

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta informačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2016

Miroslav Pospíšil



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

WEBOVÁ APLIKACE PRO VYHLEDÁVÁNÍ PANORAMAT

WEB APPLICATION FOR PANORAMA SEARCH

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

VEDOUCÍ PRÁCE
SUPERVISOR

MIROSLAV POSPÍŠIL

Ing. JAN BREJCHA

BRNO 2016

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií

Akademický rok 2015/2016

Zadání bakalářské práce

Řešitel: **Pospíšil Miroslav**

Obor: Informační technologie

Téma: **Webová aplikace pro vyhledávání panoramat**
Web Application for Panorama Search

Kategorie: Uživatelská rozhraní

Pokyny:

1. Proveďte rešerši existujících aplikací pro zobrazování panoramat.
2. Seznamte se s technologiemi, které budete potřebovat pro vývoj - např. Flickr API pro načítání fotek, Google Maps API pro zobrazování polohy panoramat.
3. Navrhněte architekturu a uživatelské rozhraní aplikace.
4. Implementujte webovou aplikaci umožňující uživateli nahrát/vybrat fotografii, označit křivku horizontu a dle této křivky vyhledat nejvíce podobná panoramata. Vizualizujte polohu nalezených panoramat. Pro samotné vyhledávání panoramat využijte hotovou implementaci, kterou dodá vedoucí.
5. Proveďte uživatelské testování.
6. Diskutujte možnosti dalšího vývoje.

Literatura:

- SAURER, Olivier, et al. Image Based Geo-localization in the Alps. *International Journal of Computer Vision*, 2015, 1-13.
- UNGER, Russ; CHANDLER, Carolyn. *A Project Guide to UX Design: For user experience designers in the field or in the making*. New Riders, 2012.
- RUMBAUGH, James; JACOBSON, Ivar; BOOCH, Grady. *Unified Modeling Language Reference Manual, The*. Pearson Higher Education, 2004.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Brejcha Jan, Ing.**, UPGM FIT VUT

Datum zadání: 1. listopadu 2015

Datum odevzdání: 18. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
602 00 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstrakt

Cílem této bakalářské práce je vytvořit webový front-end, který umožňuje lokalizaci místa na fotografii v horách. Fotografie pro lokalizaci mohou být do aplikace nahrány přímo, nebo mohou být použity fotografie z webové služby Flickr. Služba Flickr je integrována v aplikaci pomocí Flickr API. Pro lokalizaci vybrané fotografie je využito horizontu hor. Pro jeho poloautomatickou detekci s asistencí uživatele je implementováno speciální uživatelské rozhraní s využitím segmentačního algoritmu GrabCut. Lokalizaci zajišťuje serverový back-end, se kterým je navrhovaná aplikace integrována. Výsledky lokalizace pro detekovaný horizont jsou webovou aplikací vizualizovány ve formě zarovnaných křivek horizontu s jejich pozicí zobrazenou pomocí Google Maps API. Výsledná, plně funkční aplikace byla testována moderovaným testem s uživateli. V tomto testování si aplikace vedla velmi dobře. I přes nedostatky, které test identifikoval, je aplikace snadno použitelná.

Abstract

The aim of my bachelor thesis is to create a web front-end, which provides mountain localisation of images. Images can be uploaded directly to the application or you can use images from the Flickr web service. Flickr is integrated in the application by Flickr API. Localisation of selected image is based on mountain horizon. User-aided semi-automatic horizon detection is approached by special user interface utilising the GrabCut algorithm for segmentation. Localisation is provided by server back-end which is connected and integrated with the designed application. Results for the detected horizon are visualised in a form of aligned horizon curves with position displayed using the Google Maps API. The final fully functional application was tested by moderated usability test with users. Application in this test performed well. Despite several shortcomings detected by the test the application was easy to use.

Klíčová slova

aplikace, horizont, panorama, křivka horizontu, fotografie, testování, uživatelské rozhraní, flickr, locate, vyhledávání, lokalizace

Keywords

application, horizon, panorama, curve horizon, photo, testing, user interface, flickr, locate, searching, localisation

Citace

Miroslav Pospíšil: Webová aplikace pro vyhledávání panoramat, bakalářská práce, Brno, FIT VUT v Brně, 2016

Webová aplikace pro vyhledávání panoramat

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jana Brejchy.

.....

Miroslav Pospíšil

11. května 2016

Poděkování

Děkuji Ing. Brejchovi za pomoc, rady a podporu při vypracování této práce.

© Miroslav Pospíšil, 2016.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	3
1.1 Rešerše existujících řešení	4
1.2 Motivace	5
2 Návrh aplikace	6
2.1 Definice požadavků	6
2.1.1 Selektce skupin uživatelů	7
2.2 Výběr technologií	7
2.3 Návrh architektury	8
2.4 Návrh uživatelského rozhraní	8
2.5 Uživatelské testování designu	8
2.5.1 Výběr uživatelů	9
2.5.2 Testování mockupu	9
2.5.3 Vyhodnocení testování	9
3 Implementace	11
3.1 Kostra aplikace	11
3.2 Registrace a autentizace uživatelů	12
3.3 Nahrání a výběr obrázků	14
3.3.1 Nahrávání obrázků	14
3.3.2 Výběr obrázků ze služby Flickr	14
3.3.3 Výběr obrázků z veřejné složky	15
3.4 Odhadnutí křivky horizontu	16
3.4.1 Proces extrakce popředí a algoritmus GrabCut	17
3.4.2 Zobrazení výsledků a úprava křivky	18
3.4.3 Odhadnutí horizontu	19
3.5 Použití Locate	21
3.5.1 Locate	22
3.5.2 Propojení aplikace s Locate	22
3.6 Zpracování výsledků	22
3.6.1 Zobrazení výsledků	23
3.7 Uživatelské sekce	24
3.7.1 Moje poslední aktivita	24
3.7.2 Moje nahrané obrázky	24

4 Testování a srovnání	25
4.1 Definice testování	25
4.1.1 Výběr uživatelů	25
4.2 Proces testování	26
4.3 Vyhodnocení testování	26
4.3.1 Srovnání skupin uživatelů	30
5 Závěr	31
5.1 Zhodnocení dosažených výsledků	31
5.2 Další vývoj projektu	31
A Obsah CD	35
B Diagram případů užití	36
C Diagram tříd	37
D ER diagram	38
E Ukázky mockupu aplikace	39
F Vyplněné dotazníky testování mockupu	41
G Vyplněné dotazníky finálního testování	46

Kapitola 1

Úvod

V dnešní moderní době se lidé velmi často snaží využít možností vyhledávání a lokalizace osob, míst a dalších cílů jak pomocí GPS systému¹, tak z fotografií. Využití lokalizace je nezměrné a je mu věnována čím dál větší pozornost, zvláště pak z technických oborů, které na tomto základě mohou stavět složitá zařízení nebo programovat zajímavé aplikace². Tato práce je zaměřena na okruh webových aplikací a vyhledávání podobných míst z fotografií. Vyhledávání je omezeno hlavně na hory a pohoří, protože porovnávání funguje na základě křivky horizontu v systému Locate. Tento systém disponuje velkým množstvím vyrenderovaných obrázků, které používá pro porovnání s danou křivkou. Webová aplikace bude plnit formu grafického uživatelského rozhraní (front-end) pro práci s Locate (back-end).

Cílem je vytvořit webovou aplikaci, která umožňuje uživatelsky přívětivé a intuitivní vyhledávání panoramat z fotografií a obrázků. Vyhledávání spočívá v lokalizaci panoramatu pohoří fotografie na základě křivky horizontu. Aplikace poskytne uživateli možnost nahrát vlastní fotografii nebo vybrat ze služby Flickr, určit křivku horizontu a odeslat data, získat výsledky a vizualizovat je.

V podsekcích první kapitoly je naznačena motivace pro realizaci tohoto projektu. Druhá kapitola pojednává o teoretickém návrhu aplikace, definice požadavků, výběru technologií a architektury, návrhu uživatelského rozhraní a testování návrhu.

Třetí kapitola se již plně věnuje praktické části - implementaci aplikace. Proces implementace zahrnuje vytvoření základní kostry aplikace, registraci a autentifikaci uživatelů. Dále zahrnuje nahrání a výběr obrázku s využitím Flickr API a odhadnutí křivky horizontu z fotografie založeném na algoritmu GrabCut. Data se odešlou do Locate a získají se výsledky, které se vizualizují ve formě renderovaných obrázků nejlepších kandidátních pozic. Tyto pozice jsou také zobrazeny na mapě s využitím Google Maps API.

Poslední kapitola se věnuje testování aplikace a uživatelského rozhraní, popisu procesu testování, vyhodnocení výsledků a srovnání výsledků oproti výsledkům z testování návrhu aplikace.

¹Například: OpenStreetMap (<https://www.openstreetmap.org/>), Google Maps (<https://www.google.cz/maps/>), OpenGPS (<http://www.opengps.net/>)

²Například: PeakFinder (<http://www.peakfinder.org/>), Google Earth (<http://www.google.cz/intl/cs/earth/>)

1.1 Rešerše existujících řešení

Pro vytvoření vhodné a použitelné aplikace je velmi důležité zjistit zda-li taková aplikace již existuje, případně se informovat o podobných aplikacích a jejich hlavních kladech a záporech. Nemá příliš velký smysl vytvářet aplikaci, která by již na trhu existovala. Provedl jsem rešerši existujících příbuzných aplikací, které dokážou vyhledávat panoramata. Rešerše se skládá ze dvou skupin aplikací: webové/desktopové aplikace a mobilní aplikace.

Webové aplikace jsem našel čtyři: PeakFinder, Mountain Panoramas, Generate a panorama a Google Earth. Všechny tyto aplikace umožňují pohled 360 ° a nastavení výšky pohledu od zemského povrchu. Nabízí označení vrcholů a poskytují další informace o vrcholu jako GPS souřadnice, nadmořskou výšku a podobné. Podrobnější informace nabízí tabulka 1.1.

Aplikace	Typ zobrazení	Pokrytí	Vyhledání	Export
PeakFinder ³	Vizualizace v odstínech šedé	celý svět (250 000 vrcholů)	GPS, Název hory	URL, PDF
Mountain Panoramas ⁴	Z barevných fotografií ve velkém rozlišení	celý svět (do 1 000 vrcholů)	Název hory, Název pohoří, Název země, Ze seznamu, Na mapě	žádný
Generate a panorama ⁵	Vizualizace renderu v odstínech šedé	celý svět	GPS, Název hory, Na mapě	URL
Google Earth ⁶	Ze satelitních snímků	celý svět	Název hory, Název pohoří, Název země, Na mapě	JPG, KMZ (vlastní formát)

Tabulka 1.1: Srovnání webových aplikací

Mobilních aplikací jsem našel již více: PeakFinder, FindTheHill, Mountain Navigator, Peak Scanner, Peak Search a Peak AR. V tabulce 1.2 jsou podrobnější informace. Každou aplikaci jsem nainstaloval a spustil v emulovacím nástroji BlueStacks [5], který podporuje emulaci platformy Android. Většina aplikací poskytuje přímé promítání z kamery, které po určení polohy okamžitě označí vrcholy ze snímaného pohledu kamerou. Tyto aplikace jsem z drtivé většiny nemohl přímo vyzkoušet, protože většina pracuje jen na určitých místech a s aktuálními souřadnicemi GPS. Aplikace jsem hledal hlavně pro platformu Android, vzhledem k tomu, že je to nejrozšířenější mobilní platforma [27]. Informace o aplikacích jsem našel hlavně na popisu u profilů na Obchodu Play nebo z recenzovaných videí na YouTube.

³<http://www.peakfinder.org/>

⁴<http://www.mountainpanoramas.com/>

⁵http://www.udeutschle.selfhost.pro/panoramas/makepanoramas_en.htm

⁶<http://www.google.cz/intl/cs/earth/>

Aplikace	Typ zobrazení	Pokrytí	Další informace o vrcholu	Distribuce a cena	Platforma
PeakFinder ⁷	Vizualizace obrysů a hran z obrazu kamery	celý svět (250 000 vrcholů)	ano	Placená verze (93 Kč)	Android, Mac OS
FindTheHill ⁸	Přímé promítání obrazu kamery	celý svět (240 000 vrcholů)	ne	Neplacená verze	Android
Mountain Navigator ⁹	Přímé promítání obrazu kamery	velké vrcholy Velké Británie a Irska	ano	Neplacená verze, Placená verze (19,5 Kč)	Android
Peak Scanner ¹⁰	Přímé promítání obrazu kamery	nezveřejněno	ano	Neplacená verze, Placená verze (114 Kč)	Android
Peak Search ¹¹	Přímé promítání obrazu kamery	Alpy, Pyreneje, Norsko, Yellowstone, Yosemite, Nový Zéland	ano	Neplacená verze	Android
Peak AR ¹²	Přímé promítání obrazu kamery	nezveřejněno	ano	Neplacená verze	Android

Tabulka 1.2: Srovnání mobilních aplikací

1.2 Motivace

Průzkum aplikací poskytující vyhledávání panoramat ukázal, že všechny webové aplikace podporují vyhledání panoramatu pomocí názvu vrcholu, pohoří, země a jiné. Mobilní aplikace jsou zaměřeny na označování vrcholů pomocí aktuálních GPS souřadnic zařízení a přímé projekce obrazu z kamery. Není mi známa žádná veřejně dostupná aplikace, která by umožňovala vyhledávání panoramat na základě křivky horizontu.

⁷<https://play.google.com/store/apps/details?id=org.peakfinder.area.alps&hl=cs>

⁸<https://play.google.com/store/apps/details?id=eu.starbug.ourhills>

⁹<https://play.google.com/store/apps/details?id=mountain.identifier>

¹⁰<https://play.google.com/store/apps/details?id=com.quantq.peakscanner>

¹¹<https://play.google.com/store/apps/details?id=com.rachApps.Alps>

¹²<https://play.google.com/store/apps/details?id=at.srfg.peakar>

Kapitola 2

Návrh aplikace

V této kapitole je rozveden proces návrhu. Jsou zde definovány požadavky na aplikaci. Selekce cílové skupiny možných uživatelů, včetně příkladů, jakým způsobem by dané skupiny mohly aplikaci použít. Součástí návrhu je výběr vhodných technologií a architektury, na kterých je projekt postaven. Dalším krokem je návrh uživatelského rozhraní. Pro zjištění správnosti návrhu bylo provedeno uživatelské testování designu, díky kterému byly zjištěny chyby a nedostatky již v samotném návrhu aplikace.

2.1 Definice požadavků

Nefunkční požadavky

Webová aplikace by měla být stabilní, bezpečná, škálovatelná a snadno použitelná. Měla by poskytovat jednoduché a přívětivé prostředí pro práci. Celá aplikace by tak měla být maximálně jednoduchá a zaměřená na uživatele.

Funkční požadavky

Požadujeme webovou aplikaci, která by uživateli umožňovala vyhledávání panoramat z obrázků. Aplikace má obsahovat možnost registrace a přihlášení uživatelů. Nahrané a použité obrázky nepřihlášených uživatelů budou dále k dispozici ve veřejné složce pro další uživatele. Registrací uživatel získá možnost prohlédnout si svá poslední vyhledávání panoramat. Uživatel bude mít možnost znovu použít své nahrané obrázky. Použití aplikace je znázorněno na diagramu případů užití (viz příloha B) v jazyce UML [23]

Aplikace se skládá z těchto částí:

1. Nahrání obrázku z lokálního úložiště nebo z URL adresy. Nebo výběr obrázku ze služby Flickr nebo z veřejné složky.

Nahrání nebo výběr obrázku je pro uživatele dostupné hned z úvodní stránky aplikace, kde si vybere odkud chce obrazový materiál čerpat. Možnost pro přímé nahrání obrázku bude výběrem souboru nebo vložením adresy vystaveného obrázku na webu. Pokud si vybere službu Flickr nebo veřejnou složku, zobrazí se miniatury obrázků - obsahu dané varianty. Kliknutím na miniaturu se zobrazí větší rozměr fotky.

