

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra kybernetiky

System pro správu divadla

Miroslav Falcmann

Vedoucí: Ing. Božena Mannová Ph.D.

Studijní program: Otevřená informatika

Specializace: Základy umělé inteligence a počítačových věd

Květen 2023

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Falcmann** Jméno: **Miroslav** Osobní číslo: **499266**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra kybernetiky**
Studijní program: **Otevřená informatika**
Specializace: **Základy umělé inteligence a počítačových věd**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Systém pro správu divadla

Název bakalářské práce anglicky:

Theater Management System

Pokyny pro vypracování:

1. Seznamte se s problematikou správy divadla.
2. Provedte analýzu vám dostupných konkrétních řešení v několika divadlech a proveďte jejich porovnání a vyhodnocení. Seznamte se i s bakalářskými projekty zpracovanými na podobná témata.
3. Na základě provedené analýzy navrhnete základní funkcionality aplikace, která by měla nahradit stávající „ruční“ vedení agendy divadla, včetně vytváření pánu představení a vytížení herců.
4. Při návrhu se zaměřte na využití AI při plánování představení a sestavování fermanů.
5. Zvolte architekturu aplikace a technologie pro implementaci.
6. Aplikaci implementujte a otestujte.
7. Zhodnoťte výsledky a navrhnete případné další funkcionality nebo jiná zlepšení.
8. Při řešení využijte vhodných prostředků softwarového inženýrství.

Seznam doporučené literatury:

- [1] Roger S. Pressmann Bruce Maxim: Software Engineering: A Practitioner's Approach , ISBN-10: 9780078022128
- [2] Bakalářská práce Anna Zlámalová <https://dspace.cvut.cz/handle/10467/79316>
- [3] <https://www.cinicloud.com/>
- [4] <https://www.theatron.eu/>
- [5] <https://eduoorbit.co.in/edutime.html>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Božena Mannová, Ph.D. kabinet výuky informatiky FEL

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

Datum zadání bakalářské práce: **13.02.2023**

Termín odevzdání bakalářské práce: **26.05.2023**

Platnost zadání bakalářské práce: **22.09.2024**

Ing. Božena Mannová, Ph.D.
podpis vedoucí(ho) práce

prof. Ing. Tomáš Svoboda, Ph.D.
podpis vedoucí(ho) ústavu/katedry

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

III. PŘEVZETÍ ZADÁNÍ

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

Datum převzetí zadání

Podpis studenta

Poděkování

Chtěl bych v první řadě poděkovat paní Ing. Boženě Mannové Ph.D., která byla mou vedoucí a ochotně mi na pravidelných konzultacích dávala cenné rady a pomohla mi práci dovést až do konce. Dále bych rád poděkoval Kateřině Vorudové a Radce Hukové, které si na mě našly čas a poskytly mi cenné informace týkající se správy umělecké činnosti v divadle. Nakonec bych rozhodně rád poděkoval své rodině, přítelkyni a kamarádům za podporu v průběhu studia i tvorby této bakalářské práce.

Prohlášení

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

V Praze, 10. května 2023

.....
Miroslav Falcmann

Abstrakt

Tato bakalářská práce se zabývá vytvořením systému pro správu divadla, který by usnadnil stávající problematiku domluvy s herci, tvorby fermanu či zápisu zaměstnanců na směny. Současně se zaměřuje na možné využití umělé inteligence (dále jen AI) při plánování představení.

Na základě provedeného průzkumu byly navrženy funkční i nefunkční požadavky, podle kterých byl systém naimplementován. Nezávisle na implementaci byly také navrženy možné strategie využívající AI pro automatické plánování představení.

Výsledný systém byl otestován, je uživatelsky přívětivý a splňuje všechny nezbytné požadavky pro nasazení do provozu reálných divadel.

Klíčová slova: divadlo, správa, Python, Django, AI

Vedoucí: Ing. Božena Mannová Ph.D.

Abstract

This bachelor thesis deals with the creation of a theater management system that would partially eliminate the existing problems connected with the arrangement with actors, creating monthly plans or assigning employees to shifts. It also focuses on the possible use of artificial intelligence (hereafter AI) in performance planning.

Based on the research conducted, functional and non-functional requirements were proposed, according to which the system was implemented. Independent of the implementation, possible strategies using AI for automatic performance planning were also proposed.

The resulting system has been tested, is user-friendly and meets all the necessary requirements for deployment in real theaters.

Keywords: theatre, management, Python, Django, AI

Title translation: Theater Management System

Obsah

1 Úvod	1	4.3.7 Představení	27
1.1 Motivace	1	4.3.8 Směny	28
1.2 Popis problému	1	4.3.9 Indispozice	31
1.3 Cíl práce	1	4.3.10 Fermany	34
2 Analýza řešení	3	4.4 Databáze	37
2.1 Průzkum organizace v divadlech	3	4.4.1 Doménový model	37
2.1.1 Domluva s herci	3	4.4.2 Entity a vazby mezi nimi	38
2.1.2 Tvorba fermanů	4	5 Implementace	41
2.1.3 Domluva s UTP	7	5.1 Backend	41
2.1.4 Domluva s uvaděči	7	5.1.1 Použité nástroje a technologie	41
2.1.5 Závěr průzkumu divadel	8	5.1.2 Databáze	43
2.2 Průzkum existujících řešení	8	5.1.3 Logika a komunikace s databází	43
2.2.1 Propared	8	5.2 Frontend	45
2.2.2 Theatron	9	5.2.1 Použité nástroje a technologie	45
2.2.3 Theasoft	10	5.2.2 Uživatelské rozhraní	46
2.2.4 Závěr průzkumu existujících řešení	10	6 Testování	51
3 Využití umělé inteligence	11	6.1 Automatizované testování	51
3.1 Motivace a hlavní přínos AI	11	6.2 Uživatelské testování	52
3.2 Specifikace problému	11	6.2.1 Inspektorka hlediště	53
3.3 Návrhy řešení	12	6.2.2 Uvaděčka, Barmanka	53
3.3.1 Příprava dat	12	6.2.3 Produkční divadla	54
3.3.2 1. metoda, využití genetického algoritmu	12	7 Závěr	55
3.3.3 2. metoda, využití horolezeckého algoritmu	16	7.1 Budoucí vývoj aplikace	55
4 Návrh vlastností systému	19	7.1.1 Automatické sestavení fermanu	55
4.1 Funkční požadavky	19	7.1.2 Tvorba týdenních fermanů	56
4.1.1 Přístup do systému	19	7.1.3 Exportování fermanu jako PDF či XML soubor	56
4.1.2 Uživatelský profil	19	7.1.4 Potvrzovací okna pro požadavky smazání	56
4.1.3 Domovská stránka	19	7.1.5 Filtrování zaměstnanců	56
4.1.4 Zaměstnanci	20	7.1.6 Poznámka pro indispozici	56
4.1.5 Představení	20	7.1.7 Kontrola fermanu po přidání nové indispozice	56
4.1.6 Směny	21	7.1.8 Omezení náhledu nedokončených fermanů	56
4.1.7 Indispozice	21	Literatura	57
4.1.8 Fermany	22	A Obsah příložených souborů	61
4.2 Nefunkční požadavky	22	A.1 Zdrojový kód	61
4.3 Případy užití	23		
4.3.1 Rozdělení uživatelských rolí a práv	23		
4.3.2 Jednotlivé případy užití	24		
4.3.3 Přístup do systému	24		
4.3.4 Uživatelský profil	24		
4.3.5 Domovská stránka	25		
4.3.6 Zaměstnanci	26		

Kapitola 1

Úvod

1.1 Motivace

Hlavní motivací ke zpracování tématu této bakalářské práce byly mé pracovní zkušenosti z divadla Minor a vztah k divadlu jako takovému. Z vlastních zkušeností vím, že správa umělecké činnosti může být pro divadla s větším počtem herců velmi náročná. Využil jsem proto svých dosavadních dovedností a vytvořil webovou aplikaci, která by tento proces zjednodušila a zároveň se snažila být k jejím uživatelům co nejpřívětivější, aby bylo snazší její zasazení do praxe.

1.2 Popis problému

Organizace umělecké činnosti v divadle se skládá z více dílčích částí, počínaje domluvou s herci, přes vytvoření fermanu, až po obsazení konkrétních směn obsluhou představení. Při absenci jednotné softwarové aplikace jsou tak obvykle důležité informace pro tuto celkovou správu roz distribuované na více místech a bývají zbytečně komplikace s jejich dohledáním. Současně jsou některé organizační úkoly pro člověka zbytečně náročné, a jsou tak náchylné na jeho chybu. Mělo by tedy smysl vytvořit takové prostředí, které udržuje všechny tyto informace na jednom místě a zároveň nabízí svým uživatelům nástroje, jak usnadnit jejich každodenní činnosti.

1.3 Cíl práce

Cílem práce je navrhnout, implementovat a otestovat webovou aplikaci pro správu divadla. Aplikace má sloužit ke správě směn, vytváření tzv. fermanů, k zobrazení směn uživatelům na základě jejich přístupových práv, úpravu profilů zaměstnanců, vyhledávání informací o směnách a zaměstnancích. Práce se nebude zabývat řízením financí divadla a řízením prodeje lístků. Mimo implementaci aplikace bude proveden průzkum na využití umělé inteligence při plánování představení a sestavování fermanů.

Kapitola 2

Analýza řešení

V rámci analýzy řešení jsou porovnány řešení správy umělecké činnosti ve více různých divadlech a představeny různé již existující webové aplikace, které se tento problém snaží zjednodušit. V závěru kapitoly jsou tyto systémy mezi sebou porovnány a zmíněny jednotlivé klady, kterými se lze inspirovat, a zápory, kterým bude snaha se při návrhu systému vyvarovat.

2.1 Průzkum organizace v divadlech

Sestavení hracího plánu (tzv. fermanu) pro dané období má na starosti produkce divadla či tajemníci nebo tajemnice. S vytvořením fermanu je spjata i domluva s herci, kterou mají rovněž na starost.

Proto jsem se pro záležitosti průzkumu spojil s produkční divadla Minor, paní Kateřinou Vorudovou, a s tajemnicí Divadla V Dlouhé, paní Radkou Hukovou. Paní Vorudová navíc dlouhá léta pracovala v Divadle Komédie či v Národním divadle. Její zkušenosti z těchto divadel byly tedy rovněž zahrnuty do průzkumu. Na závěr bylo využito analýz z předchozích let provedených studenty Annou Zlámalovou [1] a Petrem Jeřábkem [2] v rámci jejich bakalářských prací na stejné téma. Jedná se o divadla: ABC, Rokoko, Semafor, Na Zábradlí.

Výsledky z těchto analýz jsou rozděleny na jednotlivé dílčí části, které popisují hlavní podproblémy celkové správy umělecké činnosti divadla.

2.1.1 Domluva s herci

Nejprve je důležité zmínit, že herce lze podle jejich závazku k danému divadlu rozdělit do dvou skupin:

- **Herci v angažmá:**

Divadlo tyto herce zaměstnává. Jsou tedy součástí jejich divadelního souboru a mají stálý měsíční plat.

■ **Hosté:**

Divadlo tyto herce nezaměstnává. Herci mohou být již zaměstnání v jiném divadle nebo pracují tzv. na volné noze. Dané divadlo je tedy vyplácí podle počtu odehraných představení.

Divadlo se nemusí dotazovat svých herců na to, kdy jsou ochotni hrát, protože herci mají jako zaměstnanci povinnost být pro své divadlo vždy k dispozici. Tedy jediní herci, kteří mají vliv na výslednou podobu fermanu, jsou hosté. Domluva s nimi probíhá obvykle telefonicky, emailem či pomocí sdílených tabulek. Lze ji pak dále rozdělit podle toho, zda je herec v angažmá cizího divadla či není zaměstnán v žádném divadle.

■ **Herci v angažmá cizího divadla:**

Na herce mají plná práva jiná divadla. Domluva tedy musí probíhat s jejich domovskými divadly. V praxi je snaha se takovým hercům zcela vyvarovat.

■ **Herci pracující na volné noze:**

Herci nejsou vázání na žádné divadlo. Tudíž domluva probíhá přímo s daným hercem nebo s jejich agenty. Divadlo je vyzve pro sdělení časů, kdy nejsou k dispozici, a podle nich pak sestaví daný měsíční ferman.

■ **2.1.2 Tvorba fermanů**

Fermany představují výsledek celého procesu plánování. Vydávají se obvykle měsíční, týdenní a v některých divadlech i denní.

■ **Měsíční fermany**

Obsahují seznam všech naplánovaných akcí a představení, včetně jejich nutných zkoušek a oprašovaček.

■ **Týdenní fermany**

Vytvářejí se z hotového měsíčního fermanu a jsou doplněny všemi ostatními potřebnými informacemi pro zaměstnance, jako jsou plánované revize či rozpis jednotlivých osob přítomných na směně.

■ **Denní fermany**

Obsahem informací se nijak neliší od týdenních fermanů. Vytvářejí se pouze pro větší přehlednost.

Pro vytvoření měsíčního fermanu je nejdříve potřeba získat časy, kdy mohou být obsazeni hosté. Podle těch se pak bude daný ferman sestavovat. Divadla, která nemají vlastní soubor herců, proto plánují svá představení klidně i v půlročních blocích, aby se podařilo zajistit, že bude mít daný herec čas. Vytvoří tak kostru daného měsíčního fermanu, kterou pak ještě musí schválit vedení divadla a šéf techniků. Později se ferman ještě upraví o další akce,

2.1. Průzkum organizace v divadlech

kteřé nejsou závislé na hercích ani technicích. Při sestavování fermanu se produkce také často inspiruje loňskými fermany. Týdenní fermany se vytváří vždy týden předem, když už jsou jasné i ty nejmenší detaily. Všechny hotové fermany se pak posílají (většinou emailem) všem ostatním zaměstnancům divadla. Obvykle se měsíční fermany posílají i napříč divadly, aby se provedla vzájemná kontrola, zda-li je vše v pořádku a nevzniká žádná kolize.

minor		ŘÍJEN 2022		
datum	začátek	velký sál	začátek	Malá scéna
1. 10. so	15:00, 18:00	Mia a velryba [20, 21]	10:00	Dobré ráno, Viktore [29]
			10:00 - 12:00	Příběhy novoměstských ulic, outdoorová hra
2. 10. ne	15:00	Sněhurka is not dead [25]	10:00	Zachraňte číslo 6! [30]
	10:00 - 14:00	oprašovačka Sněhurka is not dead	10:00 - 12:00	Petřinský maják, outdoorová hra
3. 10. po	9:30	Sněhurka is not dead [26]	9:30	Zachraňte číslo 6! [31]
	12:00 - 16:00	zk. Tramtárie		
4. 10. út	10:00 - 14:00	zk. Tramtárie		
5. 10. st	10:00 - 14:00	zk. Tramtárie	9:30	Jsem v pohodě... [15]
6. 10. čt	10:00 - 14:00	zk. Tramtárie	9:30	Jsem v pohodě... [16]
7. 10. pá	10:00 - 14:00	zk. Tramtárie	9:30	Koukej, svět! [110]
	18:00	Lipany [116] (Hádr za švece)	16:30	Čteme hrou, Barloutka
8. 10. so	15:00	Demokracie [152]	10:00	Koukej, svět! [111]
			10:00 - 12:00	Vysílač GO84, outdoorová hra
9. 10. ne	15:00	Ztroskotání vzducholodi Italia [33]	10:00, 13:00	Tajemství temné komory, dílna MS
	10:00 - 14:00	oprašovačka Ztroskotání vzducholodi		
10. 10. po	10:00 - 18:00	zk. Tramtárie		
11. 10. út	10:00 - 18:00	zk. Tramtárie	9:30	Koukej, svět! [112]
			18:00	Zpověď Jonatána Papírníka, DRDS, host
12. 10. st	10:00 - 18:00	zk. Tramtárie	9:30	To byl jen vtip! [109]
13. 10. čt	10:00 - 18:00	zk. Tramtárie	9:30	To byl jen vtip! [110]
14. 10. pá	10:00 - 18:00	zk. Tramtárie	9:30	Bruncvík a lev [168]
				(Hádr za Hrovatitsche)
15. 10. so		pronájem divadla (Muzeum Karla Zemana)	10:00	Bruncvík a lev [169]
16. 10. ne	11:00, 15:00	Zá-to-pek! [63, 64]	10:00	Bruncvík a lev (přeloženo na 15. 10.)
			10:00 - 12:00	Republika, outdoorová hra
17. 10. po	čas upřesníme	zk. Tramtárie		
18. 10. út	čas upřesníme	zk. Tramtárie	9:30	Polní žínka Evelínka [71]
19. 10. st	10:00 - 18:00	zk. Tramtárie	9:30	Polní žínka Evelínka [72]
20. 10. čt	10:00 - 18:00	zk. Tramtárie	18:00	Polní žínka Evelínka [73]
21. 10. pá	9:30	Tramtárie - veřejná generálka [1]	18:00	Mamí, už tam budem? [185]
		dílina Staň se divákem	16:30	Čteme hrou, Barloutka
	10:00 - 18:00	generálka Tramtárie		
22. 10. so	10:00	Tramtárie - veřejná generálka [2]	10:00	Mamí, už tam budem? [186]
		(a do 18:00 rezerva na zk.)	10:00 - 12:00	Příběhy novoměstských ulic, outdoorová hra
			14:00, 16:00	Zvuk mi není fuk, dílna MS
23. 10. ne	18:00	Tramtárie - premiéra [3]	10:00 - 12:00	Republika, outdoorová hra
	11:00	Dnes provází Minor (NR)		
24. 10. po	9:30	Dnes provází Minor (NR)	20:00	Listování slaví 20 let!
25. 10. út	9:30	Tramtárie [4]	9:30	Jsem v pohodě... [17]
		dílina Staň se divákem		
26. 10. st	18:00	Tramtárie [5]	9:30, 11:00	O veliké řepě [86, 87]
		podzimní prázdniny		
27. 10. čt	18:00	Sněhurka is not dead [27]	9:30, 11:00	O veliké řepě [88, 89]
		podzimní prázdniny	9:30 - 11:30	Ledová ŠtVánice, outdoorová hra
28. 10. pá	18:00	Záhada hlavolamu [92]	10:00 - 12:00	Republika, outdoorová hra
29. 10. so	15:00	Libozvuky [127]	10:00	Dobré ráno, Viktore [30]
	15:00	Dnes provází Minor (NR)	9:30 - 11:30	Ledová ŠtVánice, outdoorová hra
30. 10. ne	15:00	Bratří naděje [36]	10:00	Zachraňte číslo 6! [32]
	10:00 - 14:00	oprašovačka Bratří naděje		
	15:00	Dnes provází Minor (NR)		
31. 10. po	9:30	Dnes provází Minor (NR)	12:00	Hamletek - začátek zkoušení

