

Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta
Ústav aplikované informatiky

**Vytvoření CRM aplikace včetně API pro
napojení e-shopu**

Diplomová práce

Bc. Jan Bečka

Školitel: doc. Ing. Ladislav Beránek, CSc

České Budějovice 2021

Bibliografické údaje

Bc. Bečka, Jan, 2021: Vytvoření CRM aplikace včetně API pro napojení e-shopu. [Creating a CRM application, including an API for connecting an e-shop. Mgr. Thesis, in Czech.] – 102 p., Faculty of Science, University of South Bohemia, České Budějovice, Czech Republic.

Anotace

Tato diplomová práce se zabývá tvorbou univerzální webové CRM aplikace pro e-shop. Součástí webové aplikace je REST API, pomocí kterého se e-shop na CRM aplikaci napojuje. API je napojeno na e-commerce službu Mailchimp, která zajišťuje rozesílání newsletterů a e-mailových kampaní. Dále je API napojeno na službu Mandrill, která se stará o transakční e-maily. Webová aplikace přehledně zobrazuje data z e-shopu a umožňuje s nimi případnou manipulaci. Jednou ze součástí CRM aplikace je statistický modul, který vyhodnocuje data nashromážděná prostřednictvím API. Aplikace má za cíl pomoci provozovatelům e-shopů získat nové zákazníky, udržet si stávající zákazníky a zefektivnit e-commerce propagaci. Aplikace dále zprostředkovává UAC pro zákazníky e-shopu a zajišťuje rozesílání transakčních e-mailů z e-shopu. Součástí práce je dokumentace, která usnadňuje napojení e-shopu na CRM API.

Annotation

This diploma thesis deals with the creation of a universal web CRM application for an e-shop. Part of the web application is the REST API, which is used to connect the e-shop to the CRM application. The API is connected to the Mailchimp e-commerce service, which provides newsletters and e-mail campaigns. Furthermore, the API is connected to the Mandrill service, which takes care of transactional e-mails. The web application clearly displays data from the e-shop and allows for possible manipulation with them. One of the components of the CRM application is a statistical module that evaluates the data collected through the API. The application aims to help e-shop operators gain new customers, keep existing customers and improve e-commerce promotion effectiveness. The application also mediates UAC for e-shop customers and ensures the sending of transaction e-mails from the e-shop. Part of the work is documentation that facilitates the connection of the e-shop to the CRM API.

Prohlášení

Prohlašuji, že jsem autorem této kvalifikační práce a že jsem ji vypracoval pouze s použitím pramenů a literatury uvedených v seznamu použitých zdrojů.

V Českých Budějovicích dne 14.04.2021

Bečka
.....

Bc. Jan Bečka

Poděkování

Na tomto místě bych rád poděkoval panu doc. Ing. Ladislavu Beránkovi, CSc, za možnost vytvářet diplomovou práci pod jeho vedením, ochotu a čas, který mi poskytl při konzultacích. Dále bych chtěl poděkovat celé své rodině a zejména rodičům, kteří mě podporovali po celou dobu studia. V neposlední řadě bych rád poděkoval své přítelkyni za podporu a trpělivost během vytváření této práce.

Obsah

1	Úvod.....	8
1.1	Cíle práce	9
1.2	Abstraktní popis řešení.....	9
2	Analýza CRM	11
2.1	Vztah se zákazníkem.....	11
2.2	Vývoj vztahu se zákazníkem.....	12
2.3	CRM.....	13
2.4	Přínosy CRM.....	14
2.5	Aktivita spojené s CRM.....	15
3	Analýza a specifikace požadavků CRM aplikace.....	16
3.1	Mailchimp	16
3.2	Mandrill.....	18
3.3	Podobné typy aplikací	19
3.3.1	Shoptet	19
3.3.2	SmartEmailing	20
3.4	Přínos CRM aplikace oproti existujícím řešením	21
3.5	Metodika vývoje aplikace	22
3.5.1	Agilní metodika Scrum.....	23
3.5.2	Použití metodiky Scrum v tomto projektu.....	24
3.6	Funkční požadavky	25
3.6.1	API.....	25
3.6.2	Ovládání API nastavení CRM aplikace	27
3.6.3	CRM aplikace	29
3.6.4	Dokumentace k CRM aplikaci a AP	33
3.7	Nefunkční požadavky.....	34
3.8	Popis zvolené technologie.....	34
3.9	Uživatelské role.....	35
3.10	Uživatelské scénáře	36
3.10.1	Správa CRM aplikace	36
3.10.2	CRM aplikace	39
3.10.3	Dokumentace CRM aplikace	43
4	Design	45
4.1	Architektura systému	45
4.1.1	MVC	45
4.1.2	RESTful API.....	46

4.2	Procesy	49
4.2.1	Registrace zákazníka.....	49
4.2.2	Přidání odběratele newsletterů bez registrace.....	49
4.2.3	Odstranění zákaznického účtu	50
4.2.4	Vytvoření objednávky s věrnostním programem.....	51
4.2.5	Změna stavu objednávky	52
4.2.6	Spuštění připravené kampaně pro Mailchimp	53
4.2.7	Zapomenuté heslo k zákaznickému účtu	53
4.3	Datový model databáze	54
4.3.1	Věrnostní program	54
4.3.2	Transakční e-maily	55
4.3.3	CRM API.....	56
4.4	Návrh CRM API	57
5	Použité technologie.....	59
5.1	Backend.....	59
5.1.1	Nette framework	59
5.1.2	MySQL	60
5.2	Klientská část	60
5.2.1	Latte	60
5.2.2	jQuery	60
5.2.3	LESS	61
5.3	Podpůrné nástroje.....	61
5.3.1	Gulp	61
5.3.2	Balíčkovací systémy a knihovny	61
6	Použitý software pro vývoj.....	63
6.1	PhpStorm.....	63
6.2	Postman	64
6.3	Sourcetree.....	64
6.4	Docker	65
7	Implementace.....	66
7.1	Zásady vývoje	66
7.1.1	Verzování kódu.....	66
7.2	CRM API	67
7.2.1	Směrování endpointů API.....	67
7.2.2	Řízení přístupu do API	68
7.2.3	Druhy odpovědí API.....	69
7.2.4	Logování požadavků a odpovědí na API.....	70

7.2.5	Vybrané endpointy API	71
7.3	Webová CRM aplikace	75
7.4	Integrace služby Mailchimp	76
7.5	Integrace služby Mandrill	78
7.6	Statistické moduly	80
7.6.1	Vyhodnocování dat	80
7.6.2	Vykreslování grafů	81
7.6.3	Ukázky statistických modulů	83
7.7	Nasazení do produkčního prostředí	83
8	Testování	84
8.1	CRM aplikace	84
8.2	CRM API	85
9	Měření úrovně a výkonnosti CRM	86
10	Plánovaný rozvoj aplikace	86
11	Závěr	87
12	Seznam použité literatury	88
13	Seznam obrázků a tabulek	93
13.1	Obrázky	93
13.2	Tabulky	96
14	Seznam použitých zkratk	97
15	Přílohy	99
15.1	Obsah příložených souborů	99
15.2	Tabulky měření úrovně a výkonnosti CRM	100

1 Úvod

Žijeme v moderní době, které vládne internet. Většina lidí si bez něj nedokáže ani představit svůj život. Lidé si zvykli některé své potřeby upokojuvat prostřednictvím internetu a s dnešním trendem chytrých telefonů to mohou dělat prakticky kdekoliv. Těmito potřebami jsou např. čtení zpráv, sledování filmů, poslouchání muziky, komunikace přes sociální sítě, vyhledávání různých informací a další. Jednou z nich je nakupování, konkrétně tedy nakupování přes internet. Spousta kamenných obchodů má v dnešní době e-shop, a tak si zákazník může z internetu objednat prakticky cokoli, od elektroniky přes kosmetiku, nábytek, až po jízdenku na vlak. Oblíbenost nakupování přes internet dokládají také statistiky. Podle Asociace pro elektronickou komerci (APEK) bylo v roce 2019 v České republice více jak 35 tisíc e-shopů a dohromady vygenerovaly roční obrát ve výši 155 miliard korun. Tato částka činí oproti předchozímu roku nárůst o 15 % a v roce 2020 se očekává, že čísla ještě porostou. [1]

Výše uvedená statistika ale nepotvrzuje skutečnost, že každý e-shop je úspěšný. Například v roce 2019 u nás zaniklo více jak 4 tisíce e-shopů. Častým důvodem je, že si lidé myslí, že vytvořit a následně provozovat internetový obchod je jednoduchá věc. Provozovatelé si neuvědomují, jaké jsou celkové náklady na to, aby se internetový obchod mohl stát rentabilním. Často zanedbávanou disciplínou je online marketing, na který část majitelů internetových obchodů zapomíná, a to buď z nevědomosti, nebo protože mu nekladou velkou váhu. Online marketing je velmi důležitou součástí každého úspěšného internetového obchodu. Jeho hlavními nástroji jsou sociální sítě, placené PPC kampaně, kvalitní SEO, blogy a e-mailový marketing. [2]

Právě s e-mailovým marketingem a budováním vztahu se zákazníkem by měla pomoci CRM aplikace, která bude výsledkem této práce. O e-mailový marketing se bude z velké části starat služba Mailchimp, která zajišťuje rozesílání newsletterů a e-mailových kampaní. CRM aplikace bude na službu Mailchimp napojena a úzce s ní spolupracovat. Stejná společnost, která vyvíjí Mailchimp, vyvíjí také službu Mandrill. Služba Mandrill zajišťuje rozesílání transakčních e-mailů spojených s provozováním internetového obchodu a bude rovněž napojena na CRM aplikaci.

1.1 Cíle práce

Cílem práce je vytvořit funkční webovou CRM aplikaci včetně REST API, pomocí kterého se napojí do CRM aplikace internetový obchod. Aplikace bude zaštiťovat e-mailovou komunikaci mezi odesílatelem, tedy internetovým obchodem a příjemcem, tedy zákazníkem. K procesu komunikace budou využity služby Mailchimp a Mandrill, přes které bude probíhat rozesílání veškerých e-mailů. Do služby Mailchimp, se budou odesílat veškerá relevantní data, která se nasbírají přes API z internetového obchodu. Tyto data, pak mohou být využita pro e-mailové kampaně. Dále bude CRM aplikace poskytovat internetovému obchodu konkurenční výhodu, která bude zajištěna například věrnostním programem pro zákazníky nebo předpřipravenými e-mailovými kampaněmi, které budou rovněž součástí CRM aplikace.

CRM bude dále obsahovat statistický modul, který bude vyhodnocovat data nasbíraná prostřednictvím API a výsledky bude přehledně zobrazovat uživateli, aby mohl optimalizovat svou online propagaci. Kromě statistik budou k dispozici přehledy nasbíraných dat.

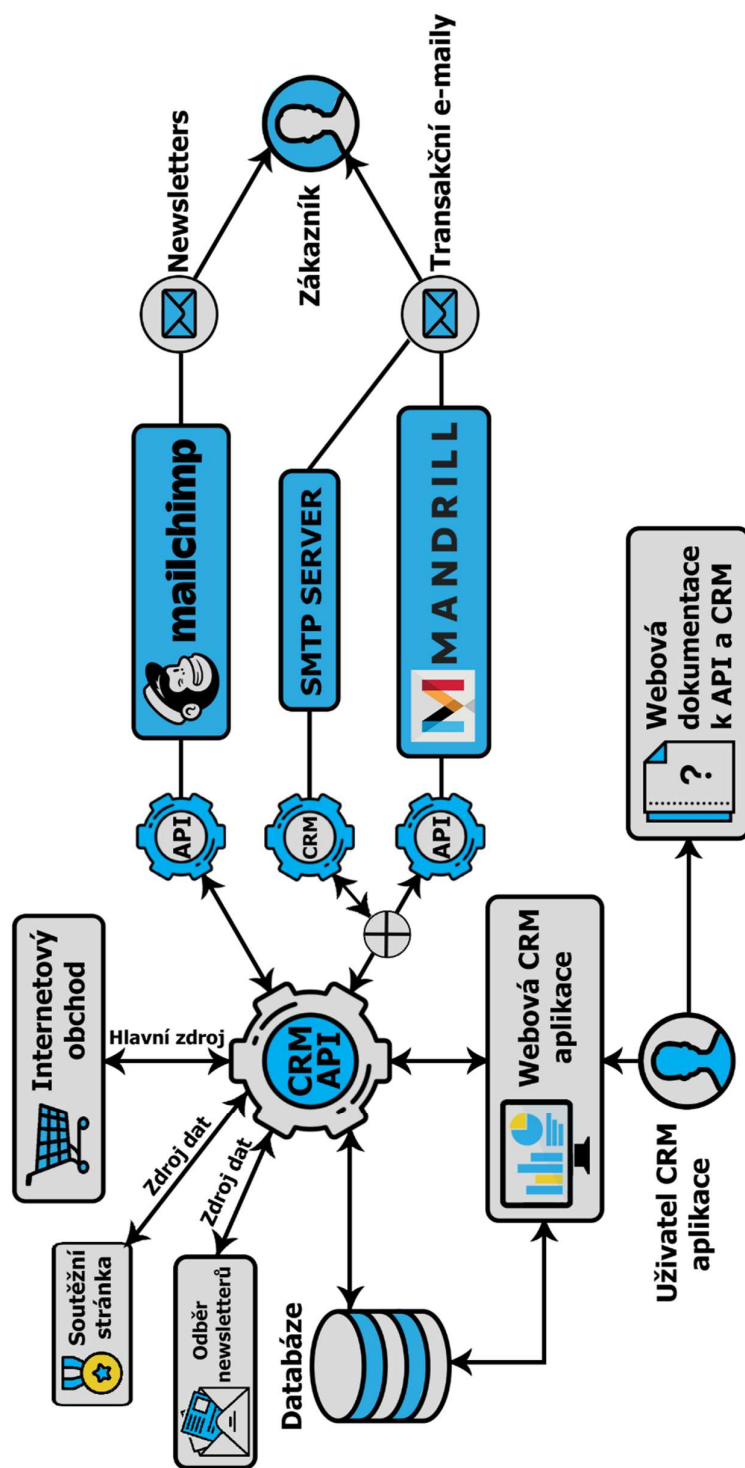
API bude rovněž obsahovat UAC modul, neboli modul pro řízení uživatelských účtů pro zákazníky internetového obchodu. UAC modul bude zprostředkovávat zákazníkům například registraci nového zákaznického účtu, přihlášení a odhlášení z účtu, změnu hesla a další funkcionality spojené s řízením uživatelských účtů.

Součástí práce bude rovněž dokumentace k CRM API a webové CRM aplikaci, která bude zjednodušovat napojování internetového obchodu na API, zjednodušit používání a nastavování CRM aplikace. Dokumentace bude vytvořena formou webové stránky a bude součástí CRM aplikace.

1.2 Abstraktní popis řešení

Jak už tedy z cílů vyplývá, projekt se bude skládat z CRM API, které bude zprostředkovávat spojení s internetovým obchodem. Dále zde bude webová CRM aplikace, do které mají přístup uživatelé systému. Uživatelé mají rovněž přístup k dokumentaci API a CRM aplikace. CRM API bude zprostředkovávat komunikaci se službami Mailchimp a Mandrill, prostřednictvím kterých se budou distribuovat veškeré e-maily pro zákazníka z napojeného internetového obchodu. Transakční e-maily je možné rovněž rozesílat prostřednictvím vlastního SMTP serveru.

Abstraktní popis řešení je znázorněno na následujícím obrázku, z kterého je patrné, co bude aplikace obsahovat a s čím bude komunikovat.



Obrázek 1: Abstraktní popis řešení (Zdroje: Logo Mandrill - [3], Logo Mailchimp – [4], Ostatní: vlastní)

2 Analýza CRM

2.1 Vztah se zákazníkem

Ve druhé polovině 20. století došlo k výrazné proměně vztahu mezi obchodníkem a zákazníkem. Právě zákazník se stal klíčovou osobou celého nákupního procesu. Převís nabídky nad poptávkou a vysoké konkurenční prostředí nutí obchodníky, aby zákazníka poznávali a vycházeli mu vstříc. [5]

Vzhledem k digitální revoluci získal zákazník v dnešní době mnohem větší moc nad obchodníkem, také snadněji získává informace o produktech. V současné době obchodníci nabízejí více služeb a výrobků, než kdykoliv předtím a tím vzniká větší konkurenční prostředí. V neposlední řadě je díky digitalizaci výměna mezi obchodníky a zákazníky stále více interaktivní a okamžitá. Pro podnikatele přichází další výhoda v podobě snadného a rychlého shromažďování informací o spotřebitelích. [6]

V interakci mezi podnikem a zákazníkem nacházíme faktory, které jsou základem pro utváření pout mezi nimi. Tyto faktory označujeme je jako „9E“ a nahlížíme na ně odlišně z pohledu zákazníka a z pohledu podniku. Faktory a pohledy na ně jsou popsány v následující tabulce.

EN	CZ	Pohled zákazníka	Pohled podniku
Economy	Ekonomie	Slevy; hodnota za peníze	Zákaznická základna
Efficiency	Efektivita	Dostupnost a	Efektivita nákladů
Esthetics	Estetika	Krása; sebeidentifikace	Řízení vývoje
Epics	Epika	Reference	Metody komunikace
Ethics	Etika	Ve vztahu s vlastní etikou	Příslušnost k podniku
Emocion	Emoce	Přizpůsobení hodnot	Spolehlivost
Education	Edukace/Vzdělání	Znalosti a schopnosti	Poznání zákazníka
Energy	Energie	Vynaložená námaha	Údržba vztahu se
Enthusiasm	Entuziazmus	Sebeprofilace a příspěvek	Prohlubování, další prodeje

Tabulka 1: Porovnávání oblastí interakce z pohledu zákazníka a podniku
(Zdroj: [7])

2.2 Vývoj vztahu se zákazníkem

J. Lehtinen rozepisuje vztah se zákazníkem do třech stádií. V první stádiu je vztah založen, poté dochází k jeho rozvíjení a na konci musí být ukončen. Ve stádium založení vztahu obchodníka se zákazníkem poukazuje na první dojem, který je ústředním prvkem řízení tvorby zákaznických vztahů (angl. Customer Creation Managment, CCM). Bylo dokázáno, že prožitek prvního dojmu velmi ovlivňuje pokračování vztahu a později je obtížné názor zákazníka změnit. Je tedy mimořádně důležité, dbát na to, aby byl první dojem příznivý. Klíčovou roli u vytváření prvního dojmu hraje emocionální oblast. Dalším stádiem rozvoje vztahu se zabývá přístup řízení vztahu se zákazníkem (angl. Customer Relationship Managment, CRM), který bude popsán níže. Poslední fází ukončení vztahu se zatím příliš mnoho firem nezabývá. Právě řízení ukončení zákaznických vztahů (angl. Customer Termination Managment, CTM) může mít obrovský význam, ať už je vztah ukončen ze strany zákazníka, či ze strany firmy, měl by být ukončen tak, aby obě strany byly vypořádány přijatelně. [8]



Obrázek 2: Celková analýza řízení vztahů se zákazníky (Zdroj: [8])

H. Wessling popisuje fáze vztahu o něco podrobněji. Vztah se podle něj rozvíjí od vnímání hodnot, které podnik zákazníkovi nabízí a vedou k navázání kontaktu. Poté se vztah rozvíjí a postupem času je konsolidován, tedy ustálen. V předposlední fázi vztah upadá, a poté je ukončen. Autor soubor všech fází nazývá jako CRM a dále vedení vztahu nijak nedělí. [8]



Obrázek 3: Jednotlivé fáze vztahu (Zdroj: [8])

2.3 CRM

Řízení vztahů se zákazníky (angl. Customer Relationship Management, CRM) znamená aktivní formou udržovat dlouhodobě prospěšné vztahy se zákazníky. CRM představuje tři hlavní prvky, mezi kterými existuje bezprostřední souvislost a k nim se přidává prvek čtvrtý, doplňující:

1. lidé (lidský kapitál, zákazníci)
2. obchodní procesy (zaměření, prolínání)
3. technologie (druh, rozsah, oblast, použití a ustálenost)
4. obsahy (data, obsah)

Pracovníci firmy by měli disponovat příslušnými znalostmi, které od nich zákazník očekává. Obchodní procesy je třeba koncipovat tak, aby zákazníkovi usnadňovaly život, ulehčovaly mu práci a příjemně jej překvapovaly. Zákazník pomocí technologií získává přístup k požadovanému uspokojení. Pokud obchodník nenabízí přístup ke zboží či službě prostřednictvím internetu, tak je dokázáno, že zákazník hodnotí službu podniku nízko, protože nebylo dosaženo požadované kvality. Zákazník bude firmu hodnotit na vysoké úrovni, pokud mu budou informace předávány rychle a kvalitně. Obsahy dat se vztahují, jak na informace o zákazníkovi, tak na informace, které zákazník požaduje. [8]

Podle Novotného (2015) můžeme CRM označit pouze jako filozofii, strategii, či přístup firmy k jejím zákazníkům, pomocí kterého se navzájem liší mezi ostatními firmami na trhu. Cílem této filozofie je identifikovat, získat a udržet si zákazníka a dále s ním řídit interakce. Hlavní přístupy CRM se liší podle pilířů, na kterých je strategie postavená. Prvním přístupem je CRM orientované na technologie, kterým je přikládána velmi

vysoká důležitost. U druhého přístupu je hlavním pilířem životní cyklus zákazníka a u posledního přístupu se orientujeme na firemní strategii. [9]

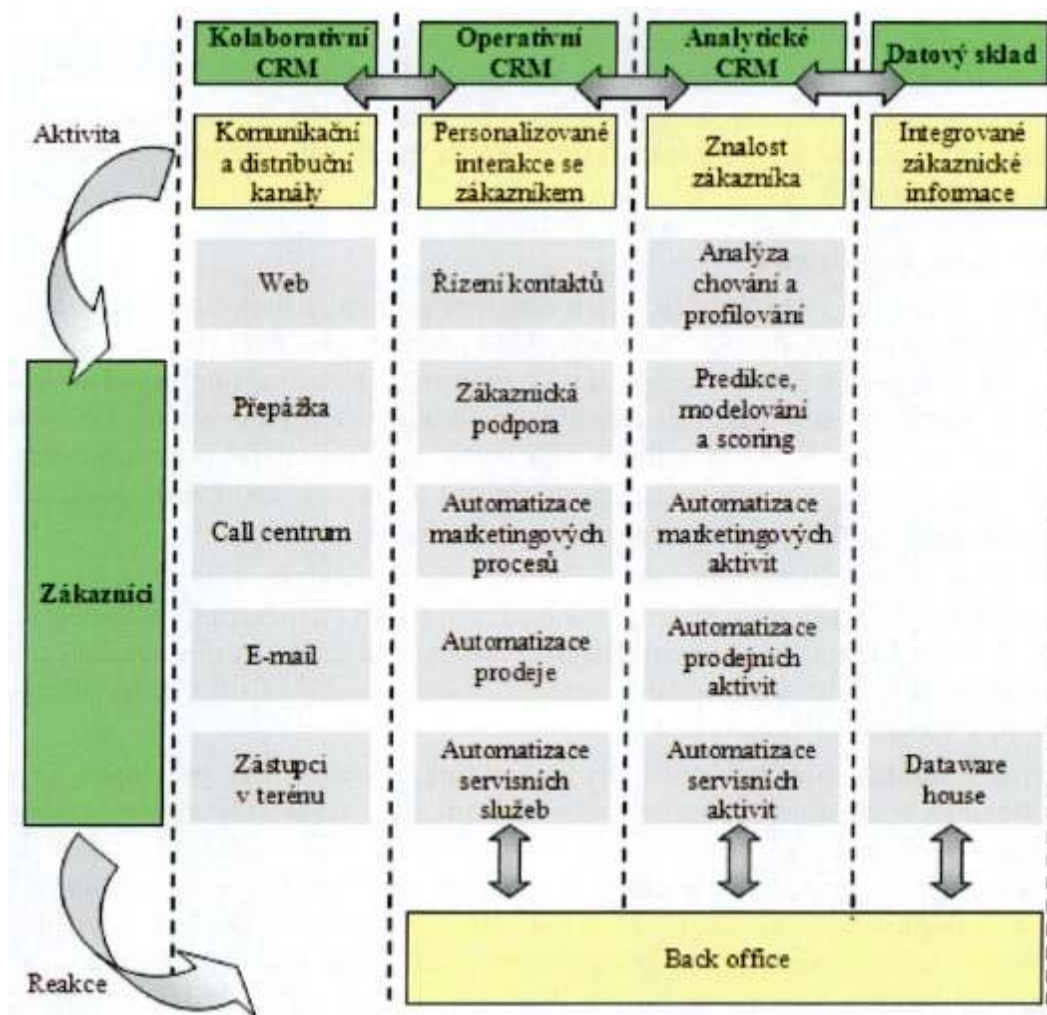
Lošťáková uvádí tři úrovně uplatňování CRM. První úroveň je strategie **masové personalizace**, při které je zákazník rozpoznán podle jména, adresy a předchozího nákupního chování. Tyto informace jsou použity k vytvoření individuální marketingové komunikace se zákazníkem, který si poté vytváří dojem individuální péče o něj. Další strategie je cílená na zákazníky, kteří využívají extra služby, za které jsou ochotni zaplatit. Tato úroveň se nazývá **masová kustomizace** a spočívá v uspokojení individuálních potřeb zákazníků. Firmy dovolují spotřebiteli spoluutvářet produkt, ovšem pouze ze sortimentu, který mají v nabídce. Na tomto principu pracují např. výrobci nábytku. V konečném výsledku se podnik snaží uspokojit každého zákazníka zvlášť, ale míra péče je pro všechny zákazníky stejná. Poslední úroveň je **diferencovaná kustomizace**, či diferencované CRM, pro nějž je příznačné aktivní spoluvytváření jedinečné hodnoty pro zákazníka na základě úzké spolupráce. Tuto strategii využívají obchodníci operující na B2B trzích, ale také výrobní firmy, které zboží nabízejí svým distributorům a prodejním mezičlánkům. Tuto strategii je vhodné použít pro velmi významné zákazníky, naopak strategii masové personalizace se používá pro méně významné zákazníky. Autorka uvádí, že pro účelné podnikání je vhodné spojení všech tří strategií CRM zároveň. [10]

2.4 Přínosy CRM

Mezi základní výhody CRM podle webu CRM-software patří především: [11]

- Identifikace klíčových zákazníků a jejich potřeb
- Detailní profil klienta na základě uchování historie komunikace, nabídek, nákupů, objednávek a kontaktů servisu
- Nabídka produktů přizpůsobených individuálním požadavkům zákazníků na základě dokonalé znalosti jejich potřeb
- Zvýšení efektivity marketingové kampaně
- Zkrácení prodejního cyklu
- Budování loajality zákazníků
- Celkové snížení nákladů a zvýšení efektivity péče o zákazníky

2.5 Aktivity spojené s CRM



Obrázek 4: Technologie řízení vztahů se zákazníky čili CRM (Zdroj: [9])

Z výše uvedeného obrázku lze vyhodnotit, že řízení vztahů se zákazníky a následná péče o ně vyžaduje především tyto aktivity: [9]

- Nikdy nekončící zjišťování momentálních zákaznických potřeb a motivace k nákupu.
- Neustálé zlepšování komunikace se zákazníkem.
- Vylepšování automatizovaných procesů, které zefektivňují prodej a komunikaci.
- Činnosti v oblasti marketingu, zákaznického servisu a prodeje, by měli být udržovány v rovnováze, tedy žádná by neměla být nadřazená nebo preferovaná a zároveň by tyto aktivity měly tvořit jeden funkční celek.

3 Analýza a specifikace požadavků CRM aplikace

V analýze jsou představeny služby Mailchimp a Mandrill, které CRM aplikace využívá. Také jsou zde popsány podobné existující řešení, které tyto služby rovněž integrují, nebo pracují na stejném principu. Dále v této kapitole můžeme najít veškeré funkční a nefunkční požadavky na CRM aplikaci. Součástí analýzy je také popis zvolené technologie a uživatelské scénáře.

3.1 Mailchimp

Službu Mailchimp vlastní stejnojmenná společnost, která byla založena v roce 2001 a své hlavní sídlo má v USA ve městě Atlanta. V současné době patří služba Mailchimp mezi přední světové platformy na e-mailový marketing. To dokazují také čísla, která udávají, že se přes Mailchimp každý den odešle více jak miliarda e-mailů. Mailchimp je celosvětově populární služba a Česká republika není výjimkou, jelikož i u nás se těší velké oblíbenosti i přes absenci české lokalizace. [12]

Mailchimp je online marketingová služba, určená k hromadnému rozesílání e-mailových kampaní. Kromě hromadného rozesílání e-mailů je v Mailchimu možné vytvářet automatizované kampaně. Mailchimp není jenom o rozesílání e-mailů, dokáže toho daleko více. I přesto, že Mailchimp o sobě tvrdí, že je určen pro malé podniky, je v něm možné vytvořit velmi propracované a profesionální e-mailové kampaně, a to i bez znalosti programování nebo e-mailového marketingu. Jak již bylo zmíněno, Mailchimp je bez české lokalizace, proto je nutností základní znalost angličtiny. [12]

Mailchimp disponuje propracovaným editorem pro e-mailové šablony, jehož ovládání je velice příjemné a šablonu v něm zvládne vytvořit téměř každý. K propisování hodnot, používání podmínek nebo používání funkcí v e-mailových šablonách používá Mailchimp vlastní značkovací jazyk, který má své základy v HTML. Šablony je možné nechat otestovat v různých e-mailových klientech z důvodu kontroly správnosti zobrazení. [12]

Nastavení e-mailových kampaní je také velice jednoduché a intuitivní, ať už se jedná o klasickou kampaň, nebo automatizovanou. Pokud chce uživatel vytvořit složitější kampaň, může být její nastavení celkem oříšek i pro zkušeného uživatele. I přes toto omezení lze v Mailchimu s trochou trpělivosti vytvořit velice komplexní kampaně.

Dále zde jsou k dispozici, stejně jako u šablon, některé předpřipravené kampaně, ze kterých si může uživatel vzít inspiraci. Při vytváření a optimalizaci e-mailových kampaní může uživatel využít mnohých statistik, kterými také Mailchimp disponuje. [12]

Výhody Mailchimu:

- Zdarma do 2000 odběratelů a 10 000 e-mailů měsíčně
- Vysoká doručitelnost e-mailů
- Přehledné prostředí a jednoduché ovládání
- Pokročilé statistiky a reporty z e-mailových kampaní
- Předpřipravené šablony a kampaně
- Oficiální API a její dokumentace
- Propracovaná správa šablon a obrázků

Nevýhody Mailchimu:

- Nedisponuje českou lokalizací
- Verze zdarma neobsahuje víceřadkové automatizované kampaně
- Vyšší cena u placené verze

Cena za službu Mailchimp se zakládá na počtu odběratelů a měsíčně odeslaných e-mailů. Zákazník má na výběr ze 3 placených balíčků, kde nejnižší cena začíná na 10\$. Za tento poplatek je možné rozesílat 500 tisíc e-mailů měsíčně až 50 tisícům odběratelů. [13]

V tomto projektu je Mailchimp napojen na CRM aplikaci a má uživateli práci s ním ještě více zpříjemnit. Prostřednictvím CRM aplikace se do Mailchimu distribuují veškerá důležitá data z internetového obchodu, které pak uživatel může pohodlně využít při tvorbě kampaní. Dalším cílem je usnadnit vytváření kampaní, které je možné provádět přímo z CRM aplikace. U kampaní v CRM aplikaci jsou k dispozici také předpřipravené texty v češtině. Dále je pro uživatele CRM aplikace k dispozici průvodce pro práci s Mailchimpem.

