

Aplikacija za koordinaciju procesa izrade pekarskih proizvoda

Hasanagić, Sean

Master's thesis / Diplomski rad

2019

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

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:137:514773>

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

Download date / Datum preuzimanja: **2025-02-01**



Repository / Repozitorij:

[Digital Repository Juraj Dobrila University of Pula](#)



Sveučilište Jurja Dobrile u Puli

Fakultet infomatike

SEAN HASANAGIĆ

**APLIKACIJA ZA KOORDINACIJU PROCESA IZRADE
PEKARSKIH PROIZVODA**

Diplomski rad

Pula, 18. lipnja 2019. godine

Sveučilište Jurja Dobrile u Puli
Fakultet infomatike

SEAN HASANAGIĆ

**APLIKACIJA ZA KOORDINACIJU PROCESA IZRADA
PEKARSKIH PROIZVODA**

Diplomski rad

JMBAG: 0303046692, redoviti student

Studijski smjer: Diplomski studij informatike

Predmet: Izrada informatičkog projekta

Znanstveno područje: Društvene znanosti

Znanstveno polje: Informacijske i komunikacijske znanosti

Znanstvena grana: Informatički sustavi i informatologija

Mentor: doc. dr. sc. Siniša Sovilj

Pula, 18. lipnja 2019. godine



IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani Sean Hasanagić, kandidat za magistra informatike ovime izjavljujem da je ovaj Završni rad rezultat isključivo mogega 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

Pula, 18. lipnja 2019. godine



IZJAVA

o korištenju autorskog djela

Ja, Sean Hasanagić dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj završni rad pod nazivom „Aplikacija za koordinaciju procesa izrade pekarskih proizvoda“ 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, 18. lipnja 2019. godine

Potpis

Pula, 1. ožujka 2018.

DIPLOMSKI ZADATAK

Pristupnik **Sean Hasanagić (0303046692)**

Studij: Sveučilišni diplomski studij Informatike

Naslov (hrv.): Aplikacija za koordinaciju procesa izrade pekarskih proizvoda.

Naslov (eng.): Application for coordination of bakery production processes.

Opis zadatka: Realizirati sustav namjnen koordinaciji procesa izrade pekarskih proizvoda (točan izračun velikog broja zamjesa različitih receptura i namirnica) radi smanjena pogrešaka, kašnjenja i škartiranja proizvoda. Sustav se treba sastojati od: 1) mobilne/web aplikacije voditelja poslovnice za dodavanje i mijenjanje recepata (modul recepture), uvid u stanje skladišta (modul skladiša) i izvršavanje narudžbi (modul naloga) te 2) mobilne (klijentske) aplikacije namjenjene procesiranju naloga. Predviđene tehnologije za izradu aplikacije su: Firebase baza (*backend*) i Android (*frontend*).
U istraživačkome dijelu samoga rada potrebno je napraviti usporedbu između Firebase baze podataka i drugih alternativnih baza podataka te načina pohrane te izvesti zaključak o svim prednostima i manama analiziranih rješenja.

Zadatak uručen pristupniku: 1. ožujka 2018.

Rok za predaju rada: 1. veljače 2019.

Mentor:



doc.dr.sc. Siniša Sovilj

SADRŽAJ

Uvod.....	7
Firestore real time database.....	8
Pravila pristupa Firestore bazi podataka u razvoju.....	10
Postavljanje pravila i indeksiranje podataka u Firestore bazi podataka.....	12
JSON stablo (eng. JSON Tree).....	14
Cijene i funkcionalnosti paketa Firestore baze podataka.....	15
Spajanje s Firestore bazom podataka.....	16
Cloud Firestore.....	19
Ključne mogućnosti.....	19
Usporedba Firestore database i Cloud Firestore.....	21
Relacijske baze podataka.....	25
SQL (eng. Structured Query Language).....	26
Usporeba Firestore baze podataka i relacijske baze podataka.....	28
Opis dosadašnjeg procesa.....	31
Korisnički zahtjevi.....	33
Opis informatiziranog procesa vođenja pekarskog poslovanja.....	34
Funkcionalnosti aplikacije.....	35
Korištenje tehnologije i programski alati.....	39
Android Studio.....	39
Android SDK.....	41
HTML (eng. HyperText markup Language).....	42
Stilski jezik (eng. Cascading Style Sheets CSS).....	43
Javascript.....	44
Bootstrap.....	44
Implementacija.....	45
Izrada baze podataka.....	46

Implementacija Firebase baze podataka s Web aplikacijom	48
Korisničke upute	49
Unos novog djelatnika	49
Unos novog artikla	53
Unos nove recepture	61
Unos novog naloga	67
Android aplikacija	70
Prijava korisnika	70
Pregled naloga	72
Zaključak	75
Literatura	76
Popis knjiga	76
Popis internetskih izvora	77
Popis slika	78
Popis tablica:	81
Sažetak	82
Summary	82

Uvod

Sve bržim razvojem tehnologije mobilnih i tabletnih uređaja te rapidnim razvojem u oblaku baze podataka, poslovno informacijski sustavi (PIS) postaju sve pristupačniji i potrebniiji manjim i srednjim poduzećima te obrtima. Poslovno informacijski sustavi prilagođeni određenoj vrsti poslovanja postaju sve popularniji u našem okruženju, ponajviše zbog povećanja konkurencije aplikacijskih rješenja što je direktno dovelo do smanjenja cijene za izradu prilagođenog poslovnog informacijskog sustava za naručioce sustava. Zbog dostupnosti informacija u stvarnom vremenu naručiocu donosi određene uštede bilo to ljudskih, vremenaskih, materijalnih te financijskih resursa. Rad se sastoji od pet poglavlja. U prvom poglavlju analizirati će se *Firebase* baza podataka u realnom vremenu, način implementacije, navesti primjer zapisa podataka u JSON stablo, te prikazati način definiranja pravila za pristup podacima u *Firebase* bazi podataka. Usporediti će se novo rješenje baze podataka u oblaku odnosno *Cloud Firestore*, razvijeno od strane Google-a koje je trenutno u Beta verziji i sve više korištena *Firebase* baza podataka. Drugo poglavlje opisuje relacijske baze podataka, te analizira razlike između standardne baze podataka *SQL* i baze podataka vođene dokumentom. Naredno poglavlje opisuje dosadašnju praksu u pekarskoj proizvodnoj liniji. Uz dobivene korisničke zahtjeve projektirana i implementirana je arhitektura sustava koji pomaže administraciji pekarske proizvodne linije tako i njenim djelatnicima. U navedenom poglavlju detaljno će se prikazati način implementacije *Firebase* baze podataka u stvarnom vremenu u web aplikaciji, gdje će se provoditi definiranje i postavljanje svih parametara cjelokupnog poslovanja kao što su: definiranje artikla, definiranje uposlenika, definiranje recepta te izdavanje naloga. Iduće poglavlje opisuje sve korištene alate i tehnologije za izradu praktičnog dijela diplomskog rada. Prikazan je sam način implementacije *Firebase* baze podataka u *Android Studiju* i u web aplikaciji. U slijedećem poglavlju opisane su sve funkcionalnosti praktičnog dijela rada odnosno internetske aplikacije i mobilne aplikacije te korisnička dokumentacija.

Firestore real time database

Firestore baza podataka u realnom vremenu integrirana je u oblaku. Podaci se pohranjuju kao *JSON* i sinkroniziraju u realnom vremenu sa svim povezanim klijentima, koji mogu biti implementirani na više platformi kao što su: *iOS*, *Android*, *JavaScript*, *Unity* i *C++*. Svaki od navedenih klijenata dijeli jednu instancu baze podataka zbog čega se u stvarnom vremenu automatski ažuriraju najnoviji podaci. Umjesto tipičnih *HTTP* zahtjeva, *Firestore* baza podataka u realnom vremenu koristi sinkronizaciju podataka. Svaki put kada se podaci promijene, svaki povezani uređaj neovisno o implementacijskom jeziku prima ažurirani podatak u nekoliko milisekundi. Aplikacijska rješenja implementirana s *Firestore* bazom podataka u realnom vremenu ostaju u funkciji dok nemaju konekciju s internetskom mrežom. Naime, unešeni podaci ili naredbe spremaju se na disk uređaja. Nakon ponovnog uspostavljanja veze klijentski uređaj prima sve promjene koje je propustio dok je bio u izvanmrežnom stanju sinkronizirajući ga s trenutnim stanjem poslužitelja. *Firestore* bazi podataka u realnom vremenu može se pristupiti izravno s mobilnog uređaja ili internetskog preglednika. Naime, nema potrebe za poslužiteljskim aplikacijama. Sigurnost i provjera valjanosti podataka dostupni su putem *Firestore* sigurnosnih pravila baze podataka u realnom vremenu. To su, dakle, pravila temeljena na izrazima koja se izvršavaju kada se podaci čitaju ili zapisuju. *Firestore* baza podataka u realnom vremenu licencirana je za komercijalnu upotrebu, ali nudi besplatni osnovni paket naziva „*Blaze*“ (plamen) koji može podržati većinu podatkovnih potreba aplikacije. Također, nudi provjeru autentičnosti pomoću *Firestore* identifikacije korisnika. *Firestore* baza podataka u realnom vremenu omogućuje izgradnju bogatih kolaborativnih aplikacija dopuštajući siguran pristup podataka izravno iz koda na strani klijenta. *Firestore* baza podataka omogućuje programerima fleksibilna sigurnosna pravila koje definira sam programer. Na taj način programeri mogu definirati kada se podaci mogu čitati ili pisati, a uz pomoć *Firestore* identifikacije korisnika mogu definirati tko ima pravo pristupa određenim podacima. *Firestore* baza podataka u stvarnom vremenu je nerelacijska baza podataka (eng. *NoSQL-Structured Query Language*) i kao takva ima različite optimizacije i funkcionalnosti u usporedbi sa relacijskom bazom podataka. *Firestore* baza podataka u realnom vremenu dizajnirana je za izvršavanje operacija koje se mogu brzo izvršiti. Takav dizajn omogućuje korištenje aplikacija milijunima korisnika bez ugrožavanja odaziva. Stoga je od velikog značaja koji korisnik ima pravo pristupa kojem podatku i kako, te shodno tome strukturirati podatke.

Firestore je skalabilna baza u stvarnom vremenu (eng. *Real time database*) osnovana 2012 godine te preuzeta od Google-a u 2014 godini. Do listopada 2018 godine *Firestore* ima osamnaest različitih proizvoda koji su korišteni od strane 1.5 miliona aplikacija. Za razliku od tradicionalnih baza podataka čiji je model relacijski, *Firestore* baza podataka u realnom vremenu modelirana je po sistemu pohranjivanja dokumenata (eng. Document store).

Pohranjivanje dokumenata naziva se još i dokumentima orijentiran sustav podataka, što znači:

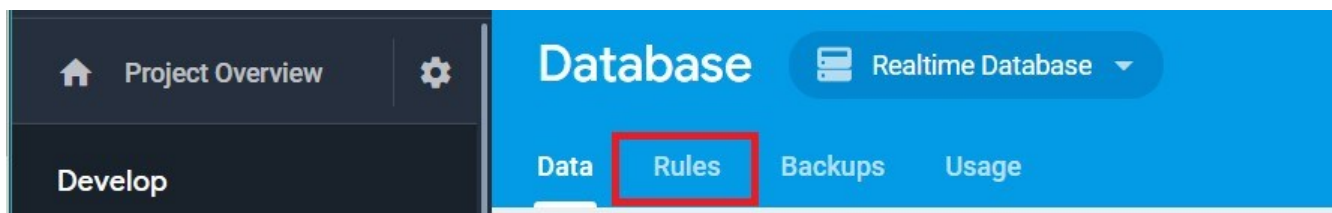
Određeni zapisi u bazi mogu biti u drugim čvorovima, ovisno o potrebi.

Svaka kolona može imati samo jednu vrijednost.

Svaki zapis može imati ugniježdenu strukturu.

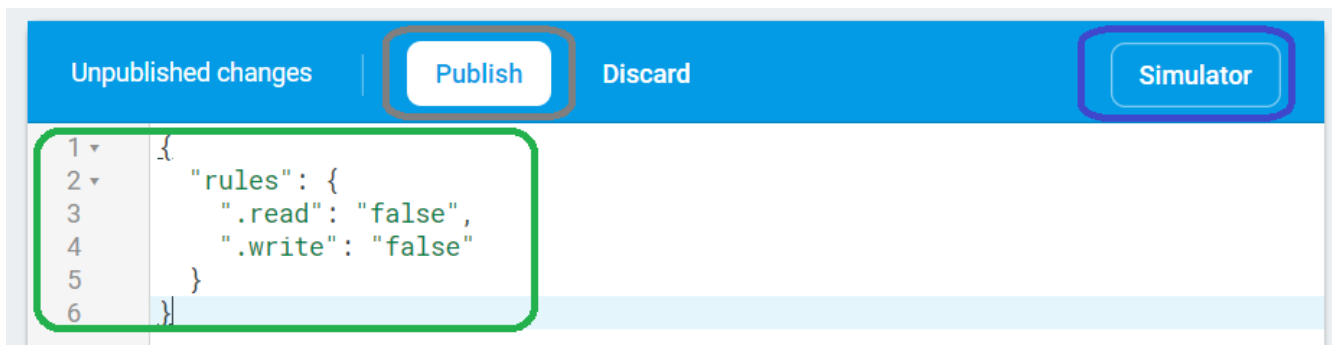
Pravila pristupa Firebase bazi podataka u razvoju

Firebase baza podataka u realnom vremenu pruža fleksibilan jezik pisanja pravila. Sintaksa pisanja pravila slična je *JavaScript*-u. Ta pravila određuju kako se podaci zapisuju, odnosno čitaju i kako bi koji podaci trebali biti indeksirani radi brže pretrage ili brži pronalazak određenog podatka.



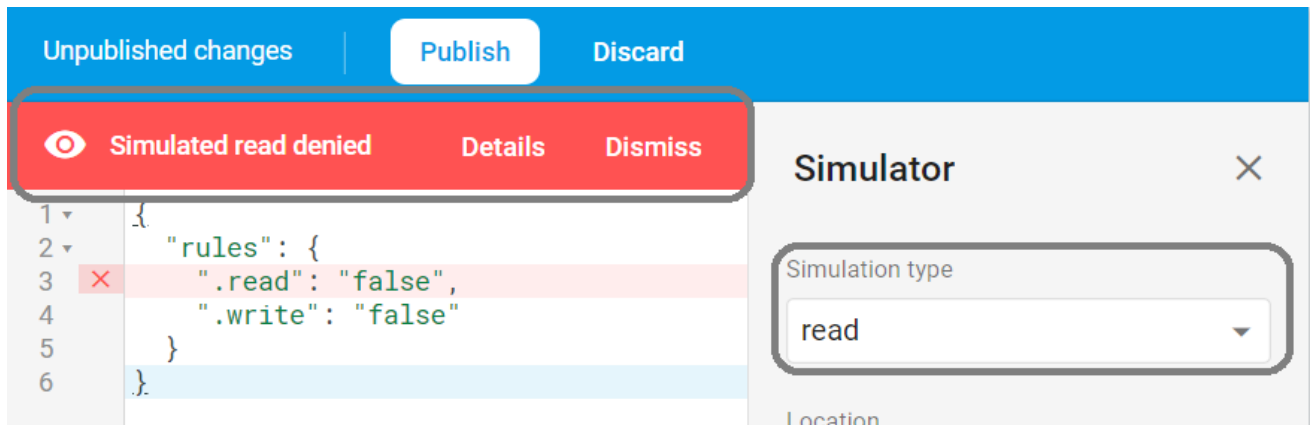
Slika 1. prikaz taba pravila u konzoli firebase database. Izradio: autor.

Podrazumijevana pravila u *Firebase* bazi podataka su da nitko nema prava zapisa i čitanja podataka:

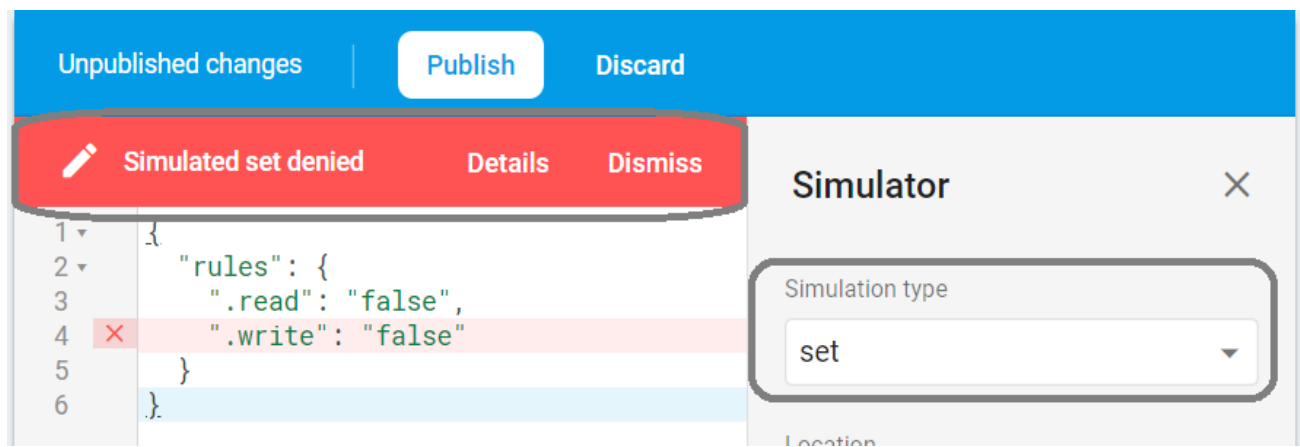


Slika 2. prikaz podrazumijevanih pravila u konzoli firebase database. Izradio: autor.

Gore navedena slika prikazuje zadani set pravila od strane *Firebase*-a, kojeg je nužno promijeniti kako bi se mogla testirati mogućnost pristupa podacima programerima u razvojnoj fazi. Jednako tako, *Google Firebase* je razvio alat za simuliranje pravila postavljenih u *Firebase* bazi podataka. To znači ako korisnik pokrene simulator na zadana pravila, nitko nema pravo zapisivanja ni čitanja podataka.



Slika 3. prikaz alata za simulacije pravila u Firebase konzloi. Izradio: autor.



Slika 4. Prikaz rada alata za simuliranje pravila u firebase bazi podataka. Izradio: autor.

