

Mendelova univerzita v Brně
Provozně ekonomická fakulta

Geodatabáze pro GIS arboreta MENDELU

Bakalářská práce

Vedoucí práce:
Ing. Jiří Fejfar, Ph.D.

David Ruber

Brno 2014

Zde prostor pro zadání práce

Rád bych poděkoval svému vedoucímu práce Jiřímu Fejfarovi za ochotu, odborné konzultace v rámci této práce a také za zapůjčení GPS přijímače.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Geodatabáze pro GIS 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 *Směrnici 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.

Abstract

Ruber, D. The proposal of geodatabase for GIS of arboretum MENDELU. Brno 2014. Bachelor thesis. Mendelova univerzita v Brně. Thesis supervisor Ing. Jiří Fejfar, Ph.D.

This bachelor thesis is about measuring of geographic data in the arboretum of MENDELU, design and implementation of geodatabase. For the implementation of the geodatabase is use PostgreSQL with PostGIS extension. The measurement data is measure by Locus Map application for the operating system Android. The practical part describe the database design and creation, the methodology measurement of the arboretum, inserting data into database and final web presentation of geographical data. The aim of this work is evaluate created design and the possibility of extension into complete GIS.

Keywords

Geographic information system (GIS), Geodatabase, PosGIS, Locus Map, GeoServer, WMS, OpenLayers

Abstrakt

Ruber, D. Návrh geodatabáze pro GIS arboreta MENDELU. Brno, 2014. Bakalářská práce. Mendelova univerzita v Brně. Vedoucí práce Ing. Jiří Fejfar, Ph.D.

Tato práce se zabývá měřením geodat v terénu arboreta MENDELU, návrhem a tvorbou geodatabáze. K implementaci je využita datábáze PostgreSQL s rozšířením PostGIS. Pro měření dat je využita aplikace Locus Map, pro operační systém Android. V praktické části je popsán návrh databáze a její tvorba, metodika měření v arboretu, vložení dat do databáze a výsledná webová prezentace geografických dat. Cílem práce je také zhodnocení návrhu a možnost rozšíření do podoby kompletního GIS.

Klíčová slova

Geografické informační systémy (GIS), Geodatabáze, PosGIS, Locus Map, GeoServer, WMS, OpenLayers

Obsah

1 Úvod a cíl práce	8
1.1 Úvod	8
1.2 Cíl práce	8
2 Geografické informační systémy(GIS)	9
2.1 Data se kterými pracuje	9
2.1.1 Rastrová data	9
2.1.2 Vektorová data	9
2.2 Souřadnicové systémy	10
2.2.1 Systémy používané v ČR	10
3 Dostupný software	11
3.1 Prostorové databáze	11
3.1.1 MySQL Spatial	11
3.1.2 SQL Server	11
3.1.3 Oracle Spatial	12
3.1.4 PostGis	12
3.2 GeoServer	12
3.3 OpenLayers	13
4 Návrh geodatabáze	14
4.1 Popis tabulek	14
5 Postup při tvorbě databáze	19
6 Měření dat	21
6.1 Arboretum	21
6.2 Měřící přístroje	21
6.2.1 Mio DigiWalker A701	21
6.2.2 Aplikace pro operační systém android	22
6.2.3 Locus Map	22
6.3 Přesnost měřící přístrojů	23
6.3.1 Vliv na přesnost	23
6.4 Metodika měření v arboretu	25
6.4.1 Měření pomocí Mia	25
6.4.2 Měření bodu pomocí Locus Map	26
6.4.3 Záznam trasy pomocí Locus Map	27
6.5 Vložení dat do databáze	28
6.5.1 QGIS	29
6.5.2 Utility	31
7 Publikace dat z databáze na web	33

8 Diskuze	35
8.1 Srovnání databáze s BotanGIS	35
8.2 Možnosti rozšíření	35
9 Závěr	37
10 Reference	38
Přílohy	40
A SQL pro vytvoření databáze	41
B Schéma databáze	51
C Hesla pro přiložené DVD	52

1 Úvod a cíl práce

1.1 Úvod

Geografické informační systémy jsou dnes již běžně využívané napříč mnoha odvětvími. Hlavní výhodou těchto systémů ve srovnání s papírovými mapami je, že lze provádět prostorové analýzy nad daty. Analýzy potřebují data, která jsou v dnešní době lehce získatelná. Data můžeme dělit na volně dostupná a vlastní data. Mnoho firem poskytuje svá data volně k dispozici, některé za poplatek. Dat k reprezentaci v okolním světě je velké množství, proto je uživatel/poskytovatel nucen měřit v terénu. Měření dat obstarávají přenosné počítače s GPS přijímačem a potřebným programovým vybavením. Technologický pokrok přinesl velké množství zařízení splňujících tyto podmínky. V práci bude srovnáno měření pomocí satelitního navigátoru Mio a smartphone s operačním systémem Android.

Provádění prostorových dotazů by nebylo možné bez geodatabáze a data je potřebné někam uložit. Existují prostorové databázové systémy, fungující většinou jako nádstavba relační databáze, nebo geodatabáze (ESRI). Nové možnosti databází přinesly nové datové typy (bod, linie, polygon, ...) a funkce. Velmi důležitým faktorem se stává návrh struktury databáze, pro efektivnost prostorových dotazů a lehké uložení dat. K ukládání dat z mapových souborů (shp, kml, pgx, ...) existují programové nástroje, které vytvoří z dat sql dotazy pro vkládání (INSERT). Nutné je však vstupní data dobře připravit.

1.2 Cíl práce

Hlavním cílem práce je navrhnout strukturu geodatabáze pro uložení dat arboreta MENDELU. Vytvořit databázi podle návrhu, naplnit ji cvičnými daty a tyto data prezentovat pomocí jednoduché webové aplikace. Databáze bude využita pro uchování souřadnic rostlin, cest, informačních značek a dalších součástí arboreta.

Dalším cílem této práce je vytvořit metodiku měření dat v arboretu, která se bude dále používat na cvičeních z předmětu Geografické informační systémy. Porovnat přesnost měřících zařízení a díky tomu určit nejvhodnější aplikaci pro uložení aktuální polohy uživatele, na operačních systémech Android.

2 Geografické informační systémy(GIS)

Geografický informační systém zahrnuje hardware, software a grafická data, určený pro zachycení, správu, analýzu a zobrazení geografických informací. Umožňuje prohlížet, interpretovat a vizualizovat data, porozumět jím a dotazovat se na data v mnoha podobách, jako například mapy, zprávy nebo grafy. Příkladem pojmu GIS je aplikace pro práci s geografickými daty. Oblast využití je velice rozsáhlá. Od státní správy, jako jsou aplikace pro prohlížení katastrálního území, přes vojenství, až po školství (výuka zeměpisu). Data využívána se dají rozdělit do dvou skupin. Data rastrová a vektorová.(Esri, 2014)

2.1 Data se kterými pracuje

Využívaná data se dají rozdělit do dvou skupin. Data rastrová a vektorová.

2.1.1 Rastrová data

Nejjednodušší forma dat. Rastrový model se skládá z mřížky buněk (pixelů) uspořádaných v řádcích a sloupcích. Nejčastěji se jedná o čtvercovou mřížku, nebo o nepravidelného trojúhelníkového tvaru pro reprezentaci povrchů. Hodnota každé buňky reprezentuje určitou informaci. Rastrová data jsou digitální letecké snímky, snímky z družic, digitální obrázky, nebo naskenované mapy. (Rigaux, 2002)

Rastrová data představují reálné jevy, například:

- Tématická (diskrétní) data - reprezentují funkce, jako jsou údaje o využití půdy
- Průběžná (kontinuální) data - reprezentují jevy, jako je teplota, nadmořská výška nebo spektrální data, například satelitní a letecké snímky
- Obrázky - naskenované mapy nebo výkresy a stavební fotografie

Rastrová data jsou jednoduchá a také velmi vhodná pro velkou škálu aplikací. Použití rastrových dat spadá do hlavních čtyř kategorií:

- Rastry jako základní mapy
- Rastry jako povrchové mapy
- Rastry jako tématické mapy
- Rastry jako atributy prvku

2.1.2 Vektorová data

Vektorová data jsou rozdělena do tří typů: body, linie, polygony (mnohoúhelníky). Bodová data se nejčastěji používají k reprezentaci nesousedících prvků a zastupují samostatné datové body. Body mají nulové rozměry, proto nelze měřit délku

ani plochu dat. Příkladem mohou být body reprezentující města, památky, restaurace, vrcholek hory a mnoho dalšího. ()

Liniová data jsou využívána k reprezentaci lineárních prvků. Běžným příkladem je tok řeky, silnice nebo ulice. Linie mají jeden rozměr a proto mohou být použity jen pro měření délky. Prvky mají počáteční a koncový bod. K rozlišení linií, které reprezentují odlišné prvky mapy, se používají různé typy čar (plné čáry, čárkované, čerchované, ...), tloušťky a barvy čar.

Polygony se používají k reprezentování oblastí, jako jsou hranice (velikosti) měst, jezera, louky, lesy. Polygon je dvourozměrný, a proto se může určit plocha a obvod. Nejčastěji se rozlišují pomocí tématického mapování (barevná schémata), vzorů nebo v případě číselného stupňování pomocí stupňování barev.

2.2 Souřadnicové systémy

Geografický souřadnicový systém je trojrozměrný vztažný systém, který lokalizuje bodu na Zemském povrchu. Body mají dvě hodnoty souřadnic: zeměpisná šířka a délka, které se meří v úhlech. Zeměpisná šířka je definována jako úhel tvořený průsečíkem přímky kolmé k zemskému povrchu v bodě, pro který souřadnici zjišťujeme, a rovinou rovníku. Hodnoty úhlu se pohybují v rozmezí od -90 do 90 stupňů. Body na sever od rovníku mají kladné hodnoty zeměpisné šířky, a body jižně od rovníku mají záporné hodnoty. Zeměpisná délka je definována pomocí úhlu mezi poledníkem (přímka procházející bodem) a nultým poledníkem. Poledník je tvořen rovinou, která prochází bodem a severním a jižním pólem. Nejběžněji používaný nultý poledník prochází částí Greenwich v Londýně. Hodnoty zeměpisné délky jsou v rozmezí od -180 do 180 stupňů. (Esri, 2014)

2.2.1 Systémy používané v ČR

Hlavní souřadnicové systémy používaná na území České republiky:

- WGS 84 (World Geodetic System 1984)
Světový geodetický referenční systém z roku 1984, založený na referenčním elipsoidu. Tento systém se používá jako standardní pro Satelitní navigaci GPS. Využívá se například i v armádě USA nebo jako geodetický systém NATO. Od roku 1998 je používán v letectvu a Armádě ČR v rámci kooperace s armádami NATO. Tento souřadnicový systém byl použit pro data v této práci.
- S-JTSK
Souřadnicový referenční systém S-JTSK je definován Křovákovým kartografickým zobrazením. Křovákovo zobrazení je konformní kuželové zobrazení v obecné poloze, které v roce 1922 navrhl Ing. Josef Křovák. Zobrazení zachovává úhly, ale nezachovává délky.

