



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Systém pro podporu přímého prodeje
Student:	Tereza Živnůstková
Vedoucí:	Ing. Eliška Šestáková
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2018/19

Pokyny pro vypracování

Cílem práce je návrh a implementace systému pro podporu přímého prodeje pro podnikatele. Funkčnost požadované aplikace lze rozdělit do následujících čtyřech oblastí:

- ukládání informací o zákaznících
- evidence objednávek a statistiky prodeje
- evidence počtu zboží na skladě
- plánování telefonátů a schůzek

V první fázi proveďte detailní analýzu požadavků ve výše uvedených oblastech. Dále se seznamte s existujícími nástroji pro podporu přímého prodeje či aplikacemi zaměřenými na jednotlivé oblasti. Následně navrhnete architekturu a GUI požadované aplikace. V rámci návrhu systému zohledněte specifické požadavky budoucích zákazníků.

Prototyp systému implementujte jako desktopovou aplikaci v jazyce Java a vytvořte dokumentaci. Bude-li to vhodné, využijte při implementaci existujících komponent a knihoven. Aplikaci otestujte s budoucími uživateli a vyvoďte případné změny (ať už v rámci této práce či do budoucna).

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

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

V Praze dne 4. prosince 2017



**FAKULTA
INFORMAČNÍCH
TECHNOLGIÍ
ČVUT V PRAZE**

Bakalářská práce

System pro podporu přímého prodeje

Tereza Živňůstková

Katedra softwarového inženýrství
Vedoucí práce: Ing. Eliška Šestáková

15. května 2018

Poděkování

Tímto bych ráda upřímně poděkovala své vedoucí, Ing. Elišce Šestákové, za její ochotu, pozitivní přístup a obětovaný čas, který mi věnovala během psaní této bakalářské práce. Děkuji také své rodině a přátelům za podporu během celého mého studia.

Prohlášení

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

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

V Praze dne 15. května 2018

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2018 Tereza Živnůstková. Všechna práva vyhrazena.

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

Odkaz na tuto práci

Živnůstková, Tereza. *Systém pro podporu přímého prodeje*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.

Abstrakt

Tato práce se zaměřuje na tvorbu podpůrného systému pro přímé prodejce. Jedná se o systém, ve kterém si prodejci mohou uchovávat informace o zákaznících, objednávkách, schůzkách, apod. Součástí práce je analýza existujících řešení a analýza požadavků, na jejichž základě byl navrhnout a implementován prototyp výše zmíněného systému. Prototyp byl vytvořen v jazyce Java a JavaFX jako desktopová aplikace.

Klíčová slova přímý prodej, desktopová aplikace, CRM, podnikatel

Abstract

This thesis deals with creation of the Direct Sales System (DSS). The main purpose of DSS is to store information about customers, orders, meetings, etc. First part of the thesis contains analysis of existing systems and future requirements. The second part deals with design and implementation of a functional DSS prototype. The prototype was implemented as a desktop application in Java and JavaFX.

Keywords direct sales, desktop application, CRM, enterpriser

Obsah

Úvod	1
Přímý prodej	1
Cíl práce	2
Využití	2
Struktura práce	3
1 Analýza	5
1.1 CRM systém	5
1.2 Analýza uživatelských požadavků	5
1.3 Případy užití	8
1.4 GDPR	13
1.5 Analýza existujících CRM systémů	14
2 Návrh a implementace prototypu	21
2.1 Zvolené technologie	21
2.2 Návrh architektury systému	21
2.3 Způsob uložení dat	24
2.4 Návrh grafického rozhraní	28
2.5 Návrh řešení GDPR	34
2.6 Zhodnocení prototypu	34
3 Testování	35
3.1 Průběžné testování	35
3.2 Uživatelské testování	35
3.3 Změny plynoucí z provedení testování	36
4 Dokumentace	37
4.1 První přihlášení	37
4.2 Nastavení aplikace	37
4.3 Domovská stránka	38

4.4	Zákazníci	38
4.5	Objednávky	38
4.6	Sklad	39
Závěr		41
Literatura		43
A Testovací scénáře		45
A.1	Vytvoření účtu	45
A.2	Přihlášení	45
A.3	Domovská stránka	46
A.4	Zákazníci	46
A.5	Objednávky – objednávky zákazníků	47
A.6	Objednávky – objednávky k vyřízení	48
A.7	Sklad	48
B Seznam použitých zkratk		51
C Obsah příloženého CD		53

Seznam obrázků

1.1	Use case diagram	9
1.2	Ukázka prostředí systému InTouch CRM	15
1.3	Ukázka prostředí systému BPM online	16
1.4	Ukázka prostředí systému FollowFox	17
1.5	Ukázka prostředí systému 1CRM	18
2.1	Diagram balíčků	22
2.2	Ukázka volání metody <code>getAllProducts()</code>	23
2.3	Ukázka implementace metody <code>getAllProducts()</code>	23
2.4	Ukázka konstant na volání SQL dotazů	23
2.5	Ukázka implementace metody <code>getAllProducts(ResultSet)</code>	24
2.6	Struktura databáze	25
2.7	Databáze k přihlašování	27
2.8	Databáze k nastavení aplikace	27
2.9	Návrh obrazovky <i>domů</i>	30
2.10	Návrh obrazovky <i>zákazníci</i>	30
2.11	Návrh obrazovky <i>zákazníci – profil</i>	31
2.12	Návrh obrazovky <i>objednávka</i>	31
2.13	Návrh obrazovky <i>objednávka – nová</i>	32
2.14	Návrh obrazovky <i>sklad</i>	32
2.15	Návrh obrazovky <i>kalendář</i>	33
2.16	Návrh obrazovky <i>statistiky</i>	33

Seznam tabulek

1.1	Scénář k vyřízení objednávky zákazníka	11
1.2	Scénář k vytvoření události	12
1.3	Mapování případů užití na požadavky	13
1.4	Tabulka výhod a nevýhod systému InTouch CRM	15
1.5	Tabulka výhod a nevýhod systému BPM Online	16
1.6	Tabulka výhod a nevýhod systému FollowFox	17
1.7	Tabulka výhod a nevýhod systému 1CRM	18

Úvod

Být přímým prodejcem je záležitost, která vyžaduje uchovávání velkého množství dat a poctivou organizaci času. Proto je k tomu vhodné mít určitého pomocníka. Někdo volí klasické papíry a diář, pro většinu je však pohodlnější mít informační systém, ve kterém je vše. Výsledek této práce je tedy určen pro podporu právě přímého prodeje.

Téma jsem si zvolila, neboť jsem měla tu možnost si přímý prodej vyzkoušet a narazila jsem na jeho úskalí. V neposlední řadě pro mě byla motivace to, že jsem doposud na takový informační systém nenarazila. V rámci této práce se proto zabývám analýzou, návrhem a implementací prototypu aplikace tohoto typu. Prototyp je navržen jako desktopová lokální aplikace a rozhodla jsem se ji poskytnout jako open source.

Přímý prodej

Přímý prodej je v dnešní době stále poměrně rozšířené odvětví podnikání. Nejvíce populární je v oblasti zdraví a kosmetiky (například společnost Mary Kay nebo Nu Skin). Společnost touto formou nabízí své zboží zákazníkům prostřednictvím jednotlivých obchodníků (přímých prodejců). Ti ovšem nejsou přímo jejich zaměstnanci. Mezi společností a prodejcem je uzavřená smlouva, na základě které prodejce jako podnikatel odkupuje od společnosti zboží. Vlastní prodej poté probíhá tak, že si prodejce se zákazníkem domluví schůzku, kde mu zboží představí a zákazník se následně rozhodne, zda si zboží od prodejce koupí nebo ne.

Úplným základem, bez kterého se prodejce neobejde, jsou kontakty. Databázi kontaktů si musí každý postupně vybudovat. Když už je koho oslovit, tak následuje pozvánka na osobní schůzku. Schůzky probíhají většinou po 1 až 4 lidech a v závislosti na konkrétní společnosti trvají zhruba 30 až 60 minut, někdy i déle. Během této schůzky musí prodejce své zákazníky co nejvíce poznat, určit co je pro ně vhodné a produkt jim následně ukázat, aby si ho

mohli vyzkoušet. Jeho cílem je samozřejmě zboží prodat a navíc ještě získat co nejvíce dalších kontaktů (doporučení).

Jedny z největších výhod přímého prodeje jsou osobní přístup ke klientům a zákaznický servis – kontrola, zda jsou zákazníci se zbožím spokojeni a zda nepotřebují doobjednat další.

Cíl práce

Cílem této bakalářské práce je navrhnout a implementovat prototyp aplikace, která bude usnadňovat práci výše zmíněným přímým prodejcům. Prototyp bude vytvořen jako open source desktopová aplikace v jazyce Java a JavaFX.

Hlavní výhodou výsledné aplikace by měla být co největší možnost přizpůsobení, aby ji šlo využít v jakémkoliv odvětví přímého prodeje. Každý prodejce navíc dělá svou práci trochu jiným způsobem.

Funkce aplikace lze rozdělit na následujících čtyř oblastí:

1. Evidence zákazníků a možnost u nich ukládat jakékoliv informace. U každého zákazníka by měla být vedená historie schůzek, korespondencí, objednávek apod.
2. Vedení historie objednávek a statistik prodeje. Systém by měl umožňovat ukládání jak objednávek zboží k naskladnění, tak jednotlivé objednávky zákazníků.
3. Evidence počtu kusů zboží na skladě.
4. Plánování úkolů, schůzek a korespondence formou kalendáře.

Využití

Cílovou skupinou této aplikace jsou přímí prodejci. Ti si domlouvají schůzky se svými zákazníky, o nichž si ukládají různé informace, zaznamenávají objednané zboží, uskutečněné schůzky, hovory, apod.

Systém by měl zjednodušit celý proces prodeje. Od získávání kontaktů po následnou péči o zákazníky. Díky tomuto systému budou mít prodejci všechny informace na jednom místě. Přes evidenci zákazníků, objednávek a statistik prodeje až po kalendář.

Aplikace by měla být navržena tak, aby si ji každý mohl jednoduše upravit dle svých představ. Zároveň by ale měla zůstat stále intuitivní a jednoduchá. Díky tomu ji budou moct využívat prodejci z nejrůznějších odvětví. Každý si navíc ukládá o zákaznících trochu jiné informace nebo nevyužije všechny části, které bude systém nabízet.

Struktura práce

Tato práce dále pokračuje v následující struktuře. V kapitole 1 provádím analýzu existujících řešení a detailní analýzu požadavků systému ve všech jeho oblastech, na základě kterých vytvářím v rámci kapitoly 2 (kapitola obsahující návrh a popis implementace) návrh aplikace. Sekce návrhu se skládá z návržení grafického rozhraní aplikace a následujících diagramů: případy užití, databázový model a diagram balíčků. Poté následuje implementace systému a jeho otestování, viz kapitola 3. V poslední části práce (kapitola 4) se věnuji dokumentaci detailně popisující funkce implementovaného prototypu.

Analýza

Systém, který navrhuji a implementuji v rámci této práce, je ve své podstatě CRM systém. Proto se v následující sekci věnuji vysvětlení tohoto pojmu.

1.1 CRM systém

CRM systém je pro většinu firem a podnikatelů nezbytná součást k jejich fungování vůbec. Umožňuje lépe třídit důležité informace o zákaznících a zároveň efektivněji vyhodnocovat obchodní aktivity společnosti. Zkratka CRM vychází z anglického názvu *Customer Relationship Management* – tedy v překladu řízení vztahů se zákazníky. Tento typ softwaru je tedy aplikace, díky které můžeme přehledně shromažďovat veškeré potřebné údaje o zákaznících, jako jsou například kontaktní informace, celkové obraty, podíly na obratech v určitém segmentu apod.

Tyto informace pomáhají firmám lépe porozumět zákazníkům a jejich potřebám a jednotlivé obchodní procesy těmto potřebám přizpůsobit. [1]

1.2 Analýza uživatelských požadavků

Rozbor požadavků je důležitá část analýzy. Má-li být výsledný systém dle očekávání, je potřeba přesně pochopit, jak se bude ideálně používat. V této práci tuto část řeším stylem funkčních a nefunkčních požadavků.

Požadavky jsem dala dohromady společně s lidmi z oboru. Díky tomu, že jsem v tomto typu podnikání byla sama činná, mám však poměrně dobrou představu o tom, jak by výsledný systém měl fungovat.

1.2.1 Funkční požadavky

Funkční požadavky se používají k upřesnění chování aplikace. [2] Uvedla jsem je postupně podle funkcí aplikace. Nejdříve však začínám těmi obecnými.

