Sustav za geolociranje, fotografiranje i prijavu problema u okolišu

Bartaković, Ivan

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:137:355161

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-24**

Repository / Repozitorij:

Digital Repository Juraj Dobrila University of Pula

Sveučilište Jurja Dobrile u Puli
Fakultet informatike u Puli

IVAN BARTAKOVIĆ

SUSTAV ZA GEOLOCIRANJE, FOTOGRAFIRANJE I PRIJAVU PROBLEMA U OKOLIŠA

Diplomski rad

Pula, rujan 2020.

Sveučilište Jurja Dobrile u Puli
Fakultet informatike u Puli

IVAN BARTAKOVIĆ

SUSTAV ZA GEOLOCIRANJE, FOTOGRAFIRANJE I PRIJAVU PROBLEMA U OKOLIŠA

Diplomski rad

JMBAG:0303046234, redoviti student
Studijski smjer: Informatika

Predmet: Izrada informatičkih projekata
Znanstveno područje: Društvene znanosti
Znanstveno polje: Informacijske i komunikacijske znanosti
Znanstvena grana: Informacijski sustavi i informatologija

Mentor: doc. dr. sc. Siniša Sovilj

Pula, rujan, 2020.

IZJAVA O AKADEMSKOJ ČESTITOST

Ja, dolje potpisani _____, kandidat za magistra _____ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Diplomskog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student _____

U Puli, _____, _____ godine

IZJAVA

o korištenju autorskog djela

Ja, _____ dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom _____

_____

_____ koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, _____ (datum)

Potpis

Sadržaj

1. UVOD ... 1

2. O SUSTAVU ... 2

3. MOBILNA I WEB RAZVOJNA OKRUŽENJA ... 6

3.1. WEB RAZVOJNO OKRUŽENJE U JAVI ... 6
3.2. MOBILNO RAZVOJNO OKRUŽENJE - NATIVESCRIPT ... 10
3.2.1. Uvod u nativescript ... 10
3.2.2. Prednosti Nativescripta ... 11
3.2.3. Mane Nativescripta ... 12
3.2.4. Funkcionalnost Nativescripta ... 13
3.2.5. Tehnologije unutar Nativescripta ... 13
3.2.6. Vrste aplikacija razvijene s Nativescriptom ... 15
3.2.7. Usporedba Nativescripta s hibridnim razvojnim okvirima ... 16
3.2.8. Zaključak Nativescripta ... 18

4. MOBILNA APLIKACIJA SUSTAVA ZA PRIJAVU PROBLEMA U OKOLIŠU ... 19

4.1. EKRANI UNUTAR APLIKACIJE ... 19
4.1.1. Prijavni ekran ... 19
4.1.2. Početni ekran ... 22
4.1.3. Izbornik ... 28
4.1.4. Ekran postavki ... 30
4.2. DODATNE TEHNOLOGIJE I PAKETI ZA IZRADU MOBILNE APLIKACIJE ... 31
4.2.1. Dodatni nativescript paket - firebase ... 31
4.2.2. Dodatni nativescript paket – cfalert-dialog ... 32
4.2.3. Dodatni nativescript paket - geolocation ... 34
4.2.4. Dodatni nativescript paket - mapbox ... 35

5. WEB APLIKACIJA SUSTAVA ZA PRIJAVU PROBLEMA U OKOLIŠU ... 37

5.1. DIJELOVI WEB APLIKACIJE ... 37
5.1.1. Prijava u web aplikaciju ... 37
5.1.2. Moderatorski dio web aplikacije ... 38
5.1.3. Administrativni dio web aplikacije ... 42
5.2. TEHNOLOGIJE I PAKETI KORIŠTENI ZA IZRADU WEB APLIKACIJE ... 44
5.2.1. Tehnologije korištene za poslužiteljski dio web aplikacije ... 44
5.2.2. Tehnologije korištene za klijentski dio web aplikacije ... 46

6. BUDUĆNOST SUSTAVA ... 49

6.1. DODAVANJE NAGRADA ... 49
6.2. WEB STRANICA ZA REKLAMIRANJE SUSTAVA ... 49
6.3. DODATNO OPTIMIZIRANJE SUSTAVA ... 52

7. ZAKLJUČAK ... 53

LITERATURA ...

POPIS SLIKA ...

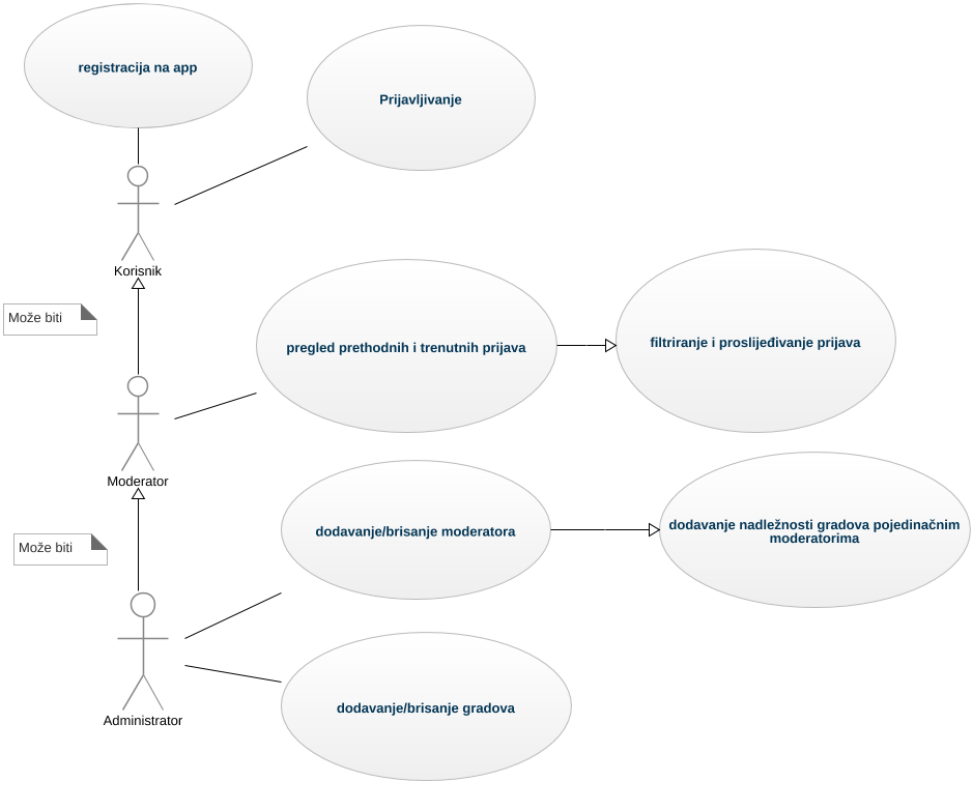
1. Uvod

Zadatak ovog diplomskog rada je razviti sustav za prijavu problema u okolišu (divlje odlagalište otpada, problem buke, bespravna gradnja, napuštene olupine, zagađenje mora, zagađenje zraka, ilegalno eksploatiranje kamena ili šljunka, nezakonita sječa stabala, ispuštanje zagađenih voda, zanemarivanje I zlostavljanje životinja, paljenje otpada, problem starih kuća kojima prijeti urušavanje, svjetlosno onečišćenje, itd.). Sustav je podijeljen u dva velika dijela. Prvi dio se sastoji od mobilne aplikacije koja služi za fotografiranje i lociranje problema u okolišu te za prosljeđivanje istih podataka u vidu predodređene prijave unutar same aplikacije. Drugi dio sustava je web aplikacija koja služi za prikaz liste problema i faze njihovog rješavanja. Web aplikacija je prvotno trebala biti izrađena u nekom od Java web programskih okvira kao što su GWT, Vaadin, Spring MVC, JSF, itd no zbog jednostavnosti aplikacije ista je izrađena na popularnim web tehnologijama koje su više odgovarale zadatku. U prvom dijelu rada biti će opisani razlozi i primjeri zbog čega su uobičajene web tehnologije više odgovarale ovom zadatku nasuprot nekom programskom okviru za Java okruženje. Zatim će se detaljnije prikazati dijagrami korištenja web i mobilne aplikacije. (dijagrami korištenja sustava, dijagrami korištenja podataka te klasni dijagrami). Nakon toga opisati će se tehnologije koje su se koristile za izradu web aplikacije (php, javascript, itd.) te web tehnologije za izradu mobilne aplikacije (Nativescript). Nakon predstavljanja tehnologija ući ćemo u dubinu objašnjavanja kompletnog sustava počevši od mobilne aplikacije te procesima koji čine istu te završavajući sa web sustavom koji se sastoji od administrativnog te korisničkog dijela. Buduća ažuriranja te poboljšanja sustava (npr. Izraditi nagradni sustav za povećanje motivacije korisnika koji će nagrađivati korisnike koji imaju najviše pozitivnih prijava) biti će detaljno obrađena u zadnjem poglavlju nakon čega će uslijediti zaključak u kojem ćemo zaključiti svrhu te budućnost ovog sustava/aplikacije.

2. O sustavu

Sustav se sastoji od dva velika dijela. Prvi dio je mobilna aplikacija koje je namijenjena svim korisnicima koji žele pomoći održati njihov grad čistim i bez zagađenja. Funkcionalnosti aplikacije su jednostavne, sastoji se od slanja prijave onečiščenja ili zagađenja na način da se pošalju slike zagađenja te lokacija na kojoj se zagađenje nalazi te opis u kojem korisnik može opisati problem te detaljnije objasniti lokaciju. Nakon slanja prijave, prijava stiže na web aplikaciju te se prikazuje određenim moderatorima za pregled prijava na web aplikaciji što je ujedno i drugi dio našeg sustava. Web aplikacija se sastoji od nekoliko dijelova. Prvi dio je administratorski dio u kojem glavni administrator može postavljati moderatore za specifične gradove koji će pregledavati prijave koje korisnici šalju. Drugi dio je moderatorski dio u kojem se prikazuju prijave koje su korisnici poslali na način da su nove prijave označene kao neobrađene te moderator imaju mogućnost pregleda prijave te mogućnost odbijanja ili prihvaćanja prijave.

Slika 1 prikazuje mogućnosti koje ima administrator te moderator web aplikacije te korisnici mobilne aplikacije.

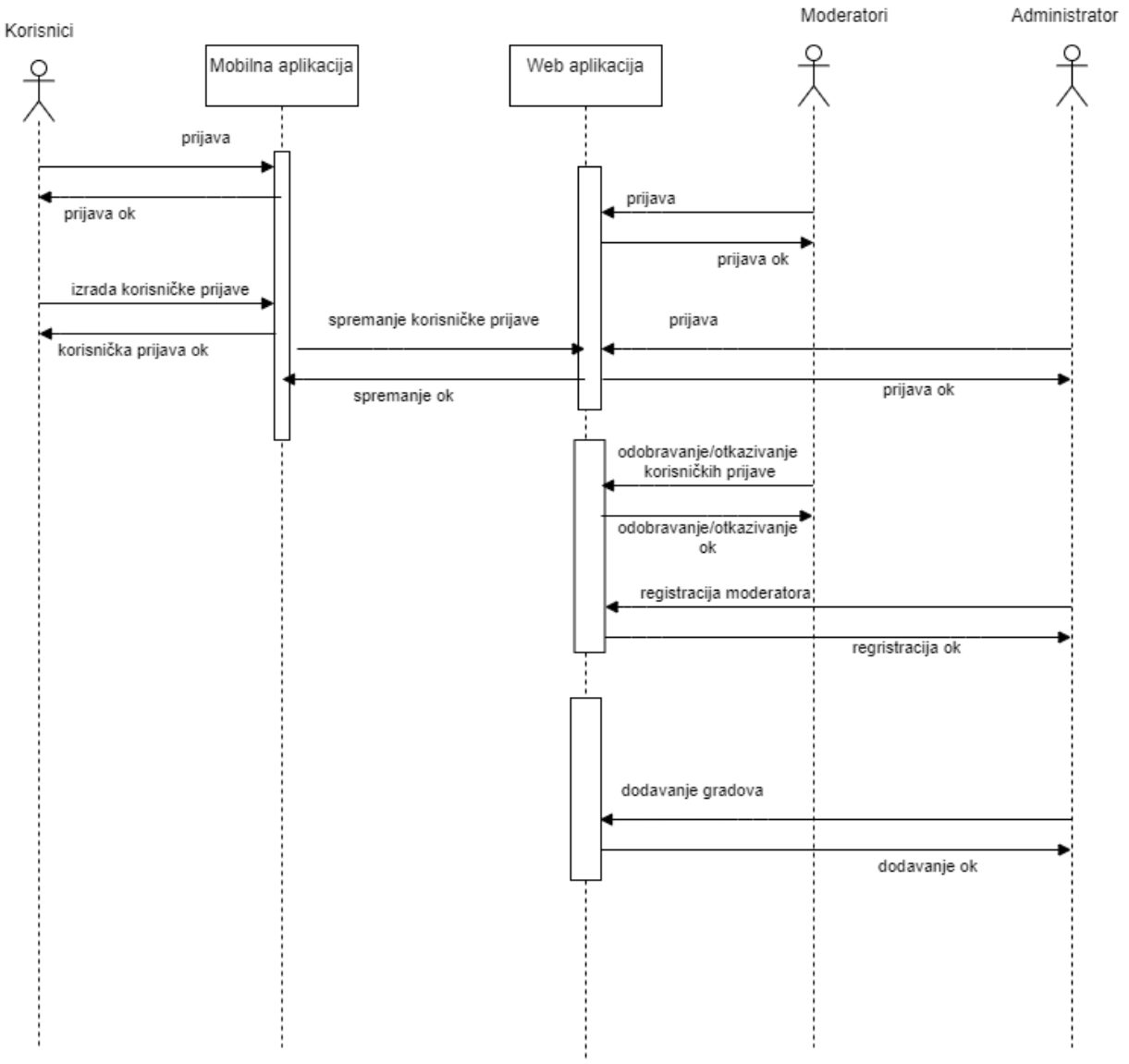


1.Dijagram korištenja sustava (eng. use-case)

Izvor: izradio autor

Administrator ima mogućnost dodavanja te brisanja gradova te dodavanja i brisanja moderatora kojima mogu dodavati nadležnosti za pojedinačne gradove. Moderatori imaju mogućnost pregleda prethodnih i trenutnih prijava te filtriranja i prosljeđivanje tih prijava dok korisnici imaju mogućnost registracije na aplikaciju te izrađivanje i slanje prijava.

Na sljedećoj slici prikazuje se dijagram korištenja podataka (eng. sequence diagram).

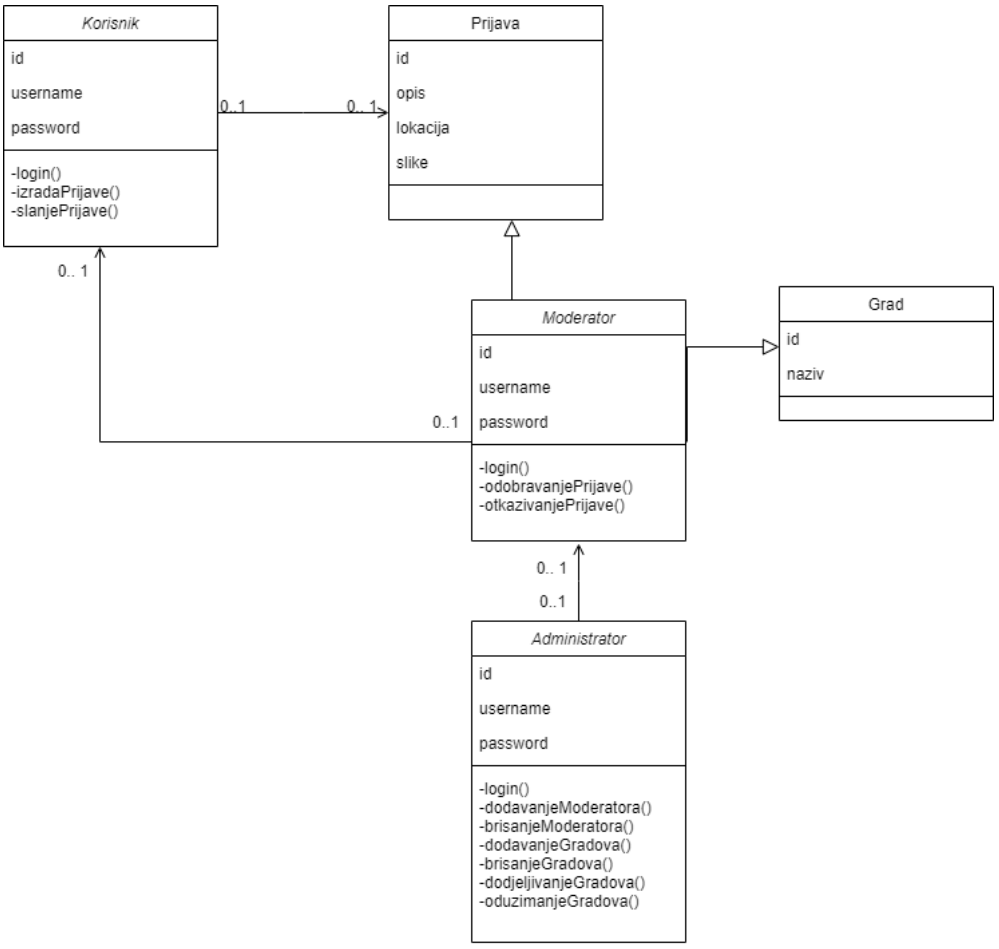


2. Dijagram korištenja podataka

Izvor: izradio autor

Dijagram korištenja podataka sa slike prikazuje na koji način korisnici koriste aplikaciju. Najprije se korisnik registrira u sustav aplikacije te se prijavljuje u nju. Nakon uspješne prijave korisnik izrađuje novu prijavu. Nakon izrađivanja i slanja prijave web aplikacija zaprima prijavu. Zatim moderatori se prijavljuju u web sustav. Nakon uspješne prijave moderator imaju mogućnost pregleda te prihvaćanja ili odbijanja prijava. Uz to administrator ima mogućnost prijavljivanja kao administrator te kao moderator, te mogućnost registriranja novih moderatora te dodavanje pojedinih gradova specifičnim moderatorima.

