

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Nástroj pro zrcadlení obrazovek



2017

Daniel Hájek

Vedoucí práce: Mgr. Martin
Trnečka, Ph.D.

Studijní obor: Informatika, prezenční
forma

Bibliografické údaje

Autor: Daniel Hájek
Název práce: Nástroj pro zrcadlení obrazovek
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2017
Studijní obor: Informatika, prezenční forma
Vedoucí práce: Mgr. Martin Trnečka, Ph.D.
Počet stran: 31
Přílohy: 1 CD
Jazyk práce: český

Bibliographic info

Author: Daniel Hájek
Title: Tool for screen mirroring
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2017
Study field: Computer Science, full-time form
Supervisor: Mgr. Martin Trnečka, Ph.D.
Page count: 31
Supplements: 1 CD
Thesis language: Czech

Anotace

Práce pojednává o nástroji umožňujícím sdílení obrazu pracovní plochy mezi počítači v síti. Popisuje řešení pořizování snímků pracovní plochy, jejich přenos a zobrazení koncovému klientovi, který kromě snímku samotného nemá k datům přístup. Práce obsahuje také uživatelský manuál.

Synopsis

The thesis is focused on the tool for sharing desktop images among computers on the network. It describes solution for capturing, transferring and displaying desktop images to the end client who, except for the image itself, has no access to data. The thesis includes a user manual as well.

Klíčová slova: zrcadlení obrazovek; mirroring; sdílení obrazu mezi počítači

Keywords: screen mirroring; mirroring; sharing desktop images among computers

Rád bych poděkoval Mgr. Martinu Trnečkovi, Ph.D. za vedení mé bakalářské práce a cenné rady.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce

podpis autora

Obsah

1	Úvod	7
1.1	Volba názvu aplikace	7
1.2	Motivace	7
2	Dostupné možnosti nabízející zrcadlení	8
2.1	Skype	8
2.2	WiDi	8
2.3	Google Chromecast	8
2.4	DLNA	9
2.5	TeamViewer	9
2.6	Porovnání s aplikací Mirroring	9
3	Implementace aplikace	10
3.1	Java aplikace Mirroring	10
3.1.1	Snímání obrazu	10
3.1.1.1	Třída CapturingThread	11
3.1.1.2	Třída AvgDelayCalculator	11
3.1.1.3	Třída ScreenCapture	11
3.1.2	Kompresce a příprava obrazu pro přenos	12
3.1.2.1	Třída EncodingThread	12
3.1.2.2	Třída JPEGEncoder	13
3.1.3	Síťová komunikace a přenos	13
3.1.3.1	Třída StreamingAppServer	13
3.1.3.2	Třída StreamingThread	13
3.1.3.3	Třída StreamingHttpServer	14
3.1.3.4	Třída ConnectionManager	14
3.1.4	Zobrazení přijatého snímku	14
3.2	Java aplikace MirroringServer	15
3.2.1	Udržování informací o aktivních spojení	15
3.2.1.1	Třída AppServer	16
3.2.1.2	Třída AppServerThread	16
3.2.1.3	Třída AppSessionManager	16
3.2.2	Příjem a příprava snímků pro webovou službu	16
3.2.2.1	Třída HttpServer	17
3.2.2.2	Třída HttpSessionManager	17
3.2.2.3	Třída HttpServerThread	17
3.3	Webový server	17
3.3.1	Zobrazení snímku pomocí webového prohlížeče	18
4	Spuštění aplikace Mirroring	19
4.1	JRE 8	19
4.2	Spuštění aplikace Mirroring	19
4.3	Spuštění aplikace MirroringServer	19

4.4	Webový server	19
4.4.1	Příprava chodu v lokální síti	19
4.4.2	Příprava chodu v síti internet	20
4.4.2.1	Poznámky	20
5	Technologie použité při vývoji	21
5.1	HTML	21
5.2	CSS	21
5.3	PHP	21
5.4	JavaScript	22
5.4.1	jQuery	22
5.5	Java	22
6	Uživatelský manuál aplikace	23
6.1	Funkce pro sdílení mezi aplikacemi	24
6.1.1	Pravý panel	24
6.1.2	Levý panel	24
6.2	Funkce pro sdílení do webového prohlížeče	25
6.3	Funkce pro příjem vysílaného obrazu	25
7	Uživatelský manuál serveru	26
7.1	Nastavení konfiguračního souboru	26
7.2	Spuštění a obsluha serveru	26
	Závěr	28
	Conclusions	29
	A Obsah přiloženého CD	30
	Literatura	31

Seznam obrázků

1	Logo aplikace Mirroring	7
2	Struktura desktopové aplikace	10
3	Graf ilustrující komunikaci instancí při snímání	12
4	Struktura a jednotlivé třídy serverové aplikace	15
5	Adresářová struktura webové aplikace	17
6	Úvodní scéna	23
7	Grafické rozhraní pro sdílení mezi aplikacemi	25
8	Ukázka konzolové aplikace MirroringServer	27

1 Úvod

Nástroj pro zrcadlení obrazovek je aplikace, která poskytuje funkci sdílení obrazu pracovní plochy mezi počítači v síti. Zachycuje tedy aktuální stav obrazovky v podobě snímků, které následně komprimuje a rozesílá mezi naslouchající klienty. Uživatelé uvidí pouze obraz, ale není jim umožněno systémem vysílající obraz ovládat ani do něj jakkoli zasahovat.

Zobrazení snímků sdílené plochy je možné dvěma způsoby, od kterých se následně odvíjí i frekvence zasílání obrazů klientům a nabízené doplňkové možnosti. První možností je zasílání obrazů mezi aplikacemi, kdy je umožněno chránit přístup do jednotlivých sezení heslem. Frekvence snímků je u této možnosti vyšší a obraz tedy plynulejší. Tato možnost dále podporuje nastavení kvality zasílaného obrazu. Druhá možnost nabízí službu sdílející snímky přímo do webového prohlížeče. Toto rozšíření umožňuje multiplatformní přístup z libovolného zařízení, které disponuje připojením do sítě a internetovým prohlížečem. Z důvodu velikosti přenosu obrazů zde byla využita nižší frekvence zasílání a to jednoho snímku za sekundu. Výhodou toho rozšíření je tedy přenos obrazu do zařízení, která nemají k dispozici aplikaci samotnou.

