

Mendelova univerzita v Brně
Provozně ekonomická fakulta

Tvorba geografického informačního systému arboreta MENDELU

Bakalářská práce

Vedoucí práce:
Ing. David Procházka, Ph.D.

Bc. David Ruber

Brno 2014

Zde prostor pro zadání práce

Rád bych poděkoval svému vedoucímu práce Davidu Procházce za ochotu,
odborné konzultace v rámci této práce.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Tvorba geografického informačního systému arboreta MENDELU**

vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 15. ledna 2017

.....

Abstract

Bc. Ruber, D. Creating a geographic information system of MENDELU arboretum.
Diploma thesis. Brno: MENDELU, 2016

abstrakt

Keywords

klicova slova

Abstrakt

Bc. Ruber, D. Tvorba geografického informačního systému arboreta MENDELU.
Diplomová práce. Brno: MENDELU, 2016

abstrakt

Klíčová slova

klicova slova

Obsah

| | | |
|----------|---|-----------|
| 1 | Úvod a cíl práce | 7 |
| 1.1 | Úvod | 7 |
| 1.2 | Cíl práce | 7 |
| 2 | Dostupný software | 8 |
| 2.1 | Django | 8 |
| 2.1.1 | GeoDjango | 8 |
| 2.2 | Google Maps | 8 |
| 2.3 | Leaflet | 9 |
| 2.4 | OpenLayers | 9 |
| 2.5 | Srovnání Leaflet a OpenLayers | 9 |
| 2.6 | PHP Brick\Geo | 10 |
| 3 | Crowdfunding | 11 |
| 3.1 | Výhody crowdfundingu | 11 |
| 4 | Návrh databáze | 12 |
| 4.1 | Popis tabulek | 12 |
| 5 | Závěr | 17 |
| 6 | Reference | 18 |
| | Přílohy | 19 |

1 Úvod a cíl práce

1.1 Úvod

1.2 Cíl práce

2 Dostupný software

2.1 Django

Django je webový framework napsaný v jazyce Python. Byl vyvinut aby pomohl vývojářům urychlit vývoj aplikací od základního konceptu až po finální dokončení. K těmto účelům poslouží jak oficiální návod krok po kroku, ve kterém jsou popsány všechny základní kroky, tak i volně dostupná dokumentace obsahující všechny specifiky. Django obsahuje velké množství již kompletně připravených modulů, které lze využít, jako jsou například autentizace uživatelů, mapu stránek, RSS kanál a další. Ověřování uživatelů je jedním z bezpečnostních prvků, které usnadňují práci vývojářům. Mezi další bezpečnostní prvky patří SQL injection¹, cross-site scripting², cross-site request forgery³ a clickjacking⁴. Django využívá architekturu "sdílené nic", což znamená že každý uzel je soběstačný a nezávislý. Přesněji žádný z uzlů nesdílí paměť nebo diskové úložiště. Framework odděluje jednotlivé komponenty (datábázovou vrstvu, aplikační vrstvu atd.).

2.1.1 GeoDjango

GeoDjango je samostatný modul pro Django, který obsahuje geografické rozšíření pro webové aplikace. Modul usiluje o to, aby byl, stejně jako samostatný Django framework, co nejvíce jednoduchý pro vývojáře. Obsahuje vlastní modelář pro OGC⁵ geometrii a rastrová data. Rozšiřuje Django ORM⁶ pro potřeby geografických požadavků a prostorových dat. Python rozhraní pro geometrii, rastrové operace a práci s daty v různých formátech je na vysoké úrovni.

2.2 Google Maps

Nejznámější mapovou službou jsou bezpochyby Google Maps. První verze Google Maps API vznikla v roce 2005. API není jen JavaScript pro koncového uživatele, zahrnuje více projektů. Nabízí několik různých způsobů jak službu vložit do webových stránek s hodně možnostmi přizpůsobení. Dostupné API služby zahrnují například Google Maps API Javascript, Google Static Maps API, Google Maps SDK pro iOS nebo Google Maps pro Android. Základní API jsou zdarma, ale platí určitá omezení. Prvním je, že mapy musí být volně a veřejně přístupné. Druhým je omezení využití, které omezuje to, aby web negeneroval více než povolené denní množství map. Existuje i placené řešení API.

