

Mobilne aplikacije u javnim poslovima

Minić, Petra

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka / Sveučilište u Rijeci**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:195:260520>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-12-28**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Informatics and Digital Technologies - INFORI Repository](#)



Sveučilište u Rijeci – Odjel za informatiku
Preddiplomski studij jednopredmetne informatike

Petra Minić

Mobilne aplikacije u javnim poslovima

Završni rad

Mentorica: izv. prof. dr. sc. Marina Ivašić-Kos

Rijeka, rujan 2019.

Sadržaj

Sažetak	3
1. Uvod	4
2. Aplikacije	5
3. Vizualno programiranje	6
3.1 Dizajn aplikacije	7
3.2 Početna stranica	8
3.3 Registracija	10
3.4 Prijava	12
3.5 Lokacija – Liječnik	14
3.6 Naručivanje lijeka	17
4. Poboljšanja	20
5. Zaključak	21
6. Literatura	22
7. Popis slika	23

Sažetak

U završnom radu predstavljen je razvoj prototipa aplikacije za naručivanje lijekova u ljekarni. Na temelju prototipa može se definirati osnovna funkcionalnost aplikacije i potrebne karakteristike kako bi se zadovoljili zahtjevi korisnika, a dizajn aplikacije i unaprjeđenje funkcionalnost doraditi će se prije nastanka završnog proizvoda koji će biti plasiran na tržište. Završni rad se sastoji od ukupno pet dijelova, uključujući uvod i zaključak.

U prvom, uvodnom dijelu iznesena je osnovna ideja koja stoji iza razvoja aplikacije koja je namijenjena zdravstvenom sektoru i koji je konačni cilj same aplikacije. Također, definirani su osnovni pojmovi poput mobilne aplikacije, javnih poslova, primjenjivih programa.

U drugom dijelu se objašnjavaju vrste aplikacije te njihove prednosti i nedostaci. Također, definirano je što je to prototip i kojim se softverom služimo za njegovu izradu.

Treći dio objašnjava što je to vizualno programiranje, koji je cilj vizualnog programiranja i definirana su tri različita stupnja vizualnog programiranja. Također, prikazani su svi ekrani aplikacije sa detaljnim objašnjenjima kako funkcioniraju ti ekrani pomoću prototipa i vizualnog programiranja.

Četvrti dio opisuje dodatne mogućnosti koje aplikacije može imati te kako se može unaprijediti uz pomoć vizualnog programiranja.

U posljednjem, završnom dijelu predstavljen je zaključak razvoju aplikacije uz pomoć vizualnog programiranja.

Ključne riječi: mobilne aplikacije, javni poslovi, primjenjivi program, prototip, vizualno programiranje.

1. Uvod

Mobilne aplikacije su vrsta primjenjivog programa koji se koristi na mobilnom uređaju, poput tablet računala ili pametnog telefona. Njihova primarna svrha na početku razvoja je bila asistencija u produktivnosti¹, no tokom vremena zahtjevi ljudi su se počeli sve više mijenjati tražeći veće mogućnosti, veću brzinu i memoriju novih aplikacija. Postoje tri vrste mobilnih aplikacija, a to su: native, hibridne i progresivne web aplikacije koje će biti unutar drugog poglavlja više objašnjenje.

Primjenjivi program je program izrađen u svrhu da pomogne korisnicima prilikom izvođenja jednog ili više određenih zadataka.²

Javni posao je izraz kojim se opisuje projekti koje posredno ili neposredno izvodi država u općem interesu za boljitak javnog života ljudi.³

U sljedećim poglavljima bit će opisan razvoj aplikacije za naručivanje i dobavu lijekova pacijentima u zdravstvenom sektoru. Aplikacija je primarno namijenjena korisnicima koji imaju ponavljajuće recepte prepisane od strane svojih doktora. Ponavljajući recepti su uputnice propisane od strane liječnika prilikom neke dugotrajne terapije, najčešće za kronične bolesti u kojem pacijenti trebaju uzimati odgovarajući lijek dugotrajno ili više puta do prestanka terapije. Međutim, kako ponavljajući recept ima rok trajanja za korištenje koji je najčešće šest mjeseci⁴, pacijenti se moraju opet obratiti svom liječniku kako bi obnovili terapiju. Zbog ponovnog propisivanja terapije koja se za kronične bolesti u pravilu ne mijenja, pacijent treba posjetiti liječnika u ordinaciji ili ga kontaktirati telefonom, čekajući red u ordinaciji ili slobodnu telefonsku liniju i stvarajući nepotrebnu gužvu i dodatni posao liječniku koji u tom slučaju izvršava samo administrativnu funkciju ponovnog propisivanja recepta.

Aplikacija je zamišljena da pomogne i pacijentima i liječnicima na način da bude povezana sa sustavom ljekarni i doktora pri čemu će korisnici moći s obzirom na svoj liječnički karton pretražiti bazu lijekova i naručiti lijek koji im je propisan terapijom. Baza lijekova će biti temeljena na liječničkom kartonu korisnika. Ukoliko je njihov odabir ispravan, odnosno recept je istekao nakon provjere liječničke baze, korisnik će dobiti automatsku potvrdu aplikacije o ponovnoj mogućnosti preuzimanja lijeka unutar ljekarne Republike Hrvatske. Na taj način bi se postiglo smanjeno opterećenje doktora koji imaju pacijente s ponavljajućom terapijom. Također, danas veliki broj ljudi, njih čak 90% po statistici koristi mobilne aplikacije za vrijeme korištenja mobitela⁵ te na ovaj način bi se olakšao pristup korisnicima jer bi u bilo koje doba dana, tjedna mogli poslati zahtjev za receptom te ga preuzeti u ljekarni.

¹ https://en.wikipedia.org/wiki/Mobile_app

² <https://hr.wikipedia.org/wiki/Aplikacija>

³ https://en.wikipedia.org/wiki/Public_service

⁴ <https://www.ljekarne-hrvatinkajfes.com/faq-items/pitanje-2/>

⁵ <https://buildfire.com/app-statistics/>

2. Aplikacije

Postoje tri vrste aplikacija: native, hibridne i progresivne web aplikacije.

Native mobilne aplikacije⁶ su aplikacije koje su prvenstveno rađene samo za jedan mobilni operacijski sustav. Prednosti nativnih aplikacija su brzina, responzivnost i lagana distribucija, dok su nedostaci znatno skuplja izrada same aplikacije te zahtijevaju češće ažuriranje za razliku od ostalih vrsta mobilnih aplikacija. Mobilni operacijski sustav⁷ je sustav koji kombinira elemente klasičnog operacijskog sustava za računala i elementa namijenjenih isključivo mobilnim uređajima. Vrste mobilnih platformi su: IOS, Android, BlackBerry i Windows.

Progresivne web aplikacije⁸ predstavljaju prijelaz između mobilnih web stranica i mobilnih aplikacija. Zahtijevaju minimalnu memoriju jer se baza podataka nalazi na samom serveru. Jednostavne su za održavanje i izradu. Međutim, takve aplikacije su puno manje interaktivne i intuitivne za korisnika. Nisu dostupne na mobilnim platformama već korisnici tih aplikacija trebaju poznavati URL aplikacije.

Hibridne mobilne aplikacije⁹ predstavljaju kombinaciju ranije navedenih vrsta, nativnih i progresivnih web aplikacija. Prednost hibridnih aplikacije je mogućnost korištenja na svim mobilnim operacijskim sustavima, laka distribucija i razvoj aplikacije je znatno brži. Nedostaci su minimalni i zanemarivi s obzirom da se svakim danom tehnologija sve više razvija i poboljšava.

Kroz sljedeća poglavlja biti će prikazani okvirni izgled prototipa i željene funkcije aplikacije . Aplikaciju bi svrstali u hibridnu vrstu aplikaciju iz razloga što će se baza podataka ljekarne i liječnika nalaziti na samom serveru, a isto tako će imat kvalitete brzine i responzivnosti.

