



Št. naloge: 01647/2010

Datum: 15.03.2010

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

Kandidat: **IVAN TURKOVIČ**

Naslov: **LOKACIJSKO OZNAČEVANJE PODATKOV V SOCIALNIH OMREŽJIH
DATA GEO-TAGGING IN SOCIAL NETWORKS**

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

V diplomskem delu se posvetite spletnim in mobilnim aplikacijam za delitev vsebine v realnem času na določeni lokaciji. Poudarek storitve naj bo v možnosti definiranja svojih osebnih geolociranih točk, ki delujejo kot nekakšen virtualni oddajnik za vsebino. Z osebnimi točkami lahko uporabnik deli vsebino z svojimi prijatelji in javnostjo ne glede kjer se trenutno nahaja, pri tem pa ima na voljo možnost povezovanja in sinhronizacije z drugimi spletnimi družbenimi storitvami kot sta Twitter in Facebook. Za implementacijo sistema za upravljanje vsebin izberite programski jezik Ruby z razvojnim ogrodjem Ruby on Rails, ki v povezavi z relacijsko podatkovno bazo MySQL tvorita ogrodje spletne aplikacije. Za implementacijo mobilne aplikacije uporabite programski jezik Java in relacijsko podatkovno bazo SQLite.

Mentor:

prof. dr. Saša Divjak



Dekan:

prof. dr. Franc Solina

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Ivan Turković

**Lokacijsko označevanje podatkov v socialnih
omrežjih**

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: prof. Dr. Saša Divjak

Ljubljana, 2010

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani/-a _____Ivan_Turkovič_____

z vpisno številko _____63030072_____

sem avtor/-ica diplomskega dela z naslovom:

___Lokacijsko_označevanje_podatkov_v_socialnih_omrežjih___

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal/-a samostojno pod mentorstvom (naziv, ime in priimek)

_____prof._dr._Saša_Divjak_____

in somentorstvom (naziv, ime in priimek)

- 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 __15.6.2010__ Podpis avtorja/-ice: _____

Zahvala

Zahvaljujem se mentorju prof. dr. Saša Divjaku za vse nasvete in pomoč pri izdelavi diplomske naloge.

Posebna zahvala moji družini za vso skrb in podporo, ki mi jo je namenjala skozi celotno študijsko obdobje.

Kazalo

| | |
|---|-----------|
| POVZETEK | 2 |
| ABSTRACT | 4 |
| 1 UVOD | 6 |
| 2 IZBIRA RAZVOJNIH TEHNOLOGIJ | 10 |
| 2.1 Pregled spletnih tehnologij | 10 |
| 2.2 Izbor mobilne tehnologije | 13 |
| 2.3 Namestitev izbrane rešitve | 14 |
| 2.4 Urejevalnik kode..... | 14 |
| 2.5 Dodatna teoretična znanja za pomoč pri izdelavi rešitve..... | 15 |
| 3 NAČRTOVANJE PODATKOVNE BAZE, SPLETNE IN MOBILNE APLIKACIJE | 18 |
| 3.1 Spletna aplikacija | 19 |
| 3.1.1 Načrt..... | 19 |
| 3.1.2 Datotečna organizacija | 21 |
| 3.1.3 Načrtovanje uporabniškega vmesnika | 22 |
| 3.2 Mobilna aplikacija | 26 |
| 3.2.1 Načrt..... | 26 |
| 3.2.2 Načrtovanje uporabniškega vmesnika | 28 |
| 3.3 Podatkovna baza..... | 30 |
| 4 UPORABA STORITVE | 36 |
| 4.1 Spletna aplikacija | 36 |
| 4.2 Mobilna aplikacija | 37 |
| 4.3 Primer uporabe rešitve | 37 |
| 5 ZAKLJUČEK..... | 40 |
| SEZNAM SLIK..... | 42 |
| SEZNAM TABEL..... | 44 |

| | |
|------------------|----|
| LITERATURA | 46 |
|------------------|----|

Seznam uporabljenih kratic in simbolov

API - nabor javno dostopnih knjižnic

CGI - skupni prehodni vmesnik

CMS – sistem za upravljanje vsebine spletnih strani

CSS - določanje izgleda spletnih strani za posamezne oznake

GPS - globalni sistem za določanje lokacije

GUI – grafični uporabniški vmesnik

HTML – označevalni jezik za definiranje tekstnih spletnih strani

IDE – napredni urejevalnik programske kode

MVC – model programiranja, kjer je sistem razdeljen na model, pogled in kontroler

Ruby – skriptni programski jezik

RoR – ruby on rails, spletna tehnologija za izdelavo dinamičnih spletnih strani

XML – razširljivi označevalni jezik

Povzetek

Diplomsko delo opisuje spletno in mobilno aplikacijo za delitev vsebine v realnem času na določeni lokaciji. Tako mobilna kot spletna aplikacija ponujata možnost oddajanja in pregledovanja vsebin, pri čemer je spletna aplikacija bolj namenjena administraciji. Mobilna aplikacija pa izkorišča vse napredne tehnologije, ki se nahajajo v pametnih telefonih, tako da lahko natančno določimo lokacijo uporabnika in mu na podlagi le te postrežemo z ustrežno vsebino. Obe skupaj tvorita storitev za delitev statusnih sporočil in multimedijskih vsebin v realnem času. Vsa vsebina je tudi označena s podatki o lokaciji. Na podlagi teh podatkov se v ozadju poskušajo pridobiti povezave med podatki, ki so si lokacijsko blizu. Poudarek storitve je v možnosti definiranja svojih osebnih geolociranih točk, ki delujejo kot nekakšen virtualni oddajnik za vsebino. Z osebnimi točkami lahko uporabnik deli vsebino s svojimi prijatelji in javnostjo ne glede, kje se trenutno nahaja, pri tem pa ima na voljo možnost povezovanja in sinhronizacije z drugimi spletnimi družbenimi storitvami, kot sta Twitter in Facebook. Vsa vsebina se nahaja na strežniku, tako da si uporabnik lahko kadarkoli prenese vsebino. Storitev temelji na čim bolj enostavni delitvi vsebine s prijatelji, zato je osnovna stran namenjena časovnemu pregledu vsebin, ki jih delijo uporabnikovi prijatelji. Za implementacijo sistema za upravljanje vsebin je bil izbran programski jezik Ruby z razvojnim ogrodjem Ruby on Rails, ki v povezavi z relacijsko podatkovno bazo MySQL tvorita ogrodje spletne aplikacije. Za implementacijo mobilne aplikacije je bil uporabljen programski jezik Java, v ozadju pa se nahaja relacijska podatkovna baza SQLite. V nadaljevanju so podane še dodatne možnosti s smernicami za nadaljni razvoj storitve. Na koncu imamo opisanih nekaj različnih primerov uporabe v vsakdanjem življenju.

Ključne besede:

socialna omrežja, lokacija, označevanje podatkov

Abstract

This thesis describes a web and mobile application for sharing location-tagged content in real time. Both mobile and web applications offer the possibility of broadcasting and viewing content. The web-based application is more focused on the administration part. The mobile application uses all the advanced technology found in the new generation of smart phones, so you can accurately determine the user's location and with the solution can show only relevant content. All together form a service for sharing status messages and multimedia content in real time. Content is marked with location data. With all this information provided, the service is doing some background processing and trying to link relevant data near you. The focus of the service is to have the possibility of defining their personal location points, which acts as a virtual transmitter for the content. With the personal items user can share his personal items with his friends and the public no matter where currently is located, while having the option of integration and synchronization with other online social services like Twitter and Facebook. Content data is located on the server and user can download content at anytime. The solution is based on simple content sharing with friends. For the implementation of the content management system, the programming language Ruby and for the development framework Ruby on Rails, which in conjunction with the relational database MySQL form the web application framework. For the implementation of mobile applications, the programming language used was Java and the background is a relational database SQLite. Additional guidelines for further development of the service are following. Finally, we described various cases of use in everyday life.

Keywords:

social network, location, data tagging

1 Uvod

Socialna omrežja na spletu so se začela oblikovati že na samem začetku komercializacije interneta. Zgodnja socialna omrežja so nastala kot splošne spletne skupnosti. Eden izmed teh je bil Geocities, ustanovljen leta 1994. Te začetne skupnosti so bile usmerjene k združevanju ljudi in medsebojni interakciji v spletnih klepetalnicah, deljenju osebnih podatkov in informacij. Pri tem uporabniki niso imeli veliko vpliva na celotno izkušnjo. Pri prvih socialnih omrežjih je imela elektronska pošta veliko vlogo pri izmenjavi informacij.

Kmalu so se začele pojavljati nove metode socialnega mreženja, ki pa so spremenile model delovanja socialnih omrežij. Iz enostavnih, kjer si lahko priporočal informacije, v takšne, kjer dejansko upravljaš s svojo spletno prisotnostjo. Profili na teh socialnih omrežjih so začeli predstavljati fizično osebo, ki se je poistovetila s svojo spletno identiteto. Na tak način so uporabniki začeli deliti svoje osebne podatke. S tem se je začela zmanjševati meja med spletno prisotnostjo na socialnih omrežjih in zasebnim življenjem.

Ta socialna omrežja so se pojavila množično z vzponom Friendsterja leta 2002 in pa s Facebookom leta 2004. Počasi so ljudje začeli sprejemati socialna omrežja kot del njihovega življenja, s tem so tudi začeli svoje zasebne aktivnosti deliti z ostalimi uporabniki na spletu. To jim je omogočilo ostati v stiku z drugimi ljudmi, ki jih vsak dan redkeje srečujejo. Z rastjo popularnosti socialnih omrežjih so se začeli pojavljati novi trendi in tehnologije, ki omogočajo še bolj napredno funkcionalnost.

Z vzponom popularnosti Twitterja prihajajo v ospredje novi trendi v socialnih omrežjih, kot sta koncept deljenja vsebine v realnem času in glede na lokacijo. Koncept realnega časa omogoča uporabnikom prispevanje vsebine kadarkoli in kjerkoli, ki pa se potem oddaja po socialnem omrežju. Ta še posebej doživlja hiter vzpon rasti v zadnjem času. Povezan je s hitrim širjenjem naprednih mobilnih tehnologij, pametnih telefonov z velikimi zasloni in operacijskimi sistemi, nižanju cen prenosov podatkov prek mobilnih omrežij. Prav Twitter je eden izmed socialnih omrežij, ki je začel svoj vzpon z mobilnimi tehnologijami in odprtostjo razvoja tretjih ponudnikov. Je eden izmed prvih, ki je ponudil podatkovne tokove v realnem času. Zaradi tega lahko uporabniki dobivajo informacijo v skoraj enakem času, kot so bile oddane.

Drugi koncept deljenja aktivnosti je prikaz svoje trenutne lokacije z ostalimi. Storitev je še v povojih in doživlja rast v zadnjem času. Razlog za to leži v vgradnji GPS čipov v nove moderne mobilnike. S tem telefoni postajajo najbolj nepogrešljivo orodje za uporabo socialnih omrežij. Z njimi lahko ustvarjamo multimedijsko vsebino, jo prenašamo na socialna omrežja in spremljamo

dogajanje na socialnih omrežjih. Prva generacija socialnih storitev, ki sprejemajo lokacijo uporabnika, še vedno zahtevajo od njega ročno potrjevanje lokacije. Medtem ko pa naslednja generacija storitev že označuje avtomatično vsebino z lokacijo v realnem času ter na tak način sledi uporabniku.

Napredek v mobilni tehnologiji nam prinaša nove možnosti uporabe socialnih omrežij, ki se vse bolj usmerjajo v uporabo na mobilnih napravah. Pri tem se pojavljajo nove funkcionalnosti, kot so deljenje življenjskih izkušenj, ki se trenutno dogajajo, z lokacijsko označenimi uporabniško generiranimi multimedijskimi vsebinami, izmenjavo priporočil o lokacijah, pregled trenutne lokacije prijateljev in pa možnost spoznavanja ljudi v tvoji neposredni bližini.

Poleg uporabe socialnih omrežij v zasebne namene se je začela le ta širiti tudi v podjetja. Podjetja so spoznala moč socialnih omrežij kot eno izmed močnih orodij za izgradnjo podobe znamke. Največ uporabljajo socialna omrežja za ustvarjanje prepoznavnosti znamke, orodje za upravljanje spletnega ugleda znamke, pridobivanje novih uporabnikov, izobraževanje o novih tehnologijah in konkurentih ter vodilno orodje za generiranje potencialnih možnosti.

