

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Leon Makorič

**Prototip platforme za sistem POS za
restavracije**

DIPLOMSKO DELO

VISOKOŠOLSKI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Rok Rupnik

Ljubljana, 2018

COPYRIGHT. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil L^AT_EX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Restavracije in gostinski lokali za svoje delovanje potrebujejo informacijski sistem, ki podpira njihove procese prodaje: vnos jedi in pijač, vnos cenikov ter izdajanje računov. Ena od možnosti je, da bi POS sistem bil na voljo kot storitev v oblaku, ki bi jo najeli in lahko začeli takoj uporabljati. Proučite informacijske potrebe restavracij in gostinskih lokalov ter določite funkcionalnosti, ki jih boste razvili za prototip. Za razvoj uporabite tehnologije po vašem izboru. Implementirajte tudi možnost tiskanja računa preko mobilne telefona natakarja in majhnega tiskalnika, ki je s telefonom povezan preko tehnologije Bluetooth.

Kazalo

Povzetek

Abstract

1	Uvod	1
1.1	Cilj diplomske naloge	2
1.2	POS in sistem POS	2
1.3	Pregled področja	3
1.4	Struktura diplomske naloge	3
2	Pregled problema in rešitve	5
2.1	Pregled problema	5
2.2	Rešitev problema	6
3	Tehnologije in uporabljena programska oprema	9
3.1	HTML in HTML5	9
3.2	CSS in CSS3	10
3.3	JavaScript	10
3.4	Node.js	11
3.5	Express.js	11
3.6	React	12
3.7	Git	12
3.8	Heroku	13
3.9	XML	13

3.10	Java	14
3.11	Operacijski sistem Android	15
3.12	Android Studio	15
3.13	PowerDesigner	16
3.14	SublimeText	16
4	Analiza in načrt prototipa platforme	17
4.1	Diagram primerov uporabe	17
4.2	Podatkovni model	20
4.3	Arhitektura platforme	21
5	Predstavitev spletne aplikacije	25
5.1	Dostop	25
5.2	Ponudba	26
5.3	Grafi	28
5.4	Profil	28
5.5	Administrativna forma	29
6	Predstavitev mobilne aplikacije	31
6.1	Dostop	31
6.2	Forma za izdelavo računov	32
7	Nadaljnji razvoj	37
7.1	Integracija s FURS	37
7.2	Možnost plačila s plačilnimi karticami	37
7.3	Možnost izdelave posebnega tipa uporabniških računov za uporabnike mobilne aplikacije	38
7.4	Avtomatska izdelava spletnih strani	38
7.5	Razširitev selekcije grafov	38
7.6	Mobilna aplikacija tudi za stranke	39
7.7	Rezervacija mize in ali menija	39
7.8	Uber za prevoz na lokacijo	39

8 Zaključek	41
Literatura	43

Seznam uporabljenih kratic

kratica	angleško	slovensko
API	Application Programming Interface	Aplikacijski programski vmesnik
FURS	Financial Administration of the republic of Slovenia	Finančna uprava republike Slovenije
HTML	HyperText Markup Language	jezik za označevanje besedila
HTTP	Hypertext Transfer Protocol	protokol za prenos hiperteksta
IDE	Integrated Development Environment	Integrirano razvojno okolje
JSON	JavaScript Object Notation	Označevanje objektov JavaScript
MathML	Mathematical Markup Language	Matematični označevalni jezik
ORM	Object-relational mapping	Objektno-relacijsko preslikovanje
OS	Operating System	Operacijski sistem
POS	Point of sale	Točka prodaje
REST	Representational State Transfer	Predstavitveni prenos stanja
RSS	Rich Site Summary	Zgoščeni povzetek strani
SDK	Software Development Kit	Komplet za razvoj programske opreme
WORA	Write once, run anywhere	Napišite enkrat, zaženite kjerkoli
XML	eXtensible Markup Language	Razširljiv označevalni jezik

Povzetek

Naslov: Prototip platforme za sistem POS za restavracije

Avtor: Leon Makorič

V diplomskem delu smo se posvetili razvoju prototipa platforme za storitveno dejavnost namenjeno predvsem za uporabo v gostinstvu (bari, restavracije, picerije) in majhnim storitvenim podjetjem (fitnes centri, frizerski saloni, avtoprevozniki). Platforma je na voljo kot storitev (ang. Platform as a service), kar podjetjem še bolj poenostavi njen sprejem.

Platformo smo razdelili na spletno aplikacijo, mobilno aplikacijo in zaledje. V spletni aplikaciji lahko lastnik storitvene dejavnosti ustvari svojo ponudbo, medtem ko z uporabo mobilne aplikacije lahko zaposleni na hiter, učinkovit ter enostaven način izdelava in natisne račun z uporabo Bluetooth termičnega tiskalnika. Zaledje pa skrbi za komunikacijo in pravilno delovanje obeh delov platforme.

Ključne besede: mobilna aplikacija, spletna aplikacija, POS, termični tiskalnik, mala storitvena podjetja, gostinstvo.

Abstract

Title: Prototype of platform for POS system for restaurants

Author: Leon Makorič

In the diploma thesis, we developed a prototype of a platform intended primarily for use in catering (bars, restaurants, pizzerias) and small sized enterprises (fitness centers, hairdressers, hauliers). The platform is available as a service (PaaS), which makes it even easier for businesses to adopt.

We have divided the platform into three parts: web application, mobile application and backend. A business owner can use the web application to create his own offer whereas an employee can use the mobile application to create and print an invoice in a quick, easy and efficient way. The backend is responsible for the communication and proper functioning of both parts of the platform.

Keywords: mobile application, web application, POS, thermal printer, small enterprises, catering.

Poglavje 1

Uvod

Dne 2. januarja 2016 je postala uporaba davčnih blagajn (na katerih teče sistem POS) obvezna. Zakon o davčnem potrjevanju računov je predvidel tudi predhodno dveletno obdobje do 31. decembra 2017, v katerem je bilo še najprej možno uporabljati zgolj vezano knjigo računov. Zakon je bil nato spet spremenjen in dopolnjen, zavezanci so se od 1. januarja 2018 lahko sami odločili ali bodo pri gotovinskem poslovanju uporabljali vezano knjigo računov ali davčno blagajno [4].

Zaradi tega je uporaba sistemov POS v Sloveniji hitro narasla. Ti sistemi lahko tečejo na večini naprav kot so računalniki, mobilne naprave in blagajne. Prav zaradi te prilagodljivosti jih srečamo na mnogih področjih npr.: v gostinstvu (čajnice, bari, kavarne, klubi, picerije, pivnice, restavracije, slaščičarne) in pri raznih storitvah (fitnes centri, frizerski saloni, foto ateljeji, kioski, kozmetični saloni, taksisti, avtoprevozniki, krojači, čevljarji).

Razvili smo prototip platforme za storitveno dejavnost, ki ima veliko skupnih lastnosti s sistemom POS. V nadaljnjih poglavjih bomo opisali področje platforme, kakšne tehnologije smo uporabili za razvoj, njeno zasnovano, kako deluje ter možnosti nadaljnjega razvoja.

1.1 Cilj diplomske naloge

Cilj diplomske naloge je izdelava prototipa platforme za storitveno dejavnost, namenjeno za uporabo v gostinstvu in majhnim storitvenim podjetjem. Platforma je na voljo kot storitev (ang. Platform as a service).

Platformo smo razdelili na spletni ter mobilni del in zaledni del, ki je skupšen obema. Cilj spletnega dela je omogočiti uporabniku prijavo in registracijo v spletno aplikacijo, pregled ter urejanje svojih podatkov, sestavo in urejanje ponudbe ter prikaz statističnih podatkov v obliki grafov. Cilj mobilnega dela je omogočiti uporabniku prikaz ponudbe, izdelavo računa z možnostjo uveljavitve popusta ter tiskanje računa prek Bluetooth termičnega tiskalnika. Za pravilno delovanje in komunikacijo spletne aplikacije in mobilne aplikacije skrbi zaledni del.

1.2 POS in sistem POS

Točka prodaje (ang. Point of sale) je kraj in čas kjer je bila opravljena malo-prodajna transakcija [17]. Na POS točki prodajalec stranki zaračuna znesek in ji predstavi možnosti plačila. Najpogostejše možnosti plačila so gotovina ali plačilne kartice. Nato stranki izdela in izda račun, ki je pogosto natisnjen s termičnim tiskalnikom. Pri izdaji računa prodajalci pogosto uporabljajo naprave kot so tehtnice, čitalniki črtnih kod in blagajne.

Sistem POS je aplikacija, ki omogoča izvajanje plačevanja v okviru prodaje. Te aplikacije so pogosto nameščene na računalnike, mobilne naprave in blagajne. V današnjem času je uporaba teh aplikacij vedno bolj pogosta na mobilnih napravah, saj podjetja vse bolj pogosto sprejemajo sisteme POS zaradi njihovih prednosti kot so: enostavnost implementacije raznih popustov, enostavnost implementacije sheme zvestobe za kupce in enostavnost uporabe sistema.