3 Dostupný software

3.1 Prostorové databáze

Prostorová databáze (spatial database) je nádstavba relační databáze, určená pro manipulaci s prostorovými daty. (Arctur, 2004) Většina prostorových databází umožňuje pracovat s jednoduchými geometrickými objekty, jako jsou body, linie a polygony. Některé jsou schopny pracovat i s 3D objekty, topologickými modely nebo s trojúhelníkovou sítí (TIN¹). (Rigaux, 2002) Tyto databáze neobsahují jen nové datové typy, ale také nové funkce.

Funkce jsou pro efektivní zpracování prostorových dat, například funkce pro výpočet obsahu polygonu, délky linie a mnoho dalších. Další odlišností od relační databáze je i prostorový index (spatial index). Na rozdíl od obyčejného indexu, který je určen pro indexování standardních datových typů (jako jsou číselné typy, textové typy, datumy) je určen pro řazení geometrických dat, z důvodu efektivního provedení prostorových dotazů. Nejčastější metodou prostorového indexu je R-tree².

3.1.1 MySQL Spatial

Rozšíření Spatial pro MySQL se řídí dle specifikací Open Geospatial Consortium (OGC). Z toho vyplývá, že obsahuje prostorové typy, funkce a index. Oproti jiným databázovým možnostem poskytuje málo funkcí, takže je vhodný pro menší projekty. (Westra, 2010)

3.1.2 SQL Server

Microsoft uvedl svoji prostorovou databázi poprvé ve svém SQL Serveru 2008. Od tohoto vydání každé další plně podporuje prostorovou funkčnost. SQL Server je ve své podstatě objektově orientovaný, což z něj dělá hlavní odlišnost oproti ostatním databázím. Další odlišností je používání multi-level grid scheme, který umožňuje používání standartního stromu B-tree pro prostorový index. Hlavní výhoda této databáze je v možnosti grafického zobrazení SQL dotazu pomocí SQL Server Management Studio. Ovšem SQL Server postrádá důležitou funkci transformace souřadnicového systému a další nevýhodu lze uvést i omezení, že jej lze spustit jen na serverech se systémem Windows. (Boundless, 2013)

¹TIN (Triangulated Irregular Network) - Nepravidelná trojúhelníková síť je datová struktura používaná pro reprezentaci povrchu.

²R-tree je vyvážený vyhledávací strom

3.1.3 Oracle Spatial

Firma Oracle poskytuje dvě možnosti pro práci s prostorovými daty. Oracle Locator je zahrnut ve všech databázích firmy Oracle, poskytuje funkce a mapování potřebné pro umožnění práce s polohou. Oracle Spatial je doplněk, který zahrnuje prostorové funkce, podporu 3D, správu územních, vektorových a rastrových dat, topologii a síťové modely. Mezi výhody lze řadit to, že vyhovuje potřebám moderních geografických informačních systémů nebo disponuje zabezpečením na vysoké úrovni. (Oracle, 2013) Z hlediska množství funkcí je Oracle Spatial oproti ostatním geodatabázím velký kus napřed. Pro složitější use case (případy užití) v geoprostorové doméně poskytuje rozšiřující balíčky:

- podpora Linear referencing system (LRS)
- prostorové analýzy a mining funkce a procedury (balíček SDO_SAM)
- geokódování (balíček SDO_GCDR)
- GeoRaster³

3.1.4 PostGis

PostGis je rozšíření databáze PostgreSQL, které ji přemění na prostorovou pomocí prostorových datových typů, indexů a funkcí. Zahrnuje podporu pro prostorové operace, jako je tvorba a konverze dat, convex hull (konvexní obal⁴), převod mezi souřadnými systémy. Užitečná funkce je import a vykreslení různých vektorových (například KML, GML, GeoJSON, GeoHash) a rasterových (například GeoTIFF, PNG, JPG) datových formátů. PostGis stejně jako Oracle Spatial podporuje 3D objekty, umožňuje prostorové indexování a topologii sítě. (Hsu, 2014)

Mezi nejvýznamnější uživatele PostGisu patří IGN⁵, pro mapování země ve vysokém rozlišení. PostGis je široce používán uživateli, protože je podporován mnoha programy. Z nejznámějších to jsou QGIS, GRASS, Shp2Psql, ogr2ogr, MapServer, GeoServer, AutoCAD Map 3D. PostGis jsem si vybral pro tvorbu geodatabáze Arboreta právě z důvodu podpory ostatních programů pro tvorbu GIS a jeho velké oblíbenosti mezi uživateli. (Boundless, 2013)

3.2 GeoServer

GeoServer je open source software, vytvořený pomocí programovacího jazyku Java, který umožňuje sdílet a upravovat geografická data.(Hsu, 2014) Využívá otevřené standardy jako například standard Web Map Service (WMS) nebo Web Feature

³GeoRaster umožňuje ukládání, indexaci, dotazování, analyzování a doručování rastrových obrazů a gridů.

⁴Konvexní obal určité množiny bodů je nejmenší konvexní množina, které tuto množinu obsahuje.

⁵Institut Geographique National - národní mapovací agentura Francie.

Service (WFS). Podporuje mnoho formátů, mezi které patří ArcSDE, Oracle Spatial, Microsoft SQL Server, ESRI Shapefile, GeoTIFF a další. Mezi výstupní soubory patří ESRI Shapefile, KML, GML, GeoJSON, PNG, JPEG, TIFF a mnoho dalších. Hlavní plusem GeoServeru je bezpochyby plně vybavené webové rozhraní určené pro administrativu.

3.3 OpenLayers

OpenLayers je javascriptová knihovna, která usnadňuje práci s mapami ve webovém prostředí. Umožňuje zobrazovat geografická data z mnoha typů souborů a služeb, například SHP, KML nebo WMS. Na rozdíl od jiných softwarových nástrojů pro tvorbu bohatých webových aplikací je tato knihovna OpenSource. OpenLayers také implementuje například OpenStreetMap nebo Google Maps. (openlayers.org, 2014)

4 Návrh geodatabáze

Před samotnou tvorbou databáze je potřeba rozhodnout jaká data se budou zaznamenávat. Hlavní požadavek je, že se budou evidovat jednotlivé rostliny nacházející se v arboretu. Záznamy by měly obsahovat důkladný popis rostlin. Dostupné informace o rostlině, které se nacházejí na informační tabulce přímo u jednotlivých exemplářů, ukazují nutnost zaznamenání latinského názvu, třídy, čeledi, rádu, identifikačního čísla přiřazeného jednotlivým rostlinám, poznámky (například krátký popis). Dalším požadavkem je možnost evidence cest, budov, informačních tabulí a dalšího vybavení, které je součástí areálu. Při návrhu databáze byl využit grafický modelovací nástroj pgModeler - PostgreSQL Database Modeler. Databáze se skladá z 16 tabulek.

4.1 Popis tabulek

Z prvního požadavku je vytvořena tabulka **rostlina** :

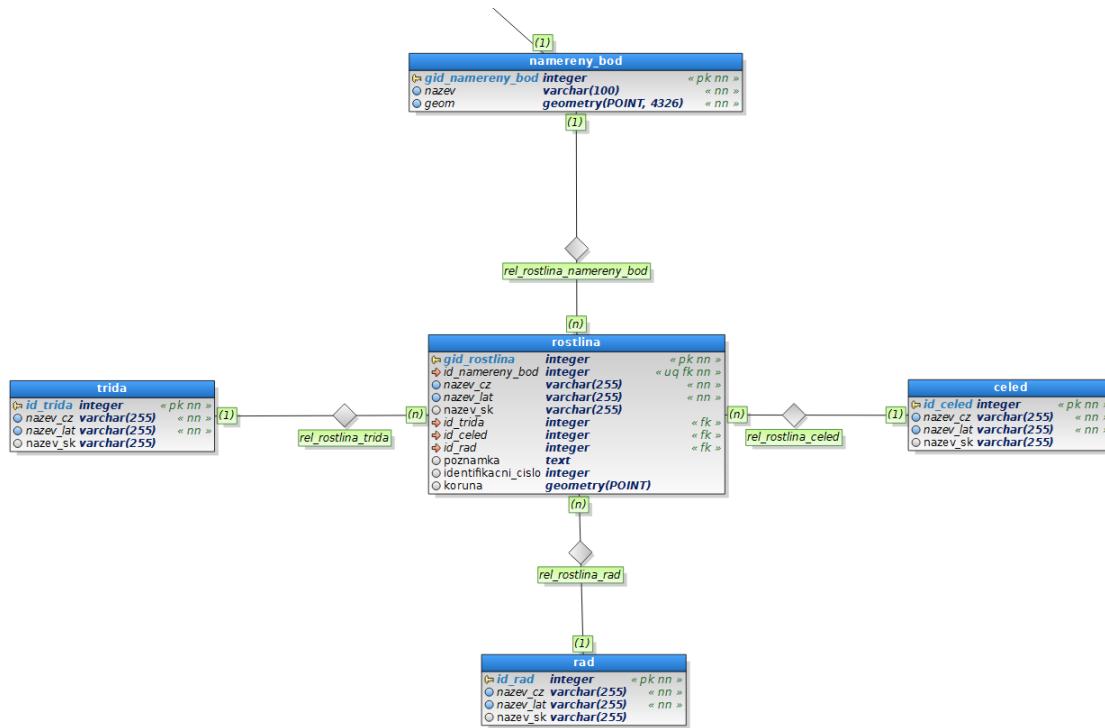
rostlina

- 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 rád, 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, rády 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 1: Část databáze pro rostlinu.

cested

- 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 vyžít 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 **oznacení** 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), vedejsi_cislo(integer), popis (varchar), geom (point)
- Primární klíč: gid_znacka
- Popis: Uložené informační tabule v areálu. Obsahují hlavní číslo oblasti a vedejší číslo, které se nachází přímo na tabuly. Sloupec **poznamka** 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_typy_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.

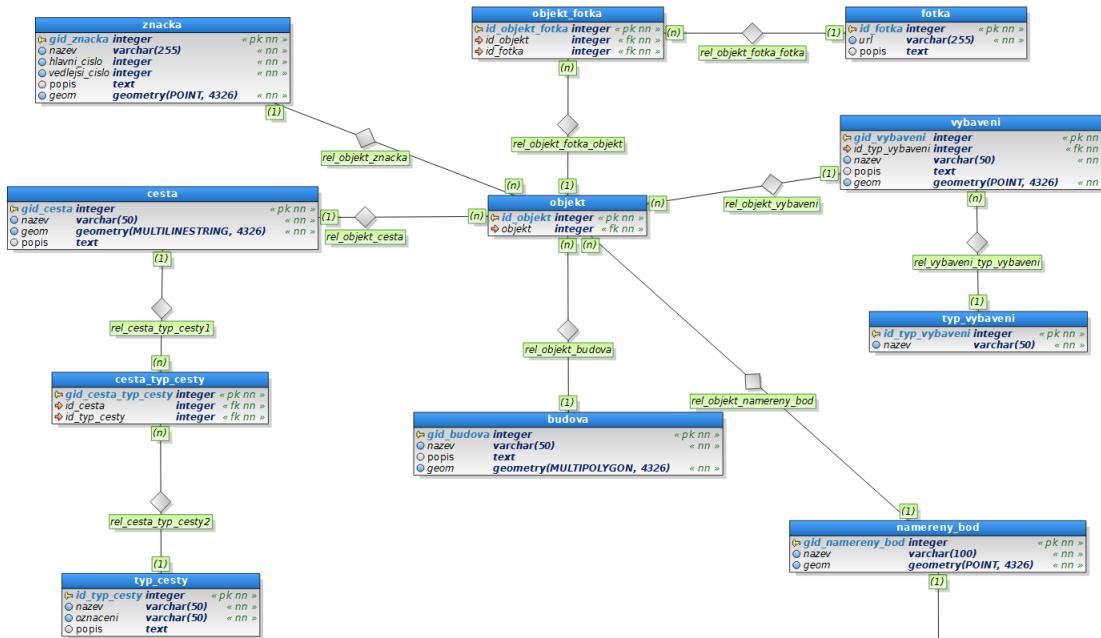
V dnešní době je téměř nemyslitelné zobrazovat data mapy bez reálné fotky určitého místa. Typickým příkladem využití fotek s mapami mohou být Google mapy, které umožňují sledovat fotky přiřazené k místům nebo virtuální prohlídce místa, díky aplikaci Street View. Zde jsou fotky přiřazovány k objektům 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 url⁶ 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 2: Návrh databáze.