1.1 Volba názvu aplikace

Vzhledem k největší rozšířenosti byla jako výchozí jazyk aplikace zvolena angličtina. Z toho důvodu aplikace nese název *Mirroring* (logo aplikace viz obr. 1), jenž je právě překlad českého slova *zrcadlení* do anglického jazyka, které bylo intuitivně zvoleno z názvu této práce (*Tool for screen mirroring*).



Obrázek 1: Logo aplikace Mirroring

1.2 Motivace

Motivací této práce bylo vytvořit aplikaci, která umožní snadné prezentování fotografií, textu nebo určitého postupu. Mnohdy pro prezentaci musíme přenést data na jiné zařízení, které nám zprostředkuje připojení například s dataprojekto-rem, z důvodů znemožnění přímého přístupu nebo absence konektorů u laptopů. To může znamenat bezpečnostní riziko, jak z pohledu krádeže dat, tak možnosti nakopírování nežádoucího softwaru na datové medium a následné rozšíření při opětovném použití. Pokud jsou zařízení připojená do sítě, můžeme pomocí apli-

kace *Mirroring* odstranit problémy s redukcemi konektorů, překážejícími kabely a riziky spojenými s datovými médii.

2 Dostupné možnosti nabízející zrcadlení

Aplikace, které zprostředkovávají funkci sdílení obrazovky již existují, ale možnost je ovšem brána často jako doplněk jiné funkcionality nebo je potřeba zakoupení speciálního hardwaru, který přenos obrazu umožní. Zde si ukážeme několik aplikací, které zrcadlení obrazovky umožňují a porovnáme je s aplikací *Mirroring*.

2.1 Skype

Skype je aplikace od firmy *Microsoft*. Hlavními funkcemi je možnost telefonování v síti internet (*VoIP*¹) a *instant messaging*² s možností zasílání souborů, hlasových zpráv a dalších, které poskytuje zdarma. Telefonování a zasílání zpráv mimo síť internet je zpoplatněno.

Pro možnost sdílení obrazovek je nutné mít na obou zařízeních aplikaci *Skype* a nutnou podmínkou pro spojení je mít založené rozdílné účty. Pokud bychom chtěli aplikaci použít pouze pro přenos snímků obrazovky za účelem prezentace, je toto řešení značně složité a nepraktické.

2.2 WiDi

Technologie od firmy *Intel*, která je součástí pouze vybraných typů procesorů. Umožňuje duplikovat nebo rozšířit obrazovku v rozlišení až 1080p. Nevýhodou je zde nutnost vlastnit procesor podporující *WiDi* a zakoupení speciálního hardwaru, který připojení zprostředkuje. Výhodou je možnost, že není požadováno připojení obou zařízení do sítě, spojení je řešeno právě přes zmiňovaný hardware [3].

2.3 Google Chromecast

Zařízení pro bezdrátové streamování³ médií do zařízení, která disponují *HDMI* portem. Nabízí možnost streamování videa, hudby, her, fotografií a dalších [4]. Opět nutnost speciálního hardwaru. Podobné technologii *WiDi*.

¹Voice over Internet Protocol (VoIP) je souhrn řídicích pravidel přenos hlasu v síti internet.

²Komunikace v reálném čase, při kterém si uživatelé navzájem po síti posílají textové zprávy [1].

³Technologie používaná k poskytování obsahu, obvykle audio a video, do počítačů a mobilních zařízení [2]

2.4 DLNA

Digital Living Network Alliance (DLNA) je společnost definující standardy, které dovolují sdílet fotografie, video, a muziku mezi zařízeními podporujícími *DLNA* a jsou ve stejné síti. Tyto standardy jsou dnes hojně podporované, kromě produktů firmy *Apple*. například i operačním systémem *Windows* [5]. Je tedy možné sdílet jednotlivé soubory, ale ne právě prováděný postup na daném počítači.

2.5 TeamViewer

Aplikace, která umožňuje vzdálený přístup k zařízení. Hlavní funkcí je tedy získání plné kontroly například nad počítačem nebo serverem nacházejícím se v síti internet.

Zrcadlení obrazovek zde tedy tvoří nedílnou součást aplikace, která je umožňuje zařízení pohodlně ovládat. Pokud bychom chtěli této vlastnosti sdílení obrazu využít pro prezentaci, vystavujeme se bezpečnostnímu riziku. Důvodem je i fakt, že kromě samotného vzdáleného ovládání *TeamViewer* podporuje mezi spojenými klienty i přenos souborů.

2.6 Porovnání s aplikací Mirroring

Mirroring je aplikace vysílající obraz pracovní plochy do jiných zařízení připojených do sítě, která disponují aplikací *Mirroring*, nebo webovým prohlížečem.

Mirroring se soustředí pouze na přenos obrazu a vytváří tak snadno ovladatelné prostředí, kde se uživatel neztratí v nepřehledném množství funkcí. Nevyžaduje, kromě dnes hojně dostupného připojení do sítě, koupi speciálního hardwaru. Představuje tedy nenákladnou možnost prezentování fotografií, textu a postupu do několika zařízení najednou. Této vlastnosti můžeme využít například při absenci dataprojektoru⁴, nebo sdílet obrazovku klientovi, který právě zobrazení pomocí jiného zařízení (například zmiňovaného dataprojektoru) zprostředkuje.

⁴Za předpokladu, že máme zařízení, na která je možnost vysílat (například mobilní telefon s webovým prohlížečem).

3 Implementace aplikace

Celkovou funkcionalitu zajišťují tři prvky. Jedná se o Java programy *Mirroring*, *MirroringServer* a webový server. Tyto tři celky svojí spoluprací tvoří výslednou aplikaci *Mirroring*, jejíž struktura je znázorněna na obrázku 2.

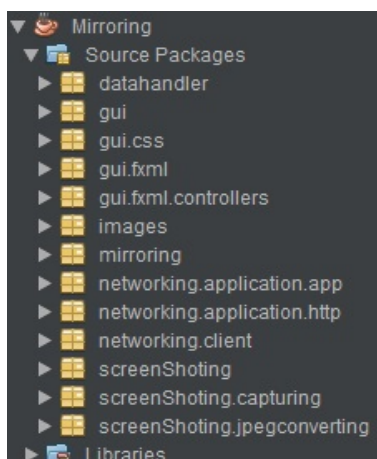
Tato sekce se zabývá popisem způsobu implementace jednotlivých celků.

3.1 Java aplikace Mirroring

Mirroring je desktopová aplikace s grafickým rozhraním.

Hlavní funkce probírané v této kapitole:

- Snímání obrazu.
- Komprese a příprava obrazu pro přenos.
- Síťová komunikace a přenos.
- Zobrazení přijatého snímku.



