iDeliver

Pavlović, Zvonimir

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:172413

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-07-27**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij računarstva

iDeliver

Diplomski rad

Zvonimir Pavlović

Osijek, 2021.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit

Osijek, 14.06.2021.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za diplomski ispit

Ime i prezime studenta:	Zvonimir Pavlović
Studij, smjer:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D-1077R, 06.10.2019.
OIB studenta:	65721715067
Mentor:	Izv. prof. dr. sc. Ivica Lukić
Sumentor:	
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	Izv. prof. dr. sc. Krešimir Nenadić
Član Povjerenstva 1:	Izv. prof. dr. sc. Ivica Lukić
Član Povjerenstva 2:	Izv. prof. dr. sc. Mirko Köhler
Naslov diplomskog rada:	iDeliver
Znanstvena grana rada:	**Informacijski sustavi (zn. polje računarstvo)**
Zadatak diplomskog rada:	Aplikacija omogućava komunikaciju između trgovine i kurira. Trgovina odabire kurire sa kojima želi surađivati, te stvara dostave. Kuriri koji su povezani sa određenom trgovinom, nakon stvorene dostave dobiju notifikaciju, te dostavu mogu prihvatiti ili jednostavno ignorirati. Nakon prihvaćanja dostave, imaju mogućnost otkazivanja ili dostavljanja paketa, s tim da se broj otkazivanja zapisuje u bazi, kako bi trgovine imale uvid u statistike o uspješnosti dostava svakog pojedinog kurira. Trgovina se povezuje sa kurirom na način da stvara jedinstveni kod koji se povezuje sa kurirom, te mu ga šalje pomoću SMS-a ili notifikacijom. Kurir zatim upisuje taj kod, te stvara vezu između sebe i trgovine.
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene mentora:	14.06.2021.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 19.07.2021.

Ime i prezime studenta:	Zvonimir Pavlović
Studij:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D-1077R, 06.10.2019.
Turnitin podudaranje [%]:	5%

Ovom izjavom izjavljujem da je rad pod nazivom**: iDeliver**

izrađen pod vodstvom mentora Izv. prof. dr. sc. Ivica Lukić

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD *1*

1.1. Zadatak diplomskog rada **1**

2. Prikaz postojećih rješenja za komunikaciju između dostavljača i radnji *1*

2.1. Glovo **2**

2.2. Pauza **2**

2.3. Wolt **3**

2.4. Idejno rješenje vlastite aplikacije iDeliver **4**

3. MODEL APLIKACIJE *5*

3.1. Korisnički zahtjevi na aplikaciju iDeliver **5**

3.2. Registracija **5**

3.2.1. Registracija radnje 5

3.2.2. Registracija dostavljača 5

3.3. Prijava **6**

3.4. Pregled dostava **6**

3.5. Stvaranje dostave **6**

3.6. Pregled povezanih radnji i dostavljača **6**

3.7. Pozivanje dostavljača na suradnju **7**

3.8. Dodavanje radnje **7**

3.9. Interakcija sa pojedinom dostavom **7**

3.10. Dodatne postavke **8**

4. OPIS KORIŠTENIH ALATA I TEHNOLOGIJA *10*

4.1. Xcode **10**

4.2. Programski jezik Swift **11**

4.3. CocoaPods **11**

4.4. Model – pogled – pogledModel arhitekturni obrazac **11**

5. PROGRAMSKO RJEŠENJE APLIKACIJE *12*

5.1. Registracija i prijava **12**

5.2. Pregled dostava **14**

5.3. Detaljan prikaz dostave **16**

5.4. Kreiranje dostava **16**

5.5. Povezivanje radnje i dostavljača **17**

5.5.1. Pozivanje dostavljača za suradnju 17

5.5.1. Dodavanje radnje 18

5.6. Dodatne postavke **19**

5.6.1. Odabir jezika aplikacije 19

5.6.2. Pregled dodanih dostavljača/radnji 21

5.6.3. Upravljanje skočnim obavijestima 21

5.6.4. Odjava 22

6. PRIKAZ RADA APLIKACIJE ***24***

6.1. Prikaz omogućenih funkcionalnosti aplikacije **24**

6.2. Prikaz korištenja aplikacije **24**

6.2.1. Prikaz prijave 25

6.2.2. Prikaz registracije 26

6.2.3. Pregled i stvaranje dostava 27

6.2.4. Prikaz obavljanja akcija nad dostavom 29

6.2.5. Prikaz detalja dostave 31

6.2.6. Prikaz povezivanja dostavljača i radnje 32

6.2.7. Prikaz dodatnih postavki 34

7. ZAKLJUČAK ***38***

LITERATURA ***39***

SAŽETAK ***40***

ABSTRACT ***41***

ŽIVOTOPIS ***42***

1. UVOD

U današnjem društvu sve više ljudi sklono je kupovini hrane, odjeće pa čak i kućnih potrepština putem interneta, radije nego da odlaze u trgovinu ili restoran uživo. Kupovina putem interneta može biti puno brža i jednostavnija, no tu se uviđa potencijalni problem dostave kupljenih proizvoda. Taj problem se ne pojavljuje u slučajevima kada trgovina ili restoran već imaju svoju vlastitu dostavu, nego u slučajevima kada ta dostava izostaje te se vrši preko treće strane koja većinom nije toliko brza efikasna koliko bi mogla biti.

Tema ovog završnog rada je izrada mobilne aplikacije koja će omogućiti radnjama koje nemaju vlastitu dostavu, brz, siguran i efikasan način dostave svojih proizvoda na adresu krajnjeg kupca. Aplikacija omogućava registraciju dvije vrste korisnika, te interakciju između njih putem stvaranja novih dostava, te prihvaćanja ili odbijanja istih od strane dostavljača.

U nastavku ovog poglavlja navedena su neka postojeća rješenja koja nude cjelokupne sustave za naručivanje i dostavu proizvoda. Drugo poglavlje prikazuje nam model aplikacije uz detaljan opis svake pojedine komponente i mogućnosti. Treće poglavlje sadrži opis tehnologija koje su korištene za izradu aplikacije i arhitektura koje omogućuju da aplikacija bude proširiva, laka za održavanje i prilagodbu, te pisana na visokoj tehničkoj razini. Četvrto poglavlje predstavlja programsko rješenje razdijeljeno po pojedinim komponentama aplikacije. Za svaku komponentu pružen je detaljan plan i opis izrade, te dio programskog koda koji je zanimljiv ili specifičan za nju. Peto poglavlje nudi nam prikaz rada i vizualnog izgleda aplikacije, te postupak njenog korištenja.

1.1. Zadatak diplomskog rada

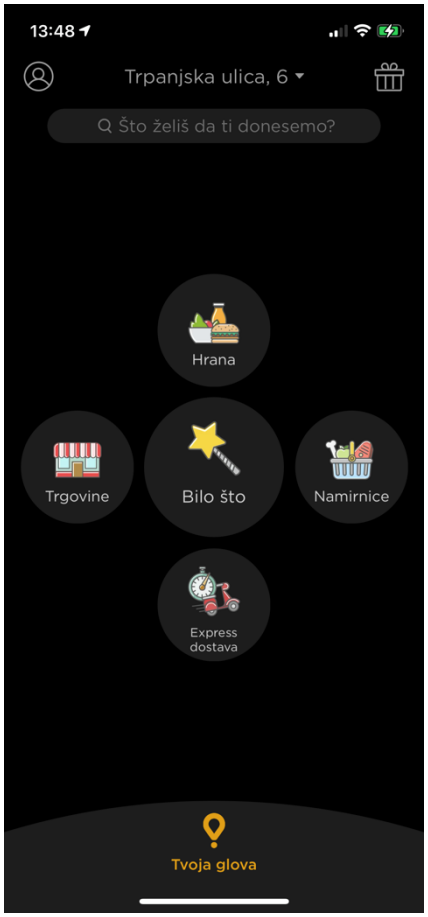
Zadatak ovog diplomskog rada bio je izraditi mobilnu aplikaciju na platformi iOS, koja će predstavljati sustav zadužen za upravljanje i komunikaciju između određene radnje i dostavljača. Radnja može biti trgovina, restoran ili nešto treće, a dostavljač predstavlja osobu koja je zadužena za dostavu paketa od radnje do krajnjeg kupca.

2. Prikaz postojećih rješenja za komunikaciju između dostavljača i radnji

Tržište je prepuno mnoštvom razvijenih rješenja komunikacije i suradnju između dostavljača, radnji pa čak i klijenata. U nastavku će biti prikazana tri vrlo sofisticirana i razvijena rješenja koja su vlasništvo velikih kompanija.

2.1. Glovo

Glovo je prvenstveno rješenje koje je razvijeno u cilju dostave hrane iz obližnjih restorana [1]. Stvorena je mobilna aplikacija, vidljiva na slici 2.1. i web sučelje u kojem korisnici mogu naručiti hranu. Preko iste aplikacije restorani zaprimaju narudžbe, a dostavljači ih prihvaćaju. Ugrađeno je i kartično plaćanje što ovu aplikaciju čini vrlo jednostavnom za korištenje.



Slika 2.1. Prikaz sučelja aplikacije Glovo.

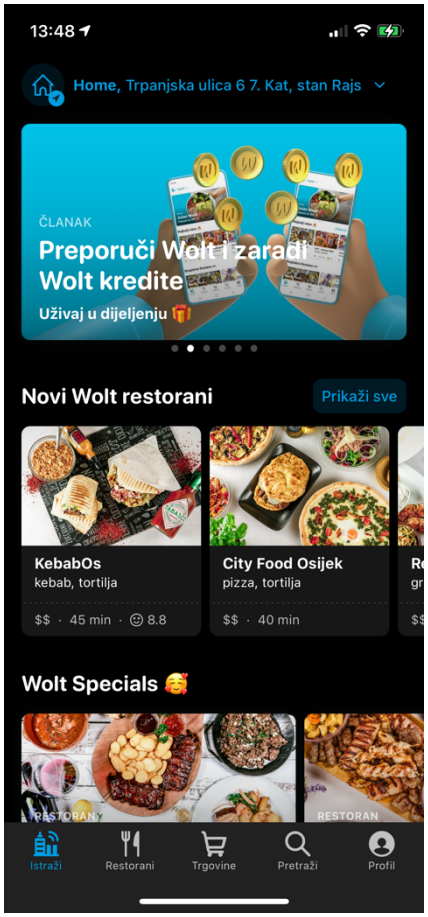
2.2. Pauza

Pauza, čiji je početni zaslon vidljiv na slici 2.2., je aplikacija koja se bazira većinski na dostavi hrane, no postoji i mali broj trgovina koje podržavaju njihove usluge. Isto kao i kod Glovo aplikacije, preko jednog sustava odvija se korak narudžbe od strane klijenta, prihvaćanja narudžbe od strane radnje i dostave od strane dostavljača [2].

Slika 2.2. Prikaz početnog ekrana aplikacije Pauza.

2.3. Wolt

Prema [3], Wolt je kao i prijašnje dvije aplikacije, započeo poslovanje kao sustav dostave hrane. Međutim velikom potražnjom, u sustav se uvodi dostava namirnica, lijekova i kućanskih potrepština. Uz gotovinsko plaćanje direktno dostavljaču, postoji i kartično koje je za većinu korisnika puno jednostavnije, pogotovo u kombinaciji za bezkontaktnom dostavom gdje dostavljač ostavlja paket ispred korisnikovih vrata. Sučelje aplikacije vidljivo je na slici 2.3.



Slika 2.3. Prikaz sučelja aplikacije Wolt.

2.4. Idejno rješenje vlastite aplikacije iDeliver

Prikaz dosadašnjih izazova i aplikacija koje ih rješavaju bio je motivacija za stvaranje vlastite aplikacije za komunikaciju dostavljača i radnji. Za razliku od vrlo ozbiljnih sustava koji su razvijani dugi niz godina i pružaju mnoštvo mogućnosti, ova aplikacija je za početak zamišljena kao vrlo jednostavan način komunikacije, koji trenutno ne uključuje klijenta u sustav. Plaćanja i naručivanja neće biti uključena u aplikaciju, a suradnja će se bazirati na međusobnom povjerenju radnje i dostavljača. Značajka aplikacije koja ju razlikuje od gore navedenih je ta da ne bi bila ograničena na dostavu hrane, već bi se kao radnja mogao registrirati bilo koji objekt, kao na primjer supermarket, trgovina namještaja, tržnica i drugi. Detaljan opis modela aplikacije biti će opisan u drugom poglavlju.

3. MODEL APLIKACIJE

U ovom poglavlju se nalazi detaljan opis modela aplikacije iDeliver. Raspisana je zamisao o svakom dijelu aplikacije, te su detaljno opisani zahtjevi aplikacije prema korisniku.

