

Informační systém pro řízení projektů

Information system for a Project Management

Pavel Konečný

Bakalářská práce

Vedoucí práce: doc. Ing. Radim Bača, Ph.D.

Ostrava, 2021

Abstrakt

Cílem této bakalářské práce je vytvoření informačního systému pro řízení projektů pro společnost zabývající se vývojem software na zakázku. Hlavní část systému je vytvořen jako desktopová aplikace, další části jsou ve formě webové aplikace a webové služby. Systém využívá .NET Framework, WPF a .NET Core MVC technologie.

Teoretická část práce popisuje problematiku projektového řízení a řeší některých existujících řešení na trhu. Praktická část práce se zabývá analýzou, návrhem, popisem použitých technologií, implementací systému a jeho testováním.

Klíčová slova

Informační systém; Projekty; Řízení projektů; .NET Framework; .NET Core

Abstract

The goal of this bachelor thesis is to create a project management information system for a custom software development company. The main part of system is to be made as a desktop application, other parts as web application and web service using .NET Framework, WPF and .NET Core MVC technologies.

The theoretical part of the thesis describes the issues of project management and research of some of existing solutions. The practical part of the thesis contains an analysis, design, description of used technologies and components, implementation and testing of the system.

Keywords

Information system; Projects; Project management; .NET Framework; .NET Core

Poděkování

Rád bych na tomto místě poděkoval všem, kteří mi s prací pomohli, protože bez nich by tato práce nevznikla, a zejména pak svému vedoucímu práce doc. Ing. Radimu Bačovi, Ph.D za odborné a metodické vedení, ochotu a pomoc v průběhu vypracování práce.

Obsah

Seznam použitých symbolů a zkratk	7
Seznam obrázků	8
Seznam tabulek	10
1 Úvod	11
1.1 Sekvenční a agilní vývojový proces	11
1.2 Stávající stav	12
1.3 Modelový příklad	12
1.4 Struktura práce	13
2 Problematika oblasti projektového řízení	14
2.1 Projekt	14
2.2 Plánování projektu	14
2.3 Projektové řízení	15
2.4 Rešerše existujících řešení na trhu	15
2.5 Shrnutí	17
3 Definice pojmů	19
4 Analýza a návrh	21
4.1 Use case scénáře – procesní diagramy popisující přijetí a zpracování projektu	21
4.2 Přístupová a výkonná práva uživatelů	21
4.3 Návrh datového modelu	23
5 Implementace	25
5.1 Popis architektury systému	25
5.2 Použité návrhové vzory	28
5.3 Použití různých typů databází	28
5.4 Uložení nastavení systému a definice datových klientů	29

5.5	Formát pro přenos dat – JSON	30
5.6	Webová služba – brána pro přístup k MSSQL databázím	30
5.7	Číselníky	35
5.8	Eskalace	35
5.9	Tiskové výstupy	36
5.10	Webová aplikace	37
6	Testování a souhrn dosažených výsledků	39
6.1	Unit test	39
6.2	Systémové testování	41
6.3	Akceptační uživatelské testování (UAT)	43
6.4	Výkonnostní testy	44
6.5	Souhrn výsledků	45
7	Závěr	46
7.1	Shrnutí výsledků práce	46
7.2	Možností dalšího budoucího rozvoje <i>IS ProŘíz</i>	47
	Literatura	48
	Přílohy	48
A	Use case scénáře	49
A.1	UC Nový projekt	49
A.2	UC Vyúčtování prací projektu	50
A.3	UC Detail faktury	53
B	Projektová dokumentace	54
B.1	Spuštění	55
B.2	Speciál	56
B.3	Číselníky	63
B.4	Zákazníci	66
B.5	Úkoly	69
B.6	Projekty	70
B.7	Fakturace	76
C	Ostatní přílohy	86
C.1	Struktura souboru XML pro uložení nastavení aplikace	86
C.2	Ukázka XML souboru Settings.xml	86
C.3	Seznam rozlišení tříd číselníků	88

C.4 Výpis JSON komunikace webové služby	90
D Elektronické přílohy	96
D.1 Příloha v IS EDISON	96

Seznam použitých zkratek a symbolů

.NET	– Aplikační platforma společnosti Microsoft
WPF	– Windows Presentation Foundation
MVC	– Architektura Model–View–Controller
MS	– Microsoft
MSSQL	– Microsoft SQL server
TSQL	– Transact Structured Query Language
CRUD	– Čtyři základní operace nad záznamem v trvalém úložišti (Vytvořit/Create, Číst/Read, Editovat/Update, Smazat/Delete)
DLL	– Dynamic Link Library
XML	– eXtensible Markup Language
MDB	– Souborová databáze MS Access
PDF	– Portable Document Format – souborový formát vyvinutý firmou Adobe pro ukládání dokumentů
JSON	– JavaScript Object Notation – zápis dat určený pro přenos dat
HTTP	– Hypertext Transfer Protocol – internetový protokol umožňující nezábezpečenou komunikaci s WWW servery
HTTPS	– Hypertext Transfer Protocol Secure – internetový protokol umožňující zabezpečenou komunikaci s WWW servery
URL	– Uniform Resource Locator – webová adresa
SMTP	– Internetový protokol určený pro odesílání elektronické pošty
IS	– Informační systém
ARES	– Administrativní registr ekonomických subjektů
API	– Application Programming Interface, rozhraní pro programování aplikací
IČ/IČO	– Identifikační číslo organizace
DIČ	– Daňové identifikační číslo organizace
DPH	– Daň z přidané hodnoty

Seznam obrázků

4.1	Návrh datového modelu	24
5.1	Schéma architektury systému	26
5.2	Ukázka uložení nastavení v XML a JSON	30
5.3	Ukázka JSON obálky požadavku a odpovědi webové služby	32
5.4	Schéma práce webové služby	33
5.5	Sekce webové aplikace	38
5.6	Change log produktu ve webové aplikaci	38
6.1	Systémové testování – Přístupová práva	42
A.1	Diagram aktivit Nový projekt	51
A.2	Diagram aktivit Vyúčtování prací projektu	52
A.3	Diagram aktivit Detail faktury	53
B.1	Hlavní obrazovka aplikace	54
B.2	Výběr databáze po spuštění aplikace	55
B.3	Přihlášení uživatele	55
B.4	Definice uživatele	56
B.5	Nastavení role uživatele	57
B.6	Rozsah omezení položky uživatelské role	58
B.7	Nastavení aplikace	59
B.8	Nastavení SMTP serveru	59
B.9	Nastavení pravidel pro eskalaci projektů	60
B.10	Nastavení pravidel pro eskalaci úkolů	62
B.11	Definice sekce	64
B.12	Definice obchodní struktury	65
B.13	Definice produktu	65
B.14	Ukázka verze je produktu	66
B.15	Číselník států	66

B.16	Definice základních údajů zákazníka	67
B.17	Ostatní údaje a sekce zákazníka	68
B.18	Kontaktní osoby zákazníka	68
B.19	Pobočky a provozovny zákazníka	69
B.20	Bankovní účty zákazníka	69
B.21	Definice fakturace zákazníka	69
B.22	Možnosti filtrování úkolů	70
B.23	Detail úkolu	70
B.24	Možnosti filtrování projektů	71
B.25	Definice základních údajů o projektu	72
B.26	Fáze a etapy projektu	73
B.27	Kalkulace projektu	74
B.28	Výkaz práce projektu	74
B.29	Evidence verzí produktu projektu	75
B.30	Evidence úkolů projektu	76
B.31	Číselník bankovních účtů	77
B.32	Číselník cizích měn	77
B.33	Číselník sazeb DPH	77
B.34	Číselník způsobů úhrad faktur	78
B.35	Číselník způsobů dopravy	78
B.36	Číselník číselných řad faktur	79
B.37	Možnosti vystavení faktury	80
B.38	Záhlaví faktury	81
B.39	Přehled prací vyúčtovaných dokladem	82
B.40	Položky faktury	83
B.41	Definice textů na faktuře	83
B.42	Náhled tiskové sestavy faktury	85

Seznam tabulek

2.1	Přehled splněných požadavků existujících řešení	18
B.1	Mapování polí šablony pro projekty	61
B.2	Mapování polí šablony pro úkoly	63
C.1	Podřízené elementy pro <Databases>	87
C.2	Podřízené elementy pro <Params>	87
C.3	Seznam rozlišení tříd číselníků	88

Kapitola 1

Úvod

Cílem této práce je vytvoření informačního systému pro řízení projektů pro společnost zabývající se vývojem software na zakázku (dále zadavatel). Výsledná desktopová aplikace bude integrovat více existujících systémů a umožní efektivní evidenci a organizaci práce a požadavků.

Informační systém pro řízení projektů – *IS ProŘíz* – musí obsahovat evidenci zákazníků, produktů a projektů, správu požadavků k projektům, rozdělovat projekty a požadavky na etapy a části. Pro každý projekt i jeho části a etapy naplánovat rozsah prací a evidovat provedené práce, provádět vyúčtování prací (všech i jen vybraných). Další součástí *IS ProŘíz* musí být i evidence reklamací a automatické upozorňování na úkoly a termíny. Z hlediska zabezpečení musí *IS ProŘíz* umožňovat řízení přístupu uživatelů k jednotlivým funkcím a při komunikaci přes Internet používat šifrovaná spojení.

1.1 Sekvenční a agilní vývojový proces

Tvorba software, ostatně jako jakákoliv činnost, prochází historickým vývojem a prakticky se v období 30 let, kdy se tvorbou programů zabývám, zcela změnil způsob přístupu k vývoji software. Sekvenční vývojový proces nahradily agilní metodiky vývoje.

Při sekvenčním vývojovém procesu je na vývoj nahlíženo jako na neustále se svažující tok (odtud se tento způsob nazývá vodopádový model) na sebe navazujících fází – analýza požadavků, návrh, implementace, validace (testování a ladění), instalace a údržba. Každý krok se provede celý, než se přikročí k dalšímu. Z hlediska zákazníka je nevýhodou velká prodleva mezi specifikací požadavků a odsouhlasení zadání a instalací programu. Výhodou naopak je, že zákazníkovi se předává více méně hotový produkt. Z hlediska řízení projektu je sekvenční vývojový proces relativně jednoduchý, změny v zadání a požadavcích jsou během tvorby minimální, v ideálním případě žádné, není potřeba zachytávat změny, vydávat a evidovat dílčí verze programu, evidovat všechny změny, vytvářet ad hoc dílčí etapy a fáze, vyúčtování probíhá v pevně daných krocích, případně jen jednou souhrou fakturou.

Při agilní metodiky vývoje se jednotlivé fáze překrývají, zákazníkovi se software dodává průběžně, na základě jeho připomínek se provádí změny zadání a potřebné úpravy, které se opět předkládají zákazníkovi ke schválení a otestování. Toto klade velký nárok na projektové řízení, protože vzniká velké množství souběžných úkolů, které je potřeba evidovat, řešit a vyhodnocovat, vznikají dílčí fáze a etapy, je potřeba evidovat historii všech změn a verzí, sledovat termíny a milníky, je nutno poskytovat zákazníkovi zpětnou vazbu a se zákazníkem pravidelně komunikovat, je tedy potřeba evidovat i tuto komunikaci, vyúčtování může probíhat po jednotlivých etapách, sledují se vícepráce.

1.2 Stávající stav

Stávající evidence zákazníků a řízení projektů, kterou má náš *IS ProŘíz* nahradit, probíhá v několika vzájemně nepropojených a z většiny nekompatibilních a mnohdy zastaralých systémů (z roku 2001). Z důvodu nutnosti oddělení různých obchodních subjektů se evidence většiny částí vede duplicitně (např. fakturace, evidence licencí), řízení projektů probíhá částečně v interně vyvinuté aplikaci a částečně v „souborové evidenci“ (textové soubory), výkaz práce je veden v Excelu, informace o vydaných nových verzích programů s odkazy na stažení nových verzí jsou zasílány jednotlivě e-maily, není evidence vydaných verzí.

Nový informační systém *IS ProŘíz* nahradí všechny zastaralé stávající systémy, kromě evidence licencí, tato musí být oddělena z důvodů provozu přes webovou službu (programy zákazníků jsou aktivovány přes webovou službu) a z důvodu bezpečnosti (vlastních algoritmy šifrování), nový informační systém bude z evidence licencí přebírat a integrovat údaje (zákazníky, termíny vypršení licenci, podklad pro fakturaci).

1.3 Modelový příklad

Popíšme si na modelovém příkladu scénář zpracování požadavku.

Pro zákazníka A byl vytvořen/vytváří se program. Zákazník požaduje rozšíření o modul X. Požadavek se zanes do *IS ProŘíz* a tímto se vytvoří nový projekt. Provede se analýza požadavku, podle povahy se rozdělí na vhodné dílčí etapy a kroky, např. analýza, úprava funkcionalit databáze, úprava funkcionalit aplikace, testování). Zákazníkovi se zašle kalkulace rozsahu prací s předpokládaným objemem člověkohodin. Projekt se v *IS ProŘíz* rozdělí na dílčí etapy a kroky a na tyto se provádí a evidují postupy prací a zejména poznatky/poznámky k jednotlivým krokům. Pro každou etapu a krok jsou v *IS ProŘíz* zaznamenávány všechny poznámky, kroky, pracovní postupy, vč. kusů zdrojových kódů, T-SQL příkazů apod. a to v čase a postupně, jak jsou pořizovány. Po ukončení prací a předání zákazníkovi se provede vyúčtování (fakturace) provedených prací. Vyúčtování může být i dílčí.

1.4 Struktura práce

Tato práce se zabývá vytvořením informačního systému pro řízení projektů. Celá práce je rozdělena do 7 hlavních kapitol včetně úvodní a závěrečné kapitoly. Poslední částí práce jsou přílohy (Use case scénáře A, Projektová dokumentace B a Ostatní přílohy C).

Úvodní část (kapitola č. 2) práce pojednává o řízení projektu a dále obsahuje rešerše existujících řešení na trhu.

V další části (kapitola č. 3) jsou vysvětleny důležité pojmy.

V analýze a návrhu (kapitola č. 4) se zabývám analýzou požadavků systému a návrh jejich řešení.

V části Implementace (kapitola č. 5) je popsána vlastní implementace systému.

V kapitole č. 6 je popsáno testování *IS ProŘíz* se souhrnem dosažených výsledků.

Poslední kapitolou č. 7 je závěr práce, který obsahuje shrnutí výsledků práce, zhodnocení splnění cílů a možností dalšího budoucího rozvoje.

Kapitola 2

Problematika oblasti projektového řízení

V této části bude vysvětlena problematika řízení projektů a budou popsány a porovnány některé existující systémy pro správu projektů.

2.1 Projekt

Projekt je soustava činností s jasně definovaným cílem, který má určený začátek a konec. Projekt vyžaduje spoluprací různých profesí, váže jejich kapacity a pro dosažení cílů využívá a spotřebovává informace, materiál, finance, schopnosti a dovednosti zúčastněných lidí. [1]

2.2 Plánování projektu

Aby bylo možné projekt úspěšně zrealizovat, musí se projekt řídit. K řízení projektu je nutný nějaký scénář, což jsou plány projektu.

Plánování je postupná odpověď na otázky v tomto pořadí:

1. CO,
2. JAK,
3. S KÝM,
4. KDY,
5. ZA KOLIK.

Nejprve se musí přesně naplánovat, CO se má udělat. Je potřeba jasně a jednoznačně zformulovat, co bude výsledkem (cílem) projektu a případně které činnosti budou vykonány. Po stanovení cíle projektu je nutno popsat postup, JAK cíle dosáhnout. Určí se, co je potřeba udělat nejprve, co potom, co lze dělat současně a co musí být hotovo, aby bylo možno pokračovat dál. V této části se

neřeší otázka času, kdy se který krok udělá. Při plánování postupu JAK dosáhnout cíle projektu je výhodné použít třístupňové hierarchie:

1. Etapy,
2. Kroky,
3. Úkony.

Etapy následují za sebou sekvenčně, projekt přechází do nové etapy vždy po ukončení té předchozí. V rámci etapy se provádí kroky, které mohou být prováděny i současně. Každý krok vytvoří nějaký dílčí produkt, komponentu, apod. Kroky se stejně jako etapy plánují na celý projekt.

Nejmenšími díly práce jsou úkony. Tyto jsou přidělovány jednotlivcům nebo malým týmům. Úkony se neplánují dále, než na následující etapu. Neustálé změny, které projekt vždy provází, by způsobily, že by se detailní plán úkonů v pozdních etapách neustále předělával.

Po stanovení CO a JAK se bude dělat je dostatek údajů pro určení S KÝM se co bude dělat. Povahy cílů a zvolené postupové činnosti určují množinu profesních znalostí, schopností a dovedností, které budou pro realizaci projektu potřeba. Tyto se obsadí konkrétními lidmi, čímž vznikne tým projektu. Po stanovení týmu lze teprve odpovědět na zbylé dvě otázky – KDY a ZA KOLIK se projekt zrealizuje. [1]

2.3 Projektové řízení

Řídit projekt znamená způsobit, že co je naplánováno, bude taky uděláno. Pomocí etap projektu řídíme hlavně riziko, zejména riziko zbytečně proinvestovaných peněz. Proto etapy následují bez překrývání jedna za druhou.

Každý krok etapy projektu vytváří nějaký dílčí produkt, jehož kvalitu lze jednoznačně vyhodnotit tak, aby bylo možno posoudit, zda práce vede k očekávaným výsledkům. Je-li výsledek kroku ve shodě se specifikací, projekt pokračuje podle plánů. Není-li tomu tak, musí se naplánovat a provést nápravné práce.

Vlastní průběh prací na projektu se řídí pomocí úkonů. Úkony mohou probíhat paralelně a provedení určité množiny úkonů znamená vykonání nějakého kroku. Úkony jsou přidělovány manažerem projektu a sledovány pomocí rozpisu prací. Během realizace úkonů se provádí kontrola kvality vytvářeného dílčího produktu či komponenty. [2]

2.4 Rešerše existujících řešení na trhu

V této části budou popsány některá existující řešení informačních systémů pro správu projektů. U každého budou uvedeny základní informace o řešení, jaké požadavky splňuje a jaké jsou jeho případné výhody a nevýhody. Závěrem proběhne shrnutí získaných poznatků se závěrem, zda stávající existující řešení splňují naše požadavky.

2.4.1 PROJEKTOVĚ.CZ

Projektově.CZ s.r.o. [3] je ostravská společnost, která dlouhodobě spolupracuje s univerzitou Vysoká škola báňská v Ostravě. Ve spolupráci s odborníky na projektové řízení, ať už z akademických řad nebo od partnerských firem, poskytuje nástroj pro přehledné řízení firmy, projektů a úkolů.

Služba Projektově.CZ je nástrojem projektového managementu pro řízení firmy, plánování a řízení projektů, zakládání projektů ze šablon a úkolování v rámci běžné provozní agendy. Celý systém běží na internetovém serveru v cloudu. Službu Projektově.CZ je také možné při zakoupení programu Projektově.CZ On–premise nainstalovat na firemní servery a následně služba běží uvnitř firemní sítě. Součástí Projektově.CZ On–premise je i neomezená licence (neomezený počet uživatelů, diskový prostor je v režii klienta).

Systém je primárně zaměřen na správu úkolů. Úkoly jsou zpracovány velmi dobře a přehledně, obsahují dostatek možností pro zadání potřebných informací. Systém hlídá termíny úkolů, tyto jsou automaticky připomínány. Projekty jsou v systému spíše jen „obalem“, jsou reprezentovány názvem, popisem, zkratkou a položkou související s „cíli“. Do projektů lze úkoly přidávat, systém poskytuje informace a souhrny o dosažených úkolech, nelze ale k projektu například připojit soubory, což by mohlo být použito pro projektovou dokumentaci či pro tvorbu change logu produktu.

Nevýhody systému jsou chybějící modul fakturace, chybějící evidence verzí produktů (programů), nemožnost zákaznických sekcí pro stahování nových verzí a change log produktů. Provoz na doméně společnosti 321 má na jedné straně výhodu v snadném nasazení systému (nemusí se nic instalovat), zároveň se ale jedná o nevýhodu z pohledu bezpečnosti dat a přístupu k nim.

Cena je podle počtu uživatelů, měsíční poplatky se pohybují od 390 Kč za dva uživatele po 6890 Kč za 50 uživatelů.

2.4.2 321 INTRA

321 INTRA [4] je jednoduchý online nástroj pro firemní komunikaci, řízení zakázek a správu dokumentů na jednom místě bez nutnosti instalace od společnosti 321 CREATIVE CREW, s.r.o. Brno.

Systém je provozován jako webová aplikace na stránkách 321 INTRA, konkrétně na doméně 321intra.com, kde každá „instalace“ běží na vlastní subdoméně. Systém obsahuje přehled o zákaznících, informace o zákazníkovi se mohou předvyplňovat z databáze ARES. Ke každému zákazníkovi lze přidat fotografii (logo), kontaktní informace a jakékoliv další poznámky.

V rámci projektového řízení je možné nastavit termíny (vytvoření, začátek, lhůta, deadline), prioritu úkolu / projektu a popis. Dále je možno delegování jednotlivých úkolů na řešitele, sdílet podklady a vést diskuzi k úkolům projektu. Každý úkol může mít vlastní rozpočet a lze k němu tvořit časový výkaz pro fakturaci. Součástí systému je i modul technické podpory ve formě techhelp ticketů s možností delegování požadavků na řešitele a on–line komunikaci se zákazníkem. Projektové

řízení není klíčová vlastnost systému, je jen jedna z položek, podstatnější část systému se zabývá vnitrofiremní komunikací, firemními zprávami a e-mail, on-line docházkou, apod.

Nevýhody systému jsou chybějící modul fakturace, chybějící evidence verzí produktů (programů), neexistence zákaznických sekcí pro stahování nových verzí a change log produktů. Provoz na doméne společnosti 321 má na jedné straně výhodu v snadném nasazení systému (nemusí se nic instalovat), zároveň se ale jedná o nevýhodu z pohledu bezpečnosti dat a přístupu k nim.

Základní cena za uživatele a měsíc je 259 Kč.

2.4.3 Bugzilla

Bugzilla [4] je webová aplikace pro sledování chyb (bug tracking) původně vyvinutá a používaná organizací Mozilla. Jedná se o open source software s licencí Mozilla Public Licence. Bugzilla potřebuje pro svůj chod vlastní webový server (například Apache) s podporou Perl a databázový systém (například MySQL).

Základem systému je hlášení chyby (bugu). Pro chybu se určí klasifikace, produkt a komponenta, ke kterým se vztahuje. Dále lze nastavit prioritu, termín splnění (deadline), stav a přiřadit uživatele. Také je možno nastavit propojení na další bug, čímž lze provádět dělení na etapy. K přehledu práce slouží odhad pracnosti (kalkulace) a odpracované hodiny. K bugu lze nahrávat přílohy, které lze označit jako novou verzi komponenty, případně že nahrazuje starší přílohu. Systém dále umožňuje nastavení e-mailových notifikací.