31.08.22

Obrázek 2.1: Ukázka měsíčního fermanu divadla Minor

Rozvrh práce od 17. 10. do 23. 10. 2022		Zkouška jeviště	Zač.	Představení MS	Zkouška MS	Ostatní	Ostatní prostory
Datum	Zač.	Představení jeviště	Zač.	Představení MS	Zkouška MS	Ostatní	Ostatní prostory
17. 10. Po		12:00 - 17:00 zkouška Tramtáře vede: [redacted] inspicie: [redacted] světlo, zvuk, rekvizity, garderoba, 12 - 15 [redacted] 15 - 17 [redacted] od 17:00 [redacted] do 16:45 17:00 - svícení!		nehrajeme	stavba Evelinky [redacted]		Pokl.: 10:00 - 13:30, 14:30 - 20:00 Vrámce: 10:00 - 0:00 Uklid: 6:00 - 8:30
18. 10. Út		nehrajeme	9:30	Polní žínka Evelinka [71] technika: [redacted] garderoba: [redacted]			Pokl.: 8:30 - 13:30, 14:30 - 20:00 Vrámce: 8:30 - 22:00 Uklid: 6:00 - 8:30 13:30 VELKÁ PORADA v kanceláři
19. 10. St		nehrajeme	9:30	Polní žínka Evelinka [72] technika: [redacted] garderoba: [redacted]		17:00 - 20:00 Barboulka [redacted]	Pokl.: 8:30 - 13:30, 14:30 - 20:00 Vrámce: 8:30 - 20:00 Uklid: 6:00 - 8:30, telečelína 12:50 - 13:50 pláště
20. 10. Čt		nehrajeme	18:00	Polní žínka Evelinka [73] technika: [redacted] garderoba: [redacted]		9:30 - 12:15 masáže v telečelíně 8:30 školení bezpečnosti pro ohň, revize pojízdná lávka celá technika VS [redacted]	Pokl.: 10:00 - 13:30, 14:30 - 20:00 Vrámce: 8:00 - 20:30 Uklid: 6:00 - 8:30
21. 10. Pá	9:30	Tramtáře [1] veřejná generálka inspicie: [redacted] světlo, zvuk, garderoba: rekvizity: technika: [redacted] do 17:30	18:00	Mami, už tam budem? [18] technika: [redacted] garderoba: [redacted]	stavba Mami [redacted]	8:45 a po představení dlna Šar se divákem Barboulka vede: [redacted] 16:30 Čeme hrou dlna, Barboulka vede: [redacted]	Pokl.: 8:30 - 13:30, 14:30 - 20:00 Vrámce: 7:00 - 20:00 Uklid: 6:00 - 8:30, Barboulka,
22. 10. So	10:00	Tramtáře [2] veřejná generálka inspicie: [redacted] světlo, zvuk: garderoba: rekvizity: technika: 9:00 korepce [redacted] do 15:30 odháš tluhu na NR	10:00	Mami, už tam budem? [18] technika: [redacted] garderoba: [redacted]		10:00 - 12:00 Příběhy novoměstských ulic, outdoorová hra (start a cíl v Mínonu) 14:00, 16:00 Zvuk mi není fuk dlna MS vede: [redacted] technika: [redacted]	Pokl.: 9:00 - 16:00 Vrámce: 7:30 - 20:00 Uklid: 6:00 - 8:30
23. 10. Ne	18:00	Tramtáře [3] PREMIERA inspicie: [redacted] světlo, zvuk, garderoba: rekvizity: technika: [redacted]		nehrajeme		10:00 - 12:00 Republika I, outdoorová hra (start v Mínonu) premiérový aut., Tramtáře v pasáži pro diváky (poslone připravá, 1 prakták do pasáže, 1 prakták do Barboulky, kulaté stoly do pasáže, ubrusy)	Pokl.: 10:00 - 18:00 Vrámce: 9:00 - ? Uklid: 6:00 - 8:30
11:00		Dnes provází Minor [59] na novo městské radnici inspicie a rekvizity: [redacted]					

Obrázek 2.2: Ukázka týdenního fermanu divadla Minor (poznámka: byla z černěna jména skutečných zaměstnanců pro neporušení práv o sdílení osobních údajů)

Různá divadla s různou kapacitou pak používají rozdílné metody, kterými dospějí k výslednému hracímu plánu. Metody často zahrnují i použití volně dostupných nástrojů.

Nejčastější metody a nástroje pro plánování představení, které vycházejí z průzkumu, jsou následující:

■ "Papír a tužka"

Tato poněkud zastaralá metoda se dodnes v praxi používá v menších divadlech, kde produkční činnost není tak náročná jako ve větších divadlech. Hlavní výhodou tohoto přístupu je, že to nestojí divadlo žádné další finanční prostředky. Velkou nevýhodou je, že se člověk při dané metodě snadněji dopustí chyby, kterou by mohl normálně kontrolovat počítač. Zároveň jsou všechny potřebné informace často rozdělovány na více místech a je pak zbytečně složité je průběžně dohledávat.

■ Microsoft Excel či jiné online tabulky

Hlavní výhodou tohoto přístupu je existence nějakého sdíleného prostoru, kam mohou zaměstnanci zapisovat své poznámky v reálném čase. Herci zde mohou uložit informace o tom, kdy jsou k dispozici, a vedení divadla může přepisovat poznámky k vytvořenému fermanu. Všechny tyto důležité informace jsou pak na jednom místě, což velmi usnadňuje tvorbu fermanu. Tato metoda je obvykle používána ve středně velkých divadlech, kde působí obvykle více jak 100 herců a představení mohou probíhat ve dvou a více sálech najednou.

■ Softwarová aplikace pro divadlo

Jedná se přesně o ten typ aplikace, jejíž vývojem se tato práce zabývá. V praxi tento nástroj používají zejména větší divadla, která se svou velikostí podobají Národnímu divadlu. Správa umělecké činnosti by bez použití takové aplikace byla v dnešní době jen stěží proveditelná.

■ 2.1.3 Domluva s UTP

Umělecko-technický provoz (UTP) divadla sestává ze zaměstnanců zastávající nutné technické role pro obsluhu představení. Jeho součástí je inspicie, technici, osvětlovači, zvukaři, maskérky, ale často i vrátní. Domluva s nimi probíhá mimo produkci a až po vytvoření fermanu. UTP je zpravidla tvořen zaměstnanci divadla, takže se musí fermanu přizpůsobit. Zároveň jsou tyto role schopny tzv. alternací (zastoupení), takže se přidělením směn konkrétním osobám již nezabývá produkce, ale zaměstnanci si je rozdělí sami mezi sebou.

■ 2.1.4 Domluva s uvaděči

Rozdělením směn mezi uvaděče se obvykle zabývá inspektor hlediště. Pro zapisování směn se obvykle využívá nějaké jednoduché webové aplikace, sdíleného dokumentu nebo obyčejného papíru, na který se zapíše uvaděči v divadle. Často je to ovlivněno tím, kolik je uvaděčů let a do jaké míry jsou schopni používat modernější technologie. Někdy se kombinuje i více způsobů zápisu dohromady. Zápis probíhá nejčastěji měsíc dopředu, ale v některých divadlech klidně jen týden předem. Záleží na místní organizaci.

■ 2.1.5 Závěr průzkumu divadel

Produkce divadel, která fungují bez softwarových aplikací pro usnadnění správy umělecké činnosti divadla, by uvítaly práci s takovým produktem. Jsou avšak často skeptičtí k nasazení onoho systému do provozu. Zejména kvůli přesvědčení všech ostatních zaměstnanců k práci se systémem. Takovýto přechod by mohl ohrozit běžný chod divadla, který už teď bývá náročný sám o sobě. Zároveň z průzkumu vyplývá, že současná komunikace s herci je náročnější, než bývala před rokem 2020, kdy propukla pandemie virové choroby covid-19. Herci jsou z nějakého důvodu v komunikaci méně aktivní. Celkové upadnutí zájmu o kulturu ve společnosti dokazuje i pomalejší prodej lístků.

Je proto o to důležitější udělat systém co nejjednodušší a nejpréhlednější, aby se v něm jednotliví zaměstnanci dokázali pohybovat intuitivně a aby systém nabízel zjednodušení práce nejen produkci při plánování, ale i všem ostatním zaměstnancům, kteří by pak měli větší zájem s daným systémem pracovat a měli motivaci ho používat.

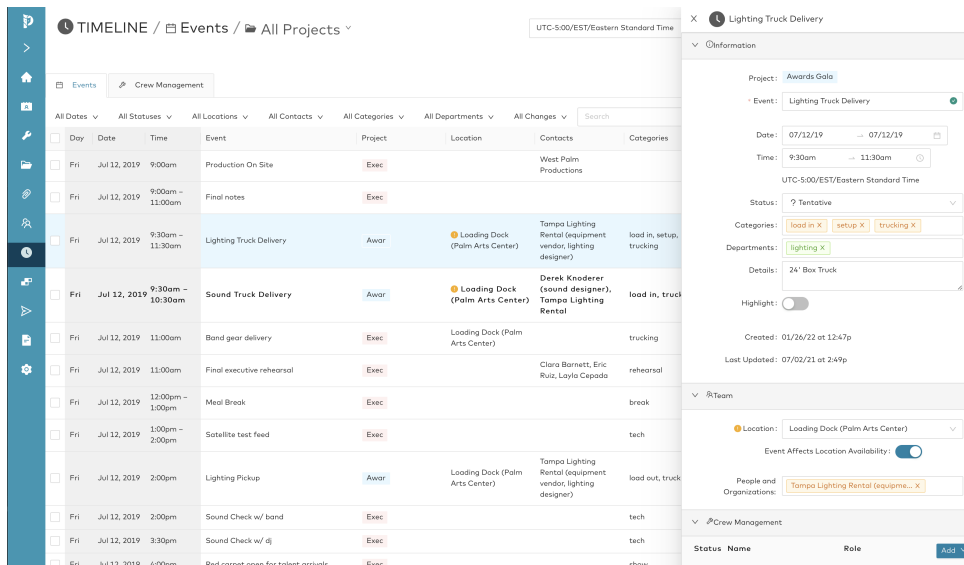
■ 2.2 Průzkum existujících řešení

V této části práce bylo prozkoumáno několik již fungujících systémů pro zjednodušení správy umělecké činnosti. Nejprve byly popsány jejich hlavní funkcionality, dále porovnány jednotlivé rozdíly a v závěru zhodnoceny jejich největší klady a zápory, aby bylo zřejmé, co je pro takovou aplikaci důležité a čemu je naopak dobré se vyhnout. Pro průzkum byly vybrány celkem tři systémy: Propared, Theatron a Theasoft. Všechny systémy jsou v praxi využívané desítky divadly a zároveň jsou dostatečně robustní, aby si poradily i se správou větších divadel.

■ 2.2.1 Propared

Propared je americký software pro sestavení plánu událostí a jeho následnou správu. Systém lze použít pro události různého typu, ale specializuje se na řízení umělecké činnosti. Využíván je zejména v divadlech, operách, uměleckých školách či pro organizaci festivalů.

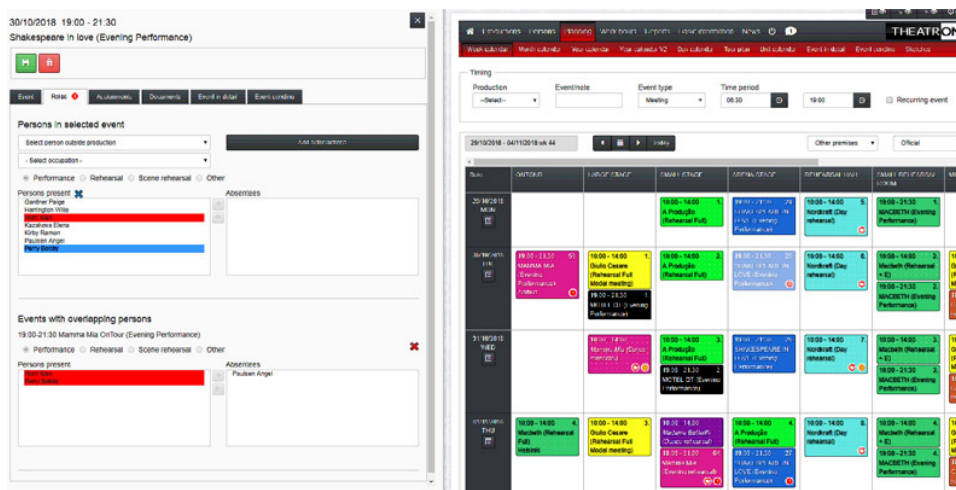
V aplikaci lze sestavit divadelní plán, na který se dá posléze nahlížet ve formě seznamu či kalendáře. Jednotlivé události lze uložit jako šablony pro budoucí využití. Zároveň je možnost vytvořené události naklonovat a posunout na jiný den. V kolonce *kontakty* lze uchovat všechny důležité informace o jednotlivých zaměstnancích. Jakmile je plán hotový, lze ho nasdílet přímo z aplikace zaměstnancům pomocí samoaktualizujícího URL, na kterém si mohou plán zobrazit, aniž by museli být v aplikaci přihlášení. V aplikaci lze udržovat i seznam rekvizit a dohlížet na jejich pravidelné revize. [3]



Obrázek 2.3: Ukázka softwaru Prepared

2.2.2 Theatron

Theatron je finský software vyvinutý přímo pro produkční plánování divadel a řízení jejich akcí. Lze zde namíru vytvořit denní, týdenní, měsíční i roční plány se všemi běžně potřebnými informacemi. Plány se zobrazují v podobě kalendáře a jednotlivé typy událostí jsou barevně rozlišené. Systém zároveň udržuje seznam reportů týkajících se představení, rekvizit či jiných důležitých věcí. Nakonec nabízí jednoduchý přístup sdílení důležitých informací pro své zaměstnance. Po přihlášení do aplikace vidí všechny nadcházející akce divadla nebo může zobrazit pouze akce, které se jich týkají přímo. Na takové akce jsou zároveň včas upozorňováni SMS notifikací. [4]



Obrázek 2.4: Ukázka softwaru Theatron

■ 2.2.3 Theasoft

Theasoft, podobně jako Theatron, obsahuje všechny důležité nástroje pro správu umělecké, orchestrální i technické činnosti divadel. Tento německý software nabízí kromě nástroje pro manuální vytvoření hracího plánu i automatické vygenerování takového plánu na základě zadaných parametrů o absencích jednotlivých herců. V průběhu manuálního sestavování fermanu (či jen jeho úpravy) zároveň aktivně upozorňuje na případnou kolizi herců. V systému lze vygenerovat i smlouvy pro zaměstnance. [5]



Obrázek 2.5: Logo softwaru Theasoft

■ 2.2.4 Závěr průzkumu existujících řešení

Největší výhoda zkoumaných softwarových aplikací je bezpochyby ta, že všichni uživatelé vidí změny provedené ostatními uživateli v reálném čase. Další velkou výhodou je upozorňování na případné kolize při manuálním sestavování hracího plánu.

Na druhou stranu tyto aplikace nabízejí až příliš mnoho funkcí, které nejsou pro chod menších (často i větších) divadel zásadní a dělají tak celý systém méně přehledný.

Byla proto snaha vybavit aplikaci jen těmi nejn nutnějšími nástroji a učinit tak používání aplikace novým uživatelům co nejsnadnější.

