

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

# Dav proti vizuálnemu smogu

Inžinierske dielo

---

**Vedúci tímu:** Ing. Jakub Šimko, PhD.

**Členovia tímu:** Bc. Jana Egriová, Bc. Alexander Ferenčík, Bc. Richard Filipčík, Bc. Tomáš Melicher, Bc. Juraj Slavíček, Bc. Jaroslav Zigo

**Akademický rok:** 2014/2015

# OBSAH

1	Úvod .....	4
2	Slovník pojmov .....	4
3	Ciele.....	5
3.1	Používateľ odfotoí obrázok .....	5
3.1.1	Používateľ v teréne pridá voliteľné informácie o úlovku.....	5
3.2	Domased si zobrazí úlovky na mape .....	5
3.3	Domased nahrá úlovok do mapy .....	5
3.3.1	Domased pridá voliteľné informácie o úlovku.....	5
3.4	Domased upraví detaily úlovku.....	6
3.5	Domased pridá návrh na zlúčenie úlovkov .....	6
3.6	Domased vyrieši dobrovoľnú úlohu .....	6
3.7	Domased získa ocenenia .....	6
3.8	Registrácia a prihlásenie .....	6
4	Opis prototypu .....	7
4.1	Funkcionálne požiadavky .....	7
4.1.1	Webová aplikácia .....	7
4.1.2	Mobilná aplikácia .....	7
4.2	Architektúra.....	8
4.2.1	Serverová časť .....	8
4.2.2	Mobilná časť.....	10
4.3	Dátový model .....	11
4.3.1	Logický model .....	11
4.3.2	Fyzický model.....	13
5	Moduly .....	17
5.1	Webová aplikácia .....	17
5.1.1	Registrácia a prihlasovanie používateľov.....	17
5.1.2	Mapa .....	18
5.1.3	Upload obrázka .....	21

5.1.4	Zlučovanie úlovkov.....	24
5.1.5	Úprava úlovkov .....	26
5.1.6	Presunutie úlovkov .....	27
5.1.7	Vymazanie úlovkov .....	28
5.1.8	Filter úlovkov.....	28
5.1.9	Profil používateľa .....	29
5.1.10	Zlučovanie nosičov .....	31
5.2	Android aplikácia .....	32
5.2.1	Záznam neestetickéj reklamy .....	32
5.2.2	Offline režim.....	37
6	Používateľská príručka .....	40
6.1	Mobilná aplikácia .....	40
6.1.1	Prihlásenie a odhlásenie .....	40
6.1.2	Nastavenia offline režimu .....	40
6.1.3	Odfotenie a odoslanie úlovku .....	40
6.1.4	Pridanie rozširujúcich informácií k úlovku .....	41
6.2	Webová aplikácia .....	41
6.2.1	Registrácia.....	41
6.2.2	Prihlásenie a odhlásenie .....	42
6.2.3	Mapa .....	42
7	Technická dokumentácia .....	47

# 1 ÚVOD

V dnešnej dobe plnej produktov sa vo svete vo veľkej miere rozšírila reklama každého druhu. Reklamám v televízii rádiu a tlačovinách sa človek pri určitej snahe dokáže vyhnúť avšak skryť sa reklamným pútačom, billboardom a svetelným reklamám v uliciach našich miest je takmer nemožné. Všetky tieto druhy pouličných reklám môžeme pomenovať súhrnným názvom vizuálny smog. Vo vyspelejších krajinách bol problém vizuálneho smogu vypuklý už v dávnejšej minulosti a preto je boj s ním v plnom prúde. Naopak u nás sa stáva problém čoraz závažnejším a zároveň stále neriešeným.

V rámci práce na projekte sme vyvinuli mobilnú aplikáciu, ktorá slúži na zber týchto informácií používateľmi priamo v teréne. Táto aplikácia je veľmi jednoduchá, je navrhnutá tak, aby používateľ mohol za pár sekúnd svoj úlovok priamo v uliciach vyfotiť a odoslať na server. Pre aktívnych používateľov poskytujeme možnosť vyplniť doplňujúce informácie k úlovku. K používaniu mobilnej aplikácie nie je nutné aktívne internetové pripojenie, aplikácia funguje rovnako aj v offline režime. Pri najbližšom pripojení zariadenia sa po schválení používateľom uploadnu na server všetky úlovky vyhotovené v offline režime, ktoré boli doposiaľ uložené iba v pamäti telefónu.

Druhou časťou nášho projektu je back-end spolu s webovou aplikáciou. Webová aplikácia zobrazuje všetky nahraté úlovky priamo na mape Slovenska. Používateľ si na mape môže prezerať detaily úlovkov, môže použiť filter na zobrazenie len svojich vlastných úlovkov, ktoré môže dodatočne upravovať alebo upresniť ich polohu na mape. Webová aplikácia poskytuje širšie rozhranie na interakciu s úlovkami, napríklad formou návrhov na zlúčenie viacerých úlovkov, ktoré zachytávajú jeden a ten istý reklamný nosič.

## 2 SLOVNÍK POJMOV

Pojem	Vysvetlenie
<b>Terén</b>	Vonkajšie prostredie, v ktorom má používateľ možnosť nasnímať neestetickú reklamu
<b>Špina</b>	Akákoľvek neestetická, nelegálna alebo iným spôsobom nevhodná reklama v meste
<b>Úlovok</b>	Nasnímaný záznam špiny v teréne, ktorý bol odoslaný na spracovanie
<b>Domased</b>	používateľ webového rozhrania, ktorý má aplikáciu spustenú na osobnom počítači z domova.

<b>Aktivista</b>	používateľ mobilného rozhrania v teréne, ktorý má záujem odstrániť neestetickú reklamu
------------------	--

## 3 CIELE

Našími prioritnými cieľmi na zimný a letný semester je zapracovanie nasledovných user stories do funkcionality či už mobilnej alebo webovej aplikácie.

### 3.1 Používateľ odfotí obrázok

Pomocou webovej aplikácie používateľ v teréne odfotí úlovok a v prípade záujmu aj vyplní doplňujúce informácie. Záznam sa automaticky odošle na server po pripojení k sieti wi-fi.

#### 3.1.1 POUŽÍVATEĽ V TERÉNE PRIDÁ VOLITEĽNÉ INFORMÁCIE O ÚLOVKU

Po odfotení úlovku je používateľovi v prípade záujmu umožnené zadať ďalšie zistené informácie o úlovkoch, akými sú napríklad vlastník a typ nosiča. Odoslať úlovok je však možné aj bez týchto informácií.

### 3.2 Domased si zobrazí úlovky na mape

Používateľ webovej aplikácie si priamo na mape zobrazí jednotlivé úlovky. Použitím filtra si používateľ môže nechať zobrazíť len svoje vlastné úlovky. Podrobné informácie zobrazí kliknutím na ikonku vybraného úlovku z mapy. V prípade záujmu môže z vlastnej iniciatívy údaje o úlovku upraviť.

### 3.3 Domased nahrá úlovok do mapy

Pomocou drag&drop používateľ webovej aplikácie pridá nový úlovok priamo na požadované miesto na mape. V prípade záujmu môže k úlovku napísať aj doplňujúce informácie o danom úlovku.

#### 3.3.1 DOMASED PRIDÁ VOLITEĽNÉ INFORMÁCIE O ÚLOVKU

Pri nahrávaní úlovku pomocou webovej aplikácie je používateľovi v prípade záujmu umožnené zadať do formulára aj ďalšie zistené informácie, akými sú napríklad vlastník a typ nosiča. Uploadovať obrázok je však používateľovi umožnené aj bez týchto informácií.

### 3.4 Domased upraví detaily úlovku

Prostřednictvím webové aplikace uživatel dodatočně upraví informace svého vlastního úlovku. Uživatel může takýmto způsobem interagovat jen s těmi úlovky, které nahral on sám.

### 3.5 Domased přidá návrh na zlúčenie úlovkov

Pri prezeraní úlovkov si používateľ všimol, že viacerí používatelia nahrali ten istý úlovok. Pre tieto úlovky môže vytvoriť návrh na ich zlúčenie. Používateľ vyberie všetky úlovky, ktoré chce zlúčiť do „poolu“, kde nakoniec určí hlavný úlovok, do ktorého budú ostatné zlúčené. Návrh sa uloží na schválenie (proces schvaľovania nie je cieľom implementovať v tomto období).

### 3.6 Domased vyrieši dobrovoľnú úlohu

Aktívny používateľ má možnosť nechať si vygenerovať úlohu, kde je cieľom určiť, či sa na dvoch rôznych úlovkoch nachádza jeden a ten istý nosič. Pravdepodobnosť zhody určí na zadefinovanej miere.

### 3.7 Domased získa ocenenia

Používateľ na svoju aktivitu pri zbieraní úlovkov zbiera ocenenia. Tieto ocenenia si vie prezerať vo svojom profile prostredníctvom webovej aplikácie. Oceneniami môžu byť napríklad odznaky alebo umiestnenie v rebríčkoch.

### 3.8 Registrácia a prihlásenie

Používateľ si vytvorí účet, pomocou ktorého sa môže do aplikácie prihlasovať a používať tak funkcionality len pre registrovaných používateľov. Aplikáciu môžu používať aj neregistrovaní používatelia.

## 4 OPIS PROTOTYPU

### 4.1 Funkcionálne požiadavky

Táto kapitola sa zaoberá funkcionálnymi požiadavkami, ktoré sme identifikovali po diskusiách v rámci tímu, ale hlavne s potencionálnymi používateľmi. Je rozdelená na dve logické časti podľa častí z ktorých sa celá aplikácia skladá.

#### 4.1.1 WEBOVÁ APLIKÁCIA

- registrácia používateľa - používateľ musí byť schopný registrovať sa do aplikácie pomocou formulára vo webovom rozhraní.
- autentifikácia používateľa - systém musí byť schopný overiť správnosť údajov zadaných používateľom a na ich základe mu povoliť prípadne odoprieť prístup do aplikácie
- prezeranie mapy s vizuálnym smogom - systém musí umožňovať zobrazenie prehľadnej mapy všetkých pridaných billboardov aj s podrobnými informáciami ku každému z nich
- odborné upravenie úlovkov
- pridanie nového billboardu - systém musí umožňovať pridať nový billboard na ľubovoľné miesto na mape a pridať k nemu podrobné informácie vo webovom rozhraní
- spájanie rovnakých billboardov - systém musí privilegovanému používateľovi umožniť spojiť dva a viacero pridaných billboardov po zistení že sa jedná o ten istý billboard pridaný duplicitne
- zobrazenie štatistík - systém musí umožňovať zobrazenie rôznych štatistík o zozbieraných billboardoch či už globálne alebo pre jednotlivých používateľov

#### 4.1.2 MOBILNÁ APLIKÁCIA

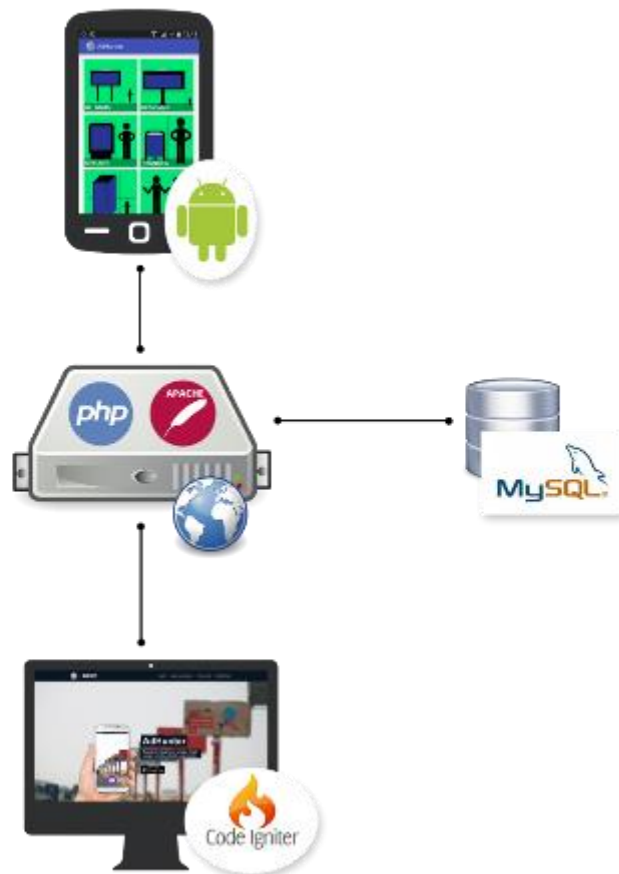
- odfotenie billboardu - aplikácia musí umožňovať odfoťiť fotku billboardu a zároveň jednoznačne identifikovať rámčekom billboard v prostredí fotky
- upload billboardu - aplikácia musí používateľovi umožniť nahrať fotku na server kde sa trvalo uloží do databázy spolu s meta informáciami získanými z používaného zariadenia ako napríklad GPS lokácia
- zadanie dodatočných informácií - používateľ musí byť schopný zadať informácie k odfoťenému vizuálnemu smogu bližšie špecifikujúce danú fotku (typ reklamy, zadávateľ, inzerovaný produkt)
- offline režim - aplikáciu musí byť možné používať aj bez aktívneho internetového pripojenia so spätným odoslaním všetkých nazbieraných úlovkov, hneď po pripojení

- prihlásenie - systém musí umožňovať prihlásenie používateľa na základe údajov poskytnutých pri registrácii cez webovú časť aplikácie

## 4.2 Architektúra

### 4.2.1 SERVEROVÁ ČASŤ

V rámci serverovej časti bol použitý webový server Apache a na ňom postavený aplikačný server PHP. Serverová časť bola navrhnutá v štýle štandardnej MVC architektúry. Pôvodne sme navrhli celý MVC model od základov v čistom PHP bez použitia akéhokoľvek frameworku, čo nám prinieslo výhodu "rýchleho odpichu". Nebolo totiž potrebné zaškoľovať členov tímu na framework, s ktorým sa pred tým ešte nestretli. V neskorších štádiách projektu, kde bolo treba riešiť bezpečnostné prvky aplikácie a integráciu s databázou sa stalo použitie existujúceho frameworku priam nevyhnutným krokom nášho projektu. Nakoniec sme sa zhodli na minimalistickom frameworku CodeIgniter, ktorý disponuje iba funkcionalitou akoby presne prispôbenou potrebám nášho projektu. Architektúru nášho riešenia približuje obrázok nižšie.





## RESTful rozhranie

Komunikácia medzi mobilnou aplikáciou a webovým serverom prebieha prostredníctvom RESTful rozhrania. Ide o moderný spôsob komunikácie medzi zariadeniami prostredníctvom internetu, ktorý ponúka viacero výhod, napríklad prehľadnosť, či relatívne jednoduchá implementácia.



