

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Uroš Golob

**RAZVOJ MOBILNE APLIKACIJE Z UPORABO OGRODJA SENCHI TOUCH 2**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

Mentor: viš. pred. dr. Igor Rožanc

Ljubljana, 2014



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





Št. naloge: 00520 / 2013  
Datum: 3.9.2013

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

Kandidat: **UROŠ GOLOB**

Naslov: **RAZVOJ MOBILNE APLIKACIJE Z UPORABO OGRODJA SENCHA TOUCH 2**  
**MOBILE APPLICATION DEVELOPMENT USING SENCHA TOUCH 2**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

V diplomski nalogi celovito predstavite ogrodje za razvoj mobilnih aplikacij Sencha Touch 2 ter ga objektivno ocenite z vidika razvijalca aplikacij. V prvem delu naloge predstavite zasnovo, sistematičen razvoj in izgled uporabne mobilne aplikacije razvite z uporabo tega ogrodja. V drugem delu določite nabor kriterijev za celovito oceno uporabnosti ogrodja ter orodje z njihovo uporabo ocenite.

Mentor:

viš. pred. dr. Igor Rožanc



Dekan:

prof. dr. Nikolaj Zimic



# IZJAVA O AVTORSTVU

## DIPLOMSKEGA DELA

Spodaj podpisani      Uroš Golob,

z vpisno številko      63060054

sem avtor diplomskega dela z naslovom:

*Razvoj mobilne aplikacije z uporabo ogrodja Sencha Touch 2*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod vodstvom viš. pred. dr. Igorja Rožanca,
- so elektronske oblike 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

Podpis avtorja:



# Zahvala

Zahvaljujem se viš. pred. dr. Igorju Rožancu za nasvete in dosledno strokovno pomoč.

Zahvaljujem se tudi svoji materi, sestram in ženi Staši, ki so me med študijem podpirali, bodrili in ne nazadnje omogočili, da dosežem želene cilje. Obenem velja zahvala tudi Alojziji Drnovšek, ki je lektorirala besedilo diplomske naloge.



# Kazalo

|       |                                                        |    |
|-------|--------------------------------------------------------|----|
| 1     | Uvod.....                                              | 1  |
| 2     | Strojna in programska oprema .....                     | 3  |
| 2.1   | Strojna oprema.....                                    | 3  |
| 2.2   | Programska oprema .....                                | 4  |
| 2.2.1 | Microsoft Visual Studio 2012 Professional.....         | 4  |
| 2.2.2 | Microsoft SQL Server 2012 Express.....                 | 5  |
| 2.2.3 | SQL Server Management Studio.....                      | 5  |
| 2.2.4 | Sencha CMD .....                                       | 6  |
| 2.2.5 | Aptana Studio 3 .....                                  | 7  |
| 2.2.6 | Android SDK.....                                       | 7  |
| 3     | Predstavitev Sencha Touch 2 .....                      | 9  |
| 3.1   | Kratka zgodovina verzij in podpore.....                | 9  |
| 3.2   | Predstavitev Sencha Touch 2 .....                      | 9  |
| 4     | Izdelava aplikacije.....                               | 13 |
| 4.1   | Ideja aplikacije .....                                 | 13 |
| 4.2   | Priprava podatkov OpenStreetMaps .....                 | 15 |
| 4.3   | Izdelava spletnega servisa.....                        | 17 |
| 4.4   | Priprava logičnega podatkovnega modela .....           | 18 |
| 4.5   | Izdelava aplikacije Sencha Touch 2.....                | 19 |
| 4.5.1 | Priprava projekta.....                                 | 19 |
| 4.5.2 | Izdelava ogledov, modelov in kontrolnih razredov ..... | 20 |
| 5     | Ocena Sencha Touch 2 po kriterijih.....                | 27 |
| 5.1   | Ocenjevanje, kriteriji in uteži kriterijev .....       | 27 |
| 5.2   | Krivulja učenja.....                                   | 28 |
| 5.3   | Dokumentiranost.....                                   | 29 |
| 5.4   | Čas realizacije .....                                  | 30 |
| 5.5   | Prilagodljivost.....                                   | 31 |
| 5.6   | Umestitev .....                                        | 32 |
| 5.7   | Hitrost delovanja.....                                 | 32 |

|     |                          |    |
|-----|--------------------------|----|
| 5.8 | Rezultati.....           | 33 |
| 6   | Zaključek .....          | 35 |
|     | Slike .....              | 37 |
|     | Izvorna koda.....        | 37 |
|     | Tabele.....              | 37 |
|     | Literatura in viri ..... | 39 |

# Seznam uporabljenih kratic in simbolov

API - Application Programming Interface: določa interakcijo različnih programskih komponent med seboj

COM - Component Object Model: predhodnik .NET tehnologije

CSS - Cascading Style Sheets: predloge za določanje izgleda spletne strani

GET - HTTP metoda, ki zahteva podatke iz podanega vira

HTML - Hyper Text Markup Language: označevalni jezik za izdelavo spletnih strani

HTML5 - Hyper Text Markup Language: novejša verzija označevalnega jezika za izdelavo spletnih strani

IDE - Integrated Development Environment: integrirano razvojno okolje

JSON - JavaScript Object Notation: preprost format za prenos podatkov, ki temelji na podmnožici programskega jezika JavaScript

MVC - Model View Controller: tip arhitekture programske opreme

ORM - Object-Relational Mapping: tehnika za pretvorbo podatkov med nezdružljivima sistemoma v objektni sistem

OSM - OpenStreetMaps: prosto dostopni zemljevidi

POST - HTTP metoda, ki predloži podatke za obdelavo podanemu viru

REST - Representational state transfer: arhitekturni stil za izmenjavo podatkov

SQL - Structured Query Language: strukturirani podatkovni jezik za delo s podatkovnimi bazami

SSMS - SQL Server Management Studio: program za delo z podatkovnimi bazami Mikrosft SQL

URI - Uniform Resource Identifier: enoten in enoličen niz znakov, s katerim se opredelita ime in/ali lokacija vira, navadno na internetu

WCF - Windows Communication Foundation: del .NET ogrodja za izdelavo servisno usmerjenih aplikacij

WPF - Windows Presentation Foundation: grafični podsistem za upodabljanje uporabniških vmesnikov, ki temeljijo na operacijskem sistemu Windows.

XML - Extensible Markup Language: označevalni jezik za opisovanje strukturiranih podatkov



# Povzetek

Namen diplomskega dela je predstavitev ogrodja Sencha Touch 2, ki služi kot alternativa pri razvoju mobilnih aplikacij s HTML5 pristopom. V diplomskem delu smo predstavili glavne lastnosti in prednosti ogrodja Sencha Touch 2, nakar smo opisali celoten proces izdelave uporabne aplikacije. Diplomsko delo smo sklenili z oceno ogrodja, ki bo uporabniku služila kot referenca in mu olajšala odločitev o uporabi ogrodja Sencha Touch 2.

Z uporabo ogrodja Sencha Touch 2 smo razvili uporabno aplikacijo, ki uporabniku podaja koristne informacije o okolju, v katerem se nahaja. Za potrebe aplikacije smo razvili spletni servis, podatke, s katerimi smo napolnili podatkovno bazo, pa smo pridobili iz OpenStreetMaps.

Ogrodje Sencha Touch 2 smo ocenili na podlagi sledečih kriterijev: krivulje učenja, dokumentacije, časa izvedbe, prilagodljivosti, možnosti umestitve na različne sisteme in hitrosti delovanja. Vsak kriterij smo ocenili z oceno od 1 do 5, končno oceno vseh kriterijev pa smo predstavili z uteženim povprečjem. Kriterije smo utežili po lastni presoji.

Z oceno ogrodja Sencha Touch 2 smo ugotovili, da je ogrodje dobra alternativa pri izdelavi aplikacije na podlagi HTML5 pristopa. Neizkušen uporabnik bo sicer v uporabo ogrodja moral vložiti kar nekaj časa, vendar bo v zameno dobil prilagodljivo aplikacijo z dobro dokumentacijo in možnostjo njene namestitve na različne platforme. Na starejših mobilnih napravah obstaja možnost, da bo aplikacija delovala slabše, vendar zaradi vse večje zmogljivosti mobilnih naprav to ne bi smelo predstavljati ovire.

## Ključne besede:

Sencha Touch 2, HTML5 pristop, mobilna aplikacija, ocena uporabnosti



# Abstract

The purpose of this thesis is presentation of the Sencha Touch 2 framework, which serves as an alternative in developing mobile applications with HTML5 approach. In this thesis, we presented the main features and advantages of the Sencha Touch 2 framework, then we describe the whole process of building a useful application. The thesis was concluded with an assessment of the framework, which will serve as a reference to user and made his decision easier if he would want to use this framework.

With Sencha Touch 2 framework we have developed a useful application that will provide useful information about the environment in which user is located. For the purpose of application we have developed a web service and fill the database with data from OpenStreetMaps.

The Sencha Touch 2 framework was assessed on the basis of the following criteria: the learning curve, documentation, execution time, flexibility, deployment on different systems, and performance. Each criterion was assessed with a score between 1 and 5, the final evaluation of all criteria were presented with a weighted average. Criteria were weighted according to our own assessment.

With the evaluation of Sencha Touch 2 framework, we found that the framework is a good alternative when you create an application based on HTML5 approach. Inexperienced user will have to invest quite some time to learn this framework but in return he will get a flexible application with good documentation and the possibility of its deployment on various mobile platforms. On older mobile devices, there is a possibility that the application will not work very well, but with the increasing capabilities of mobile devices, this should not be considered as an obstacle.

## Key words:

Sencha Touch 2, HTML5 approach, mobile application, assessment of the usefulness



# 1 Uvod

Pri izdelavi mobilnih aplikacij pride pogosto do odločitve o izbiri mobilne platforme, ki bi jo podprli najprej. Podpreti vsako mobilno platformo v njenem jeziku je zelo potratno. Lahko bi podprli platformo s spletno aplikacijo in tako imenovanim HTML5 pristopom, vendar je ta rešitev velikokrat nepregledna, zmogljivostno spremenljiva, obenem pa se velikokrat odrečemo funkcionalnostim, ki jih ponuja mobilna naprava. V tej diplomski nalogi predstavljamo alternativo, in sicer Sencha Touch 2. Sencha Touch 2 je ogrodje, ki temelji na HTML5, CSS in JavaScript tehnologiji. Je ogrodje, ki močno poenostavi razvoj aplikacij za mobilne platforme in nudi dobro uporabniško izkušnjo.

V tem diplomskem delu bomo predstavili ogrodje Sencha Touch 2 in prikazali celoten proces izdelave aplikacije. Izdelali bomo aplikacijo, ki uporabniku podaja koristne informacije o danem okolju. Na podlagi izdelane aplikacije pa bomo podali oceno ogrodja Sencha Touch 2.

Namen diplomskega dela je:

- predstaviti ogrodje Sencha Touch 2,
- predstaviti celoten proces izdelave aplikacije z uporabo Sencha Touch 2,
- predstaviti ustrezen nabor kriterijev za oceno ogrodja in
- podati oceno na podlagi izbranih kriterijev.

Cilj tega diplomskega dela je:

- bralcu predstaviti uporabnost ogrodja Sencha Touch 2,
- bralcu predstaviti prednosti in slabosti uporabe ogrodja Sencha Touch 2 in
- na podlagi ocene podati uporabno mnenje, s pomočjo katerega se bo bralec lahko odločil za uporabo ogrodja Sencha Touch 2.



