

Sem vložte zadání Vaší práce.

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



Diplomová práce

Children Usability Lab - management video streamů

Bc. Patrik Faistaver

Vedoucí práce: Ing. Jiří Chludil

15. března 2018

Poděkování

Rád bych poděkoval vedoucímu mé diplomové práce Ing. Jiřímu Chludilovi i oponentovi Ing. Jiřímu Melnikovovi za užitečné připomínky a mnoho cenných rad. Dále chci poděkovat své rodině a blízkým přátelům za podporu a motivaci při tvorbě této práce i během celého studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) 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. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 15. března 2018

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2018 Patrik Faistaver. 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

Faistaver, Patrik. *Children Usability Lab - management video streamů*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.

Abstrakt

Tato diplomová práce navazuje na bakalářskou práci Karolíny Solanské s názvem Children Usability Lab - aplikace pro správu laboratoře. Její bakalářská práce je zde rozšířena o návrh a implementaci subsystému pro management veškerých záznamů nahrávaných v laboratoři Children Usability Lab. Součástí této práce je také analýza současného systému i existujících řešení pro požadovanou správu záznamů, dále pak návrh, realizace a její následné testování. Implementované rozšíření stávajícího systému plně pokrývá dohodnuté zadání i specifikované požadavky.

Závěrem této diplomové práce je podrobení subsystému integračním i akceptačním testům. Subsystém se podařilo úspěšně nasadit do počítačového prostředí laboratoře Children Usability Lab, kde usnadňuje a zefektivňuje tamní usability testování.

Klíčová slova testování použitelnosti, webová aplikace, laborator uživatelského testování, management záznamů, správa videa

Abstract

Sem doplňte ekvivalent abstraktu Vaší práce v angličtině.

Keywords Nahradte seznamem klíčových slov v angličtině oddělených čárkou.

Obsah

Úvod	1
1 Cíl práce	3
2 Analýza	5
2.1 Definice pojmů	5
2.2 Rozbor zadání	5
2.3 Předchozí práce a aktuální stav	7
2.4 Specifikace požadavků	8
2.5 Existující komplexní řešení	12
2.6 Výběr technologií	16
3 Návrh	19
3.1 Praktiky a vzory	19
3.2 Architektura systému	19
3.3 Frontend	19
3.4 Backend	19
4 Implementace	21
4.1 Prostředí a kompatibilita	21
4.2 Použití knihovny Libyuri	21
4.3 Vybrané zajímavé funkce	21
4.4 Instalační a uživatelská příručka	21
5 Testování	23
5.1 Integrovaní testování	23
5.2 Akceptační testování	23
Závěr	25

Literatura	27
A Seznam použitých zkratek	31
B Obsah přiloženého CD	33

Seznam obrázků

Seznam tabulek

2.1	Porovnání technologií pro backend	18
-----	---	----

Úvod

Testování použitelnosti je proces při kterém se aplikace testuje přímo za pomoci vzorku koncových uživatelů a díky kterému se autoři aplikace dozvědí, jak jsou různé prvky uživatelského rozhraní srozumitelné a jak dobře jsou uživatelé schopni se v aplikaci orientovat. V současném světě technologií tento proces vyžaduje informační systém jako řešení pro usnadnění testování, plánování i komunikace mezi jednotlivými subjekty v laboratoři testování použitelnosti. Vývoj takového systému byl zahájen bakalářskou prací Karolíny Solanské s názvem Children Usability Lab - aplikace pro správu laboratoře (odkazem [1]). Její práce vyvinula systém, který slouží jako webová aplikace zajišťující plánování a organizaci testů použitelnosti. Zmíněná bakalářská práce měla neobvykle větší počet stran, avšak podařilo se implementovat zcela funkční systém, ale i přesto je v něm spousta možností pro vyvinutí nových funkcí. Na tomto základě se dá říci, že se jedná o celkem rozsáhlý systém. Jeden ze směrů, kterým by se systém mohl začít rozšiřovat byla veškerá práce s videem což bylo ostatně zmíněno v uvedené práci v kapitole o dalším vývoji.

Laboratoř použitelnosti na Fakultě informačních technologií ČVUT v Praze je vybavena spoustou zařízení, která jsou schopná zaznamenávat různorodá data. Jedná se nejen o všechna výpočetní zařízení, na kterých probíhají samotné testy, jako hlavní stolní počítač, tablet apod., ale také spousta kamer, mikrofónů či dokonce snímač pohybu očí po obrazovce. Není pochyb o tom, že data z těchto nahrávání schopných zařízení mohou značně přispět ke zkvalitnění vyhodnocování testů použitelnosti. V aktuálním stavu je práce se záznamy velice obtížná a neúčinná, jak je uvedeno v kapitole o současném stavu (2.3). Z těchto důvodů je nutné rozšířit stávající systém o funkcionalitu, která práci s těmito záznamy značně ulehčí a zefektivní. Jedná se o to, aby uživatelé systému mohli jednoduše pouze přes webové rozhraní spustit a zastavit nahrávání z vybraných zařízení, záznamy roztrždit, uložit v požadovaném formátu a kompresi, upravovat zadaným výčtem způsobů, posílat přes síť do systému SAGE a podobně.

Cíl práce

Cílem této diplomové práce je nejprve seznámení se s bakalářskou prací Karolíny Solanské s názvem Children Usability Lab - aplikace pro správu laboratoře ([1]). V uvedené práci je třeba analyzovat veškeré specifické požadavky zaměřené na správu videa. Tyto požadavky je nutné znovu probrat s koordinátory laboratoře Children Usability Lab, aktualizovat je a zakomponovat do specifických požadavků této práce. Vedle převzatých požadavků se nadefinuje i spousta dalších, nových funkčních i nefunkčních požadavků, které byly nashromážděny během posledních let běhu testů v laboratoři. Další částí analýzy je pak ponoření se do světa technologií, které usnadňují práci s videem a z existujících je třeba vybrat pokud možno ty nejvhodnější, které by se daly použít v praktické části této práce.

