

Vysoká škola ekonomická v Praze
Fakulta informatiky a statistiky



**Analýza a implementace Salesforce
CRM s integrací na účetní systém pro
střední firmu nebo neziskovou
organizaci**

Bakalářská práce

Studijní program: Aplikovaná Informatika

Studijní obor: Aplikovaná informatika

Autor: Tomáš Chour

Vedoucí práce: prof. Ing. Alena Buchalcevová, Ph.D.

Praha, Duben 2021

Prohlášení

Prohlašuji, že jsem bakalářskou práci Analýza a implementace Salesforce CRM s integrací na účetní systém pro střední firmu nebo neziskovou organizaci vypracoval samostatně za použití v práci uvedených pramenů a literatury.

V Praze dne 9. května 2021

.....
Tomáš Chour

Abstrakt

Tato bakalářská práce se zabývá analýzou integrace dat mezi systémy Salesforce CRM a ABRA Flexi a návrhem obecně použitelné aplikace pro dynamické nastavení mapování dat mezi systémy, vypínání synchronizace podle jednotlivých entit a parametrizace integrace skrze uživatelské rozhraní. Aplikace byla navržena a realizována jako open-source pomocí standardních nástrojů Salesforce platformy s ohledem na budoucí rozšíření a snadné distribuce pomocí Salesforce SFDX.

Klíčová slova

Salesforce integrace dat, ABRA Flexi integrace dat, SFDX, Salesforce open-source aplikace

Abstract

This bachelor thesis deals with the analysis of data integration between Salesforce CRM and ABRA Flexi systems and the design of a generally applicable application for dynamic setting of data mappings between both systems. It also provides functionality for disabling synchronization according to individual entities and parameterization of integration through the user interface. The application was designed and implemented as open-source using standard Salesforce platform tools with a future extensions and ease of distribution through Salesforce SFDX in mind.

Keywords

Salesforce data integration, ABRA Flexi data integration, SFDX, Salesforce open-souce application

Obsah

	Strana
Úvod	11
Cíle práce	11
Metoda dosažení cílů	12
Očekávané přínosy	12
Omezení	12
Výstupy	12
1 Rešerše zdrojů	13
1.1 Odborné knihy a akademické práce	13
1.2 Internetové zdroje a odborné články	13
2 Teoretická východiska práce	15
2.1 Informační systém	15
2.2 Salesforce CRM	15
2.2.1 Datový model Salesforce CRM	16
2.2.2 Vývoj na platformě Salesforce	16
2.2.3 Customizace a nasazení do prostředí	19
2.3 Abra Flexibee	19
2.3.1 REST API	20
2.3.2 WebHooks	20
2.4 Datová integrace v informačním systému	20
2.4.1 Typy datové integrace podle práce s daty	21
2.4.2 Typy datové integrace podle spouštěče	21
2.4.3 Způsoby integrace dat	22
2.4.4 Autentizace a autorizace	23

3	Analýza požadavků	24
3.1	Funkční požadavky	24
3.2	Nefunkční požadavky	25
3.3	Případy užití a role aktérů	26
3.3.1	Role aktérů	26
3.3.2	Případy užití	27
4	Analýza datových modelů	33
4.1	Datový model Salesforce CRM	33
4.2	Datový model ABRA Flexibee	33
4.2.1	Datové schéma entit	33
4.2.2	Objednávky a faktury	33
4.3	Ukládání mapování a parametrizace	34
4.3.1	Struktura pro mapování dat mezi systémy	34
4.3.2	Struktura pro ukládání parametrizace	35
4.4	Propojení datových modelů a kustomizace	36
4.4.1	Externí identifikátor a příznak pro export	36
4.4.2	Faktury	36
4.4.3	Mapování entit	36
4.4.4	Entita pro aplikační logování	37
4.4.5	Výsledný datový model Salesforce CRM	38
5	Uživatelské skupiny a práva	39
6	Datová integrace	40
6.1	Diagram tříd	40
6.2	Automatizace datové integrace	42
6.3	Asynchronní hromadný export	43
6.4	Struktura přenášených dat	44
6.5	Autentizace	45
6.6	Rozšíření o podporu dalších entit	45

7	Uživatelská aplikace	47
7.1	Abra Integration Settings	47
7.1.1	Komponenta pro mapování entit	47
7.1.2	Komponenta pro parametrizaci aplikace	49
7.2	Aplikační logy	50
7.3	Lokalizace	51
8	Testování	52
8.1	Jednotkové testy	52
8.2	Uživatelské testy	52
8.2.1	Testovací scénáře	52
8.3	Výsledky testů	55
9	Distribuce	56
10	Závěr	57
	Seznam použité literatury	58

Seznam tabulek

1	Hlavní limity pro operace v Salesforce (Zdroj: (Salesforce, Execution Governors and Limits)	19
2	Přehled požadavků	24
3	Případ užití - Zobrazení nastavení aplikace	28
4	Případ užití - Zobrazení aplikačních logů	29
5	Případ užití - Zadaní mapování atributů	30
6	Případ užití - Parametrizace aplikace	30
7	Případ užití - Synchronizace dat	31
8	Případ užití - Manuální spuštění synchronizace dat	32
9	ABRA Flexi API názvy entit pro objednávky a faktury	34
10	Specifikace ABRA Flexi Data Mapping metada atributů	35
11	Specifikace ABRA Flexi Data Integration Settings custom settings atributů	35
12	Mapování objednávky na objednavka-prijata	36
13	Mapování faktury na faktura-vydana	37
14	Specifikace atributů Application Log SObjectu	37
15	Přehled oprávnění podle skupin uživatelů	39
16	1TC kroky	53
17	2TC kroky	53
18	3TC kroky	54
19	4TC kroky	55

Seznam zkratek

SFSC - Salesforce Sales/Service Cloud

CRM - customer relationship management

API - application programming interface

SSOT - single source of truth

LEX - Lightning Experience

LWC - Lightning web components

SaaS - Software as a service

PaaS - Platform as a service

Úvod

V posledních letech v České republice výrazně narůstá počet implementací CRM cloudových služeb (ČSÚ, 2016; ČSÚ, 2019; ČSÚ, 2021). Největším cloudovým CRM řešením na světě je Salesforce CRM od Salesforce.com s podílem na trhu 19,5% (Gartner, 2019). Jednou velkou kapitolou během vývojové fáze implementace je integrace na ostatní systémy, které daná firma pro svou činnost a provoz používá. Integrace s ostatními systémy bývá náročná na analýzu i vývoj, protože musí běžet bezchybně, zvládat výkyvy v objemu dat i dočasné výpadky jednoho nebo obou systémů a zajistit bezpečnost přenosu. Jedna z nejčastějších integrací bývá na účetní systém.

Další obsáhlou fází implementace je samotné nasazení celého systému nebo jeho části. Běžně dostupné komponenty systému by měly být snadno nasaditelné ideálně pomocí jedné akce, případně přesně popsaného souboru navazujících akcí. Komponenty systému, které vyžadují velké množství dalších částí třetích stran jsou složitější na nasazení, než komponenty, které využívají pouze standardní funkcionality a technologie daného systému. Ideálním stavem pak je, když můžeme do systému přidávat komponenty bez nutnosti zásahu do stávající architektury. (Martin, 2018)

Pro to, aby informační systém dokázal pružně a rychle reagovat na změny v jeho okolí - změny ve vstupech a výstupech či v legislativě, je potřeba, aby s těmito změnami dokázaly pracovat jeho jednotlivé komponenty. Pokud firma používá řešení, které si sama vyvinula, nebo používá proprietární software od svého dodavatele, je za podmínky, že dojde ke změně okolí jejího IS, v prvním případě odkázána na své kapacity, v druhém případě na kapacity svého dodavatele, aby dané změny zapracoval do systému a jeho konkrétní komponenty. V případě veřejně využívaného open source řešení je možnost zvýšit flexibilitu a snížit náklady díky komunitnímu vývoji. (VonFange a Lavigne 2011)

Z důvodů popsaných v předchozích odstavcích jsem se rozhodl vytvořit open source Salesforce aplikaci, která umožní jednoduchou konfiguraci integrace dat mezi Salesforce a ABRA Flexibee.

Cíle práce

Cílem bakalářské práce je definovat požadavky pro obecně použitelnou integraci Salesforce Sales Cloud (dále jen SFSC) s účetním systémem ABRA Flexibee, realizovat ji a testováním ověřit.

Dílní cíle práce:

- DC1: Analyzovat data, která účetní systém potřebuje získávat.
- DC2: Zanalyzovat a popsat část standardního datového modelu SFSC obsahujícího data potřebná pro účetní systém.
- DC3: Navrhnout a realizovat datový model pro data, která nejsou pokryta ve standardním modelu.
- DC4: Navrhnout a realizovat webové služby pro integraci dat mezi systémy. Automatizovat integraci pomocí vhodných nástrojů.
- DC5: Vytvořit SFSC aplikaci s uživatelským rozhraním pro konfiguraci integrace.
- DC6: Ověřit funkčnost pomocí testovacích dat a regresních testů

Metoda dosažení cílů

Pro zpracování teoretické části práce byla provedena rešerše zdrojů, které jsou relevantní pro návrh a implementaci informačních systémů a architekturu čitelného a škálovatelného software. V praktické části jsem analyzoval datový model a webové služby systému ABRA Flexibee, poté analyzoval část datového modelu SFSC, která je relevantní pro účetní systém, pomocí modelování zmapoval uchovávaná data v jednotlivých systémech a poté na základě komparace těchto modelů namapoval přenos dat mezi systémy. Díky komparaci modelů je také zjištěno, jestli je potřeba datový model SFSC rozšířit. Samotný vývoj webových služeb, uživatelské aplikace a vytvoření testů je realizováno pomocí vlastních zkušeností a využitím získaných poznatků a best practices z teoretické části.

Očekávané přínosy

Přínosem práce má být zefektivnění a zlevnění vývojové fáze implementace SFSC s integrací na účetní systém ABRA Flexibee. Po úpravách zdrojového kódu lze tuto práci použít i pro integraci s jinými podobnými účetními systémy. Z důvodu využití standardních SFSC objektů a automatizačních nástrojů lze toto řešení implementovat bez dodatečných úprav v organizacích SFSC, které využívají standardní datový model.

Omezení

Z důvodu využití standardních objektů nelze bez nutnosti změn v mapování implementovat pro organizace SFSC, které využívají nestandardní datový model. Ovládání skrze custom UI je omezeno pouze na Salesforce Lightning, použití v Classic není podporováno, i v něm ale lze využít datový model, automatizace a webové služby.

Výstupy

Hlavním výstupem práce je aplikace pro SFSC, která umožňuje jednoduchou konfiguraci integrace SFSC na ABRA Flexibee skrze UI. Aplikace je veřejně dostupná jako open source metadata a zdrojový kód ve verzovacím systému na platformě GitHub. Metadata a zdrojový jsou uloženy ve formátu pro práci se SFDX.

1 Rešerše zdrojů

Tato kapitola má za cíl zmapovat a prostudovat relevantní zdroje k práci. Integrace dat mezi systémy je obsáhlé téma, které čerpá znalosti z mnoha částí IT, proto zde jsou uvedeny zdroje, které se týkají architektury a návrhu softwaru a informačních systémů, bezpečnosti webových služeb a vývoji na platformě Salesforce. Platforma Salesforce i systém ABRA Flexibee mají na webu veřejně dostupné podrobné dokumentace. Salesforce navíc nabízí komunitní fóra, kde je možné dohledat konkrétní specifické nebo výjimečné problémy, které nejsou popsány v dokumentaci a již se s nimi někdo jiný setkal.

1.1 Odborné knihy a akademické práce

Kniha *Tvorba informačních systémů: principy, metodiky, architektury* (Bruckner et al., 2012) dává dohromady poznatky v analýze a návrhu podnikových informačních systémů. Nabízí seznam metodik a způsobu jejich uplatnění při řešení jednotlivých částí analýzy a návrhu IS. Dále také shrnuje jednotlivé notace pro modelování částí IS.

Baklářská práce *Continuous Delivery on the Salesforce Platform* (Kratochvíl, 2019) se zabývá vývojem a jeho automatizací na cloudové platformě Salesforce s využitím verzovacího systému a vytvořených Javovských scriptů. Popisuje technologii SFDX, Salesforce metadata a způsob práce s nimi. Z této práce je hlavně důležitá část, která se zabývá práce s metadaty, jejich exportem z vývojového prostředí, ukládáním a migrací do konkrétních Salesforce organizací.

Další kniha *Clean architecture: a craftsman's guide to software structure and design* (Martin, 2018) je teoretický souhrn z oblasti architektury a designu softwaru pro jeho jednoduchou čitelnost a škálovatelnost. Pokrývá jak funkční, tak objektově orientované programování. Zabývá se vhodností použití dědičnosti, rozhraní a polymorphismem v určitých situacích, jak navrhovat strukturu dat pro přenos v rámci systému a mimo něj. Popisuje hlavní best practices pro škálovatelnost softwaru a jeho komponent.

Kniha *Requirements Writing for System Engineering* (Koelsch, 2016) pokrývá oblast analýzy požadavků. Seznamuje s principy, jak sesbírat všechny funkční i nefunkční požadavky, které bude v projektu potřeba brát na vědomí a implementovat, jejich znázornění pomocí modelování a převedení na use cases a user stories pro agilní vývoj.

1.2 Internetové zdroje a odborné články

ABRA poskytuje ke svému produktu Flexibee přehlednou dokumentaci *Dokumentace REST API: Dokumentace a referenční příručka FlexiBee REST API* (Abra, nedatováno), ze které lze vycházet při samotné implementaci integrace dat. Lze z ní vyčíst jednotlivé webové služby pro import a export dat a samotnou strukturu dat pro komunikaci mezi systémy.