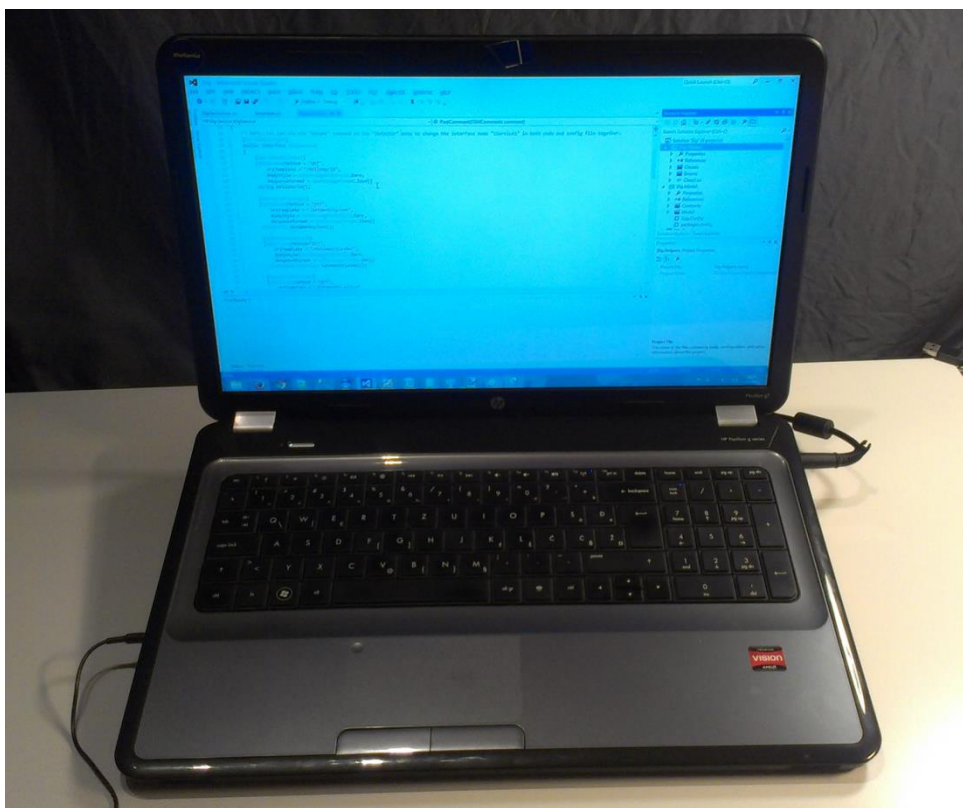
## 2 Strojna in programska oprema

Strojna in programska oprema nista sama sebi namen. Namenjeni sta uporabnikom, da nam olajšata delo in skrajšata pot za doseg zadane cilja. V tem odstavku bomo predstavili komponente, ki smo jih uporabili za doseg tega cilja, se dotaknili omejitev in izrazili mnenje o uporabnosti določenih komponent.

### 2.1 Strojna oprema

Za izdelavo diplomske naloge smo uporabili prenosni računalnik HP Pavilion g7 (na sliki 1), s sledečimi specifikacijami:

- procesor AMD Phenom II N660 Dual Core Processor 3.00 GHz,
- 8 GB DDR3 1333MHz pomnilnika,
- SATA trdi disk WD 150 GB 7200,
- grafična kartica ATI Mobility Radeon HD 4200 Series,
- operacijski sistem Microsoft Windows 8 x64.



Slika 1: Prenosni računalnik HP Pavilion g7

Za testiranje aplikacije smo uporabili dve mobilni napravi HTC z nameščenim operacijskim sistemom Android:

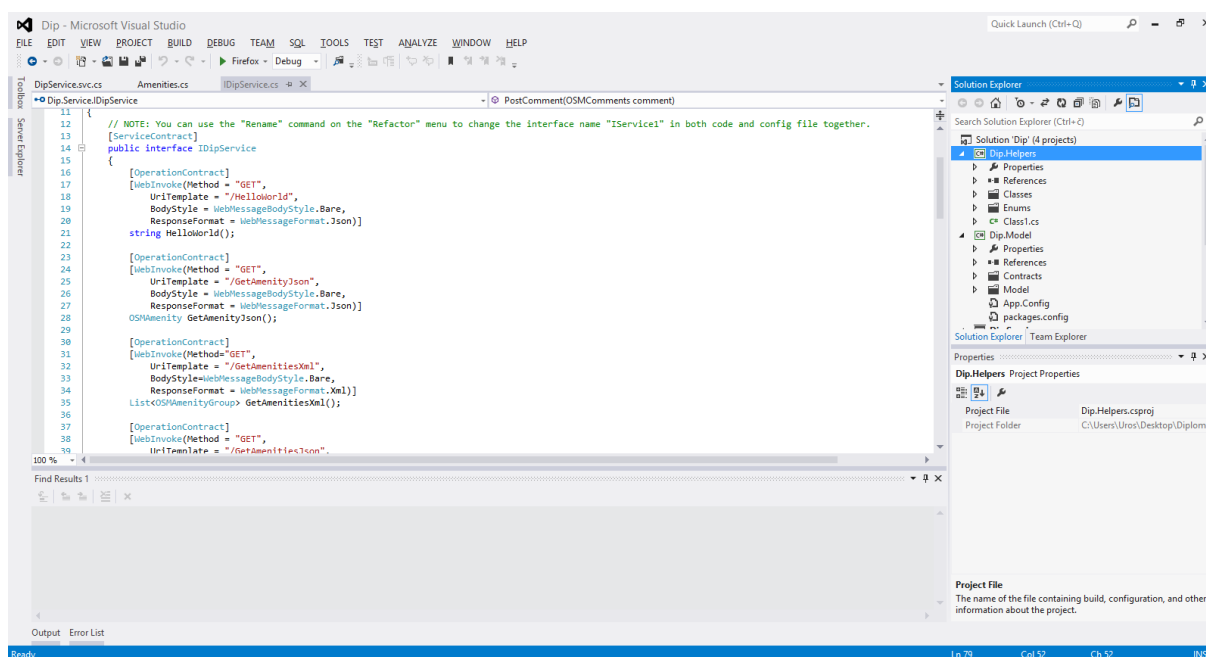
- HTC Wildfire z verzijo Androida 2.2.1 (starejša naprava)
- HTC One V z verzijo Androida 4.2.2 (novejša naprava)

## 2.2 Programska oprema

### 2.2.1 Microsoft Visual Studio 2012 Professional

Visual Studio [1] je izredno močno in priljubljeno razvojno okolje, ki uporabniku ponuja najrazličnejše možnosti razvoja aplikacij. Uporablja se za razvoj tako konzolnih aplikacij kot aplikacij z grafičnim vmesnikom, na primer Windows Forms in WPF. Uporablja se za razvoj spletnih aplikacij, za razvoj servisov, za programiranje vtičev Microsoft Office, SharePoint in še mnogo več. Uporabniku omogoča širok izbor že vgrajenih programskih jezikov C, C++, C#, VisualBasic, in F#. Z dodatnimi namestitvami pa podpira tudi Python ter Ruby. Prav tako nudi podporo za XML, HTML in JavaScript ter CSS.

Visual Studio uporabniku omogoči hiter razvoj aplikacij z uporabo čarovnikov, ki pripravijo ogrodje programske kode. Omogoča grafični razvoj aplikacij z metodo "potegni in spusti". Vgrajen ima tudi možnost "intellisense" [2], ki uporabniku na podlagi vtipkanega niza v obliki spustnega seznama ponudi vgrajene funkcije, spremenljivke, objekte ...



Slika 2: Visual Studio 2012 Professional

Visual Studio je orodje, ki ga najbolj poznamo, zato ni bilo potrebno veliko premišljevati o tem, kaj uporabiti, ko je bilo potrebno pripraviti teren za izdelavo Sencha Touch 2 aplikacije. Orodje Visual Studio 2012 in programski jezik C# smo uporabili za izdelavo razčlenjevalnika XML datoteke OpenStreetMaps ter za vnos podatkov v podatkovno bazo. Uporabili smo ga tudi za izdelavo spletnega servisa (po smernicah REST [3]), ki oskrbuje našo aplikacijo s podatki. Visual Studio 2012 je bil deležen prenove, ki je najbolj opazna na grafičnem vmesniku.

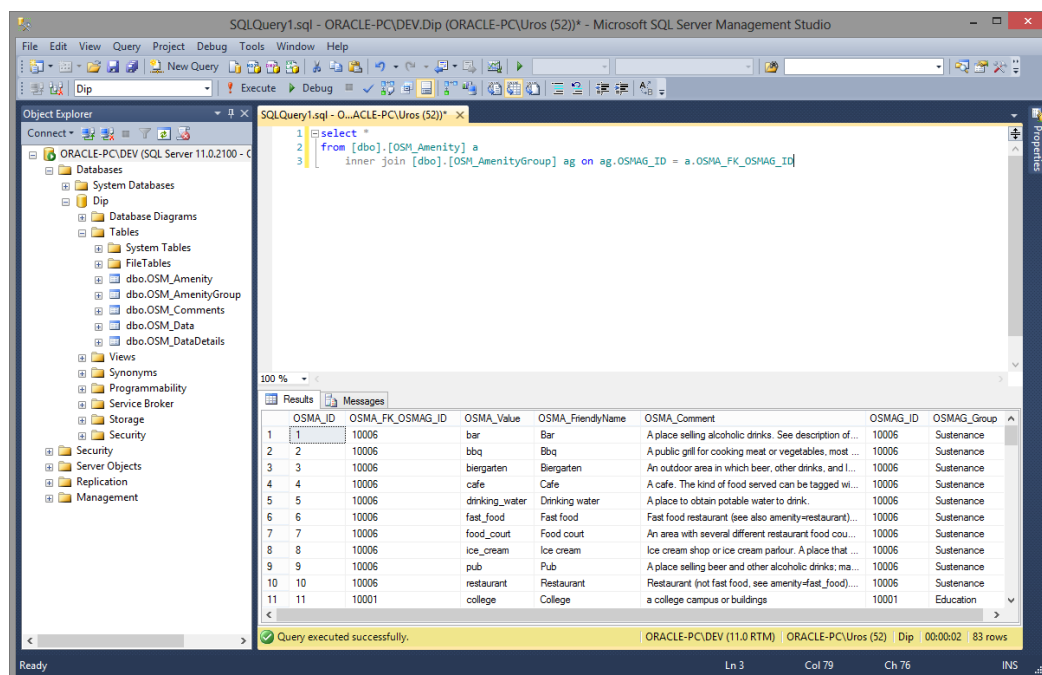
### *2.2.2 Microsoft SQL Server 2012 Express*

SQL Server Express [4] je brezplačna verzija Microsoft SQL strežnika za delo z relacijskimi podatkovnimi bazami. Namenjena je predvsem manj zahtevnim aplikacijam in pa uporabi v domačem okolju. Express verzija podatkovne baze ima številne omejitve [5]:

- maksimalno 10 GB podatkov na podatkovno bazo,
- 1 procesor oz. 4 jedra in
- 1 GB delavnega pomnilnika.

### *2.2.3 SQL Server Management Studio*

SQL Server Management Studio (v nadaljevanju SSMS) (slika 3) je programsko orodje, ki se uporablja za konfiguriranje, upravljanje in administracijo z vsemi komponentami, ki jih ponuja Microsoft SQL Server. SSMS vsebuje orodje za delo s skripti ter različna orodja v obliki grafičnih vmesnikov. Osrednja funkcija SSMS je raziskovalec objektov (ang. Object Explorer), ki uporabniku omogoča, da brska, izbira in izvaja akcije nad vsemi objekti znotraj SSMS [6]. Od verzije 11 (SQL Server 2012) je bil SSMS prepisan v WPF [7] in je vizualno podoben Visual Studiu 2010.



Slika 3: SQL Server Management Studio

## 2.2.4 Sencha CMD

Sencha CMD [8] je konzolna aplikacija, podprta na različnih platformah, ki zagotavlja velik nabor avtomatiziranih opravil med življenjskim ciklom aplikacije. Od preproste izdelave novega projekta do umestitve aplikacije v produkcijsko okolje. Nudi zbirko močnih funkcij, ki delujejo celostno in v povezavi s Sencha Ext JS in Sencha Touch 2. Sencha CMD omogoča naslednji nabor zmogljivosti [9].

- **Orodja za generiranje programske kode:** nudi orodja za generiranje celotnih aplikacije in za razširitev aplikacij z MVC komponentami.
- **JavaScript prevajalnik:** vsebuje prevajalnik za izdelavo paketov z minimalnim odtisom. V prihodnje bo prevajalnik nudil optimizacijo visoko nivojskim objektom in posledično skrajšal čas zagona aplikacije.
- **Spletni strežnik:** nudi lahkoten spletni strežnik, ki servira podatke iz lokalnega gostitelja.
- **Umestitveni paketi:** omogoča izdelavo paketov, ki jih je mogoče umestiti na mobilne platforme in objaviti na spletu.
- **Sistem za upravljanje:** nudi porazdeljen sistem za upravljanje s paketi, ki so jih ustvarili drugi.
- **Skript za izgradnjo:** izdelava skript za izgradnjo, ki ga uporabnik lahko prilagodi svojim potrebam.
- **Orodja za usklajevanje:** vsebuje močno orodje za selekcijo kode in usklajevanje, kaj bo vključeno v končnem paketu.
- **Upravljanje delovnega prostora:** nudi podporo pri delitvi ogrodij, paketov in kode po meri med aplikacijami.

