

Tvorba WWW prezentace pro autonomního robota

Bakalářská práce

Vedoucí práce:

Ing. Jiří Lýsek, Ph.D.

Jan Zálešák

Brno 2015

Srdečně děkuji vedoucímu bakalářské práce Ing. Jiřímu Lýskovi, Ph.D. za jeho velmi cenné rady a pomoc, zpracovávání panoramatických fotografií. A v neposlední řadě také děkuji Bc. Michalovi Šturmovi a Monice Brázdilové za jejich cenné rady, zapůjčení stativu a pomoc s kontrolou výsledné dokumentace bakalářské práce.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Tvorba WWW prezentace pro autonomního robota** 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 1. května 2015

Abstract

Zálešák, J. Production of WWW presentation for autonomous robot. Bachelor thesis. Brno: Mendel university in Brno, 2015.

This bachelor's thesis deals with designing and production of web application for presentation of panoramic photographs of Mendel university arboretum in Brno. Taking photographs is temporarily substituted by hand-operated camera until autonomous robot is completely functional. This thesis covers particular description of used technologies and procedures of application. The result is detailed instruction manual of application and its economic evaluation.

Keywords

Mendel university arboretum, panoramic photographs, web application, PHP, jQuery, MySQL, Nette, Google Maps

Abstrakt

Zálešák, J. Tvorba WWW prezentace pro autonomního robota. Bakalářská práce. Brno: Mendelova univerzita v Brně, 2015.

Tato bakalářská práce se zabývá návrhem a tvorbou webové aplikace pro prezentaci panoramatických fotografií arboreta v areálu Mendelovy univerzity v Brně. Fotografování arboreta je dočasně nahrazeno ručním fotoaparátem, dokud nebude autonomní robot plně funkční. Obsahem jsou jednotlivé popisy použitých technologií a postupy tvorby aplikace. Ve výsledku je popsán podrobný návod na obsluhu celé aplikace a její ekonomické zhodnocení.

Klíčová slova

Arboretum Mendelovy univerzity, panoramatické fotografie, webová aplikace, PHP, jQuery, MySQL, Nette, Mapy Google

Obsah

1	Úvod	9
2	Cíle bakalářské práce	10
2.1	Cíle teoretické části práce	10
2.2	Cíle praktické části práce	10
3	Použité technologie pro vývoj aplikace	11
3.1	Tvorba statických stránek pomocí jazyka HTML.....	11
3.2	Kaskádové styly – CSS	11
3.2.1	Responzivní design.....	12
3.3	Databázový systém MySQL.....	12
3.4	Skriptovací programovací jazyk PHP	13
3.4.1	PHP jako samostatný skript.....	14
3.4.2	PHP ve spolupráci s databázovým systémem	14
3.4.3	MVC a MVP architektura	15
3.5	ER Diagram a jazyk UML	17
3.6	MVC a MVP frameworky pro vývoj webových aplikací.....	17
3.7	Programovací jazyk JavaScript	18
3.7.1	Asynchronní zpracování webových stránek technologií AJAX.....	19
3.8	Zpracování fotek specializovaným softwarem	20
4	Metodika práce	21
4.1	Analýza a výběr daného frameworku	21
4.2	Analýza a výběr dané databáze.....	21
4.3	Analýza a výběr dané knihovny JavaScriptu.....	23
4.4	Postup návrhu aplikace.....	25
4.4.1	Vývojové prostředí	25
4.4.2	Struktura aplikace.....	25
4.4.3	Struktura databáze	27
4.4.4	Vzhled aplikace	28

4.5	Pořízení a zpracování panoramatických fotografií	29
5	Výsledná aplikace	30
5.1	Prezentační část	30
5.1.1	Interaktivní mapa školního arboreta	30
5.1.2	Galerie jízd	31
5.2	Administrační část	31
5.2.1	Stránka statistik a zaznamenaných činností uživatelů	31
5.2.2	Přidání jízdy	31
5.2.3	Seznam jízd	32
5.2.4	Editace jízdy	33
5.2.5	Přidání uživatele	33
5.2.6	Seznam uživatelů	33
5.2.7	Editace uživatele	34
5.3	Ekonomické zhodnocení aplikace	34
6	Závěr	36
7	Literatura	37
A	Dokumentace problému s roamingem	41
B	Interaktivní mapa jízd	42
C	Zobrazení panoramatické fotografie	43
D	Seznam stejných pozic zobrazené fotografie	44
E	Zobrazení fotografie stejné pozice jiného období	45
F	Galerie jízd	46
G	Statistiky a provedené činnosti uživatelů	47
H	Vytvoření nové jízdy	48
I	Seznam jízd	49
J	Editace existující jízdy	50
K	Vytvoření nového uživatele	51

L Seznam uživatelů 52

M Editace uživatele 53

1 Úvod

Bakalářská práce se zabývá tvorbou webové aplikace prezentující panoramatické fotografie arboreta Mendelovy univerzity v Brně. Tato unikátní prezentační aplikace, kterou v aktuální době nedisponuje žádná jiná univerzita jak v Jihomoravském kraji, tak v celé České republice, je vytvořena za účelem propagace školního arboreta a školy samotné. Díky ní má široká veřejnost možnost vidět, co v našem arboretu mohou osobně shlédnout a jedná se také o velmi zajímavou aplikaci vyzdvihující fakultu informatiky mezi fakultami jiných univerzit.

Ve finální práci budou fotografie pořizovány autonomním robotem vytvořeným týmem AiStorm z ústavu informatiky Provozně ekonomické fakulty Mendelovy univerzity v Brně. Tento robot však zatím není plně funkční, čímž je do práce provizorně doplněno vlastní fotografování školního arboreta. Fotografie se následně zpracují ve specializovaném softwaru do panoramatického zobrazení a nahrají do aplikace.

Obsahem bakalářské práce jsou postupy tvorby webové aplikace, popis jednotlivých použitých technologií a kompletní návod pro obsluhu aplikace.

Pro celkovou tvorbu WWW aplikace bude použit framework Nette, který programování dodává potřebnou MVP strukturu a zahrnuje mnoho dalších důležitých prvků pro kvalitní aplikaci. Jednotlivými prvky pro vytvoření statických stránek se použijí technologie HTML5 a CSS3. Programovací jazyk PHP zajistí spojení stránek s databází a ostatní technologie a frameworky, jako například JQuery, GMap3 a další doplní interaktivitu.

2 Cíle bakalářské práce

2.1 Cíle teoretické části práce

Cílem teoretické části bakalářské práce je v hlavních bodech popsat technologie použité v aplikaci, analyzovat je a zdůvodnit, proč byly tyto technologie vybrány.

2.2 Cíle praktické části práce

Cílem praktické části bakalářské práce je vytvořit aplikaci schopnou pojmout velké množství panoramatických fotografií nejprve pořízených vlastním fotoaparátem a zpracovaných specializovaným softwarem a po určité době, jakmile bude autonomní robot schopen samostatné práce, snímky pořízené jím a praktickým způsobem je prezentovat pro širokou veřejnost na stránkách Mendelovy univerzity.

V aplikaci musí být zahrnuta jak interaktivní prezentace pořízených fotografií s možností porovnávání jízd z různých období, tak administrační část, kam budou mít přístup pouze oprávnění uživatelé, kteří budou mít na starosti administraci fotografií a administrátorů aplikace. Jednou z částí prezentace bude zapojení interaktivní mapy společnosti Google Inc., zobrazující arboretum Mendelovy univerzity. V posledním bodě je také shrnuto ekonomické zhodnocení aplikace. Ve shrnutí se tedy jedná o popis implementace a popis výsledku a obsluhy aplikace.

3 Použité technologie pro vývoj aplikace

Výsledná webová aplikace musí splňovat mnoho předpokladů pro to, aby mohla být bezpečně a funkčně zveřejněna. Programovací jazyky, pro splnění těchto předpokladů, se postupem času stále zdokonalují. S tím i jejich frameworky, což zahrnuje zdokonalování bezpečnostní stránky aplikace a zjednodušování práce s daným jazykem.

O vzhledovou stránku a zobrazení prvků na stránce se stará jazyk HTML a kaskádové styly, které jsou pro tvorbu webové aplikace nezbytné. Dalším nepostradatelným prvkem je ukládání dat pro pozdější použití. Je tedy nutné vytvořit databázi, která bude data uchovávat a naimplementovat komunikaci mezi touto databází a aplikací. K tomuto účelu byl vybrán databázový systém MySQL, který je v současné době nejpoužívanějším systémem s rozsáhlou komunitní podporou. Stránku komunikační zaopatřuje skriptovací jazyk PHP. Další, méně nutné složky, jsou volitelné. Dodávají však aplikaci stránku propracovanosti, lepší uživatelské ovládání a hlavně bezpečnost používání aplikace a důvěryhodnost dat.

Další částí bakalářské práce je focení školního arboreta Mendelovy univerzity a zpracování fotografií do panoramatického zobrazení ve specializovaném softwaru. Jako nejvhodnější aplikace pro toto zpracování, s ohledem na komplexnost výsledné aplikace, byl vybrán program vytvořený týmem AiStorm.

3.1 Tvorba statických stránek pomocí jazyka HTML

Tento značkovací jazyk je základním pilířem pro tvorbu webových stránek a aplikací. Základy spadají až do roku 1993, kdy byla zveřejněna první verze jazyka. V průběhu 90. let byly velmi svižným tempem vydávány další verze 2, 3 a 4. I přes tento pokrok měl však jazyk konkurenci u více preferovaných jazyků XML a XHTML. To se však změnilo v roce 2004, kdy nově vytvořená skupina WHATWG (Web Hypertext Application Working Group), se kterou později začala spolupracovat skupina W3C (WorldWide Web Consortium), přinesla jazyk HTML5, který se používá jako webový standard do současnosti (Lubbers, Albers a Salim, 2011).

Postupem doby se z pouhého značkovacího jazyka stala celá aplikační platforma, schopná vytvářet jak statické webové stránky, tak vkládat audio a video soubory, vytvářet hry, animace a spoustu dalšího (Brown, Butters a Panda, 2014).

3.2 Kaskádové styly – CSS

CSS je úzce spojen se značkovacím jazykem HTML, o kterém byla řeč v kapitole 3.1. Značkám HTML totiž dodává vzhled a umístění na stránce.

CSS byl vydán až několik let po vzniku jazyka HTML a v současnosti je nejnovější verze 3, která je stejně jako jazyk HTML5 aktuálním standardem (Castro a Hyslop, 2012).

V prvních verzích jazyk nabízel pouze základní prvky vzhledu, ale postupem času se vyvinul na úroveň, kdy je schopen vytvářet profesionální designy obsahující například vizuální efekty, přechody a další interaktivní prvky (Castro a Hyslop, 2012).

3.2.1 Responzivní design

V dnešní době se aplikace neprezentují pouze na desktopových počítačích nebo noteboocích. Jsou zobrazovány uživateli také na mobilních telefonech, tabletech a dalších zařízeních. Proto není možné vytvořit jednoduchou vzhledovou šablonu, která bude fungovat pouze v konkrétním rozlišení, ale je nutné podle změny rozlišení měnit i rozložení a velikost jednotlivých prvků v aplikaci. Každé zařízení má totiž různou velikost a různé rozlišení obrazovky. Tuto potřebu řeší takzvaný responzivní design (Kadlec, 2014).

Na výběr jsou 4 různé typy layoutů vzhledu stránky (Kadlec, 2014):

- První, s pevnou šířkou, je základní statický design, který je naprogramován pro jedno konkrétní rozlišení. V dnešní době tedy poněkud nepraktické.
- Plovoucí layout, který nemá naimplementované rozměry v pixelech, nýbrž v procentech. Tím je dosažen o něco flexibilnější design stránky, který se změnou velikosti rozlišení lineárně zmenšuje nebo zvětšuje.
- Elastický layout je podobný plovoucímu. Základní jednotkou, podle které se určuje velikost všech ostatních prvků, je aktuálně nastavená velikost fontu v prohlížeči. Pomocí jednotky em (jedna jednotka em je rovna aktuální velikosti fontu) se poté určují násobky velikosti fontu pro dané prvky. Tím je zaručeno, že pokud se změní velikost rozlišení a také velikost výsledného fontu, změní se následně základní jednotka pro určování velikosti prvků.
- Posledním typem je hybridní layout, který kombinuje předešlé typy dohromady.

Mezi výběrem z těchto čtyř typů se programátor rozhoduje podle toho, o jaký projekt se jedná. Všechny typy mají své výhody a nevýhody.

3.3 Databázový systém MySQL

Databázový systém MySQL, spolu se skriptovacím jazykem PHP a webovým serverem Apache, patří do projektů s otevřeným zdrojovým kódem, tedy open source. Hnutí Open source se skládá ze schopných vývojářů a programátorů, kteří mezi sebou spolupracují. Díky této skupině mohou programátoři z celého světa přispívat svými nápady, a tak stále vylepšovat úroveň těchto technologií. Tyto tři technologie jsou převážně určeny pro instalaci do operačních systémů Linux. Uživatelé je ale stále častěji využívají v operačních systémech Windows, které jsou pro obsluhu přívětivější. Linux je však bezpečnější operační systém pro servery a je zdarma (Boronczyk, 2009).

K výše uvedeným třem technologiím je potřeba vysvětlit vzájemnou spolupráci pro kompletní chod webové aplikace, která uchovává nějaká data. Boronczyk (2009) velmi dobře vystihl tuto spolupráci pomocí příkladu s restaurací. Pouhá statická stránka je pro uživatele nedostatečně interaktivní. Uživatel potřebuje mít možnost si vybrat, co chce vidět na stránce. Proto, když si zákazník v restauraci (uživatel stránky) objedná jídlo, tak tuto objednávku oznámí číšník (Apache server) kuchaři (PHP), který pomocí surovin ze skladu (MySQL) toto jídlo připraví, předá zpět číšníkovi a ten jej zákazníkovi naservíruje.