Na straně Salesforce také existuje rozsáhlá dokumentace *Salesforce Developer Documentation* (Salesforce, nedatováno), která pokrývá oblasti datového modelování, psaní kódu pomocí jazyku Apex a vytváření uživatelského rozhraní pomocí LWC a standardních salesforce nástrojů a komponent. Oblasti, které nepokrývá dokumentace lze dohledat na Salesforce vývojářských fórech <https://trailblazers.salesforce.com/answers>,

ovšem zde je nutno být opatrný a skeptický, jelikož se nejedná o oficiální řešení problému a je nutné si navrhovaný postup ověřit i z jiného zdroje.

On the Need for a General REST-Security Framework (Iacono, Gorski, 2019) analyzuje bezpečnostní slabiny architektury rozhraní REST, uvádí jeho slabiny a navrhuje metodiku, jak zabezpečit služby, které jsou na REST založeny. Dále také porovnává jednotlivé metody autentizace webových služeb.

Pro oblast designu UI vycházím hlavně ze standardů *Salesforce Lightning Design System* (Salesforce Lightning Design System, nedatováno), kde jsou k dispozici oficiální nástroje, návody a blueprinty pro návrh a implementaci UI v prostředí Salesforce Lightning.

2 Teoretická východiska práce

Účelem této kapitoly práce je popsat teoretické podklady a vysvětlit základní termíny, na jejichž základě bude implementováno řešení v praktické části. První část se zabývá informačními systémy Salesforce CRM a ABRA Flexibee, jejich představením a základními charakteristikami. Další části se zabývají integrací dat mezi systémy, způsoby a technologiemi, kterými ji lze docílit.

2.1 Informační systém

Informační systém je soubor technologií, lidí, dat a prostředků jehož cílem je zajištění správných informací ve správný čas na správném místě. Správným místem je myšlena osoba, jiný systém nebo jiná část systému, která tu konkrétní informaci využívá k podpoře svých rozhodovacích a řídicích procesů. (Bruckner et al., 2012)

Systém pro řízení vztahů se zákazníky (CRM z anglického názvu Customer Relationship Management) je informační systém pro podporu všech procesů, které zahrnují zákazníky podniku. Tím jsou myšleny procesy pro získávání nových zákazníků a poskytování služeb těm stávajícím.

Hlavními úkoly ekonomického informačního systému je evidence dokladů spojených s účetnictvím, zpracování a výpočet mezd a výkaznictví. Měly by pracovat s aktuálními právními předpisy, převážně se zákonem o účetnictví - Zákon č. 563/1991 Sb.

2.2 Salesforce CRM

Salesforce je nejrozšířenější cloudová platforma na světě, zaměřená na zajištění procesů spjatých s CRM. (Gartner, 2019) Skládá se z několika modulů, které si každý může uživatel nakombinovat tak, aby jimi zajistil své konkrétní procesy.

Salesforce je SaaS¹ i PaaS² zároveň, jelikož nabízí předem hotová řešení - viz. Sales Cloud, Marketing Cloud a další, tak umožňuje uživatelům vyvíjet své vlastní aplikace, což je i předmětem této práce.

Hlavními moduly jsou (Salesforce EMEA, Customer 360):

- Sales Cloud - Zahrnuje podporu procesů spjatých s managementem obchodních příležitostí a získáváním nových leadů
- Service Cloud - Podporuje procesy pro řízení zákaznického servisu - dotazy, reklamace, stížnosti, zpětná vazba apod.
- Marketing Cloud - Nástroj pro podporu procesů spojených s marketingem - automatizování newsletterů, analýza marketingových kampaní, tvorba cíleného obsahu na určité skupiny zákazníků

¹SaaS: Model distribuce a licencování aplikačního softwaru, kdy poskytovatel cloudu spravuje aplikační software a zpřístupňuje jej svým zákazníkům skrze internet na bázi průběžných plateb (subskripce). (Salesforce, What is SaaS?)

²PaaS: Soubor služeb a nástrojů, které poskytovatel cloudu nabízí svým zákazníkům a vývojářům, kteří pomocí nich a s využitím infrastruktury poskytovatele, který ji spravuje, vyvíjet nové aplikace. (Salesforce, What is PaaS?)

- Commerce Cloud - Platforma pro vytváření a správu škálovatelných eshopů, skládá se ze samostatných B2B a B2C částí.
- Analytics - Nástroj pro BI
- Experience Cloud - Nástroj, který umožňuje vystavit funkcionalitu se kterou mohou přímo interagovat zákazníci a dodavatelé společnosti - typicky self-service služby.
- a další

2.2.1 Datový model Salesforce CRM

Salesforce poskytuje hotový datový model pro podporu procesů jednotlivých modulů. Tento předem hotový datový model se nazývá standardní.

Standardní datový model pro Sales a Service Cloud zahrnuje entity (SObject v Salesforce názvosloví) pro správu zákazníků a kontaktů, objednávek a faktur, obchodních příležitostí, uživatelů, dokumentů a spoustu dalších - všechny lze nalézt v dokumentaci : https://developer.salesforce.com/docs/atlas.en-us.api.meta/api/data_model.htm. (Salesforce, *Salesforce Developer Documentation*)

Datový model, který vytvoří uživatel se nazývá custom (od custom made), uživatel může rozšířit standardní entity o jednotlivé atributy (pole SObjectu v Salesforce názvosloví), nebo vytvářet celé entity včetně relací na standardní i custom datový model. V salesforce existují dva způsoby relací - lookup a master-detail.

Lookup relace

Lookup pole umožňuje nadefinovat relaci mezi dvěma SObjecty, funguje na principu cizího klíče, relace na rodiče není povinná - může být null. (Salesforce, *Salesforce Developer Documentation*)

Master-Detail relace

Master-Detail je striktnější formou relace, která poskytuje funkcionalitu navíc. Umožňuje na rodiči vytvářet tzv. roll-up summary fields, které poskytují kalkulaci SUM, AVG, MIN, MAX z dceřiných záznamů. Master-Detail ovlivňuje i CRUD práva na dceřinnou entitu - přístup na rodiče uerčuje přístup na dceřinnou entitu. Používá se pro vytváření M:N relací skrze junction tabulku. (Salesforce, *Salesforce Developer Documentation*)

2.2.2 Vývoj na platformě Salesforce

Následující kapitola se zabývá způsobem vývoje a vývojářskými nástroji, které Salesforce poskytuje. Popisují zde nástroje pro tradiční i low-code vývoj, kapitola předpokládá základní znalosti objektového programování a algoritmizace.

APEX

APEX je objektově orientovaný programovací jazyk, který byl vyvinut přímo pro platformu Salesforce. Má syntaxi podobné jazyku Java a využívá se pro psaní logiky všech server - side operací. Zahrnuje v

sobě podporů pro práci s databází, jako je psaní DML operací (insert, update, delete, select, upsert) a triggerů včetně jejich logiky. Dále se pomocí něj také pší webové služby a API callouty. (Salesforce, *Salesforce Developer Documentation*)

Automatizační nástroje

Salesforce nabízí svým uživatelům širokou škálu automatizačních nástrojů, jejichž spouštěčem může být událost nad záznamem v databázi nebo mohou být spouštěny pravidelně v nějakém časovém intervalu.

Dostupnými automatizačními nástroji jsou:

- Apex - Umožňuje psaní after a before triggerů nad objekty (insert, update, delete undelete) a časově spustitelných událostí.
- Flow - komplexní grafický automatizační nástroj, který se skládá z mnoha jednotlivých komponent, umožňuje kombinaci s apexem
- Process Builder - jednoduchý grafický nástroj, funguje na principu několika na sebe navazujících if / then tvrzení
- Workflow - jednoduchý automatizační nástroj, používající pouze jedno if / then tvrzení

Obrázek 1 znázorňuje porovnání automatizačních nástrojů podle jednoduchosti na použití, škály případů použití, možnost narazení na limity a schopnost je obejít, škálovatelnosti, schopnosti daný nástroj debugovat, uživatelské přívětivosti v případě chyby, možnosti testování daného nástroje, rychlosti exekuce, jednoduchosti nasazení a jak moc je nástroj odolný proti riziku, že kvůli němu rozbijem již existující funkcionalitu.



Obrázek 1: Porovnání automatizačních nástrojů (Zdroj: Liu, 2018)

Omezení

Jelikož Salesforce organizace je součástí multitenantního prostředí - skupina organizací sdílí zdroje jednoho serveru (instance) tak, aby nedocházelo k výkyvům výkonu, Salesforce vynucuje limity pro operace v rámci jedné transakce, které čerpají serverové zdroje, které nesmějí jednotlivé organizace překročit. Operace, která takový limit překročí, je ukončena výjimkou. Při psaní kódu a vytváření automatizací logiky je tedy potřeba aby byla daná operace co nejvíce optimalizovaná. Některé limity se liší pro synchronní a asynchronní operace.

Hlavní limity platformy Salesforce jsou uvedeny v tabulce 1.

Tabulka 1: Hlavní limity pro operace v Salesforce (Zdroj: (Salesforce, Execution Governors and Limits))

Omezení	Synchronní	Asynchronní
Maximální počet záznamů načtených pomocí SOQL dotazů	50 000	
Maximální počet DML operací (vyjma selectů)	150	
Maximální počet HTTP volání	100	
Maximální kumulativní čas pro všechny HTTP volání	120 s	
Maximální čas využití CPU serveru	10 000 ms	60 000 ms
Maximální velikost haldy	6 MB	12 MB
Maximální počet SOQL dotazů	100	200

2.2.3 Customizace a nasazení do prostředí

Všechny součásti systému (datový model, profily, trigger, automatizace, UI, webové služby a další) jsou v Salesforce reprezentovány soubory metadat. Vytváření některých souborů metadat je, aby se předešlo poškození současného řešení, v produkčním prostředí zakázáno a proto musí být tato metadata vytvořena na nižším prostředí (sandbox, developer edition) a následně zmigrována do produkce. Existuje několik základních způsobů nasazení: (Kratochvíl, 2019)

- Change set - Jde o přímé přetažení dat mezi dvěma propojenými prostředími bez nutnosti vytváření kopie metadat na lokálním zařízení. Change set se vytváří pomocí deklarativního nástroje, který je součástí Salesforce. Nevýhodou tohoto řešení je absence verzovacího systému a absence SSOT při použití více vývojových prostředí.
- Unmanaged package - Je balíček metadat, který je vytvořený v libovolném prostředí a může být nainstalován na kterékoliv nepropojené prostředí, používá se k distribuci open source funkcionalit.
- Managed package - Podobný jako unmanaged package s tím rozdílem, že jde o closed source, po nainstalování balíčku, nemá uživatel možnost zobrazit ani upravovat daná metadata. Používá se pro distribuci placených funkcionalit nezávislými vývojářskými společnostmi nebo Salesforce partnery.
- Metadata API - Rozhraní, které umožňuje CRUD nad metadata soubory v daném prostředí, na rozdíl od change setů umožňuje použít jako zdroj souborů i jiné možnosti než Salesforce organizaci - např. lokální strom souborů, nebo verzovací systém. K využití Metadata API se používají další nástroje, které soubory zpracují a odešlou HTTP request - např. Ant Migration Tool nebo Salesforce CLI. Výhodou využití takového middlewaru je to, že pro práci s metadaty můžeme využít formát SFDX, který je pro člověka lépe čitelný a snadněji se v něm sledují změny oproti klasickému Metadata formátu. (Kratochvíl, 2019)

2.3 Abra Flexibee

Abra FlexiBee je ekonomický systém od firmy ABRA Flexi s.r.o., který funguje v cloudu i on-premise s tenkým klientem. Nabízí moderní přehledné prostředí se zpracováním dat do dashboardů. Je multiplatformní, podporuje webový přístup z webových prohlížečů, klientský přístup z macOS, Windows a Linux i mobilní přístup z Android a iOS zařízení. (ABRA FLEXI S.R.O., Abra Flexi)

V současnosti používá FlexiBee přes 6500 zákazníků. (System Online, Abra FlexiBee)

2.3.1 REST API

FlexiBee obsahuje rozhraní REST API pro integraci s externími informačními systémy, poskytuje pro něj přehlednou dokumentaci, která je dostupná na : <https://intercom.help/podpora-flexi/cs/collections/2592813-dokumentace-rest-api> . Rozhraní podporuje všechny CRUD operace a základním formátem dat pro komunikaci je JSON a XML, mimo to dokáže importovat a exportovat data i v CSV a XLS.

2.3.2 WebHooks

WebHooks je způsob automatického propojení systému / aplikace (publisher) s několika externími systémy / aplikacemi (subscribers). Při použití WebHooks se zpráva posílá z publisher systému při nějaké události v něm (např. změna záznamu v databázi nebo vnitřní event) na všechny zaregistrované subscriber URL. (Guay, 2020)

FlexiBee má možnost implementace WebHooks, kdy se HTTP POST request posílá při změně obsahu databáze ABRA Flexi a obsahem jeho zprávy je výpis všech změn od posledního zavolání hooku. Pokud nastane chyba při doručování zprávy, je zpráva odesílána opakovaně v postupně se prodlužujících intervalech. Aby FlexiBee považovalo zprávu za úspěšně doručenou, nesmí její zpracování v cílovém systému přesáhnout 30 vteřin, respektive musí do 30 vteřin subscriber odpovědět se status kódem 2xx. Ideální implementací je pak, že subscriber odpoví úspěšným statusem 2xx okamžitě poté, co obdrží zprávu se změnami a samotné jejich zpracování probíhá asynchronně. (Abra, Dokumentace REST API).

