

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačů



Webová aplikace pro tvorbu audiovizuálních prezentací v multiprojekčním systému

Bakalářská práce

Vít Říha

Vedoucí práce: Ing. Ivo Malý, Ph.D.
Bakalářský program: Softwarové Inženýrství a Technologie

V Praze, květen 2021

Vedoucí práce:

Ing. Ivo Malý, Ph.D.
Katedra počítačové grafiky a interakce
Fakulta Elektrotechnická
České Vysoké Učení Technické v Praze
Karlovo náměstí 13
121 35 Praha 2
Česká Republika

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Říha** Jméno: **Vít** Osobní číslo: **465826**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra počítačů**
Studijní program: **Softwarové inženýrství a technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Webová aplikace pro tvorbu audiovizuálních prezentací v multiprojekčním systému

Název bakalářské práce anglicky:

Web application for creation of audiovisual presentations in multi-projection system

Pokyny pro vypracování:

Seznamte se s projektem 5D projekce připravovaným pro Muzeum hlavního města Prahy (MMP) a s prvním prototypem jeho ovládacího systému [1]. Zaměřte se zejména na webovou aplikaci pro řízení a správu obsahu a způsob tvorby jednotlivých audiovizuálních prezentací.

Navrhněte novou verzi webové aplikace, která se zaměří na proces tvorby jednotlivých prezentací tvořených audiovizuálními soubory. Zaměřte se nejen na sestavení prezentací s existujícími audiovizuálními materiály, ale i na možnost jednoduchých úprav videa, např. doplnění titulků, zkrácení videa a podobně. Rozšiřte také prezentace o kotvy, které vytvoří nové navigační body uvnitř jednotlivých stop, tak aby nebylo nutné tyto stopy rozdělovat v externích editorech. Webovou aplikaci implementujte pomocí vhodných nástrojů a knihoven. Aplikaci otestujte s alespoň 2 různými prezentacemi.

Seznam doporučené literatury:

1. O. Trojan, Řídící systém pro multiprojekci, Diplomová práce, České vysoké učení technické v Praze, 2020.
2. T. Lowdermilk, User-Centered Design, O'Reilly Media, 2013.
3. P. Hong, Practical web design: Learn the fundamentals of web design with HTML5, CSS3, bootstrap, jQuery, and vue.js, Packt Publishing Ltd, 2018.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Ivo Malý, Ph.D., katedra počítačové grafiky a interakce FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **12.02.2021**

Termín odevzdání bakalářské práce: **21.05.2021**

Platnost zadání bakalářské práce: **30.09.2022**

Ing. Ivo Malý, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Prohlášení

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze, květen 2021

.....
Vít Říha

Abstrakt

Práce se zabývá analýzou aplikace pro správu multiprojekčního systému Muzea Hlavního Města Prahy a následným návrhem a implementací nového uživatelského prostředí se zaměřením na uživatelskou přívětivost. Zároveň tato práce rozšíří funkcionalitu tohoto systému o nové prvky, které usnadní ovládání aplikace a umožní tvořit audiovizuální prezentace s titulky vytvořenými v rámci aplikace.

Klíčová slova: UI, uživatelské prostředí, front-end, webová aplikace, Vue.js, Node.js, multiprojekční systém

Abstract

The thesis deals with analysis of application for management of a multi-projection system in The City of Prague Museum and subsequent design and implementation of new user interface with focus on user friendliness. This project will also extend functionality of the system by adding new features, that will facilitate use of the application and allow creating audiovisual presentation with subtitles made within the application.

Keywords: UI, user interface, front-end, web application, Vue.js, Node.js, multi-projection system

Poděkování

Děkuji mé rodině za morální podporu při práci na tomto projektu. Největší poděkování však patří bezesporu mému vedoucímu práce, panu Ing. Ivovi Malému, Ph.D. za jeho trpělivost, obětavost a příkladné vedení.

Obsah

Abstrakt	vi
Poděkování	vii
1 Úvod	1
1.1 Motivace	1
1.2 Cíle	1
2 Stávající řešení	2
2.1 Technologie	2
2.2 Funkcionalita	2
2.3 Struktura	3
3 Analýza	4
3.1 Uživatelské role	4
3.1.1 Moderátor	4
3.1.2 Administrátor	4
3.2 Funkční požadavky	4
3.3 Případy užití	5
4 Návrh řešení	6
4.1 Návrh uživatelského rozhraní	6
5 Příprava pro implementaci	14
5.1 Framework Vue.js	14
5.1.1 Struktura projektu	15
6 Implementace	16
6.1 Použité knihovny	16
6.1.1 Axios	16
6.1.2 Vuex	17
6.1.3 Vue Router	18
6.1.4 Node sass	19
6.1.5 Element Plus	19
6.2 Klíčové vlastnosti	19
6.2.1 Správa prezentací	19
6.2.2 Správa projekce	20
6.2.3 Správa uživatelů a logy	20
6.2.4 Nová funkcionalita	21
7 Evaluace	25

<i>OBSAH</i>	ix
8 Závěr	26
8.1 Budoucnost	26
A Zprovoznění aplikace	27
Bibliografie	28

Kapitola 1

Úvod

1.1 Motivace

Tato bakalářská práce navazuje na diplomovou práci pana Ing. Ondřeje Trojana. Hlavním cílem jeho práce bylo vytvořit multiprojekční systém pro Langweilův model Prahy v Muzeu hlavního města Prahy. K tomuto systému zároveň připravil mobilní aplikaci pro ovládání projekce a webovou aplikaci, která má sloužit především pro správu systému, ale i jako záložní systém pro jeho ovládání. Ze závěru jeho práce vyplývá, že koncoví uživatelé jsou vcelku spokojeni se systémem a jeho funkcemi, ale webové uživatelské prostředí jim připadá nepřehledné.

1.2 Cíle

Cílem práce je analýza stávajícího řešení, návrh nového uživatelského prostředí se zaměřením na uživatelskou přívětivost a následná implementace nového front-endu systému.

Kapitola 2

Stávající řešení

Tato kapitola je věnována analýze stávajícího řešení podle popisu v diplomové práci [1] pana Ing. Ondřeje Trojana.

2.1 Technologie

Webová aplikace je postavena na knihovně Bootstrap a jQuery. Pro uložení uživatelské relace používá JWT token, který prohlížeč ukládá ve formě Cookie po přihlášení uživatele. Kód Javascriptu je rozdělený do modulů pomocí import/export statementů.

Backend je implementovaný pomocí Spring Boot frameworku a PostgreSQL databáze. Systém běží na lokální síti bez nutnosti přístupu k internetu. Proto nebylo nutné implementovat složité bezpečnostní prvky. Přesto je však implementována základní ochrana, konkrétně proti SQL injection a XSS. Mobilní i webová aplikace implementují autorizační a autentizační mechanismy.

2.2 Funkcionalita

Úvodní stránka aplikace zobrazuje krátký popis a tlačítko pro přihlášení uživatele. Uživatel se po přesměrování přihlásí pomocí platných přihlašovacích údajů. Po přihlášení je uživatel přesměrován na hlavní stránku aplikace, kde může sledovat seznam nakonfigurovaných klientů, jejich stav a přehrávání. Dále je zobrazena aktuálně přehrávaná stopa, čas přehrávání a seznam stop ve frontě.

Navigace se nachází v pravém horním rohu okna prohlížeče, kde jsou zobrazena tlačítka podle role přiřazené přihlášenému uživateli, přepínač jazyka a tlačítko pro odhlášení. Admin i běžný uživatel mohou ovládat projekci, a to základními příkazy play, nebo pause, ale i pokročilejšími, jako přechod do/z autonomního režimu, nebo změna Presentace. Admin může spravovat uživatelské účty aplikace (vytváření/úprava účtů, reset hesla, přiřazení role, aktivace/deaktivace účtu), prohlížet logy související s chodem systému (např. záznamy příkazů pro přehrání, pauzu, apod.), vytvářet Presentace, nebo nahrávat nové videosobory. Dále může také

provést restart instance multimediálního přehrávače na klientských zařízeních pomocí tlačítek na obrazovce s ovládacími prvky.

2.3 Struktura

Vzhledem k tomu, že se jedná o multiprojekční systém, je zapotřebí mít možnost zobrazovat různé stopy na jednotlivých zobrazovacích zařízeních. Abychom snadno docílili synchronizovaného zobrazování různých stop, byla navržena adresářová struktura, ve které se nachází jednotlivé videosoubory.

Základní jednotkou je videosoubor. Ty se skládají do Seznamu stop (v původní terminologii Display Track), což je seznam videí, která se budou zobrazovat na všech klientských zařízeních zároveň (počet videí v seznamu stop by tedy měl být stejný jako počet aktivních klientských zařízení). Tyto Seznamy stop jsou vloženy do Prezentací (původně Display Set), čímž se vytvoří ucelený seznam po sobě jdoucích videí na každé obrazovce. Tyto adresáře jsou pojmenovány pouze čísly - id jednotlivých Prezentací a Seznamů stop.¹ Jednotlivé videosoubory jsou potom pojmenované následovně *1-2-nazevVidea.mp4*, v tomto případě se jedná o video s názvem *nazevVidea*, které se nachází v Prezentaci s id 1 a v Seznamu stop s id 2.

Nevýhoda tohoto přístupu je především v tom, že pokud uživatelé budou chtít použít jedno video ve více Prezentacích nebo Seznamech stop, bude potřeba ukládat stejné soubory ve více různých adresářích (redundantně). Naopak výhodou adresářové struktury je to, že uživatel technicky může upravovat jednotlivé Prezentace a Seznamy stop manuálně v jednotlivých adresářích vytvářením, změnou, nebo mazáním složek a souborů.

¹id Prezentací jsou unikátní v systému, id Seznamu stop je unikátní v rámci Prezentace.

