



## ZADÁNÍ DIPLOMOVÉ PRÁCE

**Název:** Webová aplikace pro prezentace panoramatických snímků  
**Student:** Bc. David Krupička  
**Vedoucí:** Ing. Petr Pulc  
**Studijní program:** Informatika  
**Studijní obor:** Webové a softwarové inženýrství  
**Katedra:** Katedra softwarového inženýrství  
**Platnost zadání:** Do konce zimního semestru 2017/18

### Pokyny pro vypracování

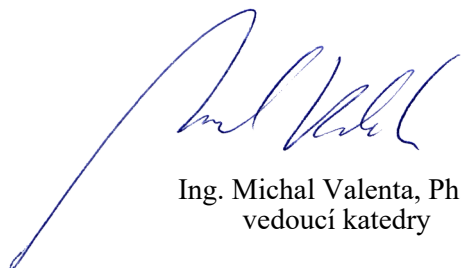
Vytvořte webovou aplikaci pro vytváření, správu a přehrávání prezentací fotografií s důrazem na podporu promítání panoramatických snímků a snímků ve velmi vysokém rozlišení. Aplikace musí podporovat zobrazení pro promítacího, kdy bude fungovat ve dvou (nebo více) instancích prohlížeče zároveň a procházení obsahu bude vhodnými prostředky synchronizováno.

Konkrétně:

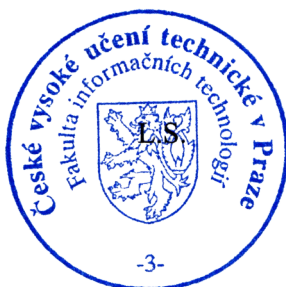
- 1) Analyzujte potřeby prezentujícího, zohledněte například časová omezení prezentace.
- 2) Prozkoumejte možnosti synchronizace prohlíženého obsahu.
- 3) Navrhněte a implementujte uživatelské rozhraní pro tvorbu prezentací, včetně časovaného procházení panoramatických snímků.
- 4) Navrhněte a implementujte rozhraní prezentujícího, včetně ruční navigace (např. ve snímcích s velmi vysokým rozlišením).
- 5) Implementujte synchronizaci prezentace na jednom i více zařízeních, zohledněte případně využití v prostředí SAGE.
- 6) Proveďte testování a porovnání s aktuálně používanými způsoby prezentace.

### Seznam odborné literatury

Dodá vedoucí práce.



Ing. Michal Valenta, Ph.D.  
vedoucí katedry



prof. Ing. Pavel Tvrdík, CSc.  
děkan

V Praze dne 30. září 2016



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Diplomová práce

# **Webová aplikace pro prezentace panoramatických snímků**

***Bc. David Krupička***

Vedoucí práce: Ing. Petr Pulc

5. května 2017





---

## Poděkování

V první řadě bych rád poděkoval vedoucímu práce Ing. Petru Pulcovi za konzultace, připomínky a rady, které mi dal. Dále také Ing. Jiřímu Melnikovovi za pomoc při testování aplikace v SAGElab. A v neposlední řadě také děkuji autorům všech knihoven, které jsem v aplikaci použil.



---

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne 5. května 2017

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2017 David Krupička. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.*

## **Odkaz na tuto práci**

Krupička, David. *Webová aplikace pro prezentace panoramatických snímků*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

---

## Abstract

This master's thesis deals with creating a web application for user friendly creation and management of photo presentations. Emphasis is given to support of panoramic images. The thesis analyses possibility of synchronization of multiple browser windows, which is used in the application for a separate view for the presenter.

**Keywords** photography, presentations, synchronizations, SAGE

---

## Abstrakt

Tato diplomová práce se zabývá tvorbou webové aplikace pro uživatelsky přívětivé vytváření a správu prezentací fotografií. Důraz je kladen na podporu promítání panoramatických snímků. Práce obsahuje analýzu možností synchronizace více oken prohlížečů, která je v aplikaci využita k oddělenému zobrazení pro přednášejícího.

**Klíčová slova** fotografie, prezentace, synchronizace, SAGE



---

# Obsah

<b>Úvod</b>	<b>1</b>
<b>1 Analýza</b>	<b>3</b>
1.1 Požadavky přednášejících na aplikaci . . . . .	3
1.1.1 Funkční požadavky . . . . .	3
1.1.2 Nefunkční požadavky . . . . .	4
1.2 Synchronizace více oken prohlížeče . . . . .	5
1.2.1 Zásuvné moduly . . . . .	5
1.2.2 Opakované dotazování . . . . .	6
1.2.3 Dlouhé dotazy . . . . .	7
1.2.4 Metoda Push . . . . .	8
<b>2 Návrh</b>	<b>11</b>
2.1 Serverová strana aplikace . . . . .	11
2.2 Databáze . . . . .	11
2.3 Zobrazení aplikace . . . . .	13
2.4 WebSocket server . . . . .	15
2.5 Certifikát . . . . .	16
<b>3 Realizace</b>	<b>17</b>
3.1 HTML5 . . . . .	17
3.1.1 Typ dokumentu . . . . .	17
3.1.2 Odkaz . . . . .	18
3.1.3 Struktura dokumentu . . . . .	18
3.1.4 Formuláře . . . . .	18
3.1.5 Vlastní data k elementům . . . . .	19
3.1.6 Upravitelný obsah . . . . .	19

3.2	Kaskádové styly . . . . .	20
3.2.1	Zobrazení na mobilních zařízeních . . . . .	20
3.3	PHP . . . . .	21
3.3.1	Registrace, přihlášení, relace . . . . .	21
3.3.2	Nahrávání a zpracování fotografií . . . . .	22
3.3.3	Asynchronní požadavky . . . . .	23
3.4	Databáze . . . . .	23
3.5	JavaScript na straně klienta . . . . .	23
3.5.1	Knihovny . . . . .	24
3.5.2	Funkce . . . . .	27
3.5.3	WebSocket . . . . .	33
3.6	WebSocket server . . . . .	33
3.6.1	Formát WebSocket zpráv . . . . .	34
3.7	Manifest webové aplikace . . . . .	36
<b>4</b>	<b>Návod k instalaci</b>	<b>37</b>
4.1	Potřebné vybavení . . . . .	37
4.2	Zkopírování souborů . . . . .	38
4.3	Databáze . . . . .	38
4.4	WebSocket . . . . .	38
<b>5</b>	<b>Testování</b>	<b>41</b>
5.1	Funkčnost . . . . .	41
5.2	Validita kódu . . . . .	44
5.3	Bezpečnost . . . . .	44
5.3.1	SQL injection . . . . .	44
5.3.2	Cross-site scripting . . . . .	45
5.4	Nielsenova heuristická analýza . . . . .	45
5.5	SAGE . . . . .	48
5.6	Porovnání s jinými nástroji . . . . .	50
5.6.1	Nativní prohlížeče fotografií . . . . .	50
5.6.2	Video prezentace . . . . .	50
5.6.3	Microsoft PowerPoint . . . . .	50
5.6.4	Scrollshow . . . . .	51
5.6.5	Prezi . . . . .	51
<b>6</b>	<b>Další vývoj</b>	<b>53</b>
6.1	Video . . . . .	53
6.2	Mapa . . . . .	53
6.3	Stream přednášek . . . . .	54
6.4	Efekty . . . . .	54



6.5	Přihlášení . . . . .	55
6.6	Aplikace . . . . .	55
<b>Závěr</b>		<b>57</b>
<b>Literatura</b>		<b>59</b>
<b>A</b>	<b>Seznam použitých zkratk</b>	<b>63</b>
<b>B</b>	<b>Obsah přiloženého DVD</b>	<b>67</b>



---

## Seznam obrázků

1.1	Schéma komunikace při opakovaném dotazování . . . . .	7
1.2	Schéma komunikace při použití metody dlouhých dotazů . . . . .	8
1.3	Schéma komunikace při použití metody Push . . . . .	9
2.1	EER model databáze . . . . .	12
2.2	Návrh uživatelského rozhraní seznamu prezentací . . . . .	12
2.3	Návrh uživatelského rozhraní pro úpravy prezentace . . . . .	13
2.4	Návrh uživatelského rozhraní zobrazení pro přednášejícího . . . . .	15
3.1	Zobrazení stavu nahrávání fotografií . . . . .	31
3.2	Příklad zobrazení pro přednášejícího . . . . .	32
4.1	Rozhraní pro správu služeb s WebSocket serverem . . . . .	40
5.1	Zobrazení aplikace na mobilním zařízení . . . . .	43
5.2	Ukázka ikonek použitých v aplikaci . . . . .	46
5.3	Testování v SAGElab na 4K displeji . . . . .	48
5.4	Testování v SAGElab na 8K+ stěně . . . . .	49



---

# Úvod

Většinu svého volného času trávím na cestách. Při svých cestách rád fotografuji a na zajímavých místech také často pořizuji panoramatické fotografie. Několikrát do roka rovněž pořádám různá cestovatelská promítání z mých cest, abych se podělil o krásy vzdálenějších míst a inspiroval další cestovatele k tomu, aby se vydali na cestu. Během vytváření svých prezentací jsem již vyzkoušel několik různých programů pro jejich tvorbu, žádný ale nevyhovoval mým požadavkům. Rozhodl jsem se tedy v rámci diplomové práce vytvořit systém, který by pracoval tak, jak potřebuji, a vytváření prezentací by v něm bylo jednoduché, rychlé, přehledné a bez dalších zbytečných nastavování, která v tomto případě nejsou potřeba. A jelikož studuji zaměření webové inženýrství, tak bylo nasnadě, že se bude jednat o webovou aplikaci.

Panoramatická fotografie je fotografický žánr, který zachycuje velký úhel pohledu, nejčastěji krajiny nebo jakéhokoliv jiného celku. Panoramatický formát je divácky vděčný, je bližší způsobu, kterým lidé vnímají svět kolem sebe, a proto patří k atraktivním oblastem fotografie [1]. Dříve se pro zachycení panoramatických snímků používaly speciální panoramatické kamery, dnes však již mnoho zařízení umí samo vytvářet panoramatické snímky postupným natočením přístroje. Nejinak je tomu u mého fotoaparátu, nicméně výsledná fotografie má poměrně nízké rozlišení. Proto zatím stále panoramatické snímky pořizuji v manuálním režimu, tedy tím způsobem, že pořídím několik fotografií, které se částečně překrývají a v počítači je teprve spojím v jednu panoramatickou fotografii ve vyšším rozlišení než za použití automatického režimu ve fotoaparátu. Jednotlivé fotografie většinou pořizuji na výšku, abych měl co nejširší záběr odshora dolů. Vzhledem k použití širokoúhlého objektivu tak většinou stačí pořídít jednu řadu fotografií.

Pro spojování fotografií jsem dříve používal Adobe Photoshop, což je však poměrně drahý nástroj. V dnešní době existuje ale také mnoho šikov-

ných bezplatných nástrojů. Mezi v současné době nejpropracovanější dle mého názoru patří Image Composite Editor od společnosti Microsoft [2], který je vyvíjen od roku 2008. V aktuální verzi obsahuje také funkci pro automatické doplnění obsahu fotografie tak, aby se po spojení nemusela ořezávat příliš velká plocha fotografie. Tato funkce dle mého názoru funguje lépe než obdobná funkce ve Photoshopu CS6, což je nejnovější verze Photoshopu, se kterou jsem zatím pracoval. V novějších verzích Photoshopu, označovaných jako CC bude již tato funkce pravděpodobně fungovat také lépe.

Při mých prvních prezentacích jsem ale řešil problém, jak divákům panoramatické snímky zobrazit. Doslova mě irituje, když přednášející promítá panoramatické snímky tím stylem, že promítne pouze úzký proužek fotografie uprostřed promítací plochy. Například taková průměrná panoramatická fotografie, která má původní rozlišení 20 000 x 3 500 pixelů, se při promítání takovýmto způsobem na projektoru s nativním rozlišením 800 x 600 pixelů promítne s vertikálním rozlišením pouhých 112 pixelů, veškeré detaily se tak z fotografie vytratí. Naopak promítat fotografii přímo v původním rozlišení, by také na tak malé zobrazovací ploše nebyl nejšťastnější nápad. Diváci by v jeden moment viděli pouze velmi malý výřez fotografie, muselo by se pak hodně posouvat a divák by z toho byl akorát zmatený. Moji nejoblíbenější volbou, jak takovéto fotografie promítat, je promítnout je v takovém rozlišení, kdy jejich vertikální rozlišení je shodné s vertikálním rozlišením plochy, na kterou se promítá. Tedy fotografie z příkladu se promítne v rozlišení 4 286 x 600 pixelů. Divákovi se tak zobrazí téměř 29krát více obrazové informace v porovnání s promítáním fotografie zmenšené dle horizontálního rozlišení.

Fotografii je ale nutné v horizontální ose pomalu posouvat, aby ji diváci viděli celou. To lze zajistit buď posouváním fotografie myší nebo automatickým posunem fotografie. Vzhledem k tomu, že při prezentacích se většinou soustředím hlavně na to, o čem mluvím, tak já osobně preferuji automatický posun. Doposud jsem si prezentace připravoval v programu Microsoft PowerPoint, kde je možné automatický posun nastavit v panelu animací přes cesty pohybu. Tato funkce však není uzpůsobená přímo pro posouvání obrázků ve vysokém rozlišení, proto posun často není úplně plynulý a takovéto nastavení pro každý panoramatický snímek zvlášť je velmi zdlouhavé. Musí se nakreslit cesta posunu, nastavit časování, začátek a konec posunu a určit, zda se má posunout i zpět. Tak mě napadlo, proč nevytvořit aplikaci, která by toto vše zařídila sama a přednášející by se pak při přípravě prezentace mohl věnovat důležitějším věcem, jakožto třeba tříděním fotografií či přípravě mluveného projevu.

## Analýza

Prvním krokem bylo stanovit si požadavky na vytvářenou aplikaci a vybrat vhodné prostředky k její implementaci.

### 1.1 Požadavky přednášejících na aplikaci

Většina požadavků byla shromážděna z osobních poznatků, protože sám jsem již mnohokrát prezentace připravoval. Zároveň se ale také v průměru jedenkrát týdně účastním cestovatelských promítání jiných přednášejících, se kterými si vzájemně vyměňujeme zkušenosti a nápady, které také přispěly k níže uvedeným požadavkům. Požadavky jsou ovlivněny také tím, že se bude jednat o webovou aplikaci hostovanou na webovém serveru. Můžeme je rozdělit na funkční a nefunkční.

#### 1.1.1 Funkční požadavky

##### 1.1.1.1 Autentizace, autorizace

Pro vstup do aplikace je vyžadována jednoduchá registrace a přihlašování uživatelů pomocí uživatelského jména a hesla. Heslo si může uživatel změnit pomocí formuláře v aplikaci.

##### 1.1.1.2 Správa prezentací

**Základní úkony** Uživatel může vytvořit novou prezentaci, změnit její název a odstranit prezentaci.

## 1. ANALÝZA

---

**Úkony s fotografiemi** Uživatel může do prezentace nahrát své fotografie, může měnit jejich pořadí v prezentaci a může libovolnou fotografii z prezentace odstranit. Ke každé fotografii může vložit krátký text, který se zobrazí všem divákům, a poznámku, která se bude zobrazovat pouze jemu v zobrazení pro přednášejícího. Poznámka slouží k uložení myšlenek, aby přednášející během své přednášky neztratil nit nebo nezapomněl na některé důležité informace, které chce divákům sdělit.

### 1.1.1.3 Promítání

Prezentaci bude možné zobrazit na více oknech prohlížeče zároveň, a to i na jiných zařízeních. Z jednoho okna prohlížeče bude ovládaná a na dalších, kterých může být i více než jedno, bude zobrazena divákům. Mezi jednotlivými snímky se bude možno pohybovat dopředu, ale i dozadu. Prezentaci také bude možné zahájit z libovolného snímku prezentace.

**Zobrazení pro přednášejícího** Krom klasického zobrazení na celou plochu bude u toho zobrazení, které prezentaci ovládá, možnost zobrazení v režimu pro přednášejícího, které vedle náhledu aktuálního snímku bude obsahovat i další informace. Mezi hlavní patří již zmíněná poznámka přednášejícího, aktuální čas, graficky znázorněný stav prezentace se stopkami, aby přednášející na první pohled získal přehled, jak je na tom s časem, a malý náhled následujícího snímku.

### 1.1.2 Nefunkční požadavky

#### 1.1.2.1 Bezpečnost

Aplikace musí ošetřovat vstupy od uživatele pro zabránění SQL injection a XSS útokům. Do databáze se bude ukládat pouze otisk hesla, tak aby z databáze toto heslo nebylo možno zjistit. Bude používat šifrovaný HTTPS protokol a musí zajistit integritu přenášených synchronizačních zpráv.

#### 1.1.2.2 Použití v SAGE

Aplikace by měla umožnit použití v prostředí SAGE [3]. Musí tak podporovat zobrazení fotografií s rozlišením až 4320 řádků.

#### 1.1.2.3 Responzivní rozhraní

Responzivní rozhraní je důležitá vlastnost pro tuto aplikaci, aby bylo možno prezentaci ovládat z tabletu nebo mobilu. Vhodné je také doplnit možnost



ovládání prezentace gesty na takovýchto zařízeních. Například posun prstem po obrazovce doleva pro přechod na následující snímek a analogicky přejetí prstem doprava pro předchozí.