Za nevýhody systému lze označit určení primárně pro Linux, chybějící evidence zákazníků s možností předvyplnění údajů z databáze ARES (zřejmě by místo zákazníka šla použít klasifikace chyby), není podpora MSSQL, je nutnost dodatečné implementace například vystavování a tisku faktur. Nasazení systému by znamenalo poměrně velké nároky na seznámení se způsobem instalace a implementace tohoto systému.

Systém je poskytován zdarma.

2.5 Shrnutí

V této kapitole byly předloženy požadavky na nový informační systém pro správu projektů a byly porovnány s některými existujícími řešeními, které jsou na trhu k dispozici. Přehled porovnání splněných požadavků existujícími řešeními je uveden v tabulce 2.1.

Z uvedených tří systémů se k požadavkům na nový informační systém pro řízení projektů nejvíce blíží Bugzilla, zbývající dva systémy nemají řízení projektu tak propracováno. U každého ze tří systémů by bylo potřeba rozšíření o chybějící moduly, jak bylo popsáno u nevýhod systémů. U systému Bugzilla by rozšíření znamenalo časově náročné seznámení se systémem, aby bylo možné rozšíření provést, u zbývajících dvou by se rozšíření muselo řešit objednávkou prací u dodavatelů systémů.

Tabulka 2.1: Přehled splněných požadavků existujících řešení

<i>Požadavek</i>	<i>PROJEKTOVĚ.CZ</i>	<i>321 INTRA</i>	<i>Bugzilla</i>
Evidence zákazníků s možností automatického předvyplnění údajů z databáze ARES	Ano	Ano	Ne
Projektové řízení - možnost nastavení priority, stavu a termínů, dělení na etapy, kalkulace člověkohodin	Ano	Ano	Ano
Pracovní výkaz	Ano	Ano	Ano
Vyúčtování prací, vystavení a tisk faktur	Ne	Ne	Ne
Eskalace – upozorňování na úkoly, termíny (e-mailem)	Ano	Ano	Ano
Informace o vydaných verzích SW pro jednotlivé projekty (change log)	Ne	Ne	Ano
Možnost zákaznické sekce na webu chráněné heslem pro stažení nových verzí SW zákazníky	Ne	Ne	Ano
Provoz databáze na vlastním serveru	Ne (v základu)	Ne	Ano

Při porovnání celkových nákladů na instalaci, nasazení a servis systémů vychází systém Bugzilla nejhůře, systém je sice poskytován zdarma, ale instalaci, veškerá nastavení, případné úpravy, hardware a servis jsou plně v interní režii, znamená to se na tyto činnosti vyškolit a tyto činnosti provádět. Žádná z těchto činností není předmětem činnosti zadavatele. Zbývající dva systémy běží v základní verzi na zařízeních dodavatelů, respektive na serverech v cloudu.

Z hlediska nejdůležitějšího hlediska – bezpečnosti dat – je jediné možné řešení provoz databázi na vlastním databázovém serveru. K databázi a do systému nesmí mít přístup žádná třetí strana, některé projekty podléhají režimu utajení a se zákazníky je podepsána dohoda o mlčenlivosti. Součástí dohod je i závazek snížit možnost úniku jakýchkoliv informací na co možná nejmenší úroveň, což v případě umístění databáze na externích serverech nelze garantovat. Provoz na vlastním databázovém systému nabízí v základní verzi pouze Buggzila. U Projektově.CZ je tento způsob provozu databáze možný jen při zakoupení rozšiřujícího programu On–Premise. Ceny i podmínky tohoto programu jsou individuální, jejich konkrétní výše a podoba je Projektově.CZ určena až na základně poptávky.

Kapitola 3

Definice pojmů

V této části budou vysvětleny pojmy důležité pro správné pochopení procesů v *IS ProŘíz* .

Uživatel – Uživatel je uživatelský účet sloužící pro přihlášení do *IS ProŘíz* pomocí přihlašovacího jména a heslo a k identifikaci uživatele v systému. Pro uživatelský účet lze definovat pomocí uživatelských rolí a obchodní struktury a rozsah zobrazovaných informací.

Uživatelská role – Uživatelská role slouží k definici přístupových a výkonných práv uživatele. Uživatel může být zařazen do více rolí.

Produkt – Produktem se rozumí jednotlivá aplikace, program nebo komponenta softwarového balíku. Každý produkt je zařazen do právě jedné sekce.

Sekce – Sekce sdružuje produkty do skupin produktů (např. Účetnictví, Mzdy, Skladové hospodářství, Majetek náleží do softwarového balíku), sekci také může být samostatný program či aplikace. Sekce může mít zastupovat jinou sekci, u některých zákazníků je jiný obchodní název pro produkt/skupinu produktů. V sekci je rovněž vedena evidence verzí produktů a historie změn (takzvaný change log). Každý zákazník může mít libovolný počet sekcí.

Zákazník – Zákazník je ten, pro kterého se vyvíjí produkt. Zákazník může mít více kontaktních osob.

Obchodní struktura – Obchodní struktura definuje, které zákazníky a jaké sekce uživatel vidí a má k nim přístup.

Fakturace za dodavatele – Fakturace za dodavatele je vystavení faktury jménem zákazníka.

Účetní jednotka – Účetní jednotka je výchozí dodavatel systému v evidenci faktur, definuje se v Nastavení (viz kapitola B.2.3, str. 58 v příloze) výběrem ze seznamu zákazníků.

Webová služba – *Webová služba* [5] je softwarový systém umožňující komunikaci dvou strojů na síti nad protokolem HTTP/HTTPS. S webovou službou ostatní stroje komunikují způsobem, který je předepsaný v definici služby.

ARES – *ARES* [6] (Administrativní registr ekonomických subjektů) je informační systém, který umožňuje vyhledávání nad ekonomickými subjekty registrovanými v České republice. Zprostředkovává zobrazení údajů vedených v jednotlivých registrech státní správy, ze kterých čerpá data (tzv. zdrojové registry). Součástí informačního systému ARES je i XML rozhraní pro vyhledání subjektu a zpřístupnění jeho veřejných údajů ze zdrojových registrů.

Eskalace – Eskalace je systém stupňování upozornění na termín úkolu nebo projektu.

Token – Token je typ autorizačního hesla, používá se k zajištění bezpečnosti dat. Token je unikátně generován pro každé připojení a jeho platnost je obvykle časově omezena.

Smazání záznamu – Smazáním záznamu se rozumí jeho označení v databázi příznakem, že je smazán s informací kdy a kým tak bylo učiněno, fyzicky v databázi zůstává.

DPH – DPH je daň z přidané hodnoty, která je placena při nákupu většiny zboží a služeb. Aktuálně jsou v České republice definovány tři pásma sazeb: základní ve výši 21 %, snížená 15 % a druhá snížena 10 %.

Kapitola 4

Analýza a návrh

Tato část práce obsahuje analýzy a návrhy. Jsou zde popsány etapy prací při vývoji *IS ProŘíz*, návrh procesních diagramů popisující přijetí a zpracování projektu, přístupová práva uživatelů a návrh struktury databáze potřebné pro uložení všech dat.

4.1 Use case scénáře – procesní diagramy popisující přijetí a zpracování projektu

Use case scénáře popisující přijetí a zpracování projektu se nacházejí v příloze A.

4.2 Přístupová a výkonná práva uživatelů

Při řešení uživatelských rolí jsou možné dva přístupy. Prvním jsou pevně definované role, kdy role určuje typ uživatele (například Uživatel, Administrátor, Manažer). Druhý přístup je pro každou roli mít možnost uživatelsky definovat rozsah oprávnění a uživatele do rolí přiřazovat.

Výhodou pevně definovaných rolí je snadnější implementace, každá role má předem daná oprávnění, zařazení uživatele do role se provede jednoduchým výběrem ze seznamu rolí. Nevýhodou je nemožnost změny oprávnění rolí bez změny kódu, rekompile a instalace nové verze aplikace, nutnost důkladné analýzy rozsahu oprávnění jednotlivých rolí před implementací a obtížná možnost kombinace rolí.

V případě *IS ProŘíz* nelze pevně definované uživatelské role použít, protože každý uživatel může mít rozdílný rozsah využití systému a požadavků na rozsah jeho činností. Vyžaduje se tedy dynamické nastavení práv uživatele. Dále je požadována možnost omezení rozsahu zobrazovaných a editovatelných záznamů například podle vlastnictví záznamu. Vlastnictvím záznamu se rozumí, kterému uživateli je záznam přidělen, či jej má na starost a zodpovídá za něj.

K definici přístupových a výkonných práv uživatele budou tedy sloužit uživatelsky nastavitelné číselníky uživatelských rolí. Uživatel bude moci být zařazen do více rolí. V případě zařazení uživatele do více rolí má uživatel přístup k záznamu, pokud je přístup povolen alespoň v jedné roli.

Pro každou položku uživatelské role bude možno nastavit:

Oprávnění pro přístup – definuje, zda uživatel má do záznamu přístup a bude mu přístup zobrazen v menu.

Oprávnění pro přidání – definuje, zda uživatel může pořizovat nové záznamy.

Rozsah zobrazovaných záznamů – omezení rozsahu zobrazovaných záznamů seznamu (více viz 4.2.1).

Rozsah editovatelných záznamů – omezení rozsahu záznamů seznamu, které uživatel může editovat (více viz 4.2.1).

Oprávnění pro mazání – zda uživatel může záznam/y mazat.

Tímto způsobem je možno nastavit prakticky libovolnou kombinaci práv uživatele, lze vytvářet skupiny uživatelů se stejnými právy a to bez nutnosti programových úprav.

4.2.1 Rozsahy povolení

Rozsah povolení určuje, které záznamy má uživatel přístupné. Základní možnosti jsou **Vše** anebo **Žádné**.

Je-li možné určit vlastníka záznamu, rozsah je rozšířen o možnost omezení dle vlastnictví záznamu:

Vlastní – omezení jen na záznamy vlastněné uživatelem.

Vlastní skupina – omezení jen na záznamy vlastněné uživatelem a dalšími uživateli stejných uživatelských rolí jako uživatel.

Vlastní struktura – omezení jen na záznamy z obchodní struktury uživatele bez ohledu na vlastníka záznamu.

Pokud je možné určit aktivní status záznamu, je rozsah omezení rozšířen o možnosti zobrazení jen aktivních záznamů. Aktivní záznam je určen prostřednictvím statusu záznamu, v případě úkolů se jedná o nesplněný úkol. Pro každý status lze v příslušném číselníku statusů určit, zda se jedná o aktivní status.

4.3 Návrh datového modelu

Návrh datového modelu *IS ProŘíz* je zobrazen na obrázku 4.1.

Schéma lze rozdělit na několik celků – uživatelský subsystém, informace o zákaznících, subsystém projektů a fakturační subsystém.

4.3.1 Seznam tabulek

V této části je stručný popis všech tabulek v databázi.

Address – do tabulky se ukládají adresy a kontaktní informace pro zákazníky a osoby, uživatele a adresy na faktuře.

Country – tabulka obsahuje seznam států.

User – tabulka obsahuje seznam uživatelů.

UserRole – tabulka obsahuje seznam uživatelských rolí.

LoginLog – tabulka slouží pro uložení informací o přihlášení uživatele do *IS ProŘíz* .

Customer – tabulka obsahuje identifikační údaje o zákaznících.

Person – tabulka obsahuje seznam kontaktních osob zákazníka.

Project – tabulka slouží pro uložení informací o projektu. Projekt může zároveň být i etapa, krok či úkon. U etapy, kroku a úkonu je vždy definován nadřazený objekt, tímto se vytváří stromová struktura definice plánování projektu. Počet úrovní zanoření je neomezen.

ProjectWork – tabulka obsahuje seznam provedených prací na projektu, etapě, kroku či úkonu.

Invoice – tabulka obsahuje hlavičky faktur.

InvoiceItem – tabulka obsahuje položky faktur.

GeneralList – tabulka slouží k uložení všech číselníků a parametrů. Způsob uložení číselníků v tabulce je popsán v popisu implementace číselníků v kapitole 5.7.

Kapitola 5

Implementace

V této části práce je popsána architektura systému, použité návrhové vzory, nástroje použité při vývoji a dále obsahuje seznámení s klíčovými částmi systému.

5.1 Popis architektury systému

IS ProŘíz využívá principu třívrstvé architektury. Desktopová část *IS ProŘíz* je vytvořena pomocí technologie WPF (Windows Presentation Foundation) .Net Framework a webové části pomocí technologie .NET Core na architektuře MVC (Model-View-Controller). Ostatní komponenty systému jsou vytvořeny v .NET Framework. Podrobnější informace o architektuře a použitých technologiích jsou popsány v následujících podkapitolách.

Při vývoji byl kladen důraz na dekompozici. Části systému, které bylo možné vyčlenit do samostatných modulů, jsou vytvořeny jako knihovny tříd (anglicky Class library) kompilované do jednotlivých DLL souborů. Výhodou je přehlednost, snadnější vývoj a testování a zejména možnost sdílení komponent jednotlivými částmi systému.

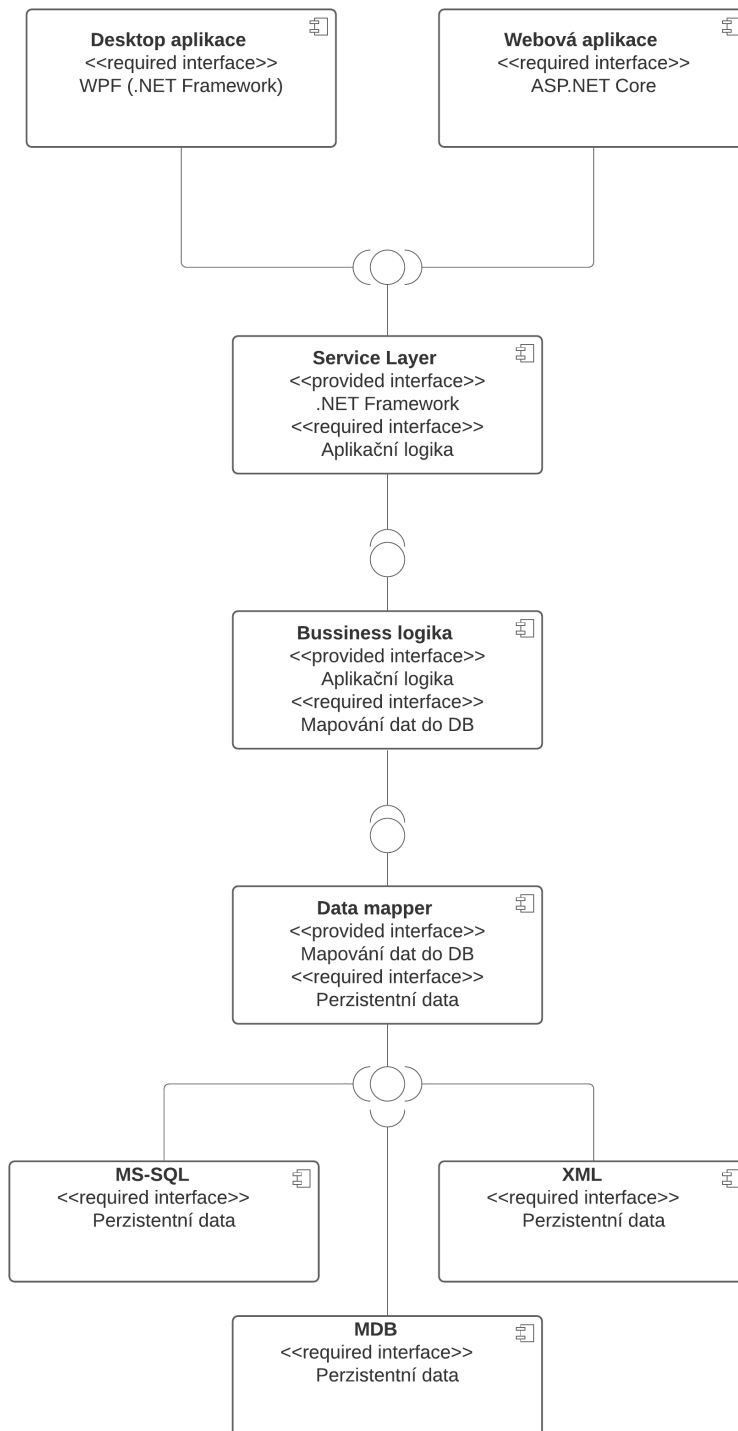
Jako prostředí pro vývoj aplikace bylo použito Microsoft Visual Studio 2017 verze 15.9.30, všechny jeho části jsou napsány v jazyce C#.

Schéma architektury systému je zobrazen na obrázku 5.1.

5.1.1 Třívrstvá architektura

Třívrstvá architektura (anglicky Three-tier architecture) je jeden z typu architektur informačních systémů. Aplikace netvoří jeden celistvý program, ale je rozdělena do více samostatných vzájemně spolupracujících vrstev, které mohou běžet na různé výpočetní infrastruktury. Sousedící vrstvy spolupracují přes definovaná rozhraní a mohou být zaměňovány bez dopadu na funkčnost celé aplikace.

Vrstvy této architektury jsou:



Obrázek 5.1: Schéma architektury systému

Prezentační vrstva – zabezpečuje komunikaci s uživatele. Zobrazuje informace pro uživatele formou grafického uživatelského rozhraní, může kontrolovat zadávané vstupy a předává požadavky do aplikační vrstvy.

Aplikační vrstva – též business logika či doménová logika, zpracovává informace a provádí jejich dočasné uchování. Obsahuje veškerou logiku aplikace, funkce, výpočty a zpracování dat.

Datová vrstva – přístup k datům. Zajišťuje trvalé uchování informací (dat), obsahuje základní datově-funkční CRUD operace.

5.1.2 WPF

WPF (Windows Presentation Foundation) je knihovna tříd pro tvorbu grafického rozhraní, jedná se o nástupce Windows Forms.. Je součástí .NET Frameworku firmy Microsoft od verze 3.0. Prostředí WPF je určeno pouze pro desktopové aplikace v prostředí Windows. WPF podporuje i starší knihovny, je možná spolupráce s Windows Forms.

Veškerá grafika včetně samotných WPF oken funguje pomocí Direct3D knihoven. To umožňuje hardwarovou akceleraci pomocí grafické karty a pokročilejší grafické schopnosti. Objekty vektorové grafiky jsou popsány matematicky, je tedy možné jejich plynulé přibližování bez rozmazávání.

WPF jsem zvolil kvůli modernějšímu vzhledu a pro větší možnosti nastavení vlastností u prvků při tvorbě uživatelského rozhraní, než umožňují starší Windows Form.

5.1.3 .NET Core

.NET Core je nástupcem technologie .NET Framework společnosti Microsoft. Na rozdíl od .NET Framework již není vázán pouze na operační systém Windows, je multiplatformní a je open source. Dalšími výhodami jsou lepší výkon, možnost provozovat více verzí frameworku v jednom stroji a fakt, že .NET Core není nutno instalovat. Nevýhodou je částečná nekompatibilita se starším .NET Framework a menší dostupnost balíčků a knihoven třetích stran. [7]

5.1.4 MVC architektura

Základní myšlenkou MVC (Model-View-Controller) architektury je oddělení logiky od výstupu. Celá aplikace je rozdělena na komponenty tří typů – Model, View a Controller. Hlavní výhodou je oddělení HTML kódu od zbytku aplikace, což přináší jednodušší testování a údržbu, celek je pro vývojáře přehlednější. [7]

Popíšme si jednotlivé komponenty podrobněji:

- **Model** obsahuje logiku a vše, co do ní spadá. Modely jsou klasické třídy reprezentující podobu dat, se kterými aplikace pracuje. Modely se mohou chovat také jako domény, mohou obsahovat operace (metody) a provádět transformace dat. Úlohou modelu není získávání či zobrazení dat.

- **View** (pohled) se stará o zobrazení výstupu uživateli. Jedná se o HTML šablonu obsahující definici HTML stránky. Do šablony lze pomocí speciálních tagů vkládat proměnné, provádět iterace a cykly. Pohledy obsahují pouze logiku potřebnou pro zobrazení dat.
- **Controller** má za úkol přijímat požadavky od uživatele, provádět operace s modely a vybírat pohledy, kterými budou data zobrazena.

5.2 Použité návrhové vzory

V implementaci jsou použity následující *návrhové vzory* [8]:

Data mapper – pracuje přímo s objekty tříd. Doménový objekt neobsahuje žádné CRUD operace, o tyto operace se stará oddělený (mapovací) objekt, doménový objekt je zcela nezávislý na databázi.

Layer Supertype – společný předek, interface, kterého dědí všechny třídy. Jedná jako rodič pro všechny typy v dané vrstvě, poskytuje funkce, které mají všechny třídy společné. V systému je implementováno ve třídě BaseHeader (obsahuje ID, příznak Smazáno a informace kdo a kdy založil, změnil a smazal).

Identity Map – každý objekt je načten pouze jednou do Mapy (kolekce), pokud objekt neexistuje, zažádá se o něj v datové vrstvě.

Identity Field – uloží databázové id do objektu jako atribut. Zajišťuje synchronizaci mezi DB a objektem.

Serialized LOB – uloží serializovanou skupinu nebo strukturu objektů do jednoho záznamu v databázi.

Single Table Inheritance – všechna data z jedné objektové hierarchie tříd jsou uložena v tabulce *GeneralList*, v tabulce je označen typ záznamu. Popis uložení číselníků je popsán v kapitole 5.7.

5.3 Použití různých typů databází

Jedním s požadavků *IS ProŘíz* je možnost používat jako zdroj dat různé databázové zdroje. Důvodem je potřeba oddělení testovacích dat od produkčních a dále možnost jako datovou vrstvu použít různé druhy databází, respektive jakýkoliv typ databáze. Nezávislost na konkrétním databázovém řešení (např. MSSQL) je požadováno kvůli možnosti instalace *IS ProŘíz* u dalších zákazníků a dále možnosti provozu i bez databázového serveru s použitím například souborové databáze.

Toto řešení vyžadovalo oddělit vlastní komunikaci s databází z data mapperů do další vrstvy a všechny CRUD operace v data mapperech definovat pomocí interface databáze. Interface databáze

obsahuje tedy jen definici povinných tříd a metod. Pro každý konkrétní typ databáze je vytvořeno samostatné assembly (DLL soubor) implementující interface databáze. Toto umožňuje použít libovolnou databázi, stačí jen vytvořit pro příslušnou databázi novou implementaci interface databáze a v definici datových klientů (viz kapitola 5.4) určit název DLL souboru, který se má pro komunikaci s databází použít.

Příslušný DLL soubor je dynamicky načten a pomocí reflexe je provedena kontrola, zda implementuje požadované rozhraní. Implementace interface databáze obsahuje i kontrolu datové struktury a její případné založení nebo úpravy, pokud to povaha databázového systému vyžaduje, respektive umožňuje (např. u XML souborové databáze není kontrola a založení struktur potřebná). V rámci bakalářské práce jsou plně implementovány databáze MSSQL, souborové databáze typu MDB a XML.

Speciálním případem připojení k MSSQL databázi je přístup prostřednictvím webové služby (více v kapitole 5.6). Přístup prostřednictvím webové služby se uplatní automaticky, pokud není databázový server dostupný lokálně a je definován přístup přes webovou službu.

