

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Bakalářská práce

Návrh a implementace webového rezervačního systému

František Křivanec

© 2020 ČZU v Praze

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

František Křivanec

Systémové inženýrství a informatika
Informatika

Název práce

Návrh a implementace webového rezervačního systému

Název anglicky

Design and Implementation of Web Reservation System

Cíle práce

Cílem práce je vytvoření návrhu a následné implementace webového rezervačního systému v zvoleném kině.

Metodika

1. Na základě studia odborných zdrojů popište problémovou doménu formou literární rešerše
2. Provedte analýzu uživatelských požadavků na webovou aplikaci a popište je vhodnými nástroji (Use Case)
3. Provedte analýzu a zhodnocení existujících webových aplikací
4. V souladu s doporučenými postupy softwarového inženýrství vytvořte návrh vlastní webové aplikace (konceptuální model)
5. Návrh aplikace implementujte ve vhodném prostředí a řádně otestujte

Doporučený rozsah práce

30-40 stran

Klíčová slova

webová aplikace, webdesign, softwarové inženýrství, programování

Doporučené zdroje informací

- BÖHMER, M. *Návrhové vzory v PHP : [23 vzorových postupů pro rychlejší vývoj]*. Brno: Computer Press, 2012. ISBN 978-80-251-3338-5.
- BUCHALCEVOVÁ, A. *Metodiky vývoje a údržby informačních systémů : kategorizace, agilní metodiky, vzory pro návrh metodiky*. Praha: Grada, 2005. ISBN 80-247-1075-7.
- MERUNKA, V. – CARDA, A. – ČESKÁ SPOLEČNOST PRO SYSTÉMOVOU INTEGRACI, – POLÁK, J. *Umění systémového návrhu : objektově orientovaná tvorba informačních systémů pomocí původní metody BORM*. Praha: Grada, 2003. ISBN 80-247-0424-2.
- NEUSTADT, I. – ARLOW, J. *UML 2 a unifikovaný proces vývoje aplikací : objektově orientovaná analýza a návrh prakticky*. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.
- SALIM, F. – ALBERS, B. – LUBBERS, P. *HTML5 : programujeme moderní webové aplikace*. Brno: Computer Press, 2011. ISBN 978-80-251-3539-6.
- WILLIAMS, H E. – KRÁSENSKÝ, D. – LANE, D. *PHP a MySQL : vytváříme webové databázové aplikace : podrobný průvodce tvůrce WWW stránek*. Praha: Computer Press, 2002. ISBN 80-7226-760-4.

Předběžný termín obhajoby

2019/20 LS – PEF

Vedoucí práce

Ing. David Buchtela, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 14. 3. 2020

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 14. 3. 2020

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 22. 03. 2020

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Návrh a implementace webového rezervačního systému" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 23.3.2020

Poděkování

Rád bych touto cestou poděkoval Ing. Davidu Buchtelovi, Ph.D., za odborné vedení při tvorbě bakalářské práce. Bez jeho rad a připomínek by dokončení bakalářské práce bylo mnohem složitější.

Návrh a implementace webového rezervačního systému

Abstrakt

Tato práce se zabývá návrhem a následnou implementací webové aplikace, konkrétně rezervačního systému v kině. Vytvoření webové aplikace bude probíhat pomocí jazyku PHP a HTML s pomocí frameworku Bootstrap. Teoretická část se věnuje seznámení s problematikou vývoje webové aplikace pomocí uvedených technologií a seznámení s několika možnými postupy při vývoji. V praktické části je postupně popsán návrh a následný vývoj webové aplikace pracující s databází.

Klíčová slova: webdesign, webová aplikace, softwarové inženýrství, programování

Design and Implementation of Web Reservation System

Abstract

This thesis deals with the design and subsequent implementation of a web application, specifically a reservation system in a cinema. This web application will be created in PHP and HTML using the Bootstrap framework. The theoretical part is devoted to familiarization with the problems of web application development using the mentioned technologies and familiarization with several possible methods of development. The practical part describes the design and subsequent development of database web application in detail.

Keywords: webdesign, web application, software engineering, programing

Obsah

1 Úvod.....	12
2 Cíl práce a metodika	13
2.1 Cíl práce	13
2.2 Metodika	13
3 Teoretická východiska	14
3.1 Pojmy	14
3.1.1 Webová aplikace	14
3.1.2 Webový server	14
3.1.3 PHP	14
3.1.4 UML.....	15
3.1.5 HTML	15
3.1.6 Kaskádové styly	16
3.1.7 Bootstrap	17
3.1.8 XAMPP.....	17
3.2 Životní cyklus systému	17
3.2.1 Fáze životního cyklu	17
3.2.1.1 Analýza.....	18
3.2.1.2 Návrh	18
3.2.1.3 Implementace	18
3.2.1.4 Testování	18
3.2.1.5 Zahájení provozu	19
3.2.1.6 Údržba a provoz	19
3.3 Metodiky budování informačních systémů	19
3.3.1 Rigorózní metodiky	19
3.3.1.1 Model zralosti SW	19
3.3.1.2 Metodika open	20
3.3.1.3 Metodika Rational Unified Process	20
3.3.1.4 Metodika Enterprise Unified Process	21
3.3.2 Agilní metodiky	21
3.3.2.1 Dynamic Systems Development Method	21
3.3.2.2 Adaptive Software Development	22
3.3.2.3 Feature-Driven Development	22

3.3.2.4	Crystal metodiky.....	22
3.3.2.5	Scrum.....	22
4	Vlastní práce	24
4.1	Průzkum trhu	24
4.1.1	Nu Metro	24
4.1.2	Light House Cinema	24
4.1.3	Cinema Nova.....	25
4.1.4	Zhodnocení.....	25
4.2	Vzorové osoby.....	25
4.3	Stanovení cílů aplikace.....	26
4.4	Omezení práce s aplikací.....	27
4.5	Use Case specifikace	28
4.5.1	Registrace.....	28
4.5.2	Přihlášení.....	28
4.5.3	Zobrazení filmů a zobrazení představení	28
4.5.4	Odhlášení	29
4.5.5	Smazání účtu	29
4.5.6	Rezervace místa	29
4.5.7	Přehled rezervací.....	29
4.5.8	Smazání rezervací	29
4.5.9	Přidání filmu a přidání představení	30
4.5.10	Mazání filmu a mazání představení	30
4.6	Databázová vrstva	30
4.7	Drátěný model	32
4.7.1	Úvodní strana	32
4.7.2	Registrace.....	33
4.7.3	Výběr představení	34
4.7.4	Rezervace sedadla	35
4.7.5	Přehled rezervací.....	36
4.7.6	Přidání a odebrání filmů.....	37
4.7.7	Přidání a odebrání představení	38
4.8	Implementace	38
4.8.1	Přidání dat do tabulky	39
4.8.2	Mazání dat z tabulky	43
4.8.3	Přihlášení uživatele	44
4.9	Testování	46
4.9.1	Průběh testování.....	46

5	Výsledky a diskuse	47
5.1	Výsledek práce	47
5.2	Možnosti zlepšení.....	47
6	Závěr.....	48
7	Seznam použitých zdrojů	49
8	Přílohy	51
8.1	Příloha A Vytvořená aplikace – úvodní obrazovka	52
8.2	Příloha B Vytvořená aplikace – registrace.....	53
8.3	Příloha C Vytvořená aplikace – rezervování místa.....	54
8.4	Příloha D CD se zdrojovými kódy	55

Seznam obrázků

Obrázek 1-Use Case diagram	27
Obrázek 2-Přehled tabulek v relační databázi	31
Obrázek 3-ER diagram	31
Obrázek 4-Wireframe úvodní stránky	32
Obrázek 5-Wireframe registrace.....	33
Obrázek 6-Wireframe s přehledem představení	34
Obrázek 7-Wireframe rezervace sedadla	35
Obrázek 8-Wireframe přehled rezervací.....	36
Obrázek 9-Wireframe přidání a odebrání filmu	37
Obrázek 10-Wireframe přidání a odebrání představení.....	38
Obrázek 11-Varování při chybné registraci.....	46

Seznam tabulek

Tabulka 1-Vzorová persona 1	25
Tabulka 2-Vzorová persona 2.....	26
Tabulka 3-Vzorová persona 3	26

1 Úvod

Webové aplikace jsou v dnešní době součástí každodenního života většiny z nás. Ačkoliv si to někteří nemusí uvědomovat. Při dnešním využití internetu by bylo velmi nepravděpodobné se webovým aplikacím vyhnout. Setkáme se s nimi skoro na každé webové stránce. Objednáváme-li si zboží z e-shopu, používáme k tomu webovou aplikaci. Chceme-li zjistit, kdy nám jede dopravní spojení, nejjednodušší a nejrychlejší je to pomocí konkrétní webové aplikace. Pokud si budeme chtít přečíst nebo odeslat e-mail pravděpodobně toho dosáhneme využitím webové aplikace.

Webová aplikace nám oproti běžným aplikacím, které musíme instalovat do našich zařízení, nabízejí mnoho výhod. Je to především jejich jednoduchá dostupnost, ke které potřebujeme pouze webový prohlížeč a připojení k internetu, které je v dnešní době již všudypřítomné. Za obrovskou výhodu pro vývojáře lze považovat fakt, že nemusí vytvářet více verzí aplikace na různá zařízení a operační systémy, neboť webovou aplikaci je možné spustit jakýmkoli zařízením disponujícím webovým prohlížečem. Webové aplikace z našich životů určitě jen tak nezmizí, a dle mého názoru se s nimi budeme setkávat stále častěji, až do doby, dokud používání internetu nenahradíme něčím jiným.

V současnosti je možné vyvíjet webové aplikace mnoha způsoby. V mé práci sem se rozhodl použít jazyků HTML, PHP a databázi MySQL.

2 Cíl práce a metodika

2.1 Cíl práce

Hlavním cílem této práce je seznámení s problematikou vývoje webových aplikací. Následné demonstrování poznatků na konkrétním vytvoření webové aplikace, která bude umožňovat rezervování filmových představení.

Vedlejším cílem práce je navržení takovým způsobem, který zajistí jednoduché a přehledné ovládání, díky kterému bude schopen jejího používání každý uživatel bez větších problémů.

2.2 Metodika

Metodika práce bude zakládat na získávání informací z odborných, tištěných a elektronických zdrojů, které se zabývají problematikou návrhu a vývoje informačních systémů za pomoci jazyků HTML a PHP. V teoretické části budou ze zjištěných informací popsány použité technologie a frameworky s nimi související. Praktická část se bude zabývat návrhem konkrétní webové aplikace, vymezením jejích funkcí, návrhem designu, zpracováním a otestováním. Všechny tyto kroky zde budou detailně popsány a proloženy útržky kódu.