1.3 Pregled področja

Pri pregledu področja smo ugotovili, da je takih aplikacij (še posebej za platformo Android) veliko. Pregledali smo najbolj pogosto uporabljene aplikacije kot so TRONpos, Čebelca MOBI in Birokrat. Ugotovili smo, da so si te aplikacije zelo podobne saj ponujajo enake glavne možnosti in se večinoma razlikujejo v izgledu ter ne nujnih dodatnih funkcijah kot na primer Google drive backup in izvoz podatkovne baze.

Naš prototip platforme se od njih razlikuje v tem, da uporabniku ponudimo tudi spletno aplikacijo, prek katerega lahko urejajo svoje podatke, izdelajo ponudbo ter jo po želji urejajo in imajo pregled nad statističnimi podatki v obliki grafov. Prav zaradi tega je naš prototip aplikacije veliko bolj razširljiv. Možnosti razširitve bomo podrobneje opisali v poglavju namenjenemu nadaljnjemu razvoju platforme.

1.4 Struktura diplomske naloge

Na začetku diplomskega dela pričnemo s pregledom področja, problema in opisom predlagane rešitve (poglavje 1 in 2). Nadaljujemo s predstavbo glavnih tehnologij, katere smo uporabili za realiziranje prototipa platforme (poglavje 3). Nato predstavimo način s katerimi so nam v prejšnjem poglavju omenjene tehnologije pomagale pri zgradnji moderne platforme (poglavje 4). Nadaljujemo s predstavitvijo spletne aplikacije in njenih funkcionalnosti (poglavje 5). Sledi predstavitev mobilne aplikacije (poglavje 6), zaključimo z nekaj predlogi scenarijev za nadaljnji razvoj (poglavje 7) in zaključkom (poglavje 8).

Poglavje 2

Pregled problema in rešitve

2.1 Pregled problema

Gostinstvo ter mala storitvena podjetja potrebujejo hiter in učinkovit način za izdelavo in izdajo računa. Vedno pogosteje se odločajo za uporabo sistema POS, ki to omogoča ter hkrati prinaša veliko dodatnih funkcionalnosti in prednosti. Takih komercialnih sistemov je tudi v Sloveniji mnogo in so si glede funkcionalnosti med seboj zelo podobni.

Večje prednosti sistemov POS so naslednje:

- odprava potrebe po cenovnih oznakah,
- enostavna implementacija raznih popustov,
- enostavna implementacija shem zvestob,
- učinkovitejša kontrola zalog.

Pomembnejše funkcionalnosti sistemov POS so naslednje:

- pregled ponudbe artiklov in storitev v realnem času skupaj z njihovimi podatki,
- izbira načina plačila,

- izdelava in tisk računa,
- urejevanje artiklov,
- urejevanje dokumentov,
- spremembe nastavitev aplikacije,
- urejanje uporabniških podatkov,
- analize uporabe,
- izdelava in pregled raznih poročil,
- izvoz na FURS,
- pregled prejemkov ter izdatkov,
- test tiskanja,
- ponastavitev aplikacije.

2.2 Rešitev problema

Pri izdelavi rešitve smo omejili obseg prototipa platforme za storitveno dejavnost. Platformo smo zasnovali kot storitev, kar podjetjem še bolj poenostavi njen sprejem. Prototip smo razdelili na spletni ter mobilni del in zaledni del, ki je skupšen obema.

Spletni del omogoča registracijo in prijavo, pregled ponudbe, izdelavo ponudbe, urejanje ponudbe, pregled statističnih podatkov v obliki grafov in urejanje lastnih podatkov. Administrator sistema lahko uporabnike tudi briše ali jim ureja podatke.

Mobilni del omogoča prijavo, pregled ponudbe, povezavo ter odklop Bluetooth termičnega tiskalnika, izdelavo in tiskanje računa, dodajo fiksne ali procentnega popusta na račun, ponovno tiskanje zadnjega računa, izbris prejšnje nezaključenega računa in odjavo iz aplikacije.

Za pravilno delovanje in komunikacijo spletne aplikacije in mobilne aplikacije skrbi zaledni del.

Poglavje 3

Tehnologije in uporabljena programska oprema

3.1 HTML in HTML5

HTML je označevalni jezik uporabljen za izdelavo spletnih strani ter spletnih aplikacij. Spletni brskalniki prejemaajo HTML dokumente iz spletnega strežnika ali iz lokalnega pomnilnika in jih nato upodobijo v multimedijske spletne strani. HTML semantično opisuje strukturo spletne strani [9]. HTML značke so gradniki HTML dokumentov. Teh značk brskalniki in odjemalci ne prikazujejo ampak jih uporabijo za definiranje in prikaz vsebine.

HTML5 je trenutno peta in najnovejša večja različica standarda HTML, ki je izšla oktobra 2014 [7]. HTML5 je standard nadgradil z vključitvijo dodatnih značk (<video>, <audio> in <canvas>) omogočil podporo prilagodljivi vektorski grafiki (ang. Scalable Vector Graphics) in MathML formul, ki nativno vključujejo in obdelujejo večpredstavnostno vsebino. Za dodatno obogatitev semantične vsebine dokumentov so bile uvedene značke <main>, <section>, <article>, <header>, <footer>, <aside>, <nav>, in <figure>. HTML5 tudi bolje definira procesiranje za vse neveljavne dokumente [10].

3.2 CSS in CSS3

Jezik stilskega besedila (ang. Cascading Styling Sheets) opisuje izgled dokumenta oziroma spletne strani, napisanega v označevalnem jeziku, kot je HTML5. Čeprav je CSS mogoče uporabiti za vsak dokument, ki se drži standarda XML, se najpogosteje uporablja za dokumente HTML. CSS je zasnovan na način, da omogoča ločitev predstavitev in vsebine, vključno s postavitvijo, barvami in pisavami [3]. Ta ločitev lahko izboljša dostopnost vsebin, zagotovi večjo prilagodljivost ter nadzor v specifikaciji značilnosti predstavitev in omogoči več spletnim stranem skupno rabo okolja, tako da v ločeni datoteki .css določimo ustrezen CSS. S tem tudi zmanjšamo kompleksnost in ponavljanje v strukturi vsebini [2]. Za omogočitev zelo preproste sintakse CSS uporablja preproste angleške besede za različne lastnosti sloga.

CSS3 vsebuje vse kar je bilo vključeno v prejšnji verziji CSS2.1. Doda pa nove funkcije, ki razvijalcem pomagajo pri reševanju številnih problemov, povezanih s postavitvijo, ne da bi pri tem morali uporabljati skripte ali dodatne slike. Te funkcije vključujejo podporo za dodatne selektorje, padajoče sence, zaokrožene vogale in posodobljene funkcije postavitve [7].

3.3 JavaScript

JavaScript je programski jezik za splet. Podpirajo ga vsi popularni brskalniki, kot so Chrome, Firefox, Safari, Internet Explorer, Edge, Opera itd.. Prav tako JavaScript podpira tudi večina mobilnih brskalnikov pametnih telefonov. Uporablja se predvsem za izboljšanje uporabnikove izkušnje spletnih strani. Pogosteje uporabljene funkcionalnosti JavaScripta ponujajo dinamično posodabljanje spletnih strani, izboljšave uporabniškega vmesnika, kot so meniji in pogovorna okna, animacije, 2D in 3D grafika, interaktivni zemljevidi in video predvajalniki. Ta način uporabe JavaScripta v brskalniku je poznan kot JavaScript na strani odjemalca [26]. Čeprav je bil na začetku JavaScript implementiran le na strani odjemalca v spletnih brskalnikih je sedaj pogon JavaScripta lahko tudi vgrajen v številne druge vrste gostiteljske programske

opreme [12]. Eden izmed takih načinov uporabe je JavaScript kot strežnik. Čeprav je med JavaScriptom in Javo veliko podobnosti, sta jezika različna in posebej različna pri oblikovanju.

3.4 Node.js

Node.js je strežniška platforma, zgrajena na JavaScript pogonom V8, ki je del brskalnika Chrome. Razvil jo je Ryan Dahl leta 2009, njegova trenutna najnovejša različica je 10.7.0. Node.js je odprto koden in je najpogosteje uporabljen za razvoj aplikacij na strani strežnika in ostalih mrežnih aplikacij. Node.js aplikacije so napisane v JavaScriptu in se lahko izvajajo v Node.js izvajalnem okolju na okoljih Microsoft Windows, Mac OS in Linux. Node.js ponuja tudi bogato knjižnico modulov JavaScript, ki v veliki meri poenostavijo razvoj spletnih aplikacij [15].

Node.js je:

- asinhron in dogodkovno usmerjen,
- zelo hiter,
- enoniten ampak zelo stopnjevalen.

