Razvoj mobilne aplikacije za upravljanje javnim gradskim prijevozom

Šamija, Valentin

Undergraduate thesis / Završni rad

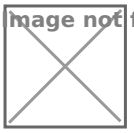
2019

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:137:838338

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

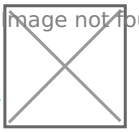
Download date / Datum preuzimanja: **2024-03-28**



Repository / Repozitorij:

Digital Repository Juraj Dobrila University of Pula





Sveučilište Jurja Dobrile u Puli
Fakultet informatike u Puli

VALENTIN ŠAMIJA

IZRADA MOBILNE APLIKACIJE ZA JAVNI GRADSKI PRIJEVOZ

Završni rad

Pula, rujan, 2019.

Sveučilište Jurja Dobrile u Puli
Fakultet informatike u Puli

VALENTIN ŠAMIJA

IZRADA MOBILNE APLIKACIJE ZA JAVNI GRADSKI PRIJEVOZ

Završni rad

JMBAG: 0303068763, redoviti student
Studijski smjer: Informatika

Predmet: Programsko inženjerstvo
Znanstveno područje: Društvene znanosti
Znanstveno polje: Informacijske i komunikacijske znanosti
Znanstvena grana: Informacijski sustavi i informatologija
Mentor: doc. dr. sc. Tihomir Orehovački

Pula, rujan, 2019.

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani _____, kandidat za prvostupnika _____ovime izjavljujem da je ovaj Završni 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 Završnog 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 završni 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

_____

SAŽETAK

Aplikacija koja je opisana u ovom radu je mobilna aplikacija. Što se tiče izrade i biranja platforme ili sustava na kojoj/em će raditi precizno je izabrano kako bi osvojila ciljane korisnike. Korisnici koji će koristiti ovu aplikaciju su upravo ljudi koji svoje pametne telefone ne ispuštaju iz ruku i žele da im je sve „blizu“, odnosno na jednome mjestu – pametnom telefonu. Aplikacija se bazira na javni gradski prijevoz i omogućava korisniku da pronađe točno onu liniju (rutu) koja ga vodi do njegove ciljne točke. Osim što nudi popis svih linija kojima autobusi prometuju, korisnik može provjeriti i cijene za pojedinu liniju kako ne bi došlo do konfuzija prilikom polaska, odnosno dolaska na odredište.

KLJUČNE RIJEČI: *AndroidStudio, Android Emulator, React-Native, NodeJS, Python2, JDK, Chocolatey, Visual Studio Code, Android okruženje, Android SDK, Kreiranje nove aplikacije, Mobilna aplikacija, Javni gradski prijevoz, Linije, Informacije, Cjenik*

ABSTRACT

The application described in this documentation is a mobile application. In terms of designing and selecting the platform or system on which it will work, it has been carefully selected to get to targeted users. The users who will use this application are they who want everything to be on their Smartphones. The application is based on public transport and allows the user to find exactly the line (route) that leads him to his destination. In addition to providing a list of all bus routes, the user can also check the prices for each line to avoid any confusion on departure or arrival at the destination.

KEYWORDS: *AndroidStudio, Android Emulator, React-Native, NodeJS, Python2, JDK, Chocolatey, Visual Studio Code, Android development environment, Android SDK, Creating new application, Mobile app, Public Transport, Schedule, Info, Price list*

SADRŽAJ

1.UVOD 7
2. ALATI KORIŠTENI ZA IZRADU APLIKACIJE 9
3. RAZRADA FUNKCIONALNOSTI.....14
4. KORISNIČKE UPUTE.....21
5. ZAKLJUČAK.....26
6. LITERATURA 27
7. POPIS SLIKA 28

1.UVOD

Svijet u kojem se nalazi današnje društvo je dostiglo vrijeme u kojem se čovjek ne može zamisliti bez mobilnoga uređaja. U prošlosti mobiteli nisu predstavljali neki oblik „zabave“ i načina pomoću kojega bi se moglo skratiti vrijeme ili trenutak u kojem bi se obavio neki posao. Napredovanjem tehnologija mobiteli su preuzeli gotovo svaki ljudski pokret, toliko snažno da prepoznaju i ljudske osjećaje.

Nezamislivo je prošetati ulicom i susresti nekoga bez mobitela u „ruci“ ili na „uhu“. Mobiteli, odnosno današnji pametni telefoni skraćuju vrijeme, odrađuju bilo koji posao ili zadaću koju treba obaviti, ali naravno i donose štetu za društvo. Cijeli život je „pretoćen“ u mobilni uređaj; obitelj, rodbina, prijatelji, razgovor, pismo, osjećaji, ljubav, posao, novac, društvene mreže. Kako bi čovjek uspio izvršiti dnevne zadatke potreban mu je mobilni uređaj jer je upravo „tu“ sve zabilježeno.

Dan započinje buđenjem (alarmom), zdravlje provjeravamo preko „gadgeta“, smjena na poslu je popraćena kreiranom (Viber, Whatsapp) grupom u kojoj kolege razmjenjuju mišljenja, obrok je „naručen“ ili „dostavljen“ pomoću aplikacija, cijeli dan obilježen je beskonačnim brojem vibracija i zvukova koje proizvede jedan „Smartphone“.

Ideja za izradu mobilne aplikacije za javni gradski prijevoz upravo je „rezultat“ svih gore navedenih razloga. Bilo kakav korak koji treba učiniti, iziskuje neku veličinu informacija koje će omogućiti izvršavanje istoga. Korisniku trebaju brze, točne i precizne informacije. Ako se pronađe u situaciji u kojoj gubi vrijeme, ne pronalazi točno ono što mu treba vrlo lako postaje nervozan, a u najgorem slučaju i agresivan. Situacije zbog kojih je nastala ova aplikacija su sljedeće: neinformiranost korisnika, netočne informacije, gubitak vremena „traženjem“ iskrenih odgovora od slučajnih prolaznika, u turističkim gradovima gost ima volju i želju posjetiti što više destinacija, ali uz prethodne informacije koje će mu to i omogućiti. Mobilna aplikacija koja će biti opisana u ovome radu upravo ispunjava i zadovoljava sve kriterije koje korisnik u današnje vrijeme traži. Jednostavnim preuzimanjem aplikacije korisnik ima uvid u sve ono što ga zanima, a to je: popis svih linija (ruta), stanica kojima autobusi

prometuju, zone kroz koje prometuju te naravno cijena je ono najbitnije što svatko želi znati.

