

Akademia Nauk Stosowanych w Tarnowie

Katedra Informatyki



Filip Broniek

Jakub Chochółowicz

Bartłomiej Cich

Szymon Grzeszczuk

Projekt MAXSAT
System rekomendacji miejsc parkingowych

Spis treści

1. Projekt / zarys wyglądu aplikacji	3
2. Baza danych	4
3. Wykorzystane technologie	5
4. Zmienne zdaniowe	6
5. Klauzule	7
6. Wzór na obliczanie wag od K1 do K7	7
7. Budowanie klauzul ze zmiennych zdaniowych	8
8. Klasa Solver	9
9. Stworzone endpointy	9
10. Przykład działania wybranego endpointa	10
11. Wyniki	10
12. Instrukcja obsługi	13

1. Projekt / zarys wyglądu aplikacji

Parking spot recommendation system

Do you want to park in a paid parking lot?

☒ Yes ☐ No

Do you want to park in a guarded parking lot?

☒ Yes ☐ No

Do you want to park in a disabled parking?

☒ Yes ☐ No

Do you want to park in a car park where at least 15 spaces are available?

☒ Yes ☐ No

Do you want to park in a private parking lot?

☒ Yes ☐ No

Do you want to park in the parking lot with an electric car?

☒ Yes ☐ No

Choose a place where you want to park



SUBMIT

Parking spot recommendation system

Selected parking space for you

Parking ID	Score
S1	3
S6	12
S12	20

2. Baza danych

parking_lot	zone
count_of_picks integer	accessibility_factor double
free_spaces integer	attractiveness_factor double
have_space_for_handicapped integer	cordx integer
have_spaces_for_electrics integer	cordy integer
is_guarded integer	demand_factor double
is_paid integer	zone_id integer
is_private integer	
zone_id integer	
parking_lot_id integer	

Tabela ParkingLot zawiera szczegółowe informacje dotyczące parkingu, takie jak posiadanie specjalnych miejsc postojowych dla niepełnosprawnych czy koszt pozostawienia samochodu. Oprócz tego przechowuje informacje o numerze strefy, w której się znajduje.

Przykładowe dane dla tabeli Parking Lot (pierwsze dziesięć wierszy):

parking_lot_id	count_of_picks	free_spaces	have_space_for_handicapped	have_spaces_for_electrics	is_guarded	is_paid	is_private	zone_id
1	0	6	1	1	0	0	1	1
2	0	5	1	1	1	1	0	1
3	0	2	1	1	1	0	1	1
4	0	8	0	1	0	1	1	1
5	0	2	0	0	1	0	0	2
6	0	0	0	1	1	1	0	2
7	0	4	0	0	1	1	1	3
8	0	1	1	0	0	1	1	3
9	0	9	1	0	0	0	0	3
10	0	0	1	1	0	1	1	4

Tabela Zone zawiera informacje o wartości współczynnika popytowego na samochody w danej strefie, współczynnik atrakcyjności strefy oraz współczynnik dostępności dla osób niepełnosprawnych.

Przykładowe dane dla tabeli Zone (pierwsze dziesięć wierszy):

zone_id	accessibility_factor	attractiveness_factor	cordx	cordy	demand_factor
1	0,78	0,12	0	1	0,56
2	0,22	0,98	0	2	0,47
3	0,89	0,84	1	0	0,52
4	0,42	0,54	1	1	0,94
5	0,05	0,76	1	2	0,93
6	0,09	0,22	2	0	0,25
7	0,8	0,59	2	1	0,81
8	0,26	0,85	2	2	0,2
9	0,28	0,74	2	3	0,6
10	0,36	0,63	2	4	0,14

3. Wykorzystane technologie

- Java - Spring Framework, wraz z wykorzystaniem biblioteki SAT4J
- Javascript
- HTML
- SASS / CSS
- SQLite
- JPA

4. Zmienne zdaniowe

Z1	parking znajduje się w strefie nr 1
Z2	parking znajduje się w strefie nr 2
Z3	parking znajduje się w strefie nr 3
Z4	parking znajduje się w strefie nr 4
Z5	parking znajduje się w strefie nr 5
Z6	parking znajduje się w strefie nr 6
Z7	parking znajduje się w strefie nr 7
Z8	parking posiada miejsca dla niepełnosprawnych
Z9	parking jest prywatny
Z10	parking jest płatny
Z11	parking jest strzeżony
Z12	parking ma miejsca dla aut elektrycznych
Z13	na parkingu jest przynajmniej 10 wolnych miejsc parkingowych
Z14	parking jest wśród trzech ulubionych klienta