2. Poloautomatické odhadnutí křivky horizontu s možností upravení křivky do konečné podoby.

Po výběru obrázku se aplikace pokusí odhadnout křivku horizontu. Uživatel křivku upraví a nechá provést přesnější odhadnutí křivky aplikací. Výsledný horizont uživatel upraví do konečné podoby a aplikace odešle křivku ke zpracování.

3. Odeslání dat do Locate a zpracování výsledků, které se vizualizují uživateli formou vyrenderovaných obrázků. Výsledky jsou také zobrazeny na mapě pomocí Google Maps API.

Konečná podoba křivky je zpracována a odeslána do Locate, kde proběhne lokalizace. Výsledky jsou odeslány z Locate zpět do aplikace, kde jsou interpretovány ve formě zarovnaných křivek panoramatu a horizontu.

2.1.1 Selektce skupin uživatelů

Každá aplikace je vytvářena vždy pro předem známé skupiny potenciálních uživatelů. Bylo třeba se zamyslet nad otázkami: Kdo bude aplikaci používat? Proč by měl dotyčný aplikaci používat? Dle definovaných skupin jsem se zaměřil právě na tyto budoucí uživatele.

Definované skupiny a možné příklady použití:

Turisté: Turista vyleze na vrchol. Vyfotí horizont a pomocí vyhledávače zjistí informace o své poloze a okolí.

Architekti výškových pozorovacích bodů: Vytvoření informačního panelu pro konkrétní úhel horizontu na vrcholcích hor, rozhleden, hradů, výškových budov.

Náhodní zájemci a badatelé: Na sociální síti se uživatel líbí fotografie hor. Zjistí tak o jaké hory se jedná a kde se nachází. Uživatel nalezne na půdě starou fotografii jeho pradědečka na horách. Opět zjistí, kde to tehdy pradědeček byl. Uživatel má rok starou fotografii, když byl v Alpách. Je si vědom, kde přibližně byl, ale neví to přesně. Použije vyhledávač pro upřesnění. V rámci výzkumu algoritmů pro vizuální geolokalizaci v přírodě je možné aplikaci použít pro porovnání s novými metodami.

Kantoři geografických a příbuzných oborů: Učitel využije aplikaci jako učební pomůcku.

2.2 Výběr technologií

Pro návrh vyhledávače jsem využil jazyka UML a jeho návrhu: diagram tříd. Tyto návrhy jsem vytvořil v programu Dia. Návrh databáze a jeho ER diagramu (viz příloha D) byl proveden v MySQL Workbench, který mimo jiné nabízí také pohodlný export navrženého ER diagramu přímo do databázové instance.

Aplikaci jsem vyvíjel na virtuálním počítači: Systém Linux Ubuntu 14.04 64-bit; 2 GB RAM; 2 jádra procesoru; 30 GB HDD. Emulátorem pro virtuální počítač byl program VMware Workstation 12 Player.

Vytvoření aplikace jsem se rozhodl provést ve frameworku Nette napsaném v jazyce PHP [4]. Poskytuje podporu pro nejnovější webové technologie. Má na vysoké úrovni zabezpečení a ochranu proti zneužití aplikace. Obsahuje také ladicí nástroj Tracy - Laděnka poskytující okamžité informace a metriky. Nette má tedy spoustu výhod, možná proto se umístil na 3. místě nejlepších frameworků [1].

Pro databázové účely jsem použil klasickou kombinaci Apache, tedy webový server. MySQL poskytující databázovou vrstvu. A nástroj phpMyAdmin pro přívětivou a pohodlnou práci s databází.

2.3 Návrh architektury

Architektura aplikace byla postavena na MVC (Model-View-Controller), rozdělující jednotlivé vrstvy aplikace a definující, jak a kdo spolu komunikuje. Model se stará pouze o uchování a práci s daty. Dále udržuje spojení s databází. Controller, v Nette prostředí spíše označován jako prezentér, se stará o logiku aplikace. Obsahuje směrování. Zajišťuje přesuny dat z modelu do jednotlivých „view“. View - šablony pak zajišťují prezentační vrstvu aplikace. Na této úrovni probíhá formátování dat a jejich výstup.

Detailní návrh architektury aplikace je vizualizován pomocí diagramu tříd (viz příloha D). Aplikace se sestává ze tříd: tří prezentérů a dvou modelů. Metody v třídách a proměnné se v průběhu vývoje postupně doplňovaly.

2.4 Návrh uživatelského rozhraní

Uživatel by v nejlepším případě měl na první pohled vědět, kde se v aplikaci nachází a co dělat dále. Měly by v aplikaci být srozumitelné a dobře viditelné navigační prvky, popisky a varovná hlášení. Toho chci docílit drobečkovou navigací a velkými navigačními prvky po stranách aplikace pro možnost jít zpět nebo pokračovat.

Definovaných požadavků jsem dosáhl vykreslením lomené úsečky napříč obrázkem dle odhadnutého horizontu a určených referenčních bodů - pivotů, se kterými bude pro uživatele jednoduchá možnost úpravy pomocí Drag n Drop. K dispozici budou také tlačítka pro vložení nového pivotu nebo smazání již existujícího.

Návrh uživatelského rozhraní byl ovlivněn pomůckami a radami pro tvorbu jednoduchých, přehledných a poutavých designů [25]. Návrh GUI jsem provedl v programu Pencil [6], ve kterém jsem vytvořil mockup aplikace obsahující všechny ovládací a grafické prvky budoucí aplikace. Tento mockup (viz příloha E) jsem vyexportoval do HTML formátu s funkčními odkazy a tlačítky.

2.5 Uživatelské testování designu

Vytvořil jsem návrh grafického rozhraní aplikace. To však nezaručuje, že je návrh opravdu správný, a že splňuje požadavky definované v kapitole 2.1. Mockup byl podroben uživatelskému testování [32] pro zjištění chyb a nedostatků již v samotném návrhu. Zjištěné chyby tak mohly být opraveny ještě před samotnou implementací aplikace. Testování jsem zvolil kvalitativní s uživatelem [22] a skládá se ze tří kroků: výběr uživatelů, testování a vyhodnocení.

2.5.1 Výběr uživatelů

Testování by mělo probíhat na uživateli, kteří spadají do definovaných skupin uživatelů. Nemá smysl testovat aplikaci na lidech, kteří by aplikaci pravděpodobně nikdy nepoužili nebo s tímto oborem nemají absolutně nic společného. Pro zjištění vhodnosti uživatelů jsem dal každému uchazeči úvodní dotazník: Screener.

Dotazník se skládal z otázek:

- Máte zkušenosti s těmito webovými aplikacemi pro nahrávání a sdílení obrázků?
- Zajímala vás někdy poloha nějakých vrcholů, hor, pohoří z obrázků / fotografií?
- Cestujete do hornatých oblastí? Podnikáte turistické / cyklistické / lyžařské výjezdy na hory?
- Pokud jste na předchozí otázku odpověděli ANO: Pořizujete z těchto oblastí fotografie vrcholů, hor, horizontů, panoramat?

Každé otázce jsem definoval 3-5 volitelných odpovědí, přičemž jsem každé odpovědi určil, zda-li je odpověď přijatelná nebo ne. Pomocí tohoto dotazníku jsem získal 5 vhodných respondentů ve věku 20-65 let.

2.5.2 Testování mockupu

Průběh testování se skládal z uživatelského testování mockupu. Uživatelé byly postupně předloženy 4 úkoly, které provedly uživatele všemi obrazovkami mockupu aplikace. Každý uživatel byl testován za přítomnosti moderátora¹, který pozoroval a kontroloval, že testování probíhá v pořádku. Při testování byl pořizován obrazový materiál snímající obrazovku uživatele. V průběhu testování dostali uživatelé k vyplnění dva dotazníky. První dotazník testoval kolik informací uživatel získá z úvodní obrazovky aplikace za prvních 20 sekund prohlížení. Druhý dotazník uživatel vyplňoval až na konci testování po proběhnutí všech úkolů. Na vyplňování dotazníku měli uživatelé libovolné množství času. Dotazníky se skládaly z otevřených otázek, z dichotomických otázek („ano/ne“), a z otázek Likertovy škály (metoda měření postojů) [29].

2.5.3 Vyhodnocení testování

Vyplněné dotazníky jsem prošel (viz příloha F) a získal tak cenné informace o návrhu celkovém a jednotlivých aspektech aplikace. Při vyhodnocování také pomohly pořízené video záznamy, ze kterých šlo jednoduše zjistit, kam uživatel kliknul. V mockupu nebyly funkční všechny odkazy (například pro vyvolání dialogového okna pro nahrávání obrázku), a také nebylo možno prvkům nastavit „hover“ efekty. Někteří uživatelé tak byli občas vyvedeni z míry, když po kliknutí neměli zpětnou vazbu jakou by čekali.

Zobrazení výsledků (majoritní problém) Největší problémy dělala stránka se zobrazením výsledků. Čtyři z pěti uživatelů nepochopili význam výsledků. Uživatelé nevěděli co je jim zobrazováno a k čemu vlastně celý proces vyhledání panoramat vedl. Je možné,

¹Moderátorem byl autor práce.

že uživatelé byli částečně zmatení protože obrázek vyrenderovaných hor neodpovídal příkladovému obrázku použitému pro vyhledávání. Dále se pod všemi výsledky nacházela ještě Google mapa s určenými polohami jednotlivých kandidátních pozic. Vzhledem k tomu, že výsledky byly uspořádány pod sebou neměli uživatelé vůli se posouvat níže a narazit tak na Google mapu, která by jim mohla poskytnout další informace co výsledky vlastně znamenají.

Moje poslední aktivita (majoritní problém) V této uživatelské sekci si jen tři z pěti uživatelů byli jistí, kde se právě nachází a co je jim zobrazováno. Tedy dva uživatelé si tuto sekci na první pohled spletli se sekci Moje nahrané obrázky. Až po dalším zkoumání uživateli došlo, kde se nacházejí. Tři z pěti uživatelů se také pokusili zobrazená data v této sekci nějak editovat, přičemž je tato sekce určena pouze pro prohlížení předešlých vyhledávání.

Řešením problému je přidat výraznější prvky označení sekcí a popisek, že se zde jen prohlíží vyhledávání uživatele.

Určování křivky horizontu (střední problém) Na stránce s určováním křivky horizontu se tři z pěti uživatelů pokusili křivku horizontu upravit. Zbytku přišla určená křivka dostatečně přesná. Uživatelé neměli s ovládáním a úpravou křivky žádné problémy. Pro určování horizontu tedy nebylo třeba významnějších úprav.

Podokno s vybraným obrázkem (minoritní problém) Podokno s obrázkem, které se uživateli zobrazí po výběru obrázku ze služby Flickr, z veřejné složky nebo s uživatelské sekce Moje nahrané obrázky, dělalo uživatelům problémy ve smyslu, že nevěděli jak podokno zrušit nebo jak se vrátit zpět.

Řešením je přidání tlačítka zavřít.

Ovládací prvky - zpět/pokračovat (minoritní problém) V mockupu byly po stránkách ovládací prvky pro návrat na předchozí stránku - zpět, a pro pokračování - pokračovat. Což bylo realizováno směrovací šipkou. To přišlo třem z pěti uživatelům zmatečné neboť na první pohled nevěděli zda-li kliknout na šipku nebo na nápis.

Umístěním nápisu do šipek bude problém odstraněn.

Nadpisy stránek (minoritní problém) Zcela náhodně se stávalo, že si uživatelé nebyli jistí, kde se právě nachází, a co všechno mohou na dané stránce dělat.

Přidáním a zvýrazněním nadpisu ke každé stránce včetně stručného popisu k čemu stránka slouží.

Uživatelé měli z testování různé pocity: příjemné, neutrální. Někteří přišlo nepříjemné, že nevěděli co je vlastně čeká. Nebo že byl pořizován video záznam. Testování oceňovali důraz na detail, způsob práce s křivkou horizontu a jednoduchost a srozumitelnost aplikace. Největší problémy dělalo uživatelům pochopení výsledků. Uživatelé by na aplikaci uvítali více popisných a pomocných prvků jako popisky, „hover“ efekty, načítací symbol při čekání: nahrávání obrázku, určování horizontu a podobné.

Z vyplněných dotazníků jsem zjistil také obecné názory na celkovou aplikaci. Aplikace byla pro uživatele srozumitelná a v budoucnu by tuto aplikaci použili znovu. Většina si myslí, že je aplikace zajímavá pro širokou veřejnost. Testování, dle předpokladu, odhalilo slabá a nepřehledná místa aplikace. Tyto informace jsem zohlednil při implementaci aplikace.

Kapitola 3

Implementace

Kapitola 3 popisuje etapu implementace projektu. Počínaje zprovoznění základní kostry aplikace a zprovoznění registrace a autentifikace uživatelů. Dále je popsána implementace možností nahrání obrázku včetně propojení aplikace se službou Flickr. Velkou částí je zprovoznění procesu odhadnutí křivky horizontu s použitím algoritmu GrabCut. S tím je spojené zpracování dat do vhodné formy pro odeslání do Locate. Další částí implementace je přijetí výsledků z Locate a jejich vizualizace uživateli. Posledním krokem bylo zprovoznění uživatelských sekcí Moje poslední aktivita a Moje nahrané obrázky.

3.1 Kostra aplikace

Dle výběru technologií v kapitole 2.2 jsem použil Nette framework. Nainstaloval jsem základní Nette projekt pomocí programu Composer [12]. K spustitelné aplikaci jsem vytvořil MySQL databázi dle navrhnutého ER diagramu a v konfiguračním souboru spojil aplikaci s databází.

Adresářová struktura s nejdůležitějšími soubory:

- */app* - adresář s aplikací
 - /config* - konfigurační soubory
 - /model* - třídy modelů
 - /presenters* - třídy presentérů
 - /router* - třídy pro routování
 - /templates* - adresář se šablonami
 - bootstrap.php* - zaváděcí soubor aplikace
- */log* - obsahuje logy, chyby
- */temp* - pro dočasné soubory
- */vendor* - adresář pro nette a další knihovny
- */www* - veřejný adresář, document root projektu
 - /css* - složka pro css styly
 - /js* - složka pro javascript skripty
 - index.php* - spouští aplikaci

Do složek s modely jsem doplnil podsložky a soubory s prozatím prázdnými třídami dle diagramu tříd. Dále jsem k modelovým třídám doplnil třídu pro budoucí zajištění přihlášení uživatele. V konstruktoru každé třídy jsem již připravil předání databázového spojení z DI kontejneru aplikace, kterou zajišťuje Nette. Všechny třídy modelů byly zaregistrovány jako služby (viz 3.1) v konfiguračním souboru. Při spuštění aplikace se automaticky vytvoří objekty modelů a budou dostupné v DI kontejneru jako služby.

```
services :
    - App\Model\PersonAuthenticator
    - App\Model\ApplicationModel
    - App\Model\ImagesModel
```

Kód 3.1: Zaregistrování modelů jako služby v konfiguračním souboru.

Prezentéry jsem také doplnil dle diagramu tříd. Vytvořil jsem abstraktní třídu BasePresenter, ze které pak dědí všechny ostatní třídy presentérů. V BasePresenter se budou nacházet metody společné pro všechny třídy, a také se zde do proměnných nainjektují objekty (viz 3.2) modelů opět z DI kontejneru. Bude tak možné použít proměnné i metody jednotlivých modelových tříd v jakémkoliv presentéru, a to jednoduchým přístupem přes proměnnou, do které byl model nainjektován.

```
/** @var \App\Model\ApplicationModel @inject */
public $appModel;
```

Kód 3.2: Ukázka injekce modelu ApplicationModel v presentéru BasePresenter.

