

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

**Návrh a implementace informačního
systému zkušební laboratoře**

Bc. Jiří Eichler

Diplomová práce

2018

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jiří Eichler**
Osobní číslo: **I16210**
Studijní program: **N2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Návrh a implementace informačního systému zkušební laboratoře**
Zadávací katedra: **Katedra softwarových technologií**

Z á s a d y p r o v y p r a c o v á n í :

Cílem práce je vytvoření informačního systému zkušební laboratoře, který bude vybudován okolo document management systému. V teoretické části je očekávána analýza potřeb zkušební laboratoře a volně dostupných document management systémů. Na základě těchto analýz bude zvolen vhodný document management systém a navržen potřebná rozšíření s ohledem na požadované normy a vnitřní předpisy zkušební laboratoře. V praktické části by měla být implementována tato vlastní rozšíření základního document management systému s ohledem na skutečné potřeby zkušební laboratoře. Informační systém by měl být implementován jako desktopová aplikace. Na závěr by měly být zhodnoceny její vlastnosti a vypracován uživatelský manuál.

Rozsah grafických prací: 10
Rozsah pracovní zprávy: 60
Forma zpracování diplomové práce: tištěná
Seznam odborné literatury:

ADÁMEK, Jiří. Kódování. SNTL - Státní nakladatelství technické literatury. 1. vyd. Praha: SNTL, 1989.

PUŽMANOVÁ R., Šmrha P.: Propojování sítí s TCP/IP. Kopp, České Budějovice, 1999.

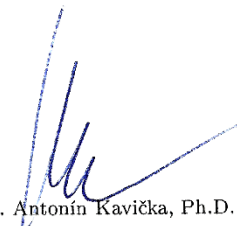
KESHAV, Srinivasan. An engineering approach to computer networking: ATM networks, the internet, and the telephone network. Reading, Mass.: Addison-Wesley, c1997. ISBN 978-0201634426.

Vedoucí diplomové práce: **doc. Ing. Tomáš Brandejský, Dr.**
Katedra softwarových technologií

Datum zadání diplomové práce: **30. října 2017**
Termín odevzdání diplomové práce: **18. května 2018**



Ing. Zdeněk Němec, Ph.D.
děkan



prof. Ing. Antonín Kavička, Ph.D.
vedoucí katedry

V Pardubicích dne 15. listopadu 2017

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 19. 8. 2018

Jiří Eichler

Poděkování

Na tomto místě bych rád poděkoval doc. Ing. Tomáši Brandejskému, Dr. za cenné rady, které mi poskytl v průběhu tvorby této práce. Dále bych rád poděkoval pracovníkům zkušební laboratoře COV FD za čas, který mi věnovali při konzultacích a vůbec za možnost tuto práci realizovat.

Anotace

Hlavním cílem této diplomové práce je návrh a implementace informačního systému pro zkušební laboratoř COV FD. Na začátku práce je popsána problematika elektronických dokument management systémů a realizováno srovnání vybraných existujících řešení. Zároveň jsou zde uvedeny legislativní požadavky na správu dokumentů vyplývající z norem a zákonů jako je GDPR. Následuje představení laboratoře COV FD a popis stávajícího stavu vedení dokumentů. Další kapitoly práce se pak již zabývají samotným návrhem a implementací. V závěru práce je uveden uživatelský manuál k vytvořené aplikaci.

Klíčová slova

informační systém zkušební laboratoře, C#, MVVM, WPF, správa dokumentů, elektronický dokument management systém, GDPR

Title

Design and implementation of information system for Certification Body

Annotation

The main aim of this thesis is the design and implementation of information system for Certification Body COV FD. At the beginning of the thesis is described the issues of electronic documents management systems and comparison of selected existing solutions. At the same time, there are the legislative requirements for document management resulting from standards and laws such as the GDPR. The following is a presentation of the COV FD Laboratory and a description of the current state of their document management. The next chapters of the thesis deal with the design and implementation itself. At the end of the thesis is a user manual for the created application.

Keywords

Information System for Certification Body, C#, MVVM, WPF, Document Management, Electronic Document Management System, GDPR

Obsah

Seznam zkratk	8
Seznam obrázků	10
Seznam tabulek	12
Úvod	13
1 Elektronický dokument management systém	14
1.1 Základní komponenty EDMS	15
1.1.1 Repozitář	15
1.1.2 Rozhraní pro přístup	15
1.1.3 Struktura složek	15
1.2 Základní vlastnosti.....	16
1.2.1 Správa verzí.....	16
1.2.2 Omezení přístupu.....	16
1.2.3 Klasifikace a indexování.....	16
1.2.4 Vyhledávání.....	16
1.2.5 Check-in a check-out.....	17
1.2.6 Auditování.....	17
2 Srovnání existujících řešení EDMS	18
2.1 Závěrečné zhodnocení	24
3 Legislativní požadavky na EDMS	25
3.1 GDPR	25
3.2 ISO 9001, ISO 14001, ISO 27001, ISO/IEC 17065	26
4 Certifikační orgán pro výrobky při Fakultě dopravní	28
4.1 Stávající stav vedení dokumentace v laboratoři COV.....	30
5 Analýza a návrh	33
5.1 Požadavky.....	33

5.2	Uživatelské role	36
5.2.1	Uživatel	36
5.2.2	Administrátor	36
6	Zvolené technologie.....	37
6.1	.NET	37
6.2	Jazyk C#.....	38
6.3	WPF & XAML.....	38
6.4	LiteDb	39
7	Implementace	41
7.1	MVVM.....	41
7.2	Model v ISCOV	42
7.3	Databáze a přístup k datům	43
7.4	ViewModel v ISCOV	48
7.5	Vrstva View v ISCOV	50
7.5.1	Navigace mezi View	51
7.5.2	Databinding.....	51
7.5.3	Konvertory.....	52
7.5.4	Vlastní komponenty	53
7.6	Bezpečnost.....	54
8	Uživatelský manuál k aplikaci ISCOV	56
8.1	Instalace.....	56
8.2	První spuštění.....	56
8.3	Přihlášení do ISCOV.....	57
8.4	Obrazovka „Dashboard“	57
8.5	Obrazovka „Moje zakázky“	59
8.6	Obrazovka „Detail zakázky“	61
8.6.1	Záložka „Detail zakázky“	61

8.6.2	Záložka „Soubory“	62
8.6.3	Záložka „Oprávnění“	68
8.6.4	Záložka „Aktivity“	69
8.7	Obrazovka „Nová zakázka“	70
8.8	Obrazovka „Lidé“	71
8.8.1	Přidání nového uživatele	72
8.8.2	Editace stávajícího uživatele.....	72
8.8.3	Odstranění uživatele.....	72
8.9	Obrazovka „Nastavení“	73
Závěr	74
9 Přílohy	76
9.1	Příloha A	76
9.2	Příloha B	77
9.3	Příloha C.....	78
9.4	Příloha D	79
10 Citovaná literatura	80

Seznam zkratek

ACID	Atomicity, Consistency, Isolation, Durability
AES	Advanced Encryption Standard
API	Application Programming Interface
BSON	Binary JavaScript Object Notation
CD	Compact disc
CLR	Common Language Runtime
COV FD	Certifikační orgán pro výrobky při Fakultě dopravní
ČSN	Česká státní norma
ČVUT	České vysoké učení technické
DB	Databáze
DLL	Dynamic-link library
DVD	Digital Video Disc
ECM	Enterprise Content Management
EDMS	Elektronický dokument management systém
EN	Evropská norma
EU	Evropská unie
FTP	File Transfer Protocol
FTPS	File Transfer Protocol Secure
GDI	Graphics Device Interface
GDPR	General Data Protection Regulation
GPL	General Public License
GUI	Graphical User Interface
HTML	Hypertext Markup Language
IEC	International Electrotechnical Commission
IOT	Internet of Things
IS	Informační systém
ISCOV	Informační systém pro Certifikační orgán pro výrobky při Fakultě dopravní
ISO	International Organization for Standardization
LINQ	Language Integrated Query

MVVM	Model-view-viewmodel
NASA	National Aeronautics and Space Administration
PHP	Hypertext Preprocessor
REQ	Requirements
REQN	Requirements non-functional
SMTP	Simple Mail Transfer Protocol
UWP	Universal Windows Platform
WCF	Windows Communication Foundation
WPF	Windows presentation foundation
XAML	Extensible Application Markup Language
XML	Extensible Markup Language

Seznam obrázků

Obr. 1 – Screen Document Managment SpringCM	18
Obr. 2 – Screen Online Document Managementu	19
Obr. 3 – Screen LogicalDOC	20
Obr. 4 - Screen OpenDocMan	21
Obr. 5 – Screen programu EasyCapsa.....	22
Obr. 6 – Screen Alfresco	23
Obr. 7 – Organizační struktura COV FD ČVUT.....	28
Obr. 8 – Architektura .NET.....	37
Obr. 9 – Ukázka vytvoření okna v jazyce XAML	39
Obr. 10 – Ukázka POJO třídy	40
Obr. 11 – Struktura návrhového vzoru MVVM	42
Obr. 12 – Schéma modelové vrstvy systému ISCOV	42
Obr. 13 – Ukázka implementace metody OnPropertyChanged.....	43
Obr. 14 – Ukázka definice pravidel pro validaci v datové třídě	43
Obr. 15 – Rozhraní pro návrhový vzor Repository	44
Obr. 16 – Ukázka přístupu k poskytovateli dat	45
Obr. 17 – Schématické zobrazení knihovny DataAccess	45
Obr. 18 – Využití referenčního odkazu mezi kolekcemi.....	45
Obr. 19 – Ukázka uchování objednávky ve formátu JSON.....	46
Obr. 20 – Ukázka vloženého dokumentu	47
Obr. 21 – Manažery v aplikaci ISCOV.....	47
Obr. 22 – Ukázka použití manažeru	47
Obr. 23 – Ukázka získání objednávek pro přihlášeného uživatele	48
Obr. 24 – Vlastnost ICommand pro navigaci na detail objednávky	49
Obr. 25 – Ukázka využití commandu ve View	49
Obr. 26 – Ukázka použití třídy Messenger pro komunikaci mezi MyOrderViewModelem a AddOrderViewModelem.....	49
Obr. 27 – Ukázka odeslání notifikací při přidání nové zakázky	50
Obr. 28 – View v aplikaci ISCOV.....	50
Obr. 29 – Definování data template v XAML.....	51
Obr. 30 – Schématické zobrazení třídy Navigator	51