### 1.1.2.4 Plynulost zobrazení

Plynulost zobrazení by měla být co nejvyšší při jakémkoliv smysluplném rozlišení. Z tohoto důvodu je fotografie nutné optimalizovat na dané rozlišení.

### 1.1.2.5 Uživatelská přívětivost

Přednášející jsou málokdy počítačovými experty, proto by aplikace měla být co nejjednodušší a velmi intuitivní na ovládání.

## 1.2 Synchronizace více oken prohlížeče

HTTP(S) je protokol na aplikační vrstvě používaný pro komunikaci mezi klientem (webovým prohlížečem) a serverem. Protokol HTTP(S) využívá protokolu TCP z transportní vrstvy sady protokolů TCP/IP, který garantuje spolehlivé doručování zpráv ve správném pořadí. Komunikace je založená na principu dotazu a jedné odpovědi [4]. Server nemůže sám poslat odpověď klientovi, aniž by od něj před tím dostal dotaz a nemůže sám iniciovat spojení ke klientovi. Níže se podíváme, jaké tedy máme možnosti k synchronizaci více oken prohlížeče.

### 1.2.1 Zásuvné moduly

S rozvojem internetu a webových aplikací sílila potřeba odezvy v reálném čase, a tak se hledaly různé metody, jak zajistit komunikaci klienta se serverem i po načtení webové stránky. Zprvu se začaly využívat různé zásuvné moduly, které mohly přijímat zprávy od serveru.

#### 1.2.1.1 Java aplety

Historie v Javě naprogramovaných apletů sahá až do roku 1995 [5]. Největší rozmach zaznamenaly v prvním desetiletí nového tisíciletí. Pro správnou funkčnost si uživatel musel nainstalovat prostředí JRE. Po načtení kódu byl aplet spuštěn z webové stránky v JVM jako samostatný proces oddělený od prohlížeče. Hlavní výhoda apletů byla, že mohly přímo přistupovat k 3D hardwarové akceleraci, díky čemuž byly velmi rychlé. S vývojem

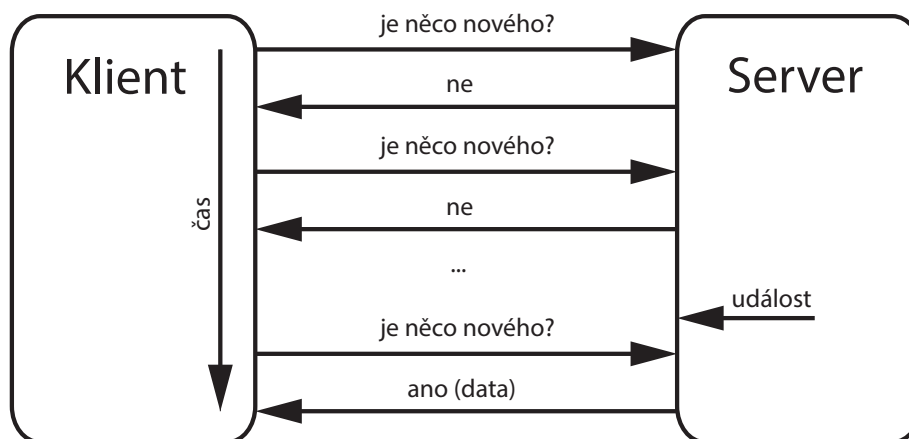
dalších technologií se však tato výhoda pomalu stírala. Aplety využívaly k asynchronnímu přenosu dat různých funkcí, z nichž však většina v jádru používala metody, které jsou popsány dále v této kapitole. Aplety již nejsou z důvodu bezpečnostních rizik v nových prohlížečích podporovány, prohlížeč Chrome je přestal podporovat od září 2015 [6]. Společnost Oracle také oznámila, že JDK nebude od verze 9 obsahovat zásuvný modul pro prohlížeče [7].

### 1.2.1.2 Flash

Flash se vyvinul z produktu zvaného SmartSketch, který sloužil pro kreslení vektorových obrázků. Později byla již jako FutureSplash Animator aplikaci přidána možnost animace po jednotlivých oknech. Po převzetí společnosti FutureWave společností Macromedia se z FutureSplash Animator stal Macromedia Flash 1.0. Macromedia současně s Flash technologií vyvinula RTMP protokol, který sloužil nejen pro streaming audia a videa, ale také dat. Díky tomu byl Flash velkým pokrokem v oblasti interaktivity a vektorové grafiky na webu. Největšího vrcholu dosáhl kolem roku 2005, kdy Macromedii převzalo Adobe [8]. Flash má vlastní propracovaný objektově orientovaný programovací jazyk ActionScript. Flash využívají i jiné metody pro asynchronní komunikaci, např. Flash XMLSocket relays používá Flash o velikosti jednoho pixelu, který obsahuje objekt XMLSocket, k němuž je ze stránky přistupováno pomocí JavaScriptu. Stejně jako aplety je Flash technologie pro bezpečnostní rizika na ústupu. Synchronizací za použití zásuvných modulů se tedy již dále nebudeme zabývat.

### 1.2.2 Opakované dotazování

Opakované dotazování je metoda, kdy se načtená stránka dotazuje na server, typicky XHR požadavkem, zda pro ni není k dispozici nový obsah. Už z principu této metody je jasné, že nebude fungovat úplně v reálném čase, bude záležet na nastavení času mezi jednotlivými požadavky. Čím kratší doba mezi požadavky, tím více se budeme blížit ke komunikaci v reálném čase. Na druhou stranu velmi poroste zátěž sítě a serveru. Ve vyvíjené aplikaci požadujeme nízkou latenci, tato metoda tedy pro náš účel není vhodná. Svoje uplatnění může nalézt například v některých chatovacích místnostech, kde nové příspěvky přibývají rychle a nebude nám příliš vadit drobné zpoždění zobrazení zprávy. Na obrázku 1.1 je znázorněno schéma komunikace při opakovaném dotazování.

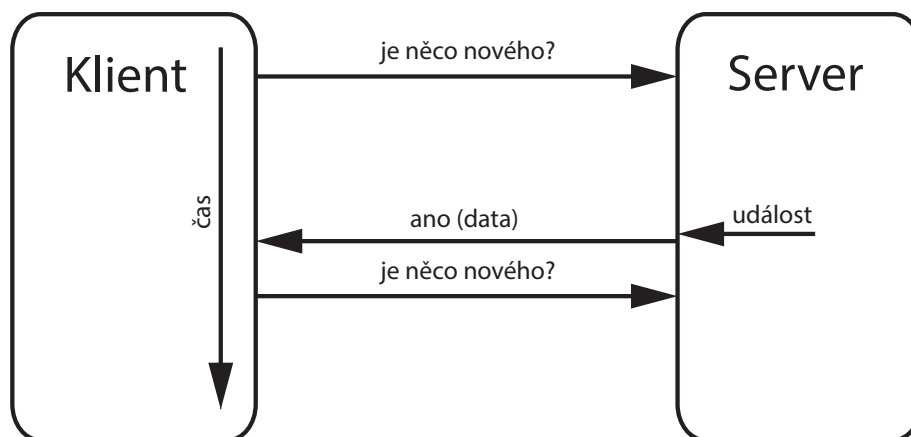


Obrázek 1.1: Schéma komunikace při opakovaném dotazování

### 1.2.3 Dlouhé dotazy

Dlouhé dotazy jsou mezistupněm mezi opakovaným dotazováním a metodou Push. Nabízí vylepšení techniky opakovaného dotazování tak, že se chová obdobně jako metoda Push. Klient pošle serveru požadavek, ten ale neodpoví hned. Drží spojení otevřené, dokud nemá dostupný obsah pro klienta a v ten moment jej odešle, čímž dojde k uzavření spojení. Nicméně klient ihned pošle požadavek nový, aby byl připraven na následující zprávy. S tím je ale spojené riziko, že ne všechny zprávy se musí ke klientovi dostat. Pokud na serveru nastane událost krátce po uzavření spojení, tedy dříve než je obdržén nový požadavek od klienta, tak se druhá událost nemusí ke klientovi dostat. Pro zachování integrity by tedy bylo nutné držet zprávu v mezipaměti na serveru do doby přijetí následujícího požadavku, nebo místo aktualizacních zpráv vždy předávat celý stav aplikace. Schéma komunikace je znázorněno na obrázku 1.2.

Tato metoda byla velmi populární s nástupem RIA aplikací. Svůj největší rozmach zaznamenala kolem roku 2010. Nevýhodou je, že nemůže běžet přímo na stejném webovém serveru jako aplikace a pro její funkčnost je potřeba COMET server. Jednou z nejznámějších implementací je APE Project [9].



Obrázek 1.2: Schéma komunikace při použití metody dlouhých dotazů

### 1.2.4 Metoda Push

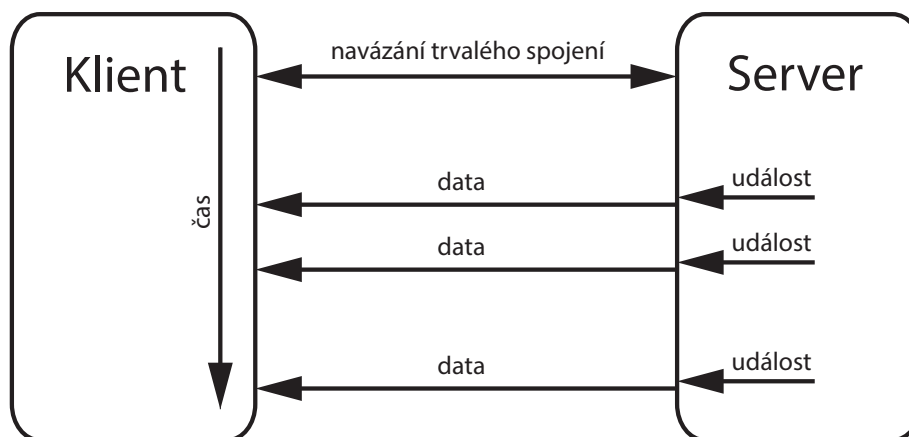
Metoda Push naopak ke komunikaci používá trvalého spojení. Jedná se o nejnovější metody synchronizace využívající nové možnosti jazyka HTML z verze HTML5. Schéma komunikace je znázorněno na obrázku 1.3. Základní metody, které je možno použít jsou Server-sent events a WebSocket. Nyní se na obě podíváme detailněji.

#### 1.2.4.1 Server-sent events

Server-sent events slouží pouze k přenosu dat ze serveru ke klientovi, neslouží tedy k obousměrné komunikaci. Vhodné využití nalezne v aplikacích jako je Twitter, kde dochází k aktualizaci časové osy [10]. Komunikace od klienta k serveru je možná některou z klasických metod jako např. pomocí XHR požadavku. Výhodou Server-sent events je, že běží přes klasický HTTP protokol a nepotřebuje tedy další server. Nevýhodou je zatím nízká podpora ze strany prohlížečů, což však lze nahradit polyfillem<sup>1</sup> napsaným v JavaScriptu.

---

<sup>1</sup>Polyfill je skript, který do prohlížečů nepodporujících konkrétní vlastnost dodá potřebnou funkcionalitu.



Obrázek 1.3: Schéma komunikace při použití metody Push

#### 1.2.4.2 WebSocket

WebSocket je protokol, který podporuje obousměrnou komunikaci z klienta na server i ze serveru na klienta pomocí TCP protokolu. Má větší podporu ze strany prohlížečů než Server-sent events. Oproti předchozím metodám také značně šetří přenášená data, ve srovnání s metodou opakovaného dotazování přibližně 800krát [11]. V současné době se jedná o nejpoužívanější nástroj pro synchronizaci. Nevýhodou je opět potřeba samostatného serveru, který bude WebSocket spojení obsluhovat. Je součástí i větších softwarových řešení jako např. SignalR [12] pro platformu .NET. WebSocket také definuje vlastní schéma URI **ws** pro nešifrovanou komunikaci a **wss** pro šifrované spojení přes SSL/TLS.

Z výše uvedeného vyplývá, že pro náš případ bude ideálním řešením použití WebSocket. Naše požadavky na synchronizaci, tedy nízkou latenci a obousměrnou komunikaci, splňuje ze všech metod nejlépe. Navíc pro programátory nabízí jednoduché implementační rozhraní na straně klienta, kde je WebSocket dostupný v JavaScriptu jako třída `WebSocket`.

```
var connection = new WebSocket('wss://server');
```

Vytvořením instance této třídy výše uvedeným příkazem se naváže spojení se serverem a je možné serveru zasílat zprávy. Zprávy mohou obsahovat

## 1. ANALÝZA

---

libovolné textové řetězce, přesto je pro komunikaci vhodné použít např. formát JSON.

Objekt `WebSocket` nabízí 2 metody. První `send()` pro odeslání zprávy a druhou `close()` pro uzavření spojení. Dále obsahuje 4 atributy typu `EventListener`. První událost `onopen` je volána při otevření spojení. Analogicky událost `onclose` oznamuje uzavření spojení. Pro vlastní výměnu dat slouží událost `onmessage`. Jak už název napovídá, je zavolána ve chvíli, kdy klient obdrží od serveru zprávu. Poslední událost `onerror` je volána v případě, že nastane nějaká chyba [13].

---

## Návrh

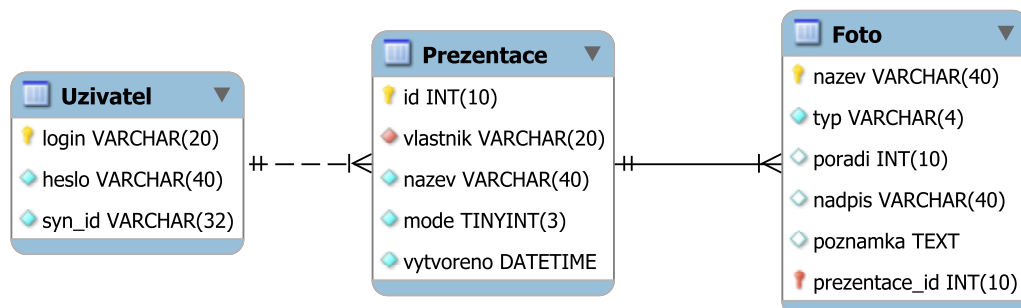
### 2.1 Serverová strana aplikace

Posun fotografií, ať už automatický či ruční, jsem vytvořil jako knihovnu v rámci semestrální práce z předmětu MI-MAI. Knihovna je napsaná v JavaScriptu za použití jQuery a využívá metody `animate()` pro automatický posun a `draggable()` pro manuální. Dále bylo třeba rozmyslet v jakém jazyce bude základ aplikace. Volba padla na PHP kvůli široké podpoře na webových serverech a dobré znalosti jazyka z mé strany. Zvažoval jsem použití různých frameworků, ale z toho důvodu, že tato vrstva na serveru bude zajišťovat jen základní funkcionalitu a hlavní těžiště bude v JavaScriptu na straně klienta, jsem se rozhodl pro čisté PHP. Jako základ jsem použil svoji semestrální práci z předmětu X36WWW, kde jsem již měl připravenou registraci a přihlašování uživatelů.

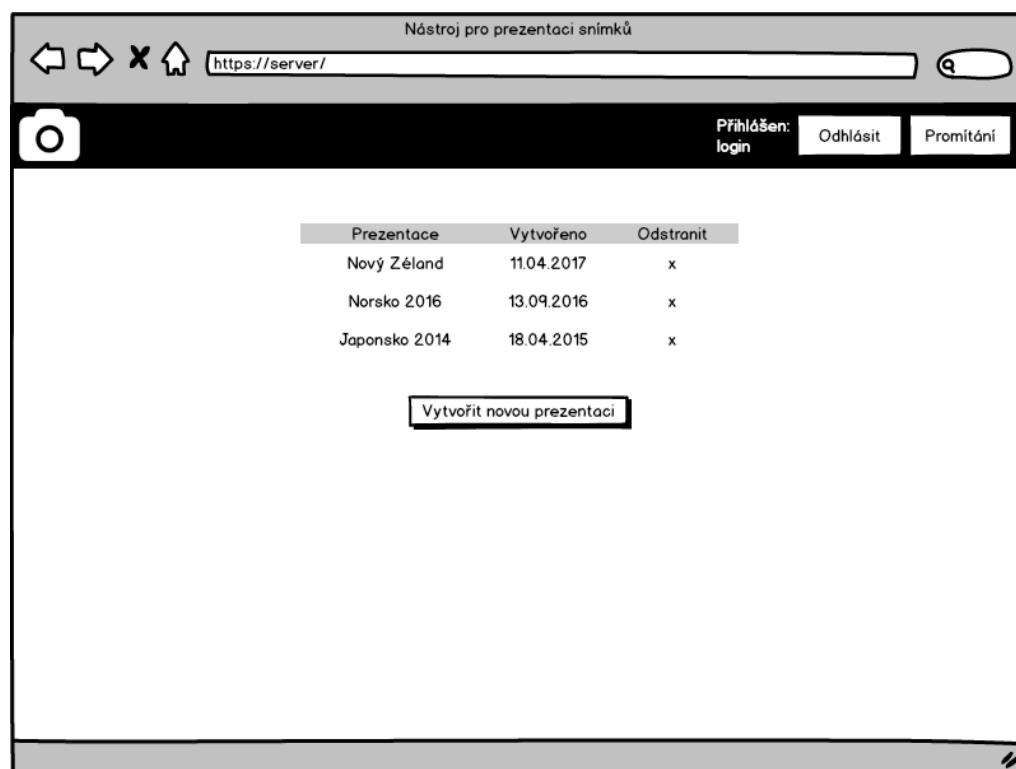
### 2.2 Databáze

Pro uložení klíčových informací aplikace postačí tři datové tabulky. První tabulkou je „Uživatel“, kde bude uloženo uživatelské jméno, otisk hesla a synchronizační id, které bude sloužit k identifikaci při WebSocket spojení. Dále tabulka „Prezentace“, kde bude id prezentace, vlastník, název, mód a datum vytvoření. Poslední tabulkou bude tabulka „Foto“ pro fotografie, ta bude obsahovat název fotografie, formát souboru, pořadí, nadpis, poznámku a cizí klíč id prezentace, do které patří. EER model databáze je znázorněn na obrázku 2.1.

