

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

**Marko Hlupič**

**PRIPOROČILNI SISTEM ZA NAČRTOVANJE  
IZLETOV**

**DIPLOMSKO DELO NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU**

Mentor: doc. dr. Janez Demšar

Ljubljana, 2011

Št. naloge: 00034/2010

Datum: 04.10.2010



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

Kandidat: **MARKO HLUPIČ**

Naslov: **PRIPOROČILNI SISTEM ZA NAČRTOVANJE IZLETOV**  
**RECOMMENDATION SYSTEM FOR TRIP PLANNING**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Priporočilni sistemi so predvsem zaradi razmaha spletne trgovine in podobnih storitev ena pomembnih aplikacij odkrivanja zakonitosti v podatkih. Kot primer takšnega sistema v okviru diplomske naloge implementirajte priporočilni sistem za svetovanje izletniških lokacij. Aplikacija naj podpira tako iskanje lokacij na podlagi uporabniško podanih kriterijev (vrsta izleta in aktivnosti, oddaljenost in podobno), registriranim uporabnikom pa naj nudi priporočila na podlagi njihovih profilov in profilov podobnih uporabnikov, kakor to počnejo drugi priporočilni sistemi. Aplikacijo poskusite na primernem številu testnih uporabnikov.

Mentor:

  
doc. dr. Janez Demšar

Dekan:

  
prof. dr. Nikolaj Zimic



# IZJAVA O AVTORSTVU

## diplomskega dela

Spodaj podpisani **Marko Hlupič**,

z vpisno številko **63030215**,

sem avtor diplomskega dela z naslovom:

### **Priporočilni sistem za načrtovanje izletov**

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom  
doc. dr. Janeza Demšarja
- 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. 04. 2011

Podpis avtorja:

## ***Zahvala***

*Rad bi se zahvalil svojemu mentorju, doc. dr. Janezu Demšarju, za strokovno vodenje in pomoč pri izdelavi diplomskega dela.*

*Zahvaljujem se svojim staršem za vso podporo in pomoč.*

*Zahvaljujem se tudi vsem ostalim prijateljem, sošolcem in sodelavcem, ki so mi na kakršenkoli način pomagali v času študija in pri pripravi tega diplomskega dela.*

# KAZALO VSEBINE

POVZETEK .....	1
ABSTRACT .....	3
1 UVOD.....	5
2 PODATKOVNI MODEL.....	6
2.1 Relacijski model .....	6
2.2 Zapisi v podatkovni bazi.....	10
2.3 Postopek za rangiranje lokacij .....	11
3 SPLETNA APLIKACIJA .....	14
3.1 Administrativni del .....	15
3.2 Uporabniški del.....	19
4 ZAKLJUČEK .....	26
VIRI IN LITERATURA.....	29

## **SEZNAM UPORABLJENIH KRATIC**

**RS** - priporočilni sistemi (angl. recommendation systems)

**kNN** - algoritem k najbližjih sosedov (angl. k nearest neighbors)

**ASP** – aktivne spletne strani (angl. active server pages)

**SQL** - povpraševalni jezik (angl. structured query language)

## **POVZETEK**

V diplomskem delu je predstavljena spletna aplikacija za priporočanje izletniških lokacij. Ker je na eni lokaciji mogoče početi več stvari, je lokacija razdeljena na lokacijo, ki vsebuje zgolj ime in v katero regijo spada, ter na možno aktivnost na tej lokaciji. Uporabniki beležijo izbrane aktivnosti in kasneje ocenjujejo, kako so bili zadovoljni z njimi. Aplikacija je ločena na dva dela in sicer na del, kjer priporočamo lokacije oziroma njihove aktivnosti neprijavljenim uporabnikom in na del, kjer priporočamo aktivnosti prijavljenim uporabnikom. Ker o neprijavljenem uporabniku ne vemo ničesar, mu lahko predlagamo le aktivnosti glede na kriterije, ki jih izbere na strani. Prijavljenemu uporabniku pa lahko predlagamo tudi aktivnosti, ki so jih obiskali drugi uporabniki, in to takšne, za katere vemo, da jih uporabnik še ni obiskal.

Ključne besede: priporočilni sistemi, sodelovalno filtriranje, algoritem kNN

## **ABSTRACT**

The thesis describes a web application used for recommending tourist destinations and trip planning. Given that one location offers several tourist attractions and activities, the destination is divided into a location, containing only the name and the corresponding region, and the possible activity that could take place in that location. The users choose some activities and later on decide whether these locations met their expectations or not. The application is divided into two parts: the first part, where we recommend locations and activities to users who are not registered, and the second part, intended for recommending locations to registered users. Since we do not have any information about the unregistered user, the activities can only be suggested to him or her based on the criteria that the user chooses on the website; whereas to the registered user, we are able to recommend also the activities that other users have visited, i.e. the ones that we know the user in question has not yet visited.

Key words: recommendation systems, collaborative filtering, kNN algorithm



# 1 UVOD

V današnjem obdobju se srečujemo s problemom pomanjkanja prostega časa. Tega, ki ga imamo na voljo, hočemo najbolje izkoristiti. Velikokrat si želimo obiskati kraje, ki ponujajo lepo naravo, sprostitve, ipd.

Slovenija ponuja veliko zanimivih lokacij, ki jih marsikdo ne pozna in da bi prišel do njih, mora porabiti kar nekaj časa, da poišče takšne, ki se mu zdijo primerne in zanimive. Pri tem mora obiskati veliko spletnih strani, prebrati par forumov in povprašati prijatelje, kje so že bili in kako jim je bilo všeč.

Kar nekaj spletnih strani ponuja predloge za izlete, toda vse te strani so usmerjene lokalno ali pa se osredotočajo na specifične vrste lokacij. Na primer, spletna stran [www.zaplana.net](http://www.zaplana.net) ponuja množico izletov, toda ti izleti so predvsem osredotočeni na slovenske vrhove. Edina lokacija, kjer so izbrane lokacije in različne zvrsti po celi Sloveniji je spletna stran [www.slovenia.info](http://www.slovenia.info), kjer imamo po nekih smiselnih kategorijah razporejene lokacije. Toda to je samo informativna stran in nam ne omogoča, da bi si lahko beležili, kje vse smo že bili.

V tej diplomski nalogi je predstavljena rešitev tega problema, tako da lahko uporabniki brskajo med vnesenimi lokacijami ter iščejo predloge za izlete na podlagi podanih kriterijev (vrsta izleta, aktivnost, oddaljenost). Sistem pa registriranim uporabnikom tudi nudi priporočila na podlagi njihovih profilov in profilov podobnih uporabnikov, kakor to počnejo drugi priporočilni sistemi.

Priporočilni sistemi (*recommendation systems*, RS) [1, 4] so programska orodja in tehnike, ki priskrbijo predloge za predmete (*items*), ki bi za uporabnika utegnili biti zanimivi. Predlogi so odvisni od mnogih odločitvenih procesov, kot so, kakšen izdelek kupiti, kakšno vrsto glasbe poslušati ali katere spletne novice prebrati. V našem primeru se odločamo za izletniške lokacije oz. njihove aktivnosti, ki nas zanimajo.

Predmet je splošni tîrmin uporabljen za označitev, kaj sistem predlaga uporabniku. RS se ponavadi osredotočijo na specifični tip predmeta. V našem primeru so to lokacije, ki jih predlagamo uporabniku.

Za izvajanje svoje temeljne funkcije, opredelitev uporabnih predmetov za uporabnika, mora RS predvideti, če je predmet sploh vreden predloga. Da bi to naredil, mora biti sistem sposoben predvideti, kakšna je *koristnost* nekega predmeta ali vsaj primerjati koristnosti nekaterih predmetov in se glede na te primerjave odločiti, katere predmete naj predlaga.

V večini primerov priporočilni sistemi delujejo na podlagi sodelovalnega filtriranja (*collaborative filtering*), kar pomeni, da sistem primerja uporabnika z drugimi uporabniki glede na njihov okus, in mu tako priporoči predmete, ki bi bili zanj zanimivi. V našem primeru sistem primerja uporabnike, glede na ocene aktivnosti, ki so jim skupne.

## 2 PODATKOVNI MODEL

V Sloveniji obstaja 12 statističnih regij (Slika 1), v katerih imamo na voljo številne lokacije primerne za izlete. Ker lahko na vseh teh lokacijah počnemo več različnih stvari (ogled narave, pohod, ...), je smiselno da se lokacija razdeli na dva dela, in sicer na samo lokacijo in na aktivnosti, ki so na tej lokaciji. To je potrebno tudi zaradi različnega časovnega obdobja aktivnosti, ki jo lahko opravljamo. Primer za to je gora Golica, kjer lahko občudujemo planine in pokrajino skoraj skozi celotno leto, če pa si hočemo pogledati narcise, lahko to storimo le v mesecu maju.