Slika 3 ima prikaz klasnog dijagrama (eng. class diagram) sustava.



3. Klasni dijagram aplikacije

Izvor: izradio autor

Na klasnom dijagramu možemo vidjeti da korisnik aplikacije ima mogućnost prijave u sustav te kreiranja i slanja prijave. Svaka nova prijava ima id, opis, lokaciju te id slika koje pripadaju toj prijavi. Također kreiraju se i “Gradovi”, koji su vezani za klasu moderator koji imaju mogućnost logina na web sustav te odobravanja i otkazivanja prijava. Dok klasa “Administrator” ima mogućnost glavne prijave u sustav, dodavanja moderatora, brisanja moderatora, dodavanja gradova, brisanje gradova, dodjeljivanje gradova te oduzimanje gradova.

3. Mobilna i web razvojna okruženja

U prvom potpoglavlju proći ćemo kroz web razvojno okruženje u javi te prednosti i mane istoga te ćemo ukratko objasniti koje tehnologije se koriste za izradu našeg projekta. U drugom potpoglavlju objasniti ćemo razvoj mobilnih aplikacija u okruženju Nativescript.

3.1. Web razvojno okruženje u Javi

Programski jezik Java razvijen je od strane tvrtke Sun Microsystem u 1995. te ga je kasnije u 2010-toj godini kupio Oracle. Kroz tih 25 godina Java je danas postala temelj većine umreženih aplikacija i dokazala je da je vrlo korisna za poslovne softvere, mobilne aplikacije, itd. Java se koristila za izradu mnogo bankarskih i financijskih aplikacija koje su i dalje podloga mnogim bankama posvuda u svijetu. Java se također koristi u velikim poslovnim okruženjima kao što su Amazon, Google i Facebook. Zbog navedenih razloga lako možemo zaključiti da je korištenje Jave za web razvoj moguć, no svejedno ćemo navesti nekoliko razloga kada bi trebali izabrati Javu kao web razvojno okruženje.

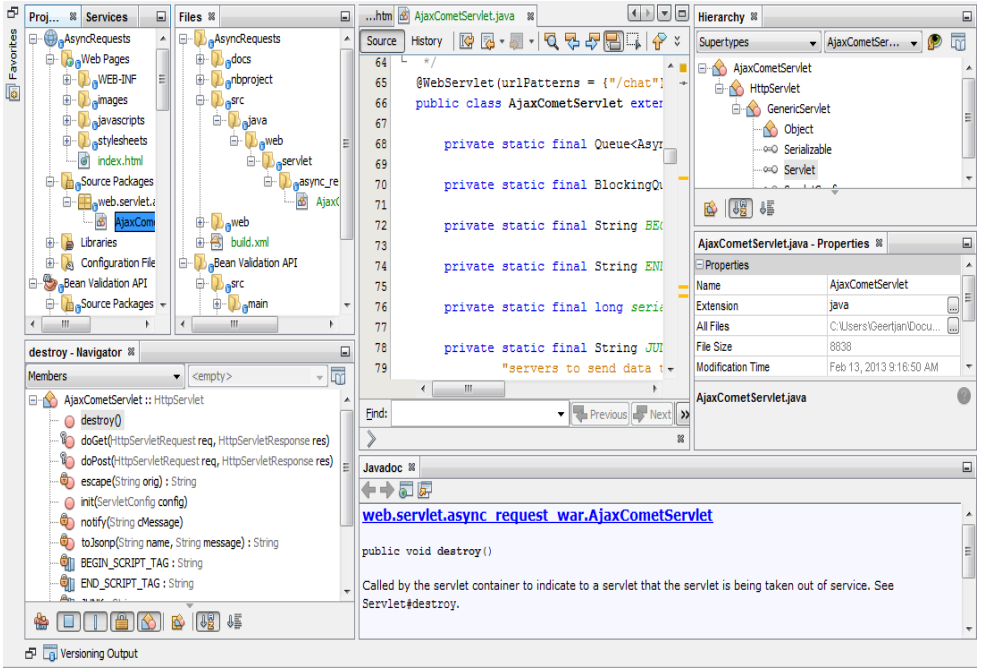
Jedan od najboljih razloga za biranje Jave kao razvojno okruženje je taj što je Java nezavisna o platformi. Java omogućuje programerima da jednom napišu ko di koriste ga tj. ponovo ga upotrebljavaju bilo gdje. Ovo je jedna od najvažnijih značajki Jave koje ju čine lagano prenosivom te se zbog toga može lako koristiti u više platformskim aplikacijama. Stoga se tvrtke ne moraju brinuti o promjeni hardvera ili operativnih sustava i više se ne moraju suočavati s gnjavažom i troškovima rješavanja nekompatibilnosti postojećeg softvera napisanog na Javi. Ovo je jedna od najvažnijih prednosti Java programskog jezika nad drugim programskim jezicima koji još uvijek ovise o platformi.

Java je veoma sigurna. Puno programera se odlučuju za razvoj web aplikacija u okruženju Java jer smatraju da je Java vrlo sigurna. Java ima brojne ugrađene sigurnosne značajke poput napredne provjere autentičnosti, kriptografije i kontrole pristupa, što ju je učinilo vrlo korisnom za web razvoj. Koristeći ove značajke, programmer može koristiti svoj sveobuhvatni API, uključujući mehanizme sigurne prijave, digitalne potpise, šifre, stvaranje prilagođenih sigurnosnih pravila i još mnogo toga.

Velike web razvojne firme također preferiraju Javu za razvoj web aplikacija jer Java ima bogat skup API-ja. To je važna prednost jer postoji toliko puno API-ja koji su besplatno dostupni Java programerima. Java pruža API-je za umrežavanje, XML raščlanjivanje, povezivanje s bazom podataka, uslužne programe i gotovo sve čega se možemo sjetiti. Sve ostalo je pokriveno s raznim bibliotekama otvorenog izvora kao što su Apache POI, Apache Commons, Jackson, Google Guava i mnoge druge.

Još jedna u nizu dobrih stvari kod Jave je to što se koristi u svim vrstama aplikacija iz stvarnog svijeta što će definitivno dati povjerenje programerima Jave. Npr. Java se koristi na mnogim poznatim web mjestima poput eBay.com, Amazon.com, LinkedIn.com i Facebook.com . Java je najbolja za razvoj bilo koje vrste poslovnih web aplikacija u gotovo bilo kojoj industriji, uključujući zdravstvo, proizvodnju i financijske usluge. Jave je čak idealna za najnovije tehnologije poput IoT-a, razvoja oblaka, AI-ja, Data Mininga, igara, pametnih uređaja i aplikacija temeljenih na AR / VR-u.

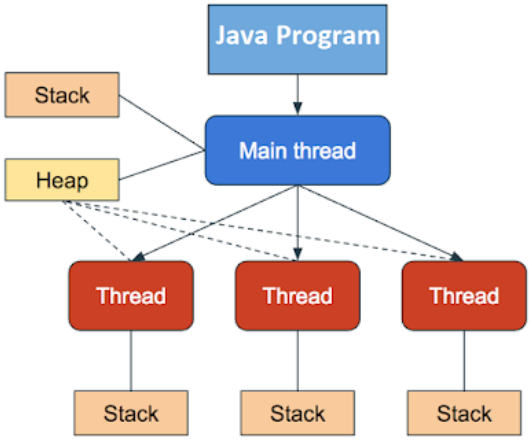
Java je jedan od najboljih jezika za razvoj web aplikacija jer ima izvrsne razvojne alate i IDE koji su slobodno dostupni na tržištu kako bi se osigurala veća produktivnost. NetBeans i Eclipse su dva najpopularnija IDE za razvoj Java aplikacija. Dostupni su i mnogi drugi besplatni alati i dodaci otvorenog koda za izvršavanje bilo koje vrste Java programiranja i testiranja Java aplikacija.



4. Netbeans Java IDE

Izvor: netbeans.org

Java ima mogućnost višedretvenosti, stoga aplikacije napisane na Javi mogu istodobno upravljati s više korisnika ili dretvi. Stoga programeri ne moraju izvršiti više kopija koda na istom hardveru jer Java pomaže stvoriti toliko dretvi unutar jedne kopije softvera koliko zahtjeva aplikacija. Višedretveno okruženje koje podržava Java također korisniku pruža brže vrijeme odaziva, manje problema, više operacija, bolje performance i brži istodobni pristup.



5. Višedretvenost u Javi

Izvor:javarevisited

Java je vrlo skalabilna. Razna softverska rješenja koja su izrađena pomoću Jave mogu lako podnijeti povećani teret posla. To mogu biti veće stope zahtjeva, veći skupovi podataka itd. Java se može prilagoditi potrebama web aplikacije i pruža programeru mogućnost da se skalira vertikalno i vodoravno prema novim zahtjevima. To znači da kad želite poboljšati vrijeme odaziva i performance svoje web aplikacije, morate dodati minimalne i jednostavne kodove.

Jedan od glavnih razloga popularnosti Jave je taj što je objektno orijentirani programski jezik. Za web razvoj ova značajka je vrlo poželjna jer je OOP aplikacijom lakše upravljati i kodirati te se sustav održava modularnim, fleksibilnim i skalabilnim. Objekti stvoreni u Javi mogu se ponovno koristiti u cijeloj aplikaciji. Također ponovna upotreba softvera smanjuje troškove razvoja aplikacija.

Jedna od najvećih blagodati Java programskog jezika je što oko sebe ima ogromnu zajednicu koja često komunicira putem internetskih foruma, grupa, kao i putem fizičkih konferencija, seminara itd. Ti članovi zajednice si međusobno pomažu, podržavaju i dijele svoje znanje, itd. Java ima toliko aktivnih foruma i organizacija otvorenog koda i nekoliko Java grupa koje pomažu u gotovo svemu što se odnosi na Javu. Postoje zajednice koje pomažu u gotovo svemu što se odnosi na Javu. Postoje zajednice koje pomažu početnicima, srednjim i čak stručnim programerima Java. Java otvoreno promovira trend uzimanja i vraćanja zajednici. Postoji toliko puno programera Java otvorenog koda koji puno doprinose zajednici kao programeri ili kao testeri.

Upravo smo opisali neke od glavnih razloga zbog kojih bi netko odabrao Javu za web razvoj. I unatoč svim prednostima koje donosi Java web okruženje, te prednosti se ukazuju samo kod specifičnih zahtjeva projekata. Kao što smo naveli Java je odlična kod velikih projekata gdje je skalabilnost bitna, odlična je u slučaju da su developeri stručni u Javi te trebaju proširiti svoj posao na Web platformu, itd.

No u našem slučaju tj. za naš projekt, razvoj u Java okruženju ne bi donio niti jednu od ovih prednosti,dapače, odabir Jave kao naše web okruženje bio bi detrimentalan izradi ovog projekta. Razlozi tome su ti što su zahtjevi naše web aplikacije veoma jednostavni te bi korištenje Jave samo zakompliciralo izradu iste, stoga smo odstupili od projektnog zadatka te izrade u Java web okruženju te su iz navedenih razloga izabrane jednostavne tehnologije za izradu web aplikacija, a to su PHP, Javascript, HTML, CSS, te pojedinačne biblioteke koje komplementiraju razvoj u ovome okruženju kao što su Bootstrap te Jquery te će sve ove tehnologije i biblioteke biti detaljnije objašnjene u budućim poglavljima.

3.2. Mobilno razvojno okruženje – Nativescript

U ovom poglavlju ćemo obraditi što je to Nativescript. Nativescript je okvir otvorenog koda za izgradnju mobilnih aplikacija s Javascriptom. Na kraju poglavlja biste trebali imati prilično dobru predodžbu o tome što je Nativescript, kako funkcionira i koje tehnologije koristi. Osim toga, također ćemo odgovoriti na uobičajena pitanja koja biste mogli imati pri istraživanju nove tehnologije, poput toga kako se ona razlikuje od alternative poput Cordove i React Native-a.

3.2.1. Uvod u Nativescript

Nativescript je okvir za izgradnju nativnih aplikacija za više platformi. Omogućuje programerima da koriste XML, CSS i Javascript za izgradnju aplikacija za Android, iOS, pa čak i za Windows Universal Platform. Za razliku od Cordove, koja koristi WebView za prikazivanje korisničkog sučelja aplikacije, Nativescript koristi mehanizam za prikazivanje izvorne platforme, što znači da pruža istinsko izvorno korisničko iskustvo.

3.2.2. Prednosti Nativescripta

- Nativescript je besplatan i otvoren. To znači da možete doprinijeti kodu i koristiti ga kako god želite, pod uvjetom da ne kršite licencu Apache 2.0

- Omogućuje vam izradu uistinu izvornih aplikacija za Android i iOS uređaje. Svaka od komponenata korisničkog sučelja koje Nativescript izlaže prevedena je u odgovarajuće izvorne komponente korisničkog sučelja.

- Omogućuje vam pristup API-ima nativne platforme putem Javascript koda. Ne morate imati znanje o Javi ili Objective-C da biste koristili API-je matične platforme, jer sve to možete napisati na Javascriptu. To znači da ako trebate pristupiti određenoj značajki uređaja, možete jednostavno naučiti kako pristupiti izvornim API-jevima s Javascriptom

- Korisnicima pruža iskustvo bliže nativnom neko od drugih hibridnih okvira za razvoj mobilnih aplikacija kao što je Cordova

- Ima podršku nultog dana za nove nativne platforme. To znači da možete odmah koristiti najnovije izvorne API-je i komponente korisničkog sučelja kad god Google ili Apple ažuriraju svoju platformu.

- Dokumentacija nudi puno informacija o tome kako započeti, temeljnim konceptima i korisničkom sučelju. Tu su i primjeri, vodiči, forumska pitanja i odgovori te uzorak aplikacije koje početnicima mogu pomoći da započnu rad s Nativescriptom

- Nudi mogućnost programiranja aplikacije s Typescriptom. Typescript je jezik koji se transpilira u Javascript i dodaje objektno orijentirane mogućnosti programiranja u Javascript.

- Bilo koja Javascript knjižnica koju možete pronaći na npm-u koja se ne oslanja na web preglednik i DOM može se koristiti u Nativescript izradi aplikacije. Primjeri takvih knjižnica uključuju i korisne biblioteke kao što su lodash i underscore.

- S nativescript komadnom linijom možete učiniti gotovo sve. Uključene su osnove poput stvaranja novog projekta, dodavanje platforme, pokretanja na uređaju i postavljanja na određenu platformu. Osim toga, također možete instalirati dodatke, debagirati pogreške u aplikaciji te prenijeti aplikaciju direktno na tržište.

3.2.3. Mane Nativescripta

- U Nativescriptu nema HTML-a i DOM-a. Tako da programeri moraju naučiti kako koristiti različite komponente okvira za izgradnju korisničkog sučelja aplikacije.

- Nedostaju potvrđeni pluginovi. Iako na npm-u postoji puno dodataka za Nativescript, nikada ne možete biti sigurni u njihovu kvalitetu.

- Programeri i dalje moraju znati izvorne API-je za Android i iOS kako bi pristupili hardveru uređaju i ostalim značajkama specifičnim za platformu.

- Zbog svoje izvorne prirode aplikacije možete testirati samo na stvarnom uređaju ili emulatoru. To čini početnu postavku za testiranje sporijom. No nakon što ga pokrenute na uređaju, onda postoji vruće preuzimanje. To znači ta se svaki put kada napravite promjenu u izvornom kodu, aplikacija odmah ponovo učitava aplikaciju kako bi odrazila promjene.

3.2.4. Funkcionalnost Nativescripta

Nativescript se sastoji od Javascript virtualnog stroja, runetimea i poveznog modula. Javascript virtualni stroj interpretira i izvršava Javascript kod. Tada povezni modul prevodi pozive u API pozive specifične za platformu i vraća rezultat pozivatelju. Pojednostavljeno rečeno, Nativescript pruža programerima način da nativnom platformom zapovijedaju putem Javascripta umjesto Objective-C-a na iOS-u ili Jave na Androidu.

3.2.5. Tehnologije unutar Nativescripta

S Nativescriptom, upotrebljavate XML za opis korisničkog sučelja aplikacije, CSS za oblikovanje i Javascript za dodavanje funkcionalnosti.
Možete koristiti i Typescript s Angular okvirom za razvoj.

Za oblikovanje Nativescript koristi samo podskup CSS-a. To znači da se ne mogu koristiti sve značajke CSS-a koje su dostupne u okruženju preglednika. Na primjer, ne možete koristiti float ili atribute položaja. Baš kao i u web pregledniku, možete dodati stilove koji se primjenjuju na cijelu aplikaciju, na određene stranice ili samo na određenu komponentu korisničkog sučelja. Ako radije želite koristiti Sass, možete instalirati dodatak Nativescript Dev Sass.