Jak bylo v předešlém odstavci popsáno, databázový systém MySQL je tedy takový objemný sklad informací, který data uchovává a umožňuje k nim přístup. Díky tomu, že se jedná o relační databázový systém, je možné informace rozdělit do skupin (tabulek). Každá tabulka obsahuje pole (sloupce), které obsahují jednotlivé části informace. Tím je myšlena například informace o uživateli, tzn. jméno, příjmení, adresa bydliště atd. Výhodou je, že jednotlivá pole tabulky se mohou lišit typem informace. Typ informace může být text, samotné číslo, ale také strukturované typy, jako například kalendářní datum (Oracle Corporation, 2014).

Aktuální verze serveru MySQL je verze 5.7, která byla zveřejněna 15. 12. 2014. V blízké budoucnosti je však v plánu vydání nové verze 6, která bude zahrnovat mnoho vylepšení. Příkladem jsou cizí klíče i pro jiné typy tabulek, než pouze s mechanismem InnoDB (Oracle Corporation, 2014).

Datové typy

Každé pole v tabulce může mít rozdílný typ informace. To ale neznamená, že typ informace může být v průběhu chodu volně změněn. Typ se databázi zadává při vytváření dané tabulky a v průběhu chodu se může narazit na několik omezení. Pokud tabulka již obsahuje data, která odporují novému typu, není změna možná. Každý datový typ má určitý rozsah hodnot, do kterého je možno přistupovat. Proto je nutné datový typ vybírat s rozvahou, aby byl pro daný účel vhodný a zároveň šetřil paměť.

3.4 Skriptovací programovací jazyk PHP

Počátky skriptovacího jazyka PHP spadají až do roku 1994, kdy sloužil pouze pro jednoduché počítání přístupů na stránky a byl napsán v jazyce Perl. Později však byl přepsán do jazyka C, kvůli menší zátěži serveru. Celým názvem se PHP původně jmenoval „Personal Home Page Tools“. Název byl ale v průběhu času několikrát upraven. V roce 1998 vyšla verze PHP 3.0 a fungovala již na operačních systémech Windows (Zajíc, 2004). Nejaktuálnější verze je PHP 5.6 vydaná 28. 04. 2014.

Jazyk PHP je v dnešní době nejčastěji používaný modul webového serveru Apache, jak již bylo řečeno v kapitole 3.3 (Zajíc, 2004). V základní knihovně je obsaženo několik tisíc funkcí včetně funkcí webového serveru a jazyk samotný využívá také nativních funkcí operačních systémů, na kterých je nainstalován. Další vý-

hodou je, že nativně podporuje většinu databázových systémů díky vestavěné API (The PHP Group, 2014).

Kód v jazyku PHP je plně integrován do kódu v jazyku HTML. Uživatel, který si přečte výstup stránky, nemůže kód PHP vidět, protože se tento jazyk zpracovává na straně serveru. V tomto ohledu je tedy bezpečnější než kód HTML nebo JavaScript, který si uživatel může zobrazit (Boronczyk, 2009).

3.4.1 PHP jako samostatný skript

Velikou výhodou je jednoduchá syntaxe, která se podobá většině programovacích jazyků, jako jsou například C++ nebo Java. Jazyk PHP, stejně jako předešlé dva zmíněné jazyky, podporuje objektově orientované programování, což umožňuje implementaci tříd, objektů, rozhraní a možnost dědičnosti, polymorfizmu a zapouzdření. Ve výsledku tyto prvky zpřehledňují celý kód a také umožňují opětovné použití části kódu (Lecky-Thompson a Nowicki, 2010).

3.4.2 PHP ve spolupráci s databázovým systémem

V kapitole 3.3 je uvedeno, že PHP úzce spolupracuje se systémem MySQL. Funkční skripty probrané v odstavci výše se dají také implementovat do této spolupráce a to nám umožňuje výsledky skriptu ukládat do databáze pro pozdější použití. Pro aplikace je nejčastější volbou datové uložení relační databáze. Je to z toho důvodu, že relační databáze má oproti objektové databázi mnoho výhod, jako například lepší stabilitu, spolehlivost a odolnost, snadnější komunikaci a integritu dat (Lecky-Thompson a Nowicki, 2010).

Pokud je MySQL povoleno v PHP nastavení, na propojení těchto dvou systémů stačí poté několik řádků kódu naimplementovaných ve výsledném PHP souboru (Boronczyk, 2009). Z důvodu, že původní komunikace mezi PHP a MySQL neumožňuje v dnešní době nové funkce MySQL, na komunikaci s databází se začaly používat následující dvě technologie. Abstrakce databáze pomocí PDO a rozšíření MySQLi (MySQL Improved), které umožňují lepší spolupráci než původní funkce pro komunikaci (Moravec, 2010).

PDO (PHP Data Objects)

Protože platformy jako .NET nebo J2EE (Java 2 Enterprise Edition) již mechanismy pro abstrakci databáze obsahují, je pro platformu PHP potřeba také tuto funkci obsáhnout. K tomu slouží právě zmiňované PDO, které pro komunitu přišlo spolu s vydáním platformy PHP 5.2 (Lecky-Thompson a Nowicki, 2010).

Co tedy tato abstrakce databáze přináší? Z důvodu, že oficiální PHP dokumentace upřednostňuje přímou konektivitu od abstrakce a stále mnoho programátorů tuto metodu upřednostňuje, musí se vzít v potaz, že abstrakce databáze všeobecně snižuje počet míst, na kterých se musí implementovat kód pro komunikaci aplikace s databází. Tím se zjednodušuje správa aplikace a zlepšuje přenositelnost kódu (Lecky-Thompson a Nowicki, 2010). PHP Data Objects má mnoho dalších výhod,

keré ale není potřeba zmiňovat, protože tato technologie nebyla v bakalářské práci zahrnuta.

Rozšíření MySQLi

Jedná se o rozšíření určené pro verze MySQL 4.1.3 a novější. Jak už z názvu vyplývá (MySQL Improved), je to pouze vylepšená a optimalizovaná verze klasické komunikace mezi PHP a MySQL, která navíc obsahuje přístup k nově naimplementovaným funkcím a podporuje také objektově orientovaný a procedurální styl programování (Moravec, 2010).

3.4.3 MVC a MVP architektura

Rozdíl mezi kvalitně a nekvalitně naprogramovanou aplikací je právě v její architektuře. Většina začínajících programátorů a těch, kteří si chtějí práci na projektu ulehčit, krok s dobře propracovanou architekturou velmi ráda vynechává, protože to je pro ně práce navíc. Webové aplikace se pak ve výsledku skládají z pouhého jednoho PHP souboru pro stránku, kde jsou dohromady naimplementovány všechny prvky aplikace. To tedy zahrnuje strukturu stránky, vzhled, funkční stránku aplikace a spojení a komunikaci s databází. Mnoho programátorů je s touto strukturou spokojeno, ale neuvědomují si, že to zahrnuje jak absolutní nepřehlednost kódu, tak to nejdůležitější, a to bezpečnost aplikace. Takové stránky poté bývají ve většině případů pro škodlivého uživatele procházka růžovým sadem a nedělá mu problém do takovéto aplikace vniknout a způsobit škodu. V lepším případě pak uživatelé používají, jednoduše řečeno, architekturu MV, kde oddělí alespoň komunikaci s databází od ostatních prvků. Struktura, vzhled a funkcionalita stránky mají své oddělené soubory. I to však není ideální případ a taková aplikace se nedá označit za profesionální.

Proto je třeba používat správnou architekturu. Tou je architektura třívrstvá. Zde se ale objevuje otázka, zda použít architekturu MVC (model-pohled-řadič) nebo MVP (model-pohled-presenter). Uživatelé ze začátku ani nenapadne, že by mohl být nějaký rozdíl mezi řadičem a presenterem. Oba prvky vykonávají velmi podobné funkce. Oba obdrží od pohledu reakci od uživatele, zavolají danou funkci, při které komunikují s modelem a ten vrátí určitá data, která předá pohledu (Grudl, 2009). Tato problematika je velmi zásadní, a proto je nutné v následujících dvou kapitolách podrobněji vysvětlit funkce jednotlivých prvků a hlavně rozdíl mezi řadičem a presenterem.

Jednotlivé součásti MVC a MVP architektury

Pro správné pochopení je potřeba nejprve vysvětlit, co jednotlivé součásti znamenají a jak fungují. Důležitým prvkem je oddělit uživatelské rozhraní (view), komplexní logiku aplikace (model), a vnitřní zpracování a rozhodování (controller nebo presenter). Pokud je potřeba provést změnu v aplikaci, stačí pouze upravit nebo vyměnit jednu část architektury. Dále se, pomocí tohoto oddělení, mnohem jednodušeji zachycují a opravují, které aplikace obsahuje. Proto je výhodnější mít

několik menších komponent, které mezi sebou komunikují, než pár velkých nepřehledných souborů. Tato architektura také zabraňuje duplikování kódu, pokud potřebujeme mít aplikaci funkční na několika různých platformách, a to za pomoci implementace více presenterů a pohledů (view). Modelu už je pak jedno, se kterým řadičem aktuálně komunikuje (Lecky-Thompson a Nowicki, 2010).

Nyní tedy k jednotlivým prvkům. Model zastupuje komplexní logiku celé aplikace. Ta zahrnuje obsluhu nejrůznějších procesů aplikace a komunikaci s databází (získávání, zapisování a editace uchovávaných dat) (Lecky-Thompson a Nowicki, 2010).

Dalším prvkem je pohled, který zastupuje uživatelské rozhraní. Ten je koncovým bodem aplikace. Vykresluje vzhled stránky pro uživatele, prezentuje data, která získá od řadiče a umožňuje uživateli interakci s aplikací (Lecky-Thompson a Nowicki, 2010).

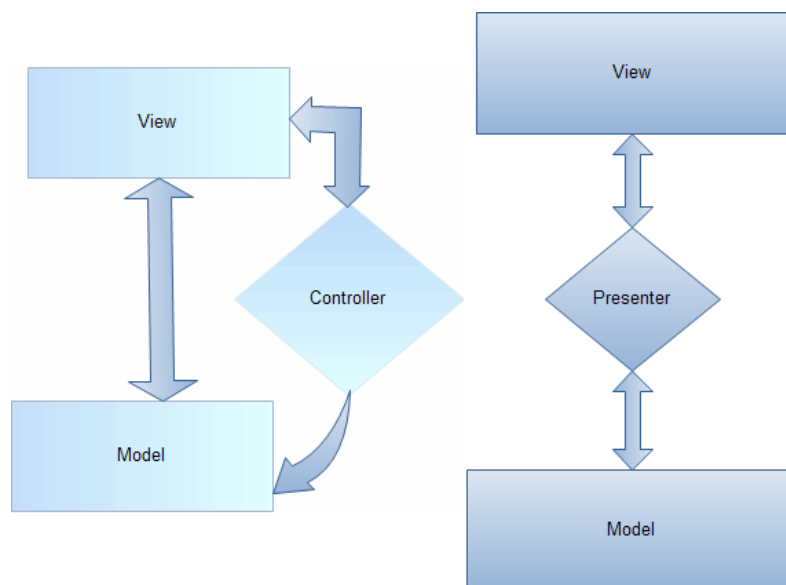
Controller vs. Presenter

Poslední prvek z architektury má dvě možnosti implementace, a to řadič (MVC architektura) a presenter (MVP architektura). Z primitivního hlediska se dá usoudit, že řadič i presenter zaujímají stejné postavení v architektuře a jejich úkoly jsou stejné. Mezi těmito dvěma prvky jsou však rozdíly, které není možné slučovat dohromady.

Řadič neboli controller, patří do starší architektury MVC a jedná se o pouhý doplněk k ostatním dvěma prvkům. V této architektuře je totiž možné zasílat příkazy jak z pohledu přes řadič, tak z pohledu přímo do modelu. V prvním případě má řadič stejné funkce jako presenter v architektuře MVP (Emmatty, 2011).

Z důvodu lepší bezpečnosti, a aby byla eliminována možnost přímé komunikace pohledu s modelem, byla vyvinuta novější architektura MVP obsahující presenter, který stojí přímo mezi pohledem a modelem a zamezuje tak možnosti přímé komunikace těchto dvou prvků. Presenter je tedy takový překladatel, který pohledu a modelu zpracovává informace do jejich požadovaného jazyka, zasílá příkazy modelu a servíruje data pohledu (Lecky-Thompson a Nowicki, 2010). Dalším drobným rozdílem presenteru s řadičem je návaznost pouze na jeden pohled. Řadič totiž může obsluhovat pohledů více.

Na následujícím obrázku je zobrazen rozdíl struktury architektury MVC a MVP.



Obr. 1 Struktura architektur MVC a MVP
Zdroj: Emmatty, 2011

3.5 ER Diagram a jazyk UML

ER Diagram (Entity-Relationship Diagram) primárně graficky reprezentuje logickou strukturu databáze aplikace a je používán ve strukturovaných analýzách a konceptuálním modelování. Je složen ze tří komponent: entity – reprezentující objekty, osoby, akce aj., které je potřeba uchovávat. Dále atributy, což jsou vlastnosti entit a v poslední řadě vztahy mezi jednotlivými entitami (DifferenceBetween.info, 2014).

UML na druhou stranu řeší celkový projekt se vším, co program obsahuje, tedy jednotlivé objekty a také entity, které s programem přicházejí do kontaktu. Je navržen tak, aby jeho univerzální notaci rozuměli všichni, kteří na projektu pracují a nedošlo tak k nedorozumění v některé z částí projektu (Lecky-Thompson a Nowicki, 2010). Jeho hlavní funkcí je znázornit všechny možné entity, které přicházejí do styku s aplikací a jaké funkce aplikace mohou využívat.