Prototip je prvotni oblik ili model na kojem se temelji program.¹⁰ Za razvoj prototipa korištena je aplikacija UXPin¹¹ i App Inventor for Android¹². UXPin je online softver koji omogućuje izradu aplikacije na način kako bi dizajneri aplikacija mogli investitorima na jednostavan način prikazati željeni proizvod, ideju pod kojom se vode ili jednostavno rečeno online softver za izgradnju prototipa. Međutim, nedostatak ovog programa je nemogućnost programiranja iz tog razloga koristi se i App Inventor koji služi kao alat za vizualno programiranje.

⁶ <https://analissa-digital.hr/vrste-mobilnih-aplikacija/>

⁷ https://en.wikipedia.org/wiki/Mobile_operating_system

⁸ <https://analissa-digital.hr/vrste-mobilnih-aplikacija/>

⁹ <https://analissa-digital.hr/vrste-mobilnih-aplikacija/>

¹⁰ <http://www.enciklopedija.hr/natuknica.aspx?id=50773>

¹¹ <https://www.uxpin.com/>

¹² <https://appinventor.mit.edu/>

3. Vizualno programiranje

Vizualno programiranje ili kratica VPL je vrsta modeliranja u kojem se algoritam stvara slikama umjesto pisanim unosom koda. Zbog svoje jednostavnosti, vizualno programiranje je alat koji smanjuje apstrakciju u programiranju.¹³

Generalni cilj vizualnog programiranja je da početnicima i programerima omogući lakši pristup programiranju i da podrži programere na tri različita stupnja.¹⁴

Prvi stupanj je sintaksa. Vizualno programiranje koristi ikone, forme i dijagrame. Na taj način smanjuje mogućnost za potencijalne gramatičke i jezične greške tokom programiranja koji se znaju pojaviti.

Drugi stupanj je semantička. Vizualno programiranje može ponuditi neke mehanizme pomoću kojih se otkriva značenje primitivnih programskih jezika, npr. DRAKON. Također, može pomoći u izradi dokumentacije za programske jezike.

Treći stupanj je pragmatika. Vizualno programiranje omogućava izrađenim programima stvoriti testnu radnu okolinu u kojoj se onda promatra ponašanje programa s obzirom na dane ulazne podatke te se gleda jesu li izlazni rezultati očekivani.

Postoji lista vizualnih programa za programiranje namijenjeni različitim vrstama, poput edukacijski, multimedijski, video igre, sustavi, poslovno inteligencija i mnogi drugi.

Za aplikaciju izabrana je edukacijska vrsta vizualnog programiranja. Program za izradu aplikacije pomoću vizualnog programiranja je App Inventor for Android (Slika 1.) koji je napravljen kako bi se što široj populaciji omogućilo da izrade vlastite aplikacije. Također, omogućava instalaciju i pokretanje vlastitih izgrađenih aplikacija.



Slika 1 - App Inventor for Android logo

¹³ <https://bim-hrvatska.hr/vizualno-programiranje-kao-pomoc-projektiranju/>

¹⁴ https://en.wikipedia.org/wiki/Visual_programming_language

3. 1 Dizajn aplikacije

Prilikom dizajna aplikacije korišteni su jednostavni elementi i boje kako bi aplikacija bila primjenjiva za sve uzraste.

Prevladavajuća boja u aplikacija je plava. Plava boja je povezana sa značenjima povjerenja, iskrenosti, mudrosti, stabilnosti, vjere i inteligencije, sve kvalitete koje bi ova aplikacija trebala predstavljati svojim korisnicima koji na ovaj način dolaze u kontakt sa svojim liječnicima. Također, plava boja ima pozitivni efekt na um i tijelo. Kao boja koja predstavlja duh, može utjecati na proizvodnju kemikalija unutar tijela koji potiču umirujući efekt na tijelo. Na slici 2. prikazano je nijansa plave boje koja se proteže kroz cijelu aplikaciju.

Naziv aplikacije je CroVella (Slika 3). Cro je kratica engleske riječi Croatia što znači Hrvatska, a Vella je talijanska riječ čije je značenje jedrilica. Konačni prijevod naziva aplikacije je Hrvatska jedrilica. Razlog odabira tog imena poseže za činjenicama iz povijesti. Grad Rijeka je najveća hrvatska luka, treći po veličini grad u Hrvatskoj te je u prošlosti bila jedna od najjačih luka na ovom području zajedno sa Venecijom.¹⁵ Nekad davno svi bitni proizvodi koji su bili potrebni ljudima su dolazili brodovima, još od samih početaka kada su se koristili jedrenjaci. Gledajući metaforički, ova aplikacija je kao jedrenjak koji povezuje korisnika, doktora i ljekarnu. Korisnik „šalje jedrenjak“, odnosno zahtjev bazi koja je povezana sa njegovim doktorom da mu se pošalje lijek. Poslani zahtjev zatim „plovi“ dalje u ljekarnu kako bi korisnik prilikom dolaska u nju mogao podignuti svoj lijek.

Korištenjem ove aplikacije, korisnik nije ograničen na ljekarnu u nekom području već može podignuti lijek u bilo kojem gradu unutar Republike Hrvatske.



Slika 2 - Plava boja (#3366cc)



