

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA APLIKOVANÝCH VĚD
KATEDRA INFORMATIKY A VÝPOČETNÍ TECHNIKY

Bakalářská práce

Komunitní web pro hledače pokladů

Plzeň, 2014

Jiří Vyhnálek

Obsah

1 Úvod	5
Úvod	5
1.1 Kdo jsou hledači pokladů	5
1.2 Motivace pro řešení	5
2 Analýza potřeb a očekávání cílové skupiny	6
2.1 Zaměření projektu	6
2.2 Registrace	6
2.3 Přihlášení	7
2.4 Přátelé	7
2.5 Výpis a filtrace nálezů	7
2.5.1 Zobrazení na mapě	7
2.5.2 Filtrace a vyhledávání	8
2.5.3 Seznam nálezů	8
2.6 Nález	8
2.6.1 Základní informace	8
2.6.2 Umístění	9
2.6.3 Fotogalerie	9
2.6.4 Sociální aspekt a sdílení	9
2.6.5 Komentáře	10
2.7 Přidávání nálezu	10
3 Platforma a datový model	11
3.1 Výběr platformy	11
3.1.1 Vlastní řešení	11
3.1.2 PHP Framework	12
3.2 Výběr frameworku	14
3.2.1 Nette Framework	14
3.2.2 Symfony2	15
3.2.3 Zend Framework	15
3.2.4 Phalcon	16
3.2.5 Laravel	16
3.3 Datový model	17

4 Implementace	18
4.1 Modely	18
4.1.1 Relace mezi modely	18
4.1.2 User	19
4.1.3 Finding	20
4.1.4 Controllery	21
4.2 Šablona	23
4.2.1 Použité technologie a knihovny	24
4.3 Popis funkčnosti	24
4.3.1 Přihlášení a registrace	24
4.3.2 Výpis a filtrace nálezů	26
4.3.3 Detail nálezu - umístění	29
4.3.4 Detail nálezu - komentáře	30
4.3.5 Editace nálezu	30
4.3.6 Editace fotogalerie	31
5 Ohlasy a návrhy na vylepšení	33
5.1 Co se uživatelům líbí	33
5.1.1 Přehlednost	33
5.1.2 Mapa a filtrování	33
5.1.3 Dělení na veřejnou a neveřejnou sekci	33
5.2 Návrhy na vylepšení	34
5.2.1 Barevné téma	34
5.2.2 Seznámení nově příchozího	34
5.3 Návrhy na rozšíření	34
5.3.1 Diskusní fórum	34
5.3.2 Anketní systém	34
5.3.3 Články	34
5.3.4 Bazar	35
6 Závěr	36
Použitá literatura a zdroje	37
Seznam obrázků	38

Poděkování

Rád bych touto cestou poděkoval vedoucímu své práce Ing. Daliboru Fialovi Ph.D. za trpělivost a cenné rady poskytnuté při zhotovování bakalářské práce. Dále bych chtěl poděkovat kamarádovi Patriku Voglovi, který mi pomohl navrhnout grafickou podobu některých sekcí na webu.

Čestné prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracoval(a) samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne 10. 6. 2014

Abstrakt

English

The main objective of this work was to create a community site for treasure seekers, sometimes referred to as mrcasnik. The idea was to create an application that would mainly work as a tool for cataloging findings for individuals. Basic features of social networks should be also implemented.

The theoretical part of the work was devoted to identifying the requirements that similar application arises. An integral part of it was an understanding of the current situation on the Czech internet, finding any similar applications already working. It was necessary to establish requirements for the functionality of the app from the perspective of an average PC user. It was also necessary to decide which implementation techniques to choose.

The web application was implemented using techniques selected and tested by a group of real users. The output from users was collected. This will serve as the basis for future expansion and improvement of the application.

Česky

Hlavním cílem této práce bylo vytvoření komunitního webu pro hledače pokladů, takzvané mrcasníky. Záměrem bylo vytvořit aplikaci, která by hlavně fungovala jako nástroj pro katalogizaci nálezů pro jednotlivce. Zároveň by měla implementovat základní znaky komunitních sítí, tj. interakce mezi uživateli.

Teoretická část práce byla věnována zjišťování požadavků, které na podobnou aplikaci vznikají. Nedílnou součástí bylo také zjištění aktuální situace na českém internetu, kdy cílem bylo zjistit, zda podobná aplikace již neexistuje. Bylo třeba sestavit požadavky na funkčnost aplikace z pohledu průměrně zdatného uživatele PC. Dále bylo třeba rozhodnout, jaké implementační techniky zvolit.

Webová aplikace byla implementována za pomoci zvolených technik a otestována skupinou reálných uživatelů. Byly sezbírány výstupy od uživatelů, které budou sloužit jako základ pro budoucí rozšiřování a zlepšování aplikace.

Kapitola 1

Úvod

1.1 Kdo jsou hledači pokladů

Každý jistě někdy zahlédl někde na poli, či za domem muže s podivným nástrojem, který prochází půdu centimetr po centimetru. Čas od času se zastaví, kope a pak jde dál. Jedná se o hledače pokladů, kterým se také občas přezdívá „mrcasníci“.

Mrcasníci jsou lidé, kteří rádi tráví svůj volný čas v přírodě hledáním cenných kovů, válečných artefaktů a podobně. Správný mrcasník by měl dodržovat právo na soukromí a respektovat cizí pozemek, nesmí zničit přírodu, neměl by za sebou zanechávat žádný odpad a místo nálezu opustit v původním stavu.

Hledač pokladů si svůj detektor považuje nade vše, je to pro něj zdroj zábavy, pomocník a útěk před všední, nudnou realitou. Mým cílem v této práci bude vytvořit webovou aplikaci, kde si budou nálezci moci evidovat své úlovky, chlubit se s nimi známým či veřejně, filtrovat si je podle data nálezu, místa, kategorie a podobně. Hledači si budou moci navzájem nálezy komentovat a navazovat přátelství. Budou tak mít na očích vždy nálezy od lidí, kteří je zajímají.

1.2 Motivace pro řešení

Nápad vznikl spontánně někdy v létě roku 2013, kdy si můj otec pořídil svůj první detektor. Hledání ho pohltilo natolik, že většinu víkendu tráví po lesích a na polích v okolí Chebu.

Po nějaké době se mu začaly nálezy hromadit, a když jsem se ho zeptal, zda si vede nějaký katalog či sešit, odpověděl, že zatím nenašel vhodný způsob. Nálezy se začaly kupit a po nějaké době jsem mu navrhl, že bych mohl vhodné řešení zpracovat v rámci bakalářské práce, načež otec nadšeně kývnul.

Kapitola 2

Analýza potřeb a očekávání cílové skupiny

2.1 Zaměření projektu

Jak vyplývá z motivace, projekt je cílený hlavně na konkrétního uživatele jako takového, s prvky komunitního webu. Hlavní důraz je kladen na to, aby měl uživatel po přihlášení přehledně k vidění všechny své nálezy, včetně fotografií, možnosti filtrace a vyhledávání.

Aplikace bude zaměřena na průměrně zdatného uživatele osobního počítače, kterému nedělá problém navigace po internetu a používání e-mailu. Uživatelské prostředí tak musí být jednoduché, přímočaré a vést uživatele tak, aby nemusel tápat, kde najít konkrétní funkcionalitu.

2.2 Registrace

Uživatel se bude muset před použitím aplikace zaregistrovat. Registrace by měla být co nejjednodušší, aby ji zvládl opravdu každý. Registrovaný uživatel bude muset potvrdit registraci pomocí kódu v emailu. Kdyby se ho totiž rozhodl jiný uživatel, případně administrátor, pomocí emailu kontaktovat, musí být dostupný.

Alternativou ke klasické registraci je registrace přes Facebook. V době, kdy Facebook má přes 1.3 miliardy aktivních uživatelů [1] by bylo škoda nevyužít jeho funkcionalitu pro zjednodušení registrace na naší aplikaci.

V případě, že uživatel nebude registrovaný, bude nabídnuta pouze velmi základní funkcionalita. Může tak například prohlížet nálezy, které jsou veřejné. Toto vyplývá z požadavku, kdy nálezce chce svůj poklad dát prohlédnout například jinému členovi rodiny, který se ale mrcasnictví nevěnuje a vytváření uživatelského účtu jen pro tento účel by bylo zbytečné a obtěžující.

2.3 Přihlášení

Při použití klasické registrace se po aktivaci účtu bude moci uživatel přihlásit. Bude mít možnost zvolit volbu „Pamatovat si mě“ pro uložení cookie¹ a možnost automatického přihlášení po restartu prohlížeče. V případě využití registrace přes facebook probíhá přihlašování přes stejné tlačítko.