V případě výše zmíněného navrhovaného způsobu implementace WebHooks podotýkám, že je nutné ošetřit případ, kdy subscriber odpoví úspěšným statusem a pak dojde k chybě v asynchronním zpracování dat. V takovém případě je nutné, aby se daný systém dotázal dodatečně na data, která nedokázal zpracovat, zalogoval chybu a takový záznam označil, pokud je ani po dodatečném dotazu nelze zpracovat, aby nebyla porušována integrita dat mezi systémy.

2.4 Datová integrace v informačním systému

Cílem této kapitoly je prozkoumat a zmapovat způsoby integrace dat mezi komponentami informačního systému podniku. Druhá část kapitoly se zabývá zabezpečením přenosu a přístupu k datům.

„Integrace softwarového produktu je proces, kterým se softwarový produkt propojuje s ostatními softwarovými komponentami informačního systému podniku (sdílení společných dat, využívání funkcionality komponenty jinou komponentou apod.).“ (Bruckner et al., 2012)

2.4.1 Typy datové integrace podle práce s daty

Konsolidace

Konsolidace dat je způsob integrace, kdy data získaná z několika různých zdrojů, jsou převedena do jednotného formátu, očištěna od duplicit a nahrána do jednoho zdrojového systému / databáze. Cílem konsolidace je zredukovat počet různých míst, na kterém jsou data uložena a tím snížit nároky na úložný prostor a údržbu jejich integrity. Datová konsolidace využívá nástrojů pro ETL (extract, transform, load). (Foote, 2019)

Propagace

Propagace dat se používá k přenesení dat z jednoho systému (nebo jeho komponenty) do jiného, který tato data využívá k zajištění svých procesů. Může být řešena synchronně i asynchronně. (Foote, 2019) Často se používá v případě, že jako uživatel zdrojového systému potřebuji získat near-realtime odpověď na základě zpracovaných odeslaných dat. (Foote, 2019)

Virtualizace

Datová virtualizace využívá rozhraní mezi místy, kde jsou data fyzicky uložena a a systémem nebo jeho částí, který k daným datům přistupuje. Rozhraní slouží jako abstrakční vrstva a stará se o transformaci dat z různých zdrojů do jednoho formátu, tím pádem se během přístupu k datům nemusí odlišovat formát při přístupu k rozdílným zdrojům. Při virtualizaci nedochází k dlouhodobému ukládání transformovaných dat. (Foote, 2019)

Federace

Datová federace je způsob abstrakce dat, při kterém je využívána virtuální databáze, která sjednocuje data z mnoha databází do jednoho datového modelu. Data jsou uložena samostatně v jednotlivých databázích a při dotazu virtuální databáze řeší, ze kterých konkrétních databázích má data získat a sjednotit do společného předem nadefinovaného datového modelu. To umožňuje například získat v rámci jednoho dotazu data o entitě zákazník, jejichž jedna část je uložena v databázi pro CRM a druhá v databázi pro ERP. (Foote, 2019)

2.4.2 Typy datové integrace podle spouštěče

Přenos dat v informačním systému může být iniciován na základě různých typů spouštěčů (trigger). Každý druh spouštěče je vhodný pro jiný případ užití a lze je v rámci informačního systému kombinovat.

Event-driven

O událostí řízenou integraci jde v případě, že přenos dat je spouštěn jakoukoliv předem definovanou událostí (event). Může jít o kliknutí uživatelem na tlačítko, načtení čipové karty. Na událost z výchozího systému je vždy navázána událost v cílových systémech. Používá se především v případech, kdy nedochází ve výchozím systému ke změně v datech, ale chceme z něj pouze dát vědět jinému systému, že má něco udělat.

Data-driven

Integrace řízená na základě změny v datech jsou spouštěny událostí v jedné z databází v systému (create, update, delete, undelete). Je vhodná v případě, že data nebo jejich část uchováváme v obou systémech.

Jak v případě event-driven, tak data-driven spouštěčů může výměna zpráv probíhat několika způsoby (Hohpe, 2017):

- Fire-and-Forget: Výchozí systém vyšle zprávu a nezajímá se o to, jestli byla zpracována.
- Request-Response: Žadatel pošle dotaz a čeká na odpověď, uživatel mezitím nemůže nic dělat, po získání odpovědi je zpracována a uživatel může pokračovat.
- Request-Delayed Response: Asynchronní způsob request-response, žadatel pošle dotaz, ale nečeká na odpověď, uživatel může v systému dále pracovat a ten mezitím poslouchá na asynchronním vlákne na odpověď, kterou zpracuje poté, co dorazí.
- Data Enrichment: Výchozí systém vyšle neúplnou zprávu (například pouze identifikační číslo objednávky) a cílové systémy se poté samostatně dotazují na pro ně relevantní data.

Polling

Jedná se o dotazování žadatele poskytovatele v pravidelných časových intervalech. Nevýhodou tohoto řešení je, že dotazy jsou realizovány i v případě, že nedošlo k žádné změně v datech a data mohou být v mezičase mezi jednotlivými dotazy desynchronizována. (Hohpe, 2017)

2.4.3 Způsoby integrace dat

V této kapitole jsou popsány různé způsoby a nástroje, jakými lze docílit integrace dat mezi systémy.

Manuální

Manuální způsob integrace dat vyžaduje po osobě, která je za danou integraci nebo její část zodpovědná, ručně sesbírat relevantní data ze všech zdrojů, následně tato data očistit a uložit do společné databáze. Přitom není zaručena konzistence dat a nabízí se široký prostor pro chyby. Tento způsob se téměř nevyužívá a dává smysl jen pro malé jednorázové integrace - např. zanesení málo často se neměnných číselníků o malém množství záznamů.

Application-Based

Softwarová aplikace se stará o vyzvednutí, transformování a přenesení dat. Tyto aplikace jsou náročné na vývoj, ale nabízí vysokou míru customizace a škálovatelnosti.

Middleware

Middleware je software, který funguje jako prostředník pro předávání dat mezi aplikacemi. Často se používá pro integraci zastaralých aplikací, které nemají moderní API a vývoj integračního softwaru by

pro ně byl velmi náročný, nad těmito legacy aplikacemi poskytuje abstrakci, díky níž je práce s nimi jednodušší. Middleware má v sobě často zabudovány nástroje pro logování a debugging.

Jednotný přístup k datům

Jednotný přístup k datům (Uniform data access) je metoda, kdy data jsou uložena v původním formátu ze zdrojových databází, ale při jejich vyzvednutí jsou přeloženy do jednotného formátu. Nabízí jednotný pohled na data, která jsou z několika různých databází.

Common Storage

Jde o způsob, kdy data jsou vyzvednuta z různých databází očištěna od duplicit a sjednocena do jedné databáze. Aplikace poté přistupuje pouze do této databáze.

2.4.4 Autentizace a autorizace

Autentizace je ověření identity uživatele, může být prováděna pomocí znalosti hesla, biometrických údajů, nebo hardwarových tokenů. (Iacono, Gorski, 2019)

Autorizace spočívá v přiřazení a ověření uživateli určitých oprávnění v systému. Může jít o práva k čtení či úpravě určitých dat, zpřístupnění aplikace či jejích dílčích funkcionalit. (Iacono, Gorski, 2019)

3 Analýza požadavků

Následující kapitola definuje požadavky na aplikaci. Na základě rešerše a zkušeností s podobnými implementacemi byly definovány požadavky na výsledné řešení. Požadavky jsou rozděleny do skupin na funkční a nefunkční, jejich přehled je v tabulce 2.

Všechny požadavky jsou následně samostatně popsány.

Tabulka 2: Přehled požadavků

Funkční požadavky	Nefunkční požadavky
F1 Automatická synchronizace uzavřených objednávek ze SFSC do ABRA Flexibee	NF1 Zabezpečení přístupu k aplikaci
F2 Automatická synchronizace synchronizace faktur ze SFSC do ABRA Flexibee	NF2 Zabezpečení přístupových údajů do systémů
F3 Parametrizování synchronizace skrze UI	NF3 Možnost lokalizace
F4 Přehled aktuálního nastavení synchronizace	NF4 Podpora Salesforce LEX
F5 Možnost okamžitého spuštění synchronizace skrze UI	NF5 Škálovatelnost
F6 Logování	NF6 Open-source
	NF7 Nezávislost na licencích třetích stran a speciálních Salesforce licencích

3.1 Funkční požadavky

F1 Automatická synchronizace uzavřených objednávek ze SFSC do ABRA Flexibee

Aplikace bude umožňovat jednostranou synchronizaci uzavřených objednávek ze systému Salesforce do systému ABRA Flexibee. Tato synchronizace bude probíhat automaticky v pravidelných intervalech. Data nebudou odesílána vícekrát, pokud nedojde k jejich změně.

F2 Automatická synchronizace synchronizace faktur ze SFSC do ABRA Flexibee

Společně s objednávkami bude aplikace synchronizovat všechny faktury, které se vážou k objednávkám. Tato synchronizace musí proběhnout i poté co již objednávka byla propsána do systému Flexibee, ale poté k ní přibyl další dokument.

F3 Parametrizování synchronizace skrze UI

Aplikace bude poskytovat uživatelské rozhraní pro parametrizaci synchronizace dat. Tato parametrizace bude zahrnovat:

- Nastavení intervalu synchronizace
- Definice mapování atributů mezi systémy

- Možnost přidat či odebrat atribut pro synchronizaci
- Možnost vypnutí / zapnutí synchronizace

Parametrizovat aplikaci budou moci jen uživatelé s příslušným oprávněním.

F4 Přehled aktuálního nastavení synchronizace

Aplikace bude nabízet přehled aktuálního nastavení synchronizace včetně mapování dat, času poslední proběhlé a následující synchronizace

F5 Možnost okamžitého spuštění synchronizace skrze UI

V aplikaci bude možnost spustit synchronizaci okamžitě a nečekat tak na pravidelný interval. Pravidelná synchronizace nebude tímto okamžitým spuštěním ovlivněna a proběhne podle stávajícího nastavení.

F6 Logování

Aplikace bude zaznamenávat logy o všech chybách a proběhlých synchronizacích.

3.2 Nefunkční požadavky

NF1 Zabezpečení přístupu k aplikaci

Je nutné zabezpečit aby pouze určití uživatelé měli k aplikaci přístup pro čtení a úpravu dat. Budou tři úrovně přístupu: nemá přístup k aplikaci vůbec, přístup pro čtení dat a spouštění synchronizace a přístup pro čtení, úpravu dat a spuštění synchronizace.

NF2 Zabezpečení přístupových údajů do systémů

Přihlašovací údaje z externího systému musí být vhodně zabezpečena proti zneužití.

NF3 Možnost lokalizace

Všechn text, názvy entit a jejich atributů musí být možné lokalizovat do jazyka, který dané uživatel chce používat. Aplikace bude vyvinuta pouze v anglickém jazyce, ale musí být možné vše přeložit bez zásahu do kódu a aplikace musí být o tento překlad jednoduše rozšiřitelná.

NF4 Podpora Salesforce LEX

Aplikaci bude možné používat v prostředí Salesforce Lightning Experience a bude podporovat všechny desktopové webové prohlížeče, které jsou aktuálně podporovány Salesforce LEX. Těmito prohlížeči jsou (jde o poslední stabilní verzi k datu 1.4.2020 - Salesforce Spring '21 release):

- Microsoft® Edge Chromium
- Google Chrome™
- Mozilla® Firefox®
- Apple® Safari®

NF5 Škálovatelnost

Aplikace bude umožňovat škálovatelnost pro synchronizaci dalších entit a atributů jež nejsou pokryty v aktuálním rozsahu práce. Dále musí umožňovat škálovatelnost překladů viz NF3 a možnost rozšíření o synchronizaci dat z ABRA Flexibee do SFSC, což není v rozsahu této práce.

NF6 Open-source

Všechny části aplikace včetně datového modelu pro data a metadata budou open-source ve formátu SFDX.

NF7 Nezávislost na licencích třetích stran a speciálních Salesforce licencích

Aplikace musí být nezávislá na všech licencích třetích stran vyjma Salesforce (nutná pouze standardní Salesforce licence) a ABRA Flexi s jedním API uživatelem. Pro její používání nebude potřeba žádná Salesforce speciální licence.

3.3 Případy užití a role aktérů

3.3.1 Role aktérů

Na základě požadavků byly identifikovány tři skupiny aktérů aplikace - Standardní uživatel, Administrátor aplikace a Salesforce CRM systém. Skupina určuje jaká oprávnění a kompetence bude daný uživatel v aplikaci mít.

Standardní uživatel

Slouží pro běžné uživatele aplikace. Takový uživatel si může zobrazit aktuální nastavení aplikace. Z důvodu, aby v případě chybějících dat v cílovém systému mohl identifikovat to, jestli se stala chyba

během jejich přenosu, nebo jestli se tato data synchronizovat nemají. Může také spouštět okamžitou synchronizaci pro případ, že by potřeboval nová data okamžitě v cílovém systému.

