

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Matej Vovko

**Prototip informacijskega sistema za shranjevanje
računov v digitalni obliki**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI 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.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Kupci ob nakupu tehničnega izdelka ali drugega tipa izdelka z garancijo prejmejo račun, ki lahko velja tudi kot garancija. Lahko pa prejmejo garancijo tudi posebej. V vsakem primeru pa račun ali garancijo večkrat izgubijo in imajo potem težave pri uveljavljanju le-te.

Zasnujte prototip informacijskega sistema, ki kupcu ob nakupu preko mobilne aplikacije omogoča na enostaven način shraniti račun in/ali garancijo v digitalni obliki.

Kazalo

Povzetek

Abstract

1. Uvod.....	1
2. Opis uporabljenih tehnologij in razvojnih orodij	3
2.1. Storitvena platforma Microsoft Azure	3
2.2. Razvojno okolje Microsoft Visual Studio	5
2.3. Mobilni operacijski sistem Android	7
2.4. Razvojno okolje Android Studio	10
3. Analiza problema in definicija zahtev informacijskega sistema.....	13
3.1. Diagram primerov uporabe za spletno aplikacijo	13
3.2. Diagram primerov uporabe za mobilno aplikacijo	14
3.3. Podatkovni model	15
4. Načrtovanje informacijskega sistema.....	21
4.1. Arhitektura informacijskega sistema	21
4.2. Elementi informacijskega sistema	22
4.3. Povezava: spletna aplikacija in podatkovna baza	26
4.4. Komunikacija: spletna aplikacija in podatkovna baza.....	28
4.5. Komunikacija: mobilna aplikacija in spletni servis	31
4.6. Komunikacija: konzolna aplikacija in spletni servis	33
5. Delovanje informacijskega sistema s primeri uporabe	37
5.1. Primeri uporabe: spletna aplikacija.....	37
5.2. Primeri uporabe: mobilna aplikacija	55
6. Sklepne ugotovitve.....	67
6.1. Ideje za nadaljnji razvoj	68
Literatura	69
Priloge.....	71

Kazalo slik

Slika 2.1: Spletni portal Microsoft Azure.....	3
Slika 2.2: Razvojno okolje Microsoft Visual Studio.....	6
Slika 2.3: Domači zaslon Androida 8.1 Oreo [2].	8
Slika 2.4: Razvojno okolje Android Studio.....	10
Slika 4.1: Shema arhitekture informacijskega sistema.....	21
Slika 4.2: Izbrani brezplačni plan na Microsoft Azure.....	22
Slika 4.3: Pregled osnovnih podatkov strežnika MS SQL Server na Microsoft Azure.....	23
Slika 4.4: Pregled osnovnih podatkov podatkovne baze na Microsoft Azure.....	24
Slika 4.5: Pregled nastavitve WebJobs v spletni aplikaciji na Microsoft Azure.....	25
Slika 4.6: Pregled konfiguracije mobilne aplikacije v Android Studiu.....	26
Slika 4.7: Nastavitve povezave do podatkovne baze v spletni aplikaciji.....	27
Slika 4.8: Definicija ukaza za pridobivanje podatkov v gradniku SqlDataSource.....	29
Slika 4.9: Ukazi za upravljanje s podatki podatkovne baze iz kode C# v ozadju spletne strani.	31
Slika 4.10: Razred ServiceCaller v mobilni aplikaciji.....	33
Slika 4.11: Konzolna aplikacija za pošiljanje zahtevkov spletnemu servisu za pošiljanje e-poštnih obvestil o izteku garancije.....	35
Slika 5.1: Prijavna stran spletne aplikacije IS.....	37
Slika 5.2: Stran za ponastavitev gesla v spletni aplikaciji IS.....	38
Slika 5.3: Upravljanje uporabnikov v spletni aplikaciji IS.....	39
Slika 5.4: Administrator lahko v spletni aplikaciji IS doda novega uporabnika ali obstoječemu ponastavi geslo.....	40
Slika 5.5: Upravljanje obstoječih in dodajanje novih trgovin v spletni aplikaciji IS.....	41
Slika 5.6: Upravljanje izdelkov v ponudbi v spletni aplikaciji IS.....	42
Slika 5.7: Dodajanje izdelka v ponudbo v spletni aplikaciji IS.....	43
Slika 5.8: Dodajanje dodatnih gradiv o izdelku v spletni aplikaciji IS.....	44
Slika 5.9: Pregled izdanih računov v spletni aplikaciji IS.....	45
Slika 5.10: Ustvarjanje računa v spletni aplikaciji IS.....	45

Slika 5.11: Prikaz kod ustvarjenega računa v spletni aplikaciji IS.	46
Slika 5.12: Pregled podrobnosti izdanega računa s strani trgovca v spletni aplikaciji IS.....	47
Slika 5.13: Upravljanje računov uporabnika v spletni aplikaciji IS.	48
Slika 5.14: Pregled in urejanje podrobnosti računa dodanega prek slike v spletni aplikaciji IS.	49
Slika 5.15: Pregled podrobnosti računa dodanega prek kode v spletni aplikaciji IS.	50
Slika 5.16: Prikaz dodatnih gradiv izdelkov z vseh računov v spletni aplikaciji IS.	51
Slika 5.17: Urejanje uporabniškega profila in sprememba gesla v spletni aplikaciji IS.....	52
Slika 5.18: Pregled izdelkov za nakup v spletni aplikaciji IS.	53
Slika 5.19: Pregled podatkov o izdelku iz ponudbe v spletni aplikaciji IS.....	54
Slika 5.20: Prijava, ponastavitev gesla in registracija v mobilni aplikaciji IS.....	55
Slika 5.21: Pregled računov uporabnika v mobilni aplikaciji IS.	56
Slika 5.22: Dodajanje računa prek slike v mobilni aplikaciji IS.....	57
Slika 5.23: Dodajanje računa prek kode v mobilni aplikaciji IS.	58
Slika 5.24: Pregled in urejanje podrobnosti računa dodanega prek slike v mobilni aplikaciji IS.	59
Slika 5.25: Pregled podrobnosti računa dodanega prek kode v mobilni aplikaciji IS.	60
Slika 5.26: Meni v mobilni aplikaciji IS.	61
Slika 5.27: Prikaz dodatnih gradiv izdelkov z vseh računov v mobilni aplikaciji IS.	62
Slika 5.28: Urejanje uporabniškega profila in sprememba gesla v mobilni aplikaciji IS.	63
Slika 5.29: Pregled izdelkov za nakup v mobilni aplikaciji IS.	64
Slika 5.30: Pregled podatkov o izdelku iz ponudbe v mobilni aplikaciji IS.	65

Kazalo tabel

Tabela 2.1: Delež uporabe različic Androida, dne 25. 7. 2018 [8].....	9
Tabela 3.1: Uporabljeni podatkovni tipi v podatkovni bazi [7].	16

Seznam uporabljenih kratic

kratica	angleško	slovensko
ADT	Android Development Tools	razvojna orodja za Android
API	application programming interface	aplikacijski programski vmesnik
APK	Android application package	aplikacijski paket za Android
ASP	Active Server Pages	aktivne strežniške strani
AVD	Android Virtual Device	virtualna naprava Android
CSS	Cascading Style Sheets	kaskadne stilske podloge
DB	database	podatkovna baza
DTU	Database Transaction Unit	transakcijska enota podatkovne baze
GB	gigabyte	gigabajt
HTML	Hypertext Markup Language	jezik za označevanje nadbesedila
HTTP	Hypertext Transfer Protocol	protokol za prenos nadbesedila
HTTPS	Hypertext Transfer Protocol Secure	protokol za varovan prenos nadbesedila
IS	information system	informacijski sistem
JPEG	Joint Photographic Experts Group	skupina združenih strokovnjakov fotografije
JSON	JavaScript Object Notation	zapis objektov JavaScript
MARS	Multiple Active Result Sets	večkratni aktivni nabori rezultatov
MB	megabyte	megabajt
MD5	message-digest algorithm 5	algoritem za povzemanje sporočil 5
MS	Microsoft	Microsoft
OTA	over-the-air	po zraku
PDF	Portable Document Format	prenosni format dokumenta
PHP	Personal Home Page	osebna domača stran
PNG	Portable Network Graphics	prenosljiva spletna grafika
QR	quick response	hiter odziv
REST	Representational State Transfer	predstavitveni prenos stanja
SDK	software development kit	komplet za razvoj programske opreme
SMB	Server Message Block	strežniški sporočilni blok
SQL	Structured Query Language	strukturirani povpraševalni jezik
SSL	Secure Sockets Layer	varna plast vtičnic
TLS	Transport Layer Security	varnost transportne plasti
URL	Uniform Resource Locator	enolični krajevnik vira
WCF	Windows Communication Foundation	komunikacijska osnova Windows
WPF	Windows Presentation Foundation	prikazovalna osnova Windows

XML

Extensible Markup Language

razširljivi označevalni jezik

XSL

Extensible Stylesheet Language

razširljivi jezik stilskih podlog

Povzetek

Naslov: Prototip informacijskega sistema za shranjevanje računov v digitalni obliki

Avtor: Matej Vovko

Diplomska naloga predstavlja prototip informacijskega sistema za shranjevanje računov v digitalni obliki. Prototip informacijskega sistema uporabnikom omogoča dodajanje klasičnih papirnatih računov in računov ustvarjenih znotraj razvitega informacijskega sistema. Uporabnikom ponuja obvestila pred iztekom garancije na posameznem računu ali posameznem izdelku, v kolikor je datum garancije vpisan. Prototip informacijskega sistema sestoji iz dveh glavnih delov: spletne in mobilne aplikacije. Spletna aplikacija je bila razvita na Microsoftovi tehnologiji ASP.NET in skupaj s podatkovno bazo delujeta na platformi Microsoft Azure. Mobilna aplikacija je bila razvita za operacijski sistem Android. V tem diplomskem delu sta predstavljeni platformi Microsoft Azure in Android, ki omogočata delovanje tega informacijskega sistema. Predstavljeni sta tudi uporabljeni razvojni okolji Microsoft Visual Studio in Android Studio. Razložen je problem, ki je privedel do razvoja tega informacijskega sistema. Predstavljene so uporabniške zahteve sistema in podatkovni model. Sledi poglavje o arhitekturi dotičnega prototipa informacijskega sistema in medsebojni komunikaciji posameznih elementov. Nenazadnje je delovanje predstavljeno s praktičnimi primeri uporabe. Na koncu pa sledijo še ideje za morebiten nadaljnji razvoj.

Ključne besede: informacijski sistem, mobilna aplikacija, račun, garancija, Azure, Android.

Abstract

Title: Prototype of information system for archiving invoices in digital form

Author: Matej Vovko

This thesis is about a prototype of information system for archiving invoices in digital form. The prototype of information system enables users to add classic paper invoices and invoices created inside this information system. It offers users notifications before expiration of guarantee on individual invoice or individual product if the date of guarantee is entered. The prototype of information system consists of two major parts: web and mobile application. The web application was developed with Microsoft's technology ASP.NET and together with the database they run on the Microsoft Azure platform. The mobile application was developed for the operating system Android. The platforms Microsoft Azure and Android, which enable running this information system, are presented in this thesis. The development environments Microsoft Visual Studio and Android Studio are also presented here. The problem which led to the development of this information system is explained. The user requirements of the system and the data model are presented. A chapter about the prototype information system's architecture and the communication between individual elements follows. The system's operation with practical examples of usage is presented next. Finally, ideas for possible future development are written.

Keywords: information system, mobile application, invoice, guarantee, Azure, Android.

1. Uvod

Dandanes se vsakodnevno srečujemo z obilico računov za najrazličnejše nakupe. Nekateri od teh so manj pomembni in nam njihova izguba ne predstavlja skrbi. Spet drugi, predvsem tisti, ki so bili izdani ob nakupu dražjih izdelkov, pa imajo večjo vrednost. Na podlagi računa lahko dokazujemo nakup določenega izdelka pri morebitni uveljavitvi garancije. Poleg tega, da lahko račun kaj kmalu izgubimo, pa klasični papirnati računi sčasoma zbledijo do neberljivosti. Sploh, če niso ustrezno shranjeni in so izpostavljeni močni svetlobi.

Razvit je bil prototip informacijskega sistema, ki uporabnikom omogoča shranjevanje računov na enem mestu. Tako klasičnih papirnatih, ki jih prejmemo ob vsakem nakupu, kot tudi tistih, ki so ustvarjeni znotraj tega informacijskega sistema. V sklopu tega diplomskega dela razvit prototip informacijskega sistema uporabnikom ponuja spletno in mobilno aplikacijo. Trgovinam je na voljo le spletni del, ki trgovcu z administratorskimi pravicami omogoča dodajanje izdelkov v ponudbo in ustvarjanje računov. Običajnemu trgovcu pa samo slednje. Registrirani uporabnik kot kupec lahko prek zajete fotografije dodaja klasične papirnate račune ali pa prek kode računa digitalne račune ustvarjene znotraj tega informacijskega sistema. Z uporabo mobilne aplikacije pa je omogočeno dodajanje digitalnih računov tudi prek kode QR, ki se samodejno generira na strani trgovca ob zaključku ustvarjanja računa, če seveda uporablja dotični informacijski sistem. En mesec pred iztekom garancije je uporabnik prek e-poštnega sporočila obveščen o bližajočem se izteku garancije na določenem računu, če je to papirnati račun izdan izven tega informacijskega sistema oz. na določenem izdelku, če je to digitalni račun ustvarjen znotraj tega informacijskega sistema. Obvestilo je seveda poslano le, če je na računu oz. izdelku določen datum izteka garancije. Obstaja še ena vrsta uporabnika. To je sistemski administrator, ki lahko dodaja nove uporabnike, ureja podatke o uporabniških računih, vključno z vrsto uporabniških pravic, določi v kateri trgovini deluje določen trgovec in nenazadnje dodaja trgovine v informacijski sistem.

Spletna aplikacija je bila razvita na Microsoftovem ogrodju ASP.NET znotraj razvojnega okolja Microsoft Visual Studio in deluje znotraj Microsoftove storitvene platforme Microsoft Azure. Ravno tako na platformi Azure gostuje podatkovna baza, ki deluje na strežniku Microsoft SQL Server. Spletni servis, ki omogoča komunikacijo med samo podatkovno bazo in mobilno aplikacijo, je del spletne aplikacije in je posledično tudi ta dostopen prek platforme Azure. Mobilna aplikacija je bila razvita za operacijski sistem Android. Razvita je bila znotraj razvojnega okolja Android Studio. Obstaja pa še ločen spletni servis, ravno tako del spletne aplikacije, ki skrbi za pošiljanje e-poštnih obvestil pred iztekom garancije. Za njegovo proženje je zadolžena konzolna aplikacija, ki se izvede enkrat dnevno znotraj Microsoft Azure.

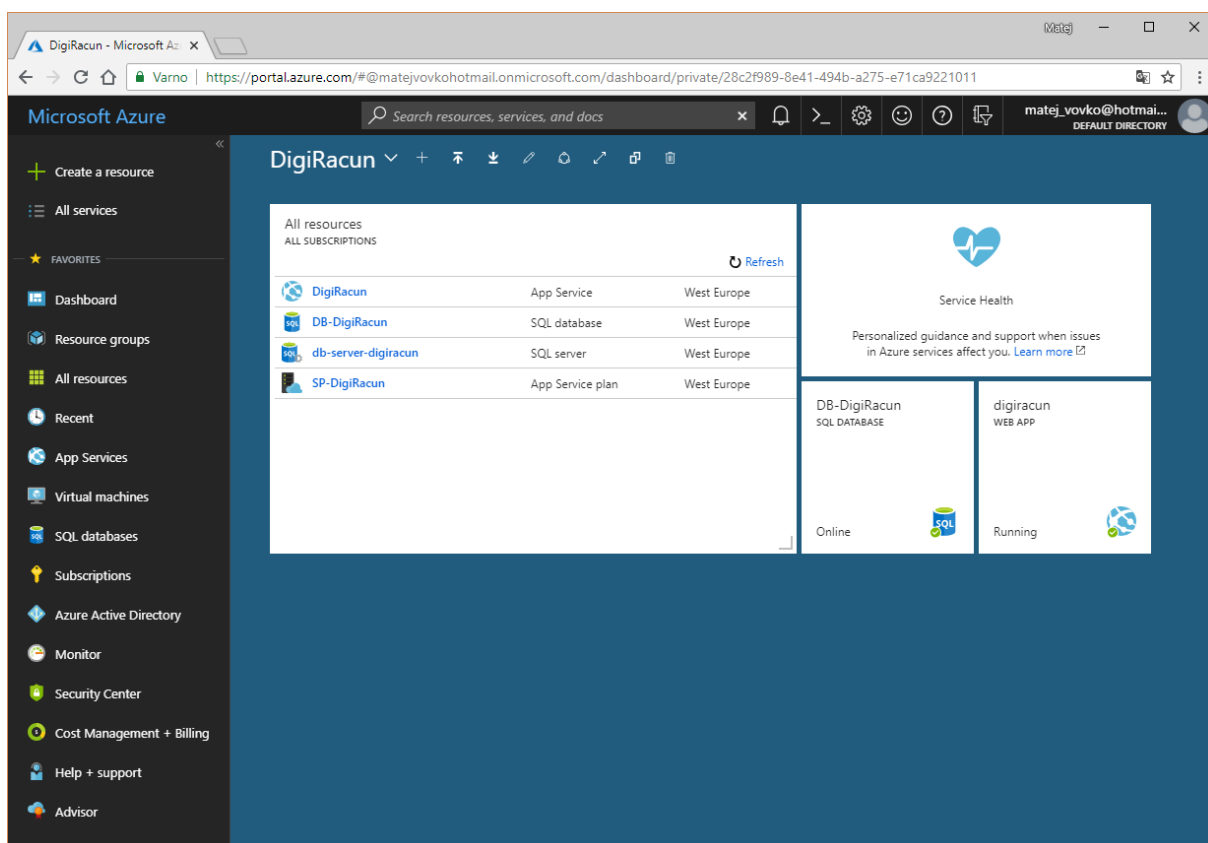
V 2. poglavju sledi predstavitev tehnologij in platform, ki so omogočile razvoj končnega izdelka. V tem poglavju je zajeta tudi predstavitev razvojnih orodij, ki so bila uporabljena pri

razvoju informacijskega sistema. V 3. poglavju so prikazani diagrami primerov uporabe. Opisan je tudi podatkovni model z razlago zgradbe same podatkovne baze. 4. poglavje zajema arhitekturo informacijskega sistema in predstavitev vseh pripadajočih elementov. Poglavje govori tudi o medsebojni komunikaciji posameznih elementov. V 5. poglavju je na praktičnih primerih uporabe in s podporo zaslonskih posnetkov prikazano delovanje tega informacijskega sistema. Zadnje, 6. poglavje je namenjeno sklepnim ugotovitvam in idejam za morebiten nadaljnji razvoj z novimi funkcionalnostmi in izboljšavami obstoječih. Sledita le še poglavji s seznamom uporabljenih literature ter prilogami.

2. Opis uporabljenih tehnologij in razvojnih orodij

2.1. Storitvena platforma Microsoft Azure

Microsoft Azure (Slika 2.1) je oblachna storitvena platforma za razvoj, testiranje in upravljanje spletnih aplikacij in servisov. Prek zakupljene naročnine ponuja programsko opremo, platformo in infrastrukturo kot storitev. Podpira veliko število programskih jezikov, orodij in ogrodij, tako Microsoftovih, kot tistih od drugih ponudnikov. [10]



Slika 2.1: Spletni portal Microsoft Azure.

Storitev Microsoft Azure uporablja poseben istoimenski operacijski sistem za delovanje Microsoftovih podatkovnih centrov, ki upravljajo računske in shranjevalne vire. Storitve je predstavljena kot »oblačni sloj«, ki temelji na sistemih Windows Server in prilagojeni različici virtualizacijske tehnologije Hyper-V za zagotavljanje virtualizacijskih storitev.

Microsoft Azure Fabric Controller skrbi za zagotavljanje zanesljivosti storitve pri morebitni odpovedi kakšnega izmed strežnikov v podatkovnem centru. Uporabnikom prek spletnega vmesnika omogoča tudi upravljanje z dodeljevanjem spomina in uravnavanjem obremenitev.

Microsoft Azure uporabnikom ponuja programske vmesnike za komunikacijo s storitvijo, ki temeljijo na protokolih REST in HTTP ter označevalnem jeziku XML. Na voljo je tudi

knjižnica za komunikacijo s storitvijo. Omogočeno je povezovanje z razvojnima okoljema Microsoft Visual Studio in Eclipse ter sistemom za sledenje sprememb Git. Poleg upravljanja gostujočih storitev prek programskih vmesnikov je na voljo tudi spletni portal. Uporabnikom omogoča upravljanje obstoječih virov, spreminjanje nastavitev, ustvarjanje novih in spremljanje osnovnih podatkov iz aktivnih virtualnih naprav in storitev.

Za upravljanje z viri se uporabljata dva modela. Pri prvem, klasičnem, se vsak vir (virtualna naprava, strežnik za podatkovno bazo SQL ...) upravlja ločeno. Pri drugem (t. i. *Azure Resource Manager*) pa se ustvarijo skupine medsebojno povezanih storitev, ki se upravljajo in nadzorujejo skupaj.

Microsoft v sklopu platforme Azure ponuja več kot 600 storitev. Nekaj izmed teh je navedenih v naslednjih kategorijah.

Računske storitve:

- virtualne naprave (infrastruktura kot storitev) – Microsoft Windows in Linux,
- aplikacijske storitve (platforma kot storitev) – objava in upravljanje spletnih strani,
- gostovanje spletnih strani ustvarjenih s tehnologijami ASP.NET, PHP, Node.js, Python,
- aplikacije, ki se glede na urnik ali na zahtevo prožijo v ozadju in so del aplikacijskih storitev (t. i. WebJobs).

Mobilne storitve:

- potisna obvestila,
- zbiranje analitike uporabe.

Podatkovne storitve:

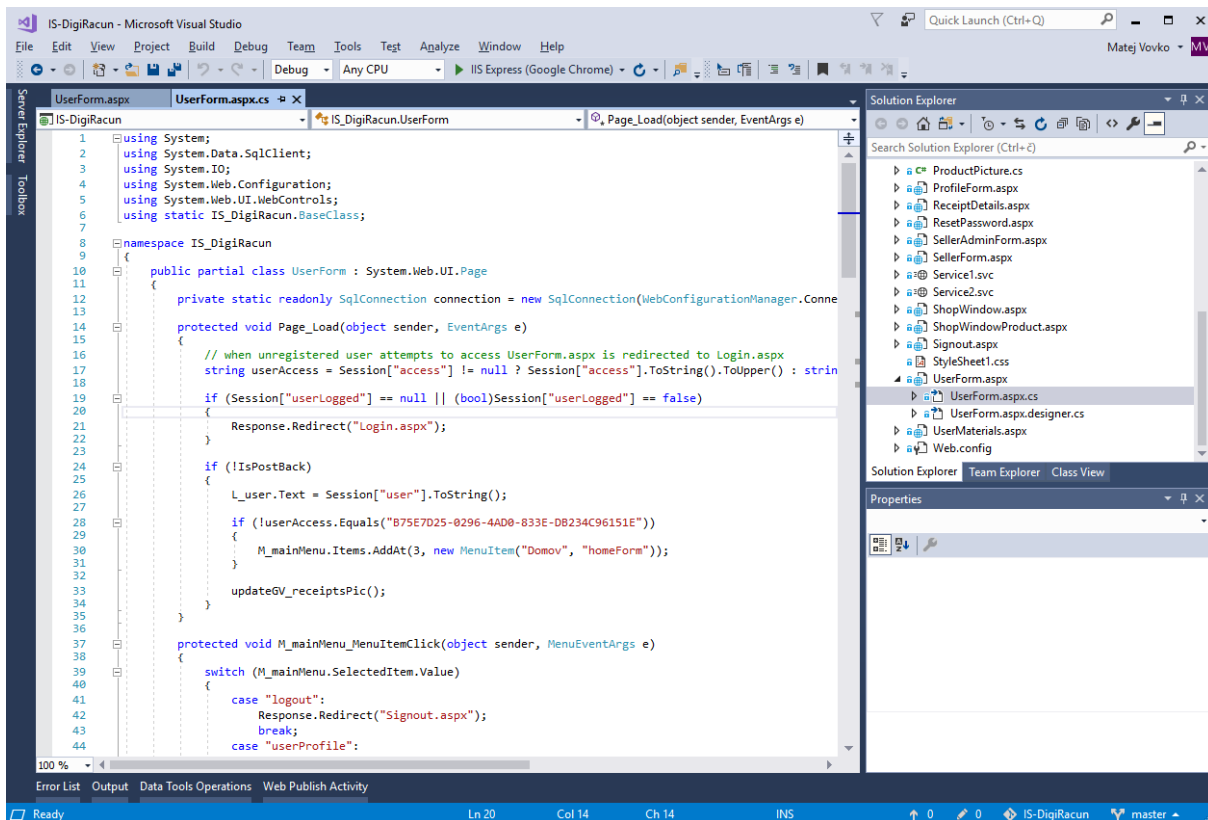
- shranjevanje in dostop do podatkov prek spletnih servisov REST in programskih vmesnikov API,
- datotečne storitve za shranjevanje in dostop do podatkov prek spletnih servisov REST ali protokola SMB,
- različne vrste relacijskih in nerelacijskih podatkovnih baz (Cosmos DB, NoSQL, MS SQL Server),
- podatkovna skladišča,
- analitične storitve večjih količin podatkov.

Druge storitve:

- različne vrste sporočanja,
- medijske storitve za kodiranje, zaščito vsebine in prenos v realnem času,
- storitve za dostavo multimedijskih vsebin (slika, video, zvok, aplikacije, statistične datoteke),
- storitve za razvijalce (spremljanje obremenitev, sodelovanje pri razvoju z uporabo Visual Studio Team Services),
- strojno učenje,
- internet stvari.

2.2. Razvojno okolje Microsoft Visual Studio

Microsoft Visual Studio (Slika 2.2) je integrirano razvojno okolje, ki se uporablja za razvoj namiznih programov, spletnih strani, spletnih aplikacij, spletnih servisov in mobilnih aplikacij. Visual Studio uporablja Microsoftove razvojne platforme, kot so Windows API (programski vmesniki na voljo v operacijskih sistemih Windows), Windows Forms (grafična knjižnica za izdelovanje uporabniških vmesnikov), Windows Presentation Foundation (skrajšano WPF, t. j. grafični podsistem za izris uporabniškega vmesnika), Windows Store (platforma za distribucijo digitalnih vsebin, npr. aplikacij). [11]



Slika 2.2: Razvojno okolje Microsoft Visual Studio.

Visual Studio ponuja urejevalnik programske kode s samodejnim dopolnjevanjem med pisanjem programske kode. Kot je za sodobna razvoja okolja običajno, razvijalcu ponuja razhroščevalnik za lažje iskanje napak v kodi. Vanj so vgrajena tudi orodja za razvoj aplikacij z grafičnimi uporabniškimi vmesniki, spletnih strani, načrtovanje shem podatkovnih baz in načrtovanje razredov pri objektne programiranju. Ponuja tudi orodje za spremljanje porabe delovnega spomina, časa delovanja same aplikacije in drugih parametrov med poganjanjem razvite aplikacije. Z uporabo dodatkov se da njegove funkcionalnosti še razširiti in si tako olajšati delo. Visual Studio podpira sisteme za sledenje sprememb programske kode kot sta Git in njihov lastni Team Foundation Server.

Microsoftovo integrirano razvojno okolje podpira več programskih jezikov:

- C#,
- C,
- C++,
- Visual Basic .NET,
- JavaScript,
- TypeScript,

- XML,
- HTML,
- CSS.

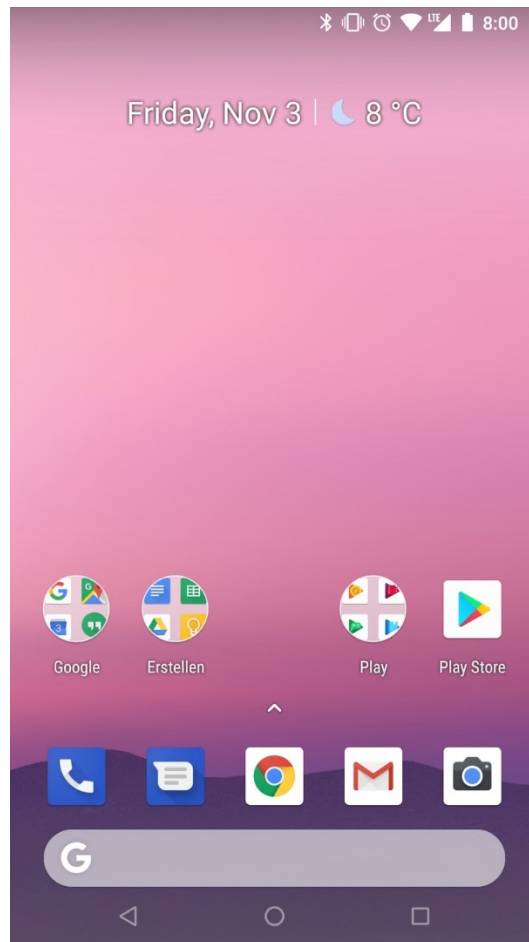
Prek dodatkov pa se nabor podprtih programskih jezikov lahko še poveča.