3.2 Mandrill

V roce 2012 spustila společnost Mailchimp doplňkovou službu Mandrill. Reagovala tak na volání zákazníků po možnosti odesílat individuální transakční e-maily dle vlastních potřeb. Služba Mandrill si získala během 3 let více jak 800 tisíc uživatelů a v roce 2015 se přes ní odeslalo více jak 88 miliard transakčních e-mailů. [14]

Mandrill je e-mailová služba, kterou je možné použít pro odesílání personalizovaných e-commerce e-mailů nebo automatizovaných transakčních e-mailů. Služba Mandrill funguje pouze jako doplněk Mailchimu a není možné ji používat samostatně. V Mandrillu je k dispozici editor e-mailových šablon, testování šablon v různých e-mailových klientech, statistiky a další funkcionality. Šablony pro e-maily je možné vytvářet rovněž v Mailchimu a poté je odeslat do Mandrillu. E-maily přes Mandrill je možné posílat prostřednictvím Mandrill API. [15]

Výhody Mandrillu:

- Vysoká garance doručení e-mailu
- Trackování e-mailů
- Podrobné statistiky k e-mailům
- Zabrání opakovaně odesílat e-maily na hardbounce adresy
- Propracované testování šablon

Nevýhody Mandrillu:

- Cena

Cena za používání Mandrillu se odvíjí od počtu odeslaných e-mailů za měsíc. Nejnižší možná cena je 20\$, za kterou je možné odeslat až 25 tisíc e-mailů za měsíc. [16]

V tomto projektu se služba Mandrill využívá k odesílání transakčních e-mailů z e-shopu. Při zadávání té práce, byla služba zdarma do určitého počtu odeslaných e-mailů. Nyní je služba Mandrill dostupná pouze v placené verzi, proto bylo do projektu přidání ještě odesílání e-mailů přes vlastní SMTP server. Pro práci se službou Mandrill je v CRM aplikaci přiložen průvodce.

3.3 Podobné typy aplikací

V této kapitole jsou analyzovány aplikace, které zprostředkovávají podobné služby, jako CRM aplikace popisovaná v této práci.

3.3.1 Shoptet

Služba Shoptet má jiný koncept než klasická CRM aplikace, protože zprostředkovává kompletní provoz internetového obchodu. Shoptet ale obsahuje také prvky pro řízení vztahů se zákazníky spojené s internetovým obchodem, proto je zařazen do analýzy. Služba obsahuje vše od vytvoření e-shopu, přes vyřizování objednávek, přijímání plateb od zákazníků, správu produktů až po řízení vztahu se zákazníkem. Jedná se o velice komplexní službu, do které je možné přidávat doplňky od externích dodavatelů a integrovat různé služby, čímž ji ještě rozšířit. [39]



Obrázek 5: Logo Shoptet (Zdroj: [40])

Výhody a nevýhody, které jsou zmíněné níže, jsou hodnoceny pouze z hlediska integrování služeb Mailchimp, Mandrill a poskytování prostředků pro řízení vztahů se zákazníky.

Výhody Shoptetu:

- Kompletní řešení pro e-shop
- Zprostředkovává transakční e-maily
- Umožňuje napojení na službu Mailchimp
- Synchronizace objednávek a kontaktů do služby Mailchimp
- Statistiky internetového obchodu
- Testovací plná verze zdarma na 30dní

Nevýhody Shoptetu:

- Kompletní řešení pro e-shop
- Napojení na Mailchimp pouze ve vyšší placené verzi
- Není možné napojit na službu Mandrill
- Omezené nastavení e-mailových šablon
- Neumožňuje napojení na již existující řešení e-shopu

Kompletní řešení pro e-shop je zařazeno, jak do výhod, tak do nevýhod, protože jedné skupině zákazníků koncept služby vyhovuje, ale druhé nikoliv. CRM aplikace v této práci cílí na druhou skupinu uživatelů, kteří chtějí internetový obchod provozovat po svém a spojit své řešení se CRM aplikací pro řízení vztahů se zákazníky, která je určena přímo pro e-shop.

3.3.2 SmartEmailing

SmartEmailing je webová aplikace pro online marketing a zprostředkovává rozesílání SMS zpráv a e-mailů. Jedná se o českou obdobu služby Mailchimp, na kterou je možné napojit internetový obchod. SmartEmailing obsahuje nástroje pro tvorbu e-mailových kampaní a responzivních šablon. Šablony umožňuje testovat v různých e-mailových klientech a dále podporuje automatizované e-mailové kampaně i přímo pro internetový obchod, jako například kampaň na opuštěný nákupní košík. [41]

Kromě samotné aplikace poskytuje SmartEmailing služby spojené s e-mailovým marketingem. Tyto služby jsou však placené. Je jimi například tvorba kampaní na míru, tvorba a úprava šablon, vypracování kompletní e-mailové strategie, či různá školení a workshopy. [42]



Obrázek 6: Logo SmartEmailing (Zdroj: [43])

SmartEmailing disponuje API, přes kterou je SmartEmailing možné napojit na vlastní systém. Prostřednictvím API je možné rozesílat e-maily, odesílat kontakty do aplikace, odesílat objednávky z obchodu nebo posílat SMS zprávy.

Výhody SmartEmailingu:

- E-mailové kampaně a transakční e-maily v jedné aplikaci
- Rozesílání e-mailů a SMS
- Podpora tvorby a testování šablon
- Připravené e-mailové kampaně
- Automatizované kampaně
- Statistiky kampaní
- API pro napojení

Nevýhody SmartEmailingu:

- Cena
- Nedisponuje UAC pro zákazníky obchodu
- Nízká podpora pro internetové obchody
- Neumožňuje posílat přes API produkty, aby mohli být použity v šablonách

SmartEmailing obsahuje množství funkcionalit, které má CRM aplikace také. CRM aplikace se ovšem zaměřuje pouze na internetové obchody a poskytuje uživateli více statistik plynoucích z provozování obchodu.

SmartEmailing není určený přímo pro internetové obchody, ale je spíše pro e-mailový marketing. CRM aplikace se zaměřuje přímo na internetové obchody.

3.4 Přínos CRM aplikace oproti existujícím řešením

Z analýzy existujících řešení vyplývá, že existuje velké množství různých řešení, ať už se jedná o aplikace na řízení vztahů se zákazníky, nebo aplikace na rozesílání e-mailů. Velká část existujících CRM řešení není orientovaná čistě na internetové obchody, ale zaměřují se i na jiné oblasti podnikání. Z toho důvodu jsou některé činnosti internetového obchodu v CRM řešeních úplně vypuštěny a uživatel je připraven o statistiky plynoucí z provozování obchodu. Další problém je, že když už je některé existující CRM řešení orientované na internetový obchod, neposkytuje rozesílání transakčních e-mailů z obchodu nebo řízení uživatelských účtů. Dále často chybí statistiky určené produktům, které internetový obchod prodává. S podporou věrnostního programu pro internetový obchod se u jiných aplikací téměř nesetkáme.

Přínosy CRM aplikace:

- Integrace služby Mailchimp pro tvorbu a rozesílání e-mailových kampaní, které povedou ke zvýšení obrátu obchodu
- Provedení základního nastavení Mailchimu pro internetový obchod prostřednictvím CRM aplikace
- Distribuce dat zákazníků, objednávek, nákupních košíků a produktů do Mailchimu z internetového obchodu prostřednictvím CRM aplikace

- Integrace služby Mandrill pro rozesílání transakčních e-mailů z internetového obchodu s vysokou doručitelností
- Možnost rozesílat transakční e-maily také přes vlastní SMTP server
- Připravené události na rozesílání transakčních e-mailů společně se vzorovými texty e-mailových šablon
- Věrnostní program pro zákazníky internetového obchodu
- Univerzální rozhraní API pro napojení obchodu
- Poskytování UAC pro zákazníky internetové obchodu
- Statistické moduly pro zákazníky, objednávky, produkty, věrnostní program a transakční e-maily
- Průvodce pro práci se službami Mailchimp a Mandrill
- Webová dokumentace, jako průvodce napojením CRM aplikace na internetový obchod
- Průvodce pro práci s CRM aplikací

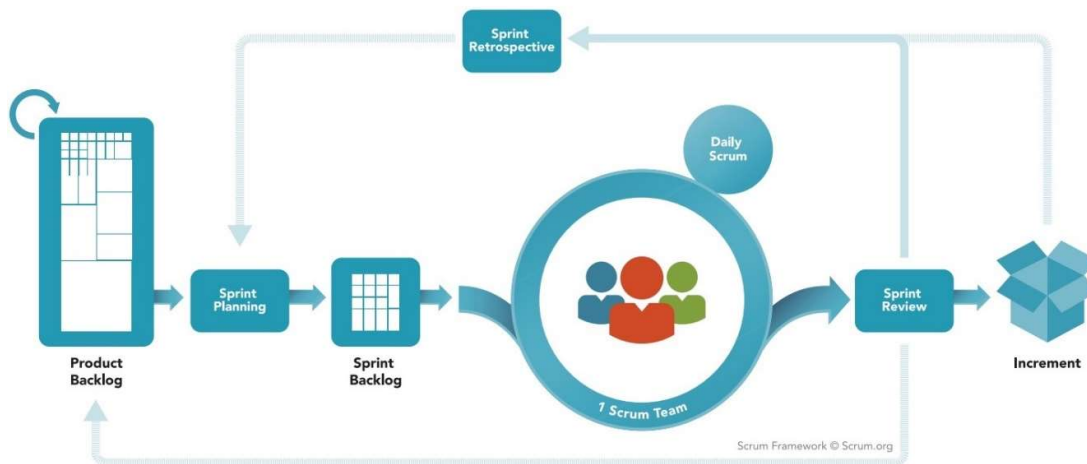
Mezi hlavní přínosy CRM aplikace patří kompletní správa e-mailové komunikace se zákazníky, zajištění řízení uživatelských účtů (UAC) pro zákazníky obchodu, zprostředkování věrnostního programu a vyhodnocování statistik vyplývajících z provozování internetového obchodu.

3.5 Metodika vývoje aplikace

Při vývoji aplikací a vůbec SW je zapotřebí mít práci zorganizovanou a naplánovanou, zvláště pokud se aplikace vyvíjí ve větším počtu lidí, kteří musí mezi sebou spolupracovat. Pokud práce není správně zorganizovaná, může se například v důsledku toho stát, že se vývoj SW zpozdí a tím se většinou zvýší i celkové náklady nebo se v horším případě vývoj úplně zasekne. I když na vývoji aplikace pracuji sám, práci jsem se snažil rozvrhnout a naplánovat. Při plánování jsem vycházel z metodiky agilního vývoje softwaru Scrum, kterou jsem si upravil podle svých potřeb s ohledem na to, že jsem jediný člen vývojářského týmu. Metodika Scrum je velmi flexibilní a dokáže rychle reagovat na změny, proto byla zvolena pro tento projekt.

3.5.1 Agilní metodika Scrum

Za autory agilní metodiky Scrum, která je určená pro projektové řízení, jsou považováni Ken Schwaber a Jeff Sutherland, kteří Scrum v roce 1995 společně prezentovali na konferenci OOPSLA. Prezentace se zabírala shrnutím všech nabytých zkušeností, které oba autoři nasbírali během předchozích let při používání této metodiky. Na následující obrázku je zjednodušeně znázorněno, jak Scrum Framework vypadá a jaké jednotlivé části obsahuje. [17]



Obrázek 7: Scrum Framework (Zdroj: [18])

V Scrum týmu jsou obsaženy 4 hlavní role. Product Owner, vlastník produktu, zodpovídá za Product Backlog a plánování sprintů. Druhou rolí je Scrum Master, který se stará o dodržování a osvojení pravidel Scrumu. Třetí rolí jsou Stakeholders. Touto rolí disponují většinou testéři a lidé od zákazníka, kteří vznášejí připomínky k vyvíjenému produktu. Nad připomínkami se pak vedou diskuse a v případě potřeby je Product Owner zanesse do Product Backlogu jako funkcionalitu. Poslední rolí je vývojářský tým.

Product Backlog, který je zmíněn v předchozím odstavci, obsahuje seznam všech funkcionalit a vlastností, které má výsledný produkt mít s přiřazenou prioritou. Product Backlog není nikdy úplný a průběhu vývoje se do něj přidávají nové funkcionality a změny. Z Product Backlogu se následně vytváří sprint backlog, který obsahuje úkoly pro aktuální sprint.

Sprint je časový úsek Scrumu, který má dopředu stanovenou délku svého trvání. Každý sprint začíná naplánováním prací a naplněním Sprint Backlogu úkoly, které jsou

zapotřebí v daném sprintu splnit. Na každého konci sprintu je provedeno vyhodnocení, ve kterém je zhodnoceno, jaké úkoly byly splněny, co se nesplnilo, popř. jestli během sprintu docházelo k nějakým problémům. Na základě vyhodnocení se následně naplánuje další sprint. Během sprintu probíhají každý den krátké schůzky členů vývojového týmu, tzv. Daily Scrum. Na těchto schůzkách každý člen týmu referuje, jaký učinil pokrok od poslední schůzky a co má naplánováno do další schůzky. [19]

3.5.2 Použití metodiky Scrum v tomto projektu

Jak již bylo zmíněno, metodiku Scrum jsem si upravil podle vlastních potřeb při práci na projektu. Osobně zastávám role vývojového týmu, Product Ownera a Scrum Mastera. Vedoucí práce zastává roli Stakeholdera.

Před začátkem vývoje byl vytvořen Product Backlog, který obsahoval veškeré funkcionality a vlastnosti, které má CRM aplikace poskytovat. Jako zdroj informací, na základě kterých se Product Backlog sestavil, posloužila analýza funkčních a nefunkčních požadavků, specifikace SW požadavků a návrh architektury systému.

Jednotlivé sprinty a jejich časová osa se vždy plánovaly dle aktuální potřeby a časových možností. Ze začátku projektu se plánovaly sprinty s časovou osou klidně i jeden měsíc. Ke konci projektu se naopak sprinty plánovaly kratší a intenzivnější. Na konci každého sprintu se provedlo zrevidování Sprint Backlogu a Product Backlogu. Může se však stát, že se nepodaří dokončit všechny úkoly, které byly pro daný sprint naplánované. To při vývoji agilní metodikou není žádný problém, protože nedokončené úkoly se jednoduše přesunou do dalšího sprintu.

Abych měl o úkolech přehled a věděl, které jsem si naplánoval v dalším sprintu, použil jsem software Jira. Tento software vyvíjí společnost Atlassian. Nástroj je určený k řízení projektů, analýze rizik, evidenci a celkovou správu projektů. Uživatelská přívětivost a přehlednost byla jedním z důvodů pro použití softwaru Jira v této práci. Dalším důvodem použití je, že společnost Atlassian vyvíjí také aplikaci BitBucket, kterou rovněž používám při vývoji SW. Tyto dva nástroje jsou vzájemně propojeny a lze k nim přistupovat pod jedním uživatelským účtem. Aplikace BitBucket se používá na verzování vytvořeného kódu, protože podporuje vývoj softwaru s využitím verzovacího nástroje Git.

3.6 Funkční požadavky

V následujícím textu jsou popsány základní funkční požadavky na CRM systém. Systém je složen z API, které je určené pro napojení internetového obchodu na CRM aplikaci. Funkční požadavky vychází z analýzy CRM a procesů, které probíhají v internetovém obchodě.

3.6.1 API

Funkční požadavky CRM API, které slouží k napojení internetového obchodu na CRM aplikaci, jsou popsány níže. API umožňuje napojení více internetových obchodů.

3.6.1.1 Řízení uživatelských účtů (UAC) pro zákazníky internetového obchodu

CRM aplikace zajišťuje veškeré potřebné operace spojené s řízením uživatelských účtů pro zákazníky internetového obchodu. UAC probíhá prostřednictvím endpointů v CRM API, na které internetový obchod odesílá requesty.

Mezi operace UAC, které API zprostředkovává, patří:

- Registrace zákaznického účtu
- Přihlášení k účtu
- Odhlášení od účtu
- Změna hesla
- Zapomenuté heslo
- Přístup k informacím v zákaznickém profilu
- Úprava informací v zákaznickém profilu
- Přidání odběru newsletterů
- Odebrání odběru newsletterů
- Odstranění účtu

U operací registrace zákaznického účtu, zapomenuté heslo a odstranění účtu se zákazníkovi odesílá informační transakční e-mail.

3.6.1.2 Odběratelé newsletterů bez registrace

Tato funkcionality je určená pro zákazníky, kteří se v internetovém obchodě nezaregistrovali, ale i přesto chtějí odebírat novinky neboli newslettery z obchodu. Přes API je možné přidávat a odebírat zákazníkům odběr novinek pouze na základě jejich

e-mailové adresy. Tyto adresy se odesílají do Mailchimu, kde se ukládají do listu odběratelů. Prostřednictvím Mailchimu probíhá rozesílání newsletterů.

3.6.1.3 Načtení produktů z obchodu

Načítání produktů do CRM aplikace probíhá přes XML feed, který musí být vytvořen podle specifikace produktového feedu Google Merchant Center. Do CRM aplikace je možné zadat buď URL adresu k XML feedu, nebo nahrát soubor s XML feedem. Načtení ze souboru probíhá jednorázově, ale pokud je zadána URL adresa XML feedu, tak synchronizace produktů probíhá každý den. Produkty se rovněž distribuují do Mailchimu, kde mohou být využity pro potřeby e-mailových kampaní.

3.6.1.4 Opuštěný nákupní košík

U registrovaných zákazníků se do API odesílají data, která obsahují obsah nákupního košíku zákazníka. Na základě těchto dat se v případě nedokončené objednávky se prostřednictvím Mailchimu odešle zákazníkovi e-mail s obsahem jeho opuštěného nákupního košíku. Prostřednictvím odkazu v e-mailu pak může zákazník přejít na dokončení objednávky.

3.6.1.5 Evidence objednávek

Do API se z internetového obchodu mohou odesílat všechny provedené objednávky. Data z objednávek následně slouží pro výpočet statistik a cílenou propagaci produktů. U objednávky je následně možné měnit přes API její stavy. Na základě změn těchto stavů se zákazníkovi odesílají transakční e-maily, které ho informují o stavu objednávky. Po vytvoření objednávky se přes API odešle transakční e-mail. Tento e-mail obsahuje souhrn všech informací evidovaných u objednávky.

U objednávek se evidují zákaznické údaje, stavy, fakturační údaje, doručovací údaje, zakoupené produkty a ceny. Dále se u objednávek eviduje, zda zákazník při dokončení souhlasil, nebo nesouhlasil s odběrem newsletterů. Pokud ano, tak je zákazník zařazen do seznamu odběratelů newsletterů.

Pokud některý z produktů v objednávce není zaevidován v CRM aplikaci, automaticky se do ní přidá. Objednávky se dále posílají do Mailchimu, kde se využijí k optimalizaci e-mailových kampaní.

Při vytvoření objednávky registrovaným zákazníkem je možné rovnou uplatnit body z věrnostního programu. Věrnostní program je více popsán níže.

3.6.1.6 Věrnostní program

Věrnostní body zákazník získává za dokončené a zaplacené objednávky. Počet a peněžní hodnota věrnostních bodů, které zákazník za objednávku dostane, se určí podle nastavení věrnostního programu v CRM aplikaci. Funkční požadavky na nastavení věrnostního programu jsou popsány v následující kapitole. K tomu, aby mohl být zákazník zařazen do věrnostního programu, je nutné, aby se zaregistroval. Pro správné fungování věrnostního programu je zapotřebí, aby napojený obchod přes API pravidelně aktualizoval stavy objednávek.

Věrnostní body je možné zákazníkovi také připsat na věrnostní konto přes endpoint v API, na základě jeho uživatelského tokenu nebo e-mailu.

Věrnostní body může zákazník uplatnit při dokončení objednávky, kdy obchod přes API zjistí stav bodů věrnostního konta zákazníka společně s jejich peněžní hodnotou. Obchod poté o hodnotu věrnostních bodů sníží zákazníkovi celkovou cenu objednávky.

O přičtení nebo uplatnění věrnostních bodů je zákazník informován prostřednictvím transakčních e-mailů. V e-mailu se zobrazí informace o dané transakci, která byla provedena na věrnostním kontě zákazníka společně s aktuálním stavem věrnostního konta.

3.6.2 Ovládání API nastavení CRM aplikace

Níže jsou uvedeny funkční požadavky na ovládání API, které bude k dispozici v CRM aplikaci.

3.6.2.1 Nastavení obchodu

Pro napojení internetového obchodu na CRM aplikaci, je nutné v aplikaci daný obchod založit vyplnit všechny potřebné údaje. Dále je pro obchod potřeba vytvořit přístupový bod do CRM API, jejíž správa je popsána níže. Pro každý napojený obchod, je možné vytvořit více přístupových bodů, pro získávání dat z více zdrojů. Mezi napojenými obchody je možné v CRM aplikaci jednoduše přepínat a ovládat každý samostatně.

Pro každý internetový obchod, který bude na CRM aplikaci napojen, se v aplikaci založí nový obchod a přidělí se mu přístup do API. Ke každému obchodu se z Mailchimpu přiřadí list odběratelů, do kterého se budou odesílat zákaznické údaje. Mezi napojenými obchody je možné v CRM aplikaci jednoduše přepínat a ovládat každý samostatně.

3.6.2.2 Nastavení přístupu k API

Veškerá správa a nastavení přístupu k API se provádí v CRM aplikaci v nastavení obchodu. V nastavení je možné přidávat nové přístupy do API. Při přidání nového přístupu do API se automaticky vygeneruje unikátní přístupový klíč. Vytvořené přístupy je možné deaktivovat. Deaktivováním určitého přístupu do API se zablokuje přidělený přístupový klíč do CRM API. Deaktivovaný přístup do API je možné opět aktivovat nebo úplně odstranit. Pro přístup k API je možné zvolit, jaké endpointy budou s daným přístupovým klíčem dostupné a jaké mu budou zablokovány.

3.6.2.3 Přehled requestů na API

V CRM aplikaci je k dispozici přehled všech requestů, neboli požadavků na API. Všechny požadavky, které API zpracovává, se zaznamenávají společně s odpovědí, kterou API na požadavek odeslalo. Přehled zahrnuje celkové statistiky provozu API a statistiky pro jednotlivé přístupy vytvořené v nastavení API.

Requesty je v přehledu možné filtrovat například podle časové osy, endpointu nebo statusu odpovědi. V detailu requestu je zobrazen kompletní požadavek se všemi parametry společně s odeslanou odpovědí.

3.6.2.4 Nastavení napojení na Mailchimp API

Pro integraci služby Mailchimp do CRM aplikace je nutné zadat validní přístupový klíč k Mailchimp API. S validním klíčem, je uživateli umožněn přístup do nastavení obchodu používaného v Mailchimpu a umožní mu přiřadit k obchodu list odběratelů, do kterého se budou odesílat zákaznické údaje. Veškerá komunikace mezi CRM aplikací a Mailchimp API se loguje. Uživatel se na komunikaci může podívat v přehledu logů.

3.6.2.5 Nastavení napojení na Mandrill API

Pro integraci služby Mandrill do CRM aplikace je nutné zadat validní přístupový klíč k Mandrill API. Nastavení je doplněno o kontrolu dostupnosti služby Mandrill

pro zadaný přístupový klíč. Veškerá komunikace mezi CRM aplikací a Mandrill API se loguje. Uživatel se na komunikaci může podívat v přehledu logů.

3.6.2.6 Správa uživatelských účtů v CRM aplikaci

V aplikaci figurují celkem tři uživatelské role. První role je admin, druhá je developer a třetí je user. Uživatelské role a jejich oprávnění jsou více popsány níže. Ke správě uživatelských účtů má oprávnění pouze uživatel s rolí admin. Ten může uživatele v CRM aplikaci vytvářet, odstraňovat, upravovat jejich uživatelské údaje a nastavovat jim oprávnění. Dále má možnost zadat uživateli nové heslo v případě, že by ho zapomněl.

3.6.3 CRM aplikace

Přes CRM API je možné napojit více obchodů najednou, proto je zapotřebí umožnit uživateli v CRM aplikaci snadné přepínání mezi jednotlivými obchody. Níže jsou popsány funkční požadavky CRM aplikace.

3.6.3.1 Ovládání služby Mailchimp

V CRM aplikaci je možné vytvářet a nastavovat e-mailové kampaně pro odběratele newsletterů. CRM aplikace obsahuje předpřipravené kampaně pro Mailchimp. U každé kampaně je připravena také vzorová šablona. Pokud chce uživatel kampaň spustit, musí ji aktivovat a provést základní nastavení. Poté je kampaň odeslána do Mailchimu, kde ji uživatel může dále přizpůsobovat svým potřebám a rozeslat. E-mailové kampaně je možné odesílat pouze zákazníkům, kteří souhlasili s odběrem novinek z internetového obchodu.

Seznam některých předpřipravených e-mailových kampaní:

- Zapomenutý nákupní košík (nedokončená objednávka)
- Přihlášení k odběru novinek (uvítací e-mail)
- Narozeninový dárek v podobě věrnostních bodů
- Novinky z blogu podle RSS feedu
- Novinky z internetového obchodu
- Dárek pro ženy k MDŽ v podobě věrnostních bodů

3.6.3.2 Nastavení transakčních e-mailů

U obchodu je možné nastavit, jaké transakční e-maily se budou odesílat. Dále se nastaví, zda se tyto e-maily budou rozesílat prostřednictvím služby Mandrill, nebo vlastního SMTP serveru. Odesílání transakčních e-mailů je možné úplně zakázat.

U každého transakčního e-mailu se nastaví HTML šablona. Tato šablona může být vytřena prostřednictvím CRM aplikace, nahraná ze souboru, nebo přiřazená ze služby Mandrill.

Seznam transakčních e-mailů:

- Dokončení registrace zákazníka
- Zapomenuté heslo
- Dokončení objednávky
- Předání objednávky dopravci
- Potvrzení zaplacení objednávky
- Výzva k uhrazení objednávky
- Připsání věrnostních bodů do věrnostního konta
- Expirace věrnostních bodů na věrnostním kontě
- Potvrzení odstranění zákaznického účtu

3.6.3.3 Věrnostní program

Základ věrnostního programu je založen na principu získávání věrnostních bodů za dokončené a zaplacené objednávky. Získané body je možné použít na snížení celkové ceny objednávky. Věrnostní program je možné nastavit pro každý napojený obchod zvlášť.

Nastavení věrnostního programu může například vypadat takto:

- Za každých utracených **200 Kč** s DPH zákazník dostane 1 věrnostní bod.
- Jeden věrnostní bod má hodnotu **10 Kč** s DHP.
- Jeden věrnostní bod má hodnotu **7.9 Kč** bez DHP.
- Výše DPH u věrnostního programu je **21 %**.
- Věrnostní body mají platnost 12 měsíců od připsání.

3.6.3.4 Objednávky

V CRM aplikaci je k dispozici přehled objednávek z internetového obchodu. Objednávky v přehledu je možné filtrovat podle čísla objednávky, stavů, typu platby, zákaznických údajů, zakoupených produktů, uplatněných věrnostních bodů a času vytvoření objednávky.

Z přehledu se může uživatel dostat na detail konkrétní objednávky. V detailu se zobrazují veškeré dostupné informace k objednávce, včetně informací k věrnostnímu programu. Objednávku je možné z CRM aplikace odstranit.

3.6.3.5 Produkty

V CRM aplikaci můžeme nalézt přehled produktů načtených z internetového obchodu přes XML feed. Produkty v přehledu je možné filtrovat podle identifikátoru, kódu, názvu, značky, dostupnosti, či výhody akční nabídky.

Z přehledu se může uživatel dostat na detail konkrétního produktu. V detailu se zobrazují veškeré dostupné informace k danému produktu, včetně celkových statistik a statistik za určité časové období. Statistiky obsahují informace o tom, jak se daný produkt prodává.

3.6.3.6 Zákazníci

V CRM aplikaci je přehled zákazníků zaregistrovaných v internetovém obchodě. Zákazníky je možné v přehledu filtrovat podle jména, příjmení, e-mailu, pohlaví, adresy, odběru newsletterů a dalších možných parametrů.

Z přehledu se může uživatel dostat na detail konkrétního zákazníka. V detailu se zobrazují veškeré dostupné informace k danému zákazníkovi. U zákazníka je možné najít jeho objednávky, stav věrnostního konta, transakce na věrnostním kontě, statistiky z dokončených objednávek a jeho TOP 5 nejnakupovanějších produktů.

3.6.3.7 Statistiky

Mimo již výše uvedené statistiky v detailech zákazníka, nebo produktu, které jsou zmíněné v požadavcích výše, jsou v CRM aplikaci obsaženy další statistiky. Jsou to statistiky k objednávkám, zákazníkům, produktům, věrnostnímu programu, e-mailovým kampaním a transakčním e-mailům. Ve statistikách jsou informace za určité časové

období a celkové počty. Statistiky jsou zobrazeny pomocí grafů a tabulek s výslednými daty. Vyhodnocování statistik je popsáno v kapitole implementace.

V následujícím textu jsou popsány jednotlivé statistické moduly, které CRM aplikace zprostředkovává. Každý modul zprostředkovává statistiky za určité časové období a celkové statistiky.

Statistiky zákazníků

Statistiky zákazníků za určité časové období obsahují informace o provedených registracích, smazaných účtech, přihlášení k účtům, provedených objednávkách registrovanými zákazníky a přidáných, nebo odebraných odběrů newsletterů.

Celkové statistiky obsahují informace o celkových počtech registrovaných zákazníků, jejich rozdělení podle zákaznických údajů, historickém vývoji počtu registrovaných zákazníků, vývoji počtu odběratelů a informace o celkových počtech smazaných účtů a odebraných odběrů newsletterů.

Statistiky objednávek

Statistiky za určité časové období obsahují informace o provedených objednávkách, jejich stavech (zda je dokončená, expedovaná, atd.), věrnostním programem, peněžní hodnotě dokončených objednávek a další statistické informace. Dále je k dispozici segmentovaný přehled zákazníků, kteří dané objednávky provedli.

Celkové statistiky objednávek obsahují celkové vyhodnocení dat nashromážděných z objednávek a celkový segmentovaný přehled zákazníků, kteří v daném obchodě nakupují.

Statistiky produktů

Statistiky za určité časové období obsahují informace o tom, jaké produkty se právě nejvíce prodávají, jaké se nejméně prodávají, kolik kusů se daného produktu již prodalo a další údaje o produktech, které se získají z dokončených objednávek.

Celkové statistiky obsahují informace jako statistiky za časové období, ale jsou vyhodnocovány za celou dobu provozování obchodu.