3.6 MVC a MVP frameworky pro vývoj webových aplikací

Frameworky samy o sobě mají hlavní funkci zjednodušit práci při tvorbě nejrůznějších aplikací. Nejlepší využití je u rozsáhlých projektů, které potřebují nespočetné množství funkcí, protože frameworky mají ve většině případů tyto funkce již předem naprogramované. Programátoři je ale také používají na jednoduché aplikace, protože i když framework sám o sobě objektivně nepotřebují, nabízí jim stále spoustu zjednodušení a vylepšení. Dále zde stále mají připravenou správnou architekturu aplikace a nemůže se jim stát, že by sešli ze správné cesty. Fra-

mework je, dá se říct, předpřipravený kód, který nutí dělat věci správně (Lecky-Thompson a Nowicki, 2010).

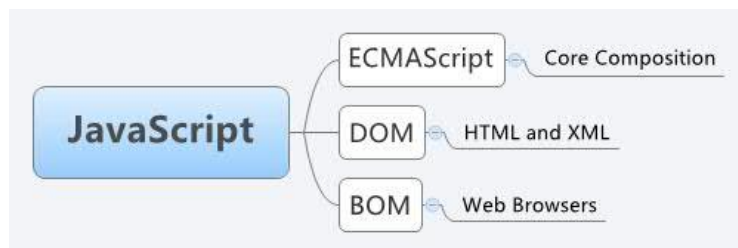
Frameworky pro PHP jsou obvykle založeny na třívrstvé architektuře, což primárně dodává lepší zabezpečení celého systému a přehlednost kódu. Těchto frameworků je v současnosti mnoho. Můžeme jmenovat například frameworky Yii, CodeIgniter, CakePHP nebo český Nette.

3.7 Programovací jazyk JavaScript

V roce 1995, kdy se programovací jazyk JavaScript objevil na veřejnosti, bylo jeho hlavním účelem ověřování údajů zadaných uživateli bez toho, aby nedocházelo k ovlivnění funkcionalit serveru. V dnešní době je programátor schopný téměř všechny funkční možnosti aplikace programovat pomocí JavaScriptu. Stal se tak nepostradatelnou součástí každé webové aplikace a je podporován všemi prohlížeči, včetně těch na mobilních zařízeních (Zakas, 2009).

JavaScript tvoří následující tři části (viz Obr. 2) (Zakas, 2009):

- jádro (ECMAScript),
- objektový model dokumentu (DOM – Document Object Model),
- objektový model prohlížeče (BOM – Browser Object Model).



Obr. 2 Součásti jazyka JavaScript
Zdroj: Zheng, 2011

První součást ECMAScript se často zaměňuje s termínem JavaScript. To však není správně, neboť ECMAScript je jazyk, který není propojen s webovými prohlížeči a bere se pouze jako základ (implementace funkčních prvků), na kterém stavějí robustnější skriptovací jazyky (Majda, 2009). To je přesně případ JavaScriptu. Další součástí DOM (Document Object Model) tvoří aplikační programovací rozhraní (API) pro jazyk XML, který byl rozšířen pro použití v jazyku HTML. Stručně řečeno, zahrnuje mapování celé struktury stránky do uzlů. Poslední součástí jazyka JavaScript je objektový model prohlížeče (BOM – Browser Object Model), který umožňuje pracovat s oknem prohlížeče (Zakas, 2009).

Objektové modely se u jednotlivých prohlížečů liší. Postupem času se proto začaly vyvíjet knihovny pro JavaScript, které tento problém řeší a zároveň zjednodušují přístupy ke komplexním funkcím. V dnešní době již existuje mnoho knihoven, které se mohou rozdělit do dvou kategorií: všeobecné a specializované (Zakas, 2009).

Všeobecné knihovny fungují, velmi jednoduše řečeno, jako náhrada za samotný JavaScript, tudíž mohou být použity jako základ pro webovou aplikaci. Stále se jedná o JavaScript, avšak s tím rozdílem, že je použita jiná syntaxe jazyka. Všeobecné knihovny zahrnují základní funkce JavaScriptu a některé funkce ze specializovaných knihoven (Zakas, 2009). Jako příklady všeobecných knihoven je nutné zmínit knihovnu jQuery, která je velmi oblíbená a velmi využívaná i velikány, jako jsou firmy Google, Microsoft nebo IBM. Dalšími knihovnami jsou například knihovny Prototype nebo MooTools (Refsnes Data, 2015).

Specializované knihovny se dělí do typů podle jejich účelu a dále, zda jsou nebo nejsou kompatibilní. Typů knihoven je mnoho a je těžké vyjmenovat všechny. Mezi ty nejdůležitější a nejpoužívanější patří například knihovny pro tvorbu uživatelského rozhraní a knihovny pro internetové aplikace, jako je například Rico nebo qooxdoo ulehčující vývoj aplikace natolik, že obsahují celé koncepční rámce webové aplikace a drobné funkce. Také existují specializované kryptografické knihovny pro zabezpečení komunikace (Zakas, 2009). Tyto knihovny se však mohou stát velice rychle zastaralými, protože postupem času dochází k prolomení dané kryptovací technologie. Názorným příkladem je algoritmus MD5. Dalším typem specializovaných knihoven mohou být v neposlední řadě knihovny pro tvorbu her, animací a efektů. Patří sem knihovny Lightbox, Canvas 3D JS Library (C3DL), Pixastic a mnoho dalších (Xul.fr, 2014).

3.7.1 Asynchronní zpracování webových stránek technologií AJAX

V obsahu použitých technologií je také potřeba zmínit techniku AJAX, která posouvá aplikaci na další úroveň. Jedná se o asynchronní zpracování dat, což ve zkratce znamená změnu obsahu aktuálně zobrazené stránky nebo změnu v databázi bez nutnosti uvolňovat webovou stránku. Uživatel tak může zasílat požadavky serveru, aniž by musel obnovovat stránku pro nová data (Refsnes Data, 2015).

Klíčovou technologií, která umožňuje AJAXu tyto možnosti, je objekt XMLHttpRequest (XHR), vyvinutý společností Microsoft, který se později stal webovým standardem pro všechny prohlížeče. Tento objekt zavedl propracované rozhraní pro provádění serverových požadavků a vyhodnocování odpovědí. Díky tomu lze ze serveru získat asynchronně načtené dodatečné informace a poté je vložit do stránky pomocí modelu DOM (Zakas, 2009).

JSON

Technologie AJAX pracuje s jazykem XML. Práce s jazykem XML se však v jednotlivých prohlížečích liší a extrahování dat ze struktury je zbytečně komplikované. Odpověď na tyto problémy přinesl Douglas Crockford, který představil datový formát používaný pro značení objektů a polí dat s názvem JavaScript Object Notation (JSON). Formát JSON je založen na podmnožině syntaxe JavaScriptu a díky němu je možné vytvářet strukturovaný datový formát schopný reprezentovat stejné typy dat jako jazyk XML, například kolekce dvojic název-hodnota, který reprezentuje objekt JavaScriptu obsahující pojmenované vlastnosti (Zakas, 2009).

Zde je uveden příklad:

```
{  
  jméno: "Jan",  
  příjmení: "Novák",  
  věk: 23  
}
```

3.8 Zpracování fotek specializovaným softwarem

Pořízené fotografie byly zpracovávány do panoramatického zobrazení Ing. Jiřím Lýskem Ph.D., který software naprogramoval. Software vytváří fotografie pomocí technologie válcového (cylindrického) panorama. To funguje tak, že jednotlivé pořízené fotografie jsou nanášeny na válec a každá fotografie vyžaduje určité procento překrytí s předešlou fotografií (Srna, 2010). Vyhledávání významných společných bodů je řešeno metodou SURF a samotné spojování fotografií je realizováno pomocí algoritmu RANSAC. Uživatel ve finále prohlíží panorama, jako kdyby stál ve středu válce.

4 Metodika práce

Při vývoji webové aplikace je pro programátora velmi důležité se správně rozhodnout, jaké technologie použije. V následujících podkapitolách jsou řešena tato rozhodování.

4.1 Analýza a výběr daného frameworku

Pro programovací jazyk PHP je frameworků v dnešní době nespočetné množství. Z toho důvodu budou v této kapitole rozebrány pouze ty nejznámější a nejlepší frameworky, které byly zmíněny v kapitole 3.6. Jedná se o frameworky Yii, CodeIgniter, CakePHP a v poslední řadě framework Nette, který ve světě není moc známý, ale pro českou komunitu velice oblíbený. Všechny tyto frameworky splňují následující požadavky: kompatibilita s poslední verzí jazyka PHP, dostatečná rozsáhlost pro komplexní vývoj webové aplikace, nikoliv však obsažení pouze sady skriptů (knihoven) a založenost na architektuře MVC nebo MVP (Škrášek, 2008).

Rozhodnutí, jaký z nich použít, jsou velmi individuální. Každý uživatel má jiné preference a vyhovuje mu jiná struktura a postup práce. Jediné rozdíly, které je třeba zmínit, jsou, že framework CodeIgniter nepoužívá systém šablon a frameworky Nette a Yii mají oproti ostatním zmíněným vyšší systém zabezpečení. Ostatní parametry nebudou v této práci rozebrány, protože každý framework má své výhody a nevýhody a disponuje jinými prvky než jejich konkurenti. Hlavní a rozhodující výhodou, proč byl ve finále použit v práci framework Nette, je podpora v českém jazyce a v naší zemi je velice populární a firmami žádaný (Jakoubě, 2013).

4.2 Analýza a výběr dané databáze

Pokud nejsou brány v potaz programátorovy osobní preference, při vytváření nové dynamické webové aplikace je nutné řešit otázku, jaký databázový systém bude v aplikaci použit. Jazyk PHP podporuje několik desítek různých relačních databázových systémů. Těch, o kterých se však má cenu zmiňovat, je pouze pět. Jedná se o systémy MySQL, PostgreSQL, Oracle, Microsoft SQL Server a IBM DB2 (Lecky-Thompson a Nowicki, 2010). Za zmínku stojí také nový typ databázových systémů, který se stává poslední dobou oblíbený, a to je souborová databáze, jako například SQLite. Výhodou tohoto typu databáze je, že není nutné využívat server k ukládání dat. Vše si ukládá do jediného obyčejného souboru, který je možné přesouvat a kopírovat skrz jakoukoliv platformu operačních systémů (About SQLite, 2015). Tento typ se ale používá převážně pro desktopové aplikace z důvodu, že při přístupu více klientů do totožné databáze najednou, dochází ke komplikacím (Tezer, 2014).

Z ekonomického a marketingového hlediska vítězí společnost Oracle, neboť vlastní jak databázový systém Oracle, který je velmi využíván, tak nejpoblárnější

systém MySQL (Fontecchio, 2012). Databázový systém společnosti IBM poněkud zaostávají v popularitě oproti jejím konkurentům, a systém MS SQL Server se využívá spíše pro programovací jazyk .NET a jeho plná verze je placená. Z tohoto důvodu se výběr zužuje na poslední tři zmiňované kandidáty od společnosti Oracle Corporation a The PostgreSQL Global Development Group. I přesto, že se firma PostgreSQL jako ekonomický subjekt nemůže s gigantem Oracle srovnávat, je z hlediska databázového systému pro ostatní dva velkým konkurentem.

Nyní už k jednotlivým třem vybraným adeptům. Oracle je bez pochyby nejlepším systémem z vyjmenovaných tří. Je tomu tak ale pouze v případě, že se jedná o placenou verzi. Placená verze totiž obsahuje mnohem více vylepšení, největší podporu operačních systémů a výbornou systémovou podporu. Tedy mnohem více, než systémy MySQL a PostgreSQL. Na druhou stranu je však cena za placenou verzi pro většinu programátorů a subjektů nepřijatelná nebo zbytečná a práce se systémem a správou databáze je mnohem komplikovanější, než u zbylých dvou konkurentů. Neplacená verze Oraculu je velmi omezená a ve výsledku pak slabší než ostatní dva systémy.

Další konkurent, PostgreSQL, vyvinutý stejnojmennou firmou, má výhodu oproti systému MySQL v tom, že byl od počátku vývoje zaměřen výhradně jako relační databáze. MySQL byl totiž původně vytvářen k jiným cílům a až postupem času se přetvořil v relační databázi. To také s sebou nese své následky v tom smyslu, že oproti PostgreSQL není MySQL tak dobře propracovaná relační databáze a obsahuje místy integritní problémy. PostgreSQL je tak ve výsledku výkonnější a funguje na více platformách (The Regents of the University of California, 2012).

Přes všechny tyto klady PostgreSQL oproti MySQL by se zdálo, že PostgreSQL bude více upřednostňovaný systém. Opak je však pravdou. Pro většinu uživatelů je velmi důležitá jednoduchá práce se systémem a dostupnost různých nástrojů (MySQL Workbench, phpMyAdmin atd.). Těmito body MySQL disponuje na výbornou a je proto mnohem více používaným systémem než PostgreSQL. Důkazem jsou populární internetové stránky a aplikace, které systém MySQL využívá (Facebook, YouTube, Google, Wikipedia, Twitter a další) a navíc také velmi oblíbené redakční systémy WordPress a Joomla (Ehinger, 2014). I přes lehkou výkonnostní výhodu systému PostgreSQL, byl v bakalářské práci použit jeho populárnější a používanější konkurent MySQL.