Za opisivanje strukture korisničkog sučelja koristite XML. Svaka stranica u aplikaciji trebala bi biti u svojoj XML datoteci. Nativescript dolazi s unaprijed definiranim widgetima ili komponentama korisničkog sučelja koje možete koristiti za izradu korisničkog sučelja aplikacije. Neke od ovih komponenata slične su različitim HTML elementima koje upotrebljavate u pregledniku. Na primjer, postoji komponenta Image, koja je ekvivalent img elementa, ili komponenta TextField, koja je ekvivalentna ulaznom elementu s vrstom teksta. Obrađivači događaja kao što je klik gumba dodani su u samo komponentu. Evo primjera:

```
<Button tap="sendInfo" text="Pošalji" class="prijavi"></Button>
```

6. Primjer XML koda za gumb

Izvor: izradio autor

```
exports.sendInfo = function(){
    // do something
}
```

7. Primjer Javascript koda za gumb

Izvor: izradio autor

Još jedna stvar koju treba napomenuti jest da komponente također služe kao predlošci. Ako ste upoznati s bibliotekama predložaka poput Handlebars ili Mustache, trebali biste biti upoznati sa sljedećom sintaksom:

```
<ListView items="{{ prijave }}">
  <ListView.itemTemplate>
    <StackLayout class="prijava">
      <Label text="{{ . }}"/>
    </StackLayout>
  </ListView.itemTemplate>
</ListView>
```

8. Prikaz ListView Komponente u Nativescriptu

Izvor: izradio autor

Gornji primjer na slici 5 koristi komponentu ListView. Kao što i samo ime govori, ova komponenta omogućuje stvaranje popisa. Prijave unutar vitičastih zagrada je polje definirano u Javascript datoteci te je vezano uz varijablu pri učitavanju stranice. To čini polje prijave dostupnom za upotrebu unutar XML datoteke. Na slici 6 možete vidjeti da ovo ispisuje svaku stavku u polju unutar ListViewa.

```
var prijave = ['prijava1', 'prijava2', 'prijava3'];
function pageLoaded(args){
    var page = args.object;

    page.bindingContext = {
        prijave: prijave
    }
}
```

9. Prikaz povezivanja polja s kontekstom stranice kroz javascript

Izvor: izradio autor

3.2.6. Vrste aplikacija razvijene s Nativescriptom

Zbog nativne prirode Nativescript-a, s njim možete izgraditi gotovo bilo koju vrstu aplikacije. Evo nekoliko primjera aplikacija koje možete graditi pomoću Nativescript-a:

- Aplikacije koje razgovaraju s poslužiteljem, poput aplikacija za vijesti i aplikacija za društvene mreže.
- Jednostavne igre poput šaha, tic-tac-toe-a ili flipera.
- Aplikacije u stvarnom vremenu poput aplikacija za chat ili feedova uživo.Postoji dodatak Firebase za Nativescript koji možete koristiti za implementaciju aplikacija u stvarnom vremenu.
- Aplikacije za streaming glazbe i videa uživo. Postoji dodatak za video player koji vam omogućuje reprodukciju lokalno pohranjenih videozapisa ili strujanje udaljenih videozapisa poput onih na YouTubeu
- Karte i aplikacije za geolokaciju. Postoje dodaci za Google Maps i API-ji izvornih mapa.
- Aplikacije koje pristupaju hardveru uređaja. Primjeri uključuju aplikacije za kamere i IoT aplikacije.

Što se tiče vrsta aplikacija koje možete graditi pomoću Nativescript-a, jedini ograničavajući čimbenici su izvedba i dostupnost dodataka. Pisanje izvornih mobilnih aplikacija u Javascriptu dolazi s cijenom. Ne možete očekivati da ćete izraditi aplikacije koje zahtijevaju visoke performanse. Primjeri uključuju igre sa složenom grafikom i animacijama, aplikacije s puno pokretnih dijelova i pozadinske procese. Još jedno ograničenje je dostupnost dodatka. Većina programera dolazi na Nativescript iz pozadine web razvoja. To znači da većina njih nije upoznata ili ima ograničeno znanje o API-jima nativne platforme koji bi se mogli koristiti za stvaranje dodataka za pristup hardveru uređaja ili značajkama specifičnim za platformu, kao što su kontakti ili poruke.

3.2.7. Usporedba Nativescripta s Hibridnim razvojnim okvirima

Ako niste novi u razvoju hibridnih mobilnih aplikacija, možda ste naišli na okvire poput Cordove i React Native. Nativescript je povezan s ova dva okvira time što im je cilj riješiti problem: “napišite jednom, pokrenite svugdje”. Na slici 7. možemo vidjeti usporedbu tih okvira. Da rezimiramo, Cordova je stara generacija hibridnih okvira mobilnih aplikacija. Koristi WebView za generiranje korisničkog sučelja aplikacije i pristup dodacima matičnog uređaja pomoću dodataka. React Native i Nativescript nova su generacija jer prevode vaš Javascript kod tako da ga može izvršiti nativna platforma. Možda će netko u budućnosti mogao smisliti bolji naziv za okvire poput React Native i Nativescript. Ali za sada ih klasificiramo kao “Nativni hibridni okviri” jer oboje koriste Javascript za kreiranje aplikacija i oboje nude izvorno iskustvo i performanse korisnicima.

	Cordova	React Native	NativeScript
Creator	Nitobi; Later purchased by Adobe Systems	Facebook	Telerik
UI	HTML	UI components are translated to their native counterparts	UI components are translated to their native counterparts
Can test on	Browser, emulator, device	Emulator, device	Emulator, device
Code with	HTML, CSS, JavaScript	UI components, JavaScript, subset of CSS	UI components, subset of CSS, JavaScript
Native functionality access	Through plugins	Native modules	Native API access through JavaScript
Deploys to	Android, iOS, Ubuntu, Windows, OS X, Blackberry 10	Android and iOS. Windows Universal and Samsung Tizen coming soon	Android and iOS. Windows Universal coming soon
JavaScript libraries and framework	Any front-end library or framework (e.g. Angular, Ember)	Any library that doesn't depend on the browser	Any library that doesn't depend on the browser
Coding pattern	Any front-end framework can be used to structure the code	The UI markup, JavaScript and CSS are all lumped together in a single file by default	MVVM/MVC pattern
How JavaScript code is executed	WebView	JavaScriptCore Engine to execute app code on both Android and iOS	Webkit JavaScriptCore engine to execute app code on iOS and Google's V8 JavaScript engine on Android

10. Usporedba Nativescripta s ostalim hibridnim okvirima

Izvor: code.tutsplus.com

3.2.8. Zaključak Nativescripta

U ovom poglavlju naučili smo same osnove Nativescripta. Kao smo vidjeli, Nativescript je dobar opcija za izgradnju mobilnih aplikacija pomoću vještina koje već imamo kao web programeri. Za naš projekt izabrali smo izradu aplikacije s Nativescriptom zbog jednostavnosti projekta te upoznatosti s web razvojnim tehnologijama kao što su Javascript i CSS. Nativescript nam olakšava izradu samog projekta te nam omogućava potencijalnu mogućnost da lakše objavimo aplikaciju na nekoliko tržišta odjednom kao što su Google Play te iOS bez da radimo dvije različite aplikacije kroz dva različita razvojna okvira.

4. Mobilna aplikacija sustava za prijavu problema u okolišu

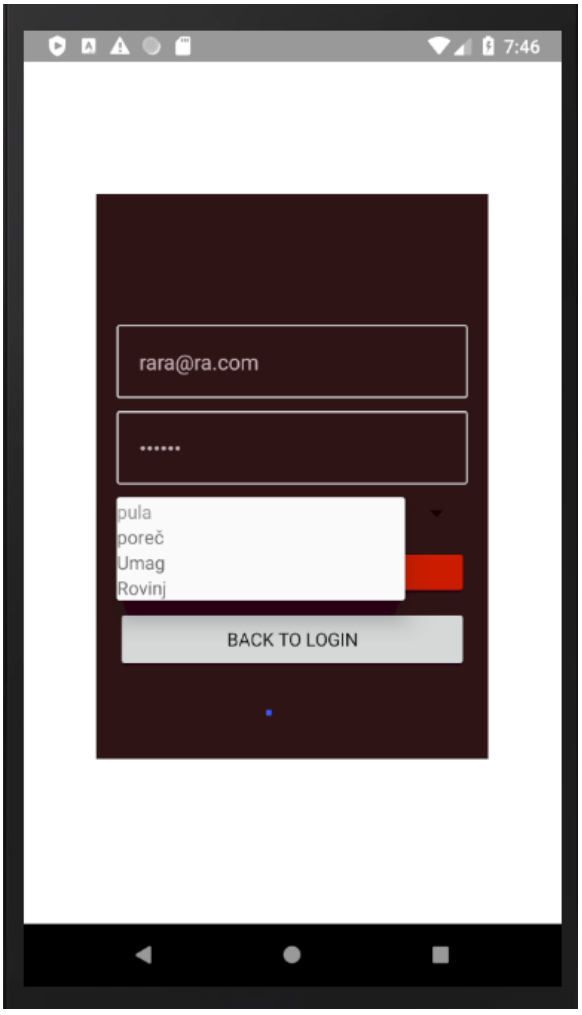
Zadatak ovog projekta je bio razvoj sustava za rješavanja problema i zagađenja u okolišu te je prvi dio tog sustava mobilna aplikacija koja će služiti za dobivanje prijava o zagađenju na način da će korisnici slati lokaciju, slike i opis problema. U ovom poglavlju ćemo detaljnije proći kroz izradu našeg rješenja za ovu mobilnu aplikaciju.

4.1. Ekrani unutar aplikacije

Aplikacija se sastoji od tri glavna ekrana. Prvi ekran je ekran prijave gdje se prijavljujemo u aplikaciju, drugi ekran je početni ekran na kojem kreiramo prijavu zagađenja dok je zadnji ekran vezan za postavke.

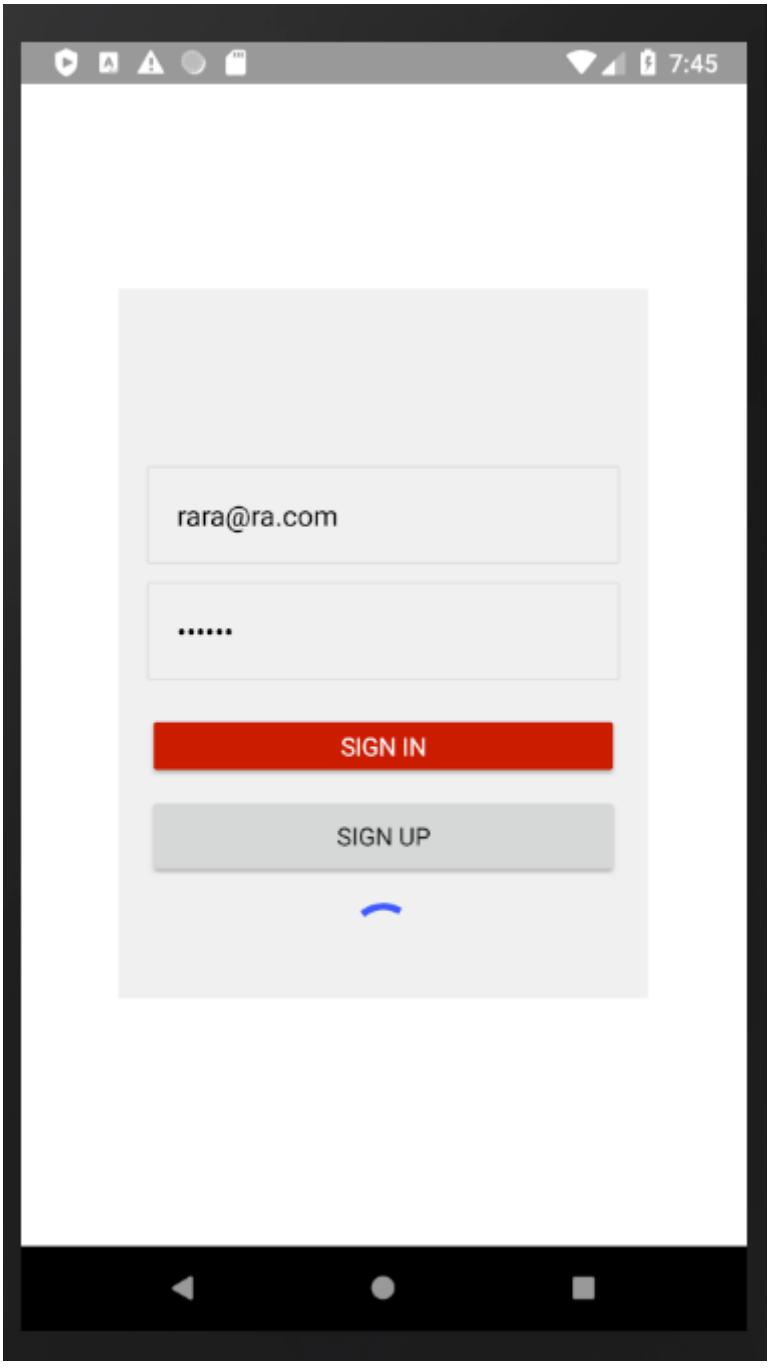
4.1.1. Prijavni ekran

Prijavni ekran se sastoji od dva dijela, prvi dio je ekran za registraciju. Ekran za registraciju se sastoji od dodavanja emaila, lozinke te padajućeg izbornika od ponuđenih gradova za koje je aplikacija dostupna.



11. Prijavni ekran mobilne aplikacije - registracija

Izvor: izradio autor

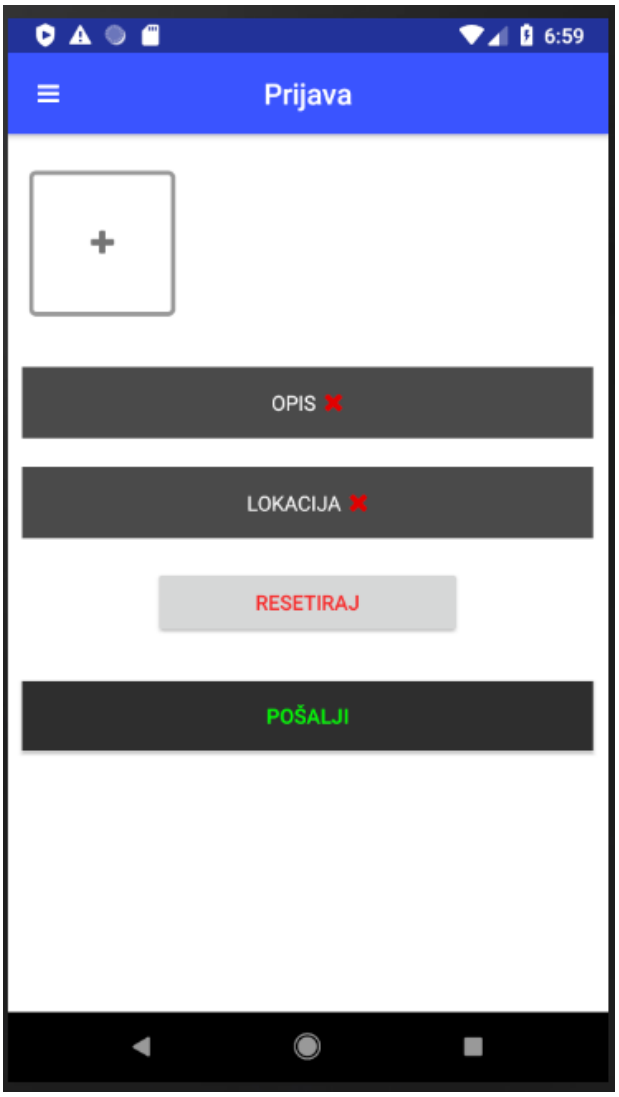


12. Prijavni ekran mobilne aplikacije - prijava

Izvor: izradio autor

Drugi dio ekrana se sastoji od mogućnosti prijave s istim podacima s kojim ste se registrirali. Nakon prijave aplikacija vas vodi na početni ekran.

4.1.2. Početni ekran



13. Početni ekran mobilne aplikacije

Izvor: izradio autor

Na prethodnoj slici možemo vidjeti početni ekran mobilne aplikacije te da se on sastoji od nekoliko dijelova. Prvi dio je prostor za dodavanje slika zagađenja unutar same prijave.

14. prozor za dodavanje slike unutar prijave

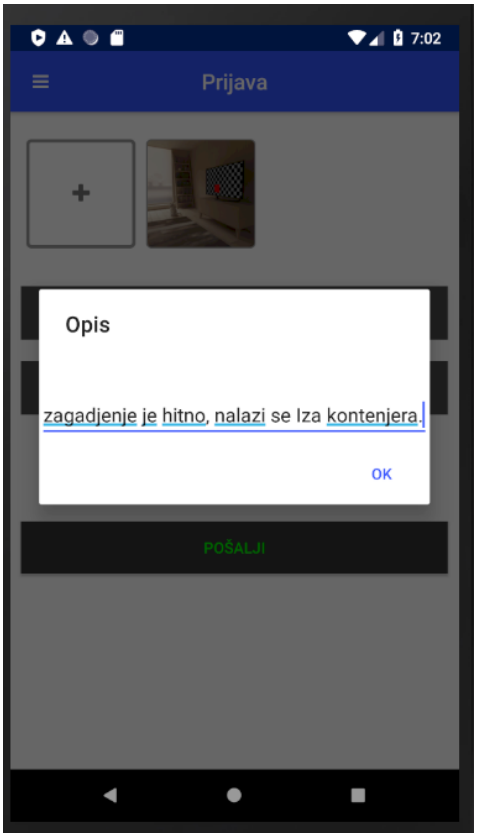
Izvor: izradio autor

Nakon pritiska na gumb za dodavanje otvara se prozor s dvije mogućnosti. Prva mogućnost je ulazak u kameru te da se uslika trenutno stanje te doda nova slika, dok je druga mogućnost dodavanje postojeće slike iz telefona.