Pohledy aplikace - šablony se v aplikaci rozdělují na šablony pro obecný layout aplikace, kde se definuje celkový design, a na šablony, již pro konkrétní zobrazování obsahů v layoutu. Upravil jsem layout aplikace, protože pro celou aplikaci bude stále stejný. Napojil jsem CSS styly a jQuery. Do složky se šablonami jsem doplnil podsložky a šablony k jednotlivým pohledům (view).

3.2 Registrace a autentizace uživatelů

Uživatelé se mohou pro získání výhod registrovat a přihlašovat. Mezi výhody přihlášeného uživatele patří sekce Moje poslední aktivita, kde se zálohují veškerá vyhledávání uživatele, které si může později prohlédnout a vracet se k nim. Další výhodou je sekce Moje nahrané obrázky, kde se ukládají všechny nahrané obrázky uživatele, které může použít znovu. Při registraci uživatelé vyplňují uživatelské jméno a heslo, kterými se přihlašují do aplikace. Dále jméno a příjmení, které se zobrazuje po přihlášení v pravém horním rohu aplikace. Posledním údajem v registraci uživatele je emailová adresa, která je použita pro ověření a zaslání nového hesla v případě ztráty původního.

O registraci a přihlašování se stará v prvé řadě formulář (viz ukázka 3.3), který je vytvořen pomocí Nette a jeho předchystanému rozhraní pro vytváření, kontrolu a zpracování formulářů. Všechny formuláře se vytvářejí a zpracovávají v presentéru SignPresenter. V jednotlivých formulářích jsou vytvářeny vstupní formulářové prvky. K dispozici jsou všechny formulářové prvky HTML. Těmto prvkům je pak možné jednoduše definovat identifikátor, atribut „vyžadován“. Nebo podmínky minimálního/maximálního počtu znaků, interval

povolených číselných hodnot, případně lze také definovat kontrolu oproti vlastnímu regulárnímu dotazu. Podmínky jsou na straně klienta kontrolovány javascriptem.

```
protected function createComponentSignInForm() {
    $form = new UI\Form;

    $form    ->addText('nick', 'Uživatelské_jméno:')
            ->setRequired('')
            ->addRule(Form::PATTERN, '', '([\w]+)')
            ->addRule(Form::MAX_LENGTH, '', 25);

    $form    ->addPassword('password', 'Heslo:')
            ->setRequired('')
            ->addRule(Form::MIN_LENGTH, '', 6);

    $form->addSubmit('send', 'Přihlásit');

    $form->onSuccess[] = array($this, 'signInFormSucceeded');

    return $form;
}
```

Kód 3.3: Ukázka vytváření formuláře pomocí Nette.

Registrace je tvořena formulářem obsahujícím vstupní pole pro uživatelské jméno, jméno, příjmení, email, heslo a heslo pro kontrolu. Uživatelské jméno a heslo je použito pro přihlašování. Jméno a příjmení je uvedeno po přihlášení v pravém horním rohu aplikace. Registrace probíhá v modelu ApplicationModel. Pro zabezpečení hesel je použit algoritmus solení [24]. Vygeneruje se náhodné 64bitové číslo – sůl. Vytvoří se řetězec ze jména, hesla a soli a celý je zahešován (viz 3.4) algoritmem SHA-2. Do databáze se uloží zahešovaný řetězec hesla a soli, a do dalšího sloupce je uložena sůl v nehešované podobě.

```
$salt = $this->makeSalt();
$password = hash("sha256", $values->nick.$values->password.$salt);
```

Kód 3.4: Ukázka vytvoření soli a zahešování hesla.

„V Nette je přihlašování zajištěno službou user, což je objekt třídy Nette\Security\User. Službu můžeme získat voláním \$user = \$this->getUser().“ [3] Přihlašování je tvořeno formulářem pro uživatelské jméno a heslo a probíhá pomocí speciální třídy - autentizátoru PersonAuthenticator. Autentizátor ověřuje uživatelské jméno a heslo oproti databázové tabulce. Pro ověření hesla se načte sůl podle uživatelského jména. Heslo se solí se společně opět zahešuje a probíhá porovnávání hešovaných řetězců. V případě úspěchu autentizátor vrací identitu, což je soubor informací o uživateli. „Jedná se o objekt implementující rozhraní Nette\Security\IIdentity. Třída disponuje metodou getId(), která vrací ID uživatele (např. primární klíč v databázi), a getRoles(), která vrátí seznam všech rolí, ve kterých uživatel vystupuje. K uživatelským datům lze přistupovat jako k proměnným objektu.“ [3]

V případě zapomenutí hesla jsem připravil možnost vygenerování nového hesla, která

také pracuje pod `SignPresenter` a `ApplicationModel`. Formulář obsahuje pole pro uživatelské jméno a email. Na základě kontroly jména i emailové adresy je pak vygenerováno náhodné desetimístné heslo, které je pomocí knihovny `PHPMailer` [2] odesláno mailovou službou na email uživatele. Taktéž se vygeneruje nová sůl a společně s novým zahešovaným heslem se informace aktualizují v databázi. Z bezpečnostních důvodů je uživatelům doporučováno při příštím přihlášení změnit heslo za vlastní.

Uživatelská data může uživatel změnit po přihlášení v uživatelské sekci Nastavení účtu. Zde jsou připraveny dva formuláře. Jeden pro aktualizaci údajů jako je uživatelské jméno, email a podobně. Druhý je pro změnu hesla. V případě změny hesla se opět generuje nová sůl.

3.3 Nahrání a výběr obrázků

V této podkapitole je rozvedena implementace nahrání a výběru obrázku. Všechny použité fotografie se ukládají do podsložek v adresáři `www/uploaded/`. Pro nepřihlášené uživatele se obrázky ukládají do podsložky `shared/`. Přihlášeným uživatelům je při použití prvního obrázku vytvořen adresář, do kterého jsou pak ukládány všechny obrázky konkrétního uživatele.

3.3.1 Nahrávání obrázků

Obrázky je možné nahrát z lokálního úložiště nebo z internetu - vložením URL adresy obrázku. Obě metody jsou implementovány jako dva formuláře se vstupními prvky pro výběr souboru a pro vložení URL. Po odeslání formuláře proběhne nahrání obrázku na server, přejmenování a uložení do složky podle přihlášeného nebo nepřihlášeného uživatele. Pokud je uživatel přihlášen, uloží se do databáze záznam o obrázku.

3.3.2 Výběr obrázků ze služby Flickr

Použití obrázku ze služby Flickr je přímo vestavěno v aplikaci, a to použitím Flickr API [9] a knihovny `phpFlickr` [18], která obaluje Flickr API a poskytuje tak jednodušší a přívětivější práci. Flickr API obsahuje metody pro práci s obrázky Flickr. Knihovna se stará o spojení aplikace s Flickr a obsahuje implementované všechny metody Flickr API, ke kterým se dá přistupovat přes jednotné rozhraní.

```
$api = new phpFlickr(FLICKR_API_KEY, FLICKR_SECRET, false);
```

Kód 3.5: Vytvoření instance `phpFlickr`.

Pro používání služeb Flickr jsem nejdříve musel aplikaci zaregistrovat a získat přístupové tokeny `FLICKR_API_KEY` a `FLICKR_SECRET`, bez kterých není možné Flickr API používat. Knihovna `phpFlickr` se používá vytvořením instance třídy (viz 3.5) s argumenty přístupových tokenů a parametrem zda-li má být instance zničena při chybě. Všechny parametry jsou nastavitelné v konfiguračním souboru ve složce `app/model/`. Pro získání obrázků jsem zvolil metodu `photos_search` [7] (viz 3.6), kterou lze zavolat s velkým množstvím nepovinných parametrů. V našem případě je metoda volaná s parametry: vyhledávací řetězec „alps panorama“, viditelnost „pouze veřejné“ a počet obrázků na jednu stránku „50“.

Flickr API vrácí XML odpověď, kterou knihovna phpFlickr automaticky zpracuje a vytvoří asociativní pole s jednotlivými údaji o obrázcích.

```
$response = $api->photos_search (array (
    'text' => SEARCH_STRING,
    'privacy_filter' => SEARCH_PRIVACY_FILTER,
    'per_page' => SEARCH_PER_PAGE,
    'page' => $page
));
```

Kód 3.6: Zavolání metody Flickr API pomocí instance knihovny phpFlickr.

Tyto údaje lze použít pro získání konkrétních obrázků různých velikostí. Flickr používá vlastní konstruování URL adres [8] pro přístup k fotografiím a jejím různým rozměrům. Čili k získání URL adresy obrázku je nutné znát ID obrázku, ID serveru, ID „farmy“ a „secret“ řetězec fotografie. Šablona pro konstrukci URL obrázku pak vypadá takto:

`https://farm{farm-id}.staticflickr.com/{server-id}/{id}_{secret}_o.jpg`

Tyto informace jsem ke každému obrázku získal v asociativním poli, které mi vrátila metoda `photos_search`. Celé pole předávám do šablony, kde se nad polem iteruje a vykresluje se mřížka náhledů (75x75px). Pro zobrazování a prohlížení obrázků jsem použil knihovnu Lightbox2 [19]. Prohlížení obrázku v Lightboxu jsem upravil, a na pravé straně zobrazeného obrázku jsem umístil kontrolní prvek, který uživateli umožňuje vybrat konkrétní obrázek a pokračovat k určování křivky horizontu.

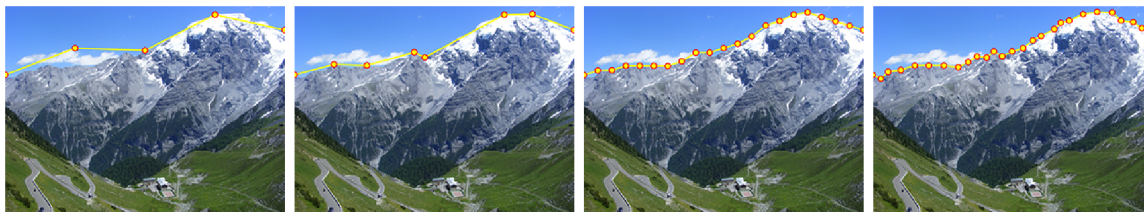
Flickr poskytuje různé velikosti původních obrázků, do kterých však nekládá původní EXIF data (informace, která do fotografie ukládá digitální fotoaparát). EXIF data jsou však pro použití obrázku v aplikaci nezbytné. Z EXIF metadat se vypočítává informace o úhlu pohledu – FOV, která je vyžadována Locate. Různé verze obrázků na Flickr tuto informaci neobsahují. Některé obrázky však mají tyto informace k dispozici skrze další metodu Flickr API. Při výběru konkrétního obrázku se aplikace pokusí tyto informace stáhnout a pomocí programu ExifTool [21] vložit do obrázku. ExifTool je zavolán znovu, a to pro pokusný výpočet FOV informace. V případě úspěchu je přikročeno k určování horizontu. V opačném případě je uživateli zobrazeno hlášení, že tato fotografie nelze použít.

3.3.3 Výběr obrázků z veřejné složky

Každý obrázek použitý v aplikaci nepřihlášeným uživatelem je ukládán do veřejné složky *www/uploaded/shared*. Tyto obrázky jsou dostupné v možnosti výběru obrázku z veřejné složky. Uživatelé tak mohou použít pro svá vyhledávání jakýkoliv obrázek z veřejné složky. Do obrázků veřejné složky jsou zařazeny i všechny obrázky, kterým přihlášení uživatelé nastavili atribut „veřejný“. Zobrazení náhledů obrázků a prohlížení je realizováno stejně jako v případě Flickr obrázků pomocí javascriptové knihovny Lightbox2. Výběr obrázku probíhá také stejně jako v případě obrázků ze služby Flickr.

3.4 Odhadnutí křivky horizontu

Uživatel nahraje obrázek z lokálního úložiště nebo URL adresy, nebo si jej vybere z Flickru nebo z veřejné složky. Dalším krokem je proces určení křivky horizontu, který se skládá z těchto kroků:



(a) Hrubý odhad (b) Upravený horizont (c) Přesný odhad (d) Konečná podoba

Obrázek 3.1: Podoba horizontu v jednotlivých krocích.

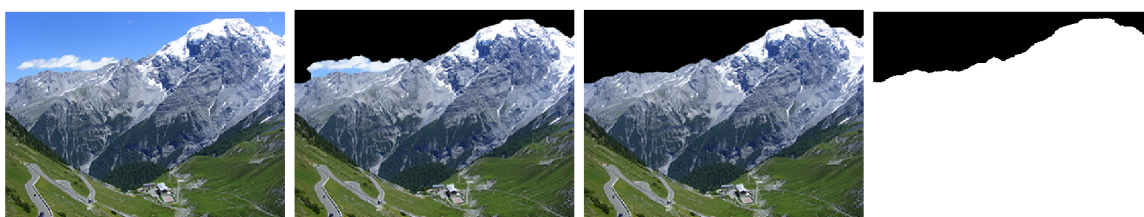
První odhadnutí křivky horizontu 3.1a: Aplikace v prvním kroku automaticky vytvoří hrubý odhad křivky horizontu. Je-li uživatel přihlášen, jsou souřadnice pivotů uloženy do databáze.

Uživatelské upravení hrubého odhadu horizontu 3.1b: Odhad je zobrazen uživateli v editovatelné formě. Uživatel upraví hrubý odhad.

Druhé odhadnutí křivky horizontu 3.1c: Uživatelem upravená křivka je použita k určení již téměř přesné křivky horizontu. Je-li uživatel přihlášen, jsou souřadnice pivotů uloženy do databáze.

Uživatelské upravení křivky do konečné podoby 3.1d: Křivka je opět zobrazena uživateli s možností úpravy křivky do konečného stavu.

V průběhu procesu odhadnutí křivky horizontu můžeme identifikovat celkem čtyři verze obrázku:



(a) Původní (b) První odhad (c) Druhý odhad (d) Konečný

Obrázek 3.2: Typy obrázků v jednotlivých krocích.

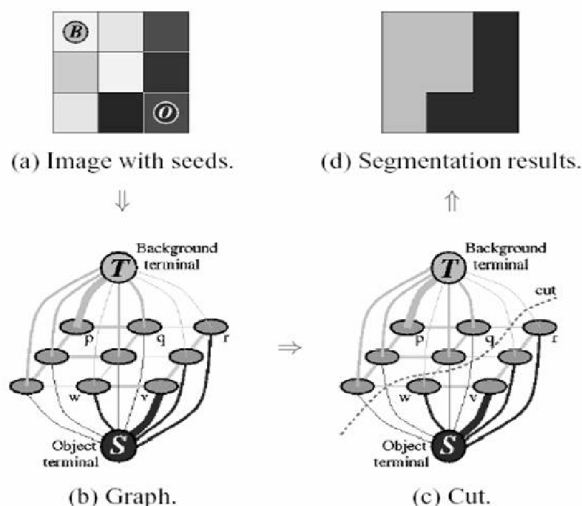
- Původní obrázek. 3.2a
- Obrázek vytvořený po prvním odhadnutí horizontu. 3.2b
- Obrázek vytvořený po určování horizontu z prvního kroku. 3.2c
- Výsledný obrázek vytvořený z přesné křivky horizontu. 3.2d

Pro uchování a zpracování všech verzí jsem vytvořil pomocnou modelovou třídu Images, která uchovává cestu k obrázku a jeho rozměry. Objekty těchto tříd se vytvoří při vytváření třídy ImagesModel při spuštění aplikace. Implementace konkrétních částí jednotlivých kroků procesu odhadnutí křivky horizontu je rozebrána v následujících podkapitolách.