5. Klauzule

K0	solver ma wybrać najkorzystniejszą strefę, w której ma znajdować się rekomendowany parking
K1	solver ma nie wybierać strefy nr 1
K2	solver ma nie wybierać strefy nr 2
K3	solver ma nie wybierać strefy nr 3
K4	solver ma nie wybierać strefy nr 4
K5	solver ma nie wybierać strefy nr 5
K6	solver ma nie wybierać strefy nr 6
K7	solver ma nie wybierać strefy nr 7
K8	Klient preferuje parkingi z miejscami dla osób niepełnosprawnych
K9	Klient preferuje parkingi prywatne
K10	Klient preferuje parkingi płatne
K11	Klient preferuje parkingi strzezone
K12	Klient preferuje parkingi z miejscami dla aut elektrycznych
K13	Klient preferuje parkingi z wieloma wolnymi miejscami
K14	solver ma wybierać ten parking

6. Wzór na obliczanie wag od K1 do K7

$\text{weight} = \text{Math.round}((\text{demandFactor} + \text{accessibilityFactor} + \text{attractivenessFactor}) / 0.03)$
--

7. Budowanie klauzul ze zmiennych zdaniowych

$$K0 \rightarrow Z1 \vee Z2 \vee Z3 \vee Z4 \vee Z5 \vee Z6 \vee Z7 (10000)$$

$$K1 \rightarrow \neg Z1 \text{ (priorytet strefy nr 1)}$$

$$K2 \rightarrow \neg Z2 \text{ (priorytet strefy nr 2)}$$

$$K3 \rightarrow \neg Z3 \text{ (priorytet strefy nr 3)}$$

$$K4 \rightarrow \neg Z4 \text{ (priorytet strefy nr 4)}$$

$$K5 \rightarrow \neg Z5 \text{ (priorytet strefy nr 5)}$$

$$K6 \rightarrow \neg Z6 \text{ (priorytet strefy nr 6)}$$

$$K7 \rightarrow \neg Z7 \text{ (priorytet strefy nr 7)}$$

$$K8 \rightarrow Z8 \wedge Z13 (35, 30)$$

$$\neg K8 \rightarrow \neg Z8 \wedge (\neg Z11 \vee Z9) (30, 20)$$

$$K9 \rightarrow Z9 \wedge Z11 \wedge Z10 (30, 25, 10)$$

$$\neg K9 \rightarrow \neg Z9 \wedge (\neg Z13 \vee \neg Z11) (25, 20)$$

$$K10 \rightarrow Z10 \wedge Z11 \wedge Z9 (20, 15, 10)$$

$$\neg K10 \rightarrow \neg Z10 \wedge (\neg Z13 \vee \neg Z11) (30, 15)$$

$$K11 \rightarrow Z11 \wedge Z9 \wedge Z10 (30, 25, 10)$$

$$\neg K11 \rightarrow \neg Z11 \wedge (\neg Z13 \vee \neg Z9) (20, 15)$$

$$K12 \rightarrow Z12 \wedge Z13 (30, 25)$$

$$\neg K12 \rightarrow \neg Z12 \wedge (Z9 \vee Z10 \vee \neg Z13) (20, 15)$$

$$K13 \rightarrow Z13 \wedge (Z10 \wedge Z11 \wedge Z9) (15, 10)$$

$$\neg K13 \rightarrow \neg Z13 (30)$$

$$K14 \rightarrow K14 (25)$$

8. Klasa Solver

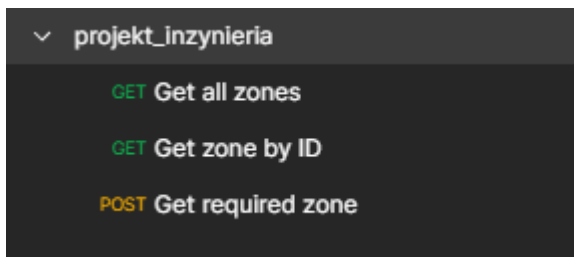
Klasa reprezentuje ważony MaxSat solver będący głównym silnikiem systemu.

