



**FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Analýza a návrh informačního systému pro evidenci stravenek
Student:	Lukáš Pechač
Vedoucí:	Ing. David Buchtela, Ph.D.
Studijní program:	Informatika
Studijní obor:	Informační systémy a management
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2019/20

Pokyny pro vypracování

Cílem práce je návrh, implementace a finanční analýza informačního systému pro evidenci a správu stravenek ve velké korporátní společnosti.

1. Provedte analýzu uživatelských požadavků na informační systém.
2. Navrhněte hlavní případy užití a uživatelské scénáře, za kterými by uživatelé přicházeli do systému.
3. Implementujte "proof-of-concept" systém ve webovém frameworku .NET MVC od společnosti Microsoft v jazyce C#.
4. Vypracujte finanční analýzu implementace tohoto systému, včetně nákladů, benefitů a návratnosti nasazení systému ve společnosti.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 11. února 2019



**FAKULTA
INFORMAČNÍCH
TECHNOLÓGIÍ
ČVUT V PRAZE**

Bakalářská práce

Analýza a návrh informačního systému pro evidenci stravenek

Lukáš Pechač

Katedra softwarového inženýrství

Vedoucí práce: Ing. David Buchtela, Ph.D.

3. května 2019

Poděkování

Na tomto místě bych chtěl poděkovat Ing. Davidu Buchtelovi, Ph.D. za vedení při tvorbě této práce a za užitečné rady, které mi poskytl. Dále bych chtěl poděkovat své rodině a přítelkyni za morální a psychickou podporu po celou dobu studia. V neposlední řadě bych rád poděkoval svým kolegům a nadřízeným v práci za to, že mi poskytli časovou flexibilitu, kdykoliv jsem ji kvůli studiu potřeboval.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 3. května 2019

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2019 Lukáš Pechač. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Pechač, Lukáš. *Analýza a návrh informačního systému pro evidenci stravek*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2019.

Abstrakt

Cílem této bakalářské práce je vytvoření informačního systému pro správu stravenek ve velké korporátní firmě. Evidence stravenek, která se v současné době ve firmě využívá, je již zcela nevyhovující kvůli personálnímu růstu firmy a s tím spojené časové náročnosti evidence. Na základě analýzy a požadavků budoucích uživatelů informačního systému je navržena a implementována nová webová aplikace. Webová aplikace, která je součástí informačního systému usnadňuje a urychluje práci zaměstnancům firmy, kteří zajišťují správu stravenek ve firmě. Tito zaměstnanci jsou evidováni jako administrátoři informačního systému. Běžným zaměstnancům firmy webová aplikace poskytuje přehled o jejich aktuálním nebo historickém nároku na stravenky. Aplikace dále administrátorům umožňuje generovat důležité reporty nebo provádět základní administraci informačního systému. Nasazením informačního systému ve firmě se zkrátí čas evidence stravenek každý měsíc přibližně o 60 %. Informační systém je implementován pomocí jazyka C# v technologii Microsoft .NET s pomocí architektury MVC. Práce je zakončena zhodnocením nákladů na vývoj systému, srovnáním současné evidence s novým systémem, vyhodnocení nákladů a benefitů plynoucích z nasazení nového informačního systému.

Klíčová slova analýza evidence stravenek, stravenkový informační systém, webová aplikace, návrh, implementace, C#, .NET, MVC architektura

Abstract

The main aim of this bachelor thesis is to create an information system for evidence meal tickets in large corporate company. Current evidence system is completely unsatisfactory today due to the company's personal growth and the associated time-consuming evidence. On the basis of analysis, revealed requirements of future users is designed new web application. Web application, which is part of the information system, speed up the work with evidence for the company employees, who ensure the management of meal tickets. These employees are registered as administrators of the information system. For regular company employee application provide a simple overview about their current or historical meal tickets state. In addition, web application allows generate important reports and perform a basic administration which is provided by administrators. By deploying the information system in the company, the evidence time with meal tickets is reduced by approximately 60 % each month. The system is developed in C# language Microsoft .NET. technology with MVC architecture. The thesis is completed by evaluating the developed program, comparing the current system evidence with the new application and calculating the costs.

Keywords meal tickets evidence analysis, meal tickets information system, web application, design, implementation, C#, .NET, MVC architecture

Obsah

Úvod	1
1 Cíl práce	3
2 Literární rešerše	5
2.1 Představení oblasti stravenek	5
2.2 Představení zadavatele	6
2.2.1 Obecné informace	6
2.2.2 Identifikační číslo zaměstnance	6
2.2.3 Engagement	6
2.3 Komunikace s firmou Edenred CZ	7
2.4 Způsob výpočtu měsíčního nároku zaměstnance	7
2.5 Stávající evidence stravenek	8
2.5.1 Popis systému evidence	9
2.5.2 Průběh evidence	9
2.5.3 Zhodnocení systému evidence	11
3 Analýza	13
3.1 Požadavky	13
3.1.1 Funkční požadavky	13
3.1.2 Nefunkční požadavky	15
3.2 Výběr technologie	16
3.3 Hlavní případy užití	16
3.4 Uživatelské scénáře	16
3.4.1 Běžní uživatelé	17
3.4.2 Administrátoři	17
4 Realizace	21
4.1 Architektura systému	21

4.1.1	Model	21
4.1.2	View	22
4.1.3	Controller	22
4.1.4	Service	23
4.2	Hlavní použité technologie	23
4.2.1	EntityFramework	23
4.2.2	AutoMapper	23
4.2.3	Castle Windsor	24
4.2.3.1	Dependency Injection	24
4.2.4	HangFire	25
4.2.5	Log4Net	26
4.2.6	DevExpress	26
4.2.7	Bootstrap	27
4.2.8	FontAwesome	28
4.2.9	jQuery	29
4.3	Databáze	30
4.3.1	Databázový model	30
4.3.2	Uložené procedury	30
4.3.3	Funkce	32
4.4	Uživatelské rozhraní	32
4.5	Popis funkcionality informačního systému	33
4.5.1	Domovská stránka	33
4.5.2	Detail uživatele	33
4.5.3	Správa měsíčních period	34
4.5.3.1	Práce s měsíční periodou	34
4.5.4	Administrace systému	36
5	Ekonomicko-manažerské vyhodnocení	37
5.1	Náklady na informační systém	37
5.2	Benefity	38
5.3	Finanční návratnost	38
	Závěr	41
	Seznam použité literatury	43
	A Seznam použitých zkratk	47
	B Ukázky systému	49
	C Přílohy	59
	D Obsah příloženého CD	61

Seznam obrázků

2.1	Engagement	7
2.2	Soubor pro Edenred CZ – první list obsahující zaměstnance, kteří dosud nepobírali stravenky a mají na ně nově nárok	8
2.3	Soubor pro Edenred CZ – druhý list obsahující měsíční nárok každého zaměstnance, který už v minulosti pobíral stravenky	8
2.4	Komunikace jednotlivých tabulek na listech evidenčního souboru mezi sebou	10
3.1	Diagram případů užití informačního systému	16
4.1	Komunikace uživatele s aplikací a komunikace jednotlivých částí mezi sebou	22
4.2	Funkce controller vrstvy	22
4.3	Entity Framework [9]	23
4.4	Ukázka komponenty GridView od firmy DevExpress [18]	27
4.5	Bootstrap tlačítka	28
4.6	Bootstrap vyskakovací hlášky	28
4.7	Ukázka vybraných FontAwesome ikon z oficiálních stránek produktu [23]	29
4.8	Relační model databáze	31
5.1	Návratnost nákladů na nový informační systém vzhledem ke starému systému od 1. měsíce nasazení	39
B.1	Úvodní stránka administrátora informačního systému	50
B.2	Osobní stránka zaměstnance, jak ji vidí běžný uživatel informačního systému	51
B.3	Osobní stránka zaměstnance, jak ji vidí administrátor informačního systému	52
B.4	Formulář pro editaci měsíční periody na detailní stránce zaměstnance	53
B.5	Stránka pro import souborů nové měsíční periody	54

B.6	Stránka pro generování reportů pro již uzavřené měsíční periody	55
B.7	Formulář pro přidání nového administrátora do informačního systému	56
B.8	Kontrolní report pro administrátora systému s možností exportování dat do Excelu	57
B.9	Administrační tabulka s engagement kódy u kterých nevzniká nárok na stravenky	58
C.1	Grafické znázornění práce administrátora s měsíční periodou pomocí nového informačního systému	60

Seznam tabulek

5.1	Časový odhad a cena práce .NET Developera	38
5.2	Časový odhad a cena práce specialisty na benefity ve firmě	38

Seznam výpisů kódu

4.1	Kód bez použití AutoMapperu	24
4.2	Použití AutoMapperu v kódu	24
4.3	Ukázka použití návrhového vzoru Dependency Injection	25
4.4	Ukázka použití logovacího nástroje Log4Net	26
4.5	Začlenění písma FontAwesome do vlastních stránek	29

Úvod

V dnešní době, kdy je na trhu práce nedostatek kvalifikovaných uchazečů o práci, se mnoho firem snaží nové zaměstnance nalákat na velké množství nabízených benefitů. Oblíbeným a vyhledávaným benefitem jsou i stravenky. Tento benefit nabízí i nejmenovaná celosvětově známá korporátní firma, pro kterou už druhým rokem pracuji na pozici IT junior developer interního oddělení a aktuálně řeším úkol navrhnout a vyvinout nový informační systém na správu stravenek pro zaměstnance ve firmě.

Současný zastaralý systém správy stravenek, který se ve firmě (dále zadavatel) používá, funguje pomocí několika generovaných Excel souborů, ze kterých se vytvoří jeden výsledný Excel soubor a ten se odešle externí firmě, která nabízí stravenkové karty. Toto začíná být vzhledem k personálnímu růstu zadavatele časově neefektivní a každý měsíc zaměstnává příliš mnoho zaměstnanců personálního oddělení.

Při výběru tématu bakalářské práce pro mě bylo prioritou vybrat si takové téma, které vychází z reálné praxe a práce bude v budoucnu sloužit reálným uživatelům a usnadní práci zaměstnancům zadavatele.

V bakalářské práci navrhnu a vytvořím informační systém pro správu stravenek, který ulehčí každoměsíční práci zaměstnancům zadavatele. Umožní jim lepší přehled o této agendě a nabídne možnost generování důležitých reportů a přehledů. Dále poskytne lepší přehled ostatním zaměstnancům zadavatele o jejich měsíčním nároku na stravenky a personálnímu oddělení tím uleví od nekonečného odpovídání na otázky, které jsou stále podobné: „Proč jsem dostal/dostala za minulý měsíc pouze tolik stravenek?“.

Bakalářská práce je rozdělena na čtyři části. První, literární rešerše, seznamuje čtenáře s aktuálním systémem správy stravenek ve firmě, jeho hlavními nevýhodami a s pravidly firmy, které určují způsob výpočtu nároku na stravenky. Čtenář také získá informace o tom, jak probíhá komunikace s externí firmou dodávající stravenky a jaké jsou dnes možnosti v této oblasti.

Ve druhé části je představena základní funkčnost nového systému a popsána specifikace hlavních uživatelských požadavků. Dále jsou zde navrženy hlavní případy užití a uživatelské scénáře, za kterými budou uživatelé přicházet do systému.

Třetí část se zabývá samotným návrhem, který vychází z analýzy provedené ve druhé části. Jsou zde představeny dílčí části nového informačního systému. Dále je zde také představen navržený databázový model, použité technologie a vzhled uživatelského rozhraní.

V poslední čtvrté části je ekonomicko-manažerské zhodnocení celého projektu obsahující finanční analýzu implementace systému, včetně spojených nákladů, benefitů a návratnosti nasazení systému ve firmě.

Cíl práce

Cílem bakalářské práce je navrhnout a implementovat informační systém pro evidenci a správu stravenek ve velké korporátní firmě. Systém bude vyvinut ve webovém frameworku .NET MVC od společnosti Microsoft v jazyce C# a navržen podle provedené analýzy funkčních požadavků budoucích uživatelů, hlavních případů užití a uživatelských scénářů. Dílčím cílem práce bude finanční analýza systému, včetně nákladů, benefitů a návratnosti nasazení systému ve firmě.

Navržený informační systém usnadní a zrychlí práci odpovědným zaměstnancům personálního oddělení ve firmě a v neposlední řadě připraví firmu na další budoucí personální růst. Hlavními požadavky na informační systém jsou jednoduchost, přehlednost a intuitivní ovládání.

Literární rešerše

V této kapitole týkající se literární rešerše práce čtenáře seznámím s možnostmi, které nyní problematika stravenek nabízí. Představím způsob, jak se určuje měsíční nárok každého zaměstnance na stravenky a popíši aktuální nevyhovující systémem správy stravenek s jeho hlavními nevýhodami pro firmu.

2.1 Představení oblasti stravenek

Stravenka je příspěvek na jídlo, který je z části nebo celý dotován zaměstnavatelem. Dá se jimi platit jídlo a nealkoholické nápoje ve vybraných podnicích a restauracích. Dnes je to velmi oblíbený a vyhledávaný benefit zaměstnanci a velké množství společností už ho nabízí. Pro zaměstnavatele má vydávání stravenek místo klasického peněžního příspěvku na jídlo také své výhody a to takové, že si zaměstnavatel může odečíst z daní až 55 % hodnoty jedné stravenky [1]. Následující seznam uvádí tři nejrozšířenější poskytovatele stravenek na českém trhu a v závorce je u každého poskytovatele uveden název vydávaných stravenek:

- Sodexo Pass ČR (Gastro Pass)
- Up Česká republika (Chèque Déjeuner)
- Edenred CZ (Ticket Restaurant)

Dříve se stravenky vydávaly v podobě papírových poukázek, které měly hodnotu přibližně od 20 Kč až do 130 Kč [2]. Dnes poskytovatelé nabízí také elektronickou stravenku, která se stává stále více oblíbenější. Ta funguje na principu nabité karty a platí se s ní podobně jako s kreditní kartou. Hlavní výhodou elektronické stravenky je možnost za jídlo zaplatit přesnou částkou, která je poté stržena z karty a nemusí se vypořádat přesná hodnota jedné nebo několika stravenek, jako to bylo s papírovými stravenkami.

V této práci bude poskytovatel společnost Edenred CZ se svými elektronickými Ticket Restaurant stravenkami.

2.2 Představení zadavatele

Informační systém (IS), který je cílem této bakalářské práce, je vyvíjen pro konkrétní firmu (dále zadavatel), která si však nepřála být v této práci uvedena. Pro pochopení některých pojmů, které se u zadavatele používají a v této práci budou použity slouží kapitoly 2.2.2 *Identifikační číslo zaměstnance* a 2.2.3 *Engagement* ve kterých budou vysvětleny.