3 Teoretická východiska

3.1 Pojmy

3.1.1 Webová aplikace

Za webovou aplikaci se označují takové aplikace, které se ke svému běhu nemusí instalovat a být nahané na počítači. Jsou to aplikace, jež se spouští přes webový prohlížeč (google chrome, internet explorer, mozilla firefox), který nezná logiku samotné aplikace. Tyto aplikace běží na webovém serveru a pomocí internetu nebo intranetu se aplikace zobrazuje přes klienta, kterého tu představuje zmíněný webový prohlížeč. Díky všudypřítomnosti webového prohlížeče, jsou webové aplikace snadněji dostupné než klasické desktopové aplikace. Jelikož webová aplikace běží pomocí webového prohlížeče, nemusí být brán ohled na operační systém, nebo druh zařízení. Všude by měla fungovat stejně. (1)

3.1.2 Webový server

Jako webový server se rozumí počítač, na kterém jsou nahané webové stránky. Tyto stránky se díky propojení přes síť mohou zobrazit v jiných zařízeních, i když v nich nejsou fyzicky uloženy. Webové servery fungují v nepřetržitě zapnutém režimu, a tak musí mít odpovídající hardware. Důležitá je jeho rychlost a místo pro ukládání webových stránek. I přes takové úskalí může být server instalován na běžný počítač, čímž se docílí jakéhosi personálního web serveru. (2)

3.1.3 PHP

PHP je zkratkou názvu PHP: hypertextový preprocesor. Jedná se o velmi rozšířený skriptovací jazyk vhodný pro programování webů, jelikož může být vložen do HTML. PHP se dá používat celkem jednoduše, proto je ideální pro nováčky. Pokročilý programátoři v něm ocení velký rozsah funkcí, který PHP nabízí. Tento jazyk byl vyvinut v roce 1994 pro vlastní užití Rasmusem Lerdorfem a následující rok vyšel i pro veřejnost. Syntaxe jazyka je podobná jazykům Perl, C a Java a rozšiřuje se o několik svých věcí např. příkazy k integraci s databázemi. Pro programátora je jednoduché se v PHP naučit tvorbu

dynamických webů. Jeho užití je možné jak na Windows, tak UNIXových operačních systémech. PHP stahuje informace ze serverů, a poté je vypisuje v uživatelské prohlídce. Díky tomu je možná interakce s uživateli pomocí formulářů, či komplexní práce s databázemi a mnoho dalších možností. (3, s. 13)

3.1.4 UML

UML je zkratka pro jazyk „Unified Modeling Language“, který je určený k vizuálnímu modelování systémů. Díky jeho zabudovaným rozšiřovacím mechanismům má mnohem větší využití než jen modelování objektově orientovaných softwarových systémů. Jazyk byl navržen k tomu, aby sloučil nejlepší postupy modelovacích technik a softwarového inženýrství. Jeho navržení je uděláno tak, že je možné ho implementovat nástroji CASE (computer-aided software engineering). Tato koncepce je založena na faktu, že velké systémy by bez nástrojů CASE fungovali jen stěží. Ačkoliv v jazyku UML můžeme pozorovat nějaké aspekty metodiky modelování, jazyk jako takový žádné tyto funkce nemá. Pouze dokáže poskytnout vizuální syntaxi, kterou je možné následně použít k sestavení dalších modelů. (4, s. 28)

3.1.5 HTML

HTML je jazyk, sloužící k vytváření dokumentů s hypertextovými odkazy a pokročilejším formátováním. Díky jeho elementům lze vytvářet webové stránky mnoha způsoby:

- Dokáže měnit vzhled, velikost či styl písma.
- Vkládá do dokumentu obrázky a animace jakýchkoliv tvarů.
- Vytvoří formuláře, kterými lze odpovídat na dotazníky nebo objednávat z e-shopu.
- Pomocí tabulek pomůže zlepšit kontrolu nad dokumentem a udělat ho srozumitelnější.
- Vytvořit pozadí celého, nebo jen části dokumentu.
- Vytvořit odkaz buď to v rozsahu dokumentu, nebo i na jiné dokumenty, stránky, případně obrázky a jiné soubory.

Z toho lze vyvodit, že HTML má velké spektrum využití. I přes to jsou zde značná omezení, která by začínajícího programátora mohla zaskočit. Je třeba vědět, že mnoho funkcí, jež na webových stránkách mohou nalézt, nelze vytvořit jen pomocí HTML:

- HTML nedokáže ověřit uživatele.
- Jsou jím dělány jen statické dokumenty, nedokáže vytvořit obsah, který by se automaticky měnil.
- Nedokáže vytvořit dynamicky se měnící obsah jakým je např. vysouvací menu a jemu podobné věci.

Od svého vzniku mělo HTML několik fází. První verze vznikla roku 1990 a neoficiální HTML+ byla uvedena až 1993. Rysy této verze byly formuláře, tabulky a obrázky, nebylo zde ale formátování odstavců a úprava textu. Dalo se zde použít 78 elementů, ale spousta jich již v tomto jazyce není. V roce 1994 přišla verze, která se stala standardem. Měla formální specifikaci a 49 elementů. S verzí HTML 3 bylo možné obtékat obrázky textem, přidávat tabulky a matematické elementy. HTML 4 přidalo podporu prvku OBJECT, který je významný pro vkládání multimédií. Podporuje CSS a skriptování na straně uživatele. Nyní by aktuální verzí mělo být HTML 5, které přišlo s novými elementy a zjednodušilo zápis u starších. Dále je možnost přehrát multimédia přímo v prohlížeči. (5, s. 16)

3.1.6 Kaskádové styly

Kaskádové styly zkráceně CSS jsou opravdovým přelomem na poli formátování webových stránek. Ačkoli CSS nemění obsah dokumentů, dokáže je přísně strukturovat tak, aby vyhovovali logice i standardům. Díky tomu mohou se stránkami bezproblémově pracovat i jiné aplikace než běžné prohlížeče. (6, s. 15)

HTML dokumenty se předtím, než bylo možné používat CSS, formátovali skrz značky a jejich atributy. Stránky se potom rozvrhovali pomocí tabulky. Kvůli tomu, jak se HTML dokumenty stávali složitějšími, narostli i problémy jako např. že datový objem formátování dokumentu byl větší, než samotný objem obsahu dokumentu. Na základě toho bylo pomalé načítání stránek a narůstali tím i nároky na server. Pokud formátují dokumenty přes kaskádové styly, tak jednoduše odstraní tyto problémy a umožní vytvoření čistého HTML dokumentu. Takový dokument vyhovuje přísným standardům a je přístupný všem aplikacím. (6, s. 18)

3.1.7 Bootstrap

Byl vyvinut Markem Ottem a Jacobem Thorntonem pro Twitter jako framework podporující konzistenci mezi interními nástroji. V srpnu 2011 byl vydán jako open source a v únoru následujícího roku se stal nejoblíbenějším developerským projektem na GitHubu. Výhodou je snadné zpracování všech uživatelských rozhraní ve webových aplikacích. Je kompatibilní s poslední verzí všech hlavních prohlížečů a dokáže se přizpůsobit použití i na starších verzích. Od verze 2.0 podporuje responzivní design, který zajistí dynamické rozložení na stránky v závislosti na použitém zařízení. (7)

3.1.8 XAMPP

Jedná se o odlehčenou distribuci Apache, která vývojáři vytvoří lokální webový server, na němž se dá provádět vývoj a testování webových aplikací. Instalační balíček obsahuje serverovou aplikaci (Apache), databázi (MariaDB) a skriptovací jazyk (PHP). Dokáže běžet jak na Windows, tak i na Mac OS a Linux. Díky tomu, že většina webových serverů využívá stejné komponenty, je přechod z testovacího serveru na živý vcelku jednoduchý. (8)

3.2 Životní cyklus systému

U složitých softwarových projektů je nevyhnutelné členění na dílčí jasně a přehledné fáze. Ty se poté dají ilustrovat v praktických příkladech. V začátku je vymodelován uživatelský proces a následný reengineering. Děje se tomu z důvodů, že nový software zasahuje do mechanismů řídicích struktur. Dalším krokem je vytvoření konceptuálních a softwarových modelů což vyústí v softwarovou implementaci. (9, s. 81)

3.2.1 Fáze životního cyklu

Životní cyklus systému je posloupnost etap, která u daného systému vede od chvíle, kdy byl zadán požadavek, přes vývoj, uvedení do provozu, údržbu a končí až s ukončením provozu. Během zadání je dobré vycházet ze stávajícího stavu a postupně definovat požadované chování systému. (9, s. 81)

3.2.1.1 Analýza

Analýza je problém z podnikatelského prostředí. Během této etapy se vymezuje problém a zjišťují procesy, které se v systému budou odehrávat. Je třeba rozebrat samotný problém a seznámit se s procesy v systému. Postupně se dostáváme z AS-IS (fáze jak to je) k TO-BE (jak to má být). To někdy může vyvolat i značné změny procesu a organizace prostředí, ve kterém se má celý systém implementovat. (9, s. 82)

3.2.1.2 Návrh

Návrh se dá označit za první fázi, během které je snaha o samotné sestavení systému, k tomu, aby bylo možné implementovat software. Na rozdíl od analýzy už je v tuto chvíli vše ze zadání hotovo a rozpoznáno. Dále se rozhoduje, který programovací jazyk, databáze apod. má pro implementaci nejlepší vlastnosti. Též se zjišťuje problematika okolo opakované použitelnosti softwarových artefaktů a napojení systému na komponenty celku. (9, s. 82)

3.2.1.3 Implementace

Během implementace se programuje samotný program. Tím se rozumí, že vše, co bylo do této doby navrženo, naplánováno a odsouhlaseno převedu na požadovaný software. Je třeba, aby byla vedena aktuální dokumentace. Ta je nezbytná do budoucna, pokud bude software následně rozšiřován a, nebo udržován. (9, s. 82)

3.2.1.4 Testování

Během testování softwaru jde o nalezení všech chyb, které se při implementaci v programu vyskytli. V podstatě se při něm kontroluje, zda výsledný program odpovídá tomu, jak byl navrhnout a nepostrádá žádné funkce, které měly být jeho součástí. (10)

Pokud je během testování nalezen nějaký nedostatek, vrací se daný program zpět do fáze implementace, aby se objevený problém opravil. Poté co je oprava hotová je nutné provádět testování znovu, aby se zjistilo, zda nedošlo k dalším chybám během opravy.

