb »Login with Microsoft« v slovenskem jeziku. Microsoft to dovoljuje, vendar so neke stvari, na katere je potrebno biti pozoren, kot na primer oddaljenost teksta od loga; dovoljena je samo sprememba teksta v drug jezik s tem, da ne sme izgubiti svojega prvotnega pomena.



Slika 18: Prikaz, kako uporabljati gumb "Login with Microsoft" (Vir: Microsoft)



Slika 19: Urejen gumb "Vpis z Microsoft računom". (Vir: lasten)

3.5 Platforme aplikacije

Po končanem izgledu sva svojo idejo spravila v kodo.

3.5.1 Čelni del spletne aplikacije

Za izdelavo čelnega dela sistema sva uporabila Javascript knjižnico React. Najprej moramo narediti usmerjevalne spletne poti, ki bodo vodile na različne dele spletne strani, kot so prijava, domača stran, eAsistent ... To sva naredila z uporabo knjižnice, imenovane »react-router-dom«.


```
import HomePage from "../homePage/homePage";
import LoginPage from "../loginPage/loginPage";
import React from "react";
import { BrowserRouter, Routes, Route } from "react-router-dom";
import NotFoundPage from "../404page/404page";

import AboutPage from "../aboutPage/aboutPage.js"

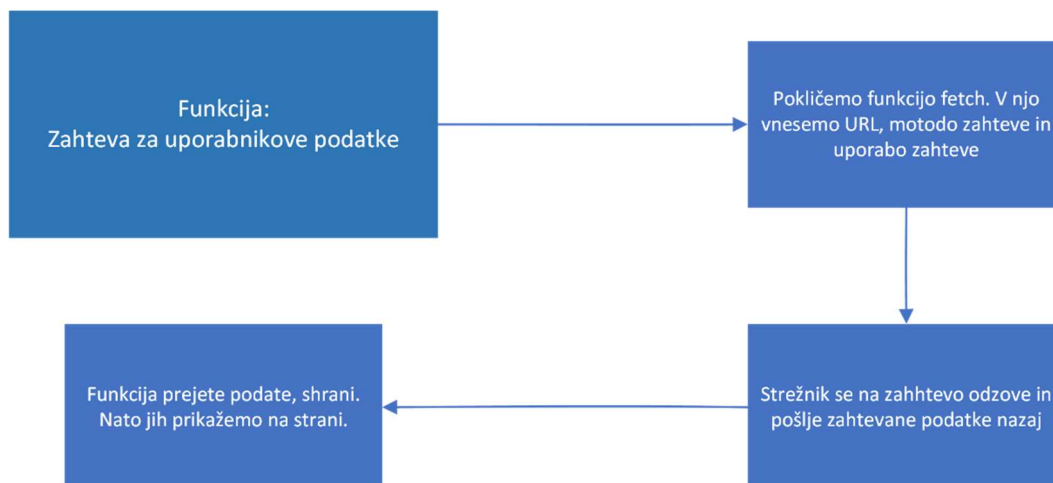
const RoutePage = () => {
  return (
    <BrowserRouter>
      <Routes>
        <Route path="/">
          <Route index element={<HomePage />} />
          <Route path="/domov" element={<HomePage />} />
          <Route path="/malice" element={<HomePage />} />
          <Route path="/easistent" element={<HomePage />} />
          <Route path="/nastavitve" element={<HomePage />} />
          <Route path="/arnes-ucilnice" element={<HomePage />} />
          <Route path="/prijava" element={<LoginPage />} />
          <Route path="/o-nas" element={<AboutPage />} />
          <Route path="*" element={<NotFoundPage />} />
        </Route>
      </Routes>
    </BrowserRouter>
  )
}

export default RoutePage
```

Slika 20: React komponenta za urejanje spletnih poti (Vir: lasten)

Kot varnostno opcijo sva dodala pot z * pa sva označila, če nobena od zgornjih poti ni bila vpisana, da uporabniku prikaže stran, ki uporabnika obvešča, da takšna stran ne obstaja.

Ko sva uredila spletne poti, sva morala narediti strani; začela sva s prijavo. Prijavno stran sva samo uredila in dodala gumb za prijavo, ki vodi do URL-ja našega zalednega dela za prijavo. Potem sva naredila domačo stran, ki vključuje stranski meni in glavni del strani. Za pridobivanje podatkov iz zalednega dela sva uporabila funkcijo »fetch«, ki je že vključena v javascript. Prikaz delovanja je spodaj Slika 21.



Slika 21: Diagram za zahtavo uporabnikovih podatkov (Vir: lasten)

Potem sva te podatke prikazala na spletni strani in to sva naredila tako, da sva v zavite oklepaje vnesla spremenljivko, kot je prikazano spodaj (Slika 22).

```
<p>{props.userData.displayName}</p>
<span>{props.userData.mail}</span>
```

Slika 22: Primer vnosa uporabnikovih podatkov na spletno stran (Vir: lasten)

Potem sva naredila različne strani na domači strani, kot so šolska spletna stran, nastavitve, malice ...

Za spletne strani sva uporabila »iframe«, kot je prikazano spodaj Slika 23.

```
export default function SchoolPage(props){//Prikaz šolske spletne strani
  ... return(
  ... <iframe src={props.url/* URL šolske spletne strani*/} style={{width:'100%',height:'100%',border:'none'}} title="Šolska spletna stran">
  ... </iframe>
  ... )
}
```

Slika 23: Prikaz šolske spletne strani (Vir: lasten)

Ostale strani, kot so nastavitve, pa sva naredila sama.

Za premikanje med stranmi sva uporabila komponento (komponenta je majhen izoliran kos kode) »Link«, ki te pelje na drugo stran, ampak naloži samo del strani, ki se spremeni. (To se uporablja namesto komponente a.)

```
<Link to={this.props.href||""}>
  <div className="link">
    <div className="icon">{this.props.children}</div>
    {
      this.props.size >= 300 ? <span className="item">{this.props.name}</span> : ""
    }
  </div>
</Link>
```

Slika 24: Komponenta »Link« (Vir: lasten)