2.2.1 Obecné informace

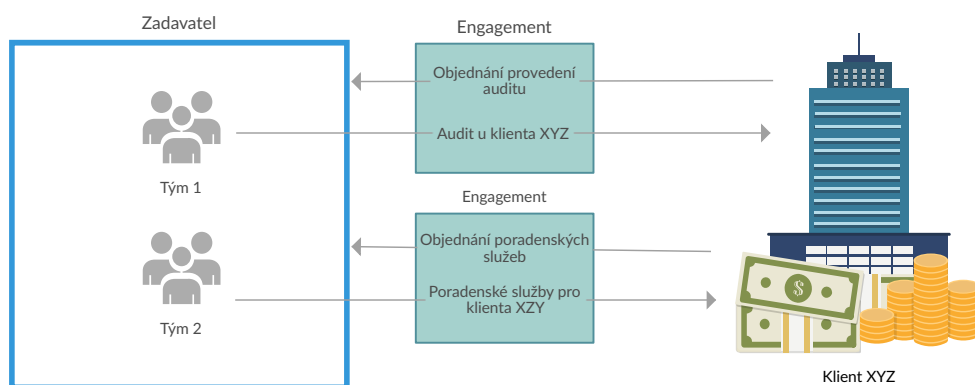
Zadavatelem je firma působící po celém světě v oblasti auditu nebo poradenství. V rámci celého světa je rozdělena na regiony podle celého kontinentu nebo podle jeho části ve kterém působí. Regiony jsou poté ještě rozděleny podle jednotlivých zemí. Česká republika je zařazena do regionu nazvaného Central Europe. Vyvíjený informační systém (IS) bude sloužit pouze pro používání v České republice, kde je nyní zaměstnáno přibližně 1500 zaměstnanců na čtyř různých místech po celé zemi. Počet zaměstnanců každým rokem neustále roste a proto je vytvoření IS nezbytné pro další fungování zadavatele v České republice.

2.2.2 Identifikační číslo zaměstnance

Všichni zaměstnanci zadavatele jsou evidováni v centrální databázi, která je společná pro celý region a každému je tak v rámci regionu přiřazeno unikátní identifikační číslo. Toto identifikační číslo se nazývá *EmpId* a vzniklo zkrácením anglických slov *employee* a *identification*.

2.2.3 Engagement

Anglické slovo *engagement* v českém překladu znamená podstatné jméno „střetnutí“ a v rámci zadavatele se tímto pojmem označuje konkrétní zakázka, která má přiřazeného klienta, pro kterého je tato zakázka vypracovávána. Pro lepší pochopení tohoto pojmu slouží obrázek 2.1 znázorňující klienta XZY, který má od zadavatele objednaný audit ve své firmě a poradenské služby například pro optimalizaci jeho interních procesů. Každý člen týmu zadavatele, pracující na daném úkolu vykazuje práci a přiřazuje ji k engagementu. Každému engagementu je přiřazen jednoznačný identifikátor tzv. „engagement code“.



Obrázek 2.1: Engagement

2.3 Komunikace s firmou Edenred CZ

Komunikace firmy která nabízí svým zaměstnancům stravenky, s poskytovatelem stravenek – firmou Edenred CZ (dále Edenred) – probíhá každý měsíc prostřednictvím Excel souboru. Edenred firmě poskytne soubor, který obsahuje dva listy. První list je určený pro zaměstnance, kteří mají za minulý měsíc nárok na stravenky, ale ještě nevlastní stravenkovou kartu. Edenred těmto lidem kartu založí a pošle ji do firmy s nabitou částkou. Druhý list je pro zaměstnance, kteří již vlastní stravenkovou kartu a firma Edenred tuto kartu pouze dálkově nabije odpovídající částkou. Tento komunikační soubor má pevně daný formát a strukturu, kterou firma nesmí měnit přidáváním ani odebíráním sloupců. Ukázkou prvního listu souboru s vyplněnými daty zobrazuje obrázek 2.2, druhý list souboru je znázorněn obrázkem 2.3.

2.4 Způsob výpočtu měsíčního nároku zaměstnance

Měsíční nárok každého zaměstnance na počet stravenek se určuje pomocí několika pravidel. Asi nejzákladnější pravidlo, které napadne každého čtenáře je takové, že za každý odpracovaný pracovní den v měsíci, kdy zaměstnanec pracoval více než 4 hodiny vzniká nárok na jednu stravenku, resp. nabití stravenkové karty na částku odpovídající hodnotě jedné stravenky.

Jsou ale i pravidla, která měsíční nárok snižují. Zaměstnanec nemá nárok na stravenku za den, kdy si nárokuje diety. To znamená za den, kdy byl zaměstnanec na služební cestě a zaměstnavatel má tedy povinnost, podle zákona § 163 odst. 1 uvedeného v zákoníku práce vyplatit zaměstnanci stravné neboli diety. Dalším dnem, za který nemá zaměstnanec nárok na stravenku je den, kdy zaměstnanec čerpal tzv. *Home office*. To je den, kdy zaměstnanec

2. LITERÁRNÍ REŠEŘŠE

Jméno zaměstnance	Příjmení zaměstnance	Datum narození zaměstnance	Adresa zaměstnance	Město	PSČ	Jméno zaměstnance na kartě	Název firmy na kartě	E-mail zaměstnance	Osobní číslo zaměstnance na obědce	Počet odpracovaných dní	Denní objednávaná hodnota v Kč
Povinný údaj, Příklad: Ivo	Povinný údaj, Příklad: Novák	Povinný údaj, Formát DDMM/YYYY Příklad: 30/08/1994	Nepovinný údaj, Příklad: Na Poříčí 123	Nepovinný údaj, Příklad: Brno	Nepovinný údaj, Příklad: Ivo Novák	Nepovinný údaj, Max. 21 znaků, Příklad: Ivo Novák	Nepovinný údaj, Max. 21 znaků, Příklad: Edenred CZ	Nepovinný údaj, Příklad: Ivo.novak@edenred.com	Nepovinný údaj, Příklad: IS112345	Povinný údaj, Musí být vyšší než 0, Příklad: 1	Povinný údaj, Musí být vyšší než 0, Příklad: 1600
Leigh	Long	30/08/1994				Long	Název firmy		24054	1	1600
Jesse	Hoover	10/05/1994				Hoover	Název firmy		24738	1	5560
Shane	Beck	15/04/1984				Beck	Název firmy		18378	1	1260
Hedwig	Duffy	29/05/1993				Duffy	Název firmy		6591	1	2080
Brenna	Stevenson	31/12/1969				Stevenson	Název firmy		6760	1	1120
Nathan	Ashley	31/12/1969				Ashley	Název firmy		9262	1	1440
Kamal	Newman	31/12/1969				Newman	Název firmy		24824	1	2000
Venus	Cummings	31/12/1969				Cummings	Název firmy		4448	1	2480
Austin	Coffey	31/12/1969				Coffey	Název firmy		37954	1	1520
Keely	Cantrell	31/12/1969				Cantrell	Název firmy		27412	1	3040
Adria	Raymond	31/12/1969				Raymond	Název firmy		18958	1	160
Ivy	Head	31/12/1969				Head	Název firmy		4301	1	4160

Obrázek 2.2: Soubor pro Edenred CZ – první list obsahující zaměstnance, kteří dosud nepobírali stravenky a mají na ně nově nárok

Jméno zaměstnance	Příjmení zaměstnance	Datum narození zaměstnance	Osobní číslo zaměstnance	Počet odpracovaných dní	Denní objednávaná hodnota v Kč	Sériové číslo karty
Nepovinný údaj, Příklad: Ivo	Nepovinný údaj, Příklad: Novák	Nepovinný údaj, Formát DDMM/YYYY Příklad: 18/12/1984	Nepovinný údaj, Příklad: IS112345	Povinný údaj, Musí být vyšší než 0, Příklad: 1	Povinný údaj, Musí být vyšší než 0, Příklad: 100	Příklad: 7010000000
Jacqueline	Jacobson	13/03/1992	19866	1	1600	1625031938099
Malcolm	Wilkins	26/09/1982	9266	1	1600	1614042127199
Jessamine	Lloyd	07/04/1998	29370	1	320	1618042420299
Stephanie	George	09/01/1985	8978	1	2600	1693033079199
Geoffrey	Willis	08/12/1977	13216	1	3060	1600051298099
Zephania	Guzman	08/06/1996	20801	1	80	1687021672799
Paula	Green	26/11/1985	2084	1	1280	1684505886999
Jordan	Reyes	05/01/1991	13714	1	1440	1679030439999
Octavia	Wong	28/05/1987	25331	1	1520	1638041723199
Zeus	Jefferson	20/09/1992	27050	1	1440	1648090106399
Emerald	Hutchinson	30/03/1973	1861	1	1360	1645012812199
Reagan	Riggs	06/01/1992	28994	1	1600	1677062812699
Mikayla	Koch	10/03/1978	1491	1	1280	1698052106399
Kaden	Burgess	18/11/1976	10747	1	1600	1640090170499
Hollie	O'Neill	22/01/1986	22819	1	1520	1633040385699
Phoebe	Galloway	09/12/1972	5738	1	1520	1614102206999
Grant	Mack	04/12/1996	1680	1	1520	1633061102199
Quinn	Riley	18/11/1971	20551	1	1200	1665091560999
Juliet	English	11/09/1995	28843	1	1600	1607121219899
Ishmael	Wyatt	29/01/1990	28370	1	1600	1662092272699
Cade	White	23/09/1987	2333	1	1600	1690010718499
Thor	Cherry	01/04/1973	10253	1	2880	1690080700099
Nayda	Landry	26/11/1993	27270	1	1200	1661100631999
Colby	Pena	15/07/1987	3196	1	1600	1679122029399
Jessamine	Wolman	04/11/1976	8577	1	3420	1664020414899
Paul	Wallace	25/02/1991	7422	1	1600	1692011502099
Ignacia	Diaz	04/02/1983	28364	1	1360	1605101480799
Orson	Rivers	08/01/1998	12638	1	2640	1682022829999
Chantale	Dunn	18/10/1985	19416	1	1440	1676115720999
Joseph	Howell	11/01/1980	22419	1	1280	1654032511699
Lee	Best	10/07/1988	28771	1	1600	1647070630999
Flynn	Harrison	15/06/1992	3471	1	3400	1618062316999

Obrázek 2.3: Soubor pro Edenred CZ – druhý list obsahující měsíční nárok každého zaměstnance, který už v minulosti pobíral stravenky

pracoval, ale pracoval ze svého domova místo toho aby byl v kanceláři firmy. Posledním pravidlem, kdy nevzniká nárok na stravenku je pokud zaměstnanec pracuje o víkend, mimo svůj pracovní kalendář nebo ve dnech pracovního volna.

2.5 Stávající evidence stravenek

V této kapitole představím průběh stávající systém evidence stravenek, který se již ve firmě stává nepoužitelným a je nutné tento systém nahradit novým informačním systémem. Dále zhodnocení tohoto stávající systém a popíši jeho hlavní nevýhody.

Obsah této kapitoly vychází z rozhovoru se zaměstnankyní firmy Martinou Bernardovou [3], která využívala stávající nevyhovující systém evidence.

2.5.1 Popis systému evidence

Současný systém evidence stravenek probíhá v jednom souboru formátu Microsoft Excel, který spravuje pověřený zaměstnanec personálního oddělení (dále jako administrátor), který má tuto agendu na starosti. Tento soubor v sobě obsahuje mnoho různých listů s tabulkami. Tyto tabulky slouží jako zdroj dat jiné výsledné tabulce na jiném listu v tomto souboru. Z výsledné tabulky se na konci evidence vytvoří report, který se pošle firmě Edenred ke zpracování. Komunikaci jednotlivých tabulek mezi sebou a strukturu souboru znázorňuje obrázek 2.4. Popis jednotlivých listů souboru a jejich význam:

Giude Jednoduchý návod pro administrátora v jakém pořadí se má postup evidence provádět.

Holidays Na tomto listu je tabulka, do které se vyplňují státní svátky nebo dny celofiremního volna.

MealVouchers List s tabulkou, do které jsou z tabulek na listech „Diety“, „PaidVacation“ a *Holidays* importována data a tyto data jsou ještě následně ručně administrátorem upravována. Z této tabulky vzniká na konci výsledný report pro firmu Edenred.

Diety Tabulku na tomto listu musí administrátor naplnit daty, které mu vygeneruje mzdové oddělení firmy a tyto data říkají, kolik čerpali jednotliví zaměstnanci diet při svých služebních cestách.

PaidVacation Data pro tabulku na tomto listu znovu generuje mzdové oddělení a tyto data obsahují informace o počtu dní dovolené jednotlivých zaměstnanců za daný měsíc.

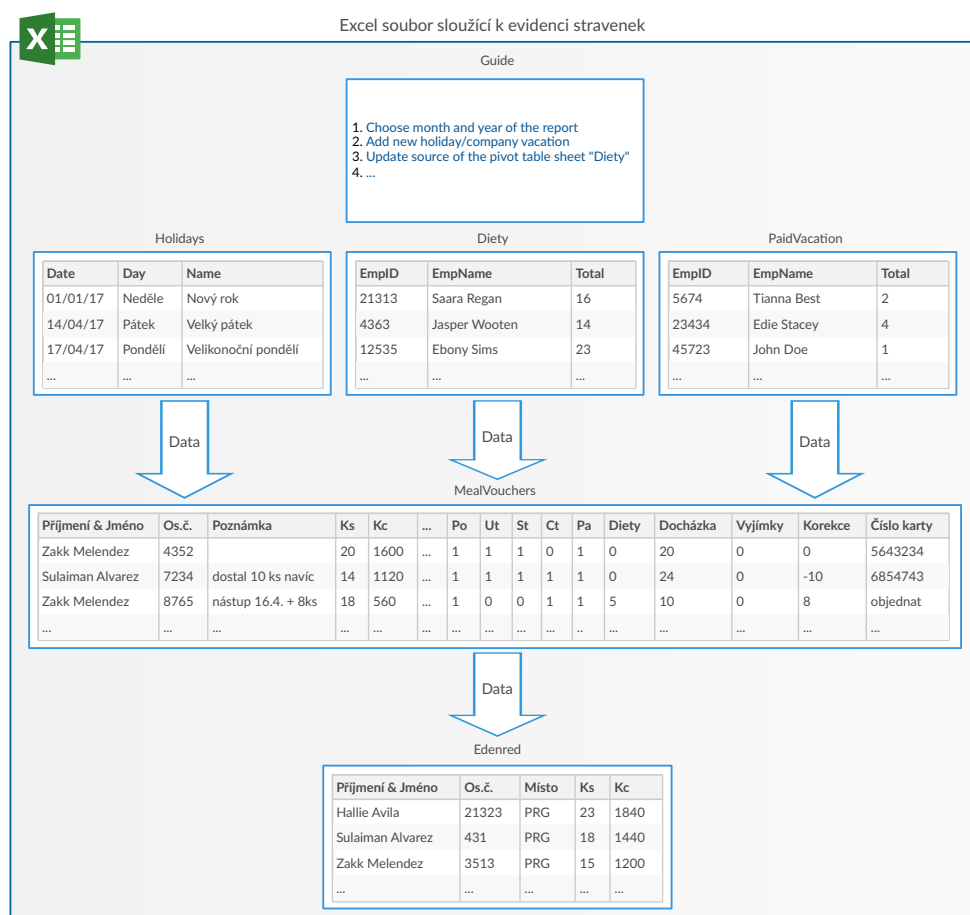
Edenred Obsahuje tabulku, která se automaticky plní daty z hlavní evidenční tabulky na listu „MealVouchers“. Po dokončení evidence se obsah této tabulky obsahují výsledná data překopíruje do souboru ke komunikaci s firmou Edenred.