Statistiky věrnostního programu

Statistiky za určité časové období obsahují informace o tom, kolik bylo na věrnostní konta zákazníků připsáno věrnostních bodů a za co je zákazníci obdrželi. Dále jsou zde počty uplatněných a expirovaných věrnostních bodů. Kromě počtu věrnostních bodů je ve statistikách uvedena i peněžní hodnota bodů.

Celkové statistiky věrnostního programu obsahují informace o tom, kolik je celkem alokováno věrnostních bodů, kolik bodů již zákazníci uplatnili, kolika bodům vypršela platnost. Dále je zde informace o tom, kolik registrovaných zákazníků je zařazeno ve věrnostním programu. Stejně jako u statistik za časové období zde najdeme informaci o peněžní hodnotě věrnostních bodů.

Statistiky e-mailových kampaní

Služba Mailchimp obsahuje podrobné statistiky z e-mailových kampaní, tak není potřeba je duplikovat v CRM aplikaci. Tyto statistiky e-mailovým kampaním má Mailchimp velice dopodrobna propracované a statistiky ke konkrétní e-mailové kampani se uživatel může pohodlně prokliknout přímo z CRM aplikace. Podrobnější statistiky si uživatel může zobrazit přímo v Mandrillu, který je má stejně jako Mailchimp dopodrobna propracované.

Statistiky transakčních e-mailů.

Celkové statistiky transakčních e-mailů obsahují celkové počty rozeslaných e-mailů rozdělených podle druhu transakce. Statistiky za určité období jsou podrobnější a zobrazují počty odchozích e-mailů, jejich úspěšnost doručení. V případě napojení na Mandrill, také kolik z nich bylo otevřeno, míru prokliků a odkaz na podrobnější statistiky o doručitelnosti v Mandrillu.

3.6.4 Dokumentace k CRM aplikaci a AP

Dokumentace k CRM aplikaci a API obsahuje návod na napojení internetového obchodu včetně popisu všech endpointů a parametrů v API. Dále obsahuje základní principy pro práci s CRM aplikací a zlepšení komunikace se zákazníkem. Nakonec obsahuje krátké manuály pro práci s Mailchimpem a Mandrillem.

3.7 Nefunkční požadavky

CRM aplikace funguje jako webová aplikace dostupná online na internetu. API, které spojuje CRM s internetovým obchodem, je vytvořeno v architektuře REST API. Komunikace s API probíhá pomocí HTTP protokolu s povinným SSL. Všechny validní requesty na API musí být provedeny přes POST metodu HTTP protokolu. Všechny požadavky které API zpracovává, se zaznamenávají do logů. Logy obsahují, jak informace o daném požadavku na API, tak odpověď, kterou na něj API odeslalo.

Všechna data, která do API přicházejí, musí projít kontrolou, aby se zamezilo napadení API přes XSS. Všechny funkce, kterými API disponuje a stavy, do kterých se může dostat, nalezne uživatel v dokumentaci, která je součástí CRM aplikace.

V případě výpadku služby Mandrill se transakční e-maily shromažďují do front, ze kterých se odešlou ihned po znovunavázání spojení se službou Mandrill. Podobný princip funguje i při případném výpadku služby Mailchimp, při kterém se do front neukládají e-maily, ale data, která se do Mailchimu distribuují. Data z CRM aplikace se pravidelně synchronizují se službou Mailchimp. Data v Mailchimu by měly vždy odpovídat aktuálním datům v CRM aplikaci.

3.8 Popis zvolené technologie

Webová aplikace CRM je realizována v jazyce PHP 8 s využitím frameworku Nette 3. Aplikace je vytvořena dle softwarové architektury MVC. Pro front-endovou část aplikace jsou použity technologie, jako šablonovací systém Latte, JavaScriptová knihovna jQuery, AJAX, HTML a kaskádové styly, neboli CSS, které se vytváří v preprocesoru LESS.

Webová CRM API je vytvořena v architektuře REST. Požadavky na API se přenáší přes HTTP protokol a odpovědi se odesílají ve formátu JSON. API je také vytvořeno v PHP 8 s využitím Nette 3.

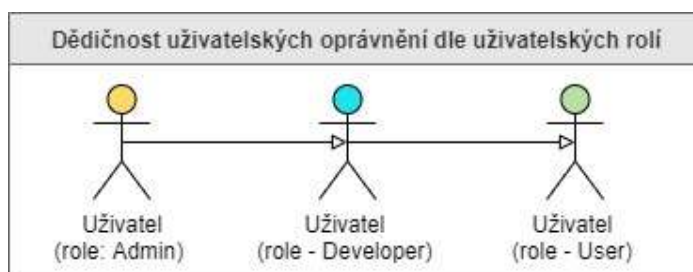
Databáze CRM aplikace a CRM API je tvořena v relačním databázovém systému MySQL.

3.9 Uživatelské role

V CRM aplikaci figurují uživatelské role admin, user a developer. Role admin patří administrátorovi. Administrátor vládne všem uživatelům a má nejvyšší oprávnění. Administrátor jako jediný může přidávat nové uživatele do CRM aplikace a nastavovat jejich uživatelské role. Dále může obnovit zapomenuté heslo uživatele. Role admin zahrnuje veškerá oprávnění rolí developer a user.

Uživatel s rolí user má přístup k webové aplikaci CRM, kde přehledně může procházet data nashromážděná přes API. Dále může nastavovat e-mailové kampaně pro Mailchimp a odchozí transakční e-maily s možným rozšířením pro Mandrill. Tyto funkce jsou více popsány ve funkčních požadavcích. User má přístup k návodům, které by mu měly pomoci při používání CRM aplikace a práci v Mailchimu a Mandrillu.

Třetí uživatelskou rolí je developer. Uživatel s touto rolí má vyšší oprávnění než uživatel s rolí user. Tato role je určena osobě, která bude napojovat internetový obchod na CRM API, proto má rozdíl od role user přístup ještě k nastavení, logům a dokumentaci CRM API.



Obrázek 8: UC diagram – Uživatelské role a dědičnost uživatelských oprávnění (Zdroj: vlastní)

Uživatelé od sebe dědí oprávnění podle rolí. Jak můžeme vidět na obrázku výše, tak uživatel s rolí developer může provádět všechny akce, které může provádět uživatel s rolí user. Dále uživatel s rolí admin může provádět všechny akce, které může provádět uživatel s rolí developer.

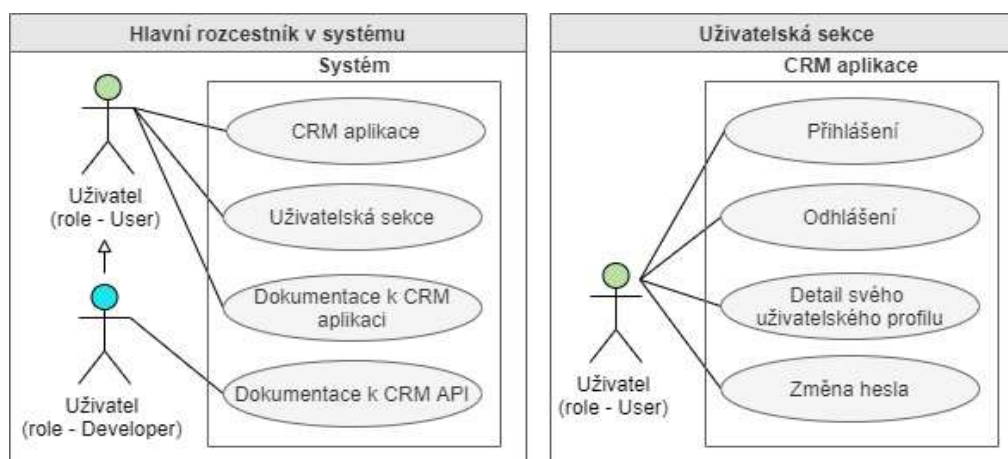
3.10 Uživatelské scénáře

V této kapitole jsou k dispozici uživatelské scénáře v CRM aplikaci popsané pomocí jazyka UML formou UC diagramů. Uživatelské scénáře jsou tvořeny tak, aby splňovaly funkční požadavky uvedené výše. V uživatelských scénářích se objevují aktéři, kterými jsou uživatelé CRM aplikace s určitou uživatelskou rolí.

3.10.1 Správa CRM aplikace

Zde jsou popsány use case diagramy související se správou CRM aplikace.

3.10.1.1 Hlavní části systému a uživatelská sekce

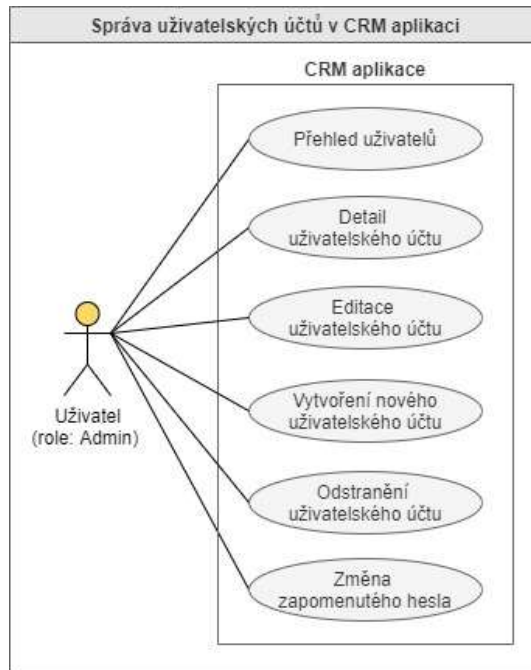


Obrázek 9: UC diagram – Hlavní části systému a uživatelská sekce (Zdroj: vlastní)

Levá část obrázku 9 ukazuje hlavní části, kterými systém CRM aplikace disponuje. Jedná se o samostatnou CRM aplikaci, uživatelskou sekci a dokumentaci k CRM aplikaci a CRM API. Pravá část obrázku 9 popisuje základní akce uživatele v CRM aplikaci spojené s jeho uživatelským účtem.

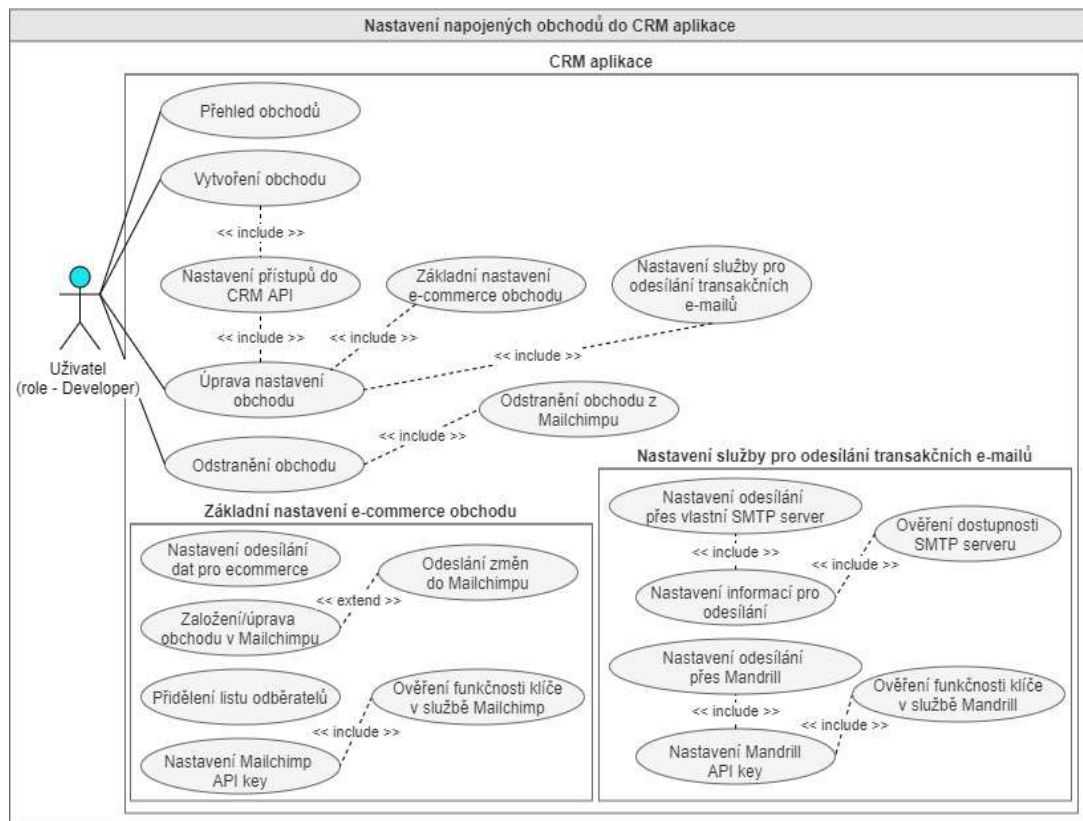
3.10.1.2 Správa uživatelských účtů

Na obrázku 10 jsou popsány akce pro správu uživatelských účtů v CRM aplikaci. Tyto akce může provádět pouze uživatel s rolí admin. Správa uživatelských účtů zahrnuje vytvoření, úpravu, odstranění a detail uživatelského účtu. Součástí této sekce je přehled všech uživatelů, kteří mají přístup do CRM aplikace a možnost změnit jejich heslo v případě, že by ho zapomněli. Uživatel si nemůže sám měnit své údaje v uživatelském profilu, kromě uživatele s rolí admin. Jediné, co může běžný uživatel u svého účtu měnit, je přístupové heslo. O ostatní změny musí požádat uživatele s rolí admin.



Obrázek 10: UC diagram – Správa uživatelských účtů v CRM aplikaci (Zdroj: vlastní)

3.10.1.3 Nastavení obchodů v CRM aplikaci

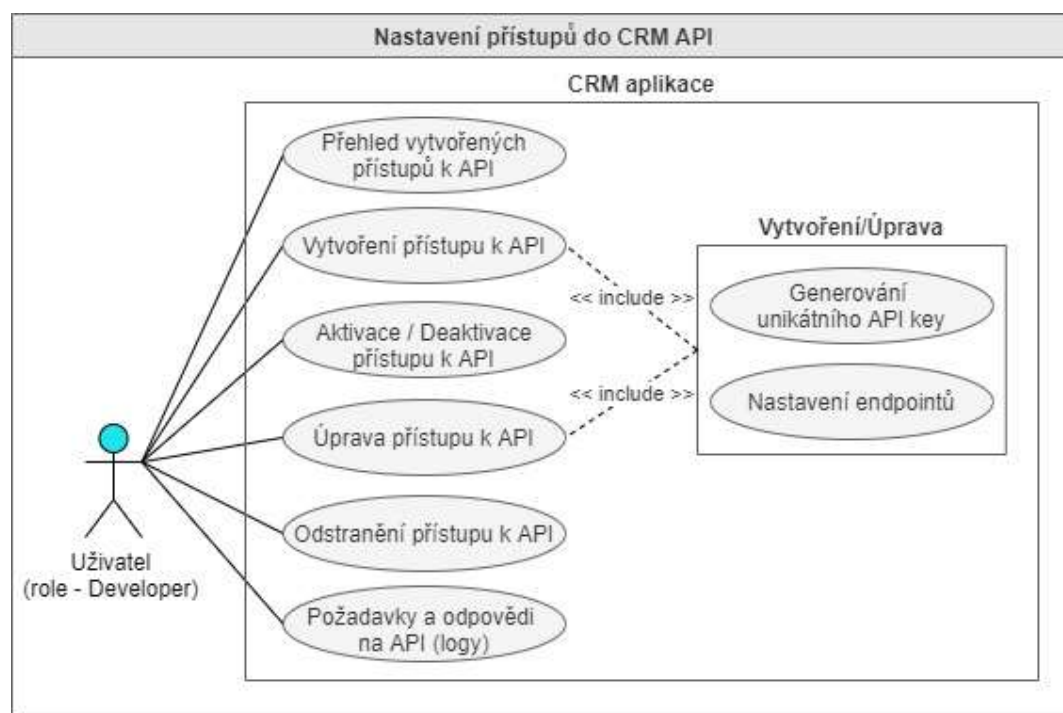


Obrázek 11: UC diagram – Nastavení obchodu v CRM aplikaci (Zdroj: vlastní)

Na diagramu číslo 11 je znázorněna správa napojených obchodů v CRM aplikaci. U obchodu se nastavují základní údaje a přidělují se mu přístupy do API. Nastavení přístupů je popsáno na obrázku 12. U obchodu se dále nastavuje rozšíření pro e-commerce, které v CRM aplikaci reprezentuje Mailchimp. Pokud chceme Mailchimp plně využívat, je nutné nastavit platný klíč k API, integrovat obchod do Mailchimu a přidělit mu list odběratelů. Dále se zde nastavuje služba pro rozesílání transakčních e-mailů. Mezi tyto služby patří vlastní SMTP server, nebo Mandrill. Je možné natavit, aby se transakční e-maily neodesílali. Nastavení obchodu mohou provádět pouze uživatelé s rolí developer nebo admin. Totéž platí o nastavování přístupů do CRM API, které je popsáno níže.

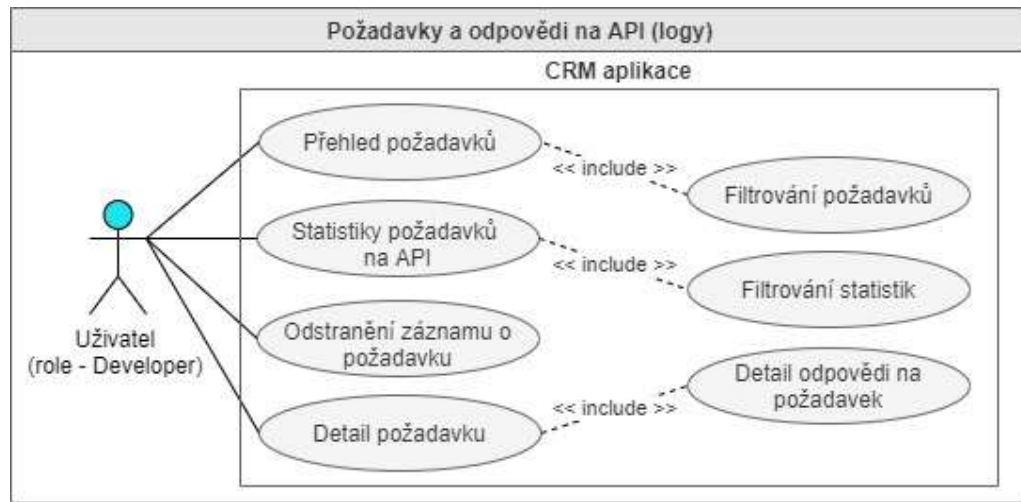
3.10.1.4 Nastavení přístupů do CRM API

Přístupy se vždy nastavují ke konkrétnímu obchodu. Tyto přístupy je možné libovolně vytvářet, upravovat či mazat. Uživatel může přístup do API podle potřeby deaktivovat, aktivovat nebo vygenerovat nový přístupový klíč. U každého přístupu se nastavují endpointy, které bude daný přístup poskytovat. Veškerou komunikaci API si může uživatel prohlédnout v přehledu požadavků a odpovědí na API. Požadavky a odpovědi, jsou více popsány na obrázku 13.



Obrázek 12: UC diagram – Nastavení přístupů do CRM API (Zdroj: vlastní)

3.10.1.5 Požadavky na API



Obrázek 13: UC diagram – Přehled logů z API (Zdroj: vlastní)

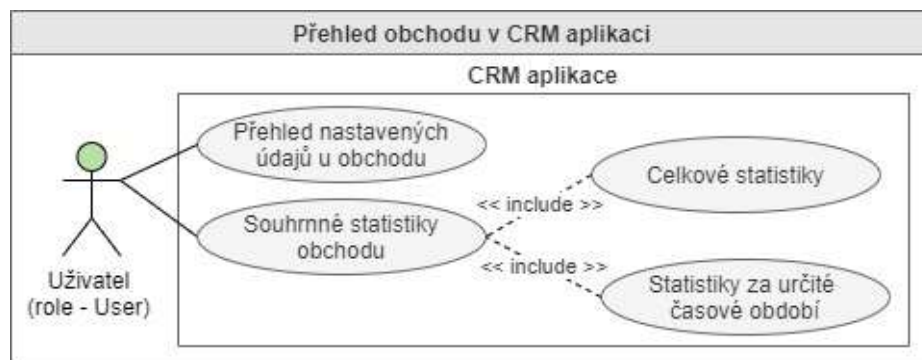
Scénář na obrázku 13 se zaměřuje na prohlížení požadavků na API. Součástí požadavku je vždy i odpověď, kterou API na požadavek odeslalo. V tomto scénáři se také vyskytují statistiky požadavků na API.

3.10.2 CRM aplikace

Zde jsou popsány use case diagramy související s CRM aplikací.

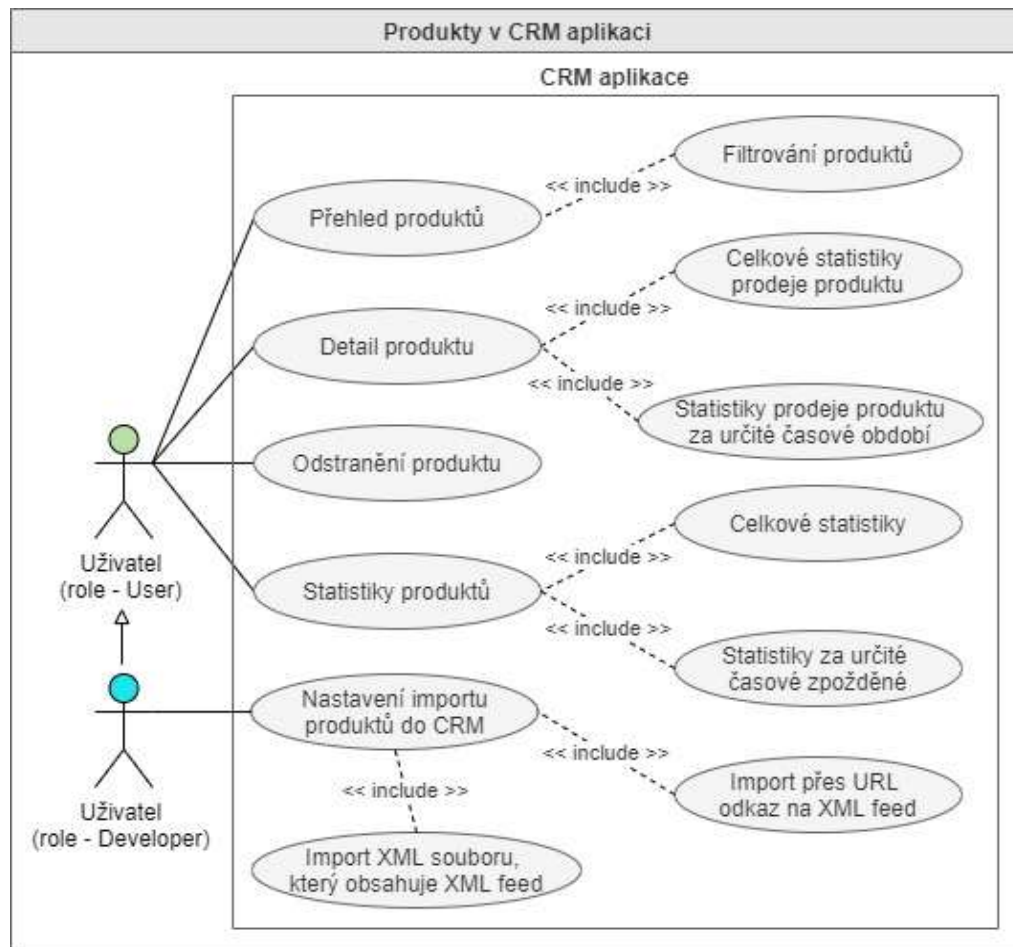
3.10.2.1 Obchod

Na obrázku 14 vidíme, jaké případy užití v souvislosti s obchodem má k dispozici uživatel v roli user. Jedná se o souhrnné statistiky obchodu a přehled nastavení obchodu.



Obrázek 14: UC diagram – Přehled obchodu (Zdroj: vlastní)

3.10.2.2 Produkty

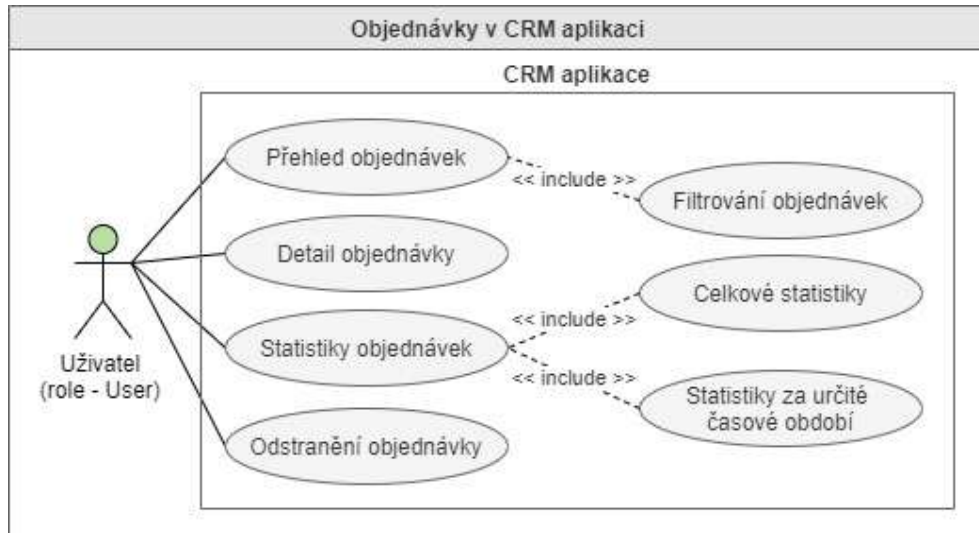


Obrázek 15: UC diagram – Produkty (Zdroj: vlastní)

Na tomto UC diagramu jsou všechny případy užití spojené s produkty. Počínaje přehledem produktů, nastavením importu a statistikami konče. Nastavení importu produktů mohou provádět pouze uživatelé s rolí developer nebo admin.

3.10.2.3 Objednávky

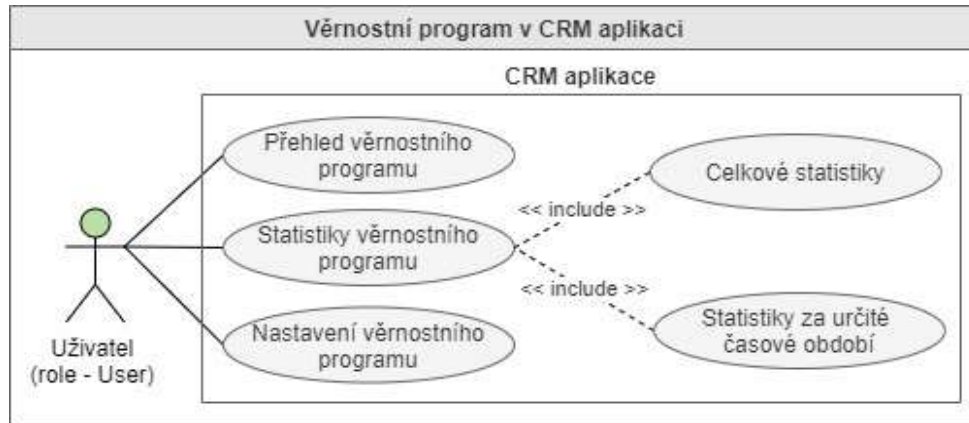
Objednávky, které jsou z obchodu odeslány do CRM aplikace, jsou k dispozici v přehledu s filtrováním. Každá objednávka má svůj detail, ve kterém jsou veškeré informace o objednavce. Objednávku je z CRM aplikace možné odstranit. Z dat objednávek jsou následně tvořeny statistiky.



Obrázek 16: UC diagram – Objednávky (Zdroj: vlastní)

3.10.2.4 Věrnostní program

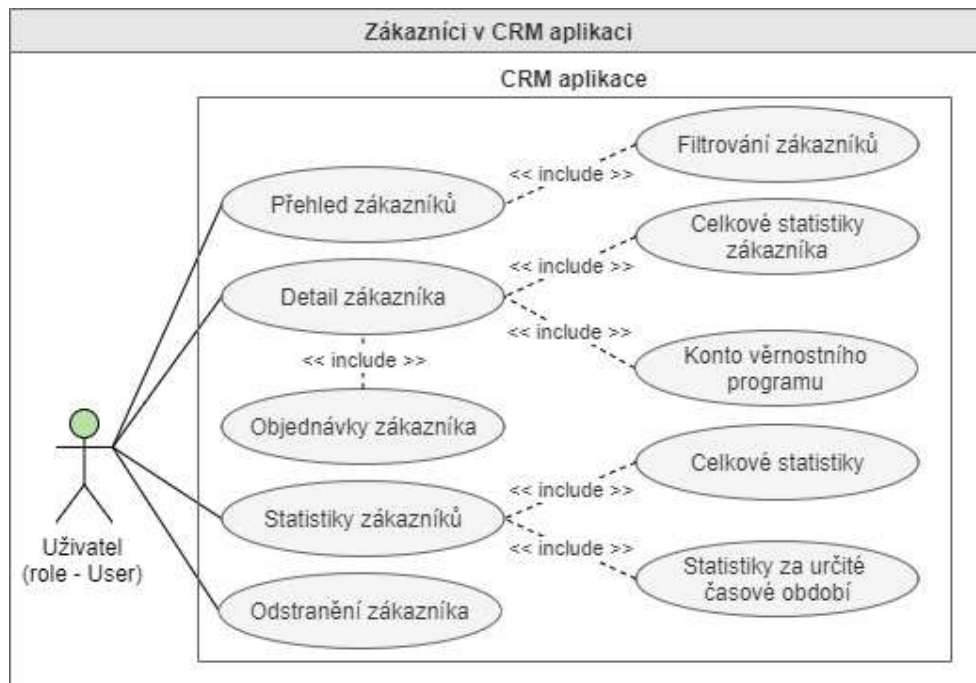
Obrázek 17 ukazuje případy užití spojené s věrnostním programem. Zahrnují přehled, nastavení a celkové statistiky věrnostního programu. Uživatel má k dispozici buď statistiky celkové, nebo za určité časové období.