Da sva prikazovala različne stani na različnih spletnih poteh, sva uporabila »switch« stavek, ker sva potem samo vrnila komponento, ki je namenjena na tisti strani.

```
export default function MainPage(props){//Prikaz določene komponente glede na spletno pot
... let location = useLocation();//Dobimo URL

... let pathname = location.pathname//Dobimo pot po imenu domene in porta npr. .../domov

...//Preverimo ali se ime poti ujema z besedo in temu ustrezno prikazemo komponento
... switch(pathname){
... case "/domov":
...     return (
...         <div style={props.style}>
...             {
...                 <SchoolPage url={props.userData.school&props.userData.school.schoolUrl}/>
...             }
...         </div>
...     )

... case "/malice":
...     return (
...         <div style={props.style}>
...             <MalicePage url={props.maliceUrlLink}/>
...         </div>
...     )

... case "/easistent":
...     return (
...         <div style={props.style}>
...             <EasistentPage url={props.eAUrlLink}/>
...         </div>
...     )

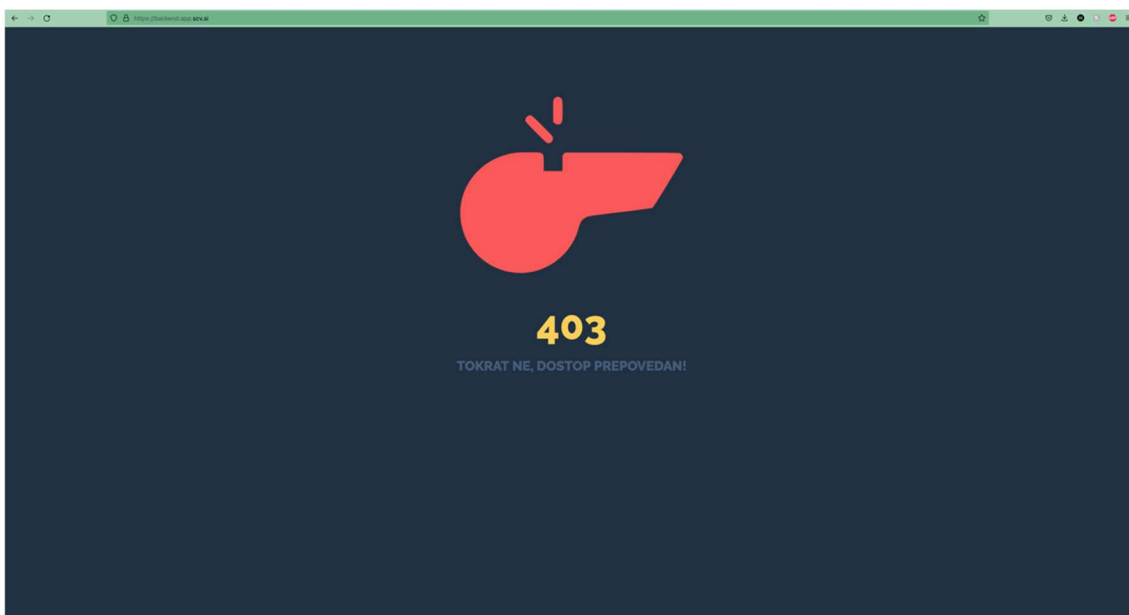
... case "/nastavitve":
...     return(
...         <div style={props.style}>
...             <SettingsPage userData={props.userData}/>
...         </div>
...     )

... case "/arnes-ucilnice":
...     return(
...         <div style={props.style}>
...             <ArnesUcilnicePage />
...         </div>
...     )
...     default:
...         return (
...             <div style={props.style}>
...                 <SchoolPage url={props.userData.school&props.userData.school.schoolUrl}/>
...             </div>
...         )
...     }
}
```

Slika 25: Prikaz določene komponente glede na spletno pot (Vir: lasten)

3.6 Gostovanje in domena aplikacije

Ker je aplikacija na spletu, sva potrebovala domeno in gostovanje. Za domeno sva uporabila app.scv.si, katero nama je pridobil najin mentor Iztok Osredkar. Za gostovanje spletne strani uporabljava cPanel, ki ga gostijo šolski strežniki. Za gostovanje zalednega dela uporabljava cPanel z NodeJS storitvijo, ki pa je gostovana na Neoserv strežnikih. Domena za zaledni del je backend.app.scv.si. Preden sva pa zaledni del prestavila na podomeno backend.app.scv.si, sva za gostovanje uporabljala Heroku.



Slika 26: Spletna stran zalednega dela (Vir: lasten)

4 REZULTATI

4.1 Prijava s šolskim Microsoftovim računom

Z implementacijo prijave s šolskim Microsoftovim računom ni bilo veliko težav. Imela sva eno težavo, da si uporabnik ni mogel izbrati šolskega Microsoftovega računa za prijavo, ampak se je prijavil z tistim, s katerim je že bil prijavljen. Ta problem sva odpravila tako, da sva v funkcijo za generiranje URLja za prijavo dodala Slika 27 to vrstico kode.

```
prompt: 'select_account'//Funkciji za generiranje vnesemo ta podatek, da si lahko uporabnik zmeraj izbere profil
```

Slika 27: Podatek, da si lahko uporabnik zmeraj izbere račun za prijavo

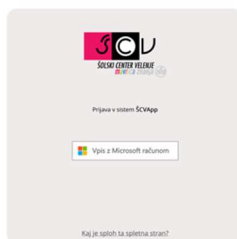
Edini večji problem je bil, da se v spletno aplikacijo nisi mogel prijaviti zaradi različnih nastavljanj piškotkov na spletno stran iz zalednega dela. Ta problem je bil zaradi varnosti, saj ne moreš nastavljati piškotkov iz druge domene, kot je spletna stran; zato sva to težavo rešila tako, da sva zaledni del prestavila na poddomeno in zdaj deluje tako, kot mora. Tudi pri odjavi sva naredila, da uporabnika najprej odjavi iz Microsoftovega računa in nato še počisti sejo.

4.2 Platforme aplikacije

4.2.1 Spletna platforma

Spletna stran se nahaja na <https://app.scv.si> in je na voljo vsem dijakom in zaposlenim na Šolskem centru Velenje. Najina spletna stran vsebuje:

- prijavo z Microsoftovim računom,



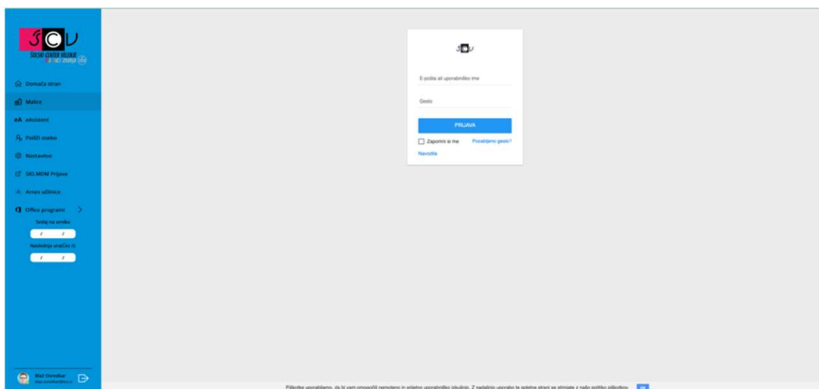
Slika 28: Prijavna stran ŠCVApp-a (Vir: lasten)

- spletno stran šole, ki jo dijak oz. učitelj obiskuje (v primeru spodaj je ERŠ);



