

# System do organizacji protestów

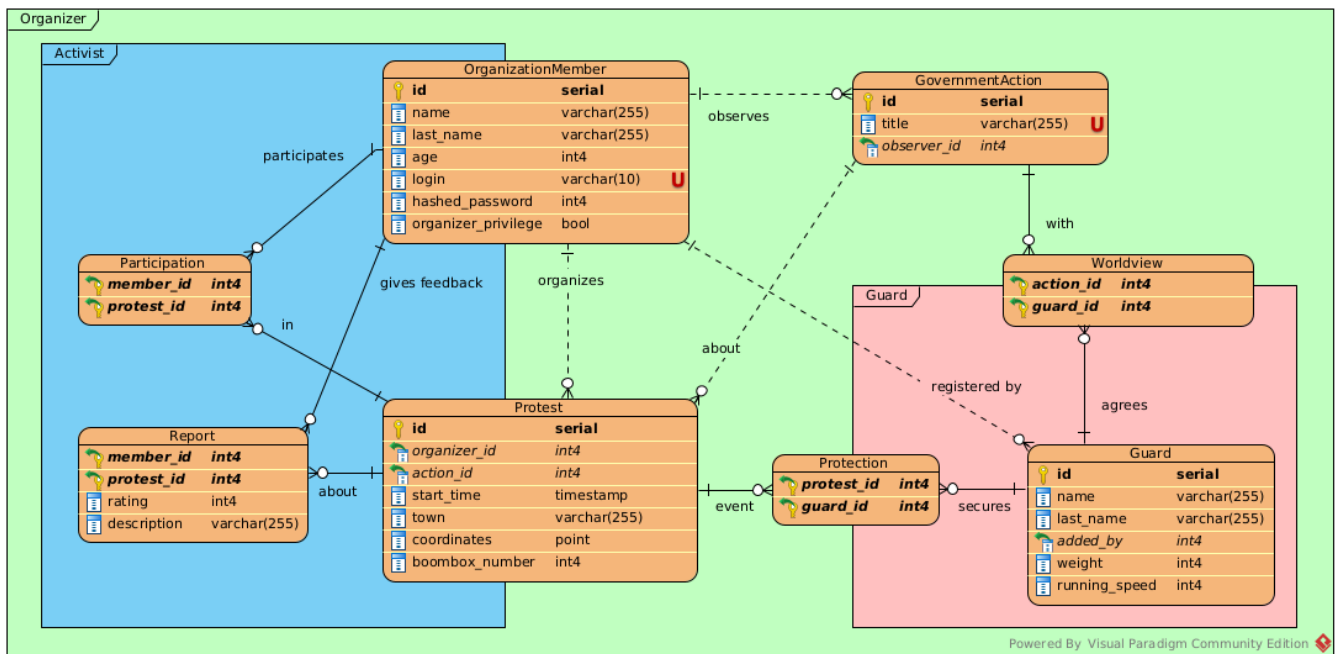
Marcin Martowicz

## Spis treści

<b>1 Model konceptualny</b>	<b>1</b>
1.1 Diagram ER	1
1.2 Role	2
1.2.1 Activist	2
1.2.2 Organizer	2
1.2.3 Guard	2
1.3 Więzy	2
1.3.1 OrganizationMember	2
1.3.2 Participation	2
1.3.3 Report	2
1.3.4 Protest	2
1.3.5 GovernmentAction	2
1.3.6 Guard	2
<b>2 Model fizyczny</b>	<b>3</b>
<b>3 API</b>	<b>3</b>
3.1 Dodawanie	3
3.2 Zapytania	3
<b>4 Obiecanka</b>	<b>4</b>

## 1 Model konceptualny

### 1.1 Diagram ER



## **1.2 Role**

### **1.2.1 Activist**

- bierze udział w protestach
- składa sprawozdania z protestów
- za zasługi może awansować na pozycję organizatora

### **1.2.2 Organizer**

- przypisuje sobie rządowe akcje do obserwacji
- planuje protesty na temat akcji, do których jest przydzielony
- rejestruje ochroniarzy do systemu, sprawdzając poglądy polityczne wykrywaczem kłamstw
- wybiera ochroniarzy do swoich protestów
- może uczestniczyć w protestach jako aktywista