Na závěr souhrn tří porovnávaných databázových systémů a jejich podpory:

Tabulka 1 Souhrn podpor tří porovnávaných databázových systémů

Název systému	Oracle	PostgreSQL	MySQL
Licence	Komerční (omezená verze zdarma)	Open Source	Open Source
Podporované operační systémy	AIX HP-UX Linux OS X Solaris Windows z/OS	FreeBSD HP-UX Linux NetBSD OpenBSD OS X Solaris Unix Windows	FreeBSD Linux OS X Solaris Windows
Podporované programovací jazyky	C C# C++ Groovy Java JavaScript Objective C Perl PHP Python Ruby Tcl Visual Basic a další	.Net C C++ Java Perl Python Tcl	C C# C++ D Java Objective C Perl PHP Python Ruby Scheme Tcl a další

Zdroj: Solid IT dmbh, 2015

4.3 Analýza a výběr dané knihovny JavaScriptu

JavaScriptová složka je pro propracovanou a funkční webovou aplikaci nezbytná. V samotném JavaScriptu je také možné bez problému programovat. Mnohem výhodnější je však využít některou ze všeobecných knihoven, které nabízí mnoho výhod a zjednodušení. Na výběr jich je však několik. Proto se musí řešit otázka, kterou z nich použít. Nutným kritériem je dostatečná komunitní podpora, jednoduchá práce s knihovnou a otevřený zdrojový kód, aby bylo použití zdarma. Tato kritéria výběr poněkud zužují.

První možnou knihovnou je MooTools. Tato knihovna s otevřeným zdrojovým kódem je navržena tak, aby byla maximálně jednoduchá a nenáročná na výkon serveru. K základním objektům JavaScriptu navíc doplňuje nové metody (Zakas, 2009).

Pravděpodobně druhou nejčastěji využívanou knihovnou JavaScriptu je Prototype. Ta je řízená třídami nabízejícími běžné i složité funkce, které jsou zapouzdřeny do jednoduchých aplikačních rozhraní pro činnost webové stránky (Zakas, 2009).

Nejznámější a nejpoužívanější knihovnou, která v současnosti existuje, je však knihovna jQuery. Její intuitivní syntaxe, která využívá elementy jazyka CSS, velmi zjednodušuje práci s funkcemi. Na rozdíl od jiných knihoven, práce s jQuery je založena na stylu implementace, kdy programátor říká jednotlivým složkám stránky (pomocí elementů modelu DOM), jak se mají chovat (Zakas, 2009).

Tímto však není myšleno, že by byla knihovna jQuery nejlepší. Každá knihovna má zajisté své výhody i nevýhody. Proto je použitelná pouze pro určitý typ úloh. jQuery je pravděpodobně určena pro uživatele, kteří raději pracují s prvky HTML, než s objekty a jejich vlastnostmi. Z uvedených důvodů byla také použita v bakalářské práci.

Knihovna jQuery

Přesto, že zvyknout si na styl zápisu jQuery oproti klasickému jazyku JavaScript je ze začátku poněkud složitější. Po několika odpracovaných projektech však uživatel pochopí, že zápis pomocí elementů modelu DOM je velmi intuitivní a praktický. Knihovna klade důraz na řetězení, což pro programátora znamená postupné volání metod, které zužují výběr požadovaných elementů, a těm následně přikazuje, co vybrané elementy mají dělat (jQuery - Kuchařka programátora, 2010).

Použití knihovny je volně k dispozici (open source), nijak nezatěžuje server svou malou velikostí a stejně, jako ostatní knihovny řeší rozdíly mezi jednotlivými webovými prohlížeči, takže se o tento problém nemusíme starat. Hlavní výhodou je však její rozsáhlá podpora a komunita uživatelů. Díky tomu se neustále rozšiřuje databáze zásuvných modulů (pluginů), které uživatelé mohou volně tvořit a zveřejňovat (jQuery - Kuchařka programátora, 2010). Zásuvných modulů je pro knihovnu jQuery nespočetné množství a nemá smysl je zde popisovat a ani také vyjmenovávat typy těchto modulů, protože na co si programátor vzpomene, že by potřeboval udělat, na to je pravděpodobně v knihovně jQuery již někým vytvořený plugin. V následujících dvou odstavcích jsou stručně popsány dva zásuvné moduly, které byly v bakalářské práci použity pro jednoduchou aplikaci interaktivní Google mapy a vytvoření panoramatického ovládání pro fotky.

Gmap3

V aplikaci je potřeba využívat interaktivní mapu pro zobrazení školního arboreta. Pro tento prvek lze použít například rozšíření Google maps nebo mapy společnosti Seznam.cz. Výhodou Google maps je v první řadě propracovanější technologie a celosvětová podpora komunitou. Další a rozhodující výhodou je, že Google mapy

mají komunitou vytvořené rozšíření, které umožňují implementaci map přímo v knihovně jQuery. Samotná technologie Google maps nebo Seznam map tuto možnost nemá.

Jako nejvhodnější nastavba pro jQuery byl vybrán plugin Gmap3, který umožňuje jednoduchou implementací všechny potřebné funkce pro tuto výslednou aplikaci (Demonte, Gerits a Lesire, 2010).

Panorama Viewer

Pro panoramatické ovládání fotek v aplikaci bylo použito rozšíření Panorama Viewer pro knihovnu jQuery, vytvořené uživatelem *peachananr*. Toto rozšíření je kompatibilní ve všech prohlížečích a na všech zařízeních (Rojwongsuriya, 2010).

4.4 Postup návrhu aplikace

Důležitým prvkem kvalitní aplikace je předem navržený plán, který je možné v průběhu vývoje měnit. Předpřipravený návrh aplikace zabrání chaotickým krokům a vývojář má tak jasně navrženo, jak bude postupovat s implementacemi. Prvním krokem je rozhodnout, jaký framework a databázový systém se použije. Tento krok byl již vyřešen v předešlých kapitolách.

4.4.1 Vývojové prostředí

K otázce výběru vývojového prostředí má každý programátor jinou odpověď. Je to výhradně individuální rozhodnutí z důvodu zvyku a přívětivosti uživatelského rozhraní. Velmi oblíbeným vývojovým prostředím je PSPad Editor. Ten je však využíván převážně začátečníky, protože pro pokročilé uživatele neobsahuje našeptávače a další velmi užitečné funkce. Pokročilých vývojových prostředí, která podporují práci s nejpoužívanějšími programovacími jazyky, je mnoho, a proto nemá smysl argumentovat výhody a nevýhody těchto aplikací.

V bakalářské práci byla použita open source aplikace NetBeans IDE, která podporuje vývoj všech potřebných programovacích jazyků použitých v práci a zároveň obsahuje dvě velmi praktické funkce, a to našeptávač a kontrola chyb v kódu (Oracle Corporation, 2015).

4.4.2 Struktura aplikace

Dalším, velmi důležitým, bodem je návrh struktury celé aplikace. Vzhledem k tomu, že se jedná o prezentační aplikaci pro veřejnost, je nutné bezpečně oddělit prezentační část od části administrační. Tento problém řeší šifrované přihlašování, které předejde průniku neautorizovaného uživatele do administrační části.

Prezentační část

Prezentační část bude rozdělena do dvou oddílů. V první části bude pro návštěvníka spuštěna mapa školního arboreta, na které si bude moci zobrazit jednotlivé

jízdy robota a podrobněji vidět jednotlivé panoramatické fotografie včetně bližších informací z určitého místa focení. Druhá část bude obsahovat galerii fotografií, kde si návštěvník bude moci přehledně prohlédnout všechny snímky pohromadě spolu s podrobnými informacemi. Tyto stránky musí být interaktivní, a co nejpřehlednější, protože jde o veřejnou část, kterou bude používat většina návštěvníků stránek.

Jelikož se jedná o aplikaci prezentující velké množství fotografií, je zde kritické celkové využití paměti načítání stránek. Proto je v aplikaci nutné naimplementovat tzv. generování náhledů pro fotografie, které se v mnoha případech v aplikaci zobrazují pouze jako miniatury a nikoliv v originální velikosti a kvalitě. Mnoho programátorů si tuto problematiku neuvědomuje a nechávají ve svých aplikacích načítat pouze vzhledově zmenšené fotografie a obrázky bez paměťového zmenšení, což je neúnosné množství dat při každém načítání stránky, i když to ve skutečnosti není vůbec nutné. Aby se ušetřilo zatížení serveru a také drahocenný čas uživatelů, je potřeba vytvořit poněkud komplikovaný PHP skript, přes který se bude ve výsledku každá fotografie zobrazovat. Aplikace skriptu pošle zdroj požadované fotografie a skript pomocí několika funkcí fotografii zmenší jak vzhledově, tak hlavně paměťově a uloží ji do paměti cache. Načítaná fotografie má ve výsledku pouze několik kB oproti původním několika MB a při opětovném načítání stejné fotografie s danou velikostí už ani skript novou fotografii nemusí generovat, ale přímo načítá fotografii uloženou v této paměti. Cache musí být možné také jednoduše promazat jediným kliknutím v administraci například z důvodu, že uživatel nahraje opravenou verzi již uložené fotografie.

Administrační část

Administrační část už nemusí být tolik intuitivní jako část prezentační. Musí však obsahovat všechny potřebné funkce pro vytváření a editaci jízd a vytváření a editaci administrátorů. Administrátoři budou rozděleni do adminů a super-adminů s tím rozdílem, že běžný admin bude mít přístup pouze k vytváření a editaci jízd. Do části správy administrátorů bude mít přístup pouze super-admin. Tím se opět zvýší zabezpečení správy aplikace.

Super-admin v části správy administrátorů bude moci vytvořit nového uživatele a bude mít přístup k seznamu všech administrátorů s možností editace a smazání.

Administrace jízd bude naopak komplikovaná. Aby pověřená osoba mohla vložit novou jízdu, je nutné nahrát sérii fotografií a ke každé doplnit příslušné informace. To se neobejde bez importování několika souborů. Pro jednoduchost se do importu jízdy zavede možnost vložení komprimovaného souboru typu zip, který bude mít danou strukturu obsahující fotografie a textový soubor s informacemi k jednotlivým fotografiím. Tato jednodušší verze importu má však komplikaci. Server, na kterém bude aplikace umístěna, má omezení ve velikosti paměti pro nahrávání souborů. Z toho důvodu bude moci administrátor nahrát jízdy pouze do dané velikosti výsledného komprimovaného archivu. Administrátor se však nevyhne situaci, kdy bude potřebovat vložit soubor s větší

velikostí paměti. Proto je nutné zavést druhou verzi importu jízd. Tato bude fungovat na způsobu ručního nahrání fotek na FTP server do dané složky a poté jednoduchého vložení data jízdy a textového souboru s informacemi. K oběma těmto způsobům musí být připravena nápověda pro správné vkládání souborů.

Editace jízd musí být také naimplementována. Pokud dojde ke zjištění chybné informace v existující jízdě, musí mít administrátor možnost změnit danou informaci pomocí pár kliků. To bude fungovat na buňkovém systému MS Excelu bez nutnosti potvrzení uložení za pomoci technologie AJAX.

Posledním prvkem v administrační části bude jednoduchá statistika počtu jízd a fotografií a dále logování všech činností administrátorů do databáze pro zajištění bezpečnostní kontroly. Pro větší bezpečnost bude také mazání administrátorů a jízd fungovat na způsobu změny jednoho atributu v záznamu tak, že se pro aplikaci záznam zneviditelní a nikoliv smaže, čímž by došlo k nenávratné ztrátě této informace. V některých případech je však nutné i celkové smazání a to z bezpečnostních důvodů může provést pouze super-admin.

V příloze A je znázorněn use-case diagram aplikace.

4.4.3 Struktura databáze

Do databáze bude potřeba ukládat následující záznamy: jízdy, uživatele a provedené činnosti administrátorů.

U jízd je z důvodu správné struktury potřeba rozdělit jízdy a jednotlivé pozice jízdy. To zahrnuje mít obsaženy dvě tabulky propojené mezi sebou. U jízd je potřeba zaznamenávat pouze datum jízdy a nepovinný popis. Jednotlivé pozice musí naopak obsahovat informací několik. Tou nejdůležitější je poloha GPS, podle které se zobrazí umístění na mapě, dále číslo fotografie pro přiřazení k pozici a nepovinný čas pořízení, nadmořská výška a popis.

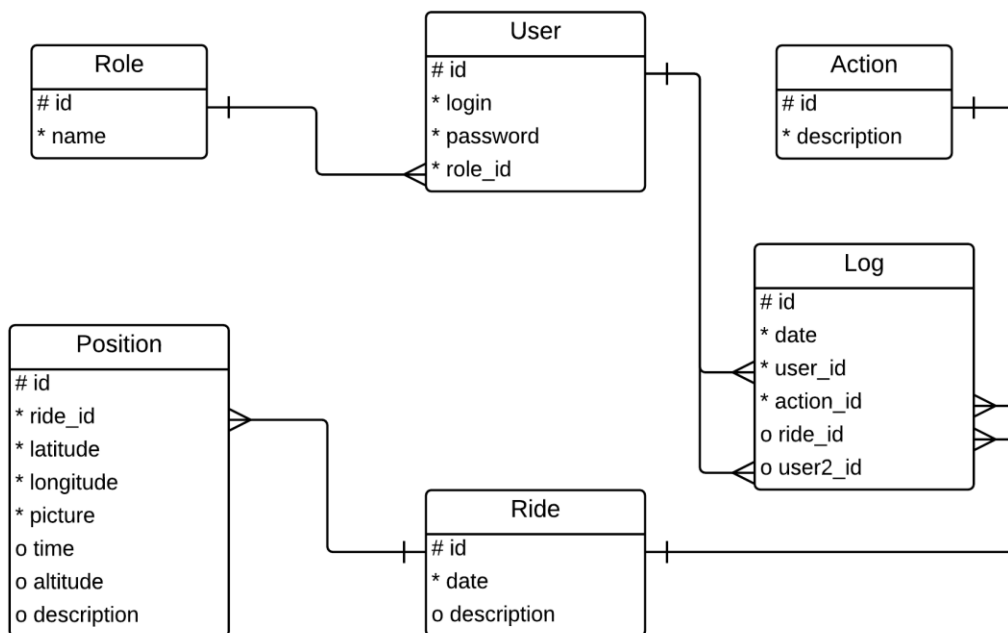
Uživatelé musí být také rozděleni do dvou tabulek a to na uživatele a jejich role. V první je nutné mít zaznamenáno přihlašovací jméno, heslo a přiřazení role. Tabulka role bude obsahovat pouze název role.