- **Zajem slik:** pretvori CSS3 lastnosti za podporo starim brskalnikom.
- **Prilagodljiv sistem za konfiguracijo:** omogoča določanje privzetih nastavitev čez vse delovne prostore.
- **Logiranje:** nudi robustno logiranje za razumevanje delovanja komponent in za namen razhroščevanja.
- **Dodatna programska oprema:** vsebuje kompatibilno verzijo Compass-a, Sass-a in Apache Ant-a.
- **Ukane za generiranje kode:** so lahko določene za eno stran ali v skupni rabi za celotni delovni prostor, na primer za preverjanje smernic ali kodiranja, ko so ustvarjeni novi modeli.

### 2.2.5 Aptana Studio 3

Aptana Studio 3 [10] je odprtokodno integrirano razvojno okolje (v nadaljevanju IRO) za izdelavo AJAX spletnih aplikacij. Aptana Studio je zasnovan na Eclipse platformi in podpira JavaScript, HTML, DOM, in CSS obvestila o napakah, dokončevanje kode in integrirano dokumentacijo. Z namestitvijo dodatnih vtičev podpira tudi PHP, Ruby on Rails, Python, Perl, Adobe AIR ... Aptana Studio je na voljo kot samostojna aplikacija za Windows, Mac OS X in Linux operacijske sisteme ali kot vtič za Eclipse. Aptana studio 3 je na voljo izključno pod GPL licenco, medtem ko je bila Aptana Studio 2 na voljo pod dvojno licenco [11].

### 2.2.6 Android SDK

Android SDK [12] zagotavlja razvijalcu orodja za razvoj programske opreme, ki so potrebna za razvoj, testiranje, izgradnjo in umestitev aplikacije na Android naprave. Vsebuje orodja za razhroščevanje, knjižnice, posnemovalnik (ang. emulator), dokumentacijo, primere kode in vodiče. AndroidSDK je podprt na platformah Linux, Mac OS X in Windows. Uradno podprto integrirano razvojno okolje je Eclipse, ki uporablja Android razvojna orodja, na voljo pa je tudi kot vtič za NetBeans [13].

V tem odstavku smo na kratko opisali programsko in strojno opremo, ki smo jo uporabljali za izdelavo diplomskega dela. V naslednjem poglavju pa bomo podrobneje predstavili ogrodje Sencha Touch 2.



## 3 Predstavitev Sencha Touch 2

V tem poglavju se bomo najprej na kratko dotaknili zgodovine verzij in podpore, nato pa bomo predstavili pomembnejše lastnosti in komponente ogrodja. Na koncu bomo še predstavili, za izdelavo kakšnih aplikacij je primerna Sencha Touch 2.

### 3.1 Kratka zgodovina verzij in podpore

Sencha Touch je produkt Senche, ki je bil ustvarjen, ko so združili tri popularne JavaScript knjižnice; Ext JS [14], jQTouch [15] in Raphaël [16]. Prva objava Senche Touch (verzija 0.9 beta) je bila na voljo 17. 7. 2010. Ta verzija je bila podprta na napravah, ki so imele nameščen Android ali iOS. Prvo stabilno verzijo so objavili novembra 2010, z verzijo 1.1.0 pa so dodali podporo za BlackBerry OS 6. Trenutno je na voljo verzija 2.3.1., ki je bila objavljena 8. 11. 2013 in podpira naslednje brskalnike in platforme [17]:

- Android brskalnik,
- Google Chrome brskalnik za Android,
- BlackBerry 10,
- Bada Mobile brskalnik,
- Kindle Fire brskalnik
- Windows Phone 8
- Windows 8 Internet Explorer 10 in
- Safari.

### 3.2 Predstavitev Sencha Touch 2

Sencha Touch 2 [18] je ogrodje za izdelavo mobilnih aplikacij. Ogrodje je posebej prilagojeno za uporabo na napravah z zaslonom na dotik ter uporablja MVC obliko arhitekturnega tipa. Popolnoma sloni na spletnih standardih HTML5, JavaScript in CSS3. Sencha Touch omogoča uporabniku hiter razvoj mobilne aplikacije in njeno umestitev na mobilne platforme Android, iOS, BlackBerry, Windows Phone in še mnoge druge.

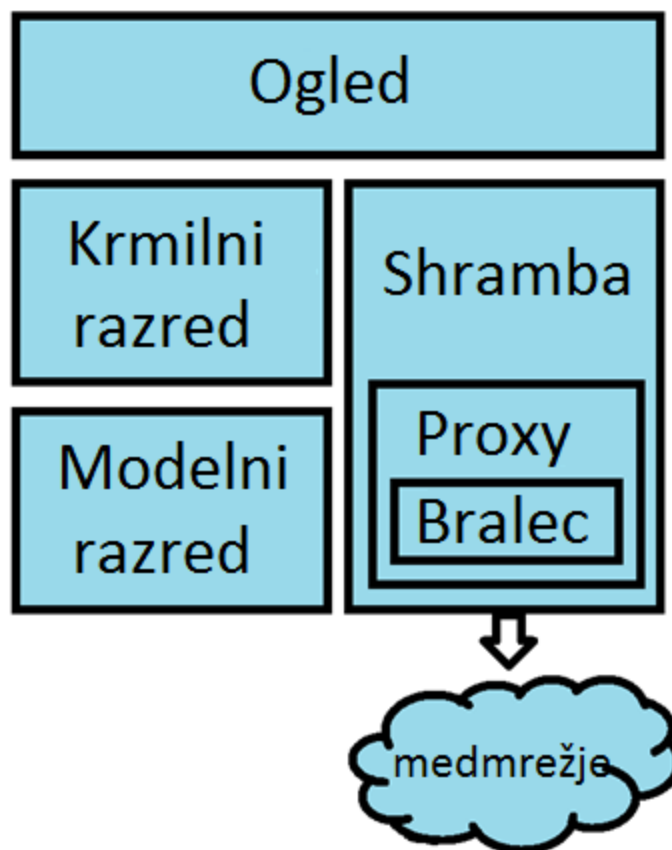
Sencha Touch 2 [19] je veliko več kot samo funkcionalno ogrodje za izgradnjo uporabnih in produktivnih aplikacij. Je dejansko popolna platforma za spletne aplikacije s sledečimi lastnostmi:

- vsebuje veliko UI gradnikov, navdihnjenih na podlagi iOS, tako vizualno kot funkcijsko,

- ima hiter pogon za upodabljanje, ki v celoti temelji na CSS in je lahko strojno pospešen na sodobnih mobilnih napravah,
- ima dobro opredeljeno arhitekturo, ki že od samega začetka temelji na MVC arhitekturnem tipu,
- ima vgrajene priključke za storitve podatkovnega omrežja (REST),
- podpira hranjenje podatkov za delo brez povezave,
- vsebuje napreden mehanizem za nalaganje razredov in uveljavlja smernice za njihovo poimenovanje,
- ima sistem, ki omogoča preprosto izdelavo novega projekta, izgradnjo paketa za mobilne naprave in uvajanje programske rešitve na mobilni napravi,
- ima obsežno dokumentacijo, ki je na voljo tako preko spleta kot za delo brez povezave.

Sencha Touch 2 prihaja v paketu z obsežnim seznamom komponent, objektov, funkcij in programske kode pripravljene za uporabo. Ta paket je primerljiv s tistim, ki ga vidimo pri .NET ali Java in zagotavlja vse osnovne gradnike za hitro izdelavo zahtevnih aplikacij. Poleg tega je ta paket zasnovan po principu MVC arhitekturnega tipa. Ta paket vsebuje sledeče družine razredov [20].

- **Modelni razredi** (ang. model) ponujajo razvijalcu možnost izdelave entitet, ki se ustvarijo, uporabljajo, posodablajo in uničijo tekom življenjskega cikla aplikacije.
- **Komunikacijski razredi** (ang. communication) opisujejo hrambo podatkov in uporabljeno strategijo ob ustvarjanju nove aplikacije. Najpomembnejši komunikacijski razredi so *Store*, *Proxy* in *Reader*, ki ponujajo abstrakcijo in omogočajo aplikaciji, da pridobi podatke preko oddaljene povezave ali lokalne shrambe.
- **Ogledi** (ang. view) se uporabljajo za enostavno izgradnjo uporabniških vmesnikov. So najbolj vidni del aplikacije, uporabljajo napredno strojno pospeševanje in temeljijo izključno na CSS.
- **Krmilni razredi** (ang. controller) so vez med ogledi in modelnimi razredi. Centralizirajo vse interakcije v specializiranih razredih, kjer se nahaja najpomembnejši del aplikacije, logika. Razvijalcu ponuja dobro organizacijo programske kode, ki jo je lažje vzdrževati in nadgrajevati.
- **Uporabnostni razredi** ponujajo uporabniku obsežno količino programske kode pripravljene za uporabo.



Slika 4: MVC struktura Sencha Touch 2

Sencha Touch z verzijo 2.3 podpira IRO Apache Cordova za dostop do strojnih komponent mobilne naprave kot so kamera, kompas, kontakti, GPS senzor, obvestila, povezava in podobno [21].

Apple je v eni prvih verzij tehničnega dokumenta o načrtovanju in smernicah pri razvoju iOS aplikacij izjavil, da se aplikacije, ki jih je mogoče razviti za iOS, v grobem delijo na tri vrste [22].

- Koristne oziroma uporabne aplikacije, ki uporabniku zagotavljajo potrebne informacije, kot so podatki o vremenu, informacije o delnicah, ipd.
- Aplikacije, ki povečujejo uporabnikovo produktivnost. Te so lahko poslovno ali dokumentno usmerjene.
- Večpredstavne aplikacije, kot so igre.

S Sencho Touch 2 je mogoče ustvariti igre ali aplikacije z zahtevnim uporabniškim vmesnikom, vendar je primernejša za koristne oziroma uporabne aplikacije in aplikacije, ki povečujejo uporabnikovo produktivnost.



## 4 Izdelava aplikacije

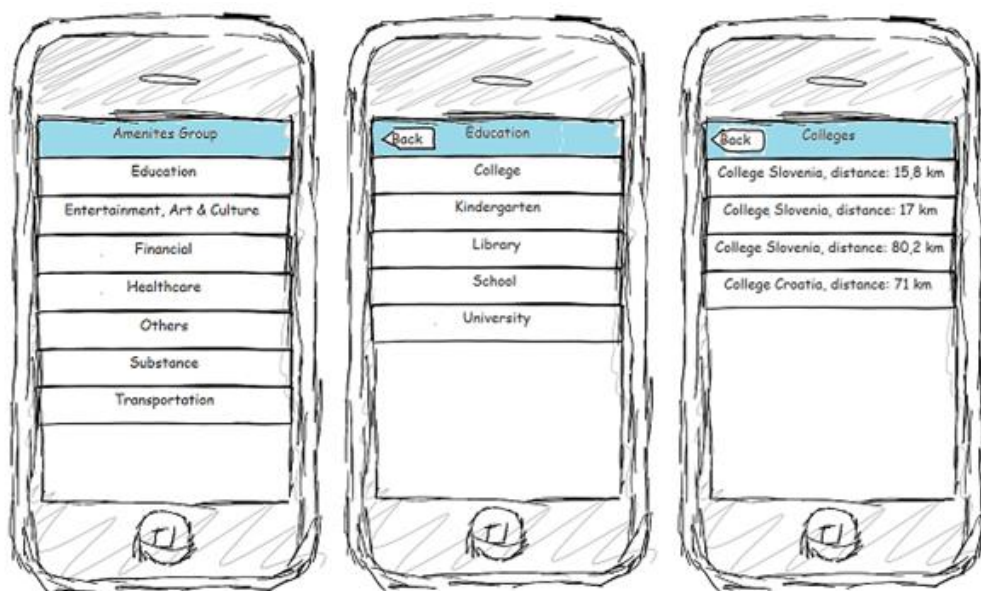
V tem poglavju bomo podrobneje predstavili proces izdelave Sencha Touch aplikacije in vsega, kar je bilo potrebno za izvedbo diplomske naloge.