## 2. NÁVRH

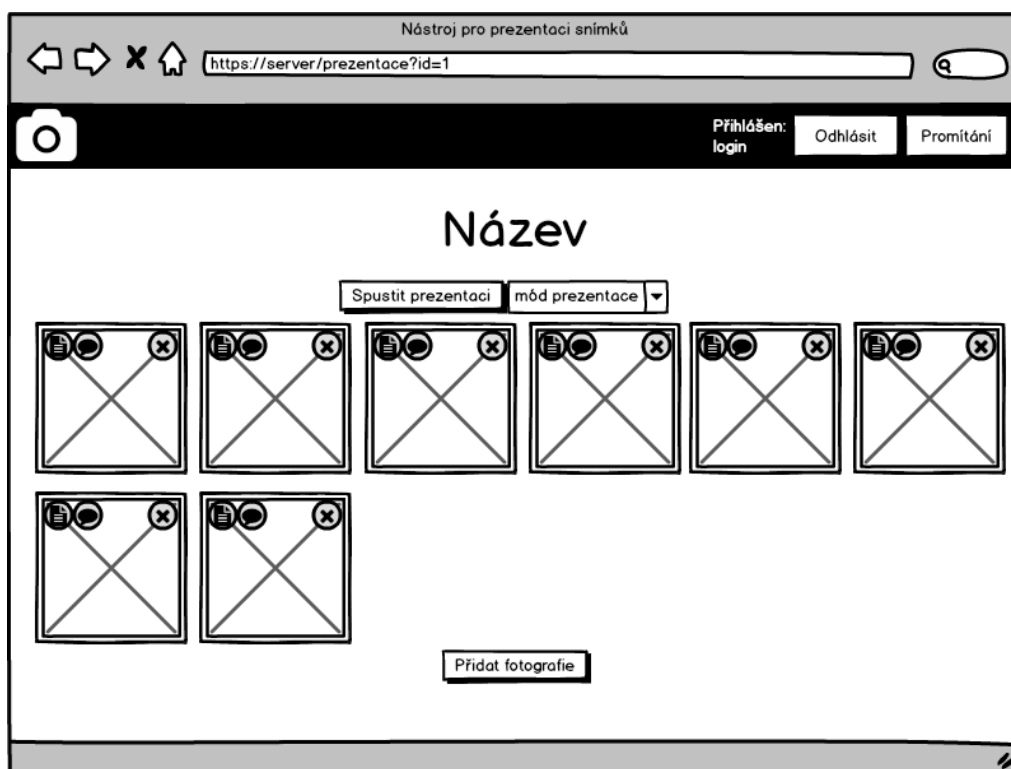


Obrázek 2.1: EER model databáze



Obrázek 2.2: Návrh uživatelského rozhraní seznamu prezentací (vstupní strana pro přihlášeného uživatele)





Obrázek 2.3: Návrh uživatelského rozhraní pro úpravy prezentace

## 2.3 Zobrazení aplikace

Správa prezentací bude fungovat přes grafické uživatelské rozhraní. Při vlastní prezentaci bude zobrazení rozšířeno na celou obrazovku. Zobrazení bude ve dvou možných rolích. První nazvaná „master“ bude řídit prezentace na všech ostatních zobrazeních téhož uživatele, tato další zobrazení budou v roli „slave“. Zobrazení v roli „master“ může mít každý uživatel v jednom momentě puštěné pouze jedno, zatímco zobrazeních v roli „slave“ může být neomezeně mnoho.

Při vstupu do aplikace je uživatel, pokud již není přihlášen, uvítán přihlašovacím dialogem s odkazem na registraci. Přihlášenému uživateli se zobrazí výpis jeho prezentací. Kliknutím na název prezentace ve výpisu se zobrazí její podrobnosti. Nahoře v hlavičce se přihlášenému uživateli zobrazí jeho uživatelské jméno s odkazem na úpravu profilu, tlačítko pro odhlášení a tlačítko ke spuštění zobrazení prezentace na celou plochu v roli „slave“. Návrh rozhraní vytvořený v Balsamiq Mockups [14] je znázorněn na obrázku 2.2.

## 2. NÁVRH

---

Registrace, přihlášení a úprava profilu budou jednoduché formuláře se vstupními poli typu text a heslo a s tlačítkem. Aplikace bude ke kontrole zadaných dat na straně klienta používat nových možností jazyka HTML z verze HTML5. Bližší podrobnosti naleznete v následující kapitole. Kontrola dat samozřejmě bude probíhat i na serveru, a to nejen ověření platnosti vyplněných hodnot, ale také zda není uživatelské jméno při registraci již obsazeno.

Při změně hesla je pro zamezení CSRF<sup>2</sup> útoků vyžadováno zadání i původního hesla, tak aby útočník nemohl převzít kontrolu nad cizím účtem. Pro eliminaci dalších možných CSRF útoků by bylo potřeba při každém přihlášení uživatele vygenerovat náhodný autorizační token, který by byl odeslán na server s každým požadavkem na provedení akce [15].

Z hlediska návrhu je zajímavější zobrazení konkrétní prezentace, kde uživatel může změnit její název, přidat fotografie, mazat fotografie, měnit poznámky a nadpisy fotografií a přetažením měnit pořadí fotografií v rámci prezentace. Návrh rozhraní je na obrázku 2.3.

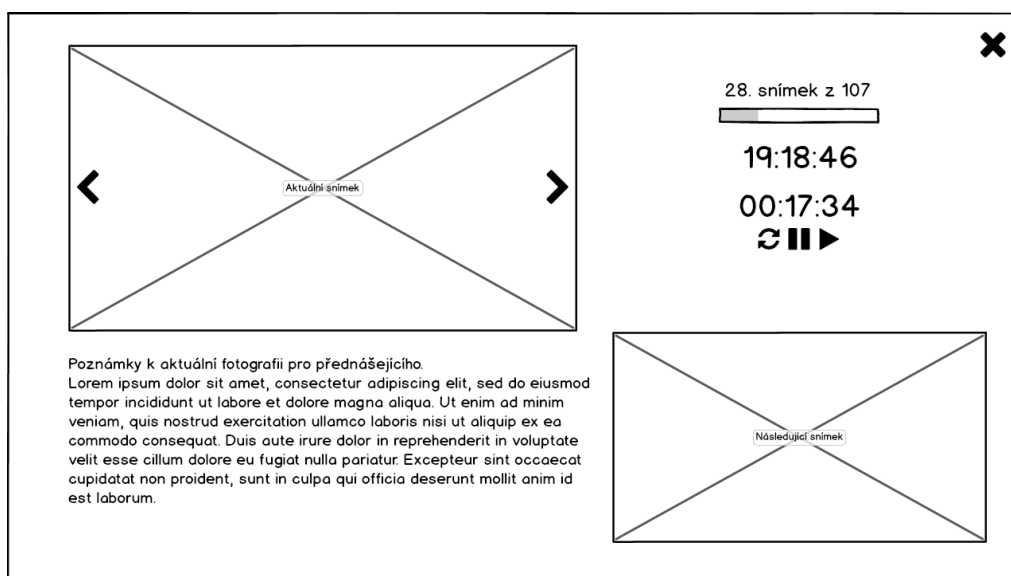
Během prezentace, při zobrazení na celou plochu, se zobrazí fotografie ve výšce odpovídající výšce zobrazovací plochy a pokud fotografie obsahuje nadpis, tak jej aplikace zobrazí jako text v levém dolním rohu. Velikost fontu se musí měnit s velikostí zobrazovací plochy, tak aby byl nadpis vždy dobře čitelný.

O mnoho zajímavější, z hlediska návrhu, je zobrazení pro přednášejícího. Návrh tohoto zobrazení je na obrázku 2.4. Je navržen tak, aby splňoval požadavky z kapitoly 1.1.1.3 o analýze. Obsahuje náhled současného a následujícího snímku, poznámku přednášejícího, stav prezentace a aktuální čas se stopkami. Navíc obsahuje ikonky pro přesun na následující, nebo předchozí snímek, ikonku pro ukončení prezentace a ikonky pro ovládání stopek (vynulování, pozastavení a spuštění). Prezentaci bude krom ikonky také možno ovládat pomocí klávesnice a pomocí gest na dotykových zařízeních.

Panoramatické snímky se budou zobrazovat stejně jako klasické fotografie. Aplikace rozpozná, pokud je fotografie širší než zobrazovací plocha. V tom případě, pokud je zvolen režim automatického posunu, se fotografie začne sama posouvat, tak aby byla postupně zobrazena celá. Pro lepší představu diváků se po dokončení posunu začne posouvat zpět. V případě ručního režimu je fotografii možné posouvat pomocí myši nebo pomocí prstu na dotykových zařízeních.

---

<sup>2</sup>CSRF (Cross-site Request Forgery) je jedna z metod útoku na internetové aplikace pracující na bázi nezamýšleného požadavku pro vykonání určité akce v této aplikaci.



Obrázek 2.4: Návrh uživatelského rozhraní zobrazení pro přednášejícího

## 2.4 WebSocket server

WebSocket server může být napsán v libovolném programovacím jazyce, který se používá pro serverovou stranu, jako například C++, Python, PHP nebo JavaScript. Pro ty nejpoužívanější existují předpřipravené knihovny podle WebSocket specifikace. Programátor se tedy nemusí sám zabírat všemi detaily komunikace. Tedy například jak přesně probíhá handshake a podobně. Po prostudování různých metod jsem se rozhodl využít JavaScript na straně serveru v prostředí Node.js. Jednak je JavaScript mým oblíbeným jazykem a s Node.js jsem se seznámil v rámci předmětu MI-W20. Mezi hlavní výhody Node.js patří škálovatelnost a rychlost [16].

### 2.5 Certifikát

Má-li aplikace používat šifrované protokoly HTTPS a wss, musíme mít pro ni zajištěn certifikát. Kromě těch placených, které je samozřejmě možné bez problémů použít, lze certifikát získat i bezplatně, a to díky certifikační autoritě Let's Encrypt [17], která nabízí důvěryhodné certifikáty zdarma. Let's Encrypt ale vydává pouze DV certifikáty pro ověření domény. Dokud tedy nebudeme mít pro aplikaci přiřazeno doménové jméno, musíme se v rámci testování spolehnout na tzv. certifikát podepsaný sám sebou, který si můžeme vygenerovat pomocí `openssl` příkazu následovně [18]:

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048  
-keyout /etc/ssl/private/apache-selfsigned.key  
-out /etc/ssl/certs/apache-selfsigned.crt
```

---

## Realizace

Nyní, když máme jasné požadavky a návrh aplikace, již přistoupíme k samotné realizaci.

### 3.1 HTML5

HTML je značkovací jazyk používaný k tvorbě webových stránek. Vznikl roku 1990 v CERNu pro jednodušší tvorbu dokumentů. Spolu s HTML byl navržen také protokol HTTP. Aplikace je psána ve verzi HTML5 [19]. Oproti předchozím verzím přináší mnohá vylepšení. V roce 2016 byla dokončena verze HTML 5.1, která již má status „doporučení“ a přináší další drobná vylepšení. V této sekci se podíváme na ta, která jsou v rámci aplikace využita.

#### 3.1.1 Typ dokumentu

První změnou oproti starším verzím je definice typu dokumentu, která se v HTML5 zapisuje následovně:

```
<!DOCTYPE html>
```

Není tedy třeba již uvádět verzi a odkaz na DTD<sup>3</sup> jako v předchozích verzích.

---

<sup>3</sup>DTD (Document Type Definition) je jazyk pro popis struktury XML. Slouží k vymezení elementů, které je možné použít, a popisuje, jakým způsobem mohou být vnořené.

### 3.1.2 Odkaz

Do elementu `<a>` je nyní možné vložit i blokové elementy. V aplikaci je toto využito pro vložení odkazu na logo aplikace, které je do stránky vloženo jako pozadí elementu `<div>`.

### 3.1.3 Struktura dokumentu

HTML5 definuje mnoho nových elementů pro strukturu dokumentu. Tam kde se dříve používaly pouze elementy `<div>`, lze nyní použít nové elementy, které mají navíc definovaný sémantický význam. V aplikaci jsou z nich použity:

**header** Je element určený pro hlavičku stránky. V rámci popisované aplikace se jedná o horní lištu s logem, informací o uživateli s odkazem na změnu profilu a tlačítka k odhlášení a zahájení prezentace v roli „slave“.

**main** Je element, který obsahuje hlavní obsah dokumentu.

### 3.1.4 Formuláře

Mimo struktury je nejvíce novinek v HTML5 právě ve formulářích. Krom mnoha nových typů pro vstupní elementy je zde také několik nových atributů, z nichž se v aplikaci vyskytují:

**placeholder** Je atribut, jehož hodnota je zobrazena ve vstupním poli do té doby, než uživatel začne toto pole vyplňovat. Obvykle bývá méně výrazný než zadaný text. Obsahuje popis toho, jaký údaj případně v jakém formátu se má do tohoto pole zadat.

**pattern** Hodnota tohoto atributu obsahuje regulární výraz se kterým se porovnává zadaná hodnota. Formulář je možno odeslat až v momentě, kdy hodnoty všech vstupních polí odpovídají svým regulárním výrazům. Kontrola vstupních dat je přesto stále nutná i na straně serveru. Příklad pro kontrolu uživatelského jména, které musí mít mezi 4 a 20 znaky a kde jsou povolena pouze malá a velká písmena a číslice:

```
pattern="[A-Za-z0-9]{4,20}"
```

**required** Určuje, zda je povinné dané pole vyplnit.

**autofocus** Určuje, zda bude po načtení stránky dané pole vybráno k zadávání textu.

**multiple** U vstupního elementu typu `file` je pomocí tohoto atributu řečeno, že vybraných souborů může být více než jeden. Příklad vstupního elementu pro nahrávání fotografií:

```
<input name="files[]" id="files" type="file" title=
"Přidat fotografie" accept=".jpg,.jpeg,.png,.gif,
image/png,image/jpeg,image/gif,image/jpeg" multiple>
```

### 3.1.5 Vlastní data k elementům

HTML5 také umožňuje k jednotlivým elementům vkládat vlastní atributy, které se následně dají využít například v JavaScriptu. Takový atribut musí začínat prefixem `data-` a neměl by mít v názvu velká písmena. Vlastní atribut `data` by také neměl být použit, pokud pro danou hodnotu existuje sémanticky vhodnější atribut nebo element. V této aplikaci je tato možnost využita například pro uložení nadpisu k dané fotografii do atributu `data-nadpis`. K hodnotě je pak možno přistoupit pomocí jQuery metody `data()`. Například následující kód vrátí nadpis fotografie s daným id:

```
$("#img_" + id).data('nadpis');
```

### 3.1.6 Upravitelný obsah

`Contenteditable` je globální atribut, který, pokud je nastaven na hodnotu `true`, říká o daném prvku, že může být upravován uživatelem. V aplikaci je tento atribut použit k možnosti změny názvu prezentace a k možnosti změny poznámky u fotografie, a to i v zobrazení pro přednášejícího. Pokud tedy uživatele napadne cokoliv do poznámky k dané fotografii při zobrazení v tomto režimu, tak jej nemusí ukončit, ale může poznámku změnit přímo v něm. Při změně se nový obsah ihned přenáší pomocí XHR požadavku na server a ukládá se do databáze. Příklad pro název prezentace:

```
<h1 id="nazev" contenteditable="true" data-input-length="40">
  <?php echo htmlspecialchars($nazev, ENT_QUOTES); ?></h1>
```

V tomto případě je ve vlastním atributu `data-input-length` uložena maximální povolená délka názvu. Ta je pak kontrolována pomocí JavaScriptu.

## 3.2 Kaskádové styly

Kaskádové styly (CSS) jsou jazykem popisujícím pravidla pro zobrazení jednotlivých elementů v rámci dokumentu. Každé pravidlo obsahuje selektor, který definuje elementy, na které se dané pravidlo vztahuje, a blok deklarací. Každý blok deklarací pak obsahuje jednotlivé deklarace oddělené středníkem [20]. Vyvíjená aplikace používá verzi CSS3, která vešla v platnost roku 2015. CSS3 nově podporuje například zaoblené rohy, stíny blokových elementů i textu a průhlednost. Všechny kaskádové styly jsou uloženy v externích souborech a do aplikace vloženy v hlavičce dokumentu pomocí elementu `<link>` s atributem `rel` nastaveným na hodnotu `stylesheet`:

```
<link href="layout.css" rel="stylesheet" type="text/css">
```

Tím je oddělen vzhled dokumentu od jeho struktury a obsahu.

V aplikaci jsou použity celkem čtyři CSS soubory. Tři z nich jsou součástí jQuery UI a popisují způsob zobrazení jQuery UI prvků. Čtvrtým je soubor `layout.css`, kde jsou popsána pravidla pro zobrazení celé aplikace s výjimkou výše zmíněných jQuery UI prvků.

V rámci zobrazení pro přednášejícího jsou použity výhradně relativní jednotky, a to převážně hodnoty v procentech nebo jednotkách `vmin`. Hodnota `1 vmin` je velikost jedné setiny kratšího z rozměrů zobrazené plochy (výšky nebo šířky). Tím je docíleno responzivního zobrazení na jakémkoliv možném rozlišení a poměru stran zobrazovací plochy.

### 3.2.1 Zobrazení na mobilních zařízeních

CSS umožňuje pomocí pravidla `@media` omezit platnost vložených deklarací pouze za určených podmínek. Podmínky je možné logicky negovat i kombinovat. V aplikaci jsou například pomocí následující ukázky kódu schovány prvky pro spuštění prezentace, pokud je zařízení otočeno na výšku. V případě otočení na šířku se zobrazují.