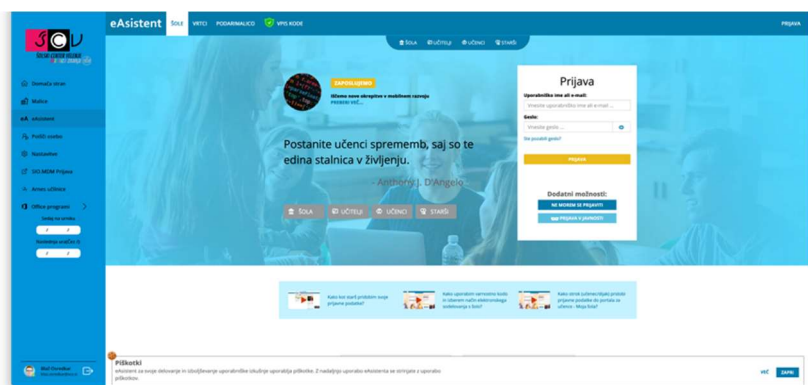
Slika 29: Spletna stran šole (Vir: lasten)

- spletni portal malice,



Slika 30: Spletni portal malice na spletnem mestu ŠCVApp (Vir: lasten)

- spletni portal eAsistent,



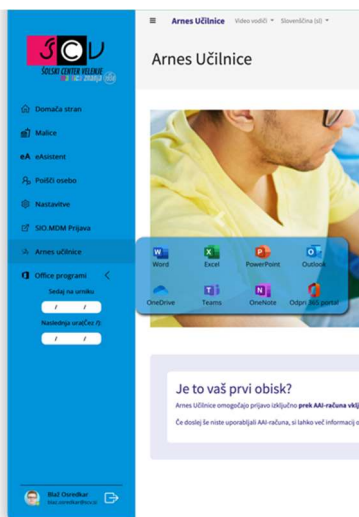
Slika 31: Spletni portal eAsistent na spletnem mestu ŠCVApp (Vir: lasten)

- portal Poišči osebo (ki je še v izdelavi),
- zavihek Nastavitve,



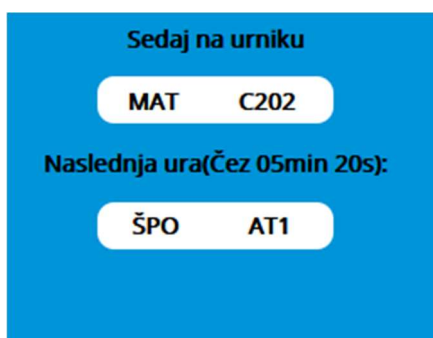
Slika 32: Zavihek nastavitve na spletnem mestu ŠCVApp (Vir: lasten)

- SiO.MDM prijavo,



Slika 35: Orodja Office (Word, Excel ...), (Vir: lasten)

- prikaz ure, ki jo ima dijak takrat na urniku.



Slika 36: Prikaz ur na spletnem mestu ŠCVApp (Vir: lasten)

Na koncu menija je tudi prikaz dijakove profilne slike, aktivno stanje, ki se osvežuje preko Office programov, imena in priimka, naš šolski e-naslov in gumb za odjavo.



Slika 37: Prikaz profila na spletnem mestu ŠCVApp (Vir: lasten)

4.2.2 Mobilna platforma

Mobilna aplikacija je bila prvotno narejena v Android Studiu za platformo Android in v Xcode-u za platformo iOS in zdaj jo bova naredila v Flutterju, da bo imela funkcionalnosti takšne kot spletna platforma.

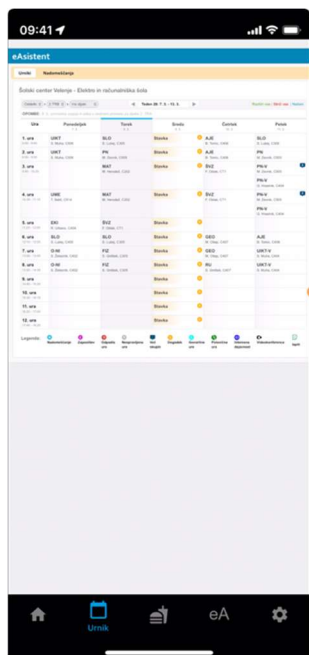
V prvotni aplikaciji je vključena:

- spletna stran šole, ki jo je uporabnik izbral;



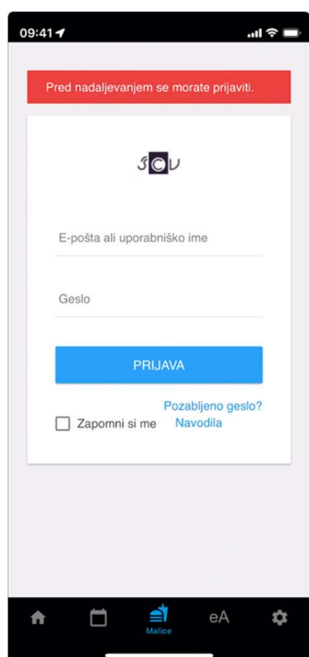
Slika 38: Spletna stran šole (Vir: lasten)

- urnik, ki si ga dijak izbere sam;



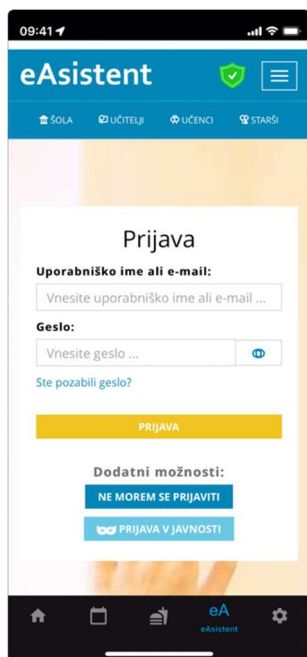
Slika 39: Urnik (Vir: lasten)

- spletni portal malice,



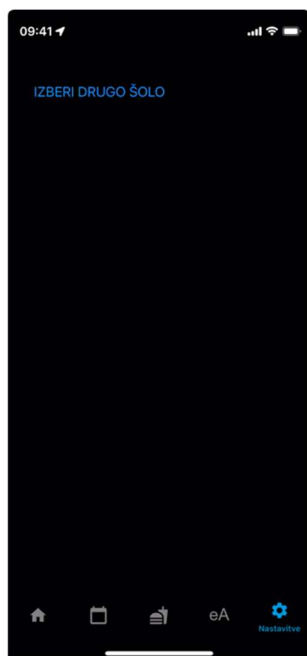
Slika 40: Spletni portal malice (Vir: lasten)

- spletni portal eAsistent,



Slika 41: Spletni portal eAsistent (Vir: lasten)

- nastavitve,



Slika 42: Nastavitve v aplikaciji (Vir: lasten)

V nastavitvah je zavihek, kjer si lahko uporabnik izbere šolo.



Slika 43: Izbor šole (Vir: lasten)

Nova verzija aplikacije bo vključevala takšne funkcije, kot so na spletni platformi (npr. prijavo s šolskim Microsoft računom, trenutno uro ...).