Na príklade nižšie môžeme vidieť, akým spôsobom webový server ponúka zoznam vlastníkov billboardov. K serveru prístupujeme pomocou adresy v tvare:

*`/adhunter/owners/current_list/zoradenie/podla/`*

Výstup vo formáte JSON je možné prispôbiť niekoľkými parametrami:

- **zoradenie**
  - určuje, či sa zoznam vlastníkov zoradení zostupne, alebo vzostupne
  - prípustné hodnoty
    - asc - zoradiť vzostupne
    - desc - zoradiť zostupne
  - predvolená hodnota
    - asc
- **podla**
  - určuje stĺpec, podľa ktorého sa budú hodnoty zoraďovať
  - prípustné hodnoty
    - id - ID vlastníka v databáze (čím vyššie ID, tým neskôr bol do databázy vložený)
    - name - názov/meno vlastníka
  - predvolená hodnota
    - name

## CodeIgniter

Pri implementácii webovej časti nášho projektu sme sa rozhodli využiť prostredie PHP frameworku CodeIgniter. Okrem základných požiadaviek, ktoré sme pri výbere frameworku kládli na jeho funkcionality (framework založený na jazyku PHP s podporou databáz MySQL, postavený na MVC architektúre, so strmou krivkou učenia a nízkymi požiadavkami na výkon), ponúka CodeIgniter aj niekoľko črt dôležitých pre ďalšiu prácu na webovej časti projektu.

- Zabudovaná trieda pre prácu s URL adresami značne uľahčuje implementáciu RESTful rozhrania potrebného pre komunikáciu s mobilnou aplikáciou.
- Trieda pre prácu s databázou umožňuje jednoduché a bezpečné spojenie so serverom MySQL. Možnosť využiť zápis SQL požiadaviek pomocou Active Record zabezpečuje rýchlejší zápis i väčšiu zrozumiteľnosť týchto požiadaviek a rovnako i základnú ochranu pred útokmi typu SQL injection.
- Framework tiež poskytuje triedu umožňujúcu písať jednoduché unit testy, ktoré sú nevyhnutné pri vývoji riadenom testami (TDD).
- Prítomný je aj jednoduchý šablónovací systém, ktorý značne zjednodušuje a zlepšuje využitie architektúry MVC.
- V prípade nutnosti existuje celý rad rozšírení dopĺňajúcich chýbajúcu funkcionality, napr. jednoduchý objektovo-relačný mapovač.

### **Používateľské rozhranie**

Keďže naša aplikácia bola navrhnutá tak, aby oslovila čo najväčšie spektrum používateľov, bolo potrebné prispôbiť používateľské rozhranie tak, aby prácu s ním zvládol aj človek, ktorého pôsobenie je absolútne mimo oblasť informačných technológií.

## **4.2.2 MOBILNÁ ČASŤ**

### **Kostra mobilnej aplikácie**

Mobilné rozhranie je vyhotovené pre platformu Android v jazyku Java. Je rozdelené na štyri obrazovky s rôznymi komponentmi navzájom tvoriacimi funkcionality mobilnej aplikácie. Informácie o typoch nosičov reklamy a vlastníkoch reklamy je potrebné doťahovať zo servera prostredníctvom rozhrania Restful.

## 4.3 Dátový model

### 4.3.1 LOGICKÝ MODEL

Ústrednú časť aktuálnej verzie nášho dátového modelu predstavujú hneď niekoľké entity. Každá používateľom vyhotovená fotografia reklamy sa v dátovom modeli reprezentuje entitou *Úlovok*. Z toho implicitne vyplýva, že fotografiu jednej konkrétnej reklamy umiestnenej na jednom nosiči môže používateľ vyhotoviť v podstate v neobmedzenom množstve a vždy sa bude jednať o jedinečný úlovok. Náš dátový model rozlišuje medzi samotnou reklamou a nosičom (reprezentovanom entitou *Nosič*), v ktorom je fyzicky umiestnená, z čoho plynú viaceré výhody.

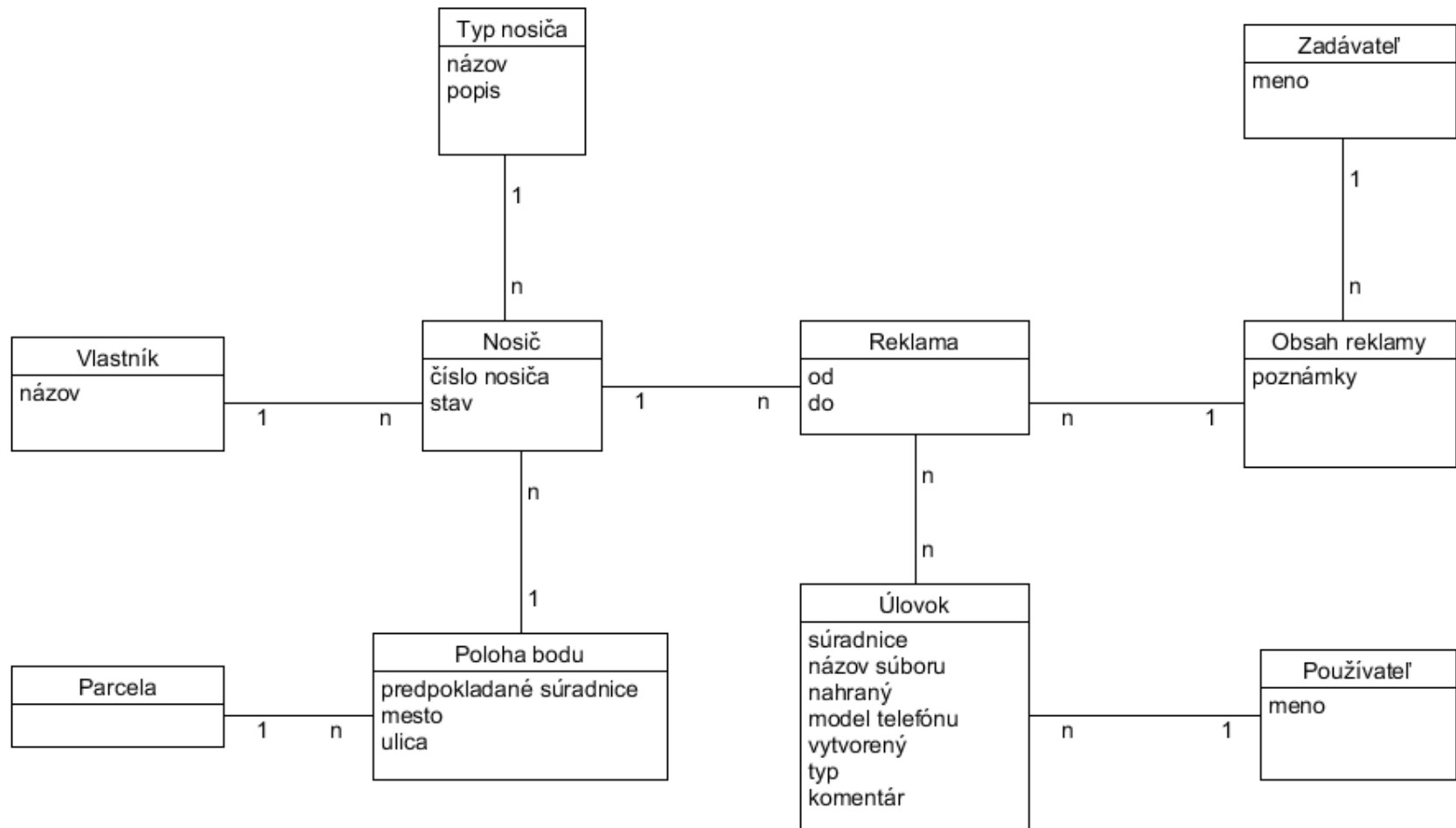
Takéto rozdelenie je však zároveň aj veľmi dôležité, ak uvažujeme časové závislosti umiestňovania reklám, pretože v rôznych časových úsekoch je možné predpokladať, že jeden nosič môže obsahovať množstvo rôznych reklám. Jedna reklama teda môže byť v modeli reprezentovaná viacerými úlovkami a jeden nosič môže mať priradených viacero reklám.

Rovnaké reklamy sú zvyčajne v rámci reklamnej kampane umiestňované na väčšom množstve nosičov. Zachytenie tejto skutočnosti môže byť zaujímavé najmä zo štatistických dôvodov, preto náš model ráta aj s takouto situáciou prepojením medzi entitami *Reklama* a *Obsah reklamy*, ku ktorej je tiež naviazaná entita *Zadávateľ*.

Aktuálna verzia modelu ráta tiež s väčším množstvom atribútov vzťahujúcim sa k nosičom.

- Entita *Typ nosiča* hovorí o spôsobe, ako sú v nosiči umiestnené reklamy, resp. aký tvar má samotný nosič, napr. „bilbord“, „megabord“.
- Entita *Vlastník* hovorí o vlastníkovi konkrétneho umiestneného nosiča. Na Slovensku existuje niekoľko najväčších prevádzkovateľov bilbordov, preto je ich zachytenie v našom modeli relatívne jednoduché.
- *Poloha bodu* určuje pravdepodobné umiestnenie nosiča z hľadiska geografických súradníc. V procese určovania skutočnej polohy môže existovať viacero pravdepodobných polôh nosiča určených na základe GPS modulu telefónov, z ktorých sa bude odhadovať finálna poloha.
- Pomocou entity *Poloha bodu* sa určuje aj parcela, na ktorej nosič leží.

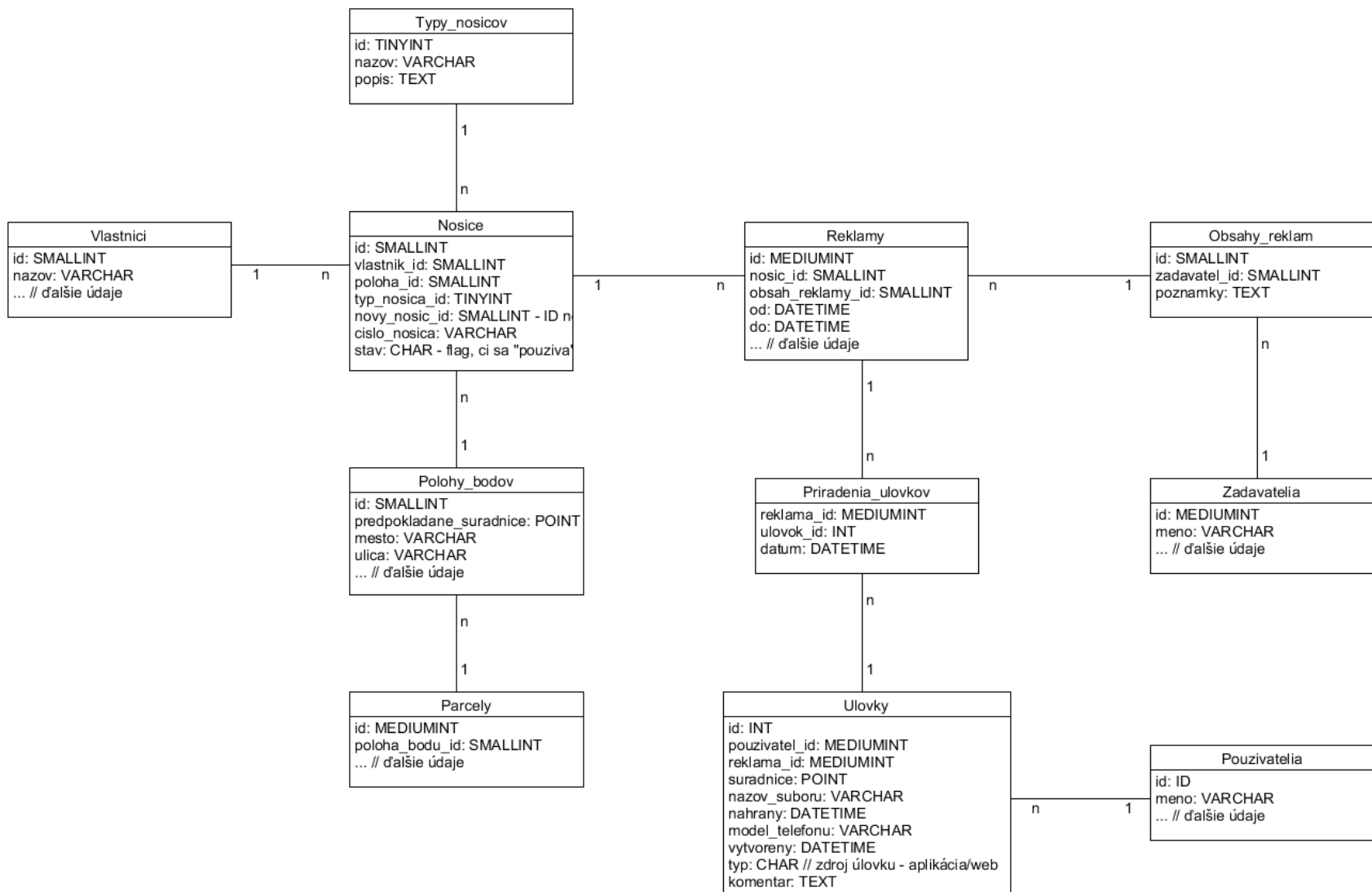
Na obrázku nižšie sa nachádza logická reprezentácia nášho dátového modelu.



### 4.3.2 FYZICKÝ MODEL

Vo fyzickom dátovom modeli si môžeme všimnúť realizáciu N:N väzby medzi Úlovkom a Reklamou. Tá je dosiahnutá pomocou entity *Priradenia\_úlovkov*, ktorá mapuje jednotlivé úlovky na reklamy, a naopak. Príčina N:N väzby medzi týmito dvoma entitami spočíva v možnosti, že istá časť úlovkov sa kvôli nepresnosti GPS modulov priradia k nesprávnym reklamám. V záujme zachovania konzistencie nášho modelu však chcem uchovávať aj tieto nesprávne priradenia, z čoho vzniká potreba mať možnosť priradiť jeden úlovok k viacerým reklamám (úlovok sa postupom času môže priradiť k inej reklame, než tej pôvodnej), no taktiež mať priradených viac úlovkov k tej istej reklame (z rovnakého dôvodu).