Obrázek 2: Struktura desktopové aplikace

3.1.1 Snímání obrazu

Pro snímání obrazu je využívána třída `Robot` obsahující metodu `createScreenCapture`, která při nastavení správné velikosti, vrací snímek celé obrazovky. Tato třída má ovšem zásadní nedostatek pro použití v této aplikaci. Třída `Robot` zvládá pořizovat snímky pouze s frekvencí okolo 12 fps⁵. Lidské oko pro iluzi plynulého pohybu potřebuje minimálně 25fps. Problém řeší Java balíček `capturing`.

⁵Frames per second - počet snímků za vteřinu.

Obsah tříd v balíčku je následující:

- `CapturingThread`
- `ScreenCapture`

3.1.1.1 Třída `CapturingThread`

Třída je implementována tak, že její instance po spuštění vyčkává na signál. Dostane-li signál, uspí se na nastavenou dobu. Tato doba odpovídá hodnotě podílu jedné sekundy a nastaveného počtu snímků za vteřinu. Výsledek je čas spánku v milisekundách. Po probuzení je signál předán následníkovi (vysvětleno ve třídě `ScreenCapture` níže) a instance začne pořizovat snímek obrazovky. K pořízení snímku využívá instanci třídy `Robot`. Vytvořený snímek je dále škálován na menší velikost podle nastaveného parametru.

Škálování je časově nenáročná metoda, která při použití hodnoty mezi 0 a 1 zmenší obraz a tím i jeho velikost, kterou zabírá v paměti. To je výhodné pro přenos snímku po síti. Použití ovšem ovlivňuje kvalitu výsledného obrazu.

3.1.1.2 Třída `AvgDelayCalculator`

Při spuštění aplikace se spustí také instance třídy `AvgDelayCalculator` z balíčku `screenShoting`. Jejím úkolem je spustit jednu instanci tříd `CapturingThread` a `EncodingThread` (třída je popsána níže v části *Kompresa a příprava obrazu pro přenos*). Cyklicky poté vyvolává implementovanou funkcionalitu a měří čas potřebný pro dokončení. Ze součtu časů, počtu volání a informaci o požadovaném počtu snímků za jednu sekundu odvodí počet potřebných instancí pro plynulý chod. K výsledku je přičtena jednička, která představuje jednu rezervní instanci, kdyby nastalo zpoždění. Vzorec pro výpočet množství instancí:

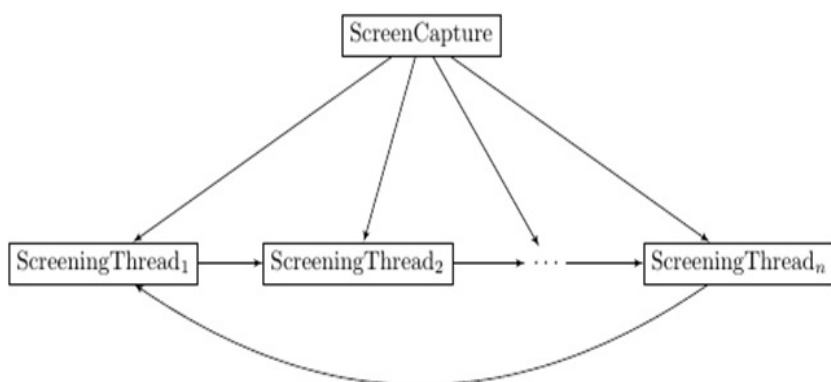
$$\left\lceil \frac{\text{Součet časů} / \text{Počet volání}}{1000 / \text{Počet snímku za sekundu}} \right\rceil + 1$$

3.1.1.3 Třída `ScreenCapture`

Instance této třídy paralelně spustí vypočítané množství instancí třídy `CapturingThread`. Tyto instance pomyslně zapojí do kruhu tak, že každé z instancí nastaví následníka, přičemž poslední instance má jako následníka nastavenou instanci první (grafické znázornění viz obr. 3). Tato třída následně pomocí metody `getScreenShot` zjišťuje, která z instancí v kruhu disponuje pořízeným snímkem a daný snímek vrací.

Díky posílání signálů mezi instancemi třídy `CapturingThread` je možné časově synchronizovat pořizování snímků. Každá z instancí tedy snímá obraz s daným časovým odstupem od předchozí instance. Tímto způsobem eliminujeme problém s nízkým počtem fps u třídy `Robot` tak, že paralelně spustíme synchronizované instance třídy `CapturingThread`. Výsledkem je pořizování snímků s frekvencí 30 fps.

Podrobněji, máme-li n instancí třídy `CapturingThread`, každé instanci nastavíme požadované zpoždění a další volnou instanci, aby zapojení tvořilo zmiňovaný pomyslný kruh. Každá z instancí po spuštění vyčkává. Jeli požadováno snímání, instance třídy `ScreenCapture` zašle signál instanci v kruhu s indexem 0 (na volbě indexu nezáleží). Instance započne odpočítávání nastaveného zpoždění. Po vypršení limitu zašle signál následující instanci a spustí snímání. Odpočet před samotným snímáním eliminuje nutnost brát v potaz proměnlivé doby na pořízení snímku. Je tedy zajištěno, že každá instance započne snímání po dané uplynulé době.



Obrázek 3: Graf ilustrující komunikaci instancí při snímání

3.1.2 Komprese a příprava obrazu pro přenos

Pro kompresi pořízených snímků je využito knihovny *Java Advanced Imaging (JAI)* od firmy *Oracle*. Knihovna obsahuje třídu `JPEGImageEncoder`, která pomocí metody `encode` pořízený obraz ztrátově zkomprimuje do formátu *JPEG* a převede jej na pole bajtů. Převod následně využijeme při odesílání snímku. Komprimace je ovšem časově náročná, a pro dodržení frekvence 30 fps je potřeba opět paralelního komprimování více instancemi, jako při problému snímání obrazu. Ovšem zde již není potřeba synchronizace signálem.

Tuto funkčnost nabízí Java balíček `jpegconverting` obsahující třídy:

- `EncodingThread`
- `JPEGEncoder`

3.1.2.1 Třída `EncodingThread`

Instance vyčkává na příchod snímku, který následně zkomprimuje pomocí metody `encode` instance třídy `JPEGImageEncoder` z knihovny *Java Advanced Imaging*. Zkomprimovaná data v podobě pole bajtů vloží na společnou frontu.