Kapitola 3

Využití umělé inteligence

Umělá inteligence (anglicky *Artificial intelligence*, zkráceně *AI*) se v dnešní době aplikuje v čím dál větším množství nejrůznějších odvětví. Přesto se ve správě divadla AI algoritmy téměř nepoužívají. Tato kapitola se proto věnuje zdůvodnění, proč by bylo i v tomto odvětví výhodné takové metody použít, a jak by zde její využití vypadalo.

3.1 Motivace a hlavní přínos AI

Nejnáročnější činností správy umělecké činnosti bývá hledání optimálního měsíčního fermanu. Několik již existujících divadelních systémů vytvořených zejména pro správu větších divadel se snaží s touto činností pomoci tak, že nabízí automatické vygenerování tohoto optimálního hracího plánu. Aby tak mohly učinit, uživatel jim musí nejprve specifikovat řadu nutných parametrů, které je při hledání optimálního řešení nutné dodržet.

V praxi se ale ukazuje, že i divadla, která takový systém vlastní, funkcionality automatického generování fermanů nevyužívají. Je to zejména proto, že každé divadlo má rozdílnou představu o optimálním fermanu a zadání všech potřebných parametrů systému je příliš náročné, někdy až nemožné.

Umělá inteligence nabízí způsob, jak uživateli sestavit optimální ferman bez nutnosti vyplňování řady parametrů. Divadelní systém tak lze učinit ještě jednodušším pro uživatele, což je jeden z nejdůležitějších předpokladů pro plynulé zavedení systému do provozu divadla.

3.2 Specifikace problému

Jak bylo naznačeno v předchozí sekci, problém, který by bylo výhodné vyřešit za pomoci metod umělé inteligence, je sestavení optimálního divadelního měsíčního plánu - fermanu. Nejprve je tento problém nutno podrobněji specifikovat.

Jedná se o problém plánování (anglicky *scheduling problem*), který je obvykle velmi náročný. Zejména kvůli velikosti množiny všech řešení, ve které je zapotřebí najít to optimální. V našem případě potřebujeme nalézt takový měsíční plán, který má optimálně rozvržená jednotlivá představení

tak, aby se podobala rozvržení v minulých fermanech, a zároveň byla pro každé představení zaručena podmínka, že jsou na něj v naplánovaném čase dostupní všichni potřební herci.

■ 3.3 Návrhy řešení

Pro nalezení optimálního měsíčního plánu byly navrženy celkem dvě různé metody, protože každá nabízí rozdílné výhody, a může být tak výhodné mezi nimi volit na základě požadavků pro výsledné řešení. Obě metody pro nalezení řešení zároveň využívají historická data o fermanech z minulých měsíců.

Nejprve bude zmíněna nutná příprava dat a poté návrh obou metod.

■ 3.3.1 Příprava dat

Jak už bylo zmíněno, každé divadlo má vlastní harmonogram, ve kterém je zvyklé uvádět jednotlivá představení. Proto navržené metody generují rozdílné fermány na základě toho, v jakém divadle jsou použity. Je tedy nezbytné, aby měly v těchto divadlech přístup k datům z místních fermanů z uplynulých měsíců.

Pro práci s daty je výhodné data z každého měsíčního fermanu reprezentovat maticí rozměru $V \times P$, kde V je počet vlastností charakterizující představení a P je počet představení pro daný měsíc. Každý sloupec tak bude odpovídat jednomu představení a každý řádek jedné vlastnosti. Jedná se o vlastnosti představení jako je čas začátku, žánr, délka představení, jeho identifikátor a další vlastnosti, které jsou shledány důležitými.

■ 3.3.2 1. metoda, využití genetického algoritmu

V 1. metodě je snaha hledat optimální měsíční ferman jako celek, aby mohly být zachyceny všechny vlastnosti fermanu, a byla tak možnost najít skutečně nejlepší či alespoň skoro nejlepší řešení. Pro takový přístup je efektivní využít Genetický algoritmus.

Genetický algoritmus (anglicky *Genetic algorithm*, zkráceně *GA*) je optimalizační technika založená na vyhledávání, která vychází z principů genetiky a přírodního výběru. Obvykle se používá k nalezení optimálních nebo téměř optimálních řešení obtížných problémů s velkou množinou možných řešení. Proto je často využívána v optimalizaci, ve výzkumu či ve strojovém učení. V GA máme k dispozici soubor nebo populaci možných řešení daného problému. Tato řešení pak podléhají rekombinaci a mutaci (podobně jako v přirozené genetice), čímž vznikají nová řešení, a tento proces se opakuje v různých generacích. Každé řešení je ohodnoceno na základě své kvality. Ta lepší řešení mají pak větší šanci se spářit a vytvořit další vhodná řešení. To je v souladu s Darwinovou teorií "přežití nejsilnějších". Tímto způsobem algoritmus vytváří kvalitnější řešení v průběhu generací, dokud není dosaženo kritéria pro zastavení algoritmu. [6]

■ Popis algoritmu

Tato sekce zobrazuje pseudokód daného algoritmu a následně podrobněji popisuje jeho jednotlivé části či úkony, které s ním souvisí. Pro názvy proměnných, funkcí a popis pseudokódu byl vybrán anglický jazyk, protože pseudokód již využívá anglická klíčová slova a nechtěla být porušena konzistence jazyka v rámci jedné sekce.

Algorithm 1 Monthly plan generation using Generation Algorithm

```

1: procedure GENERATEMONTHLYPLAN(populationSize, mutationRate,
   crossoverRate, numGenerations)
2:   population ← initializePopulation(populationSize)
3:   bestIndividual ← findBestIndividual(population)
4:   for generation in numGenerations do
5:     parents ← selectParents(population)
6:     offspring ← crossover(parents, crossoverRate)
7:     mutatedOffspring ← mutate(offspring, mutationRate)
8:     newPopulation ← selectSurvivors(population, mutatedOffspring)
9:     bestIndividual ← findBestIndividual(newPopulation)
10:    population ← newPopulation
11:  return bestIndividual

```

1. Repräsentace jedince

Ještě před samotným spuštěním algoritmu je nutné rozmyslet reprezentaci jedince, protože jsou na tom závislé jednotlivé části algoritmu a nevhodná reprezentace může významně ovlivnit výkon celého algoritmu.

Jedinec populace je v GA reprezentován pomocí chromozomu, který slouží jako struktura pro zakódování všech potřebných informací o jedinci. Dle specifikace problému byl chromozom navržen jako pole délky 28, 30 nebo 31, kde každý gen (buňka pole) reprezentuje jeden den v měsíci. Hodnota tohoto genu je seznam představení naplánovaných v den reprezentovaný genem. Nakonec je každé představení ze seznamu reprezentováno již zmiňovaným vektorem vlastností.

2. Inicializace populace

Jako základ pro prvotní populaci je zvolen náhodný historický měsíční ferman. Z tohoto fermanu jsou řadou náhodných změn vygenerováni jedinci populace. Počet nových jedinců (fermanů) se odvíjí od hodnoty proměnné *populationSize*. Náhodné změny na původním fermanu zahrnují změny představení, odebrání představení či přidání nových představení.

3. Ohodnocení jedinců Fitness funkcí

Jakmile je sestavena populace, je třeba ohodnotit její jedince použitím takzvané *fitness funkce*. Tato funkce vrací číselnou hodnotu na základě kvality parametrů jedince. Čím vyšší je kvalita, tím vyšší je navrácená hodnota. Dle specifikace problému uděluje fitness funkce hodnocení na základě těchto vlastností jedince:

- **počet naplánovaných představení**

Kladná hodnota skóre za každé přidané představení

- **podobnost s průměrným fermanem pro daný měsíc**

Naplánovaná představení jsou porovnána s průměrnými historickými představeními pro stejný měsíc. Jejich rozdíl je spočten jako euklidovská vzdálenost jejich vektorů. Suma všech těchto vzdáleností určuje pak celkovou penalizaci pro daného jedince.

- **proveditelnost**

Pokud plán obsahuje představení, které není proveditelné (herci, kteří v něm účinkují nejsou v daný časový interval k dispozici) je mu udělena velmi vysoká penalizace. Tato penalizace se sčítá za každé takové představení. Celkové ohodnocení plánu tak může díky této podmínce mít zápornou hodnotu.

4. Výběr rodičů

Po ohodnocení jedinců jsou vybráni jedinci, kteří se "spáří", aby vytvořili potomstvo pro další generaci. Jako tyto rodiče je důležité vybrat dobře ohodnocené jedince, protože vedou své potomstvo ke vhodnějším řešením a GA tak rychleji konverguje. Je však třeba dbát na to, aby jedno extrémně vhodné řešení neovládlo celou populaci během několika generací, protože to vede k tomu, že řešení jsou si v prostoru řešení blízká, což vede ke ztrátě rozmanitosti. Udržení dobré rozmanitosti v populaci je pro úspěch GA nesmírně důležité. Toto ovládnutí celé populace jedním extrémně vhodným řešením se nazývá předčasná konvergence (anglicky *premature convergence*) a je v GA nežádoucím stavem. [7]

Rodiče jsou proto vybráni turnajově (anglicky *Tournament Selection*), kdy vybereme z populace náhodně K jedinců a z nich vybereme nejlepšího. Stejný postup je opakován pro výběr druhého rodiče. Tato metoda může navíc pracovat se zápornými hodnotami fitness funkce, které výše navržená fitness funkce dovoluje.

5. Křížení

Křížení (anglicky *Crossover*) je obdobou reprodukce a biologického křížení. Jsou vybráni alespoň dva rodičové a z jejich genetického materiálu vzniká jeden nebo více potomků. [8]

Mezi nejběžnější operace pro křížení patří *One Point Crossover*, *Multi Point Crossover* či *Uniform Crossover*. Všechny spočívají v kombinování chromozomů obou rodičů. Vzhledem k tomu, že každá strategie může mít rozdílné dopady na výkon algoritmu, volba konkrétní strategie bude učiněna během implementace na základě dosažených výsledků při vyzkoušení každé z nich.

6. Mutace

Mutace představuje malou náhodnou změnu v chromozomu, která vede k získání nového řešení. Používá se k udržování a vnášení rozmanitosti do genetické populace a tím pádem způsob pro vyvarování se lokálního optima. Obvykle se uplatňuje s malou pravděpodobností. Pokud by byla pravděpodobnost velmi vysoká, z GA by se stalo náhodné hledání (takzvaný *random search*). [9]

Dle reprezentace jedince lze jako mutační operaci použít takzvanou *Swap Mutation*. V této mutaci se náhodně vyberou dva geny chromozomu a navzájem si prohodí pozice.

7. Výběr přeživších

Před vstupem do nové generace je populace redukována o některé jedince. Opět je důležité zajistit, aby vhodnější řešení postoupila do nové generace, ale zároveň byla zachována její diverzita. [10]

Mezi nejčastější strategie redukce populace patří *Age Based Selection* a *Fitness Based Selection*. Jak vyplývá z názvu, jedinci jsou z populace vyloučeni buď podle toho, jak dlouho existují (kolik generací už mají za sebou), nebo podle toho, jak moc kvalitní řešení reprezentují. Vzhledem k tomu, že každá strategie může mít rozdílné dopady na výkon algoritmu, volba konkrétní strategie bude učiněna během implementace na základě dosažených výsledků při vyzkoušení každé z nich.

8. Uložení nejlepšího potomka

Na konci každé *for* smyčky je průběžně ukládáno řešení s nejlepším hodnocením, které je pak funkcí navraceno při dosažení stanoveného maximálního počtu generací uloženého v proměnné *numGenerations*.

■ Výhody a nevýhody

1. Výhody

■ Velká míra podobnosti s předchozími fermany

Vzhledem k tomu, že není měsíční plán hledán po částech, ale jako celek, je možnost zaměřit se na více závislostí, které uvnitř plánu vznikají. Lze se pak vyhnout opakovanosti představení, nevyváženosti obsazení jednotlivých herců a podobně.

2. Nevýhody

■ Náročnost

Množina všech dostupných řešení exponenciálně roste s počtem představení, které je divadlo schopno v rámci jednoho měsíce uvádět. Pro velký počet představení může být pak tato množina příliš velká a konvergence k optimálnímu plánu tak může být pomalá.

■ 3.3.3 2. metoda, využití horolezeckého algoritmu

Ve druhé metodě není snaha hledat optimální měsíční ferman jako celek, ale po částech. Jsou tak hledány jednotlivé optimální denní plány pro každý den v měsíci. Množina možných řešení je tak výrazně zredukována a problém se tak stává mnohem jednodušším. Namísto hledání nejlepšího řešení je hledáno dostatečně dobré řešení, které vyhovuje potřebám pro ferman. Pro hledání takového řešení je efektivní využít Horolezecký algoritmus.

Horolezecký algoritmus (anglicky *Hill climbing algorithm*) je jednoduchý optimalizační algoritmus používaný v umělé inteligenci (AI) k nalezení nejlepšího či alespoň dobrého řešení daného problému. Patří do rodiny algoritmů lokálního prohledávání. Algoritmus začíná s počátečním řešením a poté v něm iterativně provádí malé změny s cílem řešení zlepšit. Tyto změny jsou založeny na heuristické funkci, která vyhodnocuje kvalitu řešení. Algoritmus pokračuje v provádění těchto malých změn, dokud nedosáhne lokálního maxima. Toto lokální maximum je pak považováno za optimální řešení. [11]

■ Popis algoritmu

Tato sekce zobrazuje pseudokód daného algoritmu a následně podrobněji popisuje jeho jednotlivé části. Pro názvy proměnných, funkcí a popis pseudokódu byl vybrán anglický jazyk, protože pseudokód již využívá anglická klíčová slova a nechtěla být porušena konzistence jazyka v rámci jedné sekce.

Algorithm 2 Monthly plan generation using Hill Climbing Algorithm

```

1: procedure GENERATEMONTHLYPLAN(month)
2:   monthlyPlan ← initializeMonthlyPlan()
3:   for day in month do
4:     currentPlan ← getRandomDailyPlan()
5:     while not converged do
6:       neighborhood ← generateNeighborhood(currentPlan)
7:       bestNeighbor ← findBestNeighbor(neighborhood)
8:       if bestNeighbor is better than currentPlan then
9:         currentPlan ← bestNeighbor
10:      else
11:        converged ← True
12:      updateMonthlyPlan(day, monthlyPlan, currentPlan)
13:   return monthlyPlan

```

1. Inicializace měsíčního plánu

Měsíční plán stačí inicializovat na prázdný seznam, do kterého budou postupně přidávány jednotlivá představení.

2. Inicializace denního plánu

Jako denní plán je náhodně vybrán jeden z historických denních plánů odpovídajících danému dni. Při výběru jsou zvýhodněny denní plány spadající pod stejnojmenný měsíc, jako je měsíc, pro který je hledán optimální plán. Tyto plány jsou vybrány s vyšší pravděpodobností. Takto definovaná inicializace pomůže k rychlejší konvergenci k dobrému řešení.

3. Generování sousedů

Z historického denního plánu jsou vytvořené nové denní plány náhodnými změnami. Tyto změny zahrnují změny představení, odebrání představení či přidání nových představení. Na závěr je vytvořen denní plán neobsahující žádné představení pro případ, aby byl alespoň jeden z těchto plánů proveditelný.

4. Ohodnocení Fitness funkcí

Jakmile je vygenerována množina sousedů, je třeba každého souseda ohodnotit použitím *fitness funkce*. Fitness funkce této metody uděluje hodnocení na základě těchto vlastností jedince:

- **počet naplánovaných představení**

Kladná hodnota skóre za každé přidané představení.

- **podobnost s průměrným denním plánem**

Naplánovaná představení jsou porovnána s průměrnými historickými představeními pro stejný den. Jejich rozdíl je spočten jako euklidovská vzdálenost jejich vektorů. Suma všech těchto vzdáleností určuje pak celkovou penalizaci pro daného souseda.

- **proveditelnost**

Pokud plán obsahuje představení, které není proveditelné (herci, kteří v něm účinkují, nejsou v daný časový interval k dispozici) je mu udělena velmi vysoká penalizace. Tato penalizace se sčítá za každé takové představení. Celkové ohodnocení plánu tak může díky této podmínce mít zápornou hodnotu.

5. **Výběr nejlepšího souseda**

Na základě zhodnocení sousedů je vybrán nejlepší z nich. Ten, pokud je lepší, nahradí současný denní plán. Pokud není, současný denní plán je považován za optimum a hledání je u konce.

6. **Aktualizace měsíčního plánu**

Nalezený denní plán je přidán do měsíčního plánu a algoritmus pokračuje hledáním plánu pro následující den.

- **Výhody a nevýhody**

1. **Výhody**

- **Rychlost a spolehlivost**

Vzhledem k tomu, že je měsíční plán hledán po jednotlivých dnech a ne jako celek, je tak množina všech dostupných řešení výrazně zredukována. Nemusí být tak nutně nalezeno nejlepší řešení, ale přinejmenším dobré, které je pro účely fermanu dostačující. Toto řešení je pak nalezeno s větší jistotou a ve výrazně lepším čase.

- **Jednoduchost**

Tato metoda je v porovnání s první metodou snazší pro implementaci. Není třeba definovat podobu chromozomů a řadu operací, které s nimi souvisí.