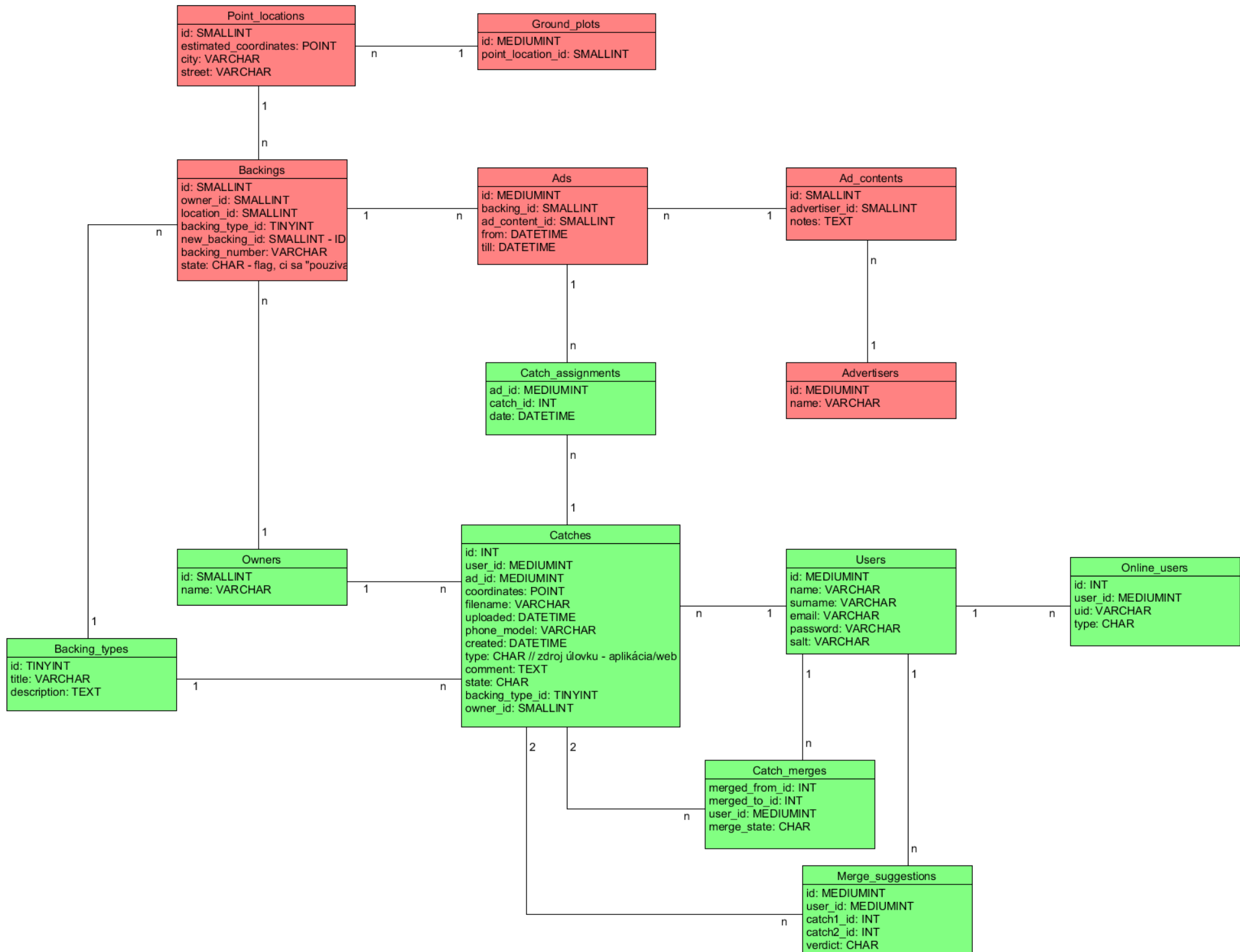
Na obrázku nižšie môžeme vidieť realizáciu všetkých spomínaných entít vo fyzickom modeli.



Na obrázku nižšie môžeme vidieť aktualizovaný fyzický model tak, ako sa vyvíjal počas práce v letnom semestri. Markantnejšou zmenou prešiel najmä spôsob evidencie zachytených úlovkov, ktorý sme museli prispôbiť našim aktuálnym možnostiam. Tým sa do dátového modelu dostali aj dočasné atribúty, ktoré budú v budúcnosti opäť odstránené. Medzi tieto dočasné atribúty patrí najmä prepojenie medzi tabuľkou úlovkov (Catches) s tabuľkou vlastníkov (Owners) a s tabuľkou typov nosičov (Backing\_types). To sa udialo predovšetkým preto, že pokročilý systém automatického rozpoznávania reklám, ktorý by využíval aj tabuľku reklám a tým by umožňoval nepriame prepojenie medzi spomenutými tabuľkami práve prostredníctvom tabuľky reklám (Ads), nebol v našich časových možnostiach a bude predmetom budúcej práce na projekte.

Medzi novo pridané tabuľky patria tiež Online\_users, Catch\_merges a Merge\_suggestions. Tabuľka Online\_users zabezpečuje ukladanie session prihlásených používateľov do databázy, tabuľka Catch\_merges uchováva informácie o manuálne zlúčených úlovkoch prostredníctvom webovej aplikácie a tabuľka Merge\_suggestions naopak uchováva informácie o návrhoch zlúčení používateľmi pomocou miniaplikácie, ktorá sa pýta na podobnosti obrázkov.

V diagrame sú tiež pre prehľadnosť zvýraznené dve skupiny tabuliek - zelenou farbou sú vyznačené tabuľky, ktoré sa už teraz reálne využívajú v našej webovej aplikácii, zatiaľ čo červenou farbou sú vyznačené tabuľky, ktoré sa síce nachádzajú vo fyzickom modeli, no aplikácia ich zatiaľ nevyužíva a ich využitie je naplánované na neskoršie fázy vývoja aplikácie. Výnimkou je tabuľka Ads, ktorá momentálne obsahuje jeden predvolený záznam, reálne sa však nepoužíva a spomenutý záznam plní iba úlohu povinnej externej entity pri namapovaní úlovku (Catches) na reklamu (Ads).





# 5 MODULY

## 5.1 Webová aplikácia

### 5.1.1 REGISTRÁCIA A PRIHLASOVANIE POUŽÍVATEĽOV

#### **Analýza**

Pri návrhu tohto modulu bolo nutné uvažovať nad rôznymi aspektami registrácie a s ňou úzko súvisiaceho prihlasovania. Prvým z nich je objem údajov ktorý chceme o našich používateľoch vedieť a ktoré sú nám stále ochotní poskytnúť cez registračný formulár. Mimo nevyhnutných údajov ako sú email a heslo by mal byť povinný aj údaj o pohlaví používateľa . Je totiž zaujímavé neskôr zo zozbieraných dát analyzovať aké reklamy považujú za neestetické muži a ženy. Ďalšou otázkou bolo využívanie prezývky tzv. nicku alebo mena a priezviska daného používateľa. Na základe rozhovorov s aktivistami v oblasti vizuálneho smogu sa ako lepšia možnosť javí meno a priezvisko keďže ľudia snažiaci sa zlepšiť verejný priestor sa týmto radi verejne prezentujú a pochvália a málokedy sa radi skrývajú za prezývky.

Vzhľadom na zadávanie hesiel pri registrácii je v súčasnosti kladený dôraz na jeho čo najväčšiu bezpečnosť. Bežným štandardom v tejto oblasti je hešovanie hesla. Tento proces predstavuje aplikáciu komplexnej matematickej funkcie na zadané heslo prevedené na odlišný reťazec znakov ktorý spätne nie je prevediteľný na pôvodné heslo. Na výber bolo viacero algoritmov ktorých implementácie jazyk PHP poskytuje. Zvažovali sme funkcie MD5(Message Digest Algorithm) a funkcie typu SHA (Secure Hash Algorithm). Problémom funkcie MD5 je, že je síce rýchlejšia ako ostatné ale jej algoritmus bol prelomený je to takisto najbežnejšie používaná funkcia a preto voči nej existuje veľké množstvo tzv. dúhových tabuliek (rainbow tables) čo znamená veľkú zraniteľnosť navyše je známe že v nej ľahšie nastávajú tzv. kolízie. To v praxi znamená že dva a viacero používateľských hesiel môžu byť zahašované ako rovnaký výstupný reťazec. Funkcie typu SHA majú už niekoľko generácií. Funkcie z rodiny SHA-1 trpia menšími aj keď podobnými problémami ako MD5. Ich nástupnícke funkcie z rodiny SHA-2 sú momentálne ešte stále na dobrej úrovni aj keď aj k nim existuje lepšia alternatíva v podobe funkcii SHA-3. Tie však nie sú bežne poskytované v základnom balíku jazyka PHP a bolo by nutné používať rozširujúce balíčky.

#### **Návrh**

V module registrácie a prihlasovania sme sa na základe vyššie uvedenej analýzy rozhodli od používateľov žiadať nasledujúce údaje. Sú nimi email, heslo, meno, priezvisko a pohlavie. Email bude zároveň vzhľadom na svoju unikátnosť slúžiť aj ako používateľské meno pre vstup do

aplikácie. Za hešovací algoritmus bola zvolená funkcia z rodiny SHA-2 a to konkrétne SHA512 čo prináša rozumný kompromis medzi bezpečnosťou a zložitou implementáciou. Pre zvýšenie bezpečnosti tohto algoritmu bude použitá tzv. salt ktorá bude náhodne generovaná unikátne pre každého používateľa a bude spolu s jeho ďalšími údajmi uložená v databáze znemožňujúca prípadne pokusy o prelomenie hesla na základe tabuliek.

## **Implementácia**

Modul bol implementovaný na základe navrhnutého riešenia. Používateľ zadá všetky požadované údaje do bežného formulára poskytovaného jazykom PHP. Následne údaje prejdú validáciou ako napríklad na správny formát emailovej adresy ([xx@yy.zz](mailto:xx@yy.zz)) a je takisto skontrolovaná prítomnosť všetkých povinných údajov. Následne sa pomocou zabudovaného generátoru náhodných 32bitových reťazcov vygeneruje pre nového používateľa jeho unikátny salt. takto vytvorený salt je pripojený k heslu a následne zahešovaný pri návrhu zvolenou funkciou SHA512. Zahešované heslo a salt spolu s ostatnými zadanými údajmi sa následne uložia do databázy.

Pri autentifikácii používateľa je opäť využitý štandardný PHP formulár. Formulár obsahuje textové polia meno a heslo pričom menom je používateľov email zadaný pri registrácii a ten je opäť validovaný. Následne je z databázy vytiahnutá salt a rovnakým postupom ako pri registrácii je vytvorený hash. tento hash je následne porovnávaný so záznamom uloženým v databáze a na základe výsledku porovnania je používateľ oprávnený vstúpiť do aplikácie alebo mu je prístup odopretý.

## **Testovanie**

Pre potreby overenia správnej funkčnosti tohto modulu bolo implementovaných niekoľko testov. testy sa zameriavali na testovacie vytváranie používateľov a následné overovanie správnosti uloženia údajov do databázy. V neposlednom rade boli napísané testy overujúce správnú autentifikáciu už registrovaných používateľov na základe správnych či nesprávnych prihlasovacích údajov.

### **5.1.2 MAPA**

#### **Analýza**

Pri výbere API pre našu mapu sme uvažovali medzi slovenským poskytovateľom mapy.sk, ktorý v dnešnej dobe spadá pod spoločnosť Atlas a mapy majú od spoločnosti eurosense s.r.o. Napriek tomu, že ide o slovenského poskytovateľa, k dispozícii sú aj mapy krajín celej Európy. API daného poskytovateľa je však veľmi zle zdokumentovaná a pre developera je k dispozícii relatívne málo funkcií.

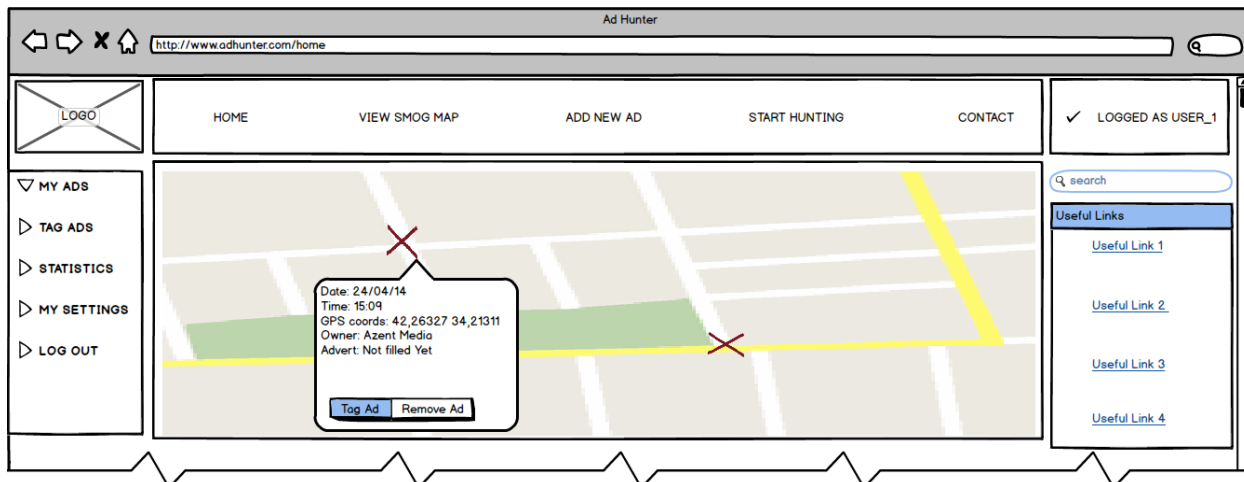
Ďalšou možnosťou bola použitie API od poskytovateľa Google Maps. Nakoľko v pozadí tohto projektu stojí gigant Google, je tu množstvo výhod oproti slovenským mapy.sk. Tím vývojárov ustavične pracuje na nových vychytávkach a odstraňovaní chýb. Celá API je výborne zdokumentovaná, pričom je k dispozícii množstvo tutoriálov, ktoré prispievajú k rýchlejšiemu pochopeniu celej API. Oproti spomínaným mapy.sk sú omnoho detailnejšie, sú na nich zobrazené zastávky MHD, ku významným budovám sú častokrát doložené aj fotografie z okolia, recenzie na daný objekt a otváracie hodiny. Takéto detailné zobrazenie je pre náš projekt dôležité nakoľko používateľ si častokrát nepamätá presnú adresu, na ktorej odfotil neestetický billboard, pamätá si, že to bolo v blízkosti nejakej autobusovej zastávky prípadne nejakej významnej budovy. To bolo kľúčové pri výbere technológie, a preto sme sa nakoniec rozhodli pre mapy od spoločnosti Google.