```
@media screen and (orientation: portrait) {  
    #show, #prezentace, .fotka .play {  
        display: none;  
    }  
}
```



Podmínka **screen** značí, že se tato pravidla aplikují pouze při zobrazení dokumentu na displeji. Druhá podmínka **orientation: portrait** znamená, že se použijí pouze při orientaci obrazovky na výšku. Mimo to existují další možné podmínky. Často bývají použity podmínky pro maximální nebo minimální hodnotu šířky nebo výšky zobrazené plochy.

#### 3.2.1.1 Viewport

Krom vlastních kaskádových stylů je pro lepší čitelnost na mobilních zařízeních v hlavičce dokumentu v elementu `<meta>` s atributem **name** nastaveným na hodnotu **viewport** definován výřez zobrazení, který zajistí, aby se stránka nezobrazila zmenšená, ale aby se zobrazila v nastaveném měřítku. Příklad užití:

```
<meta name="viewport" content="user-scalable=no,
initial-scale=1, maximum-scale=1, width=device-width">
```

## 3.3 PHP

PHP je jedním z nejrozšířenějších jazyků pro psaní webových stránek a aplikací. Každý skript, který generuje stránku k zobrazení, nejprve vloží soubor `ses.php`, ten obstarává obsluhu relací a vkládá soubor `db.php`, který definuje připojení k databázi. Soubor `ses.php` dále obstarává přihlašování a odhlašování uživatelů, kontroluje, zda je přihlášený uživatel stále v databázi a obstarává vytvoření nové prezentace. Dále je vkládán soubor `head.php`, který obsahuje hlavičku HTML dokumentu. Při změně nějakého elementu v hlavičce tak stačí provést změnu na jednom místě a projeví se na všech stránkách. Dalším vloženým skriptem je `hlavicka.php`, ten vkládá do stránky element `<header>`, který tvoří horní lištu. Pak již každý skript vkládá svůj vlastní obsah. Na konci stránky je vložen poslední soubor `footer.php`, který dokument uzavře.

### 3.3.1 Registrace, přihlášení, relace

Uživatel se musí nejdříve zaregistrovat. Registraci včetně zobrazení registračního formuláře obstarává soubor `registrace.php`. Formulář obsahuje 3 vstupní pole. První je pro zvolené uživatelské jméno a druhá dvě pro heslo a pro kontrolní zadání hesla. Pokud všechna pole splňují své regulární výrazy, může uživatel formulář odeslat. Na straně serveru pak znovu proběhne kontrola vyplněných údajů. Je-li vše v pořádku, tak se ověří, zda je

zvolené uživatelské jméno volné. Pokud ano, tak se nový uživatel uloží do databáze. Uživatelské jméno se uloží tak jak je, heslo se uloží pouze jako tzv. „osolený“ otisk. To proto aby bylo odolnější proti útoku s předpočítanými slovy odpovídajícími nalezenému otisku (tzv. duhovými tabulkami). Na heslo se nejprve aplikuje funkce `md5()`. Výsledná hodnota se zřetězí s uživatelským jménem daného uživatele a nad vzniklým řetězcem se zavolá funkce `sha1()`, která vygeneruje 40 znaků dlouhý otisk, který se uloží do databáze. Jako třetí údaj se uloží synchronizační id uživatele, které je použito pro WebSocket spojení. To se získá jako `md5` otisk uživatelského jména uživatele zřetěženého s časovou značkou registrace. Po úspěšné registraci je uživatel přesměrován na vstupní stránku, kde se zobrazí informace o úspěšné registraci a uživatel se může ihned přihlásit.

Při přihlašování se na serveru také nejprve ověří platnost vyplněných dat, poté se z databáze získá otisk hesla a synchronizační id uživatele. Pokud SQL dotaz nevrátí žádný řádek, znamená to, že uživatel zadaného uživatelského jména není registrován a uživateli se zobrazí příslušná chybová hláška. V opačném případě se ověří, zda otisk zadaného hesla souhlasí s otiskem hesla z databáze. Pokud ano, tak je uživatel přihlášen nastavením relace. Do globálního pole `$_SESSION` se uloží identifikace přihlášeného uživatele a jeho synchronizační id.

Při odhlášení uživatele se z globálního pole odstraní jeho identifikace a synchronizační id. Poté je uživatel přesměrován na vstupní stránku. Při každém načtení stránky, pokud je uživatel přihlášen, se kontroluje, zdali je přihlášený uživatel stále v databázi.

#### 3.3.2 Nahrávání a zpracování fotografií

Zpracování fotografií na serveru má na starost soubor `upload.php`. V jednom požadavku je poslána vždy právě jedna fotografie. Na začátku skript zkontroluje, jestli je přihlášený uživatel vlastníkem prezentace, do které chce fotografii nahrát. Poté je zkontrolována velikost souboru, zda nepřekračuje limity serveru, a typ souboru, je-li podporován. Následně se vytvoří unikátní název souboru pomocí funkce `sha1_file()` aplikované na nahrávaný soubor, poté se ještě zřetězí s časovou značkou nahrání a provede se funkce `sha1()`, tak aby jedna fotografie mohla být v prezentaci i vícekrát, což je užitečné například pro mapu cesty, kterou může přednášející chtít ukázat na začátku i na konci prezentace. Pokud fotografie obsahuje metadata EXIF, tak jsou z nich extrahovány informace o datu pořízení a přístroji, jakým byla fotografie pořízena. Tyto údaje jsou následně vloženy do poznámky k dané fotografii, tu si později může uživatel libovolně měnit. Následuje uložení fotografie v celkem pěti různých rozlišeních. Fotografie se ukládá s vertikálním

rozlišením 600, 768, 1080, 2160 a 4320 pixelů, aby aplikace mohla vybrat vždy nejvhodnější soubor pro danou zobrazovací plochu. Následně se ještě uloží čtvercový náhled o velikosti 140 x 140 pixelů. Všechny soubory se na serveru ukládají do adresáře, který je pojmenován jako id dané prezentace. Tyto adresáře jsou umístěny v adresáři **presentations**. Po úspěšném zpracování se ještě informace o fotografii uloží do databáze a odešle se odpověď na požadavek ve formátu JSON obsahující vygenerovaný název fotografie, typ a poznámku. Jakým způsobem jsou zasílány požadavky ze strany klienta je popsáno dále v sekci 3.5.2.1 věnované JavaScriptu.

### 3.3.3 Asynchronní požadavky

Krom nahrávání souborů obstarává PHP na serveru také obsluhu všech dalších asynchronních požadavků jako je např. ukládání názvu prezentace, poznámky k fotografii, uložení pořadí snímků a jiné. Obsluha probíhá vždy podle stejného scénáře. Požadavek se uloží do databáze a pokud vše proběhne bez problémů vrátí se ve formátu JSON hodnota „1“. Dojde-li k chybě při zpracování požadavku, tak se místo hodnoty „1“ vrátí řetězec s popisem chyby, který je poté zobrazen uživateli.

## 3.4 Databáze

Databázi jsem vytvořil v MySQL dle návrhu z předchozí kapitoly. MySQL je v tomto případě vzhledem k vlastnostem ukládaných dat a očekávanému počtu uživatelů nejrychlejší, ale díky použití PDO [21] pro připojení k databázi, je možné použít i jiné databáze. V případě, kdy by aplikaci využívalo větší množství uživatelů, bylo by vhodnější použití např. PostgreSQL. Připojení k databázi obstarává soubor `db.php`.

## 3.5 JavaScript na straně klienta

Hlavní těžiště mé aplikace je právě v JavaScriptu na straně klienta. JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk, jehož autorem je Brendan Eich z tehdejší společnosti Netscape. Nyní se zpravidla používá jako interpretovaný programovací jazyk pro webové stránky, byť některé z posledních verzí prohlížečů používají JIT<sup>4</sup> kompilaci. Slouží k interakci s DOM<sup>5</sup>. Jako příklad nejčastějšího užití můžeme uvést ovlá-

---

<sup>4</sup>JIT (Just In Time) je metoda překladač, kdy je kód zkompileován při spuštění.

<sup>5</sup>DOM (Document Object Model) je objektově orientovaná reprezentace HTML dokumentu.

dání různých interaktivních prvků uživatelského rozhraní, tvoření animací a efektů obrázků nebo asynchronní zpracování datových požadavků.

Jeho syntaxe patří do rodiny jazyků C/C++/Java. Ale JavaScript je od těchto jazyků zásadně odlišný, sémanticky jde o jiný jazyk. Slovo Java je součástí jeho názvu pouze z marketingových důvodů. JavaScript byl v červenci 1997 standardizován asociací ECMA a v srpnu 1998 ISO. Standardizovaná verze JavaScriptu je pojmenována ECMAScript a byly z ní odvozeny i další implementace, jako je například ActionScript [22].

#### 3.5.1 Knihovny

Pro zjednodušení práce na straně klienta používám několik knihoven. Nyní se na ně podíváme a popíšeme si, k čemu slouží.

##### 3.5.1.1 jQuery

Knihovna jQuery je otevřený software pod licencí MIT a velmi usnadňuje psaní kódu v JavaScriptu použitém na straně klienta. První verze byla vydána Johnem Resigem v lednu 2006 na newyorském BarCampu. Vhodně také odděluje jednotlivé funkce od HTML kódu. Knihovnu je potřeba do stránek vložit dříve než samotný kód využívající jQuery. Vložení se provede pomocí elementu `<script>`, který je vložen v hlavičce dokumentu s atributem `defer`, který značí že se skript spustí až po zpracování celého HTML dokumentu, aby nebylo zpomaleno jeho zobrazení. V rámci této aplikace je použita verze 3.2.1.

```
<script src="jquery-3.2.1.min.js" defer></script>
```

Další atributy, na které jsme byli zvyklí z předchozích verzí HTML, již nejsou v HTML5 potřeba. V produkčním nasazení lze jQuery načítat z některé CDN<sup>6</sup> (nejčastěji bývají použity Google a jQuery), nebo pomocí Google AJAX Libraries API. To má výhodu v unifikovaném ukládání do mezipaměti a snížení odezvy. Pro kratší zápisy kódu je v jQuery definována tzv. funkce „\$“, což je alias pro jmenný prostor „jQuery“, ta na základě selektoru vrátí příslušné prvky dokumentu [23].

---

<sup>6</sup>CDN (Content Delivery Network) je infrastruktura distribuovaných serverů, které umožňují rychlejší přístup k datům než v případě umístění dat na vlastním serveru.

### 3.5.1.2 jQuery UI

jQuery UI je knihovna pod licencí MIT zaměřená na uživatelské rozhraní. Jejím cílem je ulehčit vývojářům implementaci různých ovládacích prvků a pokročilých efektů. Ty jsou navrženy pro použití na širokém spektru prohlížečů. Aktuálně je ve verzi 1.12.1. Vyvíjena je jako součást projektu The jQuery Project. jQuery UI se skládá z jádra, metod pro interakci, grafických komponent, efektů a pomůcek [24]. V rámci této aplikace jsou použity následující metody pro interakci:

**draggable** Je metoda použitá pro ruční posun fotografie. V callbacku funkce je při posunu odeslána WebSocket zpráva typu `move`, která obsahuje normalizovanou hodnotu posunu.

**sortable** Tato metoda umožňuje řadit jednotlivé snímky v prezentaci pouhým přetažením. Při aktualizaci pořadí odesílá mnou navržený skript XHR požadavek pro uložení nového pořadí do databáze.

Z grafických komponent jsou v aplikaci použity:

**button** Pro zobrazení tlačítek a dalších formulářových prvků, pro ty je poté ještě potřeba upravit jejich CSS vlastnosti, tak aby se chovaly dle očekávání uživatele.

**controlgroup** Pro zobrazení ovládacích prvků prezentace, které leží vedle sebe, tak aby se graficky lépe vykreslily.

**dialog** Pro zobrazení dialogů a chybových hlášek.

**progressbar** Pro grafické znázornění stavu aplikaci při nahrávání souborů.

**selectmenu** Pro nastavení módu prezentace.

**tooltip** K zobrazení nápovědy.

### 3.5.1.3 jQuery UI Touch Punch

Knihovna jQuery UI Touch Punch je rozšíření pro jQuery UI, tak aby jednotlivé funkce bylo možné používat na zařízeních s dotykovým ovládáním [25]. Knihovna funguje na principu mapování dotykových událostí na události myši a je licencovaná pod MIT i GPL licencí. Knihovnu jsem musel sám trochu upravit, protože v původním stavu nebylo možné používat ikonky umístěné na elementu s vlastností **sortable**, tedy v našem případě ikonky

### 3. REALIZACE

---

u fotografií ve výpisu prezentace. Nebylo tak možné na dotykových zařízeních měnit poznámku, nadpis, mazat fotografie z prezentace ani spustit prezentaci z konkrétního snímku. Všechny tyto ikonky mají třídu `nosort`. Do knihovny jsem tedy do funkce `simulateMouseEvent` přidal následující kód:

```
if ($(touch.target).is(".nosort")) {  
    event.stopPropagation();  
} else {  
    event.preventDefault();  
}
```

Ten zajistí, že na elementy s třídou `nosort` bude možné kliknout i na dotykových zařízeních.

#### 3.5.1.4 Hammer.js

Hammer.js [26] je knihovna určená pro podporu rozpoznávání ovládání pomocí gest. Licencovaná je pod MIT licencí. Dokáže rozpoznat gesta jakými jsou:

**tap** Registruje dotyk.

**doubletap** Registruje dvojité poklepání.

**swipe** Registruje posun prstem po obrazovce. Lze zvolit buď libovolný, nebo konkrétní směr pohybu.

**press** Registruje dotyk na delší časový okamžik, ve výchozím nastavení nejméně 251 ms.

**pinch** Registruje posun dvou, nebo více, prstů k sobě nebo od sebe.

**rotate** Registruje otočení dvou, nebo více, prstů kolem jedné osy.

Navíc existuje zásuvný modul `jquery.hammer.js` pro jQuery, s jehož pomocí je možné jednotlivá gesta registrovat stejně jednoduše jako ostatní události. Příklad konstrukce pro spuštění kódu po detekci události `swiperight` na elementu s id `container`:

```
$('#container').hammer().bind('swiperight',function(ev){...});
```

V aplikaci používám právě gesta `swiperight` a `swipeleft` pro ovládání přechodu mezi jednotlivými snímky prezentace.

### 3.5.1.5 NoSleep.js

Při průběžném testování mi při ovládání z mobilního zařízení vadilo uzamykání zařízení. Původně jsem si myslel, že vzhledem k tomu, že se jedná o webovou aplikaci, která z bezpečnostních důvodů samozřejmě nemá přístup k ovládání systému, s tím, krom nastavení delšího času do uzamknutí, mnoho udělat nepůjde. Naštěstí jsem se mýlil. Bylo vysledováno, že k uzamknutí nedojde, pokud se na stránce přehrává video. Pár vývojářů pak vytvořilo různé knihovny, které do stránky vloží malé video, jenž se však nezobrazí a přehrává se ve smyčce. Z dostupných knihoven jsem si vybral NoSleep.js [27], která na rozdíl od jiných neodkazuje na externí video, ale video má obsaženo přímo v JavaScriptovém kódu a je licencována pod MIT licenci.

Knihovna obsahuje dvě funkce. První je `noSleep.enable()`, která spustí smyčku přehrávání videa, tedy zahájí část kódu, během které zařízení neušne. Tuto funkci je nutné volat z události, která byla vyvolána uživatelem, jako např. kliknutí myši. V aplikaci ji volám při začátku prezentace. Druhou je analogicky `noSleep.disable()`, která smyčku ukončí a zařízení tak po určené době nečinnosti může zase usnout. V aplikaci ji volám při ukončení prezentace.

## 3.5.2 Funkce

JavaScriptý kód ovládající aplikaci je rozdělen do dvou souborů. Prvním je `jquery-ui-init.js`, který obsahuje pouze inicializaci jQuery UI prvků, které se zobrazují, dokud není uživatel přihlášen. Druhým, hlavním, souborem je `functions.js`, který obsahuje veškerou funkcionální aplikaci potřebnou pro přihlášeného uživatele, tento soubor má po minifikaci necelých 20 kiB a načítá se pouze pokud je uživatel přihlášen.

Kód používá několik globálních proměnných. Dvě z nich jsou vloženy do kódu v souboru `head.php`, tak aby mohly obsahovat hodnoty převzaté z PHP kódu:

**clientId** Obsahuje synchronizační id klienta pro identifikace na WebSocket serveru.

**server** Obsahuje adresu serveru z proměnné `$_SERVER['HTTP_HOST']`, ta je použita pro navázání WebSocket spojení, běží-li WebSocket server na stejné adrese jako webový server.

### 3. REALIZACE

---

Dalších pět proměnných, potřebných pouze na stránce s prezentací, je do kódu vloženo v souboru **prezentace.php**:

**prezentace\_id** Obsahuje id aktuálně otevřené prezentace.

**file\_max\_size** Obsahuje maximální povolenou velikost jednoho souboru pro odeslání na server. Hodnota je získána z **php.ini** pomocí funkce `ini_get('upload_max_filesize')` v bajtech a slouží ke kontrole nahrávaných souborů před odesláním na server, tak aby se data zbytečně neposílala, když by nahrávání stejně ve výsledku skončilo s chybou.

**file\_max\_size\_string** Obsahuje předchozí hodnotu převedenou na vhodné jednotky pro zobrazení uživateli.

**post\_max\_size** Obsahuje maximální povolenou velikost jednoho požadavku pro odeslání na server. Hodnota je získána z **php.ini** pomocí funkce `ini_get('post_max_size')` v bajtech a slouží také ke kontrole dat před odesláním.

**post\_max\_size\_string** Obsahuje předchozí hodnotu převedenou na vhodné jednotky pro zobrazení uživateli.

Následují proměnné deklarované přímo v souboru **functions.js**:

**role** Obsahuje informace o roli aktuálně spuštěné prezentace. Může obsahovat hodnoty:

**0** značící, že aktuálně není spuštěna prezentace

**1** pro roli „master“

**2** pro roli „slave“

**connection** Při prezentaci obsahuje instanci třídy **WebSocket**.

**position** Do této proměnné se uloží aktuální hodnota posunu na stránce před zahájením prezentace, aby se po jejím skončení mohl obnovit původní stav.

**actual** Obsahuje informace o aktuálním snímku prezentace nebo hodnotu „0“, pokud prezentace není spuštěna.

**mode** Obsahuje zvolený režim prezentace. Může obsahovat hodnoty:

**0** při neaktivní prezentaci

**1** pro automatický posun při zobrazení na celou plochu



**2** pro ruční posun při zobrazení na celou plochu

**3** pro automatický posun v režimu zobrazení pro přednášejícího

**4** pro ruční posun v režimu zobrazení pro přednášejícího

**offset** Obsahuje hodnotu rozdílu systémového času na webovém serveru a na klientovi. Tato hodnota je použita pro zobrazení aktuálního času v režimu pro přednášejícího. Může se stát, že přednášející bude prezentaci ovládat z nějakého zapůjčeného počítače, kde nebude čas nastaven správně. Z tohoto důvodu je jako přesnější brán čas na webovém serveru.

**time** Je proměnná pro identifikátor funkce `displayTime` zobrazující čas a stopky, která je při prezentaci v zobrazení pro přednášejícího volána každou sekundu.

**stopky** Obsahuje logickou hodnotu, která určuje, zda stopky běží.

**casOd** Obsahuje čas, kdy došlo k poslednímu spuštění stopek. Protože funkce `displayTime` nemusí být spuštěna vždy přesně po sekundě. Je čas na stopkách počítán jako rozdíl aktuálního času a času z této proměnné.

**uplynulo** Pokud dojde k pozastavení stopek, tak se do této proměnné uloží aktuální hodnota zobrazená na stopkách, aby se po znovuspuštění stopek přičítala k naměřenému času.

**fotky** Obsahuje pole s informacemi o fotografiích aktuálně přehrávané prezentace.

**show\_id** Obsahuje id prezentace aktuálně spuštěné prezentace.

**rychlost** Rychlost posunu fotografie při režimu automatického posunu je vypočítána z poměru jejích stran vynásobené touto proměnnou. Změnou této hodnoty je možné ovlivnit čas posunu dané fotografie. Při nižší hodnotě bude posun rychlejší, naopak při vyšší bude pomalejší.

**noSleep** Obsahuje instanci třídy `NoSleep`, nad kterou je možné volat metody pro zahájení a ukončení přehrávání smyčky videa pro zamezení uzamknutí přístroje během prezentace.

JavaScript v souboru `functions.js` obstarává obsluhu všech možných událostí, jako například kliknutí myši na určité prvky, stisknutí určité klávesy na klávesnici apod. Pokud událost vyvolá změnu dat, která jsou uložena

### 3. REALIZACE

---

v databázi, tak je pomocí jQuery metody `post()` odeslán XHR požadavek na server k uložení nových dat do databáze. Zpracování požadavku na straně serveru je uvedeno v sekci 3.3.3 o PHP. Následuje příklad odeslání požadavku na uložení nového pořadí snímků prezentace.

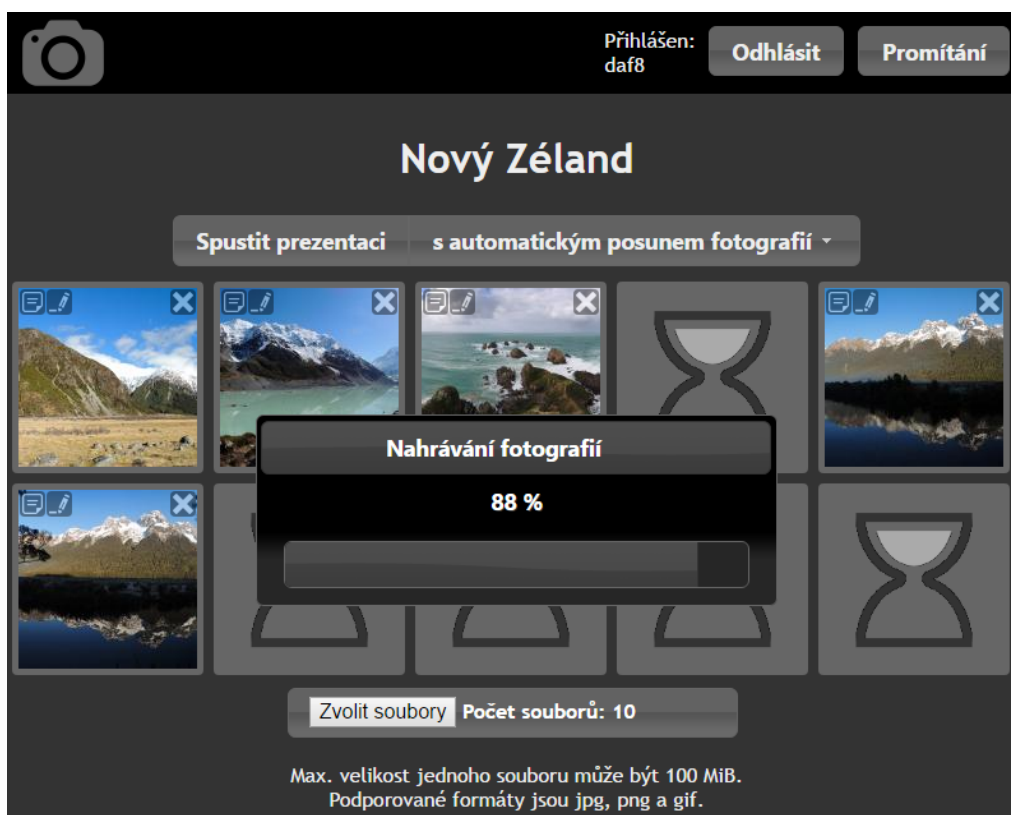
```
var poradi = $("#vypis").sortable('serialize');
$.post("sort.php",{
    data: poradi,
    id: prezentace_id
},function(data){
    if (data!=1){
        chyba("Došlo k chybě při ukládání pořadí snímků.<br>" + data);
    }
});
```

#### 3.5.2.1 Nahrávání souborů

Aby mohl být uživatel v průběhu nahrávání souborů informován o průběhu nahrávání, je nutné k nahrávání použít asynchronní požadavky. Proces nahrávání fotografií se spustí změnou obsahu HTML elementu pro výběr souborů. Poté jsou všechny vybrané soubory zkontrolovány, zda jsou správného typu a jestli jejich velikost nepřekračuje nastavené hodnoty na serveru. Pokud je vše v pořádku, zobrazí se dialog s jQuery UI prvkem **progressbar**. Ten zobrazuje pouze stav odesílání souborů, nikoliv stav zpracování dat na serveru. Do výpisu položek jsou ihned na začátku vloženy nové položky. Místo náhledu fotografie obsahují zatím animaci přesýpacích hodin viz obrázek 3.1. Poté jsou postupně v krátkých časových intervalech odeslány XHR požadavky s jednotlivými soubory. V průběhu odesílání se neustále aktualizuje dialog oznamující stav nahrávání. Po skončení odesílání je dialog ihned schován a zatím nezpracované fotografie mají stále místo náhledu animaci přesýpacích hodin. Ihned po zpracování konkrétní fotografie, a to i před dokončením nahrávání ostatních fotografií, se místo přesýpacích hodin zobrazí její miniatura a tlačítka pro změnu popisu, vkládání nadpisu a odstranění fotografie. Pořadí fotografií je možné měnit i po celou dobu nahrávání nových fotografií.

#### 3.5.2.2 Prezentace v roli „master“

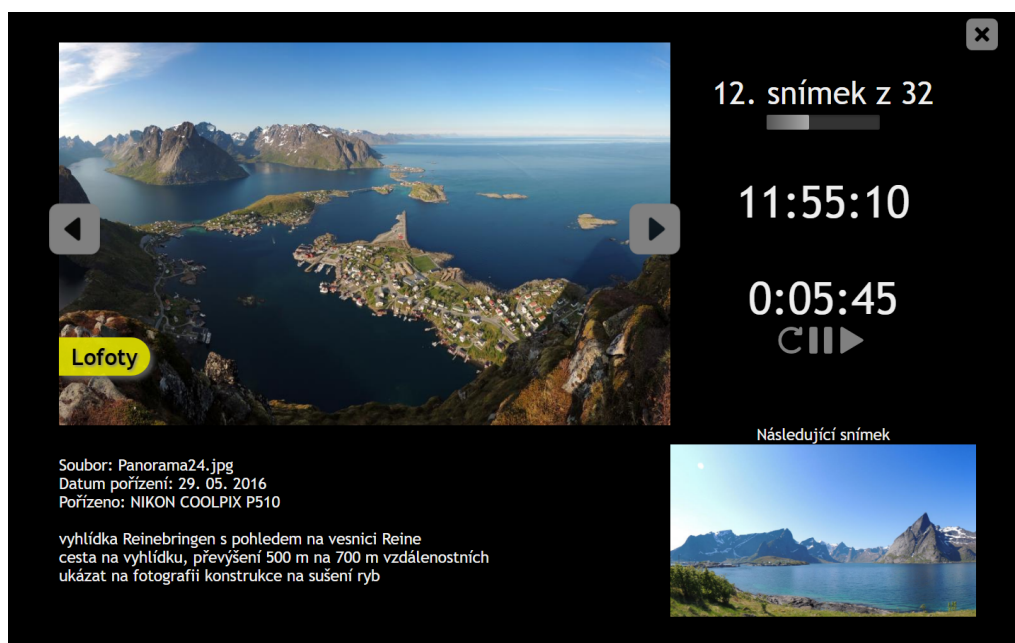
Prezentaci v roli „master“ je možné spustit pomocí tlačítka nebo kliknutím na konkrétní snímek. Nejprve je zobrazení rozšířeno na celou obrazovku.



Obrázek 3.1: Zobrazení stavu nahrávání fotografií. Pět souborů je již zpracováno. Z dalších pěti fotografií se některé nahrávají a některé zpracovávají.

Poté se odešle WebSocket zpráva typu `start` a zavolá se funkce `start()`. Ta do dokumentu vloží element `<div>` s id `container`, který se zobrazí na celou zobrazenou plochu a překryje vše ostatní na stránce. V tomto elementu jsou v případě režimu zobrazení na celou obrazovku prvky `<img>`, ve kterém bude zobrazena aktuální fotografie, a `<div>` pro zobrazení nadpisu fotografie. Poté se zavolá funkce `show()`, ve které se načte aktuální fotografie v nejvhodnějším rozlišení na danou zobrazovací plochu a dle zvoleného režimu se po načtení fotografie aplikují příslušná nastavení pro automatický nebo ruční posun.

Dále jsou zde funkce `next()` a `prev()` pro přechod na následující, nebo předchozí snímek. Při ukončení prezentace se pošle WebSocket zpráva typu `close` a zavolá se funkce `close()`, která z dokumentu odstraní element `<div>` s id `container`. Poté se ukončí zobrazení na celou obrazovku a stránka se zobrazí ve stejném stavu jako před začátkem prezentace. Všechny použité globální proměnné se vynulují.



Obrázek 3.2: Příklad zobrazení pro přednášejícího

**Zobrazení pro přednášejícího** Při spuštění zobrazení pro přednášejícího, které je zobrazeno na obrázku 3.2, se nejprve z webového serveru získá aktuální čas pro vypočtení časového rozdílu mezi serverem a klientem. Synchronizace není úplně přesná, protože v tomto případě nám rozdíl jedné sekundy vadit nebude. Je to zde z toho důvodu, kdyby prezentace probíhala na počítači se špatně nastaveným časem. Dále toto zobrazení funguje stejně jako zobrazení na celou plochu s tím rozdílem, že element `<div>` s id `container` obsahuje více prvků pro zobrazení všech potřebných údajů. Je zde element s id `nahled`, který ukazuje náhled toho, co se promítá na zobrazení v roli „slave“ včetně nadpisu fotografie. Navíc jsou zde tlačítka pro přechod mezi snímky. Dále je zde element s id `infobox` obsahující informace o postupu v rámci prezentace, aktuálním čase a jsou zde umístěny i stopky. Pod elementem s id `nahled` je umístěn element s id `poznamkybox`, ve kterém je umístěna upravitelná poznámka k fotografii. A napravo od něj je poslední element s id `nextbox`, kde je zobrazen náhled následujícího snímku. V pravém horním rohu je ještě tlačítko pro ukončení prezentace.

#### 3.5.2.3 Prezentace v roli „slave“

Prezentace v roli „slave“ funguje obdobně jako ta v roli „master“. Odlišný je začátek prezentace, kdy po kliknutí na tlačítko Promítání v horní liště,

se v elementu s id `container` zatím zobrazí informace: „Čekám na zahájení prezentace.“ Naváže se WebSocket spojení a čeká se na přijetí zprávy typu `start`. Po přijetí této zprávy se nastaví proměnné `show_id`, `actual` a `mode`, která je omezena pouze na režimy zobrazení na celou plochu, a zavolá se funkce `start()`, jenž funguje stejně jako v případě role „master“. Při přijetí zprávy typu `actual` se zavolá funkce `act()`, ta změní hodnotu v proměnné `actual` a zavolá funkci `show()`. Při přijetí zprávy typu `close` se zavolá funkce `endSlave()`, která změní element s id `container` tak, že obsahuje text: „Konec prezentace. Stiskněte ESC pro ukončení nebo můžete zahájit novou prezentaci.“ Globální proměnné `actual`, `mode` a `show_id` se vynulují a buď je možno prezentaci ukončit nebo počkat na zahájení nové. Kurzor je v této roli po celou dobu prezentace skryt, aby nezakrýval část fotografie.

### 3.5.3 WebSocket

Funkce obstarávající WebSocket spojení jsou umístěny v souboru `ws.js` a jde o následující funkce:

**open** Funkce `open` je volána při spuštění prezentace. Nejprve pošle serveru zprávu, ve které se klient představí svým unikátním synchronizačním id. Tato funkce obsahuje nepovinný parametr pro odeslání další zprávy, ten je použit v případě zobrazení v roli „master“, kdy je na server rovnou předána zpráva typu `start` pro zahájení prezentace na všech dalších zobrazeních v roli „slave“.

**close** Tato funkce je volána bez parametru a pokud existuje aktivní WebSocket spojení, tak jej ukončí.

**send** Pokud je navázané spojení, tak tato funkce zašle serveru zprávu, která je jí předána v parametru.

Další podrobnosti o implementaci WebSocket protokolu v této aplikaci najdete v následující sekci věnované WebSocket serveru.

## 3.6 WebSocket server

Jak již bylo zmíněno v předchozí kapitole, pro vytvoření WebSocket serveru jsem se rozhodl použít Node.js, tedy prostředí, ve kterém je možné napsat serverovou aplikaci v jazyce JavaScript. Programy pro Node.js hojně využívají model událostí a asynchronní I/O operace pro minimalizaci režie

### 3. REALIZACE

---

procesoru a maximalizaci výkonu [16]. Pokud stránky běží na zabezpečeném HTTPS protokolu, což je náš příklad, tak není možné použít klasický ws protokol a z bezpečnostních důvodů je potřeba použít zabezpečený wss.

Na straně serveru jsou použity následující moduly:

**fs** Tento modul slouží k práci se soubory. V naší aplikaci je potřeba k načtení soukromého klíče a certifikátu.

**express** ExpressJS je minimalistický Node.js framework, který poskytuje sadu funkcí pro webové a mobilní aplikace.

**https** HTTPS je modul, který je potřeba pro zabezpečenou komunikaci pomocí certifikátu.

**ws** Modul ws je implementace WebSocket serveru pro Node.js [28] licencovaná pod MIT licenci. V případě serveru pro nezabezpečený ws protokol bychom si vystačili pouze s tímto modulem.

WebSocket spojení je vždy navázáno až v momentě zahájení prezentace, a to v obou rolích „master“ i „slave“. Při zahájení spojení server očekává zprávu typu `open` s identifikací klienta. Pokud přijde jiná zpráva, tak spojení s chybou ukončí. Následující zprávy, pokud obsahují povolený typ, se přeposílají na všechny další klienty v roli „slave“ daného uživatele.

#### 3.6.1 Formát WebSocket zpráv

WebSocket posílá zprávy jako prosté řetězce textu. Pro snazší práci používám pro zprávy formát JSON, který je vždy před odesláním převeden na řetězec pomocí funkce `JSON.stringify(data)` a na druhé straně zpět na objekt pomocí funkce `JSON.parse(e.data)`. Prvním atributem zprávy je `type` označující typ zprávy. Další atributy záleží dle konkrétní instrukce. Následuje příklad zprávy:

```
var msg = {
  type: "open",
  role: role,
  id: clientId
};
```

Do aplikace jsem implementoval 6 typů zpráv:

**open** Zpráva typu `open` je první zpráva, kterou klient odešle serveru po navázání spojení, aby se představil. Obsahuje atributy `role`, který určuje, v jaké roli je aktuální klient spuštěn a `id` obsahující synchronizační id přihlášeného uživatele. Tato zpráva se ze serveru dále nepropaguje.

**error** Je jediný typ zprávy, kterou může server poslat klientovi sám od sebe. Obsahuje atribut `chyba` obsahující popis vzniklé chyby. Pokud se jedná o fatální chybu, po které není možné pokračovat, tak server spojení ukončí.

Následující typy zpráv odesílá pouze klient v roli „master“ a server tyto zprávy následně přepośle všem klientům v roli „slave“ daného uživatele.

**start** Tato zpráva je poslána při zahájení prezentace. Obsahuje atributy `id` obsahující id dané prezentace, `actual` obsahující aktuální snímek, ze kterého se prezentace spouští a `mode`, který říká, v jakém režimu je prezentace spuštěna. Na straně klienta v roli „slave“ je atribut `mode` upraven, protože zde bude prezentace probíhat vždy v režimu zobrazení na celou obrazovku.

**actual** Informuje o přechodu na následující, nebo předchozí snímek a obsahuje atribut `actual` označující aktuální snímek prezentace.

**close** Ukončuje prezentaci a neobsahuje žádný další atribut. Na zobrazení v roli „slave“ se zobrazí hláška o konci prezentace. Z režimu prezentace je pak možné vystoupit klávesou ESC nebo je možné rovnou bez ukončení spustit prezentaci novou.

**move** Při režimu ručního posunu informuje o aktuální poloze fotografie. Obsahuje atribut `left`, který obsahuje normalizovanou hodnotu, nabývajících hodnot od  $-1$  do  $0$ , aktuálního posunu fotografie. Hodnota je normalizována z toho důvodu, aby fungovala na všech možných poměrech stran zobrazovací plochy a aby nemohlo dojít k tomu, že by fotografie na některých zobrazeních nebyla zobrazena celá nebo byla vykreslována mimo zobrazenou plochu.

Po ukončení režimu prezentace se v obou použitých rolích WebSocket spojení ukončí. Pokud nedojde k žádné chybě, tak spojení ukončuje vždy klient.

WebSocket protokol je schopný zvládnout i několikaminutové výpadky spojení klienta v roli „slave“. Po opětovném spojení se přenesou všechny zprávy, které klient během přerušeného spojení zmeškal. Při delším výpadku dojde k uzavření WebSocket spojení a tím k ukončení prezentace. U klienta v roli „master“ je rozhodující, zda zrovna posílá synchronizační zprávy. V případě že zrovna žádné zprávy neodesílá platí totéž, co pro roli „slave“. V případě že se však snaží odeslat zprávu, tak je nutné, aby spojení bylo obnoveno do několika sekund, jinak dojde k ukončení spojení a tím i prezentace.

## 3.7 Manifest webové aplikace

Aplikace obsahuje také manifest webové aplikace [29], umístěný v souboru `manifest.json`. Ten je do stránky vložen v hlavičce následujícím tagem:

```
<link rel="manifest" href="manifest.json">
```

Manifest obsahuje informace o názvu aplikace, ikonky aplikace v různých rozlišeních, nastavení zobrazení bez adresního řádku prohlížeče, barvu pozadí aplikace a další. Aplikaci lze na mobilních zařízeních přidat mezi ostatní aplikace kliknutím v menu prohlížeče na možnost „Přidat na plochu“. Na ploše poté vznikne nová ikonka aplikace, tu je pak možné spustit jako kteroukoliv jinou nativní aplikaci.



## Návod k instalaci

### 4.1 Potřebné vybavení

K běhu této aplikace je potřeba webový server podporující PHP a databáze, tedy takzvaná sada LAMP. Doporučit mohu i například balík XAMPP [30], který je dostupný pro všechny možné platformy a je distribuován pod GPL licencí. Na webovém serveru je potřeba vhodně nastavit parametry v souboru `php.ini`, zejména:

**max\_execution\_time** Maximální čas, který může běžet jeden skript v sekundách (výchozí hodnota v XAMPP 30 s). Nahrávání a zpracování panoramatické fotografie ve vysokém rozlišení na starém nevykonném serveru trvá obvykle do jedné minuty. Doporučuji nastavit na 5 minut, tedy 300 sekund.

**max\_input\_time** Maximální čas, který může aplikace zpracovávat vstupní data (výchozí hodnota v XAMPP 60 s). Tato výchozí hodnota je dostačující, případně je možno nastavit hodnotu `-1`, která značí že se převezme hodnota z parametru `max_execution_time`.

**memory\_limit** Maximum operační paměti, kterou si může alokovat jeden skript (výchozí hodnota v XAMPP 128 MiB). Tato výchozí hodnota je dostatečná pro tuto aplikaci. Při testování je možno také nastavit hodnotu na `-1`, což je hodnota, která neomezuje velikost alokované paměti.

**post\_max\_size** Maximální velikost jednoho POST požadavku (výchozí hodnota v XAMPP 8 MiB). Aplikace je napsána tak, že každou fotografii posílá vlastním požadavkem. Tato hodnota by tedy měla reflektovat maximální očekávanou velikost jedné fotografie. Panoramatické

fotografie mohou mít i několik desítek MiB (obvykle mívají mezi 10 a 20 MiB). Nastavení hodnoty na 100 MiB by mělo být dostatečné. Pro neomezený limit v rámci testování je možno nastavit hodnotu na 0. V produkčním nasazení je ale vhodné hodnotu omezit.

**upload\_max\_filesize** Maximální velikost jednoho nahrávaného souboru (výchozí hodnota v XAMPP 2 MiB). V tomto případě by měla být nastavena stejná hodnota jako v parametru `post_max_size`, tedy 100 MiB.

**max\_file\_uploads** Maximální počet nahrávaných souborů v rámci jednoho požadavku (výchozí hodnota v XAMPP 20). Tuto hodnotu není potřeba navyšovat, pro tuto aplikaci můžeme snížit na 1.

Dále bude potřeba Node.js server, na kterém poběží WebSocket server.

## 4.2 Zkopírování souborů

Do kořenové složky dokumentů vašeho webového serveru zkopírujte obsah složky `php`.

## 4.3 Databáze

Vytvořte databázi s názvem „photogallery“ a spuštěním SQL instrukce ze souboru `photogallery.sql` vytvořte příslušné tabulky. V souboru `db.php`, umístěném v kořenové složce dokumentů webového serveru, nakonfigurujte přístup k vaší databázi. K zvýšení bezpečnosti je vhodné vytvořit nový uživatelský účet, který bude mít omezená práva jen k potřebné databázi a příkazům, tedy v rámci této aplikace k příkazům: `SELECT`, `INSERT`, `UPDATE` a `DELETE`. Pro připojení k databázi je použit PDO. Krom MySQL databáze (a databázích odvozených z MySQL, jako například MariaDB) lze tedy použít libovolnou databázi, která je PDO podporovaná, tedy: Cubrid, FreeTDS, Microsoft SQL Server, Sybase, Firebird, IBM DB2, IBM Informix Dynamic Server, Oracle Call Interface, ODBC v3, PostgreSQL, SQLite 3 a SQLite 2, Microsoft SQL Server, SQL Azure a 4D [21].

## 4.4 WebSocket

Dle návodu na [31] nainstalujte Node.js pro vaši platformu. Ve složce `server` jsou 2 soubory ke spuštění. Soubor `server.js` spustí WebSocket server

používající zabezpečený protokol wss. Pro jeho použití je potřeba nastavit v něm cestu k použitému certifikátu a jeho soukromému klíči. Druhý soubor **server-unsecured.js**, určený pro testovací účely, spustí WebSocket server na nezabezpečeném ws protokolu. Server spustíte příkazem:

```
node server.js
```

Samozřejmě je možné a vhodné nastavit spuštění serveru po spuštění systému, ideálně jej nastavit jako službu. Na platformě Windows toho můžeme docílit pomocí node-windows [32]. Na systémech s operačním systémem Linux je vhodné použít forever-service [33]. Příklad souboru **service.js**, který obsahuje registraci služby pomocí node-windows:

```
var Service = require('node-windows').Service;

var svc = new Service({
  name: 'WebSocket',
  description: 'WebSocket server.',
  script: 'D:\\DIP\\server\\server.js'
});

svc.on('install', function() {
  svc.start();
});

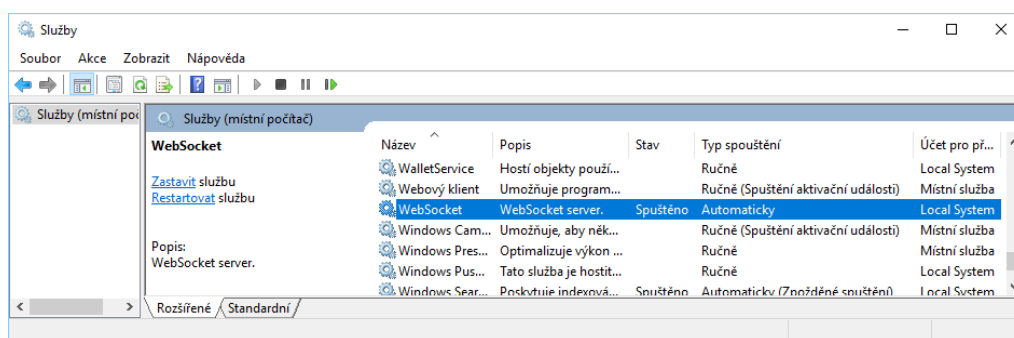
svc.install();
```

Služba se zaregistruje spuštěním **service.js** souboru v Node.js. Poté je možné službu spravovat přes klasické rozhraní pro správu služeb viz obrázek 4.1.

Aplikace v základním nastavení očekává WebSocket server na portu 8080 na stejné adrese jako běží PHP server. V případě, že váš WebSocket server poběží na jiné adrese nebo portu, tak změňte nastavení proměnných **server** a **port** na začátku souboru **ws.js** umístěném v kořenové složce dokumentů webového serveru.

#### 4. NÁVOD K INSTALACI

---



Obrázek 4.1: Rozhraní pro správu služeb na počítači s OS Windows 10 s nainstalovaným WebSocket serverem

---

# Testování

## 5.1 Funkčnost

Aplikaci jsem ze svého serveru zpřístupnil na internetu a požádal jsem několik přátel, z nichž někteří také často promítají, aby ji vyzkoušeli a dali mi zpětnou vazbu, jak se jim s aplikací pracovalo a aby nahlásili veškeré chyby a problémy, se kterými se při testování setkali, spolu s verzí jejich operačního systému a prohlížeče. Protože z důvodu použití certifikátu podepsaného sám sebou by testování bylo možné pouze v prohlížeči Chrome, tak jsem pouze pro účely testování použil verzi používající nezabezpečené protokoly HTTP a ws. Ostatní prohlížeče totiž blokují wss spojení za použití takového certifikátu, i když uživatel před tím souhlasil pokračovat na stránku se stejným certifikátem pro HTTPS spojení. Pro testování jsem vytvořil následující scénář:

1. Přejděte na stránky.
2. Zaregistrujte se v aplikaci.
3. Přihlaste se.
4. Vytvořte novou prezentaci.
5. Pojmenujte prezentaci.
6. Vložte do prezentace fotografie.
7. Změňte pořadí některých fotografií a u některých změňte popis nebo nadpis.

## 5. TESTOVÁNÍ

---

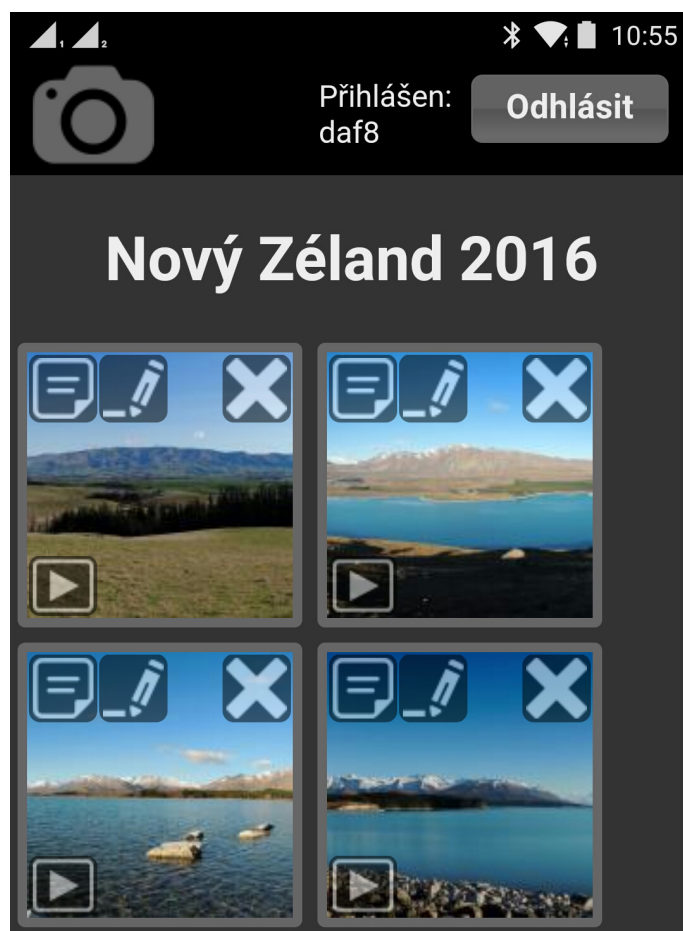
8. Přihlaste se na stránky z jiného počítače nebo mobilního zařízení, případně z jiného okna prohlížeče, a stiskněte tlačítko Promítání.
9. V prvním okně zahajte prezentaci tlačítkem Spustit prezentaci.
10. Projděte celou prezentaci. Při použití zobrazení pro přednášejícího změňte u některého snímku poznámku.
11. Po skončení prezentace zvolte jiný mód prezentace a vraťte se k bodu 9.
12. Dále zkoušejte aplikaci dle svého uvážení.

Z výsledků vyplynulo několik chyb, které bylo třeba opravit. Prvním problémem bylo, že vzhledem k zachování možnosti přesouvání pořadí jednotlivých snímků i na mobilních zařízeních občas nebylo možné spustit prezentaci z konkrétního snímku (často právě po přesunu některé fotografie). Jako řešení jsem ke každé fotografii přidal ikonku pro spuštění přehrávání z dané fotografie do levého dolního rohu náhledu (viz obrázek 5.1). Tato ikonka se zobrazuje pouze na zařízeních podporujících dotykové ovládání, na ostatních lze prezentaci spustit jednoduše kliknutím na náhled daného snímku.

Tím ale problémy spojené s dotykovým ovládáním neskončily. Po kliknutí na ikonku se totiž občas objevila pouze černá obrazovka a prezentace se nespustila. Nějakou dobu se mi nedařilo přijít na chybu, která to způsobuje. Při emulaci mobilního zařízení v nástroji pro vývojáře, obsaženého v prohlížeči Chrome na počítači, k tomuto problému nedocházelo. Při zkoumání problému jsem na [34] objevil metodu, kde lze použít stejné nástroje pro vývojáře známé z počítačů, i pro mobilní zařízení připojené k počítači USB kabelem. První chyba, kterou jsem tímto způsobem odhalil byla:

```
Uncaught DOMException: Failed to execute 'send' on  
'WebSocket': Still in CONNECTING state.
```

Funkci pro odesílání WebSocket zpráv jsem tedy upravil, aby před odesláním počkala vždy dokud spojení nebude ve stavu OPEN. To však problém s černou obrazovkou nevyřešilo. Z konzole jsem vysledoval, že událost `click` dané ikonky je z neznámého důvodu volána dvakrát. Protože je zobrazena pouze na dotykových zařízeních, tak jsem sledovanou událost zaměnil za `touchend` a ta už je volána pouze jednou. Po této záměně již problém s černou obrazovkou nebyl zaznamenán.



Obrázek 5.1: Zobrazení aplikace na mobilním zařízení (snímek obrazovky je zdola oříznut)

Ovládání prezentace v režimu pro přednášejícího na mobilních zařízeních funguje velmi spolehlivě včetně přechodu mezi snímky pomocí gest v případě automatického posunu fotografií. Při ručním posunu by totiž nebylo možné gesto rozeznat od posunu fotografie, a tak je v tomto režimu přechod mezi snímky možný pouze pomocí ikonek. V režimu zobrazení na celou obrazovku byly na dotykových zařízeních také přidány ikonky, protože při ručním posunu by bez připojení externí klávesnice nebylo možné přecházet mezi jednotlivými snímky. Pro pohodlnější ovládání aplikace z dotykových zařízeních byly také zvětšeny ovládací ikonky na těchto zařízeních. Na obrázku 5.1 je znázorněno zobrazení aplikace na mobilním zařízení se zvětšenými ikonkami.

Plynulost zobrazení automatického posunu je závislá pouze na výkonu počítače, na kterém je dané zobrazení spuštěno. Aplikace je díky uložení fotografií v různých rozlišeních dobře škálovatelná, a zatímco promítání na ploše s vertikálním rozlišením 600 řádků jde plynule i na starých počítačích, pro zcela plynulé zobrazení vyšších rozlišení je potřeba výkonnějších počítačů.

### 5.2 Validita kódu

Validita všech HTML kódů, včetně těch generovaných JavaScriptem, byla ověřena pomocí W3C validátoru. Krom několika málo upozornění byly nalezeny následující chyby:

`Error: Bad value toolbar for attribute role on element form.`

Tato chyba byla způsobena zavoláním jQuery UI funkce `controlgroup()` na element `<form>`. Problém jsem vyřešil změnou elementu `<form>` na `<div>`. Druhá chyba se vyskytla ve formulářích na vstupních elementech typu `text` a `password`.

`Error: Bad value password for attribute type on element input.`

Ta byla zapříčiněna atributem `role` nastaveným na hodnotu `button`, který se do kódu dostal při stylizování prvků pomocí jQuery UI funkce `button()`, tak aby zobrazení prvků v rámci GUI bylo jednotné. Problém byl vyřešen odebráním atributu `role` po zavolání funkce `button()` v souboru `jquery-ui-init.js`. Otestována byla i validita CSS souborů. Mnou vytvořené CSS soubory neobsahovaly žádné chyby.

### 5.3 Bezpečnost

Všechny vstupy od uživatele byly otestovány proti SQL injection a XSS útokům.

#### 5.3.1 SQL injection

Aplikace využívá k přístupu do databáze PDO. Díky použití tzv. připravených dotazů a nastavení atributu `ATTR_EMULATE_PREPARES` na hodnotu `false` je aplikace odolná vůči SQL injection.



### 5.3.2 Cross-site scripting

Aplikace je proti XSS chráněna použitím PHP funkce `htmlspecialchars()` s nastaveným příznakem `ENT_QUOTES` aplikované před zobrazením jakéhokoliv vstupu od uživatele a dat načtených z databáze.

## 5.4 Nielsenova heuristická analýza

Jak při návrhu, tak během realizace jsem bral v úvahu jednotlivé body heuristické analýzy, proto každý z bodů splňuje.

- 1. Viditelnost stavu systému** Většina aplikace funguje interaktivně bez potřeby čekat. Výjimkou je nahrávání fotografií do prezentace, které může trvat delší dobu. Proto je nahrávání řešeno pomocí XHR dotazů, aby se aktuální stav nahrávání mohl zobrazit pomocí jQuery UI prvku Progressbar.
- 2. Shoda mezi systémem a realitou** Akce po kliknutí na ikonku se shoduje s očekáváním uživatele. Aplikace v některých případech využívá jQuery UI ikonky a to konkrétně:
  - ui-icon-alert** K zobrazení vykřičníku u chybové hlášky a v dialogu, kde je potřeba potvrdit nevratnou operaci.
  - ui-icon-help** Ikonka otazníku sloužící k zobrazení nápovědy.
  - ui-icon-closethick** Ikonka křížku pro odstranění prezentace, zobrazená v seznamu prezentací.
  - ui-icon-info** Ikonka zobrazená spolu s informativní zprávou.
  - ui-icon-check** Ikonka zobrazená pro potvrzení nějaké operace (například úspěšná registrace).

Protože jQuery UI ikonky jsou dostupné pouze v rozlišení 16 x 16 pixelů, není jejich použití vhodné pro prvky, které jsou využity k ovládání aplikace na dotykových zařízeních. Pro tyto účely jsem použil ikonky z iconsDB.com [35]. Níže jsou uvedené ikonky použité v aplikaci a také jsou znázorněny na obrázku 5.2 zleva doprava.



Obrázek 5.2: Ukázka ikoněk použitých v aplikaci (barva ikoněk v aplikaci je přizpůsobena aktuálnímu použití, aby nenarušovaly grafický návrh a byly dobře čitelné)

**screenshot** Ikonka vyobrazující fotoaparát, která je použita jako logo celé aplikace.

**close window** Ikonka křížku, která je použita k ukončení prezentace ze zobrazení pro přednášejícího.

**previous** Pro přechod na předchozí snímek.

**next** Pro přechod na následující snímek.

**refresh 2** Pro vynulování stopek.

**pause** Pro pozastavení stopek.

**play** Pro spuštění stopek.

**delete 2** Pro odstranění fotografie z prezentace.

**note 2** K uložení poznámky k fotografii.

**edit 6** Pro změnu nadpisu u fotografie.

**play 4** Slouží ke spuštění prezentace z daného snímku.

**3. Minimální zodpovědnost (a stres)** Před odstraněním prezentace, je zobrazen dialog varující před provedením nevratné akce. V průběhu prezentace je samozřejmě možné se pohybovat oběma směry, tedy při nechtěném přesunu na další snímek se uživatel jednoduše vrátí k předchozímu snímku.

**4. Shoda s použitou platformou a obecnými standardy** Aplikace používá všeobecně známé komponenty jQuery UI, konkrétně:

**button** Pro tlačítka, ale v rámci jednotnosti grafického návrhu také pro vstupní pole, která jsou v CSS upraveny, aby se chovala dle očekávání uživatele.

**dialog** Slouží pro potvrzovací dialogy, chybové hlášky a zobrazení stavu nahrávání souborů.

**controlgroup** Seskupuje ovládací prvky prezentace (tlačítko pro spuštění a rozbalovací nabídku pro výběr módu) do jedné vizuální sady.

**progressbar** Slouží ke grafickému znázornění stavu nahrávání souborů.

**selectmenu** Je jQuery UI obdoba klasické rozbalovací nabídky známé z HTML.

**tooltip** Slouží k zobrazení nápovědy po najetí myši na daný prvek nebo dotyku prstu na ikonku nápovědy.

5. **Prevence chyb** Formuláře není možno odeslat dokud všechna vstupní pole nesplňují své regulární výrazy. Validace probíhá dle verze HTML5 jazyka HTML, takže zobrazení chyb je závislé na použitém prohlížeči. Mělo by tak reflektovat zvyklosti z použité platformy.
6. **Kouknu a vidím** Aplikace je navržena tak, aby byla pro uživatele co nejjednodušší a aby uživatel nemusel procházet zbytečnými kroky. Tlačítka, která nejsou momentálně potřeba, jako například tlačítko pro přechod na předchozí snímek v případě zobrazení prvního snímku prezentace, se nezobrazují. V zobrazení pro přenášejícího se zobrazuje pořadí aktuálního snímku prezentace, a to i graficky.
7. **Flexibilita a efektivita** Prezentaci je možné ovládat i pomocí klávesnice. Na dotykových zařízeních pak, v souladu s očekáváním uživatele, pomocí gest.
8. **Minimalita (Klapky na očích)** Aplikace zobrazuje jen nutné informace, tak aby práce v ní byla co nejrychlejší a přímočará.
9. **Smysluplné chybové hlášky** Chybové hlášky jsou v souladu se stavem aplikace a používají obecná slovní označení, tak aby uživatel snadno pochopil, k čemu došlo a mohl se z chyby poučit.
10. **Nápověda a dokumentace** Ovládací prvky používají atribut `title` pro krátkou nápovědu. V místech, kde by uživatel mohl znejistit, jsou malé ikonky se znakem otazníku, které po najetí myši zobrazí nápovědu pro daný úkon.

## 5. TESTOVÁNÍ

---



Obrázek 5.3: Testování v SAGELab na 4K displeji

### 5.5 SAGE

Laboratoř SAGELab je společným pracovištěm sdružení CESNET, Fakulty informačních technologií ČVUT a Fakulty elektrotechnické ČVUT. Účelem laboratoře je podpora výuky předmětů a výchovy odborníků v oblastech síťových technologií, multimédií a virtuální reality a vytvoření prostředí pro výzkum a vývoj nových síťových aplikací. Laboratoř je vybavena velkoplošným vizualizačním zařízením s vysokým rozlišením a vysokorychlostním připojením k počítačové síti. Vizualizační zařízení s rozlišením 8K+ (9600 x 4320 px) je tvořeno LCD displeji s tenkými rámečky a je řízeno softwarovým systémem SAGE a SAGE2, umožňujícím současné zobrazení výstupů řady aplikací, které mohou být provozovány na lokálních počítačích, na výpočetních prostředcích e-infrastruktury CESNET nebo na počítačích vzdálených spolupracujících institucí. Laboratoř je dále vybavena řadou dalších prostředků pro výuku, výzkum a vývoj v oblastech síťových a multimediálních technologií [3].



Obrázek 5.4: Testování v SAGElab na 8K+ stěně

Dne 12. dubna 2017 proběhlo první testování aplikace v SAGElab. Nejprve jsme aplikaci otestovali na 4K displeji, protože můj 10 let starý notebook by 4K rozlišení nezvládl, tak byl k testování použit zapůjčený výkonný MacBook. Ten si se 4K rozlišením poradil hravě a posuny fotografií vykresloval velmi plynule, fotografie z testování je na obrázku 5.3. Následně jsme se pokusili otestovat aplikaci přímo na 8K+ ploše. Bohužel náš první test ztroskotal na tom, že prohlížeč v SAGE2 nebyl schopen spolupracovat s certifikátem podepsaným sám sebou. Museli jsme tedy také použít testovací verzi s nezabezpečenými HTTP a ws protokoly.

Při druhém testu dne 19. dubna 2017 jsme byli již se zobrazením na SAGE úspěšní. Fotografie z testování na 8K+ stěně je na obrázku 5.4. Posun fotografií již však není tak hladce plynulý jako v případě zobrazení na 4K displeji, což je zřejmě způsobeno tím, že aplikace prohlížeče v SAGE2 již sama běží v prohlížeči. Synchronizace jednotlivých zobrazovacích sloupců není pro plynulost zobrazení zcela postačující, jedná se vlastně o 5 nezávislých instancí, kde při automatickém posunu začne každá z instancí s posunem fotografie po jejím načtení, k čemuž na každé instanci může dojít v rozdílný čas. Jedním z možných řešení by mohlo být naprogramovat přímo nativní aplikaci, která by běžela v prostředí SAGE2. Druhým jednodušším a ihned proveditelným řešením pro plynulejší posun fotografií by bylo promítat je na počítači s 4K výstupem, ten zaznamenávat a zobrazovat na 8K+ stěně. Sice se fotografie místo 8K+ rozlišení zobrazí divákům pouze ve 4K,

ale z běžné pozorovací vzdálenosti bude rozdíl téměř neznatelný a přínos plynulejšího zobrazení zcela jistě převáží tuto nevýhodu. Bohužel v době testování v SAGElab nebyl dostupný žádný takový modul k zachytávání HDMI výstupu ve 4K rozlišení.

## 5.6 Porovnání s jinými nástroji

### 5.6.1 Nativní prohlížeče fotografií

Programy pro prohlížení fotografií často mají také režim prezentace, kde je možné nastavit promítání fotografií buď automaticky nebo při stisku klávesy nebo tlačítka. Nenarazil jsem však zatím na prohlížeč, který by umožňoval oddělené zobrazení pro přednášejícího a měl možnost uložení poznámek přednášejícího či textových popisků k fotografiím. Přednášející pak musí mít poznámky v jiné formě například napsané na papíru. Panoramatické fotografie také většinou zobrazují pouze v úzkém proužku, zobrazení takové fotografie by tedy bylo nutné zvětšit a posouvat pomocí myši, což pro náš případ užití není uživatelsky přívětivé chování.

### 5.6.2 Video prezentace

Další způsob, který však mnoho přednášejících nepoužívá, je vytvoření videa z prezentovaných fotografií. Tento způsob není vhodný pro předvádění při živé prezentaci. Není zde možnost uložení poznámek přednášejícího ani zobrazení pro přednášejícího. Dále je zde problém se škálovatelností rozlišení videa oproti fotografiím a nutnost zastavovat video, aby nebylo napřed oproti mluvenému komentáři. Tento způsob se hodí spíše pro sdílení prezentace přes internet.

### 5.6.3 Microsoft PowerPoint

Dosud jsem pro své prezentace používal výhradně právě program Microsoft PowerPoint. První nevýhoda programu PowerPoint je zřejmá, nejedná se o svobodný software. K jeho legálnímu používání je potřeba licence, která není zrovna levná. Umí však vše, co potřebuji včetně zobrazení pro přednášejícího i s poznámkami. Protože však není pro tento účel primárně určen, není pro tento způsob použití moc uživatelsky přívětivý. Nastavení pro každý panoramatický snímek je velmi zdlouhavé a zabere i několik minut. Musí se nakreslit dráhy pohybu, nastavit časování, začátek a konec posunu a určit, zda se má posunout i zpět. Problém je i s kompatibilitou prezentací.

Výsledné prezentace nelze přehrát na počítačích, kde není PowerPoint nainstalován. Plynulost automatického posunu fotografií je na stejném zařízení při použití programu PowerPoint horší než za použití mé aplikace. Navíc si PowerPoint sám zmenšuje vložené fotografie a prezentaci není poté možné přehrát ve velmi vysokém rozlišení.

### 5.6.4 Scrollshow

Scrollshow je aplikace, která dle demonstračního videa [36] vypadá, že by mohla splňovat některé požadavky. Aplikace stojí \$3,99 a je dostupná pouze pro iPad. Jelikož iPad nevlastním, tak jsem tuto aplikaci a její možnosti nemohl osobně vyzkoušet. Výhoda mé webové aplikace ve srovnání se Scrollshow je v tomto případě jasná, moji aplikaci je možné použít na jakékoli platformě.

### 5.6.5 Prezi

Prezi je cloudový prezentační software a nástroj pro vytváření prezentací [37]. Produkt používá ZUI<sup>7</sup> a umožňuje uživatelům zobrazování a procházení informací v rámci 2.5D nebo paralaxy 3D prostoru. Prezi byla oficiálně založena v roce 2009, popularitu si získává hlavně v posledních letech. Prezi je alternativou ke klasickým prezentačním obsahující slidy. Pro náš případ užití se nejedná o příliš vhodný nástroj, nemá totiž žádnou podporu pro panoramatické snímky, byť efekt posunu by se zde dal vytvořit, byla by tvorba takových prezentací v Prezi značně nepohodlná. Nikde jsem nenašel ani možnost zobrazení pro přednášejícího.

---

<sup>7</sup>ZUI (Zooming user interface) je grafické prostředí, ve kterém může uživatel měnit měřítko zobrazení.





## Další vývoj

Současný stav aplikace zcela jistě není konečný. Už teď mám v hlavě mnoho vlastních nápadů na rozšíření a několik dalších, které vzešly z podnětů uživatelů, kteří aplikaci testovali. Mezi ty hlavní patří následující.

### 6.1 Video

Jedním z dalších nutných vylepšení je zapracování podpory pro přehrávání videí, protože přeci jen i když má fotografie své kouzlo, je lepší některé okamžiky z cest zachytit na video. Problémem může být velikost nahrávaných videí, která může velmi snadno překročit limit maximální velikosti souboru nastaveném na serveru. Řešení mě napadají dvě.

Prvním by mohlo být využití jiného serveru pro ukládání videí, typicky youtube.com. Video by tak přednášející nahrál na server [www.youtube.com](http://www.youtube.com) a do aplikace by pouze vložil odkaz na toto video, které by se při prezentaci načítalo přímo ze serverů youtube.com. Tato možnost má nevýhodu v nutnosti připojení k internetu s dostatečnou rychlostí pro přenos videa, což v dnešní době už není takový problém.