Oczywistym jest, że nie zawsze istniał będzie idealny parking dla każdego użytkownika, w związku z czym system powinien wybrać ten najbliższy ideałowi. Działanie silnika problemu rekomendacji miejsc parkingowych można przedstawić następujących krokach:

1. Utworzenie odpowiedniej ilości zmiennych zdaniowych w zależności od danych pobranych od użytkownika.
2. Zbudowanie klauzul złożonych ze stworzonych wcześniej literałów.
3. Analiza odpowiedzi ważonego Max-SAT solvera oraz mapowanie jej na stosunek polecenia danego parkingu w celu wybrania najlepszych dla klienta aplikacji.

Wynikiem solvera jest lista trzech parkingów z najlepszym wynikiem wraz z ilością punktów jakie uzyskały.

9. Stworzone endpointy



Get all zones ("/zones") zwraca wszystkie strefy z bazy danych.

Get zone by ID ("/zones/{id}") zwraca tylko jedną strefę o podanym id.

Get required zone ("/requiredparkinglot") zwraca id trzech najlepszych parkingów wraz z wynikiem dla podanych koordynatów i preferencji.

10. Przykład działania wybranego endpointa

```
projekt_inzynieria / Get all zones
GET http://localhost:8081/zones
Status: 200 OK Time: 49 ms Size: 22.82 KB
Save Response
Pretty Raw Preview Visualize JSON
1 {
2   {
3     "zoneId": 1,
4     "coordX": 0,
5     "coordY": 1,
6     "demandFactor": 0.43,
7     "accessibilityFactor": 0.97,
8     "attractivenessFactor": 0.43,
9     "parkingLots": [
10      {
11        "parkingLotId": 1,
12        "haveSpaceForHandicapped": false,
13        "isGuarded": false,
14        "isPaid": false,
15        "freeSpaces": 36,
16        "isPrivate": true,
17        "haveSpacesForElectrics": true,
18        "private": true,
19        "guarded": false,
20        "paid": false
21      }
22    ]
23  },
24  {
25    "zoneId": 2,
26    "coordX": 0,
27    "coordY": 2,
28    "demandFactor": 0.71,
29    "accessibilityFactor": 0.29,
30    "attractivenessFactor": 0.36,
31    "parkingLots": [
32      {
33        "parkingLotId": 5,
34        "haveSpaceForHandicapped": true,
35        "isGuarded": false,
36        "isPaid": true,
37        "freeSpaces": 13,
38        "isPrivate": false,
```

W powyższym przykładzie użyto endpoint Get all zones ("/zones"). Wynikiem jest lista wszystkich stref. Każda strefa ma unikatowy id, koordynaty, wartości poszczególnych współczynników oraz listę parkingów znajdujących się w niej.

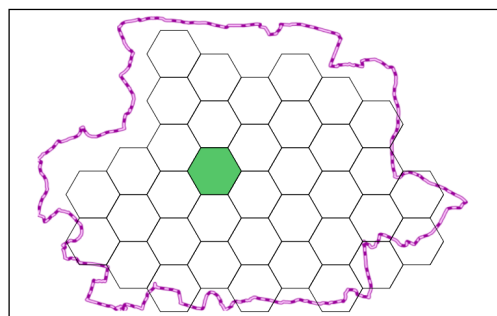
11. Wyniki

a) wejście:

Parking spot recommendation system

Choose a place where you want to park

- Do you want to park in a paid parking lot?
☒ Yes ☐ No
- Do you want to park in a guarded parking lot?
☐ Yes ☒ No
- Do you want to park in a disabled parking?
☒ Yes ☐ No
- Do you want to park in a car park where at least 15 spaces are available?
☒ Yes ☐ No
- Do you want to park in a private parking lot?
☒ Yes ☐ No
- Do you want to park in the parking lot with an electric car?
☐ Yes ☒ No



Find best parking

wyjście:

Parking spot recommendation system

Selected parking space for you

Parking Id	Score
36	15
35	14
37	13

b) wejście

Parking spot recommendation system

Choose a place where you want to park

Do you want to park in a paid parking lot?

☒ Yes ☐ No

Do you want to park in a guarded parking lot?

☐ Yes ☒ No

Do you want to park in a disabled parking?

☐ Yes ☒ No

Do you want to park in a car park where at least 15 spaces are available?

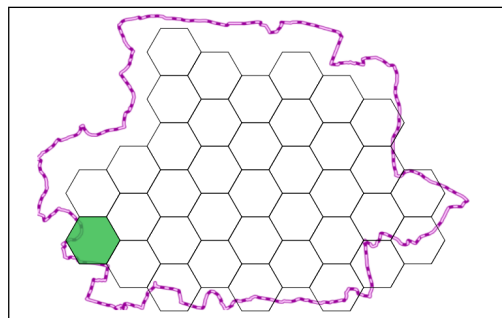
☒ Yes ☐ No

Do you want to park in a private parking lot?