Kapitola 3

Analýza

3.1 Uživatelské role

Uživatelé budou rozděleni do dvou typů uživatelů: Administrátor a Moderátor. Administrátor bude nadstavbou Moderátora - tedy bude mít všechna ta práva, která bude mít Moderátor a k tomu bude mít přidána další práva.

3.1.1 Moderátor

Moderátor bude mít pouze práva spojená s ovládáním projekce - výběr Prezentací, play, pause, stop, přepínání autonomního režimu.

3.1.2 Administrátor

Administrátor bude mít plný přístup ke všem funkcionalitám aplikace - tedy správa uživatelských účtů, správa videosouborů, Prezentací a Seznamů stop, apod.

3.2 Funkční požadavky

Funkční požadavky definují funkcionalitu, kterou musí výsledná webová aplikace poskytovat. Nová aplikace musí umožnit všechno to, co umožňuje ta stávající, tedy:

- přihlášení/odhlášení uživatele
- ovládání projekce
 - play, pause, stop
 - změna Prezentace
 - přepínání autonomního režimu
- nahrání videa na server
- odstranění videa ze serveru
- vytváření, úpravu a odstranění Prezentací a Seznamů stop

- správu uživatelských účtů
- zobrazení systémových logů

Součástí nové aplikace budou také následující nové funkcionality:

- vytváření titulků pro videosoubory v rámci webové aplikace
- vytváření kotev, na které lze přeskakovat v rámci Presentace

3.3 Případy užití

Aplikace bude primárně určena pro správu videosouborů a vytváření struktury na těmito soubory. Aplikace bude intuitivní a uživatele bude provázet jednotlivými kroky při správě systému. Bude například existovat více než jeden způsob, jak nahrát videosoubor. Uživatel bude moci tímto krokem začít - tedy nejprve nahrát videosoubor a ten následně buď vložit do existujícího Seznamu stop, nebo pro tento soubor vytvořit nový Seznam stop v existující Presentaci, nebo vytvořit i novou Presentaci.

Zároveň ale bude uživatel moci začít z druhé strany - tedy tím, že vytvoří novou Presentaci, v něm nový Seznam stop a poté do tohoto Seznamu stop nahraje nové video. Z toho plyne, že na obrazovce zobrazující Seznam stop musí být možnost nahrát nové video - tedy i že do existujícího Seznamu stop, ve kterém už může být jedno nebo více videí přítomno, uživatel bude mít možnost nahrát nové video.

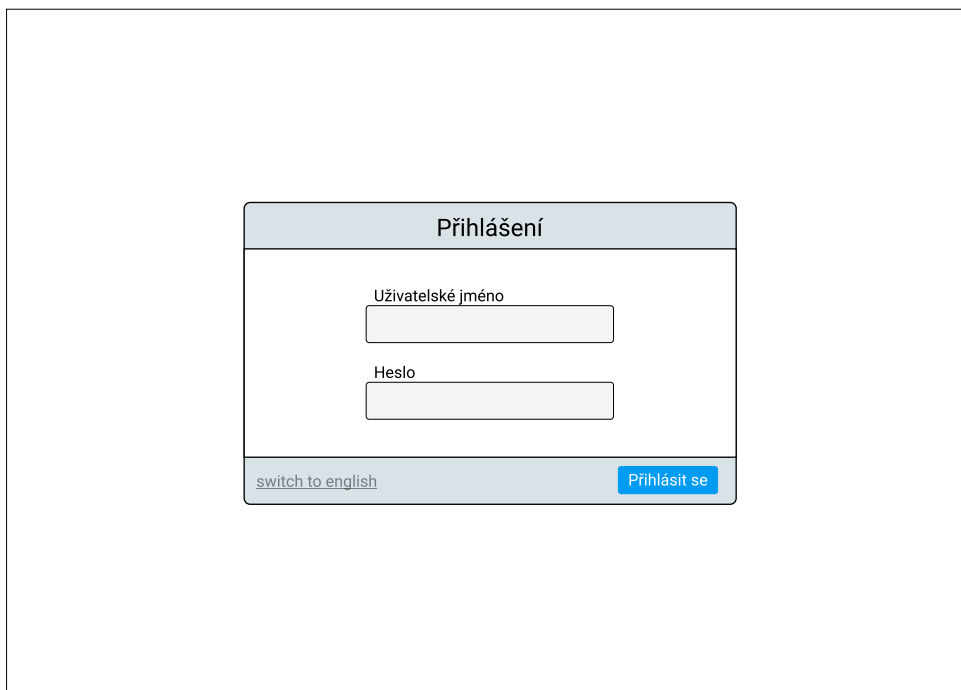
Systém také bude podporovat snadné přidání jedno videa do více Seznamů stop. Uživatel toho bude moci docílit výběrem videa z jiného Seznamu stop, ale také ihned po nahrání nového videa. Naopak bude také možné přidat do jednoho seznamu stop více souborů naráz. Uživatel tedy bude mít možnost připravit si soubory předem a potom jen vybrat, do kterého seznamu stop tato videa patří.

Kapitola 4

Návrh řešení

4.1 Návrh uživatelského rozhraní

Pro vytvoření vizuálního návrhu prostředí aplikace jsem využil nástroj Figma¹. Následuje náhled jednotlivých navržených obrazovek. Na obrázku 4.1 vidíme jednoduchou přihlašovací obrazovku, která se zobrazí uživateli, pokud se pokusí přistoupit na doménu bez vygenerovaného JWT tokenu. Obsahuje pouze dvě pole pro zadání přihlašovacích údajů a tlačítko pro přihlášení. Na



Obrázek 4.1: Přihlašovací obrazovka

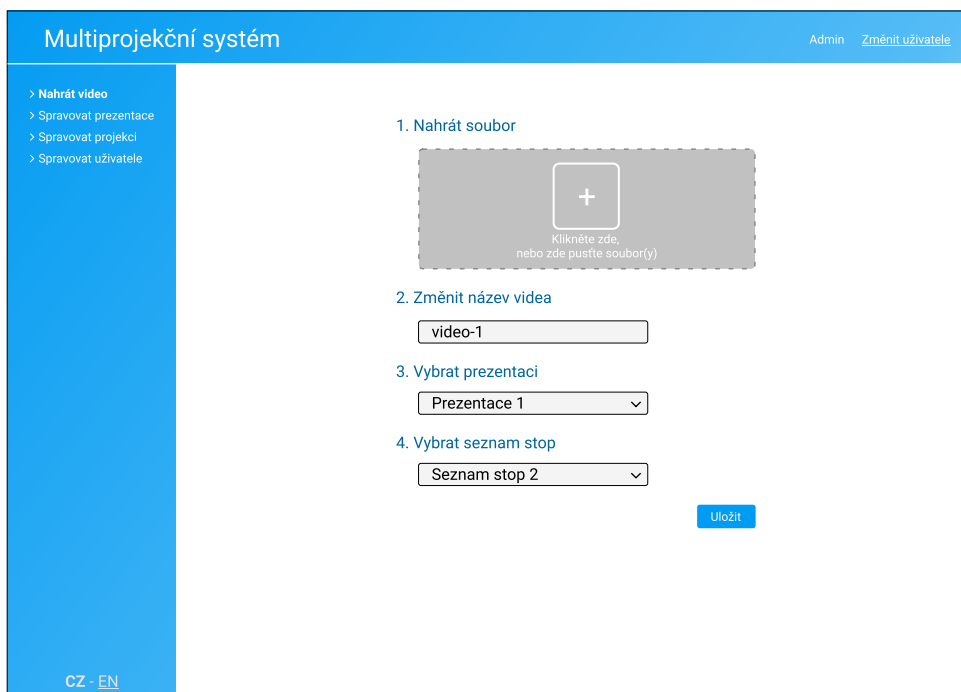
této obrazovce je také možné přepínat jazyk aplikace mezi angličtinou a češtinou.

Podle principu blízkosti (principle of proximity [2]) by měly být prvky, které spolu souvisejí, umístěny blízko u sebe, aby byly pro uživatele přehledné. S výjimkou přihlašovací obrazovky je na každé obrazovce zobrazeno menu po levé straně s navigačními odkazy. Ve spodní části levého

¹K dispozici na webové adrese <https://www.figma.com/>

panelu je přepínač jazyka. V pravé horní části obrazovky se nachází tlačítko pro odhlášení uživatele a uživatelské jméno aktuálně přihlášeného uživatele.