Výstup analytické části si poté převezme návrhová a implementační část, ve kterých se pomocí nástrojů softwarového inženýrství promění nabyté poznatky v reálný produkt, tedy modul pro správu videa, který půjde se stávajícím systémem provázat.

Se vzniklým testovatelným řešením se pak postoupí do testovací části, kde se otestují funkčnosti zadané specifikací, provede se nasazení, případné opravy nalezených chyb a dále akceptační testování. Součástí práce bude také dokumentace včetně instalační příručky pomocí které bude možné systém nasadit v různých laboratořích pro testování použitelnosti.

Analýza

2.1 Definice pojmů

Frontend

Jedná se o část aplikace, která je viditelná běžným uživatelům.

Backend

Jedná se o část aplikace, uživatelům „schovaná“ za frontendem a slouží především ke zpracování dat.

CHUL

Children Usability Lab - labratoř testování použitelnosti na Fakultě informačních technologií ČVUT v Praze.

CDN

CDN neboli Content Delivery Network je systém serverů rozmístěných po internetu, které spolupracují pro zajištění rychlého doručení dat klientovi.

Hosting

Pronájem úložného prostoru uživateli na proprietárních serverech.

Framework

Jedná se o softwarovou strukturu fungující jako knihovna funkcí a podpůrných nástrojů, která usnadňuje vývoj jiných softwarových projektů.

2.2 Rozbor zadání

1. Analyzujte a aktualizujte funkční i nefunkční požadavky zaměřené na správu videa, uvedené v předchozí práci.

Jak již bylo zmíněno, tak tato práce navazuje na bakalářskou práci Karolíny Solanské s názvem Children Usability Lab - aplikace pro správu laboratoře (odkazem [1]). Aby navázání na předchozí práci mělo ten správný směr, je nutné analyzovat specifické požadavky z předchozí práce, ty projednat s aktuálním vedoucím práce, aktualizovat jejich znění a zařadit do specifických požadavků této práce. Náplň tohoto kroku zadání je detailněji probrána v sekci zabývající se požadavky předchozí práce (vizte 2.3.1) a také v sekci s definicí specifických požadavků této práce (2.4.1).

2. Analyzujte nástroje pro zpracování videa použitelné v infrastruktuře SAGE.

Většinová část této práce se týká zpracování videa, proto je nezbytné nutné se s oblastí zpracování videa důkladně seznámit. Nejprve je nutné si stanovit, co se bude po pomocných technologiích chtít, tedy seznam obecných funkcí a vlastností a jejich priorit. Na základě tohoto seznamu je třeba provést rešerši existujících technologií a pak všechny nashromážděné poznatky je třeba využít v porovnávací metodice pomocí které se vyberou ty nejvhodnější technologie. Rešerši existujících technologií a různých řešení se zabývá podkapitola o existujících komplexních řešeních (2.5) a to, které technologie byly zvoleny je podrobně shrnuto v podkapitole o vybraných technologiích (2.6).

3. Pomocí nástrojů a metod softwarového inženýrství navrhnete rozšíření stávající aplikace o moduly pro správu videa s následující funkcionalitou: přenos po síti včetně ukládání, střih dle zadaných parametrů, změna rozlišení, možnost komprese, editace meta-informací, transformace obrazu atd.

Tím nejzajímavějším bodem zadání je právě návrh zadaného řešení. V kapitole o návrhu (3) se jedná především o sloučení poznatků nabytých v analýze se znalostmi softwarového návrhu. Za užití nástrojů a metod softwarového inženýrství je proveden návrh frontendové i backendové části řešení. Pro moduly zajišťující jednotlivé typy úprav videa je využito technologií studovaných v analytické kapitole (vizte 2.6).

4. Implementujte zmíněné rozšíření stávající aplikace (schopné kooperace s aplikací pro správu laboratoře CHUL).

V této produktivní části je na základě návrhu implementován prototyp rozšíření stávající aplikace se všemi zadanými funkcionalitami. Výstupem implementace je pak produkt splňující specifikované požadavky alespoň rámcově, ale zároveň tak, aby bylo možné jej začít testovat a v případě nalezení chyb je opravovat.

5. Aplikaci podrobte integračním a akceptačním testům.

Tímto bodem se zabývá celá kapitola o testování (vizte 5), ve které se

provedou nejprve zmíněné integrační testy, kterými se ověří bezchybná komunikace mezi jednotlivými komponentami uvnitř rozšíření stávající aplikace. V poslední řadě se provedou akceptační testy právě v CHUL laboratoři.

2.3 Předchozí práce a aktuální stav

Jak bylo řečeno v úvodu, bakalářská práce Karolíny Solanské vyvinula systém, který slouží jako webová aplikace zajišťující plánování a organizaci testů použitelnosti. Tento systém byl vytvořen jako plně funkční vůči zadaným požadavkům a je nasazen v laboratoři CHUL. Systém je celý implementován v jazyku PHP s použitím frameworku Nette. Systém je nasazený na externím serveru s přístupem přes protokoly HTTP a HTTPS. Technologie pro použití databázi je PostgreSQL. Aktuální stav správy videa je příliš nepohodlný a zdoluhavý. Webová aplikace nijak nekomunikuje s jakýmkoliv zařízením v laboratoři CHUL. Vše okolo videí se tedy musí dělat víceméně ručně. V tomto kontextu se pracuje s úžasnou knihovnou *libyuri*(vizte [2]), která za pomoci dvou konfiguračních souborů spustí nahrávání, sloučí obrazy a zvuk, uloží a případně zobrazí v systému SAGE. Naneštěstí existují pouze tyto 2 konfigurace, které se ještě ke všemu musí ručně pouštět z příkazové řádky. Nyní neexistuje žádná jednoduchá možnost, jak pohodlně pracovat se záznamy (například ořezávat, přidávat titulek apod.) a tím se tedy bude zabývat tato práce.