⁶URL (Uniform Resource Locators) - Umístění webové stránky nebo souboru na internetu.

5 Postup při tvorbě databáze

Po návrhu databáze zbývá už jen pár kroků k celému dokončení. Prvním je instalace databáze PostgreSQL s rozšířením PostGIS. Instalace, z které byl návod vytvořen, byla provedena na operačním systému Debian, verze Wheezy 7.5.

Prvním krokem je třeba vytvořit soubor pgdg.list:

```
/etc/apt/sources.list.d/pgdg.list
```

Do něj přidat řádek:

```
deb http://apt.postgresql.org/pub/repos/apt/ wheezy-pgdg main
```

Po té se dostaneme k instalaci PostgreSQL:

```
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc  
| sudo apt-key add -  
sudo apt-get update  
sudo apt-get upgrade  
sudo apt-get install postgresql-9.3 pgadmin3
```

Nyní instalace PostGISu:

```
sudo apt-get install Postgresql-9.3-postgis pgadmin3 postgresql-contrib
```

Vytvoříme si nového uživatele a databázi pomocí terminálu psql, který umožňuje psát SQL dotazy:

```
su - postgres  
psql  
CREATE USER uzivatel_arboretum WITH PASSWORD 'arboretum';  
CREATE DATABASE gis OWNER uzivatel_arboretum;  
\q
```

Přidáme rozšíření pro PostGIS:

```
psql -d gis -c "CREATE EXTENSION postgis;"  
psql -d gis -c "CREATE EXTENSION postgis_topology;"
```

Samotná tvorba databáze je jednoduší díky tvorbě návrhu pomocí pgModeler, který umožňuje export SQL příkazů pro tvorbu vytvořeného modelu do souboru s příponou **.sql**. V méém případě se jedná o soubor **arboretum.sql**, který spustím nad databází:

```
psql -d gis -f arboretum.sql
```

Po výše uvedených příkazech je databáze vytvořená a připravená pro vložení dat. Abychom mohli data vložit, musíme je nejdříve získat v terénu.

6 Měření dat

Tato kapitola představí místo měření dat (arboreum), měřící přístroje, možnosti aplikací pro chytré telefony a metodiku měření.

6.1 Arboretum

Arboretum, přesněji botanická zahrada a arboretum, je výukové zařízení Mendelovy Univerzity v Brně. Celkově zde roste přibližně na 13 000 rostlin a na 5000 dřevin. Informační panely uvádějí, že arboretum o rozloze 10,96 ha má průměrnou nadmořskou výšku 230 m.n.m., průměrná roční teplota vzduchu je 8,4 °C a průměrný úhrn srážek tvoří 531 mm.

Základní historická data:

- 1926 - založena botanická zahrada Vysoké školy zemědělské v Brně v areálu
- 1938 - založena Prof. Bayerem sbírka dřevin - arboretum - na rozloze cca 1 ha, vzdálená asi 500 m od objektu školy
- 60. léta - původní botanická zahrada kvůli nové výstavbě zrušena
- 1968 - založena nová botanická zahrada a arboretum podle návrhu Prof. Ivana Otruby v návaznosti na stávající sbírku dřevin

Areál je členěn do 5 částí, a to:

- Okolí správní budovy
- Jižní svahy
- Centrální část
- Staré arboretum
- Botanický systém

6.2 Měřící přístroje

V této práci byl použit jako hlavní měřící přístroj Mio DigiWalker A701. Pojdeme si je tedy představit.

6.2.1 Mio DigiWalker A701

Mio A701 je kombinací mobilního telefonu, kapesního počítače a navigace. Skrývá v sobě kvalitní modul SiRFstar III pro GPS. Na operačním systému Microsoft Windows Mobile 5.0 běží aplikace ArcPad 10, která byla použita pro měření dat. Přístroj disponuje také 1,3 megapixelovým foťákem. Hlavní součást je samozřejmě procesor Intel XScale PXA270 520 MHz. GPS přijímač s rozměry 107 × 57 × 18,8 mm, je

napájen 1350 mAh baterií, s kterou vydrží až 20 hodin ve stand-by režimu. Tyto parametry z něj dělají kvalitní, lehce přenosný GPS přijímač.

6.2.2 Aplikace pro operační systém android

Hlavním parametrem při výběru aplikace pro operační systém android byla cena. Hledal jsem aplikace, které je možné nainstalovat zadarmo. Další parametry byly kvalitní offline mapy a export naměřených dat v obvyklých formátech. Všechny předpoklady splňuje aplikace Locus Map. V obchodě Google Play lze stáhnout verzi zdarma nebo placenou. K funkčnosti, kterou potřebujeme k zaznamenání pozice a zaznamenání poznámek stačí verze zadarmo.

6.2.3 Locus Map

Locus map nabízí mnoho funkcí, které nabízejí využití při mnoha outdoorových aktivitách, jako cyklistika, turistika, lyžování, nebo pro zaznamenávání vlastních bodů a tras. Hlavní funkce aplikace:

- Offline/online mapy - vektorové mapy, možnost použití vlastních OpenStreetMap, podpora formátů SQLite, TAR, MBT, GEMF, RMAP, SmartMaps
- Vyhledávání - podle souřadnic, adresy, vlastních bodů
- Body zájmu - tvorba vlastních bodů a stop, import a export dat
- Geocaching - Geocaching Live
- Záznam trasy - možnost nahrávání tras s velkou možností nastavení
- Hlasová navigace - plná podpora pro online hlasovou navigaci
- Import/Export dat
- GPS - chytrý systém pro úsporu energie, satelitní obrazovka s aktuálním stavem, podpora Bluetooth GPS

Funkci, která je nejvíce využita v této práci je práce s Body zájmu. Uložené body lze ukládat do kategorií, které si uživatel tvoří sám. Na mapě se tak zobrazují jen ty skupiny kategorií, které jsou potřeba. Po uložení bodu je možnost bod dále upravovat. Všechny editace jsou jednoduché a snadné na zvládnutí. Další funkci je záznam trasy. Po zapnutí této funkce, se na spodní liště zobrazí ikony potřebné pro provedení kompletního záznamu. V nastavení uživatel definuje za jakých okolností záznam pořídit a po jaké dlouhé době nebo vzdálenosti uložit bod.

Locus se snaží být co nejvíce variabilní s neomezenými možnostmi. To dokazují i podporované formáty.

- Import: GPX, KML, KMZ, SQLite, MBT, TAR, GEMF
- Export: GPX, KML, KMZ, CSV, DXF
- Online služby: GPSies, Breadcrumbs, Ride With GPS, RunKeeper, Strava, Trainingstagebuch.org, OpenStreetMap, EveryTrail

Import lze využít pro zlepšení organizace měření dat v arboretu. Například vrstva s rozdelením částí arboreta, která poslouží pro rozdelení pracovních skupin, z nichž každá má na starosti jednu plochu.

6.3 Přesnost měřící přístrojů

Přesnost přístroje je závislá na použitých metodách při jeho využití. Lze získat přesnost v řádech centimetrů, ale také v řádech metrů. GPS určuje polohu pomocí družic, které vysílají signály pro uživatele v podobě signálu. Každá družice vysílá zprávy o své poloze a přibližné poloze ostatních družic systému. K určení aktuální polohy přijímač počítá tzv. pseudovzdálenosti, což jsou vzdálenosti mezi přijímačem a viditelnými družicemi. Výpočet pseudovzdálenosti vychází ze znalosti rychlosti šíření družicového signálu a rozdílu času mezi vysláním a příjemem signálu. K určení dvojrozměrné polohy postačí příjem signálu z minimálně čtyř družic. Pokud přijímač přijímá signál z více družic, pak bude měření více přesné. (Šebesta, 2012)

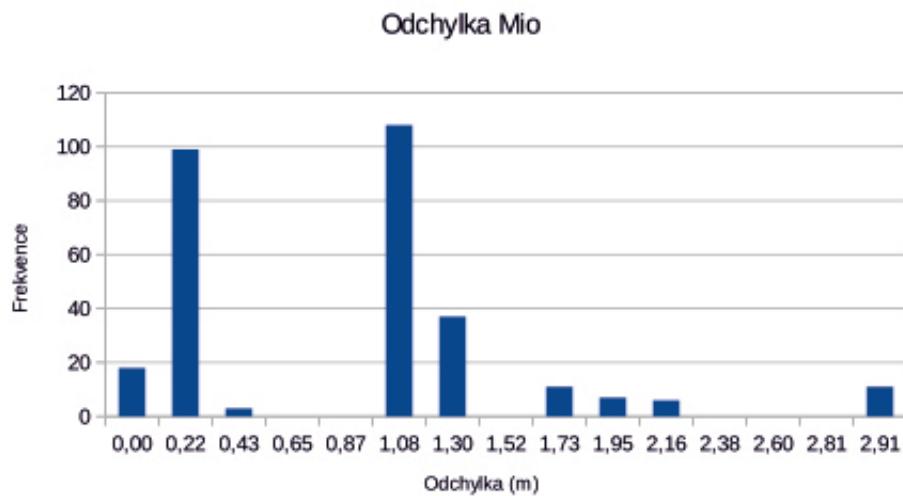
6.3.1 Vliv na přesnost

Vlivů na výslednou přesnost měření je mnoho. Zde uvádím jen pár základních, k jejich určení jsem vycházel z článku Přesnost satelitního navigačního systému GPS a jeho dostupnost v kritických podmínkách, publikovaného autory Marka a Štěpánka.