3.5 Express.js

Express.js je spletno aplikacijsko ogrodje, ki ponuja enostaven API za izdelavo spletnih strani, spletnih aplikacij in zaledja aplikacij. Zagotavlja minimalen vmesnik in orodja, ki so potrebna za izdelavo aplikacij. Je daleč najbolj uporabljen modul Node.js in zelo fleksibilen, saj so pri upravitelju paketov Node.js (ang. Node.js packet manager) številni moduli, ki jih je mogoče neposredno priključiti na Express. Razvil ga je TJ Holowaychuk in ga trenutno vzdržuje fundacija Node.js in številni odprtokodni prispevki [5].

3.6 React

React oziroma React.js je JavaScript knjižnica za gradnjo uporabniških vmesnikov, ki ga vzdržuje podjetje Facebook. Razvil ga je Joran Walke, programski inženir, zaposlen na Facebooku. V javnosti je bil objavljen maja 2013. React se najpogosteje uporablja kot osnova za razvoj enostranskih ali mobilnih aplikacij. Kompleksne React aplikacije običajno potrebujejo tudi druge knjižnice za upravljanje, usmerjanje in interakcijo z API [19]. Trenutno je ena izmed najbolj popularnih opcij. Ima trdno podlago in veliko skupnost, ki jo podpira. V Reactu je vse komponenta oziroma blok, te bloke se nato integrira za izgradnjo večje in dinamične aplikacije. To je velika prednost saj lahko katerokoli komponento kadarkoli spremenimo, brez da bi s tem vplivali na ostale aplikacije [27]. Ta funkcija je še učinkovitejša, če se izvaja z večjimi aplikacijami v realnem času, kjer se pogosto spreminjajo podatki.

3.7 Git

Git je danes najpogosteje uporabljen sistem za nadzor različic in hitro postaja standard za nadzor nad različicami [21]. Git je leta 2005 ustvaril Linus Torvalds za razvoj jedra Linuxa s pomočjo drugih razvijalcev jeder. Od leta 2005 ga vzdržuje Junio Hamando [6]. Ker je Git distribuiran sistem za nadzor različic, to pomeni, da je lokalna kopija kode popolni repozitorij nadzora različic. Te lokalne repozitorije omogočajo enostavno delo brez povezave ali na daljavo. Vsakič, ko je delo shranjeno, Git to zapiše. Ta zapis je posnetek stanja vseh datotek projekta v določenem času. Zapisi ustvarijo povezave do ostalih zapisov in s tem tvorijo graf zgodovine razvoja. Koda je lahko kadarkoli spremenjena na poljubni posnetek stanja. Te posnetke Git identificira z edinstvenim kriptografskim razprševanjem vsebine. Zaradi tega je posnetku nemogoče spremeniti vsebino, izgubiti ali poškodovati podatke, ne da bi to Git odkril. Git ponuja orodja za izolacijo sprememb, katere lahko kasneje združimo nazaj skupaj. Veje, ki so kazalci za delo v teku, upravljajo to ločitev. Končano delo ustvarjeno v veji lahko združimo v glavno (ali

mojstrsko) vejo projekta [20].

3.8 Heroku

Heroku je platforma na voljo kot storitev (ang. Platform as a service), ki podpira več programskih jezikov [13]. Razvita je bila kot ena prvih platform za oblake kot storitev junija 2007, ko je podpirala le programski jezik Ruby, trenutno pa podpira Javo, Node.js, Scalo, Clojure, Python, PHP in Go. Velika izbira jezikov razvijalcu omogoča gradnjo, izvajanje in razširitev aplikacije na podoben način. Heroku je leta 2010 kupilo podjetje Salesforce.com [8]. Glavne funkcionalnosti Herokuja so:

- hipna postavitve z Git push,
- veliko dodatnih modulov in virov,
- skaliranje procesov,
- izolacija,
- popolno beleženje in vidnost [24].

3.9 XML

XML oziroma razširljiv označevalni jezik opredeljuje pravila za kodiranje dokumentov v formatu, ki je berljiv za ljudi in stroje. Cilji oblikovanja XML poudarjajo preprostost, splošnost in uporabnost na internetu. Čeprav se oblikovanje XML posveča dokumentom, se jezik pogosto uporablja za predstavitev poljubnih struktur podatkov, na primer tistih, ki se uporabljajo v spletnih storitvah [28]. Ker je XML razširljiv omogoča določbo lastnih oznak, vrstnega reda, v katerim se pojavijo in kako jih je treba obdelati ali prikazati. Prav te oznake oziroma elementi so tudi njegova najbolj prepoznavna značilnost. Dejansko bodo ti elementi, ki bodo ustvarjeni v XML, zelo podobni elementom ustvarjenih v dokumentih HTML. Bistvena razlika je, da

XML omogoča določitev svojega nabora oznak. Prav zaradi tega je bolj prilagodljiv kot HTML [29]. XML je meta-jezik torej jezik, ki nam omogoča ustvarjanje ali določanje drugih jezikov. Popularni primeri tega so RSS in MathML [23].

3.10 Java

Java je sočasen in predmetno usmerjen splošen programski jezik. Prevedena Java koda deluje na večini operacijskih sistemov, vključno z operacijskimi sistemi Microsoft Windows, Linux in Mac OS, saj temelji na principu WORA. Večina sintakse je pridobljena iz programskih jezikov C in C++ [11]. Jezik je leta 1995 razvil James A. Gosling, računalniški znanstvenik, ki je bil v preteklosti zaposlen v podjetju SunMicrosystems [14]. Java je deljena na navidezno napravo Java (ang. Java Virtual Machine), izvajalno okolje Java (ang. Java Runtime Environment) in Java razvojni komplet (ang. Java Development Kit). Glavne lastnosti Jave so:

- preprostost - Java je odstranila kazalce in preobremenitev operaterja,
- prenosnost - Java je neodvisna od platforme, kar pomeni, da je koda napisana na eni platformi lahko preprosto prenesena na drugo,
- objektna usmerjenost - vse velja za objekt, ki ima določeno stanje in obnašanje,
- varnost - vsa koda je pretvorjena v bytekodo po prevajanju, ki človeku ni berljiva,
- dinamičnost - sposobna se je prilagoditi razvijajočemu okolju,
- porazdeljenost - ponuja funkcijo, ki pomaga ustvarjati porazdeljene aplikacije,
- robustnost - ima močan sistem upravljanja pomnilnika, ki pomaga pri odpravljanju napak,

- visoka zmogljivost - Java svojo bytekodo omogoča enostavno prevesti v strojno kodo,
- interpretacija - Java je prevedena v bytekodo, katero interpretira izvajalno okolje Java,
- večnitnost - Java podpira več niti za izvajanje [25].

3.11 Operacijski sistem Android

Android je mobilni operacijski sistem, katerega je razvil Google. Temelji na spremenjeni različici jedra Linuxa in druge odprtokodne programske opreme. Zasnovan je predvsem za mobilne naprave na dotik, kot so pametni telefoni in tablični računalniki. Poleg tega je Google razvil tudi Android TV za televizorje, Android Auto za avtomobile in Wear OS za zapestne ure, od katerih ima vsak posebej poseben uporabniški vmesnik. Različice Androida se uporabljajo tudi na igralni konzolah, digitalnih fotoaparatih, osebnih računalnikih in drugi elektroniki. Prvotno je operacijski sistem razvijal Android Inc., katerega je Google kupil leta 2005, leta 2007 je predstavil operacijski sistem Android, s prvo komercialno napravo Android, ki je prišla na trg leta 2008. Trenutna najnovejša verzija operacijskega sistema je 8.1, ki je izšla decembra 2017.

Android je od leta 2011 najbolj prodajan operacijski sistem (ang. OS) po vsem svetu na pametnih telefonih in na tabličnih računalnikih od leta 2013. Od maja 2017 ima več kot dve milijardi mesečnih aktivnih uporabnikov, od junija 2018 trgovina GooglePlay shranjuje že več kot 3,3 milijone aplikacij [16].

3.12 Android Studio

Android Studio je uradno integrirano razvojno okolje (angl. IDE) za Googleov operacijski sistem Android, ki temelji na programski opremi JetBrains

IntelliJ IDEA in je zasnovan posebej za razvoj Androida. Podpira operacijske sisteme Microsoft Windows, Mac OS in Linux. Nadomestil je razvojno okolje Eclipse Android kot primarni IDE za razvoj aplikacij Android. Android Studio je bil objavljen decembra 2014 z različico 1.0. Trenutna stabilna različica je 3.1.3 [1]. Uporabili smo ga za razvoj mobilne aplikacije.