3.1. Korisnički zahtjevi na aplikaciju iDeliver

U ovoj mobilnoj aplikaciji, od korisnika se, nakon što preuzme aplikaciju, očekuje da se registrira. Ovisno o vrsti korisnika, može se obaviti registracija radnje ili dostavljača, a ona se razlikuje u podatcima koji su potrebni. Nakon toga, obavlja se prijava u aplikaciju i postoje različite mogućnosti ovisno o vrsti korisnika. Tako radnja može stvarati dostave i uređivati njihov status, pozivati dostavljače na suradnju, dodavati ih i uklanjati sa liste dostavljača sa kojima surađuje. Korisnik koji je dostavljač može prihvaćati i odbijati dostave, te prihvaćati pozive radnji i tako pristati na suradnju sa istom.

3.2. Registracija

Pri ulasku u aplikaciju, korisniku se prikazuje zaslon sa dva gumba, od kojih jedan vodi na zaslon prijave namijenjen radnjama, a drugi dostavljačima. Na tom zaslonu postoji tekst koji glasi „Nemate račun? Molimo, registrirajte se.“ . Klikom na taj tekst odlazimo na zaslon registracije.

3.2.1. Registracija radnje

Pri dolasku na zaslon registracije, vidi se forma koju korisnik mora popuniti kako bi uspješno registrirao svoju radnju. Forma se sastoji od adrese elektroničke pošte i OIB-a radnje, te oni moraju biti jedinstveni podatci. To znači da jedan OIB i jednu adresu elektroničke pošte može imati samo jedna radnja u tablici. Nadalje, dodatni podatci koji se moraju popuniti su ime radnje, adresa na kojoj se radnja može pronaći, te broj telefona na koji se radnja može kontaktirati ukoliko dođe do potrebe za tim. Naravno, uz sve te podatke, potrebno je unijeti lozinku koja će se koristiti za prijavu. Nju je potrebno unijeti dva puta kako ne bi došlo do greške pri unosu.

3.2.2. Registracija dostavljača

Zaslon registracije dostavljača se razlikuje od registracije radnje u podatcima koje je potrebno unijeti. Za razliku od radnje, dostavljač mora unijeti svoje ime, prezime i broj mobitela koji je jedinstven kao u slučaju OIB-a kod radnje. Kao neobavezan parametar, postoji i registracijski kod, koji dostavljač može dobiti od radnje prije nego što se registrirao. Ukoliko dostavljač unese registracijski kod, on će automatski biti u suradnji sa radnjom kojoj taj kod pripada. Kako bi se mogao prijaviti, korisnik također unosi lozinku koju mora ponoviti radi sigurnosti.

3.3. Prijava

Prijava korisnika se razlikuje u tome što radnja za prijavu koristi adresu elektroničke pošte i lozinku, dok dostavljač koristi broj mobilnog uređaja koji je upisao pri registraciji. Oba korisnika moraju upisati lozinku. Ukoliko je prijava uspješno obavljena, aplikacija odlazi na zaslon pregleda dostava.

3.4. Pregled dostava

Nakon što se korisnik bilo kojeg tipa, uspješno autentificira, prvo što će vidjeti je lista dostava. Ukoliko je korisnik radnja, vidjet će sve dostave koje su stvorene od strane same sebe. Ovisno o statusu dostave, svaka je predstavljena drugačijim vizualnim pregledom kako bi korisnik lako razlikovao dostavu jednog statusa od dostave drugog statusa. Mogući statusi dostave su: „status čekanja“, „u tijeku“, „dostavljeno“ i „otkazano“. Korisnik tipa dostavljač će u svojoj listi dostava imati dostave svih radnji sa kojima surađuje. Pri prvoj prijavi, ukoliko dostavljač nema registracijski kod, lista će mu biti prazna. Kako bi iskoristio sve beneficije aplikacije, mora biti pozvan na suradnju od strane neke radnje, te biti odabran pri stvaranju novih dostava.

3.5. Stvaranje dostave

Svaka radnja ima mogućnost kreiranja nove dostave koja će biti na prikazana svim dostavljačima u suradnji s njom. Kako bi se dostava kreirala, potrebno je unijeti određene informacije. Najbitnije su ime, prezime primatelja dostave, te adresa na koju dostava mora biti isporučena. Potrebno je unijeti i kratak opis sadržaja dostave, te ukoliko postoji, naputak za dostavljača (ukoliko pošiljka sadrži nešto lomljivo ili slično). Neobavezan podatak je kontakt broj na koji se može dobiti primatelja, kako bi dostavljač, ukoliko dođe do određenih problema pri dostavi, mogao kontaktirati primatelja i u što kraćem roku otkloniti problem. Nakon što korisnik pravilno popuni tražene podatke, otvara mu se mogućnost prelaska na drugi zaslon na kojem bira kojim točno dostavljačima želi napraviti ovu dostavu vidljivom. Minimalno jedan dostavljač mora biti odabran kako bi dostava mogla biti stvorena. Oni dostavljači koji nisu odabrani, neće niti vidjeti dostavu u listi njihovih dostava.

3.6. Pregled povezanih radnji i dostavljača

Svaki korisnik ima mogućnost pregleda ostalih korisnika sa kojima surađuje. Tako radnja može vidjeti sve dostavljače koji su povezani s njom i mogu prenositi njene pakete. Osim imena dostavljača, radnja ima uvid i u broj dostava koje je dostavljač odbio preuzeti. Ukoliko primijeti

da dostavljač ne obavlja svoj posao kako bi trebao i odbija sve dostave ili ih ne dostavlja na dovoljno dobar način, postoji mogućnost prekida suradnje. Suradnju može prekinuti samo radnja no ne i dostavljač. On može samo vidjeti imena radnji sa kojima surađuje i adrese njihovih lokacija.

3.7. Pozivanje dostavljača na suradnju

Ukoliko se obujam posla neke radnje poveća, povećava se i potreba za više dostava, pa tako i za više dostavljača sa kojima surađuju. Odlaskom na zaslon pozivanja novih dostavljača, radnja ima uvid u one koji imaju registriran račun na aplikaciji te u broj njihovih otkazanih dostava. Tako može pozvati samo najbolje za koje su sigurni da će posao obavljati dobro. Klikom na pojedinog dostavljača, automatski mu se šalje obavijest na mobitel na čijem je broju registriran. Također, postoji i drugi, ručni način pozivanja dostavljača. Ukoliko on nije na popisu dostavljača dostupnih za pozivanje, a radnja ima uvid u njegov broj mobilnog uređaja, može ručno unijeti njegov broj i poslati poziv. Na njegov mobitel poslati će se link preko kojega on može prihvatiti poziv.

3.8. Dodavanje radnje

Postoji više načina na koji dostavljač može stvoriti suradnju. Registracijski kod može primiti preko skočne obavijesti ili preko SMS poruke. Ukoliko već ima registriran račun, primiti će obavijest, te će mu se klikom na nju otvoriti forma za dodavanje radnje sa već unesenim kodom. Zatim je potrebno samo kliknuti na gumb „Dodaj“ i radnja je dodana. Ukoliko korisnik nema izrađen račun, a primio je poruku sa linkom koja ga poziva na suradnju, klikom na taj link, otvara mu se aplikacija. Početni zaslon je zaslon registracije dostavljača, sa već unaprijed ispunjenim kodom registracije, na korisniku je samo da popuni ostatak podataka i registrira se. Nakon uspješne registracije, biti će povezan sa radnjom koja mu je poslala kod.

3.9. Interakcija sa pojedinom dostavom

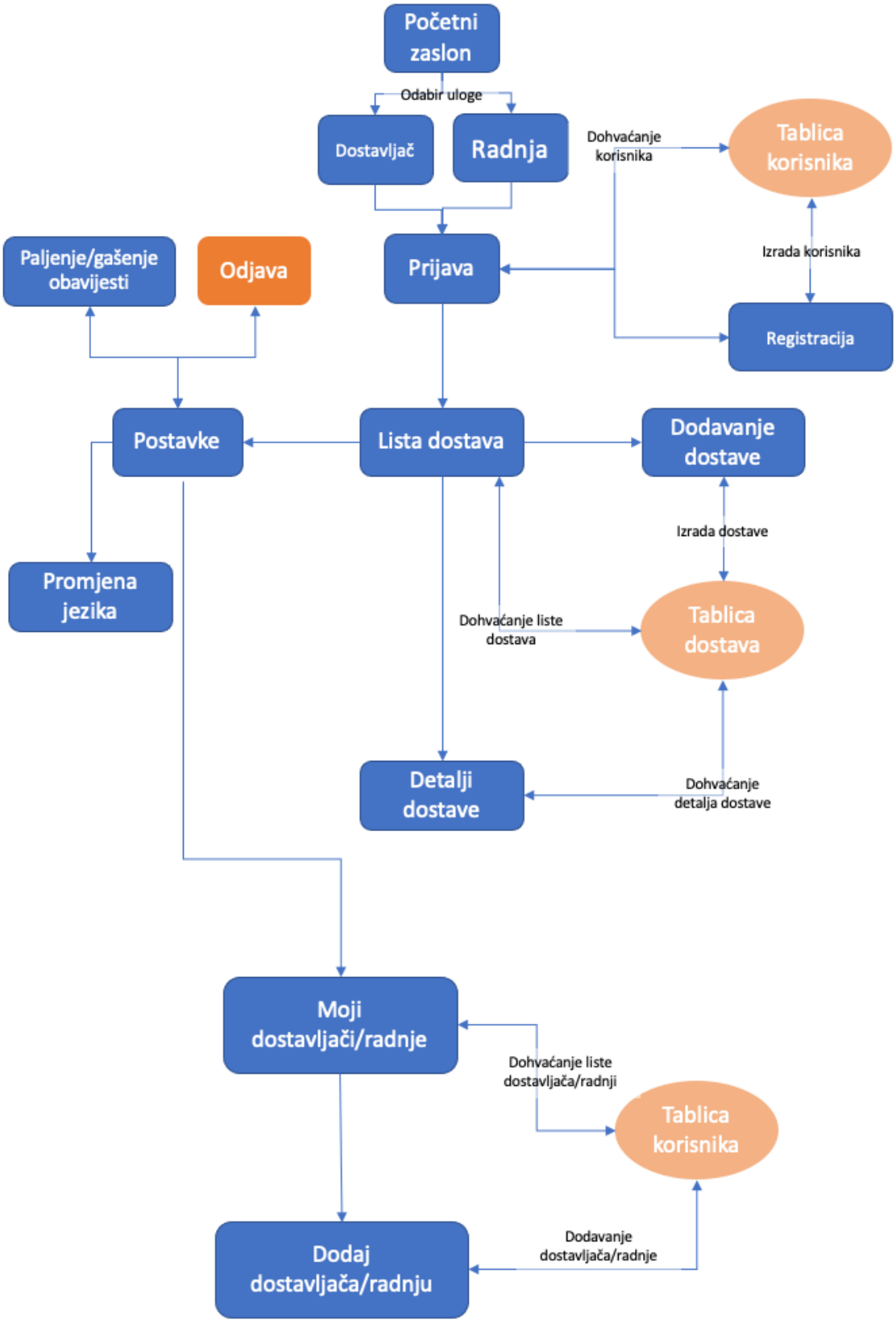
Svaka dostava ima svoj jedinstveni identifikacijski broj, koji ju predstavlja. Svaka interakcija sa dostavom odvija se pomoću tog broja. Tako dostavljač svaku dostavu koja ima „status čekanja“ može prihvatiti. Nakon prihvaćanja dostave, njen status se mijenja na „u tijeku“. Dostave koje su u tijeku mogu biti uspješno dostavljene, čime njihov status prelazi u „dostavljeno“, ili mogu biti poništene, čime se status vraća na „status čekanja“. Ukoliko dostavljač poništi dostavu, nema više pravo na prihvaćanje te dostave, te mu se povećava brojač poništenih dostava. Svakom dostavljaču bi trebalo biti u cilju imati što manje takvih dostava, jer će svaka radnja htjeti surađivati sa dostavljačima koji imaju najmanji broj poništenih dostava. Nakon kreiranja dostave, radnja može

svaku dostavu, koja nije „otkazana“ ili „dostavljena“, poništiti. Ukoliko je status dostave bio „u tijeku“, dostavljač koji ju je prihvatio biti će obavješten o njenoj poništavanju.

3.10. Dodatne postavke

Kako bi aplikacija bila osobnija, korisnik ima par mogućnosti kako bi je što bolje prilagodio sebi. Prva takva mogućnost je odabir jezika aplikacije. Za početak je zamišljeno dodavanje dva osnovna jezika, a to su Hrvatski i Engleski, a rastom aplikacije bi postojala vjerojatnost dodavanja još jezika. Osim jezika, korisnik može birati hoće li primati skočne obavijesti.

Model aplikacije prema opisima navedenim u ovom poglavlju može se vidjeti na slici 3.1.