4.2.3 Posodobljena mobilna verzija aplikacije

Na podlagi povratnih informacij sva aplikacijo nadgradila. Nadgrajena aplikacija nas naprej pozdravi z nekaj osnovnimi navodili, kaj vsebuje in kako pridobimo dostop do nje. Ko se prijavimo (z Microsoftovim računom) nam aplikacija najprej prikaže spletno stran šole, ki jo obiskujemo, barvo šole, urnik razreda ter naše podatke, kot so ime, priimek ter naš šolski e-poštni naslov.

Aplikacija vsebuje:

- uvod v aplikacijo, ki vsebuje osnovna navodila, kako dostopati do funkcij aplikacije, ter kakšen je namen aplikacije;



Slika 44: Prva stran uvodnega dela aplikacije v aplikaciji ŠCVApp (Vir: lasten)

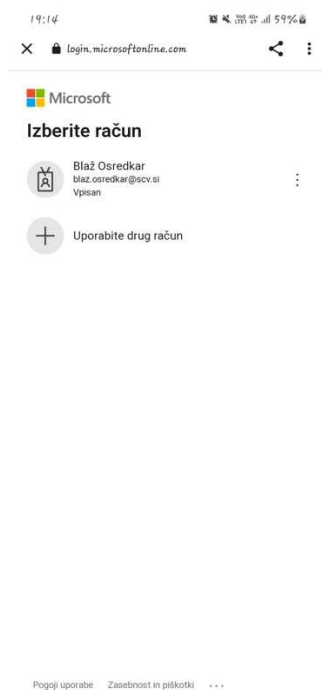


Slika 45: Druga stran uvodnega dela aplikacije v aplikaciji ŠCVApp (Vir: lasten)



Slika 46: Tretja stran uvodnega dela aplikacije v aplikaciji ŠCVApp (Vir: lasten)

- v aplikacijo se nato prijavimo s šolskim Microsoft računom.



Slika 47: Prijava z šolskim Microsoft računom v aplikaciji ŠCVApp (Vir: lasten)

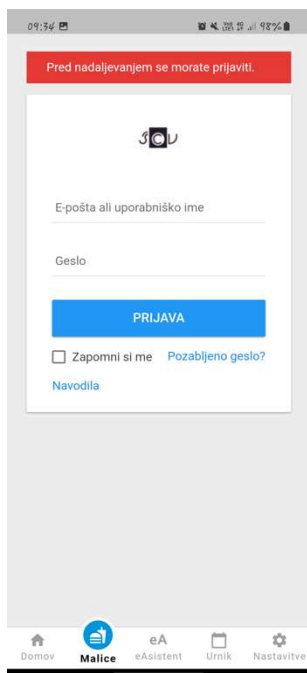
Po prijavi imamo polni dostop do funkcij aplikacije. Med to se šteje:

- domača stran šole, ki jo obiskujemo;



Slika 48: Domača stran šole v aplikaciji v aplikaciji ŠCVApp (Vir: lasten)

- spletni portal šolskih malic,



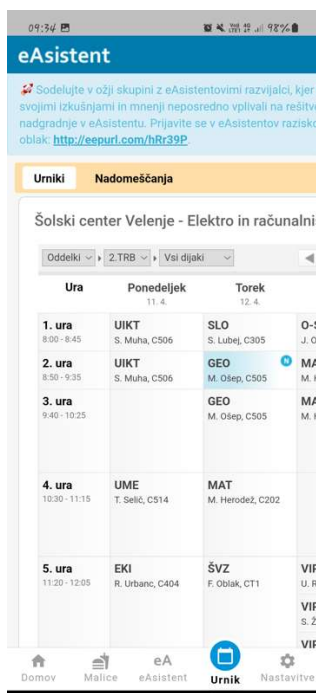
Slika 49: Spletni portal malic v aplikaciji ŠCVApp (Vir: lasten)

- spletni portal eAsistent,



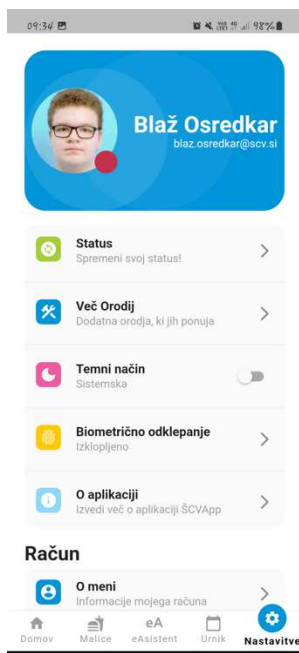
Slika 50: Spletni portal eAsistent v aplikaciji ŠCVApp (Vir: lasten)

- urnik za razred, ki ga obiskujemo;



Slika 51: Urnik razreda, ki ga obiskujemo v aplikaciji ŠCVApp (Vir: lasten)

- zavihek nastavitve, kjer vidimo svoje osebne podatke, kot so ime, priimek, šolski e-poštni naslov.



Slika 52: Zavihek nastavitve v aplikaciji ŠCVApp (Vir: lasten)

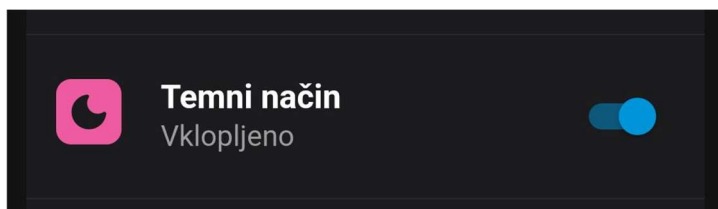
Pod zavihkom nastavitve lahko:

- zamenjamo svoj status, ki se posodobi tudi v vseh aplikacijah Office 365;



Slika 53: Menjava statusa v aplikaciji ŠCVApp (Vir: lasten)

- menjavamo svetle teme v temno (privzeta tema je sistemska);



Slika 54: Vklop temnega načina (Vir: lasten)

- vklopimo biometrično odklepanje (ob vklopu pomeni, da se moramo ob vstopu v aplikacijo avtorizirati z biometričnim sistemom telefona (prstnim odtisem, prepoznavo obrazom ali s pin številko).



