

Univerzita Hradec Králové

Fakulta informatiky a managementu

Katedra informačních technologií

Webové rozhraní pro aplikaci LocalMovieList

(front-end)

Diplomová práce

Autor: Bc. Michal Petras

Studijní obor: Aplikovaná informatika (ai2-p)

Vedoucí práce: prof. RNDr. Peter Mikulecký, PhD.

Odborný konzultant:

Pracoviště: Katedra informačních technologií

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracoval samostatně s použitím uvedené literatury.

V Hradci Králové dne 1.4.2024

Bc. Michal Petras

Poděkování:

Děkuji vedoucímu diplomové práce prof. RNDr. Peteru Mikuleckému, PhD. za metodické vedení práce a veškerou pomoc a podporu při tvorbě projektu.

Abstrakt

Cílem této práce je navrhnout a implementovat front-end webové aplikace pro správu a přehled filmů na vlastním zařízení nebo serveru. Aplikace má za úkol poskytnout uživatelům přehledné a snadno ovladatelné uživatelské rozhraní. V úvodu práce probíhá analýza dostupných technologií, které jsou následně použity při návrhu funkčního uživatelského rozhraní.

V implementační části práce jsou podrobně popsány jednotlivé stránky aplikace, včetně komponent a jejich funkcionalit. Zvláštní důraz je kladen na efektivní využití technologií a návrh uživatelsky příjemného prostředí. Popsané stránky zahrnují přehled filmů, nastavení uživatelského profilu a detailní informace o jednotlivých filmech. Zohledněním uživatelských potřeb a moderních designových trendů bylo dosaženo optimálního uživatelského zážitku.

Klíčová slova: Filmy, databáze filmů, front-end aplikace, NextJs

Abstract

The aim of this thesis is to design and implement the front-end of a web application for managing and reviewing movies on a personal device or server. The application is intended to provide users with a clear and easily navigable user interface. The introduction of the thesis includes an analysis of available technologies, which are subsequently utilized in the design of a functional user interface.

In the implementation part of the thesis, individual pages of the application are detailed, including components and their functionalities. Special emphasis is placed on the efficient utilization of technologies and the design of a user-friendly environment. The described pages encompass an overview of movies, user profile settings, and detailed information about individual films. By considering user needs and incorporating modern design trends, an optimal user experience has been achieved.

Keywords: Movies, movie database, front-end application, NextJs

Obsah

1	ÚVOD	1
2	ANALÝZA ŘEŠENÍ	3
2.1	VÝVOJ HTML	4
2.2	POSTUPY A JAZYKY	4
2.2.1	<i>Multi Page Application (MPA)</i>	4
2.2.2	<i>Single Page Application (SPA)</i>	4
2.2.3	<i>Jazyky</i>	5
2.2.4	<i>Frameworky</i>	5
2.3	UŽIVATELSKÉ PROSTŘEDÍ	7
2.3.1	<i>Komunikace s BE</i>	8
2.3.2	<i>Zabezpečení dat uživatele</i>	9
2.3.3	<i>Přehled a zobrazení filmů</i>	10
2.3.4	<i>Správa filmů</i>	10
2.3.5	<i>Volba jazyka</i>	10
2.3.6	<i>Design a responsibilita</i>	11
2.3.7	<i>Export a import filmů</i>	14
3	NÁVRHOVÁ ČÁST	15
3.1	OBECNÉ POŽADAVKY	15
3.2	FUNKČNÍ POŽADAVKY	15
3.3	ZVOLENÉ TECHNOLOGIE	16
3.4	UŽIVATELSKÉ ROZHŘANÍ	16
3.4.1	<i>Konceptuální design</i>	16
3.4.2	<i>Základní interakční prvky</i>	17
3.4.3	<i>Wireframy</i>	18
4	IMPLEMENTACE	26
4.1	ZALOŽENÍ PROJEKTU	26
4.1.1	<i>Verzovací nástroj</i>	26
4.1.2	<i>Vývojové prostředí</i>	26
4.1.3	<i>Inicializace projektu NextJS</i>	26
4.2	IMPLEMENTACE STRÁNEK A KOMPONENT	27
4.2.1	<i>Komponenta výběru jazyka a překlad stránek</i>	27
4.2.2	<i>Přihlašovací stránka</i>	28
4.2.3	<i>Komponenta NavBar</i>	31
4.2.4	<i>Komponenta MovieCard</i>	33

4.2.5	<i>Komponenta MyCarousel</i>	37
4.2.6	<i>Komponenta ProgressBar</i>	40
4.2.7	<i>Komponenta Toast</i>	40
4.2.8	<i>Komponenta EditMovieModal</i>	41
4.2.9	<i>Komponenta ChangeMovieModal</i>	41
4.2.10	<i>Komponenta ImportedMovieCard</i>	43
4.2.11	<i>Komponenta Layout</i>	44
4.2.12	<i>Komponenta Pagination</i>	45
4.2.13	<i>Komponenta MovieDetail</i>	46
4.2.14	<i>Stránka Detail filmu</i>	48
4.2.15	<i>Stránka Filmy</i>	49
4.2.16	<i>Stránka Sdílené filmy</i>	51
4.2.17	<i>Stránka Prohlížeč</i>	51
4.2.18	<i>Stránka Nahrát</i>	52
4.2.19	<i>Stránka Stáhnout</i>	56
4.2.20	<i>Stránka Profil</i>	57
5	SHRNUTÍ VÝSLEDKŮ	63
6	ZÁVĚRY A DOPORUČENÍ	64
7	SEZNAM POUŽITÉ LITERATURY	65
8	PŘÍLOHY	67

Seznam obrázků

Obr. 1 Wireframe – Přihlášení	19
Obr. 2 Wireframe – Registrace.....	20
Obr. 3 Wireframe – Přehled	20
Obr. 4 Wireframe – Filmy.....	21
Obr. 5 Wireframe – Sdílené filmy	21
Obr. 6 Wireframe – Prohlížeč.....	22
Obr. 7 Wireframe – Nahrát.....	23
Obr. 8 Wireframe – Stáhnout.....	23
Obr. 9 Wireframe – Profil.....	24
Obr. 10 Wireframe – Detail filmu	25
Obr. 11 Inicializace projektu NextJs	26
Obr. 12 Přepínání jazyků.....	28
Obr. 13 Přihlašovací formulář.....	29
Obr. 14 Registrační formulář.....	29
Obr. 15 Navigační menu pro větší zařízení	32
Obr. 16 Navigační menu pro mobilní zařízení	33
Obr. 17 Možnosti zobrazení MovieCard	36
Obr. 18 Přehled – carousel na stránce.....	39
Obr. 19 Modální okno pro úpravu filmu.....	41
Obr. 20 Modální okno pro změnu filmu.....	42
Obr. 21 Ukázka komponenty ImportedMovieCard.....	44
Obr. 22 Detail filmu (anglicky)	47
Obr. 23 Detail filmu (česky)	48
Obr. 24 Detail filmu – Herci a doporučení	48
Obr. 25 Stránkování.....	49
Obr. 26 Stránka s filmy – základní filtry	50
Obr. 27 Stránka s filmy – rozšířené filtry	50
Obr. 28 Stránka Prohlížeč	52
Obr. 29 Šablona Google tabulek	56
Obr. 30 Export filmů.....	57
Obr. 31 Profil – Změna profilového obrázku	58
Obr. 32 Profil – základní informace.....	58

Obr. 33 Profil – Změna hesla.....	59
Obr. 34 Profil – notifikace změny hesla	59
Obr. 35 Profil – nastavení veřejného profilu (upozornění).....	60
Obr. 36 Profil – mazání všech filmů (upozornění)	61
Obr. 37 Profil – ukázka balíčku country-codes-list.....	61
Obr. 38 Profil – sekce Nastavení filmů	62

1 Úvod

Původní diplomovou prací mělo být navázání na bakalářskou práci autora této práce, kdy se mělo jednat o znovu vytvoření frontend webové aplikace pro jednodušší správu fondů v systému Blackboard Learn, konkrétně v interním systému oliva. [1]

Bohužel při implementaci bylo zjištěno, že aktuální verze Blackboard Learn nasazená v Oliva nesplňuje požadavky pro práci s API. Více informací o problémech popsal kolega Ondřej Kašpar ve své práci diplomové Tvorba poskytovatele dat (API) volnočasové aplikace LocalMovieList [2]. Z tohoto důvodu bylo od projektu odstoupeno a byl vybrán náhradní projekt s názvem LocalMovieList, který bude sloužit jako volnočasová aplikace.

V dnešní digitální éře existuje bohatá paleta služeb, které nabízejí široký výběr filmů dostupných prostřednictvím měsíčních nebo ročních předplatných. Mezi ně patří renomované platformy, jako jsou Netflix, HBO Max a Amazon Prime. Existují jedinci, kteří preferují vlastnictví a stahování filmů přímo na svůj počítač, vytvářejíce tak bohatou sbírku filmů různých žánrů.

Cílem této diplomové práce je zlepšit a zjednodušit práci těchto uživatelů, optimalizovat jejich čas a poskytnout jim přehled nad jejich filmovou kolekcí pomocí moderní webové aplikace. Uživatel bude schopen snadno nalézt konkrétní film a zjistit, kde je uložený.

Aplikace je strukturována do dvou klíčových částí: frontend a backend. Frontend (dále uváděno také jako FE) představuje uživatelské rozhraní, ovládání aplikace a zobrazování dat. Backend (dále uváděno také jako BE) je serverová část, která se stará o zpracování a ukládání dat.

Výslednou funkční aplikací (spojení BE a FE) bylo docíleno spoluprací dvou autorských týmů, kde autor této práce, Michal Petras, se věnoval frontend části, zatímco kolega Ondřej Kašpar pracoval na backend části. [2]

Diplomová práce je rozčleněna do tří klíčových částí. Analytická část se zaměřuje na průzkum vhodných technologií a definici základních funkcionalit.

Návrhová část stanovuje specifické požadavky na aplikaci, definuje komunikační procesy s backendem a navrhuje vzhled aplikace. Implementační část poskytuje podrobný popis samotné implementace aplikace.

2 Analýza řešení

Cílem této práce je vytvořit frontend webové aplikace pro správu a přehled filmů, které uživatel vlastní. Aplikace má poskytnout uživateli přehledné informace o filmech v jeho kolekci včetně dostupných jazyků. Důležitou funkcionalitou je umožnit uživateli procházet stránky s různými typy filmů, jako jsou naposledy přidané, nejlepší filmy z jeho kolekce nebo nejlepší sdílené filmy od ostatních uživatelů.

Další klíčovou stránkou aplikace bude zobrazení filmové kolekce. Zde bude uživatel mít možnost prohlížet své filmy, provádět úpravy, mazat, přidávat nebo měnit filmy za jiné. Uživatelské rozhraní bude přehledné, intuitivní a podporovat vyhledávání filmů.

Důležitou součástí webové aplikace bude možnost nahrát či stáhnout seznam filmů do externího formátu, jako je CSV soubor. Tato funkcionalita umožní uživatelům pohodlněji spravovat svou filmovou kolekci a usnadní jim vyhledávání a procházení filmů.

Pro úspěšné řešení této práce je nezbytné pečlivě zvolit vhodný jazyk a framework pro vývoj frontendu, aby bylo dosaženo efektivního a spolehlivého fungování webové aplikace pro správu filmů. Zohledňuje se rozšířenost a vhodnost zvoleného jazyka v konkrétním vývojovém prostředí.

Dále je klíčová analýza komunikace s backendem (BE), která zahrnuje detailní rozbor komunikačních protokolů a návrh API rozhraní. Specifikace formátu dat, zabezpečení a efektivní komunikace jsou nezbytné pro bezproblémovou spolupráci mezi frontendem a backendem, což může mít vliv na celkový výkon aplikace.

Součástí strategie je i důkladná analýza dostupných balíčků a nástrojů, které mohou vylepšit vývojový proces a přispět k modernímu vzhledu aplikace. Správný výběr balíčků může urychlit implementaci klíčových funkcí a zlepšit celkový uživatelský zážitek.

Integrace těchto kritických bodů do jednotného přístupu bude klíčem k úspěšnému návrhu a implementaci webové aplikace pro správu filmů, která bude nejen funkční, ale též uživatelsky přívětivá a odpovídající moderním standardům.

2.1 Vývoj HTML

Jazyk HTML neboli Hypertext Markup Language - hypertextový značkovací jazyk slouží pro tvorbu www stránek. Strukturu tvoří tzv. HTML tagy. HTML soubory mají koncovku .html.

V roce 1989 spolupracovali Tims Berners-Lee a Robert Cailliau na informačním systému pro CERN. Vzhledem ke složitosti doposud používaného řešení byl navržen jazyk HTML a protokol HTTP. Zároveň byl napsán první prohlížeč s názvem WorldWideWeb. První verze HTML pochází z roku 1991. První prohlížeč přichází o rok později a nepodporují grafický režim. V roce 1993 přichází verze 2.0 a přichází první grafický prohlížeč. V roce 1994 přichází Netscape. V roce 1996 se oficiálně objevuje verze 3.2 (doplněná verze 2.0 o tabulky) a na scénu přichází prohlížeč od Microsoftu MSIE. Roku 1997 vychází verze HTML 4.0. Jedná se o verzi 3.2 + rozšíření tabulek, obsahuje rámy, formuláře a CSS. V roce 1999 přichází verze 4.1. [3]

2.2 Postupy a jazyky

2.2.1 Multi Page Application (MPA)

Multi Page Application (MPA) představuje tradiční přístup k vývoji webových aplikací, kde každá stránka vyžaduje plné načtení ze serveru. Tento model poskytuje jednoduchou navigaci mezi různými stránkami, což je užitečné pro projekty s víceoddílovou strukturou. Vývojáři mohou snadno spravovat obsah jednotlivých stránek, což usnadňuje údržbu. Nicméně, MPA může vykazovat delší načítací časy, zejména při inicializaci aplikace, a vyžaduje více komunikace se serverem. [4]

2.2.2 Single Page Application (SPA)

Naopak Single Page Application (SPA) představuje modernější koncept, kde veškerý obsah je načten jednou při prvním přístupu a následné interakce s aplikací

probíhají bez nutnosti načítání celých stránek. To vede k rychlejší odezvě a lepšímu uživatelskému zážitku, zejména při plynulém přechodu mezi jednotlivými částmi aplikace. SPA může využívat pokročilé techniky načítání dat, aby minimalizovala komunikaci se serverem. [4]

2.2.3 Jazyky

Pro implementaci webové aplikace jsou k dispozici různé jazyky. JavaScript, jakožto klíčový jazyk pro interaktivní uživatelská rozhraní, je široce využíván. V kombinaci s TypeScriptem můžeme dosáhnout vyšší úrovně bezpečnosti a strukturovanosti kódu. HTML a CSS pak slouží pro definici struktury a stylu.

2.2.4 Frameworky

Při vývoji moderních webových aplikací jsou frameworky klíčovým prvkem. Vue.js, React, Angular, Next.js a Nuxt.js jsou populární frameworky, které usnadňují vývoj, správu stavu a optimalizaci výkonu. Každý z těchto nástrojů má své výhody a jsou vhodné pro konkrétní scénáře.

2.2.4.1 VueJS

Mezi **výhody** patří:

- **Jednoduchost:** Vue.js je známý svou jednoduchostí a snadnou „učitelností“, což usnadňuje rychlý vývoj aplikací.
- **Flexibilita:** Umožňuje postupnou integraci do existujících projektů bez velkých změn.
- **Dobrá dokumentace:** Framework má velmi dobře strukturovanou a srozumitelnou dokumentaci, což usnadňuje práci vývojářům.

Mezi **nevýhody** patří:

- **Menší komunita:** Ve srovnání s Reactem má Vue.js menší komunitu, což může ovlivnit dostupnost rozsáhlejších knihoven a nástrojů.
- **Menší korporátní podpora:** V porovnání s AngularJS nemá Vue.js tolik podpory ze strany velkých firem.

2.2.4.2 React

Mezi **výhody** patří:

- **Vysoká výkonnost:** React je známý svou vysokou výkonností díky virtuálnímu DOM a možnosti použití server-side renderingu.

- **Rozsáhlá komunita:** Má obrovskou komunitu vývojářů, což vede k bohatství dostupných knihoven, nástrojů a dokumentace.
- **Flexibilita:** React je velmi flexibilní a umožňuje vývojářům pracovat s komponentami, jak chtějí.

Mezi **nevýhody** patří:

- **Složitost:** Pro začátečníky může být učení Reactu složitější než u jiných frameworků.
- **Závislost na třetích stranách:** Ačkoli je velká komunita, závislost na knihovnách třetích stran může být nevýhodou, pokud se některé z nich přestanou udržovat.

2.2.4.3 Angular

Mezi **výhody** patří:

- **Kompletní řešení:** Angular poskytuje kompletní řešení pro vývoj aplikací včetně routování, stavového řízení a testování.
- **Silná typová kontrola:** Angular je napsaný v TypeScriptu, což umožňuje silnou typovou kontrolu a zvyšuje bezpečnost kódu.
- **Korporátní podpora:** Angular je vyvíjen a podporován společností Google, což poskytuje jistotu v dlouhodobém výhledu.

Mezi **nevýhody** patří:

- **Složitost:** Angular je obecně považován za složitější než jiné frameworky, což může pro začátečníky být obtížné.
- **Větší velikost:** Aplikace napsané v Angularu mohou mít větší velikost než aplikace napsané v jiných frameworkách, což může ovlivnit výkon.
- **Omezená flexibilita:** Angular je poměrně striktní framework, což může ztížit experimentování a prototypování.

2.2.4.4 NextJS

Mezi **výhody** patří:

- **Server-side rendering:** Next.js podporuje server-side rendering, což přináší výhody výkonu a SEO.

- **Statické generování:** Umožňuje generování statických stránek, což zlepšuje rychlost načítání a uživatelskou zkušenost.
- **Jednoduché nasazení:** Next.js umožňuje snadné nasazení aplikací na různé platformy, což zkracuje čas potřebný k uvádění aplikací do provozu.

Mezi **nevýhody** patří:

- **Složitější konfigurace:** Pro některé vývojáře může být konfigurace Next.js složitější než u jiných frameworků.
- **Omezená flexibilita:** Přestože Next.js poskytuje mnoho funkcí, některé pokročilé scénáře mohou vyžadovat dodatečnou konfiguraci.