3.13 PowerDesigner

PowerDesigner je orodje za modeliranje. Namenjeno je podjetjem, izdelal ga je Sybase in je trenutno v lasti podjetja SAP. Nativno deluje pod operacijskim sistemom Microsoft Windows in teče skozi Eclipse preko vtičnika. PowerDesigner podpira modelsko arhitekturo načrtovanja programske opreme. Shranjuje modele, ki uporabljajo različne razširitve datotek, kot so .bpm, .cdm in .pdm. Notranja struktura datoteke je lahko XML ali stisnjen format binarne datoteke. Prav tako PowerDesigner omogoča shranjevanje modelov v repozitorij baz podatkov [18]. Uporabili smo ga za izdelavo modelov diagramov primera uporabe.

3.14 SublimeText

SublimeText je lastniški urejevalnik teksta. Deluje na sistemih Microsoft Windows, Mac OS in Linux. Podpira številne programske in označevalne jezike. Dodatne funkcionalnosti se lahko dodajo z uporabo vtičnikov [22]. Uporabili smo ga pri pisanju kode za zaledje ter za spletno aplikacijo.

Poglavje 4

Analiza in načrt prototipa platforme

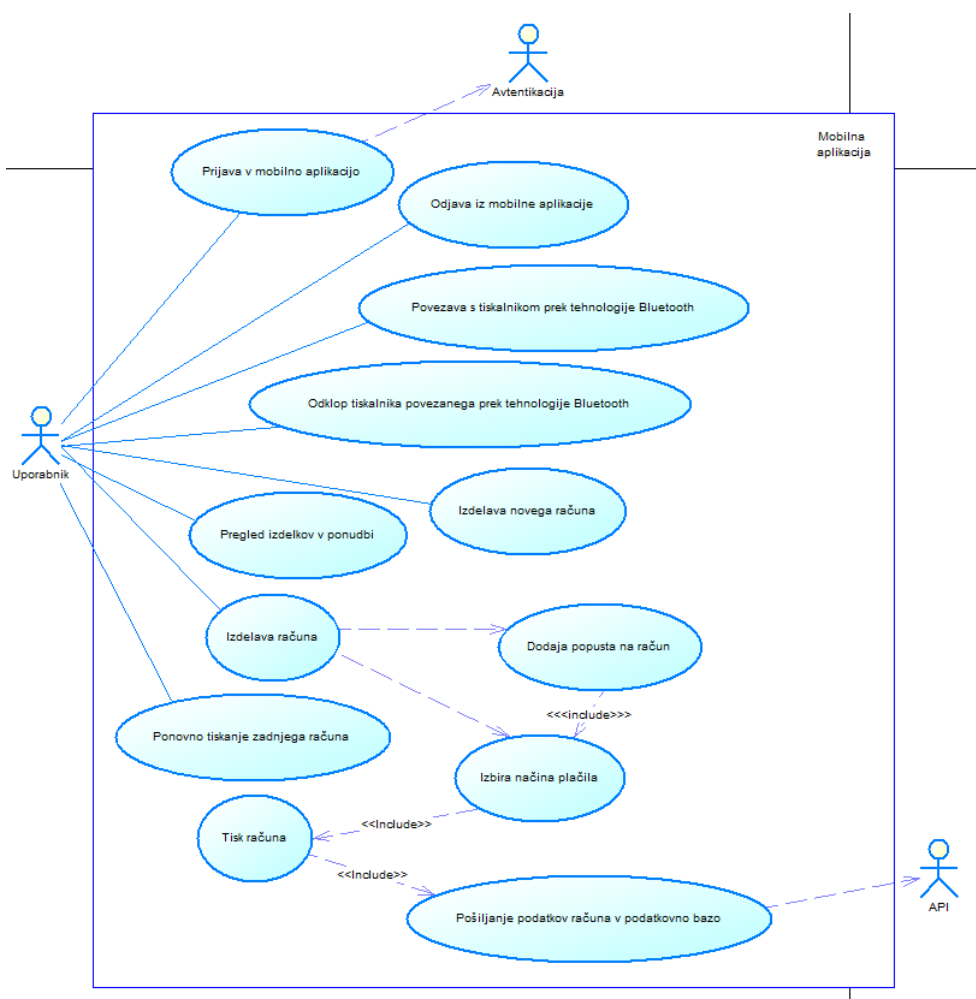
Poglavje predstavlja izdelke in rezultate faz analize in načrtovanja. V okviru faze analize smo izdelali diagram primerov uporabe in podatkovni model. V okviru faze načrtovanja pa določili arhitekturo platforme.

4.1 Diagram primerov uporabe

Pred začetkom načrtovanja izdelave zaledja, uporabniških vmesnikov in podatkovne baze smo funkcionalnosti zabeležili preko diagramov primerov uporabe. Nato smo za predstavitev funkcionalnosti platforme s programom PowerDesigner izdelali diagrame primerov uporabe (Slika 4.1 in 4.2). S tem smo spoznali katere funkcionalnosti naj aplikacija omogoča kateremu tipu uporabnika.



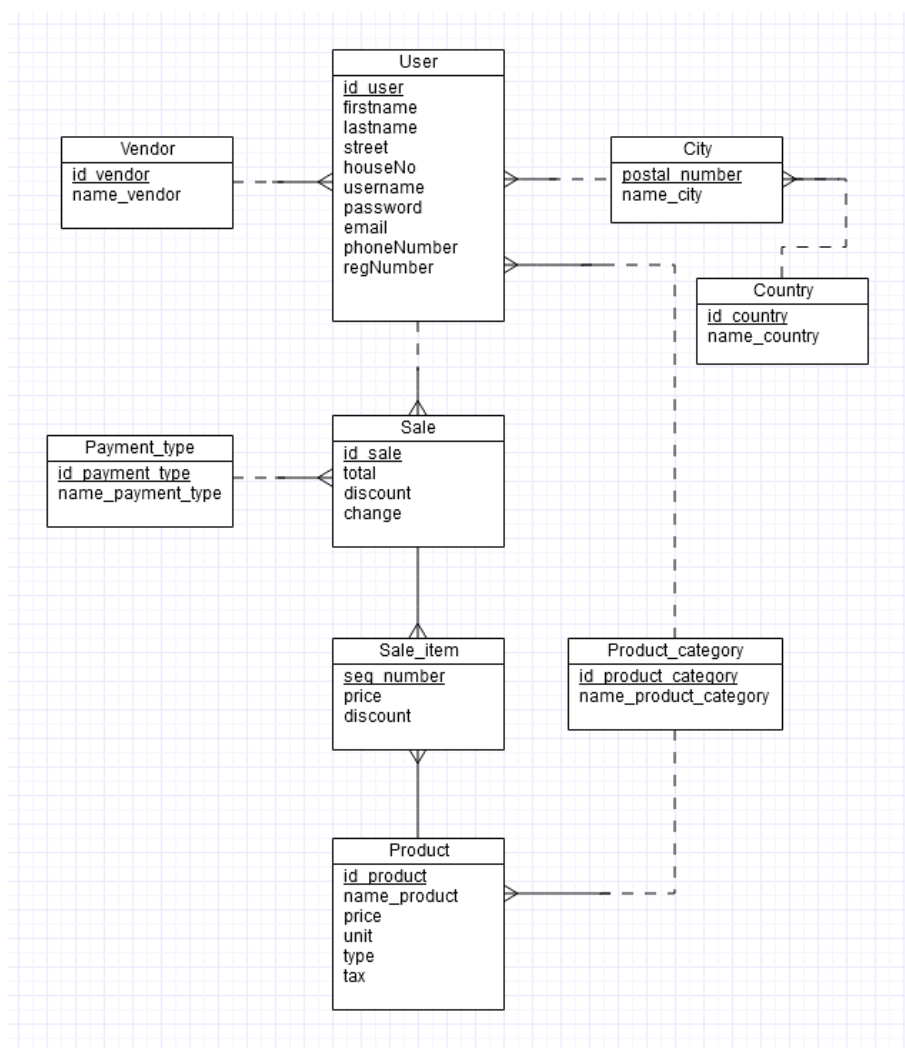
Slika 4.1: Diagram primera uporabe za spletno aplikacijo



Slika 4.2: Diagram primera uporabe za mobilno aplikacijo

4.2 Podatkovni model

Podatkovni model (Slika 4.3) smo izdelali na osnovi diagramov primera uporabe in zahtev, ki so se pokazale med načrtovanjem uporabniških vmesnikov in podatkov, potrebnih za izdelavo ter prikaz grafov. Podatkovni model ima več skupin entitet. Najprej so entitete, ki hranijo podatke o uporabnikih in sicer: njihovo ime, priimek, uporabniško ime, e-pošto, ulico, hišno številko, mesto, državo, poštno številko, davčno številko, telefonsko številko, matično številko ter geslo. Nato imamo entitete, ki hranita podatke o artiklih ter kategorije, katerim artikel pripada. V njih hranimo torej: ime produkta, cena produkta, procent davka, tip produkta (blago ali servis), enota produkta (KOS, Kg, m, čas, g, l ali Km) in ime kategorije. Ti entitete sta med seboj povezani, da lahko vemo kateri kategoriji pripada posamezen artikel. Z entiteto Vendor poskrbimo, da vemo kakšen status ima določen uporabnik. Imamo tudi entitete, ki omogočajo hrambo vseh izdelanih računov. Vemo torej kolikšen je bil znesek računa in ali je na bil na računu dodan popust ter njegova veličina, koliko gotovine je stranka plačala, s katero valuto je plačala, kolikšno je bilo njeno vračilo in katere artikle je kupila. Poskrbimo tudi, da vsak uporabnik platforme vidi le svoje podatke in nima dostopa do podatkov drugih.