3.1.2.2 Třída JPEGEncoder

Instance této třídy získá počet potřebných instancí třídy `EncodingThread` z instance třídy `AvgDelayCalculator` a následně je vytvoří. Vždy když je požadavek na komprimaci obrazu, je vybrána jedna vyčkávající instance a té je snímek ke komprimaci přidělen. Připravená data k odeslání jsou odebírána z fronty a předávána dál ke zpracování.

3.1.3 Síťová komunikace a přenos

Instance třídy `DataManager` z balíčku `datahandler` udržuje informace o připojených klientech a má za úkol periodicky předávat připravená data třídám určeným pro odesílání dat. O samotný přenos se starají třídy z balíčku `networking`:

- `StreamingAppServer`
- `StreamingThread`
- `StreamingHttpServer`

3.1.3.1 Třída StreamingAppServer

Slouží pro přenos obrazu přímo mezi aplikacemi *Mirroring*, využívá k tomu tříd `StreamingThread` a `DataManager`. Instance této třídy naslouchá na portu 4245⁶, na který se připojují klienti. Připojení klienti tvoří instance třídy `StreamingThread`, které jsou dále předány instanci třídy `DataManager`.

3.1.3.2 Třída StreamingThread

Třída jejíž úkol je komunikovat s cílovým klientem a odesílat jednotlivé snímky. Používá k tomu instance tříd `Socket` pro navázání spojení a `DataOutputStream`, `DataInputStream` pro odesílání dat.

Komunikace mezi aplikacemi *Mirroring* je řízena způsobem dotaz - odpověď. Instance obsluhuje tyto dotazy:

- HELLO (Hello) - Tímto dotazem klient oznamuje úmysl připojit se k sezení a přijímat vysílané snímky. Instance vrací odpověď zda je sezení chráněno heslem, nebo je volně přístupné.
- PASSWD (Password) - Je-li sezení zaheslováno, odešle klient tento dotaz spolu se zadaným heslem. Instance odpoví zda heslo je správné. Pokud heslo souhlasí, je instanci nastaven příznak, že je možné touto instancí odesílat obraz.
- READY (Ready to receive) - Tímto dotazem dává klient oznamuje, že je připraven pro příjem snímků a instance může zahájit vysílání.

⁶Port byl při implementaci třídy náhodně zvolen z povoleného rozsahu.

- QUIT (Leave session) - Při přijetí tohoto dotazu je ukončeno spojení s klientem.
- ACK (Acknowledgement) - Klient periodicky zasílá potvrzení, že je stále připojen. Pokud se do daného časového úseku neozve je vyřazen z evidence a spojení je ukončeno.

3.1.3.3 Třída StreamingHttpServer

Instance této třídy slouží k odesílání snímků na server za účelem zobrazení pomocí webové služby. Třída využívá instancí tříd `screenCapture` a `JPEGEncoder` pro pořizování a komprimaci obrazu. Po připojení k serverové aplikaci *MirroringServer* ihned zasílá pořizené obrazy.

3.1.3.4 Třída ConnectionManager

Tato třída je součástí balíčku `datahandler`. Instance této třídy zodpovídá za přenos metadat mezi aplikací a serverem. Tím je myšleno zasílání a příjem dat pro zahájení sdílení, udržování sezení na serveru a následně příznak o ukončení sdílení a odpojení od serveru (komunikace vysvětlena viz. [3.2.1.2](#)).

3.1.4 Zobrazení přijatého snímku

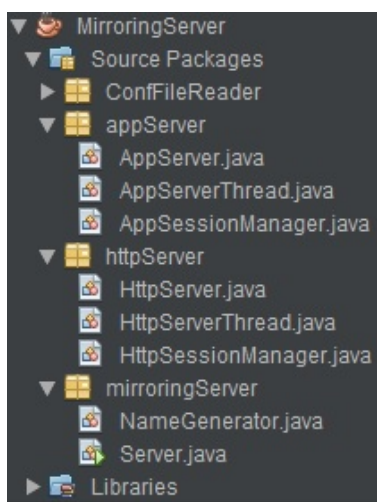
Přijímání a zobrazování snímků má za úkol třída `ReceivingClient`, která je součástí balíčku `networking`. Využívá k tomu třídu `ConnectionManager`, která nejprve vyžádá ze serveru informace nutné pro připojení k aplikaci. Když jsou tyto informace dostupné, připojí se aplikace pomocí instancí tříd `ReceivingClient` a `Socket` k vysílající aplikaci *Mirroring*. Z přijímaných dat posléze vytváří nové instance třídy `Image`, které dále instance objektu `ImageView` zobrazí.

3.2 Java aplikace MirroringServer

Java sever *MirroringServer* je konzolová aplikace využívající port 4243 pro vyřizování dotazů a port 4242 pro přenos snímků při zobrazení webovou službou.

Hlavní funkce jsou:

- Udržování informací o aktivních spojení.
- Příjem a příprava snímků pro webovou službu.



Obrázek 4: Struktura a jednotlivé třídy serverové aplikace

3.2.1 Udržování informací o aktivních spojení

Při přenosu obrazu mezi aplikacemi server udržuje informace pro navázání spojení, které na vyžádání poskytuje. Tedy pokud je založeno sezení, zaeviduje se na serveru IP adresa vysílajícího zařízení a vygenerovaný název sezení (generování zamezí možnosti odhadnutí názvu sezení). Chce-li se klient připojit do sezení, zašle na server dotaz s názvem. Pokud název existuje, server zpět zašle IP adresu vysílajícího zařízení. Klient dále komunikuje pouze se zařízením na navrácené IP adrese a přenos obrazu následně řeší samy aplikace mezi sebou a server zde již nehraje roli.

Tato funkcionalita je zajištěna třídami z balíčku (ostatní balíčky viz obr. 4) `appServer`:

- `AppServer`
- `AppServerThread`
- `AppSessionManager`

3.2.1.1 Třída `AppServer`

Server pomocí instance této třídy naslouchá na portu 4243. Zašle-li klient na tento port dotaz, je předán instancí třídy `AppServerThread`.

3.2.1.2 Třída `AppServerThread`

Tato třída obstarává komunikaci mezi serverem (*MirroringServer*) a aplikací (*Mirroring*), která je řešena stylem dotaz - odpověď. Klient zašle dotaz, server jej vyhodnotí, odpoví a komunikace se ukončí. Po identifikaci dotazu v komunikaci je samotné vyřízení dále předáno instancí třídy `AppSessionManager`.