Do úvahy prišla aj myšlienka implementácie vlastnej mapy, to by však ale bolo časovo veľmi náročne a neefektívne. Google mapy je možné implementovať dvomi spôsobmi. Prvý je jednoduché vloženie vygenerovaného iframe. Takáto implementácia však ponúka iba obmedzenú funkcionálnosť a ide o rapídne pomalšie riešenie, čo sa týka času potrebného na kompletne načítanie stránky. Nad kódom v iframe totižto nie je možné vykonávať vlastný JavaScript kód z bezpečnostných dôvodov, aby sa predišlo cross-site útokom. Podobne nie je možné ani dodatočne štýlovať kód v iframe pomocou jazyka CSS.

Druhým spôsobom pre implementáciu google máp je pomocou JavaScriptu. Ide o náročnejšiu možnosť v rámci implementácie, prináša však viacero výhod. Je tu možné vykonávanie vlastného JavaScript kódu a používanie vlastných CSS štýlov, čo možno využiť napríklad pri zmene markerov v mape, prípadne ovládacích prvkov mapy.

## **Návrh**

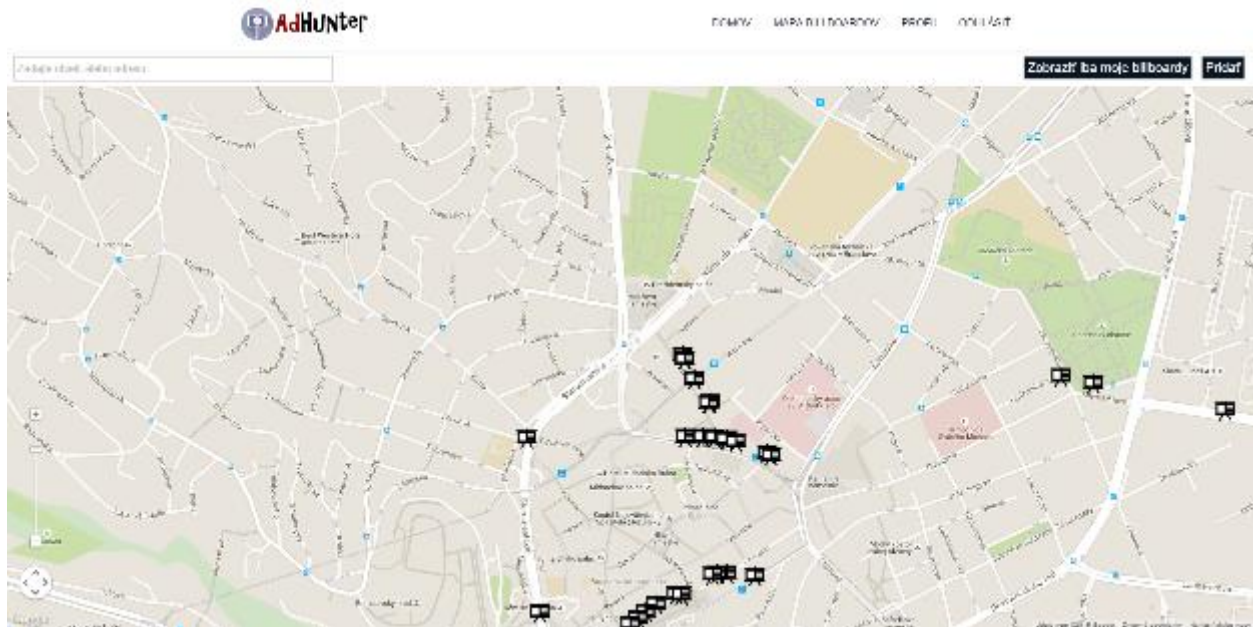
Pre potreby nášho projektu sa ukázali byť najlepšou možnosťou mapy od spoločnosti Google, pričom je potrebná implementácia aj dodatočných funkcií ako napríklad street view, press & drag a podobne. Ovládanie mapy musí byť jednoduché a intuitívne, ideálne s prvkami, na ktoré je používateľ zvyknutý z používania webovej aplikácie maps.google.sk. Na mape musia byť prehľadne zobrazené zozbierané billboardy, ideálne ikonou, ktorá by ich jasne vystihovala.



## Implementácia

Naše mapy boli nakoniec implementované pomocou API ponúkané spoločnosťou Google. Vzhľadom na množstvo výhod sme ich implementovali pomocou technológie JavaScript na strane klienta a na strane servera je využitý jazyk PHP. Zo strany servera sú najprv posielané geodata vo forme JSON. Následne sa na strane klienta spracujú zaznačia do mapy. Vykreslenie mapy prebieha zavolaním externého JavaScriptu, umiestneného na serveroch spoločnosti Google. Po úspešnom vykonaní daného skriptu sa vykoná definovaná callback funkcia, konkrétne v našom projekte ide o funkciu `initMap()`. V spomenutej callback funkcii je následne definovaný štýl zobrazenia našej mapy, rozloženie a správanie jednotlivých ovládacích prvkov a samotných billboardov. Následné sa spracuje získaný objekt v tvare JSON a naplní sa tak naša mapa billboardmi z databázy. Pridávanie billboardu prebieha jednoduchým zaznačením miesta na mape, pričom používateľovi je zobrazený sprievodný text. Po zaznačení polohy na mape je používateľovi zobrazený formulár, v ktorom zadá dodatočné informácie o danom billboarde a tie sú následne odosielané na server formou POST requestu. Na serveri sa následne spracujú prijaté dáta a uložia do databázy v požadovanom tvare. Naša mapa ponúka funkcionality pre `press & drag`, čiže používateľ sa vie jednoducho pohybovať po mape bez použitia klávesnice prípadne smerových tlačidiel na mape. Priblíženie a oddialenie mapy je možné taktiež pomocou štandardného ovládacieho prvku, prípadne pomocou kolieska myši. Samozrejmosťou je aj podpora funkcie `street view`.

Na nasledujúcom obrázku sa nachádza ukážka z prostredia webovej aplikácie spomínaného modulu `mapa`:



## Testovanie

Počas vývoja bola vykonaných niekoľko unit testov, ktoré boli zamerané na predídenie rôznym typom útokom vo webovom prostredí a na korektné spracovanie dát z databázy, prípadne na korektný zápis dát do databázy. Jeden z unit testov bol prídanie billboardu s názvami obsahujúcimi predpripravený kód pre útok typu SQL injection, ďalšie testy boli na strane klienta zamerané na kontrolu odpovedí(response) zo servera. V rámci finálneho testovania bolo vykonaných niekoľko skúšobných prídání billboardu, zamerané na hraničné prípady vzhľadom na vytvorený zdrojový kód.

### 5.1.3 UPLOAD OBRÁZKA

#### Analýza

Pri uploade obrázka je dôležité určiť v akom formáte budú dáta posielané na server a následne spracované. Vyskytlo sa viacero možností pričom bolo potrebné dbať na objem prenášaných dát, keďže dnešné moderné fotomobily vytvárajú fotografie o veľkosti niekoľkých megabajtov. Keďže bolo potrebné myslieť aj na to, že naša aplikácia pozostáva aj z mobilnej časti, ktorá je implementovaná ako aplikácia v jazyku JAVA, bolo teda potrebné nájsť riešenie, ktoré by fungovalo aj vo webovej aplikácii aj v mobilnej. Vo webovom prostredí je možné dáta prijímať v týchto formátoch:

- application/x-www-form-urlencoded: ide o defaultný formát pre posielané dáta cez web formuláre. Pri tomto formáte sú medzery v posielaných dátach konvertované na znak '+'

a špeciálne znaky sú konvertované na ich ascii hodnotu v hexadecimálnom kóde. Nevýhodou je, že sa konvertuje dosť veľa znakov, čo má za následok dlhší čas na spracovanie dát a výsledný objem konvertovaných dát je relatívne veľký.

- multipart/form-data: tento formát dát sa odporúča pri spracovaní rôznych typov súborov cez formuláre. Parametre sú posielané ako čisté dáta bez akéhokoľvek konvertovania, čo má za následok kratší čas, potrebný na spracovanie zasielaných dát. Keďže pri tomto formáte neprebíha žiadna konverzia špeciálnych znakov na sekvenciu čísel určujúcich ich ASCII hodnotu, objem dát je podstatne nižší ako pri prvom spomínanom formáte.
- text/plain: pri tomto formáte sú medzery konvertované na znak '+' podobne ako pri formáte application/x-www-form-urlencoded, avšak špeciálne znaky nie sú konvertované. Z HTML5 špecifikácie (<http://www.w3.org/TR/html5/forms.html#text/plain-encoding-algorithm>) vyplýva, že tento formát sa používa iba na účely debugovania zasielaných dát.
- base64 reťazec - v prípade zasielania textov a reťazcov prostredníctvom web formulárov, je tu možnosť aj každý obrázok prekonvertovať na base64 reťazec a následne poslať na server iba textové parametre. Pri tejto možnosti však treba zobrať do úvahy, že objem prenášaných dát je podstatne vyšší než pri hociktorom inom formáte a rovnako aj čas potrebný spracovanie dát je podstatne dlhší. Výhodou je jednoduchá implementácia či už v mobilnej alebo webovej aplikácii.

V mobilnej aplikácii je možné dáta odosielať v ľubovoľnom formáte, niektoré sú natívne podporované, na iné je potrebné použiť externú knižnicu.

Osobitne sme analyzovali najčastejšie sa vyskytujúce typy nosičov, kedy sme identifikovali niekoľko nasledujúcich typov:

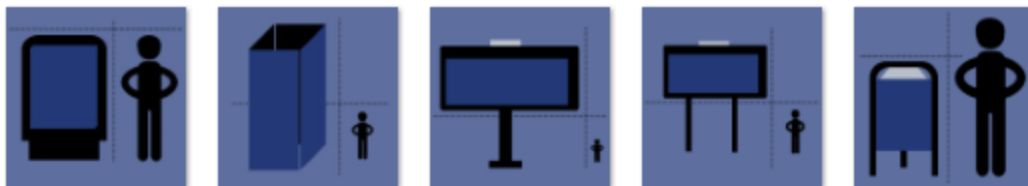
- billboard
  - rozmery: 510 x 240 cm
  - najčastejší výskyt: pri chodníkoch, na plotoch okolo stavieb, pri cestách
- megaboard
  - rozmery: 16 x 9 m
  - najčastejší výskyt: pri cestách, nákupných centrách, okolo diaľnic
- citylight
  - rozmery: 118,5 x 175 cm
  - najčastejší výskyt: na autobusových zastávkach a peších zónach
- hypercube
  - rozmery: 2,5 x 6 m x 4 strany
  - najčastejší výskyt: na autobusových zastávkach a peších zónach
- trojnožka:

- rozmery: 60 x 85 cm
- najčastejší výskyt: na chodníkoch okolo zariadení verejného osvetlenia alebo stĺpov elektrického vedenia

## Návrh

Primárnym faktorom pri voľbe formátu dát bola rýchlosť uploadu nasnímanej fotografie. Bežného používateľa mobilnej aplikácie by mohlo odradiť od používania našej aplikácie neúmerne dlhé čakanie na upload jedinej fotografie. Preto sme pri výbere brali ohľad na množstvo odosielaných dát na server a rýchlosť spracovania dát. Ako najlepšou voľbou sa ukázalo upload fotografie a k nej prislúchajúcich údajov prostredníctvom formátu multipart/form-data. Dôležitým faktorom pri implementácii však musí byť konzistencia medzi jednotlivými modulmi, z mobilnej i webovej aplikácie teda musia byť zasielané dáta v rovnakom formáte. Pri uploade úlovk bolo potrebné určiť dodatočné informácie k úlovk. Ako relevantné informácie, ktoré sú prijímané z mobilnej i webovej aplikácie sme určili polohu billboardu, dátum a čas pridania úlovk, dátum a čas vyhotovenia samotnej fotografie, voliteľný komentár a typ nosiča.

- poloha billboardu - je určená v prípade mobilného zariadenia pomocou GPS senzora, v prípade webovej aplikácie používateľ vyznačí polohu úlovk na mape. Udáva sa ako hodnota latitúdy a longitúdy pre zvolenú pozíciu.
- dátum a čas pridania úlovk - bude získavaný zo servera aby sa predišlo nekonzistencii v hodnotách
- dátum a čas vyhotovnia fotografie - bude získavaný z atribútov fotografie
- voliteľný komentár - používateľ môže zadať komentár napríklad s popisom obsahu daného billboardu
- typ nosiča - identifikovali sme 5 základných typov nosičov spomínaných v analýze, ktoré sú znázornené na nasledujúcom obrázku:



- typ "citylight"
- typ "hypercube"
- typ "megaboard"
- typ "billboard"
- typ "trojnožka"

Okrem spomínaných informácií je potrebné posilať aj špecifické informácie len v rámci mobilného klienta. Ide konkrétne o atribút model telefónu. Na základe tohto atribútu je možné zistiť zdroj úlovku, v prípade, že úlovok disponuje daným atribútom ide o úlovok z mobilnej aplikácie.

## Implementácia

Do existujúceho kontrolera billboards bola implementovaná funkcia add, ktorá pri volaní spomínaného kontrolera zistí, či ide o request uploadu obrázka. Ak áno, je potrebné skontrolovať, či sú splnené podmienky pre uloženie fotografie na server (priečinkok musí existovať a musíme mať práva doňho zapisovať), následne je nutné overiť, či boli zadané potrebné informácie k danej fotografii. Ide konkrétne o súradnice úlovku, dátum a čas vyhotovenia fotografie úlovku. Následne sa spracuje voliteľný parameter určujúci komentár k danej fotke. Pri komentári má používateľ k dispozícii najväčšie spektrum povolených znakov, tento vstup musí byť obzvlášť zabezpečený proti cross-site útokom. Použili sme preto PHP funkciu htmlspecialchars, ktorá nahradí špeciálne znaky ich bezpečnou alternatívou (< je nahrádzané entitou &lt; " je nahrádzané &quot;...).

Ďalším voliteľným parametrom je model telefónu, ktorý určuje, typ mobilného zariadenia. V prípade, že tento parameter nebol zadaný alebo obsahuje prázdnu hodnotu, ide o úlovok nahratý prostredníctvom webovej aplikácie. Po spracovaní vstupných parametrov sa do priečinka assets/pics sa uloží zaslaná fotografia s názvom očisteným od špeciálnych znakov a znakov s diakritikou. Následne sa zavolá funkcia save\_ulovek pre model Ulovok\_model, ktorá uloží dodatočné informácie ku konkrétnemu úlovku do databázy. Ak upload fotografie prebehol úspešne, zavolá sa view uploaded\_billboard, ktorý používateľa informuje o úspešnom pridaní jeho úlovku.

