

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Klavdij Oberstar

**Stroškovni vidik optimizacije mobilne
uporabniške izkušnje**

MAGISTRSKO DELO
ŠTUDIJSKI PROGRAM DRUGE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Dejan Lavbič

Ljubljana, 2014

Rezultati magistrskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov magistrskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

IZJAVA O AVTORSTVU MAGISTRSKEGA DELA

Spodaj podpisani Klavdij Oberstar, z vpisno številko **63080118**, sem avtor magistrskega dela z naslovom:

Stroškovni vidik optimizacije mobilne uporabniške izkušnje

S svojim podpisom zagotavljam, da:

- sem magistrsko delo izdelal samostojno pod mentorstvom doc. dr. Dejana Lavbiča,
- so elektronska oblika magistrskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko magistrskega dela,
- soglašam z javno objavo elektronske oblike magistrskega dela v zbirki "Dela FRI".

V Ljubljani, 14. junij 2014

Podpis avtorja:

*Zahvaljujem se mentorju doc. dr. Dejanu Lavbiču za vse nasvete in pomoč.
Zahvalil bi se tudi vsem, ki so mi na tak ali drugačen način pomagali pri
izdelavi magistrskega dela.*

Kazalo

Povzetek

Abstract

1	Uvod	1
1.1	Motivacija	1
1.2	Cilji	3
1.3	Nadaljnja vsebina	3
2	Metode za analizo podatkov	5
2.1	Odkrivanje znanja iz podatkov	5
2.2	Proces odkrivanja znanja iz podatkov	6
2.3	Uporabniški profili	10
2.4	Etika in varstvo podatkov	12
3	Pregled uporabljenih tehnologij	15
3.1	MongoDB	15
3.2	Java EE, Tomcat in RESTful	16
3.3	Highcharts in jQuery	17
3.4	Android SDK	18
3.5	JSON	19
4	Sorodne rešitve na področju spremljanja in porabe mobilnih storitev	21

5	Predlog rešitve optimizacije stroškovnega vidika mobilne uporabniške izkušnje	25
6	Opis inteligentnega sistema	29
6.1	Podatkovna baza	29
6.2	Nastavitvene datoteke	32
6.3	Strežniška aplikacija	32
6.4	Mobilna aplikacija	46
6.5	Generator podatkov	48
7	Podrobnosti implementacije prototipa inteligentnega sistema	49
7.1	Podatkovna baza	49
7.2	Strežniška aplikacija	50
7.3	Mobilna aplikacija	53
8	Evalvacija predlagane rešitve	55
8.1	Prva faza - splošno testiranje	56
8.2	Druga faza - testiranje učinkovitosti	57
8.3	Tretja faza - SWOT analiza	62
9	Zaključek	67

Seznam uporabljenih kratic

kratica	angleško	slovensko
HTML	hypertext markup language	označevalni jezik za oblikovanje večpredstavnostnih dokumentov
HTTP	hypertext transfer protocol	protokol za prenos večpredstavnostnih dokumentov
JAX-RS	Java API for RESTful web services	Javanski dostop do REST spletnih storitev
JSON	JavaScript object notation	JavaScript objektna notacija
JSP	JavaServer pages	JavaServer strani
MD5	message-digest algorithm 5	kriptografska zgoščevalna funkcija
MMS	multimedia messaging service	storitev za multimedijsko sporočanje
NoSQL	not only structured query language	alternativna rešitev relacijskim podatkovnim bazam
PHP	hypertext preprocessor	skriptni programski jezik
REST	representational state transfer	arhitekturni stil načrtovanja spletnih storitev
SDK	software development kit	orodja za razvijanje aplikacij
SMS	short message service	storitev za kratka sporočila
SWOT	strengths, weaknesses, opportunities, and threats	prednosti, slabosti, priložnosti in nevarnosti
URI	uniform resource identifier	enotni identifikator vira
XML	extensible markup language	razširljivi označevalni jezik

Povzetek

Mesečni strošek uporabe mobilnih storitev lahko hitro preseže načrtovano mejo. Zato smo si zastavili cilj, izdelati pametnega mobilnega pomočnika. Pametni mobilni pomočnik bi uporabniku svetoval, kako in na kakšen način naj uporablja mobilne storitve, da bo mesečni strošek uporabe najnižji. Implementirali smo prototip, ki s pomočjo metod podatkovnega rudarjenja in izdelave uporabniškega profila, uporabniku prikazuje porabo in statistiko ter svetuje o boljši izbiri mobilnih storitev glede na njegov način porabe mobilnih enot. Pametni mobilni pomočnik ponuja nasvete o cenejši izbiri enot v tujini, najprimernejši storitvi, najboljšemu obračunskemu intervalu itd. Uporabnik lahko s pomočjo pametnega mobilnega pomočnika optimalno izkoristi ponujene mobilne storitve in s tem zniža mesečni strošek. Konkurenčnost pametnega mobilnega pomočnika smo preverili s pomočjo analize SWOT. Ugotovili smo, da ima prednost pred ostalimi rešitvami v naprednih funkcionalnostih za boljšo izrabo zakupljenih enot.

Ključne besede: pametni mobilni pomočnik, mobilne storitve, optimizacija cene, pametno svetovanje, podatkovno rudarjenje, uporabniški profili

Abstract

The monthly cost of mobile services can easily exceed the designated limit. Therefore, we set a goal to create smart mobile assistant. Smart mobile assistant would advise user how and in what manner he should use mobile services, so he will get the lowest monthly cost. We have implemented a prototype, which use methods of data mining and user profiling so that can show user consumption, statistics and advises on making better use of mobile services, according to his way of consumption of mobile units. Smart mobile assistant offers advice on the use of cheaper abroad consumption of mobile units, the most appropriate mobile service, best accounting interval, etc. User can, with use of smart mobile assistant, optimally use mobile services and thus reduce the monthly cost. Competitiveness of smart mobile assistant was tested using SWOT analysis. We have found that it has an advantage over other solutions in the advanced functionalities for a better use of leased units.

Key words: smart mobile assistant, mobile services, price optimization, smart consulting, data mining, user profiles

Poglavje 1

Uvod

V današnjem času so pametni mobilniki postali nekaj povsem vsakdanjega. Te naprave uporabljamo tako za brskanje po spletu, branje pošte, iskanje informacij in za vse ostale stvari, ki nam olajšajo vsakdanja opravila. Še vedno pa je njihova primarna funkcionalnost izvajanje telefonskih klicev, pošiljanje SMS/MMS¹ sporočil in posredno uporaba internetne povezave. Pogosto se zgodi, da uporabnik porabi vse zakupljene količine enot in tako mesečni račun naraste za več deset evrov. Naša osrednja motivacija v okviru magistrskega dela je bila, uporabnikom olajšati nadzor nad porabljenimi sredstvi in kar se da pametno svetovati glede izrabe mobilnih storitev. S tem pa uporabniku pomagati pri zmanjšanju porabljenih sredstev.

1.1 Motivacija

Na tržišču obstaja veliko mobilnih aplikacij, ki omogočajo spremljanje porabe mobilnih storitev. Takšne mobilne aplikacije so dodol Phone [17], Stats Free [24], DroidStats [22], My Data Manager [26], Call Meter NG [14], Call Stats [16] in še mnoge druge. Podrobnejši opis mobilnih aplikacij za spremljanje porabe, sledi v enem od nadaljnjih poglavij. Mobilne aplikacije za spremljanje porabe mobilnih enot večinoma ponujajo zgolj osnovne funkci-

¹Storitev za kratka sporočila (SMS) oziroma multimedijsko sporočanje (MMS).

onalnosti, kot so: beleženje klicev, štetje poslanih ter prejetih SMS/MMS sporočil, beleženje prenosa podatkov, prikaz enostavne statistike, nastavljanje omejitev, ter nastavljanje različnih načinov zaračunavanja. Uporabnik lahko z njihovo pomočjo spremlja in beleži porabo mobilnih enot.

Dodana vrednost teh mobilnih aplikacij je opozarjanje na presežek porabljenih enot, prikaz preostalih enot in spremljanje sprememb porabe enot skozi zgodovino. Uporabnik lahko s pomočjo teh informacij prepreči preveliko porabo mobilnih enot in s tem omeji mesečni strošek na zastavljenega. Tu se postavi vprašanje, ali lahko s pomočjo določenih metod avtomatizirano izluščimo na prvi pogled skrita pravila, iz katerih lahko pridobimo pomembne informacije. Informacije s katerimi bi lahko uporabniku svetovali o boljši izrabi enot, nižjem mesečnem strošku, napotkih o načinu porabe mobilnih storitev ipd. Obstoječe rešitve takšnih funkcionalnosti ne ponujajo. Pridobivanje takšnih informacij je prepuščeno uporabniku.

V računalništvu obstaja panoga, ki se imenuje odkrivanje znanja iz podatkov. Osredotoča se na pridobivanje informacij iz velike količine podatkov. Podatki so dejstva o nekem objektu, dogodku ali dejavnosti, ki nimajo nekega pomena. Informacijo pridobimo iz podatkov in prejemu pove nekaj novega. Prejemnik s pomočjo prejetih informacij, ki jih poveže s svojimi izkušnjami pridobi znanja [7]. Odkrivanje znanja iz podatkov je kompleksen proces, ki s pomočjo različnih metod in pristopov iz podatkov izlušči uporabne informacije [6]. Proces je sestavljen iz več različnih korakov. Ključen korak pri celotnem procesu je podatkovno rudarjenje. To je korak s katerim dobimo iz podatkov uporabne informacije. Ostali koraki se bolj osredotočajo na pripravo podatkov, izbiro pristopov in posredovanje informacij. Korak podatkovnega rudarjenja uporablja različne metode za procesiranje podatkov [4]. Takšne metode bi lahko uporabili pri realizaciji našega problema.

1.2 Cilji

Zastavili smo si cilj, izdelati prototip pametnega mobilnega pomočnika, kateri uporabniku prikazuje porabo in statistiko ter mu poleg tega svetuje na kakšen način lahko zmanjša mesečni strošek pri uporabi mobilnih storitev. Celoten proces pridobivanja informacij je avtomatiziran. Uporabnik dobi različne informacije, s katerimi si lahko pomaga pri zmanjšanju mesečnega stroška oziroma bolje izrabi ponujene mobilne storitve.

Seveda pa se tu postavijo vprašanja, ali bo takšen pametni mobilni pomočnik, pravilno svetoval uporabniku in bo uporabnik prejel dovolj koristne informacije, da mu bodo pomagale pri zmanjšanju mesečnega stroška. Ali bo uporabnik znal razbrati ponujene informacije za boljšo izrabo mobilnih storitev? Prav tako se pojavi dvom o tem, ali bo takšen mobilni pomočnik konkurenčen ostalim sorodnim rešitvam? Ima takšen pametni mobilni pomočnik dovolj veliko dodano vrednost, da je njegova uporaba smiselna? Odgovore na postavljena vprašanja bomo poskušali podati v nadaljnjih poglavjih magistrskega dela.

1.3 Nadaljnja vsebina

V drugem poglavju opišemo teoretično ozadje odkrivanja znanja iz podatkov s poudarkom na metodah podatkovnega rudarjenja. V naslednjem poglavju se osredotočimo na pregled tehnologij, ki jih bomo uporabili pri izdelavi prototipa. V četrtem poglavju so bolj podrobno opisane sorodne rešitve, ki so na voljo na tržišču. V poglavju z naslovom predlog rešitve optimizacije stroškovnega vidika mobilne uporabniške izkušnje predstavimo koncept naše rešitve. Opis rešitve in delovanje posameznih delov je opisano v šestem poglavju. V sedmem poglavju predstavimo implementacijo inteligentnega sistema. V osmem poglavju je opisna evalvacija inteligentnega sistema. V zaključnem poglavju je povzetek celotnega dela in podane so možnosti nadaljnjega dela.

Poglavje 2

Metode za analizo podatkov

V tem poglavju si bomo pogledali teoretično ozadje pristopov, metod in načinov analize velike količine podatkov. Pregledali bomo kakšne metode se uporabljajo za zbiranje in obdelavo podatkov ter za pridobivanje relevantnih informacij. Seznanili se bomo tudi z uporabniškimi profili in različnimi pristopi izdelave. Poleg tega pa bomo pregledali praktične primere, pri katerih se uporabljajo določeni pristopi analize podatkov.

Ustvarili bomo teoretično podlago za lažje razumevanje nadaljnjih poglavij. Ključni del predloga naše rešitve temelji na idejah in konceptih, ki so opisani v nadaljevanju. Teoretično področje pristopov analize podatkov je zelo obsežno, zato se bomo osredotočili le na najpomembnejše in tiste dele, ki so posredno ali neposredno povezani s predlagano rešitvijo.

2.1 Odkrivanje znanja iz podatkov

V zadnjem času so se zelo povečale zmogljivosti tako generiranja kot zbiranja podatkov. Podatki se zbirajo oziroma generirajo na različne načine. Največji doprinos k temu je imela uporaba različnih elektronskih naprav. Elektronske naprave zbirajo vse vrste podatkov tako posameznikov kot tudi podjetij in ustanov. Za shranjevanje podatkov se uporablja na milijone različnih podatkovnih baz. Količine podatkov v sodobnih podatkovnih bazah pa se

povečujejo zelo hitro. Ta velikanska količina podatkov je ustvarila potrebo po novih tehnikah in orodjih, katera lahko inteligentno in avtomatizirano procesirajo podatke v uporabne informacije in znanje.

Odkrivanje znanja iz podatkov je proces odkrivanja nepoznanih in potencialno uporabnih informacij iz podatkov ali podatkovnih baz. S postopkom odkrivanja znanja iz podatkov izluščimo znanje, pravilnosti ali skrite informacije. Podatkovne baze lahko služijo kot bogat in zanesljiv vir informacij. Odkrita znanja in informacije se lahko uporabijo v različnih nadaljnjih procesih. Zato se proces odkrivanja znanja iz podatkov uporablja pri različnih panogah, kot je strojno učenje, umetna inteligenca, statistika, vizualizacija podatkov in drugih [4].

2.2 Proces odkrivanja znanja iz podatkov

Odkrivanje znanja iz podatkov je kompleksen proces identifikacije veljavnih, uporabnih ter predvsem razumljivih informacij. Proces odkrivanja znanja iz podatkov je interaktiven in ponavljajoč, ter sestavljen iz več korakov. Koraki večkrat vključujejo tudi veliko odločitev s strani uporabnika. V nadaljevanju bomo pregledali posamezne korake procesa.

Prvi korak je namenjen seznanjanju s problemom, ki ga želimo rešiti in odkrivanju relevantnih informacij. Predvsem pa določitvi cilja procesa odkrivanja znanj iz podatkov s stališča končnega uporabnika. V drugem koraku določimo ciljno skupino podatkov. Lahko izberemo skupino podatkov ali pa se osredotočimo na vzorčno podskupino podatkov, med katerimi bomo odkrivali znanje. Filtriranje podatkov in predprocesiranje se zgodi v tretjem koraku. Osnovne operacije vsebujejo: odstranjevanje šuma (če je potrebno), zbiranje potrebnih informacij za modeliranje šuma, odločanje o strategijah za obravnavo manjkajočih podatkov in druge operacije namenjene predpripravi podatkov. Četrty korak zreducira podatke in jih projicira. Namen koraka je poiskati lastnosti, ki predstavljajo podatke glede na zastavljen cilj. Z metodami dimenzionalne redukcije ali transformacije zmanjšamo število

primernih lastnosti.

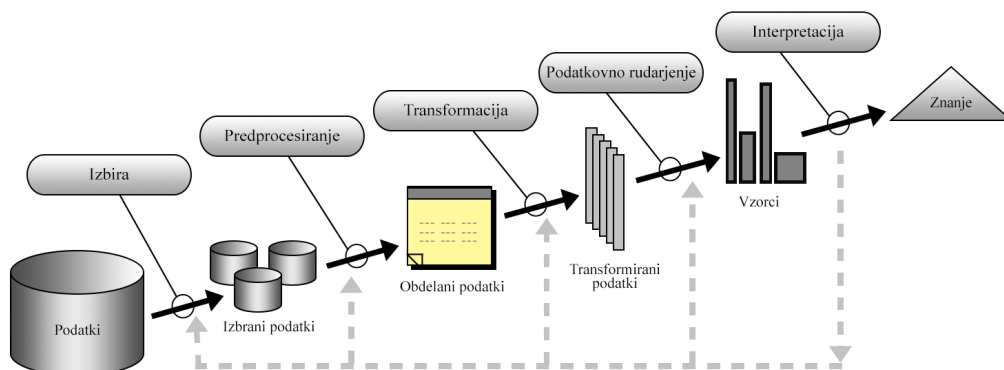
Prvi štirje koraki so namenjeni določitvi cilja in predpripravi podatkov. Nadaljnji koraki se osredotočajo na odkrivanje znanja, zato v petem koraku iščemo najbolj primerne metode podatkovnega rudarjenja glede na naš cilj. Te metode so: povzemanje, klasifikacija, regresija, strnitev in druge. Bolj podrobno jih bomo opisali v nadaljevanju. V šestem koraku izberemo metodo oziroma metode podatkovnega rudarjenja, katere bodo uporabljene za iskanje vzorcev v podatkih. Ta korak tudi vključuje odločitve o najbolj primernih parametrih metode. Sedmi korak je podatkovno rudarjenje. Podatkovno rudarjenje vključuje iskanje vzorcev interesa, kateri so v določeni predstavitveni obliki, kot so klasifikacijska drevesa, grozdi in drugih.

Ko smo izbrali metodo podatkovnega rudarjenja in poiskali vzorce, pride na vrsto pridobivanje in uporaba znanj. V osmem koraku interpretiramo pridobljene vzorce. Ta korak lahko vključuje tudi vizualizacijo zbranih vzorcev in modelov. Končni deveti korak je uporaba pridobljenega znanja oziroma informacij. Pridobljeno znanje lahko uporabimo v drugem sistemu za izvajanje nadaljnjih akcij ali pa ga preprosto le dokumentiramo. Prav tako pa v tem koraku lahko odkrivamo in razrešujemo konflikte med pridobljenim in obstoječim znanjem [6].

