

MASARYKOVA UNIVERZITA
FAKULTA INFORMATIKY



System pro plánování schůzek ve firmě

BAKALÁŘSKÁ PRÁCE

Dan Makalouš

Brno, podzim 2019

MASARYKOVA UNIVERZITA
FAKULTA INFORMATIKY



System pro plánování schůzek ve firmě

BAKALÁŘSKÁ PRÁCE

Dan Makalouš

Brno, podzim 2019

Na tomto místě se v tištěné práci nachází oficiální podepsané zadání práce a prohlášení autora školního díla.

Prohlášení

Prohlašuji, že tato bakalářská práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Dan Makalouš

Vedoucí práce: Mgr. Luděk Bártek, Ph.D.

Poděkování

Děkuji vedoucímu této bakalářské práce Mgr. Ludkovi Bártkovi, Ph.D., za pomoc při psaní této bakalářské práce a za jeho věcné připomínky. Děkuji také doc. Mgr. Haně Rudové, Ph.D., za pomoc při vymýšlení a upravování algoritmu rozvrhování. A v neposlední řadě děkuji firmě HARTMANN - RICO a.s. a všem jejím členům, kteří umožnili této práci vzniknout.

Shrnutí

Tato bakalářská práce je představením intranetové aplikace na platformě Adode Experience Manager, která řeší problém plánování a rozvrhování pracovních schůzek v rámci sjezdu firmy HARTMANN - RICO a.s.

Klíčová slova

algoritmus, AEM, rozvrhování, plánování, webová aplikace

Obsah

1	Úvod	3
2	Současný stav plánování ve firmě	5
2.1	<i>Porovnání s již vyvinutými aplikacemi</i>	6
3	Požadavky na aplikaci	9
3.1	<i>Funkční požadavky</i>	9
3.2	<i>Nefunkční požadavky</i>	10
4	Analýza a návrh	13
4.1	<i>Případy užití</i>	13
4.2	<i>Model využití</i>	14
4.3	<i>Databázový pohled</i>	16
4.4	<i>Design</i>	16
5	Rozvrhování a algoritmy rozvrhování	21
5.1	<i>Konkrétní algoritmy</i>	23
6	Algoritmus pro rozvrhování schůzek	25
6.1	<i>Délky schůzek</i>	26
6.2	<i>Účelová funkce</i>	27
6.3	<i>Implementace algoritmu Hill climbing</i>	29
7	Popis implementace aplikace	33
7.1	<i>Správa místností</i>	33
7.2	<i>Správa schůzek</i>	35
7.3	<i>Uživatelská práva</i>	42
7.4	<i>Zobrazení rozvrhu</i>	44
8	Využité vývojové nástroje	49
9	Závěr	51
	Literatura	53
A	Seznam příloh	55

1 Úvod

Plánování a rozvrhování je náročný a obtížně algoritmizovatelný problém. Při rozvrhování dochází k přidělování úloh ke zdrojům v čase. Typickým příkladem rozvrhování je organizace aktivit v diáři. V takovém případě je zdroj jen jeden a tím je majitel diáře a úlohami jsou aktivity, které mají být do diáře vepsány.

Pro tyto typy úloh obecný algoritmus neexistuje a ani existovat nemůže, protože problém rozvrhování spadá do třídy NP-úplných problémů [1]. Tato problematika je podrobněji rozepsána v kapitole 5.

Cílem, který tato práce sleduje, bylo vytvořit dostatečně dobrý systém, aby lidem, kteří byli za sestavování rozvrhu doposud zodpovědní, ulehčil práci a markantně zmenšil penzum času, který musí jeho sestavování pravidelně věnovat. Systém prezentovaný tímto textem by měl umět získat od uživatelů informace, které dříve posílali prostřednictvím elektronické pošty, v takové formě, aby s nimi mohl systém pracovat a vygenerovat z nich co nejlepší rozvrh. Systém také musí nechat zaměstnance v rozvrhu provádět úpravy a měnit zadání parametrů, podle kterých se systém při generování rozvrhu řídí.

V následující kapitole představí tento text konkrétní problematiku, která bude v práci řešena. Také budou rozebrány důvody, kvůli nimž bylo potřeba vyvinout tuto konkrétní aplikaci. S tím souvisí i analýza možných dalších řešení a plánované využití aplikace.

Třetí kapitolou práce přechází k analytické části. Tato kapitola se zabývá požadavky na aplikaci, které byly na systém před vývojem kladeny. Požadavky jsou v této kapitole rozděleny na funkční a nefunkční, přičemž funkční z velké části určují, jakým způsobem bude probíhat komunikace systému s uživateli, a nefunkční se zabývají uživatelskými přístupy, požadovanými technologiemi a požadavky na jednotlivé části systému, jako jsou rychlost odpovědi nebo formát generovaného souboru.

Čtvrtá kapitola „Analýza a návrh“ zobrazuje, jakým způsobem byla aplikace vyvíjena. Na začátku kapitoly lze nalézt diagram případů užití, který shrnuje požadavky, kterým byla věnována kapitola druhá. Dále je v kapitole popsáno, jakým způsobem se bude aplikace používat. Aplikace bude sloužit především jako nástroj, takže je nutné uvést, který člověk bude s jakou částí aplikace pracovat v daný mo-

1. Úvod

ment. Následuje ERD a s ním spojený databázový pohled na aplikaci. Poslední část kapitoly je věnována designu aplikace.

Pátá kapitola se zabývá rozvrhováním a algoritmy rozvrhování. Nejdříve je rozvrhování obecně popsáno. Poté se v kapitole vysvětluje, jakými problémy se rozvrhování zabývá, jak jsou tyto problémy těžké a následně jsou popsány způsoby, kterými se tyto problémy řeší. Blíže jsou rozpracována řešení pomocí lokálního prohledávání, která jsou pro tento projekt ze všech přístupů nejrelevantnější.

Kapitola „Algoritmus pro rozvrhování schůzek“ úzce souvisí s kapitolou předchozí. Bude zde popsán způsob aplikace obecných principů a algoritmů na konkrétní problém plánování schůzek. Následně bude argumentováno, z jakého důvodu byl použit konkrétní algoritmus nebo kombinace algoritmů a především jaké jsou výhody zvoleného přístupu. Nakonec bude popsána konkrétní implementace zvoleného algoritmu doplněná pseudokódy.

Následuje kapitola „Popis implementace aplikace“. Zde je podrobně rozebrána funkcionality každé ze čtyř komponent, ze kterých se aplikace skládá. Jsou zde popsány a pomocí obrázků zobrazeny všechny situace, do kterých se uživatel může při používání dostat.

V poslední části textu jsou sepsány použité technologie s krátkým popisem obsahujícím, co daná technologie dělá a jaké má výhody či nevýhody v porovnání s alternativami. Za touto kapitolou následuje závěr, který krátce shrnuje, do jaké míry bylo zadání práce splněno a nastiňuje způsoby, kterými by mohl být systém v budoucnosti rozšířen.

2 Současný stav plánování ve firmě

Cílem této práce je zjednodušit systém plánování a rozvrhování schůzek na pravidelných sjezdech firmy Hartmann - RICO a.s. Úkony, které s plánováním těchto sjezdů souvisí, jsou získat a uložit informace o tom, kdo bude mít s kým schůzku, a následně podle získaných informací přiřadit jednotlivým schůzkám podle preferencí časy a místnosti. V úvodu práce bylo nastíněno, že algoritmizovat plánování není jednoduchý úkol, protože se musí zohlednit velké množství faktorů, které lidem při plánování přijdou přirozené a samozřejmé. V žádném případě si systém neklade za cíl plně nahradit práci, kterou do té doby ve firmě zastávali lidé, ale spíše být dobrým nástrojem, který těmto lidem ušetří spoustu času.

Funkce, kterou by měl systém v provozu zastávat, je přinejmenším dvojí. První, tou jasnější, je propočítání časů schůzek, vzhledem k daným požadavkům tak, aby vznikl přijatelný rozvrh pro všechny zúčastněné. Tou druhou, na první pohled méně zřetelnou, je schopnost přijímat a zaznamenávat informace od účastníků schůzky. Vzhledem k tomu, že cílem této práce je zjednodušit a usnadnit systém plánování, je tato druhá funkce stejně důležitá, ne-li důležitější než funkce první.

Pro plánování těchto sjezdů se ve firmě Hartmann - RICO a.s. doposud používalo následující řešení. Za sestavení rozvrhu byla odpovědná určitá skupina zaměstnanců. Ti, kdo měli zájem o schůzku, museli těmto zaměstnancům sdělit, jakou schůzku potřebují a jaké jsou jejich časové možnosti. To probíhalo většinou pomocí elektronické korespondence. Zájemci o schůzku typicky uváděli, s kým chtějí mít schůzku, jak dlouhou schůzku potřebují a případně, zda jsou k dispozici pouze v určitou část dne nebo určitou hodinu. Při přípravě rozvrhu museli zaměstnanci zodpovědní za výsledný rozvrh zohlednit všechny zaslané požadavky a podle nich vytvořit výsledný rozvrh. Takový přístup má své výhody i nevýhody. Výhodou tohoto přístupu je, že zaměstnanci jako lidé zvládnou z popisu uživatelů výborně extrahovat omezení na rozvrh a určit jejich důležitost. Ví, jaké preference rozvrh splnit bezpodmínečně musí a jaké rozvrh splní, jen když to bude možné. Na druhou stranu zásadní nevýhodou je množství času, které lidem trvá, než projdou všechny požadavky, než si s budoucími účastníky ujasní, co potřebují, a než výsledný rozvrh sestaví. Také se může stát,

že i když ví, jak by rozvrh měl vypadat, nemají výpočetní sílu, jakou disponuje algoritmus, a proto se jim dobrý rozvrh nepodaří sestavit.

Aplikace by měla především zbavit zaměstnance nutnosti sestavovat z požadavků kompletní rozvrh. Celý proces by se měl z velké části zautomatizovat, žádosti o schůzku by se neměly posílat e-maily, ale pomocí speciálního rozhraní pro to určeného. Základní model využití je:

- Žadatelé o schůzky pošlou své žádosti s parametry schůzek.
- Zaměstnanec zodpovědný za plánování nechá vytvořit rozvrh, který upraví podle svého uvážení.
- Tento zaměstnanec vytiskne výsledný rozvrh v předem daném formátu.

2.1 Porovnání s již vyvinutými aplikacemi

Před začátkem vývoje rozvrhovacího systému, kterým se tato práce zabývá, byla firmou Hartmann - RICO a.s. provedena analýza již existujících řešení pro plánování a rozvrhování. Cílem této analýzy bylo zjistit, zda není výhodnější využít již existující aplikaci nebo případně upravit existující řešení.

Ukázalo se, že problém, který má tato aplikace řešit, není zcela běžným problémem, s nímž se firmy setkávají. Ve většině případů potřebují společnosti pouze prostředí, kde si dávají zaměstnanci vědět, kdy budou mít s kým schůzku. Právě proto je možné nalézt velké množství aplikací poskytující právě tuto funkcionalitu. Mezi takové aplikace patří například:

- EMS [2] — plánovací systém stejnojmenné společnosti, který umožňuje rezervovat místnosti pro konání schůzek a rozesílat e-maily účastníkům. Je dostupný z více zařízení a poskytuje různé úrovně přístupu.
- YAROOMS [3] („Meeting Room Booking System“) — další ze softwarů, které umožňují plánovat a zobrazovat, jak je rozvrh zaplněn v reálném čase. Jako ostatní softwary tohoto typu umožňuje například správu místností.