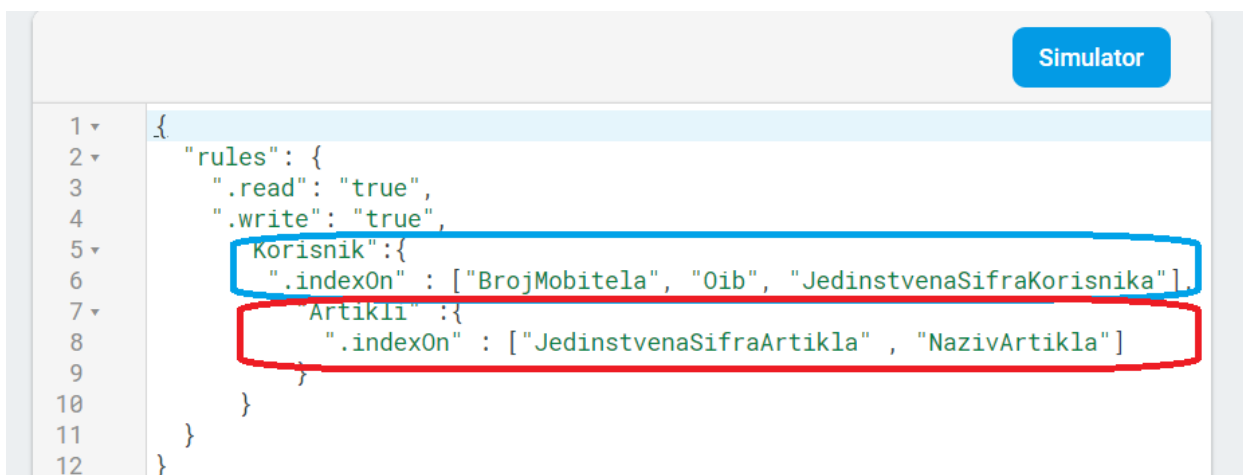
Postavljanje pravila i indeksiranje podataka u Firebase bazi podataka

U konzoli *Firebase* po zadanim postavkama pravila baze podataka zahtjev za čitanje i pisanje može izvršiti isključivo autorizirani korisnik baze podataka, a nakon toga programer lako može promijeniti pravila u fazi razvoja aplikativnog rješenja tako što jednostavno odabere svoj projekt i u kartici pravila (end. rules) postavi pravila za pisanje i čitanje na lažno (eng. false). Na taj način svatko tko ima skriptu za pristup može pristupiti podacima, što omogućava većem broju programera koji razvijaju određenu aplikaciju simultani pristup podacima. Tako se kroz obradu na klijentskoj strani ograničava tko od korisnika ima pravo pisanja i čitanja podataka u bazu.

```
1 {
2   "rules": {
3     ".read": "true",
4     ".write": "true"
5   }
6 }
```

Slika 5. prikaz postavljanja pravila za razvoj. Izradio: autor.

Kroz korištenje sučelja za regulaciju pravila jednostavno se može promijeniti način zapisa i čitanja podataka, kao što je prikazano u slici br. 5. prikaz i postavljanje pravila za razvoj aplikativnih rješenja.



```
1 {
2   "rules": {
3     ".read": "true",
4     ".write": "true",
5     "Korisnik": {
6       ".indexOn": ["BrojMobitela", "Oib", "JedinstvenaSifraKorisnika"]
7     },
8     "Artikli": {
9       ".indexOn": ["JedinstvenaSifraArtikla", "NazivArtikla"]
10    }
11  }
12 }
```

Slika 6. prikaz postavljanja pravila za razvoj, i indeksiranje po čvoru. Izradio: autor.

Kroz sučelje pravila u *Firebase* bazi podataka s realnim vremenom programeru je također omogućeno indeksiranje po zadanom čvoru. Pravila baze podataka u *Firebase*-u bazi podataka su deklarativna konfiguracija baze podataka. Ovo znači da su pravila definirana odvojeno od logike proizvoda, što ima brojne prednosti. Klijenti odnosno programeri nisu odgovorni za provođenje sigurnosti podataka, a ono najvažnije nema potrebe za posrednim servisom kao što je poslužitelj koji će služiti za zaštitu podataka unutar baze podataka. Pravila baze podataka u *Firebase*-u sastavljena su od izraza sličnih *JavaScriptu*-u koji su zapisani u *JSON*-u dokumentu, te naredna struktura podataka trebala bi slijediti strukturu podataka koja je pohranjena u bazi podataka.¹

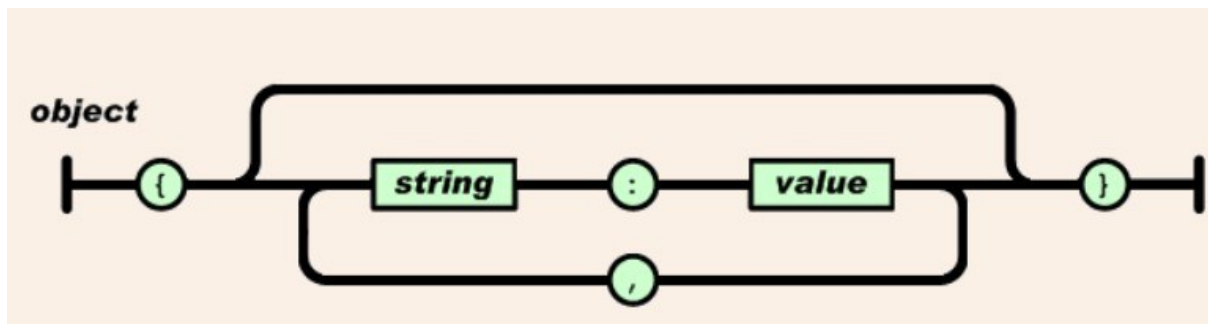
¹ (Firebase, 2019)

JSON stablo (eng. JSON Tree)

Svi podaci u *Firebase* bazi podataka pohranjuju se kao *JSON* (eng. *JavaScript Object Notation*) objekti. Za razliku od *SQL* baza podataka, *Firebase* baza podataka nema tablica niti relacija. Kada dodamo podatke u *JSON* stablo on postaje ključ u već postojećoj strukturi. *JSON* je jednostavan format razmjene podataka, zbog čega je lako čitati i zapisivati podatke a strojevi ga lako raščlanjuju i generiraju. *JSON* je tekstualni format koji je potpuno neovisan o jeziku, a koristi konvencije koje su poznate programerima iz C obitelji jezika. Ta svojstva čine *JSON* idealnim jezikom za razmjenu podataka.

JSON stablo sagrađeno je u dvije strukture:

1. Zbirka parova: imena – vrijednosti. U različitim jezicima ovo se ostvaruje kao objekt, zapis, struktura rječnik, popis s ključevima ili asocijativni niz.
2. Određen popis vrijednosti. U većini jezika to se ostvaruje kao niz, vektor, popis ili slijed.



Slika 7. Prikaz strukture objekta u JSON formi. Preuzeto sa: <https://www.json.org/>.

To su univerzalne strukture podataka. Gotovo svi suvremeni programski jezici ih podržavaju u jednom ili drugom obliku. *JSON* objekt imena/vrijednosti započinje s “{“ (lijeva zagrada) i završava s “}“ (desna zagrada). Nakon svakog imena slijedi “:“ (dvotočka) a parovi imena/vrijednosti odvojeni su “,“ (zarezom).²

² (Json.org, 2019)

Cijene i funkcionalnosti paketa Firebase baze podataka

Firestore baza podataka u realnom vremenu dolazi u tri platna paketa: ³

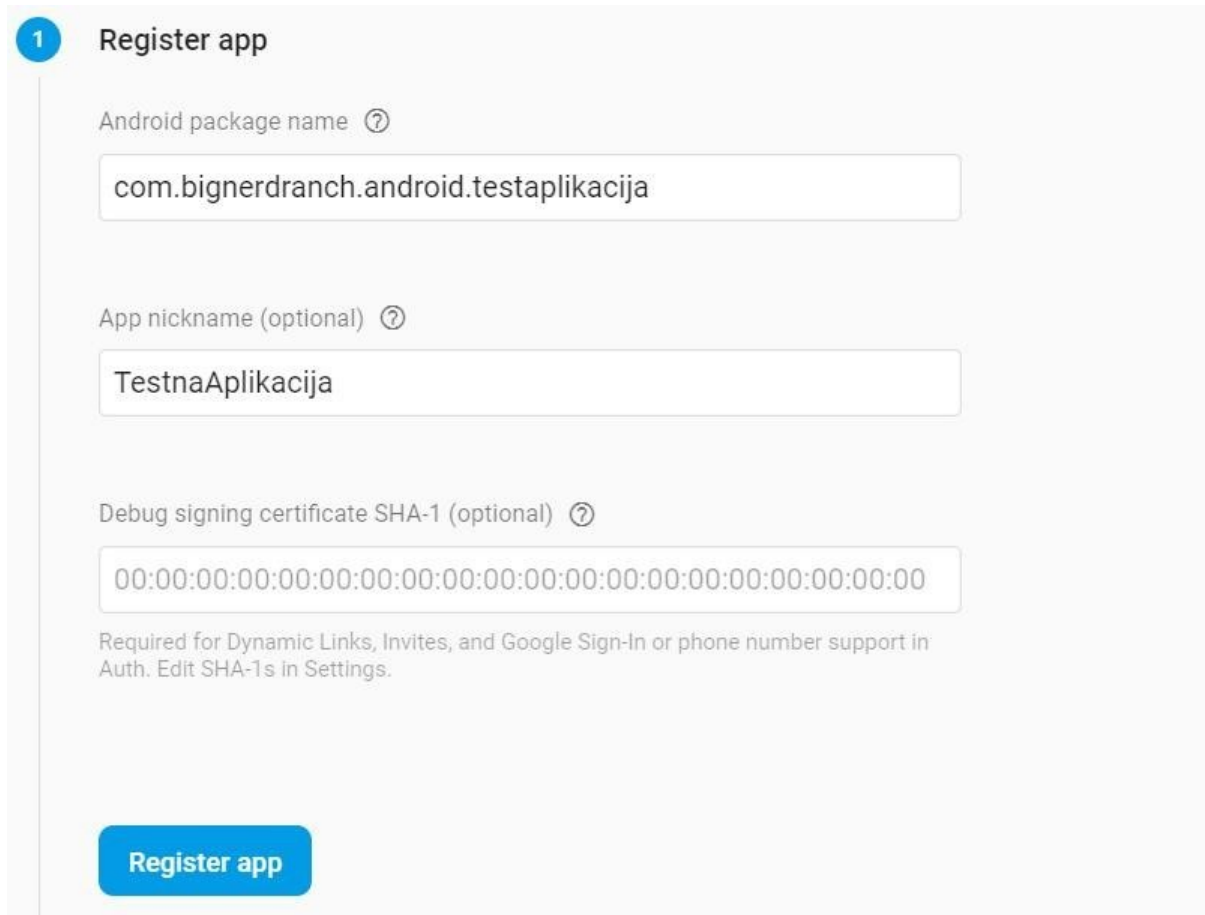
Proizvodi		Plan "Iskra" (eng. <i>spark</i>) Besplatan plan	Plan "Plamen" (eng. <i>flame</i>) 25\$/mesečno	Plan "Požar" (eng. <i>blaze</i>) Plaća se u ovisnosti veličine baze
Baza u realnom vremenu	Simultanih konekcija	100	Neograničeno	Neograničeno
	Pohrana (GB)	1 GB	2.5 GB	5\$ / GB
	Preuzimanje (GB)	10 GB	20 GB	1\$ / GB
	Dnevna back-up pohrana	Ne	Ne	Da
Analize	Da	Da	Da	
Indeksiranje baze podataka	Da	Da	Da	
Autentifikacija	Da	Da	Da	
Poruke u oblaku	Da	Da	Da	
Report o padu baze podataka	Da	Da	Da	

Tablica 1. Preuzeto sa: Firebase official: Dostupno na: <https://firebase.google.com/pricing/> Pristupljeno: 27.01.2019

³ (Firebase, 2019)

Spajanje s Firebase bazom podataka

U slučaju da je razvojna platforma *Android* studio, spajanje na bazu je nešto drugačije.



1 Register app

Android package name ⓘ

com.bignerdranch.android.testaplikacija

App nickname (optional) ⓘ

TestnaAplikacija

Debug signing certificate SHA-1 (optional) ⓘ

00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00

Required for Dynamic Links, Invites, and Google Sign-In or phone number support in Auth. Edit SHA-1s in Settings.

Register app

Slika 8. Registracija android aplikacije. Izradio: autor.

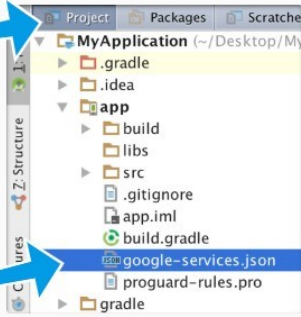
Ukoliko razvojna aplikacija zahtjeva neke od dodatnih paketa *Firebase* baze podataka u realnom vremenu, korisnik mora unijeti certifikat aplikacije *SHA-1*.


2 Download config file Instructions for Android Studio below | C++

[Download google-services.json](#)

Switch to the **Project** view in Android Studio to see your project root directory.

Move the google-services.json file you just downloaded into your Android app module root directory.



 google-services.json

[Previous](#) [Next](#)

Slika 9. Dodavanje JSNO datoteke. Izradio: autor.

U sljedećem koraku korisnik preuzima zadanu *JSON* datoteku, zatim je kopira u razvojnu aplikaciju u *Android* studiju.

3

Add Firebase SDK

Instructions for Gradle | [C++](#)

The Google services plugin for [Gradle](#) loads the `google-services.json` file you just downloaded. Modify your `build.gradle` files to use the plugin.

Project-level `build.gradle` (<project>/`build.gradle`):

```
buildscript {
    dependencies {
        // Add this line
        classpath 'com.google.gms:google-services:4.0.1'
    }
}
```



App-level `build.gradle` (<project>/<app-module>/`build.gradle`):

```
dependencies {
    // Add this line
    implementation 'com.google.firebase:firebase-core:16.0.1'
}
...
// Add to the bottom of the file
apply plugin: 'com.google.gms.google-services'
```



Includes Analytics by default

Finally, press "Sync now" in the bar that appears in the IDE:

Gradle files have changed sir

Previous

Next

Slika 10. Registracija android aplikacije. Izradio: autor.

U `gradle` datoteci u *Android* studiju potrebno je dodati slijedeće priključke i sinkronizirati `gradle` datoteku. Ukoliko se ne javi greška, aplikacija razvijana u *Android* studiju povezana je s *Firebase* bazom podataka u realnom vremenu.

Cloud Firestore

Firestore nudi dva rješenja za baze podataka utemeljena na oblaku i bazirana na klijentu, koja podržavaju sinkronizaciju podataka u stvarnom vremenu. Prvo rješenje je *Firestore* baza podataka u realnom vremenu koju smo obradili u prvom dijelu rada, a drugo rješenje je *Cloud Firestore* baza podataka. *Cloud Firestore* je nova baza podataka razvijena od *Google*-a u listopadu 2017 godine. namijenjena za razvoj mobilnih aplikacija. *Cloud Firestore* je fleksibilna, skalabilna baza podataka i omogućuje razvoj aplikacijskih rješenja u više različitih poslužitelja. Poput *Firestore* baze podataka u realnom vremenu, ona sinkronizira podatke na klijentskoj strani. Također nudi i podršku za mobilne uređaje i internetske aplikacije (eng. web application) u izvanmrežnom stanju, kako bi mogli izgraditi prilagodljive aplikacije koje rade bez obzira na latenciju mreže ili internetsku vezu. *Cloud Firestore* također nudi besprijeckornu integraciju s drugim proizvodima *Firestore* i *Google Cloud Platform*, uključujući *Cloud* funkcije.

Ključne mogućnosti:

1. **Fleksibilnost:** Podatkovni model *Cloud Firestore* omogućuje fleksibilne hijerarhijske strukture podataka tako što se podaci spremaju u dokumente koji su organizirani kao zbirke podataka. Dokumenti mogu sadržavati složene ugniježdene objekte u podzbirkama.
2. **Upiti:** U *Cloud Firestore* možete koristiti upite za dohvaćanje pojedinačnih određenih dokumenata ili dohvaćanje svih dokumenata u zbirci koji odgovaraju postavljenim parametrima upita. Upiti mogu uključivati višestruke ulančane filtre i kombinirano filtriranje i sortiranje.
3. **Ažuriranje u stvarnom vremenu:** Kao i u *Firestore* bazi podataka u realnom vremenu, *Cloud Firestore* koristi sinkronizaciju podataka na bilo kojem povezanom uređaju.
4. **Podrška za izvanmrežni rad:** *Cloud Firestore* sprema podatke koje aplikacija aktivno koristi. Korisnik aplikacije može zapisivati, čitati i tražiti podatke čak i ako je uređaj izvan mreže. Kada se uređaj ponovno poveže na internetsku mrežu *Cloud Firestore* sinkronizira sve lokalne promjene koje su se promijenile u odnosu na posljednju sinkronizaciju.

Cloud Firestore je *NoSQL* baza podataka pohranjena u oblaku gdje neovisno o klijentu programeri mogu pristupiti izravno putem *SDK* (eng. *Software development kit*). Podatkovni model *Cloud Firestore* pohranjuje podatke u dokumente koji sadrže polja koja se pohranjuju u određene vrijednosti. Takvi su dokumenti pohranjeni u zbirinama. Zbirke su spremnici za dokumente koje programer strukturira kako bi mogao izvršavati organiziranje podataka i izradu upita. Dokumenti podržavaju mnoge različite vrste podataka, od jednostavnih nizova brojeva i tekstualnih podataka do složenih ugniježdenih objekata. Također se mogu kreirati podzbirke unutar dokumenata i izgraditi hijerarhijske strukture podataka koje se šire dok se baza podataka širi. Podatkovni model *Cloud Firestore* podržava bilo koju strukturu podataka koja najbolje funkcionira za aplikacijsko rješenje.⁴

⁴ (Anon, 2019)

Usporedba Firebase database i Cloud Firestore

Oba rješenja *Cloud Firestore* i *Firebase* baza u realnom vremenu su *NoSQL* baze podataka, te su podaci pohranjeni u oblaku. *Cloud Firestore* nudi dodatnu funkcionalnost i skalabilnost na infrastrukturi osmišljenoj da podrži snažnije značajke u budućim izdanjima. *Google* predviđa kako će *Cloud Firestore* imati nove vrste upita i snažnija sigurnosna pravila. Trenutno je *Cloud Firestore* u beta verziji.