Dotazy, které třída obsluhuje jsou:

- `ACK` (Acknowledgment) - Přijímán s názvem sezení a slouží pro udržování informace, že sezení je stále aktivní. Sezení se následně periodicky kontrolují a pokud nějaká aplikace do daného časového úseku `ACK` nezašle, je sezení vyřazeno z evidence.
- `QUIT` (Quit Session) - Přijetí tohoto dotazu znamená úmysl klienta ukončit vysílání a server vyřadí dané sezení z evidence.
- `INRQ` (Information Request) - Opět přijímán s názvem sezení, kdy klient pro daný název požaduje zaslání IP adresy. Server zasílá IP adresu pokud sezení existuje, jinak zasílá prázdnou odpověď jako indikátor neexistujícího sezení.
- `NEW` (New Session) - Žádost o vygenerování názvu a zaevidování informací o sezení. Server navrácí vygenerovaný název pro následnou distribuci klientem.

3.2.1.3 Třída `AppSessionManager`

Instance této třídy generuje názvy sezení pomocí instance třídy `NameGenerator` a udržuje je s danými IP adresami v paměti. Dále obsluhuje dotazy `ACK`, `QUIT`, `INRQ`, `NEW` - popsáno výše.

3.2.2 Příjem a příprava snímků pro webovou službu

Chce-li klient sdílet obrazovku pomocí webové služby, využívá se server *MirroringServer* jako prostředník, na který klient pořízené snímky zasílá a server je následně připraví pro zobrazení webovým serverem. Příprava spočívá ve vytvoření složky s daným jménem, do které se nakopíruje soubor `index.php` a následně zde budou vkládány přijaté snímky.

Využité třídy z balíčku `httpServer`:

- `HttpServer`
- `HttpSessionManager`
- `HttpServerThread`

3.2.2.1 Třída `HttpServer`

Nejdříve instance třídy načte konfigurační soubor `mirroring.conf`, kde testuje jestli zadaná složka i soubor `index.php` existuje (pokud neexistuje je hlášena chyba). Načtená data jsou předána instanci třídy `HttpServerThread`. Následně instance třídy `HttpServer` začne naslouchat na portu 4242, který slouží pro přenos obrazu. Klient je rovnou předán instanci třídy `HttpServerThread`. Zde se nevyužívá žádných dotazů, pokud se klient na tento port připojí, ihned je navrácena url adresa s požadovaným obrazem a zahajuje se přenos snímků.

3.2.2.2 Třída `HttpSessionManager`

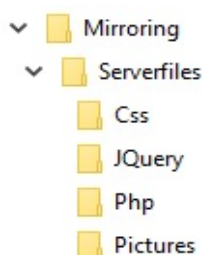
Instance generuje názvy složek pomocí instance třídy `NameGenerator`. Tyto názvy eviduje a v kombinaci s daty z konfiguračního souboru vytváří složky, kam jsou následně ukládány přijaté snímky daného klienta. Ke každé složce je vytvořena i odpovídající url adresa. Pokud je spojení ukončeno má tato instance za úkol smazat daný adresář a celý jeho obsah.

3.2.2.3 Třída `HttpServerThread`

Tato instance nejdříve nakopíruje do vytvořené složky soubor `index.php`. Adresu složky a souboru poskytla instance třídy `HttpSessionManager`. Následně začne přijímat obrazy klienta a přijatá data převádí pomocí třídy `ImageIO` na soubory formátu *JPG*. Tyto soubory následně ukládá do vytvořené složky odkud jej čte webový server.

3.3 Webový server

Úlohou této webové aplikace, je zobrazovat obrazy připravené aplikací `MirroringServer` na dané url adrese.



Obrázek 5: Adresářová struktura webové aplikace

3.3.1 Zobrazení snímku pomocí webového prohlížeče

Zobrazení snímku v prohlížeči je řízeno skriptem situovaném v souboru `js.js`, který se nachází ve složce `Serverfiles\jQuery` (struktura webového serveru viz obr. 5). Skript periodicky načítá měnící se snímek a vkládá ho do webové stránky tvořené souborem `index.php`. Uchovávání snímku je prováděno ve dvou vrstvách a pokud se snímek nepodaří načíst, tak se první vrstva schová a zobrazí se vrstva druhá s minulým obrazem. Není tedy vidět, že nastala chyba a je zřetelné pouze prodlení mezi opětovným načtením obrazu. Pokud se objeví dvě chyby po sobě, zobrazí se třetí vrstva která obsahuje informaci o ukončení sezení.

4 Spuštění aplikace Mirroring

Tato část textu slouží jako návod, jak aplikaci připravit a nastavit pro bezproblémový chod. Aplikaci není nutno instalovat, projekt obsahuje pouze spustitelné soubory. Spuštění je podmíněno následujícím.

4.1 JRE 8

Aby bylo možné aplikaci spustit, je nutné mít na počítači instalované běhové prostředí Java verze 8 (*JRE 8*) a vyšší. Aplikace *Mirroring* navíc vyžaduje knihovnu *JavaFX*. Software je volně dostupný z webových stránek společnosti *Oracle*.

4.2 Spuštění aplikace Mirroring

Podmínkou pro umožnění síťové komunikace je nastavení adresy serveru se spuštěnou aplikací *MirroringServer* a volný port 4245. Port nesmí být tedy využíván jinou aplikací. Jeli port volný, je nutné zkontrolovat zda firewall systému aplikaci neblokuje a popřípadě jej povolit.

4.3 Spuštění aplikace MirroringServer

Pro spuštění je nutné, stejně jako v případě aplikace *Mirroring*, aby porty 4242 a 4243 byly před spuštěním volné. Stejně tak je nutné zajistit, aby aplikaci neblokoval systémový firewall. Pro komunikaci s webovým serverem je nutné správně nastavit hodnoty v konfiguračním souboru `mirroring.conf`.

4.4 Webový server

Adresářovou strukturu nakopírujeme do kořenové složky webového serveru. Pro zajištění správné funkčnosti je nutné, aby se na serveru nacházela i spuštěná aplikace *MirroringServer*.

4.4.1 Příprava chodu v lokální síti

Pro využití v lokální síti se předchozí postup spuštění nemění. Je pouze nutné mít v síti připojený webový server. Jednou z možností je využití lokálního webového serveru pomocí technologie *WAMP* popřípadě *XAMPP*. Tyto technologie vytvoří na osobním počítači webový server, který dále podporuje webové technologie jako jsou *PHP*, *HTML*, *JavaScript* a další.