Slika 1: Statistične regije v Sloveniji (vir: [http://sl.wikipedia.org/wiki/Statistične\\_regije\\_Slovenije](http://sl.wikipedia.org/wiki/Statistične_regije_Slovenije))

Vsaka izmed teh aktivnosti ima tudi različne lastnosti. Za pohodništvo je tako pomembno, da se ve, kolikšno pot se opravi, višina, ki se doseže ter višinska razlika. Pri ogledu znamenitosti pa moramo vedeti, če moramo plačati vstopnino, ali nam je pri tem na voljo vodič, ipd. Iz vseh teh informacij smo sestavili podatkovni model, pri čemer se je tudi upoštevalo izletnika, ki to lokacijo obišče in si pri tem lahko zabeleži, kje je bil. Tako je nastalo 14 tabel (Slika 2), ki se povezujejo med sabo.

### 2.1 Relacijski model

Tabela lokacija (*Location*) (Slika 2) vsebuje zgolj zapis o imenu lokacije in v katero regijo spada. Služi nam za predstavitev neke aktivnosti, da vemo, na kateri lokaciji se nahaja.

Tabela aktivnost (*Activity*) vsebuje različne aktivnosti, kot so: Zimski športi, Vodni športi, Pohodništvo, Kolesarjenje, Ostali športi, Ogled znamenitosti, Ogled narave, Kulinarika &

vino, Obisk prireditve Zdravilišče & Wellness, Drugo. Vsebuje ime te aktivnosti ter vrstni red aktivnosti, ki se prikažejo uporabniku.

Osrednji del podatkovnega modela tako predstavlja povezava med tabelama lokacija in aktivnost, ki jo predstavimo s tabelo lokacijska aktivnost (*LocationActivity*). Ta vsebuje podatke, na kateri lokaciji se nahaja določena aktivnost ter njen kratek opis. Opis je potreben zaradi več istoimenskih aktivnostih na določeni lokaciji, ter tudi zato, da uporabnik iz opisa izve več podrobnosti o tej aktivnosti.

Tabela lokacijska lastnost (*LAttribut*) vsebuje podatke o lastnostih, ki veljajo za aktivnosti. Vsebuje ime lastnosti (npr. zahtevnost), tip lastnosti (številska vrednost, datum, opisne vrednosti ter da ali ne odgovori) ter vrednosti tipa. Vrednosti tipa se vpišejo v primeru, da je tip lastnosti opisne vrednosti. Tukaj naštejemo te vrednosti, ločene s podpičjem (npr. Zahtevno, manj zahtevno, nezahtevno).

Vrednosti atributov so zbrane v tabeli lastnost lokacije (*LocationAttribut*). V tej tabeli so zbrani vsi atributi, ki so pomembni za določeno aktivnost na lokaciji ter njihove vrednosti. Te vrednosti se določijo pri vnosu nove aktivnosti na določeni lokaciji.

Ker moramo vedeti, katere lastnosti pripadajo določeni aktivnosti, sta tabeli aktivnost ter lokacijska lastnost povezani z vmesno tabelo lastnost aktivnosti (*ActivityAttribut*). Tukaj so zbrane vse lastnosti, ki so povezane z določeno aktivnostjo. Obstaja pa še polje obvezno (*mandatory*), ki nam pove, če je vrednost atributa pri dodajanju nove aktivnosti potrebno vpisati. To polje je zaradi aktivnosti drugo, kjer uporabnik sam izbere, katere lastnosti pripadajo neki aktivnosti.

Kot že omenjeno ima vsaka aktivnost svoje časovno obdobje. To časovno obdobje predstavlja tabela sezona (*Season*). Sezona je predstavljena z meseci v letu. Da vemo kateri meseci veljajo za določeno aktivnost, pa obstaja tabela sezona aktivnosti (*ActivitySeason*), ki povezuje tabelo aktivnost s tabelo sezona.

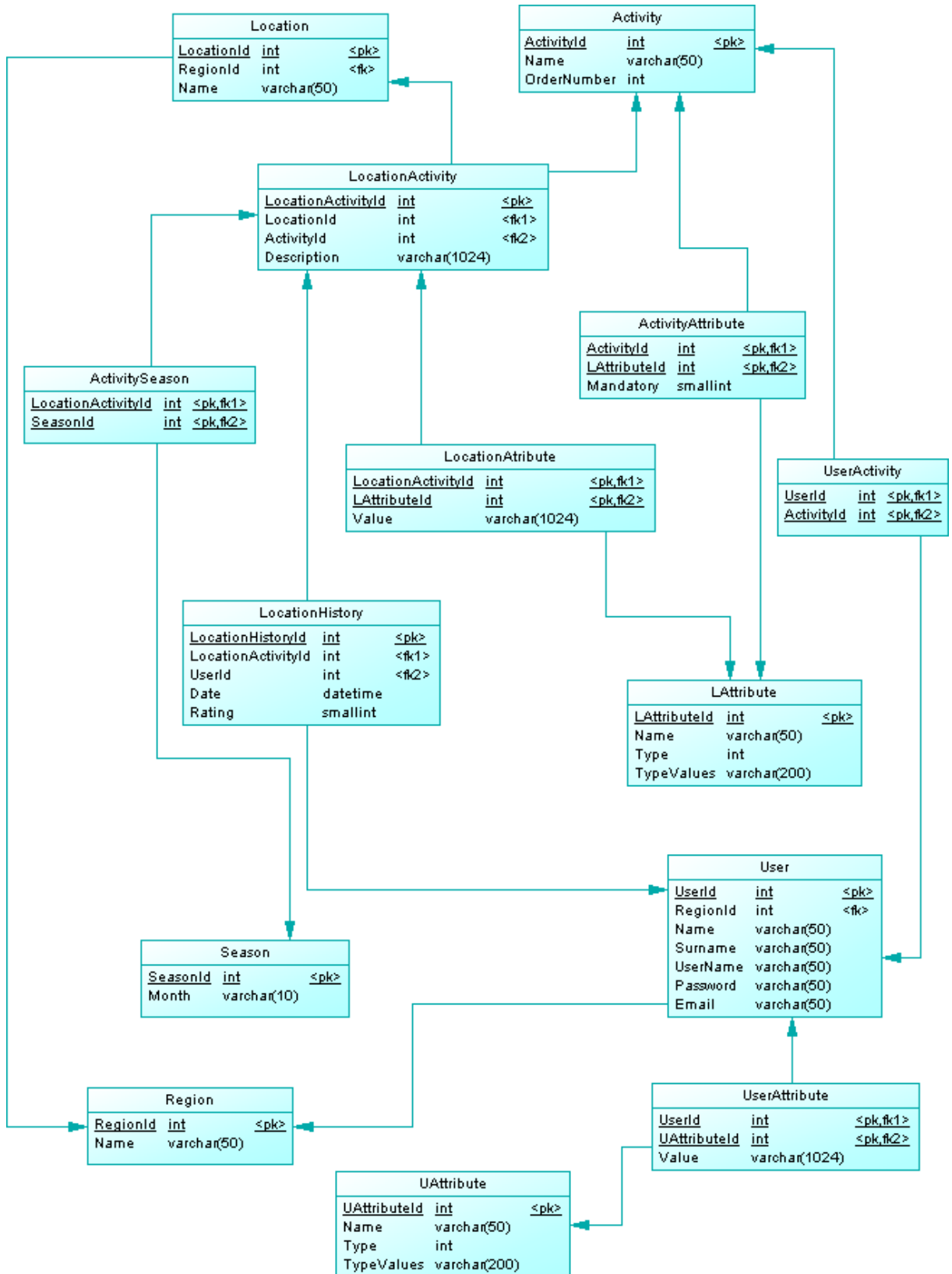
Tabela uporabnik (*User*) vsebuje osnovne informacije o registriranemu uporabniku: ime, priimek, elektronski naslov ter uporabniško ime in zgoščeno vrednost (*hash*) gesla, ki se izračuna po zgoščevalni funkciji SHA1. Slednja podatka sta namenjena prijavi uporabnika v sistem.

Tabela lastnost uporabnika (*UserAttributes*) predstavlja lastnosti ter njihove vrednosti, ki jih je vnesel uporabnik. Povezana je s tabelo uporabniška lastnost (*UAttribut*), ki je namenjena podatkom o vseh lastnostih, ki jih lahko uporabnik izbere.

