

Dokumentacja projektu – Wypożyczalnia hulajnóg  
wykonywanego w ramach zajęć BAZY DANYCH I  
Gabriela Leśniak

## **I. Projekt koncepcji, założenia**

### **Zdefiniowanie tematu projektu**

Zrealizowanie bazy danych, reprezentującej wypożyczalnię hulajnóg. Hulajnogi znajdują się w różnych lokalizacjach w mieście – hulajnogę można wypożyczyć, jechać długo ile się chce, a następnie zakończyć wypożyczenie.

### **Analiza wymagań użytkownika**

Z bazy korzystają dwa rodzaje użytkowników: klient oraz pracownik

#### **Klient:**

- może założyć swoje konto w wypożyczalni
- może logować się na swoje konto
- może zmienić hasło
- może doładować swój portfel w wypożyczalni
- może wypożyczyć hulajnogę
- przesłać swoją lokalizację, co pozwala zlokalizować dostępne hulajnogi w okolicy
- może zobaczyć historię swoich transakcji
- może zobaczyć ile ma środków na swoim koncie
- może zarezerwować hulajnogę (jeśli zmieni zdanie może zrezygnować z rezerwacji)
- wylogować się

#### **Pracownik:**

- może naładować hulajnogę
- ma podgląd na hulajnogi wymagające ładowania
- ma podgląd na wszystkie hulajnogi
- może serwisować hulajnogę
- ma podgląd na raport z serwisów
- może usunąć hulajnogę
- może zrobić raport serwisu hulajnóg – ile razy hulajnoga była serwisowana i jaki był łączny koszt
- może dodać nową hulajnogę
- może dodać nowego pracownika
- może usunąć klienta
- może zmienić swoje hasło
- może się wylogować

#### **System:**

- z konta klienta ściągane są środki po zakończeniu wypożyczenia
- po zakończeniu wypożyczenia hulajnoga rozładowuje się proporcjonalnie do tego ile czasu była używana
- po doładowaniu konta przez klienta automatycznie zmienia się suma jego środków

### **Zaprojektowanie funkcji**

- podczas tworzenia konta klienta, jego środki automatycznie ustawiane są na zero, a lokalizacja może mieć wartość NULL, ponieważ jest ona potrzebna tylko, kiedy klient chce znaleźć najbliższe hulajnogi
- klient może doładować swój portfel – powoduje to dodanie do bazy nowej transakcji oraz aktualizację dostępnych środków klienta

- klient może wypożyczyć hulajnogę – odnotowywany jest start wypożyczenia. Status hulajnogi zmienia się na „zajęta”. Podczas wypożyczenia klient może zechcieć się zatrzymać – dlatego odnotowywany jest zarówno czas jazdy, jak i czas postojów – są one liczone według różnych stawek. W momencie kiedy klient kończy wypożyczać hulajnogę, czas wypożyczenia się zatrzymuje, a hulajnoga ma status „wolna”.
- kiedy klient zarezerwuje hulajnogę, jej status zmienia się na zarezerwowana
- każda hulajnoga się rozładowuje, jeśli jej poziom naładowania spadnie poniżej poziomu granicznego (5%), jej status zmienia się na „naładuj”
- kiedy pracownik naładuje hulajnogę jej poziom naładowania zmienia się na 100% i ma status „wolna”

## **II. Projekt diagramów (konceptualny)**

### **Zdefiniowanie encji oraz ich atrybutów**

Klienci: klient\_id – klucz główny, imię, nazwisko, email, dostępne\_srodki, dlugosc\_geo, szerokosc\_geo, haslo, aktualne\_wypozyczenie

Transakcje: transakcja\_id – klucz główny, kwota, status\_transakcji, klient\_id

Cennik: uslug\_id – klucz główny, nazwa, cena

Status\_hulajnogi: status\_id – klucz główny, nazwa\_statusu

Hulajnogi: hulajnoga\_id – klucz główny, id\_statusu, dlugosc\_geo, szerokosc\_geo, poziom\_baterii

