Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Atrej Gognjavec

Kontekstno odvisna mobilna aplikacija in uporaba kontekstnega strežnika

diplomsko delo

univerzitetni študijski program prve stopnje
računalništvo in informatika

Mentor: doc. dr. Rok Rupnik

Ljubljana 2013

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Univerza *v Ljubljani*
Fakulteta *za računalništvo in informatiko*

Tržaška 25
1001 Ljubljana, Slovenija
telefon: 01 476 84 11
01 476 83 87
faks: 01 426 46 47
01 476 87 11
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00117/2013
Datum: 15.04.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ATREJ GOGNJAVEC**

Naslov: **KONTEKSTNO ODVISNA MOBILNA APLIKACIJA IN UPORABA KONTEKSTNEGA STREŽNIKA**

CONTEXT-AWARE MOBILE APPLICATION AND THE USE OF CONTEXT SERVER

Vrsta naloge: Diplomsko delo univerzitetnega študija prve stopnje

Tematika naloge:

Kontekstno odvisne mobilne aplikacije se prilagajajo trenutnim potrebam uporabnika. Razvoj takšnih aplikacij je res učinkovit le v primeru, ko temelji na kontekstnem strežniku, ki zagotavlja vse potrebne podatke za opredelitev trenutnega konteksta uporabnika.
Raziščite področje kontekstno odvisnih mobilnih aplikacij. Na podlagi arhitekture in načina delovanja kontekstnega strežnika izdelajte načrt arhitekture in delovanja prototipa kontekstno odvisne mobilne aplikacije za vodstveni kader v podjetjih. Aplikacijo razvijte na Android platformi.

Mentor:

doc. dr. Rok Rupnik

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI •

Dekan:

prof. dr. Nikolaj Zimic

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Atrej Gognjavec, z vpisno številko **63070087**, sem avtor diplomskega dela z naslovom:

Kontekstno odvisna mobilna aplikacija in uporaba kontekstnega strežnika

S svojim podpisom zagotavljam, da:

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

V Ljubljani, dne 10. septembra 2013 Podpis avtorja:

Kazalo

Povzetek

Cilj diplomske naloge je zasnova in izdelava kontekstno odvisne mobilne aplikacije za namen razvoja in preizkusa delovanja kontekstnega strežnika. Za dosego cilja je bilo potrebno opredeliti kontekst in kontekstno odvisnost ter se seznaniti s konceptom kontekstno odvisnega sistema in delovanjem kontekstnega strežnika. Opredeljene so bile naloge kontekstno odvisnih aplikacij ter z njimi zastavljene zahteve kontekstno odvisne mobilne aplikacije za razvoj in preizkus kontekstnega strežnika. Spoznati je bilo potrebno operacijski sistem Android, programska jezika Java in JavaScript, knjižnico za grafikone Highcharts, ter razvojno okolje Eclipse z vtičnikom ADT. Z uporabo naštetih tehnologij je bila kontekstno odvisna aplikacija razvita. Izmišljen je bil preprost primer uporabe in z njim predstavljeno delovanje aplikacije. Za konec so bile podane nekatere možne izboljšave in možnosti nadaljnjega razvoja.

Ključne besede: kontekst uporabnika, kontekstna odvisnost, relevantni podatki, obveščanje, mobilna aplikacija, Android

Abstract

The aim of the thesis was the design and creation of a context-aware mobile application for the purpose of developing and testing a context server. To achieve this goal, it was necessary to define context and context-dependency and become familiar with the concept of context-aware systems and the operations of a context server. A set of context-aware application tasks was defined, and on that basis, the requirements for the context-aware mobile application were established. It was necessary to get to know the Android operating system, Java and JavaScript programming languages, Highcharts charts library, and Eclipse development environment (including the ADT plugin). Using the aforementioned technologies, the context-dependent application was developed. A simple use case was devised to exemplify application usage. Finally, some possible improvements and opportunities for further development were given.

Key words: user context, contextual dependence, relevant data, notification, mobile application, Android

Poglavje 1

Uvod

Število aplikacij in mobilnih aplikacij hitro narašča. Veliko uporabnikov ima več kot dve napravi. Poleg računalnikov imamo še pametne telefone in tablične računalnike. Vse te naprave poganjajo številne aplikacije, ki uporabnikom omogočajo dostop do velike količine informacij. To lahko privede do prenasičenosti z informacijami. Kot je rekel Benjamin Franklin: »Čas je denar«. Kar v našem primeru pomeni, da uporabniki nimajo časa pregledovati informacij v iskanju njim relevantnih. Uporabniki tako zgrešijo pomembne informacije. Želijo pa dobiti prave informacije ob pravem času.

Dostavo relevantnih podatkov in obveščanje uporabnikov lahko rešimo s kontekstno odvisnim sistemom. Kontekstni sistem je sestavljen iz kontekstnega strežnika in kontekstno odvisnih aplikacij. Kontekstni strežnik služi ugotavljanju konteksta in dostavi relevantnih podatkov aplikacijam. Pri razvoju kontekstnega strežnika je potrebno imeti kontekstno odvisno aplikacijo, s pomočjo katere strežnik razvijamo.

V okviru diplomske naloge je bila za operacijski sistem Android razvita taka kontekstno odvisna mobilna aplikacija. Na začetku dela so predstavljene tehnologije in orodja uporabljena pri razvoju aplikacije. Nato sta opredeljena kontekst in kontekstna odvisnost. Na kratko je predstavljena arhitektura

kontekstnega strežnika in njegovo delovanje. Določene so naloge kontekstno odvisnih aplikacij in z njimi definirane specifikacije aplikacije. Opisan je tehnološki del razvoja aplikacije. Z zaslonskimi izrisi in opisom je predstavljeno delovanje aplikacije.

Poglavje 2

Tehnologije in orodja

2.1 Android

Android je celovita odprtokodna platforma zasnovana za mobilne naprave [5]. Razvijalcem Android nudi vsa potrebna orodja za hiter in preprost razvoj mobilnih aplikacij. Vse kar je potrebno za začetek razvoja za Android je Android SDK, potrebna ni niti mobilna naprava. Za uporabnike je intuitivna rešitev, ki omogoča veliko prilagoditev in izboljša uporabniško izkušnjo z mobilnimi napravami. Za proizvajalce mobilnih naprav pa je celovita rešitev za poganjanje njihovih naprav.

Z idejo o pametnejših mobilnih napravah z zavedanjem o uporabnikovi lokaciji in uporabnikovih željah, so leta 2003 ustanovili podjetje Android. Sprva so nameravali razviti napredni operacijski sistem za digitalne fotoaparate, vendar so se po ugotovitvi, da je trg premajhen, odločili narediti operacijski sistem za pametne telefone, ki bi konkuriral Symbianu in Windows Mobile.

Leta 2005 je Google kupil podjetje Android in pojavijo se predvidevanja o Googlovem mobilnem telefonu. Po 2 letih zatišja, 5. novembra 2007, ustanovijo združenje tehnoloških podjetij Open Handset Alliance. Predstavijo

Verzija	Psevdonim	Datum izdaje	API	Razširjenost
1.5	Cupcake	30. 4. 2009	3	0%
1.6	Donut	15. 9. 2009	4	0.1%
2.0–2.1	Eclair	26. 10. 2009	7	1.5%
2.2	Froyo	20. 5. 2010	8	3.1%
2.3–2.3.2	Gingerbread	6. 12. 2010	9	0%
2.3.3–2.3.7	Gingerbread	9. 2. 2011	10	34.1%
3.1	Honeycomb	10. 5. 2011	12	0%
3.2	Honeycomb	15. 7. 2011	13	0.1%
4.0.3–4.0.4	Ice Cream Sandwich	16. 12. 2011	15	23.3%
4.1.x	Jelly Bean	9. 7. 2012	16	32.5%
4.2.x	Jelly Bean	13. 11. 2012	17	5.6%
4.3	Jelly Bean	24. 7. 2013	18	0.0%