Tyto již vytvořené systémy vůbec nepřicházely v úvahu, jelikož řeší jiný problém, než je třeba řešit, a žádná z těchto aplikací není jednoduše rozšiřitelná o požadovanou funkcionalitu. Všechny totiž počítají s tím, že se schůzkám zadá čas schůzky. Dále byly analyzovány optimalizační systémy, jež jsou schopny na základě zadaných parametrů navrhnout optimální nebo dostatečně dobré řešení. Systémy, které byly analyzovány, jsou:

- OptaPlanner [4] — software vyvinutý společností Redhat [5] sloužící k řešení optimalizačních problémů na základě jejich omezujících podmínek. Využívá rozličné optimalizační algoritmy, z nichž některé jsou rozebrány v kapitole 5. OptaPlanner ale není systém, kterým by se dal tento plánovač nahradit nebo rozšířit, aby splňoval zadané požadavky. Je to spíše nástroj pro vývojáře v programovacím jazyku Java k řešení optimalizačních úloh. V tomto projektu není využit ze 2 důvodů. Tím prvním je, že tento problém patří mezi jednodušší optimalizační problémy, a proto je zbytečné využívat tak silný nástroj. Druhým důvodem je, že OptaPlanner využívá vlastní prostředí pro zobrazování výsledků rozvrhování, a to nesplňuje požadavky na tento projekt.
- IBM CPLEX Optimization Studio [6] — komerční nástroj firmy IBM sloužící k řešení optimalizačních úloh pomocí matematického programování a optimalizací pomocí omezujících podmínek. Vytváří rozvrh na základě zadaných omezení a cíle optimalizace. Firma Hartmann - RICO a.s. se tento optimalizační nástroj rozhodla nepoužít.

3 Požadavky na aplikaci

Základním požadavkem, z něhož vyplynuly všechny ostatní, bylo vytvořit systém, který ušetří čas zaměstnancům plánujícím schůzky. Pevně dané byly pouze požadavky ukládané firmou, jež platí pro všechny intranetové aplikace v ní vyvíjené. Všechny ostatní požadavky byly projednány na osobních setkáních.

3.1 Funkční požadavky

Funkčních požadavků na tento projekt bylo velmi mnoho. Pro přehlednost jsou zde popsány odděleně po odstavcích pro jednotlivé části systému.

Do funkčních požadavků patří možnost **správy místností**, ve kterých se schůzky budou moci konat. Mělo by být možné místnosti přidávat, upravovat a odebírat. Některé místnosti mohou sloužit jako místo konání schůzek pouze za určitých podmínek. Například v soukromé kanceláři se může konat schůzka, jen pokud je majitel kanceláře mezi jejími účastníky. Při úpravě či přidávání místnosti bude administrátor moci specifikovat následující parametry:

- název místnosti,
- kapacitu místnosti,
- zda místnost disponuje projektorem,
- nutného účastníka schůzky.

V projektu také musí existovat systém **správy schůzek**. Uživatel zde bude schůzky schopen zobrazovat, přidávat, upravovat a odebírat. Při upravování nebo přidávání schůzky bude moci specifikovat následující parametry:

- téma schůzky,
- organizátora schůzky,
- délku schůzky,

3. POŽADAVKY NA APLIKACI

- účastníky schůzky,
- textové pole pro speciální požadavky.

Administrátoři budou mít k tomuto výčtu navíc zobrazené tyto možnosti:

- přidání speciální schůzky „Hlavní část“, které se účastní všichni účastníci sjezdu,
- potřeba projektoru,
- konání schůzky dopoledne/odpoledne,
- preferovaná místnost,
- čas začátku schůzky (plánované použití především pro schůzku „Hlavní část“),
- administrátoři budou mít také přístup k zobrazení a úpravě libovolné schůzky.

Uživatel si bude moci **zobrazit všechny schůzky**, které byly zadány do systému, a všechny schůzky, které jsou relevantní pro specifikovaného uživatele. Administrátor si navíc bude moci zobrazit i schůzky s neprázdným polem pro poznámky.

Součástí projektu bude také **zobrazení vygenerovaného rozvrhu**. Administrátor bude moci časy schůzek změnit podle toho, jak mu to více vyhovuje. S tímto souvisí také generování souboru k tisku, který bude odpovídat momentálnímu rozložení schůzek.

3.2 Nefunkční požadavky

Základními nefunkčními požadavky byly:

- Bude se jednat o intranetovou aplikaci veřejně nedostupnou z internetu.
- Při vytvoření nebo zrušení schůzky systém rozešle e-maily, které o této akci budou informovat všechny uživatele, kterých se tato schůzka týkala.

- Rozvrh vygenerovaný algoritmem půjde uživatelsky přívětivě měnit, ideálně přístupem „drag and drop“.
- Mezi schůzkami musí mít účastníci 5 minut na přesun.
- Operace vygenerování rozvrhu a export do formátu k tisku by měly vrátit uživateli odezvu maximálně do 2 minut. Všechny ostatní operace, především ty, které bude využívat běžný uživatel, musí reagovat téměř okamžitě, to znamená do několika sekund.
- Design aplikace musí korespondovat s ostatními intranetovými aplikacemi firmy Hartmann - RICO a.s.

Jedním ze základních požadavků a konečným výstupem této aplikace je vygenerovaný dokument. Součástí nefunkčních požadavků je i vzhled vygenerovaného souboru ve formátu Excel. Na každém řádku tabulky bude schůzka a na sloupcích nejdříve informace o schůzce (délka, účastníci, organizátor) a dále časová osa. Tento styl vychází z rozvrhů použitých v minulosti, viz Obrázek 3.1. Čas konání schůzky specifikuje umístění buňky s názvem místnosti, ve které se tato schůzka koná. Každá místnost v tomto souboru má pro přehlednost přidělenou svou barvu, kterou je buňka s místností vyplněna.

Projekt musí být součástí firemní struktury. Je tedy z důvodu vnitřní integrity firmy zapotřebí využívat určité technologie. Omezení na použité technologie jsou proto následující:

- Pro vývoj vnitřní části projektu („back end“) je nutné použít programovací jazyk Java 7.
- Pro vývoj prezentační části projektu („front end“) je nutné použít kombinaci jazyka Javascript, HTML a CSS.
- Aplikace bude přístupná přes platformu Adobe Experience Manager [7].
- Pro ukládání dat potřebných pro běh aplikace je nutné využít databázový systém Microsoft SQL Server s přístupem pomocí SQL server 2014 Management Studio [8].

3. POŽADAVKY NA APLIKACI

A	B	C	D	E	F	G
Téma	Svolává	Účastníci	Čas (minuty)	8:00	9:00	10:00
Politika kvality - refresh	M. Jurová	S. Řihošková	30	8:15 - 8:45 Mistnost Joseph C		
Resselleři Alza/Mall prodeje do třetích zemí	M. Mošová	J. Hraběová, P. Dvořák, M. Kubát, J. Svobodová	30	8:15 - 8:45 Mistnost Albert D		
Codexis	L. Schneider	E. Bútorová, A. Pavlisová, A. Kratochvíl, M. Katolický	15	8:40 - 8:55 zasedací místnost HR		
Hlavní část			50		9:00 - 9:50 Mistnost Paul+Ludwig (A+B) 9:00 - 9:50 Mistnost Paul+Ludwig (A+B)	
Proces objednávky a zásoba etiket pro produkty BODE	P. Fafílek	P. Havlíček, V. Černochová, D. Roučka	30			10:00 - 10:30 Mistnost Albert D 10:00 - 10:30 Mistnost Albert D
Aktivace investic	P. Fuchs	Kaspar, J. Bartošková, B. Slovák, R. Šikula, O. Krejza, M. Skřivánek, P.	45			10:00 - 10:45 Mistnost Florence (GR) 10:00 - 10:45 Mistnost Florence (GR)
Statistiky a prezentace nových možností a změn pro rok 2019	T. Groh	Gutyan, V. Halama (nepřítomen), M. Šefl, M. Mošová, P. Šedivec, B.	45			10:00 - 10:45 Mistnost Ludwig B 10:00 - 10:45 Mistnost Ludwig B

Obrázek 3.1: Rozpis schůzek — leden 2019. Formát buněk je pozměněn kvůli čitelnosti textu.

4 Analýza a návrh

Projekt se bude skládat z komponent, přičemž každá bude plnit určité požadavky kladené na systém. Vzhledem k funkčním požadavkům bude mít projekt 4 následující komponenty.

- Správa místností — bude sloužit k vytváření, upravování a mazání místností určených pro schůzky. Součástí komponenty bude také zobrazení již vytvořených místností.
- Správa schůzek — v této komponentě se budou vytvářet, upravovat a mazat schůzky. Při zobrazování vytvořených schůzek zde bude možnost zvolit si člověka, jehož schůzky budou zobrazeny.
- Uživatelská práva — zde bude zobrazeno, kdo má administrátorská práva, a bude zde možnost přidat a odebrat administrátory.
- Zobrazení rozvrhu — tato komponenta bude zobrazovat rozvrh vygenerovaný algoritmem tak, aby v něm bylo možné dělat změny a také zde bude možnost generovat výsledný rozvrh, viz Obrázek 3.1.

4.1 Případy užití

Z požadavků kladených na aplikaci vyplývá nutnost řízení přístupu uživatelů k jednotlivým komponentám systému. Systém bude rozeznávat dva typy uživatelů, a to běžné uživatele a administrátory. K uchovávání informací o tom, kdo je administrátorem, bude sloužit komponenta Uživatelská práva.

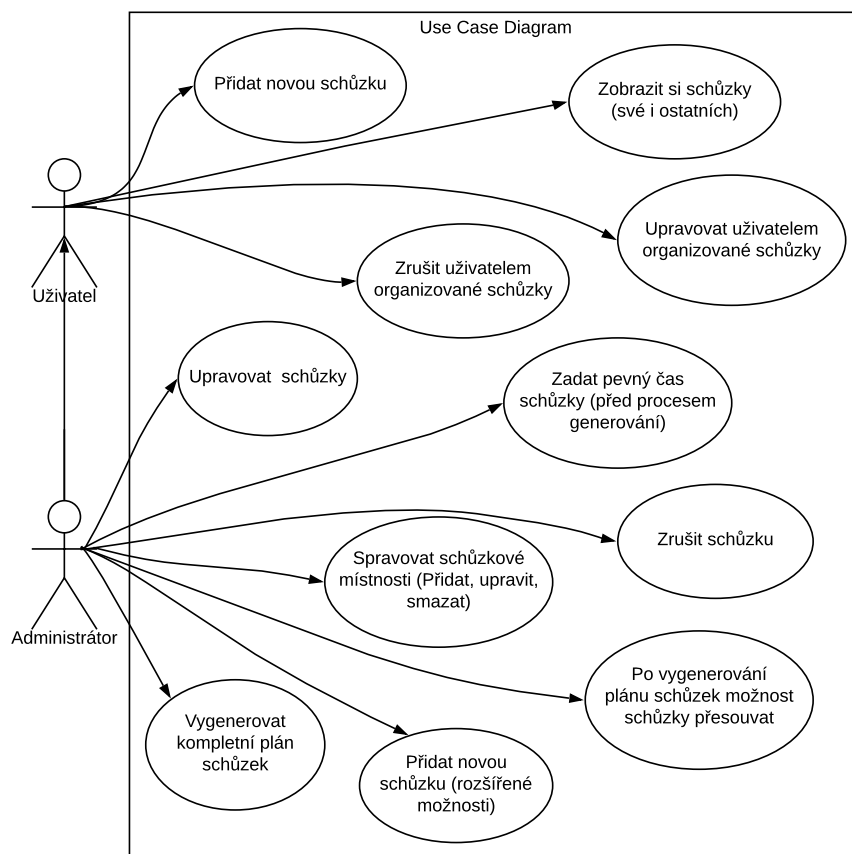
Na use case diagramu 4.1 lze vidět, že administrátor dědí od běžného uživatele, tudíž může v systému provádět všechny operace, které může provádět běžný uživatel. Běžný uživatel bude mít přístup k vytváření a zobrazování schůzek. Při vytváření bude mít nějaké možnosti nedostupné a při zobrazování si nebude moci nechat zobrazit všechny schůzky s poznámkou (je mu nepřístupný tento filtr). Administrátor bude mít v těchto akcích rozšířené možnosti (např. možnost dopoledne/odpoledne) a navíc bude moci pracovat s ostatními komponentami, ve kterých bude moci upravovat přístupová práva ostatních uživatelů,

nechat vytvořit rozvrh, upravovat vytvořený rozvrh, přidávat, upravovat a odebírat místnosti a vygenerovat výsledný rozvrh..