Proces odkrivanja znanja iz podatkov lahko vsebuje veliko iteracij, prav tako lahko vsebuje zanko med katerima koli korakoma v procesu [6]. Celoten proces z možnimi zankami je prikazan na sliki 2.1. Sedaj ko smo v grobem predstavili celoten proces odkrivanja znanj iz podatkov, se bomo v nadaljevanju osredotočili na korak podatkovnega rudarjenja. Korak podatkovnega rudarjenja je najpomembnejši korak v celotnem procesu.

2.2.1 Korak podatkovnega rudarjenja

Korak podatkovnega rudarjenja pogosto vključuje ponavljajočo uporabo izbranih metod podatkovnega rudarjenja. Rezultati odkrivanja znanj so definirani z načinom uporabe sistema. Razlikujemo dve obliki rezultatov: verifikacija in odkritje. Pri verifikaciji je sistem omejen na preverjanje hipoteze



Slika 2.1: Proces podatkovnega rudarjenja. Posamezni koraki so smiselno združeni.

uporabnika. Pri odkritju sistem avtonomno najde nove vzorce. Odkritje še dodatno razdelimo na napoved in opis. Pri napovedi sistem najde vzorce za odkrivanje lastnosti obnašanja nekaterih entitet v prihodnosti. Pri opisu pa sistem najde vzorce za predstavitev rezultatov v človeško berljivi obliki.

Podatkovno rudarjenje vključuje prilaganje modelov opazovanim podatkom ali pa določanje vzorcev iz opazovanih podatkov. Večina metod podatkovnega rudarjenja temelji na preizkušanih in testiranih tehnikah iz strojnega učenja, zaznavanja vzorcev in statistike. Algoritmi teh metod so lahko zelo kompleksni in obsežni [6].

2.2.2 Metode podatkovnega rudarjenja

Za doseganje prej omenjenih rezultatov odkrivanja znanja, uporabljamo različne metode podatkovnega rudarjenja. Metode uporabljajo različne pristope in vračajo različne rezultate [4]. Zelo pogosto se uporabi več različnih metod, saj le tako lahko dobimo zadovoljive rezultate.

Povzemanje

Povzemanje je abstrakcija ali posplošitev podatkov. Po povzemanju podatkov dobimo kot rezultat manjšo množico podatkov, ki predstavljajo splošen pregled celotnih podatkov. Kot primer lahko vzamemo čas trajanja klicev, katere združimo po dnevih, tednih, mesecih. Pri povzemanju lahko gre za različne abstraktne razrede. Prav tako pa lahko predstavi različne aspekte podatkov.

Klasifikacija

Klasifikacija je izpeljava funkcije ali modela, katera določi razred objekta glede na njegove attribute. Zbirka objektov je podana kot učna množica, v kateri je vsak objekt predstavljen z vektorjem atributov skupaj s svojim razredom. Klasifikacijska funkcija ali model je zgrajen z analizo relacij med atributi in razredi objektov v učni množici. Takšna klasifikacijska funkcija ali model se lahko potem uporabi za klasifikacijo ostalih objektov in razvijanja boljšega razumevanja razredov objektov.

Združevanje

Združevanje je odkrivanje povezanosti ali povezav med objekti. Takšno povezanost oziroma povezavi med objekti poimenujemo pravilo združevanja. Pravila združevanja prikažejo zveze med objekti. Takšna pravila se zelo uporabna v marketingu, oglaševanju in drugje.

Razvrščanje v skupine

Pri postopku razvrščanja v skupine gre za identifikacijo razredov oziroma skupin za zbirko podatkov, katerih razredi so neznani. Objekti so razvrščeni v skupine, tako da vsak objekt spada v tisto skupino objektov, s katerimi ima najbolj podobne attribute. Ko so skupine določene, se skupne lastnosti objektov uporabijo kot opis razreda.

Ostale metode

Veliko podatkov je zbranih skozi daljše časovno obdobje. Takšni podatki so lahko predstavljeni kot objekti z atributom časa. Te objekte lahko analiziramo z metodo časovne analize, pri čemer dobimo informacije, kako se vzorci spreminjajo skozi čas. Obstaja še ena takšna metoda, ki se neposredno navezuje na metodo časovne analize. Metoda odkriva večje spremembe v podatkih skozi iz prej izmerjenih vrednostih.

2.3 Uporabniški profili

Vse več aplikacij v današnjem času je uporabniško usmerjenih. Aplikacije uporabljajo osebni pristop za namen oglaševanja, priporočilnih sistemov, prilagajanje vsebine itd. Takšen pristop ponudi pomembne prednosti v primerjavi s tradicionalnimi metodami. Pri takšnih posebljenih aplikacijah pa se pojavi tehnični problem, kako zgraditi točen in celovit profil uporabnika, kateri bo zagotovil vse pomembne informacije o uporabniku in njegovem obnašanju [2].

Uporabniški profili so velikokrat zelo tesno povezani s podatkovnim rudarjenjem. Uporabniške profile so v podatkovnem rudarjenju prvič uporabili pri odkrivanju goljufij v industriji mobilne tehnologije. To so storili tako, da so zgradili pravila o tem, kako je uporabnik uporabljal mobilne storitve, s pomočjo učnega sistema. Nato pa so generirali splošne profile za različne segmente uporabnikov z namenom, da bi ugotovili pogoje goljufanja za različne skupine uporabnikov [5].

2.3.1 Izdelava uporabniškega profila

Obstajajo različni načini in pristopi za izdelavo uporabniških profilov. Prav tako pa različni načini in pristopi vračajo različne rezultate. Načini izdelave uporabniških profilov se delijo v dve večji skupini. Prva gradi profil iz podatkov in druga z medsebojnim filtriranjem. Postopki izdelave posame-

znega profila v določeni skupini so opisani v nadaljevanju. Obstajajo pa tudi posebni pristopi, ki črpajo principe iz obeh skupin. Vedno pa se način izdelave profila in končni profil uporabnika prilagodi na domeno problema, ki ga rešujemo.

Prva skupina metod se problema loti tako, da profil zgradi iz zbranih podatkov o uporabniku. Te metode delujejo v dveh korakih. V prvem koraku se izluščijo lastnosti iz podatkov o uporabniku. V drugem koraku uporabimo tehnike strojnega učenja za izgradnjo uporabniških profilov. Takšen pristop je natančen in ustvarjeni profili so lahko uporabni pri reševanju različnih problemov. Toda tak pristop ima tudi nekatere omejitve. Prva omejitev je, da profili zajemajo le direktno posplošene podatke. Druga omejitev pa je, da za izgradnjo dobrega profila potrebujejo zelo veliko podatkov o uporabniku. In tretja, sistem ne dobi informacij iz podobnosti med uporabniki, ampak jih mora zgraditi le iz podatkov o posamezniku [8].

Druga skupina metod uporablja drugačen pristop, poimenovan medsebojno filtriranje. Uporabniški profili so zgrajeni samo iz podatkov katere je uporabnik določil. Ti podatki se primerjajo s podatki drugih uporabnikov, z namenom iskanja enakih interesov. Vsakemu uporabniku se nato dodajo lastnosti drugih uporabnikov z enakimi interesi. Takšen pristop potrebuje manj računanja, ker ni potrebno razumevanje uporabniških podatkov. Slaba lastnost teh metod je, da za učinkovito delovanje potrebuje zelo veliko podatkov od različnih uporabnikov. Informacije lahko podamo samo specifični populaciji uporabnikov [8].

2.3.2 Načini izdelave uporabniškega profila

Profil lahko izdelamo tako, da podatke uporabnika razdelimo v dva tipa: demografske in transakcijske. Demografski podatki predstavljajo podrobnosti o tem kdo je uporabnik, pri čemer transakcijski združujejo navade uporabnika. Primer demografskih podatkov so ime, priimek, spol, starost. Transakcijski podatki pa bi bili na primer, kaj uporabnik kupuje na spletu. Iz obeh tipov podatkov je zgrajen profil uporabnika. Profil je prav tako kot podatki razde-

ljen na dva dela. Ključni del izdelave takšnega profila je validacija ugotovljenih navad uporabnika oziroma pravil. Za pridobivanje najboljših rezultatov se pravila preverijo s strani strokovnjakov. Pomanjkljivost takšnega načina je veliko pravil, ki jih je potrebno preveriti ročno. To je možno do neke mere rešiti tako, da se preveri in potrди ali zavrne več pravil hkrati [2]

Na podoben princip deluje tudi naslednji način izgradnje uporabniškega profila. Podatki se zberejo in razvrstijo v več različnih kategorij. Te kategorije so osebni podatki, zbrani podatki, dostavljeni podatki, podatki o akcijah in varnostni podatki. Iz zbranih podatkov se po kategorijah zgradi uporabniški profil. Na tak način dobimo preglednejši profil in lažje pridemo do iskanih informacij [3]. Namesto kategorij lahko uporabimo par ključ-atribut. Paziti moramo le na to, da ne pride do podvojenih parov [1].

V prvih dveh odstavkih smo opisali izdelavo profila iz izbranih podatkov. V tem primeru izdelave profila pa gre za postopek medsebojnega filtriranja. Profil izdelamo tako, da otežimo vse uporabnike, glede na podobnost s trenutno izbranim uporabnikom. Podobnost med uporabniki se določi tako, da se izmeri število enakih lastnosti. Uporabniki, ki so si najbolj podobni tvorijo soseško. Število uporabnikov v soseški določimo vnaprej [12].

2.4 Etika in varstvo podatkov

Ko zbiramo podatke, predvsem tiste o osebah, moramo paziti na etiko. Paziti moramo, kako zbrane podatke uporabljamo v posameznih primerih, da ne pride do diskriminacije. Posebej moramo biti pozorni na to, da preprečimo identifikacijo posameznikov iz izbranih podatkov. Še posebej če podatke objavljamo javno. Ko zbiramo zelo osebne podatke, moramo osebo, še preden poda podatke, obvestiti o tem, kako in zakaj bomo uporabili zbrane podatke in kako bomo zagotovili zaupnost in integriteto [13].

Pri prenašanju in shranjevanju moramo podatke šifrirati. Dostop do nešifriranih podatkov, imajo lahko samo določeni posamezniki oziroma deli sistema. Prav tako moramo odstraniti neposredno povezavo med podatki in

osebo, če za to ni potrebe. Prav tako zbrane podatke ne smemo posredovati tretjim osebam. Vedno shranimo le nujno potrebne podatke za delovanje sistema.

Poglavje 3

Pregled uporabljenih tehnologij

Za izdelavo kompleksnih informacijskih sistemov se uporabljajo različne tehnologije. Za vsak element sistema imamo na voljo več različnih tehnologij, katere imajo svoje prednosti in slabosti. Primernost tehnologij je odvisna od problema in zahtev. V nadaljevanju bomo bolj podrobno opisali tehnologije, katere bomo uporabljali pri realizaciji naše rešitve. Osredotočili se bomo tudi na to, zakaj so izbrane tehnologije najbolj primerne za naš problem.

3.1 MongoDB

MongoDB je dokumentno orientirana NoSQL (Not Only SQL) podatkovna baza. NoSQL podatkovne baze ne uporabljajo tradicionalnih tabel in relacijske strukture. MongoDB podatke shranjuje kot JSON (JavaScript Object Notation) dokumente z dinamičnimi shemami. Zaradi svoje dokumentne strukture je zelo zmogljiva podatkovna baza, ki ima enostaven podatkovni model. Glavne prednosti NoSQL baze pred relacijsko so skalabilnost, objektna orientiranost in dostopnost. NoSQL podatkovna baza ponuja stalni dostop do podatkov, ne glede na to, ali so konsistentni [10].

Z mislijo da se bo uporaba našega sistema lahko v prihodnosti povečala, smo se odločili za podatkovno bazo MongoDB. Podatkovna baza MongoDB

je zelo skalabilna¹. Uporabili smo jo tudi z razlogom, ker lahko podatke predstavimo kot objekte in s tem zmanjšamo kompleksnost uporabniškega profila. Ključni faktor pri izbiri je dostopnost do podatkov. Zelo pomembno je, da uporabnik dobi informacije, pri čemer pa ažurnost ni ključnega pomena. Večina informacij se nanaša na podatke iz preteklosti.

3.2 Java EE, Tomcat in RESTful

3.2.1 Java EE in JSP

Java EE (Enterprise Edition) je namenjena izdelavi poslovnih aplikacij. Java EE razširja Java SE (Standard Edition) in ponuja možnost izdelave spletnih storitev, distribuiranih arhitektur in drugih storitev. Del tega so tudi JavaServer Pages ali JSP. JSP ponuja možnost izdelave dinamičnih spletnih strani. S pomočjo programskega jezika Java dinamično spreminjamo vsebino HTML (HyperText Markup Language) dokumenta. Največja prednost uporabe JSP je dostop do vseh standardnih knjižnic programskega jezika Java [9].

JSP je zelo primeren zaradi samega programskega jezika Java. Java ponuja zelo obsežno standardno knjižnico. Poleg tega pa obstaja še veliko drugih knjižnic, katere se lahko zelo enostavno uporabi. Velik vpliv pri izbiri pa ima tudi zelo dobra objektna orientiranost programskega jezika Java. Objektna orientiranost nam omogoča enostavno izdelavo modularne strukture programa.

3.2.2 Tomcat

Za generiranje spletne strani, s pomočjo programskega jezika Java, se uporablja Apache Tomcat. Apache Tomcat je odprtokodni aplikacijski strežnik, kateri implementira JavaServer Pages in poganja HTTP spletni strežnik.

¹Skalabilnost podatkovne baze je lastnost, ki omogoča da bazo horizontalno razširimo čez več strežnikov.

Tomcat ima implementirane samo JavaServer Pages in ne tudi ostalih komponent Java EE, kot na primer GlassFish. Prav tako Tomcat hitreje nalaga spletne strani [15]. Ker ostalih komponent Java EE ne bomo potrebovali, nam spletni strežnik Apache Tomcat zadostuje za izdelavo naše rešitve.

3.2.3 Jersey in RESTful

Jersey je odprtokodno ogrodje za RESTful spletne storitve, napisan v Javi. Jersey razširja funkcionalnosti JAX-RS² s svojimi funkcionalnostmi, ki poenostavijo izdelavo spletnih storitev [19]. RESTful je vmesnik za programiranje spletnih storitev, kateri sloni na arhitekturnem slogu REST (Representational State Transfer). RESTful za delovanje izrablja protokol HTTP (HyperText Transfer Protocol) in njegove metode GET, PUT, POST in DELETE [23].

Izdelava spletnih storitev v arhitekturi REST je zelo preprosta in enostavna zaradi uporabe standardnega protokola HTTP. Ker uporablja protokol HTTP ni potrebe po uporabi dodatnih knjižnic, saj ima večina sistemov podporo že privzeto vgrajeno. Jersey poenostavi uporabo REST arhitekture, toda ne predstavlja potrebe po uporabi Jersey knjižnice na obeh straneh komunikacijskega kanala.

3.3 Highcharts in jQuery

Knjižnica Highcharts omogoča izris različnih interaktivnih grafov na spletni strani. Uporablja programski jezik JavaScript in funkcionalnosti HTML5 (HyperText Markup Language). Highcharts ponuja prikaz zelo veliko dodatnih elementov grafa, kot so legende, označbe podatkov itd. Prav tako lahko zelo podrobno prilagajamo izgled grafa in prikaz podatkov [18]. Za lažje programiranje funkcionalnosti v JavaScriptu se uporablja ogrodje jQuery. JQuery ponuja veliko poenostavljenih osnovnih funkcionalnosti, kot so na-

²JAX-RS je del Java EE, ki ponuja funkcionalnosti za izdelavo REST spletnih storitev.

predno izbiranje HTML elementov, ustvarjanje animacij, spreminjanje elementov, rokovanje z dogodki in mnoge druge funkcionalnosti [20].

3.4 Android SDK

Android je operacijski sistem za mobilne naprave, ki temelji na jedru Linuxa. Operacijski sistem Android je razvilo podjetje Google, ki tudi skrbi za nadgradnje. Razvijalcem aplikacij, Google ponuja Android razvojna orodja (SDK). Android SDK (Software Development Kit) vključuje emulator, knjižnice, dokumentacijo, razvojno okolje in ostala orodja namenjena razvoju in testiranju.

Za razvoj aplikacij za operacijski sistem Android imamo ponujene štiri osnovne elemente. To so aktivnosti, storitve, sprejemniki in ponudniki vsebin. Aktivnosti so odseki kode, ki skrbijo za izvajanje glavnih aktivnosti, kot so interakcija z uporabniškimi vnosi, izvajanje ukazov, klici storitev itd. Aktivnosti se po navadi ujemajo s prikazom na zaslonu. Vsaka aktivnost predstavlja enega izmed zaslonov (npr. nastavitve, glavni prikaz, izbira časa). Drugi osnovni del so storitve. Storitve so procesi, ki se večinoma izvajajo v ozadju. V večini primerov nimajo uporabniškega vmesnika. Primer storitve je proces, ki skrbi za predvajanje glasbe v ozadju. Neposredno s storitvami in aktivnostmi so povezani sprejemniki. Sprejemniki skrbijo za posredovanje dostopa do funkcionalnosti aktivnosti ali storitev ob prejemu zahtevka iz drugih aplikacij. Zadnji del pa so ponudniki vsebin. Ponudniki vsebin so namenjeni deljenju podatkov z drugimi aktivnostmi ali storitvami. Ponudniki definirajo standarden vmesnik z dostopom preko naslova URI (Uniform Resource Identifier). Preko vmesnika ostale aplikacije dostopajo do podatkov, ki jih aktivnost ali storitev ponuja [11].

Mobilna platforma Android je zelo odprt sistem. Na platformi Android lahko dostopamo do podatkov kot so zgodovina klicev, poslana in prejeta SMS/MMS sporočila in število prenesenih podatkov preko mobilnega interneta. Tip aplikacije, kjer bi dostopali do prej omenjenih podatkov na plat-

formi iOS in Windows Phone ni mogoč. Ti dve platformi omejujeta dostopa do teh podatkov. Na platformi Android je dostop do podatkov omejen le s potrditvijo uporabnika.

3.5 JSON

JavaScript Object Notation (JSON) je odprta standardna oblika zapisa podatkov. Izvira iz JavaScript programskega jezika in je tam tudi privzeto podprt. Zapis je sestavljen iz podatkovnih objektov, kateri vsebujejo par ključa in vrednosti. Vrednosti so lahko tipa niz, številka, objekt, tabela, logične vrednosti ali null. Oblika zapisa je človeku prijazna, prav tako pa jo lahko zelo enostavno strojno generiramo in razčlenimo. Format JSON je popolnoma neodvisen od programskega jezika, kar ga naredi zelo uporabnega za izmenjavo podatkov [25].