### 4.1 Ideja aplikacije

Da razumemo celoten koncept, je potrebno najprej poznati izvorno idejo. Sencha Touch 2 je primernejša za izdelavo koristnih/uporabnih aplikacij in aplikacij, ki povečujejo uporabnikovo produktivnost. Zadali smo si cilj, da zmogljivost in lepoto tega ogrodja predstavimo v eni izmed teh dveh oblik. Rodila se je ideja, da izdelamo aplikacijo, ki bo uporabniku podajala koristne informacije (dobrine, ang. amenities) o okolju, v katerem se nahaja na strukturiran način. Najprej bi uporabniku dobrine predstavili kot skupino (z nadpomenko), nato pa vse dobrine znotraj skupine. Sledil bi prikaz vseh lokacij, kjer so izbrane dobrine dosegljive in na koncu vse podrobnosti, ki bi bile na voljo o neki lokaciji.

Naše glavno vodilo pri izdelavi aplikacije je bila njena uporabnost. Zanimalo nas je, koliko truda je potrebno vložiti, da bi uporabnikom ponudili koristne informacije o nečem, da aplikacija ne bi bila zastavljena preveč zahtevno in obenem ne preveč preprosto ter da bi obenem spoznali kar največ o Sencha Touch 2.

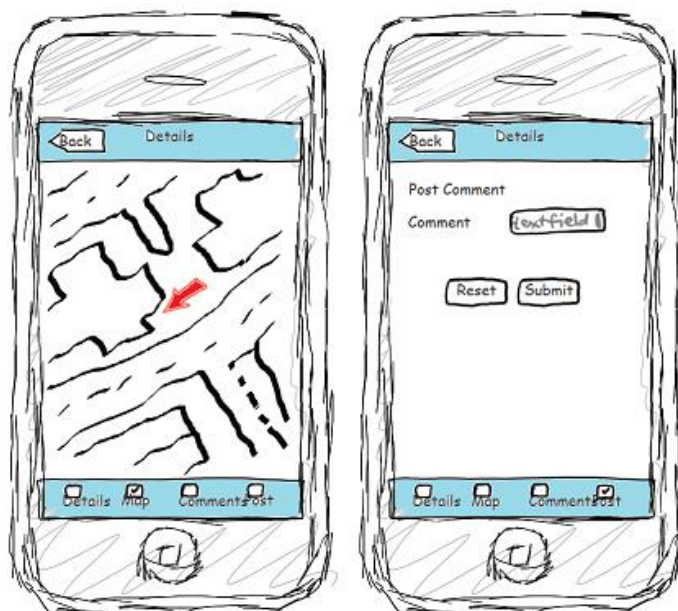
Preden smo se lotili programiranja, smo pripravili podrobnejši koncept aplikacije, določili njen obseg in zahtevnost. Določili smo, da bomo najprej prikazali skupine dobrin. Ker je pri tem pomembno, da imamo jasen prikaz skupin, smo se odločili, da jih prikažemo v seznamu z nazivom dobrine. Klik oziroma dotik (ang. tap) določene skupine nam bo pokazal vse dobrine, ki pripadajo določeni skupini. V prikazu dobrin mora obstajati tudi gumb, ki nas bo vrnil na začetno stran pregleda skupin. Tudi za dobrine smo se odločili, da jih prikažemo pregledno v obliki seznama. Prikazali pa bomo le nazive, saj so sami po sebi dovolj zgovorni. Dotik določene dobrine nam bo v obliki seznama prikazal vse lokacije, ki so od nas oddaljene za eno kotno stopinjo v pozitivno in negativno smer. Poleg lokacije bomo prikazali še državo, v kateri se lokacija nahaja, in njeno oddaljenost od naše lokacije. V pregledu bo na voljo gumb, ki nas bo vrnil nazaj na pregled dobrin (slika 5).



Slika 5: Ogléd skupin dobrin, dobrin znotraj skupine in seznam lokacij dobrin

Dotik na lokacijo nam bo prikazal nov ogled, kjer bodo v obliki zavihkov prikazani vsi podatki o izbrani lokaciji. Na voljo bodo štirje zavihki, in sicer:

- zavihék z vsemi podrobnostmi lokacije,
- zavihék z zemljevidom in točko, kjer se lokacija nahaja (slika 6),
- zavihék z vsemi komentarji, ki so na voljo za določeno lokacijo,
- zavihék s formo, kjer lahko oddamo komentar (slika 6).



Slika 6: Tretji ogled, prikaz zemljevida in forme za oddajo komentarja

Na pregledu podrobnosti izbrane lokacije mora biti na voljo tudi gumb, ki nas vrne na seznam lokacij.

Če pogledamo podrobneje, vidimo, da imamo definiranih večino klicev servisa, ki nam vračajo podatke. Gre za tako imenovane GET klice. Da aplikacija ne bi bila preveč enostavno zasnovana, smo se odločili, da dodamo še POST klic servisa. Tako smo dodali možnost komentiranja določene lokacije.

Ko je bil podrobnejši načrt izdelan, smo se lotili priprave podatkov.

## 4.2 Priprava podatkov OpenStreetMaps

Za izdelavo aplikacije, ki smo jo realizirali z uporabo ogrodja Sencha Touch 2, smo potrebovali ustrezen nabor podatkov. Podatke bi si seveda lahko izmislili, vendar nas je težnja po uporabnosti in pristnosti nagnila k odločitvi, da uporabimo prave podatke. Za vir podatkov smo izbrali prosto dostopne OpenStreetMaps (v nadaljevanju OSM), konkretnije podatke OSM za Slovenijo, Hrvaško in Madžarsko [23].

Podatke smo morali prenesti iz XML datoteke (izvorna koda 1) v podatkovno bazo. Odločili smo se, da bomo iz XML datoteke izluščili vse podatke, ki označujejo dobrine (ang. amenity). Dobrine predstavljajo stvari, ki človeku tako ali drugače koristijo. Dobrine so recimo izobraževalne ustanove, knjižnice, bencinske črpalke, bolnišnice, restavracije, zdravilišča, avtobusne postaje, bankomati ... Dobrine so v XML datoteki označene z lastnostjo "amenity" in tipom udobja (na primer "restaurant").

```
<?xml version='1.0' encoding='UTF-8'?>
<osm version="0.6" generator="osmconvert 0.7P" timestamp="2013-10-30T
  <node id="303584694" lat="62.0107311" lon="-6.7713231" version="5"
    <tag k="amenity" v="restaurant"/>
    <tag k="cuisine" v="chinese"/>
    <tag k="name" v="Nantong"/>
    <tag k="wheelchair" v="yes"/>
  </node>
  <node id="29023725" lat="62.2547643" lon="-6.5781653" version="3"
    <tag k="is_in" v="Klaksvík (Borðoy), Føroyar"/>
    <tag k="name" v="Ánirnar"/>
    <tag k="place" v="village"/>
    <tag k="population" v="8"/>
    <tag k="postal_code" v="726"/>
  </node>
  <node id="29023727" lat="62.255" lon="-6.533" version="2" timesta
```

Izvorna koda 1: Primer strukture XML datoteke OpenStreetMaps

Za razčlenitev podatkov iz XML datoteke (izvorna koda 2) v MS SQL podatkovno bazo smo sprogramirali razčlenjevalnik v jeziku C#. Ker bomo razčlenjevali podatke iz relativno velike XML datoteke, smo uporabili knjižnico XmlReader.

XmlReader nam zagotavlja hitro branje podatkov, ki jih ne hrani v pomnilniku, ampak jih bere preko toka (and. stream) iz XML datoteke. Mogoč je le zaporedni dostop do podatkov [24], tako vračanje na predhodnje zapise ni mogoče.

```
private void ParseOSMXMLFile(){
    decimal lat, lon;
    string tagKey, tagValue;
    bool insertElement = false;

    using (TransactionScope scope = new TransactionScope()){

        DipEntities context = null;
        try {
            context = new DipEntities();
            context.Configuration.AutoDetectChangesEnabled = false;
            context.Configuration.ValidateOnSaveEnabled = false;
            int counter = 1, limit = 100;

            using (XmlReader reader = XmlReader.Create(file)) {
                while (reader.Read()) {
                    if (reader.NodeType == XmlNodeType.Element && (reader.Name.ToLower() == "node")) {
                        XElement nodeElement = (XElement)XNode.ReadFrom(reader);

                        if (nodeElement != null) {
                            lat = decimal.Parse(nodeElement.Attribute("lat").Value, CultureInfo.InvariantCulture);
                            lon = decimal.Parse(nodeElement.Attribute("lon").Value, CultureInfo.InvariantCulture);

                            Node node = new Node(lat, lon);

                            foreach (XElement e in nodeElement.Elements("tag")) {
                                tagKey = e.Attribute("k").Value;
                                tagValue = e.Attribute("v").Value;

                                if (tagKey == "amenity")
                                    insertElement = true;

                                Tag tag = new Tag(tagKey, tagValue);
                                node.AddTag(tag);
                            }

                            if (insertElement) {
                                context = AddToContext(node, context, limit, counter);
                                counter++;
                            }
                            insertElement = false;
                        }
                    }
                }
            }
            context.SaveChanges();
        }
        finally {
            if (context != null)
                context.Dispose();
        }
        scope.Complete();
    }
}
```

Izvorna koda 2: Razčlenjevalnik XML datoteke OpenStreetMaps

Za dostop razčlenjevalnika XML datoteke do podatkovne baze smo uporabili Entity Framework [25]. Entity Framework je ORM, ki omogoča razvijalcu delo z MSSQL

Server podatkovno bazo na .NET platformi. Tako smo brez večjih težav prenesli podatke iz XML datoteke v podatkovno bazo ter napisali vnose (ang. insert) na optimiziran način [26].

V podatkovno bazo smo shranili geografsko dolžino in širino določene lokacije, državo, v kateri se nahaja, ter vse dodatne podatke, ki podrobneje opisujejo lokacijo. Po končanem razčlenjevanju XML datoteke in vnosu podatkov v podatkovno bazo se je v naši podatkovni bazi nabralo 37649 zapisov lokacij iz Slovenije, Hrvaške in Madžarske, ki jih bomo lahko uporabili v naši aplikaciji.

### 4.3 Izdelava spletnega servisa

Potrebovali smo vezni člen med podatki v podatkovni bazi in aplikacijo. Sencha Touch 2 lahko zajame podatke v XML ali JSON zapisu. Zato smo se odločili, da za potrebe aplikacije ustvarimo spletni servis v REST obliki arhitekturnega tipa [27]:

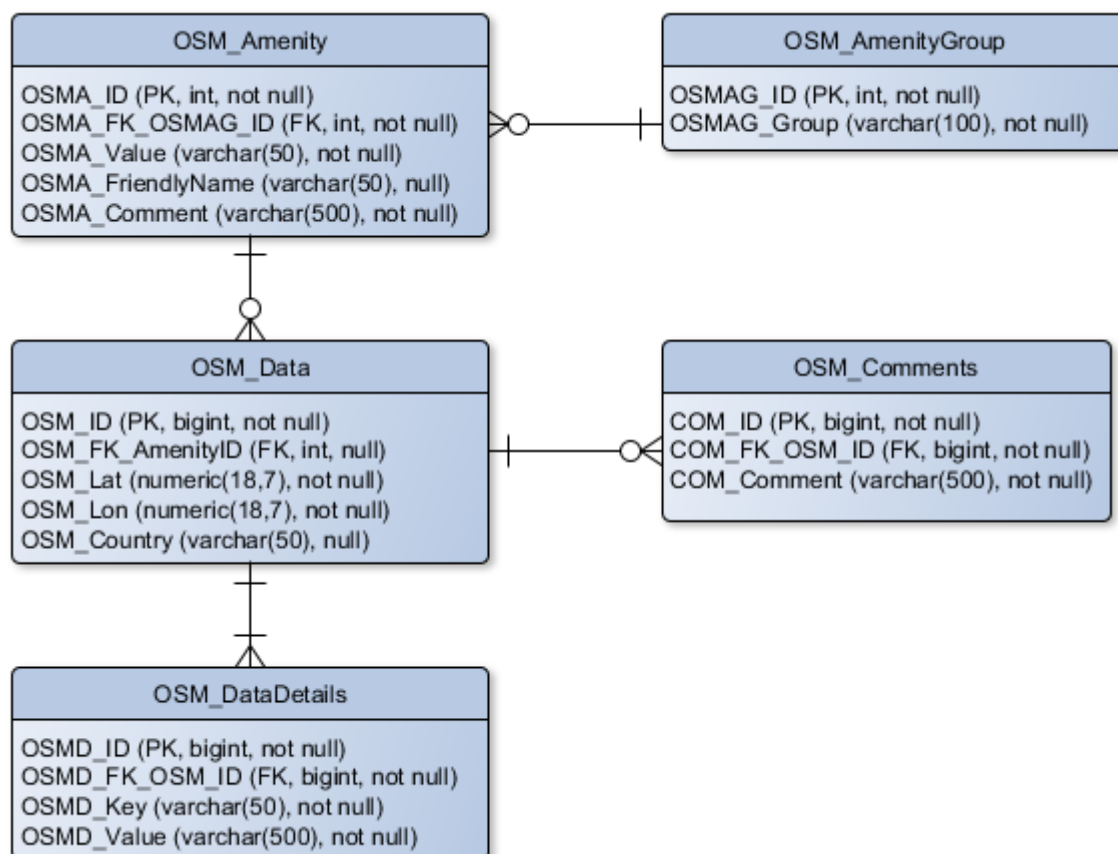
- uporablja obliko URI za spletni API,
- nudi podporo določenim oblikam podatkovnih zapisov (primer XML in JSON),
- podpira HTTP metode (GET, POST, PUT, DELETE),
- prenaša vsebina kot nadbesedilo oziroma hipertekst.