## Testovania

V rámci testovania bolo odoslaných niekoľko fotografií, či už vo webovom prostredí alebo z mobilnej aplikácie a následne ich naša webová aplikácie úspešne spracovala.

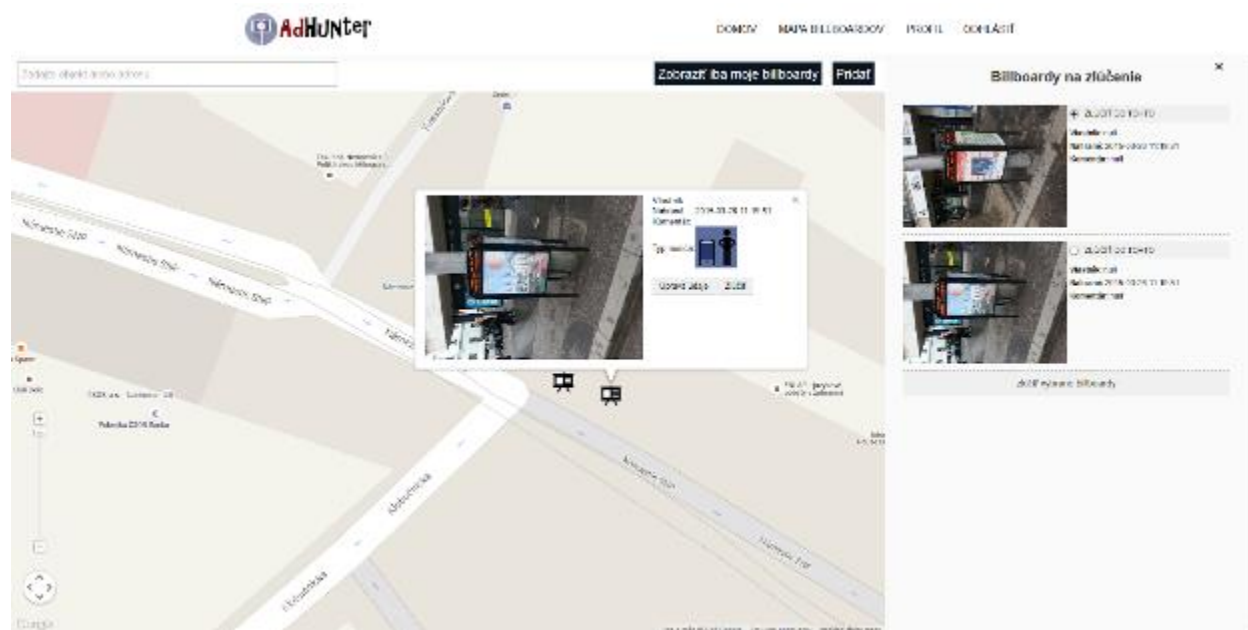
### 5.1.4 ZLUČOVANIE ÚLOVKOV

Jednou z nových funkcií webovej aplikácie, ktorú sme počas ďalšej práce na projekte pripravili, je možnosť zlučovania zhodných úlovkov priamo pre zaregistrovaných a prihlásených používateľov. Pri nahrávaní úlovkov prostredníctvom mobilnej (prípadne i webovej) aplikácie totiž určite často nastanú prípady, že sa na náš server dostanú i duplicitné zachytenia reklám. To nášmu zámeru nevedí (práve naopak), je však veľmi vhodné mať informáciu o tom, ktoré zo zachytených reklám sú totožné a ktoré nie. Pretože automatizované rozhodovanie sme z dôvodu príliš veľkej



náročnosti zamietli, rozhodli sme sa do tohto procesu zapojiť našich zaregistrovaných “lovcov reklám”.

Zlučovanie úlovkov sme sa snažili navrhnuť tak, aby bolo čo najjednoduchšie a zároveň aj ľahko prístupné pre každého. Zabudovali sme ho preto priamo do mapy úlovkov. Používateľ si po zobrazení náhľadu úlovkov na mape môže jednoduchým kliknutím zvoliť ktoré úlovky chce pridať do skupiny “rovnakých” úlovkov. Všetky takto nahromadené úlovky môže vidieť v paneli napravo od mapy. V nej si napokon zvolí “hlavný” úloвок, do ktorého sa zlúčia všetky ostatné a svoju voľbu potvrdí. Tým je proces zlučovanie úlovkov úspešne završený. Približuje ho nasledujúci obrázok.



Úlovky, ktoré boli zlúčené do iných úlovkov, sa následne na mape zobrazujú polopriehľadne, čím sa dajú jednoduchšie odfiltrovať od “plnohodnotných” úlovkov. Po potvrdení tohto zlúčenia privilegovanou osobou (tento proces ešte momentálne nie je implementovaný) zlúčené úlovky zmiznú z mapy úplne. Náhľad na skupinu zlúčených úlovkov na mape môžeme vidieť nižšie.

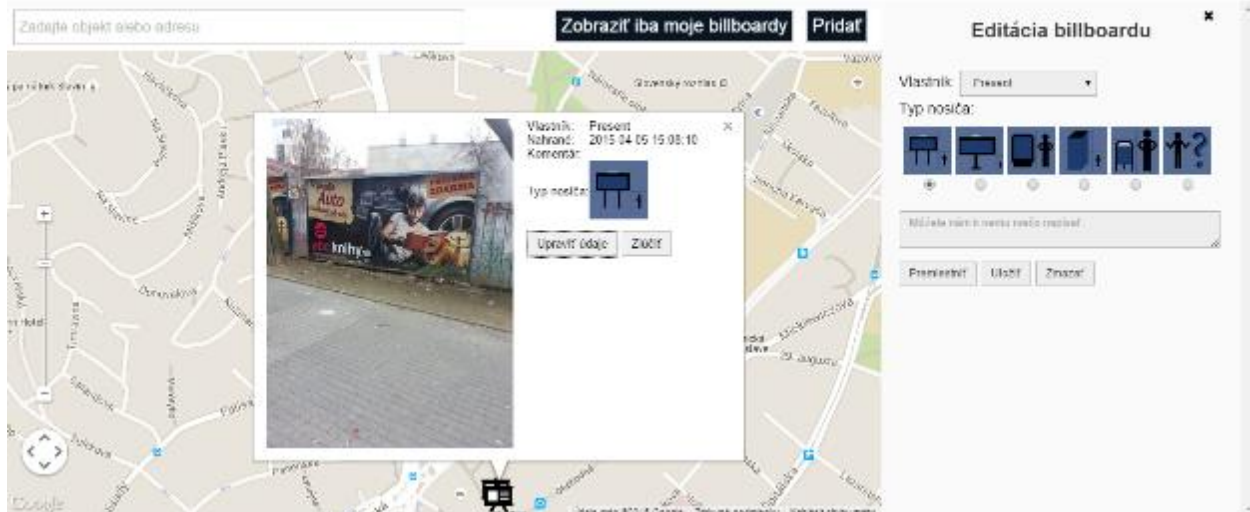


### 5.1.5 ÚPRAVA ÚLOVKOV

Neraz sa stalo, že používateľ zle odpísal vlastníka billboardu a následne po pridaní na server túto informáciu potreboval aktualizovať. Taktiež používateľ v teréne nemusí v rýchlosti správne posúdiť typ billboardu, správny typ billboardu zistí, až keď si svoj úlovok zobrazí doma spolu s ostatnými na mape billboardov. Bolo preto potrebné implementovať funkcionality pre úpravu dodatočných informácií o konkrétnom úlovku. V aktuálnej fáze projektu je možné aktualizovať tieto informácie o úlovku:

- vlastníka
- typ nosiča
- komentár k úlovku

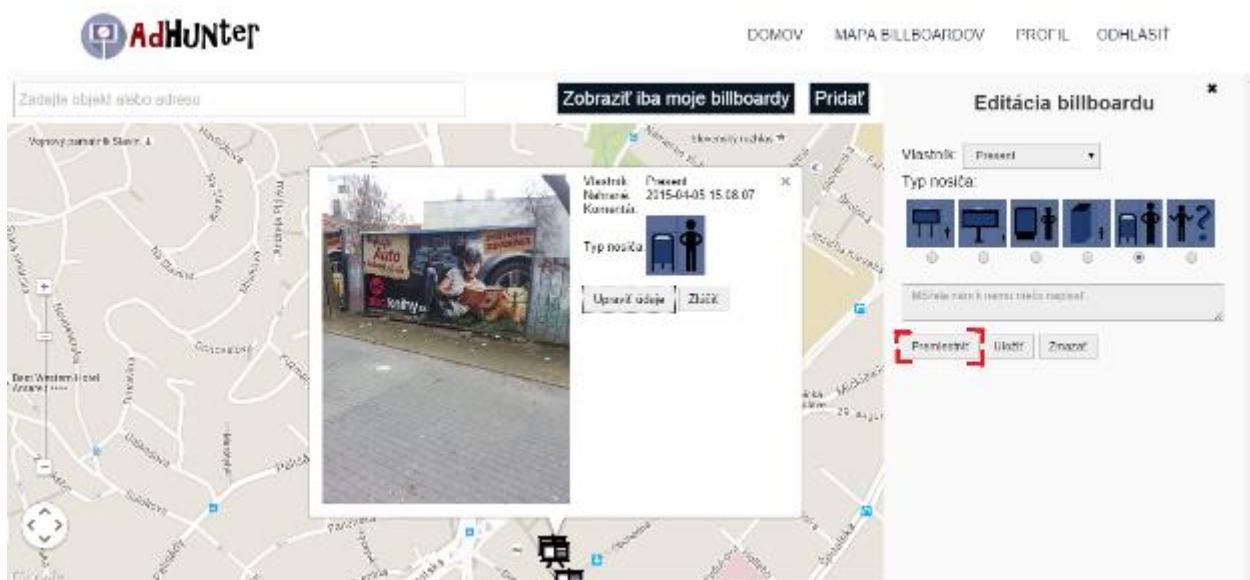
Grafické rozhranie pre úpravu billboardu sme zakomponovali do bočného panela, rovnako ako pri funkcionalite zlučovania úlovkov.



Po kliknutí na tlačidlo uložiť sa vykoná asynchrónna požiadavka na server, ktorá uloží vykonané zmeny pre konkrétny úlovok do databázy a následne sa na strane klienta aktualizuje iba zmenený úlovok. Môžeme teda vidieť, že celý proces prebieha v pozadí, nie je ani potrebné znovu načítať stránku.

### 5.1.6 PRESUNUTIE ÚLOVKOV

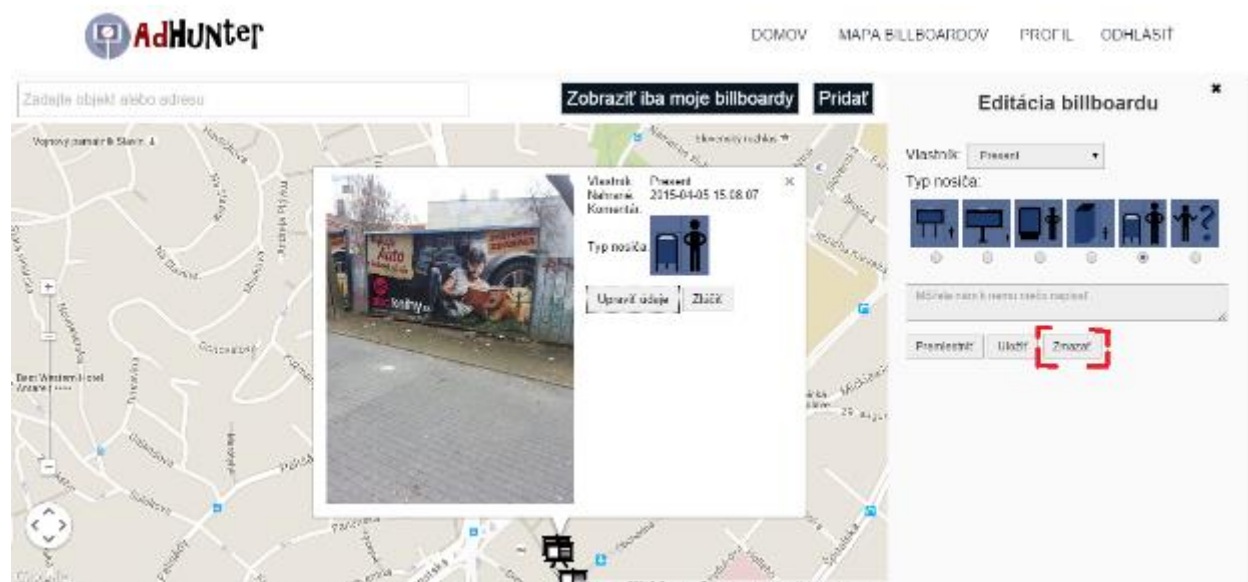
GPS moduly súčasných mobilných zariadení majú odchýku aj niekoľko metrov. Neraz sa stalo, že odfotený úlovok nemal na mape pozíciu, akú by mal mať. Z tohto dôvodu bolo potrebné do našej webovej aplikácie implementovať aj funkcionality pre zmenu polohy úlovku. Cieľom bolo implementovať túto funkciu prostredníctvom metódy drag and drop, čo sa nám aj podarilo.



Po kliknutí na zvýraznené tlačidlo v bočnom paneli má používateľ možnosť presunúť daný úlovok na novú pozíciu. Túto možnosť má používateľ k dispozícii len pre svoje úlovky.

### 5.1.7 VYMAZANIE ÚLOVKOV

Pre náš projekt bolo nevyhnutné implementovať aj funkcionality pre zmazanie úlovku. Takúto možnosť má samozrejme používateľ k dispozícii len pre svoje úlovky. Používateľ nájde takúto možnosť v bočnom paneli.



Po kliknutí na zvýraznené tlačidlo sa vykoná asynchrónna požiadavka na server, pričom sa zmaže záznam o konkrétnom úlovku z databázy. Samotná fotografia úlovku ostáva aj naďalej na serveri. Celý proces prebieha v pozadí, na strane klienta sa iba odstráni ikona zmazaného úlovku.

### 5.1.8 FILTER ÚLOVKOV

Do našej webovej aplikácie bolo taktiež potrebné implementovať filter úlovkov, nakoľko v našej aplikácii sa nachádza čoraz viac úlovkov. Neraz sa stalo, že v zhluku úlovkov bolo ťažké nájsť konkrétny úlovok aj pri najväčšom priblížení. Keďže sme boli časovo obmedzení, stihli sme implementovať iba filter mojich vlastných úlovkov, ktorý funguje ako prepínacie tlačítko, viď obrázok nižšie:

Zadajte objekt alebo adresu

Zobrazit iba moje billboardy Pridat



Po kliknutí na tlačidlo “Zobrazit iba moje billboardy” sa vykoná asynchrónna požiadavka na server, ktorá bez opätovného načítania stránky zobrazí iba moje úlovky. Po kliknutí sa funkcionality tlačidla zmení na filter všetkých úlovkov.