Zapis podatkov v formatu JSON je zelo enostaven. Je veliko bolj enostaven in manj kompleksen, kot na primer XML (Extensible Markup Language). MongoDB ga uporablja za obliko zapisa podatkov in prav tako za oblikovanje pogojev in ostalih atributov. Odstrani nam potrebo po spreminjanju oblike zapisa podatkov pri shranjevanju podatkov v bazo pridobljenih preko spletne storitve.

Poglavje 4

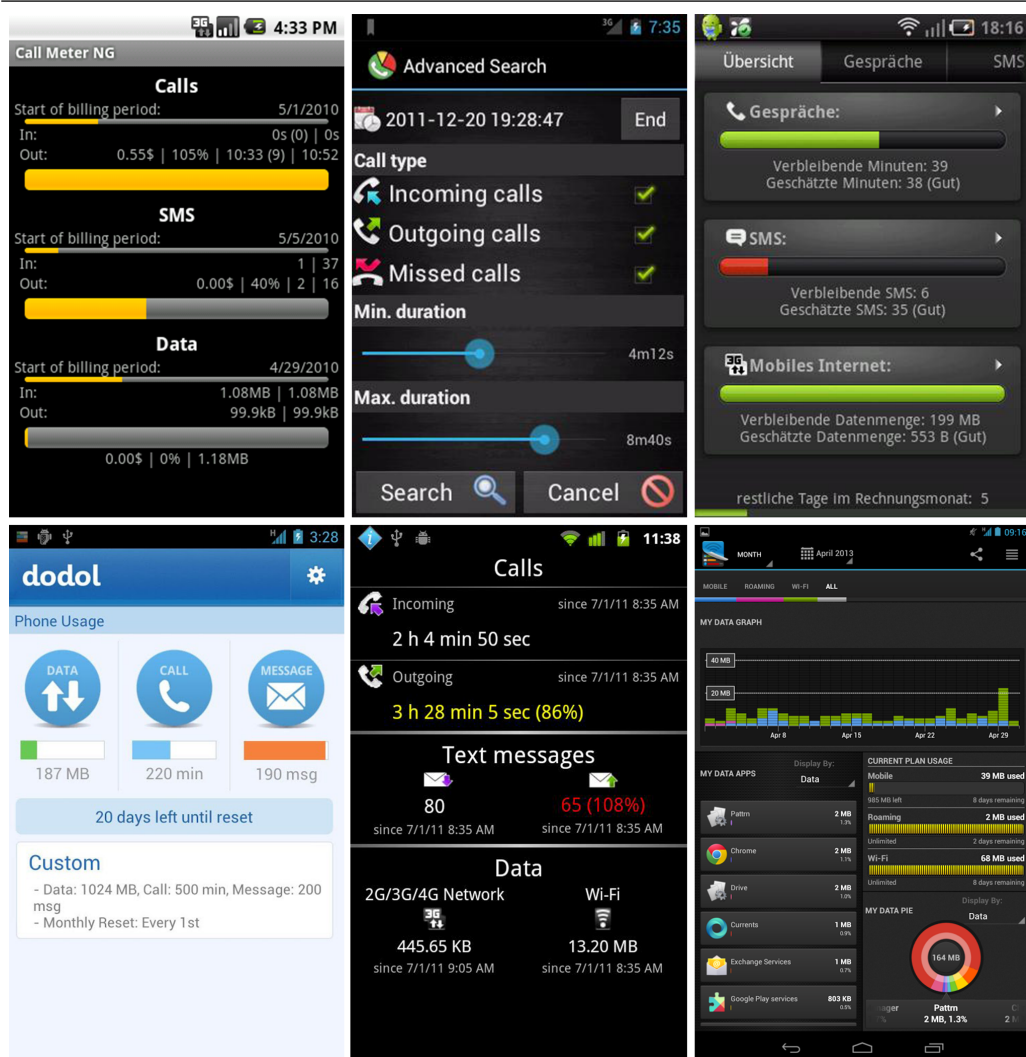
Sorodne rešitve na področju spremljanja in porabe mobilnih storitev

V uvodu smo zapisali, da na tržišču obstajajo različni programi za spremljanje in merjenje porabe mobilnih storitev. Nekateri ponujajo tudi naprednejše funkcionalnosti. Na sliki 4.1 so prikazani različni uporabniški vmesniki takšnih aplikacij. V nadaljevanju si bomo pogledali omenjene programe in izpostavili njihove dobre in slabe lastnosti. Glavni namen pa je ugotoviti njihovo skupno pomanjkljivost in predlagati izboljšano rešitev.

Dodal Phone aplikacija ponuja prikaz porabe za prenos podatkov, SMS/-MMS sporočil in klicev. Poraba se loči na odhodno in dohodno. Določimo lahko mesečno kvoto porabe enot. Statistika porabe se prikazuje po mesecih in dnevih. Napredne funkcije so prikaz števila predlagane porabe dnevni enot in nastavljanje omejitev. Program ne ločuje porabljenih enot na omrežja in državo, prav tako ne ponuja naprednejšega svetovanja o porabi enot mobilne storitve [17].

Aplikacija Stats Free ponuja zgolj pregled porabe mobilnih enot za posamezen dan, teden in mesec [24]. Ima enake pomanjkljivosti kot Dodal Phone, poleg tega pa ne ponuja niti prikaza osnovne statistike niti dodatnih nasvetov

POGLAVJE 4. SORODNE REŠITVE NA PODROČJU SPREMLJANJA
IN PORABE MOBILNIH STORITEV



Slika 4.1: Aplikacije za merjenje porabe mobilnih enot. Zgoraj levo Call Meter NG, sredina Call Stats, desno DroidStats, spodaj levo Dodal Phone, sredina Stats Free in desno My Data Manager

o porabi mobilnih enot. Zelo podobni aplikaciji sta DroidStats in Call Meter NG. Prednost DroidStats aplikacije je dodatna funkcionalnost prikaza treh največkrat klicanih telefonskih števil [22]. Call Meter NG pa ima prednastavljene podatke o količinah enot pri nekaterih mobilnih storitvah [14]. Pri obeh so pomanjkljivosti enake kot pri Stats Free.

Aplikaciji ki nekoliko odstopata od prejšnjih sta Call Stats in My Data Manager. Call Stats ponuja zgolj pregled porabe in statistike klicev. My Data Manager omogoča spremljanje porabljenih enot mobilnega interneta. Naprednejše funkcionalnosti aplikacije Call Stats so prikaz napredne statistike, ki ponuja pregled povprečnega in skupnega časa ter število klicev po posameznih številkah. Klici se ločujejo na odhodne in dohodne ter na to, ali so bili izvedeni v tujini ali doma [16]. Prednost aplikacije My Data Manager je prikaz porabe enot mobilnega interneta za posamezno aplikacijo. Poleg tega ponuja prikaz različnih grafov porabe za različna časovna obdobja in aplikacije [26].

Nobena izmed aplikacij ne ponuja naprednejšega svetovanja o bolj učinkoviti izrabi enot. Želeli bi, da nam aplikacija poleg naprednejše statistike in prikaza porabljenih ter preostalih enot prikaže tudi, katera mobilna storitev je za nas najbolj primerna glede na našo porabo enot. Prav tako pa tudi koliko enot smo porabili v tujini in kaj naj storimo, da zmanjšamo stroške. Želeli bi tudi, da aplikacija zazna odstopanja v porabi enot, nas o tem obvesti in prikaže spremembe. Gledano v celoti želimo pametno aplikacijo, ki nadzoruje porabo in poskrbi, da dobimo informacije o najcenejši izrabi mobilnih enot.

Poglavje 5

Predlog rešitve optimizacije stroškovnega vidika mobilne uporabniške izkušnje

Kot smo zapisali v prejšnjem poglavju, je naš namen izdelati pametnega mobilnega pomočnika. Pametni mobilni pomočnik bo uporabniku prikazoval podatke o izbrani storitvi, trenutno porabo posameznih enot, porabo enot v preteklosti in primerjavo porabe enot s povprečjem porabe enot ostalih uporabnikov. Ponujal bo tudi prikaz naprednejše statistike. Pod naprednejšo statistiko spada procentni prikaz porabljenih enot pri posameznem mobilnem omrežju skozi različna časovna obdobja in razporeditev mobilnih storitev v prostoru glede na ponudnika in ceno mesečne naročnine. Glavni del mobilnega pomočnika in nadgradnja obstoječih rešitev bodo napredne funkcionalnosti:

- prikaz najprimernejše mobilne storitve glede na povprečno porabo mobilnih enot;
- svetovanje glede porabe enot v tujini;
- odkrivanje odstopanja pri mesečni porabi enot;
- prikaz najprimernejših obračunskih intervalov glede na trajanje klicev.

V začetni fazi je bil cilj, izdelati mobilno aplikacijo, katera zbira uporabniške podatke o uporabi mobilnih storitev. Iz zbranih podatkov izlušči informacije in uporabniku prikaže porabo ter svetuje kako zmanjšati ceno uporabe mobilnih storitev. Celoten proces bi se izvajal na mobilni napravi in prav tako bi se na mobilno napravo shranjevali vsi podatki. Ko smo boljše preiskali področje smo ugotovili, da je takšna aplikacija zelo omejena ter sistemsko potratna.

Glavni problem pri mobilnih aplikacijah je ta, da je sistemska zmogljivost telefonov še vedno relativno majhna, predvsem ko gre za procesiranje velike količine podatkov. Prav tako se pojavi problem shranjevanja podatkov. Ne samo velika poraba prostora na mobilni napravi, ampak tudi izguba podatkov, če uporabnik izbriše aplikacijo oziroma zamenja telefon. Sistem naj bi najbolje deloval tako, da ima podatke shranjene za čim daljše časovno obdobje. Poleg tega v primeru, če imamo podatke shranjene lokalno na napravi, ne moremo izvajati operacij, kjer bi na primer podatke enega uporabnika primerjali z drugimi. Drugi problem pa je v tem, da je v tem primeru ogled informacij omejen na mobilno napravo. Ideja je bila, da lahko uporabnik pregleduje oziroma dostopa do informacij od kjer koli in iz katere koli naprave.

Glede na vse omenjene pomanjkljivosti smo si celoten sistem zamislili kot skupek treh osnovnih komponent. Glavna komponenta bi bila strežniška aplikacija, katera skrbi za obdelavo podatkov. Za zbiranje podatkov bi skrbela preprosta mobilna aplikacija in baza kjer se vsi podatki shranjujejo. Na sliki 5.1 je prikazana shema celotnega sistema.

Mobilna aplikacija naj bi bila kar se da enostavna in bi potrebovala čim manj sistemskih virov. Sprva smo hoteli narediti aplikacijo, ki bi neprestano tekla v ozadju in zbirala podatke. Toda takšen način delovanja je zelo potraten. Uporabniki bi bili nezadovoljni, saj bi jim aplikacija skrajšala čas med polnjenji baterije mobilne naprave. Zato smo razmišljali v drugi smeri. Mobilne naprave že same shranjujejo podatke o klicih, poslanih SMS/MMS sporočilih, prenesenih podatkih. Zato smo mobilno aplikacijo zastavili tako, da na začetku, ko odpremo aplikacijo, zbere vse podatke iz mobilne naprave,



Slika 5.1: Shema sistema z izbranimi tehnologijami.

jih zapakira in jih preko spletne storitve pošlje aplikaciji na strežniku. Poleg tega naj bi mobilna aplikacija poskrbela še za prikaz spletne strani s povratnimi informacijami. Mobilna aplikacija za delovanje tako potrebuje le dostop do spleta in po potrebi geolokacijske storitve za bolj natančno določitev izvora podatkov. Podatkom določimo izvor, zato da lahko spremljamo tudi porabo v tujini in s tem še bolj natančno določimo porabo sredstev in odločitve glede ponujenih možnosti boljše izrabe mobilnih storitev.

Strežniška aplikacija v prvi vrsti ponuja spletne storitve, preko katerih sprejme poslano podatke iz mobilne aplikacije. Prejete podatke shranjuje v bazo. Strežniška aplikacija skrbi za agregacijo in obdelavo podatkov. Prvič se obdelajo podatki za vse mesece nazaj, pri čemer se spusti trenutni mesec. V nadaljevanju se podatki ne obdelujejo nenehoma, ampak se to zgodi le enkrat na mesec. Rezultati obdelave podatkov se shranijo v bazo. Namen tega je, da ni potrebno vsakič, ko želimo prikazati informacije, ponovno obdelati vseh podatkov. Strežniška aplikacija rezultate obdelave in ugotovitve iz podatkov prikazuje na spletni strani. Spletne strani je prilagojena za prikaz na različnih napravah. Na spletni strani se uporabljajo različni pristopi prikazovanja

informacij. Eden teh so različni grafi. Strežniška aplikacija mora biti hitra in odzivna ter sposobna obdelave velike količine podatkov, zato je pomembno, da je izbran napreden programski jezik.

Najbolj pomembna lastnost baze je, da je sposobna sprejeti in zapisati veliko število podatkov. Prav tako pa da jo je možno razširiti. Ažurnost podatkov ni ključnega pomena. Želimo čim bolj pretočno in enostavno branje ter zapisovanje podatkov. Če število uporabnikov naraste, želimo da se poslani podatki shranijo brez večjih zakasnitev.

Poglavje 6

Opis inteligentnega sistema

V prejšnjem poglavju smo opisali konceptualni predlog celotnega sistema in na splošno predstavili način delovanja. V tem poglavju se osredotočimo na način realizacije zastavljenih problemov in zahtev. Opisani so posamezni sklopi sistema in njihov glavni namen. Predvsem smo se osredotočili na to, kako sklopi delujejo in kako smo uporabili nekatere koncepte iz podatkovnega rudarjenja ter izdelave uporabniških profilov pri realizaciji rešitev.

6.1 Podatkovna baza

Profil uporabnika je sestavljen iz več različnih delov. Vsi podatki, ki definirajo profil so shranjeni v posameznih tabelah v bazi (slika 6.1). Osnovni del so podatki kateri obsegajo ime, priimek, geslo, EMŠO, naslov ter mobilne storitve, katere je uporabnik uporabljal oziroma storitev katero uporablja. Osnovni podatki so shranjeni v zbirki 'uporabnik'. Drugi del profila so podatki, kateri se navezujejo na uporabo mobilnih storitev. Ti so shranjeni v zbirki 'podatki'. Podatki o porabi storitve so opisani z naslednjimi atributi datum, tip, lokacija, trajanje pri klicih oz. količina pri prenosu podatkov ter seveda ali gre za odhodni ali dohodni podatek. Ti podatki definirajo klice, poslana SMS/MMS sporočila ter prenesene podatke. Prej omenjeni podatki se prenašajo z mobilne naprave uporabnika s pomočjo mobilne aplikacije,

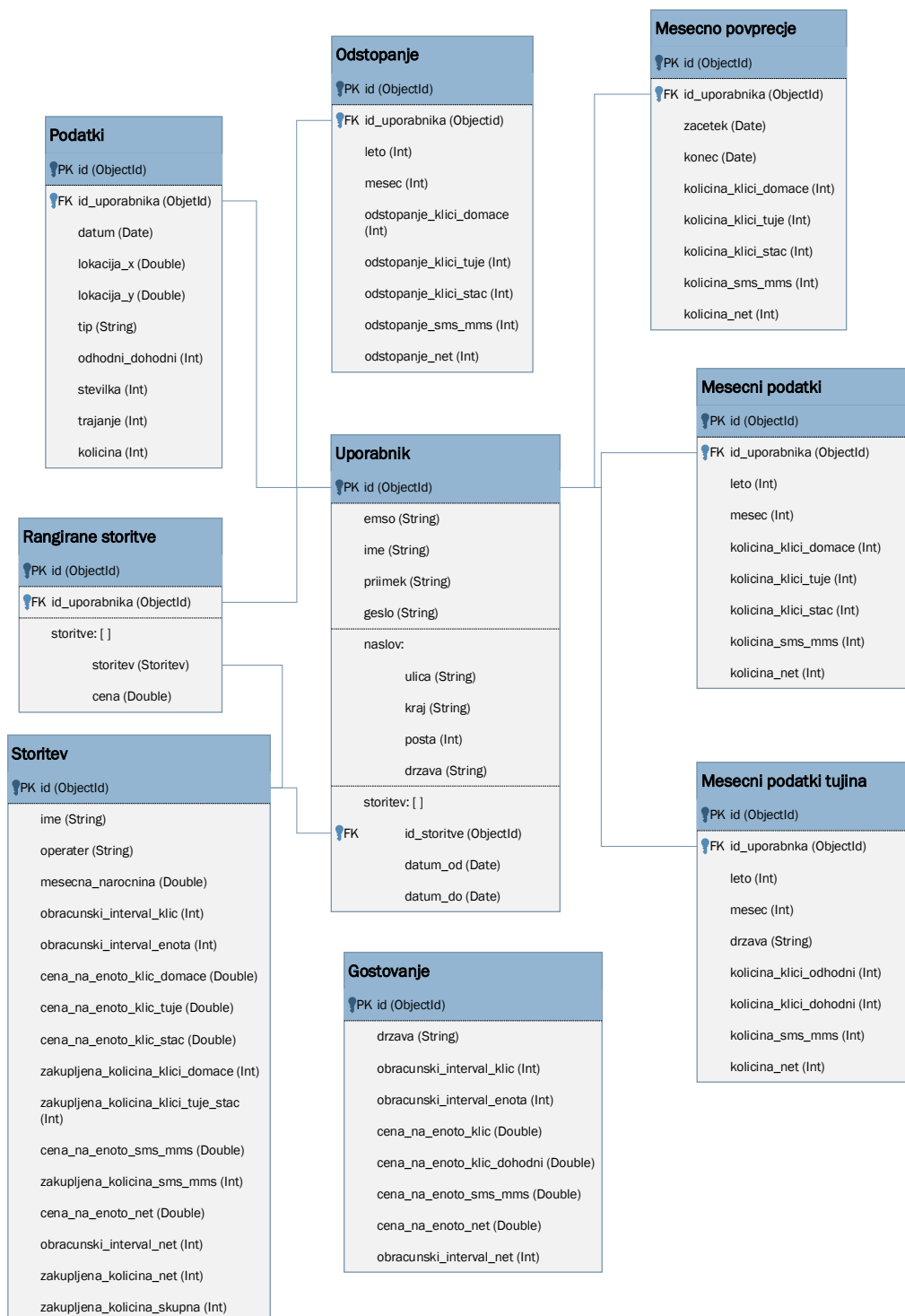
katera je opisana v nadaljevanju.