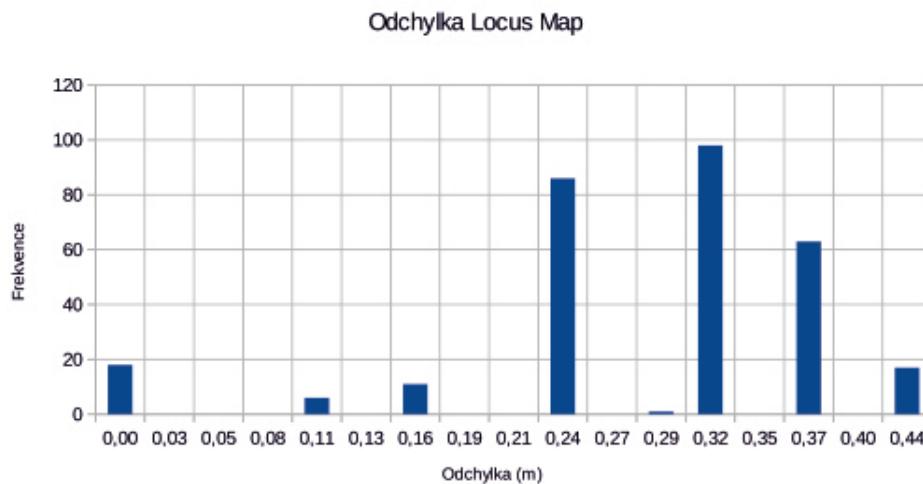
- **poloha satelitů** - tento vliv lze rozlišit na dva případy, prvním je ten, že přijímač má dobrý výhled na oblohu, v druhém případě má přijímač jen částečný výhled na oblohu
- **přesnost udávané polohy satelitů** - chyba polohy satelitu závisí na přesnosti výpočtu a na časové prodlevě sledování pohybu ostatních satelitů
- **atmosféra** - atmosféra má zásadní vliv při průchodu signálu, běžný uživatel s touto chybou nedokáže nic udělat
- **typ přijímače** - rozlišují se přijímače pro kódová nebo fázová měření, do první skupiny spadají klasické ruční navigační přístroje, druhá skupina se využívá zejména pro geodetické práce

- **šum signálu** - šum signálu může způsobovat vzrostlá vegetace, blízkost výškových budov nebo nízká poloha družice nad horizontem
- **vícecestné šíření signálu** - chyba vlivem vícecestného šíření signálu je jednou z nejobtížněji odstranitelných chyb při určování polohy a obecně nelze odstranit ani diferenciální korekcí (DGPS). Signály se mohou odrážet od pevných ale i pohyblivých objektů, zejména staveb a terénních nerovností, vozidel, letadel. Důsledkem toho je delší dráha signálu a tím i chyba v měření pseudovzdálenosti.

Pro porovnání jsem zvolil metodu určení odchylky podle referenčního bodu. Přístroje jsem umístil na otevřenou plochu s výhledem na oblohu. Bylo nutné zaměřit referenční bod a na každém přístroji alespoň 300 pokusných bodů. Aplikace Locus Map nabízí funkci Záznam trasy, pro kterou jsem nastavil interval záznamu po 1 sekundě. Obdobnou možnost poskytuje také aplikace ArcPad, běžící na přístroji Mio, ve formě uložení záznamu trasy jako soubor typu shapefile. Po naměření bodu bylo nutné tyto body porovnat s referenčním bodem. Využil jsem funkce v programu ArcMap Point Distance, která spočítá vzdálenost mezi všemi body jedné vrstvy a body druhé vrstvy. Referenční bod byl uložen jako jediný bod vrstvy. Reprezentace odchylky byla provedena pomocí grafu, ukazující četnost bodů v jednotlivých intervalech.



Obrázek 3: Graf odchylky pro přístroj Mio.



Obrázek 4: Graf odchylky pro přístroj s aplikací Locus Map.

Z grafů je patrné, že přístroj Mio měl nejčastěji odchylku 0,22 m a 1,08 m, ovšem objevuje se často i pro hodnotu 2,91 m. Odchylka u aplikace Locus Map se nejčastěji pohybuje v rozmezí 0,24 m až 0,40 m, ale občas se objeví i odchylka přibližně 0,44 m. Tyto měření ukázaly přesnost přístrojů a větší přesností disponuje aplikace Locus Map. Mobilní telefon HTC, na kterém aplikace běžela, má novější GPS modul, proto větší přesnost.

6.4 Metodika měření v arboretu

6.4.1 Měření pomocí Mia

Představíme si měření dat pomocí GPS navigace Mio DigiWalker A701 krok po kroku:

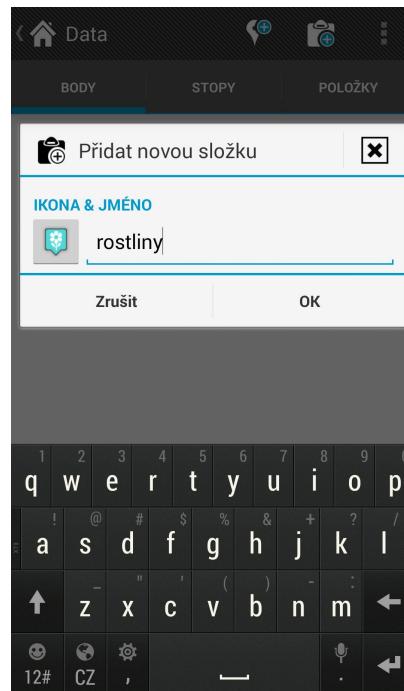
- spustíme program ArcPad 10
- vytvoříme rychlý projekt: šablonu necháme jako výchozí a souřadnicový systém zvolíme S-JTSK_Krovak_East_North
- na panelu zvolíme první záložku (Main - obrázek složky), vybereme pátou možnost, tedy GPS, a aktivujeme jej, mezitím než zařízení najde satelity si připravíme vrstvu pro data
- otevřeme menu vybereme položku Nový a Shapefile
- vybereme typ Point, a po kliknutí na tlačítko se znakem plus přidáme nové pole
- protože data budeme nahrávat do databázové tabulky **namereny_bod**, je potřeba přidat pole určené pro sloupec **nazev**

- pojmenujeme si ji název, typ zvolíme znak a délku přepíšeme na 100
- dokončíme vytvoření vrstvy, pojmenujme si jí například rostliny
- na panelu nástrojů zvolíme třetí možnost (Draw - obrázek kreslení), zkонтrolujeme zda je v třetí výběrové možnosti vybrán bod, a zvolíme čtvrtou možnost uložení bodu podle aktuálních GPS souřadnic
- otevře se atributová tabulka, zde máme jen pole název, do kterého napišeme pracovní název a uložíme
- toto opakujeme dokud nebudeme mít všechny požadované body naměřené

6.4.2 Měření bodu pomocí Locus Map

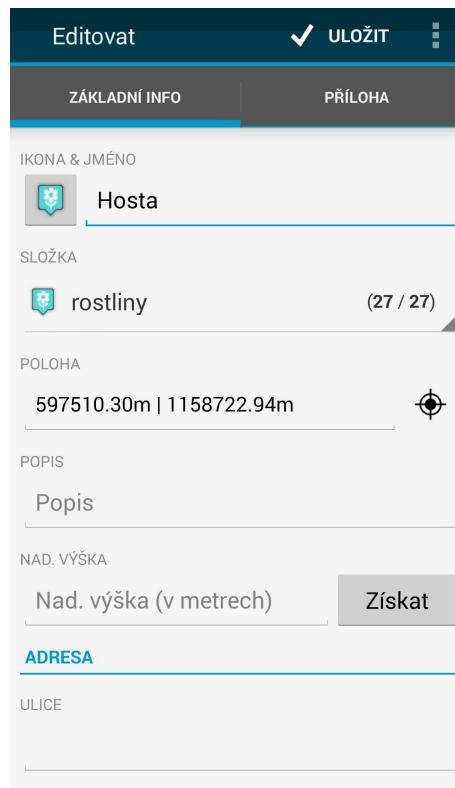
Měření bylo provedeno na verzi programu 2.20.2, verze systému Android 4.4.2. Podkladová mapa byla použita online mapa SHOCart, Cyklo.

- nejdříve si připravíme složku pro naměřená data
- po stisku čtvrtého tlačítka na horním panelu se přepneme do správce dat
- zvolíme záložku BODY a stiskneme tlačítko pro přidání nové složky



Obrázek 5: Přidání složky.

- pojmenujeme si ji například rostliny a ikonu zvolíme obrázek květiny
- prvním tlačítkem na spodním panelu si zobrazíme aktuální polohu
- znovu se přepneme do správce dat a stiskneme tlačítko pro přidání nového bodu
- v novém okně napíšeme název bodu zvolíme složku rostliny a uložíme pomocí tlačítka uložit

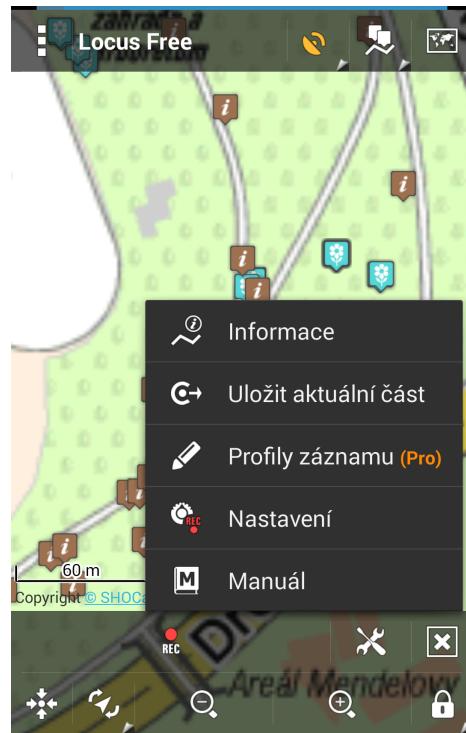


Obrázek 6: Přidání bodu.

- tímto způsobem uložíme další požadované body, před každým uložením doporučuji aktualizovat polohu

6.4.3 Záznam trasy pomocí Locus Map

- v hodném panelu zvolíme první položku, otevře se Hlavní menu, ve kterém zvolíme položku Více
- objeví se menu se všemi funkcemi, vybereme Záznam stopy
- nad spodním panelem se zobrazí nový panel pro ovládání záznamu
- otevřeme Nastavení, pomocí druhého tlačítka zprava (Možnosti)



Obrázek 7: Možnosti záznamu.

- v sekci Parametry nastavíme Inteval na 1 m a 1 s, Podmínky nahrávání zvolíme možnost Jedna
- toto nastavení bude přidávat body trasy po sekundě nebo po 1 metru, důležité je zaškrtnout možnost Záznam pouze při pohybu, abychom předešli zaznamenání více bodů na jednom místě
- nyní stlačíme tlačítko pro začátku mapování, první zleva (REC)
- měření můžeme pozastavit pomocí možnosti PAUSE a pokud naměříme celou trasu zvolíme STOP a uložíme ji
- uložit trasu lze, i když jen pozastavíme záznam, díky možnosti Uložit aktuální část nacházející se znova pod tlačítkem Možností

6.5 Vložení dat do databáze

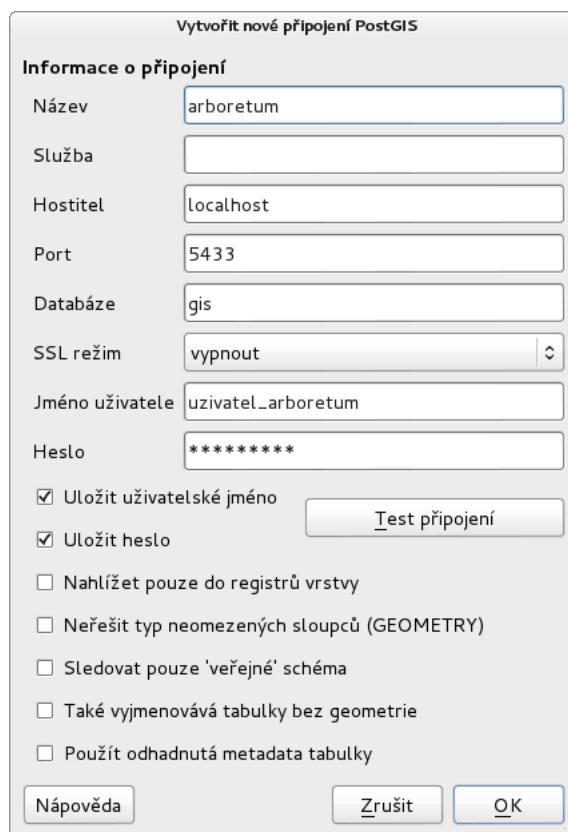
Naměřená data je nutno uložit do databáze. Mezi hlavní postupy nahrávání jsem vybral:

- program QGIS
- utility ogr2ogr a shp2pgsql

6.5.1 QGIS

QGIS (Quantum GIS) je multi platformní, uživatelská, open source GIS aplikace. Umožňuje prohlížení, úpravu a analýzu dat.