5.4 Uložení nastavení systému a definice datových klientů

Pro uložení nastavení aplikace jsem zvolil ukládání do formátovaného textového souboru. Je to kvůli možnosti jednoduché editace bez nutnosti vytvářet další podpůrnou aplikaci, protože nastavení dat je nutno provést před spuštěním aplikace.

Z různých možných formátů jsem vybíral mezi formátem XML a JSON. XML formát je oproti JSON poměrně snadno čitelný pro člověka, JSON formát vhodný více pro strojové zpracování, používá více speciálních znaků, má striktní strukturu a některé znaky je potřeba speciálně zadávat (např. zpětné lomítko musí být zdvojené). Na obrázku 5.2 je srovnání obou formátů pro uložení shodných dat. U JSON formátu (obrázek 5.2b) je ve druhém záznamu záměrně syntaktická chyba ve struktuře „ServerName“. Více o problematice přenosů dat pomocí JSON formátu pojednává kapitola 5.5.

Primárním kritériem pro výběr mezi formáty XML a JSON je v tomto případě čitelnost a snadnější zpracování souboru člověkem a proto jsem zvolil formát XML. Nastavení aplikace a definice datových klientů je tedy uloženo v XML souboru s názvem Settings.xml.

Kořenovým (root) elementem XML dokumentu je <Settings>. V rámci root elementu je možno definovat dvě základní sekce <Databases> a <Params>. Element <Databases> je určen pro definici databází, které aplikace může používat, jednotlivé databáze jsou definovány v elementu <Client ID="#id">, kde id je unikátní identifikační číslo databáze. Struktura podřízených elementů je vypsána v tabulce C.1 v příloze na straně 87. Element <Params> je určen pro definici databáze, pokud nejsou databáze definovány v elementu <Databases> a pro případné další globální parametry aplikace, struktura podřízených elementů je vypsána v tabulce C.2 v příloze na straně 87. Ukázka XML souboru je v příloze C.1 na straně 86.

<pre> <?xml version="1.0"?> <DataClientList> <DataClient> <ID>1</ID> <Name>ISProRiz DEVELOPER</Name> <ServerName>SQLSERVERS\MSSQL2019</ServerName> <Catalog>ISProRiz_DEV</Catalog> </DataClient> <DataClient> <ID>2</ID> <Name>ISProRiz PROD</Name> <ServerName>SQLSERVERS\MSSQL2019</ServerName> <Catalog>ISProRiz_PROD</Catalog> </DataClient> </DataClientList> </pre>	<pre> [{ "ID": 1, "Name": "ISProRiz DEVELOPER", "ServerName": "SQLSERVERS\MSSQL2019", "Catalog": "ISProRiz_DEV", }, { "ID": 2, "Name": "ISProRiz PROD", "ServerName": "SQLSERVERS\MSSQL2019", "Catalog": "ISProRiz_PROD", }] </pre>
(a) Ukázka uložení nastavení v XML	(b) Ukázka uložení nastavení v JSON

Obrázek 5.2: Ukázka uložení nastavení v XML a JSON

5.5 Formát pro přenos dat – JSON

Pro přenos dat jsem se stejně jako u uložení nastavení aplikace rozhodoval mezi dvěma formáty – XML a JSON.

XML je strojově dobře čitelné a výhodou je i poměrně dobrá čitelnost pro člověka. Z hlediska přenosu dat je ale nevýhodou XML jeho velikost. Obsahuje pro strojové zpracování poměrně velké množství nadbytečného obsahu, každá hodnota je vždy „obalena“ svým názvem. Proto se v současné době zejména u přenosů dat stále více prosazuje formát JSON (JavaScript Object Notation). Ukázka srovnání formátů XML a JSON je na obrázku 5.2.

JSON je formát primárně určený pro přenos dat, je u něj kladen důraz na jeho velikost, která je oproti XML zmenšena pouze syntaxí a nikoliv kompresí dat. JSON má dvě struktury – seznam hodnot (reprezentují se jako seznam či pole) a kolekci párů název a hodnota (reprezentují se jako objekt, slovník, struktura, atd.). Tyto struktury lze kombinovat, lze tedy vytvářet i seznamy objektů.

Krom menší velikosti JSON oproti XML při přenosu stejného objemu dat je další výhodou jednodušší strojové zpracování. JSON se snadno mapuje na datové struktury moderních programovacích jazyků jako C#, Python, Java, atd. Tyto jazyky obsahují nástroje pro efektivní práci programátora, převod do (serializace) a z (deserializace) JSON se provádí voláním jedné metody.

V *IS ProŘíz* se JSON používá pro přenos dat mezi klientem a webovou službou, která poskytuje bránu pro přístup k MSSQL databázím. O této webové službě pojednává kapitola 5.6.

5.6 Webová služba – brána pro přístup k MSSQL databázím

Další problematikou, kterou je v *IS ProŘíz* potřeba řešit, je plně šifrovaný přístup do MSSQL databáze prostřednictvím webové služby. Tento přístup bude použit v případech, kdy desktopová aplikace nemá přímý přístup do MSSQL databáze a je tedy potřeba k databázi přistupovat přes Internet. Zvažoval jsem vytvoření webové služby pomocí ASP.NET Web Services s použitím protokolu SOAP při komunikaci anebo navrhnout řešení vlastní.

SOAP (Simple Object Access Protocol) je protokol určený k zasílání zpráv a volání vzdálených procedur nad existujícím transportním protokolem (např. HTTP). Protokol využívá syntaxi jazyka XML. SOAP protokol nepopisuje způsob komunikace se službou, k tomu je určen WSDL (Web Service Description Language), který ve formátu XML popisuje rozhraní služby, typy argumentů a návratových hodnot.

Výhodou SOAP komunikace je standardizace a přímá podpora v .NET Framework, tedy poměrně jednoduchá implementace v prostředí ASP.NET. Standardizace může být paradoxně i nevýhodou, potencionální útočník se pomocí WSDL snadno dostane k popisu služby. Další relativní nevýhodou je komunikace pomocí XML protokolu (o této problematice pojednává kapitola 5.5) a závislost na platformě Windows.

Po zvážení všech výhod a nevýhod jsem zvolil vytvoření vlastního řešení založeného na JSON formátu se šifrování požadavků v prostředí .NET Core s architekturou MVC. Použitím .NET Core je řešení nezávislé na platformě a díky JSON formátu i úspornější na objem přenesených dat. Mimo to .NET Core MVC nativně podporuje konverzi JSON formát na příslušný model. Dále je díky použití MVC řešení snadno rozšiřitelné o další typy databází, stačí vytvořit další controller a upravit příslušnou DLL komponentu pro komunikaci s databází (mapper), aby využívala i přístup k databázi přes webovou službu.

5.6.1 Popis řešení

Řešení klade důraz na zabezpečení a zamezení neoprávněného přístupu k datům. První linie je použití protokolu HTTPS pro veškerou komunikaci přes Internet, druhá linie je zabezpečení pomocí veřejného klíče sloužící jako heslo pro přístup do webové služby a třetí je pak šifrování požadavků, jak je popsáno dále. Pro komunikaci je v obou směrech používán protokol JSON. Vytváření a zpracování JSON komunikace je uloženo ve vlastní DLL komponentě, která je sdílena webovou službou a MSSQL mapperem.

Požadavek je webové službě odeslán ve formě JSON obálky s pevnou strukturou hodnot. Šifrování a dešifrování probíhá pomocí veřejného a soukromého klíče. Ukázka JSON obálky požadavku a odpovědi webové služby je na obrázku 5.3.

5.6.1.1 JSON obálka

Každý požadavek je webové službě odeslán formou JSON obálky s pevně danou strukturou.

Popis struktury JSON obálky:

Token – vygenerovaný token připojení (session).

PublicKey – veřejný klíč.

Checksum – kontrolní součet. Zašifrovaný kontrolní řetězec.

Data – odesílaná data. Zašifrovaná data pro zpracování procedurou webové služby.

```

// Request to start a new session
{
  "Token": "",
  "PublicKey": "7FqaOjgyg8NkE0CO",
  "Checksum": "hEsgbYZszQvbLMAtrjQCYFhJLCpocCoPNvxNbedSj01vZpUeVS1JbWcewNADdCZ1",
  "Data": "iaeV/O1rv0fCVZM1CbIk+t8mccb+9pTZ+iQQurn9Vr/uEpA08bkuSsziT0rS7q0a5Euote1
}
// Result: Token
{
  "Token": "L66uLYbo2EhFL7jh+4ErSJO2uadRRnbB"
}

```

Obrázek 5.3: Ukázka JSON obálky požadavku a odpovědi webové služby

5.6.1.2 Zpracování požadavku

Požadavek ve formě JSON obálky je přijat webovou službou a proběhne jeho zpracování.

Zpracování JSON obálky probíhá následujícím postupem:

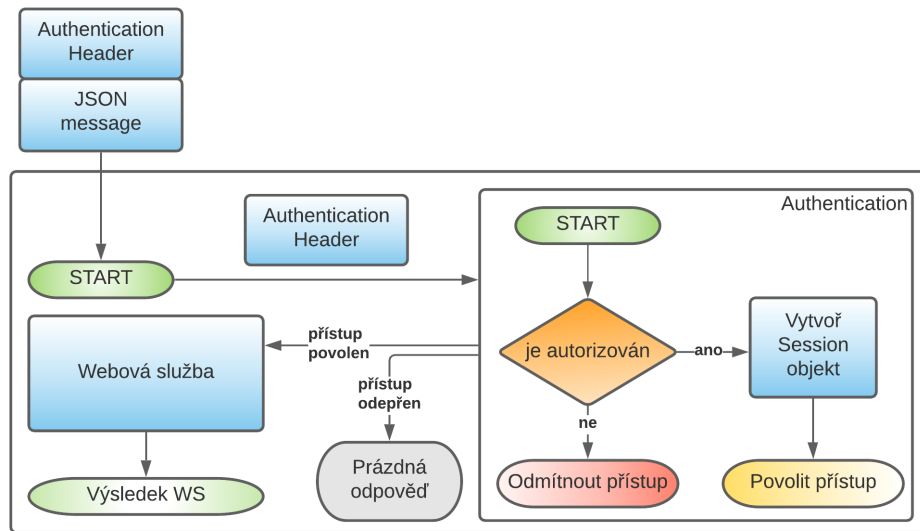
1. Provede se konverze (deserializace) JSON do příslušné třídy. V případě nezdaru je zpracování ukončeno a je vrácena chyba.
2. Kontrola, zda zaslaný veřejný klíč (PublicKey) je shodný s definovaným veřejným klíčem. V případě nezdaru je zpracování ukončeno a je vrácena chyba.
3. Pomocí definovaného veřejného klíče se dešifruje kontrolní součet (Checksum), výsledek se porovná s definovaným kontrolním řetězcem. Tímto se ověří, zda šifrování bylo provedeno správnou komponentou. V případě rozdílu je zpracování ukončeno a je vrácena chyba.
4. Provede se validita tokenu, tj. zda token existuje, zda je platný a zda je aktivní session spojená s tokenem. Kontrola se neprovádí v případě požadavku na připojení. Není-li token validní, je zpracování ukončeno a je vrácena chyba.
5. Pomocí definovaného veřejného klíče se provede dešifrování dat a provede se jejich zpracování podle povahy procedury webové služby.

5.6.1.3 Navázání spojení

Před zahájením odesílání požadavků na data je nutno nejprve navázat s webovou službou spojení. V rámci tohoto webová služba ověří dostupnost databáze, založí nové spojení (session) a vygeneruje token, pomocí kterého se následně identifikují další požadavky na data. Webová služba tím „pozná“, ze které databáze má data poskytovat. Schéma práce webové služby je na obrázku 5.4.

Proces zahájení spojení (session) probíhá následujícím způsobem:

1. Vytvoří se JSON obálka požadavku, hodnota pro Token je prázdná a hodnota pro Data je validní definice připojení k databázovému serveru.



Obrázek 5.4: Schéma práce webové služby

2. JSON obálka se odešle metodou POST na příslušné API webové služby.
3. Webová služba zpracuje přijatou JSON obálku. V případě nezdaru je zpracování ukončeno a je vrácena chyba.
4. Provede se zkušební připojení k databázovému serveru dle údajů předaných v hodnotě Data. V případě nezdaru je zpracování ukončeno a je vrácena chyba.
5. Vygeneruje se nový token, platnost tokenu je 24 hodin.
6. Pro vygenerovaný token se vytvoří nové spojení (session). Spojení (session) obsahuje údaj pro připojení k databázovému serveru.
7. Vytvořené spojení (session) je uloženo pro následné použití.
8. Webová služba vrátí JSON odpověď s vygenerovaným tokenem.

5.6.1.4 Požadavek vykonání databázového příkazu

Po navázání spojení (session) může klient pomocí tokenu začít zasílat požadavky na databázové operace, které se mají nad databází vykonat.

Proces odeslání požadavku na vykonání databázového příkazu databázovým serverem:

1. Vytvoří se JSON obálka požadavku, do hodnoty Token je uložen token vrácený webovou službou při požadavku na nové spojení (session). Do hodnoty Data je pomocí serializace do JSON uložen databázový příkaz včetně případných parametrů.

2. JSON obálka se odešle metodou POST na příslušné API webové služby.
3. Webová služba zpracuje přijatou JSON obálku. V případě nezdaru je zpracování ukončeno a je vrácena chyba.
4. Pomocí předaného tokenu webová služba dohledá příslušnou definici spojení (session) a ověří její platnost (24 hodin). V případě nezdaru je zpracování ukončeno a je vrácena chyba.
5. Z definice spojení (session) se použije definice připojení k databázovému serveru a vykoná se požadovaný databázový příkaz. V případě nezdaru je zpracování ukončeno a je vrácena chyba.
6. Webová služba vrátí JSON hodnoty získané vykonáním databázového příkazu.

5.6.2 API rozhraní webové služby

Přístup do jednotlivých procedur webové služby je pomocí parametru webové adresy (URL). Požadavek se odesílá metodou POST ve formě JSON obálky. Každá procedura vrací JSON paket s příslušnými údaji anebo s chybou a jejím popisem. Kompletní výpis JSON komunikace je uveden v příloze C.4.

Popis API rozhraní webové služby:

/MSSQL/GetToken – získání tokenu. V JSON obálce v části Data je předáno připojení k MSSQL databázi. Procedura zkontroluje validitu připojení k MSSQL databázi. V případě úspěšného připojení je vytvořena nová session a vygenerován token pro přístup k této session. Token je vrácen formou JSON paketu. V případě neúspěchu je vrácen JSON paket s chybou a jejím popisem.

/MSSQL/TokenIsValid – kontrola, zda token je validní. Procedura zkontroluje, zda předaný token je validní, jestli existuje aktivní session pro daný token. V případě vypršení platnosti tokenu je příslušná session uzavřena. V případě validity token je tento vrácen formou JSON paketu, v opačném případě je vrácen JSON paket s chybou a jejím popisem.

/MSSQL/ExecuteNonQuery – vykonání T-SQL příkazu, nevrací žádná data. V JSON obálce v části Data je ve formátu JSON předán T-SQL příkaz včetně případných parametrů. V případě úspěšného provedení je ve formě JSON paketu vrácen počet změněných záznamů, v případě chyby JSON paket s chybou a jejím popisem.

/MSSQL/ExecuteScalar – vykonání T-SQL příkazu, vrací první sloupec prvního záznamu. V JSON obálce v části Data je ve formátu JSON předán T-SQL příkaz včetně případných parametrů. V případě úspěšného provedení je ve formě JSON paketu vrácen první sloupec prvního záznamu, v případě chyby JSON paket s chybou a jejím popisem.

`/MSSQL/ExecuteReader` – vykonání T–SQL příkazu, vrací výslednou sadu záznamů. V JSON obálce v části Data je ve formátu JSON předán T–SQL příkaz včetně případných parametrů. V případě úspěšného provedení je ve formě datového JSON paketu vrácena výsledná sada záznamů, v případě chyby JSON paket s chybou a jejím popisem.

5.7 Číselníky

V *IS ProŘíz* se používá vícero jednodušších číselníků (např. Status zákazníka, Priorita projektu), které mají shodnou základní strukturu (Název, příznak Aktivní, Poznámka a Kód). Další číselníky (např. Obchodní struktura, Produkty, Sekce) mají základní strukturu rozšířenu jen o další atribut (např. do které sekce patří produkt).

Uložení číselníků v databázi lze řešit dvěma základními způsoby – pro každý číselník definovat v databázi vlastní tabulku anebo všechny číselníky uložit do jedné společné tabulky. Výhodou uložení číselníků v jedné společné tabulce je podstatné zjednodušení návrhu datové struktury databáze.

S ohledem na možnost *IS ProŘíz* používat různé datové zdroje, a u každého z nich udržovat i datovou strukturu, jsem pro uložení číselníků použil řešení ukládání do jedné společné tabulky *GeneralList* s využitím mapování dědičnosti Single Table Inheritance (viz kapitola Použité návrhové vzory č. 5.2). Stejná tabulka je použita i pro ukládání hodnot pro různá nastavení systému. Rozlišení třídy v záznamu je prostřednictvím atributu ClassID. Seznam rozlišení tříd je uveden v tabulce C.3 v příloze na straně 88.

5.8 Eskalace

Nezbytnou součástí *IS ProŘíz* je upozorňování na termíny úkolů a projektů s eskalací upozorňování na blížící se termín splnění úkolu a projektu a po vypršení termínu splnění. Pro tento účel jsem vytvořil systém eskalační politiky, která zabezpečuje odesílání informační zprávy na e–mail vlastníka projektu a úkolu. Zpráva obsahuje v obsahu zprávy informace o projektu či úkolu. Text zprávy je generován podle zadané šablony, lze tedy uživatelsky nastavit potřebný text. Předmět zprávy se mění podle stupně eskalace.

V rámci eskalační politiky se dále nastaví počet dnů před termínem úkolu a projektu, kdy se stupeň Informace změní na stupeň Termín. Pokud se nemá pro příslušný stupeň eskalace zpráva odesílat, obsluha nezadá předmět zprávy pro příslušný stupeň.

Stupně eskalace jsou:

Přiřazeno – Přiřazení anebo změna vlastníka projektu a úkolu.

Informace – Informační zpráva. Projekt a úkol je do termínu, přesněji zadaný počet dnů před termínem. Pro stupeň se definuje počet hodin pro opakování odeslání.

Termín – Zpráva o blížícím se konce termínu. Projekt a úkol je po zadaném počtu dnů před termínem, ale před vypršením termínu. Pro stupeň se definuje počet hodin pro opakování odeslání.

Po termínu – Zpráva, že projekt a úkole je po termínu. Pro stupeň se definuje počet hodin pro opakování odeslání.

Do eskalace se zahrnou pouze projekty a úkoly, které mají zadaný termín splnění, mají aktivní status a přiřazený uživatel je aktivní a má definován e-mailovou adresu. Systém eskalace uloží informaci o odeslané zprávě (adresát, stupeň eskalace, datum a čas odeslání), tyto informace se použijí pro znovu odeslání eskalační zprávy po uplynutí zadané doby pro opakování anebo pro zjištění, zda nedošlo ke změně vlastníka projektu a úkolu.

5.8.1 Konzolová aplikace pro spuštění eskalačního procesu

V rámci eskalace je potřeba zajistit pravidelné spuštění procesu kontroly a odesílání eskalačních zpráv. Jako nejjednodušší řešení se mi jeví vytvoření jednoduché konzolové aplikace, která kontrolu a odeslání zabezpečí. Její pravidelné spuštění se zajistí pomocí naplánované úlohy systému Windows.

Databáze, která se má pro kontrolu eskalace použít, se konzolové aplikaci předá pomocí ID databáze v konfiguračním souboru Settings.xml (viz kapitola 5.4). Bez parametru (anebo parametr 0) se použije defaultní definice databáze z konfiguračního souboru.

5.9 Tiskové výstupy

Další z nezbytných částí *IS ProŘíz* jsou tiskové výstupy, konkrétně tisk vystavených faktur. Uživatel si zvolí, které faktury vytisknout a tyto se jí zobrazí v tiskovém náhledu s možností výběru tiskárny, na kterou se faktury vytisknou.

Pro tvorbu náhledů tiskových sestav se v prostředí .NET Framework používá třída `PrintPreviewDialog`. Je praktická a má snadné používání, ale je poměrně omezena a vypadá zastarale (od verze .NET 1.0 je prakticky beze změny). Při hledání lepšího řešení jsem na stránkách www.grapacity.com našel ukázkou kódu rozšíření této třídy. V ukázce kódu je implementace nahrazení třídy `PrintPreviewDialog` vytvořenou třídou `CoolPrintPreview` [9] s následujícími vylepšeními:

- Průběžné zobrazování stránek při generování dokumentu (původní třída zobrazí až celý vygenerovaný obsah).
- Zobrazuje aktuální číslo stránky a celkový počet stran (původní třída zobrazí jen aktuální číslo stránky).
- Implementace zobrazuje tlačítko pro výběr tiskárny a nastavení vzhledu stránky.

- Podporuje tisk jen určeného rozsahu stran (původní třída umožňuje pouze tisk celého dokumentu).

Dalším benefitem je možnost použití třídy `CoolPrintPreviewDialog` zdarma (citace z webu: *If you have an application that uses the `PrintPreviewDialog` class, you can replace it with the `CoolPrintPreviewDialog` and instantly get all these features for free!* [9]).

Tímto byla vyřešena problematika zobrazení náhledu tiskových sestav a jejich tisku, ale uvedené třídy neřeší vlastní generování obsahu sestav.

Vlastní generování obsahu sestav se provádí pomocí mnou vytvořených tříd. Tyto třídy umožňují definici sekcí sestavy (záhlaví sestavy, záhlaví stránky, tělo sestavy, zápatí sestavy, zápatí stránky) a pro každou sekci definovat její obsah (text, obdélník, čára). Výška sekce je přizpůsobena obsahu, tj. podle výšky textu se určí výška sekce. Generátor umožňuje definici více sestav do jednoho tiskového dokumentu. Toto se využívá při tisku více faktur, kdy každá faktura je samostatnou sestavou v jednom tiskovém dokumentu.

Celé řešení tiskového výstupu je v samostatné dll komponentě. Výhodou tohoto řešení je oddělení tisku od vlastní aplikace. V případě dalších rozšíření a úprav, např. implementace výstupu do PDF, není potřeba znovu kompilovat a instalovat celou aplikaci, ale jen tuto komponentu.

5.10 Webová aplikace

Dalším prvkem *IS ProŘíz* je možnost stahování nových verzí produktů zákazníky a automatická tvorba change logu (evidence verzí). Zaslání nových verzí se ve stávajícím stavu řešilo prostřednictvím e-mailových zpráv o jednotlivých verzích, informace se kopírovaly a docházelo k chybám typu chybný odkaz na aktualizací soubor. Tvorba change logu a jeho zaslání zákazníkům byla také problematická, opět se kopírovaly ty stejné informace, docházelo k chybovosti.

Tento manuální způsob je nahrazen webovou aplikací.