¹SQL injection je technika napadení databázové vrstvy prostřednictvím neošetřeného vstupu.

²Metoda využívající bezpečnostní chyby ve skriptech a vložení vlastních skriptů.

³Metoda pracující na bázi nezamýšleného požadavku pro vykonání určité akce.

⁴Způsob útoku kdy uživatel spustí akci, kterou nepředpokládal.

⁵Open Geospatial Consortium je mezinárodní standartizační organizace.

⁶Objektově relační zobrazení je programovací technika zajišťující konverzi dat mezi relační databází a objektově orientovaným programovacím jazykem

2.3 Leaflet

Leaflet je nejrozšířenější open-source knihovna v jazyce JavaScript pro tvorbu interaktivních map. Obsahuje většinu potřebných mapových funkcí, jako například:

- podkladové mapy, WMS⁷
- vektorové vrstvy: polygony, linie atd.
- mapové značky
- GeoJSON

Knihovna obsahuje již naimplementované prvky pro ovládání mapy, mezi které patří gesta prsty nebo dvojité kliknutí myši pro přiblížení, ovládání skrze klávesnici (pohyb pomocí šipek), nebo připravené ovládací panely zobrazené přímo v mapě. Leaflet je navržen tak, aby byl jednoduchý, výkonný a použitelný. Jeho hlavním kladem je funkčnost skrze všechny internetové prohlížeče stolních počítačů i mobilních zařízení. Samotné jádro knihovny klade důraz na jednoduchost, výkonnost a použitelnost.

2.4 OpenLayers

OpenLayers je stejně jako Leaflet jedna z nejrozšířenějších open-source JavaScript knihoven pro dynamické mapy umístěné na webových stránkách. Byl vyvinut společností MetaCraft jako ekvivalent ke Google Maps. Umožňuje zobrazovat podkladové mapy, vektorová data, mapové značky a další. Podkladové mapy je možná načíst z mnoha zdrojů, např: OSM, Bing, MapBox. Stejně tak jsou podporovány OGC mapové služby. OpenLayers umožňuje i vykreslení vektorových vrstev GeoJSON, KML, GML a dalších formátů.

2.5 Srovnání Leaflet a OpenLayers

Porovnat Leaflet s OpenLayers je v celku jednoduché. Leaflet je zaměřen na jednoduchost a proto jeho použití je jednodušší. Je správným řešením pro jednoduché běžné mapové úkoly jako je zobrazení mapových podkladů a částí, posouvání mapy nebo zoomování. Celkové API je na vyšší úrovni, což znamená méně kódu a je jednodušší na pochopení, není tak rozsáhlé jako u OpenLayers. Openlayers nabízí více funkcí ovšem za cenu většího množství kódu, nutnosti inicializace a nastavení mapy. API je na nižší úrovni a proto je těžší na pochopení a naučení se potřeb pro použití. Díky inicializaci je dosažena vyšší kontrola nad mapou a daty. V rámci kontroly je dobře zvládnuta podpora editace. Leaflet nabízí méně funkcí, ale existuje mnoho plug-in⁸, které chybějící prvky doplní.

⁷Web Map Service je otevřený standart pro sdílení geografických informací.

⁸Plug-in je zásuvný modul, který doplňuje aplikaci.

2.6 PHP Brick\Geo

3 Crowdfunding

Crowdfunding je způsob získávání peněz na projekt formou veřejné sbírky. Nejedná se o tradiční přístup k financování podnikatelské činnosti. Většinou je nutné provést průzkum trhu, nastavit svůj podnikatelský plán vytvořit prototyp a následně produkt propagovat před investory. Zdroje financování jsou pak banky, investoři, nebo společnosti poskytující kapitál. Ve výsledku podobným přístupem je na konci omezené množství investorů. Crowdfunding tvoří přesný opak, na konci lze mít velké množství investorů. Pro získání peněz postačí vytvořit profil produktu na jedné z mnoho webových stránek, nabídnout za příspěvek například první vyrobený kus, oficiální uvedení přispěvatele nebo jiné bonusy a potom jen čekat na příspěvky, dokud není naplněn základní kapitál.