Trenutna različica Visual Studia 2017 je na voljo v treh izvedenkah [19]:

- Visual Studio Community 2017 (brezplačen, namenjen odprtokodnim projektom, samostojnim razvijalcem in študentom),
- Visual Studio Professional 2017 (namenjen uporabi za profesionalne namene z naročnino za manjše razvojne ekipe),
- Visual Studio Enterprise 2017 (namenjen uporabi za profesionalne namene, tudi za večje razvojne ekipe).

2.3. Mobilni operacijski sistem Android

Google je razvijalec odprtokodnega mobilnega operacijskega sistema Android (Slika 2.3). Slednji temelji na modificiranem jedru Linux in drugi odprtokodni programski opremi. Primarno je razvit za naprave z zasloni občutljivimi na dotik, kot so pametni mobilni telefoni in tablični računalniki. Google z Androidom čedalje bolj prodira tudi na druga področja. Android TV je namenjen pametnim televizijskim sprejemnikom, Android Auto zaslonom v avtomobilih in Wear OS pametnim uram. Prilagojene različice tega operacijskega sistema so na voljo še na igralnih konzolah, kamerah in razni drugi elektroniki. Obstajajo celo različice za namizne in prenosne računalnike. [1]



Slika 2.3: Domači zaslon Androida 8.1 Oreo [2].

Običajno je Android opremljen s paketom Googlovih aplikacij, kot so odjemalec elektronske pošte Gmail, Googlov iskalnik, trgovina z aplikacijami Google Play Store, zemljevidi Google Maps in drugimi. Seveda pa posamezni proizvajalci lahko te aplikacije nadomestijo s svojimi.

Od leta 2011 je Android vodilni operacijski sistem glede na število prodanih pametnih telefonov. Težava naprav, ki jih poganja ta operacijski sistem je posodabljanje. Kljub temu da večje izdaje novih različic Androida izhajajo na letni ravni, večina naprav poganja eno izmed starejših različic (Tabela 2.1). Posodobitve operacijskega sistema se sicer namestijo samodejno po principu OTA (t. j. *over-the-air*). To pomeni, da so posodobitve poslane z osrednje lokacije k vsem uporabnikom. Težava z zastarelimi različicami Androida nastane zaradi najrazličnejše strojne opreme, ki jo uporabljajo dotične naprave. Varnosti popravki dosežejo končne uporabnike z velikimi zamudami, ali pa jih sploh ne. Izjemi sta Googlovi liniji naprav Nexus in Pixel, ki poganjata Android brez sprememb, kakršen je v osnovi. Vsaka nova izdaja Androida mora biti namreč prilagojena za posamezno napravo, proizvajalci pa dajejo prednost novejšim napravam. Dodatne zakasnitve povzročajo tudi telekomunikacijski ponudniki, ki izvajajo dodatne prilagoditve in testiranja, ko prejmejo posodobitve s strani proizvajalcev. Obstajajo

tudi primeri, ko posodobitev ni mogoča, ker proizvajalci posameznih komponent strojne opreme ne izdajo novejših gonilnikov. Z novejšimi različicami Androida se stanje izboljšuje, saj je Google z Androidom 8.0 predstavil spremenjeno arhitekturo ogrodja, ki proizvajalcem omogoča enostavnejše, hitrejše in cenejše posodabljanje starejših naprav. Z novim vmesnikom je bil ločen proizvajalčev del (specifičen za posamezno napravo) od samega ogrodja operacijskega sistema Android.

Različica	Kodno ime	API	Delež
2.3.3 – 2.3.7	Gingerbread	10	0,2 %
4.0.3 – 4.0.4	Ice Cream Sandwich	15	0,3 %
4.1.x	Jelly Bean	16	1,2 %
4.2.x		17	1,9 %
4.3		18	0,5 %
4.4	KitKat	19	9,1 %
5.0	Lollipop	21	4,2 %
5.1		22	16,2 %
6.0	Marshmallow	23	23,5 %
7.0	Nougat	24	21,2 %
7.1		25	9,6 %
8.0	Oreo	26	10,1 %
8.1		27	2 %

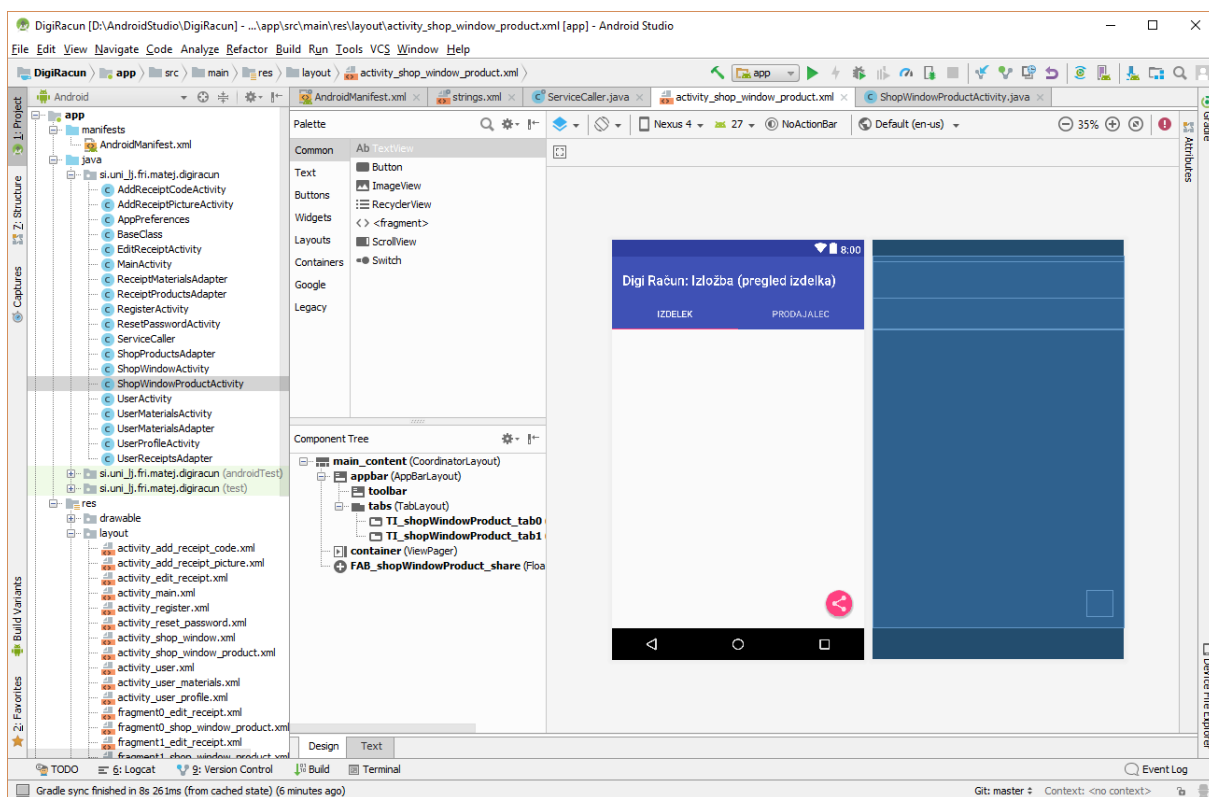
Tabela 2.1: Delež uporabe različic Androida, dne 25. 7. 2018 [8].

Dodatne aplikacije razširjajo zmožnosti naprav z operacijskim sistemom Android. Za izdelovanje aplikacij je na voljo Android SDK (t. i. *software development kit* oz. komplet orodij za razvoj programske opreme) [13]. Le-ta vključuje različna razvojna orodja, kot so razhroščevalnik, programske knjižnice, virtualno napravo za poganjanje Androida na osebem računalniku (t. i. *emulator*), dokumentacijo, vodiče in primere programske kode. Sprva je bilo s strani Googla uradno podprto razvojno okolje Eclipse, kasneje pa je Google izdal Android Studio. To razvojno okolje temelji na integriranem razvojnem okolju IntelliJ IDEA. Za razvoj aplikacij so na voljo tudi druga razvojna okolja. Običajno so aplikacije za Android napisane v programskem jeziku Java. Slednjega se lahko uporabi tudi v kombinaciji s programskima jezikoma C in C++. Za razvoj aplikacij sta podprta tudi programska jezika Go in Kotlin. Nameščanje aplikacij je mogoče prek datoteke APK ali prek ene izmed aplikacijskih trgovin. Običajno je primarna trgovina z aplikacijami Google Play Store. Ta omogoča nameščanje, posodabljanje in odstranjevanje aplikacij. Zaradi odprtosti operacijskega sistema Android pa so

na voljo tudi druge aplikacijske trgovine. Nekatere aplikacije je tako moč najti v drugih trgovinah, saj zaradi različnih razlogov niso na voljo v Google Play Store.

2.4. Razvojno okolje Android Studio

Android Studio (Slika 2.4) je integrirano razvojno okolje za razvoj aplikacij za mobilni operacijski sistem Android. Je uradno podprto razvojno okolje s strani Googla. Temelji na razvojnem okolju IntelliJ IDEA proizvajalca JetBrains. Na voljo je za različne operacijske sisteme: Windows, macOS in tiste, ki temeljijo na jedru Linux. Android Studio nadomešča Eclipse ADT (t. j. *Android Development Tools*), kot primarno razvojno okolje za operacijski sistem Android. [3]



Slika 2.4: Razvojno okolje Android Studio.

Android Studio omogoča razvoj aplikacij z odprtokodnim sistemom za grajenje aplikacij Gradle [9]. Slednji uporablja direktni aciklični graf za ugotavljanje vrstnega reda nalog. Android Studio razvijalcu ponuja orodja za prikaz učinkovitosti delovanja, uporabnosti, združljivosti z različnimi različicami Androida ter drugih težav, ki se lahko pojavijo pri poganjanju aplikacije. Vgrajeno ima tudi orodje ProGuard za podpisovanje aplikacij. Čarovniki za izdelavo aktivnosti aplikacije ponujajo predloge za izdelavo vnaprej pripravljenih najpogostejših oblik uporabniških vmesnikov aktivnosti. Izdelava uporabniških vmesnikov je mogoča z vlečenjem posameznih komponent na aktivnost. Android Studio omogoča tudi razvoj

aplikacij za pametne ure Android Wear, pametne televizijske sprejemnike Android TV in zaslone v avtomobilih Android Auto. Vgrajeno ima podporo za platformo Google Cloud Platform, ki omogoča integracijo s Firebase Cloud Messaging (rešitev za pošiljanje sporočil med različnimi platformami) in Google App Engine (spletno ogrodje za razvoj in gostovanje spletnih aplikacij). Prek virtualne naprave AVD (t. j. *Android Virtual Device*) je omogočeno poganjanje in razhroščevanje razvitih aplikacij brez uporabe fizične naprave z mobilnim operacijskim sistemom Android. Novejše različice Android Studia pa se lahko pohvalijo tudi z možnostjo takojšnjega zagona aplikacije. To pomeni, da razvojno okolje zazna in razume spremembe v kodi in jih pogosto prikaže brez ponovne gradnje datoteke APK in brez ponovnega zagona aplikacije. [4]

Android Studio podpira programske jezike, ki jih že podpirata razvojni okolji IntelliJ IDEA (na tem temelji Android Studio) in PyCharm, razvojni okolji izdelovalca JetBrains.

Programski jeziki, ki jih brez dodatkov podpira Android Studio:

- Java,
- Groovy,
- C,
- C++,
- XML/XSL,
- Kotlin,
- Python.

Tudi razvojno okolje Android Studio podpira namestitvev najrazličnejših dodatkov s katerimi je mogoče razširiti nabor podprtih funkcionalnosti.

3. Analiza problema in definicija zahtev informacijskega sistema

Za shranjevanje računov in pregled le-teh na enem mestu je bil razvit informacijski sistem. Ta uporabnikom omogoča dodajanje klasičnih papirnatih računov in tistih ustvarjenih znotraj tega informacijskega sistema. V kolikor je vnesen datum izteka garancije, informacijski sistem en mesec pred iztekom garancije uporabniku pošlje obvestilo na e-poštni naslov. Na drugi strani pa trgovcem omogoča izdajanje računov ter dodajanje artiklov v ponudbo. Sistemski administrator ima pravice upravljati z uporabniki in trgovinami ter dodeljevati uporabnike trgovce v trgovine. Pred samim začetkom razvoja tega informacijskega sistema so bile določene zahteve in možnosti, ki jih mora informacijski sistem podpirati. Določen je bil tudi podatkovni model za shranjevanje vseh podatkov v podatkovni bazi. Na podlagi tega je bila ustvarjena relacijska podatkovna baza s pripadajočimi tabelami, atributi in povezavami.

3.1. Diagram primerov uporabe za spletno aplikacijo

Diagram primerov uporabe za spletno aplikacijo (Priloga 1) prikazuje uporabniške zahteve za funkcionalnosti, ki jih mora omogočati spletna aplikacija. Spletni del informacijskega sistema mora podpirati prijavo za štiri vrste uporabnikov: uporabnik, trgovec, trgovec administrator in administrator. Ob prijavi v spletno aplikacijo je vsaka vrsta uporabnika samodejno preusmerjena na prilagojeno domačo stran. Uporabnik je preusmerjen na stran, ki omogoča dodajanje računov in upravljanje obstoječih. Trgovec je preusmerjen na stran, ki omogoča ustvarjanje novih računov in pregled že ustvarjenih. Trgovec administrator je ob prijavi preusmerjen na stran, ki omogoča dodajanje novih izdelkov v ponudbo in upravljanje obstoječih. Administrator je preusmerjen na stran z upravljanjem registriranih uporabnikov in dodanih trgovin. Administratorjeva stran omogoča tudi dodajanje novih uporabnikov in trgovin.

Spletna aplikacija mora neprijavljenim uporabnikom omogočati registracijo, prijavo in ponastavitev pozabljenega gesla. Prijavljenemu uporabniku mora omogočati dodajanje klasičnega računa prek nalaganja slike iz računalnika. Podpirati mora tudi dodajanje računa ustvarjenega znotraj tega informacijskega sistema prek vnesene kode. Uporabniku mora biti na voljo pregled vseh dodanih računov. Za posamezen račun morajo biti na ogled vsi pripadajoči podatki. Omogočeno mora biti urejanje podatkov posameznega računa, a le pri tistih, ki so dodani prek slike. Urejanje podatkov na računih ustvarjenih znotraj tega informacijskega sistema ni dovoljeno. Uporabniki morajo imeti možnost brisanja dodanega računa. Dodatna zmožnost tega informacijskega sistema je prikaz dodatnih gradiv, kot so slike in dokumenti PDF, o posameznem izdelku. Dodatna gradiva so prikazana na posameznem računu, ki je bil ustvarjen znotraj tega informacijskega sistema, uporabnik pa mora imeti tudi možnost pregleda vseh dodatnih gradiv z vseh dodanih računov na posebni strani. Na tej strani mora biti

omogočeno tudi razvrščanje dodatnih gradiv in filtriranje po kategorijah izdelkov. Omogočeno mora biti obveščanje uporabnikov pred iztekom garancije, kar je izvedeno z e-poštnim sporočilom na uporabnikov e-poštni naslov. Vsem uporabnikom, prijavljenim in neprijavljenim, pa mora biti omogočen pregled izdelkov, ki jih ponujajo trgovine znotraj tega informacijskega sistema.

Trgovec mora imeti na voljo možnost za kreiranje novega računa. Za izdajo računa ima omogočen tudi pregled vseh izdelkov iz ponudbe svoje trgovine. Prav tako mora imeti pregled nad vsemi računi, ki jih je izdal v sklopu dotičnega uporabniškega računa. Na voljo mora imeti pregled vseh podatkov o posameznem izdanem računu. Trgovec mora imeti na voljo tudi brisanje posameznega računa.

Le trgovec administrator mora imeti možnost dodajanja novih izdelkov v ponudbo svoje trgovine. Na voljo mora imeti pregled vseh izdelkov v ponudbi svoje trgovine. Zanj mora biti omogočeno urejanje in brisanje posameznega izdelka. Tako kot običajen trgovec mora imeti možnost izdajanja računov tudi trgovec administrator. Pri pregledu izdanih računov je razlika od običajnega trgovca ta, da mora imeti trgovec administrator vpogled v izdane račune od vseh trgovcev v sklopu svoje trgovine. Tudi zanj mora biti omogočen pregled podatkov o kateremkoli računu in brisanje le-tega.

Administrator informacijskega sistema mora imeti pregled nad vsemi registriranimi uporabniki. Informacijski sistem mu mora omogočati dodajanje novih uporabnikov in ponastavitev gesla obstoječim uporabnikom. Administrator mora imeti tudi možnost onemogočiti posamezen uporabniški profil in s tem preprečiti nadaljnjo uporabo dotičnemu uporabniku. Uporabnikom lahko spreminja uporabniške pravice. Trgovcem lahko dodeli trgovino. Imeti mora pregled nad vsemi trgovinami v sistemu in možnost dodajanja novih.

Ne glede na dodeljene uporabniške pravice lahko vsaka skupina uporabnikov informacijski sistem z obstoječim uporabniškim računom uporablja tudi kot običajen uporabnik. S tem so vsakemu uporabniku na voljo vse zmožnosti, ki jih informacijski sistem ponuja običajnemu uporabniku, vključno z dodajanjem in pregledom lastnih računov. V meniju spletne aplikacije je vsaki vrsti uporabnika dosegljiva povezava na uporabniško stran.

3.2. Diagram primerov uporabe za mobilno aplikacijo

Diagram primerov uporabe za mobilno aplikacijo (Priloga 2) prikazuje uporabniške zahteve za funkcionalnosti, ki jih mora omogočati mobilna aplikacija. Mobilna aplikacija je na voljo samo običajnim uporabnikom. Zato mora ponujati vse zmožnosti, ki jih običajnim uporabnikom nudi že spletna aplikacija. Nekatere zmožnosti pa mobilna aplikacija razširja.

Mobilna aplikacija mora omogočati registracijo novega uporabnika, prijavo in ponastavitev gesla na obstoječem uporabniškem profilu. Prijava v mobilno aplikacijo mora biti stalna in se ohraniti tudi, ko uporabnik aplikacijo povsem zapre. Odjava iz mobilne aplikacije mora biti mogoča, ko uporabnik izbere možnost odjave v meniju v sami aplikaciji. Dodajanje računa mora biti omogočeno prek slike ali kode. Za prvo možnost mora biti uporabniku znotraj aplikacije na voljo zagon fotoaparata za zajemanje slike računa ali izbor že obstoječe slike iz galerije na sami napravi. Za drugo možnost pa mora biti uporabniku na voljo vnos kode računa prek tipkovnice ali zagon fotoaparata za branje kode QR v kateri je zapisana koda računa. Mobilna aplikacija mora uporabniku nuditi pregled vseh shranjenih računov. Ob izbiri posameznega računa mora uporabniku prikazati vse podatke o izbranem računu. Pri računih dodanih prek slike in posledično ročno vpisanih podatkih, mora uporabniku nuditi urejanje podatkov. Pri računih dodanih prek kode sestavljene iz črk in števil ali kode QR, torej tistih ustvarjenih znotraj tega informacijskega sistema, urejanje podatkov ne sme biti omogočeno. Ravno tako kot pri spletni aplikaciji. Uporabnik mora imeti možnost brisanja računa. Na računih ustvarjenih znotraj tega informacijskega sistema mora imeti uporabnik pregled vseh dodatnih gradiv o izdelkih, kot so slike in datoteke PDF. V aplikaciji mora biti tudi aktivnost namenjena dodatnim gradivom o izdelkih, ki mora prikazovati dodatna gradiva o vseh izdelkih z vseh računov. Na tej aktivnosti mora biti mogoče razvrščanje dodatnih gradiv po določenih kriterijih in filtriranje njih po kategorijah. Tudi pri računih dodanih prek mobilne aplikacije mora uporabnik na svoj e-poštni naslov prejeti obvestilo en mesec pred iztekom garancije, če je bil datum izteka garancije določen. Tudi mobilna aplikacija mora tako prijavljenim kot tudi neprijavljenim uporabnikom omogočiti pregled vseh izdelkov, ki so na voljo za nakup znotraj tega informacijskega sistema.

3.3. Podatkovni model

Entitetno-relacijski diagram (Priloga 3) prikazuje zgradbo podatkovne baze tega informacijskega sistema. Relacijska podatkovna baza vsebuje devet tabel. Deluje na strežniku Microsoft SQL Server, ki gostuje na oblačni platformi Microsoft Azure. Podatki vseh tabel so na disku zapisani po naraščajočem vrstnem redu primarnega ključa. V podatkovni bazi tega informacijskega sistema se uporabljajo podatkovni tipi, ki so prikazani v spodnji tabeli (Tabela 3.1).

Podatkovni tip	Opis	Privzeta natančnost
<i>bit</i>	Shranjuje vrednosti 0 ali 1. Rezervirana beseda <i>true</i> je pretvorjena v 1, <i>false</i> pa v 0. Vrednost 0 običajno pomeni »ne drži«, 1 pa »drži«.	/
<i>datetime2</i>	Shranjuje vrednosti datuma v kombinaciji s časom temelječim na 24-urni obliki zapisa. Je razširitev podatkovnega tipa <i>datetime</i> s podporo večjemu datumskemu razponu, večjo privzeto natančnostjo in opsijsko natančnostjo določeno s strani uporabnika.	7 decimalnih mest za sekunde
<i>float</i>	Shranjuje približke vrednosti številskih podatkov s plavajočo vejico. Vse vrednosti v podatkovnem razponu namreč ne morejo biti natančno predstavljene. Uporablja se npr. za zapis decimalnih števil.	53 bitov za shranjevanje mantise
<i>int</i>	Shranjuje natančne vrednosti celih števil od -2^{31} do $2^{31} - 1$.	/
<i>nvarchar</i>	Shranjuje podatke v obliki niza znakov s podporo standardu Unicode (podpira šumnike). Primeren je za podatke, kjer se dolžina niza razlikuje.	1, z uporabo funkcije CAST 30
<i>uniqueidentifier</i>	Shranjuje enolične identifikatorje.	/
<i>varchar</i>	Shranjuje podatke v obliki niza znakov in ne podpira standarda Unicode. Primeren je za podatke, kjer se dolžina niza razlikuje.	1, z uporabo funkcij CAST ali CONVERT 30

Tabela 3.1: Uporabljeni podatkovni tipi v podatkovni bazi [7].

Tabela *Shop* je namenjena hranjenju podatkov o trgovinah. Vsebuje attribute *id* (podatkovni tip: *uniqueidentifier*), *name* (podatkovni tip: *nvarchar(50)*), *address* (podatkovni tip: *nvarchar(300)*), *phone* (podatkovni tip: *varchar(15)*), *email* (podatkovni tip: *nvarchar(100)*) in *webpage* (podatkovni tip: *nvarchar(100)*). Atribut *id* hrani enolični identifikator trgovine. Atribut *name* hrani ime trgovine do dolžine 50 znakov. Atribut *address* hrani naslov trgovine do dolžine 300 znakov. Atribut *phone* hrani telefonsko številko trgovine do dolžine 15 znakov. Atribut *email* hrani e-poštni naslov trgovine do dolžine 100 znakov. Atribut *webpage* pa hrani

povezavo do spletne strani trgovine do dolžine 100 znakov. Atributi *phone*, *email* in *webpage* dopuščajo prazne vrednosti v tabeli. Primarni ključ tabele je atribut *id*.

Tabela *UserAccess* je namenjena hranjenju podatkov o vrstah uporabnikov. Vsebuje atributa *id* (podatkovni tip: *uniqueidentifier*) in *userType* (podatkovni tip: *nvarchar(50)*). Atribut *id* hrani enolični identifikator vrste uporabnika. Atribut *userType* pa hrani ime vrste uporabnika do dolžine 50 znakov. Nobeden izmed atributov v tej tabeli ne dopušča praznih vrednosti. Primarni ključ tabele je *id*.

Tabela *User* je namenjena hranjenju podatkov o uporabnikih. Vsebuje attribute *username* (podatkovni tip: *varchar(20)*), *password* (podatkovni tip: *varchar(max)*), *email* (podatkovni tip: *varchar(100)*), *name* (podatkovni tip: *nvarchar(50)*), *surname* (podatkovni tip: *nvarchar(50)*), *access* (podatkovni tip: *uniqueidentifier*), *active* (podatkovni tip: *bit*) in *idShop* (podatkovni tip: *uniqueidentifier*). Atribut *username* hrani uporabniško ime do dolžine 20 znakov. Atribut *password* hrani uporabniško geslo največje mogoče dolžine (geslo je shranjeno kot izvleček kodirne funkcije MD5). Atribut *email* hrani uporabnikov e-poštni naslov do dolžine 100 znakov. Atribut *name* hrani ime uporabnika do dolžine 50 znakov. Atribut *surname* hrani priimek uporabnika do dolžine 50 znakov. Atribut *access* hrani enolični identifikator vrste uporabnika. Atribut *active* hrani vrednost, ali je uporabnik omogočen. Atribut *idShop* pa hrani enolični identifikator trgovine, če je uporabnik morebiti trgovec. Atributa *name* in *surname* dopuščata prazne vrednosti v tabeli. Atribut *active* ima nastavljeno privzeto vrednost 1, kar določa da je uporabnik privzeto omogočen. Primarni ključ tabele je *username*, saj ima eno uporabniško ime lahko le en uporabnik. Atribut *access* je tuji ključ, ki kaže na atribut *id* v tabeli *UserAccess*. Gre za povezavo 1 proti mnogo, saj ima lahko en uporabnik določeno samo eno vrsto uporabniških pravic, neko vrsto uporabniških pravic pa ima lahko več uporabnikov. Atribut *idShop* je tuji ključ, ki kaže na atribut *id* v tabeli *Shop*. Tudi v tem primeru gre za povezavo 1 proti mnogo, saj je en uporabnik lahko trgovec v največ eni trgovini, trgovina pa ima lahko več trgovcev.

Tabela *ProductCategory* je namenjena hranjenju podatkov o kategorijah izdelkov. Vsebuje atributa *id* (podatkovni tip: *uniqueidentifier*) in *category* (podatkovni tip: *nvarchar(50)*). Atribut *id* hrani enolični identifikator kategorije izdelka. Atribut *category* pa hrani ime kategorije izdelka do dolžine 50 znakov. Nobeden izmed atributov v tej tabeli ne dopušča praznih vrednosti. Primarni ključ tabele je *id*.

Tabela *Product* je namenjena hranjenju podatkov o izdelkih. Vsebuje attribute *id* (podatkovni tip: *uniqueidentifier*), *productName* (podatkovni tip: *nvarchar(50)*), *description* (podatkovni tip: *nvarchar(1000)*), *techDetails* (podatkovni tip: *nvarchar(1000)*), *price* (podatkovni tip: *float(53)*), *active* (podatkovni tip: *bit*), *idShop* (podatkovni tip: *uniqueidentifier*) in *idCategory*

(podatkovni tip: *uniqueidentifier*). Atribut *id* hrani enolični identifikator izdelka. Atribut *productName* hrani ime izdelka do dolžine 50 znakov. Atribut *description* hrani opis izdelka do dolžine 1000 znakov. Atribut *techDetails* hrani tehnične značilnosti izdelka do dolžine 1000 znakov. Atribut *price* hrani ceno izdelka v decimalnem številu z natančnostjo do 53 bitov. Atribut *active* hrani vrednost, ali je izdelek omogočen. Atribut *idShop* hrani enolični identifikator trgovine, ki ponuja izdelek. Atribut *idCategory* pa hrani enolični identifikator kategorije izdelka. Atributa *description* in *techDetails* dopuščata prazne vrednosti v tabeli. Atribut *active* ima nastavljeno privzeto vrednost 1, kar določa da je izdelek privzeto omogočen. Primarni ključ tabele je *id*. Atribut *idShop* je tuji ključ, ki kaže na atribut *id* v tabeli *Shop*. Gre za povezavo 1 proti mnogo, saj je en izdelek lahko na voljo le v eni trgovini, ena trgovina pa ima lahko v ponudbi več izdelkov. Atribut *idCategory* je tuji ključ, ki kaže na atribut *id* v tabeli *ProductCategory*. Tudi v tem primeru gre za povezavo 1 proti mnogo, saj en izdelek lahko spada v le eno kategorijo, v eni kategoriji pa je lahko več izdelkov.