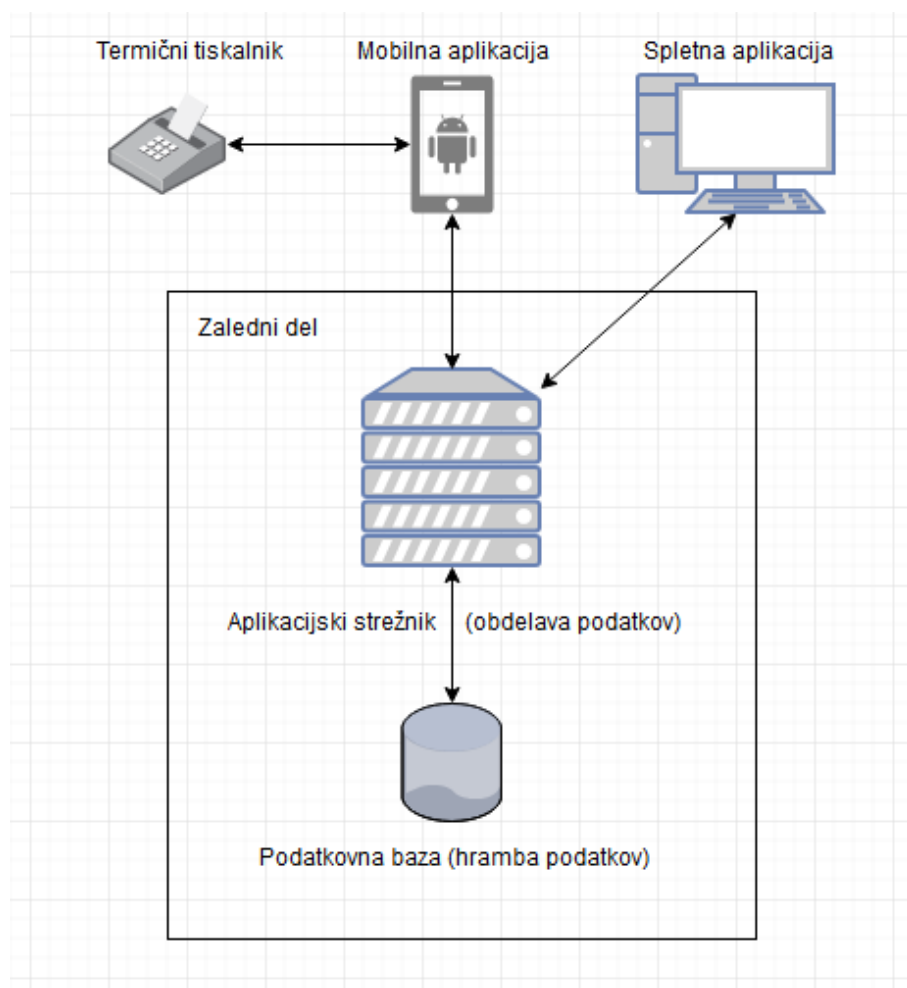


Slika 4.3: Entitetno-relacijski model podatkovne baze

4.3 Arhitektura platforme

Iz sheme arhitekture platforme (Slika 4.4) vidimo, da je platforma razdeljena na spletno aplikacijo, mobilno aplikacijo in zaledni del, ki je skupšen obema. Na mobilno aplikacijo lahko z uporabo tehnologije Bluetooth priključimo termični tiskalnik. Za pravilno delovanje in komunikacijo spletne aplikacije in mobilne aplikacije skrbi zaledni del.

Spletna aplikacija in mobilna aplikacija sta popolnoma ločena od zalednega dela, saj je njun namen prikaz in delo s podatki. Zaledni del je izdelan z uporabo JavaScript pogona Node.js in ogrodja Express.js ter je gostovan na platformi Heroku. Aplikacijski strežnik teče v okolju Linux in poganja JavaScript. Odločili smo se za uporabo PostgreSQL podatkovne baze. Načrtovanje podatkovnega modela na ravni aplikacije smo zgradili preko predmetov v JavaScriptu, ki jih zna modul Sequelize (ORM za Node.js) za komunikacijo med JavaScriptom in PostgreSQL pretvoriti v spremembe podatkovnega modela na fizični osnovi. Heroku na naslovu URL pripravi, odpre in ponudi celoten aplikacijski vmesnik REST. Za obravnavo POST HTTP zahtevkov strežnik uporablja vmesno opremo body-parser, ki izlušči celotno telo prihajajočega toka zahtev in ga izpostavi na req.body. Vsa komunikacija med mobilno aplikacijo, spletno aplikacijo in strežnikom poteka prek objektov JSON. Spletna aplikacija pridobi podatke z uporabo zahtevkov HTTP na vnaprej definiranih končnih točkah v aplikacijskem vmesniku REST. Spletna aplikacija je bila zgrajena z uporabo JavaScript knjižnice React in uporablja modul axios (HTTP odjemalec med brskalnikom in node.js) za komuniciranje. Z uporabo raznih kombinacij klicev na vmesnike odjemalec poskrbi za branje, pisanje in posodabljanje podatkov v aplikaciji. Mobilno aplikacijo, ki teče na platformi Android smo razvili z uporabo jezika Java in XML s programom Android Studio. Za uporabo APIja mobilna aplikacija uporablja Retrofit, ki je REST odjemalec za sisteme Android. V mobilni aplikaciji smo naredili modele za Retrofit, kateri mu omogočajo obravnavo zahtevkov iz JSON APIja. Za Bluetooth komunikacijo med termičnim tiskalnikom in mobilno aplikacijo smo uporabili SDK, ki je prišel skupaj s tiskalnikom.



Slika 4.4: Arhitektura platforme

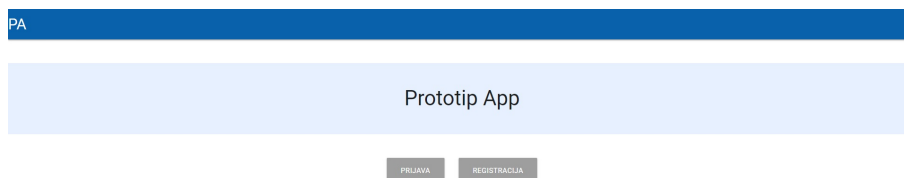
Poglavje 5

Predstavitev spletne aplikacije

5.1 Dostop

Podpoglavje opisuje primera uporabe prijava v spletno aplikacijo in registracija v spletno aplikacijo.

Ob odpiranju spletne strani se prikaže prva forma spletne aplikacije (Slika 5.1), ki nudi uporabniku možnost prijave ali registracije. Če je uporabnik nov lahko s klikom na gumb registracija vnese potrebne podatke in se s tem uspešno registrira. V primeru, da je uporabnik že registriran se v spletno aplikacijo lahko prijavi s svojim uporabniškim imenom ali e-pošto in geslom. Ker spletna aplikacija trenutno ne podpira potrditve računa z e-pošto smo se odločili, da bo uporabnik v primeru želje po spremembi gesla moral kontaktirati administratorja sistema.

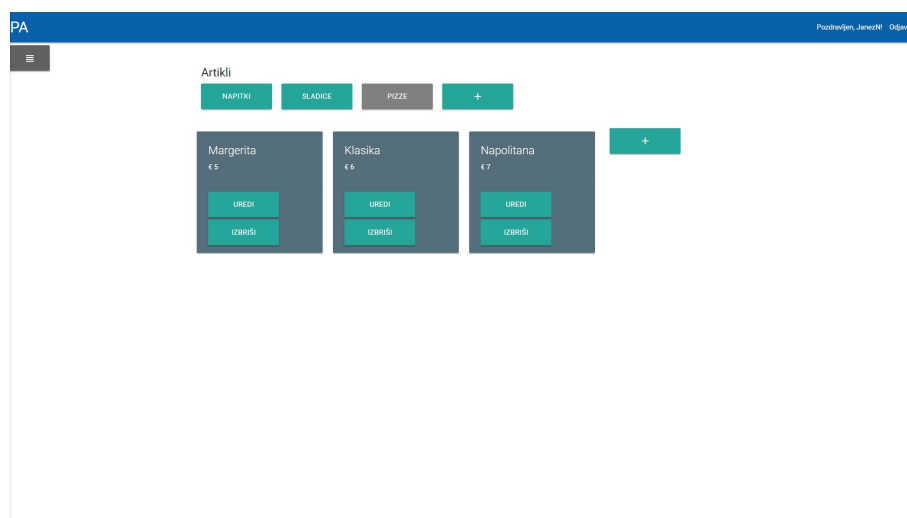


Slika 5.1: Vstopna točka spletne aplikacije

5.2 Ponudba

Podpoglavje opisuje primere uporabe pregled ponudbe, dodaja kategorije, dodaja artikla, izbris artikla in sprememba podatkov artikla.