3.4.1 Proces extrakce popředí a algoritmus GrabCut

Pro odhadnutí křivky horizontu jsem zvolil algoritmus GrabCut [30] z knihovny OpenCV [10]. Knihovna OpenCV má dostupná rozhraní v jazycích C++, C, Python a Java. Pro spojení aplikace napsané v jazyce PHP a OpenCV jsem zvolil distribuci rozhraní v jazyce Python. Bylo tedy třeba na virtuální počítač nainstalovat OpenCV s náležitými závislostmi pro jazyk Python.

Proces odhadnutí je založený na extrakci popředí obrázku užitím algoritmu GrabCut. GrabCut je segmentační technika, která pro segmentaci používá metodu maximálního toku (minimálního řezu) v síti. GrabCut používá informace obsažené v obrázku. Pro segmentaci většina segmentačních technik využívá informace o hranici nebo informace o okolí. GrabCut využívá obě metody pro dosažení kvalitních výsledků segmentace.



Obrázek 3.3: Diagram GrabCut algoritmu

zdroj: <http://www.cs.ru.ac.za/research/g02m1682/>

Celý proces segmentace pomocí metody maximálních toků v síti je znázorněn na obrázku 3.3. Nejdříve je vytvořen graf, kde uzly v grafu reprezentují pixely obrázku. Navíc jsou vytvořeny speciální dva uzly: „Source“ a „Sink“. Každý pixel v grafu je spojen se Source nebo Sink uzlem. Source uzel představuje popředí obrázku. Sink uzel naopak pozadí. Další zpracování zajišťuje funkce, která označí váhy mezi pixelovými uzly a Source a Sink uzly. Vzájemné váhy mezi pixelovými uzly jsou označeny jako informace o hranici. Tato informace určuje hranice mezi dvěma pixely (velký rozdíl barev pixelů) a udává velmi malé váhy mezi dvěma pixelovými uzly. Informace o oblasti označuje váhy mezi pixelovými uzly a Source a Sink uzly. Tyto váhy jsou vypočítány pomocí pravděpodobnosti, že uzlový pixel je součástí pozadí nebo popředí. Jakmile jsou oblasti pozadí a popředí vytvořeny, musí některé pixely být označeny - je jim přiřazeno omezení, které během segmentace nemůže být změněno. Pro segmentování grafu je využit algoritmus, který používá energetickou

funkci pro minimalizaci. To vede k vyřešení Max-Flow/Min-cut teorému. [17]. Algoritmus určí minimální řez, který rozdělí Source a Sink uzly. Minimální řez je určen sumou všech vah uzlů, které mají být řezem rozděleny. Jakmile jsou Source a Sink uzly separovány, všechny pixelové uzly spojené se Source se stávají popředím. Všechny Sink uzly se stávají pozadím. [28]

Kompletní obsluha je naprogramována v python skriptu grabcut.py. Skript je složen z jedné třídy, která obrázek provází procesem extrakce popředí pomocí výsledků vrácených z GrabCut. Skript přijímá dva povinné argumenty: cestu k obrázku a počet iterací extrakce. Skript nejdříve normalizuje šířku obrázku na 1000 px a výšku dopočítá v původním poměru. Dále následuje inicializace potřebných polí a proměnných pro GrabCut. Samotné zpracování proběhne ihned po inicializaci (viz 3.7). Na základě výsledků je vygenerován obrázek o stejných rozměrech. V obrázku jsou obrazové body, které GrabCut označil jako pozadí, vyplněny černou barvou. Popředí zůstane v původních barvách. Tento obrázek je fyzicky uložen.

```
def initGrabCut(self):
    self.mask = np.zeros((self.height, self.width), np.uint8)
    self.bgdModel = np.zeros((1, 65), np.float64)
    self.fgdModel = np.zeros((1, 65), np.float64)
    self.rect = (0, int(self.height*0.025), self.width, self.height)
    self.iterCount = int(self.iterations)

def processGrabCut(self):
    self.initGrabCut()
    cv2.grabCut(self.img, self.mask, self.rect,
               self.bgdModel, self.fgdModel,
               self.iterCount, cv2.GC_INIT_WITH_RECT)

    mask2 = np.where(
        (self.mask == 2) | (self.mask == 0), 0, 1).astype('uint8')
    self.img = self.img * mask2[:, :, np.newaxis]
```

Kód 3.7: Ukázka inicializace a zpracování pomocí GrabCut.

Nahráním nebo výběrem obrázku se zpracování přesouvá právě do ImagesModel. Odtud je také volán python skript s extrakcí popředí. Po zpracování skriptu je zpět do ImagesModelu předána cesta k vygenerovanému výsledkovému obrázku s pozadím označeným černou barvou.

3.4.2 Zobrazení výsledků a úprava křivky

Výsledkem zpracování algoritmem GrabCut je vždy pouze obrázek, kde je pozadí označeno černou barvou. Popředí zůstává v barvách původních. Toto označené pozadí bylo nutné nějakým způsobem vizualizovat uživateli. V rámci definovaných požadavků jsem také musel přihlížet k tomu, aby vizualizace křivky horizontu byla manipulovatelná, tak aby mohl uživatel křivku upravit. Z výstupního obrázku lze extrahovat křivku horizontu, která je definovaná vrchní hranicí černé oblasti obrázku.

Vizualizace se skládá z vykreslení obrázku a na něm je pak vykreslena křivka horizontu ve formě lomené úsečky a pivotů (body zlomu). Pro vizualizaci jsem zvolil javascriptovou knihovnu Konva [26]. Knihovna je velmi rozmanitá a umožňuje velmi přívětivou práci s vizualizací téměř jakéhokoliv grafického obsahu pomocí javascriptu. Pro vykreslování v prohlížeči se vždy vytvoří dvě vrstvy canvasu, které fungují jako backend a frontend. V praxi se definuje třída, do které se pak implementují jednotlivé vytváření instancí grafických prvků a metody vyvolané přímou invokací v kódu nebo zavolané na základe určeného „Listeneru“. Každý grafický či geometrický prvek je implementován svou třídou se svými parametry i metodami. Na stránkách je k dispozici také kompletní dokumentace s celou řadou tutoriálů.

První krok určování křivky horizontu vytvoří jen velmi hrubý odhad. Proto při zpracování výstupního obrázku v ImagesModel je obrázek načten do dvourozměrného pole odpovídající rozlišení obrázku a obsahující pro každý pixel (každý index pole) hodnotu barvy RGB modelu. Pole je pak navzorkováno po 1/5 šířky. Získáme tak 6 bodů určujících odhad křivky horizontu. Souřadnice bodů jsou předány pomocí prezentéru do šablony, kde body vykreslí a propojí se lomenou úsečkou. Vykreslené body jsou navíc manipulovatelnými pivoty. Pro lepší práci jsem také implementoval metody pro přidání a smazání pivotu. K dispozici jsou tak tlačítka, které po kliknutí přidají na odpovídající pozici kurzoru nový pivot. Nebo pivot, který se při kliknutí nachází pod kurzorem smaže. Při tomto zásahu se okamžitě také překreslí lomená úsečka horizontu. Do obrázku lze přidat libovolné množství bodů a docílit tak větší přesnosti. Smazáním všech bodů se smaže i křivka horizontu. Vytvořením dvou nových bodů do prázdného obrázku se vytvoří nová křivka horizontu.

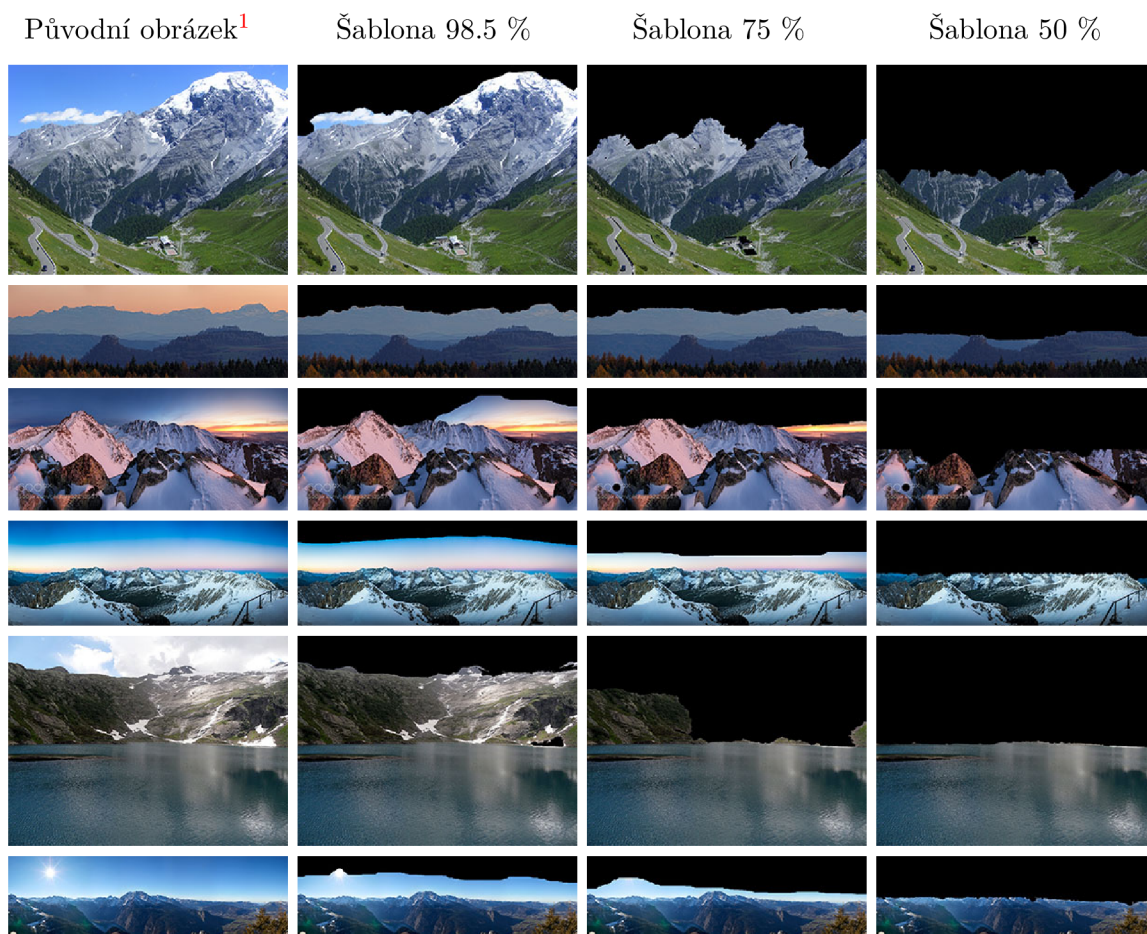
V druhém kroku určování křivky horizontu je výstupem již přesný odhad. Černá plocha je navzorkována po 1/20 šířky. Do obrázku se uživateli vykreslí opět lomená úsečka, tentokrát však s přesnější křivkou horizontu a 21 editovatelnými body. V tomto stádiu uživatel upraví křivku do finální podoby pro zpracování a odeslání do Locate.

3.4.3 Odhadnutí horizontu

Jak bylo popsáno výše, odhadnutí horizontu probíhá v python skriptu grabcut.py. V případě prvního odhadnutí křivky horizontu je pro zpracování v GrabCut vytvořena šablona velikosti obrázku. Pro každý pixel byly doplněny hodnoty pravděpodobného popředí nebo pravděpodobného pozadí. V šabloně byla vytvořena oblast pokrývající 98,5 % výšky obrázku (obdélníková plocha, zarovnaná ke spodní hranici obrázku) a označující pravděpodobné popředí. Jako pravděpodobné pozadí byl ponechán pouze úzký horizontální pruh u vrchní hranice obrázku. Velikost a umístění šablony bylo natrénováno experimentováním (obrázek 3.5). Experimentování probíhalo volbou různých výšek obdélníkové šablony a porovnáváním výsledků. Při trénování šablony jsem chtěl docílit, aby určený horizont nikdy nebyl průřezem hor. Konfigurace 98.5 % poskytovala nejlepší výsledky. Na základě této šablony je spuštěn GrabCut, který určí popředí obrázku. Určování popředí pomocí obdélníkové šablony je použito pro hrubý odhad křivky horizontu, protože výsledné popředí bylo často velmi nepřesné, zvláště u pozadí, které obsahují barevné přechody. Zpracování prvního kroku určování křivky horizontu trvalo pro 6 vzorků z 3.5 v průměru 20 122 ms. Měření probíhalo na počítači se specifikacemi: Windows 10 64-bit; 16 GB RAM DDR4; CPU Intel Core i7-4710HQ 2,5 GHz; 1 TB HDD 5400 RPM + 128 GB SSHD; na virtuálním počítači popsaném v 2.2.

Druhý krok určování křivky horizontu je založen na uživatelem upravené křivce hrubého odhadu v prvním kroku. Upravená křivka se exportuje ve formě souřadnic bodů, ze kterých

se lomená úsečka (horizont) skládá. Exportované body jsou předány opět python skriptu ke zpracování, avšak s jinými parametry. Pro přesný odhad horizontu jsem již nemohl využít jednoduchou obdélníkovou šablonu jako v prvním případě. Musel jsem vytvořit masku lépe představující oblast popředí obrázku. Pro masku bylo, stejně jako pro šablonu, vytvořeno pole o stejné velikosti jako obrázek. Levá, spodní a pravá hrana pole označují hranice pravděpodobného popředí. Horní hranici jsem musel získat z jednotlivých bodů reprezentovaných souřadnicemi, které bylo nutné propojit úsečkami. Spojení dvou bodů bylo nutné provést pomocí rasterizaci úsečky a tím získat souřadnice pro masku určující vrchní hranici pravděpodobného popředí. Masku je tedy tvořena polygonem, který má tři klasické strany obdélníku a vrchní strany, skládající se z lomené čáry.



Obrázek 3.5: Trénování šablony pro první krok určování křivky horizontu.

Pro rasterizaci jsem zvolil Bresenhamův algoritmus [16]. Vstupem jsou vždy dva body. Algoritmus pak vypočítává pixelovou cestu z prvního bodu ke druhému. Pro každý další pixel se určuje, na jakou pozici bude vytvořen s použitím vypočítaného prediktoru. Algoritmus

¹Zdroje obrázků (sestupně):

1. <http://www.fyqfz.com/287745-alps-magnificent.html>
2. Flickr image id: 23117882629, Flickr image secret: a7305f5903
3. Flickr image id: 23123881364, Flickr image secret: 3c90c86a6b
4. Flickr image id: 23273124169, Flickr image secret: 1b439821ea
5. Flickr image id: 23381402800, Flickr image secret: 6643268712
6. Flickr image id: 23622275662, Flickr image secret: 089de3e5d6

jsem implementoval v rámci python skriptu `grabcut.py`, kde probíhá celý proces rasterizace úseček mezi jednotlivými body uživatelem upravené křivky horizontu. Takto je získána vrchní hranice masky pravděpodobného popředí. Každý určený pixel hranice je ihned použit pro označení všech ostatních pixelů pod určeným pixelem hranice. Hotová šablona je pak znovu odeslána algoritmu GrabCut, který určí mnohem přesnější pozadí obrázku a vytvoří opět obrázek s pozadím označeným černou barvou. Zpracování druhého kroku určování křivky horizontu průměrně pro 6 vzorků trvalo 26 645 ms. To je dle očekávání více než u prvního kroku hlavně kvůli rasterizaci úseček. Měření probíhalo dle stejných specifikací jako v prvního kroku.

3.5 Použití Locate

Uživatel v druhém kroku určování upraví křivku horizontu do finální podoby. Následně je křivka připravena pro zpracování a odeslání do Locate. Křivka je opět exportována ve formě souřadnic bodů, ze kterých se lomená úsečka křivky horizontu skládá. Locate očekává na svém vstupu černobílý obrázek o šířce 1000 px, kde popředí je označeno bílou barvou a pozadí černou (obrázek 3.6). Kromě obrázku také potřebuje informaci o úhlu pohledu původního obrázku (tzv FOV - „Field of View“). Bylo nutné exportované souřadnice bodů přetransformovat do odpovídajícího černobílého obrázku, a z původního obrázku získat FOV data.



