

Komandno-upravljački informacioni sistem

Student: Teodora Kocić



Sadržaj

Uvod

Implementacija

Korišćenje aplikacije

Literatura

Uvod

Aplikacija je napravljena u cilju da korisnicima olakša traženje slobodnih parking mesta u gradu. Ovo se postiže tako što se svaki korisnik, koji se nalazi u određenoj zoni grada, obaveštava o tome da je novo vozilo parkirano na nekom od parkinga koji se nalazi u delu grada koji je od interesa za datog korisnika. Informacija koja se ovim putem šalje svim korisnicima, koji se nalaze u zoni grada gde je pristiglo novo vozilo, sadrži naziv datog parkinga, odnosno njegove prostorne coordinate (latituda i longituda), kao i broj preostalih slobodnih mesta na datom parkingu. Nakon sat vremena korisnicima ističe dozvoljeno vreme zadržavanja na parkingu.



Implementacija

Korišćene
tehnologije

Tehnologije – pribavljanje podataka



Tehnologije – komunikacija i vizualizacija

Obezbeđuje pokretanje mosquitto broker-a i prevođenje same python aplikacije u kontejner.



Kod logike cele aplikacije napisan je u programskom jeziku Python. Rezultat izvršenja može se pratiti u terminalu nakon pokretanja aplikacije.

Message broker kojim se obezbeđuje komunikacija u okviru aplikacije. Za svaku zonu grada koja postoji u okviru filtriranih podataka kreira se **topic**. Kada se korisnik nalazi u određenom delu grada (simulira se mobilnost korisnika kroz gps koordinate iz dataset-a) pretplaćuje se na odgovarajući topic. Kada korisnik zauzme određeno parking mesto poruka o tome se šalje svim korisnicima koji su pretplaćeni na isti topic kao i taj korisnik. Korisnik nakon isteka parkinga biva unsubscribe-ovan sa datog topic-a.

Radi lakšeg i preglednijeg rezultata izvršenja aplikacije može se koristiti softver MQTT Explorer gde se jasno vidi kada se korisnik pretplati na odgovarajući topic. Prate se i trenuci slanja/primanja poruka, pa se tako jasno vide neke od pomenutih **funkcionalnosti aplikacije** (praćenje broja slobodnih parking mesta, istek parkinga za datog korisnika, itd.)

Bitni delovi implementacije

Čitanje dataset-ova

```
35 ∨ def parking_spaces_occupied():
36 ∨     with open('client\\parking_capacity.csv', 'r') as file:
37         csv_reader = csv.DictReader(file)
38 ∨     for row in csv_reader:
39         capacity.append(int(row['Capacity']))
40         occupancy.append(int(row['Occupancy']))
41
42
43 ∨ def define_regions_of_parkings():
44 ∨     with open('client\\user_gps.csv', 'r') as file:
45         csv_reader = csv.DictReader(file)
46 ∨     for row in csv_reader:
47         municipality.add(row['Municipality'])
48 ∨     for id in range(len(location)):
49         parking_region[id] = random.choice(list(municipality))
```

Prikupljanje podataka iz baze

```
connection.autocommit = True
postgre_fetch_parkings = \
'SELECT osm_id AS parkingID, name AS parking, ST_X(ST_Transform(ST_Centroid(way), 4326)) as longitude, \
    ST_Y(ST_Transform(ST_Centroid(way), 4326)) as latitude FROM public.planet_osm_polygon_mv WHERE \
    ST_X(ST_Transform(ST_Centroid(way), 4326)) BETWEEN -74.0 AND -73.0 AND \
    ST_Y(ST_Transform(ST_Centroid(way), 4326)) BETWEEN 42.0 AND 44.0'

cursor = connection.cursor()
cursor.execute('SELECT %s as connected;', ('Connection to postgres successfull',))
print(cursor.fetchone())
cursor.execute(postgre_fetch_parkings)
record = cursor.fetchall()
```

Manipulacija broker-om

```
client.loop_start()
client.subscribe(topic)
record[client] = topic

if free_space <= 0:
    payload = 'New vehicle in the area ' + parking_region[index] + \
        '. The vehicle entered parking ' + name[index] + ' with coordinates: \
        { latitude: ' + str(location[index][0][0]) + '\t longitude: ' + \
        str(location[index][0][1]) + ' }. No more free parking spaces left within this parking area!'
else:
    payload = 'New vehicle in the area ' + parking_region[index] + \
        '. The vehicle entered parking ' + name[index] + ' with coordinates: { latitude: ' \
        + str(location[index][0][0]) + '\t longitude: ' + str(location[index][0][1]) + ' }. \
        Number of free parking spaces in this parking area is now ' + str(free_space) + '!'

client.publish(topic, payload)

expiring_parking()
```

Pokretanje aplikacije

1. Pokretanje docker-a

- Pozicionirajte se u folder gde se nalazi *docker-compose* fajl
- **docker-compose up -d**
- **python client/app.py**

2. Startovanje MQTT Explorer-a

- Otvorite aplikaciju
- Osluškuje adresu **mqtt://localhost:1883**

Korišćenje aplikacije

Demo video rada aplikacije

EXPLORER

OPEN EDITORS

main.py client M

KOMANDNO-UPRAVLJACKI IS

client

.env

Dockerfile

main.py M

parking_capacity.csv

requirements.txt

user_gps.csv

conf

mosquito.conf

~\$Prezentacija.pptx U

Commands for postgres...

docker-compose.yml

Prezentacija.pptx U

transport_new_york.qgz

main.py M X

client > main.py > generate_gps_coordinates

171 record[client] = topic

172

173

174 if free_space <= 0:

175 payload = 'New vehicle in the area ' + parking_region[index] + \

176 '. The vehicle entered parking ' + name[index] + ' with coordinates: \

177 { latitude: ' + str(location[index][0][0]) + '\t longitude: ' + \

178 str(location[index][0][1]) + ' }. No more free parking spaces left within this parking area!'

179 else:

180 payload = 'New vehicle in the area ' + parking_region[index] + \

181 '. The vehicle entered parking ' + name[index] + ' with coordinates: { latitude: ' \

182 + str(location[index][0][0]) + '\t longitude: ' + str(location[index][0][1]) + ' }. \

183 Number of free parking spaces in this parking area is now ' + str(free_space) + '!'

184

185 client.publish(topic, payload)

186

187 expiring_parking()

188

189 print('-'*50 + 'new vehicle approaching' + '-'*50 + '\n')

190

191 sleep(random.randint(1,5))

192

193 if __name__ == '__main__':

194

195 postgres_connection_and_queries()

196 generate_gps_coordinates()

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

python + - - -

Connecting to broker ...

...done

Connected OK with result code 0

Subscribed: 1 (0,)

Received message on topic parking/GRAFTON: New vehicle in the area GRAFTON. The vehicle entered parking parking_109003663 with coordinates: { latitude: -73.67682647705078

longitude: 42.77473831176758 }.

Number of free parking spaces in this parking area is now 556!

-----new vehicle approaching-----



Literatura

1. Spatial DataBases With Application To GIS (The Morgan Kaufmann Series in Data Management Systems), Philippe Rigaux, Michel Scholl, Agnes Voisard, Morgan Kaufmann Publishers, 2002.
2. Hands-On MQTT Programming with Python: Work with the lightweight IoT protocol in Python, Gaston C. Hillar, Packt Publishing, 2018.