Zákazníkovi je zřízen v *IS ProŘíz* zřízen uživatelský účet, který je pomocí definice obchodní struktury propojen s daným zákazníkem. Při vydání nové verze produktu zákazník obdrží pouze informaci o nové verzi, všechny informace, které se doposud kopírovaly, najde ve webové aplikaci rozdělené do příslušných sekcí. Tímto pro uživatele odpadá duplicitní kopírování informací, sníží se chybovost a zákazník má všechny potřebné informace o svých produktech na jednom místě.

Do webové aplikace se zákazník přihlásí svým přihlašovacím jménem a heslem. Jsou mu zobrazeny jeho sekce a v každé sekci informace o aktuální verzi produktu. Sekce je zobrazena na obrázku 5.5. Další funkcionalitou webové aplikace je možnost zobrazení kompletní historie všech vydaných verzí produktu s datem vydáním, číslem verze, popisem a aktualizacími soubory. Change log je vyobrazen na obrázku 5.6.

[🏠](#)
[CML](#)
[Mamlas](#)
[Odhlášení](#)

CML

CRM systém Centrální Mozek Lidstva

CML	
Verze	20.0.3
Datum	12.03.2020
Popis	oprava chyby při přihlášení uživatele, pokud uživatel neměl definováno heslo
Aktualizace CML	<input type="button" value="CML_20200312_20.0.3.exe"/> 475 KB
Instalátor	<input type="button" value="CMLSetup.exe"/> 475 KB

Jste přihlášen/a jako **Hans Rodšild**.

Obrázek 5.5: Sekce webové aplikace

[🏠](#)
[CML](#)
[Mamlas](#)
[Odhlášení](#)

CML

Historie změn

Verze 20.0.3 z 12.03.2020	
oprava chyby při přihlášení uživatele, pokud uživatel neměl definováno heslo	
Aktualizace CML	CML_20200312_20.0.3.exe
Instalátor	CMLSetup.exe

Verze 20.0.2 z 12.02.2020	
Druhý release - oprava chyb RTM verze	
Aktualizace CML	CML_20200212_20.0.2.exe
Instalátor	CMLSetup.exe

Verze 20.0.1 z 15.01.2020	
Prvotní verze	
Aktualizace CML	CML_20200115_20.0.1.exe
Instalátor	CMLSetup.exe

Jste přihlášen/a jako **Hans Rodšild**.

Obrázek 5.6: Change log produktu ve webové aplikaci

Kapitola 6

Testování a souhrn dosažených výsledků

V této části práce je popsáno testování *IS ProŘíz* a proveden souhrn dosažených výsledků.

Při tvorbě informačního systému hraje významnou roli testování. Je potřeba ověřit veškeré funkce, správné chování na různé vstupy, očekávané i neočekávané, od uživatele, vyplnění povinných hodnot a celkovou funkčnost systému.

Prvotní testování probíhalo při vývoji, kdy byla při implementaci jednotlivých funkcí testována odezva systému na různé vstupy a propojení jednotlivých funkcí. Dalšími stupni bylo jednotkové testování, systémové testování a akceptační uživatelské testování.

6.1 Unit test

Unit tests [10] (jednotkové testy) slouží k automatizovanému testování části kódu, kdy se kontrolují výstupy na základě konkrétních vstupů. Testy by měly ověřovat pouze jedinou vlastnost či metodu komponenty, proto se nazývají jednotkovými. Testy musí být izolované, nesmí záležet na pořadí, v jakém jsou spouštěny. Pro tvorbu unit testů existuje mnoho frameworků, například xUnit či NUnit, případně existuje přímá podpora ve Visual Studiu (framework MSTest).

Pro testování jsem použil přímou podporu ve Visual Studiu. Bylo potřeba vytvořit třídu reprezentující sadu testů, tato třída musí mít atribut `[TestClass]`. Testovací metody reprezentující jednotlivé testy musí mít atribut `[TestMethod]`. V rámci metod se nad konkrétními vstupy spouští testovaný kód a kontrolují se výstupy.

Obsahem testu je test komunikace s metodami třídy webové služby pro šifrovanou komunikaci zprostředkovávající vzdálený přístup do MSSQL databáze. Jednotlivé části unit testu, jejich popis, vstup a očekávaný výstup jsou shrnuty v následujícím přehledu:

- **DefaultPublicKeyMatches** – Test zda veřejný klíč odpovídá klíči definovanému ve třídě
 - **Vstup** – Veřejný klíč
 - **Očekávaný výstup** – Klíč vrácený novou instancí třídy je shodný

- **UserDefinedPublicKeyMatches** – Test změny veřejného klíče za uživatelsky definovaný klíč
 - **Vstup** – Uživatelský klíč předaný nové instanci třídy
 - **Očekávaný výstup** – Instance třídy vrátí shodný uživatelský klíč
- **EmptyPublicKeyException** – Test, zda pokus o vytvoření nové instance s prázdným klíčem vrátí výjimku ArgumentException
 - **Vstup** – Prázdný klíč
 - **Očekávaný výstup** – Výjimka ArgumentException
- **StartNewSession** – Navázání komunikace a zahájení nové seance
 - **Vstup** – Veřejný klíč a definice připojení k MSSQL
 - **Očekávaný výstup** – Neprázdný token
- **CheckTokenIsValid** – Test, zda token je platný
 - **Vstup** – Vygenerovaný token
 - **Očekávaný výstup** – True; při false je vrácen popis chyby (např. token je neplatný)
- **ExecuteNonQuery** – Spustí T-SQL příkaz, který nevrací žádnou návratovou hodnotu
 - **Vstup** – Veřejný klíč, definice připojení k MSSQL a SQL příkaz
 - **Očekávaný výstup** – Vracený výsledek není popis chyby
- **ExecuteScalar** – Spustí T-SQL příkaz, který vrací hodnotu první sloupec prvního řádku vrácené sady
 - **Vstup** – Veřejný klíč, definice připojení k MSSQL a SQL příkaz
 - **Očekávaný výstup** – Vracený výsledek není popis chyby
- **ExecuteReader** – Spustí T-SQL příkaz, který vrací sadu hodnot (tabulku)
 - **Vstup** – Veřejný klíč, definice připojení k MSSQL a SQL příkaz
 - **Očekávaný výstup** – Vracený výsledek není popis chyby

6.2 Systémové testování

Po testech na úrovni jednotlivých metod komponent se přechází na *systémové testování* [10]. Během těchto testů je aplikace podle připravených scénářů ověřována jako funkční celek z pohledu zákazníka, simulují se různé kroky a možnosti, které v praxi mohou nastat, jedná se o poslední úroveň testů před předáním díla zákazníkovi.

Cílem testu je otestování přístupových a výkonných práv uživatele. Uživatel bude nastaven tak, aby měl přístup jen ke svým projektům, úkolům, viděl jen zákazníky své nebo ve své struktuře a mohl zakládat nové úkoly a projekty. Vše ostatní má zakázáno.

Je založena uživatelská role definující práva podle Scénáře a nový uživatel, který je zařazen to této role. Dále byl v číselníku Struktura definován přístup tohoto uživatele pro jednoho zákazníka.

6.2.1 Průběh testování

V následujících kapitolách je popsán průběh a výsledek jednotlivých částí testu.

6.2.1.1 Přístupová práva

Obsahem tohoto kroku je test, zda uživatel má přístup jen do určených položek. Po přihlášení uživatele je zobrazena nabídka funkcí dle nastavení práv uživatele, přesněji jeho uživatelské role. V rámci tohoto testu se ověřuje, zda uživatel má přístup jen to povolených do agend.

6.2.1.2 Zákazníci

Obsahem tohoto kroku je test, zda uživatel má přístup jen k povoleným zákazníkům, tj. jen ze své obchodní struktury anebo za které zodpovídá, ostatní zákazníci mu nejsou zobrazeni.

6.2.1.3 Úkoly

Obsahem tohoto kroku je test, zda uživatel má přístup jen ke svým úkolům, tj. k úkolům, které má řešit. Úkoly pro jiné uživatele mu nejsou zobrazeny.

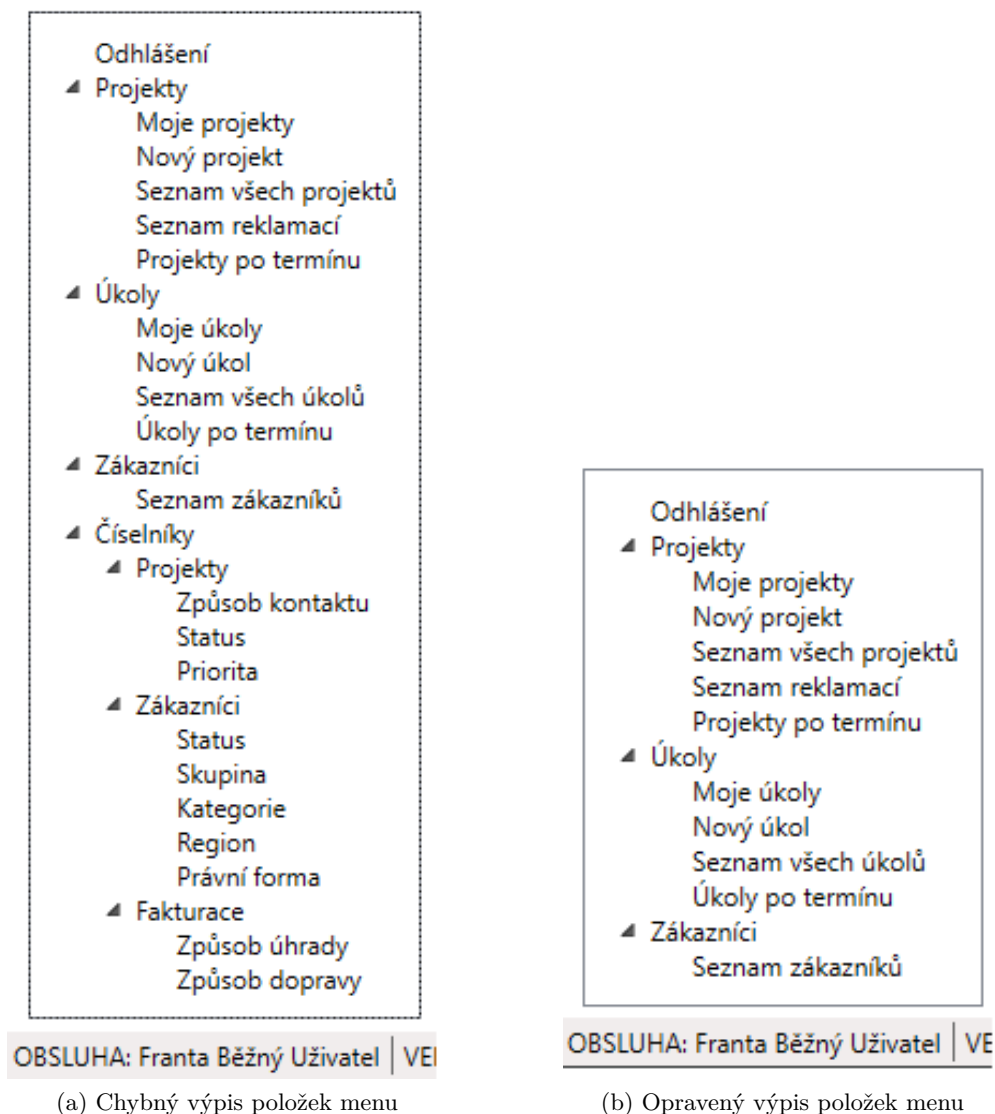
6.2.1.4 Projekty

Obsahem tohoto kroku je test, zda uživatel má přístup jen ke svým aktivním projektům, tj. projektům, u kterých je vlastníkem a které mají status s příznakem aktivní. Neaktivní projekty a projekty jiných uživatelů mu nejsou zobrazeny.

6.2.1.5 Závěr testu

Test v části přístupová práva uživatele selhal, byly zobrazeny i položky menu, ke kterým uživatel nemá mít přístup. Chyba je zobrazena na obrázku 6.1a. Po odstranění chyby byl test úspěšný i v části přístupových práv, výsledek je zobrazen na obrázku 6.1b.

V ostatních částech byl úspěšný.



Obrázek 6.1: Systémové testování – Přístupová práva

6.3 Akceptační uživatelské testování (UAT)

User acceptance test (UAT) jsou akceptační testy na straně uživatelů zadavatele. Probíhají u zadavatele v testovacím prostředí podle předem *připravených scénářů* [11]. Nalezené nesrovnalosti mezi aplikací a specifikací jsou hlášeny zpět vývojovému týmu, opravené chyby jsou nasazeny zpět do prostředí u zadavatele. Je velmi důležité definovat, jakým způsobem bude probíhat hlášení chyb od zadavatele a jakým způsobem bude zabezpečeno opravení nalezených chyb a to v co možná nejkratším čase. Zadavatel neočekává zcela bezchybný software, nicméně v případě mnoha chyb, zejména těch se zásadním dopadem na funkčnost celé aplikace, a dále trvá-li odstranění chyb dlouho, to může mít negativní dopad na úspěch celého projektu.

6.3.1 Scénář „Vytvoření nového zákazníka“

Předmětem scénáře je vytvoření nového zákazníka. Očekávaným výsledkem je založení nového zákazníka.

Kroky testu:

1. V nabídce voleb v levé části obrazovky vyberte položku Zákazníci a v rámci ní volbu nový zákazník.
2. Vyplňte položku Zkratka, dle této položky se bude zákazník vyhledávat, zvolte tedy vhodnou hodnotu.
3. Vyplňte identifikační údaje o společnosti (název, DIČ, IČ, Sídlo).
4. Vyplňte kontaktní informace, jsou-li známy.
5. Vyplňte Ostatní údaje, v případě potřeby změňte uživatele zodpovídající za zákazníka.
6. Určete omezení sekcí.
7. kliknutím na tlačítko Zavřít a po odsouhlasení uložení změn zavřete kartu zákazníka.

6.3.2 Scénář „Aktualizace zákazníka z databáze ARES“

Předmětem scénáře je aktualizace údajů zákazníka prostřednictvím ARES. Očekávaným výsledkem je aktualizace údajů zákazníka z databáze ARES.

Kroky testu:

1. V nabídce voleb v levé části obrazovky vyberte položku Zákazníci a v rámci ní volbu Seznam zákazníků.
2. Otevřete existující kartu Zákazníka.

3. Vyplňte IČ, pokud není vyplněno.
4. Vedle IČ klikněte na tlačítko ARES.
5. Zkontrolujte, zda se údaje na kartě zákazníka v části název a sídlo společnosti zaktualizovaly.

6.3.3 Scénář „Zákazník -- vytvoření adresy příjemce (poštovní adresa)“

Předmětem scénáře je přidání adresy k zákazníkovi a její nastavení jako poštovní adresa pro fakturaci. Očekávaným výsledkem je definování nové poštovní adresy.

Kroky testu:

1. V nabídce voleb v levé části obrazovky vyberte položku Zákazníci a v rámci ní volbu Seznam zákazníků.
2. Otevřete existující kartu Zákazníka.
3. V části Pobočky a provozovny klikněte do prázdného pole Název.
4. Zadejte Název adresy (doporučeno Poštovní adresa).
5. Zadejte další položky adresy (název, adresa).
6. Ukončení zadání adresy provedte klávesou Enter.
7. V části Fakturace v seznamu Příjemce faktury (poštovní adresa) vyberte zadanou adresu.
8. Kliknutím na tlačítko Zavřít a po odsouhlasení uložení změn zavřete kartu zákazníka.

6.3.4 Závěr testu

Vykonáním všech kroků jednotlivých scénářů bylo uživatelem odzkoušeno chování systému a správnost postupů. Všechny scénáře skončily s očekávaným výsledkem.

6.4 Výkonnostní testy

Součástí testů jsou i výkonnostní (performance) testy, kdy je testován výkon a odezva aplikace při velkém objemu dat. Při tomto testu jsou pro agendy aplikace, kde lze očekávat velké množství záznamů (stovky tisíc apod.), vygenerovány záznamy v předpokládaném počtu. Nad těmito daty jsou pak prováděny výkonnostní testy, zkouší se odezva aplikace a hledají se úzká místa. Nalezená úzká místa jsou vhodným a příslušným způsobem odstraněna (např. přidáním vhodného indexu do datové struktury).

S ohledem na předpokládaný počet záznamů v řádu maximálně tisíců nebyly výkonnostní testy nad velkým objemem dat provedeny.

6.5 Souhrn výsledků

Všechny testy popisované v této kapitole č. 6 byly provedeny dle jednotlivých kroků či scénářů. V případě nalezení chyby byly tyto odstraněny a opravy opět otestovány. Finální závěr testování je, že všechny testy byly úspěšné a skončily dle očekávání.

Kapitola 7

Závěr

Cílem této práce bylo vytvoření informačního systému pro řízení projektů. V rámci práce jsem se seznámil s problematikou řízení projektů, provedl analýzu požadavků na informační systém, vytvořil jsem návrh systému a následně provedl jeho implementaci. Dalšími kroky bylo provedení testů a vytvoření projektové dokumentace.

Seznam požadavků vznikl na základě konzultací a analýzou potřeb zadavatele a po konzultacích s vedoucím projektu.

7.1 Shrnutí výsledků práce

V práci jsem popsal tvorbu informačního systému včetně teoretických částí vysvětlujících problematiku řízení projektů.

Hlavního cíle práce – vytvoření informačního systému pro řízení projektů, bylo dosaženo. Systém dokáže evidovat zákazníky s možností aktualizace jejich údajů pomocí databáze ARES. Dále umožňuje evidovat produkty a projekty. Systém spravuje přijaté požadavky na projektech, požadavky lze rozdělovat na etapy, kroky a úkony do stromové struktury a definovat u každé časovou náročnost. Pro každý projekt, etapu, krok a úkon lze evidovat rozsah provedených prací. Provedené práce lze vyúčtovávat fakturami, systém obsahuje evidenci faktur a umožňuje jejich tisk. Systém umožňuje definovat obchodní strukturu (jaké produkty má zákazník, ke kterým zákazníkům a produktům má uživatel přístup) a umožňuje rozsáhlou definici uživatelských práv. Systém dále umožňuje vést informace o vydaných verzích programů (produktů) s možností stahování nových verzí zákazníkem prostřednictvím webové aplikace. Systém také umí upozorňovat na úkoly a termíny. Systém v rovněž umožňuje využívat pro svůj chod různé databáze a to včetně možnosti přístupu k databázím přes Internet prostřednictvím webové služby.

Díky této práci jsem se seznámil s technologiemi, které jsem znal jen okrajově. Zejména se jednalo o .NET Core a architekturu MVC, knihovnu tříd WPF (Windows Presentation Foundation)

a formát JSON. Při tvorbě práce jsem využil poznatky získané během studia. Zejména se jednalo o použití doménového modelu a data mapperů a .NET Core a architektury MVC.

Ačkoliv jsem se při realizaci mohl opřít o mnohaletou praxi vývojáře SW, při realizaci jsem narazil na problémy, které postup prací zpomalovaly. Zejména se jednalo o seznámení se s prostředím WPF jako takovým a dále návrh a implementace doménového modelu a mapperů, které jsem několikrát předělával téměř od základu. Oproti tomu nejzajímavější byl vývoj šifrovaného přístupu k databázi přes webovou službu s využitím JSON paketů a dále implementace eskalační politiky.

Zadání práce bylo splněno ve všech bodech zadání. *IS ProŘíz* je v současné době postupně nasazován ve výše zmíněné společnosti.

7.2 Možností dalšího budoucího rozvoje *IS ProŘíz*

Vývoj *IS ProŘíz* rozhodně není ukončen, práce na informačním systému budou dále pokračovat a bude se rozšiřovat a zdokonalovat funkcionalita aplikací.

Přehled některých možných rozšíření:

- Možnost tiskového výstupu do PDF
- Pro adresy a položky faktur přepracovat na návrhový vzor Lazy Load, tj. objekt by neobsahoval všechna data, data by se načítala až v případě potřeby (odložené načtení)
- Příkládání dokumentů k zákazníkům a projektům
- Fulltextové vyhledávání
- Další tiskové výstupy (např. statistické přehledy projektů)
- Tvůrce sestav (uživatelsky definovatelné sestavy, zejména faktury)
- Přidání analytických grafů
- Možnost notifikace zákazníka o přijetí požadavku a postupu prací
- Možnost notifikace zákazníka o nové verzi produktu
- Možnost slevy a ceníku položek ve fakturaci
- Redesign webové aplikace

Literatura

1. ZDENKO, RNDr. Staníček. *Řízení projektů* [online]. 2002 [cit. 2021-04-12]. Dostupné z: <https://www.systemonline.cz/clanky/rizeni-projektu.htm>.
2. ZDENKO, RNDr. Staníček. *Řízení projektů II. díl* [online]. 2003 [cit. 2021-04-12]. Dostupné z: <https://www.systemonline.cz/clanky/rizeni-projektu-ii-dil.htm>.
3. *Projektově.cz: Řízení projektů online* [online]. 2020 [cit. 2021-04-12]. Dostupné z: <https://www.projektove.cz>.
4. *321 INTRA: Firemní intranet* [online] [cit. 2021-04-12]. Dostupné z: <https://www.321intra.cz>.
5. *Webová služba* [online]. San Francisco (CA): Wikimedia Foundation, 2021 [cit. 2021-03-24]. Dostupné z: https://cs.wikipedia.org/wiki/Webov%C3%A1_slu%C5%BEba.
6. *ARES: Administrativní registr ekonomických subjektů* [online]. Praha: Ministerstvo financí ČR, 2013 [cit. 2021-03-24]. Dostupné z: <https://www.info.mfcr.cz/ares/ares.html.cz>.
7. PETROVAJ, Martin. *Úvod do MVC architektury v ASP.NET Core* [online]. 2018 [cit. 2021-04-17]. Dostupné z: <https://www.itnetwork.cz/csharp/asp-net-core/zaklady/uvod-do-mvc-architektury-v-aspnet-core>.
8. FOWLER, Martin. *Patterns of enterprise application architecture*. Boston: Addison-Wesley, 2003. ISBN 03-211-2742-0.
9. GRAPECITY. *CoolPrintPreview, GrapeCity Code Samples* [online]. 2019 [cit. 2021-04-12]. Dostupné z: <https://www.grapecity.com/samples/coolprintpreview>.
10. HLAVA, Tomáš. *Fáze a úroveň provádění testů* [online] [cit. 2021-03-24]. Dostupné z: <http://testovanisoftwaru.cz/tag/akceptacni-testovani>.
11. ŠTRBÁK, Martin. *Jak psát srozumitelné testovací případy pro akceptační testy* [online] [cit. 2021-03-24]. Dostupné z: <http://cz-testing.blogspot.com/2011/04/jak-psat-srozumitelne-testovaci-pripady.html>.

Příloha A

Use case scénáře

V této příloze jsou uvedeny procesních diagramy popisující přijetí a zpracování projektu.

A.1 UC Nový projekt

Use case scénář založení nového projektu, diagram aktivit je zobrazen na obrázku A.1.