3.2.1.5 Zahájení provozu

Pokud je aplikace po otestování shledána bezchybnou, je možné uvést ji do provozu. To znamená, že je nahrána na nějaký server, odkud již bude dostupná pomocí internetu či intranetu. Ačkoliv je aplikace otestována a shledána funkční. Měla by být ještě po nějakou dobu sledována, neboť všechny chyby nemusely být nalezeny a opraveny.

3.2.1.6 Údržba a provoz

Poslední etapou života aplikace je provoz a s tím související údržba. S touto etapou je spojena velká finanční a časová náročnost. Odpovídá 100% - 400% náročnosti od počáteční fáze po samotnou implementaci aplikace. (9, s. 82)

3.3 Metodiky budování informačních systémů

V současnosti jsou metodické přístupy děleny do dvou proudů, jsou jimi rigorózní metodiky a agilní metodiky. Odlišují se od sebe především podle kritéria váha metodiky, ale i několika dalšími hledisky. U rigorózní metodiky se vychází z toho, že procesy, které při budování IS/ICT mohu popsat, plánovat, řídit a měřit. Procesy zde bývají podrobně definované a z toho důvodu jsou velmi objemné. (11, s. 22)

Agilními metodikami jsou označovány ty, které dokáží velmi rychle vytvářet řešení a pružně ho přizpůsobit měnícím se požadavkům. Jsou založeny na myšlence, že prověření správnosti systému lze dosáhnout jeho rychlým vyvinutím, předložením zákazníkovi a upravením na základě zpětné vazby. (11, s. 33)

3.3.1 Rigorózní metodiky

3.3.1.1 Model zralosti SW

Jedná se o nejznámější metodiku hodnocení softwarových procesů. Byl vyvinut, aby ohodnotil softwarové řešení pro ministerstvo obrany USA. Současná podoba je dostupná od roku 1995. Tento model má 5 úrovní zralosti. Na základě zralosti jsou identifikovány důležité oblasti ke zlepšení procesů. Přínosem je porovnání zralých a nezralých procesů. Nezralých procesů se dosáhne improvizací. Z toho plyne, že i když procesy existují, není možné je opakovat a předvídat. Z těchto důvodů se na problémy pouze reaguje a opravují

se chyby. To vede k překročení plánů a rozpočtů, jelikož jsou založeny na nerealistických odhadech. Poté ve snaze dodržet časové termíny, dochází ke snížení kvality. Kvůli nedostatku času se v takové situaci neprovádějí činnosti, které by kvalitu zvětšily. Zato zralé procesy jsou ty, díky kterým je možné řídit vývoj software v rámci celé organizace. (11, s. 22)

3.3.1.2 Metodika open

Jedná se o veřejně přístupnou metodiku, která je schopna podpořit celý cyklus vývoje IS/ICT. Zaměřuje se především na objektově orientované a komponentové aplikace. Jejím cílem je navržení univerzální metodiky pro objektově orientovanou analýzu. OPEN definuje procesní rámec a generuje z něj instance specifické pro organizaci. Tyto instance jsou vybírány z činností, úloh a technik s jejich specifickou konfigurací. Metodika rozliší přizpůsobení procesu, které přizpůsobí úlohu tak, aby vyhovovala problémové doméně. Díky tomu je OPEN flexibilní v přizpůsobení se jak doméně, tak taky konkrétnímu problému. Lze jej používat na velké i malé klíčové projekty. OPEN dokáže podporovat celý životní cyklus aplikace. (11, s. 28)

3.3.1.3 Metodika Rational Unified Process

Tato metodika (zkráceně RUP) se zakládá na nejlepších praktikách softwarového vývoje:

- Iterativní vývoj
- Řízení požadavků
- Použití komponentové architektury
- Vizuální modelování
- Kontrola kvality software
- Řízení změn

Samotný proces při vývoji softwaru se dá popsat dvěma dimenzemi. První dimenze je představena dynamickým pohledem na proces vyjádřený cykly, fázemi, iteracemi a milníky. Druhá dimenze naopak nahlíží na proces staticky. Popisuje činnosti, artefakty, pracovníky a pracovní toky. Život softwaru je zde rozdělen pomocí cyklů, které se mění s každou novou verzí produktu. Každý cyklus je rozdělen na 4 fáze:

- Počáteční fáze
- Elaborační fáze
- Konstrukční fáze
- Fáze nastavení

Během počáteční fáze jsou definovány cíle a požadavky projektu, harmonogram, odhad ceny a rizika. Je možné vytvořit jednoduchý prototyp, který slouží k otestování, zda je možné splnit hlavní požadavky. Tato fáze je ukončena rozhodnutím, zda lze daný projekt realizovat. V elaborační fázi dochází k definování architektury systému pomocí vytvořeného prototypu, na kterém jsou ověřeny potřebné náležitosti a zpřesní realizaci systému. Také se definují komponenty pro opakované použití. V konstrukční fázi se implementuje a testuje systém. Pokud je to možné děje se tak pomocí paralelního vývoje. Během nasazení je zajištěna schopnost uživatelů daný systém používat. Uživatelé jsou proškoleni, vytvoří se help-desk atd. (11, s. 29)

3.3.1.4 Metodika Enterprise Unified Process

Zkráceně nazývána EUP je dílem Scotta W. Amblera a jedná se o rozšíření metodiky RUP. Rozšiřuje tuto metodiku dvěma směry. V prvním se jedná o rozšíření na úrovni celé organizace, kde EUP zahrnuje procesy realizované přes projekty. V druhém směru rozšiřuje RUP o fáze *Production*, která řeší provoz a údržbu systému. Fáze *Retirement*, která se zabývá náležitostmi kolem odstranění z užívání. (11, s. 31)

3.3.2 Agilní metodiky

3.3.2.1 Dynamic Systems Development Method

Tato metodika pochází z Velké Británie a ze všech agilních metodik je postavena na nejlépe propracovaném systému školení a kvalitní dokumentaci. Jedná se o rozšíření rychlého vývoje aplikací. Charakteristické je, že se dokáže přizpůsobit změnám ve vývoji. (11, s. 35)

3.3.2.2 Adaptive Software Development

Metodika ASD spočívá v tom, že pokud nastávají změny, je důležité se na ně adaptovat. Z toho důvodu je statický cyklus nahrazen dynamickým. Dynamický životní cyklus je postaven na tom, že probíhají neustále změny, na základě kterých, je možné se učit. V běžném životním cyklu je problémem plánování, díky němu je sice docíleno plánovaných výsledků, ale je zabráněno vzniku různých inovací. Naproti tomu spekulace nabízí možnost změn a experimentů. Běžně je odchýlení od plánu chápáno jako chyba, ASD to však vidí jako možnost naučit se něco nového, bez rozdílu, jestli odchylka znamenala chybu nebo úspěch. (11, s. 37)

3.3.2.3 Feature-Driven Development

Tato metodika má iterativní vývoj. Ten začíná vytvořením modelu a následnými dvoutýdenními iteracemi, během nichž, probíhá návrh a realizace užité vlastnosti. Za takovou vlastnost je označován malý, ale zato užitečný výsledek a dá se spolu s ostatními realizovat ve dvoutýdenních iteracích. Tak jako je tomu u ostatních agilních metod, ani FDD nepřeceňuje význam procesů během vývoje. Snaží se o to, aby výsledný produkt fungoval namísto splnění předepsaného procesu. (11, s. 41)

3.3.2.4 Crystal metodiky

Jedná se o celou rodinu metodik. Jejich výhodou je silná komunikace a lehkost produktu. Její prvky jsou přizpůsobeny každému projektu. Vhodná metodika je vybrána na základě tří hledisek. Prvním je velikost projektu, která se odvodí podle počtu lidí v týmu. Druhým hlediskem je důležitost systému. Posledním hlediskem je, pro co je metodika optimalizována (produktivita, testovatelnost apod.). (11, s. 42)

3.3.2.5 Scrum

Tato metodika má přesvědčení, že se jedná o empirický proces na rozdíl od rigorózních metodik, které předpokládají definovaný proces, a proto jsou řízeny naprosto odlišným stylem. Název vychází z ragbyového slangu, čímž naznačuje, že je podobně adaptivní, rychlá a samoorganizovaná stejně jako hra. Klade důraz na řízení projektu.

Software se vyvíjí v třiceti denních iteracích nazývaných sprint, které vedou k dodání množiny užitých vlastností. Během této metodiky je každý den pořádána patnácti minutová porada. Na tuto poradu chodí všichni členové týmu, ale také manažeři. Členové si zde předávají znalosti a zodpovídají tři základní otázky.

- Co bylo dokončeno od minulé porady
- Co nového se má řešit
- Jaká jsou omezení a překážky v řešení úkolu (11, s. 43)

4 Vlastní práce

Tato část práce pojednává o převedení informací z předchozí části na vytvoření konkrétní webové aplikace. Počínáje stanovením funkcionalit, které by daná aplikace měla zvládnout. Navržením databází a grafického modelu. Vytvořením webové aplikace. A v posledním bodě otestování funkcionalit, zda aplikace dokáže vykonat stanovené cíle. Webová aplikace je tvořena za pomoci jazyků HTML a PHP a využití frameworku Bootstrap.

4.1 Průzkum trhu

V současné době je možnost rezervace přes internet očekávaná v každém odvětví a proto i většina kin tuto možnost nabízí. Nejběžnějším způsobem je využití webové aplikace, díky čemuž je možné celý systém ovládat z jakéhokoliv zařízení s webovým prohlížečem a přístupem k internetu.

4.1.1 Nu Metro

Rezervace vstupenek u společnosti Nu Metro probíhá v několika krocích. V prvním kroku si zákazník vybere film, který chce vidět. Druhým krokem je výběr kina, jež chce navštívit, datum a čas promítání. Dalším krokem je výběr sedadel a případného výběru slevy na vstupenku. V posledním kroku zákazník zadá své kontaktní údaje a provede platbu.

4.1.2 Light House Cinema

Společnost Light House Cinema má dva základní způsoby výběru představení. Lze si zobrazit celý program, který zobrazí všechny filmy a u nich kina a časy vysílání. Druhým způsobem je postupné filtrování, kdy se nejprve volí kino, poté film a nakonec čas vysílání. Ve chvíli kdy zákazník vybere představení, na které chce provést rezervaci, je přesměrován na další stránku. Zde si vybere kolik vstupenek, v jaké kategorii by si chtěl rezervovat. Dalším krokem je výběr sedadla v sále. V posledním kroku zadá své kontaktní údaje a pokračuje k platbě.

4.1.3 Cinema Nova

U společnosti Cinema Nova rezervace začíná výběrem filmu a zvolením konkrétního vysílání. Tím se zákazník přesměruje na stránku, kde zadá kolik lístků, v jaké kategorii by si chtěl rezervovat. Zadá své kontaktní údaje a provede platbu.