2.3.1 Požadavky předchozí práce

Zde jsou uvedeny všechny funkční požadavky z předchozí práce, zaměřené na správu videa a které podpořily důvod vzniku této práce.

1. **Systém ke každému scénáři po proběhlém testu připojí video/video.**

Před začátkem testu si moderátor zvolí scénář a testera, který test provádí. Následně zvolí spuštění testu a test začne. Po dokončení scénáře pak moderátor zvolí ukončení testu. Systém si zaznamená čas a dle toho připojí k jednotlivým experimentům výsek videa odpovídající časovým značkám začátku a konce experimentu.

2. **Moderátor a zadavatel mohou video přehrávat, zastavit a převíjet.**

Webové grafické rozhraní bude umožňovat zobrazení náhledu videa, které bude fungovat, jako jednoduchý video přehrávač.

3. **Moderátor a zadavatel mohou pořídit screenshot videa a okomentovat ho, následně uložit k danému testu.**

Bude možné vybrat konkrétní snímek videa (v rámci sekund), ten okomentovat a uložit k danému testu. K testu bude možné přiložit několik

snímků (kde každý snímek bude vždy možné utvořit pouze z videí od daného testu).

4. Moderátor a zadavatel mohou video exportovat a stáhnout.

Moderátor a zadavatel mohou video vyexportovat v několika různých formátech. Možné je stažení videa, ale také umístění na server Youtube.

5. Moderátor a zadavatel mohou video konvertovat.

Moderátor a zadavatel mají možnost video převést na jiný formát, především pro kompatibilitu s různými zařízeními a prohlížeči.

6. Moderátor může k videu nahrát mluvený komentář.

Uživatel bude moci nahrát zvuk k libovolnému videu ze vstupního zařízení.

2.4 Specifikace požadavků

2.4.1 Funkční požadavky

F1 – Nahrávání záznamů

F1.1 – Zobrazení seznamu připojených nahrávacích zařízení.

Webová stránka umožní zobrazení tabulky obsahující zařízení, která mohou být aktuálně použita pro nahrávání. Zařízení nemusejí být jen kamery, ale také zvuková či jiná zařízení, jejichž komunikační protokol je v aplikaci implementován (zařízení jsou tedy ta v laboratoři CHUL, ale také zařízení nacházející na počítači uživatele, který webovou aplikaci používá). Každý záznam v tabulce bude obsahovat jednoznačnou identifikaci zařízení a případně i aktuální náhled, bude-li se jednat o kameru.

F1.2 – Možnost nahrávání záznamu z uživatelem vybraných nahrávacích zařízení.

Tabulka (z předchozího případu) obsahující zařízení schopná nahrávání bude obsahovat checkboxy, které umožní vybrat ta zařízení, na kterých se nahrávání spustí. Při nahrávání bude vytvářeno nejméně tolik záznamů, kolik zařízení bylo vybráno (i více, pokud zařízení nahrává video i zvuk současně).

F1.3 – Zobrazení seznamu již nahraných záznamů.

Webová stránka umožní zobrazení tabulky obsahující záznamy, které byly v minulosti nahrané (a jsou v úložišti dostupném webovému serveru). Každý záznam v tabulce bude obsahovat jednoznačnou identifikaci nahrávky a případně i náhled, bude-li se jednat o obrazový záznam.

F2 – Přehrávání záznamů

F2.1 – Možnost přehrát vybraný záznam.

Existující záznamy zobrazené v tabulce zmíněné v případě výše bude možné přehrát ve webovém prohlížeči. Přehrávání bude možné pozastavit v libovolném čase a bude možný přesun kamkoliv v časové ose videa.

F2.2 – Moderátor a zadavatel mohou změnit hlasitost přehrávaného záznamu.

Uživatel bude moci změnit hlasitost zvuku přehrávaného záznamu. Uživatel si v nastavení přehrávaného videa zvolí hlasitost zvuku v rozsahu 0 - 100%.

F2.3 – Moderátor a zadavatel mohou změnit rozlišení přehrávaného videa.

Uživatel bude moci změnit rozlišení videa. Uživatel si v nastavení přehrávaného videa zvolí rozlišení, ze seznamem podporovaných rozlišení.

F2.4 – Moderátor a zadavatel mohou změnit rychlost přehrávání videa.

Uživatel bude moci ve webovém přehrávači změnit snímkovou frekvenci videa. Uživatel si v nastavení přehrávaného videa zvolí rychlost, ze seznamem podporovaných hodnot.

F2.5 – Moderátor a zadavatel mohou přehrávání přepnout do celoobrazovkového režimu.

Uživatel bude moci ve webovém přehrávači zapnout či vypnout přehrávání na celou obrazovku (tzv. full screen mód).

F2.6 – Moderátor a zadavatel mohou vybrané video streamovat do systému SAGE.

Uživatel bude moci spustit streamování konkrétního videa do systému SAGE. Systém díky pomocnému softwaru zajistí stream na uživatelem zadanou IP adresu se systémem SAGE.