Možnost nahrání videa je klíčová pro fungování aplikace. Na obrázku 4.2 je zobrazena obrazovka, která toto umožňuje. Uživatel vybere soubor ve svém počítači a metodou drag-and-

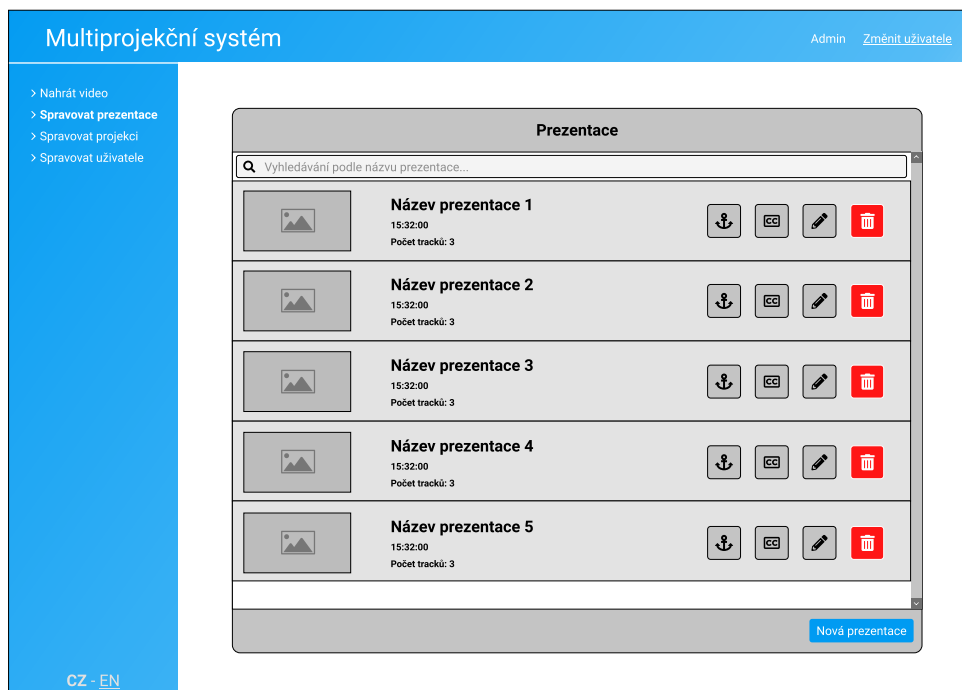


Obrázek 4.2: Nahrání videa

drop soubor připraví pro nahrání. Případně může pro přípravu souboru využít dialogové okno prohlížeče kliknutím do oblasti drag-and-drop. Dále má uživatel možnost změnit název nahraného souboru a přiřadit ho do prezentace a seznamu stop. Nahrávání souboru započne po kliknutí na tlačítko Uložit. V průběhu nahrávání bude zobrazena indikace v podobě spinneru, dokud nebude nahrávání dokončeno.

Obrázek 4.3 zobrazuje návrh obrazovky, na které je seznam vytvořených prezentací. Obsahuje vyhledávací pole, které umožní filtrovat prezentace podle jejich názvu. U každé prezentace má uživatel možnost vytvořit kotvy, přidat titulky, spravovat seznam stop příslušné prezentace, nebo prezentaci smazat.

Na obrázku 4.4 vidíme editor titulků. Obrázek použitý pro náhled videa je převzatý z webové stránky [3] Muzea hlavního města Prahy. Ve vrchní části této obrazovky vidíme dvě možnosti v podobě radio buttonů. Při výběru „Pro celou prezentaci“ se titulky aplikují na každý seznam stop v prezentaci, čímž se zajistí zobrazování stejných titulků na každé obrazovce. Pokud uživatel vybere možnost „Každá obrazovka zvlášť“, budou se titulky vytvářet pouze pro Seznam stop vybraný v položce Náhled. Pod těmito přepínači je zobrazen náhled videa, ve kterém se budou po najetí myši zobrazovat ovládací prvky videa - především možnost video přehrát, nebo pozastavit. Pod náhledem videa se nachází dvě časové osy. V té vrchní je zobrazena celá časová osa Prezentace. Na této ose může uživatel vybrat světlým zvýrazněním oblast, která je zobrazena na spodní ose. V obrázku vidíme, že je vybrána oblast o velikosti 30 sekund, tedy spodní osa má



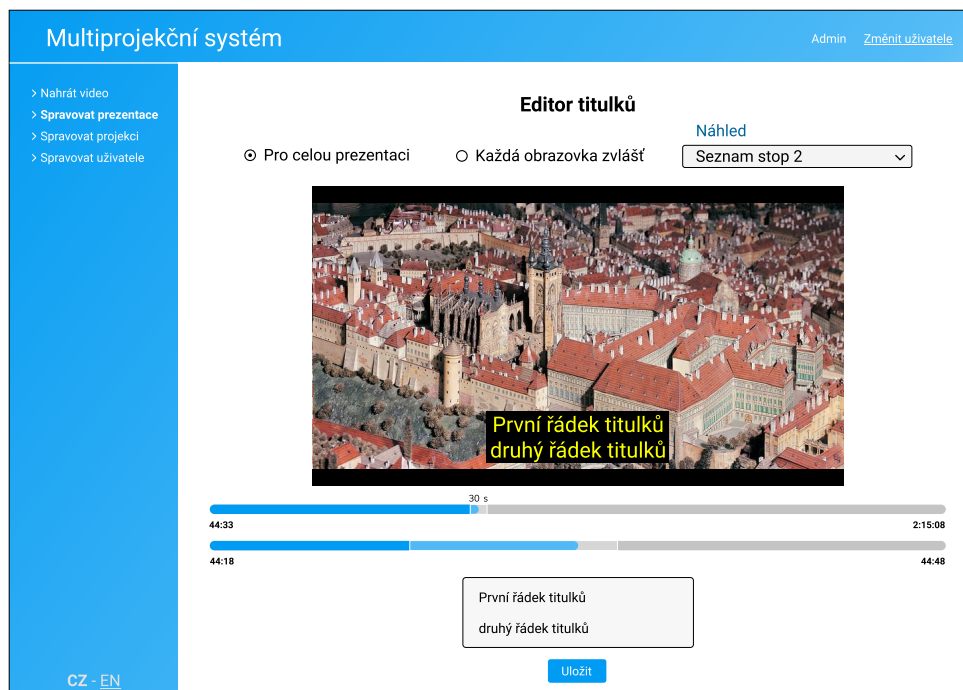
Obrázek 4.3: Správa prezentací

rozsah od 44 minut a 18 sekund do 44 minut a 48 sekund. Na spodní ose dále uživatel vybírá časový rozsah, ve kterém se má zobrazovat jeden nebo dva řádky titulků. Text titulků se zapisuje do textového pole pod časovými osami. Po dokončení úprav uživatel práci uloží kliknutím na tlačítko Uložit.

Obrazovka pro úpravu kotev (obr. 4.5) obsahuje, stejně jako editor titulků, náhled videa a časovou osu. Ve spodní části obrazovky se nachází tabulka vytvořených kotev. Poslední řádek tabulky je tlačítko, kterým lze vytvořit novou kotvu. Náhled videa je možné využít pro snadnější orientaci na časové ose. Po dokončení práce je potřeba změny potvrdit kliknutím na tlačítko Uložit.

Obrazovka pro správu prezentace (obr. 4.6) zobrazuje především všechny Seznamy stop, které jsou v této prezentaci obsaženy. Každý seznam stop lze upravit pomocí tlačítka s tužkou, nebo odstranit červeným tlačítkem s ikonou odpadkového koše. Před nevratnými úpravami, jako je odstranění videa, seznamu stop, nebo prezentace se zobrazí potvrzovací okno, aby nedošlo k nechtěným destruktivním úpravám. Nad výčtem Seznamů stop je zobrazen název Prezentace, který lze změnit pomocí tlačítka s piktogramem tužky. Nový seznam stop je možné vytvořit pomocí modrého tlačítka v pravém dolním rohu karty s výčtem Seznamů stop.

Pokud uživatel klikne na tlačítko pro úpravu Seznamu stop, zobrazí se výčet videí (obr. 4.7), který daný Seznam stop obsahuje. Každé video lze odstranit, nebo změnit jeho název. Do Seznamu stop lze přidat nové video pomocí tlačítka v dolní části karty. Aplikace uživateli umožní přidat buď nové video, nebo video z jiného seznamu stop (obr. 4.8). Pokud uživatel vybere možnost nahrát nové video, zobrazí se modální dialog (obr. 4.9) umožňující nahrání videa a úpravu jeho názvu. Pokud se uživatel rozhodne využít již existující nahrané video, zobrazí