Po uspešni prijavi se uporabniku odpre pregled nad ponudbo (Slika 5.2). Namen te forme je omogočiti uporabniku izdelavo, urejanje in prikaz njegove ponudbe. Z zgornjim gumbom z oznako + lahko uporabnik ponudbi doda novo kategorijo (npr. mesne jedi) in s tem zgradi ogrodje svoje ponudbe. Ko je s kategorijami zadovoljen lahko klikne na željeno kategorijo in nato z spodnjim gumbom z oznako + v izbrano kategorijo doda nov artikel. Ob dodaji novega artikla ga forma opomne naj vnese naslednje podatke o artiklu: ime artikla, cena artikla, procent davka, tip artikla in enoto artikla (Slika 5.3). Ponudba ni statična, uporabnik lahko kadarkoli uredi ali izbriše katerikoli artikel s klikom na gumb uredi ali izbriši.



Slika 5.2: Forma ponudba spletne aplikacije

Dodaj artikel

Ime artikla
Napolitana

Cena artikla
7

Davek (%)
20

Tip artikla
blago

Enota artikla
KOS

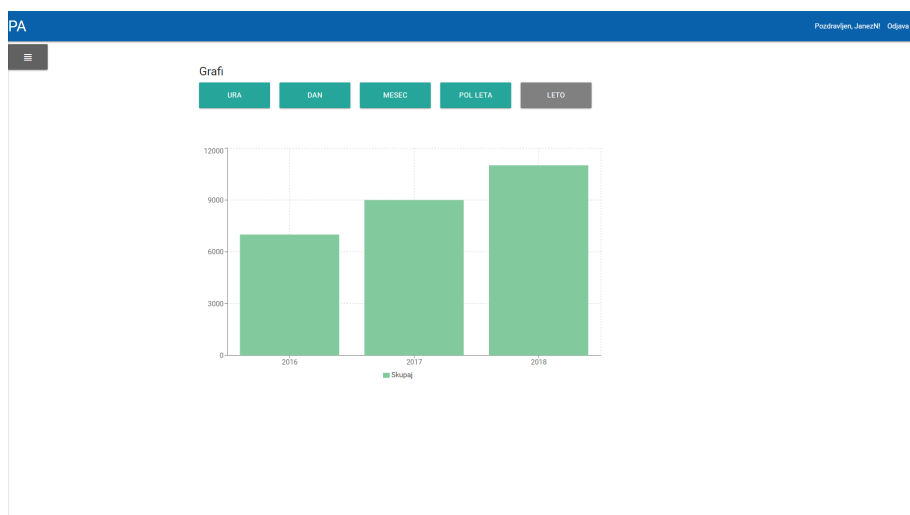
PREKLIČI POTRDI

Slika 5.3: Forma dodaja artikla

5.3 Grafi

Podpoglavje opisuje primer uporabe pregled grafov.

Forma omogoča pregled grafov, ki prikazujejo promet poslovanja (Slika 5.4). Trenutno je na voljo 5 grafov. Graf ura prikazuje promet zadnjih 24 ur za vsako uro. Graf dan prikazuje zadnji teden prometa za vsak dan, medtem ko graf mesec prikazuje zadnje leto prometa za vsak mesec. Graf pol leta prikazuje zadnja 4 polletja in nazadnje graf leto prikazuje zadnja 3 leta prometa. Slika 5.4 prikazuje primer prometa iz leta 2016, 2017 in 2018.

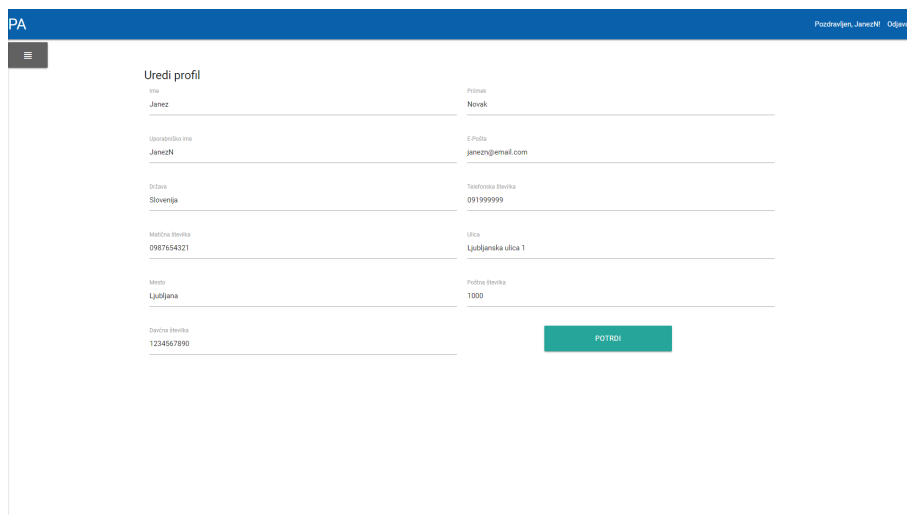


Slika 5.4: Forma grafi spletne aplikacije

5.4 Profil

Podpoglavje opisuje primera uporabe pregled profila in sprememba podatkov profila.

Forma profil je zelo preprosta saj uporabniku omogoča le pregled in urejanje svojih podatkov (Slika 5.5). Ko je uporabnik s spremembami zadovoljen jih preprosto potrdi z klikom na gumb potrdi.



The screenshot shows a web application interface for editing a user profile. At the top, there is a blue header with the text 'PA' on the left and 'Pozdravljen, JanezNI Odjela' on the right. Below the header is a navigation menu with a hamburger icon. The main content area is titled 'Uredi profil' and contains a form with the following fields:

Ime	Preimek
Janez	Novak

Uporabniško ime	E-Pošta
JanezNI	janezn@email.com

Država	Telefonska številka
Slovenija	091999999

Matična številka	Ulica
0987654321	Ljubljanska ulica 1

Mesto	Poštna številka
Ljubljana	1000

Davčna številka
1234567890

At the bottom right of the form is a green button labeled 'POTRDI'.

Slika 5.5: Forma profil spletne aplikacije

5.5 Administrativna forma

Podpoglavje opisuje primera uporabe sprememba podatkov uporabnikov in izbris uporabnika.

Ob uspešni prijavi z administrativnim računom se nam odpre posebna forma namenjena upravljanju uporabnikov (Slika 5.6). Na tej formi lahko administrator vidi vse registrirane uporabnike in njihove podatke. Uporabniki so razvrščeni po uporabniških imenih. Z klikom na ime uporabnika se administratorju odpre okno kjer se mu prikažejo podatki izbranega uporabnika. Administrator lahko te podatke poljubno ureja. Na skrajni desni strani vsakega uporabnika je tudi modra ikona z obliko smetnjaka in s klikom nanjo lahko po potrditvi odločitve administrator izbriše uporabnika iz sistema.

Izse	Primek	Uporabniško ime	E-Pošta	Ulica	Mesto	Telefonska številka	Poštna številka	Datum registracije	
JANEZ	Novak	JanezN	janezn@email.com	Ljubljanska ulica 1	Ljubljana	091999999	1000	28/06/2018	■
MARTIN	Kripan	Martink	martink@email.com	Ljubljanska ulica 3	Ljubljana	071777777	1000	28/06/2018	■
PETER	Klepec	PeterK	peterk@email.com	Ljubljanska ulica 2	Ljubljana	081888888	1000	28/06/2018	■

Slika 5.6: Administrativna forma spletne aplikacije

Poglavje 6

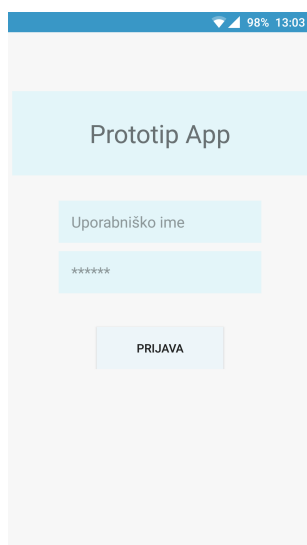
Predstavitev mobilne aplikacije

Za pravilno delovanje potrebuje mobilna aplikacija internetno povezavo in Bluetooth komunikacijo.

6.1 Dostop

Podpoglavje opisuje primer uporabe prijave v mobilno aplikacijo.

Ob zagonu mobilne aplikacije se mora za nadaljevanje uporabnik prijaviti (Slika 6.1). To lahko stori z uporabniškim imenom ali e-pošto in geslom. V primeru napačnih ali manjkajočih podatkov uporabnika ga aplikacija z obvestilom opozori.