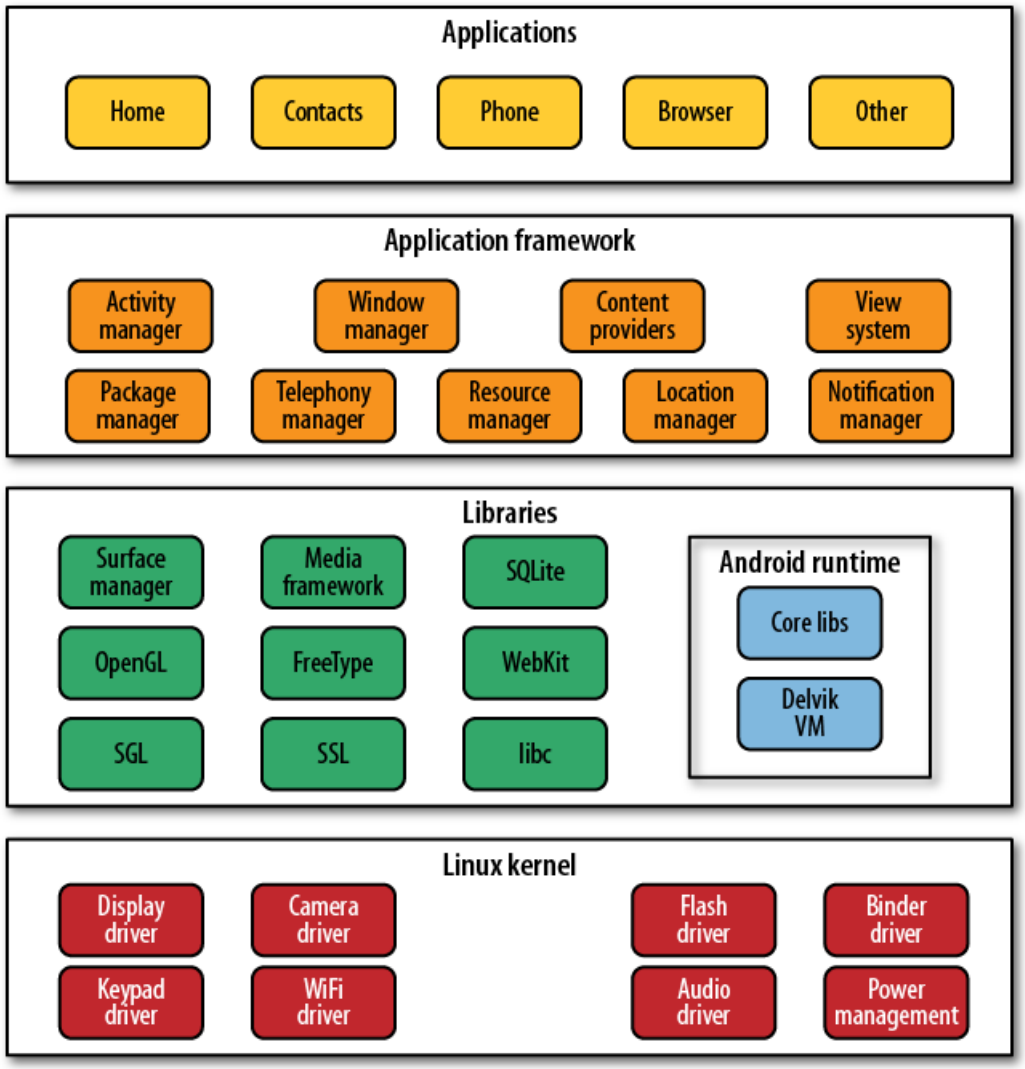
Tabela 2.1: Različie operacijskega sistema Android [13]

Android, platformo za mobilne naprave zasnovano na jedru Linux. 22. oktobra 2008 izdajo telefon HTC Dream, prvi telefon z Androidom. Istega leta izdajo tudi Android SDK. V letu 2009 pride do povečanja Androidnih naprav in izdaje novih verzij operacijskega sistema.

Android se sčasoma izboljšuje, kar se odraža v novih verzijah. V tabeli 2.1 so prikazane verzije od leta 2009 do danes in njihova razširjenost v odstotkih. Razvijalci morajo paziti predvsem na raven API, ki določa, na katerih napravah delujejo njihove aplikacije. Zaželena je čim nižja raven API, da lahko aplikacijo poganja čim več naprav, vendar ima nižja raven API manj funkcij, zato je potrebno izbrati najnižjo raven, ki ima vse potrebne funkcije.

Na sliki 2.1 je prikazanih pet elementov, iz katerih je sestavljen Android. Ti elementi so:

- Jedro Linux: Linux je odprtokodni operacijski sistem in predstavlja najnižjo plast Androida. Za izbiro Linuxa je veliko dobrih razlogov, glavni so prenosljivost, varnost, veliko funkcij [5].



Slika 2.1: Zgradba Androida [5]

- Knjižnice: Napisane so v programskih jezikih C in C++ ter aplikacijskemu ogrodju zagotavljajo dostop do storitev strojnih komponent.

- Izvajalno okolje: Sestavljajo ga javanske jedrne knjižnice in navidezni stroj Dalvik. Dalvik je mobilnim napravam prilagojena vrsta navideznega stroja Java.

- Aplikacijsko ogrodje: Je bogato okolje, ki nudi številne storitve raz-

vijalcem mobilnih aplikacij in je tudi najbolje dokumentiran element platforme. Vsebuje številne javanske knjižnice narejene posebej za Androida.

- Aplikacije: Aplikacije predstavljajo zgornjo plast zgradbe Androida. So to, kar uporabniki vidijo in uporabljajo. Nekatere aplikacije so na napravah že naložene in so potrebne za delovanje nekaterih pomembnih funkcij, druge pa si lahko uporabniki naložijo sami s spleta. Aplikacije so razvite v programskem jeziku Java z uporabo Android SDK. Prevedene so v izvršljivo datoteko s končnico .apk, ki je namenjena namestitvi na napravo. Vsaka aplikacija se zažene kot svoj Linux proces, kar omogoča medsebojno neodvisno delovanje in izoliranost.

2.2 Java

Java je inovativen programski jezik namenjen programom, ki morajo delovati na različnih računalniških sistemih [6]. Leta 1995 ga je za Sun Microsystems razvil James Gosling. Logotip Jave je prikazan na sliki 2.2.

Slika 2.2: Logotip Jave

Pri razvoju programskega jezika Java so upoštevali naslednje lastnosti programskega jezika [12]:

- Preprost, objektno orientiran, domač
- Robusten in varen
- Arhitekturno nevtralen in prenosljiv
- Visoko zmogljiv
- Tolmačen, niten, dinamičen

Neodvisnost od računalniške arhitekture omogoča navidezni stroj Java (JVM - Java Virtual Machine). Java aplikacije se prevedejo v binarno obliko in poganjajo na navideznem stroju.

2.3 JavaScript

JavaScript je programski jezik spleta. Velika večina današnjih spletnih strani uporablja JavaScript in vsi današnji spletni brskalniki na računalnikih, igralnih konzolah, tabličnih računalnikih, pametnih telefonih, ... vključujejo tolmač za JavaScript, kar naredi JavaScript najbolj razširjen programski jezik v zgodovini [4]. Uporablja se pri izdelavi interaktivnih spletnih strani v kombinaciji s HTML-jem, ki določa vsebino strani, in CSS-om, ki določa predstavitev strani.

Sintaksa JavaScripta ohlapno sledi programskemu jeziku C. JavaScript od Jave kopira veliko imen in poimenovanj, vendar sta jezika drugače nepovezana in imata drugačno semantiko. Koda JavaScript funkcij se lahko nahaja v samem HTML dokumentu ali pa v ločeni .js datoteki. Funkcije se prožijo na strani odjemalca ob dogodkih kot so: naložitev strani, klik na gumb, premik miške ... in lahko dinamično spreminjajo samo vsebino spletne strani.

2.4 JSON

JSON ali JavaScript Object Notation je standard za izmenjavo podatkov. Izvira iz programskega jezika JavaScript in se uporablja za predstavitev preprostih podatkovnih struktur in asociativnih vrst [14]. JSON je definiral Douglas Crockford.

```
{
    "firstName": "John",
    "lastName": "Smith",
    "age": 25,
    "address": {
        "streetAddress": "21 2nd Street",
        "city": "New York",
        "state": "NY",
        "postalCode": 10021
    },
    "phoneNumbers": [
        {
            "type": "home",
            "number": "212 555-1234"
        },
        {
            "type": "fax",
            "number": "646 555-4567"
        }
    ]
}
```

Slika 2.3: Primer JSON-a [14]

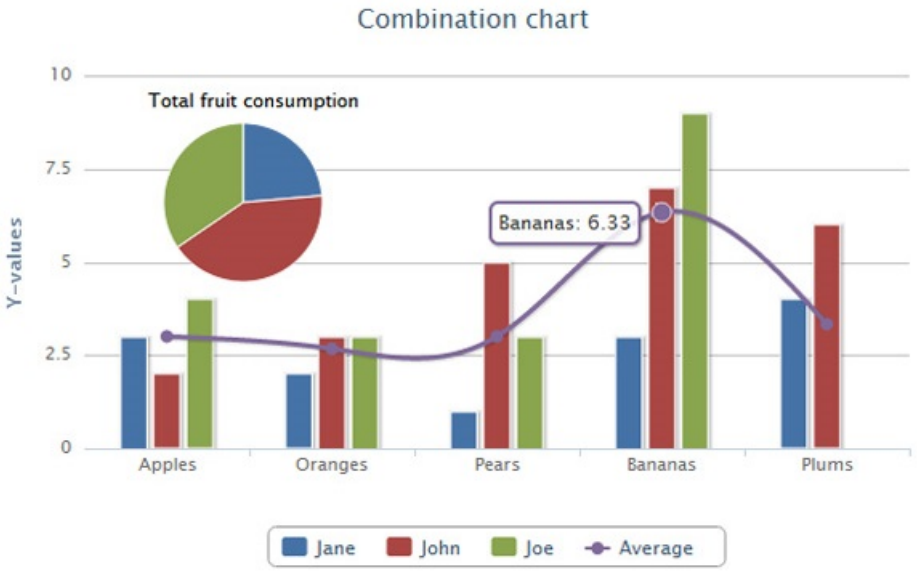
Osnovni tipi JSONA so število (*Number*), niz (*String*), Boolean, vrsta (*Array*), objekt (*Object*) in prazen (*null*). Na sliki 2.3 je prikazan JSON objekt, ki opisuje osebo. Ime in priimek sta tipa niz, starost je število, naslov objekt, telefonske številke pa so definirane kot vrsta objektov.