3.1 Výhody crowdfundingu

Existuje mnoho výhod oproti tradičním metodám, uvedené jsou jen některé z nich. Při použití crowdfunding platformy je získán přístup k tisícům akreditovaných investorů, kteří mohou komunikovat a dále předávat prezentovaný produkt. Vytvoření prezentace pomůže mapování historie vývoje produktu, jeho statistik z pohledu úspěšnosti a hlavně jako zpětná vazba od obyčejných lidí (investorů), kteří mohou vyjádřit svůj názor a tím přispět k novým vylepšením. Přístup získává na popularitě zejména díky sociálním sítím, na kterých se projekty mohou snadno šířit a díky tomu získávat více potencionálních přispěvatelů. Kromě sociálních sítí lze projekt šířit pomocí emailových newsletterů nebo vlastní webovou prezentací. Stejně jako klasických metod pro zahájení výroby, tak i typů crowdfundingu existuje více. Mezi základní patří:

- Crowdfunding založený na darech - každá kampaň, ve které není finanční návratnost pro investory
- Crowdfunding založený na odměnách - kampaň, která zahrnuje jednotlivce přispívající na produkt výměnou za odměnu (výrobek nebo služba)
- Crowdfunding založený na vlastním kapitálu - kampaň, která umožňuje přispěvatelům aby se stali součástí vlastníků společnosti