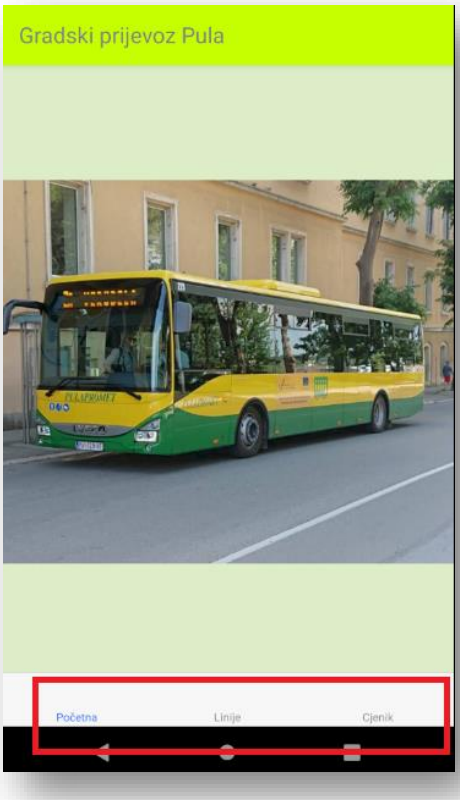
Osim pregleda svih informacija koje su potrebne, implementirane su i neke funkcije koje aplikacija izvršava. Najbitnije su one funkcije koje omogućavaju dodavanje i spremanje linija, ruta, informacija, stajališta. Svaka će biti opisana u nastavku.

2. ALATI KORIŠTENI ZA IZRADU APLIKACIJE

Na prvi pogled aplikacije izgleda vrlo jednostavno što je zapravo i cilj kako bi se korisnik što jednostavnije koristio istom, ali izrada aplikacije iziskuje dosta vremena, mnogo truda te nekoliko alata i paketa koji će biti opisani.

Da bi programer mogao započeti izradu aplikacije potrebni su mu alati te paketi za izradu same aplikacije, a to su:

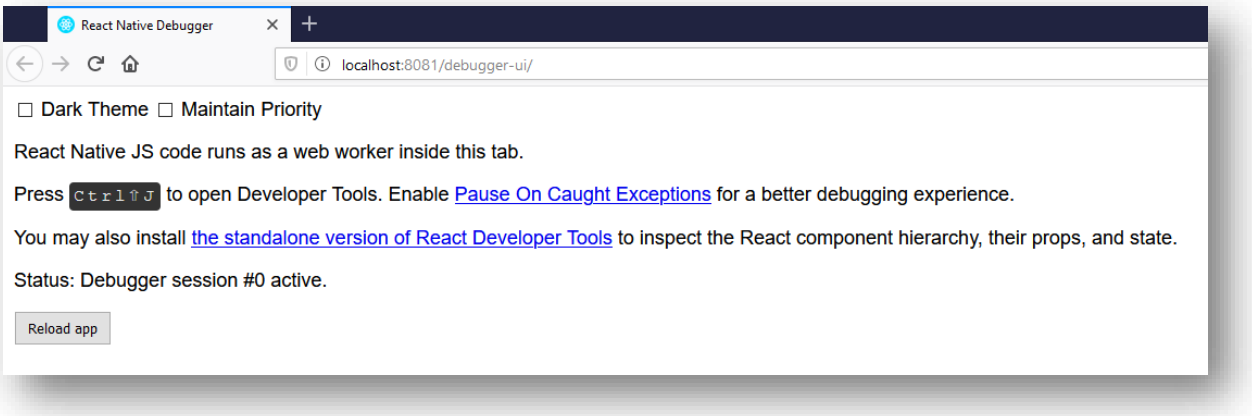
- React-Native - „Open-Source Framework“ za mobilne aplikacije kreiran od strane Facebooka. Koristi se za razvoj aplikacija za Android, iOS, Web i UWP. (https://facebook.github.io/react-native/docs/getting-started)



Slika 1. React-Native (React navigation)