Obr. 31 – Ukázka databindingu mezi View a ViewModelem	52
Obr. 32 – Ukázka použití konvertoru	52
Obr. 33 – Ukázka využití multibindingu	53
Obr. 34 – Ukázka využití vlastní komponenty	54
Obr. 35 – Ukázka použití <code>PrincipalPermissinAttribute</code>	54
Obr. 36 – Obrazovka při prvním přihlášení do ISCOV	56
Obr. 37 – Přihlašovací obrazovka aplikace ISCOV	57
Obr. 38 – Obrazovka "Dashboard"	58
Obr. 39 – Obrazovka "Detail zprávy"	58
Obr. 40 – Obrazovka "Moje zakázky"	59
Obr. 41 – Ukázka "Editace zakázky"	59
Obr. 42 – Ukázka výběru data	60
Obr. 43 – Obrazovka "Detail zakázky"	61
Obr. 44 – Obrazovka "Soubory"	62
Obr. 45 – Ukázka výběru souborů	63
Obr. 46 – Ukázka souborů přidávaných k zakázce	63
Obr. 47 – Ukázka nahrávání souboru	64
Obr. 48 – Ukázka náhledu souboru	65
Obr. 49 – Ukázka správy verzí	66
Obr. 50 – Ukázka nastavení oprávnění	67
Obr. 51 – Ukázka editace existujících oprávnění k souboru	67
Obr. 52 – Obrazovka "Oprávnění"	68
Obr. 53 – Ukázka nastavení oprávnění pro zakázku	69
Obr. 54 – Záložka "Aktivity"	70
Obr. 55 – Obrazovka "Nová zakázka"	70
Obr. 56 – Obrazovka "Lidé"	72
Obr. 57 – Obrazovka "Nastavení"	73

Seznam tabulek

Tab. 1 – Porovnání zpracování listinných a elektronických dokumentů	14
Tab. 2 – Srovnání EDMS.....	24
Tab. 3 – Funkční požadavky kladené na navrhovaný systém.....	33
Tab. 4 – Nefunkční požadavky kladené na navrhovaný systém	35

Úvod

Potřeba uchovávat, prohlížet a spravovat různé záznamy sahá na úplný počátek lidské civilizace. Již pravěcí lidé kreslili obrazy a různé symboly na stěny svých jeskyní či je vyrývali do kamene ve snaze uchovat události své doby. S tím, jak plynul čas, začala vzrůstat i potřeba uchovávat čím dál více záznamů. Ty staré bylo nutné archivovat a nahrazovat je novými aktuálními, díky čemuž začali vznikat první dokument management systémy.

S příchodem prvních počítačů se záznamy začali uchovávat elektronicky, ale potřeba jejich správy tu byla stále, a to v ještě větším měřítku. Téměř každá firma začala budovat informační systémy pro jejich správu, a jako součást takových systémů, ale i samostatně, začali vznikat specializované programy určené pro správu dokumentů nazývané elektronické dokument management systémy (EDMS).

Cílem této práce je tedy implementovat informační systém pro Certifikační orgán pro výroby při Fakultě dopravní ČVUT. Tento systém by měl být vybudován, pokud to bude možné, okolo existujícího elektronického dokument management systému. V teoretické části práce bude tedy provedena analýza požadavků zkušební laboratoře a srovnání existujících dokument management systémů. V práci budou též uvedeny legislativní požadavky na EDMS. V praktické části bude následně provedena implementace vlastního rozšíření vybraného vhodného dokument management systému, případně implementace nového řešení s ohledem na skutečné potřeby zkušební laboratoře. Vyvíjený informační systém by měl být implementován jako desktopová aplikace spustitelná na operačním systému Windows. Součástí práce bude též uživatelský manuál.

1 Elektronický dokument management systém

Elektronický dokument management systém (EDMS) je systém starající se o centrální ukládání správu a organizaci různých typů dokumentů. Mezi hlavní důvody využívání těchto systému patří kvalitní archivace nebo schvalování různých typů dokumentů, mnohonásobně rychlejší vyhledávání či lepší ochrana před možným zneužitím a neautorizovaným přístupem. Další nespornou výhodou zavedení elektronického dokument management systému je okamžitá dostupnost aktuálních dat s čím jde ruku v ruce i zvýšení produktivity zaměstnanců, ale i nižší finanční a časové nároky oproti klasické údržbě archivu. Pokročilejší EDMS nabízí například i podporu workflow pro automatizaci celého podnikového procesu, systémy pro spisovou službu a další. Srovnání zpracování elektronických a listinných dokumentů je uvedeno v Tab. 1. [1]

Tab. 1 – Porovnání zpracování listinných a elektronických dokumentů
[Převzato z [1]]

	Listinné dokumenty	Elektronické dokumenty
Náklady na pořízení	nízké	vysoké
Možnost ztráty	vysoká	minimalizována
Neproduktivní náklady (vyhledávání, manipulace)	vysoké	minimalizovány
Dostupnost dokumentu v daném okamžiku	jediný pracovník	kdokoli, kdo má právo
Ověřitelnost platnosti dokumentu	neověřitelné (vedle originálu existují nevidované kopie)	ověřitelné (jediný výskyt)
Náklady na archivaci	vysoké (skříňné, místnost, klimatizace, pracovníci)	minimální (paměťová média)

EDMS pro svou práci potřebuje základní komponenty jako je repozitář, rozhraní pro přístup a možnost třídění dokumentů viz kapitola 1.1. Zároveň musí disponovat vlastnostmi jako je správa verzí, omezení přístupu, vyhledávání, auditování, check-in & check-out, klasifikace a indexování. Těmto vlastnostem je věnována kapitola 1.2.

1.1 Základní komponenty EDMS

1.1.1 Repozitář

Všechny EDMS vyžadují nějaké uložení, repozitář, který umožní permanentní uchování dokumentů. Repozitář je tvořen s použitím dvou vrstev. První vrstva představuje archív s daným logickým uspořádáním dokumentů. Druhá, fyzická vrstva je pak tvořena fyzickým uložetím dokumentů. Toto uložetě, respektive repozitář může být na lokálním či síťovém disku nebo se může jednat o vzdálený server. Stejně tak je možné mít uložetě v jednom umístění nebo distribuované na několika serverech. [1]

Jako repozitář mohou sloužit i různé typy databází. Může se jednat např. o full-textové databáze, resp. dokumentově orientované DB, které jsou k tomuto účelu více než vhodné.

At' už je repozitář realizován jakoukoli technologií je důležité, aby k němu existoval jeden centralizovaný přístup. [2]

1.1.2 Rozhraní pro přístup

Jeho hlavním účelem je zajistit uživatelům přístup k systému, umožnit nahrávání dokumentů do repozitáře, monitorovat jeho změny atd. Rozhraní pro přístup může mít různé podoby. Jedná se především o desktopové, webové a případně dnes čím dál oblíbenější mobilní rozhraní. To může ale mít i podobu různých doplňků pro aplikace. [2]

1.1.3 Struktura složek

Umožňuje organizovat dokumenty, respektive fyzické soubory do složek. Složková struktura může být organizována například, dle typu dokumentu, dle názvů projektů, dle vlastníka souboru, případně se může jednat o kombinaci přístupů. U pokročilejších EDMS je hierarchie složek nastavitelná. [2]

1.2 Základní vlastnosti

1.2.1 Správa verzí

Jde o základní vlastnost EDMS, která umožňuje sledovat změny dokumentu. Umožňuje návrat k předchozí verzi, přiděluje čísla revizí a identifikátory dokumentům. [2]

1.2.2 Omezení přístupu

Omezení přístupu je extrémně důležitá vlastnost systému, poskytující bezpečnostní přístup k dokumentům, a to na různých úrovních. Jako příklad je možné uvést to, že administrátor systému může nastavovat přístup k souborům v repozitáři individuálně pro jednotlivé uživatele či jejich skupiny, a to s různou úrovní přístupu. Jeden uživatel tedy může dokumenty pouze číst, přičemž jiný může i nahrávat jejich revize. [2]

1.2.3 Klasifikace a indexování

Všechny dokumenty v repozitáři jsou klasifikovány na základě metadat, která později slouží pro snazší vyhledávání, filtrování případně i třídění. Metadata obsahují různé informace o dokumentu jako je například autor, titulek, datum a čas vytvoření, respektive úpravy a další. Tyto údaje mohou být rozšířeny například i o informaci jaké oddělení dokument připravilo, jaká je to verze, kdo se spolupodílel na dokumentu atd. [2]

1.2.4 Vyhledávání

Systém by měl disponovat více cestami, pomocí níž lze dokumenty dohledat. Může se jednat o procházení souborové struktury, struktury dle zakázek nebo objednávek, případně pomocí jednoduchého anebo pokročilého vyhledávání.

Jednoduché vyhledávání prochází všechny dostupné dokumenty (například včetně obsahu) a jejich metadat a hledá shodu se zadanými klíčovými slovy. Pokročilé vyhledávání umožňuje uživateli hledat dle specifických polí metadat, obsahu či jejich kombinací. Může být umožněno využívat logické operátory AND a OR k vyspecifikování požadovaného dotazu. Příkladem pokročilého vyhledávání může být vyhledání všech dokumentů jejich autorem je Jan Novák mezi roky 2016 a 2018. [2]

1.2.5 Check-in a check-out

Jde o princip, respektive vlastnost EDMS, díky níž je možné kontrolovat, pokud někdo editoval, respektive edituje dokument a zabránit tak situacím, kdy by se jedna revize dokumentu přehrála „starší“ verzí anebo kdy by se pokoušelo o úpravu daného dokumentu více uživatelů. Příkladem této situace může být to, když jeden uživatel edituje dokument a jiný uživatel by se pokusil nahrát svoji revizi. Toto mu nebude umožněno, protože dokument je v této chvíli pro všechny dostupný jen na úrovni čtení. Ve chvíli, kdy uživatel provádí úpravy na dokumentu uloží svojí práci jako revizi do repozitáře je umožněno ostatním uživatelům majícím dostatečná oprávnění dokument editovat, případně nahrát svoji revizi. [2]

1.2.6 Auditování

Auditování spolu se správou verzí kontroluje, který uživatel a kdy provedl změny, v daném dokumentu, případně s ním jinak nakládal. Jinak řečeno auditování dovoluje autorizovaným osobám zjistit, co všechno bylo s dokumentem prováděno (kdo, kdy si dokument prohlížel, stahoval, nahrál novou revizi a podobně) od chvíle kdy se poprvé objevil v systému. Díky této vlastnosti je například možné nalézt viníka chyby v dokumentaci, případně zjistit kdy k ní došlo a podobně. [2]

2 Srovnání existujících řešení EDMS

V této kapitole bude provedeno srovnání šesti existujících elektronických dokument management systémů. Při jejich výběru byl kladen důraz na to, aby byly zastoupeny produkty napříč nabízeným spektrem. Do srovnání tedy byly zařazeny jak velké placené, tak i menší či free nebo open source varianty elektronických dokument management systémů.