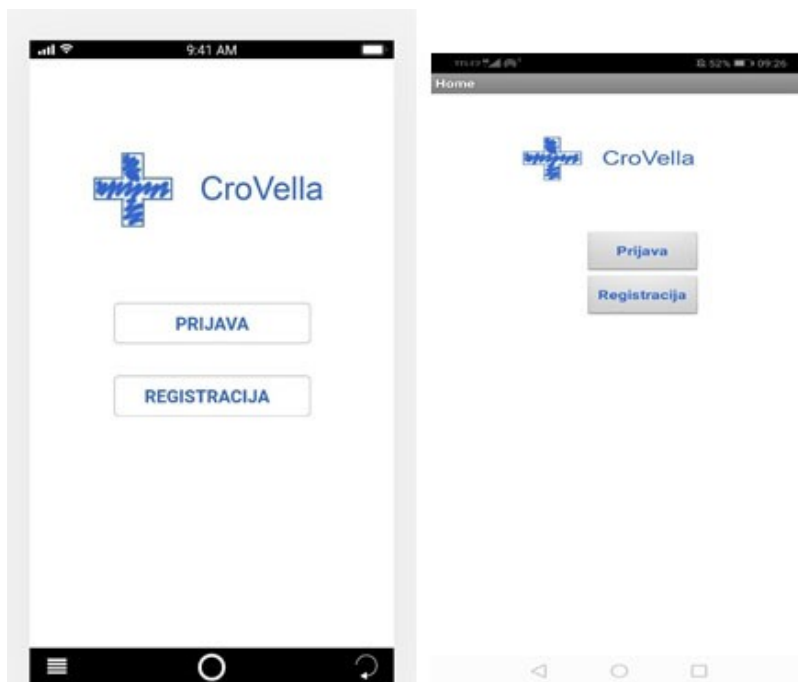
Slika 3 - Logo aplikacije CroVella

¹⁵ <https://hr.wikipedia.org/wiki/Rijeka>

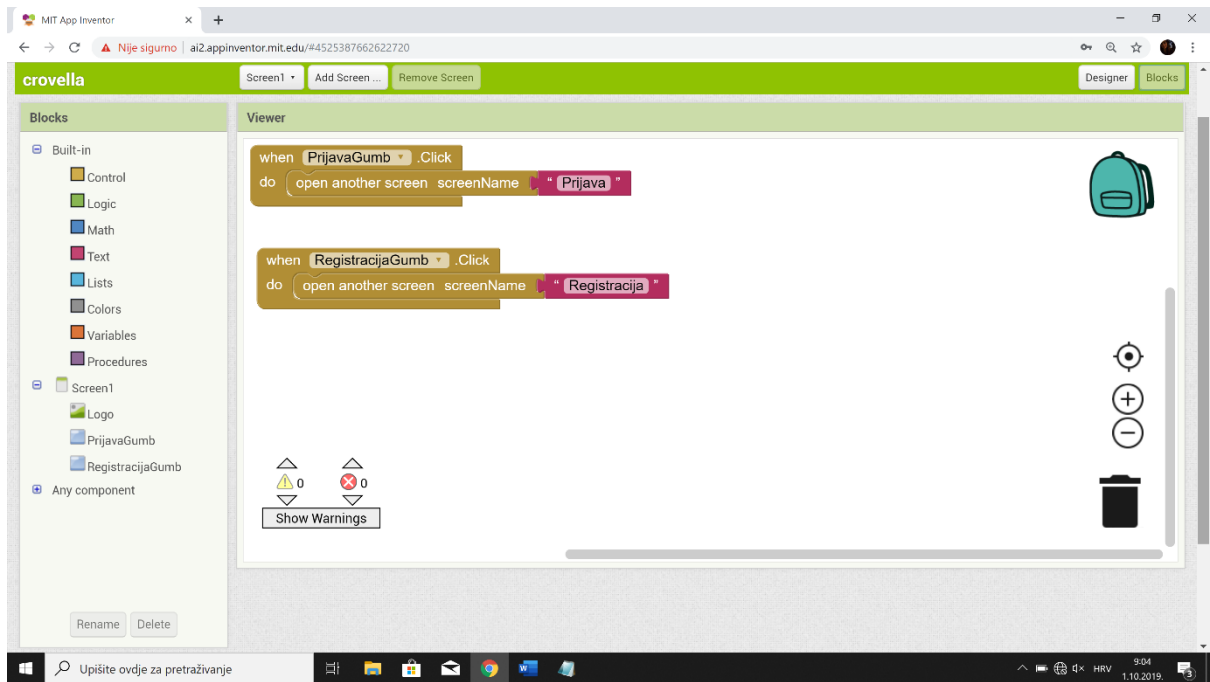
3.2 Početna stranica

Ulaskom u aplikaciju otvara se početni ekran (Slika 4.) koji nudi korisniku dvije opcije: prijavu i registraciju. Pritiskom gumba „Prijava“ otvara se prozor Prijava gdje korisnici ispunjavaju prijavni obrazac. Prijavu pritišću korisnici koji su već obavili registraciju. Ukoliko to nisu napravili, korisnici pritišću gumb „Registracija“. Otvara se prozor Registracija kojom pristupaju registracijskom obrascu. Registracija omogućava upis u bazu podataka ukoliko je to korisniku prvi pristup aplikaciji.

Slika 5 prikazuje vizualno programiranje pritiska gumba.



Slika 4 - UXPin početna stranica vs VPL početna stranica



Slika 5 - VPL blokovi početne stranice

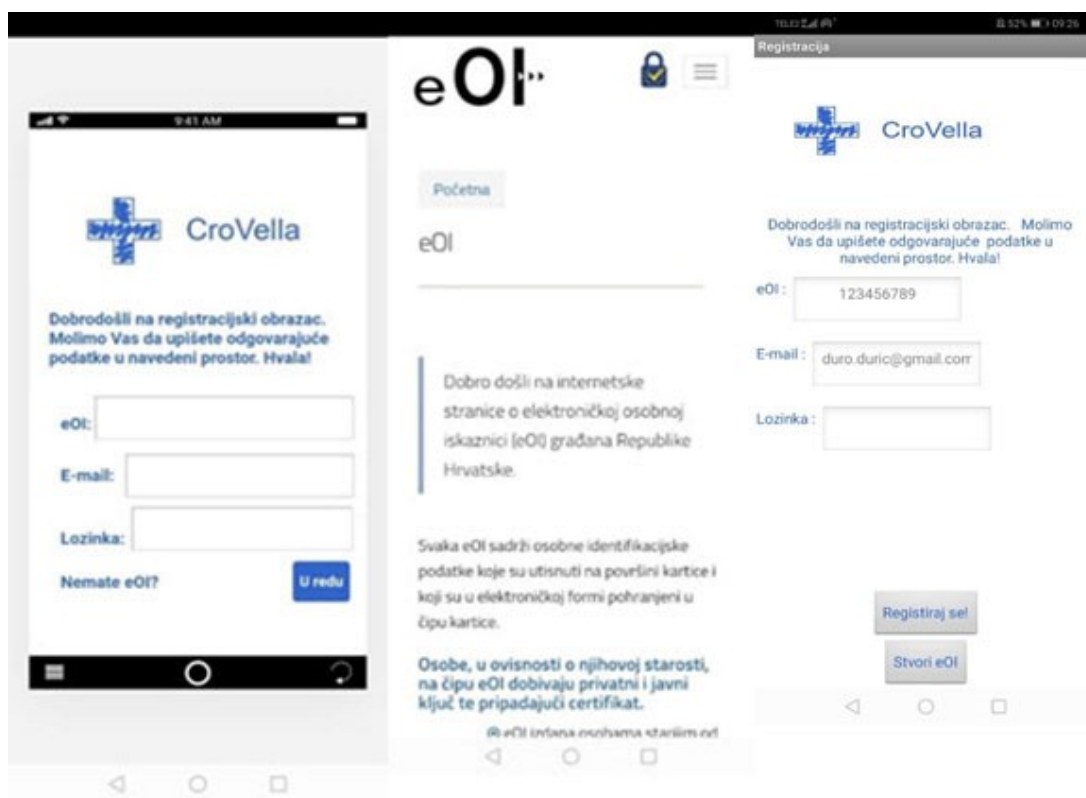
3.3 Registracija

Registraciju obavljaju korisnici aplikacije koji prvi put dolaze u dodir sa aplikacijom (Slika 6).

Prilikom registracije korisnik mora upisati eOI¹⁶ koji može nabaviti u lokalnom MUP-u prilikom izrade nove osobne iskaznice. eOI pin koji korisnik postavi prilikom prijave na stranicama MUP-a služi kao vlastoručni digitalni potpis te preko njega pristupa stranicama sa određenim ovlastima. U aplikaciji eOI bi služio za podatke o trenutnom aktivnom prebivalištu korisnika te bi im taj grad bio prvi među ponuđenima, iako naravno korisnik bi imao opciju promjene lokacije prilikom odabira za slučajeve kada se doktor ne nalazi u njihovom gradu prebivališta.

Korisnik zatim upisuje e-mail i lozinku za aplikaciju po vlastitom odabiru, ti bi se podaci spremili u bazu podataka korisnika aplikacije CroVella čiji bi primarni ključ bio eOI. Baza korisnika bi sadržavala eOI (primarni ključ), e-mail korisnika, lozinku i prebivalište koji bi se dobio na temelju eOI-a.

Lozinka unutar baze bi šifrirali Password-Based Key Derivation Function 2 (PBKDF2), koja je otpornija na napade grubom silom. Napadi grubom silom su tehnika rješavanja problema koja se sastoji od sustavnog pronalaženja svih mogućih kandidata za rješenje i isprobavanja svakog od njih.