Tabela *ItemsReceiptTEMP* je namenjena začasemu hranjenju izdelkov na računu med samim ustvarjanjem računa. Vsebuje attribute *id* (podatkovni tip: *uniqueidentifier*), *seller* (podatkovni tip: *varchar(20)*), *idProduct* (podatkovni tip: *uniqueidentifier*), *quantity* (podatkovni tip: *int*), *productSerialNo* (podatkovni tip: *varchar(50)*) in *dateOfGuarantee* (podatkovni tip: *datetime2(7)*). Atribut *id* hrani enolični identifikator začasnega zapisa izdelka na računu. Atribut *seller* hrani uporabniško ime prodajalca do dolžine 20 znakov. Atribut *idProduct* hrani enolični identifikator izdelka. Atribut *quantity* hrani količino izdelkov. Atribut *productSerialNo* hrani morebitno serijsko številko izdelka do dolžine 50 znakov. Atribut *dateOfGuarantee* pa hrani datum izteka garancije do največje možne natančnosti (7 decimalnih mest pri sekundah). Atributa *productSerialNo* in *dateOfGuarantee* dopuščata prazne vrednosti v tabeli. Primarni ključ tabele je *id*. Atribut *seller* je tuji ključ, ki kaže na atribut *username* v tabeli *User*. Gre za povezavo 1 proti mnogo, saj je en vnos posameznega izdelka na določen račun lahko dodal le en trgovec, več trgovcev pa je lahko naredilo različne vnose na različne račune. Atribut *idProduct* je tuji ključ, ki kaže na atribut *id* v tabeli *Product*. Tudi v tem primeru gre za povezavo 1 proti mnogo, saj je v enem vnosu izdelka na določenem računu lahko prisoten le en izdelek, en izdelek pa je lahko prisoten v več različnih vnosih izdelkov na različnih računih.

Tabela *Materials* je namenjena hranjenju dodatnih gradiv o izdelkih, kot so slike in dokumenti PDF. Vsebuje attribute *id* (podatkovni tip: *uniqueidentifier*), *idProduct* (podatkovni tip: *uniqueidentifier*), *material* (podatkovni tip: *varchar(max)*), *description* (podatkovni tip: *nvarchar(20)*), *picture* (podatkovni tip: *bit*), *active* (podatkovni tip: *bit*), *sortOrder* (podatkovni tip: *int*) in *fileName* (podatkovni tip: *nvarchar(100)*). Atribut *id* hrani enolični identifikator gradiva. Atribut *idProduct* hrani enolični identifikator izdelka, ki mu gradivo pripada. Atribut

material hrani gradivo do največje mogoče dolžine (gradivo je shranjeno v obliki znakovnega niza zakodiranega v Base64). Atribut *description* hrani kratek opis gradiva do dolžine 20 znakov. Atribut *picture* hrani vrednost, ali je gradivo slika. Atribut *active* hrani vrednost, ali je gradivo omogočeno. Atribut *sortOrder* hrani številsko vrednost vrstnega reda gradiva (uporablja se za vrstni red prikaza gradiv izdelka). Atribut *fileName* pa hrani ime datoteke gradiva (uporablja se pri prenosu gradiva na računalnik). Atributa *description* in *sortOrder* dopuščata prazne vrednosti v tabeli. Atribut *active* ima nastavljeno privzeto vrednost 1, kar določa da je gradivo privzeto omogočeno. Primarni ključ tabele je *id*. Atribut *idProduct* je tuji ključ, ki kaže na atribut *id* v tabeli *Product*. Gre za povezavo 1 proti mnogo, saj eno gradivo lahko pripada le enemu izdelku, nek izdelek pa ima lahko več gradiv.

Tabela *Receipt* je namenjena hranjenju podatkov o računih. Vsebuje attribute *id* (podatkovni tip: *uniqueidentifier*), *username* (podatkovni tip: *varchar(20)*), *shop* (podatkovni tip: *nvarchar(50)*), *receiptNumber* (podatkovni tip: *varchar(50)*), *dateOfPurchase* (podatkovni tip: *datetime2(7)*), *dateOfGuarantee* (podatkovni tip: *datetime2(7)*), *receiptPicture* (podatkovni tip: *varchar(max)*), *code* (podatkovni tip: *varchar(20)*), *seller* (podatkovni tip: *varchar(20)*), *receiptPicFileName* (podatkovni tip: *nvarchar(100)*), *activeUser* (podatkovni tip: *bit*) in *activeShop* (podatkovni tip: *bit*). Atribut *id* hrani enolični identifikator računa. Atribut *username* hrani uporabniško ime uporabnika, ki je imetnik računa, do dolžine 20 znakov. Atribut *shop* hrani ime trgovine do dolžine 50 znakov. Atribut *receiptNumber* hrani številko računa do dolžine 50 znakov. Atribut *dateOfPurchase* hrani datum nakupa do največje možne natančnosti (7 decimalnih mest pri sekundah). Atribut *dateOfGuarantee* hrani datum izteka garancije do največje možne natančnosti. Atribut *receiptPicture* hrani morebitno sliko papirnatega računa do največje mogoče dolžine (slika je shranjena v obliki znakovnega niza zakodiranega v Base64). Atribut *code* hrani kodo računa ustvarjenega znotraj tega informacijskega sistema do dolžine 20 znakov. Atribut *seller* hrani uporabniško ime prodajalca, ki je račun ustvaril, do dolžine 20 znakov. Atribut *receiptPicFileName* hrani ime datoteke, če je dodana slika računa (uporablja se pri prenosu slike računa na računalnik), do dolžine 100 znakov. Atribut *activeUser* hrani vrednost, ali je račun na strani uporabnika še omogočen (ni izbrisan). Atribut *activeShop* pa hrani vrednost, ali je račun na strani trgovine še omogočen (ni izbrisan). Atributi *username*, *receiptNumber*, *dateOfGuarantee*, *receiptPicture*, *code*, *seller* in *receiptPicFileName* dopuščajo prazne vrednosti v tabeli. Atributa *activeUser* in *activeShop* imata nastavljeno privzeto vrednost 1, kar določa da je račun privzeto omogočen (ni izbrisan) tako na strani uporabnika, kot tudi na strani trgovine. Primarni ključ tabele je *id*. Atribut *username* je tuji ključ, ki kaže na istoimenski atribut v tabeli *User*. Gre za povezavo 1 proti mnogo, saj en račun lahko pripada le enemu uporabniku, uporabnik pa ima lahko več računov. Atribut *seller* je tuji ključ, ki kaže na atribut *username* v tabeli *User*. Tudi v tem primeru gre za

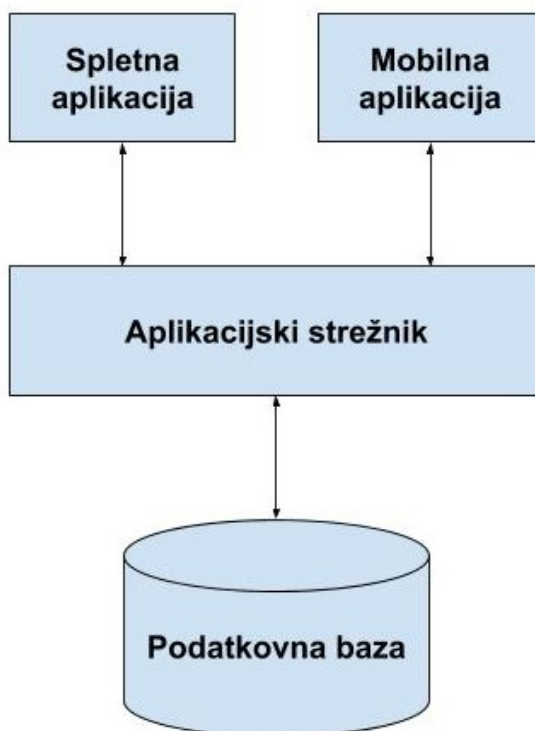
povezavo 1 proti mnogo, saj je en račun lahko izdan s strani enega prodajalca, prodajalec pa lahko izda več računov.

Tabela *ReceiptProduct* služi kot povezovalna tabela med tabelama *Receipt* in *Product*. Namenjena je hranjenju podatkov o izdelkih na računih ustvarjenih znotraj tega informacijskega sistema. Vsebuje attribute *id* (podatkovni tip: *uniqueidentifier*), *idReceipt* (podatkovni tip: *uniqueidentifier*), *idProduct* (podatkovni tip: *uniqueidentifier*), *productSerialNo* (podatkovni tip: *varchar(50)*), *priceUponPurchase* (podatkovni tip: *float(53)*), *dateOfGuarantee* (podatkovni tip: *datetime2(7)*) in *quantity* (podatkovni tip: *int*). Atribut *id* hrani enolični identifikator povezave med računom in izdelkom. Atribut *idReceipt* hrani enolični identifikator računa. Atribut *idProduct* hrani enolični identifikator izdelka. Atribut *productSerialNo* hrani serijsko številko izdelka do dolžine 50 znakov. Atribut *priceUponPurchase* hrani ceno izdelka na dan nakupa v decimalnem številu z natančnostjo do 53 bitov. Atribut *dateOfGuarantee* hrani datum izteka garancije do največje možne natančnosti (7 decimalnih mest pri sekundah). Atribut *quantity* pa hrani količino dotičnih izdelkov na računu. Atributa *productSerialNo* in *dateOfGuarantee* dopuščata prazne vrednosti v tabeli. Atribut *quantity* ima nastavljeno privzeto vrednost 1, kar določa minimalno količino izdelka, ki je dodan na račun. Primarni ključ tabele je *id*. Atribut *idReceipt* je tuji ključ, ki kaže na atribut *id* v tabeli *Receipt*. Gre za povezavo 1 proti mnogo, saj je neka kombinacija računa in izdelka lahko prisotna le na enem računu, na nekem računu pa je lahko več kombinacij. Atribut *idProduct* je tuji ključ, ki kaže na atribut *id* v tabeli *Product*. Tudi v tem primeru gre za povezavo 1 proti mnogo, saj je v eni kombinaciji računa in izdelka lahko prisoten le en izdelek, nek izdelek pa je lahko prisoten v mnogo kombinacijah.

4. Načrtovanje informacijskega sistema

Razvit informacijski sistem je zgrajen iz več elementov, ki so nujni za ustrezno delovanje in izpolnjevanje vseh zahtev. Sestavljajo ga podatkovna baza, spletna aplikacija, dva spletna servisa, konzolna aplikacija in mobilna aplikacija. Slednja za samo delovanje ni nujno potrebna, vendar služi kot razširitev zmožnosti. Uporabnikom omogoča lažje in hitrejše dodajanje računov od koderkoli. Za dostop do informacijskega sistema z računalnika prek spletne aplikacije ali s pametnega telefona z mobilno aplikacijo je potrebna internetna povezava.

4.1. Arhitektura informacijskega sistema



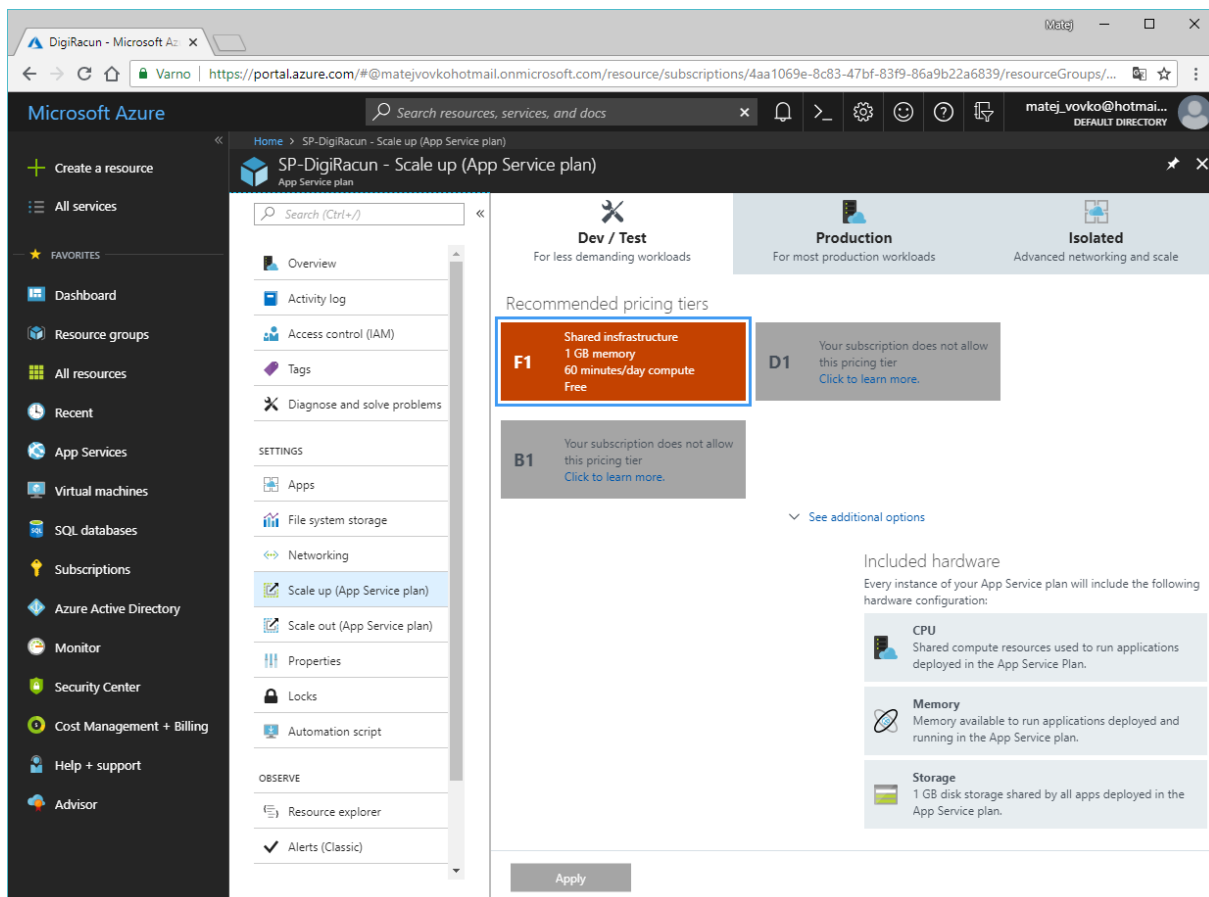
Slika 4.1: Shema arhitekture informacijskega sistema.

Arhitektura razvitega informacijskega sistema (Slika 4.1) je sestavljena iz treh nivojev. Na vrhu se nahajata spletna in mobilna aplikacija. Uporabniki imajo dostop do zgornjega nivoja informacijskega sistema preko spletnega brskalnika v primeru spletne aplikacije ali pametnega telefona z operacijskim sistemom Android v primeru mobilne aplikacije. Podporo delovanju aplikacij zagotavlja aplikacijski strežnik, ki deluje znotraj storitvene platforme Microsoft Azure. Na aplikacijskem strežniku delujeta tudi oba spletna servisa. Prvi skrbi za komunikacijo med mobilno aplikacijo in podatkovno bazo, drugi pa za preverjanje izteka garancije in pošiljanje e-poštnih obvestil. Konzolna aplikacija, ki skrbi za proženje spletnega servisa za pošiljanje e-poštnih obvestil pred iztekom garancije, ravno tako deluje na aplikacijskem strežniku. Vsi podatki informacijskega sistema so shranjeni v podatkovni bazi. Slednja deluje

na strežniku Microsoft SQL Server znotraj platforme Azure. Spletna aplikacija in spletna servisa, ki delujejo na aplikacijskem strežniku, imajo izvedeno povezavo s podatkovno bazo in s tem omogočen dostop do podatkov shranjenih znotraj tega informacijskega sistema.

4.2. Elementi informacijskega sistema

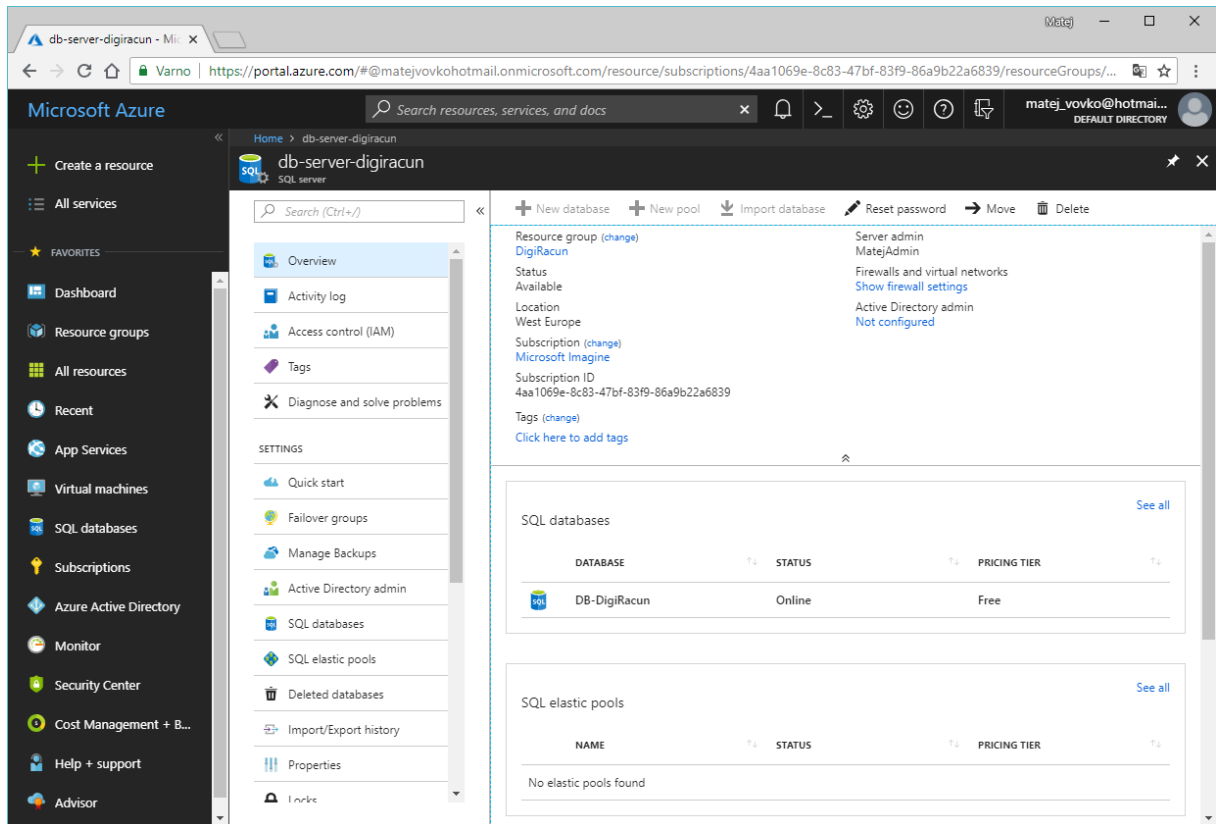
Podatkovna baza, spletna aplikacija in oba spletna servisa delujejo na platformi Microsoft Azure. Zato je bila znotraj te platforme ustvarjena skupina virov, ki so na voljo za potrebe tega informacijskega sistema. Izbran je brezplačni plan (Slika 4.2), ki je prek naročnine Microsoft Imagine na voljo za študijske namene. Ta plan ponuja deljeno infrastrukturo za poganjanje aplikacij, 1 GB trdega diska za shranjevanje in računsko moč do 60 minut na dan. Izbrana lokacija infrastrukture je Zahodna Evropa.



Slika 4.2: Izbrani brezplačni plan na Microsoft Azure.

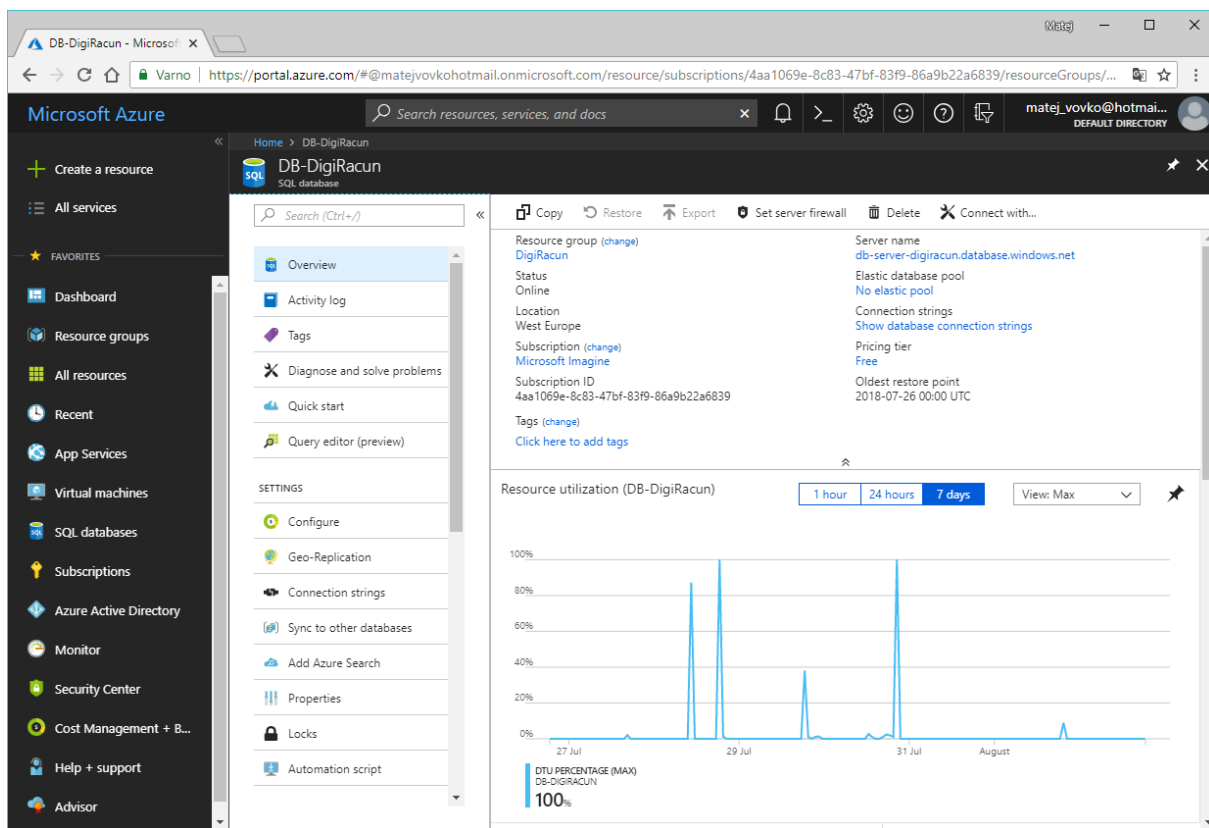
Za delovanje podatkovne baze je bil znotraj platforme Azure ustvarjen strežnik Microsoft SQL Server (Slika 4.3). Tudi lokacija tega strežnika je v Zahodni Evropi. Strežnik ima vklopljeno napredno zaščito pred grožnjami in požarni zid. Za boljše delovanje podatkovne baze so vklopljene tudi samodejne optimizacijske možnosti. Te zajemajo spremljanje

zmogljivosti in uporabo priporočenega plana na podlagi trajanja izvajanja poizvedb ter samodejno kreiranje in brisanje indeksov. [5]



Slika 4.3: Pregled osnovnih podatkov strežnika MS SQL Server na Microsoft Azure.

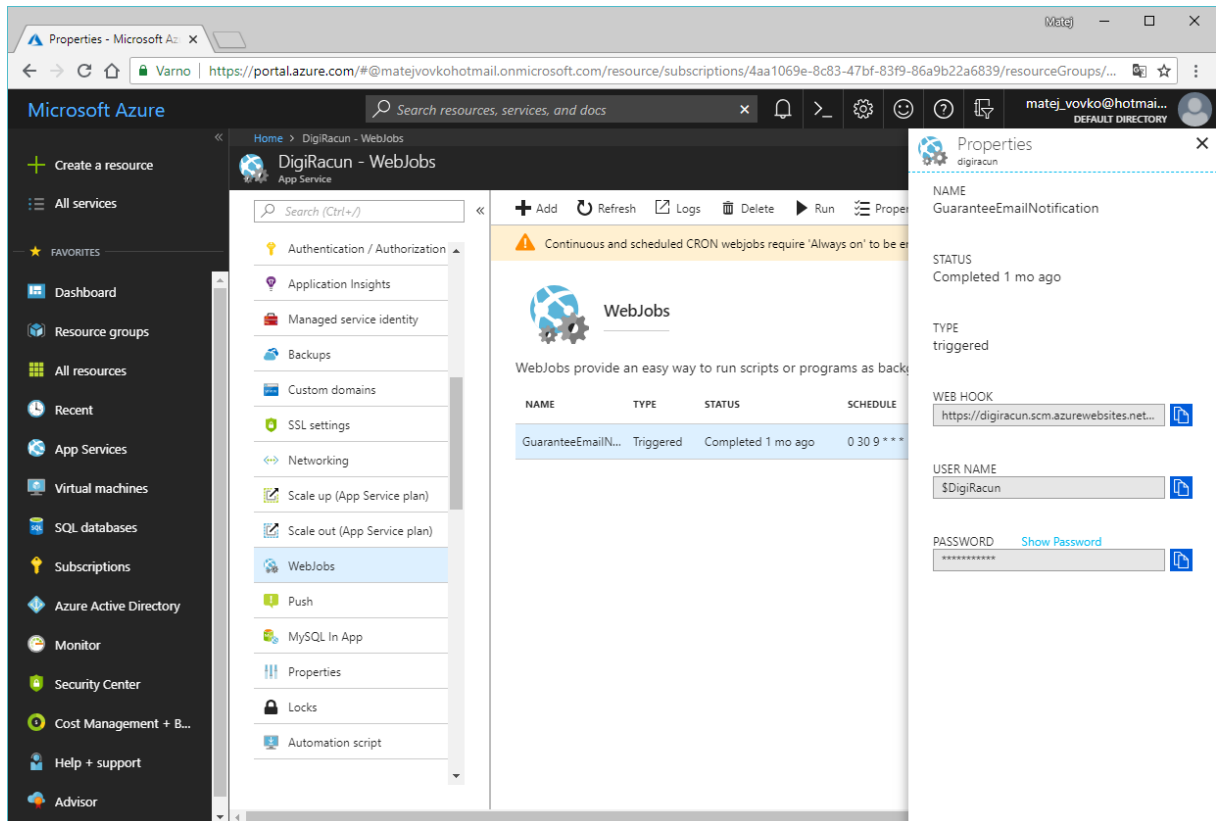
Za shranjevanje podatkov se uporablja relacijska podatkovna baza Microsoft Azure SQL Database (Slika 4.4). Glede na izbrani plan ponuja shrambo za 32 MB podatkov in zmogljivost 5 DTU. DTU (t. j. *Database Transaction Unit*) določa nivo virov znotraj platforme Azure, ki so na voljo določeni podatkovni bazi. Število virov, ki so na voljo, je izračunano kot število transakcijskih enot podatkovne baze (t. i. DTU) in je skupek meritev računskih, shranjevalnih in vhodno/izhodnih virov. Kadar pride do preobremenitve podatkovne baze in je s tem določeno število DTU preseženo, se to odraža pri nižjih zmogljivostih in večjih zakasnitvah. Ostale podatkovne baze znotraj platforme Azure na zmogljivosti neke podatkovne baze ne vplivajo. [6] Znotraj te podatkovne baze so podatki zaščiteni s transparentnim kodiranjem podatkov.



Slika 4.4: Pregled osnovnih podatkov podatkovne baze na Microsoft Azure.