Firestore baza u realnom vremenu	Cloud Firestore
<u>Modeliranje podataka</u>	
<p>Pohranjuje se u jedan JSON dokument</p> <p>Jednostavno za pohranu podataka</p> <p>Složeni, hijerarhijski podaci teže se strukturiraju i organiziraju u JSON objektu</p>	<p>Pohranjuje podatke u dokumentima koji su organizirani u zbirke</p> <p>Jednostavne podatke lako je pohraniti u dokumentima koji su vrlo slični JSON-u</p> <p>Složeni, hijerarhijski podaci lakše se organiziraju na skali, koristeći podzbirke unutar dokumenata.</p>
<u>Izvanmrežni rad</u>	
Izvanmrežni rad omogućen je na mobilnim klijentima, odnosno na iOS i Android uređajima	Izvanmrežni rad omogućen je na mobilnim klijentima iOS i Android, te i na web klijentima
<u>Upiti</u>	
Upiti s ograničenom funkcijom sortiranja i filtriranja podataka	Indeksirani upiti s sortiranjem i filtriranjem spojeva

<p>Sortiranje i filtriranje omogućeno samo na svojstvo (eng. property), a ne sortirati i filtrirati svojstvo u jednom upitu</p> <p>Upiti uvijek vraćaju cijelo podstablo</p>	<p>Omogućeno je povezati filtere i kombinirati filtriranje i sortiranje na svojstvu (eng. property) u jednom upitu</p> <p>Ukoliko su plitki upiti zapisani u podzbirke, moguće je tražiti podzbirke unutar dokumenta umjesto cijele zbirke ili čak cijelog dokumenta</p> <p>Upiti se indeksiraju prema zadanim postavkama, izvedba upita je proporcionalna veličini skupa rezultata, a ne skup podataka</p>
<p><u>Pisanje i transakcije</u></p>	
<p>Osnovne operacije pisanja i transakcija</p> <p>Podaci se zapisuju kao pojedinačna operacija</p> <p>Transakcije u izvornim SDK-ovima zahtijevaju povratni poziv (eng. callback)</p>	<p>Atomske operacije pisanja i transakcija</p> <p>Transakcije se automatski ponavljaju dok se ne izvrše</p>
<p><u>Pouzdanost i performanse</u></p>	
<p>Firestore baza podataka u realnom vremenu je na tržištu već duže vrijeme, te je većina grešaka ispravljena</p> <p>Stabilnost je na razini višegodišnjih proizvoda u polju baza podataka</p> <p>Vrlo je niska latencija, pa je odlična opcija za česte sinkronizacije podataka</p>	<p>Cloud Firestore je trenutno u beta verziji</p> <p>Stabilnost u beta proizvodu nije uvijek jednaka kod potpuno lansiranog proizvoda</p> <p>Omogućena je veća skalabilnost te veća pouzdanost</p> <p>Kada Cloud Firestore pređe u finalnu verziju imati će jaču pouzdanost od Firestore baze podataka u stvarnom vremenu</p>

Skalabilnost	
<p>Skaliranje baze podataka zahtijeva “sječkanje“ podataka na više manjih particija (eng. sharding)</p> <p>Skalira se na oko 100 000 istovremenih veza i na 1000 zapisa po sekundi u jednoj bazi podataka</p>	<p>Skaliranje se izvršava automatski</p> <p>Automatsko skaliranje znači da razvojni programer neće morati pohranjivati podatke na više instanci</p>
Sigurnost	
<p>Kaskadna pravila koja zahtijevaju zasebnu provjeru valjanosti</p> <p>Firestore baza podataka u stvarnom vremenu podržava svoja pravila koja treba definirati</p>	<p>Jednostavnija, snažnija sigurnost za SDK-ove za mobilne uređaje i web poslužitelje</p> <p>Poslužiteljski SDK koriste upravljanje identitetima i pristupom (eng. IAM Identity and access management)</p> <p>Provjera valjanosti podataka izvršava se automatski</p> <p>Pravila se mogu ograničiti na upite, primjerice ako rezultati upita mogu sadržavati podatke kojima korisnik nema prava pristupa, upit ne uspijeva</p>
Cjenovni paketi	
<p>Naplaćuje se po broju konekcija i količini pohranjenih podataka</p>	<p>Naplaćuje se prvenstveno za operacije koje se obavljaju u bazi podataka (čitanje, pisanje, brisanje podataka) te po nižoj stopi neovisno o broju konekcija i količini pohranjenih podataka</p>

	Cloud Firestore podržava dnevno ograničenje potrošnje, kako bi korisnici aplikacije bili sigurni da ne prelaze planirane troškove
--	---

Tablica 2. Usporedba Firebse databse i firebase cloud sa: Firebase official: Dostupno na: <https://firebase.google.com/pricing/>
Pristupljeno: 27.01.2019

Trenutno je moguće koristiti obje baze podataka unutar iste *Firebase* aplikacije ili projekta. Objе *NoSQL* baze podataka mogu pohraniti iste vrste podataka.⁵

⁵ (Firebase, 2019)

Relacijske baze podataka

Glavno obilježje koje razdvaja relacijske i nerelacijske baze podataka je način na koji se organiziraju i strukturiraju podaci unutar baze. Kod baze podataka relacijskog tipa podaci se spremaju u tablice, a tablice se povezuju relacijama. Stoga, da bi se podrazumijevalo da je baza podataka relacijska baza, ona mora imati dvije ili više tablica i minimalno jednu relaciju između dvije tablice.

	Code	Name	Continent	Region	SurfaceArea	Year	Population	LifeExpectancy	GNP	GNPOld	LocalName	GovernmentForm	HeadOfState	CapitalCode2
<input type="checkbox"/>	COM	Comoros	Africa	Eastern Africa	1862.00	1975	578000	60.0	4401.00	4361.00	Komori/Comores	Republic	Araki Assoumani	2295 KM
<input type="checkbox"/>	COG	Congo	Africa	Central Africa	342000.00	1960	2943000	47.4	2108.00	2287.00	Congo	Republic	Denis Sassou-Nguesso	2296 CG
<input type="checkbox"/>	COD	Congo, The Democratic Republic of the	Africa	Central Africa	2344858.00	1960	51654000	48.8	6994.00	2474.00	République Démocratique du Congo	Republic	Joseph Kabila	2298 CD
<input type="checkbox"/>	COK	Cook Islands	Oceania	Polynesia	236.00		20000	71.1	100.00		The Cook Islands	Nonmetropolitan Territory of New Zealand	Elisabeth II	583 CK
<input type="checkbox"/>	CRI	Costa Rica	North America	Central America	51100.00	1821	4023000	75.8	10226.00	9757.00	Costa Rica	Republic	Miguel Angel Rodríguez Echeverría	584 CR
<input type="checkbox"/>	CIV	Côte d'Ivoire	Africa	Western Africa	322463.00	1960	14786000	45.2	11345.00	10285.00	Côte d'Ivoire	Republic	Laurent Gbagbo	2814 CI
<input type="checkbox"/>	HRV	Croatia	Europe	Southern Europe	56538.00	1991	4473000	73.7	20208.00	19300.00	Hrvatska	Republic	Stipe Mesić	2409 HR
<input type="checkbox"/>	CUB	Cuba	North America	Caribbean	110861.00	1902	11201000	76.2	17843.00	18862.00	Cuba	Socialistic Republic	Fidel Castro Ruz	2413 CU
<input type="checkbox"/>	CYP	Cyprus	Asia	Middle East	9251.00	1960	754700	76.7	9333.00	8246.00	Kýpros/Kıbrıs	Republic	Glafkos Klontides	2430 CY
<input type="checkbox"/>	CZE	Czech Republic	Europe	Eastern Europe	78866.00	1993	10278100	74.5	55017.00	52037.00	Česko	Republic	Václav Havel	3339 CZ
<input type="checkbox"/>	DNK	Denmark	Europe	Nordic Countries	43094.00	800	5330000	76.5	174099.00	169264.00	Danmark	Constitutional Monarchy	Margrethe II	3315 DK
<input type="checkbox"/>	DJI	Djibouti	Africa	Eastern Africa	23200.00	1977	638000	50.8	382.00	373.00	Djibouti/Jibuti	Republic	Ismail Omar Guelleh	585 DJ
<input type="checkbox"/>	DMA	Dominica	North America	Caribbean	751.00	1978	71000	73.4	256.00	243.00	Dominica	Republic	Vernon Shaw	586 DM
<input type="checkbox"/>	DOM	Dominican Republic	North America	Caribbean	48511.00	1844	8495000	73.2	15846.00	15076.00	República Dominicana	Republic	Hipólito Mejía Domínguez	587 DO

Slika 11. Prikaz relacijske tablice država. Preuzeto sa: <https://demo.phpmyadmin.net/master-config/sql.php?server=2&db=world&table=Country&pos=0>

Tablice se sastoje od redaka koji predstavljaju pojedini ulaz, te stupaca koji predstavljaju jednu vrstu podatka (Npr: Ime, Prezime, KorisnikID). Prije samog početka unosa podataka u bazu, svaku tablicu odnosno svako polje treba definirati vrstu podatka u svakom stupcu te odrediti shemu relacija između svake pojedine tablice. Kako bi relacijska baza podataka bila efektivna potrebno je dizajnirati shemu kako bi spremljeni podaci bili strukturirani i organizirani. Tako ne bi dolazilo do kršenja pravila baze podataka, redundancije ili nesklada tablica.⁶

⁶ (Lorentz 2005 str: 30-31)

SQL (eng. Structured Query Language)

Predstavlja strukturirani jezik za upite u relacijskim bazama podataka (eng. *SQL*). To je programski jezik posebno namijenjen upravljanju podataka koji se čuvaju u sustavu za upravljanje relacijskom bazom podataka (eng. *Relational Database Management System, RDMS*). *SQL* pruža pogodnosti za sve vrste korisnika, uključujući razvojne programere, administratore baza podataka te krajnje korisnike. *SQL* omogućuje rad s podacima na logičnoj razini. Svi vodeći sustavi za upravljanje relacijskim bazama podataka podržavaju *SQL*, tako da se vještine koje su stečene na jednoj bazi podataka lako prenesu na drugu.

SQL Server says	Access says	Access Data Field Length
bigint	Short Text	255
int	Number	Long Integer
smallint	Number	Integer
tinyint	Number	Byte
binary(50)	Binary	50
bit	Yes/No	
char(10)	Short Text	10
char(300)	Long Text	
date	Date/Time	
datetime	Date/Time	
datetime2	Date/Time	
datetimeoffset	Short Text	255
decimal	Number	18,0
float	Number	Double
geography	Short Text	255
geometry	Short Text	255
hierarchyID	Short Text	255
image	OLE Object	
money	Currency	
nchar(10)	Short Text	10
nchar(300)	Long Text	
ntext	Long Text	
numeric	Number	18,0
nvarchar(50)	Short Text	50
nvarchar(300)	Long Text	
nvarchar(max)	Long Text	
real	Number	Single
smalldatetime	Date/Time	
smallmoney	Currency	
sqlvariant	Short Text	255
text	Long Text	
time	Short Text	255
timestamp	Binary	8
uniqueidentifier	Number	ReplicationID
varchar(10)	Short Text	10
varchar(300)	Long Text	
varbinary	Binary	50

Slika 12. Prikaz tipova podataka u SQL-u. Preuzeto sa: <https://www.brentozar.com/sql-server-access-data-types/>

Za razliku od baze podataka usmjerene prema dokumentu gdje nije potrebno definirati tip podatka koji se pohranjuje u bazu podataka, kod *SQL*-a je potrebno definirati svaki tip podatka koji se pohranjuje u bazu podataka.⁷

⁷ (ibidem, str 32,33,35)

Usporedba Firebase baze podataka i relacijske baze podataka

Firestore baza podataka u realnom vremenu je *NoSQL* baza podataka, dok je *Oracle SQL* relacijska baza podataka, koja je puno zastupljenija u programskim aplikacijama, ponajviše zato što *Oracle* baza podataka datira još iz 1980 god. Pojam *NoSQL* prvi put korišten je 1998 godine, a prva baza podataka vođena dokumentima (*NoSQL*) kakvu danas poznajemo bila je *Mongo DB* čija inicijalna verzija nastaje 2009 godine. *Firestore* baza podataka nastala je nedavne 2012 godine.⁸

Naziv baze podataka	Firestore baza podataka	Oracle
Opis	Pohranjuje podatke u oblaku, serveri nisu dostupni krajnjem korisniku.	Najšire korištena baza podataka RDBMS (relational database management systems)
Primarni model baze	Dokumentima vođen	Relacijski model podataka
Sekundarni model baze		Dokumentom vođen Ključ-vrijednost pohrana
Dostupno na	firebase.google.com/- products/realtime- database	<a href="http://www.oracle.com/-
database">www.oracle.com/- database
Razvijeno od	Google	Oracle
Inicijalno izdanje	2012 godine	1980 godine
Trenutna verzija		18 (18.1) Veljača, 2018
Licenca	Komercijalna	Komercijalna

⁸ (Fehily, 2005)

Dostupna jedino u oblaku	Da	Ne
Jezik u kojem je pisana		C i C++
Operacijski sustav poslužitelja		AIX (Advanced Interactive eXecutive) HP-UX (Hewlett Packard Unix) Linux Solaris Windows
Schema podataka	Ne	Da
XML (Extensible Markup Language) podrška	Ne	Da
Primarno indeksiranje	Da	Da
Sekundarno indeksiranje	Da	Da
SQL jezik	Ne	Da
Podržani programski jezici	Java JavaScript Objective-C C++	C C# C++ Clojure Cobol Delphi Eiffel

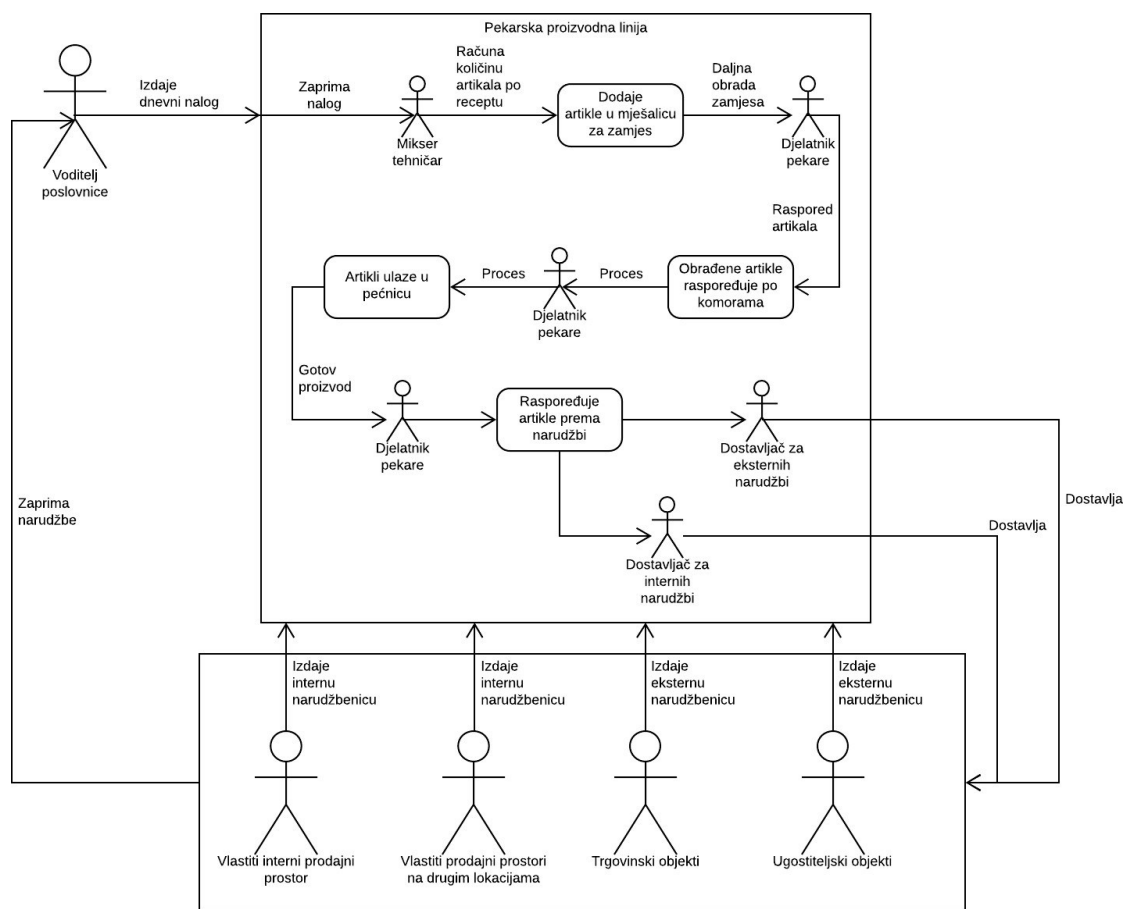
		Erlang Fortran Groovy Java JavaScript Lisp Objective-C Perl PHP Python Ruby Scala Visual Basic ⁹
--	--	---

Tablica 3. Usporedba Firebase baze podataka u stvarnom vremenu i SQL Oracle relacijske baze podataka

⁹ (Db-engines.com, 2019)

Opis dosadašnjeg procesa

Dosadašnja praksa u pekarskom postrojenju vršila se na način da voditelj postrojenja prima sve narudžbenice odnosno narudžbenice iz internih prodajnih prostora kao i narudžbenice klijentskih poslovnica (restorana, prodajnih centara, trgovina i sl.). Nakon prikupljenih narudžbenica voditelj postrojenja ih obrađuje, potom ručno ispisuje nalog s zadanim recepturama i njihovim količinama te ih šalje u postrojenje.