Poslední částí databáze je zaznamenávání provedených činností administrátorů. Z důvodu zachování jasně určených činností je tato část stejně tak, jako předšlé dvě části, rozdělena do dvou tabulek. První tabulka zaznamenává přesný typ činnosti, tedy například vytvoření jízdy nebo smazání uživatele, a druhá zaznamenává datum a čas provedení činnosti, přiřazení k uživateli, který činnost provedl, přiřazení k typu činnosti, co uživatel provedl a v poslední řadě přiřazení k jízdě nebo uživateli, tedy co bylo editováno.

Těchto 6 tabulek naprosto stačí ke kompletní funkčnosti aplikace. Na závěr uvedení struktury ER Diagramu:

Arboretum View

Entity Relationship Diagram



Obr. 3 ER Diagram aplikace

4.4.4 Vzhled aplikace

V poslední řadě je otázkou, jaký vzhled neboli design v aplikaci použít. Zde jsou hned dvě možnosti. Tou první je vlastní navržený design. U této možnosti je výhodou přesně navržený vzhled podle všech vlastních požadovaných kritérií. Na druhou stranu, aby byl design na vysoké úrovni, tato možnost vyžaduje perfektní znalost jazyka CSS a nespočet strávených hodin implementací. Proto ve většině případů bývá výhodnější si hotovou šablonu s designem koupit nebo volně dostupnou šablonu stáhnout. Samozřejmě tyto dvě možnosti většinou nesplní všechny představy uživatele a u třetí varianty volně dostupné šablony není zpracovanost zdaleka tak kvalitní jako u šablony koupené. Pro bakalářskou práci jsou však tyto dvě možnosti naprosto dostačující.

V bakalářské práci byla použita volně dostupná šablona s názvem Photography vytvořená společností Dot Com Infoway (Dot Com Infoway, 2015), která byla následně upravena podle požadavků aplikace.

Dalším kritériem je implementace responsivního designu. Aplikace musí být navržena tak, aby se správně zobrazovala na jakémkoliv zařízení s jakýmkoliv rozlišením, a také při změně velikosti aktuálně otevřeného okna prohlížeče musí aplikace na tuto změnu ihned reagovat.

4.5 Pořízení a zpracování panoramatických fotografií

Z toho důvodu, že projektový robot, který má do aplikace pořizovat panoramatické fotografie školního arboreta, není zcela kompletní, byly do práce zahrnuty fotografie vytvořené vlastní činností, které byly následně zpracovány ve specializovaném softwaru pro vytvoření panoramatického zobrazení.

Fotografie byly pořizovány v průběhu roků 2014 a 2015 pro znázornění působení proměnlivého počasí na krajinu. Tento krok je určen pouze pro bakalářskou práci. Fotografie budou následně nahrazeny, jakmile bude autonomní robot schopen samostatně pořizovat vlastní snímky. Pro prozatímní pořizování byl použit fotoaparát Canon EOS 650D a stativ značky Hama.

5 Výsledná aplikace

Výsledná aplikace se dělí na dvě části. Prezentační (veřejná) část je určena pro zobrazování výsledných panoramatických fotografií interaktivním způsobem a je přístupná pro všechny návštěvníky aplikace. Administrační část je naopak přístupná pouze osobám pověřeným. Je zde zahrnuta správa jízd a uživatelů.

Navigační menu pro jednoduchý přístup na všechny dostupné stránky daného uživatele, je umístěno v horním horizontálním pruhu spolu s názvem aplikace tak, aby bylo přehledně odděleno od zobrazeného obsahu. Menu má dvě úrovně pro prezentační a administrační část.

5.1 Prezentační část

Prezentační část, zahrnující hlavní podstatu aplikace, tedy prezentaci výsledných fotografií, je rozdělena na dvě stránky. Ta první obsahuje interaktivní mapu, na které si uživatel zobrazuje pozice, kde byly fotografie vytvořeny a následně si je může zobrazit. Druhá stránka neobsahuje mapu, ale celkový souhrn panoramatických fotografií pro jednoduché porovnávání jednotlivých jízd.

5.1.1 Interaktivní mapa školního arboreta

Interaktivní mapa (viz příloha B) je zároveň počáteční stránkou aplikace. Uživateli se zde zobrazí prázdná mapa arboreta s klasickými nastaveními Google map a seznam existujících jízd. Teprve po výběru některé z jízd se na mapě zobrazí body pozic tak, jako na obrázku v příloze B. Pokud vybraná jízda obsahuje popis, zobrazí se tento popis po kliknutí na tlačítko jízdy přímo pod tlačítkem. Body jsou propojeny čarami, aby byl zřetelně vidět začátek a konec jízdy robota. Čáry mezi body jsou však pouze orientační pro postup mezi pozicemi a nejsou fixovány na přesný pohyb robota po cestách. Následně si uživatel může zobrazit panoramatickou fotografii vybrané pozice kliknutím na příslušný bod.

Po výběru dané pozice v jízdě se uživateli zobrazí prezentace panoramatické fotografie této pozice (viz příloha C). S fotografií je možno volně otáčet pomocí tahu myši a pod fotografií jsou zobrazeny informace o pozici, jako například čas pořízení fotografie, nadmořská výška pozice nebo popis místa. Ve spodní části pod informacemi se nachází také seznam snímků všech pozic z dané jízdy, mezi kterými může uživatel volně přecházet pro zobrazení velkého náhledu. Aktuálně vybraná fotografie je v seznamu zvýrazněna bílým ohraničením. Pro pohyb mezi jednotlivými pozicemi může uživatel také využít šipky nahoru a dolů na klávesnici nebo šipky znázorněné ve spodní části velkého náhledu fotografie. Kliknutím kamkoliv do tmavé okolní části aplikace se prezentace fotografie zavře a uživatel se tak vrátí zpět na stránku s mapou.

Tlačítko „plus“ v horní pravé části zobrazené fotografie (viz příloha C) má funkci zobrazit seznam stejných pozic (do vzdálenosti 5 m) aktuálně zobrazené fotografie z jiných období v pravé části aplikace (viz příloha D). Uživatel

má tak možnost jednoduše srovnávat, jak arboretum vypadalo v různých měsících ze stejné pozice. Pokud uživatel vybere některou z fotografií ze seznamu podobných pozic, zobrazí se mu pouze samostatná otočná fotografie k prohlédnutí (viz příloha E). Kliknutím na tlačítko „x“ v pravé horní části fotografie nebo kamkoliv do tmavé okolní části, se fotografie zavře a uživatel je navrácen zpět k původní prezentaci pozice. Tlačítkem „mínus“, které je na stejné pozici, jako bylo původně tlačítko „plus“, uživatel může seznam podobných pozic opět skrýt.

5.1.2 Galerie jízd

Galerie jízd je souhrnnější alternativa k interaktivní mapě (viz příloha F). Jsou zde zobrazeny všechny existující jízdy pod sebou. Pro ušetření paměti a lepší přehlednost je u každé jízdy zobrazena pouze první pozice. Pokud si uživatel vybere určitou jízdu, u které chce zobrazit přehledně všechny fotografie pozic, stačí mu kliknout na černý, podtržený text „zobrazit“ u daného data jízdy. Všechny fotografie z této jízdy se poté načtou pod sebou. Na jakoukoliv je možno kliknout, čímž dojde k zobrazení prezentace fotografie s ostatními možnostmi tak, jako je to u interaktivní mapy (viz příloha C).

5.2 Administrační část

Do této části je uživatel vpuštěn pouze, pokud je autorizován pomocí přihlašovacího jména a hesla zašifrovaného hashovací funkcí SHA-512. Administrační část obsahuje tři generální části – souhrn činností, správa jízd a správa uživatelů.

5.2.1 Stránka statistik a zaznamenaných činností uživatelů

Stránka zaznamenaných činností uživatelů slouží pouze jako informativní tabule obsahující jednoduchou statistiku (celkový počet jízd a pozic v databázi) a již řečené zaznamenané činnosti administrátorů, kde je přesně evidované časové razítko (viz příloha G). U statistik je také informace o počtu vygenerovaných fotografií ve složce cache a možnost jejího promazání. Aby se zabránilo zahlcení databáze velkým množstvím záznamů činností, uchovávají se pouze data se stářím do dvou měsíců. Akce promazání starých dat je provedena při načtení této stránky.

5.2.2 Přidání jízdy

Přidání nové jízdy (viz příloha H) je velmi komplikovaný proces a z důvodu praktičnosti je novou jízdu možno vytvořit dvěma způsoby. Vzhledem ke skutečnosti, že webový server, na kterém je uložena aplikace, má proměnlivou velikost pro nahrávání souborů, je nutné disponovat naimplementovanou alternativou ručního nahrávání fotografií přímo na FTP server.

První možnost (horní část) vytvoření nové jízdy není ovlivněna nastavením velikosti nahrávaných souborů na server. Postup pro vytvoření nové jízdy je následující:

- uživatel si vytvoří na svém disku složku pojmenovanou podle data jízdy (ve formátu YYYY-MM-DD), která bude obsahovat všechny fotografie dané jízdy. Fotografie budou jednoduše pojmenovány vzestupnými čísly s počátečním číslem 1 (tedy 1, 2, 3, ...) a jejich přípona bude tříznaková (např. jpg, png, ...). Tuto složku následně uživatel nahraje na FTP server do složky s ostatními jízdami (www/pictures/).
- Ve formuláři uživatel vyplní datum jízdy.
- Aby byly doplněny informace k jednotlivým pozicím, je nutné vložit textový soubor obsahující tyto informace. Textový soubor s formátováním UTF-8 (pro správnou funkci diakritiky) bude obsahovat informace pro každou pozici tak, jak je znázorněno na obrázku v příloze H, v sekci Náповěda. Data jsou pod sebou v tomto pořadí: název souboru s příponou, zeměpisná šířka, zeměpisná délka, čas pořízení fotografie, nadmořská výška a popis pozice. Pokud některá informace chybí, tento řádek se ponechá prázdný. Každá pozice, tedy odstavec v textovém souboru, bude oddělena jedním volným řádkem.

Náповěda postupu je vždy nejprve skryta a uživatel si ji může kdykoliv zobrazit kliknutím na text „náповěda“.

Druhá možnost (spodní část) vytvoření nové jízdy je praktičtější a jednodušší, než první možnost. Omezena je však nastavením velikosti nahrávaných souborů na server. Pokud vkládaný soubor přesáhne povolenou velikost (pro informaci je maximální velikost zapsána u formuláře), musí uživatel použít metodu první. Postup pro vytvoření jízdy touto metodou je téměř stejný, jako u metody první. Uživatel však nemusí přistupovat na FTP server, složku s fotografiemi a textový soubor, která mají dodržena pravidla, jednoduše zabalí do formátu ZIP se stejným názvem jako název složky, tedy datum ve formátu YYYY-MM-DD, vloží jej do aplikace a formulář odešle.

O výsledku vytvoření nové jízdy je uživatel informován zprávou v levé horní části aplikace.

5.2.3 Seznam jízd

V aplikaci je důležité mít přehled o všech existujících jízdách, u kterých je dále možná jejich editace a smazání (viz příloha I). Černě zvýrazněné možnosti „Edit“ a „Smazat“ fungují jako tlačítka. Tlačítko „Edit“ uživatele přesune do podrobné editace vybrané jízdy a tlačítko „Smazat“ má kvůli bezpečnosti zaimplementovanou kontrolu potvrzení smazání proto, aby nedošlo k nechtěnému smazání jízdy. Po potvrzení se jízda smaže. Jak již bylo řečeno dříve, místo reálného smazání jízdy se aplikaci záznam pouze zneviditelní změnou jednoho atributu v databázi. Možnost úplného smazání záznamu z databáze včetně smazání souboru s fotografiemi ze serveru má pouze super-admin, který bude mít jako jediný viditelný dodatečný

seznam smazaných jízd s možností obnovy nebo kompletního smazání. O výsledku, zda se jízda smazala úspěšně či nikoliv, je uživatel informován zprávou v levé horní části aplikace.

5.2.4 Editace jízdy

Editace existující jízdy obsahuje všechna data této jízdy a jejich jednotlivých pozic. Vzhledem k tomu, že každá jízda může mít různý počet pozic, byla editace umístěna do dynamické tabulky a jednotlivá data lze měnit podobně, jako v softwaru MS Excel (viz příloha J). To pro uživatele znamená, že stačí kliknout na požadované políčko a text nebo číslo se následně zobrazí v editačním řádku. Po přepsání hodnoty stačí kliknout kamkoliv mimo dané políčko a aplikace pomocí technologie AJAX změnu uloží.

Jedinou výjimkou je změna fotografie dané pozice. Tu uživatel musí provést nahrazením souboru na FTP serveru. Do následující verze aplikace se však nabízí dodatečná implementace jednoduché změny fotografie přímo zde v editaci jízdy výběrem nové fotografie z disku počítače, kterou sama aplikace nahradí na serveru FTP.

Ve finále je opět o výsledku provedené operace uživatel informován zprávou v levé horní části aplikace.

5.2.5 Přidání uživatele

Na stránce vytvoření nového uživatele (viz příloha K) se v aplikaci zatím nachází pouze nejnutnější údaje a to přihlašovací jméno, heslo, které je kontrolováno proti chybnému zápisu opětovným zapsáním hesla, a možnost udělení role super-admina. Vyplnění prvních tří polí je povinné.