### 5.1.9 PROFIL POUŽÍVATEĽA

Počas práce v letnom semestri vznikla vo webovej aplikácii úplne nová časť Profil používateľa, ktorá obsahuje stránky zamerané špecificky na prihláseného používateľa. V súčasnosti sa delí na päť podstránok:

- Nastavenia účtu
- Moje úlovky
- Moje ocenenia
- Štatistiky
- Zlučovanie bilbordov

Podstránka Nastavenia účtu obsahuje, pochopiteľne, možnosť základných úprav účtu prihláseného používateľa. V súčasnosti si môže upraviť svoje osobné údaje, ktoré zadával pri registrácii, pri prípadnom pridaní ďalších údajov v budúcnosti je možnosť jednoduchého pridania editácie aj týchto častí.

Podstránka Moje úlovky obsahuje jednoduchú tabuľku so základnými údajmi a obrázkami pridaných úlovkov aktuálnym používateľom. Táto časť je predovšetkým doplnkom k primárnej časti prezentácie pridaných úlovkov, ktorým je hlavná mapa.

Moje ocenenia je podstránka, ktorá používateľovi predstavuje jeho doposiaľ získané gamifikačné prvky - ocenenia.

Na podstránke s názvom Štatistiky sa nachádzajú štatistické informácie o používateľovej poslednej činnosti.

Už raz spomínaná miniaplikácia navrhovania billboardov pre zlúčenie sa nachádza na podstránke Zlučovanie billboardov. V profilovej časti sa nachádza predovšetkým preto, že uložené rozhodnutia sa evidujú spolu s identifikáciou používateľa, ktorý ich urobil. V budúcnosti sa tiež ráta s určitou personalizáciou ponúkaných návrhov.

## **Moje ocenenia**

Používateľ v časti Moje ocenenia môže sledovať svoje dosiahnuté výsledky za aktivitu, ktorú zaznamenal v aplikácii AdHunter. Aktivita sa hodnotí v rôznych kategóriách a používateľ je odmeňovaný symbolickými medajlami, respektíve odznakmi, ktoré sú mu pridelené postupne, podľa počtu zaznamenaných billboardov v danej kategórii.

Kategórie ocenení sú nasledovné:

- Typ nosiča
- Zdoj úlovku
- Vlastník billboardu

V každej kategórii sú hraničné počty ocenení stanovené rôzne a adekvátne k danej kategórii. Medaily, ktoré používateľ získava sú bronzová, strieborná, zlatá a platinová, ktorá je najvyšším možným ocenením v každej kategórii.

Podľa celkového počtu ulovených billboardov používateľ tiež dosahuje určitých levelov, pričom má vždy zobrazený svoj aktuálny level v hlavičke celého webu, ale aj na tejto podstránke, kde má uvedenú tiež informáciu o počte zostávajúcich billboardov do ďalšieho levelu.

## **Štatistiky**

Táto podstránka je venovaná používateľovej poslednej aktivite. V úvodnej časti je zobrazený graf jeho aktivity, kde sú zobrazené jeho aktívne dni za posledný mesiac so znázorneným počtom úlovkov. Tento graf je vykreslený s použitím knižnice pChart, ktorá je určená práve na takéto úlohy. Na grafe je znázornený tiež rekordný deň s najväčšou aktivitou prihláseného používateľa,

ak samozrejme bol tento rekord dosiahnutý behom posledného mesiaca. Pre každý prípad je však tento rekord uvedený aj pod grafom slovne.

V ďalšej časti tejto stránky je zobrazená tabuľka prvých 10 najaktívnejších používateľov aplikácie. Pre udržanie anonymity sa zobrazujú len ich prvé mená. Táto tabuľka má taktiež motivačný charakter a má za úlohu, podobne ako ostatné sekcie venované gamifikácii, viesť používateľa k snahe dosiahnuť čo najväčší počet úlovkov.

### 5.1.10 ZLUČOVANIE NOSIČOV

Miniaplikácia funguje na základe podobných atribútov pri jednotlivých nahraných úlovkoch, kde stanovením pravdepodobnosti ponúka dvojice úlovkov, ktoré by mohli byť rovnaké (vzhľadom k nosiču). Algoritmus vyberania dvojíc najprv vyberie náhodný úloвок na mape, pomocou ktorého následne vyberie všetky ďalšie úlovky ležiace v určitej oblasti okolo náhodne vybraného úlovku. Následným porovnávaním rôznych atribútov (spoločnosť, ktorá reklamu umiestnila, typ nosiča, na ktorom sa reklama nachádza atď.) určí pravdepodobnosť pre každú dvojicu z náhodne vybraného úlovku a jedného zo skupiny blízko ležiacich bilbordov. Napokon sa náhodne vyberie jedna z týchto dvojíc, pričom dvojice s väčšou pravdepodobnosťou zhody majú väčšiu šancu, že budú vybrané, než tie ostatné.

“Vítazná” dvojica sa napokon zobrazí používateľovi, ktorý na 5-stupňovej škále zvolí svoj názor na to, či a s akou pravdepodobnosťou sú zobrazené úlovky rovnaké. Používateľ môže zvoliť jednu z možností určite áno - asi áno - neviem - asi nie - určite nie. Vzhľad miniaplikácie približuje obrázok nižšie.



## Zlučovanie úlovkov

Nastavenia účtu  
Moje úlovky  
Moje odznaky  
Zlučovanie billboardov

Sú na obrázkoch rovnaké nosiče\*?



určite áno

asi áno

neviem

asi nie

určite nie

\* pod nosičom rozumieme konštrukciu billboardu/reklamy je potrebné označiť rovnaké reklamy na základe ich **zhodnej polohy**, nie rovnakého "typu" reklamy

## 5.2 Android aplikácia

### 5.2.1 ZÁZNAM NEESTETICKEJ REKLAMY

#### Analýza

Aplikácia má slúžiť primárne na zachytávanie a ukladanie záznamov neestetickéj reklamy na server, kde sa budú zhromažďovať do databázy. Prvotná myšlienka bola, aby rozhranie pre používateľa bolo navrhnuté tak, aby s ním vedel rýchlo a intuitívne manipulovať. Na úvodnej obrazovke sme sa teda rozhodli používateľovi ihneď poskytnúť možnosť fotenia, nakoľko toto je jeho primárnym záujmom pri spustení mobilnej aplikácie. Používateľ teda môže zamieriť na cieľ, neestetickú reklamu, a odfotiť ho intuitívnym tlačidlom fotenia.

Pred uploadnutím fotky je vhodné, aby mal používateľ možnosť zadať rozširujúce informácie o reklame, ako napríklad typ nosiča reklamy alebo vlastníka reklamy. Túto obrazovku ponúkne



používateľovi, pričom bude mať možnosť rozhodnutia, či dané informácie chce zadať alebo nie. Ak nie, nechceme v používateľovi vyvolať pocit “nekompletnosti záznamu” a teda rozširujúce informácie budú ponúknuté len nenútenou formou.

Používateľ takisto nemusí súhlasiť so záznamom, ktorý odfoťil, a v tomto prípade by mal dostať možnosť zopakovať svoj pokus o zachytenie záznamu. Táto možnosť sa mu naskytne bezprostredne po odfoťení a bude sa môcť rozhodnúť, či fotenie opakovať, alebo postúpiť k uploadu záznamu.

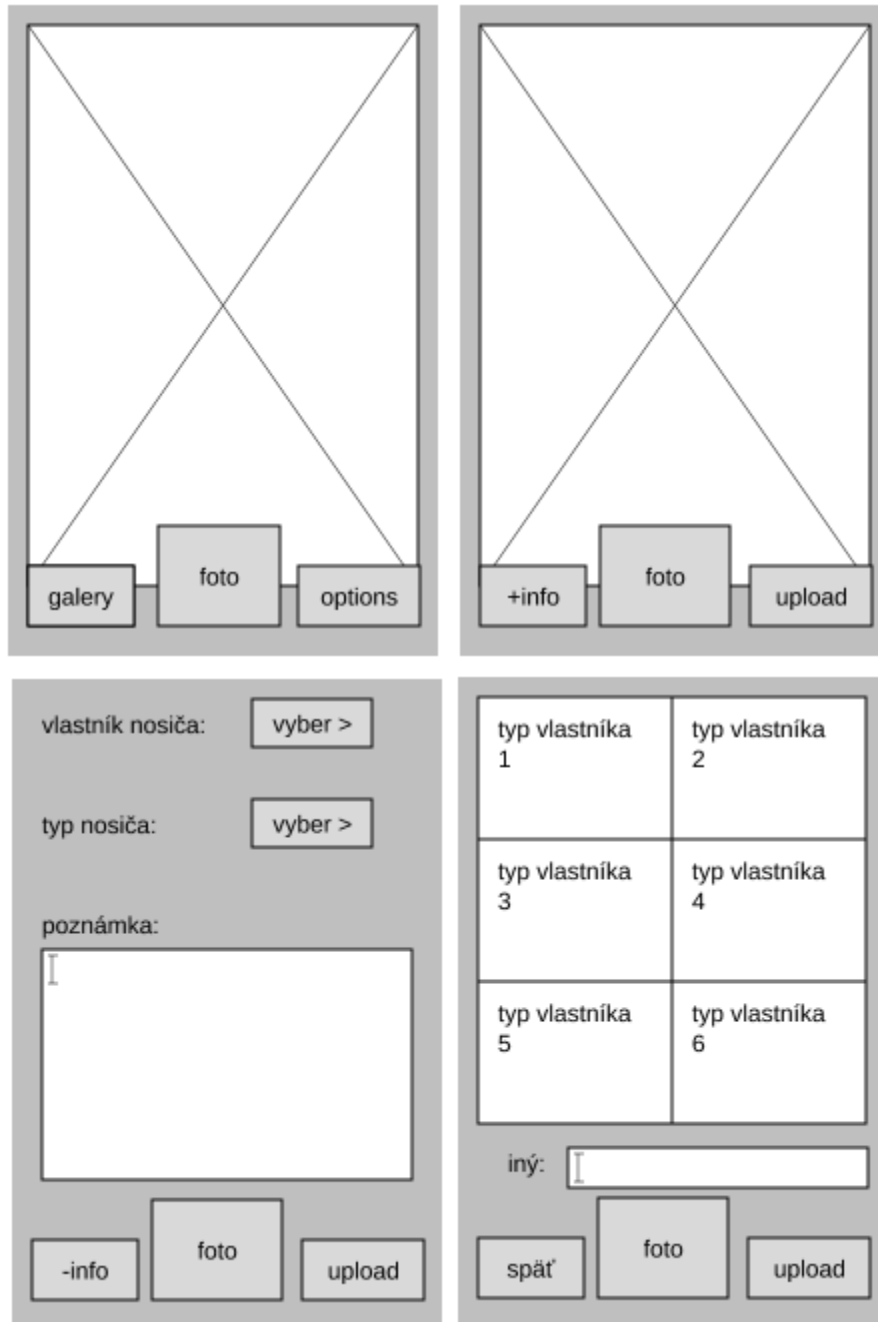
Ak sa používateľ ocitne mimo pripojenia na internet, je potrebné poskytnúť mu možnosť neskoršieho uploadu záznamu. Nie všade v teréne je totiž dostupný signál mobilného internetu a taktiež nie každý používateľ našej aplikácie bude mať k dispozícii tento spôsob pripojenia. Fotka teda počká uložená v mobilnom telefóne, až kým používateľ nemá možnosť pripojenia na internet. O čom bude upovedomený pomocou notifikácie.

## **Návrh**

Po odfoťení fotky používateľom sa daná fotka uloží do samostatného priečinka. Následne má používateľ možnosť rozhodnúť sa, či odfoťenú fotku, ktorá sa mu vyobrazí, považuje za vhodnú na odoslanie, a teda či ju chce odoslať alebo nie. Ak nechce, má možnosť vrátiť sa späť a pokračovať vo foteaní. Taktiež v prípade záujmu môže používateľ pridať rozširujúce informácie o fotke, ktoré ju pomôžu lepšie špecifikovať.

Pri foteaní reklám v teréne používateľ môže a nemusí mať pripojenie na internet. Aby aplikácia neodradila používateľov, ktorí majú väčšinu času svoje mobilné zariadenia v offline režime, aplikácia bezproblémovo funguje aj bez pripojenia na internet, avšak s tým rozdielom, že fotky sa ukládajú do samostatného priečinka, pričom pri najbližšom pripojení na internet je používateľ prostredníctvom notifikácie upovedomený o existujúcich neodoslaných fotkách a má ich možnosť odoslať.

Low-Fidelity návrhy obrazoviek mobilnej aplikácie:



## Implementácia

Aplikácia je implementovaná vo vývojom prostredí Android Studio, napísaná v programovacom jazyku Java, pričom UI je definované pomocou jazyka XML. Hlavnú časť aplikácie tvorí kamera. Nakoľko sme však potrebovali vyvinúť kameru "šitú na mieru", použili sme Android Camera API, ktorého použitie je vhodnou alternatívou k použitiu štandardnej vstavanej kamerovej aplikácie, pretože sme tým získali možnosť vytvoriť plne modifikovateľnú kamerovú aplikáciu. Aplikácia má

napr. nastavený automatický blesk a automatické zaostrovanie obrazu pre lepšiu kvalitu odfotených úlovkov.

Druhú nemenej dôležitú časť predstavuje komunikácia aplikácie so serverom. Táto komunikácia prebieha prostredníctvom REST služieb. Pri každom uploade fotky sa odosiela POST request zo strany mobilnej aplikácie, pričom server je pripravený na prijatie potrebných dát (ktoré zahŕňajú samotný bytestream fotky (vo formáte multipart/form-data) spolu so súradnicami miesta odfotenia a dodatočnými informáciami).

Aplikácia získava súradnice miesta odfotenia prostredníctvom GPS senzora, ktorý je zabudovaný v každom súčasnom Android telefóne. Nakoľko však funkcia lokalizácie štandardne nebýva zapnutá, používateľ je vyzvaný túto funkciu povoliť pri každom štarte aplikácie (pokiaľ už, samozrejme, daná funkcia nie je povolená).