UC Nový projekt

1. Uživatel vybere v nabídce volbu „Nový projekt“
2. Systém otevře formulář pro vyplnění údajů o projektu
3. Systém vyplní datum a čas kontaktu
4. Systém vyplní status
5. Systém vyplní prioritu
6. Uživatel zadá název projektu
7. Uživatel vybere zákazníka
8. Uživatel vybere produkt ze seznamu produktů zákazníka
9. Uživatel vybere kontaktní osobu ze seznamu kontaktních osob zákazníka
10. Uživatel zadá způsob kontaktu
11. Uživatel upraví datum a čas kontaktu, liší-li se od nastaveného systémem
12. Uživatel změní status, liší-li se od nastaveného systémem
13. Uživatel změní prioritu, liší-li se od nastaveného systémem

14. Uživatel zadá datum zahájení
15. Uživatel zadá datum dokončení
16. Systém zkontroluje, zda datum zahájení není později než termín dokončení
17. Systém zkontroluje, zda termín dokončení není dříve než datum zahájení
18. Uživatel zadá, zda projekt je veřejný (bude dostupný ve webové aplikaci)
19. Je-li projekt veřejný, systém vygeneruje číslo
20. Uživatel zadá popis
21. Uživatel zkontroluje, zda jsou zadané údaje správné
22. Systém zkontroluje vyplnění povinných údajů

UC končí

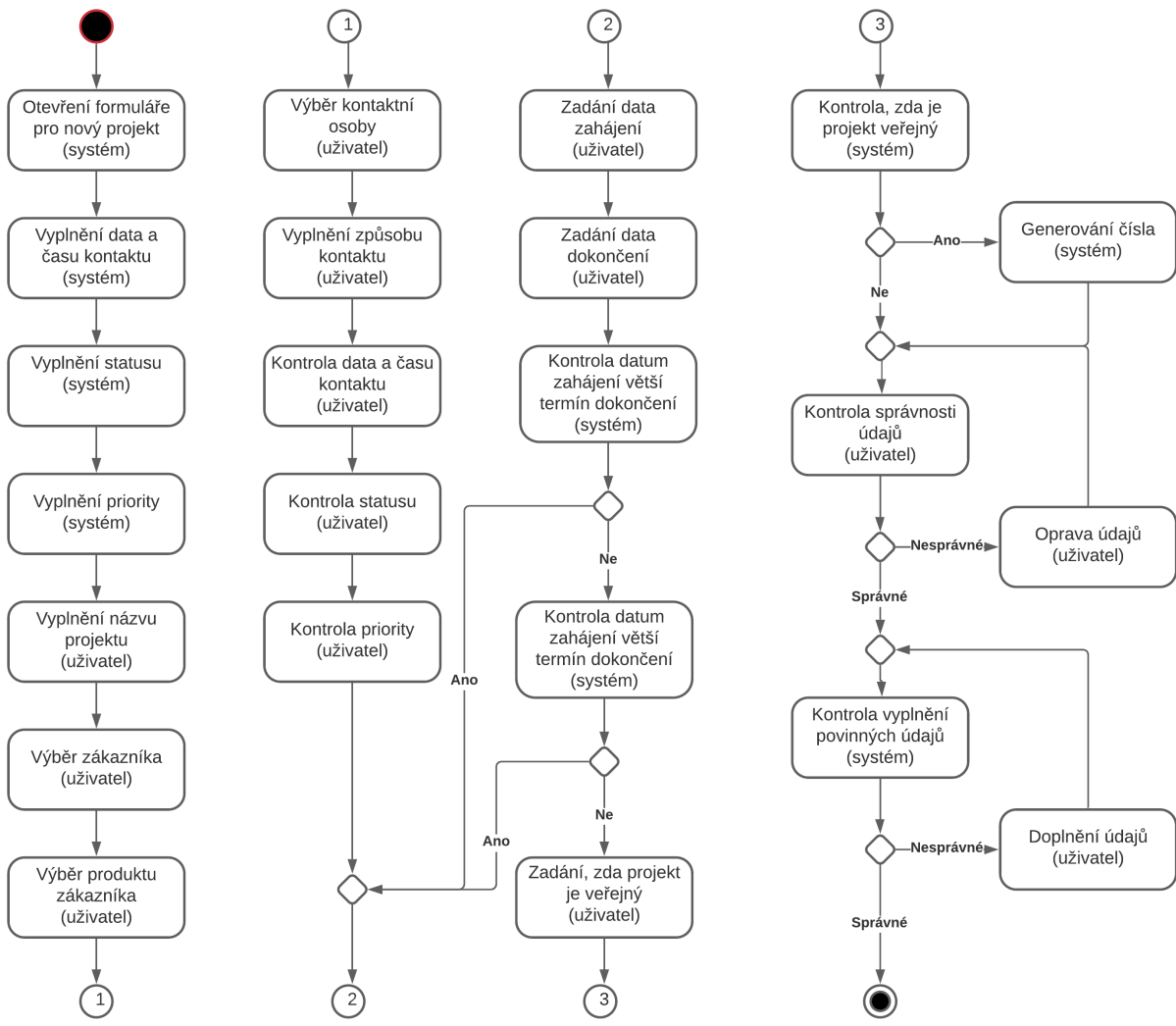
A.2 UC Vyúčtování prací projektu

Use case scénář vyúčtování prací projektu, diagram aktivit je zobrazen na obrázku A.2.

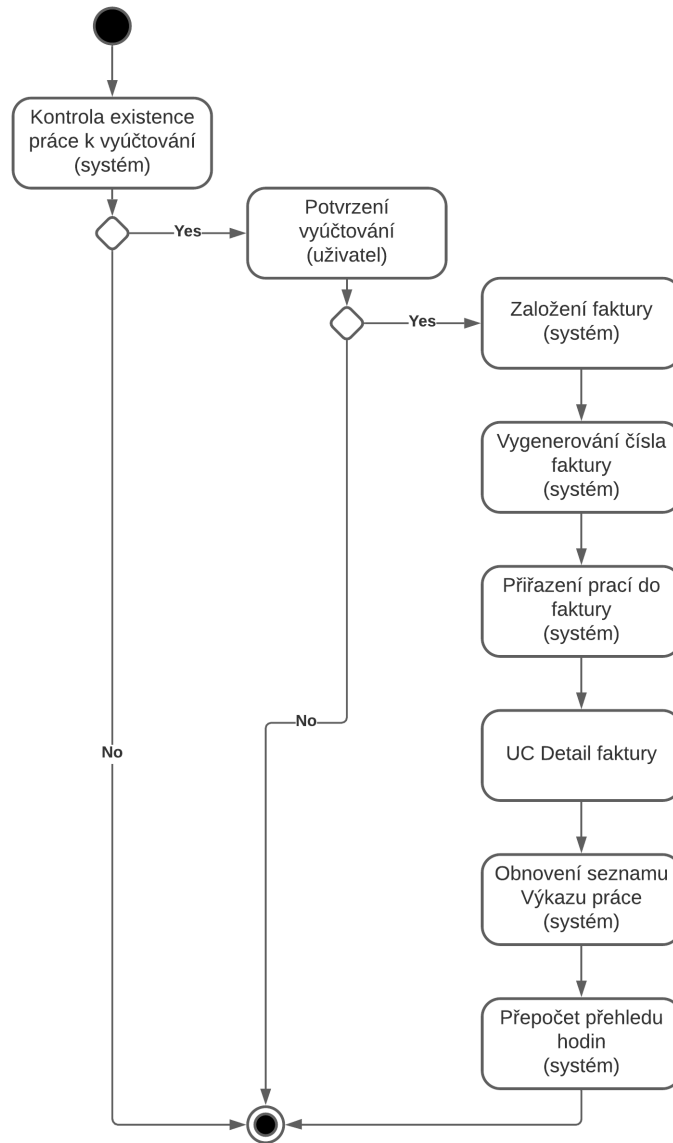
UC Vyúčtování prací projektu

1. Uživatel klikne na možnost „Vyúčtovat“
2. Systém ověří, zda je nějaká vykázaná práce k vyúčtování
3. Uživatel potvrdí, zda se má vyúčtování provést
4. Systém založí novou fakturu
5. Systém vygeneruje a přiřadí číslo faktury
6. Systém do faktury přiřadí nevyúčtované práce
7. UC Detail faktury
8. Systém obnoví seznam úkonů ve Výkazu práce
9. Systém přepočte přehled hodin projektu, podřízených projektů a celkem

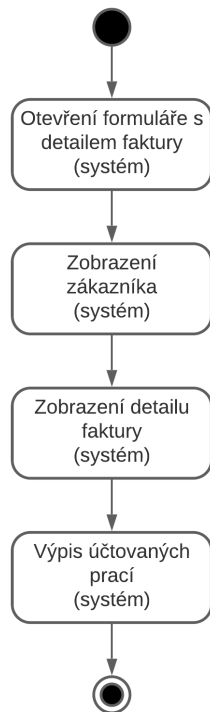
UC končí



Obrázek A.1: Diagram aktivit Nový projekt



Obrázek A.2: Diagram aktivit Vyúčtování prací projektu



Obrázek A.3: Diagram aktivit Detail faktury

A.3 UC Detail faktury

Use case scénář detail faktury, diagram aktivit je zobrazen na obrázku A.3.

UC Detail faktury

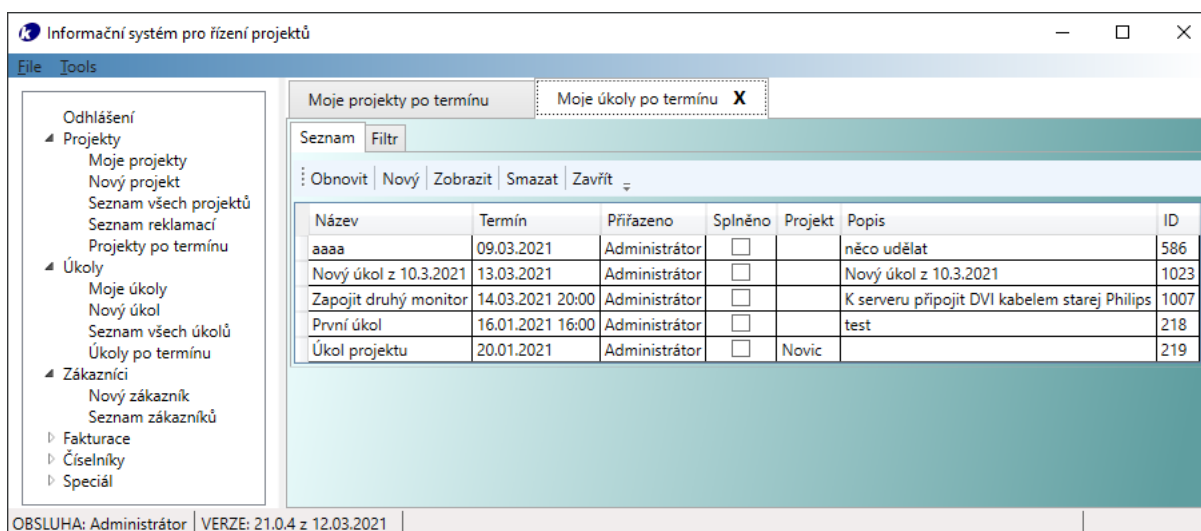
1. Systém otevře formulář s detailem faktury
2. Systém vypíše zákazníka
3. Systém vypíše detail faktury
4. Systém vypíše všechny práce účtované fakturou

UC končí

Příloha B

Projektová dokumentace

IS je vytvořen jako desktopová aplikace pro operační systém Windows 7 a vyšší. Základní prostředí je zobrazeno na obrázku B.1. Rozsah nabídky voleb v levé části základního prostředí závisí na právech uživatele, respektive podle nastavení jeho uživatelských rolí. Jednotlivé volby jsou otevírány v samostatných záložkách v pravé části základního prostředí aplikace. Volby lze rozdělit do dvou skupin: jednoduchý seznam a rozšířený seznam. Jednoduchý seznam je určen pro přidání, přímou editaci a mazání položek, má-li na uživatel příslušná práva. V rozšířeném seznamu probíhá přidání a editace položky prostřednictvím detailu položky. Některé rozšířené seznamy (například Zákazníci, viz kapitola B.4) obsahují i možnost filtru položek podle nadefinovaných parametrů. Každý seznam obsahuje v záhlaví nástrojovou lištu určenou k práci se seznamem, lišta obsahuje příkazy, které lze nad seznamem provádět (např. Obnovit obnoví podkladová data seznamu, Nový přidá novou položku), zobrazené příkazy závisí na povaze seznamu a právech uživatele.



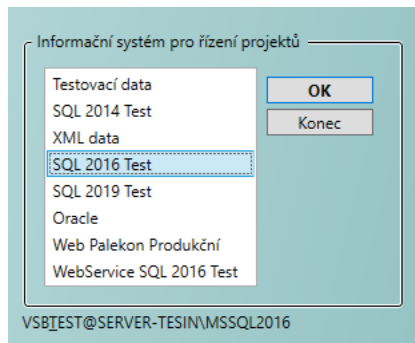
Obrázek B.1: Hlavní obrazovka aplikace

B.1 Spuštění

Aplikace se spustí příslušnou ikonou na ploše operačního systému Windows.

B.1.1 Výběr databáze

Je-li definováno více datových klientů, nabídne se po spuštění aplikace seznam definovaných databází, které aplikace použije pro práci. Výběr databáze je zobrazeno na obrázku B.2.

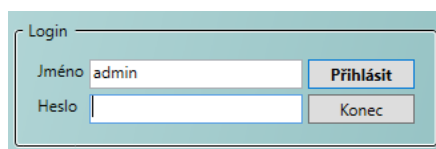


Obrázek B.2: Výběr databáze po spuštění aplikace

Seznam databází se načte ze souboru Settings.xml (implementace je popsána v kapitole 5.4), po výběru tlačítkem OK se podle typu databáze dohledá příslušný dll soubor, není-li přímo definován v xml, tento se pomocí sebereflexe zkontroluje, zda obsahuje požadované třídy a metody a provede se zkušební připojení k databázi. Pokud všechny kontroly proběhnou úspěšně, do lokálních parametrů se uloží informace, která databáze je použita a zobrazí se dialog pro přihlášení uživatele. Uložená informace o naposledy použité databázi se použije při příštím spuštění, aby se tato databáze přednastavila.

B.1.2 Přihlášení uživatele

Aplikace přednastaví předchozí úspěšně přihlášeného uživatele. Přihlášení uživatele je zobrazeno na obrázku B.3.



Obrázek B.3: Přihlášení uživatele

Uživatel zadá své přihlašovací heslo (je-li definováno, aplikace umožňuje i prázdné heslo) a pomocí tlačítka Přihlásit se přihlásí do systému. Po přihlášení jsou zobrazeny projekty a úkoly, které jsou po termínu a jsou přiřazeny přihlášenému uživateli. [viz obr Hlavní obrazovka aplikace]

B.2 Speciál

Pod volbou Speciál se nachází speciální funkce systému: nastavení uživatelů systému, správa rolí uživatelů a nastavení systému.

B.2.1 Uživatelé

Volbou se definují uživatelé systému a jejich přístupová a výkonná práva. Definice uživatele je zobrazena na obrázku B.4.

Nový uživatel X

Základní údaje

Login

Heslo

Aktivní

Titul před

Jméno

Příjmení

Titul za

Telefon

E-mail

Další kontakt

Další kontakt

Role uživatele

Administrátor

Nová role Leona

ReadOnly

Uživatel

Vlastní

Vlastní skupina

Vlastní struktura

Poznámka

Uložit Zavřít

Obrázek B.4: Definice uživatele

Login a Heslo jsou přihlašovací údaje uživatele, login je povinný údaj. Příznakem Aktivní se určuje, zda se uživatel může přihlásit do aplikace nebo nikoliv (volbou tedy lze „zablokovat“ uživatele). Položky „Další kontakt“ lze použít pro zadání dalších kontaktů mimo zobrazené (např. Skype). Pomocí seznamu Role uživatele se definuje role uživatele v systému, lze vybrat i více rolí. V případě více rolí se uplatní vyšší úroveň (např. pokud mezi rolemi je jedna s příznakem Administrátor, je uživatel administrátor systému). V Nastavení (viz kapitola B.2.3) lze definovat, která role se přednastaví u nového uživatele. Přihlašovací údaje, příznak Aktivní a zařazení uživatele do role/do rolí může změnit jen uživatel s právem administrátor.

B.2.2 Role uživatelů

Volbou se definují role uživatelů, tj. přístupová a výkonná práva. Definice role je zobrazena na obrázku B.5.

Role uživatelů
Role Uživatel **X**

Základní údaje

Název

Administrátor

Popis

Běžný uživatel systému

Nastavení práv

Položka	Přístup	Přidat	Zobrazit	Editovat	Mazat
Zákazníci	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Vše	Vše	<input type="checkbox"/>
Projekty	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Vše	Vše	<input type="checkbox"/>
Úkoly	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Vše	Vše	<input type="checkbox"/>
Fakturace	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Vše	Žádné	<input type="checkbox"/>
Uživatelé	<input type="checkbox"/>	<input type="checkbox"/>	Vše	Žádné	<input type="checkbox"/>
Role uživatelů	<input type="checkbox"/>	<input type="checkbox"/>	Vše	Žádné	<input type="checkbox"/>
Aktuální uživatel	<input checked="" type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>
Nastavení prostředí	<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>
Struktura	<input type="checkbox"/>	<input type="checkbox"/>	Vše	Žádné	<input type="checkbox"/>
Sekce	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Vše	Žádné	<input type="checkbox"/>
Produkty	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Vše	Žádné	<input type="checkbox"/>
Verze produktů	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Vše	Vše	<input type="checkbox"/>
Číselníky/Státy	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Vše	Žádné	<input type="checkbox"/>
Číselníky/Způsob kontaktu projektu	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Vše	Vše	<input type="checkbox"/>
Číselníky/Status projektu	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Vše	Žádné	<input type="checkbox"/>
Číselníky/Priorita projektu	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Vše	Žádné	<input type="checkbox"/>
Číselníky/Status zákazníka	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Vše	Žádné	<input type="checkbox"/>
Číselníky/Skupina zákazníka	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Vše	Žádné	<input type="checkbox"/>
Číselníky/Kategorie zákazníka	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Vše	Žádné	<input type="checkbox"/>
Číselníky/Region zákazníka	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Vše	Žádné	<input type="checkbox"/>
Číselníky/Právní forma zákazníka	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Vše	Vše	<input type="checkbox"/>
Fakturace/Bankovní účty	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Vše	Žádné	<input type="checkbox"/>
Fakturace/Cizí měny	<input type="checkbox"/>	<input type="checkbox"/>	Vše	Žádné	<input type="checkbox"/>
Fakturace/Typy sazeb DPH	<input type="checkbox"/>	<input type="checkbox"/>	Vše	Žádné	<input type="checkbox"/>
Fakturace/Sazby DPH	<input type="checkbox"/>	<input type="checkbox"/>	Vše	Žádné	<input type="checkbox"/>
Fakturace/Způsob úhrady	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Vše	Žádné	<input type="checkbox"/>
Fakturace/Způsob dopravy	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Vše	Žádné	<input type="checkbox"/>
Fakturace/Číselné řady	<input type="checkbox"/>	<input type="checkbox"/>	Vše	Žádné	<input type="checkbox"/>

Obrázek B.5: Nastavení role uživatele

Pro každou položku lze definovat:

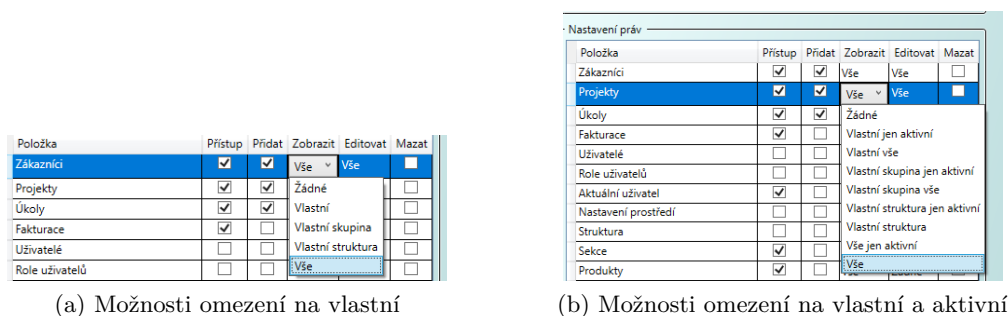
Přístup – zda k ní má uživatel přístup, tj. bude-li mu zobrazena v levém panelu s nabídkou funkcí.

Přidat – zda uživatel může pořizovat nové položky/záznamy.

Zobrazit – zda a případně jaké položky budou uživateli zobrazeny v základních seznamech (lze omezit např. jen na jeho vlastní projekty). Seznam typů omezení je zobrazeno na obrázcích Rozsah omezení položky uživatelské role B.6.

Editovat – zda případně jaké položky budou moci uživatel editovat (lze omezit např. jen na jeho vlastní). Seznam typů omezení je zobrazeno na obrázcích Rozsah omezení položky uživatelské role B.6.

Mazat – zda uživatel může záznam/y mazat.



Obrázek B.6: Rozsah omezení položky uživatelské role

B.2.3 Nastavení

Volbou se definuje nastavení prostředí a nastavení eskalace.

B.2.3.1 Nastavení prostředí

Obsahuje různá nastavení systému a prostředí aplikace. Nastavení aplikace je zobrazeno na obrázku B.7.

Význam položek a nastavení:

- přednastavený stát a status pro nové zákazníka
- přednastavená status a priorita pro nový projekt
- přednastavená role pro nové uživatele
- Fakturace:
 - Účetní jednotka – výběr účetní jednotky ze seznamu zákazníků
 - Tuzemská měna – označení tuzemské měny pro faktury (výchozí je Kč)

Nastavení aplikace X

Nastavení prostředí

Zákazník

Přednastavený stát: ČR

Přednastavený status: Nový

Projekt

Přednastavený status: Nový

Přednastavená priorita: Střední

Role uživatelů

Výchozí role: Uživatel

Fakturace

Účetní jednotka: PALEKON

Tuzemská měna: Kč

Hranice základní sazby DPH: 16,00

Přednastavená číselná řada faktur:

Přednastavený bankovní účet:

Přednastavený typ sazby DPH:

Ceníkové položky:

Zobrazit slevy:

Obchodní balení:

Obrázek B.7: Nastavení aplikace

- Hranice základní sazby DPH — pokud není sazba definována v číselníku sazeb DPH, tak nad zadanou hodnotu se považuje sazba DPH za základní
- Přednastavená číselná řada faktur -- výchozí řada pro vytvoření nové faktury (zejména z projektů)
- Přednastavený typ sazby DPH -- podle typu sazby DPH se na u položek faktur přednastaví příslušná sazba
- Ceníkové položky, Zobrazit slevy, Obchodní balení -- určeno pro budoucí rozšíření aplikace

B.2.3.2 Eskalace -- SMTP server

Nastavení SMTP serveru pro odchozí zprávy eskalace. Nastavení SMTP serveru je zobrazeno na obrázku B.8.

Eskalace - SMTP server

SMTP server: smtp.....

Port: 25

Uživatelské jméno:

Uživatelské heslo:

Adresa odesílatele: post@....