Strukturirani podatki v formatu JSON se najpogosteje uporabljajo za prenos med spletnimi aplikacijami in strežnikom, tako je uporaben kot preprostejša

alternativa XML-ju. Navkljub povezavi z JavaScriptom je JSON neodvisen od programskih jezikov, zato obstajajo razčlenjevalniki za JSON v številnih programskih jezikih.

2.5 Highcharts

Highcharts je v JavaScript-u napisana knjižnica za risanje grafikonov. Nudi preprost način za dodajanje interaktivnih grafikonov na spletno stran ali v spletno aplikacijo [11]. Highcharts podpira veliko različnih grafikonov: stolpčni, palični, črtni, tortni, raztreseni, ploščinski, kolobarni, polarni, ... Na sliki 2.4 je prikazan primer, ki vsebuje stolpčni, črtni in tortni grafikon.



Slika 2.4: Primer grafikona narejenega s Highcharts

Knjižnica Highcharts je na voljo v brezplačno uporabo za nekomercialne namene in deluje na vseh novejših brskalnikih. Ker je napisana v JavaScriptu, odjemalci ne potrebujejo nobenih vtičnikov, kot so potrebni pri uporabi Flasha ali Jave. Tudi na strežniku ni potrebna nobena dodatna namestitev. Za

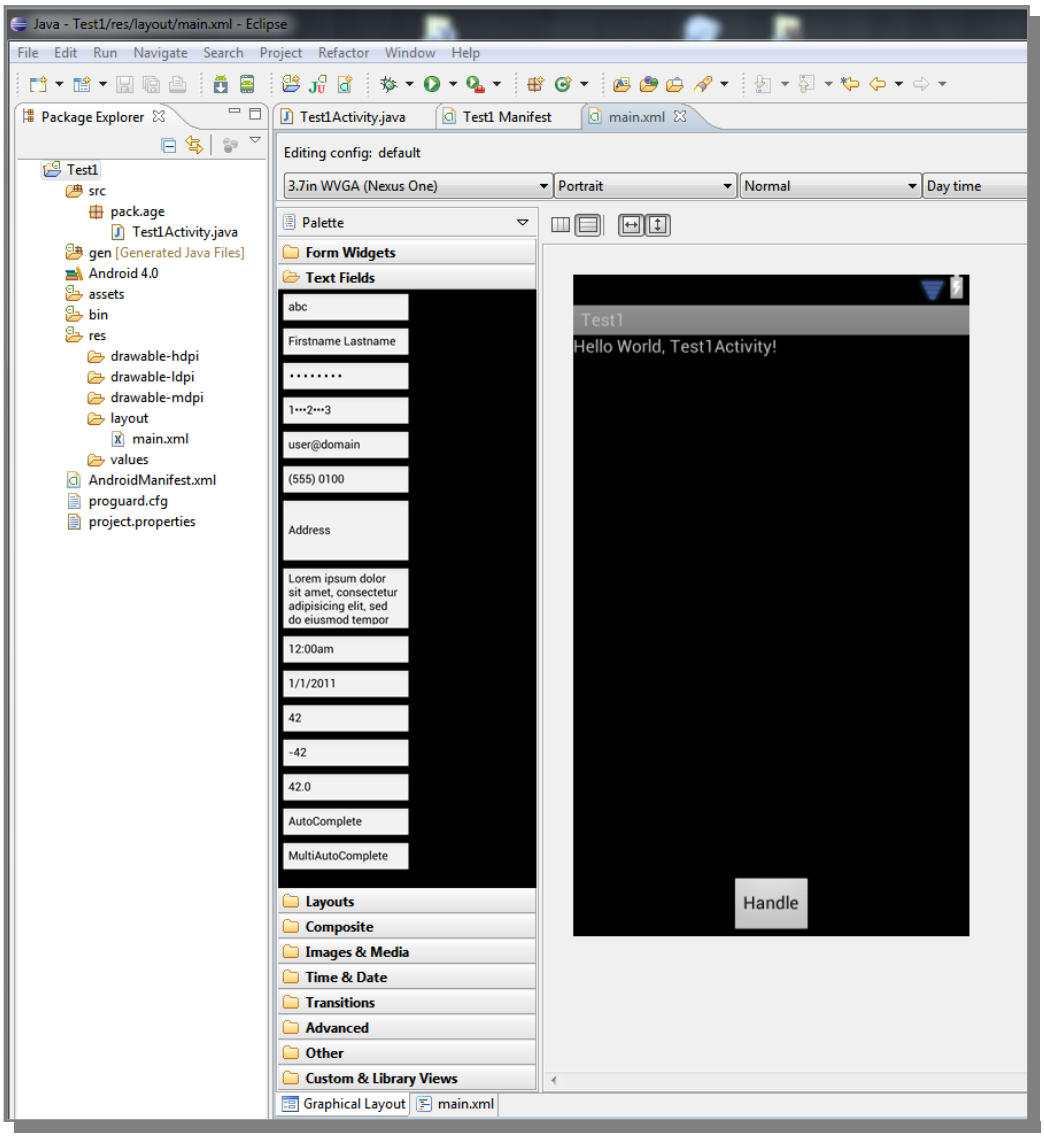
delovanje sta potrebna samo datoteka highcharts.js in jQuery. Nastavitve grafikona so podane kot JSON in ne potrebujejo nobenih posebnih programerskih znanj. Po kreiranju grafikona se podatke lahko z uporabo API-ja dinamično dodaja, spreminja in odstranjuje. Highcharts podpira tudi prikaz več osi na istem grafikonu, kar omogoča sočasen prikaz podatkov v različnih enotah. Uporabnikom omogoča tudi izvoz grafikona kot datoteko v formatih PNG, PDF, JPG in SVG ter tiskanje.

2.6 ADT in Eclipse

Android SDK je komplet za razvoj programske opreme operacijskega sistema Android. Android SDK nudi knjižnice API in razvijalna orodja potrebna za gradnjo, testiranje in razhroščevanje aplikacij za Android [10]. Najhitrejši začetek razvoja omogoča namestitev snopa orodij za razvoj Android (ADT Bundle). Z namestitvijo dobimo:

- Razvojno okolje Eclipse z vtičnikom ADT
- Orodja Android SDK
- Orodja za operacijski sistem Android
- Najnovejši operacijski sistem Android
- Najnovejšo sliko sistema Android za posnemovalnik

Eclipse je programsko razvojno okolje, ki omogoča razvijanje v veliko programskih jezikih. Sestavljen je iz osnovnega delovnega okolja in vtičnikov za prilagoditev okolja. Na sliki 2.5 je prikazan grafični urejevalnik postavitve aplikacije za Android v razvojnem okolju Eclipse z vtičnikom ADT.



Slika 2.5: Razvojno okolje Eclipse z vtičnikom ADT

Poglavje 3

Kontekst in kontekstna odvisnost

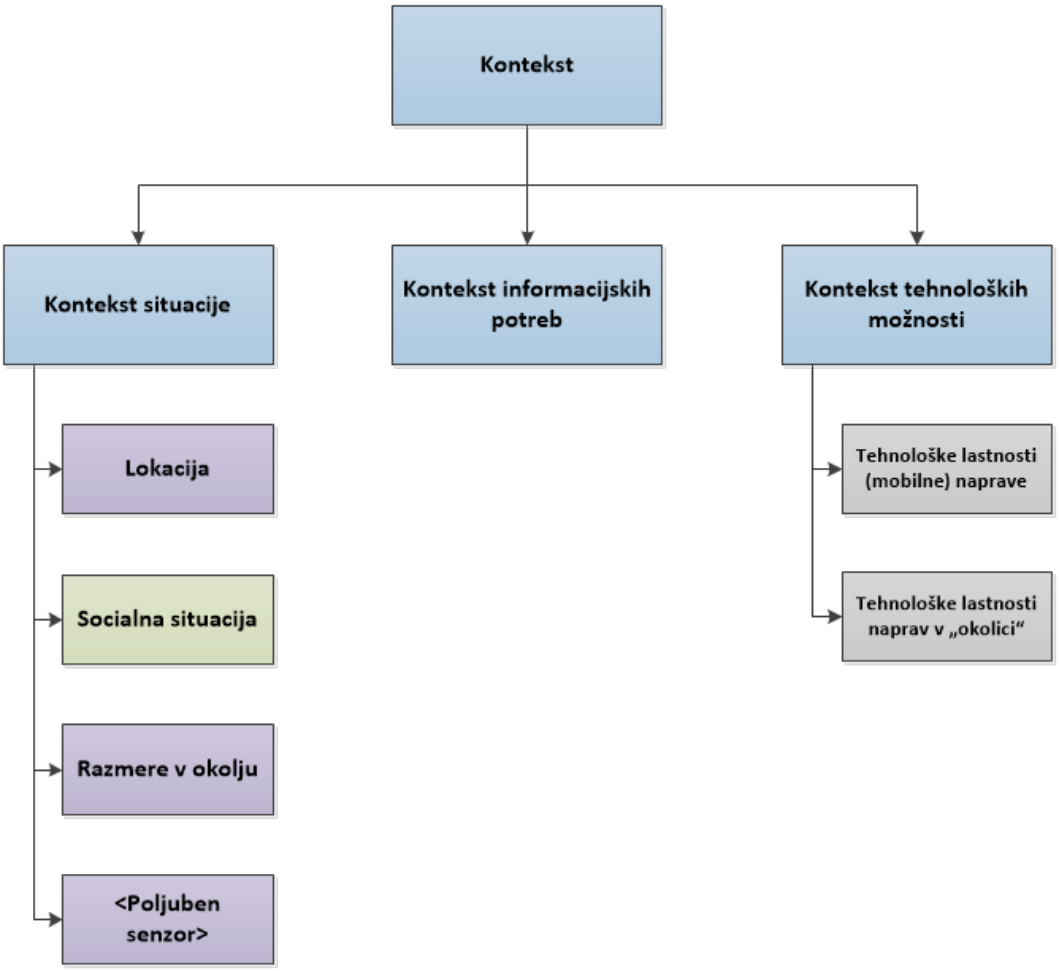
Za uporabnike aplikacij je pomembno, da dobijo prave informacije ob pravem času. Prave informacije in pravi čas sta določena s kontekstom uporabnika.