2.4 Přátelé

Uživatel má možnost navazovat s dalšími uživateli přátelství. Přátelství vytvořím na podobném principu jako na Facebooku. Uživatel si najde člověka, který jej zajímá a odešle mu žádost o přátelství. Druhý uživatel ji buď přijme a stanou se přáteli, v opačném případě žádost zamítne a přátelství se zakáže.

Přátelství bude v první řadě sloužit k tomu, abych mohl jednoduše vidět nálezy u lidí, kteří mě zajímají. Nálezy těchto přátel pak budou zobrazeny na úvodní straně po přihlášení. Uživatelé tak budou mít vždy přehled o tom, co jejich kamarádi našli nového.

2.5 Výpis a filtrace nálezů

2.5.1 Zobrazení na mapě

Uživatel po návštěvě stránky s výpisem uvidí mapu se všemi svými nálezy. Musí se jednat o interaktivní mapu, s možností přibližování a posouvání. Mapa musí být v základu dostatečně veliká, aby se v ní dalo dobře orientovat. Kliknutím na tlačítko s lupou pak bude uživateli zobrazena mapa přes celé okno prohlížeče.

Jednotlivé nálezy budou na mapě zobrazeny pomocí *markerů*². Každý typ nálezu (kategorie) bude mít svou vlastní ikonu, aby bylo na první pohled na mapu jasné, z které kategorie daný nález je. Pod mapou musí být tím pádem umístěna legenda, která bude popisovat, co která značka znamená.

„Markery“ budou navíc odděleny i barevně, kdy barevná ikona bude sloužit jako reprezentace skutečnosti, že nález byl autorem zařazen mezi jeho oblíbené. Po kliknutí na *marker* bude zobrazen název nálezu, jeho zkrácený popis a odkaz na detail.

¹Cookie je malé množství dat, které server odešle klientovi a data jsou pak uložena na počítači uživatele. Může obsahovat identifikační údaje uživatele a udržovat tak přihlášení i po zavření prohlížeče.

²Marker je ikona na mapě reprezentující umístění jednoho konkrétního bodu zájmu - zde místa nálezu.

2.5.2 Filtrace a vyhledávání

Pod mapou bude mít uživatel možnost nálezy si vyfiltrovat. Mezi parametry pro filtraci bude patřit kategorie, do které daný nález spadá, data, mezi kterými byl předmět nalezen a umístění. Uživatel zadá město či místo a jako druhý parametr uvede počet kilometrů od daného místa. Samozřejmě bude možné vybírat jak mezi všemi nálezy, tak jen mezi oblíbenými a určovat, podle čeho se mají nálezy řadit.

Kategorie pro nálezy budou ze začátku následující:

- Válečné předměty
- Mince
- Šperky a bižuterie
- Předměty z domácnosti
- Neznámý
- Ostatní

Většina nálezů se vejde do výše uvedených kategorií a pro první veřejnou verzi aplikace bude tato množina dostačující. Na základě výstupu od reálných uživatelů pak budou vhodné tipy na kategorie do aplikace doimplementovány.

Samozřejmostí je full-textové vyhledávání, které prohledá veškeré nálezy podle jejich titulku a popisu.

2.5.3 Seznam nálezů

Nálezy musí být seřazeny viditelně, ideálně do boxů. V každém boxu bude na první pohled vidět hlavní fotografie nálezu, označení, zda je oblíbený, titulek a datum nálezu. Dále pak bude obsahovat příslušnou kategorii s proklikem.

Seznam bude možné stránkovat, s nastavenou standardní hodnotou počtu nálezů na stránku. Tato hodnota musí jít v nastavení profilu změnit. Stránkování bude probíhat bez nutnosti obnovení stránky pro zajištění maximálního pohodlí při práci.

2.6 Nález

2.6.1 Základní informace

Nález musí mít v první řadě název, popis, datum nálezu a místo.

Dále bude obsahovat pole „okolnosti nálezu“, kam uživatel bude moci zaznamenat stav půdy, počasí, použitý detektor, čas, případně jakékoli informace navíc, které se nehodí do popisu předmětu, jako takového, ale je dobré informace poskytnout.

Informace bude realizovaná jako textové pole (stejně jako popis), kam si každý uživatel bude moci dopsat, co bude chtít. Samozřejmě by měl být uživatel k informacím, které má vyplnit, nějak naveden. K tomu bude sloužit například bublina, vyskakující při zaktivnění pole, nebo informační text ve formuláři pro vkládání nálezu.

2.6.2 Umístění

Místo nálezu musí být jednoznačně určené pomocí souřadnic, nestačí zadat pouze „okolí Třemošné“. Uživatel bude chtít svůj nález vidět přímo na mapě. Umístění informací o zeměpisné délce a šířce umožní filtraci právě podle těchto parametrů a zajistí dobrou udržitelnost aplikace.

2.6.3 Fotogalerie

Velmi důležitou, ne-li nejdůležitější položkou u nálezu budou fotografie.

Uživatel musí mít možnost k nálezu přiložit libovolný počet fotografií. K těm přidá pak titulek, případně vhodný popis. Musí mít také možnost zvolit určitou fotografii jako hlavní, která se bude zobrazovat ve výpisu.

K prohlížení fotografií bude sloužit vyskakovací okno – lightbox³, s podporou navigace mezi fotkami bez nutnosti okno zavírat.

2.6.4 Sociální aspekt a sdílení

Každý mrcasník se chce svým nálezem pochlubit. Musí však mít možnost, které informace o nálezu chce sdílet.

Veřejně se zobrazí standardně jen název, popis, okolnosti nálezu, a fotogalerie. Místo nálezu bude standardně skryto. Bude možné nastavit jej jako veřejný, případně mapu zobrazit jen po zadání hesla. Žádost o heslo bude probíhat pomocí emailů, návštěvník klikne na odkaz, který vygeneruje autorovi nálezu email s informací, že po něm jiný uživatel chce heslo ke konkrétnímu nález. E-mail pak navíc obsahuje rovnou odkaz, na který stačí kliknout, aby bylo heslo odesláno na adresu žadatele.

³Javascriptový plugin, který umožňuje zobrazení obrázků v okně bez nutnosti změny adresy prohlížeče. Umožňuje pohodlnější prohlížení a lepší komfort. Jmenovitě pluginy: Lightbox, FancyBox, ColorBox.

Nález jako takový musí být možné úplně schovat před ostatními, výchozí nastavení však bude povolovat zobrazování nálezu i ostatním (i neregistrovaným – samozřejmě bez mapy). Dále bude možné nález zobrazit všem registrovaným hledačům.

U nálezu můžeme kliknutím na jediné tlačítko nasdílet odkaz na Facebook pro přátelé daného hledače. Tedy za předpokladu, že nález není nastaven jako neveřejný, v tomto případě by se uživateli zobrazila pouze informace o tom, že hledaná stránka neexistuje.

2.6.5 Komentáře

Přihlášení uživatelé budou moci u nálezů – ať už u svých, či ostatních – přidávat komentáře. Majitel nálezu dostane samozřejmě možnost tyto komentáře smazat, pokud se ukážou jako nevhodné.

Autor bude také moci komentáře u daného nálezu úplně zakázat. V případě, že dojde k zakázání komentářů ve chvíli, kdy již nějaké existují, nesmí dojít k jejich smazání, pro případ, že by se autor rozhodl je znovu zaktivovat.

Komentáře budou fungovat na principu odpovědí, návštěvník tedy bude moci vybrat, na který komentář chce odpovědět. Komentáře a odpovědi na ně budou řazeny hierarchicky tak, jak k sobě patří.

Autor nálezu bude mít možnost libovolné komentáře smazat. V případě smazání dojde ke skrytí i komentářů, které na původní odpovídají. Smazání komentáře bude doprovázeno hláškou, žádající si potvrzení, aby nedošlo k omylu.

2.7 Přidávání nálezu

Pro přidávání a editaci nálezu bude sloužit jednoduchý formulář. Obsluhu musí zvládnout každý, proto musí být veškerý postup co nejjednodušší a nejintuitivnější.

V horní části formuláře bude uživateli zobrazena mapa s výzvou, aby do ní klikl. Kliknutím do mapy na patřičné místo tak vyplní zeměpisnou délku a šířku pro daný nález. Umístění na mapě bude vyobrazeno pomocí *markeru*. Dále uživatel zadá veškeré údaje k nálezu, jako titulek, popis, okolnosti nálezu, datum a podobně.