2.2.4.5 NuxtJS

Mezi **výhody** patří:

- **Server-side rendering:** Stejně jako Next.js, Nuxt.js podporuje server-side rendering, což zlepšuje výkon a SEO.
- **Jednoduchá konfigurace:** Nuxt.js poskytuje jednoduchou a srozumitelnou konfiguraci, což usnadňuje vývoj a údržbu aplikací.
- **Modulární architektura:** S modulární architekturou je snadné integrovat různé funkce a rozšíření do projektu.

Mezi **nevýhody** patří:

- **Menší komunita:** Nuxt.js má menší komunitu než jiné frameworky, což může ovlivnit dostupnost rozsáhlejších knihoven a nástrojů.
- **Méně dokumentace:** I když Nuxt.js má solidní dokumentaci, může být méně obsáhlá než dokumentace jiných frameworků.

[5–9]

2.3 Uživatelské prostředí

Uživatelské prostředí webové aplikace je klíčovým prvkem pro uživatelskou přívětivost a efektivní používání. Níže jsou detailněji popsány klíčové aspekty tohoto prostředí.

2.3.1 Komunikace s BE

Pro efektivní komunikaci s backendem lze využít již připravenou funkci `fetch()`. Nicméně existují i alternativní balíčky pro volání API požadavků, jako je například `axios`.

2.3.1.1 Fetch

- **Základní funkce**
 - poskytuje základní funkcionality pro asynchronní získávání dat z API
 - součástí standardního JavaScriptu bez potřeby instalace externích knihoven.
- **Jednoduchost**
 - jednoduchý syntax, vhodný pro jednoduché případy použití
 - vynikající pro základní HTTP požadavky
- **Flexibilita**
 - nabízí flexibilitu při manipulaci s HTTP požadavky a odpověďmi
 - vyžaduje pečlivou správu asynchronních operací pomocí Promise

2.3.1.2 Axios

- **Externí balíček**
 - externí knihovna, která zjednodušuje a rozšiřuje možnosti komunikace s BE
 - nabízí komplexnější a přívětivější rozhraní než **fetch**
- **Snadná konfigurace**
 - poskytuje jednoduchý způsob konfigurace HTTP požadavků
 - podporuje automatické zpracování odpovědí, včetně převodu JSON dat
- **Funkce**
 - Obsahuje užitečné funkce, jako je storno žádostí, správa mezičasových údajů a automatická transformace odpovědí

Volba mezi fetch a axios závisí na konkrétních požadavcích projektu. Zatímco fetch je vhodný pro jednoduché scénáře, axios nabízí rozšířenou funkcionalitu a uživatelsky přívětivější rozhraní pro složitější projekty.

Obě možnosti jsou široce využívány v moderním vývoji webových aplikací a volba tedy záleží na požadavcích projektu nebo vývojáře. [10, 11]

2.3.2 Zabezpečení dat uživatele

První krok směřuje k zabezpečení uživatelských dat. Registrace a přihlašování probíhá pomocí uživatelského jména a hesla. Pro optimalizaci uživatelské zkušenosti a bezpečnosti by mělo být implementováno používání tokenů. Access token a refresh token s přesně definovaným datem a časem expirace poskytnou pohodlný přístup s automatickým přihlášením při návštěvě, pokud předchozí odhlášení nenastalo. Na straně FE by se ukládání těchto citlivých údajů (jako jsou tokeny, uživatelské jméno, ...) mělo zabezpečit tak, aby nebylo snadné jejich čtení. Jedním způsobem může být implementace hashovací funkce a data ukládat do LocalStorage nebo Cookies. Další možností je využití balíčku „react-secure-storage“, který má již tyto funkce implementované a stačí je využít.

2.3.2.1 React-secure-storage

Balíček React Secure Storage byl vytvořen k bezpečnému zápisu dat do lokálního úložiště. Je to obal nad standardním localStorage, který umožňuje bezpečný zápis dat. Každý prohlížeč získává svůj bezpečný klíč, kterým jsou data šifrována. Tímto způsobem jsou data šifrována a dešifrována pouze v rámci jednoho prohlížeče.

React Secure Storage podporuje několik datových typů včetně řetězce, objektu, čísla a logické hodnoty. Tím odpadá potřeba explicitní konverze všech dat na řetězec.

Balíček je napsán ve vzoru Singleton, což zajišťuje rychlé čtení dat při inicializaci knihovny. Klíč je generován pomocí otisku prohlížeče, který je vytvořen z více než 10 identifikátorů prohlížeče a bezpečného klíče zadávaného uživatelem.

Bezpečný klíč pro konkrétního uživatele lze konfigurovat pomocí souboru .env pod proměnnou SECURE_LOCAL_STORAGE_HASH_KEY. Také lze konfigurovat prefix pro bezpečné lokální úložiště pomocí .env souboru pod proměnnou SECURE_LOCAL_STORAGE_PREFIX. [12]

2.3.3 Přehled a zobrazení filmů

Po úspěšném přihlášení bude uživateli prezentována stránka dashboard/přehled. Tento přehled by měl obsahovat 20 naposledy přidaných filmů, 20 nejlépe hodnocených filmů a 20 nejlépe hodnocených filmů od ostatních uživatelů, které slouží pro inspiraci. Tato struktura poskytuje uživateli rychlý přehled o nových, oblíbených a populárních filmech.

2.3.4 Správa filmů

Uživatel má možnost kompletní správy svých filmů. Může procházet všechny záznamy, používat filtry, zobrazit detaily filmu a provádět akce, jako je přidání filmu do oblíbených nebo jeho sdílení s ostatními uživateli. Dále by mělo být uživateli umožněno přidávat nové filmy, editovat existující nebo odstraňovat ty, které již nechce udržovat ve své "databázi".

2.3.5 Volba jazyka

Aplikace respektuje jazykové preference uživatele. Ten má možnost vybrat si alespoň ze dvou základních jazyků: čeština a angličtina. Tím se zajišťuje, že uživatel bude moci využívat aplikaci ve svém preferovaném jazyce. K tomuto účelu by mohl být využit balíček i18n.

I18next je knihovna pro internacionalizaci v JavaScriptu, která usnadňuje implementaci mezinárodních funkcí ve webových a mobilních aplikacích. Je navržena tak, aby byla flexibilní a snadno použitelná pro vývojáře, kteří potřebují integrovat vícejazyčnou podporu do svých projektů. [13]

Umožňuje jednoduchou správu lokalizačních řetězců a poskytuje možnost definovat různé jazyky a klíče pro překlady. S využitím jednoduchého API mohou vývojáři načítat překlady, měnit jazyky a používat lokalizované řetězce ve svých

aplikacích. Knihovna podporuje načítání překladů z různých zdrojů, jako jsou JSON soubory, backendové služby nebo dokonce databáze. [13]

Díky i18next mohou vývojáři implementovat internacionalizaci do svých projektů bez zbytečné složitosti. Tato knihovna je široce využívána ve světě vývoje webových a mobilních aplikací a poskytuje efektivní a efektivní způsob, jak přizpůsobit aplikace potřebám uživatelů z různých jazykových prostředí. [13]

Pro usnadnění procesu překladu a lepší správu těchto překladů bude v projektu využíván nástroj i18nexus.

I18nexus je nástroj navržený pro efektivní správu mezinárodních lokalizací a překladů ve webových a mobilních aplikacích. Jeho hlavním zaměřením je usnadnit proces internacionalizace a lokalizace softwaru, umožňující poskytování obsahu v různých jazycích a kulturních kontextech. Tento nástroj poskytuje řadu funkcí, mezi něž patří správa lokalizačních souborů, online překladový editor, synchronizace s verzovacími repositáři, podpora pro vývojáře, statistiky a reporty o stavu lokalizace a API pro automatizaci některých procesů. [14]

Umožňuje nahrávat, sledovat a aktualizovat lokalizační soubory a poskytuje překladatelům online editor pro pohodlný proces překladu. Integrace s verzovacími repositáři usnadňuje synchronizaci mezi kódem a lokalizačními soubory. Vývojáři mohou využívat nástroje pro implementaci internacionalizace ve svých projektech, a uživatelé získávají přehledné statistiky o stavu lokalizace a překladů. I18nexus také poskytuje API pro automatizaci některých procesů, což může být užitečné pro pokročilé scénáře integrace. Celkově přispívá k efektivní správě mezinárodních verzí softwaru a zvyšuje jeho dostupnost pro uživatele různých jazyků a regionů. [14]

2.3.6 Design a responsibilita

Design webové aplikace je klíčovým prvkem pro uživatelskou přívětivost. Kromě obecných principů designu můžeme využívat specializované grafické frameworky. Material-UI (MUI), Ant Design, Next UI, TailwindCSS a další poskytují sadu předpřipravených komponent a stylů, které usnadňují vytváření vizuálně

atraktivních a konzistentních uživatelských rozhraní. Výběr vhodného grafického frameworku závisí na designových preferencích, funkcionalitě a potřebách uživatele.

2.3.6.1 Material–UI (MUI)

Material-UI (MUI), s inspirací v designovém jazyce Material Design od Google, nabízí moderní a flexibilní prvky pro tvorbu uživatelských rozhraní.

Výhody:

- **Moderní design:** MUI poskytuje moderní designové prvky a komponenty inspirované designovým jazykem Material Design od Google.
- **Široká podpora:** MUI má rozsáhlou komunitu a je oblíbený mezi vývojáři, což znamená dostupnost mnoha připravených komponent.
- **Flexibilita:** MUI je navržen tak, aby byl flexibilní a přizpůsobitelný potřebám různých projektů.

Nevýhody:

- **Velikost bundle:** V některých případech může použití MUI zvětšit velikost výsledného bundle (celková velikost seskupení souborů), což může ovlivnit rychlost načítání.

2.3.6.2 Ant Design

Ant Design poskytuje komplexní designový systém s konzistentním vzhledem a chováním.

Výhody:

- **Komplexní designový systém:** Ant Design není pouze knihovna komponent, ale celý designový systém s konzistentním vzhledem a chováním.
- **Rozsáhlá dokumentace:** Má podrobnou a srozumitelnou dokumentaci, což usnadňuje integraci a používání.
- **Podpora pro React a Vue:** Ant Design poskytuje komponenty pro React i Vue, což zvyšuje jeho flexibilitu.

Nevýhody:

- **Designová konvence:** Při použití Ant Design by měly aplikace dodržovat designové konvence této knihovny, což může být omezující v případě odlišných designových požadavků.

2.3.6.3 NextUI

Next UI je optimalizován pro integraci s frameworkem Next.js a zajišťuje rychlý vývoj s připravenými komponenty.

Výhody:

- **Jednoduchá integrace s Next.js:** Next UI je navržen tak, aby snadno integroval s frameworkem Next.js, což zlepšuje vývoj a rychlost implementace.
- **Rychlý vývoj:** Poskytuje připravené komponenty, které urychlují vývoj a snižují potřebu psaní vlastního stylu.

Nevýhody:

- **Omezená komunita:** Má menší komunitu ve srovnání s některými jinými designovými knihovnami, což může ovlivnit dostupnost aktualizací a rozšíření.

2.3.6.4 TailwindCSS

Tailwind CSS, i když neplnohodnotný framework, umožňuje flexibilní vývoj bez vlastního overheadu.

Výhody:

- **Nízký vlastní overhead:** Tailwind CSS není plnohodnotný framework, což znamená nižší vlastní overhead v porovnání s plnohodnotnými CSS frameworky.
- **Flexibilita:** Umožňuje vývojářům snadno vytvářet vlastní design bez narůstající komplexity připravených komponent.

Nevýhody:

- **Složitost:** Může být náročnější pro začátečníky, kteří se musí naučit syntaxi a filozofii Tailwind CSS.

- **Ruční psaní stylů:** Ve srovnání s designovými knihovnami vyžaduje více psaní vlastního CSS, což může být časově náročnější.

[15][16][17][18]

2.3.7 Export a import filmů

Export a import filmů představují klíčové prvky webové aplikace, zajišťující efektivní manipulaci s filmovými daty uživatelů. Tyto funkce jsou klíčové zejména pro uživatele, kteří disponují rozsáhlým množstvím filmů, kde manuální přidávání jednotlivých filmů do systému je prakticky nemožné z časového hlediska.

2.3.7.1 Formáty a šablony

Import filmů bude podporovat soubory ve formátech JSON a CSV. Pro formát CSV bude k dispozici předem připravená šablona v GoogleSheet, což zjednoduší strukturování dat pro uživatele při importu.

2.3.7.2 Balíčky

Pro optimalizaci uživatelské interakce a poskytnutí plynulého procesu importu filmů do webové aplikace bude implementován balíček s názvem "react-drag-drop-files". Tento balíček přináší výhodu umožnění uživateli provádět operace Drag&Drop souborů přímo do prostředí webové aplikace.

Pomocí tohoto balíčku bude uživateli umožněno intuitivně a efektivně přetahovat a upouštět soubory, což vytváří uživatelsky příjemné prostředí pro import filmů. Tato funkcionality přináší snazší ovládání a zvyšuje celkovou uživatelskou spokojenost.

Implementace tohoto balíčku posiluje uživatelskou přívětivost celého procesu importu a přispívá k pohodlnějšímu a efektivnějšímu využívání webové aplikace pro správu filmové knihovny. [19]

3 Návrhová část

3.1 Obecné požadavky

1. Podpora mobilních zařízení
 - Aplikace bude vyvíjena s ohledem na plnou responzivitu, aby poskytovala uživatelům pohodlný a konzistentní zážitek z používání bez ohledu na zařízení, na kterém je aplikace spuštěna
2. Dostupnost
 - Cílem je zajistit, aby aplikace byla dostupná pro všechny uživatele s připojením k internetu, bez ohledu na to, zda používají počítače, tablety nebo mobilní telefony

3.2 Funkční požadavky

1. Prohlížení obsazení
 - Uživatelé budou mít možnost prohlížet obsazení herci ve filmech, což jim poskytne důležité informace o obsazení jednotlivých rolí
2. Trailery
 - Implementace možnosti pro přehrávání trailerů filmů umožní uživatelům získat lepší náhled na obsah filmů před jejich zhlédnutím
3. Manipulace s filmy
 - Uživatelé budou mít možnost přidávat, mazat a upravovat informace o filmech, což jim umožní personalizovanou správu jejich filmové kolekce a lepší organizaci

Tyto rozšířené funkční požadavky dodávají aplikaci další rozměr, umožňující uživatelům plně využívat a spravovat filmy nejen z počítačů, ale i z mobilních zařízení, s důrazem na dostupnost a uživatelskou přívětivost.

3.3 Zvolené technologie

Na základě provedené analýzy byly vybrány následující technologie. Aplikace bude typu [Single Page Application \(SPA\)](#) pro jednoduchost a rychlost zobrazení jednotlivých stránek. Pro implementaci webové aplikace bude zvolen framework [NextJS](#). Tato volba vychází z jeho rozšířenosti na trhu a poptávky firem po vývojářích, kteří jsou znalí práce s tímto frameworkem. Dalším důvodem pro výběr NextJS je i fakt, že autor této práce nemá žádné předchozí zkušenosti s tímto frameworkem, což mu umožní se s ním seznámit a posoudit jeho vhodnost a použitelnost.

Co se týče grafických balíčků, budou využity [Material-UI \(MUI\)](#), [NextUI](#) a [TailwindCSS](#). Material-UI byl vybrán z důvodu, že autor má s tímto balíčkem zkušenosti z předchozích prací. Obsahuje téměř všechny komponenty, které budou do této práce potřeba. Dalším grafickým balíčkem bude NextUI, který přidá moderní vzhled vyvíjené aplikaci. Bohužel NextUI neobsahuje všechny potřebné komponenty, a proto bude využit ve spojení s MUI. Posledním podpurným balíčkem bude TailwindCSS. Jak bylo zmíněno v analýze, nejedná se o framework poskytující předem vytvořené komponenty, ale o balíček usnadňující stylování při tvorbě nebo úpravě komponent.

3.4 Uživatelské rozhraní

3.4.1 Konceptuální design

V designu aplikace se budeme řídit klíčovými principy, které nám pomohou vytvořit moderní, uživatelsky přívětivé prostředí, které je jednoduché, přehledné a esteticky přitažlivé. Zde jsou hlavní designové principy a cíle, které budou aplikovány:

Použití barev:

Základní barvou aplikace bude světle šedá, která dodává aplikaci neutrální a elegantní vzhled. Tlačítka a filtry budou řízeny barevným standardem success, warning, danger/error komponent, což umožní rychlou identifikaci a orientaci uživatele v aplikaci.

Typografie:

Pro textový obsah bude zvolen font Panton s podporou diakritiky, který dodá aplikaci profesionální vzhled a přispívá k celkové estetice. Důraz je kladen na správnou volbu velikosti a stylu písma, což zvyšuje čitelnost a přehlednost textu.

Grafika:

Grafika aplikace bude minimalistická, uživatelsky přívětivá a přehledná. Základním prvkem grafiky bude filmová karta, která slouží jako univerzální prvek a je použita na více místech v aplikaci. Tato filmová karta poskytuje uživatelům snadný a intuitivní přístup k důležitým informacím o filmech, jako jsou název, obrázek, hodnocení, a další relevantní údaje. Je navržena tak, aby byla jednotná a konzistentní ve všech částech aplikace, což zlepšuje uživatelskou orientaci a plynulost používání.

Logo aplikace bude vytvořeno pomocí umělé inteligence (AI, konkrétně MidJourney), což umožní vytvořit unikátní vizuální identitu. Grafika aplikace bude minimalistická a moderní s jednoduchými formami, což zvyšuje uživatelskou přívětivost.

Navigace:

Navigační panel na levé straně obrazovky poskytne snadný přístup k hlavním funkcím a stránkám aplikace. Pro mobilní zařízení bude tento panel upraven tak, aby byl optimalizován pro menší obrazovky, což zajistí plynulý a intuitivní uživatelský zážitek.

Využití MaterialUI a NextUI balíčků:

Pro design a stylování uživatelského rozhraní budou využity balíčky MaterialUI a NextUI. Tyto balíčky poskytnou moderní prvky a komponenty, které budeme integrovat do naší aplikace, což usnadní vývoj a přispěje k celkové konzistenci a kvalitě designu.