15. Primjer dodavanja slike unutar prijave

Izvor: izradio autor

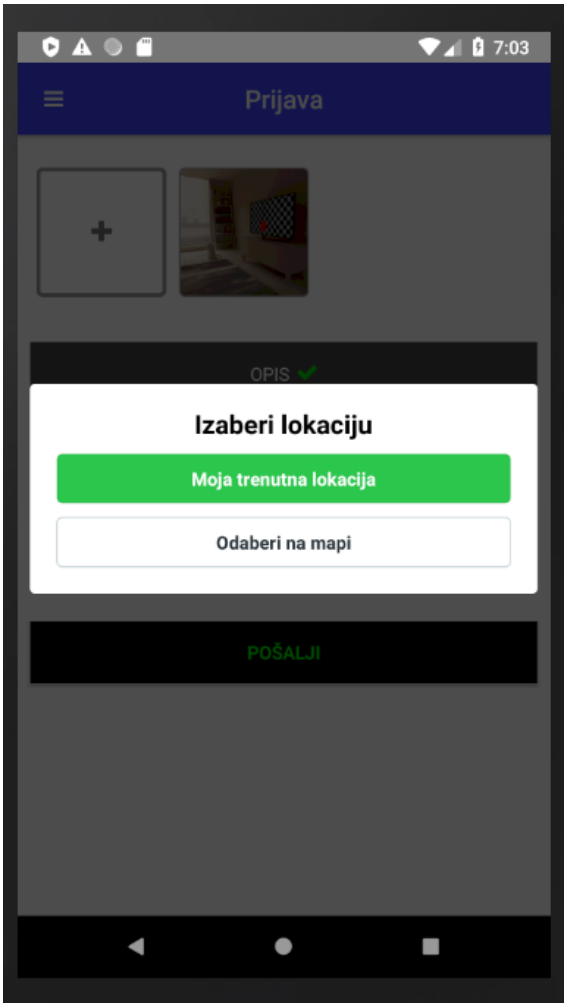
Nakon toga slijedi gumb za dodavanje opisa u kojem se nakon opisa otvara tipkovnica i prostor za pisanje opisa.



16. Primjer dodavanja opisa unutar prijave

Izvor: izradio autor

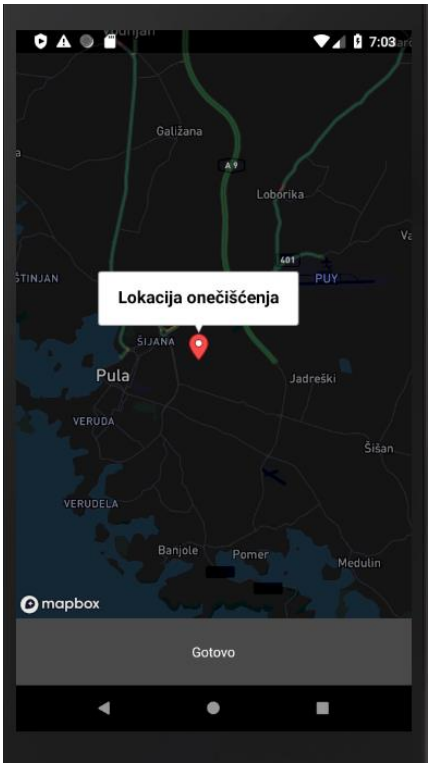
Ispod gumba za opis nalazi se gumb za dodavanje lokacije koji nakon pritiska otvara prozor sa dvije mogućnosti.



17. Prozor za dodavanje lokacije unutar prijave

Izvor: izradio autor

Prva mogućnost je za automatski odabir trenutne lokacije koju sadrži uređaj, dok je druga lokacija odabir lokacije na mapi koja izgleda ovako:



18. Prozor za dodavanje lokacije putem mape

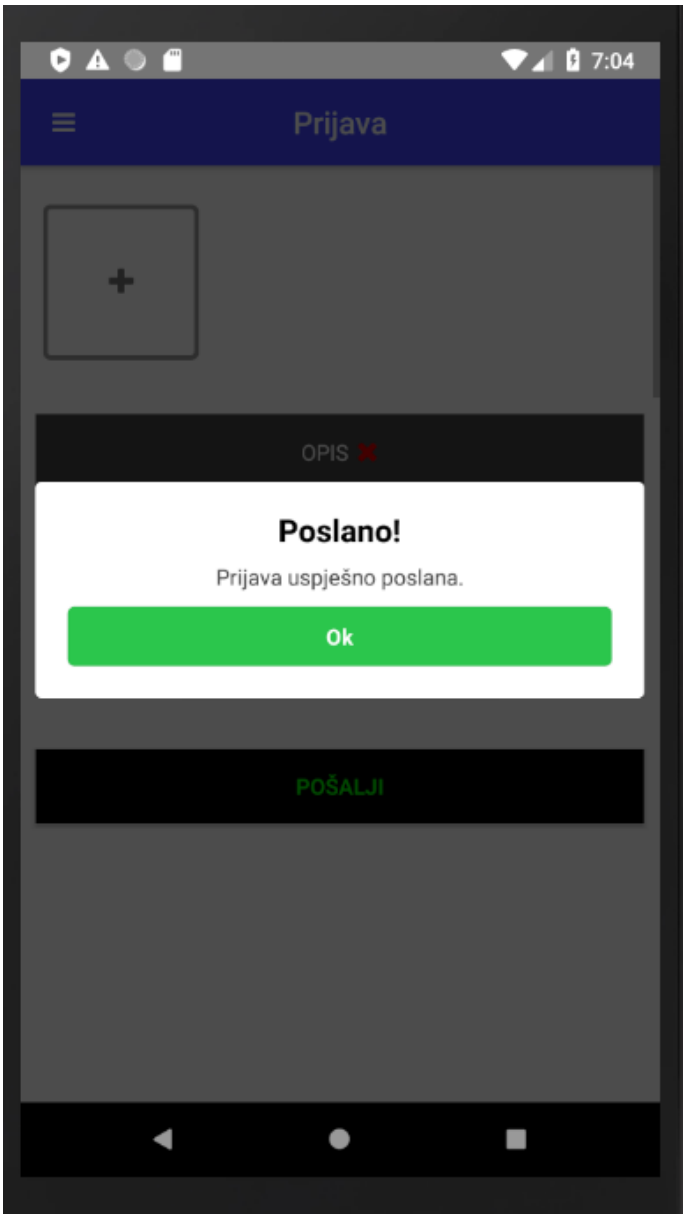
Izvor: izradio autor

Nakon gumba za lokaciju je gumb “resetiraj” koji služi za resetiranje cijele prijave. To znači nakon pritiska na taj gumb, sve slike će se maknuti iz spremnika, izbrisati će se opis te lokacija.

19. Gumb za reset prijave

Izvor: izradio autor

Zadnji element na ekranu je gumb za slanje prijave koji nakon pritiska pokrene kod za slanje prijave nakon kojeg se prikazuje loading animacija dok prijava nije završena te u slučaju uspješnog slanja dobivamo potvrdnu informaciju.

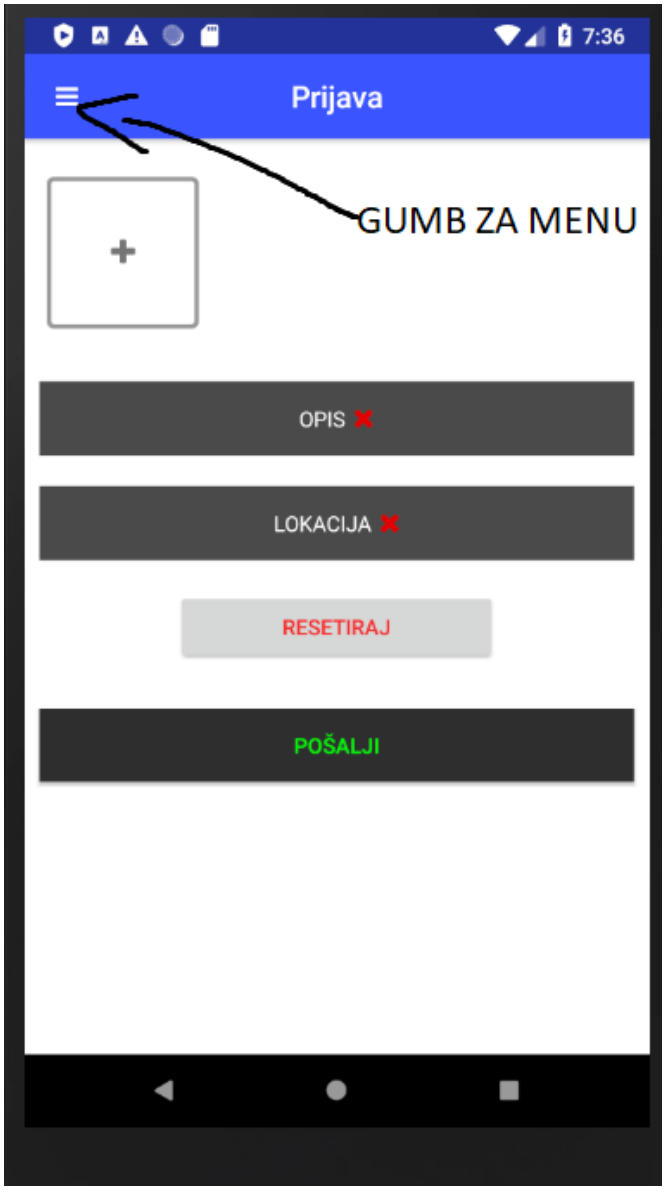


20. Primjer uspješne prijave

Izvor: izradio autor

4.1.3. Izbornik

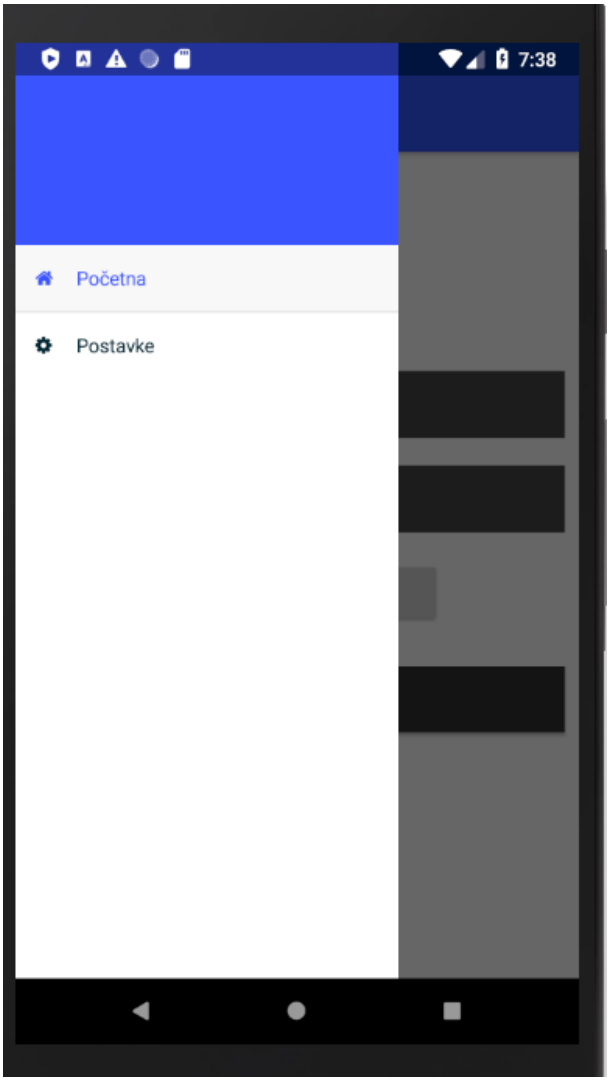
Za promjenu ekrana unutar aplikacije imamo izbornik koji sadrži gumbove koje vode na specifične ekrane.



21. Gumb za ulazak u izbornik

Izvor: izradio autor

U gornjem lijevom kutu same aplikacije nalazi se gumb za ulazak u izbornik koji izgleda ovako:

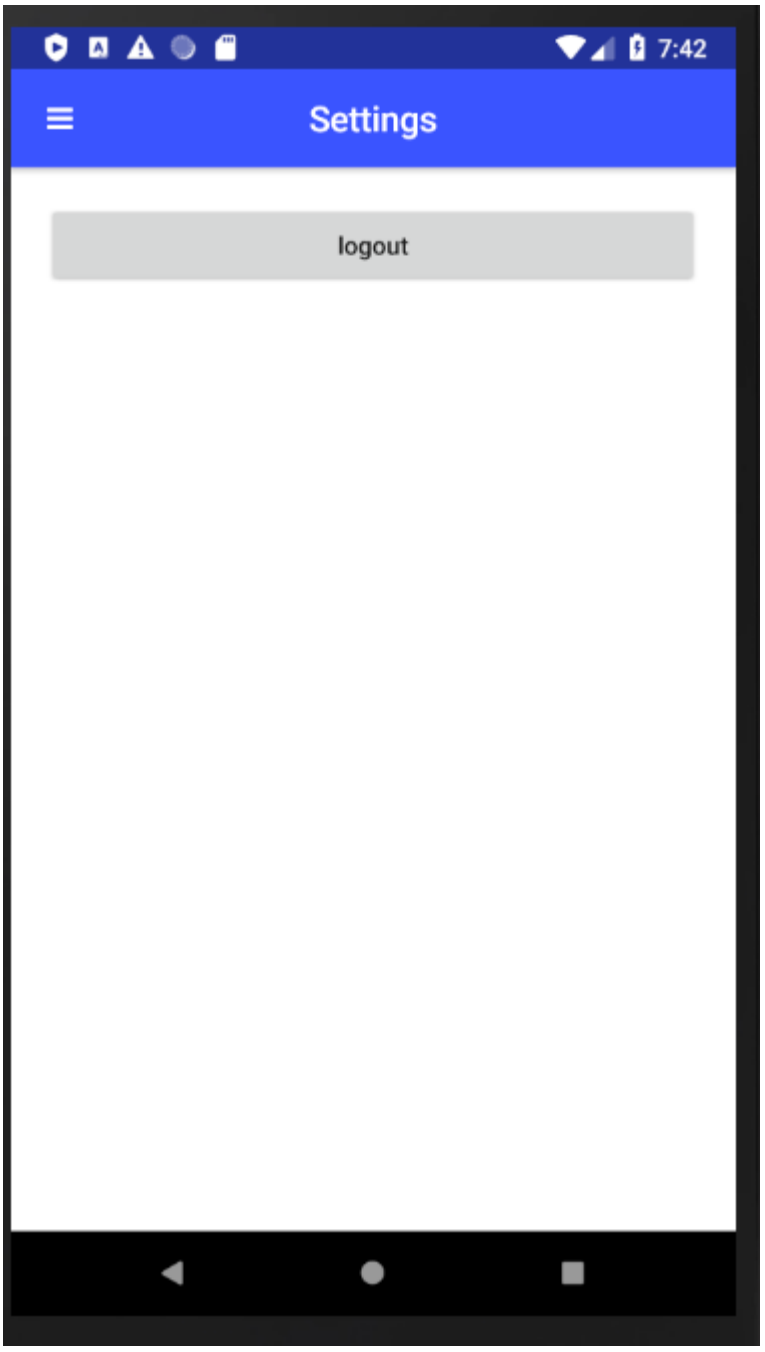


22. Izbornik unutar aplikacije

Izvor: izradio autor

Početni ekran iz izbornika smo obradili u prošlom potpoglavlju te sad slijedi ekran postavki.

4.1.4. Ekran postavki



23. Ekran postavki

Izvor: izradio autor

Ekran postavki se sastoji samo od gumba za odjavu iz aplikacije koja zatim vodi na prijavni ekran.

4.2. Dodatne tehnologije i paketi za izradu mobilne aplikacije

Uz postojeće implementirane pakete unutar samog nativescripta, postoje dodatni paketi koje možemo dodati putem npm-a.

4.2.1. Dodatni nativescript paket – firebase

Jedan od tih dodatnih paketa je paket firebase. Firebase paket se može koristiti za AdMob za dodavanje reklama unutar aplikacije, za praćenje detaljnih analitika aplikacije, za ovjeravanje autentičnosti, za bazu podataka, za “crash” analitiku, za “in-app” slanje poruka, dinamične linkove, monitoriranje performansi, bazu podataka u realnom vremenu, itd.

```
var firebase = require("nativescript-plugin-firebase");

firebase.init({
    persist: true,
    // Optionally pass in properties for database, authentication and cloud messaging,
    // see their respective docs.
    onAuthStateChanged: function (data) { // optional but useful to immediately re-logon the user when he re-visits your app
        console.log(data.loggedIn ? "Logged in to firebase" : "Logged out from firebase");
        if (data.loggedIn) {
            // eslint-disable-next-line prefer-template
            console.log("user's email address: " + (data.user.email ? data.user.email : "N/A"));
            isLoggedIn = true;
        }
        else {
            isLoggedIn = false;
            console.log("user not logged in");
        }
    }
}).then(
    () => {
        console.log("firebase.init done");
    },
    (error) => {
        // eslint-disable-next-line prefer-template
        console.log("firebase.init error: " + error);
    }
);
```

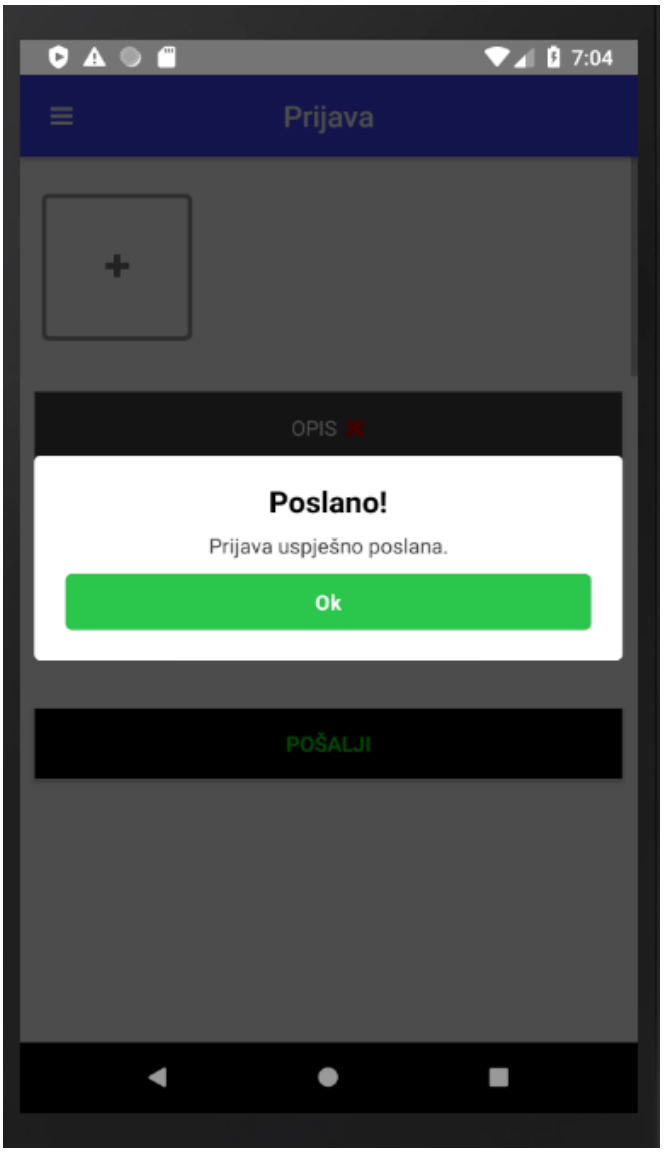
24. Inicijalizacija firebase paketa za ovjeravanje autentičnosti

Izvor: izradio autor

U našem slučaju mi koristimo ovaj dodatni paket za registraciju korisnika na aplikaciju te ovjeravanje njihove autentičnosti kod prijave u aplikaciju.