Ostali deli profila uporabnika se zgradijo s pomočjo agregacije in obdelave osnovnih podatkov. Celoten postopek je opisan v nadaljevanju. Podatki se najprej seštevajo po posameznem mesecu, tako agregirani podatki se shranijo v zbirko 'mesecniPodatki'. V zbirki so shranjeni seštevki posameznih enot ter leto in mesec sešteti podatkov. Povprečna poraba posameznih enot čez celotno obdobje je shranjena v zbirki 'mesecnoPovprecje', katera je definirana z začetnim in končnim datumom. Podatki iz tujine se obravnavajo posebej, zato so tudi shranjeni v ločeni zbirki z imenom 'mesecniPodatkiTujina'. Ti podatki vsebujejo nekoliko drugačne atribute, saj se enote v tujini obravnavajo drugače. Glavna razlika je, da se odhodni in dohodni klici seštevajo ločeno. Poleg tega je v zbirki zapisana država, kjer so bile enote porabljene.

V nadaljevanju procesiranja podatkov se ugotovijo odstopanja, ki se shranijo v zbirki 'odstopanje'. Tu je zapisano, za koliko posamezna enota odstopa od povprečja v nekem mesecu leta. Na koncu procesiranja in rangiranja storitev, se le-te shranijo v zbirko 'rangiraneStoritve'. V zbirki se poleg podatkov o storitvi shrani tudi cena storitve glede na povprečno porabo. Rangirane storitve se shranjujejo z namenom manjše porabe strojne moči. Saj se podatki le preberejo iz baze in jih ni potrebno ponovno preračunavati.

Ostali podatki, ki so neposredno povezani s samim profilom, so še podatki o storitvah in gostovanju. Podatki o storitvah so shranjeni v zbirki 'storitev' in imajo atribute, ki opisujejo cene, količine in obračunske intervale posameznih enot storitve. V zbirki 'gostovanje' so shranjeni podatki o ceni in obračunskem intervalu posameznih enot v tujini glede na državo. Celotna shema baze z vsemi zbirkami in njihovimi atributi je prikazana na sliki 6.1.

Da bi zagotovili zasebnost uporabnikov smo gesla shranili kot hash generiran z algoritmom MD5. Poleg tega pa smo pri mobilnih številkah shranili le tisti del, ki določa za katerega operaterja gre. Mobilne številke se že na mobilni napravi uporabnika skrajšajo. S tem preprečimo, da bi nekdo, ki bi prestregel poslane podatke na strežnik, lahko pridobil pomembne osebne informacije.



Slika 6.1: Shema baze s pripadajočimi zbirkami.

6.2 Nastavitvene datoteke

Nastavitve procesov, ki se izvajajo na strežniku, smo shranili v nastavitveno datoteko 'general.properties'. Tu so shranjene nastavitve kot so: meja za število odstopanj v tujini, meja za število različnih tujih držav, dovoljeni prag odstopanja posamezne enote, standardne vrednosti enot in obračunskih intervalov za tujino, število dni v testni množici pri mesečni statistiki, obračunski intervali in ostale nastavitve. Nastavitve so zapisane v standardni obliki Javanske nastavitvene datoteke. S temi nastavitvami in opcijami prilagajamo izvajanje posameznih procesov. Tako dobimo kar se da prilagodljiv sistem, katerega lahko dodatno prilagodimo za čim bolj optimalno delovanje.

Podatki za povezavo z bazo so shranjeni v datoteki 'database.properties'. Nastavitvena datoteka nam omogoča hiter prenos sistema na drug strežnik oziroma uporabo drugega strežnika za bazo. V sami kodi programske rešitve tako ni potrebe po spremembi.

6.3 Strežniška aplikacija

Glavni del sistema predstavlja strežniška aplikacija. Sestavljena je iz procesov, spletnih strani in spletnih storitev. Vsi elementi so opisani v nadaljevanju. Glavna naloga strežniške aplikacije je sprejemanje, hranjenje in obdelava podatkov. Prav tako pa tudi pridobivanje in prikaz informacij. Vsi elementi strežniške aplikacije se med seboj povezujejo in dopolnjujejo. Njihova skupna točka je podatkovna baza.

6.3.1 Procesi

Najpomembnejši element strežniške aplikacije so procesi. Procesi realizirajo napredne funkcionalnosti pametnega mobilnega pomočnika, kot so rangiranje storitev, iskanje odstopanj in ostale funkcionalnosti, ki skrbijo za pridobivanje koristnih informacij iz zbranih podatkov o uporabniku. Rezultati teh procesov se nato prikažejo na spletni strani, katera je prav tako opisana v

nadaljevanju. Procesi so med seboj zelo odvisni, saj se pridobljeni rezultati iz enega procesa uporabijo kot vhodni podatki pri drugemu. V nadaljevanju bomo opisali delovanje procesov in njihov namen.

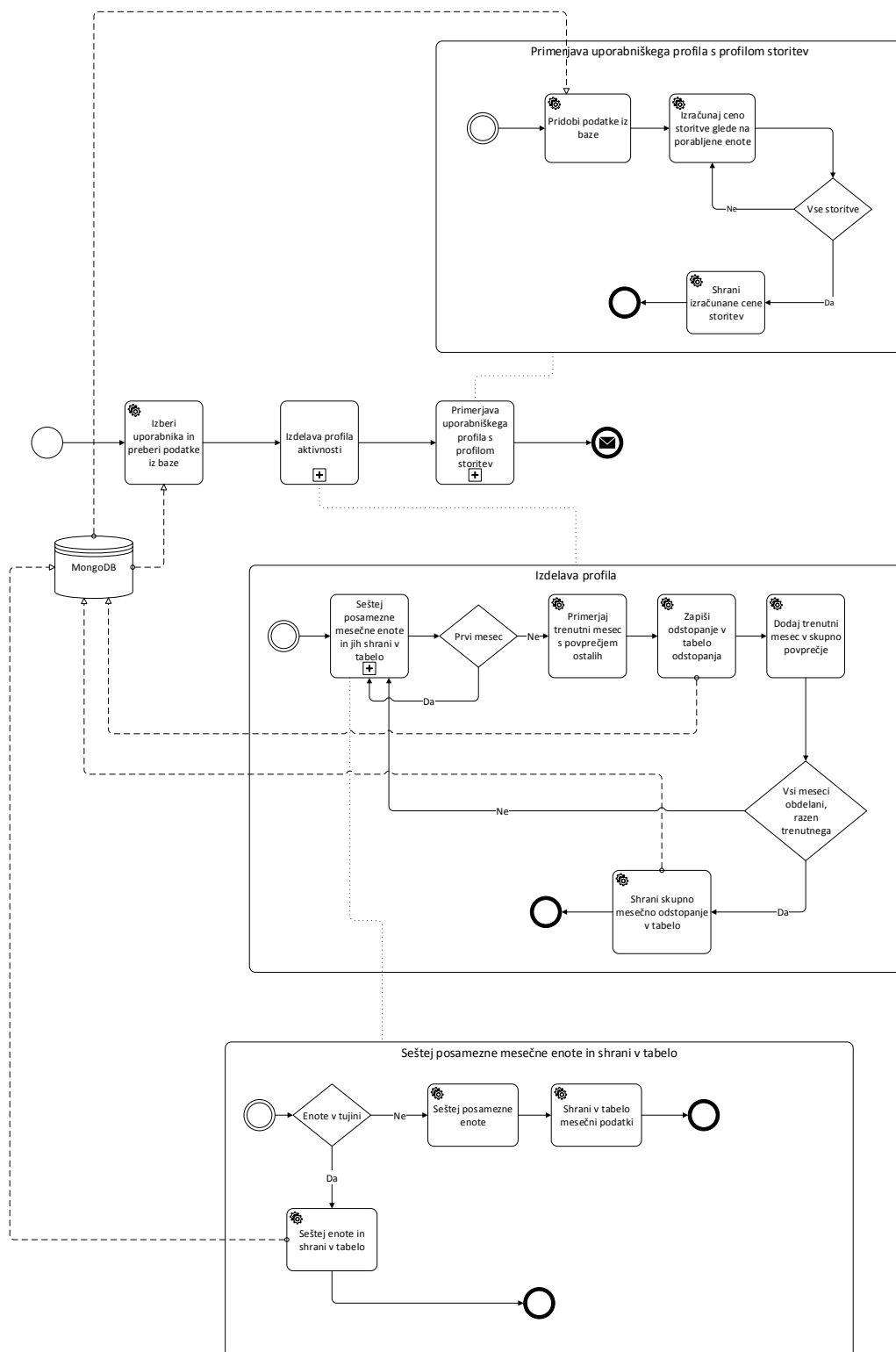
Rangiranje storitev

Rangiranje storitev je glavni proces, kateri izlušči osnovne informacije, ki se nato uporabijo tudi pri nekaterih ostalih procesih. Ključni del procesa je metoda podatkovnega rudarjenja, ki povzame oziroma posploši podatke. Namen procesa je agregirati podatke o porabi in nato izračunati njihovo povprečje. Poleg tega se zabeležijo odstopanja in poraba enot v tujini. Iz povprečja se nato za vsako storitev izračuna cena. Proces vrne razvrščene storitve, katere se na spletni strani prikažejo na seznamu. Uporabnik ima na spletni strani možnost prilagoditve prikaza storitev, tako si prikaz seznama prilagodi svojim željam. Proces uporabniku poda informacijo o najbolj primernih storitvah glede na njegovo povprečno porabo enot.

Ključnega pomena pri razvrščanju porabe enot je določitev omrežja. Za določanje omrežja številke, smo uporabili prvi del številke, kateri to privzeto določa. Ker lahko uporabnik zamenja omrežje, in obdrži isto številko, prej opisano določanje ni več pravilno. Da bi rešili ta problem, smo poskušali najti način določanja omrežja številke. Dostopanje do informacij, pri katerem operaterju se številka nahaja, je mogoče na spletni strani podjetja Teletech d.o.o., toda je omejeno oziroma onemogočeno za preverjanje več števil hkrati [21]. Podjetje Teletech d.o.o. prav tako nima dovoljenja, da bi posredovala vse informacije o telefonskih številkah tretjim osebam. Odločili smo se, da za potrebe izdelave prototipa nam bo zadostovalo določanje omrežja iz prvega dela številke.

Delovanje Osnovni oris delovanja procesa je prikazan na sliki 6.2. V nadaljevanju bomo bolj podrobno opisali, kaj se zgodi v posameznem sklopu. Bolj podrobno bomo opisali tudi, kako implementiramo metodo povzemanja.

Na začetku obdelave podatkov se določi obseg. Najprej se preveri ali



Slika 6.2: Proces rangiranja storitev s podrobnostmi izdelave profila, seštevanja enot in primerjavo profilov.

v bazi že obstajajo zapisi o obdelanih podatkih, kot so mesečna povprečja in skupno mesečno povprečje. Če ne obstajajo, potem se za obseg izberejo vsi meseci uporabe storitve brez prvega in trenutnega meseca. Če obstajajo, potem se kot obseg določijo le meseci, pri katerih podatki še niso bili obdelani. Ko je obseg določen, se iz baze naložijo podatki.

Ko so podatki naloženi, se posamezne enote v izbranem mesecu, katere niso porabljene v tujini, seštejejo. Seštevajo se le odhodne enote razen pri internetni porabi. Posebej se obravnavajo klici, saj se seštevajo glede na to ali so bili opravljeni v domačem, tujem ali stacionarnem omrežju. Seštete enote se nato shranijo v bazo kot zbirka mesečne porabe. Poleg seštevka enot se v zbirko shrani še leto in mesec združenih enot.

Enote porabljene v tujine se seštejejo na nekoliko drugačen način. Prva razlika je pri klicih, saj se ne seštevajo po omrežjih, ampak se glede na to ali so odhodni ali dohodni. V tujini se namreč zaračunavajo tudi dohodni klici. Druga razlika je, da se enote ločijo tako na mesec kot državo, v kateri so bile porabljene. Seštevek enot se skupaj z letom, mesecem in državo shranijo v zbirko mesečnih podatkov tujine.

Iz izračunanih seštevkih mesečne porabe enot, brez tistih iz tujine, se izračuna še skupno mesečno povprečje. Če mesečno povprečje že obstaja, se na novo izračunano mesečno povprečje doda skupnemu. Preden pa se mesečno povprečje doda v skupno mesečno povprečje, se preveri še odstopanje. Vsaka enota mesečnega povprečja se preverja z enoto iz skupnega povprečja. Izračuna se procent odstopanja. Če je procent odstopanja vsaj ene enote večji od meje, ki je zapisana v nastavitveni datoteki za to enoto, se obravnava kot odstopanje. Takšno mesečno odstopanje se shrani v bazo. V zbirki so shranjeni procenti odstopanj za posamezno enoto ter mesec in leto odstopanja. Na koncu se skupno mesečno povprečje shrani v bazo.

Ko so izbrani vsi podatki o mesečni in povprečni porabi, se izvede rangiranje storitev. Najprej se storitve mobilnih ponudnikov naložijo iz baze. Za vsako storitev se nato izračuna cena glede na porabo enot iz skupnega mesečnega povprečja. Cena se izračuna, tako da se upoštevajo zakupljene

količine in cene posameznih enot. Če obstajajo zakupljene količine se najprej le-te porabijo, nato se dodatne enote zaračunavajo glede na ceno enote. Storitve z izračunano ceno se shranijo v bazo.

Rezultat procesa so razvrščene storitve glede na ceno, od tiste z najnižjo ceno do tiste z najvišjo.

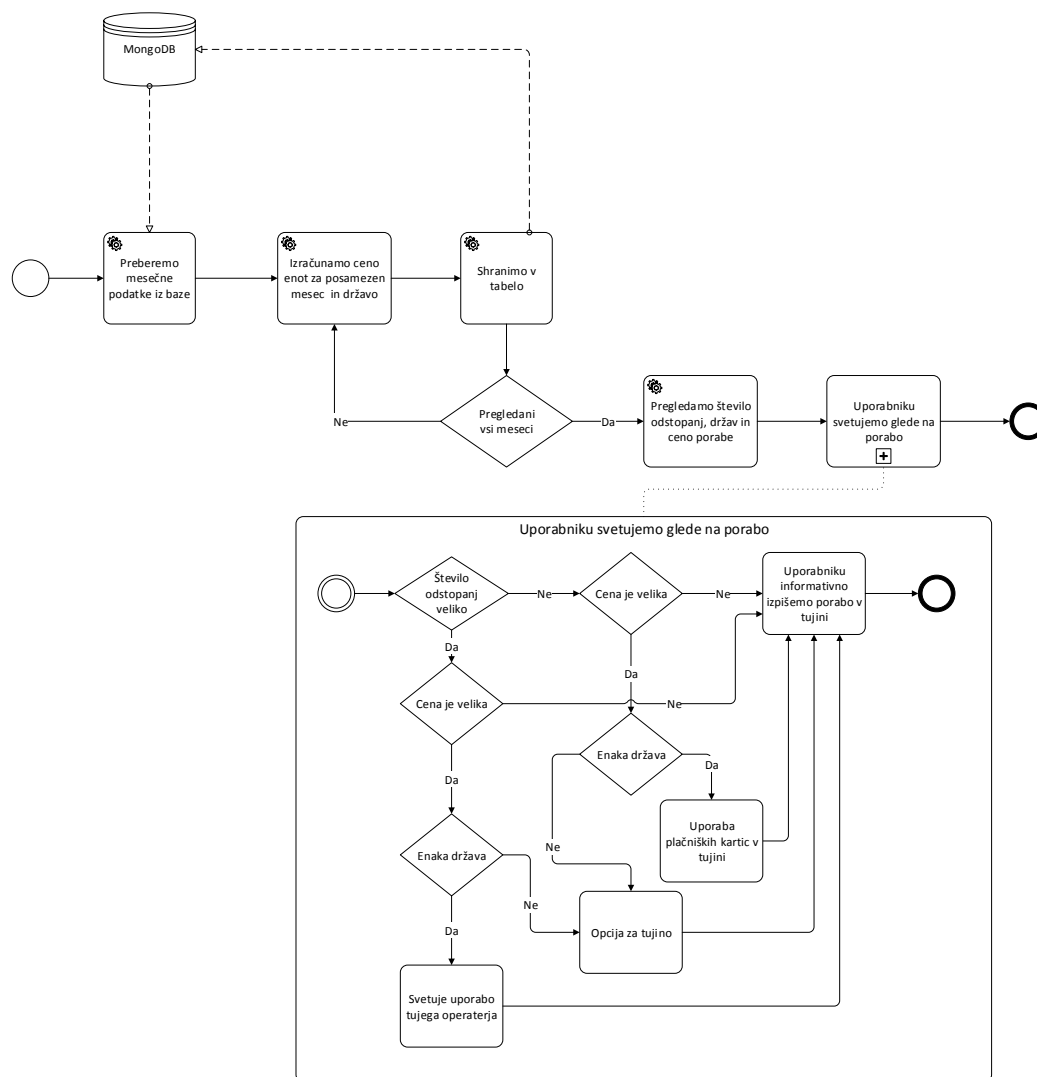
Tujina

Namen tega procesa (slika 6.3) je obveščanje uporabnika o porabi enot v tujini. Uporabniku svetuje, na kakšen način naj v prihodnosti poskrbi za uporabo mobilnih storitev v tujini, tako da bo cena porabljenih enot čim manjša. Uporabnik dobi informacije o številu obiskov v tujini, številu različnih obiskanih držav, ceni vseh porabljenih enot v tujini ter povprečno mesečno ceno. Ključna informacija, ki jo uporabnik dobi, pa je nasvet o načinu uporabe mobilnih storitev v tujini.

Delovanje Podatki o porabljenih enotah v tujini se na začetku naložijo iz baze. Cena posamezne enote v izbrani državi se prav tako naloži iz baze. Če tega podatka ni v bazi, se uporabi privzeta cena enote za tujino, zapisana v nastavitveni datoteki. Ko so podatki pripravljene, se preračuna skupna cena porabljenih enot v tujini. Poleg izračuna cene porabljenih enot se ugotovi še število različnih držav in število obiskov tujine iz podatkov o lokaciji. Iz izračunanih podatkov se oblikuje nasvet po diagramu iz zadnjega opravila prikazanega na sliki 6.3. Meja katera določa ali gre za veliko odstopanj, državi oziroma ceno se prebere iz nastavitvene datoteke. Na koncu se vrne rezultat v obliki nasveta, števila odstopanj, števila držav, skupne in povprečne mesečne cene porabljenih enot v tujini.