Obrázek 17: UC diagram – Věrnostní program (Zdroj: vlastní)

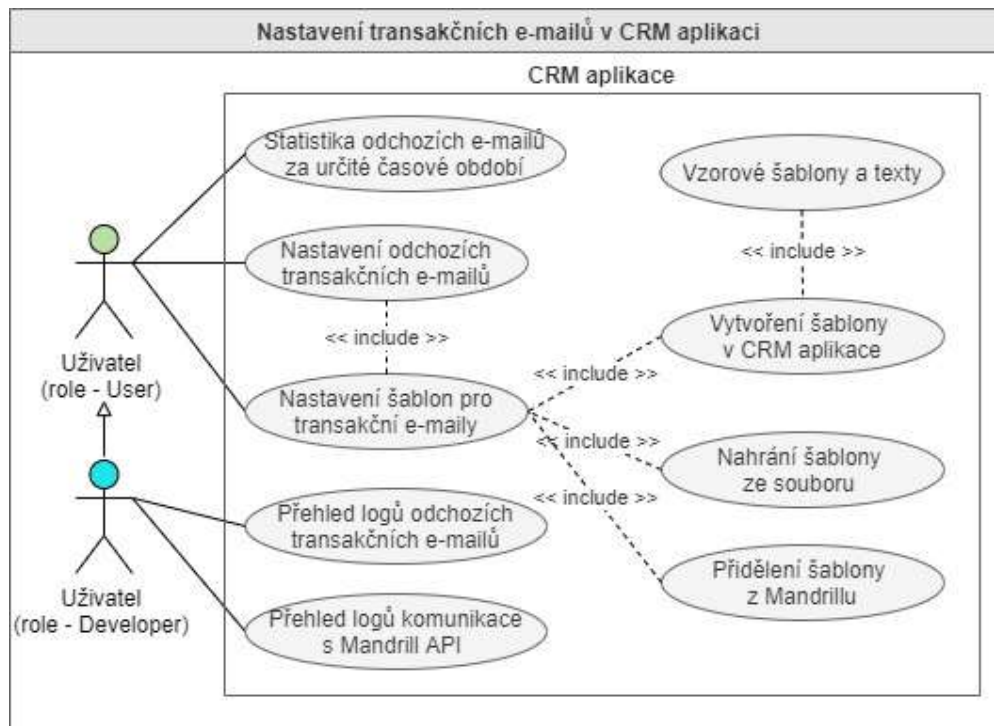
3.10.2.5 Zákazníci

Zákazníky si může uživatel procházet v přehledu, jehož součástí je filtrování. Zákazníka je možné ze CRM aplikace odstranit se všemi jeho daty. Případy užití spojené se zákazníky obsahují statistiky u každého zákazníka, celkové statistiky a statistiky za určité časové období. V detailu zákazníka kromě statistik nalezne uživatel také informace o věrnostním kontě zákazníka a jeho objednávky.



Obrázek 18: UC diagram – Zákazníci (Zdroj: vlastní)

3.10.2.6 Nastavení transakčních e-mailů

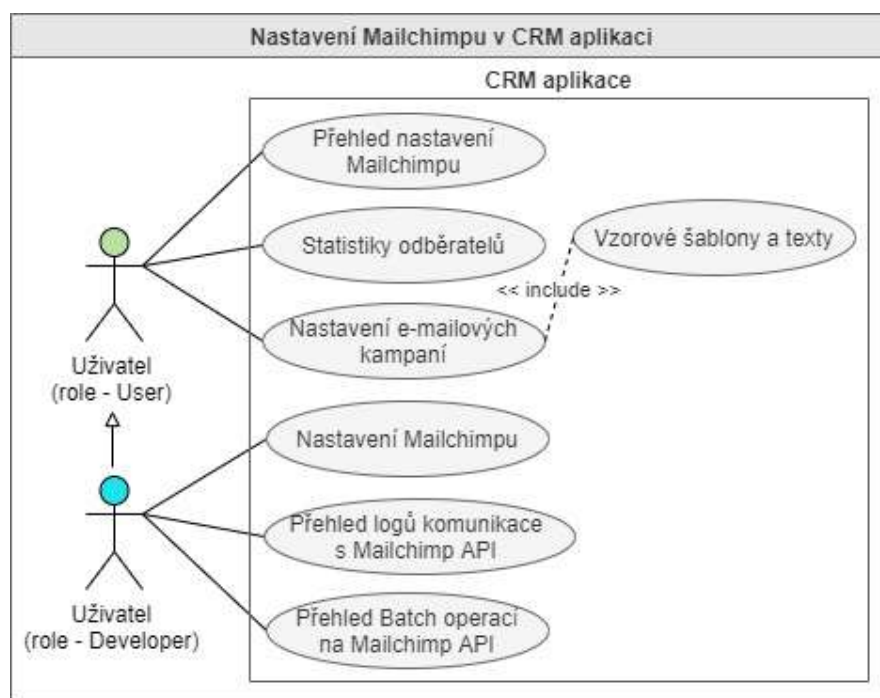


Obrázek 19: UC diagram – Nastavení transakčních e-mailů (Zdroj: vlastní)

Na obrázku číslo 19 jsou zobrazeny případy užití spojené s nastavením transakčních e-mailů. Jedná se o nastavení odchozích e-mailů a nastavení jejich šablon. V případě integrování služby Mandrill je zde navíc možnost z něj přidělit šablonu odchozímu e-mailu. Do logů o odchozích transakčních e-mailech a logů komunikace s Mandrill API může nahlížet pouze uživatel s rolí developer nebo admin.

3.10.2.7 Mailchimp

TODO Uživatel s rolí user může v CRM aplikaci prohlížet nastavení Mailchimu, statistiky odběratelů, nastavení e-mailových kampaní a nahlížet do nastavení Mailchimu, ale měnit nastavení může pouze developer. Uživatel s rolí developer může navíc spravovat odchozí Batch operace a logy komunikace s Mailchimp API.

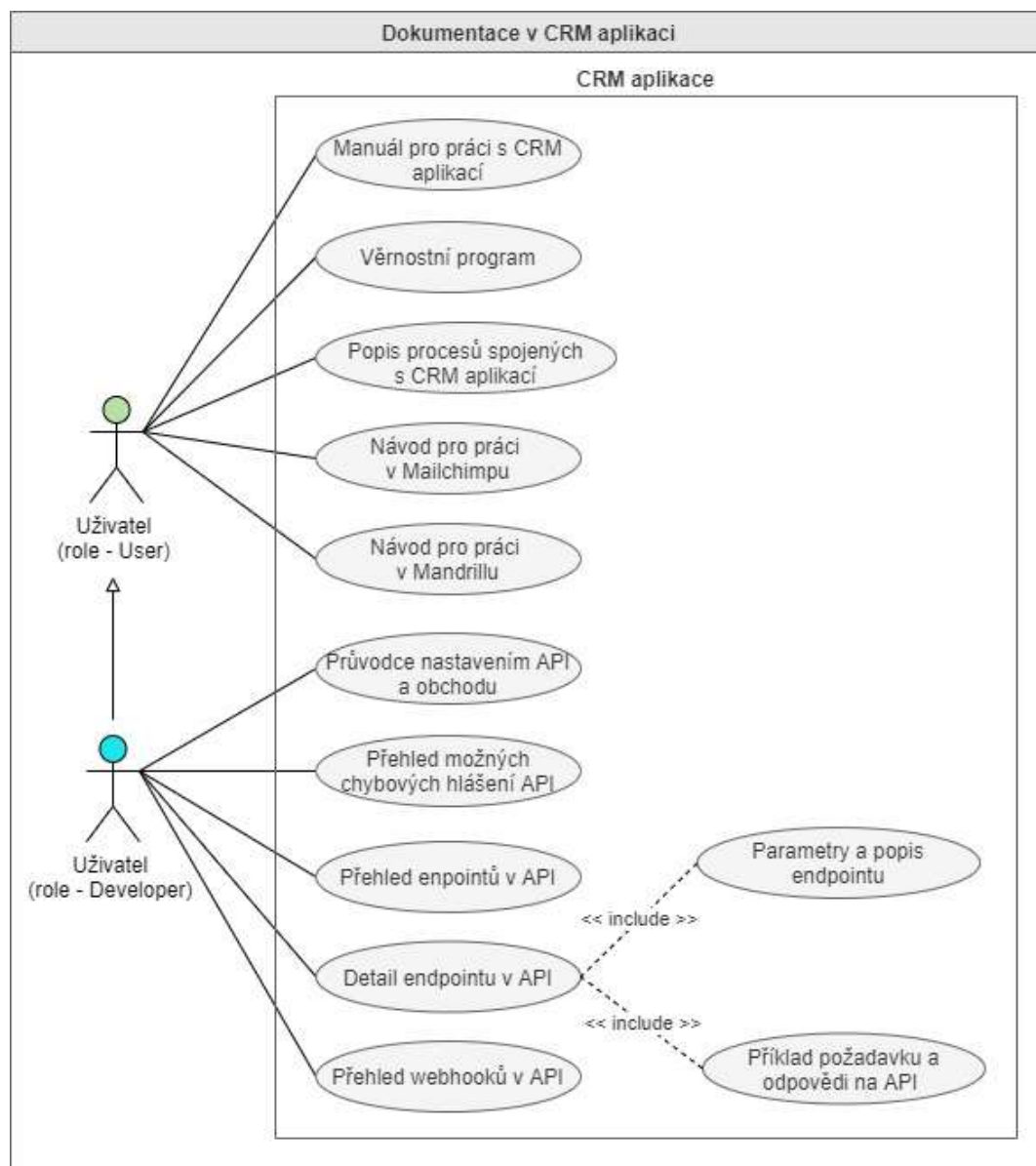


Obrázek 20: UC diagram – Nastavení Mailchimu (Zdroj: vlastní)

3.10.3 Dokumentace CRM aplikace

Na následující UC diagramu jsou znázorněny případy užití související s dokumentací v CRM aplikaci. Uživatel s rolí user může v dokumentaci přistupovat k manuálu pro práci se CRM aplikací, Mailchimpem a Mandrillem. Dále se dostane k přehledu funkcionalit věrnostního programu a popisu procesů spojených s používáním CRM aplikace.

Uživatel s rolí developer může v dokumentaci navíc najít průvodce nastavením CRM API a obchodu. Dále se může dostat k přehledům chybových hlášení, endpointů, parametrů a webhooků. V detailu endpointu může uživatel kromě jeho parametrů najít příklady požadavků a výsledných odpovědí na daný endpoint.



Obrázek 21: UC diagram – Dokumentace v CRM aplikaci (Zdroj: vlastní)

4 Design

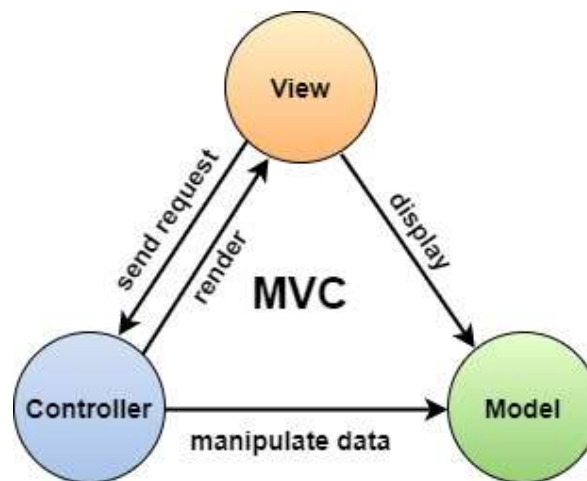
V této kapitole jsou popsány softwarové architektury, podle kterých je vytvořena CRM aplikace a CRM API. Dále jsou níže informace o databázovém modelu systému a návrh CRM API. Nakonec je zde možno nalézt nejdůležitější procesy spojené s CRM aplikací pomocí BPMN diagramů.

4.1 Architektura systému

Návrh a tvorba CRM aplikace vychází ze softwarové architektury MVC a u CRM API vychází ze softwarové architektury REST API.

4.1.1 MVC

MVC je architektura, která odděluje uživatelské rozhraní od datového modelu a řídicí logiky softwaru. Architektura MVC tedy pomyslně rozděluje systém na 3 různé samostatné části. Zkratka MVC znamená Model-View-Controller. [37]

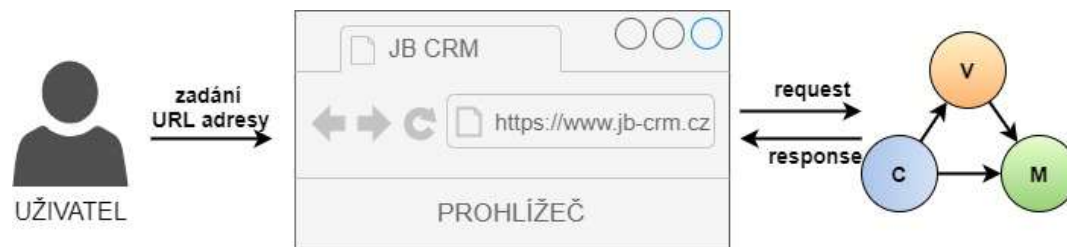


Obrázek 22: Diagram MVC architektury (Zdroj: [37])

Model reprezentuje funkční a datový základ systému. Dále je v něm zahrnuta business logika systému. Model je samostatný a neví o existenci Controlleru nebo View. Model zajišťuje veškeré operace pro manipulaci s daty a komunikaci s databází. [37]

View se stará o prezentování výsledku uživateli, který odeslal požadavek. K tomu používá šablonovací systém. View obsahuje veškeré uživatelské rozhraní systému, a také přímý odkaz na Model, aby mohl uživateli zobrazit jeho data.

Controller neboli řadič, reaguje na události a zpracovává požadavky od uživatele, na jejichž základě pak volá příslušné funkce v Modelu. Obdobou řadičů jsou v Nette frameworku presentery, které obsahují aplikační logiku systému. Controller obsahuje přímý odkaz na Model, aby mohl modifikovat jeho data. [38]



Obrázek 23: Request flow v MVC architektuře (Zdroj: [37])

Průběh toho, jak probíhá komunikace v MVC architektuře mezi jednotlivými komponentami, můžeme vidět na obrázcích 22 a 23. Komunikace začíná tím, že uživatel odešle požadavek přes internetový prohlížeč do systému. Tam požadavek obdrží View, které ho odešle ke zpracování do Controlleru. Řadič na základě požadavku rozhodne o tom, jaká akce se v Modelu zavolá. Model provede požadovanou akci a vrátí výsledek řadiči, který na základě výsledku zavolá View, které zobrazí uživateli výsledek požadavku.

4.1.2 RESTful API

REST je architektura rozhraní navržená pro distribuované systémy a nyní se převážně používá u webových služeb. Architektura REST nepopisuje konkrétní způsob implementace API, ale definuje způsob komunikace API, její vlastnosti, komponenty a postupy pro práci s daty. Když o API tvrdíme, že je RESTful, musí splňovat tyto vlastnosti: [44]

- Pracuje se síťovou architekturou klient-server
- Je bezstavová
- Využívá mezipaměť (Cache)
- Jednotné rozhraní a přístup ke zdrojům
- Skládá se z vrstev

Těchto pět vlastností popisuje hlavní myšlenky REST architektury. Díky tomu, že REST API komunikuje na principu klient-server, kde dochází k oddělení uživatelského rozhraní od datového uložení, se zvyšuje použitelnost API napříč různými platformami. Tento fakt potvrzuje i to, že se RESTful API skládá z vrstev.

Bezstavovost API znamená, že se během komunikace mezi klientem a serverem používá bezstavový protokol, například HTTP. Každý požadavek klienta na server musí obsahovat všechny nezbytné informace k tomu, aby byl server schopen požadavku porozumět a zpracovat ho. Díky tomu, že je RESTful API bezstavová, může API zpracovávat jednotlivé požadavky paralelně. [44]

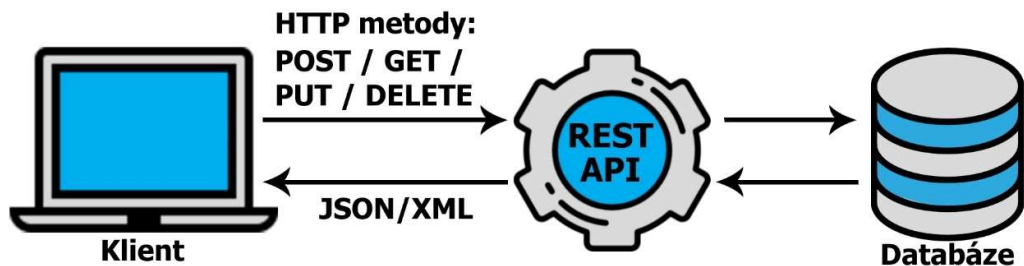
Správné využívání mezipaměti, neboli cache paměti, má pro API velký význam, protože mu může ušetřit spoustu práce. Je-li odpověď na daný požadavek uložena v cache paměti klienta, tak není nutné, aby server zpracovával požadavek znovu. K tomu, aby mohla být odpověď uložena do cache paměti, musí být v data v odpovědi implicitně, nebo explicitně označena jako cacheable, nebo non-cacheable. Cacheable odpověď je možné znovu použít a je jí umožněno uložit se do cache paměti. Non-cacheable odpověď není možné uložit do cache paměti, ani ji znovu využít. [44]

Jednotné rozhraní a přístup ke zdrojům souvisí s tím, jak probíhá odeslání požadavku do RESTful API. Zdroji mohou být data, stejně jako stav aplikace. Všechny zdroje mají svůj vlastní URI identifikátor, který je reprezentován textovým řetězcem s jasně definovanou strukturou. REST architektura definuje čtyři základní metody přístupu ke zdrojům, které nesou označení CRUD. Souhrnné označené CRUD znamená vytvoření dat (create), získání dat (retrieve), editaci dat (update) a odstranění dat (delete). Pokud tyto čtyři metody nestačí potřebám API, tak je v REST architektura možné přidat další metody s vlastním významem. [45]

Základní metody přístupu ke zdrojům v HTTP protokolu:

- POST (Create)
- GET (Retrieve)
- PUT (Update)
- DELETE (Delete)

Odpověď API na požadavek může být vrácena v různých formátech. Mezi nepoužívanější patří JSON, XML, RSS nebo YAML.



Obrázek 24: Ukázka komunikace v REST architektuře (Zdroj: vlastní)

Každá odpověď API na požadavek obsahuje stavový kód, podle kterého je možné odpověď vyhodnotit. Stavových kódů existuje mnoho, zde jsou uvedeny pouze ty nejdůležitější.

Kód	Název	Popis
200	OK	Požadavek proběhl v pořádku
201	Created	Při POST byl vytvořen nový obsah
400	Bad Request	Požadavek na server je nějakým způsobem nečitelný
401	Unauthorized	Klient není ověřený
403	Forbidden	Klient nemá přístup k danému obsahu
404	Not Found	Zdroj nebyl nalezen
405	Method Not Allowed	Zdroj není dostupný pro tuto metodu
429	Too-Many Requests	Klient překročil maximální povolený počet

Tabulka 2: Stavové kódy odpovědí (Zdroj: [45])

Na následujícím obrázku je znázorněn příklad požadavku klienta a výsledná odpověď, kterou na požadavek odešle API. Komunikace probíhá přes protokol http a požadavek i odpověď jsou ve formátu JSON.

Požadavek	Odpověď
<pre>POST /crm/customer HTTP/1.1 Content-Type: application/json; charset=utf-8 { "customer": { "email": "beckaj00@jcu.cz", "first_name": "Jan", "last_name": "Bečka" } }</pre>	<pre>HTTP/1.1 201 OK Content-Type: application/json; charset=utf-8 { "customer": { "id": "2548", "email": "beckaj00@jcu.cz", "first_name": "Jan", "last_name": "Bečka" } }</pre>

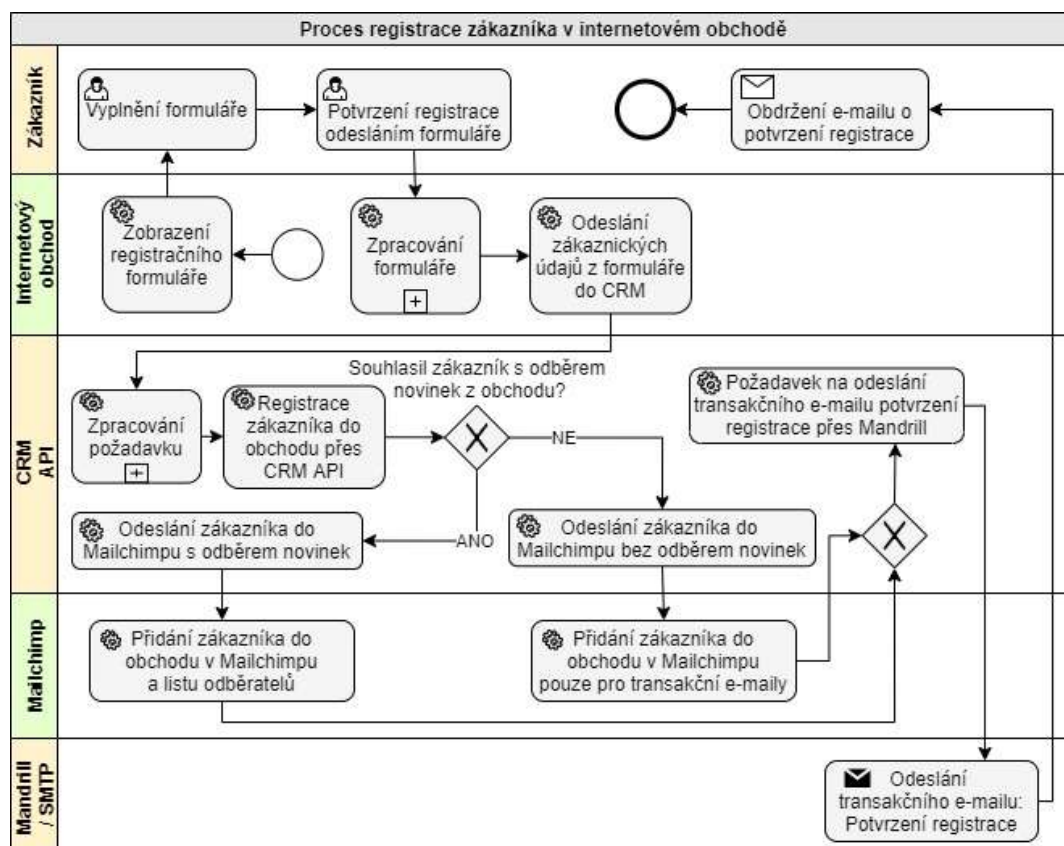
Obrázek 25: Ukázka požadavku klienta a odpovědi API v REST architektuře (Zdroj: vlastní)

4.2 Procesy

V této kapitole jsou namodelovány nejdůležitější procesy, které vznikají při používání CRM aplikace. Procesy jsou popsány v grafické notaci BPMN.

4.2.1 Registrace zákazníka

Proces na obrázku níže popisuje registraci zákazníka do internetového obchodu. Vyplnění registračního formuláře provádí zákazník na stránkách internetového obchodu, který následně formulář zpracuje a odešle do CRM API. Zákazník je následně zaregistrován a jeho údaje se odešlou do služby Mailchimp. Nakonec je zákazníkovi přes Mandrill nebo SMTP server odeslán transakční e-mail s potvrzením registrace.

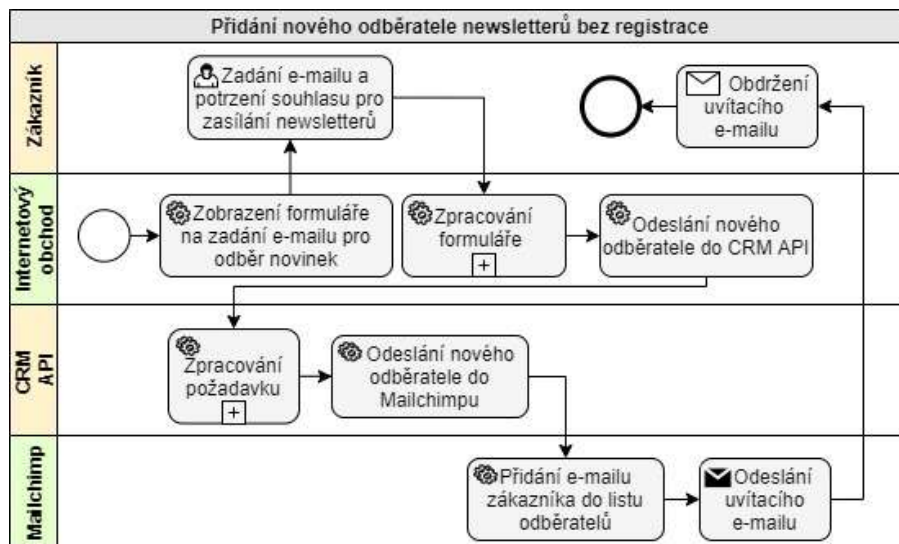


Obrázek 26: BPMN – Proces registrace zákazníka v internetovém obchodě (Zdroj: vlastní)

4.2.2 Přidání odběratele newsletterů bez registrace

K přidání odběru newsletterů musí zákazník vyplnit svůj e-mail do formuláře v internetovém obchodě. E-mail zákazníka z obchodu odeslán do CRM API, která

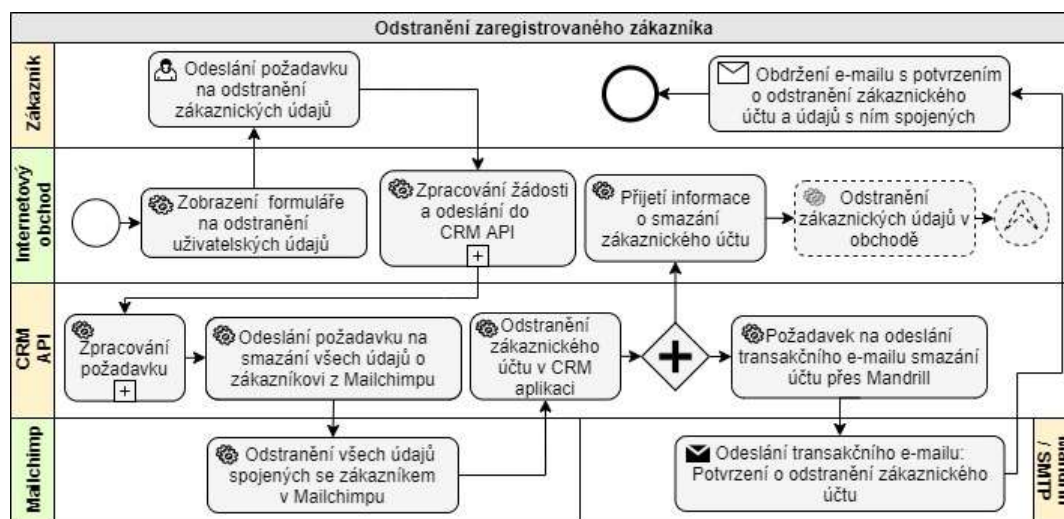
zařídí jeho přidání do listu odběratelů v Mailchimu. Z Mailchimu je nakonec zákazníkovi odeslán uvítací e-mail s potvrzením o přidání odběru.



Obrázek 27: BPMN – Proces přidání odběratele bez registrace (Zdroj: vlastní)

4.2.3 Odstranění zákaznického účtu

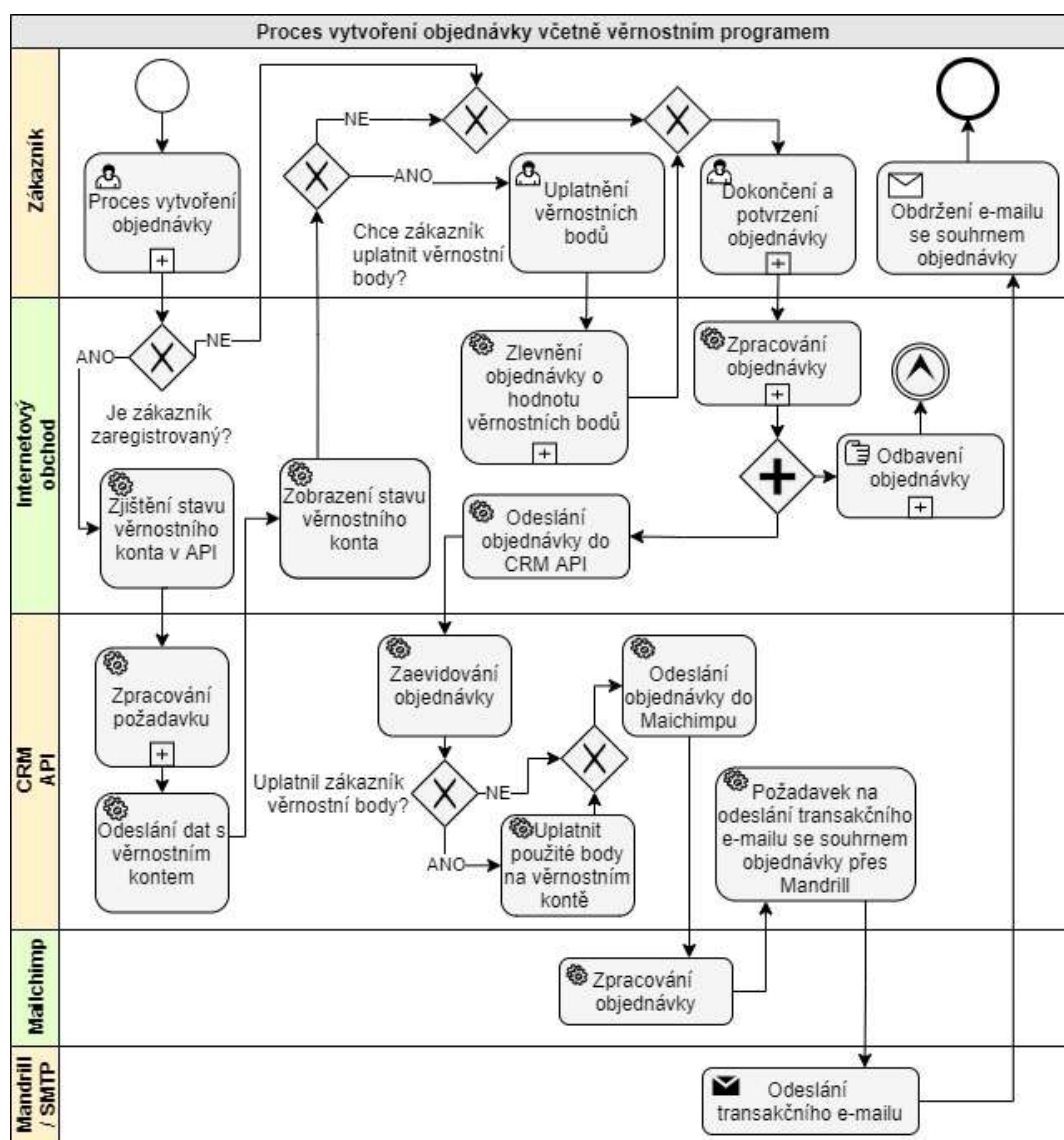
Proces odstranění zákaznického účtu se provede v případě, že zaregistrovaný zákazník chce odstranit svůj účet z internetového obchodu. Po odeslání požadavku do API se ze CRM aplikace odstraní veškerá data zákaznického účtu. Data se smažou také ve službě Mailchimp, kam se zákaznické údaje odesílají. V poslední kroku je zákazníkovi odeslán transakční e-mail s potvrzením o odstranění jeho účtu z internetového obchodu.



Obrázek 28: BPMN – Proces odstranění zákaznického účtu (Zdroj: vlastní)

4.2.4 Vytvoření objednávky s věrnostním programem

Proces vytvoření objednávky je popsán i s použitím věrnostního programu. K tomu, aby mohl být zákazník zařazen do věrnostního programu a následně uplatňovat věrnostní body, musí být v internetovém obchodě zaregistrován. Na objednávku nemůže zákazník uplatnit více bodů, než má na svém kontě a zároveň se nesmí dostat cena objednávky do záporného stavu. Stav věrnostního konta a hodnotu bodů obchod zjistí přes API. Po zpracování objednávky obchodem jsou její data odeslána do API. API objednávku zpracuje, uloží do systému a odešle do Mailchimpu. Nakonec je zákazníkovi odeslán transakční e-mail se souhrnem objednávky.

