Střední průmyslová škola elektrotechnická  
a Vyšší odborná škola Pardubice

**STŘEDNÍ PRŮMYSLOVÁ ŠKOLA ELEKTROTECHNICKÁ**

**MATURITNÍ PRÁCE – WEBOVÉ STRÁNKY**

**CINEMA TOWN**

březen 2024 Jan Rožek, 4. D

*„Prohlašuji, že jsem maturitní práci vypracoval(a) samostatně a použil(a) jsem literárních pramenů, informací a obrázků, které cituji a uvádím v seznamu použité literatury a zdrojů informací a v seznamu použitých obrázků a neporušil jsem autorská práva.*

*Souhlasím s umístěním kompletní maturitní práce nebo její části na školní internetové stránky a s použitím jejích ukázek pro výuku.“*

*V Pardubicích dne ........................... ................................................*

*podpis*

*Anotace*

*Webová aplikace pro rezervování sedadel na promítání filmů. Bude umožňovat správu kin, filmů, promítání, žánrů filmů a možnost úpravy oprávnění u některých rolí. Přihlašování a registrace, bude řešena dvoufázovým ověřením za pomoci emailů.*

*Klíčová slova: rezervování sedadel, dvoufázové ověřování*

*Annotation*

*Web application for movie screening seat reservation. It will allow management of theaters, movies, screenings, movie genres and the ability to edit permissions for some roles. Login and registration will be handled by two-phase verification using emails.*

*Keywords: seat reservation, two-phase verification*

**Obsah**

[Úvod 9](#_Toc512415384)

[1 Analýza obdobných webových stránek 10](#_Toc512415385)

[1.1 Zapište název prvního webu 10](#_Toc512415386)

[1.1.1 Kladné stránky 10](#_Toc512415387)

[1.1.2 Záporné stránky 10](#_Toc512415388)

[2 Návrh projektu 11](#_Toc512415389)

[2.1 Cílové skupiny 11](#_Toc512415390)

[2.2 Administrace webu 11](#_Toc512415391)

[2.3 Databáze 11](#_Toc512415392)

[2.4 Design a responzivita 11](#_Toc512415393)

[3 Popis projektu 12](#_Toc512415394)

[3.1 Frontend 12](#_Toc512415395)

[3.2 Backend 12](#_Toc512415396)

[Závěr 13](#_Toc512415397)

[Seznam přístupových údajů 14](#_Toc512415398)

[Seznam použité literatury a zdrojů obrázků 15](#_Toc512415399)

[Seznam obrázků 16](#_Toc512415400)

[Přílohy 17](#_Toc512415401)

# Úvod

Jako maturitní projekt jsem se rozhodl vytvořit zjednodušenou aplikaci pro rezervování sedadel na promítání filmů, která umožní kompletní správu kin včetně sálů, filmů a jejich promítání. Také bude umět posílat uživatelům, kteří jsou přihlášeni, k odběru novinky (nově přidané filmy).

Tato aplikace by neměla sloužit sama o sobě k zisku peněz. Byla by doplňkem k existujícímu podnikání ve filmovém průmyslu. Poskytla by jednoduše dosažitelné informace pro zákazníky stejně jako umožnila jednoduchou správu zaměstnancům za to zodpovědným.

Ke konci studia na střední škole jsem chtěl zhotovit nějakou webovou aplikaci, za kterou by se nemusela (s přivřenýma očima a s přihlédnutím, že na tom budu pracovat sám) stydět žádná větší firma jak po vzhledové, tak po funkční stránce.

Pro tento konkrétní projekt jsem se rozhodl, protože mi jsou filmy celkem blízké, protože jsem si nedokázal vybrat zaměření internetového obchodu, který se stal pro maturitní práce standardem.

Aplikace bude rozdělena do tří serverů, Node.js s pomocí knihovny React pro frontend, Spring Boot pro backend a MongoDB jako databáze zajistí uchovávání dat. Pro toto řešení jsem se rozhodl, protože bych se chtěl zdokonalit ve výše uvedených technologiích a to, protože v nich vidím největší budoucnost.

