

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačů



Semestrální projekt
Portál pro správu umělecké činnosti divadla
Anna Zlámalová

Vedoucí práce: Ing. Božena Mannová, Ph.D.

Softwarové inženýrství a technologie,
Bakalářský

2018

Poděkování

Tímto bych ráda poděkovala své vedoucí semestrální práce Ing. Boženě Mannové, Ph.D. za rady a trpělivost. Dále také své rodině a příteli za jejich finanční a psychickou podporu během studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracovala samostatně a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 21. 1. 2018

.....

Abstrakt

Tato bakalářská práce je zaměřena na vývoj webové aplikace sloužící k řízení vnitřní umělecké organizace divadel. Součástí práce je také průzkum, jehož cílem je zjistit, jak současně organizace v divadlech funguje. Na základě těchto poznatků jsou stanoveny konkrétní požadavky na funkcionality systému, navržena architektura systému a jsou zvoleny vhodné technologie pro jeho vytvoření. Při návrhu je také kladen důraz na funkčnost a responzibilitu uživatelského rozhraní. V praktické části se práce zabývá samotnou implementací aplikace.

Abstract

This work is focused on development of web application served to manage inner artistic organization of theatres. Research is also a part of this work and it's purpose is to find out how the organization in theatres works. Based on these findings are made concrete requirements for the functionality of the system, designed architecture of the system and appropriate technologies are chosen. The design also puts emphasis on functionality and responsiveness of user interface. The practical part of work is concerned with implementation itself.

Obsah

Úvod	1
Motivace	1
Popis problému	1
Cíl práce	1
Přínos práce	1
Stručný popis systému	1
Obsah základních kapitol	2
1 Analýza řešení	3
1.1 Průzkum organizace v divadlech	3
1.2 Průzkum existujících systémů	5
1.3 Specifikace požadavků	6
1.4 Případy užití	8
2 Návrh systému	10
2.1 Architektura systému	10
2.2 Datová struktura	11
2.3 Návrh grafického rozhraní	14
3 Implementace	16
3.1 Použité nástroje a technologie	16
4 Závěr	18
5 Literatura	19
6 Příloha 1 - Entitně-relační diagram	20

Motivace

Mou motivací pro výběr tohoto tématu byly především osobní zkušenosti z pracovního prostředí divadla, konkrétně divadla SEMAFOR, kde jsme se občas potýkali s problémy s organizací směn a napadlo mě vytvořit takový uživatelský portál, který by tuto organizaci usnadnil a zároveň zpřehlednil důležité informace, týkající se každodenního chodu divadla.

Popis problému

Problém, kterým se v této práci zabývám, se týká zpřehlednění umělecké organizace v divadlech. Pojem „umělecká organizace“ používám právě proto, že se nebudu zabývat organizací finanční. Tento pojem by se dal jinak přeložit jako „řízení lidských zdrojů“. Problém je především v tom, že podstatné informace týkající se směn zaměstnanců jsou roztroušené na více místech a jejich dohledání je zbytečně složité.

Cíl práce

Cílem práce je tedy navrhnout, implementovat a otestovat webovou aplikaci pro správu umělecké činnosti divadel. Konkrétně má sloužit ke správě současných směn, vytváření směn nových, tzv. fermanů, k zobrazení směn uživatelům na základě jejich přístupových práv, úpravu profilů zaměstnanců, vyhledávání informací o směnách a zaměstnancích. Naopak se práce nebude zabývat řízením financí divadla a řízením prodeje lístků.

Přínos práce

Aplikace má mimo jiné usnadnit proces plánování směn na další měsíce tím, že systém bude přístupný nejen lidem zajišťujícím chod divadla, ale i ostatním zaměstnancům, kteří si do systému mohou zadávat, kdy mohou a kdy naopak nemohou přijít do divadla. Nejdůležitějším přínosem práce je však zpřehlednění veškerých informací týkajících se zaměstnanců, informací o nich, o jejich směnách a jejich již výše zmíněné dostupnosti.

Stručný popis systému

V systému je udržován aktuální seznam představení, herců, kteří hrají v daných představeních a dalších zaměstnanců jako jsou technici, zvukaři, osvětlovači, uvaděčky, kostymérky či zaměstnanci kanceláře. Jednotliví zaměstnanci budou mít do aplikace přístup na základě různých uživatelských rolí, shodných s jejich funkcemi v divadle. Pokud k tomu mají pravomoc, mohou v systému přidávat, upravovat či mazat směny, upravovat svůj profil, případně profil někoho jiného, tvořit fermany, vyhledávat potřebné informace a tyto informace pak zobrazit. Co se uživatelského rozhraní týče, bude navrženo tak, aby bylo uživatelsky přívětivé a responzivní, a tím dostupné všem uživatelům i z mobilu přes webový prohlížeč.

Obsah základních kapitol

Práce se skládá celkem ze 4 kapitol, přičemž první tři kapitoly zastávají jednotlivé fáze vývoje software.

- Kapitola **1** se zabývá **analýzou** řešené problematiky. A to průzkumem existujících řešení organizace v divadlech. Na základě průzkumu jsou vytvořeny funkční a nefunkční požadavky a požadavky na zavedení řešení, zanalyzovány uživatelské role v systému a jejich případy užití.
- Kapitola **2** se věnuje **návrhu** architektury systému a jeho komponent, diagramům spojených s návrhem software a návrhu prototypu uživatelského rozhraní.
- Kapitola **3** popisuje samotnou **implementaci** uživatelského portálu a konkrétní zvolené technologie pro vývoj.
- Kapitola **4** je samotným **závěrem** práce, ve kterém shrnuje výsledky, hodnotí splnění cíle práce a uvádí možnost uplatnění řešení v praxi.

Analýza řešení

V této kapitole byl proveden průzkum v několika divadlech, na základě kterého byly stanoveny požadavky na aplikaci a jednotlivé případy užití.