Na tuto stránku má přístup pouze uživatel v roli super-admina. O výsledku vytvoření nového účtu je uživatel informován zprávou v levé horní části aplikace.

Pokud bude v budoucnosti potřeba přidat více údajů o uživateli, nebude toto doplnění nijak komplikované.

5.2.6 Seznam uživatelů

Stejně tak, jako je naimplementována stránka zobrazení seznamu jízd, je v aplikaci naimplementován seznam existujících uživatelů, které je možno dále editovat a mazat (viz příloha L). Na tuto stránku má přístup pouze uživatel v roli super-admina. Černě zvýrazněné možnosti „Edit“ a „Smazat“ fungují jako tlačítka. Tlačítko „Edit“ uživatele přesune do podrobné editace vybraného uživatele a tlačítko „Smazat“ má pro bezpečnost zaimplementovanou kontrolu potvrzení smazání, aby se předešlo nechtěnému smazání uživatele. Po potvrzení se uživatel smaže stejným způsobem jako v seznamu jízd v kapitole 5.2.3, tedy pouhou změnou jednoho atributu v databázi pro zneviditelnění. Super-admin zde má ale také seznam smazaných uživatelů s možností obnovy nebo úplného smazání. Došlo-li ke smazání uživatele či nikoliv, je uživatel informován zprávou v levé horní části aplikace.

5.2.7 Editace uživatele

Editace uživatele nemusí být nijak komplikovaná (viz příloha M). V aktuální verzi aplikace formulář o uživateli obsahuje pouze přihlašovací jméno, heslo a možnost přidání nebo odebrání role super-admina. Je zde také přidána kontrola hesla. I když má uživatel heslo, v editaci jsou pole pro heslo ponechána prázdná a vyplňují se pouze v případě, pokud chce uživatel heslo změnit. O výsledku editace existujícího uživatele je uživatel informován zprávou v levé horní části aplikace.

5.3 Ekonomické zhodnocení aplikace

Výslednou webovou aplikaci není možno žádným způsobem monetizovat. Lze ji však efektivně využít z marketingového hlediska a také z hlediska získání nových studentů pro Mendelovu univerzitu.

Marketingovou stránkou je myšlena reklama školního arboreta. Díky této aplikaci má široká veřejnost možnost nepřímo nahlédnout do arboreta a získat tak větší motivaci pro osobní návštěvu. Jelikož se jedná o ojedinělou aplikaci, kterou nedisponuje žádná jiná univerzita v Jihomoravském kraji, a dokonce ani v celé České republice, jedná se o významné zviditelnění Mendelovy univerzity a zároveň zviditelnění samotného arboreta univerzity.

Cenu aplikace je možné teoreticky vyčíslit. Její cena by při tvorbě na zakázku, nebo zaměstnancem školy, jenž by aplikaci tvořil přibližně jeden měsíc, byla 15 000 Kč. Tato částka je tedy teoretickým vyčíslením její hodnoty.

Investice může být nejlépe navracena díky výnosům z návštěvnosti školního arboreta. Měsíční návštěvnost s výnosem je následující¹:

- individuální vstupné (85 osob) – 3 660 Kč,
- kolektivní vstupné (134 osob) – 3 436 Kč,
- vstupné o výstavách (1 488 osob) – 64 298 Kč,
- soukromé akce - svatby, focení a další (6 klientů) – 21 196 Kč.

Jelikož se jedná o zimní období, je velice pravděpodobné, že v jarním, letním a podzimním období bude návštěvnost ještě vyšší. Z ročního hlediska tedy arboretum Mendelovy univerzity v Brně získá výnos minimálně:

92 590 Kč po dobu 12 měsíců, tedy 1 111 080 Kč.

¹ Data byla získána z informačního centra Botanické zahrady a arboreta Mendelovy univerzity v Brně a jedná se o data pořízená a průměrovaná z prvních čtyř měsíců roku 2015.

Pro navrácení teoreticky investované částky 15 000 Kč do aplikace po dobu jednoho roku by bylo potřebné, aby se návštěvnost zvýšila za rok v průměru o 1,35%, neboli:

$$15\,000\text{ Kč} / 1\,111\,080\text{ Kč} * 100\% = 1,35\%. \quad (1)$$

Po přepočítaném průměrném ročním počtu 20 556 návštěvníků je potřeba získat dodatečných pouze 228 návštěvníků. Pokud se bude brát v úvahu, že v dnešní době, převážně s technologií Google Street View, je tento typ aplikace v podstatě vyžadován jako posun s dobou, toto navýšení návštěvnosti je díky takové aplikaci naprosto minimální.

6 Závěr

Cílem této bakalářské práce bylo vytvořit webovou aplikaci prezentující panoramatické fotografie pořízené autonomním robotem vytvořeným školním týmem. Obsahem práce byl popis použitých technologií, analýza výběru daných technologií a celkový popis a návod k použití výsledné aplikace. Součástí je také její ekonomické zhodnocení.

Jako nejvhodnějšími pro tvorbu této aplikace byly vybrány technologie HTML5 a CSS3 pro tvorbu statických stránek, databázový systém MySQL pro uchovávání dat a PHP pro komunikaci s databází. jQuery, jako knihovna JavaScriptu pro interaktivní stránku aplikace a framework Nette, aby byla dodržena struktura aplikace a celková architektura MVP.

Co se týče přínosu práce, výsledná aplikace bude s velkou pravděpodobností na webových stránkách Mendelovy univerzity odkazována a bude přístupná veřejnosti. Tato aplikace bude jak přínosem pro zvýšení počtu návštěvníků arboreta Mendelovy univerzity, tak pro zvýšení zájmu potenciálních nových studentů školy. Dalším významným bodem bude také zviditelnění projektového autonomního robota vytvořeného školním týmem AiStorm.

V závěru došlo k vývoji programu, což však byla pouhá část celkové práce, která obsahovala především analýzu a návrh tohoto informačního systému. Zbylá část patří údržbě a úpravám. Tím je míněno, že je aplikace připravena na jakoukoliv potřebnou úpravu jak po technické, tak po vzhledové stránce.

7 Literatura

- About SQLite*. SQLite [online]. 2000, 2015 [cit. 2015-02-13]. Dostupné z: <<https://www.sqlite.org/about.html>>
- BORONCZYK, TIM. *PHP 6, MySQL, Apache: vytváříme webové aplikace*. Vyd. 1. Brno: Computer Press, 2009, 816 s. ISBN 978-80-251-2767-4.
- BROWN, TIFFANY B, KERRY BUTTERS A SANDEEP PANDA. *HTML5 okamžitě: [ovládněte HTML5 za víkend]*. 1. vyd. Brno: Computer Press, 2014, 256 s. ISBN 978-80-251-4296-7.
- CASTRO, ELIZABETH A BRUCE HYSLOP. *HTML5 a CSS3: názorný průvodce tvorbou WWW stránek*. 1. vyd. Brno: Computer Press, 2012, 439 s. ISBN 978-80-251-3733-8.
- DEMONTE, JEAN-BAPTISTE, AURELIEN GERITS A SAMUEL LESIRE. *Gmap3* [online]. 2010 [cit. 2015-02-03]. Dostupné z: <<http://gmap3.net/>>
- DIFFERENCEBETWEEN.INFO. *Difference between UML and ERD*. DifferenceBetween.info [online]. © 2012 - 2014 [cit. 2015-02-10]. Dostupné z: <<http://www.differencebetween.info/difference-between-uml-and-erd>>
- DOT COM INFOWAY. *Dot Com Infoway* [online]. © 2015 [cit. 2015-02-12]. Dostupné z: <<http://www.dotcominfoway.com>>
- EHINGER, BENJAMIN. *MySQL vs PostgreSQL*. ITX Design [online]. 21.2.2014 [cit. 2015-01-26]. Dostupné z: <<http://itxdesign.com/mysql-vs-postgresql>>
- EMMATTY, JOHN T. *Differences between MVC and MVP for Beginners*. CodeProject [online]. 23.11.2011 [cit. 2015-02-03]. Dostupné z: <<http://www.codeproject.com/Articles/288928/Difference-s-between-MVC-and-MVP-for-Beginners>>
- FORTECCHIO, MARK. *Oracle the clear leader in \$24 billion RDBMS market*. IT Knowledge Exchange [online]. 12.4.2012 [cit. 2015-01-20]. Dostupné z: <<http://itknowledgeexchange.techtarget.com/eye-on-oracle/oracle-the-clear-leader-in-24-billion-rdbms-market>>
- GRUDL, DAVID. *Nette Framework: MVC & MVP*. Zdroják [online]. 24.3.2009 [cit. 2015-01-06]. ISSN 1803-5620. Dostupné z: <<http://www.zdrojak.cz/clanky/nette-framework-mvc-mvp>>
- JAKOUBĚ, JAROSLAV. *ORM test PHP frameworků*. Zdroják [online]. 10.7.2013 [cit. 2015-02-04]. Dostupné z: <<http://www.zdrojak.cz/serialy/test-php-frameworku>>
- jQuery - kuchařka programátora*. Vyd. 1. Brno: Computer Press, 2010, 436 s. ISBN 978-80-251-3152-7.

- KADLEC, TIM. *Responzivní design profesionálně*. Vyd. 1. Brno: Zoner Press, 2014, 246 s. ISBN 978-80-7413-280-3.
- LECKY-THOMPSON, ED A STEVEN D NOWICKI. *PHP 6: programujeme profesionálně*. Vyd. 1. Brno: Computer Press, 2010, 718 s. ISBN 978-80-251-3127-5.
- LUBBERS, PETER, BRIAN ALBERS A FRANK SALIM. *HTML5: programujeme moderní webové aplikace*. Vyd. 1. Brno: Computer Press, 2011, 304 s. ISBN 978-80-251-3539-6.
- MAJDA, DAVID. *Co přináší nový ECMAScript 5?*. Zdroják [online]. 20.5.2009 [cit. 2015-02-19]. ISSN 1803-5620. Dostupné z: <<http://www.zdrojak.cz/clanky/co-prinasi-novy-ecmascript-5>>
- MORAVEC, ZDENĚK. *PHP a MySQL – MySQLi – 1. díl*. Programujte.com [online]. 14.1.2010 [cit. 2015-01-27]. ISSN 1801-1586. Dostupné z: <<http://programujte.com/clanek/2009103100-php-a-mysql-mysqli-1-dil>>
- NETTE FOUNDATION. *Creative Commons CC-BY-SA*. Dokumentace [online]. © 2008, 2014 [cit. 2014-12-17]. Dostupné z: <<http://doc.nette.org/cs>>
- ORACLE CORPORATION. *MySQL Documentation* [online]. Redwood Shores (CA), © 2014 [cit. 2014-12-15]. Dostupné z: <<http://dev.mysql.com/doc>>
- ORACLE CORPORATION. *NetBeans* [online]. © 2015 [cit. 2015-02-12]. Dostupné z: <<https://netbeans.org>>
- REFSNES DATA. *AJAX Introduction*. W3schools.com [online]. © 1999-2015 [cit. 2015-02-19]. Dostupné z: <http://www.w3schools.com/ajax/ajax_intro.asp>
- REFSNES DATA. *JavaScript Libraries*. W3schools.com [online]. © 1999-2015 [cit. 2015-01-14]. Dostupné z: <http://www.w3schools.com/js/js_libraries.asp>
- ROJWONGSURIYA. *Panorama_viewer*. GitHub, Inc. GitHub [online]. 2010, 2015 [cit. 2015-02-03]. Dostupné z: <https://github.com/peachananr/panorama_viewer>
- SRNA, MICHAL. *Panoramatická fotografie snadno a rychle*. Digimanie [online]. 11.10.2010 [cit. 2015-04-02]. Dostupné z: <<http://www.digimanie.cz/panoramaticka-fotografie-snadno-a-rychle/3441>>
- System Properties Comparison MySQL vs. Oracle vs. PostgreSQL*. SOLID IT GMBH. DB-Engines [online]. © 2015 [cit. 2015-04-07]. Dostupné z: <<http://db-engines.com/en/system/MySQL%3BOracle%3BPostgreSQL>>
- ŠKRÁŠEK, JAN. *PHP frameworky*. Programujte.com [online]. 21.2.2008 [cit. 2015-02-03]. Dostupné z: <<http://programujte.com/clanek/2008022000-php-frameworky>>