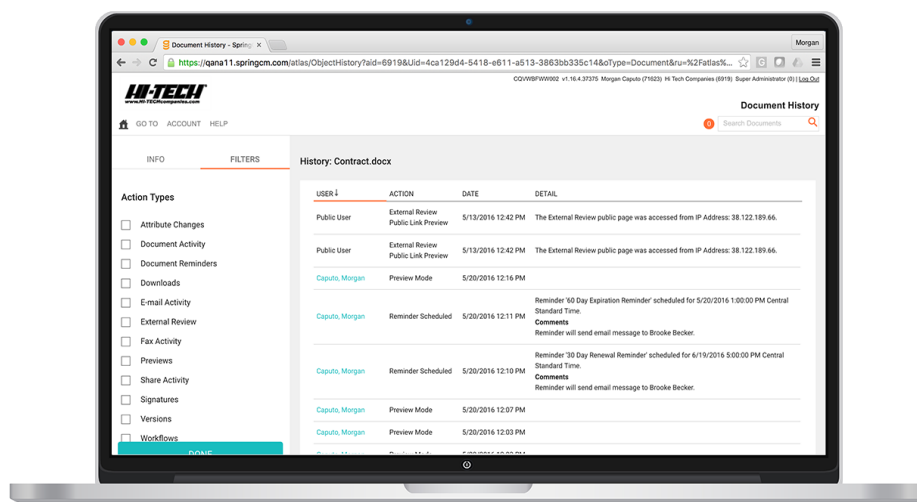
Document Management SpringCM

Web: <http://www.springcm.com/>

Document Management SpringCM je dokument management systém od Společnost SpringCM. Představuje řešení pro správu dokumentů dostupné online.

Mimo klasických funkcí jako organizování dokumentů, propojení s workflow firmy v grafickém rozhraní, verzování, nastavení práv přístupu a třídění do složek umožňuje i generování odkazů na sdílení s externími spolupracovníky online a správu komentářů. Nechybí ani možnost pokročilého vyhledávání dokumentů, a to jako pomocí názvů tak fulltextově. Ke každému dokumentu je možné navíc připojit tagy, dle kterých je posléze možné dokumenty filtrovat nebo hledat. [3] [4]

Produkt je komerční, přičemž základní cena činí 29\$ za měsíc za uživatele. K dispozici je zdarma ukázkové demo dostupné po registraci na webu výrobce. [3]



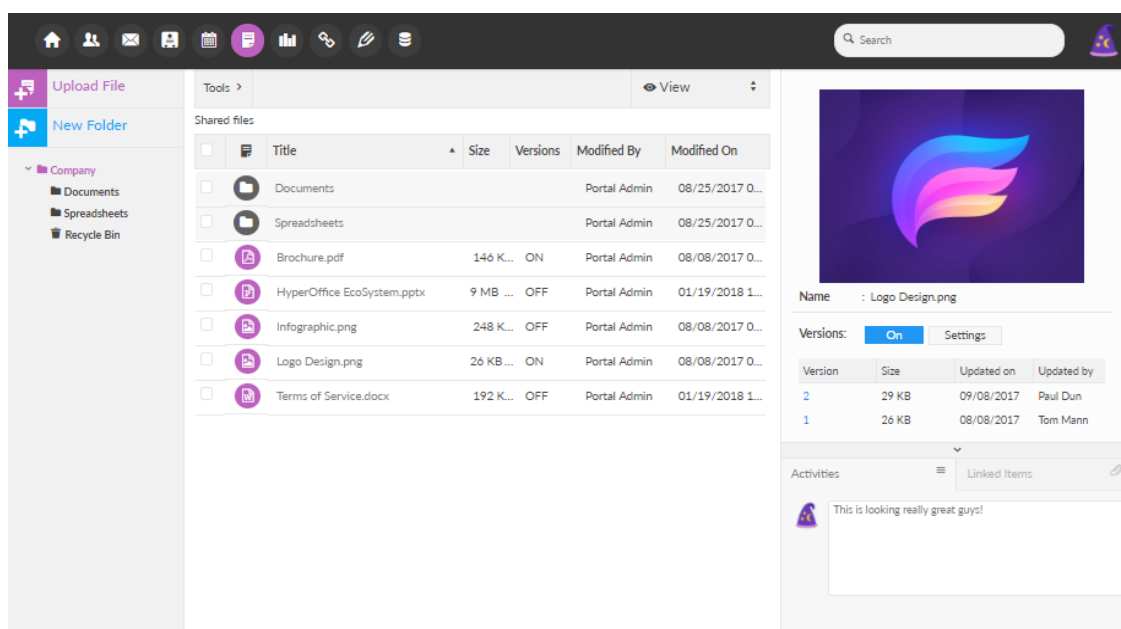
Obr. 1 – Screen Document Management SpringCM
[Zdroj: <https://www.springcm.com/products/document-management>]

Online Document Management

Web: <http://www.hyperoffice.com/>

Výrobcem tohoto řešení je společnost HyperOffice. Tento systém disponuje online prostředím pro komplexní správu dokumentů. Umožňuje sdílení dokumentů, a to jak uvnitř firmy, tak i s klienty nebo partnery online. Podporuje workflow dokumentů ve firmě a uchovávání historie verzí. Každému uživateli lze nastavit práva přístupu k dokumentům (prohlížení, editace, mazání nebo nedostupný). K dispozici je i fulltextové vyhledávání nebo vedení konverzací u jednotlivých dokumentů. Předností toho řešení je možnost vytvářet vlastní kontext okolo dokumentů a složek s možností tento kontext sdílet namísto sdílení pouhého dokumentu. Jako kontext je možné si představit uživatelsky upravenou webovou stránku s odkazy. [5] [4]

Produkt je komerční, přičemž jeho cena je stanovena individuálně. K dispozici je 15denní zkušební verze nebo 15minutové demo. Document management je součástí produktu Atlas Collaboration Suite. Dle webu výrobce jej používá přes 2 miliony lidí ve společnostech jako je například NASA nebo Foxconn. [5]



Obr. 2 – Screen Online Document Managementu
[Zdroj: <https://www.hyperoffice.com/document-management-solution/>]

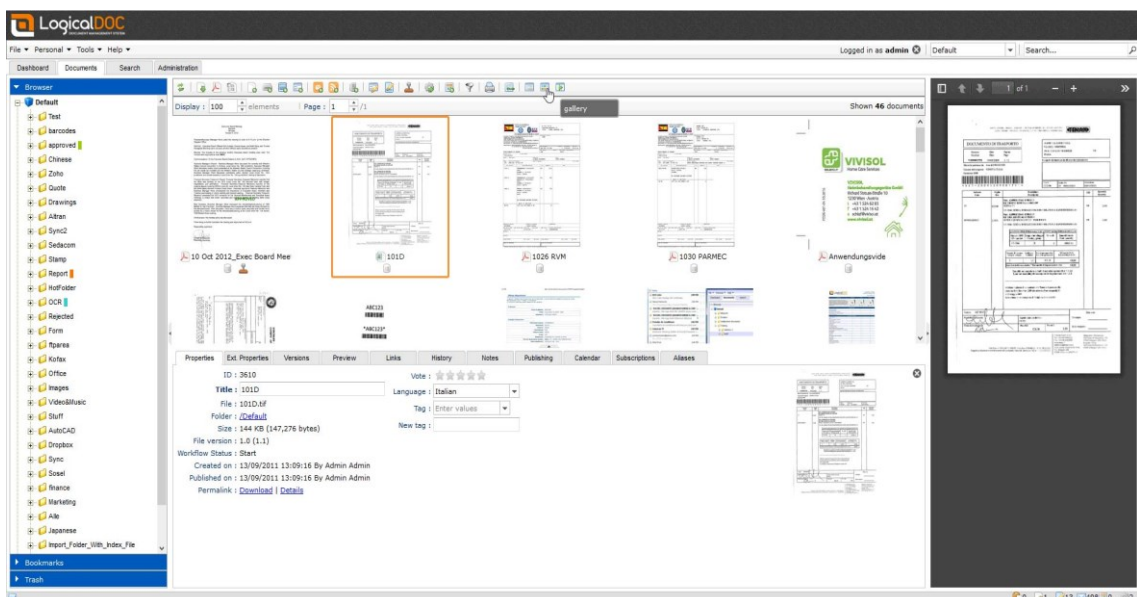
LogicalDOC

Web: <https://www.logicaldoc.com/>

LogicalDOC je softwarový produkt pro správu dokumentů od stejnojmenné společnosti. Jedná se o webové řešení, umožňující vkládání, uchování, sdílení dokumentů, jejich řazení do složek, fulltextové vyhledávání, a nechybí ani správa verzí. Systém disponuje vizuálním nástrojem pro tvorbu workflow, který umožňuje přiřadit dokumenty ke každému jednomu kroku business procesu. K dispozici je i doplněk pro Microsoft Office umožňující přímou práci s dokumenty. Součástí LogicalDOC je i správce agendy v podobě kalendáře. Tento kalendář pomáhá uživatelům sledovat termíny a události spojené s konkrétním dokumentem a umožňuje zasílat notifikace na tyto události emailem. [6]

Řešení LogicalDOC je možné nainstalovat na vlastní server s operačním systémem Windows, Linux či OS X nebo využít verzi Enterprise & Cloud. K dispozici je i rozsáhlé API, které zpřístupňuje funkce pro externí software. [6]

Tento produkt je k dispozici v několika verzích lišící se funkcionalitou. Jedná se o verze Community, Business a Enterprise & Cloud, přičemž verze Community je zdarma, zbylé dvě verze je možné otestovat v 30denní trial verzi. Ceny vyšších verzí se stanovují individuálně jako tomu bylo u předchozího produktu. [6]



Obr. 3 – Screen LogicalDOC

[Zdroj: <https://cdn.logicaldoc.com/images/assets/overview/gallery.jpg>]