2.5.2 Průběh evidence

Evidence stravenek za konkrétní měsíc začíná vždy několik málo dní po začátku následujícího kalendářního měsíce. Je to z toho důvodu, protože je potřeba evidenci začít dělat v době, kdy už jsou známy všechny výkazy odpracovaných hodin zaměstnanců, kteří ve firmě nepracují na pracovní smlouvu, ale jsou to osoby tzv. OSVČ (Osoby samostatně výdělečně činné). Tito lidé pracují pro firmu jako živnostníci na živnostenský list.

1. Jako první věc co musí administrátor udělat je upravit detaily v záhlaví listu „MealVouchers“ podle aktuálního měsíce. To znamená doplnit správné číslo měsíce, rok a hodnotu jednoho kusu stravenky. Tento

2. LITERÁRNÍ REŠERŠE



Obrázek 2.4: Komunikace jednotlivých tabulek na listech evidenčního souboru mezi sebou

soubor si poté administrátor uloží k sobě na disk a pracuje s ním, při provádění dalších kroků evidence stravenek.

2. Poté zkontroluje jestli v aktuální periodě byl nějaký státní svátek nebo se konalo celofiremní volno. Jestliže ano, tak tento volný den musí zadat do tabulky na listu „Holidays“.
3. Následuje krok, ve kterém musí administrátor projít report, který vygeneruje personální systém firmy a podle tohoto reportu přidat do tabulky na listu „MealVouchers“ nové zaměstnance, kteří mají nárok na stravenky. Dále je potřeba upravit zaměstnance, kteří změnili typ úvazku nebo odstranit již nepracující zaměstnance ve firmě.

4. Čtvrtý krok, co musí administrátor udělat je otevřít report od mzdového oddělení a data z tohoto reportu zkopírovat do tabulky na listu „Diety“.
5. Čtvrtým krokem se pro každého zaměstnance vypočítá hodnota do sloupce „Docházka“ v tabulce na listu „MealVouchers“ a administrátor musí zkontrolovat všechny řádky tohoto sloupce a ty které obsahují desetinné číslo, tak ty zaokrouhlit dolů na celé číslo.
6. Předposlední krok evidence je pro administrátora časově nejnáročnější. Musí projít všechny výkazy hodin zaměstnanců pracujících na živnostenský list a u každého ručně upravit v tabulce na listu „MealVouchers“ sloupce určující v jaké dny zaměstnanec pracoval a pokud je to nutné tak vyplnit i sloupec „Korekce“, aby zaměstnanci seděl výsledný nárok na počet kusů stravenek. Zaměstnanců pracujících na živnostenský list je v současné době přibližně 100 a s personálním růstem firmy se jejich počet neustále zvyšuje.
7. V posledním kroku evidence musí administrátor znovu projít všechny zaměstnance v tabulce na listu „MealVouchers“ a u těch, kteří mají ve svém řádku vyplněný sloupec „Poznámka“ upravit hodnotu výsledného nároku pomocí sloupce „Korekce“. Poznámku administrátor používá k tomu, aby příští měsíc věděl co je u daného zaměstnance potřeba udělat. Vyplňuje do ní například informaci o tom, zda se má zaměstnanci přidat nebo ubrat určitý počet kusů stravenek a to z důvodu chyby v minulém měsíci evidence, nebo podle jiných interních pravidel firmy.

2.5.3 Zhodnocení systému evidence

Celý tento proces měsíční evidence stravenek zabírá administrátorovi přibližně 5 pracovních dní v měsíci. Časově nejnáročnější položkou z těchto všech kroků je určitě evidence zaměstnanců pracujících pro firmu na živnostenský list. Je to dáno tím, že se musí manuálně projít všichni tyto zaměstnanci a jejich nárok na počet stravenek ručně upravit podle jejich výkazů hodin, který administrátor dostane od mzdového oddělení firmy.

Mimo časové náročnosti evidence pro administrátora vyplývají z tohoto systému také další nevýhody. Jedna z nevýhod je chybějící jednoduše dostupný přehled o nároku konkrétního zaměstnance za několik měsíců do minulosti a velmi těžko se tedy administrátorovi odpovídá na dotazy zaměstnanců typu: „Proč jsem dostal/dostala právě tolik stravenek? Neměl(a) jsem nárok na více/méně kusů?“.

Druhá nevýhoda v postupu evidence je, že se administrátor u jednotlivých „problémových zaměstnanců“ musí rozhodovat o dalším postupu podle ručně psaných poznámek z minulého měsíce. Problém nastane ve chvíli, kdyby evidenci stravenek musel dělat jiný zaměstnanec personálního oddělení než administrátor. Tato možnost může nastat například z důvodu nemoci, nebo ukon-

2. LITERÁRNÍ REŠERŠE

čení pracovního poměru administrátora. Nový nebo dočasný administrátor potom nepochopí význam poznámek u zaměstnanců a ve výsledku vzniknou chyby nebo se ještě více zvýší časová náročnost evidence.

Analýza

Analýza je jednou z nejdůležitějších fází vývoje aplikací. Při analýze jsem především zjišťoval co od nové aplikace očekávají budoucí uživatelé, jaké jsou jejich požadavky nebo jestli existují nějaké výjimky v procesech, které budou v systému implementovány. Na základě těchto znalostí jsem vybral technologii, jakou bude systém realizován a navrhl základní funkčnost systému.

3.1 Požadavky

Tato kapitola popisuje požadavky, které jsou kladeny na informační systém. Dělí se do dvou skupin – funkční a nefunkční. Funkční požadavky objasňují co musí systém umět a co se musí udělat. Nefunkční požadavky kladou omezení na design a provedení systému.

3.1.1 Funkční požadavky

FP1 Přístup do systému

Do systému budou moci vstupovat pouze administrátoři a uživatelé, kteří mají nebo v minulosti měli nárok na stravenky.

FP2 Import dat

Systém bude umět každý měsíc načítat data z vygenerovaných souborů, zpracovat je a uložit do databáze.

FP3 Detail zaměstnance

V systému bude mít každý zaměstnanec, který pobírá nebo pobíral stravenky svoji detailní stránku, kde bude budou zobrazeny tyto informace:

- Fotografie zaměstnance
- Základní údaje o zaměstnanci (zaměstnanecké číslo, datum narození, jméno, příjmení, kancelář apod...)

3. ANALÝZA

- Aktuální nárok na stravenky s detailním rozpisem položek, ze kterých byla tato hodnota vypočítána
- Historická data s nárokem na stravenky spolu s detailním rozpisem položek

FP4 Editace zaměstnance

Administrátoři systému, budou mít možnost u zaměstnance změnit a přidávat následující údaje:

- Poznámka k zaměstnanci
- Jméno a příjmení
- Volit, zda je jedná o klasického zaměstnance nebo o zaměstnance pracujícího na nějaký typ smlouvy
- Určovat, zda má zaměstnanec nárok na stravenky
- Kalendář pracovních hodin zaměstnance

Všechna tato data, která mají vliv na určování nároku stravenek se aplikují k zaměstnanci od následujícího měsíce.

FP5 Editace měsíční periody zaměstnance

Administrátoři systému, budou mít možnost u aktuálně otevřené měsíční periody měnit nebo přidávat následující údaje:

- Poznámka k periodě
- Volit, zda se perioda vztahuje ke klasickému zaměstnanci nebo k zaměstnanci, který pracuje na nějaký typ smlouvy
- Koncové datum měsíční periody
- Určovat, zda má zaměstnanec v aktuálním měsíci nárok na stravenky
- Počet korekcí a výjimek

FP6 Kontrolní report pro administrátory

Administrátoři systému budou mít v systému k dispozici stránku, kde si budou moci zobrazit data všech zaměstnanců za jednotlivé měsíční periody a exportovat tento přehled do Excel souboru.

FP7 Generování reportů

Systém bude umožňovat generovat report pro stravenkovou firmu Edenred CZ a report pro účtárnu firmy za uzavřené měsíční periody.

FP8 Editace administrátorů systému

Administrátoři systému budou mít možnost přidat nové administrátory nebo odebrat některého ze stávajících administrátorů.

FP9 Editace kalendářů

Administrátoři musí mít možnost přes webové rozhraní editovat, nebo přidat nové kalendáře, podle kterých zaměstnanci pracují.

FP10 Editace zaměstnaneckých výjimek

Administrátoři musí mít možnost přes webové rozhraní editovat, nebo přidat nové zaměstnance, jejichž identifikační číslo z personálního systému firmy neodpovídá identifikačnímu číslu v centrální databázi zaměstnanců firmy.

FP11 Editace zakázek bez nároku na stravenky

Administrátoři musí mít možnost přes webové rozhraní editovat, nebo přidat existující engagementy, které znamenají den pracovního volna a tedy nevzniká nárok na stravenku.

3.1.2 Nefunkční požadavky

NP1 Webová aplikace

Vytvořený systém bude dostupný přes web s podporou internetových prohlížečů *Internet Explorer* (verze 11.0 a vyšší), *Microsoft Edge* (verze 40 a vyšší) a *Google Chrome* (verze 43.0 a vyšší).

NP2 Uložení dat

Systém bude k uchovávání dat využívat databázi uloženou v relačním databázovém systému *Microsoft SQL Server 2016*.

NP3 Běhové prostředí

Systém bude nasazen na webovém serveru *Internet Information Services (IIS)* [4] verze 10.0 a vyšší.

NP4 Uživatelské rozhraní

Grafické rozhraní systému (GUI) bude navrženo na míru zaměstnancům personálního oddělení, aby práce s ním byla pro ně co nejjednodušší.

NP5 Typy zaměstnanců

V systému budou vedeni zaměstnanci pracující pro firmu na pracovní smlouvu, ale i zaměstnanci kteří pracují na živnostenský list a tedy nejsou běžnými zaměstnanci firmy.

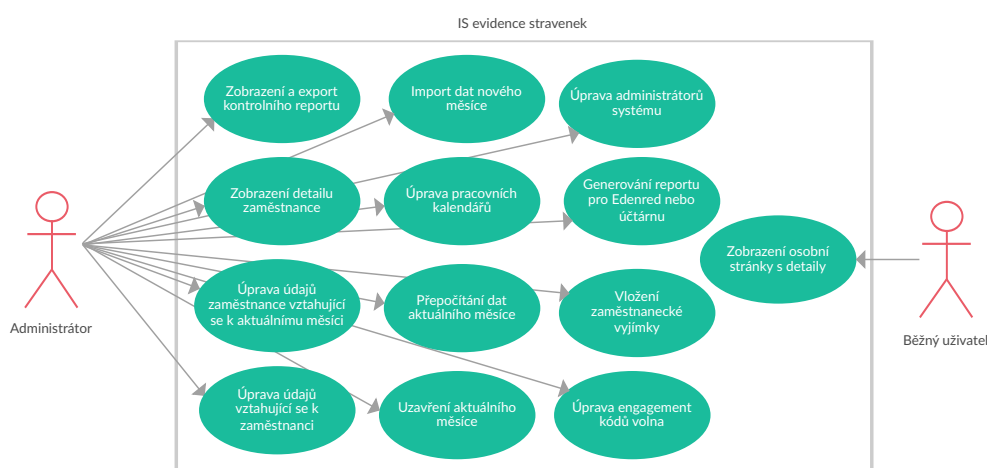
3.2 Výběr technologie

Podle nefunkčních požadavků vyplývá, že vyvíjený systém má být dostupný pomocí webového prohlížeče. Jelikož firma, pro kterou je systém určen, používá především technologie společnosti Microsoft, tak webový framework ASP.NET v kombinaci s jazykem C# jsem zvolil jako ideální řešení.

Data budou uložena na Microsoft SQL serveru, který se již ve firmě používá a pouze se na něm založí nová databáze.

3.3 Hlavní případy užití

V této kapitole představím hlavní případy užití informačního systému. Tento termín bývá také často nahrazován pojmem *Use case*, což má stejný význam. Pomocí obrázku 3.1 jsem znázornil diagram případů užití informačního systému spolu s účastníky systému a jejich oprávnění.



Obrázek 3.1: Diagram případů užití informačního systému

3.4 Uživatelské scénáře

V této kapitole popíši typické uživatelské scénáře za kterými budou zaměstnanci firmy přicházet do systému. Scénáře jsem rozdělil na scénáře pro běžného uživatele pobírající stravenky a na scénáře pro administrátory systému, kterými jsou zaměstnanci personálního oddělení, kteří mají na starosti správu stravenek.

3.4.1 Běžní uživatelé

Vzhledem k tomu, že je vyvíjený systém má sloužit především pro zaměstnance personálního oddělení firmy, kteří spravují stravenky, běžní uživatelé mají pouze jeden uživatelský scénář. Tím scénářem je kontrola výpočtu jejich měsíčního nároku na stravenky v případě, že mají nějaké pochybnosti o správnosti jejich nároku.

US1 Kontrola správnosti měsíčního nároku běžným uživatelem

1. Scénář začíná v okamžiku, kdy zaměstnanec, který má nárok na stravenky, zkontroluje stav své stravenkové karty, jestli již nemá nabytou částku odpovídající nároku za minulý měsíc a zjistí, že očekával nabití jiné částky.
2. Uživatel (zaměstnanec) si otevře web s informačním systémem, kde se mu zobrazí jako úvodní stránka osobní detail.
3. Na své detailní stránce uživatel zkontroluje všechny položky, ze kterých mu byl vypočítán nárok na stravenky a uvědomí si například, že zapomněl na den, kdy pracoval z domova a za tento den tedy neměl na stravenku nárok.

3.4.2 Administrátoři

Oproti uživatelským scénářům pro běžné uživatele systému, mají administrátoři více typických uživatelských scénářů. Mezi čtyři netypičtější scénáře jsem vybral přidání nového administrátora do systému, založení nové měsíční periody, přepočítání odpracovaných dní v měsíci u zaměstnanců s nepravidelnou pracovní dobou a jako poslední jsem vybral editaci údajů v aktuálně otevřené měsíční periodě u konkrétního zaměstnance.