☒ Yes ☐ No

Do you want to park in the parking lot with an electric car?

☐ Yes ☒ No



Find best parking

wyjście:

Parking spot recommendation system

Selected parking space for you

Parking Id	Score
1	13
7	6
3	4

c) wejście:

Parking spot recommendation system

Choose a place where you want to park

Do you want to park in a paid parking lot?

☐ Yes ☒ No

Do you want to park in a guarded parking lot?

☐ Yes ☒ No

Do you want to park in a disabled parking?

☐ Yes ☒ No

Do you want to park in a car park where at least 15 spaces are available?

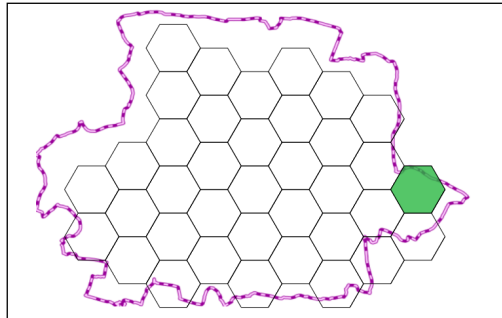
☒ Yes ☐ No

Do you want to park in a private parking lot?

☐ Yes ☒ No

Do you want to park in the parking lot with an electric car?

☐ Yes ☒ No



Find best parking

wyjście:

Parking spot recommendation system

Selected parking space for you

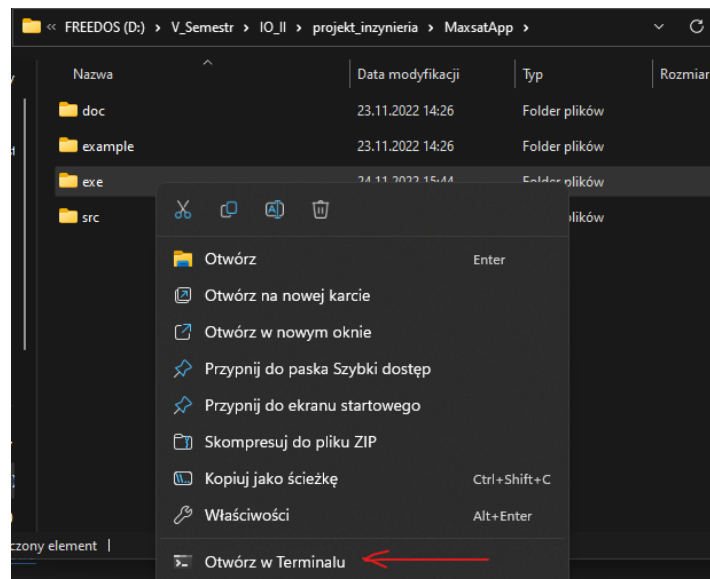
Parking Id	Score
93	16
94	13
92	12

12. Instrukcja obsługi

Należy rozpakować plik **MaxsatApp.zip**. Najpierw powinno zostać uruchomione api, następnie strona w HTMLu, odpowiadająca za wyświetlanie wyników

a) uruchomienie API:

- przejść do folderu **MaxsatApp/** i nacisnąć prawym przyciskiem myszy na folder **exe**, po czym wybrać opcję **“Otwórz w Terminalu”**



- w terminalu wpisać komendę **java -jar maxsatApi-0.0.1-SNAPSHOT.jar**

```
PS D:\V_Semestr\IO_II\projekt_inzynieria\MaxsatApp\exe> java -jar maxsatApi-0.0.1-SNAPSHOT.jar

  ____ _
 / ___| | | |
 \___ \| |_| |
  ___) |  __/ | | |
 |____|_| |_| |
:: Spring Boot :: (v2.7.4)

2022-11-24 15:49:21.511 INFO 12464 --- [ main] c.e.maxsatApi.MaxsatApiApplication : Starting MaxsatApiApplication v0.0.1-SNAPSHOT using Java 17.0.5 on DESKTOP-HB3POAI with PID 12464 (D:\V_Semestr\IO_II\projekt_inzynieria\MaxsatApp\exe\maxsatApi-0.0.1-SNAPSHOT.jar started by broni in D:\V_Semestr\IO_II\projekt_inzynieria\MaxsatApp\exe)
2022-11-24 15:49:21.516 INFO 12464 --- [ main] c.e.maxsatApi.MaxsatApiApplication : No active profile set, falling back to 1 default profile: "default"
2022-11-24 15:49:22.606 INFO 12464 --- [ main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2022-11-24 15:49:22.678 INFO 12464 --- [ main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 60 ms. Found 2 JPA repository interfaces
```

