

Modul Real-time Monitoring

Tým PARKety, č. 15

Vedúci projektu: Ing. Ivan Srba, PhD.

Predmet: Tímový projekt I

Ročník: 2017/2018

Vypracoval(i): Martin Hoang, Samuel Púčať

Babinec Peter, Bc. Hoang Martin, Bc.

Hučko Jakub, Bc. Karas Marek, Bc.

Lehotský Miroslav, Bc. Mičo Jakub, Bc.

Púčať Samuel, Bc. Vnenčák Stanislav, Bc.

13. decembra 2017

Obsah

1	Analýza	1
2	Návrh	2
2.1	Návrh UX obrazoviek	2
2.1.1	Obrazovka zobrazenie mapy s parkoviskami	2
2.1.2	Obrazovka zobrazenie detailu parkoviska	3
2.2	Model údajov pre zobrazenie parkoviska	4
3	Implementácia	7
3.1	Komunikácia s Live Objects	7
3.2	Ukladanie stavu senzorov	8
3.3	Vyhľadávanie parkovísk	8
3.3.1	Vyhľadávanie podľa názvu	8
3.3.2	Vyhľadávanie podľa adresy	9
3.4	Vizualizácia mapy	9
3.5	Vizualizácia parkoviska	10
3.6	Implementácia obrazoviek	11
3.6.1	Implementácia domovskej stránky	12
3.6.2	Implementácia obrazovky zobrazenie mapy s vyhľadavacími parkoviskami	12
3.6.3	Implementácia obrazovky zobrazenie parkoviska	13

Zoznam obrázkov

2.1	UX obrazovka pre zobrazenie mapy	3
2.2	UX obrazovka pre zobrazenie parkoviska	4
2.3	Diagram tried pre entitu parkingLot (parkovisko)	6
3.1	Komunikácia s LO platformou	7
3.2	Celkový náhľad na využitú architektúru	8
3.3	Sekvenčný diagram pre zobrazenie mapy	10
3.4	Sekvenčný diagram pre zobrazenie parkoviska	11
3.5	Implementácia domovskej stránky	12
3.6	Implementácia obrazovky zobrazenie mapy	13
3.7	Implementácia obrazovky zobrazenie parkoviska	14

1 Analýza

Modul real-time monitoring sa zaoberá zobrazením informácií v reálnom čase. Umožňuje vyhľadať parkoviská podľa názvu alebo adresy a zobrazíť parkoviská na mape s aktuálnym stavom na parkoviskách. Modul komunikuje so senzormi pomocou MQTT a Live Object API. Túto komunikáciu nám sprostredkováva Orange Slovensko. Ostatné údaje ako sú parking lots (parkoviska), PL img (obrázok parkoviska) a parking lot history (históriu parkoviska) získava z dátového úložiska, presnejšie z MongoDB.

2 Návrh

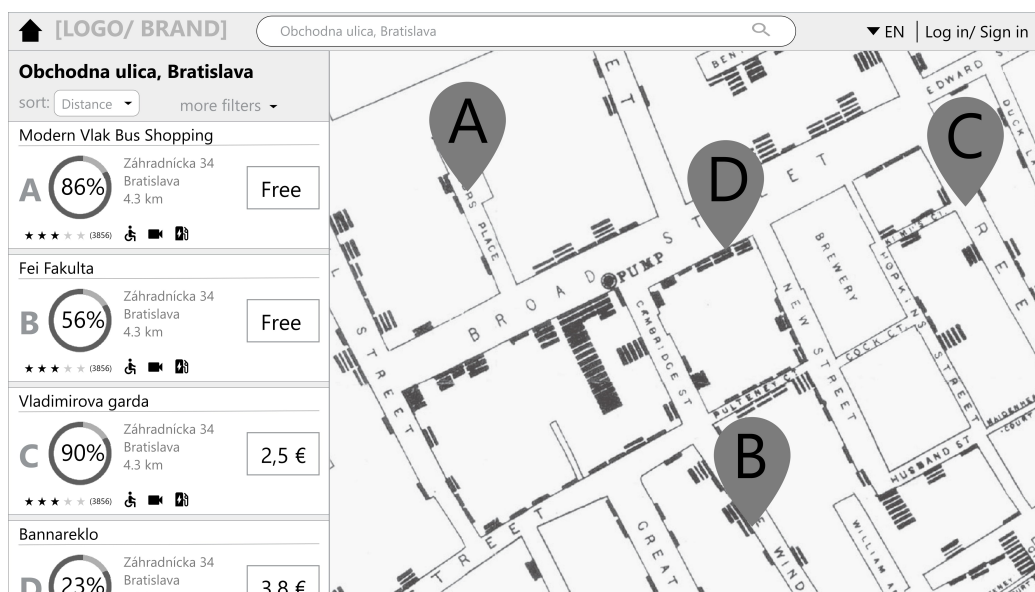
Implementácií predchádza návrh. Navrhovali sme viacero dátových entít ale aj UX obrazovky pre ľahšiu komunikáciu s product ownermi.