4.1.4 Zhodnocení

Z průzkumu je patrné, že společnosti využívají podobných postupů při rezervování. Výhodou všech společností je možnost vytvoření rezervace i bez nutnosti uživatelského účtu. Za slabinu lze považovat absenci možnosti zvolení sedadel u společnosti Cinema Nova.

4.2 Vzorové persony

Vzorová persona je pojem, se kterým se setkáváme především v marketingu. Jedná se o smyšlené osoby, které by měli představovat cílovou skupinu pro vytvořený produkt. Díky správně zvoleným vzorovým personám se dá jednodušeji cílit na požadavky, jež bude konkrétní skupina od produktu očekávat. Kromě marketingu se persony hodí i do vývoje, neboť na základě cílové skupiny je třeba navrhnout správný design a funkce aplikace.

Pro vzorovou personu je třeba si určit její pohlaví, jméno, věk, zaměstnání, záliby, průběh běžného dne atp...

Tabulka 1-Vzorová persona 1

Jméno	Leona Bílá
Pohlaví	Žena
Věk	21
Zaměstnání	Studentka vysoké školy humanitního oboru, brigádně pracuje v kině
Záliby	Četba, gymnastika, modeling
Běžný den	Probuzení, hygiena, dopolední vyučování, po obědě s přáteli jde na trénink gymnastiky nebo brigádu, večer tráví s rodinou nebo čtením knih, osobní hygiena, spánek.

Tabulka 2-Vzorová persona 2

Jméno	Šárka Svátá
Pohlaví	Žena
Věk	42
Zaměstnání	Po vystudování střední gastronomické školy pracuje jako servírka
Záliby	Rodina, cestování, vaření
Běžný den	Probuzení, hygiena, práce v restauraci během snídání a obědů, odpoledne vyzvedne děti ze školy a snaží se s nimi trávit čas doma nebo je vezme do kina či divadla, večer příprava večeře, hygiena, spánek.

Tabulka 3-Vzorová persona 3

Jméno	Leoš Vratký
Pohlaví	Muž
Věk	19
Zaměstnání	Student gymnázia
Záliby	Fotbal, fitness, video hry
Běžný den	Probuzení, hygiena, škola, po škole jde na fotbalový trénink nebo do posilovny, večer jde na rande s přítelkyní, nebo hraje počítačové hry, hygiena, spánek.

4.3 Stanovení cílů aplikace

Cílem této webové aplikace je umožnění rezervací na filmová představení a věci s touto činností související. Aplikace by tedy měla splňovat následující body:

- Vytvoření uživatelského účtu
- Smazání uživatelského účtu
- Vytvoření rezervace na konkrétní představení
- Přehled rezervací uživatele
- Zrušení vytvořené rezervace

- Možnost vytváření představení administrátorem
- Možnost rušení představení administrátorem
- Možnost přidání filmu do databáze administrátorem
- Možnost smazání filmu z databáze administrátorem

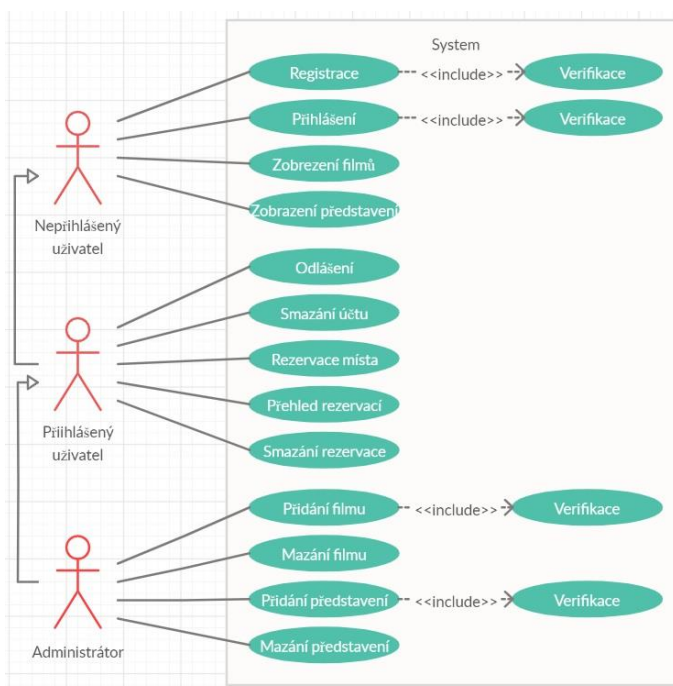
4.4 Omezení práce s aplikací

Aplikace má tři vrstvy pravomocí pro práci s ní. Tyto vrstvy se od sebe liší tím, zda aplikaci používá nepřihlášený uživatel, přihlášený uživatel, nebo administrátor.

Pokud se aplikaci rozhodne používat pouze nepřihlášený uživatel, jeho možnosti práce s aplikací jsou značně omezené. V jeho pravomocích je pouze nahlédnutí do přehledu filmů, které v kině jsou, přehledu představení, obsazenost sálu a možnosti založení vlastního účtu.

Pokud je aplikace používána přihlášeným uživatelem, přibude k výše uvedenému možnost smazání svého účtu, vytvoření rezervace ke konkrétním představením a přehled svých rezervací s možností jejich zrušení.

Administrátorovi bude navíc umožněno starat se o přidávání a mazání filmů a představení, které je v kině možné navštívit.



Obrázek 1-Use Case diagram

4.5 Use Case specifikace

4.5.1 Registrace

Popis: umožní vytvoření nového uživatelského účtu.

Akteři: nepřihlášený uživatel, systém.

Podmínky spuštění: uživatel není přihlášen.

Základní tok: systém vygeneruje formulář. Uživatel tento formulář vyplní a odešle.

Systém provede validaci dat. Systém uloží nového uživatele.

Alternativní tok: při chybném vyplnění formuláře systém upozorní uživatele a formulář neodešle. Uživatel znovu vyplní formulář a odešle.

Podmínky pro dokončení: Bude vytvořen nový uživatelský účet

4.5.2 Přihlášení

Popis: změní aktéra v systému z nepřihlášeného uživatele na přihlášeného uživatele.

Akteři: nepřihlášený uživatel, systém.

Podmínky spuštění: uživatel není přihlášen.

Základní tok: uživatel zadá své přihlašovací údaje do příslušného formuláře, který odešle. Systém provede validaci přihlašovacích údajů. Systém přihlásí uživatele.

Alternativní tok: při chybně zadaných přihlašovacích údajích systém uživatele na tuto skutečnost upozorní. Uživatel znovu vyplní formulář a odešle.

Podmínky pro dokončení: uživatel změněn z nepřihlášeného na přihlášeného.

4.5.3 Zobrazení filmů a zobrazení představení

Popis: zobrazení seznamu filmů dostupných v kině.

Akteři: všichni uživatelé, systém.

Podmínky spuštění: uživatel se nachází na příslušné stránce. V systému již jsou vytvořeny filmy.

Základní tok: systém uživateli zobrazí všechny dostupné filmy v kině se základními informacemi.

4.5.4 Odhlášení

Popis: umožní změnu uživatele z přihlášeného na nepřihlášeného.

Akteři: přihlášený uživatel, systém.

Podmínky pro spuštění: uživatel je přihlášen.

Základní tok: uživatel zvolí možnost odhlášení.

Podmínky pro dokončení: systém odhlásí uživatele.

4.5.5 Smazání účtu

Popis: umožňuje smazání uživatelského účtu.

Akteři: přihlášený uživatel, systém.

Podmínky pro spuštění: uživatel je přihlášen.

Základní tok: uživatel zvolí možnost smazání účtu.

Podmínky pro dokončení: systém odhlásí uživatele a vymaže ho z databáze.

4.5.6 Rezervace místa

Popis: umožňuje vytvoření rezervace.

Akteři: přihlášený uživatel, systém.

Podmínky pro spuštění: uživatel je přihlášen, nachází se na příslušné stránce.

Základní tok: systém zobrazí sedačky v sále. Sedačky, které budou obsazené, budou označeny a nebude možné si je vybrat. Uživatel si vybere volná místa a výběr odešle.

Systém provede rezervaci vybraných míst.

Podmínky pro dokončení: systém uloží nové rezervace do databáze.

4.5.7 Přehled rezervací

Popis: zobrazí seznam rezervací konkrétního uživatele.

Akteři: přihlášený uživatel, systém.

Podmínky pro spuštění: uživatel je přihlášen, nachází se na příslušné stránce.

Základní tok: systém uživateli zobrazí jím vytvořené rezervace a informace o nich.

4.5.8 Smazání rezervací

Popis: umožňuje zrušení konkrétní rezervace.

Aktéři: přihlášený uživatel, systém.

Podmínky pro spuštění: uživatel je přihlášen, nachází se na příslušné stránce.

Základní tok: uživatel odešle systému informaci, kterou rezervaci chce smazat.

Podmínky pro dokončení: systém odebere rezervaci z databáze.

4.5.9 Přidání filmu a přidání představení

Popis: umožní přidání nového filmu do databáze kina.

Aktéři: administrátor, systém.

Podmínky pro spuštění: je přihlášen admin a nachází se na příslušné stránce.

Základní tok: systém vygeneruje formulář. Administrátor vyplní formulář a odešle.

Systém zkontroluje validitu vstupů. Systém uloží nový film.

Alternativní tok: pokud administrátor zadá vstupy špatně, systém jej na to upozorní.

Administrátor zadá nové vstupy a formulář odešle.

Podmínky pro dokončení: systém přidá film do databáze.

4.5.10 Mazání filmu a mazání představení

Popis: umožní smazání filmu z databáze.

Aktéři: administrátor, systém.

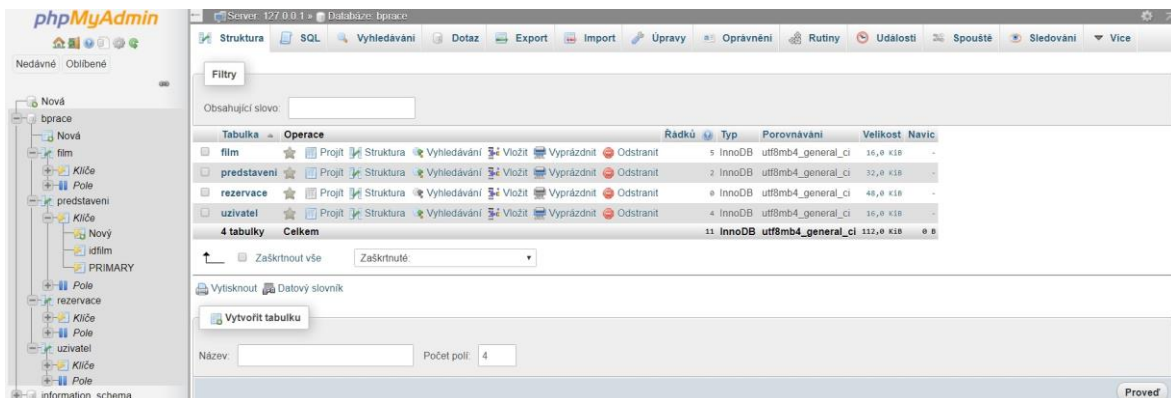
Podmínky pro spuštění: je přihlášen admin, nachází se na příslušné stránce a v databázi již existují filmy.