1.1 Průzkum organizace v divadlech

1.1.1 Divadlo SEMAFOR

Divadlo SEMAFOR je středně velké divadlo umístěné v Praze v Dejvicích a proslavilo se hlavně díky Jiřímu Suchému a Jitce Molavcové, kteří spolupracují již od roku 1970. Co se organizace týče, je divadlo rozdělené do více částí podle funkcí, které jednotliví zaměstnanci zastávají. Mezi tyto funkce patří herci, muzikanti, technici, osvětlovači, zvukaři, kostymérky, uvaděčky, barmanky a kancelářská administrativa.

1.1.1.1 Herci a muzikanti

Tyto dvě funkce se liší pouze v názvu, ale jsou téměř totožné, protože každý herec v divadle SEMAFOR je muzikant a každý muzikant je herec. Zároveň jsou to velmi podstatné funkce, od kterých se odvíjí určování tzv. fermanů, neboli rozpisů, na další měsíc. Tvorbu fermanů má na starost jedna osoba, která se s herci/muzikanty domlouvá přes e-mail, kdy mohou a kdy naopak nemohou hrát. Na základě těchto informací sestaví rozpis na další měsíc, podle kterého si pak rozdělují směny i ostatní zaměstnanci.

1.1.1.2 Technici, osvětlovači, zvukaři, barmanky a kostymérky

Každá z těchto částí má nejvýše 2 až 4 tzv. alternace, neboli zastoupení. Přičemž se v každé části domlouvají zaměstnanci mezi sebou a směny si rozdělí. To pak nahlásí v kanceláři, kde tyto informace zapíše do excel tabulek.

1.1.1.3 Uvaděčky

Princip tvorby rozpisů uvaděček je téměř podobný jako výše zmíněný, liší se však v tom, že je zde kolem 16 alternací a jedna hlavní uvaděčka. Ta dohlíží na to, aby byly všechny směny obsazené a aby uvaděčky na představení dorazily. Počet uvaděček na směnu se liší podle představení. Jednotlivé směny jsou variabilní, tj. se může stát, že si uvaděčky směny vymění i po zapsání do rozpisu.

1.1.1.4 Kancelář

Lidé v kanceláři jsou v divadle i během dne a starají se o veškerou administrativu. Pokud si chtějí zjistit, kdo kdy má jakou směnu, musí informaci najít v excelovských tabulkách nebo se jít zeptat daného člověka.

1. Analýza řešení

Představení se pak dělí na 3 druhy, kterými jsou regulární/standardní představení, zájezdy, což jsou semaforová představení, ale hraná v jiných divadlech, a dále pronájmy jiných divadel.

1.1.2 Divadlo ABC a Rokoko

Divadlo ABC a Rokoko se řadí mezi činoherní divadla a od roku 2005 působí jako společný subjekt s jedním souborem herců působícím na dvou scénách. Každodenní chod divadla zajišťují tajemníci a režiséři, kteří organizují herce v době zkoušení nových představení. Nová představení se zkouší vždy čtyřikrát do roka na premiéry v listopadu, březnu a červnu.

Fermany se dělí na měsíční, týdenní a denní. Přičemž měsíční musí být souboru rozeslán 2 měsíce předem, nejpozději do 15. dne měsíce, například do 15.11. musí být zaslán ferman na leden. Týdenní je pak posílán vždy ve čtvrtek do 14:00 a denní do 12:00 předcházejícího dne. Do fermanů se zapisují jak jednotlivá představení, která se ten den hrají, tak i například zájezdová představení, pronájmy, zkoušky na nová představení, tiskové konference, veřejná vystoupení, focení, apod.

S herci všechny záležitosti vyřizuje tajemník, který na základě domluvy s herci sestaví fermany. Ty pak rozešle ostatním zaměstnancům e-mailem v podobě excelovské tabulky. Všichni zaměstnanci, jako uvaďeči, kostymérky, maskérky, zvukaři, osvětlovači nebo technici mají každý své oddělení a vedoucího, který s nimi domlouvá směny.

1.1.2.1 Herci

Herecké obsazení se dělí na 3 skupiny podle závazku vůči divadlům:

1. Herci, kteří mají **stálé angažmá** – Ti musí být vždycky k dispozici a přizpůsobit se rozpisům. Také nemohou hostovat v ostatních divadlech.
2. Herci **s prioritou** – Také se musí přizpůsobovat rozpisům, ale na rozdíl od herců z 1. skupiny mohou hostovat i v jiných divadlech. Divadlo ABC a Rokoko však mají přednost před ostatními divadly, která se musí domluvit s tajemnicí, zda mohou herci v jiných divadlech hrát.
3. **Hosté** – Co se tvorby rozpisů týče, jsou hosté ti nejdůležitější a domlouvají se jako první. Na základě toho, kdy mohou a kdy naopak nemohou se totiž určují termíny představení, kterým se pak ostatní zaměstnanci musí přizpůsobit.

Herci vždy 14 dní před premiérou nesmí hostovat v jiných divadlech a musí chodit na zkoušky. Ty se dělí na hlavní zkoušky, generální zkoušky, technické svícené zkoušky a kostýmové zkoušky.

1.1.2.2 Režisér

Režisér určuje termíny zkoušek, kterým se všichni zaměstnanci musí přizpůsobit.

1.1.2.3 Tajemník

Přes tajemníka jde veškerá domluva, která se týká tvorby fermanů. Domlouvá se se všemi jak ohledně termínů zkoušek, které mu předá režisér, tak i s hosty, kteří mu nahlásí

termíny, kdy nemohou hrát. Při tvorbě fermanů musí brát také v potaz, že má přednost divadlo ABC, které má větší kapacitu sálu.

1.1.2.4 Uvaděči

Domluva uvaděčů se v obou divadlech liší. V divadle Rokoko se směny domlouvají na měsíc dopředu a v divadle ABC vždy na týden dopředu přes Google dokument.

1.1.3 Divadlo Comica Economica

Divadlo Comica Economica je amatérské divadlo, které založil vysokoškolský pedagog Doc. RNDr. Bohumír Štědroň v roce 2010. Divadlo má vlastního organizátora a PR. Je složené převážně ze studentů a dobrovolníků, kteří se scházejí každý týden v učebně, aby zkoušeli na nová představení. Představení hrají dvakrát až třikrát do roka v jiných divadlech nebo na školách. Domluva mezi členy spolku probíhá přes e-mail. Rozpisy a aktuální alternace a herecké obsazení či aktuální představení nejsou nikde uvedeny.