Pri izdelavi servisa smo se odločili, da uporabimo WCF v .NET platformi [28]. WCF je ogrodje za izgradnjo povezanih, servisno usmerjenih aplikacij. Odjemalec lahko uporablja več servisov, servis pa lahko uporablja več aplikacij hkrati. Servisi so ohlapno povezani med seboj, uporaba servisa pa ni vezana na .NET platformo. Spletni servis mora nuditi celovito podporo podatkov naši aplikaciji. Po tehtnem premisleku in analizi obsega naše mobilne aplikacije, smo pripravili štiri klice servisa:

- `GetAmenitiesJson2`: GET metoda vrne JSON zapis s seznamom vseh skupin dobrin in dobrin, ki pripadajo določeni skupini, ki so na voljo v naši podatkovni bazi,
- `GetDataByAmenityIdJsonLatLon/{amenityId}/{lat}/{lon}`: GET metoda vrne JSON zapis z vsemi lokacijami, ki ustrezajo izbrani dobrini in se nahajajo v kvadratu podane geografske širine in dolžine. Kvadrat je določen z odmikom ene kotne stopinje v pozitivno in negativno smer za obe koordinati,
- `GetDataDetailsJson/{osmId}`: GET metoda vrne JSON zapis, ki vsebuje vse podrobnosti določene lokacije,
- `PostComment`: POST metoda, ki sprejme JSON zapis, v katerem se nahaja komentar, ki ga je vnesel uporabnik preko aplikacije. Zapis se vstavi v podatkovno bazo in se veže na določeno lokacijo.

## 4.4 Priprava logičnega podatkovnega modela

Za vnesene podatke smo za potrebe aplikacije predhodno pripravili ustrezen podatkovni model (slika 7).



Slika 7: Logični podatkovni model

V logičnem podatkovnem modelu najdemo naslednje entitete.

- **OSM\_Data**: v tej tabeli se hranijo podatki o lokacijah. O lokaciji se hrani geografska širina in dolžina kot obvezna podatka ter država, v kateri se lokacija nahaja. Lokacija je vezana na točno eno dobrotno. Lokacija ima lahko enega ali več zapisov, ki opisujejo podrobnost lokacije.
- **OSM\_DataDetails**: v tej tabeli se nahajajo zapisi, ki opisujejo podrobnosti lokacije. Ima dve polji za hrambo podatkov. Prvo polje je ključ, ki opisuje vrsto podrobnosti (recimo ime), drugo polje pa hrani vrednost ključa. Vsak podatek o podrobnosti lokacije je vezan na eno lokacijo.

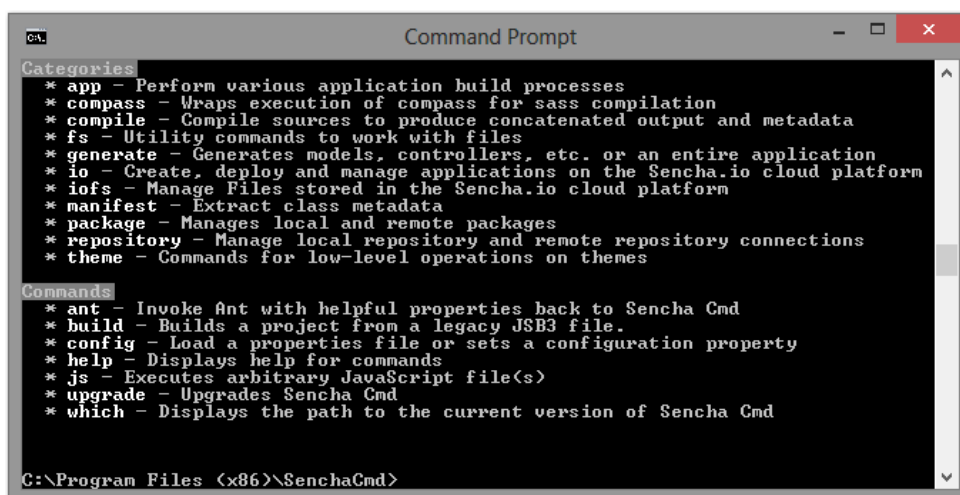
- OSM\_Comments: v tej tabeli se hranijo komentarji, ki jih objavijo uporabniki aplikacije. Komentar se veže na eno lokacijo, ena lokacija pa ima lahko nič ali več komentarjev
- OSM\_Amenity: v tej tabeli se hranijo vse dobrine, ki so na voljo. Gre za šifrant dobrin. O dobrini se hrani šifra, uporabniku prijazno ime in komentar, ki podrobneje opiše dobrino. Dobrina se lahko veže na nič ali več lokacij, lokacija pa ima lahko samo en tip dobrine.
- OSM\_AmenityGroup: v tej tabeli se hrani šifrant skupin, kamor pripada določena dobrina. Dobrina pripada točno eni skupini, ena skupina pa lahko vsebuje nič ali več dobrin.

## 4.5 Izdelava aplikacije Sencha Touch 2

### 4.5.1 Priprava projekta

Po določenih smernicah in obsegu aplikacije, pripravljenem servisu in napolnjeni podatkovni bazi je sledil bistveni del diplomske naloge, izdelava Sencha Touch aplikacije. Začeli smo seveda na začetku, s pripravo novega Sencha Touch projekta. Najhitrejši in obenem najenostavnejši način za izdelavo novega projekta je z uporabo orodja SenchaCMD. SenchaCMD je konzolna aplikacije (slika 8), ki nudi celovito podporo aplikaciji med njenim razvojem. Tako smo za pripravo novega projekta vnesli ukaz podoben sledečemu :

"sencha -sdk /pot/do/sencha/sdk generate app ImeAplikacije /pot/do/aplikacije" [29]. Pri tem smo morali podati pot, kjer se nahaja Sencha Touch SDK, vpisati ukaz generate za generiranje aplikacije, podati ime naše aplikacije in lokacijo direktorija, kjer se bo ustvarila.

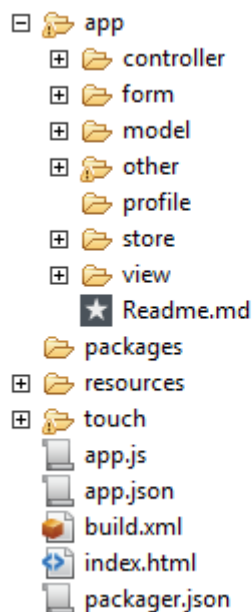


Slika 8: SenchaCMD

Po uspešno ustvarjenem projektu smo le tega dodali v Aptana Studio, ki nam je služil kot urejevalnik.

#### 4.5.2 Izdelava ogledov, modelov in kontrolnih razredov

Ko odpremo imenik, kamor smo ustvarili našo Sencha Touch aplikacijo, nas pričaka drevesna struktura, kot jo vidimo na sliki 9.



Slika 9: Struktura projekta Sencha Touch 2

V imeniku `touch` se nahajajo JavaScript datoteke, knjižnice, ki jih Sencha Touch potrebuje za delovanje. V imeniku `resources` so shranjene CSS datoteke, ki določajo izgled aplikacije. Tu se nahajajo tudi datoteke, ki jih potrebuje aplikacija, na primer ikone, slike, JSON ali XML datoteke s podatki, ipd. Imenik `app` je bistveni del aplikacije s stališča razvijalca. Tu se nahajajo ogledi, modeli in kontrolni razredi, ki jih je razvijalec ustvaril med razvojem aplikacije.

Najprej smo ustvarili potrebne datoteke v modelnem, oglednem in kontrolnem imeniku. Reference na te datoteke smo morali dodati še v `app.js` datoteko, ki se nahaja v korenu drevesne strukture. `app.js` je na nek način bistveni del aplikacije, saj je potrebno v njem navesti vse razrede, ki smo jih ustvarili, hkrati pa služi kot neke vrste glavna (ang. main) metoda, saj se v njej ustvari prvi ogled.

Prvi ogled smo predstavili v obliki ugnezdenega seznama. Seznam mora uporabniku najprej prikazati skupine dobrin, nato pa se ob dotiku na določeno skupino prikažejo vse dobrine znotraj skupine. Naloga ni bila pretirano zahtevna, saj Sencha Touch ponuja že izdelano komponento ugnezden seznam [30] (ang. nested list), ki deluje na tak način, kot smo si zamislili. Z uporabo te komponente smo dva koraka združili v enega, saj nam ni bilo potrebno ustvariti dveh ločenih seznamov za prikaz skupin dobrin in dobrin, ki pripadajo določeni skupini. Z uporabo ugnezdenega seznama smo naš ogled enostavno razširili (ang. extend) s komponento `Ext.dataview.NestedList`. Ogledu smo določili še obliko glave in elementov in urejanje ogleda je s tem zaključeno (izvorna koda 3).

```
Ext.define('DipApp.view.Amenities', {
    extend: 'Ext.dataview.NestedList',
    xtype: 'amenititeslist',

    requires: [
        'DipApp.store.Amenities'
    ],

    config: {
        title: 'Amenities Group',
        useTitleAsBackText: false,
        onItemDisclosure: false,
        store: 'Amenities',
        zIndex: 0
    },

    getTitleTextTpl: function() {
        return '<div>{name}</div>';
    },
    getItemTextTpl: function(node) {
        return '<div><strong>{name}</strong></div>';
    }
});
```

Izvorna koda 3: Prvi ogled

Poleg ogleda smo ustvarili še njemu pripadajoči kontrolni in modelni razred ter shrambo (ang. store). V kontrolni razred smo dodali funkcijo, ki naloži drugi ogled, ki prikaže lokacije z izbrano dobrino v okolici, v modelu pa smo opisali strukturo podatkov. Z uporabo shrambe (izvorna koda 4) smo pridobili podatke preko klica na spletni servis ter jih shranili v modele. Pri tem smo morali biti pozorni, da smo imena nosilcev podatkov v modelu poimenovali enako kot imena nosilcev podatkov v JSONu, ki ga pridobimo preko spletnega servisa.

```

Ext.define('DipApp.store.Amenities',{
    extend: 'Ext.data.TreeStore',
    config: {
        autoLoad: true,
        model: 'DipApp.model.Amenities',
        proxy: {
            type:'rest',
            url: Consts.service_url + 'GetAmenitiesJson2',
            reader: {
                type: 'json',
                rootProperty: 'items'
            }
        }
    }
});

```

Izvorna koda 4: Primer klica servisa v shrambi

Ob koncu izdelave prvega ogleda je bila aplikacija videti, kot jo prikazuje slika 10. Ob tem moramo tudi povedati, da se z izgledom aplikacije nismo posebno ukvarjali in smo uporabili kar privzet izgled komponent.

| Amenities Group               |
|-------------------------------|
| Education                     |
| Entertainment, Arts & Culture |
| Financial                     |
| Healthcare                    |
| Others                        |
| Sustenance                    |
| Transportation                |

Slika 10: Izgled dokončanega prvega ogleda

Drugi ogled je v marsičem podoben prvemu, le da smo tu podatke predstavili z navadnim seznamom. Tako smo naš ogled razširili s komponento `Ext.dataview.List` [31]. Ogledu smo dodali še naslovno vrstico (ang. titlebar) in gumb, ki nas vrne nazaj na prvi ogled. Obe komponenti sta dodani kot `xtype` v konfiguraciji ogleda. `Xtype: 'ime komponenta'` lahko razumemo kot bližnjico do polnega imena komponente. Tako na primer komponento gumb namesto z `Ext.button.Button` ustvarimo z `xtype:`

'button'. Xtype (izvorna koda 5) se uporablja za namene optimizacije komponente in prinaša večjo preglednost aplikacijske kode, posebno če je ta obsežnejša [32].

```
// eksplicitno ustvarjena komponenta:
var panel = new Ext.Panel({
    ...
    items: [
        Ext.create('Ext.button.Button', {
            text: 'OK'
        })
    ]
});

// implicitno ustvarjena komponenta:
var panel = new Ext.Panel({
    ...
    items: [{
        xtype: 'button',
        text: 'OK'
    }]
});
```

Izvorna koda 5: Primer uporabe xtype

Poleg drugega ogleda smo ustvarili še njemu pripadajoča kontrolni in modelni razred ter shrambo. Na dogodek `dotik` v kontrolnem razredu smo dodali funkcijo, ki naloži podatke o izbrani lokaciji na podlagi identifikacijske številke lokacije ter jih posreduje tretjemu ogledu. Dodali smo še funkcijo za vračanje nazaj na prvi ogled.