4.4.2 Příprava chodu v síti internet

Pro chod v síti internet je nutné aby server i vysílající aplikace měli veřejnou adresu a bylo možné se k nim připojit. Nutnost veřejné adresy plyne z překladu síťových adres v internetu, takzvaný *IP NAT*. Ten ten počítače z lokální sítě při vstupu do internetu „schová“ za jedinou adresu. Pro spojení aplikací je ovšem nutné znát přesnou adresu vysílajícího klienta. To můžeme vyřešit nastavením veřejných adres.

4.4.2.1 Poznámky

Pro bezpečný chod serveru je doporučeno využívat šifrované komunikace pomocí protokolu *HTTPS (Hypertext Transfer Protocol Secure)*.

Tato nastavení jsou pro daný webový server jedinečná a je zapotřebí, aby tuto komunikaci webový server umožňoval. Tyto možnosti jsou tedy ponechány na volbě uživatele.

5 Technologie použité při vývoji

5.1 HTML

Hyper Text Markup Language (HTML) je standardní značkovací jazyk pro vytváření webových stránek [6].

- Popisuje strukturu webových stránek pomocí jednotlivých značek (tagů) a jejich atributů.
- Značky (tagy) označují části obsahu, například nadpis nebo tabulku.
- Prohlížeče nezobrazují značky HTML, ale používají je pro vykreslení obsahu stránky.
- Zveřejňuje online dokumenty s titulky, tabulkami, seznamy, fotkami, atd.

5.2 CSS

Cascading Style Sheets (CSS) je jazyk pro popis vzhledu webových stránek, včetně barev, rozvržení a písma. Umožňuje přizpůsobit obsah různým typům zařízení, jako jsou velké obrazovky počítačů nebo malé obrazovky mobilních zařízení. Oddělení HTML od CSS usnadňuje údržbu stránek, sdílení stylů mezi stránkami a přizpůsobení stránek různým prostředím [7].

5.3 PHP

Hypertext Preprocessor (PHP) je široce používaný open source⁷ skriptovací jazyk. Jednotlivé skripty jsou prováděny na straně serveru, takzvaný back-end jazyk [8].

- Generuje statický i dynamický obsah stránky.
- Provádí operace se soubory na serveru a databázemi.
- Shromažďuje data z formulářů a provádí nad nimi operace.
- Odesílá a přijímá soubory cookie.
- Umožňuje šifrování dat data.

⁷Software s veřejně dostupnými zdrojovými kódy.

5.4 JavaScript

JavaScript je interpretovaný⁸, objektový⁹, case sensitivní¹⁰ klientský skriptovací jazyk. *JavaScript* není typickým objektovým jazykem. Dané objekty nejsou instancemi třídy, které v klasickém objektově orientovaném programování slouží jako předloha objektu.

- Jazyk vykonává uvedené skripty ve webovém prohlížeči.
- Jedná se o front-end dynamicky typovaný jazyk.
- Je to slabě typovaný jazyk. Proměnné se uvádějí pomocí klíčového slova `var`.
- Datové typy jazyka jsou: číselná hodnota, řetězec a objekt.

5.4.1 jQuery

jQuery je knihovna programovacího jazyka *JavaScript*, jejímž účelem je usnadnit používání *JavaScriptu* na webových stránkách [9]. Daná knihovna pouze rozšiřuje *JavaScript* o upravené funkce pro pohodlnější programování. Při vykonávání metod a funkcí se zdrojový kód knihovny *jQuery* překládá do čistého *JavaScriptu*.

5.5 Java

Java je statický, silně typovaný, objektově orientovaný programovací jazyk. Přenositelnost, jednoduchost a fakt, že jazyk je dostupný zdarma, dělá z *Javy* jeden z celosvětově nejrozšířenějších programovacích jazyků. Nalezneme jej v počítačích, embedded zařízení¹¹, herních konzolách, v internetu a je součástí známého mobilního operačního systému *Android*. Přenositelnost je zajištěna překladem zdrojového kódu v *Javě* do *Java bytecode*¹², který se následně interpretuje nebo se pomocí běhového prostředí *JRE*¹³ překládá do instrukční sady cílového procesoru.

⁸Nemusí se kompilovat.

⁹Využívá objektů prohlížeče a zabudovaných objektů.

¹⁰Záleží na velikosti písem v zápisu.

¹¹Jednouúčelový počítač, který je součástí daného zařízení, například pračka

¹²Instrukční sada virtuálního stroje Java (JVM).

¹³*Java Runtime Environment* (JRE) je běhové prostředí obsahující virtuální stroj.

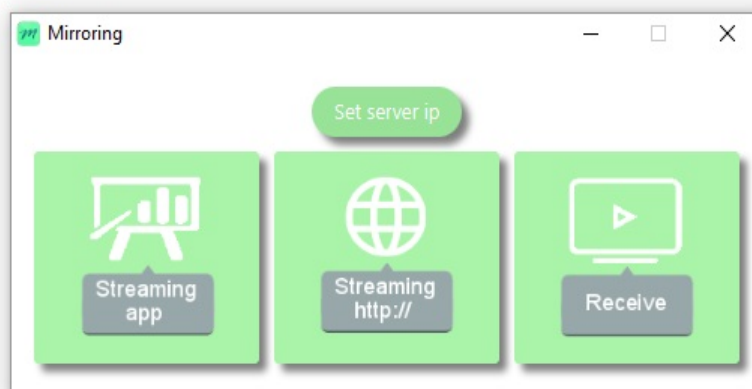
6 Uživatelský manuál aplikace

Uživatel si po spuštění aplikace *Mirroring* pro operační systémy *Windows* může vybrat jednu ze tří možností, které jsou popořadě reprezentovány dlaždicemi *Streaming app*, *Streaming http://* a *Receive* (viz obr. 6).

Nad dlaždicemi se nachází tlačítko pro nastavení IP adresy. Pro správný chod aplikace *Mirroring* je nutné před volbou funkce nejprve nastavit správnou IP adresu severu se spuštěnou aplikací *MirroringServer*.

Funkce dlaždic jsou následující:

- *Streaming app* - Sdílení obrazu do jiné aplikace *Mirroring*. Při této volbě je obraz plynulý a je možné nastavit kvalitu snímků nebo chránit přístup do sezení heslem.
- *Streaming http://* - Sdílení obrazu do webového prohlížeče. Pro příjem snímků není potřeba aplikace *Mirroring*. Obnovovací frekvence snímků je výrazně nižší.
- *Receive* - Připojení k sezení a příjem obrazu vysílaného jinou aplikací *Mirroring*.