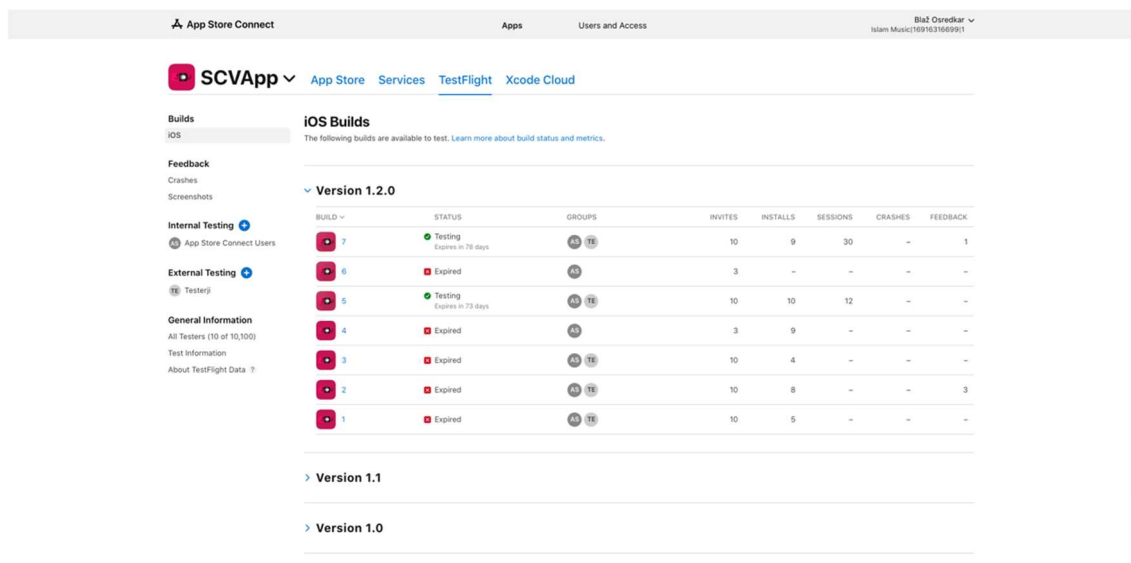
Slika 55: Nastavitev biometričnega odklepanja v aplikaciji ŠCVApp (Vir: lasten)

4.3 Uvajanje preizkušne različice programa za mobilno aplikacijo

Ker sva želela izvedeti kaj o najini prenovljeni aplikaciji menijo najini prijatelji in profesorji, sva se odločila ustvariti razvojno različico za prenovljeno aplikacijo ŠCVApp.

4.3.1 Razvoj preizkušne različice za iOS naprave

Za platformo iOS sva ustvarila preizkušnjo preko App Store Connect, ki omogoča razvoj preizkusne različice preko TestFlight. S tem lahko objavlja več preizkusnih različic aplikacije, ki jo nato določeni testatorji, prijavljeni z računom, ki so nama ga podali za namene dostopa do prej omenjene preizkušnje, testirajo.

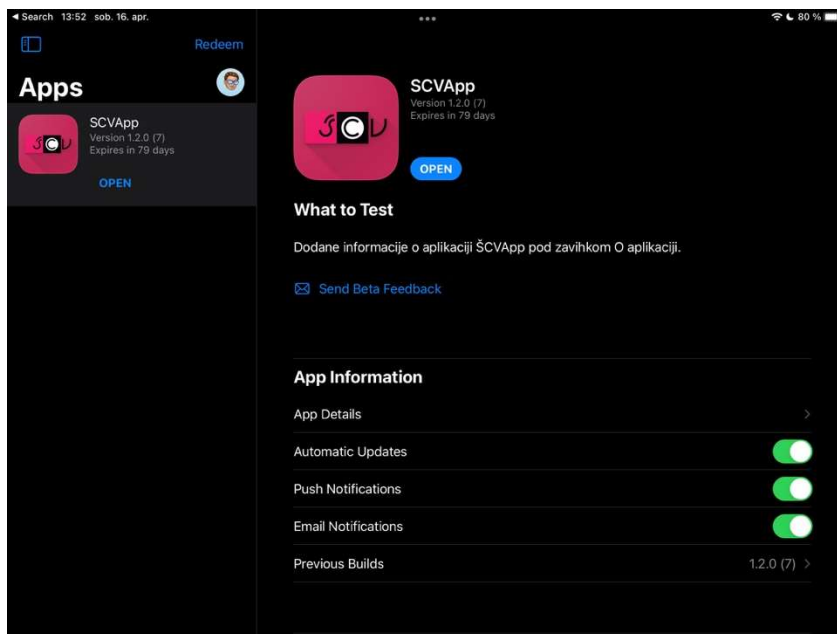


Slika 56: Prikaz TestFlight zavihka v App Store Connect-u (Vir: lasten)

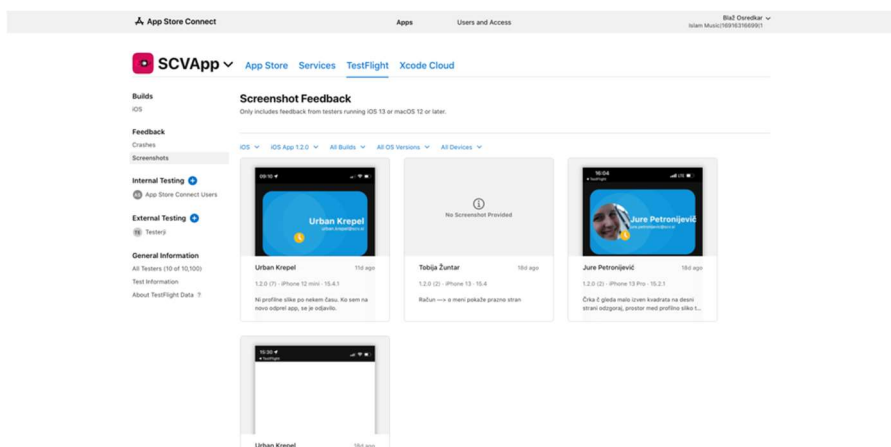
Vstopanje v preizkušnjo in nalaganje le-te na našo napravo za naprave iOS je preprosto.

1. Uporabnik nama poda e-poštni naslov, s katerim je prijavljen v App Store na svoji napravi.
2. Ta e-poštni naslov nato vpiševa v listo e-poštnih naslovov.
3. Uporabnik prejme e-pošto z navodili, kako naložiti aplikacijo.

Uporabnik v 3. koraku naloži aplikacijo TestFlight, preko katere lahko nato pošilja povratne informacije, ki jih midva nato vidiva v portalu.



Slika 57: Aplikacija TestFlight (Vir: lasten)



Slika 58: Povratne informacije preizkuševalcev (Vir: lasten)

4.3.2 Razvojna različica za Android naprave

Ko sva končala z izdelavo razvojne različice za iOS platformo, sva se odločila narediti podobno za naprave Android. Za ustvarjanje preizkušnje na platformi Android sva ustvarila zaprto preizkušnjo, ki da dostop do razvojne različice le tistim, ki so prijavljeni v Play trgovino z računom, ki so ga nama podali.

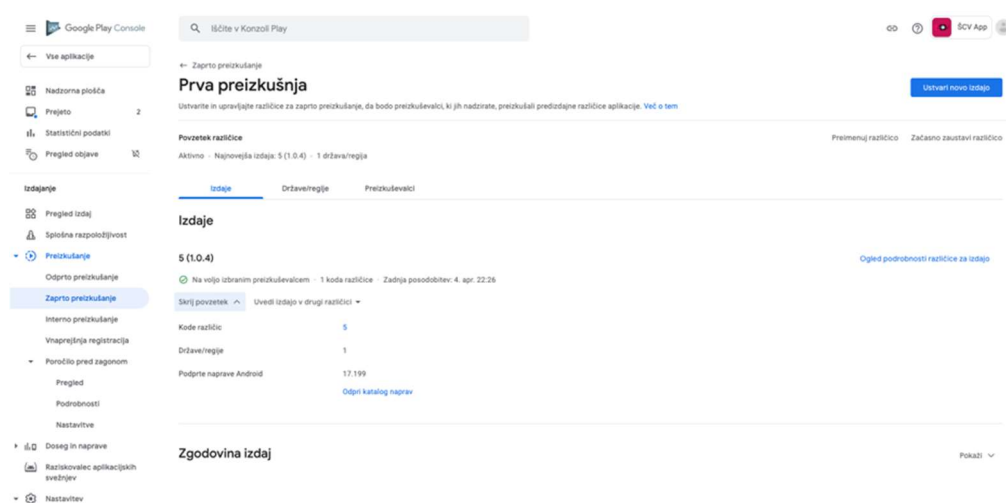
Vstop v preizkušnjo z uporabniške strani je podoben kot pri iOS.