2.1 Návrh UX obrazoviek

Pred implementáciou sme začali najprv návrhom UX obrazoviek. Tieto obrazovky majú výhodu, že je možné ich rýchlo nakresliť a pri nejakej nezhode s product ownermi sa dajú jednoducho prerobiť.

2.1.1 Obrazovka zobrazenie mapy s parkoviskami

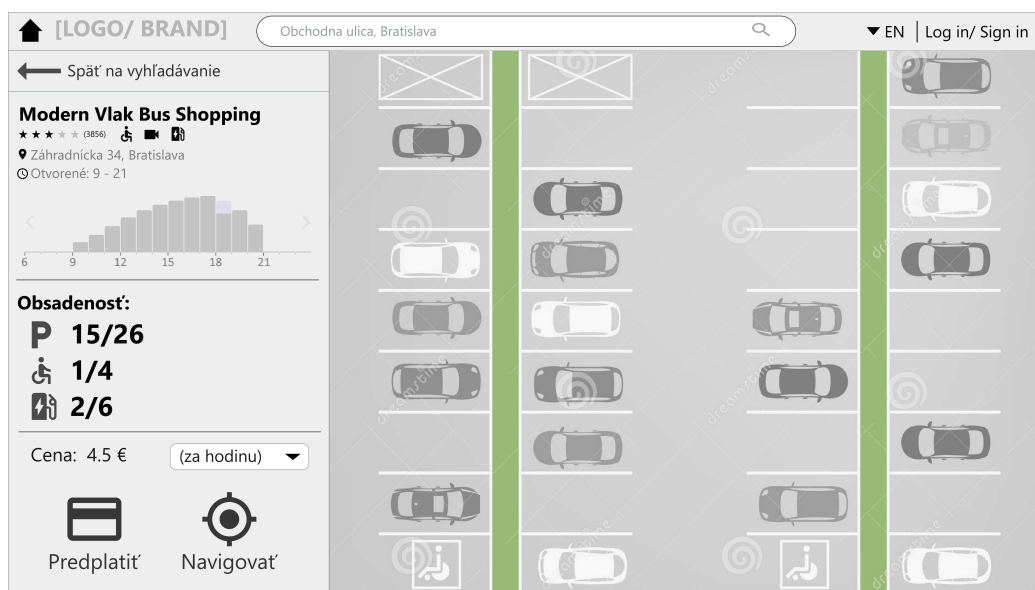
Na obrázku 2.1 môžeme vidieť návrh obrazovky mapy. Návrh obsahuje v hornej časti vyhľadávacie pole pre hľadanie mesta, mestskej časti alebo ulice. V ľavej časti môžeme vidieť panel s vyhľadávaným pojmom a zobrazené dostupné parkoviská zoradené podľa filtra, v tomto prípade od vzdialenosti od vyhľadávaného pojmu. V strede obrazovky sa nachádza mapa na ktorej sú zobrazené piny s parkoviskami. Príslušné písmeno značí, o ktoré parkovisko ide v ľavom paneli.



Obr. 2.1: UX obrazovka pre zobrazenie mapy

2.1.2 Obrazovka zobrazenie detailu parkoviska

Na obrázku 2.2 môžeme vidieť návrh obrazovky pre zobrazenie detailu parkoviska. Návrh znova obsahuje v hornej časti vyhľadávacie pole. V ľavej časti sa nám zmenil obsah na informácie o konkrétnom parkovisku. Môžeme tu vidieť obsadenosť v čase, otváracie hodiny, aktuálnu obsadenosť... V strede sa nachádza vizualizácia konkrétného parkoviska, ktorá nám zobrazuje voľné a obsadené parkovacie plochy.