F3 – Úprava záznamů**F3.1 – Moderátor a zadavatel mohou sloučit několik videí paralelně do mřížky.**

Uživatel bude moci spojit několik videí vedle sebe do mřížky (např. do čtverce). Tyto videa se budou ve finálním složeném videu přehrávat souběžně. Pokud některé video bude kratší než ostatní, bude doplněno opakujícím se snímkem jednolitě barvy.

F3.2 – Moderátor a zadavatel mohou video oříznout na menší časový úsek.

Uživatel bude moci vybrat časy, na které chce aktuální video oříznout. Vzniklé oříznutí bude možné na video aplikovat a poté jej upravovat dále.

F3.3 – Moderátor a zadavatel mohou přidat či nahradit zvukovou stopu videa.

Uživatelé budou zobrazeny existující zvukové záznamy, ze kterých si bude moci jeden vybrat. Vybraný záznam bude poté možné přidat k vybranému videu. Pokud video již zvukový záznam má, přepíše se za nový. Pokud bude zvukový záznam delší než video tak se ořízne na délku videa.

F3.4 – Moderátor a zadavatel mohou transformovat obraz všech snímků videa.

Uživatel bude moci zvolit transformaci obrazu pro celé video. Uživatel na stránce s úpravou videa si vybere transformaci, kterou bude moci aplikovat na dané video. Systém bude podporovat základní transformace jako rotace, změna kontrastu, apod.

F4 – Export záznamů

F4.1 – Moderátor a zadavatel mohou změnit rozlišení exportovaného videa.

Uživatel bude moci změnit rozlišení videa na rozlišení nabízené systémem. Na stránce s exportem videa bude seznam podporovaných rozlišení, uživatel si vybere jedno a systém poté při exportu provede změnu rozlišení daného videa.

F4.2 – Moderátor a zadavatel mohou změnit rychlost exportovaného videa.

Uživatel bude moci změnit snímkovou frekvenci videa. Na stránce s exportem videa bude seznam podporovaných hodnot, a při exportu se provede změna s vybranou hodnotou.

F4.3 – Moderátor a zadavatel mohou zvolit formát a kompresi videa.

Uživatel bude moci zvolit formát a kompresní metodu, kterou systém použije ke kompresi vybraného záznamu.

F4.4 – Moderátor a zadavatel mohou editovat meta-informace u videa.

Na stránce pro export záznamu bude uživatel moci navigovat na formulář s meta-informacemi k videu, kde bude moci tyto informace měnit.

F4.5 – Moderátor a zadavatel mohou video uložit na webový server nebo stáhnout.

Video bude možné uložit do úložiště dostupné na webovém serveru nebo stáhnout do vlastního počítače.

F4.6 – Moderátor a zadavatel mohou video umístit na server youtube.

Video bude možné pomocí komunikace s Youtube API umístit na server Youtube.

F5 – Ostatní

F5.1 – Zobrazení volného místa pro nahrávání na lokálním úložišti.

Webová aplikace bude uživateli zobrazovat vždy aktuální volné dostupné místo v lokálním úložišti pro nahrávání. Údaj bude zobrazen v jednotkách MiB/GiB a případně i minutách nahrávání při standardním rozlišení a formátu.

F5.2 – Moderátor a zadavatel mohou pořídit screenshot videa, okomentovat ho a následně uložit k danému testu.

Bude možné vybrat konkrétní snímek videa (v rámci sekund), ten okomentovat a uložit k danému testu. K testu bude možné přiložit několik snímků (kde každý snímek bude vždy možné utvořit pouze z videí od daného testu).

2.4.2 Nefunkční požadavky

N1 – Doba odezvy.

Rozšíření stávající aplikace bude zaručovat nízkou dobu odezvy při jeho používání. Každá elementární operace, jako spuštění videa či změna hlasitosti přehrávání nebude trvat déle než 2 sekundy.

N2 – Udržitelnost.

Frontend i backend tohoto rozšíření bude efektivně navržen a rozdělen do komponent a to tak, aby schopnost opravení nedostatků systému ovlivnila pouze tu komponentu, ve které se problém vyskytl a nikoliv celý systém.

N3 – Dostupnost.

Rozdělení rozšíření do komponent bude zajišťovat celkovou dostupnost tak, že pokud některá komponenta přestane fungovat, ostatní budou stále dostupné.

N4 – Rozšiřitelnost.

Řešení bude navrženo a vytvořeno tak, aby bylo snadno rozšiřitelné a modifikovatelné. Schopnost přidat novou funkcionalitu nebo modifikovat stávající funkcionalitu ovlivní minimální část celého systému.

N5 – Webové uživatelské rozhraní.

Webové uživatelské rozhraní bude uživatelsky přívětivé, responzivní, jednoznačné, intuitivní a jednoduché. Úprava jeho případných nedostatků bude řešena akceptačními testy.

N6 – Prohlížeče a jejich verze.

Implementované rozšíření bude fungovat nejméně na prohlížečích Firefox, Chrome a Internet Explorer, na jejich stabilních a stále podporovaných verzích.

N7 – Technologie použité pro frontend.

Frontend webové aplikace používá framework Nette, značkovací jazyk HTML(verze 5) a javascript.

N8 – Technologie pro backend.

Jako pomocný software pro úpravu záznamů i přenos po síti bude použita knihovna Libyuri.

N9 – Provázání frontendu s backendovými moduly.

Funkcionality frontendu, jako nahrávání, úprava či export videa budou realizovány moduly v backendu. Kontrakt mezi frontendem a těmito moduly bude jednoznačný, vhodně zobecněný a jednoduše rozšiřitelný.

N10 – Technologie pro databázový systém.