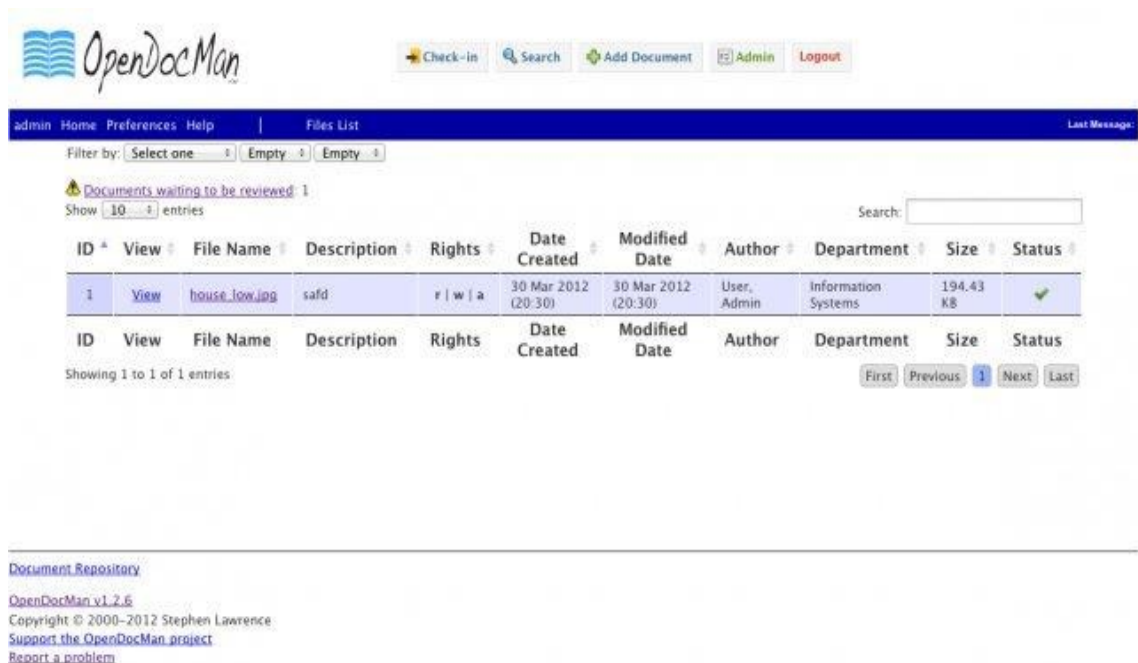
OpenDocMan

Web: <http://www.opendocman.com/>

OpenDocMan je vyvíjen jako webový, open source dokument management systém pod licencí GPL kompletně napsaný v PHP. [7]

Obsahuje téměř stejné základní funkce jako placené systémy. Centralizované uložení, možnost přidělování práv přístupu k dokumentům jednotlivým uživatelům, možnost správy revizí, přidávání komentářů nechybí ani možnost vyhledávání, a to, především pomocí metadat jako je název nebo kategorie dokumentu. Systém disponuje i možností tvorby jednoduchého workflow s možností nastavení expirace dokumentu nebo zasílání emailových upozornění před a po kontrole. [7] [4]

Tento systém je možné provozovat na vlastním serveru nebo použít placený hosting, který ve variantě Basic stojí \$24.95 za měsíc a je optimalizován pro provozování systému OpenDocMan. [7]



The screenshot displays the OpenDocMan web interface. At the top, there is a navigation bar with the OpenDocMan logo and several buttons: Check-in, Search, Add Document, Admin, and Logout. Below this is a secondary navigation bar with links for admin, Home, Preferences, Help, and Files List. The main content area shows a filter section with 'Filter by: Select one', 'Empty', and 'Empty'. A notification indicates 'Documents waiting to be reviewed: 1'. Below this is a table with columns: ID, View, File Name, Description, Rights, Date Created, Modified Date, Author, Department, Size, and Status. The table contains one entry with ID 1, File Name 'house_low.jpg', Description 'safd', Rights 'r | w | a', Date Created '30 Mar 2012 (20:30)', Modified Date '30 Mar 2012 (20:30)', Author 'User, Admin', Department 'Information Systems', Size '194.43 KB', and Status '✓'. At the bottom of the table, it says 'Showing 1 to 1 of 1 entries' and there are navigation buttons: First, Previous, 1, Next, Last. Below the table, there is a footer section with links for Document Repository, OpenDocMan v1.2.6, Copyright © 2000-2012 Stephen Lawrence, Support the OpenDocMan project, and Report a problem.

Obr. 4 - Screen OpenDocMan

[Zdroj: <https://sourceforge.net/p/opendocman/screenshot/324040.jpg>]

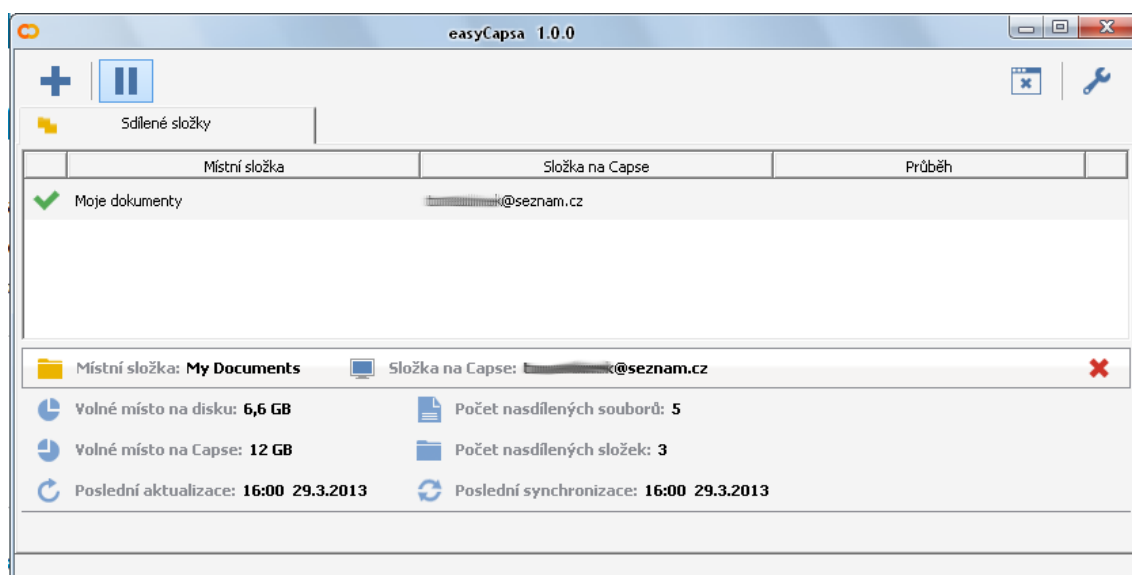
Capsa

Web: <http://www.capsa.cz/>

Capsa je jednoduchý online dokument management systém od českých tvůrců. Poskytuje základní požadavky jako je vkládání, správu a sdílení dokumentů, a to jak ve firmě, tak s externími spolupracovníky případně klienty, organizaci do složek, možnost omezení práva přístupu pro konkrétní uživatele, nebo nastavení uživatelských práv k jednotlivým dokumentům. Capsa také disponuje možností upravovat některé typy souborů online bez nutnosti stahovat je do lokálního počítače. Podporována je také správa verzí a fulltextový vyhledávač. Systém disponuje správou kontaktů nebo sdíleným kalendářem. Pro přístup k dokumentům je možné využít protokol FTP anebo si připojit sdílenou složku jako síťový disk. [8]
[4]

Mimo to výrobce poskytuje program easyCapsa, který slouží pro synchronizaci složek mezi počítačem a účtem Capsa. Hlavní stránka programu je na Obr. 5. [8]

Tento produkt je nabízen ve verzi zdarma s 5 GB prostoru a 5 uživateli. Případně ve verzích Sejf za 250Kč/měsíc, Profi za 550Kč/měsíc, Luxus za 2000Kč/měsíc a Enterprise za 7500Kč/měsíc. [8]



Obr. 5 – Screen programu EasyCapsa
[Zdroj: <http://www.capsa.cz/aplikace>]

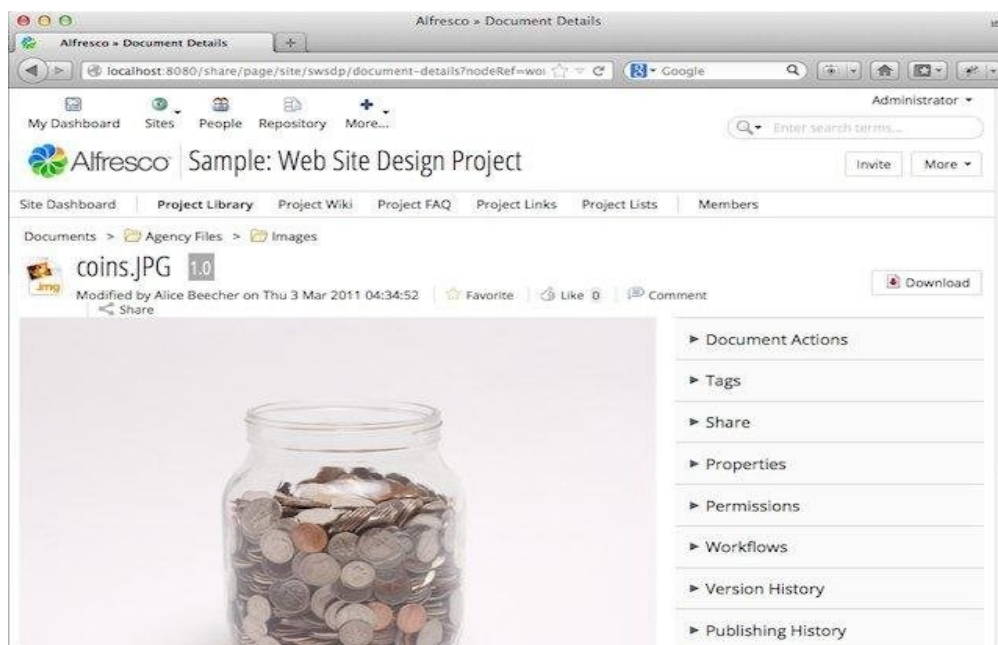
Alfresco

Web: <https://www.alfresco.com>

Produkt Alfresco poskytuje integrované řešení pro správu dokumentů a obsahu. Nabízí celou řadu funkcí a nástrojů, které usnadňují vyhledávání a přístup k dokumentům. Stejně jako ostatní produkty poskytuje základní funkce elektronického dokument management systému jako jsou vkládání, úprava a sdílení. Podporuje přidávání komentářů k dokumentům, informační kanály a seznamy úkolů. Nechybí ani možnost správy revizí a nastavení oprávnění pro jednotlivé osoby. Díky tomu že se jedná o online řešení, které je optimalizováno i na mobilní platformy je možné se do systému přihlásit i např. z mobilního telefonu. [9]

Alfresco je k dispozici pro Windows, MacOS a Linux s rozsáhlým API pro přístup z externích aplikací. Tento systém je možné provozovat na vlastním JAVA aplikačním serveru (např.: Tomcat) a databázi MySQL. [9]

Alfresco je k dispozici ve verzi Community zdarma jako součást Enterprise Content Management (ECM). [9]



Obr. 6 – Screen Alfresco

[Zdroj: <https://a.fsdn.com/con/app/proj/alfresco/screenshots/332795.jpg/max/max/1>]

2.1 Závěrečné zhodnocení

Všechny srovnávané produkty obsahují základní komponenty a splňují hlavní požadavky na elektronický dokument management systém. To znamená, že podporují správu verzí, disponují možností nastavení oprávnění pro uživatele, umožňují určitým způsobem vyhledávat a zaznamenávají aktivity prováděné s dokumentem atd. Liší se však cenou, kvalitou podpory, zabezpečením a množstvím nadstandardních vlastností.