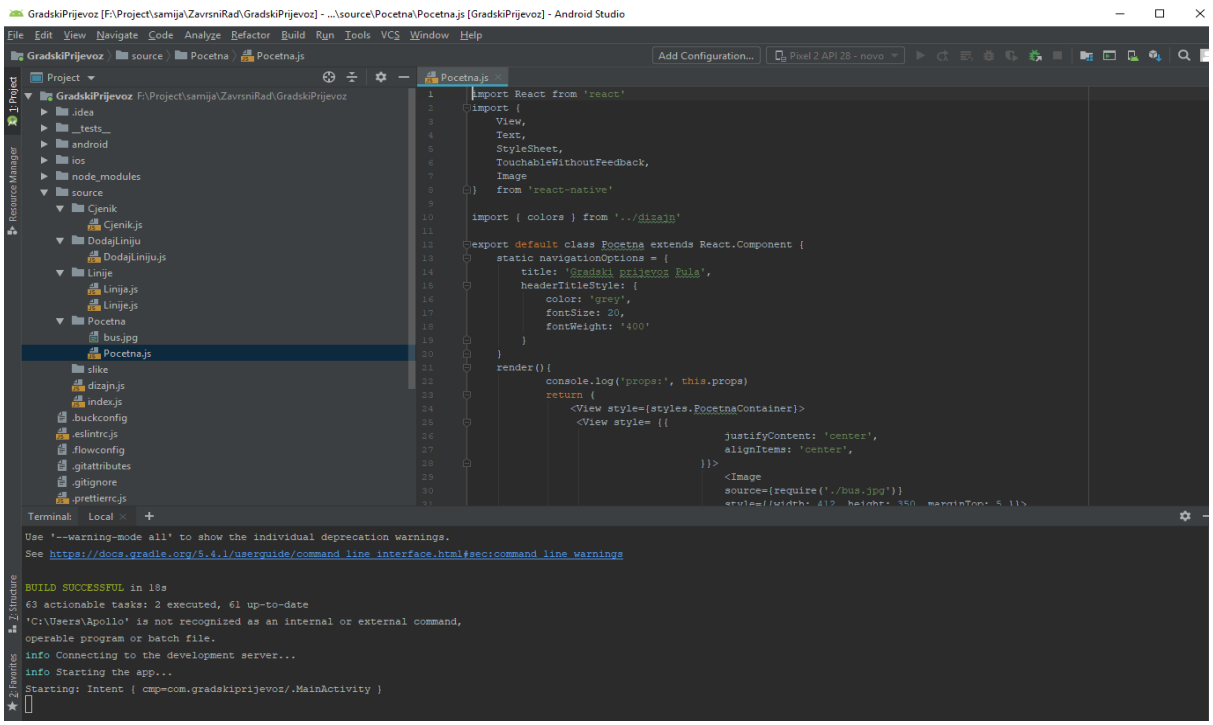
Na slici iznad, dio koji je istaknut crvenom bojom prikazuje navigacijsku traku. Navigacija unutar aplikacije omogućena je „uvozom“ navigacijske trake iz paketa pod nazivom „react-navigation. (https://snack.expo.io/)

- NodeJS – događaj pokrenut za vrijeme izvršavanja JavaScriptw, web preglednik (https://nodejs.org/en/docs/)



Slika 2. NodeJS

- Android Studio - službeno je integrirano razvojno okruženje za Googleov operativni sustav Android i dizajnirano posebno za razvoj Androida. Dostupan je za preuzimanje na Windows, macOS i Linux operativnim sustavima. (https://developer.android.com/studio/run/managing-avds.html)



Slika 3. AndroidStudio

Preko „Command Prompt“ potrebno je instalirati sljedeće pakete:

- Chocolatey - NodeJS, Python2 (https://chocolatey.org/install)

```
choco install -y nodejs.install python2 jdk8
```

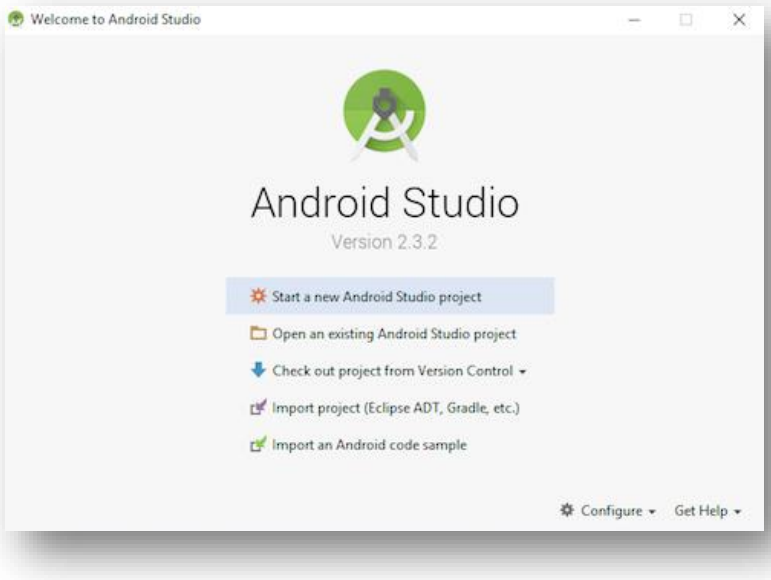
Slika 4. Chocolatey

- React-Native CLI:

```
npm install -g react-native-cli
```

Slika 5. CLI

- Android okruženje:

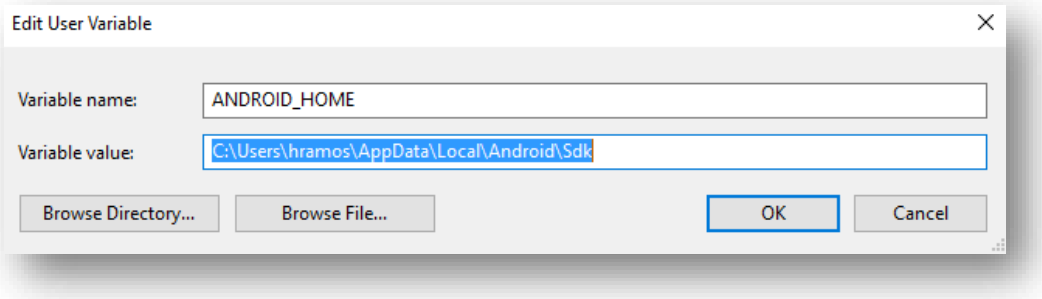


Slika 6. Android development environment

- Configure → SDK Manager → SDK Platforms → Show details (Android 9) Pie → Android SDK Platform 28 → Google APIs Intel x86 System Image
- Configure → SDK Manager → SDK Tools → Show details → Android SDK Build-Tools → 28.0.3 (https://developer.android.com/studio/index.html)

- Varijable okruženja:

  - System and Security → Change Settings → Advanced → Environment Variables → New → ANDROID_HOME