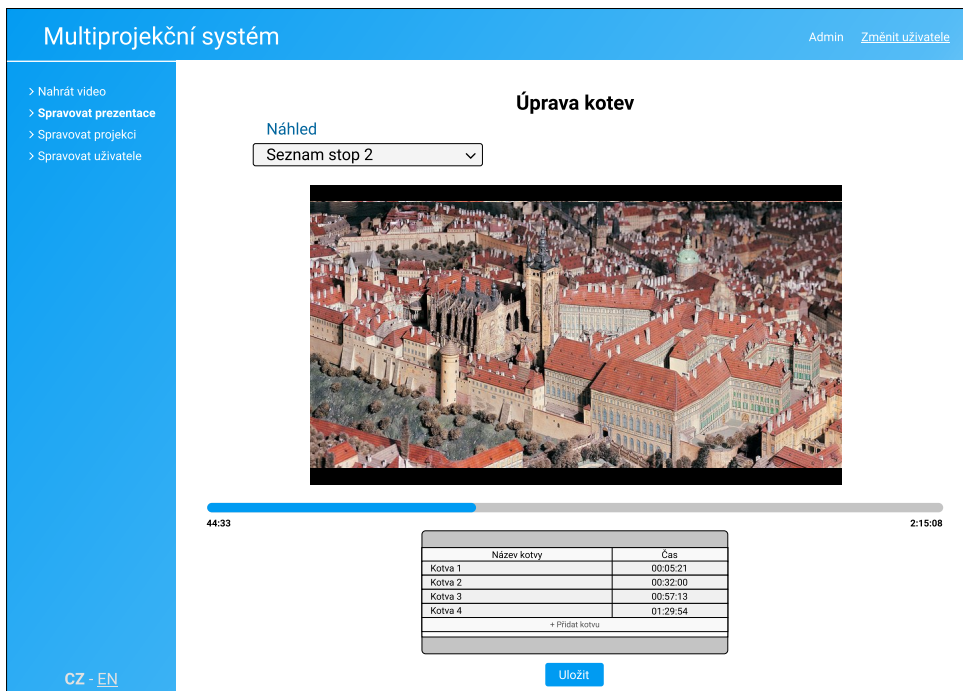


Obrázek 4.4: Editor titulků

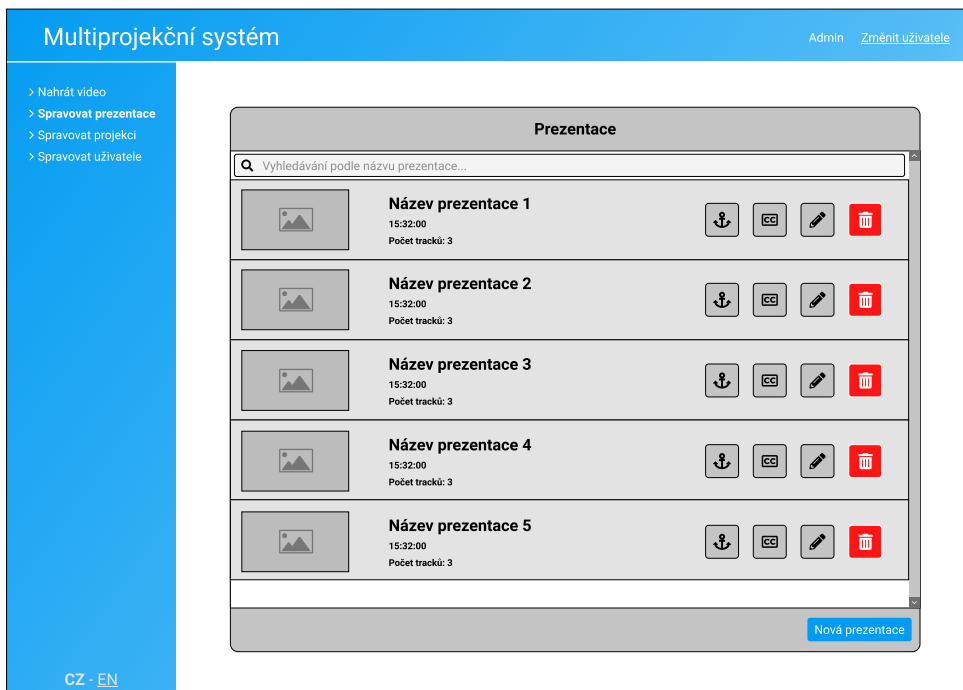
se dialog (obr. 4.10) umožňující výběr Prezentace a Seznamu stop, odkud se má video použít. Poté uživatel vybere video, které chce v seznamu stop použít a volbu potvrdí tlačítkem Uložit.

Aplikace slouží i jako záložní systém pro ovládání projekce. Proto jsem navrhl i obrazovku, která slouží k tomuto účelu (obr. 4.11). Obsahuje indikační časovou osu pro rychlou identifikaci časového průběhu prezentace, dále obsahuje sadu tlačítek typickou pro ovládání multimédií. Tlačítko play se zobrazuje, pokud je prezentace pozastavena. Naopak, pokud prezentace probíhá, zobrazuje se tlačítko pause. Pod tímto ovládacím panelem je seznam kotev vytvořených pro tuto prezentaci. Na každou kotvu se dá přeskočit kdykoliv v průběhu prezentace.

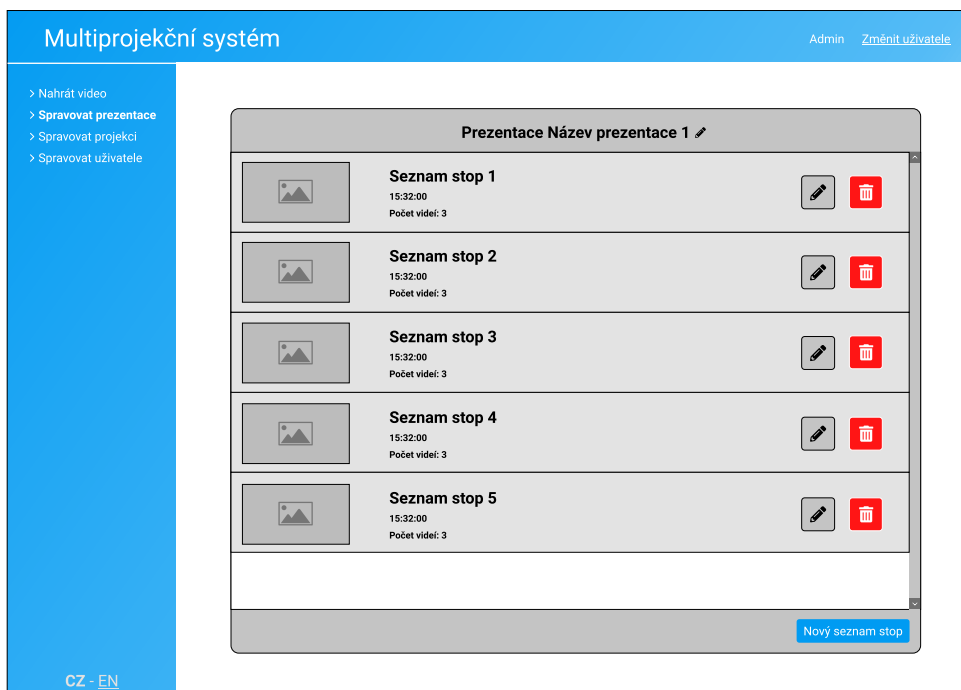
Pro správu uživatelů slouží obrazovka (obr. 4.12), která je přístupná jako poslední možnost v nabídce na levé straně okna. Obrazovka nabízí možnost přidat nového uživatele kliknutím na tlačítko pod tabulkou existujících uživatelů. Dále je možné u již existujících uživatelů měnit jejich role, aktivovat, nebo deaktivovat účet, resetovat heslo účtu, nebo účet smazat. Všechny změny je potřeba potvrdit tlačítkem uložit v dolní části obrazovky.