Obrázek 6: Úvodní scéna

6.1 Funkce pro sdílení mezi aplikacemi

Po zvolení funkce *Streaming app* se zpřístupní panel pro ovládání sdílení obrazu (viz obr. 7). Nad panely se nachází textové pole, které slouží k vypisování informací pro uživatele. Pod panely se poté nachází tlačítka *Pause* pozastavující přenos, který je možné posléze kliknutím na *Continue* obnovit. Vedle *Pause* je situováno tlačítka *Stop stream* ukončující sezení a navracející uživatele na úvodní scénu.

6.1.1 Pravý panel

V pravém panelu se nachází indikátor zda je přenos pozastaven (indikátor ukazuje možnost *PAUSED*), nebo zda je sdílení stále aktivní (indikátor ukazuje možnost *REC*). Dále se zde nachází možnost upravení kvality snímků a tedy i rychlosti přenosu.

Na výběr jsou tři možnosti:

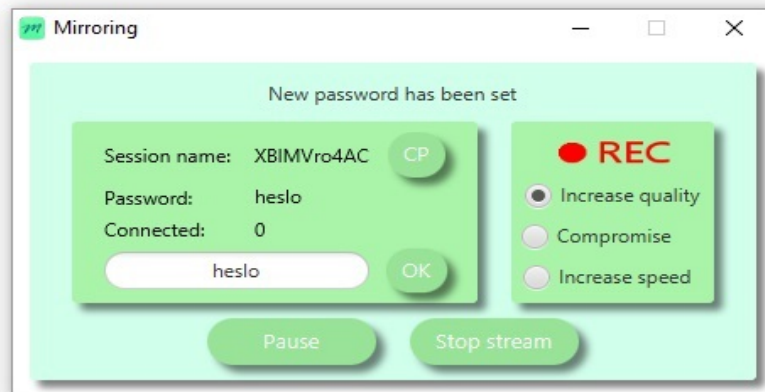
- *Increase quality* - Upřednostnění kvality obrazu. Náročnější na rychlost přenosu.
- *Compromise* - Kompromis mezi kvalitou a náročností na přenos.
- *Increase speed* - Upřednostnění rychlosti přenosu před kvalitou obrazu.

6.1.2 Levý panel

Na levém panelu se nacházejí jednotlivé informace o spojení a textové pole pro nastavení hesla. Panel obsahuje dvě tlačítka. Tlačítka *CP* uloží název sezení do schránky počítače, *OK* uloží vložené heslo. Při potvrzení prázdného pole je heslo vymazáno.

Informace obsahující levý panel:

- *Sessions name* - Zobrazuje identifikátor sezení, který nutný pro připojení klientů.
- *Password* - Zobrazuje heslo, chránící přístup do sezení. Není-li sezení chráněno heslem, obsahuje pole hodnotu *NOT SET*.
- *Connected* - Počet připojených uživatelů.



Obrázek 7: Grafické rozhraní pro sdílení mezi aplikacemi

6.2 Funkce pro sdílení do webového prohlížeče

Při zvolení funkce *Streaming http://* se zpřístupní panel obsahující informace o adrese, na které je zprostředkované zobrazení vysílaných snímků. Nacházejí se zde tlačítka *Pause/Continue*, *Stop stream* a indikátor přenosu se stejnou funkcionalitou jako při sdílení mezi aplikacemi.

6.3 Funkce pro příjem vysílaného obrazu

Funkce *Recieve* otevře okno požadující identifikátor sezení. Jeli sezení chráněno heslem, objeví s v dalším kroku okno na ověření hesla.

Následně se objeví okno s přijímaným obrazem, které obsahuje pouze dvě tlačítka:

- *Stop* - Přeruší příjem obrazu a navrátí klienta na úvodní scénu.
- *Fullscreen/Smallscreen mode* - Spustí nebo ukončí mód celé obrazovky.

7 Uživatelský manuál serveru

Manuál konzolové aplikace *MirroringServer* pro operační systému *Windows*, která má následující funkce:

- Udržovat informace o IP adrese vysílajícího klienta a identifikátoru sezení, která jsou aktivní.
- Poskytovat informace o sezení pro možnost navázání spojení mezi klienty.
- Přijímat vysílané snímky a připravit je na zobrazení pomocí webové služby.
- Po ukončení sezení smazat zbylé soubory.

7.1 Nastavení konfiguračního souboru

Před spuštěním aplikace je nutné správně nastavit jednotlivé informace v konfiguračním souboru `mirroring.conf`, který se musí nacházet ve stejné složce jako aplikace. Řádky začínající znakem „#“ označují komentář a jsou ignorované. Všechny ostatní řádky jsou považovány za hodnoty. Pořadí hodnot v souboru nesmí být změněno.

Soubor `mirroring.conf` obsahuje:

1. „Path to index.php“
 - Cesta k souboru `index.php`, který se nachází ve složce webového serveru „`Serverfiles\PhP`“.
2. „Directory for new sessions“
 - Cesta ke kořenové složce webového serveru.
3. „URL“
 - Doménová adresa serveru.

7.2 Spuštění a obsluha serveru

Spuštění je možné příkazem „`java -jar "MirroringServer.jar"`“ v příkazové řádce s nastaveným adresářem, kde se aplikace nachází. Pro zjednodušení spuštění se u aplikace nachází dávkový soubor `runServer.bat`, který otevře příkazovou řádku a aplikaci spustí. Jeli k dispozici více síťových rozhraní, jsou jejich informace zobrazeny a je nutné vybrat požadované rozhraní. Jeli rozhraní pouze jedno, tato možnost odpadá.