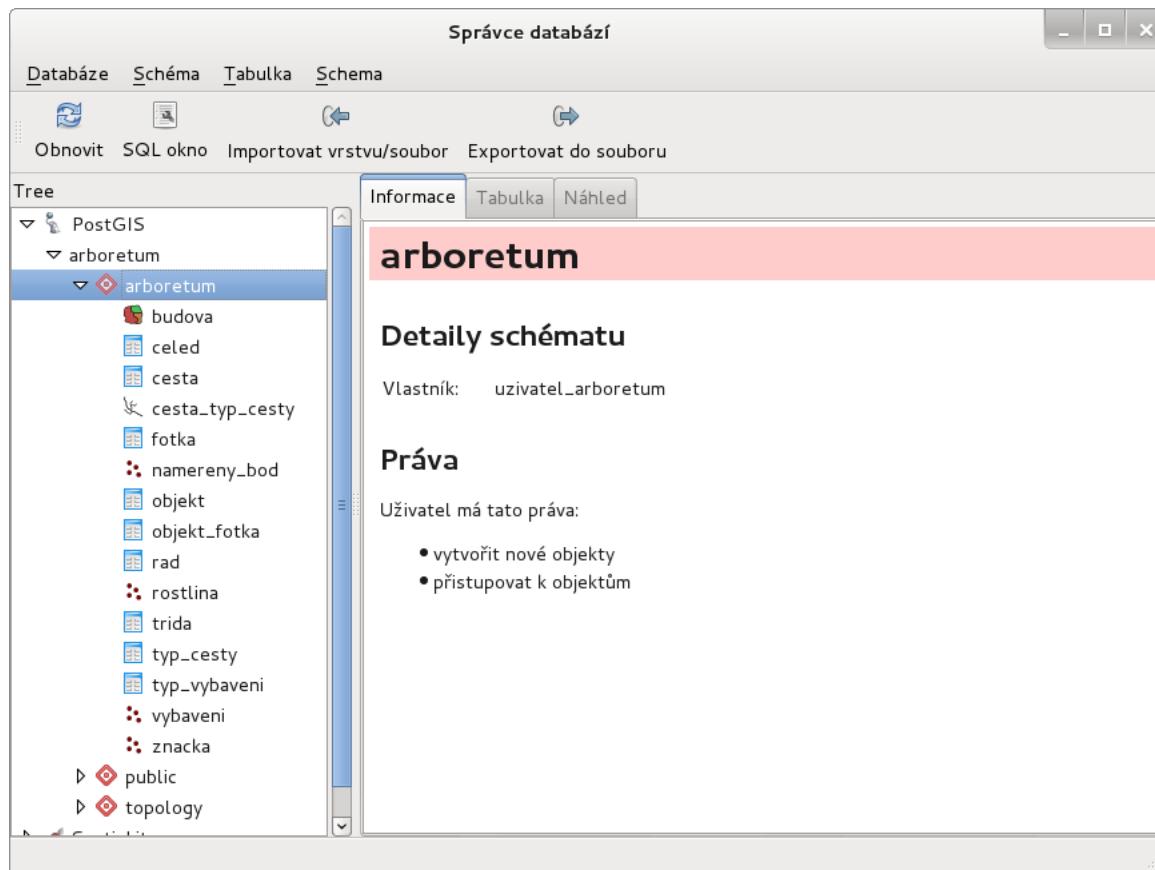
- naměřené vrstvy otevřeme pomocí záložky Vrstva, Přidat vektorovou vrstvu...
- připojíme databázi, v panelu Prohlížeč klikneme pravým tlačítkem na položku PostGIS a zvolíme Nové připojení...
- zobrazí se formulář pro údaje o databázovém serveru, který vyplníme



Obrázek 8: Připojení k PostGIS.

Pokud v databázové tabulce, do které chceme data vložit, žádná data nejsou musíme postupovat takto:

- otevřeme záložku Databáze, Správce databází, vybereme naše PostGIS připojení

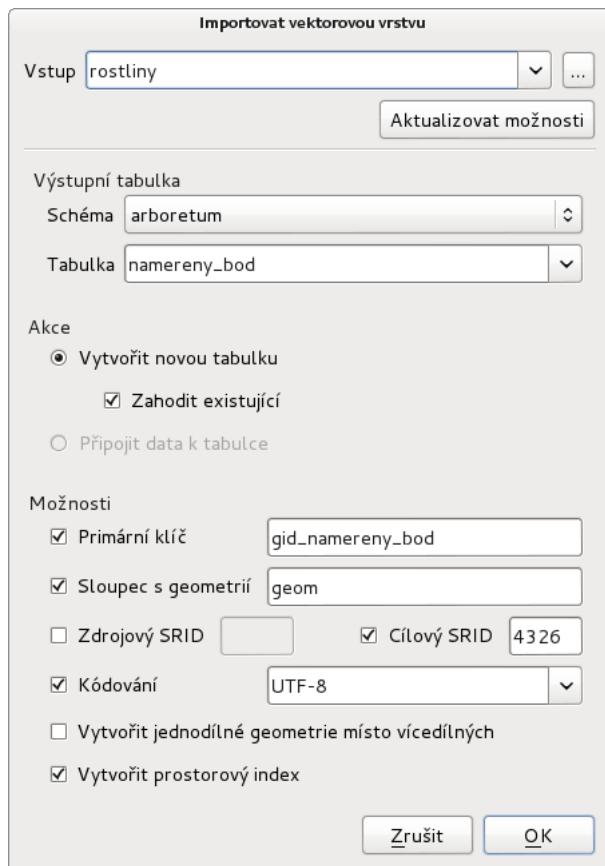


Obrázek 9: Import vektorové vrstvy.

- klikneme na Importovat vrstvu/soubor, zobrazí se formulář, ve kterém vybereme požadovanou vrstvu dat, kterou chceme importovat a tabulkou do které chceme import provést

Když už v tabulce nějaká data jsou a chceme nová data jen přidat k těm starým, doporučuji postup:

- přidáme vrstvu z databáze znovu přes záložku Vrstva, Přidat PostGIS Layers...
- připojíme se k databázi a vybereme požadovanou tabulkou (namereny_bod)
- prvky, které chceme přidat do databáze vybereme výběrovým nástrojem, v záložce Editovat zvolíme Kopírovat prvky
- zkopiované prvky je potřeba vložit do vrstvy s databázovou tabulkou, zapneme editaci pro vrstvu, v záložce Editovat zvolíme Vložit prvky, uložíme editaci a ukončíme ji



Obrázek 10: Správce databází.

6.5.2 Utility

Nejlepší možnost, která je ve bětí míře univerzální, importu dat do databáze je pomocí utilit `ogr2ogr`, `shp2pgsql`. `ogr2ogr` je open source nástroj pro příkazový řádek, určen pro konverzi vektorových formátů a dat. Příkaz použijeme pro export dat získaných aplikací Locus Map, která jsme exportovali do formátu .gpx:

```
ogr2ogr -f "ESRI Shapefile" rostliny.shp rostliny.gpx
-sql "SELECT name AS nazev FROM waypoints"
```

Parametr `-f "ESRI Shapefile"` určuje, že exportovaný soubor má být ve formátu ESRI Shapefile, jehož název je definován dalším parametrem `rostliny.shp`. `rostliny.gpx` určuje vstupní soubor. Důležitý je parametr `-sql` určující, pomocí SQL dotazu, jaká data chceme exportovat do nového souboru. Soubor GPX (GPS eXchange Format) je XML schéma navržen jako společný formát dat GPS aplikací. Je použit pro uložení uložení bodů se souřadnicemi a popisky (waypoints), tras vytvořených například GPS přijímačem (track) a tras složených z waypointů (route). Při měření byla zaznamenána data ve formě bodů s názvem, proto je parametr `-sql`

```
"SELECT name AS nazev FROM waypoints".
```

Příkaz `shp2pgsql` slouží pro převední souboru ESRI Shapefile do souboru SQL, který obsahuje příkazy nutné k vložení dat do databáze PostGIS. Data získaná pomocí GPS přijímače Mio jsou uložena v souboru s příponou `.shp` a data získaná aplikací Locus Map jsou již převedena na tento formát, takže můžeme použít příkaz:

```
shp2pgsql -s 4326 -I -a -W UTF-8 rostliny.shp arboretum.namereny_bod
> rostliny.sql
```

Popis parametrů:

- `-s 4326` určuje uložení v souřadnicovém systému WGS 84
- `-I` je pro vytvoření GiSt indexu na geometrickém sloupci
- `-a` označuje, že se data budou přidávat do dabázové tabulky
- `-W UTF-8` určuje kódování znaků
- `rostliny.shp` vstupní soubor
- `arboretum.namereny_bod` název tabulky, do které se data vkládají
- `> rostliny.sql` příkaz pro vytvoření SQL souboru

Vytvořili jsme SQL soubor, který je potřeba spustit nad databází, proto je potřeba se přepnout jako postgres uživatel:

```
su - postgres
```

Pomocí nástroje `psql` provedeme konečné přidání dat do databáze:

```
psql -d gis -f rostliny.sql
```

Nyní jsou naše naměřená data nahraná v databázi.