3.4.2 Základní interakční prvky

Carousel

Carousel je interaktivní prvek uživatelského rozhraní, který umožňuje uživatelům procházet obsah (v tomto případě filmy) jednoduchým posouváním. Uživatelé mohou snadno procházet filmy pomocí gest a tlačítek pro předchozí nebo další položku. Tento prvek přináší uživatelům příjemný zážitek z prohlížení obsahu a umožňuje jim rychle najít filmy, které je zajímají.

Navigační panel

Navigační panel je důležitým prvkem aplikace, který poskytuje uživatelům snadný přístup k hlavním funkcím a stránkám aplikace. Obsahuje odkazy na domovskou stránku, seznamy filmů, vyhledávání a další klíčové oblasti. Tento panel zajišťuje uživatelům plynulý pohyb po aplikaci a usnadňuje jim orientaci v aplikaci.

Akční panel na kartě filmu

Akční panel na kartě filmu je interaktivní prvek, který umožňuje uživatelům provádět různé akce spojené s daným filmem. Zahrnuje možnosti úpravy/editace informací o filmu, jeho smazání, změnu za jiný film, přidání do oblíbených či mezi sdílené. Tento panel poskytuje uživatelům kontrolu nad jejich filmovou kolekcí a umožňuje jim snadno spravovat a personalizovat svůj obsah.

Filtry

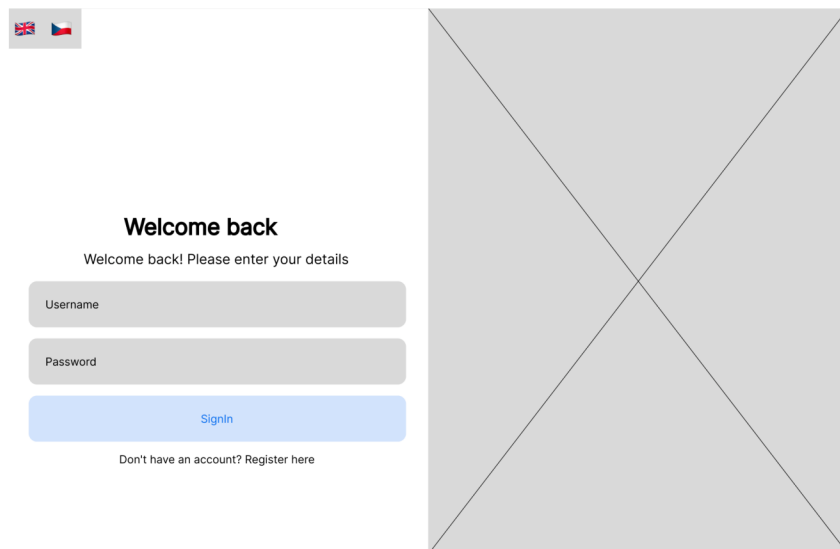
Filtry jsou důležitým prvkem pro vyhledávání konkrétních žánrů nebo filmů v aplikaci. Uživatelé mohou pomocí filtrů omezit zobrazení filmů na základě určitých kritérií, jako je žánr, uložení atd. Tento prvek umožňuje uživatelům rychle najít filmy, které odpovídají jejich preferencím a zájmům, a zlepšuje jejich uživatelský zážitek z procházení aplikace.

3.4.3 Wireframy

Pro lepší představu o struktuře a uspořádání aplikace jsou připraveny wireframy, což jsou jednoduché náčrty, které zobrazují rozložení jednotlivých obrazovek a základní funkčnost aplikace.

Přihlášení

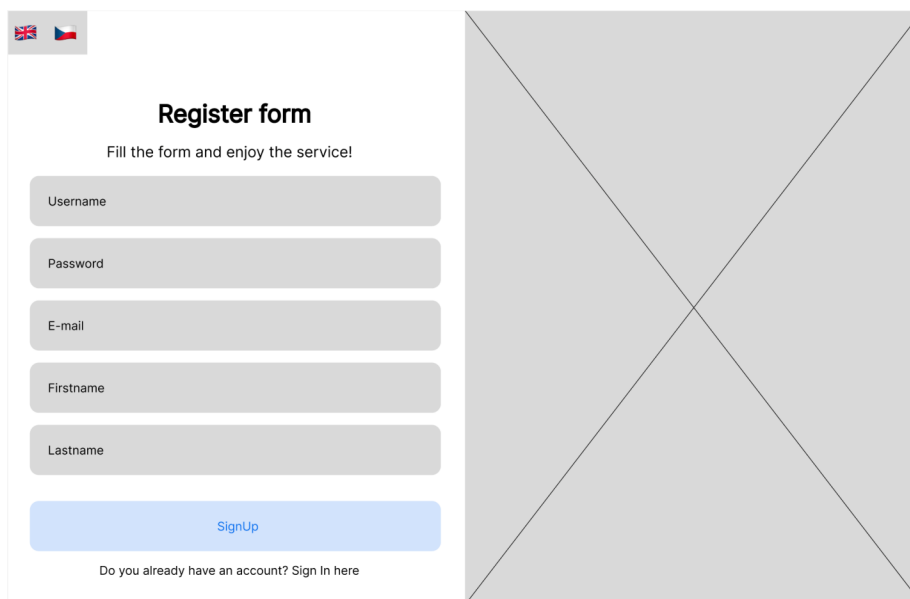
Tato obrazovka je vertikálně rozdělena na dvě části. V první části se nachází formulář pro přihlášení uživatele, který zahrnuje pole pro uživatelské jméno a heslo, a tlačítko pro odeslání přihlašovacích údajů. Druhá část (napravo) obsahuje carousel s aktuálně populárními filmy pro inspiraci.



Obr. 1 Wireframe – Přihlášení

Registrace

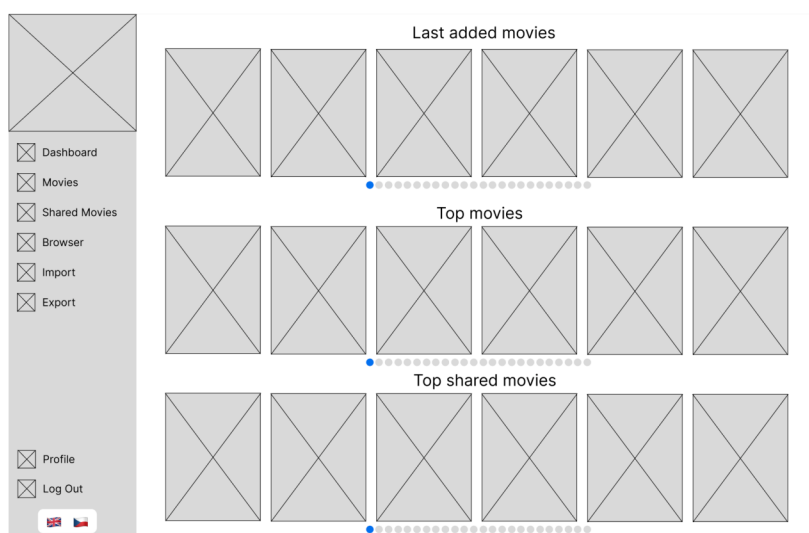
Tato obrazovka je taktéž rozdělena na dvě vertikální části, aby byla zachována konzistence s přihlašovací stránkou. V první části uživatelé mohou vyplnit formulář pro vytvoření nového účtu v aplikaci. Formulář zahrnuje pole pro základní informace o uživateli a tlačítko pro potvrzení registrace. Požadované informace o uživateli jsou přezdívka, heslo, email, jméno a příjmení. Druhá část (napravo) obsahuje carousel s aktuálně populárními filmy pro inspiraci.



Obr. 2 Wireframe – Registrace

Přehled

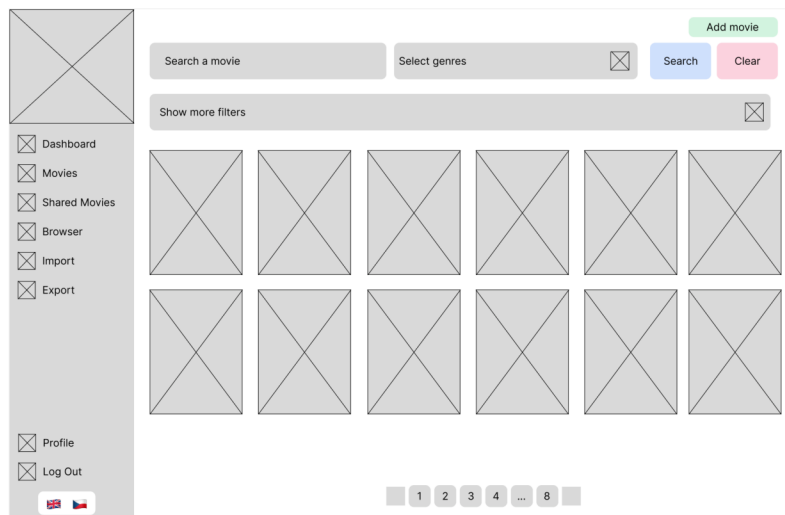
Tato obrazovka zobrazuje základní přehled filmů pomocí carouselu, který zahrnuje kategorie Naposledy přidané filmy, Nejlepší filmy a Nejlepší sdílené filmy. Uživatelé mohou jednoduše posouvat mezi různými kategoriemi a prohlížet si tak své/sdílené filmy.



Obr. 3 Wireframe – Přehled

Filmy

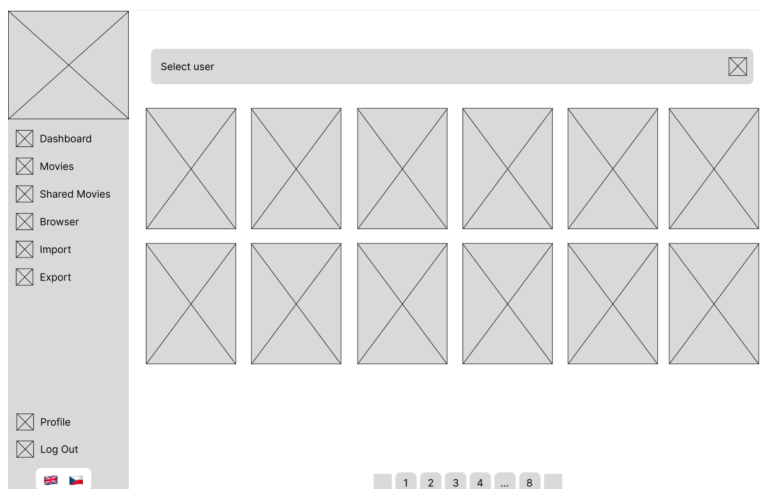
Zde uživatelé mohou procházet svou sbírku filmů a zobrazit detaily každého filmu. Každý film je zobrazen formou obrázku daného filmu. Uživatel má možnost využít filtry pro vyhledání konkrétního filmu nebo filmových žánrů. Dále se zde nachází tlačítko pro přidání nového filmu do své kolekce.



Obr. 4 Wireframe – Filmy

Sdílené filmy

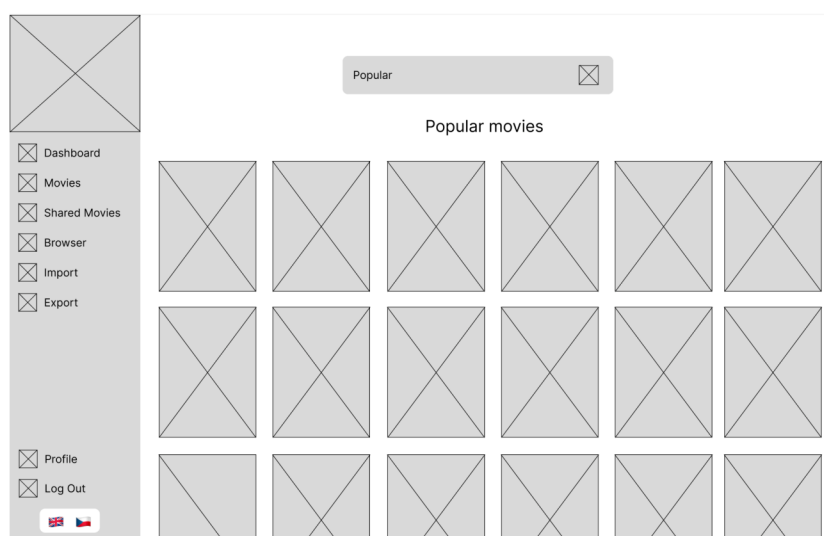
Uživatelé zde mohou najít filmy, které byly sdíleny ostatními členy komunity. Obrazovka obsahuje základní filtry, které umožňují uživatelům filtrovat filmy podle různých kritérií, včetně filtrů na základě uživatelů, kteří tyto filmy sdíleli. Díky tomuto filtru mohou uživatelé snadno najít filmy, které byly sdíleny konkrétními osobami, a prohlédnout si je.



Obr. 5 Wireframe – Sdílené filmy

Prohlížeč

Tato stránka zobrazuje uživateli filmy dle vybraného filtru, který může být nastaven na Oblíbené, Nadcházející nebo Nejlépe hodnocené filmy. Filmy jsou získávány z databáze filmů TMDB, což uživatelům poskytuje širokou škálu filmových titulů k prohlížení. Hlavním účelem této stránky je poskytnout uživatelům inspiraci a možnost objevovat nové filmy, které by je mohly zaujmout. Uživatelé mohou jednoduše procházet dostupné filmy a vybírat si ty, které chtějí zobrazit nebo si o nich přečíst více informací.



Obr. 6 Wireframe – Prohlížeč

Nahrát

Tato obrazovka nabízí uživateli možnost nahrát soubor ve formátu JSON nebo CSV, což umožňuje rychle a efektivně přidat svou sbírku filmů do aplikace. Uživatelé nemusí postupně přidávat filmy manuálně, ale mohou je nahrát v hromadném formátu, což ušetří čas a usnadní proces správy jejich filmové kolekce. Tímto způsobem mohou uživatelé snadno importovat své filmy do aplikace a začít je spravovat a prohlížet v rámci prostředí aplikace.



Obr. 7 Wireframe – Nahrát

Stáhnout

Tato stránka umožňuje uživatelům exportovat nebo stáhnout filmy do formátu CSV pro lokální prohlížení filmů. Export do formátu CSV umožňuje uživatelům otevřít svou sbírku filmů například v Google Sheets nebo Excelu, kde mohou snadno prohlížet, upravovat nebo analyzovat své filmy podle svých potřeb. Tímto způsobem mají uživatelé možnost efektivněji pracovat se svou filmovou sbírkou mimo prostředí aplikace, pokud to potřebují.



Obr. 8 Wireframe – Stáhnout

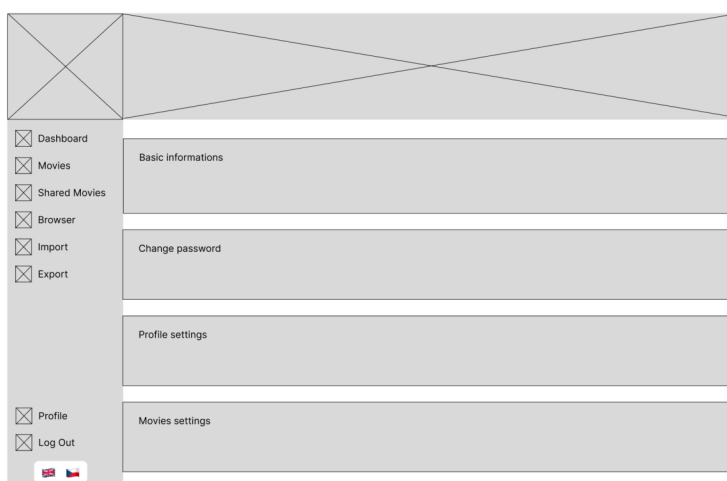
Profil

Stránka Profil poskytuje uživatelům možnost úpravy svých osobních údajů a nastavení souvisejících s jejich filmovou sbírkou. Uživatel zde může nalézt několik sekcí, které usnadňují správu jejich účtu.

Na prvním místě je sekce Základní informace, kde může uživatel upravit své základní údaje, jako je například jméno a příjmení. Následuje sekce Změna hesla, kde uživatel má možnost změnit své současné heslo na jiné, což posiluje bezpečnost jejich účtu.

Další důležitou částí je sekce Nastavení profilu, kde uživatelé mohou nastavit svůj profil jako veřejný/sdílený nebo mají možnost smazat všechny filmy ze své sbírky, což jim poskytuje větší kontrolu nad tím, jak jsou jejich filmy sdíleny ostatním uživatelům.

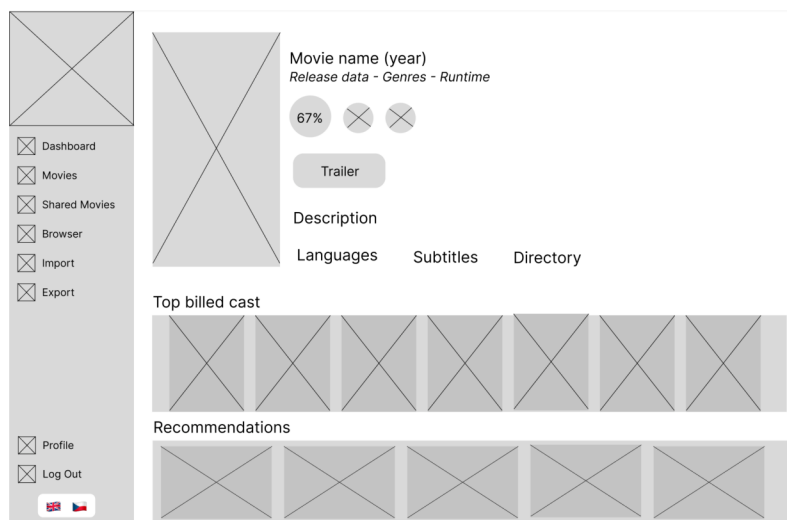
Poslední sekce je Nastavení filmů, kde uživatelé mohou provádět různá nastavení související s filmy, jako je například nastavení složek, jazyků nebo titulků aplikace. Tato část umožňuje uživatelům personalizovat svou filmovou sbírku a přizpůsobit ji svým preferencím. Celkově stránka Profil slouží jako centrální místo, kde uživatelé mohou pohodlně spravovat svůj účet a upravovat svou filmovou sbírku podle svých potřeb.