1. Uporabnik nama poda e-poštni naslov, s katerim je prijavljen v Play trgovino na svoji napravi
2. Ta e-poštni naslov nato vpiševa v listo e-poštnih naslovov.
3. Uporabniku sporočiva, da sva ga uspešno dodala v preizkušnjo.
4. Uporabnik na Play Store poišče aplikacijo ŠCVApp in se pridruži preizkušnji.
5. Uporabnik nato počaka nekaj minut ter si nato naloži razvojno različico programa.



Slika 59: Uporabnik, ki je pridružen preizkušnji na Android platformi (Vir: lasten)

Izdaje preizkušenj ter preizkuševalce za naprave Android urejava na spletni strani, poimenovani Play Console.



Slika 60: Spletna stran Play Console, kjer uvajamo zaprto preizkušnjo (Vir: lasten)

4.4 Težave

4.4.1 Težave čelnega dela aplikacije

Kot sva že prej omenila, za prikazovanje drugih spletnih strani na najini spletni strani uporabljava komponento »iframe«. Zaradi vse večje varnosti brskalnikov, nekateri brskalniki (kot so Google Chrome, Safari, Microsoft Edge, Opera) onemogočajo shranjevanje piškotkov 3. entitet (angl. third party cookie), zaradi tega se na najini spletni strani ne morete prijaviti v nekatera orodja (kot sta eAsistent in Arnes Učilnice). To bi lahko rešila na dva načina. Prvi način je, da bi spletno stran orodja odprla v drugem zavihku, vendar bi s tem izgubila »vsa orodja na enem mestu«. Drugi način je, da bi skrbnik spletne strani orodja, ki ga prikazujeva, dovolil, da bi se njihova spletna stran prikazala na njihovi spletni strani v »iframe«, zato sva raje izbrala drugo opcijo. Za orodje malice sva za dogovorila z najinim učiteljem, da je dovolil najino domeno. Za orodje eAsistent in Arnes Učilnice pa sva pa kontaktirala s podjetjema in ju prosila za dovoljenje za našo domeno.

4.4.2 Težave zalednega dela aplikacije

Pri varnostni žetonov, sva naletela na napako v najini kodi. Ob pošiljanju osvežitvenih žetonov za prijavo sva uporabnika prijavila kot napačno osebo. To sva nadgradila tako, da sva preverjala tudi, komu sva poslala kateri osvežitveni žeton. Zdaj osveževanje deluje pravilno.

4.4.3 Težave z objavo aplikacije za naprave Android

Kot omenjeno, je aplikacija že objavljena na Play Store in App Store. Za objavljanje aplikacije na Play Store je potrebno imeti shranjen ključ, preko katerega se koda tiste verzije aplikacije indentificira, da je aplikacija legitimna. Med tem časom se nama je zgodilo, da sva ta ključ izgubila in morala sva poslati zahtevo za nov ključ. Celoten postopek za pridobitev novega ključa je trajal okoli tri dni. Po tem času je bilo potrebno počakati še dva dni, da se je nov ključ aktiviral.

5 RAZPRAVA

Rezultati so najina pričakovanja potrdili. Hipoteze so bile sledeče:

1. hipoteza: S pomočjo šolskega Microsoft računa lahko ustvarimo prijavo na našo aplikacijo.

»Login with Microsoft« obstaja na večjih spletnih mestih že nekaj časa. Uporablja se za prijavo v marsikatero spletno aplikacijo. Na Šolskem centru Velenje šolski računi potekajo preko Microsoftovih računov, zato je bila implementacija te prijave pametna odločitev. Ta odločitev je tudi zelo varna, saj za gesla teh šolskih računov skrbi Microsoft, kar nama v tem primeru malce olajša delo. Ker sva rešitev uspešno implementirala v najino aplikacijo lahko to hipotezo potrdiva.

2. hipoteza: V najino aplikacijo bova lahko vključila še 10 ostalih orodij, ki so potrebna za šolanje.

Za šolanje v tako imenovani dobi tehnologije potrebujemo, oz. uporabljamo veliko spletnih orodij različnih podjetij (eAsistent, Arnes ...). Najin cilj je bil povezati vsa ta orodja v enotno orodje, da je lahko vse na enem mestu in si s tem olajšamo iskanje ter pridobimo nekaj dragocenega časa, ki bi ga drugače zapravili za iskanje teh orodij drugje po spletu. Implementacija orodij, kot so eAsistent, Arnes učilnice, SIO MDM, je v naši aplikacij bila uspešna, vendar ni bila preprosta, saj je prijava oz. "session" ostala na brskalniku in tako bi lahko uporabnik, ki bi prišel za prejšnjim, videl in urejal podatke prejšnjega, zato sva pri odjavi iz najine aplikacije dodala poziv odjavnih strani teh ponudnikov. V aplikacijo sva dodala 10 orodij (spletno stran šole, portal za malice, orodje eAsistent, poišči osebo, nastavitve, SIO.MDM prijavo, Arnes učilnice, prikaz urnika ter programe, ki so vključeni v paket Office). S tem lahko najino hipotezo potrdiva.

3. hipoteza: Spletna verzija ŠCVApp-a bo imela podporo na vseh brskalnikih.

Poznamo več modernih brskalnikov in vsak dan nas večina uporablja vsaj enega izmed njih. Vsi ti brskalniki imajo na privzeto izklopljeno “cross-site tracking”, ki prepričuje uporabo piškotkov ene spletne strani z drugo. Najina začetna spletna aplikacija je imela glavno domeno na domeni .scv.si, vendar se je najin zaledni del nahajal na scvapp-backend.herokuapp.com. Zaradi tega je v začetni fazi spletna stran delovala le, če si to zaščito izklopil, vendar je tudi to delovalo le na brskalnikih Firefox in Safari. V nadaljnjem delu raziskovanja sva zaledni del prestavila na poddomeno backend.app.scv.si, kar sedaj dovoljuje dostop do te spletne strani na vseh brskalnikih, ki jih uporabljamo danes. S tem to hipotezo potrjujeva.

4. hipoteza: Z uporabo programskega ogrodja Flutter bova pridobila podporo na iOS in Android platformo.