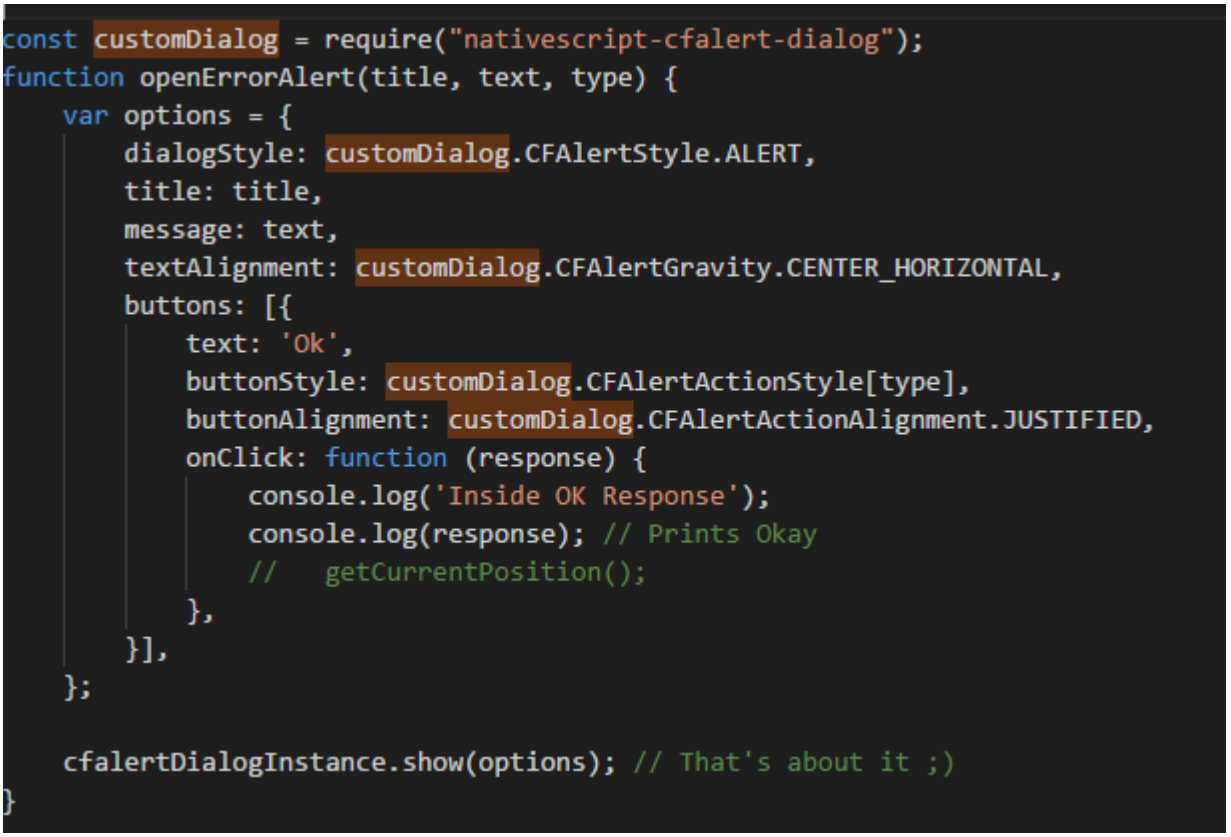
4.2.2. Dodatni nativescript paket – cfalert-dialog

Ovaj dodatni paket služi za kreiranje skočnog prozora te pruža mogućnost kreiranje istog na nekoliko načina. Za android moguće je kreirati prozor kao notifikaciju na vrhu ekrana, alertni prozor na sredini ekrana, prozor kao listu sa jednostrukim izborom ili višestrukim izborom.



25. Primjer skočnog prozora unutar aplikacije

Izvor: izradio autor



```
const customDialog = require("nativescript-cfalert-dialog");
function openErrorAlert(title, text, type) {
    var options = {
        dialogStyle: customDialog.CFAlertStyle.ALERT,
        title: title,
        message: text,
        textAlignment: customDialog.CFAlertGravity.CENTER_HORIZONTAL,
        buttons: [{
            text: 'Ok',
            buttonStyle: customDialog.CFAlertActionStyle[type],
            buttonAlignment: customDialog.CFAlertActionAlignment.JUSTIFIED,
            onClick: function (response) {
                console.log('Inside OK Response');
                console.log(response); // Prints Okay
                //   getCurrentPosition();
            },
        }],
    };

    cfalertDialogInstance.show(options); // That's about it ;)
}
```

26. Inicijalizacija cfalert-dialog paketa te primjer korištenja

Izvor: izradio autor

Na slici gore pokazujemo primjer korištenja paketa “cfalert-dialog” u našoj aplikaciji. Koristimo alert skočni prozor unutar funkcije kojoj prosljeđujemo naziv, poruku i vrstu skočnog prozora.

4.2.3. Dodatni nativescript paket – geolocation

Nativescript dodatni paket geolocation služi za dobivanje trenutne lokacije, monitoriranje kretanja, itd. U našem slučaju mi koristimo ovaj paket za dobivanje trenutne lokacije uređaja kada korisnik želi poslati lokaciju zagađenja.

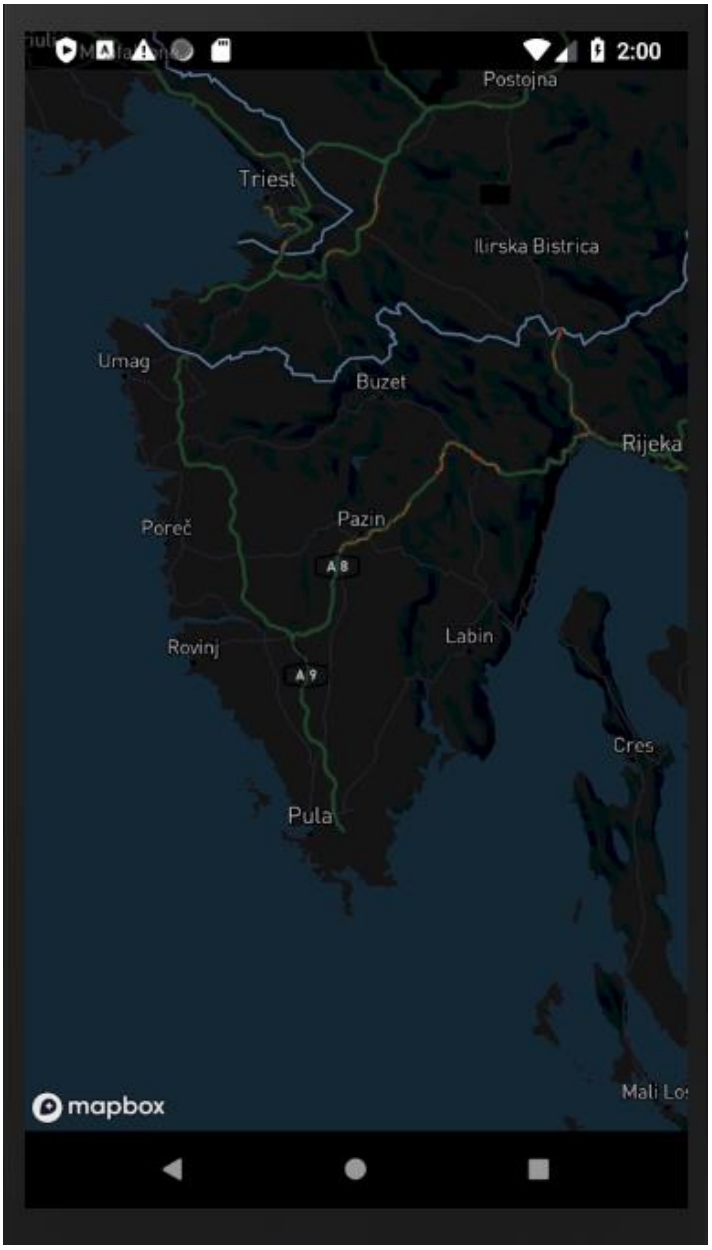
```
var Geolocation = require("nativescript-geolocation");
function getCurrentPosition(options) {
    var settings = Object.assign({
        'desiredAccuracy': 3,
        'updateDistance': 10,
        'maximumAge': 20000,
        'timeout': 20000
    }, options || {});

    var p = Promise.resolve() // Start promise chain with a resolved native Promise.
        .then(function () {
            if (!Geolocation.isEnabled()) {
                return Geolocation.enableLocationRequest(); // return a Promise
            }
            else {
                // No need to return anything here.
                // `undefined` will suffice at next step in the chain.
            }
        })
        .then(function () {
            if (Geolocation.isEnabled()) {
                return Geolocation.getCurrentLocation(settings); // return a Promise
            }
            else { // <<< necessary to handle case where Geolocation didn't enable.
                throw new Error('Geolocation could not be enabled');
            }
        })
        .then(function (loc) {
            if (loc) {
                console.log("Current location is: " + loc.latitude + ", " + loc.longitude);
                currentLocation = [loc.latitude, loc.longitude];
                console.log("currentLocation", currentLocation);
                pageModel.lokacijaDone = true;
                return loc;
            }
            else { // <<< necessary to handle case where loc was not derived.
                throw new Error('Geolocation enabled, but failed to derive current location');
            }
        })
        .catch(function (e) {
            console.error(e);
            throw e; // Rethrow the error otherwise it is considered caught and the promise chain will continue down its success path.
            // Alternatively, return a manually-coded default `loc` object.
        });

    // Now race `p` against a timeout in case enableLocationRequest() hangs.
    return Promise.race(p, new Promise(function (resolve, reject) {
        setTimeout(function () {
            reject(new Error('viewModel.getCurrentPosition() timed out'));
        }, settings.timeout);
    }));
}
```

27. Inicijalizacija geolocation paketa te primjer korištenja

Izvor: izradio autor

4.2.4. Dodatni nativescript paket – mapbox

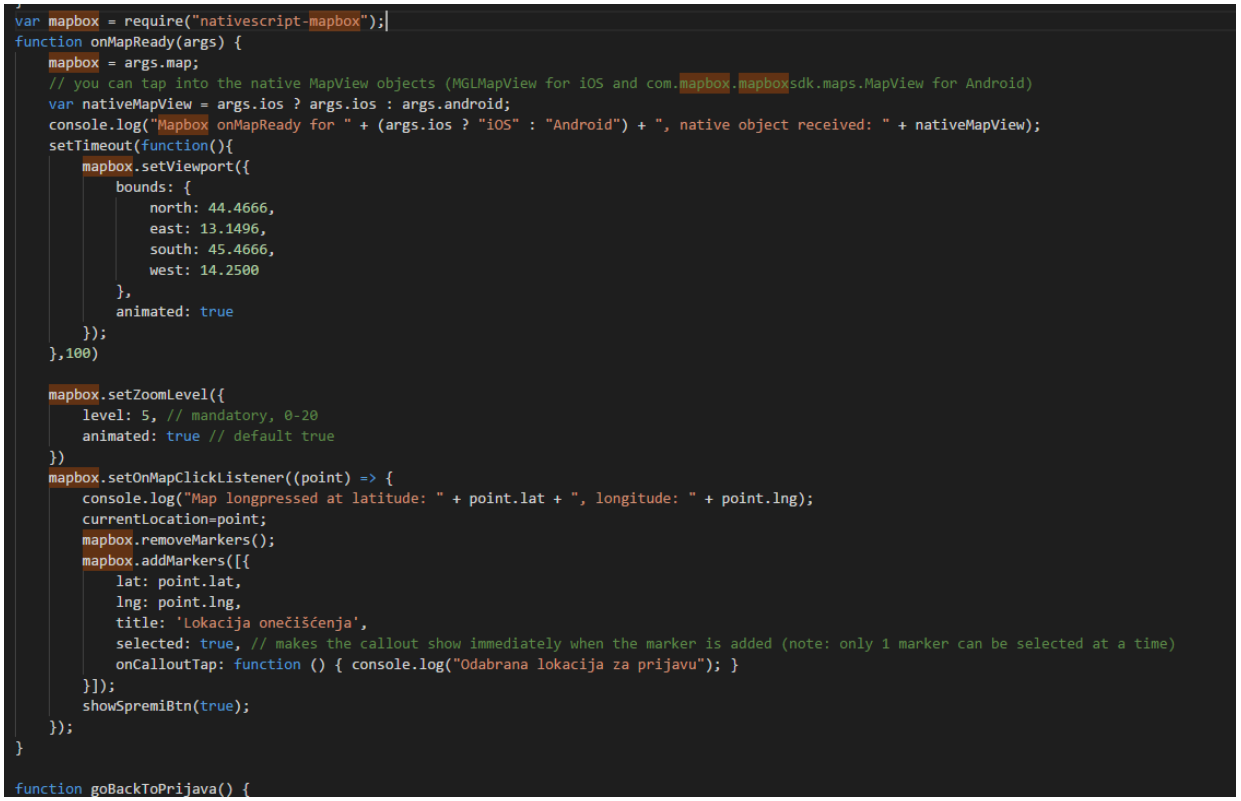
Mapbox dodatni paket služi za prikazivanje karte određenog područja po koordinatama, pruža mogućnost dodavanja markera na mapu, itd.



28. Mapbox primjer

Izvor: izradio autor

U našem projektu mi koristimo ovaj paket za pružanje korisnicima dodatne mogućnosti dodavanja lokacije zagađenja.



```
var mapbox = require("nativescript-mapbox");
function onMapReady(args) {
    mapbox = args.map;
    // you can tap into the native MapView objects (MGLMapView for iOS and com.mapbox.mapboxsdk.maps.MapView for Android)
    var nativeMapView = args.ios ? args.ios : args.android;
    console.log("Mapbox onMapReady for " + (args.ios ? "iOS" : "Android") + ", native object received: " + nativeMapView);
    setTimeout(function(){
        mapbox.setViewport({
            bounds: {
                north: 44.4666,
                east: 13.1496,
                south: 45.4666,
                west: 14.2500
            },
            animated: true
        });
    },100)

    mapbox.setZoomLevel({
        level: 5, // mandatory, 0-20
        animated: true // default true
    })
    mapbox.setOnMapClickListener((point) => {
        console.log("Map longpressed at latitude: " + point.lat + ", longitude: " + point.lng);
        currentLocation=point;
        mapbox.removeMarkers();
        mapbox.addMarkers([{
            lat: point.lat,
            lng: point.lng,
            title: 'Lokacija onečišćenja',
            selected: true, // makes the callout show immediately when the marker is added (note: only 1 marker can be selected at a time)
            onCalloutTap: function () { console.log("Odabrana lokacija za prijavu"); }
        }]);
        showSpremiBtn(true);
    });
}

function goBackToPrijava() {
```

29. Inicijalizacija mapbox paketa te primjer korištenja

Izvor: izradio autor

Kod inicijalizacije karte postavimo granične koordinate koje želimo prikazati korisniku, postavimo zoom level te slušać na kartu koji će dodati marker na koordinate koje korisnik dotakne.