Obr. 9 Wireframe - Profil

Detail filmu

Zde uživatelé najdou podrobné informace o konkrétním filmu, včetně obsazení, hodnocení a dalších detailů.



Obr. 10 Wireframe – Detail filmu

4 Implementace

Tato kapitola je zaměřena na popis postupů vývoje, problémů během vývoje a popisu funkcionalit, které aplikace přináší.

4.1 Založení projektu

4.1.1 Verzovací nástroj

Pro sledování a ukládání změn v kódu bylo rozhodnuto použít verzovací systém Git a hostovací službu GitHub. Git umožňuje systematické verzování projektu, zatímco GitHub usnadňuje spolupráci a správu změn.

4.1.2 Vývojové prostředí

Jako hlavní nástroj pro psaní kódu bylo vybráno Visual Studio Code (VSCode). Tento bezplatný editor je oblíbený pro svou přizpůsobitelnost a širokou škálu rozšíření, což výrazně podporuje efektivitu vývojářů.

4.1.3 Inicializace projektu NextJS

Pro rychlý start vývoje byl použit příkaz `npx create-next-app@latest`. Tento příkaz umožňuje jednoduchou inicializaci NextJS aplikace s možností konfigurace podle specifických požadavků projektu.

```
1 What is your project named? my-app
2 Would you like to use TypeScript? No / Yes
3 Would you like to use ESLint? No / Yes
4 Would you like to use Tailwind CSS? No / Yes
5 Would you like to use `src/` directory? No / Yes
6 Would you like to use App Router? (recommended) No / Yes
7 Would you like to customize the default import alias (@/*)? No / Yes
8 What import alias would you like configured? @/*
```

Obr. 11 Inicializace projektu NextJs

4.2 Implementace stránek a komponent

4.2.1 Komponenta výběru jazyka a překlad stránek

Při implementaci webových stránek je klíčové zajistit správný výběr jazyka, což poskytuje uživatelům pohodlnou navigaci v rámci aplikace. Z tohoto důvodu byla vyvinuta a implementována speciální komponenta pro výběr jazyka a překlad stránek.

Tato komponenta umožňuje uživatelům snadno měnit jazyk aplikace podle svých preferencí. Nabízí přehlednou možnost výběru jazyka a zároveň zajistí efektivní překlad obsahu stránek. Uživatelé tak mohou plynule přepínat mezi různými jazykovými verzemi aplikace, což výrazně zvyšuje uživatelskou přívětivost.

Implementace této komponenty respektuje moderní postupy v oblasti internacionalizace a lokalizace webových aplikací. Zajišťuje jednoduchý a intuitivní proces výběru jazyka, což přispívá k celkově kvalitnímu uživatelskému zážitku při používání aplikace.

4.2.1.1 Balíčky

Pro efektivní implementaci přepínání textů byl vybrán balíček `i18next`. Tento nástroj umožňuje jednoduché a flexibilní řešení pro lokalizaci a překlady v rámci webové aplikace. Konkrétní nastavení balíčku bylo specifikováno v konfiguračním souboru `next-i18next.config`:

```
i18n: {  
  locales: ['en', 'cs'],  
  defaultLocale: 'en',  
  localeDetection: false,  
}
```

Zde bylo definováno, že aplikace podporuje dva jazyky (angličtina a čeština) a výchozím jazykem je angličtina.

4.2.1.2 Implementace komponenty

Vytvoření komponenty pro výběr jazyka umožňuje uživateli snadné přepínání mezi angličtinou a češtinou. Výchozím jazykem je angličtina, a uživatel může provádět výběr kliknutím na ikonu odpovídající vlajky.



Obr. 12 Přepínání jazyků

```
<div className="hidden md:flex items-center bg-brown-200 rounded-lg text-sm font-bold selector justify-center">
  <Link href={router.asPath} locale="en" className={button-lang${router.locale === 'en' ? ' active' : ''}}>
    <Image src={flagEn} width="50" height="40" draggable="false" alt="English" className="pt-1" />
  </Link>
  <Link href={router.asPath} locale="cs" className={button-lang${router.locale === 'cs' ? ' active' : ''}}>
    <Image src={flagCz} width="50" height="40" draggable="false" alt="Čeština" className="pt-1" />
  </Link>
</div>
```

Po výběru jazyka se automaticky aktualizuje URL adresa stránky podle zvoleného jazyka. Překlad textů v aplikaci je založen na dynamickém zobrazení konkrétních řetězců v souladu s vybraným jazykem.

4.2.2 Přihlašovací stránka

Přihlašovací stránka webové aplikace poskytuje uživateli několik funkcionalit, které zahrnují přihlášení, registraci a změnu jazyka. Následující kroky detailně popisují implementaci této stránky.



Welcome back

Welcome back! Please enter your details.

Don't have an account? [Register here](#)

Obr. 13 Přihlašovací formulář

4.2.2.1 Struktura přihlašovací stránky

Uživatel na přihlašovací stránce nalezne dvě pole pro vyplnění: Uživatelské jméno a Heslo. Existuje také možnost přepnutí jazyka zobrazené stránky. Pro ty, kteří nemají účet, je k dispozici odkaz "Registrujte se zde", který vede na registrační formulář.



Register form

Fill the form and enjoy the service!

Do you already have an account? [Sign In here](#)

Obr. 14 Registrační formulář

4.2.2.2 Komunikace s BE

Při snaze o přihlášení nebo registraci probíhá důležitá komunikace s backendem (BE), který zajišťuje správný chod a bezpečnost aplikace. V rámci procesu registrace je pečlivě ověřováno, zda zadaný email a uživatelské jméno nejsou již obsazené v systému. Tato kontrola zabraňuje vytváření duplicitních účtů a přispívá k celkové integritě uživatelských dat.

Při přihlašování probíhá komunikace s backendem prostřednictvím POST API volání, kde je prováděna kontrola shody uživatelského jména s heslem. To zajišťuje, že uživatelé mohou bezpečně přistupovat ke svým účtům a chrání se tak před neoprávněným přístupem.

Níže jsou uvedeny části kódu pro přihlášení a registraci, které využívají axios pro provádění HTTP požadavků na backend:

Přihlášení

```
const API_URL = process.env.NEXT_PUBLIC_APP_BE_DOMAIN + 'api/auth/'
const response = await axios.post(API_URL + 'signin', {username, password})
```

Registrace

```
const API_URL = process.env.NEXT_PUBLIC_APP_BE_DOMAIN + 'api/auth/'
const response = await axios.post(API_URL + 'signup', {username, email, password, name, surname})
```

4.2.2.3 Zabezpečení přihlašovacích údajů

Pokud jsou přihlašovací údaje správné, klient (FE) ukládá data do LocalStorage pomocí balíčku [react-secure-storage](#). Tato data jsou zašifrována, což zvyšuje bezpečnost a znesnadňuje neautorizovaný přístup.

4.2.2.4 Přesměrování

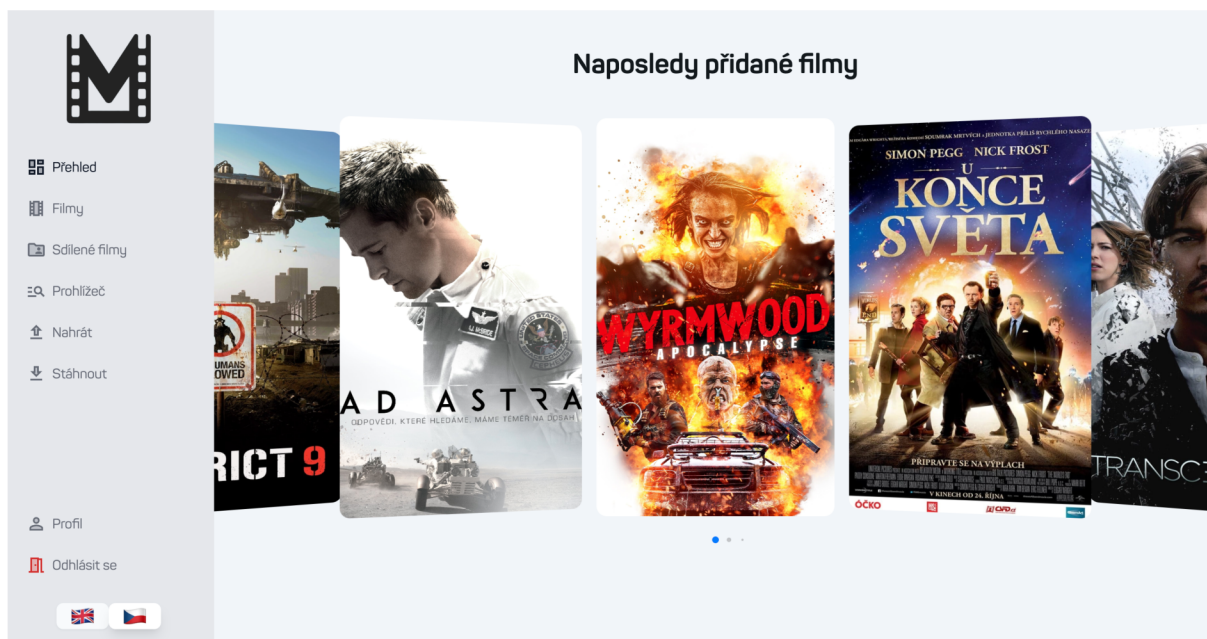
Pro zajištění plynulého uživatelského zážitku bylo implementováno přesměrování na různých úrovních uživatelské interakce. Pokud není uživatel přihlášen a pokusí se přistoupit k jiné stránce než přihlášení, je automaticky přesměrován na přihlašovací stránku. Tento krok přispívá k zabezpečení neoprávněného přístupu a udržuje konzistentní chování aplikace.

Po úspěšném přihlášení je uživatel přesměrován na hlavní stránku aplikace, tj. na stránku s přehledem filmů. Tato stránka poskytuje uživateli okamžitý přístup k relevantním informacím o filmech, což usnadňuje navigaci a zvyšuje efektivitu užívání aplikace.

4.2.3 Komponenta NavBar

Komponenta NavBar slouží k navigaci uživatele v rámci webové aplikace a poskytuje snadný přístup k hlavním funkcím. Obsahuje následující položky, z nichž každá je reprezentována odkazem na příslušnou stránku:

- Logo aplikace (stránka Filmy)
- Přehled
- Filmy
- Sdílené filmy
- Prohlížeč
- Nahrát
- Stáhnout
- Profil
- Odhlášení
- Výběr jazyka



Obr. 15 Navigační menu pro větší zařízení

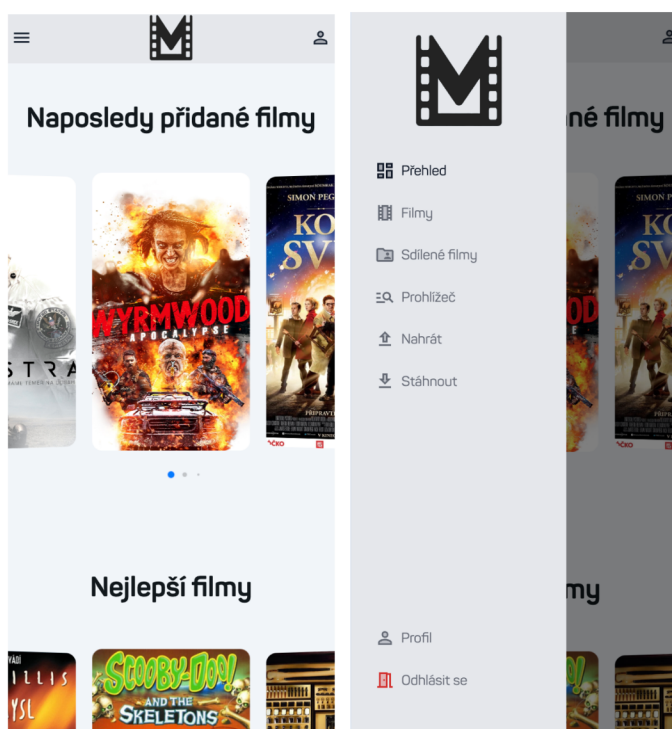
Pro tvorbu tohoto navigačního panelu byl využit balíček MUI (Material-UI) a jeho ikony pro jednotlivé položky. Každý odkaz je implementován jako Link, který reprezentuje konkrétní stránku.

```
const MenuItem = ({ icon, name, route }) => {
  const colorClass = router.pathname === route ? 'text-gray-900 dark:text-white' : 'text-gray-900/60
  hover:text-gray-900 dark:text-white/50 dark: hover:text-white';

  return (
    <Link href={route} className={flex gap-1 [&*>]:my-auto text-md pl-6 py-3 border-b-[1px] dark:border-
    b-white/10 transition-[color] ease-in-out duration-500 ${colorClass}}>
      <div className="text-xl flex [&*>]:mx-auto w-[30px]">{icon}</div>
      <div>{name}</div>
    </Link>
  );
};
```

```
<MenuItem name={t('dashboard')} route="/dashboard" icon={<DashboardTwoToneIcon />} />
<MenuItem name={t('movies')} route="/movies" icon={<LocalMoviesTwoToneIcon />} />
<MenuItem name={t('shared_movies')} route="/shared" icon={<FolderSharedTwoToneIcon />} />
```

Komponenta je implementována pro obě prostředí – desktopová i mobilní zařízení. Při dosažení určité šířky se komponenta přepíná do mobilního zobrazení, což zajišťuje optimální uživatelský zážitek a snadné ovládání na mobilních zařízeních.



Obr. 16 Navigační menu pro mobilní zařízení

4.2.3.1 Integrace komponenty pro změnu jazyka

Do NavBar komponenty byla integrována již existující komponenta pro změnu jazyka, označená jako `<LanguageSelector />`. Tato komponenta umožňuje uživateli pohodlně měnit jazykové nastavení přímo z navigačního panelu.

4.2.4 Komponenta MovieCard

Komponenta MovieCard představuje klíčový prvek v rámci uživatelského rozhraní aplikace, a to zejména na stránkách Přehled, Filmy, Sdílené filmy a Prohlížeč. Její flexibilita a rozmanité možnosti zobrazení umožňují uživatelům

snadný přístup k informacím o filmech a interakci s nimi. Níže jsou uvedeny hlavní funkce a možnosti komponenty:

1. Zobrazení obrázku filmu
 - Komponenta poskytuje vizuální prezentaci filmu pomocí zobrazení odpovídajícího obrázku
2. Přesměrování na detail filmu při kliknutí na komponentu
 - Kliknutím na komponentu uživatel přejde na detailní stránku daného filmu, což umožňuje podrobnější prozkoumání obsahu
3. Zobrazení ovládacího panelu
 - Komponenta nabízí ovládací prvky umožňující uživateli interakci s filmy, jako je odstraňování, upravování a další
4. Možnost zobrazení detailu při najetí
 - Při najetí myši na komponentu se zobrazí zkrácený detail filmu, což poskytuje uživateli rychlý náhled
5. Změna filmu za jiný
 - Uživatel má možnost dynamicky změnit zobrazený film, což usnadňuje rychlost editace

4.2.4.1 Implementace

Pro implementaci této komponenty byl zvolen Material Design. Aby komponenta mohla správně zobrazit data (obrázek, popis, ...) je zapotřebí tato data poskytnout při volání komponenty. Komponenta očekává parametr „movie“ který je jako jediný povinný.

```
<MovieCard movie={jsonObject}/>
```

Parametr movie musí obsahovat JSON data filmu, které jsou získány přes REST API z backendu. Pomocí GET API url jsou staženy veškeré filmy dle použitého API. Jedno z těchto GET API url je například.

```
api/list/movies?page=${page - 1}&size=20&search=${name}${genres2Url}
```

Page a size parametr žádá BE o vrácení konkrétní stránky, ve které má být celkem 20 záznamů (filmů). Parametr name a genres2Url jsou filtry, které si uživatel na stránce může nastavit.

Tento GET je využit na stránce Filmy a za předpokladu, že je uživatel autorizovaný (přihlášený) a má na svém profilu nahrané / přidané filmy, vrací pole objektů filmů.

Pomocí javascriptové funkce `.map()` se tyto objekty filmu postupně posílají jako parametr `movie` do komponenty `MovieCard` a tímto způsobem komponenta získá potřebná data pro zobrazení dat uživateli.

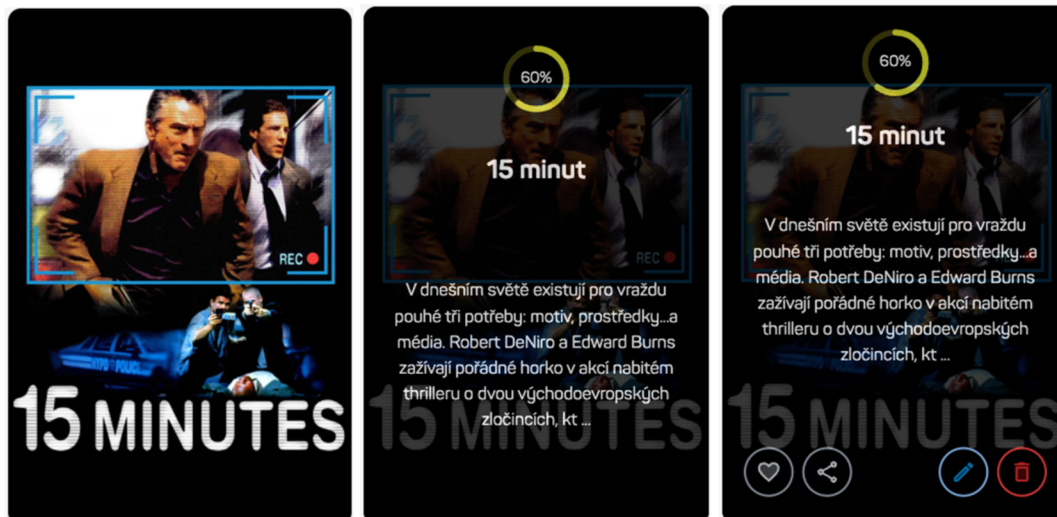
Další parametry této komponenty jsou nepovinné. Jedná se parametry:

```
<MovieCard movie={jsonObject} key={index} removeMovieFromList={deleteMovieFromList}
directoriesData={directories} languagesData={languages} subtitlesData={subtitles} action hover
detailAvailable />
```

Key se využívá při použití funkce `map()` aby každá zobrazená karta/komponenta měla unikátní klíč. Zde se nastaví klíč podle indexu (pořadí zobrazení karty od čísla 0). Parametry `removeMovieFromList`, `editMovie`, `directoriesData`, `languagesData` a `subtitlesData` jsou povinné parametry, pokud se využije parametr `action`. Při použití `action` by měl být správně použit i parametr `hover`. Pokud `hover` není do komponenty zadán a je použit parametr `action`, žádná změna by se na kartě nestala, jelikož panel `action` se zobrazuje při najetí kurzoru na kartu. Posledním parametrem je `detailAvailable`, který umožní uživateli po kliknutí na kartu zobrazit kompletní detail filmu.