7 Publikace dat z databáze na web

Pro reprezentaci dat je nutné data z PostGIS databáze zobrazit pomocí služby WMS. Pro tento účel jsem využil GeoServer. GeoServer jsem zprovoznil na aplikačním serveru Apache Tomcat 7, který je založený na jazyce Java. Instalace serveru se provede příkazem:

```
sudo apt-get update  
sudo apt-get install tomcat7
```

Je také nutné nainstalovat soubor základních nástrojů pro platformu Java

```
sudo apt-get install openjdk-8-jre
```

Samotná instalace Geoserveru je jednoduchá, z oficiálních stránek je nutné stáhnout archiv `geoserver.war.zip`, který potřebuje rozbalit. Rozbalená složka je připravena pro přesunutí do složky `webapps` nacházející se v Tomcat serveru, v méém případě: `/var/lib/tomcat7/webapps`

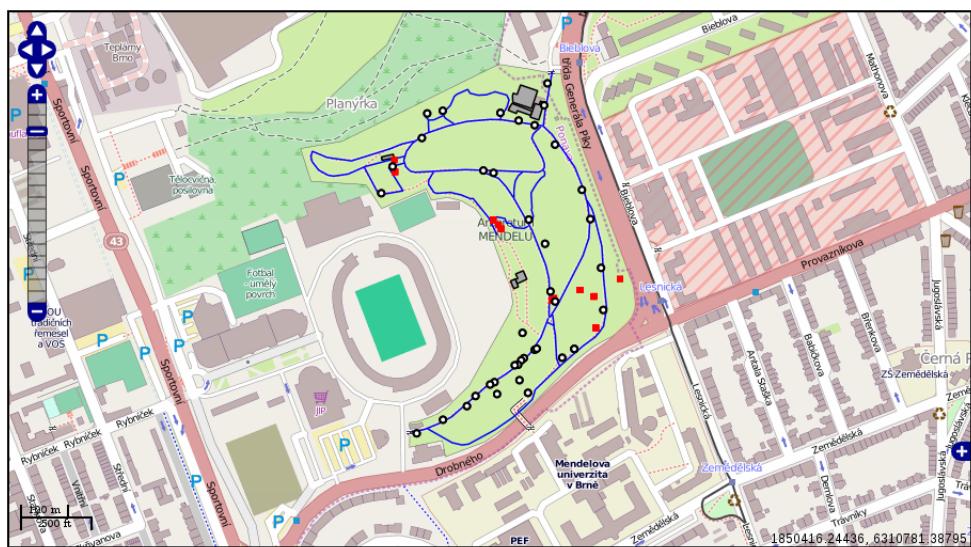
Pomocí uživatelského prostředí jsem se připojil k databázi, vybral tabulky, které chci zobrazovat a vytvořil vrstvy WMS, se souřadnicovým systémem WGS84. Zobrazení výsledné mapy, jsem dosáhnul javascriptovou knihovnou OpenLayers. Jako nejlepším řešení pro podkladovou vrstvu lze použít základní OpenStreetMap mapu. OpenStreetMap (OSM) je komunita uživatelů, kteří přidávají a upravují mapová data. OSM tvorí otevřená data, což znamená, že je lze použít pro libovolný účel.

Ukázka scriptu tvořícího mapu:

```
<script defer="defer" type="text/javascript">  
var map = new OpenLayers.Map('map', {  
    numZoomLevels: 30,  
    projection: "EPSG:4326"  
});  
var geoServer = new OpenLayers.Layer.WMS("OpenLayers WMS",  
"http://localhost:8080/geoserver/wms", {  
    layers: [namereny_bod, budova, znacka, cesta, vybaveni],  
    isBaseLayer: false,  
    transparent: true  
});  
var podkladovaMapa = new OpenLayers.Layer.OSM("Simple OSM Map");  
  
map.addLayer(podkladovaMapa);  
map.addLayer(geoServer);  
  
map.zoomToExtent(geoServer.getExtent());  
</script>
```

Důležité je přidání tagu script `<script src="http://openlayers.org/api/OpenLayers.js"></script>`, který přidává funkcionality javascriptové knihovny OpenLayers. Příkazem `new OpenLayers.Map()` se vytváří mapa, jeden povinný parametr je id HTML tagu, do kterého chceme mapu zobrazit, nejčastěji se jedná o tag `<div>`. Další parametr je nepovinný a určuje nastavení a možnosti mapy, v tomto příkladě to je `numZoomLevels` - počet urovní pro zoom a `projection` - určuje v jaké projekci se mapa zobrazí. Vlastní WMS vrstva, vytvořena pomocí GeoServeru, s datami z databáze je připojena pomocí příkazu `new OpenLayers.Layer.WMS()`, s prvním parametrem určující název, druhým definující url adresu s vrstvou a třetím nepovinným udávající možnosti vrstvy. Parametr `isBaseLayer`: `false` definuje, že vrstva není základní a `transparent`: `true` určuje průhlednou vrstvu. Podkladová mapa je vytvořena pomocí `new OpenLayers.Layer.OSM`. Poslední série příkazů přidává vytvořené vrstvy (WMS, OSM) do mapy (`map.addLayer()`) a nastaví přiblžení na celou WMS vrstvu (`map.zoomToExtent(geoServer.getExtent())`).

Mapa arboreta MENDELU



Obrázek 11: Ukázka jednoduché webové stránky.

8 Diskuze

8.1 Srovnání databáze s BotanGIS

Ve srovnání například s geodatabází projektu BotanGis má tato geodatabáze méně tabulek, což odpovídá požadavkům na oba projekty. BotanGis je zaměřen na inovaci a zvýšení atraktivity studijních oborů botaniky a geoinformatiky na Univerzitě Palackého v Olomouci.(Dobešová, 2013) Výsledkem projektu je botanický informační systém a studijní, výukové a metodické materiály pro oba studijní obory. Databáze je zaměřena na uchování rostlin s mnoha parametry, mezi které patří informace o listech, tvaru květu, vývojovém stádiu rostliny, kdo rostlinu zasadil a mnoho dalších odborných informací. BotanGis je více odborný projekt s úplnou implementací prohlížecí služby pro venkovní botanickou zahradu a skleníky. Podobné části návrhu jsou fotodokumentace rostlin, číselníky s čeledí a rodem a základní informace o rostlinách.

Jak již jsem zmínil dříve projekt BotanGIS je kompletní GIS, proto jeho tvorba trvala přibližně necelé dva roky. Obsahuje všechny data dostupná v botanické zahradě a jejích skleníků. S tímto souvisí i sběr dat. V rámci BotanGISu probíhal sběr dat pomocí kvalitních geodetických přístrojů, kterým se použité přístroje v této práci nemohou kvalitou ani cenou rovnat.

8.2 Možnosti rozšíření

Nejvíce se nabízí možnost rozšíření této práce do podoby kompletního GIS, který by mohl být použit pro účely arboreta a pro veřejnost. Hlavní část by bezpochyby byla prohlížecí služba, v podobě volně dostupných služeb jako WMS, WFS, nebo interaktivní mapa určená pro webový prohlížeč. S interaktivní mapou by se mohl vytvořit kompletní informační systém, který by obyčejnému uživateli jen zobrazoval data a další volně dostupné informace. Po přihlášení by uživatel mohl mít určité role, díky kterým by měl různé stupně práce s daty. Mezi takové práce by patřilo například nahrávání nových dat, které by bylo vyřešeno pomocí uživatelsky přívětivého prostředí, kterým je například nahrávací formulář. Uživatel by mohl jen načíst vrstvu s geografickými daty uloženou ve svém počítači a funkce informačního systému by se postarala o převedení dat do databáze. Oprávnění uživatelé by také mohli mít možnost data mazat a samozřejmě upravovat, například název, styl zobrazení, shromažďovat data do vybraných zobrazovacích skupin. Webová část takového systému by mohla sloužit i pro studenty univerzity jako učící pomůcka. Například studenti informatických předmětů by mohli vytvářet nová data, která by mohla sloužit i pracovníkům arboreta. Zatímco studenti botanických oborů by mohli důkladně studovat rostliny jak v terénu, tak i v pohodlí svého domova. V rámci GIS arboreta, by bylo dobré vytvořit aplikaci pro mobilní telefony určenou pro měření dat a jejich rychlé ukládání do databáze. Aplikace by musela mít mnoho omezení a sloužila by jen pro vybrané uživatele/zaměstnance arboreta.

Další možností pro další využití se nabízí mobilní aplikace pro návštěvníky arboreta. Pokud by byl vytvořen komplexní GIS arboreta, mohla komunikovat s tímto serverem a byla jeho zjednodušenou podobou. V jiném případě by aplikace sloužila jen jako plánek s možnostmi zjištění informací o oblasti, ve které se uživatel právě nachází. Kromě mapy by mohla obsahovat také atlas rostlin, které se nacházejí v arboretu s jejich podrobným popisem. V atlasu by se mohlo vyhledávat podle identifikačního čísla rostliny, umístěného na štítku, nebo podle názvu.

9 Závěr

V první části práce byly shrnuty teoretické informace. Byly představeny geografické informační systémy, data se kterými pracují a souřadnicové systémy. Také byly představeny softwarové aplikace, které byly využity pro praktickou část.

Volba programových prostředků byla hlavně o jejich dostupnosti. Všechny využité prostředky byly volně dostupné, bez jakýchkoliv poplatků. Z dostupných databází vyšla nejlépe PostgreSQL s rozšířením PostGIS. Nástroj pro úpravu dat byl vybrán QGIS, který nabízí mnoho funkcí a splňuje na rozdíl od jiných podobných aplikací hlavní požadavek o dostupnosti. Pro sdílení dat byl vybrán GeoServer, který zajišťuje uživatelsky přívětivé prostředí a kvalitní výsledek. Umožnuje připojení k PostGIS a poskytuje její data v optimálním množství různých typů souborů a služeb. Webová prezentace byla usnadněna javascriptovou knihovnou OpenLayers.

Vytvoření návrhu databáze bylo nejdelší časovou částí strávenou tvorbou práce. Bylo nutné uvažovat mnoho faktorů jak by měl návrh vypadat a jak by měla databáze fungovat. Návrh vyžadoval prozkoumání samotného arboreta, pro zjištění možností ukladaných dat. Dalším krokem bylo prostudování prostorových datových typů a jejich funkcionality. Na tomto základě vznikly tabulky sloužící k jejich uložení. Věřím, že databáze může mít nedostatky, které se však mohou projevit až po delším používání. Například pokud bude chtít uložit data s jiným datovým typem.

Dalším cílem bylo určení přesnosti použitych měřících přístrojů. Pro tento cíl bylo nezbytné prostudovat měřící přístroje a aplikace pomocí, kterých byla měření prováděna. Další postup vyžadoval prostudování možností určení přesnosti přijímače GPS signálu. Byla zvolena metoda pro určení odchylky. Po provedení měření, výpočtu a sestavení grafu, byl znám výsledek, který určil jako přesnější meření aplikací Locus Map na zařízení HTC. Následující cíl uváděl vytvoření metodiky měření v areálu arboreta. To vyžadovalo provést pokusná měření, z kterých byl návod sestaven.

V práci je ukázan celkový postup pro tvorbu databáze, její instalaci, sběr dat, zprovoznění mapového serveru a ukázka tvorby jednoduché mapy zobrazené formou HTML stránky. V příloze naleznete SQL skript, ve kterém jsou všechny příkazy potřebné k vytvoření databáze, návrh databáze a vzhled webové prezentace. Součást práce je i samostatně přiložené DVD, obsahující obraz operačního systému Debian, na kterém jsou všechny popsané části práce vytvořené.