Na podlagi pregleda zmožnosti trenutnih socialnih omrežij še vedno obstajajo nove storitve, ki se še niso uveljavile. Med njimi je tudi možnost oddajanja vsebine za določeno lokacijo. Vsa socialna omrežja do sedaj so bila namenjena oddajanju vsebine, tako da so lahko drugi uporabniki sprejemali informacije od ostalih uporabnikov. Vsak uporabnik je lahko spremljal v spletu, kar drugi počnejo in kar delijo z ostalimi. Pri tem pa raste količina informacij in je težko slediti vsem spremembam. Ni pa možno določiti ene fizične lokacije za uporabnika, da bi lahko delil svojo vsebino z ostalimi, ki se trenutno nahajajo na tisti lokaciji. Podobno kot so danes radijske postaje omejene z dosegom oddajnih anten in s tem tudi vsebinsko orietirane na ta prostor, bi tudi uporabniki lahko imeli možnost določiti svoj prostor, kjer bi lahko delili svojo vsebino z ostalimi. Poleg tega bi tudi naključnim osebam olajšali spoznavanje novega okolja, saj bi se s tako pridobljenimi informacijami lažje orientirali v prostoru. Možnosti za uporabo je veliko. Zaradi tega se bo v diplomski nalogi obravnavala možnost uporabe storitve v realnosti in skušal ponuditi platformo, na kateri bi se lahko takšna storitev izvajala.

Celotno diplomsko delo je razdeljeno na več vsebinskih delov. Začnemo s pregledom in izbiro možnih tehnologij za izdelavo delujočega sistema. Za izdelavo sistema potrebujemo dve ločeni tehnologiji. Prva je izdelava spletne aplikacije. Možnosti za izbiro spletnih tehnologij je veliko, pri tem pa se dotaknemo samo sledečih tehnologij: PHP, Ruby on Rails, J2EEE, ASP.NET ter Django. Vse spletne tehnologije naj bi bile enakovredne, ampak kot se izkaže ni temu tako. Vsaka tehnologija ima svojo posebnost in hitrost izvajanja, od česa je tudi odvisno, kako odzivna bo aplikacija. Zaradi zahtev izvajanja v realnem času sem se odločil, da bom pri izdelavi diplomskega dela uporabil spletno tehnologijo Ruby on Rails skupaj z relacijsko podatkovno bazo MySQL.

V tem sklopu bom skušal navesti razloge, zakaj sem se odločil za in tudi navodila za vzpostavitev delujočega okolja, ki so potrebna za začetek razvoja spletne tehnologije. Poleg spletne tehnologije je potrebno izbrati tudi orodja, katera se bodo uporabljala pri izdelavi. Med njimi je najbolj pomemben urejevalnik kode IDE, ki nam lahko zelo pomaga pri hitrejši in bolj optimalni izdelavi programske kode. Na tem področju obstaja veliko različnih rešitev, ampak glede na specifičnost spletne tehnologije in programskega jezika se lahko omejimo na nekaj rešitev. Najbolj pomembna primerjava je med naprednimi tekstovnimi urejevalniki. Drugi del je izdelava mobilne aplikacije. Na voljo imamo množico različnih rešitev in operacijskih sistemov. Pri tem se bo primerjalo nekaj sledečih rešitev, kot so iPhone, Android, Blackberry in Symbian. Zaradi tehničnih razlogov in potrebnega dodatnega znanja sem se odločil za izdelavo Android aplikacije. Poleg tega bom zraven še omenil potrebna dodatna znanja, ki so potrebna za izdelavo in razvoj spletnih aplikacij.

Naslednji del diplomskega dela je namenjen načrtovanju spletne in mobilne aplikacije ter podatkovne baze. Pred tem pa moramo natančno opredeliti vse potrebne zahteve, ki jih mora rešitev sistema izpolnjevati. Tako je na začetku definirana vsa funkcionalnost in uporabniški grafični vmesnik (GUI). Celotna rešitev temelji na dinamičnem spreminjanju vsebin. Temu je tudi potrebno definirati procese in korake delovanja, podobno kot pri klasični aplikaciji. Izdelati je potrebno v ozadju sistem za upravljanje z vsebino (CMS), ki nam bo olajšal uporabo storitve. Poleg tega je potrebno pozornost nameniti mobilnemu delu, kjer veljajo omejitve z velikostjo zaslona. V nadaljevanju so ponazorjeni procesi izvajanja na posameznih nivojih prikazovanja strani. Z zaključkom definiranja sistema sledi implementacija posameznih delov sistema. Na koncu imamo še definiran in opisan konceptualni model podatkovne baze.

V nadaljevanju sledi obravnava sestavnih delov spletne in mobilne aplikacije. Pri tem predstavimo prednosti uporabe Ruby on Rails tehnologije za spletni del ter uporabo Java na Android platformi. Namembnost rešitve je omogočiti uporabniku, da lahko enostavno določi svoje oddajne točke in možnost sprejemanja informacij iz ostalih točk. Zato v nadaljevanju načrtujemo zgradbo takšne rešitve, v kateri so predstavljeni razširitveni moduli in možnost oddajanja v ostala socialna omrežja. Za lažji nadaljni razvoj je priložen tudi način, kako lahko izdelamo javno dosopno knjižnico (API), razširimo rešitev tudi na druga področja. Na koncu se nahaja opis izvedbe nekaterih delov in postopkov uporabe.

Pred koncem diplomske naloge prikažemo nekaj primerov uporabe rešitve na praktičnih primerih. Primeri uporabe so za različne profile uporabnikov: uporaba samo spletne aplikacije, uporaba mobilne aplikacije, uporaba obeh in pa intenzivno deljenje aktivnosti z ostalimi.

V zaključnem delu diplomske naloge naredimo pregled čez zastavljene cilje in

trenutno funkcionalnost sistema. Pri tem so tudi definirane smernice za nadaljni razvoj rešitve.

2 Izbira razvojnih tehnologij

V tem delu so predstavljene trenutne aktualne programske tehnologije za izgradnjo naprednih spletnih aplikacij. Na podlagi izbire namestimo potrebne programske pakete, programsko opremo in podatkovno bazo. Opisan je pregled trenutnih razvojnih okolij za mobilne naprave, za izdelavo je bila izbrana ena izmed prespektivnih platform.

Preden začnemo z razvojem je še potrebno narediti en korak, in sicer izbira ustreznega urejevalnika kode (IDE). Pri tem še dodamo nekaj virov informacij, ki so namenjene utrjevanju teoretičnega znanja izbranih tehnologij. Ko s tem končamo, smo pripravljeni na načrtovanje in dejansko izvedbo določene rešitve.

2.1 Pregled spletnih tehnologij

Začetki svetovnega spletnega omrežja segajo v leto 1990. Pri načrtovanju le tega so imeli idejo o statični vsebini, statične spletne strani so vsebovale povezave na druge strani. Vse skupaj pa je prebivalo na spletnih strežnikih, ki so stregli vsebino. Za vsako spremembo so morali avtorji ročno spreminjati vsebino datoteke. Na možnost dinamičnega vnašanja vsebine je bilo potrebno počakati do predstavitve skupnega prehodnega vmesnika (CGI), ki je omogočil možnost komunikacije zunanjih aplikacij s spletnim strežnikom. S tem so začele nastajati različne rešitve pri generiranju dinamične vsebine.

V začetku so programerji uporabljali programske jezike, ki so jih poznali iz razvoja računalniških aplikacij. Zaradi različnih omejitev v razvoju pa so v osredje začeli prihajati skriptni jeziki, ki pa so imeli to možnost, da so se pognali takoj. Razvijalci so pri tem največ prihranili na času, saj ni bilo potrebno sprotno prevajanje in posodabljanje prevedenih objektov in s tem celotne rešitve. Največkrat pri razvoju nove rešitve ponovno uporabljamo delčke kode, ki smo jo uporabili že v preteklosti. Iz tega so se sčasoma začele nabirati programske knjižnice, ki pa so z leti prerasle v programsko ogrodje za lažjo izvedbo rešitve.

Možnosti na trgu je veliko. Med njimi najbolj pogosto naletimo na naslednja programska ogrodja: PHP, ASP.NET, J2EE (Java Enterprise Edition) ali Ruby on Rails. Slednji je najmlajši med naštetimi in je bil razvit tudi v razmeroma novem programskem jeziku Ruby. V svetu razvijalcev obstaja ogromno različnih mnenj in debat o tem, katera rešitev je najprimernejša za razvoj spletnih aplikacij. Omeniti je vredno, da ima vsaka svoje prednosti in slabosti. Odločitev za izbiro najbolj primernega ogrodja je bila omejena na zahteve izdelave rešitve.

Kriteriji, po katerih se je ocenjevala ustreznost posamezne spletne tehnologije:

- Model, pogled, kontroler (MVC) je model grajenja platforme. Temelji na razdelitvi rešitve na tri dele, in sicer ločitev na podatkovni model, poslovno logiko in uporabniški vmesnik. Na splošno je to dobra rešitev, saj z uporabo razdelimo kodo na module in na možnost ponovne uporabe kode.
- Programski jezik, v katerem je ogrodje napisano, omogoča napredne zmogljivosti, kot je metaprogramiranje.
- Omogoča dostop do podatkovne baze in možnost preslikavanja objektov v podatkovne vnose v bazi.
- Preslikavo URL naslovov. Je mehanizem, s katerim lahko ogrodje interpretira URL naslov. Pri tem je potrebno poudariti možnost naprednih preslikav z regularnimi izrazi (regex).
- Sistem spletnih predlog. Dinamične strani so sestavljene iz dveh funkcionalnih enot –dinamičnega in statičnega. Z možnostjo vstavljanja spletnih predlog v dinamično zmanjšamo količino podvajanja in lažje urejamo vsebino na enem mestu.
- Ajax je okrajšava za Asinhroni Javascript in XML. Je tehnika pri spletnem razvoju, s katero dosežemo interaktivnost spletnih aplikacij. S tem postanejo aplikacije bolj odzivne in ni potrebe po ponovnem nalaganju celotne strani. Zmanjšamo obremenitev strežnika in količino pošiljanja podatkov.
- Spletne storitve omogočajo hiter razvoj programskih knjižnic namenjenih komunikaciji z zunanjimi aplikacijami. Pri tem omogoča enostavno komunikacijo v XML ali JSON podatkovnemu formatu.

V tabeli 2.1 je prikazana primerjava med različnimi spletnimi tehnologijami po predlaganih kriterijih.

Iz tabele 2.1 ni takoj razvidno, za katero tehnologijo se naj odločimo. Pri tej diplomski nalogi smo se odločili za spletno tehnologijo Ruby on Rails. Ta spletna tehnologija ima poleg zgornjih zahtev še eno prednost pred ostalimi, in sicer hitro izdelavo prototipnih aplikacij. Ta lastnost je pomembna pri agilnih metodah razvoja programske opreme, ki pa so bile uporabljene pri izdelavi rešitve.

| | PHP | J2EE | ASP.NET | Ruby on Rails |
|--|--------------------------|---|--|---------------|
| MVC model | Da (ni privzeti način) | Da (zapletena logika) | Da (dodatne knjižnice) | Da |
| Metaprogramiranje | Da (ni veliko v uporabi) | Ne (možno s skriptnim jezikom, ki je nastal v Javi) | Odvisno od programskega jezika v ozadju (F#) | Da |
| Enostaven dostop do podatkovne baze | Da (Zend) | Da | Da | Da |
| Preslikava URLjev | Da | Da (potrebno je zelo natančno definirano) | Da | Da |
| Sistem spletnih predlog | Da | Da | Da | Da |
| AJAX podpora | Da | Da | Da | Da |
| Spletne storitve | Da (SOAP) | Da | Da | Da (XML) |

Tabela 2.1: Primerjava različnih spletnih tehnologij z zahtevami spletne aplikacije.

Poleg tega ima še nekaj dodatnih prednosti:

- Celotna tehnologija je zasnovana na odprti kodi.
- Ima veliko bazo podpore s strani razvijalcev.
- Zaradi velike podpore in odprtosti se sproti odpravljajo hrošči v kodi, pri tem pa je obvezno testiranje za vsak popravek.
- Možno izvajanje na vseh platformah.
- Na spletu se nahajajo velike knjižnice različnih dodatkov in vtičnikov.
- Zaradi podprtosti metaprogramiranja v programskem jeziku Ruby je možno spreminjati praktično vse razrede, vključno s sistemskimi razredi.
- To ga naredi zelo prilagodljivega.

Kljub veliko prednostim ima tudi nekaj pomanjkljivosti:

- Ni možnosti prevajanja kode in s tem tudi počasnejše izvajanje.
- Napredno metaprogramiranje je tudi lahko ovira, saj lahko kjerkoli spremenimo definicijo poljubnega razreda.
- Ni še dovolj »zrel« za uporabo v podjetniških sistemih.