Kvôli zjednodušeniu vývoja sme spodnú hranicu podpory nastavili na verziu Android 3.0 (čiže aplikácia podporuje všetky funkcie pre API 11+), nakoľko pre staršie verzie je pri implementácii potrebné využívať viaceré podporné funkcie.

Zdrojový kód aplikácie dodržiava zásady objektovo-orientovaného programovania, pričom zachováva pravidlá písania kódu pre Android podľa oficiálnej Android dokumentácie a taktiež zohľadňuje metodiku správneho písania zdrojového kódu, ktorá je súčasťou našich interných tímových metodík.

Po odfotení fotky užívateľom sa daná fotka uloží do samostatného priečinka. Následne je užívateľ vyzvaný rozhodnúť sa, či danú fotku považuje za vhodnú na odoslanie, a teda či ju chce odoslať alebo nie. Ak nechce, má možnosť zrušiť dialógové okno a pokračovať vo foteaní.

Pri foteaní reklám v teréne používateľ môže a nemusí mať pripojenie na internet. Aby aplikácia neodradila užívateľov, ktorí majú väčšinu času svoje mobilné zariadenia v offline režime, aplikácia bezproblémovo funguje aj bez pripojenia na internet, avšak s tým rozdielom, že fotky sa ukladajú do samostatného priečinka, pričom pri najbližšom pripojení na internet je používateľ prostredníctvom notifikácie upovedomený o existujúcich neodoslaných fotkách a má ich možnosť odoslať.

Keďže používatelia majú vo všeobecnosti dve možnosti pripojenia na internet cez mobilné zariadenia (prostredníctvom Wi-Fi alebo mobilného operátora), aplikácia tiež obsahuje obrazovku s nastaveniami, kde si môže používateľ vybrať, či chce, aby sa jeho úlovky po kliknutí na tlačidlo "Odoslať" odosieli aj vtedy, ak je práve pripojený k internetu prostredníctvom mobilného operátora alebo nie. Dôvodom pre povolenie odosielania úlovkov iba počas pripojenia na Wi-Fi môže byť obmedzený povolený počet prenesených dát pri pripojení cez mobilného

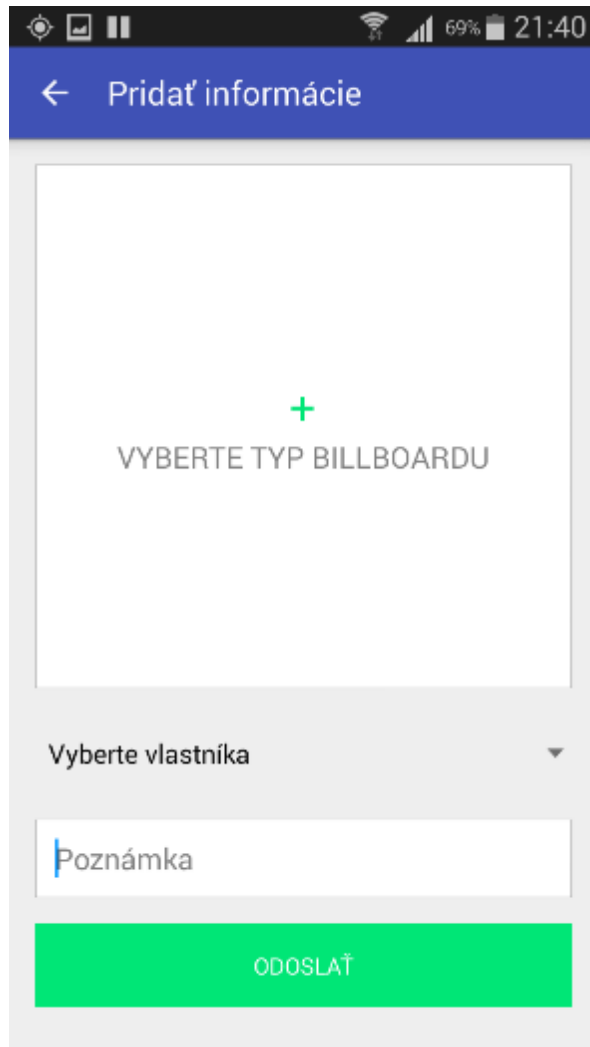
operátora. Nakoľko však zachytené obrázky predsa len nie sú extrémnej veľkosti, dali sme prostredníctvom tejto obrazovky používateľom na výber.

Celá aplikácia je obalená v novom Android 5.0 dizajne. Na to sme použili knižnicu z radu AppCompat, ktorá tento dizajn aplikuje aj na nižšie verzie operačného systému. Zobrazuje tiež pravidelne aktualizované údaje o latitúde a longitúde, resp. či je GPS signál dostupný alebo nie.

Screenshots z mobilnej aplikácie pre odfotenie úlovky:



Screenshots z mobilnej aplikácie pre pridanie voliteľných informácií:



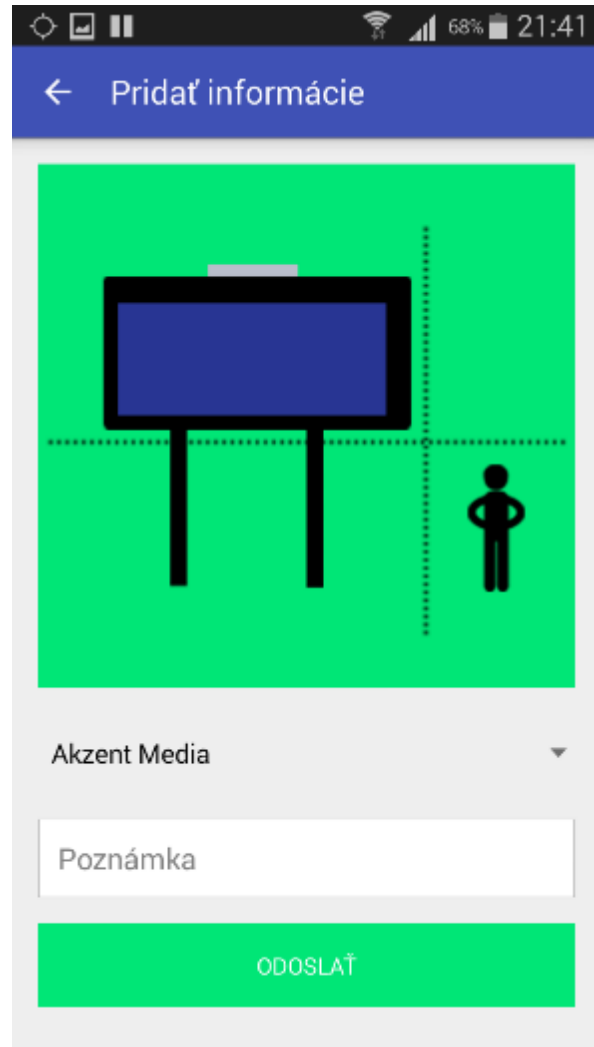
← Pridať informácie

+  
VYBERTE TYP BILLBOARDU

Vyberte vlastníka ▾

Poznámka

ODOSLAŤ



← Pridať informácie

Akzent Media ▾

Poznámka

ODOSLAŤ

## 5.2.2 OFFLINE REŽIM

### Analýza

Pred návrhom modulu bola spravená analýza s potenciálnymi používateľmi. Z analýzy vyplynula potreba návrhu modulu tak, aby mal nad ňou používateľ plnú kontrolu. Napríklad aby aplikácia nezačala upload nazbieraných úlovkou na server svojvoľne, hneď po pripojení online, pretože používateľovi to v danú chvíľu nemusí vyhovovať, napríklad, ak má práve pomalé pripojenie.

Potreba vhodnej spätnej väzby o stave aplikácie, ktorá v dostatočnej miere používateľa informuje, avšak nie je ani príliš otravná.

Používateľská analýza preukázala tiež nejednoznačné názory používateľov na offline režim pri aktívnych mobilných dátach.

## **Návrh**

Zariadenie nemá aktívne internetové pripojenie (wifi alebo mobilné dáta) pri pokuse o odoslanie úlovku na server. V takom prípade sa v aplikácii aktivuje tzv. offline režim a úlovok sa uloží do pamäte mobilného zariadenia. Používateľ bude o stave aplikácie a úlovku informovaný pomocou vhodnej spätnej väzby, tj. dostane informáciu o tom, že je práve v offline režime a jeho úlovok bude odoslaný neskôr.

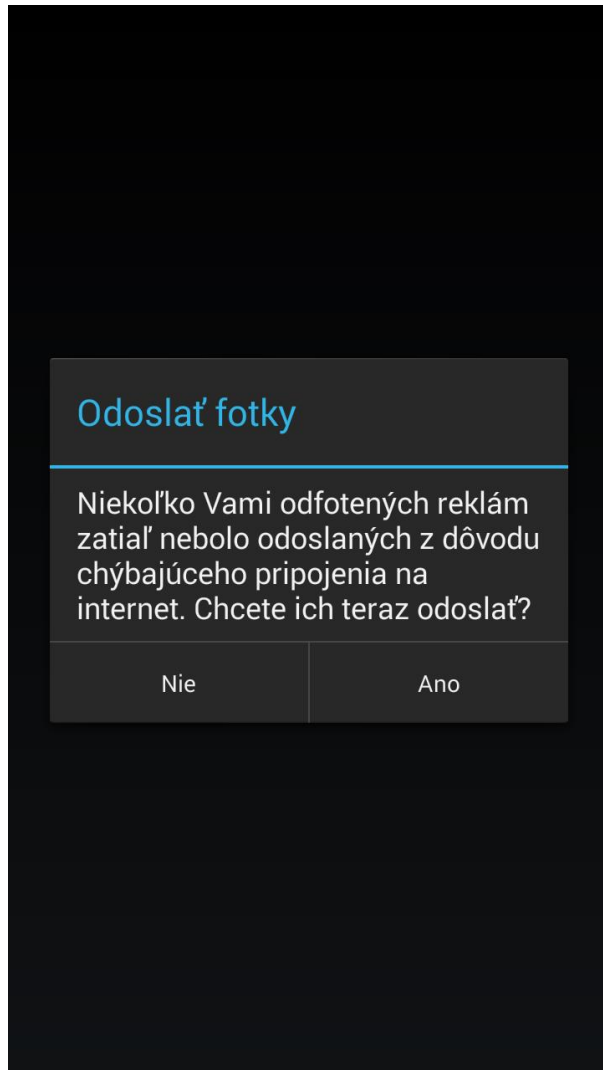
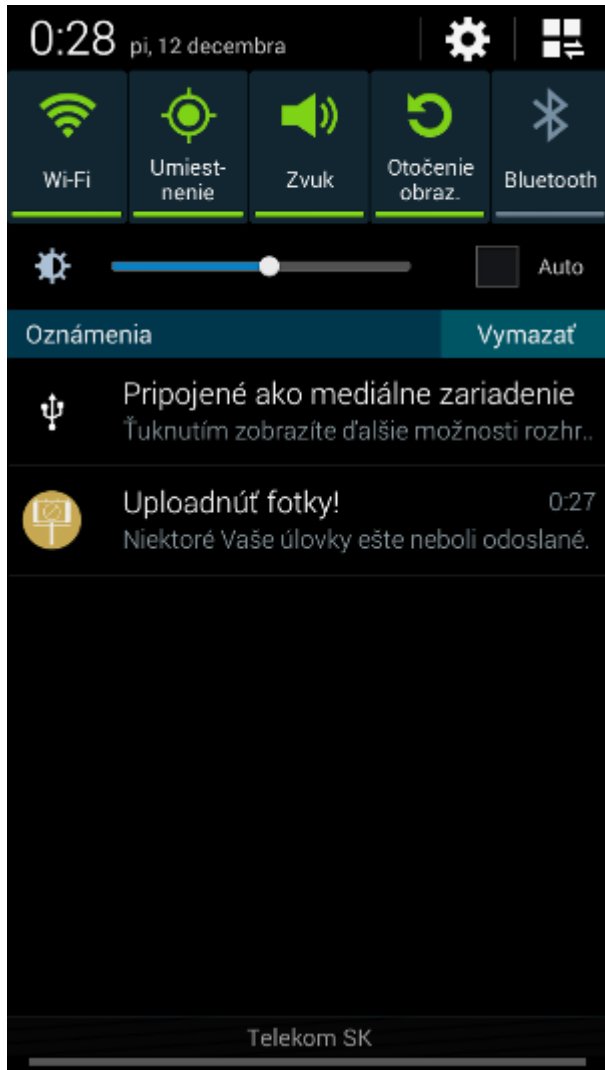
Pri rozpoznaní aktívneho internetového pripojenia sa zobrazí notifikácia, ktorá používateľa informuje o možnosti odoslania uložených úlovkov na server. Používateľ kliknutím na notifikáciu spustí aplikáciu, ktorá automaticky odošle všetky doposiaľ neodoslané príspevky na server.

Vyplývajúc z analýzy sme navrhli aj možnosť nastavenia používateľom, či si želá aktivovať offline režim aj v prípade, že nie je dostupné wifi pripojenie, ale sú aktívne mobilné dáta.

## **Implementácia**

Modul bol doimplementovaný ako rozšírenie k funkcionalite z predchádzajúcej verzie, konkrétne modul záznam neestetickéj reklamy. Pri implementácii sme sa pridržali návrhu v pôvodnom znení. Možnosť nastavenia offline režimu aj pri používaní mobilných dát však bola presunutá do budúcej verzie aplikácie z dôvodu opravovania niekoľkých chýb odhalených testovaním.

Screenshots z mobilnej aplikácie pre offline režim:



## Testovanie

Offline režim bol testovaný na niekoľkých rôznych Android zariadeniach. Pri testovaní bolo odhalených niekoľko chýb:

- padanie aplikácie po minimalizácii za účelom aktivovania GPS lokalizácie
- zobrazenie notifikácie pri pripojení online aj v prípade, že neexistujú žiadne neodoslané úlovky

Všetky tieto odhalené chyby používateľským testovaním priamo na zariadeniach boli zapracované a úspešne opravené.

## 6 POUŽÍVATEĽSKÁ PRÍRUČKA


### 6.1 Mobilná aplikácia

Rozhranie mobilnej aplikácie je veľmi jednoduché a intuitívne.


#### 6.1.1 PRIHLÁSENIE A ODHLÁSENIE

Používateľ sa prihlási hneď po prvom spustení aplikácie pomocou zobrazeného formulára. Do položky e-mail zadáva e-mailovú adresu, pomocou ktorej sa registroval.






Používateľ sa z aplikácie odhlási kliknutím na túto ikonu , ktorá sa nachádza na hlavnej obrazovke.