10 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.
- BOUNLESS *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>.
- DOBEŠOVÁ, ZDENA *Tvorba geografického informačního systému malého území*. 1. vyd. Olomouc: Univerzita Palackého v Olomouci pro katedru geoinformatiky, 2013, 106 s. ISBN 978-80-244-3825-2.
- ESRI *Coordinate systems, map projections, and geographic (datum) transformations*. ArcGIS Resources [online]. [cit. 2014-05-01]. Dostupné z: <http://resources.esri.com/help/9.3/arcgisengine/dotnet/89b720a5-7339-44b0-8b58-0f5bf2843393.htm>.
- ESRI *Geographic Information System*. GIS Mapping Software, Solutions, Services, Map Apps, and Data [online]. [cit. 2014-04-10]. Dostupné z: <http://www.esri.com/what-is-gis/overview>.
- GIS LOUNGE *GIS Data Explored*. Maps and GIS [online]. [cit. 2014-04-12]. Dostupné z: <http://www.gislounge.com/geodatabases-explored-vector-and-raster-data/>.
- HSU, L S. – OBE, R O *PostGIS in Action, Second Edition*. USA: Manning Publications Co, 2014. 625 s. ISBN 978-1-6172-9139-5.
- MAREK, JOSEF A LADISLAV ŠTĚPÁNEK *Přesnost satelitního navigačního systému GPS a jeho dostupnost v kritických podmínkách*. 2009, roč. 4, č. 3, s. 12. Dostupné z: http://pernerscontacts.upce.cz/15_2009/Marek.pdf.
- OpenLayers*. dokumentace [online]. 2014 [cit. 2014-05-09]. Dostupné z: <http://trac.osgeo.org/openlayers/wiki/Documentation>.
- ORACLE *Oracle Spatial*. dokumentace [online]. 2013 [cit. 2014-05-05]. Dostupné z: <http://www.oracle.com/cz/products/database/options/spatial/index.html>.
- PostGIS*. dokumentace [online]. 2014 [cit. 2014-05-09]. Dostupné z: <http://postgis.net/documentation>.
- RIGAUX, P. – SCHOLL, M O. – VOISARD, A *Spatial databases : with application to GIS*. San Francisco: Morgan Kaufmann Publishers, 2002. 410 s. The Morgan Kaufmann series in data management systems. ISBN 1-55860-588-6.

ŠEBESTA, JIŘÍ *Globální navigační systémy*. Vyd. 1. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2012, 132 s. ISBN 978-80-214-4500-0.

WESTRA, ERIK *Python geospatial development: build a complete and sophisticated mapping application from scratch using Python tools for GIS development*. 1st pub. Birmingham: Packt, 2010, viii, 487 s. ISBN 978-184-9511-544.

Přílohy

A SQL pro vytvoření databáze

```
CREATE SCHEMA arboretum;
ALTER SCHEMA arboretum OWNER TO uzivatel_arboretum;

SET search_path TO pg_catalog,public,arboretum;

CREATE SEQUENCE arboretum.objekty_id_objekty_seq
INCREMENT BY 1
MINVALUE 1
MAXVALUE 9223372036854775807
START WITH 1
CACHE 1
NO CYCLE
OWNED BY NONE;
ALTER SEQUENCE arboretum.objekty_id_objekty_seq
OWNER TO uzivatel_arboretum;

CREATE TABLE arboretum.objekt(
id_objekt integer NOT NULL DEFAULT
nextval('arboretum.objekty_id_objekty_seq'::regclass),
objekt integer NOT NULL,
CONSTRAINT objekt_pkey PRIMARY KEY (id_objekt)
);
ALTER TABLE arboretum.objekt OWNER TO uzivatel_arboretum;

CREATE SEQUENCE arboretum.znacky_gid_znacky_seq
INCREMENT BY 1
MINVALUE 1
MAXVALUE 9223372036854775807
START WITH 1
CACHE 1
NO CYCLE
OWNED BY NONE;
ALTER SEQUENCE arboretum.znacky_gid_znacky_seq
OWNER TO uzivatel_arboretum;

CREATE TABLE arboretum.znacka(
gid_znacka integer NOT NULL DEFAULT
nextval('arboretum.znacky_gid_znacky_seq'::regclass),
nazev varchar(255) NOT NULL,
hlavni_cislo integer NOT NULL,
vedlejsi_cislo integer NOT NULL,
```

```
popis text,
geom geometry(POINT, 4326) NOT NULL,
CONSTRAINT znacka_pkey PRIMARY KEY (gid_znacka)
);
ALTER TABLE arboretum.znacka OWNER TO uzivatel_arboretum;

CREATE SEQUENCE arboretum.budovy_gid_budovy_seq
INCREMENT BY 1
MINVALUE 1
MAXVALUE 9223372036854775807
START WITH 1
CACHE 1
NO CYCLE
OWNED BY NONE;
ALTER SEQUENCE arboretum.budovy_gid_budovy_seq
OWNER TO uzivatel_arboretum;

CREATE TABLE arboretum.budova(
gid_budova integer NOT NULL DEFAULT
nextval('arboretum.budovy_gid_budovy_seq'::regclass),
nazev varchar(50) NOT NULL,
popis text,
geom geometry(MULTIPOLYGON, 4326) NOT NULL,
CONSTRAINT budova_pkey PRIMARY KEY (gid_budova)
);
ALTER TABLE arboretum.budova OWNER TO uzivatel_arboretum;

CREATE SEQUENCE arboretum.vybaveni_gid_vybaveni_seq
INCREMENT BY 1
MINVALUE 1
MAXVALUE 9223372036854775807
START WITH 1
CACHE 1
NO CYCLE
OWNED BY NONE;
ALTER SEQUENCE arboretum.vybaveni_gid_vybaveni_seq
OWNER TO uzivatel_arboretum;

CREATE TABLE arboretum.vybaveni(
gid_vybaveni integer NOT NULL DEFAULT
nextval('arboretum.vybaveni_gid_vybaveni_seq'::regclass),
id_typ_vybaveni integer NOT NULL,
nazev varchar(50) NOT NULL,
```

```
popis text,
geom geometry(POINT, 4326) NOT NULL,
CONSTRAINT vybaveni_pkey PRIMARY KEY (gid_vybaveni)
);
ALTER TABLE arboretum.vybaveni OWNER TO uzivatel_arboretum;

CREATE SEQUENCE arboretum.cesty_gid_cesty_seq
INCREMENT BY 1
MINVALUE 1
MAXVALUE 9223372036854775807
START WITH 1
CACHE 1
NO CYCLE
OWNED BY NONE;
ALTER SEQUENCE arboretum.cesty_gid_cesty_seq
OWNER TO uzivatel_arboretum;

CREATE TABLE arboretum.cesta(
gid_cesta integer NOT NULL DEFAULT
nextval('arboretum.cesty_gid_cesty_seq'::regclass),
nazev varchar(50) NOT NULL,
geom geometry(MULTILINESTRING, 4326) NOT NULL,
popis text,
CONSTRAINT cesta_pkey PRIMARY KEY (gid_cesta)
);
ALTER TABLE arboretum.cesta OWNER TO uzivatel_arboretum;

CREATE SEQUENCE arboretum.typy_vybaveni_id_typy_vybaveni_seq
INCREMENT BY 1
MINVALUE 1
MAXVALUE 9223372036854775807
START WITH 1
CACHE 1
NO CYCLE
OWNED BY NONE;
ALTER SEQUENCE arboretum.typy_vybaveni_id_typy_vybaveni_seq
OWNER TO uzivatel_arboretum;

CREATE TABLE arboretum.typ_vybaveni(
id_typ_vybaveni integer NOT NULL DEFAULT
nextval('arboretum.typy_vybaveni_id_typy_vybaveni_seq'::regclass),
nazev varchar(50) NOT NULL,
CONSTRAINT typ_vybaveni_pkey PRIMARY KEY (id_typ_vybaveni)
```

```
);

ALTER TABLE arboretum.typ_vybaveni OWNER TO uzivatel_arboretum;

CREATE SEQUENCE arboretum.namerene_body_gid_namerene_body_seq
INCREMENT BY 1
MINVALUE 1
MAXVALUE 9223372036854775807
START WITH 1
CACHE 1
NO CYCLE
OWNED BY NONE;
ALTER SEQUENCE arboretum.namerene_body_gid_namerene_body_seq
OWNER TO uzivatel_arboretum;

CREATE TABLE arboretum.namereny_bod(
gid_namereny_bod integer NOT NULL DEFAULT
nextval('arboretum.namerene_body_gid_namerene_body_seq'::regclass),
nazev varchar(100) NOT NULL,
geom geometry(POINT, 4326) NOT NULL,
CONSTRAINT namereny_bod_pkey PRIMARY KEY (gid_namereny_bod)
);
ALTER TABLE arboretum.namereny_bod OWNER TO uzivatel_arboretum;

CREATE SEQUENCE arboretum.cesty_typy_cesty_id_cesty_typt_cesty_seq
INCREMENT BY 1
MINVALUE 1
MAXVALUE 9223372036854775807
START WITH 1
CACHE 1
NO CYCLE
OWNED BY NONE;
ALTER SEQUENCE arboretum.cesty_typy_cesty_id_cesty_typt_cesty_seq
OWNER TO uzivatel_arboretum;

CREATE TABLE arboretum.cesta_typ_cesty(
gid_cesta_typ_cesty integer NOT NULL DEFAULT
nextval('arboretum.cesty_typy_cesty_id_cesty_typt_cesty_seq'::regclass),
id_cesta integer NOT NULL,
id_typ_cesty integer NOT NULL,
CONSTRAINT cesta_typ_cesty_pkey PRIMARY KEY (gid_cesta_typ_cesty)
);
ALTER TABLE arboretum.cesta_typ_cesty OWNER TO uzivatel_arboretum;
```

```
CREATE SEQUENCE arboretum.typy_cesty_id_typy_cesty_seq
INCREMENT BY 1
MINVALUE 1
MAXVALUE 9223372036854775807
START WITH 1
CACHE 1
NO CYCLE
OWNED BY NONE;
ALTER SEQUENCE arboretum.typy_cesty_id_typy_cesty_seq
OWNER TO uzivatel_arboretum;

CREATE TABLE arboretum.typ_cesty(
id_typ_cesty integer NOT NULL DEFAULT
nextval('arboretum.typy_cesty_id_typy_cesty_seq'::regclass),
nazev varchar(50) NOT NULL,
oznaceni varchar(50) NOT NULL,
popis text,
CONSTRAINT typ_cesty_pkey PRIMARY KEY (id_typ_cesty)
);
ALTER TABLE arboretum.typ_cesty OWNER TO uzivatel_arboretum;

CREATE SEQUENCE arboretum.tridy_id_tridy_seq
INCREMENT BY 1
MINVALUE 1
MAXVALUE 9223372036854775807
START WITH 1
CACHE 1
NO CYCLE
OWNED BY NONE;
ALTER SEQUENCE arboretum.tridy_id_tridy_seq
OWNER TO uzivatel_arboretum;

CREATE TABLE arboretum.trida(
id_trida integer NOT NULL DEFAULT
nextval('arboretum.tridy_id_tridy_seq'::regclass),
nazev_cz varchar(255) NOT NULL,
nazev_lat varchar(255) NOT NULL,
nazev_sk varchar(255),
CONSTRAINT trida_pkey PRIMARY KEY (id_trida)
);
ALTER TABLE arboretum.trida OWNER TO uzivatel_arboretum;

CREATE SEQUENCE arboretum.celede_id_celede_seq
```

```
INCREMENT BY 1
MINVALUE 1
MAXVALUE 9223372036854775807
START WITH 1
CACHE 1
NO CYCLE
OWNED BY NONE;
ALTER SEQUENCE arboretum.celede_id_celede_seq
OWNER TO uzivatel_arboretum;

CREATE TABLE arboretum.celed(
id_celed integer NOT NULL DEFAULT
nextval('arboretum.celede_id_celede_seq'::regclass),
nazev_cz varchar(255) NOT NULL,
nazev_lat varchar(255) NOT NULL,
nazev_sk varchar(255),
CONSTRAINT celed_pkey PRIMARY KEY (id_celed)
);
ALTER TABLE arboretum.celed OWNER TO uzivatel_arboretum;

CREATE SEQUENCE arboretum.rady_id_rady_seq
INCREMENT BY 1
MINVALUE 1
MAXVALUE 9223372036854775807
START WITH 1
CACHE 1
NO CYCLE
OWNED BY NONE;
ALTER SEQUENCE arboretum.rady_id_rady_seq OWNER TO uzivatel_arboretum;

CREATE TABLE arboretum.rad(
id_rad integer NOT NULL DEFAULT
nextval('arboretum.rady_id_rady_seq'::regclass),
nazev_cz varchar(255) NOT NULL,
nazev_lat varchar(255) NOT NULL,
nazev_sk varchar(255),
CONSTRAINT rad_pkey PRIMARY KEY (id_rad)
);
ALTER TABLE arboretum.rad OWNER TO uzivatel_arboretum;

CREATE SEQUENCE arboretum.objekty_fotky_id_objekty_fotky_seq
INCREMENT BY 1
MINVALUE 1
```

```
MAXVALUE 9223372036854775807
START WITH 1
CACHE 1
NO CYCLE
OWNED BY NONE;
ALTER SEQUENCE arboretum.objekty_fotky_id_objekty_fotky_seq
OWNER TO uzivatel_arboretum;

CREATE TABLE arboretum.objekt_fotka(
id_objekt_fotka integer NOT NULL DEFAULT
nextval('arboretum.objekty_fotky_id_objekty_fotky_seq'::regclass),
id_objekt integer NOT NULL,
id_fotka integer NOT NULL,
CONSTRAINT objekt_fotka_pkey PRIMARY KEY (id_objekt_fotka)
);
ALTER TABLE arboretum.objekt_fotka OWNER TO uzivatel_arboretum;

CREATE SEQUENCE arboretum.fotky_id_fotky_seq
INCREMENT BY 1
MINVALUE 1
MAXVALUE 9223372036854775807
START WITH 1
CACHE 1
NO CYCLE
OWNED BY NONE;
ALTER SEQUENCE arboretum.fotky_id_fotky_seq
OWNER TO uzivatel_arboretum;

CREATE TABLE arboretum.fotka(
id_fotka integer NOT NULL DEFAULT
nextval('arboretum.fotky_id_fotky_seq'::regclass),
url varchar(255) NOT NULL,
popis text,
CONSTRAINT fotka_pkey PRIMARY KEY (id_fotka)
);
ALTER TABLE arboretum.fotka OWNER TO uzivatel_arboretum;

CREATE SEQUENCE arboretum.dreviny_id_dreviny_seq
INCREMENT BY 1
MINVALUE 1
MAXVALUE 9223372036854775807
START WITH 1
CACHE 1
```

```
NO CYCLE
OWNED BY NONE;
ALTER SEQUENCE arboretum.dreviny_id_dreviny_seq
OWNER TO uzivatel_arboretum;

CREATE SEQUENCE arboretum.rostliny_id_rostliny_seq
INCREMENT BY 1
MINVALUE 1
MAXVALUE 9223372036854775807
START WITH 1
CACHE 1
NO CYCLE
OWNED BY NONE;
ALTER SEQUENCE arboretum.rostliny_id_rostliny_seq
OWNER TO uzivatel_arboretum;

CREATE TABLE arboretum.rostlina(
gid_rostlina integer NOT NULL DEFAULT
nextval('arboretum.rostliny_id_rostliny_seq'::regclass),
id_namereny_bod integer NOT NULL,
nazev_cz varchar(255) NOT NULL,
nazev_lat varchar(255) NOT NULL,
nazev_sk varchar(255),
id_trida integer,
id_celed integer,
id_rad integer,
poznamka text,
identifikacni_cislo integer,
koruna geometry(POINT),
CONSTRAINT rostlina_pkey PRIMARY KEY (gid_rostlina),
CONSTRAINT rostlina_id_namereny_bod_key UNIQUE (id_namereny_bod)
);
ALTER TABLE arboretum.rostlina OWNER TO uzivatel_arboretum;

ALTER TABLE arboretum.objekt ADD CONSTRAINT objekt_fkey1
FOREIGN KEY (objekt)
REFERENCES arboretum.znacka (gid_znacka) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;

ALTER TABLE arboretum.objekt ADD CONSTRAINT objekt_fkey2
FOREIGN KEY (objekt)
REFERENCES arboretum.budova (gid_budova) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
```

```
ALTER TABLE arboretum.objekt ADD CONSTRAINT objekt_fkey3
FOREIGN KEY (objekt)
REFERENCES arboretum.cesta (gid_cesta) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;

ALTER TABLE arboretum.objekt ADD CONSTRAINT objekt_fkey4
FOREIGN KEY (objekt)
REFERENCES arboretum.vybaveni (gid_vybaveni) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;

ALTER TABLE arboretum.objekt ADD CONSTRAINT objekt_fkey5
FOREIGN KEY (objekt)
REFERENCES arboretum.namereny_bod (gid_namereny_bod) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;

ALTER TABLE arboretum.vybaveni ADD CONSTRAINT vybaveni_fkey1
FOREIGN KEY (id_typ_vybaveni)
REFERENCES arboretum.typ_vybaveni (id_typ_vybaveni) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;

ALTER TABLE arboretum.cesta_typ_cesty
ADD CONSTRAINT cesta_typ_cesty_fkey1
FOREIGN KEY (id_cesta)
REFERENCES arboretum.cesta (gid_cesta) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;

ALTER TABLE arboretum.cesta_typ_cesty
ADD CONSTRAINT cesta_typ_cesty_fkey2
FOREIGN KEY (id_typ_cesty)
REFERENCES arboretum.typ_cesty (id_typ_cesty) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;

ALTER TABLE arboretum.objekt_fotka ADD CONSTRAINT objekt_fotka_fkey1
FOREIGN KEY (id_objekt)
REFERENCES arboretum.objekt (id_objekt) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;

ALTER TABLE arboretum.objekt_fotka ADD CONSTRAINT objekt_fotka_fkey2
FOREIGN KEY (id_fotka)
REFERENCES arboretum.fotka (id_fotka) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
```

```
ALTER TABLE arboretum.rostlina ADD CONSTRAINT rostlina_fkey1
FOREIGN KEY (id_trida)
```

```
REFERENCES arboretum.trida (id_trida) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
```

```
ALTER TABLE arboretum.rostlina ADD CONSTRAINT rostlina_fkey2
FOREIGN KEY (id_celed)
```

```
REFERENCES arboretum.celed (id_celed) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
```