Slika 13. Shematski prikaz pekarske proizvodne linije. Izradio: autor

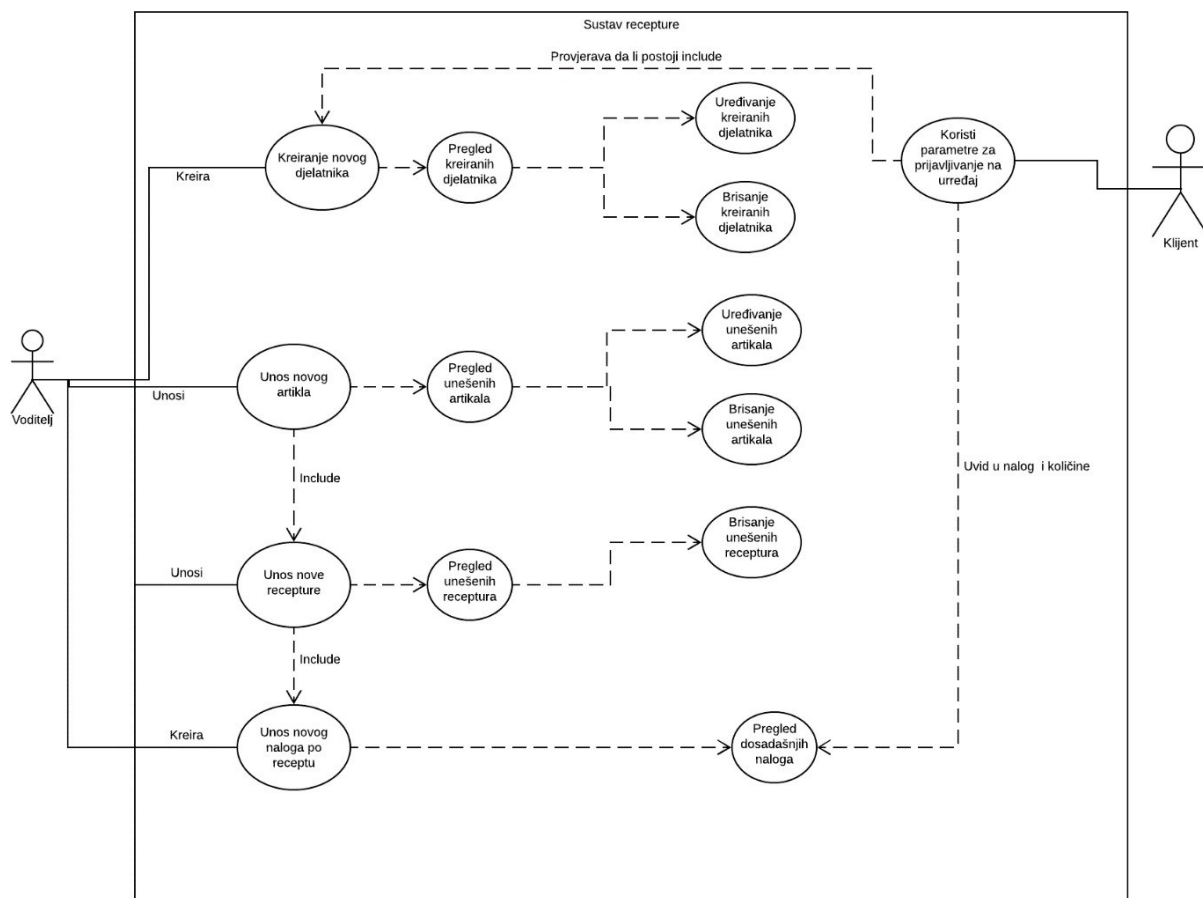
Nakon što voditelj obradi interne i eksterne narudžbe te izradi nalog, prosljeđuje ga pekarskoj proizvodnoj liniji. Zaprimanje naloga vrši mikser tehničar, te po količinama određenih receptura iz naloga svaki put iznova računa količinu artikla po normativu potrebnih za određeni zamjes. Ukoliko su postotci artikala u određenoj recepturi dobro izračunati daljnji nastavak procesa u većini slučajeva biti će uspješan. Ukoliko se potkrala greška mikser tehničaru postoji mogućnost da se ne uspije detektirati sve do kraja proizvodnog procesa, što dovodi do gubitka kako materijalnih resursa tako i vremenskih resursa. Ukoliko je greška ranije detektirana postoji mogućnost da se receptura ispravi ili doradi, a ukoliko je ta mogućnost isključena onda je gubitak samo materijalan. Kritična točka u pekarskoj proizvodnoj liniji je davanje naloga mikser tehničaru te računanje potrebnih artikala za određeni zamjes.

Korisnički zahtjevi

U svrhu poboljšanja rada i unaprjeđivanja proizvodnje u pekarstvu korisnički zahtjevi bazirani su na olakšavanju i ubrzavanju posla mikser tehničaru i lakšem prosljeđivanju naloga. Drugi upit u korisničkom zahtjevu je i ograničiti dostupnost informacija određenim djelatnicima. Korisnički zahtjevi omogućili bi brži i efikasniji posao mikser tehničara te smanjenje kašnjenja proizvoda i smanjenje škartiranih zamjesa, što direktno utječe na poboljšanje poslovnog procesa, financijske i vremenske uštede.

Opis informatiziranog procesa vođenja pekarskog poslovanja

Aplikacija za koordinaciju procesa izrade pekarskih proizvoda podijeljena je u dva segmenta. Prvi segment odnosno *WEB* aplikacija sastoji se od 4 modula i baziran je na: definiranju djelatnika, artikala, receptura i naloga kojima upravlja voditelj poslovnice. Drugi dio namijenjen je radnicima pekarskog postrojenja, koje je prethodno definirao voditelj poslovnice. *Android* aplikacija je u suštini digitalni nalog koji im omogućuje brži i jednostavniji pregled naloga i detalje određenog zamjesa iz naloga.



Slika 14. Dijagram slučajeva korištenja (eng. Use case diagram). Izradio: Autor

Funkcionalnosti aplikacije

Izrada *WEB* aplikacije za praćenje rada pekarskog postrojenja namijenjena voditelju poslovnice odnosno administratoru, temeljena je na vizualno privlačnom i jednostavnom dizajnu što omogućuje svim tipovima korisnika lagano korištenje i upravljanje aplikacijom. Sastoji se od četiri modula, a svi su organizirani na sličan identičan način.

U modulu Djelatnici implementirane su sljedeće funkcionalnosti:

Unos novog korisnika:

Ime djelatnika

Prezime djelatnika

Lozinka

OIB djelatnika

Broj mobitela djelatnika

Pregled unesenih djelatnika:

Pregled unesenih djelatnika

Uređivanje/izmjena djelatnika

Uređivanje/brisanje postojećeg djelatnika

U modulu Skladište implementirane su sljedeće funkcionalnosti:

Unos novog artikla:

Naziv artikla

Jedinica mjere artikla

Barkod artikla

Naziv dobavljača artikla

Ulazna cijena proizvoda

Kratki opis artikla

Pregled unesenih artikala:

Pregled unesenih artikala

Uređivanje/izmjena artikla

Uređivanje/brisanje postojećeg artikla

U modulu Recepture implementirane su slijedeće funkcionalnosti:

Unos novog recepture:

Naziv recepture

Odabir naziva artikla u zamjesu

Udio artikla u zamjesu (količina artikla po jednom proizvodu)

Spremanje recepture u bazu

Pregled unesenih recepata:

Pregled unesenih recepata

Uređivanje/brisanje postojećeg recepta

U modulu Naloga implementirane su slijedeće funkcionalnosti:

Unos novog naloga:

Odabir recepture po nazivu

Unos količine odabranog recepta

Pregled unesenih naloga:

Pregled unesenih naloga

Pretpregled izdanog naloga

Implementirana *Android* aplikacija za praćenje rada pekarskog postrojenja namijenjena je djelatnicima pekare odnosno korisniku. Aplikacija omogućuje implementaciju na mobilnim ili tablet uređajima, zbog čega je jednostavna i laka za korištenje. Sadrži dva modula, odnosno modul za prijavu i modul naloga.

U modulu Prijava korisnika implementirane su sljedeće funkcionalnosti:

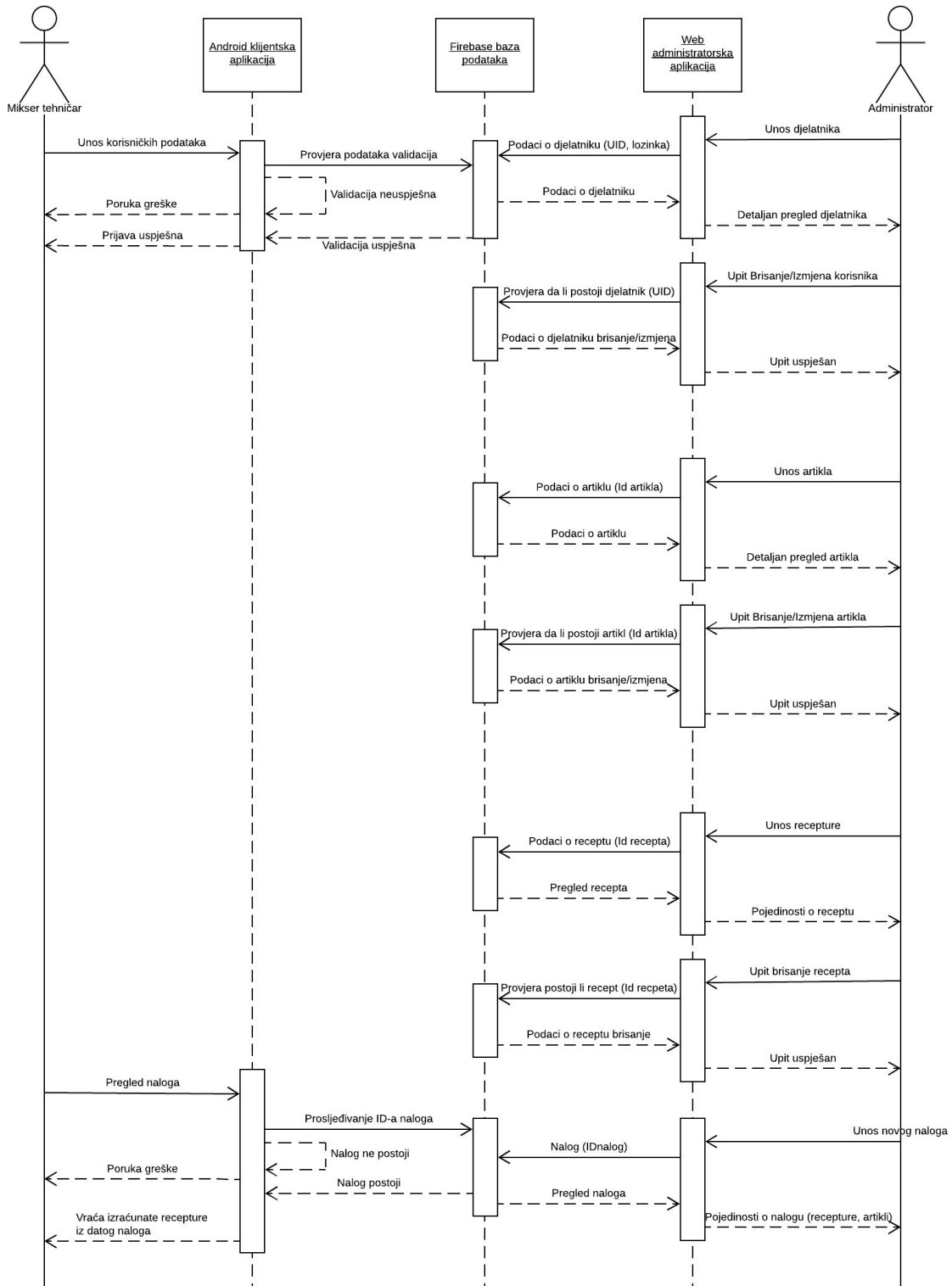
Prijava korisnika koji je prethodno definiran od strane voditelja poslovnice

U modulu Naloga implementirane su sljedeće funkcionalnosti:

Pregled izdanih naloga po datumima

Pregled stavki naloga

Precizan izračun artikla za svaku stavku naloga



Slika 15. Sekvencijski dijagram (eng. Sequence diagram). Izradio: Autor

Korištenje tehnologije i programski alati

Na temelju analize arhitekture programskog zadatka te potrebe za brzim pristupom potrebnih informacija korištena je *Firebase* baza podataka u realnom vremenu. Internetska aplikacija za voditelja poslovnice rađena je u HTML (*HyperText Markup Language*), CSS (*Cascading Style Sheet*) i *JavaScript* tehnologijama za razvoj, dok je za implementaciju aplikacije za krajnjeg korisnika odnosno mikser tehničara korišten *Android Studio*.

Android Studio

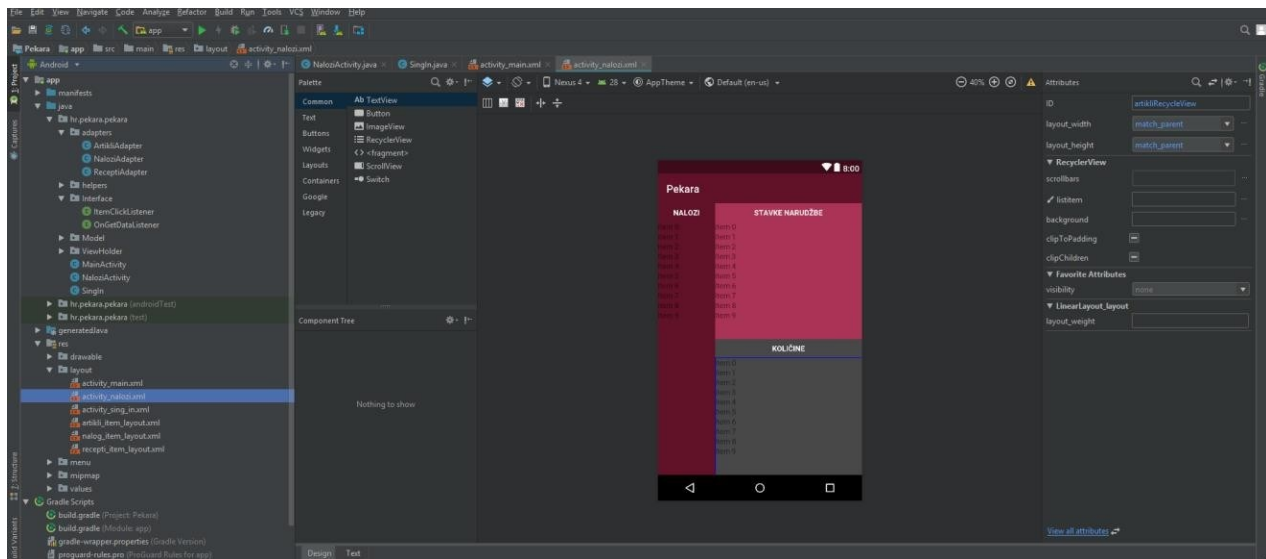
Android Studio danas je najkorištenija aplikacija za razvoj mobilnih aplikacija na Android platformi. Razvijena je od strane *Google*-a, a prva inačica objelodanjena je u svibnju 2013 godine. Danas je dostupna verzija 3.3 izdana u Siječnju 2019 godine, koja donosi brojne nadogradnje a najznačajnija je podrška za programski jezik *Kotlin*, koji ima tendenciju da zamjeni dosadašnji jezik *Javu* koji je jedan od najzastupljenijih jezika za razvoj mobilnih aplikacija. Za razliku od *Jave*, *Kotlin* razvojni jezik ima čišću sintaksu pisanja koda i manju potrebu koda u odnosu na rezultate. *Android* studio omogućava razvoj aplikativnih rješenja za sve uređaje koji koriste *Android* kao operacijski sustav:

Wear OS (Pametni sat)

Android TV

Android Auto

Mobilni i tabletni uređaji



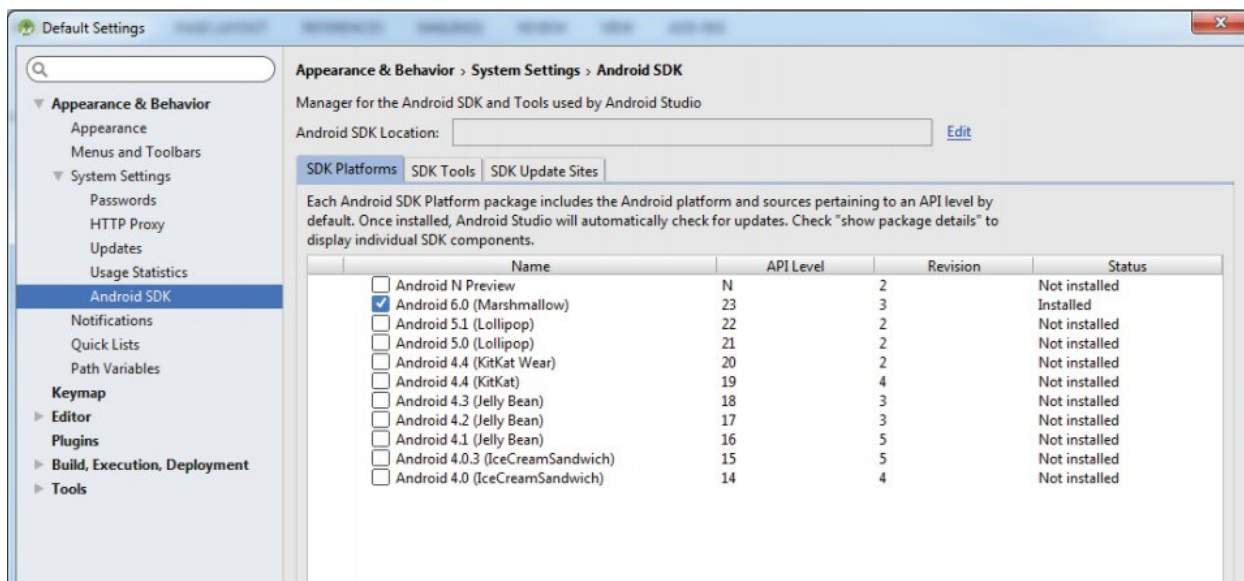
Slika 16. Android studio. Izradio: Autor

Android studio nudi više značajki koje povećavaju vašu produktivnost pri izradi raznih aplikacijskih rješenja kao što su: fleksibilni *Gradle* sustav za gradnju aplikacija, brz i efikasan emulator, integrirani predlošci koda koji pomažu u izgradnji uobičajenih značajki aplikacije, opsežni alati za testiranje i sl.¹⁰ Velika prednost korištenja *Android Studija* za razvoj je to što je najviše rješenja pisano u *Java* programskom jeziku koji se koristi već duži niz godina, te je veoma jednostavno pronaći već gotova programska rješenja i primjere. Nedostatak korištenja *Android studija* za razvoj mobilnih aplikacija je ta što svako rješenje može plasirati jedino na uređaje koji koriste *Android* operacijski sustav, dok je za druge operacijske platforme programsko rješenje potrebno implementirati iznova, što direktno utječe na veće troškove kod samog razvoja mobilne aplikacije.