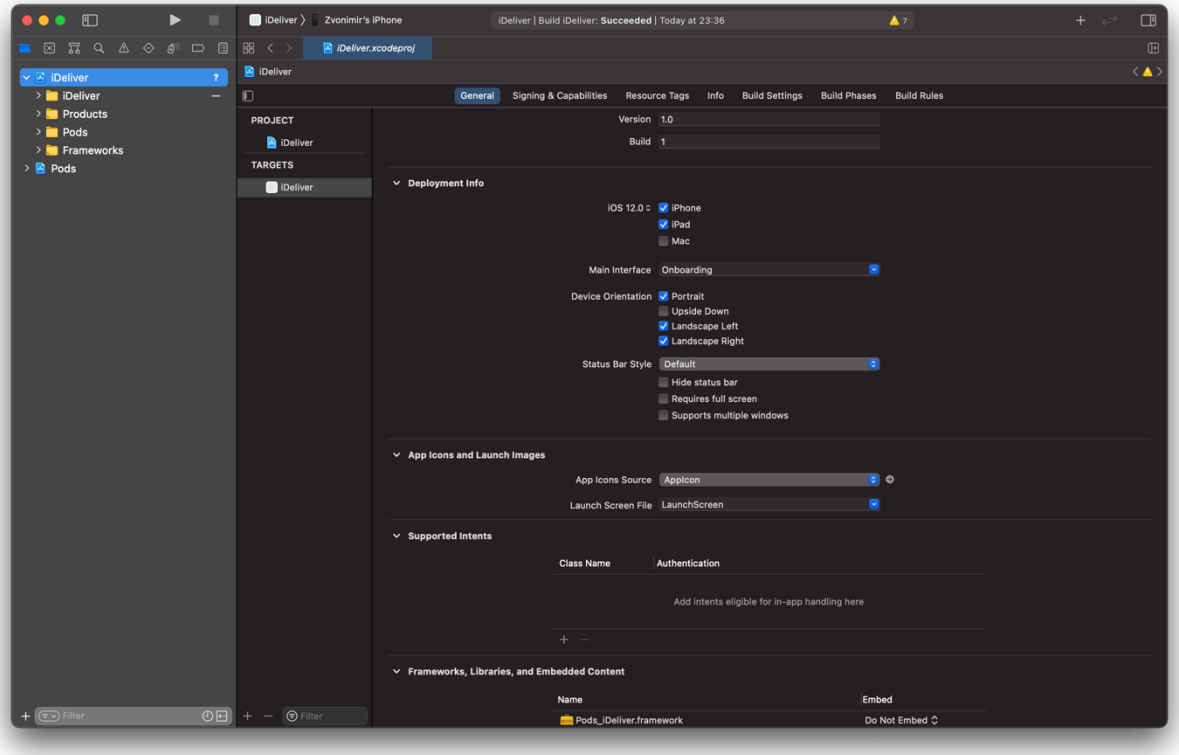
Slika 3.1 Model aplikacije.

4. OPIS KORIŠTENIH ALATA I TEHNOLOGIJA

Cijela mobilna aplikacija je napravljena je u integriranom razvojnom okruženju Xcode. Kod je pisan u objektno orijentiranom programskom jeziku Swift u kombinaciji sa korištenjem CocoaPods upraviteljem ovisnosti. Kako bi se aplikacija u budućnosti lakše održavala, ispravljala i unaprjeđivala, pisana je po uzoru na model- pogled- pogledmodel arhitekturni obrazac.

4.1. Xcode

Prema [4], Xcode je službeno Appleovo integrirano razvojno okruženje za macOS čije sučelje možemo vidjeti na slici 4.1. Na njemu se mogu stvarati programska rješenja prilagođena za macOS, iOS, iPadOS, watchOS i tvOS.



Slika 4.1. Prikaz sučelja integriranog razvojnog okruženja Xcode-a.

Objavljen je 2003. godine, a posljednja stabilna verzija je 12.5. koja se besplatno može preuzeti iz App store-a. Xcode je strogo ograničen na macOS, te ga nije moguće pokrenuti na ostalim platformama kao Windows ili Linux. Ovaj sustav ima ugrađen kreator grafičkog sučelja koji se naziva „Storyboard“, te korisnici mogu lako slagati izgled svojih zaslona pomoću već postojećih komponenti. U svrhu testiranja, uz Xcode dolazi i mnoštvo već instaliranih simulatora, kako

mobilnih uređaja, tako i satova, tableta itd. Kao i većina modernih razvojnih okruženja, i ovaj ima sustav automatskog detektiranja i ispravljanja pogrešaka pri pisanju, što programerima uvelike skraćuje i olakšava posao. Također postoji automatsko nadopunjavanje koda što ubrzava razvijanje programa.

4.2. Programski jezik Swift

Kompaktniji, bolji, intuitivniji i napredniji nasljednik Objektnog- C programskog jezika je Swift [5]. Građen je da bude brz, koristeći najmodernije tehnologije kompajliranja. Preveden je na optimizirani nativni kod koji iskorištava najviše od današnjeg hardvera. Vrlo se lako usvaja pa je savršen programski jezik za početnike. Neke od značajki ovog jezika su da se varijable uvijek inicijaliziraju prije korištenja, polja imaju mogućnost provjere izlaska iz dometa, brojčane vrijednosti su osigurane od prelijevanja. Također jako bitna značajka je da je korištenje memorije automatski kontrolirano od strane sustava.

4.3. CocoaPods

Prema [6], CocoaPods je upravitelj ovisnostima za Swift i Objective- C programske jezike. Posjeduje preko 83 tisuće biblioteka i korišten je u više od 3 milijuna mobilnih aplikacija. Građen je na Ruby objektno orijentiranom programskom jeziku, pa ga je lako instalirati sa komandom u terminalu sa slike 4.2. CocoaPods omogućava uvoz biblioteka koje mogu uvelike olakšati programiranje ili implementiranje neke značajke.

```
$ sudo gem install cocoapods
```

Slika 4.2. Prikaz komande za instalaciju CocoaPods- a

4.4. Model – pogled – pogledModel arhitekturni obrazac

Model – pogled – pogledModel, prema [7] poznatiji kao MVVM (Model – View – ViewModel), je strukturni obrazac koji objekte razdvaja na tri prepoznatljive skupine. Model upravlja sa podatcima aplikacije koji su u većini slučajeva male klase ili strukture. Pogled (View) prikazuje vizualne elemente i kontrole na ekranu. Ti vizualni elementi su najčešće podklase klase UIView. MVVM je vrlo sličan MVC obrascu no MVVM izbjegava problem pretrpanih ViewControllera.

5. PROGRAMSKO RJEŠENJE APLIKACIJE

U ovom poglavlju prikazana je svaka pojedina komponenta od koje je sačinjena ova aplikacija. Svaka je popraćena dijelom programskog koda koji je specifičan za nju.

5.1. Registracija i prijava

Cijela aplikacija se može podijeliti na dva slučaja, od kojih je jedan slučaj kada je korisnik dostavljač, a drugi kada je korisnik radnja. Iz tog razloga stvorena je enumeracija „WGUserType“ sa ta dva slučaja. Na početnom zaslonu postoje dva gumba, svaki predstavlja jedan slučaj. Kako bi kod bio ponovno iskoristiv, na oba gumba poziva se ista funkcija vidljiva na slici 5.1., kojoj se predaje različit parametar, ovisno o tome koji točno gumb je stisnut.

```
// MARK: - Actions
@IBAction private func onStoreButtonTapped(_ sender: UIButton) {
    WGOnboardingControllerHelper.pushLoginViewController(with: .store)
}

@IBAction private func onCourierButtonTapped(_ sender: UIButton) {
    WGOnboardingControllerHelper.pushLoginViewController(with: .courier)
}
```

Slika 5.1 Prikaz poziva funkcije za prikaz zaslona prijave.

Nakon što korisnik odaberi slučaj želi, odlazi na zaslon za prijavu. Taj zaslon se konfigurira ovisno o odabranom slučaju, što je vidljivo na slici 5.2. Jedina razlika između dva slučaja ovdje je ta da dostavljač unosi broj mobilnog uređaja za prijavu, a radnja unosi adresu električne pošte.

```
private func setupUI() {
    switch viewModel.type {
    case .courier:
        emailOrPhoneNumberInputView.type = .phoneNumber

    case .store:
        emailOrPhoneNumberInputView.type = .email

    default:
        emailOrPhoneNumberInputView.type = .email
    }
}
```

Slika 5.2 Prikaz konfiguracije zaslona ovisno o vrsti korisnika.

Kako bi kod bio jednostavniji i lakši za razumijevanje, napravljeno je univerzalno polje za unos. To polje sastoji se od naslova, polja za unos, teksta upozorenja te ikonice koja može biti skrivena. Svakom polju pridodaje se vrsta teksta koji će biti unošen u njega. Tako je stvorena enumeracija „WGFormInputViewType“ prikazana na slici 5.3., koja se sastoji od svih slučajeva unosa u aplikaciji. Ti slučajevi pokrivaju prijavu, registraciju, stvaranje dostave, dodavanje radnje i slično. Svaka vrsta ima informaciju o tome kako joj glasi naslov, treba li biti prikazana ikonica i ako da, njeno ime, kao i tekst koji će biti ispisan u slučaju krivog unosa u polje. Stvaranjem univerzalnog polja za unos smanjuje se ponavljanje koda, te se olakšava stvaranje izmjena. Npr. ako se promijeni naslov nekog polja, njega je potrebno izmijeniti samo na jednom mjestu, umjesto na svakom mjestu gdje se to polje pojavljuje.

```
12  // MARK: - Input view type
13  enum IDFormInputViewType: String {
14      case username
15      case email = "email"
16      case password = "password"
17      case passwordConfirmation = "password_confirmation"
18      case address = "address"
19      case phoneNumber = "phone"
20      case name = "name"
21      case storeKey = "store_key"
22      case registrationKey = "registration_key"
23      case optionalPhoneNumber = "optional_phone"
24
```

Slika 5.3 Prikaz enumeracije vrsta polja za unos.

Na svaku promjenu unosa u polju, poziva se funkcija koja provjerava jesu li sva polja popunjena ispravnim podatcima. Ukoliko je to slučaj, gumb za prijavu postaje aktivan te ga korisnik može stisnuti. Ukoliko tekst koji je unesen u polje, ne zadovoljava uvjete koji su za to polje postavljeni, automatski se ispisuje poruka upozorenja koja korisniku daje do znanja na koji način može popraviti neispravni unos. Nakon klika na gumb prijave, preuzimaju se tekstovi upisani u polja, te se šalju na server. Naravno točna adresa zahtjeva na koji se podatci šalju razlikuje se ovisno o vrsti prijave. Server zatim provjerava postoji li korisnik sa unesenim podatcima. U slučaju da korisnik ne postoji, server vraća poruku pogreške te obavještava korisnika da je unio krive podatke. Ako postoji, vraća mu autorizacijski ključ koji korisnik nadalje koristi u svim sljedećim zahtjevima na server i aplikacija prelazi na zaslon pregleda dostava. Na sličan način odvija se i registracija korisnika. Ona se sastoji od više različitih polja za unos, od kojih nisu sva obavezna. Nakon ispravnog unosa svih potrebnih podataka, gumb za registraciju postaje aktivan, te klikom na njega

korisnik šalje zahtjev na server. Server zatim provjerava postoji li već korisnik sa tom jedinstvenom identifikacijskom oznakom (adresa električne pošte u slučaju radnje, broj mobilnog telefona u slučaju dostavljača), te ukoliko postoji vraća korisniku adekvatnu poruku. Ukoliko ne postoji, stvara se novi korisnik, a zaslon aplikacije se mijenja na zaslon prijave. Za prikazivanje poruka sa servera također se koristi prilagođena komponenta. Ona je napravljena na način da se njen izgled lako može promijeniti i prilagoditi, isto kao i funkcionalnost Moguće ju je pozvati bilo gdje u aplikaciji, potrebno je samo predati tekst koji će se ispisivati u njoj.

5.2. Pregled dostava

Nakon uspješne prijave prikazan je zaslon pregleda dostava. Dostave se pomoću zahtjeva na server dohvaćaju u JSON formatu. Kako bi bilo moguće prikazati dostave na zaslonu, potrebno je napraviti model dostave. On se sastoji od desetak parametara, vidljivih na slici 5.4., koji opisuju svaku pojedinu dostavu.

```
10  struct IDDelivery {
11      let id: Int
12      let storeId: Int
13      let userId: Int?
14      let customerId: Int
15      let customer: IDCustomer
16      let description: String
17      let status: IDDeliveryStatus
18      let store: IDStore
19      let createdAt: Date
20
21      init(json: JSON) {
22          self.id = json["id"].intValue
23          self.store = IDStore(json: json["store"])
24          self.storeId = json["store_id"].intValue
25          self.userId = json["user_id"].intValue
26          self.customer = IDCustomer(with: json["customer"])
27          self.customerId = json["customer_id"].intValue
28          self.description = json["description"].stringValue
29          self.status =  IDDeliveryStatus(rawValue: json["status"].stringValue) ?? .pending
30          self.createdAt = Date().createDate(with: json["created_at"].stringValue)
31      }
32  }
```

Slika 5.4 Prikaz modela dostave.