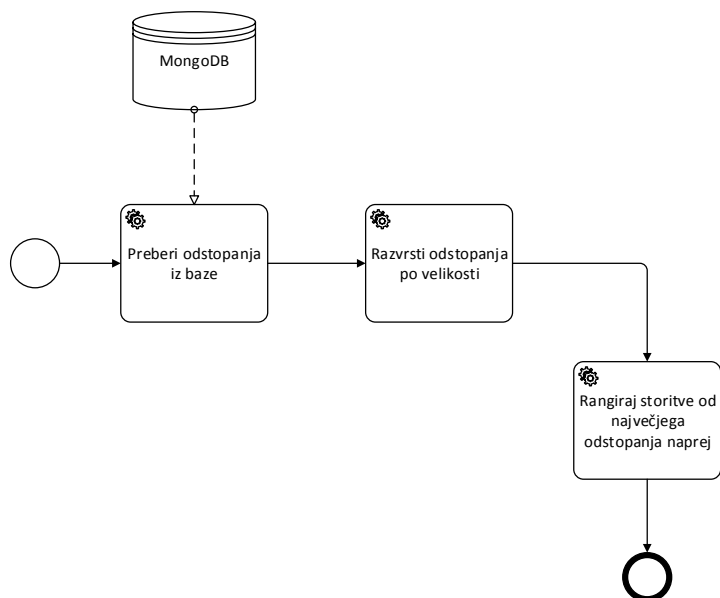
Odstopanja

Proces odstopanja zazna spremembe v uporabi mobilnih storitev in ponovno rangira storitve. Če uporabnik spremeni navade, želimo te spremembe zaznati in jih upoštevati pri rangiranju storitev. Proces poišče največje odsto-



Slika 6.3: Diagram procesa obdelave porabljenih enot v tujini.

panje in od tam naprej ponovno rangira storitve. Ponovno rangirane storitve se shranijo v obliki seznama. Splošni oris tega procesa je prikazan na sliki 6.4.



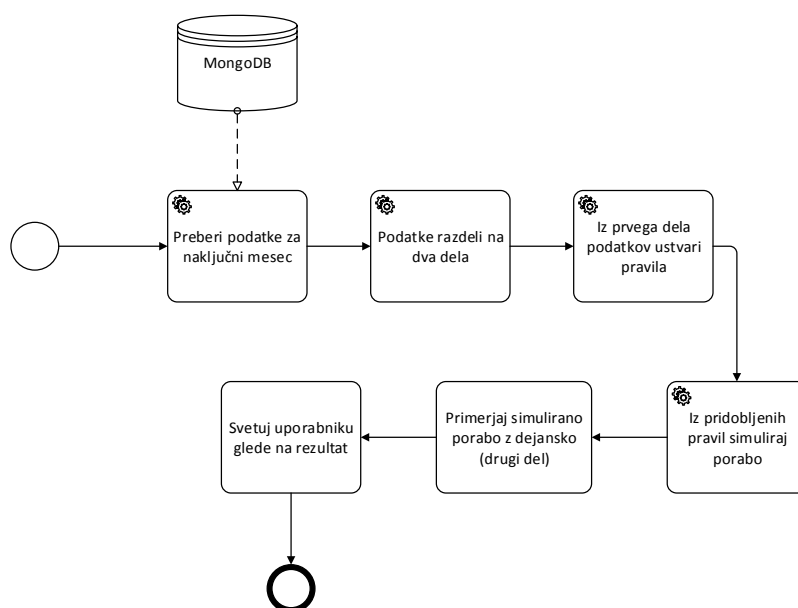
Slika 6.4: Proces iskanja odstopanj.

Delovanje Podatki o odstopanjih se najprej naložijo iz baze. Ko so odstopanja naložena, začnemo s procesom iskanja največjega odstopanja. Eno odstopanje vsebuje podatke o razliki posamezne enote od skupnega mesečnega povprečja. Največje odstopanje se poišče tako, da se posamezna odstopanja po enotah seštejejo za vsak mesec posebej. Mesečna odstopanja se primerjajo med seboj in izbere se največje odstopanje. Če največje odstopanje ni enako zadnjemu odstopanju se ponovno požene sistem za rangiranje storitev od datuma največjega odstopanja naprej. Namen pogoja je, da ne poženemo procesa rangiranja storitev nad neobstoječimi podatki. Postopek ponovnega rangiranja je enak opisu zgoraj, le da so podatki že agregirani in se le naložijo iz baze. Poleg tega se rezultati ne shranijo v bazo. Ko se postopek ponovnega rangiranja konča, proces odstopanja vrne kot rezultat ponovno rangirane storitve in datum največjega odstopanja.

Mesečna statistika

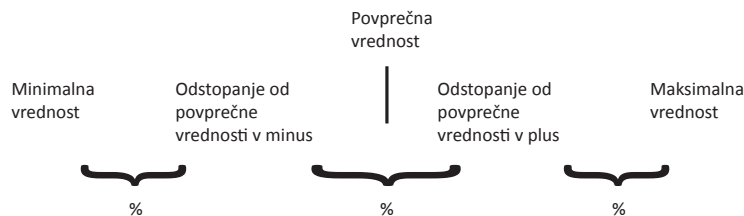
Nekateri uporabniki spremenijo način uporabe storitev, ko opazijo da jim zmanjkuje zakupljenih enot. Začnejo paziti koliko enot porabijo in posledično imajo manjšo porabo. Takšen način uporabe ni običajen in privede do slabših rezultatov razvrščanja storitev. S procesom mesečne statistike poskušamo zaznati takšne spremembe uporabe mobilnih storitev. Uporabnika želimo, ob zaznavi takšnih sprememb, opozoriti. Proces vrne informacijo, za koliko procentov se je zmanjšala poraba posameznih enot in uporabniku svetuje o izbiri druge mobilne storitve.

Za odkrivanje sprememb pri mesečni porabi smo uporabili princip klasifikacije podatkov. To je metoda podatkovnega rudarjenja, ki deluje po postopku prikazanem na sliki 6.5.



Slika 6.5: Postopek klasifikacije podatkov za potrebe odkrivanja sprememb pri porabi enot.

Delovanje Za obdelavo se avtomatsko izbere mesec pred trenutnim mesecem. Obstaja pa tudi opcija ročnega vnosa meseca in leta. Iz baze se za izbrani mesec naložijo podatki za posamezne enote. Naloženi podatki se razdelijo na učno ter testno množico v razmerju, ki je podano v nastavitveni datoteki. Učna množica obsega prvo polovico meseca, testna pa drugo. Iz učne množice ustvarimo klasifikatorje. Klasifikator definira porabo posamezne enote v enem dnevu. Definiran je s povprečno vrednostjo enote, povprečnim odstopanjem od povprečja v pozitivni in negativni smeri, minimalno in maksimalno vrednostjo enote, ter procentno razdelitev enot. Procentna razdelitev enot definira, koliko procentov enot se nahaja med minimalno vrednostjo in povprečnim odstopanjem v negativni smeri, med povprečnim odstopanjem v negativni in pozitivni smeri in med povprečnim odstopanjem v pozitivni smeri in maksimalno vrednostjo. Za lažjo predstavitev je klasifikator prikazan na sliki 6.6.



Slika 6.6: Klasifikator mesečnih podatkov.

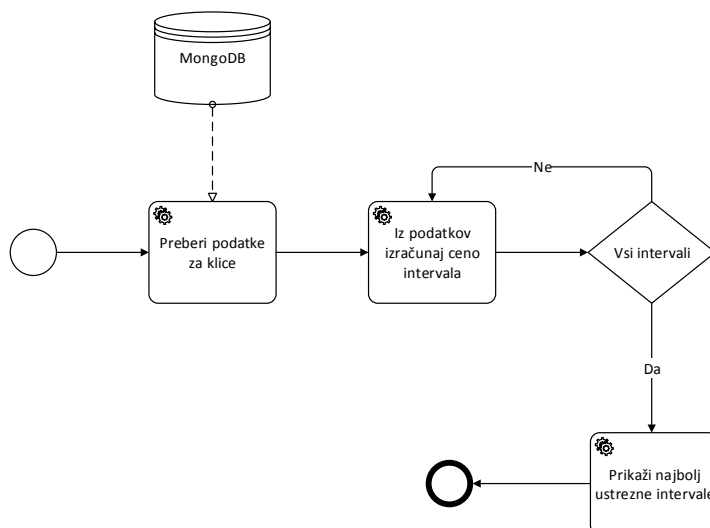
S pomočjo klasifikatorjev simuliramo porabo enot za drugo polovico meseca. Preštejejo se enote simulirane porabe in dejanske porabe za drugo polovico meseca. Enote se primerjajo med seboj in izračunajo se procenti odstopanja. V rezultat se zapišejo le tisti procenti, ki so večji od meje zapisane v nastavitveni datoteki in število enot presega minimalno mejo prav tako zapisano v nastavitveni datoteki. Zapišejo se le odstopanja v negativni smeri. Odstopanja v pozitivni smeri nimajo pomena pri odkrivanju zmanjšane porabe. Kot končni rezultat se vrne besedilo, ki pove za koliko procentov odstopajo posamezne enote pri simulirani porabi v primerjavi z

dejansko.

Obračunski interval

Uporabniki imajo različne navade klicanja. Nekateri imajo daljše pogovore in manj pogoste, drugi imajo krajše in bolj pogoste, tretji spet daljše in bolj pogoste. Namen procesa je ugotoviti kateri obračunski interval je najbolj primeren za uporabnika glede na njegov način klicanja. Uporabniku želimo pokazati graf z vrednostmi obračunskih intervalov. Na grafu (slika 6.9) je označen obračunski interval mobilne storitve uporabnika. Uporabnik tako lahko presodi ali je njegov obračunski interval primeren in kateri bi bil boljši.

Izračun cene klica glede na obračunski interval nismo mogli vključiti v sam proces rangiranja storitev, zato smo se odločili za ločen proces prikazan na sliki 6.7. Pri procesu rangiranja storitev namreč uporabljamo mesečno povprečje klicev, s povprečjem pa izgubimo podatek o načinu klicanja.



Slika 6.7: Proces za izbiro najustreznejših intervalov.

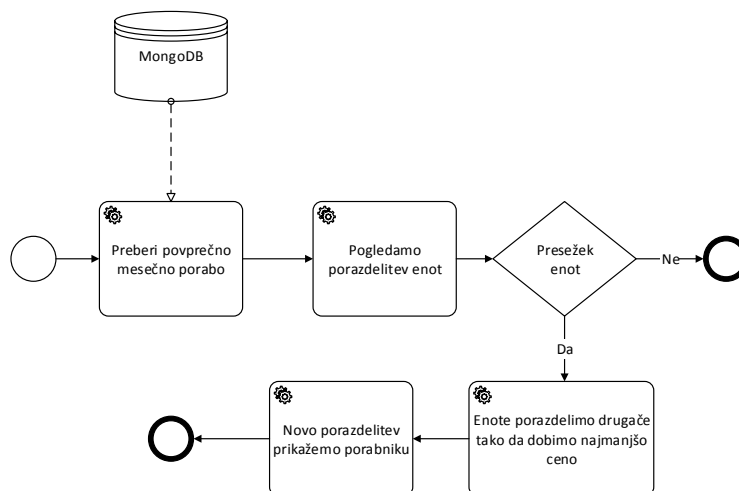
Delovanje V nastavitveni datoteki je podatek o številu odhodnih klicev, nad katerimi bomo izračunali ceno obračunskih intervalov. Izbere se na ključni časovni obseg podatkov v bazi, iz katerega se preberejo podatki o klicih. Obračunske intervale preberemo iz nastavitvene datoteke. Za posamezen obračunski interval se izračuna cena prej prebranih klicev. Obračunski interval je sestavljen iz števila sekund za prvi interval in števila sekund za ostale intervale. Cena se izračuna tako, da se najprej odštejejo sekunde za prvi interval in se k ceni prišteje vrednost enote. Nato pa se od časa klica odštevajo sekunde za ostale intervale, k ceni pa se prišteva vrednost enote. Vrednost je deljena glede na sekunde intervala (60 sekund je polna vrednost enote). To se ponavlja toliko časa dokler ne porabimo vseh sekund klica. Kot rezultat se vrnejo intervali s pripadajočimi cenami.

Porazdelitev enot

Proces na sliki 6.8 prikazuje optimalno porazdelitev zakupljenih enot. Enote bi se razporedile tako, da bi bila cena ob presežku porabe enot najmanjša. Procesu ni bilo smiselno realizirati, saj ima večina mobilnih storitev z zakupljenimi količinami enako ceno na enoto za vse presežene enote. Pri tistih z različno ceno na enoto, pa je razlika med cenami zelo majhna. Po navadi odstopa le ena cena na enoto.

6.3.2 Spletne storitve

V sistemu uporabljamo dve spletni storitvi, kateri sta namenjeni komunikaciji med mobilno in strežniško aplikacijo. Prva storitev vrne delne podatke o uporabniku. S temi podatki lahko v mobilni aplikaciji ustvarimo prijavno okno, poleg tega pa dobimo potrebne podatke za identifikacijo uporabnika. Identifikacijo uporabimo pri pošiljanju podatkov na spletni strežnik. Za sprejemanje podatkov na strani strežnika pa se uporablja druga storitev. Druga storitev iz mobilne aplikacije prejeme podatke o porabi sredstev. Obe storitvi prejmeta oziroma vračata podatke zapisane v JSON formatu.



Slika 6.8: Postopek optimalne porazdelitev enot.

6.3.3 Spletna stran

Na spletni strani so prikazani vsi podatki, generirani pri prej opisanih procesih in še vsi ostali kateri so pomembni pri svetovanju o najboljši izrabi storitev. Za optimalen prikaz je izris spletne strani prilagojen na velikost zaslona. Prikazani podatki so razdeljeni po posameznih podstraneh (slika 6.9). Poleg prve strani so tu še naslednje podstrani: poraba, analiza, statistika, primerjava in nastavitve. V nadaljevanju bomo opisali vsebino in delovanje spletne strani, ter namen posameznih elementov na strani.

V spletno stran se prijavimo tako, da izberemo uporabnika in vpišemo geslo. Če je prijava uspešna, se nam na prvi strani poleg opisa sistema prikaže trenutna cena porabe. Trenutna cena porabe je sestavljena iz mesečne naročnine, cene dodatnih enot in cene porabljenih enot v tujini. Cena dodatnih enot je cena vseh preseženih enot zakupljenih količin storitve, oziroma cena vseh enot če storitev nima zakupljenih količin. Vse cene se seštejejo v skupno ceno. Trenutna poraba predstavlja tekoči mesec, to velja tudi za ceno enot porabljenih v tujini. Poleg cene je na prvi strani prikazana tudi

Nadzor porabe

[Domov](#)
[Poraba](#)
[Analiza](#)
[Statistika](#)
[Primerjava](#)
[Nastavitve](#)

Tujina

Zaznali smo **45** odstopanj. Število različnih držav je **2**. Skupna cena porabljenih enot je **2,502.38 €**. Povprečna mesečna cena enot je **55.61 €**. Glede na vašo porabo v tujini vam svetujemo da **uporabite opcijo za tujino**.

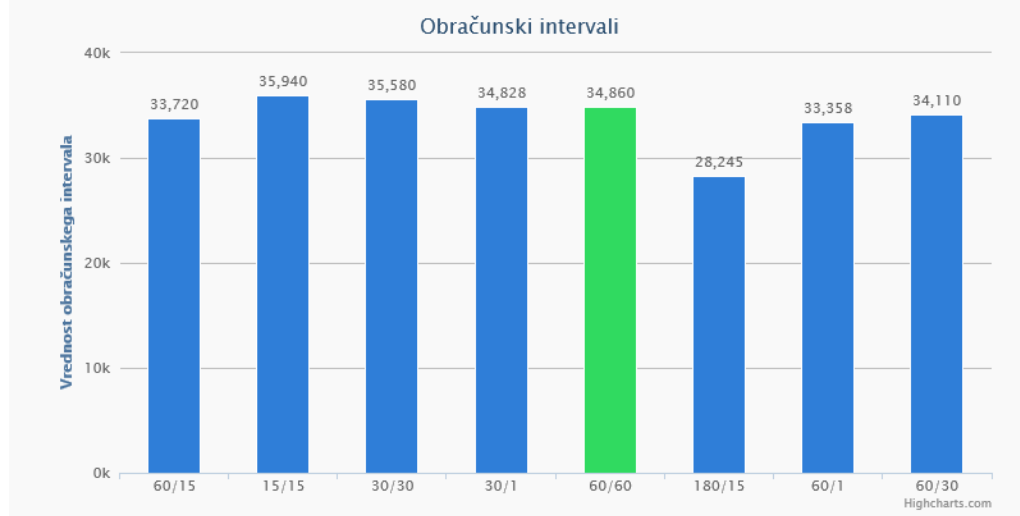
Mesečna statistika

Mesečna statistika se izračuna tako, da se vzame prvo polovico in se izračuna klasifikator. Nato se s klasifikatorjem simulira drugo polovico. Simulirane podatke se nato primerja z dejanskimi.

Vaša mesečna poraba v drugi polovici meseca, ne odstopa od povprečja.

Obračunski intervali

Obračunski interval z najmanjšo ceno je najbolj primeren. Z zeleno je označen vaš trenutni interval.