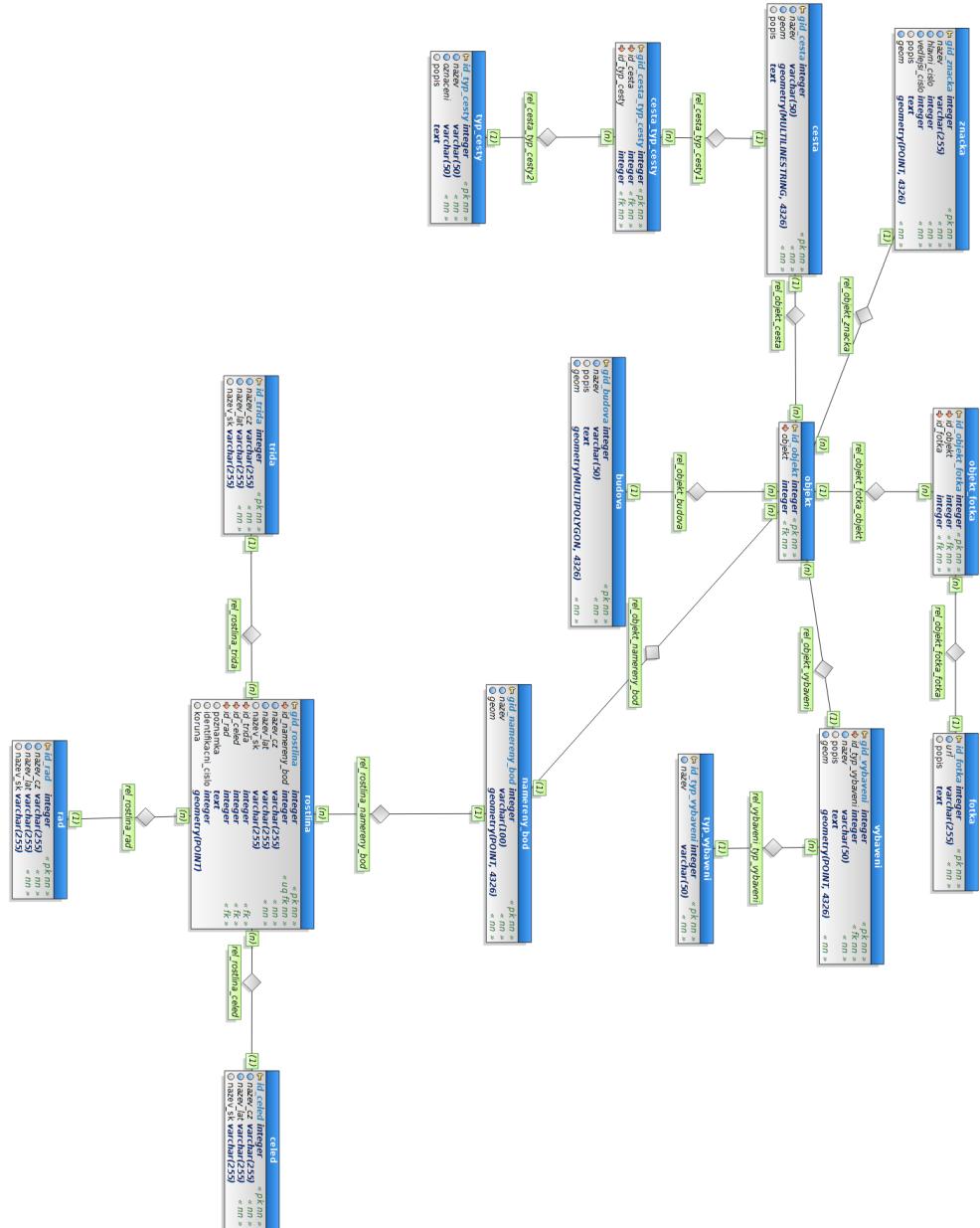
```
ALTER TABLE arboretum.rostlina ADD CONSTRAINT rostlina_fkey3
FOREIGN KEY (id_rad)
```

```
REFERENCES arboretum.rad (id_rad) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
```

```
ALTER TABLE arboretum.rostlina ADD CONSTRAINT rostlina_fkey4
FOREIGN KEY (id_namereny_bod)
```

```
REFERENCES arboretum.namereny_bod (gid_namereny_bod) MATCH FULL
ON DELETE NO ACTION ON UPDATE NO ACTION;
```

B Schéma databáze



C Hesla pro přiložené DVD

Přihlašovací heslo do systému - Uživatel: uzivatel

Heslo superuživatele - Root: root

Uživatelé PostgreSQL:

postgres: postgres

arboretum_uzivatel: arboretum

Přihlašování do webové administrace GeoServeru - admin: geoserver