Jak již bylo zmíněno, `action` zobrazí uživateli control panel, kterým je schopen film nastavovat, upravovat a mazat. Po kliknutí na ikonu tužky může uživatel upravit základní informace o videu, jako jsou titulky, jazyk, popřípadě upravit nebo přidat uživatelské poznámky.

Parametr `hover` umožní uživateli vidět detail filmu při najetí na komponentu kurzorem myši.



Obr. 17 Možnosti zobrazení MovieCard

Action panel neboli ovládací panel uživateli nabízí čtyři možnosti.

Jednou z nich je ikona srdce, která film přidá do oblíbených. Pokud se již film v oblíbených nachází, ikona je zvýrazněna červenou barvou. Po kliknutí na ikonu je film z oblíbených odebrán.

Další ikona „grafu“ přidá film mezi sdílené. Sdílené filmy mohou ostatní uživatelé vidět na stránce Sdílené filmy. Pokud uživatel nechce, aby byl film již na dále sdílený, tak na ikonu „grafu“ která je obarvena na modrou barvu a značí tak sdílený film, klikne a film je poté odebrán ze sdílených.

Oblíbený

Přidání do oblíbených a sdílení filmu využívají společnou API url, která je připravena pro editaci filmu. Jedná se o PATCH API

```
api/list/movie/' + movieId
```

Parametr movieId určí, jaký film v databázi má být upraven. Toto API vyžaduje zaslání body ve formátu JSON, který musí obsahovat pouze atributy, které mají být u filmu změněny. Tedy pro přidání filmu do by body příkazu vypadalo následovně:

```
{"favorite": true}
```

Jak už samotné názvy ostatních parametrů napovídají, jedná se o pomocné parametry pro parametr action.

Odstranění filmu

RemoveMovieFromList odstraní film z uživatelské prostředí (FE). Uživateli film zmizí z jeho seznamu filmů, který vidí. Samotné tlačítko „koše“ pak zašle DELETE požadavek smazání filmu na BE

```
api/list/movie/' + movieId
```

Úprava filmu

Při kliknutí na ikonu tužky má komponenta za úkol zobrazit komponentu [EditMovieModal](#), která umožňuje základní editaci filmu.

Změna filmu

Při kliknutí na ikonu „lupy“ zobrazí komponenta MovieCard komponentu [ChangeMovieModal](#), která umožní změnu vybraného filmu za jiný.

4.2.5 Komponenta MyCarousel

Pro zobrazení filmů na Přehledu bylo zapotřebí implementovat carousel, ve kterém je uživatel schopný listovat a procházet filmy. Jak již bylo zmíněno na Přehledu se zobrazují filmy „Naposledy přidané“, „Nejlepší filmy“ a „Nejlepší sdílené filmy“. Každá tato sekce obsahuje 20 filmů, které uživatel může pomocí carouselu procházet.

Komponenta MyCarousel přijímá povinný parametr „movies“, který je typu pole objektů (stejně jako komponenta [MovieCard](#)). Stejně jako [MovieCard](#) lze do komponenty zaslat nepovinné parametry „hover“, „detailAvailable“ a „action“. Tyto parametry jsou poté předány do komponent [MovieCard](#), které se v MyCarousel zobrazují.

```
<MyCarousel movies={lastAdded} hover detailAvailable action={true} className='mt-10' />
```

Pro funkcionalitu carouselu byl využit balíček 'swiper/react'. 'Swiper/react' v NextJS je nástroj pro implementaci responzivních sliderů ve webových aplikacích. Jeho využití přináší několik klíčových výhod:

- Responzivní Slider
 - 'Swiper/react' umožňuje jednoduché vytváření responzivních sliderů, které se automaticky přizpůsobují různým zařízením a velikostem obrazovek.
- Moderní Efekty a Hardwarová Akcelerace
 - Poskytuje širokou škálu moderních efektů pro přechody mezi snímky a využívá hardwarovou akceleraci pro plynulý uživatelský zážitek.
- Jednoduchá Integrace s NextJS
 - Snadná integrace s NextJS umožňuje rychlé a efektivní vytváření sliderů bez zdlouhavé konfigurace.
- Široká Podpora a Komunita
 - Balíček získal popularitu díky široké podpoře a aktivní komunitě vývojářů. To znamená dostupnost návodů, příkladů a podpory.
- Přizpůsobení Designu
 - Poskytuje možnosti rozsáhlého přizpůsobení designu slideru, což umožňuje vytvářet vizuálně atraktivní uživatelská rozhraní.
- Efektivní Řízení Navigace
 - Umožňuje snadné řízení navigace slideru pomocí různých možností, jako jsou tlačítka, ukazatele a další.

Použití 'swiper/react' v NextJS přináší jednoduchý a efektivní způsob, jak implementovat moderní a responzivní slidery do webových projektů. [20]

Tento balíček umožnil snadnou implementaci a nastavení požadovaného vzhledu.



Obr. 18 Přehled – carousel na stránce

Při využití parametrů „action“ a „hover“ se při najetí kurzorem myši na kartu filmu zobrazí detail a ovládací panel. Tyto parametry jsou využity pro zobrazení „Naposledy přidané“ a „Nejlepší filmy“. Pro sekci „Nejlepší sdílené filmy“ není použit parametr „action“. Cizí sdílené filmy nelze upravovat jinému uživateli.

V každém carouselu je zobrazeno celkem 20 filmů, které lze procházet „ve smyčce“. To znamená, že při dosažení konce seznamu se začne procházení opět od začátku."

```

<Swiper
  effect={'coverflow'}
  grabCursor={true}
  centeredSlides={true}
  slidesPerView={5}
  spaceBetween={20}
  loop={true}
  coverflowEffect={{
    rotate: 20,
    stretch: 0,
    depth: 200,
    modifier: 1,
    slideShadows: true,
  }}
  pagination={{
    dynamicBullets: true,
  }}
  modules={[EffectCoverflow, Pagination, Autoplay]}

```

```

className={mySwiper ${props.className}}
>
{movies.map((movie, index) => {
  return <SwiperSlide className='my-5'>
    <MovieCard movie={movie} key={index} hover={hover} detailAvailable={detailAvailable} action={action} />
  </SwiperSlide>
  )}}
</Swiper>

```

4.2.6 Komponenta *ProgressBar*

Pro dosažení elegantního a interaktivního ukazatele postupu byl v projektu využit balíček '@ramonak/react-progress-bar'. Tento ukazatel postupu, známý též jako *ProgressBar*, se vyznačuje jednoduchou integrací a flexibilitou ve vizuálním designu.

```

<ProgressBar completed={importProgress} maxCompleted={100} bgColor='#60a5fa' className='md:w-full
px-4 pb-4 lg:w-2/3 m-auto' />

```

ProgressBar přijímá klíčové parametry, které umožňují jeho konfiguraci podle konkrétních potřeb projektu.

- **completed:** Udává aktuální postup v procentech.
- **maxCompleted:** Stanovuje maximální hodnotu postupu, v tomto případě 100 procent.
- **bgColor:** Určuje barvu pozadí ukazatele postupu.
- **className:** Poskytuje možnost přizpůsobit vzhled *ProgressBaru* pomocí CSS tříd.

4.2.7 Komponenta *Toast*

Pro implementaci elegantních notificačních zpráv byl v projektu využit balíček 'react-toastify'. Tato komponenta poskytuje snadné a efektivní zobrazování notifikací, což zvyšuje uživatelský komfort a informovanost o stavu aplikace.

Ukázka použití:

Komponenta *Toast* využívá několik metod balíčku 'react-toastify' pro různé typy notifikací:

```

toast.promise(promise, {pending: t('movies_uploading'), success: t('movies_added'), error: t('error')})

```

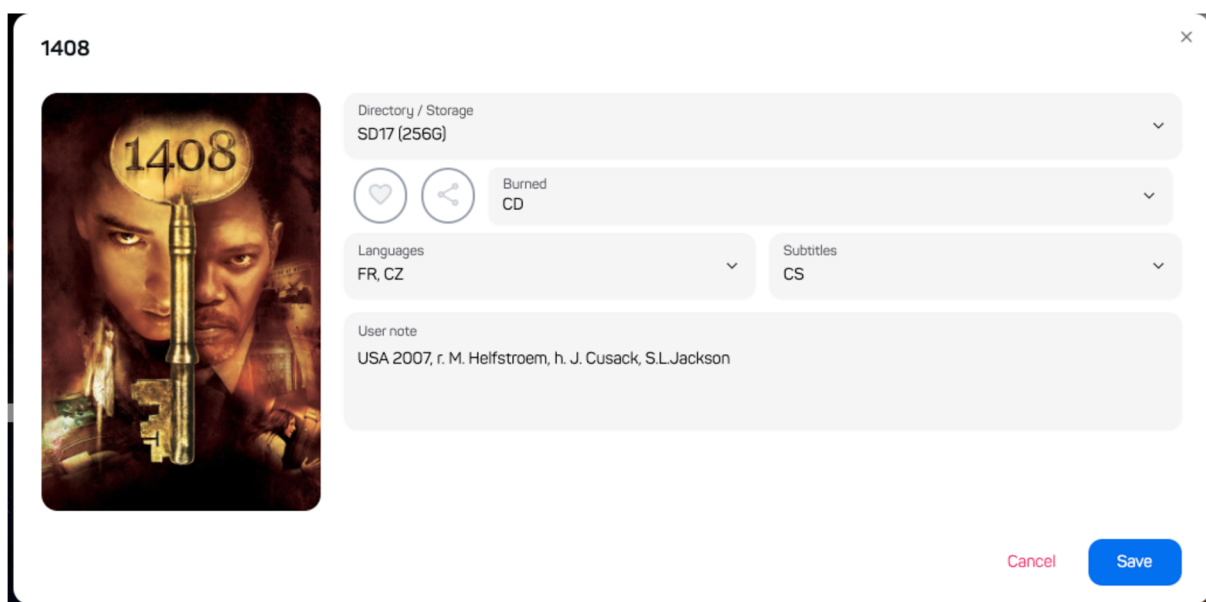


```
toast.loading(t('loading'))
```

```
toast.update(id, { render: t('rendered_movies'), type: "success", isLoading: false, autoClose: 2000 });
```

4.2.8 Komponenta *EditMovieModal*

Tato komponenta má využití jak na stránce Přehled, tak i na stránce Filmy. Je aktivována po kliknutí na ikonu tužky, což uživateli umožňuje editovat důležitá data o filmu. Po zobrazení modálního okna má uživatel možnost upravit klíčová data o filmu, mezi něž patří volba složky, označení filmu jako oblíbeného, sdíleného, uloženého na CD/DVD, volba jazyků a titulků a také přidání nebo editace uživatelské poznámky.



Obr. 19 Modální okno pro úpravu filmu

4.2.9 Komponenta *ChangeMovieModal*

Komponenta *ChangeMovieModal* představuje klíčový prvek uživatelského rozhraní, který umožňuje flexibilní a intuitivní změnu filmu v uživatelské kolekci. Aktivuje se kliknutím na ikonu "lupy" v rámci komponenty [MovieCard](#). Jejím hlavním cílem je poskytnout uživateli nástroj pro rychlou a snadnou aktualizaci filmu.

Po stisknutí "lupy" se zobrazí modální okno s předvyplněným názvem filmu, a automaticky vyhledává v The Movie Database (TMDB) nové výsledky. Uživatel má možnost vybrat nový film ze seznamu výsledků. Po výběru je nový film automaticky

uložen do databáze s aktualizovanými informacemi, včetně obrázku, názvu, data vydání, žánrů a popisu filmu.

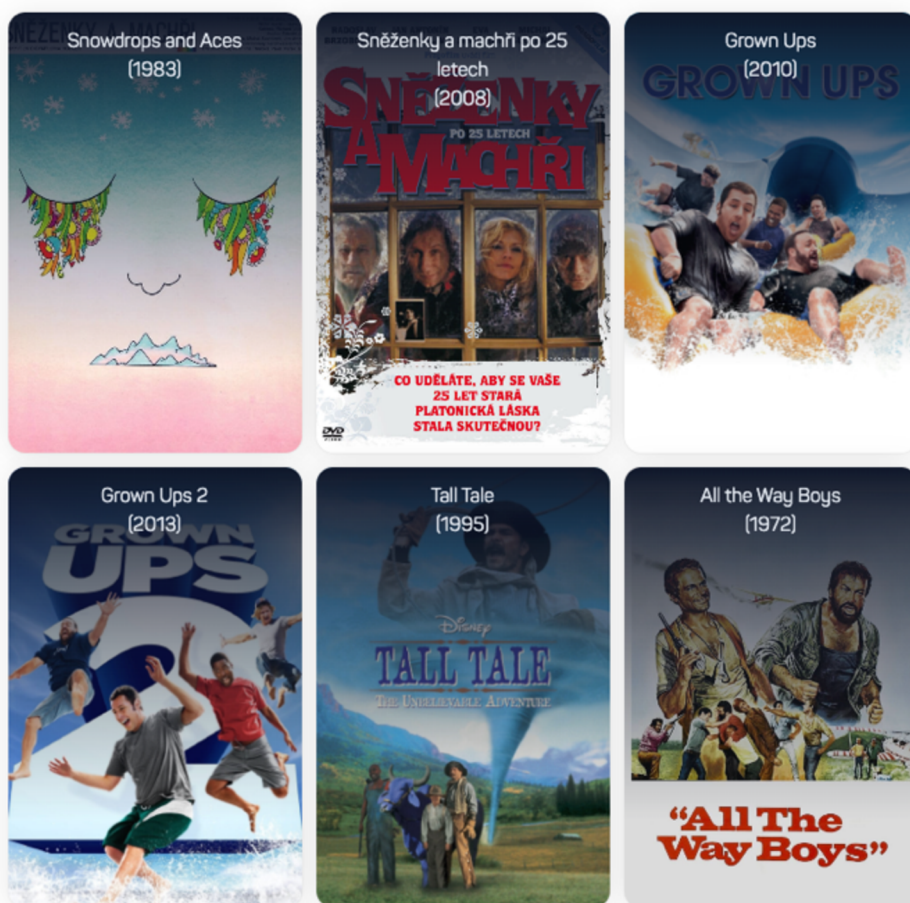
Změny zahrnují aktualizaci důležitých aspektů filmu, přičemž ostatní nastavení zůstávají nezměněna podle předchozího stavu. To zahrnuje oblíbenost filmu, zařazení do adresáře a další klíčové atributy. Tímto způsobem komponenta ChangeMovieModal umožňuje uživateli efektivně spravovat a aktualizovat svou filmovou kolekci při zachování personalizovaných nastavení. Je implementována na stránkách importu filmů, seznamu filmů a přehledu filmů.

Ava (2020)

x

Ava: Bez soucitu (2020)

Hledat nový film
machři



Obr. 20 Modální okno pro změnu filmu

4.2.10 **Komponenta ImportedMovieCard**

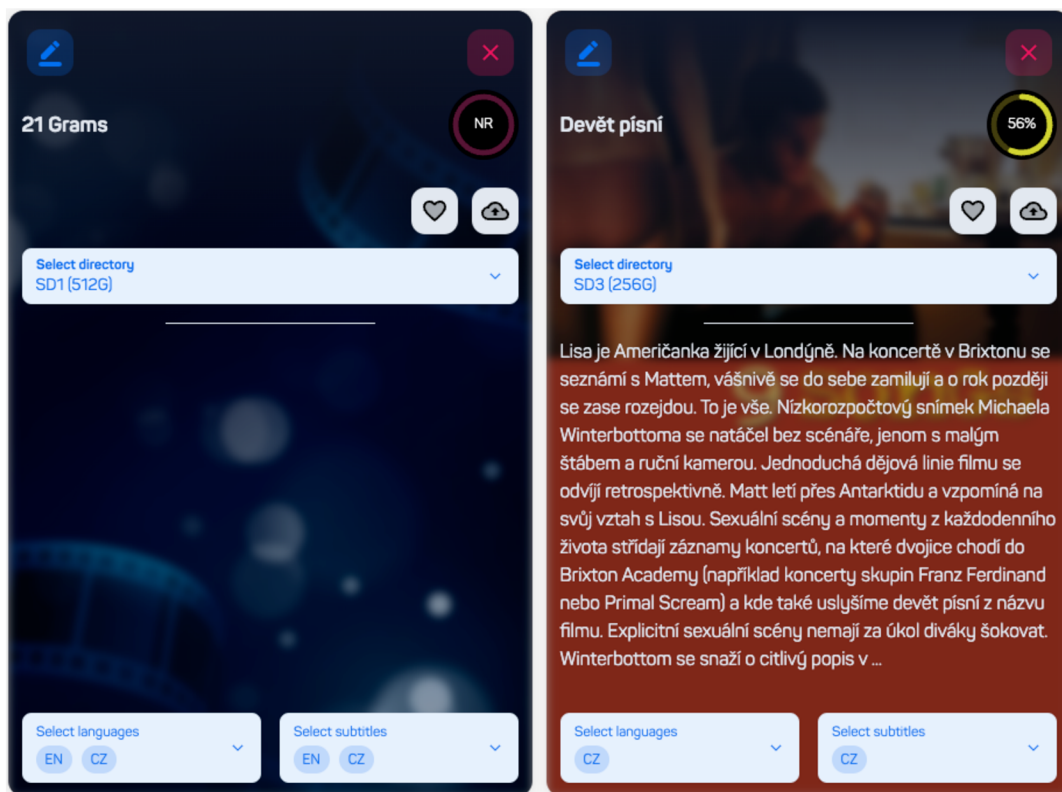
Pro implementaci ImportedMovieCard byl zvolen grafický balíček MUI. Tato komponenta je využita například při nahrání filmů na stránce „[Nahrát](#)“. Vlastnosti komponenty jsou zobrazení pozadí obrázku (pokud existuje), zobrazení názvu, zobrazení složky/uložiště, krátkého popisu, hodnocení, jazyky a titulky filmu. Dále je zobrazuje možnost smazání filmu, změnu filmu za jiný přidání do oblíbených a sdílených. Pokud film nebyl rozpoznán, jsou uvedena základní data, která jsou známa a uživatel film musí ručně dohledat.