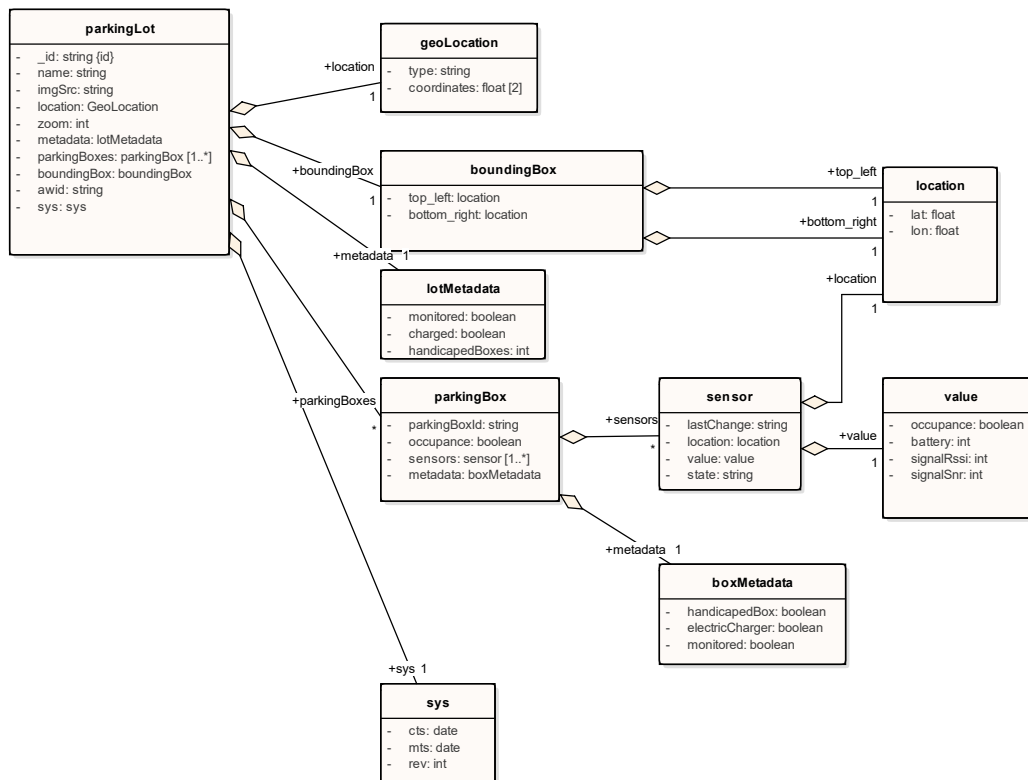
Obr. 2.2: UX obrazovka pre zobrazenie parkoviska

2.2 Model údajov pre zobrazenie parkoviska

Pre ukladanie stavu parkovísk sa používa kolekcia *parkingLot*. Každý záznam parkoviska má nasledovné údaje (viď. obr.: 2.3):

- *name* - názov parkoviska
- *imgSrc* - referencia na obrázok parkoviska
- *location* - gejson pozície parkoviska
- *zoom* - atribút pre nastavenie vzdialenosti pohľadu na parkovisko
- *metadata* - doplnkové údaje o parkovisku:
 - *monitored* - či je parkovisko monitorované
 - *charged* - či parkovisko poskytuje prípojku na elektrinu
 - *handicapedBoxes* - počet miest pre hendikepovaných
- *parkingBoxes* - zoznam parkovacích miest s údajmi:

- *parkingBoxId* - identifikátor parkovacieho miesta
- *occupance* - či je miesto obsadené
- *sensors* - hash mapa senzorov inštalovaných na parkovisku s údajmi:
 - * *lastChange* - timestamp poslednej zmeny
 - * *location* - geografická pozícia senzora
 - * *value*
 - *occupance* - obsadenosť parkovacieho miesta na ktorom je senzor inštalovaný
 - *battery* - stav batérie senzora
 - *signalRssi* - sila signálu senzora (received signal strength indicator)
 - *signalSnr* - pomer signálu k šumu (signal to noise ratio)
 - * *state* - stav senzora
- *metadata* - doplnkové údaje:
 - *boundingBox* - geografické pozície pre umiestnenie obrázka parkoviska (ľavý horný a pravý dolný roh)
 - *awid*
 - *sys*