Možemo vidjeti da svaka dostava ima svoj identifikacijski broj po kojemu je jedinstvena, podatke o radnji koja je stvorila dostavu, kao i o krajnjem kupcu koji tu dostavu treba primiti. Jedan od podataka je i status dostave koji se spominjao ranije u tekstu. O njemu ovisi na koji način će dostava biti prikazana korisniku, te koje akcije će mu biti ponuđene. Dostave su na ekranu prikazane u tablici [8] sa jednim stupcem. Nju je potrebno konfigurirati na način da joj se pruže

podatci o broju redaka, izgleda pojedinog retka. Za potrebe konfiguriranja stvara se varijabla koju nazivamo „struktura“. Ona sadrži sekcije i tipove redaka koji će biti prikazani u tablici. Pri svakom pozivu „strukture“, ona ponovno računa sve sekcije i retke kako bi pri osvježivanju tablice uvijek bili prikazani najnoviji podatci. Logika „strukture“ je sljedeća. Ukoliko postoje dostave, dodaje se jedna sekcija, te će broj redaka u toj sekciji biti jednak broju dostava. Ovaj primjer, vidljiv na slici 5.5., je dosta jednostavan jer svaka dostava može biti predstavljena jednim tipom ćelije pa nema potrebe za umetanjem više od jedne sekcije u tablicu.

```
31      // MARK: - Computed properties
32      private var structure: [(header: Header, cells: [Cell])] {
33          var structure: [(header: Header, cells: [Cell])] = []
34          guard tableView.isInitialLoadInProgress == false else {return []}
35
36          if viewModel.deliveries.count > 0 {
37              tableView.hideEmptyStateView()
38              structure.append((.deliveries, viewModel.deliveries.map({_ in .delivery})))
39          } else {
40              tableView.showEmptyStateView(using: GDEmptyStateView.Configurator(title: "no_deliveries".localized(),
41                                                                                message: "no_deliveries_\(type)_message".localized(),
42                                                                                actionTitle: nil,
43                                                                                action: nil))
44          }
45
46          return structure
47      }
```

Slika 5.5 Prikaz varijable strukture tablice.

Ukoliko nema niti jedne dostave, kao pozadina tablice biti će prikazana poruka „praznog stanja“ u obliku komponente. Toj komponenti se predaje naslov i podnaslov koji će biti prikazani kako bi korisnik bio prikladno obavješten o zbivanju u aplikaciji. Komponenta ima mogućnost dodavanja i gumba proizvoljne akcije, no u ovom slučaju ona nije iskorištena. Nakon stvaranja strukture, potrebno je stvoriti prilagođenu ćeliju vidljivu na slici 5.6., koja će predstavljati svaku pojedinu dostavu.

```
class IDDeliveryTableViewCell: UITableViewCell {

    // MARK: - Outlets
    @IBOutlet private weak var containerView: UIView!
    @IBOutlet private weak var descriptionLabel: GDLabel!
    @IBOutlet private weak var acceptOrDeliveredButton: GDButton!
    @IBOutlet private weak var cancelButon: GDButton!
    @IBOutlet private weak var overlayView: UIView!
    @IBOutlet private weak var deliveredLabel: GDLabel!
    @IBOutlet private weak var dateLabel: GDLabel!
    @IBOutlet private weak var createdAtLabel: GDLabel!
```

Slika 5.6 Prilagođena ćelija dostave.

Svaka ćelija sadrži kratki opis dostave, koji će u ovoj ćeliji biti ograničen samo na jedan redak, te ukoliko korisnik želi vidjeti cijeli opis, mora ući u detalje dostave. Osim opisa, postoje i akcijski gumbovi, maksimalno dva, a mogu biti prikazani oba odjednom, jedan ili nijedan, ovisno o statusu dostave i o vrsti korisnika. Radi estetike, svakoj ćeliji su dodani zakrivljeni rubovi, sjena te različita boja pozadine po kojoj korisnik može razlikovati dostave po statusu. Postoji i tekst koji prekriva cijeli sadržaj ćelije, ali je inicijalno skriven. Otkriva se tek kada je status dostave „dostavljeno“ ili „otkazano“ te se ovisno o statusu mijenja i prikazani tekst.

5.3. Detaljan prikaz dostave

Pošto u ćeliji koja prikazuje dostave u listi, nije moguće prikazati sve podatke o njoj, potrebno je stvoriti zaslon koji prikazuje pojedinu dostavu i njene detalje. Klikom na ćeliju jedne dostave, inicijalizira se zaslon detalja dostave, te mu se prosljeđuje dostava iz ćelije. Zatim se parametri iz dostave ispisuju na ekranu, te ovisno o statusu dostave, postavljaju se potrebni gumbovi za interakciju. Pošto broj mobilnog uređaja klijenta nije obavezan za stvaranje dostave, potrebno je implementirati provjeru o njegovom prikazu. Ukoliko broj postoji, biti će prikazan na ekranu, a ukoliko ne postoji, biti će prikazan tekst „Nije dostupan“. Postavljanje parametara vidljivo je na slici 5.7.

```
private func setupUI() {
    self.statusValueLabel.text = delivery.status.rawValue.localized()
    self.descriptionValueLabel.text = delivery.description
    self.addresValueLabel.text = delivery.customer.address
    self.customerValueLabel.text = delivery.customer.fullName
    self.storeValueLabel.text = delivery.store.name
    self.storePhoneLabelValue.text = delivery.store.phoneNumber
    self.storeAddressValueLabel.text = delivery.store.address

    customerPhoneValueLabel.text = delivery.customer.phoneNumber.isEmpty ? "not_available".localized() : delivery.customer.phoneNumber
}
```

Slika 5.7. Prikaz postavljanja parametara dostave.

5.4. Kreiranje dostava

Svaka radnja ima mogućnost kreiranja novih dostava klikom na znak plus u gornjem desnom dijelu ekrana. Nakon klika, postavljen je zaslon sa formom koju je potrebno ispuniti za odlazak na sljedeći korak. Taj zaslon je također izveden pomoću tablice i dvije vrste ćelija. Jedna vrsta je ćelija tekstualnog unosa, a druga ćelija gumba za nastavak. Pošto postoji pet parametara za ispunu, ćelija tekstualnog unosa je prikazana pet puta, a gumba za nastavak jednom. Četiri parametra su obavezna, dok jedan nije. Gumb za nastavak će inicijalno biti postavljen u onemogućeno stanje i ostati će u njemu sve dok korisnik ne popuni obavezna polja. Nakon toga gumb mijenja svoj izgled i time naznačuje da ga je nadalje moguće kliknuti. Nakon klika na gumb za nastavak, prikazuje se

zaslon popisa svih dostavljača koji su povezani sa radnjom pa samim time imaju mogućnost dostave paketa za tu radnju. Korisnik na tom zaslonu vidi ime, prezime i broj otkazanih dostava svakog pojedinog dostavljača. Kako bi kreirao dostavu, mora odabrati minimalno jednog, ali može ih odabrati i sve, ovisno o potrebama i željama klijenata. Nakon što je odabran barem jedan dostavljač, gumb za završetak u gornjem desnom dijelu ekrana postaje omogućen, te klikom na njega radnja stvara dostavu. Dostava se stvara na način da se serveru šalju pojedini parametri uneseni u formu, te jedinstveni identifikatori odabranih dostavljača u obliku polja. Zatim server sprema dostavu u bazu i obavještava o tome aplikaciju. Ukoliko se sve odvilo uspješno, aplikacija obavještava zaslon sa listom dostava da treba osvježiti podatke, te se zatim vraća na taj zaslon, što vidimo na slici 5.8. Korisnik zatim vidi osvježene podatke sa novom dostavom na vrhu tablice.

```
func didAddDelivery() {
    self.dismissProgress()
    NotificationCenter.default.post(name: .WGRefreshDeliveries, object: nil)
    navigationController?.popToRootViewController(animated: true)
}
```

Slika 5.8 Funkcija koja se poziva nakon uspješnog kreiranja dostave.

5.5. Povezivanje radnje i dostavljača

5.5.1. Pozivanje dostavljača za suradnju

Radnja ima mogućnost pozivanja dostavljača na suradnju, a to se odvija na sljedeći način. Odlaskom na zaslon dodavanja, svi dostavljači koji nisu povezani sa prijavljenom radnjom, su prikazani u tablici pomoću već postojeće ćelije. Klikom na pojedinu ćeliju dostavljača, šalje se zahtjev na server sa njegovim identifikacijskim brojem. Server zatim kreira registracijski kod, pomoću kojeg se dostavljač može povezati sa radnjom. Sljedeća stavka je dostavljanje registracijskog koda dostavljača, a ono se obavlja funkcijom sa slike 5.9. Postoji dva načina za to, a oni su skočna obavijest [9] ili SMS poruka. Ukoliko je dostavljač kojem se treba poslati kod prijavljen na skočne obavijesti, biti će mu poslana obavijest koja u sebi sadrži registracijski kod. Ukoliko nije prijavljen na skočne obavijesti, radnji se prikazuje prozor za izradu SMS poruke u koji je automatski ispunjen broj mobitela dostavljača te je u sadržaj poruke stavljena univerzalna veza sa registracijskim kodom.

```
func didInviteCourier(registrationKey: String, phoneNumber: String, shouldSendSMS: Bool, isRegistered:
    Bool) {
    self.dismissProgress()
    if shouldSendSMS {
        self.sendSMS(phoneNumber: phoneNumber, registrationKey: registrationKey, isRegistered:
            isRegistered)
    } else {
        showMessageHud(with: "courier_successfully_invited".localized())
    }
}
```

Slika 5.9 Prikaz funkcije dostave registracijskog koda dostavljaču.

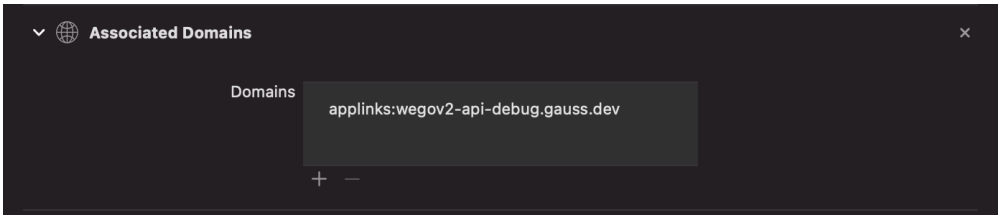
5.5.1. Dodavanje radnje

Ukoliko radnja primi skočnu obavijest, klikom na nju korisniku se prikazuje prozor u koji je automatski upisan kod iz obavijesti, slika 5.10.

```
        case .registrationKey:
            let topVC = UIApplication.topViewController()
            if let vc = topVC as? WGDialogViewController {
                // if top VC is the dialog, just input the registration key from the push notification
                vc.setRegistrationKey(with: notification.registrationKey)
            } else {
                // if the top VC is any other controller than dialog, push dialog on top and input the
                    registration key
                WGUsersControllerHelper.presentAddStoreController(with: notification.registrationKey)
            }
```

Slika 5.10 Prikaz rukovanja skočnim obavijestima.

Korisnik zatim potvrđuje unos klikom na gumb i šalje zahtjev za povezivanje sa radnjom. Ako zahtjev bude uspješan, radnja se nadalje prikazuje na popisu spojenih radnji. Opcija kada dostavljač primi poruku sa univerzalnom vezom je malo kompliciranija jer je potrebno implementirati logiku univerzalne veze. Prema [10], univerzalna veza je tehnologija koja omogućava da korisnik klikom na vezu, direktno ode na neki od sadržaja aplikacije. Kako bi se takvo ponašanje postiglo, prvo je potrebno dodati domenu, koju planiramo koristiti za univerzalnu vezu, u grupu povezanih domena vidljivu na slici 5.11.



Slika 5.11 Prikaz povezanih domena.

Nakon što je domena dodana, potrebno je u AppDelegate datoteku dodati funkciju koja će rukovati sa zahtjevima koji odgovaraju na univerzalne veze. U njoj rastavimo univerzalnu vezu na komponente, zatim ovisno o sadržaju pojedine komponente, odlučujemo kako ćemo rukovati vezom, odnosno, koji dio aplikacije će biti otvoren i što će biti odrađeno nakon klika na vezu. U slučaju ove aplikacije, postoje dvije vrste univerzalne veze. Dostavljač može već imati izrađen korisnički račun. U tom slučaju, klikom na univerzalnu vezu otvara se prozor gdje je već upisan registracijski kod, te on samo potvrđuje unos i povezuje se sa radnjom. Ako nema izrađen korisnički račun, klikom na link odlazi na zaslon registracije. Rastavljanjem univerzalne veze na komponente, vidljivo na slici 5.12., dolazi se do registracijskog koda, te je on unaprijed upisan na zaslonu registracije. Ostale podatke za registraciju korisnik ispunjava samostalno, a nakon ispune i uspješne registracije računa, automatski je povezan za radnjom čiji kod je bio unesen.

```
static func react(on userActivity: NSUserActivity) -> Bool {
    guard let url = userActivity.webpageURL,
          let components = URLComponents(url: url, resolvingAgainstBaseURL: true),
          let type = WGUniversalLinkType.from(components.path) else { return false }

    switch type {
    case .registerStoreKey:
        guard WGManager.shared.loggedUser != nil, WGManager.shared.loggedUser?.userType == .courier else
            { return false }
        let registrationKey = url.lastPathComponent

        WGUsersControllerHelper.presentAddStoreController(with: registrationKey)

    case .registerCourierKey:
        guard WGManager.shared.loggedUser == nil, let topVC = UIApplication.topViewController() else {
            return false }
        let registrationKey = url.lastPathComponent
        if let vc = topVC as? WGRegisterViewController {
            vc.setCourierRegisterKey(with: registrationKey)
        } else {
            WGOnboardingControllerHelper.pushCourierRegisterViewController(with: registrationKey)
        }

    }
    return true
}
```

Slika 5.12 Prikaz rastavljanja univerzalne veze.