- F1: Přihlašování a účty** Vstup do aplikace bude podmíněn přihlášením. Při prvním spuštění aplikace si tedy bude muset uživatel nejprve nastavit e-mail a heslo, přes které se bude uživatel přihlašovat. Při ztrátě hesla se na pokyn uživatele zašle na jeho e-mail nové.
- F2: Obecné nastavení** Obecné nastavení bude obsahovat vypínání konkrétních modulů, zálohu dat a nastavení související s účtem.
- F3: Evidence zákazníků** V aplikaci bude veden seznam všech zákazníků prodejce. Informace, které se o něm budou ukládat, budou editovatelné.
- F4: Historie schůzek, korespondencí a objednávek** Kvůli přehlednosti o daném zákazníkovi budou v jeho profilu zobrazené seznamy schůzek, korespondencí, objednávek a upomínek. Půjdou navíc v rámci profilu zákazníka i vytvořit. Jednotlivé záznamy budou navíc barevně rozlišené podle stavu (provedené od neprovedených, apod.).
- F5: Evidence vztahů mezi zákazníky** Při větším množství kontaktů je těžké si o jejich vzájemných vztazích udržet přehled. Díky této funkci budou mít prodejci představu např. o rodině nebo přátelích zákazníka. Což se hodí pro zlepšení osobních vztahů s klienty.
- F6: Automatické upomínky** Systém bude automaticky připomínat např. narozeniny zákazníka. V nastavení půjde také zapnout upomínky na zavolání po zakoupení produktu (např. kvůli spokojenosti). Doba, po které se upomínka vytvoří, bude editovatelná. Upomínky se budou zobrazovat v kalendáři a půjde u nich naplánovat korespondence. Lze přidávat i své vlastní upomínky.
- F7: Nastavení stránky o zákaznících** V tomto nastavení půjdou zapnout, vypnout a nastavit jednotlivé funkce profilu – upomínky, vztahy mezi zákazníky, historie schůzek, korespondencí a objednávek. Dále zde půjde editovat sekci *informace o zákazníkovi* (tzn. přidávat nebo odebírat pole s informacemi, které si o zákaznících chceme ukládat).
- F8: Naskladnění** Uživatel si bude moct zadávat veškeré objednávky sloužící k naskladnění zboží. Podle toho se bude upravovat počet kusů produktů na skladě.
- F9: Objednávky od zákazníků** Při přidávání produktů do objednávky zákazníka bude systém u jednotlivých produktů nabízet možnost přidání produktu do předpřipravené objednávky k naskladnění (i pokud daný produkt bude na skladě).
- F10: Předpřipravené objednávky** V aplikaci bude možnost tzv. předpřipravené objednávky. To znamená, že v seznamu objednávek od dodavatele bude jedna *předpřipravená objednávka*, ve které si uživatel bude

připravovat produkty, které bude objednávat. Po objednání ji jen označí za hotovou. Zároveň se do ní budou přidávat automaticky produkty, u kterých se při objednávce zákazníka zvolí způsob *objednat*.

- F11: Seznam objednávek k vyřízení** Objednávky zákazníků se stavem nedoručeno budou zobrazeny v seznamu objednávek k vyřízení. Budou mít dva možné stavy – *objednat* a *doručit*. V seznamu objednávek zákazníků budou nevyřízené objednávky barevně odlišeny od ostatních.
- F12: Nastavení stránky o objednávkách** V této části nastavení půjde vypnout záložku *objednávky k vyřízení*. Dále se zde bude dát vypnout možnost přípravné objednávky.
- F13: Sklad** Aplikace bude také fungovat jako skladový systém. Najdeme v ní seznam všech produktů a počet kusů na skladě. U každého produktu bude navíc uvedena doporučená cena. Při objednání produktu se potom doplní daná cena, která ovšem půjde změnit např. o slevu o určité procento nebo na konkrétní jinou částku.
- F14: Kapacitní schůzky** V kalendáři půjde vytvořit tzv. kapacitní schůzka. To znamená, že se schůzce nastaví kapacita pro určitý počet lidí. Ke schůzce se následně mohou postupně přidávat zákazníci, dokud se nenaplní. Až bude schůzka plně obsazená, označí se v kalendáři červeně.
- F15: Korespondence** Kontaktování zákazníka prostřednictvím různých prostředků – SMS, hovor nebo e-mail. U konkrétního hovoru půjde navíc označit, zda zákazník odpověděl nebo zdali například hovor nepřijal. Při nepřijetí půjde telefonát naplánovat znovu – buď za určitý počet dní nebo na další hromadné telefonování (viz **F16**).
- F16: Hromadné telefonáty** Hromadný telefonát bude určitý časový úsek určený pro obvolávání zákazníků. Do této události si bude uživatel přidávat zákazníky, které potřebuje kontaktovat s popsáním důvodem. Jednotlivé hovory se potom budou u jednotlivých zákazníků zaznamenávat v historii korespondence.
- F17: Ostatní události v kalendáři** Kromě dříve zmíněných typů událostí bude v kalendáři možnost vytvořit také obyčejnou schůzku nebo úkol.
- F18: Statistiky** Zobrazené statistiky půjdou přizpůsobit v nastavení. Budou se moct zobrazovat čítače nebo grafy za určité období a to s různými typy informací.
- F19: Smazání zákazníka** V aplikaci bude možnost smazat veškerou evidenci o zákazníkovi.

1.2.2 Nefunkční požadavky

Nefunkční požadavky popisují další nezbytné vlastnosti systému. Jsou to například nároky na výkonnost, spolehlivost, bezpečnost systému nebo udržovatelnost. [3]

N1: Desktopová multiplatformní aplikace Ze zadání plyne, že aplikace bude psaná jako desktopová aplikace v jazyce Java. Celá bude fungovat lokálně, včetně její databáze.

N2: Udržovatelnost aplikace Uživatel si bude moct měnit některé funkce a informace v aplikaci. Díky čemuž bude aplikace použitelná i v budoucnu při případné změně principů přímého prodeje.

N3: Bezpečnost aplikace a dat V aplikaci půjde měnit spoustu parametrů, nikoli však její vnitřní funkce a algoritmy. Její bezpečnost by měla především spočívat v tom, že se k uloženým datům půjde dostat pouze po přihlášení. Jinak budou zašifrovaná. Jako prevence proti ztrátě dat bude v aplikaci možnost zálohy.

1.3 Případy užití

Use Case Diagram (česky diagram případů užití) zobrazuje chování systému tak, jak ho vidí uživatel. Účelem diagramu je na základě funkčních požadavků popsat funkcionalitu systému, tedy co od něj bude uživatel očekávat. Diagram vypovídá o tom, co má systém umět, ale neříká, jak to bude dělat.

Diagram se skládá z jednotlivých případů užití, aktérů a vztahů mezi nimi. [4] Aktér je v tomto případě pouze jeden. Jediný, kdo bude aplikaci používat, je přímý prodejce.

Na obrázku 1.1 lze vidět diagram, který jsem vytvořila na základní funkce aplikace. K hlavním případům užití jsem rozepsala scénář a případně další alternativní scénáře.

U1: Přihlášení

Primární aktér: Prodejce

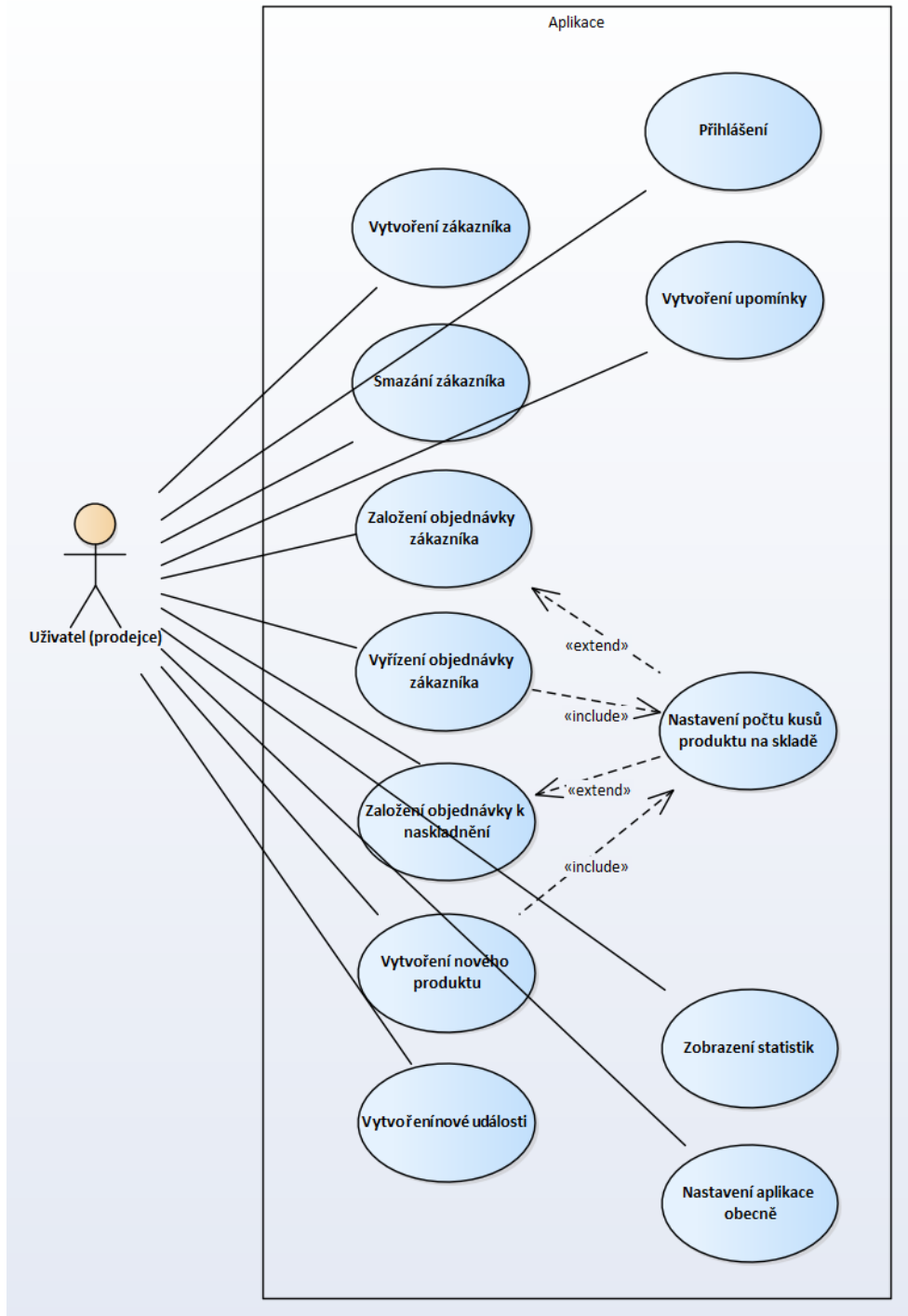
Popis: Při prvním spuštění aplikace, tedy prázdné databázi, si uživatel bude muset vytvořit účet. Následně se pomocí vytvořených přihlašovacích údajů bude přihlašovat.

U2: Nastavení aplikace obecně

Primární aktér: Prodejce

Popis: V aplikaci se bude nacházet více druhů nastavení. Jedno hlavní nastavení a pro každou jednotlivou část aplikace další. Obecně si v každém z nich uživatel vždy vybere sobě vyhovující nastavení a systém mu následně podle toho aplikaci přizpůsobí.

Obrázek 1.1: Use case diagram



U3: Vytvoření zákazníka

Primární aktér: Prodejce

Popis: Prodejce vyplní informace získané o zákazníkovi a formulář potvrdí. Systém informace uloží do databáze a zobrazí přehled zákazníků. Při vytváření bude mít uživatel také možnost navázat vztahy (evidence rodinných a přátelských vztahů klienta).

UC4: Smazání zákazníka

Primární aktér: Prodejce

Popis: Na požádání zákazníka ho prodejce bude muset ze systému smazat. Systém vyhledá všechny zmínky o něm v databázi a odstraní je.

UC5: Vytvoření upomínky

Primární aktér: Prodejce

Popis: Upomínky se buď vytvoří automaticky nebo uživatelem samotným. Automaticky se generují upomínky například na narozeniny zákazníka nebo připomenutí, že mu pravděpodobně dochází dříve objednaný produkt. Zobrazují se v profilu daného zákazníka a také v kalendáři.

U6: Založení objednávky zákazníka

Primární aktér: Prodejce

Popis: Po zadání objednaných produktů a určitého zákazníka systém objednávku uloží. Pokud prodejce objednávku neoznačil jako uzavřenou, tak ji zařadí mezi objednávky k vyřízení. Ty produkty, co nejsou na skladě, systém přidá do předpřipravené objednávky.

UC7: Vyřízení objednávky zákazníka

Primární aktér: Prodejce

Popis: Po dodání zboží zákazníkovi se objednávka označí jako vyřízená. Scénář jsem popsala v tabulce 1.1.