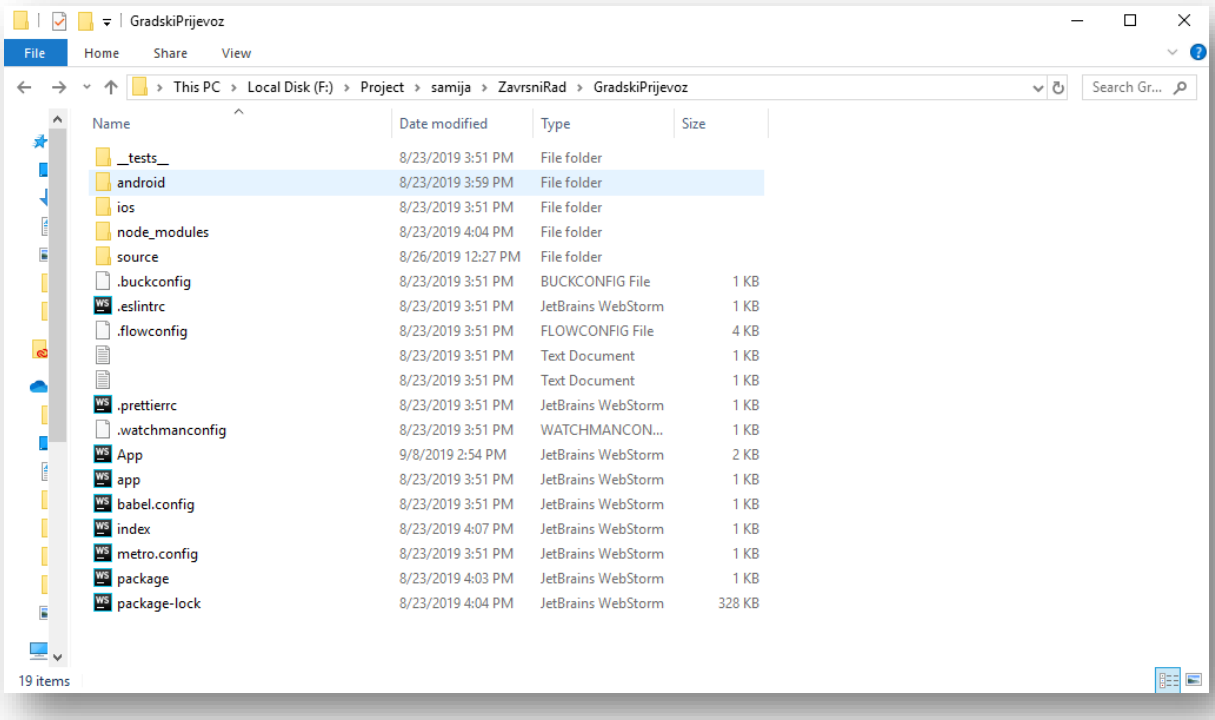
Slika 7. Environment Variables

- Kreiranje novog projekta (aplikacije):

```
react-native init AwesomeProject
```

Slika 8. New app

Ova naredba kreira novi projekt u željenoj datoteci ili na željenoj lokaciji te dodaje sve potrebne elemente za izradu aplikacije.



Slika 9. Project files

- Pokretanje projekta:

```
cd AwesomeProject
react-native run-android
```

Slika 10. Project start

Ako nije postavljen „pravi put“ pokretanje će rezultirati greškom.

3. RAZRADA FUNKCIONALNOSTI

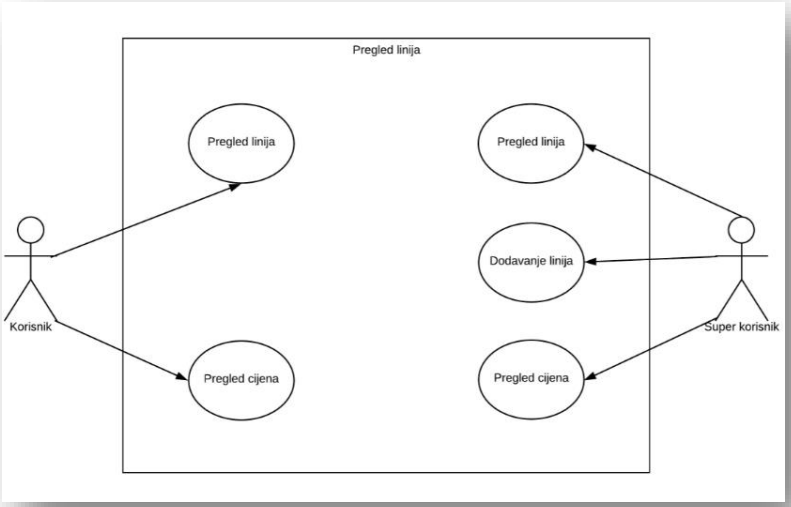
Ovo poglavlje opisuje funkcionalnosti aplikacije. Na početku će biti prikazano kako funkcionira aplikacija na temelju Use-Case dijagrama te kakva je interakcija između korisnika i aplikacije. Potom je prikazan dijagram slijeda na kojem se jasno vide funkcije ove aplikacije. U konačnici je prikazan klasni dijagram te su opisane veze između klasa.

U ovoj aplikaciji definirana je zapravo samo jedna skupina korisnika, tj. korisnik, ali postoji i "super korisnik" koji može upravljati linijama. Jednostavniji način u kojem "super korisnik" nije potreban je dodavanje liste koja će prikazivati popis svih dostupnih linija. "Super korisnik" je implementiran zbog toga da se prikaže funkcija dodavanje linija te svih informacija vezanih uz pojedinu liniju.