Vsak uporabnik lahko beleži zgodovino lokacijskih aktivnosti, ki jih je že obiskal. Ti podatki so zapisani v tabeli zgodovina lokacij (*LocationHistory*), kjer se vodi evidenca vseh dosedaj obiskanih aktivnostih na neki lokaciji za vsakega uporabnika. Vsebuje podatke o datumu obiska in pa oceno, ki jo je izbral uporabnik.

Ker uporabnik lahko določi, katere aktivnosti ga še posebej zanimajo, je ustvarjena povezava med tabelama uporabnik in aktivnost. Povezovalna tabela se imenuje aktivnosti uporabnika (*UserActivity*).

Za vsako lokacijo je potrebno vedeti, kje se nahaja. Tabela regija (*Region*) predstavlja podatek, v katero regijo spada neka lokacija, oz. v kateri regiji živi uporabnik, ki mu predlagamo neko lokacijo. Vsebuje zgolj ime statistične regije.



Slika 2: Podatkovni model

## 2.2 Zapisi v podatkovni bazi

Za potrebe prikazovanja v naprej določenih podatkov (regije, aktivnosti, ..), so ti že vnaprej vneseni v podatkovno bazo. Teh podatkov ni mogoče spreminjati na sami aplikaciji. V primeru da bi se izkazalo, da je te podatke potrebno spremeniti ali pa jim dodati nov podatek je tako potrebno to storiti na strani baze.

Podatki, ki so že vnaprej vneseni so zapisani v tabelah (Slika 2):

- *aktivnost*
- *lokacijska lastnost*
- *lastnost aktivnosti*
- *uporabniška lastnost*
- *sezona*
- *regija*

V tabeli aktivnost so vneseni podatki o aktivnostih, ki se lahko uporabijo za sestavljanje aktivnosti na določeni lokaciji. Aktivnosti, ki so vnesene v bazo so zimski športi, vodni športi, pohodništvo, kolesarjenje, ostali športi, ogled znamenitosti, ogled narave, kulinarika in vino, obisk prireditve, zdravilišče in wellness ter drugo. Pri tem je v polju zaporedna številka (*OrderNumber*) določeno, v kakšnem vrstnem redu se prikažejo pri izbiri. Polje za določanje vrstnega reda se je dodalo v bazo, ker smo želeli vedno imeti v seznamu aktivnost drugo na zadnjem mestu. Ker so se podatki sproti dopolnjevali bi bilo potrebno to aktivnost pri vsakem vnosu izbrisati in ponovno vnesti, tako pa se samo spremeni vrstni red, da je vedno na zadnjem mestu prikazana aktivnost drugo.

V tabeli lokacijska lastnost so vneseni podatki o vseh lastnostih, ki se lahko pojavijo na lokacijski aktivnosti. Tukaj so zbrane predvsem lastnosti kot so višina[m], dolžina[m], zaščiten?, potrebna dobra oprema?, zahtevnost, vrsta športa, prireditve, zdravilišča, ipd. Pri podatkih, kot sta na primer vrsta športa in zahtevnost so v polju vrednosti tipa (*TypeValues*) zapisane opisne vrednosti teh lastnosti. Tako so v primeru vrsta športa v tem polju našteje vse vrste športa, ki so ločene s podpičjem (golf;jahanje;ribolov, ...), v primeru zahtevnosti pa so vpisane vrednosti nezahtevno, manj zahtevno, zahtevno ter zelo zahtevno. Kjer so pa drugačni tipi tega polja, pa je polje prazno oz. nima vrednosti (*null*), ker pri številskih vrednostih in datumu lahko uporabnik vpiše karkoli. Razen pri odgovorih z da ali ne(primer je lastnost, če je vstopnina), kjer sistem ve, kaj mora prikazati uporabniku.

V tabeli lastnost aktivnosti so vneseni podatki o povezavah med aktivnostjo in njenimi lastnostmi. Ti podatki so povezani zato, da se pri vnosu nove aktivnosti lahko izpišejo lastnosti, ki so pomembne za to aktivnost. Tukaj so našteje vrednosti o identifikacijski številki aktivnosti in o identifikacijski številki lastnosti. Za potrebe vnosa podatkov se je izkazalo, da

je v primeru aktivnosti drugo potrebno upoštevati samo podatke, ki jih vnese uporabnik. Zato je v polju obvezno (*Mandatory*) zapisano, če je ta podatek obvezno potrebno vpisati ali ne.

V tabeli uporabniška lastnost so vneseni podatki, ki jih lahko uporabnik vnese na svojem profilu. Tukaj so zapisani podatki, kot so datum rojstva, spol, izobrazba, ali imate avto ter ali ste pripravljeni plačati vstopnino. Podobno kot v tabeli lokacijska lastnost so tudi tukaj za opisne vrednosti podane njihove vrednosti ločene s podpičjem. Za primer spola sta tako zapisani vrednosti M in Ž.

V tabeli sezona so vneseni podatki o vseh mesecih v letu (januar, februar, marec, ...). Pri teh podatkih številko meseca predstavlja kar njegova identifikacijska številka.

V tabeli regija pa so vnesena vsa imena o statističnih podatkih v Republiki Sloveniji. Katere regije so vsebovane prikazuje Slika 1.

### 2.3 Postopek za rangiranje lokacij

Eden od glavnih pristopov pri sodelovalnem filtriranju je uporaba klasifikatorja k najbližjih sosedov (kNN - k nearest neighbors) [2, 3]. Ta klasifikacijska metoda, tako kot večina klasifikatorjev in povezovalnih tehnik, je zelo odvisna od opredelitve ustrezne podobnosti ali razdalje med predmeti. Osnovni enačbi za izračun razdalje sta Evklidska (*Euclidian distance*) (enačba 1) in Manhattanska razdalja (*Manhattan distance*) (enačba 2):

$$d(x, y) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2} \quad (1)$$

$$d(x, y) = \sqrt{\sum_{k=1}^n |x_k - y_k|}, \quad (2)$$

kjer sta  $x$  in  $y$  primerjana predmeta,  $x_k$  in  $y_k$  pa predstavljajo posamezne lastnosti predmetov [1, 2, 3, 4].

V našem primeru vrednost  $x_k$  predstavlja oceno uporabnika, za katerega želimo pridobiti priporočila, vrednost  $y_k$  pa predstavlja oceno uporabnika do katerega računamo razdaljo. Ocena je zapisana v tabeli zgodovina lokacij, kjer se za vsako lokacijsko aktivnost ustvari nov zapis, ko jo uporabnik vnese.

Primer izračuna razdalje bomo prikazali na resničnih podatkih izvzetih iz naše baze. Iz baze sta vzeta dva uporabnika, pri čemer smo jih zaradi varstva osebnih podatkov preimenovali v izletnik  $x$  in izletnik  $y$ . Tabela 1 prikazuje vse obiske teh dveh uporabnikov in njune ocene za posamezno lokacijsko aktivnost. Če ne obstaja ocena izletnika za lokacijsko aktivnost, je na

tem mestu zapisan znak »?«. Ocene imajo celoštevilске vrednosti med 1 in 5, pri čemer ocena 1 pomeni, da uporabniku ni bilo všeč na tej lokacijski aktivnosti, ocena 5 pa pomeni, da mu je bilo zelo všeč.

**Tabela 1: Prikaz ocen dveh uporabnikov izvzete iz baze podatkov**

lokacijska aktivnost	ocena izletnika x	ocena izletnika y
4	4	?
5	4	4
6	5	5
7	3	?
9	4	4
10	?	5
14	?	4
15	?	4
17	5	?
18	5	4
19	?	4
21	4	4
23	?	4
29	3	?
30	1	?
31	5	?
32	1	?

Da bi lahko izračunali razdaljo med uporabnikoma lahko vzamemo samo tiste lokacijske aktivnosti, ki ta jih oba izletnika ocenila. Te lokacijske aktivnosti so v našem primeru 5, 6, 9, 18 ter 21. Iz teh podatkov lahko izračunamo razdaljo med tema dvema izletnikoma. Za izračun razdalje smo uporabili enačbo za izračun evklidske razdalje (enačba 1).

Če izračunamo razdaljo za ta dva izletnika dobimo rezultat:

$$\begin{aligned}
 d(x, y) &= \sqrt{(4 - 4)^2 + (5 - 5)^2 + (4 - 4)^2 + (5 - 4)^2 + (4 - 4)^2} \\
 &= \sqrt{0 + 0 + 0 + 1 + 0} = 1
 \end{aligned}$$

Nato še izračunamo prilagoditev tega rezultata po enačbi:

$$\frac{1}{1 + d(x, y)}, \quad (3)$$

da dobimo rezultate med 0 in 1, pri čemer število 1 pove, da se uporabnika popolnoma ujemata, 0 pa da se uporabnika sploh ne ujemata. Število 1 v spodnjem delu prištejemo, da se izognemo deljenju s številom 0.