Jméno odesílatele:

Obrázek B.8: Nastavení SMTP serveru

Definuje se adresa SMTP serveru a port, případné přihlašovací jméno a heslo k SMTP serveru (lze nastavit i prázdné, pokud server nevyžaduje autentifikaci) a e-mailová adresa odesílatele zprávy (plus případně zobrazované jméno místo e-mailové adresy).

B.2.3.3 Eskalace — Projekty

Nastavení pravidel pro eskalaci projektů. Nastavení pravidel pro eskalaci projektů je zobrazeno na obrázku B.9.

Název	Eskalace projekty
Počet dnů před termínem	7
Text zprávy	Datum zahájení: <#StartDate#> Termín splnění: <#DeadLineDate#> Přiřazeno: <#OwnerFullName#> Zákazník: <#CustomerFullName#> (<#CustomerShortName#>) Předmět Přiřazeno: Přiřazen projekt Předmět Informace: Informace o přiřazeném projektu Hodiny opakování Informace: 48 Předmět Termín: Blíží se termín splnění projektu! Hodiny opakování Termín: 24 Předmět Po termínu: Projekt po termínu! Hodiny opakování Po termínu: 36
Aktivní	<input checked="" type="checkbox"/>

Obrázek B.9: Nastavení pravidel pro eskalaci projektů

Popis položek:

Název – název eskalace, používá se ve výpisu konsolové aplikace pro spuštění eskalačního procesu.

Počet dnů před termínem – počet dnů před termínem, kdy se mění stupeň eskalace Přiřazeno na Termín, eskalace.

Text zprávy – šablona textu zprávy, v textu lze pomocí speciálních značek přenést hodnoty některých položek projektu. Seznam značek je uveden v tabulce B.1. Ukázka definice je ve výpisu B.1.

Předmět Přiřazeno – předmět odesílané zprávy při zadání projektu vlastníkovvi nebo jeho změně; není-li vyplněno, nebude odeslána žádná zpráva.

Předmět Informace – předmět odesílané informativní zprávy pro stupeň eskalace Informace; není-li vyplněno, nebude odeslána žádná zpráva.

Hodiny opakování Informace – počet hodin od předchozí zprávy pro stupeň eskalace Informace, kdy bude znovu odeslána informační zpráva; není-li vyplněna kladná nenulová hodnota, nebude odeslána žádná zpráva.

Tabulka B.1: Mapování polí šablony pro projekty

Značka	Význam
<#Name#>	Název projektu
<#OwnerEmail#>	E-mailová adresa vlastníka projektu
<#OwnerFullName#>	Jméno a příjmení vlastníka projektu
<#CustomerShortName#>	Zkratka zákazníka
<#CustomerFullName#>	Obchodní název zákazníka
<#ProductCode#>	Kód produktu
<#ProductName#>	Název produktu
<#PersonFullName#>	Kontaktní osoba
<#SourceName#>	Zdroj kontaktu
<#ReceiptDateTime#>	Datum a případný čas přijetí
<#StatusName#>	Status projektu
<#PriorityName#>	Priorita projektu
<#StartDate#>	Datum zahájení
<#DeadLineDate#>	Termín dokončení
<#EndDate#>	Datum ukončení
<#TicketNumber#>	Číslo projektu
<#PublicAssignment#>	Zadání projektu
<#PublicNotes#>	Veřejné poznámky projektu
<#Notes#>	Interní poznámky projektu

Předmět Termín – předmět odesílané informativní zprávy pro stupeň eskalace Termín (blíží se konec termínu); není-li vyplněno, nebude odeslána žádná zpráva.

Hodiny opakování Termín – počet hodin od předchozí zprávy pro stupeň eskalace Termín (blíží se konec termínu), kdy bude znovu odeslána informační zpráva; není-li vyplněna kladná nenulová hodnota, nebude odeslána žádná zpráva.

Předmět Po termínu – předmět odesílané informativní zprávy pro stupeň eskalace Po termínu (termín vypršel); není-li vyplněno, nebude odeslána žádná zpráva.

Hodiny opakování Po termínu – počet hodin od předchozí zprávy pro stupeň eskalace Po termínu (termín vypršel), kdy bude znovu odeslána informační zpráva; není-li vyplněna kladná nenulová hodnota, nebude odeslána žádná zpráva.

Projekt: <#Name#>

Číslo: <#TicketNumber#>

Termín splnění: <#DeadLineDate#>

Přiřazeno: <#OwnerFullName#>

Zákazník: <#CustomerFullName#> (<#CustomerShortName#>)

Produkt: <#ProductName#> (<#ProductCode#>)

Zadání projektu: <#PublicAssignment#>
 Zadavatel: <#PersonFullName#>
 Datum zadání: <#ReceiptDateTime#>
 Status: <#StatusName#>
 Priorita: <#PriorityName#>
 Datum zahájení: <#StartDate#>
 Poznámky k projektu: <#PublicNotes#>

Listing B.1: Ukázka definice šablony zprávy pro eskalaci projektů

B.2.3.4 Eskalace — Úkoly

Nastavení pravidel pro eskalaci úkolů. Nastavení pravidel pro eskalaci úkolů je zobrazeno na obrázku B.10.

Obrázek B.10: Nastavení pravidel pro eskalaci úkolů

Popis položek:

Název – název eskalace, používá se ve výpisu konsolové aplikace pro spuštění eskalačního procesu.

Počet dnů před termínem – počet dnů před termínem, kdy se mění stupeň eskalace Přiřazeno na Termín, eskalace.

Text zprávy – šablona textu zprávy, v textu lze pomocí speciálních značek přenést hodnoty některých položek úkolu. Seznam značek je uveden v tabulce B.2. Ukázka definice je ve výpisu B.2.

Předmět Přiřazeno – předmět odesílané zprávy při zadání projektu vlastníkovvi nebo jeho změně; není-li vyplněno, nebude odeslána žádná zpráva.

Tabulka B.2: Mapování polí šablony pro úkoly

<i>Značka</i>	<i>Význam</i>
<#Name#>	Název úkolu
<#DeadLineDate#>	Termín (vč. případného času
<#Notes#>	Popis úkolu
<#UserEmail#>	E-mailová adresa, komu byl úkol přiřazen
<#UserFullName#>	Jméno a příjmení, komu byl úkol přiřazen

Předmět Informace – předmět odesílané informativní zprávy pro stupeň eskalace Informace; není-li vyplněno, nebude odeslána žádná zpráva.

Hodiny opakování Informace – počet hodin od předchozí zprávy pro stupeň eskalace Informace, kdy bude znovu odeslána informační zpráva; není-li vyplněna kladná nenulová hodnota, nebude odeslána žádná zpráva.

Předmět Termín – předmět odesílané informativní zprávy pro stupeň eskalace Termín (blíží se konec termínu); není-li vyplněno, nebude odeslána žádná zpráva.

Hodiny opakování Termín – počet hodin od předchozí zprávy pro stupeň eskalace Termín (blíží se konec termínu), kdy bude znovu odeslána informační zpráva; není-li vyplněna kladná nenulová hodnota, nebude odeslána žádná zpráva.

Předmět Po termínu – předmět odesílané informativní zprávy pro stupeň eskalace Po termínu (termín vypršel); není-li vyplněno, nebude odeslána žádná zpráva.

Hodiny opakování Po termínu – počet hodin od předchozí zprávy pro stupeň eskalace Po termínu (termín vypršel), kdy bude znovu odeslána informační zpráva; není-li vyplněna kladná nenulová hodnota, nebude odeslána žádná zpráva.

```

Úkol: <#Name#>
Termín splnění: <#DeadLineDate#>
Přiřazeno: <#UserFullName#>
Popis úkolu:
<#Notes#>

```

Listing B.2: Ukázka definice šablony zprávy pro eskalaci úkolů

B.3 Číselníky

Volba obsahuje číselníky společné pro více agend.

- Seznam sekcí (více viz kapitola B.3.1)

- Struktura (více viz kapitola B.3.2)
- Seznam produktů (více viz kapitola B.3.3)
- Seznam států (více viz kapitola B.3.4)

B.3.1 Seznam sekcí

Číselník sekcí. Definice sekce je zobrazena na obrázku B.11.

Uložit Zavřít

Kód	Název	Aktivní	Popis	ID
Účetnictví	Účetnictví Mamlas	<input checked="" type="checkbox"/>	Modul Účetnictví informačního systému Mamlas	1043
Mzdy	Mzdy Mamlas	<input checked="" type="checkbox"/>	Modul Mezd informačního systému Mamlas	1046
Majetek	Majetek Mamlas	<input checked="" type="checkbox"/>	Modul Majetku informačního systému Mamlas	1049
Sklady	Sklady Mamlas	<input checked="" type="checkbox"/>	Modul Sklady informačního systému Mamlas	1052

Obrázek B.11: Definice sekce

Kód a Název jsou povinné údaje.

Alias sekce -- výběr sekce, pro kterou je sekce alias (obsahuje stejné produkty, jen má jiný název), uplatní se zejména u webové aplikace (popis implementace je v kapitole 5.10).

B.3.2 Struktura

Definice obchodní struktury. Definice obchodní struktury je zobrazena na obrázku B.12.

Definice, do kterých sekcí zákazníka má uživatel přístup a které bude uživatel mít zobrazeny ve webové aplikaci. Omezení lze nastavit na Povoleno vše (všechny sekce zákazníka), Pouze určené (přístup pouze do zvolených sekcí), Všechny kromě (přístup do všech sekcí kromě zvolených) a Zakázáno vše (uživatel nemá přístup do žádné sekce zákazníka).

B.3.3 Seznam produktů

Číselník produktů. Definice produktu je zobrazena na obrázku B.13.

Kód, Název a Sekce jsou povinné údaje. V Sekci se definuje, do které sekce produkt náleží. V části Aktualizační soubory se definují soubory produktu, které je možno stáhnout u webové aplikace.

Obrázek B.12: Definice obchodní struktury

Popis aktualizačního souboru	Název souboru	Řazení	Aktivní
Aktualizace Účetnictví	UctoUpdate.exe	0	<input checked="" type="checkbox"/>
Instalátor	Setup.exe	99	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Verze	Datum	ID verze	Popis	Aktivní	ID
20.0.1	01.01.2020	2001	Prvotní verze	<input checked="" type="checkbox"/>	1069

Obrázek B.13: Definice produktu

V části Přehled verzí jsou zobrazeny všechny verze produktu tvořící change log. Ukázka verze je produktu zobrazena na obrázku B.14.

Název verze a Datum jsou povinné údaje. ID Verze je vnitřní ID (číslo) verze. V části Aktualizační soubory se definují soubory verze, každá verze může mít své vlastní aktualizační soubory.

B.3.4 Seznam států

. Číselník států, používá se na adresách. Číselník států je zobrazen na obrázku B.15.

Seznam produktů | Produkt Účetnictví | Verze 20.0.1 X

Základní údaje

Název verze 20.0.1

Datum 01.01.2020 15

ID Verze 2 001

Aktivní

Aktualizační soubory

Popis aktualizacího souboru	Název souboru	Řazení	Aktivní
Aktualizace Účetnictví	UctoUpdate.exe	0	<input checked="" type="checkbox"/>
Instalátor	Setup.exe	99	<input checked="" type="checkbox"/>

Popis

Prvotní verze

Uložit Zavřít

Obrázek B.14: Ukázka verze je produktu

Seznam států X

Obnovit Nový Smazat Uložit Zavřít

Název	Tuzemsko	ISO kod	EU kod	Poznámka	ID
ČR	<input checked="" type="checkbox"/>			Kde domov můj	1
Německo	<input type="checkbox"/>				55
Slovensko	<input type="checkbox"/>			Nad Tatrou sa blýská	2
	<input type="checkbox"/>				

Obrázek B.15: Číselník států

Tuzemsko je příznak, že stát je tuzemsko (na faktuře se v adrese zobrazí státy mimo tuzemské). ISO kód a EU kód jsou určeny pro případné budoucí rozlišení -- přenos faktur do účetního systému z důvodu rozlišení státu odběratele v Kontrolním hlášení DPH.

B.4 Zákazníci

Evidence zákazníků je dostupná pod volbou Zákazníci. V seznamu zákazníků je možno filtrovat záznamy podle různých kritérií.

B.4.1 Číselníky

Volba obsahuje číselníky pro evidenci zákazníků.

- Status
- Skupina
- Kategorie
- Region

- Právní forma

B.4.2 Detail zákazníka

V detailu zákazníka se zadávají informace o zákazníkovi (název, adresa sídla), kontaktní a další informace, sekce zákazníka, kontaktní osoby zákazníka, pobočky a provozovny, bankovní účty a údaje pro fakturaci.

B.4.2.1 Základní údaje

V části základní údaje se zadává název a sídlo zákazníka a kontaktní informace. Definice základních údajů zákazníka je zobrazena na obrázku B.16.

Seznam zákazníků		Zákazník Zákazník X
Základní údaje		
Zkratka	Zákazník	
Název společnosti	Rodšildové a spol, a.s.	
Doplněk názvu		
DIČ	IČ	ARES
CZ88664422	88664422	
Zápis do OR		
Právní forma		
Sídlo		
Ulice	Číslo	
K pivovaru	13	
Doplněk adresy		
PSC	Město/Obec	
70800	Ostrava	
Městská část	Poruba	
Stát	ČR	
Kontaktní informace		
Kontaktní osoba		
Telefon		
Fax		
E-mail		
WWW		
Další kontakt		
Další kontakt		

Obrázek B.16: Definice základních údajů zákazníka

Zkratka je údaj, podle kterého se vybírá zákazník, jedná se o povinný údaj. Následují identifikační údaje společnosti a adresa sídla, u nového záznamu je stát nastaven podle Nastavení (viz B.2.3). Tyto údaje lze pomocí tlačítka ARES vyhledat a doplnit z databáze ARES. Poslední částí jsou kontaktní informace včetně možnosti dvou uživatelsky definovaných údajů (např. datová schránka).

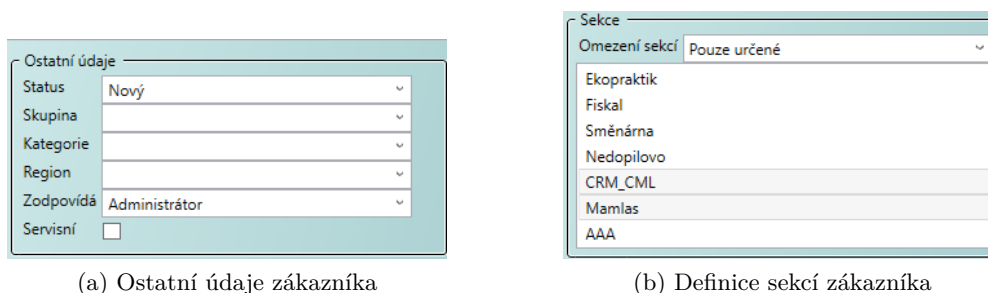
B.4.2.2 Ostatní údaje

V části ostatní údaje se zadávají další údaje k zákazníkovi. Nastavení ostatní údajů je zobrazeno na obrázku B.17a.

Status zákazníka, u nového záznamu je status nastaven podle Nastavení (viz B.2.3) slouží zejména kvůli určení aktivního či neaktivního zákazníka. Skupina, Kategorie a Region slouží k dalšímu členění zákazníků podle struktury a potřeb zadavatele. Zodpovídá je uživatel systému, který zákazníka „vlastní“, používá se zejména při řešení výkonných práv uživatele (např. uživatel systému vidí v základním seznamu jen své zákazníky). Při založení zákazníka se jako vlastník nastaví aktuálně přihlášený uživatel. Servisní je evidenční příznak zákazníka, určeno zejména pro další rozvoj systému.

B.4.2.3 Sekce

V části Sekce se definují sekce zákazníka. V projektu je možno zadat pouze produkty spadající pod povolené sekce zákazníka. Omezení lze nastavit na Povolené vše (všechny sekce), Pouze určené (produkty pouze ze zvolených sekcí), Všechny kromě (produkty všech sekcí kromě zvolených) a Zakázáno vše (zákazník nemá žádnou sekci). Nastavení sekcí je zobrazeno na obrázku B.17b.



Obrázek B.17: Ostatní údaje a sekce zákazníka

B.4.2.4 Kontaktní osoby

Část Kontaktní osoby obsahuje seznam kontaktních osob zákazníka, jejich jméno, funkci a kontaktní údaje. Část kontaktní osoby je zobrazena na obrázku B.18.

Kontaktní osoby

Titul před Telefon

Jméno Fax

Příjmení E-mail

Titul za WWW

Funkce Další kontakt

Další kontakt

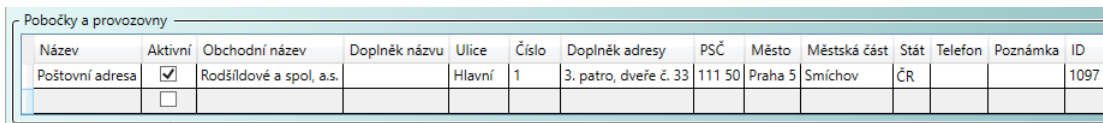
Poznámka

Jméno	Funkce	Telefon	E-mail	Poznámka	ID
Ing. Jan Hus	mistr				34
Josef Mák	ekonomický ředitel				35

Obrázek B.18: Kontaktní osoby zákazníka

B.4.2.5 Pobočky a provozovny

V části Pobočky a provozovny lze definovat pobočky a další provozovny (kontaktní adresy) zákazníka. Tyto se uplatní zejména ve fakturaci. Pobočky a provozovny zákazníka jsou zobrazeny na obrázku B.19.

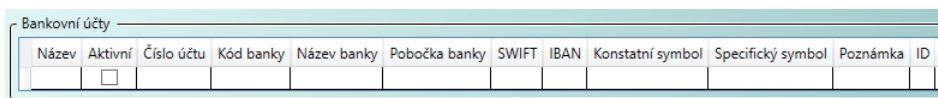


Název	Aktivní	Obchodní název	Doplňk názvu	Ulice	Číslo	Doplňk adresy	PSČ	Město	Městská část	Stát	Telefon	Poznámka	ID
Poštovní adresa	<input checked="" type="checkbox"/>	Rodšildové a spol, a.s.		Hlavní	1	3. patro, dveře č. 33	111 50	Praha 5	Smíchov	ČR			1097
	<input type="checkbox"/>												

Obrázek B.19: Pobočky a provozovny zákazníka

B.4.2.6 Bankovní účty

Bankovní účty zákazníka, využije se zejména u účetní jednotky a dodavatelů. Bankovní účty zákazníka jsou zobrazeny na obrázku B.20.

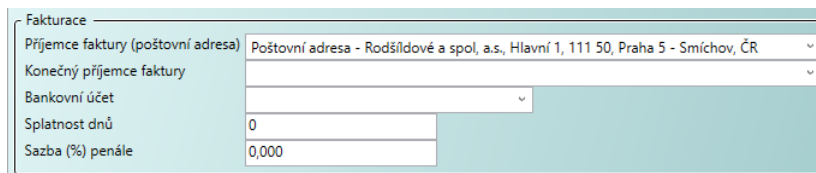


Název	Aktivní	Číslo účtu	Kód banky	Název banky	Pobočka banky	SWIFT	IBAN	Konstatní symbol	Specifický symbol	Poznámka	ID
	<input type="checkbox"/>										

Obrázek B.20: Bankovní účty zákazníka

B.4.2.7 Fakturace

V části fakturace jsou definovány údaje a nastavení, které jsou použity při vytvoření faktury pro zákazníka (odběratele). Definice fakturace zákazníka je zobrazena na obrázku B.21.



Příjemce faktury (poštovní adresa)	Poštovní adresa - Rodšildové a spol, a.s., Hlavní 1, 111 50, Praha 5 - Smíchov, ČR
Konečný příjemce faktury	
Bankovní účet	
Splatnost dnů	0
Sazba (%) penále	0,000

Obrázek B.21: Definice fakturace zákazníka

B.5 Úkoly

Evidence úkolů je dostupná pod volbou Úkoly. V seznamu úkolů je možno filtrovat záznamy podle různých kritérií. Možnosti filtrování úkolů jsou zobrazeny na obrázku B.22.

Popis filtrů:

Moje úkoly – úkoly přiřazené aktuálně přihlášenému uživateli.

Obrázek B.22: Možnosti filtrování úkolů

Hotovo – příznak, zda je či není úkol splněn.

Přiřazeno – jméno uživatele, kterému je úkol přiřazen s možností filtru na vyplnění/nevyplnění.

Projekt – filtr na projekt, ke kterému se úkol váže.

Termín od do – interval termínu od data do data.

Po termínu – výběr úkolů po termínu.

B.5.0.1 Detail úkolu

Úkol se definuje v detailu úkolu. Detail úkolu je zobrazen na obrázku B.23.

Obrázek B.23: Detail úkolu

Název úkolu je povinný údaj. Splněno je příznak, zda je či není úkol splněn. Termín je datum a případně čas požadovaného splnění úkolu. Přiřazeno je jméno uživatele, kterému je úkol přiřazen s možností filtru na vyplnění/nevyplnění.

B.6 Projekty

Evidence projektů. V seznamu projektů je možno filtrovat záznamy podle různých kritérií. Možnosti filtrování projektů jsou zobrazeny na obrázku B.24.

Popis filtrů:

Obrázek B.24: Možnosti filtrování projektů

Moje projekty – projekty vlastněné aktuálně přihlášeným uživatelem.

Zákazník – zákazník projektu; seznam zákazníků případně omezen dle obchodní struktury uživatele.

Vlastník – jméno uživatele, kterému je projekt přiřazen s možností filtru na vyplnění/nevyplnění.

Reklamace – filtr na příznak reklamace (Ano/Ne).

Termín dokončení od do – interval termínu dokončení projektu od data do data.

Po termínu – výběr projektů po termínu.

Produkty – výběr produktů, lze vybrat více produktů; seznam produktů případně omezen dle obchodní struktury uživatele s možností omezení na Aktivní produkty.

Status – status projektu s možností omezení na aktivní nebo neaktivní status.

Priorita – priorita projektu.

B.6.1 Číselníky

Číselníky pro evidenci projektů.

Způsob kontaktu – způsob získání kontaktu, informace o projektu (e-mail, telefon, apod.).

Status – status projektu s příznakem aktivní (dle něj se určuje, zda projekt je či není aktivní).

Priorita – priorita projektu.

B.6.2 Detail projektu

V detailu projektu se definují základní údaje o projektu, fáze a etapy, kalkulace, výkaz práce s celkovým přehledem kalkulovaných a provedených prací, evidence verzí (úprav) produktu v rámci projektu a evidenci úkolů projektu.

B.6.2.1 Základní údaje

V části základní údaje projektu se definují základní údaje o projektu. Definice základních údajů o projektu je zobrazena na obrázku B.25.