Administrátor aplikace

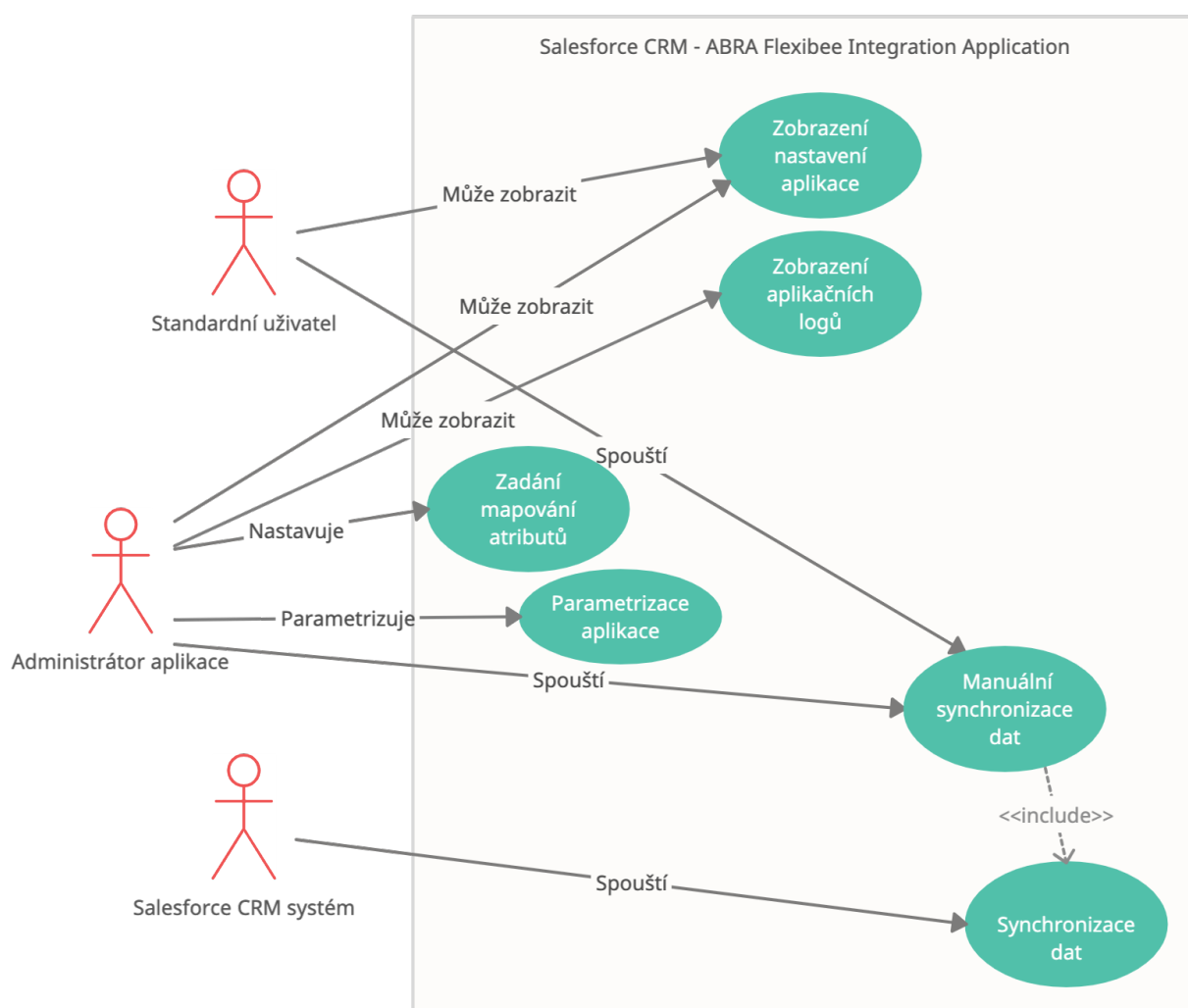
Role administrátora slouží pro uživatele, kteří zodpovídají za parametrizaci aplikace a nastavení mapování dat mezi systémy. Administrátor má také možnost zobrazit aplikační logy z důvodu debugování. Zároveň zahrnuje všechna oprávnění z role Standardního uživatele.

Salesforce CRM systém

Nejedná se o roli pro uživatele, ale jde o systémového aktéra, který zajišťuje automatické spouštění synchronizace dat.

3.3.2 Případy užití

Podle požadavků a identifikace aktérů byly navrženy a popsány případy užití. (viz tabulky 3,4,5,6,7,8)
Diagram případů užití na obrázku 2 znázorňuje relace mezi aktéry a jednotlivými případy užití.



Obrázek 2: UML Diagram případů užití

Tabulka 3: Příklad užití - Zobrazení nastavení aplikace

Identifikátor	IUC
Název	Zobrazení nastavení aplikace
Cíl	Přehled aktuálního nastavení aplikace včetně mapování dat, posledního a následujícího času synchronizace. Rozděleno do jednotlivých sekcí.
Aktéři	Standardní uživatel, Administrátor aplikace
Vstupní podmínky	Uživatel je přihlášen a má otevřenou aplikaci „ABRA Flexibee Integration“
Výstupní podmínky	_____
Scénář	<ol style="list-style-type: none"> 1. Uživatel klikne na záložku pro nastavení aplikace 2. Systém načte a zobrazí aktuální nastavení aplikace a mapování z metadat rozdělených do jednotlivých sekcí podle entit 3. Uživatel klikne na konkrétní podzáložku 4. Systém zobrazí uživateli přehled nastavení a mapování podle vybrané sekce
Alternativní scénář	<ol style="list-style-type: none"> 1. Uživatel klikne na záložku pro nastavení aplikace 2. Během načítání metadat dojde k chybě 3. Systém zobrazí uživateli chybová hláška 4. Systém zalogue chybu do aplikačních logů

Tabulka 4: Příklad užití - Zobrazení aplikačních logů

Identifikátor	2UC
Název	Zobrazení aplikačních logů
Cíl	Zobrazení seznamu aplikačních logů rozděleného do sekcí podle úrovně s možností vyhledávání a seřazení podle datumu nebo abecedně
Aktéři	Administrátor aplikace
Vstupní podmínky	Uživatel je přihlášen a má otevřenou aplikaci „ABRA Flexibee Integration“
Výstupní podmínky	_____
Scénář	<ol style="list-style-type: none"> 1. Uživatel klikne na záložku pro Aplikační logy 2. Systém načte a zobrazí seznam všech logů 3. Uživatel vybere úroveň logů 4. Systém zobrazí se pouze logy z vybrané úrovně 5. Uživatel klikne na sloupec podle kterého se mají záznamy seřadit 6. Systém seřadí záznamy sestupně / vzestupně podle vybraného sloupce 7. Uživatel klikne na číslo konkrétního logu 8. Systém načte a zobrazí detail daného záznamu

Tabulka 5: Příklad užití - Zadaní mapování atributů

Identifikátor	3UC
Název	Zadání mapování atributů
Cíl	Nastavení mapování dat mezi systémy pro jednotlivé entity.
Aktéři	Administrátor aplikace
Vstupní podmínky	Uživatel je přihlášen a má otevřenou aplikaci „ABRA Flexibee Integration“
Výstupní podmínky	Uložení metadat proběhlo úspěšně
Scénář	<ol style="list-style-type: none"> 1. Uživatel klikne na záložku pro nastavení aplikace 2. Systém zobrazí uživateli pole pro výběr entity k namapování 3. Uživatel vybere entitu 4. Systém zobrazí možnost přidat atributy 5. Uživatel klikne na přidat atribut 6. Systém načte a zobrazí se seznam dostupných atributů k namapování s checkboxem pro aktivaci synchronizace daného atributu a pole pro název odpovídajícího atributu v systému ABRA Flexibee s převyplněnými hodnotami podle aktuálního nastavení 7. Uživatel vyplní mapování pro danou entitu a klikne na „Uložit“ 8. Systém uloží metadata a zobrazí uživateli hlášku o úspěšném uložení

Tabulka 6: Příklad užití - Parametrizace aplikace

Identifikátor	4UC
Název	Parametrizace aplikace
Cíl	Nastavení zapnutí či vypnutí synchronizace pro jednotlivé entity, intervalu a objemu záznamů v jedné transakci pro automatickou synchronizaci
Aktéři	Administrátor aplikace
Vstupní podmínky	Uživatel je přihlášen a má otevřenou aplikaci „ABRA Flexibee Integration“
Výstupní podmínky	Uložení metadat proběhlo úspěšně
Scénář	<ol style="list-style-type: none"> 1. Uživatel klikne na záložku pro nastavení aplikace 2. Systém zobrazí uživateli pole pro zadání intervalu synchronizace v minutách, pole pro počet záznamů v jedné transakci a možnost vypnutí synchronizace jednotlivých objektů. 3. Uživatel změní některou z hodnot a klikne na „Uložit“ 4. Systém uloží metadata a zobrazí uživateli hlášku o úspěšném uložení

Tabulka 7: Příklad užití - Synchronizace dat

Identifikátor	5UC
Název	Synchronizace dat
Cíl	Propsání dat z Salesforce CRM do ABRA Flexibee
Aktéři	Systém
Vstupní podmínky	Jsou vyplněna metadata pro synchronizaci dat
Výstupní podmínky	Synchronizace dat mezi systémy proběhla úspěšně
Scénář	<ol style="list-style-type: none"> 1. Uběhne doba intervalu od poslední synchronizace 2. Systém načte nastavení synchronizace 3. Systém paralelně spouští synchronizace pro jednotlivé entity podle daného nastavení 4. Po dokončení synchronizace systém zaloguje informace o proběhnutí pro jednotlivé entity 5. Systém naplánuje další synchronizaci podle daného intervalu
Alternativní scénář	<ol style="list-style-type: none"> 1. Synchronizace je spuštěna uživatelem přes okamžitou synchronizaci 2. Systém načte nastavení synchronizace 3. Systém paralelně spouští synchronizace pro jednotlivé entity podle daného nastavení 4. Po dokončení synchronizace je zalogována informace o proběhnutí pro jednotlivé entity

Tabulka 8: Příklad užití - Manuální spuštění synchronizace dat

Identifikátor	6UC
Název	Manuální spuštění synchronizace dat
Cíl	Okamžité spuštění propování dat z Salesforce CRM do ABRA Flexibee
Aktéři	Standardní uživatel
Vstupní podmínky	Jsou vyplněna metadata pro synchronizaci dat
Výstupní podmínky	Synchronizace dat mezi systémy proběhla úspěšně
Scénář	<ol style="list-style-type: none"> 1. Uživatel klikne na záložku pro nastavení aplikace 2. Systém zobrazí uživateli obrazovku s posledním a dalším časem synchronizace a tlačítkem pro okamžitou synchronizaci 3. Uživatel klikne na tlačítko pro okamžitou synchronizaci 4. Systém uživateli zobrazí hlášku o začátku synchronizace 5. Systém tlačítko pro manuální synchronizaci deaktivuje
Alternativní scénář	<ol style="list-style-type: none"> 1. Uživatel klikne na záložku pro nastavení aplikace 2. Synchronizace dat právě běží 3. Systém uživateli zobrazí obrazovku s posledním a dalším časem synchronizace a deaktivovaným tlačítkem pro okamžitou synchronizaci

4 Analýza datových modelů

Následující kapitola se zabývá částí datových modelů v obou systémech, která je relevantní pro požadavky z předchozí kapitoly. Pro splnění požadavků na přenos objednávek, faktur a dobropisů je potřeba identifikovat a případně doplnit datové modely v obou systémech. Dále je potřeba navrhnout a zvolit vhodný způsob pro ukládání nastavení aplikace a mapování datových modelů.

4.1 Datový model Salesforce CRM

Salesforce ve standardním datovém modelu obsahuje entity pro Objednávku a Produkty, jež jsou ve vztahu M:N propojené přes mezilehlou tabulku Order Product (API název OrderItem). Faktury a dobropisy nejsou ve standardním datovém modelu pro základní licence obsaženy. Tyto entity jsou zpřístupněny pouze pokud má organizace zakoupenou licenci „Salesforce Order Management“, pro splnění nefunkčního požadavku NF7 bude proto potřeba pro tyto objekty vytvořit kustomizovanou část datového modelu.

4.2 Datový model ABRA Flexibee

ABRA umožňuje na jednom serveru spravovat více firem, pro integraci je tedy nutné specifikovat pro jakou firmu má integrace fungovat.

4.2.1 Datové schéma entit

ABRA Flexi umožňuje v rámci REST API načtení datového modelu jednotlivých entit ve formátu XSD. Schémata jsou rozdělena na verze pro export a import dat a je možné je získat pomocí webové služby na adrese `/c/<identifikátor firmy>/<evidence>/<typ schématu.xsd>` (ABRA, Dokumentace XML Schéma)

konkrétní relativní adresa pak může vypadat například takto:

`/c/demo/faktura-vydana/schema-import.xsd`

XML Schema Definition (XSD) popisuje strukturu XML dokumentu nebo jeho JSON varianty, definuje podporované elementy, atributy a datové typy s povolenými hodnotami. (W3Schools, XML Schema Tutorial)

Pomocí nich bude možné docílit interaktivního mapování atributů v aplikaci.

4.2.2 Objednávky a faktury

Datový model ABRA Flexi má již připravené entity pro objednávky, faktury a dobropisy. Entity pro přijaté a vydané faktury jsou různé, stejně tak pro přijaté a vydané objednávky. Z hlediska integrace dat z Salesforce Sales Cloud jsou relevantní pouze přijaté objednávky a vydané faktury. Tabulka 9 znázorňuje názvy a API názvy pro jednotlivé relevantní entity objednávek a faktur v ABRA Flexi.

Tabulka 9: ABRA Flexi API názvy entit pro objednávky a faktury

Název	API název
Přijatá objednávka	objednavka-prijata
Vydaná faktura	faktura-vydana
Vydaná objednávka	objednavka-vydana
Přijatá faktura	faktura-prijata

4.3 Ukládání mapování a parametrizace

V případě mapování a parametrů aplikace se z pohledu systému nejedná o data ale o metadata z toho důvodu není vhodné je ukládat do datových struktur, kterými jsou SObjecty. Salesforce pro tento případ nabízí dva různé způsoby Custom Metadata (Salesforce, Docs What are Custom Metadata Types?) a Custom Settings (Salesforce, Docs Custom Settings).

Custom Metadata jsou kustomizovatelné entity, jejichž obsah je nasaditelný společně se samotnou aplikací. Jsou vhodné pro uchovávání mapování, whitelist seznamů a master dat jako jsou například hodnoty daňových sazeb. (Salesforce, Docs What are Custom Metadata Types?)