Če izračunamo po tej enačbi dobimo rezultat:

$$\frac{1}{1 + 1} = 0.5$$

Iz tega rezultata lahko razberemo, da sta si uporabnika po željah dokaj podobnega mnenja in da jima lahko priporočamo lokacijske aktivnosti, ki jih drugi še ni obiskal.

### 3 SPLETNA APLIKACIJA

Glavni del te diplomske naloge predstavlja spletna aplikacija, ki uporabnikom priporoča lokacije za izlete. Zgrajena je s tehnologijo ASP.NET WebForms v povezavi s programskim jezikom C#.NET verzije 3.5. Povezuje se na podatkovno bazo, ki teče na strežniku Microsoft SQL 2008. Pri izdelavi spletne aplikacije se je uporabilo programsko orodje Visual Studio 2008 ter za izdelavo in dostop do podatkovne baze Microsoft SQL Server Management Studio 2008. V spletni aplikaciji so se uporabili tudi gradniki iz paketa ASP.NET AJAX Control Toolkit. Iz tega paketa sta se uporabili dve razširitvi in sicer Calendar ter FilteredTextBox. Razširitev Calendar smo uporabili pri poljih, kjer je potrebno vnesti datum. Ta razširitev nam omogoča povezavo na gradnik TextBox in s tem omogoča lažjo izbiro datuma s strani uporabnika. Razširitev FilteredTextBox pa smo uporabili pri poljih, ki zahtevajo vnos številčnih vrednosti. Ta razširitev se tako kot prejšnja lahko poveže na katerikoli gradnik TextBox in nam omogoča omejitev nabora znakov. To razširitev smo uporabili na poljih za vnašanje lastnosti določene aktivnosti, kjer od uporabnika zahtevamo vnos le številčnih vrednosti (primer je recimo višina, dolžina, ...). S tem smo pridobili, da se s strani uporabnika že vpišejo pravilne vrednosti.

Spletna aplikacija je zgrajena iz dveh delov in sicer iz same aplikacije ter v ozadju iz logike. Vsa logika sistema je shranjena kot posebna knjižnica, katero vključuje spletna aplikacija in kliče njene metode za shranitev in pridobivanje podatkov o lokacijah, aktivnostih, ipd.. Logika nam tudi poda seznam vseh priporočil, na strani aplikacije moramo samo podati podatke o uporabniku (če je prijavljen) in o njegovih željah. Ta logika poskrbi za celotno pridobivanje in zapis podatkov v bazo. Za pridobivanje podatkov iz podatkovne baze in shranjevanje v njo so se uporabili shranjeni podprogrami (*stored procedures*). Tako ni pri vsaki spremembi potrebno iskati v kodi določenega vprašanja SQL, ampak se popravi podprogram brez da je potrebno prevajati celotno spletno aplikacijo.

Aplikacija omogoča uporabnikom iskanje lokacij, ki so vneseni v sistem, kakor tudi iskanje lokacij po določenih kriterijih. Iskanje samih lokacij pride v poštev takrat, ko je uporabnik že obiskal določeno lokacijo in si jo želi zabeležiti v sistem in ji pri tem tudi podati oceno, da bodo lahko ostali uporabniki vedeli, če se to lokacijo splača obiskati. Iskanje lokacij po kriterijih pa pride v poštev, ko uporabnik ne ve, kam bi se lahko v določenem trenutku odpravil na izlet.

Spletno aplikacijo lahko pregledujejo tako neregistrirani (anonimni) uporabniki, kot tudi registrirani. Sestavljena je iz petih strani, tri izmed njih so skupne obema vrstama uporabnikov.

Strani, ki so skupne obema vrstama uporabnikov :

- *Seznam lokacij*

Na tej strani je zbran seznam vseh lokacij, če imamo pravice, pa lahko dodajamo nove lokacije.

- *Urejanje lokacij*

Na tej strani se urejajo podatki vezani na lokacijo oz., če ne obstaja pravica za urejanje, se samo pregledujejo podatki o tej lokaciji in njenih aktivnostih.

- *Priporočila lokacij*

Na tej strani sistem uporabniku priporoča lokacije.

Strani, ki so vezane na registriranega uporabnika:

- *Moj profil*

Na tej strani so zbrane vse informacije, ki se nanašajo na uporabnikov profil.

- *Moje lokacije*

Na tej strani lahko uporabnik vidi, katere lokacije je že obiskal.

Podatki o lokacijah in njihovih aktivnostih se nahajajo v podatkovni bazi. Ker so ti podatki ključnega pomena pri sestavljanju priporočila uporabnikom, jih je potrebno zbrati in vnesti. To se stori preko administrativnega dela aplikacije. To so iste strani, kot so opisane zgoraj, s to razliko, da se pri seznamu lokacij prikaže obrazec za vnos nove lokacije, na strani za urejanje lokacij pa se prikaže gumb za dodajanje nove aktivnosti.

### **3.1 Administrativni del**

Da bi zagotovili resnične podatke za potrebe te diplomske naloge imajo vsi registrirani uporabniki pravico dodajanja novih lokacij ter njihovih aktivnostih, saj smo s tem omogočili uporabnikom, ki so obiskali neko lokacijo in je sistem še ne pozna, da jo lahko dodajo in jo označijo v sistemu. Pri tem lahko nastane problem, če bi uporabniki vnašali podatke, ki niso točni in s tem onemogočili pravilno delovanje sistema. Zato bi bilo na tem mestu smiselno

vpeljati uporabniške pravice, na podlagi katerih bi sistem preveril, kaj uporabnik lahko počne in kaj ne.

Če ima uporabnik administrativno pravico, se mu na strani, kjer se nahaja seznam lokacij, prikaže obrazec za vnos nove lokacije (Slika 3). Tukaj se enostavno vnese ime lokacije ter izbere regija v katero spada. Po pritisku na gumb Shrani se podatki shranijo v tabelo lokacija. Če pri shranjevanju pride do napake, je uporabnik o tem obveščen z besedilom nad obrazcem. Prav tako je uporabnik obveščen, če ni vnesel vseh zahtevanih podatkov ali pa če je vnesel ime lokacije, ki že obstaja. Po uspešnem vnosu nove lokacije nam shranjeni podprogram, ki shrani lokacijo posreduje identifikacijsko številko lokacije. Sistem preusmeri uporabnika na stran z urejanjem te lokacije. Pri tem se v spletnem naslovu kot parameter poda identifikacijska številka lokacije. Za primer, da shranimo novo lokacijo in se v bazo zapiše pod zaporedno številko 39, se v spletni naslov zapiše »EditLocation.aspx?locationId=39«.

To ime za lokacijo že obstaja.

Ime lokacije\*:

Regija\*:

**Slika 3: Primer vnosa lokacije, ki že obstaja v sistemu**

Na strani z urejanjem lokacije ni mogoče več spreminjati imena in regije, na katero spada. V prihodnosti bi bilo smiselno uvesti, da ima glavni administrator sistema tudi to pravico. Sedaj je potreben poseg v bazo podatkov, kjer pa se lahko hitro zgodi, da oseba, ki to vnaša po pomoti spremeni ali izbriše kak drug podatek.

Na strani z urejanjem lokacije ima tako uporabnik, možnost vnesti novo aktivnost, ki se izvaja na tej lokaciji. To stori tako, da pritisne gumb »Dodaj aktivnost« in odpre se obrazec (Slika 4), kjer se v spustnem meniju izbere aktivnost, katero želi dodati.

Da bi uporabnikom zagotovili čim več informacij o določeni aktivnosti, se po izbiri le te prikaže obrazec z lastnostmi (Slika 4), ki jih vsebuje ta aktivnost. Vsaka aktivnost vsebuje drugačno vrsto lastnosti, saj recimo, če se odpravimo na ogled znamenitost, nas ne zanima na kakšni višini se nahaja. Pri aktivnosti je potrebno dodati še njen opis, saj iz opisa lahko uporabnik sklepa, če je aktivnost res primerna zanj ali ne. Za vse aktivnosti pa velja, da se lahko obišejejo v določenem obdobju. Zato je potrebno še pri vseh izbrati mesece, v katerih se lahko obišejejo, saj se na primer lahko odpravimo smučat le v zimskem času, ko je sneg. Pri vseh aktivnostih je vnos vseh lastnosti obvezen, razen pri aktivnosti drugo, kjer uporabnik sam izbere, katere lastnosti so značilne za to aktivnost. Po pritisku na gumb »Dodaj« se v

tabelo lokacijska aktivnost shranijo podatki, na kateri lokaciji se nahaja ta aktivnost, kakšna je vrsta aktivnosti ter opis. Identifikacijska številka, ki določa lokacijo, za katero vnašamo aktivnost se pridobi iz parametra v spletnem naslovu urejanja lokacije. V tabelo lokacijska lastnost pa se zapišejo vse lastnosti in njihove vrednosti.