The screenshot shows a web application interface for project management. At the top, there are navigation tabs: 'Seznam projektů', 'Projekt Instalace', and 'Projekt Implementace systému CML X'. Below the tabs is a form titled 'Základní údaje'. The form contains the following fields:

Název	Implementace systému CML				
Zákazník	Zákazník	Status	Potvrzeno	Veřejný	<input type="checkbox"/>
Produkt	CML	Priorita	Střední	Číslo	<input type="text"/>
Kontaktní osoba	Mák Josef	Datum zahájení	12.03.2020	Reklamacie	<input type="checkbox"/>
Způsob kontaktu		Termín dokončení	30.04.2020	Vlastník	Administrátor
Datum kontaktu	12.03.2020 00:00	Datum ukončení	Vybrat datum		

Below the form are three text areas for notes:

- Zadání (veřejné): Implementace systému CML
- Veřejná poznámka: (empty)
- Interní poznámky: (empty)

Obrázek B.25: Definice základních údajů o projektu

Název, Zákazník a Produkt je povinný údaj.

Seznam zákazníků je omezen dle obchodní struktury uživatele.

Seznam produktů je omezen na sekce zákazníka a případně dále omezen dle obchodní struktury uživatele.

Seznam Kontaktní osoba je ze seznamu kontaktních osob zákazníka, do seznamu lze zadat i osobu mimo seznam.

Způsob kontaktu je zdroj požadavku projektu (např. e-mail, Skype)

Datum a čas kontaktu se u nového projektu nastaví podle aktuálního data a času.

Status projektu slouží zejména kvůli určení aktivního či neaktivního projektu. U nového záznamu je status nastaven podle Nastavení (viz B.2.3)

Priorita řešení projektu. U nového záznamu je priorita nastavena podle Nastavení (viz B.2.3).

Datum zahájení je datum přijetí či zahájení prací na projektu.

Termín dokončení je datum termínu dokončení projektu, ovlivňuje stupeň eskalace. Veřejný je příznak veřejného projektu (do budoucna bude zobrazeno ve webové aplikaci).

Číslo je pro veřejný projekt generované číslo projektu.

Reklamacie je příznak projektu, že se jedná o reklamaci (pro evidenci reklamací).

Vlastník je uživatel systému, který projekt „vlastní“, používá se zejména při řešení výkonných práv uživatele (např. uživatel systému vidí v základním seznamu jen své projekty). Při založení projektu se jako vlastník nastaví aktuálně přihlášený uživatel.

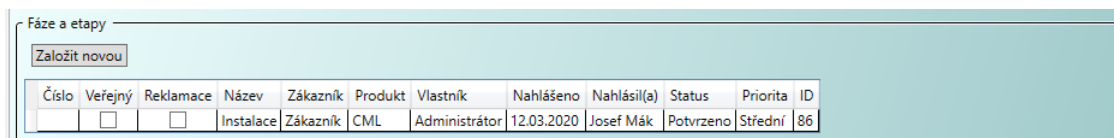
Zadání (veřejné) je zadání/popis požadavku poskytnuté zákazníkem (do budoucna bude zobrazeno ve webové aplikaci).

Veřejná poznámka je poznámka k řešení projektu, případně zpětná zpráva pro zákazníka (do budoucna bude zobrazeno ve webové aplikaci).

Interní poznámky jsou interní poznámky řešitele, použité postupy, odkazy na zdroje, informace k řešení (nebude nikdy zobrazeno zákazníkovi).

B.6.2.2 Fáze a etapy

Fáze a etapy projektu. Každý projekt může mít neomezený počet podřízených projektů a etap, uspořádání je do stromové struktury vždy s právě jedním kořenovým projektem. Fáze a etapy projektu jsou zobrazeny na obrázku B.26.



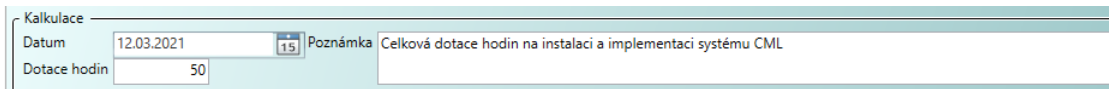
Číslo	Veřejný	Reklamacie	Název	Zákazník	Produkt	Vlastník	Nahlášeno	Nahlásil(a)	Status	Priorita	ID
	<input type="checkbox"/>	<input type="checkbox"/>	Instalace	Zákazník	CML	Administrátor	12.03.2020	Josef Mák	Potvrzeno	Střední	86

Obrázek B.26: Fáze a etapy projektu

Pomocí tlačítka Založit novou se založí nová fáze či etapa projektu (aktuálně přihlášený uživatel musí mít právo na přidávání projektů). Do nově založené fáze projektu jsou zkopírovány všechny základní údaje z rodičovského projektu (název s prefixem „FÁZE:“) a interní poznámka. Nově založená etapa je následně zobrazena v seznamu etap a je otevřen její detail pro další editaci a doplnění.

B.6.2.3 Kalkulace

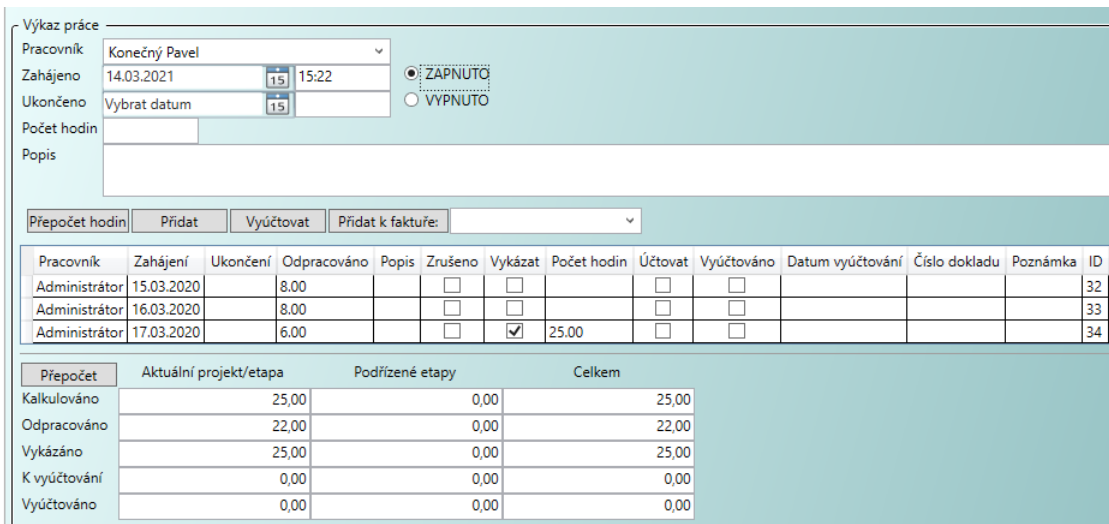
V části Kalkulace se zadává počet kalkulovaných hodin na realizaci projektu, případně fáze či etapy projektu. Kalkulace projektu je zobrazena na obrázku B.27.



Obrázek B.27: Kalkulace projektu

B.6.2.4 Výkaz práce

V části Výkaz práce se evidují provedené práce na projektu a provádí se jejich vyúčtování. Další součástí je souhrnný přehled prací a to včetně podřízených etap. Výkaz práce projektu je zobrazen na obrázku B.28.



Pracovník	Zahájení	Ukončení	Odpracováno	Popis	Zrušeno	Vykázáno	Počet hodin	Účtovat	Vyúčtováno	Datum vyúčtování	Číslo dokladu	Poznámka	ID
Administrátor	15.03.2020		8,00		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>				32
Administrátor	16.03.2020		8,00		<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>				33
Administrátor	17.03.2020		6,00		<input type="checkbox"/>	<input checked="" type="checkbox"/>	25,00	<input type="checkbox"/>	<input type="checkbox"/>				34

Přepočít	Aktuální projekt/etapa	Podřízené etapy	Celkem	
Kalkulováno		25,00	0,00	25,00
Odpracováno		22,00	0,00	22,00
Vykázáno		25,00	0,00	25,00
K vyúčtování		0,00	0,00	0,00
Vyúčtováno		0,00	0,00	0,00

Obrázek B.28: Výkaz práce projektu

Přehled aktivit ve výkazu práce:

Zadání provedené práce – zadání provedené práce lze provést dvěma způsoby – ručně nebo pomocí časovače (stopek).

Při ručním zadání obsluha vybere pracovníka a zadá počet hodin buď přímo anebo přes datum a čas zahájení a datum a čas ukončení (variantně lze zadat jen čas zahájení a čas ukončení, případně datum a čas zahájení a čas ukončení) a volitelně zadá popis provedené práce. Pomocí tlačítka Přepočít hodin se vypočte počet odpracovaných hodin zaokrouhlených na čtvrt hodiny a poté pomocí tlačítka Přidat vloží zadanou práci do výkazu práce.

Druhou možností je využít časovače (stopek). Pomocí přepínače ZAPNOU se do data a času zahájení vloží aktuální datum a čas a jako pracovník se nastaví aktuálně přihlášený uživatel. Po ukončení práce se pomocí přepínače VYPNOU časovač vypne, spočítá se počet hodin zaokrouhlených na čtvrt hodiny a nabídne se vložení do výkazu práce, které lze v případě potřeby zrušit, vypočtené údaje upravit a práci do výkazu vložit ručně pomocí tlačítka Přidat.

Editace výkazu práce – V seznamu provedených prací lze vykázané hodiny upravovat a doplňovat, včetně popisu, hodiny pomocí příznaku Zrušit z výkazu hodiny zrušit

Hodiny, které se budou vykazovat zákazníkovi se označí pomocí příznaku Vykázat. Počet vykazovaných hodin lze ručně upravit.

Práce určené k vyúčtování (fakturou) se označí pomocí příznaku Účtovat. Vyúčtované práce jsou označeny příznakem Vyúčtováno s informací o datu vyúčtování a čísle dokladu, kterým byla práce vyúčtována. Příznak Vyúčtováno lze ručně změnit a takto „odznačená“ práce lze vyúčtovat znovu (např. na jinou fakturu).

Vyúčtování prací – je možno provést dvěma způsoby: Pomocí tlačítka Vyúčtovat vyúčtování do nové faktury anebo pomocí tlačítka Přidat k faktuře přidat na vybranou fakturu (nabízí se všechny neuzamčené faktury vystavené na zákazníka). Vyúčtovávají se všechny práce ve výkazu, které nejsou zrušené, jsou označeny Účtovat a nejsou vyúčtovány.

Souhrnný přehled prací – souhrnný přehled kalkulace a provedených prací a to včetně všech podřízených fází a etap v celé stromové struktuře. Přehled se aktualizuje ručně pomocí tlačítka Přepočít.

B.6.2.5 Přehled verzí

Evidence verzí produktu vytvořených realizací projektu. V sekci lze pro produkt projektu zakládat, přidávat, odstraňovat či měnit verzi produktu, která souvisí s projektem. Evidence verzí produktu projektu je zobrazena na obrázku B.29.

Verze	Datum	ID verze	Produkt	Popis
20.0.3	12.03.2020	2003	CML	oprava chyby při přihlášení uživatele, pokud uživatel neměl definováno heslo

Obrázek B.29: Evidence verzí produktu projektu

Přehled aktivit v evidenci verzí:

Zobrazit – zobrazí vybranou verzi (totéž provede i dvojklik na verzi v seznamu).

Obnovit – ruční obnovení seznamu verzí.

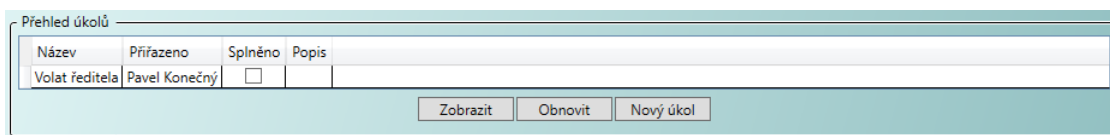
Založit verzi – založí novou verzi produktu.

Přidat k verzi – přidá k produktu verzi nastavenou v seznamu Aktuální verze produktu (přednastavena je poslední verze produktu).

Odstranit z verze – odstraní vazbu projektu na vybranou verzi seznamu.

B.6.2.6 Přehled úkolů

Evidenci úkolů spojených s projektem. Evidence úkolů projektu je zobrazena na obrázku B.30.



Název	Přřazeno	Splněno	Popis
Volat ředitela	Pavel Konečný	<input type="checkbox"/>	

Zobrazit Obnovit Nový úkol

Obrázek B.30: Evidence úkolů projektu

Přehled aktivit v evidenci úkolů:

Zobrazit – zobrazí vybraný úkol (totéž provede i dvojklik na úkol v seznamu).

Obnovit – ruční obnovení seznamu verzí.

Nový úkol – založí nový úkol k projektu.

B.7 Fakturace

Evidence faktur vydaných faktur. Systém umožňuje vystavit řádný daňový doklad a zálohou fakturu. Druh dokladu je nastaven v číselné řadě faktury.

B.7.1 Číselníky

Číselníky pro evidenci faktur.

B.7.1.1 Bankovní účty

Číselník bankovních účtů dodavatele faktury. Číselník bankovních účtů je zobrazen na obrázku B.31.

B.7.1.2 Cizí měny

Číselník cizích měn, které lze použít pro vystavení faktury. Číselník cizích měn je zobrazen na obrázku B.32.

Množství je jednotkové množství měny z kursovního lístku (kurz xx,xxx Kč za množství měny).

Bankovní účty X											
⋮ Obnovit Nový Smazat Uložit Zavřít ⌵											
Název	Číslo účtu	Kód banky	Název banky	Pobočka banky	SWIFT	IBAN	Aktivní	Konstatní symbol	Specifický symbol	Poznámka	ID
FIO	1234567890	2010	FIO banka a.s.				<input checked="" type="checkbox"/>				588
KB	666	0100	Komerční banka				<input checked="" type="checkbox"/>				589
							<input type="checkbox"/>				

Obrázek B.31: Číselník bankovních účtů

Cizí měny X						
⋮ Obnovit Nový Smazat Uložit Zavřít ⌵						
Kód	Název	Aktivní	Množství	Popis	ID	
EUR	Euro	<input checked="" type="checkbox"/>			592	
HUF	Forint	<input checked="" type="checkbox"/>	100		623	
		<input type="checkbox"/>				

Obrázek B.32: Číselník cizích měn

B.7.1.3 Sazby DPH

Číselník sazeb DPH. Číselník sazeb DPH je zobrazen na obrázku B.33.

Sazby DPH X						
⋮ Obnovit Nový Smazat Uložit Zavřít ⌵						
Druh sazby	Procentní sazba	Označení	Platnost od	Platnost do	Popis	ID
Bez DPH	0.00	0 %				639
Základní sazba	23.00	23 %	01/01/1993	31/12/1994		645
Základní sazba	22.00	22 %	01/01/1995	30/04/2004		646
Základní sazba	20.00	20 %	01/01/2010	31/12/2012		648
Základní sazba	21.00	21 %	01/01/2013			649
Základní sazba	19.00	19 %	01/05/2004	31/12/2009		647
Snížená sazba	5.00	5 %	01/01/1993	31/12/2007		640
Snížená sazba	9.00	9 %	01/01/2008	31/12/2009		641
Snížená sazba	10.00	10 %	01/01/2010	31/12/2011		642
Snížená sazba	14.00	14 %	01/01/2012	31/12/2012		643
Snížená sazba	15.00	15 %	01/01/2013			644
Druhá snížená sazba	10.00	10 %	01/01/2015			650

Obrázek B.33: Číselník sazeb DPH

Obsahuje sazby DPH pro příslušný druh (Základní sazba, Snížená sazba, Druhá snížená sazba) s časovou platností sazby.

B.7.1.4 Způsob úhrady

Číselník způsobů úhrad faktur. Číselník způsobů úhrad faktur je zobrazen na obrázku B.34.

Příznak Hotovost znamená úhradu v hotovosti (pro různé zaokrouhlení celkové částky faktury při úhradě v hotovosti anebo jiným způsobem, např. bankovním převodem).

Způsob úhrady X				
: Obnovit Nový Smazat Uložit Zavřít ▾				
Název	Aktivní	Hotovost	Popis	ID
Převodní příkaz	<input checked="" type="checkbox"/>	<input type="checkbox"/>		569
V hotovosti	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		570
	<input type="checkbox"/>	<input type="checkbox"/>		

Obrázek B.34: Číselník způsobů úhrad faktur

B.7.1.5 Způsob dopravy

Číselník způsobů dopravy. Číselník způsobů dopravy je zobrazen na obrázku B.35.

Způsob dopravy X				
: Obnovit Nový Smazat Uložit Zavřít ▾				
Název	Aktivní	Popis	ID	
Česká pošta	<input checked="" type="checkbox"/>		626	
	<input type="checkbox"/>			

Obrázek B.35: Číselník způsobů dopravy

B.7.1.6 Číselné řady faktur

Číselník číselných řad faktur. Číselník je určen pro definici různých řad faktur včetně nastavení typu faktury (daňový doklad nebo zálohová faktura). Číselník číselných řad faktur je zobrazeno na obrázku B.36.

Význam položek číselníku:

Kód – kód číselné řady.

Druh dokladu – druh dokladu (daňový doklad, zálohová faktura).

Dodavatel – dodavatel na faktuře, pokud je jiný, než účetní jednotka, použije se při fakturaci za dodavatele.

Název – název číselné řady, je zobrazen společně s kódem ve fakturaci pro založení nové faktury.

Cizí měna – fakturace v cizí měně, neurčeno znamená fakturu v tuzemské měně dle Nastavení (viz B.2.3).

Bankovní účet – přednastavený bankovní účet dodavatele.

Konstantní symbol, Specifický symbol – doplňující symboly pro úhradu faktury.

Prefix – prefix/předpona čísla faktury.

Obrázek B.36: Číselník číselných řad faktur

Maska číslování – maska formátu zobrazovaného pořadového čísla faktury (maska 00000 zobrazí pořadové číslo 12 jako 00012).

Sufix – sufix/přípona čísla faktury.

Počáteční číslo – počáteční číslo faktury; použije se v případě, že v systému není v číselné řadě vystavena žádná faktura.

Forma úhrady – přednastavená forma úhrady faktury.

Způsob dopravy – přednastavený způsob dopravy.

Splatnost dnů – přednastavená splatnost ve dnech.

Výpočet splatnosti – možnost definice výpočtu data splatnosti, pokud vypočtená splatnost připadne na víkend (Jen přičíst – neprovádí se kontrola, zda datum splatnosti připadne na víkend nebo zda v tomto případě bude datum splatnosti upraveno na splatnosti v pátek před víkendem anebo v pondělí po víkendu).

Sazba % penále – přednastavená sazba penále v procentech.

Zaokrouhlení v tuzemské měně, Zaokrouhlení v cizí měně – způsob zaokrouhlení celkové částky faktury (bez zaokrouhlení, na celé matematicky, vždy nahoru, vždy dolů).

Typ sazby DPH – přednastavený typ sazby DPH pro položky faktury.

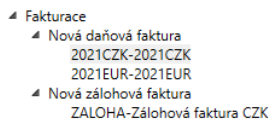
Text před položkami – text před položkami faktury (např. Fakturujeme vám za).

Text pod položkami – text pod položkami a rekapitulací DPH a částky faktury (např. Těšíme se na další spolupráci).

Aktivní – příznak, zda je číselná řada aktivní.

B.7.2 Vystavení nové faktury

Vystavení faktury se provede výběrem požadovaného druhu a číselné řady faktury pod nabídkou Fakturace. Možnosti vystavení faktury jsou zobrazeny na obrázku B.37.



Obrázek B.37: Možnosti vystavení faktury

B.7.3 Detail faktury

Detail faktury se skládá z několika částí: hlavička faktury, přehled účtovaných prací, položky faktury a definice textů na faktuře.

B.7.3.1 Záhlaví faktury

Záhlaví faktury obsahuje základní údaje o faktuře: číselná řada, dodavatel, odběratel, údaje o vystavení a splatnosti, rekapitulace částek faktury a další doplňující údaje. Záhlaví faktury je zobrazeno na obrázku B.38.

Přehled sekcí záhlaví faktury:

Číselná řada faktury – číselná řada faktury. Řadu lze měnit jen u nových faktur vzniklých vyúčtováním prací projektu.

Uzamčeno — příznak faktury, že je uzamčena a nelze na ní již žádné údaje měnit. Změnit příznak může jen uživatel s právem administrátora.

Seznam projektů	Projekt Instalace	Faktura 210100011CZK X
Číselná řada faktury Číselná řada <input type="text" value="2021CZK"/> Uzamčeno <input type="checkbox"/>		
Dodavatel PALEKON <input type="text"/> Sídlo - PALEKON SOFT s.r.o., Ahepjukova <input type="text"/> IČ 29441277 DIČ CZ29441277 PALEKON SOFT s.r.o. Ahepjukova 2788/2 70200 Ostrava - Moravská Ostrava Bankovní účet <input type="text" value="FIO 1234567890/2010"/>		Číselné hodnoty Číslo faktury <input type="text" value="2101"/> <input type="text" value="00011"/> <input type="text" value="CZK"/> Dobropis <input type="checkbox"/> k <input type="text"/> Druh dodávky <input type="text" value="software"/> Smlouva/Objednávka <input type="text" value="PO 806251/2021"/> Dodací list číslo <input type="text"/>
Příjemce faktury (poštovní adresa) Zákazník <input type="text"/> Poštovní adresa - Rodšildové a spol, a.s., <input type="text"/> Rodšildové a spol, a.s. Hlavní 1 3. patro, dveře č. 33 111 50 Praha 5 - Smíchov		Odběratel Zákazník <input type="text"/> Sídlo - Rodšildové a spol, a.s., K pivovaru <input type="text"/> IČ 88664422 DIČ CZ88664422 Rodšildové a spol, a.s. K pivovaru 13 70800 Ostrava - Poruba
Konečný příjemce / Provozovna <input type="text"/> <input type="text"/>		Údaje o vystavení a splatnosti Forma úhrady <input type="text" value="Převodní příkaz"/> Datum UZP <input type="text" value="14.03.2021"/> <input type="text" value="15"/> Datum vystavení <input type="text" value="14.03.2021"/> <input type="text" value="15"/> Splatnost dnů <input type="text" value="30"/> Datum splatnosti <input type="text" value="14.03.2021"/> <input type="text" value="15"/> Datum odeslání <input type="text" value="Vybrat datum"/> <input type="text" value="15"/>
Doplňující údaje Sazba (%) penále <input type="text" value="0,00"/> Způsob dopravy <input type="text"/> Referent <input type="text" value="Administrátor"/>		Rekapitulace Částka celkem <input type="text" value="48,400.00"/> Zaokrouhlení <input type="text"/> Celkem netto <input type="text" value="48,400.00"/> Zaplacená záloha <input type="text"/> Částka k úhradě <input type="text" value="48,400.00"/> Kč

Obrázek B.38: Záhlaví faktury

Dodavatel – dodavatel faktury, u nové faktury se nastaví dle nastavení účetní jednotky v Nastavení systému, případně podle nastaveného dodavatele v číselníku číselných řad faktur. Údaj může změnit uživatel s právem administrátora.

Číselné hodnoty – číslo faktury a další údaje o faktuře.

Číslo faktury — je složeno z prefixu/předpony, pořadového čísla v rámci číselné řady a prefixu a sufixu/přípony.

Dobropis — příznak, že doklad je dobropisem k jinému daňovému dokladu. V případě, že se jedná o dobropis, uvede, ke kterému číslu dokladu.