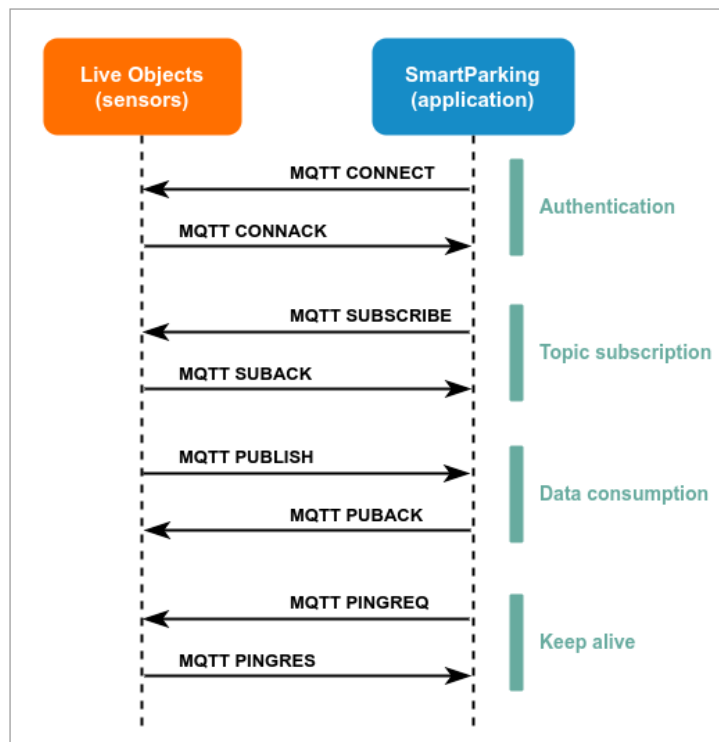


Obr. 2.3: Diagram tried pre entitu parkingLot (parkovisko)

3 Implementácia

3.1 Komunikácia s Live Objects

Komunikácia backendu so senzormi bola implementovaná použitím knižnice MQTT.js. Priebeh komunikácie medzi aplikáciou (klientom) a platformou Live Objects zobrazuje nasledovný diagram (Obr. 3.1):

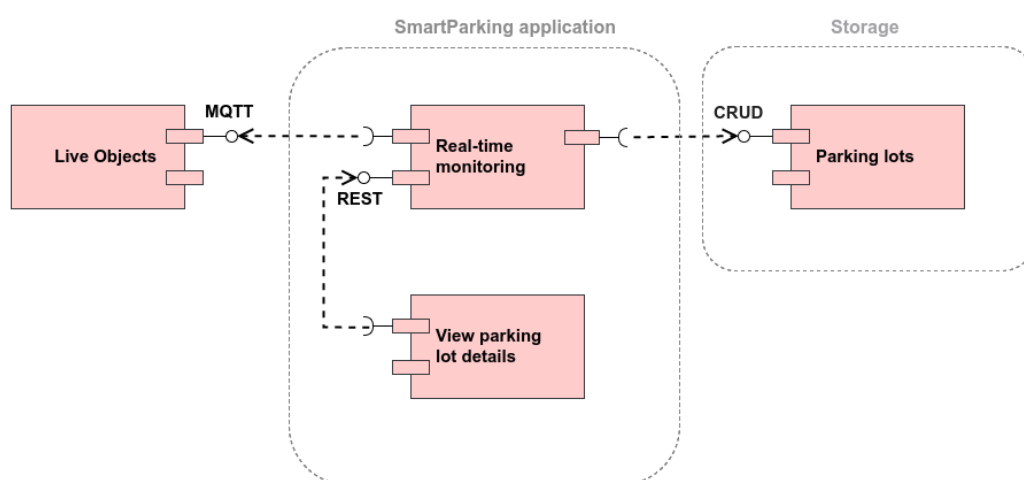


Obr. 3.1: Komunikácia s LO platformou

Komunikácia bola v projekte implementovaná v rámci skriptu `mqtt/mqtt-client.js`. Tento skript exportuje metódu `connectAndSub()`, ktorá vytvára spojenie s Live Objects platformou a zabezpečuje prihlásenie sa na odber, a to vždy pri spustení aplikačného servera.

3.2 Ukladanie stavu senzorov

Pre ukladanie prichádzajúcich zmien do MongoDB sme implementovali API endpoint `controller/updateParkingLotSensor()`. Ten je volaný automaticky pri každej novej zachytenej správe o zmene stavu senzora. Celkový pohľad na architektúru využitú pri implementácii zobrazuje nasledovný diagram (Obr. 3.2):



Obr. 3.2: Celkový náhľad na využitú architektúru

3.3 Vyhľadávanie parkovísk

Na vyhľadanie parkovísk sa použil komponent `<Select/>` z knižnice `react-select`. Na získanie názvov s pozíciami parkovísk podľa hľadaného názvu sa používa rest API `getParkingLotsByName`. Na získanie adries podľa hľadanej adresy sa používa `googleAPIGeocode`. Na získanie parkovísk v okruhu hľadaného sa používa rest API `getNearByParkingLots`.

3.3.1 Vyhľadávanie podľa názvu

Keď je vybrané vyhľadávanie podľa názvu a dĺžka textu v poli je rovná 1, odošle sa http request `getParkingLotsByName`, ktorý vráti názvy a pozície parkovísk zodpovedajúce hľadanému reťazcu. Tieto názvy sa zobrazia pod

textovým poľom. Po vybratí jedného z ponúknutých názvov sa zobrazí mapa vycentrovaná na toto parkovisko. Pri zobrazení mapy sa načítajú taktiež všetky parkoviská v okruhu 50km prostredníctvom *getParkingLotsByName*.