Ostali športi ▼      Dodaj    Prekliči

Vsa polja označena z zvezdico (\*) so obvezna

Zahtevano predhodno znanje?\*      DA ▼

Potrebna dobra oprema?\*      NE ▼

Je vstopnina?\*      NE ▼

Zahtevnost\*      Manj Zahtevno ▼

Vrsta športa\*      Ribolov ▼

Opis\*:

Mesec\*:

Izberi vse

Januar     Maj       September

Februar     Junij     Oktober

Marec     Julij     November

April     Avgust     December

**Slika 4: Primer dodajanja nove aktivnosti na lokacijo**

Če se uporabnik v določenem trenutku premisli in noče dodati nove aktivnosti na lokacijo ima na voljo gumb »Prekliči«, ki pobriše vse vnesene podatke, ter zapre obrazec za dodajanje. V tem primeru se zopet prikaže gumb za dodajanje aktivnosti.

Pri vnosu nove aktivnosti se lahko izbira med naslednjimi aktivnostmi:

- *zimski športi*
- *vodni športi*
- *ostali športi*
- *pohodništvo*
- *kolesarjenje*
- *ogled znamenitosti*
- *ogled narave*
- *kulinarika in vino*
- *obisk prireditve*
- *zdravilišče & wellness*
- *drugo*

Te aktivnosti so predhodno vnesene v tabeli aktivnost. Če bi se pojavila potreba dodajanja nove aktivnosti bi tako bilo še potrebno narediti stran, kjer bi lahko administratorji strani dodajali nove vrednosti.

Pri zimskih športih je pomembno, da se določi vrsta športa, ki se odvija na določeni lokaciji. Tukaj pridejo v poštev športi, kot so smučanje, deskanje na snegu, tek na smučeh, drsanje, sankanje, ipd. Če se je na določeni lokaciji možno udeleževati v večjih športih, je potrebno vsak šport dodati kot posebno aktivnost na to lokacijo. Izpolniti je potrebno še druge kriterije ki so značilni za zimske športe, saj tako lahko uporabnik vidi, če je pripravljen za tako obliko športa. V primeru, da neka vrsta športa zahteva predhodno znanje in se uporabnik s tem športom še ni srečal, potem se ne bo odločil za obisk te lokacije.

Pri vodnih športih je podobno kot pri zimskih športih, le da so tukaj zbrani športi, ki se odvijajo na vodi. Primeri teh športov so rafting, hydrospeed, kajakaštvo, soteskanje, plavanje, ipd.

Pod ostale športe spadajo predvsem športi, ki jih nismo mogli uvrstiti med nobene prej našteje. Tukaj so zbrani dvoranski športi, golf, jahanje, ipd. Kriteriji so podobni kot pri zimskih in vodnih športih.

Če v seznamu zimskih, vodnih ali ostalih športov ne najdemo primerne vrste športa, lahko pri vseh uporabimo vrednost drugo in nato v opisu aktivnosti napišemo kakšna je vrsta tega športa.

Pri pohodništvu je pomembno, da sistem pozna vse lastnosti, ki so pomembne za določeno lokacijo, saj le tako uporabnik ve, če je dovolj pripravljen, da se odpravi na takšno pot. Tukaj je zelo pomembno, da uporabnik, ki se odpravlja na določen hrib ali goro, ve, kolikšna je njegova višina, kako dolgo pot bo prehodil, kakšna je višinska razlika, ki jo bo opravil, ipd.

Pri kolesarjenju mora uporabnik, podobno kot pri pohodništvu, vedeti, kaj ga čaka, če se odpravi na določeno pot. Tukaj pridejo v poštev predvsem podatki, kolikšno pot bo opravil s kolesom in, če se le-ta vzpenja na kakšen hrib, tudi kakšna bo opravljena višinska razlika. Pomembni podatki so tudi, ali so poti dobro označene, da se ne bi izgubili na poti in ali je na voljo zemljevid, ki bi nam bil v pomoč pri doseganju cilja.

Pri ogledu znamenitosti je pomembno, da vemo, kakšna je vrsta te znamenitosti, saj tako lahko ocenimo, ali bi nas takšna vrsta zanimala. Tukaj je tudi pomembno ali je ogled znamenitosti primeren za učno uro in s tem pridobimo informacijo, če lahko odpeljemo zraven otroke, da bi se kaj naučili.

Pri ogledu narave je pomemben podatek, na kakšni dolžini se razprostira narava, ki si jo lahko ogledamo in ali se lahko na to mesto pripeljemo za avtomobilom ali ne. Ker je potrebno na določenem območju upoštevati določena pravila, je za to namenjena lastnost zaščitenost, s katero povemo, če ta aktivnost leži v zaščitenem območju. Primer je Triglavski narodni park, kjer je s strani človeka poseg v naravo zelo omejen.

Pri kulinariki in vinu nas zanimajo predvsem informacije, za kakšno vrsto gre, ali je to restavracija, gostilna, vinska klet, ipd. Pomemben podatek je tudi, ali vsebujejo ponudbo, ki je značilna za določeno območje.

Pri obisku prireditve so predvsem mišljene prireditve, ki se ponavljajo v določenem obdobju vsako leto, saj sistem ne predvideva vnosa točnega datuma prireditve.

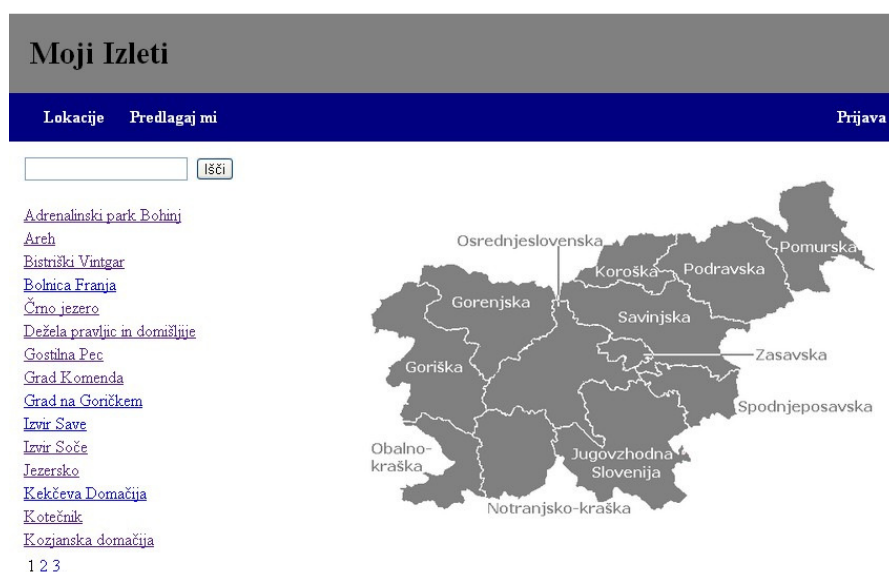
Pri zdravilišču nas predvsem zanima, kakšna je vrsta zdravilišča, saj s tem lahko predvidevamo, kakšno ponudbo lahko pričakujemo.

Če pri vnosu ne najdemo prave aktivnosti, ki bi opisala dogajanje na tej lokaciji, je za to namenjena aktivnost drugo, kjer si sami izberemo katere lastnosti so značilni za to lokacijsko aktivnost. To storimo tako, da izpolnimo polja, ki so značilna za to lokacijo. Sistem nato preveri, katere vrednosti smo vnesli in tako samo te vrednosti shrani v bazo podatkov.

## 3.2 Uporabniški del

Uporabniški del aplikacije je namenjen uporabnikom, ki si želijo priporočila izletniških lokacij. Na tem mestu lahko uporabnik pregleduje vse lokacije ali pa mu sistem priporoča, katere lokacije bi bile zanj zanimive.

Ko se uporabniku odpre prva stran, se mu prikaže seznam vseh že vnesenih lokacij (Slika 5). Da uporabniku ni potrebno brskati po vseh straneh z lokacijami ima na voljo iskalnik, preko katerega išče po lokacijah in njegovih aktivnostih. Ta del pride zelo prav v primeru, da uporabnik išče določeno lokacijo in bi o njej rad izvedel več podrobnosti.