Pracownicy: pracownik\_id – klucz główny, imię, nazwisko, email, haslo

Serwis\_hulajnogi – serwis\_id – klucz główny, koszt, data, hulajnoga\_id, pracownik\_id

Wypozyczenie: wypozyczenie\_id – klucz główny, klient\_id, hulajnoga\_id, poczatek\_czasu, koniec\_czasu

Czas: czas\_id, wypozyczenie\_id, czas\_start, czas\_stop, uslug\_id

### **Zaprojektowanie relacji pomiędzy encjami**

Relacje te zostały przedstawione na diagramie ERD w sekcji III.

### III. Projekt logiczny

#### Diagram ERD

Tabele i zależności między nimi zostały stworzone z użyciem pliku: 'create\_tables.sql'.



Rysunek 1: Diagram ERD

**Klienci:** tabela jest połączona 1:N z tabelą transakcje, ponieważ jeden klient może wiele razy doładowywać konto o raz mieć z niego ściągane środki, jest też połączona 1:N z tabelą wypozyczenie, ponieważ klient może wypożyczać wiele razy

**Transakcje:** tabela przechowuje informacje o transakcjach klientów

**Hulajnogi:** tabela jest połączona 1:N z tabelą wypożyczenia, ponieważ jedna hulajnoga może być wypożyczana wiele razy oraz 1:N z tabelą serwis\_hulajnogi, ponieważ jedna hulajnoga może być serwisowana wiele razy

**Status\_hulajnogi:** tabela jest połączona 1:N z tabelą hulajnogi, ponieważ każda hulajnoga ma swój status

**Wypożyczenie:** tabela jest połączona 1:N z tabelą czas, ponieważ w trakcie wypożyczenia klient może korzystać z usługi w dwóch trybach (postój i jazda), za które są inne stawki, a koszt jest naliczany od czasu korzystania w danym trybie.

**Cennik:** tabela jest połączona 1:N z tabelą czas, ponieważ każda usługa ma swoją cenę  
**Pracownicy:** tabela przechowuje informacje o pracownikach

**Serwis\_hulajnogi:** przechowuje informacje o serwisowaniu (łądownie to też serwisowanie) wszystkich hulajnóg

**Pracownicy:** tabela jest połączona 1:N z tabelą serwis\_hulajnogi, ponieważ jeden pracownik może wykonać wiele serwisów

## Słowniki danych

*Tabela 1: klienci*

klient_id	serial	NOT NULL	Klucz główny generowany automatycznie
imie	varchar(30)	NOT NULL	
nazwisko	varchar(40)	NOT NULL	
email	varchar(40)	NOT NULL, UNIQUE	
dostepne_srodki	numeric(7,2)	NOT NULL	
dlugosc_geo	numeric		
szerokosc_geo	numeric		
haslo	varchar(40)	NOT NULL	
aktualne_wypozyczenie	integer	NOT NULL	

*Tabela 2: transakcje*

transakcja_id	serial	NOT NULL	Klucz główny generowany automatycznie
kwota	numeric(7,2)	NOT NULL	
status_transakcji	integer	NOT NULL	
klient_id	integer	NOT NULL	Klucz obcy (tabela klienci)

*Tabela 3: cennik*

usluga_id	integer	NOT NULL	Klucz główny
nazwa	varchar	NOT NULL	
cena	numeric(7,2)	NOT NULL	

*Tabela 4: status\_hulajnogi*

status_id	integer	NOT NULL	Klucz główny
nazwa_statusu	varchar	NOT NULL	

*Tabela 5: hulajnogi*

hulajnoga_id	serial	NOT NULL	Klucz główny generowany automatycznie
id_statusu	integer	NOT NULL	Klucz obcy(tabela: status_hulajnogi)
dlugosc_geo	numeric	NOT NULL	
szerokosc_geo	numeric	NOT NULL	
poziom_baterii	integer	NOT NULL	

