

Универзитет у Београду  
Факултет организационих наука

Пројектни рад  
**WhereToGo**

Предмет: Алати и методе вештачке интелигенције и  
софтверског инжењерства

Ментор:

др Драган Ђурић

Студент:

Небојша Бранковић 2023/3828

Београд, 2025.

# Садржај

1	Вербални опис .....	3
2	Предуслов .....	4
3	Корисничко упутство .....	5
3.1	Логовање и креирање корисника .....	5
3.2	Главни мени.....	6
3.2.1	Пронађи место .....	7
3.2.2	Остављање рецензије .....	13
3.2.3	Преглед рецензија .....	15
3.2.4	Измена профила .....	16
4	Имплементација .....	19
5	Закључак.....	27

# 1 Вербални опис

WhereToGo је CLI апликација намењена проналажењу најбољих места у околини корисника. Корисник приликом првог покретања уноси основне податке (корисничко име, адресу, поштански број, град, државу и опцију да ли му је потребан приступ за инвалидска колица).

Апликација, користећи OpenStreetMap Nominatim API, добија координате на основу адресе. Подаци о местима (ресторани, кафићи, барови, библиотеке, позоришта, биоскопи, клубови итд.) учитавају се из JSON фајлова у Codaх базу. Подаци који су коришћени за места преузети су преко Overpass Turbo алата и представљају стварне објекте у Београду. За свако место доступне су информације као што су: назив, тип објекта, адреса и доступност за кориснике инвалидских колица. Пошто у OpenStreetMap-у не постоје подаци о просечној оцени и рецензијама, у core.clj је имплементирана функција која сваком месту додељује насумичне оцене и коментаре, како би апликација имала реалистичнији приказ и омогућила тестирање функционалности.

Кориснику се затим предлажу најбоља места на основу:

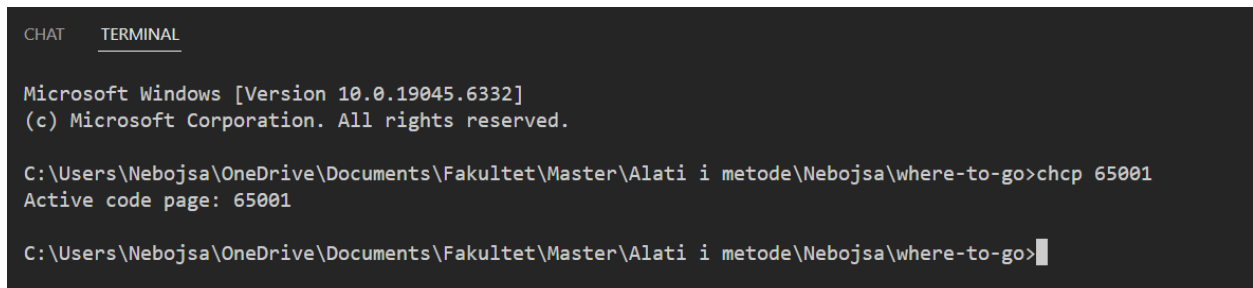
- удаљености,
- просечне оцене места,
- приступачности за инвалидска колица (ако корисник наведе да му је потребна).

Осим тога, апликација омогућава остављање и читање рецензија, измену профила, као и приказ историје рецензија тренутно улогованог корисника.

## 2 Предуслов

Предуслов:

Иако је интерфејс апликације на енглеском језику, подаци који се користе (називи објеката и адресе) дати су на српској ћирилици. Да би се подаци правилно приказали у терминалу, неопходно је пре покретања апликације подесити кодну страницу на UTF-8. То се ради покретањем следеће команде у терминалу:



```
CHAT  TERMINAL

Microsoft Windows [Version 10.0.19045.6332]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Nebojsa\OneDrive\Documents\Fakultet\Master\Alati i metode\Nebojsa\where-to-go>chcp 65001
Active code page: 65001

C:\Users\Nebojsa\OneDrive\Documents\Fakultet\Master\Alati i metode\Nebojsa\where-to-go>
```

*Слика 1 Предуслов за коришћење апликације*

## 3 Корисничко упутство

Апликација се покреће извршавањем команде `lein run`, након чега се кориснику приказује упит за пријаву (логовање).