Základní tok: administrátor odešle systému, jaký film má být smazán.

Podmínky pro dokončení: systém smaže film z databáze.

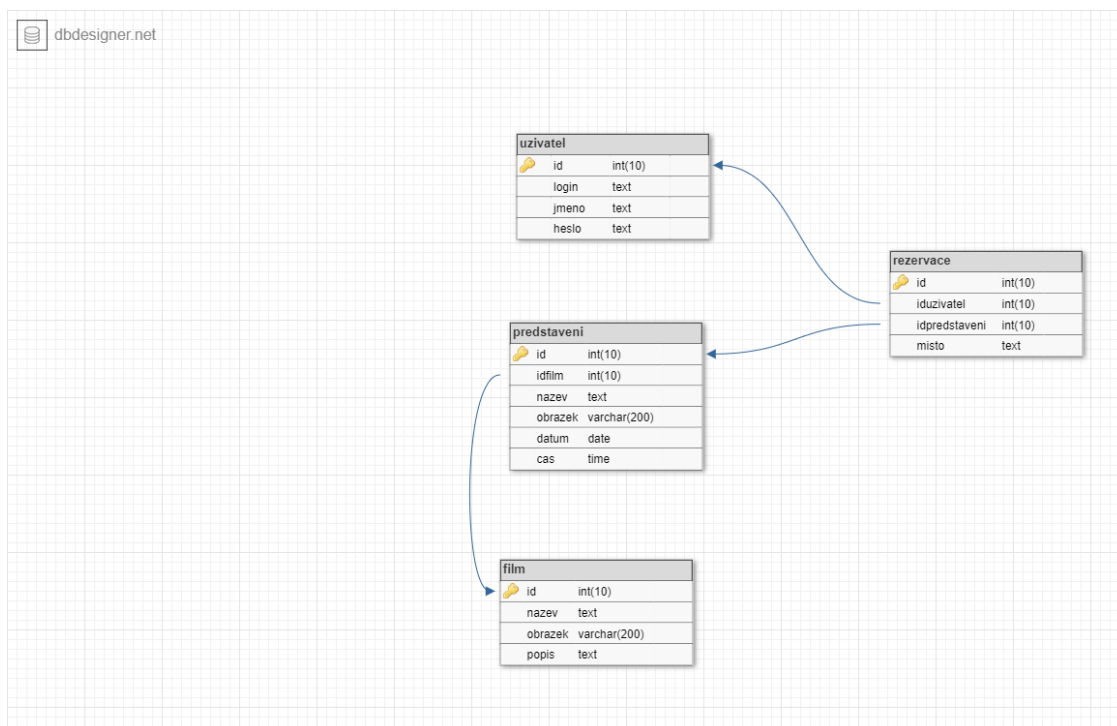
4.6 Databázová vrstva

Kvůli velkému množství dat, které aplikace ke svému běhu potřebuje je třeba tyto data ukládat do relačních databází. Databáze je vytvořena pomocí phpMyAdmin, kde se můžou jednoduše vytvořit potřebné tabulky.



Obrázek 2-Přehled tabulek v relační databázi

Pro ukládání všech potřebných dat bylo třeba do databáze vložit čtyři tabulky. Do první se ukládají filmy, které jsou v nabídce kina. Konkrétně specifické ID, název filmu, obrázek k filmu a popis filmu. Do druhé tabulky se ukládají registrovaní uživatelé, tedy jejich ID, jméno, login a heslo k přihlášení. Třetí tabulka slouží k ukládání představení v kině, má tedy konkrétní ID, ID filmu, který se má vysílat, datum a čas. Poslední čtvrtá tabulka slouží k uložení jednotlivých rezervací, proto se do ní ukládá ID rezervace, ID uživatele, který si rezervaci udělal, ID představení, na které rezervace je a sedadlo, které si uživatel rezervoval.

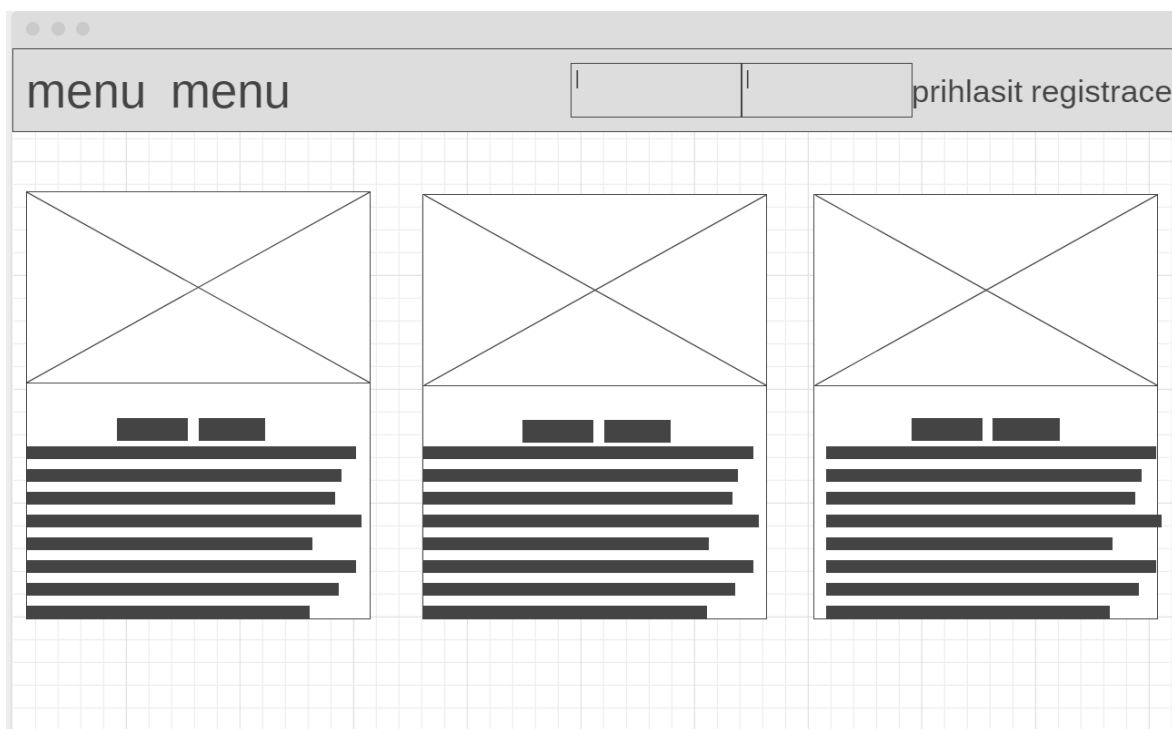


Obrázek 3-ER diagram

4.7 Drátěný model

Drátěné modely (wireframe) jsou tvořeny na stránce wireframe.cc kde se dají jednoduše navrhnout. Vytvořeny byly wireframy pro všechny části aplikace.

4.7.1 Úvodní strana



Obrázek 4-Wireframe úvodní stránky

Na úvodní stránce se uživateli zobrazí menu s odkazy na další stránky a možnost přihlášení nebo registrace. Toto menu bude dostupné na každé stránce. Pokud bude přihlášen administrátor, objeví se i odkazy na stránky, které běžný uživatel navštívit nemůže. Jedná se o stránky, na nichž bude možné přidávat a odebírat filmy a představení. Dále bude na úvodní stránce přehled filmů, které má kino právě v nabídce a jejich stručný popis.

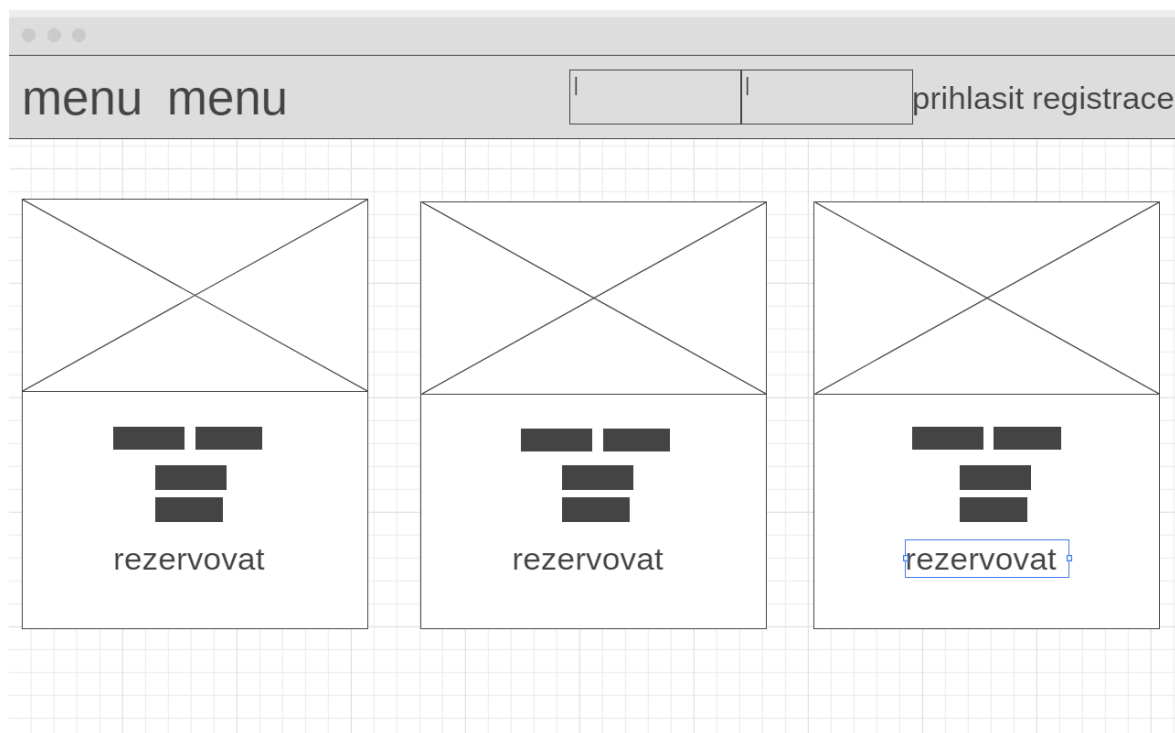
4.7.2 Registrace

The image shows a wireframe of a registration form within a browser window. The browser's address bar contains the text "menu menu" and a "přihlasit registrace" button. The main content area features a grid background with the following elements: two black rectangular boxes at the top, three input fields stacked vertically, and two buttons at the bottom labeled "založit" (grey) and "storno" (red).