2. **Nevýhody**

- **Menší míra podobnosti s předchozími fermany**

Vzhledem k tomu, že se optimální řešení hledá po jednotlivých dnech, nelze tak zachytit závislosti mezi jednotlivými dny, kterými jsou postiženy historické fermany. V takovém plánu se pak mohou jednotlivá představení opakovat či k nim budou jednotliví herci obsazeni nerovnoměrně.

Kapitola 4

Návrh vlastností systému

Na základě provedené analýzy byly specifikovány funkční požadavky, nefunkční požadavky, případy užití a potřebná databáze. [12]

4.1 Funkční požadavky

Funkční požadavky (anglicky *functional requirement*, zkráceně *FR*) představují všechny funkce, které budou uživatelům v rámci aplikace k dispozici. Výčet konkrétních funkčních požadavků je rozdělen podle jednotlivých modulů aplikace, kde budou tyto funkce zapotřebí.

4.1.1 Přístup do systému

■ FR-01 - Přihlášení uživatele

Všichni uživatelé se pro přístup do aplikace přihlásí zadáním uživatelského jména a hesla.

■ FR-02 - Odhlášení uživatele

Všichni přihlášení uživatelé se mohou odhlásit.

4.1.2 Uživatelský profil

■ FR-03 - Zobrazení svého profilu

Všichni uživatelé si mohou zobrazit svůj profil se všemi důležitými informacemi.

■ FR-04 - Úprava svého profilu

Všichni uživatelé si mohou v rámci svého profilu upravit své kontaktní údaje včetně hesla.

4.1.3 Domovská stránka

■ FR-05 - Zobrazení svých nadcházejících přihlášených směn

Oprávnění uživatelé si mohou zobrazit své nadcházející přihlášené směny.

■ **FR-06 - Zobrazení vzkazů na nástěnce**

Všichni uživatelé si mohou zobrazit vzkazy na nástěnce.

■ **FR-07 - Přidat vzkaz na nástěнку**

Oprávnění uživatelé mohou přidat vzkaz na nástěнку.

■ **FR-08 - Odebrat vzkaz z nástěinky**

Oprávnění uživatelé mohou odebrat vzkaz z nástěinky.

■ **4.1.4 Zaměstnanci**

■ **FR-09 - Zobrazení všech zaměstnanců**

Všichni uživatelé si mohou zobrazit seznam všech zaměstnanců včetně jejich kontaktních údajů.

■ **FR-10 - Vytvoření nového zaměstnance**

Oprávnění uživatelé mohou vytvořit účet novému zaměstnanci.

■ **FR-11 - Úprava profilu zaměstnance**

Oprávnění uživatelé mohou upravit profil libovolného zaměstnance.

■ **FR-12 - Smazání existujícího zaměstnaneckého účtu**

Oprávnění uživatelé mohou smazat účet existujícího zaměstnance.

■ **4.1.5 Představení**

■ **FR-13 - Zobrazení všech představení**

Všichni uživatelé si mohou zobrazit přehled všech představení.

■ **FR-14 - Zobrazení detailu představení**

Všichni uživatelé si mohou zobrazit detailní informace o konkrétním představení.

■ **FR-15 - Vytvoření nového představení**

Oprávnění uživatelé mohou vytvořit nové představení s nastavením parametrů jako je název, délka trvání, určený sál či místo, role, obsazení, služba, datum premiéry, datum derniéry a poznámka.

■ **FR-16 - Úprava existujícího představení**

Oprávnění uživatelé mohou změnit parametry existujícího představení.

■ **FR-17 - Smazání existujícího představení**

Oprávnění uživatelé mohou smazat existující představení.

■ 4.1.6 Směny

■ **FR-18 - Přihlášení se na směnu**

Oprávnění uživatelé se mohou přihlásit na vypsanou směnu.

■ **FR-19 - Odhlášení se ze směny**

Oprávnění uživatelé se mohou odhlásit ze směny, na které jsou přihlášení.

■ **FR-20 - Přihlášení cizího uživatele na směnu**

Oprávnění uživatelé mohou přihlásit cizí uživatele na směnu.

■ **FR-21 - Odhlášení cizího uživatele ze směny**

Oprávnění uživatelé mohou odhlásit cizího uživatele ze směny.

■ **FR-22 - Přidání poznámky ke směně**

Oprávnění uživatelé mohou přidat poznámku k libovolné směně.

■ 4.1.7 Indispozice

■ **FR-23 - Přidání své nové indispozice**

Oprávnění uživatelé mohou zaznamenat své indispozice - tedy konkrétní dny a časy, v nichž nejsou k dispozici.

■ **FR-24 - Úprava své existující indispozice**

Oprávnění uživatelé mohou upravit informace u svých existujících indispozic.

■ **FR-25 - Smazání své existující indispozice**

Oprávnění uživatelé mohou smazat své existující indispozice.

■ **FR-26 - Zobrazit své indispozice**

Oprávnění uživatelé si mohou zobrazit své nahlašené indispozice.

■ **FR-27 - Zobrazit indispozice všech uživatelů**

Oprávnění uživatelé si mohou zobrazit indispozice všech uživatelů.

■ **FR-28 - Přidání indispozice cizímu uživateli**

Oprávnění uživatelé mohou cizímu uživateli přidat novou indispozici.

■ **FR-29 - Úprava existující indispozice cizímu uživateli**

Oprávnění uživatelé mohou cizímu uživateli upravit jeho existující indispozici.

■ **FR-30 - Smazání existující indispozice cizímu uživateli**

Oprávnění uživatelé mohou cizímu uživateli smazat jeho existující indispozici.

■ 4.1.8 Fermany

■ FR-31 - Zobrazení fermanů

Všichni uživatelé si mohou zobrazit všechny fermany. Aktuální i minulé.

■ FR-32 - Vytvoření měsíčního fermanu

Oprávnění uživatelé mohou vytvořit nový měsíční ferman. Pro zvolený den a čas systém nabídne pouze ta představení, která jsou v danou dobu realizovatelná (dle možností herců). Pro tato představení lze vybrat herce, kteří budou do daných rolí obsazeni a nastavené představení pak následně vložit do plánu. Představení, která už jsou v plánu zasazena, lze opět odebrat.

■ FR-34 - Úprava měsíčního fermanu

Oprávnění uživatelé mohou upravit existující ferman - zasadit do něj nová představení či odebrat ta stávající.

■ FR-34 - Smazání měsíčního fermanu

Oprávnění uživatelé mohou smazat existující ferman a s ním i všechny směny, které se k němu vztahují.

■ FR-35 - Zobrazení historických směn

Všichni uživatelé mohou zobrazit všechny proběhlé směny.

■ 4.2 Nefunkční požadavky

Nefunkční požadavky (anglicky *non-functional requirement*, zkráceně *NFR*) představují takové vlastnosti systému, které uživatel nemusí nijak ovládat a jsou mu systémem automaticky poskytovány.

■ NFR-01 - Zabránění vzniku kolize při tvorbě fermanu

Při přidávání představení do měsíčního plánu systém dovolí přidat pouze taková představení, po jejichž přidání nevznikne kolize s jiným představením či nejsou obsazeni herci, kteří nemají v danou dobu čas.

■ NFR-02 - Zobrazení fermanu

Ferman bude jak při tvorbě, tak při běžném náhledu zobrazen ve formě tabulky, kde každý řádek bude symbolizovat jeden den v měsíci, a každý sloupec bude prostor, ve kterém daná akce (představení) probíhá.

■ NFR-03 - Dostupnost přes webový prohlížeč

Systém bude dostupný svým uživatelům přes webový prohlížeč.

■ NFR-04 - Uživatelská práva

Systém bude nabízet rozdílné funkcionality svým uživatelům na základě jejich pracovní pozice. Zároveň bude zabraňovat uživatelům učinit takové akce, na které nemají právo.

■ 4.3 Případy užití

Případ užití (anglicky *use case*, zkráceně *UC*) je koncept používaný při vývoji softwaru, návrhu produktů a v dalších oblastech, který popisuje, jak lze systém použít k vykonání určitých úkolů. Popisuje interakce mezi uživateli a systémem za účelem dosažení konkrétního výsledku.[15]

Nejprve byly definovány uživatelské role, které systém nabízí, a pak jednotlivé případy užití.

■ 4.3.1 Rozdělení uživatelských rolí a práv

Protože je výhodné reagovat na uživatelské akce na základě jejich role, byly specifikovány následující uživatelské role, které popisují jejich pracovní pozice:

■ Herec, Herečka

Může si zobrazit přehled všech svých indispozic. Ty může následně mazat, či přidávat nové.

■ UTP - Inspice

Může si zobrazit směny a přihlašovat se na ně, či se z nich odhlašovat. Má kontrolu nad zápisem směn všech zaměstnanců patřící k *UTP* kromě zaměstnanců s rolí *UTP - Inspice*.

■ Služba - Inspektor hlediště

Může si zobrazit směny a přihlašovat se na ně, či se z nich odhlašovat. Má kontrolu nad zápisem směn všech zaměstnanců patřící k *Služba* kromě zaměstnanců s rolí *Služba - Inspektor hlediště*.

■ UTP - Technik/Osvětlovač/Zvukař/Stavěč/Garderóba/ Maskérna/Rekvizity/Ostatní

Může si zobrazit směny, vybrat libovolnou z nich a přihlásit se na ni, či se z ní odhlásit.

■ Služba - Uvaděč/Bar/Úklid/Ostatní

Může si zobrazit směny, vybrat libovolnou z nich a přihlásit se na ni, či se z ní odhlásit.

■ Kancelář

Má přístup ke všem funkcionalitám systému.

Kromě toho mají všichni zaměstnanci přístup k základním funkcionalitám, jako je zobrazení kontaktních údajů zaměstnanců, zobrazení fermanů, zobrazení parametrů představení a náhled i úprava uživatelského profilu včetně obnovy hesla.

■ 4.3.2 Jednotlivé případy užití

V této sekci byly definovány jednotlivé případy užití, které byly pro větší přehlednost rozděleny podle jednotlivých modulů aplikace. Zároveň bylo u každého specifikováno, zda je společný pro všechny uživatele, či pouze pro oprávněné uživatele s určitou rolí.

■ 4.3.3 Přístup do systému

■ UC-01 - Přihlášení

Oprávnění uživatelé: Všichni

Scénář:

1. Uživatel je automaticky přesměrován na stránku s přihlašovacím formulářem, pokud se snaží přistoupit do aplikace a není přihlášen
2. Uživatel zadá přihlašovací jméno a heslo a klikne na tlačítko *Přihlásit se*
3. **IF** zadané údaje souhlasí s údaji uloženými v databázi
 - a. **THEN** Systém uživatele přihlásí a udělí přístup do aplikace
 - b. **ELSE** Systém vypíše chybovou hlášku a znovu načte stránku s přihlašovacím formulářem

■ UC-02 - Odhlášení

Oprávnění uživatelé: Všichni

Scénář:

1. Uživatel klikne na tlačítko *Odhlásit se*
2. Systém uživatele odhlásí a přesměruje ho na stránku s přihlašovacím formulářem

■ 4.3.4 Uživatelský profil

■ UC-03 - Zobrazení svého profilu

Oprávnění uživatelé: Všichni

Scénář:

1. Uživatel klikne na tlačítko *Můj profil*
2. Systém přesměruje uživatele na stránku zobrazující jeho profil

■ UC-04 - Úprava svého profilu

Oprávnění uživatelé: Všichni

Scénář:

1. **IF** Uživatel klikne na tlačítko *Obnovit heslo*
 - a. **THEN** Systém zobrazí dvě nová pole, kam uživatel může zadat své nové heslo
2. Uživatel změní své osobní informace na stránce svého profilu
3. Uživatel klikne na tlačítko *Uložit změny*
4. **IF** jsou zadané informace ve správném formátu
 - a. **THEN** Systém nahradí staré informace novými a uloží tyto změny
 - b. **ELSE** Systém vypíše chybovou hlášku a neprovede žádné změny

■ 4.3.5 Domovská stránka

■ UC-05 - Zobrazení svých nadcházejících přihlášených směn

Oprávnění uživatelé: *Herci, UTP, Služba*

Scénář:

1. Uživatel klikne na tlačítko *Domovská stránka*
2. Systém přesměruje uživatele na domovskou stránku, která zobrazuje jeho nadcházející směny

■ UC-06 - Zobrazení vzkazů na nástěnce

Oprávnění uživatelé: Všichni

Scénář:

1. Uživatel klikne na tlačítko *Domovská stránka*
2. Systém přesměruje uživatele na domovskou stránku, která zobrazuje nástěnku se všemi vzkazy

■ UC-07 - Přidat vzkaz na nástěnku

Oprávnění uživatelé: *Kancelář, UTP - Inspice, Služba - Inspektor hlediště*

Scénář:

1. Uživatel klikne v pravém horním rohu nástěnky na tlačítko *Přidat novou poznámku*
2. Systém zobrazí formulář pro přidání nové poznámky
3. Uživatel vyplní formulář a na konci potvrdí tlačítkem *Přidat*
4. Systém uloží novou poznámku a načte znovu domovskou stránku

■ UC-08 - Odebrat vzkaz z nástěnky

Oprávnění uživatelé: *Kancelář, UTP - Inspice, Služba - Inspektor hlediště*

Scénář:

1. Uživatel klikne v pravém horním rohu konkrétní poznámky na tlačítko *Smazat poznámku*
2. Systém smaže odpovídající poznámku a načte znovu domovskou stránku

■ 4.3.6 Zaměstnanci

■ UC-09 - Zobrazení všech zaměstnanců

Oprávnění uživatelé: Všichni

Scénář:

1. Uživatel klikne na tlačítko *Zaměstnanci*
2. Systém přeměruje uživatele na stránku *Zaměstnanci*, která zobrazuje všechny zaměstnance uložené v systému

■ UC-10 - Vytvoření nového zaměstnance

Oprávnění uživatelé: *Kancelář*

Scénář:

1. Uživatel klikne na tlačítko *Přidat zaměstnance*
2. Systém zobrazí formulář pro přidání nového zaměstnance
3. Uživatel vyplní formulář a potvrdí uložení zaměstnance stisknutím tlačítka *Uložit*
4. **IF** jsou zadané informace ve správném formátu
 - a. **THEN** Systém vytvoří nového zaměstnance na základě zadaných informací
 - b. **ELSE** Systém vypíše chybovou hlášku a načte znovu stránku

■ UC-11 - Úprava profilu zaměstnance

Oprávnění uživatelé: *Kancelář*

Scénář:

1. Uživatel klikne do tabulky zaměstnanců na řádek obsahující informace o konkrétním zaměstnanci
2. Systém zobrazí formulář pro úpravu informací vybraného zaměstnance
3. Uživatel změní potřebné parametry zaměstnance a potvrdí své změny stisknutím tlačítka *Uložit*
4. **IF** jsou zadané informace ve správném formátu
 - a. **THEN** Systém uloží změny u konkrétního zaměstnance
 - b. **ELSE** Systém vypíše chybovou hlášku a načte znovu stránku

■ UC-12 - Smazání existujícího zaměstnaneckého účtuOprávnění uživatelé: *Kancelář*

Scénář:

1. Uživatel klikne do tabulky zaměstnanců na řádek obsahující informace o konkrétním zaměstnanci
2. Systém zobrazí formulář pro úpravu informací vybraného zaměstnance
3. Uživatel klikne na tlačítko *Smazat*
4. **IF** uživatel neúčinkuje v žádném představení
 - a. **THEN** Systém smaže vybraného uživatele z databáze spolu se všemi dalšími objekty, které jsou na něm závislé
 - b. **ELSE** Systém vypíše chybovou hlášku a načte znovu stránku

■ 4.3.7 Představení**■ UC-13 - Zobrazení všech představení**

Oprávnění uživatelé: Všichni

Scénář:

1. Uživatel klikne na tlačítko *Představení*
2. Systém přeměruje uživatele na stránku *Představení*, která zobrazuje všechna představení uložená v systému

■ UC-14 - Zobrazení detailu představení

Oprávnění uživatelé: Všichni

Scénář:

1. Uživatel klikne do tabulky představení na řádek obsahující informace o konkrétním představení
2. Systém zobrazí detailní informace vybraného představení

■ UC-15 - Vytvoření nového představeníOprávnění uživatelé: *Kancelář*

Scénář:

1. Uživatel klikne na tlačítko *Přidat představení*
2. Systém zobrazí formulář pro přidání nového představení
3. Uživatel vyplní formulář a potvrdí uložení představení stisknutím tlačítka *Uložit*
4. **IF** jsou zadané informace ve správném formátu
 - a. **THEN** Systém vytvoří nové představení na základě zadaných informací
 - b. **ELSE** Systém vypíše chybovou hlášku a načte znovu stránku

■ UC-16 - Úprava existujícího představení

Oprávnění uživatelé: *Kancelář, UTP - Inspice, Služba - Inspektor hlediště*

Scénář:

1. Uživatel klikne do tabulky představení na řádek obsahující informace o konkrétním představení
2. Systém zobrazí formulář pro úpravu parametrů vybraného představení
3. Uživatel změní potřebné parametry představení a potvrdí své změny stisknutím tlačítka *Uložit*
4. **IF** jsou zadané informace ve správném formátu
 - a. **THEN** Systém uloží změny u konkrétního představení
 - b. **ELSE** Systém vypíše chybovou hlášku a načte znovu stránku

■ UC-17 - Smazání existujícího představení

Oprávnění uživatelé: *Kancelář*

Scénář:

1. Uživatel klikne do tabulky představení na řádek obsahující informace o konkrétním představení
2. Systém zobrazí formulář pro úpravu parametrů vybraného představení
3. Uživatel klikne na tlačítko *Smazat*
4. Vybrané představení je smazáno z databáze spolu se všemi dalšími objekty, které jsou na něm závislé

■ 4.3.8 Směny

■ UC-18 - Přihlášení se na směnu

Oprávnění uživatelé: *UTP, Služba*

Scénář:

1. Uživatel klikne na tlačítko *Zápis směn*
2. Systém přesměruje uživatele na stránku *Zápis směn*, která zobrazuje všechny vypsane směny pro aktuální měsíc
3. Uživatel si vyfiltruje směny pro měsíc, který potřebuje
4. Uživatel klikne do tabulky směn na řádek, který odpovídá směně, na kterou se chce přihlásit
5. Systém zobrazí náhled vybrané směny a uživatel v tomto náhledu klikne na tlačítko *Přihlásit se*
6. Systém přihlásí uživatele na zvolenou směnu a znovu načte stránku

■ UC-19 - Odhlášení se ze směnyOprávnění uživatelé: *UTP, Služba*

Scénář:

1. Uživatel klikne na tlačítko *Zápis směn*
2. Systém přesměruje uživatele na stránku *Zápis směn*, která zobrazuje všechny vypsané směny pro aktuální měsíc
3. Uživatel si vyfiltruje směny pro měsíc, který potřebuje
4. Uživatel klikne do tabulky směn na řádek, který odpovídá směně, ze které se chce odhlásit
5. Systém zobrazí náhled vybrané směny a uživatel v tomto náhledu klikne na tlačítko *Odhlásit se*
6. Systém odhlásí uživatele ze zvolené směny a znovu načte stránku

■ UC-20 - Přihlášení cizího uživatele na směnuOprávnění uživatelé: *Kancelář, UTP - Inspice, Služba - Inspektor hlediště*

Scénář:

1. Uživatel klikne na tlačítko *Zápis směn* nebo *Zápis směn a indispozic*
2. **IF** stisknuté tlačítko bylo *Zápis směn a indispozic*
 - a. **THEN** Uživatel vybere na nově načtené stránce záložku *Zápis směn*
3. Systém přesměruje uživatele na stránku *Zápis směn*, která zobrazuje všechny vypsané směny pro aktuální měsíc
4. Uživatel si vyfiltruje směny pro měsíc, který potřebuje
5. Uživatel klikne do tabulky směn na řádek, který odpovídá směně, na kterou chce přihlásit cizího uživatele
6. Systém zobrazí náhled vybrané směny a uživatel zapíše uživatele na jejich pozice
7. Uživatel klikne na tlačítko *Uložit změny*
8. Systém přihlásí dané uživatele na zvolenou směnu a znovu načte stránku

■ UC-21 - Odhlášení cizího uživatele ze směnyOprávnění uživatelé: *Kancelář, UTP - Inspice, Služba - Inspektor hlediště*

Scénář:

1. Uživatel klikne na tlačítko *Zápis směň* nebo *Zápis směň a indispozic*
2. **IF** stisknuté tlačítko bylo *Zápis směň a indispozic*
 - a. **THEN** Uživatel vybere na nově načtené stránce záložku *Zápis směň*
3. Systém přesměruje uživatele na stránku *Zápis směň*, která zobrazuje všechny vypsané směny pro aktuální měsíc
4. Uživatel si vyfiltruje směny pro měsíc, který potřebuje
5. Uživatel klikne do tabulky směň na řádek, který odpovídá směně, ze které chce odhlásit cizího uživatele
6. Systém zobrazí náhled vybrané směny a uživatel odepíše uživatele z jejich pozic
7. Uživatel klikne na tlačítko *Uložit změny*
8. Systém odhlásí dané uživatele ze zvolené směny a znovu načte stránku

■ UC-22 - Přidání poznámky ke směněOprávnění uživatelé: *Kancelář, UTP - Inspice, Služba - Inspektor hlediště*

Scénář:

1. Uživatel klikne na tlačítko *Zápis směň* nebo *Zápis směň a indispozic*
2. **IF** stisknuté tlačítko bylo *Zápis směň a indispozic*
 - a. **THEN** Uživatel vybere na nově načtené stránce záložku *Zápis směň*
3. Systém přesměruje uživatele na stránku *Zápis směň*, která zobrazuje všechny vypsané směny pro aktuální měsíc
4. Uživatel si vyfiltruje směny pro měsíc, který potřebuje
5. Uživatel klikne do tabulky směň na řádek, který odpovídá směně, ke které chce přidat poznámku
6. Systém zobrazí náhled vybrané směny a uživatel změní text poznámky
7. Uživatel klikne na tlačítko *Uložit změny*
8. Systém uloží novou poznámku ke zvolené směně a znovu načte stránku

■ 4.3.9 Indispozice

■ UC-23 - Přidání své nové indispozice

Oprávnění uživatelé: *Herci*

Scénář:

1. Uživatel klikne na tlačítko *Zápis indispozic*
2. Systém přesměruje uživatele na stránku *Zápis indispozic*, která zobrazuje všechny zapsané indispozice pro aktuální měsíc
3. Uživatel klikne na tlačítko *Přidat indispozici*
4. Systém zobrazí formulář pro přidání nové indispozice
5. Uživatel vyplní formulář a potvrdí vytvoření indispozice stisknutím tlačítka *Uložit*
6. **IF** je formulář správně vyplněný
 - a. **THEN** Systém vytvoří novou indispozici na základě zadaných parametrů
 - b. **ELSE** Systém vypíše chybovou hlášku a znovu načte stránku

■ UC-24 - Úprava své existující indispozice

Oprávnění uživatelé: *Herci*

Scénář:

1. Uživatel klikne na tlačítko *Zápis indispozic*
2. Systém přesměruje uživatele na stránku *Zápis indispozic*, která zobrazuje všechny zapsané indispozice pro aktuální měsíc
3. Uživatel si vyfiltruje indispozice pro měsíc, který potřebuje
4. Uživatel klikne do tabulky indispozic na řádek, který odpovídá indispozici, kterou chce změnit
5. Systém zobrazí náhled vybrané indispozice
6. Uživatel změní parametry v náhledu indispozice a potvrdí změny stisknutím tlačítka *Uložit*
7. **IF** je formulář správně vyplněný
 - a. **THEN** Systém uloží změny zvolené indispozice na základě zadaných parametrů
 - b. **ELSE** Systém vypíše chybovou hlášku a znovu načte stránku

■ UC-25 - Smazání své existující indispoziceOprávnění uživatelé: *Herci*

Scénář:

1. Uživatel klikne na tlačítko *Zápis indispozic*
2. Systém přesměruje uživatele na stránku *Zápis indispozic*, která zobrazuje všechny zapsané indispozice pro aktuální měsíc
3. Uživatel si vyfiltruje indispozice pro měsíc, který potřebuje
4. Uživatel klikne do tabulky indispozic na řádek, který odpovídá indispozici, kterou chce smazat
5. Systém zobrazí náhled vybrané indispozice
6. Uživatel klikne v náhledu na tlačítko *Smazat*
7. Systém smaže záznam zvolené indispozice z databáze

■ UC-26 - Zobrazit své indispoziceOprávnění uživatelé: *Herci*

Scénář:

1. Uživatel klikne na tlačítko *Zápis indispozic*
2. Systém přesměruje uživatele na stránku *Zápis indispozic*, která zobrazuje všechny zapsané indispozice pro aktuální měsíc
3. Uživatel si vyfiltruje indispozice pro měsíc, který potřebuje
4. Systém zobrazí indispozice přihlášeného uživatele pro zvolený měsíc

■ UC-27 - Zobrazit indispozice všech uživatelůOprávnění uživatelé: *Kancelář*

Scénář:

1. Uživatel klikne na tlačítko *Zápis směn a indispozic*
2. Systém přesměruje uživatele na stránku *Zápis indispozic*, která zobrazuje všechny zapsané indispozice pro aktuální měsíc
3. Uživatel si vyfiltruje indispozice pro měsíc, který potřebuje
4. Systém zobrazí indispozice všech uživatelů pro zvolený měsíc

■ UC-28 - Přidání indispozice cizímu uživateliOprávnění uživatelé: *Kancelář*

Scénář:

1. Uživatel klikne na tlačítko *Zápis směn a indispozic*
2. Systém přesměruje uživatele na stránku *Zápis indispozic*, která zobrazuje všechny zapsané indispozice pro aktuální měsíc
3. Uživatel klikne na tlačítko *Přidat indispozici*
4. Systém zobrazí formulář pro přidání nové indispozice
5. Uživatel vyplní formulář a potvrdí vytvoření indispozice stisknutím tlačítka *Uložit*
6. **IF** je formulář správně vyplněný
 - a. **THEN** Systém vytvoří novou indispozici na základě zadaných parametrů
 - b. **ELSE** Systém vypíše chybovou hlášku a znovu načte stránku

■ UC-29 - Úprava existující indispozice cizímu uživateliOprávnění uživatelé: *Kancelář*

Scénář:

1. Uživatel klikne na tlačítko *Zápis směn a indispozic*
2. Systém přesměruje uživatele na stránku *Zápis indispozic*, která zobrazuje všechny zapsané indispozice pro aktuální měsíc
3. Uživatel si vyfiltruje indispozice pro měsíc, který potřebuje
4. Uživatel klikne do tabulky indispozic na řádek, který odpovídá indispozici, kterou chce změnit
5. Systém zobrazí náhled vybrané indispozice
6. Uživatel změní parametry v náhledu indispozice a potvrdí změny stisknutím tlačítka *Uložit*
7. **IF** je formulář správně vyplněný
 - a. **THEN** Systém uloží změny zvolené indispozice na základě zadaných parametrů
 - b. **ELSE** Systém vypíše chybovou hlášku a znovu načte stránku

■ UC-30 - Smazání existující indispozice cizímu uživateli

Oprávnění uživatelé: *Kancelář*

Scénář:

1. Uživatel klikne na tlačítko *Zápis indispozic* nebo *Zápis směn a indispozic*
2. Systém přesměruje uživatele na stránku *Zápis indispozic*, která zobrazuje všechny zapsané indispozice pro aktuální měsíc
3. Uživatel si vyfiltruje indispozice pro měsíc, který potřebuje
4. Uživatel klikne do tabulky indispozic na řádek, který odpovídá indispozici, kterou chce smazat
5. Systém zobrazí náhled vybrané indispozice a uživatel klikne na tlačítko *Smazat*
6. Systém smaže zvolenou indispozici a znovu načte stránku

■ 4.3.10 Fermany

■ UC-31 - Zobrazení fermanů

Oprávnění uživatelé: Všichni

Scénář:

1. Uživatel klikne na tlačítko *Ferman*
2. Systém přesměruje uživatele na stránku *Zobrazení fermanů*, která zobrazuje všechny vytvořené fermany
3. Uživatel klikne na tlačítko *Zobrazit/upravit* v tabulce fermanů na řádku, který odpovídá fermanu, který chce zobrazit
4. Systém přesměruje uživatele na stránku zobrazující zvolený ferman

■ UC-32 - Vytvoření měsíčního fermanu

Oprávnění uživatelé: *Kancelář*

Scénář:

1. Uživatel klikne na tlačítko *Ferman*
2. Systém přesměruje uživatele na stránku *Zobrazení fermanů*, která zobrazuje všechny vytvořené fermany
3. Uživatel klikne na záložku *Tvorba fermanu*
4. Systém přesměruje uživatele na stránku *Volba data pro nový ferman* zobrazující formulář pro vytvoření nového fermanu
5. Uživatel vybere ve formuláři měsíc pro nový ferman a potvrdí svou volbu stisknutím tlačítka *Pokračovat*
6. Systém vytvoří nový ferman pro měsíc, který uživatel zadal a přesměruje uživatele na stránku *Úprava fermanu* pro nově vytvořený ferman

■ UC-33 - Úprava měsíčního fermanuOprávnění uživatelé: *Kancelář*

Scénář:

1. Uživatel klikne na tlačítko *Fermany*
2. Systém přesměruje uživatele na stránku *Zobrazení fermanů*, která zobrazuje všechny vytvořené fermany
3. Uživatel klikne na tlačítko *Zobrazit/upravit* v tabulce fermanů na řádku, který odpovídá fermanu, který chce upravit
4. Systém přesměruje uživatele na stránku *Úprava fermanu*
5. V rámci této stránky může uživatel provádět změny na zvoleném fermanu - přidávat do plánu nová představení s konkrétními parametry či z něj odebírat ta stávající

■ UC-34 - Smazání měsíčního fermanuOprávnění uživatelé: *Kancelář*

Scénář:

1. Uživatel klikne na tlačítko *Fermany*
2. Systém přesměruje uživatele na stránku *Fermany*, která zobrazuje všechny vytvořené fermany
3. Uživatel klikne na tlačítko *Smazat* v tabulce fermanů na řádku, který odpovídá fermanu, který chce smazat
4. Systém smaže odpovídající ferman z databáze a načte znovu stránku

■ UC-35 - Zobrazení historických směn

Oprávnění uživatelé: Všichni

Scénář:

1. Uživatel klikne na tlačítko *Fermany*
2. Systém přesměruje uživatele na stránku *Fermany*, která zobrazuje všechny vytvořené fermany
3. Uživatel klikne na tlačítko *Zobrazit/upravit* v tabulce fermanů na řádku, který odpovídá měsíčnímu fermanu, který obsahuje směny, které chce uživatel zobrazit
4. Systém přesměruje uživatele na stránku *Úprava fermanu*
5. V rámci této stránky uživatel vidí všechna představení (směny) v měsíci, kterému odpovídá měsíc fermanu. Při kliknutí na nějakou z nich, systém zobrazí detail směny

■ UC-36 - Přidání představení do fermanu

Oprávnění uživatelé: *Kancelář*

Scénář:

1. Uživatel klikne na stránce *Úprava fermanu* na tlačítko *Přidat nové představení* v buňce tabulky, která odpovídá řádkem dni, ve kterém se bude nové představení hrát, a sloupci, ke kterému sálu se toto představení váže
2. Systém zobrazí formulář pro zvolení času, kdy by mělo představení začínat
3. Uživatel vyplní čas a potvrdí stisknutím tlačítka *Potvrdit*
4. **IF** je v tento den, v tento čas a v tomto sále možno hrát alespoň jedno představení
 - a. **THEN** Systém zobrazí formulář a nabídne uživateli všechna představení, která lze do plánu zasadit na základě dostupnosti herců či sálu
 - (i) Uživatel vybere jedno z nabízených představení a potvrdí stisknutím tlačítka *Potvrdit*
 - (ii) Systém zobrazí finální parametry zvoleného představení
 - (iii) Uživatel může změnit obsazení pro představení (pokud lze) a potvrdí přidání tohoto představení do fermanu stisknutím tlačítka *Přidat*
 - (iv) Systém přidá představení na základě nastavených parametrů do právě upravovaného fermanu
 - b. **ELSE** Systém zobrazí informativní hlášku, že pro tento den, čas a sál nelze přidat do plánu žádné představení

■ UC-37 - Odebrání představení z fermanu

Oprávnění uživatelé: *Kancelář*

Scénář:

1. Uživatel klikne na stránce *Úprava fermanu* na štítek popisující konkrétní představení ve fermanu
2. Systém zobrazí panel s parametry tohoto představení
3. Uživatel klikne na tlačítko *Odebrat*
4. Systém odebere zvolené představení z právě upravovaného fermanu

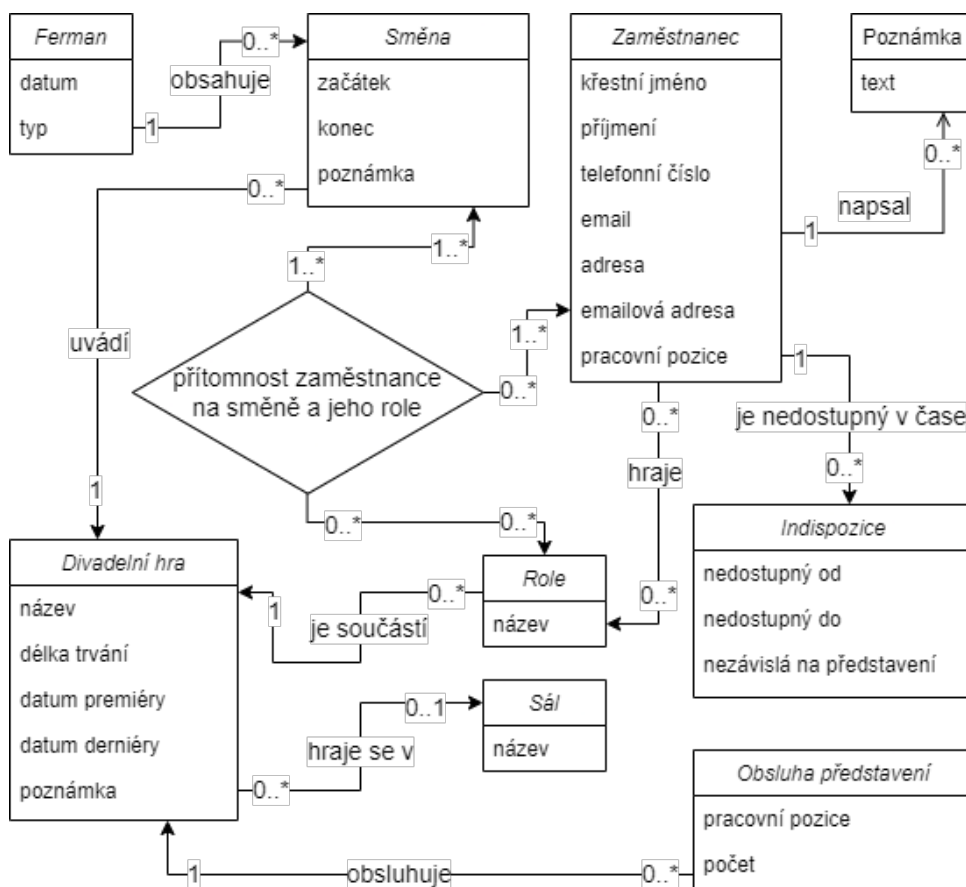
4.4 Databáze

Na základě analýzy byla navržena databáze, kterou musí systém implementovat, aby byly splněny všechny výše zmíněné požadavky.