Do následující srovnávací tabulky (Tab. 2) byly tedy vybrány spíše nadstandardní vlastnosti, kterými se jednotlivé produkty odlišují. V případě, že je produkt distribuován ve více verzích je ve srovnání zařazena verze nejnižší bez použití přídatných modulů.

Tab. 2 – Srovnání EDMS

	Online Document Management	LogicalDOC	Document Management	OpenDocMan	Capsa	Alfresco
Cena	individuálně	FREE / open source	29\$/měsíc/ uživatel	Open Source	FREE	FREE/ open source
Rozhraní	web	web	web	web	web	web
API	ANO	ANO	ANO	ANO	NE	ANO
Možnost modifikace	NE	ANO	NE	ANO	NE	ANO
Plnohodnotná desktopová aplikace	NE	NE	NE	NE	NE	NE
OCR	NE	NE	ANO	NE	NE	NE
Online editace dokumentu	NE	NE	NE	NE	ANO	ANO
Šifrování dat	ANO	NE	ANO	NE	NE	NE
Log	ANO	ANO	ANO	ANO	ANO	ANO
FTP	NE	NE	NE	NE	ANO	NE
Možnost interních zpráv	ANO	ANO	NE	NE	NE	ANO
Komentář k dokumentu	ANO	ANO	ANO	ANO	ANO	ANO

3 Legislativní požadavky na EDMS

3.1 GDPR

Celým názvem Nařízení Evropského parlamentu a Rady (EU) 2016/679 ze dne 27. dubna 2016 o ochraně fyzických osob v souvislosti se zpracováním osobních údajů a o volném pohybu těchto údajů a o zrušení směrnice 95/46/ES (obecné nařízení o ochraně osobních údajů). V textu dále bude použita zkratka GDPR (z anglického názvu General Data Protection Regulation). [10]

GDPR si klade za cíl zvýšit ochranu osobních dat občanů evropské unie. Stanovuje pravidla pro zpracování osobních dat a pravidla pro jejich pohyb. Toto nařízení se vztahuje jak na automatizované, tak na neautomatizované zpracování. GDPR vyžaduje, aby pro shromažďování osobních údajů zaměstnanců nebo zákazníků existoval nějaký zákonný důvod například pracovní či kupní smlouva, případně zákon. V případě, že žádný takovýto důvod neexistuje je nutné osobu, o níž jsou osobní údaje shromažďovány, požádat o souhlas. Tento souhlas nesmí být ničím podmiňován a musí být udělen dobrovolně. Údaje, jež jsou shromažďovány by měly mít co nejmenší rozsah a neměly by být používány v rozporu s původním cílem. [11] [10]

Již při získávání osobních údajů je nutné osobě, od níž jsou údaje vyžadovány poskytnout informace o tom, jak s těmito údaji bude nakládáno, případně komu dalšímu budou předány. Další věcí je, že dotyčný člověk může kdykoli požádat o poskytnutí všech o něm shromažďovaných údajů, případně požadovat jejich vymazání či úpravu. V takové případě však již nesmí existovat žádný zákonný důvod pro jejich další uchování. [11] [10]

Všechny získané údaje nebo dokumenty je zapotřebí chránit před neoprávněným přístupem. V případě tištěných dokumentů například uzamykatelnou skříní nebo alespoň zásuvkou stolu v uzamčené místnosti. Dokumenty uchovávané elektronicky je pak potřeba zajistit tak, aby k nim měl přístup pouze pověřený pracovník po zadání svého hesla či za použití jiné technologie autentizace. Dalším bezpečnostním prvkem je pořizování elektronický záznamů o přístupu k danému dokumentu tzv.

„logy“. Tyto logy obsahují informace, jak bylo s daným dokumentem nakládáno, jakým uživatelem a kdy. [10] [11]

Při práci s osobními údaji mimo zabezpečené prostředí informačního systému je za potřebí dbát na jejich ochranu. Citlivé dokumenty obsahující osobní údaje by tedy neměly být zasílány pomocí emailů či různých chatovacích služeb. Tyto služby rozhodně nelze považovat za bezpečný kanál a při jejich využití je více než vhodné opatřit tyto dokumenty či zprávy šifrováním. Pokud jsou údaje poskytovány někomu jinému mimo organizaci, aby s nimi pracoval na místo subjektu, kterému byl udělen souhlas je nutné uzavřít smlouvu, ve které se zaváže k ochraně těchto údajů. Bez této smlouvy by bylo nutné získat opět souhlas zainteresovaných osob. [11] [12]

Doba, po kterou je možné záznamy uchovávat je různá, a to od několika dnů, až po několik desítek let. Zásadou však je, aby údaje byli uchováni vždy po dobu co nejkratší. Tato doba může končit společně s koncem činnosti (naplnění předmětu smlouvy, ukončení pracovního poměru) nebo může být dána zákonem. [10] [11]

3.2 ISO 9001, ISO 14001, ISO 27001, ISO/IEC 17065

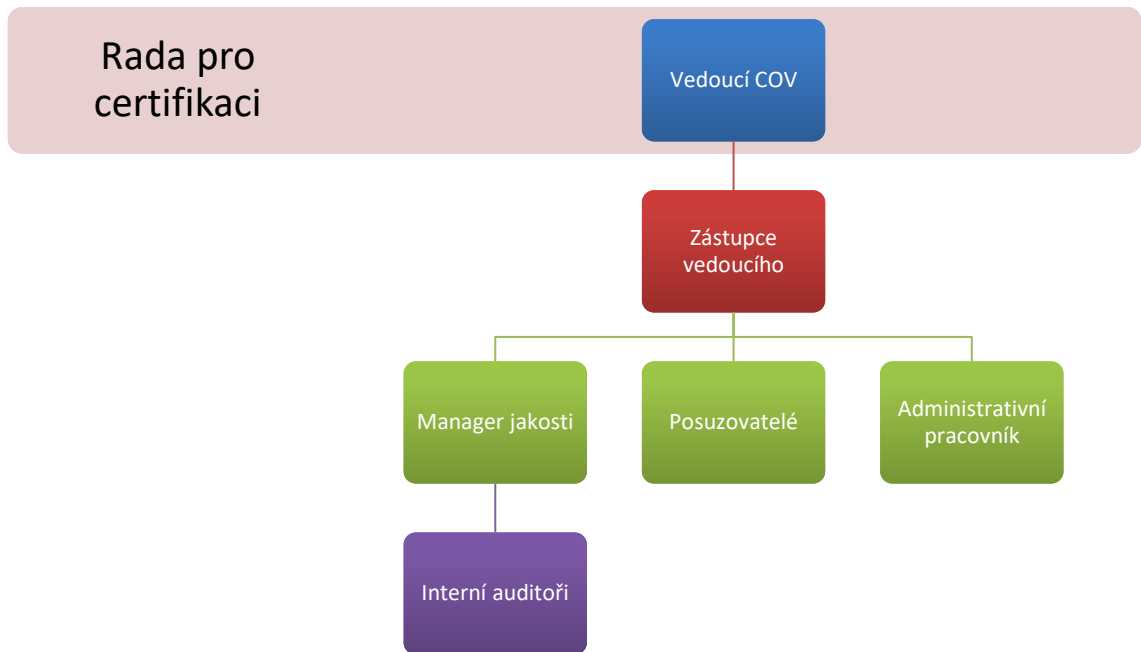
Normy ISO 9001, ISO 14001, ISO 27001, ČSN EN ISO/IEC 17065 a další obecně definují následující požadavky na vytvoření dokumentovaného postupu řízení dokumentů pro [13] [14] [15] [16]:

- schvalování dokumentů před jejich vydáním,
- přezkoumání dokumentů případně jejich aktualizaci pro opakované schvalování,
- zajištění identifikace změn a aktuálního stavu revizí v dokumentech,
- zajištění dostupnosti aktuálních verzí dokumentů v místech jejich používání,
- zajištění čitelnosti a identifikovatelnosti dokumentu,
- zajištění identifikace dokumentů, a to i dokumentů externího původu např.: číslem, názvem atd.,
- zabránění neúmyslného použití starší verze dokumentu,

- zajištění důvěryhodnosti (tj. že pouze určité osoby mají přístup k dokumentům) a integrity dokumentů (tj. že pouze někteří mohou dokumenty měnit),
- uchování a likvidaci dokumentů.

4 Certifikační orgán pro výrobky při Fakultě dopravní

Certifikační orgán pro výrobky při Fakultě dopravní (COV FD) se zabývá prováděním certifikace výrobků pro oblast železniční zabezpečovací techniky. Tento certifikační orgán spadá pod Fakultu dopravní Českého vysokého učení technického, která se dlouhodobě zabývá problematikou bezpečnosti a spolehlivosti řídicích systémů zejména v aplikaci na železniční dopravu.



Obr. 7 – Organizační struktura COV FD ČVUT

V rámci své činnosti přijímá COV FD objednávky certifikačních služeb, a to jak od tuzemských, tak od zahraničních firem. Objednávky jsou zasílány přímo vedoucímu COV, a to elektronicky pomocí emailu nebo poštou v tištěné, případně elektronické podobě na CD/DVD. Vzhledem k citlivosti některých informací mnozí zákazníci využívají právě druhou možnost a zasílají dokumenty v elektronické podobě na CD/DVD nosičích. V rámci zpracování objednávky jsou tyto dokumenty nakopírovány na cloudové úložiště bez přístupu k internetu, aby se zabránilo možnému uniku dat. Jako další krok určí vedoucí COV FD posuzovatele (vedoucího zakázky), který odpovídá za řešení zakázky a předá mu potřebné pracovní dokumenty. Zodpovědný posuzovatel může v případě potřeby přizvat další posuzovatele. Hlavním úkolem posuzovatele je pak přezkoumat objednávku (zakázku) po odborné stránce.