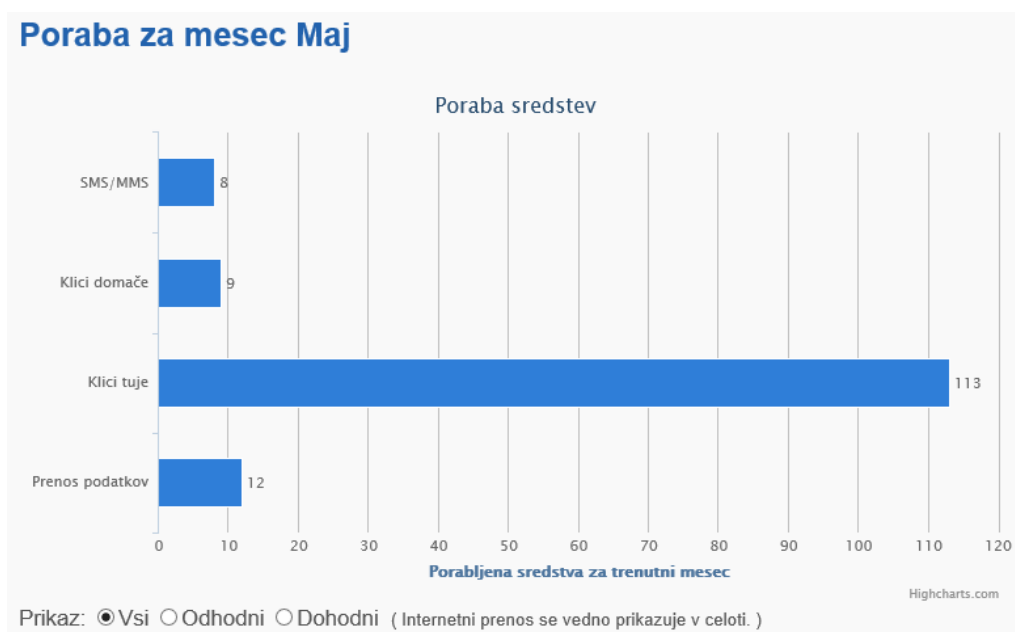


Slika 6.9: Spletna stran inteligentnega sistema.

trenutno izbrana storitev uporabnika. Izpisane so vse podrobnosti storitve kot so zakupljene količine enot, cena posamezne enote, mesečna naročnina, vezava in ostale podrobnosti.

Na podstrani poraba je na prvem grafu (slika 6.10) prikazana poraba posameznih enot (SMS/MMS, klici v domače omrežje, klici v tuje omrežje

in internetni prenos) za tekoči mesec. Na ostalih dveh grafih je prikazana procentna poraba enot po omrežjih. Na prvem grafu je prikazana procentna poraba čez celotno obdobje uporabe sistema. Na drugem grafu pa je prikazana procentna poraba za vsako leto posebej. Uporabnik lahko s pomočjo teh grafov spremlja spremembe porabe enot glede na omrežja.

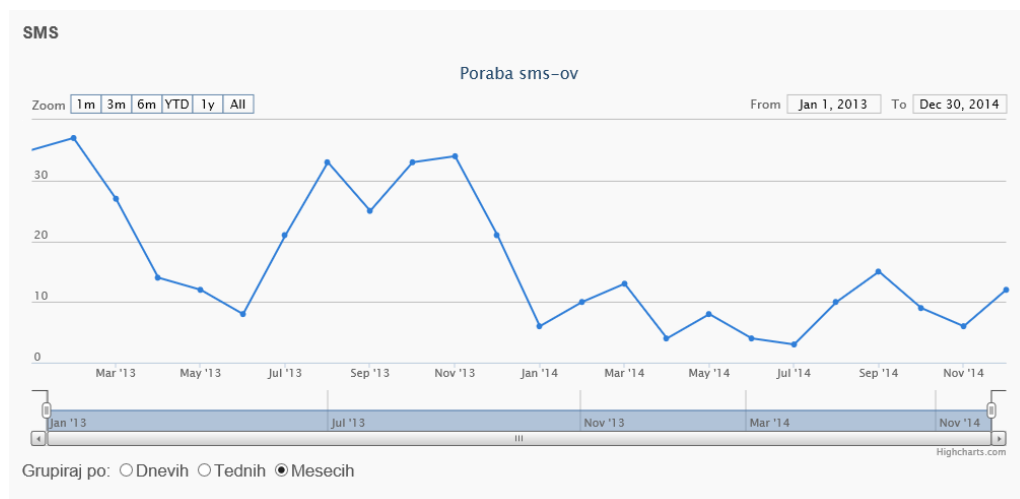


Slika 6.10: Prikaz porabe mobilnih enot za trenutni mesec.

Rezultati procesiranja podatkov so prikazani na podstrani analiza. Tu se prikazujejo vse pridobljene informacije iz prej opisanih procesov, kot so rangiranje storitev (slika 7.1), mesečna statistika, odstopanja in ostali procesi. Poleg podatkov iz procesov je tu še graf, ki razporedi storitve mobilnih ponudnikov glede na njihovo mesečno naročnino. Uporabnik lahko iz porazdelitve mobilnih storitev razbere kakšne vrste (glede na ceno) storitev mobilni ponudnik ponuja.

Podstrani statistika in primerjava na grafih (slika 6.11) prikazujeta porabo posameznih enot čez celotno uporabo sistema. S to razliko, da je na podstrani primerjava na grafu poleg uporabe enot uporabnika, prikazana še

povprečna poraba enot ostalih uporabnikov. Na podstrani statistika lahko na grafu filtriramo prikaz enot na odhodne, dohodne ali vse enote. Na obeh podstraneh lahko enote grupiramo po dnevih, tednih in mesecih.



Slika 6.11: Graf na spletni strani, ki prikazuje statistiko porabe.

Na podstrani nastavitve imamo možnost generiranja podatkov (slika 6.12) za določenega uporabnika. Generiranje podatkov je bolj podrobno opisano v nadaljevanju. Poleg generiranja podatkov imamo tudi možnost urejanja, brisanja in dodajanja mobilnih storitev. Čisto na koncu pa je gumb za zamenjavo uporabnika.

6.4 Mobilna aplikacija

Mobilna aplikacija skrbi za zbiranje in prenos podatkov iz mobilne naprave na strežnik. Aplikacija najprej zbere podatke o klicih, SMS/MMS sporočilih in internetnem prenosu podatkov. Vse podatke nato preko spletne storitve pošlje na strežnik. Ko se pošiljanje podatkov zaključi, aplikacija prikaže spletno stran z informacijami kot so graf porabe, lista najprimernejših storitev itd.

Generiraj podatke

Generira podatke za trenutno izbranega uporabnika.

Opis:

UTEZI_MESECI - Razpored klicev skozi posamezne mesece. Razdelimo jih po procentih (od 0 do 1).

Skupen seštevek je 1.

UTEZI_MESEC - Procent klicev v prvi polovici meseca.

UTEZI_URE - Procent klicev skozi dan.

UTEZI_OMREZJA - Procent klicev v omrežja (mobilna, stacionarna, posebna, voIP).

UTEZ_LASTNO_OMREZJE - Procent klicev v lastno omrežje.

X_ST - Stevilo klicev, SMS, MMS, Podatkov.

KLICI_TRAJANJE_X - Obseg trajanje klica v sekundah.

X_KOLICINA_KB_OBSEG_X - Obseg prenesenih kB.

```
# Splosno
EMSO=1910989500147
LETO=2014
LOKACIJA_X=46.057181
LOKACIJA_Y=14.507354
# Utezi
UTEZI_MESECI=0.08,0.08,0.08,0.08,0.08,0.08,0.1,0.1,0.08,0.08,0.08,0.08,0.08
UTEZI_MESEC=0.5
UTEZI_URE=0.15,0.35,0.35,0.15
UTEZI_OMREZJA=0.8,0.1,0.05,0.05
UTEZ_LASTNO_OMREZJE=0.5
# Kljci
KLICI_ST=24000
KLICI_TRAJANJE_OD=20
KLICI_TRAJANJE_DO=200
# SMS
SMS_ST=24000
# MMS
MMS_ST=2400
# NET
NET_ST=1200
NET_KOLICINA_KB_OBSEG_OD=100
NET_KOLICINA_KB_OBSEG_DO=4096
```

Generiraj

Slika 6.12: Forma za generiranje podatkov.

Delovanje Mobilna aplikacija se najprej poveže na strežnik preko spletne storitve in prenese podatke o uporabnikih. Prikaže se dialog za izbiro uporabnika. Ko uporabnik mobilne aplikacije izbere uporabnika, se zberejo vsi še ne poslani podatki o porabljenih enotah. Najprej se podatki nekoliko prilagodijo. Mobilne številke se skrajšajo do te mere, da ostane le del ki določa za katerega operaterja gre. Podatkom se določi lokacija glede na kraj, kjer so bile enote porabljene. Prilagojeni podatki se skupaj z identifikacijo upo-

rabnika zapakirajo v JSON format. Da si zapomnimo katere podatke smo že prenesli, se v globalno spremenljivko zapiše datum pošiljanja. Ob naslednjem pošiljanju pošljemo le novejšo podatke od prej zapisnega datuma. Aplikacija zapakirane podatke preko spletne storitve pošlje na strežnik.

Ko je pošiljanje končano, se ustvari spletni pregledovalnik v katerem se naloži spletna stran s povratnimi informacijami. Uporabniku se ni potrebno še enkrat prijavljati v sistem na spletni strani. Spletna stran se prilagodi za prikaz na mobilni napravi. Uporabnik ima dostop do vseh zbranih informacij.

6.5 Generator podatkov

Za potrebe testiranja in evalvacije sistema smo izdelali generator podatkov (izvorna koda na naslovu <https://bitbucket.org/Klavdij/randomizer>). Generator generira podatke s pomočjo pravil zapisanih v nastavitveni datoteki. V nastavitveni datoteki so parametri (slika 6.12) razdeljeni v več kategorij. V splošni kategoriji je zapisano za katerega uporabnika se generirajo podatki, za katero leto ter lokacija podatkov. V kategoriji uteži nastavljammo kako so podatki razdeljeni po mesecih, kako so razdeljeni v polovicah meseca in kako po četrtinah dneva. Prav tako nastavljammo procentno razporeditev po omrežjih ter kolikšen procent od tega je v lastnem omrežju. Ostale kategorije določajo še število klicev, SMS/MMS sporočil in število prenosov mobilnega interneta. Pri klicih lahko določimo obseg trajanja in pri prenosu podatkov obseg količine enega prenosa. Po koncu generiranja se podatki shranijo v bazo.

Poglavje 7

Podrobnosti implementacije prototipa inteligentnega sistema

V prejšnjem poglavju smo opisali posamezne dele inteligentnega sistema in njihovo delovanje. Sedaj se bomo bolj podrobno osredotočili na implementacijo prototipa inteligentnega sistema s pomočjo tehnologij opisanih v prejšnjih poglavjih.

7.1 Podatkovna baza

Glavni del uporabniškega profila predstavljajo zbirke podatkov v podatkovni bazi MongoDB. Poleg podatkov, ki predstavljajo uporabniški profil, se v bazo shranjujejo tudi vsi ostali podatki potrebni za delovanje celotnega sistema. Celoten diagram baze je prikazan na sliki 6.1.

Strežniška aplikacija za izvajanje operacij nad bazo uporablja Javanski gonilnik. Glavni namen gonilnika je komunikacija med bazo in aplikacijo, branje in pisanje podatkov iz baze ter preoblikovanje podatkov v zapis JSON. Za lažje izvajanje operacij nad bazo smo sprogramirali razred Baza, ki skrbi za avtomatsko komunikacijo med bazo in procesi, ter ima implementirane

metode za lažje shranjevanje in vračanje rezultatov iz baze. Rezultati se vrnejo v obliki objektov. Objekti razširjajo razred *Entiteta*, ki olajša pretvorbo podatkov iz baze v objekt in obratno. *Entiteta* definira metodo, katera vsebino razreda pretvori v zapis JSON. Prav tako pa ima prilagojen konstruktor za nastavljanje atributov razreda iz zapisa JSON.

7.2 Strežniška aplikacija

Strežniška aplikacija je v celoti implementirana v programskem jeziku Java. Sestavljena je iz treh večjih sklopov. Ti sklopi so procesi, spletna stran in spletne storitve. Spletne storitve so narejene s pomočjo knjižnice Jersey. Za dinamični prikaz spletne strani se uporablja JSP. Izvorna koda strežniške aplikacije se nahaja na naslovu <https://bitbucket.org/Klavdij/streznik>. V podpoglavjih je bolj podroben opis posameznih sklopov strežniške aplikacije.

7.2.1 Proces

Glavni del strežniške aplikacije predstavljajo procesi. Procese, kot je rangiranje storitev, mesečna statistika in ostale, smo bolj podrobno opisali v prejšnjem poglavju. Tudi vsaka podstran na spletni strani ima svoj proces. Posamezen proces je sprogramiran tako, da nima direktne povezave z ostalimi procesi, če je to mogoče. Proces si posredujejo podatke preko podatkovne baze. Namen takšnega pristopa je lažje spreminjanje procesov brez vpliva na ostale. Tudi dodajanje in odstranjevanje novih procesov je bolj enostavno.

Procesi na podstrane spletne strani skrbijo za operacije kot so pretvorba in zbiranje podatkov iz baze, prilagajanje oblike izpisa informacij, dinamično spreminjanje seznamov in druge operacije potrebne za dinamični prikaz informacij na spletni strani.

Definirali smo tudi standarden razred *Rezultat*, kateri definira metode za shranjevanje in vračanje rezultatov. Rezultati se shranjujejo kot ključ in vrednost. S takim pristopom smo poenotili in poenostavili prenos rezultatov iz procesov na spletno stran.

Poleg razreda Rezultat imamo še razred Lokacija, ki preko Googleove storitve določi, katera država se nahaja na neki lokaciji. Razred Operator pa poskrbi za lažje določanje operaterja številki in obratno. Za nalaganje nastavitev iz nastavitvene datoteke skrbi razred Nastavitve. Ti razredi nam služijo kot orodja pri operacijah procesov.

7.2.2 Spletna stran

Filtri
Naročnina: Vsi Brez naročnine Z naročnino

Operaterji: Bob Debitel Izimobil Mobitel Simobil Tusmobil

Prikaz: Skrčeno Vse

Najbolj primerne storitve po pregledu
čez celotno obdobje.

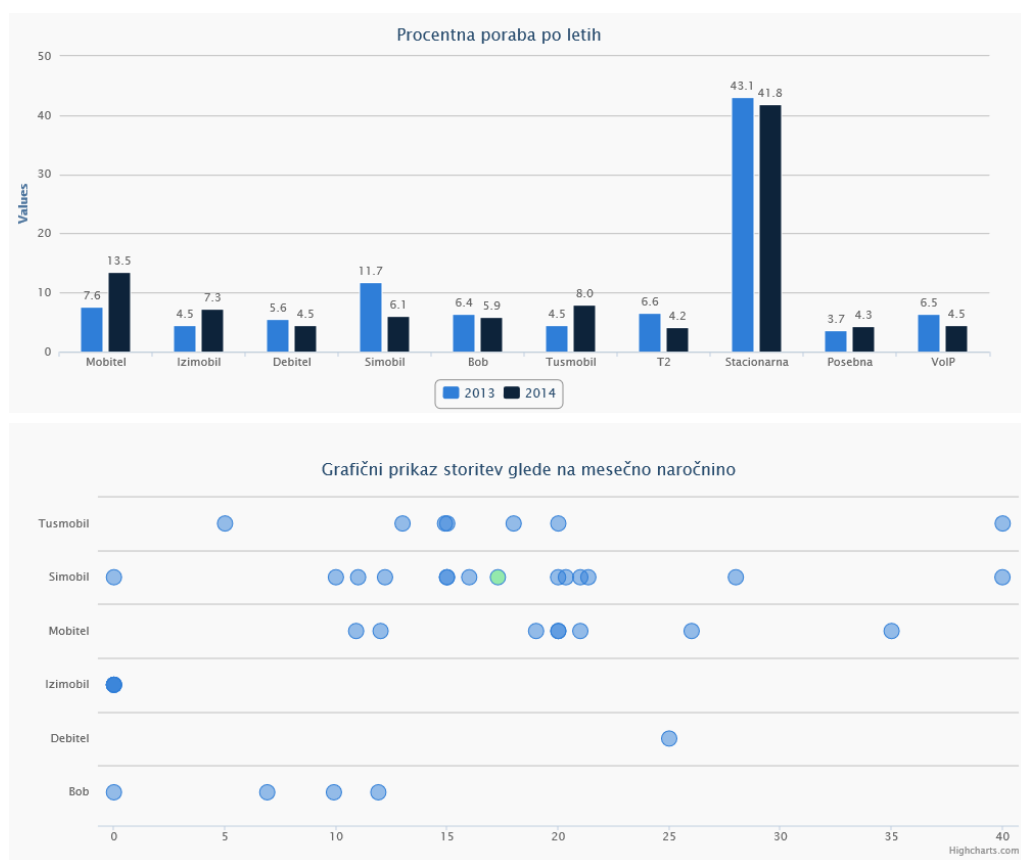
Najbolj primerne storitve od največjega
odstopanju (2. 2014) naprej.

Rank	Service	Rank	Service
1.	bob 4 cente	1.	bob 4 cente
2.	iziklepet	2.	iziklepet
3.	izi008	3.	izi008
4.	Simpl	4.	Free 5
5.	izidoma	5.	Simpl
...		...	
18.	Free 15	18.	Free 15
19.	Osnovni m	19.	Osnovni m
20.	Orto u nulo 201	20.	Orto u nulo 201
21.	Orto u nulo	21.	Mingl 17
22.	Orto startastik	22.	Itak dzabest
...		...	

Slika 7.1: Prikaz seznama rangiranih storitev in opcij filtriranja na spletni strani.

Na spletni strani se prikazujejo različni podatki in informacije. Informacije, kot so število porabljenih enot, število klicev v posamezna omrežja,

seznam rangiranih storitev (slika 7.1), poraba enot skozi daljše časovno obdobje, so prikazane v različnih oblikah. Najbolj pogosta oblika prikaza so grafi (slika 7.2). Grafe smo prikazali s pomočjo knjižnic Highcharts in jQuery. Za različne tipe informacij smo uporabili različne tipe grafov. Za prikaz podatkov v časovnem odboju smo uporabili linijski graf, za prikaz podatkov odvisnosti od spremenljivk smo uporabili stolpičast graf (slika 7.2) in za prikaz podatkov v treh dimenzijah graf z mehurčki (slika 7.2).



Slika 7.2: Zgoraj graf z mehurčki za prikaz porazdelitve mobilnih storitev. Spodaj stolpičast graf za prikaz porabe enot po mobilnih omrežjih.

Podatke za prikaz na grafu smo na strežnika zapisali v JSON format in jih poslali odjemalcu. Na strani odjemalca smo jih razčlenili in jih posredovali knjižnici HighChart. Poleg posredovanih podatkov smo grafu nastavili še

ostale metrike kot so možnost prikaza legende, označbe stolpcev in ostalih vizualnih elementov.