Kontekst je celovita informacija o situaciji, v kateri se nahaja mobilni uporabnik, njegovih potrebah oz. potrebah informacijskega sistema ter objektih, lokacijah in napravah, ki so del logičnega virtualnega prostora uporabnika [8].

Kontekst je sestavljen iz treh delov:

- Kontekst situacije – celovita informacija o situaciji uporabnika
- Kontekst informacijskih potreb – celovita informacija o informacijskih potrebah uporabnika
- Kontekst tehnoloških možnosti – celovita informacija o tehnoloških lastnosti in zmožnosti uporabnikovih naprav

Podrobnejši model strukture konteksta uporabnika je prikazan nasliki 3.1. Kontekst uporabnika je sestavljen iz treh delov. Vsakega od teh delov pa



Slika 3.1: Model strukture konteksta uporabnika

določajo različne informacije o uporabniku.

Uporabnikova lokacija, socialna situacija in razmere v okolju določajo kontekst uporabnikove situacije. Lokacija poleg koordinat, na katerih se nahaja uporabnik, pove tudi nekaj o tem, kaj ta lokacija predstavlja. Uporabnik je lahko v pisarni, doma, na poti ... Uporabnikova socialna situacija je odvisna od tega, s kom je uporabnik trenutno, koga pozna in s kom dela. Razmere v okolju opisujejo dogajanje v okolju, v katerem se uporabnik nahaja. Poleg razmer v okolju je kontekst situacije uporabnika določen še s podatki senzorjev uporabnikovih naprav in naprav v uporabnikovi okolici.

Tehnološke lastnosti uporabnikovih naprav in naprav v okolici določijo kontekst tehnoloških možnosti uporabnika. Ta kontekst določa, katere aplikacije in storitve lahko uporabnik uporablja ter v kakšnem obsegu jih lahko uporablja. Tehnološke lastnosti uporabnikovih naprav povejo, katere aplikacije in storitve delujejo na teh napravah. Tehnološke lastnosti v okolici uporabnika pa povejo, katere zunanje storitve lahko aplikacije na uporabnikovih napravah uporabljajo za obogatitev svojega delovanja.

Informacijske potrebe uporabnika so odražene s kontekstom informacijskih potreb. Informacijske potrebe uporabnika so odvisne od uporabnikove vloge, podatkov o pretekli uporabi aplikacije in preteklih informacijskih potrebah ter tudi od aplikacijskih podatkov samih.

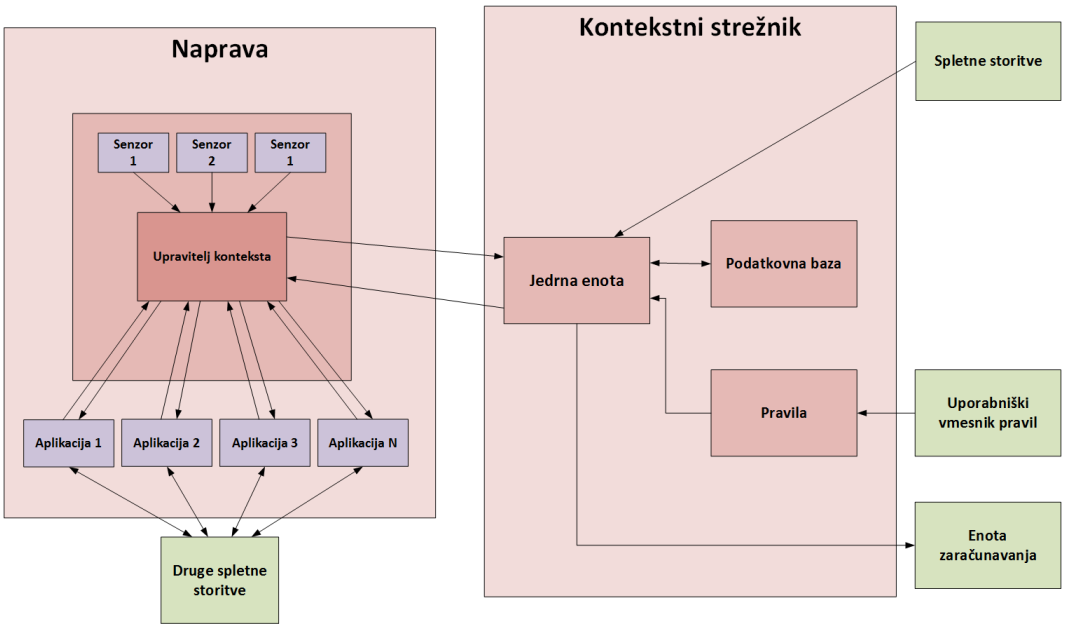
Kontekst uporabnikom prinaša dodano vrednost pri uporabi storitev in aplikacij. Pri kontekstno odvisnih aplikacijah in storitvah pa poleg naprav potrebujemo še kontekstni strežnik.

Poglavje 4

Kontekstni strežnik

Sprotno izračunavanje konteksta zahteva neprestano poganjanje procedur in upravljanje stalno naraščajoče baze kontekstnih podatkov. To zahteva neprestano delovanje naprave, ki kontekst izračunava. Kljub vse zmogljivejšim mobilnim napravam izračunavanje konteksta presega njihove zmožnosti, predvsem zmožnosti baterij. Zato in zaradi želje po centralnem sistemu, ki uporabniku nudi kontekstno relevantne podatke na več napravah, je kontekstno odvisni sistem ločen na kontekstni strežnik ter napravo. Naprave so zadolžene za pošiljanje podatkov o uporabi aplikacij ter podatkov s senzorjev naprav in prikaz podatkov, ki jih dobi v odgovor s kontekstnega strežnika.

Arhitektura kontekstno odvisnega sistema je prikazana na sliki 4.1. V našem primeru aplikacija prevzema vodenje konteksta na napravi in združi ter poenostavi delovanje naprave. Podrobnejša razlaga kontekstno odvisne aplikacije sledi v poglavju 5. Arhitektura kontekstnega strežnika pa je enaka sliki 4.1.



Slika 4.1: Arhitektura kontekstno odvisnega sistema

4.1 Arhitektura kontekstnega strežnika

V okviru kontekstnega sistema je kontekstni strežnik razdeljen na več komponent, ki opravljajo naslednje naloge [7]:

- Izmenjava podatkov z napravami
- Obogatitev konteksta
- Shranjevanje in vzdrževanje kontekstnih podatkov
- Vzdrževanje in izvajanje poslovnih pravil
- Zaračunavanje

Kontekstni podatki uporabnikov, podatki uporabnikov o uporabi aplikacije, podatki za aplikacije in drugi pomembni podatki, ki se tičejo uporabnikov ter njihovih naprav, so shranjeni v podatkovni bazi. Kontekstni strežnik prejme podatke za aplikacije in podatke za uporabnike od zunanjih virov preko spletnih storitev. Uporabniške naprave pa neprestano pošiljajo podatke o upo-

rabi aplikacije in vrednosti s senzorjev potrebne za izračun konteksta.

Skupek poslovnih pravil, glede na uporabnikov kontekst in uporabo aplikacije, določa, kateri podatki se pošljejo s kontekstnega strežnika na uporabnikovo napravo ter na katero napravo se pošljejo. Pravila se določa preko zunanjega uporabniškega vmesnika, kar omogoča hitro prilagajanje ter spreminjanje pravil brez posegov v programsko kodo. Nekatera pravila si lahko prilagodijo tudi uporabniki sami.