Spletna aplikacija uporabnikom omogoča dostop do informacijskega sistema prek spletnega brskalnika. S spletno stranjo je zagotovljen dostop z vseh naprav, ki ne poganjajo mobilnega operacijskega sistema Android. Za slednje je namreč razvita mobilna aplikacija. Spletna aplikacija je dostopna le prek varne povezave HTTPS. Protokol HTTPS uporablja šifriranje prometa in ga s tem ščiti pred vmesnimi opazovalci. Znotraj nastavitve varne povezave SSL je nastavljena tudi minimalna zahtevana različica kodirnega protokola TLS in sicer TLS 1.2. Gre za najvišjo različico, ki jo dopušča Azure. Znotraj spletne aplikacije delujeta tudi oba spletna servisa WCF (t. j. *Windows Communication Foundation*). Prvi služi komunikaciji z mobilno aplikacijo. Drugi pa je namenjen preverjanju izteka garancije in pošiljanju e-poštnih obvestil o izteku garancije. Pošiljanje zahtevkov HTTPS temu spletnemu servisu je izvedeno prek nastavitve WebJobs (Slika 4.5) v spletni aplikaciji na platformi Azure. Šele, ko se pošlje zahtevek spletnemu servisu, ta izvede ustrezno metodo za pošiljanje e-poštnih obvestil. Pošiljanje zahtevkov je izvedeno samodejno. Nastavitve WebJobs namreč omogoča poganjanje skript v ozadju, ki so prožene prek urnika ali ročno. Skripta za pošiljanje zahtevkov spletnemu servisu je napisana kot konzolna aplikacija v programskem jeziku C# in se sproži vsak dan ob 9.30. To pomeni, da se e-poštna obvestila o izteku garancije uporabnikom pošljejo ob 9.30. Obvestilo o izteku garancije pa se pošlje 1 mesec pred iztekom. Delovanje nastavitve WebJobs prek urnika zahteva, da je spletna aplikacija stalno vklopljena. Azure namreč dalj časa

nedejavne aplikacije ugaša in jih ponovno zažene, ko pride do zahteve po njihovem delovanju. Žal pa izbrani brezplačni plan ne omogoča, da je spletna aplikacija ves čas vklopljena, torej tudi takrat, ko je nihče ne uporablja. Zato pošiljanje e-poštnih obvestil o izteku garancije prek nastavitve urnika s to konfiguracijo ni zanesljivo, lahko pa služi za demonstracijo prek ročnega proženja skripte.



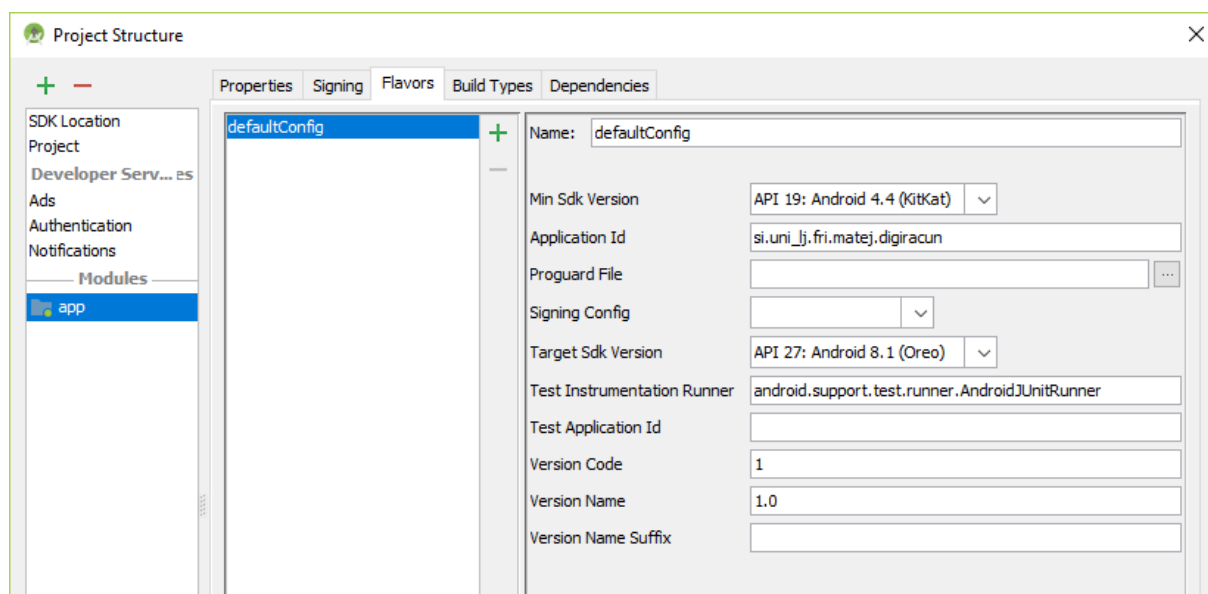
Slika 4.5: Pregled nastavitve WebJobs v spletni aplikaciji na Microsoft Azure.

Zadnji modul informacijskega sistema predstavlja mobilna aplikacija, napisana v programskem jeziku Java. Le-ta za samo delovanje informacijskega sistema ni nujno potrebna, uporabnikom pa omogoča udobnejše upravljanje z računi. S stalno prijavljenostjo uporabnika in s podporo sprotnemu zajemanju slike računa oz. branju kode QR računa ustvarjenega znotraj tega informacijskega sistema omogoča hitrejšo dodajanje računov. Aplikacija je razvita za pametne telefone, ki poganjajo mobilni operacijski sistem Android. Za poganjanje aplikacije je zahtevana različica Androida vsaj 4.4 (programski vmesnik API različice 19) s kodnim imenom KitKat (Slika 4.6). S tem je zajeta velika večina uporabnikov operacijskega sistema Android, in sicer 95,9 % (Tabela 2.1). Vse aktivnosti aplikacije uporabljajo le pokončni oz. portretni način postavitve zaslona. Za branje kode QR se uporablja odprtokodna knjižnica ZXing.

Za pravilno delovanje vseh funkcionalnosti, ki jih omogoča mobilna aplikacija, so zahtevane naslednje pravice:

- dostop do interneta,
- bralni dostop do zunanega pomnilnika,
- pisalni dostop do zunanega pomnilnika,
- dostop do fotoaparata.

Dostop do interneta je potreben za komunikacijo s spletnim servisom, saj so vsi podatki shranjeni v oddaljeni podatkovni bazi. Trajna hramba podatkov namreč ni izvedena lokalno na napravi. Bralni dostop do zunanega pomnilnika je potreben zaradi dostopa do že ustvarjenih slik, saj lahko uporabnik pri dodajanju novega računa izbere že obstoječo sliko iz galerije naprave. Pisalni dostop do zunanega pomnilnika pa je potreben za ogled dodanih gradiv o izdelkih in ogled slike papirnatega računa. Te datoteke se namreč začasno shranijo v zunanji pomnilnik naprave. V aplikaciji je poskrbljeno tudi za samodejno brisanje začasnih datotek in s tem posledično čiščenje začasnega pomnilnika. Dostop do fotoaparata je potreben za zajemanje slik pri dodajanju klasičnih papirnatih računov in pri branju kode QR ob dodajanju računov ustvarjenih znotraj tega informacijskega sistema.

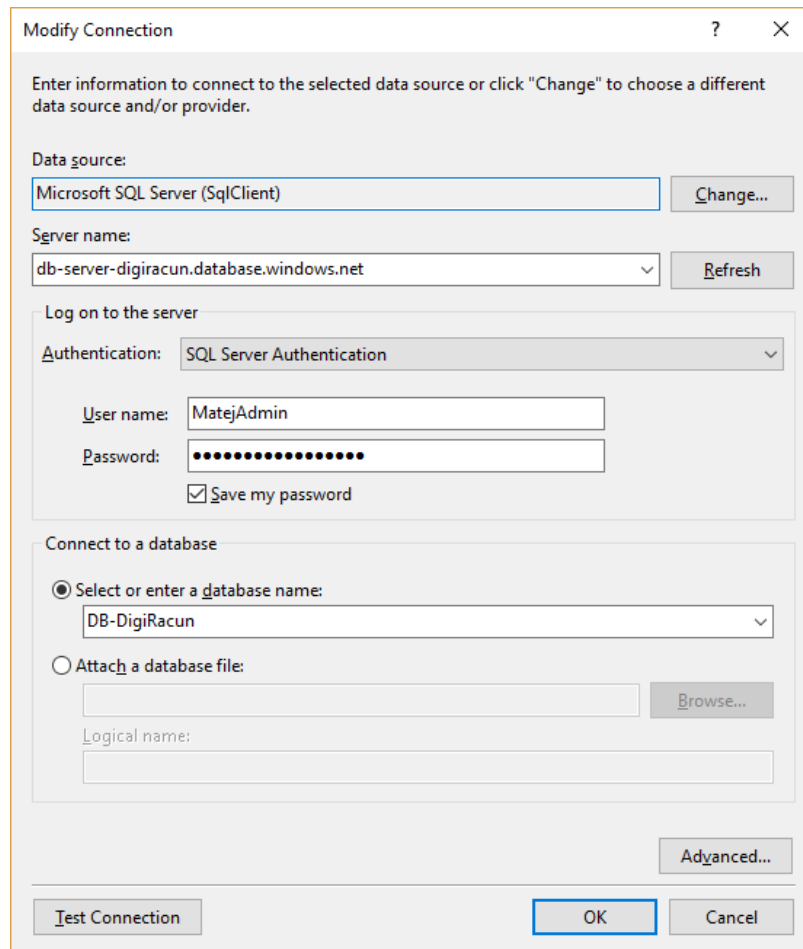


Slika 4.6: Pregled konfiguracije mobilne aplikacije v Android Studiu.

4.3. Povezava: spletna aplikacija in podatkovna baza

Spletna aplikacija znotraj katerih delujeta tudi spletna servisa ima direktni dostop do podatkovne baze. Mobilna aplikacija s podatkovno bazo komunicira posredno prek spletnega

servisa. Povezava do podatkovne baze je v spletni aplikaciji definirana znotraj datoteke Web.config pod oznako connectionStrings. V tej datoteki oblike XML so namreč definirane nastavitve delovanja in obnašanja spletne aplikacije in spletnih servisov.



Slika 4.7: Nastavitve povezave do podatkovne baze v spletni aplikaciji.

Razvojno okolje Microsoft Visual Studio omogoča ustvarjanje povezave do podatkovne baze na strežniku Microsoft SQL Server tudi prek uporabniškega vmesnika (Slika 4.7). Omenjeni strežnik je za ta informacijski sistem postavljen na platformi Microsoft Azure.

Povezavo do podatkovne baze določajo naslednji parametri:

- ime povezave,
- ime strežnika oz. povezava do strežnika,
- ime podatkovne baze na strežniku (katalog),
- uporabniško ime administratorja strežnika,
- geslo administratorja strežnika,

- opcijsko dovoljeno izvajanje več poizvedb hkrati prek ene povezave, kar je uporabljeno v tem primeru (t. i. MARS oz. *Multiple Active Result Sets*),
- ime ponudnika podatkovne baze.

T. i. MARS se uporablja na strani za ustvarjanje novega računa *NewReceipt.aspx*. Ko prodajalec dodaja izdelke na račun, se ti podatki shranjujejo v začasno tabelo *ItemsReceiptTEMP*, ki se uporablja le med ustvarjanjem računa za trenutni prikaz izdelkov na računu. Ko pa prodajalec zaključi račun in ga s tem dokončno ustvari se ti podatki iz začasne tabele berejo zaporedno, istočasno pa se vsak prebran zapis shrani v tabelo *ReceiptProduct*, ki trajno hrani vse ustvarjene povezave med računi in pripadajočimi izdelki. Zaradi tega je uporaba načina MARS nujna, saj se med izvajanjem ene poizvedbe (branje iz začasne tabele) sproti izvaja dodajanje vnosov v drugo tabelo in se s tem istočasno izvajata do dve operaciji nad podatki v tabelah. [12]

4.4. Komunikacija: spletna aplikacija in podatkovna baza

Spletna aplikacija s podatkovno bazo komunicira na dva načina. Prvi je z uporabo gradnikov *SqlDataSource* na spletnih straneh ustvarjenih v ASP.NET, drugi pa iz programske kode C#, ki se izvaja v ozadju spletnih strani.

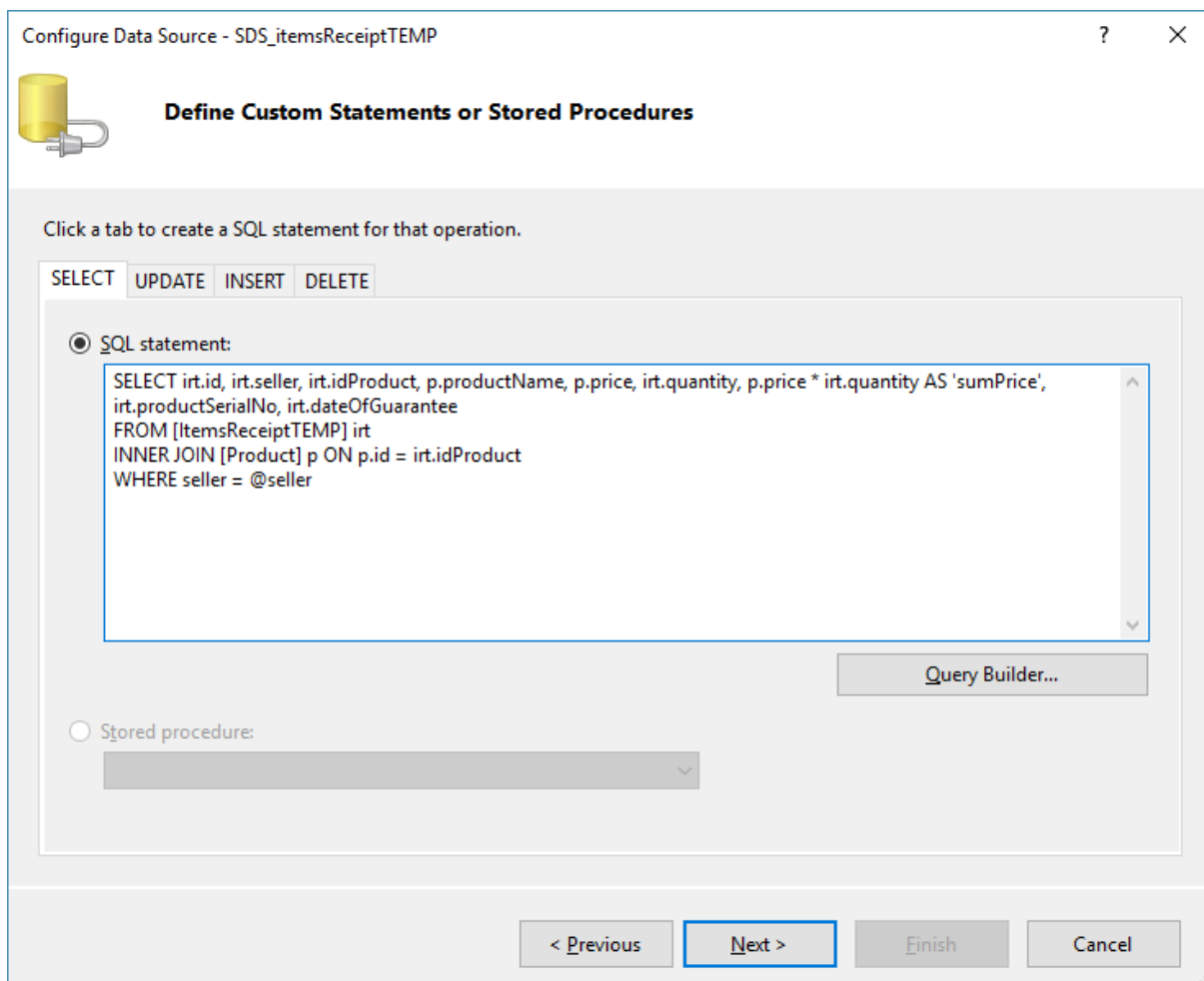
Gradniki *SqlDataSource* (Slika 4.8) se uporabljajo za prikazovanje in spreminjanje podatkov iz podatkovne baze znotraj drugih gradnikov. V tem informacijskem sistemu se uporabljajo za prikazovanje podatkov v tabelah *GridView* in spustnih seznamih *DropDownList*. Pri nekaterih tabelah, kjer je dovoljeno urejanje podatkov, pa služijo tudi za spreminjanje le-teh v podatkovni bazi. Gradnik *SqlDataSource* med nalaganjem strani samodejno odpre povezavo do podatkovne baze, izvede stavek SQL ali shranjeno proceduro, vrne pridobljene podatke in zapre povezavo.

SqlDataSource podpira štiri različne ukaze:

- ukaz za pridobivanje podatkov (stavek *SELECT*),
- ukaz za spreminjanje podatkov (stavek *UPDATE*),
- ukaz za brisanje podatkov (stavek *DELETE*),
- ukaz za dodajanje podatkov (stavek *INSERT*).

Za vsak ukaz se določi stavek SQL ali shranjena procedura. V teh gradnikih znotraj tega informacijskega sistema se najbolj pogosto uporablja ukaz za pridobivanje podatkov. V tabelah *GridView* na spletnih straneh, ki omogočajo urejanje podatkov, pa se uporabljata tudi ukaza za spreminjanje in brisanje podatkov. Vsakega izmed ukazov je mogoče tudi ročno poklicati, kadar so želene spremembe nad prikazanimi podatki. Nekateri gradniki, kot npr. tabela *GridView*, pa

ukaze za upravljanje podatkov kličejo samodejno glede na uporabnikove zahteve. `SqlDataSource` podpira stavke s parametri, ki se med izvajanjem zamenjajo s podanimi vrednostmi. Parametri pred imenom vsebujejo znak afna (@). Vrednosti parametrov so lahko podane na različne načine: piškotek, seja, poizvedba SQL, gradnik ... V tem informacijskem sistemu so vrednosti parametrov običajno podane znotraj spremenljivke seje. [18]



Slika 4.8: Definicija ukaza za pridobivanje podatkov v gradniku `SqlDataSource`.

Dostop do podatkovne baze je izveden tudi iz kode napisane v programskem jeziku C# (Slika 4.9), ki deluje v ozadju spletnih strani. Povezava do relacijske podatkovne baze je ustvarjena kot objekt razreda `SqlConnection`. Pri ustvarjanju objekta je konstruktorju podano ime povezave definirane v datoteki `Web.config`. Objekt razreda `SqlCommand` predstavlja stavek SQL ali shranjeno proceduro. Pri ustvarjanju objekta tega razreda je poleg niza, ki predstavlja stavek SQL ali shranjeno proceduro, podan še objekt razreda `SqlConnection` s povezavo do podatkovne baze. Objekt razreda `SqlCommand` omogoča dodajanje parametrov. Tudi v tem primeru se mora pred imenom parametra nahajati znak afna (@). Vrednosti parametrov se določi s klicem metode `AddWithValue()` na zbirki parametrov `Parameters`, ki privzeto obstaja

na ustvarjenem objektu. Omenjeni metodi je najprej podano ime parametra in nato še njegova vrednost. S klicem ustreznih metod nad objektom se izvede zapisan stavek SQL ali shranjena procedura v podani podatkovni bazi. [14]

V kodi tega informacijskega sistema se za izvajanje stavkov SQL uporabljajo naslednje metode nad objekti razreda SqlCommand:

- ExecuteNonQuery(),
- ExecuteReader(),
- ExecuteReaderAsync(),
- ExecuteScalar().

Metoda ExecuteNonQuery() ne vrne vrstic rezultatov iz podatkovne baze, temveč vrne število vrstic na katere je izvedena operacija vplivala. Uporablja se za izvajanje stavkov SQL za spreminjanje podatkov (stavek *UPDATE*), brisanje podatkov (stavek *DELETE*) in dodajanje podatkov (stavek *INSERT*). Lahko se uporablja tudi za izvajanje drugih stavkov (npr. spreminjanje tabel v podatkovni bazi). Pri izvajanju takih stavkov ali v primeru razveljavitve transakcije, vrne vrednost - 1. [15]

Metoda ExecuteReader() vrne objekt razreda SqlDataReader. S klicem metode Read() na vrnjenem objektu je mogoče pridobiti podatke iz podatkovne baze po posameznih vrsticah. Vrednosti posameznih stolpcev znotraj vrstice pa je mogoče pridobiti s klici različic metode katere ime se prične z besedo »Get« in nadaljuje s podatkovnim tipom kakršnega je podatek na želenem mestu. Tem metodam je kot argument treba podati mesto na katerem se nahaja želeni podatek v vrstici. [16]

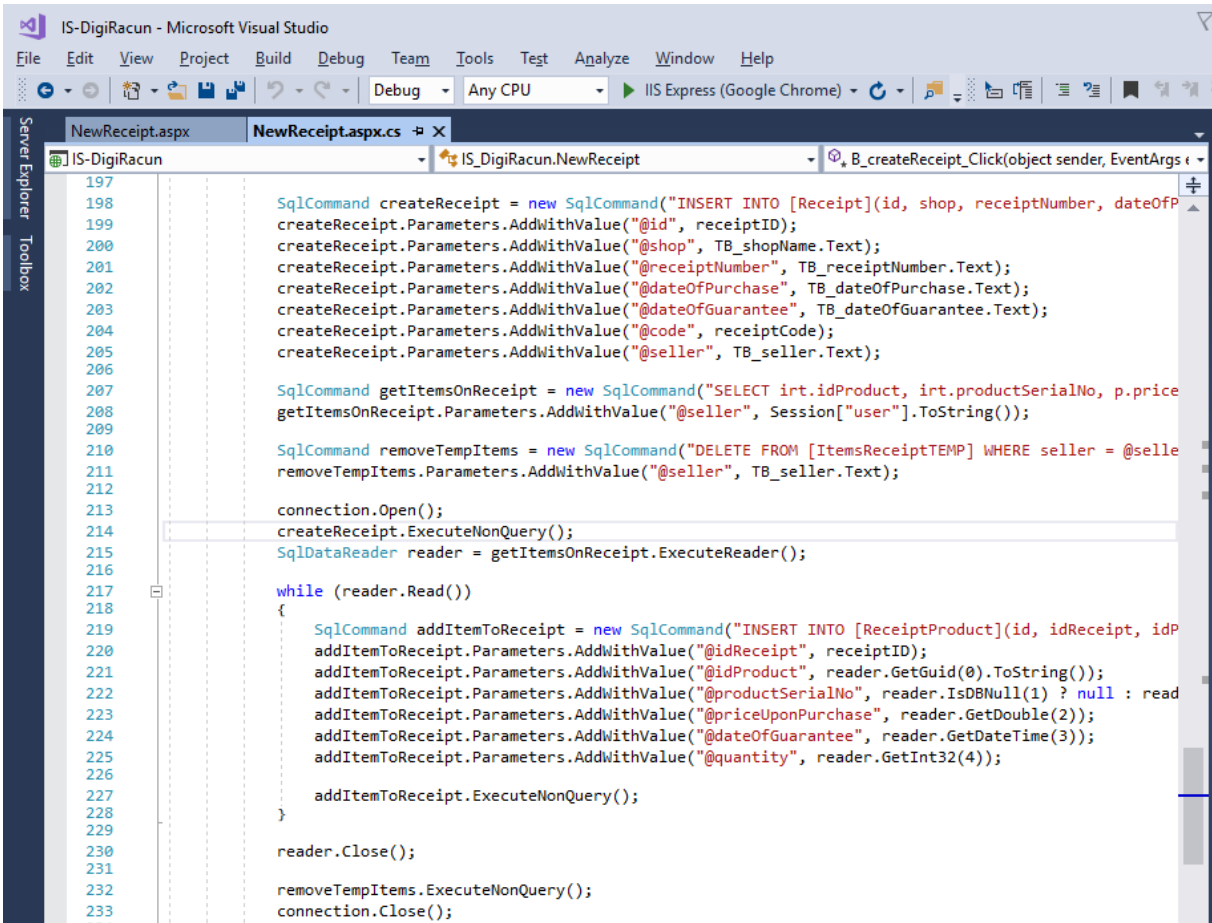
Primeri metod, ki se začnejo z besedo »Get« in so uporabljeni v kodi tega informacijskega sistema, so sledeči:

- GetBoolean() za pridobivanje podatkov tipa *bit*,
- GetDateTime() za pridobivanje podatkov tipa *datetime2*,
- GetDouble() za pridobivanje podatkov tipa *float*,
- GetInt32() za pridobivanje podatkov tipa *int* (32-bitov za zapis celega predznačenega števila),
- GetString() za pridobivanje podatkov tipov *nvarchar* in *varchar*,
- GetGuid() za pridobivanje podatkov tipa *uniqueidentifier*.

Metoda ExecuteReaderAsync() je asinhrona različica metode ExecuteReader(). Asinhrono programiranje omogoča izvajanje druge programske kode medtem, ko se čaka na dokončanje

dalj časa trajajočih operacij. Ta metoda se v tem informacijskem sistemu uporablja za pridobivanje podatkov iz podatkovne baze v spletnem servisu, ki skrbi za komunikacijo z mobilno aplikacijo.

Metoda `ExecuteScalar()` izvede poizvedbo SQL in vrne vrednost prvega stolpca prve vrstice. Ostale vrstice se ne upoštevajo. Uporabna je pri pridobivanju podatkov, kjer se pričakuje le ena vrednost. V takih primerih je za pridobivanje podatka potrebno napisati manj programske kode kot z uporabo objekta `SqlDataReader` in metode `ExecuteReader()`. V kodi tega informacijskega sistema se uporablja npr. pri registraciji uporabnikov za preverjanje, ali je uporabniško ime že zasedeno in pri preverjanju uspešnosti prijave glede na podano uporabniško ime in geslo. [17]



```
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234

SqlCommand createReceipt = new SqlCommand("INSERT INTO [Receipt](id, shop, receiptNumber, dateOfP
createReceipt.Parameters.AddWithValue("@id", receiptID);
createReceipt.Parameters.AddWithValue("@shop", TB_shopName.Text);
createReceipt.Parameters.AddWithValue("@receiptNumber", TB_receiptNumber.Text);
createReceipt.Parameters.AddWithValue("@dateOfPurchase", TB_dateOfPurchase.Text);
createReceipt.Parameters.AddWithValue("@dateOfGuarantee", TB_dateOfGuarantee.Text);
createReceipt.Parameters.AddWithValue("@code", receiptCode);
createReceipt.Parameters.AddWithValue("@seller", TB_seller.Text);

SqlCommand getItemsOnReceipt = new SqlCommand("SELECT irt.idProduct, irt.productSerialNo, p.price
getItemsOnReceipt.Parameters.AddWithValue("@seller", Session["user"].ToString());

SqlCommand removeTempItems = new SqlCommand("DELETE FROM [ItemsReceiptTEMP] WHERE seller = @sell
removeTempItems.Parameters.AddWithValue("@seller", TB_seller.Text);

connection.Open();
createReceipt.ExecuteNonQuery();
SqlDataReader reader = getItemsOnReceipt.ExecuteReader();

while (reader.Read())
{
    SqlCommand addItemToReceipt = new SqlCommand("INSERT INTO [ReceiptProduct](id, idReceipt, idP
addItemToReceipt.Parameters.AddWithValue("@idReceipt", receiptID);
addItemToReceipt.Parameters.AddWithValue("@idProduct", reader.GetGuid(0).ToString());
addItemToReceipt.Parameters.AddWithValue("@productSerialNo", reader.IsDBNull(1) ? null : read
addItemToReceipt.Parameters.AddWithValue("@priceUponPurchase", reader.GetDouble(2));
addItemToReceipt.Parameters.AddWithValue("@dateOfGuarantee", reader.GetDateTime(3));
addItemToReceipt.Parameters.AddWithValue("@quantity", reader.GetInt32(4));

    addItemToReceipt.ExecuteNonQuery();
}

reader.Close();

removeTempItems.ExecuteNonQuery();
connection.Close();
```

Slika 4.9: Ukazi za upravljanje s podatki podatkovne baze iz kode C# v ozadju spletne strani.