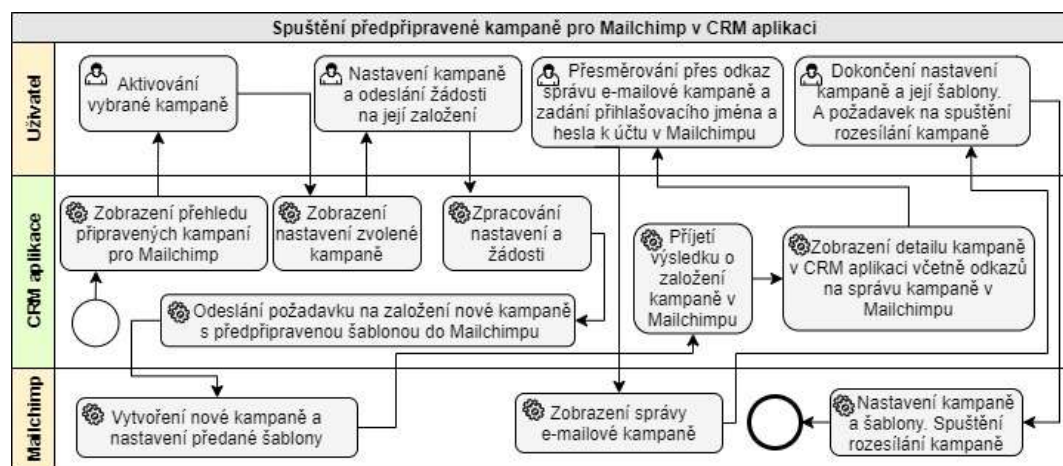


Obrázek 29: BPMN – Proces vytvoření objednávky včetně věrnostního programu (Zdroj: vlastní)

Proces uvedený výše modeluje akce související se změnou stavu objednávky. Pokaždé když internetový obchod indikuje změnu stavu objednávky, odešle stav dané objednávky do CRM API. Podle stavu objednávky se rozhodne, jaká větev procesu se provede. Změna stavu se uloží do CRM aplikace, následně se ve většině případů odešle do Mailchimu a nakonec se odešle transakční e-mail zákazníkovi se stavem objednávky. Při zpracování stavu objednávky „Dokončená“ jsou registrovaným zákazníkům v rámci věrnostního programu přičteny na konto body.

4.2.6 Spuštění připravené kampaně pro Mailchimp

Vytvoření e-mailové kampaně pro Mailchimp v CRM aplikaci a její následné spuštění, probíhá tak, jak je znázorněno na obrázku níže. Uživatel si nejprve vybere v CRM aplikaci z připravených kampaní pro Mailchimp, následně vybranou kampaň aktivuje, čímž se dostane na její nastavení. Po dokončení nastavení v CRM aplikaci se kampaň vytvoří v Mailchimu. Uživatel poté pokračuje v nastavení kampaně v Mailchimu, kde může kampaň a její šablonu upravovat dle svých potřeb. Po dokončení nastavení spustí uživatel kampaň, čímž se zahájí její rozesílání.

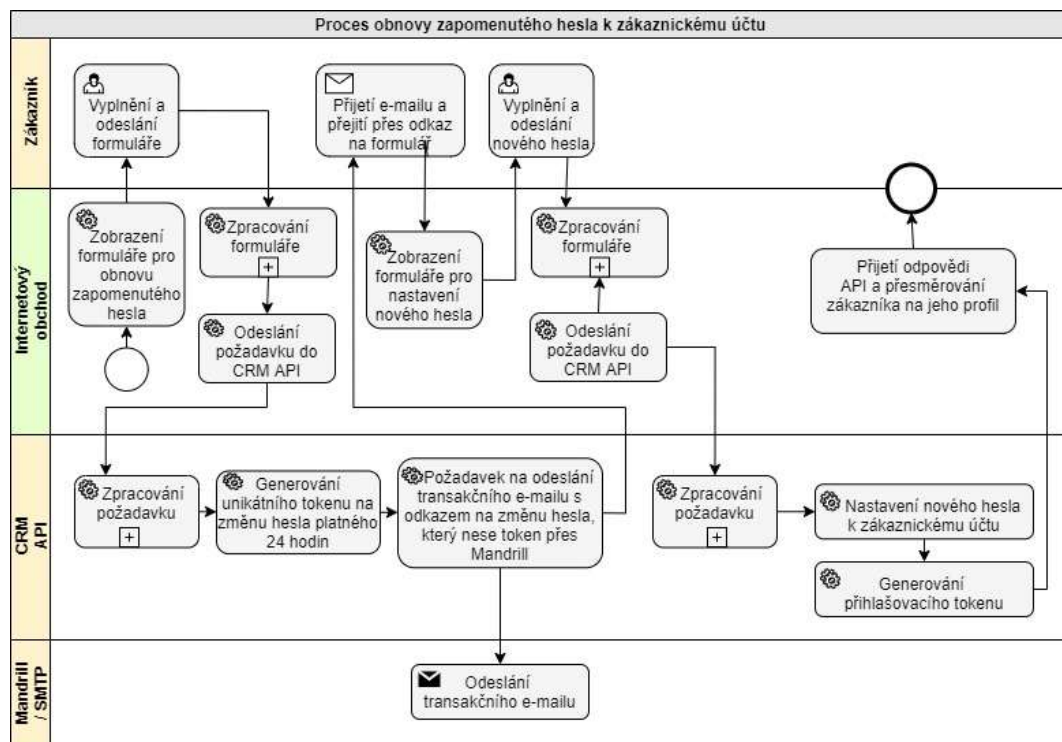


Obrázek 31: BPMN – Proces spuštění kampaně pro Mailchimp (Zdroj: vlastní)

4.2.7 Zapomenuté heslo k zákaznickému účtu

Proces níže popisuje obnovu zapomenutého hesla k účtu zákazníka. Zákazníkovi je vygenerován 24 hodin platný token, který mu je odeslán e-mailem. V e-mailu zákazník nalezne odkaz na formulář internetového obchodu pro změnu zapomenutého hesla, kde vyplní nové heslo. Vygenerovaný token je předán parametrem v odkazu v e-mailu a

obchod ho pošle do CRM API společně s novým heslem. Podle tokenu API ověří totožnost zákazníka, který žádá o nové heslo.



Obrázek 32: BPMN – Proces zapomenuté heslo k účtu (Zdroj: vlastní)

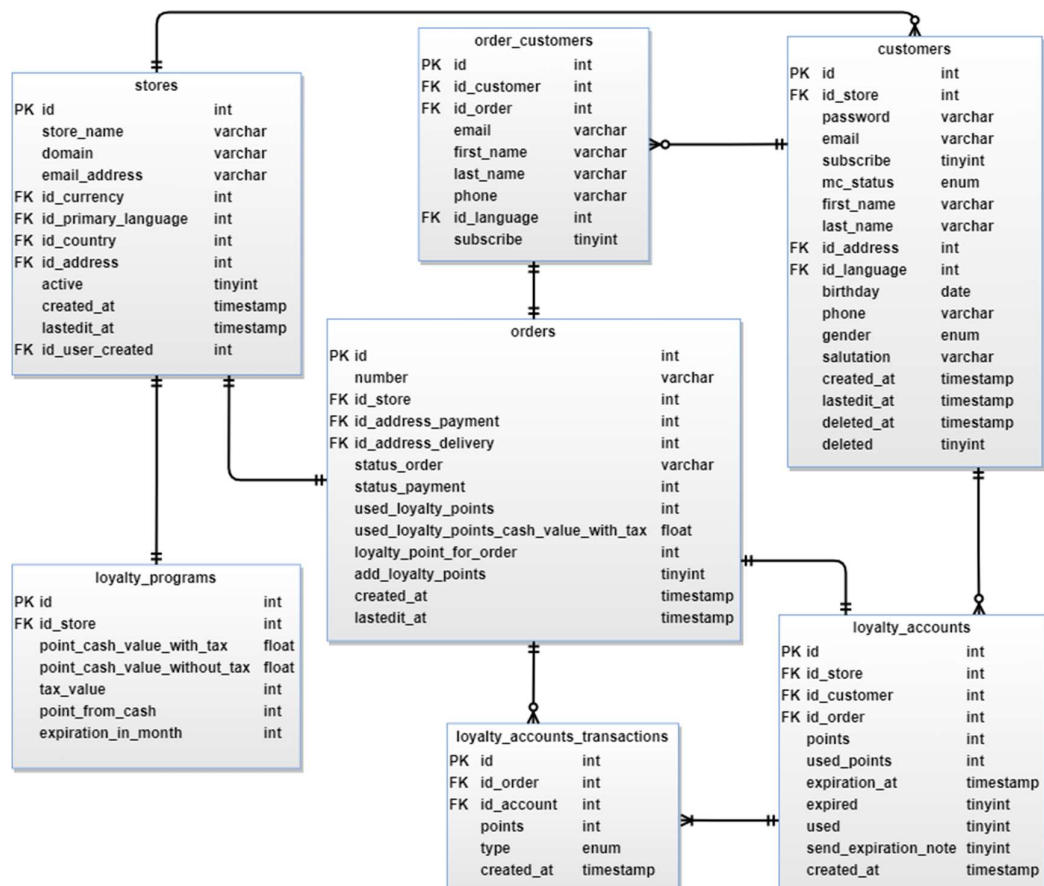
4.3 Datový model databáze

Text níže popisuje způsob ukládání dat CRM aplikace v databázi. Databázový model je popsán pomocí ER diagramu, jehož vybrané části jsou v této kapitole popsány. Kompletní datový model databáze je k dispozici v příložených souborech této práce.

4.3.1 Věrnostní program

Na obrázku číslo 33 je znázorněný ER diagram věrnostního programu. Každý obchod v aplikaci má vlastní nastavení věrnostního programu. Jedním ze způsobů získávání věrnostních bodů je odměňování zákazníků za dokončené objednávky, proto má entita orders vazby na entity věrnostního programu. Dále jsou zde znázorněny věrnostní konta zákazníků, které reprezentují přičtené body. Zákazník může mít několik kont, protože připsané věrnostní body mají podle data dokončení objednávky rozdílnou dobu expirace. Na konta jsou navázány transakce, do kterých se ukládají veškeré operace provedené na věrnostním kontě zákazníka, ať už se jedná o připsání, nebo uplatnění bodů.

Entity stores, customers, orders a order_customers obsahují vazby na entity, které nejsou na tomto diagramu zahrnuty.



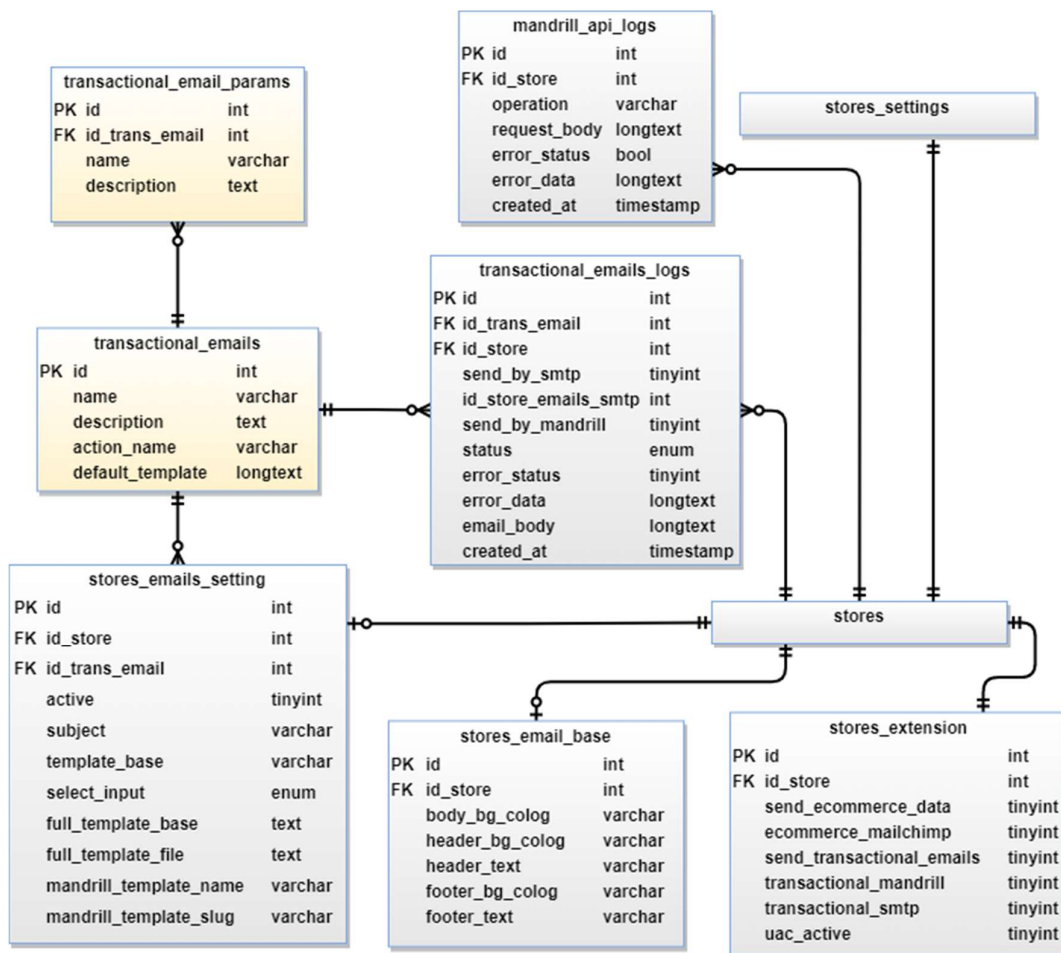
Obrázek 33: ER diagram – věrnostní program (Zdroj: vlastní)

4.3.2 Transakční e-mail

ER diagram na obrázku 34 znázorňuje ukládání dat, které souvisí transakčními e-maily. V entitě stores_extensions se uchovávají data související s nastavením obchodu, jako odesílání e-commerce dat a způsob rozesílání transakčních e-mailů. Transakční e-mail lze buďto zakázat, nebo odesílat přes SMTP protokol či službu Mandrill. Entity transactional_emails a transactional_email_params obsahují transakční e-mail který je možné prostřednictvím CRM aplikace rozesílat společně s předpřipravenou šablonou.

Šablony pro e-mail je možné vytvářet přímo v CRM aplikaci, nahrát je ze souboru nebo v případě integrace Mandrillu můžeme použít šablonu z této služby. Tyto informace se uchovávají v stores_emails_setting. Pokud se šablona tvoří v CRM aplikaci používá se entita stores_email_base, která uchovává výchozí nastavení e-mailů, jako je hlavička nebo patička.

Nastavení pro komunikaci s Mandrillem nebo pro vlastní SMTP server se ukládá v entitě stores_settings. Pokud se pro odesílání používá služba Mandrill, pak se komunikace s Mandrill API loguje a ukládá do mandrill_api_logs. V aplikaci se také logují veškeré odchozí transakční e-maily. Tyto záznamy se uchovávají v tabulce transactional_emails_log.

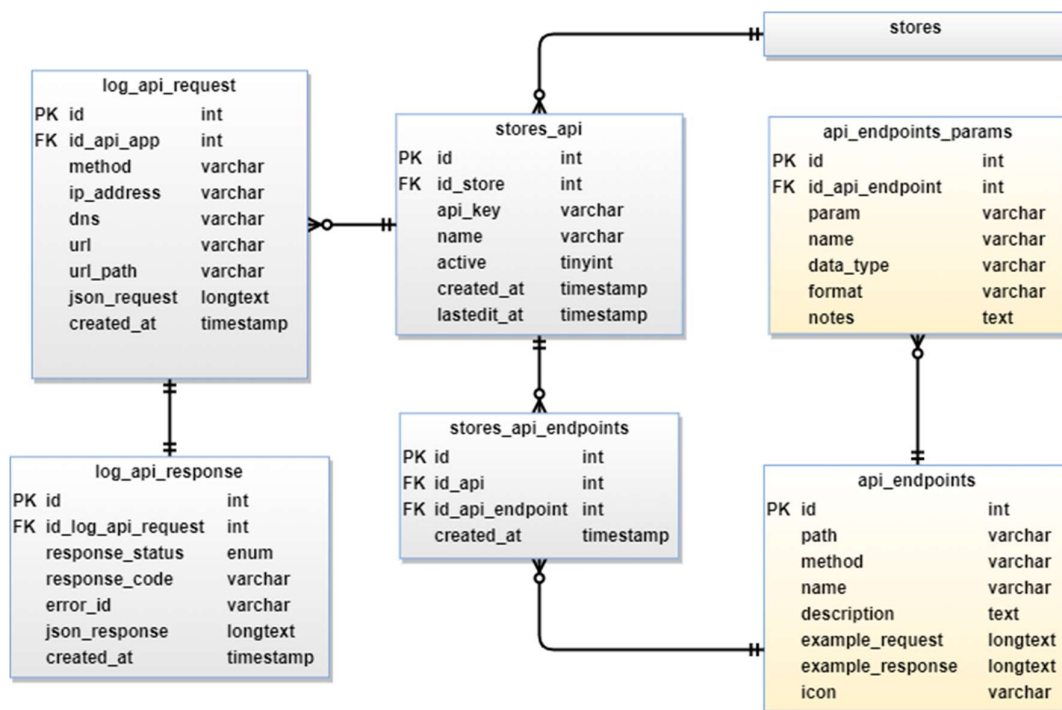


Obrázek 34: ER diagram – transakční e-maily (Zdroj: vlastní)

4.3.3 CRM API

ER diagram na obrázku 35 představuje způsob ukládání dat spojených s CRM API. Základem jsou entity api_endpoints a api_endpoints_params, protože obsahují všechny dostupné endpointy API a jejich parametry. Přístup do API se vždy vytváří pro určitý obchod. Základ přístupu je pak uložen v tabulce store_api a to jaké endpointy jsou přístupu přiděleny se uchovává v tabulce store_api_endpoints.

Veškerá komunikace s CRM API se loguje. Ukládá se jak požadavek na API, tak odpověď na daný požadavek. K uchování logů slouží tabulky `log_api_request` a `log_api_response`.

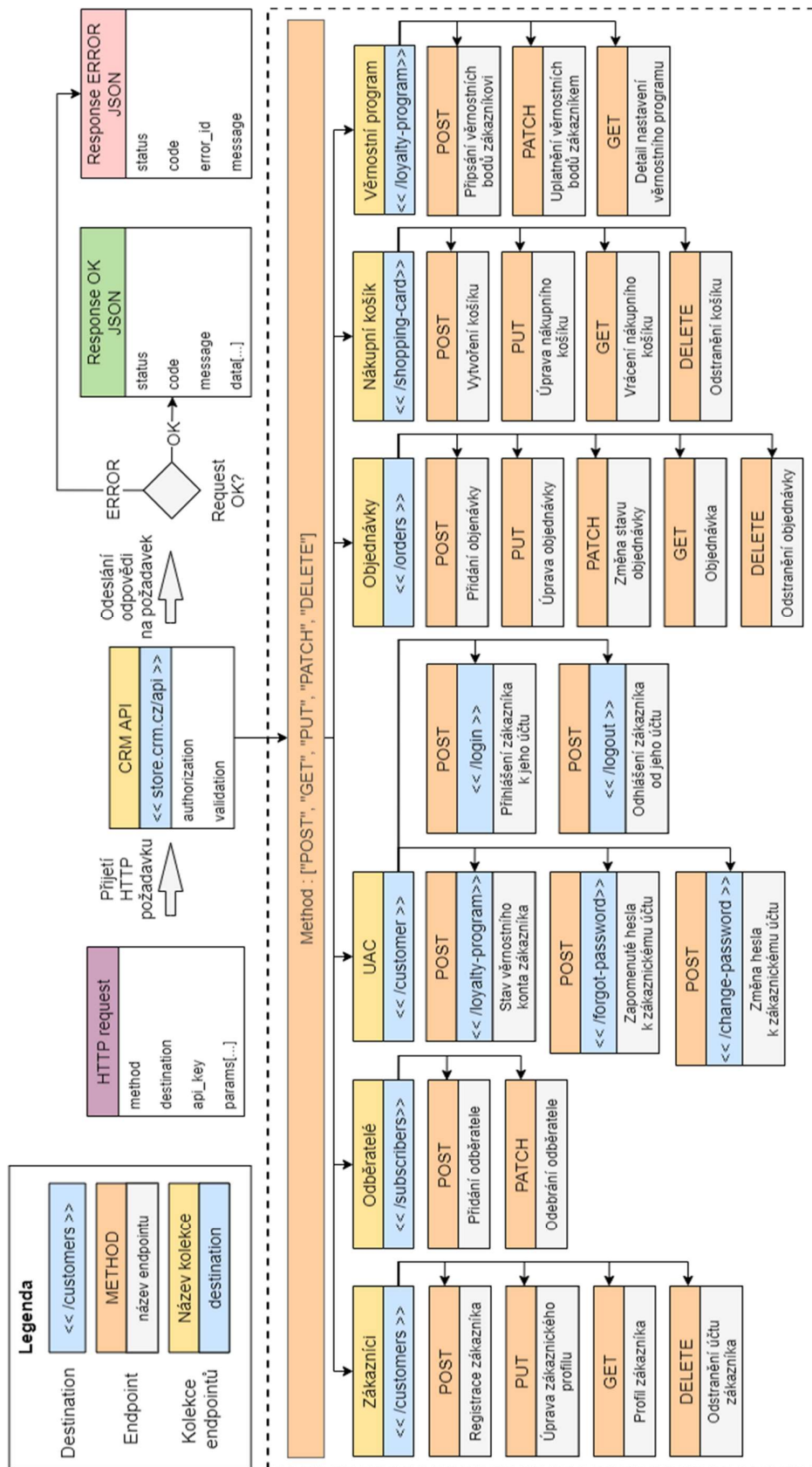


Obrázek 35: ER diagram – Mailchimp (Zdroj: vlastní)

4.4 Návrh CRM API

Návrh, jak CRM API komunikuje a jaké odpovědi vrací na požadavky, je k dispozici na obrázku 36. V návrhu jsou znázorněny jednotlivé endpointy, kterými API disponuje. Každý endpoint má své destination, neboli URI. Pokud má více endpointů stejné URI, rozhodne o tom, jaká akce se provede typ metody v HTTP protokolu. Tato situace nastává u endpointů, které pracují s daty jednoho objektu (např. zákazník).

Kompletní diagram API s parametry endpointů a příklady odpovědí je k dispozici v příloze této práce.



Obrázek 36: Návrh endpointů v CRM API (Zdroj: vlastní)

5 Použité technologie

V této kapitole jsou popsány technologie, které jsou použity v projektu. Technologie jsou rozděleny na klientskou a serverovou část řešení.

5.1 Backend

Nette framework a MySQL jsou technologie použité pro backend, neboli serverovou část řešení.

5.1.1 Nette framework

Nette je open source framework určený pro tvorbu webových aplikací stavěný tak, aby byl co nepoužitelnější a nejvstřícnější. Za autora Nette je považován český vývojář David Grudl. O další rozvoj frameworku Nette se stará organizace Nette Foundation, ve které působí David Grudl jako hlavním vývojářem. Vývoj Nette trvá už více než 10 let, a protože se jedná o framework, na kterém pracují čeští vývojáři, těší se velké oblibě v České republice. Výhodou je, že celá dokumentace k Nette je kromě anglického jazyka také v češtině. [20] Popularitu Nette frameworku nejen v České republice dokládá anketa časopisu SitePoint z roku 2015 s názvem „Best PHP Framework for 2015“, neboli v češtině „Nejlepší PHP framework pro rok 2015“, kde skončil Nette framework na třetím místě. [21]

Nyní je k dispozici Nette ve verzi 3 a mezi jeho hlavní vlastnosti patří podpora PHP 8, HTML5 a AJAX. Nette dále podporuje vývoj v MVC architektuře, objektové programování a eliminuje jistá bezpečnostní rizika spojená s XSS, CSRF a dalšími možnými útoky. K tomu mu pomáhá také šablonovací systém Latte, který bude více popsán níže.

Nette je ve své podstatě soubor samostatně použitelných komponent pro PHP. Filozofie Nette klade mimořádný důraz na produktivitu, best practice a co možná nejlepší možnou bezpečnost. V Nette je možné vyvářet například webové stránky a aplikace, e-shopy, blogy, CMS systémy, různé API a další. K tomu je možné využít některé z velkého množství doplňků a komponent, které jsou k dispozici na www.componette.org. [22]

Díky rozsáhlé komunitě, se Nette neustále vyvíjí a přibývají do něj nové nástroje a vychytávky.

5.1.2 MySQL

MySQL je jeden z nejoblíbenějších relačních databázových systémů na světě. MySQL je open source systém a nyní ho vlastní společnost Oracle Corporation. Existuje již od roku 1995 a díky své výkonnosti a spolehlivosti, se MySQL stalo častou volbou i pro velké společnosti, jako je Facebook, Twitter, YouTube, Yahoo! a další. [24]

5.2 Klientská část

Mezi technologie použité na klientskou část řešení patří Latte, jQuery a LESS.

5.2.1 Latte

Latte je šablonovací systém, který je součástí Nette frameworku a překládá šablony na jednoduchý a optimalizovaný PHP kód. Disponuje kontextově sensitivním escapováním a kontrolou odkazů, pomocí kterých lze předcházet XSS útokům.

V šablonách je možné používat Latte tagy nebo makra. Systém nabízí celou řadu tagů, které programátorovi značně urychlují psaní kódu. V šablonách je možné používat různé funkce, pomocí kterých lze snadno upravovat a formátovat data. Tyto funkce se v Latte nazývají filtry. V Latte je možné psát šablony v klasickém HTML a snadno do něj propisovat PHP funkce. Není potřeba se nějak složitě učit syntaxi Latte, protože je úplně stejná jako PHP syntaxe. Při používání tagů a filtrů je zápis kódu ještě jednodušší. Programátor si v případě potřeby může vytvořit vlastní filtry a tagy. Latte dále disponuje pokročilou tvorbou layoutů, vnořenými bloky a dalšími nástroji. [23]

5.2.2 jQuery

JavaScriptová knihovna jQuery má jednoduchou syntaxi a disponuje velice širokou podporou u webových prohlížečů. Bez základní znalosti samotného JavaScriptu se programátor ovšem neobejde, protože vše co jde udělat pomocí jQuery, jde udělat i pomocí JavaScriptu. V jQuery to lze většinou udělat mnohem kratším zápisem kódu. S jQuery lze manipulovat s CSS nebo HTML, používat selektory, vytvářet události, používat různé efekty, přidávat prvkům animace, používat AJAX a spoustu dalšího. [25]

5.2.3 LESS

LESS je zpětně kompatibilní rozšíření pro CSS. CSS neboli kaskádové styly se používají pro popis způsobu zobrazení elementů na webových stránkách napsaných například v HTML. Hlavním smyslem CSS je umožnit vývojářům oddělit definování vzhledu od struktury HTML. Díky tomu je možné psát styly do externího souboru a poté je pouze vložit do hlavičky HTML dokumentu. [26]

LESS rozšiřuje CSS o dynamické prvky, proměnné, mixiny, zanořená pravidla, výpočty a další funkce. Pro jeho použití je nutné zajistit jeho kompilaci do CSS. To zařizuje podpůrný nástroj Gulp, který je popsán níže. [27]

5.3 Podpůrné nástroje

Podpůrné nástroje usnadňují programátorovi práci a umožňují mu spoustu věcí automatizovat. V textu níže jsou popsány ty, které jsou použity v tomto projektu.

5.3.1 Gulp

Gulp je JavaScriptový nástroj pro automatizaci funkcí, který je k dispozici open source. Vznikl pod záštitou společnosti Fractal Innovations a za hlavního autora je považován Eric Schoffstall. [28] V tomto projektu je nástroj Gulp použit pro kompilaci LESS do CSS, minifikaci CSS, kompilaci a minifikaci JavaScriptu, komprimování obrázků, favicon a dalších. Gulp disponuje možností vytvořit externí adresu na lokálním serveru, na kterou je možné se připojit z různých zařízení, například tabletu, mobilu nebo počítače a v reálném čase sledovat změny na všech obrazovkách najednou.

5.3.2 Balíčkovací systémy a knihovny

Pomocí balíčkovacích nástrojů se do projektu přidávají nové knihovny velice jednoduše a rychle. To platí i pro jejich údržbu. Jak NPM, tak Composer disponuje velice nepřehledným počtem dostupným knihoven.

NPM

NPM je nástroj pro správu JavaScriptových balíčků a knihoven. Usnadňuje instalovat, upravovat a aktualizovat knihovny, které jsou definované v souboru package.json. Tyto knihovny je pak možné jednoduše nainstalovat pomocí příkazového řádku nebo terminálu. [29]

Composer

Composer je nástroj pro správu závislostí v PHP. Umožňuje deklarovat libovolně složité závislosti jednotlivých knihoven v souboru composer.json, a poté je jedním příkazem v terminálu nainstalovat. Knihovny v souboru composer.json je možné dále libovolně rozšiřovat a aktualizovat. [30]

6 Použitý software pro vývoj

V této kapitole je ve stručnosti popsán hlavní software, který byl používán při práci na tomto projektu.

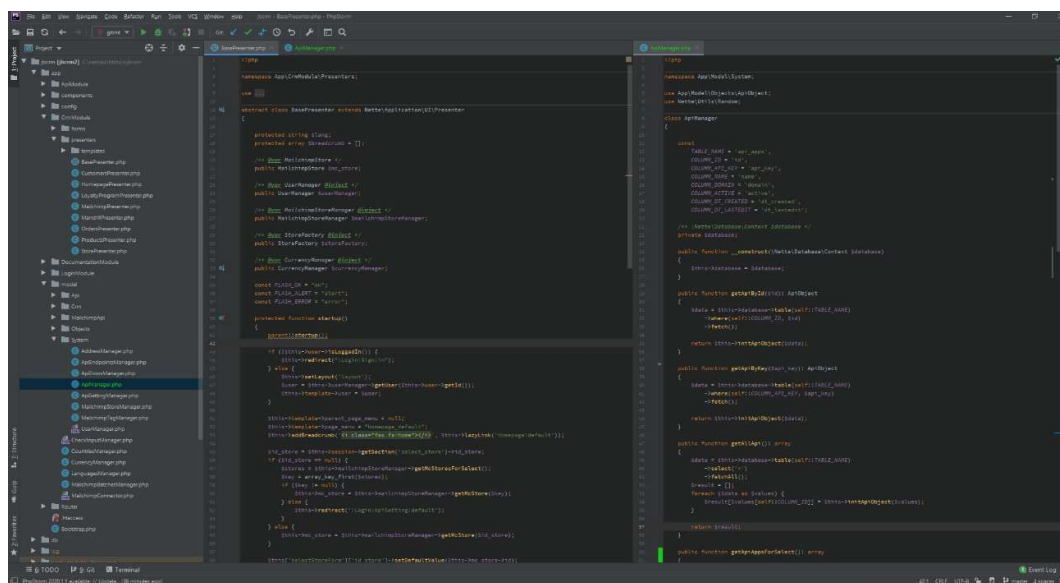
6.1 PhpStorm

PhpStorm je vývojové prostředí určené pro vývoj v PHP s podporou mnoha různých frameworků. Kromě nástrojů pro PHP disponuje PhpStorm také řadou nástrojů pro front-endové technologie, jako např. HTML, CSS, SASS, LESS, JavaScript a další. Dále disponuje nástroji pro ladění a testování, příkazovým řádkem nad projektem, podporou pro MySQL databázi, Gulpem, Composerem, Npm a dalšími užitečnými nástroji. [31]

PhpStorm v základu nedisponuje podporou pro Nette framework. Nicméně díky pluginům, které vytvořila vývojářská komunita kolem Nette, je tento problém již minulostí. Pluginy jsou dostupné jak pro samotné Nette, tak pro jeho šablonovací systém Latte.