Obrázek 3.6: Vygenerovaný obrázek pro lokalizaci v Locate.

Transformace probíhá v ImagesModelu a je při něm využito procesu extrakce popředí z obrázku implementovaného v python skriptu. Jednotlivé souřadnice bodů, které uživatel v druhém kroku určování horizontu upravil do konečné podoby jsou exportovány a společně s obrázkem jsou předány ke zpracování python skriptu. Výstupem skriptu je opět obrázek jehož pozadí je černé a popředí v původních barvách. Zde se v ImagesModelu obrázek načte, a po sloupcích jsou kontrolovány pixely na černou barvu. Při prvním výskytu jiné barvy je zbytek pixelů v daném sloupci obarveno bílou. Výsledný obrázek je uložen do souboru. FOV informaci původního obrázku jsem získal pomocí programu ExifTool. Tímto je vše připraveno pro odeslání do Locate.

3.5.1 Locate

Locate [13] je projekt vedený Doc. Ing. Martinem Čadíkem, Ph.D. na Ústavu počítačové grafiky a multimédií. Projekt je zaměřen na vizuální lokalizaci v přírodě. Geo-lokalizace probíhá na základě informací obsažených v obrázku a jejím cílem je zjistit pozici a orientaci při pořizování fotografie [31]. Je pro to využito metod výpočetní fotografie a 3D digitální modely terénu DEM (Digital Elevation Model) a reprezentují digitální reliéf terénu. Po lokalizaci je možné s fotografií dále pracovat s pomocí DEM modelů.

Jednou z částí tohoto projektu také lokalizace fotografie na základě křivky horizontu a úhlu pohledu, kterou využívám v této bakalářské práci. Pro vzdálenou komunikaci webové aplikace a Locate mi vedoucí práce Ing. Jan Brejcha zprovoznil webovou službu, která je přímo napojená na zpracování v Locate. Webová služba je typu REST [20] a očekává HTTP požadavek s parametrem content-type: application/json. Obsahem požadavku musí být JSON ve formátu: { query : image_data, fov: float }, kde image_data je vygenerovaný černobílý obrázek zakódovaný do base64.

3.5.2 Propojení aplikace s Locate

Propojení Locate s aplikací, je implementováno v modelové třídě ImagesModel a probíhá po vytvoření černobílého obrázku a získání úhlu pohledu. Obrázek je zakódován do base64 formátu. Společně s FOV je vytvořen json řetězec a tvoří tak tělo požadavku. HTTP požadavek jsem realizoval pomocí php knihovny cURL (Client URL Library). Nejdříve je inicializováno nové sezení, kterému jsou připojeny parametry. Například typ HTTP požadavku, tělo požadavku, obsah HTTP hlavičky a podobné. Jakmile je požadavek vytvořen, je sezení spuštěno. Komunikace končí kontrolou úspěšnosti odpovědi.

```
$imageData = base64_encode(file_get_contents($image));
$data = array("query" => $imageData, "fov" => $exif);
$data_string = json_encode($data);

$curl = curl_init(ImagesModel::LOCATE_URL);
curl_setopt($curl, CURLOPT_CUSTOMREQUEST, "POST");
curl_setopt($curl, CURLOPT_POSTFIELDS, $data_string);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_HTTPHEADER, array(
    'Content-Type: application/json',
    'Content-Length: ' . strlen($data_string)
));

$result = curl_exec($curl);
```

Kód 3.8: Propojení aplikace s Locate.

3.6 Zpracování výsledků

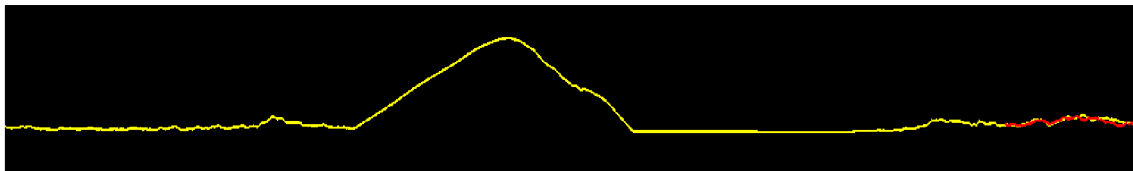
Odpověď přijde z webové služby Locate v zazipované formě v jednom souboru formátu json s informacemi o zeměpisné šířce, zeměpisné délce, souřadnice nalezeného panoramatu, souřadnice hledaného horizontu (vzhledem k nalezenému panoramatu) a úhel „yaw“ (udávající

úhel posunutí pohledu kamery vzhledem k panoramatu). Informace jsou zde pro každého nalezeného kandidáta. Jednotlivé výsledky jsou seřazeny podle velikosti chyby zarovnání horizontu. Počet bodů nalezeného panoramatu je vždy 8192 a má stejný rozměr jako šířka renderovaných obrázků panoramat ze Locate. Hodnoty bodů se pohybují v intervalu 0 - 4000, kde horní hranice je dána druhým rozměrem renderovaných obrázků. Výsledků z Locate přichází vždy 100. Aplikace uživateli však zpracuje pouze prvních 10 nejlepších kandidátů, protože zpracování trvá dlouhou dobu. Je-li uživatel přihlášen, jsou souřadnice nalezeného panoramatu a hledaného horizontu uloženy do databáze.

Musel jsem vyřešit jakým způsobem vizualizovat body nalezených panoramat. A jakým způsobem vizualizaci přeškálovat na šířku 800 pixelů, což je největší velikost, kterou je možné výsledky v aplikaci zobrazit (kvůli omezení maximální šířky aplikace). Problém jsem vyřešil způsobem, že se vytvoří prázdný obrázek velikosti 8192x4000 px vyplněný černou barvou. Body panoramatu se použijí pro vykreslení do obrázku žlutou barvou. Body hledaného horizontu se také vykreslí do obrázku červenou barvou. Obrázek se nechá vytvořit pro každého kandidáta. V šabloně se nechají obrázky zobrazit s pevnou šířkou 800 px pomocí CSS stylů. Úlohu přeškálování tedy vykoná prohlížeč.

3.6.1 Zobrazení výsledků

Každý výsledek se tedy skládá z černého obrázku, do kterého je vykreslena žlutá křivka přes celou šířku obrázku a červená křivka zahrnující jen část šířky obrázku. Žlutá křivka představuje nalezené panorama. Červená křivka představuje původní uživatelem hledaný horizont. Správný výsledek by tedy měl vypadat tak, že žlutá křivka bude mít v místě červené křivky stejný tvar jako červená křivka.



Obrázek 3.7: Ukázka výsledku

Výsledky jsou zobrazeny na jedné stránce pod sebou. Jako první je zobrazena legenda, která má uživateli umožnit rychlejší pochopení výsledků a hlavně významu černých obrázků se žlutými a červenými křivkami. Každý výsledek se skládá ze zobrazeného vygenerovaného obrázku a z části, kde se vykresluje Google mapa zaměřená na zeměpisné souřadnice, které jsou rovněž zobrazeny uživateli.

Pro zobrazení mapy je využito Google Maps API [11], a to konkrétně možnosti vestavěné mapy „embedded“. Pro použití Google Maps API jsem musel pro aplikaci vygenerovat Google API klíč, díky kterému pak lze službu Google Maps využít přímo z aplikace. Vestavěná mapa od Google lze použít vložením jednoduchého HTML tagu `<iframe>` a do něj doplnění atributů jako: šířka, výška a ohraničení. Nejdůležitějším atributem je však zdroj, který se vytváří pomocí přesně daného schématu pro URL adresu 3.9. Do adresy se doplňuje Google Maps API klíč, zeměpisná šířka a délka, velikost zoomu na mapě a typ mapy.

```
https://www.google.com/maps/embed/v1/view
?key=API_KEY
&center=LATITUDE, LONGITUDE
&zoom=ZOOM_VOLUME
&maptype=MAP_TYPE
```

Kód 3.9: Schéma URL adresy.

3.7 Uživatelské sekce

Uživatelské sekce jsou zpřístupněné pouze registrovaným a přihlášeným uživatelům a nabízejí drobná rozšíření a výhody oproti neregistrovaným. V uživatelské sekci má uživatel možnost změnit údaje o svém účtu nebo změnit heslo. Dalšími podsekcemi jsou Moje poslední aktivita a Moje nahrané obrázky. Mezi hlavní výhody patří, že uživatelem nahrané obrázky nejsou vkládány do veřejné složky, ale ukládají se na server pro možnost dalšího použití uživatelem. Další výhodou je ukládání vyhledávání uživatele.

3.7.1 Moje poslední aktivita

Sekce poskytuje rozhraní pro prohlížení uživatelské historie vyhledávání. Jsou zde v levé části uloženy názvy jednotlivých vyhledávání. Při výběru konkrétního vyhledávání je zobrazen jeho určený horizont a případné výsledky podobně jako při zobrazení výsledků vyhledávání. Uživatel se tak může vrátit ke svým předešlým vyhledáváním a může je porovnávat mezi sebou.

3.7.2 Moje nahrané obrázky

Zde se ukládají uživatelem nahrané obrázky, které může znovu použít pro další vyhledávání. V návrhu aplikace zde byla nastíněna možnost, kdy by uživatel mohl obrázky označit jako veřejné a zpřístupnit je tak ostatním uživatelům ve veřejné složce. Dále také možnost obrázky řadit a filtrovat. Tyto funkcionality však přesahují rámec této práce, nejsou tedy ve výsledné aplikaci implementovány.

Kapitola 4

Testování a srovnání

Kapitola testování a srovnání pojednává o závěrečné fázi testování hotového programu. Příprava testování se skládala z definice testování a výběru uživatelů. Dalším krokem je provedení testování. Finální částí testování je jeho vyhodnocení a získání podnětů pro zlepšení aplikace či upozornění na chyby.

4.1 Definice testování

Pro zjištění a zaručení kvality byl produkt otestován. Pomocí testování aplikace byla zjištěna slabá místa a podněty pro vylepšení. Testování jsem, stejně jako v případě testování designu, zvolil kvalitativní s uživatelem a skládá se ze čtyř kroků, které uživatele provedou nejdůležitějšími aspekty aplikace.

- 1. Úkol:** Vyberte obrázek ze služby Flickr a nechte si určit křivku horizontu. Křivku upravte a nechte vyhodnotit v Locate. Prohlédněte si výsledky a kriticky ohodnoťte správnost výsledků.
- 2. Úkol:** Registrujte se do aplikace. Přihlaste se pod novým účtem.
- 3. Úkol:** Vyberte obrázek z veřejné složky a nechte si určit křivku horizontu. Křivku upravte a nechte vyhodnotit v Locate. Prohlédněte si výsledky a kriticky ohodnoťte správnost výsledků.
- 4. Úkol:** Prohlédněte si historii vašich vyhledávání.

4.1.1 Výběr uživatelů

Výběr uživatelů byl stejně jako v testování mockupu v kapitole 2.5 zaměřen na uživatele, kteří spadají do definovaných skupin uživatelů v kapitole 2.1.1. Všichni uživatelé museli nejdříve vyplnit screener – dotazník, který má předem definované přijatelné odpovědi. Pro účely závěrečného testování jsem vybral celkem 10 testovaných uživatelů. Polovinu tvořila pětice uživatelů, kteří testovali mockup aplikace před půl rokem. Druhou polovinu tvořili nezaujatí uživatelé, kteří aplikaci poprvé uviděli až při závěrečném testování. Testované respondenty jsem vybral tímto způsobem, neboť mi přišlo velmi zajímavé srovnat, jak si vedly obě skupiny navzájem. Jedna skupina uživatelů se s aplikací ve své podstatě

setkala, zatímco druhá ne. Uživatelé mají různé úrovně počítačové gramotnosti od začátečnicků až po pokročilé. Věkový interval uživatelů je od 20-65 let.

4.2 Proces testování

Testování aplikace probíhalo prezenční formou. V úvodu testování jsem každému respondentovi vysvětlil jak bude probíhat testování. Stručně jsem uvedl jakým cílem se celý projekt zabývá. Při testování jsem zaznamenával video obrazovky monitoru pro možnost zpětného stopování uživatelových kroků. Pořízená videa mi umožnila lepší rozbor a vyhodnocení testování. Uživatelům byl vždy předložen úkol, který se následně pokoušeli splnit. V případě, že se uživatel ztratil a nevěděl co dále dělat, jsem poradil, jak postupovat. Po splnění všech úkolů byl uživatelům předložen stejný dotazník, jako v testování mockupu. K potřebám vyplnění dotazníku měli uživatelé neomezený přístup k aplikaci. Dotazník se opět skládal z otevřených otázek, z dichotomických otázek („ano/ne“), a z otázek Likertovy škály (metoda měření postojů) jako v kapitole 2.5.2.

4.3 Vyhodnocení testování

Vyhodnocení probíhalo na základě mých postřehů během testování, z vyplněných dotazníků a ze zpětného shlednutí záznamů obrazovky z průběhu testování. Tyto cenné informace se týkají částí aplikace, ale i jednotlivých detailů. Testování probíhalo na předposlední verzi aplikace. V jednom případě se při testování objevila vnitřní chyba, která byla po testování opravena, abych zamezil opětovnému výskytu při testování dalších uživatelů. Chyba se samozřejmě podepsala na uživatelových celkových dojmech z aplikace.

Uživatelé měli z testování různé pocity, většinou dobré a příjemné. Také převládala zvědavost a zájem uživatelů. Výjimečně se uživatelům líbil také neformální přístup při testování. Jeden uživatel měl negativní a nepříjemný pocit z prostředí operačního systému Linux Ubuntu. Tento operační systém však patří virtuálnímu počítači, na kterém je aplikace lokálně spuštěna, a na kterém také probíhalo testování (specifikace v kapitole 2.2).

Testovaným uživatelům se nejvíce líbila jednoduchost ovládání, rychlost použití a přehlednost celé aplikace. Dle názorů aplikace vede uživatele, co má dělat a poskytuje rychlé zorientování. Dále uživatele zaujala možnost určení místa pořízení fotografie. Grafická stránka aplikace také nezůstala bez povšimnutí. Většinou uživatelů se líbil design aplikace a rozmístění ovládacích prvků. Pozitivní ohlasy získaly také ukazatele načítání při určování křivky horizontu a zpracování výsledků. Téměř všem se velmi zamlouval celkový nápad a uplatnění aplikace v reálném životě.

Největší problémy, které museli uživatelé překonávat byly do určité míry individuální. Jeden uživatel neměl problémy žádné. Dalšímu uživateli nešel napsat znak zavináče na OS Linux (nepodporovaná zkratka Alt + 64). Jak už jsem uvedl v odstavci výše jeden uživatel se setkal i s technickými problémy. Většinou se ale individuální problémy týkaly spíše méně zkušených uživatelů. Tyto individuální problémy však nejsou příliš relevantní. Mnohem zajímavější a významnější jsou problémy, které byly pro většinu společné, a se kterými se potýkalo více než pět lidí.

Nepochopení výsledků (majoritní problém) Výsledky jsou zobrazeny uživateli formou žluté křivky, která reprezentuje nalezené panorama a formou červené křivky, reprezentující uživatelův vyhledávaný horizont. Bohužel i přes poměrně jednoduchou legendu,

která by měla vše potřebné pro pochopení výsledků vysvětlit, a která se zobrazuje v horní části stránky s výsledky, ji 7 z 10 uživatelů přehlédlo nebo si jí nevšimlo vůbec. Uživatelé z velkého očekávání mnohem raději přistupovali ihned k výsledkům, kterým však nedokázali porozumět. Zájem uživatele se tak většinou přímo soustředil na výsledky a význam výsledků zůstal nepochopen.