1.2 Průzkum existujících systémů

1.2.1 Theatron

Theatron je software specializovaný na plánování produkce divadel. Konkrétně zajišťuje přehlednost informací v rámci vnitřního fungování divadla, jejich sdílení, správu rozpisů a responzivitu aplikace.

1.2.2 Zákaznický portál Vema

Vema portál je systém poskytující zaměstnancům přístup k těm informacím, které jsou pro jeho pozici potřebné. Data lze vybírat, třídit, filtrovat či uspořádat dle požadovaných kritérií. Součástí systému jsou i výplatní listky.

1.2.3 Portál Filmové akademie Miroslava Ondříčka v Písku

Tento portál slouží k přehlednosti informací pro studenty, konkrétně k zobrazování rozvrhů studentů, informací o nich, seznamu studentů, učitelů a předmětů a slouží také jako „nástěnka“ pro zveřejňování nejnovějších informací týkajících se studia.

1.3 Specifikace požadavků

Specifikace požadavků je dokument, který musí být odsouhlasený uživateli na základě předchozí analýzy. [1] Jsou zde specifikovány funkční požadavky, požadavky nefunkční, neboli kvalitativní a požadavky na zavedení řešení.

Neformálně se jednotlivé požadavky značí buď jako „Must have“, zahrnující funkce, které v systému musí být, anebo „Nice to have“, které jsou pro systém spíše doplňkové a nemají vliv na datovou strukturu systému. Pro tuto prioritizaci požadavků budu používat klíčová slova MUST a SHOULD. [2]

1.3.1 Funkční požadavky

Funkční požadavky definují všechny funkce a možnosti, které bude systém uživatelům umožňovat.

- **FR01 – Přihlašování do systému** – Uživatelé se budou přihlašovat do systému pomocí uživatelského jména či e-mailu a hesla. [MUST]
- **FR02 – Zobrazení rozpisu směn na aktuální týden** – Uživateli se po přihlášení zobrazí rozpis jeho směn na aktuální týden. [MUST]
- **FR03 – Zobrazení rozpisu směn na aktuální i následující měsíc** – Uživatel si může zvolit, zda chce zobrazit rozpis směn na celý aktuální či následující měsíc. [MUST]
- **FR04 – Zobrazení směn ostatních uživatelů** – Uživatel si může zobrazit směny ostatních uživatelů. [SHOULD]
- **FR05 – Správa rozpisů služeb** – Oprávnění uživatelé mohou v systému vytvářet nové rozpisy, upravovat stávající či mazat ty staré. [MUST]
- **FR05 – Historie směn** – Uživatel může v systému dohledat i směny z minulých měsíců. [SHOULD]
- **FR06 – Zobrazení informací o představeních a o zaměstnancích** – Všichni uživatelé mají přístup k zobrazení informací o představeních a někteří oprávnění uživatelé mají přístup i k informacím o zaměstnancích. [MUST]
- **FR07 – Přidání a odebrání uživatele** – Uživatelé s oprávněním mohou přidávat nové zaměstnance a odebírat ty, kteří už v divadle nepracují. [MUST]
- **FR08 – Správa profilu uživatele** – Uživatel má možnost upravovat svůj vlastní profil a oprávnění uživatelé i profily ostatních uživatelů. [SHOULD]
- **FR09 – Zobrazení směn ve formě kalendáře** – Směny se na úvodní stránce po přihlášení zobrazí ve formě kalendáře, který si může uživatel přizpůsobit podle svého. [SHOULD]
- **FR10 – Možnost automatického sčítání směn** – Na konci každého měsíce se provádí sčítání směn a jejich následné proplácení. Systém tedy poskytne funkci, která tyto směny automaticky spočte. [SHOULD]
- **FR11 – Správa nástěnky se vzkazy** – Uživatel má možnost přidávat, upravovat a mazat vlastní příspěvky na nástěnce a zobrazovat vzkazy ostatních. [SHOULD]
- **FR12 – Odhlášení uživatele** – Uživatel se může ze systému odhlásit. [MUST]

1.3.2 Nefunkční požadavky

Nefunkční požadavky, neboli kvalitativní, jsou jakési doplňky k funkčním požadavkům. Týkají se například výkonu, délky odezvy, dostupnosti, spolehlivosti či bezpečnosti. [1]

- **NFR01 – Dostupnost přes webový prohlížeč** – Systém jako webová aplikace bude dostupný přes webový prohlížeč, tzv. tenkého klienta, ze serveru. [MUST]
- **NFR02 – Responzivita** – Aplikace bude responzivní. [MUST]
- **NFR03 – Zabezpečení systému** – Systém bude zabezpečený a všechny chyby budou správně odchyceny, tj. nedostanou se k uživateli. [MUST]
- **NFR04 – Zajištění práv uživatelů** – K určitým funkcionalitám systému budou mít přístup pouze oprávnění uživatelé. [MUST]
- **NFR05 – Vnitřní komunikace** – Komunikace mezi frontendem a backendem probíhá pomocí RESTful API. [MUST]
- **NFR06 – Technologie** – Aplikace je psána v jazycích Java (Enterprise Edition), JavaScript, HTML5, SQL a kaskádových stylů CSS3. – Konkrétně PostgreSQL, JPA, Hibernate, Maven a frameworky jQuery, Angular 2 (React?), Spring (Security, Boot), Bootstrap, Node.js, kalendář. [MUST]

1.3.3 Požadavky na zavedení řešení

Požadavky na zavedení řešení jsou „dočasné“ požadavky, které po zavedení řešení ztrácejí smysl. Typicky jsou to požadavky na tvorbu uživatelské či administrátorské dokumentace, školení klíčových a ostatních uživatelů, požadavky na migraci dat, přípravu náhradních řešení, apod. [3]