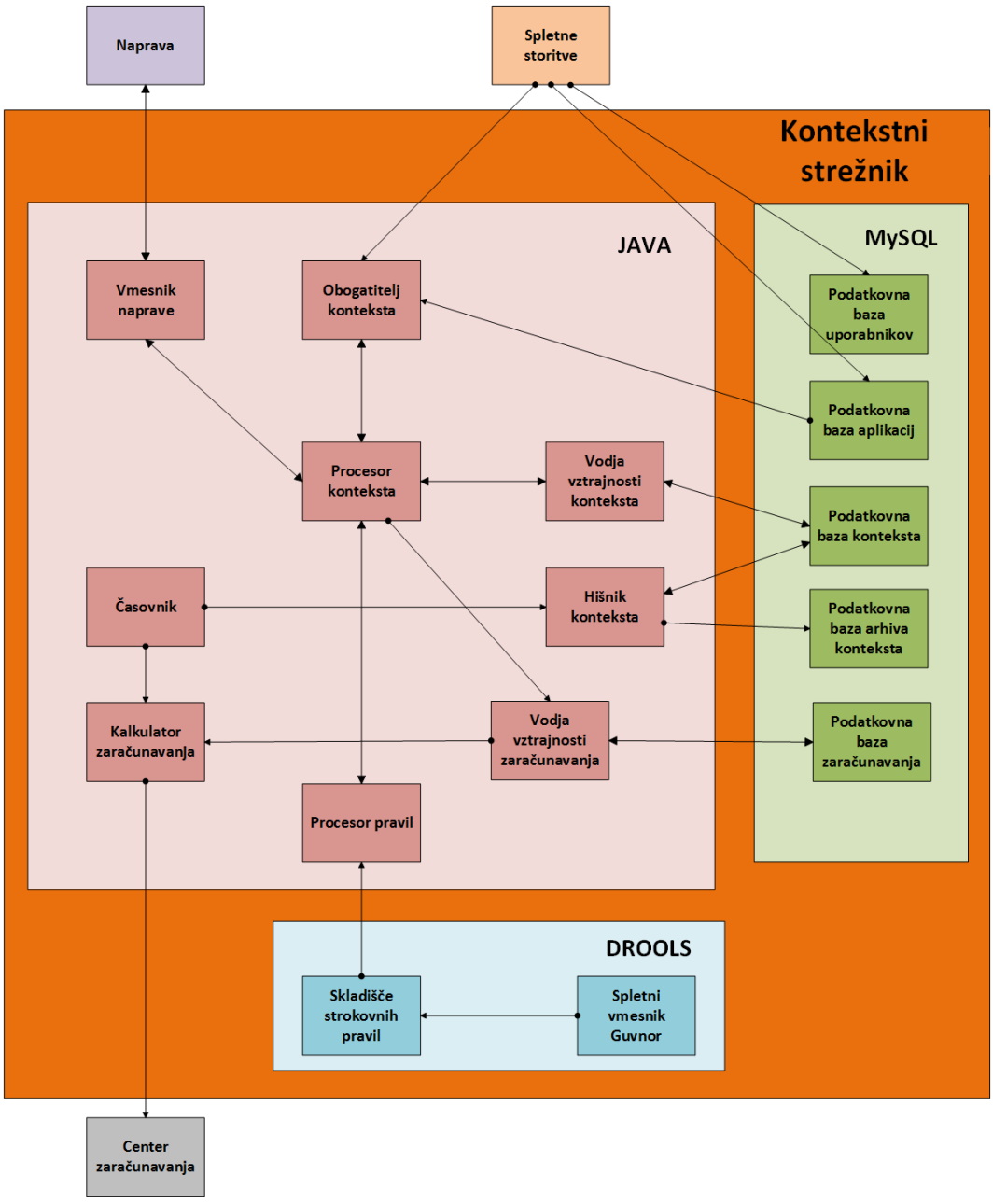
V enoti za zaračunavanje se izračunajo in tabelirajo računi uporabnikov. Zaračunavanje temelji na številu uporabnikovih aplikacij, pogostosti uporabe, številu pravil, intervalu izračunavanja konteksta, ... in se izvaja periodično vsak teden ali mesec.

Glavna enota kontekstnega strežnika je jedrna enota (Core Unit). Jedrna enota deluje kot posrednik med napravo in ostalimi enotami strežnika. Ko jedrna enota prejme podatke od naprave, izvede naslednje zaporedje operacij [7]:

1. Obogati kontekst in pri tem lahko uporabi zunanje spletne storitve.
2. Shrani obogateni kontekst v podatkovno bazo.
3. Iz baze pridobi nedavno zgodovino konteksta.
4. S poslovnimi pravili preveri pridobljeno zgodovino konteksta.
5. Napravi pošlje podatke pridobljene v točki 4.

4.2 Vmesnik naprave

Vmesnik naprave je tisti del jedrne enote na kontekstnem strežniku, ki skrbi za vso komunikacijo z napravami. Na sliki 4.2 je podrobnejši načrt kontekstnega strežnika, ki razčleni enote prikazane na sliki 4.1 na manjše elemente. Zgoraj levo se nahaja naprava, pod njo pa je narisan vmesnik naprave.



Slika 4.2: Podrobnejši načrt kontekstnega strežnika

Pri razvoju naše kontekstno odvisne mobilne aplikacije vmesnik naprave ne bo komuniciral z napravo, ampak z aplikacijo. Vmesnik naprave bo aplikaciji dostavljal vse podatke potrebne za delovanje, aplikacija pa bo vmesniku

naprave posredovala vse podatke o uporabi aplikacije in podatke senzorjev naprave. To bo tudi edini del kontekstnega strežnika, s katerim bo aplikacija komunicirala.

Poglavje 5

Zasnova kontekstno odvisne mobilne aplikacije

Obstajajo številne definicije kontekstno odvisnih aplikacij [9]:

- Kontekstno odvisne aplikacije so aplikacije, ki dinamično spreminjajo in prilagajajo svoje delovanje glede na kontekst aplikacije in uporabnika [2].

- Kontekstno odvisne aplikacije so aplikacije, ki avtomatično nudijo informacije in/ali ukrepajo v skladu s trenutnim kontekstom uporabnika, ki ga zaznavajo senzorji [1].

- Kontekstno odvisne aplikacije so aplikacije, ki uporabljajo kontekst za podajanje relevantnih informacij in/ali storitev uporabniku, kjer je relevantnost odvisna od uporabnikove naloge [3].

Pri zasnovi naše aplikacije nam je najbližja tretja definicija, ker poudarja uporabnikove informacijske potrebe. Da je aplikacija kontekstno odvisna, mora tako poleg običajnih aplikacijskih nalog opravljati še dodatne naloge.

5.1 Naloge kontekstno odvisne mobilne aplikacije

Kontekstno odvisna aplikacija je poleg splošnega prikazovanja podatkov sposobna prikazovati tudi uporabniku relevantne podatke. Način prikaza podatkov zna aplikacija prilagajati uporabnikovim potrebam. Struktura aplikacije zato ne sme biti preveč toga. Z obveščanjem in samodejnim proženjem aplikacija uporabniku zagotavlja pravočasnost informacij. Aplikacija zajema kontekstne podatke preko senzorjev in preko sledenja uporabe aplikacije. Podatke posreduje kontekstnemu strežniku, ki jih shranjuje in aplikaciji posreduje nove kontekstno odvisne podatke. Za boljše delovanje je zaželena stalna povezljivost naprave s strežnikom.

5.2 Specifikacija aplikacije

Za namen preizkusa delovanja kontekstnega strežnika smo naredili kontekstno odvisno aplikacijo z naslednjimi lastnostmi:

- Komunikacija s kontekstnim strežnikom poteka preko vmesnika naprave
- Podatke prejme s kontekstnega strežnika
- Podatki so prikazani z grafikoni
- Število grafikonov ni omejeno
- Vrstni red grafikonov je spremenljiv
- Na grafikonih je prikazano različno število serij podatkov
- Vrstni red serij podatkov je spremenljiv
- Vidnost serij podatkov je spremenljiva
- Vsaka serija podatkov je lahko prikazana na različne načine

- Aplikacija lahko obvešča uporabnika tudi, ko ni aktivna
- Aplikacijo lahko uporabnik zažene preko obvestila
- Aplikacija zajema lokacijo uporabnika
- Beležijo se podatki o uporabi
- Podatki o lokaciji in uporabi se pošljejo na strežnik

Poglavje 6

Delovanje aplikacije

Aplikacija je razvita tako, da se lahko uporablja za prikaz podatkov iz zelo različnih zgodb. Za potrebe preizkusa kontekstnega strežnika smo si zamislili uporabo v nekem podjetju. Podatki so namenjeni različnim vodilnim osebam v podjetju. Tako kot v podjetju ima tudi vsak uporabnik aplikacije svoje vloge. V našem primeru obstajajo vloge: generalni direktor, finančni direktor, direktor prodaje in direktor nabave. Vsako vlogo zanima drugačen nabor podatkov in vsaka vloga ima svoj skupek pravil. Uporabniki in njihove vloge so določene na strežniku.

6.1 Primer uporabe

Vzemimo za primer uporabnika z vlogo direktorja nabave. Direktor nabave je zadolžen za stanje zaloge v skladiščih. Njegovo podjetje ima tri skladišča. Direktorja nabave zanima vrednost zaloge v skladiščih in stanje bančnega računa podjetja. Ima tudi svoja pravila, med njimi je pravilo, ki direktorja nabave obvesti, če vrednost zaloge v določenem skladišču pade pod dovoljeno vrednost.

Ob prvem zagonu aplikacije se odpre forma z nastavitvami s poljem za vnos

naslova kontekstnega strežnika, poljem za uporabniško ime, poljem za geslo, gumbom za prijavo/odjavo, polje za vpis intervala storitve prejemanja in pošiljanja podatkov ter gumb za vklop/izklop te storitve. Po uspešni prijavi kontekstni strežnik pošlje napravi relevantne podatke za prikaz in druge podatke potrebne za delovanje aplikacije.