PhpStorm vlastní a nadále vyvíjí společnost JetBrains s.r.o., která se zabývá vývojem vývojových prostředí pro celou řadu programovacích jazyků. [32]

Licence pro PhpStorm a další IDE poskytuje společnost JetBrains studentům zdarma.

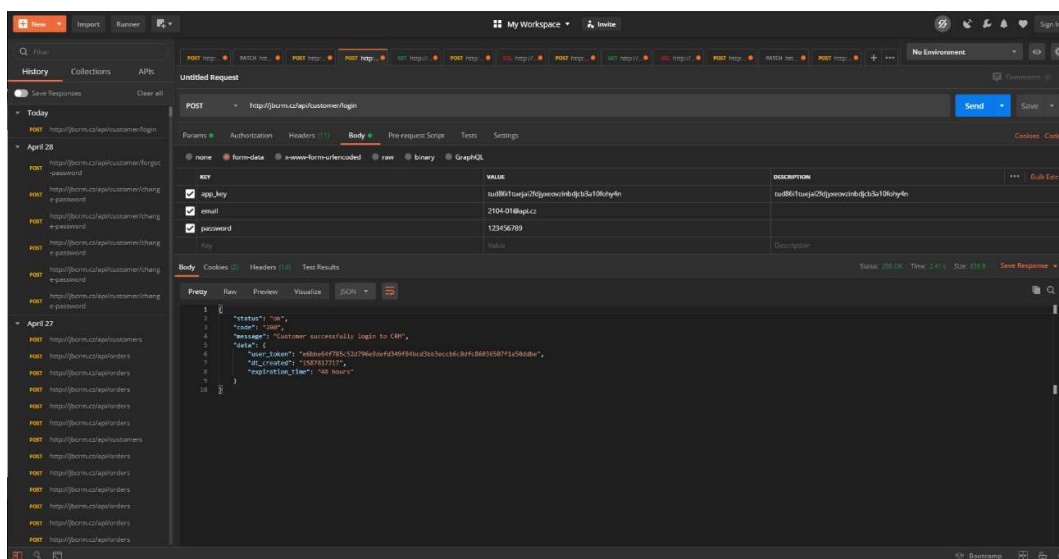


Obrázek 37: Ukázka IDE prostředí PhpStorm (Zdroj: vlastní)

6.2 Postman

Postman je multiplatformní vývojové prostředí pro API. Pomocí aplikace Postman je možné navrhovat, testovat a monitorovat REST a SOAP API. Dále umožňuje pracovat v týmu, automaticky generovat dokumentace k API, odesílat požadavky na API a následně zobrazovat obdrženou odpověď na požadavek. [33]

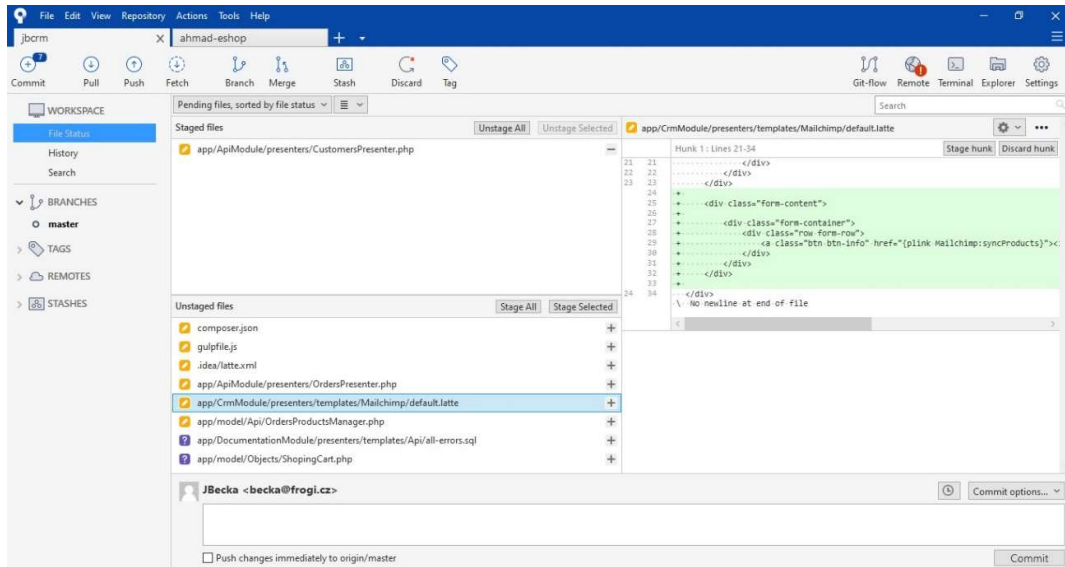
Základní verze softwaru Postman je k dostání zdarma a pro potřeby tohoto projektu je naprosto dostačující. Postman je pak k dostání v podobě placených balíčků, jejichž cena začíná na 12\$. Zakoupené balíčky zpřístupní uživateli různé vývojové nástroje, kolaboraci ve větších týmech, ověřování totožnosti uživatelů pomocí SSO a další. [34]



Obrázek 38: Ukázka IDE prostředí Postman určeného pro vývoj API (Zdroj: vlastní)

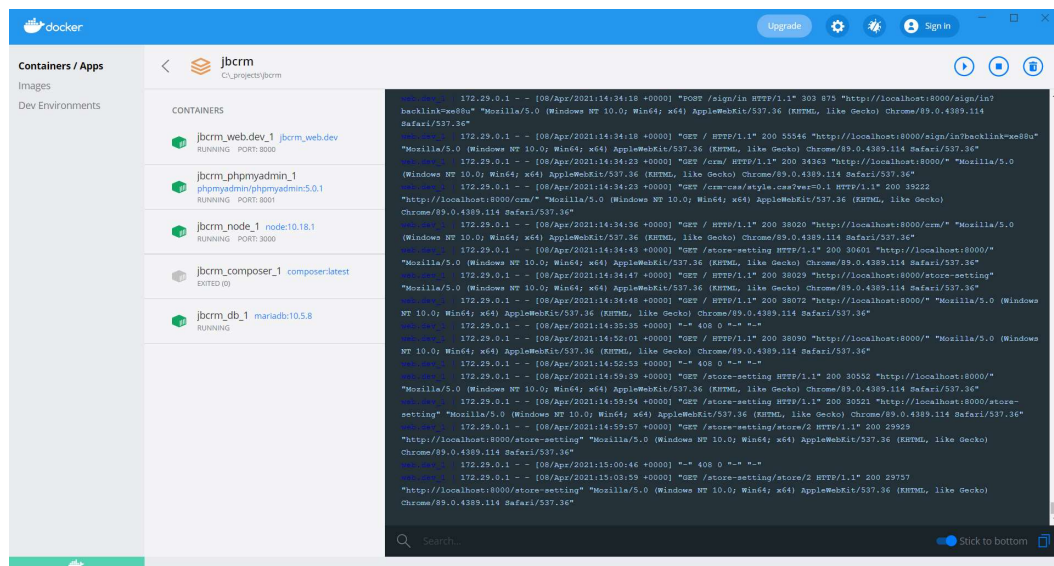
6.3 Sourcetree

Aplikaci Sourcetree vyvíjí společnost Atlassian, která ji poskytuje zdarma ke stažení. Aplikace poskytuje přehledné a snadno ovladatelné GUI verzovacího nástroje Git. Prostřednictvím Sourcetree je možné spravovat několik projektů zároveň a pracovat v týmech. Dále umožňuje zaznamenání změn v kódu(commits), správu větví v projektu (branches), spojování větví (merges), prohlížení historie změn v projektu, porovnávání řešení v různých stádiích vývoje a revizi kódu po ostatních členech týmu. [46]



Obrázek 39: Ukázka aplikace Sourcetree – GUI pro Git (Zdroj: vlastní)

6.4 Docker



Obrázek 40: Ukázka aplikace Docker určené pro Windows (Zdroj: vlastní)

Docker vyvíjí stejnojmenná společnost Docker, Inc. a byl představen světu v roce 2013. Od té doby získal značnou popularitu. Podle oficiálních stránek aktuálně používá Docker více jak 11 milionů vývojářů a běží na něm více jak 7 milionů aplikací. [35]

Docker umožňuje spouštět balíčky ve formě kontejnerů s využitím virtualizace na úrovni operačního systému. Díky tomu je možné si dopředu definovat prostředí, kde bude aplikace fungovat a prostřednictvím Dockeru je virtualizovat. Jako například webový server. Pro usnadnění ovládání Docker engine je použita oficiální aplikace Docker App určená pro Windows. [36]

7 Implementace

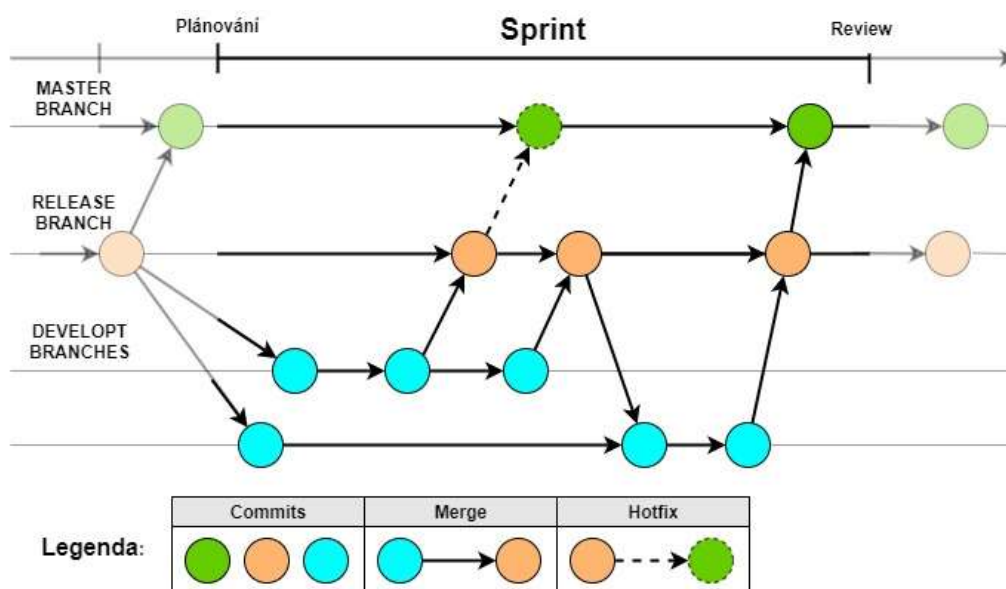
7.1 Zásady vývoje

Mezi zásady vývoje patří určení pravidel průběhu verzování naprogramovaného kódu, kterému je věnována samostatná podkapitola níže. Mezi další zásady vývoje, které byly na projektu uplatněny, patří dodržování:

- Zásad OOP během celého vývoje
- Pravidel MVC architektury při vývoji aplikace
- Pravidel REST architektury při vývoji API
- Pravidelného testování
- Obecných pravidel Coding Standards při psaní kódu
- Pravidel agilní metodiky SCRUM

7.1.1 Verzování kódu

Verzování kódu probíhá pomocí verzovacího nástroje Git. Pro práci s nástrojem Git se v projektu používá aplikace Sourcetree, která práci s ním značně zjednodušuje. Při verzování kódu se během vývoje aplikace dodržuje Git workflow, která je znázorněna na následujícím obrázku a popsána v textu pod ním.



Obrázek 41: Git workflow s metodikou SCRUM – znázornění jednoho sprintu (Zdroj: vlastní)

Master branch ve workflow reprezentuje hlavní větev, která představuje aktuální funkční verzi aplikace. Develop branches, neboli vývojové větve, představují průběh implementace jednotlivých úkolů z Product Backlogu. Ve chvíli, kdy je úkol dokončen, spojí se vývojová větev s Release branch. Nad každým commitem ve větvi Release se provádí revize kódu a testování vytvořeného řešení. Pokud commit projde úspěšně testováním, je uvolněn do větve Master, ze které proběhne nahrání řešení na server. Ovšem pokud je při testování ve větvi Release objevena chyba, pokračuje se v práci na úkolu, dokud není chyba opravena. Po opravě se výsledek opět odešle do větve Release, kde znovu probíhá proces testování.

7.2 CRM API

Jak je již zmíněno výše, API je pro napojení internetových obchodů do CRM aplikace tvořena jako RESTful. API je vytvořena jako modul CRM aplikace psaná v Nette frameworku. Požadavky na API probíhají pomocí HTTP protokolu s povinným SSL. Odpovědi jsou odeslány ve formátu JSON.

7.2.1 Směrování endpointů API

Směrování (routování) endpointů API probíhá překladem URL adresy zadané v požadavku na API. Adresa se rozloží a převede na akce v presenteru. Pro směrování v API je použit doplněk k Nette, s názvem Ublaboo ApiRouter. Díky tomuto doplňku je možné nastavit routování dle vlastností RESTful API. Pokud má endpoint vlastní URI, je jeho přístupová metoda vždy POST. V případě, že jedno URI sdílí více endpointů najednou, rozhodne o akci, kterou API provede, metoda v HTTP požadavku na API. Metody, které se v API objevují jsou POST, GET, PUT, PATCH a DELETE. Základní metody se používají podle REST architektury. Výjimkou je metoda PATCH, která je doplněna do architektury pro speciální požadavky na API.

```
$router[] = new ApiRoute('/api/loyalty-program', 'Api:LoyaltyProgram', [
    'methods' => [
        'POST' => 'addLoyaltyPoints', //Připsání věrnostních bodů na konto
        'PATCH' => 'applyLoyaltyPoints', //Uplatnění věrnostních bodů
        'GET' => 'getLoyaltyProgram' //Vrácení nastavení věrnostního programu
    ]
]);
```

Obrázek 42: Ukázka kódu – router CRM API (Zdroj: vlastní)

Pokud chce uživatel odeslat požadavek na endpoint s metodou POST na připsání věrnostních bodů, musí zadat adresu `.../api/loyalty-program/add-loyalty-points`.

7.2.2 Řízení přístupu do API

Na následujícím obrázku je znázorněno, jak v API probíhá kontrola zabezpečení komunikace (SSL) a kontrola typu metody HTTP protokolu. Tento proces probíhá při každém požadavku na API. Pokud není komunikace bezpečná nebo metoda není podporovaná, odešle API v odpovědi na požadavek chybové hlášení.

```
if (!$this->getHttpRequest()->isSecured()) {
    $this->sendApiResponseError("SSL required", IResponse::S403_FORBIDDEN,
        self::ERROR_SSL_REQUIRED);
}

if (!in_array($this->request->method, ["POST", "PUT", "GET", "PATCH", "DELETE"])) {
    $this->sendApiResponseError('method ' . $this->request->method . "not allowed",
        IResponse::S405_METHOD_NOT_ALLOWED, self::ERROR_METHOD);
}
```

Obrázek 43: Ukázka kódu – povinné SSL a kontrola HTTP metody (Zdroj: vlastní)

Dále musí každý požadavek na API obsahovat parametr `api_key`, podle kterého se ověřuje totožnost odesílatele požadavku. Pokud `api_key` chybí nebo je neplatný, odešle se chybové hlášení o neautorizovaném požadavku, jak je znázorněno na obrázku níže.

```
$this->sendApiResponseError('Unauthorized application',
    IResponse::S401_UNAUTHORIZED, self::ERROR_APP_KEY
);
```

Obrázek 44: Ukázka kódu – neautorizovaný požadavek (Zdroj: vlastní)

Přístupy do API je možné spravovat prostřednictvím CRM aplikace. Na obrázku 44 je ukázka toho, jak v aplikaci vypadá přehled přístupů do API. Při vytvoření nového přístupu je automaticky vygenerován náhodný a unikátní klíč. Přístup do API je možné deaktivovat, čímž se okamžitě zablokuje přidělený klíč. Přístup je dále možné znovu aktivovat nebo ho naopak úplně odstranit.

Nastavení CRM

Obchody

#	Název	Doména	Měna	Hlavní jazyk	Země	Vytvoření
1	Crm Stores	https://www.crmstore.cz/	CZK	cs	CZE	07.04.2021

+ Přidat nový obchod

Nastavení API

Nastavení API

#	Api key	Název API	Obchod	Aktivní	Vytvoření
1	jb8md27a85x470n374f08jkhyms8l1iuld3ztd5u6jcrm	test	Crm Stores	✓	07.04.2021

+ Přidat novou API

Obrázek 45: Ukázka obrazovky – nastavení přístupů do API (Zdroj: vlastní)

7.2.3 Druhy odpovědí API

API vrací na požadavky dva druhy odpovědí. Odpověď na úspěšně zpracovaný požadavek, která má status 'ok' a odpověď na neúspěšně zpracovaný požadavek, která má status 'error'. Součástí obou druhů odpovědí jsou vždy atributy 'status', 'code' a 'message'. Atribut 'status' reprezentuje druh odpovědi. Atribut 'code' udává stavový kód odpovědi, který je rovněž zaznamenán v HTTP protokolu odpovědi. Posledním společným atributem je 'message', která nese textovou zprávu.

Úspěšný požadavek má v odpovědi navíc atribut 'data', z jehož názvu vyplývá, že obsahuje návratová data. Neúspěšný požadavek obsahuje navíc atribut 'error_id', podle které je možné zjistit, o jakou chybu se konkrétně jedná, protože dokumentace k API disponuje seznamem chybových kódů. Výsledná odpověď je následně odeslána přes HTTP protokol ve formátu JSON.

Na obrázku níže je ukázka zápisu funkcí pro odesílání jak úspěšných, tak neúspěšných odpovědí. V ukázce je kromě obsahu odpovědi také vidět logování odpovědí.

```
public function sendApiResponseOk(?string $message, array $data, string $code =
IResponse::S200_OK)
{
    $this->getHttpResponse()->setCode($code);

    $response = [
        'status' => 'ok',
        'code' => $code,
        'message' => $message,
        'data' => $data
    ];

    $this->responseLogsManager->addLog($this->request_log_id, $response);

    $this->sendJson($response);
}

public function sendApiResponseError(?string $message, string $code, ?string $error_id = null)
{
    $this->getHttpResponse()->setCode($code);

    $response = [
        'status' => 'error',
        'code' => $code,
        'error_id' => $error_id,
        'message' => $message,
    ];

    $this->responseLogsManager->addLog($this->request_log_id, $response);

    $this->sendJson($response);
}
```

Obrázek 46: Ukázka kódu – funkce pro odesílání odpovědí na požadavky (Zdroj: vlastní)

7.2.4 Logování požadavků a odpovědí na API

Všechny požadavky a odpovědi API jsou ukládány do databáze. Logy jsou k dispozici pouze měsíc zpětně, kvůli úspoře datového prostoru v databázi. Logy může uživatel, který má potřebná oprávnění v CRM aplikaci procházet a případně si zobrazit detail logu, ve kterém jsou zobrazena kompletně všechna data z požadavku a odpovědi.

```
public function addLog(?int $id_request, array $response)
{
    $json = Json::encode($response, Json::PRETTY);

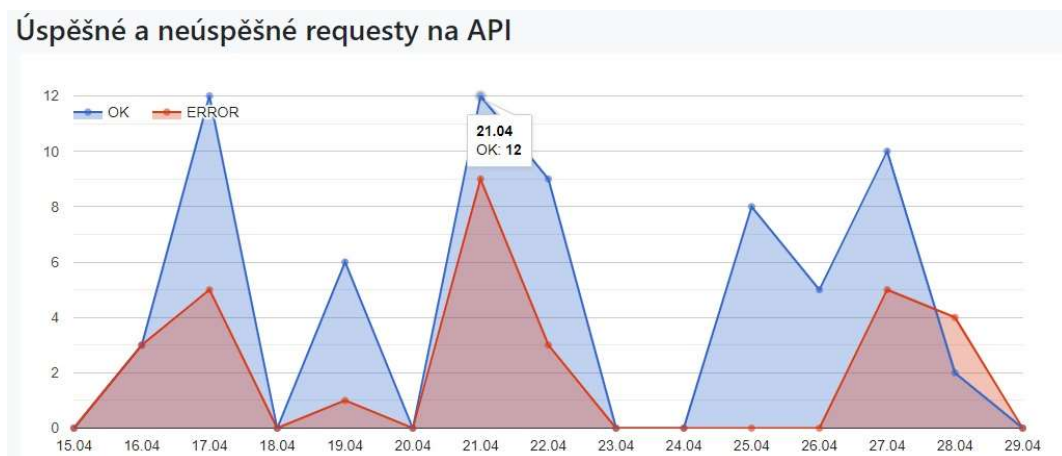
    $data = [
        self::COL_IdLogApiRequest => $id_request,
        self::COL_ResponseStatus => $response['status'],
        self::COL_ResponseCode => $response['code'],
        self::COL_JsonResponse => $json
    ];

    if ($response['status'] == 'error') {
        $data[self::COL_ErrorId] = $response['error_id'];
    }

    $this->database->table(self::TABLE_NAME)
        ->insert($data);
}
```

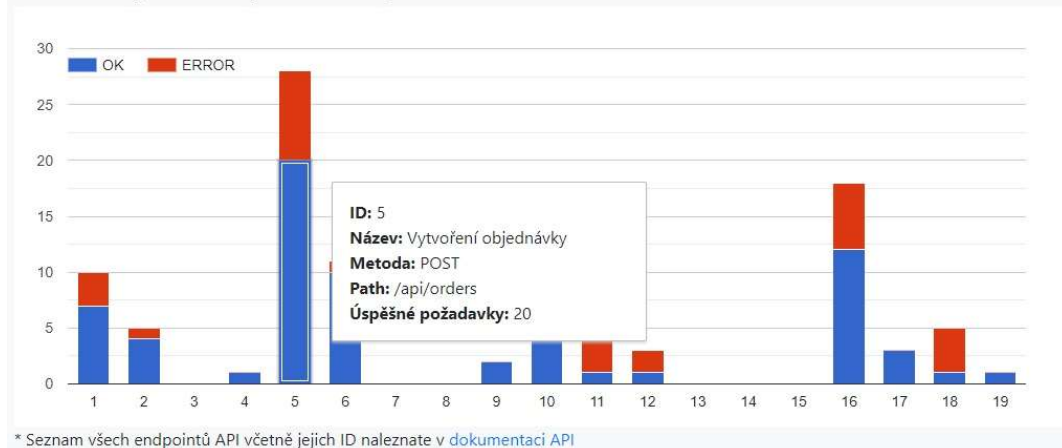
Obrázek 47: Ukázka kódu – uložení záznamu odpovědi do DB (Zdroj: vlastní)

Z dat uložených logů se vyhodnocují statistiky, které jsou k dispozici v CRM aplikaci. Statistiky zobrazují počty úspěšných a neúspěšných požadavků na API ve spojnicovém grafu. Dále je zde složený sloupcový graf, kde každý sloupec reprezentuje jeden endpoint v API a je složený z počtu úspěšných a neúspěšných požadavků na tento endpoint. Pokud je v API vytvořeno více přístupů, je ve statistikách navíc koláčový graf, který ukazuje poměry požadavků jednotlivých přístupů. Na obrázcích níže jsou snímky obrazovky z CRM aplikace, které zobrazují výše zmíněné statistiky.



Obrázek 48: Ukázka obrazovky – graf úspěšnosti požadavků na API (Zdroj: vlastní)

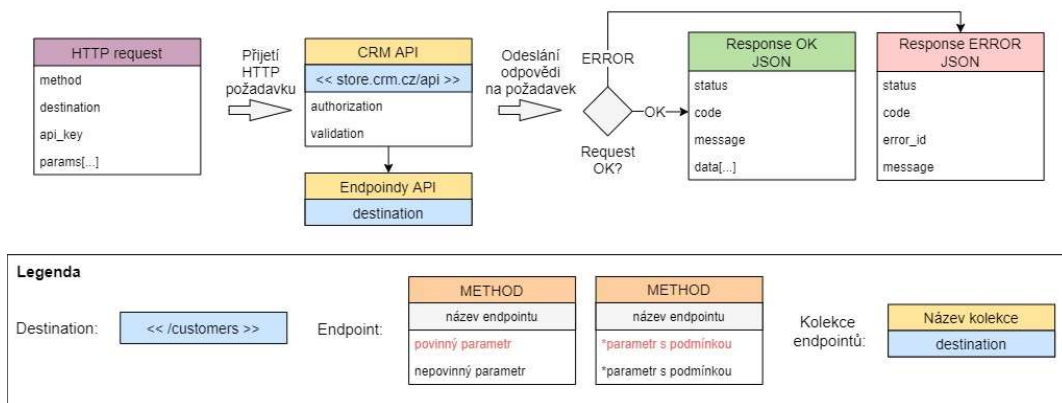
Požadavky na API podle endpointů



Obrázek 49: Ukázka obrazovky – graf požadavků na endpointy API (Zdroj: vlastní)

7.2.5 Vybrané endpointy API

Všechny endpointy, kterými API disponuje, jsou znázorněny na obrázku 36. V této kapitole jsou podrobněji popsány tři vybrané endpointy. Kompletní diagram endpointů s parametry a příklady odpovědí je k dispozici v souborech přílohy. Na následujícím obrázku je znázorněn diagram komunikace API a legenda k diagramům obsažených v této kapitole. Podrobnější popis endpointů API s příklady požadavků, parametrů a odpovědí je součástí dokumentace ve webové CRM aplikaci.



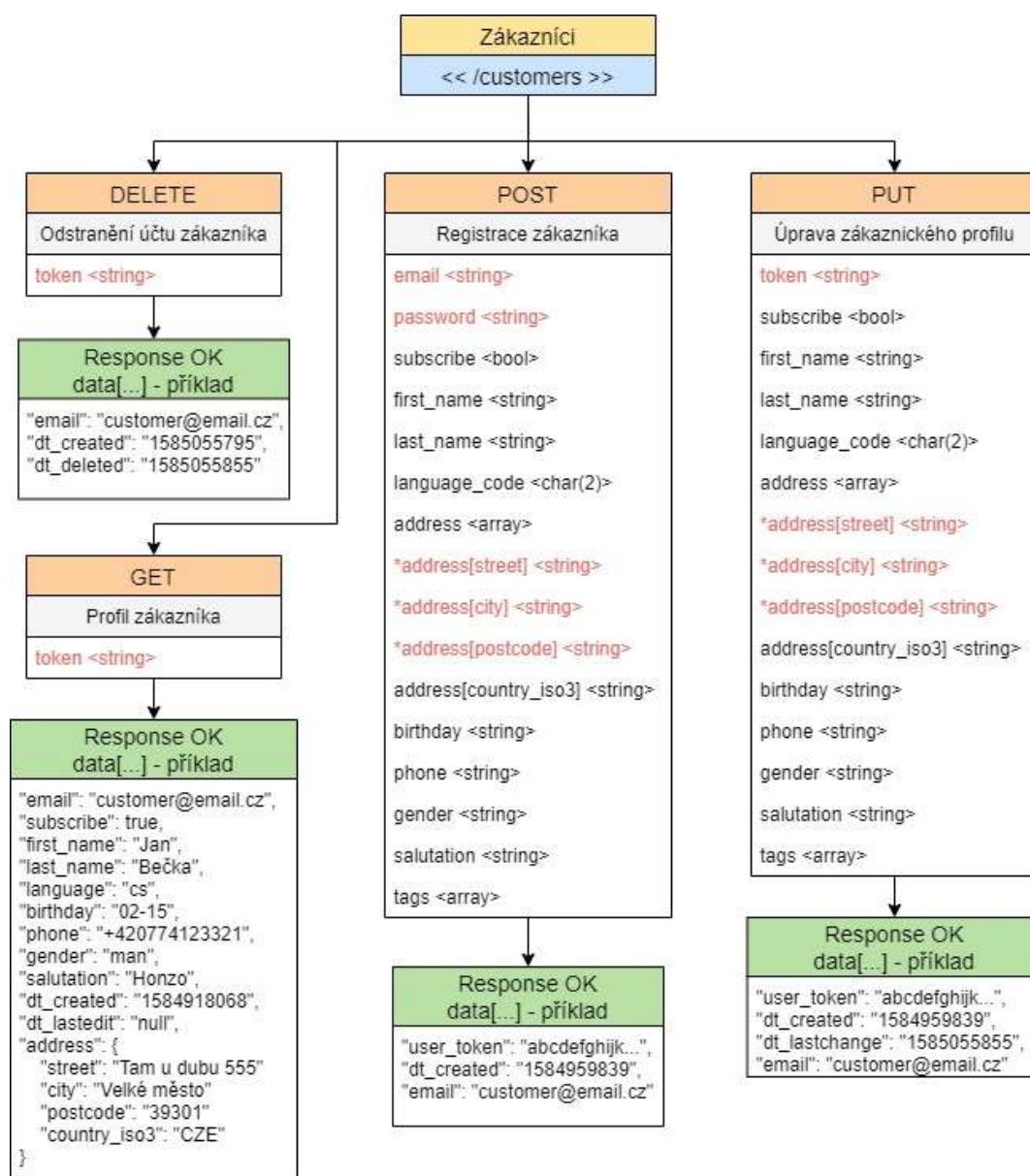
Obrázek 50: Diagram komunikace API a legenda k endpointům (Zdroj: vlastní)

7.2.5.1 Zákazníci (customers)

Endpoint customers poskytuje čtyři různé akce v závislosti na zadané metodě v HTTP protokolu. Metoda POST je určena pro registraci zákazníků do obchodu, PUT slouží

k editaci zákaznických údajů, GET poskytuje dostupné informace o zákazníkovi a DELETE slouží k odstranění zákaznického účtu. Jednotlivé akce endpointu customers jsou znázorněny na obrázku 50. K odstranění, editaci a získání účtu zákazníka je zapotřebí odeslat v požadavku na API parametr 'token', který je zákazníkovi přidělen po přihlášení.