### 3.1 Логовање и креирање корисника

Када апликација затражи унос корисничког имена, унесите своје корисничко име (username) уколико већ имате налог, или жељено корисничко име уколико први пут користите апликацију.

```
C:\Users\Nebojsa\OneDrive\Documents\Fakultet\Master\Alati i metode\Nebojsa\where-to-go>lein run
Enter your username:
> Zoki
```

*Слика 2 Унос корисничког имена*

Уколико постоји корисник са тим именом у бази, систем ће вас поздравити и приказати главни мени.

Ако не постоји корисник са тим корисничким именом, апликација ће покренути процес креирања новог профила који је сачињен од неколико питања о кориснику.

Подаци који се прикупљају односе се на тренутну локацију корисника, као и на потребу да место буде приступачно особама у колицима.

На следећој слици приказана су релевантна питања. Потребно је унети стварну, постојећу адресу, јер се користи API за израчунавање координата локације (како би се установила раздаљина до сваког места у бази). Такође, битно је напоменути да се тренутно у бази налазе искључиво подаци о местима у Београду.

```
This is your first time using WhereToGo.
Please enter your data so we can find the best place for you.

Street name:
> Maksima Gorkog

Street number:
> 21

City:
> Beograd

Country:
> Srbija

Postal code:
> 11000

Wheelchair accessible? (yes/no):
> yes

Your user has been successfully created
```

*Слика 3 Питања при креирању корисника*

## 3.2 Главни мени

После успешног логовања видећете главни мени:

```
What can I do for you today?

1. Find a place
2. Leave a review
3. Check history
4. Edit profile
0. Exit

Select an option:
> █
```

*Слика 4 Главни мени*

У даљем тексту су детаљније објашњене опције главног менија:

1. Пронађи место

2. Остави рецензију
3. Преглед историје
4. Измени профил

### 3.2.1 Пронађи место

Покретање система за проналазак места. Унутар ове опције могуће је навести тип места (ресторан, биоскоп, позориште, ноћни клуб, итд.) који корисник жели да посети, могуће је тражити насумичних пет места, а такође је могуће добити специјализовану препоруку пет места на основу неколико питања која су постављена кориснику.

#### 3.2.1.1 Случај 1: Корисник зна тип места у који жели да иде

Уколико корисник има идеју на који тип места жели да иде, ток апликације је приказан на следећим сликама:

```
What can I do for you today?

1. Find a place
2. Leave a review
3. Check history
4. Edit profile
0. Exit

Select an option:
> 1

Do you already have a type of place in mind for today's outing?
1. Yes
2. No
0. Back

Select an option:
> 1
```

Слика 5 Корисник има идеју на који тип места жели да иде

Кориснику се приказује питање: „Да ли већ имаш на уму врсту места за данашњи излазак?“.

За наш ток процеса, на ово питање је потребно одговорити потврдно (број 1).

```
Great! Which type of place are you interested in today?
1. Restaurant
2. Bar
3. Cafe
4. Pub
5. Casino
6. Library
7. Theatre
8. Cinema
9. Nightclub
0. Back

Select an option:
> 1
```

Слика 6 Одабир типа жељеног места (у овом случају је одабран ресторан)

Кориснику се приказује следећа порука: „Сјајно! За који тип места си заинтересован данас?“.

На ово питање је могуће одговорити бројевима од 0 до 9, с тим што се одговорем бројем 0 враћате на претходни мени. Било којим другим једноцифреним бројем могуће је изабрати тип места. У нашем примеру, бирамо број 1 за ресторан.

```
Here are 5 suggested places where you can go today:

NR  Name                                     Type    Distance  Rating  Wheelchair  Address
1.  Кафана „Душановачки цвет“              restaurant  1,6 km    5.0     Yes         Устаничка 78, Београд
2.  Bitolj                                   restaurant  5,4 km    4.7     Yes         Зрењанински пут 39, Београд
3.  Коло                                     restaurant  2,6 km    3.8     Yes         Булевар Вудроа Вилсона 14, Београд
4.  Пролеће                                  restaurant  2,7 km    3.7     Yes         Вука Караџића 11, Београд
5.  YXlanzh                                  restaurant  2,4 km    3.5     Yes         Југ Богданова , Београд
0.  Back

Choose a place:
> 
```

Слика 7 Приказ предложених места

Конечно, добијамо листу са 5 предложених места који су унетог типа (у нашем случају ресторан). За свако место приказујемо следеће податке: назив, тип места, удаљеност, оцена, доступност колицима и адреса објекта.