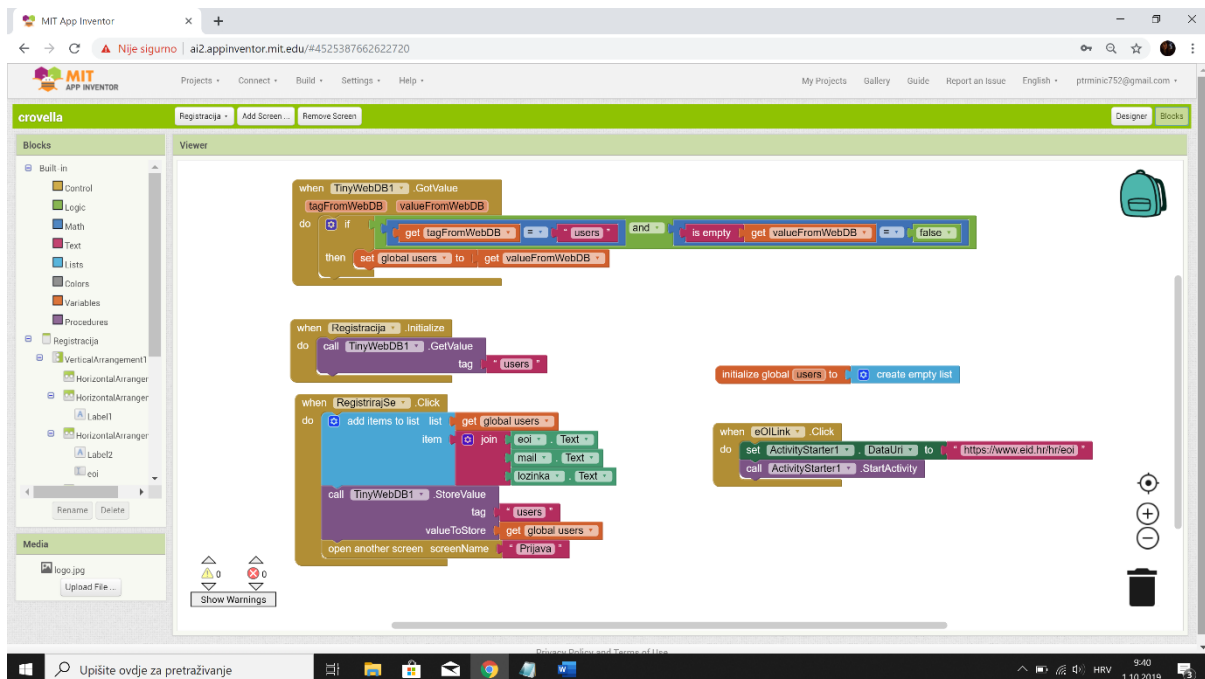


Slika 6 - UXPin registracijski obrazac vs VPL registracijski obrazac

¹⁶ <https://www.eid.hr/hr/eoi>

Pomoću vizualnog programiranja inicijaliziran je prozor Registracija i poziva se baza podataka sa globalnom varijablom „users“. Korisnici prilikom registracije upisuju potrebne podatke koji su navedeni ranije. Nakon što korisnici unesu svoje podatke u registracijski obrazac, pritiskom gumba „Registriraj se!“ dodajemo podatke unutar liste koju zatim spremamo unutar baze podataka čiji je primarni ključ eOI korisnika, potom prosljeđuje korisnika na stranicu Prijava. Također, provjerava se je li obrazac prazan prilikom upisa, ukoliko je prosljeđuje se prazna lista. U suprotnom ako to ne napravimo, aplikacija pokazuje grešku te ne može se doći do registracije korisnika i drugih ekrana unutar aplikacije. Pritiskom gumba „stvari eOI“ (unutar koda imena eOILink) otvara se web stranica na kojoj korisnici mogu aktivirati svoj eOI ukoliko ga nemaju.

Slika 7 prikazuje vizualno programiranje baze podataka u kojem se spremaju uneseni podaci od strane korisnika prilikom ispunjavanja registracijskog obrasca i programirani gumbi za prosljeđivanje korisnika u sljedeće dijelove aplikacije.



Slika 7 - VPL blokovi registracijski obrazac

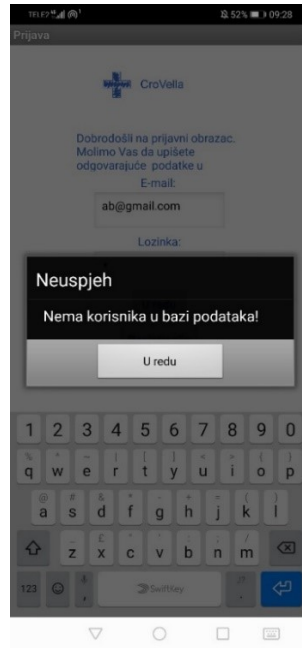
3.4 Prijava

U prijavnom obrascu (Slika 8) korisnik upisuje svoj e-mail i lozinku kojom se prethodno registrirao unutar aplikacije pomoću registracijskog obrasca kako je objašnjeno u prethodnom poglavlju.



Slika 8 - UXPin prijavni obrazac vs VPL prijavni obrazac

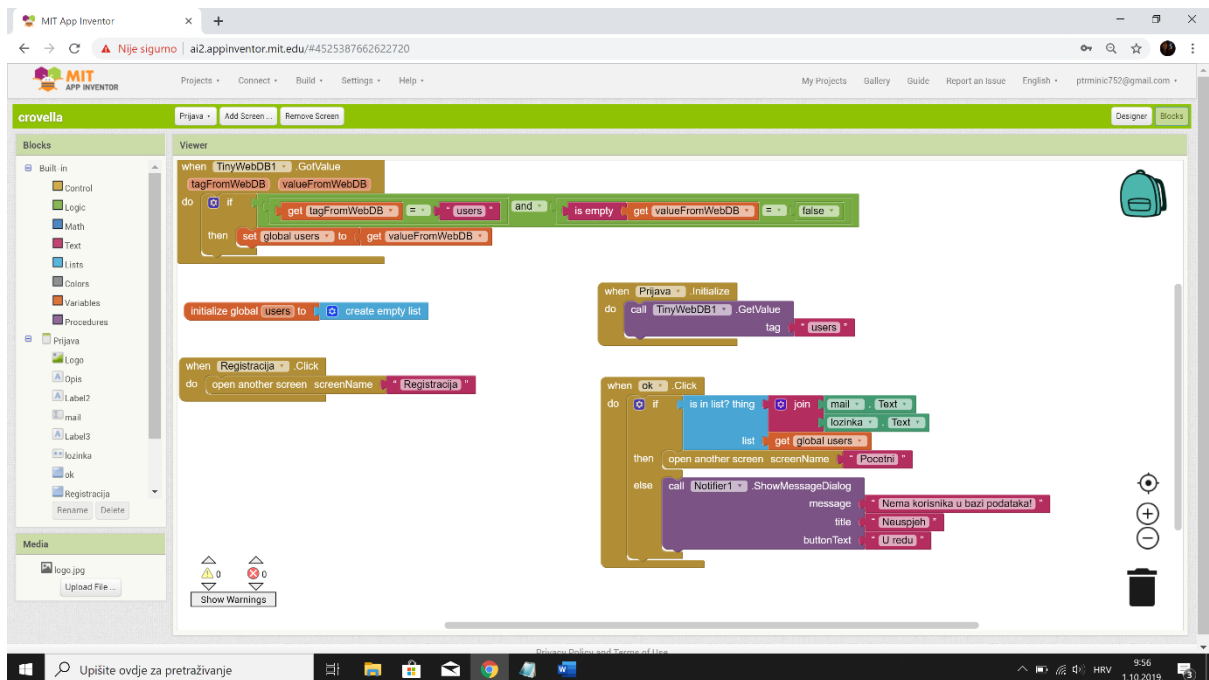
Pomoću vizualnog programiranja inicijalizira se prozor Prijava pomoću kojeg se poziva baza podataka sa globalnom varijablom „users“. Nakon što korisnici unesu podatke za prijavu, a to su e-mail i lozinka, pritiskom gumba „U redu“ provjerava se baza podataka. Ukoliko se korisnik nalazi u bazi podataka otvara se novi prozor Lokacija-Liječnik (Početni se zove unutar koda) te mu se omogućava daljnji pristup unutar aplikacije kako bi mogao izvršiti upit za lijekom. Međutim, ako korisnika nema unutar baze podataka zabranjuje se daljnji prosljeđivanje korisnika i na ekranu se ispisuje poruka „Nema korisnika u bazi podataka!“ (Slika 9). Također, provjerava se je li obrazac prazan prilikom upisa, ukoliko je prosljeđuje se prazna lista. U suprotnom, aplikacija pokazuje grešku te se korisnik ne može prijaviti unutar aplikacije. Za ponovni pokušaj ispravne prijave, korisnik mora ugasiti „prisilno“ aplikaciju i ući ponovno u aplikaciju.