Custom Settings jsou kustomizovatelné entity jejichž obsah je nutné po vytvoření vyplnit uživatelem nebo automatizovaným procesem. Používají se pro hodnoty, které mění chování systémů a aplikací (parametrizace) a ve výjimečných případech pro uchování přístupových údajů pro integraci s externím systémem. Custom settings také umožňují vytvoření záznamu, který je specifický pro konkrétního uživatele nebo skupinu uživatelů. (Salesforce, Docs Custom Settings)

4.3.1 Struktura pro mapování dat mezi systémy

Pro mapování byly na základě dokumentace využity Custom Metadata Types. Byly vytvořeny metadata s názvem **ABRA Flexi Data Mappings** s atributy popsány v tabulce 10.

Tabulka 10: Specifikace ABRA Flexi Data Mapping metada atributů

Název	API název	Datový typ	Popis
SFSC SObject Name	SFSC_SObject_Name__c	Text(100)	Specifikuje název SObjectu v Salesforce kterého se záznam mapování týká.
ABRA Flexi Entity Name	ABRA_Flexi_Entity_Name__c	Text(100)	Specifikuje název entity v ABRA Flexi kterého se záznam mapování týká.
SFSC Field Name	SFSC_Field_Name__c	Text(100)	Název atributu SObjectu v Salesforce.
ABRA Flexi Field Name	ABRA_Flexi_Field_Name__c	Text(100)	Název atributu v ABRA Flexi
SFSC To ABRA Flexi Sync	SFSC_To_ABRA_Flexi_Sync__c	Boolean	Udává, jestli je přenos dat pro tento atribut aktivní pro směr ze Salesforce do ABRA Flexi.
ABRA Flexi To SFSC Sync	ABRA_Flexi_To_SFSC_Sync__c	Boolean	Udává, jestli je přenos dat pro tento atribut aktivní pro směr z ABRA Flexi do Salesforce .

4.3.2 Struktura pro ukládání parametrizace

Pro ukládání nastavení aplikace byly na základě dokumentace zvoleny hierarchické custom settings s názvem **ABRA Flexi Data Integration Settings** s atributy popsány v tabulce 11.

Tabulka 11: Specifikace ABRA Flexi Data Integration Settings custom settings atributů

Název	API název	Datový typ	Popis
Order Data Integration Is Active	Order_Data_Integration_Is_Active__c	Boolean	Udává jestli je integrace objednávek zapnuta.
Invoice Data Integration Is Active	Invoice_Data_Integration_Is_Active__c	Boolean	Udává jestli je integrace faktur zapnuta.
Integration Interval In Minutes	Integration_Interval_In_Minutes__c	Number(5, 0)	Specifikuje interval v minutách po kterém je synchronizace spouštěna.
Records In One Batch	Records_In_One_Batch__c	Number(3, 0)	Specifikuje počet záznamů synchronizovaných v jedné transakci, maximální počet je 200.
Company Name	Company_Name__c	Text(255)	Specifikuje API název firmy v ABRA Flexi, pro kterou se data mají synchronizovat.

4.4 Propojení datových modelů a kustomizace

Následující kapitola popisuje kustomizaci entit pro potřeby aplikace a specifikuje mapování atributů mezi objednávkou a fakturou mezi systémy.

4.4.1 Externí identifikátor a příznak pro export

Pro možnost aktualizace záznamu v systému ABRA Flexi je potřeba poskytnout jeho Id. Toto Id je evidováno na každém podporovaném SObjectu pro export v atributu ABRA_Flexi_Id__c a je vedeno jako Id z externího systému. Musí být unikátní a rozlišuje velká a malá písmena.

Dále je potřeba na záznamu evidovat informaci o tom, jestli byl již exportován, nebo ne. K tomuto účelu byl vytvořen na každém podporovaném SObjectu pro export atribut ABRA_Flexi_Data_Exported__c, datového typu boolean, který se po úspěšném exportu nastaví na hodnotu „true“ a při změně relevantních dat na záznamu na „false“.

4.4.2 Faktury

Pro entity vydaných faktur byl vytvořen SObject Invoice (API název Invoice__c). Vztah k objednavce je 1:N (faktura je navázaná na jednu objednávku, jedna objednávka může mít více faktur). Vztah k produktu je M:N a je řešen pomocí mezilehlé tabulky Invoice Item (API název Invoice_Item__c) tento SObject odpovídá entitám faktura-vydana-polozka .

4.4.3 Mapování entit

V tabulkách 12 a 13 je přehled mapování základních atributů pro objednávky a faktury. Toto mapování lze měnit nebo rozšířit. Na objektu Account, který reprezentuje zákazníka musely být vytvořeny atributy pro IČO - Identification_Number__c a DIČ - VAT_Number__c.

Tabulka 12: Mapování objednávky na objednavka-prijata

SFSC API název	ABRA Flexi API název	Popis
ABRA_Flexi_Id__c	id	Id v systému ABRA Flexi
OrderReferenceNumber	kod	Interní číslo
OrderReferenceNumber	varSym	Variabilní symbol
CreatedDate	datVyst	Datum vystavení
EndDate	datTermin	Termín
EffectiveDate	datReal	Realizováno
Description	popis	Popis
Account.Name	nazFirmy	Název firmy nebo jméno osoby
BillingAddress.Street	ulice	Ulice
BillingAddress.City	mesto	Město
BillingAddress.PostalCode	psc	PSČ
Account.Identification_Number__c	ic	IČO
Account.VAT_Number__c	dic	DIČ
TotalAmount	sumOsvMen	Celková suma všech základů v základní sazbě DPH.

Tabulka 13: Mapování faktury na faktura-vydana

SFSC API název	ABRA Flexi API název	Popis
ABRA_Flexi_Id__c	id	Id v systému ABRA Flexi
Code__c	kod	kód faktury
Type__c	typDokl	typ dokladu
Reference_Order__r.OrderReferenceNumber	cisObj	číslo objednávky
Due_Date__c	datSplat	datum splatnosti
Issue_Date__c	datVyst	datum vystavení
Description__c	popis	popis
Total_Amount_With_Tax__c	sumCelkemMen	Celková suma včetně DPH v měně dokladu.
Total_Amount__c	sumCelkZaklMen	Celková suma všech základů v základní sazbě DPH.
Total_Tax_Amount__c	sumDphZaklMen	Celková částka DPH v základní sazbě DPH v měně dokladu.
Billing_Account__r.Name	nazFirmy	Název firmy nebo jméno osoby
Reference_Order__r.BillingAddress.Street	ulice	Ulice
Reference_Order__r.BillingAddress.City	mesto	Město
Reference_Order__r.BillingAddress.PostalCode	psc	PSČ
Billing_Account__r.Identification_Number__c	ic	IČO
Billing_Account__r.VAT_Number__c	dic	DIC
Date_of_Taxable_Supply__c	duzpPuv	Datum uskutečnění zdanitelného plnění
Status__c	stavUhrK	Stav úhrady dokladu

4.4.4 Entita pro aplikační logování

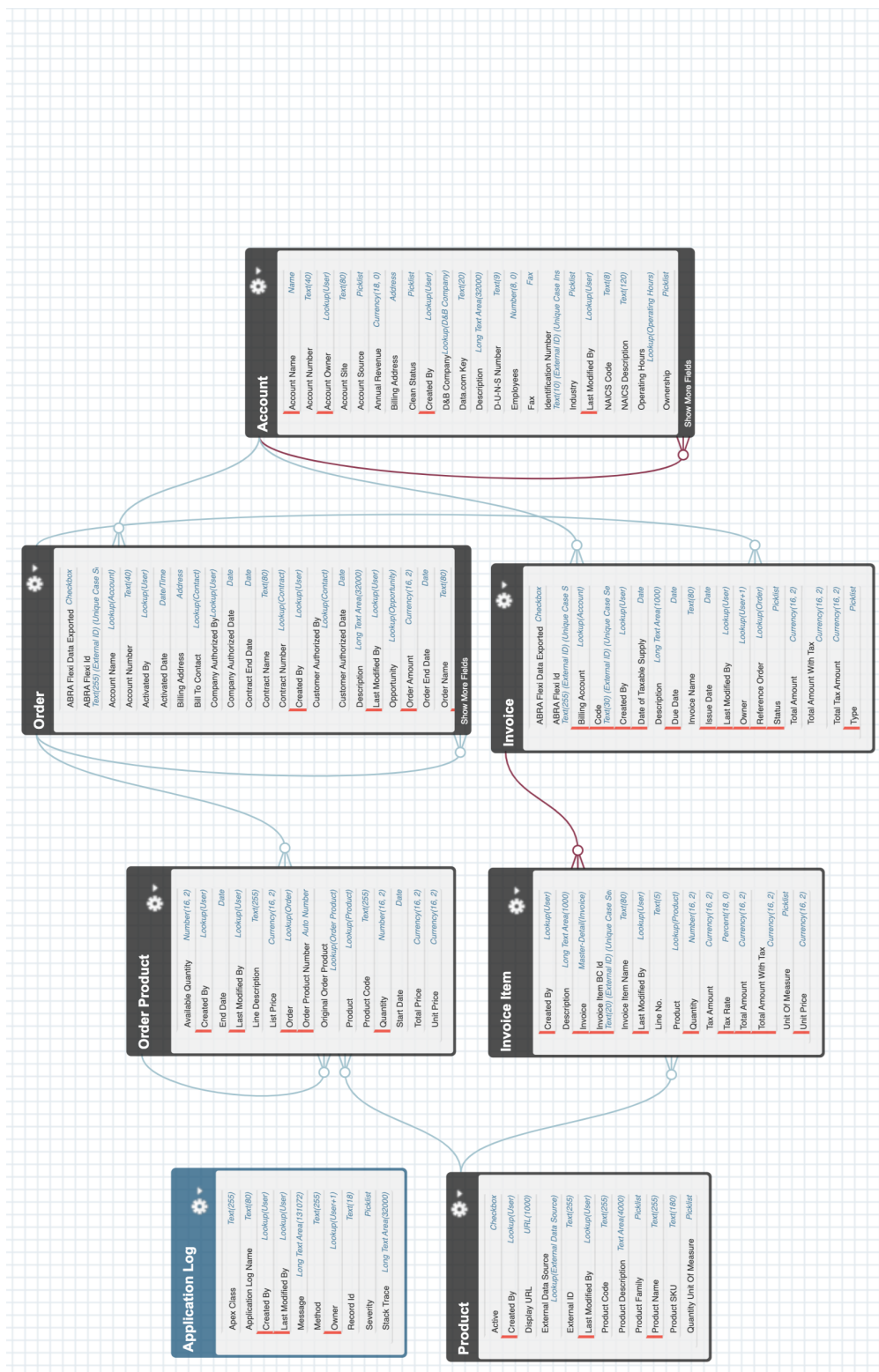
Salesforce neposkytuje žádný standardní SObject pro aplikační logy. Z toho důvodu byl vytvořen custom objekt Application_Log__c s atributy popsány v tabulce 14.

Tabulka 14: Specifikace atributů Application Log SObjectu

Název	API název	Datový typ	Popis
Created Date	CreatedDate	Datetime	Datum a čas vytvoření
Apex Class	Apex_Class__c	Text(255)	Zdrojová třída logu
Method	Method__c	Text(255)	Zdrojová metoda logu
Severity	Severity__c	Picklist	Vážnost logu (ERROR, WARNING, DEBUG, INFO)
Message	Message__c	Long(131072)	Hlavní obsah logu nebo zpráva vyjímky
Record Id	Record_Id__c	Id	Salesforce Id pro logy, které se týkají konkrétního záznamu
Stack Trace	Stack_Trace__c	Long(32000)	Obsah trasování zásobníku pro ERROR logy

4.4.5 Výsledný datový model Salesforce CRM

Následující obrázek 3 znázorňuje vazby v SFSC mezi jednotlivými entitami, které jsou relevantní pro tuto práci. Neobsahuje metadatové objekty pro mapování a parametry.



Obrázek 3: Přehled vazeb mezi relevantními entitami v Salesforce CRM

5 Uživatelské skupiny a práva

Je potřeba zajistit jednoduché přiřazování skupin oprávnění k odlišení administrátorských a standardních uživatelů pro práci s aplikací a daty a systémového uživatele, pod kterým poběží automatická synchronizace dat.

Salesforce nabízí dva základní způsoby správy oprávnění - přes Profile (Salesforce, Docs Profiles) a Permission Set (Salesforce, Docs Permission Sets). Permission sety mohou být přímo přiřazeny uživateli nebo shlukovány do Permission Set Groups (Salesforce, Docs Permission Set Groups), která je pak přiřazena uživateli. Zatímco profil může mít uživatel pouze jeden, permission setů a permission set group může mít přiřazeno více. Dále je možné profilům a permission setům přiřadit Custom Permission (Salesforce, Docs Custom Permissions), kterou lze využít v logice aplikací a chová se jako jednoduchý Boolean - má / nemá.

Z důvodu přehlednosti, škálovatelnosti a oddělení aplikace od konkrétní organizace Salesforce, do které bude nainstalována a aby nebylo potřeba zasahovat do stávajících profilů jsem se rozhodl využít kombinaci permission setů, custom permission a permission set group.

Tabulka 15 zobrazuje obsah Permission Set Group pro standardního uživatele a společné skupiny pro Administrátora a systémového uživatele.

Tabulka 15: Přehled oprávnění podle skupin uživatelů

Oblast oprávnění	Standardní uživatel (ABRA Flexi Integration Standard)	Administrátor aplikace a systémový uživatel (ABRA Flexi Integration Admin)
Objednávky	Order Standard Permission Set	Order Admin Permission Set
Faktury	Invoice Standard Permission Set	Invoice Admin Permission Set
Právo na administrátorský přístup do aplikace	——	ABRA Flexi App Admin (Custom Permission)
Právo na ABRA Flexi Data Mappings	True	True
Aplikační logy	Application Log Standard Permission Set	Application Log Admin Permission Set
Apex třídy	Apex Class Standard Permission Set	Apex Class Admin Permission Set

6 Datová integrace

Následující kapitola popisuje jakým způsobem a za pomoci jakých technologií byl implementován export dat ze SFSC do ABRA Flexi.