4.5. Komunikacija: mobilna aplikacija in spletni servis

Mobilna aplikacija do podatkov informacijskega sistema dostopa prek spletnega servisa, ki deluje znotraj spletne aplikacije. Spletni servis deluje po principu REST. Metode in tipi objektov JSON so določeni na vmesniku spletnega servisa (t. i. *interface*). Spletni servis ima

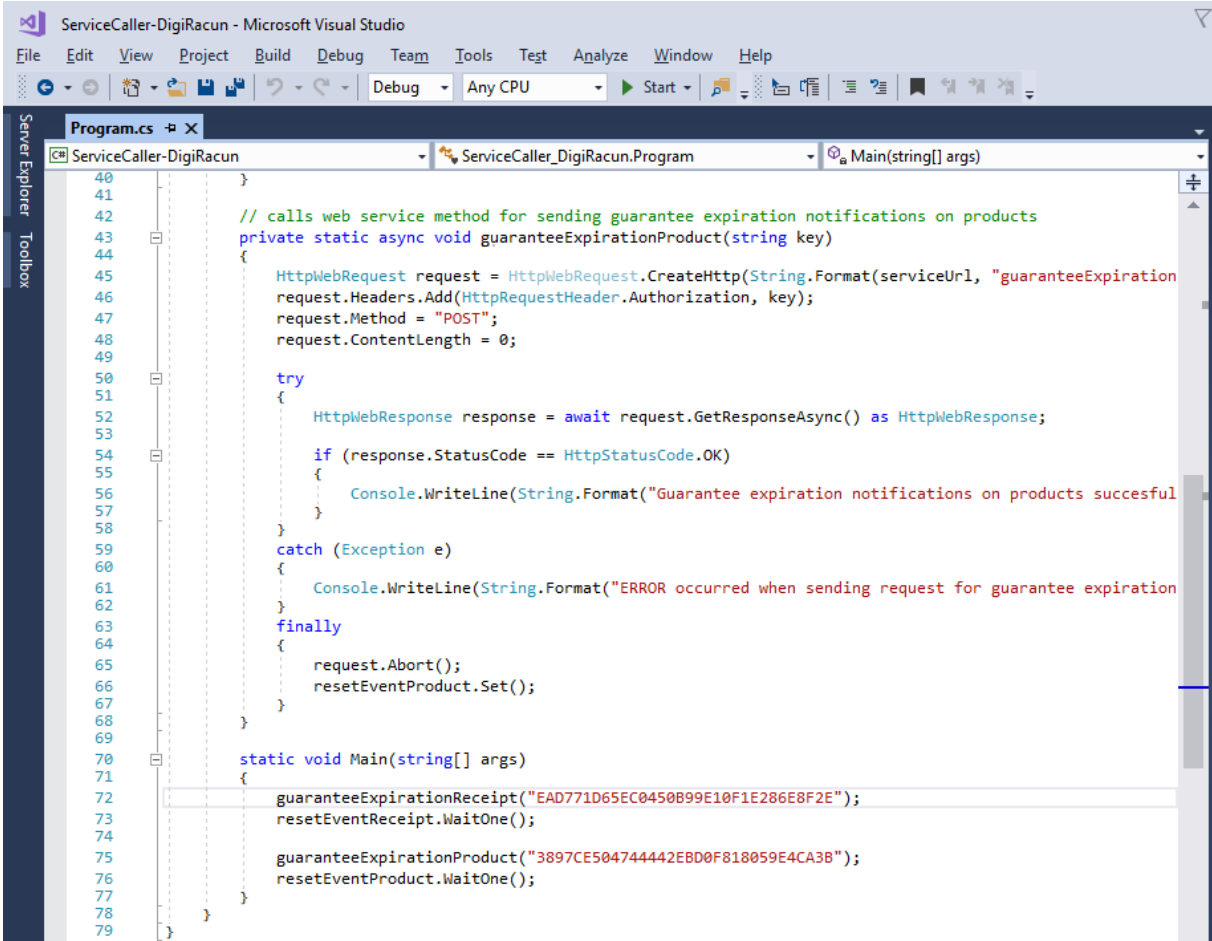
dve vrsti metod HTTPS tipov GET in POST. Prve so namenjene pridobivanju podatkov iz informacijskega sistema (npr. pridobivanje vseh računov nekega uporabnika), druge pa pošiljanju podatkov v informacijski sistem (npr. dodajanje novega računa ali spreminjanje uporabniškega gesla). Metode GET podatke sprejemajo v obliki parametrov v naslovu URL. V takih primerih metode sprejemajo enolične identifikatorje objektov, da vrnejo ostale podatke o želenih objektih (npr. pridobivanje podatkov o računu prek pripadajočega enoličnega identifikatorja). Metode POST običajno sprejemajo večjo količino podatkov. Le-ti so del telesa (t. i. *body*) zahtevka HTTPS. Podatki v telesu so lahko v obliki objektov JSON (npr. pri spreminjanju uporabniškega profila), lahko pa so zgolj v obliki neformatiranega znakovnega niza (npr. pri spreminjanju gesla). Odzivi spletnega servisa so v formatu objektov JSON, razen v primeru prijave, kjer spletni servis vrne vrednost »true« (uspešna prijava) ali »false« (neuspešna prijava). Kadar spletni servis ne vrne nobenih podatkov, se kot rezultat uspešnosti klika metode na spletnem servisu upošteva koda statusa odziva. Objekt JSON določenega tipa vsebuje attribute v katerih so shranjeni podatki. Niso pa vedno napolnjeni vsi atributi objekta. Kateri atributi objekta so napolnjeni, je odvisno od tega, kateri podatki so potrebni glede na klicano posamezno metodo.

Spletni servis ima določene naslednje tipe objektov JSON:

- *User* (podatki o uporabniku),
- *Receipt* (podatki o računu),
- *Product* (podatki o izdelku na računu),
- *ProductCategory* (podatki o kategoriji izdelka),
- *Material* (podatki o dodatnem gradivu izdelka),
- *ProductShop* (podatki o izdelku, ki je na voljo za nakup).

V mobilni aplikaciji za komunikacijo s spletnim servisom skrbi razred *ServiceCaller* (Slika 4.10), ki je napisan v programskem jeziku Java. Razred je ustvarjen kot asinhroni, kar pomeni, da je med čakanjem odziva spletnega servisa uporabniški vmesnik aplikacije normalno odziven. Razred ima ustvarjenih več konstruktorjev, s tem pa je povečana njegova prilagodljivost glede na mesto klika metod za izvedbo in mesto prikaza podatkov v aplikaciji. Glede na izbran konstruktor, so pri ustvarjanju objekta podani argumenti aktivnost aplikacije, pogled aplikacije, ime metode na spletnem servisu, uporabniško ime in geslo. Razred ima definirano povezavo do spletnega servisa. Omogočeno je le spreminjanje zadnjega dela povezave, kjer je navedena metoda spletnega servisa. Ustvarjati in pošiljati zna zahtevke metodam GET in POST na spletnem servisu. Nad ustvarjenim objektom tega razreda se za komunikacijo s spletnim servisom pokliče metodo *execute()*. Tej se kot argumente poda tip metode GET ali POST, vrednost »true«, če je za izvedbo metode na spletnem servisu zahtevana prijava z uporabniškim

metode, ima konzolna aplikacija ustvarjeni še dve asinhroni metodi. Prva kliče metodo spletnega servisa za pošiljanje obvestil o izteku garancije, če je datum izteka garancije določen na računu. Ta način se uporablja pri klasičnih papirnatih računih, ki so dodani prek slike. Druga metoda pa kliče metodo spletnega servisa za pošiljanje obvestil o izteku garancije, če je datum izteka garancije določen na izdelku. Ta način se uporablja pri računih ustvarjenih znotraj tega informacijskega sistema. Obe metodi ustvarjata zahteve HTTPS tipa POST in jih pošiljata spletnemu servisu. Vsaka pa ima kot argument podan ključ, ki je spletnemu servisu poslan v glavi zahtevka. Spletni servis izvede metodo za pošiljanje e-poštnih obvestil le, če prejme ustrezen ključ. Nekdo, ki ključev ne pozna, potemtakem ne more sprožiti pošiljanja e-poštnih obvestil. Ko je klicana ena izmed metod tega spletnega servisa, ta poišče račune ali izdelke, ki jim čez natanko 1 mesec poteče garancija in uporabnikom pošlje e-poštna obvestila. Konzolna aplikacija ne pričakuje nobenih vrnjenih podatkov, razen kode statusa odziva spletnega servisa. Ta določa uspešnost izvedbe zahteve. Glavna metoda konzolne aplikacije, ki se izvede ob poganjanju aplikacije, pa le zaporedno pokliče obe metodi za pošiljanje zahtevkov spletnemu servisu, da se izvedeta.



```
40 }
41
42 // calls web service method for sending guarantee expiration notifications on products
43 private static async void guaranteeExpirationProduct(string key)
44 {
45     HttpRequest request = HttpRequest.CreateHttp(String.Format(serviceUrl, "guaranteeExpiration
46     request.Headers.Add(HttpRequestHeader.Authorization, key);
47     request.Method = "POST";
48     request.ContentLength = 0;
49
50     try
51     {
52         HttpResponseMessage response = await request.GetResponseAsync() as HttpResponseMessage;
53
54         if (response.StatusCode == HttpStatusCode.OK)
55         {
56             Console.WriteLine(String.Format("Guarantee expiration notifications on products succesful
57         }
58     }
59     catch (Exception e)
60     {
61         Console.WriteLine(String.Format("ERROR occurred when sending request for guarantee expiration
62     }
63     finally
64     {
65         request.Abort();
66         resetEventProduct.Set();
67     }
68 }
69
70 static void Main(string[] args)
71 {
72     guaranteeExpirationReceipt("EAD771D65EC0450B99E10F1E286E8F2E");
73     resetEventReceipt.WaitOne();
74
75     guaranteeExpirationProduct("3897CE504744442EBD0F818059E4CA38");
76     resetEventProduct.WaitOne();
77 }
78 }
79
80
```

Slika 4.11: Konzolna aplikacija za pošiljanje zahtevkov spletnemu servisu za pošiljanje e-poštnih obvestil o izteku garancije.

5. Delovanje informacijskega sistema s primeri uporabe

Z vidika končnih uporabnikov se razvit informacijski sistem deli na dva dela. Prvi je spletna aplikacija, drugi pa mobilna aplikacija. Spletna aplikacija je namenjena vsem vrstam uporabnikov tega informacijskega sistema – uporabniki, trgovci, trgovci administratorji in administratorji. Mobilna aplikacija je namenjena običajnim uporabnikom in zato ne podpira funkcionalnosti, ki jih spletna aplikacija nudi drugim vrstam uporabnikov informacijskega sistema. Prek mobilne aplikacije se tako ne da upravljati uporabnikov (funkcionalnost administratorja) ali ustvarjati računov (funkcionalnost trgovca in trgovca administratorja). Vse vrste uporabnikov pa imajo možnost informacijski sistem uporabljati kot običajni uporabniki. V meniju spletne aplikacije je na voljo povezava do uporabniške strani, kjer je mogoče dodajanje računov in upravljanje obstoječih. Tudi prijava v mobilno aplikacijo je mogoča za vse vrste uporabnikov. Spletna aplikacija deluje znotraj spletnih brskalnikov. Mobilna aplikacija pa deluje na pametnih telefonih z operacijskim sistemom Android različice 4.4 s kodnim imenom KitKat ali novejšim.

5.1. Primeri uporabe: spletna aplikacija

The screenshot shows a web browser window with the URL <https://digiracun.azurewebsites.net>. The page title is 'Prijava - Digi Račun'. The main content area has a blue header with the 'Digi Račun' logo and two links: 'Pregled izdelkov za nakup' and 'Ponastavi geslo'. Below the header, there are two columns of forms. The left column is titled 'Registracija' and 'Novi uporabniki'. It contains five input fields: 'Uporabniško ime:' (with a red asterisk and 'Obvezen vnos' below it), 'Ime:' (with 'Janez' entered), 'Priimek:', 'E-pošta:' (with 'janez.vzorec@' entered and a red asterisk and 'Neveljaven e-poštni naslov' below it), and 'Geslo:' (with '****' entered and a red asterisk and 'Geslo mora vsebovati vsaj 8 znakov' below it). A 'Registracija' button is at the bottom. The right column is titled 'Prijava' and 'Obstoječi uporabniki'. It contains two input fields: 'Uporabniško ime:' and 'Geslo:'. A 'Prijava' button is below the password field, and a 'Pozabljeno geslo' link is below it.

Slika 5.1: Prijavna stran spletne aplikacije IS.

Vstopna stran (Slika 5.1) v razvit informacijski sistem je prijavna stran. Na levi se nahaja obrazec za registracijo novega uporabnika. Na desni strani se nahaja obrazec za prijavo obstoječih uporabnikov. Na strani je izvedeno preverjanje vnesenih podatkov. Za registracijo so obvezni podatki uporabniško ime, e-poštni naslov in geslo. Uporabniško ime je lahko

sestavljeno le iz malih ali velikih črk ter števil. Dolžina je omejena od vsaj 5 do 20 znakov. Uporabniško ime ne sme biti zasedeno s strani katerega drugega uporabnika, saj mora biti unikatno. E-poštni naslov mora biti pravilne oblike. Geslo ne sme biti krajše od 8 znakov. Za uspešno prijavo je treba obvezno vnesti tako uporabniško ime, kot tudi geslo. Ob prijavi je uporabnik preusmerjen na začetno stran glede na vrsto uporabnika. Običajni uporabnik je preusmerjen na stran, ki omogoča dodajanje računov in upravljanje obstoječih. Trгоvec je preusmerjen na stran, ki omogoča ustvarjanje novih računov in upravljanje obstoječih. Trгоvec administrator je preusmerjen na stran, ki omogoča dodajanje izdelkov v ponudbo in upravljanje obstoječih. Administrator pa je preusmerjen na stran, ki omogoča upravljanje uporabnikov in trgovin.



Slika 5.2: Stran za ponastavitev gesla v spletni aplikaciji IS.

Če uporabnik ne ve svojega gesla, lahko na strani za ponastavitev gesla (Slika 5.2), le-tega ponastavi. Na strani mora vnesti svoje uporabniško ime. Izvedeno je preverjanje, saj je uporabniško ime obvezen podatek in brez njega ni mogoče ponastaviti gesla. Uporabnik na svoj e-poštni naslov, ki ga je navedel med registracijo ali pa kasneje spremenil znotraj informacijskega sistema, prejme novo samodejno generirano geslo. Novo geslo je dolgo 8 znakov in vsebuje vsaj 1 poseben znak, ki ni črkovni ali številski. Geslo je mogoče spremeniti na strani za urejanje uporabniškega profila.

Admin - Digi Račun

Varno | https://digiracun.azurewebsites.net/AdminForm.aspx

sysadmin
Odjava
Uporabniški profil
Uporabniška stran

Upravljanje uporabnikov

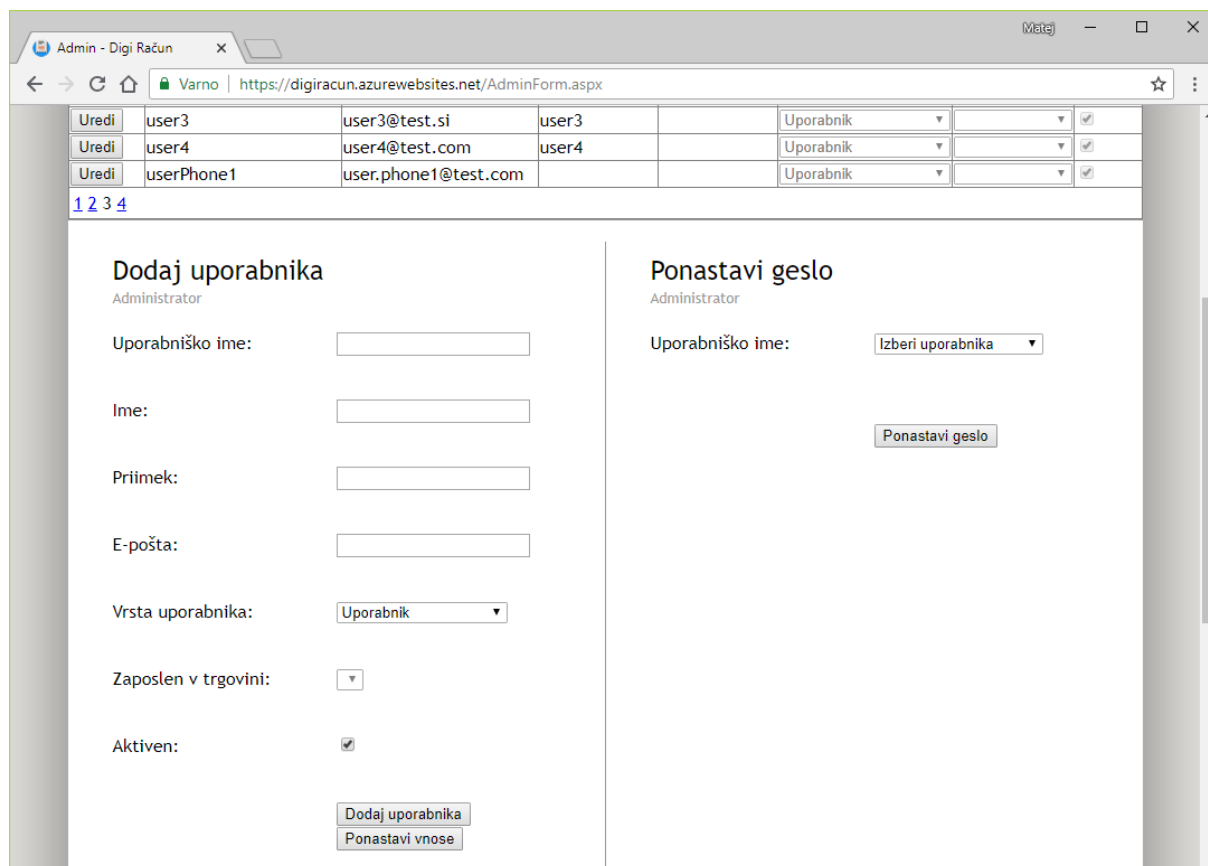
Administrator

	Uporabniško ime	E-pošta	Ime	Priimek	Vrsta uporabnika	Trgovina	Aktiven
Uredi	sysadmin	sysadmin@digiracun.net			Administrator		<input checked="" type="checkbox"/>
Uredi	test123	test@test.si			Uporabnik		<input checked="" type="checkbox"/>
Uredi	testUser25	gasilec.samo@112.si	Samo	Gasilec	Uporabnik		<input checked="" type="checkbox"/>
Uredi	trgovec7	trgovec7@trgovina7.si			Trgovec administrator	Trgovina 7	<input checked="" type="checkbox"/>
OK	TrgovecAdministrator	trgovec.administrator@trgovina7.si	Trgovec	Administrator	Trgovec administrator	Trgovina Svet	<input checked="" type="checkbox"/>
Prekliči							
Uredi	user1	user1@test.com	User		Uporabnik		<input checked="" type="checkbox"/>
Uredi	user2	user2@test2.com	user2		Uporabnik		<input type="checkbox"/>
Uredi	user3	user3@test.si	user3		Uporabnik		<input checked="" type="checkbox"/>
Uredi	user4	user4@test.com	user4		Uporabnik		<input checked="" type="checkbox"/>
Uredi	userPhone1	user.phone1@test.com			Uporabnik		<input checked="" type="checkbox"/>

1 2 3 4

Slika 5.3: Upravljanje uporabnikov v spletni aplikaciji IS.

Administrator informacijskega sistema ima možnost upravljanja z registriranimi uporabniki (Slika 5.3). Obstoječim uporabnikom lahko spremeni e-poštni naslov, ime, priimek, vrsto uporabnika, trgovino in ali je uporabnik aktiven v sistemu. Če uporabnik v sistemu ni aktiven, to pomeni, da ne more uporabljati svojega uporabniškega profila in se z njim ne more prijaviti v ta informacijski sistem. Trgovino je uporabniku mogoče določiti le, če je uporabnik trgovec ali trgovec administrator. To pomeni, da je zaposlen v določeni trgovini in lahko za to trgovino izdaja račune. Pri spreminjanju uporabnikovih podatkov veljajo enake omejitve nad podatki, kot pri registraciji, saj je izvedeno enako preverjanje. Onemogočeno pa je spreminjanje podatkov trenutnega uporabnika, torej administratorja.



Slika 5.4: Administrator lahko v spletni aplikaciji IS doda novega uporabnika ali obstoječemu ponastavi geslo.

Administrator lahko v informacijski sistem dodaja nove uporabnike (Slika 5.4). Tudi v tem primeru veljajo enake omejitve nad podatki, kot če registracijo zase izvaja nov uporabnik na prijavnih strani. Razlika je v tem, da lahko administrator pri dodajanju uporabnika določi nivo uporabniških pravic. Če nov uporabnik registracijo izvaja sam, je vedno določen nivo pravic običajnega uporabnika. Administrator pri dodajanju uporabnika lahko določi tudi v kateri trgovini deluje uporabnik, če je za vrsto uporabnika določil trgovec ali trgovec administrator. Določi lahko da novi uporabnik sprva ni aktiven in ne more uporabljati tega informacijskega sistema. Lahko ga namreč aktivira kasneje. Geslo se v tem primeru generira samodejno in je uporabniku poslano na vnesen e-poštni naslov.

Obstoječim uporabnikom lahko administrator ponastavi geslo z izbiro uporabniškega imena iz spustnega seznama (Slika 5.4). Uporabniku je na e-poštni naslov poslano novo samodejno generirano geslo.

Upravljanje trgovin

Administrator

	Trgovina	Naslov	Telefon	E-pošta	Spletna stran
Uredi	Trgovina 1	Slovenska ulica 1, 1000 Ljubljana, Slovenija	031123456	info@trgovina1.si	http://www.trgovina1.si
Uredi	Trgovina 2	Ulica 2, 2000 Maribor, Slovenija			
Uredi	Trgovina 4	Ulica 4, 4000 Koper, Slovenija			
Uredi	Trgovina 5	Naslov 5, 5000 Ne vem, Slovenija	05741852		
Uredi	Trgovina 7	Naslov 7, 7000 Neki, Slovenija	+7777777	info@trgovina7.si	http://www.trgovina7.si
Uredi	Trgovina 8	Ulica 8, 8000 Novo mesto, Slovenija	031963852	info@trgovina8.si	https://www.trgovina8.si
Uredi	Trgovina Svet	Ulica 4, 1000 Ljubljana, Slovenija	01123456	info@trgovina-svet.si	http://www.trgovina-svet.si

Dodaj trgovino

Administrator

Ime trgovine:

* Obvezen vnos

Naslov:

* Obvezen vnos

Telefon:

* Neveljavna telefonska številka

E-pošta:

* Neveljaven e-poštni naslov

Spletna stran:

* Neveljaven spletni naslov

Slika 5.5: Upravljanje obstoječih in dodajanje novih trgovin v spletni aplikaciji IS.

Administrator lahko dodaja tudi nove trgovine ali spreminja podatke obstoječim (Slika 5.5). Pri dodajanju trgovine sta obvezna podatka ime trgovine in naslov. Telefon, e-pošta in spletna stran so opcijski podatki. Če pa so vneseni, je izvedeno preverjanje le-teh, da so napisani v pravilni obliki. Preverjanje podatkov je izvedeno pri spreminjanju podatkov obstoječe trgovine in pri dodajanju nove trgovine.

Trgovec Admin - Digi Račun

Varno | https://digiracun.azurewebsites.net/SellerAdminForm.aspx

Digi Račun

TrgovecAdministrator
 Odjava
 Uporabniški profil
 Uporabniška stran
 Pregled izdanih računov
 Dodaj izdelek

DODAJ IZDELEK

Pregled izdelkov v ponudbi
 Trgovec administrator

	Izdelek	Opis	Cena	Kategorija	V ponudbi
Uredi	Microsoft docking station za Surface Pro 3/4	Priklopna postaja Surface Dock za Surface Pro 3 in 4 - spremenite svoj tablični računalnik v prenosnik.	198,70 €	Računalništvo in telefonija	<input checked="" type="checkbox"/>
Uredi	Microsoft tablični računalnik Surface Pro 2017	Tablični računalnik Microsoft Surface Pro 2017 s 31,2 cm (12.3") zaslonom v ločljivosti 2736x1824. Poganja ga Intel® Core™ i5 procesor, ima ...	1.175,60 €	Računalništvo in telefonija	<input checked="" type="checkbox"/>

Slika 5.6: Upravljanje izdelkov v ponudbi v spletni aplikaciji IS.

Trgovec administrator na svoji začetni strani vidi izdelke, ki so v ponudbi trgovine v kateri je zaposlen (Slika 5.6). Dodaja lahko nove izdelke ali pa ureja obstoječe. Za dodajanje ali urejanje izdelka se odpre nova stran s podrobnostmi o izdelku (Slika 5.7). Trgovec administrator ima v meniju desno zgoraj na voljo tudi povezavo do strani, kjer ima pregled nad vsemi računi izdanimi znotraj svoje trgovine (Slika 5.9). Tam lahko ustvarja nove račune (Slika 5.10).

Dodajanje/urejanje artikla: X

Varno | <https://digiracun.azurewebsites.net/SellerAdminForm.aspx>

Digi Račun

TrgovecAdministrator
Odjava
Uporabniški profil
Uporabniška stran
Pregled izdanih računov
Nazaj na pregled izdelkov v ponudbi

Dodaj izdelek

Trgovec administrator

Naziv izdelka: Kategorija izdelka:

Cena: V ponudbi:

Opis:

Surface svinčnik vam ponuja izredno naraven občutek pri pisanju in risanju na pametne naprave. Ponuja poseben kot senčenja, 4096 nivoji površinske občutljivosti in gladko delovanje, brez težav. Pišite in rišite s preciznostjo svinčnika na eni strani in radirke na drugi. Pisanje pod določenim kotom omogoči senčenje skic, prav tako kot z grafitnim svinčnikom. 4096 nivoji površinske občutljivosti se odzivajo na najnežnejši dotik. Črte tečejo direktno s svinčnika na ekran, brez zamika. Svinčnik je kompatibilen z OneNote, Microsoft Office, Adobe Suite, Sketchable, Bluebeam Revu, Drawboard in Staffpad.

Tehnične podrobnosti:

Kompatibilnost: OneNote, Microsoft Office, Adobe Suite, Sketchable, Bluebeam Revu, Drawboard in Staffpad\n
Tip konektorja: Bluetooth 4.0\n
Teža: 20 g\n
Baterija: 1 x AAAA\n

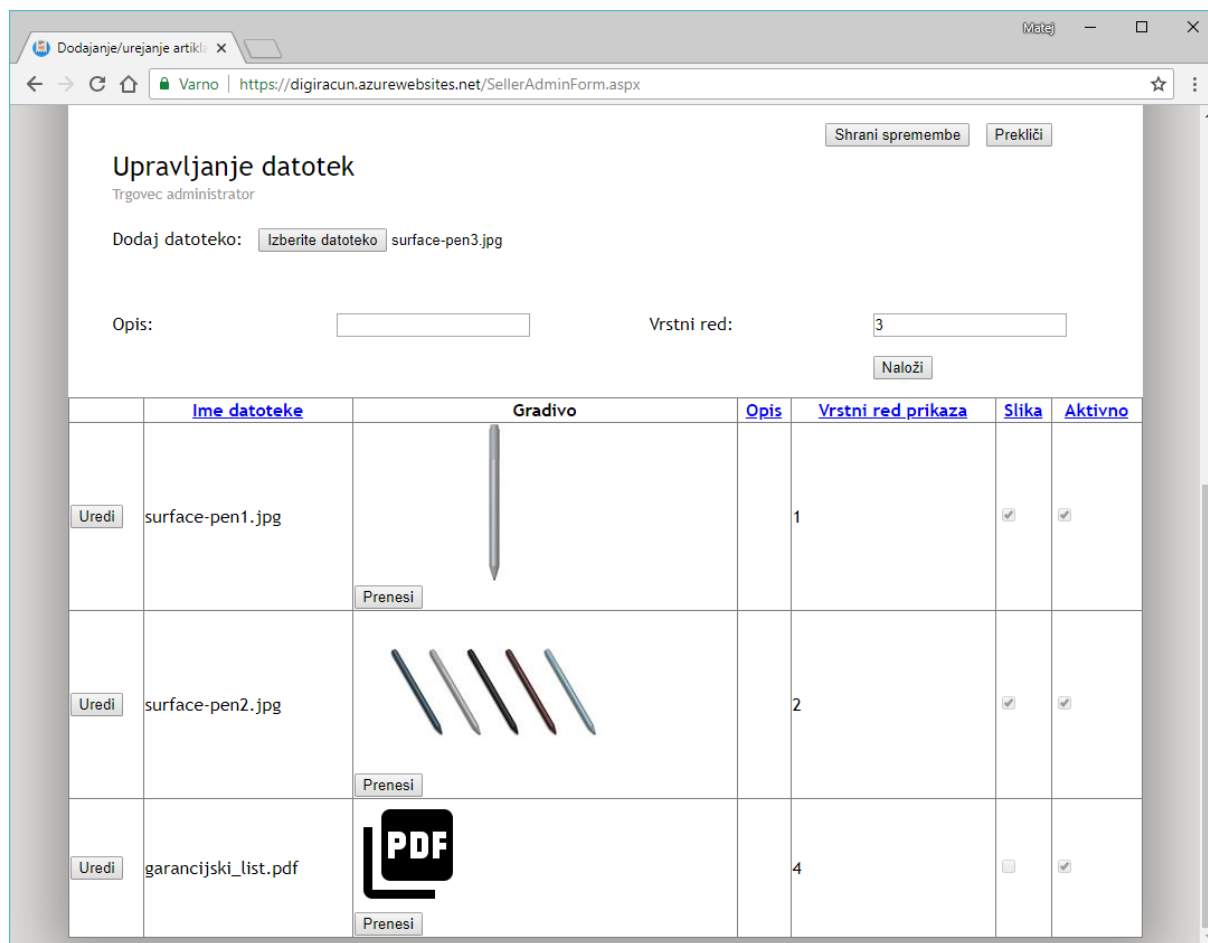
Upravljanje datotek

Trgovec administrator

Dodaj datoteko: Nobena datoteka ni izbrana

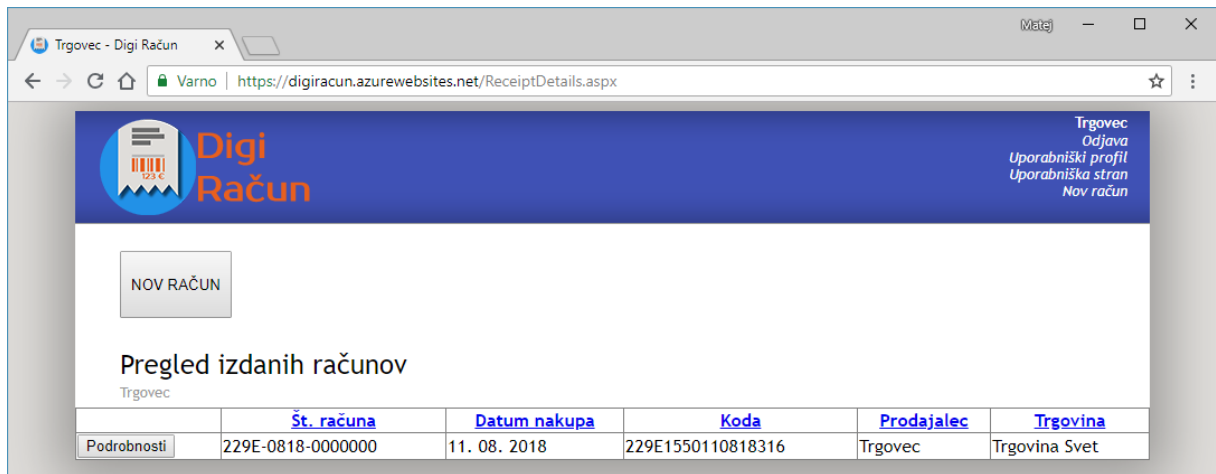
Slika 5.7: Dodajanje izdelka v ponudbo v spletni aplikaciji IS.