Tretji ogled smo razširili iz komponente `panel`. Plošči smo ugnezdili naslovno vrstico (ang. titlebar) z gumbom, ki nas vrne nazaj na drugi ogled. Plošči smo ugnezdili tudi komponento `tabpanel`, v ploščo z zavihki pa dodali štiri zavihke:

- prvi zavihek za pregled podrobnosti lokacije,
- drugi zavihek za prikaz lokacije na zemljevidu,
- tretji zavihek za pregled komentarjev lokacije in
- četrti zavihek za oddajo komentarja.

Prvi zavihek vsebuje komponento `list` in prikaže vse informacije, ki so na voljo o določeni lokaciji. Drugi zavihek uporablja komponento `zemljevid` (ang. map). Komponenta `zemljevid` je razširjena iz Google zemljevidov in uporablja njegove API [33]. Poleg prikaza zemljevida je bilo v kontrolnem razredu potrebno nastaviti še geografsko širino in dolžino, kamor naj nastavi pogled zemljevida in označi lokacijo na zemljevidu. Tretji zavihek uporablja komponento `seznam` in prikaže vse komentarje, ki so na voljo o izbrani lokaciji. Četrti zavihek uporablja komponento `formpanel`, ki vsebuje vnosno polje za komentar in dva

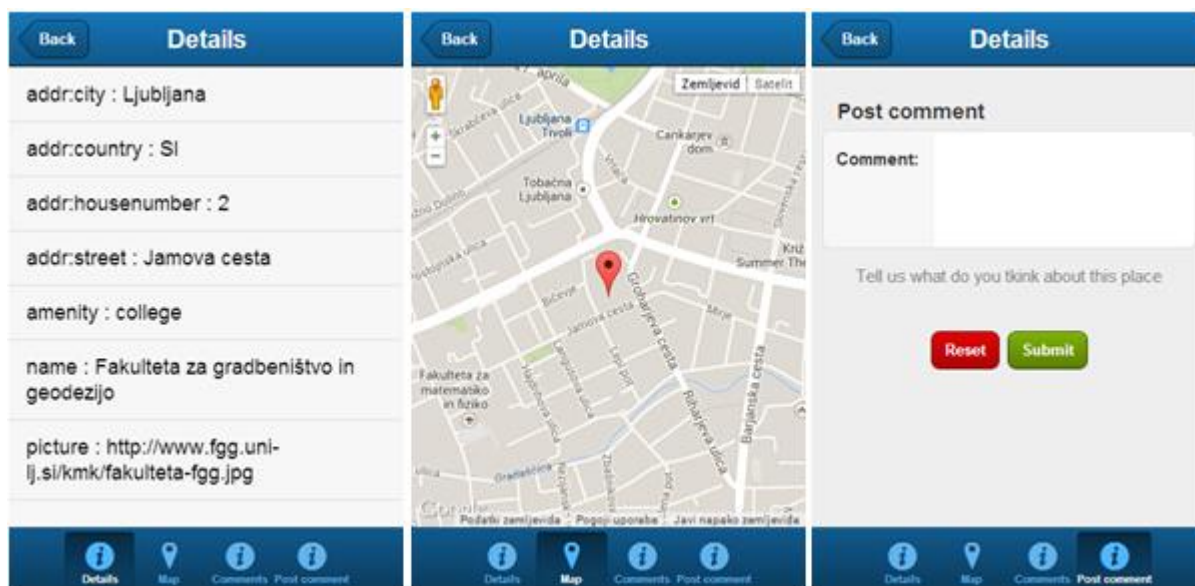
gumba; enega za ponastavitev vrednosti v vnosnem polju in drugega za oddajo komentarja. Funkciji za oddajo in ponastavitev komentarja se nahajata v kontrolnem razredu.

Podatki, ki so bili prikazani v tretjem ogledu, so bili pridobljeni s samo enim klicem spletnega servisa in so bili v aplikacijo prineseni v obliki ugnезdenega JSON zapisa. Temu je bilo potrebno prilagoditi model, in sicer z uporabo asociacij. Z uporabo asociacije (izvorna koda 6) v Sencha Touch 2 dosežemo, da lahko v modele shranimo ugnезden JSON, saj nam omogočijo, da tako izrazimo razmerja med različnimi modeli [34].

```
Ext.define('DipApp.model.Place', {
    extend: 'Ext.data.Model',
    requires: ['DipApp.model.Details', 'DipApp.model.Comments'],
    config: {
        fields: [
            {name: 'OSMDataID', type: 'int'},
            {name: 'Lat', type: 'float'},
            {name: 'Lon', type: 'float'},
            {name: 'Country', type: 'string'}
        ],
        associations : [ //zaradi associations se bodo naložili tudi podatki o Details in Comments
            {
                name: 'Details',
                type: 'hasMany',
                model: 'DipApp.model.Details',
                //primaryKey: 'OSMDataDetailsID',
                //foreignKey: 'OSMDataID',
                autoLoad: true,
                associationKey: 'Details'
            },
            {
                name: 'Comments',
                type: 'hasMany',
                model: 'DipApp.model.Comments',
                //primaryKey: 'CommentID',
                //foreignKey: 'OSMDataID',
                autoLoad: true,
                associationKey: 'Comments'
            }
        ]
    }
});
```

Izvorna koda 6: Prikaz uporabe asociacij v modelih Sencha Touch 2

Ko smo končali z urejanjem tretjega ogleda, je bila naša aplikacija videti tako, kot jo prikazuje slika 11. S samim izgledom aplikacije (razen s postavitvijo komponent) se nismo ukvarjali.



Slika 11: Prikaz dokončanega tretjega ogleda

Na tej točki smo razvoj aplikacije zaključili. Končen produkt je bila uporabna aplikacija, ki nam je za izbrano dobrino prikazala vse lokacije in vse podrobnosti določene dobrine na določeni lokaciji. Sama aplikacija je bila razvita kot zaključena celota, ki bi se jo dalo objaviti na spletni trgovini. Preostalo nam je samo še, da podamo oceno ogrodja Sencha Touch 2, ki se je izoblikovala med razvojem aplikacije.



## 5 Ocena Sencha Touch 2 po kriterijih

### 5.1 Ocenjevanje, kriteriji in uteži kriterijev

Za ocenjevanje izvedbe aplikacije smo se odločili, da uporabimo princip Libertove skale [35] z ocenami od 1 do 5, pri čemer je:

- 1 = zelo slabo,
- 2 = slabo,
- 3 = povprečno,
- 4 = dobro,
- 5 = zelo dobro.

Končno oceno izvedbe bomo določili po principu uteženega povpračja [36].

$$\sum \frac{\omega_1 \times o_1 + \omega_2 \times o_2 + \dots}{\omega_1 + \omega_2 + \dots}$$

Ker uporabniki aplikacij največkrat cenijo druge kriterije kot razvijalci, je potrebno poudariti, da bomo kriterije izbrali in utežili z vidika razvijalca aplikacije. Izvedbo aplikacije Sencha Touch 2 bomo merili na podlagi naslednjih kriterijev:

- krivulja učenja,
- dokumentacija,
- čas izvedbe,
- prilagodljivost,
- umestitev na različne sisteme,
- hitrost delovanja.

**Krivulja učenja** bo imela najmanjšo težo. Učenje neke tehnologije je največkrat enkratno dejanje, hkrati pa zelo subjektivno. Učenje novih funkcionalnosti neke tehnologije pa po navadi ne vzame veliko napora ob predpostavki, da tehnologijo predhodno dobro poznamo.

**Utež: 5.**

**Dokumentiranost** je navadno za programerja zelo pomembna. Praktično nemogoče je poznati vse podrobnosti tehnologije, posebej če je ta deležna nenehnih dopolnitev in razvoja. Dobra dokumentiranost programerju tudi olajša delo, saj med svojim delom velikokrat naleti na problem, ki ga je nekdo že uspešno rešil, tako pa si velikokrat prihrani dragoceni čas. **Utež: 20.**

**Čas realizacije** je velikokrat pomemben tako razvijalcu kot naročniku aplikacije. Hitrejši čas realizacije prinese naročniku manjše stroške, razvijalcu navadno prinese večji dobiček, saj lahko v istem času obdela več naročil. **Utež: 25.**

**Prilagodljivost** je pomembna iz vidika nadaljnjega razvoja aplikacije. Premišljena zasnova lahko močno poenostavi vse nadaljnje nadgradnje. **Utež: 15.**

**Umestitev aplikacije na različne mobilne sisteme** je zelo pomembna. Cilj naročnika aplikacije je prav gotovo v tem, da bi njegovo aplikacijo uporabljalo čim večje število ljudi, hkrati pa je pomembno, da je cena umestitve na različna okolja čim nižja. **Utež: 25.**

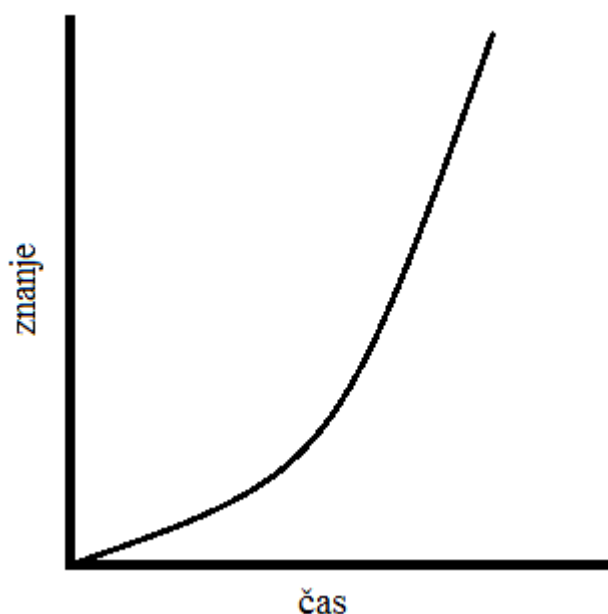
**Hitrost delovanja** aplikacije je največkrat relativen pojem. Veliko je odvisno od same zmogljivosti mobilne naprave, na kateri teče aplikacija, odvisno pa je tudi od uporabnika samega. Kar je za nekatere sprejemljiva hitrost delovanja, je lahko za druge popolnoma nespremenljivo. Zaradi tega hitrost delovanja nima pretirano visoke uteži. **Utež: 10.**

## 5.2 Krivulja učenja

Krivulja učenja je grafična predstavitev povečanja znanja, predstavljenega na vertikalni osi s količino vložene časa, predstavljenega na horizontalni osi. Krivuljo učenja lahko pri velikem številu ponovitev opišemo z matematično funkcijo. Krivulja učenja igra pomembno vlogo, ko se moramo naučiti uporabljati novo tehnologijo, hkrati pa smo časovno omejeni [37].