3.3.2 Vyhľadávanie podľa adresy

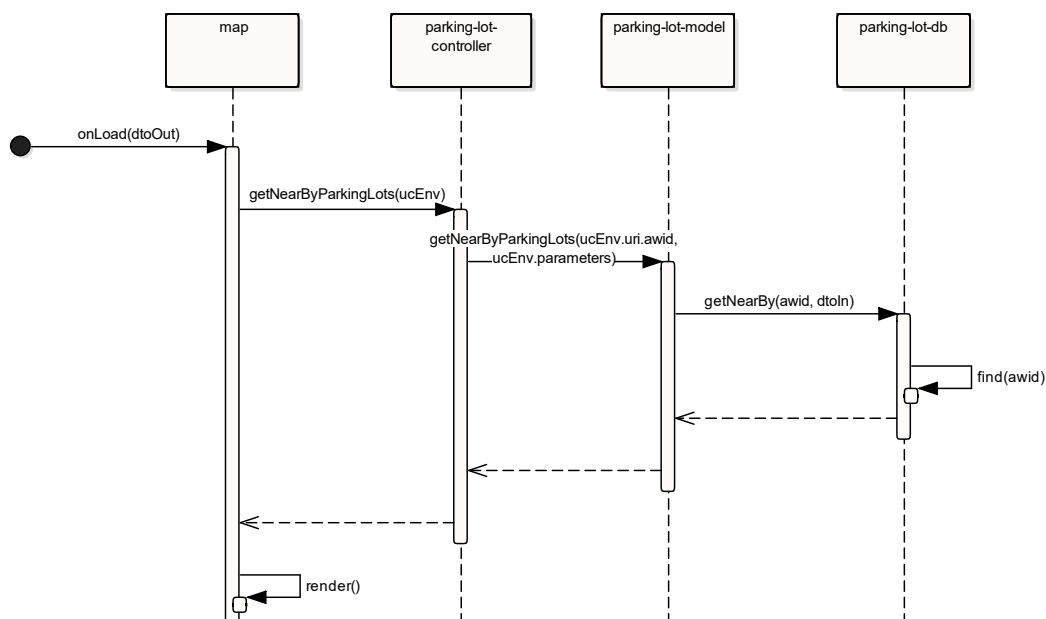
Keď je vybrané vyhľadávanie podľa adresy a zachytí sa zmena v textovom poli vyhľadávača, odošle sa request na *googleAPIGeocode*, ktorý vráti názvy adries/ulíc zodpovedajúcim hľadanému reťazcu. Tieto názvy sa zobrazia pod textovým poľom. Po vybratí jednej z ponúknutých adries/ulíc sa zobrazí mapa vycentrovaná na túto adresu/ulicu. Pri zobrazení mapy sa načítajú taktiež všetky parkoviská v okruhu 50km prostredníctvom *getParkingLotsByName*.

3.4 Vizualizácia mapy

Na mape sa používateľovi zobrazujú markery označujúce pozície parkovísk. Jednotlivé parkoviská si môže používateľ priblížiť kliknutím na príslušný marker.

Pre zobrazenie mapy boli implementované komponenty *<MapComponent/>* a *<GoogleMap/>*. Tieto komponenty taktiež zobrazujú markery na pozíciách parkovísk.

Pri načítaní mapy sa odošle http request na API poskytujúcu funkciu *getNearByPakingLots()*, ktorá vráti údaje o parkoviskách z databázy. Z týchto údajov sa následne vyberú geografické pozície jednotlivých parkovísk a na príslušných pozíciách na mape sa zobrazia markery s odkazmi na zobrazenie konkrétnych parkovísk. (Vid' Obr. 3.3)



Obr. 3.3: Sekvenčný diagram pre zobrazenie mapy

3.5 Vizualizácia parkoviska

Po vybratí parkoviska sa zobrazí mapa s obrázkom parkoviska a indikátormi obsadenosti pre každé parkovacie miesto.

Pre zobrazenie vybraného parkoviska sa okrem komponentov `<MapComponent/>` a `<GoogleMap/>` používa aj komponent `<GroundOverlay/>` zodpovedný za zobrazenie obrázka parkoviska na mape. Taktiež sa na obrázku zobrazujú indikátory obsadenosti (zelené a červené body) pre každé parkovacie miesto.