*Tabela 6: pracownicy*

pracownik_id	serial	NOT NULL	Klucz główny generowany automatycznie
imie	varchar(30)	NOT NULL	
nazwisko	varchar(40)	NOT NULL	
email	varchar(40)	NOT NULL, UNIQUE	
haslo	varchar	NOT NULL	

*Tabela 7: serwis\_hulajnogi*

serwis_id	serial	NOT NULL	Klucz główny generowany automatycznie
koszt	numeric(2)	NOT NULL	
data	date	NOT NULL	
hulajnoga_id	integer	NOT NULL	Klucz obcy(tabela: hulajnogi)
pracownik_id	integer	NOT NULL	Klucz obcy (tabela: pracownicy)

Tabela 8: wypozyczenie

wypozyczenie_id	serial	NOT NULL	Klucz główny generowany automatycznie
klient_id	integer	NOT NULL	Klucz obcy (tabela: klienci)
hulajnoga_id	integer	NOT NULL	Klucz obcy (tabela: hulajnogi)
poczatek_czasu	timestamp	NOT NULL	
koniec_czasu	timestamp		

Tabela 9: czas

czas_id	serial	NOT NULL	Klucz główny generowany automatycznie
wypozyczenie_id	integer	NOT NULL	Klucz obcy (tabela: wypozyczenie)
czas_start	timestamp	NOT NULL	
czas_stop	timestamp		
usluga_id	integer	NOT NULL	Klucz obcy (tabela: cennik )

## Zaprojektowanie operacji na danych

Poniżej zaprezentowane są zapytania do bazy, można je znaleźć w pliku `app.js`.

### Użytkownik:

#### Rejestracja użytkownika - **addClient**

```
'INSERT INTO klienci (imie, nazwisko,email,haslo) SELECT $1, $2, CAST($3 AS VARCHAR), $4 WHERE NOT EXISTS(SELECT 1 FROM klienci WHERE email = $3)'; [imie, nazwisko, email, haslo]
```

Logowanie użytkownika **loginCustomer** (jeśli zostanie zwrócony jeden wiersz to znaczy że podane dane są poprawne)

```
'SELECT email, haslo, klient_id FROM klienci WHERE email = $1 AND haslo=$2', [email, haslo]
```

#### Podstawowe informacje o użytkowniku **getInfoCustomer**

```
'SELECT imie, nazwisko, email, dostepne_srodki FROM klienci WHERE klient_id=$1', [k_id]
```

#### Obsługa doładowania konta przez użytkownika **paymentCustomer**

```
'INSERT INTO transakcje (kwota, status_transakcji, klient_id)
```

```
VALUES ($1, 1, $2)', [kwota, klient_id]
```

Aktualizacja lokalizacji użytkownika **updateClientLocation**

```
'UPDATE klienci SET dlugosc_geo = $1, szerokosc_geo = $2 WHERE klient_id = $3;', [dlugosc_geo, szerokosc_geo, klient_id]
```

Znalezienie najbliższych hulajnóg na podstawie lokalizacji użytkownika **getNearHulajnogi**

```
'SELECT * FROM near($1) order by odleglosc;', [k_id]
```

Obsługa nowego wypożyczenia(klient wsiada na hulajnogę) **newWypozyczenie**

```
'INSERT INTO wypozyczenie (klient_id, hulajnoga_id, poczatek_czasu) SELECT $1, $2, $3 WHERE EXISTS(SELECT 1 FROM klienci WHERE klient_id=$1 AND dostepne_srodki>5)', [klient_id, hulajnoga_id, poczatek_czasu]
```

Obsługa rezerwacji hulajnogi **newRezerwacja**

```
'SELECT * FROM rezerwacja($1,$2)', [klient_id, hulajnoga_id]
```

Obsługa rezygnacji z rezerwacji hulajnogi **newOdRezerwacja**

```
'SELECT * FROM odrezerwacja($1,$2)', [klient_id, hulajnoga_id]
```