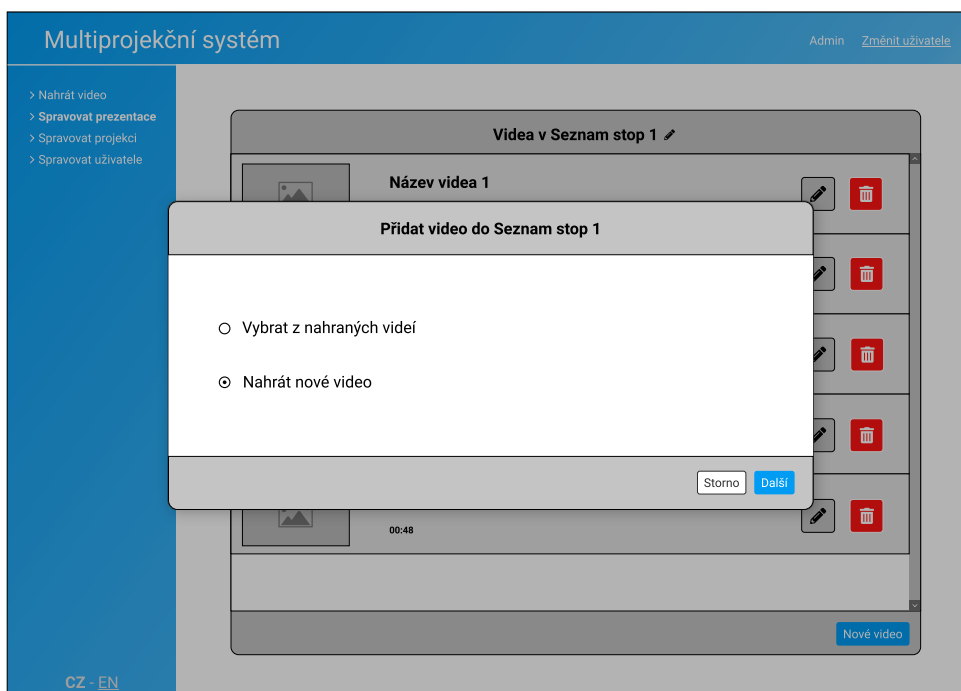
Obrázek 4.5: Úprava kotev



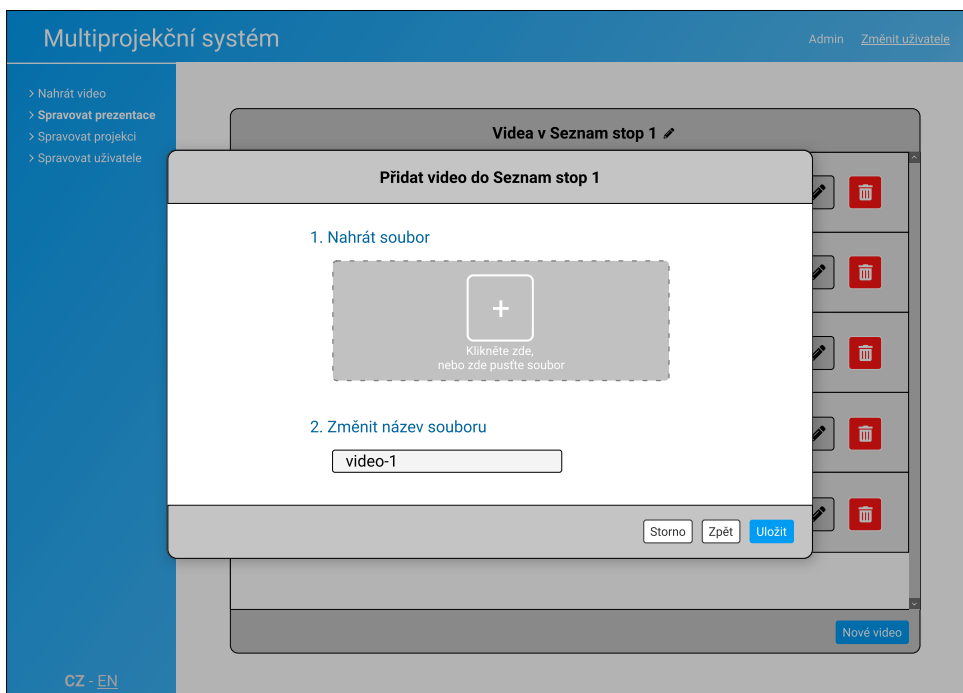
Obrázek 4.6: Správa prezentace



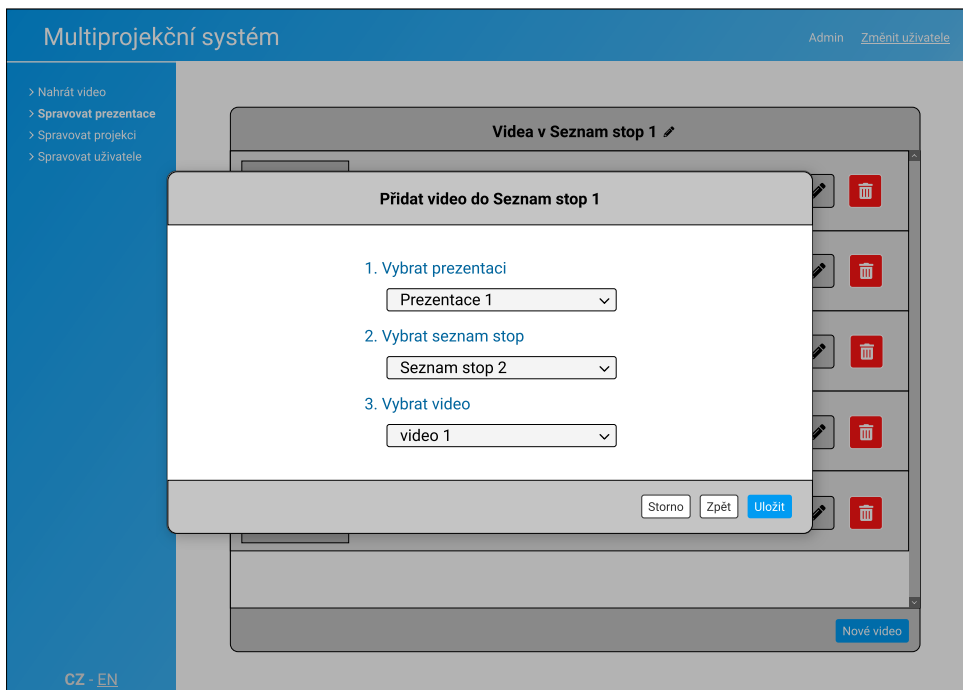
Obrázek 4.7: Úprava Seznamu stop



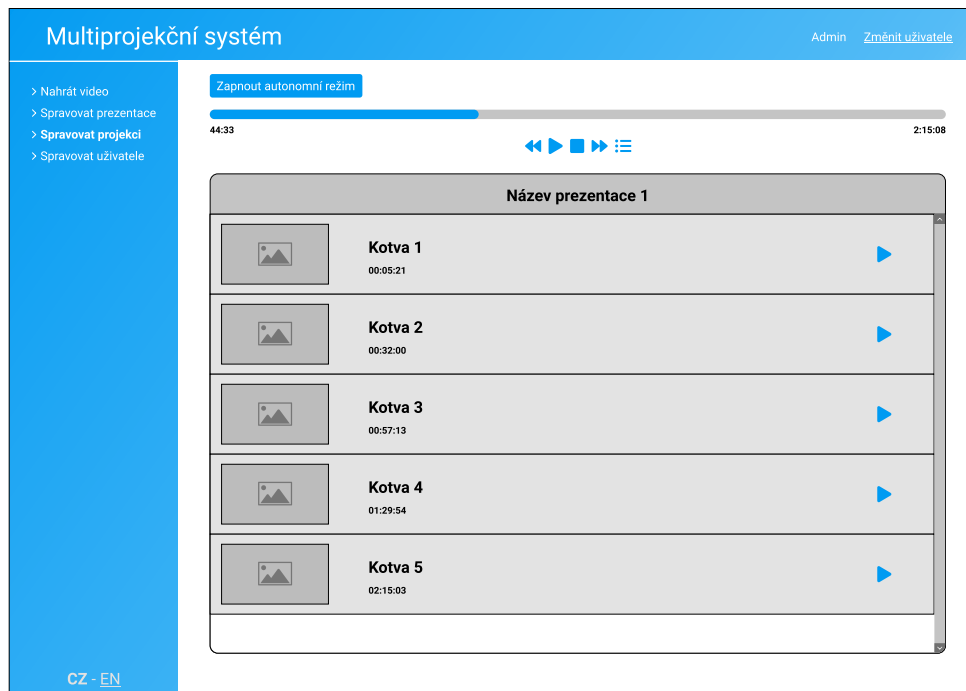
Obrázek 4.8: Výběr způsobu přidání nového videa



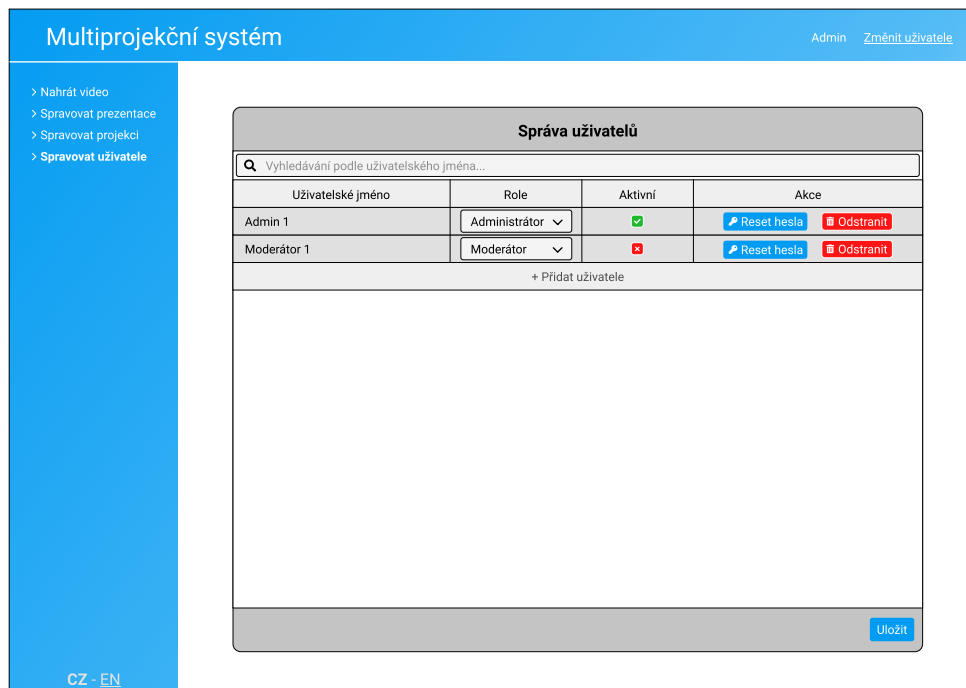
Obrázek 4.9: Přidání videa do Seznamu stop - nové video



Obrázek 4.10: Přidání videa do Seznamu stop - existující video



Obrázek 4.11: Správa projekce



Obrázek 4.12: Správa uživatelů

Kapitola 5

Příprava pro implementaci

5.1 Framework Vue.js

Podle oficiální webové stránky [4] je Vue (výslovnost /vjũ/) progresivní framework pro tvorbu uživatelských rozhraní. Typicky se používá pro tvorbu single-page aplikací za použití moderních nástrojů pro vytváření přepoužitelných komponent a podpůrných knihoven.

Vue nabízí snadné deklarativní renderování, díky kterému docílíme velmi snadné propagace dat z JS do DOM webové aplikace. Díky podmíněnému renderování můžeme snadno umožnit zobrazování dat v závislosti na stavu aplikace. Vykreslování pomocí smyček využijí pro renderování seznamů - např. pro výčet videí v Seznamu stop, apod. Aplikace bude přehledně rozdělena do více komponent, díky tomu bude snadno udržovatelná.

Komunita Vue vytvořila velké množství volně dostupných komponent a knihoven [5], díky kterým je velice snadné implementovat tabulky, notifikace, načítací spinnery, progress bary, tooltips, menu, carousely a spoustu dalších prvků používaných v moderních webových aplikacích.

Vue využívá Client Side Rendering, díky čemuž je používání aplikace velmi svižné. [6] Nevýhodou tohoto přístupu je především v tom, že je složitější takovéto aplikace procházet automaty, které indexují webové stránky např. pro Google, aby je uživatelé snáze našli v internetu. Vzhledem k tomu, že tato aplikace je neveřejná a použitelná pouze v lokální síti, tato nevýhoda zaniká a využijeme naplno předností Client Side Renderingu.

Velkou výhodou Vue je existence oficiálních knihoven pro routing a state management, díky čemuž je velmi usnadněn vývoj a debugging. Přístup je jednotný napříč aplikacemi a proto je jednoduché najít pomoc na fórech, pokud při vývoji narazíme na překážky. Dokumentace Vue je velice podrobná a jen výjimečně se střetneme s nedostatky.

Aplikace se skládají z komponent, které obsahují šablonu, kód JavaScriptu a CSS stylování v jednom souboru pro každou komponentu. Šablona je založená na syntaxi HTML a umožňuje bez námahy propojit uživatelské rozhraní s datovým modelem pod ním.

Tento framework jsem vybral především pro jeho jednoduchost a strmou křivku učení. Navíc jeho výkon a množství přenášených dat je srovnatelný nebo i lepší v porovnání s frameworky jako třeba Angular, nebo React. [7]

5.1.1 Struktura projektu

Na webové stránce [8] je k dispozici vzorový projekt webové aplikace s implementací front-endu pomocí Vue.js a back-endu založeném na Spring Bootu. Pro vytvoření struktury webové aplikace využijí balíček Vue Router, kde nadefinují cesty k jednotlivým stránkám. Napojení na Spring Boot back-end probíhá pomocí balíčku axios. Poté vytvořím jednotlivé služby, kterými budou pomocí axiosu posílat HTTP dotazy na server.