Ови предлози су приказани на основу претходно поменутих параметра, а то су: удаљеност између унете локације корисника и места, просечне оцене места, као и приступа колицима (у случају потребе за истим).



Могуће је погледати детаљније једно од места одабиром броја од 1 до 5. Ми ћемо приказати детаљније другу опцију (ресторан Битољ):

```
Here is some more information:
Name: Bitolj
Type: restaurant
Distance: 5,4 km
Rating: 4.7
Wheelchair: Yes
Address: Зрењанински пут 39, Београд

Reviews:
- Nevena11 (Rating: 4,4) - Amazing!
- Jovana (Rating: 4,9) - Pretty good.

1. Back
0. Exit

Choose an option:
> 
```

*Слика 8 Детаљан приказ одабраног места*

У детаљном приказу места могуће је видети рецензије других корисника, поред претходно приказаних података у табели. У овом случају за одабрани ресторан Битољ постоје две рецензије које можемо видети.

#### *3.2.1.2    Случај 2: Корисник жели пет насумичних места*

Уколико корисник нема представе где жели да иде и хоће пет насумичних места ток апликације је приказан у наредним сликама.

```
What can I do for you today?

1. Find a place
2. Leave a review
3. Check history
4. Edit profile
0. Exit

Select an option:
> 1

Do you already have a type of place in mind for today's outing?
1. Yes
2. No
0. Back

Select an option:
> 2
```

Слика 9 Корисник нема идеју на који тип места жели да иде

Кориснику се приказује питање: „Да ли већ имаш на уму врсту места за данашњи излазак?“.

За наш ток процеса, на ово питање је потребно одговорити одрично (број 2).

```
That's okay! What would you prefer?
1. Suggest 5 random places
2. Give bespoke suggestion after couple of questions
0. Back

Select an option:
> 1
```

Слика 10 Одабир приказивања пет насумичних места

Кориснику се приказује порука:

„То је океј. Шта би више волео:

1. Предлог пет насумичних места
2. Јединствен предлог на основу пар питања
0. Назад“

За наш ток процеса потребно је одговорити бројем 1, како бисмо добили приказ пет насумичних места.

```
Here are 5 suggested places where you can go today:
```

NR	Name	Type	Distance	Rating	Wheelchair	Address
1.	Кафана „Душановачки цвет“	restaurant	1,6 km	5.0	Yes	Устаничка 78, Београд
2.	Кофилин	cafe	3,0 km	4.8	Yes	Цара Душана 29, Београд
3.	Bitolj	restaurant	5,4 km	4.7	Yes	Зрењанински пут 39, Београд
4.	Пивски забавник	pub	0,5 km	4.1	Yes	Булевар краља Александра 136, Београд
5.	Cafe & Factory	cafe	1,7 km	4.1	Yes	Хајдук Вељков венац 12, Београд
0.	Back					

```
Choose a place:  
> 
```

Слика 11 Приказ пет насумичних места

Кориснику је приказано пет места било којег типа. С обзиром да је овај приказ већ објашњен у претходном тексту, битно је само напоменути да се и даље користе подаци о удаљености, просечној оцени и приступачности колицима како би се приказали што адекватнији предлози. Места нису потпуно насумична, како не би били приказани објекти са лошим оценама или превеликом удаљеношћу од корисника.

Такође је могуће приказати рецензије сваког објекта одабиром једног од објекта (Слика 8).

### 3.2.1.3 Случај 3: Јединствен предлог на основу пар питања

Овај случај представља сценарио када корисник не зна тип места на који жели да иде и хоће кроз пар питања да добије бољи предлог за себе. Почетак процеса тока овог случаја је исти као у случају 2, приказано на слици 9. Следећи кораци су приказани у наредном делу:

```
That's okay! What would you prefer?  
1. Suggest 5 random places  
2. Give bespoke suggestion after couple of questions  
0. Back  
  
Select an option:  
> 2
```