Obsługa postoju podczas wypożyczenia **newPostoj**

```
`INSERT INTO czas (wypozyczenie_id, czas_start,usluga_id)VALUES ( (SELECT aktualne_wypozyczenie FROM klienci where klient_id=$1), $2, 2);`, [klient_id, poczatek_czasu]
```

Obsługa wznowienia jazdy (po postoju) **newWznowienieJazdy**

```
`INSERT INTO czas (wypozyczenie_id, czas_start,usluga_id)VALUES ( (SELECT aktualne_wypozyczenie FROM klienci where klient_id=$1), $2, 1);`, [klient_id, poczatek_czasu]
```

Obsługa zakończenia wypożyczenia **zakonczJazde**

```
UPDATE wypozyczenie SET koniec_czasu=$2 WHERE wypozyczenie_id=(SELECT aktualne_wypozyczenie from klienci where klient_id=$1)', [klient_id, koniec_czasu]
```

Podliczenie kosztów oraz rozładowanie hulajnogi po zakończeniu jazdy **kosztJazdy**

```
SELECT * FROM sumuj($1)
```

Wylistowanie historii ostatnich transakcji (wpłaty na konto i pieniądze ściągnięte z konta)

**getTransakcje**

```
SELECT kwota, status_transakcji FROM transakcje WHERE klient_id = $1 ORDER BY transakcja_id DESC LIMIT 15', [klient_id]
```

Zmiana swojego hasła **updateHaslo**

```
'UPDATE klienci SET haslo = $1 WHERE klient_id=$2 AND haslo=$3', [nowe_haslo,klient_id, stare_haslo],
```

## Pracownik

Logowanie pracownika **loginWorker** (jeśli zostanie zwrócony jeden wiersz to znaczy że podane dane są poprawne)

```
'SELECT email, haslo, pracownik_id FROM pracownicy WHERE  
email = $1 AND haslo=$2',[email, haslo]
```

Podstawowe informacje o pracowniku **getInfoWorker**

```
SELECT imie, nazwisko FROM pracownicy WHERE pracownik_id=$1', [k_id]
```

Lista hulajnóg które powinny być naładowane **getHulajnogiLadowanie**

```
SELECT * FROM wszystkie_hulajnogi WHERE nazwa_statusu='naladuj';
```

Lista informacji o wszystkich hulajnogach **getWszystkieHulajnogi**

```
SELECT * FROM wszystkie_hulajnogi;
```

Ładowanie hulajnogi **handleLadowanie**

```
`INSERT INTO serwis_hulajnogi(koszt, data, hulajnoga_id, pracownik_id,  
nazwa) VALUES(0, $3,$2,$1,'ladowanie');',[pracownik_id, hulajnoga_id,  
data],
```

Dodanie nowego pracownika **addPracownik**

```
'INSERT INTO pracownicy (imie, nazwisko,email,haslo) SELECT $1, $2, CAST(  
$3 AS VARCHAR), $4 WHERE NOT EXISTS(SELECT 1 FROM pracownicy WHERE  
email = $3);',[imie, nazwisko, email, haslo]
```

Dodanie nowej hulajnogi **addHulajnoga**

```
'INSERT INTO hulajnogi (id_statusu, szerokosc_geo, dlugosc_geo,  
poziom_baterii) VALUES (1, $1, $2,100)', [szerokosc_geo, dlugosc_geo]
```

Dodanie nowego serwisu **addSerwis**

```
'INSERT INTO serwis_hulajnogi (koszt, data, hulajnoga_id, pracownik_id,  
nazwa) SELECT $1, $2, $3, $4, $5 WHERE EXISTS(SELECT 1 FROM hulajnogi  
WHERE hulajnoga_id = $3);',[koszt, data, id_hulajnogi, id_pracownika,  
nazwa],
```