Slika 9 - Poruka o neuspjehu prijave

Pritiskom gumba „Registracija“ otvara se prozor obrasca za registraciju te se novi korisnici mogu registrirati, nakon čega će biti prebačeni natrag na stranicu Prijava.

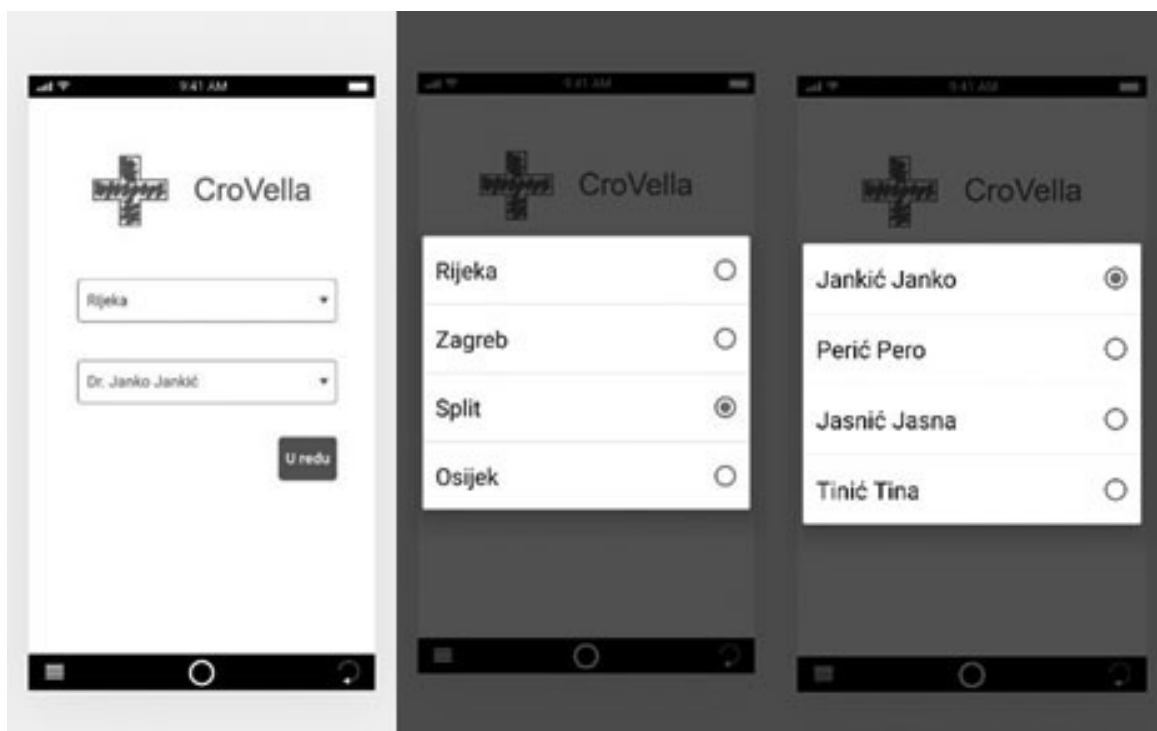
Slika 10. prikazuje vizualno programiranje baze podataka u kojem se provjeravaju uneseni podaci od strane korisnika prilikom ispunjavanja prijavnog obrasca i programirani su gumbi za prosljeđivanje korisnika u sljedeće dijelove aplikacije.



Slika 10 - VPL blokovi prijavnog obrasca

3.5 Lokacija – Liječnik

Ulaskom u aplikaciju (Slika 11, Slika 12) nakon što se korisnik uspješno prijavio dolazi do ekrana Lokacija – Liječnik. U tom dijelu korisnik bira lokaciju liječnika, odnosno grad i ime liječnika koji je ponuđen unutar izbornika temeljom upisane lokacije. Lokacija na početku bude postavljena automatski temeljeno na korisnikovom prebivalištu, tako što taj podatak povlačimo iz sustava eOI kako je objašnjeno u poglavlju 3.3 Registracija.



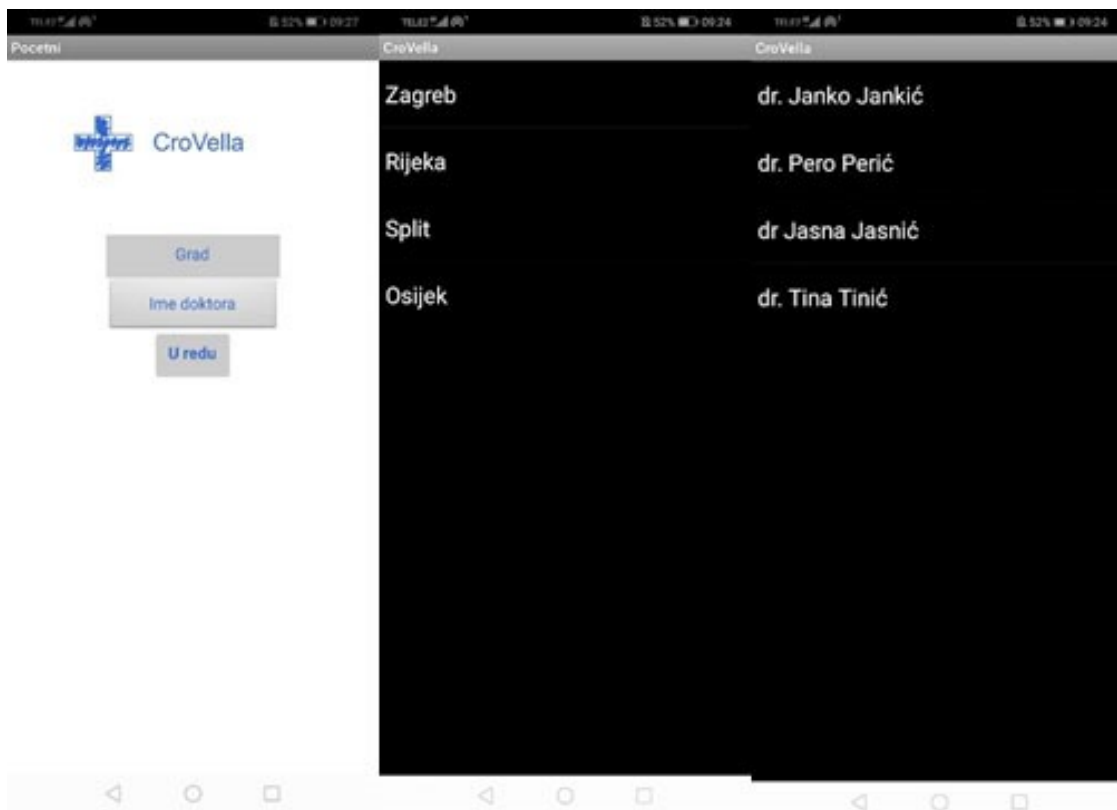
Slika 11 - UXPin ekran Lokacija - Liječnik sa izbornicima

Međutim, korisnik ima mogućnost promjene lokacije iz razloga što se liječnik može nalaziti na nekoj drugoj lokaciji neovisno o prebivalištu korisnika. Ovisno o odabranoj i/ili automatski povučenoj lokaciji korisnika, korisniku se nudi izbornik sa popisom liječnika te se odabir vrši na temelju kod koga se nalazi liječnički karton korisnika za traženi lijek.