V učenje ogrodja Sencha Touch 2 smo vložili kar nekaj časa. Z ogrođjem smo se namreč srečali prvič. Najtežji del na začetku je učenje in razumevanje strukturiranosti ogrodja in učenje sintakse. MVC arhitekturni tip pa še dodatno oteži zadevo. Ko smo se dobro seznanili s strukturo ogrodja in se naučili, kako deluje MVC v Sencha Touch 2, smo veliko lažje in hitreje nadgrajevali znanje.

Glede na to, da je HTML, CSS in JavaScript tema, ki smo se jo učili tudi na fakulteti, lahko rečemo, da bi bilo razvijalcu lažje in hitreje narediti neko HTML5 aplikacijo za mobilne naprave brez uporabe ogrodja, vendar samo na začetku in do določene zahtevnosti. Slika 12 prikazuje oceno učne krivulje za ogrodje Sencha Touch 2. **Število točk: 2**



Slika 12: Krivulja učenja Sencha Touch 2

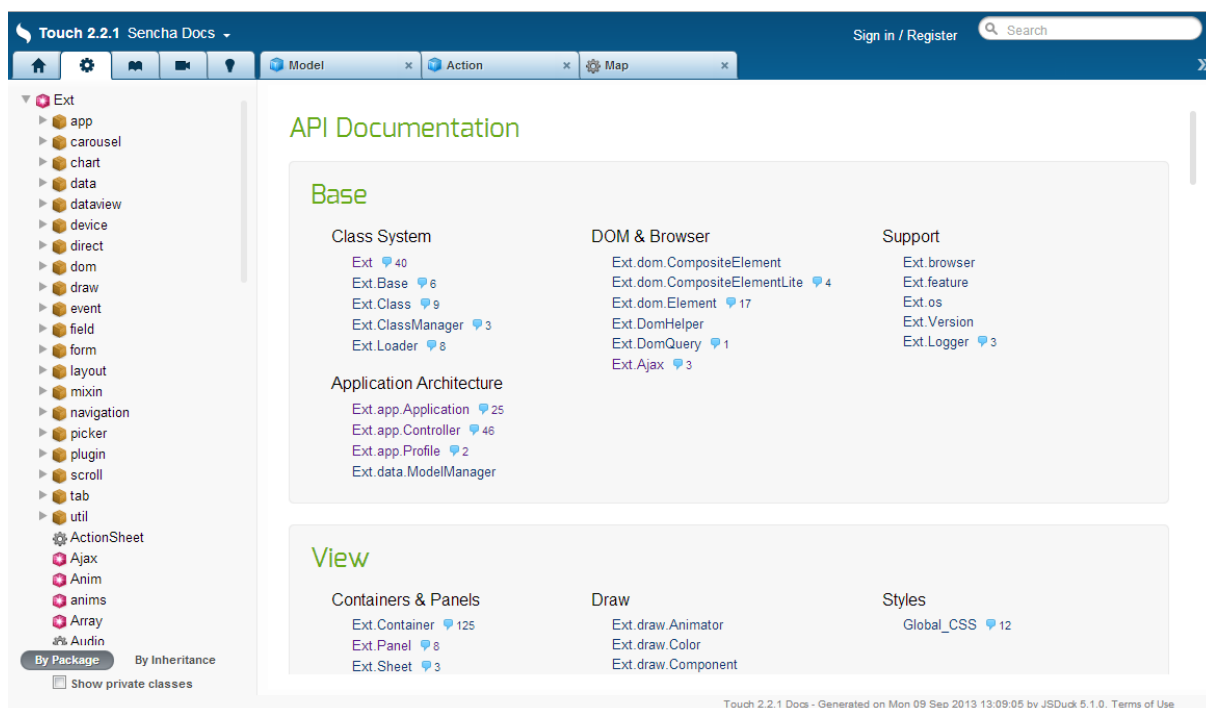
### 5.3 Dokumentiranost

Dobra dokumentiranost vpliva na več faktorjev razvoja. Dobro dokumentirana ogrodja zmanjšujejo čas učenja arhitekture ter povečujejo produktivnost razvijalca, posledično pa skrajšujejo čas realizacije. Za dokumentiranost šteje tako uradna dokumentacija, objavljena s strani razvijalca arhitekture, kot neuradna dokumentacija, ki zajame različne primere, vodiče, forume, ipd.

Dokumentiranost ogrodja Sencha Touch 2 je izjemna. Uradna dokumentacija je na voljo tako preko spleta kot tudi za delo brez povezave, ki jo dobimo ob namestitvi ogrodja. Dokumentacija je narejena strukturirano in pregledno (slika 13). Razvijalcu so na voljo tako popisi posameznih komponent kot primeri v celoti izdelanih aplikacij. Poleg dokumentacije je na voljo tako uraden forum Sencha Touch, kjer razvijalci lahko postavijo vprašanje glede njihovega problema, odgovor pa lahko podajo drugi razvijalci ali pa velikokrat ljudje, ki jih zaposluje Sencha. Proti plačilu je na voljo tudi direktna pomoč tehnika. Poleg uradne dokumentacije pa je na spletu na voljo veliko število primerov in vodičev za izvedbo raznih zadev.

V primerjavi z dokumentacijo, ki jo ponuja recimo Microsoft Developer Network, ki smo jo prav tako uporabljali med izdelavo diplomskega dela, je dokumentacija Sencha Touch 2 veliko bolj pregledna. Iskalnik deluje hitreje in ti hitreje ponudi možne rezultate v spustnem seznamu. Ko izbereš iskano stvar, ti prikaže rezultat in ne nabora povezav do možnih rezultatov iskanja. Dokumentacija vsebuje tudi več primerov, pri katerih je velikokrat mogoče

preveriti delovanje kar na spletu. Obenem pa omogoča tudi komentar uporabnika oziroma razvijalca k vsaki komponenti ogrodja. **Število točk: 5.**



Slika 13: Uradna dokumentacija Sencha Touch 2 verzija 2.2.1.

## 5.4 Čas realizacije

Za čas realizacije šteje čas od začetka pa do konca razvoja nekega segmenta aplikacije, lahko tudi celotne aplikacije.

Če zanemarimo vse priprave, ki so bile potrebne za realizacijo naše aplikacije (priprava podatkov, priprava spletnega servisa), lahko aplikacijo na grobo razdelimo po ogledih na tri segmente.

Za realizacijo prvega ogleda smo porabili približno 4 ure. Če bi ogrodje bolje poznali, bi bil čas realizacije prvega ogleda bistveno krajši. Vendar če primerjamo količino časa, ki bi ga morali vložiti, če bi želeli primerljivo funkcionalnost z vsemi animiranimi prehodi, ne da bi uporabili katerokoli ogrodje, je 4 ure slišati skoraj zanemarljivo malo. Zato smo se odločili, da hitrost realizacije prvega ogleda ocenimo z oceno 5.

Za realizacijo drugega ogleda smo porabili 8 ur časa. Ogled je dokaj preprost in v primerjavi s prvim ogledom tudi na videz preprostejši. Vendar smo tu naleteli na težavo s preходом iz prvega ogleda na drugega in nazaj. Prvega ogleda nismo smeli uničiti, saj bi ob prehodu nanj ponovno čakali, da se naložijo podatki in tako nesmiselno trčili čas in količino prenesenih

podatkov. Prav tako pa smo imeli nekaj težav, da smo pridobili geografsko širino in dolžino, ki smo jo potrebovali za prikaz rezultatov. Za drugi ogled smo se odločili, da ga ocenimo z oceno 3.

Za realizacijo tretjega ogleda smo porabili 25 ur časa. Izkazal se je za najzahtevnejšega. Tretji ogled vsebuje štiri zavihke. Na prvem zavihku je seznam vseh podrobnosti, ki so na voljo o določeni lokaciji. Glede na to, da smo imeli s seznamom opravka že v prvem in drugem ogledu, nam tu ni povzročal kakšnega dodatnega dela. Prav tako nismo imeli težav s prikazom seznama vseh komentarjev na tretjem zavihku. Veliko več težav nam je povzročil drugi zavihke, na katerem se je nahajal zemljevid, in sicer dinamičen prikaz točke na zemljevidu in pozicioniranje zemljevida na izbrano točko. Izkazalo se je, da je bila težava v tem, da smo klicali funkcijo za dodajanje točke na zemljevid, še preden so se zemljevidi v celoti naložili. Četrty zavihke je vseboval obrazec za oddajo komentarja. Z uporabo dobrega primera nam izvedba ni povzročala težav. Še nekaj preglavic nam je povzročil gumb za vračanje nazaj na drugi ogled. Za tretji ogled smo se odločili, da ga ocenimo z oceno 2, predvsem zaradi velike količine časa, ki smo ga izgubili z zemljevidom.

Če predpostavimo, da smo za izdelavo aplikacije v celoti porabili približno 37 ur časa, ter ob tem zanemarimo čas, ki smo ga porabili za pripravo spletnega servisa, podatkov in prebiranja dokumentacije, je 37 ur slišati relativno malo. Ob tem je potrebno imeti v mislih oceno časa, ki bi ga porabili za izdelavo aplikacije brez uporabe ogrodja. Z gotovostjo lahko trdimo, da tudi najbolj izkušenim HTML5, CSS, JavaScript razvijalcem ne bi uspelo v enakem časovnem intervalu izdelati enakovredno primerljive aplikacije z vsemi animiranimi prehodi, s prilagodljivostjo na velikost zaslona in grafično podobo, ki na videz deluje enako kot aplikacija, ki je bila ustvarjena na rodni platformi mobilne naprave. Čeprav smo imeli pri izdelavi nekaj težav, ki so bile delno povezane tudi z našim nepoznavanjem platforme, lahko podamo oceno časa realizacije, ki znaša  $(5 + 3 + 2) / 3 = 3,33$ . **Število točk: 3,33.**

## 5.5 Prilagodljivost

Ob predpostavki, da bo aplikacija deležna dopolnitev in novih funkcionalnosti, je prilagodljivost pomemben faktor. Premišljena zasnova aplikacije lahko zelo pripomore k poenostavitvi vseh nadaljnjih nadgradenj ter skrajšuje čas realizacije.

Sencha Touch 2 uporablja MVC arhitekturni slog, ki je sam po sebi zelo prilagodljiv za spremembe. Tako smo na primer porabili samo 15 minut časa, da smo odstranili ogled, ki je prikazoval podrobnosti dobrin med prvim in drugim ogledom. Ob tem smo morali biti tudi pozorni, da smo odstranili vse reference, ki so se navezovala na izbrisan ogledni in kontrolni razred. Ker Sencha Touch 2 sama po sebi nima namenskega orodja za razhroščevanje in iskanje napak (kot ga ima na primer Visual Studio), je potrebno aplikacijo skrbno spreminjati in obenem dobro poznati strukturo ogrodja. **Število točk: 4.**

## 5.6 Umestitev

Z razvojem aplikacije si želimo, da bi ta bila podprta na vseh večjih mobilnih platformah, kar pa velikokrat pomeni velike stroške pri izvedbi, sploh če razvijamo aplikacijo za vsako platformo posebej.

Sencha Touch 2 temelji na HTML5, CSS in JavaScriptu. Gledano s tega vidika gre za spletno aplikacijo, ki jo zaženemo v spletnem brskalniku. Splošno rečeno je ustrezen brskalniki edina stvar, ki jo potrebujemo za delovanje Sencha Touch 2 aplikacije. Tako potrebujemo samo programsko opremo, v kateri izdelamo namestitveni paket za specifično mobilno platformo. Ob tem se seveda ne bomo mogli izogniti nakupu in plačilu razvijalskih računov za različne platforme, kot so Android, iOS in Windows 8. Vendar se to izkaže za zanemarljivo stvar v primerjavi, da bi morali aplikacijo razvijati za vsako platformo posebej. **Število točk: 5.**

## 5.7 Hitrost delovanja

Hitrost delovanja aplikacije je poleg dobre strukturiranosti eden pomembnejših temeljev dobre uporabniške izkušnje. Vendar se ob tem kriteriju velikokrat sklene kompromis, ko igrata hitrost razvoja aplikacije in njena umestitev na mobilne platforme pomembnejšo vlogo. Obenem pa je lahko hitrost delovanja zelo relativen pojem.