6.1 Diagram tříd

Obrázek 4 znázorňuje UML diagram jejich tříd. Je na něm vidět definice tříd logiky pro export dat, jejich atributů, metod a vazeb mezi nimi.

6.2 Automatizace datové integrace

O automatické naplánování dalšího exportu se stará třída `AbraDataExportScheduler`, která implementuje rozhraní `Schedulable`. Po prvním spuštění `AbraDataExportScheduler`, které je nutné provést ručně nebo za pomoci post-install scriptu po nasazení kódu, se další průběhy plánují automaticky podle hodnoty zadané v ABRA Flexi Data Integration Settings - Integration Interval In Minutes. (viz Výpis1)

`Schedulable` rozhraní umožňuje vykonat logiku kódu za nějaký určitý čas a nebo tuto logiku opakovat pravidelně v určitém intervalu. Čas průběhu se plánuje pomocí cron nástroje. (Salesforce, Docs Apex Scheduler)

Výpis 1: Plánování dalšího exportu dat

```
1  /**
2      * @description executes sync batches for each object defined in
3      * batchRunsConfigurations and schedules next run
4      *
5      * @param ctx SchedulableContext
6      * @author tomaschour
7      * @date 2021-04-25
8      */
9  public void execute(SchedulableContext ctx) {
10      // run batch export for every supported object
11      executeBatchForEveryObject();
12      // Delete old scheduled jobs
13      clearOldScheduledApexJobs();
14      // schedule next run
15      scheduleNext();
16      // TODO commit logs
17  }
18  /**
19      * @description Schedules next run of SfcccObjectSyncScheduler
20      *
21      * @author tomaschour
22      * @date 2021-04-25
23      */
24  private void scheduleNext() {
25      Integer scheduleInterval =
26          ((Decimal) AbraFlexiDataIntegrationSettings.getAttributeValue(
27              ABRA_Flexi_Data_Integration_Settings__c
28                  .Integration_Interval_In_Minutes__c
29          ).intValue();
30      Datetime nextTime = System.now();
31      nextTime = nextTime.addMinutes(scheduleInterval);
32      String strJobName = JOB_NAME + System.now().getTime();
33      String strSchedule = nextTime.format('0 m H d MM ? yyyy');
```

```
34         System.schedule(strJobName,  
35             strSchedule,  
36             new AbraDataExportScheduler());  
37     }
```

6.3 Asynchronní hromadný export

Jelikož může docházet k tomu, že v rámci jednoho běhu exportu bude potřeba odeslat velké množství záznamů, je potřeba toto množství dávkovat, aby nedocházelo k přetečení limitů. Tohoto dávkování je v Salesforce možné docílit pomocí implementace interface Database.Batchable. Třída poté zpracovává záznamy po dávkách, jejichž velikost se specifikuje při spuštění dané třídy. (Salesforce, Docs Using Batch Apex) Batch se skládá ze třech hlavních metod - start , execute a finish. (viz Výpis2)

Výpis 2: Ukázka batch třídy

```
1 public class BatchClass implements Database.Batchable<SObject>{  
2  
3     public Database.QueryLocator start(Database.BatchableContext BC){  
4         // načte záznamy ke zpracování  
5     }  
6  
7     public void execute(Database.BatchableContext BC, List<SObject> scope){  
8         // zpracuje záznamy  
9     }  
10  
11     public void finish(Database.BatchableContext BC) {  
12         // logika po tom, co doběhnou všechny dávky, např. log  
13     }  
14 }
```

Nejdříve batch skrze metodu start načte všechny záznamy ke zpracování a vrací je ve formě Iterable třídy nebo Database.QueryLocator. Následuje několik průběhů metody execute, počet průběhů lze dopočítat vzorcem $\frac{\text{celkovýPočetZáznamůKeZpracování}}{\text{velikostDávky}}$ se zaokrouhlením na celá čísla směrem nahoru. Jednotlivé průběhy běží asynchronně v samostatných transakcích. Po zpracování všech dávek se zavolá metoda finish. Pokud chceme v rámci batch třídy provádět volání ven ze systému, je potřeba implementovat rozhraní Database.AllowsCallouts. (Salesforce, Docs Using Batch Apex)

Třída AbraSobjectExportBatch po dávkách exportuje objekty stejného typu. Jaké objekty se mají exportovat se parametrizuje skrze proměnnou typu ExportBatchRunConfiguration, což je třída, která má atributy pro:

- API název SObjectu v SFSC - např. Order nebo Invoice__c
- API název entity v ABRA Flexi - např. faktura-vydana

- Název SFSC pole na objektu, který udává, zda již byl záznam exportován - podle něho se filtrují záznamy ve start metodě
- Dodatečné podmínky pro filtraci záznamů ke zpracování - např. pokud je potřeba rozlišit mezi vydanými a přijatými fakturami - nepovinný atribut

V execute metodě se poté volá implementace AbraSynchronizationInterface rozhraní pro konkrétní SObject, ve které dochází k odeslání dat a zpracování odpovědi - uložení vygenerovaného ABRA Flexi Id na záznamy a označení jako synchronizovaných v případě úspěšného vytvoření. Pokud ve kterékoliv fázi exportu dojde k výjimce, je tato výjimka zalogována.

6.4 Struktura přenášených dat

Tělo HTTP dotazu, které obsahuje data pro upsert (insert nebo update) je ve formátu JSON. Celý objekt má kořenový atribut „winstrom“, který obsahuje vnořený objekt s atributem api názvu entity, jenž obsahuje list záznamů pro upsert. (viz Výpis3)

Výpis 3: JSON pro import dat do ABRA Flexi

```

1  {
2      "winstrom": {
3          "faktura-vydana": [
4              {
5                  "kod": "KOD_FAKTURY001",
6                  "typDokl": 68,
7                  "popis": "popis dokladu Salesforce updated 5"
8              },
9              {
10                 "id": "4228",
11                 "popis": "popis dokladu Salesforce updated 8"
12             }
13         ]
14     }
15 }
```

Odpověď poté obsahuje ve stejném kořenovém atributu informaci o úspěšnosti zpracování, sumarizaci provedených operací - počet vytvořených, aktualizovaných, vymazaných, vynechaných záznamů a kolik záznamů skončilo neúspěchem a následně atribut s detailem výsledků (viz Výpis4)

Výpis 4: JSON odpověď z ABRA Flexi

```

1  {
2      "winstrom": {
3          "@version": "1.0",
4          "success": "true",
5          "stats": {
6              "created": "1",
```

```

7         "updated": "1",
8         "deleted": "0",
9         "skipped": "0",
10        "failed": "0"
11    },
12    "results": [
13        {
14            "id": "4239",
15            "ref": "/c/demo/faktura-vydana/4239"
16        },
17        {
18            "id": "4228",
19            "request-id": "4228",
20            "ref": "/c/demo/faktura-vydana/4228"
21        }
22    ]
23 }
24 }

```

6.5 Autentizace

Pro autorizaci do externích systémů poskytuje Salesforce doporučuje využití Named Credentials. Named Credentials umožňují specifikovat url adresu pro HTTP dotazy a parametry autentizace v jednom. Výhoda tohoto je, že nemusíme přístupové údaje ukládat někam mimo a nebo při nejhorším je držet v kódu. Salesforce také tyto přístupové údaje šifruje. (Salesforce, Docs Named Credentials)

Pro autentizaci dotazů do ABRA Flexi byly vytvořeny ABRA_Flexi Named Credentials odkazující na url cloudové demo instance, na které byla tato práce vyvíjena: <https://demo.flexibee.eu>. Byla zvolena basic autentizace pomocí přihlašovacího jména a hesla, které ABRA pro demo poskytuje. Uživatel, který je používán pro integraci dat by měl mít v ABRA Flexi systému dostatečná oprávnění na úpravu a zakládání všech podporovaných entit a na čtení jejich XSD schématu.

Do ABRA Flexi je možné se mimo jednoduchou HTTP autentizaci autentizovat pomocí JSON autentizace, SAMLv2 a OpenId. Bohužel poslední dvě zmiňované jsou podporovány pouze na lokální stanici. (ABRA, Dokumentace Autentizace)

Named Credentials také nabízí možnost autentizace pomocí OpenId a jde z hlediska bezpečnosti o preferovanou variantu před základní HTTP autentizací. (Iacono, Gorski, 2019) Pokud je to tedy možné, je vhodné použít OpenId.

6.6 Rozšíření o podporu dalších entit

Pro rozšíření logiky o export dalších entit je nutné pro tyto entity, pokud již neexistuje, vytvořit třídu s názvoslovím `Abra<API název SFSC SObjectu bez '_' a '__c'>DataSyncService`, která implementuje

rozhraní `AbraSynchronizationInterface`. V této vytvořené třídě je pak implementována metoda `exportRecords`, která obsahuje logiku pro přípravu dat, volání export HTTP dotazu, a zpracování výsledku.

Dále je potřeba pro danou entitu přidat instanci konfigurace batche do proměnné `batchRunsConfigurations` ve třídě `AbraDataExportScheduler`. (viz Výpis5)

Výpis 5: Konfigurace exportu entit pro batch

```
1 private final List<AbraSubjectExportBatch.ExportBatchRunConfiguration> batchRunsConfigurations =  
2     new List<AbraSubjectExportBatch.ExportBatchRunConfiguration> {  
3         new AbraSubjectExportBatch.ExportBatchRunConfiguration('Order',  
4             'objednavka-prijata', 'ABRA_Flexi_Data_Exported__c'),  
5         new AbraSubjectExportBatch.ExportBatchRunConfiguration('Invoice__c',  
6             'faktura-vydana', 'ABRA_Flexi_Data_Exported__c')  
7     };
```

Také je vhodné přidat boolean atribut do ABRA Flexi Data Integration Settings, který pomocí kterého lze integraci pro daný objekt vypínat a zapínat.

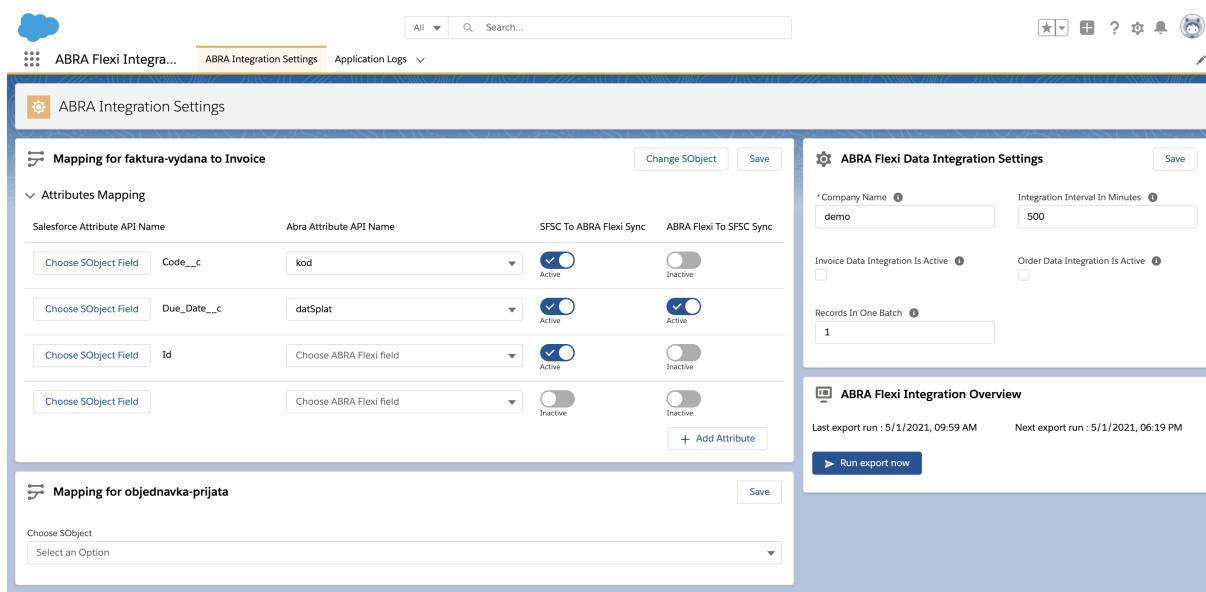
Daná entita musí mít alespoň jeden odpovídající záznam v ABRA Flexi Data Mappings, pokud takový záznam neexistuje, asynchronní process pro danou entitu skončí výjimkou.

7 Uživatelská aplikace

Uživatelská aplikace umožňuje uživatelům spravovat mapování pro podporované entity.

Byla vytvořena lightning aplikace s názvem ABRA Flexi Integration, která obsahuje dvě stránky:

- ABRA Integration Settings - aplikační stránka, která obsahuje mapování a parametrizaci integrace
- Application Logs - stránka záznamů aplikačních logů



Obrázek 5: Uživatelské rozhraní aplikace

7.1 Abra Integration Settings

Stránka s nastaveními se skládá ze dvou sloupců, v levém sloupci jsou jednotlivé komponenty pro nastavení mapování a v pravém jsou pak komponenty pro parametrizaci aplikace a manuálního spouštění exportu dat. Stránku lze upravovat, přidávat a odebírat její komponenty pomocí Salesforce App Builder. (Salesforce, Docs Lightning App Builder)

7.1.1 Komponenta pro mapování entit

Komponenta pro mapování entit se nazývá abraMappingManagement a je tvořena dynamickou tabulkou, která má čtyři sloupce pro výběr Salesforce atributu, ABRA Flexi atributu a dvou přepínačů na aktivaci směru synchronizace. Při načtení si komponenta vždy dynamicky načte názvy atributů na daném objektu v SFSC a ABRA Flexi, předejde se tak tomu, že by se uživatel v názvu atributu přepsal a nebo zvolil neexistující atribut. Komponenta má dvě veřejné proměnné pro API název entity v ABRA Flexi, pro kterou chceme mapování tvořit a ze které entity se načítá schéma pro získání atributů. Pokud pro danou entitu neexistuje alespoň jeden záznam v mapování, musí uživatel vybrat SObject v Salesforce na který chce danou ABRA entitu namapovat viz obrázek 6. Pokud již existuje alespoň jeden záznam mapování,

je SObject automaticky přiřazen z onoho mapování a uživatel může rovnou mapování upravovat viz obrázek 7.