AS1 Přidání nového administrátora do systému

1. Scénář začíná, jestliže se administrátor rozhodne přidat do IS nového administrátora.
2. Systém zobrazí pop-up okno, kde bude možnost zadat identifikační číslo zaměstnance (EmpId), který má být přidán do systému.
3. Administrátor vyplní EmpId a stiskne tlačítko pro potvrzení.
4. Pokud zaměstnanec s daným EmpId ve firmě existuje, systém uloží tohoto zaměstnance mezi administrátory systému.

AS2 Import nové měsíční periody do systému

1. Scénář začíná ve chvíli, kdy administrátor systému má k dispozici všechny potřebné soubory k vytvoření nové měsíční periody.
2. Administrátor všechny tyto soubory přesune do určeného síťového adresáře.
3. Administrátor otevře informační systém a na stránce týkající editace period klikne na tlačítko pro import nové periody.
4. Systém soubory zpracuje, přiřadí k odpovídajícím zaměstnancům nebo založí nového zaměstnance a jejich data uloží do své databáze.

AS3 Import odpracovaných dní zaměstnanců s nepravidelnou pracovní dobou, za aktuálně otevřenou měsíční periodu

1. Scénář začíná ve chvíli, kdy vyprší lhůta ve které musí všichni zaměstnanci s nepravidelnou pracovní dobou doplnit do docházkového systému firmy svoje odpracované hodiny za minulý měsíc.
2. Administrátor otevře informační systém na stránce týkající editace period a klikne na tlačítko pro načtení odpracovaných dní zaměstnanců s nepravidelnou pracovní dobou.
3. Systém vyfiltruje pouze zaměstnance kterých se to týká a pro tyto zaměstnance vypočítá počet odpracovaných dní s nárokem na stravenku a tyto informace uloží k zaměstnanci.

AS4 Editace údajů v aktuálně otevřené měsíční periodě u konkrétního zaměstnance

1. Scénář začíná ve chvíli, kdy administrátor stravenek například zjistí, že konkrétní zaměstnanec v minulém měsíci dostal špatný počet stravenek a chce tuto chybu opravit.
2. Administrátor si zobrazí seznam všech zaměstnanců v aktuálním otevřeném měsíci a vybere „problémového“ zaměstnance.
3. Systém zobrazí administrátorovi osobní stránku zaměstnance.
4. Administrátor vybere možnost sloužící pro úpravy periody.

5. Systém zobrazí formulář umožňující zadat mimo jiné korekci k nároku za aktuální měsíc.
6. Administrátor vyplní pole pro korekci, podle toho jestli chce zaměstnanci stravenek přidat nebo ubrat a formulář uloží.
7. Systém zpracuje vloženou hodnotu, upraví podle hodnoty měsíční nárok a data uloží.

Realizace

V této kapitole se zabývám popisem implementace informačního systému pro správu stravenek ve firmě. Nejdříve představím architekturu systému a použité technologie. Poté následně představím navržený databázový model a jako poslední zde popíši strukturu uživatelského rozhraní informačního systému.

4.1 Architektura systému

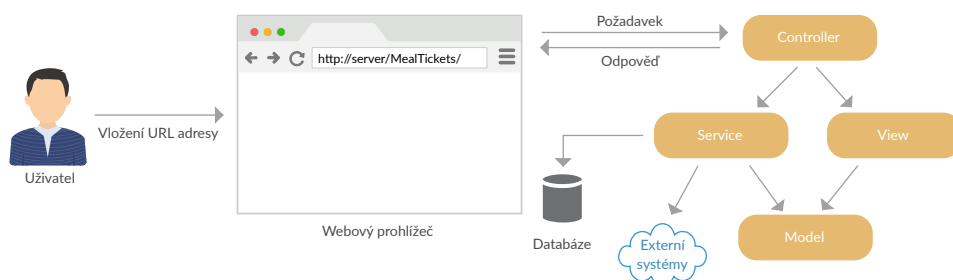
Systém jsem realizoval jako webovou aplikaci ve frameworku ASP.NET MVC5. Jak už název frameworku napovídá, jedná se o MVC, neboli Model-View-Controller architekturu. MVC je třívrstvá architektura používaná převážně na webu a v současnosti je to jedna z nejpoužívanějších a nejobecnějších architektur díky její jednoduché základní myšlence. Základní myšlenkou této architektury je rozdělení programu do tří částí, které spolu komunikují. Jedná se o části *model*, *view* a *controller*. [5]

V aplikaci je však potřeba, aby aplikace komunikovala i s vlastní nebo externí databází, proto jsem architekturu MVC rozšířil ještě o jednu vrstvu a to o tzv. servisní vrstvu. Tato vrstva má na starosti komunikaci s databází a externími systémy. Výhodou servisní vrstvy je snadnější testování aplikace, díky nahrazení této vrstvy „mock“ objektem, se kterým při testování aplikace mohou pracovat [6]. Vznikla tedy architektura, která se někdy označuje jako Model-View-Controller-Service (MVCS). Ukázkou komunikace uživatele s aplikací a komunikaci jednotlivých částí mezi sebou znázorňuji obrázkem 4.1.

4.1.1 Model

Modely jsou klasické C# objekty ve kterých jsou uložena data. V případě použití technologie EntityFramework, kterou popíši v kapitole 4.2.1 *EntityFramework*, modely přímo odpovídají jednotlivým tabulkám databáze.

4. REALIZACE



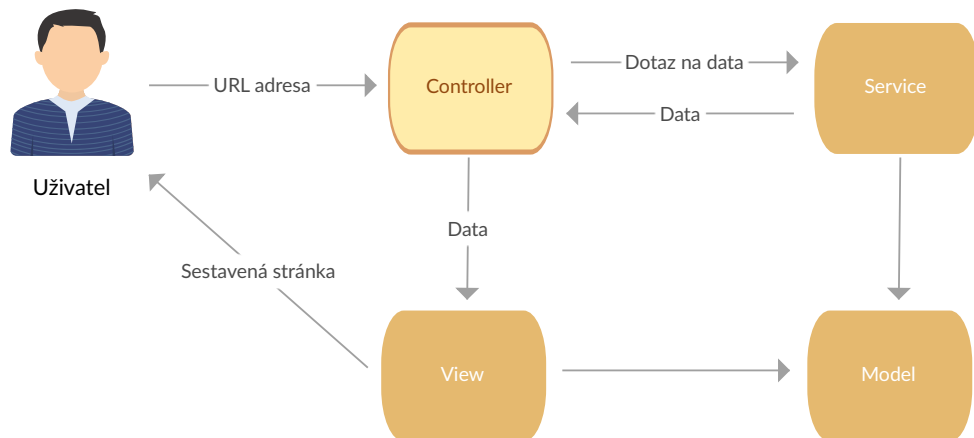
Obrázek 4.1: Komunikace uživatele s aplikací a komunikace jednotlivých částí mezi sebou

4.1.2 View

View vrstva má za úkol zobrazit výstup uživateli. Jedná se o HTML šablonu, do které vstupuje model s daty. Je to HTML stránka rozšířená o tagy speciálního jazyka, který umožňuje do šablony vkládat proměnné z modelu, provádět cykly nebo definovat podmínky. V tomto případě se jedná o speciální tagy jazyka Razor, který je založen na programovacím jazyku C# [7].

4.1.3 Controller

Controller je vrstva, která plní funkci takového prostředníka. Komunikuje s ní uživatel, pomocí url adresy, ale i service a view vrstva. Drží tedy celý systém pohromadě a propojuje spolu jednotlivé komponenty. Funkci controlleru lze snadněji pochopit pomocí obrázku 4.2.



Obrázek 4.2: Funkce controller vrstvy

4.1.4 Service

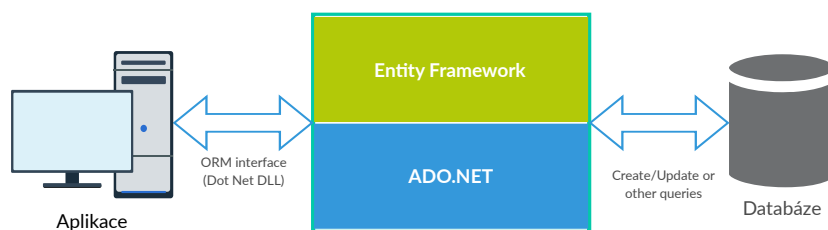
Servisní vrstva slouží ke komunikaci s „okolním světem“. Tato vrstva například pracuje s databází, komunikuje s externími systémy nebo obsahuje výpočty a business logiku systému. Vrstva zajistí data, udělá s nimi nějakou operaci a poskytne je ostatním součástem ke zpracování. Dá se říci, že tato vrstva vznikla odebráním části funkcionality view vrstvy v klasické MVC architektuře.

4.2 Hlavní použité technologie

Při vývoji aplikace jsem použil mnoho externích knihoven a frameworků. V této kapitole jsem vybral ty nejdůležitější a ty zde představím.

4.2.1 EntityFramework

Entity Framework (EF) je jednou z neznámějších technologií pro vývoj v .NET Frameworku. Jedná se o oblíbený open-source ORM (Object-relation mapper) mapper pro ADO.NET, který je součástí .NET Frameworku. Mapuje objekty .NET aplikace s tabulkami relační databáze. To znamená, že pokud v aplikaci chci získat například veškerý obsah jedné tabulky z databáze, požádám o to EF a ten mi vrátí kolekci výsledků, které odpovídají záznamům v tabulce. V ideálním případě se vývojář při použití EF může úplně zbavit používání jazyka SQL a pracuje s daty, jako kdyby se jednalo o kolekce přímo v .NET prostředí. Tento nástroj v roce 2008 představila společnost Microsoft a do dnešního dne si získal mnoho příznivců z řad .NET vývojářů, ale i mnoho odpůrců [8]. Hlavní věcí, která je jeho odpůrci kritizována, je právě to odříznutí od SQL a vývojář tedy nemá kontrolu nad tím, jaké dotazy v databázi EF provede.



Obrázek 4.3: Entity Framework [9]

4.2.2 AutoMapper

AutoMapper je jednoduchá open-source knihovna poskytovaná firmou vlastníci webovou službu *GitHub*. Umožňuje mapovat jeden objekt do druhého. Je to tedy object-object mapper. Mapování funguje transformací vstupního

objektu jednoho typu na výstupní objekt jiného typu. Během mapování můžeme například provést výpočty založené na datech vstupního objektu a výsledek uložit do výstupního objektu. Bez použití AutoMapperu by se vývojáři při psaní aplikace dokázali obejít, ale AutoMapper pomáhá zkracovat kód a udržovat ho čitelnější.

Pokud vývojář chce mapovat jeden objekt do druhého bez použití AutoMapperu, musí nejprve vytvořit novou instanci výstupního objektu a tu naplnit pomocí vstupního objektu. Použitím AutoMapperu vznikne možnost toto vše zkrátit do jednoho krátkého řádku kódu, za předpokladu, že jsme někde dříve v aplikaci definovali „mapu“, podle které bude AutoMapper vědět jak namapovat položky vstupního objektu na položky výstupního objektu. Na ukázkách zdrojového kódu 4.1 a 4.2 je vidět, jak použití AutoMapperu zkracuje kód. [10]

Kód 4.1: Kód bez použití AutoMapperu

```
var userDto = new UserDto();

var user = db.Users.Where(u => u.user_id == id).FirstOrDefault();
if (user != null)
{
    userDto.UserId = user.userId;
    userDto.Name = user.userName;
    userDto.Address = user.userAddress;
    userDto.PhoneNo = user.userPhone;
}
```

Kód 4.2: Použití AutoMapperu v kódu

```
var user = db.Users.Where(u => u.user_id == id).FirstOrDefault();
var userDto = Mapper.Map<User, UserDto>(user);
```

4.2.3 Castle Windsor

Castle Windsor je kontejner pro návrhový vzor *Dependency Injection* (DI), což se dá do češtiny přeložit jako „vkládání závislostí“. Pro pochopení co znamená kontejner pro DI musím nejdříve vysvětlit návrhový vzor DI.

4.2.3.1 Dependency Injection

Návrhový vzor DI slouží pro snížení závislostí mezi jednotlivými částmi systému. Jeho provedení je inspirováno obecnějším návrhovým vzorem *Inversion of Control* (IoC). Tyto dva pojmy jsou často zaměňovány – IoC je obecný princip, který upřednostňuje kontrolu zvenku objektů nad situací, kdy si objekt sám říká o věci v rámci svého kódu. Výsledkem je větší kontrola nad objekty a tento návrh zároveň umožňuje lepší znovupoužitelnost. [11]

Pokud chce nějaký objekt v programu používat služby jiného objektu (dále poskytovatele) bez použití DI, musí nejdříve poskytovatele v sobě inicializovat a až potom začít využívat jeho služby. Při použití návrhového vzoru DI

je objekt oproštěn od inicializace poskytovatele, protože inicializaci zařizuje prvek, který se nazývá kontejner. Objekt potřebuje jen referenci na kontejner, který mu je schopen dodat hned několik různých poskytovatelů služeb. Poskytovatelé mohou objektu dodat rozdílné služby a právě díky tomuto je návrhový vzor DI tak silný. [12]

Použití návrhového vzoru DI ukazuje kód 4.3. V okamžiku zavolání konstrukturu HomeController DI kontejner dodá tomuto konstrukturu do vstupních parametrů již vytvořené instance objektů UserService a DashboardService a s těmi poté objekt HomeController pracuje.