Řešením tohoto problému by bylo přidání tutoriálu na čtení výsledků. Odeslání finálního horizontu a zpracování výsledků trval poměrně dlouhou dobu 2-4 minuty (což bylo po testování vylepšeno na 1-3 minuty). Během této doby by se mohl uživatel v jednotlivých krocích zobrazit jednoduchý návod, který by uživatel rozhodně nepřehlédnul. Na výsledky by se tak díval již informován a využil by se tak užitečně čas při čekání na výsledky. Další možností je legendu pro vysvětlení zvýraznit například jinými barvami a vizuálně ji tak odlišit od výsledků. Případnou alternativou by bylo také přidání komentářů ke každému výsledku nebo malou verzi legendy přímo do výsledků. Uživatelé by tak v každém výsledku měli po ruce pomocnou legendu s vysvětlivkami.

Výběr fotografie a určování křivky horizontu (majoritní problém) Při využití možnosti výběru fotografie ze služby Flickr nebo z veřejné složky se uživatelé po rozkliknutí miniaturní fotografie zobrazí větší náhled. Na pravé straně je šipka s možností pokračovat, kde si uživatel potvrdí výběr fotografie a započne proces určování křivky horizontu. Po kliknutí na odkaz pokračovat však chvíli trvá než se příslušný obrázek stáhne ze služby Flickr, než proběhne jeho zpracování a než se obrázek použije pro prvotní automatické určení hrubého odhadu křivky horizontu. Může se zde čekat 5-20 sekund a uživatel zde po kliknutí nemá žádnou odezvu, že se skutečně v pozadí něco děje, kromě načítacího kolečka prohlázeče (čehož si uživatelé prakticky nevšimnou). Uživatelé tak mají tendenci na odkaz klikat opakovaně.

Řešením tohoto problému by bylo přidání vlastního načítacího kolečka a popisného textu ihned pod odkaz, podobně jako tomu je při obou krocích určování křivky horizontu. Uživatelům tato zpětná vazba poskytuje dostatek vůle čekat než se zpracování dokončí.

Přístup do uživatelské sekce (majoritní problém) Pro přístup do uživatelské sekce jsem vytvořil odkaz dostupný v hlavičce vedle odkazu pro odhlášení vždy se jménem konkrétního uživatele. To se bohužel ukázalo jako nešťastné řešení neboť 7 z 10 uživatelů nenapadlo, že jeho jméno může být bránou do uživatelské sekce. Uživatelé tak vůbec nevěděli, kde v aplikaci hledat možnost navštívit uživatelskou sekci.

Řešením by mohlo být přesunutí jména uživatele mimo odkaz a na jeho místo dosadit „Můj profil“. Uživatelé jsou již zvyklí hledat uživatelské nastavení a funkce pod tímto popiskem vzhledem k jejich zkušenostem ze sociálních sítí nebo jiných webových aplikací.

Technické problémy (střední problém) Aplikace v průběhu testování jednoho uživatele selhala na vnitřní chybě, kterou jsem po testování opravil. Další chyba se objevila při zadání špatného hesla uživatele, kdy se namísto varovné hlášky uživateli zobrazila Nette výjimka s ladícími informacemi.

Tuto chybu jsem po testování ošetřil. Zbytek technických chyb by mělo odstranit překlenutí aplikace z testovacího do produkčního režimu.

Výběr fotografie (střední problém) Polovina uživatelů se někdy v aplikaci ztrácela. Uživatelé si nebyli úplně jistí, kde se právě nacházejí nebo co přesně by měli udělat.

Ve výběru fotografie ze služby Flickr nebo z veřejné složky se zobrazí větší verze obrázku s odkazem pokračovat na pravé straně od obrázku. Tři uživatelé si přesto bočního odkazu vůbec nevšimli a nebyli si jistí, co dále dělat nebo jakým způsobem zde určovat křivku horizontu. Lightbox, který obrázek otevírá, a do kterého je také doplněn odkaz pro pokračování nezobrazuje žádné další informace (kromě názvu obrázku).

Řešením problému vyznat se v tomto dialogovém okně, pro uživatele, kteří se s funkcionalitou Lightboxu ještě nesetkali, by bylo přidáním popisného nadpisu nad fotografii, která by uživatele rychle navedla co zde mají udělat. Dále je vhodné šipku s odkazem pokračovat na pravé straně zvýraznit, aby ji uživatelé tak snadno nepřehlédli.

Určování křivky horizontu (střední problém) Podobně jako ve výběru fotografie zde uživatelé měli občasné problémy pochopit, že se určování křivky horizontu skládá ze dvou kroků. Dále že v prvním kroku stačí horizont označit jen zhruba, a že v druhém kroku bude již určování detailnější. V obou krocích jsou zobrazeny popisné nadpisy a texty s instrukcí, co mají uživatelé udělat. Téměř všichni uživatelé si však popisných textů s instrukcí nevšimli i přesto, že je barevně odlišen od zbytku aplikace.

Řešením tohoto problému by bylo zobrazením dialogového okna s instrukcí přes celou obrazovku, která by uživatele donutila si text přečíst a až následně přistoupit k manipulaci s křivkou horizontu. Za zvážení by také stálo zvětšit a zvýraznit popisný text.

Mapy u výsledků (střední problém) Ke každému výsledku je zobrazena Google mapa zaměřená na zeměpisné souřadnice výsledku příslušící. Rozhraní Google API však nemá velkou podporu pro zobrazování připínáčku a popisků ve vestavěné verzi „embedded“ Google mapy. Mapa se tedy nachází s geografickými popisky, ale to je vše. Uživatelé tak měli problémy zjistit co je na mapě vlastně zobrazeno.

Řešením by bylo minimálně manuálním přidáním připínáčku doprostřed mapy a přidáním popisku. Nejlepším řešením by však bylo využití práce kolegy Petra Bečky [15], který implementoval zobrazení vestavěné mapy (z OpenStreetMap) společně s připínáčkem uprostřed a vykreslenou výsečí, kam je pohled mířen. Toto rozšíření by bylo pro doplnění výsledků v této aplikaci dobrým rozšířením, protože by tak uživatel ihned věděl, na kterou světovou stranu je vlastně pohled fotografie mířen vzhledem k pozici digitálního fotoaparátu.



Obrázek 4.1: Ukázka zobrazení mapy s pomocí řešení od Petra Bečky

Manipulace s křivkou horizontu (minoritní problém) S ovládáním křivky horizontu neměli uživatelé problémy. Princip fungování přidání a odebrání bodu však 4 z 10 uživatelů na první pokus nepochopili. Ostatním uživatelům nedělal problém pochopit princip

přidání a odebrání bodů. Nyní přidání bodu funguje způsobem, že uživatel klikne na tlačítko přidání bodu a následně klikne do obrázku, kde se na místo kurzoru přidá nový bod. Podobně funguje tlačítko odebrání bodu, kde se nejdříve klikne na tlačítko Odebrat bod a následně se v obrázku kliknutím vybere bod, který se má odebrat. Menšina uživatelů, která nejdříve tento systém nepochopila by raději uvítala možnost ihned po kliknutí levým tlačítkem na obrázek na místo kurzoru vytvořit bod a pravým tlačítkem bod odebrat. S přidáním popisného obrázku myši a vysvětlením funkcionality levého a pravého tlačítka myši v obrázku by to mohlo být dobrý řešením, které by mohlo být méně problematické.

Rychlost vyhledávání (minoritní problém) Zpracování finální křivky horizontu, zpracování v Locate a interpretování výsledků v aplikaci je velmi časově náročné. Běžně od 150-300 sekund. To je až příliš dlouhá doba, než na kterou jsou dnešní internetoví uživatelé zvyklí. Celý tento proces by tedy potřeboval nějakým způsobem zrychlit. Zrychlení však není triviální neboť samotné vyhledávání v Locate trvá 60-90 sekund. Musíme vzít v úvahu několik skutečností. Finální křivka horizontu, kterou uživatel upravil do konečné podoby, je použita pro třetí a poslední určení křivky horizontu pro vyhledávání v Locate. Zpracování v Locate je velmi výpočetně náročné. Výsledky jsou z Locate přenášeny v json souboru, který má velikost běžně kolem 14 MB, takže rychlost závisí na uživatelově připojení k internetu. Určitého drobného zrychlení bylo docíleno zazipováním souboru s výsledky pro přenos, kde se dokázal snížit objem přenášených dat ze 14 MB na 3 MB. Dále jsou výstupy z Locate interpretovány vytvořením 8192 pixelů širokých obrázků, které jsou následně zmenšeny na rozměry aplikace.

Existuje více možností zrychlení, například by se mohly přeskákat pozice bodů a zamezilo by se tak vykreslování zbytečně velkých výsledků. Další možností je zrychlení na straně serveru pomocí lepšího algoritmu pro geometrickou verifikaci horizontů. Jinou možností zrychlení stránkování, kde by se uživateli zobrazilo například jen prvních pět výsledků a další by se renderovaly z výsledků až na dalších stránkách.

Uživatelé v dotaznících a ústní formou uváděli mnoho námětů a tipů jak by šla aplikace vylepšit.

- Možnost zoomu fotografie při určování křivky horizontu.
- Možnost více pracovat s mapou u výsledků - možnost náklonu mapy podobně jako v Google Earth a možnost ověřit si křivku horizontu i touto formou.
- Obrázek u určování křivky horizontu zvětšit. Přidat více základních bodů v druhém kroku určování křivky horizontu.
- Vypisovat statusy při zpracování a podrobněji tak informovat uživatele co se právě děje.
- Přidat jazykové mutace.
- Na úvodní stránce vytvořit slider ukazující postupně v bannerech screeny z průchodu vyhledáváním. Dát již na úvodní stránce více najevo k čemu je aplikace pro uživatele užitečná.
- Přidat stránku s FAQ (Frequently Asked Questions) - Stránka s nejčastějšími otázkami uživatelů a odpověďmi.

- Zvýraznit více možnost výběru vyhledávání v sekci Moje poslední aktivita.
- U registrace přidat popis co uživatel registrací získá.
- Přidat popis co je veřejná složka.

Webová aplikace pro vyhledávání panoramat měla celkově u uživatelů úspěch. Testování uživatelé si myslí, že by aplikace mohla lidi zaujmout vzhledem ke stále rostoucímu zájmu o cestování. Dále, že je poměrně široký okruh možností, kde by se dala aplikace využít. Také dle uživatelů by aplikace by mohla usnadnit plánování při turistice nebo pomoci při úpravě fotografií. Kvality aplikace také dokládají tabulky 4.1 a 4.2. Na otázky v tabulce uživatelé odpovídali v dotaznících.

	ANO	NE
Byla pro vás aplikace srozumitelná?	9	1
Po této zkušenosti, použili byste znovu tuto aplikaci?	9	1
Myslíte, že je aplikace zajímavá pro veřejnost?	10	0

Tabulka 4.1: Dichotomické otázky dotazníku

	Ano	Spíše ano	Ani ano ani ne	Spíše ne	Ne
Jak moc jste spokojen s funkčností?	4	3	2	1	0
Jak moc jste spokojen se vzhledem?	5	4	0	1	0
Jak jste spokojen se službami aplikace?	3	6	0	1	0

Tabulka 4.2: Otázky Likertovy škály v dotazníku

4.3.1 Srovnání skupin uživatelů

Testování se zúčastnili uživatelé, kteří by se dali rozdělit do dvou skupin. První skupinu tvoří stejní uživatelé, kteří před šesti měsíci testovali také návrh (mockup) aplikace. Druhá skupina je tvořena uživateli, kteří se s aplikací setkali poprvé až při samotném testování. Když porovnáme obě skupiny obecně, tak nelze s určitostí tvrdit, že jedna či druhá skupina měla s aplikací větší problémy. V obou skupinách byli uživatelé, kteří neměli problémy aplikací použít. A na druhou stranu v obou skupinách byli uživatelé jímž trvalo použití delší dobu. Skupina, která se již s aplikací setkala neměla prakticky žádnou výhodu oproti druhé skupině, protože šest měsíců je až příliš dlouhá doba, než aby si uživatelé něco z dřívějšího testování pamatovali.

Kapitola 5

Závěr

Cílem této práce bylo vytvořit aplikaci poskytující přívětivé a jednoduché prostředí pro vyhledání panoramat. Vyhledávání spočívá v lokalizaci panoramatu pohoří fotografie na základě křivky horizontu, kterou uživatel s pomocí aplikace určí vhodným nástrojem. Byla vypracována rešerše existujících řešení a bylo zjištěno, zda-li už takové aplikace existují a jak se od nich bude aplikace odlišovat. Před implementací bylo nutné se seznámit s technologiemi potřebnými pro vývoj aplikace. Byla navržena architektura a uživatelské rozhraní aplikace. Webová aplikace byla implementována a propojena s lokalizačním systémem. Na hotové aplikaci jsem provedl uživatelské testování a získal jsem informace o slabých místech aplikace a dalších možnostech pro vylepšení.

5.1 Zhodnocení dosažených výsledků

Vytvořil jsem webovou aplikaci, která podle mého názoru splňuje cíle uvedené v zadání této práce. Webová aplikace poskytuje jednoduché a příjemné prostředí pro uživatele. Aplikace prošla uživatelským testováním, kde si uživatelé aplikaci vyzkoušeli a kriticky hodnotili výsledky aplikace i jednotlivé části. Výsledky testování tak ukázaly, že je aplikace v dobrém stavu. Nicméně obsahuje několik kritických bodů, které je třeba vylepšit.

5.2 Další vývoj projektu

Celá aplikace by samozřejmě mohla být dále vyvíjena. Výsledky testování odhalily nedostatky, ve kterých by aplikace mohla být vylepšena. Nejvíce by se dalo zaměřit na podporu pro snadnější pochopení výsledků. Začátek budoucího vývoje by se měl nejdříve zaměřit na body ve vyhodnocení uživatelského testování aplikace.

Po doplnění těchto oprav nabízí aplikace možnosti pro další vývoj. Například převést jednotlivé kroky (nebo celou aplikaci) do ajaxové verze, což by dodalo aplikaci moderní nádech. Za zvážení by stálo nechat vypracovat profesionální webdesign grafikem. Dalšími možnostmi by bylo například vylepšení určování křivky horizontu způsobem, že by se dva kroky určování křivky horizontu omezily jen na jeden za použití lepší segmentace. Pro určení křivky horizontu by také mohly být dostupné další možnosti jak s křivkou pracovat. Možností rozšíření by mohlo být použití lokalizačního algoritmu sémantických segmentů, kde se využívá označení vrstev obrázku s pomocí topografických map, podobně jako v [14].

Literatura

- [1] The Best PHP Framework for 2015: SitePoint Survey Results [online]. <http://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/>, 2000 [cit. 2016-01-31].
- [2] PHPMailer - A full-featured email creation and transfer class for PHP [online]. <https://github.com/PHPMailer/PHPMailer>, 2001 [cit. 2016-02-22].
- [3] Nette Framework - Přihlašování & oprávnění uživatelů [online]. <https://doc.nette.org/cs/2.3/access-control>, 2008 [cit. 2016-01-31].
- [4] Nette Framework [online]. <https://nette.org/cs/>, 2008 [cit. 2016-01-31].
- [5] BlueStacks play bigger [online]. <http://www.bluestacks.com/cs/index.html>, 2016 [cit. 2016-01-31].
- [6] Pencil Project [online]. <http://pencil.evolus.vn/>, 2016 [cit. 2016-02-14].
- [7] Flickr API - flickr.photos.search [online]. <https://www.flickr.com/services/api/flickr.photos.search.html>, 2016 [cit. 2016-02-29].
- [8] Flickr API - Photo Source URLs [online]. <https://www.flickr.com/services/api/misc.urls.html>, 2016 [cit. 2016-02-29].
- [9] Flickr API [online]. <https://www.flickr.com/services/api/>, 2016 [cit. 2016-02-29].
- [10] OpenCV [online]. <http://opencv.org/>, 2016 [cit. 2016-03-06].
- [11] Google Maps APIs [online]. <https://developers.google.com/maps/>, 2016 [cit. 2016-05-02].
- [12] Adermann, N.; Boggiano, J.: Composer [online]. <https://getcomposer.org/>, 2012 [cit. 2016-02-21].
- [13] Čadík, M.: LOCATE - vizuální lokalizace v přírodě [online]. http://cadik.posvete.cz/locate/locate_cz.pdf, 2011 [cit. 2016-04-28].
- [14] Baatz, G.; Saurer, O.; Köser, K.; aj.: Leveraging Topographic Maps for Image to Terrain Alignment. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization Transmission*, Oct 2012, ISSN 1550-6185, s. 487–492, doi:10.1109/3DIMPVT.2012.33.