5. Web aplikacija sustava za prijavu problema u okolišu

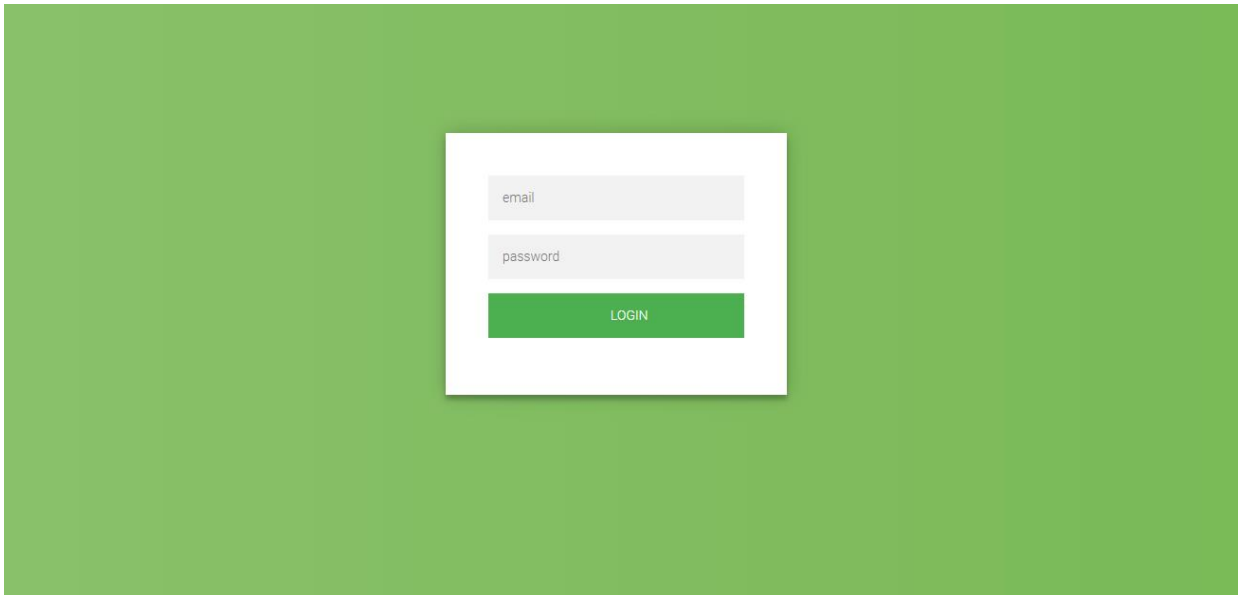
Zadatak ovog projekta je bio razvoj sustava za rješavanja problema i zagađenja u okolišu te je drugi dio tog sustava web aplikacija koja će služiti za dobivanje prijava o zagađenju na način da će zaprimati lokaciju, slike i opis problema koje su korisnici poslali preko mobilne aplikacije. U ovom poglavlju ćemo detaljnije proći kroz izradu našeg rješenja za ovu web aplikaciju.

5.1. Dijelovi web aplikacije

Web aplikacija se sastoji od nekoliko glavnih dijelova. Prvi dio je dio za autentikaciju s kojim se moderator/administrator prijavljuje u sustav i stiže na pripadajući dio aplikacije, drugi dio je moderatorski dio dok je treći dio administrativni dio.

5.1.1. Prijava u web aplikaciju

Prvi dio web aplikacije je prijava u sami sustav. Moderatori ili administrator se na ovom ekranu mogu prijaviti u web aplikaciju sa kredencijama koje su dobili od glavnog administratora sustava.



30. Prijavni dio web aplikacije

Izvor: izradio autor

Ovisno o tome da li je korisnik koji se prijavljuje moderator ili administrator, prijava će ga odvesti na pripadajući ekran ovisno o kredencijama koje je korisnik unio.

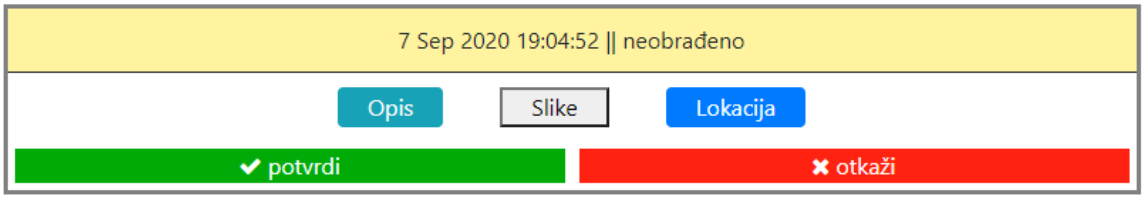
5.1.2. Moderatorski dio web aplikacije

Glavni dio web aplikacije je moderatorski dio aplikacije koji služi za filtriranje i sortiranje prijava zagađenja koje korisnici pošalju.

31. Moderatorski dio web aplikacije

Izvor: izradio autor

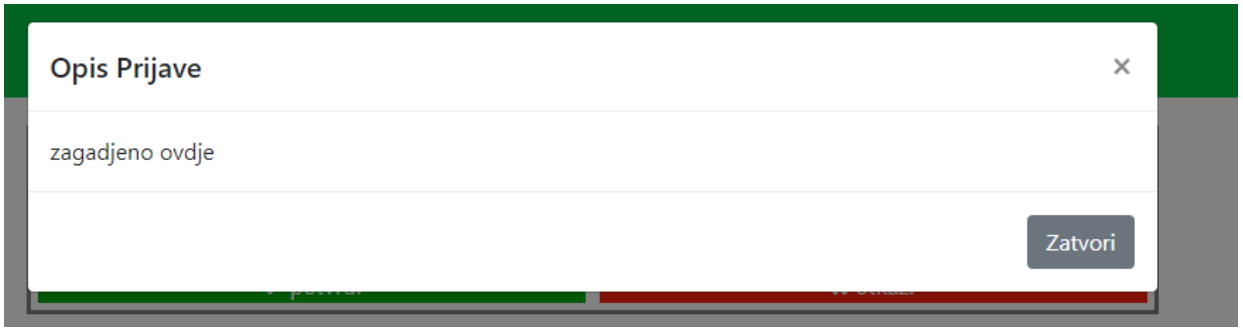
Kada prijava koju je korisnik poslao preko mobilne aplikacije stigne na web aplikaciju, prijava se smatra neobrađenom. Na slici ispod možemo vidjeti primjer neobrađene prijave.



32. Primjer neobrađene prijave

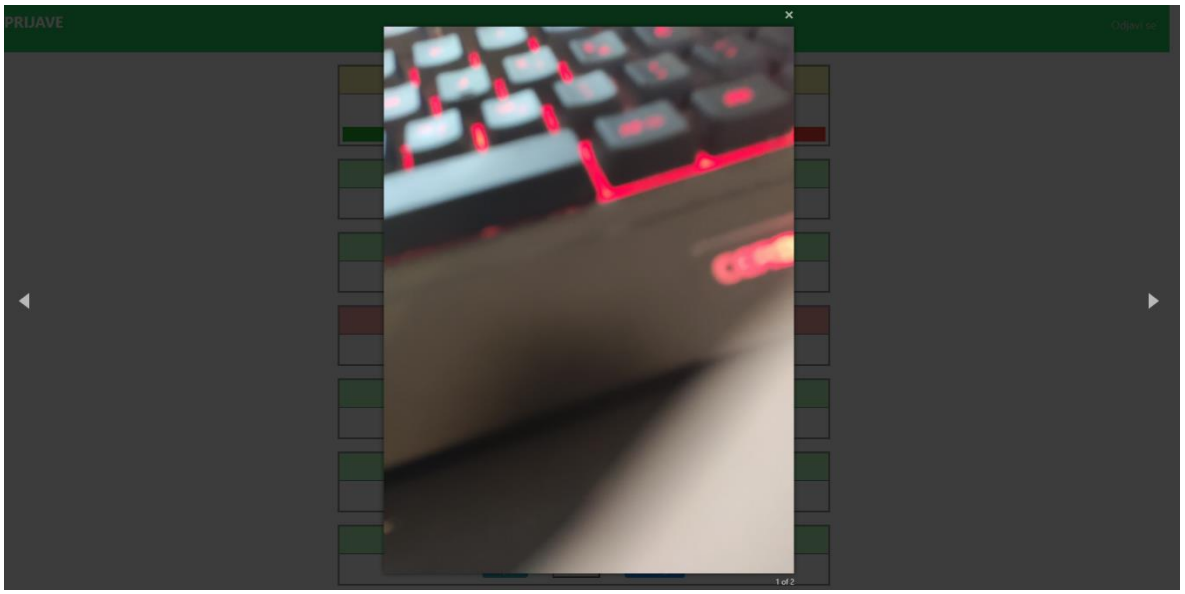
Izvor: izradio autor

Neobrađena prijava je prijava koja je tek stigla na web aplikaciju te još nije bila provjerena od strane moderatora. Moderatoru se pruža mogućnost da otvori skočni prozor sa opisom prijave, skočni prozor sa slikama zagađenja, te skočni prozor sa kartom koja prikazuje lokaciju.



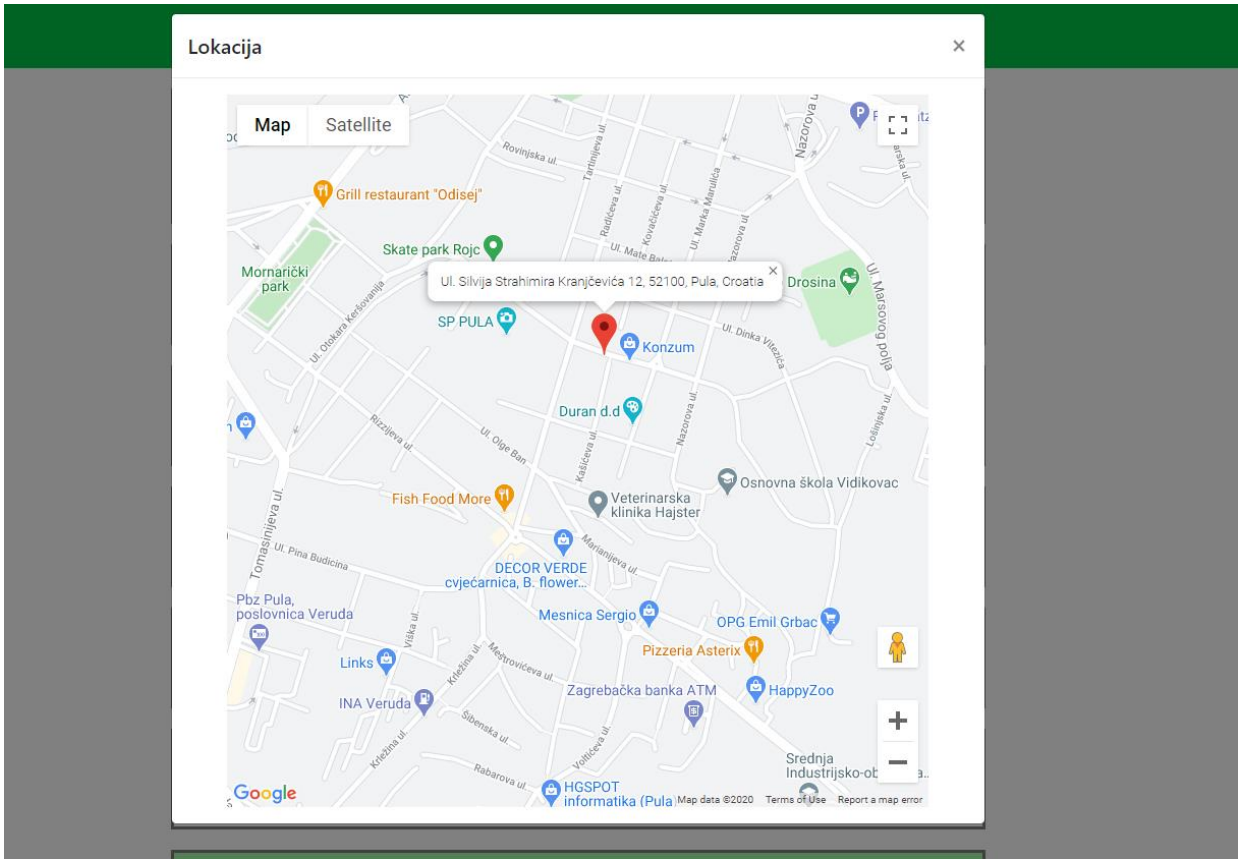
33. Primjer opisa prijave

Izvor: izradio autor



34. Primjer slika prijave

Izvor: izradio autor

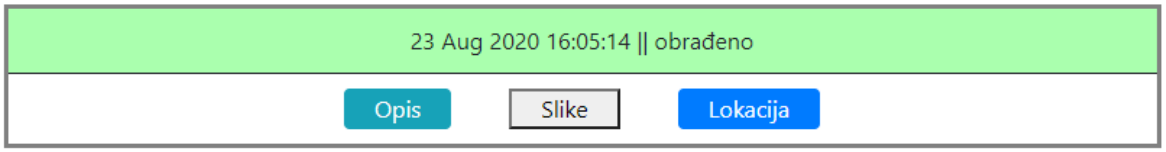


35. Primjer lokacije prijave

Izvor: izradio autor

Klikom na opis otvara se skočni prozor sa opisom prijave koju je korisnik poslao, klikom na slike otvara se skočni prozor sa slikama i mogućnost listanja istih te klikom na lokaciju otvara se skočni prozor s kartom područja te označenim markerom gdje se zagađenje nalazi.

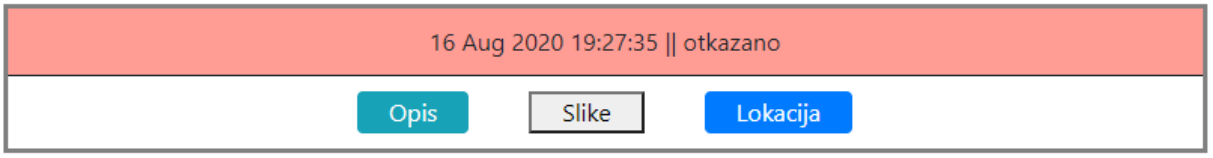
Nakon što moderator provjeri sve ove podatke, on odlučuje da li je prijava validna ili invalidna te ukoliko je validna označava je kao potvrdno obrađenom i prosljeđuje podatke i adresu nadležnim službama koje će rješiti to zagađenje.



36. Primjer obrađene prijave

Izvor: izradio autor

Ukoliko je prijava invalidna moderator označava prijavu kao otkazanom.

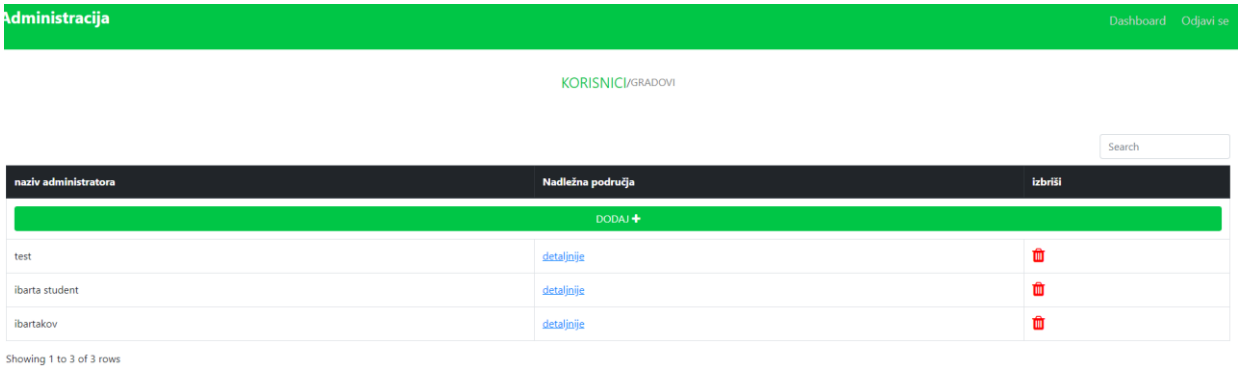


37. Primjer otkazane prijave

Izvor: izradio autor

5.1.3. Administrativni dio web aplikacije

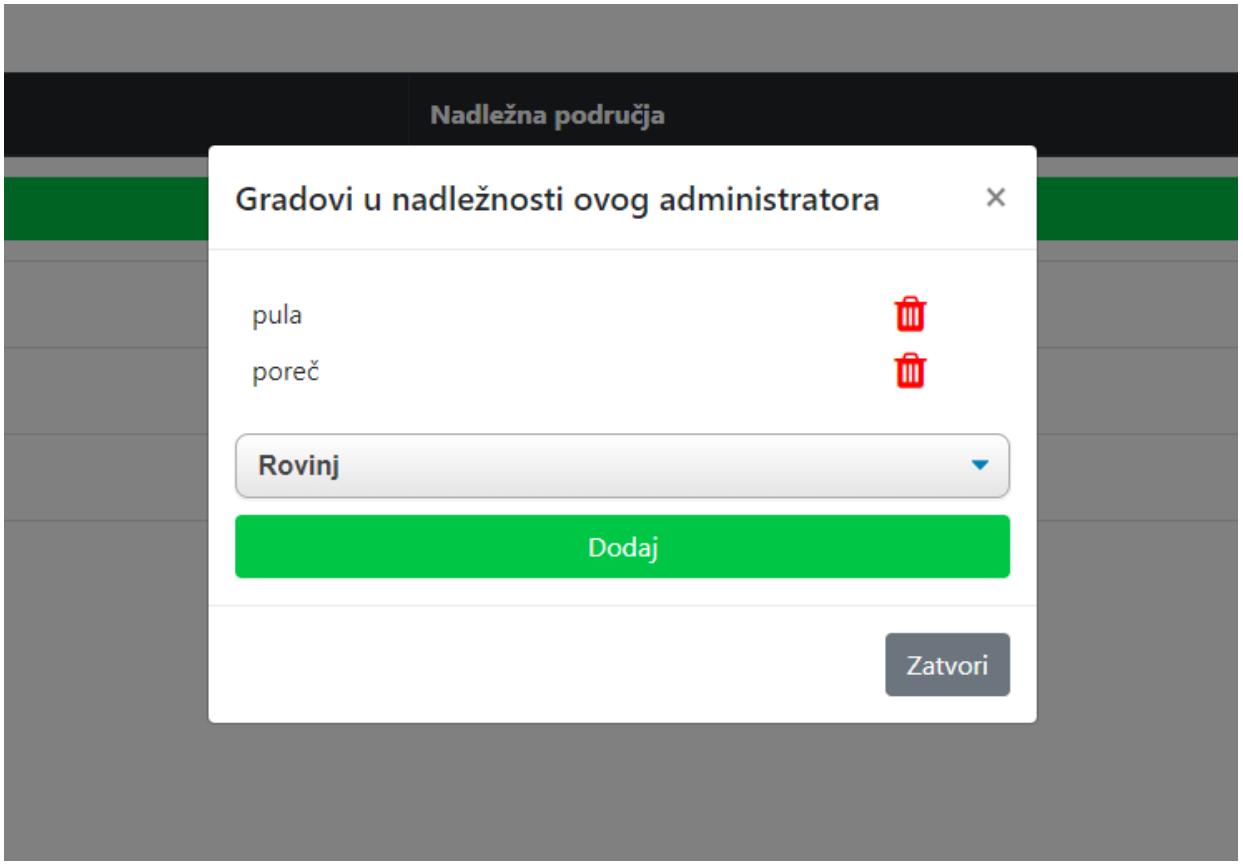
Administrativni dio web aplikacije se sastoji od dva dijela. Prvi dio je dio za dodavanje korisnika, tj. moderatora koji će se brinuti o autentičnosti prijava.