U8: Založení objednávky k naskladnění

Primární aktér: Prodejce

Popis: Při zakládání objednávky od dodavatele uživatel vyplní seznam doobjednaných produktů. U každého má možnost zvolit, zda je určený na sklad, pro určitého zákazníka nebo k jinému účelu. Pokud je zboží určeno na sklad, tak se po doručení automaticky navýší počet jeho kusů.

U9: Vytvoření nového produktu

Primární aktér: Prodejce

Popis: Uživatel vyplní základní informace o produktu (název, cena, apod.) a přidá aktuální počet kusů na skladě. Dále požadavek odešle a systém jej zpracuje.

U10: Vytvoření nové události**Primární aktér:** Prodejce

Popis: Události mohou být různých druhů. Ty jsou rozepsány ve funkčních v požadavcích **F14** – **F17**. Událost lze také vytvořit z různých formulářů. Z kalendáře, úvodní záložky nebo profilu zákazníka. Podrobnější scénář jsem znázornila v tabulce 1.2.

U11: Zobrazení statistik**Primární aktér:** Prodejce

Popis: Když si uživatel otevře statistiky, systém z databáze dopočítá údaje podle aktuálního nastavení.

Tabulka 1.1: Scénář k vyřízení objednávky zákazníka

Krok	Role	Akce
1	Uživatel	Přejde na přehled objednávek zákazníků.
2	Systém	Načte a zobrazí objednávky z databáze.
3	Uživatel	Otevře objednávku, kterou chce vyřídit.
4	Uživatel	Nastaví datum doručení zboží (volitelné) a označí ji za vyřízenou.
5	Systém	Zkontroluje zadané informace.
6	Systém	Aktualizuje změněná data.
7	Systém	Pokud byly některé produkty ze skladu, tak upraví jejich počet kusů na skladě.
8	Systém	Zobrazí potvrzení o uložení.

Alternativní scénář: přes formulář objednávek k vyřízení		
1a	Uživatel	Přejde na přehled objednávek k vyřízení.
2a	Systém	Načte a zobrazí všechny nevyřízené objednávky zákazníka.

Alternativní scénář: přes profil zákazníka		
1b	Uživatel	Přejde na profil zákazníka a otevře záložku <i>objednávky</i> .
2b	Systém	Načte a zobrazí všechny nevyřízené objednávky z databáze.

Alternativní scénář: chybně zadané údaje		
5c1	Systém	Vyzývá uživatele k opravě chybně zadaných údajů.
5c2	Uživatel	Údaje opraví a znovu odešle.
5c3	Systém	Znovu údaje překontroluje.

Tabulka 1.2: Scénář k vytvoření události

Krok	Role	Akce
1	Uživatel	Přejde na kalendář.
2	System	Načte a zobrazí existující události z databáze.
3	Uživatel	Zvolí vytvoření nové události.
4	Uživatel	Nastaví typ události a vyplní příslušné informace.
5	System	Zkontroluje zadané informace.
6	System	Uloží událost do databáze.
8	System	Zobrazí potvrzení o uložení.

Alternativní scénář: přes úvodní záložku

1a	Uživatel	Přejde na domovskou stránku, kde má zobrazený přehled schůzek.
2a	System	Načte a zobrazí všechny existující schůzky z databáze.
3a	Uživatel	Zvolí vytvoření nové schůzky.
4a	Uživatel	Vyplní příslušné informace a odešle.

Alternativní scénář: přes profil zákazníka

1b	Uživatel	Přejde na přehled objednávek k vyřízení.
2b	System	Načte a zobrazí všechny nevyřízené objednávky z databáze.

Alternativní scénář: chybně zadané údaje

5c1	System	Vyzývá uživatele k opravě chybně zadaných údajů.
5c2	Uživatel	Údaje opraví a znovu odešle.
5c3	System	Znovu údaje překontroluje.

1.3.1 Mapování případů užití na požadavky

Mapování případů užití na všechny požadavky je shrnuto v tabulce 1.3

Tabulka 1.3: Mapování případů užití na požadavky

	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11
F1	x										
F2		x									
F3			x								
F4					x						
F5			x								
F6					x				x		
F7		x									
F8								x			
F9						x					
F10						x					
F11							x				
F12		x									
F13									x		
F14										x	
F15										x	
F16										x	
F17										x	
F18											x
F19				x							

1.4 GDPR

Jelikož bude výsledný systém sloužit k ukládání informací o zákaznících, týká se jej nařízení o ochraně osobních údajů (GDPR¹). Což je nová legislativa, která se zaměřuje na ochranu osobních údajů s cílem hájit práva občanů EU proti neoprávněnému zacházení s jejich daty a osobními údaji. Toto nařízení vstoupí v účinnost 25. května 2018.

Obecné nařízení na ochranu osobních údajů se týká všech firem, institucí, podnikatelů a online služeb, které zpracovávají data uživatelů. Za porušení stanovených pravidel se zavádí vysoké pokuty. [5]

Nařízení přináší celou řadu nových pravidel. Jejich platnost a dodržování bude muset každý prokazatelně doložit po celou dobu zpracování.

Občané musí být informováni o svých právech. Měli by mít např. přístup k údajům, které jsou o nich shromažďovány. Nové je právo také být zapomenut, díky kterému může jedinec požadovat, aby byly bez zbytečného odkladu

¹z anglického General Data Protection Regulation

vymazány jeho osobní údaje (pokud ovšem neexistuje právní důvod pro jejich další zpracování).

S novou legislativou dochází také k rozšíření definice osobních údajů. Spadají sem navíc i technické parametry (e-mail, IP adresa, apod.).

Další povinností pro zpracovatele bude ohlásit únik či ohrožení zabezpečení osobních dat *Úřadu pro zpracování osobních údajů* nejpozději do 72 hodin od chvíle, kdy se o incidentu dozvěděl. V některých případech bude muset též informovat osoby, kterých se únik týkal. Už by se tedy nemělo stávat, že se o kauzách masivních úniků dat dozvíme až s odstupem několika let. [6]

S ohledem na toto nařízení jsem navrhla řešení, které je popsáno v kapitole 2.5.

1.5 Analýza existujících CRM systémů

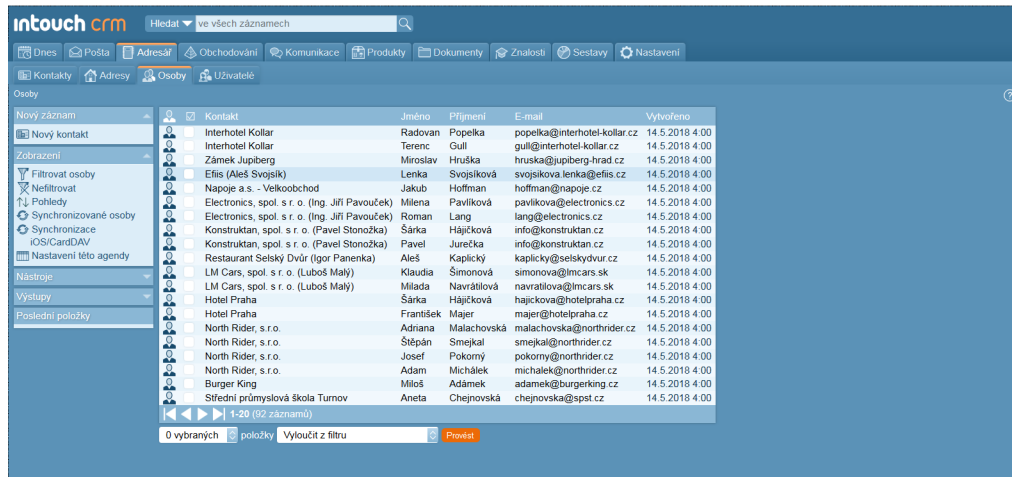
V dnešní době existuje CRM systémů velké množství, ale ve většině případů jsou přizpůsobené potřebám větších firem, nikoliv přímých prodejců. Další zřejmou nevýhodou je jazyková podpora, protože v českém jazyce se mi jich podařilo najít pouze pár (jeden pro firmy a jeden částečně pro přímé prodejce).

Systémy pouze pro přímé prodejce podnikatele existují, ale mají je převážně konkrétní společnosti pro své vlastní prodejce. Proto se k nim běžný člověk nedostane. Příkladem je systém pro Mary Kay, který je obchodním tajemstvím společnosti, čili ho zde nesmím zanalyzovat.

Při hledání systémů k analýze jsem často narážela na problém, že po mně chtěli údaje o platební kartě jenom k tomu, abych si je mohla vyzkoušet. Kvůli tomu jsem některé zavrhla. U BPM Online (viz kapitola 1.5.2), které později přeložili i do českého jazyka, jsem měla problém s registrací. Ve formuláři nešlo vyplnit jedno povinné pole a kvůli tomu jsem se nemohla zaregistrovat. Po upozornění e-mailem to však opravili.

1.5.1 InTouch CRM

Obrázek 1.2: Ukázka prostředí systému InTouch CRM



InTouch CRM² je poměrně rozsáhlý český systém. Ukázka prostředí systému je na obrázku 1.2. Verze ke stažení je zdarma a na cloudu placená. Je v něm možnost vést databázi klientů, objednávek, zboží na skladě a organizaci schůzek, hovorů a úkolů. Přesto, že nabízí tyto funkčnosti, jsem zhodnotila, že pro potřeby přímého prodeje není vhodný. V určitých oblastech je možností až příliš a v jiných naopak nedostatek, viz tabulka 1.4. Je jasné, že tento systém není dělaný pro přímé prodejce, ale pro běžné větší firmy.

Tabulka 1.4: Tabulka výhod a nevýhod systému InTouch CRM

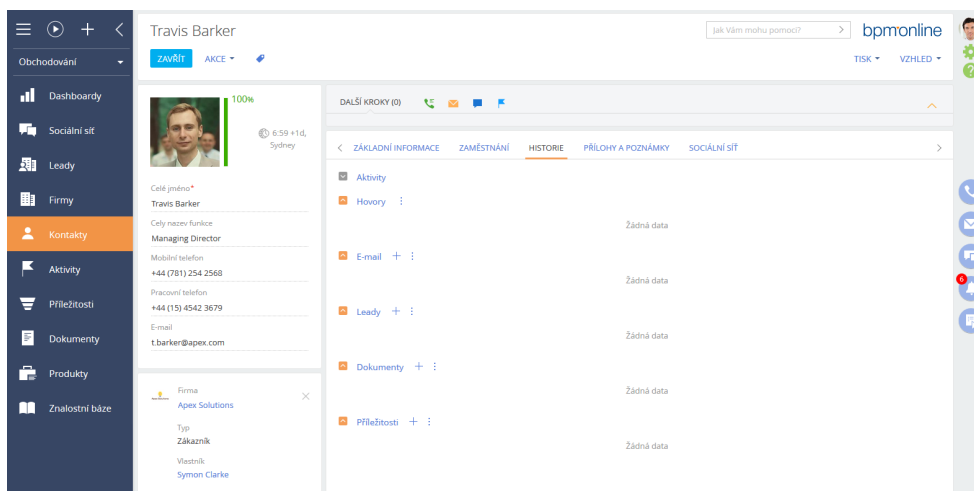
Výhody	Nevýhody
<ul style="list-style-type: none"> Možnost přidávat, co si o kontaktech chceme ukládat. V profilu kontaktu možnost přidat různé události (kalendář, objednávka, vztahy). Seznam nabízených produktů a jejich ceny. Lze vypnout/zapnout jednotlivé funkce. 	<ul style="list-style-type: none"> O kontaktech vedeny informace, které jsou pro přímý prodej nepotřebné a které nelze odebrat (např. IČ, DIČ). Objednávky jsou pouze <i>Služby</i>, <i>Dlouhodobé služby</i>, <i>Reklamace</i> a <i>Projekt</i>. Ke schůzce do kalendáře nelze přidat více osob. Pro účely přímého prodeje se v aplikaci nachází příliš funkcí navíc.

²viz <http://intouch-crm.cz/>

1. ANALÝZA

1.5.2 BPM online

Obrázek 1.3: Ukázka prostředí systému BPM online



BPM Online³ je dle mého názoru ze všech systémů nejvhodnější pro přímé prodejce. Přesto má ale stále své nevýhody, jak jsem poznamenala v tabulce 1.5. Prostředí systému je vidět na obrázku 1.3. Nabízí různé verze pro menší i větší firmy s přímými prodejci. Původně byl pouze v anglickém jazyce, před nedávnem byl však přeložen do češtiny.

Nabízí širokou škálu možností v oblasti vedení zákazníků. Informace v profilu se dají přidat nebo změnit a je zde možnost je i jakkoliv přeskupit. V tom vidím největší výhodu tohoto systému. Systém také nabízí funkci ukládání různých článků a dokumentů, což se může hodit i přímým prodejcům.