Technologie databázového systému použitého pro celou webovou aplikaci je PostgreSQL.

N11 – Technologie pro tvorbu modulů.

Moduly budou vytvořeny jako kompatibilní s knihovnou libyuri a případně budou použity existující moduly.

2.5 Existující komplexní řešení

Při hledání pomocného softwaru, který by řešení značně ulehčil je třeba postupovat od komplexních řešení, které jsou již funkční a otestované komunitou a které by se vypořádaly s velkým počtem funkcionalit. Pokud se nepodaří najít řešení o vhodné úrovni komplexnosti, pak se musí úroveň rozsáhlosti snižovat a poohlížet se tak po technologiích řešící pouze jednotlivé funkcionality. Přednost mají open-source řešení, které je možné libovolně použít a rozšířit.

Pro úsporu místa jsou zde uvedeny především ta nalezená řešení, která byla při rešerši velice dobře hodnocena a několik málo ostatních průměrných nebo podprůměrných adeptů je pak s těmito společně shrnuto v sekci o výběru technologií (2.6).

2.5.1 Technologie pro backend

V této části jsou uvedeny některé technologie, které byly rešerší nalezeny jako kandidátní technologie pro backendovou část aplikace. Je třeba nejprve zohlednit technologie, které byly zmíněny ve výčtu nefunkčních požadavků (2.4.2), pokud nějaké. V nefunkčních požadavcích N8 a N10 jsou zadány technologie

pro backendovou část a to jsou knihovna Libyuri pro správu videa a databázový systém PostgreSQL. V rešeršní části je ze zmíněných dvou analyzována pouze knihovna Libyuri, neboť databázový systém PostgreSQL byl již použit v předchozí bakalářské práci, je tedy zakomponován v existujícím systému a nevyskytl se závažný důvod to měnit.

2.5.1.1 Libyuri

Knihovna libyuri ([2]) je framework, poskytující prostředky pro vytváření vícevláknových aplikací zpracovávajících video a audio. Knihovna je modulární, což znamená, že jednotlivá úprava videa je zpracovávána jedním modulem (například ořez videa). Zpracování videa a audia v Libyuri je pak možné pomocí vytvoření orientovaného grafu, kde jednotlivé fáze zpracování (moduly) jsou reprezentovány jako uzly grafu a datová propojení mezi moduly jako orientované hrany grafu. Modulů v této knihovně je již nepřehledná spousta, počínaje moduly, které přebírají obrazová data z různých zdrojů (například kamery, V4L zařízení, Decklink zařízení, ...) přes moduly pro zpracování obrazových i zvukových rámců a podobně. Knihovna je také velice snadno rozšiřitelná a díky dobré dokumentaci a šablonám pro vytvoření modulů je opravdu snadné rozšířit tento mocný nástroj o téměř libovolnou funkcionalitu.

Licence libyuri je dle ([2]) pod modifikovanou BSD licenci přičemž moduly používající technologie jiných autorů jsou pod licencemi dalšími, vesměs to však jsou open-source licence, což je další plus.

2.5.1.2 Ultragrid

Ultragrid (odkazem [3]) je software vyvíjený lidmi ze sdružení CESNET z brněnské laboratoře SITOLA, pro nízko-latenční přenos video dat po síti. Software podporuje mnoho známých video standardů, jako například PAL/NTSC, HD, 2K i 4K. Pro zobrazování dat využívá technologie OpenGL a SDL. UltraGrid používá streamy videa bez komprese, nebo s velmi malou kompresí k zaručení až 8K rozlišení s až 100ms latencí mezi koncovými body. Software se již používá v mnoha oblastech jako jsou kolaborující prostředí, lékařská kinematografie, různé vzdělávací aktivity a jiné. Je schopný přenosového módu tzv. dual-link, což je ve zkratce posílání dat dvěma různými cestami pro zajištění větší šířky pásma. Ultragrid je dle ([4]) open-source software pod BSD licenci.

Tento software je také možné použít, jako komplexní řešení neboť podobně jako Libyuri funguje na základě propojování uzlů, které pracují s videem. Ultragrid má oproti Libyuri o trochu lepší dokumentaci, ale na druhou stranu má o mnohem menší počet modulů což je mnohem větší slabina než zmíněná dokumentace. Navíc se nepodařilo nalézt srozumitelný návod či šablonu pro vytvoření vlastních modulů.

2.5.1.3 FFmpeg

FFmpeg (odkazem [5]) je jeden z nejlepších a nejznámějších open-source multimediálních frameworků vůbec, který je schopný kódovat, dekódovat, konvertovat, streamovat, filtrovat i přehrávat nejen audio i video záznamy. Tento software podporuje téměř všechny známé formáty záznamů od zastaralých, po ty nejnovější. Je také velmi dobře portabilní, neboť se dá zkompileovat a spustit na distribucích operačních systémů typu Linux, Mac OS X, Microsoft Windows, BSD, Solaris, a tak podobně.

Díky známosti, spolehlivosti a komunitě tohoto frameworku by byla chyba jej nepoužít právě při zpracování videa. Naštěstí výše zmíněná softwarová řešení jako Libyuri a Ultragrid tento framework mají v sobě již zakomponován (pokud jsou ovšem zkompileovány s podporou pro FFmpeg).

2.5.1.4 GStreamer

GStreamer ([6]) je multimediální knihovna pro konstruování grafu z komponentů pro práci s multimediálním obsahem (podobně, jako Libyuri (2.5.1.1)). Knihovna je kompatibilní se všemi majoritními Linux, Windows, Mac OS X, iOS, stejně tak jako většina systémů BSD, také komerční jako Unixy, Solaris, ale také Android i Symbian. GStreamer má rozsáhlou dokumentaci ([7]) a ještě rozsáhlejší komunitu, takže je velká pravděpodobnost, že potenciální problém je již někde na diskuzním fóru vyřešen. Knihovna je modulární a je velice snadné napsat plugin pomocí poskytovaného generického rozhraní.