4.2 Model využití

Následující výčet shrnuje, jakým způsobem by se aplikace měla v plném provozu využívat.

- Vytvoření místností administrátory — administrátoři musí pomocí komponenty Správa místností zadat do systému místnosti, které mají k dispozici pro schůzky. Při vytváření každé místnosti budou specifikovat základní údaje o místnosti — název, kapacitu, zda místnost obsahuje projektor, a případně zaměstnance, který musí být přítomen u schůzky v dané místnosti (osobní kanceláře).
- Zadávání schůzek běžnými uživateli — běžní uživatelé budou pracovat pouze s komponentou Správa schůzek, v níž budou vytvářet schůzky. Schůzka se bude vytvářet zadáním základních parametrů schůzky (téma, délka, účastníci), přičemž účastníci schůzky svou účast nijak nepotvrzují, ani se ze schůzky nemohou odhlásit. Podle zadavatele tato funkcionality nebude v systému nutná. Organizátoři schůzek budou moci své schůzky průběžně upravovat a odebírat. Součástí vytváření schůzky bude také pole pro speciální požadavky, kam mohou uživatelé psát své preference pro schůzku — např. konání schůzky dopoledne/odpoledne, potřeba projektoru atd.
- Úprava administrátory — ve fázi po zasílání požadavků uživateli musí administrátor projít všechny schůzky, ve kterých jsou nějaké speciální požadavky, a tyto speciální požadavky vyřešit. Administrátoři mají u vytváření a upravování schůzek několik možností navíc, jmenovitě to jsou: potřeba projektoru, konání dopoledne/odpoledne, přesný čas a preferovaná místnost. Pro vyhledání všech schůzek s poznámkou bude v komponentě speciální vyhledávací tlačítko dostupné pouze administrátorům. Administrátor si tedy nechá zobrazit všechny schůzky s poznámkou, každou z nich projde, doplní schůzku svými rozšířenými možnostmi a smaže poznámku.



Obrázek 4.1: Use case diagram plánovacího systému.

- Zobrazení rozvrhu — po úpravě schůzek si administrátor nechá vygenerovat rozvrh v komponentě Zobrazení rozvrhu. V tomto rozvrhu lze jednotlivé schůzky přesouvat, ale zde už systém nehlídá, zda administrátor nepřesune schůzku tak, že vznikne konflikt na účastnících.
- Vygenerování výsledného rozvrhu — pokud je administrátor spokojený se zobrazeným rozvrhem, ať už po jeho úpravě nebo bez ní, nechá si vygenerovat soubor připravený k tisku, viz Obrázek 3.1.

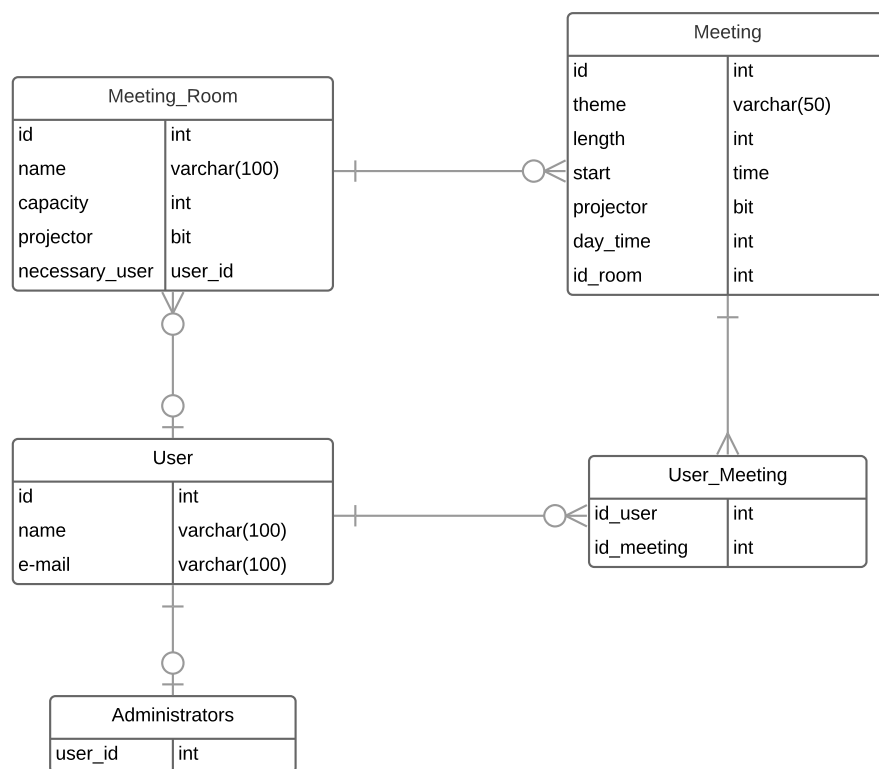
4.3 Databázový pohled

ERD na Obrázku 4.2 odpovídá databázovému pohledu aplikace. Aplikace pracuje s pěti databázovými tabulkami, z nichž jedna je tabulkou propojovací. Jelikož je v databázovém systému, který firma Hartmann - RICO a.s. využívá, tabulka User neměnná, model nemůže mít měnný atribut „is_administrator“. Proto vzniká vztah 1:1 mezi tabulkami User a Administrators. V této databázové tabulce se uchovávají informace, který uživatel má administrátorská práva. Tabulka Meeting obsahuje informace o vytvořených schůzkách a tabulka Meeting_Room drží informace o uložených místnostech. Tabulka User_Meeting je tabulkou propojovací, jenž zajišťuje, aby mezi tabulkami User a Meeting nevznikla vazba N:N. Jeden uživatel se totiž může účastnit více schůzek a schůzka má více uživatelů.

4.4 Design

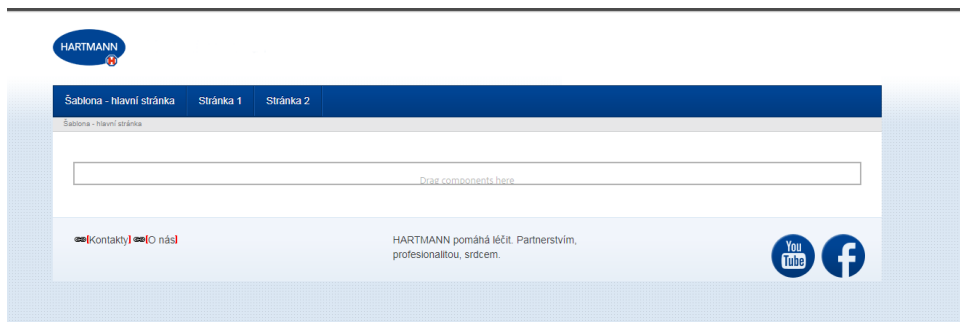
Mezi nefunkčními požadavky na aplikaci je, že design aplikace musí souhlasit s designem ostatních intranetových aplikací firmy Hartmann - RICO a.s. Všechny tyto aplikace jsou navrženy dle následujících pravidel.

- Aplikace se skládají z jednotlivých komponent zasazených do předem daného návrhu, který lze vidět na Obrázku 4.3.
- Pro výčet položek se používá jQuery plug-in DataTables [9].



Obrázek 4.2: Entity-relationship diagram plánovacího systému.

4. ANALÝZA A NÁVRH



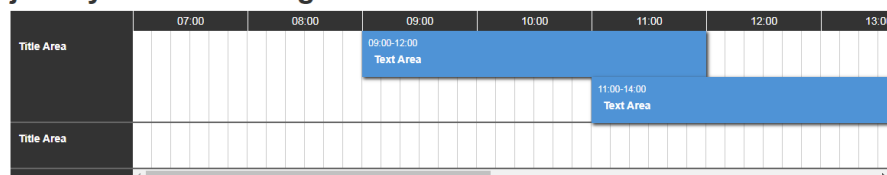
Obrázek 4.3: Šablona intranetových aplikací v Hartmann - RICO a.s.

- Styly tlačítek, nadpisů a ostatních prvků stránky, například volby plug-inů, souhlasí s designem prostředí, do kterého jsou komponenty zasazovány. Je například nevhodné používat červená tlačítka, když design je v modré a bílé.
- Pro přívětivější vzhled a pro účely responzivity je použita front-endová knihovna Bootstrap [10].

Komponenty Správa místností, Správa schůzek a Uživatelská práva všechny zobrazují informace a mají možnost tyto informace určitým způsobem upravovat. Pro zobrazení informací bude využita výše zmíněná knihovna DataTables a pro ostatní možnosti (přidat, odebrat, upravit) budou použita tlačítka v modré nebo černé barvě. Při přidávání a upravování se otevře běžný dialog v bootstrapovém designu, do kterého bude uživatel moci zadat potřebné informace.

Pro zobrazení rozvrhu bude použit volně dostupný jQuery plug-in Simple daily schedule [11]. Tento plug-in byl vybrán, protože vzhledově zapadá do motivu intranetových stránek firmy Hartmann - RICO a.s., viz Obrázek 4.4. Splňuje také všechny požadavky, které byly na rozvrh kladeny. Zobrazuje schůzky v esteticky přívětivém formátu, schůzky lze přesouvat na jiná místa a je možné získat zpět pozice jednotlivých schůzek v rozvrhu. Ostatní uvažované rozvrhy, např. Zoomable & Scrollable Schedule [12], měly velmi často možnost plánování na více dní, což je pro účely tohoto projektu zbytečné.

jQuery Schedule Plugin Demo



Obrázek 4.4: Daily schedule demo. Zdroj: [11].

5 Rozvrhování a algoritmy rozvrhování

Rozvrhování je proces, při kterém dochází k soupeření úloh o zdroje v čase, a rozvrh je dán umístěním úloh do konkrétního času a na konkrétní zdroje, kde mají být úlohy prováděny [13].

S problémem rozvrhování souvisí omezení. Omezení jsou několika druhů, v tomto problému jsou rozeznávána omezení pevná, to jsou omezení, která musí být splněna, aby byl rozvrh přijatelný, a omezení měkká neboli preference, což jsou omezení, která být nutně splněna nemusí. Rozvrh je úplným, pokud obsahuje všechny úlohy, a konzistentním, pokud jsou splněny všechny omezující podmínky [13].