Kapitola 6

Implementace

Projekt jsem realizoval za pomoci frameworku Vue.js 3, který jsem doplnil především následujícími knihovnamí:

- axios pro odesílání HTTP požadavků na back-end server
- vuex pro state management
- vue-router pro navigaci mezi komponentami
- node-sass pro podporu CSS preprocessingu
- element-plus, balíček základních UI komponent

Celkem jsem vytvořil 15 různých komponent. Jejich hrubá struktura je k nahlédnutí na obrázku 6.3. Analýza vychází z diplomové práce pana Ing. Trojana, avšak současný stav back-endu je mírně odlišný. Proto jsou některé části aplikace rozdílné od původního návrhu. Velkou část aplikace se podařilo napojit na back-end. REST API bylo vytvářeno souběžně s vývojem této aplikace, protože původní řešení REST API nevyužívalo. Některé funkcionality nejsou tím pádem plně zprovozněny. Mezi funkční části patří především přihlášení, upload videa a vytváření a úprava prezentací a seznamů stop.

6.1 Použité knihovny

6.1.1 Axios

Axios je HTTP klient pro prohlížeče a node.js založený na Promise - objekt reprezentující dokončení (nebo selhání) asynchronní operace v JavaScriptu. V mé práci ho používám pro komunikaci s back-end serverem, který má přístup k databázi. Jeho použití je velice jednoduché. Po jeho importu do skriptu je třeba zavolat jednu z metod, podle toho, jaký HTTP dotaz chceme odeslat. Např. pro dotaz na existující Prezentace je připraveno následující API: `GET /api/displaysets`. Data získáme následujícím voláním 6.1.

Úryvek kódu 6.1: Použití knihovny Axios

```
import axios;

axios.get('/api/displaysets')
  .then(response => {
    const displaySets = response.data;
  })
  .catch(error => handleError);
```

V tomto příkladu by byl seznam prezentací uložen do konstanty `displaySets`. `handleError` by měla být funkce, která např. zaznamená informace o chybě a uživatele upozorní (např. pomocí `toast message`), že došlo k chybě.

6.1.2 Vuex

Vuex je knihovna pro aplikace Vue.js. Slouží jako centralizované úložiště pro všechny komponenty aplikace a umožňuje vytvářet pravidla, která zajistí, že data jsou upravena předvídatelným způsobem. [9] Toto úložiště je definované čtyřmi koncepty:

- State - samotná data
- Getters - metody, které komponentám poskytují „read-only“ přístup k datům
- Mutations - metody, které umožňují upravovat data
- Actions - metody, které zjednodušují práci s daty a používají Mutation pro jejich úpravu.

Komponenty mají přístup pouze ke metodám Getter a Action. Tuto knihovnu jsem použil pro ukládání stavu třech základních objektů: Uživatel, Prezentace a Seznamy stop.

Jako příklad použití v mé práci uvedu manipulaci s Prezentacemi. Prezentace ukládám v objektu 6.2 (podobně jako ukládání do datové struktury Dictionary nebo Map), kde klíčem k prezentaci je její ID. To mi umožňuje rychlý přístup ke konkrétní prezentaci bez procházení pole.

Úryvek kódu 6.2: State

```
const state = {
  displaySetsById: {}
}
```

Navrhl jsem dva Gettery `displaySets` poskytuje Prezentace v jednoduchém poli, např. pro zobrazení v seznamu Prezentací a `displaySetsById` vrací Prezentace tak, jak jsou uloženy ve State.

Úryvek kódu 6.3: Getters

```
const getters = {
  displaySets: state => Object.values(state.displaySetsById),
  displaySetsById: state => state.displaySetsById
}
```

Nejjednodušší Mutation, který jsem implementoval je `UPDATE_DISPLAY_SET` 6.4, který aktualizuje jednu prezentaci v úložišti.

Úryvek kódu 6.4: Mutation

```
const mutations = {
  UPDATE_DISPLAY_SET: (state, payload) => {
    state.displaySetsById[payload.id] = payload;
  },
  ...
}
```

Jako příklad Action jsem vybral funkci `loadDisplaySets` 6.5. Vrací Promise, díky čemuž může komponenta indikovat, že probíhá komunikace se serverem např. zobrazením spinneru. Tato funkce pošle dotaz na server pomocí knihovny `Axios` a data z odpovědi uloží pomocí mutation `UPDATE_DISPLAY_SET` do úložiště. K těmto datům mají poté přístup všechny komponenty.

Úryvek kódu 6.5: Action

```
const actions = {
  loadDisplaySets({ commit }) {
    return new Promise((resolve, reject) => {
      axios.get('/api/displaysets')
        .then(({ data }) => {
          commit('UPDATE_DISPLAY_SETS', data)
          resolve()
        }, err => reject(err))
    });
  },
  ...
}
```

6.1.3 Vue Router

Vue Router je oficiální router pro Vue.js. Díky této knihovně je možné se v aplikaci pohybovat za použitím běžných URL adres, přestože se jedná o Single-Page Aplikaci. V tomto projektu jsem připravil 9 různých cest, z nichž každá představuje jednu z hlavních komponent aplikace. Jedna z výhod požití Vue Routeru je omezení přístupu na jednotlivé adresy automatickým přesměrováním. Pokaždé, když se změní cesta v URL se zavolá funkce 6.6, která rozhodne, jestli je potřeba uživatele přesměrovat jinam, než kam se aktuálně snaží přistoupit. Logika funguje jednoduše: Pokud uživatel není přihlášený, přesměruj ho na přihlašovací stránku. Pokud je přihlášený a není Administrátor, přesměruj ho na Správu projekce. Pokud není specifikována žádná cesta, přesměruj uživatele na Správu prezentací.

Úryvek kódu 6.6: Omezení přístupu přesměrováním

```
router.beforeEach((to, from, next) => {
  const user = store.getters.user;
  if (!user && to.path !== '/login') {
    return next('/login');
  } else if (user && !user.isAdmin && to.path !== '/projection') {
    return next('/projection');
  } else if (to.path === '/') {
    return next('/displaySets');
  }
  return next();
});
```

6.1.4 Node sass

CSS preprocesor je program, který umožňuje vygenerování CSS z unikátní syntaxe daného preprocesoru. [10] Vue.js podporuje CSS preprocessing, pokud aplikace obsahuje knihovnu node-sass. Po její instalaci stačí definovat jazyk preprocesoru v tagu `script` komponenty, např. `<script lang='scss'>`. V tomto projektu využívám především zanořování pravidel, díky kterému je stylování přehlednější a čitelnější.

6.1.5 Element Plus

Element Plus je knihovna pro Vue.js, která poskytuje základní prvky uživatelského rozhraní s jednotným designem. Obsahuje formulářové prvky, jako třeba radio buttony, checkboxy, základní textová pole, přepínače, slidery, nebo prvek pro nahrání souboru s podporou drag&drop, který jsem využil. Dále jsem použil některé ikony a další vizuální prvky, které podporují snazší orientaci v systému. Další důležitá funkce, kterou takto knihovna nabízí, je zobrazení notifikačních hlášek, např. při úspěšném uložení dat, nebo naopak pokud se vyskytne chyba.

6.2 Klíčové vlastnosti

6.2.1 Správa prezentací

Základem každé prezentace je video. Proto je velmi důležité, aby bylo možné vložit do systému nová videa. To je umožněno komponentou `VideoUploader`, ve které má uživatel možnost vybrat video z úložiště v počítači, změnit jeho název a určit, do které složky v Multiprojekčním systému bude nahráno. Tato komponenta je použita na dvou místech: ve Správě videí, dostupné z navigačního panelu, a při vytváření nebo úpravě seznamu stop, jako jeden z tabů modálního dialogu.

Kromě možnosti nahrát nové video umožňuje sekce Správa videí také jejich mazání (není napojeno na back-end) a stažení ze serveru do uživatelského počítače. Dále je zde možné vytvářet nové složky v systému.

Když má uživatel videa připravená v systému, potřebuje připravit prezentace. To je umožněno v sekci Spravovat Prezentace. Zde uživatel vidí kartu se seznamem existujících prezentací. Pod nadpisem karty se nachází pole pro filtrování Prezentací podle názvu. Pokud se v tomto poli nachází více než dva znaky, bude seznam Prezentací obsahovat pouze takové Prezentace, které mají v názvu obsažen zadaný výraz. Každý řádek reprezentuje jednu prezentaci. Na jeho levé straně se nachází náhled, vedle něj název Prezentace a po pravé straně akční tlačítka. První tlačítko slouží k úpravě názvu Prezentace, druhé k vytváření kotev pro Prezentaci a poslední ke smazání Prezentace. V pravé dolní části karty se nachází tlačítko pro vytvoření nové Prezentace. Kliknutím na název prezentace může uživatel přejít na obrazovku se Seznamy stop prezentace. Ta je velice podobná předchozí obrazovce - znovu vidíme kartu s polem pro filtrování pod nadpisem. V každém řádku vidíme tentokrát pouze dvě tlačítka: jedno pro úpravu a druhé pro smazání. Při vytváření nebo úpravě Seznamu stop se otevře modální dialog, ve kterém uživatel určí jeho název a vybere videa, která do daného Seznamu stop patří. Bohužel backend nemá připraven, endpoint, který by vracel počet aktivních klientů. Aktuálně je tedy potřeba tento údaj zjistit předem v původním systému a v dialogu vybrat správný počet klientů. Každý klient musí mít přiřazené video (jedno video může být přiřazeno více klientům).