6.2 Prejeti podatki

Primer podatkov, ki jih naprava prejme od kontekstnega strežnika:

```
{
  "deliveryID":314159,
  "sections":[
    {
      "name":"Warehouse Stock",
      "yTitle":"Stock Value (in 1000\u20AC)",
      "categories":["'Thu'","'Fri'","'Sat'","'Sun'","'
         Mon'","'Tue'","'Wed'"],
      "series":[
        {
          "name":"Warehouse A",
          "data":[28.6,24.8,21.5,21.5,36.5,34.5,30.2],
          "visible":false,
          "type":"column"
        },
        {
          "name":"Warehouse B",
          "data"":[33.6,28.8,22.5,20.0,16.0,13.5,8.6],
          "visible":true
        },
        . . .
      ]
    },{
```

```
      "name":"Account Balance",
      "yTitle":"Balance (in \u20AC)",
      "categories":[ . . . ],
      "series":[ . . . ]
    }
  ],
  "notification":{
    "title":"Warning in Warehouse B",
    "text":"Warning! The value of stock in Warehouse B
       has fallen below the legal limit of 10.000\u20AC
       ."
  }
}
```

Na začetku prejetih podatkov je identifikacijska številka dostavljenih podatkov. Pomembna je pri posodabljanju podatkov in pošiljanju podatkov o uporabi na strežnik. Strežnik iz identifikacijske številke dostavljenih podatkov ve na katerega uporabnika, napravo, aplikacijo in katere podatke se nanašajo podatki o uporabi aplikacije.

Identifikacijski številki dostavljenih podatkov sledi seznam sekcij. Seznam ima lahko eno ali več sekcij. Vsaka sekcija je objekt, ki vsebuje ime, naslov osi y grafikona, kategorije in seznam serij podatkov. Sekcija predstavlja stran z grafikonom, tako je ime oz. naslov sekcije hkrati tudi naslov grafikona. Naslov osi y je vertikalen napis ob osi y. Kategorije so seznam nizov, ki so napisani ob osi x. V tem primeru jih je sedem in so prve črke dni v tednu v angleščini. Podrobnejši prikaz sekcij in grafikonov z razlago elementov sledi v poglavju 6.4.

Seznam serij podatkov vsebuje eno ali več serij podatkov za prikaz na grafikonu. Serija podatkov je objekt z imenom, seznamom podatkov, tipom grafikona in vidnostjo. Vse serije v sekciji imajo navadno enako število podatkov v seznamu podatkov. Število podatkov je tudi enako številu nizov

v kategorijah, saj kategorije opisujejo, kdaj ima določena serija podatkov kakšno vrednost. Tip grafikona in vidnost sta opcijska parametra. Če nista podana, je izbran stolpični tip grafikona, podatki pa so vidni.

Poleg identifikacijske številke dostavljenih podatkov in seznama sekcij se lahko pošlje tudi obvestilo. Obvestilo sestavljata naslov obvestila ter tekst obvestila.

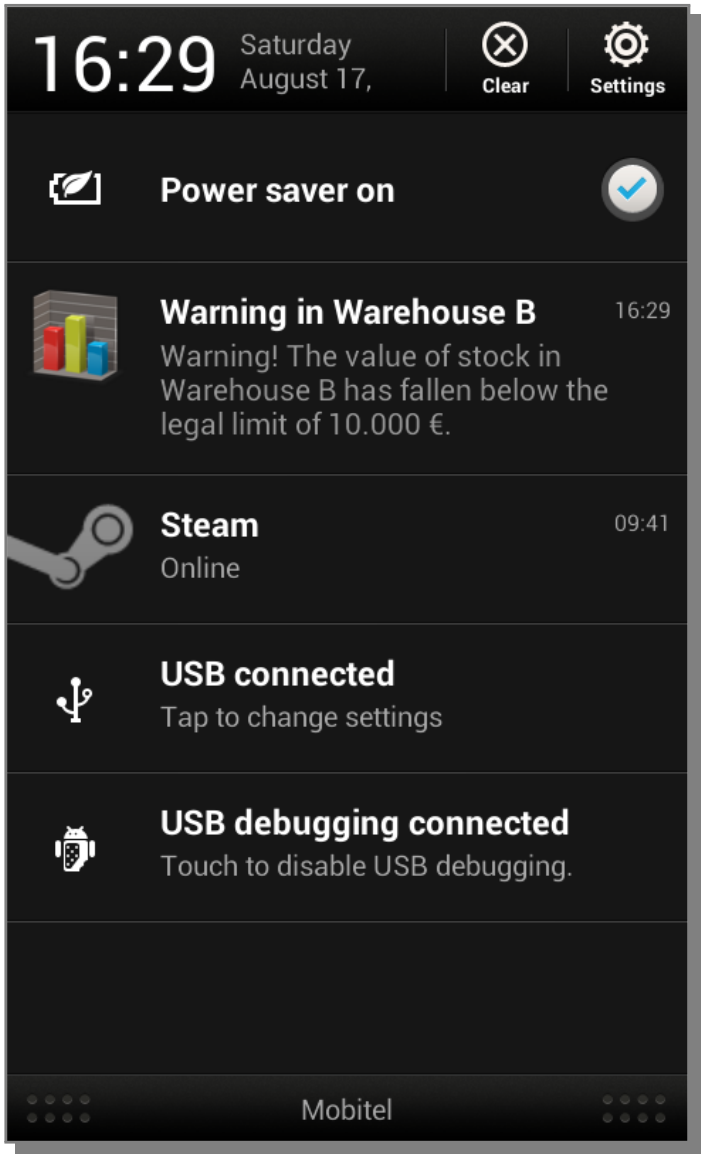
6.3 Obvestilo

V primeru, ko želi kontekstni strežnik obvestiti uporabnika o nekem dogodku, skupaj s poslanimi podatki pošlje tudi obvestilo. Obveščanje se sproži v primeru, da pride do sprožitve nekega pravila. Na primer padanja neke določene vrednosti. Če storitev prejemanja in pošiljanja podatkov prejme med podatki obvestilo, mobilna naprava zapiska, zavibrira in začne utripati z LED diodo. Potem, ko naprava uporabnika uspe opozoriti nase, lahko uporabnik med obvestili opazi novo obvestilo.

Na sliki 6.1 je zaslonski izris, ki prikazuje primer obvestila. Obvestilo je prikazano z ikono aplikacije, naslovom obvestila, vsebino obvestila ter časom obveščanja. Naslov obvestila v dveh ali treh besedah pove razlog obveščanja. V vsebini obvestila pa je razlog obveščanja še obrazložen v stavku ali dveh. V tem primeru direktorja nabave opozarja na prenizko vrednost zaloge v skladišču B.

Ob naslednjem zagonu storitve prejemanja in pošiljanja podatkov lahko ta ugotovi, da obvestilo ni več med prejetimi in ga preneha prikazovati tudi, če ga uporabnik ni opazil. Do tega pride, če kontekstni strežnik ugotovi, da obvestilo ni več aktualno, ker je uporabnik informacijo že opazil s kakšnim drugim programom ali na drugi napravi ali ker se je vrednost vrnila v meje normale. Kontekstni strežnik to ugotavlja s pomočjo pravil.

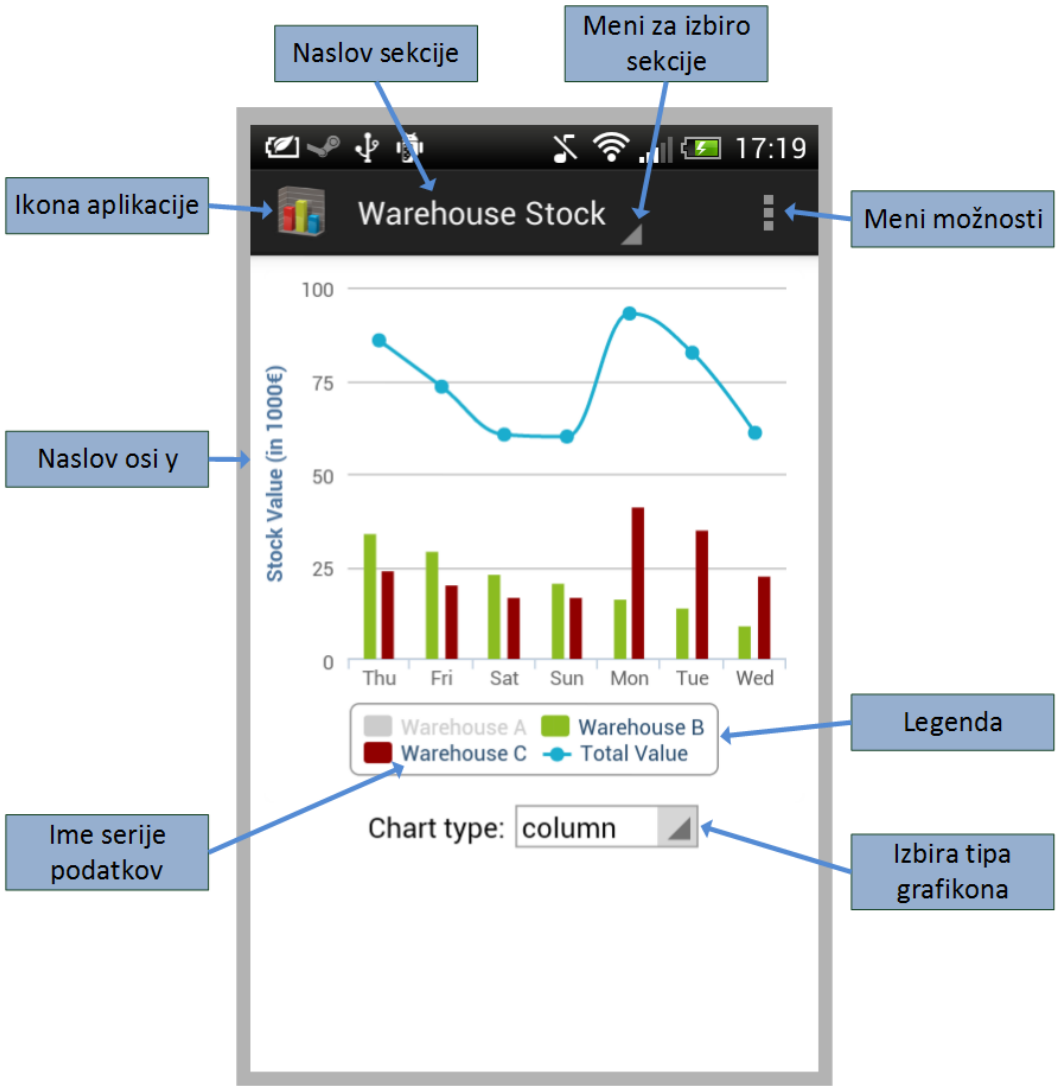
Na obvestilo lahko uporabnik klikne in s tem zažene aplikacijo. Aplikacija