2.2 Izbor mobilne tehnologije

Izbor tehnologije za razvoj mobilne aplikacije je potekala nekoliko drugače, saj razvoj na mobilnih napravah običajno poteka drugače. Tukaj smo omejeni s prevladujočo tehnologijo izbrano s strani razvijalcev operacijskega sistema.

Izbira je potekala med štirimi različnimi rešitvami: iPhone, Android, Blackberry in Symbian. Za izbiro mobilne aplikacije smo zastavili drugačne kriterije. Ob tem je bilo upoštevano dejstvo, da gre za napredno storitev, ki za delovanje potrebuje funkcije telefona, ki pa so trenutno vgrajene v naprave višjega cenovnega razreda. Med te funkcije štejemo možnost pridobivanja prostorskih koordinat s pomočjo GPS-a, zaslon na dotik, ki nam omogoča enostavno pomikanje po mapi in omejitve pri razvojnem okolju. Poleg tega je tudi pogojna možnost vnašanja znakov s fizično tipkovnico.

V nadaljevanju, v tabeli 2.2, je prikazana primerjava razširjenosti kriterijev na mobilnih napravah z različnimi operacijskimi sistemi.

| | iPhone | Android | Blackberry | Symbian |
|--------------------|--------------------------------------|----------------|------------------------------------|---------------------------------|
| GPS enota | Da | Da | Da (samo določeni modeli) | Da (samo določeni modeli) |
| Programski jezik | Objective-C | Java | Java (J2ME + blackberry knjižnice) | C++ (Qt za uporabniški vmesnik) |
| Zaslon na dotik | Da | Da | Da (določeni modeli) | Da (določeni modeli) |
| Razvojno okolje | Xcode (omejeno na Apple računalnike) | Eclipse | Eclipse, Blackberry IDE | Eclipse, ostale možnosti |
| Fizična tipkovnica | Ne | Ne (par izjem) | Da | Da |

Tabela 2.2: Primerjava funkcionalnosti posameznih mobilnih platform.

Za rešitev mobilne aplikacije v okviru diplomske naloge je bil izbran operacijski sistem Android. Za uspešen zagon in sprejem storitve v širšo množico je še vedno potreben razvoj za vse možne rešitve.

2.3 Namestitev izbrane rešitve

Z določitvijo potrebnih razvojnih tehnologij je povezana tudi izbrana rešitev za razvojno okolje in potrebna programska oprema. Veliko vlogo pri izbiri je imela odprtost programske opreme.

Za razvoj je potrebno vzpostaviti dve okolji: razvojni računalnik in strežnik. Pri obeh okoljih je na začetku potrebno vzpostaviti vse nastavitve različnih paketov, tako da bodo delovali pravilno. Postopkov za nameščanje in konfiguracijo razvojnega okolja je več. Ob tem je pomembno tudi podrobno poznavanje izbranega postopka, kajti v nasprotnem primeru se lahko zgodi, da bo potrebno še enkrat začeti nastavljanje od začetka.

Na podlagi izkušenj je za diplomsko nalogo (delo) za razvojno okolje pri razvoju spletne in mobilne aplikacije izbran operacijski sistem Ubuntu. Pri tem se je za izbiro strežnika izbral Apache spletni strežnik, ki je na voljo kot odprta koda. Zaradi dejstva, da gre za izvajanje dinamičnih spletnih strani, je bil uporabljen CGI vtičnik Phusion Passenger. Ta med drugim omogoča povezavo med Apache spletnim strežnikom in spletno tehnologijo Ruby on Rails. Tako da je potrebno naložiti še podporo za programski jezik Ruby, spletno ogrodje Ruby on Rails, orodji rake in gem. Za podatkovno bazo smo uporabili MySQL. Programska rešitev Git je namenjena za nadzor nad verzijami različic kode.

Za dostavljanje kode smo uporabili ssh povezavo v navezi z orodjem Capistrano. Orodje Capistrano je napisano v ruby programskem jeziku, s katerim lahko olajšamo dostavljanje različic kode na strežnik.

2.4 Urejevalnik kode

Pri razvoju programskih rešitev potrebujemo urejevalnik kode, ki nam čimbolj olajša proces programiranja in nas sproti opozarja na pomanjkljivosti v kodi. Danes so se predvsem razširila integrirana razvojna okolja (IDE), ki nam ponujajo nešteto možnosti in funkcij za olajšanje pri delu. Nekateri so namenjeni le kot pomoč pri pisanju kode, avtomatičnemu dopolnjevanju, barvnemu označevanju rezerviranih besed, drugi – bolj napredni – pa nam ponujajo napredne zmogljivosti, kot je sprotno prevajanje in opozarjanje na napake, razhroščevanje kode in možnost izgradnje grafičnih uporabniških vmesnikov (GUI). Po drugi strani pa ta nešteta množica funkcionalnosti poveča nepreglednost pri razvoju in se pogosto znajdemo pred tem, da več pozornosti namenimo razvojnemu okolju kot kodi. Ne smemo pa zanemariti dejstva, da moderni IDE urejevalniki porabijo veliko sistemskih sredstev za svoje delovanje in s tem hkrati upočasnjujejo delovanje celotnega sistema.

Pri tem lahko omenimo določene IDE rešitve na področju razvoja spletne aplikacije s spletno tehnologijo Ruby on Rails:

- NetBeans,
- Eclipse,
- TextMate (na voljo samo za Mac OS),
- RedCar,
- e-text editor,
- Aptana Studio.

Iz naštetih smo se odločili za RedCar urejevalnik. Odločitev za izbiro tega urejevalnika pred drugimi je enostavnost odpiranja datotek v projektu. RedCar je odprtokodna rešitev za IDE okolje kot odgovor na TextMate, ki pa je bil narejen samo za Mac OS operacijski sistem. Enostavnost uporabe in možnost razvoja dodatnih vtičnikov sta ena izmed glavnih razlogov, zakaj je postal razširjen med ruby razvijalci. Pri tem RedCar ni samo kopija TextMate za druga okolja, saj je popolnoma razvit v ruby programskem jeziku in za delovanje potrebuje JRuby.

Za razvoj mobilne aplikacije ni veliko možnosti na izbiro. Uradna podpora je na voljo edino za Eclipse IDE in s tem tudi edina izbira za izdelavo aplikacije.

2.5 Dodatna teoretična znanja za pomoč pri izdelavi rešitve

Za zaključek tega poglavja se še dotaknimo vprašanja glede pridobivanja znanja za razvoj spletnih in mobilnih aplikacij.

Najbolj priporočljiv vir za pridobivanje informacij o razvoju v RubyOnRails je spletna stran RailsGuides, kjer imamo razložene osnove delovanja in izgradnje demonstracijskih spletnih aplikacij. Poleg tega imamo na voljo video predavanja na strani Railscasts. Predvsem gre za krajša predavanja, kjer v vsakem nadaljevanju opišejo delovanje določenga področja. Na Ruby toolbox imamo na voljo urejene vtičnike za različne funkcionalnosti, ki pa so logično razporejene po priljubljenosti.

Ker je celotno Ruby on Rails ogrodje na voljo kot odprta koda, se lahko za naprednejše razumevanje spustimo v branje kode in s tem opazujemo, kako najboljši razvijalci uporabljajo ruby programski jezik. Poleg ogrodja pa je priporočljivo spremljanje drugih odprtokodnih rešitev narejenih z ruby programskim jezikom, ki se nahajajo na GitHub spletni strani.

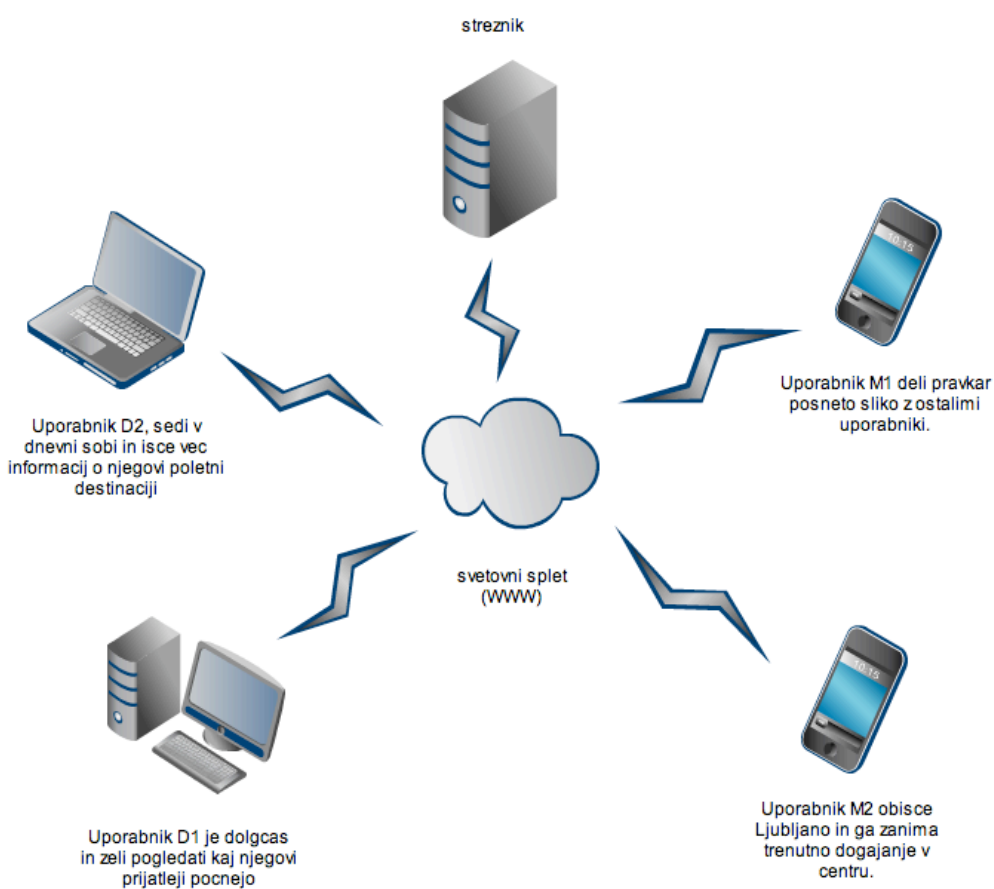
Pri izdelovanju diplomske naloge je bila uporabljena tudi naslednja literatura: The Rails way, The Ruby way in pa žepni izdaji Rails in MySQL za pomoč pri hitrem pregledu funkcionalnosti.

Z izdelavo mobilne aplikacije za Android se lahko seznanimo na uradni strani. Tam se že nahajajo navodila in pomoč pri razumevanju izdelave aplikacije. Poleg tega pa si lahko pomagamo z naslednjo literaturo: *Beginning Android*, *Pro Android* in *Professional Android Application development*.

3 Načrtovanje podatkovne baze, spletne in mobilne aplikacije

Pri izdelavi najprej izdelamo shemo delovanja sistema. Nato začnemo z načrtovanjem programske rešitve za problem. Sistem je sestavljen iz strežnika, računalnikov in mobilnih naprav, ki so povezani v svetovno omrežje. Mobilne naprave za uporabo potrebujejo vgrajen GPS čip ali drugo rešitev za določanje položaja.

Shema delovanja sistema je prikazana na sliki 3.1



create and share your own diagrams at gliffy.com



Slika 3.1: Shema delovanja sistema izmenjave podatkov na podlagi lokacije uporabnikov.

Uporabnik M1 uporablja svojo mobilno napravo za oddajanje multimedijskih slik na podlagi svoje trenutne lokacije. Uporabnik M2 je obiskovalec, ki s

pomočjo vgrajenega GPS čipa v mobilni napravi gleda, katere dejavnosti so trenutno aktualne v njegovi bližini. Uporabnik D1 se prijavi na storitev preko svojega osebnega računalnika in preverja, kaj so njegovi prijatelji počeli danes. Uporabnik D2 načrtuje naslednje počitnice, ker ima na voljo možnost pregleda aktivnosti po mapi.

Za izvedbo celotnega sistema so določeni kriteriji, ki jih mora rešitev vsebovati, da bo pokrito delovanje celotnega sistema. Osnovni pogoj za delovanje je možnost lokacijskega označevanja vseh podatkov, ki ji uporabniki delijo in možnost definiranja oddajnih točk »hotspots« za lažje raziskovanje v prostoru.