Vlastnosti

- **Pozadí obrázku:** Zobrazuje pozadí obrázku filmu, pokud existuje.
- **Název filmu:** Zobrazuje název filmu.
- **Složka/Uložiště:** Informuje o umístění filmu v rámci složky nebo úložiště.
- **Krátký popis:** Poskytuje krátký popis obsahu filmu.
- **Hodnocení:** Zobrazuje hodnocení filmu.
- **Jazyky a titulky:** Informuje o dostupných jazycích a titulcích k filmu.
- **Možnosti operací:** Umožňuje uživateli provádět různé operace s filmem, včetně smazání, změny, přidání do oblíbených a sdílených.

Interaktivní funkce

- **Dohledání filmu:** Kliknutím na ikonu "lupy" má uživatel možnost dohledat film. Při této funkci je využívána a zobrazena komponenta [EditMovieModal](#).
- **Odstranění filmu:** Uživatel může smazat film po kliknutí na ikonu "křížku" a potvrzení operace.
- **Výběr jazyků a titulků:** Uživatel může vybírat z jazyků a titulků dostupných v jeho profilu.



Obr. 21 Ukázka komponenty ImportedMovieCard

4.2.11 Komponenta Layout

Komponenta Layout je jednou z nejpoužívanějších komponent v celé aplikaci. Uživatel se s ní může setkat na všech stránkách (kromě přihlášení/registrace). Tato komponenta byla implementována za účelem zachování vzhledu aplikace na všech stránkách, tzn. zobrazuje [Navbar](#) a zbytek obsahu, který je komponentě „zaslán“ jako children. Tato komponenta poté obaluje téměř všechny stránky a je využívána jako „wrapper“.

4.2.11.1 Ukázka implementace

Rozvržení je koncipováno tak, aby bylo uživatelsky přívětivé, s možností skrývání a zobrazování bočního menu na mobilních zařízeních. Díky responzivnímu chování se zajišťuje optimální zobrazení na různých obrazovkách.

Plynulý přechod a animace přidávají dojem sofistikovanosti a zdokonalují uživatelskou zkušenost. Zároveň je kód strukturován tak, aby byl čitelný a snadno

udržovatelný, což usnadňuje další vývoj a rozšiřování aplikace. Celkově je patrný důraz na moderní přístup k tvorbě uživatelských rozhraní.

```
const Layout = ({ children, ...props }) => {
const [showSidebar, setShowSidebar] = useState(false);
return (
  <div className="min-h-screen flex dark:bg-gray-900">
    <MenuBarMobile setter={setShowSidebar} />
    <Sidebar show={showSidebar} setter={setShowSidebar} />
    <div className="flex flex-col flex-grow w-screen pt-[60px] transition-[padding-top,padding-left] ease-in-out duration-500 md:w-full md:pl-[250px] md:pt-0 min-h-screen">
      {children}
    </div>
  </div>
)
}
```

4.2.11.2 Ukázka použití

Následující kód představuje ukázkou použití komponenty **Layout** pro stránku Dashboard.

```
<Layout>
  <div className='flex flex-col min-h-screen bg-slate-100 dark:bg-slate-900'>
    {isLoading ?
      <div className='text-center pt-10'>
        <h1 className="heading-1">{t('last_movies')}</h1>
        {lastAdded ?
          <MyCarousel movies={lastAdded} directoriesData={directories} fetchData={fetchData}
languagesData={languages} subtitlesData={subtitles} changeMovie={changeMovie} hover detailAvailable
action={true} className='mt-10' /> : <h1 className="heading-2 mb-2">{t('movies_not_found')}
</h1>}
        </div> : <DashboardSkeleton />
      }
    ...
  </div>
</Layout>
```

4.2.12 Komponenta Pagination

Komponenta Pagination je implementována pomocí již existující komponenty z balíčku NextUI, která nese též název Pagination. Tato komponenta přijímá několik parametrů pro konfiguraci:

- **total:** Udává maximální počet stránek nebo celkový počet položek, které lze rozdělit mezi stránky.
- **page:** Určuje aktuální stránku, na které se uživatel nachází nebo kterou vybral.
- **onChange:** Definuje funkci, která se volá při kliknutí na jinou stránku. Tato funkce má za úkol aktualizovat hodnotu aktuální stránky.
- **variant:** Nastavuje vzhled a velikost komponenty Pagination.
- **showControls:** Indikuje, zda se mají zobrazovat ovládací tlačítka pro přechod mezi stránkami.
- **showShadow:** Určuje, zda se má zobrazovat stín komponenty.

```
const handleChange = (value) => {
  setCurrentPage(value);
  props.setPage(value)
};
<Pagination total={totalPages} page={currentPage} onChange={(value) => handleChange(value)}
variant='bordered' size='lg' showControls showShadow />
```

V inicializaci komponenty je parametr page nastaven na výchozí hodnotou 1. Kdykoliv uživatel přejde na jinou stránku, volá se funkce handleChange, která aktualizuje stav aktuální stránky a předává tuto informaci rodičovské komponentě prostřednictvím props.setPage(value).

4.2.13 **Komponenta MovieDetail**

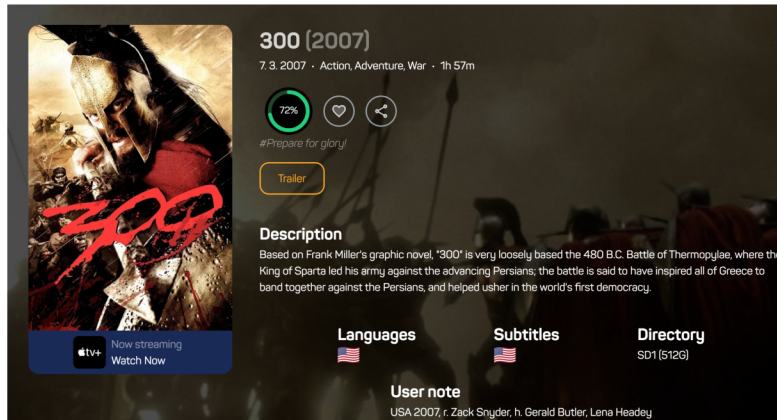
Komponenta MovieDetail přijímá jako parametr objekt filmu (movie), který obsahuje klíčová data, jako jsou název filmu, jazyky, titulky a další. Navíc, v případě, že některá data chybí, jsou tato doplňková data stahována z TMDB API. Mezi tato doplňková data patří seznam herců, obrázek pozadí filmu, dostupnost na online platformách (např. Netflix, HBO), doporučené filmy, ukázka filmu, obrázek v české a anglické verzi a další.

Komponenta se vynikajícím způsobem stará o přehledné zobrazení všech důležitých informací spojených s daným filmem ve spojení s daty z backendu a TMDB. Uživateli poskytuje bohatý pohled na film a jeho kontext.

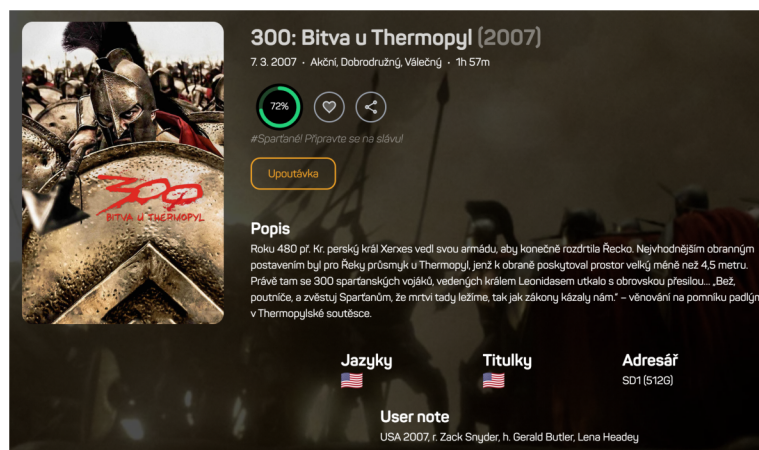
Tato komponenta elegantně prezentuje titulní obrázek filmu spolu s filmovým pozadím, které je umístěno za ostatními informacemi. Kromě toho zobrazuje název filmu, datum vydání, žánry, délku, popis, jazyky, titulky, hodnocení filmu a složku, kde je film uložen. Velmi užitečným prvkem je také zobrazení online služby, která daný film nabízí ve vybraném jazyce stránky (čeština, angličtina), umožňující uživateli jednoduše přehrát film online, pokud má předplatné dané služby. Další užitečnou funkcionalitou pro uživatele je zobrazení herců, kteří ve filmu hrají, a doporučení podobných filmů.

Významnou vlastností komponenty je také schopnost dynamického přizpůsobení dat podle zvoleného jazyka uživatele. To znamená, že pokud uživatel preferuje češtinu, zobrazí se mu informace v češtině, a naopak.

Kromě toho uživatel získává možnost přidat film do oblíbených, sdílených nebo přehrát ukázkou filmu pomocí tlačítka "Trailer". Tímto způsobem se komponenta MovieDetail stává klíčovým prvkem pro uživatelský zážitek při prohlížení detailů filmu.



Obr. 22 Detail filmu (anglicky)

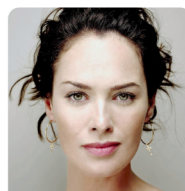


Obr. 23 Detail filmu (česky)

Nejlépe účtované obsazení



Gerard Butler
King Leonidas

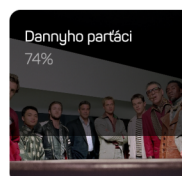
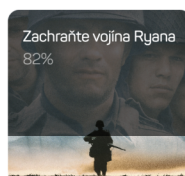
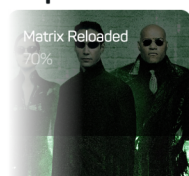


Lena Headey
Gorgo



Dominic West
Theron

Doporučení



Obr. 24 Detail filmu – Herci a doporučení

4.2.14 Stránka Detail filmu

Stránka Detail filmu má jednoduchý úkol, a to získání dat ze serveru a předání těchto informací do komponenty [MovieDetail](#). Komponenta [MovieDetail](#) poté zobrazí uživateli všechna důležitá data o daném filmu. Klíčovou funkcí této stránky je propojení s komponentou [MovieDetail](#), která se stará o prezentaci detailů filmu uživateli. Tímto zajišťuje uživatelsky přívětivé a esteticky zpracované zobrazení informací o filmech na stránce Detail filmu.

4.2.14.1 **Komponenta MovieModal**

Komponenta MovieModal je vyvolána při kliknutí na tlačítko „Přidat film“ na stránce [Filmy](#). Přidání filmu je rozděleno do 2 sekcí. První sekcí je výběr filmu a načtení důležitých informací z TMDB jako jsou například popis a hodnocení filmu.

Poté co uživatel vybere film, který chce přidat, je mu zobrazena další sekce, kde je nutné vyplnit doplňující informace o filmu, který vlastní. Mezi tyto informace patří místo uložení (Adresář / Úložiště), dabing filmu, titulky a uživatelská poznámka. Adresář a dabing filmu je povinná položka a nelze je přeskočit.

Po vyplnění informací může uživatel kliknout na tlačítko „Přidat film“ a film se uloží uživateli na profil.

4.2.15 **Stránka Filmy**

Stránka s filmy nabízí uživatelům přehled všech filmů v jejich sbírce. Maximálně je zobrazeno 20 filmů na stránce, a v případě, že uživatel má více filmů, může jednoduše přejít na další nebo předchozí stránku pomocí komponenty `Pagination`.



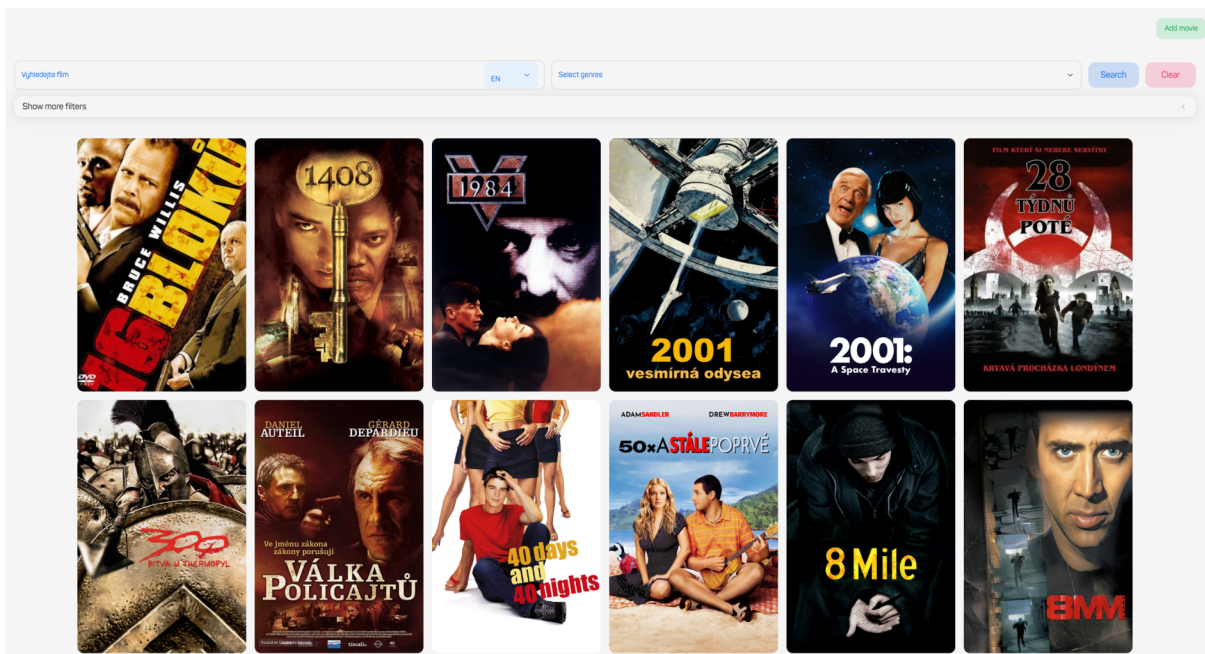
Obr. 25 Stránkování

Na této stránce se nenachází žádný carousel; každý film je prezentován ve formě MovieCard. Na úvodní části stránky se nachází tlačítko "Přidat film", panel s filtry, tlačítko "Vyhledat" a "Vyčistit". Kliknutím na "Přidat film" se zobrazí komponenta MovieModal, která umožňuje uživateli přidat nový film do své sbírky.

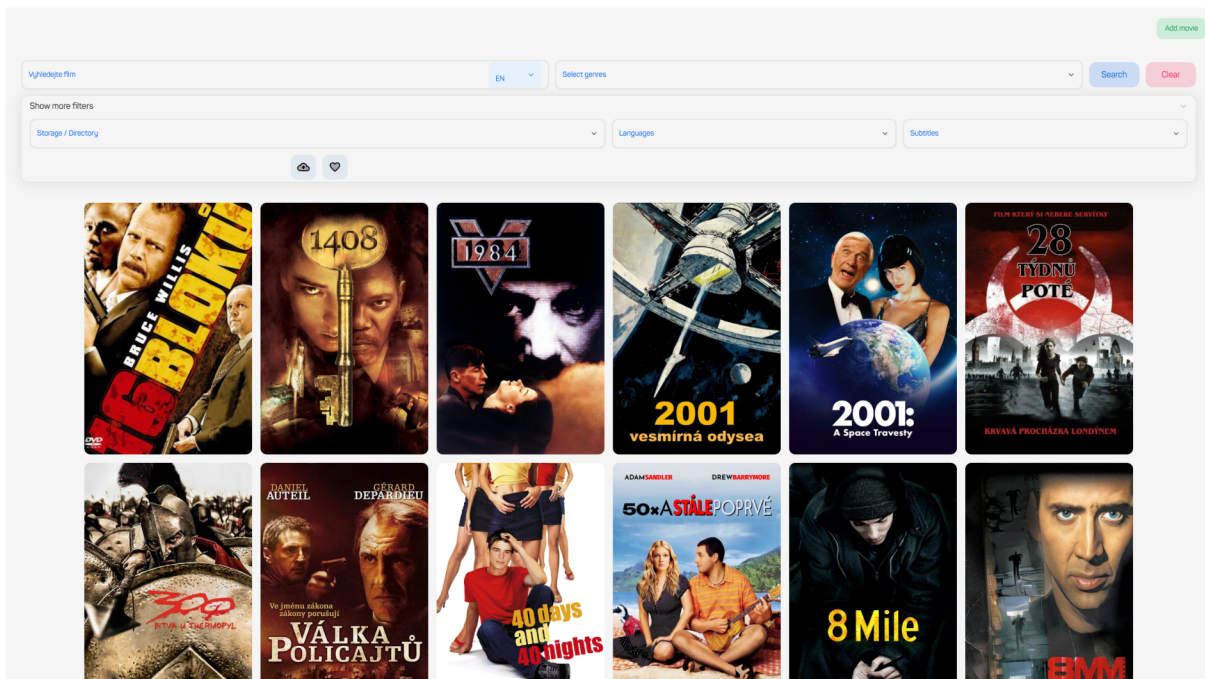
Základní filtry, které jsou přítomny při načítání stránky, zahrnují "Vyhledejte film" (hledání podle názvu) a "Vyberte žánry" (filtrování podle žánrů). Uživatel má možnost zobrazit více filtrů, kliknutím na tlačítko "Zobrazit více filtrů". Mezi tyto filtry patří "Úložiště / Složka" (filtr dle umístění), "Jazyky" (podle jazyka dabingu) a

"Titulky" (podle dostupných titulků). Dále jsou k dispozici ikony "srdce" (oblíbené filmy) a "mrak se šipkou" (uživatelsky sdílené filmy).

Při použití tlačítka "Vyhledat" jsou aplikovány všechny nastavené filtry. Pro rychlé obnovení filtrů do výchozího stavu slouží tlačítko "Vyčistit".