#### 6.1.2 NASTAVENIA OFFLINE REŽIMU

Používateľ zmení svoje nastavenia pre offline režim po kliknutí na ikonu ozubeného kolieska  nachádzajúceho sa na hlavnej obrazovke. Následne sa aplikácia prepne do nastavení, kde používateľ môže ľubovoľne zmeniť svoje nastavenia a zmeny uložiť.


#### 6.1.3 ODFOTENIE A ODOSLANIE ÚLOVKU

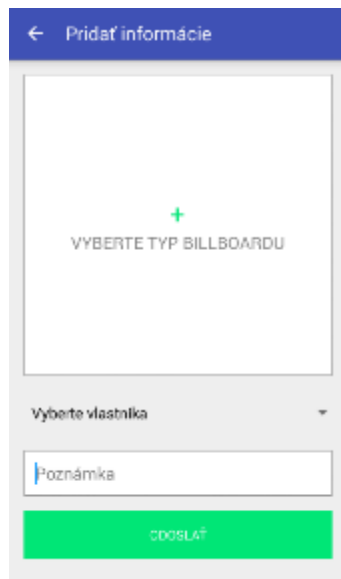
Používateľ úlovok môže odfotiť hneď po spustení aplikácie. V prípade, že ju spúšťa prvý krát, je potrebné sa najprv prihlásiť. Používateľ úlovok odfotí stlačením tlačidla fotoaparátu .

Následne používateľ môže úlovok odoslať na server pomocou tlačidla uploadu . V prípade, že používateľ nie je spokojný s odfoteným obrázkom, môže sa vrátiť späť na fotoaparát pomocou tlačidla refresh  a vyfotiť nový obrázok.

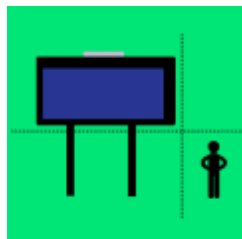


## 6.1.4 PRIDANIE ROZŠIRUJÚCICH INFORMÁCIÍ K ÚLOVKU

Používateľ môže ešte pred odoslaním pridať rozširujúce informácie k úlovk, po kliknutí na tlačidlo . Aplikácia zobrazí obrazovku pre pridanie rozširujúcich informácií.



Používateľ kliknutím na časť obrazovky označenú ako „vyberte typ billboardu“ zobrazí zoznam typov nosičov. Kliknutím na jeden z ponúkaných obrázkov vyberie konkrétny typ nosiča. Napríklad:



Kliknutím na „vyberte vlastníka“ používateľ zobrazí drop down menu pre výber vlastníka alebo majiteľa nosiča. Tento vlastník je uvedený na každom legálne osadenom nosiči, na viditeľnom mieste.

Do položky poznámka používateľ vyplní ľubovoľné doplňujúce informácie. Pomocou tlačidla „ODOSLAŤ“ používateľ odošle svoj úlovk na server.

## 6.2 Webová aplikácia

### 6.2.1 REGISTRÁCIA

Používateľ zobrazí registračný formulár po kliknutí na tlačidlo „REGISTRÁCIA“. Do registračného formulára vyplní svoje kontaktné informácie.



Po poľa „E-mail“ používateľ zadáva svoju e-mailovú adresu, ktorá bude slúžiť zároveň ako jeho prihlasovanie meno. Do poľa pod ním vyplní svoje heslo, ktoré chce používať pri prihlasovaní. Do polí „Meno“ a „Priezvisko“ vyplní svoje meno a priezvisko. Používateľ registráciu ukončí kliknutím na tlačidlo „Registrovať“.

## 6.2.2 PRIHLÁSENIE A ODHLÁSENIE

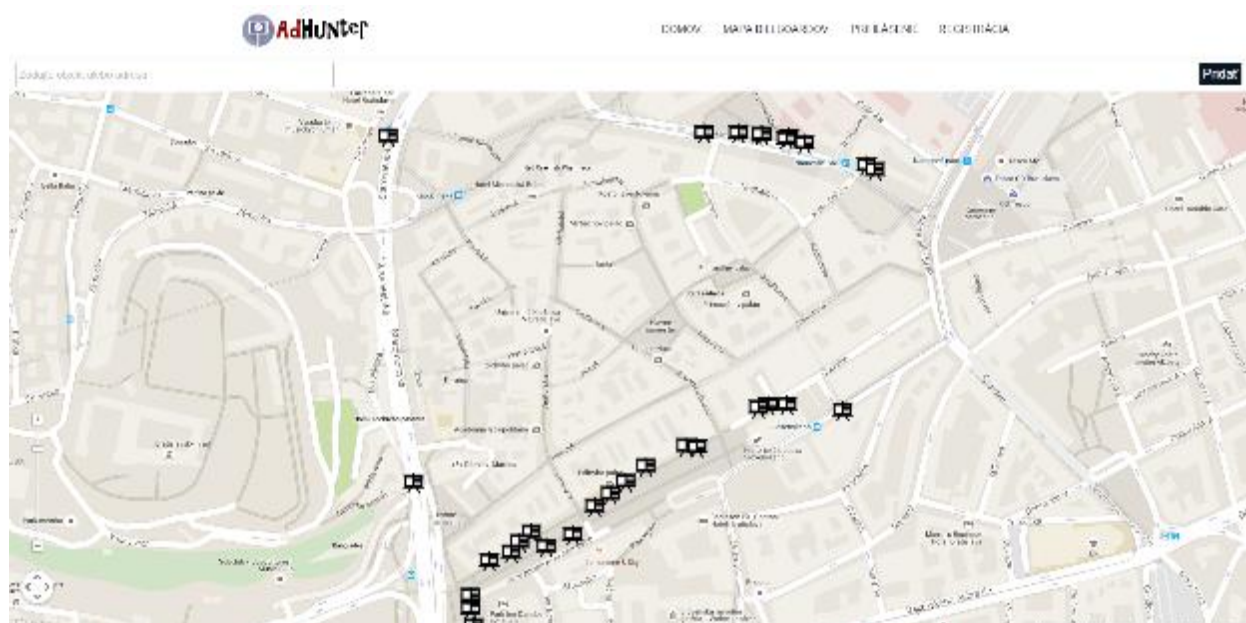
Používateľ zobrazí prihlasovací formulár kliknutím na tlačidlo „PRIHLÁSENIE“. Do polí vyplní svoj e-mail, pod ktorým sa registroval a heslo, ktoré vyplnil pri registrácii. Používateľ dokončí prihlásenie kliknutím na tlačidlo „Prihlásiť“.



The image shows a login form with a red header containing the word "Prihlásenie" in white text. Below the header are two input fields: the first is labeled "E-mail" and the second contains masked characters "\*\*\*\*\*". At the bottom of the form is a button labeled "Prihlásiť".

## 6.2.3 MAPA

Používateľ zobrazí mapu so všetkými úlovkami pomocou tlačidla „MAPA BILLBOARDOV“.



Používateľ vyfiltruje na mape iba svoje vlastné úlovky pomocou tlačidla „Zobraziť iba moje billboardy.“

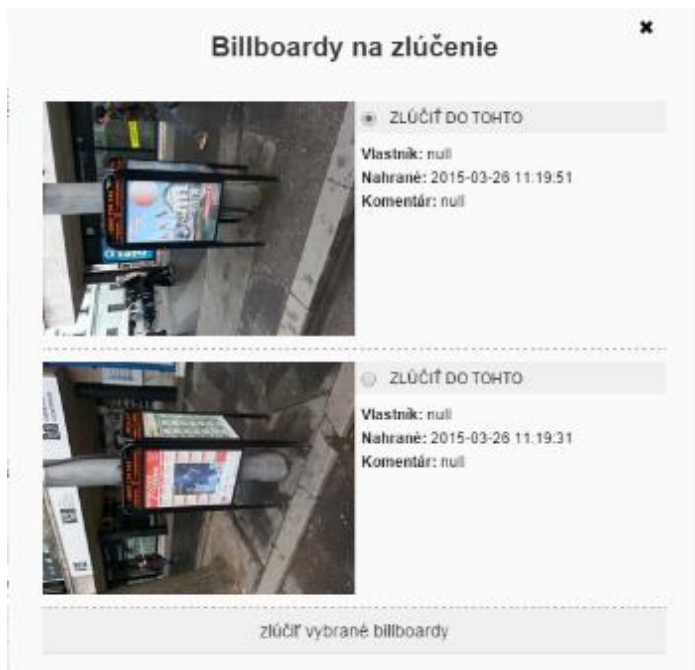
### Zobrazenie a upravenie detailu úlovku

Používateľ zobrazí detaily vybraného úlovku kliknutím na jeho ikonku priamo na mape. Používateľ môže svoj vlastný úlovok upraviť kliknutím na tlačidlo „Upraviť údaje“. Aplikácia následne zobrazí postranný panel, kde používateľ upraví detaily a zmeny uloží pomocou tlačidla „Uložiť.“ Používateľ môže spresniť polohu úlovku kliknutím na tlačidlo „Premiestniť“ v postrannom menu a následne kliknutím na mapu uloží úlovok na jeho novú pozíciu. Používateľ úlovok zmaže použitím tlačidla „Zmazať.“



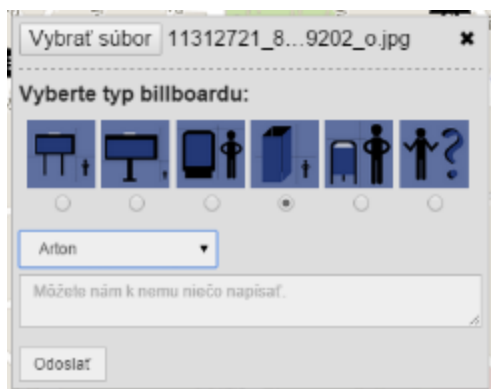
### Návrh na zlúčenie úlovkov

Používateľ môže navrhnúť zlúčenie viacerých úlovkov do jedného v prípade, že sa na dvoch úlovkoch nachádza jeden a ten istý reklamný nosič. Po zobrazení detailu úlovku používateľ klikne na tlačidlo „zlúčiť“. Aplikácia zobrazí postranné menu s prvým označeným nosičom. Používateľ klikaním na detail ďalších nosičov a následne na tlačidlo „zlúčiť“, pridáva ďalšie úlovky do návrhu na zlúčenie. Všetky takto označené úlovky sa zobrazujú v postrannom paneli. Používateľ nakoniec zaklikne hlavný úlovok, čiže ten, do ktorého sa ostatné zlúčia. Svoj návrh uloží pomocou tlačidla „zlúčiť vybrané billboardy“.



## Pridanie úlovku

Používateľ vloží nový úlovok kliknutím na tlačidlo „Pridať“ a následným kliknutím pomocou kurzora na konkrétne miesto na mape v lokalite, kam chce úlovok umiestniť. Po kliknutí na miesto na mape sa mu zobrazí formulár, pomocou ktorého vyberie fotografiu úlovku a voliteľne môže zadať doplňujúce informácie o úlovku. Typ nosiča na základe intuitívnych znázorňujúcich obrázkov, vlastníka reklamného nosiča z drop down menu a ľubovoľnú poznámku. Používateľ proces dokončí kliknutím na tlačidlo „Odoslať.“

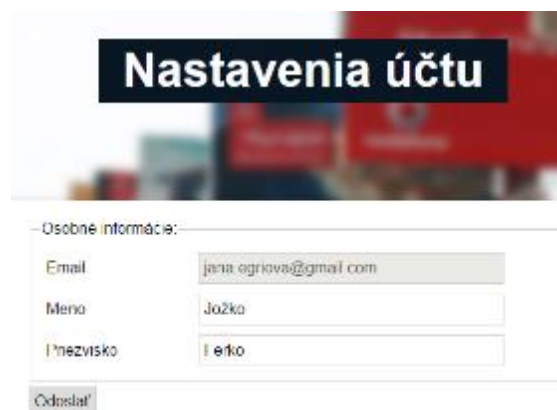


## Profil používateľa

Používateľ zobrazí svoj profil kliknutím na tlačidlo „PROFIL“.

### Zmena informácií

Na záložke „Nastavenia účtu“ vľavo sa nachádzajú jeho aktuálne informácie, ktoré môže hneď aj aktualizovať vyplnením aktuálnych údajov do formulára v časti „Osobné informácie“ a stlačením tlačidla „Odoslať“.



Osobné informácie:

Email: jana.ogriova@gmail.com

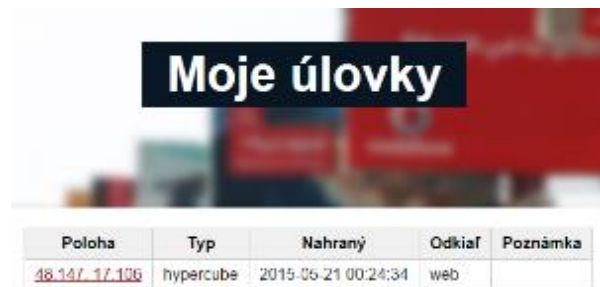
Meno: Jozko

Priezvisko: Ierko

Odoslať

## Moje úlovky

Používateľ zobrazí prehľad svojich vlastných nahraných úlovkov na záložke „Moje úlovky“ vľavo. Zoznam úlovkov je zobrazený formou tabuľky.



Poloha	Typ	Nahraný	Odkiaľ	Poznámka
48.147.17.106	hypercube	2015-06-21 00:24:34	web	

Tabuľka má nasledovné stĺpce:

- Poloha – vyjadruje GPS súradnice umiestnenia úlovku spolu s odkazom na miesto na mape
- Typ – vyjadruje typ reklamného nosiča
- Nahraný – udáva presný dátum a čas, kedy bol úlovok uploadnutý

- Odkiaľ – informácia, či bol úlovok uploadnutý prostredníctvom mobilnej alebo webovej aplikácie
- Poznámka – informácie, ktoré používateľ vložil pri uploadovaní úlovku do poznámky

### **Moje úspechy**

Používateľ si môže prehliadať svoje rebríčky, štatistiky a získané odznaky na záložke „Moje odznaky“.

### **Riešenie úlohy**

Používateľ môže riešiť úlohu určenia, či sú dva zobrazené úlovky jeden a ten istý nosič na záložke „Zlučovanie bilbordov“. Používateľovi sa zobrazia dva úlovky, u ktorých je percentuálna šanca, že je to rovnaký nosič. Používateľ obrázky vyhodnotí a určí ich zhodu pomocou nasledovnej mierky.

určite áno

asi áno

neviem

asi nie

určite nie

## 7 TECHNICKÁ DOKUMENTÁCIA

- Vygenerovaná dokumentácia mobilnej časti (Javadoc)
- Vygenerovaná dokumentácia webovej časti (phpDoc)