¹⁰(Android Developers, 2019)

Android SDK

Najvažniji dio *Android Studija* je oprema za razvijanje programskog rješenja (eng. *Android SDK Software Development Kit*). Android SDK sadrži sve pakete i alate potrebne za razvoj funkcionalne *Android* aplikacije. *Android SDK*-ovi nazvani su prema verziji *Android OS* (eng. operacijskog sustava) kojima odgovaraju. Zadani *Android SDK* je *Marshmallow SDK*, što znači da je omogućen razvitak aplikacija za sve android uređaje s *Android Marshmallow*.



Slika 17. Prikaz SDK Manager. Preuzeto sa: KNJIGE O KOJOJ JE PISANO SVE O SDK-OM

Ovisno o potrebi instalacija druge verzije *Android SDK* može se izvršiti iz *SDK Managera* koji se nalazi u *Android Studiju*.¹¹

¹¹ (DiMarzio, n.d. str. 15-16)

HTML (eng. HyperText markup Language)

HyperText Markup Language (HTML) jedan je od tri ključna jezika za razvoj i izradu internetskih stranica. Nastao je 1993. godine i glavna namjena je opisni izgled internetske stranice, zbog čega se ne smatra programskim jezikom.

```
1 <!doctype html>
2 <html>
3 <head>
4 <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
5 integrity="sha256-3dr9oy0665f8f8b8sqwzj2lM60nkC1P0py+7E="
6 crossorigin="anonymous"></script>
7 <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
8 <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.min.js"
9 integrity="sha256-VazP97ZCutekAsvqP85bPFKdrD3unHf5GVrahlqJ="
10 crossorigin="anonymous"></script>
11 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js" integrity="sha384-smFYKDLADwXn1Eml1qk/HfndicbVRZYmZ4qpPea65jB/pT30euyQp0M8ck+ST" crossorigin="anonymous"></script>
12 <script src="C:\Users\Sean\Documents\VS2017\PrijavaForma\prijava.js"></script>
13 <link rel="stylesheet" href="C:\Users\Sean\Documents\VS2017\PrijavaForma\prijava.css">
14 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/css/bootstrap.min.css" integrity="sha384-MskhaSGFgHMDcbw70/dFVBj47jz9qbsMIId/1RN3ewGhXQFZC5ftd1LZCfmhktB" crossorigin="anonymous">
15 <meta charset="utf-8">
16 <title> Prijava vec unešenih korisnika </title>
17 <script src="prijava.js"></script>
18 <link href="css/bootstrap.min.css" rel="stylesheet">
19 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384-BvPi11f166W/vTAE/8SI41G/Af4s300luTbcv7/3I6fYp73f7237z80fY6316A" crossorigin="anonymous">
20 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap-theme.min.css" integrity="sha384-rhdyb1113/V3Y4B0tUn6eJD1h1kOUg9rt9G78J2h8B/i27CafD7gnJfQk2i5e" crossorigin="anonymous">
21 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js" integrity="sha384-Tc5V93124/P0U04tJy6W/jh8I/s0p0IC793kYyVW/Z98Q6wK64J3" crossorigin="anonymous"></script>
22 <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
23 <script src="js/bootstrap.min.js"></script>
24 </head>
25 <body>
26 <div>
27 <h1>Dobrodošli</h1>
28 <p>Dobrodošli natrag na web aplikaciju</p>
29 </div>
30 </body>
31 </html>
```

Slika 18. Primjer izgleda HTML stranice. Izradio: autor

Na slici je prikazan odsječak *HTML* koda, koji se sastoji od glave dokumenta (eng. *head*) i tijela dokumenta (eng. *body*). U glavi dokumenta unose se vanjski dokumenti za estetski izgled stranice, funkcionalnost stranice te spajanje na bazu. U tijelu *HTML* dokumenta unose se oznake (eng. *Tag*) koje se koriste za izgled stranice. *HTML* se može opisati i kao jezik za estetsko uređenje internetske stranice iako to u modernom razvoju nije pravilo.¹²

¹²(Html.com, 2019)

Stilski jezik (eng. Cascading Style Sheets CSS)

CSS (eng. *Cascading Style Sheets*) je jezik kojim se definira dizajn stranice, koji je određen u *HTML* opisnom jeziku. CSS-om možemo definirati niz vizualnih atributa nekog elementa na stranici kao što su: veličina i stil fonta, boje teksta, pozadinske slike, pozicija elementa na stranici i slično.

```
48
49 .hiddenDugme {
50     visibility: hidden;
51 }
52
53
54 .sidenavDesni {
55     position: static;
56     padding-top: 20px;
57     background-color: #a6a6a6;
58     height: 1250px;
59 }
60
61
62 .glavniDioForme {
63     position: relative;
64     padding-top: 20px;
65     background-color: #ffffff;
66     height: 1000px;
67 }
68
69 .linkSideNavUredeno {
70     color: #333333;
71     font-size: 18px;
72 }
```

Slika 19. Primjer CSS datoteke. Izradio: autor

Velika prednost kod korištenja CSS jezika za uređenje internetske stranice je to što određene parametre definiramo jednom te taj parametar možemo proslijediti za nebrojeni broj elemenata. Najveća mana korištenja CSS-a je što ne radi jednako na svim internetskim preglednicima.¹³

¹³ (Devdocs.io, 2019)

Javascript

Uz *HTML* i *CSS* *JavaScript* je jedan od temeljnih karika u razvoju internetskih aplikacija. *JavaScript* nastao je 1995 godine, i namijenjen je za razvoj interaktivnih internetskih stranica te je podržan u svim internetskim preglednicima. *JavaScript* omogućuje programerima razvitak dinamičnih funkcionalnosti kod statičnih internetskih stranica. ¹⁴

Bootstrap

Bootstrap je besplatni alat (eng. *Open source*) namijenjen za dizajniranje internetskih aplikacija. Sastoji se od *HTML*, *CSS* i *JavaScript* predložaka za pomoć u izgradnji korisničkog sučelja. Jednostavan za korištenje i implementaciju, izuzev toga, olakšava razvojnim programerima tako što ne moraju kreirati većinu *CSS* klasa, već samo pojedine. ¹⁵

¹⁴ (Flanagan, 2011 str: 8-9)

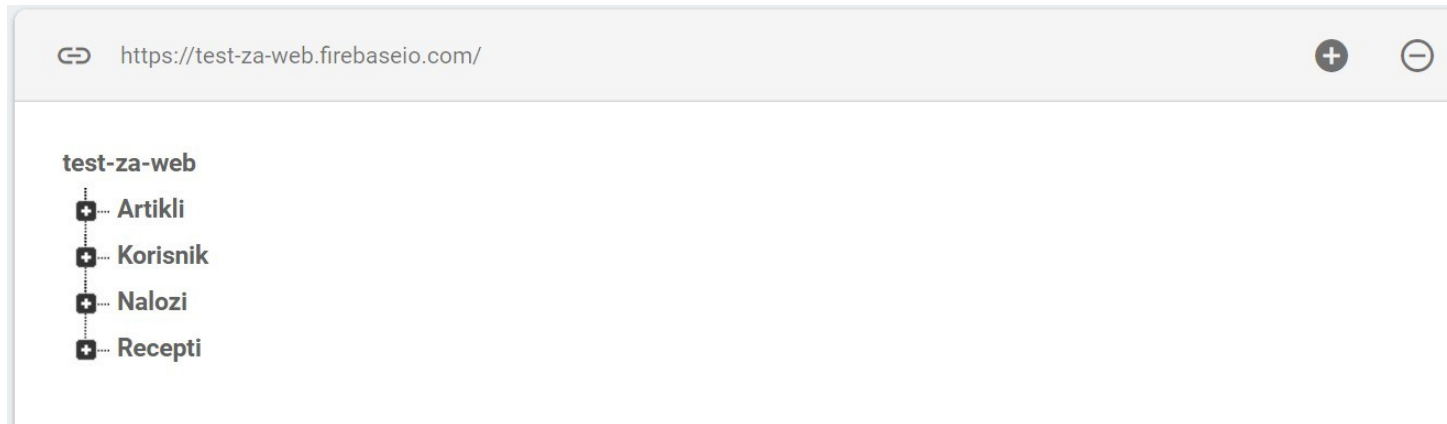
¹⁵ (Mark Otto, 2019)

Implementacija

Implementacija projekta započinje s analizom opisanog problema, analize korisničkih zahtjeva i analize dosadašnjeg procesa rada. Projekt se temelji na izradi *Firebase* relacijske baze podataka u stvarnom vremenu, projektiranju i izradi administratorske aplikacije odnosno web aplikacije te izradi klijentske aplikacije namijenjene mikser tehničaru koja je implementirana u *Android* studiju. Za korištenje internetske aplikacije potrebno je da računalo na kojem se koristi ima pristup internetskoj mreži i instaliran neki od internetskih preglednika kao što su *Google Chrome*, *Mozilla Firefox*, *Internet Explorer* ili *Opera*. Za korištenje *Android* aplikacije potrebno je imati pristup internetskoj mreži i *Android* uređaj operacijskog sustava *Marshmallow* 6.0 ili veći. Administratorska web aplikacija i *Android* aplikacija ne zahtijevaju pretjerane hardverske zahtjeve što dovodi do smanjenja troškova samog sustava.

Izrada baze podataka

Za izradu projekta korištena je nerelacijska *Firebase* baza podataka u stvarnom vremenu (eng. *NoSQL real-time database*). Prvi korak kod implementacije *Firebase* baze podataka u realnom vremenu je izrada čvorova u bazi koje će kasnije korisnik popunjavati potrebnim podacima.



Slika 20. Prikaz strukture čvorova u Firebase bazi podataka. Izradio: Autor

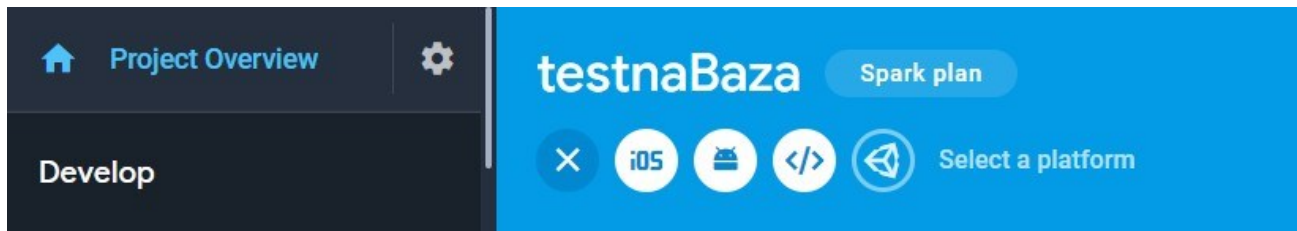


Slika 21. Prikaz strukture pod-čvorova u Firebase bazi podataka. Izradio: Autor

NoSQL baza je vođena dokumentima te zahtjeva od programera da uz ključ unese i neku vrijednost. Kod *SQL* baze podataka ta vrijednost može biti null, dakle omogućeno je hijerarhijski unositi čvorove ali u jednom trenutku uz ključ mora biti i definirana vrijednost.

Implementacija Firebase baze podataka s Web aplikacijom

Prije samog početka potrebna je postojana web aplikacija u nekom tekstualnom uređivaču (*Atom*, *Sublimetext* ili *Nodpad++*) koja ima uspostavljenu konekciju s internetskom mrežom. Nakon toga spajanje s bazom podataka odvija se u par koraka.



Slika 22. Odabir razvojne platforme. Izradio: autor.

Programer mora jasno definirati koju razvojnu platformu želi povezati s bazom. Kada odabere platformu, web razvoj programer dobije generirani isječak koda koji je potrebno kopirati u *HTML* datoteku.

```
<script src="https://www.gstatic.com/firebasejs/5.8.0/firebase.js"></script>
<script>
  // Initialize Firebase
  // TODO: Replace with your project's customized code snippet
  var config = {
    apiKey: "<API_KEY>",
    authDomain: "<PROJECT_ID>.firebaseapp.com",
    databaseURL: "https://<DATABASE_NAME>.firebaseio.com",
    projectId: "<PROJECT_ID>",
    storageBucket: "<BUCKET>.appspot.com",
    messagingSenderId: "<SENDER_ID>",
  };
  firebase.initializeApp(config);
</script>
```

Slika 23. Prikaz pravilnog postavljanja konekcije na bazu podataka. Preuzeto sa: <https://firebase.google.com/docs/web/setup>.

Korisničke upute

Unos novog djelatnika

Unos novog djelatnika omogućuje korisniku unos novog djelatnika u sistem. Novi djelatnik se unosi u formi unos novog djelatnika:

The screenshot shows a web application interface for adding a new user. The main heading is "Unos novog korisnika". The form contains the following fields and controls:

- Ime korisnika:** Text input field with placeholder "Unesite ime korisnika".
- Prezime korisnika:** Text input field with placeholder "Unesite prezime korisnika".
- Lozinka:** Text input field with placeholder "Unesite lozinku". Below it is a checkbox labeled "Pogledaj lozinku".
- Potvrdi lozinku:** Text input field with placeholder "Ponovi lozinku". Below it is a checkbox labeled "Pogledaj lozinku".
- OIB korisnika:** Text input field with placeholder "Unesite OIB korisnika".
- Broj mobitela korisnika:** Text input field with placeholder "Unesite ime korisnika".

At the bottom of the form are two buttons: a green "Potvrdi unos" button and a red "Odbaci unos" button.

On the left sidebar, there are two buttons: "Unos novog djelatnika" (highlighted) and "Pregled dosadašnjih djelatnika".

On the right, under the heading "Pomoć:", there is explanatory text:

- O unosu djelatnika:** * Modul djelatnici omogućuje unos novih korisnika sistema, svaki djelatnik se unosi po sljedećim navodima: Ime djelatnika, prezime djelatnika, Broj mobitela djelatnika (koji je ujedno i njegovo korisničko ime), lozinka djelatnika te OIB djelatnika.
- * Na pritisak dugmeta potvrdi unos, korisnik se sprema u bazu.
- * Na pritisak odbaci unos, polja za unos se brišu.

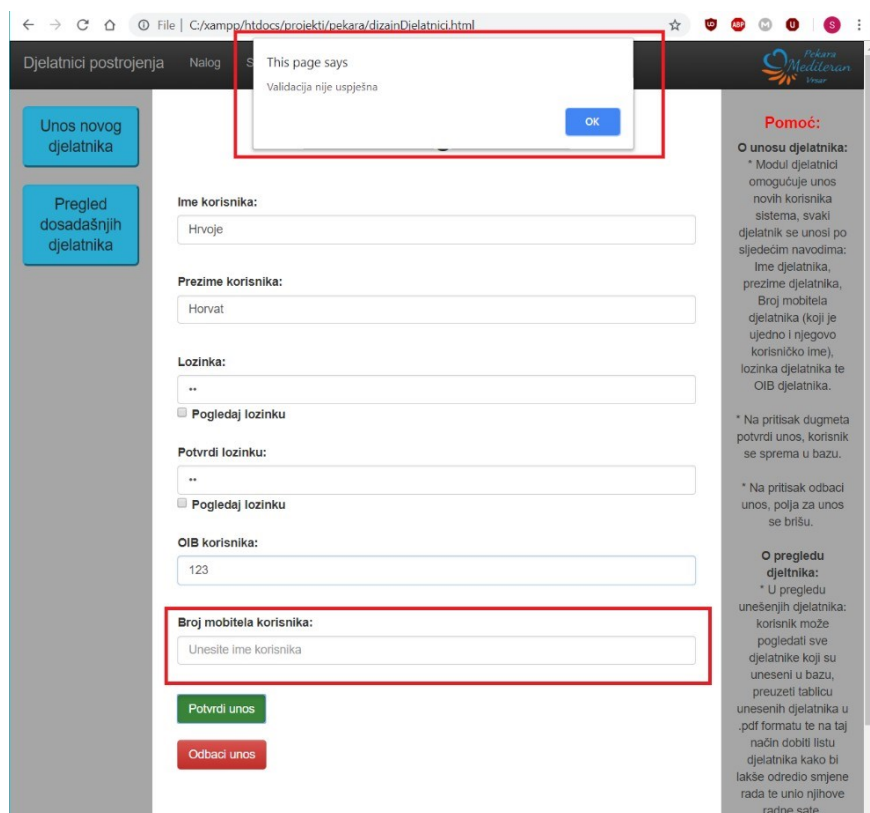
Below this is another section:

- O pregledu djelatnika:** * U pregledu unesenih djelatnika: korisnik može pogledati sve djelatnike koji su uneseni u bazu, preuzeti tablicu unesenih djelatnika u pdf formatu te na taj način dobiti listu djelatnika kako bi lakše odredio smjene

Slika 24. Prikaz forme unosa djelatnika. Izradio: autor.

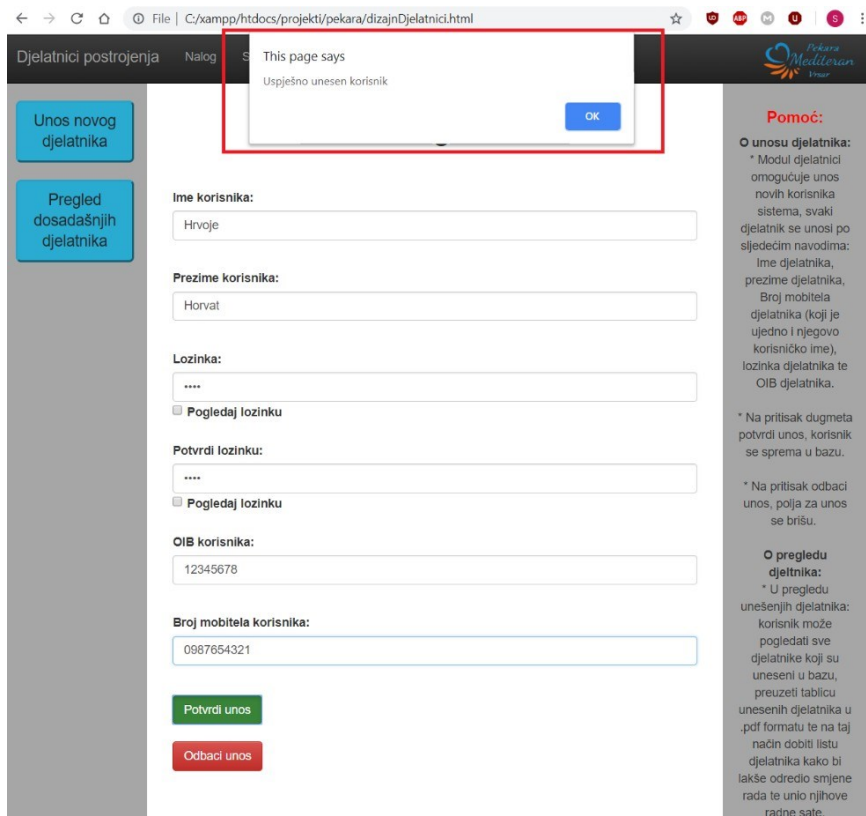
Svaki od polja u formi unosa novog djelatnika mora se popuniti kako bi validacija unesenih podataka bila uspješna.