K osnovnim kriterijem je potrebno dodati tudi naslednje:

- Vsak uporabnik sistema se mora registrirati pred začetkom uporabe.
- Spletna in mobilna aplikacija omogočata deljenje aktivnosti in drugih multimedijjskih vsebin z ostalimi uporabniki.
- Spletna aplikacija vsebuje napredne možnosti nadzora na objavljeno vsebino ali, če se je uporabnik zmotil, popraviti vsebino.
- Mobilna aplikacija mora biti zmožna pridobiti trenutno lokacijo mobilne naprave in z njo lokacijsko označiti vsebino.
- Na spletni in mobilni aplikaciji mora biti pregled aktivnosti prikazan na mapi in, če je možno, tudi osveževati v realnem času.
- Celotna rešitev temelji na izmenjavi informacij, zato je potrebna tudi možnost dodajanja uporabnikov kot tvojih prijateljev.
- Storitev je zmožna dodati zunanja socialna omrežja in zmožna posodabljeni vsebino.
- Enostavno definiranje oddajnih točk na obeh sistemih.

3.1 Spletna aplikacija

3.1.1 Načrt

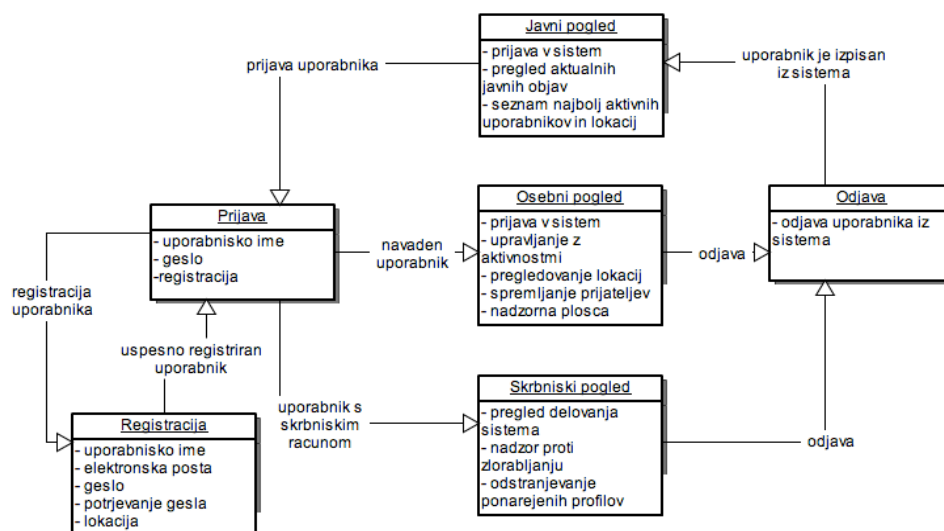
Pri načrtovanju je potrebno definirati posamezne funkcije in pa delovanje sistema. Na sliki 3.2 je predstavljen načrt za spletno aplikacijo deljenja lokacijsko označenih informacij. Na njej je ponazorjen globalen pregled delovanja aplikacije in možna stanja sistema. Stanje, v katerem se sistem nahaja, lahko odseva generiranje ene ali več dinamičnih strani in pripadajočih akcij. Pri tem je potrebno tudi definirati dva pogleda – osebni in javni pogled.

Javni pogled sestavljajo vse spletne strani in akcije, ki si jih lahko ogledajo tudi uporabniki, ki niso registrirani v sistem. Med njimi so naslednje strani:

- registracija in prijava v storitev,
- pregled trenutno aktualnih javnih objav,
- najbolj aktivni uporabniki in oddajne točke.

Osebni pogled je namenjen registriranim uporabnikom. Za dostop do svojega osebnega pogleda se mora uporabnik uspešno prijaviti v sistem. Osebni pogled omogoča ustvarjanje, urejanje in pregledovanje vsebin. Komponente, ki sestavljajo osebni pogled:

- Razdelitev funkcionalnosti na tri dele: aktivnosti, lokacije in prijatelji.
- Uporabnik ima na voljo tudi nadzorno ploščo svojega računa.
- Pri aktivnosti je možno deljenje aktivnosti, dodajanje oddajnih točk, pregled zadnjih objav svojih prijateljev in njihovih lokacij. Poleg tega imajo še možnost komentiranja posameznim uporabnikom.
- Za lokacijo imajo na voljo iskanje po lokaciji, pregled aktivnosti na iskani lokaciji ter enostavno navigacijo po mapi.
- Pri prijateljih je na voljo pregled aktivnosti za prijatelje in ločen pregled javnega dogajanja. Na voljo pa je tudi seznam prijateljev in enostaven način dodajanja novih prijateljev.
- Uporabnik ima svoj osebni profil.



create and share your own diagrams at gliffy.com

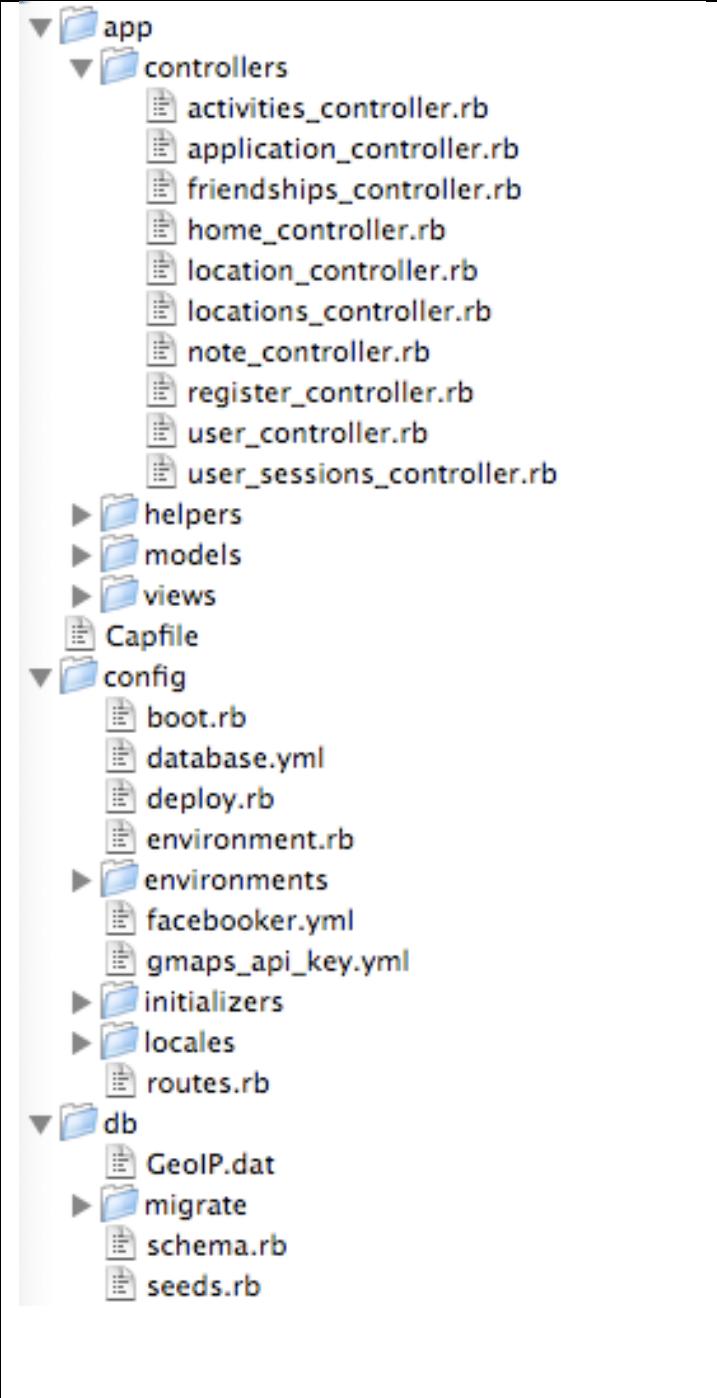


Slika 3.2: Načrt stanj spletne aplikacije za izmenjavo lokacijsko označenih podatkov v realnem času. Sistem je sestavljen iz množice stanj, v katerem se lahko nahaja uporabnik. Prehodi med stanji so označeni kot akcije, ki jih uporabnik izvede.

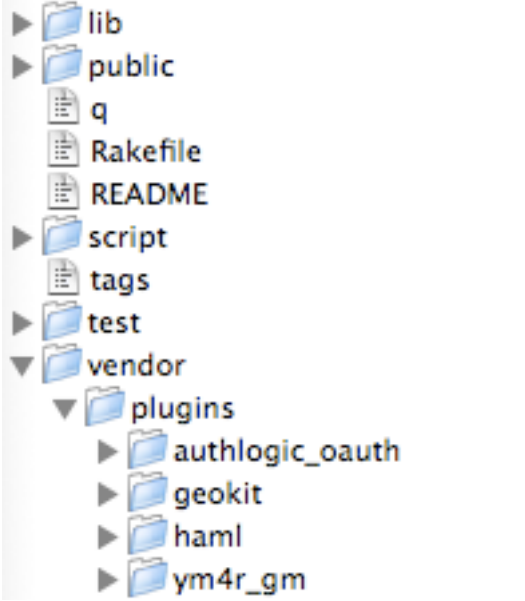
Poleg dveh javnih funkcionalnih enot obstaja v ozadju še skrbniški pogled. Ta skrbi, da ne prihaja do zlorabe storitve in redno pregleduje delovanje sistema. Na voljo sta dve glavni opravili: odstranjevanje ponarejenih profilov in podpora uporabnikom.

3.1.2 Datotečna organizacija

Zaradi lažje preglednosti je pri spletni tehnologiji definirana datotečna hierarhija. S tem prihranimo čas pri vzpostavitvi razvojnega okolja in novi razvijalci se lahko hitreje seznanijo z delovanjem sistema. Zato datotečna struktura na sliki 3.3 in 3.4 predstavlja uvod v spoznavanje delovanja spletne storitve.

| | |
|--|--|
|  | <p>Mapa app sledi MVC smernicam in loči na podatkovni model v mapi model, poslovno logiko v mapi controllers in kodo za dinamično generiranje v mapi views. Poleg pa se nahaja še mapa helpers, kjer je koda, ki si deli uporabo med temi logičnimi sklopi. Mapa controllers je razširjena in prikazuje kontrolerje za posamezno funkcionalnost.</p> <p>Capfile je nastavitvena datoteka za dostavljanje kode.</p> <p>V mapi config se nahajajo nastavitvene datoteke. Boot.rb je namenjen nastavitvam ob zagonu. Database.yml vsebuje podatke za povezavo s podatkovno bazo. Environment.rb poskrbi za nalaganje dodatnih knjižnic. Facebooker.yml vsebuje nastavitve za povezavo s Facebook API-jem. gmaps_api_key.yml vsebuje ključ za povezavo z Google Maps. Routes.rb je ena izmed pomembnejših nastavitev, saj so v njej definirane URL preslikave.</p> <p>V mapi db se nahajajo definicije za migracijo podatkovne baze. GeoIP.dat vsebuje lokacije IP naslovov.</p> |
|--|--|

Slika 3.3 Datotečna struktura spletne aplikacije.

| | |
|---|--|
|  | <p>V mapi lib se nahajajo vsi dodatni programski razredi, ki so potrebni pri delovanju sistema, ampak jih drugam logično ne moremo uvrstiti. V mapi public se nahajajo vse statične vsebine, kot so spletne strani HTML, javascript in css datoteke.</p> <p>V mapi test se nahajajo datotetke z razredi, ki so namenjeni testiranju delovanja sistema. Vendor mapa vsebuje vse dodatne vtičnike, ki so potrebni za delovanje spletne aplikacije. Authlogic_oauth za Oauth povezovanje računov, geokit za pomoč pri pridobivanju lokacijskih podatkov, haml je vmesni skriptni jezik za prevajanje v dinamične strani in ym4r_gm je pomoč pri izdelavi interaktivnih map.</p> |
|---|--|

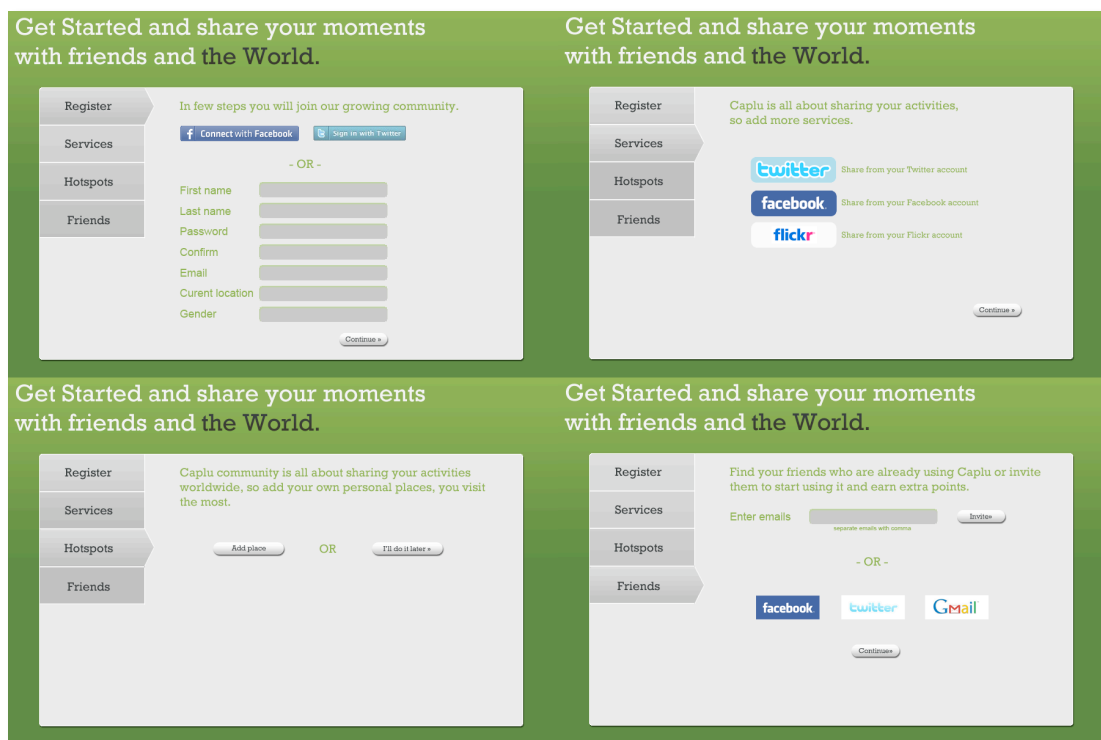
Slika 3.4 Prikazuje nadaljevanje datotečne strukture spletne aplikacije.