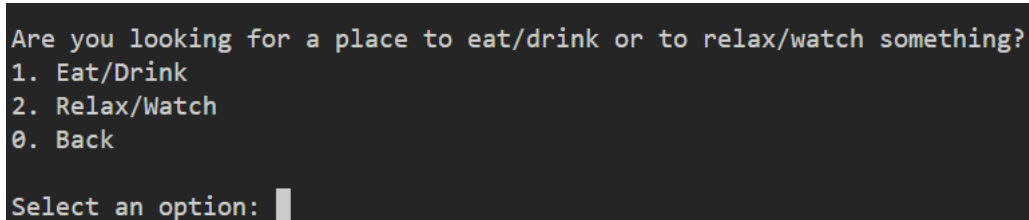
Слика 12 Одабир јединственог предлога на основу пар питања

Кориснику се приказује порука:

„То је океј. Шта би више волео:

1. Предлог пет насумичних места
2. Јединствен предлог на основу пар питања
1. Назад“

За наш ток процеса потребно је одговорити бројем 2.



```
Are you looking for a place to eat/drink or to relax/watch something?  
1. Eat/Drink  
2. Relax/Watch  
0. Back  
Select an option: |
```

*Слика 13 Питање 1*

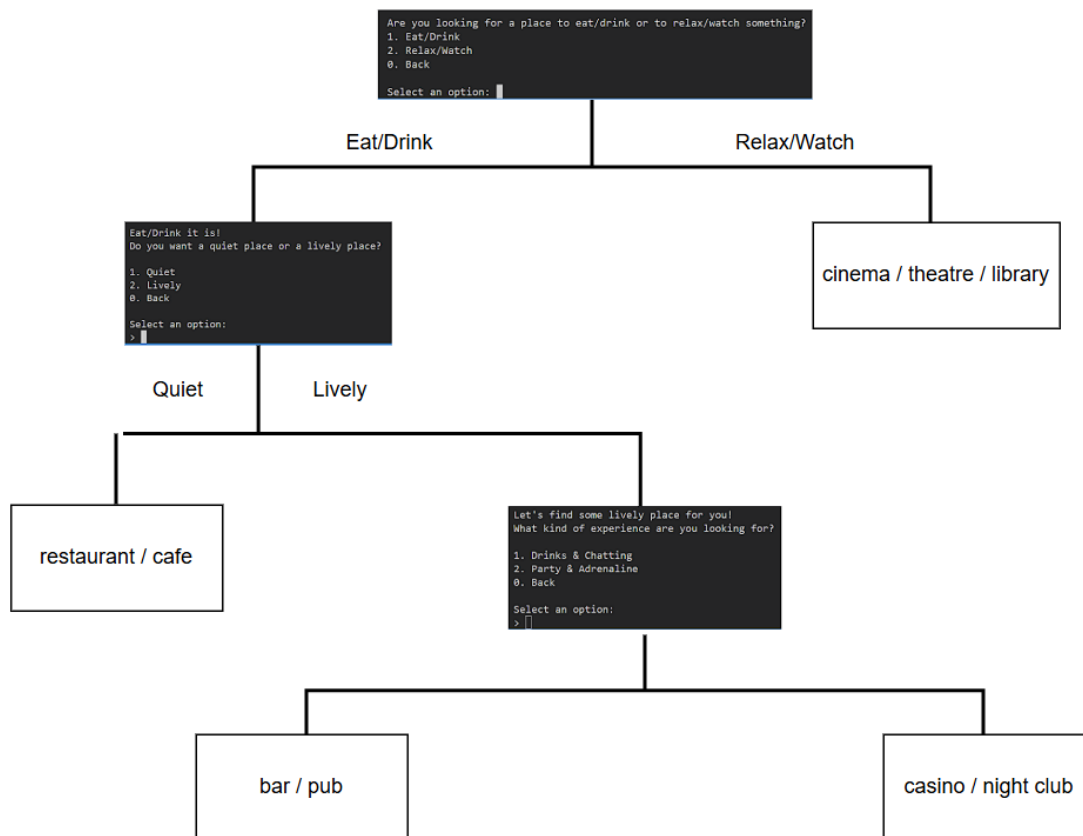
Даљи ток процеса је приказан у виду стабла одлучивања који је приказан испод због једноставнијег прегледа (слика 14).

Као што можемо приметити, уколико корисник изабере одмор/гледање (Relax/Watch), добиће предлог пет најбољих места који су биоскоп, позориште или библиотека.