Kako bi se mogao prikazati ispravan izbornik liječnika korisniku s obzirom na konačnu lokaciju treba se izvršiti SQL upit. Baza podataka liječnika sadrži titulu, ime, prezime liječnika i lokaciju. SQL upit bi glasio:

```
SELECT ime_lijecnika, prezime_lijecnika
```

```
WHERE grad = #grad (odabrani grad).
```

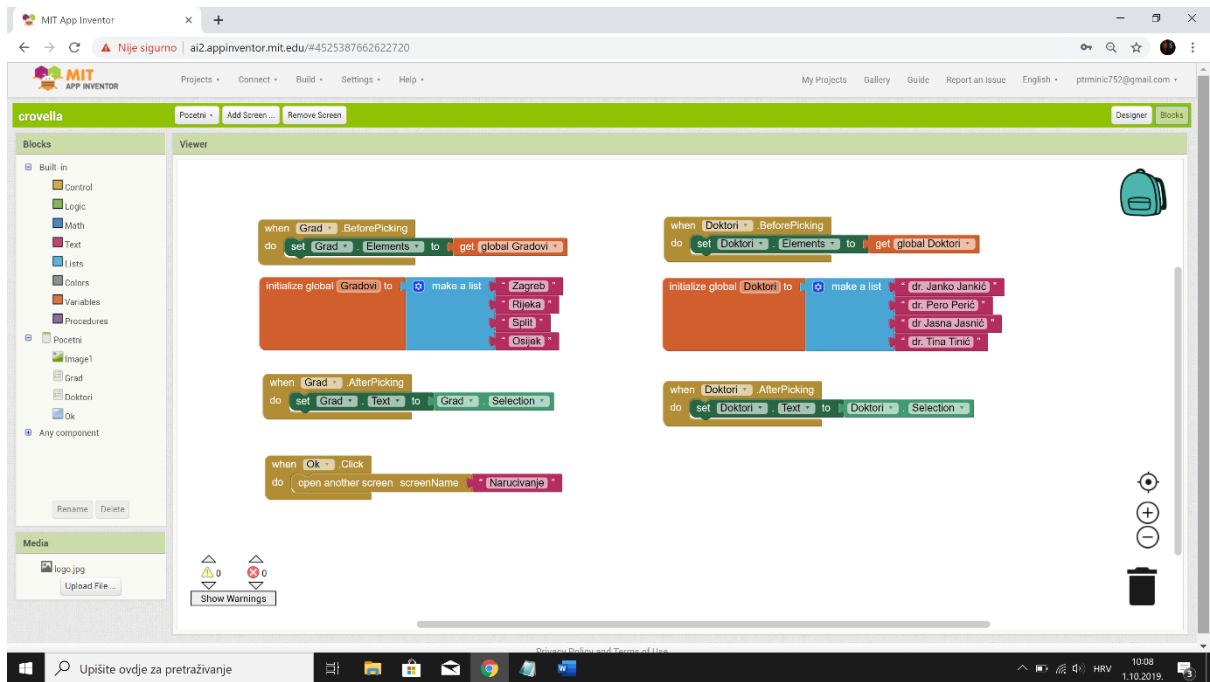


Slika 12 - VPL ekran Lokacija - Liječnik sa izbornicima

Pomoću vizualnog programiranja inicijalizira se globalna varijabla „Gradovi“ u koju je spremljena lista gradova, a to su u ovom slučaju: Zagreb, Rijeka, Split, Osijek. Prije odabira prikaže se lista gradova, korisnik odabire lokaciju liječnika te se vrijednost postavlja na odabrano.

Zatim se inicijalizira globalna varijabla „Doktori“ u koju je spremljena lista doktora, a neki od njih su: dr. Janko Jankić, dr. Pero Perić. Pomoću SQL upita ranije navedenog bi napravili sortiranje popisa liječnika da se prikažu oni koji se nalaze na odabranoj lokaciji korisnika. Prikaže se lista doktora, korisnik odabire liječnika te se vrijednost postavlja na odabrano ime i prezime. Pritiskom gumba „U redu“ čime korisnik potvrđuje svoj izbor, otvara se novi ekran Naručivanje lijekova (u kodu pod imenom „Narucivanje“).

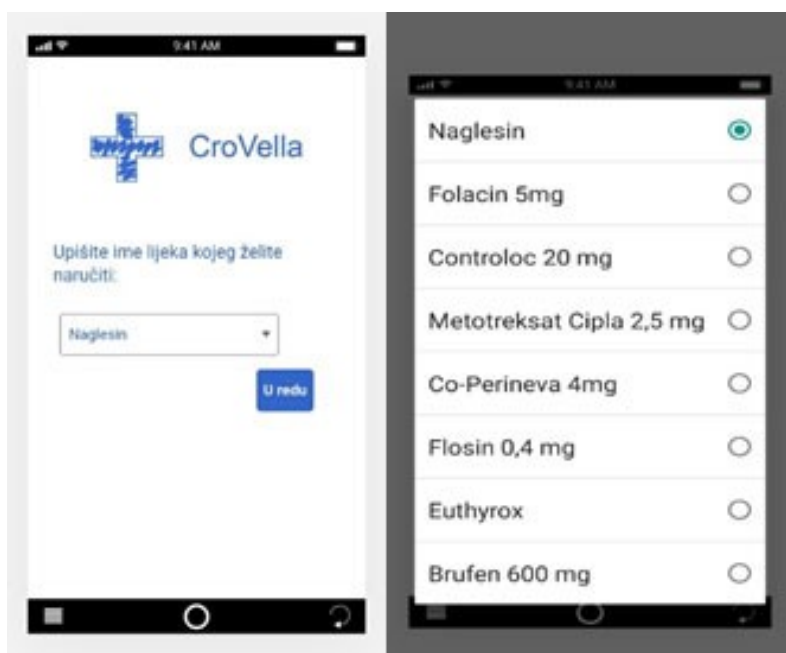
Slika 13 prikazuje vizualno programiranje padajućih izbornika gradova i doktora te „pamćenja“ odabranih vrijednosti. Također, vidimo gumb „U redu“ koji nas prosljeđuje u idući korak u aplikaciji prilikom naručivanja.



Slika 13 - VPL blokovi ekrana Lokacija – Liječnik

3.6 Naručivanje lijeka

Poslije ekrana Lokacija – Liječnik gdje su korisnici odabrali lokaciju i ime liječnika, otvara se novi ekran Naručivanje lijeka. Korisnik ovdje ima mogućnost otvaranja padajućeg izbornika lijekova temeljen na pacijentovom kartonu (Slika 14). Na temelju padajućeg izbornika odabire se lijek kojem je korisniku potreban te taj odabir postaje zahtjev.