- **TR01 – Vytvoření uživatelské příručky** – Je třeba vytvořit uživatelskou příručku, která bude obsahovat návod, jak s aplikací pracovat. [SHOULD]
- **TR02 – Příprava náhradního řešení** – Do uživatelské příručky budou přidány postupy, jak se zachovat při neočekávaném chování aplikace. [SHOULD]
- **TR03 – Zaškolení klíčových uživatelů** – Bude nutné vysvětlit klíčovým uživatelům, jak mají v aplikaci pracovat, aby mohli vykonávat svou práci. [MUST]
- **TR04 – Uložení dat do databáze aplikace** – Před uvedením aplikace do chodu budou hromadně přidána data do databáze. Tato data se týkají informací o představeních, zaměstnancích, apod. [MUST]
- **TR05 – Instalace aplikace na server** – Aplikace musí být instalována na server divadla, aby byla přístupná všem uživatelům. [MUST]

1.4 Případy užití

Případ užití (anglicky use case, zkráceně UC) popisuje chování systému z pohledu uživatele za různých situací ve formě reakcí systému na funkční požadavky jednoho z uživatelů systému, nazývaného primární aktér. Tento aktér vyvolává v rámci případu užití interakci se systémem za účelem dosažení určitého cíle. V závislosti na požadavcích a podmínkách u původního požadavku se mohou objevit odlišné způsoby chování, neboli scénáře. Případy užití sdružují tyto rozdílné scénáře dohromady v jeden celek. [4]Error! Reference source not found.Error! Reference source not found.

1.4.1 Definice uživatelských rolí

Všichni uživatelé musí být do systému přihlášení.

- **Administrátor** – Má přístup ke všem funkcionalitám systému, včetně zdrojových kódů a databáze.
- **Jiří Suchý** – Má přístup ke všem informacím, které ale nemůže je měnit.
- **Kancelář** – Zaměstnanci kanceláře mají přístup ke všem informacím v systému, které mohou měnit.
- **Tajemník** – Má přístup ke všem informacím, ale může pouze tvořit, upravovat a mazat fermany. Dále také sčítat směny.
- **Herci, muzikanti** – Mohou zobrazit jejich vlastní směny a informace o ostatních hercích či muzikantech, a mohou zadávat datumy, kdy nemohou přijít. Zde však budou i nějaké výjimky.
- **Hlavní uvaděčka** – Může upravovat směny ostatních uvaděček kdykoliv.
- **Uvaděčky** – Mohou upravovat směny pouze před začátkem měsíce.
- **Zvukaři, osvětlovači, technici, kostymérky** – Mohou upravovat vlastní směny kdykoliv.

1.4.2 Jednotlivé případy užití

Jednotlivé případy užití jsou rozděleny do skupin podle výše uvedených uživatelských rolí. Je zde navíc skupina Společné, kde jsou kvůli zjednodušení UC diagramu uvedeny všechny společné případy užití. Skupina Administrátor je vyřazena úplně, aby nebyly informace duplikovány. Každý případ užití obsahuje jednoduchý popis a jeho hlavní a alternativní scénář užití.

1.4.2.1 Společné

- UC1 – Zobrazení vlastního rozpisu směn
- UC2 – Úprava a zobrazení detailních informací ve vlastním profilu
- UC3 – Zobrazení základních informací v profilu jiného zaměstnance
- UC4 – Zobrazení kalendáře a fermanů
- UC3 – Vyhledání zaměstnance
- UC5 – Vyhledání představení
- UC6 – Zobrazení informací o představení
- UC7 – Přihlášení a odhlášení

- UC8 – Přidání příspěvku na nástěnku

1.4.2.2 Kancelář

- UC9 – Správa všech informací v profilu jiného zaměstnance
- UC10 – Správa představení
- UC11 – Správa kalendáře
- UC12 – Odstranění příspěvku z nástěnky
- UC13 – Přidání nového zaměstnance
- UC14 – Odstranění zaměstnance

1.4.2.3 Tajemník

- UC15 - Správa fermanů
- UC16 – Sečtení směn zaměstnanců

1.4.2.4 Herec a muzikant

- UC17 – Přidání data, kdy nemohou

1.4.2.5 Hlavní uvaděčka

- UC18 – Úprava směn ostatních uvaděček
- UC19 – Sečtení směn uvaděček

1.4.2.6 Uvaděčka

- UC20 – Správa vlastních směn před začátkem měsíce

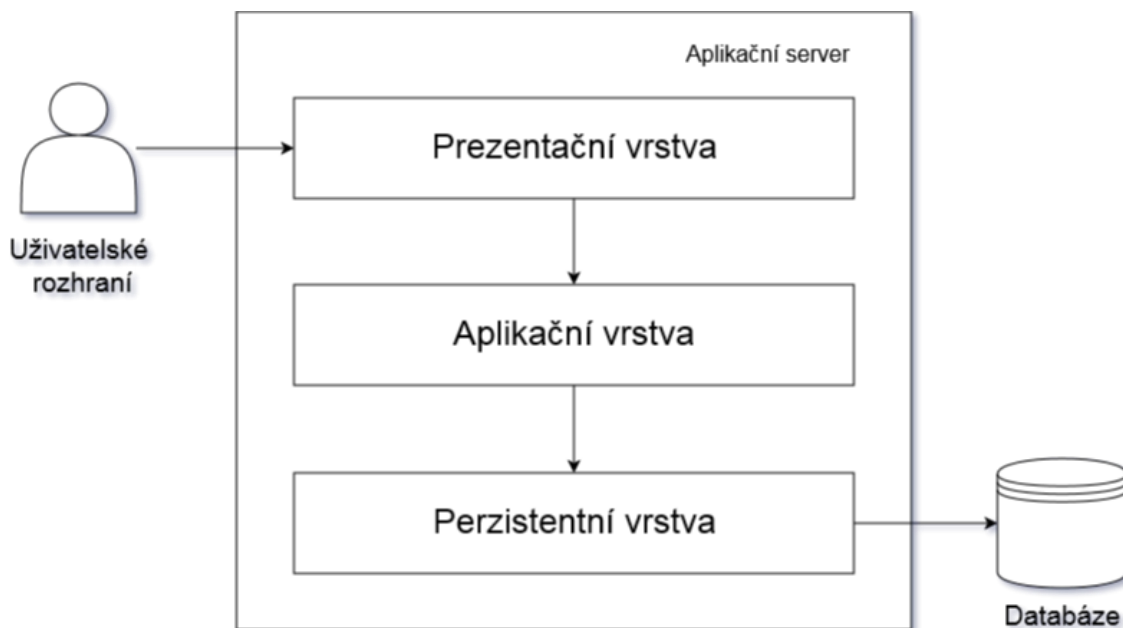
1.4.2.7 Zvukař, osvětlovač, technik, kostymérka

- UC21 – Správa vlastních směn

Návrh systému

2.1 Architektura systému

Pro vývoj aplikace použijeme architekturu vícevrstvou (viz. Obrázek 1).