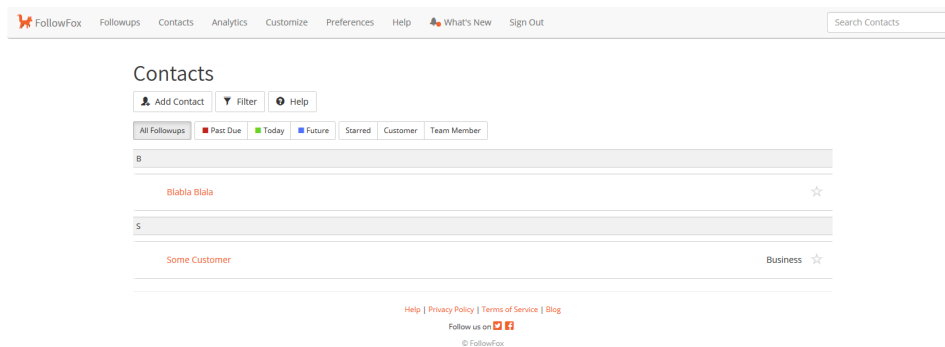
Tabulka 1.5: Tabulka výhod a nevýhod systému BPM Online

Výhody	Nevýhody
<ul style="list-style-type: none">• Aplikace se dá hodně přizpůsobit.• Kontakty lze rozdělit do složek.• Synchronizace s Google kalendářem.	<ul style="list-style-type: none">• Lze přidat pouze jednoho účastníka ke schůzce při vytvoření. Další jdou přidat až v detailu události.• Ve verzi pro menší firmy nejsou objednávky.• Dost funkcí navíc určených pro firmy.• Aplikace je placená.

³viz <https://www.bpmonline.com/crm/direct-sales>

1.5.3 FollowFox

Obrázek 1.4: Ukázka prostředí systému FollowFox



FollowFox⁴ je CRM systém v anglickém jazyce, který ovšem nabízí pouze databázi kontaktů a k nim určitých interakcí (setkání, telefonát), jak je vidět na obrázku 1.4. Je zde možnost přidávat své vlastní pole do profilu. Líbí se mi, jakým stylem mají přidávání udělané. Nejdřív si člověk určí pouze název a typ a podle toho si dále otevře stránku editace, kde si může přidávat jednotlivé položky (např. u comboboxu).

Přesto, že systém nepokrývá potřebný rozsah, zaujal mě svou jednoduchostí a propracovaností jednotlivých funkcí. Výhody a nevýhody jsem shrnula do tabulky 1.6.

Tabulka 1.6: Tabulka výhod a nevýhod systému FollowFox

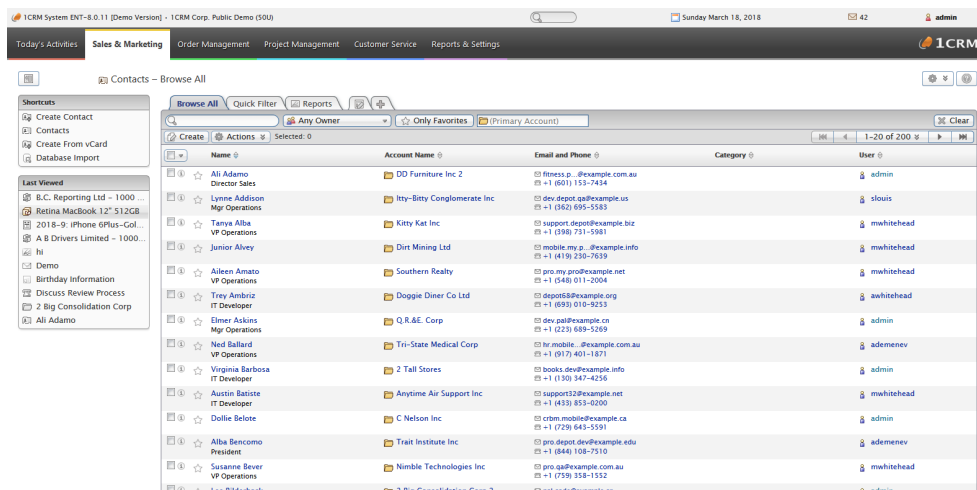
Výhody	Nevýhody
<ul style="list-style-type: none"> • Jednoduchost. • Šikovně udělané přidávání polí do profilu. • Základní verze je zdarma. 	<ul style="list-style-type: none"> • Pouze databáze kontaktů a událostí, nikoli objednávek a produktů. • Nelze přidávat schůzky po více lidech.

⁴viz <https://followfox.com/>

1. ANALÝZA

1.5.4 1CRM

Obrázek 1.5: Ukázka prostředí systému 1CRM



1CRM⁵ je dalším z anglických systémů, určených primárně pro firmy. Už na první pohled je vidět, že se nejedná o jednoduchý systém, viz obrázek 1.5. Zaujalo mě, že je zde možnost vést si katalog, objednávky zboží a objednávky od zákazníků. V nejnovější verzi si v systému můžete přidávat vlastní pole k jednotlivým formulářům. Nejdou však upravovat ty stávající. Kalendář je poměrně propracovaný, nabízí i přidání více uživatelů ke schůzce. Všechny výhody a nevýhody systému jsem shrnula v tabulce 1.7.

Tabulka 1.7: Tabulka výhod a nevýhod systému 1CRM

Výhody

- Katalog nabízeného zboží a objednávky zákazníků a zboží.
- Lze přidávat vlastní pole k různým formulářům.

Nevýhody

- Nemí v českém jazyce.
- Stávající pole u formulářů nelze upravit.
 - Pro velké firmy. Spousta pro přímý prodej nevyužitelných funkcí.
- Aplikace je placená.

⁵ viz <https://1crm.com/>

1.5.5 Shrnutí

Obecně mi systémy až na FollowFox přijdou poměrně složité. Další nevýhodou je, že valná většina CRM systémů je placená. A když už mají verzi zdarma (viz FollowFox, kapitola 1.5.3), tak rozhodně nepokrývá všechny funkce, které přímí prodejci využijí.

Nejvíce z nich se mi líbily systémy BPM Online a právě zmiňovaný Follow-Fox, které jsou pro přímý prodej alespoň částečně určeny. Ve většině z nich jsou tzv. *Příležitosti* a spolupracující firmy. Což pro přímý prodej může mít určitý význam, ale myslím si, že je pro něj mnohem využitelnější vést si objednávky zboží a objednávky od zákazníků. V žádném z uvedených systémů jsem ale nenašla možnost kapacitních schůzek, ani připravování objednávek.

Návrh a implementace prototypu

Jak je uvedeno v zadání, výsledkem této práce bude pouze prototyp systému. V této kapitole se zaměřuji na návrh prototypu a popis implementace. Na úvod však rozepíši technologie, které ve své implementaci používám.

2.1 Zvolené technologie

Systém implementuji v souladu se zadáním jako desktopovou aplikaci v jazyce Java. Pro vytvoření GUI jsem se rozhodla použít JavaFX, což je framework pro tvorbu okenních aplikací. Na výběr jsem měla ještě mezi Swingem, ale ten je již v porovnání s JavaFX zastaralý. JavaFX podporuje obrázky, videa, hudbu, CSS styly a další technologie. [7] Je poměrně snadné v ní vytvořit vzhledově moderní aplikaci, což se mi na tom líbí. V rámci JavaFX navíc používám open source knihovnu JFoenix⁶, která implemetuje dnes populární materiální design Googlu.

2.2 Návrh architektury systému

Ve své aplikaci jsem se rozhodla využít vícevrstvé architektury, konkrétně dvouvrstvé. Vícevrstvá architektura znamená, že jsou funkčnosti programu rozděleny mezi několik vzájemně spolupracujících vrstev. [8] V mém případě se jedná o vrstvu prezentační a vrstvu datovou. Jako návrhový vzor pro prezentační vrstvu používám MVC, neboli Model-View-Controller.

V balíčku `constants` se navíc nachází třídy s konstantami – například `GeneralConstants` nebo `TableConstants`.

⁶více na <http://www.jfoenix.com/>

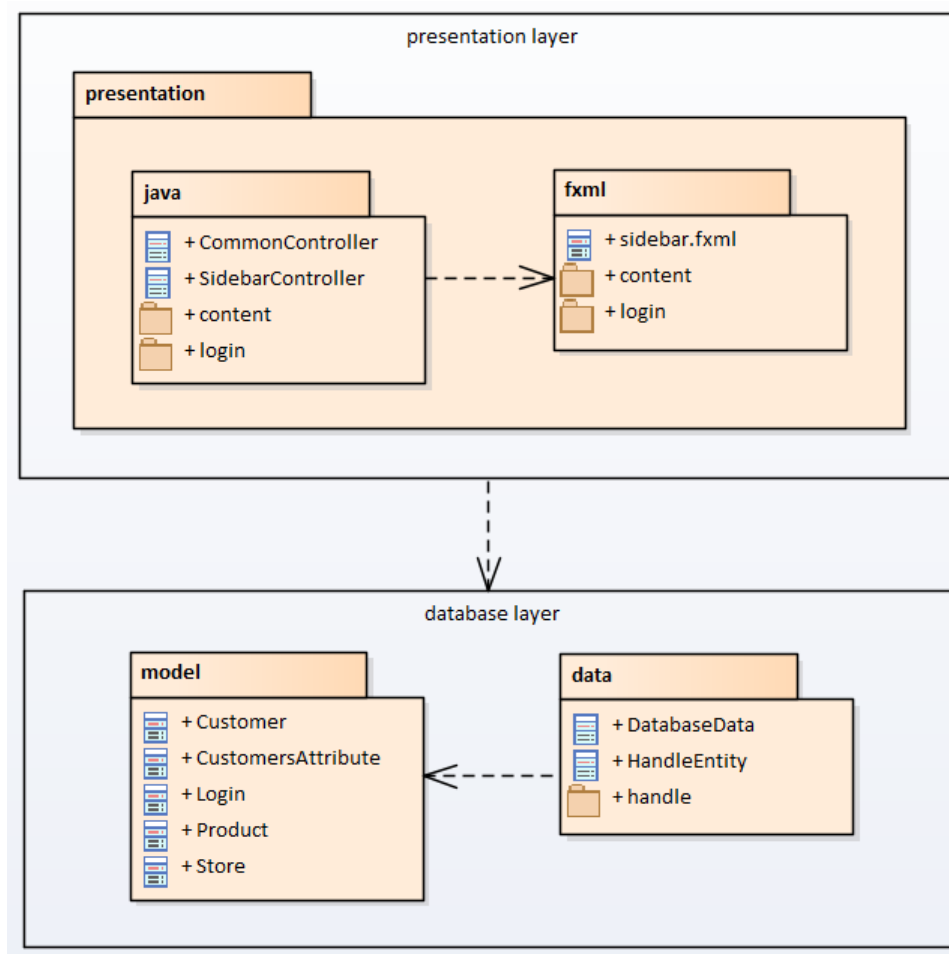
2.2.1 Prezentační vrstva

Jak lze vidět diagramu balíčků na obrázku 2.1, prezentační vrstva obsahuje ze vzoru MVC pouze View a Controller (v mém případě balíčky `fxml` a `java`). Třída `CommonController` v balíčku `java` je obecný controller, od kterého dědí všechny ostatní. Obsahuje metody, které jsou pro ostatní controllery společné. Navíc je v něm zahrnuta třída `DatabaseData`, díky čemuž mají všechny controllery přístup k databázi.

2.2.2 Datová vrstva

V datové vrstvě lze nalézt balíčky `model` a `data`, viz obrázek 2.1. V balíčku `model` (část Model z MVC vzoru) se nachází objekty reprezentující entity v databázi, ke kterým přes třídu `DatabaseData` přistupují třídy z prezentační vrstvy.

Obrázek 2.1: Diagram balíčků



V balíčku `handle` se nachází třídy `HandleEntity` (např. `HandleCustomer`, `HandleProduct`, apod.), které jsou obsaženy ve třídě `DatabaseData`. Ty slouží ke komunikaci s databází, která probíhá prostřednictvím JDBC⁷, což je API pro jazyk Java definující rozhraní pro přístup k relačním databázím. [9]. Tyto `HandleEntity` jsou vytvořené podle návrhového vzoru Data Access Object. [10] Třída `DatabaseData` slouží k vytvoření připojení k databázi a zaobalení tříd `HandleEntity`.

2.2.3 Příklad

Na ukázkou přístupu k databázi jsem vybrala získávání všech produktů do přehledu v záložce *sklad*.

Třída `StoreController` obsluhuje hlavní stránku *skladu*. Při vytváření přehledu produktů si controller volá metodu `getAllProducts()`, která se nachází v třídě `HandleProduct`, jak je vidět na ukázce kódu 2.2. Ta vrací list produktů, které následně `StoreController` zobrazí v přehledu.