38. Administrativni dio web aplikacije

Izvor: izradio autor

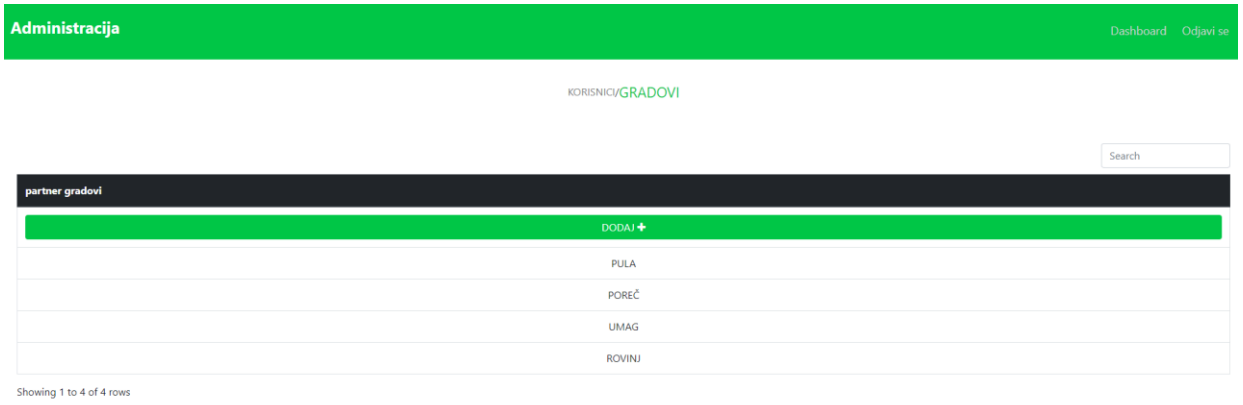
Administrator može na ovoj stranici dodati nove moderatore te im dodijeliti specifične gradove u njihovoj nadležnosti te će oni dobivati prijave samo iz tog područja što možemo vidjeti na slici dolje.



39. Primjer dodjeljivanja gradova

Izvor: izradio autor

Drugi dio administracijskog dijela aplikacije je dio na kojem se dodaju gradovi koje će sustav podržavati.



40. Dodavanje gradova

Izvor: izradio autor

Gradovi koji se dodaju ovdje biti će dostupni kod registracije na sustav za korisnike na mobilnoj aplikaciji te će omogućiti slanje prijava za taj grad. Također ovo će omogućiti dodjeljivanje istog grada specifičnim moderatorima koji će se brinuti o autentičnosti prijava te rješavanja istih sa nadležnim službama toga grada.

5.2. Tehnologije i paketi korišteni za izradu web aplikacije

Ova web aplikacija je zbog svoje jednostavnosti izrađena pomoću php-a za poslužiteljski dio, te pomoću html-a, css-a, javascripta te dodatnih paketa za klijentski dio, te ćemo u ovom poglavlju detaljnije obraditi te tehnologije.

5.2.1. Tehnologije korištene za poslužiteljski dio web aplikacije

Zbog jednostavnosti projekta, umjesto korištenja nekog “frameworka” kao što je Laravel za php, koristili smo običnu verziju php-a. PHP je skriptni jezik opće namjene koji je posebno pogodan za web razvoj koji je široko korišten, besplatan i učinkovit te je alternativa konkurentima kao što je Microsoftov ASP.

Uz PHP korišten je “Composer”. Composer je upravitelj paketa na razini aplikacije za programski jezik PHP koji pruža standardni format za upravljanje ovisnostima PHP softvera i potrebnih dodatnih paketa.

```php
<?php
    include("conn.php");
    require 'vendor/autoload.php';
    $auth = new \Delight\Auth\Auth($conn);
?>
```

41. Inicijalizacija paketa za autentifikaciju

Izvor: izradio autor

Primjer jednog takvog dodatnog paketa kojeg smo mi koristili pomoću Composera je “delight-im/auth” koji služi za autentikaciju korisnika kod prijave na web aplikaciju.

```php
<?php
include("auth_include.php");


if($_POST['email'] && $_POST['password'] ){
    try {
    $auth->login($_POST['email'], $_POST['password']);
    if ($auth->hasRole(\Delight\Auth\Role::ADMIN)) {
        header("Location: admin.php");
        exit();
    }
        header("Location: dashboard.php");
        exit();
    }
    catch (\Delight\Auth\InvalidEmailException $e) {
        $message='Wrong email address';
        header("Location: login.php?message=$message");
        exit();
    }
    catch (\Delight\Auth\InvalidPasswordException $e) {
        $message='Wrong password';
        header("Location: login.php?message=$message");
        exit();
    }
    catch (\Delight\Auth\EmailNotVerifiedException $e) {
        $message='Email not verified';
        header("Location: login.php?message=$message");
        exit();
    }
    catch (\Delight\Auth\TooManyRequestsException $e) {
        $message='Too many requests';
        header("Location: login.php?message=$message");
        exit();
    }
}else{
     $message='Wrong Data, try again.';
     header("Location: login.php?message=$message");
     exit();

}


?>
```

42. Primjer korištenja paketa kod prijave u sustav

Izvor: izradio autor

Uz PHP koristili smo MySQL kao izbor za bazu podataka. MySQL je besplatan, open source sustav za upravljanje bazom podataka te je čest izbor baze za projekte rađene u PHP-u.

5.2.2. Tehnologije korištene za klijentski dio web aplikacije

Klijentski dio web aplikacije izrađen je pomoću HTML5, CSS-a, Javascripta, Ajax-a te dodatnih paketa kao što su Bootstrap i SweetAlert2 te ćemo pojedinačno pojasniti navedene stvari.

HTML5 je najnovija iteracija HTML-a koji se koristi za strukturiranje i predstavljanje sadržaja na World Wide Webu. To je peta i najnovija glavna verzija HTML-a te je preporučeno od strane World Wide Web Consortiuma da se koristi ova verzija. Na slici ispod možemo vidjeti primjer novog “nav” elementa koji je dodan u HTML5.

```
<nav class="header">
  <span class="logo">PRIJAVE</span>
  <div class="header-right">
    <a href="logout">Odjavi se</a>
  </div>
</nav>
```

43. Primjer korištenja “nav” taga

Izvor: izradio autor

Uz HTML5 se naravno koristi CSS kao stilski jezik koji se koristi za opis prezentacije dokumenta napisanog pomoću markup jezika koji je u ovom slučaju HTML5.

```
.header .logo {
  font-size: 25px;
  font-weight: bold;
  color:white;
}
```

44. Primjer korištenja CSS-a

Izvor: izradio autor

Kao dodatak HTML-u i CSS koristi se Bootstrap. Bootstrap je besplatni CSS okvir s otvorenim kodom usmjeren na responzivni web razvoj. Sadrži CSS i Javascript temeljene predloške za tipografiju, obrasce, gumbe, navigaciju i ostale komponente sučelja.

```
<link rel="stylesheet" href="bootstrap/css/bootstrap.min.css">
<script src="bootstrap/js/bootstrap.min.js"></script>
```

45. Inicijalizacija Bootstrap-a

Izvor: izradio autor

Jedan primjer korištenja Bootstrap-a u našem projektu možemo vidjeti na slici ispod gdje smo koristili Bootstrap klase za skočni modalni prozor koje automatski dodaju stil i funkcionalnost skočnog prozora pošto smo inicijalizirali bootstrap javascript i css datoteke.

```
<!-- Modal -->
<div class="modal fade" id="mapModal" tabindex="-1" role="dialog" aria-labelledby="mapModalLabel" aria-hidden="true">
  <div class="modal-dialog  modal-lg" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="mapModalLabel">Lokacija</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div id="map-modal" class="modal-body">
        <div id="map"></div>
      </div>
    </div>
  </div>
</div>
```

46. Bootstrap modal

Izvor: izradio autor

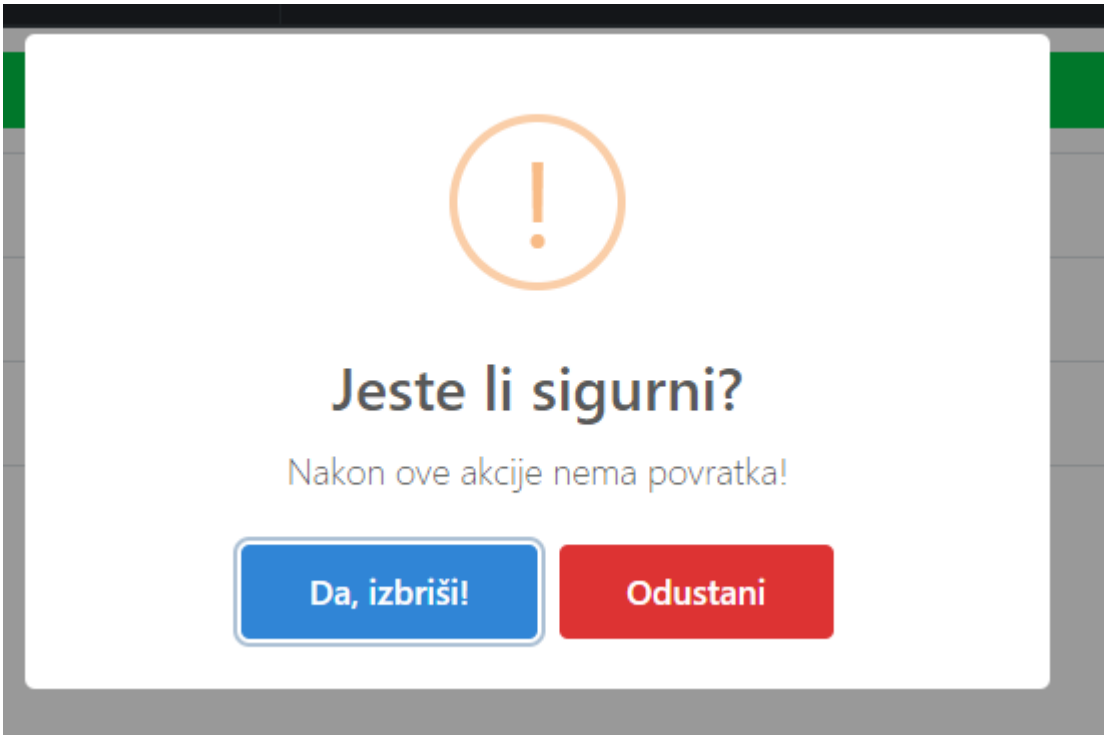
Javascript je skriptni programski jezik koji se izvršava u web pregledniku na strani korisnika. Zajedno s AJAX-om (Asynchronous javascript and XML) tehnikom omogućuje web stranicama komunikaciju sa serverskim programom, što je učinilo našu aplikaciju interaktivnijom i lakšom za korištenje.

Također kao dodatni paket koristili smo SweetAlert2 koji nam je poslužio da elegantno prikazujemo skočne poruke za potvrdu ili kao notifikaciju.

```
<script src="sweetalert2/dist/sweetalert2.all.min.js" ></script>
<script>
    function showInfoAlert(title,type){
        Swal.fire({
          type: type,
          title:title,
          timer:1000
        });

    }
```

47. Inicijalizacija SweetAlert2 paketa

Izvor: izradio autor



48. Primjer SweetAlert2 skočnog prozora

Izvor: izradio autor

Ovaj dodatni paket nudi nam mogućnost prikazivanja elegantnih skočnih prozora te u našem primjeru možemo vidjeti kako smo mi koristili ovaj paket za potvrdu kod brisanja korisnika iz sustava. Kod inicijalizacije skočnog prozora prosljeđujemo tip prozora te poruku.

6. Budućnost sustava

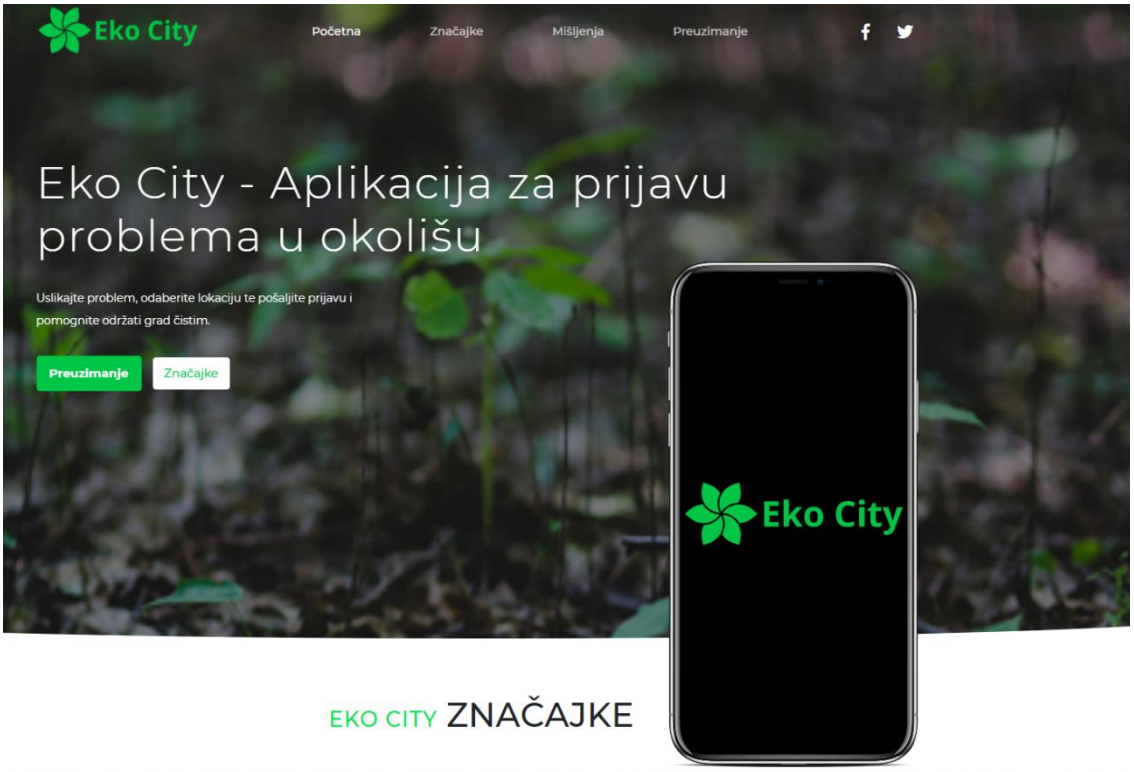
Ovaj sustav je prototip sustav koji služi kao kostur budućeg sustava. Implementirane su najosnovnije funkcionalnosti koje su esencijalne za funkcioniranje sustava. U ovom poglavlju proći ćemo kroz potencijalna unapređenja sustava ukoliko bi ovakav sustav bio stavljen na tržište.

6.1. Dodavanje nagrada

U trenutnom izdanju ovakav sustav se oslanja u potpunosti na dobru volju građana unutar grada u kojem je sustav dostupan, te je funkcioniranje sustava moguće samo ukoliko građani prijavljuju zagađenja, no ne postoji nikakva incentive za to. Ideja za incentivu je ta da dodamo nagradni sustav unutar sustava. Nagradni sustav bi funkcionirao na način da bi korisnicima za svaku uspješnu prijavu koju pošalju dobili određeni broj bodova unutar aplikacije. Također unutar specifičnih gradova, grad bi surađivao sa lokalnim firmama koji bi dodijelili kupone za kupovanje u njihovim trgovinama gradu, te bi korisnici mogli kupiti te kupone sa nagradnim bodovima unutar same mobilne aplikacije.