Dandanes imamo že skoraj vsi mobilni telefon. Uporabljamo ga več ur na dan za komuniciranje, zabavo in šolo. Nanj nalagamo aplikacije, ki nam olajšajo delo ali skrajšajo čas. Najina prvotna ideja temelji na najini že narejeni aplikaciji za mobilne naprave, katerih operacijski sistem je Android ali iOS, vendar sva razvoj aplikacij imela razdeljen tudi na dve platformi programskega jezika in orodja (iOS – Xcode in Swift; Android – Android Studio in Java). Z uporabo ogrodja Flutter, ki temelji na arhitekturi programskega jezika Dart, združimo ti dve platformi v eno samo. Tako imamo aplikacijo narejeno za iOS kot za Android z enim programskim jezikom. Prenos aplikacije iz dveh platform na eno je sicer bila na začetku težka, saj se je bilo potrebno naučiti novega programskega jezika Dart. Aplikacijo sva v tem času naredila do točke, ko lahko dijak izbere šolo in ima vse osnovne stvari, ki jih je imel v prejšnji verziji, ki je bila ustvarjena v drugem jeziku. S tem lahko to hipotezo potrdiva.

6 ZAKLJUČEK

V prvem delu raziskovanja sva se poglobila v izdelavo zalednega dela, ki ga uporabljamo za obdelavo (v katerem razredu je uporabnik, katero uro ima na urniku) in pošiljanje podatkov o uporabniku, ki jih potrebujemo za spletno stran in mobilno aplikacijo. Pri tem sva naletela na ovire, kot je delovanje spletne strani samo na brskalniku Firefox.

V drugem delu sva se posvetila spletni strani in njenemu izgledu. S programom Figma sva najprej naredila nekakšen predogled, kako bo izgledalo na spletni strani. Ko sva bila zadovoljna z izgledom, sva ga spravila na spletno stran. Pri tem sva naletela na problem z ujemanjem barv, saj vsak monitor prikazuje barve malce drugače in se nama odtenki niso ujemali.

Ustvarila sva tudi nadgradnjo že prej izdelane mobilne aplikacije. Aplikacijo sva predstavila na orodje Flutter, ki uporablja jezik Dart in s tem pridobila podporo na obeh platformah (iOS, Android) hkrati. Aplikaciji sva dodala prijavo z šolskim Microsoft računom za polni dostop do aplikacije. S to prijavo aplikacija izve, iz katere šole si ter kateri razred obiskuješ, zato se barva ikon aplikacije ter urnik nastavijo sami. Dodala sva tudi spremembo statusa, ki se ob spremembi osveži na vseh Office programih, ter dodala sva možnost odklepa aplikacije z biometriko. Dodala sva tudi možnost menjave teme med svetlo in temno. Posodobljeno verzijo aplikacije sva dodala v preizkušnjo svojim prijateljem in učiteljem.

7 POVZETEK

“Obvezna« oprema” današnjega dijaka je telefon. Uporabljamo ga več ur na dan za komuniciranje, zabavo in šolo. Sedaj, ko se svet vse bolj digitalizira, so vse informacije na spletu. Prednost tega je, da ljudi lažje obveščamo. Pri tem šola ni nobena izjema. Ideja najine raziskovalne naloge je bila izdelati in nadgraditi mobilno aplikacijo za dijake Šolskega centra Velenje, kjer je vse, kar je potrebno za šolo, na enem mestu. Poimenovala sva jo ŠCVApp. Aplikacija vsebuje informacije, potrebne za šolanje na ŠCV Velenje. Dijak z njeno pomočjo pregleduje urnik in novice s spletne strani, izbira malico ... Želela sva izdelati aplikacijo, ki je dijakom pri iskanju informacij glede šolanja v pomoč.

8 ZAHVALA

Vesela sva, da se je ta raziskovalna naloga odvila, kot sva si želela. Najprej bi se zahvalila svojima mentorjema Islamu Mušiču in Iztoku Osredkarju za vso pomoč pri delu. Nato bi se rada zahvalila gospodu Petru Vrčkovniku za slikovno gradivo. Zahvalila bi se rada tudi učiteljici Sonji Lubej za pomoč pri lektoriranju raziskovalne naloge ter učiteljici Beti Tomic za lektoriranje angleške verzije avtorskega izvlečka. Hvala tudi gospodu Samo Železniku za pomoč pri spletnem portalu malic in odjavah iz spletnih strani. Zahvalila bi se tudi vsem preizkuševalcem, ki ste nama pomagali pri razvojni različici aplikacije.

9 PRILOGE

```
@Get('/authUrl')
@Redirect('/', 302)
getAuthUrl(@Headers() headers) // Funkcija za preusmeritev na prijavn URL
let referer = headers.referer
let name = "app"
switch(referer) { // Preverjanje iz katere platforme se želi uporabnik prijaviti v aplikacijo (za kasnejše potreba)
  case 'http://localhost:3000/':
    name = "localhost"
    break;
  case 'http://app.scv.si/':
    name = "appscv"
    break;
  case 'https://app.scv.si/':
    name = "appscv"
    break;
}
let state = `${name}`
return this.loginService.getAuthUrl(state) // Pokličemo funkcijo za generiranje preusmeritvenega URL-ja in nato vrnemo preusmeritveni URL
}
```

Slika 61: Funkcija za preusmeritev uporabnika na prijavno stran (Vir: lasten)

```
async getAuthUrl(state:string) { // Funkcija za generiranje URL-ja za preusmeritev na prijavno stran
  const authCodeUrlParameters = {
    scopes: env.OAUTH_SCOPES.split(" "), // Funkcija za generiranje vnesemo pravice, ki jih zahtevamo za uporabnika
    redirectUri: env.OAUTH_REDIRECT_URI, // Funkcija za generiranje vnesemo na kateri backend URL naj posle kodo za dostop
    state: state, // Funkcija za generiranje vnesemo podatek, ki ga bomo potrebovali kasneje
    prompt: 'select_account' // Funkcija za generiranje vnesemo ta podatek, da si lahko uporabnik zmeraj izbere profil
  };
  let url = await clientApplication.getAuthCodeUrl(authCodeUrlParameters) // Generiranje URL-ja za preusmeritev na prijavno stran
  return {url:url} // tukaj poslemo generirani URL nazaj
}
```