Obrázek 2.2: Ukázka volání metody `getAllProducts()`

```
1 productsData = data.handleProduct.getAllProducts();
```

Obrázek 2.3: Ukázka implementace metody `getAllProducts()`

```
1 public ObservableList<Product> getAllProducts()
2     throws SQLException {
3     return getAllProducts(executeQueryWithText(con,
4         format(SELECT + FROM, "*", TABLE_PRODUCT)
5     ));
6 }
```

Hodnotu konstant `SELECT` a `FROM` jsem vložila v kódu níže, viz obrázek 2.4, aby bylo jasné, jak v implementaci dotazy skládám. Konstanty s SQL dotazy se nachází v třídě `SQLCommands`.

Obrázek 2.4: Ukázka konstant na volání SQL dotazů

```
1 public static final String SELECT = "SELECT_%s_";
2 public static final String FROM = "FROM_%s_";
```

Konstanta `TABLE_PRODUCT` nese název entity v databázi a nachází se v třídě `TableContents`. V této třídě jsou navíc další konstanty nesoucí název jednot-

⁷z anglického Java Database Connectivity

livých atributů. Jako například `TABLE_PRODUCT_PK`, která nese název atributu s primárním klíčem entity. Takto jsou v uvedené třídě konstanty pro všechny entity.

Výše zobrazená metoda `getAllProducts()` volá přetíženou metodu, která se nachází také ve stejné třídě `HandleProduct`, `getAllProducts(ResultSet)`. Tato metoda zpracovává přijatý `ResultSet` a vytváří z něj list objektů `Product`, který následně vrací. Metoda `executeQueryWithText(Connection, String)` provádí samotný SQL dotaz. Ukázku implementace výše zmíněné metody `getAllProducts(ResultSet)` jsem přiložila níže, viz obrázek 2.5.

Na ukázce implementace metody je vidět, že si pro každý výsledek ukládám všechny atributy do proměnných, viz `id`, `name`, a další. Ty pak jednoduše zahrnu do konstruktoru při vytváření objektu `Product`. Každý jeden výsledek navíc přidám do kolekce `ObservableList<Product> data`, kterou metoda nakonec vrací.

V implementaci používám `ObservableList` kvůli přehledu, který je dělaný pomocí třídy `JFXTreeTableView` a která daný list přijímá.

Obrázek 2.5: Ukázka implementace metody `getAllProducts(ResultSet)`

```

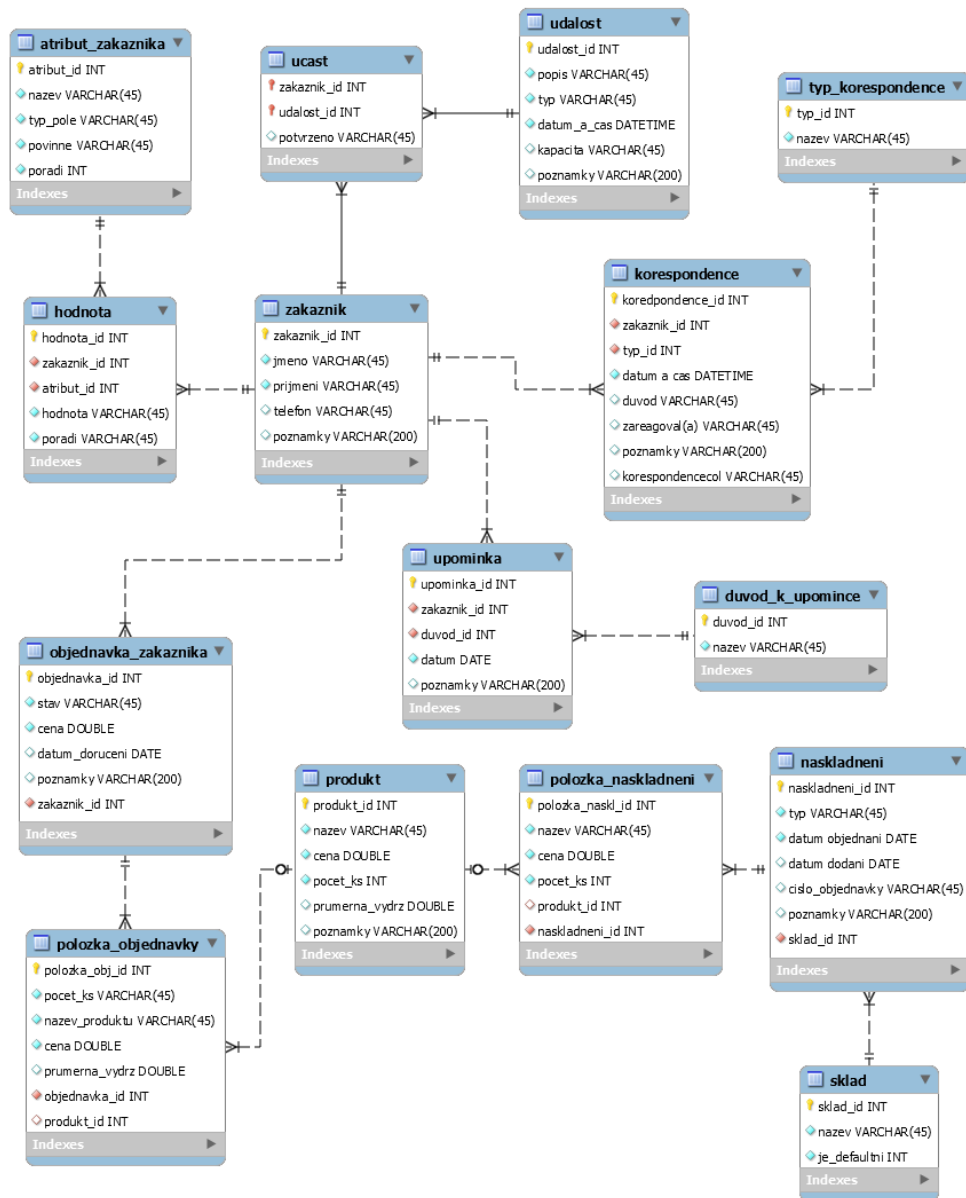
1 private ObservableList<Product> getAllProducts(ResultSet
2     resultSet) throws SQLException {
3     ObservableList<Product> data = FXCollections.
4         observableArrayList();
5     while(resultSet.next()) {
6         int id = resultSet.getInt(TABLE_PRODUCT_PK);
7         String name = resultSet.getString(TABLE_PRODUCT_NAME);
8         int price = resultSet.getInt(TABLE_PRODUCT_PRICE);
9         int count = resultSet.getInt(TABLE_PRODUCT_COUNT);
10        int endurance = resultSet.getInt(
11            TABLE_PRODUCT_ENDURANCE
12        );
13        String notes = resultSet.getString(TABLE_PRODUCT_NOTES);
14        if(notes == null) { notes = ""; }
15        data.add(new Product(id, name, price, count,
16            endurance, notes));
17    }
18    return data;
19 }

```

2.3 Způsob uložení dat

Databázi jsem vytvořila v MySQL pomocí nástroje MySQL Workbench. Pro potřeby prototypu jsem se rozhodla, že stačí databáze lokální. Na obrázku 2.6 je vidět struktura hlavní části databáze.

Obrázek 2.6: Struktura databáze



2.3.1 Zákazník

Uprostřed je entita `zakaznik`. Ta obsahuje pouze atributy, které jsou povinné a nepůjdou smazat. Dynamické přidávání libovolných atributů řeším pomocí dalších entit – `atribut_zakaznika` a `hodnota`. Entita `atribut_zakaznika` slouží k přidávání libovolného počtu polí. Jejich hodnoty jsou pak ke konkrétním zákazníkům uloženy v entitě `hodnota`. Dynamické přidávání atributů mám v prototypu jen u zákazníka. U ostatních entit se to do budoucna bude řešit stejným způsobem. Kód programu je tomu již přizpůsobený.

2.3.2 Objednávky

Naskladnění zboží mám v databázi řešené entitou `naskladneni`. Při vyplnění data doručení se změní počty kusů produktů na skladě o ty dodané. K této entitě jsou navázané další entity `polozka_naskladneni`. Každá jedna přitom obsahuje jeden produkt a počet kusů, které se od daného produktu objednal. Objednávky zákazníků řeším stejným způsobem jako naskladnění. Entita `objednavka_zakaznika` s navázanými položkami `polozka_objednavky`.

U entit `polozka_naskladneni` a `polozka_objednavky` je cizí klíč nepovinný (`produkt_id`), neboť mezi tím může uživatel produkt smazat. Potom nechceme o jednotlivé objednávky přijít. Z toho důvodu se informace o produktu kopírují do výše uvedených entit.

2.3.3 Kalendář

Kalendář v prototypu zahrnut není, databázi jsem pro něj však navrhla následovně. Jedna entita je pro události obecně a druhá pro korespondenci (ukládání údaje se zde trochu liší). Entita `udalost` představuje jak normální schůzku, tak i kapacitní schůzku (viz požadavek **F14**) nebo například úkol. Schůzku od obyčejné události přitom odlišuji tak, že jednoduše k události nenavážu žádnou účast a nenastavím její kapacitu.

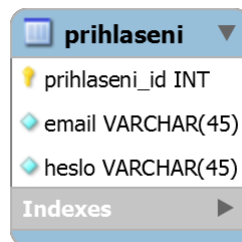
Entita zajišťující korespondenci k sobě musí mít navázaný typ, viz entita `typ_korespondence`. Díky této entitě si uživatel bude moci přidávat další nové typy korespondencí.

Entita `upominka` slouží k evidenci upomínek k jednotlivým zákazníkům. Je k ní navíc navázána entita `duvod_k_upomince`, která umožňuje uživateli přidat své vlastní důvody do nabídky. Funkce upomínek je v implementaci prototypu zahrnuta. Zobrazuje se pouze v profilu zákazníka a na domovské stránce, nikoli však v kalendáři, který implementovaný není.

2.3.4 Přihlašování

K zapamatování přihlašovacích údajů slouží entita `prihlaseni`, viz obrázek 2.7. Pro potřeby prototypu bude vždy existovat pouze jeden účet. Do budoucna však do aplikace zapracuji možnost mít více účtů na jednom počítači.

Obrázek 2.7: Databáze k přihlašování



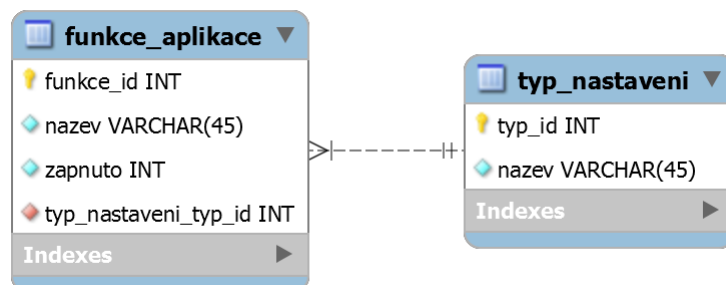
2.3.5 Nastavení

Jelikož má být aplikace přizpůsobitelná, vyžaduje potřebu mít nastavení – čili mít pro nastavení v databázi vlastní entity. Pro potřeby prototypu stačí pouze dvě, viz obrázek 2.8.

Všechno, co bude mít uživatel možnost nastavit, je pouze vypínání a zapínání funkcí v aplikaci. Až na přidávání a odebírání polí u evidovaných informací o zákazníkovi, které jsou řešené a popsané výše na obrázku 2.6 (viz entity `atribut_zakaznika` a `hodnota`).

Entita `funkce_aplikace` tedy představuje všechny vypínatelné funkce. Nese název dané funkcionality a informaci o tom, zdali je zapnutá či vypnutá. K ní navázaná entita `typ_nastaveni` potom nese informaci o tom, k jakému nastavení se daná funkce váže.

Obrázek 2.8: Databáze k nastavení aplikace



2.4 Návrh grafického rozhraní

Důkladné rozmyšlení GUI je důležitá část návrhu. Většina z analyzovaných systémů byla poměrně složitá. Proto je pro tuto konkrétní aplikaci o to podstatnější.

Nová aplikace musí být na první pohled jednoduchá a intuitivní. Je potřeba vybrat jenom ty funkce, které jsou pro toto odvětví opravdu využitelné. Na levé straně aplikace se bude nacházet menu s následujícími položkami: *domů*, *zákazníci*, *objednávky*, *sklad*, *kalendář* a *statistiky*. V horní části aplikace bude odkaz na hlavní nastavení, ve kterém půjde jakoukoliv část kromě přehledu zákazníků vypnout. Bez této položky by totiž aplikace postrádala smysl.

V rámci této části návrhu jsem neznázornila všechny obrazovky, protože je jich vzhledem k rozsáhlosti systému velké množství. Ke každé položce v menu je proto jedna až několik obrazovek pro představu. Návrh jsem si kvůli přehlednosti rozdělila podle výše zmíněných celků.