Trgovec administrator lahko dodaja izdelke v ponudbo svoje trgovine (Slika 5.7). Pri tem sta obvezna podatka naziv izdelka in cena. Slednja mora biti zapisana s številskimi vrednostmi. Izvedeno je preverjanje ustreznosti vnesenih podatkov. Izbrati je treba tudi kategorijo izdelka. V naprej izbrana vrednost je namreč tista, ki je prva po abecednem vrstnem redu. Določiti je mogoče tudi, ali je izdelek trenutno prikazan v ponudbi. Opis in tehnične podrobnosti sta opsijska vnosa. V ti dve vnosi polji je besedilo mogoče vnašati v več vrsticah. Prehod v novo vrstico se označi z »\n«. Pri prikazu bo namesto »\n« izveden prehod v novo vrstico.



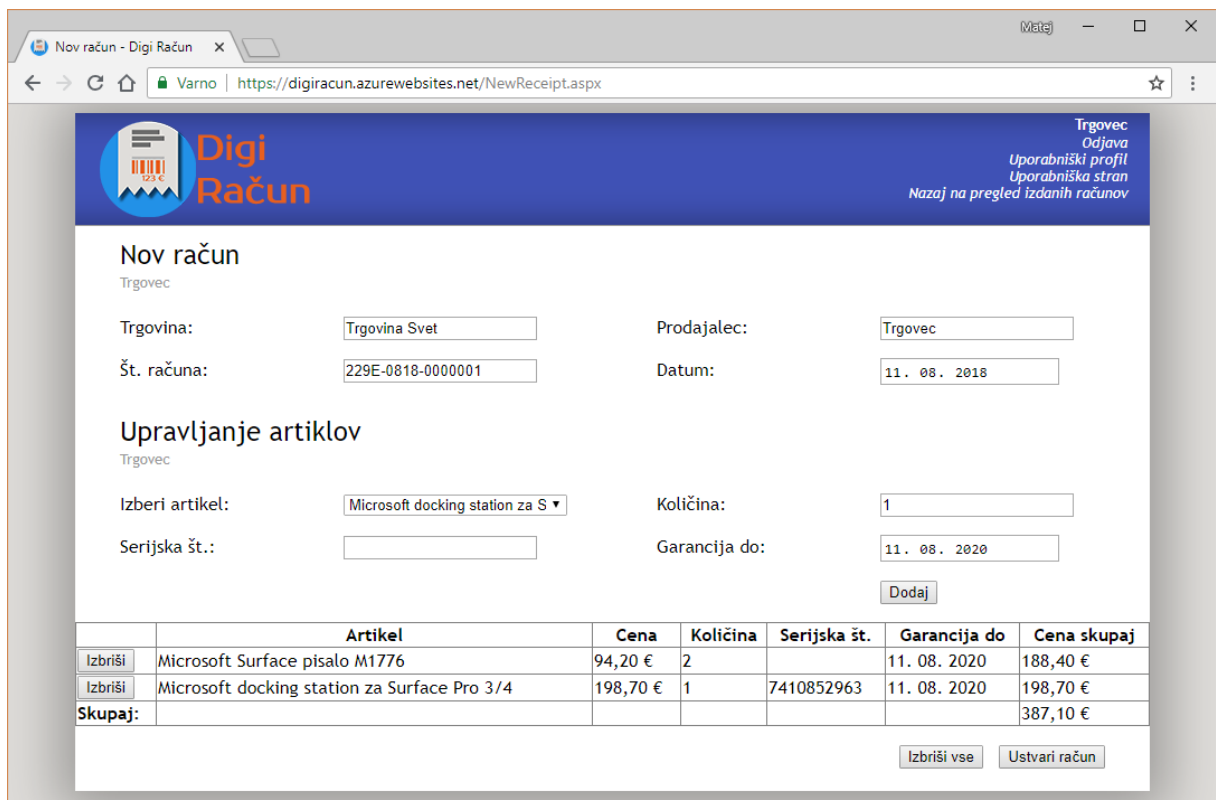
Slika 5.8: Dodajanje dodatnih gradiv o izdelku v spletni aplikaciji IS.

Ko so uspešno dodani osnovni podatki o izdelku, je mogoče dodati tudi slike in dokumente PDF (garancijski list, navodila za uporabo itd.) o izdelku (Slika 5.8). Obvezen podatek je datoteka sama. V primeru dodajanja slike se le-ta samodejno zmanjša na velikost 800 slikovnih pik po daljši stranici. Opis datoteke je opsijski. Vrstni red je samodejno v naprej izpolnjen z začetno vrednostjo 1. Določa namreč vrstni red prikaza dodatnih gradiv o izdelku. Podprti formati datotek so JPG, JPEG, PNG in PDF. Omogočeno je tudi urejanje že dodanih dodatnih gradiv o izdelku. Pri urejanju podatkov o dodatnem gradivu je mogoče spreminjati opis, vrstni red prikaza in ali je dodatno gradivo na izdelku prikazano. Vrste datoteke (ali je slika) ni mogoče spreminjati, saj je določena samodejno glede na končnico datoteke. Dodatna gradiva o izdelku je mogoče tudi prenesti na osebni računalnik.



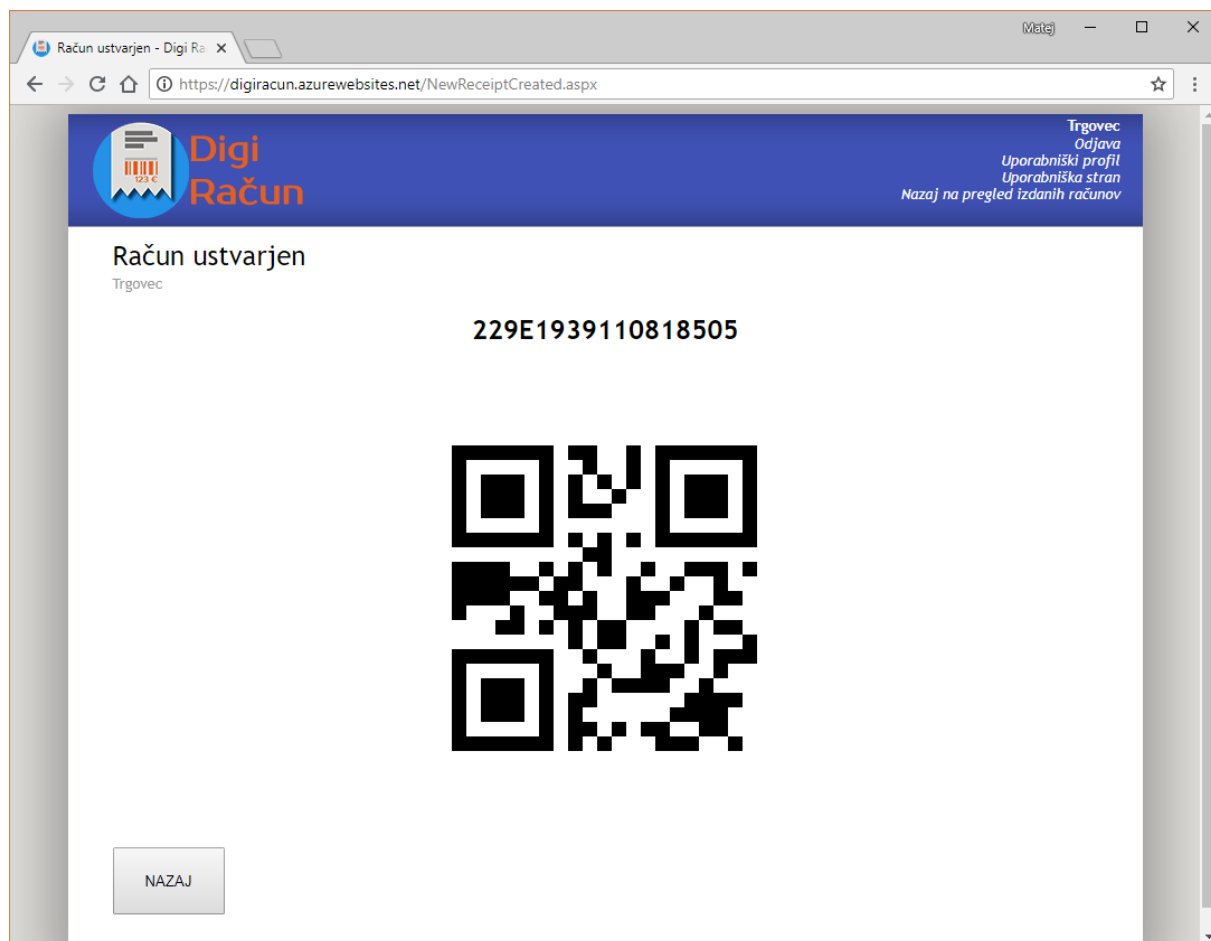
Slika 5.9: Pregled izdanih računov v spletni aplikaciji IS.

Trgovec ima na svoji začetni strani prikazane račune, ki jih je izdal sam (Slika 5.9). Dostop do strani za izdajanje računov ima tudi trgovec administrator. Ta pa ima prikazane vse račune izdane od vseh trgovcev v dotični trgovini. Trgovec lahko ustvarja nove račune in si ogleda podrobnosti že ustvarjenih računov, lahko pa jih tudi izbriše.



Slika 5.10: Ustvarjanje računa v spletni aplikaciji IS.

Pri ustvarjanju računa (Slika 5.10) se vrednosti za ime trgovine, ime prodajalca, št. računa in datum izdaje računa samodejno izpolnijo in jih ni mogoče spreminjati. Med ustvarjanjem računa je mogoče upravljati le z izdelki na računu. V spustnem seznamu so na voljo vsi izdelki, ki so v ponudbi trgovine. Izbranemu izdelku se določi še količino, opsijsko serijsko številko in datum izteka garancije. Količina je privzeto nastavljena na 1, datum izteka garancije pa na čez dve leti od trenutnega datuma. V tabeli na dnu strani so prikazani vsi izdelki, ki so dodani na račun. Med ustvarjanjem računa je mogoče izdelke z računa tudi brisati. Praznega računa, torej takega brez izdelkov, ni mogoče ustvariti.



Slika 5.11: Prikaz kod ustvarjenega računa v spletni aplikaciji IS.

Ko je račun uspešno ustvarjen, se prikaže stran s kodo računa (Slika 5.11). Pod njo je prikazana še koda QR. Z vnosom kode uporabnik doda izdan račun in ga ima s tem shranjenega v svojem uporabniškem profilu tega informacijskega sistema. Če pa uporablja mobilno aplikacijo, izdan račun lahko doda z branjem kode QR.

The screenshot shows a web browser window with the URL `https://digiracun.azurewebsites.net/SellerForm.aspx`. The page header features the 'Digi Račun' logo and navigation links for 'Trgovec', 'Odjava', 'Uporabniški profil', 'Uporabniška stran', and 'Nazaj na pregled izdanih računov'. The main content area is titled 'Podrobnosti računa' and contains the following information:

Trgovina: Prodajalec:
Št. računa: Datum:

Artikel	Cena	Količina	Serijska št.	Garancija do	Cena skupaj
Microsoft docking station za Surface Pro 3/4	198,70 €	1	7410852963	11. 08. 2020	198,70 €
Microsoft Surface pisalo M1776	94,20 €	2		11. 08. 2020	188,40 €
Skupaj:					387,10 €

Below the table, the 'Koda računa' section displays a QR code and the code `229E1939110818505`. At the bottom, there are two buttons: 'Izbrši račun' and 'Nazaj'.

Slika 5.12: Pregled podrobnosti izdanega računa s strani trgovca v spletni aplikaciji IS.

Trgovec si lahko ogleda podrobnosti o izdanem računu (Slika 5.12). Vidni so vsi enaki podatki kot med ustvarjanjem računa z dodatkom kode sestavljene iz črk in števil ter kode QR prek katerih lahko uporabnik doda račun. Spremembe na računu niso mogoče. Mogoče pa je brisanje računa. Trgovčevo brisanje računa ne vpliva na uporabnikovo dodajanje računa. Če je uporabnik pridobil kodo računa, ga lahko vedno doda, ne glede na to, če ga je trgovec izbrisal.

The screenshot shows a web browser window with the URL `https://digiracun.azurewebsites.net/EditReceipt.aspx`. The page title is 'Uporabnik - Digi Račun'. The header features the 'Digi Račun' logo and user information for 'Janez Odjava', including a profile link and a link to view purchase items. The main content area is divided into two sections: 'Dodajanje računa prek slike' and 'Dodajanje računa prek kode'. The first section contains form fields for 'Trgovina' (Big Bang), 'Št. računa' (2570-2572-1113768), 'Datum nakupa' (31. 01. 2018), and 'Garancija do' (dd. mm. 1111), along with a file upload field for 'Izberi sliko računa' (20180812_09..._iOS-2.jpg) and a 'Dodaj' button. The second section has a 'Koda računa' field and a 'Dodaj' button. Below these is the 'Upravljanje računov' section, which includes a table of existing receipts.

	Trgovina	Št. računa	Datum nakupa
Uredi	Trgovina 1	DB9C-0818-0000000	12. 08. 2018
Uredi	Trgovina Svet	229E-0818-0000001	11. 08. 2018
Uredi	Merkur		19. 07. 2018
Uredi	Trgovina Test	MT-0618-00026362	06. 06. 2018

Slika 5.13: Upravljanje računov uporabnika v spletni aplikaciji IS.

Uporabnik ima na svoji začetni strani na voljo možnosti za dodajanje računa – klasičnega papirnatega računa prek slike ali računa izdanega znotraj tega informacijskega sistema prek kode (Slika 5.13). Spodaj so prikazani že dodani računi. Pri dodajanju računa prek slike so obvezni vnosi ime trgovine, datum nakupa in izbira slike računa iz računalnika. Številka računa in datum izteka garancije na računu sta opsijska podatka. Datum nakupa je v naprej nastavljen na trenutni datum. Dodana slika se samodejno zmanjša na velikost 1000 slikovnih pik po daljši stranici. Pri dodajanju računa prek kode je edini in hkrati tudi obvezen vnos le koda računa. Posamezen račun je prek kode mogoče dodati le enkrat. Ko ga doda en uporabnik, ga ne more dodati nihče več.

The screenshot shows a web browser window with the URL `https://digiracun.azurewebsites.net/UserForm.aspx`. The page title is "Podrobnosti računa - Digi Račun". The header includes a logo for "Digi Račun" and a user profile for "Janez" with options for "Odjava", "Uporabniški profil", "Pregled izdelkov za nakup", "Pregled grafov izdelkov na računih", and "Nazaj na upravljanje računov".

The main content area is titled "Podrobnosti računa" and "Uporabnik". It contains several input fields for invoice details:

- Trgovina: Big Bang
- Št. računa: 2570-2572-1113768
- Datum nakupa: 31. 01. 2018
- Garancija do: dd. mm. 1111

Below these fields is a photograph of a receipt. The receipt text is as follows:

Big Bang, trgovina in storitve, d.o.o.
 Smartinska cesta 152, 1000 Ljubljana
 ID številka za DDV : S118224526
 BIG BANG NOVO MESTO, Otoška 5
 Tel. : 07/48 81 250 Fax. : 07/ 393 51 91
 Cas:17:34

Racun št. : 2570-2572-1113768

Kraj: Novo mesto Datum: 31.01.2018
 Blagajna: 2572
 Poslovalnica: 2570

Artikel	EM	DDV	Kol	Znesek
619659144876	KOS 22%		1,00	
USB DRIVE ULTRA FIT 16GB SANDISK				
CENA:	11,99			11,99
Vrednost blaga:				11,99
Popust:				0,00
DDV:				2,16
Znesek računa:				11,99
BA KARTICA :				11,99
Stopnja DDV	Vrednost z DDV	Znesek DDV		
22,00 %	11,99	2,16		
Skupaj DDV :		2,16		

Slika 5.14: Pregled in urejanje podrobnosti računa dodanega prek slike v spletni aplikaciji IS.

Podrobnosti posameznega računa se odprejo na novi strani. Pri tem se prikaz le-teh razlikuje glede na vrsto dodanega računa. Če je račun dodan prek slike (Slika 5.14) je mogoče spreminjati njegove podatke, kot so ime trgovine, številka računa, datum nakupa in datum izteka garancije. Trgovina in datum nakupa sta tako kot pri dodajanju računa obvezna podatka. Sliko računa je mogoče prenesti na računalnik. Račun je mogoče tudi izbrisati.

Podrobnosti računa - Digiracun

Varno | https://digiracun.azurewebsites.net/UserForm.aspx

Janez
Ojjava
Uporabniški profil
Pregled izdelkov za nakup
Pregled gradiv izdelkov na računih
Nazaj na upravljanje računov




Podrobnosti računa

Uporabnik

Trgovina: Št. računa:

Datum nakupa:





Artikel	Cena	Količina	Serijska št.	Garancija do	Cena skupaj
Microsoft docking station za Surface Pro 3/4	198,70 €	1	7410852963	11. 08. 2020	198,70 €
Microsoft Surface pisalo M1776	94,20 €	2		11. 08. 2020	188,40 €
Skupaj:					387,10 €

Artikel	Gradivo	Opis
Microsoft docking station za Surface Pro 3/4	 Prenesi	
Microsoft Surface pisalo M1776	 Prenesi	
Microsoft Surface pisalo M1776		

Slika 5.15: Pregled podrobnosti računa dodanega prek kode v spletni aplikaciji IS.

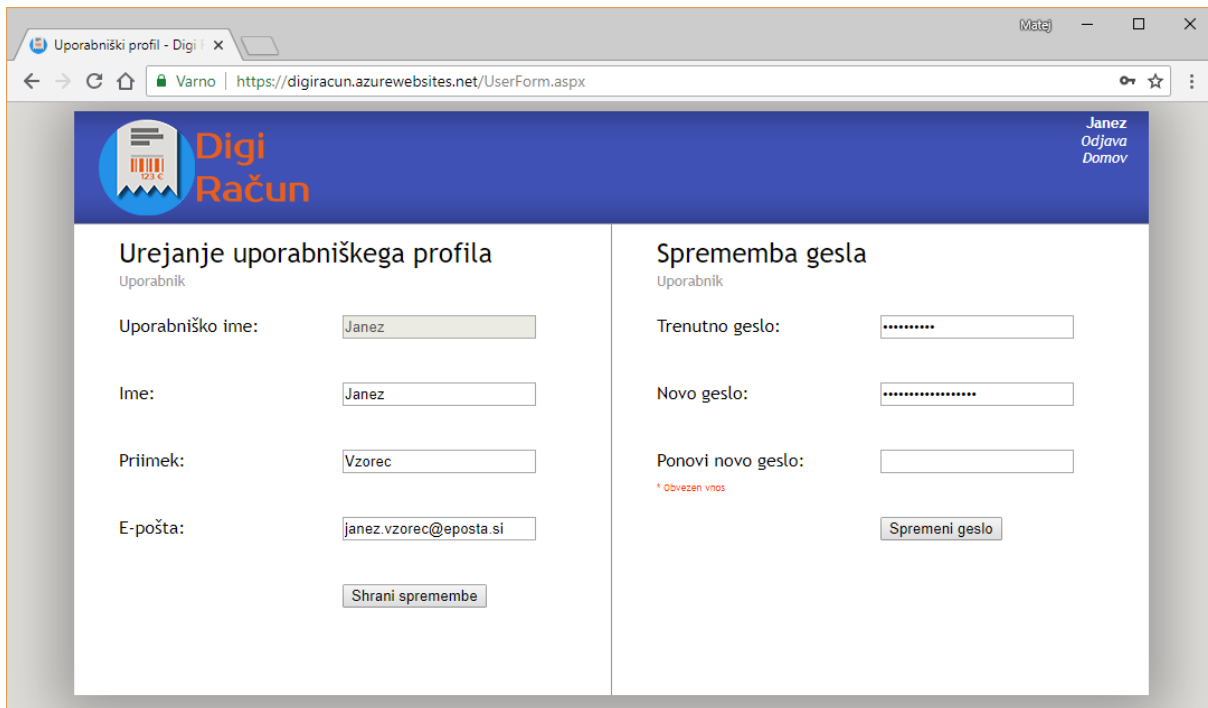
Če je račun ustvarjen znotraj tega informacijskega sistema in je nato dodan prek kode, se prikaz podrobnosti nekoliko razlikuje (Slika 5.15). Pod osnovnimi podatki, kot so ime trgovine, številka računa in datum nakupa, se nahaja tabela z izdelki na računu. Ta prikazuje ime izdelka, ceno, količino, morebitno serijsko številko, datum izteka garancije izdelka in skupno ceno glede na količino izdelkov. V zadnji vrstici tabele se nahaja skupna cena računa. Pod to tabelo se nahaja še ena, ki prikazuje dodatna gradiva o izdelkih na tem računu. To so slike oz. dokumenti PDF. V tej tabeli je navedeno ime izdelka na katerega se gradivo navezuje, gradivo samo, ki ga je mogoče prenesti in morebiten opis gradiva. Na takem računu ni mogoče spreminjati nobenih podatkov. Račun pa je seveda mogoče izbrisati. Slednje ne vpliva na prikaz računa pri trgovcu, ki ga je izdal. Račun je odstranjen le z uporabniškega profila uporabnika.

The screenshot shows a web browser window with the address bar displaying 'https://digiracun.azurewebsites.net/UserMaterials.aspx'. The page header includes the 'Digi Račun' logo and user information for 'Janez Odjava'. Below the header, there is a 'Filtriranje prikaza' section with various product category filters. The main content is a table with columns for 'Trgovina', 'Št. računa', 'Datum nakupa', 'Artikel', 'Gradivo', 'Opis', and 'Slika'. The table lists four items, all purchased on 12. 08. 2018. The first three items are Logitech Tipkovnica Wireless K360 keyboards, and the fourth is a SanDisk USB ključ Cruiser Ultra Fit V2 64GB. Each item has a 'Preinesi' button and a checkmark in the 'Slika' column.

Trgovina	Št. računa	Datum nakupa	Artikel	Gradivo	Opis	Slika
Trgovina 1	DB9C-0818-0000000	12. 08. 2018	Logitech Tipkovnica Wireless K360	 Preinesi		<input checked="" type="checkbox"/>
Trgovina 1	DB9C-0818-0000000	12. 08. 2018	Logitech Tipkovnica Wireless K360	 Preinesi		<input checked="" type="checkbox"/>
Trgovina 1	DB9C-0818-0000000	12. 08. 2018	Logitech Tipkovnica Wireless K360	 Preinesi		<input checked="" type="checkbox"/>
Trgovina 1	DB9C-0818-0000000	12. 08. 2018	SanDisk USB ključ Cruiser Ultra Fit V2 64GB	 Preinesi		<input checked="" type="checkbox"/>

Slika 5.16: Prikaz dodatnih gradiv izdelkov z vseh računov v spletni aplikaciji IS.

Uporabnik lahko znotraj svojega uporabniškega profila prikaže tudi vsa dodatna gradiva izdelkov z vseh dodanih računov (Slika 5.16). Na tej strani je omogočeno tudi filtriranje prikaza po kategorijah izdelkov. Tabela prikazuje ime trgovine, kjer je bil izdelek kupljen, številko računa, datum nakupa izdelka, ime izdelka, gradivo samo, morebiten opis in ali gre za sliko. Gradivo je mogoče prenesti. Tabela omogoča tudi razvrščanje prikaza glede na ime trgovine, št. računa, datum nakupa, ime izdelka, opis in vrsto gradiva (slika ali dokument).









The screenshot shows a web browser window with the address bar displaying 'Varno | https://digiracun.azurewebsites.net/UserForm.aspx'. The page has a blue header with the 'Digi Račun' logo on the left and 'Janez Odjava Domov' on the right. The main content is divided into two columns. The left column, titled 'Urejanje uporabniškega profila', contains a sub-header 'Uporabnik' and four input fields: 'Uporabniško ime' (filled with 'Janez'), 'Ime' (filled with 'Janez'), 'Priimek' (filled with 'Vzorec'), and 'E-pošta' (filled with 'janez.vzorec@eposta.si'). A 'Shrani spremembe' button is at the bottom. The right column, titled 'Sprememba gesla', contains a sub-header 'Uporabnik' and three input fields: 'Trenutno geslo' (filled with dots), 'Novo geslo' (filled with dots), and 'Ponovi novo geslo' (empty). A red asterisk and the text '* Obvezen vnos' are below the 'Ponovi novo geslo' field. A 'Spremeni geslo' button is at the bottom.

Slika 5.17: Urejanje uporabniškega profila in sprememba gesla v spletni aplikaciji IS.