5.6. Dodatne postavke

Klikom na ikonu kotačića na početnom zaslonu, korisniku se otvara bočna stranica sa dodatnim postavkama aplikacije.

5.6.1. Odabir jezika aplikacije

Ova značajka izvedena je pomoću Cocoapod dodatka zvanog „Localize-Swift“ [11]. Da bi omogućili višejezičnost u aplikaciji, moramo stvoriti datoteku sa prijevodima za svaki jezik koji će aplikacija sadržavati. Ova aplikacija imati će Hrvatski i Engleski jezik, tako da je potrebno

stvoriti dvije datoteke. U te datoteke pišemo prijevode tekstova koji se koriste u aplikaciji u obliku „ključ“ : „prijevod“, vidljivo na slici 5.13.

```
"login" = "Prijava";
"register" = "Registracija";

"email_upp" = "EMAIL";
"phone_number_upp" = "BROJ MOBITELA";
"password_upp" = "LOZINKA";
"password_confirmation_upp" = "POTVRDA LOZINKE";
"store_key_upp" = "STORE KEY";
"registration_key_upp" = "REGISTRATION KEY";
"store_name_upp" = "IME RADNJE";
"first_name_upp" = "IME";
"last_name_upp" = "PREZIME";
"address_upp" = "ADRESA";
"description_upp" = "OPIS";
"continue" = "NASTAVI";
```

Slika 5.13 Prikaz datoteke prijevoda.

Korištenje tih tekstova svodi se na upisivanje ključa uz dodatak .localized() na kraj. On zatim odlazi u datoteku jezika koji je postavljen i vraća prijevod koji je postavljen za taj ključ. Biranje inicijalnog jezika odvija se na način da se provjeri jezik postavljen na mobilnom uređaju. Ukoliko taj jezik postoji u ponuđenim jezicima aplikacije, on se postavlja za početni jezik. Ukoliko ne postoji, za početni jezik bira se engleski. Korisnik u svakom trenutku može u postavkama promijeniti jezik aplikacije. Jezici omogućeni u aplikaciji prikazani su mu u tablici uz prikaz zastave. Klikom na željeni jezik, poziva se funkcija za postavljanje jezika vidljiva na slici 5.14.

```
func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
    guard let language = languages[safe: indexPath.row] else {return}
    Localize.setCurrentLanguage(language.rawValue)
    self.navigationController?.popViewController(animated: true)
}
```

Slika 5.14 Postavljanje jezika u aplikaciji.

Pošto je zaslon dostava jedini zaslon na koji korisnik može biti vraćen nakon promjene jezika, samo tamo je potrebno postaviti slušatelja. On sluša na događaj promjene jezika, te nakon što se događaj okine, pokreće funkciju koja postavlja nove prijevode na taj zaslon.

5.6.2. Pregled dodanih dostavljača/radnji

Korisnik u svakom trenutku ima uvid u ostale korisnike koji sa njim surađuju. To znači da radnja može vidjeti sve dostavljače koji za nju dostavljaju pakete. Odlaskom na taj zaslon u tablici su prikazani svi dostavljači dohvaćeni sa servera. Radi lakšeg pregleda i ponovnog korištenja istih komponenti, za prikaz koristi se ćelija koja se već koristila u odabiru dostavljača pri kreiranju dostave. Tablica u svojim ugrađenim akcijama ima mogućnost da se povlačenjem ćelije ulijevo ili udesno izvrši željena funkcija. Brisanje dostavljača izvedeno je povlačenjem ćelije ulijevo što je uobičajeno ponašanje u iOS mobilnim sustavima, slika 5.15. Također, dostavljač ima uvid u sve radnje sa kojima surađuje, no on nema ovlasti za brisanje, te ne može povlačiti ćeliju ulijevo niti udesno.

```
func tableView(_ tableView: UITableView, commit editingStyle: UITableViewCell.EditingStyle, forRowAt indexPath: IndexPath) {
    if editingStyle == .delete {
        let courier = viewModel.myCouriers[indexPath.row]
        viewModel.deleteCourier(with: courier.id)
    }
}
```

Slika 5.15 Prikaz postavljanja brisanja pomoću povlačenja ćelije.

5.6.3. Upravljanje skočnim obavijestima

Upravljanje skočnim obavijestima je naizgled jednostavna funkcionalnost, no iza nje stoji puno logike i provjera. Pri prvom ulasku u aplikaciju, nakon prijave, korisniku je postavljen upit o tome želi li primati skočne obavijesti. Njegov odabir se zatim sprema u zadane postavke mobitela. Ukoliko je odabrao da želi primati skočne obavijesti, na server se šalje jedinstveni broj mobilnog uređaja pomoću kojeg se zatim šalju skočne obavijesti. Korisnik u svakom trenutku može promijeniti mišljenje, te ugasiti skočne obavijesti unutar aplikacije klikom na prekidač unutar postavki. Na server se šalje zahtjev za prekidanjem slanja skočnih obavijesti, te ukoliko se on uspješno izvrši, novi odabir se opet sprema u zadane postavke mobitela. Na svakom sljedećem ulasku u aplikaciju, vrši se provjera vidljiva na slici 5.16., koja odlučuje hoće li se na server slati jedinstveni broj mobilnog uređaja ili neće. Prvo se provjerava ima li aplikacija dopuštenje od mobilnog uređaja da prima skočne obavijesti i ukoliko nema, automatski izlazi iz funkcije. Ako ima, slijedi druga provjera, a to je postoji li korisnik prijavljen u aplikaciju i je odabir koji je spremljen u zadanim postavkama mobitela taj da želi primati skočne obavijesti. Ukoliko sve te provjere prođu, dohvaća se jedinstveni broj mobilnog uređaja i šalje na server. Postoji par krajnjih slučajeva koji su morali biti pokriveni kako bi upravljanje skočnim obavijestima ispravno radilo. Korisnik može otići u opće postavke mobilnog uređaja, te na postavkama aplikacije ugasi

dopuštenje za primanje skočnih obavijesti. Nakon što to napravi, jedinstveni broj se više neće slati na server. Ako uđe u aplikaciju, te u postavkama pokuša upaliti skočne obavijesti pomoću prekidača, biti će preusmjeren u opće postavke mobilnog uređaja kako bi dopustio primanje skočnih obavijesti, jer je to nužan uvjet rad skočnih obavijesti. Nakon što dopusti primanje, može kliknuti na prekidač i ponovno započeti primanje skočnih obavijesti.

```
func application(_ application: UIApplication, didRegisterForRemoteNotificationsWithDeviceToken deviceToken: Data) {
    UNUserNotificationCenter.current().getNotificationSettings { settings in
        guard settings.authorizationStatus == .authorized else { return }
        if IDManager.shared.loggedUser != nil && IDNotificationManager.loadPushNotificationsStatusFromDefaults() {
            let tokenParts = deviceToken.map { data in String(format: "%02.2hhx", data) }
            let token = tokenParts.joined()
            IDNotificationManager.registerPushNotifications(deviceToken: token)
        }
    }
}
```

Slika 5.16 Provjera za registraciju skočnih obavijesti.

Nakon što je aplikacija spremna na primanje skočnih obavijesti, preostaje samo napisati funkciju koja će upravljati rukovanjem sa pristiglim obavijestima. Svaka obavijest dolazi u JSON formatu, te posjeduje parametar „type“ koji otkriva o kojoj se vrsti obavijesti radi. Kako bi se lakše rukovalo njima, stvorena je enumeracija vrsta obavijesti koja se inicijalizira pomoću teksta koji se pročita iz parametara same obavijesti. Kako bi bilo moguće presresti dolazak obavijesti, te odlučiti što sa njom učiniti, potrebno je implementirati dvije funkcije iz „“UNUserNotificationCenterDelegate“. Jedna odlučuje kako će obavijest biti prikazana korisniku, a druga što će se dogoditi nakon što korisnik klikne na nju, slika 5.17.

```
func userNotificationCenter(_ center: UNUserNotificationCenter, didReceive response:
    UNNotificationResponse, withCompletionHandler completionHandler: @escaping () -> Void) {
    // called when you tap on the push notification
    let json = JSON(response.notification.request.content.userInfo)
    let notification: WGPushNotification = WGPushNotification(with: json)
    WGNotificationManager.handlePushNotificationTap(notification)
}
```

Slika 5.17 Prikaz funkcije za upravljanje klikom na obavijest.

5.6.4. Odjava

Funkcionalnost odjave je vrlo jednostavna. Nakon klika na gumb odjave, u „singletonu“ gdje se čuvaju podatci o prijavljenom korisniku, ti podatci se brišu iz njega. Nakon brisanja iz singletona, potrebno je te podatke izbrisati i iz zadanih postavki mobitela. Tamo su podatci o korisniku spremljeni pri svakoj prijavi i svakoj odjavi pomoću funkcija na slici 5.18.. To nam koristi kako bi ukoliko je korisnik prijavljen i u potpunosti ugasi aplikaciju, nakon sljedećeg ulaska bio

ponovno automatski prijavljen. Da ne postoje ti spremljeni podatci, korisnik bi se morao iznova prijavljivati nakon svakog gašenja aplikacije.

```
final func saveUserToUserDefaults() {
    if let user = self.loggedUser {
        let encoder = JSONEncoder()
        do {
            let encodedData = try encoder.encode(user)
            WGUserDefaults.userDefaults.set(encodedData, forKey: WGConstants.loggedUser)
            WGUserDefaults.userDefaults.synchronize()
        } catch {
            assertionFailure(error.localizedDescription)
        }
    }
}

final func removeUserFromUserDefaults() {
    WGUserDefaults.userDefaults.removeObject(forKey: WGConstants.loggedUser)
    WGUserDefaults.userDefaults.synchronize()
}
```

Slika 5.18. Funkcija za spremanje i brisanje korisnika.

6. PRIKAZ RADA APLIKACIJE

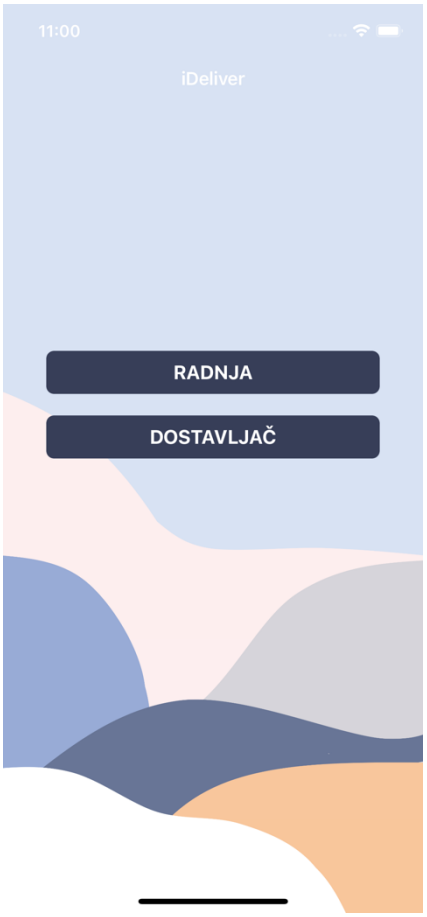
Ovo poglavlje detaljno će opisati funkcionalnosti, korištenje i korake kroz koje korisnici moraju proći da bi mogli u potpunosti iskoristiti prednosti aplikacije.

6.1. Prikaz omogućenih funkcionalnosti aplikacije

Prvi korak je naravno instalacija aplikacije na mobilni uređaj. Nakon instalacije slijedi registracija novog računa. Ako je račun uspješno stvoren, aplikacija odlazi na zaslon prijave gdje se korisnik, koristeći iste podatke kao i pri registraciji, prijavljuje u aplikaciju. Početni zaslon nakon prijave je zaslon koji prikazuje listu dostava. Za obje vrste korisnika ta lista će na početku biti prazna. Kako bi ju popunili, radnja mora stvoriti novu dostavu. Kako bi stvaranje dostave bilo moguće, radnja mora imati barem jednog dostavljača s kojim surađuje, pa je tako prvi korak nakon prijave, pozivanje dostavljača na suradnju. Nakon uspješnog povezivanja, radnja može uspješno stvoriti dostavu i dodijeliti je jednom ili više dostavljača sa kojima surađuje. Nakon stvaranja dostave, ona je prikazana na listi radnje koja ju je stvorila i na listi dostavljača koji su joj dodijeljeni. Dostavljači zatim imaju mogućnost prihvaćanja dostave, što ih obavezuje da će istu dostaviti na adresu krajnjeg kupca. Iako postoji mogućnost otkazivanja dostave nakon prihvaćanja, ona loše utječe na reputaciju dostavljača. U svakom trenutku dostavljač i radnja mogu pregledati listu korisnika sa kojima surađuju, a radnja ima dodatnu mogućnost da tu listu uređuje. Dostavljači i radnje su o promjenama o statusu dostave obaviješteni skočnim obavijestima, koje u svakom trenutku po želji mogu isključiti. Aplikacija je trenutno dvojezična te podržava Hrvatski i Engleski prijevod.