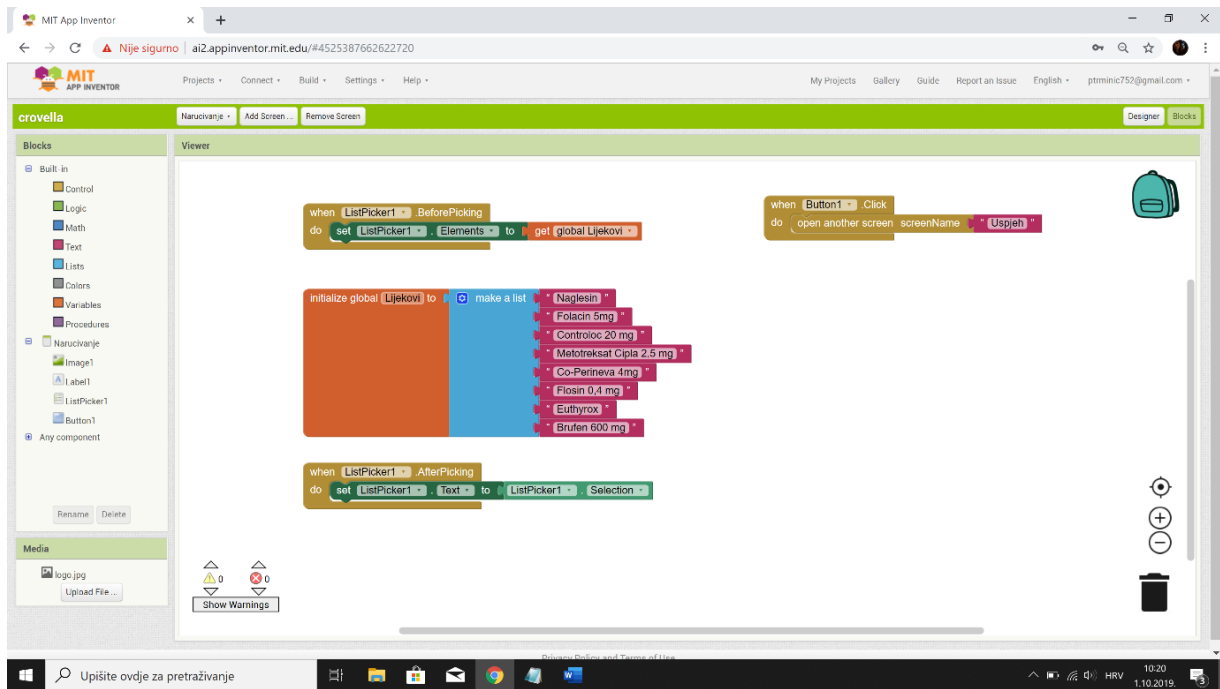


Slika 14 - UXPin naručivanje lijeka

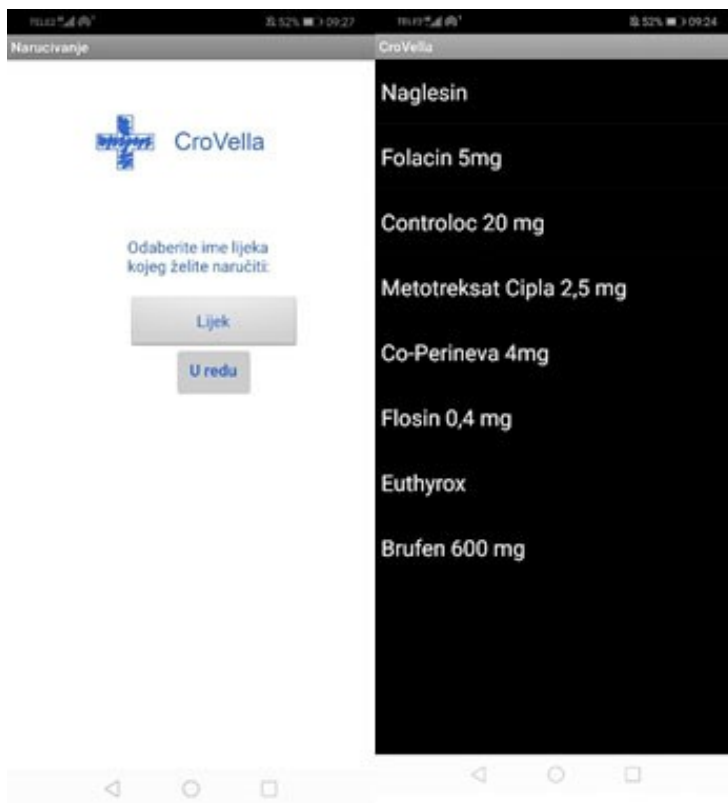
Trebalo bi postaviti i ovdje SQL upit u kojem bi u bazi podataka trebalo provjeriti je li isteklo 180 dana od valjanosti ponavljajućeg recepta. Ukoliko nije isteklo, lijek će biti ponuđen u izborniku. U suprotnom, lijek neće biti ponuđen iz razloga kako ne bi došlo do iskorištavanja sustava i preprodavanja lijekova.

Zahtjev za lijekom se provjera u bazi podataka, te ukoliko je sve u redu, zahtjev je automatski odobren od strane liječnika i zahtjev se zatim prosljeđuje u ljekarnu gdje korisnik sa važećom zdravstvenom karticom preuzima lijek.

Pomoću vizualnog programiranja (Slika 15) inicijalizira se globalna varijabla „Lijekovi“ u koju je spremljena lista lijekova, neki od navedenih su: Naglesin, Brufen 600 mg (Slika 16). Odabir je temeljen na pacijentovom kartonu, međutim ovdje nije stvorena zasebna baza podataka pomoću koje bi se to izvelo. Karton pacijenta bi trebala biti nova baza podataka koja bi se povezala sa eOI-om korisnika. U toj bazi podataka bili bi navedeni samo lijekovi koji su na ponavljajući recept. Kada korisnik odabere lijek, vrijednost se postavlja na ime lijeka.



Slika 15 - VPL blokovi naručivanje lijeka

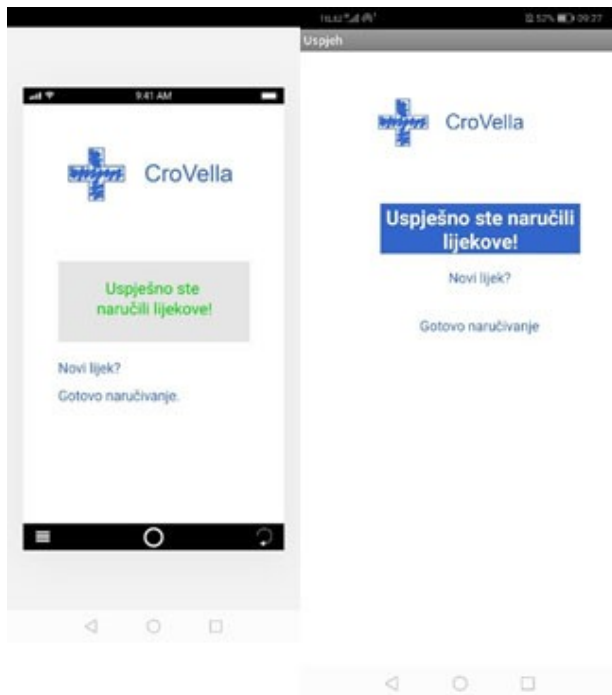


Slika 16 - VPL naručivanje lijeka

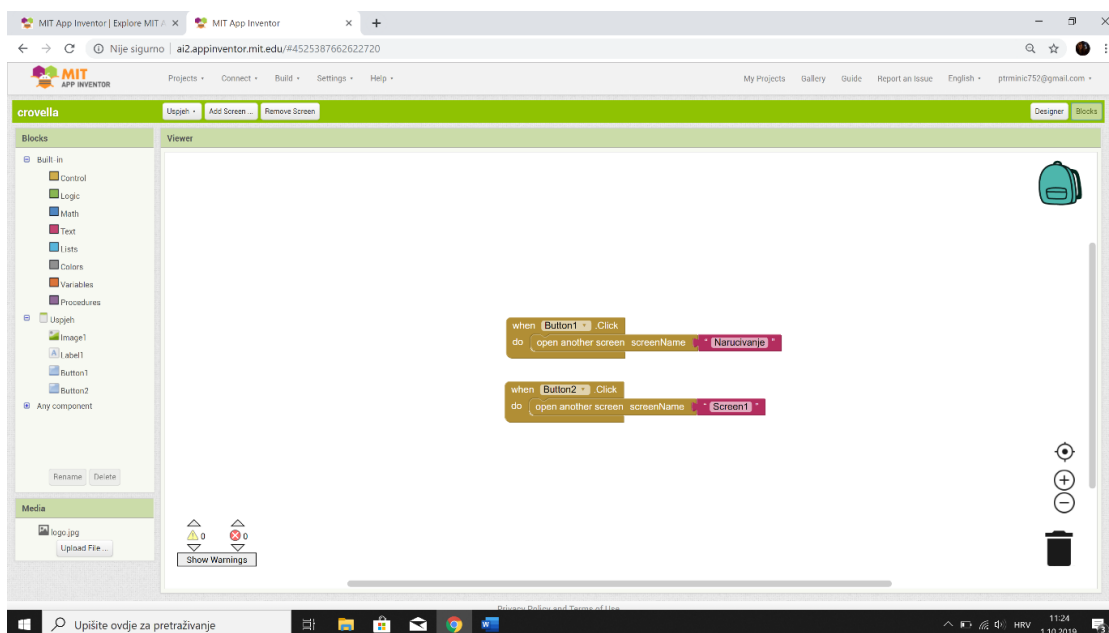
Pritiskom na gumb „U redu“ otvara se ekran uspješnosti narudžbe lijeka (Slika 17), u kodu pod imenom „Uspjeh“. Iz tog ekrana moguće je ponovno naručiti drugi lijek pritiskom na link