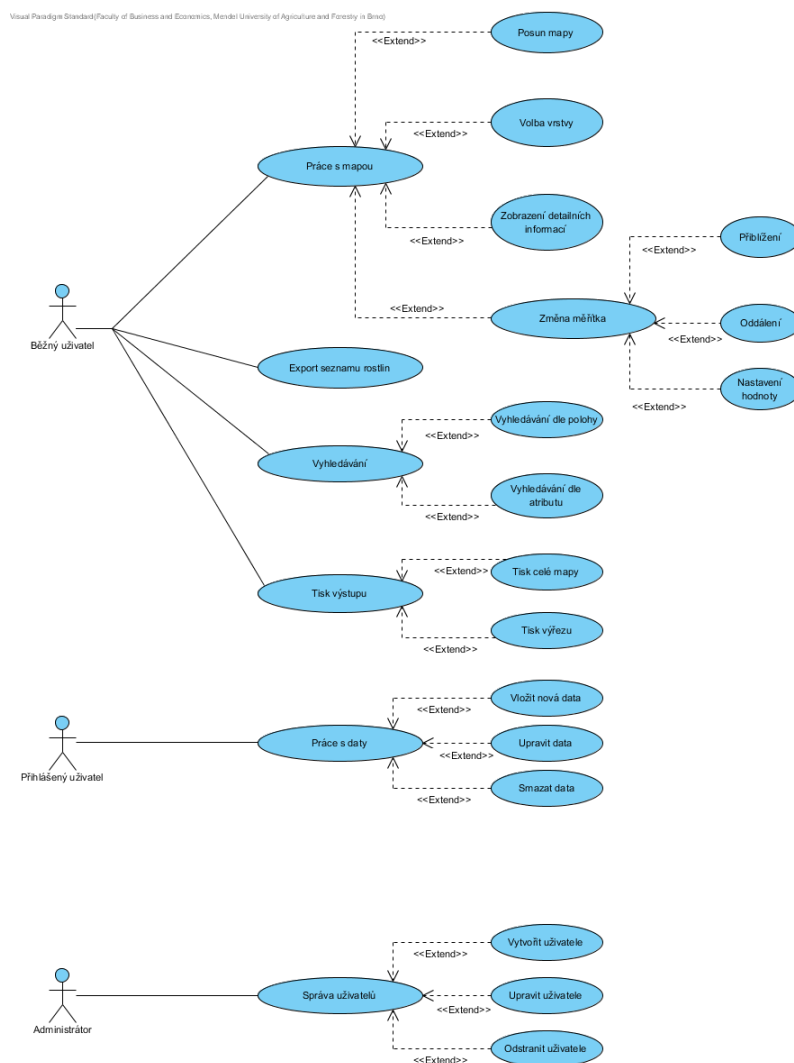
4 Návrh informačního systému

Před započítím samotné implementace by se měl sestavit návrh informačního systému, pro geografický informační systém to platí také.

4.1 Případy užití

Use Case model je jedním z UML modelů. V překladu lze diagram označit jako diagram případů užití. Jak již z názvu vyplývá, jedná se o popis chování systému z pohledů jeho uživatelů, proto je vytvářen mezi prvními diagramy. Model samotný se skládá z několika entit, mezi které patří actors(aktéři), use case(případy užití) a vztahy mezi nimi. Definují se zde interakce mezi rolemi a systémem, kde role nemusí být pouze uživatel jako člověk, ale také jiný systém.

Jednotlivé případy užití mohou být složeny z více akcí vedoucích k dosažení požadovaného cíle. Případ užití je vykreslován jako elipsa s názvem. Aktéři jsou zobrazeni jako jednoduché postavy, které komunikují s případy užití.



Obrázek 1: Use Case.

5 Návrh databáze

Kompletní aplikace by měla být postavena na kvalitním databázovém návrhu. Návrh vychází z bakalářské práce Geodatabáze pro GIS arboreta MENDELU. Protože databáze je hlavním základním kamenem a bez její správné struktury by aplikace mohla být například pomalá nebo špatně udržitelná vzhledem k uložení obsahu. Návrh, z kterého bylo vycházeno, obsahoval nedostatky, které bylo nutné eliminovat. Prvním špatným aspektem bylo složité rozšíření jazykových mutací aplikace. Původně byly zamýšleny pouze dvě mutace (česká a slovenská) s uchováním latinského názvu rostlin a jejich hierarchické struktury. Moderní aplikace by měli splňovat jazykovou rozšiřitelnost, proto bylo nutné návrh upravit do uvedené potřeby. Samotné názvy nejsou uchovávány přímo v tabulkách rostlina, třída, atd., ale ve vazebních tabulkách spojující požadovanou tabulku (rostlina, třída, atd.) s tabulkou jazyk. Tabulka jazyk umožňuje uchovat jazyky, s kterými aplikace bude pracovat. Po provedení výše popsaných změn je již jednodušší rozšířit aplikaci o další jazyk, samozřejmě za předpokladu doplnění názvů všech uložených dat.

Další problémovou částí návrhu byla hierarchie zařazení rostlin. Biologická klasifikace rostlin je způsob, jakým se kategorizují rostliny. Věda zabávající se právě tříděním organismů se označuje taxonometrie, která kromě třídění zkoumá i vzájemné příbuznosti a podobnosti. V současnosti je používán hierarchický systém založený na Linneově klasifikačním systému, který vytvořil švédský biolog Carl Linné, považovaný za otce moderní biologie. Systém klasifikace, používaný v současné době, je doplněn o nové poznatky a je univerzální pro všechny organismy. Jednotlivé organismy vzájemně ražené do různých úrovní (říše, třída, řád, atd.) se nazývají taxony, kdy vyšší taxony mohou obsahovat jeden, nebo více taxonů nižší úrovně. Jednotlivé taxonomické jednotky se nazývají:

- říše
- podříše
- oddělení
- třída
- podtřída
- řád
- podřád
- nadčeleď
- čeleď
- podčeleď
- rod

- podrod
- druh
- poddruh
- odrůda

5.1 Popis tabulek

Z prvního požadavku je vytvořena tabulka `rostlina` :