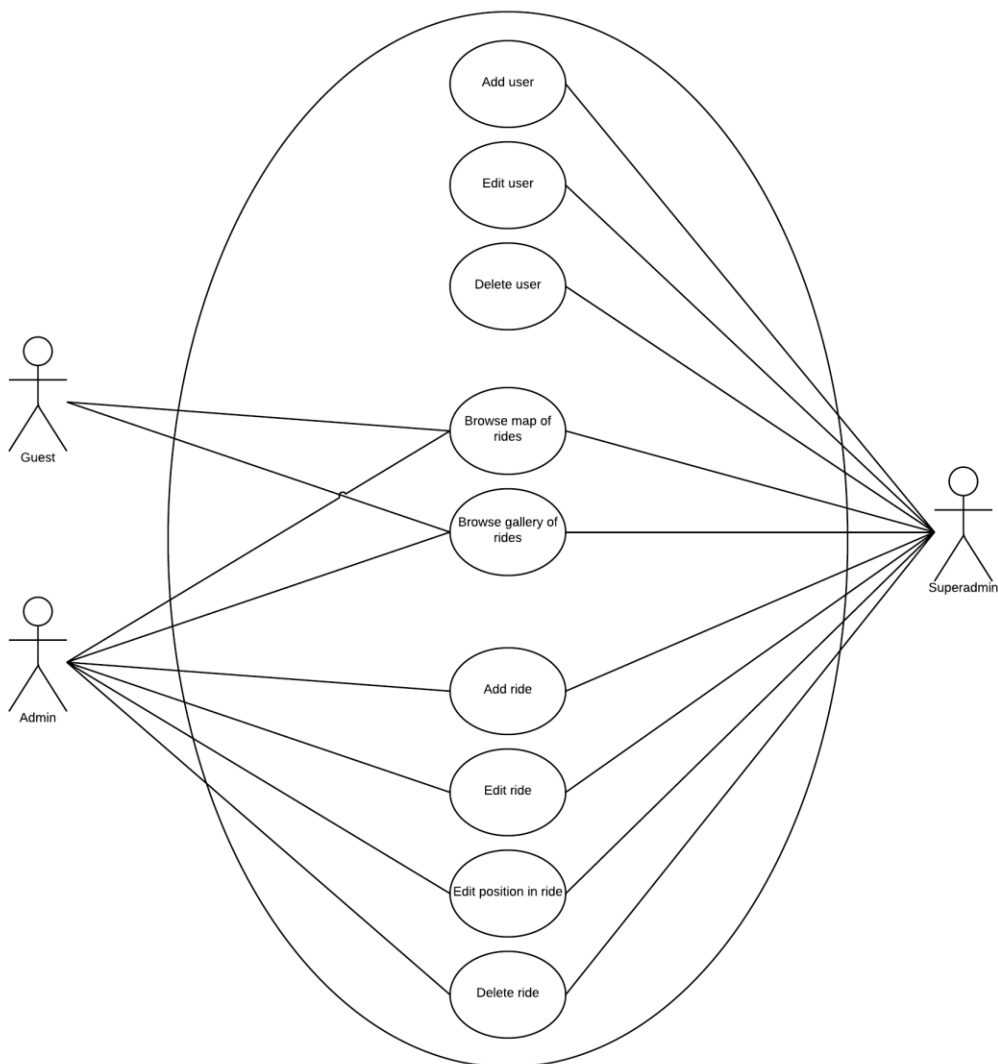
- TEZER, O.S. *SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems*. Digital Ocean [online]. © 2014 [cit. 2015-02-13]. Dostupné z: <<https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems>>
- THE PHP GROUP. *Documentation* [online]. © 2001, 2014 [cit. 2014-12-17]. Dostupné z: <<http://www.php.net/manual/en>>
- THE REGENTS OF THE UNIVERSITY OF CALIFORNIA. *Selecting Your Database Platform*. UC Berkley IST [online]. © 2007, 2012-03-21 [cit. 2015-01-20]. Dostupné z: <https://ist.berkeley.edu/is/database/services/selecting_a_platform>
- XUL.FR. *JavaScript libraries and frameworks*. Xul [online]. © 2008-2014 [cit. 2015-02-19]. Dostupné z: <<http://www.xul.fr/javascript/frameworks.php>>
- ZAJÍC, PETR. *PHP (1) - Historie a budoucnost*. Linuxsoft.cz [online]. 27.5.2004 [cit. 2014-12-17]. ISSN 1801-3805. Dostupné z: <http://www.linuxsoft.cz/article.php?id_article=171>
- ZAKAS, NICHOLAS C. *JavaScript pro webové vývojáře*. Vyd. 1. Brno: Computer Press, 2009, 832 s. ISBN 978-80-251-2509-0.
- ZHENG, HAO. [obrázek]. Toyhouse [online]. 7.12.2011 [cit. 2015-01-13]. Dostupné z: <<http://toyhouse.cc/profiles/blogs/personal-report-on-learning-javascript>>

Přílohy

A Dokumentace problému s roamingem

Arboretum View

Use-Case Diagram



Obr. 4 Use-Case diagram aplikace

B Interaktivní mapa jízd

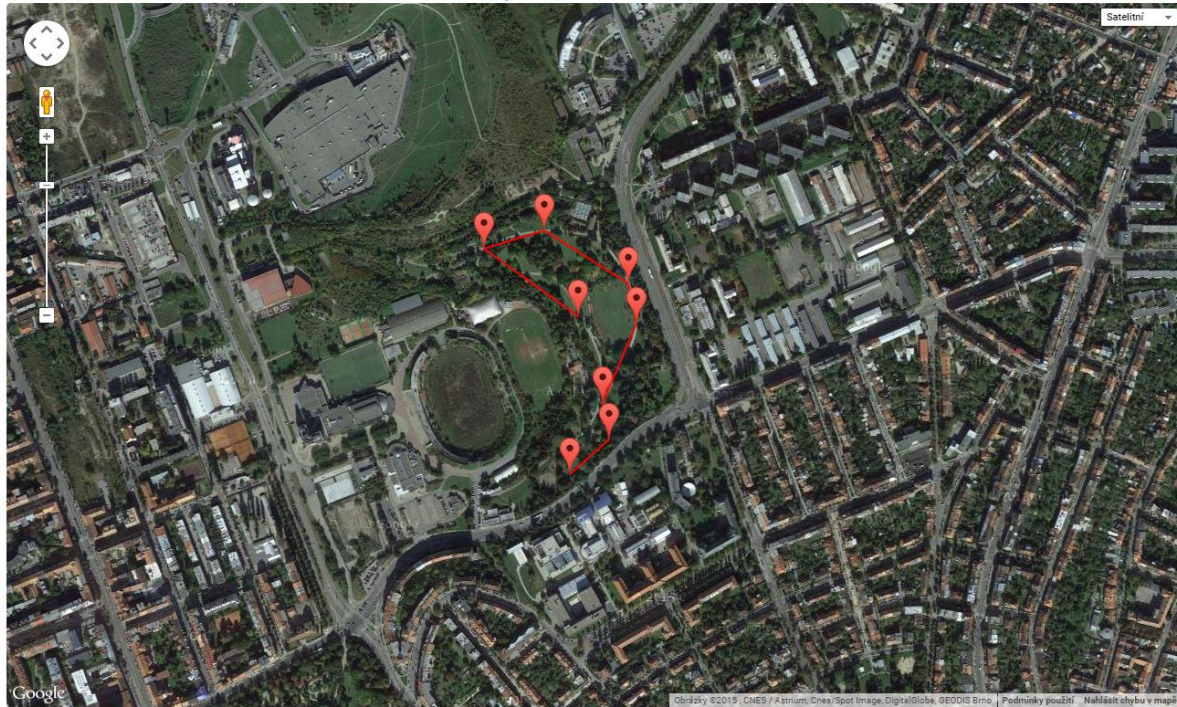
ARBORETUM VIEW

ADMINISTRACE GALERIE MAPA

Mapa arboreta - 13.04.2015

Jízdy

- 13.04.2015
- Dubnové focení
- 05.12.2014
- 04.12.2014
- 05.11.2014
- 10.03.2014



Google

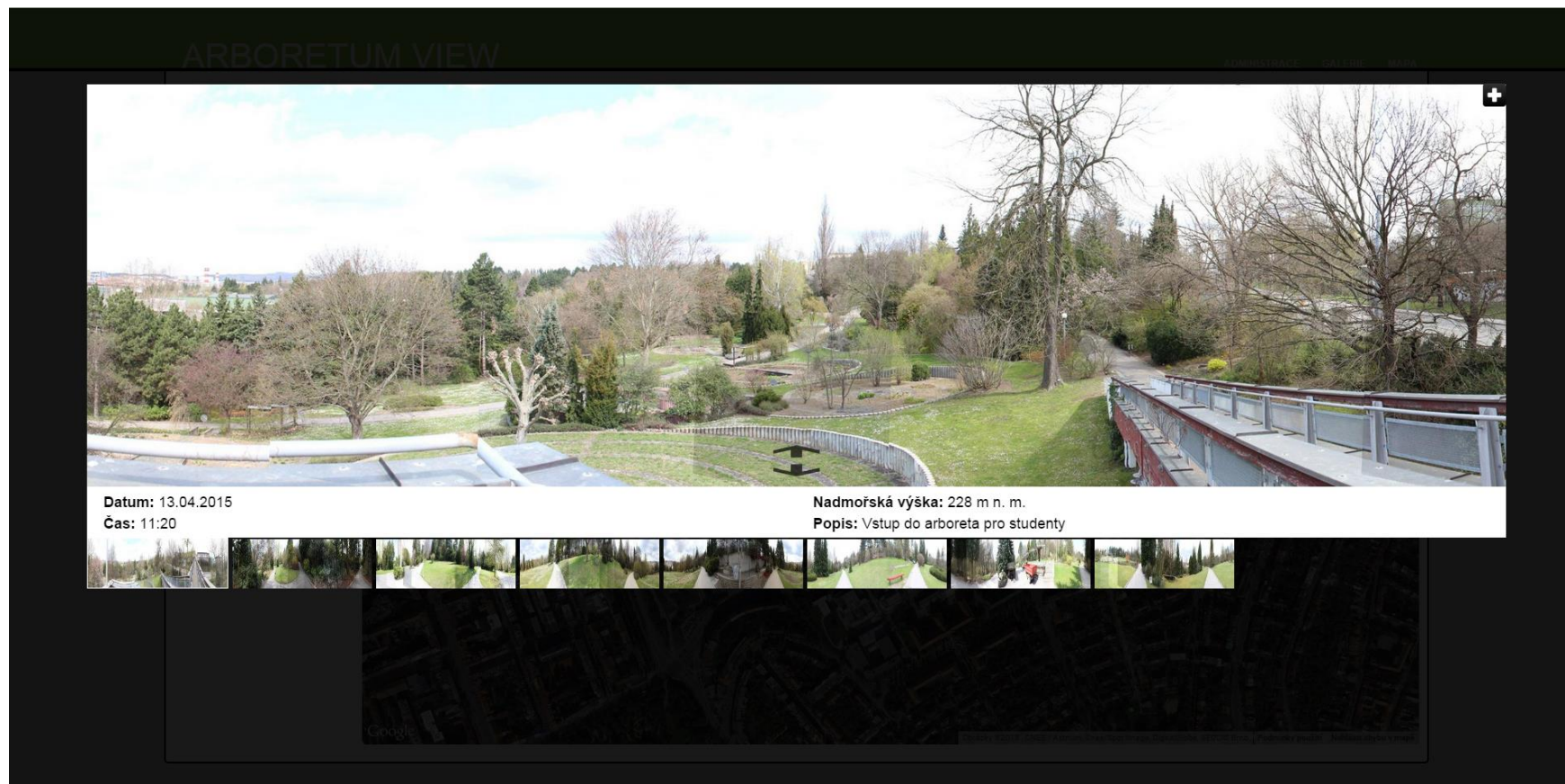
Obrázky ©2015, CNES / Astrium, Ches/Spot Image, DigitalGlobe, GEOSIS Erno, Podmínky použití, Nahlásit chybu v mapě

Copyright Jan Zálešák

The image shows a satellite map of an arboretum area. A red path is overlaid on the map, connecting several red location markers. The path starts in the upper right and moves generally south and west. The map includes a compass rose, a street view pegman, and a vertical scale bar on the left side. The title 'Mapa arboreta - 13.04.2015' is centered above the map. A green header bar at the top contains the text 'ARBORETUM VIEW' and navigation links 'ADMINISTRACE', 'GALERIE', and 'MAPA'. A sidebar on the left lists dates under the heading 'Jízdy'. The Google logo is visible in the bottom left corner of the map area, and copyright information is at the bottom.

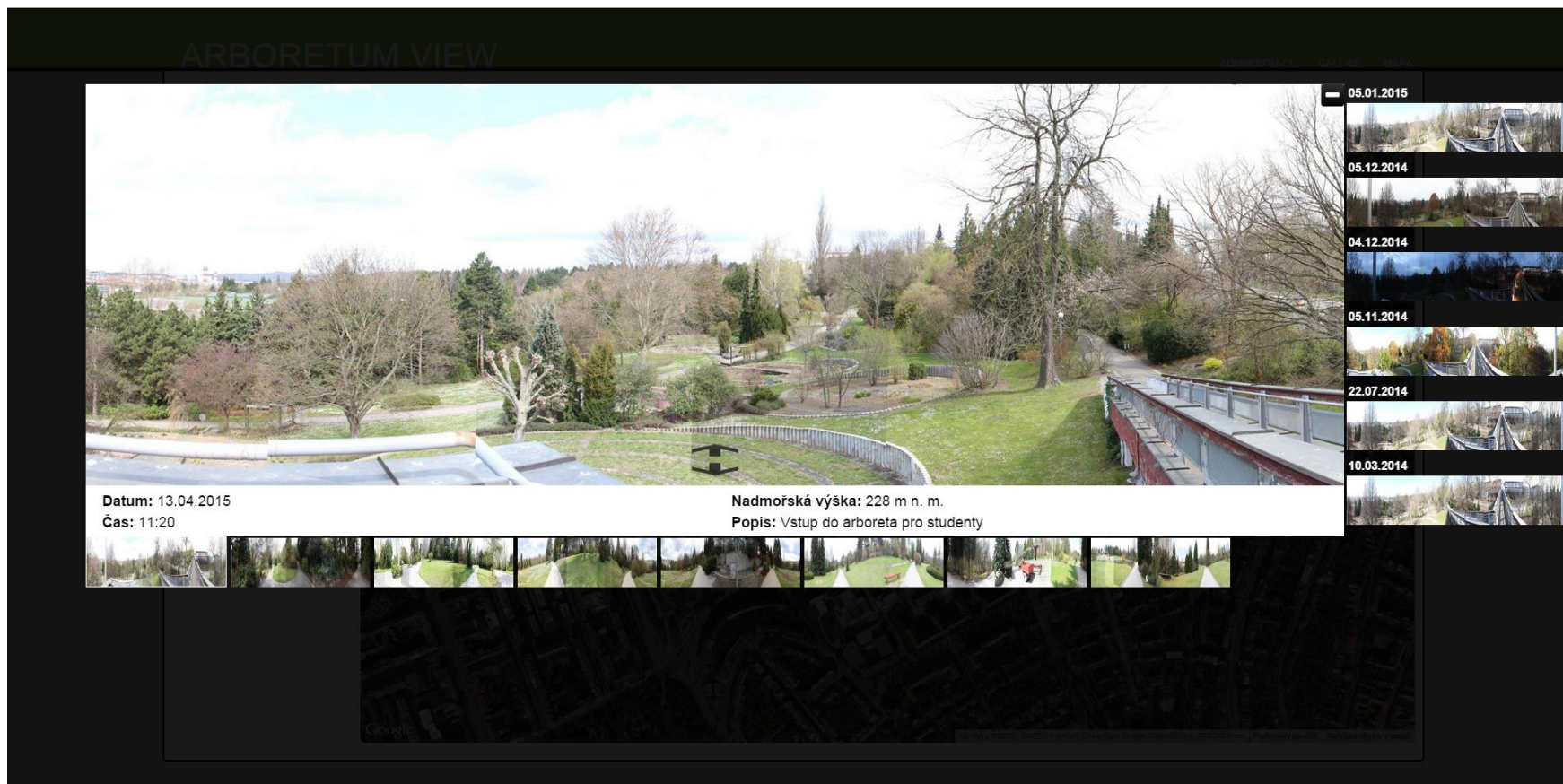
Obr. 5 Prezentáční stránka – interaktivní mapa jízd