„Novi lijek?“ ili pritiskom na poveznicu „Gotovo naručivanje.“ automatski odjavljuje i vraća na početnu stranicu. Razlog automatske odjave iz aplikacije je stavljen kao dodatna vrsta zaštite za korisnika aplikacije.

Slika 18 prikazuje vizualno programiranje gumba „U redu“ i padajućeg izbornika sa popisom lijekova koji u ponuđeni korisniku.



Slika 17 - UXPin vs VPL uspješnost naručivanja lijeka



Slika 18 - VPL blokovi ekrana Uspješnosti

4. Poboljšanja

Predstavljeni prototip aplikacije zadovoljava osnovnu funkcionalnost naručivanja lijekova, ali još je puno mogućnosti za unaprjeđenjem.

Moguće je pojednostaviti prijavu u aplikaciju na način da se na prvom ekranu postavi tipkovnica u koju bi korisnik upisivao pin koji je postavio na početku registracije kao svoju lozinku. Tipkovnica ne bi bila fiksna, već bi se pozicije brojki mijenjale svakom prijavom za dodatnu sigurnost.

Dodatna funkcionalnost unutar aplikacije koja bi se mogla napraviti je zahtjev za uputnicama. Pacijenti bi mogli poslati zahtjev doktoru za uputnicu kojom bi dobili pravo na pregled kod nekog drugog doktora i/ili specijalista. Njihova uputnica bi bila automatski poslana postojećim sustavom u našem zdravstvu. Također bi se smanjile gužve unutar ordinacija jer ne bi bilo potrebe za osobnim dolaskom u ordinaciju da se podnese zahtjev i primi uputnica.

Također mogli bi se podaci o lokaciji i liječniku pohraniti u profilu pacijenta tako da ne mora svaki put upisati svoju lokaciju i doktora kod kojeg želi naručiti lijek. Trebalo bi omogućiti promjenu podataka u profilu pacijenta tako da može mijenjati lokaciju odnosno odabranog liječnika kad to želi. Mogla bi se staviti i dodatna funkcija promjena lozinke unutar tog izbornika. Korisnik bi u pravilu tada mogao pristupiti podacima svog profila koji se nalazi unutar nove baze podataka koja bi bila proširena tako da se osim osnovnih podataka, a to su eOI, e-mail, lozinka korisnika i prebivalište dodaju i podaci o odabranom liječniku.

Pomoću vizualnog programiranja u aplikaciju se može integrirati karta, odnosno lokacije ljekarni u neposrednoj blizini. Korisnik bi mogao s obzirom na lokaciju na kojoj se trenutno nalazi vidjeti najbliže ljekarne u okolici. Također, bilo bi ponuđeno i mogućnost ruti kako doći do neke određene ljekarne koju je odabrao korisnik te bi bilo prikazano i okvirno vrijeme koje bi korisniku trebalo da dođe do željenog cilja.

Mogao bi se ugraditi i podsjetnik u obliku kalendara koji bi sadržavao datuma isteka uputnice i/ili recepta. Obavijestio bi korisnika u obliku alarma na taj datum kada mu ističe uputnica za pregled i/ili za recept nekog lijeka u ljekarni te onda može pomoći aplikacije ponovno produžiti ako je potrebno.

5. Zaključak

Glavna prednost razvoja ovakve aplikacije bi bila olakšanje odnosa doktora i pacijenta. Smanjilo bi se opterećenje telefona doktora i samih ambulanti. Bilo bi brži pristup potrebnim lijekovima i ne bi trebali misliti na radno vrijeme doktora već bi se lijek mogao naručiti u bilo koje doba dana i podignuti ga odmah ili idući dan.

Korištenjem ovakve aplikacije bi bio još jedan kvalitetan način poboljšanja zdravlja ljudi na način da bi smanjili stres i opterećenje samog zdravstvenog sustava, ali i pacijenata. Danas sve više ima informatički pismenih ljudi koji koriste mobilne aplikacije te bi ovakav pristup bio dio rutine koja bi zahtijevala još veću brigu o našem zdravlju, što automatski povlači i bolji standard života.

6. Literatura

1. <https://www.techopedia.com/definition/2953/mobile-application-mobile-app>, pristupano 14.08.2019.
2. <https://hr.wikipedia.org/wiki/Aplikacija>, pristupano 14.08.2019.
3. <https://buildfire.com/app-statistics/>, pristupano 14.08.2019.
4. <https://www.bourncreative.com/meaning-of-the-color-blue/>, pristupano 15.08.2019.
5. <https://lab.miletic.net/hr/nastava/materijali/openssl-kriptografski-algoritmi/>, pristupano 18.08.2019.
6. <https://www.cis.hr/www.edicija/Bruteforcenapadi.html>, pristupano 18.08.2019.
7. <https://analissa-digital.hr/vrste-mobilnih-aplikacija/>, pristupano 29.09.2019.
8. https://en.wikipedia.org/wiki/Mobile_app, pristupano 29.09.2019.
9. https://en.wikipedia.org/wiki/Mobile_operating_system, pristupano 29.09.2019.
10. <https://bim-hrvatska.hr/vizualno-programiranje-kaopomoc-projektiranju/>, pristupano 30.09.2019.
11. https://en.wikipedia.org/wiki/Visual_programming_language, pristupano 30.09.2019.
12. https://yt3.ggpht.com/a/AGF-l78fi-isuXCGAFZqCksZfHZUeCG-_DNHr8zhLQ=s900-c-k-c0x0000000-no-rj-mo, preuzeto dana 01.10.2019.

7. Popis slika

Slika 1 - App Inventor for Android logo	6
Slika 2 - Plava boja (#3366cc)	7
Slika 3 - Logo aplikacije CroVella	7
Slika 4 - UXPin početna stranica vs VPL početna stranica.....	8
Slika 5 - VPL blokovi početne stranice	9
Slika 6 - UXPin registracijski obrazac vs VPL registracijski obrazac	10
Slika 7 - VPL blokovi registracijski obrazac.....	11
Slika 8 - UXPin prijavni obrazac vs VPL prijavni obrazac	12
Slika 9 - Poruka o neuspjehu prijave	13
Slika 10 - VPL blokovi prijavnog obrasca.....	13
Slika 11 - UXPin ekran Lokacija - Liječnik sa izbornicima	14
Slika 12 - VPL ekran Lokacija - Liječnik sa izbornicima.....	15
Slika 13 - VPL blokovi ekrana Lokacija – Liječnik	16
Slika 14 - UXPin naručivanje lijeka.....	17
Slika 15 - VPL blokovi naručivanje lijeka	18
Slika 16 - VPL naručivanje lijeka	18
Slika 17 - UXPin vs VPL uspješnost naručivanja lijeka.....	19
Slika 18 - VPL blokovi ekrana Uspješnosti.....	19