У случају одабира храна/пиће (Eat/Drink) кориснику се поставља питање да ли жеку живље место или тихо. У случају тихог места кориснику ће бити приказано пет места који су ресторан или кафић.

Уколико корисник тражи живље место, потребно је да одговори да ли преферира пиће и причу или је више за журку и адреналин.

Одабир пића и приче прати приказ пет места који су бар или паб, а у случају одабира журке и адреналина биће приказано пет места који су или ноћни клуб (дискотека) или казино.



Слика 14 Стабло одлучивања за јединствен предлог

### 3.2.2 Остављање рецензије

Као што назив каже, одабиром ове опције могуће је оставити рецензију за једно од места у програму. На следећој слици је приказано успешно остављање рецензије.

```
What can I do for you today?

1. Find a place
2. Leave a review
3. Check history
4. Edit profile
0. Exit

Select an option:
> 2

Enter the name of the place you want to review:
> Thyme

Enter rating (1-5):
> 4.2

Enter comment:
> Great food, awesome taste. Unfortunately I had to wait for my food for some time, but it was delicious.

Review saved! New average rating for Thyme : 4.1
```

*Слика 15 Остављање рецензије места*

Потребно је унети назив места, затим оставити оцену и коментар рецензије.

Добијамо потврдну информацију о успешно сачуваној рецензији као и нову просечну оцену наведеног места.

```
What can I do for you today?

1. Find a place
2. Leave a review
3. Check history
4. Edit profile
0. Exit

Select an option:
> 2

Enter the name of the place you want to review:
> ддд

Place not found. Make sure you typed the name correctly.
```

*Слика 16 Непостојуће место*

У случају унетог назива места који не постоји, добија се порука грешке да место није пронађено. Корисник се враћа на главни мени, где може поново да покуша да унесе назив места при додавању рецензије.

```
What can I do for you today?

1. Find a place
2. Leave a review
3. Check history
4. Edit profile
0. Exit

Select an option:
> 2

Enter the name of the place you want to review:
> Thyme

Enter rating (1-5):
> 5.2
Invalid input, try again.

Enter rating (1-5):
> 
```

*Слика 17 Погрешан унос оцене*

Такође, ради се провера оцене која је додељена. Уколико оцена није у распону од 1 до 5, добија се грешка. Корисник може поново да упише оцену која је у одговарајућем распону.

### 3.2.3 Преглед рецензија

Кориснику је омогућен преглед сопствених рецензија. Одабиром ове опције се приказује листа свих претходних рецензија што се може видети на наредној слици.

```
What can I do for you today?

1. Find a place
2. Leave a review
3. Check history
4. Edit profile
0. Exit

Select an option:
> 3

Your reviews:

Place: Meze
Type: restaurant
Rating: 5.0
Comment: Sjajna atmosfera i hrana

Place: Mamma Mia
Type: restaurant
Rating: 4.5
Comment: Sjajna pica, kul atmosfera.

Place: Thyme
Type: restaurant
Rating: 4.2
Comment: Everything was great, but the food was a bit cold for my liking.

Rating: 4.2
Comment: Great food, awesome taste. Unfortunately I had to wait for my food for some time, but it was delicious.

Rating: 5.0
Comment: Awesome!

Place: Tradicija
Type: restaurant
Rating: 3.4
Comment: Okej, nista spec
```

*Слика 18 Преглед рецензија*

### 3.2.4 Измена профила

- Приказује текуће податке профила (улица, број, град, држава, поштански број, статус инвалидских колица).
- Могућност измене свих претходно наведених података корисника.



```
This is your current profile information:

Username: Zoki
Street name: Maksima Gorkog
Number: 21
City: Beograd
Postal code: 11000
Country: Srbija
Wheelchair status: Yes

What would you like to change?
1. Street name
2. Number
3. City
4. Postal code
5. Country
6. Wheelchair status
0. Back

Select an option:
> 
```

Слика 19 Измена података корисника

Одабиром неког од података корисника, могуће је изменити исти. На следећој слици је приказана измена броја куће.

```
Enter new street number:
> 24
Street number changed to: 24

What can I do for you today?

1. Find a place
2. Leave a review
3. Check history
4. Edit profile
0. Exit

Select an option:
> 
```

Слика 20 Измена броја куће

Корисник уписује нови број куће након чега добија потврду да је број промењен.