7.2.3 Spletne storitve

Za implementacijo spletnih storitev smo uporabili knjižnico Jersey. Za definiranje storitev se določi naslov oziroma URI in funkcijo, ki se kliče na tem naslovu ob izbrani HTTP metodi (npr. GET, POST). Kot smo že zapisali, spletna storitev skrbi za prenos podatkov iz mobilne v strežniško aplikacijo in obratno. Podatki se pred prenašanjem zapakirajo v zapis JSON. Za generiranje in razčlenitev zapisa JSON na strani strežnika smo uporabili Goo-glovo knjižnico Gson. Na platformi Android pa je ta knjižnica že privzeto vključena. Za lažje shranjevanje in branje podatkov iz baze, spletni storitvi tako kot ostali procesi uporabljata razred Baza.

7.3 Mobilna aplikacija

Mobilna aplikacija je implementirana zelo preprosto. Vsebuje le eno aktivnost in dve asinhroni opravili. Asinhroni opravili poskrbita za povezavo s spletno storitvijo na strežniku. Prvo opravilo preko spletne storitve pridobi podatke o uporabnikih, katere aktivnost prikaže v dialogu za izbiro. Drugo asinhrono opravilo pa pošlje zbrane podatke na strežnik prav tako preko spletne storitve. Aktivnost od odprtju aplikacije zbere vse podatke o porabljenih enotah. To so podatki o zgodovini klicev, poslanih in prejetih SMS/MMS sporočilih in količinah prenesenih podatkov preko mobilnega interneta. Aktivnost podatke o klicih in sporočilih zbere s pomočjo ponudnikov vsebin. Število prenesenih podatkov preko mobilnega interneta pa preko statistike prometa. Vsi zbrani podatki se zapakirajo v zapis JSON in posredujejo prej opisanemu asinhronemu opravilu. Ko so podatki poslani, aktivnost prikaže spletno stran z informacijami v spletnem pregledovalniku (WebView). Celotna izvorna koda mobilne aplikacije se nahaja na naslovu <https://bitbucket.org/Klavdij/mobilna-aplikacija>.

Poglavje 8

Evalvacija predlagane rešitve

Delovanje sistema smo preverili v treh fazah. V prvi fazi smo s pomočjo generatorja podatkov izdelali različne profile uporabnikov. Vsak profil se je osredotočil na posamezne funkcionalnosti sistema. Tako smo preverili pravilno delovanje sistema in poskušali odkriti pogoje, kjer sistem vrača slabe rezultate. Namen prve faze testiranja je bil, odstranjevanje napak in raziskati možne nadgradnje ter izboljšave sistema. Na koncu prve faze testiranja smo sistem ocenili glede na hitrost delovanja in število napak.

V drugi fazi testiranja smo se osredotočili na to, ali nam sistem vrača smiselne in za uporabnika primerne rezultate. Takšno preverjanje je nekoliko bolj kompleksno. Da bi čim bolj izmerili pravilnost rezultatov, smo določili metrike v obliki prihranjenega denarja s strani uporabnika. Za posamezen profil izračunamo, kakšen prihranek ima uporabnik glede na podane nasvete o uporabi mobilnih storitev. Iz vseh podanih informacij ni mogoče direktno ugotoviti denarnega prihranka, zato smo tu morali uporabiti nekoliko kompleksnejše pristope.

Testiranje v drugi fazi je potekalo tako, da smo najprej generirali specifične podatke za posameznega uporabnika. Zatem pa pognali sistem, da je obdelal podatke in izluščil informacije. Nato smo dobljene nasvete o uporabi mobilnih storitev primerjali s podanimi podatki in načinom porabe ter ugotavljali ali smo dobili smiselne rezultate. Smiselnost rezultatov smo definirali

s prihranjenim denarjem.

V zadnji tretji fazi smo celoten sistem ocenili s pomočjo SWOT analize. SWOT analiza je sistem ocenjevanja nekega produkta, prostora, sektorja ali osebe. Analiza se osredotoča na štiri aspekte in sicer prednosti, slabosti, priložnosti ter nevarnosti. Namen analize je bil oceniti naš predlog z vseh štirih aspektov in ga primerjati s konkurenčnimi pristopi. Z oceno iz analize SWOT smo dobili celotno sliko smiselnosti in primernosti našega predloga optimizacije stroškovnega vidika mobilne uporabniške izkušnje.

Podatke o mobilnih storitvah in cenah storitev za tujino smo dobili na spletnih straneh mobilnih operaterjev. Podatke smo zbrali v mesecu oktobru leta 2013. Zbrali smo podatke o osemintridesetih mobilnih storitvah. Mobilni operaterji storitev so: Simobil, Mobitel, Bob, Tušmobil, Izimobil, Debitel. Cene storitev za tujino smo vnesli v bazo za štiri sosednje države.

8.1 Prva faza - splošno testiranje

Prvo fazo testiranja smo začeli izvajali že med samim razvojem. Generali smo različne podatke, katere smo uporabili za testiranje posameznih sklopov. Generirani podatki so bili specifično usmerjeni in niso preverjali učinkovitosti sistema. Namenjeni so bili le odkrivanju napak in nepravilnostim v delovanju testiranih metod.

Med razvojem sistema smo odkrili le manjše napake. Do večine teh manjših napak je prišlo zaradi klica na prazne vrednosti, nepredvidenih podatkov, sintaktičnih napak itd. Vse napake smo sproti odstranjevali. Med razvojem se nismo osredotočali na hitrost delovanja.

Ko smo sistem implementirali do konca, smo ponovno testirali vse aspekte delovanja. Med končnim testiranjem nismo opazili nobenih napak pri delovanju sistema. Ne izključujemo pa možnosti pojavitve napak pri uporabi sistema. Povprečen odzivni čas prikaza obdelanih podatkov je okrog 200 ms. Hitrost prikaza spletne strani je tako zadovoljiva. Tudi med procesiranjem na novo generiranih podatkov ni bilo opaziti večjih sprememb v hitrosti pri-

kaza spletne strani. Čas prikaza se poveča za približno trikrat. Če vzamemo povprečje različnih dostopov dobimo približen čas 400 ms. Ko imamo 10 hkratnih dostopov, je odzivni čas še vedno v mejah normale. Čas smo izmerili tako da smo naredili deset dostopov, odstranili maksimalno in minimalno vrednost ter izračunali povprečje. Zaznali smo dve odstopanji odzivnosti sistema. Prvo nekoliko variabilno odstopanje v hitrosti delovanja se pojavi pri procesiranju podatkov, kjer imamo klic na Googlovo storitev za pridobivanje države iz lokacije. Takšno odstopanje se opazi le takrat, ko imamo veliko različnih lokacij. Drugo nekoliko večje odstopanje v hitrosti delovanja pa se pokaže pri prikazovanju statistike porabe in pri prikazu primerjave porabe uporabnika s povprečjem ostalih uporabnikov, ko imamo zelo veliko količino podatkov. Gledano kot celota, lahko rečemo da je sistem stabilen in dovolj dobro odziven. Ob povečani uporabi sistema bi lahko bazo horizontalno razširili čez več strežnikov.

8.2 Druga faza - testiranje učinkovitosti

V drugi fazi smo, za potrebe testiranja učinkovitosti sistema, izdelali tri različne profile uporabnikov. Želeli smo pokriti vse vidike sistema, kateri se nanašajo na svetovanje uporabniku o boljšem izkoristku mobilnih storitev. Vsak profil se osredotoči na dve funkcionalnosti sistema. Profile smo zgradili tako, da smo z generatorjem podatkov generali takšne podatke, da so predstavljali različno obnašanje uporabnikov pri uporabi mobilnih storitev. Dodatno smo iz šestih profilov, ki uporabljajo storitev iz različnih omrežjih, izračunali povprečni prihranek, pri čemer imajo vsi profili enako porabo.

8.2.1 Profili uporabnikov

Prvi profil se osredotoča na delovanje razvrščanja storitev po primernosti. Prav tako pa preveri porabo storitev v tujini, kadar veliko potujemo po različnih tujih državah. Profilu smo najprej določili uporabo storitve v Simobilovem omrežju. Nato smo podatke generirali v razmerju 90:10 glede

na porabo v lastnem oziroma tujem omrežju. Pri čemer je bilo 10 % podatkov porabljenih v lastnem omrežju in 90 % v tujem. Uteži po omrežjih smo nastavili tako, da je bilo 80 % klicev opravljenih v mobilnih omrežjih. Podatke za tujino smo generirali za deset različnih lokacij v tujini, pri čemer je opravljenih vsaj 15 klicev na dan.

Pri drugem profilu smo se osredotočili na odkrivanje odstopanj in porabo v tujini, kadar gre za veliko obiskov ampak enako državo. Določili smo tudi, da so klici bolj pogosti toda krajši. Povprečno trajanje klica je bilo med eno in dvema minutama. Namen tega je, primerjati ocene obračunskih intervalov z obračunskimi intervali tretjega profila, ki ima redkejšo toda daljše klice. Podatke za ta profil smo generirali, tako da se v prvi polovici leta večino enot porabi v omrežju Simobil, nato pa se v drugi polovici leta večina enot porabi v drugih tujih omrežjih. Za generiranje podatkov smo uporabili podobne metrike kot pri prvem profilu, le da smo vmes zamenjali razmerje pri porabi glede na lastno oziroma tuje omrežje. Profil uporabnika je imel izbrano mobilno storitev Orto u Nulo 201 v omrežju Simobil. Za tujino smo podatke generirali na desetih različnih lokacijah v isti državi z 20 klici pri posamezni lokaciji.

Pri tretjem pa smo preverjali mesečno statistiko, ter obračunske intervale. Število porabljenih enot smo v mesecu razdelili v razmerju 90:10. Večji delež enot je bil porabljen v prvi polovici meseca. Za obračunske intervale, kot smo že omenili, smo nastavili redkejšo toda daljše klice. Količino enot in porazdeljenost po omrežjih smo nastavili enako kot pri drugem profilu z namenom, dobiti podobne pogoje za primerjanje obračunskih intervalov.

Za namene dodatnega testiranja smo izdelali šest različnih profilov. Vsi profili so imeli enako porabo. Profili so porabili približno 1000 minut odhodnih klicev, 1000 poslanih SMS/MMS sporočil, 200 mB prenesenih podatkov med uporabo mobilnega interneta. Poraba enot je razdeljena v razmerju 50:50, lastno oz. tuje omrežje. Profilom smo spreminjali le ponudnika storitev. Profil je uporabljal storitev ponudnika, ki ima najnižji mesečni strošek. Prihranek smo izračunali kot razliko stroška uporabe izbrane storitve in naj-

primernejše storitve s seznama rangiranih storitev. Povprečni prihranek pa predstavlja povprečje prihrankov vseh profilov.

8.2.2 Testiranje in rezultati

Drugi del testiranja je obsegal preverjanje učinkovitosti delovanja in smiselnost dobljenih rezultatov. Uporabili smo prej opisane profile uporabnikov. Dobljene rezultate smo primerjali z vnesenimi podatki in ugotavljali ali so povratne informacije smiselne in kakšne prihranke smo dobili.

Prvi profil uporabnika je imel izbrano mobilno storitev Orto u Nulo 201, v omrežju Simobil. Ugotovili smo, da nam je sistem pravilno razvrstil mobilne storitve glede na generirane podatke. Storitve drugih operaterjev in tiste storitve, ki imajo več zakupljenih enot za druga mobilna omrežja so bila razvrščena višje na lestvici. Porabljene enote profila niso presegale zakupljenih količin izbrane mobilne storitve tako, da je mesečni strošek enak mesečni naročnini mobilne storitve. Mesečna naročnina znaša 17,28 €, če sedaj primerjamo ta znesek z najbolje uvrščeno mobilno storitvijo na lestvici, ki znaša 6 €, dobimo prihranek v vrednosti 11,28 €. Kar je za približno 65 % nižji mesečni strošek.

Za porabljene enote v tujini pri prvem profilu uporabnika, nam sistem svetuje, naj uporabimo opcijo za tujino. V tem primeru, kjer uporabnik obišče veliko različnih držav, je to dobra odločitev. Če vzamemo kot primer klicne opcije za tujino pri Simobilu, ki dnevno stane 2,99 € in smo v tujini 10 dni, nas stane 29,9 €. Če v teh desetih dneh opravimo več kot 10 klicev dnevno, je to po normalni tarifi (0,2928 €) brez dodatne opcije več kot 30 €. Ko uporabnik preseže mejo desetih klicev, mu sistem svetuje, naj uporabi opcijo za tujino.

Pri drugem profilu uporabnika je sistem pravilno zaznal datum odstopanja. Pri razvrstitvi storitev je bila razlika opazna. V splošnem so bile storitve operaterjev iz drugega omrežja na seznamu, kateri prikazuje razvrstitev od največjega odstopanja naprej, višje na lestvici. Prav tako so višje na seznamu storitve, ki ponujajo več zakupljenih enot. Če pogledamo izbrano

storitev uporabnika Orto u Nulo 201 je na splošnem seznamu na 15. mestu, na seznamu od največjega odstopanja naprej, pa je šele na 20 mestu. Na 15. mestu je mobilna storitev Neskončni m, ki ponuja neskončno enot pri klicih v tuja in stacionarna omrežja.

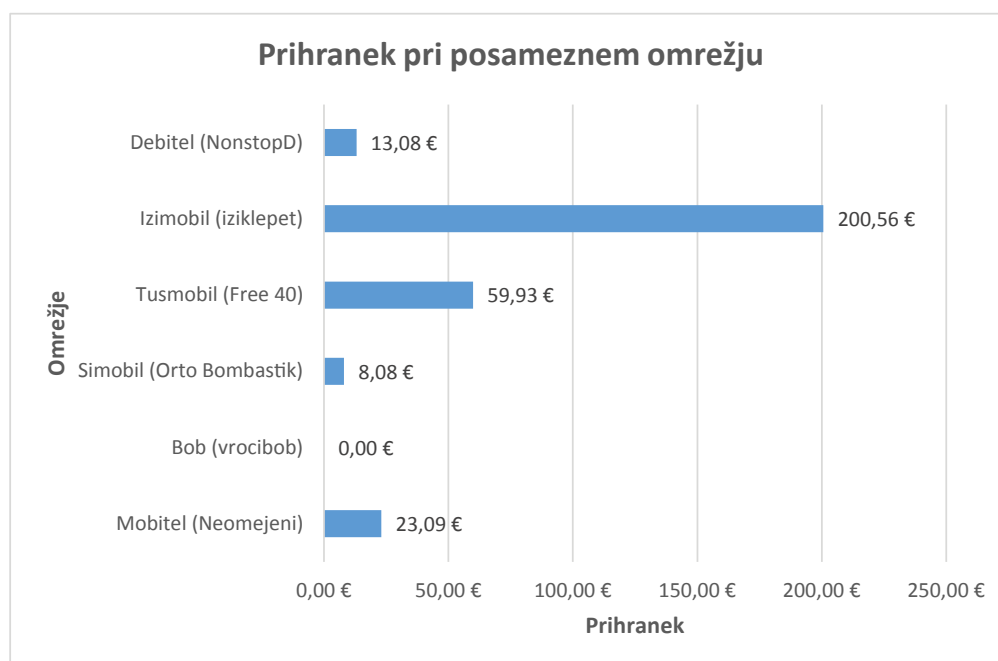
Sistem nam za porabo v tujini pri drugem profilu uporabnika svetuje, naj uporabimo opcijo uporabe tujega operaterja. Ker so enote porabljene zgolj v eni državi, lahko sklepamo, da uporabnik veliko uporablja enote tudi za uporabnike znotraj te države, ne le za uporabnike v domači državi. Prav tako imajo ponudniki mobilnih storitev, že privzeto vklopljene cenejše tarife za klice v tujino.

Mesečna statistika pri tretjem primeru zazna zmanjšano porabo enot v drugi polovici meseca. Prikaže nam procent zmanjšanja in število porabljenih enot pri simulirani porabi. Poleg tega pa seveda tudi nasvet naj uporabimo storitve z enotami, pri katerih je bila zaznana manjša poraba. Uporabnik lahko iz informacije o številu porabljenih enot pri simulirani porabi, hitro ugotovi katere mobilne storitve so za njega bolj primerne. V našem primeru smo dobili informacijo o zmanjšani porabi SMS/MMS enot za 160 %, kar je nanese na več kot 1200 enot pri simulirani porabi. Za takšen profil uporabnika so bolj primerne storitve z več kot 1000 enotami SMS/MMS. Če bi uporabljal trenutno storitev Orto u Nulo 201, bi imel pri simulirani porabi za približno 30 € več stroškov. Seveda teh stroškov uporabnik dejansko nima, toda mora paziti vsak mesec, da jih ne preseže.

Po primerjanju grafa obračunskih intervalov za drug in tretji profil smo ugotovili naslednje razlike. Pri daljših klicih obračunski intervali s krajšim časom dobijo manj točk oziroma so manj primerni. Če pa imamo krajše klice potem so bolj primerni obračunski intervali s kratkim tako prvim intervalom kot ostalimi intervali. Prvi interval nima takšnega vpliva kot ga imajo ostali intervali pri oceni obračunskega intervala. Krajši ostali intervali so bolj primerni za krajše klice, daljši pa za daljše klice.

Na grafu 8.1 vidimo prihranke profilov, kateri imajo enako porabo toda različnega ponudnika storitev. Namen testiranja je bil, ugotoviti povprečni

prihranek. Izračunali smo, da je povprečni prihranek 50,79 €. Najbolj primerno omrežje je Bob in storitev vrocibob, kar je tudi razvidno iz grafa 8.1. Bolj kot je primerna storitev manjši je prihranek. Bolj zanimivo dejstvo kot povprečni prihranek je razlika v prihrankih. Odstopanja so zelo velika. Ugotovili smo, da ima lahko že zelo majhna sprememba zelo velik vpliv. Pametni inteligentni pomočnik tu odigra pomembno vlogo. Takšen vpliv na prihranek je težko opaziti oziroma napovedati brez uporabe inteligentnega pomočnika.



Slika 8.1: Prikaz prihranjenega denarja glede na izbrano storitev omrežja in najboljšo storitev pri enaki porabi enot.