Obrazovky kalendář a statistiky jsem v rámci prototypu neimplementovala. V rámci těchto částí se tedy jedná pouze o návrh budoucí aplikace.

2.4.1 Domů

Tato stránka se zobrazí hned po přihlášení uživatelem. Její rozložení jsem znázornila na obrázku 2.9. Zde je vidět, že se na této stránce bude zobrazovat přehled událostí, upomínek, apod. Tento přehled se budou dát editovat v nastavení dané záložky. Půjde vybrat vždy jen jeden.

2.4.2 Zákazníci

Obrazovka se zákazníky obsahuje jejich seznam, jak lze vidět na obrázku 2.10. V něm je možné řadit dle všech sloupců vzestupně i sestupně.

Po rozkliknutí kontaktu se zobrazí jeho profil, viz obrázek 2.11. Zde jsou vidět všechny funkce, které lze v rámci profilu použít – *schůzky*, *korespondence*, *objednávky* a *upomínky*. Každá ze záložek se dá vypnout v příslušném nastavení.

2.4.3 Objednávky

S objednávkami už je to složitější. Je zde potřeba oddělit odebírání zboží od dodavatele a následné dodávání zákazníkům. Proto, jak jsem nastínila na obrázku 2.12, jsou na to v prostředí dvě různé záložky – *naskladnění* a *objednávky zákazníků*. Objednávky k naskladnění navíc zahrnují předpřipravenou objednávku, kterou jsem zmínila v požadavku **F10**. Nachází se vždy na prvním místě a je od ostatních barevně odlišená.

Na obrázku 2.13 jsem znázornila, jak vypadá formulář k vytvoření nové objednávky zákazníka.

Poslední záložka *objednávky k vyřízení* obsahuje přehled objednávek, které ještě nebyly zákazníkům dodány.

2.4.4 Sklad

Dále se v menu nachází *sklad*, ve kterém je seznam prodáváných produktů. U každého produktu je navíc ještě informace o počtu kusů na skladě, což jsem znázornila na obrázku 2.14.

2.4.5 Kalendář

Jak je vidět na návrhu obrazovky 2.15, kalendář s událostmi bude možné zobrazit jako denní, týdenní nebo měsíční. U týdenního kalendáře budou dny vedle sebe, přičemž nebudou zobrazené časy. Časový rozsah uvidíme až u konkrétní události. Ty budou navíc v jednom dni uspořádané tak, jak půjdou postupně po sobě.

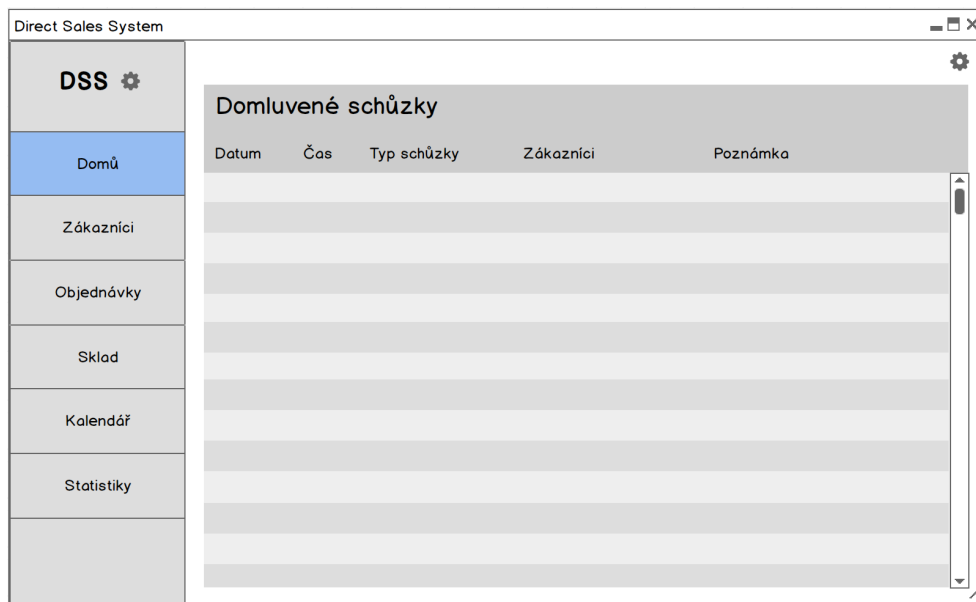
Různé druhy událostí od sebe budou barevně odlišeny. Barvy se budou dát v nastavení editovat.

2.4.6 Statistiky

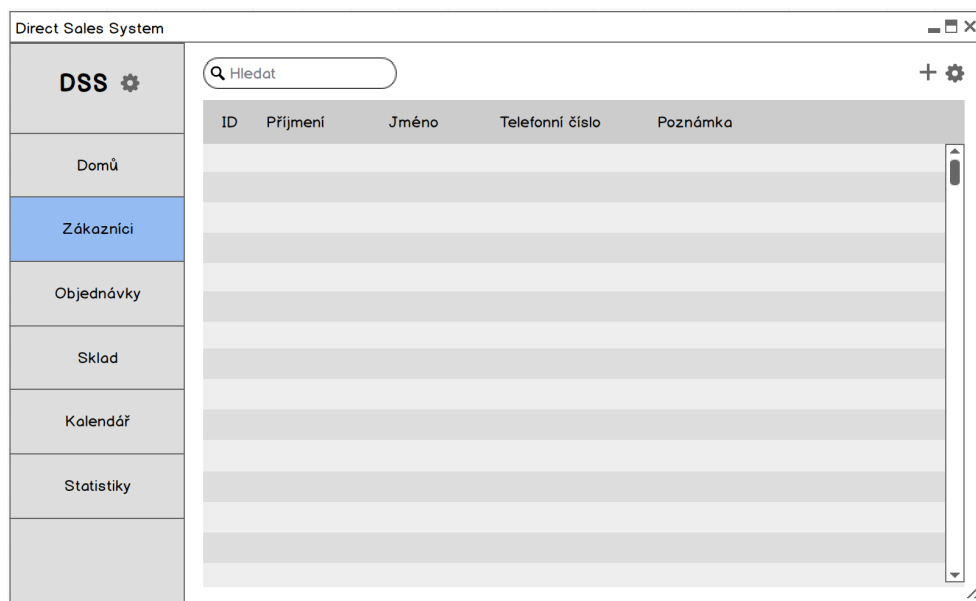
Poslední záložka se statistikami bude mít výchozí nastavení, které jsem navrhla na obrázku 2.16. V horní části budou tři čítače a v dolní graf. V nastavení půjde tato obrazovka přizpůsobit.

2. NÁVRH A IMPLEMENTACE PROTOTYPU

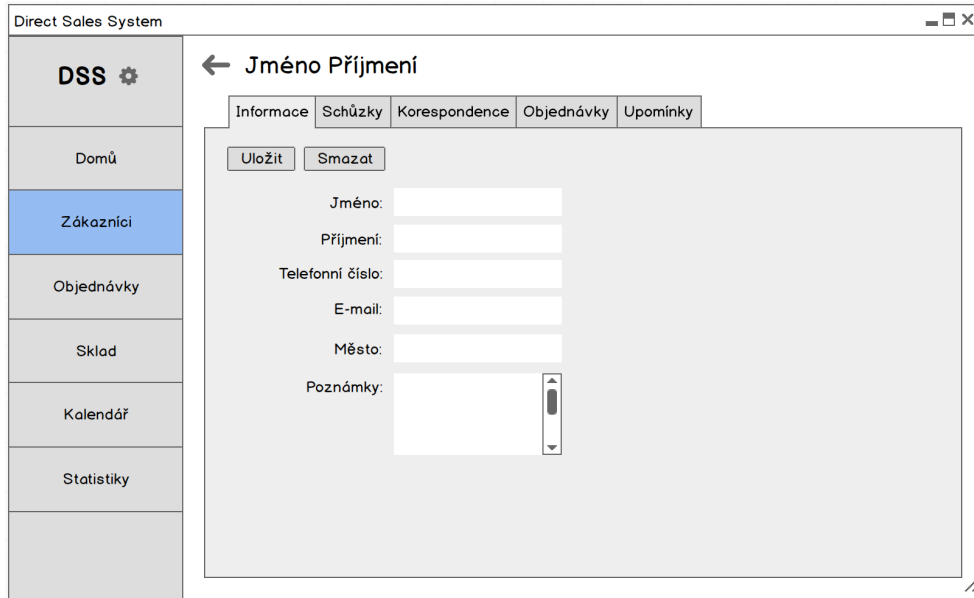
Obrázek 2.9: Návrh obrazovky *domů*



Obrázek 2.10: Návrh obrazovky *zákazníci*



Obrázek 2.11: Návrh obrazovky *zákazníci – profil*



Obrázek 2.12: Návrh obrazovky *objednávka*



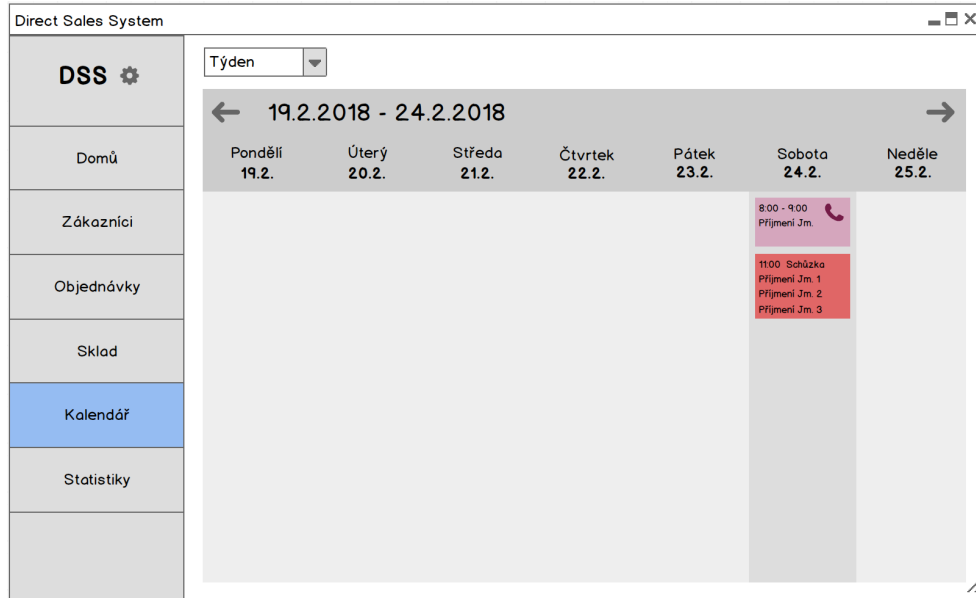
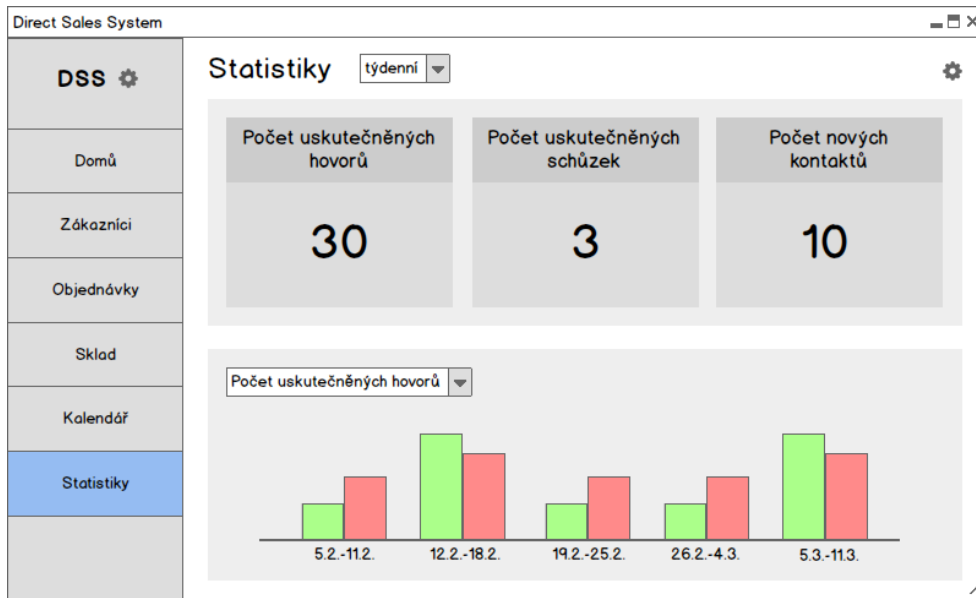
2. NÁVRH A IMPLEMENTACE PROTOTYPU

Obrázek 2.13: Návrh obrazovky *objednávka – nová*

The screenshot shows a web application window titled "Direct Sales System". On the left is a sidebar menu with the following items: "DSS" (with a gear icon), "Domů", "Zákazníci", "Objednávky" (highlighted in blue), "Sklad", "Kalendář", and "Statistiky". The main content area has three tabs: "Naskladnění", "Objednávky zákazníků", and "Objednávky k vyřízení". The active tab is "Objednávky k vyřízení", which contains a sub-header "Vytvořit novou objednávku zákazníka" and a "Vytvořit" button. Below this are several form fields: "Zákazník:" with a dropdown menu showing "Jméno Příjmení"; "Stav doručení:" with a dropdown menu showing "Objednat"; "Datum doručení:" with an empty text input field; and "Poznámky:" with a text area. Below the form fields is a section labeled "Produkty: +" containing two rows of product selection. Each row has a "Produkt" dropdown, a "počet kusů" input field, a "cena produktu" input field, and a "Smazat" button. At the bottom of the form, it displays "Cena celkem: 0".