Při hledání optimálního řešení, tedy nejlepšího možného přidělení úloh ke zdrojům v daném čase, lze využít několika přístupů nebo jejich kombinací. Blíže budou popsány dva základní přístupy, které se při hledání řešení používají. Optimálnímu řešení se běžně algoritmy snaží přiblížit pomocí heuristik a postupného zlepšování rozvrhu. Žádný lepší postup pro problémy tohoto typu neexistuje, protože problém rozvrhování obecně spadá do třídy složitosti NP-úplných problémů [14]. To jsou problémy, které lze řešit nedeterministickým algoritmem v polynomiálním čase a libovolný jiný NP problém lze na tento problém redukovat [15]. Důkaz pomocí redukcí ze známých NP-úplných problémů lze nalézt v odkazovaném zdroji [16]. Nedeterministický algoritmus se od deterministického liší tím, že umožňuje volby tam, kde musí deterministický algoritmus propočítat všechny možnosti.

Zatím není dokázáno, že se třída složitosti NP nerovná třídě složitosti P, tedy problémům, které lze řešit deterministickým algoritmem v polynomiálním čase. To znamená, že algoritmus, který by obecně řešil problém rozvrhování, by buď dokazoval, že $P=NP$, nebo by pracoval v exponenciálním čase, který je pro praktické prostředí nepoužitelný.

To, že obecný problém rozvrhování je NP-úplný problém, znamená, že neexistuje problém rozvrhování, u kterého lze nalézt optimální řešení v polynomiálním nebo dokonce lineárním čase. Je tedy nutné před řešením každého problému rozvrhování usoudit, jak moc je problém těžký a jak silný nástroj je potřeba k jeho řešení použít.

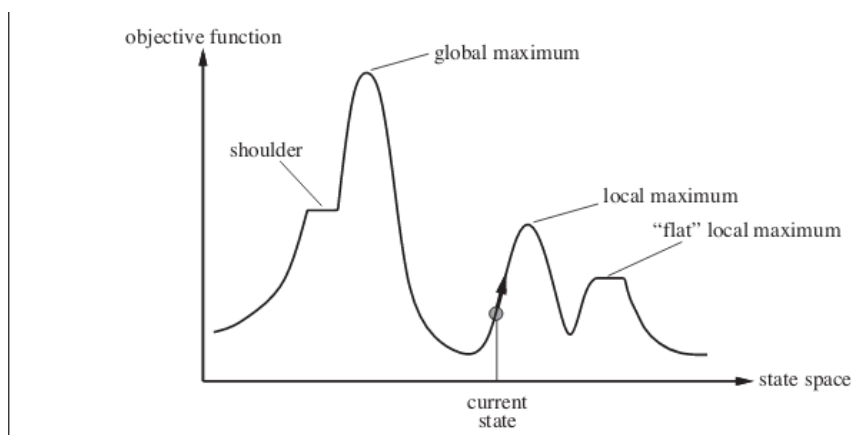
Pokud je problém dostatečně jednoduchý, lze optimální nebo dostatečně dobré řešení nalézt pomocí konstruktivního přístupu a řídicích

pravidel. Myšlenka algoritmu je vcelku jednoduchá, začne se s prázdným rozvrhem, který se postupně zaplňuje pomocí řídicích pravidel. Řídicí pravidla určují, v jakém pořadí mají být úlohy na jakých zdrojích vykonávány. Časté přístupy jsou: přidělování úloh podle jejich obtížnosti, přidělování úloh podle délky provedení úlohy, přiřazování úloh podle toho, jaká úloha byla zadána první, atd. Jednoduchým modelem na demonstraci je problém rozvrhování obsluhy zákazníků v obchodě s jednou pokladnou. Zde se použije pouze jedno řídicí pravidlo a to, že zákazníci budou obslouženi v pořadí, ve kterém přijdou [13] [15].

Pro úlohy rozvrhování, na které konstruktivní přístup nestačí, jsou dobrým nástrojem algoritmy lokálního prohledávání. Tyto algoritmy využívají k získání optimálního nebo dostatečně dobrého rozvrhu minimalizaci či maximalizaci účelové funkce. Účelová funkce je funkce, která spočítá, jak dobré je současné řešení rozvrhu. Typicky je volena vzhledem k omezením na rozvrh. Příkladem může být minimalizace zpoždění, férovost zatížení zdrojů, co nejrychlejší dokončení. Pokud dojde k úplné minimalizaci (maximalizaci) účelové funkce, tedy nemůže existovat rozvrh, který má lepší výsledek účelové funkce, lze o rozvrhu říct, že je optimální.

Při optimalizaci účelové funkce se hledají sousední řešení, která se berou jako potenciální náhradníci řešení současného. Jednotlivé implementace lokálního prohledávání jsou popsány v následujících podkapitolách.

Jak bylo již řečeno, algoritmy lokálního prohledávání pracují již s vytvořeným rozvrhem a pouze hledají sousedy tohoto řešení a zjišťují, zda sousední řešení není lepší než původní. Jelikož je pro tento algoritmus nutný již existující rozvrh, který může následně pozměňovat, je tento přístup velmi často kombinován s přístupem konstruktivním. Tedy nejdříve je konstruktivním přístupem vytvořen nějaký rozvrh, který může a nemusí být rozvrhem finálním, a poté se pomocí nějakého algoritmu lokálního prohledávání zkouší, zda může být tento rozvrh vylepšen [15].



Obrázek 5.1: Hill climbing, globální a lokální maxima. Zdroj: [17].

5.1 Konkrétní algoritmy

Algoritmus Hill climbing je nejjednodušší implementace lokálního prohledávání. Algoritmus postupně prochází všechny sousedy současného řešení a pro každé sousední řešení provede kontrolu, zda toto řešení nemá lepší výsledek účelové funkce než řešení současné. Pokud ano, nahradí současné řešení za nově nalezené a opakuje proces. Algoritmus Hill climbing má ale jednu zásadní nevýhodu, kterou demonstruje Obrázek 5.1. Touto nevýhodou je, že při snaze maximalizovat účelovou funkci se může stát, že se algoritmus ocitne v tzv. lokálním maximu. Nebude tedy z tohoto bodu možné najít žádné sousední řešení, které by bylo lepší, a proto se ke globálnímu maximu nemůže dostat. Tuto nevýhodu se další algoritmy, které budou v této kapitole zmíněny, snaží odstranit [15].

Tabu prohledávání se problém algoritmu snaží řešit možným akceptováním horšího řešení a tabu seznamem. Algoritmus se tedy chová stejně jako algoritmus Hill Climbing do momentu, ve kterém nelze nalézt lepší řešení mezi sousedními řešeními. Algoritmus Hill Climbing v tento moment skončí, ale Tabu prohledávání akceptuje zhoršující řešení a udržuje si seznam předchozích stavů, aby se v dalším kroku nevrátil zpět do lokálního minima nebo maxima [15].

Algoritmus Simulované žíhání stejně jako Tabu prohledávání hledá nejlepší možné řešení tak, že se na rozdíl od algoritmu Hill climbing vyhýbá uvíznutí v lokálním minimu nebo maximu. Simulované žíhání je inspirováno chladnutím kovů, ve kterém jsou částice kovu analogií k úlohám v rozvrhu. Při vysokých teplotách je kov kapalný, jeho částice jsou uspořádané náhodně a v jejich uspořádání probíhají velké změny. Při teplotách nízkých už se uspořádání částic nemění, protože kov je v pevném skupenství. Algoritmus simulovaného žíhání nastaví počáteční teplotu, kterou po určitém počtu nahrazení řešení za řešení sousední sníží. Při vysokých teplotách algoritmus dovolí akceptovat o hodně horší řešení a při nízkých už akceptuje pouze zlepšující. Algoritmus končí při dosažení předem stanovené teploty. Právě tím, že algoritmus dovoluje akceptovat řešení horší, se algoritmus snaží vyhnout uvíznutí v lokálním minimu nebo maximu a tím zvětšuje svou šanci na nalezení globálně nejlepšího řešení [15].

6 Algoritmus pro rozvrhování schůzek

Jak je v předchozích kapitolách popsáno, rozvrhování je přiřazení úloh zdrojům v konkrétním čase. V tomto případě jsou zdroje místnosti a úlohy jsou jednotlivé schůzky. Z požadavků na aplikaci lze určit i omezení a preference tohoto problému rozvrhování. Pevná omezení následující:

- Rozvrh musí být bez konfliktu na účastnících schůzek (tj. účastník má více schůzek najednou).
- Nesmí se stát, že se schůzka bude konat v místnosti, která nemá dostatečnou kapacitu.
- Schůzka se nesmí konat v něčí kanceláři, pokud se majitel kanceláře schůzky neúčastní.
- Hlavní část (tj. schůzka, které se účastní všichni účastníci sjezdu) nesmí být v konfliktu s žádnou jinou schůzkou a musí začínat v předem daný čas.
- Mezi schůzkami musí být alespoň pětiminutová přestávka na přesun.

Systém je postaven na těchto preferencích:

- Schůzky, na kterých je potřeba projektor, se musí konat v místnostech s projektorem.
- Schůzky, u kterých je specifikována denní doba, se musí konat v tuto denní dobu.
- Schůzky, u kterých je specifikován čas (kromě „Hlavní části“, která je v pevných omezeních), musí začít v tuto stanovenou dobu.
- Schůzky, které mají specifikovanou preferovanou místnost schůzky, se musí konat v této místnosti.
- Poslední schůzka musí skončit tak brzy, jak to jen bude možné.

K tomu, aby bylo možné zvolit konkrétní algoritmus, je nejdříve nutné si uvědomit, jak je konkrétní problém těžký, neboli jak silný nástroj potřebujeme použít k tomu, abychom dosáhli požadovaného výsledku. Tento problém rozvrhování schůzek do místností patří spíše k těm jednodušším, protože sjezd, pro který se tato aplikace vytváří, je tvořen průměrně 24 schůzkami, které jsou průměrně 60 minut dlouhé. K dispozici je 7 schůzovních místností a 2 kanceláře. Jelikož je k dispozici takový počet místností, jen zřídka se stane, že by se schůzka hodila na nějaký čas a nebyla by k dispozici místnost, kam ji umístit.

Z těchto důvodů není potřeba využívat sofistikované algoritmy, jako je algoritmus Tabu seznamy nebo algoritmus Simulované žíhání, a pro tento problém úplně stačí jednoduchý algoritmus Hill climbing kombinovaný s iniciálním rozvržením pomocí řídicích pravidel.

Při vytváření iniciálního rozvrhu jsou místnosti zaplňovány podle toho, jak obtížné je tyto místnosti obsadit. Nejdříve algoritmus zkouší obsazovat místnosti, které jsou zároveň kanceláře, protože v nich se mohou konat jen takové schůzky, kterých se účastní majitel kanceláře. Dále se pokouší zaplňovat místnosti vzestupně podle jejich kapacity. Tím je zajištěno, že schůzky, které se vejdou pouze do velkých místností, se v nich skutečně budou konat. Tím se zabrání tomu, aby velké místnosti obsadily schůzky, které se mohly konat v místnostech menších nebo v kancelářích.

Při výběru schůzky, která se bude jako další do rozvrhu vkládat, se vybírá podle toho, kolik má schůzka účastníků. Dá se předpokládat, že kdyby se algoritmus snažil tu největší schůzku přidat až nakonec, bude mít na všech pozicích velké množství konfliktů. Obtížnost schůzky se tedy určuje pouze podle množství účastníků.

Při zaplňování rozvrhu postupuje algoritmus na časové ose. To znamená, že schůzka je umístěna na danou pozici pouze pokud neexistuje pozice, která je vhodná a v rozvrhu je dříve [14].