GStreamer je open-source a je šířen pod LGPL licencí.

2.5.1.5 Open Broadcaster Software

OBS neboli Open Broadcaster Software je dle ([8]) software pro nahrávání a streamování videa vyvíjený dobrovolníky z celého světa. OBS je distribuovaný open-source software pod GPLv2 licencí. Tento software umožňuje spolehlivé nahrávání videa i audia v reálném čase a jeho následné zpracování. Je zde také možné nahrávat současně z více zdrojů, jako například monitoru, kamery, speciální tzv. „grabovací“ grafické karty, webkamery a dalších.

OBS má dobrou dokumentaci (vizte [9]), ve které je mimo jiné zmíněno, že je možné software lehce přizpůsobit pomocí skriptovacích jazyků Lua či Python. Většina funkcionalit v OBS je přidána pomocí plugin modulů, které jsou reprezentovány typicky dynamickými knihovnami nebo skripty. Jelikož se jedná o open-source software, pak je možné moduly ve formě pluginů i libovolně vytvářet na což je v dokumentaci taktéž podrobný návod. Nevýhodou tohoto řešení je velice omezená manipulace s binární podobou programu z prostředí příkazové řádky, což je v případě potenciální automatizace nežádoucí.

2.5.2 Technologie pro frontend

Stejně jako v úvodu technologií pro backend (2.5.1) je při rešerši žádoucí respektovat frontendové technologie dohodnuté nefunkčními požadavky (2.4.2). V tomto případě je nefunkčním požadavkem N7 specifikován výčet tří technologií, které jsou framework Nette jazyku PHP, dále značkovací jazyk HTML5 a skriptovací jazyk JavaScript. Framework Nette není třeba zahrnout do porovnání technologií, protože je v něm napsán téměř celý systém předchozí bakalářské práce a bude tedy použit i nadále.

2.5.2.1 HTML, Javascript

Technologie HTML5 i JavaScript jsou nejběžněji používané technologie pro vývoj frontendu webových aplikací a v tomto kontextu se tedy nedají brát jako komplexní řešení pro nějakou podmnožinu specifikovaných úkolů a není tedy důvod je porovnávat s ostatními technologiemi pro frontend, které jsou ostatně na nich postaveny.

Všechny řešení analyzované níže se však snaží vyřešit jeden komplexní problém, což je webový video přehrávač/editor. Tento problém lze vyřešit nalezením komplexního řešení, tedy nějakého frameworku, nebo také napsat vlastní přehrávač od píky s pomocí těchto technologií. Nevýhodou tvoření od píky je, že se bude implementovat řešení, které již existuje, což ovšem může snadno vyvážit výhoda toho, že struktura řešení bude známá, a bude nejsnáze modifikovatelné i rozšiřitelné.

2.5.2.2 Video.js

Video.js je open-source knihovna pro práci s videem na webu (vizte [10]). Tato knihovna je vlastně obalení pro nativní webový videopřehrávač v technologii HTML5, avšak s přidáním užitečných funkcionalitami. Jelikož je tento software open-source a byl stavěný tak, aby byl dobře rozšiřitelný, tak se komunitě v průběhu času podařilo vyvinout nemalý počet užitečných pluginů (více zde [11]). Knihovna má výbornou dokumentaci ([12]), kde je mimo jiné i popsáno, jak lze napsat vlastní plugin, což je zvládnutelné i se základními znalostmi jazyku JavaScript.

Použití této knihovny je velice vhodné pro frontendovou část řešení této práce. Díky již zmíněným přednostem je lepší a snadněji rozšiřitelný než samotný přehrávač vestavěný v jazyku HTML5. Knihovnu je možné buď stáhnout a v případě potřeby i upravit před použitím nebo použít CDN Fastly ([13]), která poskytuje hosting všem nezbytným souborům knihovny Videojs. Knihovna je kompatibilní s prohlížeči se zabudovaným HTML5 nebo případně i s technologií Flash. Podporovanými prohlížeči je většina dnes nejčastěji používaných jako například Firefox verze 3.5 a vyšší, Internet Explorer 6 a vyšší, Chrome verze 3 a vyšší, Opera a další (více na wiki projektu [14]).

2.5.2.3 Afterglow

Afterglow je open-source HTML5 videopřehrávač pod MIT licencí. Dle ([15]) je jednoduše konfigurovatelný plně responzivní a je kompatibilní s širokou škálou webových prohlížečů a zařízení. Přehrávač má kvalitní dokumentaci ([16]) s ukázkovými příklady. Přehrávač je schopen přehrát Youtube i Vimeo video. Je možné mu dodat více stejných videí avšak o jiném rozlišení i v jiném formátu a uživatel si poté může v přehrávači ze zadaných formátů a rozlišení vybírat. Afterglow je díky obalení prohlížečů nativní HTML5 technologie kompatibilní s většinou dnes používaných prohlížečů a je taktéž k dispozici pomocí CDN jsDelivr.

Tato nadstavba nativního HTML5 videopřehrávače má však několik nevýhod. Je to velice jednoduchá nadstavba a neumožňuje dostatek nových funkcionalit nad nativním HTML5 přehrávačem, než které by bylo možno dopsat ručně zanedlouho. Komunita u tohoto softwaru není zdaleka tak silná jako u zmíněného Video.js a tak ani počet pluginů není tak velký (není zde například možné jednoduše udělat playlist videí).