Funkcionalnosti unutar aplikacije:

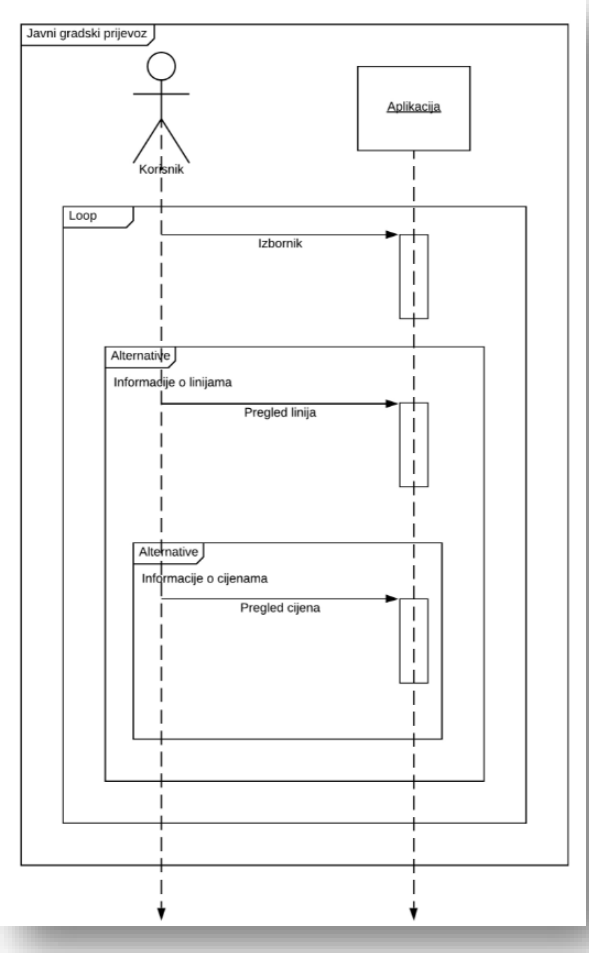
- Pregled linija
- Pregled informacija o linijama
- Pregled cijena
- Dodavanje linija
- Dodavanje stanica
- Dodavanje informacija

Prikaz Use-Case dijagrama. Kao što se može iz slike zaključiti, postoje dva sudionika, odnosno aktera – korisnik i "super korisnik. Korisnik ima dvije mogućnosti, a to su pregled linija te pregled cijena. "Super korisnik" može pregledavati linije te dodavati nove.



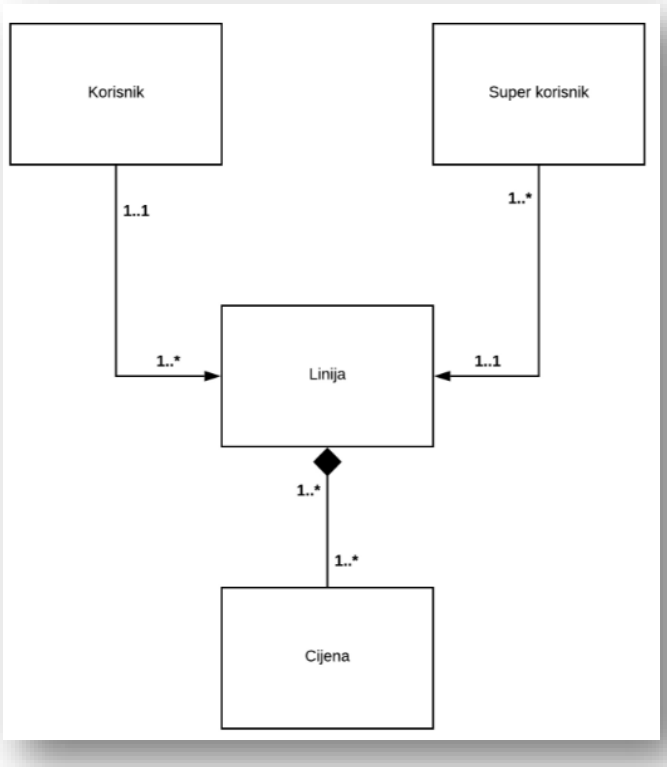
Slika 11. Use-Case dijagram

Korisnik bira što želi učiniti i to je prikazano “Sequence” (slijednim) dijagramom.



Slika 12. Dijagram slijeda

Prilikom otvaranja aplikacije korisnik ima “slobodu” birati što želi pregledavati. Ako izabere pregled linija prikazat će mu se popis dostupnih linija te može provjeriti sve informacije o liniji koju je izabrao. Nakon što je provjerio informacije o liniji i odlučio gdje želi ići može provjeriti i cijenu. Cijene se razlikuju jer postoje zone “kroz” koje autobusi prometuju.