Provede především následující kroky [17]:

1. ověří, zda informace o žadateli a požadavky na certifikaci jsou nějakým způsobem stanoveny, dokumentovány a pochopeny,
2. prověří, zda jsou vyřešena případná nedorozumění mezi COV a zákazníkem, včetně dohody ohledně norem nebo jiných normativních dokumentů,
3. prověří, zda Certifikační orgán má k dispozici prostředky pro hodnocení a je schopen poskytnout službu odpovídajícího rozsahu požadované certifikace a zvláštním požadavkům a specifičností zákazníka,
4. na základě objednávky zákazníka stanovit, jaké služby budou v souvislosti s výrobkem provedeny a dále stanovit rozsah požadované dokumentace a počet vzorků nezbytných pro provedení požadované služby. Toto se provádí v souladu s požadavky zákonů, nařízení vlády a technických norem,
5. přípravu plánu svých hodnotitelských činností včetně rozhodnutí záměru využití příslušných subdodavatelů.

Veškeré předchozí kroky jsou vedoucím zakázky zadokumentovány a zaneseny do formuláře, který je předán ke schválení vedoucímu COV FD. Tento dokument se posléze stane součástí dokumentace k zakázce.

Následně vypracuje vedoucí COV FD kalkulaci k zakázce a společně s objednávkou ji předá administrativnímu pracovníkovi, který zakázku založí. To spočívá v přípravě podkladů pro smlouvu. Hotový návrh smlouvy se předává v tištěné podobě zpět vedoucímu COV FD a děkanovi fakulty, kteří ho parafují.

Po podpisu smlouvy zákazníkem, posuzovatel řeší zakázku podle konkrétního plánu hodnotitelských činností. Výstupem hodnotitelské činnosti posuzovatele je Hodnotící zpráva, která obsahuje zjištěné shody či neshody, tato zpráva se předává zadavateli, který se k výsledku může vyjádřit případně předloží příslušná opatření vedoucí k nápravě. Po vyřešení všech neshod a nedostatků ze strany zákazníka vyhotoví zodpovědný posuzovatel Výslednou hodnotící zprávu, kterou předá společně s veškerou dokumentací vedoucímu COV FD společně s návrhem certifikátu. Vedoucí nebo zástupce COV FD provede přezkoumání úplnosti podkladů podle smlouvy, zkontroluje shodu norem, podle kterých posouzení probíhalo, provede také kontrolu správnosti používání a aplikace ostatních dokumentů

používaných při certifikaci výrobků a kompletnost zakázky. Následně rozhodne na základě výsledků hodnocení o udělení certifikátu, vrácení zakázky k přepracování nebo o neudělení certifikátu. Na závěr připraví administrativní pracovník podklady pro fakturaci.

Dokumenty, které nejsou potřebné pro dokazování zakázky jsou skartovány, přičemž ostatní dokumentace je archivována po dobu deseti let.

Jak je z výše uvedeného patrné COV FD pracuje s množstvím rozličných dokumentů. Na jedné straně se jedná o objednávku, žádost, potvrzenou smlouvu, zkušební protokol, soubor hodnotících zpráv včetně výsledné zprávy, návrh certifikátu a další administrativně související dokumenty, přičemž na straně druhé je to dokumentace schvalovaného systému včetně jeho výkresů, zdrojových kódů, textů apod.

4.1 Stávající stav vedení dokumentace v laboratoři COV

V současné době jsou dokumenty potřebné pro chod laboratoře uchovávány buď jen v elektronické, nebo jen v tištěné, nebo v elektronické i v tištěné podobě. Elektronické verze dokumentů jsou ukládány na cloudové uložení umístěné v interní síti laboratoře. Na tomto uložení je vytvořen adresář „COVFD“, který obsahuje podadresář „rizena_dokumentace“, ve kterém jsou vytvořeny další složky, do kterých jsou dokumenty tříděny. [18]

Práva pro přístup k jednotlivým elektronickým dokumentům jsou řešena pomocí přidělení potřebného oprávnění. Tato oprávnění jsou pracovníkům přidělována vedoucím COV FD a to tak, že všichni členové COV FD mají právo pro čtení a vybraní pracovníci také práva na změny.

Seznam všech dokumentů je veden pouze v elektronické podobě v jednoduché Access databázi. Tato databáze obsahuje informace o aktuální verzi, názvu, typu (interní/externí), datu vydání a informaci o způsobu uložení (elektronicky/tištěno). Soubor této databáze je uložen stejně jako ostatní dokumenty na cloud uložení v adresáři „rizena_dokumentace“.

Na aktuální verze elektronických forem dokumentů odkazují z databáze hypertextové odkazy.

Seznam dokumentů obsahuje pro každý dokument tabulku předchozích verzí dokumentu spolu s informací, kdo změny (vydání verzí) provedl a kdy. Starší elektronické verze jsou přístupné opět pomocí hypertextových odkazů. V seznamu je uložena také tabulka s výčtem případných vydaných kopií dokumentu a jejich distribuci jednotlivým osobám.

Tištěné verze dokumentů jsou uloženy v uzamykatelné skříni v označených deskách. Všichni pracovníci COV FD mají k těmto dokumentům přístup, ale pouze za doprovodu vedoucího COV FD, zástupce nebo manažera jakosti COV FD.

Všechny dokumenty podléhají následujícím pravidlům [18] :

- I. Schválená vydání příslušných dokumentů jsou dostupná všem pracovníkům v seznamu dokumentů, v elektronické nebo tištěné podobě.
- II. Přístupnost elektronických verzí dokumentů je vymezena přístupovými právy přidělenými vedoucím COV FD; úplná práva přístupu ke všem dokumentům mají vedoucí COV FD, zástupce vedoucího a manažer jakosti, ostatní pracovníci mají práva přístupu pouze pro čtení.
- III. Dokumenty jsou přezkoumávány před vydáním vedoucím COV FD. Po přezkoumání dokumentu mu vedoucí přidělí identifikační číslo, zapíše identifikaci schvalovatele a zařadí jej do seznamu (tím je dokument schválen). U tištěných dokumentů může být přezkoumání ještě stvrzeno přímo podpisem vedoucího v kolonce identifikace schvalovatele. Dále jsou dokumenty přezkoumávány pravidelně (v intervalech 3 měsíce). Přezkoumání se týká i legislativních dokumentů (nařízení vlády, TSI, dokumenty EU, dokumenty vztahující se k autorizaci, ...). Záznam o přezkoumání dokumentů je uložen v deskách společně s Přezkoumáním systému managementu; záznam obsahuje výčet dokumentů, kterých se přezkoumání týkalo.
- IV. Neplatné a/nebo zastaralé dokumenty a jejich kopie jsou stahovány manažerem jakosti, aby nebyly neúmyslně používány; dokumenty jsou označeny ZRUŠENO od: NAHRAZENO čím: a to jak v elektronické, tak i v tištěné formě. Změny a náhrady verzí jsou zaneseny do seznamu dokumentů.

- V. Podle potřeby jsou o změnách dokumentů osoby pracující v COV FD informovány na schůzkách, elektronickou poštou apod.
- VI. Neplatné dokumenty jsou uchovávány a archivovány podle příslušných předpisů, nejméně však po dobu 5 let. Po uplynutí této doby mohou být dokumenty skartovány.
- VII. Elektronické verze dokumentů jsou archivovány v půlročních intervalech, pokud došlo ke změnám. Media jsou uložena v uzamykatelné skříni.

5 Analýza a návrh

Kapitola analýza a návrh aplikace se zabývá definováním požadavků na vyvíjený informační systém Certifikačního orgánu pro výrobky při Fakultě dopravní, dále jen ISCOV a určením uživatelských rolí v systému s přihlédnutím k výše uvedeným postupům a pravidlům nastavených v laboratoři COV FD.

5.1 Požadavky

Požadavky popisují, co má systém dělat, jaké služby poskytuje a jaká jsou omezení jeho činnosti. Požadavky v podstatě odrážejí potřeby nebo představy zákazníka kladené na systém. Je možné je rozdělit do dvou základních skupin, a to na funkční a nefunkční.

Funkční požadavky určují, jaké služby bude systém nabízet či jakým způsobem bude provádět různé operace. Nefunkční požadavky kladou omezení na vyvíjený systém, často se jedná o hardwarová omezení například možnost ovládání různými vstupně-výstupními zařízeními. Mohou se též vztahovat na vlastnosti systému jako je spolehlivost, doba odezvy, vytíženost a podobně. Také mohou určovat pro jakou platformu je vyvíjený systém určen, jaké protokoly pro komunikaci bude používat atd. [19]

Funkční požadavky

Tab. 3 – Funkční požadavky kladené na navrhovaný systém

Číslo požadavku	Popis
REQ1	Systém umožní vkládat, editovat, mazat zakázky
REQ2	Systém bude zobrazovat aktuálně přihlášenému uživateli pouze zakázky ke kterým byl vedoucím přidělen
REQ3	Systém umožní vkládat, editovat, mazat potřebné dokumenty k jednotlivým zakázkám
REQ4	Systém umožní vkládat, editovat, mazat dokumenty
REQ5	Systém umožní přiřazovat oprávnění k dokumentům

Číslo požadavku	Popis
REQ6	Systém umožní vkládat revize dokumentů s komentářem
REQ7	Systém umožní uživateli s rolí „vedoucí“ přiřazovat oprávnění (čtení/zápis) ostatním uživatelům k jednotlivým dokumentům
REQ8	Systém bude zobrazovat aktuálně přihlášenému uživateli pouze dokumenty ke kterým má alespoň právo čtení
REQ9	Systém umožní provádět operace nad dokumentem pouze pověřeným uživatelům
REQ10	Systém umožní evidovat operace vykonané uživatelem nad dokumentem (čtení, mazání, revize)
REQ11	Systém umožní autentizaci pomocí uživatelského jména a hesla
REQ12	Systém umožní zasílání emailů osobám zainteresovaným u zakázky
REQ13	Systém umožní zasílání interních zpráv
REQ14	Systém bude automaticky zasílat interní zprávy zainteresovaným osobám při vytvoření zakázky
REQ15	Systém umožní při vytváření zakázky zvolit, vyhledat vedoucího k vytvářené zakázce
REQ16	Systém umožní nastavit oprávnění k zakázce vedoucímu zakázky nebo uživateli s rolí administrátor
REQ17	Systém nastaví pro zvoleného vedoucího zakázky oprávnění pro čtení, zápis, mazání, práva
REQ18	Systém umožní uchovávat verze dokumentů
REQ19	Systém umožní spravovat (mazat, přidávat, upravovat) uživatele
REQ20	Systém umožní každému uživateli přiřazovat roli
REQ21	Systém umožní změnu nastavení adresy a portu pro FTP, respektive FTPS
REQ22	Systém umožní změnu nastavení hesla, SMTP a uživatelského jména pro email, ze které jsou zasílány notifikace.
REQ23	Systém umožní při vytváření zakázky vybrat zákazníka z číselníku zákazníků
REQ24	Systém umožní vyhledávání zakázek dle jejich názvu
REQ25	Systém umožní vyhledávání zakázek dle data dokončení