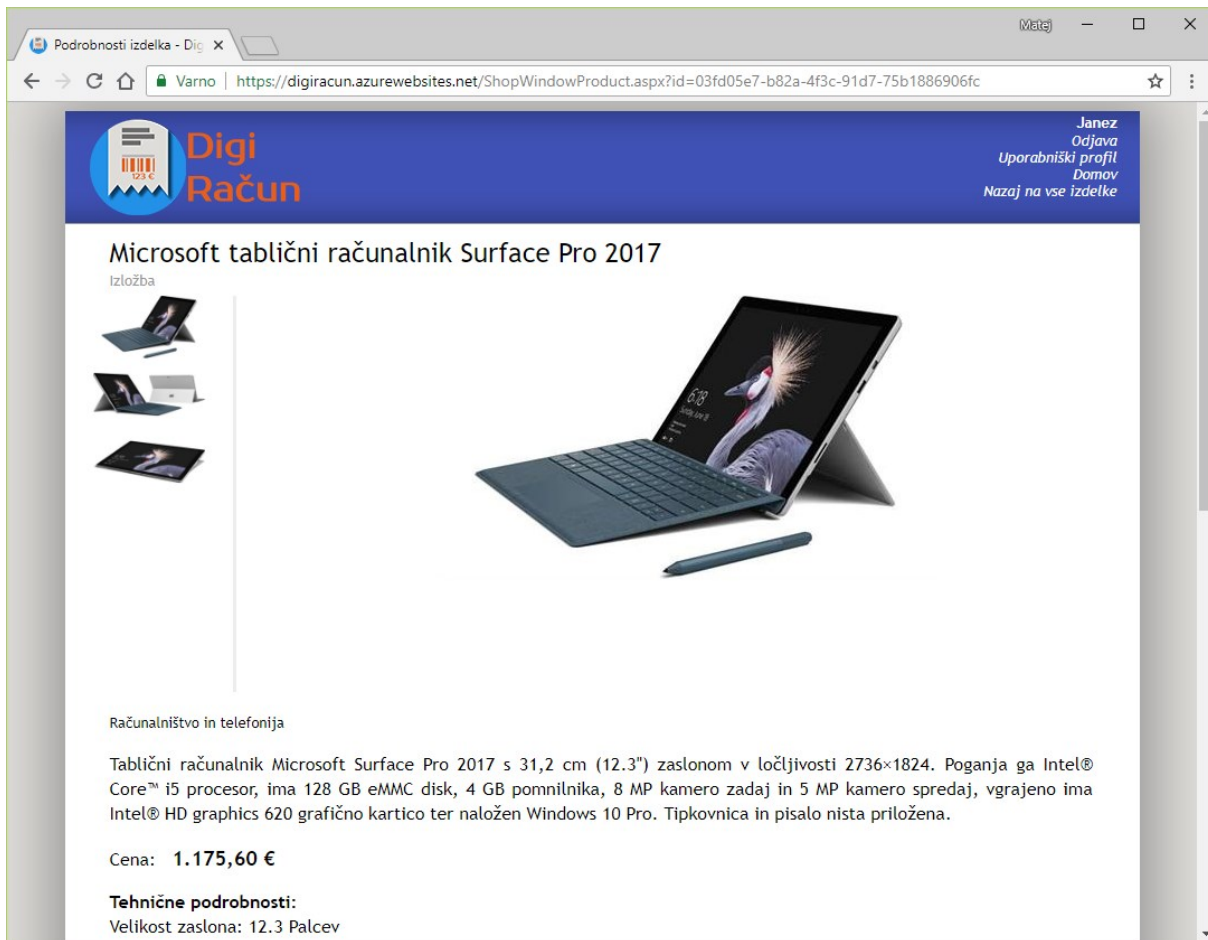
Ne glede na vrsto uporabnika, lahko vsak izmed uporabnikov ureja podatke o svojem uporabniškem profilu in spreminja geslo (Slika 5.17). Uporabnik lahko spreminja naslednje podatke: ime, priimek in e-poštni naslov. Ime in priimek nista obvezna podatka in jih lahko tudi odstrani. E-poštni naslov je obvezen podatek in mora biti zapisan v ustrezni obliki. Uporabniškega imena ni mogoče spreminjati. Pri spreminjanju gesla je treba najprej navesti trenutno geslo, ki mora biti pravilno. Nato je treba dvakrat zaporedoma vnesti novo geslo. Geslo ne sme biti krajše od 8 znakov. Dvakrat vneseno novo geslo se mora medsebojno ujemati. Trenutno geslo, novo geslo in ponovitev novega gesla so vsi obvezni podatki za uspešno spremembo gesla.

The screenshot shows a web browser window with the address bar displaying 'https://digiracun.azurewebsites.net/UserForm.aspx'. The page header features the 'Digi Račun' logo and a user profile for 'Janez' with options for 'Odjava', 'Uporabniški profil', and 'Domov'. The main content area is titled 'Izložba (pregled izdelkov za nakup)' and 'Uporabnik'. It displays a grid of six product listings:

 Microsoft docking station za Surface Pro 3/4 198,70 € Trgovina Svet	 SanDisk USB ključ Cruzer Ultra Fit V2 64GB 26,70 € Trgovina 1	 TestProdukt3 49,99 € Trgovina 1
 TestProdukt1 123,00 € Trgovina 1	 Microsoft tablični računalnik Surface Pro 2017 1.175,60 € Trgovina Svet	 Asus prenosnik Zenbook 3 Deluxe 1.815,20 € Trgovina 1

Slika 5.18: Pregled izdelkov za nakup v spletni aplikaciji IS.

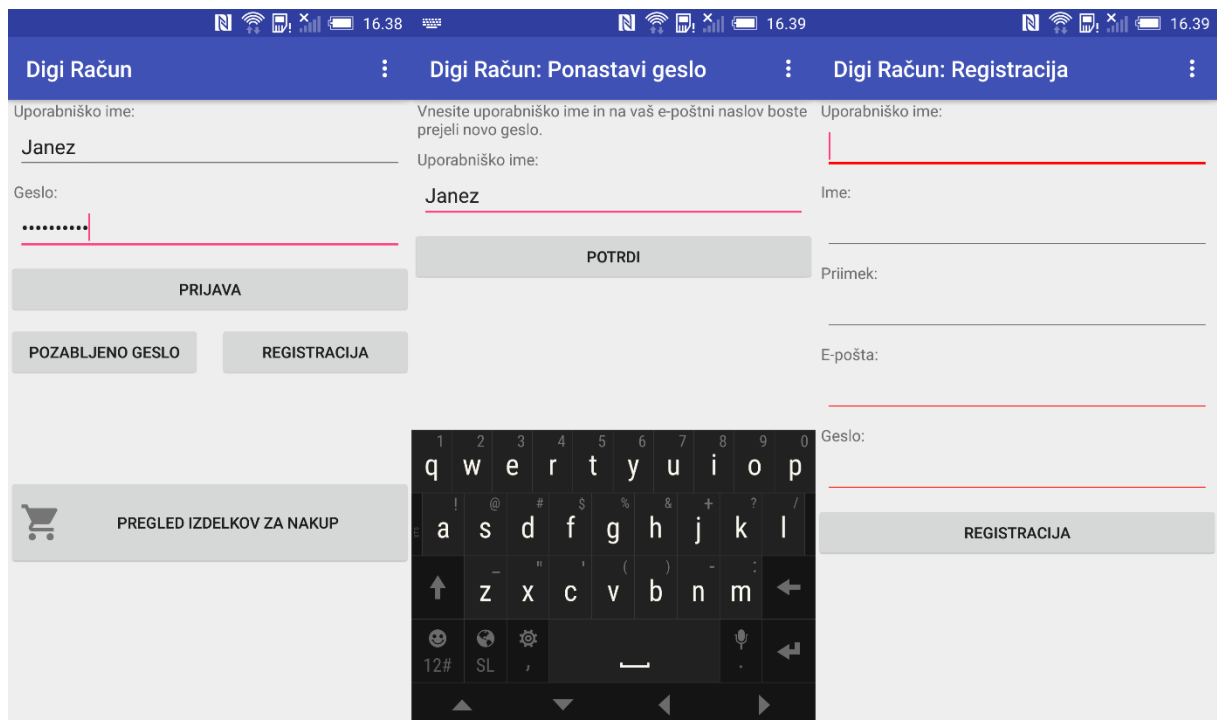
Tako prijavljenim, kot tudi neprijavljenim uporabnikom je na voljo pregled izdelkov za nakup (Slika 5.18). Na tej strani so prikazani vsi izdelki, ki jih ponujajo trgovine znotraj tega informacijskega sistema in so trenutno v ponudbi.



Slika 5.19: Pregled podatkov o izdelku iz ponudbe v spletni aplikaciji IS.

Ob izbiri enega izmed izdelkov se odpre nova stran, ki prikazuje podrobnejše podatke o izdelku (Slika 5.19). Prikazani so slike izdelka, kategorija v katero spada, opis, cena in tehnične podrobnosti. Na dnu strani se nahajajo še podatki o prodajalcu. Navedeni so ime trgovine, naslov, spletna stran, telefon in e-pošta. Prek povezave URL je izdelek mogoče deliti z drugimi osebami. Te pa za ogled izdelka ne rabijo imeti ustvarjenega uporabniškega profila v tem informacijskem sistemu.

5.2. Primeri uporabe: mobilna aplikacija



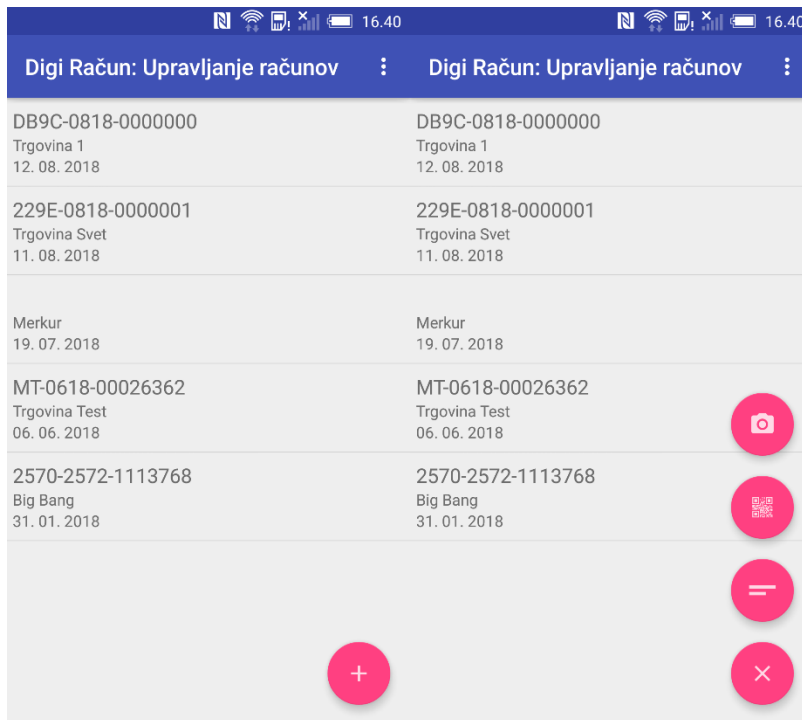
Slika 5.20: Prijava, ponastavitev gesla in registracija v mobilni aplikaciji IS.

Ob zagonu mobilne aplikacije se najprej prikaže prijavna aktivnost (Slika 5.20). Razen v primeru, ko je uporabnik v aplikacijo že prijavljen. Takrat se prikaže aktivnost, ki sledi prijavi v primeru uspešne prijave in sicer aktivnost s prikazom vseh računov. Za uspešno prijavo v mobilno aplikacijo sta obvezna vnosa uporabniško ime in geslo. Če kateri izmed obveznih vnosov manjka, se polje z manjkajočim podatkom obarva rdeče. Uporabnik je v aplikacijo prijavljen vse dokler ne izvede odjave, ki je na voljo v meniju aplikacije. S tem je uporabniku omogočeno hitrejše dodajanje računov in pregled nad že dodanimi, saj se je treba prijaviti le enkrat.

Pozabljeno geslo je mogoče ponastaviti tudi iz mobilne aplikacije (Slika 5.20). V kolikor je uporabnik na prijavi aktivnosti že vnesel uporabniško ime, se pri odpiranju aktivnosti za ponastavitev gesla to samodejno izpolni. Uporabniško ime je edini in hkrati tudi obvezen podatek za ponastavitev gesla. Novo generirano geslo uporabnik prejme na e-poštni naslov. Novo geslo se generira samodejno in je dolgo 8 znakov ter vsebuje vsaj 1 poseben znak.

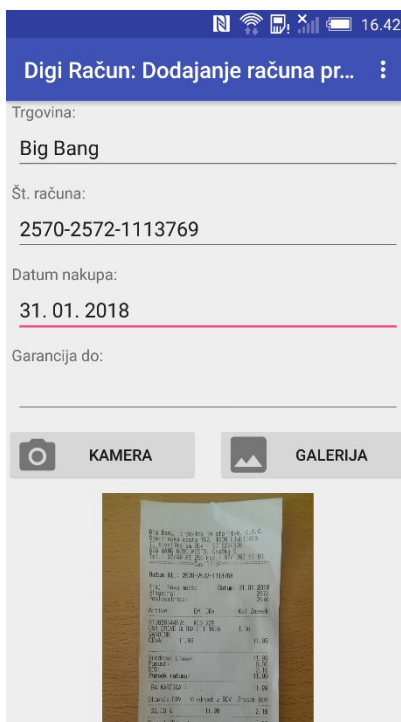
Če uporabnik še nima svojega uporabniškega profila, lahko registracijo izvede prek mobilne aplikacije (Slika 5.20). Če je na prijavi strani že vnesel uporabniško ime, se le-to samodejno izpolni ob odpiranju aktivnosti za registracijo. Pri registraciji so obvezna polja uporabniško ime, e-pošta in geslo. Uporabniško ime je lahko sestavljeno iz malih in velikih črk ter števil,

dolgo pa je lahko od 5 do 20 znakov. Uporabniško ime mora biti unikatno in tako ne sme biti že zasedeno s strani katerega drugega uporabnika. E-poštni naslov mora biti pravilne oblike. Geslo pa ne sme biti krajše od 8 znakov. Po uspešni registraciji je uporabnik samodejno preusmerjen na aktivnost s pregledom vseh računov.



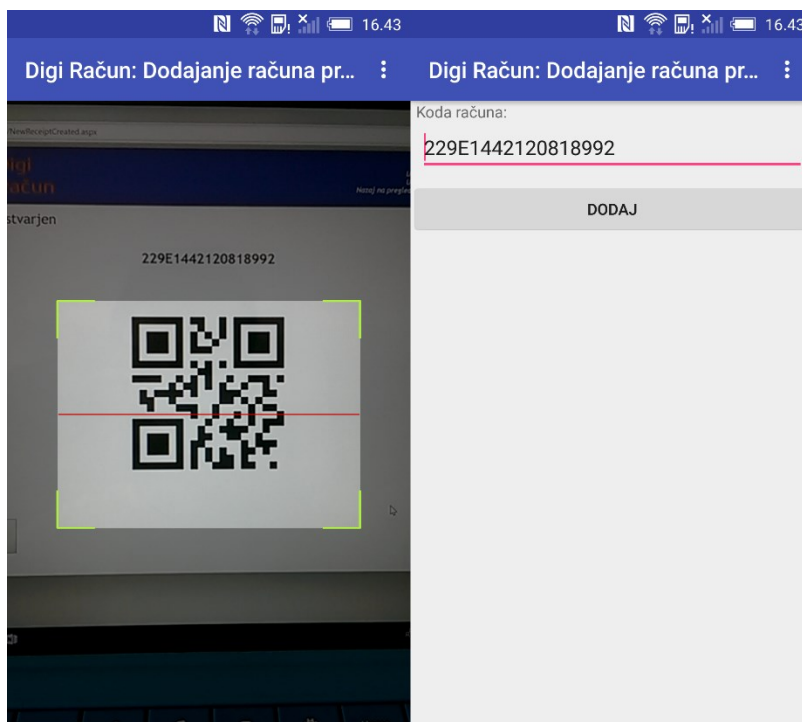
Slika 5.21: Pregled računov uporabnika v mobilni aplikaciji IS.

Mobilna aplikacija uporabniku ponuja pregled vseh dodanih računov (Slika 5.21). Računi so prikazani padajoče po datumu nakupa, tako da so najnovejši na vrhu. Od tu lahko uporabnik dodaja tudi nove račune. Doda lahko klasični papirnat račun prek slike ali pa račun ustvarjen znotraj tega informacijskega sistema prek kode. Aplikacija omogoča branje kode QR in ročni vnos kode. Za dodajanje računa je v spodnjem desnem kotu aktivnosti prikazan plavajoči akcijski gumb, ki nad seboj odpre gumbe za dodajanje računa. Zgornji omogoča dodajanje računa prek slike, pod njim se nahaja gumb za dodajanje računa prek kode QR, sledi mu gumb za dodajanje računa z ročnim vnosom kode, spodnji pa skrije vse gumbe za dodajanje računa.



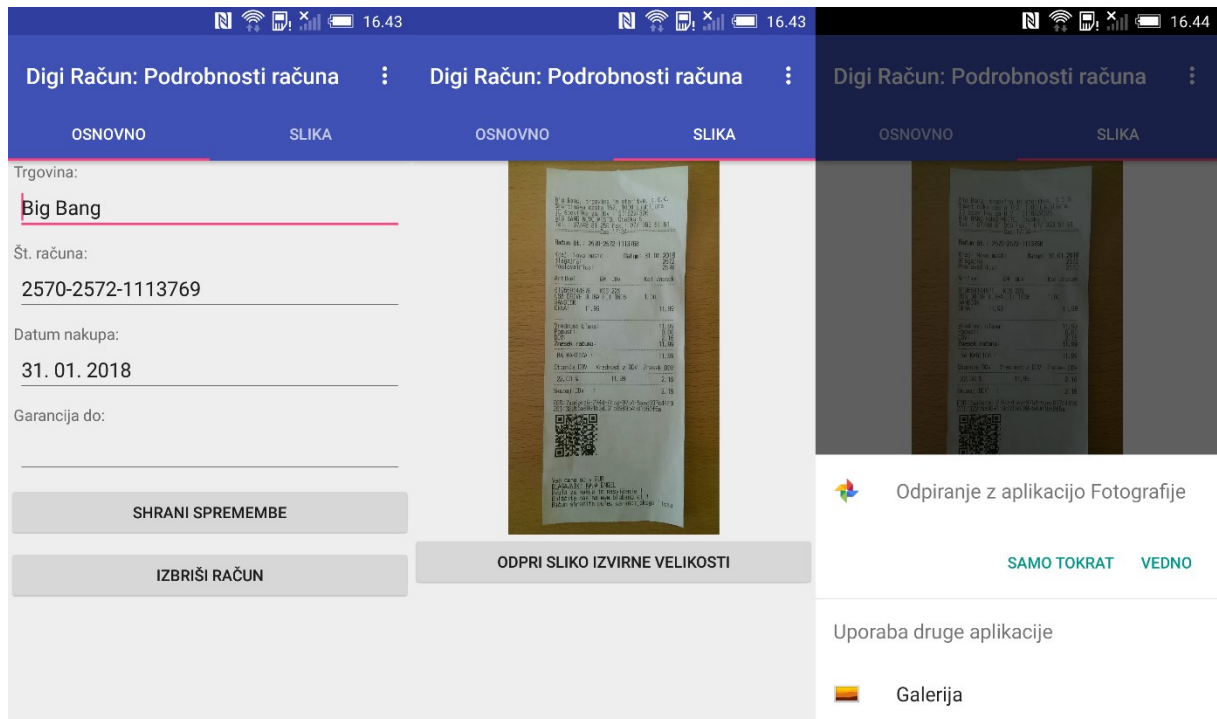
Slika 5.22: Dodajanje računa prek slike v mobilni aplikaciji IS.

Pri dodajanju papirnatega računa prek slike (Slika 5.22) so obvezni podatki ime trgovine, datum nakupa in slika računa. Datum nakupa je v naprej nastavljen na trenutni datum. Opcijski podatki so številka računa in datum izteka garancije. Sliko je mogoče dodati z odpiranjem kamere in zajemanjem slike ali iz galerije z izbiro ene izmed že ustvarjenih. Naložena slika se samodejno zmanjša na 1000 slikovnih pik po daljši stranici.



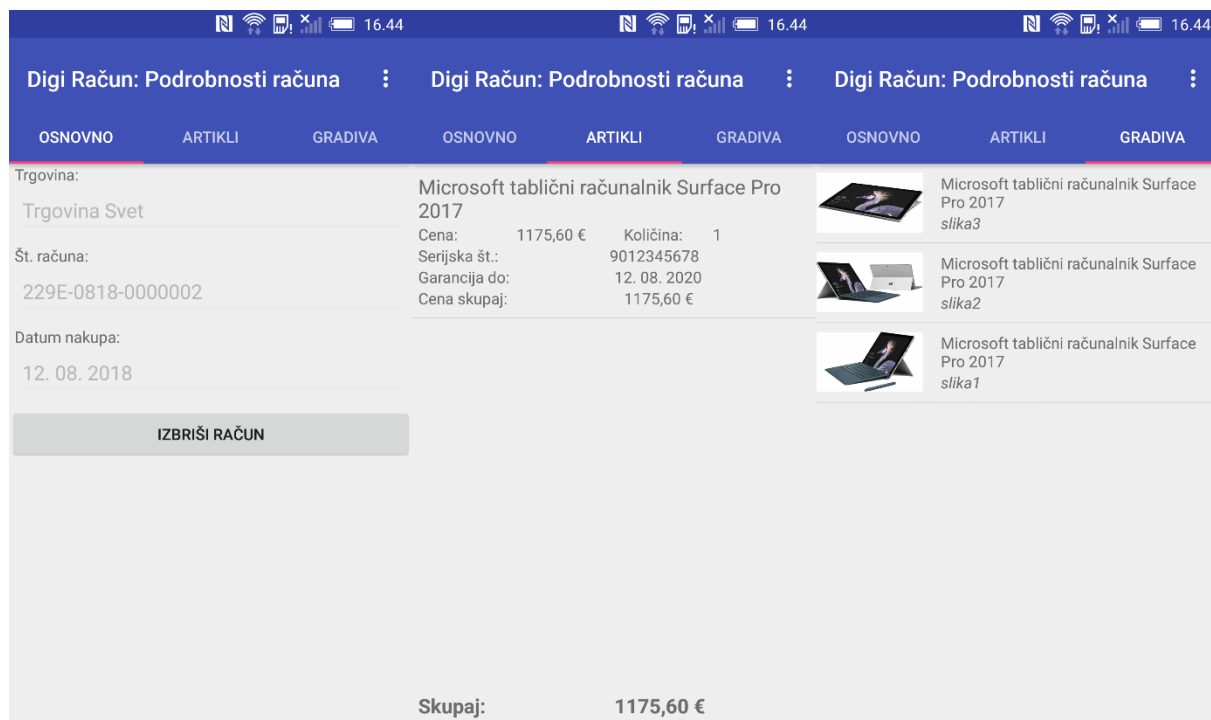
Slika 5.23: Dodajanje računa prek kode v mobilni aplikaciji IS.

Račun ustvarjen znotraj tega informacijskega sistema je mogoče dodati prek kode (Slika 5.23). Uporabnik lahko z mobilno aplikacijo prebere kodo QR v kateri je shranjena koda računa. Po uspešnem branju kode QR se odpre naslednja aktivnost z že vneseno kodo računa. V kolikor želi uporabnik ročno vpisati kodo se odpre ista aktivnost, le da je polje za vnos kode prazno. Koda računa je edini in hkrati obvezen podatek za dodajanje računa prek kode.



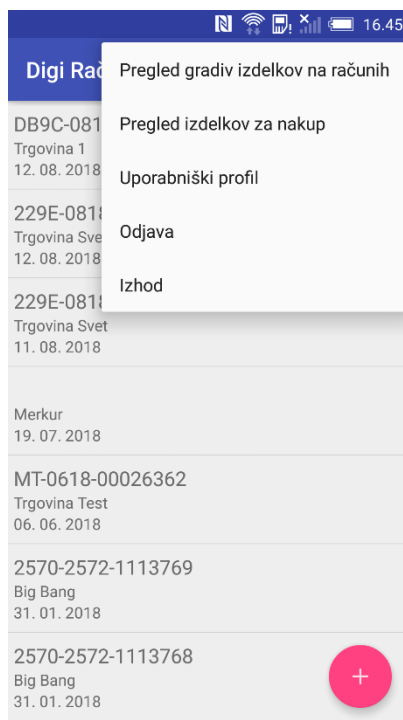
Slika 5.24: Pregled in urejanje podrobnosti računa dodanega prek slike v mobilni aplikaciji IS.

Osnovni podatki računa dodanega prek slike so prikazani na prvem zavihku (Slika 5.24). Slika računa pa je prikazana na drugem zavihku aktivnosti. Podatke računa dodanega prek slike je mogoče urejati. Ime trgovine in datum nakupa sta obvezna podatka, tako kot pri dodajanju računa. Račun je mogoče tudi izbrisati. Uporabnik lahko sliko računa odpre tudi v izvorni velikosti. V tem primeru se prikaže dialog za odpiranje slike v eni izmed nameščenih aplikacij za ogled slik. Slika polne velikosti se nato odpre v izbrani aplikaciji.



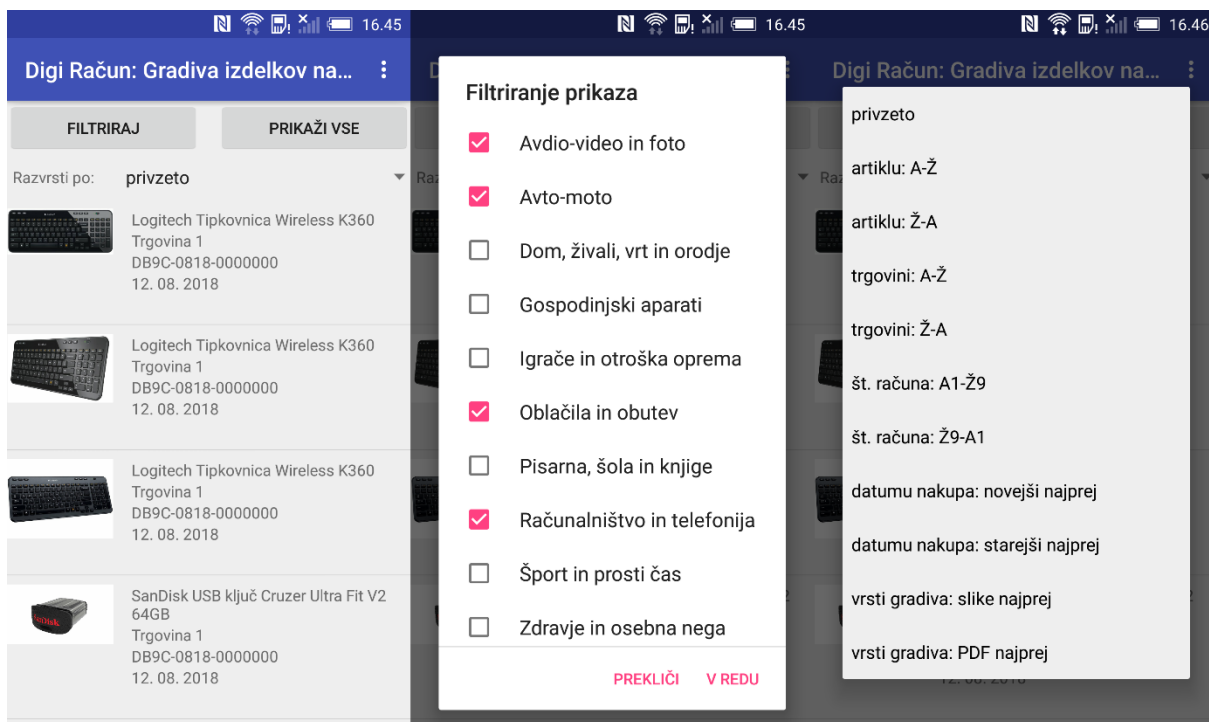
Slika 5.25: Pregled podrobnosti računa dodanega prek kode v mobilni aplikaciji IS.

Aktivnost za pregled računa dodanega prek kode vsebuje tri zavihke (Slika 5.25), enega več kot tista za pregled računa dodanega prek slike. Na prvem zavihku so prikazani osnovni podatki o računu, kot so ime trgovine, št. računa in datum nakupa. Podatkov na računu dodanem prek kode ni mogoče urejati. Naslednji zavihek prikazuje izdelke na računu. Za vsak izdelek so navedeni njegovo ime, cena, količina, morebitna serijska številka, morebiten datum izteka garancije in cena glede na količino izdelkov. Na dnu zavihka se nahaja tudi skupna cena vseh izdelkov na računu. Zadnji zavihek pa prikazuje dodatna gradiva o vseh izdelkih na računu. Omogočeno je tudi odpiranje dodatnega gradiva. Glede na vrsto datoteke (slika ali dokument PDF) je uporabniku ponujen dialog z ustreznimi aplikacijami, ki odpirajo tako vrsto datoteke.



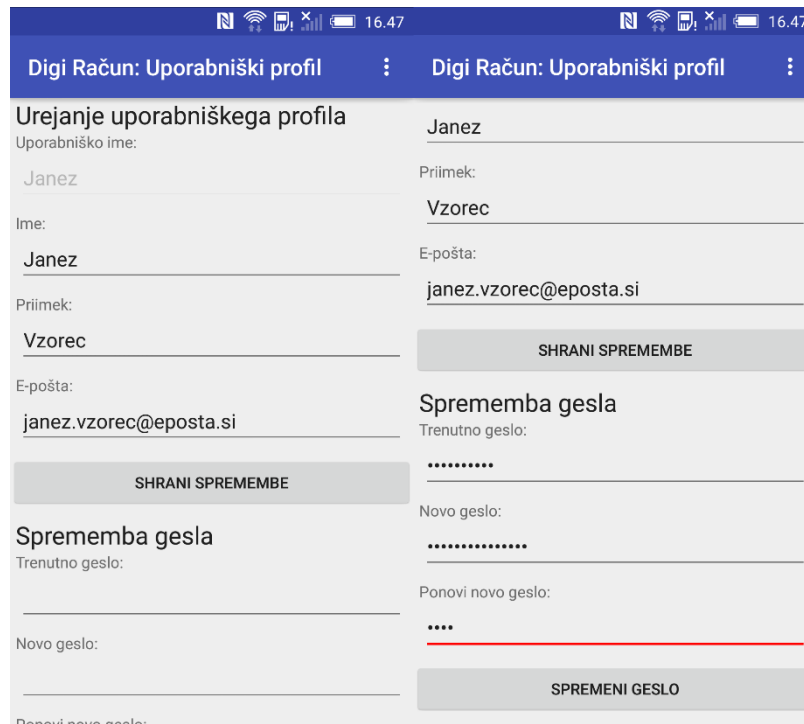
Slika 5.26: Meni v mobilni aplikaciji IS.