6.1 Délky schůzek

Aby bylo možné rozvrh upravovat a aby mohly algoritmy lokálního prohledávání správně pracovat, je nutné omezit délky schůzek tak, aby si mohly prohazovat pozice. Pokud by jedna schůzka trvala 60 minut

a druhá 50, tak je v rozvrhu nelze jednoduše prohodit a případným prohozením by byly ovlivněny okolní schůzky. Na rozdíl od školního rozvrhu, kde jsou všechny hodiny stejně dlouhé (dvouhodina jsou dvě stejně dlouhé hodiny po sobě), jsou zde délky schůzek velmi proměnlivé. Ideální scénář z pozice uživatelů je, že si mohou říct, jak přesně budou mít dlouhou schůzku. Čím proměnlivější ale délky schůzek budou, tím méně si jich bude moci vyměnit pozice.

Dalším problémem je, že když už budou uživatelé omezováni v tom, jaké délky schůzek si mohou zvolit, měly by tyto délky uživatelům vyhovovat. Dobrým řešením by bylo, kdyby nejkratší možná schůzka měla půlhodiny a délka schůzek se zvyšovala po půlhodinách. Potom by se daly v rozvrhu vyměnit tři půlhodinové schůzky za jednu devadesátiminutovou. Tomuto řešení ale brání požadavek, že mezi schůzkami musí být pauza alespoň 5 minut.

Podle zadavatele projektu je velmi oblíbená délka schůzky 45 minut, ale časté jsou i schůzky 15 minut dlouhé. Domluvou bylo tedy rozhodnuto, že nejkratší schůzka bude 20 minut dlouhá a za ní bude 5 minut dlouhá přestávka. Aby však tento 25 minut dlouhý blok mohl být prohozen, druhá nejkratší schůzka bude trvat 45 minut + 5 minut přestávka. Následuje schůzka dlouhá 70 + 5 minut atd. Zvažovala se i již zmíněná alternativa, že by délky schůzek byly po půlhodinách a účastníci schůzek by museli schůzky zkrátit tak, aby měli čas na případný přesun, ale nakonec byla vybrána varianta s pevně danou pětiminutovou přestávkou za cenu „nehezky“ délek schůzek.

To, že jsou délky schůzek takto omezeny, vytváří v rozvrhu časové sloty, do kterých mohou být schůzky umístěny. To je užitečné například při udělování penalizací za pozdní skončení schůzky.

6.2 Účelová funkce

Po vytvoření iniciálního rozvrhu, který se v tomto případě vyhýbá pouze konfliktům na lidech, algoritmus ohodnotí, jak je vytvořený rozvrh dobrý. K takovému ohodnocení slouží účelová funkce.

Účelová funkce je funkce, která pro vstup, kterým je stav, v němž se výpočet nachází, vrátí číslo, které závisí na omezeních pro daný problém. Využívá se v optimalizačních úlohách, ve kterých se hodnota této funkce minimalizuje či maximalizuje [14]. V tomto projektu je

využívána pro jednotlivé schůzky v rozvrhu. Tato funkce zhodnotí pro každou schůzku její pozici v rozvrhu a přiřadí jí příslušné ohodnocení. Kvalitu rozvrhu určuje součet ohodnocení všech schůzek.

Způsob, kterým účelová funkce přiřazuje jednotlivým schůzkám ohodnocení, je založen na základě omezení na rozvrh. Pokud schůzka porušuje nějaké z pevných omezení (například konflikt s jinými schůzkami), je jí přiřazena řádově větší penalizace než v případě, kdy schůzka porušuje některou z preferencí (například nedodržení času schůzky). Schůzky dříve umístěné v rozvrhu mají o trochu lepší ohodnocení než schůzky, které jsou v rozvrhu umístěny později. To je proto, aby měl sjezd tendenci skončit co nejdříve.

Účelová funkce v tomto systému prochází rozvrh pro každou místnost od začátku rozvrhu do konce a hledá, zda se v daný časový moment vyskytuje v prohledávané místnosti schůzka. Pokud ano, systém pro nalezenou schůzku kontroluje, zda splňuje zadaná pevná omezení a preference, a v případě porušení zvýší penalizaci rozvrhu podle pravidel popsaných v předchozím odstavci. Jak byly rozděleny penalizace v tomto projektu znázorňuje následující výčet:

- Algoritmus vůbec nedovolí přiřadit schůzku do něčí kanceláře, pokud se schůzky neúčastní. Tím pádem je toto kritérium penalizováno nejvíce.
- Hlavní část nezačíná v požadovaný čas: **100000**. Toto omezení je z důvodu implementace algoritmu nutno počítat. Je nastaveno na tak vysokou hodnotu, protože správný čas začátku je pro dostačující rozvrh nezbytným požadavkem.
- Jeden účastník musí být na více schůzkách zároveň: **1000 za každou konfliktní schůzku v jeden časový slot**. Penalizace je udělena pro každou konfliktní schůzku zvlášť, v případě konfliktu tří schůzek po dobu jednoho časového slotu je všem schůzkám udělena penalizace 1000, celková penalizace je tedy 3000.
- Počet účastníků schůzky převyšuje kapacitu schůzky: **800 za každou osobu navíc**.
- Schůzka, která žádala projektor, je v místnosti bez projektoru: **100**.

- Schůzka začíná v jiný čas, než bylo zadáno: **100**. Předpokládá se, že preference pro čas se bude vyplňovat jen velice zřídka, ale když už vyplněna bude, bude nejspíše důležité, aby schůzka v daný čas proběhla. Proto má tato položka penalizaci vyšší než následující dvě, které rovněž penalizují špatné umístění schůzky v čase.
- Schůzka se měla konat pouze odpoledne a nekoná: **80**.
- Schůzka se měla konat pouze dopoledne a nekoná: **60**. Penalizace za tuto položku je nižší než za položku předchozí, protože odpoledne je více pozic, na které může být schůzka umístěna.
- Schůzka je umístěna v jiné místnosti, než bylo preferováno: **40**. Jelikož je místností pro konání schůzek velký počet, není pravděpodobné, že bude tato penalizace často udělována.
- Každá schůzka také dostává časovou penalizaci podle toho, jak pozdě je schůzka v rozvrhu umístěna. Penalizace je **1 za každý časový slot**.

6.3 Implementace algoritmu Hill climbing

Jak je již rozebráno v kapitole 5, při implementaci tohoto algoritmu je nutné implementovat způsob, kterým se budou hledat sousední řešení. To je v tomto projektu řešeno pomocí přesouvání schůzek na prázdná místa a prohazování schůzek v daném rozvrhu. Sousední řešení je tedy v tomto případě rozvrh, ve kterém jsou prohozeny dvě oblasti, z nichž alespoň jedna byla neprázdná, tedy oblast obsahovala schůzku.

Na algoritmu Algorithm 1 je pseudokód znázorňující implementaci algoritmu Hill Climbing pro tento systém. Třetí řádek tohoto pseudokódu specifikuje ukončovací podmínku algoritmu. Tou je, že algoritmus probíhá, dokud došlo při předchozí iteraci ke zlepšení rozvrhu. Kvalitu rozvrhu určuje účelová funkce, viz podkapitola 6.2. Ta je v obou uvedených algoritmech zastoupena funkcí „getScheduleRating()“, kterou se získává aktuální ohodnocení rozvrhu.

V každé iteraci algoritmus nejdříve sestupně seřadí všechny schůzky dle jejich ohodnocení účelovou funkcí. Poté se podle tohoto pořadí

snaží pro každou schůzku najít lepší umístění v rozvrhu. Toto hledání je znázorněno pseudokódem Algorithm 2. Zde se pro všechna možná umístění schůzky počítá, zda je rozvrh po prohození dvou oblastí lepší nebo horší, a podle toho prohození zachová, nebo provede prohození zpětné. Možná umístění schůzky jsou takové pozice, které mají dostatečně dlouhý volný časový slot a zároveň jsou v místnosti, do které je možné schůzku umístit. Je také nutné hlídat, aby pro prohození nebyla vybrána oblast, která rozdělí nějakou schůzku na dvě nesouvislé oblasti, nebo aby se pro prohození vybrala oblast v místnosti, kam daná schůzka být umístěna nemůže. Tyto podmínky jsou v reálné implementaci v algoritmu kontrolovány na různých místech, ale v tomto pseudokódu jsou pro přehlednost všechny obsaženy ve funkci „isPossibleToSwap“ na řádce 5.

Pokud se pro schůzku podařilo najít lepší umístění, daná iterace končí a začíná další iterace. Znovu se tedy schůzky seřadí podle toho, jak jsou v daném rozvrhu konfliktní. Pokud se pro danou schůzku umístění vylepšit nepodařilo, tak se algoritmus snaží vylepšit další nejvíce konfliktní schůzku. Pokud nelze vylepšit pozici žádné ze schůzek, algoritmus končí. Znovu je důležité zdůraznit, že i když nelze vylepšit umístění žádné schůzky, nemusí se jednat a nejspíše se nejedná o rozvrh optimální, protože se algoritmus mohl dostat a nejspíše dostal do lokálního minima [15], viz Obrázek 5.1.

Algorithm 1 Algoritmus Hill Climbing

```
1: rating  $\leftarrow$  0
2: meetingList  $\leftarrow$  allMeetings
3: while rating  $\neq$  getCurrentRating() do
4:   meetingList.sortByRating()
5:   for all meetings do
6:     rating  $\leftarrow$  getCurrentRating()
7:     findBetterPlaceInSchedule(meeting)
8:     if rating  $\neq$  getCurrentRating() then
9:       break
10:    end if
11:  end for
12: end while
```

Algorithm 2 Hledání lepšího místa pro schůzku

```
1: meeting  $\leftarrow$  meetingToSwap
2: oldRating  $\leftarrow$  getScheduleRating()
3: for all timeSlots do
4:   for all meetingRooms do
5:     if isPossibleToSwap(meeting, timeSlot, room) then
6:       swap(meeting.timeSlot, meeting.room, timeSlot, room)
7:       newRating  $\leftarrow$  getScheduleRating()
8:       if newRating < oldRating then
9:         oldRating  $\leftarrow$  newRating
10:      else
11:        swap(meeting.timeSlot, meeting.room, timeSlot, room)
12:      end if
13:    end if
14:  end for
15: end for
```

7 Popis implementace aplikace

Tato intranetová aplikace se skládá ze 4 samostatných komponent, které budou v jednotlivých podkapitolách podrobně rozebrány. V této kapitole bude vysvětleno, jakým způsobem jsou řešeny funkční požadavky zadané na systém.

7.1 Správa místností

Komponenta Správa místností slouží k přidávání, upravování a mazání společenských místností a kanceláří, ve kterých je možné konat schůzky. Práva na zobrazení a upravování této komponenty mají pouze administrátoři. Na Obrázku 7.1 lze vidět hlavní stránku komponenty s tabulkou, která obsahuje informace o jednotlivých místnostech pro konání schůzek. U každé místnosti je specifikován název místnosti, kapacita místnosti a zda obsahuje projektor. Tyto atributy budou podrobněji rozebrány v následující podkapitole „Vytvoření nové místnosti“. Ve slupci „Možnosti“ jsou zobrazena tlačítka, přes která lze místnost upravovat nebo smazat.

Stisknutím tlačítka „Přidat“ na hlavní stránce komponenty Správa místností (viz Obrázek 7.1) se otevře dialogové okno pro vytvoření nové místnosti, které lze vidět na Obrázku 7.2. V tomto dialogovém okně lze specifikovat následující parametry.