Server se ovládá pomocí sekvence instrukcí psaných do konzole (ukázka viz obr. 8). Instrukce, které server přijímá jsou následující:

- *manual* - Vypíše použitelné příkazy s vysvětlivkami.
- *start* - Pokud je konfigurační soubor v pořádku spustí server.
- *stop* - Zastaví server a vypne aplikaci.
- *restart* - Vypne a znovu zapne server.
- *sessions* - Zobrazí aktivní sezení.
- *get url* - Vypíše doménovou adresu serveru v konfiguračním souboru.
- *get inpath* - Vypíše cestu k souboru `index.php` v konfiguračním souboru.
- *get sedir* - Vypíše cestu ke kořenovému adresáři webového serveru v konfiguračním souboru.

```

C:\WINDOWS\system32\cmd.exe
----- Manual -----
manual      - See allowed commands
start       - Start servers
stop        - End program
restart     - Restart servers
sessions    - See all sessions
get url     - See current url
get inpath  - See index.php path
get sedir   - See current sessions directory
-----
> start
Found more then one network interface, what is IP address of
this computer? Address must match to IP adress of web server!

Interface number: 0.
Display name: Intel(R) Centrino(R) Wireless-N 1030
Interface name: wlan0
IP: 10.0.0.40

Interface number: 1.
Display name: VirtualBox Host-Only Ethernet Adapter
Interface name: eth0
IP: 192.168.56.1

Interface number: 0

IP Address
10.0.0.40
Attached data
C:\wamp64\www\Mirroring\Serverfiles\Php\index.php
C:\wamp64\www\Mirroring\
http://localhost/Mirroring/
HTTP SERVER RUNNING
APP SERVER RUNNING
> sessions
----- HTTP SESSIONS -----1
U5XPXoUE75
----- APPL SESSIONS -----1
XBIMUro4AC:10.0.0.40
>

```

Obrázek 8: Ukázka konzolové aplikace MirroringServer

Závěr

Práce kladla za cíl implementovat nástroj, který jednoduše umožní sdílení obrazu pracovní plochy mezi počítači v síti. K získání této funkcionality byl projekt rozdělen do tří částí. Bylo potřeba implementovat dva samostatné programy a webovou službu.

První částí bylo vytvořit program *Mirroring*, který by byl schopný zprostředkovat obraz pracovní plochy uživatele, připravit jej k přenosu a snímek odeslat koncovým uživatelům. To vše s co nejjednodušším ovládáním pro uživatele. Druhou částí bylo vytvoření programu *MirroringServer* s dvěma hlavními funkcemi. Zaprvé evidování vysílajících aplikací a tím i umožnění ostatním získat potřebné informace pro navázání spojení. Druhá funkce se pak zabývala přípravou snímků pro zobrazení webovou službou. Poslední částí projektu byla webová služba, zobrazující snímky připravené programem *MirroringServer* koncovému uživateli.

Spojením těchto tří prvků jsme získali výsledný software, který umožňuje snadno prezentovat určitý postup, fotografie, text a jiné.

Text slouží jako popis motivace, funkcionality včetně implementačního řešení a uživatelský manuál nástroje pro zrcadlení obrazovek jménem *Mirroring*.

Conclusions

The main goal of the thesis was to implement a tool that simply allows desktop image sharing among computers on the network. To achieve this functionality, the project was divided into three parts. It was necessary to implement two separate programs and a web service.

In the first part we created program *Mirroring* which is able to take the user's desktop image, prepare it for transmission and send it to end users. Second part was dedicated to creation of the program *MirroringServer* with two main features. The first one was the registration of the streaming applications and thus allowing others to obtain the necessary information to establish the connection. The second one was aimed at the preparation of images for displaying them by the web service. The last part was focused on the web service itself which displays images prepared by *MirroringServer* to the end user. By combining these three elements, we have obtained the final software that allow users to easily present a certain procedure, photos, text, and more.

The text provides a description of motivation, functionality including the implementation solution and user manual for screen mirroring tool called *Mirroring*.

A Obsah příloženého CD

Na příloženém CD se nachází dokumentace, zdrojové kódy aplikací a implementovaný projekt. Složky na disku a jejich obsah je popsán zde:

apps/

Složka je rozdělena do tří podadresářů.

- `apps/java application` - Zde se nachází program `MIRRORING` spolu s knihovnamy a souborem nutným pro chod aplikace.
- `apps/java server` - Obsahuje program `MIRRORINGSERVER` včetně konfiguračního souboru a souboru `runServer.bat`, který slouží pro pohodlnější spuštění programu.
- `apps/web server` - Kompletní adresářová struktura webové aplikace pro zkopírování na webový server.

doc/

Text práce ve formátu PDF a potřebné soubory k vytvoření textu ve formátu ZIP.

src/

Zdrojové kódy programů `MIRRORING`, `MIRRORINGSERVER` a webové aplikace, které jsou rozděleny do stejnojmenných podadresářů.

readme.txt

Instrukce pro spuštění programů a nasazení webové aplikace na webový server, včetně požadavků pro bezproblémový chod.

Literatura

- [1] INSTANT MESSAGING MEANING IN THE CAMBRIDGE ENGLISH DICTIONARY, Cambridge Dictionary | Free English Dictionary, Translations and Thesaurus [online] Copyright © Cambridge University Press [cit. 21.5.2017]. Dostupné z: <http://dictionary.cambridge.org/dictionary/english/instant-messaging>
- [2] INTERNET STREAMING: WHAT IT IS AND HOW IT WORKS, Lifewire [online][cit. 21.5.2017]. Dostupné z: <https://www.lifewire.com/internet-streaming-how-it-works-1999513>
- [3] INTEL WIRELESS DISPLAY V PRAXI: HRAČKA DO KAŽDÉHO OBÝVÁKU - CNEWS.CZ, Cnews.cz | Od tranzistorů až po PC sestavy [online][cit. 21.5.2017]. Dostupné z: <https://www.cnews.cz/intel-wireless-display-v-praxi-hracka-do-kazdeho-obyvaku/>
- [4] WHAT IS CHROMECAST?, Chromecast - Google [online][cit. 21.5.2017]. Dostupné z: https://www.google.com/intl/en_us/chromecast/tv/chromecast/
- [5] DLNA: WHAT IT IS AND WHAT YOU NEED TO KNOW | TECHRADAR, TechRadar | The source for tech buying advice | TechRadar [online]. Copyright © [cit. 21.05.2017]. Dostupné z: <http://www.techradar.com/news/digital-home/home-networking/dlna-what-it-is-and-what-you-need-to-know-1079015>
- [6] HTML5 TUTORIAL, W3Schools Online Web Tutorials [online][cit. 21.5.2017]. Dostupné z: <https://www.w3schools.com/html/default.asp>
- [7] HTML & CSS, World Wide Web Consortium (W3C) [online][cit. 21.5.2017]. Dostupné z: <https://www.w3.org/standards/webdesign/htmlcss>
- [8] PHP 5 INTRODUCTION, W3Schools Online Web Tutorials [online][cit. 21.5.2017]. Dostupné z: https://www.w3schools.com/php/php_intro.asp
- [9] JQUERY NÁVOD, jQuery návod - vše okolo jQuery [online][cit. 21.5.2017]. Dostupné z: <http://jquery-navod.cz/kategorie-ostatni-clanky/1-uvodni-clanek>