3.1.3 Načrtovanje uporabniškega vmesnika

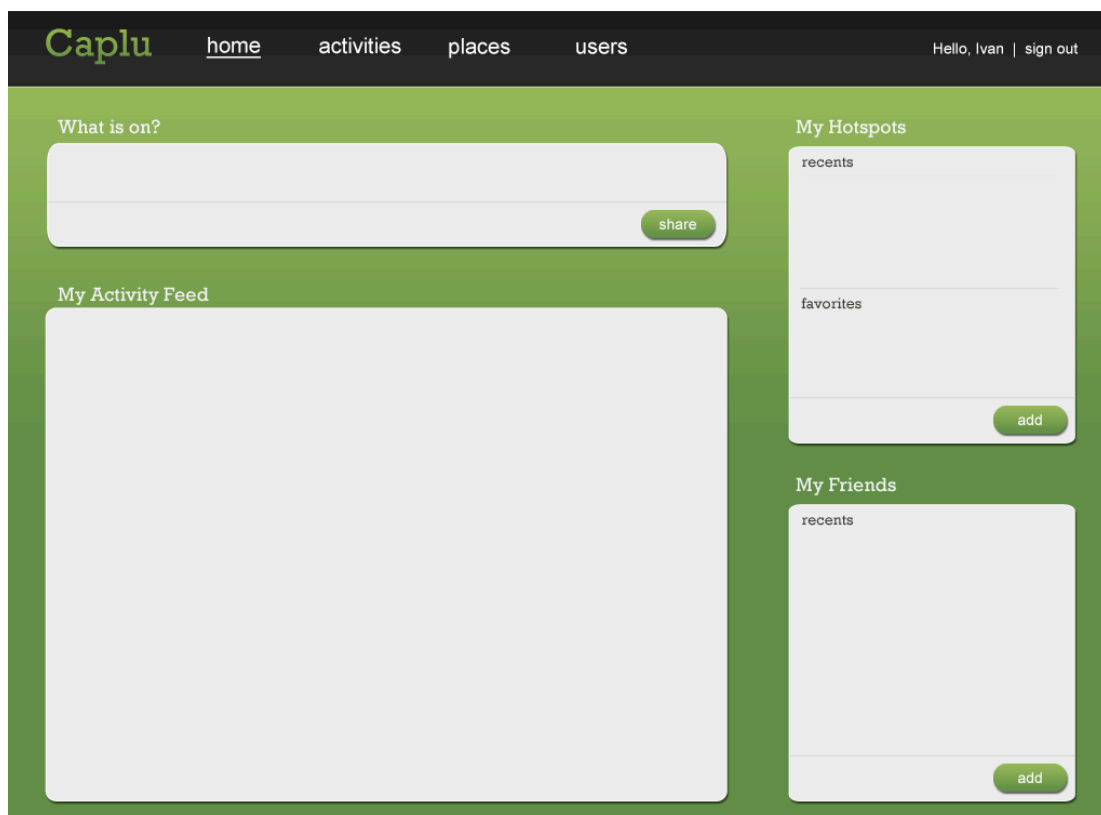
Spletna aplikacija poleg dobro načrtovane zasnove potrebuje tudi pregleden in uporaben grafični vmesnik. Celoten grafični vmesnik je bil narejen tako, da dopolnjuje izdelan po predlogi načrta spletne aplikacije. Pri tem je bilo potrebno upoštevati, katere funkcije storitve se bojo več uporabljale.

Na sliki 3.5 so vidni štirje koraki, ki so potrebni za registracijo uporabnika. Registracija uporabnika je eden izmed najbolj pomembnih elementov, saj nepreglednost registracijskega postopka odvrne uporabnika že na začetku.

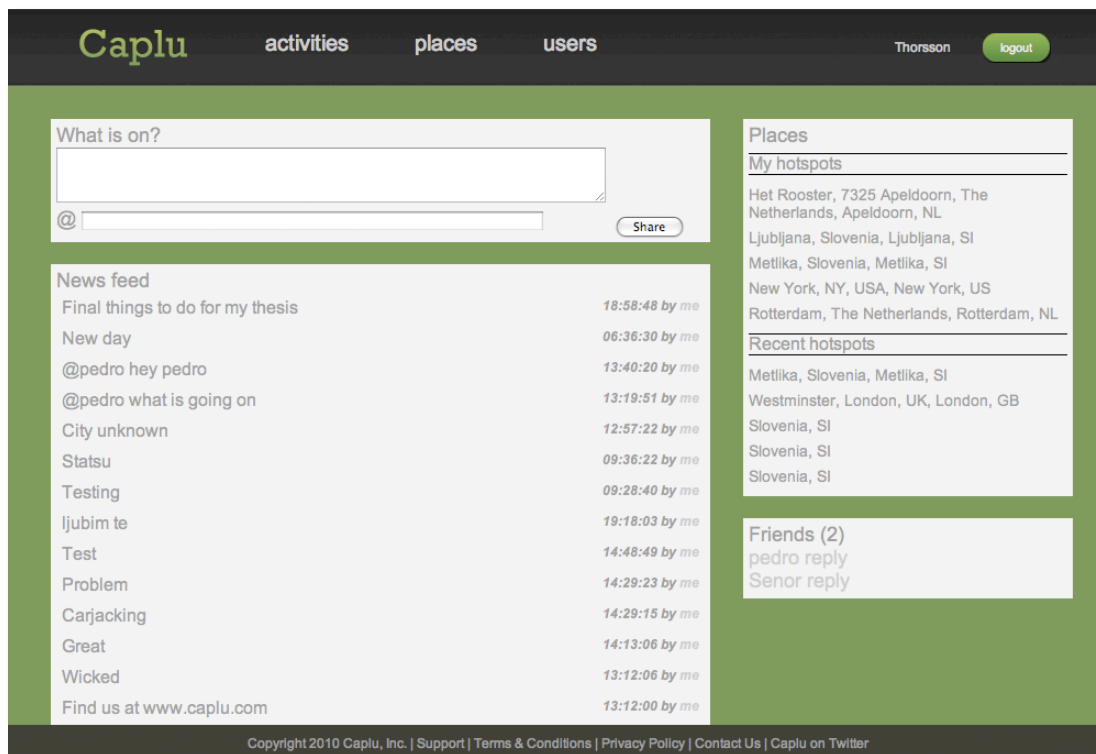
Prvi korak omogoča uporabniku vpis svojih osebnih podatkov ali prijavo z že obstoječim računom na Facebooku ali Twitterju. Drugi korak postopka omogoča uporabniku dodajanje zunanjih storitev, s katerimi hoče, da se aktivnosti sinhronizirajo. Tretji korak je namenjen dodajanju novih osebnih oddajnih točk. Zadnji korak v postopku omogoča dodajanje uporabnikov prijateljev, ki že uporabljajo storitev.



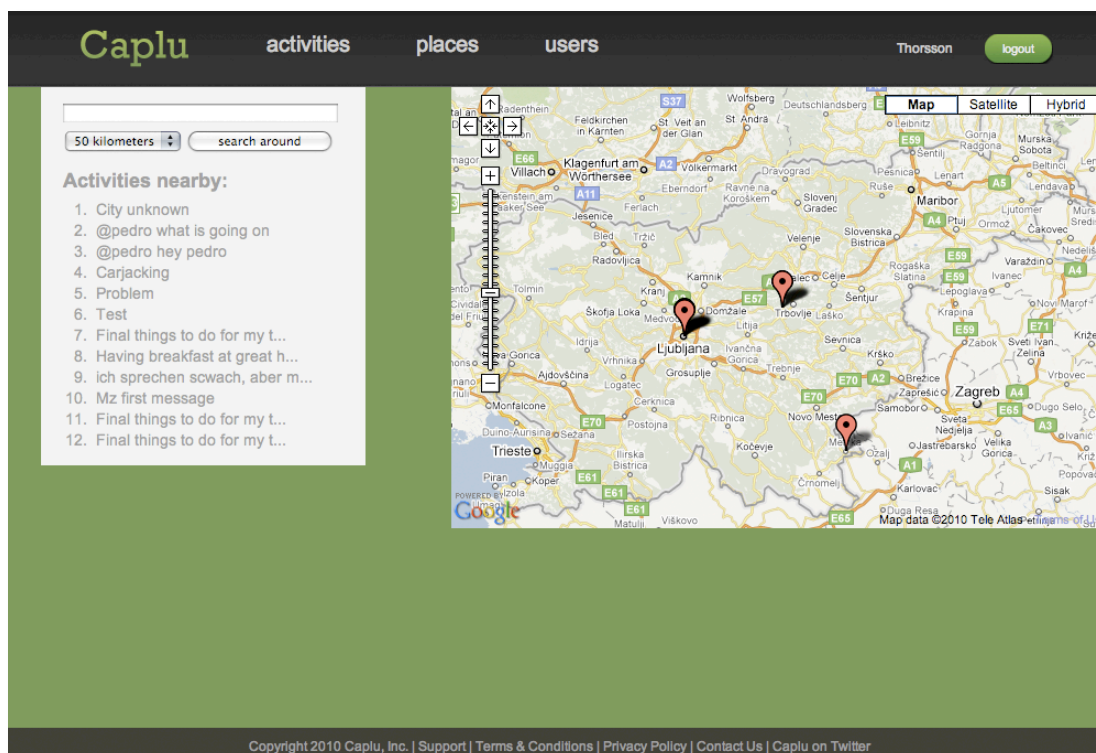
Slika 3.5: Registracija novega uporabnika je definirana s štirimi koraki. Začne se z vpisom osebnih podatkov, nadaljuje s povezovanjem ostalih storitev, nato možnost dodajanja osebnih oddajnih točk in na koncu pošiljanje povabil prijateljem.