Na obrazovce Prezentací došlo ke změně oproti návrhu především v odebrání tlačítka pro tvorbu titulků. K tomu došlo, protože oproti původnímu předpokladu adresářová struktura nekopíruje strukturu Prezentací a Seznamů stop. Proto jsem se rozhodl tvorbu titulků z této obrazovky vyjmout a dedikoval jsem jí vlastní položku v hlavním menu.

6.2.2 Správa projekce

Pro správu projekce pan Ing. Trojan speciální API, které slouží jako mock úspěšného provedení dané akce (spuštění prezentace apod.). Akce provedené ve front-end aplikaci tedy nemají vliv na reálný stav klientských zařízení. Oproti původnímu návrhu dostala obrazovka správy prezentace navíc tlačítko pro nastavení pořadí Prezentací v autonomním režimu. To se nachází v horní části obrazovky vedle přepínače autonomního a manuálního režimu. Seznam kotev pod ovládacími prvky zobrazuje navíc kromě kotev i začátek jednotlivých Seznamů stop. Toto je jediná obrazovka, ke které má přístup uživatel, který má nastavena pouze práva Moderátora a ne Administrátora.

6.2.3 Správa uživatelů a logy

Správa uživatelů ani logy klientských zařízení nemají podporu API, tedy žádné akce provedené v aplikaci nemají vliv na systém.

Správa uživatelů obsahuje jednoduchou tabulku se pěti sloupci: Uživatelské jméno, Admin, Moderátor, Aktivní a Akce. Ve sloupci Akce se tlačítko pro úpravu - ta dovoluje změnu

hesla, změnu práv a aktivaci/deaktivaci. Pod tabulkou se nachází tlačítko pro vytvoření nového uživatele. To funguje velmi podobně jako jeho úprava, jen je zde navíc možnost upravit uživatelské jméno. V režimu úpravy/vytváření uživatele je k dispozici tlačítko pro vygenerování náhodného osmimístného hesla.

Logy klientských zařízení jsou k nahlédnutí pod poslední položkou v hlavním menu s názvem „Protokol klientů“. Vzhledem k tomu, že aplikace nemá k dispozici reálná data, jsou zobrazené informace generované. Je implementováno načítání (v tomto případě generování) nových dat při dosažení konce seznamu. Tato obrazovka neobsahuje žádnou uživatelskou interakci.

6.2.4 Nová funkcionalita

Podle zadání je součástí této práce rozšířená funkcionalita. Tyto obrazovky podlely největším změnám vůči původnímu návrhu. Tvorba titulků se přesunula z obrazovky Prezentací do hlavního menu, protože oproti původnímu návrhu, adresářová struktura není závislá na Prezentacích a Seznamech stop. Proto dává větší smysl vytvářet titulky přímo pro jednotlivé soubory. Titulky se poté budou zobrazovat v každé prezentaci, kde se vyskytuje dané video. Kotvy i titulky jsou omezeny nepřítomností API pro streamování videa, nebo stažení videosouborů (endpoint, který umožňuje stažení souboru, vrací soubor ve formátu zip, který je potřeba rozbalit). Pro zjednodušení jsem připravil náhled pomocí souboru uloženého v počítači uživatele. Pokud tedy chce uživatel využít náhledu videa při vytváření titulků nebo kotev, stačí vybrat správný soubor v počítači (pouze lokálně, data neputují na server).

Kotvy ani titulky nemají připravené endpointy pro spolupráci se serverem.

Kotvy

Kotvy 6.1, na rozdíl od titulků, lze vytvářet i „naslepo“ bez náhledu videa. Rozhodně bych však uživateli doporučil náhled videa využít. Po pravé straně obrazovky se nachází seznam vytvořených kotev seřazený podle umístění na časové ose. Pod tímto seznamem je textové pole a tlačítko pro vytvoření nové kotvy. Kotva se vytvoří s časem podle umístění jezdce na časové ose pod náhledem videa. Komponenta počítá s umístěním videa v Prezentaci, takže pokud je vybrané video pro náhled, na časové ose se zobrazí pouze úsek v Prezentaci, kam patří dané video. Pokud se video nenachází v prvním Seznamu stop Prezentace, bude nejnižší hodnota časové osy rovna součtu délek předchozích Seznamů stop. Zároveň pokud dané video není nejkratší v Seznamu stop, nebude na časové ose k dispozici celá jeho délka, protože délka Seznamu stop je určena nejkratším videem (ostatní se oříznou od konce). Vytvořené kotvy je možné uložit, ale pouze do místního úložiště prohlížeče. Vytvořené kotvy jsou následně vidět ve Správě projekce.

Multiprojekční systém admin — [Odhlásit se](#)

Správa videí

Spravovat prezentace

Editor titulků

Spravovat projekci

Spravovat uživatele

Protokol klientů


Kotvy pro Zkusebni prezentace 1

Načíst video Video

bench 2.mp4 bench.mp4

00:11,83: Zkusebni kotva 🗑️

+



Zkusebni kotva

Uložit

Obrázek 6.1: Kotvy

Vytváření titulků

Obrazovka editace titulků 6.2 je podobná obrazovce z předchozího odstavce. Na časové ose je však výběr rozsahu namísto bodu. Rozsah určuje časový úsek, ve kterém se zobrazuje daný text titulků. Po pravé straně opět seznam vytvořených titulků s časovým rozsahem. Tentokrát je zde však možnost na konkrétní titulek kliknout, čímž se komponenta přepne do stavu úpravy tohoto titulku. Aktuálně upravovaný titulek je vždy indikován modrým ohraničením v seznamu titulků. Při pohybu jezdcí časové osy se zároveň pohybuje i aktuální pozice v náhledu videa. Pokud je náhled pozastaven, pohyb pravým jezdcem zobrazí náhled v místě tohoto jezdcce. Pokud je video přehráváno, pohyb oběma jezdcem posune náhled na pozici prvního jezdcce. Tím pádem je při úpravě intervalu vždy vidět v náhledu celý rozsah, kdy je titulek zobrazen. Pod časovou osou jsou po stranách prvky, které ovládají posun jednotlivých jezdců o 0,1s a 0,01s, nebo jezdec přesunou na aktuální pozici v náhledu videa. Při stisknutí tlačítka uložit se vygeneruje soubor s příponou `.srt` a uživateli je umožněno tento soubor stáhnout.

Multiprojekční systém admin – [Odhást se](#)

Správa videí

Spravovat prezentace

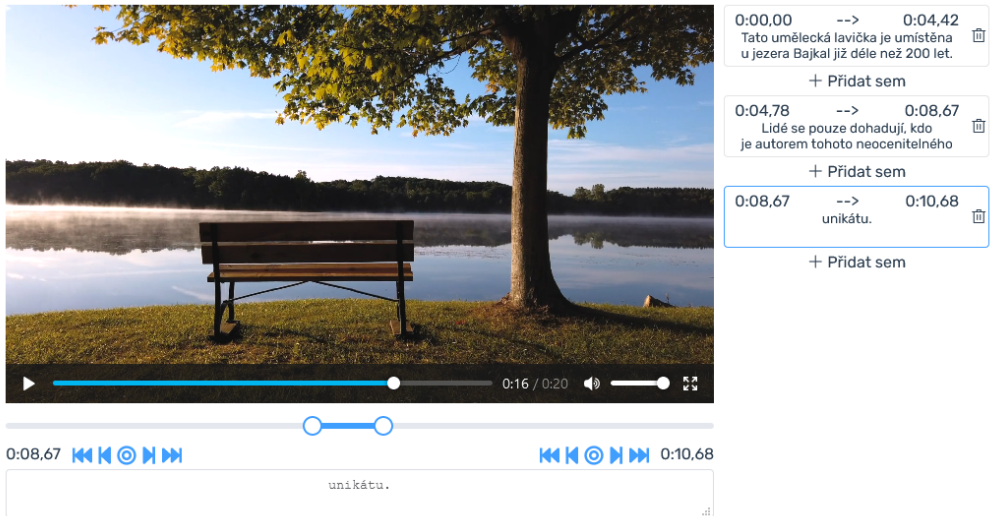
Editor titulků

Spravovat projekci

Spravovat uživatele

Protokol klientů

Tvorba titulků



0:00,00 --> 0:04,42
Tato umělecká lavička je umístěna u jezera Bajkal již déle než 200 let.