Obrázek 5-Wireframe registrace

Na stránce s registrací bude uživatel vyzván k zadání tří informací. Konkrétně se bude jednat o login, jméno uživatele a heslo k účtu. Bude kontrolováno, zda byly zadány všechny pole a zda není zadáván duplicitní účet. Založení účtu je pro fungování aplikace důležité, neboť nepřihlášený uživatel nebude moci využívat hlavní funkce aplikace.

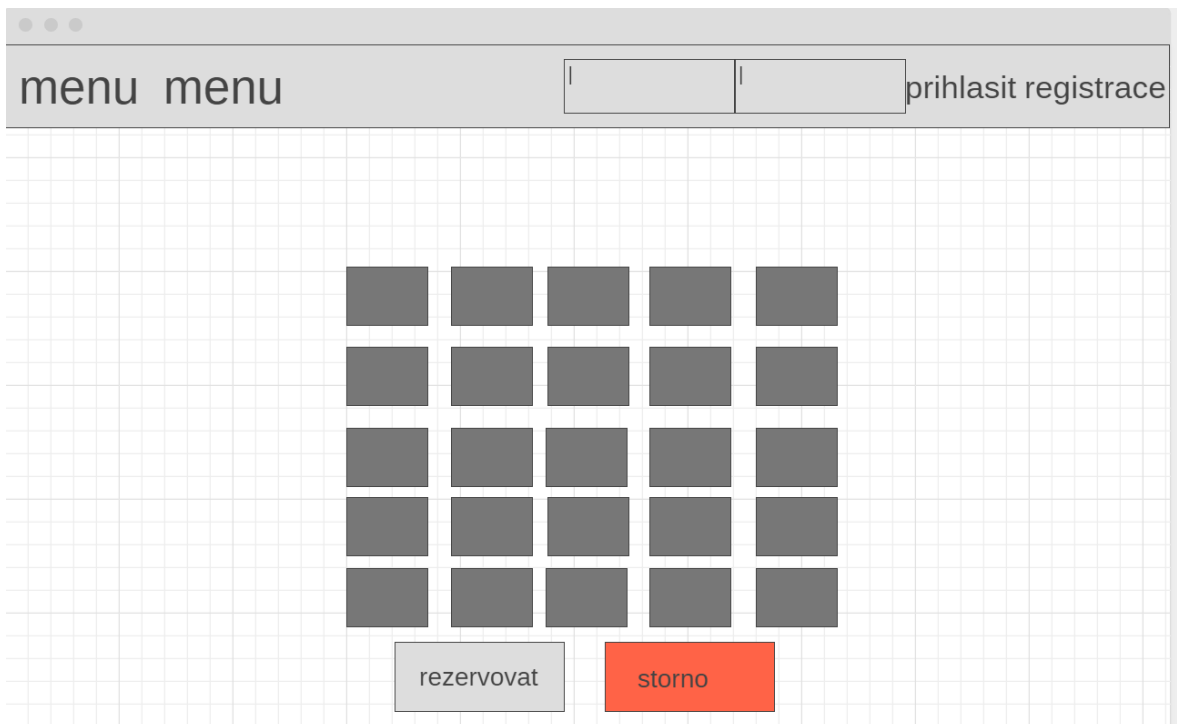
4.7.3 Výběr představení



Obrázek 6-Wireframe s přehledem představení

Na stránce rezervace se uživateli zobrazí přehled představení, které v kině budou probíhat. U představení bude uvedena informace o tom, který film bude vysílán, jaký datum se bude vysílání konat a v jaký čas. Uživatel si zde vybere představení a přesměruje se na stránku s výběrem sedadel.

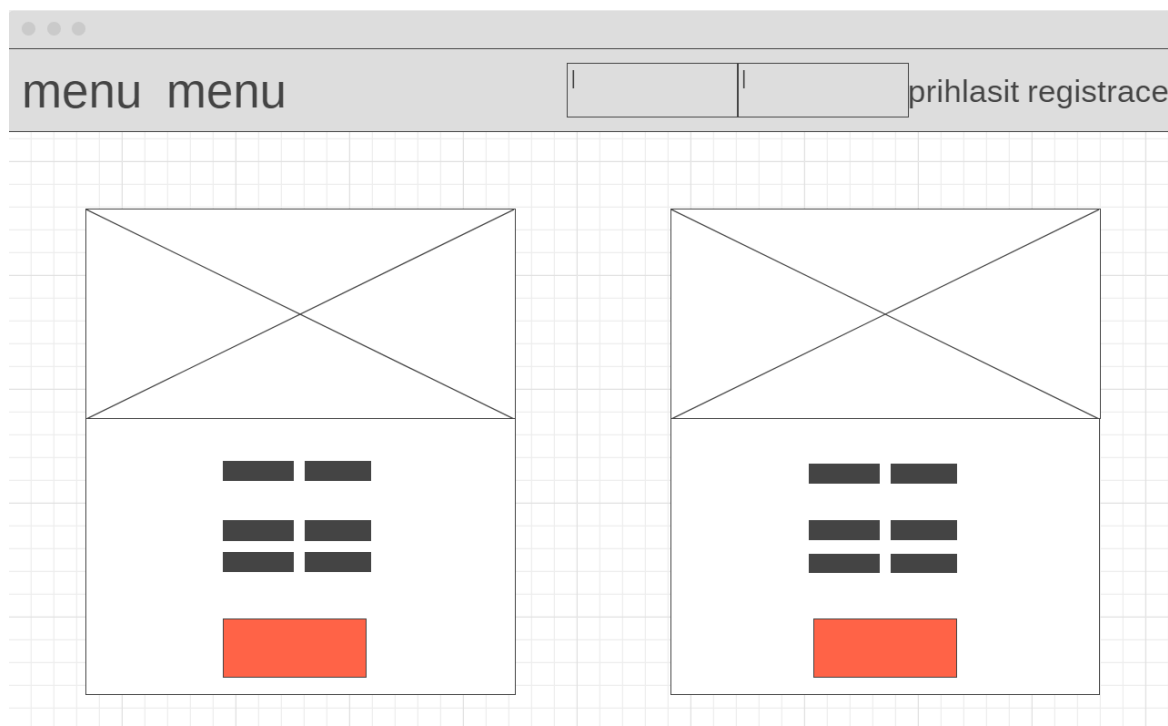
4.7.4 Rezervace sedadla



Obrázek 7-Wireframe rezervace sedadla

Na další stránce si bude moci uživatel vybrat sedadlo, které chce rezervovat. Na výběr bude mít celkem z 25 míst. Sedadla, jež jsou již zamluvena, budou označena a nebude možné si je rezervovat. Pro potvrzení rezervace je třeba být přihlášen. Nepřihlášenému uživateli bude oznámeno, že bez přihlášení není možné rezervaci provést.

4.7.5 Přehled rezervací



Obrázek 8-Wireframe přehled rezervací

Na stránce s přehledem rezervací se uživateli zobrazí, v případě že má nějakou aktivní rezervaci, seznam jeho rezervací. U každé rezervace bude uvedeno, o jaký film se jedná, kdy se bude vysílat a jaké sedadlo je rezervované. Tyto rezervace zde bude možné rušit jedním stisknutím tlačítka.

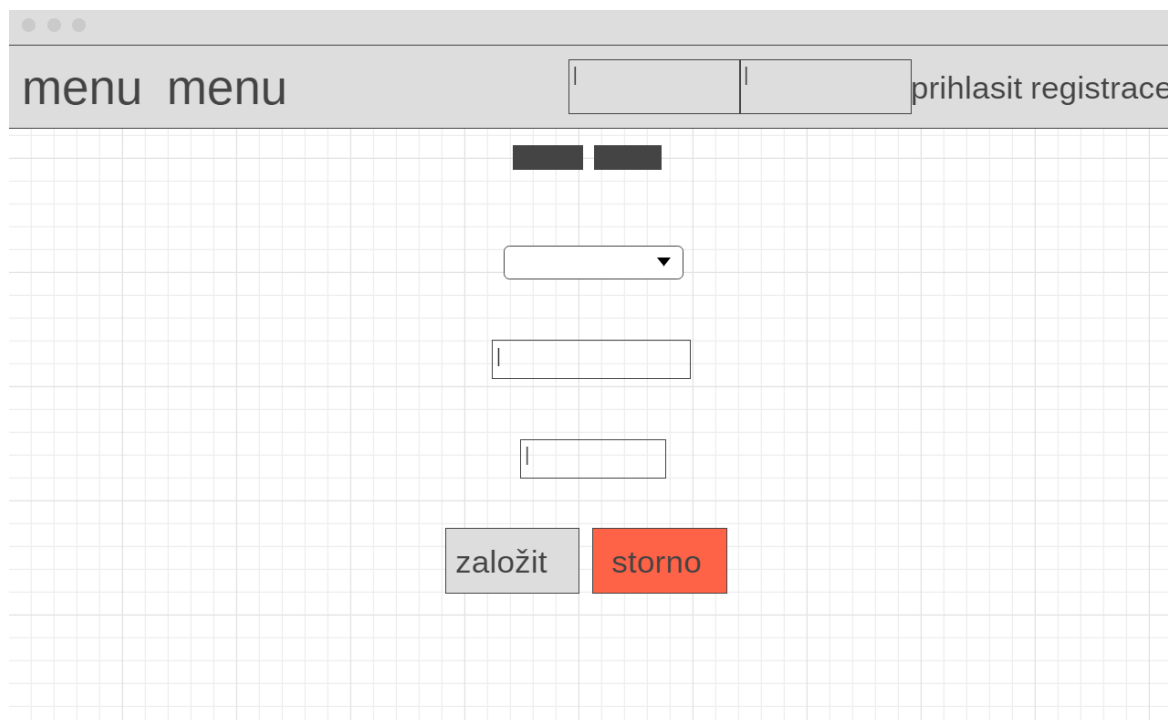
4.7.6 Přidání a odebrání filmů

The wireframe shows a web interface for managing movies. At the top, there is a navigation bar with the text "menu menu" on the left and "přihlasit registrace" on the right. Below the navigation bar, there are two small black rectangular boxes. Underneath these are two input fields, one above the other. Below the second input field is a larger, wider input field. At the bottom of the form, there are two buttons: a grey button labeled "založit" and a red button labeled "storno". The entire form is set against a light grey grid background.