Slika 62: Funkcija za generiranje URL-ja za preusmeritev uporabnika na prijavno stran (Vir: lasten)

```
@Get("/redirect/")
async redirect(@Query() query, @Session() session, @Res() res:Response) { // Funkcija za preusmeritev iz Microsofta po prijavi
  let code = query.code || "" // Koda za dostop do dostopnega žetona, ... od uporabnika
  let state = query.state || "" // Označka za platforme iz katere se je uporabnik prijavil

  if(code == "") { // Preverimo če je koda prazna
    return res.send({}) // Če je koda prazna vrnemo
  }

  const tokenRequest = { // Parametri za zahtevo za dostopni žeton, ... od uporabnika
    code: code, // Koda za dostop
    scopes: env.OAUTH_SCOPES.split(" "), // Katere pravice zahtevamo od uporabnika
    redirectUri: env.OAUTH_REDIRECT_URI, // URL od backend-a na katerega je bil preusmerjen uporabnik po prijavi
    responseType: "offline",
  };

  let respons = await clientApplication.acquireTokenByCode(tokenRequest) // Zahtevamo dostopni žeton, žeton za osvežitev, ... od uporabnika
  const tokenCache = clientApplication.getTokenCache().serialize() // Iz predpomnilnika zahtevamo podatke
  const refreshTokenObject = (JSON.parse(tokenCache)).RefreshToken
  const refreshToken = refreshTokenObject[Object.keys(refreshTokenObject)[0]].secret; // Žeton za osvežitev
  const token = { // Žeton, ki ga shranimo v uporabnikovo sejo ali v primeru mobilne aplikacije v pomnilnik in vsebuje:
    accessToken: respons.accessToken, // Dostopni žeton
    refreshToken: refreshToken, // Žeton za osvežitev
    expiresOn: respons.expiresOn, // Rok trajanja dostopnega žetona
  };

  session.token = token // Žeton shranimo v uporabnikovo sejo
  session.save() // Uporabnikovo sejo shranimo
  if(state == "appscv") { // app.scv.si
    return res.redirect("https://app.scv.si/?success=signin") // Tukaj preusmerimo uporabnika iz katere platforme je prišel
  } else if(state == "localhost") { // localhost
    return res.redirect("http://localhost:3000/?success=signin") // Tukaj preusmerimo uporabnika iz katere platforme je prišel
  }
  return res.json(token) // Tukaj samo vrnemo žeton za uporabnika na aplikaciji
}
```

Slika 63: Funkcija za ureditev preusmeritve iz Microsofta po prijavi (Vir: lasten)


```
@Get("/get/")
async getUserData(@Session() session, @Res() res:Response, @Headers() headers){//Funkcija za pridobitev osnovnih uporabnikovih podatkov
  res.setHeader('Access-Control-Allow-Methods','GET')
  res.setHeader('Access-Control-Allow-Origin',headers.origin || '')
  res.setHeader('Access-Control-Allow-Credentials','true')

  if(!session.token){//Preverimo če je v uporabnikovi seji shranjen žeton
    return res.status(HttpStatus.NOT_ACCEPTABLE).send("error");//če žetona ni vrnemo napako
  }

  let token = await getToken(session.token) || ""//Žeton
  if(token == ""){//Preverimo ali je žeton pravilen in ali če je že potekel, da ga obnovimo
    return res.status(HttpStatus.NOT_ACCEPTABLE).send("error");//če je nepravilen vrnemo napako
  }

  let accessToken = (<any>token).accessToken || ""//Iz žetona dobimo dostopni žeton
  const client = this.userService.getClient(accessToken)//Uporabnik
  let data = await client.api("me").get()//Z uporabnikovo funkcijo get dobimo želene podatke
  session.token = token//Žeton shranimo v uporabnikovo sejo
  session.save()//Uporabnikovo sejo shranimo

  return res.status(HttpStatus.OK).json(data)//Vrnemo uporabnikove podatke
}
```

Slika 64: Funkcija za pridobitev osnovnih uporabnikovih podatkov (Vir: lasten)

```
@Get("/logoutUrl/")
logoutUrl(@Res() res:Response){//Funkcija ki te preusmeri na Microsoftov URL za odjavo iz Microsoft profila
  res.redirect('https://login.microsoftonline.com/common/oauth2/logout?post_logout_redirect_uri=${env.OAUTH_REDIRECT_URI}=="http://localhost:5050/auth/redirect/"?htt
}

@Get("/logout/")
logoutUser(@Session() session, @Headers() headers, @Res() res:Response){//Funkcija, ki skrbi za zbrisanje žetona iz uporabnikove seje po izpisu iz Microsoftovega računa
  session.token = undefined//Žeton nastavimo na nedoločeno (s tem izbrisemo žeton)
  let host = headers.host
  if(host == "localhost:5050"){//tukaj preusmerimo uporabnika, glede na to iz kod je prišel
    return res.redirect("http://localhost:3000/?success=logout")
  }else if(host == "backend.app.scv.si"){
    return res.redirect("https://app.scv.si/?success=logout")
  }
  return res.redirect("https://app.scv.si/?success=logout")
}
```

Slika 65: Funkciji za odjavo uporabnika (Vir: lasten)

```
async function getUserData(){//Funkcija, ki pridobi uporabnikove podatke, ki so potrebni
  let json = await fetch(`${process.env.REACT_APP_BACKEND_URL}/user/get/`,{//URL na katerem se nahajajo osnovni podatki o uporabniku
    mode:'cors',//Prijemne podatke lahko uporabi tudi iz tujih domen
    credentials:'include',//Seja uporabnika je vključena
    method:'GET',//Uporabljena je metoda GET
  }).catch(e=>{//če spodleti uporabnika vrne na prijavno stran
    setLoaded(true)
    window.location.pathname="/prijava"
  })
  let data = {}
  if(json.ok){//Preverimo ali je zahteva ureda
    data = await json.json()//Zahteva je ureda in pretvorimo tip besede v tip predmeta in shranimo v spremenljivko
  }else{
    return window.location.pathname="/prijava"//Uporabnika vrnemo na prijavno stran
  }
}
```

Slika 66: Funkcija, ki pridobi uporabnikove podatke (Vir: lasten)

Kode za zaledni del: <https://github.com/DmgCoder/SCVAppBackend.git>

Koda za čelni del: <https://github.com/DmgCoder/scvapp-react.git>

Koda za mobilno aplikacijo: <https://github.com/DmgCoder/SCVAppFlutter.git>

10 VIRI IN LITERATURA

- (1) <http://www.flutter.dev/> (13. 3. 2022)
- (2) <https://reactjs.org/tutorial/tutorial.html> (28. 2. 2022)
- (3) <https://commons.wikimedia.org/wiki/File:React-icon.svg> (28. 2. 2022)
- (4) <https://docs.nestjs.com/> (28. 2. 2022)
- (5) <https://github.com/nestjs/nestjs.com> (28. 2. 2022)
- (6) <https://www.techtarget.com/searchcloudcomputing/definition/Windows-Azure>
(17. 4. 2022)
- (7) https://commons.wikimedia.org/wiki/File:Microsoft_Azure.svg (28. 2. 2022)
- (8) <https://cpanel.net/> (28. 2. 2022)
- (9) <https://cpanel.net/> (28. 2. 2022)
- (10) <https://www.figma.com> (8. 3. 2022)
- (11) <https://developer.android.com/studio/intro> (13.3.2022)
- (12) <https://www.zeroappstore.com/what-is-xcode-and-why-do-i-need-it.html> (13.3.2022)
- (13) <https://www.heroku.com/what> (17.4.2022)
- (14) <https://docs.microsoft.com/en-us/azure/active-directory/develop/v2-oidc-auth-code-flow> (28. 2. 2022)