- aplikację zamyka się kombinacją klawiszy **ctrl + c**

b) uruchomienie aplikacji za pomocą przeglądarki internetowej:

- przejść do folderu **MaxsatApp / exe / front end**, a następnie uruchomić plik **index.html** w przeglądarce internetowej

-po otwarciu aplikacji zaznaczyć pola wyboru zgodnie ze swoimi preferencjami dotyczącymi parkowania:



Parking spot recommendation system

Choose a place where you want to park

Do you want to park in a paid parking lot?
☒ Yes ☐ No

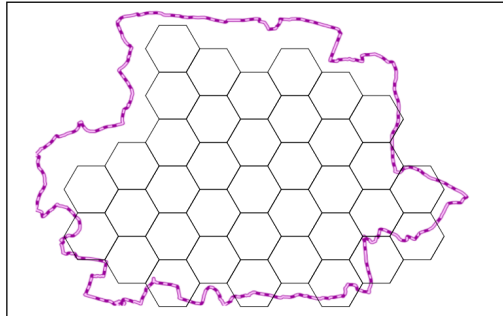
Do you want to park in a guarded parking lot?
☐ Yes ☒ No

Do you want to park in a disabled parking?
☒ Yes ☐ No

Do you want to park in a car park where at least 15 spaces are available?
☐ Yes ☒ No

Do you want to park in a private parking lot?
☐ Yes ☒ No

Do you want to park in the parking lot with an electric car?
☒ Yes ☐ No



Find best parking

-następnie wybrać na mapie miasta interesujący nas sektor za pomocą kliknięcia:

Parking spot recommendation system

Choose a place where you want to park

Do you want to park in a paid parking lot?
☒ Yes ☐ No

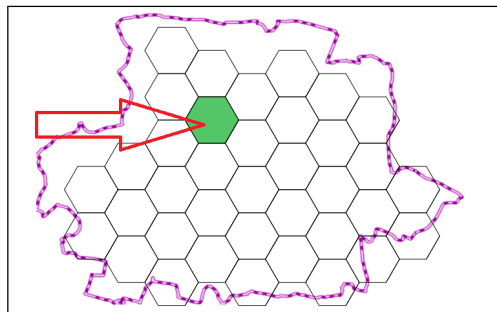
Do you want to park in a guarded parking lot?
☐ Yes ☒ No

Do you want to park in a disabled parking?
☒ Yes ☐ No

Do you want to park in a car park where at least 15 spaces are available?
☐ Yes ☒ No

Do you want to park in a private parking lot?
☐ Yes ☒ No

Do you want to park in the parking lot with an electric car?
☒ Yes ☐ No



Find best parking

-na końcu, aby wyświetlić wynik działania programu, należy kliknąć przycisk: **“Find best parking”**:

Parking spot recommendation system

Choose a place where you want to park

Do you want to park in a paid parking lot?

☒ Yes ☐ No

Do you want to park in a guarded parking lot?

☐ Yes ☒ No

Do you want to park in a disabled parking?

☒ Yes ☐ No

Do you want to park in a car park where at least 15 spaces are available?

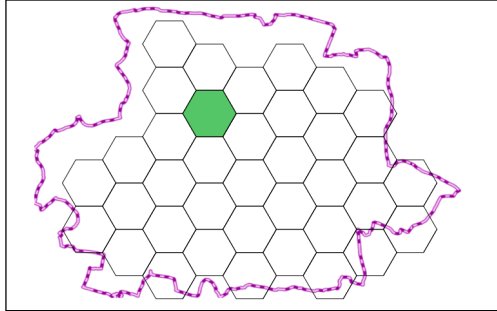
☐ Yes ☒ No

Do you want to park in a private parking lot?

☐ Yes ☒ No

Do you want to park in the parking lot with an electric car?

☒ Yes ☐ No



Find best parking

Literatura:

- 1) <http://sat4j.org>
- 2) Praca dyplomowa, Autor: Paweł Hanzlik, temat: "Projekt i implementacja systemu rekomendacji miejsc do parkowania w usłudze współdzielenia samochodów".
- 3) "Maximum Satisfiability", Autor: Phil Sung, 17 Marca 2006.