Uživatel bude moci vybrat několik fotografií najednou a ty se ihned nahrají na server. Nepovedené lze případně rovnou smazat. Bude také podporována technologie *drag and drop*, kdy stačí soubory přesunout do okna prohlížeče.

Kapitola 3

Platforma a datový model

3.1 Výběr platformy

Výběr programovacího jazyka a databáze byl pro mě poměrně jasný. Vzhledem ke svým dřívějším zkušenostem s programovacím jazykem PHP a databázovým systémem MySQL padla má volba právě na tuto možnost.

Důležité je vybrat, zda začít programovat tzv. na zelené louce, pouze čisté PHP a čisté MySQL. Druhou variantou bylo zvolit nějaký framework. S frameworky sice nemám zkušenosti, ale nabízejí podstatně rychlejší a bezpečnější vývoj.

Výhody a nevýhody obou řešení se pokusím shrnout v následující části.

3.1.1 Vlastní řešení

S čistým PHP mám poměrně dost zkušeností. Jedná se o jednoduchý, v základu strukturálně programovací jazyk. Umožňuje i ve strukturální podobě napsat středně velké weby, avšak díky tomu, že nekopíruje žádný návrhový vzor a neobsahuje (v základu) žádnou ORM¹ a vzhledem k tomu, že se ve výsledku se bude jednat o poměrně rozsáhlou aplikaci, není toto řešení vhodné.

Výhody

- Mám s ním zkušenosti

Nevýhody

- Zdlouhavé – znovu vymýšlíme kolo
- Strukturovaně psaný kód je kombinací výkonného kódu, čtení z databáze a zobrazení
- Nebezpečné – Veškerý uživatelský vstup musí být escapován a je jednoduché na to zapomenout

¹Objektově relační mapování, umožňuje s databázovými záznamy pracovat jako s objekty v OOP.

- Špatná čitelnost strukturovaného kódu, prakticky neudržitelné
- V případě, že bychom se chtěli vyhnout nevýhodám uvedeným výše, dostali bychom se do stavu, kdy píšeme vlastní MVC Framework, což by vydalo na samostatnou diplomovou práci
- Možnost použít knihovny, které ale nemají žádný jednotný interface a každá řeší jen daný problém, složitá implementace do vlastního frameworku

3.1.2 PHP Framework

Druhou možností, jak jsem již uvedl, je vybrat si nějaký PHP Framework, ideálně respektující návrhový vzor Model – Viewer – Controller. V Controlleru se vybere akce, která využije Model, obsahující business logiku, a data, která pomocí metod v Modelu vypočte, dále předá na View. View se pak stará o korektní zobrazení získaných dat v HTML kódu.

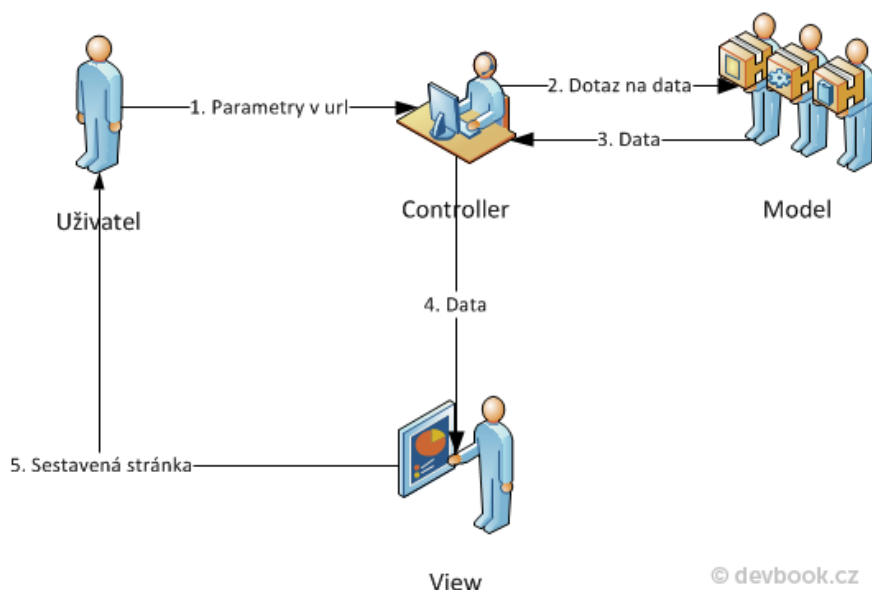
Framework by se dal definovat jako

„[Framework je...] softwarová struktura, která slouží jako podpora při programování a vývoji a organizaci jiných softwarových projektů. Může obsahovat podpůrné programy, knihovnu API, návrhové vzory nebo doporučené postupy při vývoji.“ [5]

Řešení webových aplikací za pomoci frameworku se v poslední době stává stále populárnější. Umožňuje kód, který se stále opakuje v každém projektu „vynechat“, protože je již obsažen ve frameworku a jeho metodách. Většina frameworků navíc užívá právě výše zmíněný návrhový vzor MVC, který přispívá k lepší organizaci, čitelnosti a udržitelnosti kódu. Toto řešení nám tedy umožňuje věnovat se řešení problému aplikace jako takového a nezabývat se znovu-vynalézáním kola. Často se setkávám s názorem, že frameworky vdechly PHP druhý život, vzhledem k tomu, že doplňují jazyk o zásadní věci, které si vývoj moderní aplikace žádá.

Výhody

- Rychlejší vývoj
- Méně kódu – píšeme skutečně pouze výkonný kód aplikace
- Lepší organizace, udržitelnost, univerzálnost a rozšiřitelnost



Obrázek 3.1: Diagram návrhového vzoru MVC [Zdroj: <http://devbook.cz>]

- Hezká URL²
- Netřeba samostatných knihoven, Framework je na většinu připravený, případně existují balíčky, které s frameworkem úzce spolupracují
- Integrovaný šablonovací systém³

Nevýhody

- Ze začátku složitější na pochopení
- Nutnost naučit se nad webovou aplikací přemýšlet od začátku úplně jinak

I přes to, že jsem s PHP frameworkem dosud nikdy nepracoval, rozhodl jsem se zvolit právě framework. Na takto složité aplikaci by byl čas strávený psaním čistě PHP mnohem větší, než doba, za kterou se rozumně naučím pracovat s frameworkem. Všechny výhody uvedené ve shrnutí mluví za sebe. Věřím, že aplikace takto dopadne mnohem lépe a umět s PHP frameworkem je dnes základ.

²URL znamená Uniform Resource Locator = jednoznačné určení zdroje. Je to způsob, jak jednoznačně zapsat umístění souboru na Internetu nebo na intranetu. URL je synonymem pro internetové adresy. V HTML se URL používá jak pro zacílení odkazů, tak pro načítání obrázků a podpůrných souborů. [6]

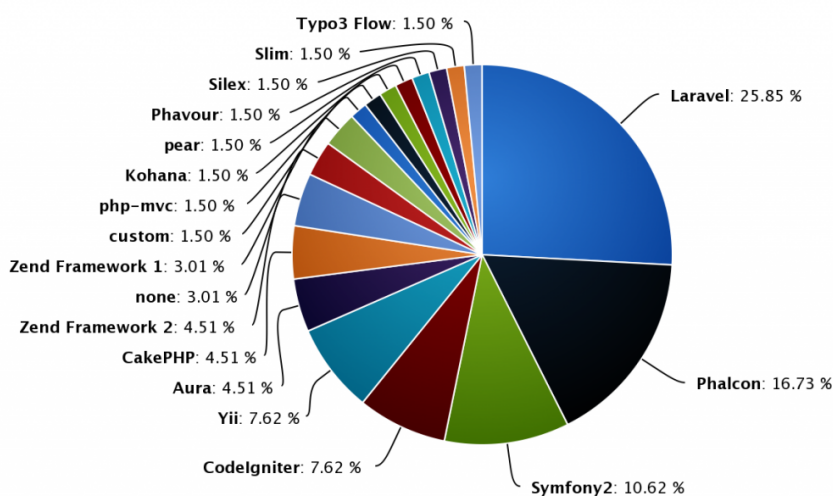
³System umožňující čitelnější zápis šablon, např Smarty.net

3.2 Výběr frameworku

Ze začátku je vhodné říci, že frameworků existuje ohromné množství. Každý umí trochu něco jiného, nebo je zaměřen na trochu jiné aplikace. Mým cílem bylo najít Framework, který se bude hodit na tento projekt a zároveň mě posune dále v mém profesním životě. Pokud si zvolím malý, neznámý Framework, nebude se mi zdaleka hodit tak, jako jeho používanější alternativa a riskuji malou podporu v dokumentaci a komunitě.