Obr. 26 Stránka s filmy – základní filtry



Obr. 27 Stránka s filmy – rozšířené filtry

4.2.16 Stránka Sdílené filmy

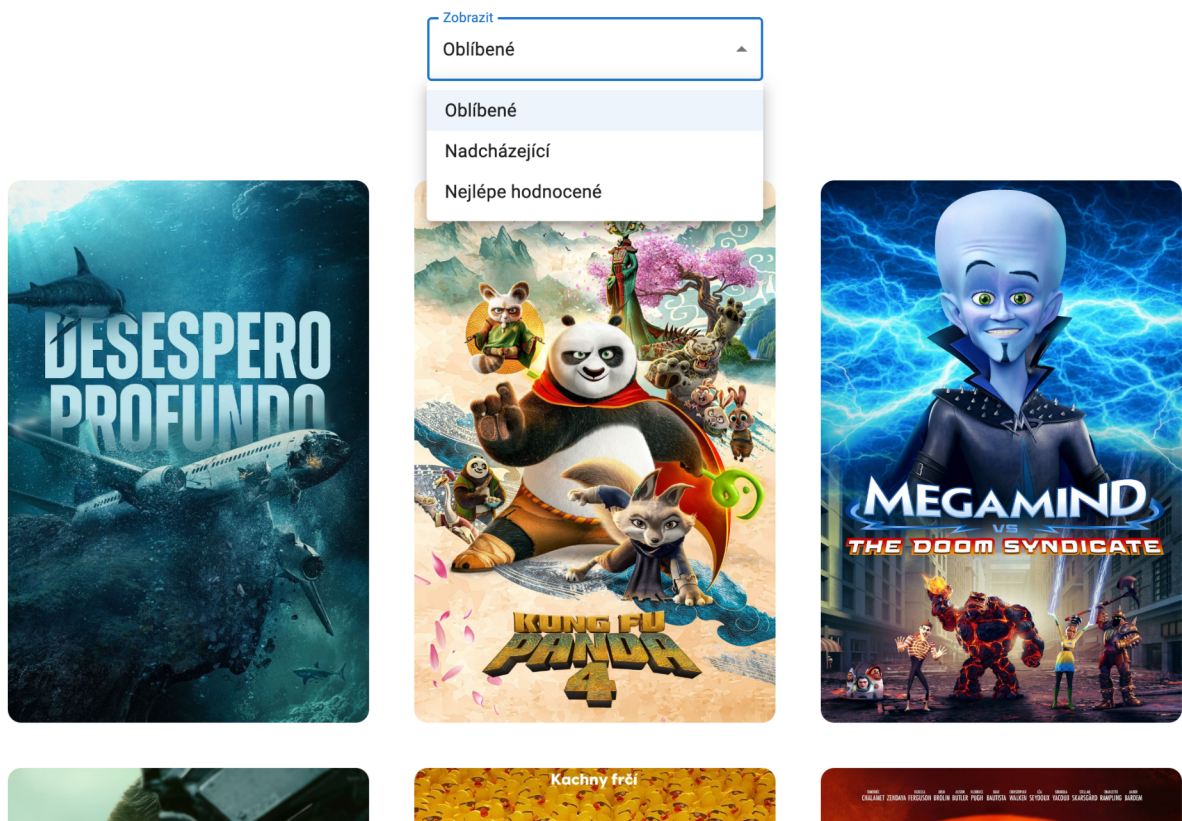
Sekce "Sdílené filmy" poskytuje uživateli komplexní náhled na všechny filmy, které jsou sdíleny ostatními uživateli v rámci platformy. Uživatelská funkcionality zahrnuje možnost filtrování obsahu na základě konkrétního uživatele a vyhledávání filmů podle jejich názvu. Je významné poznamenat, že tato funkcionality je podmíněna výběrem specifického uživatele, což je omezení nastavené na straně backendu (BE). Frontendová část (FE) aplikace byla systematicky přizpůsobena tomuto omezení, což zajišťuje konzistentní uživatelský zážitek.

Z hlediska vizuálního provedení je stránka "Sdílené filmy" koncipována s ohledem na uživatelskou přívětivost, vycházející z osvědčeného designu stránky "[Filmy](#)". Zahrnuje karty filmů, mechanismus stránkování pro pohodlný přehled obsahu a filtry pro efektivní třídění dat. Zásadní rozdíl spočívá v tom, že na kartách filmů "Sdílené filmy" není dostupný ovládací panel (action). Tímto opatřením se zabezpečuje, že uživatel, který není vlastníkem daných filmů, nemá oprávnění k jejich editaci, což dodává systému potřebnou bezpečnostní vrstvu.

4.2.17 Stránka Prohlížeč

Stránka "Přehled filmů" v rámci mé aplikace poskytuje uživatelům systematický přehled 20 filmů. Uživatel má možnost využít jediný filtr na stránce, který umožňuje zobrazení aktuálně populárních (Popular), připravovaných do kin (Upcoming) a nejlépe hodnocených (Top rated) filmů. Tato uživatelsky přívětivá stránka využívá komponentu MovieCard, kde je aktivní pouze nepovinný parametr hover.

Po kliknutí na konkrétní film je uživatel přesměrován do nové karty prohlížeče, která otevírá stránku TMDB. Tímto propojením získává uživatel možnost získat podrobnější informace o vybraném filmu přímo na oficiálním zdroji dat, což poskytuje komplexní a aktuální informace.



Obr. 28 Stránka Prohlížeč

4.2.18 Stránka Nahrát

Stránka pro nahrání souborů prochází čtyřmi stavy, poskytujícími uživateli přehledný a interaktivní zážitek.

1. Čekání na soubor

Při prvním navštívení stránky je uživatel vítán s výzvou k nahrání souboru. Má možnost buď vybrat soubor z vlastního počítače kliknutím do pole nebo použít funkci Drag&Drop pro snadné přetáhnutí souboru do příslušné oblasti. Funkce Drag&Drop je implementována pomocí balíčku "react-drag-drop-files".

2. Odesílání dat BE

Po úspěšném výběru souboru probíhá jeho odeslání na server (BE), kde dochází ke zpracování. U .json souborů probíhá částečné zpracování na straně webové aplikace, což umožňuje zobrazení "progressbaru" indikující průběh

zpracování. V případě .csv souborů je celý soubor odeslán na API bez zobrazení [progress baru](#).

API pro odeslání JSON souboru je:

```
api/load/import/json
```

API pro odeslání CSV souboru je:

```
api/load/import/csv
```

3. Obdržení a zobrazení výsledků

Po úspěšném zpracování serverem jsou uživatelé prezentováni výsledky. Má možnost překontrolovat přiřazení filmů podle názvu souboru, upravovat informace o filmech, přidávat je do oblíbených nebo sdílených a měnit další parametry. Tuto interakci umožňuje komponenta [MovieModal](#).

4. Nahrání filmů do databáze

Stiskem tlačítka "Nahrajte filmy" jsou filmy odeslány ke zpracování na straně serveru a postupně se začínají objevovat na stránce "[Filmy](#)". Uživatel je informován toast notifikacemi o aktuálním průběhu celého procesu.

Celým průběhem uživatele provází [Toast](#) notifikace, poskytující aktuální informace. Například "Prosím čekejte..." během nahrávání, "Filmy ze souboru byly vykresleny" po úspěšném zpracování a "Na váš účet byly nahrány filmy" po dokončení nahrávání. Uživatel je tak v každém okamžiku informován o stavu procesu nahrávání.

4.2.18.1 Soubor JSON

Pro generování souboru .json s filmy je nutné zdůraznit, že tento proces je možný pouze na zařízeních s operačním systémem MacOS. Je však třeba dodržet následující pravidla pro vytvoření validního souboru obsahujícího informace o filmech:

1. **Umístění filmů:** Filmy by měly být umístěny v jedné konkrétní složce, například složce s názvem "Filmy".

2. **Rozdělení filmů:** Ve složce "Filmy" by měly být filmy dále rozděleny do kategorií, například "Akční", "Komedie", atd.
3. **Názvy souborů:** Názvy souborů s filmy by měly být v angličtině nebo v češtině.
4. **Informace v názvech:** Za název souboru se uvádí jazyky dabingu a titulky pomocí hranatých závorek
Pokud film nemá titulky, hranaté závorky pro titulky zůstávají prázdné. Za titulky se dále uvádí rok vydání filmu v kulatých závorkách.

Příklad:

Animovaná pohádka "Frozen" s anglickým a českým dabingem, slovenskými a českými titulky a rokem vydání 2013:

Filmy → Animované → Frozen [CZ,EN] [CZ,SK] (2013).mkv

Akční film "Venom" s českým dabingem bez titulků a rokem vydání 2018:

Filmy → Akční → Venom [CZ] [] (2018).mkv

Pro vygenerování .json souboru je potřeba na zařízení s MacOS otevřít terminál a provést následující příkazy:

1. `cd „cesta ke složce filmy“` (např. `cd ./Desktop/Filmy/`)
2. `tree -J > „název_souboru“.json` (např. `tree -J > mojeFilmy.json`)

Tímto postupem vznikne soubor "mojeFilmy.json", který je následně možné importovat do webové aplikace. Jména složek ("Animované" a "Akční") slouží jako Úložiště/Složka daného filmu a umožňují filtrování podle kategorií (viz. například sekce [Filmy](#)).

Implementace

Při respektování definovaných pravidel probíhá částečný import souboru již na straně klienta (FE). Nahraný soubor je rozdělen na několik API volání a postupně odesílá požadavky na server. Rozdělení souboru probíhá podle předem

definovaných složek, jako například Akční a Animované, jak bylo uvedeno výše. Každá tato složka je pak odeslána jako JSON požadavek na BE ke zpracování. Implementace tohoto dělení přináší dvojí výhodu. Prvním klíčovým důvodem je optimalizace velikosti odesílaného těla na stranu BE, což vede k urychlení procesu importu filmů. Druhým důležitým aspektem je možnost uživatele sledovat průběh zpracování požadavku a zobrazovat mu aktuální stav pomocí [progress baru](#).

Pro zpracování souboru ve formátu JSON je využívána podmínka, která kontroluje typ nahraného souboru. V případě, že se jedná o formát "application/json", soubor je postupně zpracováván pomocí FileReader a následně odeslán na BE ve formě JSON požadavku. Před samotným odesláním jsou provedeny další úpravy na strukturu dat, aby byla zajištěna správná funkcionality.

```
if (fileType === "application/json") {
  const reader = new FileReader()
  reader.readAsText(changedFile, "UTF-8")
  reader.onload = changedFile => {
    json = JSON.parse(changedFile.target.result)
    const regex = /^\[.*\]$/;
    json[0].contents.forEach(folder => {
      if (folder.name.match(regex)) {
        return;
      } else {
        nJsonArr.push(getJsonTemplate(folder))
      }
    })
    console.log(nJsonArr)
    postJson(nJsonArr)
  }
  setFile(changedFile)
}
```

Postupné odesílání požadavků na BE probíhá pomocí for cyklu, kdy v každém kroku je odesílána část souboru pomocí funkce `uploadMovieJson()`. Komponenta toast je využita k informování uživatele o stavu nahrávání souboru. Při nahrání souboru je toast ve stavu „loading“. Po zpracování na straně BE a úspěšném obdržení všech výsledků je tento toast přepnut do stavu „success“ a uživateli je zobrazena hláška o úspěšném nahrání filmů na jeho účet. Tímto způsobem je zajištěna přehlednost a uživatelská přívětivost při importu filmů.

```

const id = toast.loading(t('loading'))

for (let i = 0; i < jsonArray.length; i++) {
  const data = await apiManager.uploadMovieJson(jsonArray[i])
  setProgressBar(jsonArray.length, i + 1)
  data.forEach(movie => {
    tempData.push(movie)
  });
}

toast.update(id, { render: t('rendered_movies'), type: "success", isLoading:
false, autoClose: 2000 });

```

4.2.18.2 Soubor CSV

Pro usnadnění procesu importu filmů do webové aplikace byla vytvořena šablona ve formátu CSV pomocí Google Sheets. Šablona je k dispozici zde: <https://1url.cz/Eumvr>

Uživatelům postačuje vyplnit tuto šablonu, minimálně všechny sloupce označené hvězdičkou (*), a následně exportovat/stáhnout soubor ve formátu CSV.

Příklad:

A	B	C	D	E	F	G	H
Název (original)	Název (česky)*	Rok vydání*	Poznámky	Na CD/DVD	Úložiště / Složka*	Jazyk*	Titulky
1408	Pokoj 1408	2007	USA 2007, r. M. Helfstroem, h. J. Cusack, S.L.Jackson	CD	SD17 (256G)	EN,CZ	CS
1984	1984	1984	UK 1984, r. Michael Radford, h. John Hurt, Richard Burton		SD4 (256G)	EN	CS,SK
10th and Wolf	Kofény mafie	2006	USA 2006, r. R. Moresco, h. J. Marsden, G. Ribisi, P. Perabo	CD	SD4 (256G)	FR,CZ	FR,CS

Obr. 29 Šablona Google tabulek

Pro export GoogleSheet do formátu .csv stačí kliknout na:

Soubor → Stáhnout → Hodnoty oddělené čárkou (.csv).

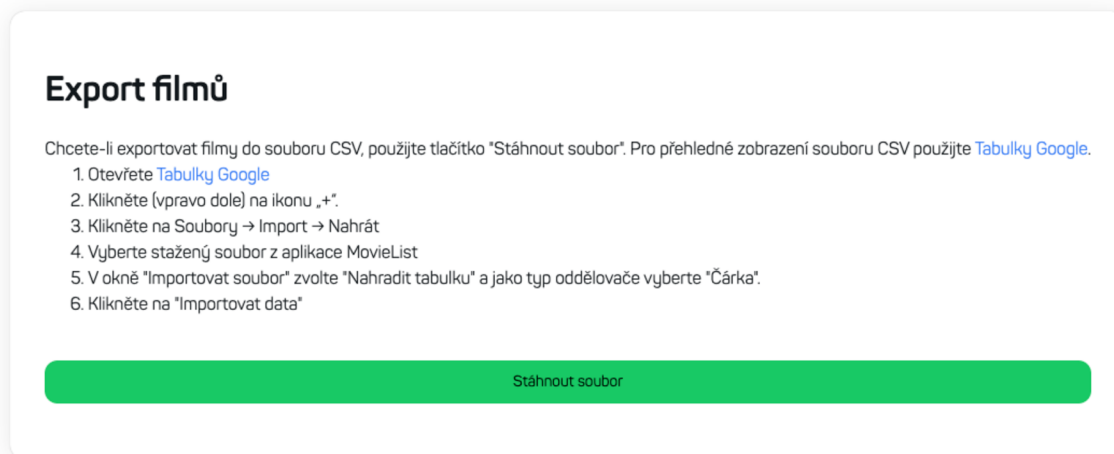
Soubor se stáhne a je připraven pro nahrání do webového rozhraní.

4.2.19 Stránka Stáhnout

Stránka Stáhnout (nebo Export) slouží ke stažení souboru ve formátu .csv, který obsahuje základní informace o uživatelských filmech. Tento soubor je poté možné importovat do GoogleSheets, které data přehledně zobrazí v tabulce ve formátu GoogleSheets šablony (viz [Soubor CSV](#)). Stránka je velice jednoduchá a obsahuje pouze jednu kartu s informacemi, jak s csv souborem pracovat a tlačítko pro stažení souboru.

Pro získání souboru z BE je využito GET API

```
api/load/export/csv
```



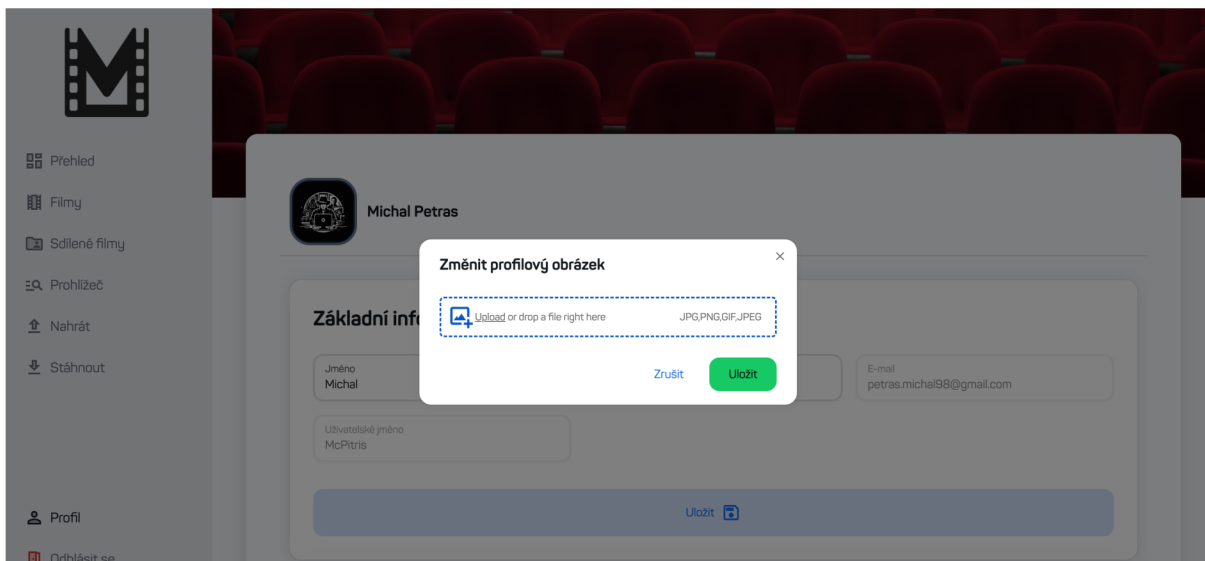
Obr. 30 Export filmů

4.2.20 Stránka Profil

Profilová stránka, klíčová pro osobní správu uživatelských informací a preferencí, byla detailně implementována s ohledem na uživatelskou přívětivost a funkcionalitu. Zde jsou podrobnosti jednotlivých sekcí:

4.2.20.1 Profilový obrázek

Na úvodní části stránky se nachází profilový obrázek uživatele, spolu s jménem a příjmením. Uživatel má možnost nahrát vlastní profilový obrázek, a to jednoduše kliknutím na aktuální obrázek. Tento proces využívá balíček [react-drag-drop-files](#), který byl úspěšně použit při implementaci [sekce importu filmů](#). Pro nahrání profilového obrázku jsou podporovány formáty JPG/JPEG, PNG a GIF.