+ Přidat sem

0:04,78 --> 0:08,67
Lidé se pouze dohadují, kdo je autorem tohoto neocenitelného

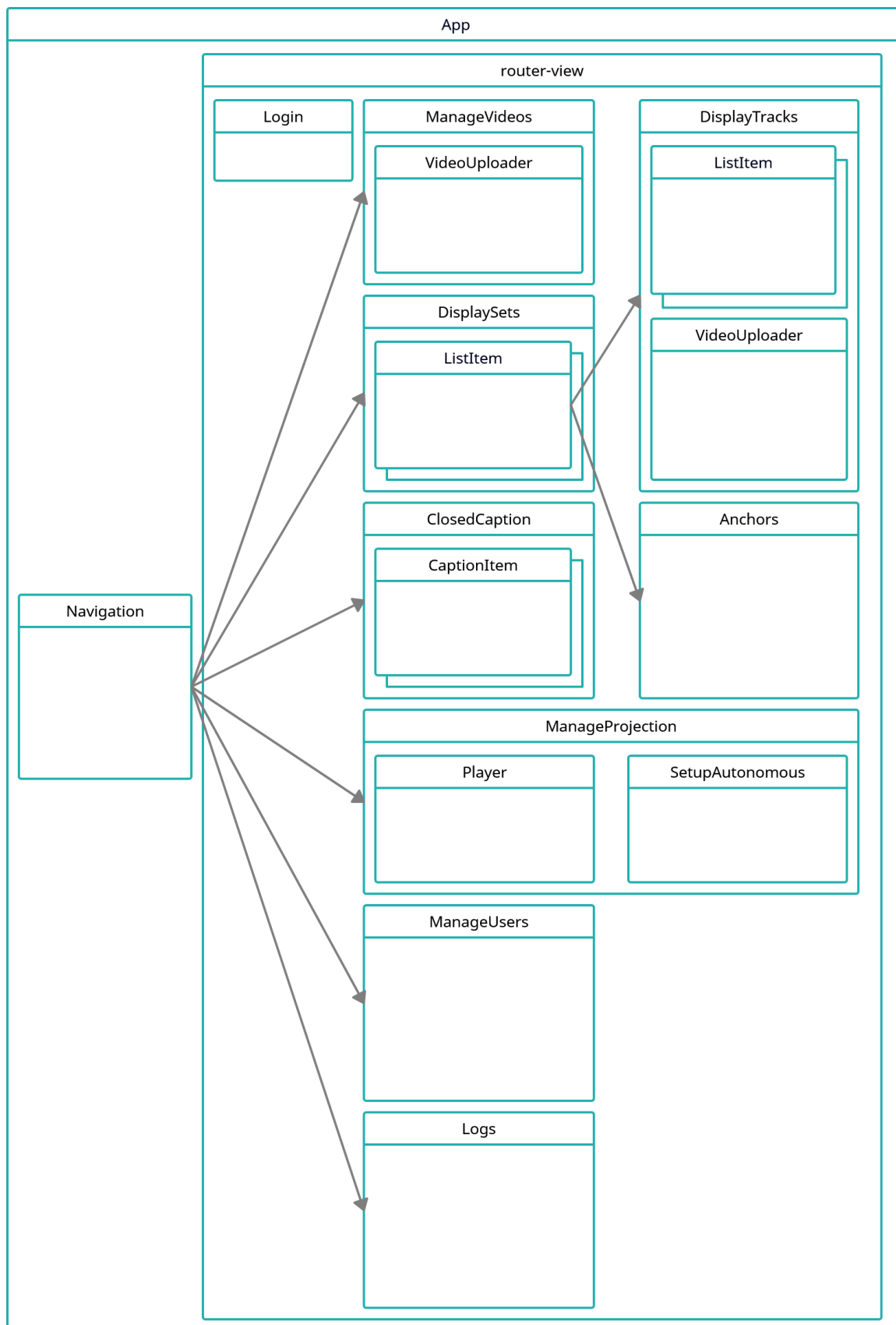
+ Přidat sem

0:08,67 --> 0:10,68
unikátu.

+ Přidat sem

Uložit

Obrázek 6.2: Struktura komponent



Obrázek 6.3: Struktura komponent

Kapitola 7

Evaluace

Součástí zadání je ověření funkcionality na dvou prezentacích. Začal jsem nahráním nového videa pomocí položky Správa videí v hlavním menu. Pro tento účel jsem zároveň vytvořil novou složku nazvanou BP_Riha. Nahrání nového videa proběhlo bez problému. V prezentacích použiji toto nové video společně s dvěma dalšími, které v systému již jsou nahrané. Dalším krokem je tedy Vytvoření nové prezentaci. Přešel jsem v hlavním menu na položku Spravovat prezentace, kde jsem klikl na tlačítko Nová prezentace. Novou prezentaci jsem nazval Zkusebni prezentace 1. Dále je potřeba vytvořit seznam stop. Kliknutím na název nové prezentace jsem přešel k seznamům stop této prezentace. Systém indikuje, že je prázdná. Vytvořil jsem tedy nový seznam stop kliknutím na tlačítko v pravém dolním rohu karty. V modálovém dialogu jsem vyplnil název prezentace Zkusebni seznam stop, pro prvního klienta jsem vybral nově nahrané video a pro další dva klienty jsem vybral náhodně dvě různá jiná videa. Stejný proces jsem opakoval pro druhou prezentaci, nazvanou Zkusebni prezentace 2. V žádném průchodu jsem nenarazil na problém.

Následně jsem pro každou prezentaci vytvořil jednu kotvu a nechal jsem si každou z nich zobrazit ve správě projekce. Každé prezentaci se správně zobrazila její délka pod časovou osou a v sekci kotev se zobrazil správně začátek seznamu stop i vytvořené kotvy.

Pro ověření funkčnosti mazání Prezentací jsem se poté pokusil smazat první Prezentaci. Zde jsem narazil na neodhalenou chybu. Prezentace, které obsahují Seznamy stop neumí back-end smazat. V rámci uživatelské přívětivosti jsem rozšířil varovnou hlášku o upozornění, že smazáním Prezentace se odstraní i její Seznamy stop a chybu jsem opravil jejich odstraněním před smazáním Prezentace.

Evaluace na základě dvou prezentací proběhla úspěšně.

Kapitola 8

Závěr

V úvodu jsem se zaměřil na analýzu existujícího řešení a pokusil jsem se ve vlastním návrhu o zlepšení uživatelské přívětivosti aplikace. Podařilo se mi splnit funkční požadavky a především základní funkcionalita v podobně celého procesu vytváření prezentací je plně funkční a napojené na existující back-end. Části, které se ještě nepodařilo napojit jsou většinou ty, které jsou v rámci této aplikace považovány pouze jako záložní, nebo jsou rozšířením původního systému. Za aktuálně největší nedostatek osobně považuji to, že náhledy videí při vytváření kotev a titulků jsou potřeba nahrávat z počítače uživatele.

8.1 Budoucnost

Prostoru pro zlepšení je bezesporu mnoho. Přinejmenším dokončení napojení na server systému, ale rád bych zlepšil i způsob práce s daty načtenými ze serveru - aktuálně totiž dotazuji server o Seznam stop pokaždé, kdy se otevře detail Prezentace. Za předpokladu, že v jednu dobu v systému pracuje pouze jeden uživatel je to zbytečné. Lepší řešení bych viděl v ukládání Seznamů stop ve struktuře podle ID Prezentace, aby se nenačítala stejná data znovu. Podobnou úpravu bych aplikoval i pro Prezentace, složky a metadata videí. To by celkově zlepšilo výkon aplikace a zkrátilo doby načítání. Poté bych uživateli umožnil manuálně aktualizovat data, pokud by uvážil, že je to potřeba.

Práce na tomto projektu mi byla velkým přínosem. Jsem rád, že jsem dostal možnost na něm pracovat.

Příloha A

Zprovoznění aplikace

1. Pro sestavení projektu je potřeba mít nainstalované prostředí Node.js¹ a mít přístup k internetu.
2. Rozbalte archiv zdroj.zip kdekoli v počítači.
3. Spusťte příkazový řádek a přesuňte se do kořenového adresáře rozbaleného projektu (uvnitř kořenového adresáře se nachází složky `src` a `public`)
4. Spusťte příkaz `npm i` a počkejte, až se dokončí instalace závislostí.
5. Spusťte příkaz `npm run serve`.
6. V příkazovém řádku by měla být zobrazena adresa lokálního serveru, kde je přístupná aplikace (pravděpodobně `http://localhost:8080/`). Adresu otevřete v internetovém prohlížeči.
7. Přihlašovací údaje pro uživatele jsou následovné:
uživatelské jméno: `bp_riha`
heslo: `heslo1`

¹ke stažení na adrese <https://nodejs.org/en/>

Bibliografie

- [1] O. Trojan, “Multiprojection control system”, dipl, České vysoké učení technické v Praze, květ. 2020.
- [2] T. Lowdermilk, *User-centered design: A developer’s guide to building user-friendly applications (Anglicky) [Uživatelsky zaměřený návrh: Vývojářova příručka pro tvorbu uživatelsky přívětivých aplikací]*. O’Reilly Media, 2013.
- [3] Autor článku neznámý, webová stránka navržena společností moderní historie, s.r.o. (2021). “Langweilův model Prahy - Muzeum Prahy”, WWW: <http://www.muzeumprahy.cz/langweiluv-model-prahy/> (cit. 08.01.2021).
- [4] Neznámý autor. (2020). “Introduction - Vue.js (Anglicky) [Úvod - Vue.js]”, WWW: <https://vuejs.org/v2/guide/> (cit. 09.01.2021).
- [5] Více než 1900 přispěvatelů. (2021). “Components & Libraries (Anglicky) [Komponenty a knihovny]”, WWW: <https://github.com/vuejs/awesome-vue#components-libraries> (cit. 09.01.2021).
- [6] P. Hong, *Practical web design: learn the fundamentals of web design with HTML5, CSS3, Bootstrap, jQuery, and Vue.js (Anglicky) [Praktický web design: naučte se základy web designu s HTML5, CSS3, Bootstrap, jQuery a Vue.js]*. O’Reilly Media, 2013.
- [7] Jacek Schae. (břez. 2020). “A RealWorld Comparison of Front-End Frameworks 2020 (Anglicky) [Reálné srovnání front-endových frameworků 2020]”, WWW: <https://medium.com/dailyjs/a-realworld-comparison-of-front-end-frameworks-2020-4e50655fe4c1> (cit. 19.05.2021).
- [8] bezkoder (pseudonym). (pros. 2020). “Spring Boot + Vue.js: CRUD example (Anglicky) [Spring Boot + Vue.js: CRUD vzorový projekt]”, WWW: <https://bezkoder.com/spring-boot-vue-js-crud-example/> (cit. 09.01.2021).
- [9] Neznámý autor. (). “What is Vuex? (Anglicky) [Co je to Vuex?]”, WWW: <https://vuex.vuejs.org/#what-is-vuex> (cit. 19.05.2021).
- [10] Mozilla a individuální přispěvatelé. (dub. 2021). “CSS preprocessor (Anglicky) [CSS preprocesor]”, WWW: https://developer.mozilla.org/en-US/docs/Glossary/CSS_preprocessor (cit. 19.05.2021).