Поновним приказом измене профила можемо видети да је број куће заиста измењен:

```
This is your current profile information:

Username: Zoki
Street name: Maksima Gorkog
Number: 24
City: Beograd
Postal code: 11000
Country: Srbija
Wheelchair status: Yes

What would you like to change?
1. Street name
2. Number
3. City
4. Postal code
5. Country
6. Wheelchair status
0. Back

Select an option:
> █
```

Слика 21 Потврда измене податка корисника

## 4 Имплементација

У апликацији *WhereToGo* имплементирана је *Codax* база података, која омогућава једноставно складиштење и учитавање података без потребе за екстерним сервером. Подаци се чувају у локалној датотеци *data/database*, док се приступ бази и манипулација подацима врши помоћу Clojure функција дефинисаних у namespace-у *where-to-go.data*.

База је организована тако да садржи кључне ентитете: **places** (места) и **users** (корисници). Свако место садржи информације о називу, типу места, координатама, доступности за особе у колицима, као и рецензије корисника са просечном оценом. Корисници садрже информације о корисничком имену, адреси и статусу доступности за особе у колицима.

Примарне функције за рад са базом укључују:

- `get-all-places` – враћа све локације из базе

```
59 ;; Get all places from db
60 √ (defn get-all-places []
61     (c/get-at! db [:places]))
62
```

Слика 22 Функција *get-all-places*

- `save-places` – чува сва места која се налазе у JSON фајловима

```
54 ;; Save into Codax DB under key :places
55 (defn save-places! []
56     (let [places (vec (load-all-places))]
57         (c/assoc-at! db [:places] places)))
```

Слика 23 Функција *save-places*

- `add-review-to-place!` – додаје нову рецензију месту и аутоматски ажурира просечну оцену

```

63 (defn add-review-to-place! [place-id review]
64   ;; Load all places from DB
65   (let [places (vec (get-all-places))] ;; ensure vector
66     (let [updated-places
67           (mapv (fn [p]
68                   (if (= (:place_id p) place-id)
69                     (let [updated-reviews (conj (or (:reviews p) []) review)
70                           avg (avg-rating updated-reviews)
71                           updated-place (assoc p
72                                                 :reviews updated-reviews
73                                                 :avg_rating (/ (Math/round (* avg 10.0)) 10.0))]
74                       updated-place)
75                     p))
76           places)]
77     ;; Persist back to the same DB path
78     (c/assoc-at! db [:places] updated-places)
79     ;; Return updated places
80     updated-places)))

```

Слика 24 Функција *add-review-to-place!*

- create-user, read-user, update-user – основне операције над корисницима

Подаци о местима иницијално се учитавају из JSON фајлова (*data/\*.json*) који дефинишу различите типове објеката (ресторани, барови, библиотеке, биоскопи итд.). Фајлови се парсирају и трансформишу у структуру базе преко функције *feature->place*. Ова функција извлачи координате, тип места, назив, статус приступачности за инвалидска колица и иницијализује празан низ рецензија и просечну оцену 0.0.

```

9   ;; Function to transform a single feature into schema
10  √ (defn feature->place [feature]
11    √ (let [props (:properties feature)
12            coords (get-in feature [:geometry :coordinates])
13            wheelchair-raw (:wheelchair props)
14            wheelchair (case wheelchair-raw
15                          "yes" true
16                          "no" false
17                          nil)] ;; leave nil if not specified
18      √ {:place_id      (or (:id feature) (get props "@id"))
19         :amenity_type  (:amenity props)
20         :name          (:name props)
21         :lat_coordinate (second coords)
22         :long_coordinate (first coords)
23         :wheelchair     wheelchair
24         :reviews        []
25         :avg_rating     0.0}))

```

Слика 25 Функција *feature->place*

За претварање адресе у координате и обрнуто, користи се *Nominatim OpenStreetMap API*. Функције *get-coordinates-from-address* и *get-address-from-coordinates* омогућавају претварање корисничког уноса у географске координате и добијање адресе по координатама, респективно. Сви API позиви укључују User-Agent заглавље, како би комуникација била прихваћена.