Ke srovnání jsem vybral následující frameworky. Mým cílem nebude vypisovat, co každý z nich umí, spíše se pokusím vyzdvihnout výhody a nevýhody oproti jeho konkurentům. Nemám se všemi samozřejmě zkušenosti, spíše se orientuji podle ohlasů na internetu, podle kterých jsem se řídil. Množina frameworků, ze kterých jsem vybíral, vznikla na základě popularity na internetu.

Framework popularity, end of 2013; SitePoint



Highcharts.com

Obrázek 3.2: Rozdělení frameworků podle popularity [Zdroj: sitepoint.com]

3.2.1 Nette Framework

Kvalitní Framework z dílny českého programátora Davida Grudla. Výhodou je velmi široké použití v České republice.

Výhody

- U nás velmi populární, aktivní česká komunita
- Školení v České republice

Nevýhody

- Ukončení vývoje – zastaralý
- Omezení pouze na českou komunitu, jelikož ve světě nikdo Nette nepoužívá

3.2.2 Symfony2

Jeden z nejpoužívanějších frameworků roku 2013, obsahuje ohromné množství balíčků, což může mít za následek občasnou pomalost. Říká se o něm, že je v základu nejvybavenější. [7]

Výhody

- Množství funkcí, modulů a rozšíření už v základu
- Vynikající ORM Doctrine2

Nevýhody

- Občas může být pomalejší [7]
- Subjektivně méně intuitivní syntaxe a použití

3.2.3 Zend Framework

Komplexní framework s dlouhou historií, avšak setkal jsem se s názorem, že verze 2 je poměrně nedospělá. Je spíše vhodnější pro opravdu velké, korporátní aplikace.

Výhody

- Obrovská základna uživatelů na celém světě
- Flexibilní
- Žádoucí v zahraničí

Nevýhody

- Pro naše účely příliš komplexní, robustní, složitý...
- Vhodný spíše pro velmi rozsáhlé aplikace
- Nemá šablonovací systém v základu, nutno zjišťovat kompatibilitu a vyhledávat vhodné řešení

3.2.4 Phalcon

Phalcon je od ostatních frameworků v tomto srovnání poněkud odlišný. Jedná se v podstatě o rozšíření PHP pomocí knihovny, což má za následek bleskovou rychlost. Na druhou stranu by mohl být problém s hostingem, kde by poskytovatel nemusel být s to instalovat knihovny do PHP navíc.

Výhody

- Nejrychlejší ve srovnání [8]

Nevýhody

- Kompilovaný (může být bráno jako výhoda)
- PHP Modul

3.2.5 Laravel

Velmi sympatický framework využívající některé součásti ze Symfony. Používá skvělý, intuitivní ORM Eloquent. Nejpoužívanější framework v roce 2013, velký trend pro 2014 [7].

Výhody

- Velmi oblíbený
- Velké množství rozšíření a modulů
- Eloquent ORM
- Jednoduchá syntaxe
- Velké množství tutoriálů
- Silný šablonovací systém Blade

Nevýhody

- Dokumentace není úplně kompletní, mohla by být vedena více do detailu

Po zhlédnutí několika video tutoriálů a vyzkoušení základního webu ve frameworkích, jsem se rozhodl pro Laravel. Vyhovuje mi syntaxí, adresářovou strukturou, ORM, má skvělou komunitní podporu a celkově je velmi oblíbený.

3.3 Datový model

Při použití frameworku je vždy dobré dbát na uspořádání databáze takovým způsobem, aby docházelo ke shodě s konvencemi a názvoslovím ve frameworku.

Každá tabulka by měla mít alespoň následující sloupce:

- `id` – automaticky se generuje, auto increment, primary key
- `created_at` – typ `datetime`, automaticky se vyplní při vytvoření modelu
- `updated_at` – typ `datetime`, automaticky se upraví při update modelu

Dále je vhodné při relacích nazývat sloupce, sloužící jako klíče do cizí tabulky, podle názvu tabulky. Tabulka je v množném čísle (samozřejmě anglicky) a klíč je pak `jednotnéčíslo_id`. Více vyplyne z popisu modelů aplikace v následující kapitole.

Vzhledem k tomu, že používáme objektově-relační mapování bude vhodnější použít popis modelů místo klasického ERA diagramu. [4]

Kapitola 4

Implementace

V této kapitole přiblížím čtenáři implementaci na vybraném frameworku Laravel. Budu se věnovat popisu modelů a jejich parametrů. Dále seznámím čtenáře s vybranými Controllery a konvencemi.

4.1 Modely

Všechny modely jsou mapovány pomocí ORM Eloquent2 a obsahují základní sloupce jako `id`, `created_at`, `updated_at`. Tyto jsou dány konvencí. Díky konvencím není v souboru modelu nutné definovat název tabulky (bere se množné číslo názvu modelu – model `User` bude používat tabulku `users`) a základní sloupce, jako je primární klíč, datum vytvoření či datum úpravy.

Modely by měly obsahovat veškerou *business logiku*¹.

4.1.1 Relace mezi modely

Jak jsem již uvedl výše, framework Laravel spolupracuje s vyspělým ORM Eloquent, který umožňuje mimo jiné vytvářet relace mezi modely rychle a efektivně. K dispozici jsou samozřejmě všechny základní relace jako 1:1, 1:N, M:N či polymorfní² Relace se definuje jako funkce modelu. V následujícím příkladu je definice 1:N relace mezi modely `User` a `Finding`.

Kód 4.1: Definice relace `findings` na modelu `User`.

```
public function findings()
{
    return $this->hasMany("Finding");
}
```

Inverzně můžeme definovat vztah z druhé strany z modelu `Finding` takto:

¹Veškeré výpočty, které jsou závislé pouze na daném modelu se musí odehrávat v něm a ne nikde jinde.

²Polymorfní relace umožňují spojit jeden model s několika dalšími, v tomto případě máme model `Image` spojený s modelem `Finding` a `User`. [9]

Kód 4.2: Definice relace author na modelu Finding.

```
public function author()
{
    return $this->belongsTo("User", "user_id");
}
```

Pokud pak budeme chtít získat ke konkrétnímu uživateli všechny jeho nálezy, učiníme tak následujícím příkazem, například v **controlleru**:

Kód 4.3: Získání nálezů modelu User.

```
$user = User::find($id);
return $user->findings;
```

4.1.2 User

Tento model slouží k reprezentaci jednoho registrovaného uživatele webu. Je spojen relací sám na sebe přes tabulku popisující přátelství.

Parametry:

- **id** – jednoznačný identifikátor modelu
- **name** – křestní jméno
- **surname** – příjmení
- **email** – e-mailová adresa
- **username** – uživatelské jméno zvolené při registraci
- **posts_per_page** – nastavení – počet nálezů na stránku
- **password** – otisk hesla uživatele
- **code** – potvrzovací kód pro aktivaci účtu
- **active** – zda je uživatel aktivovaný
- **fb_id** – pokud je vyplněno, slouží k párování lokálního účtu s facebookovým uživatelem
- **location** – město, odkud uživatel pochází
- **fb_user** – přezdívka, kterou má uživatel staženou z facebooku

Funkce

Model disponuje funkcemi například pro právu přátelství. Umí vrátit seznam přátel či seznam žádostí o přátelství. Dále umožňuje získávání relevantních dat jako profilový obrázek z Facebooku či seznam nálezů od přátel daného uživatele.

Relace

- `findings` – 1:N, vrací množinu nálezů pro daného uživatele
- `images` – Polymorfní, vrací množinu obrázků příslušných k danému uživateli.
- `friends` – Relace sama na sebe přes tabulku `friendships`.

4.1.3 Finding

Tento model slouží k uložení informací o nálezu.

Parametry:

- `id` – jednoznačný identifikátor modelu
- `title` – název nálezu
- `description` – popis nálezu
- `circumstances` – okolnosti nálezu
- `location_protection` – nastavení zobrazení mapy (`public` / `private`)
- `protection` – nastavení zobrazení nálezu (`public` / `registered` / `private`)
- `latitude` – údaj o zeměpisné šířce
- `longitude` – údaj o zeměpisné délce
- `date` – datum nálezu
- `allow_comments` – příznak, zda jsou zapnuté komentáře
- `category_id` – id kategorie, do které patří nález
- `favorite` – příznak, jestli byl nález zařazen mezi oblíbené

- `user_id` – identifikátor autora nálezu
- `password` – heslo mapě

Funkce

Model umí například na základě řetězce vrátit jeho souřadnice. Využívá k tomu statickou metodu `getLocation($locationString)`, která vrátí objekt získaný z Google Geocode Api.