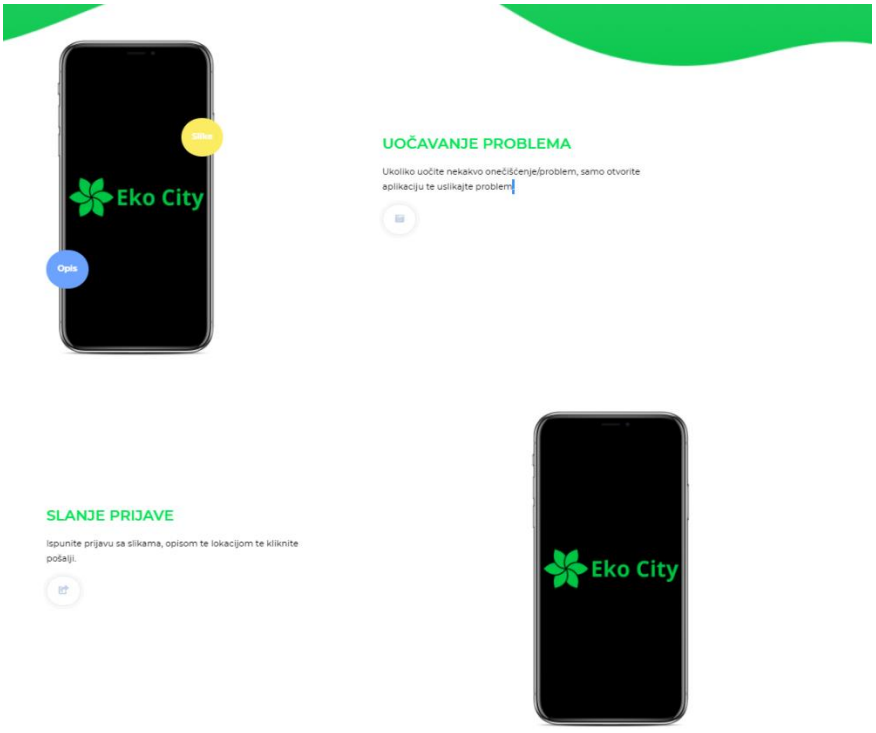
6.2. Web stranica za reklamiranje sustava

Ovakav sustav funkcionira samo ukoliko ima dovoljno korisnika koji će ga koristiti, te zbog toga razloga treba uložiti u reklamiranje aplikacije. Kao podloga za neke plaćene reklame izrađena je prototip web stranica koja bi služila kao “Landing page” nakon klika na reklami na kojoj je pojašnjeno što je naša aplikacija, kako funkcionira te gdje se je može preuzeti.



49. Prototip stranica - početna

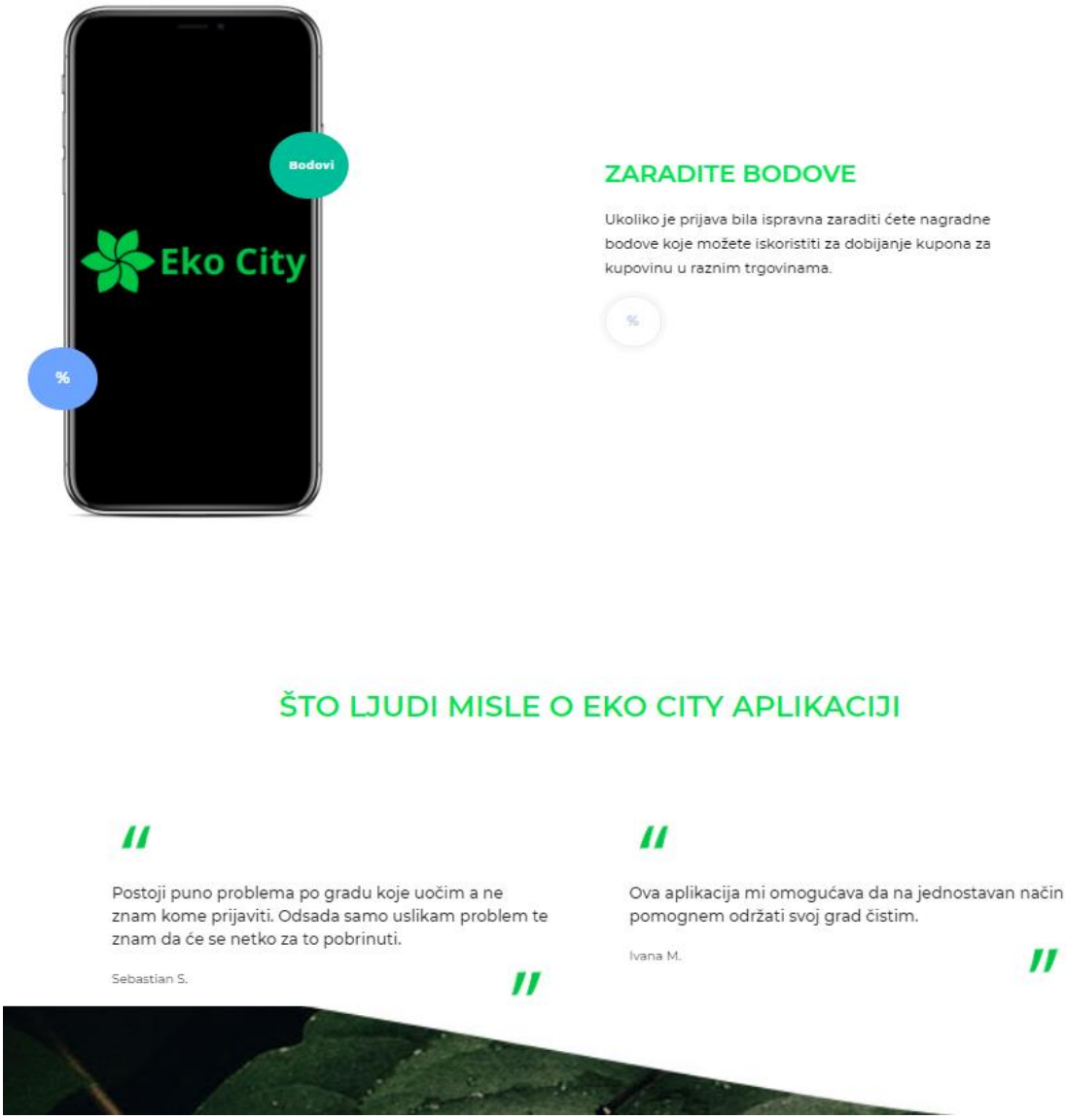
Izvor: izradio autor



50. Prototip stranica - značajke

Izvor: izradio autor

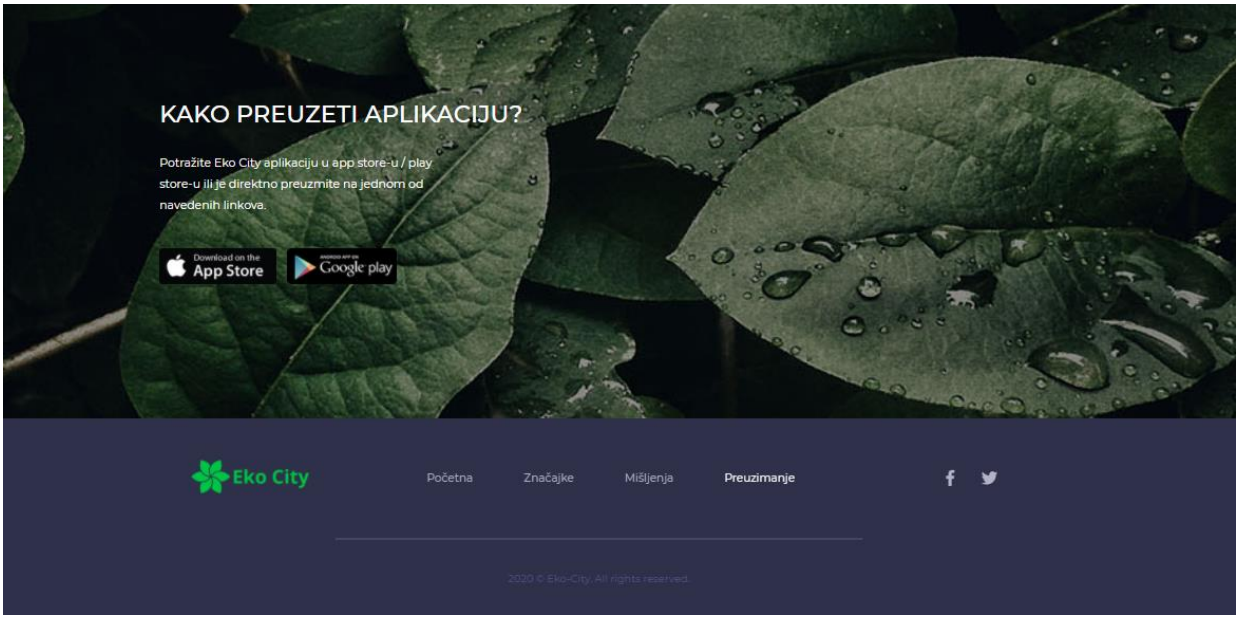
Na gornjoj slici na stranici su navedene neke značajke aplikacije kao što su uočavanje problema te slanje prijave te upute kako koristiti iste unutar same aplikacije.



51. Prototip stranica - mišljenja

Izvor: izradio autor

Na gornjoj slici možemo vidjeti opis nagradnog sustava mišljenja korisnika koji su koristili aplikaciju kao dokaz da je aplikacija vrijedna korištenja.



52. Prototip stranica - preuzimanje

Izvor: izradio autor

Na gornjoj slici možemo vidjeti upute za korisnike kako i gdje mogu preuzeti našu aplikaciju te “footer” koji vodi na ostale dijelove stranice.

6.3. Dodatno optimiziranje sustava

Također ukoliko bi sustav išao na tržište, dodatno bi se optimizirala veličina slika koja se sprema na server na način da bi se smanjila kvaliteta slika koja se sprema, optimizirao bi se dizajn samih aplikacija da bude više korisnički orijentiran te bi se dodale brojne sitne funkcionalnosti kao što su mogućnost pregleda starih prijava unutar same mobilne aplikacije, dodatak korisničke pomoći, itd.

7. Zaključak

Aplikacija za natjecanja izrađena je na moderan i suvremen način. Prednost takve aplikacije leži u tome da se ona može koristiti na modernim mobilnim uređajima ili web preglednika te je dostupna bilo gdje, na bilo kojem mjestu i u bilo kojem vremenu. Da bi se koristila na starijim uređajima potrebno ju je dodatno nadograditi i proširiti. Aplikacija se može također dodatno nadograditi i proširiti. Jedna od glavnih nadogradnji aplikacije bila bi da se ocjenjivači također mogu prijaviti u sustav, vidjeti natjecanja na kojima su sudjelovali te unaprijediti njihov korisnički račun (eng. account) na razinu administratora natjecanja. Isto tako, svaki od korisnika mogao bi imati detaljnije izvještaje o svim natjecanjima. Primjerice, uvid o učestalosti korisnika koje je administrator pozvao na natjecanja ili prikaz ukupnog broja bodova pojedinog natjecatelja za sva njegova natjecanja na kojima je sudjelovao.

Kroz izradu aplikacije, koristeći moderne tehnologije i globalno popularne alate za razvojno okruženje i pakete vidi se da je jednostavno napraviti vjerodostojnu, sigurnu i brzu aplikaciju koju se može pokrenuti na svim platformama, uređajima i web preglednicima. Brzim napretkom tehnologije napreduju i novi moderni preglednici, novi mobilni uređaji i računala također brzo razvili. Jezici i alati za razvojna okruženja i ostali paketi prate korak u tom razvoju. Za nove stvari da budu što sigurnije i brže, ljudi koji razvijaju alate za programere (developere) također moraju biti u toku s vremenom.

JavaScript jezik, jedan od najpopularnijih jezika smatra se najraširenijim, najbržim i najjednostavnijim jezikom za izradu progresivnih web aplikacija. Postoji i velik izbor alata za jednostavniji razvoj koristeći upravo taj jezik pa tako, u toj tehnologiji dominiraju razvojna okruženja kao što su: Angular (korišten je u ovom projektu), kojeg su razvili programeri iz tvrtke Google, React razvijen od strane Facebook programera i Vue alat kojeg je napravio Evan You, bivši zaposlenik Google-a. Teško se odlučiti za odabir određenog alata jer svaki od njih ima svoje prednosti i mane pa programer sam odlučuje o izboru s namjerom da najbolje iskoristi alat s kojim r

Literatura

- ANDERSON, N. (2016), *Getting started with NativeScript*, United States, Packt Publishing Ltd

- Moroney, L. (2017), *The Definitive Guide to Firebase: Build Android Apps on Google's Mobile platform,* Seattle, Washington,USA, Apress

- HAVERBEKE, M. (2015) *Eloquent JavaScript a Modern introduction to programming 2nd edition*,San Francisco, California, No Starch Press

Web

- https://nativescript.org/ [Pristupljeno: 16.8.2020.]
- https://www.php.net/ Pristupljeno: 10.8.2020.]
- https://getbootstrap.com [Pristupljeno: 10.8.2020.]
- https://sweetalert2.github.io/ [Pristupljeno: 12.8.2020.]
- https://developer.mozilla.org/en-US/docs/Web/JavaScript [Pristupljeno: 13.8.2020.]
- https://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/The-best-Java-frameworks-for-web-application-development-of-2020 [Pristupljeno: 13.8.2020.]
- https://www.xicom.biz/blog/9-best-reasons-to-choose-java-for-web-development/ [Pristupljeno: 14.8.2020.]
- https://code.tutsplus.com/articles/an-introduction-to-nativescript--cms-26771 [Pristupljeno: 15.8.2020.]
- https://firebase.google.com/ Pristupljeno: 16.8.2020.]
- https://getcomposer.org/ [Pristupljeno: 17.8.2020.]
- https://packagist.org/packages/delight-im/auth [Pristupljeno: 17.8.2020.]
- https://www.codecademy.com/articles/html5-features [Pristupljeno: 18.8.2020.]
- https://developer.mozilla.org/en-US/docs/Web/CSS/ Pristupljeno: 18.8.2020.]
- https://www.mapbox.com/ [Pristupljeno: 19.8.2020.]

Popis slika

1.Dijagram korištenja sustava (eng. use-case) 3
2. Dijagram korištenja podataka 4
3. Klasni dijagram aplikacije 5
4. Netbeans Java IDE 8
5. Višedretvenost u Javi 8
6. Primjer XML koda za gumb 14
7. Primjer Javascript koda za gumb 14
8. Prikaz ListView Komponente u Nativescriptu 14
9. Prikaz povezivanja polja s kontekstom stranice kroz javascript 15
10. Usporedba Nativescripta s ostalim hibridnim okvirima 17
11. Prijavni ekran mobilne aplikacije - registracija 20
12. Prijavni ekran mobilne aplikacije - prijava 21
13. Početni ekran mobilne aplikacije 22
14. prozor za dodavanje slike unutar prijave 23
15. Primjer dodavanja slike unutar prijave 23
16. Primjer dodavanja opisa unutar prijave 24
17. Prozor za dodavanje lokacije unutar prijave 25
18. Prozor za dodavanje lokacije putem mape 26
19. Gumb za reset prijave 26
20. Primjer uspješne prijave 27
21. Gumb za ulazak u izbornik 28
22. Izbornik unutar aplikacije 29
23. Ekran postavki 30
24. Inicijalizacija firebase paketa za ovjeravanje autentičnosti 31
25. Primjer skočnog prozora unutar aplikacije 32
26. Inicijalizacija cfalert-dialog paketa te primjer korištenja 33
27. Inicijalizacija geolocation paketa te primjer korištenja 34
28. Mapbox primjer 35
29. Inicijalizacija mapbox paketa te primjer korištenja 36
30. Prijavni dio web aplikacije 38

31. Moderatorski dio web aplikacije 39
32. Primjer neobrađene prijave 39
33. Primjer opisa prijave 40
34. Primjer slika prijave 40
35. Primjer lokacije prijave 41
36. Primjer obrađene prijave 41
37. Primjer otkazane prijave 42
38. Administrativni dio web aplikacije 42
39. Primjer dodjeljivanja gradova 43
40. Dodavanje gradova 43
41. Inicijalizacija paketa za autentifikaciju 44
42. Primjer korištenja paketa kod prijave u sustav 45
43. Primjer korištenja “nav” taga 46
44. Primjer korištenja CSS-a 46
45. Inicijalizacija Bootstrap-a 47
46. Bootstrap modal 47
47. Inicijalizacija SweetAlert2 paketa 48
48. Primjer SweetAlert2 skočnog prozora 48
49. Prototip stranica - početna 50
50. Prototip stranica - značajke 50
51. Prototip stranica - mišljenja 51
52. Prototip stranica - preuzimanje 52

Sažetak

Sustav za geolociranje, fotografiranje i prijavu problema u okolišu služi za rješavanje problema zagađenja u pojedinim gradovima pomoću dobre volje građana koji žele pomoći gradu. Sustav je podijeljen na dva dijela, mobilnu aplikaciju za korisnike (građane) te web aplikaciju za administratore i moderatore koji će analizirati prijave građana te prosljeđivati legitimne prijave nadležnim službama u tom gradu koje su zadužene za rješavanje zagađenja.

Mobilna aplikacija je izrađena pomoću web tehnologija u okruženju Nativescript. Ovakva okruženja su sve češća u izradi mobilnih aplikacija pošto se može pisati jedan kod a aplikacija se zatim može izdati na više platformi. Mobilna aplikacija je također izrađena pomoću nekoliko dodataka kao što su firebase dodatak za autentikaciju te mapbox dodatak za prikaz lokacija zagađenja na karti.

Web aplikacija izrađena je pomoću klasičnih metoda izrade kao što su PHP i MySQL na strani poslužitelja te HTML5, Javascript i razni dodaci kao što je Bootstrap na strani klijenta. Web aplikacija služi za prijem korisničkih prijava koje se zatim obrađuju od strane moderatora zaslužnih za pojedine gradove te se problemi dalje prosljeđuju nadležnim službama.

Summary

The system for geolocation, photography and reporting of environmental problems serve to solve the problem of pollution in individual cities with good will of citizens who want to help the city. The system is divided into two parts, a mobile application for users (citizens) and a web application for administrators and moderators who will analyze citizens' reports and forward legitimate reports to the competent services in that city in charge of solving pollution.

The mobile application was created using web technologies in the Nativescript environment. Such environments are becoming more common in development of mobile applications as one code base can be written for distribution on multiple platforms. The mobile app was also created using several packages such as firebase library for authentication and a mapbox library for displaying pollution locations on a map.

The web application is built using classic build methods such as PHP and MySQL on the server side and HTML5, Javascript and various libraries such as Bootstrap on the client side. The web application is used to receive user reports, which are then processed by moderators responsible for individual cities, and the problems are forwarded to the firms that deal with pollution in that jurisdiction.