# Analýza obdobných webových stránek

Analýzu je důležité provádět při každém vývoji nové aplikace zejména proto, abychom zjistili, jakou můžeme nabídnout konkurenční výhodu, čemu se vyhnout, anebo co je dobré z ostatních projektů převzít, případně vylepšit.

## CineStar

Adresa: [cinestar.cz](https://www.cinestar.cz)

Obsah obrázku text, oblečení, snímek obrazovky, Digitální kompozice

Popis byl vytvořen automaticky

Obrázek 1: CineStar (zdroj: vlastní)

CineStar je síť multikin s největší hustotou pokrytí v republice. Účelem webové stránky je informovat širokou veřejnost o programu kin (případně i jejich poloze) a samozřejmě umožňuje rezervaci míst na určitá promítání filmů. Po prvním vstoupení na stránku se ukáže mapa republiky s možností vybrat kino jehož program uživatele zajímá. Stránka s detaily kin se pak liší jen nejspodnější částí, kde se nachází program kina s jeho adresou a kontaktem. Toto řešení mi nepřijde nejlepší, protože uživatele, byť jen poprvé, zdržuje od vybrání filmu. Dále bych ocenil možnost v detailu filmu mít možnost provést rezervaci.

### Kladné stránky

* Přehledný program
* V programu jsou náhledy na filmy
* Filmy je možné filtrovat pomoví kategorií

### Záporné stránky

* Složitější cesta k programu
* V detailu filmu není možné provést rezervaci
* Je možné si prohlédnout filmy promítané jen ve vybraném kině

## Cinema City

Adresa: [cinemacity.cz](https://www.cinemacity.cz)

Obsah obrázku text, snímek obrazovky, grafický design, osoba

Popis byl vytvořen automaticky

Obrázek 2: Cinema City (zdroj: vlastní)

Cinema City je nadnárodní síť multikin s kořeny v Izraeli. Účel webové stránky se stejný jako u CineStar, a to informovat a rezervovat. Na úvodní stránce je možnost si prohlédnout nabídku filmů. Z počátku všechny filmy v různých kategoriích (promítáme, budeme promítat, speciální projekce a rodinné filmy), nebo při výběru kina už konkrétní program na určitý den (s výjimkou projekcí naplánovaných na dlouho dopředu). U každého filmu je jeho náhled (při kliku na něj se přehraje trailer na film) a nejpodstatnější informace o filmu. Posledním zobrazením filmů je „Podle filmu“ kde se po vybrání konkrétního filmu zobrazí všechna promítání tohoto filmu ve vybraný den.

### Kladné stránky

* Trailery na filmy
* Je možné rezervovat promítání z detailu filmy
* Možnost prohlédnout si celou nabídku filmů

### Záporné stránky

* Po vybrání kina se není možné vrátit do stavu nevybraného kina
* Vyhledávání filmu podle žánru je možné jen při vybraném kině
* Program by mi více vyhovoval jako one page s odkazy v hlavičce na jednotlivé dny

## Kino Světozor

Adresa: [kinosvetozor.cz](https://www.kinosvetozor.cz)

**Obsah obrázku text, snímek obrazovky, software, Multimediální software

Popis byl vytvořen automaticky**

Obrázek 3: Kino Světozor (zdroj: vlastní)

Stránka spojuje program několika menších pražských kin. I proto si může stránka dovolit být poměrně jednoduchá a přehledná. Nejdůležitější částí stránky je filtrovací formulář a program, který je umístěný hned pod ním. Program je možné přepínat mezi časovým (ten je vidět na ukázce) a filmovým (skládá se z dlaždic s náhledem na film a jeho jménem). Detail filmu se zobrazí jako popup s termíny projekce, ale rezervovat je možné jen nejbližší termín promítání.