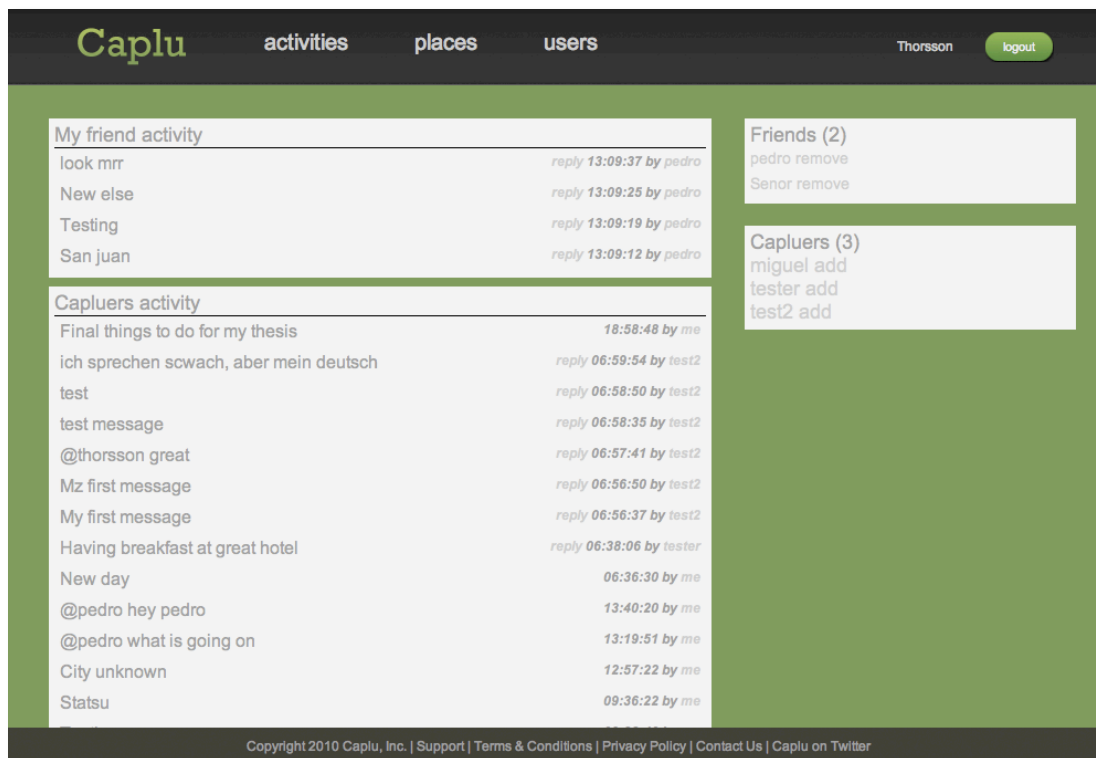
Slika 3.6: Osnutek vmesnika za pozdravno stran. Na njej vidimo definirane najbolj pomembne funkcije: možnost vnosa aktivnosti, pregled zadnjih aktualnih aktivnosti, osebne oddajne točke in seznam prijateljev.



Slika 3.7: Končni izgled spletne aplikacije je nekoliko drugačen od konceptualne zasnove. Pri tem je vključena možnost ročnega vpisa trenutne lokacije.



Slika 3.8: Na sliki je prikazan pogled za iskanje aktivnosti po mapi. Za iskanje lahko določimo radij iskanja. Spodaj se izpišejo najbolj aktualne aktivnosti razporejene po oddaljenosti. Na desni strani se nahaja prikaz rezultatov na mapi.



Slika 3.9: Prikaz strani za upravljanje s prijatelji. Na levi strani se nahajata dva prikaza aktivnosti. Prvi je namenjen prikazu aktivnosti prijateljev, drugi pa prikazuje javne objave uporabnikov.

```
<?xml version="1.0" encoding="UTF-8"?>
<activities type="array">
  <activity>
    <created-at type="datetime">2010-05-24T21:47:51Z</created-at>
    <id type="integer">12</id>
    <location-id type="integer">31</location-id>
    <message>Catching on sun rays</message>
    <updated-at type="datetime">2010-05-24T21:47:51Z</updated-at>
    <user-id type="integer">1</user-id>
  </activity>
  <activity>
    <created-at type="datetime">2010-05-24T21:47:57Z</created-at>
    <id type="integer">13</id>
    <location-id type="integer">32</location-id>
    <message>Good place to eat</message>
    <updated-at type="datetime">2010-05-24T21:47:57Z</updated-at>
    <user-id type="integer">1</user-id>
  </activity>
  <activity>
    <created-at type="datetime">2010-05-24T21:48:02Z</created-at>
    <id type="integer">14</id>
    <location-id type="integer">33</location-id>
    <message>Something special</message>
    <updated-at type="datetime">2010-05-24T21:48:02Z</updated-at>
    <user-id type="integer">1</user-id>
  </activity>
  <activity>
    <created-at type="datetime">2010-05-24T21:56:14Z</created-at>
    <id type="integer">15</id>
    <location-id type="integer">34</location-id>
    <message>Great coffee place</message>
    <updated-at type="datetime">2010-05-24T21:56:14Z</updated-at>
    <user-id type="integer">1</user-id>
  </activity>
</activities>
```

Slika 3.10: Vzorec izpisa aktivnost v XML formatu, kadar kličemo podatke z javno knjižnico.

3.2 Mobilna aplikacija

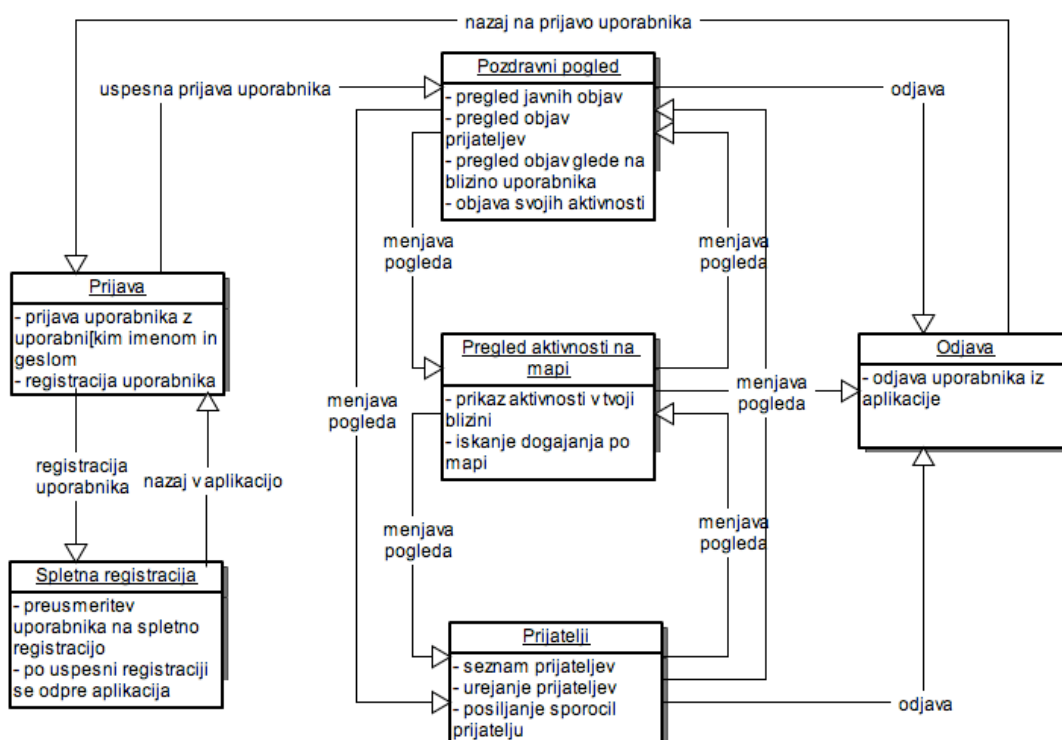
3.2.1 Načrt

Pri načrtovanju za mobilne naprave je potrebno narediti drugačen pristop kot za spletno aplikacijo. Mobilne naprave imajo svoje prednosti in omejitve, ki pa jih je potrebno upoštevati pri razvoju. Za razvoj je bil izbran operacijski sistem Android. Prednosti mobilnih naprav z Androidom so: prisotnost GPS čipa, zaslon na dotik in brezžična povezava (Wi-Fi). Slabosti so sledeče: počasnejši procesor, baterije ne omogočajo dolgega delovanja, omejena količina spomina za shranjevanje podatkov in majhen zaslon. Na podlagi zmogljivosti naprave smo določili funkcionalnost aplikacije.

Mobilna naprava je namenjena osebni rabi. Zaradi tega so za mobilno aplikacijo na voljo samo funkcije, ki so povezane z registriranim uporabnikom.

Funkcije na mobilni aplikaciji:

- Prijava registrirane osebe, če je nov uporabnik se odpre registracija v mobilnem brskalniku.
- Pozdravni pogled, kjer je na voljo možnost pregleda javnih objav ali samo objav prijateljev.
- Pregled aktivnosti na mapi glede na trenutno lokacijo, ki se določi z GPS čipom.
- Možnost dodajanja/brisanja prijateljev iz javnih objav.