Obrázek 1 - Vícevrstvá architektura aplikace

2.1.1 Prezentační vrstva

Uživatel odešle požadavek přes uživatelské rozhraní aplikačnímu serveru. Na základě požadavku aplikace přistoupí ke kontrolerům na úrovni prezentační vrstvy, které dále přistupují ke službám (services) na úrovni vrstvy aplikační. Před zavoláním služeb dochází také k autentizaci a autorizaci uživatele.

2.1.2 Aplikační vrstva

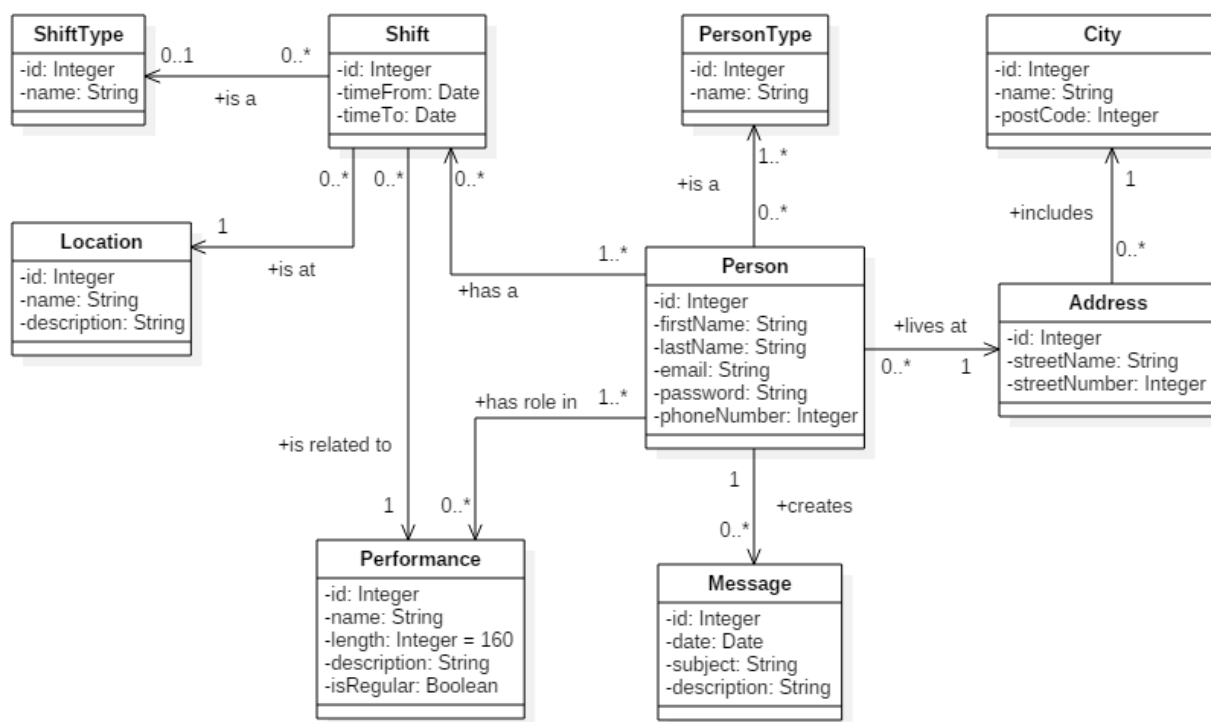
Služby zajišťují přes rozhraní přístup k datům perzistentní vrstvy a většinu aplikační logiky. Zároveň zde dochází například k ověřování hesla či existence uživatele.

2.1.3 Perzistentní vrstva

Tato vrstva nabízí přístup k úložišti dat přes tzv. DAO (Data Access Object), typicky za využití nějaké relační databáze (RDBMS) či objektově relační databáze (ORDBMS) za účelem zapouzdření veškerého přístupu ke zdroji dat.

2.2 Datová struktura

Pro převod datového modelu na logické schéma databáze byl vytvořen ER diagram (viz. příloha A). Ke znázornění jednotlivých entit a vztahů mezi nimi byl vytvořen doménový model tříd (viz. Obrázek 2).



Obrázek 2 - Doménový model tříd

2.2.1 Entity

Entita reprezentuje určitou část objektů reálného světa. Jedná se jak o objekty fyzické (například člověk) nebo abstraktní (například kategorie). Každá entita je popsána svým názvem a sadou atributů. Každý z těchto atributů pak má přiřazen datový typ. [5]

2.2.1.1 Person – Osoba

Entita Person reprezentuje zaměstnance divadla.

Atributy:

- **firstName** – křestní jméno zaměstnance,
- **lastName** – příjmení zaměstnance,
- **email** – e-mail zaměstnance,
- **password** – heslo k přihlášení zaměstnance,
- **phoneNumber** – telefonní číslo zaměstnance.

2. Návrh systému

2.2.1.2 PersonType – Typ osoby

Každý zaměstnanec má svůj typ, tj. zaměření podle zaměstnání. Například herec/čka, muzikát/ka, technik, kostymérka, uvaděčka, atd.

Atributy:

- **name** – název typu osoby.

2.2.1.3 Address - Adresa

Každý zaměstnanec má přiřazenou adresu bydliště.

Atributy:

- **streetName** - název ulice, kde zaměstnanec bydlí,
- **streetNumber** – číslo popisné.

2.2.1.4 City – Město

Ke každé adrese je přiřazeno město.

Atributy:

- **name** – název města,
- **postCode** – poštovní směrovací číslo města.

2.2.1.5 Message – Vzkaz na nástěnce

Každý zaměstnanec může přidávat vzkazy na společnou nástěnku.

Atributy:

- **date** - datum přidání vzkazu na nástěnku,
- **subject** – předmět vzkazu (nepovinný),
- **description** - obsah vzkazu.