### Kladné stránky

* V detailu filmu je možnost přehrát si trailer
* Přehledný program
* Jednoduchý a funkční dezign

### Záporné stránky

* Není možné vyhledávat podle žánrů
* V detailu filmu jsem byl chvilku zmatený při vybírání konkrétního promítání
* Na stránce není moc obsahu pro inspiraci

# Návrh projektu

## Cílové skupiny

Lidé, kteří rádi chodí do kina, aby se podívali na film.

## Popis projektu

Projekt jako celek bude rozdělen do tří samostatných částí (serverů). O ukládání dat se bude starat databázový server MongoDB, přístup k němu bude zprostředkovávat backendová část napsaná v jazyku Java za použití frameworku Spring Boot skrze REST API. O prezentaci dat uživateli se bude starat JavaScriptový kód s využitím knihovny React, který bude spuštěn v prostředí Node.js. Toto řešení by do budoucna umožnilo snadnou rozšiřitelnost funkcionality webové aplikace i například vytvoření mobilní aplikace (bylo by třeba jen napsat aplikaci, protože API už by bylo hotové).

## Use Case diagram

Obsah obrázku text, snímek obrazovky, kruh

Popis byl vytvořen automaticky

Obrázek 4: Use Case diagram

## Databáze

Obsah obrázku text, diagram, snímek obrazovky, Plán

Popis byl vytvořen automaticky

Obrázek 5: struktura databáze UML Class diagram

Na výše uvedeném diagramu je vidět struktura jednotlivých kolekcích v databázi (kolekce v NoSQL databázích představuje to co v relačních databázích tabulka), do kolekce se ukládají data ve formě dokumentů (dokument v NoSQL databázích představuje pak to co by v relačních databázích byl jeden záznam). Protože NoSQL nemá žádnou standardizovanou formu pro modelování grafů (jako ERD relačních databází), uvádím zde UML Class diagram, který výstižně zachycuje strukturu dat, kterou jsem použil.

Výše uvedený diagram reprezentuje strukturu databáze přesně až na jednu výjimku. Diagram např. říká, že v dokumentu Role jsou uložené dokumenty Permission. Toto řešení by ale mělo hned několik problémů počínaje porušením pravidel normalizace dat, přes zbytečně vysoké nároky na kapacitu databáze až po velmi náročnou editaci. Proto je ukládání dokumentů do dokumentů řešeno nahrazením dokumentu záznamem DBRef s parametrem kolekce, ve které se nachází nahrazený dokument a parametrem id nahrazeného dokumentu.

# Použité technologie

## JavaScrip

JavaScript je objektově orientovaný, interpretovaný jazyk (překládaný za běhu) s C-like syntaxí a dynamickým typováním, který využívá funkcionální paradigma (umožňuje uložení funkcí do proměnných). Jeho účel byl doplnit značkovací jazyky HTML a CSS na webových stránkách (manipulace s domem, složitější chování částí webu apod.) Dnes ale najde uplatnění i jako plnohodnotný programovací jazyk, v kterém může být napsaný i například program.

### TypeScript

Je nadstavba JavaScriptu, kterou vyvíjí Microsoft. Jedná se o samostatný programovací jazyk, který se kompiluje do JavaScriptu. Jeho přínos spočívá v zavedení statického typování do JavaScriptu, pohlídá volání nedefinovaných vlastností, zavádí třídy, rozhraní a výčtové typy. Přichází i s genericitou a union typy.

### React

React je open-source JavaScriptovou knihovnou, pro tvorbu UI (user interface), za jehož vývojem stojí společnost Meta a komunita individuálních vývojářů. React funguje na bázi tzv. „komponent“. Ty mají své vstupní vlastnost a vnitřní stavy, což usnadňuje vytváření znovupoužitelného kódu.

React dokáže automaticky reagovat na změnu stavů a překreslovat aplikaci jen když je to potřeba. Komponenty mohou být vytvořeny buď jako funkce nebo jako třídy.