Ukoliko neki od podataka koji su potrebni za unos nisu ispravni administrator dobije poruku pogreške.



Slika 25. Prikaz navazećeg unosa djelatnika. Izradio: autor.

Ukoliko su svi podaci pravilno uneseni, administrator dobije poruku uspješnog unosa. Tada može nastaviti sa daljnjim unosom djelatnika ili pregledom djelatnika.



Slika 26. Prikaz uspješnog unosa djelatnika. Izradio: autor.

Nakon unosa podataka djelatnika admin / voditelj može odabrati:

- Potvrđan unos
- Odbaci unos

Ukoliko administrator / korisnik odabere potvrđan unos, može ga urediti na pod-modulu Pregled dosadašnjih djelatnika.

Djelatnici postrojenja Nalog Skladište Recepture **Djelatnici**



Unos novog djelatnika

Pregled dosadašnjih djelatnika

Pregled unesenih djelatnika

IME	PREZIME	BROJ MOBITELA	OIB	Uredi
Sean	Hasanagić	0914605389	00000000000	Uredi

Pomoć:

O unosu djelatnika:
 * Modul djelatnici omogućuje unos novih korisnika sistema, svaki djelatnik se unosi po sljedećim navodima:
 Ime djelatnika,
 prezime djelatnika,
 Broj mobitela

Slika 27. Kratki prikaz unesenog djelatnika. Izradio: autor.

Prikaz pregleda unesenih djelatnika, na kojoj korisnik / admin može urediti unesenog korisnika na sljedeći način:

Detaljan pregled djelatnika

KORISNIK	DETALJNO
Ime	Sean
Prezime	Hasanagić
Broj Mobitela	0914605389
Oib	00000000000
ID:	1548029077667

Unesite novi broj mobitela korisnika:

Promjeni
Izbriši korisnika

Slika 28. Detaljan prikaz unesenog djelatnika. Izradio: autor.

Prikaz detaljnog pregleda djelatnika omogućuje administratoru da izvrši promjenu broja mobitela (ukoliko djelatnik ne koristi stari), tako što se unese broj novog telefonskog broja te klikom na promjeni korisniku se mijenja broj mobitela s kojim će se naknadno moći prijaviti na android uređaj. Naredna akcija je da korisnik / admin u potpunosti obriše korisnika.

Unos novog artikla

Skladište pekare Nalog **Skladište** Recepture Djelatnici

Unos novog artikla

Unos novog artikla

Pregled unesenih artikala

Unos novog artikla

Naziv artikla:

Odaberite jedinicu mjere:
Kilogram

Kada jednom odredite jedinicu mjere ne možete je kasnije mijenjati!

Unesite naziv dobavljača:

Unesite barkod proizvoda:

Ulazna cijena proizvoda:

Opis artikla:

Potvrdi unos

Odbaci unos

Pomoć:

Unosu artikala:

* Modul skladišta omogućuje korisniku unos artikala koji su mu potrebni za cjelokupan rad sistema. Korisnik mora unijeti artikl u skladište pod sljedećim parametrima : Naziv artikla, Naziv dobavljača artikla, Barkod artikla, Ulazna cijena artikla, Kratki opis artikla.

Važno: Mjernu jedinicu artikla ne možete naknadno mijenjati!

Nakon što je artikl unesen on dobija jedinstvenu šifru artikla.

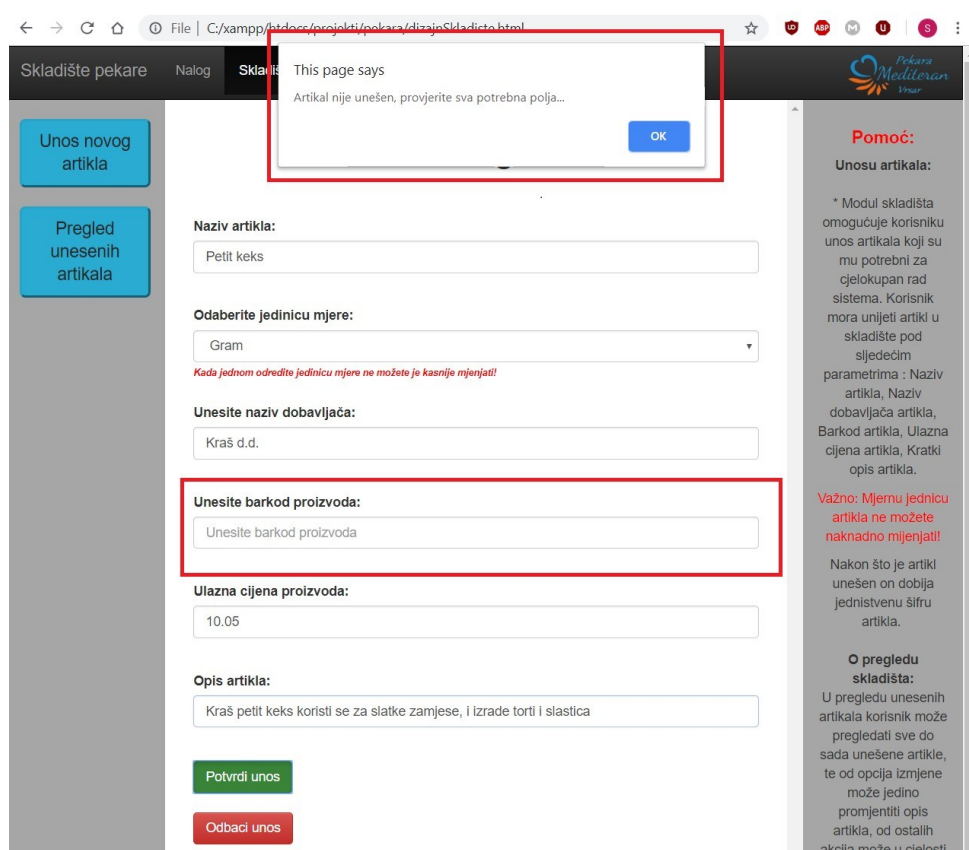
O pregledu skladišta:

U pregledu unesenih artikala korisnik može pregledati sve do sada unesene artikle, te od opcija izmjene može jedino promijeniti opis artikla, od ostalih akcija može u cjelosti obrisati artikal.

Slika 29. Prikaz forme za unosa artikla. Izradio: autor.

Modul skladišnog poslovanja omogućuje korisniku unos novog artikla. Svaki parametar određenog artikla mora se unijeti u bazu, a najvažnije je da korisnik unese mjernu jedinicu za svaki artikl jer se naknadno ne može mijenjati (moguće isključivo uz intervenciju razvojnog programera).

Ukoliko administrator ne unese sve potrebne podatke vezane za artikl, po potvrdi unosa artikla dobije poruku pogreške.



Slika 30. Prikaz neuspješnog unosa artikla. Izradio: autor.

The screenshot displays a web interface for a bakery's warehouse management system. At the top, a navigation bar includes 'Skladište pekare', 'Nalog', and 'Skladište'. A confirmation dialog box is open, displaying the message 'Uspješno unesen artikal!' (Article successfully added!) with an 'OK' button. Below the dialog, a form for adding a new article is visible. The form fields are: 'Naziv artikla:' (Petit keks), 'Odaberite jedinicu mjere:' (Gram), 'Unesite naziv dobavljača:' (Kraš d.d.), 'Unesite barkod proizvoda:' (00000111111), 'Ulazna cijena proizvoda:' (10.05), and 'Opis artikla:' (Kraš petit keks koristi se za slatke zamjese, i izrade torti i slastica). At the bottom of the form, there are two buttons: 'Potvrdi unos' (Confirm entry) and 'Odbaci unos' (Reject entry). On the right side, a 'Pomoć:' (Help) section provides instructions on how to use the warehouse module and lists the required parameters for adding an article.

Unos novog artikla

Pregled unesenih artikala

Naziv artikla:
Petit keks

Odaberite jedinicu mjere:
Gram

Kada jednom odredite jedinicu mjere ne možete je kasnije mijenjati!

Unesite naziv dobavljača:
Kraš d.d.

Unesite barkod proizvoda:
00000111111

Ulazna cijena proizvoda:
10.05

Opis artikla:
Kraš petit keks koristi se za slatke zamjese, i izrade torti i slastica

Potvrdi unos

Odbaci unos

Pomoć:
Unosu artikala:

* Modul skladišta omogućuje korisniku unos artikala koji su mu potrebni za cjelokupan rad sistema. Korisnik mora unijeti artikl u skladište pod sljedećim parametrima : Naziv artikla, Naziv dobavljača artikla, Barkod artikla, Ulazna cijena artikla, Kratki opis artikla.

Važno: Mjernu jedinicu artikla ne možete naknadno mijenjati!

Nakon što je artikl unesen on dobija jedinstvenu šifru artikla.

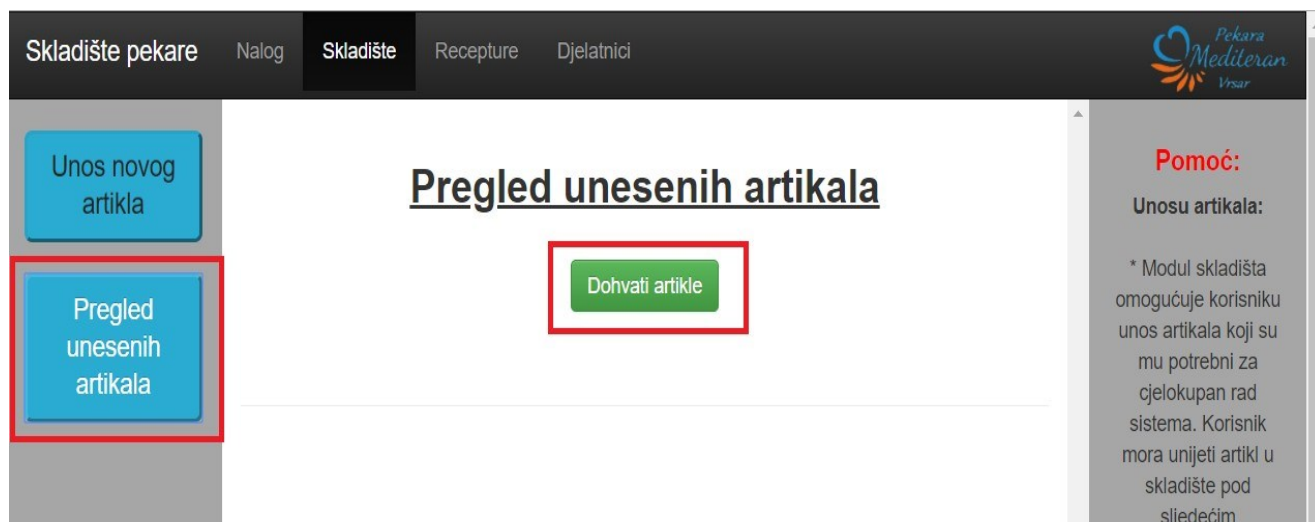
O pregledu skladišta:
U pregledu unesenih artikala korisnik može pregledati sve do sada unesene artikle, te od opcija izmjene može jedino promijeniti opis artikla, od ostalih akcija može u cjelosti obrisati artikal.

Slika 31. Prikaz uspješno unesenog artikla. Izradio: autor.

Nakon svih unesenih parametara korisnik / administrator mora odabrati da li će potvrditi unos ili ga odbaciti. Ukoliko ga potvrdi artikl ulazi u bazu. Ukoliko ga odbaci korisnik ima mogućnost unijeti novi artikl kojeg će potvrditi.

Nakon potvrde unosa novog artikla korisnik/ administrator ima omogućen pregled u modulu pregled unesenih artikala.

Pod pregledom unesenih artikala:




Slika 32. Prikaz dohvata forme detaljnog pregleda artikla. Izradio: autor.

Nakon pritiska na dugme „Dohvati artikle” administrator ima uvid u unesene artikle.

Administrator ima tablični pregled o artiklima: naziv artikla, naziv dobavljača te barkod.

Skladište pekare
Nalog
Skladište
Recepture
Djelatnici



Unos novog artikla

Pregled unesenih artikala

Pregled unesenih artikala

Dohvati artikle

NAZIV ARTIKLA	NAZIV DOBAVLJAČA	BARKOD	ŠIFRA ARTIKLA
Jaje	Koka d.o.o.	000111222	Uredi
Brašno TIP 550	Žito d.o.o.	00000000	Uredi
Vodu	Vodovod d.o.o.	12345678	Uredi
Sol	Solana Pag d.o.o.	1111111	Uredi
Kvasac	Pekarski dodatci d.o.o.	55555555	Uredi
Hit	Pekarski dodatci d.o.o.	44444444	Uredi
Kukuruzno brašno	Klas d.o.o.	7777777777	Uredi
Šećer	Konzum d.d.	3748612424	Uredi
Raženo brašno TIP 850	Klas d.o.o.	1111111111	Uredi
Ludvig brašno	klas d.o.o.	222222222	Uredi
Rogenmax	Žito do.o.	333333333	Uredi
Jasko smjesa	Mediteran d.o.o.	55555555	Uredi
Naturin	Mediteran d.o.o.	666666666	Uredi
Petit keks	Kraš d.d.	0000011111	Uredi

Pomoć:

Unosu artikala:

* Modul skladišta omogućuje korisniku unos artikala koji su mu potrebni za cjelokupan rad sistema. Korisnik mora unijeti artikl u skladište pod sljedećim parametrima : Naziv artikla, Naziv dobavljača artikla, Barkod artikla, Ulazna cijena artikla, Kratki opis artikla.

Važno: Mjernu jednicu artikla ne možete naknadno mijenjati!

Nakon što je artikl unesen on dobija jedinstvenu šifru artikla.

O pregledu skladišta:

U pregledu unesenih artikala korisnik može pregledati sve do sada unesene artikle, te od opcija izmjene može jedino promijeniti opis artikla, od ostalih akcija može u cjelosti obrisati artikal.

Slika 33. Tablični prikaz općeg pregleda artikla. Izradio: autor.

Na klik uredi artikl, administrator ima uvid u dodatne mogućnosti artikla:

Na modulu pregled unesenih artikala korisnik može promijeniti opis artikla.

Detaljan pregled artikla

KORISNIK	DETALJNO
Naziv artikla:	Petit keks
Naziv dobavljača:	Kraš d.d.
Jedinica mjere:	GR
Ulazna cijena artikla:	10.05
Barkod:	00000111111
Opis artikla:	Koristi se isključivo za ukrašavanje torti, te zamjes određenih slastica
Jedinstvena šifra artikla:	1550880904683

Promjeni opis artiklaIzbrisi artikal

Opis artikla:
Važno: Mjernu jednicu artikla ne možete naknadno mijenjati!

Nakon što je artikl unesen on dobija jedinstvenu šifru artikla.

O pregledu skladišta:
U pregledu unesenih artikala korisnik može pregledati sve do sada unesene artikle, te od opcija izmjene može jedino promjeniti opis artikla, od ostalih akcija može u cjelosti obrisati artikal.

Slika 34. Detaljan prikaz određenog artikla. Izradio: autor.

Nakon potvrde o promjeni opisa artikla, administrator dobije poruku upozorenja.

The screenshot shows a web browser window with a URL starting with 'File | C:/xampp/htdocs/projekti/pokara/dizajnSkladište.html'. A warning dialog box is displayed in the center, with the text 'This page says' and 'Updajitali ste korisnika' (You have updated the user). An 'OK' button is visible in the dialog. Below the dialog is a table of items with columns for name, company, and ID, and an 'Uredi' (Edit) button for each row. The items listed are: Raženo brašno TIP 850 (Klas d.o.o., 11111111111), Ludvig brašno (klas d.o.o., 22222222222), Rogenmax (Žito do.o., 33333333333), Jasko smjesa (Mediteran d.o.o., 555555555), Naturin (Mediteran d.o.o., 66666666666), and Petit keks (Kraš d.d., 00000111111). Below the table is a section titled 'Detaljan pregled artikla' (Detailed view of the item) with a table of details for 'Petit keks'. The details table has two columns: 'KORISNIK' and 'DETALJNO'. The details are: Naziv artikla: Petit keks; Naziv dobavljača: Kraš d.d.; Jedinica mjere: GR; Ulazna cijena artikla: 10.05; Barkod: 00000111111; Opis artikla: Kraš petit keks koristi se za slatke zamjese, i izrade torti i slastica; Jedinствена šifra artikla: 1550880904683. At the bottom of the details section, there is a text input field containing 'Koristi se isključivo za ukrašavanje torti' and two buttons: 'Promjeni opis artikla' (Change item description) and 'Izbrisi artikla' (Delete item). The 'Promjeni opis artikla' button is highlighted with a red box. On the right side of the page, there is a sidebar with a 'Pomoć:' (Help) section and a 'Unos artikala:' (Item entry) section. The 'Unos artikala:' section contains a warning message: '* Modul skladišta omogućuje korisniku unos artikala koji su mu potrebni za cjelokupan rad sistema. Korisnik mora unijeti artikl u skladište pod sljedećim parametrima : Naziv artikla, Naziv dobavljača artikla, Barkod artikla, Ulazna cijena artikla, Kratki opis artikla.' Below this is another warning: 'Važno: Mjernu jedinicu artikla ne možete naknadno mijenjati!' (Important: You cannot change the unit of measurement of the item later!). The final paragraph in the sidebar states: 'Nakon što je artikl unesen on dobija jedinstvenu šifru artikla.' (After the item is entered, it receives a unique item ID). Below this is a section titled 'O pregledu skladišta:' (About warehouse view) which says: 'U pregledu unesenih artikala korisnik može pregledati sve do sada unesene artikle, te od opcija izmjene može jedino promijeniti opis artikla, od ostalih akcija može u cjelosti obrisati artikla.' (In the view of entered items, the user can view all items entered so far, and from the options of change, it can only change the description of the item, from the other actions it can delete the item completely).