Číslo požadavku	Popis
REQ26	Systém umožní vyhledávání zakázek dle názvu zadavatele (zákazníka)
REQ27	Systém umožní každé zakázce přiřadit stav (dokončená, dokončená s auditu,
REQ28	Systém bude zobrazovat počet zakázek, které mají být zpracovány v aktuálním měsíci
REQ29	Systém bude zobrazovat počet zakázek po termínu
REQ30	Systém bude zobrazovat celkový počet zakázek
REQ31	Systém umožní řazení zakázek dle názvu zákazníka, názvu zakázky, termínu dokončení, stavu
REQ32	Systém umožní zobrazit náhled souboru

Nefunkční požadavky

Tab. 4 – Nefunkční požadavky kladené na navrhovaný systém

Číslo požadavku	popis
REQN33	Systém bude implementován jako desktopová aplikace
REQN34	Systém bude schopný běhu pod operačním systémem Windows
REQN35	Systém bude využívat dokumentově orientovanou databázi
REQN36	Systém bude soubory ukládat v souborovém systému na vzdáleném úložišti nikoli v DB
REQN37	Systém bude realizován v programovacím jazyce C#
REQN38	Systém bude mít grafické uživatelské rozhraní ve WPF
REQN39	Systém bude třít nahrané dokumenty na cloudovém úložišti do složek dle zakázky, ke které patří
REQN40	Soubory budou na cloud úložiště nahrávány pomocí protokolu FTP, respektive FTPS
REQN41	Soubory budou na server nahrávány asynchronně
REQN42	Systém bude ovladatelný pomocí myši a klávesnice

Číslo požadavku	popis
REQN43	System nebude mít žádnou serverovou část

5.2 Uživatelské role

V navrhovaném systému ISCOV budou k dispozici dvě uživatelské role. Jedná se o roli „Uživatel“ a roli „Administrátor“.

5.2.1 Uživatel

První z dostupných rolí je uživatel. Jedná se o roli se základním oprávněním. Uživatelé s touto rolí budou moci prohlížet zakázky jež jim byly přiřazeny, uvidí počty vyřízených, nevyřízených zakázek a počet zakázek jež mají vyřídit v aktuálním měsíci. Stejně tak mohou prohlížet jim zasláné zprávy. U zakázek, u kterých byli uživatelé s touto rolí zvoleni jako vedoucí pak mají stejná oprávnění jako Administrátor. Mohou přidávat spolupracovníky, měnit jejich oprávnění k zakázce případně k jejím souborům. Mají také možnost nahrávat soubory a revize souborů atd. Oprávnění u uživatelů, kteří nejsou vedoucími zakázky záleží na nastavených oprávněních, která jim byla udělena, a to buď v rámci zakázky nebo pouze pro jednotlivé soubory. Může se jednat o práva čtení, čtení/zápis, čtení/zápis/mazání nebo oprávnění nejvyšší čtení/zápis/mazání/práva.

5.2.2 Administrátor

Role Administrátor, jak již vyplývá z názvu má plnou kontrolu nad systémem. Má stejná oprávnění jako Uživatel, navíc může vytvářet nové zakázky, uživatele, měnit uživatelské role. K dispozici má přehled všech zakázek a všech jejich souborů bez ohledu na nastavená oprávnění. Má možnost nastavení přístupových údajů k SMTP a FTPS.

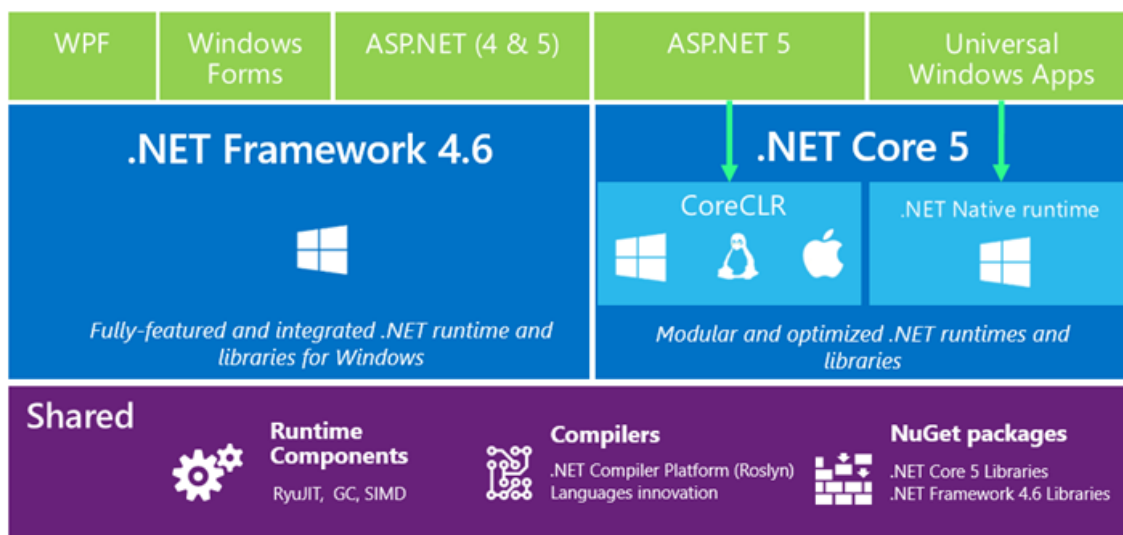
6 Zvolené technologie

Na základě provedené analýzy požadavků uvedených v kapitole 5.1 byly pro vývoj informačního systému ISCOV zvoleny následující technologie.

6.1 .NET

Jedná se o multiplatformní, dnes již open source platformu pro vývoj desktopových, webových, serverových, mobilních, herních a IoT aplikací. Součástí této platformy je několik programovacích jazyků: C#, F#, Visual Basic a rozsáhlá sada knihoven. Jako příklad těchto knihoven je možné uvést ASP.NET pro vytváření webových stránek, WCF pro budování síťových služeb nebo WPF pro tvorbu GUI aplikací. Sada základních knihoven poskytuje i prostředky pro práci s XML dokumenty, file systémem, pro komunikaci s databází a další.

Součástí .NETu je i běhové prostředí Common Language Runtime (CLR). Hlavním úkolem CLR je načítání umístování a správa .NET objektů. Stará se o nízko úrovněvé detaily jako je správa paměti, vláken, bezpečnosti, výkonu a dalších. Architektura .NET je zobrazena na Obr. 8. [20]



Obr. 8 – Architektura .NET

[Zdroj: <https://blogs.msdn.microsoft.com/bethmassi/2015/02/25/understanding-net-2015/>]

6.2 Jazyk C#

C# je objektově orientovaný, vysokoúrovňový a typově bezpečný jazyk od společnosti Microsoft, který vznikl jako součást frameworku .NET v roce 2002. Jako objektově orientovaný jazyk podporuje principy zapouzdření, dědičnosti a polymorfismu. Všechny metody a proměnné, a to včetně metody Main, jsou zapouzdřeny do tříd.

Tento jazyk disponuje jednoduchou a snadno zapamatovatelnou syntaxí, která je obdobná jako u dalších jazyků z rodiny C (například: C, C++, Objective C atd.) či jako u jazyka JAVA. C# stejně jako C++ podporuje nullable datové typy, enumeratory, lambda výrazy, překryté operátory, zpětné volání funkcí (přes delegáty) nebo dokonce přímý přístup do paměti a mnohé další. [20]

6.3 WPF & XAML

Windows Presentation Foundation je framework pro komplexní návrh a tvorbu GUI aplikací jež je součástí .NET od verze 3.0. Oproti svému předchůdci WindowsForms není závislý na rozhraní GDI+¹ a pro vykreslování grafického rozhraní využívá Direct3D, který je akcelerován grafickou kartou, což má za následek menší využití procesoru. Díky tomuto pak není GUI limitováno použitím grafických prvků operačního systému a je tedy možné vytvářet libovolně vypadající grafické aplikace, a to včetně efektů průhlednosti či animací. [21]

Pro zápis jednotlivých prvků je využíván deklarativní jazyk XAML. Tento jazyk není vnitřně spjatý s grafickým prvkem jako je tomu například u HTML, díky čemuž umožňuje oddělit grafické prvky UI od běhové logiky. XAML je v podstatě XML s tím rozdílem, že obsahuje řadu elementů, jejich atributů a hodnot, které jsou namapovány na objekty, jež jsou reprezentovány danou třídou. Při vytváření XAML dokumentu je tedy nutné dodržovat pravidla práce s XML dokumenty, a to

¹ Graphics Device Interface – slouží pro zobrazení a transformace grafických objektů v systémech Windows

především, že musí existovat pouze jeden kořenový element. Ukázka vytvoření okna je na Obr. 9 – Ukázka vytvoření okna v jazyce XAML. [22]

```
<Window x:Class="WpfAppTest.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:WpfAppTest"
        mc:Ignorable="d"
        Title="MainWindow" Height="450" Width="800">
    <Grid>

    </Grid>
</Window>
```

Obr. 9 – Ukázka vytvoření okna v jazyce XAML

6.4 LiteDb

LiteDb je malá, jednoduchá, dokumentově orientovaná NoSQL databáze kompletně napsaná v jazyce C# inspirovaná databází MongoDB. Tato databáze je vhodná pro webové, desktopové i mobilní aplikace.

Základní vlastnosti: [23]

- podpora pro UWP a Xamarin iOS/Android,
- ACID transakce,
- thread-safe a process-safe,
- jeden datový soubor,
- ochrana data pomocí journalu,
- mapování POJO tříd na BsonDocument,
- fluent API pro uživatelský mapping,
- podpora referencí mezi dokumenty (DbRef),
- uložště souborů,
- podpora LINQ,
- podpora indexů.