Dále implementuje několik jednoduchých funkcí, například na vytisknutí miniatury či výpis data nálezu ve formátu, jaký se používá v České republice.

Relace

- `author` – 1:1, relace na autora nálezu
- `category` – 1:1, vrátí kategorii nálezu
- `comments` – 1:N, nalezne komentáře nejvyšší úrovně
- `images` – polymorfní, vrátí instance modelu `Image` spojené s patřičným nálezem

4.1.4 Controllery

Controllery se starají o to, aby uživatel dostal na základě svého požadavku správná data z databáze a byla mu zobrazena v odpovídajícím pohledu (*view*). Controller je třída, která obsahuje několik akcí. V jednom uživatelském požadavku je typicky spuštěna právě jedna akce z určitého controlleru. O to, která akce bude spuštěna, se stará router - soubor `routes.php`.

`routes.php`

Jak jsem již nastínil, router se stará o to, aby konkrétně strukturovanému URL dotazu odpovídala určitá akce v konkrétním controlleru. Jednoduchá cesta (z angl. *route*) vypadá například následovně:

Kód 4.4: Jednoduchá cesta v routeru

```
Route::get("nalezy/novy", array(
    'as' => "finding-insert",
    'uses' => "FindingController@getInsert"
));
```

Zápisu výše se říká „pojmenovaná cesta“. Jméno má z důvodu jednoduchého odkazování, při vytvoření odkazu na vytvoření nového nálezu je pak možné zadat jen `URL::route(finding-insert)`. Můžeme pak použít hezká URL (zde `nalezky/novy`), aniž bychom museli mít controller pojmenovaný česky.

Druhý způsob, jaký používám k vytvoření cest je tzv „RESTful Controller“. Jedná se o způsob, kdy v routeru určíme, jaký controller k prvnímu segmentu URL použít, akce se pak zvolí sama na základu konvence.

Zde je příklad RESTful Controlleru:

Kód 4.5: RESTful controller

```
Route::controller("komentar", "CommentController");
```

Pokud v tomto případě přistoupím metodou `GET` na adresu `komentar/remove/13`, spustí se akce `getRemove` s parametrem „13“ z controlleru `CommentController`.

Výhoda RESTful Controlleru spočívá v tom, že nemusím pro každou akci nutně vytvářet další cestu v `routes.php`. Výhodou pojmenovaných cest jsou pak hezká URL. Oba přístupy v práci kombinuji.

Filtry v `routes.php`

Filtry jsou vlastně určité podmínky, které musí být splněny, aby byla cesta platná. Používám například filtr pojmenovaný `auth`, který zabezpečí, že všechny akce, ke kterým je třeba přihlášení, budou spuštěny pouze v případě, že uživatel je skutečně přihlášen.

Seznam vlastních controllerů

- AccountController
 - vytváření účtů
 - odesílání potvrzovacího kódu emailem
 - obslužení aktivace účtu
 - zobrazení / obslužení formuláře pro přihlášení
 - obsluha odhlášení
 - obsluha přihlášení / registrace facebookem
- CommentController
 - obsluha vkládání formuláře
 - obsluha mazání

- FindingController
 - zobrazení / validace a obsloužení formuláře pro nový nález
 - zobrazení / validace a obsloužení formuláře pro úpravu
 - obsluha mazání nálezu
 - zobrazení detailu nálezu
 - filtrace a výpis nálezů
 - obsluha požadavku o heslo
 - obsluha zadání hesla k nálezu
 - vrácení komentářů
 - zobrazení Google mapy k nálezu
- FriendshipController
 - vytváření žádostí o přátelství
 - potvrzování žádostí o přátelství
 - zakazování přátelství
- ImageController
 - upload obrázků
 - výpis formuláře pro editaci
 - obsluha editace titulku a popisu
 - mazání obrázku
- ProfileController
 - zobrazení profilu uživatele
 - zobrazení a obsloužení formuláře pro změny v profilu

4.2 Šablona

Šablona aplikace byla vytvořena speciálně pro účel této práce a nebyla odnikud stažena. Při návrhu prvků uživatelského rozhraní mi byl nápomocný kolega ³, který mi pomohl navrhnout obsahové boxy a záhlaví webu.

³Patrik Vogl – <http://hellodesign.cz>

4.2.1 Použité technologie a knihovny

Front-end⁴ aplikace je vytvořen v jazyce HTML 5 s použitím kaskádových stylů CSS verze 3. Pro dynamiku webu je načítána JavaScriptová knihovna jQuery⁵ ve verzi 1.10.1.

Kostra webu stojí na volně dostupném CSS frameworku Twitter Bootstrap⁶, z něhož jsou používány pouze prvky jako responsivní mřížka, tlačítka a informační hlášky pro uživatele. Bootstrap jako takový je mnohem silnější nástroj, jeho potenciál v této práci zdaleka nevyužívám naplno.

Na moderním webu se velmi často používají všemožné jQuery knihovny. Já používám jen ty nejnnutnější..

Colorbox verze 1.5.5⁷, které umožňuje zobrazení fotografie ve vyskakovacím okně s patřičnou navigací. Kód jsem modifikoval tak, aby bylo možné u fotografie zobrazit popisek.

jQuery Upload File Plugin verze 3.1.0⁸ umožňující jednodušší implementaci dynamického, plně ajaxového uploadu fotografií. Skript řeší pouze uživatelskou část nahrávání a serverová část je napsaná mnou na míru aplikaci.

Zebra Datepicker verze 1.8.7⁹ – plugin pro zobrazení kalendáře po kliknutí do vstupu určeného pro datum.

Font Awesome verze 4.0.3¹⁰ – vektorové ikonky pro volné použití. Jedná se vlastně o druh písma, který obsahuje speciální znaky – ikonky.

4.3 Popis funkčnosti

4.3.1 Přihlášení a registrace

Pro plné použití aplikace je nutné využít registrovaného účtu v aplikaci. Uživatel má možnost zvolit si mezi standardní registrací, a přihlášením přes facebook.

⁴To, co uživatel vidí a s čím pracuje

⁵<http://jquery.com>

⁶<http://getbootstrap.com>

⁷<http://www.jacklmoore.com/colorbox>

⁸<http://hayageek.com/>

⁹<http://stefangabos.ro/jquery/zebra-datepicker/>

¹⁰<http://fortawesome.github.io/Font-Awesome/>

Přihlašte se pomocí Facebooku

 Přihlášení / Registrace přes Facebook

nebo

Přihlášení

Uživatelské jméno:

Heslo:

Zůstat přihlášen Přihlásit se

Registrace

Nemáte u nás účet?

Registrovat!

Obrázek 4.1: Stránka pro přihlášení / registraci

K uživatelskému účtu se také váže fotografie uživatele. Tu je možné v případě klasické registrace nahrát v nastavení profilu, případě registrace přes facebook je tato volba skryta a použije se profilový obrázek. Obrázek z facebooku se neukládá nikam na lokální server, je vždy aktuální podle toho, jaký obrázek má uživatel zrovna nastavený jako profilový.

Standardní přihlášení

Pokud se uživatel rozhodne využít možnosti klasické registrace, po vyplnění osobních údajů mu na e-mail bude odeslán odkaz s aktivačním kódem. Na odkaz je nutné před přihlášením přijít, v opačném případě nebude možné se na účet přihlásit. Uživatel je o této skutečnosti informován ihned po registraci, tak i v případě pokusu o přihlášení v případě neaktivního účtu.

Aktivační kód je generován náhodně, kód v URL musí odpovídat kódu uloženého u uživatele v databázi. Pokud to tak je, uživateli je nastaven aktivační **flag** na jedničku. Takový uživatel se pak může přihlásit.

Ověřování pomocí e-mailu je nutností, protože e-mail je používán pro zaslání požadavků o heslo k nález. E-mail tedy nemůže být fiktivní.

Přihlášení přes Facebook

Přihlašování přes Facebook je v dnešní době velmi populární. Umožňuje zrychlit celý proces registrace a facebookové **Graph API** ¹¹ je silný nástroj jak z Facebooku získat informace nutné pro registraci.

¹¹API pro získávání informací o uživateli – <https://developers.facebook.com/docs/graph-api>

O obsluhu facebookového přihlášení se stará `AccountController`. Využívám balíček pro Laravel – FacebookConnect, který obaluje připojení ke GraphAPI do přehledné třídy.