Slika 6.1: Zaslonski izris obvestila

ob zagonu prikaže prvo sekcijo. Na njej je navadno grafikon, ki se nanaša na obvestilo. Uporabnik lahko seveda aplikacijo zažene tudi sam, v tem primeru je prva sekcija tista, za katero kontekstni strežnik meni, da je za uporabnika trenutno najbolj relevantna.

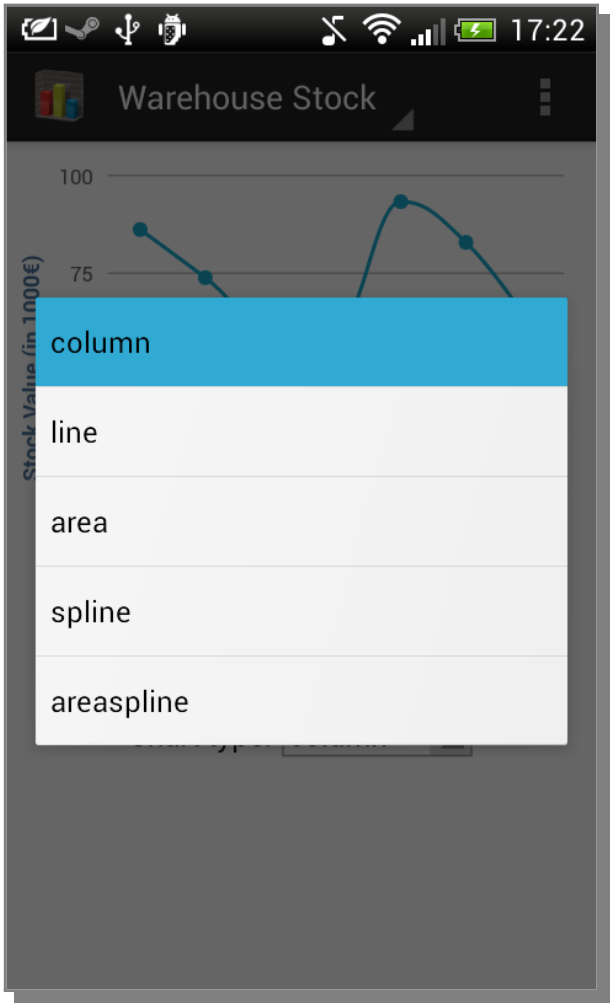
6.4 Sekcije in grafikoni



Slika 6.2: Zaslonski izris prve prikazane sekcije aplikacije z razlago

Na sliki 6.2 je prikazan zaslonski izris prve prikazane sekcije s podatki iz primera in dodano razlago posameznih elementov. Zgoraj z leve proti desni so ikona aplikacije, naslov sekcije in meni možnosti. Skrajno levo je naslov osi y, ki pojasnjuje, kaj in v kakšni enoti predstavlja grafikon. Na sredini se nahaja grafikon. Grafikon se samodejno prilagaja glede na lestvico na osi

y in na osi x. Oznake na osi y so tako določene glede na vrednosti serij podatkov, oznake osi x pa so določene glede na kategorije. Pod grafikonom se nahaja legenda z imeni serij podatkov, pod legendo pa še gumb za izbiro tipa grafikona.

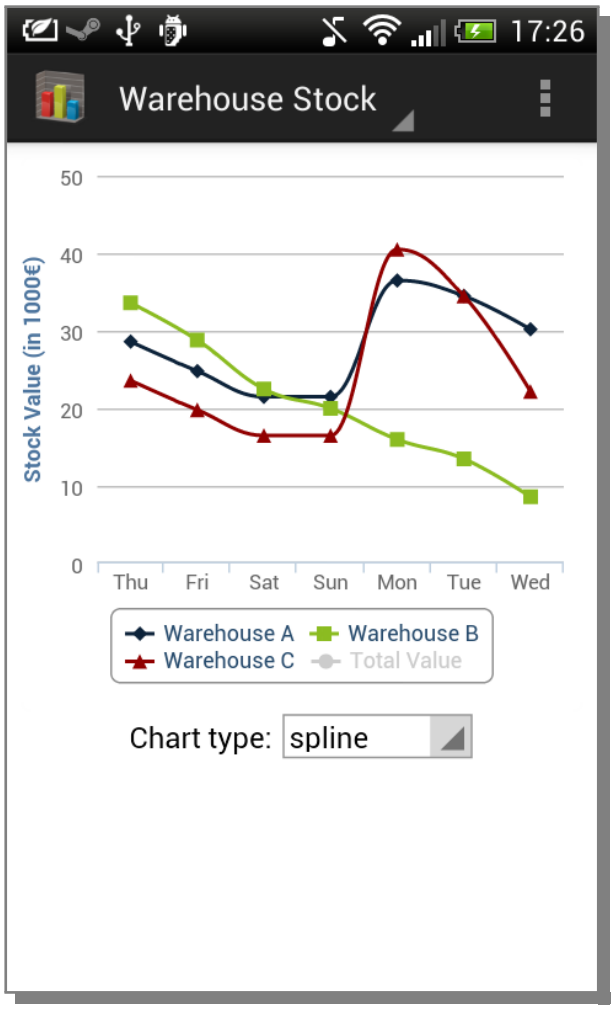


Slika 6.3: Zaslonski izris seznama za izbiro tipa grafikona

S pritiskom na gumb za izbiro tipa grafikona se odpre seznam vseh možnih tipov. Zaslonski izris seznama za izbiro tipa grafikona je prikazan na sliki 6.3. Aplikacija ponuja pet različnih tipov grafikonov:

- Stolpčni

- Črtni lomljen
- Ploščinski lomljen
- Črtni zaobljeni
- Ploščinski zaobljeni



Slika 6.4: Zaslonski izris z drugim tipom grafikona in skritimi podatki za skupno vrednost

Kontekstni strežnik lahko na istem grafikonu različnim serijam podatkov določi različne tipe grafikona. To je vidno tudi na sliki 6.2. Uporabnik

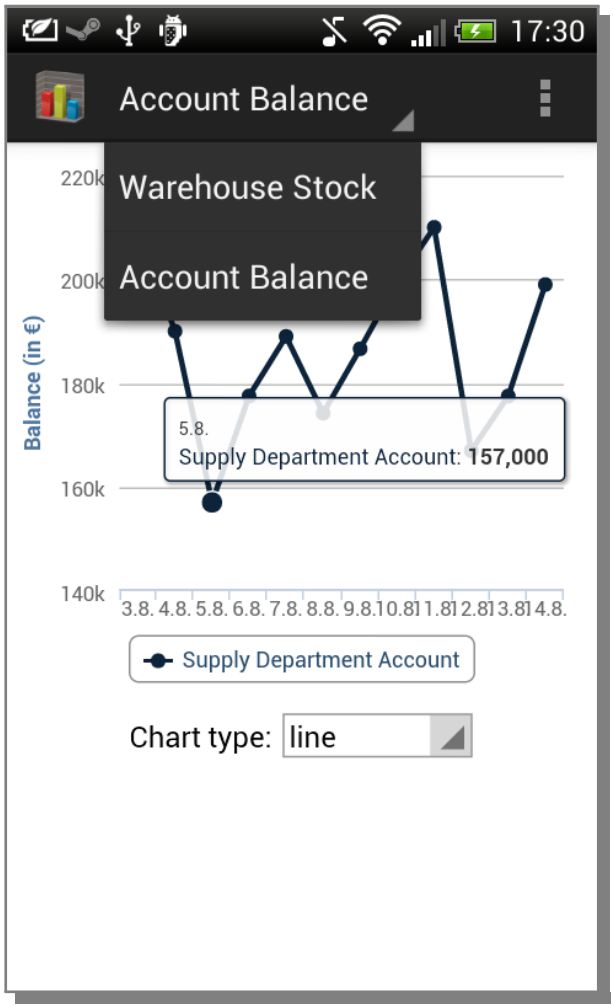
pa lahko zamenja tip grafikona samo na en tip za vse serije podatkov.