2.5.2.4 Movie Masher

Movie Masher ([17]), přesněji knihovna moviemasher.js ([18]) je knihovna jazyku JavaScript pro editaci videa a audia v prohlížeči v reálném čase. Knihovna je open-source s licencí Mozilla Public Licence v2. Tento editor pracuje s HTML5 videopřehrávačem. Editor umožňuje vizuální kompozici videoklipů s přechody, mix audia za použití API WebAudio, Undo a Redo příkazy pro vrácení se zpět a vpřed v historii provedených úprav a další. Jednotlivé efekty jsou pak například vložení textu do videa, zkrácení videa (oříznutí pouze odzadu), změna barev, kontrastu a podobně. Movie Masher je možné buď stáhnout a nainstalovat na vlastním serveru nebo použít technologii Microsoft Azure ([19]), která jej podporuje avšak její hosting je placený.

Oproti požadovaným funkcionalitám tento video editor několik vymožeností postrádá (skládání videí vedle sebe do jednoho obrazu, ořez videa z obou stran, změna rozlišení, změna formátu a jiné). Editor také oficiálně nepodporuje pluginy ([19]) a v dokumentaci nebyl nalezen návod nebo šablony pro jeho snadné rozšíření.

2.6 Výběr technologií

Technologie nashromážděné rešerši je nutné zredukovat, neboť čím menší počet jich použijeme a čím větší část požadavků pokryjí, tím lépe. Nejprve je uvedena porovnávací metodika, která obsahuje výčet funkcí a charakteristik pro výběr technologií a pomocí priorit těchto charakteristik jsou pak technologie porovnány formou tabulky (...TODO ref na tabuli).

2.6.1 Porovnávací metodika

Aby výběr těch nejvhodnějších technologií proběhl kvalitně, je zapotřebí k tomu použít určitou metodiku, tedy souhrn postupů, které pokryjí proces výběru. Jednou z těchto metod je stanovení výčtu obecných vlastností a funkcionalit softwaru, jejichž úroveň a kvalita se změří na nalezených technologiích. Dalším postupem je shrnutí nashromážděných hodnocení, zde například formou tabulek (TODO REF TABLE) a (TODO REF TABLE) a následné vyhodnocení, jehož výstupem bude redukovaný výčet technologií, které ve výběru „uspěly“ a budou použity v implementační části této práce pro frontend a pro backend.

Nejprve je tedy výčtem uveden seznam obecných prioritních charakteristik softwaru, přičemž u každé z nich je definováno, co se pod ní, v kontextu této práce, myslí.

Licence

Preferováno open-source řešení nad proprietárním. Prioritní licence jsou například GPLv2, MIT, Apache či WTFPL.

Dokumentace

Je technická i uživatelská dokumentace dostatečně specifická? Souhlasí verze dokumentace s verzí produktu a koresponduje s jejím aktuálním stavem?

Podpora

Je produkt stále ještě vyvíjen a alespoň některá z jeho verzí aktivně podporována? Existuje možnost snadné zpětné vazby od uživatele technologie k jejím udržovatelům?.

Portabilita

Je technologie multiplatformní? Je možné ji bez větších problémů zprovoznit na novějších verzích operačních systémů Windows i na několika různých distribucích systému Linux?

Rozšiřitelnost

Je možné technologii snadno rozšiřovat a upravovat, pokud to licence umožňuje? Je například možné jej rozšířit o moduly s požadovanou funkcionalitou, pokud je zatím technologie nepokrývá?

Integrace

Zda a jak je možné technologii integrovat do systému této práce. Možnost integrace a kooperace technologie s infrastrukturou SAGE.

Komunita

Velikost uživatelské komunity vybrané technologie. Existence dalších zdrojů informací o produktu např. webové stránky obsahující informace jiné než v technické dokumentaci.

2. ANALÝZA

	Libyuri	Ultragrid	FFmpeg	GStreamer
Licence				
Dokumentace				
Podpora				
Portabilita				
Rozšiřitelnost				
Integrace				
Komunita				
Zkušenosti				

Tabulka 2.1: Porovnání technologií pro backend

Zkušenosti

Má autor této práce nějaké zkušenosti s danou technologií, které by mu umožnily snadnější, rychlejší a efektivnější implementaci řešení?

Po porovnání všech nashromážděných technologií pomocí specifikovaných charakteristik bylo dosaženo výsledku, který je popsán tabulkou TODO NUM TABLE pro backendové technologie a tabulkou TODO NUM TABLE pro frontendové technologie.

2.6.2 Vybrané technologie

...nette, html5, javascript, yuri,... ps Libyuri byla specifikována v jednom z NefPoz nahore

Návrh

blablabla praktiky vzory, blabla pak frontend a backend

Návrh je rozdělen do podkapitol pro backend, kde je uvedeno, jaká je struktura té části aplikace, která je uživateli odstíněna, ale která také zajišťuje většinu funkcionalit.

3.1 Praktiky a vzory

...SOLID, architektonické vzory, návrhové vzory (GoF, GRASP), ...