KORISNIK	DETALJNO
Naziv artikla:	Petit keks
Naziv dobavljača:	Kraš d.d.
Jedinica mjere:	GR
Ulazna cijena artikla:	10.05
Barkod:	00000111111
Opis artikla:	Kraš petit keks koristi se za slatke zamjese, i izrade torti i slastica
Jedinствена šifra artikla:	1550880904683

Slika 35. Prikaz upozorenja administratoru o promijenjenom opisu artikla. Izradio: autor.

Osim promjene opisa artikla, administrator može obrisati određeni artikl ukoliko se više ne koristi. Nakon potvrde da se artikl obriše administrator dobije upozorenje o uspješnosti obrisanog artikla.

The screenshot displays a web interface for managing bakery items. On the left, there are buttons for 'Unos novog artikla' and 'Pregled unesenih artikala'. The main area shows a table of items with columns for name, company, and barcode, each with an 'Uredi' button. A red box highlights a confirmation dialog that says 'Uspješno ste obrisali korisnika' with an 'OK' button. Below the table is a 'Detaljan pregled artikla' section with a table of details for 'Petit keks'. At the bottom, there is a text input field for 'Unesite novi opis artikla' and two buttons: 'Promjeni opis artikla' and 'Izbriši artikal', with the latter highlighted by a red box. On the right, a 'Pomoć' sidebar provides instructions on how to use the system.

KORISNIK	DETALJNO
Naziv artikla:	Petit keks
Naziv dobavljača:	Kraš d.d.
Jedinica mjere:	GR
Ulazna cijena artikla:	10.05
Barkod:	00000111111
Opis artikla:	Koristi se isključivo za ukrašavanje torti
Jedinstvena šifra artikla:	1550882283647

Pomoć:
Unosu artikala:
 * Modul skladišta omogućuje korisniku unos artikala koji su mu potrebni za cjelokupan rad sistema. Korisnik mora unijeti artikl u skladište pod sljedećim parametrima : Naziv artikla, Naziv dobavljača artikla, Barkod artikla, Ulazna cijena artikla, Kratki opis artikla.
Važno: Mjernu jedinicu artikla ne možete naknadno mijenjati!
 Nakon što je artikl unesen on dobija jedinstvenu šifru artikla.
O pregledu skladišta:
 U pregledu unesenih artikala korisnik može pregledati sve do sada unesene artikle, te od opcija izmjene može jedino promjeniti opis artikla, od ostalih akcija može u cjelosti obrisati artikal.

Slika 36. Prikaz forme za brisanje artikla. Izradio: autor.

Važna napomena: Ukoliko administrator izbriše artikl koji će koristiti u nekom receptu, taj recept neće biti potpun, odnosno biti će ispisan bez tog (obrisanog) artikla.

Unos nove recepture

Recepture Nalog Skladište **Recepture** Djelatnici

Pečana Mediteran Vrsar

Unos nove recepture

Naziv recepture:

Unesite naziv nove recepture

Broj artikala u zamjesu	Odaberite naziv artikala u zamjesu	Unesite količinu artikala u zamjesu	Obriši
1	Brašno tip 500 KG		Izbrisi stavku

Dodaj novi artikal u zamjes Spremi recepturu

Pomoć:
Unosu recepture:
* Nakon što korisnik unese sve potrebne artikle koje koristi u svojim recepturama, može započeti unos receptura odnosno zamjesa. U modulu recepture važno je dodati naziv recepture, te iz padajućeg izbornika dodati sve artikle i njihove jedinice mjera te njihov udio u receptu po jednom komadu. Nakon unosa recepture

Slika 37. Prikaz forme za brisanje artikla. Izradio: autor.

Izgled modula za unos novog recepta.

Za unos novog recepta korisnik / administrator je dužan unijeti naziv recepture, i nakon toga odabrati odgovarajuće artikle u padajućem izborniku u zamjesu, zatim definirati njihove količine po jedinici mjere komad/količina.

Nakon potvrde recepture korisnik / administrator ima pravo izbrisati stavku iz recepta ili dodati novu stavku u recept.

Recepture Nalog Skladište **Recepture** Djelatnici

Unos nove recepture

Naziv recepture:

Kruh bijeli 600gr

Broj artikala u zamjesu	Odaberite naziv artikala u zamjesu	Unesite količinu artikla u zamjesu	Obrisi
1	Brašno TIP 550 KG	0.375	Izbrisi stavku
2	Vodu L	0.21	Izbrisi stavku
3	Sol KG	0.0075	Izbrisi stavku
4	Kvasac KG	0.011	Izbrisi stavku
5	Hit KG	0.0018	Izbrisi stavku

Dodaj novi artikal u zamjes Spremi recepturu

Pomoć:
Unosu recepture:
* Nakon što korisnik unese sve potrebne artikle koje koristi u svojim recepturama, može započeti unos receptura odnosno zamjesa. U modulu recepture važno je dodati naziv recepture, te iz padajućeg izbornika dodati sve artikle i njihove jedinice mjera te njihov udio u receptu po jednom komadu. Nakon unosa recepture korisnik je pregledati te ukoliko je napravio grešku izbrisati u cjelosti, te unijeti ponovno.
O pregledu receptura:
* Nakon unosa recepture korisnik je pregledati te ukoliko

Slika 38. Prikaz forme za unos recepture. Izradio: autor.

Nakon što su sve stavke zamjesa unesene s valjanim mjerama administrator potvrđuje unos nove recepture, i dobije poruku o uspješnom unosu.

Na klik podmodula Pregled dosadašnjih recepata korisnik / administrator ima uvid u sastav recepture.

This page says
Uspješno unesen recept

Naziv recepture:
Kruh bijeli 600gr

Broj artikala u zamjesu	Odaberite naziv artikala u zamjesu	Unesite količinu artikla u zamjesu	Obrisi
1	Brašno TIP 550 KG	0.375	Izbrisi stavku
2	Vodu L	0.21	Izbrisi stavku
3	Sol KG	0.0075	Izbrisi stavku
4	Kvasac KG	0.011	Izbrisi stavku
5	Hlit KG	0.0018	Izbrisi stavku

Dodaj novi artikal u zamjes Spremi recepturu

Slika 39. Prikaz načina spremanja recepture i poruke uspjehnosti. Izradio: autor.

Na klik dohvati recepte korisnik / administrator ima uvid o nazivu unesenih receptura.


Unos nove recepture

Naziv recepture:
Kruh bijeli 600gr

Broj artikala u zamjesu	Odaberite naziv artikala u zamjesu	Unesite količinu artikla u zamjesu	Obrisi
1	Brašno TIP 550 KG	0.375	Izbrisi stavku
2	Vodu L	0.21	Izbrisi stavku
3	Sol KG	0.0075	Izbrisi stavku
4	Kvasac KG	0.011	Izbrisi stavku
5	Hlit KG	0.0018	Izbrisi stavku

Slika 40. Prikaz forme unosa djelatnika. Izradio: autor.

Na klik pokaži, administrator ima uvid o detaljima recepta.

Recepture Nalog Skladište **Recepture** Djelatnici 

Unos novog recepta

Pregled dosadašnjih recepta

Pregled unesenih recepata

Dohvati Recepte

NAZIV	POKAŽI
Kruh bijeli	Pokazi
Kukuruzni kruh 450gr	Pokazi
Kukuruzni Baguette 300gr	Pokazi
Kukuruzna ciabatta 200gr	Pokazi
Ludvig 1 KG	Pokazi
Raženo 550gr	Pokazi
Bakin kruh 450gr	Pokazi
Bakin baguette 300gr	Pokazi
Kruh bijeli 600gr	Pokazi

Pomoć:

Unosu recepture:

* Nakon što korisnik unese sve potrebne artikle koje koristi u svojim recepturama, može započeti unos receptura odnosno zamjesa. U modulu recepture važno je dodati naziv recepture, te iz padajućeg izbornika dodati sve artikle i njihove jedinice mjera te njihov udio u receptu po jednom komadu. Nakon unosa recepture korisnik je pregledati te ukoliko je napravio grešku izbrisati u cjelosti, te unijeti ponovno.

O pregledu receptura:

* Nakon unosa recepture korisnik je pregledati te ukoliko je napravio grešku izbrisati u cjelosti, te unijeti ponovno.

Slika 41. Tablični prikaz unesenih receptura. Izradio: autor.

Kruh bijeli 600gr		Pokazi
-------------------	--	------------------------

RECEPT	DETALJNO
Ime	Kruh bijeli 600gr
ID:	1550882992022
Sastojci:	
Brašno TIP 550	0.375 KG
Vodu	0.21 L
Sol	0.0075 KG
Kvasac	0.011 KG
Hit	0.0018 KG

[Izbriši Recept](#)

* Nakon unosa recepture korisnik je pregledati te ukoliko je napravio grešku izbrisati u cjelosti, te unijeti ponovno.

Slika 42. Detaljan prikaz unesenih receptura. Izradio: autor.

This page says
Uspješno ste obrisali recept

OK

Kukuruzna ciabatta 200gr	Pokazi
Ludvig 1 KG	Pokazi
Raženo 550gr	Pokazi
Bakin kruh 450gr	Pokazi
Bakin baguette 300gr	Pokazi
Kruh bijeli 600gr	Pokazi

RECEPT	DETALJNO
Ime	Kruh bijeli 600gr
ID:	1550882992022
Sastojci:	
Brašno TIP 550	0.375 KG
Vodu	0.21 L
Sol	0.0075 KG
Kvasac	0.011 KG
Hit	0.0018 KG

Izbriši Recept

zamjesa. U modulu recepture važno je dodati naziv recepture, te iz padajućeg izbornika dodati sve artikle i njihove jedinice mjera te njihov udio u receptu po jednom komadu. Nakon unosa recepture korisnik je pregledati te ukoliko je napravio grešku izbrisati u cjelosti, te unijeti ponovno.

O pregledu receptura:

* Nakon unosa recepture korisnik je pregledati te ukoliko je napravio grešku izbrisati u cjelosti, te unijeti ponovno.

Slika 43. Brisanje određenog recepta. Izradio: autor.

Nakon klika na dugme pokaži, administrator osim detaljnog pregleda zamjesa ima mogućnost trajnog brisanja zamjesa. Nakon što administrator izbriše zamjes, dobije poruku upozorenja da je zamjes trajno obrisao.

Unos novog naloga

Nakon postavljanja svih parametara, korisnik / administrator može započeti s unosom naloga.

Broj recepata u nalogu	Odaberite recept	Unesite količinu izrade recepta	Opcije
1	Kruh bijeli		Izbrisi

Pomoć:
Unosu naloga:
* Nakon pravilno unšenih artikala te recepata, modul naloga služi ujedno kao provjera te izdavanje naloga na tabletni uređaj (mobilni uređaj). Korisnik odabere naziv zamjesa te unese količinu po jedinici mjere istog.
O pregledu naloga:
* Nakon unosa korisnik može pregledati uneseni

Slika 44. Prikaz forme unosa naloga. Izradio: autor.

Broj recepata u nalogu	Odaberite recept	Unesite količinu izrade recepta	Opcije
1	Kruh bijeli	600	Izbrisi
2	Ludvig 1 KG	120	Izbrisi
3	Raženo 550gr	400	Izbrisi
4	Kukuruzni Baguette 30	250	Izbrisi
5	Kukuruzni kruh 450gr	140	Izbrisi
6	Bakın baguette 300gr	30	Izbrisi

Pomoć:
Unosu naloga:
* Nakon pravilno unšenih artikala te recepata, modul naloga služi ujedno kao provjera te izdavanje naloga na tabletni uređaj (mobilni uređaj). Korisnik odabere naziv zamjesa te unese količinu po jedinici mjere istog.
O pregledu naloga:
* Nakon unosa korisnik može pregledati uneseni nalog, te potrošnu količinu materijala za određeni nalog.

Slika 45. Prikaz uspješno unesenog naloga. Izradio: autor.

Prikaz naloga:

The screenshot shows the 'Pregled unesenih naloga' interface. At the top, there is a navigation bar with 'Nalog' highlighted. Below it, a sidebar contains two blue buttons: 'Unos novog naloga' and 'Pregled dosadašnjih naloga'. The main content area is titled 'Pregled unesenih naloga' and includes a green 'Dohvati naloge' button. A table displays two orders with their IDs and 'Pokaži' buttons. The right sidebar contains a 'Pomoć' section with instructions for submitting and viewing orders.

BROJ NALOGA	POKAŽI
1550780732681	Pokaži
1550883944466	Pokaži

Slika 46. Prikaz forme unesenih naloga. Izradio: autor.

Nakon potvrde naloga administrator dobije poruku o uspješno unesenom nalogu. Nakon dodanih recepata u nalog, korisnik / administrator ima omogućen pregled pojedinačnih količina iz zamjesa. U ovom modulu voditelj / administrator može provjeriti da li je unos recepta valjan.

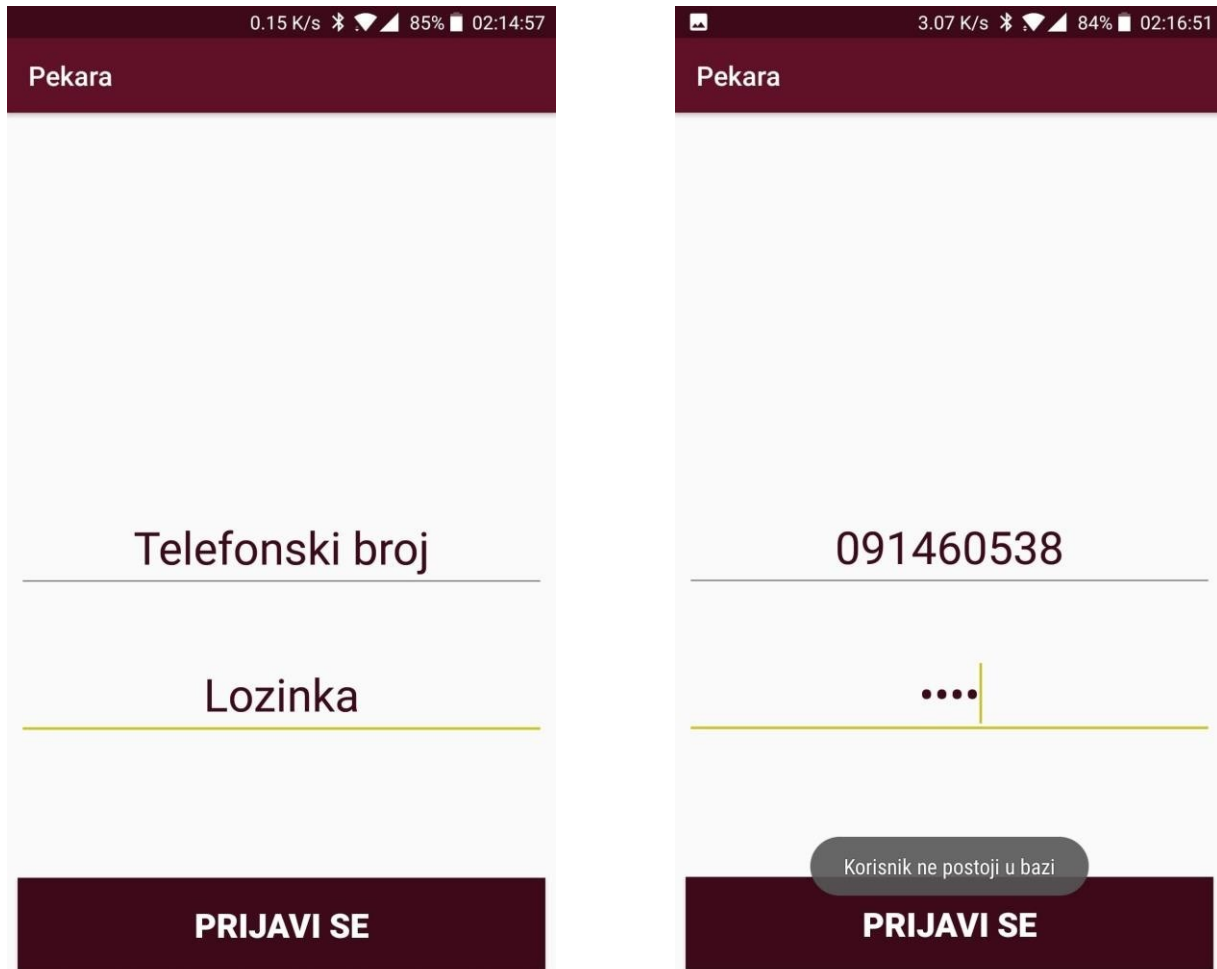
1550883944466		Pokazi	
RECEPT		DETALJNO	
ID:		1550883944466	
Recepti:			
Kruh bijeli		600	
Sastojci:			
Brašno TIP 550	0.375KG	*600=225.00KG	
Vodu	0.21L	*600=126.00L	
Sol	0.0075KG	*600=4.50KG	
Kvasac	0.011KG	*600=6.60KG	
Hit	0.0018KG	*600=1.08KG	
Kukuruzni kruh 450gr		140	
Sastojci:			
Brašno TIP 550	0.21KG	*140=29.40KG	
Vodu	0.15L	*140=21.00L	
Kvasac	0.006KG	*140=0.84KG	
Kukuruzno brašno	0.09KG	*140=12.60KG	
Kukuruzni Baguette 300gr		250	
Sastojci:			
Brašno TIP 550	0.14KG	*250=35.00KG	
Vodu	0.1L	*250=25.00L	
Kvasac	0.006KG	*250=1.50KG	
Kukuruzno brašno	0.06KG	*250=15.00KG	

Slika 47. Detaljan prikaz naloga. Izradio: autor.