Slika 5: Vstopna stran s seznamom lokacij

Po kliku na izbrano lokacijo uporabnika preusmerimo na stran s podrobnostmi (Slika 6), ki je enaka kot pri urejanju aktivnosti, le da so tukaj izpisani podatki o lokaciji ter našteje aktivnosti, ki se nahajajo na tej lokaciji. Vnos nove aktivnosti tukaj ni mogoč. Za vsako aktivnost so na voljo informacije, na katero vrsto aktivnosti se nanaša, njen opis, sezona, v kateri lahko določeno lokacijsko aktivnost obiščemo ter vse lastnosti z njihovimi vrednostmi.

#### Lokacija

Ime lokacije\*:   
 Regija\*: Gorenjska

Aktivnost	Opis	Sezona	Lastnosti
Zimski športi	Smučanje na Krvavcu	Januar , Februar , Marec , November , December	Zahtevano predhodno znanje?: NE Potrebna dobra oprema?: NE Zahtevnost: Manj Zahtevno Vrsta zimskega športa: Smučanje
Pohodništvo	Peš pot od zgornje postaje žičnice do Velikega Zvoha.	April , Maj , Junij , Julij , Avgust , September , Oktober	Dolžina[m]: 2000 Višina[m]: 1970 Višinska razlika[m]: 500 Senčna pot?: NE Dostop z avtom?: DA Potrebna planinska oprema?: DA Zahtevnost: Manj Zahtevno
Kolesarjenje	Cestna tura od Ambroža pod Krvavcem do zgornje postaje žičnice	April , Maj , Junij , Julij , Avgust , September , Oktober	Dolžina[m]: 5000 Višinska razlika[m]: 400 Dobro označene kolesarske ture in poti?: DA Na voljo zemljevidi kolesarskih tur in poti?: NE Specializirana nastanitev za kolesarje: NE Možen najem koles?: NE Na voljo popravilo koles?: NE Zahtevnost: Zahtevno

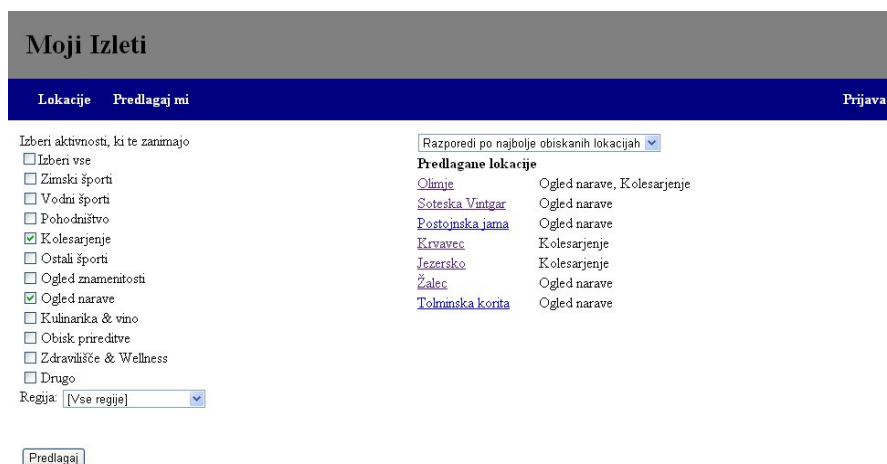
**Slika 6: Prikaz podrobnosti o določeni lokaciji**

Glede na vrsto vseh aktivnosti, ki so na tej lokaciji, lahko uporabnik oceni, če ta lokacija ustreza njegovim željam. Če to ni dovolj si lahko prebere opis vsake aktivnosti in pa njene lastnosti. Lastnosti mu povedo, če je dovolj pripravljen za določeno aktivnost. Na primer, če želi na izlet v gore, mu zelo pomagajo informacije, kolikšna je njena višina, kako dolgo pot bo opravil ter kakšna je pri tem višinska razlika.

Če uporabnik ne ve katero lokacijo naj izbere, se lahko odpravi na stran s priporočili lokacij (Slika 7). Tukaj mu sistem predlaga lokacije, ki bi bile zanj v tem trenutku zanimive. Na tem mestu uporabnik izbere, katere aktivnosti ga zanimajo in po želji v katero regijo želi odpotovati. Poiščejo se lokacije, ki vsebujejo te aktivnosti in se nahajajo v regiji, ki jo je uporabnik vnesel. Če uporabnik ne izbere regije, se izdelava priporočilo glede na lokacije iz vseh regij. Prav tako lahko uporabnik izbere, da se izbere po vseh aktivnostih, ker trenutno ne ve, kaj ga najbolj privlači.

Glede na to, da vseh aktivnosti ne moremo obiskati v določenem trenutku, se iz seznama izločijo vse tiste, ki niso značilne za to obdobje (za trenutni mesec). Predlagane lokacije se izpišejo glede na to, koliko aktivnosti vsebujejo, ki jih je izbral uporabnik. Le tako mu lahko ponudimo čim več aktivnosti na eni lokaciji, ki bi ga utegnile zanimati. Lokacije so nato

razporejene glede na število, kolikokrat je bila lokacija obiskana s strani prijavljenih uporabnikov. Če uporabnik želi, lahko naknadno lokacije razporedi glede na oceno, ki so jo ti uporabniki izbrali za aktivnosti, potem ko so jih obiskali. Uporabniku se prikaže 10 najvišje uvrščenih lokacij (če jih obstaja toliko), nato pa lahko brska po naslednjih straneh, kjer se mu prav tako prikaže toliko lokacij na vsaki strani.



**Slika 7: Primer prikaza priporočil uporabniku razporejene glede na število obiskov**

Pri najbolj ocenjenih lokacijah se zberejo podatki o ocenah lokacij, ki so jih prijavljeni uporabniki obiskali in vrne najvišje ocenjeno, ki vsebuje čim več izbranih aktivnosti. Pri tem se upošteva povprečna ocena vseh aktivnosti na določeni lokaciji. Pri najbolj obiskanih lokacijah pa se zberejo podatki, koliko prijavljenih uporabnikov je neko aktivnost na neki lokaciji obiskalo in glede na to število se vrnejo lokacije, ki vsebuje največ izbranih aktivnosti glede na število obiska. V seznamu predlaganih lokacij se izpiše ime lokacije ter imena vseh aktivnosti, ki jih je uporabnik vnesel. Na tem mestu lahko podobno kot v seznamu vseh lokacij klikne na določeno lokacijo, ki mu je še posebej zanimiva in sistem ga usmeri na stran s podrobnostmi o tej lokaciji. Tako lahko pridobi informacije o vseh aktivnostih v prejšnjem seznamu in tudi o aktivnostih, ki se poleg teh nahajajo na tej lokaciji.

Pri zgoraj opisanem postopku izdelave priporočil uporabniku ne moremo zagotoviti, da mu ne predlagamo lokacije, ki jo je že obiskal, ker o njem in njegovih navadah ne vemo ničesar. Če želi uporabnik voditi evidenco že obiskanih lokacij in glede na to pridobiti predloge za naslednji obisk, se mora prijaviti v sistem. Če še ni prijavljen, se mora predhodno registrirati. Pri registraciji je potrebno vnesti podatke, kot so ime, priimek, želena uporabniško ime, naslov elektronske pošte, regijo, v kateri uporabnik prebiva ter geslo za dostop do sistema, ki se v bazo podatkov zapiše kot zgoščena vrednost izračunana zgoščevalni funkciji SHA1. V

primeru nepooblaščenega dostopa do uporabniških podatkov s tem zagotovimo, da podatki o geslu ne pridejo v napačne roke.

V primeru, da uporabnik ne vnese vseh zahtevanih podatkov, se mu izpiše sporočilo, kateri podatek manjka in se mu obrazec z vsemi že prej vnesenimi podatki ponovno prikaže. S tem zagotovimo čim lažje delo uporabniku, da mu ni potrebno še enkrat vnašati vseh podatkov. Enako se mu pri neuspešni registraciji prikaže obvestilo, da ta ni uspela ter naj poizkusi ponovno. Preveri se tudi, če se geslo in potrditev gesla ujemata, v nasprotnem primeru je uporabnik obveščen, da naj vpiše enako potrditev gesla. Ker sistem ne omogoča pošiljanja elektronskih sporočil, se po uspešni registraciji uporabnika prijavi in preusmeri na vstopno stran. Za preverjanje, ali je res uporabnik tisti, za katerega se predstavlja, bi bilo boljše, da bi se mu v takem primeru poslalo elektronsko sporočilo s povezavo, katera bi ga preusmerila na spletno stran in bi se šele nato, če uporabnik ne bi spremenil povezave aktiviral njegov račun.