Usunięcie hulajnogi (zmiana statusu, żeby nadal mieć wgląd w historię) **deleteHulajnoga**

```
'UPDATE hulajnogi SET id_statusu=5 WHERE hulajnoga_id=$1',[hulajnoga_id]
```

Lista wszystkich klientów **getKlienci**

```
'SELECT klient_id, imie, nazwisko, email, dostepne_srodki from  
klienci ORDER BY nazwisko'
```

Raport z serwisów hulajnóg – ile razy hulajnoga była serwisowana i łączny koszt serwisów

**raportHulajnogi**

```
'SELECT * from raport_hulajnogi'
```

Zmiana swojego hasła **updateHasloWorker**

```
'UPDATE pracownicy SET haslo = $1 WHERE pracownik_id=$2 AND haslo=$3',  
[nowe_haslo, worker_id, stare_haslo]
```



Usunięcie klienta **deleteKlient**

```
'DELETE FROM klienci WHERE klient_id=$1;', [klient_id]
```

Poniżej zaprezentowane są funkcje, widoki oraz triggerzy – kod znajduje się w pliku 'functions.sql'

## Widoki:

### **raport\_hulajnogi**

Tworzy zestawienie wszystkich hulajnóg które były serwisowane (ładowanie też liczy się jako serwis) : pokazuje się id hulajnogi, ilość serwisów, suma kosztów serwisów posortowane po id hulajnogi.

### **wszystkie\_hulajnogi**

Tworzy zestawienie wszystkich hulajnóg: pokazuje się id hulajnogi, nazwa statusu, szerokość geograficzna, długość geograficzna oraz poziom baterii.

## Triggerzy oraz funkcje zwracające triggerzy

### **1.1 update\_srodkow**

Trigger wywoływany po wykonaniu nowej transakcji (doładowanie konta przez klienta lub ściągnięcie środków z konta po zakończonej jeździe), uruchamia procedurę nowa\_transakcja()

### **1.2 nowa\_transakcja()**

Funkcja uaktualnia dostępne środki klienta – dodaje lub odejmuje kwotę przeprowadzonej transakcji

### **2.1 hulajnoga\_zajeta**

Trigger wywoływany po dodaniu nowego wypożyczenia (klient zaczął jechać), uruchamia procedurę nowe\_wypozyczenie

### **2.2 nowe\_wypozyczenie()**

Funkcja zmienia status hulajnogi na 2 czyli zajęta, uaktualnia aktualnie wypożyczoną hulajnogę użytkownika na tą którą pożyczył, dodaje nowy rekord do tabeli czas z początkiem nowego wypożyczenia i id usługi 1 czyli jechanie.

### **3.1 hulajnoga\_postoj**

Trigger wywoływany po dodaniu nowego czasu(klient się zatrzymał lub wznowił jazdę), uruchamia procedurę nowy\_postoj()

### **3.2 nowy\_postoj()**

Funkcja aktualizuje poprzedni czas dla tego samego wypożyczenia – zakańcza go czasem startu nowego czasu

### **4.1 hulajnoga\_koniec\_jazdy**

Trigger wywoływany po uaktualnieniu wypożyczenia (zmiana czasu końca jazdy z null na czas w którym użytkownik zakończył jazdę), uruchamia procedurę **zakoncz\_jazde()**

### **4.2 zakoncz\_jazde()**

Funkcja ustawia koniec ostatniego czasu na taki sam czas jak czas stop wypożyczenia, zmienia status hulajnogi na 1 czyli wolną.

### 5.1 serwis\_hulajnogi

Trigger wywoływany po dodaniu nowego serwisowania hulajnogi, uruchamia procedurę naladuj()

### 5.2 naladuj()

Sprawdza czy nowy serwis to ładowanie hulajnogi, jeśli tak, to zmienia status hulajnogi na 1 (wolna), a poziom baterii ustawiany jest na 100%.