Obrázek 2.14: Návrh obrazovky *sklad*

The screenshot shows the "Direct Sales System" window with the "Sklad" (Inventory) page selected in the sidebar. The main content area features a search bar labeled "Hledat" and a table with the following columns: "ID", "Název", "Počet kusů", "Cena", and "Poznámka". The table body contains several empty rows, indicating a list of inventory items. A vertical scrollbar is visible on the right side of the table.

Obrázek 2.15: Návrh obrazovky *kalendář*Obrázek 2.16: Návrh obrazovky *statistiky*

2.5 Návrh řešení GDPR

V prototypu jsem se GDPR prozatím ještě nezabývala, ale do budoucna jsem jeho řešení navrhla následovně.

Jedním z požadavků obecného nařízení o ochraně osobních údajů je, že zákazníci musí souhlasit s tím, že budou evidováni. Navíc musí být informováni o svých právech. Navrhuji tedy vytvořit pro prodejce listinu, která by zákazníka seznámila se vším, co potřebuje znát. Pokud by s tím souhlasil, jednoduše by ji podepsal.

Zákazník si také může vyžádat vše, co si o něm prodejce ukládá nebo ho požádat, aby o něm všechny uložené informace vymazal (právo být zapomenut). Z tohoto důvodu bych do funkcí systému zahrнула i snadné generování všech dat souvisejících s daným zákazníkem. Smazání veškerých dat souvisejících se zákazníkem je zahrnuto v požadavku **F19**.

Jelikož si v systému budou prodejci ukládat citlivé údaje svých klientů, je potřeba mít tato data zabezpečena. To zajišťuje přihlášení do aplikace pod heslem (viz požadavek **F1**), které si uživatel na začátku zvolí. V databázi bude uložen pouze jeho hash a při přihlášení se vždy hashe porovnají. Při shodě se uživatel dostane do aplikace a k datům.

2.6 Zhodnocení prototypu

V prototypu se mi podařily implementovat základní funkce navrženého systému. Aplikace nyní umožňuje evidenci zákazníků (viz **F3**), zboží na skladě (viz **F13**), vytváření upomínek k jednotlivým zákazníkům (viz **F6**, akorát prozatím ještě ne automatické) a vytváření objednávek zákazníků (viz částečně **F9**). Zmíněné funkce byly v rámci této práce otestovány, jak je popsáno v následující kapitole 3. Prototyp byl prozatím doladován na operačním systému Windows 7.

Implementací zbylých funkcí a změn plynoucích z proběhlého testování (viz kapitola 3.3) vznikne plnohodnotná aplikace splňující požadavky stanovené na začátku práce – jednoduchá aplikace, která zároveň obsahuje vše, co přímý prodejce potřebuje.

Testování

3.1 Průběžné testování

V rámci průběžného testování byly prováděny pravidelné manuální smoke testy. Ty slouží k rychlé kontrole, zda všechny již implementované části aplikace fungují správně. Byly prováděny po každé větší změně v implementaci aplikace.

3.2 Uživatelské testování

Po dokončení implementace jsem provedla s dalšími testery větší testování z pohledu uživatele aplikace. Připravila jsem scénáře, které pokrývají základní funkce a podle kterých se aplikace otestovala. Scénáře se nachází v příloze A. Aplikaci jsme však otestovali i bez scénářů.

3.2.1 Nalezené chyby

V rámci uživatelského testování se objevilo pár problémů. Všechny byly reportovány a vyřešeny. Nalezené chyby jsem shrnula níže.

1. Při aktualizaci informací o zákazníkovi v jeho profilu lze změny uložit, i když nejsou vyplněna všechna povinná pole. Chybu jsem již opravila.
2. Při vytváření a úpravě produktu neprobíhá kontrola, zdali jsou v polích *cena*, *počet kusů* a *průměrná výdrž* číselné hodnoty. Dotaz na databázi tedy vyhodí výjimku, pokud se v polích číselná hodnota nenachází. Obdobný problém se později objevil i u objednávek zákazníka. Do obou formulářů jsem přidala metodu kontrolu těchto polí.

3.3 Změny plynoucí z provedeního testování

Během uživatelského testování jsem byla upozorněna hned na několik úprav a nových funkcí, na které jsem v rámci návrhu nemyslela a které by přispěly ke zlepšení celého systému. Bylo by tedy dobré je do budoucna zahrnout do implementace aplikace.

1. Vytváření šablon e-mailů a jejich následné odesílání je jedna z funkcí, kterou přímí prodejci zrovna hodně využívají. Něco takového by tedy v tomto typu systému rozhodně nemělo chybět.
2. Každý uživatel by ocenil, kdyby události vytvořené v aplikaci měl zároveň ve svém telefonu nebo jakémkoliv jiném zařízení. Proto by se do budoucna do aplikace měla zahrnout synchronizace kalendáře v aplikaci s Google kalendářem.
3. V aplikaci momentálně není historie (neboli tlačítko zpět). Tuto funkci by také bylo vhodné do budoucna do systému zakomponovat. Když se uživatel například omylem překlikne jinam, může se tímto způsobem jednoduše vrátit zpátky na místo, kde předtím byl.
4. Pro smazání zákazníků, produktů, apod. v aplikaci nyní musí uživatel nejdříve přejít na detail a až poté může požadovanou položku smazat. To je ve spoustě případech velmi zdržující. Proto bylo navrženo řešení, že by každá položka seznamu obsahovala tlačítko na smazání už v přehledu. Nemuselo by se tedy pokaždé až do detailu položky.
5. Co se týče statistik, aktuální řešení systému neumožňuje zobrazovat statistiky výdělků prodejců. Aby to bylo možné, bude se muset v systému zavést nastavení provizí. Jedna z možností řešení je, že by se v objednávce zákazníka u každého produktu nastavila výše provize – některý produkt třeba uživatel může mít od dodavatele za jinou cenu nebo ho za jinou cenu prodat.
6. Když si prodejci objednávají zboží, často za to platí poštovné. Proto by bylo vhodné přidat do objednávek k naskladnění i tuto informaci. V nastavení může být přednastavená cena poštovného, která se pak v rámci objednávky bude dát případně i změnit.
7. Pro bezpečnost aplikace by bylo lepší nastavit nějaká pravidla k vytvoření hesla. Například minimální délku a znaky z alespoň dvou sad. V nynější implementaci je možné zadat heslo jakékoliv, i pouze jednopísmenné, což bezpečné rozhodně není.
8. Při vytvoření nebo úpravě objednávky zákazníka by bylo dobré, kdyby se po nastavení data doručení stav objednávky automaticky nastavil na *doručeno*.

Dokumentace

V této kapitole se věnuji dokumentaci vytvořeného systému. Dokumentace popisuje způsob, jakým má uživatel aplikaci používat. Dokumentace prozatím obsahuje jen ty funkce, které jsou v rámci prototypu implementovány, s výjimkou nastavení. Nastavení sice implementované v rámci prototypu není, ale i tak mi přijde důležité tuto část zdokumentovat.

4.1 První přihlášení

Při prvním spuštění aplikace se zobrazí obrazovka, kde si uživatel vytvoří svůj účet. Systém po uživateli vyžaduje e-mail a heslo. Je nutné zadat správný e-mail, ke kterému má uživatel přístup, neboť při zapomenutí hesla na něj systém odešle heslo nové. Po zadání korektních údajů a potvrzení formuláře se může uživatel přihlásit. Hned po přihlášení se doporučuje si nejprve projít nastavení aplikace.

4.2 Nastavení aplikace

V aplikaci se nastavení nachází hned několik. Jedno hlavní v levé části aplikace. Další jsou v jednotlivých záložkách. Ty se vždy vztahují pouze k jednotlivým částem aplikace.

4.2.1 Hlavní nastavení

V hlavním nastavení si uživatel může vypnout jednotlivé položky v menu. Ne však všechny. Záložky, které nejdou vypnout jsou *domů* a *zákazníci*. Ty vypínatelné jsou *objednávky*, *sklad*, *kalendář* a *statistiky*. Přičemž když si uživatel vypne funkci skladu, tak se zároveň s tím vypne i funkce objednávek. Bez produktů je potom tato funkce bezpředmětná. Nikoli však naopak.

Další věc, kterou nabízí obecné nastavení, je upravení informací o účtě uživatele. Lze si zde změnit heslo a e-mail.

4.2.2 Nastavení domovské záložky

Na domovské záložce se pouze nastavuje, který přehled z aplikace se bude zobrazovat. Na výběr je prozatím z objednávek k vyřízení a upomínek. V prvotním nastavení aplikace jsou vybrané objednávky k vyřízení.

4.2.3 Nastavení záložky o zákaznících

V tomto nastavení se přizpůsobuje profil zákazníka. První, co si může uživatel nastavit, jsou informace o zákazníkovi v jeho profilu. Může si zde přidávat a odebírat pole a měnit jejich pořadí. Jediná pole, která odebrat nejde, jsou *jméno*, *příjmení*, *telefon* a *poznámky*. Ty v přehledu informací musí být vždy.

Profil každého zákazníka zahrnuje v prvotním nastavení přehled historie událostí, objednávek a upomínek s daným zákazníkem spojených. V nastavení si tyto záložky v profilu může také vypnout. Pokud uživatel vypne záložku upomínek, vypne tuto funkci úplně.

4.2.4 Nastavení záložky o objednávkách

U objednávek si uživatel může vypnout funkci předpřipravené objednávky. Potom ovšem ztratí možnost si při objednávce zákazníka připravit produkty, které nejsou na skladě, k objednání.

Dále se v nastavení objednávek vypíná a zapíná záložka *objednávky k vyřízení*.

4.3 Domovská stránka

Po přihlášení do aplikace se vždy jako první uživateli zobrazí stránka *domů*. Na té se nachází seznam objednávek k vyřízení. Jedná se ovšem pouze o přehled, nelze tedy v daném přehledu nic vytvořit ani zobrazit detail.

4.4 Zákazníci

V přehledu zákazníků se zobrazují pouze tyto informace – *jméno*, *příjmení*, *telefon* a *poznámky*. Více informací o zákazníkovi a přehledy o historii s ním spojené uživatel nalezne v profilu zákazníka. Zobrazené přehledy jsou *objednávky* a *upomínky*.

Pro vytváření upomínek jsou přednastavené důvody – *narozeniny*, *kontrola* a *kontaktovat*.

4.5 Objednávky

Objednávky obsahují záložky *objednávky zákazníků* a *objednávky k vyřízení*. U objednávek jednotlivých zákazníků se v aplikaci eviduje jejich stav. Ten

může být buď *doručeno*, *doručit* nebo *objednat* (podle toho, zda byla objednávka již doručena zákazníkovi).

Ještě nedoručené objednávky se zobrazují v poslední záložce – *objednávky k vyřízení*.

4.6 Sklad

Seznam všech produktů včetně cen se nachází v záložce *sklad*. U jednotlivých produktů se také eviduje počet kusů na skladě. To si uživatel může nastavit buď při vytvoření produktu a nebo v detailu produktu.

Průměrná výdrž produktu se eviduje v měsících.

Závěr

Cílem práce bylo navrhnout a vytvořit prototyp systému pro podporu přímého prodeje na základě analýzy požadavků a existujících aplikací. Ve vytvořeném prototypu jsem úspěšně implementovala základní funkce systému. Aplikaci jsem navrhla tak, aby byla na rozdíl od konkurenčních aplikací uživatelsky co nejvíce přívětivá. Navíc je zdarma a open source.

Implementace je umístěná v repozitáři na školním gitlabu⁸ jako open source aplikace přístupná veřejnosti. Do budoucna dokončím zbylé funkce systému, které jsem zmínila ve zhodnocení prototypu, aby byla aplikace kompletní a reálně použitelná.