Pokud se povede získat uživatele z Graph API – uživatel potvrdí přístup aplikaci Mrcaseni.cz k jeho profilu – zkusíme najít uživatele v databázi, který obsahuje stejné Facebook ID, jako aktuální. Pokud jej nalezneme, přihlásíme a přesměrujeme jej na domovskou obrazovku.

V případě že není uživatel nalezen, dojde k výjimce. Tu zachytím pomocí bloku `catch` a ověříme, jestli už náhodou neexistuje uživatel se stejným e-mailem jako je ten z Facebooku. Tato situace může nastat v případě, že se někdo zaregistruje klasickým způsobem a poté se přihlásí pomocí facebooku. E-maily nemohou být duplicitní a proto se k uživateli uloží jeho Facebook ID a příště se může přihlásit.

Pokud ani v tuto chvíli není e-mail nalezen, je jasné, že jde o nového uživatele. Vytvoříme mu tedy nový uživatelský profil založený na informacích dostupných z Facebooku. Uživatele přihlásíme a navrátíme na úvodní obrazovku.

4.3.2 Výpis a filtrace nálezů

Webová aplikace obsahuje sekci *Nálezy*, která sdružuje veškerou funkčnost pro procházení a vyhledávání mezi nálezy.

V horní části stránky si uživatel může zvolit z jaké množiny nálezů chce filtrovat. Má na výběr mezi *Svémi nálezy*, *Nálezy přátel* a *Všemi nálezy*.

Mapa nálezů

Pod výběrem se nachází mapa s informací, kde jsou aktuálně filtrované položky nalezené. U každého nálezu je ověřováno, zda má aktuálně přihlášený uživatel právo vidět umístění nálezu. Jak je toto řešené popíši v další části práce.

Nálezy na mapě jsou přehledně rozděleny podle ikonek, kdy každá ikonka značí kategorii, do které daný předmět spadá. Pod mapou je pak zobrazena legenda – ikonky, které po najetí zobrazí název dané kategorie a informace o tom, že červené ikonky jsou označeny jako oblíbené.

Mapu je možné pomocí tlačítka *lupy* v záhlaví boxu zvětšit na celou obrazovku pro pohodlnější orientaci.

Na nález v mapě je možné také kliknout, poté je uživateli zobrazena vyskakovací bublina s názvem, zkráceným popisem, datem nálezu a kategorií. Z bubliny pak může rovnou přejít na detail nálezu.

Filtrování

KATEGORIE:

- Válečné předměty
- Mince
- Šperky a bižuterie
- Předměty z domácnosti
- Neznámý
- Ostatní

OBLÍBENÉ:

- Pouze oblíbené

VYBĚR DATA

Datum od:

Datum do:

SEŘADIT PODLE

Datum vložení

Sestupně

LOKALIZACE

V okolí [km]:

Od místa:




Filtrovat

Moje nálezy | Nálezy přátel | **Všechny nálezy**

Mapa nálezů

Seznam všech nálezů

Zobrazujete všechny veřejné nálezy. Pro zobrazení svých nálezů klikněte [ZDE](#).

		
Ozdobný knoflík	Svátostka	Žaváží kváham
Šperky a bižuterie	Šperky a bižuterie	Ostatní
datum: 1.9.2013	datum: 1.9.2013	datum: 23.8.2013
Autor: Jiří Vyhňálek	Autor: Jiří Vyhňálek	Autor: Jiří Vyhňálek

Obrázek 4.2: Filtrování nálezů

Přeska od opasku

Datum: neuvedeno | Kategorie: Předměty z domácnosti

Zdravím všechny slušné mrcasníky a samozřejmě i mrcasnice. Před nedávnem, když u nás mrzlo jen praštel...

[Zobrazit detail](#)

Obrázek 4.3: Informace o nálezu po kliknutí do mapy

Seznam nálezů

Hlavním panelem pro výsledky vyhledávání / filtrace je místo pod mapou. Zde se zobrazují nálezy, které vyhovují parametrům vyhledávání.

V případě, že uživatel nelistuje mezi svými nálezy je o této skutečnosti informován hláškou nad výsledky, aby nedošlo ke zmatkům.

Výsledek vyhledávání se skládá z matice nálezů, pod kterou je zobrazeno

stránkování. Stránkování probíhá dynamicky pomocí Ajaxu bez nutnosti obnovit dokument.

Filtr

Důležitou součástí výpisu nálezů je samotný filtr. Umožňuje filtrovat podle mnoha skutečností a zobrazené výsledky zobrazit na mapě a v seznamu.

Prvním kritériem je kategorie. Výběr je realizován pomocí checkboxů ¹², kdy uživatel zaškrtná kategorie, které chce do vyhledávání zahrnout.

V druhém kroku si může zvolit, zda chce zobrazit jen nálezy, které jejich autor označil jako oblíbené.

Dále je možnost vyhledávat nálezy podle data nalezení, zadávají se dvě data – období od do. Je tak možné vybrat si například všechny nálezy, které byly nalezeny minulé léto.

Nesmí samozřejmě chybět možnost řadit výsledky podle vlastního uvážení. Existuje možnost řadit podle data vložení, data nalezení či názvu. Vše vzestupně či naopak sestupně.

Poslední položkou pro filtraci je možnost vybírat nálezy podle místa nálezů. Uživatel zadá rádius v kilometrech a místo, kolem kterého chce hledat. Toto kritérium je pak zahrnuto ve výsledcích.

Lokalizace probíhá pomocí *Google Geocoding API* ¹³. Zadané místo se lokalizuje a lokace je pak v SQL dotazu zahrnuta pomocí následující podmínky:

Kód 4.6: Podmínka pro vyhledávání umístění

```
[... and] acos(sin(radians($location->lat)) * sin(radians(latitude)) + cos(
  radians($location->lat)) * cos(radians(latitude)) * cos(radians(
  longitude) - (radians($location->lng)))) * 6371 <= $radius
```

`$location` je objekt obsahující zeměpisnou délku a šířku lokalizovaného místa pomocí *Google Geocoding API*. `$radius` je pak okruh hledání od daného místa v kilometrech.

Výkonný kód v pozadí filtrace

Ve chvíli, kdy návštěvník vyplní možnosti filtru a stiskne tlačítko *Filtrovat*, všechny podmínky se ve třídě `FindingController` postupně zpracují a vloží do SQL dotazu jako podmínky. Berme jako příklad následující modelovou situaci.

¹²Zaškrťávací pole, je možné vybrat několik možností

¹³<https://developers.google.com/maps/documentation/geocoding/>

Zobraz všechny mé nálezy z kategorií válečné předměty, mince a šperky a bižuterie, nalezené v období mezi 8. 3. 2014 a 10. 4. 2014, v okolí 100 km od Plzně a seřaď je podle data nálezu od nejnovějšího.

Kód 4.7: SQL řetězec vyhledávající podle filtru

```
select * from 'findings' where 'findings'. 'deleted_at' is null and (
  protection in ("public", "registered") or user_id = 5) and 1 and
  category_id in ("1", "2", "3") and DATEDIFF(date, '2014-03-08') >= 0 and
  DATEDIFF('2014-05-14', date) >= 0 and acos(sin(radians(49.7384314)) *
  sin(radians(latitude)) + cos(radians(49.7384314)) * cos(radians(latitude))
  * cos(radians(longitude) - (radians(13.3736371)))) * 6371 <= 100
order by 'date' desc
```

Aby dotaz fungoval co nejrychleji, sloupce, podle kterých může docházet k filtrování jsou opatřeny indexy. To zabezpečí rychlejší vrácení výsledků. [10]

V dalším kroku se zjistí, zda přišel požadavek přes Ajax. Pokud ano, znamená to, že uživatel pouze přelistoval výsledky, vrátíme pouze požadovanou stranu a výsledky pak necháme zobrazit JavaScript na straně klienta. Pokud došlo ke standardnímu přístupu, zobrazíme kompletní novou stránku.

4.3.3 Detail nálezu - umístění

Umístění u nálezu má tři možné stavy. Může být veřejné, soukromé a nebo přístupné pouze pod heslem.

V případě nastavení *soukromého* umístění je v případě prázdného hesla zobrazena pouze hláška *Litujeme, k zobrazení nálezu nemáte dostatečná oprávnění*.

Pokud se autor nálezu rozhodne zobrazit mapu konkrétní osobě, například poslat odkaz svému dobrému známému, může k soukromému nálezu nastavit heslo. U takového nálezu je pak místo mapy zobrazen formulář vyzývající k zadání hesla, případně možností požádat o zaslání hesla na e-mail.