Obrázek 9-Wireframe přidání a odebrání filmu

Na stránku s přidáním a odebráním filmu bude umožněno vstoupit pouze administrátorovi. Ten zde bude moci přidávat nové filmy, které se v kině budou vysílat. Bude nucen zadat název filmu, přidat k filmu obrázek a napsat popis. Pokud jsou v kině již nějaké filmy, zobrazí se mu a on je zde bude moci mazat z databáze.

4.7.7 Přidání a odebrání představení



Obrázek 10-Wireframe přidání a odebrání představení

Tato stránka je obdobou předchozí stránky. Bude sem mít přístup opět pouze administrátor. Zde bude moci vytvářet nová představení, která budou v kině probíhat. Bude nucen k výběru filmu, který se bude promítat vybraní data a zadání času promítání. Pokud budou již nějaká představení založená, bude zde možnost k jejich smazání.

4.8 Implementace

Aplikaci se začala implementovat tím, že se nejprve jednotlivé stránky vytvořily pomocí jazyka html tak aby odpovídali logickému designu. Jako první byl vytvořen dokument index.html, kde se vytvořilo menu a dále jednotlivé dokumenty s hlavním obsahem stránek. Poté co byly jednotlivé stránky hotové, je třeba stáhnout a nainstalovat program xampp, díky kterému je umožněno do stránek implementovat PHP a MySQL databázi. V phpMyAdmin se vytvořila databáze s konkrétními tabulkami, do kterých se budou ukládat záznamy z aplikace. Když je databáze hotova může se přesunout k vkládání PHP kódu do dokumentů. Některé PHP kódy jsou psány do samostatných souborů. Jedná se především o kódy, které přidávají a mažou údaje v tabulkách.

4.8.1 Přidání dat do tabulky

Přidání dat do tabulky je řešeno zadáním potřebných informací do inputů na stránce a následným odesláním dat do PHP scriptu. Z důvodu, že data nejdou následně upravovat, je po uživateli vyžadováno zadání všech potřebných dat, jinak založení nového záznamu není možné. Například nejde přidat do databáze film, aniž by byl zadán jeho název, nebo nemůže být zaregistrován nový účet bez toho, aby se k němu zvolilo heslo.

```

<form class="form-horizontal" action='predstaveni.php' method="POST" align="center">
<fieldset>
  <div id="legend">
    <legend class="">Nové představení</legend>
  </div>
  <div class="control-group">
    <label class="control-label" for="film">Film</label>
    <div class="controls">
      <select id="nazev" name="nazev">
        <?php
          $conn = mysqli_connect('localhost', 'root', '', 'bprace');
          if (!$conn){
            die("Connection failed: " . mysqli_connect_error());
          }
          $dotaz = "SELECT * FROM film";
          $vysledek = mysqli_query($conn, $dotaz);
          while($radek=mysqli_fetch_assoc($vysledek)){
            echo '<option value="'. $radek["nazev"]."'>'. $radek["nazev"].'</option>';
          }
        ?>
      </select>
      <p class="help-block">Vyber film</p>
    </div>
  </div>
  <div class="control-group">
    <label class="control-label" for="datum">Datum</label>
    <div class="controls">
      <input type="date" id="datum" name="datum" placeholder="" class="input-xlarge">
      <p class="help-block">Zadej datum</p>
    </div>
  </div>
  <div class="control-group">
    <label class="control-label" for="cas">Čas</label>
    <div class="controls">
      <input type="time" id="cas" name="cas" placeholder="" class="input-xlarge">
      <p class="help-block">Zadej čas</p>
    </div>
  </div>
  <div class="control-group">
    <!-- Button -->
    <div class="controls">
      <button class="btn btn-success" type="submit">Založit</button>
      <button class="btn btn-danger">Storno</button>
    </div>
  </div>
</fieldset>
</form>

```


V ukázce výše je možné vidět kód, který definuje form pro přidávání nového představení. Je zde vidět PHP kód, jež z tabulky s filmy generuje názvy, aby zajistil, že představení můžou vytvářet jen ke konkrétním filmům. Poté co se zvolí i další hodnoty jako je datum a čas představení. Odešlou se tyto hodnoty metodou post do dalšího souboru, který zajistí uložení nového představení v tabulce s představeními. Kód tohoto souboru je vidět v ukázce níže.

```

<?php
$nazev = filter_input(INPUT_POST, 'nazev');
$datum = filter_input(INPUT_POST, 'datum');
$cas = filter_input(INPUT_POST, 'cas');
if (!empty($nazev)) {
if (!empty($datum)) {
if (!empty($cas)) {
$conn = new mysqli ('localhost', 'root', '', 'bprace');
if (mysqli_connect_error()) {
die();
}
else{
$soverenidatum = mysqli_query( $conn, "SELECT * FROM predstaveni WHERE datum='".$datum."'");
if (mysqli_num_rows($soverenidatum)>0) {
echo '<script>
alert("Tento den jiz predstaveni je");
</script>';
echo '<script>
window.location.href="http://localhost/bprace/index.php?p=novepredstaveni";
</script>';
}
else{
$vsyledek = mysqli_query($conn, "SELECT * FROM film WHERE nazev='".$nazev."'");
$radek = mysqli_fetch_assoc($vsyledek);
$idfilm = $radek["id"];
$obrazek = $radek["obrazek"];
$sql = "INSERT INTO predstaveni ( idfilm, obrazek, nazev, datum, cas)
values ('$idfilm', '$obrazek', '$nazev', '$datum', '$cas)";
}
if ($conn->query($sql)) {
echo '<script>
alert("Úspěšně založeno");
</script>';
echo '<script>
window.location.href="http://localhost/bprace/index.php?p=uvod";
</script>';
}
else{
echo '<script>
alert("Predstaveni nezalozeno $idfilm");
</script>';
echo '<script>
window.location.href="http://localhost/bprace/index.php?p=novepredstaveni";
</script>';
}
$conn->close();
}
}
}

```

V této části kódu je vidět, že soubor přijmul data z inputů a uložil si je do proměnných. Zkontroloval, zda jsou zadány všechny body a navázal spojení s databází, do které bude hodnoty ukládat. Nachází se zde i podmínka, která má zaručit, aby nebylo možné přidávat více představení v jeden den. Pokud jsou všechny podmínky splněny, jsou data uloženy do databáze a je možné s nimi dále pracovat.

4.8.2 Mazání dat z tabulky

Mazání dat je prováděno jednoduše stisknutím jednoho tlačítka. Toto tlačítko zajistí odeslání ID mazaného objektu do potřebného kódu. Kód, kterým jsou data z tabulky smazána je vidět v ukázce níže.

```
<?php
if (isset($_POST['idpredstaveni'])){
$idzruseni=$_POST['idpredstaveni'];
    $conn = mysqli_connect('localhost', 'root', '', 'bprace' );
    $sql="DELETE FROM predstaveni WHERE id='$idzruseni'";
    $conn->query($sql);
}
    echo '<script>

        window.location.href="http://localhost/bprace/index.php?p=novepredstaveni";
        </script>';
?>
```

Zde je vidět, že kód přijme ID k představení, které má být smazáno. Podle specifického ID si najde konkrétní představení a odstraní jej z tabulky. Vztahy v databázi jsou nastaveny tak, že pokud je smazáno představení odstraní se i všechny rezervace k němu vytvořené.

4.8.3 Přihlášení uživatele

K přihlášení uživatele je potřeba zadat přihlašovací údaje, na základě kterých se uživatel přihlásí. Přihlášení proběhne, jen pokud je v tabulce s uživateli uložen záznam se shodným loginem a heslem. Po přihlášení se tlačítka pro přihlášení nebo registraci změní na tlačítka pro odhlášení nebo mazání účtu. Informace o přihlášeném uživateli se ukládají do session a jsou zde uloženy, až do případného odhlášení, smazání účtu a, nebo vypnutí okna.

```

<?php
session_start();
if(isset($_POST['prihlasit'])){
    $conn = mysqli_connect('localhost', 'root', '', 'bprace');
    $login = $_POST['login'];
    $heslo = $_POST['heslo'];
    if (!$conn) {
        die("Connection failed: " . mysqli_connect_error());
    }
    $dotaz = "SELECT * FROM uzivatel WHERE login='$login' AND heslo='$heslo' ";
    $radek = mysqli_fetch_assoc(mysqli_query($conn, $dotaz));
    $_SESSION['iduzivatel'] = $radek["id"];
    $result = mysqli_query($conn, $dotaz);
    if (mysqli_num_rows($result) > 0) {
        $_SESSION['heslo'] = $heslo;
        $_SESSION['login'] = $login;
        session_save_path();
    } else {
        echo '<script>
            alert("Neplatne prihlaseni");
        </script>';
        echo '<script>
            window.location.href="http://localhost/bprace/index.php?p=uvod";
        </script>';
    }
    mysqli_close($conn);
}
if(isset($_GET['logout'])){
    session_destroy();
    unset($_SESSION);
    header("location: index.php?p=uvod ");
}
if(isset($_GET['smazat'])){
    $conn = mysqli_connect('localhost', 'root', '', 'bprace');
    $login = $_SESSION['login'];
    $heslo = $_SESSION['heslo'];
    $dotaz = "SELECT * FROM uzivatel WHERE login='$login' AND heslo='$heslo' ";
    $radek = mysqli_fetch_assoc(mysqli_query($conn, $dotaz));
    $_SESSION['iduzivatel'] = $radek["id"];
    $result = mysqli_query($conn, $dotaz);
    $idzruseni=$_SESSION['iduzivatel'];
    $sql="DELETE FROM uzivatel WHERE id='$idzruseni'";
    $conn->query($sql);
    session_destroy();
    unset($_SESSION);
    header("location: index.php?p=uvod ");
}
?>

```

4.9 Testování

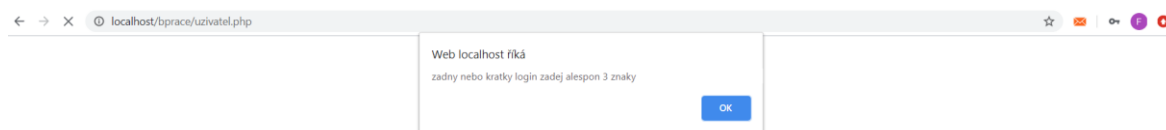
Testování bylo prováděno během vývoje aplikace. Vždy když byla dokončena implementace konkrétní funkce aplikace, bylo provedeno několik jednoduchých manuálních testů, zda aplikace funguje podle očekávání. Po dokončení celé aplikace bylo provedeno testování všech funkcí znovu, abych bylo zajištěno, že aplikace bude obsahovat co nejmenší množství chyb.