⁸viz https://gitlab.fit.cvut.cz/zivnuter/dss_prototype.git

Literatura

- [1] SMIT Services s.r.o.: Co je CRM systém? [online]. [cit. 2018-04-18]. Dostupné z: <https://www.vyber-crm.cz/co-je-crm-system>
- [2] PM Consulting: Funkční požadavky [online]. [cit. 2018-04-18]. Dostupné z: <http://www.pmconsulting.cz/slovníkový-pojem/funkční-požadavky/>
- [3] PM Consulting: Nefunkční požadavky [online]. [cit. 2018-04-18]. Dostupné z: <http://www.pmconsulting.cz/slovníkový-pojem/nefunkční-požadavky/>
- [4] Čápka, D.: 2. díl – UML – Use Case Diagram [online]. [cit. 2018-04-18]. Dostupné z: <https://www.itnetwork.cz/navrh/uml/uml-use-case-diagram>
- [5] Škorníčková, E.: Co je to GDPR? [online]. [cit. 2018-04-18]. Dostupné z: <https://gdpr.cz/gdpr>
- [6] Škorníčková, E.: Jaké zásadní změny GDPR přinese [online]. [cit. 2018-04-18]. Dostupné z: <https://gdpr.cz/gdpr/zmeny>
- [7] Čápka, D.: 1. díl – Úvod do JavaFX [online]. [cit. 2018-05-07]. Dostupné z: <https://www.itnetwork.cz/java/javafx/java-tutorial-uvod-do-javafx>
- [8] Čermák, M.: Vícevrstvá architektura: popis vrstev [online]. [cit. 2018-05-07]. Dostupné z: <https://www.cleverandsmart.cz/vicestvava-architektura-popis-vrstev/>
- [9] widskills.com: 01 – Introduction to JDBC [online]. [cit. 2018-05-07]. Dostupné z: <http://www.widskills.com/jdbc-tutorial/introduction-to-jdbc>

LITERATURA

- [10] Jenkov, J.: The DAO Design Pattern [online]. [cit. 2018-05-10]. Dostupné z: <http://tutorials.jenkov.com/java-persistence/dao-design-pattern.html>
- [11] ANNECA s.r.o.: InTouch CRM [online]. [cit. 2018-05-10]. Dostupné z: <http://www.intouch-crm.cz/cz/>
- [12] BPM Online: BPM Online [online]. [cit. 2018-05-10]. Dostupné z: <http://www.bpmonline.cz/trial?product=en-sales-enterprise>
- [13] FollowFox: FollowFox [online]. [cit. 2018-05-10]. Dostupné z: <https://followfox.com/>
- [14] 1CRM Systems Corp: 1CRM [online]. [cit. 2018-05-10]. Dostupné z: <https://1crm.com/>

Testovací scénáře

A.1 Vytvoření účtu

Krok: Poprvé spusťte aplikaci. Účet ještě nebyl vytvořen.

Očekávaný výsledek: Zobrazí se formulář pro vytvoření účtu. Ve formuláři jsou pole pro zadání e-mailu a hesla a tlačítko pro vytvoření.

Krok: Nechte pole nevyplněná a klikněte na vytvořit.

Očekávaný výsledek: Zobrazí se hláška o tom, že musíte vyplnit e-mail i heslo.

Krok: Vyplňte validní e-mail a heslo nechte nevyplněné. Formulář potvrďte.

Očekávaný výsledek: Zobrazí se hláška o tom, že musíte vyplnit e-mail i heslo.

Krok: Vyplňte heslo a e-mail nechte nevyplněný. Formulář potvrďte.

Očekávaný výsledek: Zobrazí se hláška o tom, že musíte vyplnit e-mail i heslo.

Krok: Vyplňte nevalidní e-mail a heslo. Vyzkoušejte různé varianty nevalidního e-mailu.

Očekávaný výsledek: Zobrazí se hláška o tom, že musíte zadat validní e-mail.

Krok: Zadejte validní e-mail a heslo a klikněte na vytvořit.

Očekávaný výsledek: Zobrazí se hláška o vytvoření účtu a formulář k přihlášení.

A.2 Přihlášení

Krok: Spusťte aplikaci. Účet již musí být vytvořen.

Očekávaný výsledek: Zobrazí se formulář k přihlášení. Ve formuláři jsou pole pro zadání e-mailu a hesla a tlačítko pro přihlášení.

Krok: Nechte obě pole nevyplněná a klikněte na přihlášení.

Očekávaný výsledek: Zobrazí se hláška o tom, že musíte vyplnit správný e-mail i heslo.

Krok: Vyplňte správný e-mail a heslo nechte nevyplněné. Formulář potvrďte. Poté to zkuste i naopak.

Očekávaný výsledek: Zobrazí se hláška o tom, že musíte vyplnit správný e-mail i heslo.

Krok: Vyplňte nejdříve špatný e-mail a správné heslo. Potom to vyzkoušejte naopak.

Očekávaný výsledek: Zobrazí se hláška o tom, že musíte vyplnit správný e-mail i heslo.

Krok: Vyplňte správný e-mail i heslo a formulář potvrďte.

Očekávaný výsledek: Jste přihlášen a zobrazuje se domovská stránka aplikace.

A.3 Domovská stránka

A.3.1 Přehled

Krok: Přihlašte se do aplikace. Nastavení domovské stránky je v prvotním nastavení a v databázi existují objednávky k vyřízení.

Očekávaný výsledek: Zobrazuje se domovská stránka aplikace. V přehledu jsou zobrazené objednávky k vyřízení. Na obrazovce se nenachází tlačítko vytvoření. Nahoře vpravo se nachází tlačítko nastavení.

A.4 Zákazníci

A.4.1 Přehled

Krok: Přihlašte se do aplikace. Přejděte na záložku zákazníků. V databázi jsou již vytvoření nějací zákazníci.

Očekávaný výsledek: Zobrazuje se stránka zákazníků. V přehledu jsou zobrazení existující zákazníci. Nahoře vpravo se nachází tlačítko vytvoření a napravo od něj tlačítko nastavení.

A.4.2 Vytvoření zákazníka

Krok: Přejděte na vytvoření zákazníka. Nechte pole nevyplněná a klikněte na tlačítko vytvořit.

Očekávaný výsledek: Zobrazil se formulář vytvoření zákazníka. Zobrazí se hláška o tom, že nejsou vyplněná všechna povinná pole.

Krok: Některá povinná pole vyplňte a některá nechte prázdná. Klikněte na tlačítko vytvořit.

Očekávaný výsledek: Zobrazí se hláška o tom, že nejsou vyplněná všechna povinná pole.

Krok: Vyplňte všechna povinná pole. Klikněte na tlačítko zahodit. V zobrazeném dialogovém okně klikněte na *ne*.

Očekávaný výsledek: Dialogové okno, které informuje o ztrátě informací, vyskočí. Okno se zavře a je stále zobrazena obrazovka vytvoření zákazníka.

Krok: Vyplňte všechna povinná pole a ostatní pole vyplňte libovolně. Klikněte na tlačítko vytvořit.

Očekávaný výsledek: Zobrazí hláška o úspěšném přidání zákazníka. Zobrazuje se přehled zákazníků, ve kterém se právě přidáný zákazník nachází.

A.4.3 Detail zákazníka

Krok: Zobrazte si detail libovolného zákazníka. Stiskněte tlačítko uložit.

Očekávaný výsledek: Je otevřený detail vybraného zákazníka. Údaje souhlasí. Zobrazují se navíc záložky objednávky a upomínky. Po uložení se zobrazí hláška o tom, že neproběhla žádná změna.

Krok: Změňte libovolné hodnoty v libovolném poli zákazníka. Nechte nevyplněné povinné pole a klikněte na tlačítko uložit.

Očekávaný výsledek: Zobrazí se hláška o tom, že nejsou vyplněná všechna povinná pole.

Krok: Vyplňte znovu všechna povinná pole a klikněte na uložit. Přejděte zpět na přehled a zkontrolujte, zdali se změny provedly.

Očekávaný výsledek: Zobrazí se hláška o úspěšném uložení. Změny byly uloženy.

Krok: Zobrazte si detail libovolného zákazníka. Stiskněte tlačítko smazat.

Očekávaný výsledek: Zobrazí se dialog o úspěšném smazání zákazníka. Zákazník se nadále nezobrazuje v přehledu.

Krok: Zobrazte si detail libovolného zákazníka. Přejděte na záložku objednávky.

Očekávaný výsledek: Zobrazí se pouze ty objednávky, které jsou navázané na daného zákazníka.

Krok: Přejděte na záložku upomínky. Založte libovolnou upomínku.

Očekávaný výsledek: Zobrazí se dialog o úspěšném uložení upomínky. Právě vytvořená upomínka se zobrazuje v přehledu.

A.5 Objednávky – objednávky zákazníků

Krok: Přihlašte se do aplikace. Přejděte na záložku objednávky zákazníků. V databázi jsou již vytvořené nějaké objednávky.

Očekávaný výsledek: Zobrazuje se stránka objednávek k vyřízení. V přehledu jsou zobrazené objednávky.

Krok: Přejděte na vytvoření objednávky. Nechte jakékoliv povinné pole prázdné nebo nepřidejte žádný produkt a potvrďte formulář.

Očekávaný výsledek: Zobrazuje se stránka vytvoření objednávky zákazníka. Zobrazí se hláška o tom, že nejsou vyplněna všechna povinná pole.

Krok: Vyplňte všechna povinná pole a potvrďte formulář.

Očekávaný výsledek: Objednávka byla úspěšně vytvořena a zobrazuje se v seznamu.

A.6 Objednávky – objednávky k vyřízení

Krok: Přejděte na záložku objednávky k vyřízení. V databázi jsou již vytvořené takové objednávky, které ještě nebyly doručené.

Očekávaný výsledek: Zobrazuje se stránka objednávek k vyřízení. V přehledu jsou zobrazené pouze objednávky, které ještě nejsou ve stavu doručeno.

A.7 Sklad

A.7.1 Přehled

Krok: Přihlašte se do aplikace. Přejděte na záložku skladu. V databázi jsou již vytvořené nějaké produkty.

Očekávaný výsledek: Zobrazuje se stránka skladu. V přehledu jsou zobrazené existující produkty. Nahoře vpravo se nachází tlačítko vytvoření a napravo od něj tlačítko nastavení.

A.7.2 Vytvoření produktu

Krok: Přejděte na vytvoření produktu. Nechte pole nevyplněná a klikněte na tlačítko vytvořit.

Očekávaný výsledek: Zobrazil se formulář vytvoření produktu. Zobrazí se hláška o tom, že nejsou vyplněná všechna povinná pole.

Krok: Některá povinná pole vyplňte a některá nechte prázdná. Klikněte na tlačítko vytvořit.

Očekávaný výsledek: Zobrazí se hláška o tom, že nejsou vyplněná všechna povinná pole.

Krok: Vyplňte všechna povinná pole. Klikněte na tlačítko zahodit. V zobrazeném dialogovém okně klikněte na *ne*.

Očekávaný výsledek: Dialogové okno, které informuje o ztrátě informací, vyskočí. Okno se zavře a je stále zobrazena obrazovka vytvoření produktu.

Krok: Vyplňte všechna povinná pole a ostatní pole vyplňte libovolně. Klikněte na tlačítko vytvořit.

Očekávaný výsledek: Zobrazí se hláška o úspěšném přidání produktu. Zobrazuje se přehled produktů, ve kterém se právě přidáný produkt nachází.

A.7.3 Detail produktu

Krok: Zobrazte si detail libovolného produktu. Stiskněte tlačítko uložit.

Očekávaný výsledek: Je otevřený detail vybraného produktu. Údaje souhlasí. Po uložení se zobrazí hláška o tom, že neproběhla žádná změna.

Krok: Změňte libovolné hodnoty v libovolném poli produktu. Nechte nevyplněné povinné pole a klikněte na tlačítko uložit.

Očekávaný výsledek: Zobrazí se hláška o tom, že nejsou vyplněná všechna povinná pole.

Krok: Vyplňte znovu všechna povinná pole a klikněte na uložit. Přejděte zpět na přehled a zkontrolujte, zdali se změny provedly.

Očekávaný výsledek: Zobrazí se hláška o úspěšném uložení. Změny byly uloženy.

Krok: Zobrazte si detail libovolného produktu. Stiskněte tlačítko smazat.

Očekávaný výsledek: Zobrazí se dialog o úspěšném smazání produktu. Produkt se nadále nezobrazuje v přehledu.

Seznam použitých zkratk

CRM Customer Relationship Management

GDPR General Data Protection Regulation

GUI Graphical user interface

CSS Cascading Style Sheets

MVC Model-View-Controller

JDBC Java Database Connectivity

SQL Structured Query Language

API Application Programming Interface

Obsah přiloženého CD

	readme.txt.....	stručný popis obsahu CD
	src	
	impl.....	zdrojové kódy implementace
	thesis.....	zdrojová forma práce ve formátu \LaTeX
	text.....	text práce
	thesis.pdf.....	text práce ve formátu PDF