Kód 4.3: Ukázka použití návrhového vzoru Dependency Injection

```
public class HomeController : Controller
{
    private UserService userService;
    private DashboardService dashboardService;

    public HomeController(UserService user, DashboardService
        dashboard)
    {
        this.userService = user;
        this.dashboardService = dashboard;
    }

    public ActionResult Index()
    {
        var dashboardModel = dashboardService.GetHomeDashboard(
            userService.CurrentUser());
        return View(dashboardModel);
    }
}
```

4.2.4 HangFire

Hangfire je jednoduchý open-source framework pro vytváření a spouštění plánovaných úloh běžící asynchronně na pozadí aplikace. Umožňuje úlohy ukládat do operační paměti nebo používat persistentní úložiště jako například SQL Server, Redis, Postgre a další. [13]

Úlohy se dají spouštět zpožděně pomocí definice standardního typu TimeSpan nebo periodicky vždy v určitý okamžik. Je možné vytvářet i tzv. kontinuální úlohy. Kontinuální úlohy jsou takové, u kterých je předem definováno přesné pořadí v jakém se budou spouštět a jaká úloha následuje po dokončení určité úlohy. [14]

Hangfire v sobě obsahuje také uživatelské rozhraní pro monitorování spuštěných nebo naplánovaných úloh. Dále obsahuje historii, nedokončené úlohy, statistiky a další užitečné informace. [15]

4.2.5 Log4Net

Knihovna Log4Net je jednoduchý logovací nástroj pro .NET projekty, který je volně dostupný pomocí NuGet balíčku. Konfigurace a používání této knihovny je velice jednoduchá. Jako první věc, která se musí udělat je vytvořit v projektu konfigurační soubor, pojmenovaný například `log4net.config`, a do tohoto souboru konfigurovat nastavení, jak má Log4Net pracovat. Nastavujeme například jakým způsobem se má logovat nebo od jaké úrovně a kam se budou logy ukládat. Poté se do souboru `AssemblyInfo.cs` přidá příkaz pro načtení nastavení z nově vytvořeného `log4net.config` souboru a už se můžeme začít logovat. V ukázce kódu 4.4 jsem připravil krátký program, který demonstruje použití knihovny Log4Net. [16]

Kód 4.4: Ukázka použití logovacího nástroje Log4Net

```
class Program
{
    private static readonly log4net.ILog log = log4net.LogManager.
        GetLogger(System.Reflection.MethodBase.GetCurrentMethod().
            DeclaringType);

    static void Main(string[] args)
    {
        log.Info("Program started...");
        Console.Write("Enter a number: ");
        try
        {
            int number = Convert.ToInt32(Console.ReadLine());
        }
        catch(OverflowException ex)
        {
            log.Error("User has entered a too big number!", ex);
        }
    }
}
```

4.2.6 DevExpress

Firma DevExpress je známá vývojem vizuálních komponent. I když se jedná o placenou službu, komponenty od této firmy jsou mezi vývojáři oblíbené, protože usnadňují práci a šetří spoustu času, neboť tyto komponenty už nemusí vývojáři složitě vytvářet. V poslední době se DevExpress zaměřuje převážně na komponenty pro platformu .NET. [17]

V této práci používám od firmy DevExpress pouze jedinou komponentu a tou je GridView. GridView je komponenta, která slouží pro zobrazení tabulky. Možná nejednoho čtenáře napadne, proč kvůli zobrazení tabulky používám v této práci placenou službu, když Microsoft nabízí komponentu Grid, která slouží ke stejnému účelu a je zdarma. Důvod je takový, že GridView

od DevExpress umožňuje mnohem rozsáhlejší práci s daty v tabulce a nabízí mnohem více možností.

Grid umožňuje např. přidání vlastních sloupců, schování/zobrazení sloupců, vyhledávání nad celou tabulkou, filtrování podle různě definovaných podmínek, možnost třídění řádků tabulky, seskupení dat podle předem definovaného klíče, na základě hodnot obarvovat zobrazená data, editaci dat přímo v tabulce nebo export tabulky do souboru. Ukázkou jednoduché tabulky převzatou z internetové stránky výrobce [18] se zapnutým filtrováním v záhlaví sloupců zobrazuje obrázek 4.4. [19]

Company Name	Country	City	Unit	Quantity	Discount	Total
Hanari Carnes	Brazil	Rio de Janeiro			%	\$15,810.00
QUICK-Stop	Germany	Cunewalde			%	\$15,019.50
Simons bistro	Denmark	Kobenhavn			%	\$10,540.00
Rattlesnake Canyon Grocery	USA	Albuquerque			%	\$10,540.00
Hungry Owl All-Night Grocers	Ireland	Cork			%	\$9,903.20
Piccolo und mehr	Austria	Salzburg	\$210.80	50	20 %	\$8,432.00
Mère Pailarde	Canada	Montréal	\$210.80	49	20 %	\$8,263.36
Königlich Essen	Germany	Brandenburg	\$263.50	30	0 %	\$7,905.00
QUICK-Stop	Germany	Cunewalde	\$263.50	30	0 %	\$7,905.00
Great Lakes Food Market	USA	Eugene	\$263.50	30	5 %	\$7,509.75

Page 1 of 216 (2155 items) < 1 2 3 4 5 6 7 ... 214 215 216 >

Obrázek 4.4: Ukáзка komponenty GridView od firmy DevExpress [18]

4.2.7 Bootstrap

Bootstrap se řadí mezi takzvané front-end frameworky a slouží především pro rychlejší a jednodušší tvorbu webového uživatelského rozhraní. Dnes už je minimum webových vývojářů, kteří by s Bootstrapem neměli žádné zkušenosti.

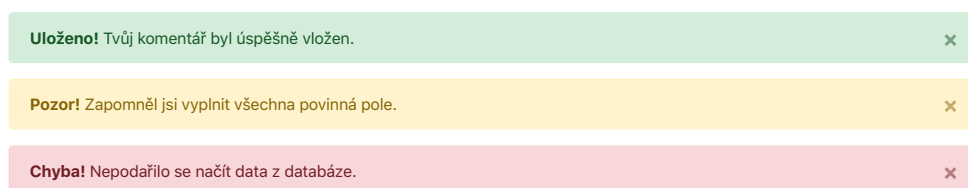
4. REALIZACE

nosti, nebo o něm ještě v životě neslyšeli. Bootstrap obsahuje HTML (*Hyper-text Markup Language*), CSS (*Cascading Style Sheets*) a JavaScript soubory, na kterých je postaven design šablon, které může vývojář umístit na svůj web a nemusí tyto prvky sám stylovat nebo dávat ke stylování grafikovi. K dispozici jsou šablony pro typografické prvky, formuláře, tlačítka, tabulky, navigace na stránce, výstražné a potvrzující hlášky, modální vyskakovací okna, probíhající obrázky a mnoho dalšího...

V této práci je většina vizuální podoby systému postavena právě na prvcích frameworku Bootstrap. Zvolil jsem toto řešení, protože vývoj webového rozhraní s Bootstrapem je rychlé, k dispozici je mnoho ukázkových materiálů a po grafické stránce je uživatelsky příjemný. Na obrázku 4.5 jsou vidět některá z předdefinovaných tlačítek a na obrázku 4.6 potvrzovací, varovná nebo chybová hláška uživateli. [20]



Obrázek 4.5: Bootstrap tlačítka



Obrázek 4.6: Bootstrap vyskakovací hlášky

4.2.8 FontAwesome

Ikonky jsou populární grafický doplněk na webu. Často má však vývojář problém najít vhodné ikonky, které mají správnou velikost, hodí se do designu webu a jsou nejlépe zdarma. Tento problém řeší právě FontAwesome.

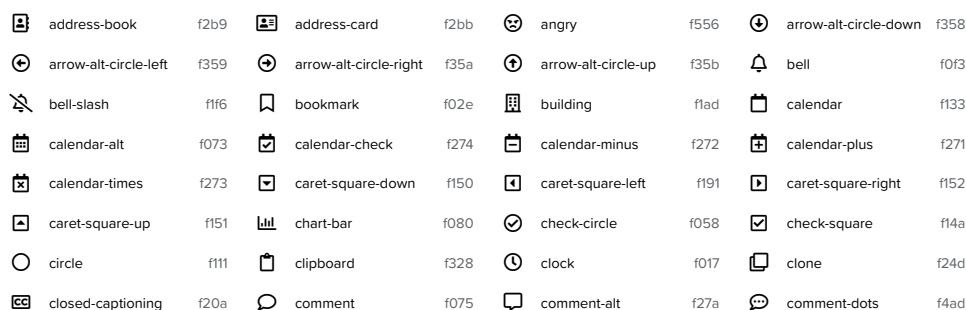
FontAwesome je volně šiřitelný druh písma použitelný na webu. Například začátkem roku 2019 v neplacené verzi obsahoval přes 1000 volně dostupných ikonků a v placené verzi Pro to bylo více jak 5000 ikonků. Je pouze na vývojáři, jaké ikonky a k jakému účelu použije, dokonce je povoleno i využití tohoto nástroje pro komerční účely. O popularitě svědčí i to, že patří mezi 6 nejnavštěvovanějších projektů na serveru GitHub.com a po celém světě má miliony uživatelů. Jelikož se jedná o speciální druh písma, lze u ikonků upravovat velikost, barvu, stín, prostě cokoli co umožňuje CSS. [21]

Nejjednodušší způsob, jak začít používat FontAwesome na vlastním webu zabere vývojáři méně než jednu minutu. Stačí pouze do hlavičky stránky vložit příkaz uvedený v ukázce 4.5, poté kamkoliv do HTML kódu napsat příkaz `<i class="fa fa-calendar"></i>` a na vybraném místě se zobrazí daná ikonka. Třída `fa-calendar` označuje název ikonky, kterou chceme na webu použít. [22]

Obrázek 4.7 ze stránek produktu zobrazuje na ukázkou několik vybraných ikonek tohoto písma.

Kód 4.5: Začlenění písma FontAwesome do vlastních stránek

```
<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.8.1/css/all.css" crossorigin="anonymous">
```



Obrázek 4.7: Ukázka vybraných FontAwesome ikoněk z oficiálních stránek produktu [23]

4.2.9 jQuery

Následující popis technologie nazvané jQuery vychází z článku [24]. JQuery je JavaScriptová knihovna sloužící k interakci mezi JavaScriptem a HTML kódem stránky. Je podporovaná všemi známými internetovými prohlížeči a vývojářům je k dispozici už od roku 2006, kdy byla představena její první verze. Filosofí jQuery je oddělení „chování“ stránky od struktury HTML podobně jako CSS odděluje vizuální styl stránky od HTML struktury. JQuery nabízí například následující funkce:

- Výběr libovolného elementu na stránce pomocí jeho identifikátoru.
- Události následované po konkrétní akci uživatele nebo prohlížeče. Například události jako je třeba kliknutí nebo pohyb myši, stisk klávesy nebo i scrollování po stránce.
- Manipulace s CSS styly stránky
- Efekty a animace HTML elementů

- AJAX (*Asynchronous JavaScript and XML*) – asynchronní dotazování serveru o data bez nutnosti znovu nahrávat celou stránku.
- Nabízení doplnění textu na základě psaní uživatele.
- a mnoho dalšího...

4.3 Databáze

Podle nefunkčního požadavku NP2, definovaného v kapitole 3.1.2 *Nefunkční požadavky*, bude pro uložení dat použita databáze uložena v relačním databázovém systému Microsoft SQL Server 2016.

4.3.1 Databázový model

V této části se zabývám databázovým modelem systému. Z obrázku modelu 4.8 je vidět, že tabulky „RoleMember“, „Sheet“ a „EmployeeException“ neobsahují vazbu na tabulku Employee a obsahují sloupeček „EmpId“ resp. „BaseEmpId“, který slouží jako cizí klíč do centrální tabulky zaměstnanců firmy v externí regionální databázi. Takovéto řešení jsem zvolil z důvodu toho, protože tabulka Employee obsahuje pouze zaměstnance s nárokem na stravenky. EmpId (Identifikační číslo zaměstnance) a centrální databáze zaměstnanců je popsána v kapitole 2.2 *Představení zadavatele*.

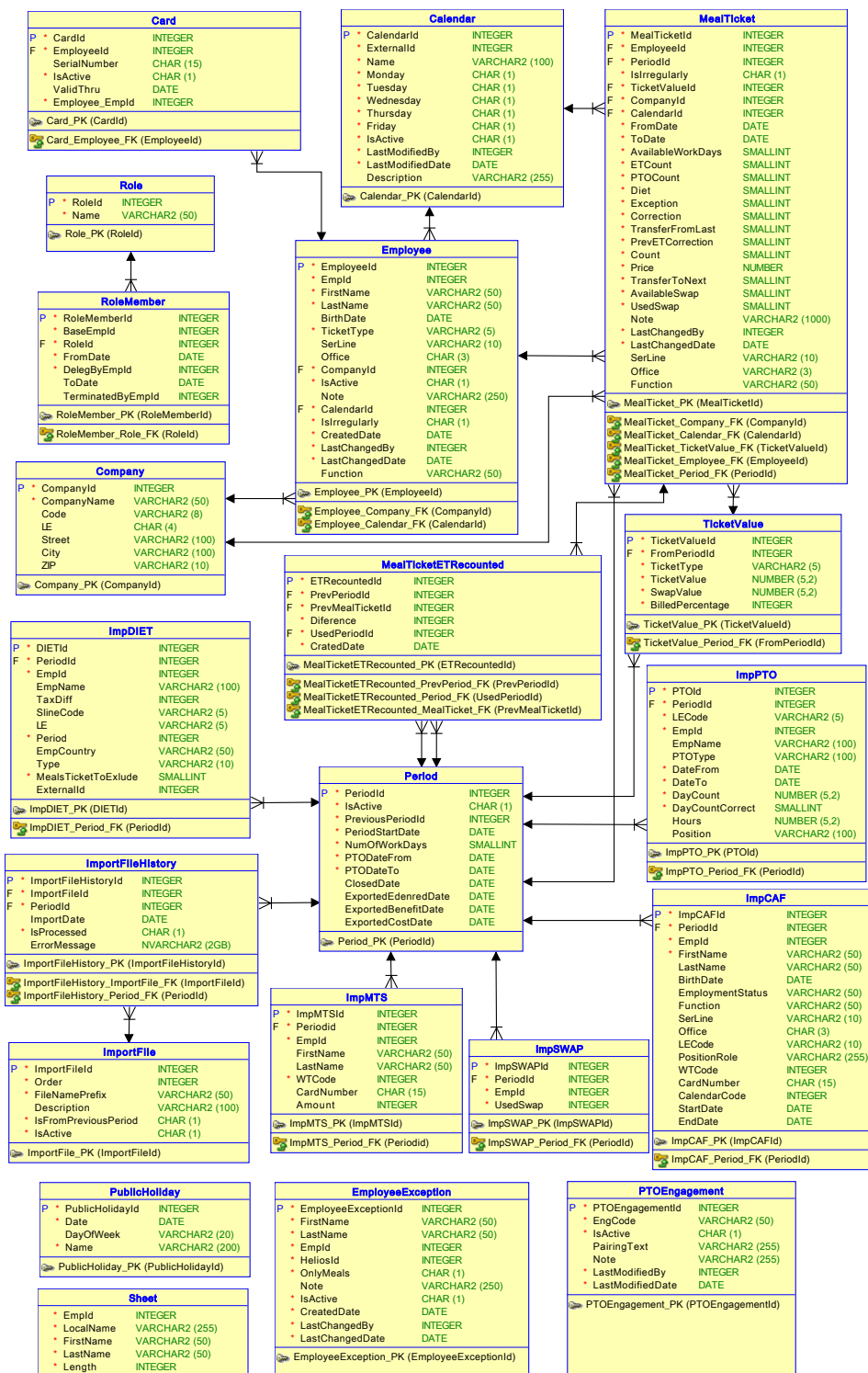
V databázovém modelu je 5 tabulek s předponou „Imp“. Jsou to tabulky *ImpDIET*, *ImpMTS*, *ImpSWAP*, *ImpCAF* a *ImpPTO*. Všechny tyto tabulky slouží k uložení dat ze souborů nutných pro otevření nové měsíční periody.