Android aplikacija

Prijava korisnika

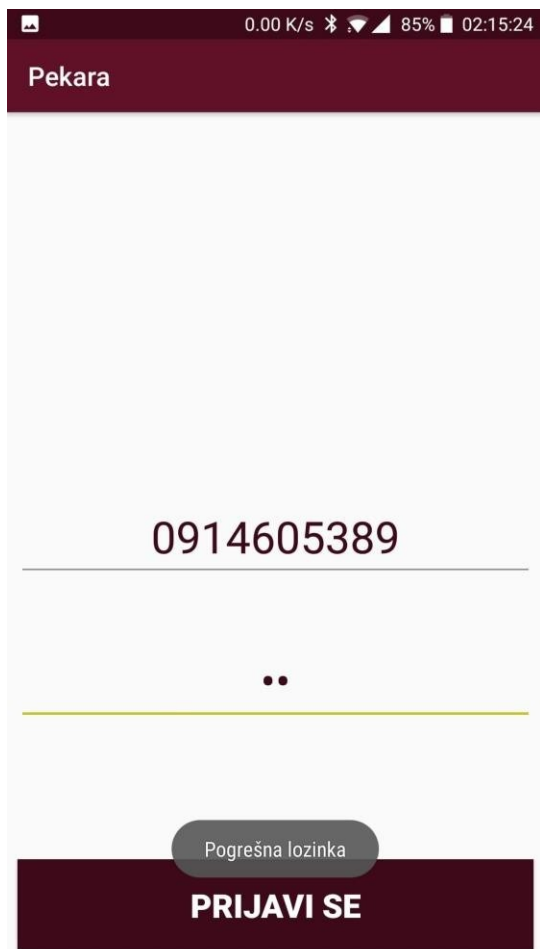
Početni prikaz ekrana na strani korisnika po svakom pokretanju traži potrebne korisničke podatke koji su definirani od strane administratora / voditelja poslovnice. Korisnik se prijavljuje sa svojim telefonskim brojem i sa svojom lozinkom koju mu je dodijelio administrator.



Slika 48. Prikaz forme unosa telefonskog broja i lozinke korisnika. Izradio: autor.

Slika 49. Prikaz greške kod unosa telefonskog broja. Izradio: autor.

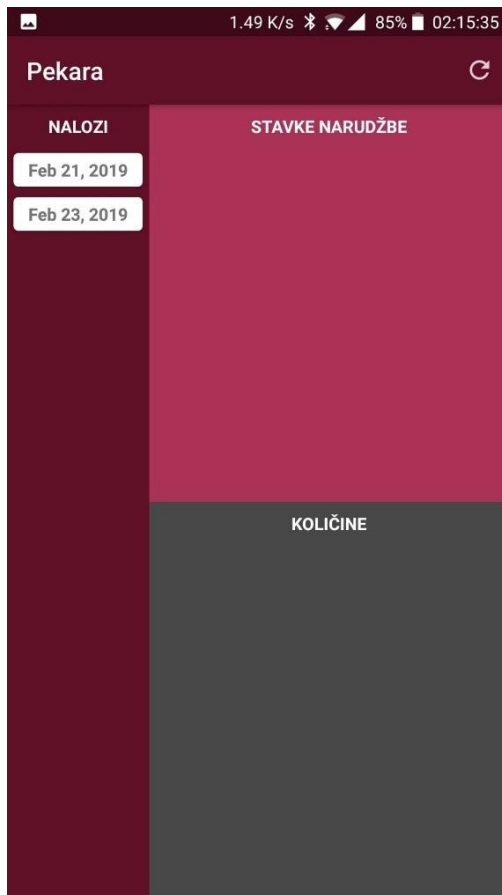
Ukoliko broj telefona ne postoji u bazi, korisnik dobije poruku greške.



Slika 50. Prikaz greške kod unosa lozinke korisnika. Izradio: autor.

Ukoliko je korisnik unesen od strane administratora, a unesena lozinka nije ispravna, korisnik dobije slijedeću poruku greške.

Pregled naloga



Slika 51. Prikaz aktivnosti naloga po datumu. Izradio: autor.



Slika 52. Prikaz stavki odabranog naloga. Izradio: autor.

Nakon što se korisnik uspješno prijavio, ulazi u aktivnost naloga, te u krajnjem lijevom stupcu ima uvid u naloge po datumu izdavanja. Po kliku na dati nalog korisnik ima uvid u stavke datog naloga.

Na klik recepture, korisnik ima uvid u stavku naloga pomnoženu s količinom iz naloga.

NALOZI		STAVKE NARUDŽBE	
Feb 21, 2019		Kruh bijeli	600.0
Feb 23, 2019		Kukuruzni kruh 450gr	140.0
		Kukuruzni Baguette 300g	250.0
		Ludvig 1 KG	120.0
		Raženo 550gr	400.0
		Bakin baguette 300gr	30.0
		KOLIČINE	
		Brašno TIP 550	225 KG
		Vodu	126 L
		Sol	4.5 KG
		Kvasac	6.6 KG
		Hit	1.08 KG

NALOZI		STAVKE NARUDŽBE	
Feb 21, 2019		Kruh bijeli	600.0
Feb 23, 2019		Kukuruzni kruh 450gr	140.0
		Kukuruzni Baguette 300g	250.0
		Ludvig 1 KG	120.0
		Raženo 550gr	400.0
		Bakin baguette 300gr	30.0
		KOLIČINE	
		Brašno TIP 550	126 KG
		Vodu	90 L
		Kvasac	3.6 KG
		Kukuruzno brašno	54 KG

Slika 53. Prikaz detaljnih količina artikala u zamjesu iz datog naloga. Izradio: autor.

Slika 54. Prikaz detaljnih količina artikala u zamjesu iz datog naloga. Izradio: autor.

NALOZI		STAVKE NARUDŽBE	
Feb 21, 2019		Kruh bijeli	600.0
Feb 23, 2019		Kukuruzni kruh 450gr	140.0
		Kukuruzni Baguette 300g	250.0
		Ludvig 1 KG	120.0
		Raženo 550gr	400.0
		Bakin baguette 300gr	30.0
		KOLIČINE	
		Brašno TIP 550	84 KG
		Vodu	60 L
		Sol	0.84 KG
		Kvasac	3.6 KG
		Hit	0.42 KG
		Jasko smjesa	36 KG
		Naturin	3.36 KG

NALOZI		STAVKE NARUDŽBE	
Feb 21, 2019		Kruh bijeli	600.0
Feb 23, 2019		Kukuruzni kruh 450gr	140.0
		Kukuruzni Baguette 300g	250.0
		Ludvig 1 KG	120.0
		Raženo 550gr	400.0
		Bakin baguette 300gr	30.0
		KOLIČINE	
		Vodu	246 L
		Kvasac	6.6 KG
		Hit	1.65 KG
		Raženo brašno TIP 850	165 KG
		Ludvig brašno	165 KG

Slika 55. Prikaz detaljnih količina artikala u zamjesu iz datog naloga. Izradio: autor.

Slika 56. Prikaz detaljnih količina artikala u zamjesu iz datog naloga. Izradio: autor.

Zaključak

Ovaj diplomski rad sastoji se od teorijskog dijela gdje se analizira razlika između *Firestore* baze podataka u realnom vremenu i standardnih baza podataka kao što je *SQL Oracle* relacijska baza podataka. Većina *NoSQL* baza podataka kao što je *Firestore* baza podataka imaju prednosti sa aspekta performansi kao što su jednostavnost implementacije u bilo koji projekt, jednostavno sigurnosno kopiranje podataka i sl. *Firestore* baza podataka u realnom vremenu kao i njena poboljšana verzija koja je još uvijek u razvoju *Cloud Firestore* su relativno nove tehnologije te nemaju kvalitetno teorijsko uporište i mnoštvo primjera korištenja koji su dostupni razvojnim programerima, za razliku od *SQL* relacijskih baza podataka koje se koriste duži niz godina. Za razliku od relacijskih baza podataka *Firestore* baza podataka u realnom vremenu nema definirana pravila modeliranja podataka. Svaki projekt ovisi o načinu inicijalnog strukturiranja podataka što može dovesti do zabune ukoliko na projektu radi više razvojnih programera. Nedefinirana pravila strukturiranja podataka dovode do problema kod postavljanja upita, filtriranja podataka kod pregleda, te osiguravanja i analize postojećih podataka. Bez obzira na postojeće nedostatke *Firestore* baza podataka u realnom vremenu idealna je za jednostavna aplikacijska rješenja. Drugi dio ovog diplomskog rada realizira praktičnu primjenu *Firestore* baze podataka u realnom vremenu na izradi aplikacije za vođenje pekarske proizvodne linije. Dobiveni korisnički zahtjevi i etnografski način analize poslovanja, doveli su do izrade mobilne i *web* aplikacije za pomoć u proizvodnji pekarskih proizvoda. Ovo aplikacijsko rješenje je predefinjirano, što znači da očekuje od korisnika da definira sve informacije potrebne za siguran i kvalitetan radni proces. Takav način izrade aplikacijskog rješenja omogućuje implementaciju kako u drugim pekarskim proizvodnim linijama, tako i u svim sferama poslovanja koje koriste neku vrstu zamjesa bilo to zamjes metalnih legura, zamjes mesnih prerađevina te zamjes određene marke betona u betonarama i slično. To bi ponajviše ubrzalo proces izrade naloga ili narudžbi određenih artikala, smanjilo bi količinu škartirane robe te u konačnici dovelo do uštede vremena, ljudstva i materijalnih dobara. Jedna od praktičnih mana korištenja *Firestore* baze podataka u realnom vremenu je ta što razvojni programer u potpunosti ovisi o *Google*-ovim odlukama koje se često mijenjaju za razliku od tradicionalnih relacijsko orjentiranih baza podataka.

Literatura

Popis knjiga

1. Oracle Database SQL Reference, 10g Release 2 (10.2), Primarni autor: Diana Lorentz izdano 2005 godine. Oracle (str: 30,31,32,33,35 Referenca se koristi i za fus notu broj 7.)
2. Fehily, C. (Izdano: 2005). *SQL*. Berkeley, Calif.: Peachpit.
3. DiMarzio, J. (n.d.). *Beginning Android programming with Android Studio*.
4. Flanagan, D. (2011). *JavaScript*. Sebastopol, CA: O'Reilly Media.

Popis internetskih izvora

1. Firebase. (2019). *Firestore Realtime Database* | *Firestore Realtime Database* | *Firestore*. [online] Dostupno na: <https://firebase.google.com/docs/database/> [Pristupljeno 23 Siječnja 2019].
2. Json.org. (2019). *JSON*. [online] Dostupno na: <https://www.json.org/> [Pristupljeno 26 Siječnja 2019].
3. Firebase. (2019). *Firestore*. [online] Available at: <https://firebase.google.com/pricing/> [Pristupljeno 27 Siječnja 2019].
4. Anon, (2019). [online] Dostupno na: <https://firebase.google.com/docs/database/rtdb-vs-firestore> [Pristupljeno 26 Siječnja 2019].
5. Firebase. (2019). *Choose a Database: Cloud Firestore or Realtime Database* | *Firestore*. [online] Dostupno na: <https://firebase.google.com/docs/database/rtdb-vs-firestore> [Pristupljeno 28 Siječnja 2019].
6. Db-engines.com. (2019). *Firestore Realtime Database vs. Oracle Comparison*. [online] Dostupno na: <https://db-engines.com/en/system/Firebase+Realtime+Database%3BOracle> [Pristupljeno 1 Veljače 2019].
7. Android Developers. (2019). *Android Studio release notes* | *Android Developers*. [online] Dostupno na: <https://developer.android.com/studio/releases/> [Pristupljeno 3 Veljače 2019].
8. Html.com. (2019). *Home*. [online] Dostupno na: <https://html.com/> [Pristupljeno 5 Veljače 2019].
9. Devdocs.io. (2019). *DevDocs — CSS documentation*. [online] Dostupno na: <https://devdocs.io/css/> [Pristupljeno 6 Veljače 2019].
10. Mark Otto, a. (2019). *Bootstrap*. [online] Getbootstrap.com. Dostupno na: <https://getbootstrap.com/> [Pristupljeno 9 Veljače 2019].

Popis slika

Slika 1. prikaz taba pravila u konzoli firebase database. Izradio: autor.

Slika 2. prikaz podrazumijevanih pravila u konzoli firebase database. Izradio: autor.

Slika 3. prikaz alata za simulacije pravila u Firebase konzloi. Izradio: autor.

Slika 4. Prikaz rada alata za simuliranje pravila u firebase bazi podataka. Izradio: autor.

Slika 5. prikaz postavljanja pravila za razvoj. Izradio: autor.

Slika 6. prikaz postavljanja pravila za razvoj, i indeksiranje po čvoru. Izradio: autor.

Slika 7. Prikaz strukture objekta u JSON formi. Preuzeto sa: <https://www.json.org/>.

Slika 8. Registracija android aplikacije. Izradio: autor.

Slika 9. Dodavanje JSNO datoteke. Izradio: autor.

Slika 10. Registracija android aplikacije. Izradio: autor.

Slika 11. Prikaz relacijske tablice država. Preuzeto sa: <https://demo.phpmyadmin.net/master-config/sql.php?server=2&db=world&table=Country&pos=0>

Slika 12. Prikaz tipova podataka u SQL-u. Preuzeto sa: <https://www.brentozar.com/sql-server-access-data-types/>

Slika 13. Shematski prikaz pekarske proizvodne linije. Izradio: autor

Slika 14. Dijagram slučajeva korištenja (eng. Use case diagram). Izradio: Autor

Slika 15. Sekvencijski dijagram (eng. Sequence diagram). Izradio: Autor

Slika 16. Android studio. Izradio: Autor

Slika 17. Prikaz SDK Manager. Preuzeto sa: Beginning Android programming with Android Studio

Slika 18. Primjer izgleda HTML stranice. Izradio: autor

Slika 19. Primjer CSS datoteke. Izradio: autor

Slika 20. Prikaz strukture čvorova u Firebase bazi podataka. Izradio: Autor

Slika 21. Prikaz strukture pod-čvorova u Firebase bazi podataka. Izradio: Autor

Slika 22. Odabir razvojne platforme. Izradio: autor.

Slika 23. Prikaz pravilnog postavljanja konekcije na bazu podataka. Preuzeto sa: <https://firebase.google.com/docs/web/setup>.

Slika 24. Prikaz forme unosa djelatnika. Izradio: autor.

Slika 25. Prikaz navažećeg unosa djelatnika. Izradio: autor

Slika 26. Prikaz uspješnog unosa djelatnika. Izradio: autor.

Slika 27. Kratki prikaz unesenog djelatnika. Izradio: autor.

Slika 28. Detaljan prikaz unešenog djelatnika. Izradio: autor.

Slika 29. Prikaz forme za unosa artikla. Izradio: autor.

Slika 30. Prikaz neuspješnog unosa artikla. Izradio: autor.

Slika 31. Prikaz uspješno unesenog artikla. Izradio: autor.

Slika 32. Prikaz dohvata forme detaljnog pregleda artikla. Izradio: autor.

Slika 33. Tablični prikaz općeg pregleda artikla. Izradio: autor.

Slika 34. Detaljan prikaz određenog artikla. Izradio: autor.

Slika 35. Prikaz upozorenja administratoru o promijenjenom opisu artikla. Izradio: autor.

Slika 36. Prikaz forme za brisanje artikla. Izradio: autor.

Slika 37. Prikaz forme za brisanje artikla. Izradio: autor.

Slika 38. Prikaz forme za unos recepture. Izradio: autor.

Slika 39. Prikaz načina spremanja recepture i poruke uspješnosti. Izradio: autor.

Slika 40. Prikaz forme unosa djelatnika. Izradio: autor.

Slika 41. Tablični prikaz unešenih receptura. Izradio: autor.

Slika 42. Detaljan prikaz unešenih receptura. Izradio: autor.

Slika 43. Brisanje određenog recepta. Izradio: autor.

Slika 44. Prikaz forme unosa naloga. Izradio: autor.

Slika 45. Prikaz uspješno unešenog naloga. Izradio: autor.

Slika 46. Prikaz forme unešenih naloga. Izradio: autor.

Slika 47. Detaljan prikaz naloga. Izradio: autor.

Slika 48. Prikaz forme unosa telefonskog broja i lozinke korisnika. Izradio: autor.

Slika 49. Prikaz greške kod unosa telefonskog broja. Izradio: autor.

Slika 50. Prikaz greške kod unosa lozinke korisnika. Izradio: autor.

Slika 51. Prikaz aktivnosti naloga po datumu. Izradio: autor.

Slika 52. Prikaz stavki odabranog naloga. Izradio: autor.

Slika 53. Prikaz detaljnih količina artikala u zamjesu iz datog naloga. Izradio: autor.

Slika 54. Prikaz detaljnih količina artikala u zamjesu iz datog naloga. Izradio: autor.

Slika 55. Prikaz detaljnih količina artikala u zamjesu iz datog naloga. Izradio: autor.

Slika 56. Prikaz detaljnih količina artikala u zamjesu iz datog naloga. Izradio: autor.

Popis tablica:

1. Tablice 1. Preuzeto sa: Firebase official: Dostupno na:
<https://firebase.google.com/pricing/>
Pristupljeno: 27.01.2019
2. Tablica 2. Usporedba firebse databse i firebase cloud sa: Firebase official:
Dostupno na: <https://firebase.google.com/pricing/> Pristupljeno: 27.01.2019
3. Tablica 3. Usporedba Firebase baze podataka u stvarnom vremenu i SQL Oracle
relacijske baze podataka. Dostupno na: <https://db-engines.com/en/ranking>

Sažetak

Ovaj rad sastoji se od teorijskog i praktičnog dijela razvoja aplikacijskog rješenja za koordinaciju procesa izrade pekarskih proizvoda. Kroz teorijski dio diplomskog rada opisano je korištenje NoSql baza podataka te je napravljena usporedba između Firebase baze podataka u realnom vremenu i drugih baza podataka. U praktičnom dijelu rada opisan je način razvoja aplikacije, korištene tehnologije i alati potrebni za razvoj. Aplikacija kao bazu koristi Firebase bazu podataka u realnom vremenu. Administratorski dio aplikacijskog rješenja izrađen je kao web aplikacija, dok je korisnički dio aplikacije izrađen kao mobilna aplikacija.

Ključne riječi: koordinacija pekarskih proizvoda, web aplikacija, mobilna aplikacija, firebase baza podataka

Summary

This work consists of a theoretical and practical part of the development of an applicational solution for co-ordination of bakery product making process. Through the theoretical part of the project, the use of a NoSql database was described and a comparison was made between the real time Firebase database and other databases. The practical part of the project describes the way the application development, the technology used and the tools needed for development. The application as a base uses the Firebase database in real time. The administration part of the application solution was created as a web application, while the user part of the application was created as a mobile application.

Key words: co-ordination of bakery products, web applications, mobile applications, firebase databases