S klikom na ime določene serije v legendi se podatki v grafikonu lahko skrijejo ali pokažejo. To je uporabno, če želi uporabnik bolj natančno pogledati določene podatke ali pa je podatkov na grafikonu preveč in kontekstni strežnik prikaže samo nekatere. Uporabnik ima še vedno možnost, da si ogleda tudi ostale.

Podatki s sekcije na sliki 6.2 so na sliki 6.4 prikazani z drugim tipom grafikona. Skrita je tudi ena izmed serij podatkov, ki je bila prej prikazana. Prej skrita serija podatkov pa je sedaj prikazana.

S klikom na ime sekcije se odpre meni sekcij. V meniju sekcij so imena sekcij v vrstnem redu, ki ga je določil kontekstni strežnik – od najbolj uporabniku relevantne sekcije do najmanj. S klikom na meni sekcij se prikaže izbrana sekcija. Zaslonski izris menija sekcij je prikazan na sliki 6.5.

V meniju možnosti ima uporabnik možnost priti do nastavitev in možnost osvežitve podatkov. S klikom na osvežitev se sproži storitev prejemanja in pošiljanja podatkov, ki pred prejetjem novih podatkov pošlje kontekstnemu strežniku podatke o uporabi.



Slika 6.5: Zaslonski izris izbire sekcije; v ozadju pa je izbrana sekcija stanje na bančnem računu

6.5 Poslani podatki

Primer podatkov, ki jih naprava pošlje kontekstnemu strežniku:

```
{
  "deliveryID":314159,
  "latitude":"46.044900"
  "longitude":"14.489297"
  "log":[
    {
      "time":"14.08.2013 04:57:32.372",
      "section":"Warehouse Stock",
      "type":"application",
      "event":"load"
    },
    {
      "time":"14.08.2013 04:57:40.321",
      "section":" Warehouse Stock",
      "type":"chartType",
      "event":"line"
    },
    . . .
    {
      "time":"14.08.2013 04:57:55.482",
      "section":"Account Balance",
      "type":"application",
      "event":"pause"
    }
  ]
}
```

Med poslanimi podatki je identifikacijska številka dostavljenih podatkov, ki strežniku pove za katerega uporabnika gre, katero napravo ter katero aplikacijo in na katere podatke se podatki o uporabi nanašajo. Sledita ji ze-

mljepisna dolžina in širina lokacije naprave. Največji del poslanih podatkov predstavljajo podatki o uporabi.

Podatki o uporabi so v urejenem seznamu dogodkov. Vsak dogodek pove, kdaj se je nekaj zgodilo, kje se je zgodilo in kaj se je zgodilo. Vsaka uporabnikova akcija in vsak dogodek v aplikaciji je zabeležen v tem seznamu. Ko uporabnik odpre aplikacijo, se to zabeleži, nato pa se beleži vsak klik do zaprtja aplikacije, ki je tudi zabeležen. Aplikacija hrani podatke o uporabi dokler jih ne pošlje kontekstnemu strežniku.

Kontekstni strežnik uporabi prejete podatke za izračun uporabnikovega konteksta. Pravila določajo, kaj kateri podatki o uporabi pomenijo in kaj se zgodi v primeru pojava določenih podatkov. Kontekst uporabnika določi vrstni red sekcij, število sekcij, tipe grafikonov, vidnost serij podatkov in potrebe obveščanja. Ko je kontekst uporabnika izračunan, kontekstni strežnik aplikaciji pošlje nove podatke.

Poglavje 7

Sklepne ugotovitve

Razvita je bila kontekstno odvisna mobilna aplikacija, ki omogoča razvoj in preizkus kontekstnega strežnika. Definirane so bile naloge kontekstno odvisnih aplikacij ter s pomočjo teh nalog zasnovana aplikacija. Iz načrta kontekstnega strežnika bo z obstojem aplikacije mogoče nadaljevati z razvojem strežnika. Po razvoju se bo aplikacija uporabljala kot demonstracija delovanja kontekstnega strežnika in dostave relevantnih podatkov. Aplikacija bo preverila zastavljen model kontekstno odvisnega sistema. Preverila bo smiselnost nalog kontekstno odvisnih mobilnih aplikacij in služila za ugotavljanje novih. Ugotovitve, do katerih bo prišlo z uporabo aplikacije, bodo uporabljene pri razvoju naslednje kontekstno odvisne aplikacije. Z nekaj popravki in spremembami pa bi lahko bila aplikacija uporabljena kot poslovna aplikacija, kakor je predstavljena pri primeru delovanja.

Kot pri vsaki aplikaciji bo tudi pri tej potrebno popraviti morebitne programske hrošče, ki jih bodo zaznali uporabniki in razvijalci kontekstnega strežnika. Dodali bi lahko prikaz tabele in še kakšnega tipa grafikona ter zajem podatkov s še kakšnega senzorja naprave, kar bi omogočalo še večjo kontekstno odvisnost.

Potrebno bi bilo poskrbeti za vidik varnosti kontekstno odvisnega sistema.

Kontekstni strežnik in kontekstno odvisna aplikacija vsebujeta precej zaupnih podatkov. Na kontekstnem strežniku se nahajajo podatki o uporabi aplikacije, uporabnikovi lokaciji in podatki za prikaz v aplikaciji. Aplikacijo pa bi lahko kdo zlorabil za nedovoljen dostop do strežnika ali za prikaz zaupnih podatkov.

Lahko bi izboljšali delovanje obveščanja. Storitev prejemanja in pošiljanja podatkov je v primeru prepogostega izvajanja energijsko precej potratna. Obvestila bi lahko kontekstni strežnik posredoval preko sporočanja v oblaku Google (Google Cloud Messaging). To omogoča, da naprava takoj prikaže obvestilo in pri tem ne potrebuje v ozadju periodično ponavljajoče storitve za sprejemanje obvestil.

Za boljše določanje uporabnikovega konteksta bi lahko uporabljali koledar Google. Tako bi glede na dogodke v koledarju lahko razbrali kje in s kom je uporabnik. To bi izboljšalo lokacijsko informacijo in socialno situacijo konteksta situacije ter s tem celoten kontekst, kar bi pripomoglo k dostavi še bolj relevantnih podatkov.

Nadaljnjemu razvoju kontekstnega strežnika bi pripomogle nove kontekstno odvisne mobilne aplikacije ter tudi kontekstno odvisne aplikacije namenjene drugim napravam. Z pojavom računalništva v oblaku se briše meja med računalniki in mobilnimi napravami. Strežniki, ki se uporabljajo v računalništvu v oblaku, bi lahko služili tudi kot kontekstni strežniki. V prihodnosti bi kontekstni strežniki lahko s pomočjo strojnega učenja samodejno odkrivali in dodajali nova pravila za določanje konteksta. Uporabniki bi ta pravila ocenjevali s svojim načinom uporabe aplikacij. Na področju dostave relevantnih podatkov in kontekstne odvisnosti tako ostaja še ogromno možnosti za nadaljnji razvoj.

Literatura

[1] P.J. Brown, "Triggering information by context", *Personal Technologies*, št. 2, zv. 1, str. 1-9., 1998.

[2] J.R. Cooperstock, K. Tanikoshi, G. Beirne., T. Narine and W. Buxton, "Evolution of a Reactive Environment", v zborniku *1995 ACM Conference on Human Factors in Computing Systems*, Denver, 1995, str. 170-177.

[3] A.K. Dey, "Providing architectural support for building context-aware applications", Phd. Thesis, Georgia Institute of Technology, 2000.

[4] D. Flanagan, *Javascript: The Definitive Guide, Sixth Edition*, Kalifornija: O'Reilly Media, 2011.

[5] M. Gargenta, *Learning Android*, Kalifornija: O'Reilly Media, 2011.

[6] I. Horton, *Beginning Java, Java 7 Edition*, Indianapolis: John Wiley & Sons, 2011.

[7] C. Laskowski, A. Gognjavec, R. Rupnik, "Context Server as the Core of Context-Aware Applications", v zborniku *OTS 2013 Sodobne tehnologije in storitve*, Maribor, junij 2013, str. 92-99.

[8] R. Rupnik, "Model kontekstno odvisnih mobilnih aplikacij in opredelitev njihove vloge v informacijskem sistemu", Doktorska disertacija, Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, Ljubljana, 2002.

[9] R. Rupnik, M. Krisper, "Context-Aware Mobile Application Model", *Elektrotehniški vestnik*, št. 71, zv. 4, str. 215-219, 2004.

[10] (2013) Get the Android SDK. Dostopno na: http://developer.android.com/sdk/index.html

[11] (2013) Highcharts. Dostopno na: http://www.highcharts.com/

[12] (2013) The Java Language Environment. Dostopno na: http://www.oracle.com/technetwork/java/intro-141325.html

[13] (2013) Wikipedia: Android (operating system). Dostopno na: http://en.wikipedia.org/wiki/Android_(operating_system)

[14] (2013) Wikipedia: JSON. Dostopno na: http://en.wikipedia.org/wiki/JSON