### **1.2.3 Guard**

- pilnuje porządku w czasie protestów
- na wstępie przechodzi test o poglądach politycznych
- zatrudnienie załatwia u organizatorów innym kanałem komunikacyjnym

## **1.3 Więzy**

### **1.3.1 OrganizationMember**

- każdy członek musi być pełnoletni
- loginy muszą być unikalne
- hasło musi być niepuste

### **1.3.2 Participation**

- organizator musi uczestniczyć w swoim proteście

### **1.3.3 Report**

- ocena protestu musi być liczbą od 1 do 10
- treść sprawozdania musi być niepusta
- osoba składająca raport musi być zapisana na protest (relacja w telewizji może być zmanipulowana)
- osoba składająca raport nie może wysłać drugiego sprawozdania do tego samego protestu

### **1.3.4 Protest**

- musi być przynajmniej jeden boombox

### **1.3.5 GovernmentAction**

- tytuły akcji rządowych muszą być unikalne i niepuste

### **1.3.6 Guard**

- jeśli ochrania jakiś protest to musi popierać dany postulat
- musi umieć biegać z prędkością co najmniej 20km/h
- musi ważyć co najmniej 80kg

## 2 Model fizyczny

Znajduje się w pliku `protest/schema.sql`.

## 3 API

### 3.1 Dodawanie

`register_member [login] [password] [name] [last_name] [age]`

Rejestracja nowego członka.

`register_organizer [login] [password] [name] [last_name] [age] [secret]`

Rejestracja nowego organizatora.

`login [login] [password]`

Logowanie użytkownika.

`logout`

Wylogowanie.

`observe_action [action_name]`

Utworzenie rządowej akcji i przypisanie jej do obserwacji.

`add_protest [action_id] [start_time] [town] [coordinates] [boombox_number]`

Stworzenie protestu.

`add_participation [protest_id] [member_id]`

Zapisanie członka do protestu.

`add_report [protest_id] [rating] [description]`

Złożenie sprawozdania.

`register_guard [name] [last_name] [weight] [running_speed]`

Wprowadzenie ochroniarza do systemu.

`add_worldview [guard_id] [action_id]`

Wprowadzenie informacji, że dany ochroniarz popiera dane działanie rządowe.

`add_protection [guard_id] [protest_id]`

Zatrudnienie ochroniarza do protestu.

### 3.2 Zapytania

`participants [protest_id]`

Zwraca członków zapisanych na protest.

`actions_stats [limit]`

Zwraca akcje wraz z ilością przypisanych protestów i sumaryczną liczbą różnych osób zaangażowanych w protesty. Posortowane po ilości protestów.

`participants_stats [limit]`

Zwraca członków wraz z ilością protestów, w których brali udział, liczbą ich sprawozdań i sumaryczną długością sprawozdań. Posortowane malejąco po długości.

`organizer_stats [limit]`

Zwraca organizatorów wraz z ilością przypisanych protestów i średnią oceną ze sprawozdań. Posortowane malejąco po ocenie.

`find_closest_protests [point] [start_time] [end_time] [limit]`

Zwraca protesty, które odbywają się w danym przedziale czasowym. Posortowane po dystansie do podanego punktu.

`find_profitable_protests [guard_id] [start_time] [end_time] [limit]`

Zwraca protesty, które odbywają się w danym przedziale czasowym i dany ochroniarz nie jest do nich przypisany. Posortowane po liczbie boomboxów, które przypadną mu do ochrony (ochroniarze dzielą się po równo).

`indirect_friends [member_id]`

Po wspólnym proteście wszyscy członkowie zostają kolegami. Zapytanie zwraca bezpośrednich i pośrednich znajomych danego członka.

## 4 Obiecanka

Użyję języka `python` i frameworku `flask`. Aplikacja będzie odpowiadała na zapytania HTTP surowym jsonem bez frontendu. API powinno być mniej więcej takie jak wyżej.