Obrázek 6: Komponenta pro mapování, pokud pro entitu neexistuje žádný záznam

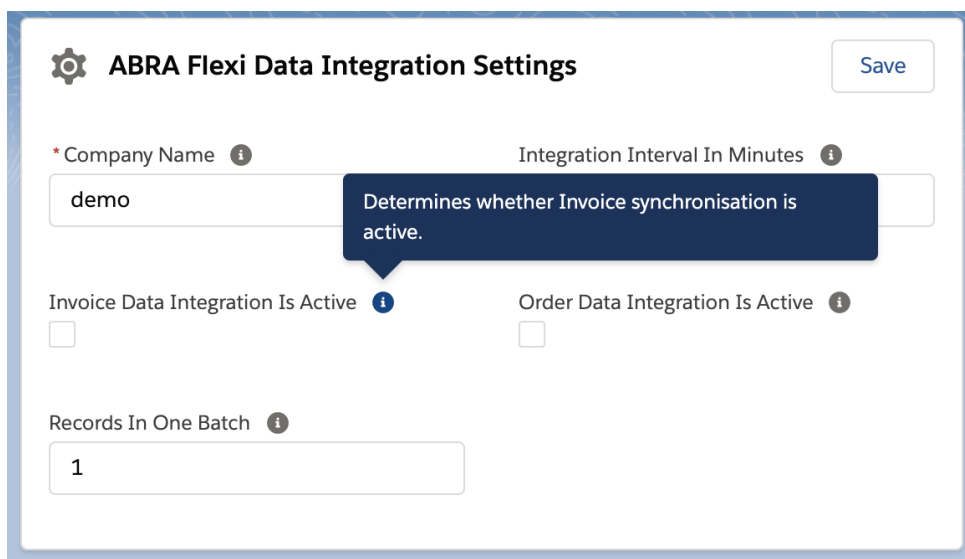
Obrázek 7: Komponenta pro mapování, pokud po výběru SObjectu nebo při existujícím mapování

Je možné změnit SObject na který má být entita namapována, tato změna vymaže všechny stávající mapování pro danou entitu a uživatel může vytvořit nové záznamy s mapováním z jiného SObjectu, obrázek 8 zobrazuje, jak vypadá potvrzovací okno pro vymazání mapování.

Obrázek 8: Komponenta pro mapování, dotaz na změnu SObjectu

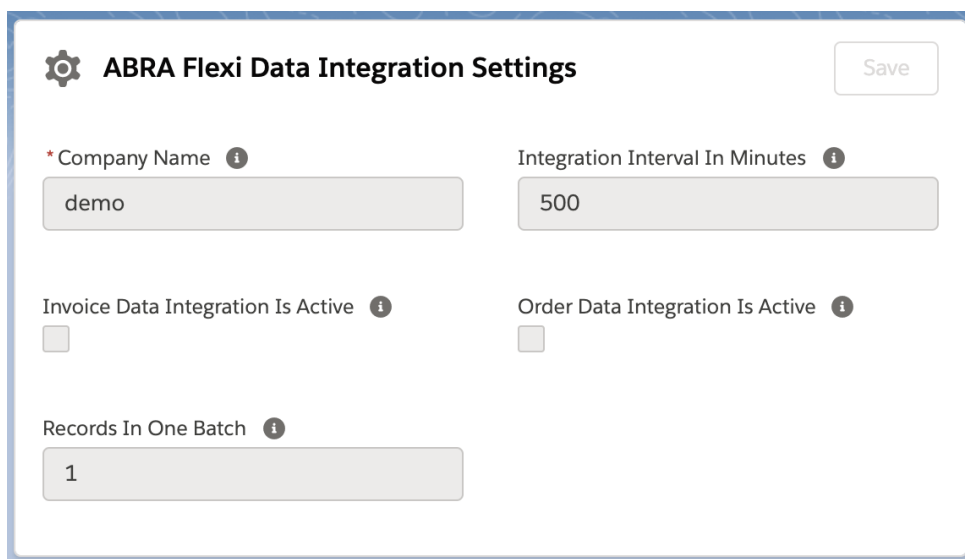
7.1.2 Komponenta pro parametrizaci aplikace

Komponenta pomocí které může uživatel upravovat parametry integrace se nazývá CustomSettingsManagement, je zobrazena na obrázku 9. Logika komponenty dynamicky načítá všechny atributy obsažené v ABRA Flexi Data Integration Settings custom settings a zobrazuje je ve správném html prvku pro vstup dat. Po najetí myši na ikonu informace u daného atributu zobrazí nápovědu, která se k němu vztahuje. Po úspěšném uložení se uživateli zobrazí hláška o uložení nastavení.



Obrázek 9: Komponenta pro úpravu parametrů v administrátorském módu

Pokud uživatel nemá právo na úpravu aplikace, komponenta se načte pouze v módu pro zobrazení viz obrázek 10.



Obrázek 10: Komponenta pro úpravu parametrů v módu pouze pro čtení

7.2 Aplikační logy

Stránka pro zobrazení aplikačních logů využívá standardní zobrazení Salesforce List View, které načítá seznam záznamů daného SObjectu, na které má aktuální uživatel práva. Nad seznamem lze provádět filtrování a řazení a je možné vytvářet předfiltrované seznamy. K defaultnímu seznamu se všemi logy byl vytvořen seznam s pouze chybovými logy.

K záložce aplikačních logů má přístup pouze administrátor.

Salesforce neposkytuje žádný způsob, jakým vytvářet logy, různé projekty si tak vytvářejí své loggery. Z toho důvodu bylo v rámci této práce využito open-source logger dostupný na <https://gist.github.com/miragedeb/70d5b70a6f73c8530db7>. (Mayank, 2015) Tento logger byl upraven, aby odpovídal standardům nových verzí Salesforce a doplněn o podporu logů o severitě varování a debug. Také bylo pozměněna logika pro ukládání logů do databáze tak, aby se jednotlivé logy v rámci transakci neukládali postupně každý zvlášť, ale všechny najednou. Jedná se o velmi jednoduchý logger s pouze jednou třídou, byl zvolen z toho důvodu, aby šel jednoduše nahradit za jakýkoliv jiný logger, který by používala cílová organizace.

Obrázek 11 zobrazuje seznam aplikačních logů seřazených podle jména vzestupně, obrázek 12 pak zobrazuje detail logu po kliknutí na název logu v seznamu.

	Application Log Name	Created Date	Severity	Apex Class	Method	Message
1	a033N000003ljcb	5/1/2021, 3:26 AM	ERROR	ApexLogTest	createErrorLog	Divide by 0
2	a033N000003ljcg	5/1/2021, 3:27 AM	INFO	ApexLogTest	createInformationLog	Logging Information from an Apex Class - ApexLogTest
3	a033N000003ljch	5/1/2021, 3:27 AM	DEBUG	ApexLogTest	createDebugLog	Logging Debug from an Apex Class - ApexLogTest
4	a033N000003ljci	5/1/2021, 3:27 AM	WARNING	ApexLogTest	createWarningLog	Logging Warning from an Apex Class - ApexLogTest
5	a033N000003ljzm	4/30/2021, 11:39 PM	DEBUG	RestIntegrationService	callWebService	The request body be send to: callout:ABRA_Flexi/c/demo/faktura-vydana/schema-import.xsdBody:

Obrázek 11: Stránka se seznamem logů

Related	Details
Application Log Name	a033N000003ljzm
Severity	DEBUG
Apex Class	RestIntegrationService
Method	callWebService
Record Id	
Message	The request body be send to: callout:ABRA_Flexi/c/demo/faktura-vydana/schema-import.xsdBody:
Stack Trace	
Created By	User User, 4/30/2021, 11:39 PM
Last Modified By	User User, 4/30/2021, 11:39 PM

Obrázek 12: Detail záznamu logu

7.3 Lokalizace

Všechny UI texty jsou řešeny skrze custom labels. To umožňuje jejich překládání skrze nastavení Salesforce bez nutnosti zásahu do kódu. Label může mít překlad do všech podporovaných jazyků a použitý překlad se určuje podle osobního nastavení konkrétního Salesforce uživatele.

8 Testování

Pro ověření funkčnosti a ověření funkčnosti při budoucích změnách byly napsány jednotkové Apex testy a navrženy uživatelské testovací scénáře.

8.1 Jednotkové testy

Jednotkové testy v jazyce Apex slouží k ověření funkčností jednotlivých tříd, metod a triggerů. Salesforce požaduje jednotkovými testy alespoň 75% pokrytí veškeré logiky v Apex třídách a triggerech. (Salesforce, Docs Testing and Code Coverage)

Pro každou třídu byla implementována testovací třída s názvem [názevTřídyProLogiku]Test. Testovací třída má několik testovacích metod pro každou metodu testované třídy. Tyto metody ověřují výsledek při úspěšném průběhu logiky, při chybných datech a výjimkách. Pro testování logiky, která provádí dotazy a získává odpovědi z externího systému ABRA Flexi byly vytvořeny mock třídy, které tyto odpovědi simulují. Tyto testy by neměly být měněny pokud nedojde ve změně stávající logiky. Zároveň také jejich výsledky musí být úspěšné při rozšíření logiky.

8.2 Uživatelské testy

Pro ověření uživatelské části aplikace byly podle případů užití navrženy a ověřeny testovací scénáře. Testování bylo prováděno ve verzi Salesforce Spring '21 reálným uživatelem, který pracuje v IT v prostředí Salesforce a byl seznámen s problematikou aplikace. Aplikace byla v anglickém jazyce a byly použity browsery

8.2.1 Testovací scénáře

Tato kapitola definuje testovací scénáře pro testy uživatelského rozhraní.

1TC: Zobrazení a uložení nastavení aplikace

Ověřuje načtení a zobrazení dat pro nastavení aplikace

Prerekvizity: Uživatel je přihlášen jako administrátor a má otevřenou aplikaci „ABRA Flexibee Integration“, ABRA Flexi Data Integration Settings nejsou vyplněny

Testovací data:

- nazevSpolecnosti: test
- interval: 10
- pocetZaznamu: 1

Tabulka 16 specifikuje jednotlivé kroky pro testovací scénář 1TC

Tabulka 16: 1TC kroky

Pořadí kroku	Popis kroku	Očekávaný výsledek
1	Uživatel klikne na záložku „ABRA Integration Settings“	Na pravé straně obrazovky se načte formulář s názvem „ABRA Flexi Data Integration Settings“ a prázdnými vstupy pro název firmy, interval integrace, zapnutí synchronizace objednávek a faktur a počet záznamů v jedné dávce.
2	Uživatel najede myší na ikonu nápovědy u vstupu pro počet záznamů v dávce	Zobrazí se popup s textem nápovědy
3	Uživatel zadá {navezSpolecnosti} do vstupu pro název společnost, {interval} do vstupu pro interval, {pocetZaznamu} do vstupu pro počet záznamů a zaškrtně synchronizaci objednávek	Data jsou vyplněna v příslušných vstupech
4	Uživatel klikne na Save	Zobrazí se hláška o úspěšném uložení
5	Uživatel provede znovunačtení stránky	Formulář se načte s daty uloženými v předchozím kroku

2TC: Zobrazení aplikačních logů

Ověřuje načtení a zobrazení aplikačních logů

Prerekvizity: Uživatel je přihlášen jako administrátor a má otevřenou aplikaci „ABRA Flexibee Integration“, existují záznamy v aplikačních logách

Testovací data:

- navezLogu: a033N000003ljcb

Tabulka 17 specifikuje jednotlivé kroky pro testovací scénář 2TC

Tabulka 17: 2TC kroky

Pořadí kroku	Popis kroku	Očekávaný výsledek
1	Uživatel klikne na záložku „Application Logs“	Zobrazí se seznam logů s nedávno zobrazenými logy.
2	Uživatel vybere zobrazení Error logů	Zobrazí se seznam logů s vyfiltrovanými error logy.
3	Uživatel kline na název sloupce „Created date“	Logy se seřadí podle datumu vytvoření sestupně
4	Uživatel klikne na název logu {navezLogu}	Uživatel je přesměrován na stránku detailu logu

3TC: Zadání a zobrazení mapování atributů

Ověřuje zadávání a mapování atributů mezi systémy

Prerekvizity: Uživatel je přihlášen a má otevřenou aplikaci „ABRA Flexibee Integration“, v metadatech pro mapování není žádný záznam

Testovací data:

- sfscField: Code__c
- sfscObject: Order
- abraAtribut: kod

Tabulka 18 specifikuje jednotlivé kroky pro testovací scénář 3TC

Tabulka 18: 3TC kroky

Pořadí kroku	Popis kroku	Očekávaný výsledek
1	Uživatel klikne na záložku „ABRA Integration Settings“	Na levé straně obrazovky se načte komponenta s názvem „Mapping for objednavka-prijata“ s vstup pro výběr objektu pro mapování
2	Uživatel vybere objekt {sfscObject}	Nadpis komponenty se změní na „Mapping for objednavka-prijata to {sfscObject}“ a zobrazí se tabulka s řádky pro mapping a tlačítko „Add Attribute“
3	Uživatel klikne na „Add Attribute“	Přidá se prázdný řádek do tabulky pro mapování
4	Uživatel klikne na „Choose SObject Field“	Zobrazí se modal pro výběr sfsc atributu
5	Uživatel vybere {sfscField} a klikne na „Confirm“	V tabulce je vybrán {sfscField}
6	Uživatel klikne na dropdown pro výběr ABRA atributu	Načtou se dostupné atributy pro objednavka-prijata
7	Uživatel vybere {abraAtribut}	V tabulce je vybrán {abraAtribut}
8	Uživatel klikne na „Save“	Zobrazí se hláška o úspěšném uložení
9	Uživatel provede znovunačtení stránky	Formulář se načte s daty uloženými v předchozích krocích

4TC: Manuální spuštění synchronizace dat

Ověřuje načtení poslední a další synchronizace dat a manuálního spuštění

Prerekvizity: Uživatel je přihlášen a má otevřenou aplikaci „ABRA Flexibee Integration“, ABRA Flexi Data Integration Settings jsou vyplněny, synchronizace dat proběhla v minulosti a je naplánována další, synchronizace dat aktuálně neprobíhá

Tabulka 19 specifikuje jednotlivé kroky pro testovací scénář 4TC

Tabulka 19: 4TC kroky

Pořadí kroku	Popis kroku	Očekávaný výsledek
1	Uživatel klikne na záložku „ABRA Integration Settings“	Na pravé straně obrazovky se načte komponenta s názvem „ABRA Flexi Integration Overview“ s informací o datu a času poslední proběhlé synchronizace, časem další naplánované a aktivním tlačítkem pro okamžité spuštění synchronizace.
2	Uživatel klikne na tlačítko pro okamžitou synchronizaci.	Zobrazí se hláška o naplánování synchronizace a tlačítko je deaktivováno.
3	Uživatel provede znovunačtení stránky	Na pravé straně obrazovky se načte komponenta s názvem „ABRA Flexi Integration Overview“ s informací o datu a času poslední proběhlé synchronizace, časem další naplánované a s neaktivním tlačítkem pro okamžité spuštění synchronizace.

8.3 Výsledky testů

Spuštění a vyhodnocení všech jednotkových testů proběhlo úspěšně. Uživatelské testy probíhaly ve dvou kolech ve kterých byly vždy provedeny všechny scénáře. Během prvního kola byla odhalena chyba v 3TC v 1. kroku, kdy se nenačetl vstup pro výběr entity, tato chyba byla opravena a v druhém kole byl scénář úspěšně dokončen.

9 Distribuce

Aplikace je uložena ve formátu SFDX na platformě GitHub na adrese <https://github.com/Calgren/Salesforce-CRM-ABRA-Flexibee-Integration>, repozitář se jmenuje „Salesforce-CRM-ABRA-Flexibee-Integration“ a verze určená pro instalaci do Salesforce organizací se nachází ve větvi „main“. Pro distribuci aplikace musí mít uživatel na své stanici nainstalovaný Salesforce CLI. (Salesforce, Salesforce CLI)

Zdrojový kód ve formátu SFDX může být přímo nahrán do prostředí Salesforce určeného k vývoji skrze příkaz `force:source:push`, pro nasazení do sandboxu nebo produkčního prostředí musí být zdrojový kód zkonvertován do Metadata API formátu, to se provede pomocí `force:source:convert` a následně `force:source:deploy`. (Salesforce, Docs source Commands)

10 Závěr

Cílem této práce bylo analyzovat a navrhnout aplikaci pro integraci dat mezi systémy Salesforce CRM a ABRA Flexi, která bude umožňovat dynamické definování mapování atributů entit mezi oběma systémy,

Nejdříve jsou v první části práce představeny oba systémy Salesforce CRM a ABRA Flexi, jejich nástroje pro vývoj, datový model a API. Dále jsou v této části popsány různé způsoby a přístupy k datové integraci v informačním systému.

Po obecném seznámení se systémy a datovou integrací následuje definice funkčních a nefunkčních požadavků, na jejich základě jsou navrženy případy užití a role aktérů. Následuje kapitola, která se zabývá analýzou datových modelů jednotlivých systémů a jejich propojením v této kapitole jsou také navrženy datové struktury pro ukládání metadat a logů. Dále byl doplněn datový model objednávek a faktur a definováno mapování atributů pro tyto entity.

Následuje kapitola, která představuje nástroje použité pro definici uživatelských práv a jejich přiřazování. Dále jsou pak navrženy konkrétní sady oprávnění pro administrátora a standardního uživatele.

Po návrhu oprávnění byly pomocí jazyka Apex navrženy třídy a implementována logika pro přenos dat mezi systémy a jeho automatizace. Tato část definuje strukturu přenášených dat, jejich načítání a zpracování. Také je zde implementováno logování pomocí open-source loggeru a specifikován způsob autentizace do systému ABRA Flexi.

Po implementaci logiky pro přenos dat byla navržena aplikace s uživatelským rozhraním pro nastavení mapování, parametrizaci a ovládání integrace. Byla vytvořena lightning aplikace a tři custom komponenty. Tyto komponenty byly následně přidány do aplikace. Dále byl vytvořen přehled logů.

Další kapitola se zabývá ověřením funkčnosti aplikace a regresními testy. Pro každou Apex třídu byla implementována testovací třída s jednotkovými testy. Pro testování uživatelské aplikace byli navrženy a provedeny testovací scénáře.

Poslední kapitola definuje, jakým způsobem lze vytvořenou aplikaci distribuovat do jednotlivých Salesforce organizací.

Výstupem práce je open-source aplikace pro integraci dat mezi Salesforce CRM a ABRA Flexi, která splňuje požadavky definované v práci a využívá standardní nástroje platformy Salesforce. Aplikace byla řádně otestována a byly poskytnuty regresní testy pro ověření funkčnosti v případě budoucích rozšíření.

Seznam použité literatury

MARTIN, C. Robert, 2018. Clean architecture: a craftsman's guide to software structure and design. London, England: Prentice Hall. ISBN 978-0134494166.

BRUCKNER, Tomáš, VOŘÍŠEK, Jiří, BUCHALCEVOVÁ, Alena, STANOVSKÁ, Iva, CHLAPEK, Dušan, ŘEPA, Václav, 2012. Tvorba informačních systémů: principy, metodiky, architektury. Praha: Grada. Management v informační společnosti. ISBN 978-80-247-4153-6.

KOELSCH, George, 2016. Requirements Writing for System Engineering. Apress, Berkeley, CA. ISBN 978-1-4842-2098-6.

KRATOCHVÍL, Ondřej, 2019. Continuous Delivery on the Salesforce Platform. Praha. Bachelor's thesis. Czech Technical University in Prague, Faculty of Electrical Engineering. Vedoucí práce ŠEBEK Jiří.

ABRA, nedatováno. Dokumentace REST API: Dokumentace a referenční příručka FlexiBee REST API [online]. [cit. 2021-03-09]. Dostupné z: <https://intercom.help/podpora-flexi/cs/collections/2592813-dokumentace-rest-api>

ABRA [online]: Dokumentace XML Schéma. [cit. 2021-04-21]. Dostupné z: <https://www.flexibee.eu/api/dokumentace/ref/xsd>

ABRA [online]: Dokumentace Autentizace. [cit. 2021-04-21]. Dostupné z: <https://www.flexibee.eu/api/dokumentace/ref/login/>

IACONO, Luigi Lo a Peter Leo GORSKI, 2019. On the Need for a General REST-Security Framework. Future Internet [online]. [cit. 2021-03-11]. Dostupné z: [doi:10.3390/fi11030056](https://doi.org/10.3390/fi11030056)

SALESFORCE, nedatováno. Salesforce Developer Documentation [online]. [cit. 2021-03-12]. Dostupné z: <https://developer.salesforce.com/docs>

SALESFORCE [online]: Docs Profiles. [cit. 2021-04-20]. Dostupné z: https://developer.salesforce.com/docs/atlas.en-us.securityImplGuide.meta/securityImplGuide/admin_userprofiles.htm

SALESFORCE [online]: Docs Permission Sets. [cit. 2021-04-20]. Dostupné z: https://developer.salesforce.com/docs/atlas.en-us.securityImplGuide.meta/securityImplGuide/perm_sets_overview.htm

SALESFORCE [online]: Docs Custom Permissions. [cit. 2021-04-20]. Dostupné z: https://developer.salesforce.com/docs/atlas.en-us.securityImplGuide.meta/securityImplGuide/custom_perms_overview.htm

SALESFORCE [online]: Docs Permission Set Groups. [cit. 2021-04-20]. Dostupné z: https://help.salesforce.com/articleView?id=sf.perm_set_groups.htm

SALESFORCE [online]: Docs Custom Settings. [cit. 2021-04-20]. Dostupné z: https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_customsettings.htm

SALESFORCE [online]: Docs What are Custom Metadata Types? [cit. 2021-04-20]. Dostupné z: https://help.salesforce.com/articleView?id=sf.custommetadatatypes_about.htm

SALESFORCE [online]: Docs Apex Scheduler [cit. 2021-04-20]. Dostupné z: https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_scheduler.htm

SALESFORCE [online]: Docs Using Batch Apex [cit. 2021-04-20]. Dostupné z: https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_batch_interface.htm

SALESFORCE [online]: Docs Named Credentials [cit. 2021-04-20]. Dostupné z: https://help.salesforce.com/articleView?id=sf.named_credentials_about.htm

SALESFORCE [online]: Docs Lightning App Builder [cit. 2021-04-25]. Dostupné z: https://help.salesforce.com/articleView?id=sf.lightning_app_builder_overview.htm

SALESFORCE [online]: Docs Testing and Code Coverage [cit. 2021-05-07]. Dostupné z: https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_code_coverage_intro.htm

SALESFORCE [online]: Docs source Commands [cit. 2021-05-07]. Dostupné z: https://developer.salesforce.com/docs/atlas.en-us.sfdx_cli_reference.meta/sfdx_cli_reference/cli_reference_force_source.htm

SALESFORCE [online]: Salesforce CLI [cit. 2021-05-07]. Dostupné z: <https://developer.salesforce.com/tools/sfdxcli>

SALESFORCE [online]: What is SaaS? [cit. 2021-05-07]. Dostupné z: <https://www.salesforce.com/eu/learning-centre/tech/saas/>

SALESFORCE [online]: What is PaaS? [cit. 2021-05-07]. Dostupné z: <https://www.salesforce.com/eu/learning-centre/tech/paas/>

SALESFORCE. Execution Governors and Limits [online]. [cit. 2021-04-09]. Dostupné z: https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_gov_limits.htm

GARTNER, 2019. Market Share: Customer Experience and Relationship Management, Worldwide, 2018 [online]. Dostupné z: <https://www.gartner.com/en/documents/3984886/market-share-customer>

MAYANK, 2015. ApexDebugLog [cit. 2021-04-25]. Dostupné z: <https://gist.github.com/miragedeb/70d5b70a6f73c8530db7>

Anon, nedatováno. Salesforce Lightning Design System [online]. [cit. 2021-03-12]. Dostupné z: <https://www.lightningdesignsystem.com/>

ČSÚ, 2016 [Český statistický úřad]. Firmy využívající cloudové CRM aplikace pro správu a využití informací o zákaznících [tabulka]. 22.12.2016. Dostupné z: https://www.czso.cz/documents/10180/32961948/06200516_104.pdf

ČSÚ, 2019 [Český statistický úřad]. Firmy využívající cloudové CRM aplikace pro správu a využití informací o zákaznících, [tabulka]. 11.01.2019. Dostupné z: <https://www.czso.cz/documents/10180/61601888/0620051843.pdf>

ČSÚ, 2021 [Český statistický úřad]. Firmy využívající cloudové CRM aplikace pro správu a využití informací o zákaznících [tabulka]. 07.01.2021 . Dostupné z: <https://www.czso.cz/documents/10180/122362688/06200520077.pdf>

VONFANGE, M. a LAVIGNE, D, 2011. IT cost optimization through open source software. The Open Source Business Resource, , 5-10. Dostupné z: <https://search-proquest-com.zdroje.vse.cz/scholarly-journals/cost-optimization-through-open-source-software/docview/1695043191/se-2>

SALESFORCE EMEA, nedatováno. Customer 360 [online]. [cit. 2021-03-20]. Dostupné z: <https://www.salesforce.com/products/>

LIU, David. Clicks versus Code! [online]. 2018 [cit. 2021-04-09]. Dostupné z: <https://www.sfdc99.com/2018/01/22/workflow-process-builder-flow-apex/>

ABRA FLEXI S.R.O. ABRA Flexi [online]. [cit. 2021-04-09]. Dostupné z: <https://www.flexibee.eu/>

SYSTEM ONLINE. ABRA FLEXIBEE [online]. [cit. 2021-04-09]. Dostupné z: <https://www.systemonline.cz/prehled-informacnich-systemu/ekonomicke-systemy/abra-flexibee-1.htm>

GUAY, Matthew. What are webhooks? [online]. 2020 [cit. 2021-04-09]. Dostupné z: <https://zapier.com/blog/what-are-webhooks/>

FOOTE, Keith D. The Fundamentals of Data Integration [online]. 2019 [cit. 2021-04-09]. Dostupné z: <https://www.dataversity.net/the-fundamentals-of-data-integration/>

HOHPE, Gregor. Conversation Patterns [online]. 2017 [cit. 2021-04-09]. Dostupné z: <https://www.enterpriseintegrationpatterns.com/patterns/conversation/index.html>

W3SCHOOLS. XML Schema Tutorial [online]. [cit. 2021-4-21]. Dostupné z: https://www.w3schools.com/xml/schema_intro.asp