Nejprve byl návrh databáze stručně popsán doménovým modelem a poté byly podrobněji popsány jednotlivé entity a vztahy mezi nimi.

4.4.1 Doménový model

Doménový model je takový druh diagramu, který zachycuje objekty reálného světa a popisuje jejich vlastnosti pomocí atributů. Cílem doménového modelu je popsat nějaké entity a vztahy mezi nimi tak, aby později bylo možné vytvořit model databázový. Zároveň může doménový model fungovat jako kontrolní prvek mezi tvůrcem softwaru a jejím zadavatelem – obě strany se tak přesvědčí, že si vzájemně porozuměly při návrhu [13].



Obrázek 4.1: Doménový diagram.

■ 4.4.2 Entity a vazby mezi nimi

Entity představují fyzické či abstraktní objekty reálného světa. Jsou definované názvem a svými atributy.

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 ke druhé. Ke každé z rolí přiřazujeme takzvanou kardinalitu. Kardinalita představuje omezení v počtu instancí druhé entity, které mají vztah s jakoukoliv instancí první entity. Definuje se vždy maximální a minimální kardinalita.[14]

■ Zaměstnanec

Entita *Zaměstnanec* představuje zaměstnance divadla. Ukládáme u ní údaje o zaměstnanci, jako je jeho křestní jméno, příjmení, telefonní číslo, emailová adresa, adresa a pracovní pozice. Zároveň může zaměstnanec být autorem libovolného množství poznámek, mít libovolné množství záznamů o jeho indispozicích, udělený libovolný počet hereckých rolí a být přiřazený k libovolnému počtu směn, kde může (a nemusí) ztvárnit nějakou určitou roli.

■ Poznámka

Entita *Poznámka* představuje poznámku připnutou na společné nástěnce pro všechny uživatele systému. Ukládáme u ní samotný text poznámku. Zároveň odkazuje na svého autora (zaměstnance, který ji napsal).

■ Indispozice

Entita *Indispozice* představuje indispozici konkrétního zaměstnance. Ukládáme u ní časovou značku začátku a konce této indispozice a zda je tato indispozice vytvořena manuálně či automaticky (například přiřazením herce k nějakému představení). Dále odkazuje na zaměstnance, pro kterého je tato indispozice vypsána.

■ Role

Entita *Role* představuje hereckou roli pro konkrétní divadelní hru. Ukládáme u ní název role. Zároveň odkazuje na libovolný počet herců, kteří ji mohou ztvárnit, na směny, ve kterých se objevila a jaký zaměstnanec ji hrál, a na konkrétní divadelní hru, ke které patří.

■ Divadelní hra

Entita *Divadelní hra* představuje divadelní hru. Ukládáme u ní údaje o hře, jako je její název, délka trvání, datum premiéry, datum derniéry a písemnou poznámku. Zároveň odkazuje na směny, které byly pro hru vytvořeny, na role a obsluhy, které jsou pro hru definovány a na sál, ve kterém se hra hraje.

■ Sál

Entita *Sál* představuje sál, ve kterém se odehrávají divadelní vystoupení. Ukládáme u ní název sálu. Zároveň odkazuje na všechny divadelní hry, které jsou k tomuto sálu přiřazeny.

■ Obsluha představení

Entita *Obsluha představení* představuje obsluhu představení, která je nutná pro odbavení konkrétní divadelní hry. Ukládáme u ní název potřebné pracovní pozice a potřebný počet zaměstnanců vykonávající tuto pozici. Odkazuje na divadelní hru, kterou má obsluhovat.

■ Směna

Entita *Směna* představuje pracovní směnu. Ukládáme u ní časovou značku začátku a konce směny a písemnou poznámku. Zároveň odkazuje na divadelní hru, kterou tato směna uvádí, na všechny zaměstnance (a jejich případné role), kteří byli na této směně přítomni a na ferman, ke kterému se směna váže.

■ Ferman

Entita *Ferman* představuje divadelní ferman. Ukládáme u ní typ fermanu (měsíční či týdenní) a datum, ke kterému se ferman váže. Zároveň odkazuje na všechny směny, které se k němu váží.

Kapitola 5

Implementace

Na základě analýzy a následného návrhu požadovaných vlastností systému byl tento systém implementován.

Pro vývoj bylo použito vývojové prostředí PyCharm od společnosti JetBrains. Toto prostředí bylo vybráno, protože poskytuje skvělou podporu všech jazyků a frameworků, které bylo třeba využít. Současně nabízí řadu užitečných nástrojů pro vývoj webových aplikací jako je například přístup k databázi přímo z vývojového prostředí. Na závěr poskytuje inteligentní našeptávač, zvýrazňování chyb za chodu, refaktoring či bohaté možnosti navigace. [16]

Podrobnější popis implementace byl rozdělen pro větší přehlednost na dvě části: Backend a Frontend.

5.1 Backend

Backend je serverová část webových aplikací. Ukládá a uspořádává data a také zajišťuje, aby vše na klientské straně webu (na frontendu) fungovalo správně. Uživatelé tuto část webu nevidí, ale nepřímo s ní komunikují pomocí rozhraní, které nabízí frontend. [17]

Nejprve bude zmíněna řada nástrojů a technologií, které byly pro vývoj backendu využity a dále bude popsána vytvořená databáze spolu s hlavní logikou backendu.

5.1.1 Použité nástroje a technologie

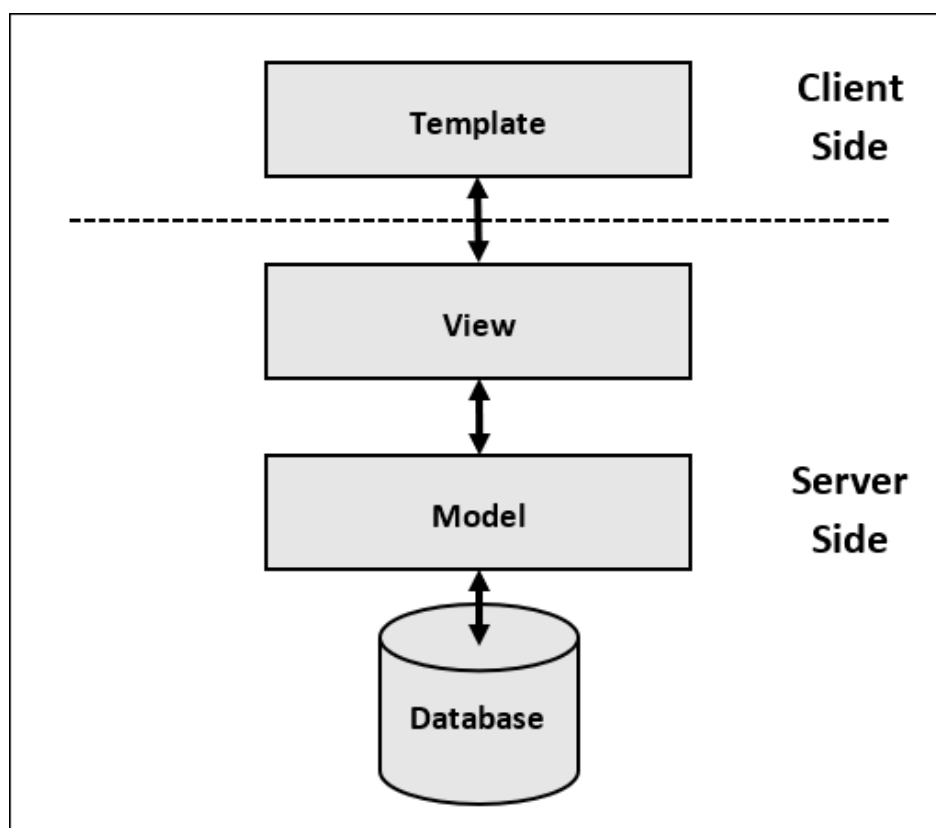
Python

Python je interpretovaný, objektově orientovaný programovací jazyk vysoké úrovně s dynamickou sémantikou. Jednoduchá a snadno naučitelná syntaxe jazyka Python klade důraz na čitelnost, a tím snižuje náklady na údržbu programu. Jeho vysokoúrovňové vestavěné datové struktury v kombinaci s dynamickým typováním a dynamickými vazbami jej činí velmi atraktivním pro Rapid Application Development (moderní přístup vývoje softwaru, kde je kladen důraz na vysokou flexibilitu a rychlost vývoje). [18]

■ Django

Django je vysokoúrovňový open-source framework pro vývoj webových aplikací v jazyce Python. Kromě poskytování desítek doplňků pro řešení problémů při vývoji se Django stará i o zabezpečení aplikace. [19]

Django implementuje architekturu MTV (Model-Template-View). Jedná se o variantu známější architektury MVC (Model-View-Controller), která ale usnadňuje práci s prezentací dat. Pro větší čitelnost a udržitelnost kódu rozděluje aplikace na tři vrstvy: Model, Template a Views. Model zajišťuje komunikaci s databází, Template reprezentuje prezentační vrstvu aplikace a View představuje logiku, která řídí, jaká data budou zobrazena a jak se budou chovat. [20]



Obrázek 5.1: Grafický diagram architektury MTV. [20]

■ PostgreSQL

PostgreSQL je výkonný spolehlivý objektově-relační databázový open-source systém. [21]

5.1.2 Databáze

Na základě návrhu databáze byla s využitím zmíněného PostgreSQL odpovídající databáze vytvořena.

Tato databáze byla později rozšířena o další tabulky, které automaticky vytvoří zmíněný framework Django. Tyto tabulky se starají o společné vlastnosti webových aplikací, jako je například uložení přihlašovacích údajů uživatele a jejich zašifrování pro větší zabezpečení. Na popis těchto tabulek odkazuje následující reference [22].

5.1.3 Logika a komunikace s databází

Vzhledem k tomu, že byl pro implementaci využit zmíněný framework Django, veškerá logika aplikace je obsažena ve vrstvě View. Naopak funkce pro práci s daty a komunikaci s databází jsou obsaženy ve vrstvě Model.

Model

Django poskytuje API pro abstrakci databáze, které umožňuje vytvářet, načítat, aktualizovat a mazat objekty. K reprezentaci dat databázových tabulek v objektech jazyka Python používá intuitivní systém: Třída modelu představuje databázovou tabulku a instance této třídy představuje konkrétní záznam v databázové tabulce. [23]

V rámci jednotlivých tříd byly definovány pomocné funkce, které lze volat na konkrétních objektech a usnadnit tak udržitelnost kódu a práci s daty.

```
class ActorIndisposition(models.Model):
    class Meta:
        managed = False
        db_table = 'actor_indisposition'

    not_available_from = models.DateTimeField()
    not_available_to = models.DateTimeField()
    employee = models.ForeignKey('Employee', on_delete=models.CASCADE)
    independent = models.BooleanField(False)
```

Obrázek 5.2: Ukázka třídy *Model*

View

Vrstva view reprezentuje logiku celé aplikace. Kód implementující tuto logiku byl pro větší přehlednost dále rozdělen do odlišných souborů podle toho, zda šlo pouze o *view logic*, či *business logic*. Do skupiny *view logic* patří pouze funkce, které reagují na požadavky od uživatelů a vrací konkrétní šablonu s odpovídajícím kontextem, zatímco do skupiny *business logic* patří funkce, které odpovídají logice specifické problematiky, kterou se aplikace zabývá.

```
@login_required(login_url='/login/')
def home(request):
    employee = get_employee(request.user)
    upcoming_shifts = tools.get_upcoming_shifts(employee)

    context = {
        'user_shifts': upcoming_shifts,
        'group': employee.get_group(),
        'notes': [x.as_dict() for x in Note.objects.all()],
        'employee_position': employee.employee_position,
        'employee_full_name': employee.get_full_name(),
        'employee_position_cz': Employee.translation_position[employee.employee_position],
    }
    return render(request, 'home.html', context)
```

Obrázek 5.3: Ukázka funkce z *view logic*

```
def is_place_occupied(place_id, time_start, time_end):
    # get shifts that overlap our time interval
    shifts = Shift.objects.filter(
        time_start__lte=time_end,
        time_end__gte=time_start,
    )
    for shift in shifts:
        if shift.theatre_play and shift.theatre_play.place_id == int(place_id):
            return True
    return False
```

Obrázek 5.4: Ukázka funkce z *business logic*

■ 5.2 Frontend

Frontend je klientská část webových aplikací, která je pro uživatele viditelná a tudíž je zodpovědná za interakci s ním. Zahrnuje vše, s čím se uživatelé setkávají přímo: barvy a styly textu, obrázky, grafy a tabulky, tlačítka, barvy a navigační menu. [24]

Nejprve bude zmíněna řada nástrojů a technologií, které byly pro vývoj frontendu využity a dále bude ukázáno samotné uživatelské rozhraní.

■ 5.2.1 Použité nástroje a technologie

■ HTML, CSS

HTML (zkratka pro *Hypertext Markup Language*) a CSS (zkratka pro *Cascading Style Sheets*) jsou dvě základní technologie pro tvorbu webových stránek. Jazyk HTML zajišťuje strukturu stránky, CSS její vizuální rozvržení pro různá zařízení. [25]

■ Django

Již zmíněný framework Django byl částečně použit i pro frontend. Spolu s nástroji pro vývoj backendu Django nabízí takzvaný *Django Template Language*, který umožňuje dynamicky rozhodovat, jaká data budou na stránce zobrazena. [26]

■ Javascript

JavaScript je skriptovací jazyk pro vytváření dynamického obsahu webových stránek. Zároveň nabízí množství frameworků a knihoven pro zjednodušení vývoje webových aplikací. [27]

■ jQuery

jQuery je rychlá, malá a funkčně bohatá knihovna jazyka JavaScript. Díky snadno použitelnému rozhraní API, které funguje v mnoha prohlížečích, usnadňuje například procházení a manipulaci s dokumenty HTML, zpracování událostí, animace a Ajax. [28]

■ Selectize

Selectize je knihovna JavaScriptu, která poskytuje bohaté uživatelské prostředí pro výběr položek ze seznamu. Podporuje výběr jedné a více hodnot, vyhledávání, označování, vzdálené datové sady a nekonečné scrollování. [29]

■ Bootstrap

Bootstrap je bezplatný front-end framework pro rychlejší a snadnější vývoj webových stránek. Obsahuje předdefinované HTML a CSS šablony pro jednotný design formulářů, tlačítek, tabulek a mnoho dalšího. [30]

■ Font Awesome

Font Awesome je internetová knihovna ikon a sada nástrojů, kterou používají miliony designérů, vývojářů a tvůrců obsahu. [31]

■ 5.2.2 Uživatelské rozhraní

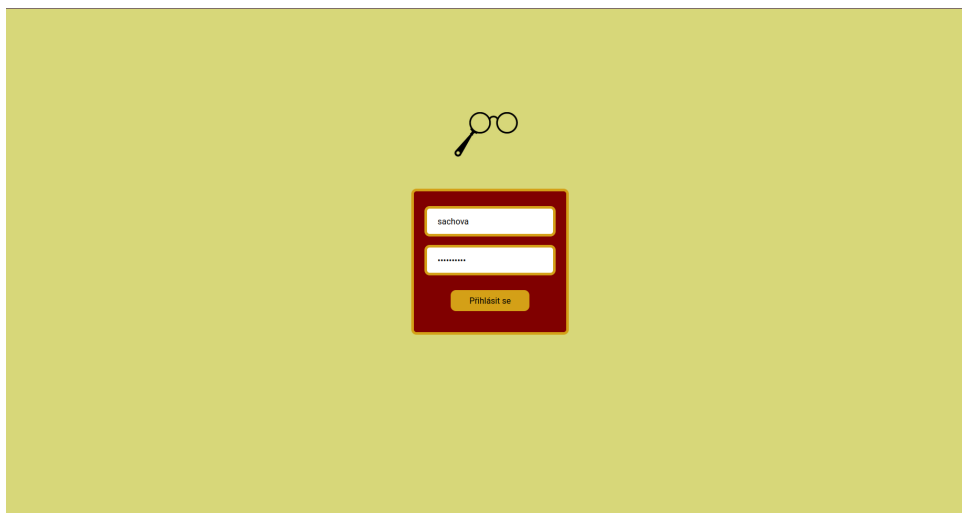
Uživatelské rozhraní bylo rozděleno do 9 modulů: *Přihlašovací stránka*, *Úvodní stránka*, *Můj Profil*, *Zaměstnanci*, *Představení*, *Přehled fermanů*, *Úprava a detailní náhled fermanu*, *Zápis směn* a *Zápis indispozic*.