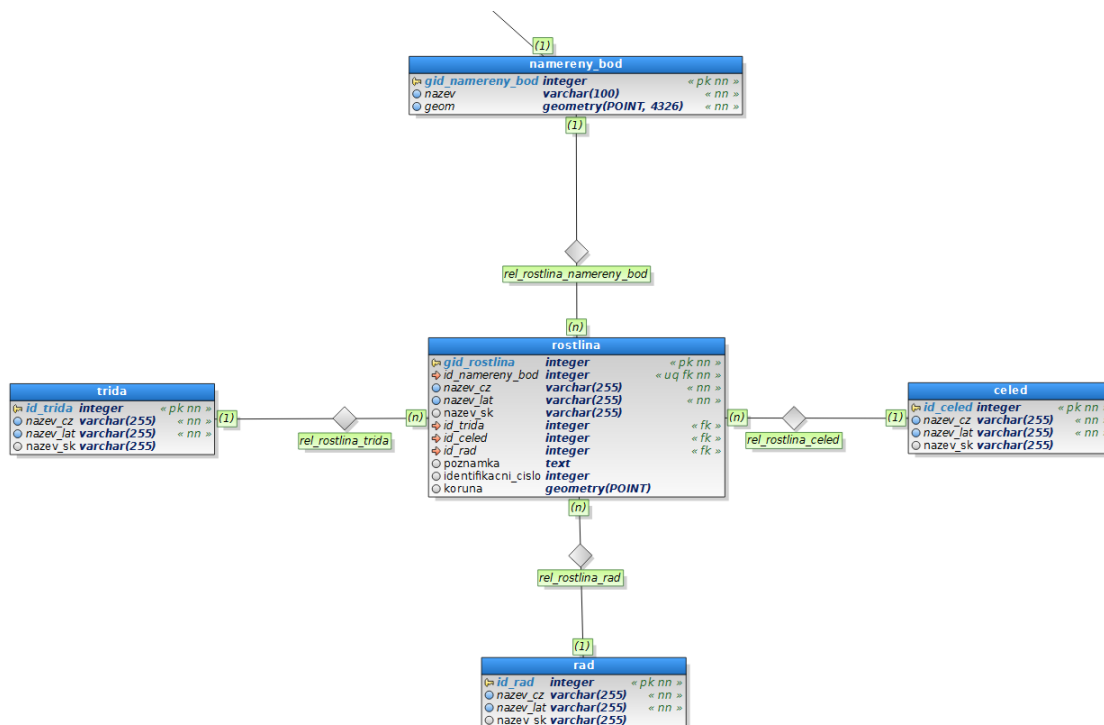
rostlina

- Sloupce: `gid_rostlina` (serial), `id_namereny_bod` (integer), `nazev_cz` (varchar), `nazev_lat` (varchar), `nazev_sk` (varchar), `id_trida` (integer), `id_celed` (integer), `id_rad` (integer), `poznamka` (text), `identifikacni_cislo` (varchar), `koruna` (point)
- Primární klíč: `gid_rostlina`
- Popis: Přiřazené informace k naměřenému bodu. Názvy rostliny se mohou vyskytnout v české, latinském a slovenském jazyce. Rostliny mají přiřazenou třídu, čeleď a řád, které jsou řešeny pomocí cizích klíčů. Sloupec `identifikacni_cislo` je určen pro uložení čísla označujícího každou rostlinu. U stromů, keřů a polokeřů je možné evidovat i šířku koruny, tento sloupec je nepovinný.

Číselníky pro všechny třídy, řády a čeledě:

trida

- Sloupce: `id_trida` (serial), `nazev_cz` (varchar), `nazev_lat` (varchar), `nazev_sk` (varchar)
- Primární klíč: `id_trida`
- Popis: Přehled tříd s možností uložení ve 3 jazycích.



Obrázek 2: Část databáze pro rostlinu.

celed

- Sloupce: id_celed (serial), nazev_cz (varchar), nazev_lat (varchar), nazev_sk (varchar)
- Primární klíč: id_celed
- Popis: Přehled čeledí s možností uložení ve 3 jazycích.

rad

- Sloupce: id_rad (serial), nazev_cz (varchar), nazev_lat (varchar), nazev_sk (varchar)
- Primární klíč: id_rad
- Popis: Přehled řádů s možností uložení ve 3 jazycích.