4.3.2 Uložené procedury

Uložená procedura je databázový objekt, který neobsahuje data, ale část programu, který se nad daty uloženými v tabulkách databáze má vykonávat.

V databázi informačního systému je vytvořeno 5 uložených procedur s příponou „Import“. Tyto procedury plní daty jimi odpovídající tabulky. Například procedura *CAFImport* vkládá data do tabulky *ImpCAF*. Dále je v databázi 5 procedur s příponou „Process“. Tyto procedury vezmou data z odpovídající tabulky a podle jejich obsahu změní obsah jiné tabulky databáze. Pokud budeme pokračovat v příkladu s procedurou *CAFImport*, tak procedura *CAFProcess* vezme data z tabulky *ImpCAF* a podle nich upraví nebo založí nové řádky v tabulce *MealTickets*. Tyto procedury se používají při importu měsíční periody.

Ostatní uložené procedury slouží například k resetování aktuálně otevřené měsíční periody, nebo pro přepočítání odpracovaných dní zaměstnancům, kteří pracují pro firmu na živnostenský list.



Obrázek 4.8: Relační model databáze

4.3.3 Funkce

Databázová funkce slouží především k výpočtu konkrétní hodnoty z dat uložených v databázi. Seznam nejdůležitějších databázových funkcí informačního systému spolu s jejich popiskem:

CheckWorkDay Podle vstupního parametru typu datum funkce vrátí hodnotu 1 jestli daný den patří do pracovních dní, nebo hodnotu 0 pokud nepatří.

CheckWorkDayByCalendar Ze vstupních parametrů *EmpId* a konkrétního data vrátí hodnotu 1 jestli daný den patří do pracovního dne zaměstnance podle jeho pracovního kalendáře, nebo hodnotu 0 pokud nepatří.

GetEmployeeAvailableWorkDaysOfPeriod Funkce vypočítá běžnému zaměstnanci firmy podle jeho kalendáře počet pracovních dní v konkrétním měsíci.

GetEmployeePTOOfPeriod Tato funkce vrátí pro konkrétního zaměstnance v konkrétním měsíci počet dní kdy byl na dovolené.

GetIfWasIrregularly Funkce pro kontrolu, jestli konkrétní zaměstnanec byl v minulém měsíční periodě uveden jako zaměstnanec pracující pro firmu na živnostenský list.

GetIrregularlyWorkdaysOfPeriod Funkce která zjistí z databáze docházkového systému firmy pro konkrétního zaměstnance pracující na živnostenský list a konkrétní měsíční periodu počet odpracovaných dní s nárokem na stravenku.

GetMonthAvailableWorkDays Funkce vypočítá počet pracovních dní v konkrétním měsíci.

4.4 Uživatelské rozhraní

Při tvorbě grafického návrhu uživatelského rozhraní (GUI) informačního systému jsem vycházel z již existujících interních aplikací zadavatele, protože všechny tyto aplikace mají podobné GUI. To je tvořeno tlustým černým pruhem u horního okraje stránky a pod ním je zobrazen veškerý dynamický obsah na stránce. V černém pruhu je zleva umístěno logo firmy a menu aplikace a u pravého okraje najdeme jméno autorizovaného uživatele. Použil jsem toto řešení, protože koncový uživatelé systému jsou na toto GUI zvyklí a při použití informačního systému si nebudou muset zvykat na jiné rozložení prvků na stránce. Ukázky vybraných stránek informačního systému najdete v příloze B *Ukázky systému*.

4.5 Popis funkcionality informačního systému

V této sekci popisují funkcionalitu informačního systému, která vychází z provedené analýzy a splňuje všechny funkční i nefunkční požadavky popsané v kapitole 3.1 *Požadavky*.

Z nefunkčního požadavku NP5 vyplývá, že systém musí evidovat běžné zaměstnance firmy, ale i zaměstnance, kteří pracují na živnostenský list. Tito lidé pracující pro firmu na živnostenský list jsou v systému nazváni jako „irregular“ zaměstnanci a tento pojem budu v následujících kapitolách používat.

Dále jsem v této kapitole použil pojem „salary swap“. Jedná se o funkcionalitu informačního systému, která je implementována, ale v této bakalářské práci nebude vysvětlen její význam.

4.5.1 Domovská stránka

Domovská stránka se liší podle role do které je uživatel vstupující do systému přiřazen. Je tedy jiná pro administrátora a jiná pro běžného uživatele systému, který pobírá nebo v minulosti pobíral stravenky.

Při vstupu do systému uživatele zařazeného do role administrátor, systém zobrazí stránku s přehledem všech zaměstnanců v aktuální měsíční periodě spolu s jejich detaily. Administrátor má možnost tuto tabulku exportovat do Excel souboru, který se mu uloží do počítače. Ukázka domovské stránky administrátora je na obrázku B.1 v příloze B této práce.

Pokud informační systém otevře zaměstnanec, který je v roli běžného uživatele systému, otevře se mu jako domovská stránka jeho detailní stránka. Význam jednotlivých informací obsažených na detailní stránce uživatele popisuje následující kapitola 4.5.2 *Detail uživatele*.

4.5.2 Detail uživatele

Osobní stránka uživatele slouží pro poskytnutí informací běžnému uživateli o jeho nároku za konkrétní měsíc a z jakých položek byl tento nárok vypočítán. Pokud si osobní stránku uživatele otevře administrátor systému, uvidí na této stránce informace a možnosti, které jsou běžnému uživateli skryty. V příloze B *Ukázky systému* obrázek B.2 zobrazuje stránku s detailem uživatele, jak ji vidí běžný uživatel systému a na obrázku B.3 je možné vidět, jak se detail uživatele zobrazí administrátorovi informačního systému.

Stránka na je rozdělena na dvě části. U pravého okraje stránky je sloupec, ve kterém je zobrazena fotka zaměstnance a EmpId. Tyto informace jsou viditelné jak pro běžného uživatele, tak i pro administrátora. Administrátorovi systému se navíc v tomto sloupečku zobrazí seznam s osobními nebo firemními informacemi o zaměstnanci. Dále je zde administrátorovi k dispozici tlačítko „Edit“, které po stisknutí zobrazí formulář prostřednictvím kterého má možnost změnit u zaměstnance poznámku, jméno a příjmení, určit zda se jedná

o „irregular“ zaměstnance, určit zda má nárok na stravenky nebo nastavit kalendář, podle kterého zaměstnanec pracuje.

Zbývající plocha stránky je rozdělena na záložky *Periods* a *Cards*. Nejdříve popíší záložku *Cards*. Záložce obsahuje tabulku, kde jsou čísla všech uživatelových stravenkových karet spolu s jejich platnostmi a označením jaká karta je aktuálně aktivní. Administrátor může navíc karty editovat nebo přidávat nové.

Poslední nepopsaná část detailní stránky uživatele je záložka *Periods*. Na této záložce je uživateli zobrazen poslední uzavřená měsíční perioda, ve kterém měl nárok na stravenky a administrátorovi aktuální měsíční perioda, která je aktuálně otevřená k editaci. Tu může administrátor upravit pomocí tlačítka „Edit period“ v levém horním rohu stránky. Editační formulář pro úpravu aktuálně otevřené periody u detailu zaměstnance zobrazuje obrázek B.4.

4.5.3 Správa měsíčních period

V této části práce popisují správu měsíčních period, kterou má na starosti administrátor systému. Představuji zde jak práci administrátora s měsíčními periodami, tak i životní cyklus jedné periody. Měsíční perioda začíná ve stavu, kdy je připravena k importu dat a končí stavem kdy je již uzavřená pro úpravy a administrátor z ní vygeneruje výsledné reporty.

4.5.3.1 Práce s měsíční periodou

V této kapitole představuji postup jak administrátor pracuje s měsíčními periodami a jaké kroky musí každý měsíc udělat.

Představme si, že je aktuálně konec měsíce června a administrátor stravenek ve firmě chce začít s evidencí stravenek pro tento měsíc. Jako první musí získat reporty o zaměstnancích, které vygenerují jiné oddělení nebo systémy firmy. Těchto souborů potřebných k otevření nové měsíční periody k editaci je potřeba 5. V následujícím seznamu jsou všechny tyto soubory uvedeny společně s popiskem jaká data obsahují.

CAF.xlsx Jedná se o nejdůležitější soubor ze všech souborů, protože nám dodává nejvíce informací, které jsou u zaměstnanců potřeba evidovat. Tento soubor je vygenerován podnikovým softwarem SAP a jsou v něm uvedeni všichni zaměstnanci, kteří mají nárok na stravenky nebo „salary swap“. U každého zaměstnance je v souboru uvedeno datum narození, pozice ve firmě, den nástupu nebo den ukončení pracovního poměru a další osobní informace...

MTS.xlsx Tento soubor je vygenerován personálním systémem firmy a obsahuje všechny zaměstnance, kteří mají za daný měsíc nárok na nějaký počet stravenek.

CEax.xlsx Toto je soubor, který vygeneruje mzdové oddělení firmy a je v něm uvedeno kolik který zaměstnanec čerpal za daný měsíc dovolené.

Diets.xlsx Jedná se o druhý soubor který přijde ze mzdového oddělení. Slouží pro informaci kolik diet čerpal jaký zaměstnanec.

SWAP.xlsx Poslední soubor nutný pro otevření nové měsíční periody slouží pro funkcionalitu salary swap, která není v této práci popisována a tedy ani účel tohoto souboru nebude popsán.

Následuje krok, kdy administrátor systému všechny výše uvedené potřebné soubory nahraje do určeného síťového adresáře ve firmě a otevře informační systém na stránce „Periods“. Zde stiskne tlačítko pro import periody. V tomto okamžiku systém nahraje data se souborů uvedených v prvním kroku do své databáze a zpracuje je podle své potřeby. Stránka „Periods“ před začátkem importu bude vypadat přesně tak, jako na obrázku B.5.

Třetím krokem po úspěšném importování dat ze souborů je ruční úprava jednotlivých zaměstnanců u kterých má administrátor uvedenou poznámku z minulého měsíce. Po tomto kroku musí administrátor počkat minimálně do 5 dne v následujícím měsíci. Toto datum je poslední den, kdy musí irregular zaměstnanci zadat svoje odpracované dny do externího docházkového systému firmy.

Po uplynutí 5. dne v měsíci administrátor znovu otevře informační systém na stránce „Periods“ a stiskne tlačítko „Recount“. Systém si v tuto chvíli zjistí odpracované dny všech irregular zaměstnanců a tuto hodnotu uloží k zaměstnanci do databáze.

V posledním kroku, před uzavřením periody administrátor jen zkontroluje správnost dat, například prostřednictvím kontrolního reportu, který patří k administrační funkcionalitě systému a pokud nejsou potřeba žádné další úpravy je měsíční perioda připravena na uzavření.

Uzavření periody se provede znovu na stránce „Periods“, ale tentokrát pomocí tlačítka s popiskem „Close period“. Po stisknutí tohoto tlačítka se aktuálně rozpracovaná měsíční perioda uzavře veškerým úpravám a systém umožní z této periody administrátorovi vygenerovat výsledné reporty. Systém umožňuje generovat 3 různé reporty:

Edenred report Report, který administrátor poskytne stravenkové firmě Edenred CZ a ta podle něj nabije zaměstnancům jejich stravenkové karty.

Cost report Report určený pro účtárnu firmy k zaúčtování stravenek do účetnictví.

Benefit report Tento report obsahuje data k funkcionalitě salary swap a tedy nebudu popisovat jeho účel.

4.5.4 Administrace systému

Administrace systému obsahuje pět funkcionalit ke kterým má přístup pouze administrátor. Tyto funkcionality bylo nutné do aplikace implementovat, neboť přímo vyplývají z funkčních požadavků na systém. Konkrétně se jedná o funkcionality:

- Správa administrátorů systému.
- Správa pracovních kalendářů zaměstnanců.
- Stránka s kontrolním reportem pro libovolnou měsíční periodu s možností exportu do Excel souboru.
- Editace zaměstnaneckých výjimek, které vycházejí z funkčního požadavku FP10 v kapitole 3.1.1 *Funkční požadavky*.
- Správa engagementů u kterých nevzniká nárok na stravenky.

Správu administrátorů systému zobrazuje obrázek B.7, stránku s kontrolním reportem obrázek B.8 a stránka sloužící pro editaci engagement kódů u kterých nevzniká nárok na stravenky je zobrazena obrázkem B.9.

Ekonomicko-manažerské vyhodnocení

V této kapitole jsem provedl ekonomicko-manažerské vyhodnocení celého projektu. To zahrnuje finanční analýzu implementace nového informačního systému, včetně jeho návratnosti po nasazení vzhledem k současnému systému a zhodnocení benefitů plynoucích pro firmu a její zaměstnance.

5.1 Náklady na informační systém

Díky mé znalosti prostředí firmy a zkušenosti v oboru, byla časová náročnost analýzy a vývoje systému výrazně nižší, než kdyby se firma rozhodla zadat tento projekt externí firmě, která není seznámena s interními procesy a technologiemi, které se ve firmě využívají.

Cenu na vývoj informačního systému jsem odhadl z průměrné měsíční mzdy IT .NET Developera, která činí 47 761 Kč a specialisty na odměňování a firemní benefity z personálního oddělení firmy, který pobírá průměrnou měsíční mzdu 43 262 Kč [25, 26]. V tabulce 5.1 je uveden časový odhad a cena práce .NET Developera a v tabulce 5.2 časový odhad a cena práce specialisty na odměňování a firemní benefity. Časová náročnost jednotlivých kroků je uvedena v MD, což znamená *Man-Day* a vyjadřuje čas jednoho člověka za jeden pracovní den. Tato jednotka se používá především pro plánování práce. Cenu za práci jsem zjistil za předpokladu, že každý měsíc obsahuje průměrně 21 MD. Tuto hodnotu jsem vypočítal jako podíl počtu pracovních dní v roce 2019, s počtem měsíců v roce. Dle webu [27] rok 2019 obsahuje 251 pracovních dní.

Celkové náklady na nový informační systém jsou **208 058 Kč**.