Mobilna aplikacija vsebuje tudi menije (Slika 5.26). Meni je na voljo v desnem zgornjem kotu. Prek menija so med drugim dostopne nekatere aktivnosti, ki sicer nimajo direktne povezave s prikazano vsebino trenutne aktivnosti. Prikaz možnosti v meniju je odvisen od trenutne aktivnosti.



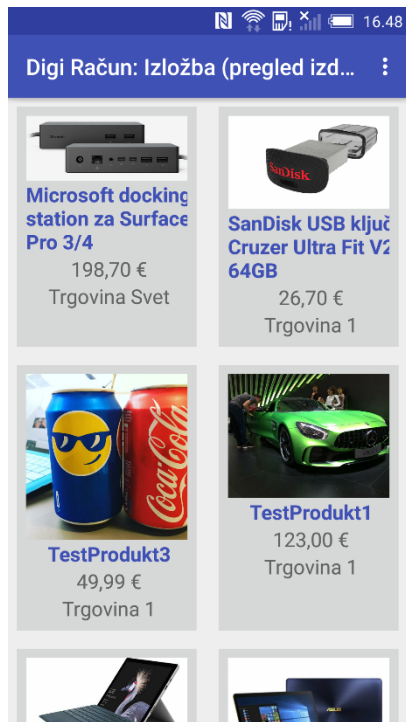
Slika 5.27: Prikaz dodatnih gradiv izdelkov z vseh računov v mobilni aplikaciji IS.

Tudi v mobilni aplikaciji je mogoč pregled vseh dodatnih gradiv izdelkov z vseh računov (Slika 5.27). Poleg predogleda dodatnega gradiva so zapisani ime izdelka, ki mu pripada, v kateri trgovini je bil izdelek kupljen, št. računa in datum nakupa. Dodatna gradiva je mogoče filtrirati po kategorijah izdelkov. Omogočeno je razvrščanje prikaza po imenu izdelka, imenu trgovine, številki računa, datumu nakupa in vrsti datoteke (slika ali dokument PDF). Dodatno gradivo si je mogoče ogledati v ustrezni aplikaciji glede na vrsto datoteke.



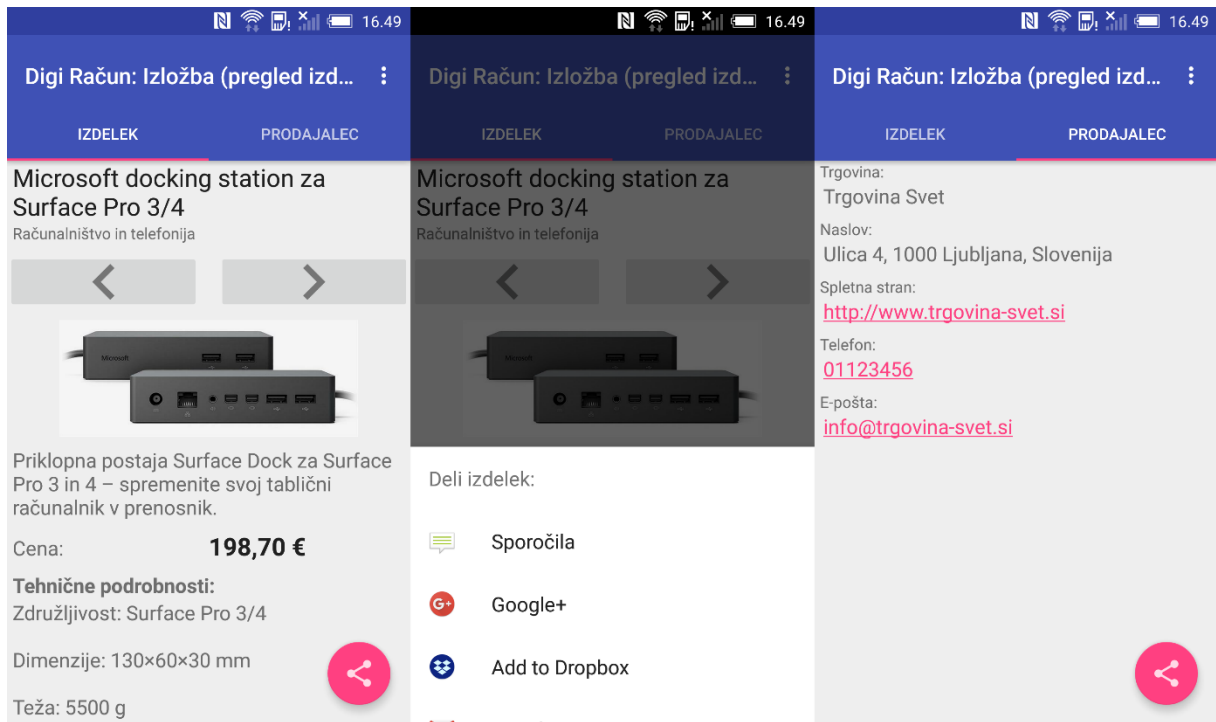
Slika 5.28: Urejanje uporabniškega profila in sprememba gesla v mobilni aplikaciji IS.

Uporabnik lahko podatke o svojem uporabniškem profilu ureja tudi v mobilni aplikaciji (Slika 5.28). Tako kot v spletni aplikaciji, tudi tukaj ni mogoče spreminjati uporabniškega imena. Spreminjati je mogoče ime, priimek in e-poštni naslov. Od teh podatkov je obvezen le e-poštni naslov, ki mora biti pravilne oblike. Uporabniku je omogočeno tudi spreminjanje gesla. Za uspešno spremembo mora vnesti pravilno trenutno geslo, novo geslo in ponovitev novega gesla. Novo geslo ne sme biti krajše od 8 znakov, ujemati pa se mora s svojo ponovitvijo.



Slika 5.29: Pregled izdelkov za nakup v mobilni aplikaciji IS.

Prijavljenim in neprijavljenim uporabnikom je znotraj mobilne aplikacije na voljo pregled izdelkov za nakup (Slika 5.29). Prikazani so vsi izdelki, ki jih imajo trenutno v ponudbi trgovine znotraj tega informacijskega sistema.



Slika 5.30: Pregled podatkov o izdelku iz ponudbe v mobilni aplikaciji IS.

Ob izbiri enega izmed izdelkov iz ponudbe, se odpre nova aktivnost s podrobnostmi izdelka (Slika 5.30). Na prvem zavihku so prikazani celotno ime izdelka, kategorija, slike, opis, cena in tehnične podrobnosti. Na drugem zavihku pa so navedeni podatki o prodajalcu. Uporabniku so prikazani ime trgovine, naslov, spletna stran, telefon in e-poštni naslov. Izdelek je mogoče tudi deliti prek aplikacij, ki omogočajo deljenje besedila. Pri deljenju izdelka se namreč ustvari sporočilo z imenom izdelka in povezavo URL, ki odpre podrobnosti izdelka v spletni aplikaciji tega informacijskega sistema (Slika 5.19). Za hitro deljenje izdelka je v spodnjem desnem kotu aktivnosti prikazan plavajoči akcijski gumb.

6. Sklepne ugotovitve

V sklopu te diplomske naloge je nastal prototip informacijskega sistema za digitalno shranjevanje računov. Uporabnikom omogoča hranjenje klasičnih papirnatih računov in računov ustvarjenih znotraj tega informacijskega sistema. Ponuja tudi pošiljanje e-poštnih obvestil en mesec pred iztekom garancije na posameznem računu ali izdelku na računu. Trgovinam pa omogoča oblikovanje ponudbe izdelkov in izdajanje računov.

Na eni strani informacijski sistem sestavljajo podatkovna baza, spletna aplikacija, dva spletna servisa in konzolna aplikacija. Vsi delujejo na storitveni platformi Microsoft Azure. Na drugi strani pa je razvita mobilna aplikacija za operacijski sistem Android. Lokalno hranjenje podatkov informacijskega sistema na posamezni napravi ni mogoče, saj so vsi podatki shranjeni v podatkovni bazi. S tem je uporabnikom zagotovljen dostop do podatkov ne glede na uporabljeno napravo. Mobilna aplikacija do podatkov informacijskega sistema namreč dostopa prek enega izmed spletnih servisov. Drug spletni servis pa služi pošiljanju e-poštnih obvestil pred iztekom garancije. Proženje slednjega izvaja konzolna aplikacija z nastavljenim urnikom delovanja znotraj nastavitev spletne aplikacije na platformi Azure. Spletna aplikacija omogoča dostop do informacijskega sistema prek spletnega brskalnika in zatorej deluje na vsaki napravi, ki ima nameščen spletni brskalnik in povezavo do spleta.

Spletna aplikacija podpira funkcionalnosti za vse vrste uporabnikov. Administrator lahko upravlja z uporabniki, dodaja nove in upravlja z obstoječimi trgovinami ter dodaja nove. Trgovinam lahko dodeljuje trgovce, ki izdajajo račune. Ti lahko poleg ustvarjanja novih pregledujejo ali celo brišejo svoje v preteklosti izdane račune. Trgovci administratorji lahko v ponudbo trgovine dodajajo nove izdelke in urejajo že dodane. Pravico imajo videti vse izdane račune znotraj svoje trgovine in ne le lastnih, kot trgovci. Običajni uporabniki lahko shranjujejo klasične papirnate račune s katerekoli trgovine, ki ne uporablja tega informacijskega sistema. Dodajo pa lahko tudi račune ustvarjene znotraj tega informacijskega sistema. Na voljo jim je tudi pregled dodatnih gradiv o kupljenih izdelkih, kot so slike in dokumenti PDF (npr. garancijski list ali navodila za uporabo). Ne glede na vrsto uporabnika pa lahko vse zmožnosti, ki so na voljo običajnemu uporabniku, koristijo vsi uporabniki z obstoječim uporabniškim profilom. Mobilna aplikacija je namenjena uporabnikom pametnih telefonov z operacijskim sistemom Android. Podpira vse funkcionalnosti, ki jih spletna aplikacija nudi običajnim uporabnikom in jih nadgrajuje. Podpira namreč stalno prijavo, da je dodajanje računov in dostop do obstoječih še hitrejši. Omogoča pa tudi dodajanje računov ustvarjenih znotraj tega informacijskega sistema z branjem kode QR. Tako spletna, kot tudi mobilna aplikacija ponujata pregled izdelkov iz ponudbe trgovin, ki uporabljajo ta informacijski sistem. Pregled ponudbe je omogočen tudi neregistriranim uporabnikom.

V kolikor bi se informacijski sistem uporabljal v resničnem življenju in bi v njem shranjeni računi imeli veljavo, bi uporabnikom olajšal upravljanje z računi. Uporabniki bi namreč imeli pomembne račune varno shranjene na enem mestu. Klasične papirnate račune se namreč lažje izgubi, ali pa ti zbledijo do neberljivosti. Poleg tega bi bili uporabniki tudi pravočasno obveščeni pred iztekom garancije na nekem računu. Če pa bi ga za izdajanje računov uporabljale še trgovine, bi se sčasoma lahko povsem izognili tiskanju računov. Po nekaj omembah ideje, se ta namreč zdi zanimiva potencialnim uporabnikom oz. potrošnikom.

6.1. Ideje za nadaljnji razvoj

Razvit prototip informacijskega sistema služi za prikaz ideje delovanja, zato bi bilo nekatere funkcionalnosti mogoče še izboljšati ali pa razviti dodatne. Uporabnikom bi lahko bile na voljo nastavitve, kjer bi si določili koliko časa pred iztekom želijo prejeti obvestilo o izteku garancije za posamezen račun ali izdelek. Nastavili bi si lahko tudi katera stran se prikaže ob prijavi v spletno aplikacijo. Trgovci administratorji so ob prijavi namreč preusmerjeni na stran za upravljanje izdelkov, mogoče pa bi si želeli najprej videti stran za upravljanje računov. Sčasoma bo trgovina imela zelo veliko izdanih računov, ki se bodo še vedno prikazovali v sistemu. Zato bi lahko bilo omogočeno arhiviranje starih računov. Zaradi velike količine podatkov bi informacijski sistem lahko imel vgrajen iskalnik. Administratorji bi tako lahko iskali po uporabnikih in trgovinah, trgovci in uporabniki pa po računih in izdelkih. Da bi se uporaba informacijskega sistema kar čim bolj razširila bi bilo smiselno uvesti povezavo z obstoječimi sistemi za izdajanje računov. Od trgovin namreč ni smotno pričakovati, da bodo zamenjali obstoječe sisteme za izdajanje računov. Povezava z njimi pa bi uporabnikom omogočala dodajanje digitalne oblike računov iz drugih sistemov prek vpisane kode ali prebrane kode QR.

Vsa izvorna koda razvitega prototipa informacijskega sistema je dostopna na repozitoriju GitLab:

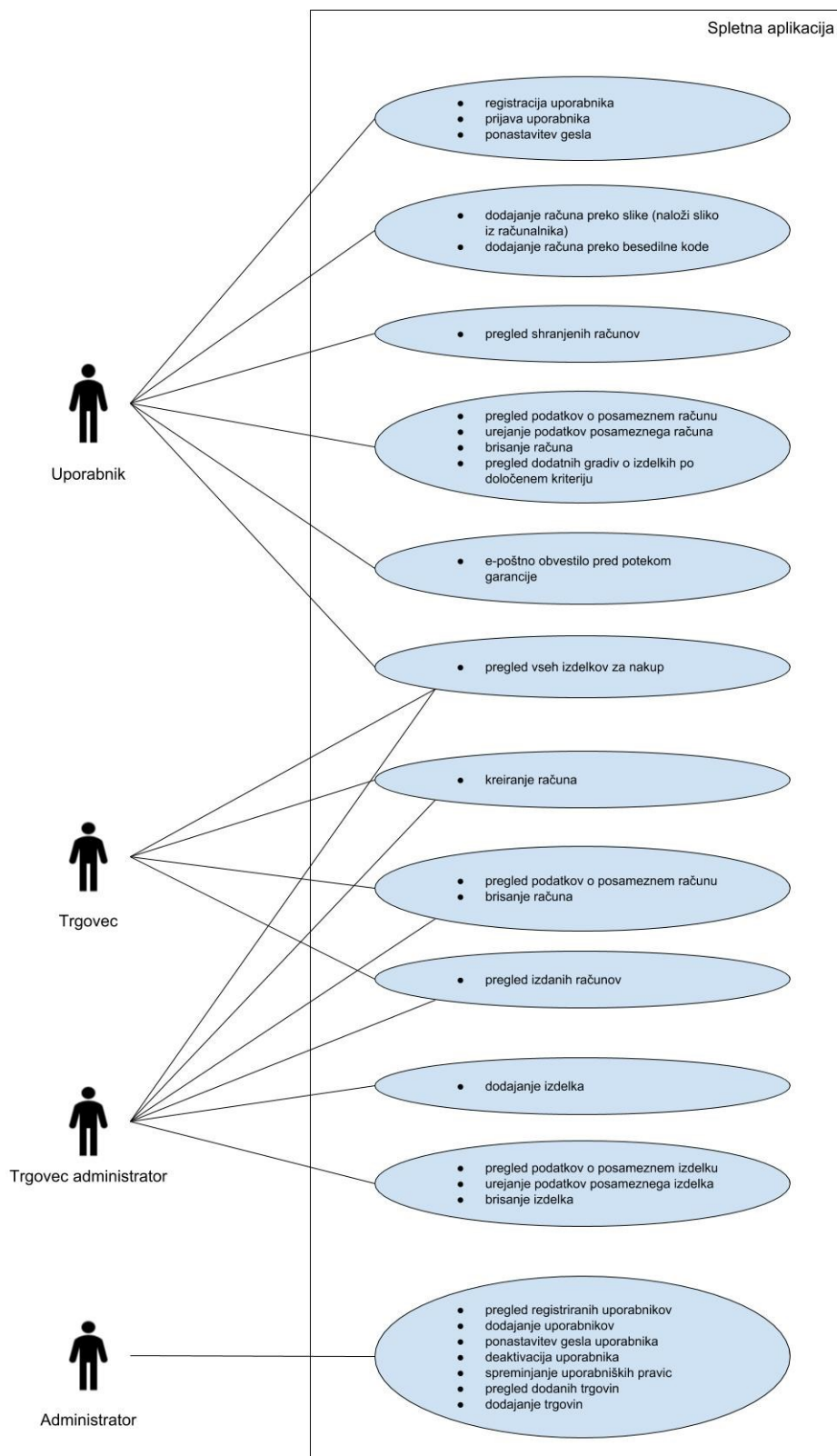
- spletna aplikacija, oba spletna servisa in podatkovna baza: <https://gitlab.com/matejv/IS-DigiRacun>,
- konzolna aplikacija: <https://gitlab.com/matejv/ServiceCaller-DigiRacun>,
- mobilna aplikacija: <https://gitlab.com/matejv/AndroidApp-DigiRacun>.

Literatura

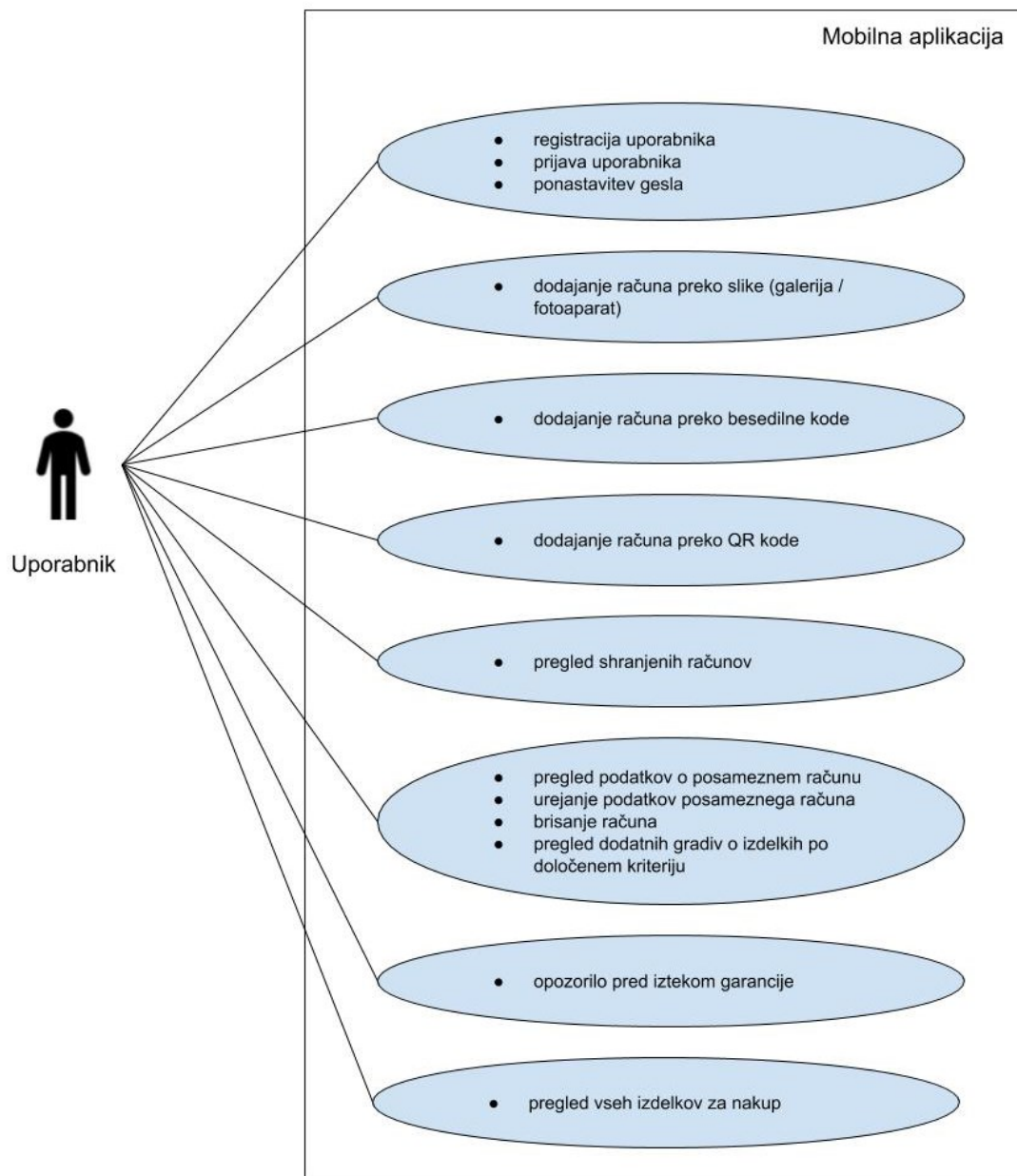
- [1] Android (operating system) - Wikipedia. Dostopno na: [https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system)). [Dostopano: 25. 7. 2018].
- [2] Android Oreo - Wikipedia. Dostopno na: https://en.wikipedia.org/wiki/Android_Oreo. [Dostopano: 25. 7. 2018].
- [3] Android Studio - Wikipedia. Dostopno na: https://en.wikipedia.org/wiki/Android_Studio. [Dostopano: 26. 7. 2018].
- [4] Android Studio features | Android Developers. Dostopno na: <https://developer.android.com/studio/features/>. [Dostopano: 26. 7. 2018].
- [5] Automatic tuning | Microsoft Docs. Dostopno na: <https://docs.microsoft.com/en-us/sql/relational-databases/automatic-tuning/automatic-tuning>. [Dostopano: 30. 7. 2018].
- [6] Azure SQL Database purchasing models | Microsoft Docs. Dostopno na: <https://docs.microsoft.com/sl-si/azure/sql-database/sql-database-service-tiers#what-are-database-transaction-units-dtus>. [Dostopano: 30. 7. 2018].
- [7] Data types (Transact-SQL) | Microsoft Docs. Dostopno na: <https://docs.microsoft.com/en-us/sql/t-sql/data-types/data-types-transact-sql>. [Dostopano: 28. 7. 2018].
- [8] Distribution dashboard | Android Developers. Dostopno na: <https://developer.android.com/about/dashboards/>. [Dostopano: 25. 7. 2018].
- [9] Gradle - Wikipedia. Dostopno na: <https://en.wikipedia.org/wiki/Gradle>. [Dostopano: 26. 7. 2018].
- [10] Microsoft Azure - Wikipedia,“ [Elektronski]. Dostopno na: https://en.wikipedia.org/wiki/Microsoft_Azure. [Dostopano: 19. 7. 2018].
- [11] Microsoft Visual Studio - Wikipedia. Dostopno na: https://en.wikipedia.org/wiki/Microsoft_Visual_Studio. [Dostopano: 24. 7. 2018].
- [12] Multiple Active Result Sets (MARS) | Microsoft Docs. Dostopno na: <https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/sql/multiple-active-result-sets-mars>. [Dostopano: 3. 8. 2018].
- [13] Software development kit - Wikipedia. Dostopno na: https://en.wikipedia.org/wiki/Software_development_kit. [Dostopano: 25. 7. 2018].

- [14] SqlCommand Class (System.Data.SqlClient). Dostopno na:
<https://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlcommand>.
[Dostopano: 7. 8. 2018].
- [15] SqlCommand.ExecuteNonQuery Method (System.Data.SqlClient). Dostopno na:
<https://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlcommand.executenonquery>. [Dostopano: 7. 8. 2018].
- [16] SqlCommand.ExecuteReader Method (System.Data.SqlClient). Dostopno na:
<https://msdn.microsoft.com/en-us/library/9kcbe65k>. [Dostopano: 7. 8. 2018].
- [17] SqlCommand.ExecuteScalar Method (System.Data.SqlClient). Dostopno na:
<https://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlcommand.executescalar>. [Dostopano: 7. 8. 2018].
- [18] SqlDataSource Web Server Control Overview. Dostopno na:
<https://msdn.microsoft.com/en-us/library/dz12d98w.aspx>. [Dostopano: 6. 8. 2018].
- [19] Visual Studio Development Features | Visual Studio. Dostopno na:
<https://visualstudio.microsoft.com/vs/features/>. [Dostopano: 24. 7. 2018].

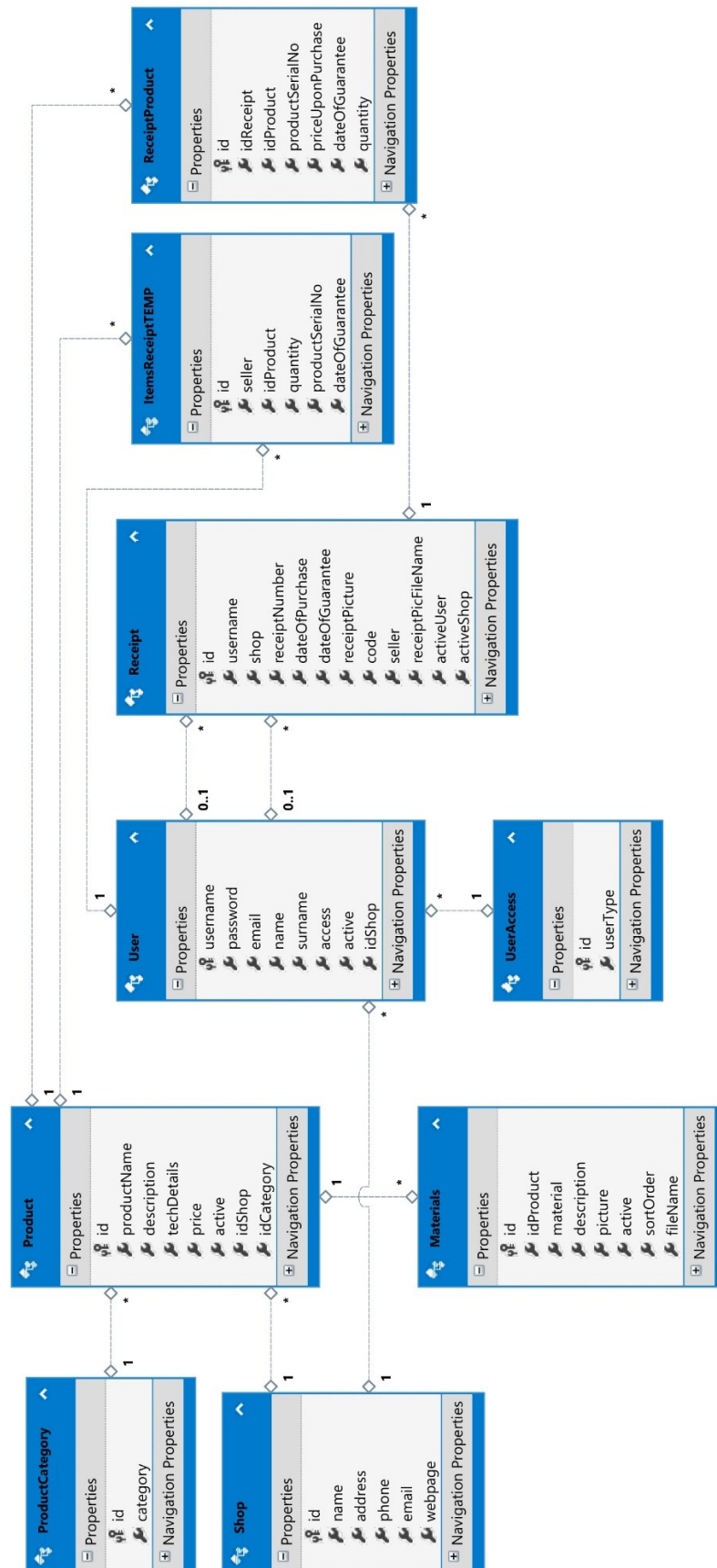
Priloge



Priloga 1: Diagram primerov uporabe za spletno aplikacijo.



Priloga 2: Diagram primerov uporabe za mobilno aplikacijo.



Priloga 3: Entitetno-relacijski diagram podatkovne baze.