2.2.1.6 Shift – Směna

Každému zaměstnanci může být přiřazena směna. Směny jsou vypisovány každý měsíc.

Atributy:

- **timeFrom** – čas, od kdy směna začíná,
- **timeTo (do)** – čas, kdy směna končí. Tento údaj je nepovinný, pokud se jedná o pravidelné představení, u kterého je čas konce přepočítán podle délky představení.

2.2.1.7 ShiftType – Typ směny

Směny rozlišujeme podle typu, ale pouze u pravidelných představení. Typy jsou: premiéra, veřejná generálka, zkouška nebo zájezd.

Atributy:

- **name** - název typu směny.

2.2.1.8 Location – Místo konání

Každé představení (resp. směna) má své umístění. Výchozí umístění je v prostorách divadla SEMAFOR. Pokud se jedná o zájezd, umístění je možné změnit.

Atributy:

- **name** - název místa, kde se představení odehrává,
- **description** - popis místa.

2.2.1.9 Performance – Představení

Tato entita reprezentuje představení.

Atributy:

- **name** – název představení,
- **length** – délka představení (výchozí je 160 minut),
- **description** – popis představení (nepovinný),
- **isRegular** – určuje, zda se jedná o představení divadla SEMAFOR či pronájem.

2.2.2 Vazby mezi entitami

Vazba mezi entitami představuje logický vztah mezi entitami. Na vazbu můžeme pohlížet jako na dvě vazby v opačných směrech. V tomto smyslu se hovoří o takzvaných rolích, které představují pohled na danou vazbu ve směru od jedné entity k druhé. Ke každé roli přiřazujeme tzv. kardinalitu. Ta představuje omezení v počtu instancí druhé entity, které mají vztah s jakoukoliv instancí první entity. [5]

2.2.2.1 Is a (Person, PersonType)

Zaměstnanec má minimálně jedno zaměření, tj. může být například herečka a zároveň uvaďčka.

2.2.2.2 Lives at (Person, Address)

Každému zaměstnanci přísluší pouze jedna adresa bydliště.

2.2.2.3 Includes (Address, City)

Každá adresa leží v jednom městě.

2.2.2.4 Creates (Person, Message)

Každý zaměstnanec může a nemusí vytvářet vzkazy na zdi.

2. Návrh systému

2.2.2.5 Has a (Person, Shift)

Každému zaměstnanci se přiřazují směny. Může se stát, že daný měsíc nemá směnu žádnou.

2.2.2.6 Has role in (Person, Performance)

Každý zaměstnanec může mít v představení nějakou roli.

2.2.2.7 Is a (Shift, ShiftType)

Pokud se jedná o semaforové představení, můžeme představení/směně přiřadit speciální typ (premiéra, veřejná generálka, zkouška, zájezd).

2.2.2.8 Is at (Shift, Location)

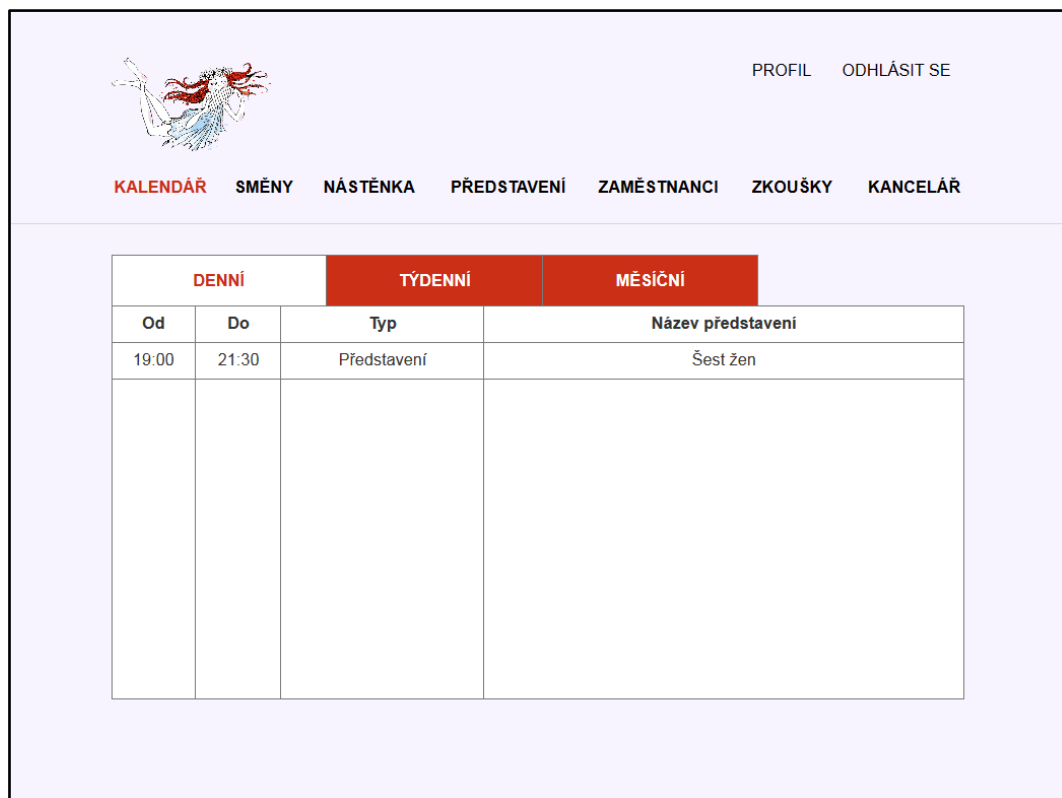
Každé představení má své umístění.

2.2.2.9 Is related to (Shift, Performance)

Každá směna je přiřazena pouze k jednomu představení.

2.3 Návrh grafického rozhraní

Při návrhu designu portálu je potřeba dbát na jednoduchost a funkčnost. Zvolila jsem tedy jednoduchý vzhled a přehledné menu v horní části stránky (viz.



Obrázek 3).

Obrázek 3 – Návrh hlavní stránky aplikace

2.3.1 Seznam obrazovek

Navigace v portálu slouží k dynamické změně obsahu stránek, které obsahují následující obrazovky.