5. EKONOMICKO-MANAŽERSKÉ VYHODNOCENÍ

Tabulka 5.1: Časový odhad a cena práce .NET Developera

aktivita	časová náročnost	cena
schůzky se specialistou na firemní benefity	2 MD	4 548 Kč
analýza a návrh systému	5 MD	11 370 Kč
implementace	50 MD	113 700 Kč
testování	20 MD	45 480 Kč
Celkem	77 MD	175 098 Kč

Tabulka 5.2: Časový odhad a cena práce specialisty na benefity ve firmě

aktivita	časová náročnost	cena
analýza požadavků na systém	5 MD	10 030 Kč
schůzky s .NET developerem	3 MD	6 180 Kč
testování	8 MD	16 480 Kč
Celkem	16 MD	32 960 Kč

5.2 Benefity

Hlavním a největším benefitem navrženého informačního systému je značná úspora času, který s evidencí stravenek každý měsíc stráví zaměstnanci personálního oddělení firmy. Srovnání a výpočet nákladů na provoz systému vzhledem k nahrazovanému řešení vyčíslím následující kapitole 5.3 *Finanční návratnost*.

Druhý benefit, který plyne z nasazení systému, se dotkne všech zaměstnanců firmy, kteří pobírají nebo někdy pobírali stravenky. Tím benefitem je možnost, zobrazit si v systému vlastní osobní stránku a na této stránce zaměstnanec najde rozpis jednotlivých položek, ze kterých byl jeho nárok na počet stravenek za poslední měsíc vypočítán. Tuto možnost, zobrazit si detailní rozpis položek, má zaměstnanec i pro libovolný měsíc v minulosti.

5.3 Finanční návratnost

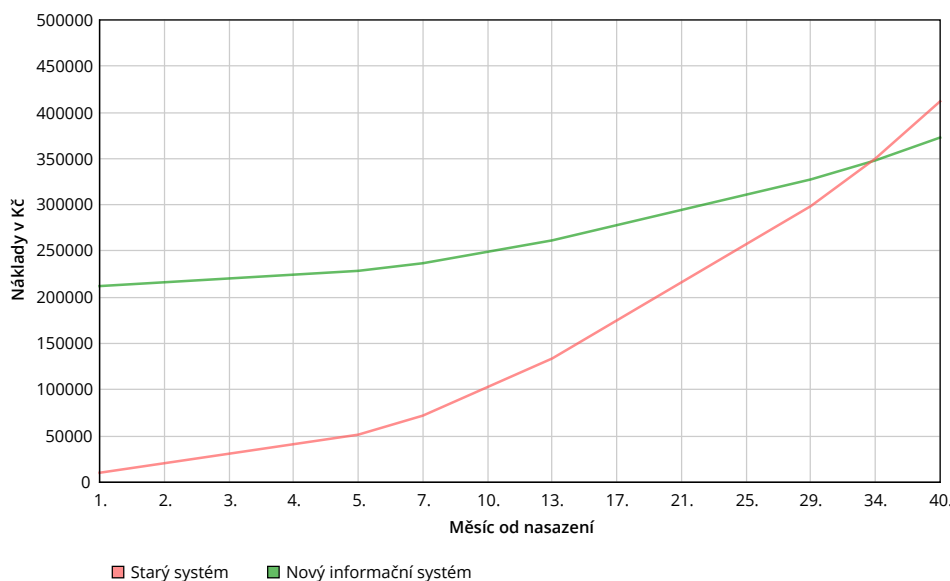
Finanční návratnost vynaložených prostředků na vývoj a nasazení nového informačního systému (IS) ve firmě jsem spočítal vyčíslením měsíčních nákladů na provoz starého systému a následným porovnáním s provozem nového IS. Tímto jsem dostal částku, kterou firma každý měsíc ušetří a pokud ještě za-

hrnu částku vynaloženou na vývoj nového IS, tak jsem zjistil za kolik měsíců se firmě vynaložená investice vrátí.

V kapitole 2.5 *Stávající evidence stravenek* jsem uvedl, že každý měsíc provozu starého systému evidence, strávil jeden zaměstnanec personální oddělení 5 pracovních dní s evidencí stravenek. Podle průměrné měsíční mzdy specialisty na firemní benefity (43 262 Kč) a průměrného počtu MD (*Man-Day*) v jednom měsíci (21 MD) jsem zjistil, že provoz starého systému stál firmu každý měsíc 10 300 Kč.

Podle prvotního testování informačního systému provedeného ve firmě se zkrátil čas evidence stravenek na 2 pracovní dny v měsíci. Každý měsíc tedy úspora času činí 3 pracovní dny administrátora stravenek vzhledem k nahrazovanému systému. Testování probíhalo pomocí paralelního provozu obou systémů současně. Po uzavření měsíční evidence se výstupy z obou systémů porovnály kvůli ověření správnosti fungování nového informačního systému. Z výsledku testování jsem dostal, že provoz nového informačního systému bude stát firmu každý měsíc pouze 4 120 Kč.

Obrázek 5.1 zobrazuje graf, ukazující náklady na provoz starého systému a nového informačního systému. Průnik červené a zelené osy těsně před 34. měsícem od nasazení ukazuje na okamžik, kdy se firmě vrátí náklady vložené do vývoje a nasazení nového informačního systému pro správu stravenek.



Obrázek 5.1: Návratnost nákladů na nový informační systém vzhledem ke starému systému od 1. měsíce nasazení

Závěr

Cílem bakalářské práce bylo navrhnout a vypracovat informační systém určený pro správu evidence stravenek ve velké korporátní společnosti, který nahradí stávající zastaralý a časově neefektivní systém. Na základě analýzy je vytvořen nový informační systém, který usnadní a urychlí práci zaměstnancům personálního oddělení, kteří mají každý měsíc správu stravenek na starosti.

Systém poskytuje zaměstnancům firmy lepší přehled o jejich měsíčním nároku v podobě detailního rozpisu všech započítaných položek. Dále administrátorům systému – zaměstnancům personálního oddělení – umožňuje generovat měsíční reporty a provádět změny u každého zaměstnance.

Vzhledem k tomu, že systém je navržen na míru a dle požadavků administrátorů systému, tak v současné době vyvinuté řešení plně pokrývá všechnu potřebnou funkčnost. V případě budoucích požadavků ze strany administrátorů, nebo uživatelů není problém systém rozšířit o další funkčnosti jako třeba administraci státních svátků přes webové rozhraní. V současném řešení tato funkčnost není implementována a státní svátky pro následující roky musejí zadávat vývojáři ručně do konkrétní tabulky v databázi.

Seznam použité literatury

1. Stravenky - historie, jejich využití, plusy a mínusy | Daňové řešení pro zaměstnavatele. *AZ-data.cz* [online] [cit. 2019-03-24]. Dostupné z: <https://www.az-data.cz/clanky/stravenky-jejich-vyuziti-plusy-minusy>.
2. Stravenky - historie, jejich využití, plusy a mínusy | Papírové stravenky černé na bílém. *AZ-data.cz* [online] [cit. 2019-03-24]. Dostupné z: <https://www.az-data.cz/clanky/stravenky-jejich-vyuziti-plusy-minusy>.
3. BERNARDOVÁ, Martina. *Současný nevyhovující systém správy stravenek ve firmě* [rozhovor]. Praha - Společnost Deloitte: Italská 2581/67, Praha 2 - Vinohrady. Datum konání: 2019-04-01.
4. PRODUCTIONS, Sharpened. IIS [online] [cit. 2019-03-31]. Dostupné z: <https://techterms.com/definition/iis>.
5. Introduction To ASP.NET MVC5. *CompleteCSharpTutorial.com* [online] [cit. 2019-03-31]. Dostupné z: <https://www.completesharptutorial.com/asp-net-mvc5/introduction-to-asp-net-mvc5.php>.
6. HORDĚJČUK, Vojtěch. Model-View-Controller [online] [cit. 2019-03-30]. Dostupné z: <http://voho.eu/wiki/model-view-controller/>.
7. FITZMACKEN, Tom. Introduction to ASP.NET Web Programming Using the Razor Syntax (C#) [online] [cit. 2019-03-31]. Dostupné z: <https://docs.microsoft.com/cs-cz/aspnet/web-pages/overview/getting-started/introducing-razor-syntax-c>.
8. Entity Framework - Overview. *TutorialsPoint.com* [online] [cit. 2019-03-31]. Dostupné z: https://www.tutorialspoint.com/entity-framework/entity_framework_overview.htm.

9. PALNE, Vinod. *Crud operation using Entity Framework in ASP.NET MVC* [online obrázek] [cit. 2019-03-31]. Dostupné z: <https://debugonweb.com/2017/06/25/crud-operation-Entity-Framework>.
10. KANJILAL, Joydip. How to work with AutoMapper in C# [online] [cit. 2019-04-14]. Dostupné z: <https://www.infoworld.com/article/3192900/how-to-work-with-automapper-in-c.html>.
11. PURCHART, Vašek. Dependency Injection: motivace [online] [cit. 2019-03-31]. Dostupné z: <https://www.zdrojak.cz/clanky/dependency-injection-motivace/>.
12. JANSSEN, Thorben. Dependency Injection with Code Examples [online] [cit. 2019-04-17]. Dostupné z: <https://stackify.com/dependency-injection/>.
13. ODINOKOV, Sergey. Hangfire - Overview [online] [cit. 2019-04-01]. Dostupné z: <https://www.hangfire.io/overview.html>.
14. ODINOKOV, Sergey. Calling methods with delay [online] [cit. 2019-04-14]. Dostupné z: <https://docs.hangfire.io/en/latest/background-methods/calling-methods-with-delay.html>.
15. Quick start – Usage. *Hangfire* [online] [cit. 2019-04-17]. Dostupné z: <https://docs.hangfire.io/en/latest/quick-start.html>.
16. log4net tutorial – Great library for logging. *C# Today* [online] [cit. 2019-04-14]. Dostupné z: <https://csharp.today/log4net-tutorial-great-library-for-logging/>.
17. SHELDON, William. Review: DevExpress ASP.NET Controls [online] [cit. 2019-04-14]. Dostupné z: <https://www.itprotoday.com/web-application-management/review-devexpress-aspnet-controls>.
18. DEVEXPRESS. *Grid View - Header Filter* [online obrázek] [cit. 2019-04-02]. Dostupné z: <https://demos.devexpress.com/MVCxGridViewDemos/Filtering/Filtering>.
19. Grid View. *DevExpress* [online] [cit. 2019-04-14]. Dostupné z: <https://demos.devexpress.com/mvcxgridviewdemos/>.
20. RASCIA, Tania. What is Bootstrap and How Do I Use It? [online] [cit. 2019-04-17]. Dostupné z: <https://www.taniarascia.com/what-is-bootstrap-and-how-do-i-use-it/>.
21. 628 ikoněk pro volné použití zdarma. *Design Portál* [online] [cit. 2019-04-03]. Dostupné z: <https://www.designportal.cz/stahnete-si-font-s-628-ikonkami-pro-volne-pouziti-zdarma/>.
22. FONTICONS, Inc. Start a New Project [online] [cit. 2019-04-17]. Dostupné z: <https://fontawesome.com/start>.
23. FONTAWESOME. *Font Awesome Free's Cheatsheet* [online obrázek] [cit. 2019-04-03]. Dostupné z: <https://fontawesome.com/cheatsheet>.

24. MORRIS, Scott. Tech 101: What is jQuery, and What is it Used for? [online] [cit. 2019-04-17]. Dostupné z: <https://skillcrush.com/2018/07/13/what-is-jquery-used-for/>.
25. Platy na pozicích - Programátor .NET. *Platy.cz* [online] [cit. 2019-03-24]. Dostupné z: <https://www.platy.cz/platy/informacni-technologie/programator-dotnet>.
26. Platy na pozicích - Specialista odměňování a benefitů. *Platy.cz* [online] [cit. 2019-03-24]. Dostupné z: <https://www.platy.cz/platy/lidske-zdroje-a-personalistika/specialista-odmenovani-a-benefitu>.
27. Pracovní kalkulačka. *Kalendar.beda.cz* [online] [cit. 2019-03-26]. Dostupné z: <https://kalendar.beda.cz/pracovni-kalkulacka>.

Seznam použitých zkratk

AJAX Asynchronous JavaScript and XML

CSS Cascading Style Sheets

DI Dependency Injection

EF Entity Framework

GUI Graphical user interface

HTML Hypertext Markup Language

IS Informační systém

IIS Internet Information Services

JS JavaScript

MD Man-Day

MVC Model-View-Controller

ORM Object-relation mapper

SQL Structured Query Language

UC Use case

Ukázky systému

John Doe ▾
MEAL TICKETS
Employees Periods Admin ▾

Search Employee

Show only current period
Export to XLS


Detail	Empid	First Name	Last Name	Irregularly emp.	Irregularly per.	Current period	
						Count	Note
	22448	Unnmrop	Aaoo	<input type="checkbox"/>	<input type="checkbox"/>	19	
	27807	Wdktblxl	Aaqxlojba	<input type="checkbox"/>	<input type="checkbox"/>	19	
	11117	Qbgn	Aarshlu	<input type="checkbox"/>	<input type="checkbox"/>	19	
	25899	Vtypovic	Acbnoyrc	<input type="checkbox"/>	<input type="checkbox"/>	19	
	27840	Gajcne	Accckdetaq	<input type="checkbox"/>	<input type="checkbox"/>	19	
	29850	Cwcmecdwa	Adayngktra	<input type="checkbox"/>	<input type="checkbox"/>	19	
	30142	Tkivdijdn	Adruaqa	<input type="checkbox"/>	<input type="checkbox"/>	19	
	31843	Rsgqx	Adladujinoj	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0	
	26712	Ayradovsbr	Adxerlwi	<input type="checkbox"/>	<input type="checkbox"/>	19	
	24607	Trjmdtm	Atelq	<input type="checkbox"/>	<input type="checkbox"/>	19	

Page 1 of 99 (998 items)

14
4
1
2
3
4
5
6
7
...
97
98
99

Page size: 10 ▾

Obrázek B.1: Úvodní stránka administrátora informačního systému



MEAL TICKETS

Ustrfxls Udlowfgcq

Periods Cards

Září 2018

▼ Hide period detail

Monday	Tuesday	Wednesday	Thursday	Friday
✓	✓	✓		✓

Calendar: PTO CZ 28 Hours 8 8 4 0 8
From date - to date: 01.09.2018 - 30.09.2018

Calculation:


Work days:	15
Transfer from last period:	(+)
Corrections:	(+)
Dilts:	(-)
Personal Time Off:	(-)
Exceptions:	(+)
Total count:	15

Swap:

Available swap:	1500
Used swap:	0
Edenred credit:	
Ticket value:	1200
Used swap from previous period:	0
Credit to Edenred card:	1200

History


- > **Sipen 2018**
- > **Červenec 2018**
- > **Červen 2018**



Ustrfxls Udlowfgcq

Employee id: 18736

Obrázek B.2: Osobní stránka zaměstnance, jak ji vidí běžný uživatel informačního systému



MEAL TICKETS

Employees Periods Admin ▾

John Doe ▾

🔍 Edit

🔍 Edit period

Září 2018

Note: [▶ Show period detail](#)

Calculation:

Work days:	19
Transfer from last period:	(+)
Corrections:	(+)
Diets:	(-)
Personal Time Off:	(-)
Exceptions:	(+)
Total count:	14

Swap:

Available swap:	1400
Used swap:	0


Edenred credit:

Ticket value:	1120
Used swap from previous period:	0
Credit to Edenred card:	1120

History

- [> Srpen 2018](#)
- [> Červenec 2018](#)
- [> Červen 2018](#)

🔍 Edit



**Wwmqsbu
Bkxhxsgu**
Employee id: 22384

Employee detail:

EmpId: 22384
Birth: 25.07.1990
LE: CZ02
Service Line: AA01
Office: BRN
Staff Level: Audit-Sen...
Calendar: PTO CZ De...
Irregular:
Ticket type: 5101

Obrázek B.3: Osobní stránka zaměstnance, jak ji vidí administrátor informačního systému

John Doe

MEAL TICKETS Employees

Wwmqsbu Bkxarxsgl
Employee Id: 22384

Period: Září 2018

Note:

Is irregular:

To date: 30.09.2018

Correction: 123 0

Exception: 123 0

Ticket type: 5101

Cancel Save

19
-5
0
0
0
0

14

Total count:

Swap:

Available swap: 1400

Used swap: 0

Edenred credit:

Ticket value: 1120

Used swap from previous period: 0

Credit to Edenred card: 1120

History

> Srpen 2018

> Červenec 2018

> Červen 2018

Employee detail:

EmpId: 22384

Birth: 25.07.1990

LE: CZ02

Service Line: AA01

Office: BRN

Staff Level: Audit-Sen...