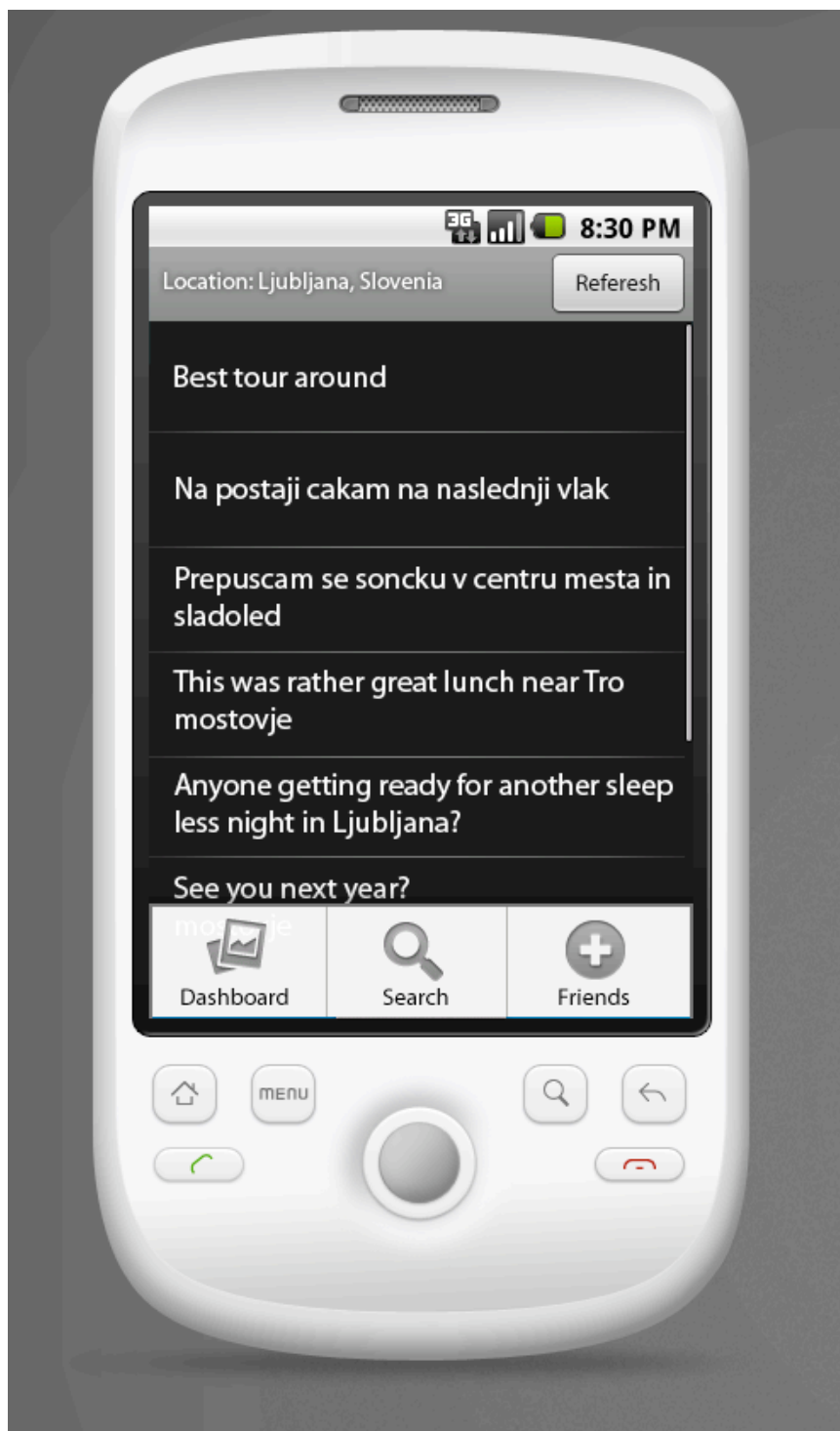


create and share your own diagrams at gliffy.com

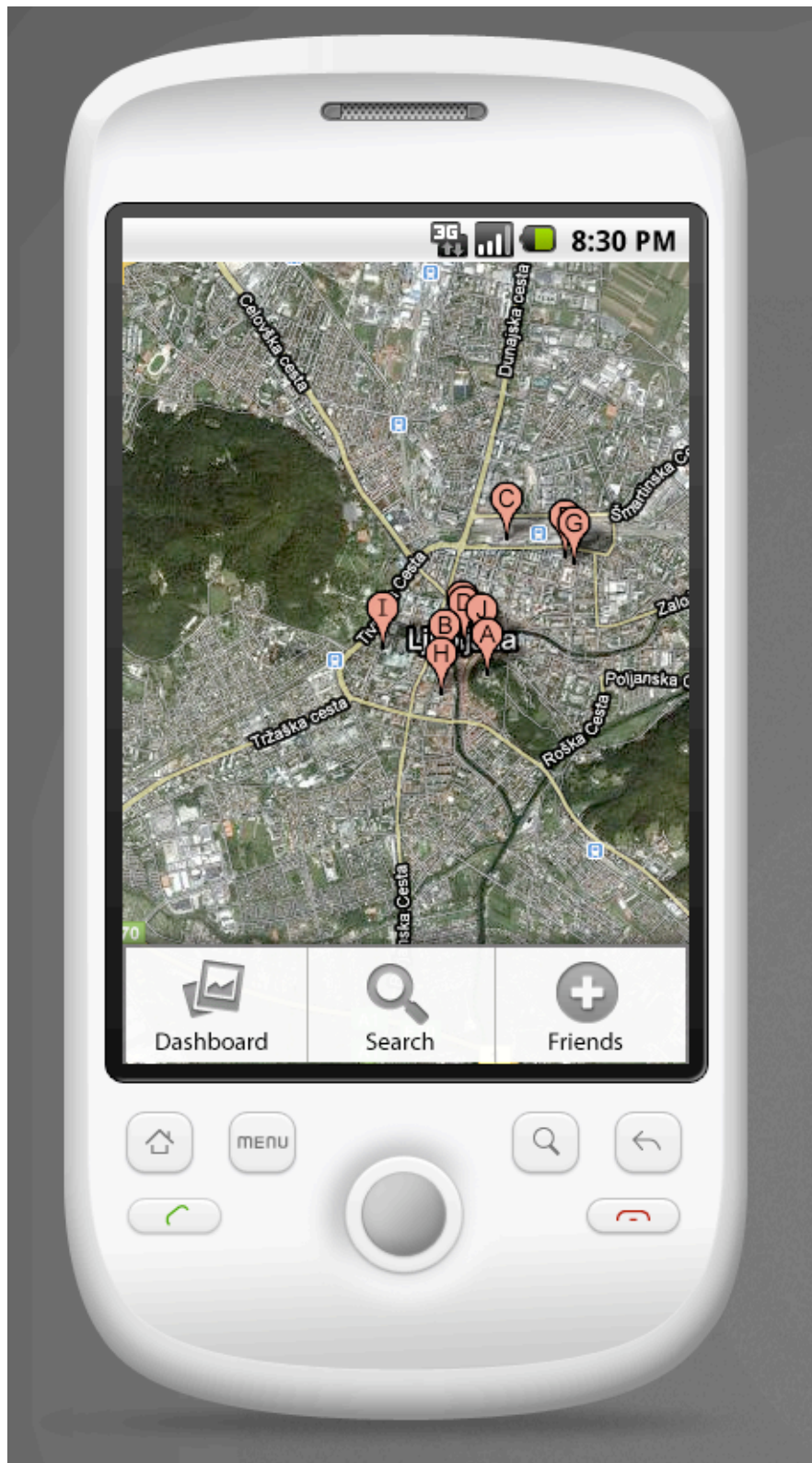


Slika 3.11: Načrt stanj sistema v mobilni aplikaciji. Ko se uporabnik uspešno prijavi v sistem, ima na voljo tri poglede, ki funkcionalno dopolnjujejo svoje delovanje.

3.2.2 Načrtovanje uporabniškega vmesnika



Slika 3.12 Dodaj dashboard sliko



Slika 3.13 Map view slika



Slika 3.14 User list slika

3.3 Podatkovna baza

Za pravilno delovanje rešitve je potrebno natančno definirati podatkovno bazo. Dobro zastavljen model podatkovne baze olajša razvoj rešitve, kajti v nasprotnem primeru vsako nadaljnje spreminjanje strukture podaljša čas razvoja.

Opisane so sledeče tabele:

- Tabela User vsebuje informacije o posameznih registriranih uporabnikih. V njo se beležijo informacije o uporabi registriranega uporabnika.

| Atribut | Tip atributa | Namen |
|-----------------------------|-------------------|---|
| user_id | Int, Primary key | Identifikator uporabnika |
| Email | Varchar | Elektronska pošta |
| Created_at | Datetime | Datum registracije računa |
| Updated_at | Datetime | Datum zadnje posodobitve |
| Crypted_password | Varchar | Šifrirano geslo |
| Password_salt | Varchar | »Salt« za šifriranje |
| Persistence_token | Varchar, NOT NULL | Uporablja se za uporabniške seje |
| Single_accesss_token | Varchar | Ključ za API dostop posameznega uporabnika |
| Perishable_token | Varchar | Identifikator za ponovno nastavljanje gesla |
| Login_count | Integer, NOT NULL | Število uspešnih vpisov |
| Failed_login_count | Integer, NOT NULL | Število neuspešnih vpisov |
| Name | Varchar | Polno ime uporabnika |
| Facebook_uid | Varchar | Identifikator Facebook računa |
| Facebook_session_key | Varchar | Seja za Facebook |
| Oauth_token | Varchar | Identifikator za Oauth prijavo |
| Oauth_secret | Varchar | Skrivnost za Oauth prijavo |
| Twitter_uid | Varchar | Identifikator Twitter računa |
| Avatar_url | Varchar | Url naslov za profilno sliko |

Tabela 3.1: Tabela uporabnika.

- Tabela Item vsebuje podatke o multimedijski vsebini. Pri tem je možno opisati različne formate podatkov. Poleg tega ima tudi podatke, komu pripadajo in kje se delijo.

| Atribut | Tip atributa | Namen |
|--------------------|--------------------------------|-----------------------------------|
| user_id | Integer, foreign key | Identifikator lastnika objekta |
| item_id | Integer, primary key, NOT NULL | Identifikar multimedijske vsebine |
| Name | Varchar | Naslov vsebine |
| Description | Varchar | Dodatni opis vsebine |

Tabela 3.2: Tabela vsebuje informacijo o multimedijski vsebini.

- Tabela Location vsebuje seznam lokacij, ki so jih uporabniki uporabili pri oddajanju vsebine. Zajete so lokacije ustvarjene ob deljenju vsebine in osebne oddajne točke.

| Atribut | Tip atributa | Namen |
|--------------------|----------------------|--|
| Location_id | Integer, primary key | Identifikator lokacije |
| Name | Varchar | Poljubno ime za lokacijo »Moja hiša« |
| Description | Varchar | Opis za lokacijo |
| Lon | Double | Longituda ali ekliptična dolžina |
| Lat | Double | Latituda ali ekliptična širina |
| Street | Varchar | Naslov ulice |
| City | Varchar | Mesto |
| Country | Varchar | Država |
| Post_code | Varchar | Poštna številka |
| Created_at | Datetime | Datum, ko je bila ustvarjena lokacija |
| State | Varchar | Zvezna država ali regija |
| Updated_at | Datetime | Datum spremembe podatkov o lokaciji |
| Address | Varchar | Celoten naslov v enem polju, običajno se nahaja niz, ki je bil uporabljen pri ustvarjanju lokacije |

Tabela 3.3: Tabela lokacije.

- Tabela Hotspots je namenjena za shranjevanju osebnih oddajnih točk posameznega uporabnika.

| Atribut | Tip atributa | Namen |
|--------------------|--------------------------------|---|
| Location_id | Integer, foreign key, NOT NULL | Identifikator lokacije, ki definira oddajno točko |
| User_id | Integer, foreign key, NOT NULL | Identifikator lastnika oddajne točke |
| Name | Varchar | Ime oddajne točke |
| Description | Varchar | Opis oddajne točke |
| Created_at | Datetime | Datum ustvarjanja točke |
| Updated_at | Datetime | Datum spremembe podatkov |

Tabela 3.4: Tabela oddajne točke.

- Tabela Activity vsebuje zapise oddanih aktivnosti. Povezana je z avtorjem sporočila in lokacijo oddajanja. Za težko določljive lokacije se uporabi natančnost na nivoju države.

| Atribut | Tip atributa | Namen |
|--------------------|--------------------------------|---|
| User_id | Integer, foreign key, NOT NULL | Identifikator avtorja sporočila |
| Location_id | Integer, foreign key, NOT NULL | Identifikator lokacije, kjer je bilo sporočilo oddano |
| Created_at | Datetime | Datum oddajanja sporočila |
| Updated_at | Datetime | Datum spremembe sporočila |
| Message | Varchar | Sporočilo o aktivnosti |

Tabela 3.5: Tabela aktivnosti.

- Tabela Bag predstavlja uporabnikov nahrbtnik, kjer shranjuje svojo multimedijsko vsebino. Vsak uporabnik ima svoj nahrbtnik s svojo vsebino, ki pa jo lahko deli z ostalimi na določeni lokaciji.

| Atribut | Tip atributa | Namen |
|-------------------|--------------------------------|-----------------------------------|
| User_id | Integer, foreign key, NOT NULL | Identifikator lastnika nahrbtnika |
| Quantity | Integer | Količina vsebine |
| Created_at | Datetime | Datum odprtja nahrbtnika |
| Updated_at | Datetime | Datum zadnje spremembe nahrbtnika |

Tabela 3.6: Tabela nahrbtnika.

- Tabela Chest ima podobno vlogo kot je tabela Bag, ampak je namenjena shranjevanju vsebine za posamezno točko v prostoru. Vsaka lokacijo ima svojo lastno skrinjo.

| Atribut | Tip atributa | Namen |
|--------------------|-----------------------------------|---|
| Location_id | Integer, foreign key, NOT NULL | Identifikator lokacije, ki je lastnik skrinje |
| Quantity | Integer | Količna vsebine |
| Created_at | Datetime | Datum odprtja skrinje |
| Updated_at | Datetime | Datum zadnje spremembe skrinje |

Tabela 3.7: Tabela skrinje.

- Zgornje naštetе tabele predstavljajo najbolj pomembne tabele za delovanje. Poleg teh so tudi sledeče: tabela Friend za določanje povezav med uporabniki, tabela Invite za pošiljanje povabil novim uporabnikom, tabela Profile vsebuje dodatne informacije o uporabniku, tabela ToDo vsebuje opise aktivnosti, ki jih je možno početi na izbrani lokaciji.

4 Uporaba storitve

To poglavje diplomske naloge je namenjeno predstavitvi primerov uporabe storitve. Razdeljeno je na tri dela: primer uporabe spletne aplikacije, primer uporabe mobilne aplikacije in primer uporabe rešitve.

4.1 Spletna aplikacija

Izvedba spletne aplikacije ima na voljo veliko možnosti. V nadaljevanju sta opisana obiskovalec in registrirani uporabnik storitve. Obe imata opis zmožnosti in omejitev pri uporabi.

Obiskovalec

Ko obiskovalec prvič odpre spletno aplikacijo ima na voljo pregled informacij, ki mu skušajo predstaviti storitev in njene možnosti uporabe v različnih situacijah. Obiskovalec nima ustvarjenega osebnega računa, zato ne more uporabljati naprednih funkcij. Ker je obiskovalec neznana oseba, se zaradi varovanja podatkov onemogoči vpogled v osebne podatke registriranih uporabnikov, zato lahko pregleduje samo zadnje objavljene aktivnosti vseh uporabnikov, aktualne lokacije objavljenih aktivnosti in pa seznam uporabnikov, ki največ uporabljajo storitev. Za uporabo drugih funkcij se mora obiskovalec registrirati in uspešno ustvariti svoj osebni račun. Po uspešno ustvarjenem računu postane registrirani uporabnik.

Registrirani uporabnik

Uspešno ustvarjen račun je predpogoj za prijavo registriranega uporabnika. Po uspešni prijavi uporabnika v sistem se uporabnika preusmeri na pozdravni zaslon. Pozdravni zaslon vsebuje vse najbolj pogosto uporabljene funkcije. Večji del zaslona na levi strani zaseda glavni pregled aktivnosti, kjer se v realnem času izpisujejo nove aktivnosti. Uporabnik lahko odpiše avtorju aktivnosti s klikom na ime avtorja. Na vrhu pregleda aktivnosti se nahaja obrazec za pisanje aktivnosti in ročnim vnosom trenutne lokacije. Na desni strani ima uporabnik možnost pregleda in urejanja svojih osebnih točk. Spodaj se nahaja seznam prijateljev, ki jih lahko hitro kontaktiramo.

Naslednji del je namenjen iskanju informacij v prostoru. Z izbiro »places« povezave se odpre spletna stran, kjer lahko uporabnik poišče aktivnosti v bližini. Z izbiro radija iskanja lahko bolj natančno pogleda dogajanja. Za lažji pregled aktivnosti se mu izrišejo na mapi. S klikom na žebliček se mu odpre oblaček, v katerem je zapisana aktivnost in povezava na osebni profil avtorja.