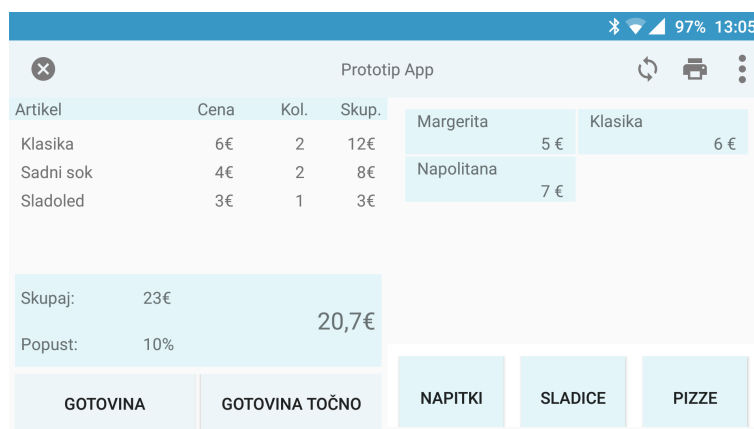


Slika 6.1: Forma prijava mobilne aplikacije

6.2 Forma za izdelavo računov

Podpoglavje opisuje formo za izdelavo računov in primer uporabe pregled izdelkov v ponudbi.

Po uspešni prijavi mobilna aplikacija uporabnika vpraša za dovoljenje uporabe in vklopa Bluetooth, v primeru, da uporabnik to dovoljenje zavrne se aplikacija zapre. Forma za izdelavo računov je ločena na zgornji, levi in desni del. Na zgornjem delu vidimo ime aplikacije in gumbe s funkcijami, ki jih bomo podrobneje opisali v naslednjih podpodpoglavjih. Na levem delu je prostor za dodane artikle na račun, njihovo ime, cena, količina, skupna cena artikla, skupna cena računa, popust na račun in končna cena računa. Pod njimi imamo gumba gotovina ter gotovina točno namenjena izdelavi računa. Desni del je namenjen prikazu kategorij in artiklov v ponudbi. Uporabniku se s pritiskom na katerikoli gumb kategorij spodaj (npr. napitki) prikažejo vsi artikli dodani v izbrani kategoriji.

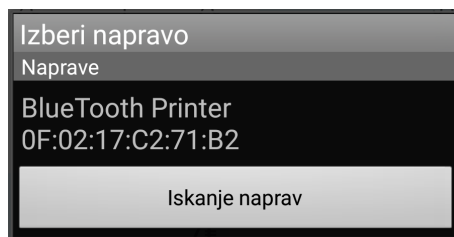


Slika 6.2: Forma za izdelavo računov mobilne aplikacije

6.2.1 Povezava ter odklop Bluetooth tiskalnika

Podpodpoglavje opisuje primera uporabe povezava s tiskalnikom prek tehnologije Bluetooth in odklop tiskalnika povezanega prek tehnologije Bluetooth.

S pritiskom na gumb z ikono tiskalnika desno zgoraj (Slika 6.2) se uporabniku pojavi meni z opcijami poveži tiskalnik ter odklopi tiskalnik. Ob pritisku na gumb poveži tiskalnik se prikaže okno za iskanje tiskalnikov kjer lahko uporabnik izbere že znane naprave ali prične z iskanjem novih s klikom na gumb iskanje naprav (Slika 6.3). Nato po iskanju uporabnik izbere napravo s katero se želi povezati in nato ob uspešni povezavi prejme obvestilo. V primeru, da želi uporabnik odklopiti tiskalnik zaradi priklopa novega ali iz kakšnega drugega razloga to lahko stori s pritiskom na gumb odklopi tiskalnik.

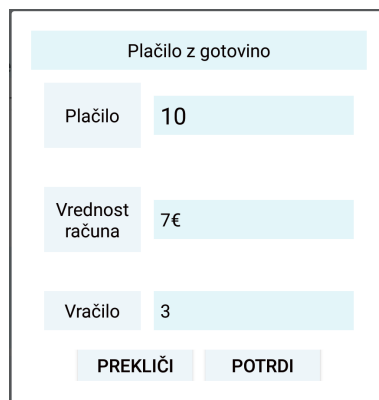


Slika 6.3: Mobilna aplikacija iskanje tiskalnika

6.2.2 Izdelava in tiskanje računa

Podpodpoglavje opisuje primere uporabe izdelava računa, izbira načina plačila, tisk računa in pošiljanje podatkov računa v podatkovno bazo.

Izdelava računa je zelo preprosta in intuitivna. Uporabnik preprosto izbere kategorijo desno spodaj in nato izbere artikel, ki se pojavi nad kategorijami. V primeru, da želi artikel iz računa odstraniti ga preprosto še enkrat izbere. Če želi spremeniti količino artikla na računu preprosto pritisne na številko pod stolpcem Kol. v isti vrstici kjer se nahaja zeleni artikel. Ko je z računom zadovoljen, ga lahko izdela s pritiskom na gumb gotovina ali gotovina točno. V primeru, da aplikacija ni povezana s tiskalnikom uporabnika opozori naj to stori. Gotovina točno je uporabljena takrat, ko stranka račun plača točno brez potrebe vračila gotovine. V nasprotnem primeru se uporabi gumb gotovina kjer uporabnik vnese znesek, katerega je stranka plačala in mu mobilna aplikacija pove, koliko denarja mora stranki vrniti (Slika 6.4). Ob uspešni potrditvi se podatki računa pošljejo v podatkovno bazo in račun natisne.



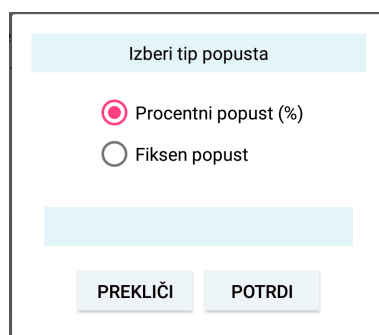
The screenshot shows a mobile application interface for cash payment. At the top, there is a title bar with the text "Plačilo z gotovino". Below the title bar, there are three input fields: "Plačilo" with the value "10", "Vrednost računa" with the value "7€", and "Vračilo" with the value "3". At the bottom of the screen, there are two buttons: "PREKLIČI" and "POTRDI".

Slika 6.4: Mobilna aplikacija vnos gotovine

6.2.3 Popust na račun

Podpodpoglavje opisuje primer uporabe dodaja popusta na račun.

V primeru, da uporabnik želi dodati popust na račun pred pritiskom na gumb gotovina ali gotovina točno izbere gumb s tremi pikami desno zgoraj (Slika 6.2) ter izbere opcijo dodaj popust na račun. Nato se uporabniku prikaže okno (Slika 6.5) in mu sporoči naj izbere tip in vnese vrednost popusta. Procentni popust zmanjša ceno računa za vneseni procent. Fiksni popust zmanjša ceno računa za vneseno vrednost. Ko je uporabnik z izbiro zadovoljen popust potrdi.



The screenshot shows a mobile application interface for selecting a discount type. At the top, there is a title bar with the text "Izberi tip popusta". Below the title bar, there are two radio button options: "Procentni popust (%)" which is selected, and "Fiksen popust". Below the options, there is a light blue rectangular area, likely a placeholder for a discount value input field. At the bottom of the screen, there are two buttons: "PREKLIČI" and "POTRDI".

Slika 6.5: Mobilna aplikacija vnos popusta

6.2.4 Ponovno tiskanje zadnjega računa

Podpodpoglavje opisuje primer uporabe ponovno tiskanje zadnjega računa.

V primeru, da uporabnik želi prejšnji natisnjeni račun tiskati ponovno lahko to stori s pritiskom na gumb s tremi pikami desno zgoraj (Slika 6.2) in izbere opcijo natisni prejšnji račun.

6.2.5 Nov račun

Podpodpoglavje opisuje primer uporabe izdelava novega računa.

Uporabnik lahko prejšnji račun počisti s klikom na gumb z ikono puščic desno zgoraj (Slika 6.2) in se tako lahko loti izdelave novega računa.

6.2.6 Odjava iz aplikacije

Podpodpoglavje opisuje primer uporabe odjava iz mobilne aplikacije.

V primeru, da se uporabnik želi odjaviti iz mobilne aplikacije to lahko stori s pritiskom na gumb z oznako X levo zgoraj (Slika 6.2). Uporabnika nato mobilna aplikacija odjavi ter vrne na prijavno okno.

Poglavje 7

Nadaljnji razvoj

Tip te platforme je tak, da se njen razvoj lahko nadaljuje v nedogled brez konca. Zato bomo opisali scenarije za nadaljnji razvoj, za katere menimo, da so najbolj pomembni in niso bili pokriti v prvi verziji prototipa te platforme.