Функција *get-address-from-coordinates* прихвата 2 параметра: *lat* (латитуда) и *lon* (лонгитуда), а као резултат враћа адресу (улица, број, град, итд.).

```

87 (defn get-address-from-coordinates [lat lon]
88   (let [url "https://nominatim.openstreetmap.org/reverse"
89         params {"format" "json"
90                  "lat" (str lat)
91                  "lon" (str lon)
92                  "zoom" "18"
93                  "addressdetails" "1"}
94         response (client/get url
95                               {:headers {"User-Agent" user-agent}
96                                :query-params params
97                                :as :json
98                                :throw-exceptions false})
99         status (:status response)]
100    (if (= 200 status)
101      (let [addr (get-in response [:body :address])]
102        (if addr
103          (format-address addr)
104          (or (get-in response [:body :display_name])
              "Unknown address"))))
106      (do
107        (println "API call failed with status:" status "body:" (:body response))
108        "Unknown address"))))

```

Слика 26 Функција *get-address-from-coordinates*

Функција *get-coordinates-from-address* прихвата 5 параметара: назив улице, број, град, држава и поштански број. Као резултат се добија *lat* (латитуда) и *lon* (лонгитуда). Ова функција је битна како би се одредила раздаљина између корисника и места.

```

110 (defn get-coordinates-from-address [street-name street-number city country postal-code]
111   (let [url "https://nominatim.openstreetmap.org/search"
112         query (str street-name " " street-number ", " city ", " postal-code ", " country)
113         response (client/get url
114                               {:headers {"User-Agent" user-agent}
115                                :query-params {"q" query
116                                                "format" "json"
117                                                "limit" "1"}
118                                :as :json
119                                :throw-exceptions false})]
120    (when (= 200 (:status response))
121      (let [result (first (:body response))]
122        {:lat (Double/parseDouble (:lat result))
123         :lon (Double/parseDouble (:lon result))})))
124

```

Слика 27 Функција *get-coordinates-from-address*

Поменуто одређивање раздаљине се извршава у функцији *distance\_calc* која враћа као резултат раздаљину између координата у километрима. Функција је приказана на следећој слици:

```

288 (defn distance_calc
289   "Returns distance in km between two lat/lon pairs."
290   [lat1 lon1 lat2 lon2]
291   (let [R 6371.0 ;; Earth radius in km
292         to-rad (fn [deg] (* deg (/ Math/PI 180)))
293         dlat (to-rad (- lat2 lat1))
294         dlon (to-rad (- lon2 lon1))
295         a (+ (Math/pow (Math/sin (/ dlat 2)) 2)
296              (* (Math/cos (to-rad lat1))
297                 (Math/cos (to-rad lat2))
298                 (Math/pow (Math/sin (/ dlon 2)) 2)))
299         c (* 2 (Math/atan2 (Math/sqrt a) (Math/sqrt (- 1 a))))]
300     (* R c)))
301

```

Слика 28 Функција *distance\_calc*

## Алгоритам препоруке места

Главна функција која реализује препоруке је *suggest-places*, а њен рад се заснива на помоћној функцији *score-place*.

### 1. *score-place* – израчунавање укупног скорa за место

За свако место у бази израчунава се скор који комбинује више фактора:

- Просечна оцена (*avg\_rating*) – мери колико су корисници добро оцењивали место.
- Удаљеност од корисника до места – рачуна се разлика у координатама корисника и места. Што је место ближе, добија већи скор.
- Приступачност за инвалидска колица – ако је место означено као доступно за инвалидска колица (*wheelchair = true*), добија додатне поене; ако није, скор је нижи.

Формула је имплементирана тако да комбинује ове параметре у један бројчани резултат, што омогућава поређење свих места.