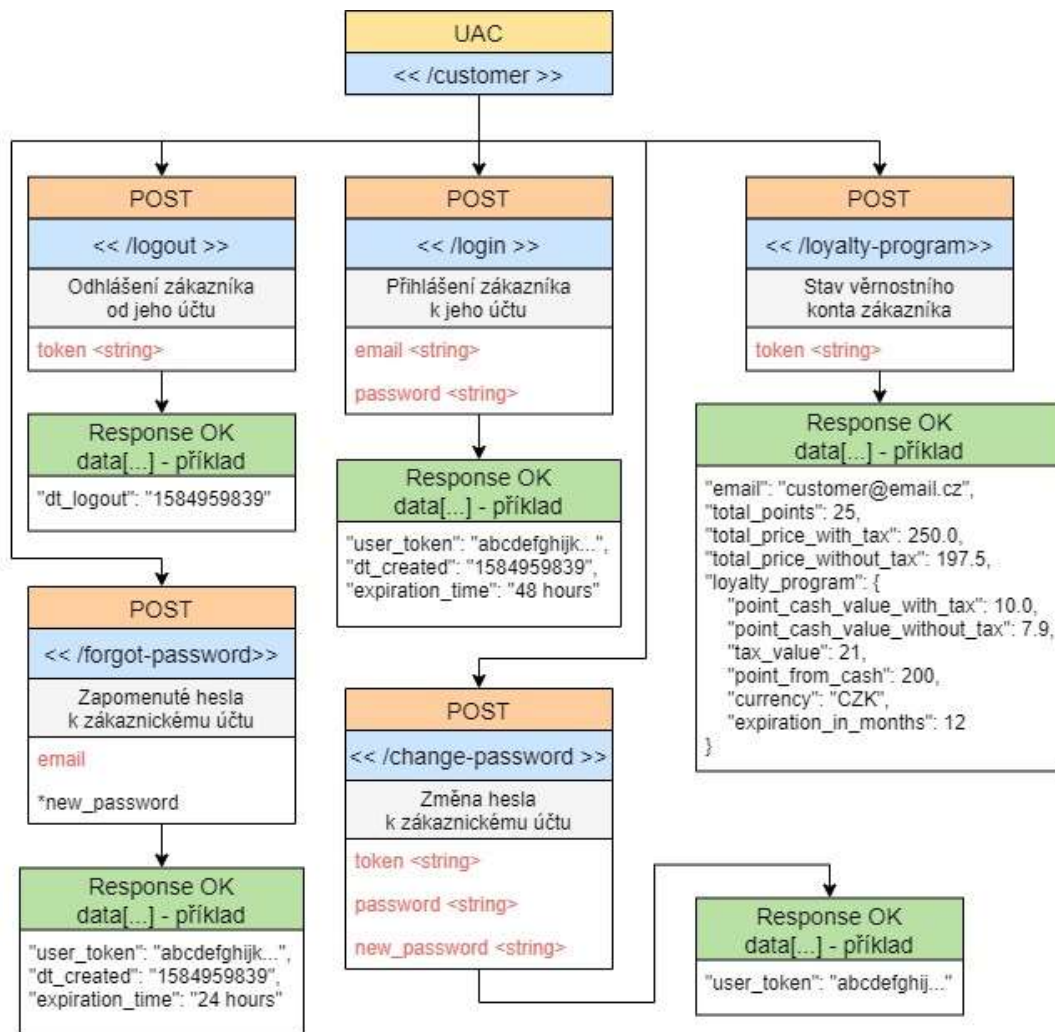
Endpoints s metodou POST, PUT a DELETE jsou napojeny na službu Mailchimp. Při jejich zpracování dochází k distribuci dat prostřednictvím Mailchimp API. Při vyřizování požadavku na registraci zákazníka a odstranění jeho účtu probíhá odesílání potvrzovacích e-mailů přes Mandrill API.



Obrázek 51: Diagram endpointů pro zákazníky (Zdroj: vlastní)

7.2.5.2 UAC (customer)

Kromě správy zákaznických údajů disponuje CRM API také endpoints pro řízení uživatelských účtů (UAC). Všechny endpoints jsou dotazovány metodou POST, ale mají různé URI. Základem je přihlášení zákazníka k jeho účtu, po kterém je vygenerován uživatelský token, který pak slouží k autorizaci zákazníka. Po odhlášení zákazníka je jeho token zneplatněn. Dále jsou zde endpoints na změnu hesla, zapomenuté heslo a zjištění stavu věrnostního konta zákazníka.

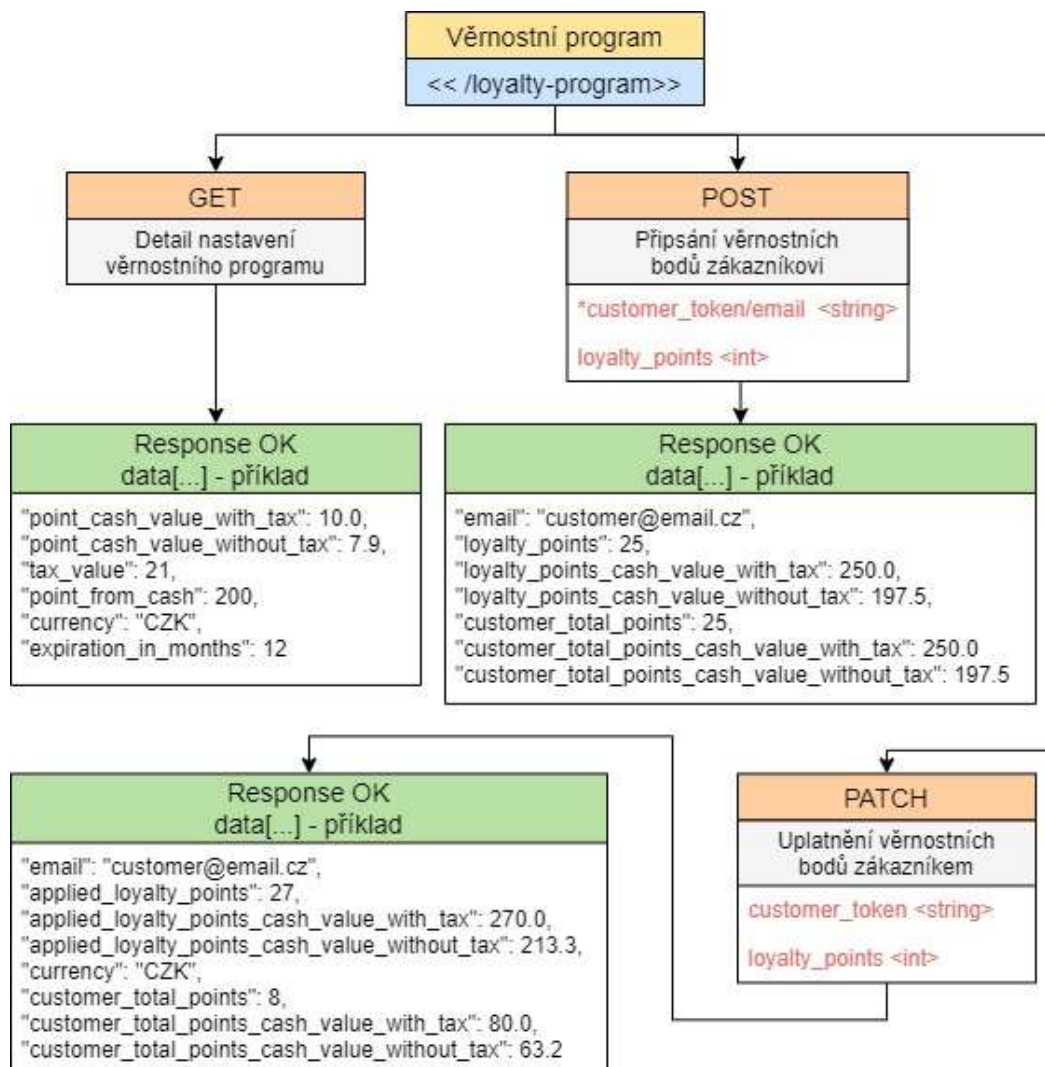


Obrázek 52: Diagram endpointů pro UAC (Zdroj: vlastní)

7.2.5.3 Věrnostní program (LoyaltyProgram)

Endpoint pro věrnostní program obsahuje tři různé akce s metodami POST, PATCH a GET. Akce s metodou POST slouží k připsání věrnostních bodů zákazníkovi buď na

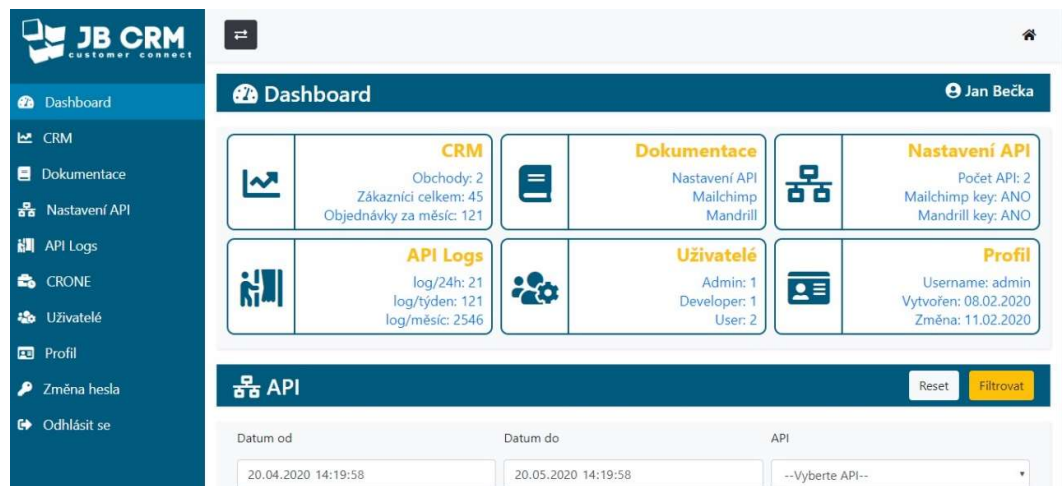
základě uživatelského tokenu, nebo e-mailu. Metoda PATCH naopak slouží k uplatnění věrnostních bodů. Věrnostní body zákazníka mohou být také uplatněny přímo při vytváření objednávky. Nakonec zbývá metoda GET, která vrací informace o tom, jak je věrnostní program nastaven.



Obrázek 53: Diagram endpointů pro věrnostní program (Zdroj: vlastní)

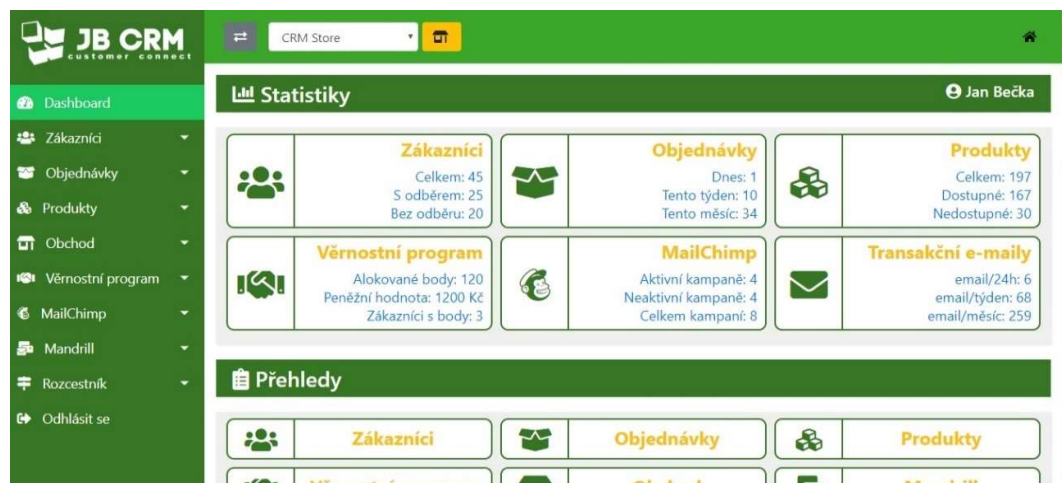
7.3 Webová CRM aplikace

Webová CRM aplikace je rozdělená na tři sekce. Po přihlášení může uživatel mezi těmito sekcemi přecházet bez nutnosti dalšího přihlášení. První sekce obsahuje přihlášení do systému, uživatelskou sekci, nastavení přístupů do CRM API, základní nastavení pro Mailchimp a Mandrill, nastavení napojených obchodů, statistiky požadavků na API, prohlížení logů a správu uživatelských účtů v aplikaci. Ukázka této sekce je k dispozici na obrázku 53.



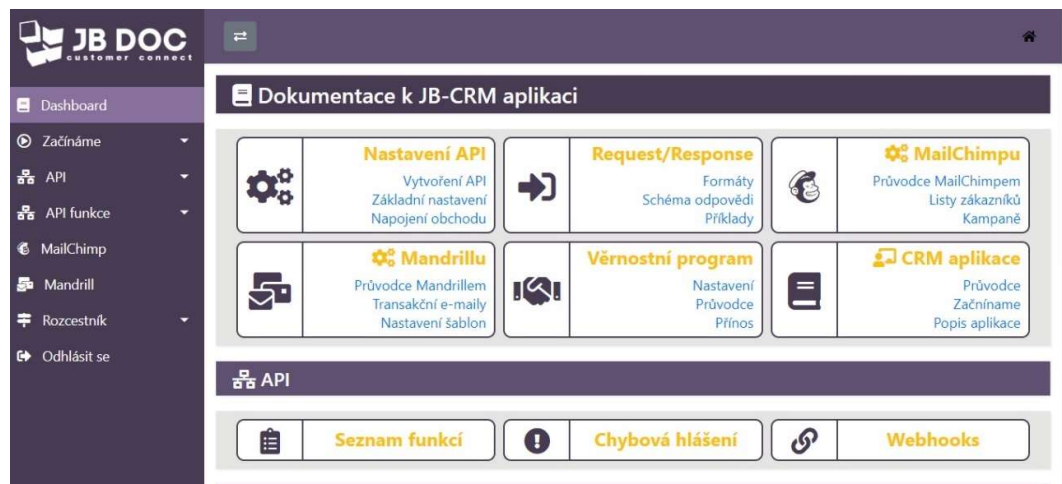
Obrázek 54: Ukázka obrazovky – základní sekce aplikace (Zdroj: vlastní)

Druhá sekce obsahuje již samotnou CRM aplikaci, v ní se vyskytují přehledy a statistiky zákazníků, odběratelů, produktů, objednávek, věrnostního programu a napojených obchodů. Dále zde probíhá nastavení rozesílání transakčních e-mailů a vytváření, nebo přidělování jejich šablon ze služby Mandrill nebo přímo v aplikaci. Dále zde probíhá nastavení spolupráce se službou Mailchimp.



Obrázek 55: Ukázka obrazovky – sekce CRM (Zdroj: vlastní)

Třetí sekce slouží jako dokumentace. Nachází se v ní dokumentace k CRM API včetně průvodce napojením a nastavením internetového obchodu na API. Dále zde můžeme najít návody pro práci s Mailchimpem a Mandrillem. Nakonec jsou zde popsány procesy, které pro internetový obchod vznikají při používání CRM aplikace.



Obrázek 56: Ukázka obrazovky – sekce dokumentace (Zdroj: vlastní)

Všechny snímky obrazovek v této kapitole znázorňují CRM aplikaci tak, jak ji vidí uživatel s rolí admin.

7.4 Integrace služby Mailchimp

Integrace služby Mailchimp do CRM aplikace probíhá prostřednictvím oficiální Mailchimp API. V tomto projektu je použit wrapper pro Mailchimp API, který se nazývá `drewm/mailchimp-api` a je volně k dispozici přes nástroj Composer. Pomocí tohoto wrapperu je možné značně zjednodušit zápis funkcí pro komunikaci s Mailchimpem. Aby komunikace s Mailchimp API mohla probíhat správně, je nutné do v nastavení CRM aplikace zadat platný API key. Při implementaci komunikace se službou Mailchimp jsem se řídil dle dokumentace Mailchimp API.

```
$mcAPI = new MailChimp($this->api_key);  
  
$subscriber_hash = $mcAPI->subscriberHash($customer->email);  
$result = $mcAPI->patch('/lists/' . $customer->store->id_list . '/members/' .  
$subscriber_hash, [  
    'status' => 'subscribe'  
]);
```

Obrázek 57: Ukázka kódu – příklad odeslání požadavku do Mailchimp API, přidání nového odběratele (Zdroj: vlastní)

Všechny požadavky a batch operace (více požadavků odeslaných v jedné dávce) odeslané do Mailchimp API se zaznamenávají do databáze jako logy. Pokud na požadavek nebo batch operaci odpoví Mailchimp API chybovým hlášením, je tato odpověď také zaznamenána v databázi. Odchozí požadavky je následně možné procházet v CRM aplikaci. V případě chybného požadavku je k dispozici výpis chybového hlášení.

```
$mcAPI = new MailChimp($this->api_key);
$mcBatch = $mcAPI->new_batch();

$data = $this->performDataOrderForMc($order);

$mcBatch->post($order->number, 'ecommerce/stores/' . $store->id . '/orders', $data);

$result = $mcBatch->execute();
if (isset($result['id'])) {
    $this->mailchimpBatchesManager->saveBatch($result['id'], 'new_order');
}
```

Obrázek 58: Ukázka kódu – logování odeslaných požadavků do Mailchimp API (Zdroj: vlastní)

Na obrázku 59 je znázorněno vytvoření webhooku v Mailchimu. Webhooky slouží k odchyťování událostí v Mailchimu, které byly provedeny mimo CRM aplikaci. Pokud si uživatel například odhlásí odběr newsletterů z obchodu přes odkaz v patičce e-commerce e-mailu, webhook tuto informaci zachytí. CRM aplikace dále disponuje webhooky pro přihlášení odběru, vytvoření nebo odstranění tagu v listu odběratelů nebo u zákazníka, spuštění integrované e-mailové kampaně a odstranění nebo přidání listu odběratelů.

```
$mcAPI = new MailChimp($this->api_key);

$url = $this->linkGenerator->link('Login:Webhook:catchNewsletter');

$data = [
    'url' => $url,
    'events' => [
        'subscribe' => true,
        'unsubscribe' => true
    ],
    'sources' => [
        'user' => true,
        'admin' => true
    ]
];

$mcAPI->post('lists/' . $id_list . '/webhooks', $data);
```

Obrázek 59: Ukázka kódu – vytvoření webhooku pro odchyťování událostí změn odběrů newsletterů v Mailchimu (Zdroj: vlastní)

Do služby Mailchimp se odesílá značné množství dat, které obdrží CRM aplikace z internetového obchodu. Odesílají se údaje registrovaných zákazníků, provedené

objednávky, nákupní košík zaregistrovaných zákazníků, produkty a odběratelé. Aby mohly být e-commerce data do Mailchimu distribuována, je zapotřebí založit v Mailchimu internetový obchod. Tento proces probíhá automaticky při napojování internetového obchodu na CRM aplikaci.

```
$result = $mcAPI->post('ecommerce/stores', [
    'id' => strval($store->id),
    'list_id' => $store->id_list,
    'name' => $store->name,
    'domain' => $store->domain,
    'is_syncing' => $store->is_syncing,
    'email_address' => $store->email_address,
    'currency_code' => $store->currency_code,
    'money_format' => $store->money_format,
    'primary_locale' => $store->primary_locale,
    'address' => [
        'address1' => $store->address->street,
        'city' => $store->address->city,
        'country_code' => $store->address->country->iso,
        'postal_code' => $store->address->postcode
    ]
]);
```

Obrázek 60: Ukázka kódu – vytvoření obchodu v Mailchimu (Zdroj: vlastní)

7.5 Integrace služby Mandrill

Integrace služby Mandrill do CRM aplikace probíhá přes oficiální API, kterou Mandrill poskytuje společně s dokumentací. Mandrill má k této API vytvořené klienty pro různé programovací jazyky, např. Python, Node.js a PHP. Tato API komunikuje pomocí HTTP protokolu metodou POST a data se odesílají ve formátu JSON. Aby komunikace s Mandrill API mohla probíhat, je nutné do v nastavení CRM aplikace zadat platný API key. Validace klíče probíhá prostřednictvím funkce na následujícím obrázku.

```
public function validateApiKey(string $key): bool
{
    try {
        $mandrill = new Mandrill($key);
        $result = $mandrill->users->ping();
    } catch (\Exception $ex) {
        return false;
    }

    return true;
}
```

Obrázek 61: Ukázka kódu – validace Mandrill API key (Zdroj: vlastní)

Do CRM aplikace se načítají e-mailové šablony z Mandrillu, které se následně přiřadí jednotlivým transakčním e-mailům. Načítání všech šablon z Mandrillu probíhá způsobem znázorněným na obrázku 62.

```

public function getAllTemplatesFromMandrill() : array
{
    $mandrill = new Mandrill($this->api_key);

    $result = $mandrill->templates->getList();

    return $result;
}

```

Obrázek 62: Ukázka kódu – načtení vytvořených šablon v Manrillu (Zdroj: vlastní)

Transakční e-maily jsou připraveny k odesílání ve chvíli, kdy jsou k nim přiřazeny šablony. V aplikaci existuje celá řada transakčních e-mailů, které jsou popsány ve funkčních požadavcích výše. Na následujícím obrázku je ukázka toho, jak vypadá odeslání e-mailu přes Mandrill API. Transakční e-mail v ukázce slouží k upozornění zákazníka na blízkou expiraci jeho bodů z věrnostního programu.

```

public function sendExpireLoyaltyPoint(string $email, int $points, int
$active_points, DateTime $expiration_date): array
{
    $action_slug = 'expiration_loyalty_points';
    $template_slug = $this->getMandrillTemplateSlugByActionSlug($action_slug);

    if ($template_slug != null) {
        $mandrill = new Mandrill($this->settingManager->getMandrillApiKey());

        $message = array(
            'to' => array(
                array(
                    'email' => $email
                )
            ),
            'merge_language' => 'handlebars',
            'merge_vars' => array(
                array(
                    'rcpt' => $email,
                    'vars' => array(
                        array(
                            'name' => 'POINTS',
                            'content' => $points
                        ),
                        array(
                            'name' => 'EXPIRATION_DATE',
                            'content' => date_format($expiration_date, 'd.m.Y H:i:s')
                        ),
                        array(
                            'name' => 'ACTIVE_POINTS',
                            'content' => $active_points
                        )
                    )
                )
            ),
        );

        $result = $mandrill->messages->sendTemplate($template_slug, '', $message);

        return $result;
    }
}

```

Obrázek 63: Ukázka kódu – odeslání e-mailu přes Mandrill API (Zdroj: vlastní)

7.6 Statistické moduly

Tato kapitola se věnuje implementaci statistických modulů, jejichž funkční požadavky jsou popsány v kapitole 3.6.3.8. Data ve statistickém modulu se zobrazují dle potřeby formou tabulek nebo grafů.

7.6.1 Vyhodnocování dat

Vyhodnocování statistik probíhá převážně z dat, které nashromáždila CRM API z internetového obchodu. U každého modulu se vyhodnocují dva typy statistik. Statistiky za určité časové období, jehož rozsah si zvolí uživatel v aplikaci a celkové statistiky.

Inspiraci pro základ vyhodnocování statistických modulů jsem čerpal z knihy, kterou napsal V. Chlebovský v kapitole Měření úrovně a výkonnosti CRM. Toto téma je více rozebráno v kapitole 9. Další statistické údaje vyplynuly z funkčních požadavků na CRM aplikaci. Zbylé statistické údaje jsou získány analytickou metodou Data mining, která slouží k získávání užitečných informací z velkého množství dat.

Vyhodnocení statistik probíhá pomocí dotazů na databázi, které z ní vyberou potřebná data. Pro určité výpočty stačí k vyhodnocení dat správně zvolený SQL dotaz. Takový případ je znázorněn na ukázce 64, kde dotaz zprostředkovává počty dokončených objednávek registrovaných zákazníků, rozdělených podle pohlaví.

```
SELECT new_table.gender, COUNT(*) AS pocet
FROM (
SELECT customers.id, customers.gender
FROM orders
LEFT JOIN orders_customers ON orders_customers.id_order = orders.id
LEFT JOIN customers ON customers.id = orders_customers.id_customer
WHERE orders_customers.id_customer IS NOT NULL
AND orders.status_order = "finished"
AND orders.id_mc_store = 10
GROUP BY orders_customers.id_customer) AS new_table
GROUP BY new_table.gender
```

Obrázek 64: Ukázka kódu – SQL dotaz na databázi (Zdroj: vlastní)

Na druhou stranu s některými daty je nutné ještě dále pracovat, například provést nad daty analýzu, další výpočty nebo připravit data do grafu. To je znázorněno na obrázku 65. Kromě dat, která graf znázorní, se v případě potřeby přidává HTML tooltip, který obsahuje podrobnější informace o datech.


```

public function getStatsNewCustomersChart(int $id_store, FilterCustomers $filter):
array
{
    $result = [];
    $query_string = 'SELECT COUNT(*) AS pocet, DATE_FORMAT(dt_created,"%d.%m") AS
datum, DATE_FORMAT(dt_created,"%d.%m.%Y") AS full_date FROM customers WHERE deleted =
0 AND dt_created >= ? AND dt_created <= ? AND id_mc_store = ? GROUP BY
DATE(dt_created) ORDER BY dt_created ASC ';

    $values = $this->database->query($query_string, $filter->dt_from, $filter->dt_to,
$id_store)->fetchAll();

    foreach ($values as $value) {
        $html = "<div class='chart_tooltip'><div><strong>" . $value['full_date'] .
"</strong></div><div><strong>Počet nových zákazníků: </strong>" . $value['pocet'] .
"</div></div>";

        $result[] = [$value['datum'], $value['pocet'], $html];
    }

    return $result;
}

```

Obrázek 65: Ukázka kódu – příprava dat do grafu (Zdroj: vlastní)

7.6.2 Vykreslování grafů

K vykreslování grafů jsem zvolil nástroj Charts od společnosti Google. Tento nástroj je k dostání zdarma a je pro potřeby CRM aplikace naprosto dostačující, protože obsahuje velké množství typů grafů a způsobů jejich vykreslení. Zápis funkcí, které zajišťují vykreslování grafů, se provádí v JavaScriptu. Veškeré možnosti vykreslování grafů jsou popsány v dokumentaci Google Charts. [51]

V CRM aplikaci jsou použity koláčové, spojnicové, sloupcové, sloupcové skládané a další grafy. Zápis funkce nastavení a následného vykreslení koláčového grafu je k dispozici na obrázku 66. Data se do grafu propisují pomocí Latte.

```

function drawPieChart() {
    var data = new google.visualization.DataTable();
    data.addColumn('string', 'Název API');
    data.addColumn('number', 'Počet přístupů');
    data.addRows({$chartApisLogs});

    var options = {
        height: 300,
        theme: 'maximized',
        is3D: true,
        titleTextStyle: {
            fontSize: 18,
        },
    },
    chartArea: {
        width: '100%',
        height: '80%'
    },
    legend: {
        position: 'right',
        textStyle: {

```

```

        fontSize: 16
    }
}
};

var chart = new
google.visualization.PieChart(document.getElementById('chart_div'));
chart.draw(data, options);
}

```

Obrázek 66: Ukázka kódu – vykreslení koláčového grafu (Zdroj: vlastní)

Na následující obrázku je znázorněna funkce pro vykreslení skládaného sloupcového grafu, který je znázorněn na obrázku 48.

```

function drawChartStack() {
    var data = new google.visualization.DataTable();
    data.addColumn('string', 'Endpoint');
    data.addColumn('number', 'OK');
    data.addColumn({
        type: 'string',
        role: 'tooltip',
        p: {
            'html': true
        }
    });
    data.addColumn('number', 'ERROR');
    data.addColumn({
        type: 'string',
        role: 'tooltip',
        p: {
            'html': true
        }
    });
    data.addRows({$chartApisLogsDividePath});

    var options = {
        height: 400,
        chartArea: {
            width: '90%',
            height: '80%'
        },
        legend: {
            position: 'in'
        },
        tooltip: {
            isHtml: true
        },
        isStacked: true,
    };

    var chart = new
    google.visualization.ColumnChart(document.getElementById('chart_div3'));
    chart.draw(data, options);
}

```

Obrázek 67: Ukázka kódu – vykreslení skládaného sloupcového grafu (Zdroj: vlastní)

7.6.3 Ukázky statistických modulů

Ukázky statistických modulů jsou k dispozici v příložených souborech formou snímků obrazovky.

7.7 Nasazení do produkčního prostředí

Před nasazením do produkčního prostředí je potřeba spustit nástroj GULP, který provede kompilaci a minifikaci kaskádových stylů, JavaScriptu a obrázků. Dále se aplikace nahraje na webový server, kde se nainstalují přes Composer a NPM všechny závislosti a nakonec se aplikace propojí s databází. Po tomto procesu je žádoucí spustit připravené automatické testy, které otestují funkčnost aplikace v produkčním prostředí.

Po zprovoznění CRM aplikace na serveru, je nutné ji nastavit. Průvodce nastavením a používáním CRM aplikace je součástí práce jako webová dokumentace. Dokumentace obsahuje popis napojení internetového obchodu na CRM API a průvodce službami Mailchimp a Mandrill.

8 Testování

Testování kódu by mělo být součástí každého projektu. Tento projekt není výjimkou, protože během implementace CRM aplikace a API se průběžně provádělo testování, ať už automatizované nebo ruční. Nástroje používané pro testování jsou popsány v následující textu.

8.1 CRM aplikace

K testování PHP kódu v CRM aplikaci je použit testovací framework Nette Tester, který je možné nainstalovat do projektu přes Composer. Testování, pokud je to možné, probíhá nad jednou třídou nebo managerem, ve kterém jsou funkce pro zpracování dat a práci s databází. Důvodem je maximalizovat rychlost prováděných testů s ohledem na vlastnosti Nette Testeru. To, že Nette tester provádí testy vysokou rychlostí, je způsobeno tím, že jednotlivé testy spouští paralelně na více vláknech současně. [47]

Díky nástroji Nette tester je psaní testovací kódu příjemnější, než psaní základních testů v PHP. Nette Tester disponuje pluginem do IDE PhpStorm, který práci s testovacím frameworkem ulehčuje. [48]

Při testování se používají takzvané aserce. Aserce jsou predikáty, o kterých je programátor přesvědčen, že jsou na daném místě vždy pravdivé. Příklad aserce si můžeme ukázat na funkci, která zpracovává vstupní data a jako výsledek vrací true nebo false. Pro ilustraci můžeme použít funkci pro validaci klíče z obrázku 61 a jeho testovací aserce na obrázku níže. `Assert::true` testuje, zda funkce vrací true a `Assert::false` kontroluje, zda se vrací false. Predikát u tohoto případu určuje, že platný klíč je pouze u `Assert::true` a ostatní klíče jsou neplatné.

```
Assert::true(
    $mandrillManager->validateApiKey('faba1ffa68a1ds82x8sd93ccfc63173c-us18')
);
Assert::false(
    $mandrillManager->validateApiKey('1234568461354987-us18')
);
Assert::false(
    $mandrillManager->validateApiKey('faba1ffa68a1ds82x8sd93ccfc63173c-us17')
);
```

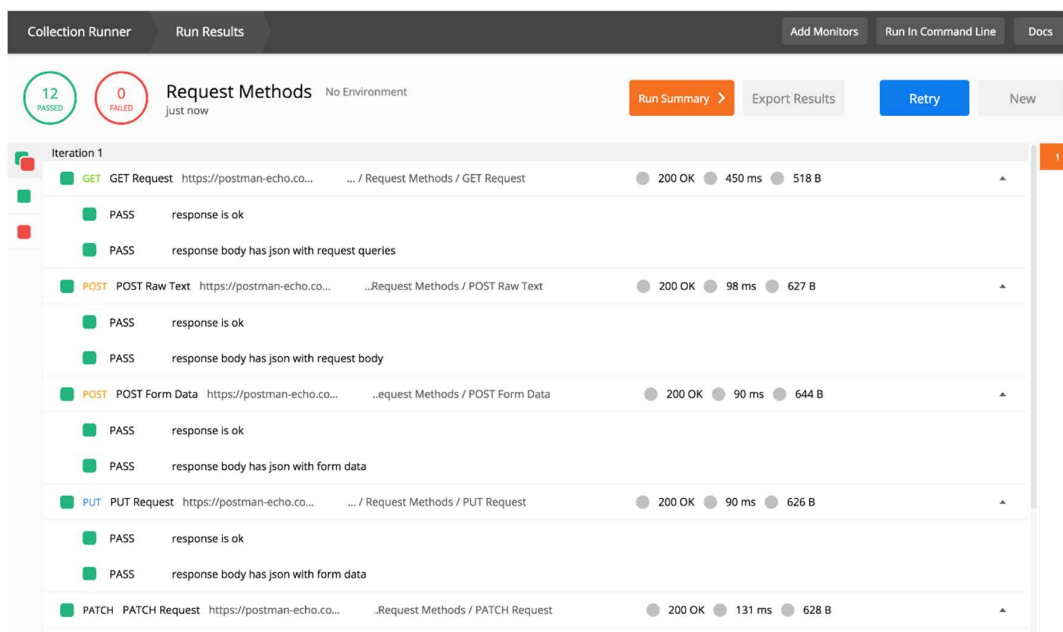
Obrázek 68: Ukázka kódu – Asserty v Nette Tester (Zdroj: vlastní)

Vzhledem k tomu, že asercí má Nette Tester velké množství, je nutné psát vlastní testovací aserce jen zřídka. [48]

8.2 CRM API

Testování kódu CRM API probíhá také s pomocí frameworku Nette Tester. K testování samotné API se používá aplikace Postman. Přes Postman je možné manuálně odesílat požadavky na API a prohlížet odpovědi, které se na ně API vrací. Tato funkce je užitečná hlavně během počáteční fáze implementace API. V pozdější fázi implementace je nutné přejít k automatizovaným testům.