Odeslání žádosti je po odeslání oznámeno a žádající musí čekat, zda se autor rozhodne o místo nálezu se podělit. Pokud ano, heslo bude žadateli zasláno na jeho e-mail. Autor nálezu klikne pouze na odkaz, který mu přijde e-mailem. Před odesláním hesla se samozřejmě ověří, zda má aktuálně přihlášený uživatel právo heslo zaslat – tedy zda je autorem nálezu.

Po zadání správného hesla k nálezu se vytvoří cookie, jejíž název je `access_finding_ID`. Hodnota této cookie je pak zašifrované ID uživatele a jeho hesla, který získal právo vidět místo nálezu. Tím je zajištěno to, že i když je cookie uložena v prohlížeči, po přihlášení jiného uživatele nedojde k zobrazení místa nálezu.

Kód 4.8: Funkce zjišťující oprávnění k zobrazení umístění

```

public function canShowLocation()
{
    $scan = false;

    //Is current user's finding
    if(Auth::check()){

        if(Auth::user()->id == $this->author->id){
            $scan = true;
        }

        //cookie with password exists
        if($this->password && (Cookie::get("access_finding_". $this->
            id) == Auth::user()->id.$this->password)){
            $scan = true;
        }
    }

    //or finding is public
    if($this->location_protection == "public"){
        $scan = true;
    }

    return $scan;
}

```

4.3.4 Detail nálezu - komentáře

Komentáře k nálezům jsou řešeny hierarchicky. Je tak možné odpovídat přímo na konkrétní komentář a skrývat – zobrazovat si jednotlivé větve konverzace. To umožňuje v komentářích udržovat pořádek.

Každý komentář má tedy uloženo číslo nadřazeného komentáře. Pokud je toto číslo rovné nule, jedná se o komentář nejvyšší úrovně.

Před samotným vložením komentáře se ověří, zda existuje komentář, na nějž odpovídáme (pokud to není nula, což značí nový komentář nejvyšší úrovně). Odstraní se přebytečné mezery na konci a na začátku řetězce, nahradíme odřádkování za tag `
`, nastavíme jako autora aktuálně přihlášeného uživatele a uložíme komentář do databáze. Toto vše se děje dynamicky přes Ajax, nakonec vrátíme nový seznam komentářů, již obsahující aktuálně vložený a pomocí JavaScriptu jej dosadíme do dokumentu.

Generování seznamu komentářů je prováděno rekurzivně, kdy jako aktuální seznam je vždy nastaven seznam podřazených komentářů.

4.3.5 Editace nálezu

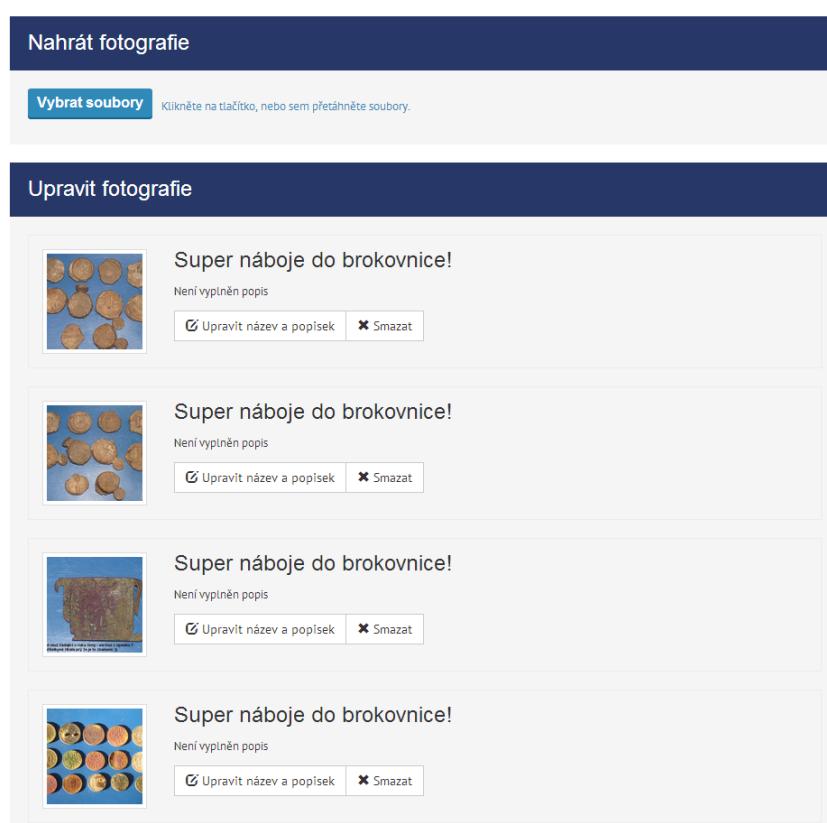
Formulář editace nálezu je přehledně rozdělen do několika sekcí.

V první sekci uživatel vybere místo, kde byl nález objeven. Výběr místa probíhá pomocí klikání do mapy, po kliknutí se uloží poslední vybrané souřadnice.

Ve druhé sekci autor může zadat název nálezu, popis a okolnosti. Následují sekce *Podrobnější údaje* – datum, kategorie, oblíbený a *Soukromí* - kde si uživatel nastavuje viditelnost pro ostatní a heslo k lokalitě.

4.3.6 Editace fotogalerie

Pro možnost editace fotogalerie je třeba znát již ID nálezu, ke kterému fotogalerii vkládáme. Z tohoto důvodu je editace galerie implementována jako samostatná stránka.



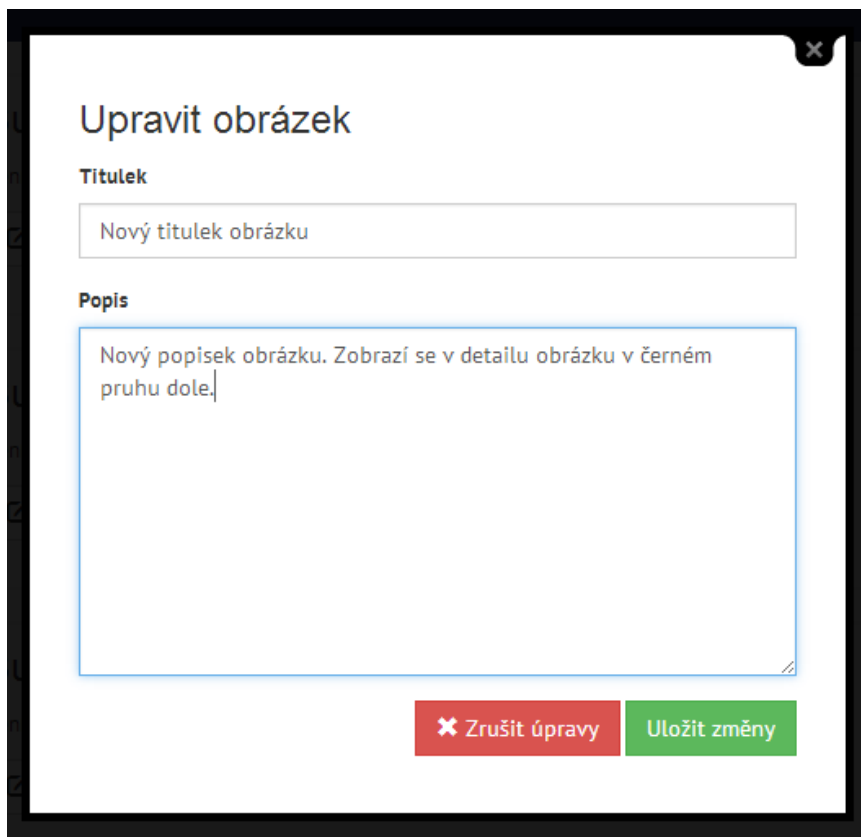
Obrázek 4.4: Editace fotogalerie

V horní části stránky je zobrazen formulář pro výběr souborů. Je možné vybrat několik obrazových souborů najednou, není nutné je vybírat postupně.

Po vybrání souborů dojde k ajaxovému odeslání souborů na server. U každého souboru se rozhodne, zda jde o typ obrázku k nálezu, či profilu.

Podle toho se pak volí cílová složka, kam se obrázek nahraje a typ, který se uloží do databáze. V případě že dojde ke správnému přesunutí obrázku do cílové složky, uložíme informaci o souboru také do databáze. Titulek obrázku nastavíme jako název nálezu, či jméno uživatele (v závislosti na typu). Popisek zůstane prozatím nevyplněný a je možné jej doplnit uživatelsky.

Po nahrání obrázku je seznam obrázků v galerii okamžitě aktualizován. Uživatel tak rovnou může obrázkům změnit popisek či titulek. Pro tyto účely slouží vyskakovací okno, kam může informace zadat.