Predpostavimo lahko, da je hitrost delovanja aplikacije v rodnom jeziku platforme višja od hitrosti spletne aplikacije oziroma aplikacije napisane v Sencha Touch 2. V tem pogledu je delovanje Sencha Touch 2 aplikacije počasno, posebno če ta teče na starejši napravi. Na zmogljivejših napravah pa postaja hitrost delovanja vedno bolj primerljiva.

Delovanje smo primerjali empirično na dveh mobilnih napravah, na katerih je nameščen operacijski sistem Android. Na novejši napravi se je aplikacija naložila hitreje kot na starejši, a je kljub temu za dobro uporabniško izkušnjo nalaganje trajalo predolgo časa. V povprečju je zagon aplikacije trajal približno 15 sekund. Na novejši napravi je bilo samo delovanje aplikacije bolj tekoče. Prehodi med ogledi so trajali različno dolgo, zaradi klicev spletnega servisa in nalaganja podatkov, vendar so se na novejši napravi izvršili hitreje. Preklapljanje med zavihki je na novejši napravi delovalo tekoče z občasnim sekundnim zamikom. Na starejši napravi so prehodi med ogledi trajali dalj časa. Tudi samo delovanje je bilo manj tekoče in pogosto zatikajoče in slabše odzivno. Gledano s perspektive srednje zahtevnega uporabnika je na novejši napravi aplikacija delovala dobro. Slabo pa se je odrezala na starejši mobilni napravi. Skupaj se je tako odrezala povprečno. **Število točk: 3.**

## 5.8 Rezultati

| KRITERIJ                      | ŠTEVILO TOČK |
|-------------------------------|--------------|
| Krivulja učenja               | 2            |
| Dokumentacija                 | 5            |
| Čas izvedbe                   | 3,33         |
| Prilagodljivost               | 4            |
| Umestitev na različne sisteme | 5            |
| Hitrost delovanja             | 3            |
| OCENA                         | <b>4,08</b>  |

Tabela 1: Ocena izračunana na podlagi uteženega povprečja

Skupna ocena izdelave aplikacije z uporabo Sencha Touch 2 znaša 4,08. Tako lahko zaključimo, da se je Sencha Touch 2 izkazala kot dobro ogrodje za izdelavo mobilnih aplikacij. Slabše se je odrezala pri hitrosti delovanja. Porabili smo kar nekaj časa, da smo se seznanili z okoljem, naše nepoznavanje ogrodja pa je tudi negativno vplivalo na čas izvedbe. Vendar če pogledamo, da smo ob koncu izdelave aplikacije imeli podprte vse pomembnejše mobilne platforme, lahko zaključimo, da je Sencha Touch 2 uporabno ogrodje in dobro služi svojemu namenu. Obenem pa je deležna stalnega razvoja in izboljšav.



## 6 Zaključek

Cilj diplomskega dela je bila predstavitev ogrodja Sencha Touch 2. Funkcionalnost in delovanje ogrodja smo predstavili na primeru koristne aplikacije, ki smo jo izdelali za potrebe diplomskega dela.

Diplomsko delo smo v splošnem razdelili na dva dela. V prvem delu smo predstavili ogrodje Sencha Touch 2. Predstavili smo kratko zgodovino ogrodja in opisali pomembnejše funkcije, ki jih ponuja. Zadali smo si smernice, ki smo jim sledili pri demonstraciji izdelave Sencha Touch 2 aplikacije. V drugem delu smo podrobneje opisali proces izdelave aplikacije za namen diplomskega dela s poudarkom na izdelavi Sencha Touch 2 aplikacije. Diplomsko nalogo smo sklenili z oceno ogrodja. Ogrodje se je izkazalo kot dobro s predpostavko, da aplikacija teče na novejših oziroma zmogljivejših napravah.

Če bi nam bilo na voljo več časa, bi diplomsko delo lahko primerno nadgradili in izboljšali. Izdelali bi enakovredno aplikacijo za eno ali več mobilnih platform in nato primerjali aplikacije z ustreznim naborom kriterijev med seboj. Za objektivnejši rezultat bi lahko ustrezno razširili nabor kriterijev in bili obenem bolj pozorni na subjektivnost rezultatov.

Končna ocena diplomskega dela lahko uporabniku služi kot referenca in mu pripomore pri odločitvi o uporabi ogrodja.



# Slike

|                                                                                     |    |
|-------------------------------------------------------------------------------------|----|
| Slika 1: Prenosni računalnik HP Pavilion g7 .....                                   | 3  |
| Slika 2: Visual Studio 2012 Professional .....                                      | 4  |
| Slika 3: SQL Server Management Studio .....                                         | 6  |
| Slika 4: MVC struktura Sencha Touch 2 .....                                         | 11 |
| Slika 5: Ogled skupin dobrin, dobrin znotraj skupine in seznam lokacij dobrin ..... | 14 |
| Slika 6: Tretji ogled, prikaz zemljevida in forme za oddajo komentarja .....        | 14 |
| Slika 7: Logični podatkovni model .....                                             | 18 |
| Slika 8: SenchaCMD .....                                                            | 19 |
| Slika 9: Struktura projekta Sencha Touch 2 .....                                    | 20 |
| Slika 10: Izgled dokončanega prvega ogleda .....                                    | 22 |
| Slika 11: Prikaz dokončanega tretjega ogleda .....                                  | 25 |
| Slika 12: Krivulja učenja Sencha Touch 2 .....                                      | 29 |
| Slika 13: Uradna dokumentacija Sencha Touch 2 verzija 2.2.1. ....                   | 30 |

# Izvorna koda

|                                                                         |    |
|-------------------------------------------------------------------------|----|
| Izvorna koda 1: Primer strukture XML datoteke OpenStreetMaps .....      | 15 |
| Izvorna koda 2: Razčlenjevalnik XML datoteke OpenStreetMaps .....       | 16 |
| Izvorna koda 3: Prvi ogled .....                                        | 21 |
| Izvorna koda 4: Primer klica servisa v shrambi .....                    | 22 |
| Izvorna koda 5: Primer uporabe xtype .....                              | 23 |
| Izvorna koda 6: Prikaz uporabe asociacij v modelih Sencha Touch 2 ..... | 24 |

# Tabele

|                                                                 |    |
|-----------------------------------------------------------------|----|
| Tabela 1: Ocena izračunana na podlagi uteženega povprečja ..... | 33 |
|-----------------------------------------------------------------|----|



# Literatura in viri

- [1] (2013) Microsoft Visual Studio. Dostopno na:  
[http://en.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](http://en.wikipedia.org/wiki/Microsoft_Visual_Studio)
- [2] (2013) IntelliSense. Dostopno na:  
[http://en.wikipedia.org/wiki/Intelligent\\_code\\_completion](http://en.wikipedia.org/wiki/Intelligent_code_completion)
- [3] (2013) REST: Dostopno na:  
[http://en.wikipedia.org/wiki/Representational\\_state\\_transfer](http://en.wikipedia.org/wiki/Representational_state_transfer)
- [4] (2013) Microsoft SQL Server 2012 Express. Dostopno na:  
<https://www.microsoft.com/en-us/sqlserver/editions/2012-editions/express.aspx>  
<http://www.microsoft.com/en-us/download/details.aspx?id=29062>
- [5] (2013) Microsoft SQL Server 2012 Express omejitve. Dostopno na:  
<http://msdn.microsoft.com/en-us/library/cc645993%28v=SQL.110%29.aspx>
- [6] (2013) SQL Server Management Studio. Dostopno na:  
[http://en.wikipedia.org/wiki/SQL\\_Server\\_Management\\_Studio](http://en.wikipedia.org/wiki/SQL_Server_Management_Studio)
- [7] (2014) Windows Presentation Foundation. Dostopno na:  
[http://en.wikipedia.org/wiki/Windows\\_Presentation\\_Foundation](http://en.wikipedia.org/wiki/Windows_Presentation_Foundation)
- [8] (2013) Sencha CMD. Dostopno na: <http://www.sencha.com/products/touch/>
- [9] (2013) Sencha CMD 3.1.2 dokumentacija. Dostopno na:  
<http://docs.sencha.com/cmd/3.1.2/#!/guide/command>
- [10] (2013) Aptana Studio 3. Dostopno na: <http://en.wikipedia.org/wiki/Aptana>
- [11] (2013) Aptana Studio. Dostopno na: <http://www.aptana.com/legal>
- [12] (2013) Android SDK. Dostopno na: <http://developer.android.com/sdk/index.html>
- [13] (2013) Android SDK. Dostopno na:  
[http://en.wikipedia.org/wiki/Android\\_software\\_development](http://en.wikipedia.org/wiki/Android_software_development)
- [14] (2014) Ext JS. Dostopno na: <https://www.sencha.com/products/extjs/>
- [15] (2014) jQTouch. Dostopno na: <http://jqtouch.com/>
- [16] (2014) Raphaël. Dostopno na: <http://dmitrybaranovskiy.github.io/raphael/>
- [17] (2013) Sencha Touch. Dostopno na: [http://en.wikipedia.org/wiki/Sencha\\_Touch](http://en.wikipedia.org/wiki/Sencha_Touch)
- [18] (2013) Sencha Touch 2. Dostopno na:  
<http://www.sencha.com/products/touch/features/>

- [19] Adrian Kosmaczewski, Sencha Touch 2 Up and Running. Sebastopol: O'Reilly Media, Inc., 2013, pogl. 1
- [20] Adrian Kosmaczewski, Sencha Touch 2 Up and Running. Sebastopol: O'Reilly Media, Inc., 2013, pogl. 1
- [21] (2013) Apache Cordova. Dostopno na: <http://www.sencha.com/products/touch>
- [22] Adrian Kosmaczewski, Sencha Touch 2 Up and Running. Sebastopol: O'Reilly Media, Inc., 2013, pogl. 1
- [23] (2013) OpenStreetMaps regional data. Dostopno na: <http://download.geofabrik.de/europe.html>
- [24] (2013) XmlReader. Dostopno na: <http://msdn.microsoft.com/en-us/library/system.xml.xmlreader%28v=vs.110%29.aspx>
- [25] (2014) Entity Framework. Dostopno na: <http://msdn.microsoft.com/en-us/data/ef.aspx>
- [26] (2013) Entity Framework - data insert optimization. Dostopno na: <http://stackoverflow.com/questions/5940225/fastest-way-of-inserting-in-entity-framework>
- [27] (2013) REST: Dostopno na: [http://en.wikipedia.org/wiki/Representational\\_state\\_transfer](http://en.wikipedia.org/wiki/Representational_state_transfer)
- [28] (2013) WCF: Dostopno na: [http://en.wikipedia.org/wiki/Windows\\_Communication\\_Foundation](http://en.wikipedia.org/wiki/Windows_Communication_Foundation)
- [29] (2013) Sencha CMD: Dostopno na: <http://docs.sencha.com/cmd/3.1.2/#!/guide/command>
- [30] (2013) Sencha Touch 2 Nested list. Dostopno na: <http://docs.sencha.com/touch/2.3.1/#!/api/Ext.dataview.NestedList>
- [31] (2013) Sencha Touch 2 List. Dostopno na: <http://docs.sencha.com/touch/2.3.1/#!/api/Ext.dataview.List>
- [32] (2013) Sencha Touch 2 xtype. Dostopno na: <http://docs.sencha.com/touch/2.2.1/#!/api/Ext.Component-cfg-xtype>
- [33] (2013) Sencha Touch 2 Google zemljevidi. Dostopno na: <http://docs.sencha.com/touch/2.2.1/#!/api/Ext.Map>
- [34] (2013) Sencha Touch 2 Association. Dostopno na: <http://docs.sencha.com/touch/2.2.1/#!/api/Ext.data.association.Association>
- [35] (2013) Libertova skala. Dostopno na: [http://en.wikipedia.org/wiki/Likert\\_scale](http://en.wikipedia.org/wiki/Likert_scale)

- [36] (2013) Uteženo povprečje. Dostopno na:  
[http://sl.wikipedia.org/wiki/Ute%C5%BEna\\_funkcija](http://sl.wikipedia.org/wiki/Ute%C5%BEna_funkcija)
- [37] (2013) Krivulja učenja. Dostopno na: [http://en.wikipedia.org/wiki/Learning\\_curve](http://en.wikipedia.org/wiki/Learning_curve)