2.3.1.1 Kalendář

- Denní rozpis
- Týdenní rozpis
- Měsíční rozpis
- Formulář pro přidání zvláštní akce

2.3.1.2 Směny

- Rozpis směn (vlastních i cizích)
- Formulář pro přidání/změnu směny podle typu

2.3.1.3 Nástěnka

- Dashboard příspěvků
- Formulář pro přidání/úpravu příspěvku

2.3.1.4 Představení

- Seznam představení
- Formulář pro přidání/úpravu představení
- Detail představení

2.3.1.5 Zaměstnanci

- Seznam zaměstnanců
- Vyhledávání zaměstnanců
- Přidání/úprava zaměstnance
- Zobrazení profilu zaměstnance

2.3.1.6 Zkoušky

- Rozpis vypsanych zkoušek
- Přidání/úprava zkoušky
- Zobrazení detailu zkoušky

2.3.1.7 Kancelář

- Seznam zaměstnanců kanceláře a jejich kontakty

2.3.1.8 Přihlašovací stránka

- Formulář pro přihlášení uživatele

2.3.1.9 Profil

- Seznam údajů o uživateli

Implementace

3.1 Použité nástroje a technologie

Na základě předchozí analýzy jsem zvolila následující technologie.

3.1.1 Java 8

Java se řadí mezi objektově orientované programovací jazyky. Zdrojový kód je nejprve zapsán do souborů končící na `.java` a poté jsou tyto soubory kompilovány do `.class` přes `javac` kompilátor. Tyto soubory pak obsahují tzv. bytecode, což je jazyk Java Virtual Machine (JVM). Příkaz `java` poté spustí celou aplikaci. Tento formát je nezávislý na architektuře počítače a může tedy běžet na libovolném zařízení, které má JVM [6]. Javu tvoří platformy jako Java ME (Java Mobile Edition) pro mobilní aplikace, Java SE (Java Standard Edition) pro desktopové aplikace či Java EE (Java Enterprise Edition) pro rozsáhlé distribuované systémy.

Java 8 s sebou přináší různé další funkčnosti, jako například Lambda výrazy, Date a Time API, Nashorn JavaScript engine, atd. [7].

3.1.2 Java EE 8

Java EE je součástí platformy Java rozšiřující Java SE o specifikace pro distribuované systémy a webové služby. Java EE API zahrnují technologie jako například Enterprise JavaBeans (EJB), JavaServer Pages (JSP), Java Server Faces (JSF), Servlet, Java Persistence API (JPA), atd. [8].

3.1.3 Maven

Apache Maven je systém pro správu projektů v jazyce Java [9]. Maven spravuje build projektu a jejich závislosti (dependencies) a dynamicky stahuje Java knihovny a Maven pluginy z repozitářů. Ty pak ukládá do lokální cache paměti. Pluginy zajišťují například kompilaci projektů do výsledných balíčků (jar, war), nasazení na servery či testování. Veškeré konfigurace a závislosti projektu poskytuje POM (Project Object Model) v souboru `pom.xml`. V tomto souboru jsou informace o projektu, jeho podprojektech či samotné závislosti.

3.1.4 Tomcat

Apache Tomcat, označován jako Tomcat Server implementuje JavaEE specifikace, jako Java Servlet, JavaServer Pages, Java Expression Language a WebSocket technologie [10].

3.1.5 PostgreSQL

PostgreSQL je objektově-relační databázový systém (ORDBMS). Běží na všech hlavních operačních systémech a například pro Javu poskytuje rozhraní JDBC [11].

3.1.6 Spring

Spring je framework pro platformu Java. Obsahuje také rozšíření pro webové aplikace stavěné na Java EE. Stal se populární především kvůli doplnění či nahrazení Enterprise JavaBean (EJB).

3.1.7 Node.js

Node.js je platformně nezávislé JavaScriptové prostředí, které umožňuje spouštět kód psaný v JavaScript na straně serveru. Skripty spouští k vytvoření dynamických webových stránek ještě předtím, než je stránka odeslána webovému prohlížeči uživatele [12].

3.1.8 ReactJS

React je JavaScript knihovna určena pro vývoj uživatelských rozhraní. React nabízí elementy, které reprezentují standardní HTML elementy (např. `div`, `img`) nebo vývojáři umožňuje vytvářet vlastní. Renderováním HTML elementů pomocí JavaScript vytvoří React virtuální reprezentaci HTML v paměti, neboli Virtual DOM (Document Object Model), což znamená, že React nejprve renderuje HTML strom virtuálně a pak pokaždé při změně stavu (state) místo přepsání celého stromu, React přepíše pouze odlišnosti mezi starým a nově vytvořeným stromem. U React rozlišujeme vlastnosti (props) a stavy (states). Vlastnosti komponenta obdrží jako parametr k renderování a stavy udržuje a předává je svým potomkům jako vlastnosti [13].