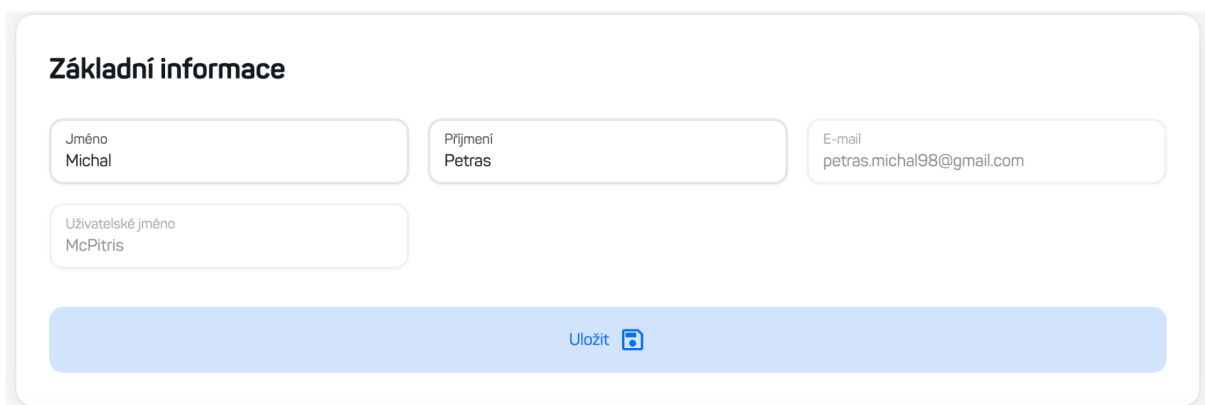


Obr. 31 Profil – Změna profilového obrázku

4.2.20.2 Základní informace

V první sekci s názvem "Základní informace" uživatel může upravovat své jméno a příjmení. Kromě toho jsou zde zobrazeny neměnné údaje, jako je email a uživatelské jméno. Po změně a kliknutí na tlačítko uložit dojde ke změně údajů. Pro změnu je využíváno API:

```
api/auth/edit-user
```



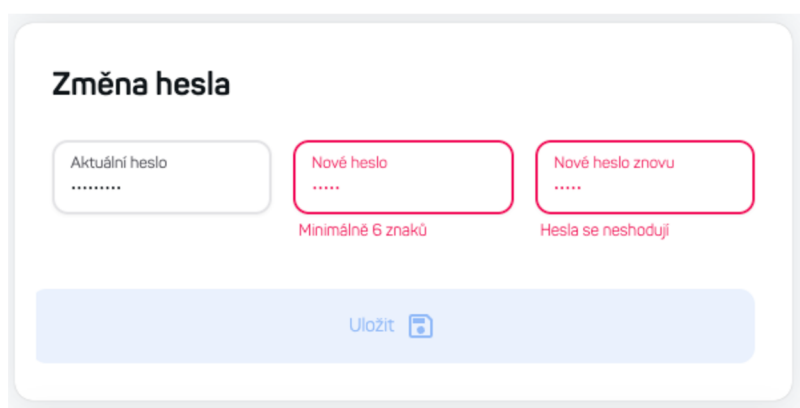
Obr. 32 Profil – základní informace

4.2.20.3 Změna hesla

Během procesu změny hesla se uživateli nabízí možnost vyplnit stávající heslo a následně zadat nové heslo spolu s jeho potvrzením. Frontend aplikace

aktivně provádí kontrolu nových hesel, zajišťuje, že splňují minimální požadavky na délku (v tomto případě minimálně 6 znaků), jak jsou definovány na straně backendu. V případě, že uživatel zadá krátké nebo neshodující se heslo, je o této situaci informován a tlačítko pro uložení změn je dočasně zakázáno.

Pro zvýšení bezpečnosti a transparentnosti uživatelských akcí jsou notifikace o úspěchu nebo neúspěchu prezentovány pomocí toast notifikací. Tento přístup garantuje, že uživatel je okamžitě informován o výsledku svých interakcí a může ihned reagovat na případné problémy nebo úspěchy.



Obr. 33 Profil – Změna hesla

Provedení změny hesla probíhá pomocí odeslání HTTP PUT požadavku na adresu:

```
api/auth/change-password
```

Tělo požadavku obsahuje informace o starém a novém hesle. Pokud uživatel zadá nesprávné staré heslo, je informován a změna hesla se neprovede. Úspěšná změna hesla je taktéž následně potvrzena prostřednictvím notifikace.



Obr. 34 Profil – notifikace změny hesla

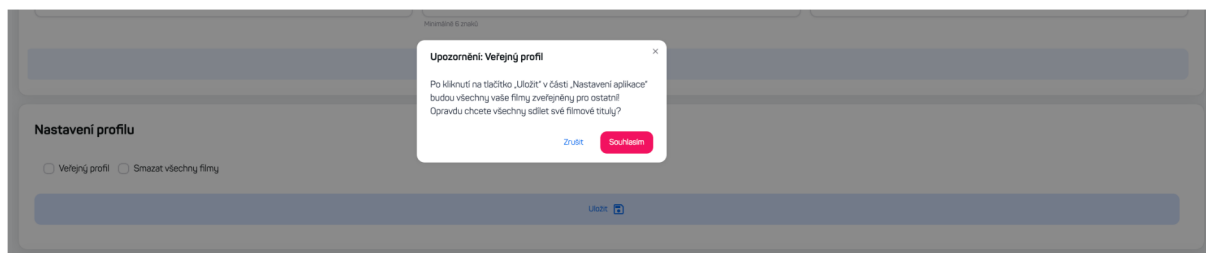
4.2.20.4 Nastavení profilu

V třetí sekci "Nastavení profilu" uživatel získává možnost přizpůsobit svůj uživatelský profil dle svých preferencí. Klíčovými funkcionalitami jsou možnost nastavení profilu na veřejný nebo soukromý a schopnost smazat všechny filmy ze své filmové kolekce.

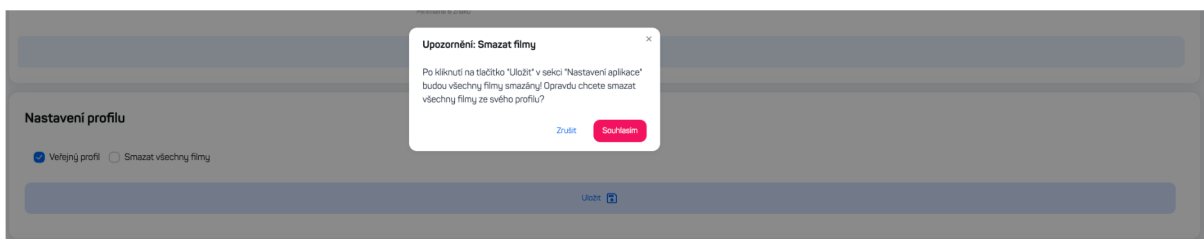
Nastavení profilu na veřejný znamená, že všechny uživatelské filmy budou sdíleny s ostatními uživateli webové aplikace, které jsou dostupné na stránce „[Sdílené filmy](#)“. Při aktivaci této volby je uživatel informován prostřednictvím vyskakovacího okna, které obsahuje detailní popis dopadu této změny. Aby uživatel mohl úspěšně nastavit svůj profil na veřejný, musí potvrdit své rozhodnutí v dialogovém okně a následně stisknout tlačítko "Uložit".

Podobný princip platí i pro možnost smazání všech filmů ze své filmové kolekce. Uživatel má možnost rozhodnout se pro tuto akci, ale předtím, než se změny provedou, je vyzván k potvrzení. Poté, co uživatel odsouhlasí, že chce nenávratně odstranit všechny filmy, klikne na tlačítko "Uložit" a všechny filmy jsou definitivně odstraněny z jeho uživatelského profilu.

Tato sekce poskytuje uživateli kontrolu nad úrovní sdílení jeho filmové sbírky a umožňuje mu provádět zásadní změny na jeho profilu v souladu s jeho preferencemi.



Obr. 35 Profil – nastavení veřejného profilu (upozornění)



Obr. 36 Profil – mazání všech filmů (upozornění)

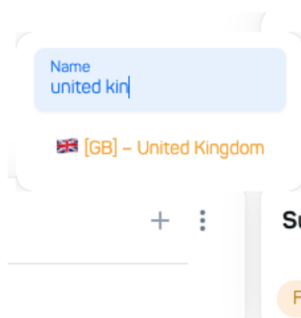
4.2.20.5 Nastavení filmů

Poslední čtvrtou sekci, označenou jako "Nastavení filmů", uživatel získává plnou kontrolu nad informacemi souvisejícími s jeho filmovou sbírkou. Klíčovým prvkem této sekce je možnost definovat, zda bude každý film označen informací o jeho uložení na DVD nebo CD. Po aktivaci této volby se u každého filmu zobrazí možnost vybrat konkrétní umístění – buď na CD, DVD nebo žádném z disků („Není“).

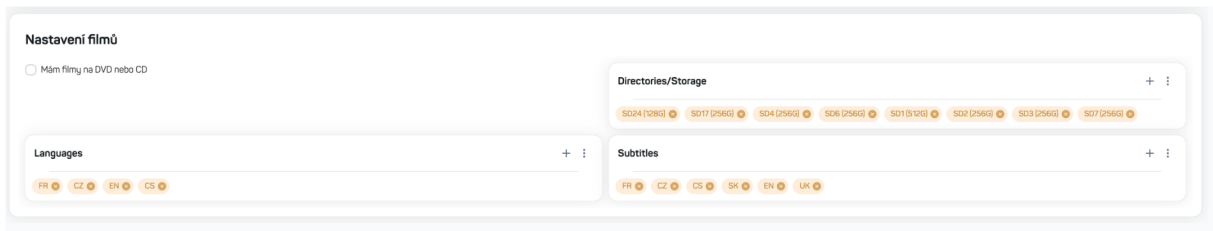
Další důležitou funkcí této sekce je správa jazyků, titulků a úložišť/složek, které jsou přiřazeny k jednotlivým filmům. Záznamy těchto prvků nelze odstranit, pokud jsou již přiřazeny ke konkrétnímu filmu. Uživatel je v takovém případě informován o této situaci.

Pro správu jazyků a titulků byl implementován balíček "country-codes-list". Tento balíček poskytuje uživateli seznam dostupných zemí a je schopen pomocí filtru najít konkrétní jazyk či titulky. Během přidávání filmu do sbírky jsou tyto číselníky využívány, což umožňuje uživateli vybírat z předem nastavených možností.

Celkově tato sekce poskytuje uživateli možnost podrobně nastavit a personalizovat informace týkající se jeho filmové sbírky. Zajišťuje tak precizní organizaci a lepší uživatelský zážitek při používání webové aplikace.



Obr. 37 Profil – ukázka balíčku country-codes-list



Obr. 38 Profil – sekce Nastavení filmů

5 Shrnutí výsledků

Aplikace uživateli poskytuje moderní přehled filmů, a to nejen základní pohled, ale pomocí služby TMDB poskytuje rozšířený pohled na filmy včetně ukázek a pomocných funkcí jako doporučené filmy. Uživatelský účet je zabezpečen pomocí přihlášení uživatelským jménem a heslem, kdy při komunikaci s BE je využíváno Bearer tokenu. Uživateli jsou zobrazovány nejen nejlepší filmy ale také naposledy přidané a sdílené filmy ostatními uživateli, což umožňuje rychlý přehled a snadnou inspiraci.

Samotná stránka s filmy pak uživateli nabízí kompletní seznam jeho přidávaných filmů. Díky filtrům může uživatel snadno a rychle najít konkrétní film, popřípadě filmy dle nastavení. Přidávání filmu je také velice snadné a díky službě TMDB pak uživatel nemusí manuálně vyplňovat informace o filmu jako takém a stačí vyplnit pouze unikátní věci jako například jazyky a titulky. Při najetí na film má uživatel možnost film editovat, vyměnit za jiný, přidat do oblíbených, sdílených nebo film ze své kolekce smazat. Dále se mu zobrazí hodnocení filmu a krátký popis. Stránka filmů využívá paginaci, což uživateli umožňuje zobrazit další filmy z jeho sbírky.

6 Závěry a doporučení

Cílem této závěrečné práce bylo vytvořit webovou aplikaci pro koncové uživatele, která umožní jednoduchou správu vlastní filmové sbírky. Celková funkční aplikace byla vyvinuta ve spolupráci s Ondřejem Kašparem, který se postaral o vývoj BE části.

Během implementace prvních částí stránek a komponent jsem se postupně seznámil s frameworkem NextJS, jehož počáteční složitost vyžadovala čas na plné pochopení. Avšak s rozvojem mé znalosti NextJS jsem byl schopen implementovat i složitější funkcionality. Z pohledu dalšího rozvoje aplikace by bylo vhodné přepracovat některé komponenty a zvýšit jejich univerzálnost.

Aplikace pro uživatele je přehledná a přístupná jak prostřednictvím webového prohlížeče na počítači či notebooku, tak i na mobilních zařízeních. Dalším krokem v rozvoji aplikace by mohlo být vytvoření samostatné mobilní aplikace pro prohlížení filmové knihovny, například využitím jazyka Dart a frameworku Flutter (pro zařízení Android a iOS).

Závěrem lze konstatovat, že aplikace je funkční a splňuje stanovené cíle. Práce, která zahrnovala nejen vytvoření front-endové webové aplikace, ale také seznámení se s frameworkem NextJS, byla úspěšně dokončena. Přestože v současnosti nejsou zaznamenány žádné chyby, je vhodné věnovat pozornost potenciálním okrajovým případům, které by mohly vést k problémům v aplikaci, a dále ji rozvíjet.

Pro uživatele je aplikace dostupná na adrese <http://imitgw.uhk.cz:59754/>.

7 Seznam použité literatury

- [1] *STAG92816.pdf* [online]. [vid. 2024-03-12]. Dostupné z: <https://theses.cz/id/xhdwsj/STAG92816.pdf>
- [2] ONDŘEJ KAŠPAR. *Tvorba poskytovatele dat (API) volnočasové aplikace LocalMovieList*. 2024. FIM UHK – Závěrečná diplomová práce
- [3] *Co je HTML a CSS - Začínáme s HTML - cs.webdev.wiki* [online]. [vid. 2024-02-14]. Dostupné z: <https://cs.webdev.wiki/howto/co-je-html-a-css.html>
- [4] *Single Page Application vs Multi Page Application - ellow.io* [online]. 27. září 2023 [vid. 2024-02-19]. Dostupné z: <https://ellow.io/single-page-application-vs-multi-page-application/>
- [5] *Vue.js - The Progressive JavaScript Framework | Vue.js* [online]. [vid. 2024-02-19]. Dostupné z: <https://vuejs.org/>
- [6] *React* [online]. [vid. 2024-02-19]. Dostupné z: <https://react.dev/>
- [7] *Angular* [online]. [vid. 2024-02-19]. Dostupné z: <https://angular.io/>
- [8] *Next.js by Vercel - The React Framework* [online]. [vid. 2024-02-19]. Dostupné z: <https://nextjs.org/>
- [9] *Nuxt 2 - The Intuitive Vue Framework* [online]. [vid. 2024-02-19]. Dostupné z: <https://v2.nuxt.com/>
- [10] *Getting Started | Axios Docs* [online]. [vid. 2024-02-21]. Dostupné z: <https://axios-http.com/docs/intro>
- [11] *Using the Fetch API - Web APIs | MDN* [online]. [vid. 2024-02-21]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch
- [12] *react-secure-storage - npm* [online]. [vid. 2024-02-21]. Dostupné z: <https://www.npmjs.com/package/react-secure-storage>
- [13] *i18next - npm* [online]. [vid. 2024-03-12]. Dostupné z: <https://www.npmjs.com/package/i18next>
- [14] *i18nexus | Translation Management for React and Next.js* [online]. [vid. 2024-03-12]. Dostupné z: <https://i18nexus.com/>
- [15] *MUI: The React component library you always wanted* [online].

[vid. 2024-02-19]. Dostupné z: <https://mui.com/>

[16] *Ant Design - The world's second most popular React UI framework* [online]. [vid. 2024-02-19]. Dostupné z: <https://ant.design/>

[17] *NextUI - Beautiful, fast and modern React UI Library* [online]. [vid. 2024-02-19]. Dostupné z: <https://nextui.org/>

[18] *Tailwind CSS - Rapidly build modern websites without ever leaving your HTML.* [online]. [vid. 2024-02-19]. Dostupné z: <https://tailwindcss.com/>

[19] *KarimMokhtar/react-drag-drop-files: Light and simple Reactjs drag and drop files library to use with very flexible options to change, so you put whatever the design you want for your drop-area. Users can drag and drop or even select the file anywhere in the window.* [online]. [vid. 2024-02-21]. Dostupné z: <https://github.com/KarimMokhtar/react-drag-drop-files>

[20] *Swiper - The Most Modern Mobile Touch Slider* [online]. [vid. 2024-02-20]. Dostupné z: <https://swiperjs.com/>

8 Přílohy

9 Zadání diplomové práce



Univerzita Hradec Králové
Fakulta informatiky a managementu

Zadání diplomové práce

Autor:	Bc. Michal Petras
Studium:	I2100075
Studijní program:	N1802 Aplikovaná informatika
Studijní obor:	Aplikovaná informatika
Název diplomové práce:	Webové rozhraní pro aplikaci LocalMovieList
Název diplomové práce AJ:	Web interface for the LocalMovieList application

Cíl, metody, literatura, předpoklady:

Cílem této práce navrhnout a vytvořit uživatelské rozhraní webové aplikace pro správu filmové databáze. Tato část aplikace bude schopna komunikovat pomocí API s BE částí aplikace a získávat tak potřebná data, která budou uživateli zobrazena. Webové rozhraní by mělo uživateli nabídnout přehlednost, intuitivnost a uživatelskou přívětivost.

K tvorbě webového rozhraní bude použit framework NextJs založen na knihovně React.

Bude poskytnuta zadavatelem po dohodě

Zadávací pracoviště:	Katedra informačních technologií, Fakulta informatiky a managementu
Vedoucí práce:	prof. RNDr. Peter Mikulecký, Ph.D.
Datum zadání závěrečné práce:	15.10.2021