- [15] Bečka, P.: Sada webových nástrojů pro geolokalizační systém, bakalářská práce. 2016 [cit. 2016-05-03].
- [16] Bresenham, J. E.: Algorithm for computer control of a digital plotter. *IBM Systems Journal*, ročník 4, č. 1, 1965: s. 25–30, ISSN 0018-8670, doi:10.1147/sj.41.0025.
- [17] Chinneck, J. W.: Practical Optimization: A Gentle Introduction - Chapter 9: Maximum Flow and the Minimum Cut [online]. <http://www.sce.carleton.ca/faculty/chinneck/po.html>, 2006 [cit. 2016-03-07].
- [18] Coulter, D.: phpFlickr [online]. <https://github.com/dan-coulter/phpflickr>, 2014 [cit. 2016-02-29].
- [19] Dhakar, L.: Lightbox2 [online]. <https://github.com/lokes/lightbox2>, 2008 [cit. 2016-02-29].
- [20] Fielding, R.; Taylor, R.: Principled design of the modern Web architecture. In *Proceedings of the 22nd international conference on Software engineering*, Jun 2000, ISBN 1-58113-206-9, s. 4007–416, doi:10.1145/337180.337228.
- [21] Harvey, P.: ExifTool [online]. <http://www.sno.phy.queensu.ca/~phil/exiftool/>, 2016 [cit. 2016-03-07].
- [22] Hendl, J.: *Kvalitativní výzkum : základní teorie, metody a aplikace*. Portál, 2008, ISBN 978-80-7367-485-4.
- [23] James, R.; Jacobson, I.; Booch, G.: *The Unified Modeling Language Reference Manual*. Pearson Higher Education, 2004, ISBN 0-321-24562-8.
- [24] Kaliski, B.: RFC 2898: Password-Based Cryptography Specification [online]. <https://www.ietf.org/rfc/rfc2898.txt>, 2000 [cit. 2016-02-22].
- [25] Krug, S.: *WEB DESIGN - Nenutte uživatele přemýšlet*. Computer Press, 2006, ISBN 80-251-1291-8.
- [26] Lavrenov, A.: KonvaJS [online]. <http://konvajs.github.io/>, 2015 [cit. 2016-03-07].
- [27] Lodl, J.: Android poráží iOS. Zbytek nestojí za řeč [online]. http://mobil.idnes.cz/android-porazi-ios-Onv-/mob_tech.aspx?c=A130131_212412_mob_tech_lhc, 2013 [cit. 2016-05-08].
- [28] Marsh, M.: Implementing the „GrabCut“ Segmentation Technique as a Plugin for the GIMP [online]. <http://www.cs.ru.ac.za/research/g02m1682/>, 2016 [cit. 2016-03-06].
- [29] Reichel, J.: *Kapitoly metodologie sociálních výzkumů*. Grada, 2009, ISBN 978-80-247-3006-6.
- [30] Rother, C.; Kolmogorov, V.; Blake, A.: GrabCut - Interactive Foreground Extraction using Iterated Graph Cuts [online]. <https://cvg.ethz.ch/teaching/cv1/2012/grabcut-siggraph04.pdf>, 2016 [cit. 2016-03-06].

- [31] Saurer, O.: Image Based Geo-localization in the Alps. In *International Journal of Computer Vision*, 2015, s. 1–13.
- [32] Unger, R.; Chandler, C.: *A Project Guide to UX Design: For user experience designers in the field or in the making*. New Riders, 2012, ISBN 978-0-321-60737-9.

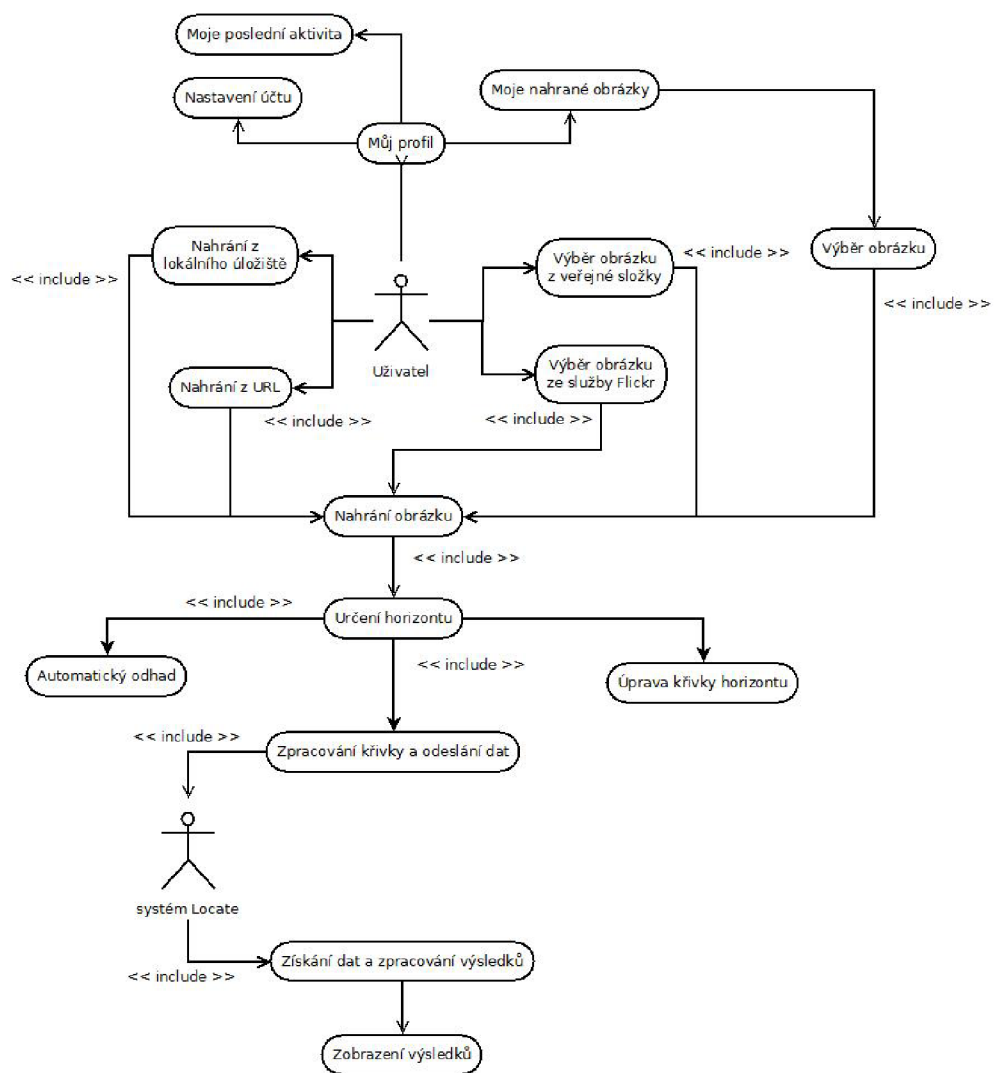
Příloha A

Obsah CD

- Zdrojové kódy aplikace.
- Návod pro instalaci.
- Mockup aplikace.
- Vyplněné dotazníky testování designu.
- Vyplněné dotazníky testování aplikace.
- Snímky aplikace.
- Zdrojové kódy dokumentace
- Diagramy návrhu aplikace

Příloha B

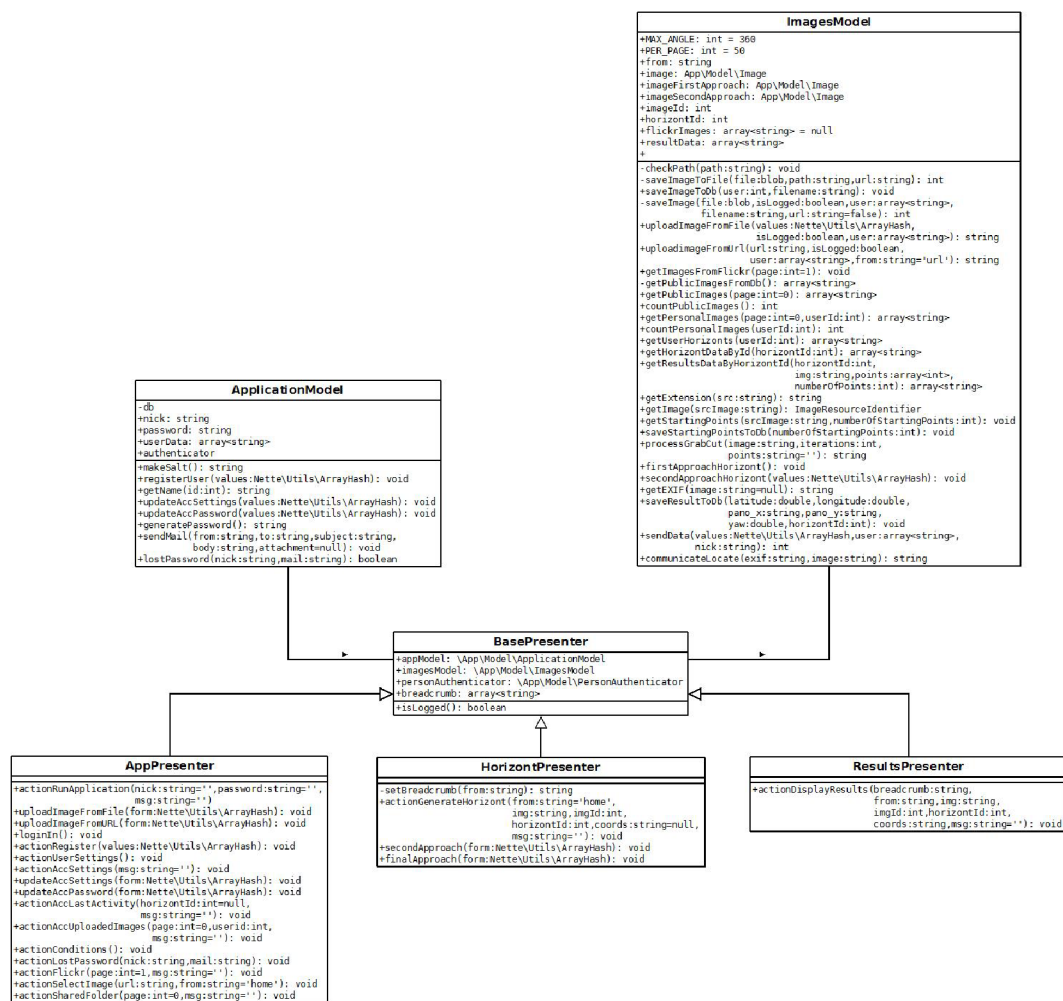
Diagram případů užití



<< include >> značí operaci, kterou systém automaticky provede v reakci na událost

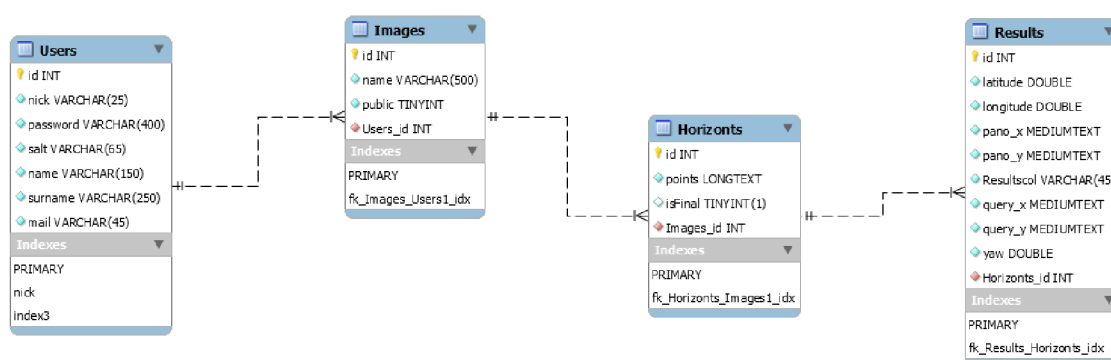
Příloha C

Diagram tříd



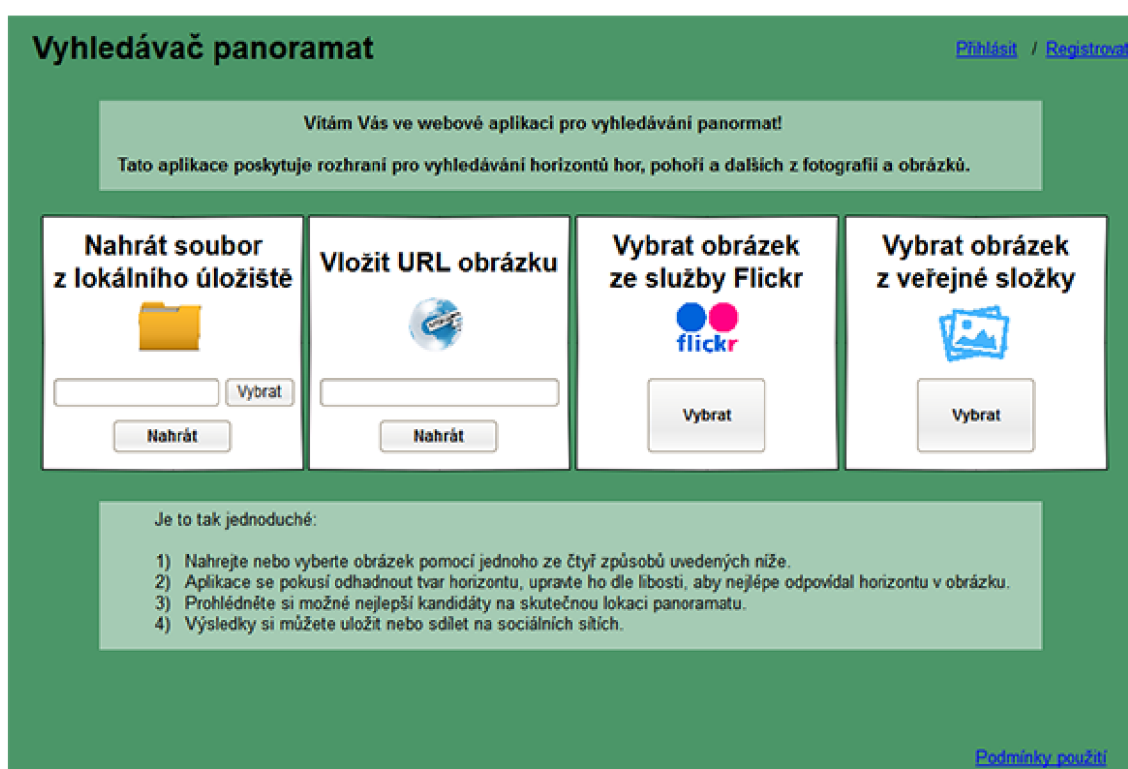
Příloha D

ER diagram



Příloha E

Ukázky mockupu aplikace




Vyhledávač panoramat

Přihlášen: [Jan Novák](#)