7.1 Integracija s FURS

Eden izmed glavnih končnih ciljev te platforme je integracija s FURS, saj bi bila s tem platforma lahko dejansko uporabljena kot sistem POS blagajne skladne z zakonodajo. Seveda bi bilo pred tem platformo potrebno še daljši čas razvijati, da bi ustrezala vsem zadanim standardom in imela vse nujne funkcionalnosti, katere se od tega tipa platforme pričakuje.

7.2 Možnost plačila s plačilnimi karticami

Trenutna verzija prototipa te platforme ne podpira plačila s plačilnimi karticami. Razvoj te funkcionalnosti bi omogočilo večji obseg uporabe platforme, saj veliko strank ne plačuje z gotovino. V nadaljevanju bi lahko podprli tudi plačevanje s storitvijo PayPal in kriptovalutami.

7.3 Možnost izdelave posebnega tipa uporabniških računov za uporabnike mobilne aplikacije

Dokler je obseg poslovanja majhen to ni problem. Pri povečanem obsegu poslovanja, ko ima poslovni subjekt storitvene dejavnosti več zaposlenih, ki uporabljajo mobilno aplikacijo za izdelavo računov, bi bilo pametno implementirati funkcionalnost dodaje novih uporabnikov za isto ponudbo. Registrirani lastnik storitvene dejavnosti bi bil edini, ki se lahko na spletno aplikacijo prijavi ter vidi in ureja podatke. Slednji bi nato po prijavi imel možnost dodaje novih uporabnikov za svojo dejavnost. Ti uporabniki bi imeli dostop samo do mobilne aplikacije.

7.4 Avtomatska izdelava spletnih strani

Poslovnim subjektom storitvene dejavnosti bi lahko ponudili servis avtomatske izdelave spletne strani. Ob sprejemu servisa bi jih pozvali za vnos dodatnih podatkov potrebnih za izdelavo spletne strani, npr. kratek opis podjetja za izdelavo strani O Nas, urnik obratovanja. Pametno bi bilo tudi implementirati možnost dodaje slike za posamezni artikel.

7.5 Razširitev selekcije grafov

Trenutno grafi prikazujejo le promet skozi čas. Grafe bi lahko razširili, da bi prikazovali veliko več različnih podatkov. Ni nujno, da bi bilo vse prikazano z grafi. Lahko bi pokazali druge zanimive statistične podatke, kot na primer popularne kombinacije artiklov v nakupih, najbolj popularni izdelki, časovno območje največjega prometa, povprečna vsota, ki jo plača stranka na nakup. Seveda bi s tem morali vzporedno ustrezno prirediti podatkovno bazo z novimi atributi, katere bi želeli prikazati.

7.6 Mobilna aplikacija tudi za stranke

Mobilno aplikacijo bi lahko razširili in omogočili njeno uporabo tudi strankam. Stranka bi se lahko v mobilno aplikacijo registrirala in se po uspešni registraciji tudi vanjo prijavila. S tem bi lahko stranki omogočili sledenje njenih nakupov, pregled statističnih podatkov nakupov, predloge glede na preteklo opravljene nakupe in zanimanja, hitrejše in enostavnejše plačevanje opravljenih storitev, lažja uporaba in sledenje programa sheme zvestobe in enostavnejši pregled podatkov poslovnih subjektov storitvene dejavnosti kot na primer pregled delovnega urnika.

7.7 Rezervacija mize in ali menija

Stranka bi si lahko prek mobilne aplikacije tudi rezervirala mizo in si po možnosti tudi sestavila meni. Na ta način bi se stranka lahko izognila problemu zasedenih miz in zelo pogosto dolgem čakanju na hrano po naročilu v restavraciji.

7.8 Uber za prevoz na lokacijo

V primeru, da stranka trenutno nima možnosti prevoza na zeleno lokacijo bi ji lahko ponudili možnost prevoza na lokacijo z uporabo storitve podjetja Uber. Stranka bi torej imela možnost, da po prijavi v mobilno aplikacijo izbere lokacijo na katero želi prevoz in to opravi s storitvijo podjetja Uber.

Poglavje 8

Zaključek

Cilj te diplomske naloge je bila izdelava prototipa platforme za storitveno dejavnost, ki omogoča uporabnikom intuitivno izdelavo in urejanje ponudbe ter izdelavo in tiskanje računov. Zadan cilj je bil uresničen, vendar menimo, da je za uporabo platforme v praksi potrebno implementirati še veliko dodatnih funkcionalnosti. Potrebovali bi razvojno ekipo, saj je obseg platforme zelo obsežen. Vsi deli platforme so sestavljeni tako, da jih je mogoče enostavno spremeniti in nadgraditi. Sistem objave sprememb kode za zaledje in spletno aplikacijo je popolnoma avtomatiziran, kar omogoča razvijalcem le posvečanje kodiranju sprememb brez ostalih skrbi.

Literatura

- [1] Android studio. Dosegljivo: https://en.wikipedia.org/wiki/Android_Studio. [Dostopano: 5. 8. 2018].
- [2] Css. Dosegljivo: https://en.wikipedia.org/wiki/Cascading_Style_Sheets. [Dostopano: 5. 8. 2018].
- [3] Css3. Dosegljivo: <https://www.htmlgoodies.com/html5/css/whats-new-in-css3.html>. [Dostopano: 5. 8. 2018].
- [4] Davčne blagajne. Dosegljivo: <https://mladipodjetnik.si/novice-in-dogodki/novice/davcne-blagajne-kdaj-in-za-koga>. [Dostopano: 5. 8. 2018].
- [5] Express.js intruduction. Dosegljivo: https://www.tutorialspoint.com/expressjs/expressjs_overview.htm. [Dostopano: 3. 8. 2018].
- [6] Git. Dosegljivo: <https://en.wikipedia.org/wiki/Git>. [Dostopano: 5. 8. 2018].
- [7] Alexis Goldstein, Louis Lazaris, and Estelle Weyl. *HTML5 & CSS3 for the Real World*. sitepoint, 2015.
- [8] Heroku. Dosegljivo: <https://en.wikipedia.org/wiki/Heroku>. [Dostopano: 3. 8. 2018].
- [9] Html. Dosegljivo: <https://en.wikipedia.org/wiki/HTML>. [Dostopano: 4. 8. 2018].

-
- [10] Html5. Dosegljivo: <https://en.wikipedia.org/wiki/HTML5>. [Dostopano: 4. 8. 2018].
- [11] Java. Dosegljivo: <https://www.techopedia.com/definition/3927/java>. [Dostopano: 4. 8. 2018].
- [12] Javascript. Dosegljivo: <https://en.wikipedia.org/wiki/JavaScript>. [Dostopano: 4. 8. 2018].
- [13] Neil Middleton and Richard Schneeman. others. 2013. heroku: Up and running.
- [14] Patrick Niemeyer and Jonathan Knudsen. *Learning java*. "O'Reilly Media, Inc.", 2005.
- [15] Node.js intruduction. Dosegljivo: https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm. [Dostopano: 3. 8. 2018].
- [16] Operacijski sistem android. Dosegljivo: [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system)). [Dostopano: 5. 8. 2018].
- [17] Point of sale. Dosegljivo: https://en.wikipedia.org/wiki/Point_of_sale. [Dostopano: 3. 8. 2018].
- [18] Powerdesigner. Dosegljivo: <https://en.wikipedia.org/wiki/PowerDesigner>. [Dostopano: 5. 8. 2018].
- [19] React. Dosegljivo: [https://en.wikipedia.org/wiki/React_\(JavaScript_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library)). [Dostopano: 5. 8. 2018].
- [20] React. Dosegljivo: <https://docs.microsoft.com/en-us/azure/devops/git/what-is-git>. [Dostopano: 2. 8. 2018].
- [21] Ravishankar Somasundaram. *Git: Version control for everyone*. Packt Publishing Ltd, 2013.
- [22] Sublimetext. Dosegljivo: https://en.wikipedia.org/wiki/Sublime_Text. [Dostopano: 3. 8. 2018].

-
- [23] Myer Thomas. No nonsense xml web development with php, sitepoint pty ltd. *Collingwood, Australia*, 2005.
- [24] What is heroku? Dosegljivo: <https://stackoverflow.com/questions/11008787/what-exactly-is-heroku>. [Dostopano: 3. 8. 2018].
- [25] What is java? Dosegljivo: <https://www.edureka.co/blog/what-is-java/>. [Dostopano: 4. 8. 2018].
- [26] What is javascript? Dosegljivo: <https://www.makeuseof.com/tag/what-is-javascript/>. [Dostopano: 4. 8. 2018].
- [27] What is react. Dosegljivo: <https://www.edureka.co/blog/what-is-react/>. [Dostopano: 5. 8. 2018].
- [28] Xml. Dosegljivo: <https://en.wikipedia.org/wiki/XML>. [Dostopano: 1. 8. 2018].
- [29] Xml introduction. Dosegljivo: <https://www.sitepoint.com/really-good-introduction-xml/>. [Dostopano: 1. 8. 2018].