Ko je uporabnik registriran in prijavljen v sistem, lahko na strani Moj profil (Slika 8) uredi svoje uporabniške nastavitve, ki jih je vnesel ob registraciji. Pomembneje pa je, da lahko na tej strani označi aktivnosti, ki so mu všeč ter izpolni par osebnih podatkov, ki se potem uporabijo pri generiranju lokacij, ki bi bile zanj zanimive.

**Slika 8: Stran s profilom uporabnika**

Za zagotavljanje čim boljše napovedi se vodi evidenca obiskanih lokacij (oz. njihovih aktivnosti) za vsakega uporabnika posebej. Ko prijavljen uporabnik vstopi na stran s podrobnostmi o lokaciji, se mu v spodnjem delu prikaže obrazec za vnos lokacijskih aktivnosti (Slika 9), ki jih je obiskal, pri seznamu lokacijskih aktivnosti pa se mu za vsako aktivnost pokaže potrditveno polje (*CheckBox*), s katerim uporabnik označi, da je to lokacijsko aktivnost obiskal. V obrazcu še izbere datum, kdaj je izbrane lokacijske aktivnosti obiskal in jim določi oceno od 1 do 5. Če mu vse aktivnosti na tej lokaciji niso bile enako

všeč, potem mora vsako lokacijsko aktivnost posebej izbrati in izpolniti obrazec z oceno in datumom obiska. Enako stori tudi v primeru, če ni vse lokacijske aktivnosti obiskal na isti datum. Če je uporabnik vnesel datum, ki je večji od današnjega, ga na to tudi opozorimo in mu ne dovolimo shraniti aktivnosti. Tega mu ne dovolimo zato, ker če si uporabnik shrani obisk za prihodnost, ni nujno da bo tudi odšel, sistem pa ne predvideva popravljanja oz. brisanja že shranjenih obiskov. Dovolimo mu pa tudi ne zato, ker lokacije dejansko še ni obiskal in ne more podati realne ocene.

Po pritisku na gumb »Označi izbrane aktivnosti kot obiskane«, se zberejo vse aktivnosti, ki jih je uporabnik izbral in se podatki vnesejo v tabelo zgodovina lokacij za vsako aktivnost posebej. Pri tem je pomembno, da se vse izbrane aktivnosti zapišejo v bazo. V primeru, da to ni mogoče, se uporabnika opozori, da je prišlo do napake in da naj poizkusi ponovno. Uporabnika se tudi opozori, da ni vnesel vseh podatkov v obrazec (Slika 9). V nasprotnem primeru se mu prikaže obvestilo, da je zapis uspel. Po uspešnem vnosu vseh lokacijskih aktivnosti, se izpraznijo vsa polja v obrazcu, ter odstranijo vse označitve potrditvenih polj. S tem omogočimo uporabniku, da lahko na isti lokaciji takoj vnese podatke še za ostale lokacijske aktivnosti.

The image displays two side-by-side screenshots of a web form for logging location activities. The left screenshot shows an error state where the 'Datum obiska' (Date of visit) field is empty, and a red message above it reads 'Polje 'Datum' ne sme biti prazno.' (The 'Date' field cannot be empty). The 'Ocena' (Rating) dropdown menu is set to '3'. The right screenshot shows a success state with a blue message above the form: 'Uspešno ste shranili lokacijo pod moje lokacije' (You have successfully saved the location under my locations). In this state, the 'Datum obiska' field is empty, and the 'Ocena' dropdown menu is set to '[Izberi oceno]' (Select rating). Both forms feature a button at the bottom labeled 'Označi izbrane aktivnosti kot obiskane' (Mark selected activities as visited).

**Slika 9: Prikaz neuspešnega in uspešnega vnosa lokacijskih aktivnosti v zgodovino**

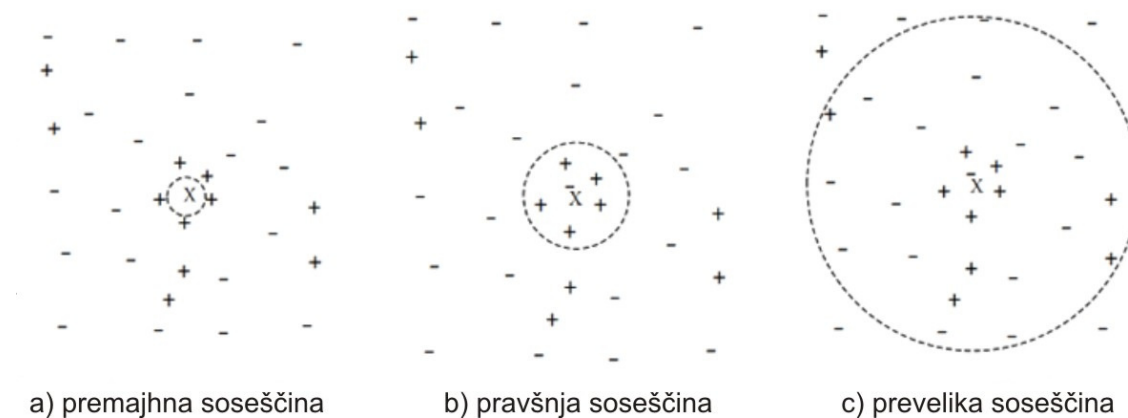
Po vnosu lokacijskih aktivnosti v zgodovino, lahko uporabnik preveri, če so se te resnično shranile. To izvrši tako, da v meniju izbere povezavo »Moje lokacije«. Na tej strani ni mogoče spreminjati že vnesene podatke, ampak ima uporabnik samo pregled nad vsemi dosedanjimi obiski. Izpišejo se mu podatki za vsako lokacijsko aktivnost posebej, na kateri lokaciji se nahaja, kdaj jo je obiskal ter ocena, ki jo je zanj določil. Enaka aktivnost se lahko v seznamu pojavi večkrat, saj jo je uporabnik lahko obiskal na različne datume. Dopusča se tudi možnost, da je neka aktivnost obiskana večkrat na isti datum, saj se lahko uporabnik v istem dnevu vrne na to lokacijsko aktivnost in ji tudi določi drugačno oceno.

Ko na novo prijavljeni uporabnik vstopi v sistem, mu na strani s predlogi lahko predlagamo lokacije po istem sistemu, kot če bi uporabnik bil neprijavljen. Za boljši rezultat mora uporabnik vnesti kriterije na njegovem profilu. Po teh kriterijih lahko uporabniku predlagamo samo tiste lokacije, ki bi ga utegnile zanimati. Če je uporabnik že obiskal kakšno lokacijo in

njene aktivnosti, je tudi priporočljivo da te takoj po registraciji vnese in se ne pojavijo v seznamu s priporočili.

Ko ima uporabnik vnesene podatke o lokacijah, ki jih je že obiskal, mu lahko priporočamo lokacije tudi glede na druge uporabnike sistema. V našem primeru smo za tovrstno priporočanje uporabili algoritem k najbližjih sosedov.

Algoritem k najbližjih sosedov [2, 3] poišče določeno ( $k$ ) število predmetov, ki so najbližje testnemu predmetu. Število  $k$  nam pove, koliko predmetov bomo uporabili pri izračunu vrednosti. Izbira  $k$  je zelo pomembna. Če bi bili podatki popolni, bi lahko za uporabili kar vrednost 1 in s tem za napoved uporabili kar prvega najbližjega soseda. Toda resnični podatki skoraj nikoli niso popolni in moramo zato uporabiti več sosedov. Vendar pa ne smemo izbrati niti premajhnega niti prevelikega števila. Pri prevelikem številu bi zajeli tudi predmete, ki niso podobni našemu (Slika 10).



**Slika 10: kNN z majnim, povprečnim in velikim k-jem**

Pri generiranju priporočil uporabniku, se glede na trenutnega uporabnika izračuna razdalja do ostalih uporabnikov po enačbi Evklidske razdalje (enačba 1). Pri tem se za  $x_k$  vzamejo vrednosti ocen trenutnega uporabnika, za  $y_k$  pa se vzamejo vse vrednosti ocen uporabnika, do katerega računamo razdaljo. Kot rezultat dobimo vrednosti med 0 in 1, kjer vrednost 1 pomeni da se uporabnika popolnoma ujemata, vrednost 0 pa da se sploh ne ujemata. Preverjajo se samo ocene, ki so skupne obema uporabnikoma. Če uporabnika nimata skupnih ocen, potem algoritem vrne kot rezultat 0, kar pomeni, da se ta dva uporabnika ne ujemata. Na takšen način dobimo vrednost, ki nam pove, koliko se želje uporabnikov ujemajo.