Druhou možností by bylo videa zpracovávat přímo na serveru aplikace stejně jako fotografie, pak by se musela přidat podpora nahrávání souborů po kouscích, aby se mohla nahrát i videa větší, než je maximální velikost pro jeden soubor nastavený v souboru `php.ini` na serveru.

### 6.2 Mapa

Další oblíbenou možností přednášejících je během prezentace posluchačům ukázat mapu své cesty. Což je možné i v aktuální verzi, ale přednášející

si musí mapu připravit v obrazovém formátu podporovaném aplikací a do prezentace ji nahrát jako klasickou fotografií. Zcela jistě by bylo vhodnější tuto možnost zapracovat přímo do aplikace, aby to bylo pro uživatele jednodušší a aby mohl mapu zobrazit kdykoliv během prezentace, kdy ji bude potřebovat.

### 6.3 Stream přednášek

Ve svém volném čase pracuji také jako dobrovolník v neziskové organizaci Jůzit z. s., kde každoročně na jaře a na podzim pořádáme série cestovatelských promítání. Já během těchto přednášek zajišťuji přímý přenos přes internet, aby přednášku mohli sledovat lidé, kteří se nemohli dostavit osobně, a aby byl k dispozici také záznam. Velkým problémem je datový tok takového přenosu. V našem infocentru, kde přednášky probíhají, máme k dispozici internetové připojení s rychlostí odesílání 2 Mbit/s, což je pro přenos Full HD videa nedostatečné. Jakožto neziskovka nemáme dostatečné prostředky pro profesionální techniku. Promítání tedy momentálně natáčím svým mobilem, kvalita takového přenosu je tedy dosti podprůměrná.

Vzhledem k tomu, že v aplikaci může být neomezený počet zobrazení v roli „slave“, tak už jen drobnou úpravou kódu a WebSocket serveru by bylo možné ke každé prezentaci vygenerovat odkaz na němž by mohli prezentaci sledovat i nepřihlášení uživatelé. Místo video streamu by se tedy z místa, kde promítání probíhá, přenášely pouze krátké zprávy k synchronizaci spolu s hlasem přednášejícího. Nároky na rychlost odesílání by tedy klesly na méně než desetinu současného stavu, přičemž fotografie by se divákům ukazovaly v mnohonásobně vyšší kvalitě v porovnání se současným video streamem.

Nikterak složitý by nebyl ani záznam takovýchto přednášek, veškeré předávané instrukce by se zaznamenávaly spolu s časovou značkou, udávající čas dané instrukce od začátku prezentace, do databáze a ukládal by se záznam mluveného projevu.

### 6.4 Efekty

Aplikaci je možné rozšířit o různé další přechodové efekty mezi jednotlivými snímky či o efekty používané během prohlížení jednotlivého snímku, jako například o Ken Burnsův efekt, který slouží ke zdramatizování záběru použitím efektu pomalého přibližování a posunu fotografie [38].

## 6.5 Přihlášení

V dnešní době také stojí za zvážení možnost přihlášení do aplikace pomocí jiné autority, kde již je uživatel zaregistrován. Jednou z možností je přihlášení pomocí populární sociální sítě Facebook. Další možností je pro přihlášení využití účtu Google. Samozřejmě je ale nutné v aplikaci nechat stávající možnost registrace pro uživatele, kteří buď nepoužívají Facebook či Google, nebo je prostě nechtějí pro přihlášení používat.

## 6.6 Aplikace

Aplikaci by bylo také možné rozšířit o nativní aplikace pro různé platformy. Například, jak již bylo zmíněno v předchozí kapitole, o aplikaci určenou přímo pro SAGE2 pro zajištění vyšší plynulosti zobrazení na distribuovaných zobrazovacích zařízeních. Užitečná by byla také aplikace pro Google Chromecast. Ze zařízení, kde by byla prezentace spuštěna v roli „master“ by se na Chromecast vysílalo zobrazení v roli „slave“. To by jistě oceňili někteří přednášející. Například pro mé každoroční promítání v objektu ČVUT v Temešváru u Písku by stačilo vzít s sebou projektor s připojeným Google Chromecast zařízením a mobil nebo tablet, ze kterého by se prezentace ovládala. Žádný notebook ani velký počítač by již k prezentaci nebyl potřeba.



---

## Závěr

Tato diplomová práce shrnuje možnosti vzdálené synchronizace více oken prohlížeče a v rámci aplikace implementuje jedno z možných řešení. V dnešní době při tvorbě webových aplikací fungujících v reálném čase již nemá cenu uvažovat o jiných technologiích než Server-sent events pro aplikace, kde je primárně potřeba přenášet data ze serveru ke klientovi nebo WebSocket pro aplikace požadující komunikaci v obou směrech.

Aplikace pro prezentaci panoramatických snímků je v plně funkční stavu a splňuje stanovené požadavky, využívá moderní technologie specifikace HTML5 a WebSocket, které jsou však již dnes dostupné pro téměř 94 % uživatelů internetu [39]. Aplikace poskytuje značné zjednodušení práce při vytváření prezentací fotografií z cest, bohužel však ještě neumí sama vybírat povedené fotografie pro promítání divákům, to stále ještě musí přednášející udělat sám. Už teď se nemohu dočkat až aplikaci poprvé vyzkouším v praxi, to bude již brzy v rámci mého cestovatelského promítání fotografií z mé loňské cesty po Nového Zélandu, které proběhne během víkendové akce pořádané turistickým oddílem VSK ČVUT Praha v Temešváru u Písku.

Odevzdáním diplomové práce však moje práce na této aplikaci nekončí, protože ji sám budu často používat, tak ji budu vyvíjet i nadále. Práce na aplikaci mě bavila. Získal jsem mnoho nových zkušeností a zároveň jsem v praxi využil znalosti nabyté z mnoha předmětů, kterými jsem během svého studia na ČVUT prošel.



---

## Literatura

- [1] Příspěvatelé Wikipedie: Panoramatická fotografie — Wikipedie: Otevřená encyklopedie. [online], 2016, [citováno 21.4.2017]. Dostupné z: [https://cs.wikipedia.org/w/index.php?title=Panoramatick%C3%A1\\_fotografie&oldid=14338630](https://cs.wikipedia.org/w/index.php?title=Panoramatick%C3%A1_fotografie&oldid=14338630)
- [2] Microsoft: Image Composite Editor - Microsoft Research. [online], [citováno 22.4.2017]. Dostupné z: <https://www.microsoft.com/en-us/research/product/computational-photography-applications/image-composite-editor/>
- [3] SAGElab | Síťová multimediální laboratoř. [online], [citováno 8.4.2017]. Dostupné z: <https://sagelab.cesnet.cz/cz/>
- [4] Vitvar, T.: Protocols for the Realtime Web. [online], [citováno 8.4.2017]. Dostupné z: <http://w20.vitvar.com/pdf/lecture8-1p.pdf>
- [5] Oracle: Lesson: Java Applets (The Java™ Tutorials > Deployment). [online], [citováno 8.4.2017]. Dostupné z: <http://docs.oracle.com/javase/tutorial/deployment/applet/index.html>
- [6] Oracle: Java and Google Chrome Browser. [online], [citováno 8.4.2017]. Dostupné z: <https://java.com/en/download/faq/chrome.xml>
- [7] Topic, D.: Migrating from Java Applets to plugin-free Java technologies. [online], [citováno 8.4.2017]. Dostupné z: <http://www.oracle.com/technetwork/java/javase/migratingfromapplets-2872444.pdf>

- [8] Waldron, R.: The Flash history > News > Flash Magazine. [online], [citováno 8. 4. 2017]. Dostupné z: [http://www.flashmagazine.com/news/detail/the\\_flash\\_history/](http://www.flashmagazine.com/news/detail/the_flash_history/)
- [9] APE (Ajax Push Engine) :: Comet server :: Real time data streaming. [online], [citováno 10. 4. 2017]. Dostupné z: <http://ape-project.org/>
- [10] W3C: HTML5 Server-Sent Events. [online], [citováno 10. 4. 2017]. Dostupné z: [https://www.w3schools.com/html/html5\\_serversentevents.asp](https://www.w3schools.com/html/html5_serversentevents.asp)
- [11] Ballou, C.: Creating Realtime Applications with PHP and Websockets. [online], [citováno 11. 4. 2017]. Dostupné z: <https://www.slideshare.net/CoreyBallou/creating-realtime-applications-with-php-and-websockets>
- [12] SignalR: ASP.NET SignalR. [online], [citováno 10. 4. 2017]. Dostupné z: <http://signalr.net/>
- [13] W3C: The WebSocket API. [online], [citováno 10. 4. 2017]. Dostupné z: <https://www.w3.org/TR/2011/WD-websockets-20110419/>
- [14] Balsamiq Studios, LLC: Balsamiq Mockups | Balsamiq. [online], [citováno 11. 4. 2017]. Dostupné z: <https://balsamiq.com/products/mockups/>
- [15] Kümmel, R.: Cross Site Request Forgery. *Hakin9*, 2 2008: s. 2–11, [citováno 4. 5. 2017]. Dostupné z: <https://www.soom.cz/data/csrf.pdf>
- [16] Teixeir, P.: *Professional Node.js: Building Javascript based scalable software*. John Wiley & Sons, 2012, ISBN 978-1-118-18546-9.
- [17] Internet Security Research Group: Let's Encrypt - Free SSL/TLS Certificates. [online], [citováno 30. 4. 2017]. Dostupné z: <https://letsencrypt.org/>
- [18] Ellingwood, J.: How To Create a Self-Signed SSL Certificate for Apache in Ubuntu 16.04 | DigitalOcean. [online], [citováno 30. 4. 2017]. Dostupné z: <https://www.digitalocean.com/community/tutorials/how-to-create-a-self-signed-ssl-certificate-for-apache-in-ubuntu-16-04>
- [19] W3C: HTML5. [online], [citováno 25. 4. 2017]. Dostupné z: <https://www.w3.org/TR/html5/>



- 
- [20] W3C: CSS Tutorial. [online], [citováno 29. 4. 2017]. Dostupné z: <https://www.w3schools.com/css/>
  - [21] The PHP Group: PHP: PDO Drivers - Manual. [online], [citováno 11. 4. 2017]. Dostupné z: <http://php.net/manual/en/pdo.drivers.php>
  - [22] Wikipedia contributors: JavaScript — Wikipedia, The Free Encyclopedia. [online], 2017, [citováno 22. 4. 2017]. Dostupné z: <https://en.wikipedia.org/w/index.php?title=JavaScript&oldid=776620899>
  - [23] Chaffer, J.; Swedberg, K.: *Learning jQuery, Fourth Edition*. Packt Publishing Ltd., 2013, ISBN 978-1-78216-314-5.
  - [24] Libby, A.; Wellman, D.: *jQuery UI 1.10: The User Interface Library for jQuery*. Packt Publishing Ltd., 2013, ISBN 978-1-78216-220-9.
  - [25] Furfero, D.: jQuery UI Touch Punch - Touch Event Support for jQuery UI. [online], [citováno 23. 4. 2017]. Dostupné z: <http://touchpunch.furf.com/>
  - [26] Tangelder, J.: Hammer.JS - Hammer.js. [online], [citováno 23. 4. 2017]. Dostupné z: <http://hammerjs.github.io/>
  - [27] Tibbett, R.: GitHub - richtr/NoSleep.js: Prevent display sleep and enable wake lock in any Android or iOS web browser. [online], [citováno 23. 4. 2017]. Dostupné z: <https://github.com/richtr/NoSleep.js>
  - [28] GitHub - websockets/ws: Simple to use, blazing fast and thoroughly tested WebSocket client and server for Node.js. [online], [citováno 22. 4. 2017]. Dostupné z: <https://github.com/websockets/ws>
  - [29] Caceres, M.; Christiansen, K. R.; Lamouri, M.; aj.: Web App Manifest. [online], [citováno 21. 4. 2017]. Dostupné z: <https://w3c.github.io/manifest/>
  - [30] Apache Friends: XAMPP Installers and Downloads for Apache Friends. [online], [citováno 21. 4. 2017]. Dostupné z: <https://www.apachefriends.org/index.html>
  - [31] Node.js Foundation: Download | Node.js. [online], [citováno 20. 4. 2017]. Dostupné z: <https://nodejs.org/en/download/>

- [32] Butler, C.: GitHub - coreybutler/node-windows: Windows support for Node.JS scripts (daemons, eventlog, UAC, etc). [online], [citováno 20.4.2017]. Dostupné z: <https://github.com/coreybutler/node-windows>
- [33] Zpty, Inc.: GitHub - zpty/forever-service: Provision node script as a service via forever, allowing it to automatically start on boot, working across various Linux distros and OS. [online], [citováno 20.4.2017]. Dostupné z: <https://github.com/zpty/forever-service>
- [34] Heller, A.: Vzdálené ladění měření na mobilním webu | Medio Blog. [online], [citováno 17.4.2017]. Dostupné z: <https://blog.medio.cz/vzdalene-ladeni-mereni-na-mobilnim-webu>
- [35] Icons DB - free custom icons. [online], [citováno 27.4.2017]. Dostupné z: <http://www.iconsdb.com/>
- [36] Scrollshow, the first panoramic presentation editor, for iPad - YouTube. [online], [citováno 21.4.2017]. Dostupné z: <https://www.youtube.com/watch?v=mrU8vuy7KKM>
- [37] Prezi Inc.: About Prezi | Company Information and History | Prezi. [online], [citováno 21.4.2017]. Dostupné z: <https://prezi.com/about/>
- [38] Wikipedia contributors: Ken Burns effect — Wikipedia, The Free Encyclopedia. [online], 2017, [citováno 21.4.2017]. Dostupné z: [https://en.wikipedia.org/w/index.php?title=Ken\\_Burns\\_effect&oldid=766809098](https://en.wikipedia.org/w/index.php?title=Ken_Burns_effect&oldid=766809098)
- [39] Deveria, A.: Can I use... Support tables for HTML5, CSS3, etc. [online], [citováno 10.4.2017]. Dostupné z: <http://caniuse.com/>

## Seznam použitých zkratk

**AJAX** Asynchronous JavaScript and XML

**APE** Ajax Push Engine

**API** Application Programming Interface

**CC** Creative Cloud

**CDN** Content Delivery Network

**CERN** Conseil Européen pour la recherche nucléaire (Evropská organizace pro jaderný výzkum)

**CESNET** Czech Education and Scientific NETwork

**CS** Creative Suite

**CSRF** Cross-site request forgery

**CSS** Cascading Style Sheets

**ČR** Česká republika

**ČVUT** České vysoké učení technické v Praze

**DNS** Domain Name System

**DOCTYPE** Document Type

**DOM** Document Object Model

**DTD** Document Type Definition

## A. SEZNAM POUŽITÝCH ZKRATEK

---

**DV** Domain Validation

**DVD** Digital Versatile Disc

**ECMA** European Computer Manufacturers Association

**EER** Enhanced entity–relationship

**EXIF** Exchangeable image file format

**GPL** General Public License

**GUI** Graphical user interface

**HD** High Definition

**HDMI** High-Definition Multimedia Interface

**HTML** Hypertext Markup Language

**HTTP** Hypertext Transfer Protocol

**HTTPS** Hypertext Transfer Protocol Secure

**IBM** International Business Machines Corporation

**ISO** International Organization for Standardization

**JIT** Just-in-time

**JRE** Java Runtime Environment

**JSON** JavaScript Object Notatio

**JVM** Java Virtual Machine

**LAMP** Linux, Apache, MySQL, and PHP

**LCD** Liquid crystal display

**MIT** Massachusetts Institute of Technology

**MI-MAI** Multimedia a internet

**MI-W20** Web 2.0

**ODBC** Open Database Connectivity

**OS** Operační systém

---

**PDF** Portable Document Format

**PDO** PHP Data Objects

**PHP** PHP: Hypertext Preprocessor (rekurzivní zkratka)

**RIA** Rich Internet Application

**RTMP** Real-Time Messaging Protocol

**SAGE** Scalable Amplified Group Environment

**SQL** Structured Query Language

**SSL** Secure Sockets Layer

**TCP** Transmission Control Protocol

**TLS** Transport Layer Security

**UI** User Interface

**URI** Uniform Resource Identifier

**USB** Universal Serial Bus

**VSK** Vysokoškolský sportovní klub

**WWW** World Wide Web

**W3C** World Wide Web Consortium

**XAMPP** Cross-platform Apache HTTP Server MySQL PHP Perl

**XHR** XMLHttpRequest

**XHTML** Extensible Hypertext Markup Language

**XML** Extensible Markup Language

**XSS** Cross-site scripting

**X36WWW** Internet a WWW

**ZUI** Zooming user interface

**z. s.** zapsaný spolek



## Obsah přiloženého DVD

readme.txt .....	stručný popis obsahu DVD
src	
├─ php .....	zdrojové kódy aplikace pro webový server
├─ server .....	zdrojové kódy WebSocket serveru
├─ thesis .....	zdrojová forma práce ve formátu $\text{\LaTeX}$
├─ photogallery.sql .....	instrukce pro vytvoření tabulek v databázi
└─ DP_Krupička_David_2017.pdf .....	text práce ve formátu PDF