C Zobrazení panoramatické fotografie



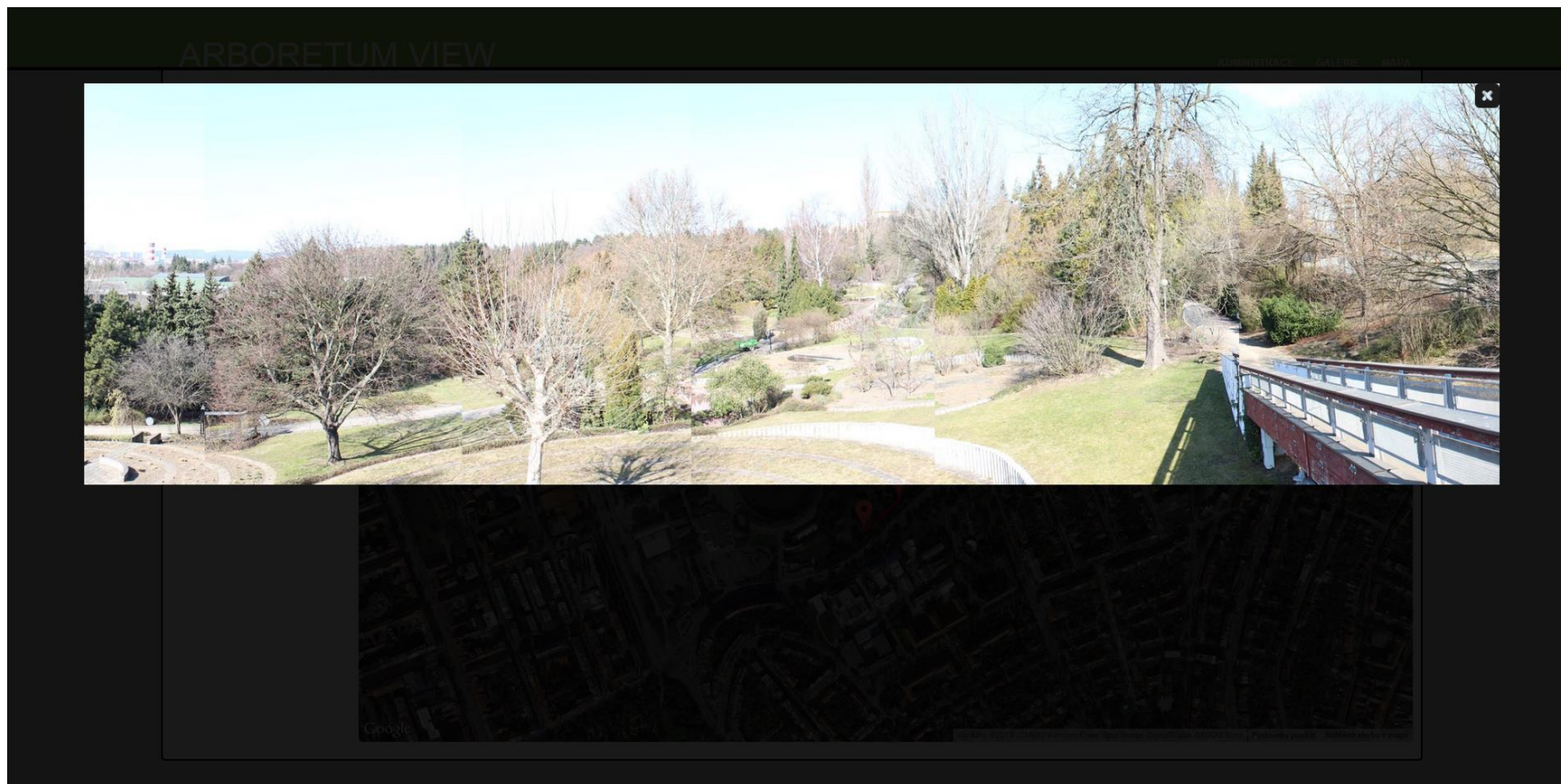
Obr. 6 Prezentací stránka – zobrazení panoramatické fotografie vybrané pozice

D Seznam stejných pozic zobrazené fotografie



Obr. 7 Prezentační stránka – seznam stejných pozic aktuálně zobrazené fotografie v jiných obdobích

E Zobrazení fotografie stejné pozice jiného období



Obr. 8 Prezentační stránka – zobrazení fotografie stejné pozice jiného období

F Galerie jízd



Obr. 9 Prezentáční stránka – galerie jízd

G Statistiky a provedené činnosti uživatelů

The screenshot displays the 'ARBORETUM VIEW' administration interface. At the top, a green navigation bar contains the title 'ARBORETUM VIEW' and several menu items: STATISTIKY, NOVÁ JÍZDA, SEZNAM JÍZD, NOVÝ UŽIVATEL, SEZNAM UŽIVATELŮ, GALERIE, MAPA, and ODHLÁSIT. The main content area is divided into two sections: 'Statistiky' and 'Provedené činnosti'.

Statistiky

- Celkový počet jízd: 7
- Celkový počet pozic: 31
- Počet fotek v cache: 60
- [Smazat cache](#)

Provedené činnosti

- 21.04.2015 18:26 - Uživatel "admin" zeditoval jízdu 13.04.2015
- 21.04.2015 18:20 - Uživatel "admin" smazal uživatele user
- 21.04.2015 18:20 - Uživatel "admin" vytvořil uživatele user
- 21.04.2015 18:19 - Uživatel "admin" smazal jízdu 05.01.2015
- 21.04.2015 18:19 - Uživatel "admin" smazal jízdu 22.07.2014
- 21.04.2015 18:18 - Uživatel "admin" vytvořil jízdu 05.01.2015
- 21.04.2015 18:18 - Uživatel "admin" vytvořil jízdu 22.07.2014
- 20.04.2015 21:13 - Uživatel "admin" obnovil jízdu 13.04.2015
- 20.04.2015 21:12 - Uživatel "admin" smazal jízdu 13.04.2015
- 20.04.2015 21:12 - Uživatel "admin" vytvořil jízdu 13.04.2015
- 20.04.2015 20:14 - Uživatel "admin" obnovil uživatele xzalesa1
- 20.04.2015 20:14 - Uživatel "admin" smazal uživatele xzalesa1
- 20.04.2015 20:11 - Uživatel "admin" smazal uživatele xzalesa1
- 20.04.2015 16:18 - Uživatel "admin" zeditoval jízdu 05.11.2014

Copyright Jan Zálešák

Obr. 10 Administrační stránka – statistiky a provedené činnosti uživatelů

H Vytvoření nové jízdy

ARBORETUM VIEW

STATISTIKY NOVÁ JÍZDA SEZNAM JÍZD NOVÝ UŽIVATEL SEZNAM UŽIVATELŮ GALERIE MAPA ODHLÁSIT

Nová jízda

Datum jízdy:

Soubor txt: Soubor nevybrán

-nápověda: -Nahrávejte do složky 'www/pictures/' složku pojmenovanou podle data jízdy ve formátu YYYY-MM-DD

- Fotky budou pojmenovány čísly, počínaje číslem 1 (1, 2, 3, ...). Soubor musí mít 3-znakovou příponu (jpg, png, ...).
- Txt soubor s formátováním UTF-8 bude obsahovat na každém řádku samostatně název souboru s příponou, lat, lng, čas, alt, popis
- Každá pozice bude oddělena volným řádkem. Název souboru, lat a lng jsou povinná. Pokud nějaká data chybí - nechat volný řádek.

```
data.txt - Poznámkový blok
Soubor  Úpravy  Formát  Zobrazení  Nápověda
1.png
40.211585
16.614111
09:45
238
Vstup z areálu Mendelů
2.png
40.21212
16.61503
09:58
235
<= vynechat pokud chybí popis
...
```

Import do 128MB.

Soubor zip: Soubor nevybrán

-nápověda:

Copyright Jan Zálešák

Obr. 11 Administrační stránka – vytvoření nové jízdy

I Seznam jízd

The screenshot displays the 'ARBORETUM VIEW' administrative interface. At the top, there is a green navigation bar with the title 'ARBORETUM VIEW' and several menu items: STATISTIKY, NOVÁ JÍZDA, SEZNAM JÍZD, NOVÝ UŽIVATEL, SEZNAM UŽIVATELŮ, GALERIE, MAPA, and ODHLÁSIT. Below the navigation bar, the main content area is divided into two sections. The first section, titled 'Seznam jízd', contains a list of five entries, each with a date and a link to 'Edit Smazat'. The second section, titled 'Seznam smazaných jízd', contains two entries, each with a date and a link to 'Obnovit Smazat nenávratně'. The interface is framed by a grey border.

Seznam jízd	
13.04.2015	Edit Smazat
05.12.2014	Edit Smazat
04.12.2014	Edit Smazat
05.11.2014	Edit Smazat
10.03.2014	Edit Smazat

Seznam smazaných jízd	
05.01.2015	Obnovit Smazat nenávratně
22.07.2014	Obnovit Smazat nenávratně

Copyright Jan Zálešák

Obr. 12 Administrační stránka – seznam jízd

J Editace existující jízdy







ARBORETUM VIEW

[STATISTIKY](#)
[NOVÁ JÍZDA](#)
[SEZNAM JÍZD](#)
[NOVÝ UŽIVATEL](#)
[SEZNAM UŽIVATELŮ](#)
[GALERIE](#)
[MAPA](#)
[ODHLÁSIT](#)

Editace jízdy

Datum jízdy:

Popis:

Fotka	Popis	Čas	Výška	Zem. šířka.	Zem. výška.
	Vstup do arboreta pro studenty	11:20:00	228 m n.m.	49.211585 s.š.	16.614111 v.d.
		11:25:00	235 m n.m.	49.21212 s.š.	16.61503 v.d.
		11:30:00	232 m n.m.	49.212678 s.š.	16.614882 v.d.
		11:35:00	242 m n.m.	49.2139 s.š.	16.615678 v.d.
		11:40:00	246 m n.m.	49.21451 s.š.	16.61549 v.d.
		11:45:00	239 m n.m.	49.21532 s.š.	16.61351 v.d.
		11:50:00	235 m n.m.	49.21504 s.š.	16.61209 v.d.
		11:55:00	232 m n.m.	49.21401 s.š.	16.61432 v.d.

Copyright Jan Zálesák

Obr. 13 Administrační stránka – editace existující jízdy

K Vytvoření nového uživatele

The screenshot shows a web interface for 'ARBORETUM VIEW'. At the top, there is a green navigation bar with the site name and several menu items: STATISTIKY, NOVÁ JÍZDA, SEZNAM JÍZD, NOVÝ UŽIVATEL, SEZNAM UŽIVATELŮ, GALERIE, MAPA, and ODHLÁSIT. Below the navigation bar, the main content area is titled 'Nový uživatel' (New user). This section contains a form with the following fields and elements:

- Login:
- Heslo:
- Heslo znovu:
- Super-admin
- A green button labeled 'Vytvořit uživatele' (Create user).

At the bottom of the page, there is a small copyright notice: 'Copyright Jan Zálešák'.

Obr. 14 Administrační stránka – vytvoření nového uživatele

L Seznam uživatelů

The screenshot displays the 'Seznam uživatelů' (User List) page in the ARBORETUM VIEW system. The page features a green header with the title 'ARBORETUM VIEW' and a navigation menu with links: STATISTIKY, NOVÁ JÍZDA, SEZNAM JÍZD, NOVÝ UŽIVATEL, SEZNAM UŽIVATELŮ, GALERIE, MAPA, and ODHLÁSIT. The main content area is divided into two sections:

- Seznam uživatelů** (Active Users):
 - admin [Edit](#) [Smazat](#)
 - xzalesa1 [Edit](#) [Smazat](#)
- Seznam smazaných uživatelů** (Deleted Users):
 - user [Obnovit](#) [Smazat](#) [nenávratně](#)

At the bottom of the page, there is a small copyright notice: Copyright Jan Zálešák.

Obr. 15 Administrační stránka – seznamu uživatelů

M Editace uživatele

ARBORETUM VIEW

STATISTIKY NOVÁ JÍZDA SEZNAM JÍZD NOVÝ UŽIVATEL SEZNAM UŽIVATELŮ GALERIE MAPA ODHLÁSIT

Editace uživatele

Login:

Heslo:

Heslo znovu:

Super-admin

Copyright Jan Zálešák

Obr. 16 Administrační stránka – editace uživatele