Do těchto modulů se lze dostat přes navigační panel na levé straně obrazovky, panel na horní straně obrazovky, či přes záložky v rámci konkrétní stránky. Byla vyvinuta snaha udělat uživatelské prostředí intuitivní a samo popisující, aby nebylo při zavedení systému třeba důkladného školení.

Pro lepší ukázkou byly vytvořeny umělé data na základě skutečných představení a herců z oficiálních stránek divadla Minor. Data týkající se osobních informací, jako jsou telefonní čísla a adresy, byla vymyšlena. [32]

■ Přihlašovací stránka

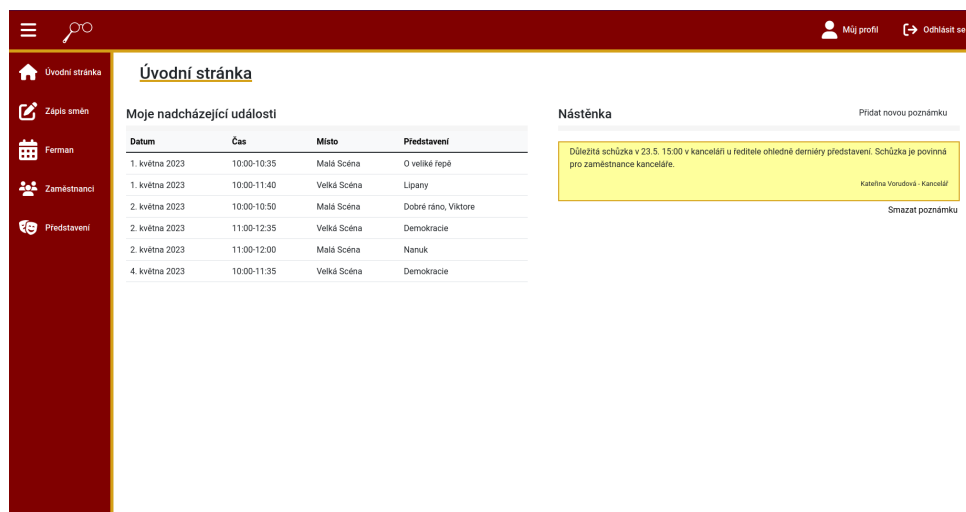
V rámci přihlašovací stránky může uživatel zadat své přihlašovací údaje a získat tak přístup do systému.



Obrázek 5.5: Podoba stránky pro přihlášení.

Úvodní stránka

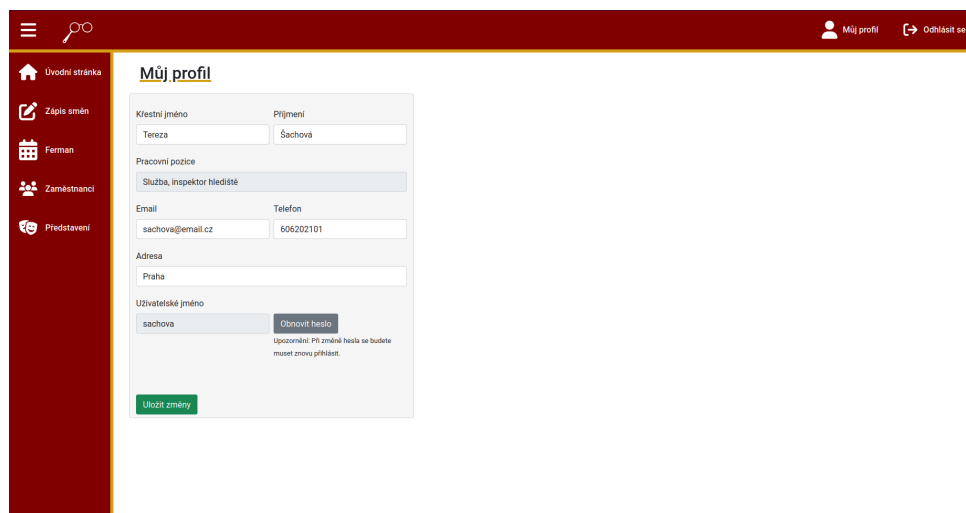
Úvodní stránka je zobrazena ihned po úspěšném přihlášení. Uživatel zde vidí seznam svých nadcházejících směn spolu s nástěnkou, kam mohou oprávnění uživatelé přidávat vzkazy.



Obrázek 5.6: Podoba úvodní stránky z pohledu uživatele s rolí *Inspektor hlediště*.

Můj Profil

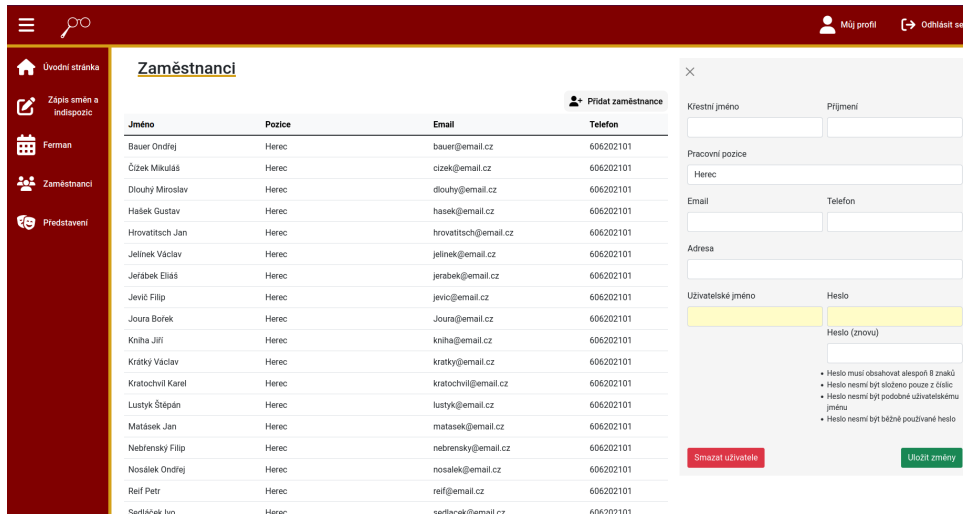
Stránka profilu uživatele obsahuje všechny důležité informace o uživateli. Některé z nich má uživatel právo změnit. Kromě změny svých osobních údajů lze i nastavit nové heslo pro přihlášení.



Obrázek 5.7: Podoba stránky vlastního profilu uživatele.

Zaměstnanci

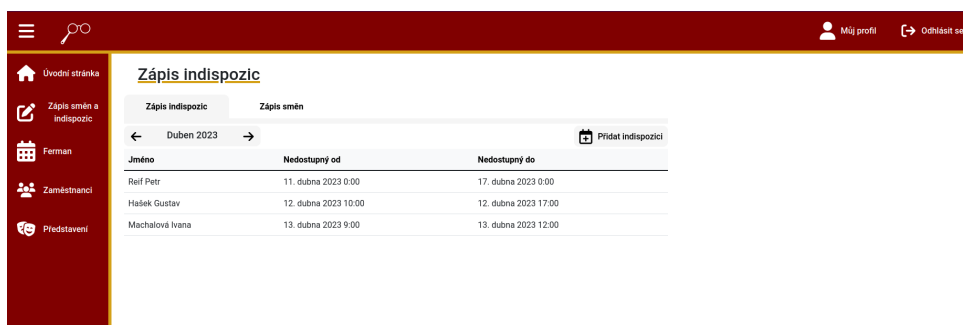
Tato stránka zobrazuje seznam všech zaměstnanců evidovaných systémem. Lze tak nalézt kontakt na každého uživatele. Oprávnění zaměstnanci mohou navíc upravovat údaje těchto zaměstnanců či je dokonce smazat z systému nebo přidat nového zaměstnance.



Obrázek 5.8: Podoba stránky *Zaměstnanci*.

Zápis indispozic

Tato stránka zobrazuje seznam indispozic pro konkrétní rok a měsíc (automaticky je zvolen ten aktuální, uživatel si však může vybrat libovolný). Uživatelé s rolí *herec/herečka* zde mohou přidávat, upravovat a mazat své indispozice. Oprávnění zaměstnanci navíc mohou provádět tyto akce pro libovolného uživatele s rolí *herec/herečka*.



Obrázek 5.9: Podoba stránky *Zápis indispozic*.

Zápis směn

Tato stránka zobrazuje seznam směn pro konkrétní rok a měsíc (automaticky je zvolen ten aktuální, uživatel si však může vybrat libovolný). Uživatelé se na tyto směny mohou přihlašovat, pokud směna vyžaduje jejich pracovní pozici a nejsou všechny tyto pozice již obsazené. Oprávnění zaměstnanci mohou na tyto směny navíc přihlašovat či odhlašovat ostatní uživatele a přidávat ke směnám písemné poznámky.

Datum	Začátek	Konec	Představení	Přihlášená obsluha
1. května 2023	10:00	11:40	Lipany	0 / 6
1. května 2023	10:00	10:35	O veliké řepě	4 / 4
2. května 2023	10:00	10:50	Dobré ráno, Viktoro	0 / 4
2. května 2023	11:00	12:35	Demokracie	0 / 6
2. května 2023	11:00	12:00	Nanuk	0 / 4
4. května 2023	10:00	11:35	Demokracie	0 / 6

Obrázek 5.10: Podoba stránky *Zápis směn*.

Představení

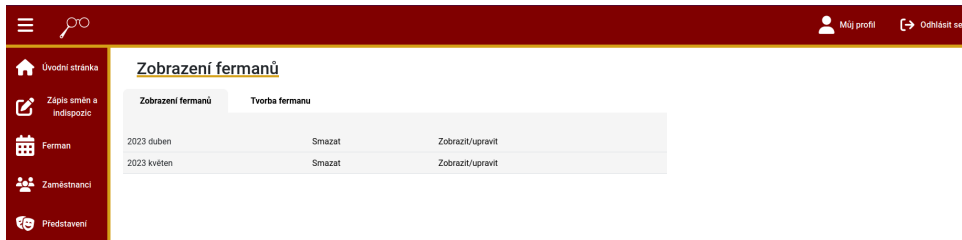
Tato stránka zobrazuje seznam všech představení evidovaných systémem. Každé představení lze navíc rozkliknout a nechat si zobrazit podrobnější informace, které nejsou uvedeny na řádku v tabulce. Oprávnění zaměstnanci mohou tyto parametry navíc měnit či představení smazat z systému nebo přidat nové. Zároveň mohou v rámci této stránky také vytvořit nový sál či smazat stávající.

Název	Sál/Místo	Délka trvání
Demokracie	Velká Scéna	95 minut
Dobré ráno, Viktoro	Malá Scéna	50 minut
Lipany	Velká Scéna	100 minut
Nanuk	Malá Scéna	60 minut
O veliké řepě	Malá Scéna	35 minut

Obrázek 5.11: Podoba stránky *Představení*.

Přehled fermanů

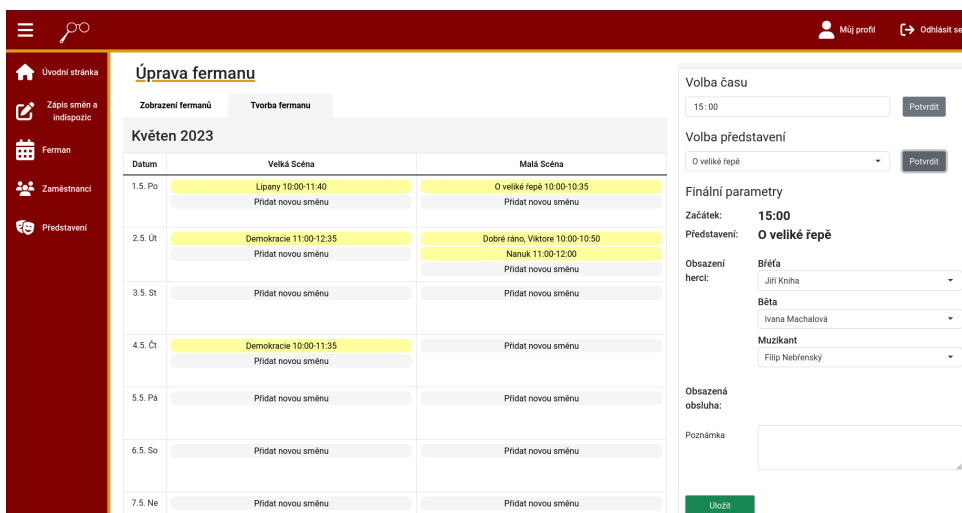
Tato stránka zobrazuje seznam všech uložených fermanů. Každý ferman si lze rozkliknout a zobrazit si jaké směny jsou v něm naplánovány. Oprávnění zaměstnanci mohou tyto fermany navíc mazat. Spolu s fermanem tak smažou i všechny směny, které mu odpovídají. Zároveň mohou v rámci této stránky zahájit tvorbu nového fermanu při stisknutí záložky *Tvorba fermanu*.



Obrázek 5.12: Podoba stránky *Přehled fermanů*.

Úprava a detailní náhled fermanu

Tato stránka zobrazuje konkrétní ferman a všechny směny, které jsou pro něj naplánovány. Každou směnu si lze rozkliknout a zobrazit si její parametry. Oprávnění zaměstnanci mohou tyto směny navíc mazat nebo přidávat nové. Při přidání nové směny je uživatel dotázán na čas a systém mu automaticky vybere taková představení, která jsou uskutečnitelná bez možnosti vzniknutí kolize s jiným představením probíhajícím ve vybraném sále či s indispozicemi herců. Po výběru představení má ještě uživatel možnost přiřadit herce k rolím podle svých preferencí. Zároveň mohou v rámci této stránky zahájit tvorbu nového fermanu při stisknutí záložky *Tvorba fermanu*.



Obrázek 5.13: Podoba stránky *Úprava fermanu*.

Kapitola 6

Testování

Pro ověření správného fungování aplikace byly zvoleny celkem dva přístupy testování: automatizované testování a uživatelské testování.

6.1 Automatizované testování

V rámci automatizovaného testování byla vytvořena řada testů pro ověření správnosti kódu na úrovni jednotlivých API endpointů.¹ Tyto testy mohou být spouštěné automaticky po každém zásáhu do kódu aplikace, aby bylo zaručeno, že nově provedené změny neovlivnily správné fungování aplikace. Kromě testování korektních odpovědí systému na validní požadavky byly testovány i odpovědi na požadavky nevalidní.

Automatizované testy byly napsány jednotlivě pro každý modul a v rámci těchto modulů byly rozděleny pro větší přehlednost do dvou souborů:

- **test_urls.py**

Obsahuje testy testující rozluštění zadaného URL a odkázání na správnou *view* funkci.

- **test_views.py**

Obsahuje testy testující správnost logiky *view* funkcí, které představují jednotlivé API endpointy.

Pro implementování testů byla nejprve vytvořena kopie stávající databáze, která je využívána čistě pro potřeby testování. Dále byly implementovány pro každý model jednotlivé třídy, které dědí od třídy *TestCase*, kterou poskytuje využívaný framework Django v rámci balíčku *django.test*.^[34] Lze tak využít užitečné testovací nástroje pro simulaci akcí skutečného uživatele či řadu *assertion*² metod, pro správné porovnání nejrůznějších navracených hodnot.