Ko se izračunajo vse razdalje se sestavi seznam vseh uporabnikov, razporejenih glede na oddaljenost trenutnega uporabnika. Nato se vzame določeno število uporabnikov, ki so najbližje trenutnemu. V našem primeru se je izkazalo, da je glede na število uporabnikov zapisanih v bazi najbolje vzeti 5 najbližjih. Glede na seznam pridobljenih najbližjih

uporabnikov se iz baze pridobijo podatki o vseh dosedanjih obiskanih aktivnostih. Iz tega seznama so izvzete vse lokacijske aktivnosti, ki jih je uporabnik že obiskal, saj s tem zagotovimo, da uporabniku ne predlagamo vedno ene in iste aktivnosti, ki jo je že obiskal v preteklosti.

Če ima uporabnik na njegovem profilu vnesene aktivnosti, ki ga zanimajo, se nato izmed vseh pridobljenih aktivnostih pridobijo lokacije, ki bi glede na te aktivnosti najbolj ustrezale. V določenem trenutku ima lahko uporabnik posebno željo, kaj hoče početi (na primer, da danes hoče oditi na pohod). V tem primeru uporabnik v obrazcu izbere katere aktivnosti ga najbolj zanimajo ali v kateri regiji hoče to početi in se nato pri predlogih upoštevajo ti kriteriji in ne tisti, katere ima uporabnik shranjene na profilu. To lahko stori tudi v primeru, da noče imeti samo priporočila lokacij glede na aktivnosti, ki jih ima shranjene v profilu, ampak lahko poda da se izberejo lokacije po vseh aktivnostih. Prav tako ima lahko uporabnik željo, za katero regijo mu naj sistem predlaga lokacije, saj se recimo v tistem trenutku nahaja v tej regiji in ne ve, kaj bi si lahko šel pogledat. V nasprotnem primeru se izdelata seznam priporočil po vseh lokacijah.

Registriranemu uporabniku se priporočila lokacij, ki jih naj obišče prikažejo v istem seznamu, kot se prikažejo neregistriranemu. Le da se prikažejo le tiste lokacije, ki jih je sistem pridobil glede na zgoraj opisan postopek

## 4 ZAKLJUČEK

Cilj te diplomske naloge je bil pripraviti spletno aplikacijo, ki uporabnikom priporoča lokacije za izlete. Aplikacija je izdelana do te mere, da zna priporočati izlete glede na kriterije, ki jih neprijavljeni uporabnik izbere v obrazcu. Uporabnik poda aktivnosti, ki ga v tem trenutku zanimajo in se glede na to priporočajo lokacije, katere vsebujejo te aktivnosti. Če pa je uporabnik prijavljen v sistem, mu že sama aplikacija zna predlagati lokacije glede na druge uporabnike, ki so te lokacije že obiskali. Tukaj lahko uporabnik tudi izbere aktivnosti, kateri ga v tem trenutku še posebej zanimajo in mu aplikacija predlaga samo lokacije, ki vsebujejo te aktivnosti.

Če uporabnik izbere aktivnost, ki še ne obstaja na nobeni lokaciji, mu sistem ne more predlagati teh lokacij in tako uporabnik pridobi prazen seznam. Tako bi bilo potrebno v bazi še zbrati veliko lokacij in njihovih aktivnosti, ki obstajajo v Sloveniji. Za najboljši rezultat preveritve delovanja sistema bi bilo potrebno tudi zbrati še več uporabnikov, ki bi vnesli svoje že obiskane lokacije in aktivnosti.

Pri izdelavi priporočilnega sistema je bilo potrebno v bazi predhodno zbrati testne podatke o uporabnikih, lokacijah ter aktivnostih. Tako se je se je zbralo 19 različnih uporabnikov (od tega ima vnesene podatke o obiskanih lokacijah 12 uporabnikov), 36 lokacij ter 53 aktivnosti na teh lokacijah. Vsakemu uporabniku, ki je vnašal svoje podatke, smo omogočili, da je lahko sproti tudi dodajal lokacije ter njihove aktivnosti. Le tako so lahko uporabniki vnašali svoje lokacije, ki so jih kasneje dodali na seznam obiskanih.

Glede na število uporabnikov, je bilo potrebno pri algoritmu najbližjih sosedov izbrati dokaj majhno število sosednjih uporabnikov ( $k$ ). Pri testiranju sistema je bilo ugotovljeno, da lahko osmim izmed vseh pridobimo takšne uporabnike, ki imajo vnesene skupne izlete. Glede na te uporabnike se je določilo, da se lahko v večini primerov za primerjavo vzame 5 sosednjih uporabnikov. Ostalim uporabnikom lahko svetujemo zgolj po kriterijih, ki jih imajo vnesene v svojem profilu. Takšnih uporabnikov je med preostalimi v bazi 5.

Vsaj v primeru našega računa lahko rečemo, da glede na želje in na dosedanje obiskane lokacije, sistem predlaga lokacije, ki so zanimive in bi jih tudi v prihodnosti z veseljem obiskali. Za ostale uporabnike pa glede na to, kaj so vnesli pod svoje kriterije in kakšne lokacije imajo zbrane pod njihovimi obiskanimi, lahko povzamemo da je večina priporočenih lokacij smiselnih.

Da bi lahko uporabnikom priporočali še več podobnih lokacij, predvsem pri uporabnikih, za katerega ne moremo določiti nobenega podobnega ali pa da je teh zelo malo, bi bilo potrebno v sistem še vpeljati algoritem, ki bi po podobnem sistemu kot določa podobnost med uporabniki, pridobil še vse podobne lokacije, ki jih je uporabnik že obiskal. To seveda ne bi bilo mogoče v primeru, da uporabnik nima vnesene še nobene lokacije.

Sistem bi bilo mogoče razširiti z več novimi funkcijami. V modelu so za sezono predvideni kar meseci v letu, kdaj lahko uporabnik obišče določeno aktivnost na lokaciji. Tukaj bi se lahko rešil problem predvsem z prireditvami, ki se ne ponavljajo vsako leto ob istem času tako, da bi se podal razpon datuma (od, do) in bi tako lahko vodili evidenco tudi za prireditve, ki so samo na določen datum (recimo koncerti).

Ker predlagamo samo določene lokacije in mora uporabnik še vedno izbirati med vsemi, da si lahko sestavi celoten izlet, bi bila tako tudi ena izmed možnih razširitev sistema, da bi se lokacijam podale tudi zemljepisne koordinate in bi na podlagi teh koordinat lahko sistem predlagal izlet kot celoto. To pomeni, da bi nam sistem podal celotno pot in katere lokacije so na tej poti, ki bi uporabnika utegnile zanimati.

V primeru, da bi želeli priporočati lokacije skupini ljudi, ki bi se hoteli na določen trenutek odpraviti na potep po Sloveniji, bi bilo potrebno te uporabnike povezati med sabo. Tako bi lahko uporabnik dodal ostale med seznam svojih prijateljev. Ko bi se ti uporabniki odločali, kaj bi si lahko šli pogledati, bi eden izmed teh dodal na seznam vse uporabnike in bi sistem predlagal lokacije, ki jih vsi ti uporabniki še niso obiskali in bi bile za vse udeležence zanimivi.

Uporabniki vedno niso zadovoljni s priporočitvami, ki jim jih ponudi sistem. V tem primeru bi lahko pri vsaki lokaciji, ki jo sistem predlaga dodali še gumb, s katerim bi uporabnik lahko povedal, da te lokacije ne želi med svojimi predlogi. Sistem bi si v posebno tabelo v bazi zabeležil, katere lokacije so to in pri naslednjih priporočitvah tudi te izpustil iz priporočil uporabniku.

Za zaključek naj še omenimo, da trenutna spletna stran po izgledu ni najbolj privlačna uporabniku, ki pride na stran. V tem primeru bi bilo potrebno izdelati celoten izgled strani. Za uporabnika bi bilo tudi zanimivo, če bi se dodala možnost naložitve slik za vsako lokacijo oz. njene aktivnosti. Pri tem bi se lahko dodala možnost vstavljanja slik uporabnikov, ki so obiskali te lokacije.



## VIRI IN LITERATURA

[1] Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor, Recommender Systems Handbook, Springer 2010

[2] Xindong Wu, Vipin Kumar, The Top Ten Algorithms in Data Mining, Chapman and Hall 2009, pogl. 8

[3] Wikipedia: k-nearest neighbor algorithm. Dostopno na [http://en.wikipedia.org/wiki/K-nearest\\_neighbor\\_algorithm](http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm)

[4] Wikipedia: Recommender system. Dostopno na [http://en.wikipedia.org/wiki/Recommender\\_systems](http://en.wikipedia.org/wiki/Recommender_systems)