6.2. Prikaz korištenja aplikacije

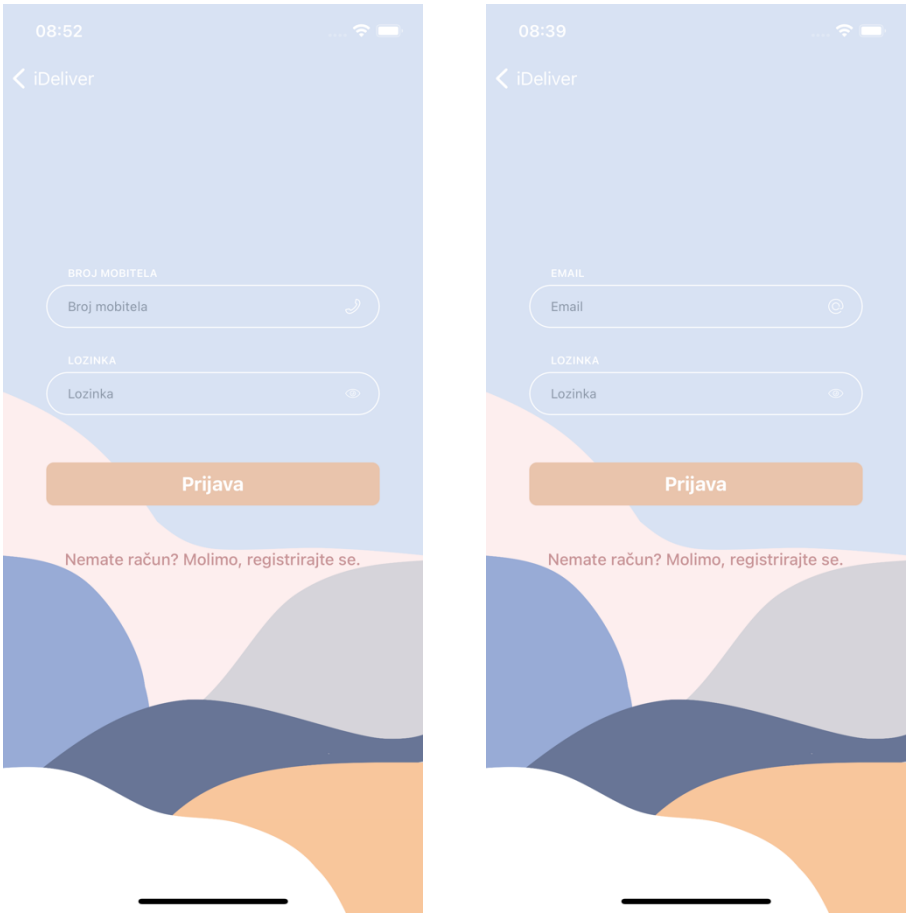
Pri prvom ulasku u aplikaciju, korisniku se otvara početni zaslon na kojem ima mogućnost biranja vrste korisnika. Korisnik zatim može kliknuti ili na vrstu dostavljača ili na vrstu radnje, vidljivo na slici 6.1.



Slika 6.1 Prikaz početnog zaslona aplikacije.

6.2.1. Prikaz prijave

Nakon odabira bilo koje vrste korisnika, sljedeći zaslon je zaslon prijave sa slike 6.2. Svaki korisnik prijavljuje se sa dva identifikacijska parametra. Za dostavljača to je broj mobilnog uređaja i lozinka, dok se kod radnje umjesto broja mobilnog uređaja koristi adresa elektroničke pošte. Nakon što su parametri uneseni, korisnik ih pritiskom na gumb „Prijava“ šalje na server. Ukoliko postoji registriran račun sa tim parametrima, aplikacija sprema korisnika i vodi ga na zaslon prikaza liste dostava. Ako je korisnik unio pogrešne podatke, te server ne pronalazi registrirani račun sa danim podatcima, biti će upozoren povratnom porukom vidljivom na vrhu ekrana. Koristi li korisnik aplikaciju po prvi puta, te još uvijek nema registriran račun, klikom na gumb pri dnu ekrana, odlazi na zaslon registracije.

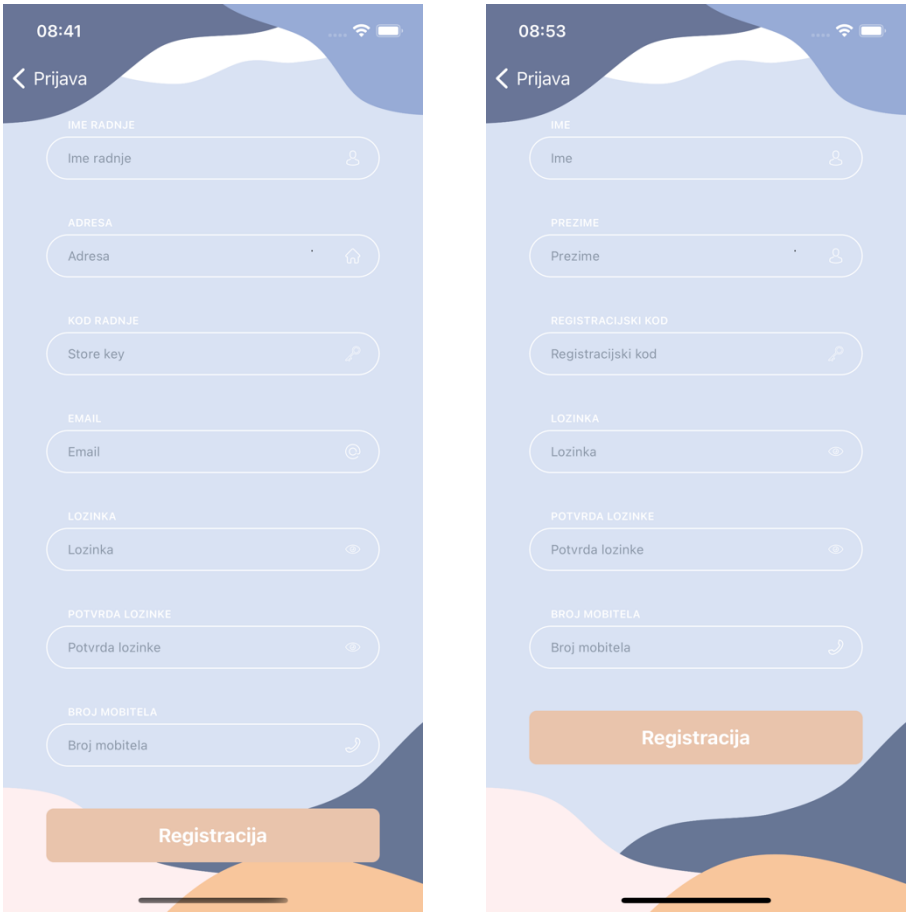


Slika 6.2 Prikaz zaslona prijave dostavljača i radnje.

6.2.2. Prikaz registracije

Zaslon registracije dostavljača i radnje razlikuje se u više polja za unos. Kako bi korisnik registrirao račun dostavljača, potrebno je unijeti njegovo ime i prezime. Nakon toga potrebno je unijeti novu lozinku, koju korisnik mora ponoviti dva puta, kako bi se izbjegla slučajna pogreška pri unosu. Lozinka mora biti duža od šest znakova, a ukoliko ne ispunjava taj uvjet, korisnik će biti upozoren crvenim tekstom ispod polja za unos, te neće moći kliknuti na gumb „Registracija“. Nakon lozinke, potreban je unos jedinstvenog broja mobilnog uređaja, koji mora biti upisan u ispravnom formatu. Podržani su formati +385XXXXXXXXX i 09XXXXXXXX. To su sva obavezna polja za registraciju, no postoji i jedno neobavezno polje koje korisnik ispunjava ovisno o tome ima li već registracijski kod od neke radnje. Ukoliko ima, nakon registracije automatski je povezan s tom radnjom. Nakon ispune podataka, klikom na gumb „Registracija“, oni se šalju na server i ukoliko već ne postoji korisnik sa poslanim brojem mobilnog uređaja, stvara se novi račun. Nakon uspješne registracije, korisnik je automatski vraćen na zaslon prijave. Ukoliko već postoji račun sa unesenim brojem mobilnog uređaja ili je došlo do neke druge pogreške te registracija nije

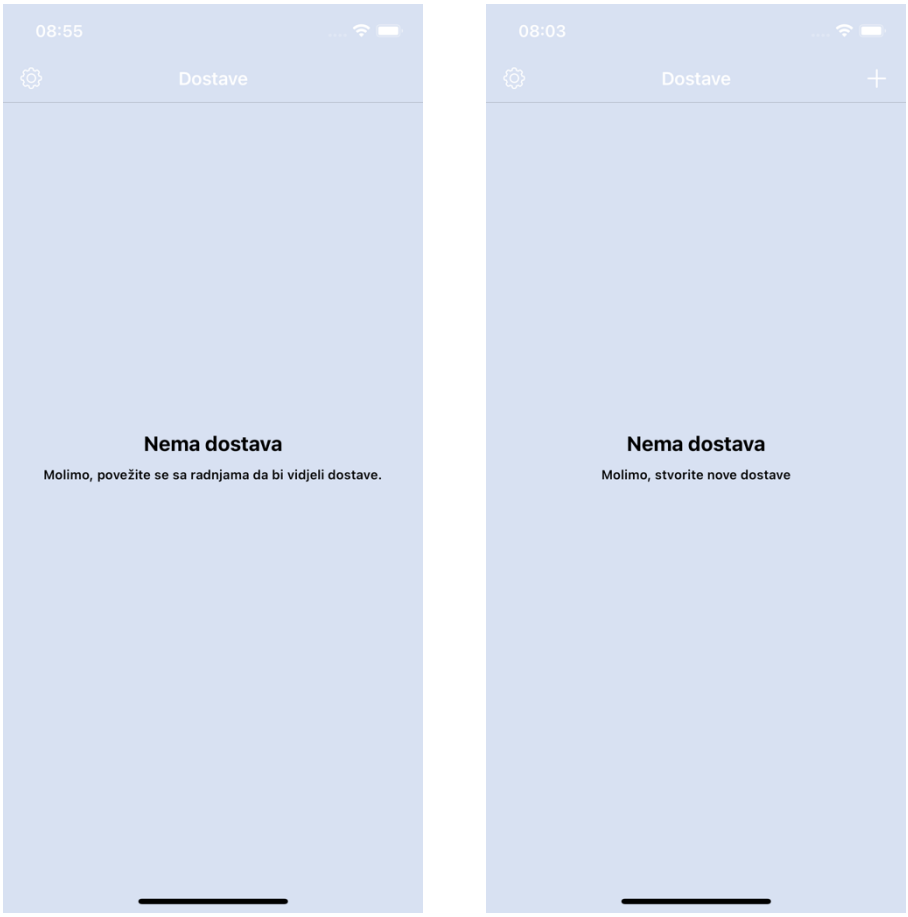
ispravno izvršena, korisnik je obaviješten preko poruke na vrhu ekrana. Isto vrijedi i za registraciju radnje, samo što se umjesto imena i prezimena unose ime radnje, adresa, jedinstveni kod radnje, te jedinstvena adresa elektroničke pošte. Jedinstveni kod radnje može se poistovjetiti sa osobnim identifikacijskim brojem neke osobe. Oba zaslona za registraciju su vidljiva na slici 6.3.



Slika 6.3. Prikaz zaslona registracije radnje i dostavljača.