Výběr obrázku · Určení horizontu

+

-



Zpět

Pokračovat

Přidat bod

Odebrat bod

Vyhledávač panoramat

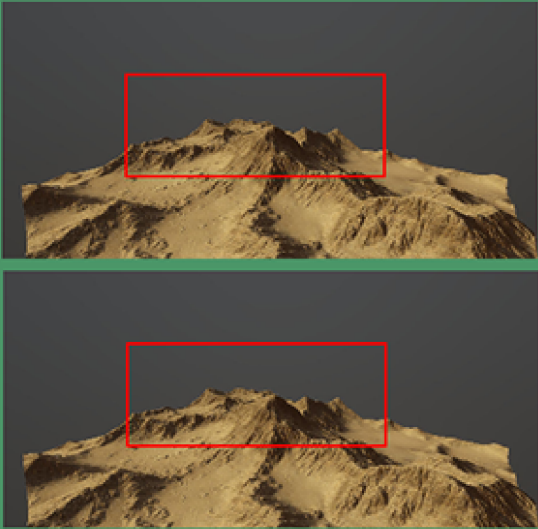
Přihlášen: [Jan Novák](#)

Výběr obrázku · Určení horizontu · Výsledky

Nejllepší kandidáti

[Zobrazit na Google mapě](#)

Zpět



[f](#)

[t](#)

[g+](#)

Export do PDF

Odeslat emailem

Příloha F

Vyplněné dotazníky testování mockupu

Finální dotazník	
Při vyplňování dotazníku můžete libovolně procházet aplikací a máte na něj libovolný čas.	
1. Jaký jste měl z testování pocit?	NEJISTOTA DO ČEHO JDU, PO PRVNÍM ÚKOLU JSE LOGICKY ODVOJOVALA CO MÁM DĚLAT, PŘI DALŠÍCH ÚKOLECH JSEM MĚ VĚDĚLA CO A JAK
2. Co se vám na této aplikaci líbilo?	ZAJÍMAVÝ NÁPAD, MŮŽE POMOCT UČIT NEZNÁMÉ FOTOGRAFIE
2.5 Proč? (odůvodněte prosím vaši odpověď)	MŮŽE BÝT UŽITEČNÁ PRO VĚTNÝ POČET LIDÍ MILUJÍCÍCH HORY A NEJEN JE
3. Jaké největší problémy jste musel při plnění úkolů překonávat?	PŘI PRVNÍM A DALŠÍM ÚKOLU JSEM BYLA NEJISTÁ A TĚ, ŽE JDE ODEHO A NEPŘIPLOU VĚŘIT, ZNEJISTŮVALA MĚ PRA ^{(SE} _{(VĚŘ} (202)
4. Co by jste na této aplikaci změnil?	GRATULACE OKOL-RODKUD
4.5 Proč? (odůvodněte prosím vaši odpověď)	ZAJÍMAVĚJŠÍ, VÍCE ZAJÍMAVĚ

Finální dotazník

Při vyplňování dotazníku můžete libovolně procházet aplikací a máte na něj libovolný čas.

1. Jaký jste měl z testování pocit?

ani pozitivní ani negativní, když to byla mává zkušenost
a je důležité pochopit proč se nepříjemným pocitem
bylo vedeno, že jde o testování a to je to neobvyklé

2. Co se vám na této aplikaci líbilo?

spíše práce s horizontem, takže se mi hodně líbilo
a v jiných editacích (afektografii - např. s koncem) ...

2.5 Proč? (odůvodněte prosím vaši odpověď)

myslím si že je to individuálnější a tím pádem
i snazší pro každou zkušenou kterou s přítelkou
má se pracuje

3. Jaké největší problémy jste musel při plnění úkolů překonávat?

že jsem např. při vytváření uživatelské horizontu
nevěděl proč se chová (co se mi na stránce zobrazuje)

4. Co by jste na této aplikaci změnil?

na stránce, asi více popisek k tomu co se na některých
částech stránky zobrazuje (např. výsledky horizontu)

4.5 Proč? (odůvodněte prosím vaši odpověď)

protože by mi to usnadnilo práci s aplikací a
případně by mi to maseklo dále, když bych
a v nějaké oblasti ztratil

Finální dotazník

Při vyplňování dotazníku můžete libovolně procházet aplikací a máte na něj libovolný čas.

1. Jaký jste měl z testování pocit?

• Příjemný

2. Co se vám na této aplikaci líbilo?

Šířka na detail

2.5 Proč? (odůvodněte prosím vaši odpověď)

zjednodučené a vizuálně

3. Jaké největší problémy jste musel při plnění úkolů překonávat?

Ložní deska mýš od používání

4. Co by jste na této aplikaci změnil?

Barvy rozložení. (všechny barvy vyměnit)

4.5 Proč? (odůvodněte prosím vaši odpověď)

~~Barvy~~ nevyhovují to dobře.

Finální dotazník

Při vyplňování dotazníku můžete libovolně procházet aplikací a máte na něj libovolný čas.

1. Jaký jste měl z testování pocit?

normální

2. Co se vám na této aplikaci líbilo?

foto

2.5 Proč? (odůvodněte prosím vaši odpověď)

připomíná mi to domov, když jsem
byl ve vys. lázeňích

3. Jaké největší problémy jste musel při plnění úkolů překonávat?

soustředil se

4. Co by jste na této aplikaci změnil?

nic

4.5 Proč? (odůvodněte prosím vaši odpověď)

Finální dotazník

Při vyplňování dotazníku můžete libovolně procházet aplikací a máte na něj libovolný čas.

1. Jaký jste měl z testování pocit?

VELICE DOBRÝ.

2. Co se vám na této aplikaci líbilo?

JEDNODUCHOST, JASOVITĚLNOST, RYCHLOST.

2.5 Proč? (odůvodněte prosím vaši odpověď)

APLIKACE VELICE JASNĚ A POMĚRNĚ RYCHLE
DOJDE KE POČTA DOVAŇM INFORMACÍM.

3. Jaké největší problémy jste musel při plnění úkolů překonávat?

ŽÁDNÉ.

4. Co by jste na této aplikaci změnil?

NIC. APLIKACE JE VADNÁ I PRO NEJLEPŠÍHO
ZAČÁTEČNÍKA. JEDNODUŠE NAVEŠTE KE CÍLI.

4.5 Proč? (odůvodněte prosím vaši odpověď)

POSTUPNĚ JAKY JSOU MATOČIK JASNĚ, JASOVITĚLNĚ
A JEDNODUŠĚ, VĚ JE NEZBE ODĚVAT S APLIKACÍ
PRACOVAT.

Příloha G

Vyplněné dotazníky finálního testování

Finální dotazník	
<i>Při vyplňování dotazníku můžete libovolně procházet aplikací a máte na něj libovolný čas.</i>	
1. Jaký jste měl z testování pocit?	BÍLO TO OK
2. Co se vám na této aplikaci líbilo?	RYCHLOST POUŽITÍ
2.5 Proč? (odůvodněte prosím vaši odpověď)	RAĎ ČEKÁM
3. Jaké největší problémy jste musel při plnění úkolů překonávat?	TLAČÍTKA PŘIDAT/ODEBRAT BOD LEGENDA
4. Co by jste na této aplikaci změnil?	TLAČÍTKA PŘIDAT/ODEBRAT BOD LEGENDU <input checked="" type="checkbox"/> PŘIDAT FAQ
4.5 Proč? (odůvodněte prosím vaši odpověď)	LEGENDA NEVÝRAZIVÁ TLAČÍTKA HŮŘE POCHOPITEČNÁ

Finální dotazník

Při vyplňování dotazníku můžete libovolně procházet aplikací a máte na něj libovolný čas.

1. Jaký jste měl z testování pocit?

čítal jsem se dobře začít s testem.

2. Co se vám na této aplikaci líbilo?

- jednoduchost, základního ovládání
- přehlednost

2.5 Proč? (odůvodněte prosím vaši odpověď)

* Nejsou zde zbytečné volby navíc, vše působí přehledně.

3. Jaké největší problémy jste musel při plnění úkolů překonávat?

- manipulace s nástrojem na vršení horizontu - přidávání a odebrání bodů

4. Co by jste na této aplikaci změnil?

1) rozlišení prvního a druhého kroku vršení horizontu
2) jméno uživatele by mělo nechat jako popis tlačítka

4.5 Proč? (odůvodněte prosím vaši odpověď)

1) Na první pohled nebylo patrné, že potřeba v prvním kroku
vršit horizont detailně.
2) Je to lehce matoucí, pokud chce zjistit např. výsledky
vzhledem k tomu.

Finální dotazník

Při vyplňování dotazníku můžete libovolně procházet aplikací a máte na něj libovolný čas.

1. Jaký jste měl z testování pocit?

celkem dobrý, příjemné prostředí, informální

2. Co se vám na této aplikaci líbilo?

krásná a vánoční stránka se spoustou o směřování širokým
okružím, kde by se tato oblast aplikace dala využít

2.5 Proč? (odůvodněte prosím vaši odpověď)

podobnost známých možností umocnění ~~pro~~ plánování tras
na turistice / úpravi fotografií

3. Jaké největší problémy jste musel při plnění úkolů překonávat?

neviděla jsem co přesně stránka ubírá (např. co
znamenají mapy / ukazují mapy při sledování trasy
a kde některé věci hledat - turistická sledování)

4. Co by jste na této aplikaci změnil?

dále bych jasněji upřesnila stránku vyhledávání -
např. při úpravě fotografií, mapy

4.5 Proč? (odůvodněte prosím vaši odpověď)

upřesnění stránky není mi úplně jasné takže jakékoliv
bych prakticky stránky by byla, případně spousta kvůli
činnosti bych na stránce měla a vyhledávání služby

Finální dotazník

Při vyplňování dotazníku můžete libovolně procházet aplikací a máte na něj libovolný čas.

1. Jaký jste měl z testování pocit?

... až naten Linux to bylo fajn.

2. Co se vám na této aplikaci líbilo?

GRAFKA
KOLEČKO NAČÍTÁNÍ
NÁPAD

2.5 Proč? (odůvodněte prosím vaši odpověď)

3. Jaké největší problémy jste musel při plnění úkolů překonávat?

EXCEPTION - ŠPATNÉ HESLO
STATUS PŘI ZPRACOVÁNÍ - RSÁT DETAILY

4. Co by jste na této aplikaci změnil?

1) PŘIDAT LANG- PACK
2) PŘIDAT KOMENTÁŘ K VÝSLEDKŮM
3) ZVÝRAZNIT ~~VAŠE~~ MENU V HISTORII VYHLEDÁVÁNÍ

4.5 Proč? (odůvodněte prosím vaši odpověď)

1) JSO!!!
2) VÝSLEDKY JSOU SLOŽITĚJŠÍ NA DOCHOPENÍ LAIKEM
3) NEKŮBIL JSEM SI, ŽE JE TO KLIKATELNÉ

Finální dotazník

Při vyplňování dotazníku můžete libovolně procházet aplikací a máte na něj libovolný čas.

1. Jaký jste měl z testování pocit?

DODATĚ, PŘÍJEMNĚ.

2. Co se vám na této aplikaci líbilo?

JE DOKONČENOST, Z VĚTŠÍ ČÁSTI PŘEHLEDNOST.

2.5 Proč? (odůvodněte prosím vaši odpověď)

AŽ NA VĚŠTÍKY JE DĚL SNAŽNO ORIENTOVAT. APLIKACE UKÁZALA, ŽE DE SCHOVAT I POMĚRNĚ PŘESNĚ UČIT HORIZONT.

3. Jaké největší problémy jste musel při plnění úkolů překonávat?

PŘI PRVNÍM POKUŠOVÁNÍ NEMÍ ZPRAVU ZAJASNĚ, ZDA APLIKACE PRACUJE. PO PŘIHLÁŠENÍ NEMÍ MIKRO UVEDENÍ, ŽE POP OMĚNĚNÍ SE JEDNÁ O HISTORIE.

4. Co by jste na této aplikaci změnil?

V ZÁKLADĚ MI PODSTATNĚHO, DEN VĚCI UVEDENÉ V DOPĚD.

4.5 Proč? (odůvodněte prosím vaši odpověď)

DE DODATĚ VĚDĚT, ŽE APLIKACE PRACUJE A CHŮCI SI TO TAVAT. NEMÍ ZAJMNĚ, KDE SE NACHÁZÍ HISTORIE, TAKŽE SI UŽIVATEL ODJEU DEN NÁHODU.

Finální dotazník

Při vyplňování dotazníku můžete libovolně procházet aplikací a máte na něj libovolný čas.

1. Jaký jste měl z testování pocit?

PŘEHLEDNĚ, VEDE UŽIVATELE, POCIT RELATIVNĚ
DOBRY

2. Co se vám na této aplikaci líbilo?

PŘEHLEDNOST

2.5 Proč? (odůvodněte prosím vaši odpověď)

VEDE UŽIVATELE CO MA' DELAT

3. Jaké největší problémy jste musel při plnění úkolů překonávat?

NEČETLA JSEM POPISY A TÍM JSEM SI
KOMPLIKOVALA PRÁCI

4. Co by jste na této aplikaci změnil?

ZRYCHLENÍ

4.5 Proč? (odůvodněte prosím vaši odpověď)

POMALU MAČKA'

Finální dotazník

Při vyplňování dotazníku můžete libovolně procházet aplikací a máte na něj libovolný čas.

1. Jaký jste měl z testování pocit?

lehká náročnost

2. Co se vám na této aplikaci líbilo?

2.5 Proč? (odůvodněte prosím vaši odpověď)

3. Jaké největší problémy jste musel při plnění úkolů překonávat?

lehká klíčnost

4. Co by jste na této aplikaci změnil?

větší obrázky, více bodů v nabídce, více
bodů s body by měla být nabídka ta hodí na první
obrázku

4.5 Proč? (odůvodněte prosím vaši odpověď)

Finální dotazník

Při vyplňování dotazníku můžete libovolně procházet aplikací a máte na něj libovolný čas.

1. Jaký jste měl z testování pocit?

ZVĚDAVOST

2. Co se vám na této aplikaci líbilo?

FOTOGRAFIE

2.5 Proč? (odůvodněte prosím vaši odpověď)

KRÁSNE LÁZEVY

3. Jaké největší problémy jste musel při plnění úkolů překonávat?

TECHNICKÉ PROBLÉMY

4. Co by jste na této aplikaci změnil?

STOULIVOST, RYCHLOST

4.5 Proč? (odůvodněte prosím vaši odpověď)

SYSTÉM ZKOLAHOVAL

Finální dotazník

Při vyplňování dotazníku můžete libovolně procházet aplikací a máte na něj libovolný čas.

1. Jaký jste měl z testování pocit?

Raz-moveri aplikace, myslím, se lidi by
mohla zaměřovat - s rozvojem obsahu

2. Co se vám na této aplikaci líbilo?

Wicem - místa na folce - členské má
brábra po čase není, dole to bylo.

2.5 Proč? (odůvodněte prosím vaši odpověď)

3. Jaké největší problémy jste musel při plnění úkolů překonávat?

nešel napsat Ruvimac @

4. Co by jste na této aplikaci změnil?

rychlost vyhledávání

4.5 Proč? (odůvodněte prosím vaši odpověď)

Finální dotazník

Při vyplňování dotazníku můžete libovolně procházet aplikací a máte na něj libovolný čas.

1. Jaký jste měl z testování pocit?

zapojenost, nový pohled na barikadu

2. Co se vám na této aplikaci líbilo?

jednoduchost ovládnutí

2.5 Proč? (odůvodněte prosím vaši odpověď)

3. Jaké největší problémy jste musel při plnění úkolů překonávat?

žádné

4. Co by jste na této aplikaci změnil?

nic

4.5 Proč? (odůvodněte prosím vaši odpověď)