Pokud je komponenta vytvořená jako funkce, pak musí vracet jeden JSX tag (JavaScript XML). Pokud je vytvořena jako třída, tak musí obsahovat metodu „render“, která vrací jeden JSX tag.

JSX má dost podobnou syntaxi jako HTML 5 s následujícími rozdíly:

* Je možné do něj vkládat reactové komponenty jako tagy
* Vlastnosti tagů mohou mít částečně jiný název (class je className, onclick je onClick )
* Mezi složené závorky {} je možné vkládat JavaScriptová data
* Každý tag musí být ukončen
* Všechny tagy jsou párové a jdou zapsat zkráceným zápisem
* Existuje prázdný tag, který se nevykreslí na stránku. Má využití, například když chceme, aby komponenta vracela více jak jeden tag.

JSX může vypadat následovně:

        <>

            {/\*\* Komentář \*/}

            {/\*\* Vložení funkce k eventu onClick \*/}

            <div className='management-reference' onClick={() => navigate(url)}>

                {/\*\* ukázka zkráceného zápisu \*/}

                <div className='management-reference-img' style={imgBg} />

                {/\*\* vložení komponenty filter \*/}

                <Filter />

                {/\*\* důkaz o párovosti všech tagů \*/}

                <br></br>

                {/\*\* vložení textu do odstavce \*/}

                <p>{text}</p>

            </div>

        </>

Webové rozhraní vytvořené pomocí reactu je pak SPA (Single Page Application – celá aplikace je obsažena v jednom HTML a jedno JS souboru).

### Node.js

Je prostředí, které umožňuje spouštění JavaScriptového kódu mimo webový prohlížeč. Jeho primárním účelem je tvorba serverové části webových aplikací s důrazem na vysokou škálovatelnost (schopností obsloužit mnoho připojených klientů současně). Té dosahuje tím, že jednotlivé požadavky klientů zachytává jedno vlákno, která je pak distribuuje mezi asynchronní vlákna. Asynchronní vlákna pak požadavky zpracovávají.

### NPM

Node.js ackage Manager je správce JavaScriptových balíčků, který usnadňuje jejich instalaci a údržbu.

## Java

Je objektově orientovaný programovací jazyk se striktním typováním a C-like syntaxí. Programy napsané v Javě jsou platformě nezávislé (i na operačním systému, tak na mikroprocesoru). Tuto vlastnost má díky JVM (Java Virtual Machine). Implementuje automatickou správu paměti (garbage colletor).

### Spring | Spring Boot

Jedná se o Framework pro jazyk Java, jehož cílem bylo ulehčení vývoje wabových aplikací v Javě.

Čistý Spring nepoužívám, protože by to vyžadovalo nastavení tzv. servlet kontejneru (webového serveru). Nastavení servlet kontejneru se obecně považuje za náročnější akci do, které jsem se nechtěl pouštět a místo toho jsem použil knihovnu Spring Boot, která má nastavený kontejner už implementovaný (tomcat). Dalším důvodem pro využití Spring Bootu pro mě bylo to, že Spring pouze definuje rozhraní, zatímco ve Spring Bootu jsou už i implementována.

### Maven

Je správce balíčků, který je možné použít pro Java projekty. Zařizuje přidání Spring Bootu a jiných knihoven do projektu.

## MongoDB

Jako databázi jsem zvolil dokumentově orientovanou NoSQL databázi MongoDB. MongoDB neukládá data v tabulkách a už vůbec ne v řádcích. Data jsou ukládána ve formátu BSON (Binary JSON) a jsou organizována ve formě dokumentů (v relačních databázích bychom to nazvali řádek), které jsou seskupeny do kolekcí (v relačních databázích bychom řekli tabulek). Každý dokument ale může mít jinou vnitřní strukturu (aplikace si musí sama hlídat jaká data do databáze ukládá).