### Pozostałe funkcje

#### sumujkoszty\_rozladujhulajnoge

Sumuje koszty wypożyczenia: bierze z tabeli czas wszystkie rekordy dla danego wypożyczenia i liczy ile trwał każdy czas w sekundach ( $\text{czas\_stop} - \text{czas\_start}$ ) oraz mnoży przez stawkę dla danej usługi, każdy taki iloczyn jest dodawany do łącznego kosztu wypożyczenia.

Dodaje nową transakcję o wartości powyżej obliczonego kosztu wypożyczenia.

Zmienia aktualne wypożyczenie klienta na null.

Rozładowuje hulajnogę proporcjonalnie do tego, ile czasu się nią jechało – odejmuje stratę baterii od poziomu baterii hulajnogi, a jeśli poziom baterii spadnie poniżej 5% zmienia status hulajnogi na „naladuj” i przyjmuje że poziom baterii to 0.

#### near()

Znajduje hulajnogi znajdujące się w okolicy klienta – na podstawie lokalizacji klienta i lokalizacji hulajnóg liczy w jakiej odległości od siebie są i znalezione wolne hulajnogi są pokazane w kolejności od najbliższej do najdalszej.

#### rezerwacja()

Uaktualnia aktualne wypożyczenie klienta na wybraną przez niego hulajnogę i zmienia status hulajnogi na 3 (zarezerwowana).

#### odrezerwacja()

Zmienia aktualną rezerwację klienta na null i status uprzednio zarezerwowanej przez niego hulajnogi na 1(wolna).

## IV. Projekt funkcjonalny

1.Strona logowania klienta: wyjściowa strona z formularzem logowania.

Witaj w wypożyczalni hulajnóg!

**Zaloguj się!**

Email:

Hasło:

Zaloguj

[Rejestracja](#)  
[Logowanie dla pracownika](#)

2. Po zalogowaniu użytkownik jest przekierowywany na swoje konto:

Miejsce z danymi i formularz, który pozwala doładować konto użytkownika:

Witaj użytkowniku!

<b>Twoje dane:</b> <b>Imię i nazwisko:</b> Gabriela Lesniak <b>Email:</b> gabriela@mail.com <b>Dostępne środki:</b> 9.68 <button>Wyloguj</button>	<b>Doładuj swoje konto wybraną kwotą</b> <input type="text" value="0,00"/> <button>Doładuj</button>
---	---

Formularz zmiany hasła oraz formularz do przesłania swojej lokalizacji:

<b>Zmiana hasła</b> <input type="text" value="wpisz swoje stare hasło"/> <input type="text" value="wpisz swoje nowe hasło"/> <button>Zmień hasło</button>
<b>Współrzędna x</b> <input type="text"/> <b>Współrzędna y</b> <input type="text"/> <button>Prześlij swoją lokalizację</button>

Przycisk pozwalający na zobaczenie dostępnych hulajnóg:

<button>Zobacz dostępne hulajnogi</button>					
id	długość geo	szerokość geo	odległość	jedź	rezerwuj
6	9	7	1.41	<button>zaczynij jazde</button>	<button>zarezerwuj</button>
4	10	50	42.05	<button>zaczynij jazde</button>	<button>zarezerwuj</button>

-kliknięcie przycisku rezerwuj:

<button>Zobacz dostępne hulajnogi</button>					
id	długość geo	szerokość geo	odległość	jedź	rezerwuj
6	9	7	1.41	<button>zaczynij jazde</button>	<button>zarezerwuj</button>
4	10	50	42.05	<button>zaczynij jazde</button>	<button>zarezerwuj</button>

Tvoja zarezerwowana hulajnoga:

<button>odrezerwuj</button>	<button>zaczynij jazde</button>
-----------------------------	---------------------------------

Na tym etapie można zdecydować się na jazdę lub zrezygnować z rezerwacji, na czas rezerwacji zablokowana jest możliwość rezerwacji innych hulajnóg

-kliknięcie przycisku zacznij jazdę:

# Jedziesz!

## Co chcesz zrobić dalej?

postójwznów jazdęzakończ jazdę

-kliknięcie przycisku „postój” sprawi, że zacznie się naliczać czas według stawki za postój – zablokowane zostaną przyciski „zakończ jazdę” oraz „postój”

-kliknięcie przycisku wznów jazdę sprawi, że zacznie się naliczać czas według stawki za jazdę – zablokowane zostanie przycisk „wznów jazdę”

-kliknięcie przycisku zakończ jazdę zakończy jazdę i podliczy koszt jazdy:

# Jazda zakończona!

## Koszt jazdy: 0.43

[Wróć do swojego konta](#)

Przycisk pozwalający na zobaczenie historii transakcji

Zobacz historię transakcji	
status	kwota
wpłata	4.00
wydatek	0.32
wpłata	10.00

3. Strona z formularzem rejestracji – po rejestracji użytkownik jest przekierowywany do strony logowania

## Nie masz jeszcze konta?

### Zarejestruj się!

**Imię:**

**Nazwisko:**

**Email:**

**Hasło:**

[Wyślij](#)

4.Strona logowania pracownika

## Witaj w pracy!

### Zaloguj się!

**Email:**

**Hasło:**

[Zaloguj](#)

[Logowanie dla użytkownika](#)

5. Po zalogowaniu pokazuje się panel pracownika

Miejsce na dane pracownika oraz przycisk wylogowania

Dzień dobry! Witaj w pracy :)

Twoje dane: Adam Mickiewicz

Wyloguj

Formularz zmiany hasła

Zmiana hasła

wpisz swoje stare hasło

wpisz swoje nowe hasło

Zmień hasło

Przycisk „Wszystkie hulajnogi” generujący raport z danymi o wszystkich hulajnogach

Wszystkie hulajnogi

id	status	dlugosc geo	szerokosc geo	bateria
3	nieaktywna	10	10	90
4	wolna	10	50	92
5	zajeta	11	11	100
6	wolna	9	7	23

Przycisk hulajnogi do ładowania tworzy raport wszystkich hulajnóg które zmieniły status na naładuj:

Hulajnogi do ładowania

id	status	dlugosc geo	szerokosc geo	bateria	naładuj
6	naładuj	9	7	0	naładuj

Przycisk „Raport serwisu hulajnóg” tworzy raport podsumowania serwisów dla danej hulajnogi posortowany od hulajnogi której serwisy kosztowały najwięcej

Raport serwisu hulajnóg

id	ilość serwisów	łączny koszt
3	1	90
4	2	60

## Formularze

- dodanie serwisu hulajnogi
- dodanie nowej hulajnogi w danej lokalizacji

Serwisuj hulajnogę

Nazwa serwisu

Koszt serwisu

id hulajnogi

Wyślij

Dodaj nową hulajnogę

Współrzędna x

Współrzędna y

Dodaj

## Usunięcie hulajnogi – zmienia status na nieaktywną

Usuń hulajnogę - zmiana statusu

id hulajnogi

Usuń

## Formularz dodania nowego pracownika

Dodaj nowego pracownika

Imię:

Nazwisko:

Email:

Hasło:

Wyślij

## Usunięcie klienta

Usuń klienta

id klienta

Usuń

## **V. Dokumentacja**

### **Wprowadzanie danych**

Ręczne: hulajnogi, klienci, pracownicy, serwis\_hulajnogi, transakcje jeśli konto jest doładowywane

Automatyczne: status\_hulajnogi, transakcje, jeśli środki są ściągane z konta, wypożyczenie, czas

### **Krótką instrukcja obsługi aplikacji:**

Aby skorzystać z aplikacji należy się zarejestrować oraz zalogować jako klient lub skorzystać z domyślnego konta pracownika – wszystkie kroki opisane są dokładnie w punkcie IV.

### **Wykaz literatury**

<https://www.taniascia.com/node-express-postgresql-heroku/>

<https://www.postgresql.org/docs/current/>