4.9.1 Průběh testování

Většina testů byla zaměřena především na validitu vstupů. Bylo to například:

- Otestování, zda je možnost založit účet bez zadání hesla nebo loginu.
- Zda mohu založit dva účty se stejným loginem.
- Jestli je možné se přihlásit neexistujícími údaji.
- Zda mohu založit více představení ve stejný den.
- Jestli je možné se dostat na stránky určené jen pro admina zadáním přímé adresy url na jiném účtu.
- Zda se při smazání představení smažou i rezervace na toto představení vytvořené.

Na obrázku níže je možné vidět varování, které se zobrazí při vynechání loginu během registrace nového uživatele.



Obrázek 11-Varování při chybné registraci

5 Výsledky a diskuse

5.1 Výsledek práce

Výsledek práce je rezervační systém pro fiktivní kino. Aplikace dokáže provádět operace nad databází jako je vytváření, mazání a čtení záznamů. Pro uživatele je snadno ovladatelná a přehledná. Přesto si ji nedovolím srovnávat s již existujícími rezervačními systémy velkých kin. Aplikace splňuje všechny funkce, které jí byly stanoveny, ale pro komerční využití by jistě potřebovala ještě v mnohém zlepšit, což zvolená architektura umožňuje.

5.2 Možnosti zlepšení

Mezi první funkce, které by se daly do aplikace naprogramovat pro její zlepšení, bych zařadil větší přehled pro administrátora okolo rezervací. Byly by to například přehled, jaká místa má konkrétní uživatel zaregistrována z důvodů vydání lístku na tyto místa. Dále možnost zablokování míst, která si přišli zákazníci zabrat osobně. Přidání rezervace na konkrétního uživatele, kdyby rezervace probíhala například telefonicky. Možnost přidání více sálů pro možnost využití aplikace ve větších kinech. Přidání možnosti vyhledávání nebo filtrů, pro jednodušší nalezení konkrétního představení. Dále by bylo dobré přidat funkci, která by smazala již proběhlá představení a rezervace. Důležité by bylo i lepší ošetření vstupů jako npř. omezení vytvoření nového představení, pokud by se v sále vysílalo, nebo kontrola věkové přístupnosti při rezervaci místa.

6 Závěr

V této práci bylo v teoretické části popsáno, co to je webová aplikace a technologie související s jejím vytvořením. Dále zde byl popsán životní cyklus a možné metodiky postupu při vývoji takové aplikace.

Praktická část se již zabývala samotným vývojem webové aplikace pracující s databází. Tato aplikace byla psána především pomocí HTML, PHP a několika frameworky s nimi souvisejícími. Byl zde ukázán postup vytvoření databáze pomocí programu xampp, který se staral o vytvoření lokálního serveru pro chod aplikace.

7 Seznam použitých zdrojů

1. Webová aplikace. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2020-02-18]. Dostupné z: https://cs.wikipedia.org/wiki/Webová_aplikace
2. Webový server. *WikiKnihovna* [online]. [cit. 2020-02-18]. Dostupné z: http://wiki.knihovna.cz/index.php/Webový_server
3. LEISS, Oliver a Jasmin SCHMIDT. *PHP v praxi: pro začátečníky a mírně pokročilé*. Praha: Grada, 2010. Průvodce (Grada). ISBN 978-80-247-3060-8.
4. ARLOW, Jim a Ila NEUSTADT. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky*. 2., aktualiz. a dopl. vyd. Brno: Computer Press, 2007. Průvodce (Grada). ISBN 978-80-251-1503-9.
5. PÍSEK, Slavoj a Ila NEUSTADT. *HTML: začínáme programovat*. 4., aktualiz. vyd. Praha: Grada, 2014. Průvodce (Grada). ISBN 978-80-247-5059-0.
6. CYROŇ, Miroslav a Ila NEUSTADT. *CSS - kaskádové styly: praktický manuál*. 4., aktualiz. vyd. Praha: Grada, 2006. Průvodce (Grada). ISBN 80-247-1420-5.
7. Bootstrap. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2020-03-11]. Dostupné z: <https://cs.wikipedia.org/wiki/Bootstrap>
8. XAMPP. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2020-03-11]. Dostupné z: <https://cs.wikipedia.org/wiki/XAMPP>
9. POLÁK, Jiří, Antonín CARDA a Vojtěch MERUNKA. *Umění systémového návrhu: objektově orientovaná tvorba informačních systémů pomocí původní metody BORM*. 4., aktualiz. vyd. Praha: Grada, 2003. Management v informační společnosti. ISBN 80-247-0424-2.
10. Co je testování softwaru? *Kitner* [online]. [cit. 2020-02-19]. Dostupné z: https://kitner.cz/testovani_softwaru/co-je-testovani-softwaru/

11. BUCHALCEVOVÁ, Alena, Antonín CARDA a Vojtěch MERUNKA. *Metodiky vývoje a údržby informačních systémů: kategorizace, agilní metodiky, vzory pro návrh metodiky*. 4., aktualiz. vyd. Praha: Grada, 2005. Management v informační společnosti. ISBN 80-247-1075-7.

8 Přílohy

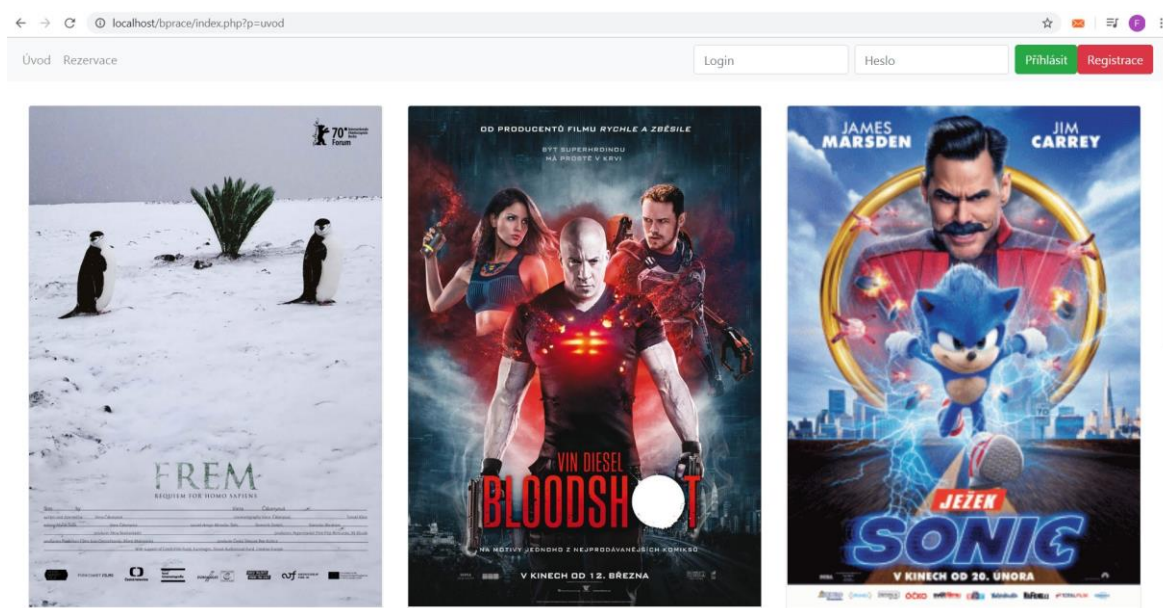
Příloha A Vytvořená aplikace – úvodní obrazovka

Příloha B Vytvořená aplikace – registrace

Příloha C Vytvořená aplikace – rezervování místa

Příloha D CD se zdrojovými kódy

8.1 Příloha A Vytvořená aplikace – úvodní obrazovka



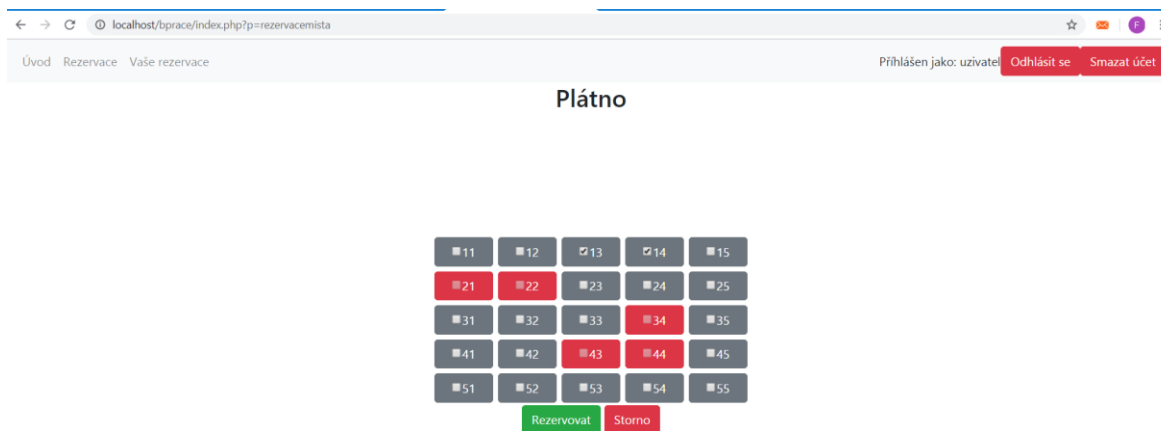
8.2 Příloha B Vytvořená aplikace – registrace

The screenshot shows a web browser window with the address bar containing the URL `localhost/bprace/index.php?p=novyuzivatel`. The browser's navigation bar includes a back arrow, a forward arrow, a refresh icon, and a search icon. Below the address bar, there is a navigation menu with the items "Úvod" and "Rezervace". To the right of the menu are two input fields labeled "Login" and "Heslo", followed by two buttons: a green "Přihlásit" button and a red "Registrace" button.

The main content area of the page is titled "Nový uživatel" and contains a registration form. The form consists of the following elements:

- A label "Login" above an empty text input field.
- A label "Zadej login" below the first input field.
- A label "Jméno" above an empty text input field.
- A label "Zadej jméno" below the second input field.
- A label "Heslo" above an empty text input field.
- A label "Zadej heslo" below the third input field.
- Two buttons at the bottom: a green "Založit" button and a red "Storno" button.

8.3 Příloha C Vytvořená aplikace – rezervování místa



8.4 Příloha D CD se zdrojovými kódy