Zatímco vztahy v relačních databázích jsou řešeny pomocí cizích klíčů MongoDB zná dvě vazby mezi objekty. Objekt může obsahovat jiný objekt anebo může obsahovat referenci (odkaz). Druhý způsob je podobný cizím klíčům relačních databází, ale přichází s problémem integrity dat, kterou musí opět vyřešit aplikace.

# Backend projektu

# Frontend projektu

# Popis jednotlivých částí aplikace

V této části se budu věnovat jednotlivým stránkám, které si může uživatel zobrazit.

## Úvodní stránka / Program

Po otevření webové stránky uživatele překvapí rovnou stránka s programem, která je rozdělena do tří částí (navigaci nepočítám).

První části tvoří automatická galerie obrázků filmů, které byly zvoleny jako propagační. Galerie v nezmenšené formě obsahuje dva filmy a ve zmenšené verzi jeden. Obrázky se automaticky mění po 10 sekundách. Nové filmy přijíždí z prava

Obsah obrázku text, oblečení, osoba, plakát

Popis byl vytvořen automaticky

Obrázek 6: Úvodní stránka 1. část

Druhou část tvoří filtr, ten umožňuje vybrat jedno kino a také použít vyhledávací pole, které vyhledává automaticky pro psaní a porovnává podle: jména filmu a názvu každého žánru každého filmu. Pokud není vybráno žádné kino (počáteční stav), tak je jedno v jakém kině bude probíhat promítání. Vyhledávací okno nerozlišuje malá a velká písmena, myslím si, že je to tak lepší (pro uživatele pohodlnější). Poslední částí filtru je pak čtrnácti denní přehled. Po kliknutí na den je uživatel přesunut na sekci, kde začínají filmy, které se v daný den budou promítat

Poslední, třetí, částí úvodní stránky jsou pak samotná naplánovaná promítání. Ta jsou dělena podle dnů, kdy k nim má dojít, do sekcí. Sekce jsou uspořádané podle data ze shora dolů. Promítání se skládá z obrázku filmu, nadpisu, stručného popisu filmu a dalších informací, které by mohly budoucího návštěvníka zajímat (žánry, režisér, herci, doba trvání …). Favikon uprostřed obrázku vyzívá uživatele, aby si přehrál trailer k filmu (trajler je umístěn na YouTobe a do stránky jen vložen). Uživatel se může přes název filmu překliknout na detail filmu a odtud provést rezervaci.

Obsah obrázku text, snímek obrazovky, Webové stránky, Webová stránka

Popis byl vytvořen automaticky

Obrázek 7: Úvodní stránka část 2 a 3

# Závěr

Popište význam projektu, zdůvodněte problémy při řešení daného tématu a odchylky od zadání, popište využití maturitní práce a plány do budoucna.

# Seznam přístupových údajů

URL adresa webu: [rozekja20.mp.spse-net.cz/](https://rozekja20.mp.spse-net.cz/)

|  |  |  |
| --- | --- | --- |
| Úroveň oprávnění | Přihlašovací jméno | Heslo |
| Administrátor |  |  |
| Editor |  |  |
| Registrovaný uživatel |  |  |

# Seznam použité literatury a zdrojů obrázků

Vložte seznam literatury a on-line stránek, které jste použili pro vyhledávání informací a studium.

Do seznamu zahrňte zdroje obrázků použité v písemné práci i v samotném webu (uveďte práva k použití – používejte pouze obrázky zdarma použitelné pro komerční i nekomerční využití). Pokud jste některé obrázky vytvořili sami (vlastní grafika či vlastní fotografie), uveďte jejich seznam nebo popis a zdroj vlastní.

Seznam můžete vygenerovat na [www.citace.com](http://www.citace.com)

# Seznam obrázků

Vygenerujte seznam obrázků použitých v dokumentaci (obrázky musí mít zapsané titulky) včetně čísel stránek (stránky menší než polovina stránky mohou být vloženy ve stránce, bez obtékání textu, velké obrázky budou v příloze).

# Přílohy

Use case diagram

Responzivita – vložte obrázek webu na počítači, na mobilu, příp. na tabletu.