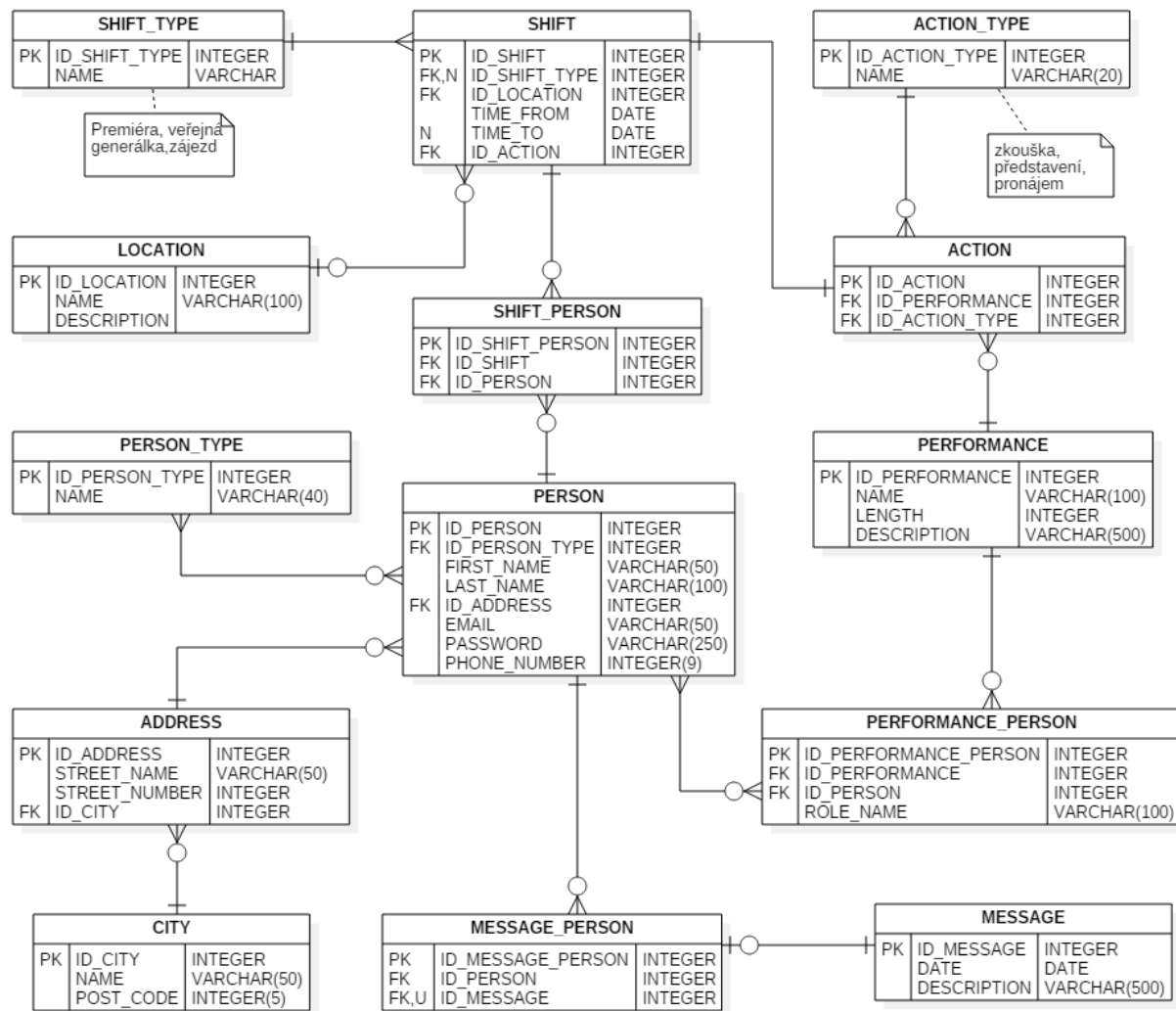
Závěr

Portál byl po předchozí analýze částečně implementován a manuálně otestován. Nejprve jsem definovala požadavky a případy užití a vytvořila datový model aplikace. Implementovány byly zatím pouze základní funkčnosti jako je správa směn a zaměstnanců, takže většina požadavků není splněna. Backend aplikace byl implementován v jazyce Java za použití systému Maven a Spring frameworku. Komunikace s PostgreSQL databází probíhala přes rozhraní JDBC a JPA. Frontend aplikace byl implementován v JavaScript s využitím Node.js a knihovny React. Aplikace byla vyvíjena v IntelliJ IDEA a spuštěna na aplikačním serveru Tomcat.

Literatura

- [1] GOMAA, Hassan. *Software modeling and design: UML, use cases, patterns, and software architectures*. Cambridge [U.K.]: Cambridge University Press, 2011. ISBN 978-0-521-76414-8 [Cit. 15.10.2017].
- [2] RADNER, S. *Key words for use in RFCs to Indicate Requirement Levels* [online]. [Cit.15.10.2017] Dostupné z: <https://www.ietf.org/rfc/rfc2119.txt>
- [3] KOMÁREK, Martin. *Analýza a dokumentace požadavků* [přednáška]. Praha, ČVUT Fakulta elektrotechnická, březen 2016. In: moodle.fel.cvut.cz [online]. [Cit.18.10.2017]. Přednáška dostupná z: https://moodle.fel.cvut.cz/pluginfile.php/42038/mod_resource/content/3/ANALYZA_A_DOKUMENTACE_POZADAVKU_3PREDNASKA.pdf
- [4] COCKBURN, Alistair. *Use Cases: jak efektivně modelovat aplikace*. Brno: Computer Press, 2005. ISBN 80-251-0721-3 [Cit. 18.10.2017].
- [5] HAJN. *Úvod do datového modelování* [online]. [Cit. 21.11.2017]. Přednáška dostupná z: www.fi.muni.cz/~hajn/vyuka/DB%20systemy%20a%20aplikace/%davod_datov%e9_modelov%eln%ed.doc
- [6] ORACLE, *About the Java Technology* [online]. [Cit.11.1.2018] Dostupné z: <https://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html>
- [7] ORACLE, *Java 8 informations* [online]. [Cit. 7.1.2018] Dostupné z: <https://www.java.com/en/download/faq/java8.xml>
- [8] ORACLE, *Overview of Enterprise Applications* [online]. [Cit.11.1.2018] Dostupné z: <https://docs.oracle.com/javaee/7/firstcup/java-ee001.htm>
- [9] SONATYPE. *Maven: The Definitive Guide*. Sebastopol, USA: O'Reilly Media, 2008. [Cit. 7.1.2018]
- [10] APACHE, *Apache Tomcat* [online]. [Cit.7.1.2018] Dostupné z: <http://tomcat.apache.org/index.html>
- [11] PostgreSQL [online]. [Cit.7.1.2018] Dostupné z: <https://www.postgresql.org/about>
- [12] MDN web docs, *Node.js* [online]. [Cit.11.1.2018] Dostupné z: <https://developer.mozilla.org/en-US/docs/Learn/Server-side>
- [13] LEDVINKA, Martin. *React* [přednáška]. Praha, ČVUT Fakulta elektrotechnická, 2016. In: cw.fel.cvut.cz [online]. [Cit. 7.1.2018]. Přednáška dostupná z: http://cw.fel.cvut.cz/wiki/_media/courses/b6b33ear/lecture-07-react-s.pdf

Entitně-relační diagram



Obrázek 4 – ER diagram