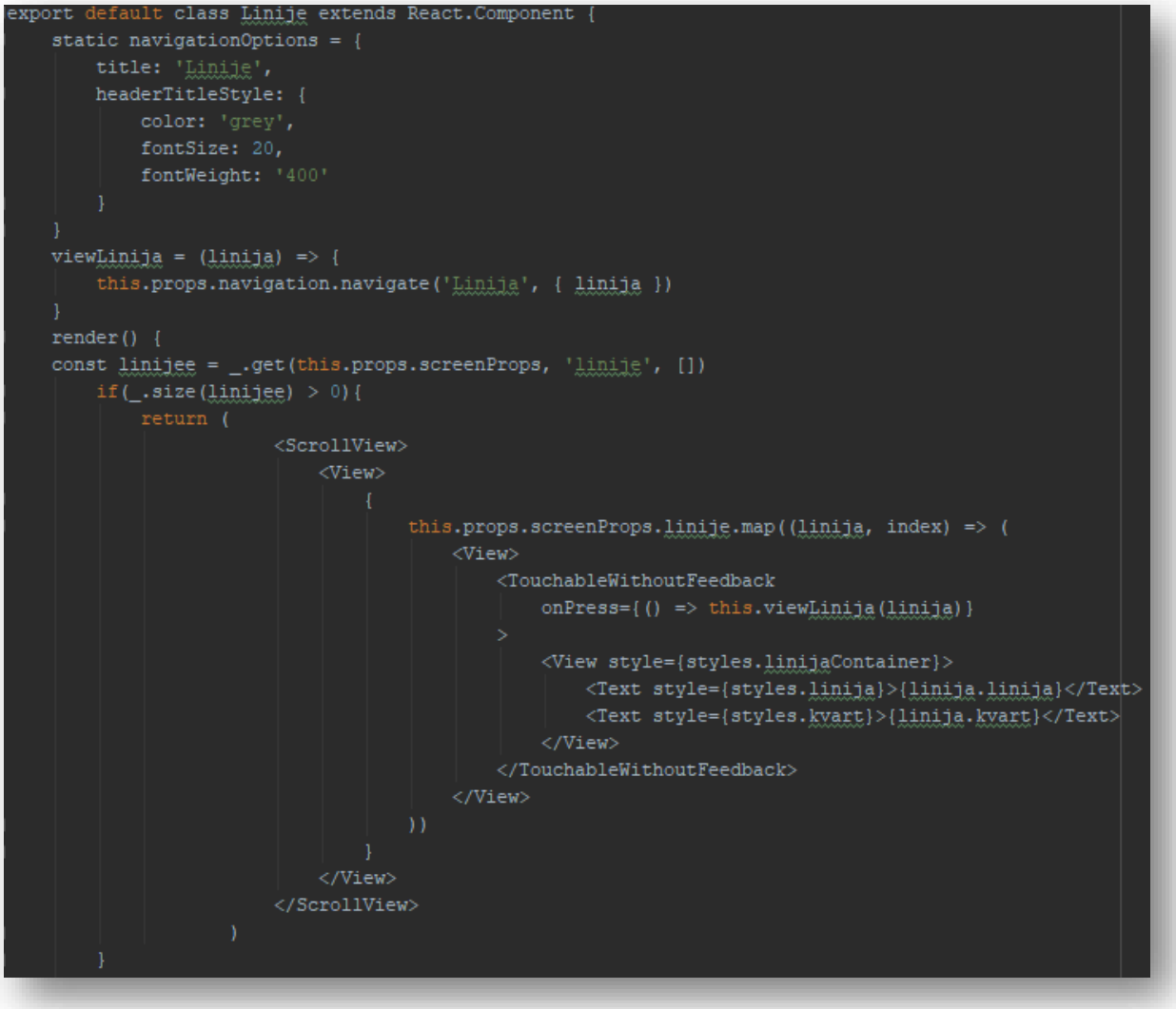


Slika 13. Klasni dijagram

Na slici je prikazan klasni dijagram. Klasni dijagram opisuje strukturu sustava, odnosno objašnjava klase unutar sustav, njihove atribute te njihove odnose. Osim toga postoje kardinalnosti koje objašnjavaju odnose među klasama. Asocijacija prikazuje vezu između klasa. Postoji i kompozicija. Ona prikazuje objekt koji nema smisla bez roditelja. Ako se izbriše linija, cijena se briše zajedno s linijom. U tom slučaju korisnik ne može provjeriti linije i cijene jer one ne postoje.

Slijedi prikaz i opis temeljnih funkcionalnosti:

- Pregled linija (popis linija)



```
export default class Linije extends React.Component {
    static navigationOptions = {
        title: 'Linije',
        headerTitleStyle: {
            color: 'grey',
            fontSize: 20,
            fontWeight: '400'
        }
    }
    viewLinija = (linija) => {
        this.props.navigation.navigate('Linija', { linija })
    }
    render() {
    const linijee = _.get(this.props.screenProps, 'linije', [])
        if(_.size(linijee) > 0){
            return (
                        <ScrollView>
                            <View>
                                {
                                    this.props.screenProps.linije.map((linija, index) => (
                                        <View>
                                            <TouchableWithoutFeedback
                                                onPress={() => this.viewLinija(linija)}
                                            >
                                                <View style={styles.linijaContainer}>
                                                    <Text style={styles.linija}>{linija.linija}</Text>
                                                    <Text style={styles.kvart}>{linija.kvart}</Text>
                                                </View>
                                            </TouchableWithoutFeedback>
                                        </View>
                                    ))
                                }
                            </View>
                        </ScrollView>
                    )
        }
```

Slika 14. Pregled linija

Dio koda koji je prikazan na slici omogućava pregled svih dostupnih linija te prikaz svih informacija za linije. Prvo je definiran izgled zaslona koji će prikazivati linije. Program najprije provjerava postoje li dostupne linije. U slučaju da postoje na zaslonu će se ispisati sve linije, a ako ne postoji niti jedna program će izbaciti poruku koja obavještava korisnika da ne postoji niti jedna linija.

- Prikaz informacija za odabranu liniju

```
render() {
    const { linija } = this.props.navigation.state.params
    return (

        <View style={{ flex: 1}}>
            {
                linija.stanice.map((stanica, index) => (
                    <View style={styles.stanicaContainer}>
                        <Text style={styles.naziv}>{stanica.naziv}</Text>
                        <Text style={styles.info}>{stanica.info}</Text>
                    </View>
                ))
            }
```

Slika 15. Prikaz informacija

Slika opisuje prikaz informacija o liniji koju je korisnik izabrao. Klikom na naziv linije, program „dohvaća“ informacije koje su spremljene za odabranu liniju te ih ispisuje na ekran. Iako postoji veliki broj linija kojima autobusi prometuju, program točno zna koje informacije treba prikazati za odabranu liniju pomoću „lokacije“ koja mu govori gdje su informacije spremljene.

- Pregled cijena

```
render () {
    console.log('props:', this.props)
            return (
                <ScrollView>
                    <View style={styles.container}>
                        <Text style={styles.heading}>I. zona: jedan smjer - 11kn</Text>
                        <Text style={styles.heading}>I. zona: opća - 230kn</Text>
                        <Text style={styles.heading}>I. zona: umirovljenička - 80kn</Text>
                        <Text style={styles.heading}>I. zona: studentska - 140kn</Text>

                        <Text style={styles.heading}>_____</Text>

                        <Text style={styles.heading}>II. zona: jedan smjer - 15kn</Text>
                        <Text style={styles.heading}>II. zona: opća - 400kn</Text>
                        <Text style={styles.heading}>II. zona: umirovljenička - 160kn</Text>
                        <Text style={styles.heading}>II. zona: studentska - 220kn</Text>

                        <Text style={styles.heading}>_____</Text>

                        <Text style={styles.heading}>III. zona: jedan smjer - 20kn</Text>
                        <Text style={styles.heading}>III. zona: opća - 550kn</Text>
                        <Text style={styles.heading}>III. zona: umirovljenička - 240kn</Text>
                        <Text style={styles.heading}>III. zona: studentska - 550kn</Text>

                    </View>
                </ScrollView>

            )
}
```

Slika 16. Prikaz cijena

Cijene su prikazane na najjednostavniji način uz pomoć „View“ objekta. Definirane su cijene koje se prikazuju u trenutku kada korisnik klikne na „Cjenik“. Na slici se jasno vidi da postoji i „ScrollView“ koji omogućava pomicanje gore-dolje.