Tabulka **namereny_bod** je určena pro uložení naměřených rostlin. Lze ji také využít jako vrstvu (shapefile) pro měření v terénu.

namereny_bod

- Sloupce: gid_namereny_bod (serial), nazev (varchar), geom (point)
- Primární klíč: gid_namereny_bod
- Popis: Sloupec **nazev** lze považovat za pracovní označení rostliny při měření v terénu

Návrh části databáze pro rostliny byl představen, nyní je potřeba představit zbylou část. Hlavní pozornost je zde věnována tabulce **objekt**, která shrnuje všechny objekty.

objekt

- Sloupce: id_objekt (serial), objekt (integer)
- Primární klíč: id_objekt
- Popis: Hlavní účel tabulky slouží ke spojení fotek s objektem (budova, rostlina, ...). Sloupec **objekt** je cizí klíč odkazující na tabulky: **znacka**, **cesta**, **budova**, **vybaveni**.

Uložení je cest je provedeno rovnou ve 3 tabulkách **cesta**, **cesta_typ_cesty** a **typ_cesty**. Cesty mohou mít přiřazené typy, které ji definují (například: pěší, schody, průjezdná atd.) a umožňují ji ve výsledné mapě rozlišovat.

cesta

- Sloupce: gid_cesta (serial), nazev (varchar), geom (multilinestring), popis (text)
- Primární klíč: id_cesta
- Popis: Uložené cesty arboreta s názvem a popisem.

cesta_typ_cesty

- Sloupce: id_cesta_typ_cesty (serial), id_cesta (integer), id_typ_cesty (integer), nazev (varchar)
- Primární klíč: id_cesta_typ_cesty
- Popis: Tabulka reprezentuje vztah M:N pro **cesta** a **typ_cesty**. Sloupec **nazev** není povinný a je určen k jednoznačnému rozpoznání části cesty.

typ_cesty

- Sloupce: id_typ_cesty (serial), nazev (varchar), oznaceni (varchar), popis (text)
- Primární klíč: id_typ_cesty
- Popis: Tabulka pro typy cest. Sloupec **oznaceni** je určen pro název nebo zkratku symbolu, kterým bude typ cesty označen v mapě.

Další důležitou součástí arboreta MENDELU jsou bezesporu informační tabule.

znacka

- Sloupce: gid_znacka (serial), nazev (varchar), hlavni_cislo (integer), vedlejsi_cislo(integer), popis (varchar), geom (point)
- Primární klíč: gid_znacka
- Popis: Uložené informační tabule v areálu. Obsahují hlavní číslo oblasti a vedlejší číslo, které se nachází přímo na tabuly. Sloupec **poznámka** je určen pro případný popis oblasti.

Jeden z požadavků úvaděl potřebu evidence budov a ostatního vybavení arboreta. Řešením jsou tabulky **budova**, **vybaveni** a **typ_vybaveni**. Vybavením jsou myšleny všechny další důležité věci jako lavičky, vodovodní kohoutky, atd.

budova

- Sloupce: gid_budova (serial), nazev (varchar), popis (text), geom (multipolygon)
- Primární klíč: gid_budova
- Popis: Tabulka pro budovy arboreta, které se ukládají jako polygony.

vybaveni

- Sloupce: gid_vybaveni (serial), id_typ_vybaveni (integer), popis (text), geom (point)
- Primární klíč: gid_vybaveni
- Popis: Tabulka pro uložení ostatního vybavení arboreta.

typ_vybaveni

- Sloupce: id_typ_vybaveni (serial), nazev (varchar)
- Primární klíč: id_typ_vybaveni
- Popis: Přehled možných typů vybavení.