Automatizované testování API probíhá také přes aplikaci Postman. I přesto, že vytvoření automatizovaných testů zabere spoustu času, v budoucnu mohou ušetřit značné množství práce a eliminují možnost průchodu chyby do produkčního řešení.



Obrázek 69: Postman – automatické testování (Zdroj:[49])

Kromě automatizovaných testů umožňuje Postman také monitoring požadavků na API. Ve verzi zdarma má aplikace Postman omezený počet požadavků, testů a monitoringu. Pro potřeby tohoto projektu je verze zdarma dostačující. [34]

9 Měření úrovně a výkonnosti CRM

Měřením úrovně a výkonnosti se dozvíme, jestli je naše CRM řešení úspěšné, anebo jestli je potřeba určité aspekty změnit. Měření je vždy nutné přizpůsobit pro konkrétní podnik, který by ho měl provádět pravidelně.

V. Chlebovský uvádí měření úrovně a výkonnosti CRM zaměřené na pozorování různých veličin, které plynou z oblasti podnikání. V tomto případě je oblastí podnikání provozování internetového obchodu. Veličiny, které Chlebovský doporučuje sledovat, jsou k dispozici v tabulkách, které se nacházejí v příloze. [50]

Měření úrovně a výkonosti CRM aplikace je možné provádět s využitím statistických modulů. Základ statistického modulu tvoří veličiny, které doporučuje Chlebovský. Kromě doporučených veličin obsahují statistické moduly další veličiny, které napomáhají k vyhodnocení výkonnosti CRM aplikace. Statistiky je možné filtrovat podle časového období a porovnávat hodnoty veličin za různá období. Obsah statistických modulů můžeme nalézt v kapitole 7.6.

10 Plánovaný rozvoj aplikace

Do CRM aplikace by bylo vhodné v budoucnu integrovat službu na rozesílání SMS zpráv, které by mohly být využity na upozornění ohledně předání objednávky dopravci nebo ověření zadaného telefonního čísla. Do CRM API by se mohl přidat endpoint pro produkty z internetového obchodu. Nyní se produkty načítají z XML feedu a přidáním endpointu by přibyla další možnost, jak produkty načítat do CRM aplikace.

Další rozvoj aplikace by sloužil k získávání zpětné vazby od zákazníků, například rozesíláním formulářů po dokončení objednávky za účelem zjištění spokojenosti s nákupem nebo rozesíláním formulářů registrovaným uživatelům na zjištění spokojenosti s fungováním obchodu a popřípadě, jakou změnu by uvítali. Jako motivace pro vyplnění dotazníku by mohl sloužit věrnostní program, protože za odeslaný formulář budou zákazníkovi připsány body na věrnostní konto.

11 Závěr

Cílem práce bylo vytvořit CRM aplikaci, jejíž součástí je API pro napojení internetového obchodu a dokumentace k API. Dále má CRM aplikace integrovat služby Mailchimp a Mandrill. Z těchto hledisek bych hodnotil práci jako úspěšnou, protože jejím výsledkem je funkční CRM aplikace, která splňuje stanovené funkční a nefunkční požadavky.

Jediné, co se nepovedlo úplně podle počátečních představ, je vytváření a nastavení e-mailových kampaní pro Mailchimp v CRM aplikaci. Je sice hotové a splňuje funkční požadavky, nicméně umožňuje práci pouze s předpřipravenými kampaněmi v CRM aplikaci. Implementace vytváření vlastních kampaní pro Mailchimp v CRM aplikaci by byla příliš časově náročná a neefektivní.

Zda CRM aplikace poskytuje internetovému obchodu přínos a prohlubuje vztahy se zákazníky se nyní nedá přesně určit, k tomu je zapotřebí používat aplikaci delší časový úsek a hlavně v reálném prostředí. Nicméně API má veškeré předpoklady k tomu, aby byla internetovému obchodu přínosem.

12 Seznam použité literatury

- [1] Česká e-commerce byla v roce 2019 opět rekordní, obraty přesáhly 155 miliard korun. Její podíl na maloobchodu dále roste. *Apek | Asociace pro elektronickou komerci – APEK* [online]. 2020 [cit. 2020-05-22]. Dostupné z: <https://www.apек.cz/clanky/ceska-e-commerce-byla-v-roce-2019-opet-rekordni-o>
- [2] FÁBORSKÝ, Jindřich. V roce 2019 zaniklo 4 300 českých e-shopů. Konsolidace je v plném proudu. *Reshoper* [online]. 2020 [cit. 2020-05-22]. Dostupné z: <https://www.reshoper.cz/datablog/v-roce-2019-zaniklo-4-300-ceskych-e-shopu-konsolidace-je-v-plnem-proudu/>
- [3] Mandrill Brand Assets and Resources Guidelines. *Mandrill* [online]. [cit. 2020-05-22]. Dostupné z: <https://mandrill.com/brand-assets/>
- [4] Mailchimp Brand Assets and Resources Guidelines. *Mailchimp* [online]. [cit. 2020-05-22]. Dostupné z: <https://mailchimp.com/about/brand-assets/>
- [5] ZAMAZALOVÁ, Marcela. *Marketing obchodní firmy*. Praha: Grada, 2009. Manažer. ISBN 978-80-247-2049-4.
- [6] SCHIFFMAN, Leon G. a Leslie Lazar KANUK. *Nákupní chování*. Brno: Computer Press, 2004. Business books (Computer Press). ISBN 80-251-0094-4.
- [7] LEHTINEN, Jarmo. *Aktivní CRM: řízení vztahů se zákazníky*. Praha: Grada, 2007. Expert (Grada). ISBN 978-80-247-1814-9.
- [8] WESSLING, Harry. *Aktivní vztah k zákazníkům pomocí CRM: strategie, praktické příklady a scénáře*. Praha: Grada, 2003. s. 18. ISBN 80-247-0569-9.
- [9] NOVOTNÝ, Jaroslav. *Instrumentarium CRM na trhu výherní techniky*. Ostrava: Key Publishing, 2015. Monografie (Key Publishing). ISBN 978-80-7418-233-4.
- [10] LOŠŤÁKOVÁ, Hana. *Řízení vztahů se zákazníky (CRM) prostřednictvím diferencovaného hodnotového managementu*. Pardubice: Univerzita Pardubice, 2008. ISBN isbn978-80-7395-140-5.
- [11] CRM Software Vtiger - Základní výhody CRM. *CRM Software Vtiger - CRM Software pro malé a střední firmy* [online]. [cit. 2020-05-22]. Dostupné z: <http://www.crm-software.cz/co-je-to-crm-software/zakladni-vyhody-crm/>

- [12] About Mailchimp - Mailchimp's Story. *Mailchimp* [online]. [cit. 2020-05-22]. Dostupné z: <https://mailchimp.com/about/>
- [13] Flexible plans that grow with you. *Mailchimp* [online]. [cit. 2020-05-22]. Dostupné z: <https://mailchimp.com/pricing/>
- [14] MailChimp's Mandrill Move Enrages Email Users. *Covering Digital Customer Experience, the Digital Workplace & Information Management* [online]. [cit. 2020-05-22]. Dostupné z: <https://www.cmswire.com/digital-marketing/mailchimps-mandrill-move-enrages-email-users/>
- [15] What is Mandrill? – Mailchimp Transactional. *Mailchimp Transactional* [online]. [cit. 2020-05-22]. Dostupné z: <https://mandrill.zendesk.com/hc/en-us/articles/360038211854-What-is-Mandrill->
- [16] Mailchimp Transactional Email Pricing |Mandrill| Mailchimp. *Mailchimp* [online]. [cit. 2020-05-22]. Dostupné z: <https://mailchimp.com/pricing/transactional-email>
- [17] What is Scrum? *Scrum.org* [online]. [cit. 2020-05-22]. Dostupné z: <https://www.scrum.org/resources/what-is-scrum>
- [18] The Scrum Framework Poster. *Scrum.org* [online]. [cit. 2020-05-22]. Dostupné z: <https://www.scrum.org/resources/scrum-framework-poster>
- [19] The Scrum Guide. *Scrum Guides* [online]. [cit. 2020-05-22]. Dostupné z: <https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-CS.pdf>
- [20] ette – Pohodlný a bezpečný vývoj webových aplikací v PHP. *Nette – Pohodlný a bezpečný vývoj webových aplikací v PHP* [online]. [cit. 2020-05-22]. Dostupné z: <https://nette.org/cs/>
- [21] The Best PHP Framework for 2015: SitePoint Survey Results — SitePoint. *SitePoint – Learn HTML, CSS, JavaScript, PHP, Ruby & Responsive Design* [online]. [cit. 2020-05-22]. Dostupné z: <https://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/>
- [22] Proč používat Nette? | Nette Docs. *Nette – Pohodlný a bezpečný vývoj webových aplikací v PHP* [online]. [cit. 2020-05-22]. Dostupné z: <https://doc.nette.org/cs/3.0/why-use-nette>

- [23] Průvodce | Latte Templating Engine. *Nette – Pohodlný a bezpečný vývoj webových aplikací v PHP* [online]. [cit. 2020-05-22]. Dostupné z: <https://latte.nette.org/cs/guide>
- [24] About MySQL. *MySQL* [online]. [cit. 2020-05-22]. Dostupné z: <https://www.mysql.com/about/>
- [25] About jQuery. *jQuery Learning Center* [online]. [cit. 2020-05-22]. Dostupné z: <https://learn.jquery.com/about-jquery/>
- [26] CSS Introduction. *W3Schools Online Web Tutorials* [online]. [cit. 2020-05-22]. Dostupné z: https://www.w3schools.com/css/css_intro.asp
- [27] *Getting started | Less.js* [online]. [cit. 2020-05-22]. Dostupné z: <http://lesscss.org/>
- [28] *Gulp.js* [online]. [cit. 2020-05-22]. Dostupné z: <https://gulpjs.com/>
- [29] What is npm. *W3Schools Online Web Tutorials* [online]. [cit. 2020-05-22]. Dostupné z: https://www.w3schools.com/whatis/whatis_npm.asp
- [30] Composer | Nette Docs. *Nette – Pohodlný a bezpečný vývoj webových aplikací v PHP* [online]. [cit. 2020-05-22]. Dostupné z: <https://doc.nette.org/cs/3.0/composer>
- [31] PhpStorm: The Lightning-Smart IDE for PHP Programming by JetBrains. *JetBrains: Developer Tools for Professionals and Teams* [online]. [cit. 2020-05-22]. Dostupné z: <https://www.jetbrains.com/phpstorm/>
- [32] *JetBrains: Developer Tools for Professionals and Teams* [online]. [cit. 2020-05-22]. Dostupné z: <https://www.jetbrains.com/>
- [33] *Postman | The Collaboration Platform for API Development* [online]. [cit. 2020-05-22]. Dostupné z: <https://www.postman.com/>
- [34] Plans & Pricing. *Postman | The Collaboration Platform for API Development* [online]. [cit.2020-05-22]. Dostupné z: <https://www.postman.com/pricing/>
- [35] *Empowering App Development for Developers | Docker* [online]. [cit. 2021-04-14]. Dostupné z: <https://www.docker.com/>
- [36] KOTRBA, Jan. *První seznámení s Docker | B2A* [online]. 2021, [cit. 2021-04-14]. Dostupné z: <https://b2a.cz/blog/seznameni-s-docker/>

- [37] ASP.NET MVC Architecture. *TutorialsTeacher-Learn Web Technologies* [online]. [cit. 2020-05-22]. Dostupné z: <https://www.tutorialsteacher.com/mvc/mvc-architecture>
- [38] Životní cyklus aplikace | Nette Docs. *Nette – Pohodlný a bezpečný vývoj webových aplikací v PHP* [online]. [cit. 2020-05-22]. Dostupné z: <https://doc.nette.org/cs/3.0/request-lifecycle>
- [39] Co umíme - Shoptet.cz. *Internetový obchod a pokladní systém Shoptet.cz* [online]. [cit. 2020-05-22]. Dostupné z: <https://www.shoptet.cz/co-umime/>
- [40] Logo Shoptet. *Internetový obchod a pokladní systém Shoptet.cz* [online]. [cit. 2020-05-22]. Dostupné z: <https://www.shoptet.cz/logo-shoptet/>
- [41] Prohlídka. *SmartEmailing* [online]. [cit. 2020-05-22]. Dostupné z: <https://www.smartemailing.cz/prohlidka/>
- [42] Služby. *SmartEmailing* [online]. [cit. 2020-05-22]. Dostupné z: <https://www.smartemailing.cz/sluzby/>
- [43] Pro média. *SmartEmailing* [online]. [cit. 2020-05-22]. Dostupné z: <https://www.smartemailing.cz/pro-media/>
- [44] *What is REST – Learn to create timeless REST APIs* [online]. [cit. 2020-05-22]. Dostupné z: <https://restfulapi.net/>
- [45] Stopařův průvodce REST API. *Itnetwork.cz - Ajtácká sociální síť a materiálová základna pro C#, Java, PHP, HTML, CSS, JavaScript a další.* [online]. [cit. 2020-05-22]. Dostupné z: <https://www.itnetwork.cz/programovani/nezarazene/stoparuv-pruvodce-rest-api/>
- [46] *Sourcetree | Free Git GUI for Mac and Windows* [online]. [cit. 2020-05-22]. Dostupné z: <https://www.sourcetreeapp.com/>
- [47] Průvodce | Nette Tester. *Nette – Pohodlný a bezpečný vývoj webových aplikací v PHP* [online]. [cit. 2020-05-22]. Dostupné z: <https://tester.nette.org/cs/guide>
- [48] Průvodce | Nette Tester. *Nette – Pohodlný a bezpečný vývoj webových aplikací v PHP* [online]. [cit. 2020-05-22]. Dostupné z: <https://tester.nette.org/>
- [49] Writing tests in Postman. *Postman Blog* [online]. [cit. 2020-05-22]. Dostupné z: <https://blog.postman.com/writing-tests-in-postman/>

[50] CHLEBOVSKÝ, Vít. *CRM: řízení vztahů se zákazníky*. Brno: Computer Press, 2005. Praxe manažera (Computer Press). ISBN 80-251-0798-1.

[51] Charts. *Google Developers* [online]. [cit. 2020-05-22]. Dostupné z: <https://developers.google.com/chart>

13 Seznam obrázků a tabulek

13.1 Obrázky

Obrázek 1: Abstraktní popis řešení.....	10
Obrázek 2: Celková analýza řízení vztahů se zákazníky	12
Obrázek 3: Jednotlivé fáze vztahu	13
Obrázek 4: Technologie řízení vztahů se zákazníky čili CRM	15
Obrázek 5: Logo Shoptet	19
Obrázek 6: Logo SmartEmailing	20
Obrázek 7: Scrum Framework.....	23
Obrázek 8: UC diagram – Uživatelské role a dědičnost uživatelských oprávnění	35
Obrázek 9: UC diagram – Hlavní části systému a uživatelská sekce	36
Obrázek 10: UC diagram – Správa uživatelských účtů v CRM aplikaci.....	37
Obrázek 11: UC diagram – Nastavení obchodu v CRM aplikaci.....	37
Obrázek 12: UC diagram – Nastavení přístupů do CRM API.....	38
Obrázek 13: UC diagram – Přehled logů z API.....	39
Obrázek 14: UC diagram – Přehled obchodu	39
Obrázek 15: UC diagram – Produkty	40
Obrázek 16: UC diagram – Objednávky.....	41
Obrázek 17: UC diagram – Věrnostní program.....	41
Obrázek 18: UC diagram – Zákazníci	42
Obrázek 19: UC diagram – Nastavení transakčních e-mailů.....	42
Obrázek 20: UC diagram – Nastavení Mailchimu.....	43
Obrázek 21: UC diagram – Dokumentace v CRM aplikaci	44
Obrázek 22: Diagram MVC architektury	45

Obrázek 23: Request flow v MVC architektuře	46
Obrázek 24: Ukázka komunikace v REST architektuře	48
Obrázek 25: Ukázka požadavku klienta a odpovědi API v REST architektuře	48
Obrázek 26: BPMN – Proces registrace zákazníka v internetovém obchodě	49
Obrázek 27: BPMN – Proces přidání odběratele bez registrace.....	50
Obrázek 28: BPMN – Proces odstranění zákaznického účtu	50
Obrázek 29: BPMN – Proces vytvoření objednávky včetně věrnostního programu.....	51
Obrázek 30: BPMN – Proces změny stavů objednávky.....	52
Obrázek 31: BPMN – Proces spuštění kampaně pro Mailchimp	53
Obrázek 32: BPMN – Proces zapomenuté heslo k účtu	54
Obrázek 33: ER diagram – věrnostní program	55
Obrázek 34: ER diagram – transakční e-maily	56
Obrázek 35: ER diagram – Mailchimp	57
Obrázek 36: Návrh endpointů v CRM API	58
Obrázek 37: Ukázka IDE prostředí PhpStorm.....	63
Obrázek 38: Ukázka IDE prostředí Postman určeného pro vývoj API	64
Obrázek 39: Ukázka aplikace Sourcetree – GUI pro Git.....	65
Obrázek 40: Ukázka aplikace Docker určené pro Windows	65
Obrázek 41: Git workflow s metodikou SCRUM – znázornění jednoho sprintu.....	66
Obrázek 42: Ukázka kódu – router CRM API.....	67
Obrázek 43: Ukázka kódu – povinné SSL a kontrola HTTP metody.....	68
Obrázek 44: Ukázka kódu – neautorizovaný požadavek.....	68
Obrázek 45: Ukázka obrazovky – nastavení přístupů do API	68
Obrázek 46: Ukázka kódu – funkce pro odesílání odpovědi na požadavky	69

Obrázek 47: Ukázka kódu – uložení záznamu odpovědi do DB	70
Obrázek 48: Ukázka obrazovky – graf úspěšnosti požadavků na API	70
Obrázek 49: Ukázka obrazovky – graf požadavků na endpointy API.....	71
Obrázek 50: Diagram komunikace API a legenda k endpointům	71
Obrázek 51: Diagram endpointů pro zákazníky	72
Obrázek 52: Diagram endpointů pro UAC	73
Obrázek 53: Diagram endpointů pro věnostní program	74
Obrázek 54: Ukázka obrazovky – základní sekce aplikace	75
Obrázek 55: Ukázka obrazovky – sekce CRM.....	75
Obrázek 56: Ukázka obrazovky – sekce dokumentace	76
Obrázek 57: Ukázka kódu – příklad odeslání požadavku do Mailchimp API, přidání nového odběratele	76
Obrázek 58: Ukázka kódu – logování odeslaných požadavků do Mailchimp API	77
Obrázek 59: Ukázka kódu – vytvoření webhooku pro odchyťování událostí změn odběrů newsletterů v Mailchimu	77
Obrázek 60: Ukázka kódu – vytvoření obchodu v Mailchimu	78
Obrázek 61: Ukázka kódu – validace Mandrill API key	78
Obrázek 62: Ukázka kódu – načtení vytvořených šablon v Manrillu.....	79
Obrázek 63: Ukázka kódu – odeslání e-mailu přes Mandrill API.....	79
Obrázek 64: Ukázka kódu – SQL dotaz na databázi	80
Obrázek 65: Ukázka kódu – příprava dat do grafu	81
Obrázek 66: Ukázka kódu – vykreslení koláčového grafu	82
Obrázek 67: Ukázka kódu – vykreslení skládaného sloupcového grafu	82
Obrázek 68: Ukázka kódu – Asserty v Nette Tester.....	84
Obrázek 69: Postman – automatické testování	85

13.2 Tabulky

Tabulka 1: Porovnávání oblastí interakce z pohledu zákazníka a podniku 11

Tabulka 2: Stavové kódy odpovědí 48

14 Seznam použitých zkratek

AJAX – Asynchronous JavaScript and XML

APEK – Asociace pro elektronickou komerci

API – Application Programming Interface

Atd. – a tak dále

B2B – Business To Business

BPMN – Business Process Model and Notation

CCM – Customer Creation Management

CMS – Content Management System

CRM – Customer Relationship Management

CRUD – Create, Read, Update, Delete

CSRF – Cross-site Request Forgery

CSS – Cascading Style Sheets

CTM – Customer Termination Management

DB – Database

DPH – Daň z přidané hodnoty

ER diagram – Entity Relationship diagram

GUI – Graphical User Interface

HTML – Hypertext Markup Language

HTTP – Hypertext Transfer Protocol

HW – Hardware

IDE – Integrated Development Environment

JSON – JavaScript Object Notation

Kč – Koruna česká

LESS – Leaner Style Sheets

MVC – Model-View-Controller

NPM – Node Package Manager

OOP – Object-oriented programming

OOPSLA – Object-Oriented Programming, Systems, Languages & Applications

PHP – Hypertext preprocessor

Popř. – popřípadě

PPC – Pay Par Click

REST – Representational State Transfer

RSS – Really Simple Syndication

SASS – Syntactically Awesome Style Sheets

SEO – Search Engine Optimization

SMS – Short message service

SMTP – Simple Mail Transfer Protocol

SOAP – Simple Object Access Protocol

SQL – Structured Query Language

SSL – Secure Sockets Layer

SSO – Single sign-on

SW – Software

Tzv. – tak zvaně

UAC – User Account Control

UC diagram – Use case diagram

UML – Unified Modeling Language

URI – Uniform Resource Identifier

URL – Uniform Resource Locator

USA – United States of America

XML – Extensible Markup Language

XSS – Cross-site scripting

15 Přílohy

15.1 Obsah přiložených souborů

Složka Diagrams:

- Use case diagramy
- BPMN diagramy procesů
- ER diagram databáze
- Diagram endpointů API

Složka Screenshots:

- Screenshoty z CRM aplikace

Složka Project:

- Projekt webová CRM aplikace včetně API a dokumentace
- Databáze s výchozími daty
- README k spuštění projektu

15.2 Tabulky měření úrovně a výkonnosti CRM [50]

Oblast	Veličina (označení)	Popis	Vzorec výpočtu	Hodnota (rozsah)	Jednotka	Optimální rozsah hodnot	Procentuální přepočít k přípustnému rozsahu	Váha	Jak často se sleduje
+	povědomí (BAW)	Jak je značka vnímána zákazníky a veřejností?	$(\text{značka znají/všichni oslovení}) * 100$	0–100	%	65+ (maximalizace)	hodnota		rok
	kvalita (BQL)	Jak je kvalita značky vnímána zákazníky?	$(\text{vysoká kvalita/všichni oslovení}) * 100$ $(\text{střední kvalita/všichni oslovení}) * 100$ $(\text{nízká kvalita/všichni oslovení}) * 100$	0–100	%	60+ (maximalizace) 15- (minimalizace)	hodnota		rok
	(ne)loajalita (BLO)	Kolik procent zákazníků přešlo v daném časovém období ke konkurenci?	$(\text{přešli ke konkurenci/všichni zákazníci}) * 100$	0–100	%	3- (minimalizace)	hodnota		rok
Marketing	2 nabídky	Jak je nabízená hodnota vnímána zákazníky?	$(\text{vysoká hodnota/všichni oslovení}) * 100$ $(\text{střední hodnota/všichni oslovení}) * 100$ $(\text{nízká hodnota/všichni oslovení}) * 100$	0–100	%	60+ (maximalizace) 15- (minimalizace)	hodnota		rok
	spokojenost (VSA)	Jak jsou zákazníci s nabídkou spokojeni?	$(\text{velmi spokojeni/všichni oslovení}) * 100$ $(\text{středně spokojeni/všichni oslovení}) * 100$ $(\text{nespokojeni/všichni oslovení}) * 100$	0–100	%	65+ (maximalizace) 8- (minimalizace)	hodnota		rok
Marketing	dosah (CRE)	Jaké procento z oslovených zákazníků kampaň zaregistrovalo?	$(\text{zaregistrovali/všichni oslovení}) * 100$	0–100	%	40+ (maximalizace)	hodnota		kampaň
Kampaň	reakční faktor (ORR)	Jaké procento z oslovených zákazníků na kampaň zareagovalo?	$(\text{zareagovali/všichni oslovení}) * 100$	0–20	%	5+ (maximalizace)	hodnota * 100 ----- 20		kampaň
	konverze (CCR)	Jaké procento z oslovených zákazníků na základě kampaně nakoupilo?	$(\text{nakoupili/všichni oslovení}) * 100$	0–20	%	2+ (maximalizace)	hodnota * 100 ----- 20		kampaň
Marketing	náklady (CCO)	Jaké jsou náklady na kampaň vztážené na její dosah, reakční faktor a konverzi?	$\text{náklady na kampaň/všichni oslovení} * \text{CRE}/100$ $\text{náklady na kampaň/všichni oslovení} * \text{ORR}/100$ $\text{náklady na kampaň/všichni oslovení} * \text{CCR}/100$	závisí na typu a rozsahu kampaně	-	minimalizace	$(\text{max-min-h}) * 100$ ----- max-min		kampaň
	počet návštěvníků (IVC)	Kolik uživatelů internetu navštíví v daném časovém intervalu firemní prezentaci?	-	závisí na typu a struktuře	ks	maximalizace	hodnota * 100 ----- max-min		čtvrtlet
Marketing	počet návštěvníků (IUV)	Jaké procento z návštěvníků sleduje firemní prezentaci poprvé?	$(\text{první návštěvníci}/\text{IVC}) * 100$	0–100	%	maximalizace	hodnota ----- max-min		čtvrtlet

Oblast	Velikost popis	Velikost - označení	Vzorec výpočtu	Hodnota (rozsah)	Jednotka	Optimální rozsah hodnot	Procentuální poměr k teoretické max. hodnotě	Váha	Jak často se sleduje
	oblíbené str. (IPH)	Které části prezentace jsou nejoblíbenější?							čtvrtletí
	průměrná doba prohlížení (IDU)	Jak dlouho průměrně stráví návštěvník prohlížením firemní prezentace?	celková doba prohlížení/IVC	závisí na typu a struktuře	min/ks	N/A	$\frac{\text{hodnota} * 100}{ \text{max} - \text{min} }$		čtvrtletí
	registrovaní uživatelé (IRU)	Jaké procento návštěvníků se na stránkách zaregistrovalo?	(registrovaní/IVC) * 100	0-100	%	7+ maximalizovat	$\frac{\text{hodnota} * 100}{ \text{max} - \text{min} }$		čtvrtletí
	procento přeušení (IBR)	Jaké procento návštěvníků předčasně opustilo prezentaci?	(opustili prezentaci/IVC) * 100	0-100	%	12- minimalizovat	$\frac{(\text{imax-min}-t) * 100}{ \text{max} - \text{min} }$		čtvrtletí
Sales - zákazníci	náklady na nové zákazníky (SAC)	Jaké jsou průměrné náklady na získání nového zákazníka?	náklady na interakci/počet nových zákazníků	závisí na typu nabízeného produktu a oslovené zákaznické skupině	USD	minimalizovat	$\frac{(\text{imax-min}-t) * 100}{ \text{max} - \text{min} }$		rok/čtvrtletí
	pravděpodobnost úspěchu (SCP)	Jaká je úspěšnost při jednání se zákazníky?	(úspěšné zakončená jednání/všechna jednání) * 100	0-100	%	15+ (maximalizace)	hodnota		rok
	velikost objednávky (SAO)	Jaká je průměrná velikost objednávky?	tržby/počet objednávek	závisí na typu produktu	USD	N/A	N/A		rok
	podíl z rozpočtu (SSW)	Jaký je průměrný podíl firmy na dodávkách zákazníků v porovnání s konkurencí?	(obrat firmy/obrat produktu)/počet zákazníků * 100	0-100	%	30+ (maximalizace)	hodnota		rok/čtvrtletí
	zisk na zákazníka (CUP)	Jaký je hrubý zisk vztahovaný k jednomu zákazníkovi?	hrubý zisk/počet zákazníků	> 0	USD	maximalizovat	$\frac{\text{hodnota} * 100}{ \text{max} - \text{min} }$		čtvrtletí
	tržby na zákazníka (CUS)	Jaké jsou průměrné tržby na jednoho zákazníka?	tržby/počet zákazníků	závisí na typu podnikání		maximalizovat	$\frac{\text{hodnota} * 100}{ \text{max} - \text{min} }$		čtvrtletí

Oblast	Veličina – popis	Veličina – označení	Vzorec výpočtu	Hodnota	Jednotka	Optimální rozsah hodnot	Procentuální poměr k teoretické max. hodnotě	Váha	Jak často se sleduje
	obrat (TOI)	Kolikrát se zásoby přemění v ostatní formy oběžného majetku?	tržby/zásoby	závisí na typu podnikání; > 5	-	maximalizovat	hodnota * 100 max - min		čtvrtlet
	obrat pohledávek (TOR)	Jak rychle jsou pohledávky přeměněny v peněžní prostředky?	tržby/pohledávky	> 4	-	maximalizovat	hodnota * 100 max - min		čtvrtlet
	top zákazníci (TOC)	Jaké procento zákazníků je ve skupině nejvyššího řádu tržeb?	(počet zákazníků v nejvyšším řádu obrátu/všichni zákazníci) * 100	0-100	%	N/A	hodnota * 100 max - min		čtvrtlet
	malí zákazníci (LEC)	Jaké procento zákazníků nedosahuje ani desátiny tržeb nejvyšší skupiny?	(- počet zákazníků s obrátem < 1/10 nejvyšší skupiny/všichni zákazníci) * 100	0-100	%	minimalizovat	(max-min)*100 max - min		čtvrtlet