- **Název** — název místnosti je pro tento systém jedinečný, a tedy zcela určující místnost. Toto pole je pro přidání nové místnosti nebo úpravy stávající místnosti povinné.
- **Kapacita** — toto pole slouží pro udání maximální možné kapacity místnosti. Jinak řečeno, kolik nejvíce lidí se může účastnit schůzky, která se bude v této místnosti konat. Toto pole je pro přidání nové místnosti nebo úpravy místnosti stávající rovněž povinné.
- **Barva místnosti** — tímto polem uživatel specifikuje, jakou barvu bude místnost mít ve výsledném rozvrhu.
- **Projektor** — zaškrťovací pole, které určuje, zda je v místnosti k dispozici projektor.

7. POPIS IMPLEMENTACE APLIKACE

Správa místností

Správa schůzek

Uživatelská práva

Zobrazení rozvrhu

Správa místností

Přidat

Zobraz 50 záznamů na stránku

Hledej

Název	Kapactia	Projektor	Možnosti
Kancelář MT	5	NE	
Kancelář TG	5	NE	
Místnost Albert	5	NE	
Místnost Florence	13	ANO	
Místnost Joseph	4	NE	
Místnost Ludwig	20	ANO	
Místnost Paul	20	ANO	
Místnost Viktor	3	NE	

Zobraz stránku 1 z 1

Předchozí 1 Další

Insert keyword

Center box

f You Tube

Obrázek 7.1: Správa místností, hlavní stránka.

- Podmíněný uživatel — toto pole značí osobu, která musí být na schůzce v této místnosti přítomná. Typicky používané pouze pro kanceláře, v nichž může být schůzka jen v případě, že se jí její majitel bude účastnit.

Pro úpravu parametrů přidané místnosti je nutné kliknout na ikonu „Upravit“ ve sloupci možnosti na daném řádku. Stisknutím této ikony se otevře dialogové okno (Obrázek 7.3), ve kterém jsou vyplněny informace o příslušné schůzce.

Pro smazání místnosti je nutné kliknout na ikonu „Smazat“ na daném řádku a následně potvrdit volbu v dialogu na Obrázku 7.4. Po smazání místnosti již nelze místnost obnovit a je ji potřeba přidat znovu.

7.2 Správa schůzek

Běžní uživatelé budou pracovat s aplikací pouze přes tuto komponentu. Mohou zde zadávat nové schůzky, upravovat a mazat schůzky jimi organizované a také si zobrazovat schůzky, které jsou pro ně na sjezd naplánovány.

Při otevření komponenty se otevře stránka zobrazená na Obrázku 7.5. V tabulce budou vypsány všechny schůzky, kterých se přihlášený uživatel účastní nebo které organizuje. Pomocí vyhledávacího pole a tlačítka „Vyhledat“ je možné vyhledat libovolného uživatele a zobrazit si jeho schůzky. Při kliknutí na toto tlačítko s prázdným vyhledávacím polem se zobrazí všechny schůzky. U každé schůzky budou 2 ikony — „Upravit“ a „Smazat“. Tyto ikony jsou u uživatelů, kteří nemají dostatečná práva, zašedlé (administrátoři mohou upravovat a rušit všechny schůzky, běžní uživatelé jen ty, které organizují). Tlačítko „Vyhledat s poznámkou“ je viditelné pouze pro administrátory a slouží k vyhledání všech schůzek, které mají napsaný text v poznámce. Největší tlačítko na stránce „Přidat schůzku“ slouží k přidání nové schůzky.

stránku Hledej

Přidat místnost ✕

Název

Kapactia

Barva místnosti

Projektor ☐

V této místnosti se schůzky musí účastnit (kancelář)

Uložit Zrušit

Obrázek 7.2: Správa místností, vytvoření nové místnosti.

schůzek Uživatelská práva Zobrazení rozvrhu Správa místností

Upravit Místnost

Název
Kancelář MT

Kapactia
5

Barva místnosti
▼

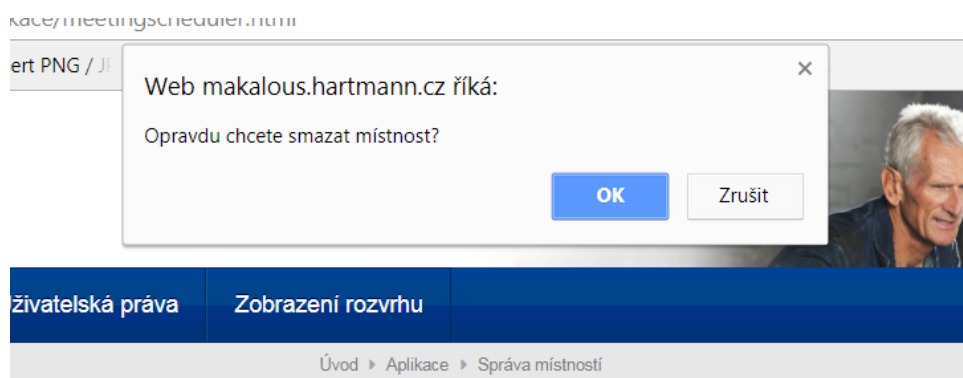
Projektor
☐

V této místnosti se schůzky musí účastnit (kancelář)
Marek Třeška (300394)

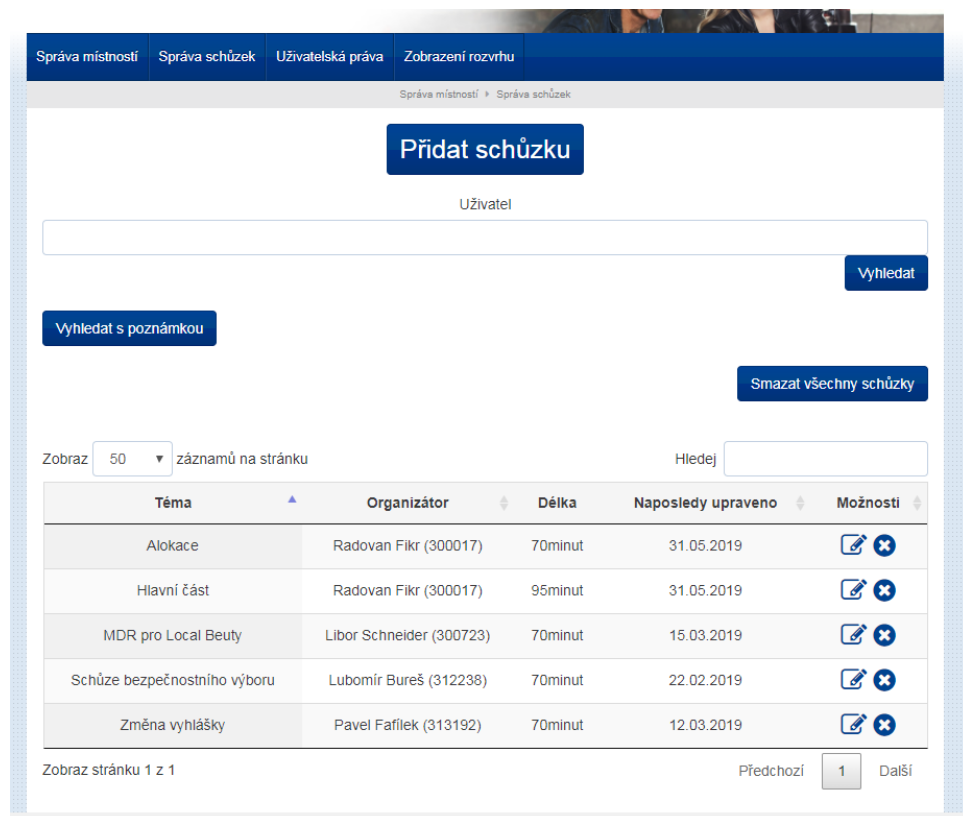
Uložit Zrušit

Obrázek 7.3: Správa místností, úprava místnosti.

7. POPIS IMPLEMENTACE APLIKACE



Obrázek 7.4: Správa místností, smazání místnosti.



Obrázek 7.5: Správa schůzek, hlavní stránka.

Na Obrázku 7.6 lze vidět dialog pro vytvoření schůzky v zobrazení pro běžné uživatele. Ti při vytvoření nové schůzky specifikují následující parametry:

- **Téma** — téma schůzky je název, pod kterým schůzka v systému vystupuje. Téma schůzky je jedinečné a lze tedy podle něj schůzku jednoznačně identifikovat. Pole je pro vytvoření nové schůzky povinné.
- **Organizátor** — toto pole udává organizátora schůzky. Při otevření dialogu je hodnota pole předvyplněna parametry přihlášeného uživatele. Organizátora lze změnit zadáním jména, příjmení nebo osobního čísla daného uživatele a pomocí našeptávače uživatele vybrat, viz Obrázek 7.7.
- **Délka schůzky** — tímto polem lze specifikovat délku vytvářené schůzky. Nelze zadat libovolnou délku, ale je nutné si vybrat z nabízených možností délek, se kterými umí systém pracovat. Pole je pro vytvoření schůzky povinné.
- **Účastníci** — zde lze specifikovat, kdo se vytvářené schůzky bude účastnit. Přidat dalšího účastníka lze pomocí pole „Přidat účastníka“ a s použitím našeptávače, který je vidět na Obrázku 7.7.
- **Specifické požadavky** — toto pole slouží pro doplnění dalších nároků na schůzku.

Dialog přidání nové schůzky je pro administrátory obohaceno o následující možnosti, viz Obrázek 7.8:

- **Projektor** — toto pole specifikuje, zda je pro schůzku potřeba projektor.
- **Část dne** — položka určuje, zda se schůzka může konat v průběhu celého dne, pouze dopoledne nebo pouze odpoledne.
- **Místnost** — v tomto poli lze z výběru přidanych místností zvolit preferovanou místnost pro konání schůzky.
- **Začátek schůzky** — v tomto poli lze z výběru všech možných časů začátků schůzky vybrat čas preferovaný.

Přihlášený uživatel
300279
Ladislav Králík

Přidat schůzku

Téma

Organizátor
Ladislav Králík (300279), HR IT PODPORA APLIKACÍ

Délka schůzky

Přidat účastníka

Přidat

Účastníci

Specifické požadavky

Uložit Zrušit

Obrázek 7.6: Správa schůzek, vytvoření nové schůzky.



Přidat schůzku

Téma

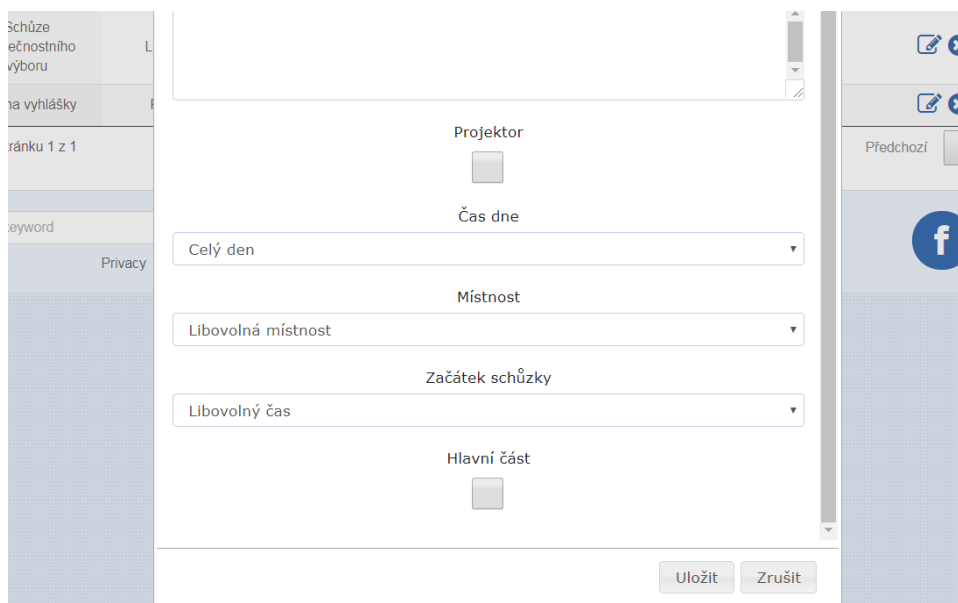
Organizátor

456

Jarmila Ivaniecová (304565), BCV SÁDROVÁ OBINADLA
 Helena Pavelková (304569), BCV ZAJIŠTĚNÍ KVALITY
 Jarmila Nováčková (312456), BVB D K1/2
 Miroslava Smejkalová (314456), BVB S VÝROBA VZORKŮ
 Dana Hvižďová (314566), BVB S VÝROBA VZORKŮ
 Zdeněk Cibulka (314568), HR IT SERVICE DESK

Přidat

Obrázek 7.7: Správa schůzek, našeptávač.