3.2 Architektura systému

3.3 Frontend

3.3.1 Návaznost na současný systém

3.3.2 Přehrávání a úprava záznamů

3.3.3 UX a UI

3.4 Backend

Implementace

4.1 Prostředí a kompatibilita

4.2 Použití knihovny Libyuri

..TODO - doplnit z hotoveho dokumentu o yuri

4.3 Vybrané zajímavé funkce

4.4 Instalační a uživatelská příručka

Testování

5.1 Integrační testování

5.2 Akceptační testování

5.2.1 Probíhající usability testování

Závěr

Literatura

- [1] Solanská, K.: Children Usability Lab - aplikace pro správu laboratoře. 2016.
- [2] Yuri [projects.iim.cz]. 2013, [cit. 2018-03-11]. Dostupné z: <http://projects.iim.cz/yuri>
- [3] Ultragrid [online]. [cit. 2018-03-12]. Dostupné z: <http://www.ultragrid.cz/>
- [4] Ultragrid - wiki [online]. [cit. 2018-03-12]. Dostupné z: <https://github.com/CESNET/UltraGrid/wiki>
- [5] FFmpeg [online]. [cit. 2018-03-13]. Dostupné z: <https://www.ffmpeg.org/about.html>
- [6] GStreamer [online]. [cit. 2018-03-14]. Dostupné z: <https://gstreamer.freedesktop.org/>
- [7] GStreamer - Documentation[online]. [cit. 2018-03-14]. Dostupné z: <https://gstreamer.freedesktop.org/documentation/>
- [8] Open Broadcaster Software [online]. [cit. 2018-03-13]. Dostupné z: <https://obsproject.com>
- [9] Open Broadcaster Software - Documentation [online]. [cit. 2018-03-13]. Dostupné z: <https://obsproject.com/docs/>
- [10] VideoJS - The Player Framework [online]. [cit. 2018-03-12]. Dostupné z: <https://videojs.com/>
- [11] VideoJS - Community plugins [online]. [cit. 2018-03-12]. Dostupné z: <http://videojs.com/plugins/>

- [12] VideoJS - Documentation [online]. [cit. 2018-03-12]. Dostupné z: <http://docs.videojs.com/>
- [13] Fastly - Content delivery and image optimization [online]. [cit. 2018-03-12]. Dostupné z: <https://www.fastly.com/products/web-and-mobile-performance>
- [14] VideoJS - Wiki, browser compatibility [online]. [cit. 2018-03-12]. Dostupné z: <https://github.com/videojs/video.js/wiki>
- [15] Afterglow player [online]. [cit. 2018-03-14]. Dostupné z: <http://afterglowplayer.com/>
- [16] Afterglow player - Documentation [online]. [cit. 2018-03-14]. Dostupné z: <http://docs.afterglowplayer.com/>
- [17] Movie Masher - Website [online]. [cit. 2018-03-14]. Dostupné z: <http://moviemasher.com>
- [18] Movie Masher - moviemasher.js [online]. [cit. 2018-03-14]. Dostupné z: <https://github.com/moviemasher/moviemasher.js>
- [19] Movie Masher - Microsoft Azure [online]. [cit. 2018-03-14]. Dostupné z: <https://azuremarketplace.microsoft.com/en-gb/marketplace/apps/moviemasher.moviemasher>
- [20] Kočíčka, P.; Blažek, F.: *Praktická typografie*. Brno: Computer Press, 2004.
- [21] Duncan, D.: GRASP Patterns [online]. 2012, [cit. 2018-03-08]. Dostupné z: <http://www.cs.colorado.edu/~kena/classes/5448/f12/presentation-materials/duncan.pdf>
- [22] SAGELab – Síťová multimediální laboratoř [online]. [cit. 2018-03-08]. Dostupné z: <https://sagelab.cesnet.cz/>
- [23] Faistaver, P.: NEC-Controller [software]. 2015. Dostupné z: <http://dx.doi.org/10.6084/m9.figshare.1408922>
- [24] The Linux man-pages project - ldd [online]. 2008, [cit. 2015-04-01]. Dostupné z: <http://man7.org/linux/man-pages/man1/ldd.1.html>
- [25] Padala, P.: Ncurses introduction [online]. 2001, [cit. 2015-04-05]. Dostupné z: <http://tldp.org/HOWTO/NCURSES-Programming-HOWTO/intro.html>
- [26] Mitchell, M.; Oldham, J.; Samuel, A.; aj.: *Advanced Linux Programming*. První vydání, ISBN 07-3571-043-0.

-
- [27] NEC Display Solutions: External Control Codes for P.V.X-series [online]. 2013, [cit. 2015-04-02]. Dostupné z: http://www.necddisplay.com/documents/UserManuals/External_Control_P.V.X-series.pdf
 - [28] The GNU C Library - dlfcn.h [online]. 2008, [cit. 2015-04-01]. Dostupné z: https://fossies.org/dox/glibc-2.21/dlfcn_2dlfcn_8h.html
 - [29] NEC Display Solutions: NaViSet Administrator 2TM[online]. 2014, [cit. 2015-05-03]. Dostupné z: <http://www.necddisplay.com/support-and-services/naviset-administrator/Overview>
 - [30] Padala, P.: Ncurses programming HOW-TO [online]. 2005, [cit. 2015-04-06]. Dostupné z: <http://tldp.org/HOWTO/NCURSES-Programming-HOWTO/>
 - [31] Smotlacha, V.: BI-PSI - Přednáška č.6 - Transportní vrstva. TCP, UDP. 2015, [cit. 2015-05-03].
 - [32] Úřad pro technickou normalizaci, metrologii a státní zkušebnictví: *ČSN ISO 690 Informace a dokumentace – Pravidla pro bibliografické odkazy a citace informačních zdrojů*. 2011.

Seznam použitých zkratk

GUI Graphical user interface

HTML Hypertext markup language

Obsah přiloženého CD

	readme.txt.....	stručný popis obsahu CD
	exe	adresář se spustitelnou formou implementace
	src	
	impl.....	zdrojové kódy implementace
	thesis	zdrojová forma práce ve formátu L ^A T _E X
	text	text práce
	thesis.pdf	text práce ve formátu PDF
	thesis.ps	text práce ve formátu PS