6.2.3. Pregled i stvaranje dostava

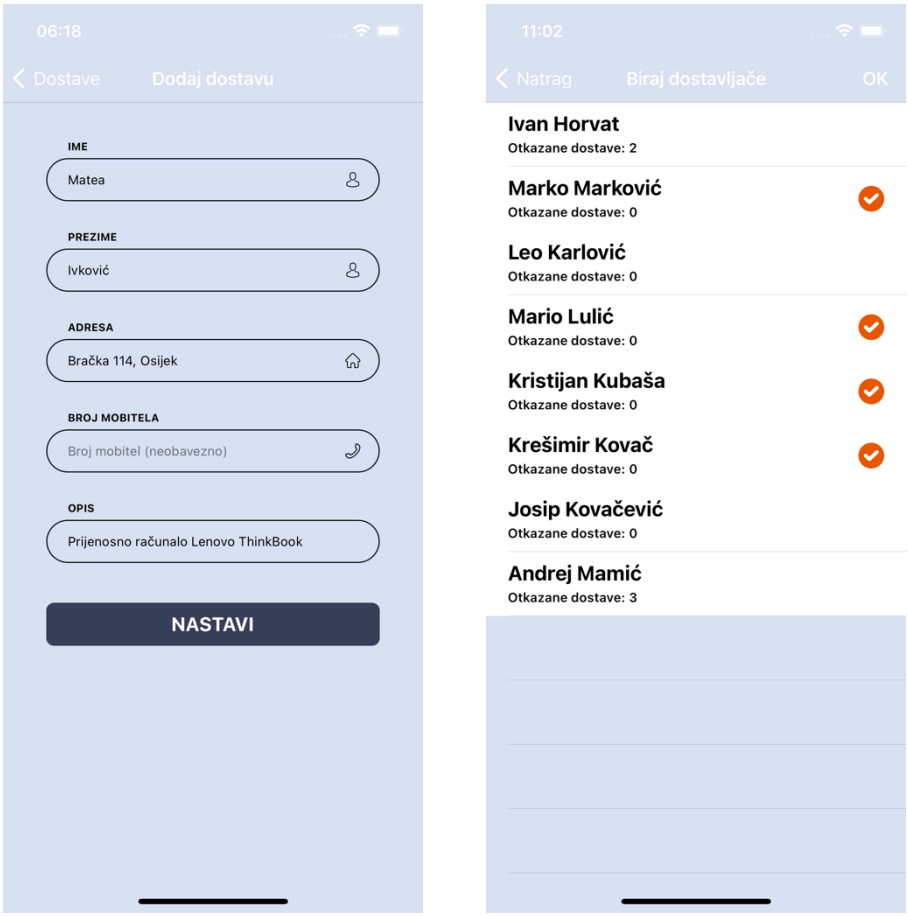
Nakon prijave korisnik je odveden na zaslon pregleda liste dostava. Svaki korisnik će isprva imati praznu listu, te će mu biti prikazana poruka „praznog stanja“, slika 6.4. Radnja neće imati dostava jer ih prvo mora stvoriti, a dostavljač iz razloga što mora biti dodijeljen na neku od njih.



Slika 6.4. Prikaz praznog stanja liste dostava.

Kako lista više ne bi bila prazna, korisnik može stvoriti novu dostavu klikom na gumb u gornjem desnom kutu zaslona. Nakon klika, otvara se forma, vidljiva na slici 6.5., koju je potrebno ispuniti kako bi se omogućio nastavak stvaranja dostave. Podatci koji su obavezni za nastavak su ime i prezime klijenta, tj. osobi kojoj se dostavlja paket, adresa na koju paket mora biti dostavljen i kratak opis paketa. U opis je zamišljeno da se upišu neki podatci kao na primjer sadržaj paketa, okvirna veličina, okvirna cijena te neke specifičnosti paketa. Ukoliko paket sadrži vrlo lomljive stvari, ta informacija će jako koristiti dostavljaču kako bi uspješno bez štete mogao dostaviti paket. Postoji i neobavezni parametar, a to je broj mobilnog uređaja osobe kojoj se isporučuje paket. On bi bio od velike koristi ukoliko dođe do komplikacija pri dostavi, te bi preko njega dostavljač mogao kontaktirati klijenta radi lakšeg razvoja situacije. Klikom na gumb „Nastavi“ prikazuje se ekran biranja dostavljača. Prikazani su svi koji su povezani sa radnjom koja kreira dostavu, a ispod njihovog imena piše broj dostava koje su isprva prihvatili, no kasnije otkazali. Klikom na pojedinog dostavljača, on se odabire kao kandidat za isporuku dostave, što je naznačeno kvačicom na desnoj strani ćelije. Nakon što je odabran barem jedan dostavljač, moguće je kliknuti na gumb

„OK“ u gornjem desnom uglu te se zahtjev za stvaranjem dostave šalje na server. Ukoliko je uspješno stvorena, aplikacija se vraća na zaslon liste dostava, gdje je pri vrhu prikazana najnovije stvorena dostava. Ukoliko dođe do greške, povratna poruka prikazuje se na vrhu ekrana.



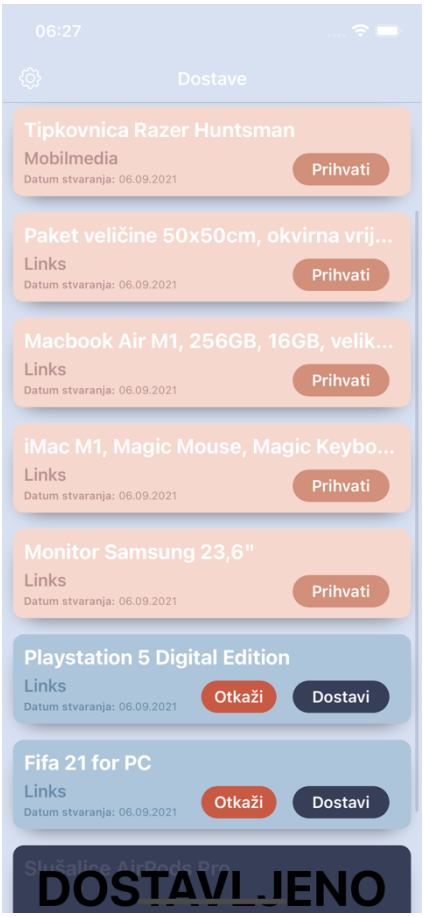
Slika 6.5. Prikaz zaslona stvaranja dostave i biranja dostavljača.

6.2.4. Prikaz obavljanja akcija nad dostavom

Nakon stvaranja dostava, one su na listi vidljive i radnji koje su ih stvorile i svim dostavljačima koji su na njih dodijeljeni. Ovisno o vrsti, svaki korisnik može imati neku vrstu interakcije sa svakom dostavom. Tako svaka radnja može otkazati bilo koju dostavu klikom na gumb „Otkaži“ u desnom dijelu ćelije. Dostava može biti otkazana neovisno je li u „stanju čekanja“ ili „prihvaćena“. Ukoliko je prihvaćena, dostavljač koji ju je prihvatio biti će istog trenutka obaviješten skočnom obavijesti o otkazivanju, te mu više neće biti vidljiva. Otkazana dostava biti će vidljiva samo radnji, te će se nalaziti na dnu liste obojana crvenkastom bojom, dok će dostava u tijeku biti obojana plavkasto, vidljivo na slici 6.6.

Slika 6.6. Prikaz liste dostava viđene od strane radnje.

Prikaz dostava korisniku koji je dostavljač izgleda slično, no on ima različitu mogućnost interakcije s njima. Nakon što bude odabran kao mogući kandidat za dostavu, stiže mu skočna obavijest. Klikom na nju, neovisno na kojem zaslonu se nalazi i nalazi li se uopće u aplikaciji, otvara mu se lista dostava sa svježe dohvaćenim dostavama. Dostave se također mogu osvježiti povlačenjem tablice prema dolje. Na početku ima jednu mogućnost, a to je prihvatiti dostavu ukoliko mu ona odgovara. Nakon što ju prihvati, ona mijenja obojanost u plavkastu boju, premješta se ispod dostava koje su „na čekanju“, te ima nove dvije mogućnosti. Ukoliko dostavljač u potpunosti obavi posao, tj. dostavi paket na željenu adresu, može kliknuti na gumb „Dostavi“. Tako završava tu dostavu i ona se premješta na kraj liste i vidljivo je označena sa „DOSTAVLJENO“, slika 6.7. Ukoliko dostavljač prihvati dostavu, no nakon toga iz bilo kojeg razloga više ne želi izvršiti tu istu dostavu, može kliknuti na gumb „Otkaži“. Nakon što otkaže dostavu, povećava mu se brojač otkazanih dostava, te bi to moglo loše utjecati na njegovu reputaciju, a samim time bi ga manje radnji odabiralo za suradnju.



Slika 6.7. Prikaz liste dostava od strane dostavljača.

6.2.5. Prikaz detalja dostave

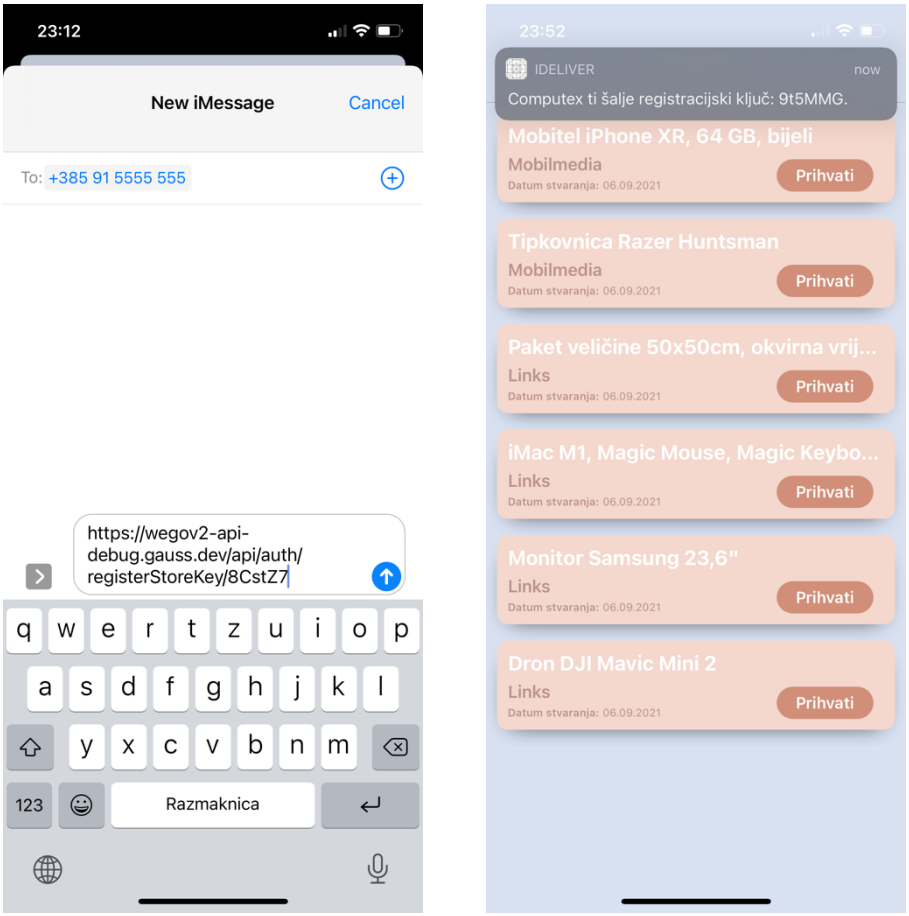
Klikom na svaku pojedinu dostavu obje vrste korisnika ima mogućnost detaljnog prikaza svih parametara dostave. U detaljnom prikazu sa slike 6.8., vidljiv je status u tekstualnom obliku, te nakon njega idu podatci o radnji koja je kreirala dostavu. Prikazano je njeno ime, adresa i broj mobilnog uređaja na koji dostavljač može kontaktirati radnju ukoliko je potrebno. Nakon tih podataka slijede podatci o klijentu kojem se paket treba dostaviti. Također je prikazano puno ime i prezime, adresa i broj mobilnog uređaja ukoliko je on unesen pri stvaranju dostave. Zadnji podatak je potpuni opis dostave, pošto je u ćeliji na listi dostava prikazan samo jedan redak opisa zbog boljeg vizualnog doživljaja. Ispod prikaza svih parametara stoje gumbovi za interakciju sa dostavom, kao što je i u prikazu na listi.

Slika 6.8. Prikaz detalja dostave.

6.2.6. Prikaz povezivanja dostavljača i radnje

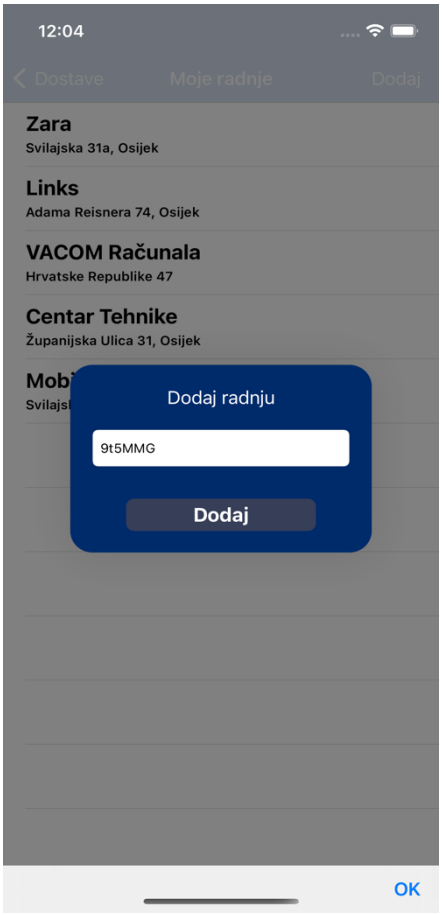
Nakon što dostavljač registrira račun, sljedeće što mora napraviti je povezati se sa radnjom, a to može jedino tako da ga radnja pozove na suradnju. Radnja ima uvid na sve povezane dostavljače koji su prikazani u listi. Do te liste se dolazi klikom na ikonicu kotačića u gornjem lijevom kutu, te nakon toga klikom na „Moji dostavljači“. Na zaslonu koji se otvori nalazi se lista dostavljača, a u gornjem desnom kutu znak plusa. Klikom na njega otvara se zaslon koji prikazuje sve dostavljače koji imaju registriran korisnički račun, ali još uvijek nisu povezani sa trenutnom radnjom. Klikom na pojedinog dostavljača, šalje se zahtjev na server za kreiranjem koda koji se koristi za povezivanje. Ukoliko se kod uspješno kreira, poslan je u odgovoru servera, te se na određeni način dostavlja dostavljaču. Ukoliko onaj za koga je kreiran kod, trenutno nije prijavljen u aplikaciju, radnji će se automatski otvoriti aktivnost za pisanje SMS poruke. Za primatelja poruke biti će već ispunjen broj mobilnog uređaja dostavljača, a u tijelu poruke će biti univerzalna veza koja sadrži kod za povezivanje. Radnja po izboru može dodatno urediti tijelo poruke, dodati proizvoljan tekst, ali najbitnije je da univerzalna veza ostane nepromijenjena. Drugi način za isporuku registracijsko

koda je preko skočne obavijesti. Ukoliko je dostavljač prijavljen u aplikaciju, automatski nakon zahtjeva za kreiranjem koda, stići će mu obavijest koja sadrži kod. Prikaz oba načina vidljiv je na slici 6.9.