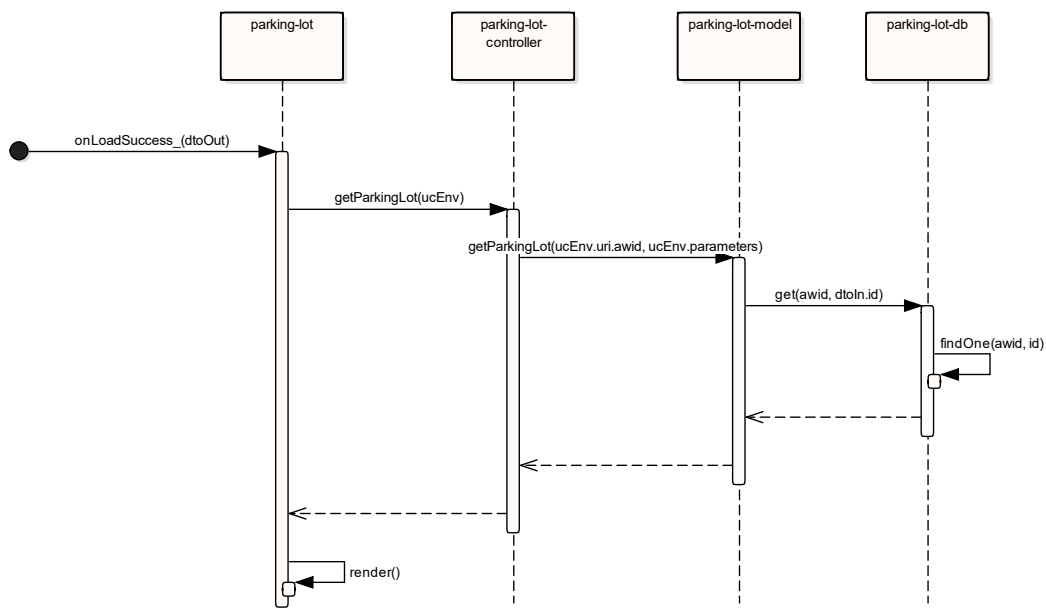
Po kliknutí na marker označujúci parkovisko sa otvorí nové okno, ktoré pri načítaní odošle 2 http requesty:

- `getParkingLot(id)`
- `getImage(id)`

API poskytujúca funkciu `getParkingLot(id)` vráti údaje o parkovisku, medzi ktorými je pole senzorov parkoviska s pozíciami senzorov (*parkin-*

gLot.parkingBoxes[i].sensors[j].location) a obsadenosťou senzorov (*parkingLot.parkingBoxes[i].sensors[j].value.occupance*). Na základe týchto atribútov sú na obrázku zobrazené indikátory (occupance == true červený, occupance == false zelený bod).

API poskytujúca funkciu *getImage(id)* vráti obrázok vo formáte *base64*. Takýto obrázok sa transformuje na formát *png* a zobrazí sa na mape. (Vid' Obr. 3.4)



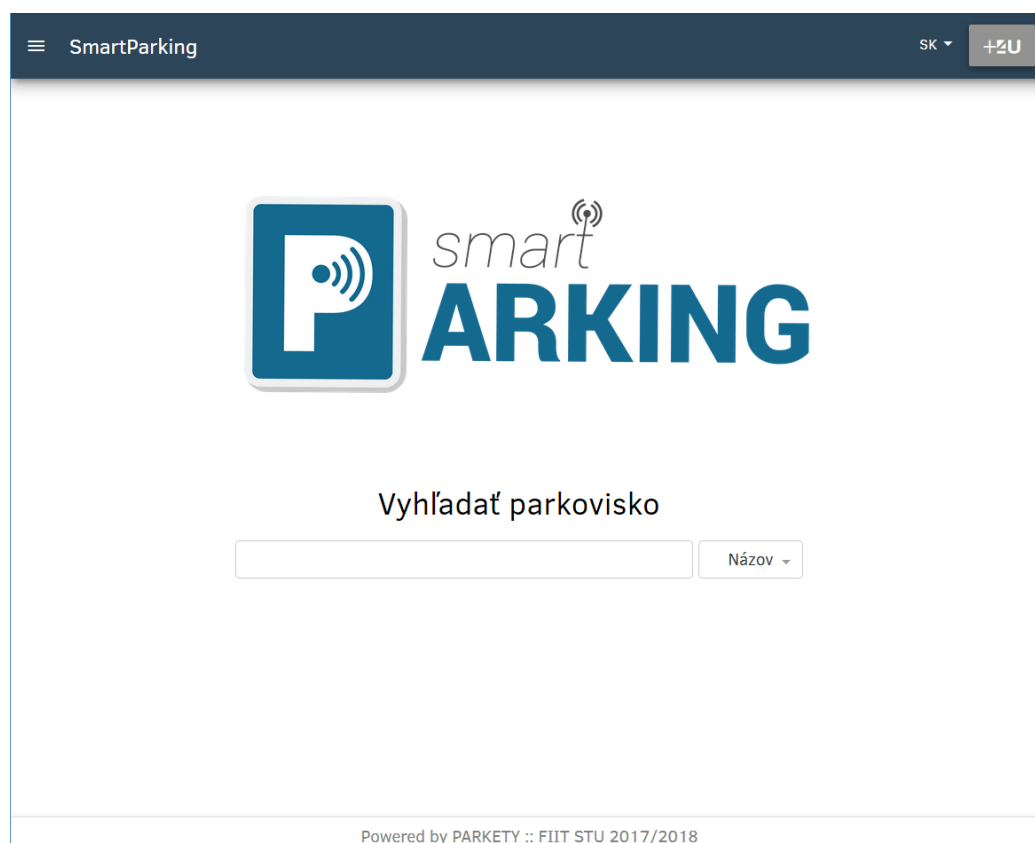
Obr. 3.4: Sekvenčný diagram pre zobrazenie parkoviska