Projektor

Čas dne

Celý den

Místnost

Libovolná místnost

Začátek schůzky

Libovolný čas

Hlavní část

Uložit Zrušit

Obrázek 7.8: Správa schůzek, rozšířené možnosti.

- Hlavní část — každý sjezd má jednu schůzku „Hlavní část“. Tuto schůzku musí vždy přidat administrátor. U této schůzky se vždy specifikuje pole „Začátek schůzky“, to je však na zodpovědnosti administrátora, který tuto schůzku přidává. „Hlavní část“ je jako schůzka specifická tím, že se jí účastní všichni účastníci sjezdu a nemůže tedy ve stejný čas probíhat žádná jiná schůzka.

Jak již bylo řečeno, pokud má uživatel dostatečná práva, může schůzku upravovat nebo mazat. Mazání schůzky probíhá kliknutím na křížek u schůzky a potvrzením ve vyskakovacím dialogu. Kliknutím na druhou z ikon se otevře dialog editace, který je stejný jako dialog přidávání s tím rozdílem, že při editaci budou již vyplněna všechna pole stejně, jak byla vyplněna při přidávání nebo poslední editaci. Stisknutím tlačítka „Uložit“ se přepíší původně uložené údaje.

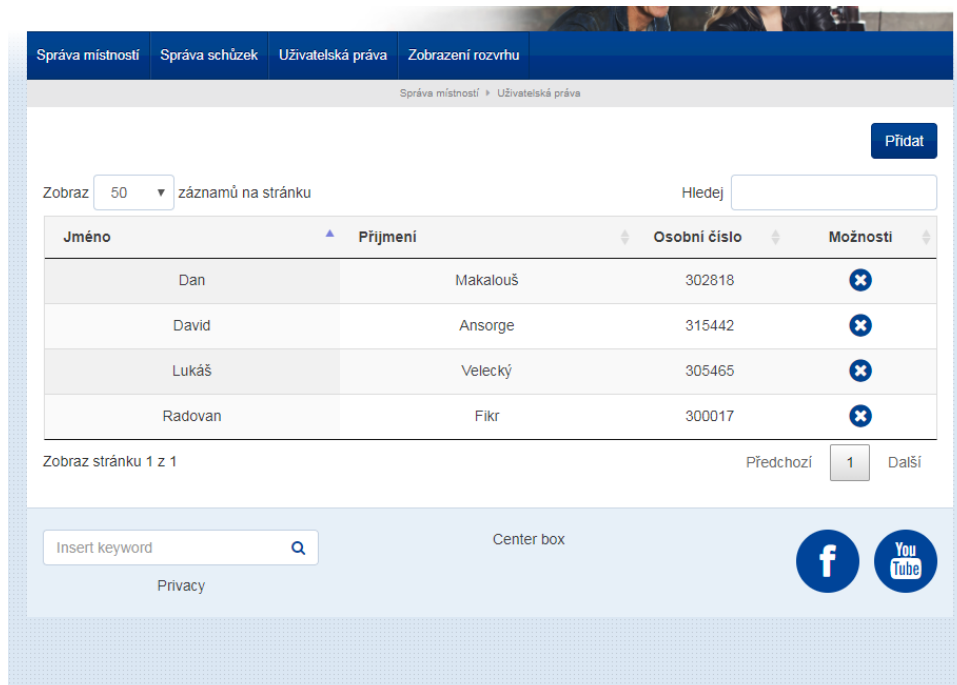
7.3 Uživatelská práva

Komponenta Uživatelská práva zobrazuje a spravuje administrátory pro tuto aplikaci. Je přístupná pouze pro administrátory. Ti si v ní mohou zobrazit všechny ostatní uživatele, kteří disponují administrátorskými právy, přidat dalšího administrátora nebo smazat existujícího. Na Obrázku 7.9 je zobrazena hlavní stránka komponenty, kde lze vidět všechny přidané administrátory. U administrátorů jsou zobrazeny pouze základní údaje (jméno, příjmení a osobní číslo). Při zapojení aplikace se přes dialog komponenty udělí práva jednomu administrátorovi, který může následně udělit práva ostatním.

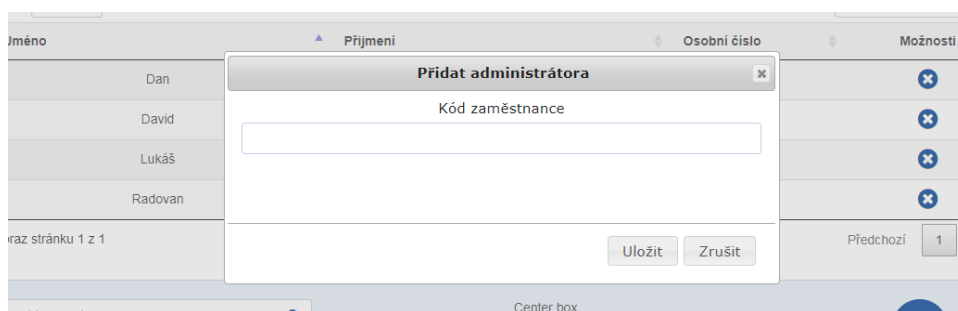
Při kliknutí na tlačítko „Přidat“ se zobrazí dialog viditelný na Obrázku 7.10, v němž je ve vyhledávacím poli možné najít uživatele, kterého je třeba přidat, a stisknutím tlačítka „Uložit“ uživatel získá administrátorská práva, dialog se zavře a přidáný uživatel se zobrazí ve výčtu viditelném na Obrázku 7.9. Mazání administrátora se provede intuitivně stisknutím křížku ve sloupci „Možnosti“ u vybraného člověka a potvrzením ve vyskakovacím dialogu.

Při otevření libovolné komponenty bez potřebných přístupových práv se obsah komponenty nenačte a zobrazí se upozornění na Obrázku 7.11.

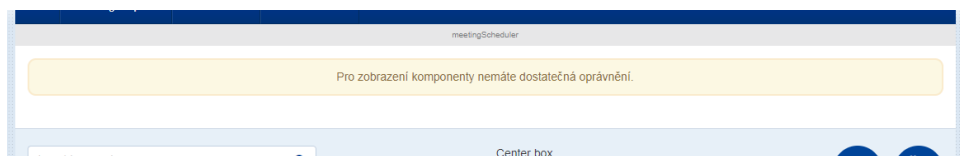
7. POPIS IMPLEMENTACE APLIKACE



Obrázek 7.9: Uživatelská práva, hlavní stránka.



Obrázek 7.10: Uživatelská práva, přidání administrátora.



Obrázek 7.11: Uživatelská práva, nedostatečná práva.

7.4 Zobrazení rozvrhu

Komponenta Zobrazení rozvrhu slouží k poslední fázi plánování. Přístup k ní mají pouze administrátoři. Při otevření komponenty se podle zadaných parametrů vygeneruje rozvrh ve formátu viditelném na Obrázku 7.12. Schůzky v rozvrhu lze přesunout do jiné místnosti nebo času pomocí přístupu „drag and drop“. Tyto změny už systémem nejsou kontrolovány. To znamená, že v důsledku takových změn by se například mohl ocitnout jeden člověk na více schůzkách zároveň. Schůzky je možné také prodlužovat nebo zkracovat. To je ovšem doporučeno provést v komponentě Správa schůzek přes dialog editace. Tímto způsobem bude totiž algoritmus moci se změněnou délkou schůzky počítat a pokusí se najít co nejlepší rozvrh s takto změněným zadáním. V komponentě je zvolen formát zobrazený na Obrázku 7.12, protože je úsporný na místo a také protože se při následné úpravě provádějí jednoduše změny místností u jednotlivých schůzek, při kterých nemůže vzniknout konflikt na účastnících schůzky.

Při kliknutí na název libovolné schůzky se v dialogu zobrazí seznam účastníků schůzky. Okno lze zavřít křížkem v pravém horním rohu, viz Obrázek 7.13

Druhá část této komponenty je generování souboru ve formátu XLSX. Stisknutím tlačítka „Generovat“ se rozvrh zobrazený v komponentě (Obrázek 7.12) předvede do souboru připraveného k tisku, který je zobrazen na Obrázku 7.14. Tento způsob zobrazení rozvrhu je přívětivější pro účastníky schůzek. Mohou si zde totiž najít schůzku, které se účastní, a pak se podívat, ve které místnosti a v který čas se koná. Každá místnost má pro přehlednost svou barvu, která jí byla přidělena při vytváření místnosti.

7. POPIS IMPLEMENTACE APLIKACE

HARTMANN

Přihlášený uživatel: admin, Administrator

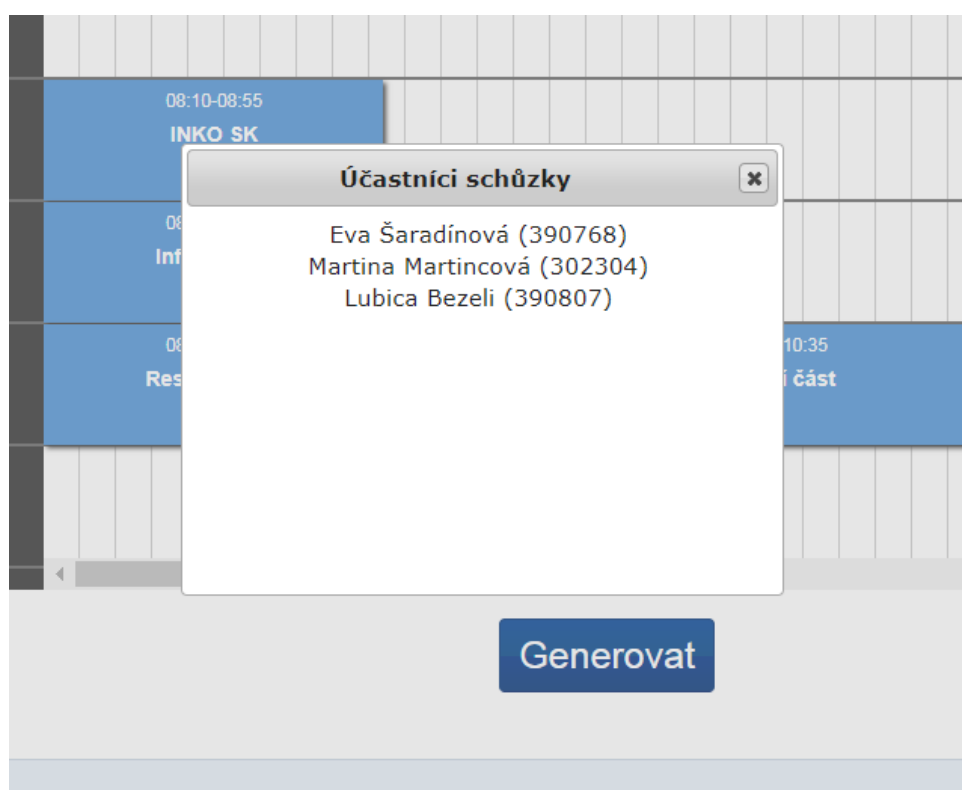
Správa místností | Správa schůzek | Uživatelská práva | Zobrazení rozvrhu

Správa místností > Zobrazení rozvrhu

	08:10	09:00	10:00
Mistnost Viktor			
Mistnost Joseph	08:10-08:55 Resseler Alza		
Mistnost Albert	08:10-08:55 Komunikace se zákazníkem	09:00-10:35 Hlavní část	
Mistnost Florence	08:10-08:55 INKO SK		
Mistnost Paul	08:10-08:55 Infokolečko		
Mistnost Ludwig			

Generovat

Obrázek 7.12: Zobrazení rozvrhu, hlavní stránka.