- Dodavanje linija



```
export default class DodajLiniju extends React.Component {
    state = {
        linija: '',
        kvart: ''
    }
    onChangeText = (key, value) => {
        this.setState({
            [key]: value
        })
    }
    submit = () => {
        if (this.state.linija === '' || this.state.kvart === '') return
        const linija = {
            linija: this.state.linija,
            kvart: this.state.kvart,
            stanice: [],
            id: uuidV4()
        }
        this.props.screenProps.dodajLiniju(linija)
        this.setState({
            linija: '',
            kvart: '',
        }, () => {
            this.props.navigation.navigate('Linije')
        })
    }
```

```
render(){
    console.log('props:', this.props)
    return (
        <View style={styles.container}>
            <Text style={styles.heading}>Dodavanje linije</Text>
            <TextInput
                placeholder='Naziv linije'
                value={this.state.linija}
                onChangeText={val => this.onChangeText('linija', val)}
                style={styles.input}
            />
            <TextInput
                placeholder='Naziv kvarta'
                value={this.state.kvart}
                onChangeText={val => this.onChangeText('kvart', val)}
                style={styles.input}
            />
            <TouchableOpacity onPress={this.submit}>
                <View style={styles.button}>
                    <Text style={styles.buttonText}>Dodaj liniju</Text>
                </View>
            </TouchableOpacity>
        </View>
    )
}
}
```

Slika 17. Dodavanje linije

Funkcija omogućava “super korisniku” dodavanje linija te svih informacija. Kreirani su objekti u koje će se spremati naziv linije i dio grada kroz koji autobus vozi. “Submit” tipka najprije provjerava je li korisnik unio sve informacije, ako nije dodavanje linije neće biti uspješno izvršeno. Tek kada unese naziv linije te naziv kvarta, nova linija će biti uspješno dodana.

- Dodavanje stanice

```
dodajStanicu = () => {
    if (this.state.naziv === '' || this.state.info === '' ) return
    const { linija } = this.props.navigation.state.params
    const stanica = {
        naziv: this.state.naziv,
        info: this.state.info
    }
    this.props.screenProps.dodajStanicu(stanica, linija)
    this.setState({
        naziv: '',
        info: ''
    })
}
```

Slika 18. Dodavanje informacija

Funkcija prikazana na slici omogućava dodavanje informacija o stanicama kojima autobus prometuje. Kao u prethodnom primjeru, korisnik mora unijeti i naziv stanice i infromacije kao što je vrijeme polaska kako bi mogao spremiti informacije koje će biti prikazane za tu stanicu.

4. KORISNIČKE UPUTE

Korisničke upute su podijeljene u dvije kategorije.

Prva kategorija opisuje upute koje su vezane za korisnika.
Druga kategorija opisuje upute vezane za „super korisnika“.



Slika 19. Početni zaslon

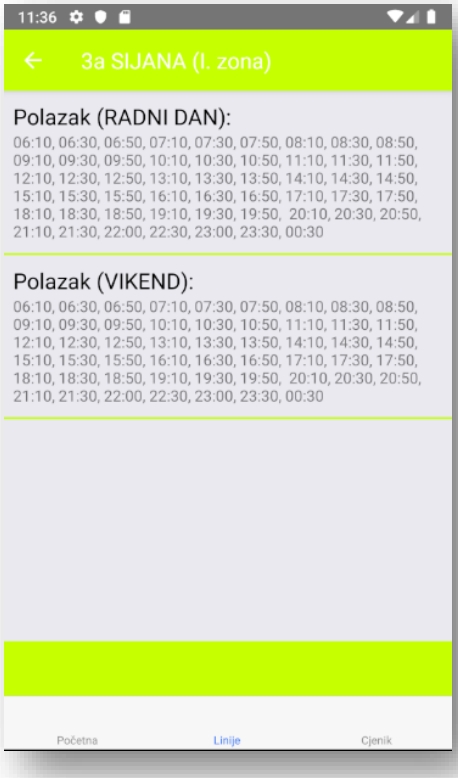
Pokretanjem aplikacije korisnik uočava početni zaslon kao što je prikazano na slici iznad. Navigacijska traka je pojednostavljena i prikazane su sve mogućnosti koje aplikacija nudi.

Nakon što korisnik odabere „Linije“ prikazat će mu se popis svih dostupnih linija:



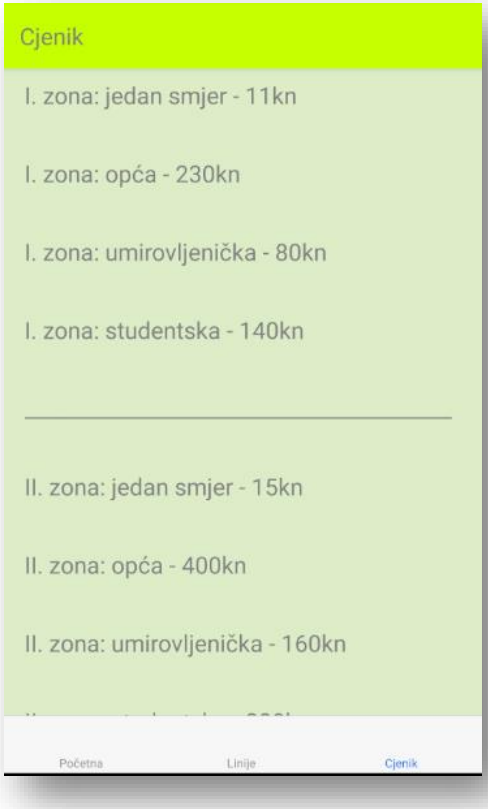
Slika 20. Popis dostupnih linija

Klikom na neku od linija, otvara se zaslon koji prikazuje sve informacije za izabranu liniju (http://pulapromet.com/hr/naslovna-2/):