3.6 Implementácia obrazoviek

Pri implementácii obrazoviek sme sa snažili vychádzať čo najviac z navrhnutých obrazoviek. Do určitej miery sa nám to podarilo, avšak je potrebné si uvedomiť, že ide len o prvotné implementácie a v neskorších iteráciách sa obrazovky budú meniť.

3.6.1 Implementácia domovskej stránky

Na obrázku 3.5 môžeme vidieť implementáciu domovskej stránky. Na tejto stránke je možné vyhľadať parkovisko podľa mena alebo adresy.

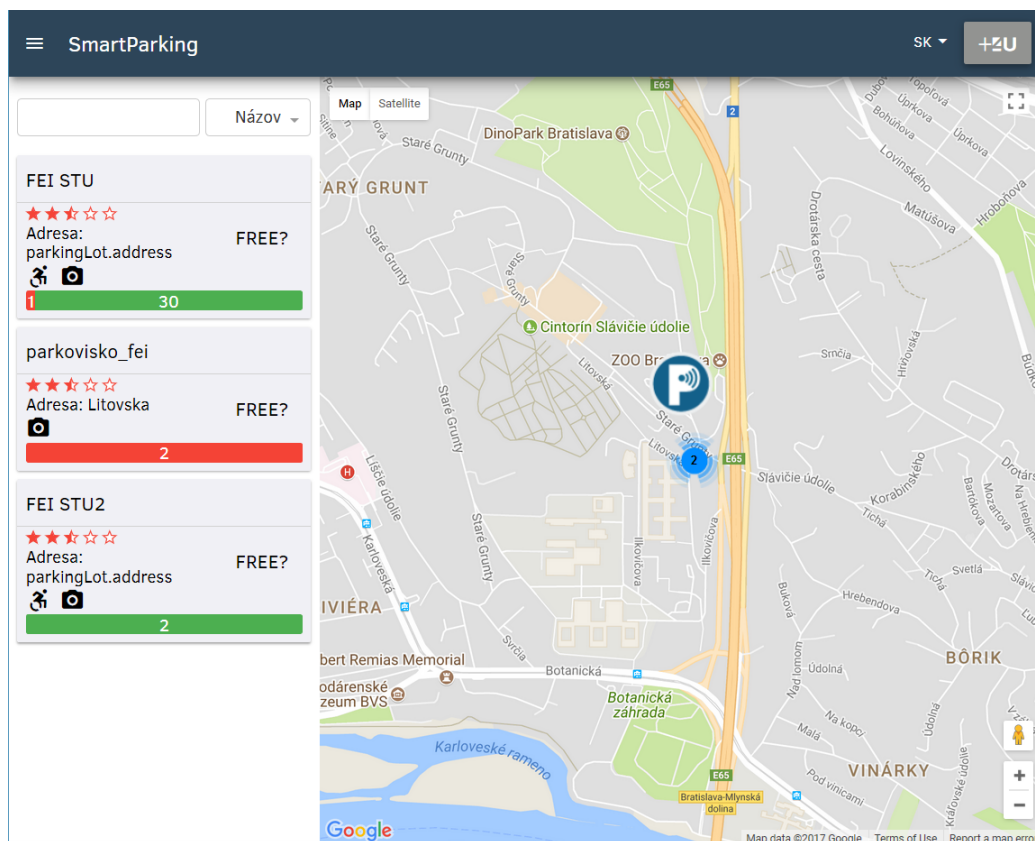


Obr. 3.5: Implementácia domovskej stránky

3.6.2 Implementácia obrazovky zobrazenie mapy s vyhladanými parkoviskami

Na obrázku 3.7 môžeme vidieť implementáciu obrazovky zobrazenie mapy s vyhladanými parkoviskami. V ľavom paneli je vyhľadávacie pole s možnosťami vyhľadávať podľa názvu alebo adresy. Pod vyhľadávacím poľom sa nachádza zoznam viditeľných parkovísk na mape. V strede vidíme zobrazenú mapu na ktorej sú rozmiestnené markery s lokalitami parkovísk. Po kliknutí na marker, alebo na položku zoznamu v ľavom paneli sa nám zobrazí detail konkrét-