Calendar: PTD CZ De...

Irregular:

Ticket type: 5101

Edit

Edit period

Obrázek B.4: Formulář pro editaci měsíční periody na detailní stránce zaměstnance

The screenshot displays the 'Periods' management interface. At the top, there is a navigation bar with the 'FIT' logo, the text 'MEAL TICKETS', and the user name 'John Doe'. Below the navigation bar, there are tabs for 'Current period' and 'Closed periods'. The main heading is 'Current 2019-06 period'. A table lists five items, each with a number, a file prefix, a description, an import date, and an 'Available' status. All items are marked as 'Missing' with a red 'x' icon. Below the table, there are four buttons: 'Reset period' (red), 'Start import' (blue), 'Recount' (white), and 'Close period' (white).

#	File prefix	Description	Import date	Available
1	CAF	Cafeteria entitlement employee		Missing
2	MTS	MealTicketSelect		Missing
3	DIET	Diets		Missing
4	CE19	PTO - Company		Missing
5	SWAP	MealTicketSWAPUsed		Missing

Obrázek B.5: Stránka pro import souborů nové měsíční periody

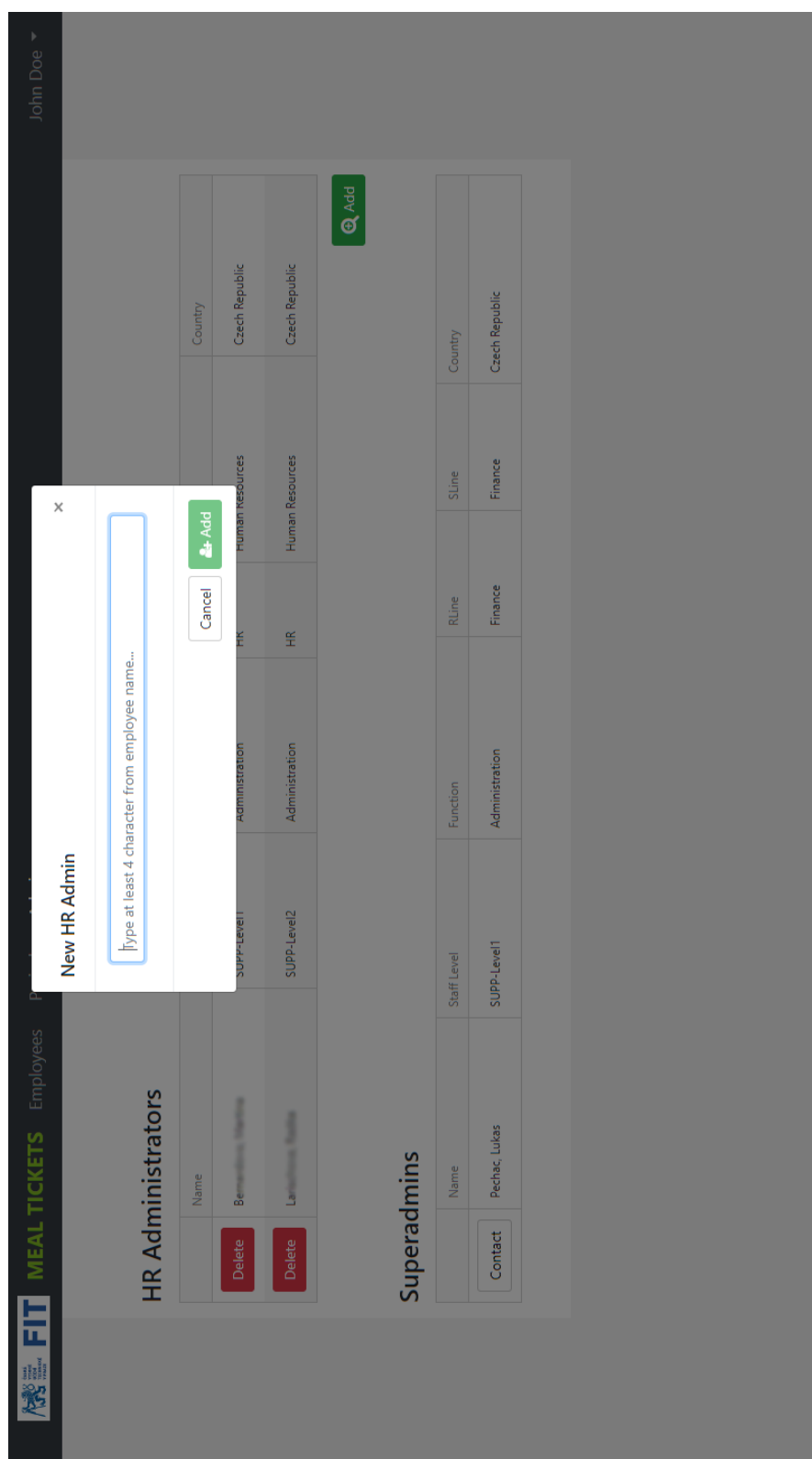
John Doe ▾

FIT MEAL TICKETS Employees Periods Admin ▾

Periods

	Current period	Closed periods
1	2018-08	<input type="button" value="Generate Edemed"/> <input type="button" value="Generate Benefit"/> <input type="button" value="Generate Cost"/>
2	2018-07	<input type="button" value="Generate Edemed"/> <input type="button" value="Generate Benefit"/> <input type="button" value="Generate Cost"/>
3	2018-06	<input type="button" value="Generate Edemed"/> <input type="button" value="Generate Benefit"/> <input type="button" value="Generate Cost"/>

Obrázek B.6: Stránka pro generování reportů pro již uzavřené měsíční periody



Obrázek B.7: Formulář pro přidání nového administrátora do informačního systému

John Doe ▾

FIT MEAL TICKETS Employees Periods Admin ▾

Control Report

2018-09

2018 ▾ 9 ▾ Show

Export to XLS


#	Emp Id	Employee Name	Irregular	Tickets Value	Count	Work Days	Diets ↓	PTO Count	Exceptions	Correction	From Last	To Next
Q	30027	Yy/qcexvu Bcqtcedgve	<input type="checkbox"/>	1520,00	19	19	0	0	0	0	0	0
Q	21698	Yy/ttrholv Xoembbij	<input type="checkbox"/>	0,00	19	19	0	0	0	0	0	0
Q	26630	Yxogj Qertugvag	<input type="checkbox"/>	0,00	0	19	0	0	0	0	-20	-1
Q	16501	Yzihhgnocq Aufgufgy	<input type="checkbox"/>	0,00	19	19	0	0	0	0	0	0
Q	18226	Yvpbry Xyck	<input type="checkbox"/>	1520,00	19	19	0	0	0	0	0	0
Q	26943	Yymfhp Qxmádkko	<input type="checkbox"/>	1520,00	19	19	0	0	0	0	0	0
Q	24329	Yyhqcthk Qfidmdgem	<input type="checkbox"/>	1520,00	19	19	0	0	0	0	0	0
Q	23673	Yyfiugeu Prqpcen	<input type="checkbox"/>	1520,00	19	19	0	0	0	0	0	0
Q	3650	Yudm Eewesef	<input checked="" type="checkbox"/>	80,00	1	1	0	0	0	0	0	0
Q	31497	Yelul Jfsabb	<input type="checkbox"/>	1520,00	19	19	0	0	0	0	0	0

Page 1 of 99 (988 items)

Page size: 10 ▾















Obrázek B.8: Kontrolní report pro administrátora systému s možností exportování dat do Excelu

John Doe ▾

 MEAL TICKETS
Employees Periods Admin ▾

PTO engagement codes

Search Eng Code...
+ Add Engagement

	Engagement code	Engagement name	Legal Entity	Engagement Status
Delete	58 	CZ PTO Compassionate Leave 2018		Active
Delete	58 	CZ PTO In Lieu of Overtime 2018		Active
Delete	58 	CZ PTO Paternity Leave 2018		Active
Delete	58 	CZ PTO Short Friday 2018		Active
Delete	58 	CZ PTO Study Leave /paid/ 2018		Active
Delete	58 	CZ PTO Visit at the Doctors /paid/ 2018		Active
Delete	58 	Unscheduled PTO Events (For ETRS only)		Active

Page 1 of 1 (7 items)

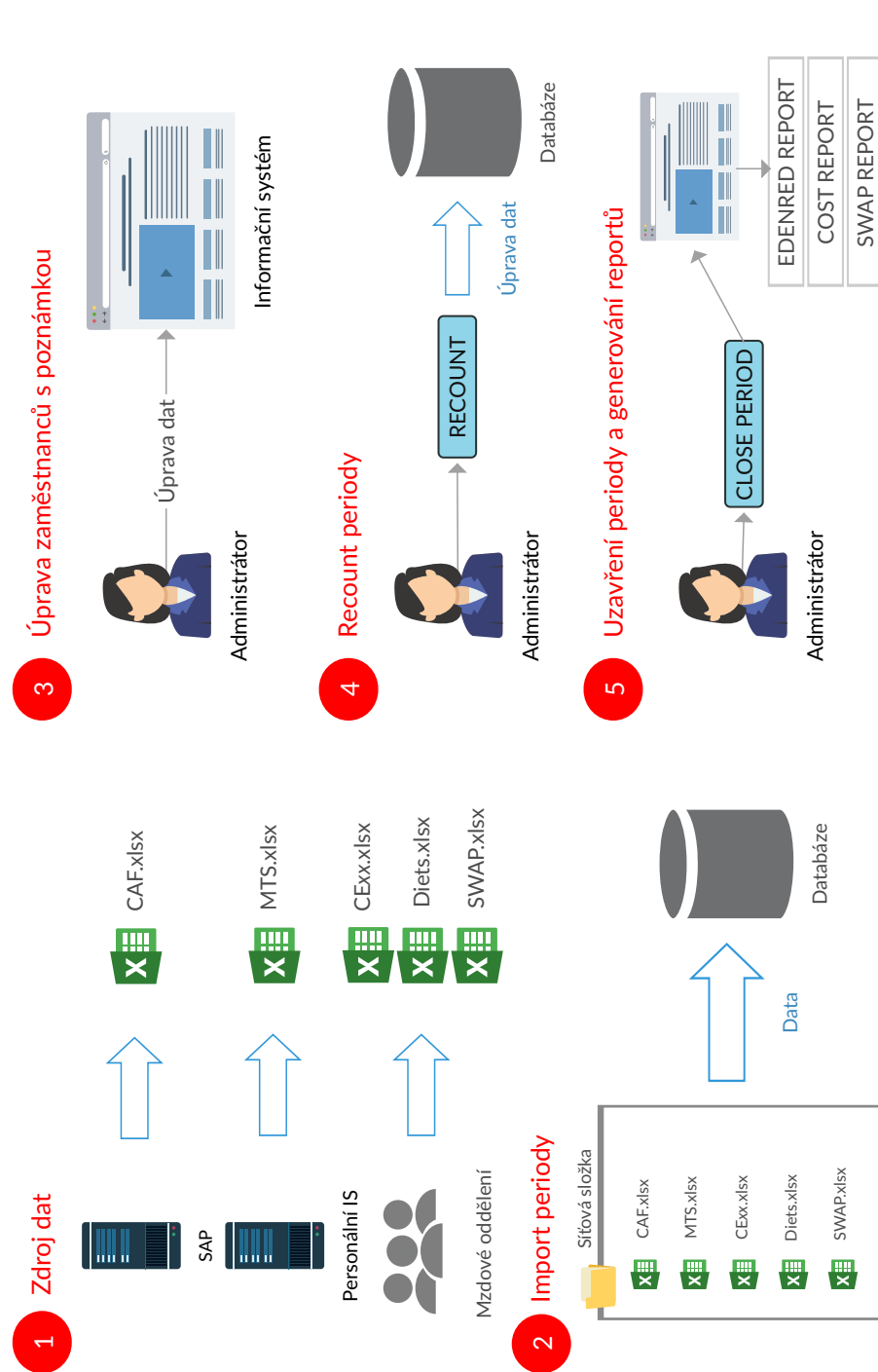
⏪ ⏩ 1 ⏪ ⏩

Page size: 10 ▾

Obrázek B.9: Administrační tabulka s engagement kódy u kterých nevzniká nárok na stravenky

PŘÍLOHA **C**

Přílohy



Obrázek C.1: Grafické znázornění práce administrátora s měsíční periodou pomocí nového informačního systému

Obsah přiloženého CD

Bachelor thesis	
└─ BP_Pechac_Lukas_2019.pdf	bakalářská práce ve formátu PDF
└─ BP_Pechac_Lukas_2019.tex.....	zdrojový kód bakalářská práce ve formátu \LaTeX
Information system.....	zdrojové kódy IS
└─ BP_MealTickets.sln....	soubor spustitelný v programu Microsoft VS
└─ Importer	adresář se zdrojovými kódy importní části IS
└─ ImporterTest	
└─ MealTickets.....	zdrojové kódy IS
└─ MealTickets.Tests	
└─ packages..	adresář s dll soubory všech balíčků použitých při vývoji IS
Database	
└─ CreateScript.sql.....	sql script pro vytvoření databáze