Додати су тежински коефицијенти за просечну оцену и удаљеност. С обзиром да је битније да место буде добро оцењено него близу, тежински коефицијент оцене је 0,7, а удаљености до корисника 0,3. Такође, уколико је корисник назначио да му је битно да место има приступ колицима, кажњава се свако место које нема приступ тако што се одузима 10 поена, а уколико има приступ колицима додају се 2 поена. С обзиром да велики број места нема наведену информацију о приступачности колицима, таквим местима се одузима само 1 поен. Могуће је да таква места имају приступ колицима, али једноставна нема потврде о томе.

```
305 (defn score-place [p user-lat user-lon]
306   (let [dist (distance_calc user-lat user-lon
307                             (:lat_coordinate p)
308                             (:long_coordinate p))
309         norm-dist (exp-decay-distance dist)
310         norm-rating (double (or (:avg_rating p) 0))
311
312         ;; wheelchair preference
313         needs-wheelchair (:wheelchair @current-user)
314         wheelchair-score (cond
315                           ;; user requires wheelchair but place explicitly says no → strong penalty
316                           (and needs-wheelchair (= false (:wheelchair p))) -10
317                           ;; user requires wheelchair and place is yes → bonus
318                           (and needs-wheelchair (= true (:wheelchair p))) 2
319                           ;; user requires wheelchair but unknown → small penalty
320                           (and needs-wheelchair (nil? (:wheelchair p))) -1
321                           ;; user doesn't care → neutral
322                           :else 0)
323
324         ;; weights
325         alpha 0.7 ;; rating weight
326         beta 0.3  ;; distance weight
327
328         score (+ (* alpha norm-rating)
329                  (* beta norm-dist)
330                  wheelchair-score)]
331     (assoc p
332           :distance dist
333           :distance-str (format "%.1f km" dist)
334           :score score)))
335
```

Слика 29 Функција *score-place*

## 2. suggest-places – избор најбољих места

Кораци које ова функција прати:

- Узима координате активног корисника из *current-user*



- Филтрира листу свих места – уколико је корисник изабрао одређен тип (нпр. кафић) приказује само тај тип места.
- За свако место тог типа позива функцију *score-place* и рачуна његов скор
- Сортира места по скоровима од највећег до најмањег
- Враћа првих 5 места као препоруке

```

336 ;; Suggests 5 places based on entered criteria and algorithm that makes a decision on which places
337 ;; are best for this user. (It uses attributes such as distance to the place and average rating)
338 (defn suggest-places
339   ([] (suggest-places nil))
340   ([types]
341    (let [user-lat (:lat_coordinate @current-user)
342          user-lon (:long_coordinate @current-user)
343          needs-wheelchair (:wheelchair @current-user)
344
345          places (->> (data/get-all-places)
346                      ;; filter by type if provided
347                      (filter #(or (nil? types)
348                                   (some #{(keyword (:amenity_type %))} types)))
349                      ;; filter wheelchair if required
350                      (filter #(or (not needs-wheelchair)
351                                   (not= false (:wheelchair %))))
352                      ;; add distance + score
353                      (map #(score-place % user-lat user-lon))
354                      ;; sort by score descending
355                      (sort-by :score >)
356                      (take 5)
357                      vec)]
358      (reset! current-places places)
359    ))

```

Слика 30 Функција *suggest-places* (део који је битан за алгоритам)

Алгоритам који је коришћен:

- омогућава персонализацију – корисник добија места која су ближа њему и боље оцењена.
- показује друштвени аспект – укључује мишљења других корисника кроз преглед рецензија и оцена.
- подржава инклузију – узима у обзир доступност за особе са инвалидитетом.

## Менији и интерфејси

Апликација користи текстуални мени са подменијима:

- `main-menu` – главни мени апликације
- `find-place` и његове подфункције (`menu-a1`, `menu-a2`, `menu-a22`, `menu-a221`, `menu-a2212`, `menu-a222`) омогућавају кориснику да бира тип места и добије персонализоване препоруке.
- `edit-profile` – користи постојеће функције у *data.clj* како би приказао и изменио податке корисника.

## Покретање апликације

Апликација се покреће функцијом `–main`, која:

- учитава места из базе у атом *places*
- покреће процес логовања корисника (*login*)
- приказује главни мени апликације (*main-menu*)

Закоментарисане линије представљају следеће:

- чување места из JSON фајлова у базу података – ова функционалност се користи само једном како би се популисала база.
- додавање насумичних рецензија – као што је већ поменуто, скуп података који је коришћен није имао доступне рецензије и оцене објеката и зато је направљена функција која додаје насумичне рецензије за свако место у бази података.
- Приказ свих места у бази података.

## 5 Закључак

*WhereToGo* је апликација која омогућава корисницима да лако пронађу места у својој околини на основу удаљености, оцена и приступачности за особе у колицима. Развијена је као самостална Clojure апликација са имплементацијом алгоритма препоруке и интеграцијом базе података, чиме пружа практично и кориснички оријентисано решење.