¹API je zkratka pro *Application Programming Interface*, které označuje rozhraní, které umožňuje dvěma softwarovým komponentám vzájemně komunikovat pomocí sady definic a protokolů. API endpoint je koncový bod v tomto komunikačním rozhraní. Patří mezi ně adresy URL serverů, služby a další specifická digitální místa, odkud jsou informace mezi systémy odesílány a přijímány.^[33]

²Assertion metody jsou funkce využívané zejména pro ověření správnosti jednotlivých částí kódu. Vyhodnocují pravdivost zadaného výrazu. Pokud je výraz pravdivý, neděje se nic. Pokud není, je vypsána chybová hláška na standartní chybový výstup.

```

class TestUrls(TestCase):
    def test_login_required(self):
        login_url = reverse('login:login_page')
        next_url = reverse('performances:default')
        response = Client().get(next_url)
        self.assertRedirects(response, f"{login_url}?next={next_url}")

    def test_default_func_resolved(self):
        url = reverse('performances:default')
        self.assertEqual(resolve(url).func, default)

```

Obrázek 6.1: Ukázka testu přesměrování na přihlašovací stránku při libovolné akci nepřihlášeného uživatele a testu rozluštění URL pro view funkci *default* v souboru *test_urls.py*.

```

def test_delete_existing_performance_valid(self):
    url = reverse('performances:performance_form_request')
    performance, role = self.create_test_performance(self.actor_employee.id)
    performance_count = TheatrePlay.objects.all().count()
    self.assertEqual(1, performance_count)
    response = self.client.post(url, data={'performance_id': performance.id,
                                          'action': 'delete'})
    self.assertEqual(response.status_code, 302)
    performance_count = TheatrePlay.objects.all().count()
    self.assertEqual(0, performance_count)

```

Obrázek 6.2: Ukázka testu smazání existujícího představení v souboru *test_views.py*

6.2 Uživatelské testování

Před zahájením uživatelského testování byla v databázi vytvořena řada ukázkových představení, směn, zaměstnanců či indispozic. Pomocí těchto předpřipravených dat byly pak lépe demonstrovány jednotlivé funkcionality systému.

Testování bylo provedeno zaměstnanci divadla, kteří mohli využít své specifické pracovní zkušenosti a poskytnout tak kvalitní zpětnou vazbu. Byli pro něj vybráni celkem čtyři zaměstnanci divadla Minor zastávající rozdílné pracovní pozice. Konkrétně Tereza Šachová jako inspektorka hlediště, Eliška Vystavělová jako uvaděčka, Kateřina Vystavělová jako barmanka a nakonec Kateřina Vorudová jako produkční divadla. Tímto jim děkuji za spolupráci.

Každému uživateli byl přidělen seznam úkolů, který byl sestaven na základě jejich pracovní pozice. Uvaděčka a barmanka sdílejí stejný seznam úkolů, protože je jejich pracovní náplň z hlediska ovládnání systému velice podobná.

Aplikace běžela na lokálním serveru a byla testována v prohlížeči Mozilla Firefox.

6.2.1 Inspektorka hlediště

Seznam úkolů:

1. Přihlašte se do systému jako uživatel **sachova** s heslem **divadlo123**.
2. Zjistěte své nejbližší nadcházející směny.
3. Nalezněte vypsané směny pro květen 2023 a přihlašte se na libovolnou z nich.
4. Zapište na libovolnou směnu uvaděče.
5. Změňte své aktuální heslo na **divadlo456** a znovu se přihlašte.
6. Přidejte novou poznámku na nástěnku.
7. Najděte kontakt na libovolného uvaděče.
8. Přidejte k libovolné směně poznámku (například o jejím průběhu).

Výsledky testování

Testování proběhlo úspěšně. Uživatel splnil všechny úkoly bez nejmenších obtíží. Po skončení testování byla pochválena práce se systémem, která je velice intuitivní, příjemná a i zábavná. Současně byl vznesen požadavek na filtr pro vyhledávání zaměstnanců podle jejich jména či pracovní pozice. Tento požadavek byl zařazen mezi funkcionality pro budoucí vývoj aplikace.

6.2.2 Uvaděčka, Barmanka

Seznam úkolů:

1. Přihlašte se do systému jako uživatel **vystavelova** s heslem **divadlo123**.
2. Zjistěte své nejbližší nadcházející směny.
3. Nalezněte vypsané směny pro květen 2023 a přihlašte se na libovolnou z nich.
4. Změňte své aktuální heslo na **divadlo456** a znovu se přihlašte.
5. Najděte kontakt na libovolného inspektora hlediště.
6. Prohlédněte si informace o libovolném představení.

Výsledky testování

Testování proběhlo úspěšně. Oba zaměstnanci neměli nejmenší problém s navigací v systému. Bylo pochváleno rozumné a přehledné rozložení jednotlivých komponent aplikace. Současně vznikla připomínka, že by bylo výhodné barevně rozlišit přihlašené a nepřihlášené směny. Daný požadavek byl vzápětí implementován.

6.2.3 Produkční divadla

Seznam úkolů:

1. Přihlašte se do systému jako uživatel **vorudova** s heslem **divadlo123**.
2. Přidejte nového zaměstnance.
3. Vytvořte u libovolného herce indispozici od 10. června 2023, 8:00 do 11. června 2023, 18:00. Poté si ji zobrazte.
4. Najděte kontakt na libovolného herce.
5. Vytvořte nové představení a nastavte mu libovolné parametry.
6. Vytvořte ferman pro červen 2023 a zasadte do něj tři libovolná představení. Poté z tvorby fermanu odejděte a znovu se k ní vraťte.
7. Zobrazte si nově vytvořené směny pro červen 2023 a zapište na její potřebné pozice odpovídající zaměstnance. Můžete přidat i poznámku specifickou pro tuto směnu.
8. Změňte své aktuální heslo na **divadlo456** a znovu se přihlašte.
9. Přidejte novou poznámku na nástěnku.

Výsledky testování

Testování proběhlo úspěšně. Systém udělal na uživatele dobrý dojem. Chválil intuitivní ovládání aplikace, vzhled stránek a shledal práci v rámci systému zábavnější než současnou podobu práce. Současně navrhl řadu užitečných funkcí, které by stávající systém doplnily. Jedná se o přidání poznámky k indispozici, zpětná kontrola správnosti fermanu při přidání nových indispozic či zamezení náhledu nedokončených fermanů všem neoprávněným uživatelům. Tyto funkce byly zařazeny mezi funkcionality pro budoucí vývoj aplikace. Po jejich doimplementování uživatel jevil skutečný zájem nasazení tohoto systému do divadla, ve kterém pracuje.

Kapitola 7

Závěr

V rámci práce byla provedena analýza správy umělecké činnosti divadla, na základě které byl navržen a implementován software, který tuto správu usnadňuje. V současné podobě software splňuje všechny potřebné požadavky pro nasazení do provozu reálného divadla.

Návrh započal definováním požadavků, případů užití a sestavením doménového modelu, na základě kterého byla vytvořena PostgreSQL databáze. Backend aplikace byl implementován v jazyce Python s použitím frameworku Django. Komunikaci s databází zajišťuje zmiňovaný framework, v rámci kterého lze naimplementovat modely jednotlivých databázových tabulek, které dohromady tvoří perzistentní vrstvu aplikace. Pro frontend aplikace bylo využito HTML, CSS, Javascript s knihovnamy jQuery a Selectize, Bootstrap a Font Awesome, které zdarma poskytuje úhledné ikony. Vývoj aplikace probíhal ve vývojovém prostředí PyCharm.

V průběhu práce jsem uplatnil znalosti z různých předmětů a současně jsem získával informace z internetu, zejména pak o vývoji backendu a frontendu, se kterým jsem neměl předchozí zkušenosti. Volil jsem proto takové technologie, které mají velkou uživatelskou základnu či nabízejí kvalitní dokumentaci. S výslednou volbou jsem byl spokojen.

7.1 Budoucí vývoj aplikace

Aplikaci lze v budoucnu rozšířit o další funkcionality, které přidají uživatelům nové možnosti a tím i ještě větší usnadnění jejich práce. V následujících sekcích budou zmíněny.

7.1.1 Automatické sestavení fermanu

Pro usnadnění vytváření měsíčních plánů lze implementovat funkci, která bude generovat nové fermany na základě těch historických. V rámci kapitoly 3 byla již provedena podrobná analýza, jak by bylo s využitím algoritmů umělé inteligence možné takovou funkci implementovat.

■ 7.1.2 Tvorba týdenních fermanů

Týdenní fermany slouží k doplnění podrobných informací k jednotlivým směnám v konkrétním týdnu. Tuto funkci sice aplikace již implementuje, ale ne formou týdenního fermanu. Některá divadla by tento náhled ovšem mohla preferovat.

■ 7.1.3 Exportování fermanu jako PDF či XML soubor

Fermany se po jejich vytvoření obvykle rozesílají ostatním divadlům pro kontrolu či ostatním zaměstnancům divadla emailem. Fermany jsou posílány obvykle jako PDF či XML soubory. Bylo by tak výhodné mít možnost exportovat hotový ferman jako soubor typu PDF či XML.

■ 7.1.4 Potvrzovací okna pro požadavky smazání

Funkce smazání nějakého objektu ze systému (zaměstnanec, směna, ferman...) jsou nevratné a bylo by tak vhodné zabezpečit kliknutí na dané tlačítko dialogovým okénkem pro potvrzení. Uživatelská kliknutí tak nebudou mít fatální důsledky.

■ 7.1.5 Filtrování zaměstnanců

Pro snazší vyhledávání jednotlivých zaměstnanců lze implementovat filtr, pomocí kterého budou uživatelům zobrazeni pouze zaměstnanci splňující zadané parametry (pracovní pozice či jména).

■ 7.1.6 Poznámka pro indispozici

Často je výhodné současně s indispozicí uvést i její důvod či jinak textem specifikovat její podobu. Je proto žádoucí možnost přidání poznámky ke každé indispozici.

■ 7.1.7 Kontrola fermanu po přidání nové indispozice

V případě přidání nové indispozice, která by narušila již vyhotovený ferman, je užitečné uživatele varovat o této situaci.

■ 7.1.8 Omezení náhledu nedokončených fermanů

Uživatelé, kteří se nestarají o vytváření fermanů, by neměli mít právo na náhled rozpracovaných či oficiálně nevydaných fermanů. Tyto fermany by byly zobrazeny ostatním uživatelům pouze po schválení od uživatelů starajících se o vytvoření fermanu.



Literatura

- [1] **ZLÁMALOVÁ, Anna.** *IS pro správu činnosti divadla.* České vysoké učení technické, Fakulta elektrotechnická, Katedra počítačů, 2018. Bakalářská práce. [Cit. 3.10.2022].
- [2] **JEŘÁBEK, Petr.** *Systém pro správu divadla.* České vysoké učení technické, Fakulta elektrotechnická, Katedra počítačů, 2021. Bakalářská práce. [Cit. 3.10.2022].
- [3] **Propared, LLC.** *How Propared Works.*[online]. Dostupné na: <https://www.propared.com/how-does-propared-work/> [Cit. 10.10.2022].
- [4] **MANSOFT.** *ABOUT THEATRON.*[online]. Dostupné na: <https://www.theatron.eu/about> [Cit. 10.10.2022].
- [5] **THEASOFT GMBH.** *Planning and managing system for artistic departments .*[online]. Dostupné na: <https://www.theasoft.de/en/products/theadispo/> [Cit. 10.10.2022].
- [6] **TUTORIALSPOINT.** *Genetic Algorithms - Introduction.*[online]. Dostupné na: https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_introduction.htm [Cit. 19.4.2023].
- [7] **TUTORIALSPOINT.** *Genetic Algorithms - Parent Selection.*[online]. Dostupné na: https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_parent_selection.htm [Cit. 19.4.2023].
- [8] **TUTORIALSPOINT.** *Genetic Algorithms - Crossover.*[online]. Dostupné na: https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_crossover.htm [Cit. 19.4.2023].
- [9] **TUTORIALSPOINT.** *Genetic Algorithms - Mutation.*[online]. Dostupné na: https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_mutation.htm [Cit. 19.4.2023].
- [10] **TUTORIALSPOINT.** *Genetic Algorithms - Survivor Selection.*[online]. Dostupné na: <https://www.tutorialspoint.com/>

- genetic_algorithms/genetic_algorithms_survivor_selection.htm [Cit. 19.4.2023].
- [11] **GEEKS FOR GEEKS**. *Introduction to Hill Climbing | Artificial Intelligence*. [online]. Dostupné na: <https://www.geeksforgeeks.org/introduction-hill-climbing-artificial-intelligence/> [Cit. 17.4.2023].
- [12] **THE APP SOLUTIONS**. *Functional vs non-functional requirements*. [online]. Dostupné na: <https://theappsolutions.com/blog/development/functional-vs-non-functional-requirements/> [Cit. 21.10.2022].
- [13] **ŠMOLÍK, David**. *Informační systém na rozúčtování nájemného a energií*. Dostupné na: <https://dspace.cvut.cz/bitstream/handle/10467/86235/F8-BP-2020-Smolik-David-thesis.pdf?sequence=-1&isAllowed=y> [Cit. 9.4.2023].
- [14] **HAJN, Pavel RNDr.**. *Úvod do datového modelování*. [online]. Dostupné na: https://is.muni.cz/el/fi/jaro2012/PV063/um/Uvod_datove_modelovani.pdf [Cit. 9.4.2023].
- [15] **DALY, Nicky**. *What is a Use Case?* [online]. Dostupné na: <https://www.wrike.com/blog/what-is-a-use-case/> [Cit. 21.10.2023].
- [16] **JETBRAINS**. *PyCharm Features*. [online]. Dostupné na: <https://www.jetbrains.com/pycharm/features/> [Cit. 6.4.2023].
- [17] **GEEKSFORGEEEKS**. *Back End Development*. [online]. Dostupné na: <https://www.geeksforgeeks.org/frontend-vs-backend/> [Cit. 7.4.2023].
- [18] **PYTHON SOFTWARE FOUNDATION**. *What is Python? Executive Summary*. [online]. Dostupné na: <https://www.python.org/doc/essays/blurb/> [Cit. 7.4.2023].
- [19] **DJANGO SOFTWARE FOUNDATION**. *Why Django?* [online]. Dostupné na: <https://www.djangoproject.com/start/overview/> [Cit. 7.4.2023].
- [20] **BIG, Nige**. *Mastering Django: Structures* [online]. Dostupné na: <https://masteringdjango.com/django-tutorials/mastering-django-structure/> [Cit. 7.4.2023].
- [21] **POSTGRESQL**. *What is PostgreSQL?* [online]. Dostupné na: <https://www.postgresql.org/about/> [Cit. 7.4.2023].
- [22] **Python Django API**. *Django Built-In DB Tables: The Complete Explanation*. [online]. Dostupné na: <https://pythondjangorestapi.com/django-built-in-db-tables-the-complete-explanation/> [Cit. 9.4.2023].

- [23] **DJANGO SOFTWARE FOUNDATION** *Making queries*. [online]. Dostupné na: <https://docs.djangoproject.com/en/4.1/topics/db/queries/> [Cit. 7.4.2023].
- [24] **GEEKSFORGEEKS** *Front End Development*. [online]. Dostupné na: <https://www.geeksforgeeks.org/frontend-vs-backend/> [Cit. 7.4.2023].
- [25] **W3C**. *HTML and CSS*. [online]. Dostupné na: <https://www.w3.org/standards/webdesign/htmlcss> [Cit. 7.4.2023].
- [26] **DJANGO SOFTWARE FOUNDATION**. *The Django template language*. [online]. Dostupné na: <https://docs.djangoproject.com/en/4.2/ref/templates/language/> [Cit. 10.4.2023].
- [27] **MDN WEB DOCS**. *What is JavaScript?* [online]. Dostupné na: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript [Cit. 7.4.2023].
- [28] **JQUERY**. *What Is jQuery?* [online]. Dostupné na: <https://jquery.com/> [Cit. 7.4.2023].
- [29] **SELECTIZE**. *Getting Started with Selectize*. [online]. Dostupné na: <https://selectize.dev/docs/intro> [Cit. 7.4.2023].
- [30] **W3SCHOOLS**. *What is Bootstrap?* [online]. Dostupné na: https://www.w3schools.com/bootstrap5/bootstrap_get_started.php [Cit. 7.4.2023].
- [31] **FONT AWESOME**. *Font Awesome's welcome page*. [online]. Dostupné na: <https://fontawesome.com/> [Cit. 21.10.2022].
- [32] **DIVADLO MINOR, oficiální stránky**. *Náš repertoár*. [online]. Dostupné na: <https://www.minor.cz/repertoar> [Cit. 10.4.2023].
- [33] **AMAZON WEB SERVICES** *What Is An API (Application Programming Interface)?* Dostupné na: <https://aws.amazon.com/what-is/api/> [Cit. 5.5.2023].
- [34] **DJANGO SOFTWARE FOUNDATION** *Writing and running tests* Dostupné na: <https://docs.djangoproject.com/en/4.2/topics/testing/overview/> [Cit. 5.5.2023].

Příloha A

Obsah přiložených souborů

A.1 Zdrojový kód

Obsah adresáře byl stručně popsán následující strukturou. Ta popisuje pouze nejdůležitější soubory zdrojového kódu. Jedná se zejména o moduly aplikace, které obsahují kód pro backend i frontend jednotlivých logických částí aplikace.

```
bp - adresář projektu
├── bp - základní nastavení aplikace
├── employee - modul Zaměstnanci
├── ferman - modul Ferman
├── home - modul Úvodní stránka
├── indisposition_and_shift - modul Indispozice a směny
├── login - modul Přihlášení
├── my_profile - modul Můj profil
├── static_shared - sdílené statické soubory mezi moduly
├── templates - sdílené Django šablony mezi moduly
├── theatre_play - modul Představení
├── tools - pomocné funkce
├── create_testing_data.py - generování testovacích dat
├── logfile.log - LOG
└── manage.py - spouštěcí skript pro Django projekt
```