Druh dodávky — doplňující údaj druh dodávky

Smlouva/Objednávka — číslo smlouvy nebo objednávky

Dodací list číslo — číslo dodacího listu

Odběratel – odběratel (zákazník) faktury. Údaj se skládá ze dvou částí: výběr ze seznamu zákazníků a výběr z adres vybraného zákazníka. Příjemce faktury (poštovní adresa) je poštovní adresa odběratele. Údaj se skládá ze dvou částí: výběr ze seznamu zákazníků a výběr z adres

vybraného zákazníka, je přenastaven se odběratel a jeho poštovní adresa, pokud je definována, anebo adresa (pobočky a provozovny) sídla. Konečný příjemce / Provozovna je konečný příjemce faktury. Údaj se skládá ze dvou částí: výběr ze seznamu zákazníků a výběr z adres (pobočky a provozovny) tohoto zákazníka.

Údaje o vystavení a splatnosti – splatnost faktury a časové údaje k faktuře.

Forma úhrady — forma úhrady, výběr z číselníku formy úhrady, lze zadat i údaj mimo číselník.

Datum UZP — datum uskutečnění zdanitelného plnění (nebo datum přijetí platby), pro nový doklad přednastaveno aktuální datum.

Datum vystavení — datum vystavení dokladu, údaj se použije s použitím počtu dnů splatnosti pro výpočet data splatnosti. pro nový doklad přednastaveno aktuální datum.

Splatnost dnů — počet dnů splatnosti, údaj se použije s použitím data vystavení pro výpočet data splatnosti, pro nový doklad přenastaveno dle nastavení systému, údaje z číselníku číselných řad faktur nebo odběratele.

Datum splatnosti – datum splatnosti faktury.

Datum odeslání – doplňující údaj datum odeslání dokladu.

Rekapitulace – rekapitulace částek faktury.

Částka celkem — celková částka za položky faktury, hodnota je vypočtena automaticky z celkových hodnot položek.

Zaokrouhlení — zaokrouhlení faktury, hodnota je vypočtena podle nastavení číselné řady.

Celkem netto — částka po zaokrouhlení, hodnota je vypočtena systémem.

Zaplacená záloha — částka uhrazené zálohy, hodnotu zadává uživatel.

Částka k úhradě — celková částka k úhradě po odečtu případné zálohy, hodnota je vypočtena systémem.

B.7.3.2 Účtované práce

Přehled všech prací vyúčtovaných dokladem. Přehled prací vyúčtovaných dokladem je zobrazen na obrázku B.39.

Projekt	Číslo	Pracovník	Zahájení	Ukončení	Odpracováno	Popis	Zrušeno	Počet hodin	Datum vyúčtování	Poznámka	ID
Instalace		Administrátor	17.03.2020		6,00		<input type="checkbox"/>	25,00	14.03.2021		34
Implementace systému CML		Administrátor			2,00	analýza	<input type="checkbox"/>	2,00	14.03.2021		36
Implementace systému CML		Eda			5,00	analýza	<input type="checkbox"/>	5,00	14.03.2021		37

Zobrazit | Vyúčtovat práce | Odúčtovat práce | Počet účtovaných hodin:

Obrázek B.39: Přehled prací vyúčtovaných dokladem

Přehled aktivit v přehledu účtovaných prací:

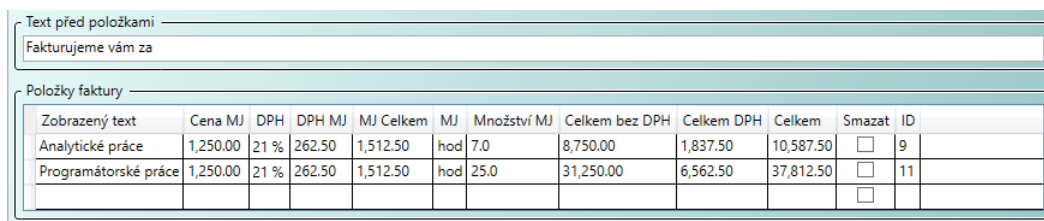
Zobrazit – zobrazí detail projektu.

Vyúčtovat práce – vyúčtuje všechny nevyúčtované práce k vyúčtování z projektů, kde je odběratel zákazníkem. Před vlastním provedení vyúčtování se provede kontrola existence nevyúčtovaných prací a zobrazí se dialog pro potvrzení akce.

Odúčtovat práce – provede odúčtování aktuálně vybrané práce z faktury.

B.7.3.3 Položky faktury

Položky faktury, zobrazí se na tiskové sestavě faktury. Položky faktury jsou zobrazeny na obrázku B.40.



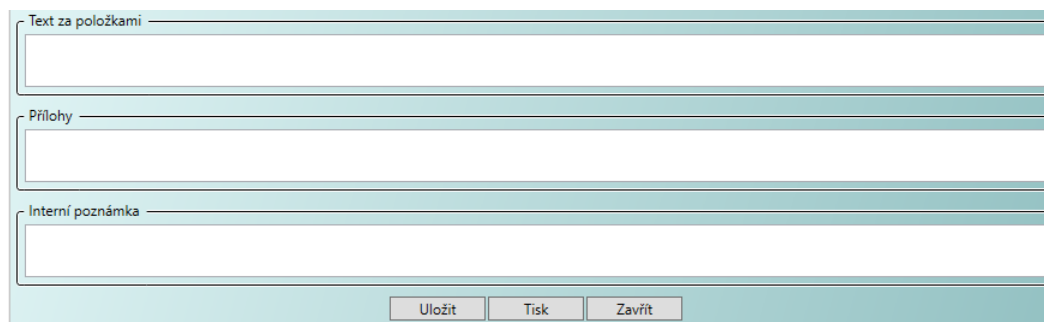
Zobrazení text	Cena MJ	DPH	DPH MJ	MJ Celkem	MJ	Množství MJ	Celkem bez DPH	Celkem DPH	Celkem	Smazat	ID
Analytické práce	1,250.00	21 %	262.50	1,512.50	hod	7.0	8,750.00	1,837.50	10,587.50	<input type="checkbox"/>	9
Programátorské práce	1,250.00	21 %	262.50	1,512.50	hod	25.0	31,250.00	6,562.50	37,812.50	<input type="checkbox"/>	11

Obrázek B.40: Položky faktury

Po zadání hodnot systém dopočítá příslušné navazující hodnoty (např. po zadání množství vypočte celkové hodnoty. Řádek lze smazat pomocí příznaku Smazat.

B.7.3.4 Texty na faktuře

Definice textů na faktuře. Definice textů na faktuře je zobrazena na obrázku B.41.



Obrázek B.41: Definice textů na faktuře

Přehled položek definice textů faktury:

Text za položkami – text bude zobrazen v tiskové sestavě faktury pod položkami.

Přílohy – text bude zobrazen v tiskové sestavě faktury pod položkami.

Interní poznámka – interní poznámka k dokladu, nezobrazuje se na tiskové sestavě.

B.7.3.5 Tisk faktury

Tisk faktury se provede pomocí tlačítka Tisk v detailu faktury anebo v seznamu faktur s možností výběru (označení) více faktur. Náhled tiskové sestavy faktury je zobrazen na obrázku B.42.

Tisk faktury

Zoom 1 of 1 Close

DODAVATEL DIČ: CZ29441277 IČ: 29441277 PALEKON SOFT s.r.o. Aheplukova 2788/2 70200 Ostrava - Moravská Ostrava		FAKTURA - DAŇOVÝ DOKLAD Daňový doklad číslo: 21010001CZK Druh dodávky: software Smlouva/Objednávka: PO 806251/2021 Dodací list číslo:				
Banka: FIO banka a.s. Číslo účtu: 1234567890/2010 IBAN, SWIFT: Telefon: E-mail:		Rodšildové a spol., a.s. Hlavní 1 3. patro, dveře č. 33 111 50 Praha 5 - Smíchov				
ODBĚRATEL DIČ: CZ388664422 IČ: 88664422 Rodšildové a spol., a.s. K pivovaru 13 70800 Ostrava - Poruba		Datum splatnosti: 14.03.2021 Způsob úhrady: Převodní příkaz Datum UZP / přijetí platby: 14.03.2021 Datum vystavení: 14.03.2021 Penalizace %: 0,00 Způsob dopravy:				
Dodávka	Množství	Cena/MJ	Cena*Množství	% DPH	Částka DPH	Celkem v Kč
<i>Fakturuje vám za</i>						
Analytické práce	7 hod	1 250,00	8 750,00	21	1 837,50	10 587,50
Programátorské práce	25 hod	1 250,00	31 250,00	21	6 562,50	37 812,50
Rozpis DPH	Základ	Výše DPH	Celkem	K úhradě celkem 48 400,00 Kč		
21 %	40 000,00	8 400,00	48 400,00			
Celkem	40 000,00	8 400,00	48 400,00			

Vystavil(a): Administrátor

Strana 1 z 1

Obrázek B.42: Náhled tiskové sestavy faktury

Příloha C

Ostatní přílohy

C.1 Struktura souboru XML pro uložení nastavení aplikace

C.2 Ukázka XML souboru Settings.xml

Ukázka XML souboru Settings.xml 5.4 (strana 29).

```
<?xml version="1.0" encoding="UTF-8"?>
<Settings>
  <Params>
    <ServerName>SERVER\MSSQL2014</ServerName>
    <DatabaseName>VSB_UDBS</DatabaseName>
    <Trusted_Connection>1</Trusted_Connection>
    <Connection>Network=DBMSSOCN</Connection>
  </Params>
  <Databases>
    <Client ID="1">
      <Name>Demo MDB data</Name>
      <DataPath>C:\Data\VSB\WpfProjektBC\Data\Demo</DataPath>
      <FileName>Data.mdb</FileName>
      <Skip>0</Skip>
    </Client>
    <Client ID="2">
      <Name>SQL 2016 Test</Name>
      <ServerName>SERVER\MSSQL2016</ServerName>
      <DatabaseName>VSB_TEST</DatabaseName>
      <Trusted_Connection>1</Trusted_Connection>
      <Connection>Network=DBMSSOCN</Connection>
    </Client>
  </Databases>
</Settings>
```

Tabulka C.1: Podřízené elementy pro <Databases>

<Name>	Název (uživatelský, zobrazovaný) datového klienta (možný alias: <Klient>)
<DBMS>	Typ databáze (MSSQL, MSSQLSERVICE, MDB, ORACLE, XML; výchozí hodnota MSSQL)
<DLL>	Název souboru dll (výchozí hodnota pro DBMS=MSSQL je DatabaseMSSQL.dll)
<PathName>	Cesta k datovému souboru, je-li souborová db (možný alias: <FilePath>)
<FileName>	Název datového souboru
<ServerName>	Název serveru (možný alias: <Server>)
<Catalog>	Název databáze (možný alias: <Database>, <DatabaseName>)
<UID>	Přihlašovací jméno
<PWD>	Přihlašovací heslo (otevřené)
<PWD_ENCRYPTED>	Zašifrované heslo
<WinAuth>	1/true: Windows authentication mode (možný alias: <Trusted_Connection>)
<Host>	Host name
<Port>	Číslo portu
<Connection>	Connection string nebo jeho část (sloučí se s dalšími elementy typu Server, Database, UID, PWD, atp.)
<AppName>	Název aplikace (titulek hlavního okna aplikace)
<Ignore>	1/true nepoužívat (možný alias: <Skip>)
<DataClientID>	Defaultní Data client id
<ServiceURL>	URL webové služby
<ServiceKey>	Public key šifrování webové služby
<ServiceNoCheck>	Vynucené zakázání testování dostupnosti MSSQL serveru a databáze při použití webové služby

Tabulka C.2: Podřízené elementy pro <Params>

<AppName>	Název aplikace (titulek hlavního okna aplikace)
<Language>	Jazyk (pro budoucí použití)

```

</Client>
<Client ID="3">
  <DBMS>XML</DBMS>
  <Klient>XML data</Klient>
  <DataPath>C:\Data\VSB\WpfProjektBC\XML</DataPath>
</Client>
<Client ID="4">
  <Klient>WebService SQL 2016 Test</Klient>
  <ServerName>SERVER\MSSQL2016</ServerName>
  <DatabaseName>VSB_TEST</DatabaseName>
  <Trusted_Connection>1</Trusted_Connection>
  <ServiceURL>https://www.*****.*/WebDB/MSSQL</ServiceURL>
  <ServiceKey>*****</ServiceKey>
  <ServiceNoCheck>1</ServiceNoCheck>
</Client>
</Databases>
</Settings>

```

Listing C.1: Ukázka XML souboru Settings.xml

C.3 Seznam rozlišení tříd číselníků

Příloha rozlišení tříd číselníků 5.7 (strana 35) v tabulce číselníků dle atributu ClassID. Číselníky systému jsou uloženy v jedné společné tabulce (mapování dědičnosti Single Table Inheritance 5.2). Rozlišení třídy v záznamu je prostřednictvím atributu ClassID. Seznam rozlišení tříd je uveden v tabulce C.3 na straně 88.

Tabulka C.3: Seznam rozlišení tříd číselníků

Seznam rozlišení tříd číselníků	
ClassID	Popis
1000	Číselník produktů
1010	Seznam souborů produktu
1020	Seznam verzí produktu
1110	Status zákazníka
1120	Skupina zákazníka
1130	Kategorie zákazníka
1140	Region zákazníka
Pokračování na další straně...	

Pokračování tabulky C.3	
ClassID	Popis
1150	Právní forma zákazníka
1161	Seznam smluv/objednávek zákazníka
1162	Seznam dodacích listů zákazníka
1170	Seznam sekcí zákazníka
1171	Rozsah povolení sekce
1172	Parametry fakturace
1180	Seznam bankovních účtů zákazníka
1190	Seznam provozoven/poboček zákazníka
1210	Zdroj požadavku projektu
1220	Status projektu
1230	Priorita projektu
1240	Seznam verzí produktu k projektu
1300	Číselník sekcí
1400	Seznam referencí
1500	Seznam souborů
1600	Seznam úkolů
1700	Seznam práv role uživatelů
1710	Definice typů práv
1800	Seznam rolí uživatele
1810	Role uživatele ve webové aplikaci
1900	Seznam práv uživatelské role
2000	Definice obchodní struktury
2010	Seznam rozsahu povolení
2020	Seznam Sekcí
2100	Seznam cizích měn
2105	Typy sazeb DPH
2110	Sazby DPH
2120	Seznam způsobů úhrady
2130	Seznam způsobů dopravy
2140	Číselné řady faktur
2150	Seznam druhů dodávek faktury
2200	Seznam sestav
2300	Seznam bankovních účtů
2400	Seznam stránek webové prezentace (public)
2500	Definice SMTP serveru
Pokračování na další straně...	

Pokračování tabulky C.3	
ClassID	Popis
2510	Definice eskalace pro projekty a úkoly
2520	Záznam historie odeslaných zpráv
10100	Parametr výchozí stát
10110	Parametr výchozí status zákazníka
10220	Parametr výchozí status projektu
10230	Parametr výchozí priorita projektu
10240	Parametr počet dnů, odkdy je projekt po termínu
10300	Parametr výchozí číslo faktury
10310	Parametry fakturace
10400	Parametr počet dnů, odkdy je úkol po termínu
10500	Výchozí role uživatelů
Konec tabulky C.3	

C.4 Výpis JSON komunikace webové služby

Výpisy JSON komunikace webové služby.

C.4.1 Validní požadavky

Výpis validní JSON komunikace webové služby.

C.4.1.1 Požadavek na přístup

Požadavek na přístup.

```
{
  "Token": "",
  "PublicKey": "7Fqa0jgyg8NkE0C0",
  "Checksum": "hEsgbYZszQvbLmaTrjQCYFhJLcPocCoPNvxNbedSj01vZpUeVS1JbWcewNADdCZ1",
  "Data": "iaeV/01rv0fCVZM1CbIk+t8mccb+9pTZ+iQQurn9Vr/
  uEpA08bkuSszIT0rS7q0a5Euote1MdS78XjertV+I/uoWMA1vFtuEguyEB1DIJJv/
  h7AERV7hoqook6oS471o35WpOD07WwZo3K6Zu7rjHTcFqgJFxFwdoncGYnc195KbQUhjNFJbJHVB7tiHt0AprM4Uac
  =="
```

Listing C.2: Validní požadavek na přístup

C.4.1.2 Návrátová hodnota

Návrátová hodnota požadavku na přístup – vrácen token.

```
{  
  "Token": "L66uLYbo2EhFL7jh+4ErSJ02uadRNnbB"  
}
```

Listing C.3: Návrátová hodnota: token

C.4.1.3 Požadavek na vykonání SQL příkazu

Požadavek na vykonání SQL příkazu.

```
{  
  "Token": "L66uLYbo2EhFL7jh+4ErSJ02uadRNnbB",  
  "PublicKey": "7Fqa0jgyg8NkE0C0",  
  "Checksum": "hEsgbYZszQvblMaTrjQCYFhJLCpocCoPNvxNbedSj01vZpUeVS1JbWcewNADdCZ1",  
  "Data": "Zbh2i96RxzrpDB0gR4p41VhFFYKwtZef7gw59x1Ie+2LBHtT70HBY9WH/  
    TKPbhmEF3epvTpyyiYkTs0AATtZSCRxQ4v+  
    kIhijb2AgBZi06v258HnI6TdwC6Uni6pTMkfyYX44Dv3ibYpY1NnILiYt6wG+HQ2/  
    XV5ptljCcToQGqzxfQkt6yJquxhnfwZSv0FK0buf79pk5m2DJQoHaRkfxmdJuJzpVU/YwQw/  
    M4b3Yr6GLYzlc3DYZw2fMU/FgBTCS3dZnvJvuMhRPNgaqy7u4s1coDEItm9j5uY+ulaNOVY/  
    y6PLITgc1Y2ciX1L8c91ZqmQiRoaxYB2+8FXs14VgdHC99shQJhptQ8RV/  
    dp7ei9Kju7R5aWKzvuDJmG0aWGo0r1X4C15NRS2qoGEpf/5tf8GbqzSqJ85khC920uTJ1nLqnEy  
    /3NTMvZ0EewSZvD52uaeDL2TvQatBb5I217UsSTivEDR2Jpu9htpoVxKeh8Woe2VaU1IV+  
    UUGIka9+MGH97aHmsRqj5Vo8C62Jw=="  
}
```

Listing C.4: Požadavek na vykonání SQL příkazu

C.4.1.4 Návrátová hodnota pro ExecuteNonQuery

Návrátová hodnota po vykonání příkazu ExecuteNonQuery nevracející záznamy.

```
{  
  "RowsAffected": "-1"  
}
```

Listing C.5: Návrátová hodnota pro ExecuteNonQuery

C.4.1.5 Návrátová hodnota pro ExecuteScalar

Návrátová hodnota po vykonání příkazu ExecuteScalar vracející první pole prvního záznamu.

```
{
  "Rows": [
    {
      "Columns": [
        {
          "Value": 1
        }
      ]
    }
  ],
  "FieldCount": 1,
  "HasRows": true
}
```

Listing C.6: Návrátová hodnota pro ExecuteScalar

C.4.1.6 Návrátová hodnota pro ExecuteReader

Návrátová hodnota po vykonání příkazu ExecuteReader vracející sadu záznamů.

```
{
  "Rows": [
    {
      "Columns": [
        {
          "Value": 1
        },
        {
          "Value": "Admin"
        },
        {
          "Value": null
        },
        {
          "Value": 1
        }
      ]
    }
  ]
}
```

```

    "Value": 1
  },
  {
    "Value": "2020-12-05T11:41:05.237"
  },
  {
    "Value": null
  },
  {
    "Value": "2021-02-26T22:02:04.05"
  },
  {
    "Value": 1
  },
  {
    "Value": false
  },
  {
    "Value": null
  },
  {
    "Value": null
  },
  {
    "Value": true
  }
]
}
],
"FieldCount": 13,
"HasRows": true
}

```

Listing C.7: Návrátová hodnota pro ExecuteReader

C.4.2 Neplatné požadavky

Ukázka komunikace pro neplatné JSON požadavky webové služby.

C.4.2.1 Požadavek na přístup s neplatným veřejným klíčem

Požadavek na přístup s neplatným veřejným klíčem.

```
{
  "Token": "",
  "PublicKey": "bflmpsvzbflmpsvz",
  "Checksum": "69PmzVNwO5Wa1zH2/Cn1CjCBHgBMwXpDZWyBMZt35WMV7pgJeTAgtCMTbzuB5F19",
  "Data": "aE+AyY1SKk79v8u3ccmwRn3C1Pd/wsw71nx2ED9t/rvIy/4
  XLpa11ZqMJbuYUV103xFS4CqGb2vsMfjT1yXAuA/
  AwqJwta01qGv5hMwwfHARISiY1KDSyT88iQh4oBETnYBwv9RilUx6xasUcb9gFwytwaGqKU83vmnFzGf1s0H5xa
  +7LmWN9+MtAo6XOPHQsn6BHVY6axleTvIDyILcyg=="
}
```

Listing C.8: Požadavek na přístup s neplatným veřejným klíčem

C.4.2.2 Návrátová hodnota pro chybný veřejný klíč

Návrátová hodnota pro chybný veřejný klíč.

```
{
  "Error": "Invalid JSON format!"
}
```

Listing C.9: Návrátová hodnota pro chybný veřejný klíč

C.4.2.3 Požadavek na přístup s chybnou definicí připojení k MSSQL serveru

Požadavek na přístup s chybnou definicí připojení k MSSQL serveru.

```
{
  "Token": "",
  "PublicKey": "7Fqa0jgyg8NkEOCO",
  "Checksum": "hEsgbYZszQvbLMAtrjQCYFhJLc pocCoPNvxNbedSj01vZpUeVS1JbWcewNADdCZ1",
  "Data": "Aa1mAXfLDroQ3Sa+qRsag7/7xd+8
  Nd3H9b9xMxLXPjJManohEGsZJsifcc22ykuy8AQM0dkaehU0dvrAHEK0awFg7Ua2xmbcco+
  x7o9cJgp+9MagrJBVxkFtGUcvqZfFF2HAh+5
  c1j5dLHEvkdX1Qo5N6NXFILAj2BwyiHdKPLKvbceBZzhmwf10mmLPmkK6"
}
```

Listing C.10: Požadavek na přístup s chybnou definicí připojení k MSSQL serveru

C.4.2.4 Návrátová hodnota pro chybnou definicí připojení k MSSQL serveru

Návrátová hodnota pro chybnou definicí připojení k MSSQL serveru.

```
{  
  "Error": "Cannot connect to the dabase!\nPři vytváření připojení k serveru SQL  
  Server došlo k chybě související se sítí nebo s instancí. Server nebyl  
  nalezen nebo nebyl přístupný. Ověřte, zda je název instance správný a zda je  
  SQL Server nakonfigurován, aby povoloval vzdálená připojení. (provider: SQL  
  Network Interfaces, error: 26 - Chyba při vyhledávání zadaného serveru či  
  instance)"  
}
```

Listing C.11: Návrátová hodnota pro chybné MSSQL připojení

Příloha D

Elektronické přílohy

D.1 Příloha v IS EDISON

Zdrojové kódy.