Upravit obrázek

Titulek

Nový titulek obrázku

Popis

Nový popisek obrázku. Zobrazí se v detailu obrázku v černém pruhu dole.

X Zrušit úpravy **Uložit změny**

Obrázek 4.5: Změna titulku a popisku u obrázku

Kapitola 5

Ohlasy a návrhy na vylepšení

S žádostí o testování aplikace jsem oslovil skupinu několika desítek hledačů. Někteří přijali nový nápad s chutí, jiní se nápadu vyzkoušet něco nového báli a nebylo jim úplně jasné, jak například funguje sdílení umístění, nebo celého nálezu. Zhruba s tuctem lidí jsem komunikoval přímo osobně a rozebíral s nimi, co se jim na webu líbí, co by například změnili, nebo vylepšili. V době odevzdání této práce funguje již aplikace zhruba měsíc on-line a hledá si své pravidelné uživatele.

5.1 Co se uživatelům líbí

5.1.1 Přehlednost

Poměrně velká část uživatelů pochválila přehlednost webu, kdy se snažím vyvarovat zbytečným prvkům, ale ty důležité jsou ihned jasně vidět. Na druhou stranu některým z nich chybí styl, který by více evokoval historii, stáří a podobně. To však mým cílem nebylo, odporuje to myšlence jednoduchého, čistého a přehledného designu.

5.1.2 Mapa a filtrování

Nejoblíbenější část webu. Lidem se líbí možnosti filtrace a hlavně přehledné zobrazení na mapě. Tato část sklízí poměrně velký úspěch.

5.1.3 Dělení na veřejnou a neveřejnou sekci

Možnost ovlivnit, kdo daný nález uvidí je velmi dobře přijata. Líbí se možnost nechat si zobrazit mapu pod heslem, případně celý nález skrýt před zraky ostatních.

5.2 Návrhy na vylepšení

5.2.1 Barevné téma

Mým záměrem při výrobě šablony a vzhledu byla jednoduchost, nicméně setkávám se s názory, že aplikace by zasloužila více barev. Některé lidi pak mate modrá lišta v záhlaví, která jim připomíná Facebook. Pokud se připomínky objeví od více uživatelů, možná by stálo za zvážení přebarvení webu.

5.2.2 Seznámení nově příchozího

Pro nově příchozího uživatele může být první dojem z aplikace trochu zavadějící. Uvidí pouze seznam posledně přidanych nálezů s výzvou k registraci. Bylo by tedy vhodné nově příchozího návštěvníka lépe seznámit s cílem aplikace a přiblížit mu, jak ji začít používat.

5.3 Návrhy na rozšíření

5.3.1 Diskusní fórum

Mezi návrhy na rozšíření aplikace se vyskytlo například diskusní forum. Naprogramovat diskusní forum není triviální záležitost, ale naštěstí existuje spousta open-source řešení, jako nejrozšířenější uvedu například *phpBB*. Nutností by byla alespoň částečná modifikace spočívající ve spojení uživatelských účtů a propojení s infrastrukturou aktuální aplikace. Naprogramovat vlastní fórum by bylo vhodnější, nicméně funkcionalita by byla značně omezenější. Jednalo se o opravdu rozsáhlý projekt.

5.3.2 Anketní systém

Ankety by mohli pomoci začátečníkům s názorem zkušených hledačů. Otázky typu *Jaký detektor používáte?*, či *Používáte malý, přesňující detektor – pinpointer?* by mohly začátečníkům usnadnit orientaci ve světě hledačů.

5.3.3 Články

Články, potažmo redakční systém se zdá být vhodným rozšířením. Umožnilo by to psát například recenze či zkušenosti s konkrétním modelem detektoru. Anketní systém uvedený výše by pak mohl využívat právě články pro zobrazení ankety, jejího průvodního textu a komentářů.

Každý uživatel by měl mít možnost psát články, musela by ale fungovat určitá skupina moderátorů, která by ty zajímavé zobrazovala v určitém rozcestníku. Bez tohoto opatření by nejspíše brzy web zaplavily roboti a nevyžádaná reklama. Tento nápad tedy stojí na nalezení skupiny lidí, která by se chtěla údržbě článků věnovat.

5.3.4 Bazar

Tak jako u každého koníčka i hledači pokladů občas vyžadují povyměňovat zařízení. K tomu by mohl sloužit bazar, kam by nálezci umísťovali věci, které již nevyužijí, případně poptávali od ostatních.

Kapitola 6

Závěr

Projekt komunitního webu pro hledače pokladů byl pro mě úplně něčím novým. Přestože tvorbě webů se věnuji již několik let, specializuji se na HTML šablony, využití CSS a JavaScriptu. Mám za sebou několik projektů v PHP, nicméně nic tak rozsáhlého, jako je aplikace *Mrcaseni.cz*, jsem zatím neprogramoval. Také z důvodu rozsahu práce jsem se rozhodl využít PHP framework Laravel, který mi umožnil napsat aplikaci bezpečně, čistým kódem a dobře udržitelně.

V zadání jsem si dal za úkol zanalyzovat požadavky na aplikaci. Z několika rozhovorů vyplynuly základní požadavky na to, jaké informace by měl nález obsahovat, jak by mělo fungovat filtrování, jak rozhodovat, kdo uvidí umístění nálezu a podobně.

Implementace webu probíhala na základě analýzy a průběžných připomínek. Všechny požadavky, které jsem si na začátku stanovil, se podařilo splnit. Věřím že se mi podařilo aplikaci na napsat dostatečně intuitivně na to, aby její použití pochopil i průměrně zdatný uživatel osobního počítače.

Po naprogramování aplikace byly sbírána odezva od cílové skupiny, kdy jsem se uživatelů ptal, co se jim na aplikaci líbí, co by vylepšili a co by se jim do budoucna líbilo, aby aplikace uměla. Tento výstup mi poskytl cenný základ pro případné další rozšiřování a vylepšování aplikace.

Další vývoj aplikace bude záležet na úspěchu aktuální verze. Pokud si program najde svou stabilní základnu uživatelů, věřím že bude možné na práci navázat a její podstatně rozšířenou a vylepšenou verzi objahovat jako diplomovou práci.

Literatura

- [1] *Facebook Statistics* **Statistic Brain** [on-line]
<http://www.statisticbrain.com/facebook-statistics/>, 1. 1. 2014
- [2] *VRÁNA, Jakub. 1001 tipů a triků pro PHP.* Vyd. 1. Brno: Computer Press, 2010, 456 s. ISBN 978-80-251-2940-1.
- [3] *Laravel – dokumentace* **Laravel Documentation** [on-line]
<http://laravel.com/docs>, verze 4.1
- [4] *SCHLOSSNAGLE, George. Pokročilé programování v PHP 5.* Vyd. 1. Brno: Zoner Press, 2004, 640 s. ISBN 80-868-1514-5.
- [5] *Programujte.com* **PHP frameworky** [on-line]
<http://programujte.com/clanek/2008022000-php-frameworky/>,
Jan Škrášek, 21. 2. 2008
- [6] *Jakpsatweb.cz* **URL** [on-line]
<http://www.jakpsatweb.cz/html/url.html>, Dušan Janovský, jakpsatweb.cz
- [7] *sitepoint.com* **Best PHP Frameworks for 2014** [on-line]
<http://www.sitepoint.com/best-php-frameworks-2014/>, Bruno Skvorc, 28. 12. 2013
- [8] *sitepoint.com* **PhalconPHP: Yet Another PHP Framework?** [on-line]
<http://www.sitepoint.com/phalconphp-yet-another-php-framework/>,
Alexander Cogneau, 29. 7. 2013
- [9] *Laravel – dokumentace* **Eloquent ORM** [on-line]
<http://laravel.com/docs/eloquent#polymorphic-relations>,
verze 4.1
- [10] *GILMORE, Jason W. Velká kniha PHP 5 MySQL* **Velká kniha PHP 5 MySQL: kompendium znalostí pro začátečníky i profesionály.** Vyd. 1. Brno: Zoner Press, 2005, 711 s. ISBN 80-868-1520-X.

Seznam obrázků

3.1	Diagram návrhového vzoru MVC [Zdroj: http://devbook.cz] .	13
3.2	Rozdělení frameworků podle popularity [Zdroj: sitepoint.com] .	14
4.1	Stránka pro přihlášení / registraci	25
4.2	Filtrování nálezů	27
4.3	Informace o nálezu po kliknutí do mapy	27
4.4	Editace fotogalerie	31
4.5	Změna titulku a popisku u obrázku	32