Obrázek 7.13: Zobrazení rozvrhu, účastníci schůzky.

7. POPIS IMPLEMENTACE APLIKACE

17							
	A	B	C	D	E	F	G
1	Téma	Svolává	Účastníci	Čas (minut)	08:10		09:00
2	Infokolečko	Eva Adamcová (303051)	Eva Adamcová (303051), Ondřej Gutyan (390219)	50	Místnost Joseph		
3	Resseler Alza	Marie Mošová (302119)	Marie Mošová (302119), Jana Hraběová (302832), Petr Dvořák (302454), Martin Kubát (302870), Jiřina Svobodová (300332)	50	Místnost Albert		
4	INKO SK	Eva Šaradínová (390768)	Eva Šaradínová (390768), Martina Martincová (302304), Lubica Bežel (390807)	50	Místnost Florence		
5	Komunikace se zákazníkem	Ilona Kučerová (302928)	Ilona Kučerová (302928), Jan Cívín (302881), Irena Malá (300429), Sylwia Řihošková (300671)	50	Místnost Paul		
6	Hlavní část	Radovan Fikr (300017)	Radovan Fikr (300017)	100			Mís
7	Prezentace Hartmann	Pavel Fuchs (302732)	Pavel Fuchs (302732), Sylwia Řihošková (300671)	50			
8	Kompresivní terapie	Luděk Ogoun (302346)	Luděk Ogoun (302346), Marie Mošová (302119), Václav Halama (300524), Tamara Jelínková (301479)	50			
9	Schůze bezpečnostního výboru	Lubomír Bureš (312238)	Lubomír Bureš (312238), Marek Třeška (300394), Radovan Fikr (300017), Martin Musil (300947), Martin Nekula (303113), Miluše Urbánková (311453)	75			

Obrázek 7.14: Zobrazení rozvrhu, generovaný soubor. Formát buněk je pozměněn kvůli čitelnosti textu.

8 Využité vývojové nástroje

Jak bylo již uvedeno v kapitole 3, použití následujících nástrojů bylo vynuceno prostředím firmy HARTMANN - RICO a.s., pro kterou byla tato aplikace vyvíjena.

- *Adobe Experience Manager* [7] je komerční systém pro správu webových stránek. Platforma AEM je založena na Apache Jackrabbit Oak, který implementuje Content repository API (JCR) [18] sloužící ke správě a údržbě dat a metadat, Apache Felix [19] zajišťující modularizaci všech částí systému a framework Apache Sling [20] umožňující komunikaci s vnitřní strukturou projektu. Mezi hlavní výhody platformy AEM patří přehlednost, škálovatelnost na různé projekty a optimalizovatelnost pro většinu zařízení (PC, mobilní telefon, tablet). Nevýhodou AEM je vysoká cena licence a monolitčnost systému, která souvisí s problémy při aktualizování systému. I přes vysokou cenu patří AEM stále mezi špičky v oblasti platforem pro správu webových stránek. Hlavním konkurentem je Sitecore CMS [21], jehož cena i kvalita služeb je srovnatelná.
- *Microsoft SQL Server* [8] je komerční databázový server používaný k ukládání a zpětnému získávání dat. Pro práci s SQL Serverem je v tomto projektu používáno SQL Server 2014 Management Studio.
- *Java* je objektově orientovaný vysokoúrovňový programovací jazyk. Kromě prezentační vrstvy je v Javě psán všechen kód této webové aplikace. Kód tohoto projektu je psán v již poněkud starší verzi Java 7 (2011), která je použita kvůli firemním standardům.
- *JSP soubory* neboli Java Server Pages jsou technologie umožňující vytvářet dynamické webové stránky pro projekty s vnitřní strukturou v jazyce Java. Stránku JSP tvoří kód HTML, CSS, JavaScriptu a případně Javy. V těchto JSP souborech jsou dostupné globální AEM objekty, které poskytují přístup k datům [22].

9 Závěr

Cílem této práce bylo navrhnout a implementovat systém, který bude dostatečně splňovat potřeby firmy. Tento cíl byl splněn, ale pouze do určité míry. Jak je uvedeno v úvodu práce, je téměř nemožné navrhnout systém, jenž plně nahradí práci, kterou vykonávali zaměstnanci. Jeden z největších ústupků v rámci automatizace problému bylo omezení délky schůzek a časů začátků schůzek. Tato omezení jsou detailně popsána v podkapitole 6.1.

Projekt byl již prezentován zaměstnancům firmy, kteří mají rozvrhování na starosti, a se systémem byli spokojeni. Nyní je projekt ve fázi před testováním v provozu. V první fázi bude projekt dostupný pouze administrátorům, kteří do systému zadají schůzky, a poté si nechají vygenerovat rozvrh. Jinak řečeno, v první fázi provozu se systémem nebude pracovat běžný uživatel. V této fázi se až skutečně zjistí, jak je systém opravdu dobrý a zda splňuje vše potřebné. Po případných úpravách a úspěšném využití systému jako pomůcky pro administrátory bude systém nasazen i pro běžné uživatele, tedy přímo účastníky schůzek.

Jak bylo již zmíněno, tyto sjezdy probíhají pravidelně. Mezi případná rozšíření by tím pádem mohla patřit implementace funkcionality, která bude mít za úkol hlídat, ve které termíny budou moci běžní uživatelé přidávat vlastní schůzky, a bude také zajišťovat resetování systému po proběhnutí sjezdu.

Není pochyb o tom, že na této aplikaci zůstává ještě mnoho k dokončení, než bude skutečně připravena nahradit zaměstnance, kteří na každém rozvrhu pracují i několik dní, ale věřím, že po dokončení všech úprav bude systém připraven ušetřit těmto lidem velké množství práce a starostí.

Literatura

- [1] Stanford. *NP-úplné problémy*. 2019. URL: <https://people.eecs.berkeley.edu/~vazirani/algorithms/chap8.pdf> (cit. 04. 11. 2019).
- [2] LLC. EMS Software. *EMS rozvrhovací systém*. 2019. URL: <https://www.emssoftware.com/platform/use-cases/meeting-room-scheduling> (cit. 07. 04. 2019).
- [3] Yarooms international S.R.L. *Yaroom, rozvrhovací systém místností pro schůzky*. 2019. URL: <https://www.yarooms.com/use-cases> (cit. 12. 04. 2019).
- [4] Red Hat, Inc. *OptaPlanner, optimalizační nástroj rozvrhování OptaPlanner*. 2019. URL: <https://www.optaplanner.org/> (cit. 12. 04. 2019).
- [5] Red Hat, Inc. *Red hat company*. 2019. URL: <https://www.redhat.com/en#> (cit. 27. 05. 2019).
- [6] International Business Machines Corporation (IBM). *IBM ILOG CPLEX Optimization Studio, optimalizační systém firmy IBM*. 2019. URL: <https://www.ibm.com/products/ilog-cplex-optimization-studio> (cit. 12. 04. 2019).
- [7] Adobe systems. *Adobe Experience Manager 6.2*. 2019. URL: <https://helpx.adobe.com/support/experience-manager/6-2.html> (cit. 27. 05. 2019).
- [8] Microsoft Corporation. *Microsoft Sql database system*. 2019. URL: <https://www.microsoft.com/en-us/sql-server/sql-server-2017> (cit. 27. 05. 2019).
- [9] MIT. *Table plug-in for jQuery*. 2019. URL: <https://datatables.net/> (cit. 17. 09. 2019).
- [10] MIT. *Front-end responsive library*. 2019. URL: <https://getbootstrap.com/> (cit. 17. 09. 2019).
- [11] MIT. *Simple Daily Schedule Plugin with jQuery and jQuery UI - Schedule*. 2019. URL: <https://www.jqueryscript.net/time-clock/Simple-Daily-Schedule-Plugin-with-jQuery-and-jQuery-UI-Schedule.html> (cit. 15. 09. 2019).
- [12] MIT. *Zoomable & Scrollable Schedule Component For jQuery - skedTape*. 2019. URL: <https://www.jqueryscript.net/time-clock/Schedule-Component-skedTape.html> (cit. 15. 09. 2019).

LITERATURA

- [13] Michael Pinedo. *Scheduling: theory, algorithms, and systems*. 2012. ISBN: 978-1-4614-2361-4. URL: <https://www.springer.com/gp/book/9781489990433>.
- [14] RUDOVÁ, Hana. *Rozvrhování. Presentace k předmětu*. 2019. URL: <https://www.fi.muni.cz/~hanka/rozvrhovani/prusvitky/all-start.pdf> (cit. 12.04.2019).
- [15] BURKE, Edmund K., KENDALL, Graham. *Search methodoligies, Introductory Tutorials in Optimalization and Decision Support Techniques*. Springer US, 2005. ISBN: 978-1-4614-6940-7. URL: <https://www.springer.com/us/book/9781461469391>.
- [16] Berkley. *Redukce na problém rozvrhování*. 2019. URL: <https://web.stanford.edu/class/archive/cs/cs103/cs103.1132/lectures/27/Small27.pdf> (cit. 21.09.2019).
- [17] GeeksforGeeks. *State Space diagram for Hill Climbing*. 2018. URL: <https://www.geeksforgeeks.org/introduction-hill-climbing-artificial-intelligence/> (cit. 30.11.2019).
- [18] The Apache Software Foundation. *Content Repository API for Java Technology*. 2019. URL: <http://jackrabbit.apache.org/jcr/jcr-api.html> (cit. 27.05.2019).
- [19] The Apache Software Foundation. *Apache Felix*. 2019. URL: <https://felix.apache.org/> (cit. 09.12.2019).
- [20] The Apache Software Foundation. *Apache sling*. 2018. URL: <https://sling.apache.org/> (cit. 27.05.2019).
- [21] Sitecore. *Application platform Sitecore*. 2019. URL: <https://www.sitecore.com/> (cit. 27.05.2019).
- [22] Javaworld. *Java server pages*. 2019. URL: <https://www.javaworld.com/article/3336161/what-is-jsp-introduction-to-jaserver-pages.html> (cit. 27.05.2019).

A Seznam příloh

Příloha 1 Návod k nasazení a testování projektu.

Příloha 2 Balíky se zkompilevaným projektem.

Příloha 3 Zdrojový kód projektu.