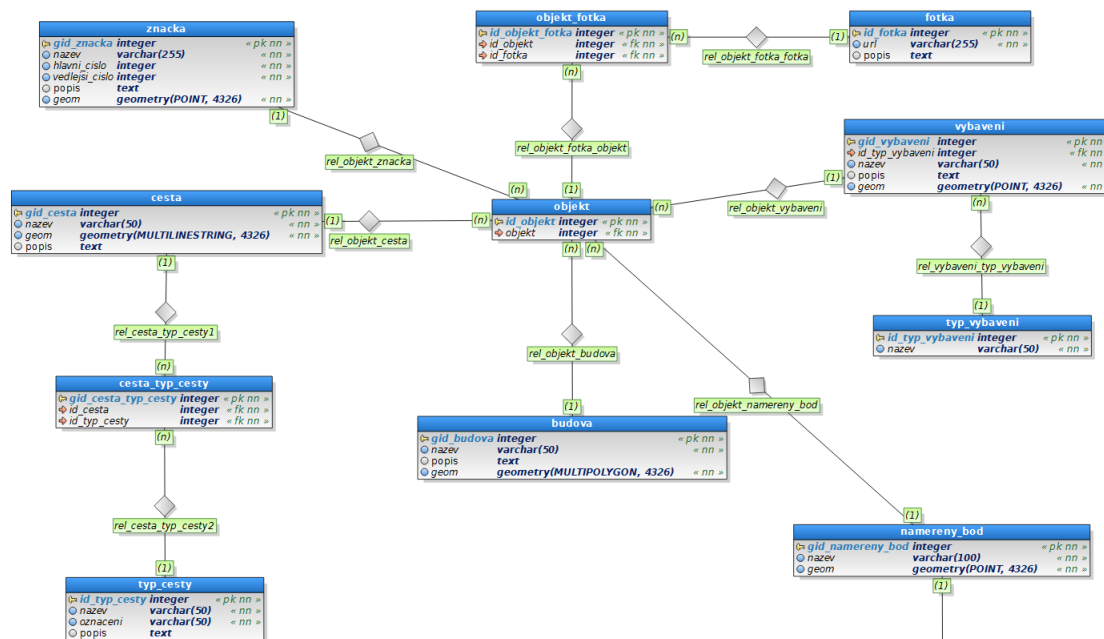
V dnešní době je téměř nemyslitelné zobrazovat data mapy bez reálné fotky určitého místa. Typickým příkladem využití fotek s mapami mohou být Google mapy, které umožňují sledovat fotky přiřazeném k místům nebo virtuální prohlídku místa, díky aplikaci Street View. Zde jsou fotky přiřazovány k objektům arboreta vztahem M:N, takže jedna fotka může mít jeden nebo více objektů a objekt může mít jednu nebo více fotek.

fotka

- Sloupce: `id_fotka` (serial), `url` (varchar), `popis` (text)
- Primární klíč: `id_fotka`
- Popis: Sloupec `url` slouží k uložení url ⁹ adresy na serveru, na kterém jsou nahrané fotky.

objekt_fotka

- Sloupce: id_objekt_fotka (serial), id_objekt (integer), id_fotka (integer)
- Primární klíč: id_objekt_fotka
- Popis: Pomocná tabulka pro vazbu M:N mezi tabulkami objekt a fotka.



Obrázek 3: Návrh databáze.

⁹URL (Uniform Resource Locators) - Umístění webové stránky nebo souboru na internetu.

6 Závěr

7 Reference

- ARCTUR, D. – ZEILER, M *Designing geodatabases: case studies in GISdata modeling*. Redlands: ESRI Press, 2004. 393 s. ISBN 1-58948-021-X.
- BERNHARDSEN, TOR *Geographic information systems: an introduction*. 3rd ed. New York: John Wiley, c2002, xiii, 428 s. ISBN 04-714-1968-0.
- BOUNDLESS *Boundless : Spatial Database Tips and Tricks : Introduction*. Boundless, formerly OpenGeo [online]. 2013 [cit. 2014-05-02]. Dostupné z: <http://workshops.boundlessgeo.com/postgis-spatialdbtips/introduction.html>.

Přílohy