Slika 6.9. Prikaz aktivnosti za pisanje SMS poruke i skočne obavijesti.

Postoji i mogućnost da radnja na suradnju pozove nekoga tko još nema registriran račun u aplikaciji. Dovoljno je da ima broj mobilnog uređaja, koji klikom na gumb „Ručno“ u gornjem desnom uglu, može unijeti u polje, te poslati registracijski kod u SMS poruci. Nakon što dostavljač, bio registriran ili ne, zaprimi registracijski kod, na njemu je da prihvati ili ne prihvati poziv radnje. Ukoliko je registriran, dostavljač će klikom na skočnu obavijest ili na univerzalnu vezu iz SMS poruke, biti odveden na zaslon unosa registracijskog koda, kao na slici 6.10. On će već biti upisan u polje, te je na dostavljaču samo da potvrdi unos. Nakon potvrde, šalje se zahtjev za povezivanjem na server, te ukoliko sve prođe bez pogreške, dostavljač i radnja su spremni za suradnju. Ako dostavljač nije registriran, klikom na univerzalnu vezu u SMS poruci, otvara mu se aplikacija na zaslonu za registraciju, gdje je već ispunjen registracijski kod. Zatim dostavljač ispunjava ostatak svojih podataka i registrira novi račun.



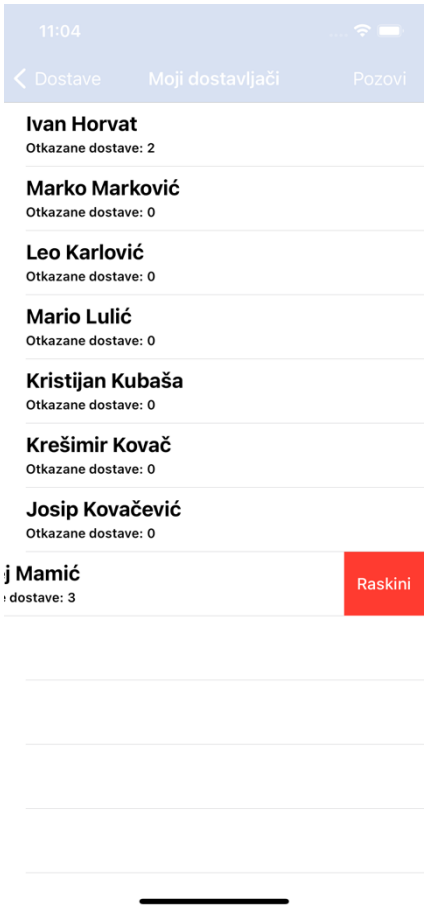
Slika 6.10. Prikaz dodavanja radnje.

6.2.7. Prikaz dodatnih postavki

Kada se korisnik nalazi na početnom zaslonu liste dostava, u gornjem lijevom uglu nalazi se ikonica kotačića. Klikom na nju na ekranu, uz prikladnu animaciju, prikazuje se popis dodatnih postavki koje korisnik može koristiti, slika 6.11.

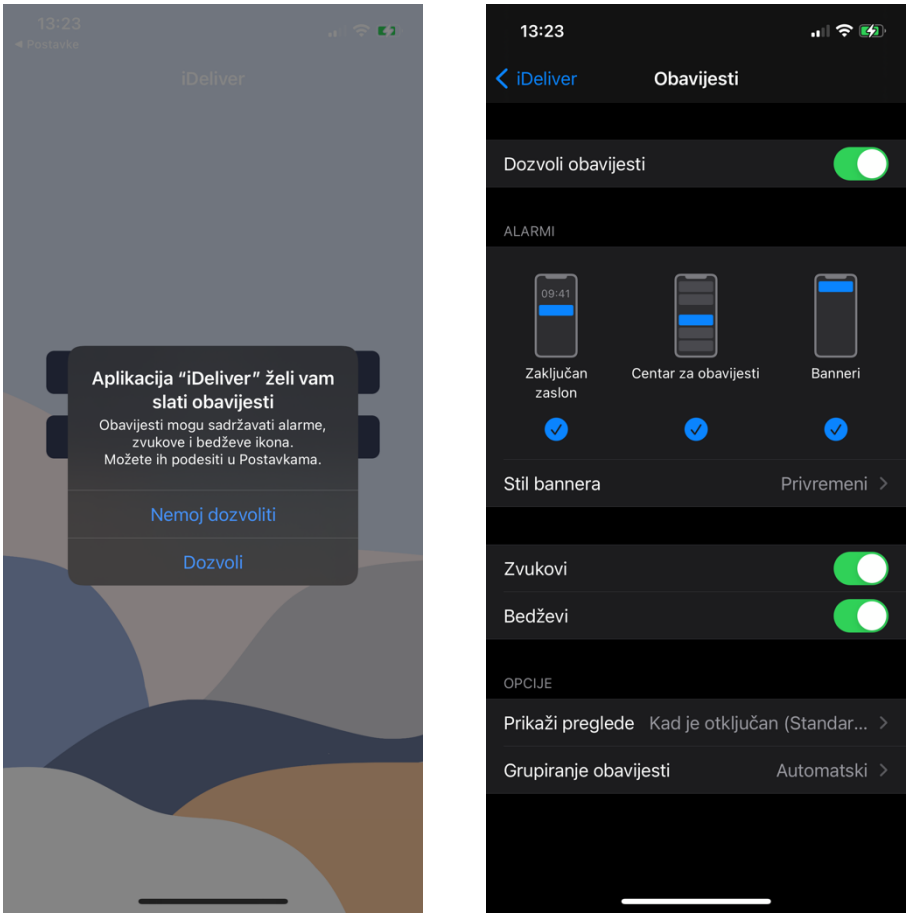
Slika 6.11. Prikaz dodatnih postavki aplikacije i popisa podržanih jezika.

Prva od postavki je postavka mijenjanja jezika aplikacije. Korisnik vidi ćeliju u kojoj se nalazi slika trenutno odabranog jezika. Klikom na tu ćeliju, otvara mu se lista svih jezika podržanih od strane aplikacije. Korisnik zatim može odabrati neki od ponuđenih jezika, nakon čega mu se lista i ekran dodatnih postavki zatvaraju, a aplikacija je nadalje prikazana prevedena na odabrani jezik. Druga postavka je omogućen pregled trenutnih suradnika, u slučaju prijavljenog dostavljača to su radnje i obrnuto. Radnja odlaskom na taj zaslon ima mogućnost prekida suradnje sa određenim dostavljačem. To može napraviti na način da povuče ćeliju ulijevo, na što će se prikazati gumb „Raskini“, prikazan na slici 6.12. Klikom na njega, prekida se svaka suradnja između radnje i dostavljača, što znači da dostavljač više nije u mogućnosti vidjeti dostave napravljene od strane te radnje.



Slika 6.12. Prikaz raskidanja suradnje sa dostavljačem.

Treća postavka je paljenje i gašenje skočnih obavijesti. Pri prvom ulasku u aplikaciju, korisnik dobije upit želi li dozvoliti skočne obavijesti u ovoj aplikaciji. Ako korisnik dozvoli, one će automatski biti upaljene, te sve dok je postavka dozvole upaljena, korisnik može paliti i gasiti skočne obavijesti unutar aplikacije. Ukoliko iz nekog razloga, u postavkama mobilnog uređaja ugasi dozvolu za primanje skočnih obavijesti, skočne obavijesti unutar aplikacije će odmah biti ugašene. Pri pokušaju paljenja obavijesti unutar aplikacije, u slučaju ugašene dozvole, korisnik će biti preusmjeren u opće postavke mobilnog uređaja, kako bi upalio dozvolu. Tek nakon paljenja dozvole moguće je upaliti skočne obavijesti. Početni upit i postavke o obavijestima vidljive su na slici 6.13.



Slika 6.13. Prikaz upita za dozvolu, te postavki dozvole mobilnog uređaja.

Zadnja postavka je akcija odjave korisnika. Klikom na nju korisnik briše sve podatke o sebi iz aplikacije te se vraća na početni zaslon aplikacije.

7. ZAKLJUČAK

U ovom diplomskom radu izrađena je mobilna aplikacija za platformu iOS koja služi za olakšanu interakciju između dostavljača i radnji pri dostavljanju proizvoda. Korištenje aplikacije korisniku omogućuje registraciju i prijavu, te upravljanje i interakciju sa dostavama stvorenim od strane radnji. Također, aplikacija nudi suradnju različitih korisnika. Ovaj sustav mogao bi pomoći u poslovanju onim radnjama koje već imaju stvorenu internet kupovinu sa sustavom plaćanja, no nedostaje im faza dostave proizvoda ili nisu zadovoljni sa trenutnim odabirom načina dostave.

Aplikacija je razvijena u razvojnoj okolini Xcode, a napisana je u popularnom programskom jeziku Swift, verzije 4. Aplikacija je pri razvoju testirana i pokretana na najnovijem iOS sustavu verzije 14, a ispitivanje njenih funkcionalnosti i komponenti pokazuje ispravan rad aplikacije. Ispitivanjem aplikacije dovelo je do zaključka kako se aplikacija može još puno poboljšati uvođenjem dodatnih mogućnosti kao na primjer, tekstualni razgovor između dostavljača i radnje, uvođenje klijenata u aplikaciju kako bi mogli pratiti svoju dostavu, opširniji i detaljniji opis dostava. Ova aplikacije može poslužiti kao dobar koncept i temelj za nadogradnju kako bi bila još korisnija pri upotrebi.

LITERATURA

[1] Glovo App, https://glovoapp.com/, pristupljeno: 02.05.2021.

[2] Pauza, https://www.pauza.hr, pristupljeno: 02.05.2021.

[3] Wolt, https://wolt.com, pristupljeno: 02.05.2021.

[4] Xcode, https://developer.apple.com/xcode/, pristupljeno: 05.05.2021.

[5] Swift programming language, https://developer.apple.com/swift/, pristupljeno: 05.05.2021.

[6] CocoaPods, https://cocoapods.org, pristupljeno: 05.05.2021.

[7] Abhilash Mathur, MVVM in OS Swift, 9 July 2020., https://medium.com/flawless-app-stories/mvvm-in-ios-swift-aa1448a66fb4, pristupljeno: 06.05.2021.

[8] TableView, https://developer.apple.com/documentation/uikit/uitableview, pristupljeno: 10.05.2021.

[9] User Notifications, https://developer.apple.com/documentation/usernotifications, pristupljeno: 14.05.2021.

[10] Universal Links, https://developer.apple.com/ios/universal-links/, pristupljeno: 16.05.2021.

[11] Localize Swift, https://github.com/marmelroy/Localize-Swift, pristupljeno: 17.05.2021.

SAŽETAK

U ovom diplomskom radu razvijena je aplikacija za interakciju i povezivanje radnji i dostavljača u cilju olakšavanja dostave paketa. Korisnici nakon registracije i prijave imaju mogućnost osnivanja suradnje na temelju povjerenja, te stvaranja i interakcija sa poslovima dostave paketa raznih veličina. Analizom aplikacije dolazi se do zaključka da postoji još puno prostora za napredak, ali da je postavljen stabilan temelj za uspješnu mobilnu aplikaciju.

Ključne riječi: dostava, dostavljač, iOS, mobilna aplikacija, radnja

ABSTRACT

Title: **iDeliver**

This final work evolves an application that is used for interaction and connection between stores and couriers in order to make package delivering easier. After registering and loging in, users have possibility to create trust based cooperation, and creating and interacting with jobs of delivering packages of various sizes. The application analysis indicates that there is a lot space for improvement, but that foundation for successfull mobile application is set.

Key words: courier, delivery, iOS, mobile application, store

ŽIVOTOPIS

Zvonimir Pavlović rođen je 07.11.1997. godine u Našicama. Pohađao je Osnovnu školu Ivane Brlić Mažuranić u Koški. Upisuje III. Gimnaziju u Osijeku 2012. godine koju uspješno završava 2016. Iste godine upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku, preddiplomski studij računarstva. 2019. godine završava preddiplomski studij, te upisuje diplomski studij Programskog inženjerstva.

_____

Potpis autora