Na voljo ima tudi pogled namenjen za prijatelje. Tukaj ima na voljo pregled vseh javnih objav in pa zasebni pregled vseh objavljenih sporočil posameznega uporabnika. Ob vsaki aktivnosti je izpisan avtor. S klikom na avtorja se nam odpre osebni profil avtorja.

Pri zadnjem delu pregleda ima uporabnik na voljo svoj osebni profil. Tukaj si lahko nastavi svoje osebne podatke, spremeni geslo. Poleg tega ima na voljo svoj osebni seznam aktivnosti. Na voljo je tudi javni osebni profil, ki ga vidijo tudi drugi uporabniki.

4.2 Mobilna aplikacija

Mobilna naprava je namenjena osebni uporabi in s tem namenom je tudi bila narejena mobilna aplikacija. Za uporabo aplikacije se mora uporabnik uspešno prijaviti v sistem. Po uspešni prijavi se znajde v glavnem oknu, kjer se prikaže pregled aktivnosti, ki pa so zmes aktivnosti uporabnikove lokacije in njegovih prijateljev. Z izbiro iz menija lahko uporabnik posodobi svojo aktivnost, ki pa v ozadju avtomatično pridobi lokacijo.

Naslednje funkcionalno okno predstavlja prikaz aktivnosti na mapi. Ob izbiri pogleda se uporabniku na podlagi pridobljene lokacije prikažejo aktivnosti v bližini. Na ta način se lahko enostavno spozna z okolico in vidi, če se kaj zanimivega dogaja.

Zadnji del je namenjen pregledovanju in urejanju prijateljev. Pri tem ima uporabnik na voljo izpisano zadnjo aktivnost vsakega prijatelja in jo lahko komentira.

4.3 Primer uporabe rešitve

Do sedaj smo samo opisali možnosti, kako lahko posameznik uporablja storitev v svoje namene. Tukaj bosta opisana primera, na katerem področju lahko aktivno uporabljamo dodane napredne zmogljivosti, ki pa trenutno niso dosegljive v komercialnih izdelkih.

V nadaljevanju si bomo ogledali dva možna primera uporabe storitve. To sta: podajanje informacij turističnim popotnikom o turističnih atrakcijah in napredne možnosti v oglaševanju glede na atraktivnost lokacije.

Turistične atrakcije

Prav gotovo je vsem znan občutek, ko se znajdeš v neznani okolici in te zanima, kje v bližini se nahaja kakšna lokalna atrakcija. Z možnostjo ustvarjanja osebnih oddajnih točk lahko poenostavimo možnost predstavitve. Lokalna turistična organizacija preprosto naredi svoj osebni račun, naredi oddajne točke tam, kjer

se nahajajo zanimive lokalne točke za obiskovalce. Z rednim objavljanjem aktivnosti lahko uporabniki dobijo uporabne informacije o dogodkih. Pri tem lahko tudi iščejo po bližini in dobijo informacijo, kje je najboljša izbira lokalnih jedi. Na tak način turistični popotniki dobijo več informacij o turističnih atrakcijah na poti, lahko pustijo sporočilo, da so obiskali ta kraj in z lahkoto poiščejo dodatno ponudbo, ki je na voljo v bližini.

Kontekstualno oglaševanje

Oglaševalske agencije se trudijo z različnimi metodami pridobiti potrošnike. Zaradi zasičenosti klasičnih tržnih kanalov se oglaševalci obračajo k novim metodam oglaševanja. Pri tem se pokaže nova priložnost za osebne oddajne točke. Oglaševalci lahko z odprtim računom ustvarijo svoje oddajne točke, ki se nahajajo na zanimivih lokacijah, kjer se giblje veliko število ljudi. Preko svoje oddajne točke lahko oglašujejo različne izdelke. Največjo priložnost imajo v oddajanju popustnih kuponov. Na določeni popularni lokaciji določijo oddajno točko, kjer ponujajo uporabnikom omejeno število kuponov. Te lahko uporabijo v izbranih poslovalnic in dobijo popust pri nakupu.

5 Zaključek

Diplomsko delo predstavlja storitev za izmenjavo aktivnosti in drugih multimedijskih vsebin v realnem času glede na avtorjevo lokacijo. Pri tem ima uporabnik možnost definirati svoje lastne osebne oddajne točke, kjer se vse njegove akcije prikazujejo ne glede na to, kje se nahaja, želi pa vedno posredovati dogajanje na izbrane lokacije. Rešitev omogoča uporabo spletne in mobilne aplikacije, ki se medsebojno dopolnjujeta v funkcionalnosti. Uporabnik ima možnost spremljanja aktivnosti svojih uporabnikov in komuniciranja z njimi.

V diplomskem delu je bil narejen pregled tehnologij, ki so trenutno na voljo za splet in mobilne naprave. Pred začetkom izdelave načrta so bili definirani kriteriji, katere mora določena tehnologija podpirati, da postane eden izmed kandidatov. Pri tem sta bili izbrani ustrezni tehnologiji, ki izpolnjujeta pogoje. Hkrati gre še za razmeroma nove tehnologije, ki pa se bodo v prihodnosti še bolj izpopolnile. Pri tem so bile opisane tudi podporne programske rešitve za razvojno okolje. Na koncu primerjave je bila narejena primerjava med različnimi IDE urejevalniki kode. Zaradi specifičnosti razvoja različnih tehnologij sta se uporabila dva urejevalnika kode.

Pred izvedbo rešitve je bilo potrebno definirati vse zahteve, ki naj bi jih ta storitev podpirala. Pri tem je bilo potrebno na začetku definirati dober načrt delovanja spletne in mobilne aplikacije ter natančen konceptualni model za podatkovno bazo. Za boljše razumevanje delovanja spletne aplikacije je sledil bolj podroben opis delovanja izbranih delov spletne aplikacije. Za boljšo vizualno predstavitev so bile izdelane konceptualne skice grafičnega uporabniškega vmesnika spletne in mobilne aplikacije, ki se je uporabil pri dejanski implementaciji rešitve.

Testiranje končne spletne aplikacije je bilo opravljeno na treh operacijskih sistemih: Windows 7, Mac OS in Ubuntu. Pri razvoju se je uporabljal Safari kot spletni brskalnik, končni izdelek pa je bil stestiran tudi na Google Chrome in Internet Explorer 8. Za testiranje mobilne rešitve je bila uporabljena naprava z operacijskim sistemom Android verzije 2.1.

V nadaljevanju smo opredelili različne profile uporabnikov, tako spletne kot mobilne aplikacije. Profiliranje uporabnikov smo ločili na dva dela: registrirani uporabnik in pa obiskovalec strani. V naslednjem delu so opisani možni načini izrabljanja funkcionalnosti storitve v komercialne namene. Navedena sta dva primera. Prvi je možnost izkoriščanja osebnih oddajnih točk za promocijo turističnih destinacij in obveščanje o dogodkih. Druga se navezuje na nove možnosti oglaševanja v prostoru. Pri tem gre za možnost izkoriščanja lokacij, kjer se dnevno giblje večje število ljudi. Namenjeno je oglaševanju za večjo prepoznavnost znamke ali možnost ponujanja različnih kuponov s popusti.

Smernic za nadaljnji razvoj je veliko. Pri tem je na voljo možnost dodajanja novih funkcionalnosti na spletni in mobilni aplikaciji. Na spletni aplikaciji je potrebno izpopolniti vmesnik za izmenjavo multimedijske vsebine, tako da lahko na enostaven način iz zunanjih storitev uvozimo vsebino. Največji izziv v prihodnosti je prav gotovo v večjem vzpodbujanju in nagrajevanju uporabnikov. S tem se tudi povečuje število uporabnikov, ki tekmujejo med seboj za vidne rezultate. Trenutno je na voljo samo pregled najbolj aktivnih uporabnikov. Ta sistem bi lahko nadgradili z značkami, ki bi jih uporabnik prejel ob določenem doseženem mejniku. Na primer uporabnik, ki je najbolj aktiven v enem dnevu, dobi posebno značko ali dobi značko za vpis na desetih različnih lokacijah. S tem bi prinesli v storitev element igre in hkrati povečali uporabno vrednost storitve, saj se bi večja količina aktivnosti dnevno izmenjala. Zelo pomemben del je izdelava javno dostopne knjižnice klicev (API). Z API knjižnico bi lahko programerji v prostem času izdelali nove aplikacije za različne platforme in pri tem mogoče prišli do kakšnega novega načina izkoriščanja storitve.

Seznam slik

| | |
|--|----|
| Slika 3.1: Shema delovanja sistema izmenjave podatkov na podlagi lokacije uporabnikov..... | 28 |
| Slika 3.2: Načrt stanj spletne aplikacije za izmenjavo lokacijsko označenih podatkov v realnem času..... | 30 |
| Slika 3.3: Datotečna struktura spletne aplikacije..... | 31 |
| Slika 3.4: Prikazuje nadaljevanje datotečne strukture spletne aplikacije..... | 32 |
| Slika 3.5: Registracija novega uporabnika je definirana z štirimi koraki..... | 33 |
| Slika 3.6: Osnutek vmesnika za pozdravno stran..... | 33 |
| Slika 3.7: Končni izgled spletne aplikacije je nekoliko drugačen od konceptualne zasnove..... | 34 |
| Slika 3.8: Na sliki je prikazano pogled za iskanje aktivnosti na mapi..... | 34 |
| Slika 3.9: Prikaz strani za upravljanje s prijatelji. Na levi strani se nahajata dva prikaza aktivnosti..... | 35 |
| Slika 3.10: Vzorec izpisa aktivnosti v XML formatu, kadar kličemo podatke z javno knjižnico..... | 36 |
| Slika 3.11 Načrt stanj sistema v mobilni aplikaciji..... | 37 |
| Slika 3.12 Dodaj dashboard sliko..... | 38 |
| Slika 3.13 Map view slika..... | 39 |
| Slika 3.14 User list slika..... | 40 |
| Slika 3.15: Konceptualni model podatkovne baze, ki zajema delovanje storitve..... | 41 |

Seznam tabel

| | |
|--|----|
| Tabela 2.1: Primerjava različnih spletnih tehnologij z zahtevami spletne aplikacije..... | 22 |
| Tabela 2.2: Primerjava funkcionalnosti posameznih mobilnih platform..... | 23 |
| Tabela 3.1: Tabela uporabnika..... | 42 |
| Tabela 3.2: Tabela vsebuje informacijo o multimedijski vsebini..... | 43 |
| Tabela 3.3: Tabela lokacije..... | 43 |
| Tabela 3.4: Tabela oddajne točke..... | 44 |
| Tabela 3.5: Tabela aktivnosti..... | 44 |
| Tabela 3.6: Tabela nahrbtnika..... | 43 |
| Tabela 3.7: Tabela skrinje..... | 45 |

Literatura

- [1] (2010) *Social network*. Dostopno na:
http://en.wikipedia.org/wiki/Social_network
- [2] (2010) *Absolute location*. Dostopno na:
http://en.wikipedia.org/wiki/Absolute_location
- [3] (2010) *Ruby on Rails*. Dostopno na:
<http://www.rubyonrails.org>
- [4] (2010) *Mysql*. Dostopno na:
<http://www.mysql.com>
- [5] (2010) *Ide comparison*. Dostopno na:
http://en.wikipedia.org/wiki/Comparison_of_integrated_development_environments
- [6] (2010) *Android*. Dostopno na:
<http://www.android.com/>
- [7] (2010) *Model, view, controller*. Dostopno na:
<http://en.wikipedia.org/wiki/Model-view-controller>
- [8] (2010) *Rails guides*. Dostopno na:
<http://guides.rails.info/>
- [9] (2010) *Railscasts*. Dostopno na:
<http://railscasts.com/>
- [10] (2010) *The Ruby toolbox*. Dostopno na:
<http://ruby-toolbox.com/>
- [11] *Obie Fernandez, The Rails Way, Addison-Wesley Professional, 2007*
- [12] *Hal Fulton, The Ruby Way, Second Edition: Solutions and Techniques in Ruby Programming, Addison-Wesley Professional, 2006*
- [13] *Eric Berry, Rails Pocket Reference, O'Reilly Media, 2008*
- [14] *George Reese, MySQL Pocket Reference: SQL Functions and Utilities, O'Reilly Media, 2007*
- [15] *Mark Murphy, Beginning Android, Apress, 2009*
- [16] *Satya Komatineni, Pro Android, Apress, 2009*

[17] Rick Rogers, John Lombardo, Zigurd Mednieks, G. Blake Meike, Android Application Development: Programming with the Google SDK, O'Reills Media, 2009