Rezultate druge faze testiranja lahko označimo kot primerne. Sistem je pravilno svetoval uporabniku glede na njegovo porabo enot. Uporabnik lahko s pomočjo sistema zmanjša mesečni strošek oziroma opazi odstopanja pri porabi enot ter si s tem pomaga pri izbiri boljše mobilne storitve oziroma opcije. Delovanje sistema lahko ocenimo kot učinkovito in primerno.

8.3 Tretja faza - SWOT analiza

S pomočjo aspektov prednosti, slabosti, priložnosti in nevarnosti, SWOT analize smo v tretji, zadnji fazi testiranja ocenili sistem kot celoto. Prva dva aspekta SWOT analize se nanašata na notranje dejavnike, druga dva na zunanje dejavnike. Notranji dejavniki se nanašajo na dobre in slabe lastnosti naše rešitve. To so dejavniki na katere lahko vplivamo. Pri zunanjih dejavnikih pa se analiza osredotoča na dejavnike okolja na katere nimamo vpliva. V vsakem podpoglavju smo opisali enega iz med aspektov SWOT analize. Aspekt prednosti opisuje elemente oziroma funkcionalnosti sistema, katere so boljše kot pri sorodnih rešitvah. Pri slabostih se osredotočimo na pomanjkljivosti sistema in predstavimo izboljšave. Pri aspektih priložnosti in nevarnosti prikažemo dobre in slabe lastnosti sorodnih rešitev in možne vplive okolja, ki lahko škodujejo oziroma pripomorejo k uporabi naše rešitve.

8.3.1 Prednosti

Naša rešitev ponuja uporabniku podrobno analizo porabe. Uporabnik lahko nastavlja različne parametre prikaza. Poraba je razdeljena na različna časovna obdobja in enote mobilne storitve. Večina konkurenčnih rešitev ponuja zgolj osnoven prikaz statistike in porabe. Naš sistem predstavi analizo porabe in ostale informacije na različne grafične načine. Vsaka informacija se prikaže na način, ki je najbolj primeren za takšen tip informacij. Za prikaz uporabljamo različne tipe grafov, sezname, razpredelnice in ostale grafične elemente. Uporabnik dobi najbolj optimalen prikaz informacij in podatkov.

Najpomembnejša prednost pred konkurenčnimi rešitvami je pametno svetovanje. Sistem uporabniku svetuje o izbiri najboljše storitve, porabi enot v tujini, mesečni porabi itd. Uporabnik lahko s pomočjo našega sistema bistveno zmanjša mesečne stroške uporabe mobilnih storitev. Poleg tega pa dobi koristne informacije o svojem načinu porabe mobilnih enot, ki mu pomagajo pri izbiri boljše mobilne storitve oziroma opcije.

Naš sistem je zelo prilagodljiv in po potrebi mu lahko dodamo nove funk-

cionalnosti. Ker je izdelan po principu odjemalec-strežnik, lahko uporabnik dostopa do informacij preko spletne strani iz katere koli naprave. Trenutno je omejeno le zbiranje podatkov o porabljenih enotah mobilne storitve. Toda zaradi same arhitekture sistema lahko zelo enostavno implementiramo še mobilne aplikacije za ostale mobilne operacijske sisteme, ne le za Androida.

8.3.2 Slabosti

Največja pomanjkljivost sistema je slabše delovanje ob majhnem številu podatkov. To je tudi pomanjkljivost uporabniških profilov, kateri se zgradijo zgolj iz podatkov o uporabniku. Sistem bi lahko izboljšali tako, da bi uporabili drug način izdelave uporabniških profilov. Pri drugem načinu izdelave profil gradimo glede na medsebojno filtriranje z ostalimi profili. Tako bi lahko dobili boljše rezultate, čeprav bi imeli le nekaj mesečnih podatkov o porabi enot. S stališča uporabnika je ta pomanjkljivost zelo kritična. Napačne informacije lahko uporabniku škodujejo in v takšnem primeru je bolje, da ne prejme nobenih informacij.

Druga manjša pomanjkljivost se pojavi pri razvrščanju storitev. Pri razvrščanju se ne upošteva, da nekatere storitve na račun dražje mesečne naročnine ponujajo nakup subvencionirane mobilne naprave ali izrabo drugih ugodnosti. To pomanjkljivost smo omilili tako, da smo uporabniku ponudili možnost filtriranja prikaza storitev (7.1). Uporabnik lahko tako razvrsti le tiste storitve, ki so za njega primerne.

Ne toliko kot pomanjkljivost ampak bolj izboljšava sistema, bi bila uporaba ugotovitev iz mesečne statistike in obračunskih intervalov pri razvrščanju storitev. Storitve bi lahko ponovno razvrstili glede na ugotovitve pri mesečni statistiki. Za izračun cene porabe enot bi uporabili simulirane podatke iz mesečne statistike. Prav tako pa bi lahko upoštevali obračunske intervale pri izračunu cene. Toda uporabiti bi morali drug pristop, pri čemer bi izračunali mesečne cene in nato povprečje cen namesto povprečne porabe enot.

Zgornji dve pomanjkljivosti sistema sta lahko pri manj večjih uporabnikih zelo problematični. Takšni uporabniki si ne bodo znali pomagati s

posredovanimi informacijami. In za njih so te funkcionalnosti skorajda neuporabne. Posredovane informacije bi morali zelo poenostaviti in jih narediti bolj jasne in direktne. Konkurenčni izdelki tu ponujajo zelo preproste ter jasne povratne informacije.

Med testiranjem smo opazili padec hitrosti pri prikazu statistike in pri prikazu primerjavi porabe med uporabnikom in povprečjem porabe ostalih uporabnikov. Največja težava tu je procesiranje podatkov na strani odjemalca. Strežnik pošlje gole podatke odjemalcu, ta pa jih procesira, preoblikuje in pokaže na grafu. Prednost takega pristopa je hitro in neomejeno spreminjanje časovne enote prikaza podatkov. Celoten proces bi lahko izboljšali tako, da bi uporabili že agregirane podatke uporabnika oziroma jih na novo agregirali glede na časovno enoto in obdobje. Naloga odjemalca pa bi bila zgolj prikazati podatke na grafu oziroma poslati nov zahtevek na strežnik ob spremembi časovne enote oziroma obdobja.

Ena izmed slabosti sistema je tudi napačna klasifikacija podatkov ob prenosu številke. Sistem iz mobilne številke ugotovi v katerem omrežju se nahaja. Če ima uporabnik veliko kontaktov s prenesenimi številkami, potem dobi napačne informacije. Rešitev tega problema je v dostopu do baze, kjer so shranjeni vsi podatki o mobilnih številkah.

8.3.3 Priložnosti in nevarnosti

Konkurenčne aplikacije trenutno še ne ponujajo naprednejšega pametnega svetovanja o izrabi mobilnih storitev. Trenutno dostopne aplikacije ponujajo zgolj enostavne rešitve za spremljanje porabe in prikazovanja različne statistike. Ker pa se trg hitro širi in se dnevno razvijajo nove aplikacije, je verjetno le vprašanje časa, kdaj bomo dobili podobno oziroma boljšo rešitev.

Prednost konkurenčnih aplikacij je možnost nastavljanja omejitev porabe zakupljenih enot in posledično opozarjanje na prekoračitev. Konkurenčne rešitve prav tako ne potrebujejo stalnega dostopa do spleta za prikaz porabe, zato lahko porabo prikazujejo na začetnem zaslonu ne le v aplikaciji. Če bi želeli imeti takšne funkcionalnosti v naši mobilni aplikaciji, bi morali korenito

spremeniti celoten sistem, predvsem pa mobilno aplikacijo.

Ker na tržišču ni podobne aplikacije, imamo prednost s stališča nekonkurenčnosti. Se pa tukaj po drugi strani pojavi problem, saj je naša rešitev neprepoznavna. Uporabniki potrebujejo dalj časa, da se privadijo oziroma začnejo uporabljati našo rešitev. Lahko se pojavi tudi faktor nezaupanja. Veliko uporabnikov ne želi posredovati svojih osebnih informacij storitvi na spletu. Pri konkurenčnih rešitvah vse informacije ostanejo shranjene na mobilni napravi uporabnika. Ker naša mobilna aplikacija deluje le ob povezavi na splet je lahko to odločilnega pomena pri izbiri naše rešitve.

Poglavje 9

Zaključek

Uporaba mobilnih telefonov je v današnjem času nekaj povsem vsakdanjega. Ker lahko uporaba mobilnih storitev postane velik strošek, smo poskušali najti rešitev v obliki pametnega mobilnega pomočnika. Mobilni pomočnik kateri bi s pomočjo odkrivanja znanja iz podatkov, izluščil informacije o boljši in cenejši izrabi mobilnih storitev.

V začetnem delu naloge smo si pogledali teoretično ozadje odkrivanja znanja iz podatkov. Seznanili smo se s koraki procesa odkrivanja znanja. Bolj podrobno smo se osredotočili na metode podatkovnega rudarjenja. Ogledali smo si pristope podatkovnega rudarjenja, katere bi lahko uporabili za realizacijo pametnega mobilnega pomočnika. Na koncu pa smo se posvetili tudi etiki in varstvu osebnih podatkov ter predstavili pristope, kako zavarovati uporabnika pred zlorabami.

V nadaljevanju smo pregledali sorodne rešitve na tržišču za spremljanje porabe enot mobilnih storitev. Osredotočili smo se na funkcionalnosti, ki jih ponujajo in poskušali najti skupne pomanjkljivosti. Ugotovili smo, da sorodne rešitve ponujajo merjenje porabe, prikaz statistike, nastavljanje omejitev porabe enot in druge enostavnejše funkcionalnosti za spremljanje porabe. Skupna pomanjkljivost sorodnih rešitev je bila pomanjkanje naprednega svetovanja, v smislu, kako zmanjšati mesečne stroške in bolje izrabiti ponujene mobilne storitve.

V okviru magistrske naloge smo izdelali prototip pametnega mobilnega pomočnika. Pametni mobilni pomočnik nadgradi obstoječe rešitve, ki so trenutno dostopne na tržišču. Ponuja napreden prikaz porabljenih enot mobilne storitve, prikaz prilagodljive statistike z možnostjo primerjave porabe s povprečno porabo ostalih uporabnikov pomočnika, vizualne predstavitve porabe enot po omrežjih in pametno svetovanje. Pametni mobilni pomočnik uporabniku svetuje, katera mobilna storitev je najbolj primerna glede na njegovo porabo mobilnih enot. Pomočnik odkriva spremembe v porabi enot in v primeru večjega odstopanja, ponovno razvrsti mobilne storitve po primernosti. Uporabnik lahko s pomočjo pametnega mobilnega pomočnika spremlja porabo enot v tujini in s pomočjo nasvetov zmanjša stroške obiskov tujine. S pomočjo informacij o spremembah porabe enot v posameznem mesecu in podatkih o najprimernejših intervalih lahko uporabnik prilagodi uporabo oziroma izbere novo mobilno storitev.

Delovanje in smiselnost pametnega mobilnega pomočnika smo preverili v treh fazah. V prvi fazi smo odkrivali in odstranjevali napake, ter preverili hitrost delovanja. V drugi fazi smo preverili ali sistem vrača smiselne rezultate in kakšno korist imajo posamezni profili uporabnikov pri uporabi pametnega mobilnega pomočnika. V tretji fazi smo s pomočjo SWOT analize preverili smiselnost in konkurenčnost pametnega mobilnega pomočnika in kakšne prednosti ponuja v primerjavi z ostalimi sorodnimi aplikacijami. Ugotovili smo, da ponuja naprednejše funkcionalnosti in da omogoča uporabniku zmanjšanje mesečnih stroškov z direktnimi informacijami in nasveti. Glavne prednosti so napreden in prilagodljiv prikaz statistike, pametno svetovanje in delovanje na različnih platformah. Odkrili pa smo tudi nekatere pomanjkljivosti in slabosti pametnega mobilnega pomočnika.

Pametnega mobilnega pomočnika bi lahko izboljšali tako, da bi ponujal enostavnejše in bolj direktno svetovanje za zmanjšanje stroškov uporabe mobilnih storitev. Razvrščanje obračunskih intervalov in odkrivanje odstopanja pri mesečni statistiki bi lahko uporabili pri razvrščanju mobilnih storitev. Prav tako se pojavi potreba po nadgradnji zbiranja podatkov in izdelave

uporabniškega profila. Zbiranje podatkov bi lahko omogočili za več različnih platform. Za izdelavo profila bi lahko uporabili boljši pristop, kateri bi se bolje obnesel pri majhni količini podatkov. Mobilno aplikacijo bi lahko nadgradili tako, da bi delovala tudi brez povezave na svetovni splet in imela implementirane dodatne funkcionalnosti, kot je na primer opozarjanje na preseženo porabo enot.

Slike

2.1	Proces podatkovnega rudarjenja. Posamezni koraki so smiselno združeni.	8
4.1	Aplikacije za merjenje porabe mobilnih enot. Zgoraj levo Call Meter NG, sredina Call Stats, desno DroidStats, spodaj levo Dodal Phone, sredina Stats Free in desno My Data Manager .	22
5.1	Shema sistema z izbranimi tehnologijami.	27
6.1	Shema baze s pripadajočimi zbirkami.	31
6.2	Proces rangiranja storitev s podrobnostmi izdelave profila, seštevanja enot in primerjavo profilov.	34
6.3	Diagram procesa obdelave porabljenih enot v tujini.	37
6.4	Proces iskanja odstopanj.	38
6.5	Postopek klasifikacije podatkov za potrebe odkrivanja sprememb pri porabi enot.	39
6.6	Klasifikator mesečnih podatkov.	40
6.7	Proces za izbiro najustreznejših intervalov.	41
6.8	Postopek optimalne porazdelitev enot.	43
6.9	Spletna stran inteligentnega sistema.	44
6.10	Prikaz porabe mobilnih enot za trenutni mesec.	45
6.11	Graf na spletni strani, ki prikazuje statistiko porabe.	46
6.12	Forma za generiranje podatkov.	47

-
- 7.1 Prikaz seznama rangiranih storitev in opcij filtriranja na spletni strani. 51
 - 7.2 Zgoraj graf z mehurčki za prikaz porazdelitve mobilnih storitev. Spodaj stolpičast graf za prikaz porabe enot po mobilnih omrežjih. 52
 - 8.1 Prikaz prihranjenega denarja glede na izbrano storitev omrežja in najboljšo storitev pri enaki porabi enot. 61

Literatura

- [1] F. Abel, E. Herder, G. Houben, N. Henze, D. Krause. “Cross-system user modeling and personalization on the Social Web”, *User Model User-Adap Inter*, št. 23, str. 169–209, 2012.
- [2] G. Adomavicius, A. Tuzhilin. “User Profiling in Personalization Applications through Rule Discovery and Validation”, *KDD-99*, str. 377–381, 1999.
- [3] G. Amato, U. Straccia. “User Profile Modeling and Applications to Digital Libraries”, *Lecture Notes in Computer Science*, zv. 1696, str. 184–197, 1999.
- [4] M. Chen, J. Han, P. S. Yu. “Data Mining: An Overview from a Database Perspective”, *IEEE Transactions on knowledge and data engineering*, št. 6, zv. 8, str. 866–883, 1996.
- [5] T. Fawcett, F. Provost. “Combining Data Mining and Machine Learning for Effective User Profiling”, *KDD-96*, str. 8–13, 1996.
- [6] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth. “From Data Mining to Knowledge Discovery in Databases”, *American Association for Artificial Intelligence*, str. 37–54, 1996.
- [7] J. Hey (2004) The Data, Information, Knowledge, Wisdom Chain: The Metaphorical link. Dostopno na: http://web.archive.org/web/20071202033948/http://ioc.unesco.org/Oceanteacher/OceanTeacher2/02_InfTchSciCmm/DIKWchain.pdf

-
- [8] B. Krulwich. “Intelligent User Profiling Using Large-Scale Demographic Data”, *AI Magazine*, št. 2, zv. 18, str. 37–46, 1997.
- [9] B. W. Perry. *Java Servlet & JSP Cookbook*. O’Reilly Media, 2004.
- [10] E. Plugge, P. Membrey, T. Hawkins. *The Definitive Guide to MongoDB: The NoSQL Database for Cloud and Desktop Computing*. Apress, 2010.
- [11] R. Rogers, J. Lombardo, Z. Mednieks, B. Meike. *Android Application Development*. O’Reilly Media, 2009.
- [12] K. Sugiyama, K. Hatano, M. Yoshikawa. “Adaptive Web Search Based on User Profile Constructed without Any Effort from Users”, *WWW2004*, str. 675–684, 2004.
- [13] I. H. Witten, E. Frank, M. A. Hall. *Data Mining : Practical Machine Learning Tools and Techniques, Third Edition*. Morgan Kaufmann, 2011.
- [14] (2014) Android Call Meter NG (Review). Dostopno na: <http://mobile-apps-games.knoji.com/android-call-meter-ng-review/>
- [15] (2014) Apache Tomcat 7. Dostopno na: <http://tomcat.apache.org/tomcat-7.0-doc/index.html>
- [16] (2014) Call Stats - Google play. Dostopno na: <https://play.google.com/store/apps/details?id=com.movinapp.calls>
- [17] (2014) Dodol Phone – Data Monitoring Android App Review. Dostopno na: <http://telecomtalk.info/dodol-phone-data-monitoring-android-app-review/98260/>
- [18] (2014) Highcharts and Highstock documentation. Dostopno na: <http://www.highcharts.com/docs>
- [19] (2014) Jersey - RESTful Web Services in Java. Dostopno na: <https://jersey.java.net/>

-
- [20] (2014) jQuery API. Dostopno na: <http://api.jquery.com/>
- [21] (2014) Kje je številka. Dostopno na: http://www.npch.si/index.php?option=com_wrapper&Itemid=27
- [22] (2014) Manage Your Android Smartphone Usage with DroidStats. Dostopno na: <http://www.groovypost.com/reviews/monitor-android-phone-usage-droidstats/>
- [23] (2014) RESTful Web services: The basics. Dostopno na: <http://www.ibm.com/developerworks/webservices/library/ws-restful/>
- [24] (2014) Stats Free - Google play. Dostopno na: <https://play.google.com/store/apps/details?id=com.sebastian.stats&hl=sl>
- [25] (2014) The JSON Data Interchange Format. Dostopno na: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>
- [26] (2014) Waste Not Want More. Dostopno na: <http://www.techsupportalert.com/content/waste-not-want-more.htm>