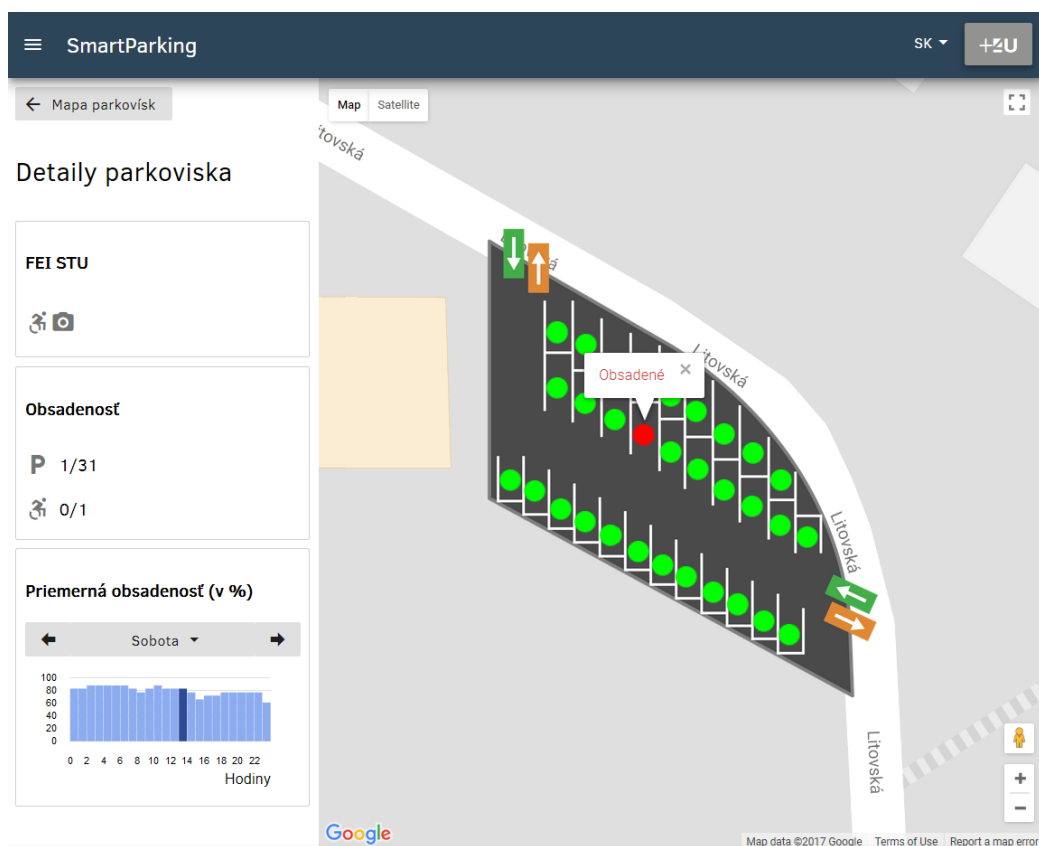
neho parkoviska. Na obrázku 3.7 taktiež vidieť implementovanie zhlukovania markerov.



Obr. 3.6: Implementácia obrazovky zobrazenie mapy

3.6.3 Implementácia obrazovky zobrazenie parkoviska

Na obrázku 3.7 môžeme vidieť implementáciu obrazovky zobrazenie parkoviska. V ľavom paneli sa nachádzajú aktuálne informácie o obsadenosti a štatistiky obsadenosti parkoviska. Môžeme vidieť že sa do určitej miery zhoduje s návrhom na obrázku 2.2. V strede máme zobrazenie konkrétneho parkoviska. Zelené body označujú parkovacie miesta, ktoré sú voľné a červené body parkovacie miesta, ktoré sú obsadené. Po rozkliknutí identifikátorov obsadenosti sa zobrazia informácie príslušného parkovacieho miesta.



Obr. 3.7: Implementácia obrazovky zobrazenie parkoviska