Slika 21. Informacije

U trenutku kada korisnik odluči gdje želi ići, klikom na „Cjenik“ može provjeriti cijenu. Cijene su prikazane za sve zone:



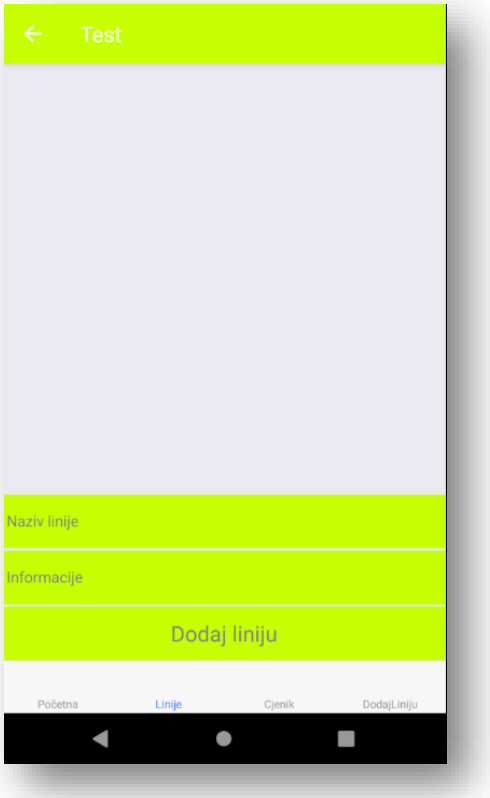
Slika 22. Cjenik

Sljedeće slike prikazuju upute za „super korisnika“. Osim pregleda linija i cjenika, „super korisnik“ ima mogućnost dodavanja linija te uređivanja svih informacija.



Slika 23. Dodavanje linije

Potrebno je upisati naziv linije i naziv kvarta kojim autobus prometuje te nakon toga kliknuti na „Dodaj liniju“. Linija će biti dodana i prikazana na popisu svih linija. Nakon dodavanje linije slijedi dodavanje informacija koje opisuju istu.



Slika 24. Informacije o liniji

5. ZAKLJUČAK

Vrijeme u kojem živimo ne možemo zamisliti bez pametnih mobitela i uređaja. Svaki naš korak i bilo koja radnja popraćena je trenutcima koje spremamo na „Smartphone“. Tehnologija je toliko napredovala i napretku nema kraja. Iz dana u dan se pojavljuju nove stvari za kojima ludimo. Korisnici imaju prevelike zahtjeve, a proizvođači nastoje zadovoljiti svaki kriterij koji im je postavljen.

Greške koje se pojavljuju su smanjene na minimum i teži se ka savršenstvu. Informacije koje dolaze do korisnika su toliko točne i precizne da su i sami proizvođači nekada iznenađeni onime što su postigli. Ljudi su dostigli „tempo“ u kojem jure za budućnošću i ostvarenju što većih ciljeva.

Aplikacija koja je opisana u ovome radu pokušava zadovoljiti svaki kriterij koji korisnik od iste traži. Upravo ona nudi točne informacije i olakšava korisniku prolazak kroz svakodnevicu. Svojim jednostavim sučeljem koje je korisniku i više nego poznato pruža sigurnost i povjerenje. Bilo to odlazak u školu, na posao ili razgledavanje grada, sve je na jednome mjestu. Svaka informacija koja može uštediti vrijeme i novac čini korisnika sretnim.

Pokretanjem aplikacije jasno se vidi što je korisniku ponuđeno i automatski privlači njegovu pozornost nakon čega slijedi provjera njemu potrebnih informacija i polazak do željene lokacija u gradu.

6. LITERATURA

[1]. https://facebook.github.io/react-native/docs/getting-started, 09.09.2019.

[2]. https://nodejs.org/en/docs/, 09.09.2019.

[3]. https://chocolatey.org/install, 09.09.2019.

[4].https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html, 09.09.2019.

[5]. https://developer.android.com/studio/index.html, 09.09.2019.

[6]. https://developer.android.com/studio/run/managing-avds.html, 09.09.2019.

[7]. https://snack.expo.io/, 09.09.2019.

[8]. http://pulapromet.com/hr/naslovna-2/, 09.09.2019.

7. POPIS SLIKA

Slika 1. React-Native (React navigation)……9
Slika 2. NodeJS..... 10
Slika 3. AndroidStudio..... 13
Slika 4. Chocolatey..... 19
Slika 5. CLI..... 26
Slika 6. Android development environment..... 11
Slika 7. Environment variables..... 12
Slika 8. New app…..... .12
Slika 9. Project files..... 13
Slika 10. Project start..... 13
Slika 11. Use-case dijagram..... 19
Slika 12. Dijagram slijeda..... 15
Slika 13. Klasni dijagram..... 16
Slika 14. Pregled linija..... 17
Slika 15. Prikaz informacija..... 18
Slika 16. Prikaz cijena..... 13
Slika 17. Dodavanje linije..... 19
Slika 18. Dodavanje informacija..... 26
Slika 19. Početni zaslon..... 26
Slika 20. Popis dostupnih linija..... 26
Slika 21. Informacije..... 26
Slika 22. Cjenik..... 26
Slika 23. Dodavanje linije..... 26
Slika 24. Informacije o liniji..... 26