LiteDb pracuje s dokumenty a jejich kolekcemi, které slouží pro uchování dat. Dokumenty se používají k ukládání a načítání dat z a do datového souboru. Za dokument lze považovat jakoukoli POJO třídu (Obr. 10), která je převedena do formátu BSON a následně uložena do databázového souboru.

```
public class Zákazník
{
    public int Id { get; set; }
    public string Jmeno { get; set; }
    public string[] Telefon { get; set; }
    public bool Aktivni { get; set; }
}
```

Obr. 10 – Ukázka POJO třídy

7 Implementace

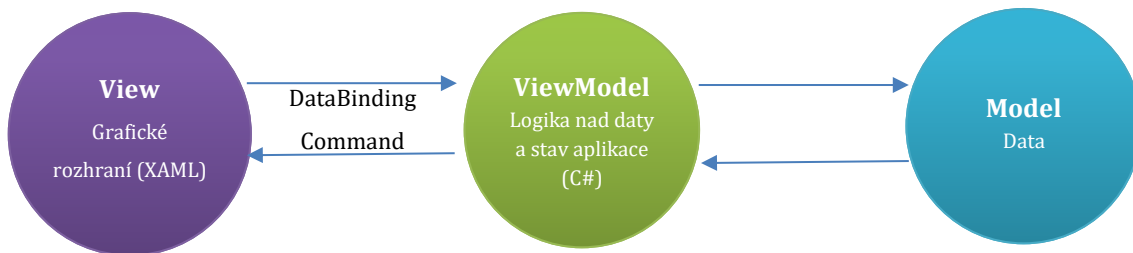
S přihlédnutím k výsledkům provedené analýzy požadavků plynoucích od pracovníků COV FD nebylo možné využít jako základ vyvíjeného informačního systému žádný z testovaných EDMS. Tato kapitola tedy popisuje tvorbu zcela nového IS nazvaného ISCOV, který bude vytvořen „na míru“ pro laboratoř COV FD.

Při vývoji byla využita agilní metodika s inkrementálním přístupem, při kterém je aplikace rozdělena na menší části a každá z nich byla řešena samostatně v úzké spolupráci se zástupci COV FD.

7.1 MVVM

Z důvodu využití WPF & XAML pro návrh a implementaci grafického uživatelského rozhraní je využit návrhový vzor MVVM jehož podporu Microsoft zabudoval přímo do WPF pro jeho snazší použití. Je možné zde tedy využívat binding a command jakožto náhradu za uživatelské rozhraní řízené za pomoci událostí. [24]

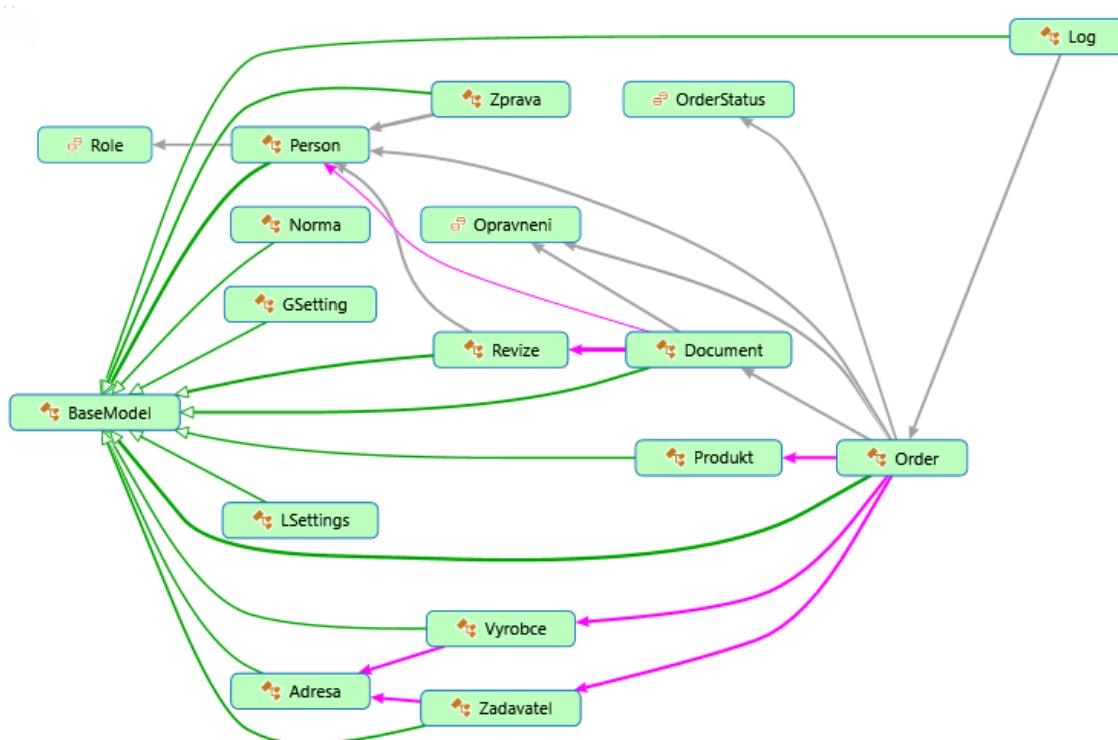
MVVM umožňuje oddělení logiky, dat a stavu aplikace od uživatelského rozhraní. Struktura aplikace je tedy rozdělena na tři hlavní části, jak to tento návrhový vzor předepisuje. První a zároveň nejdůležitější z těchto částí, která si zdrží stav aplikace a vykonává požadovanou logiku nad daty se nazývá ViewModel. ViewModel komunikuje s uživatelským rozhraním pomocí data bindingu, které na základě těchto dat vykresluje požadované uživatelské prvky. V aplikaci je použit i obousměrný binding pomocí něhož jsou uživatelem zadaná data přenášena zpět do ViewModelu k dalšímu zpracování. Třída VeiwModel zároveň poskytuje všechna potřebná data k zobrazení a provádí nad nimi uživatelem žádané operace pomocí commandu. Druhou částí nazývanou View je v podstatě uživatelské rozhraní, pomocí něhož jsou data prezentována a umožňuje uživateli interagovat s aplikací. Třetí nedílnou součástí jsou data, která jsou načítána ve ViewModelu do požadovaných datových struktur. Struktura návrhového vzoru je graficky prezentována na Obr. 11.



Obr. 11 – Struktura návrhového vzoru MVVM

7.2 Model v ISCOV

Model ve vyvíjené aplikaci ISCOV reprezentuje základní datové třídy, které v podobě objektů slouží pro uchování dat. Data se do těchto tříd načítají z databáze. Jako základní třída modelu slouží BaseModel. Jedná se o abstraktní třídu implementující rozhraní INotifyPropertyChanged a INotifyDataErrorInfo od které dědí všechny ostatní třídy modelu. Jak znázorňuje schéma Obr. 12.



Obr. 12 – Schéma modelové vrstvy systému ISCOV

Díky implementaci metod z rozhraní `INotifyPropertyChanged` je tedy možné informovat ostatní objekty o změně stavu jednotlivých vlastností v daném objektu. Implementaci metody `OnPropertyChanged` ukazuje následující kód na Obr. 13.

```
[NotifyPropertyChangedInvoker]
protected virtual void OnPropertyChanged([CallerMemberName]
    string propertyName = null, bool validate = false)
{
    PropertyChanged?.Invoke(this,
        new PropertyChangedEventArgs(propertyName));
    //Validace při změně hodnoty v modelových třídách
    if (validate)
        ValidateOneAsync(propertyName);
}
```

Obr. 13 – Ukázka implementace metody `OnPropertyChnaged`

Obdobně je tomu pro metody z rozhraní `INotifyDataErrorInfo`, které mají na starosti validaci jednotlivých vlastností dle pravidel definovaných v datové třídě. Ukázka definice pravidel je na Obr. 14.

```
[Required(ErrorMessage = "Email musí být vyplněn")]
[EmailAddress]
public string Email
{
    get => _email;
    set
    {
        if (value == _email) return;
        _email = value;
        OnPropertyChanged();
    }
}
```

Obr. 14 – Ukázka definice pravidel pro validaci v datové třídě

Validace následně probíhá buď při změně vlastnosti daného modelu nebo na vyžádání pomocí metody `Validate()`, umístěné ve třídě `BaseModel`. Ukázka kompletního kódu této metody je v příloze 9.1.

7.3 Databáze a přístup k datům

Použitá databáze `LiteDB` umožňuje vytvořit databázové schéma čistě ze softwarových tříd jazyka `C#`. Toho je docíleno především díky tomu, že využívá pro

ukládání dokumentů formát BSON². Vzhledem k této vlastnosti je možné provádět jakékoli změny v modelu aplikace, aniž by byla narušena její funkčnost. Není tedy problém například přidávat nové vlastnosti modelovým třídám, které budou do databáze automaticky uloženy. Dokonce není ani problém některou z vlastností modelu odstranit, pokud na ní samozřejmě neexistují referenční vazby v ostatních třídách. Ve starších záznamech v DB však zůstane zachována, bez vlivu na funkčnost. Pokud je potřeba tuto vlastnost odstranit i ze starších záznamů je zapotřebí to udělat ručně.

Celý přístup k datům v aplikaci ISCOV je řešen ve zvláštní DLL knihovně nazvané DataAccess. Tato knihovna je realizována pomocí návrhového vzoru Repository. Tento návrhový vzor předepisuje vytvořit generické rozhraní v tomto případě bylo nazváno IRepository obsahující metody pro základní databázové operace jako je mazání, přidávání, aktualizace a další. Kompletní rozhraní je možné vidět na Obr. 15

```
public interface IRepository : IDisposable
{
    void Delete<T>(Expression<Func<T, bool>> expression) where T : class, new();
    void Delete<T>(T item) where T : class, new();
    void DeleteAll<T>() where T : class, new();
    T FindOne<T>(Expression<Func<T, bool>> expression) where T : class, new();

    IEnumerable<T> Find<T>(Expression<Func<T, bool>> expression, int skip = 0,
        int limit = int.MaxValue) where T : class, new();
    IEnumerable<T> All<T>() where T : class, new();
    IEnumerable<T> All<T, TK>(Expression<Func<T, TK>> includeExpression)
        where T : class, new();
    void Add<T>(T item) where T : class, new();
    void Update<T>(T item) where T : class, new();
    void AddOrUpdate<T>(T item) where T : class, new();
    void AddOrUpdate<T>(IEnumerable<T> item) where T : class, new();
    void Add<T>(IEnumerable<T> items) where T : class, new();

    bool Exist<T>(Expression<Func<T, bool>> expression) where T : class,
        new();
    T LazyLoad<T>(string path, int id) where T : class, new();
    T LazyLoad<T>(string path, T co) where T : class, new();
    T LazyLoad<T, K>(Expression<Func<T, K>> co, T kdo) where T : class, new();
    bool CreateIndex<T, K>(Expression<Func<T, K>> co) where T : class, new();
    int Count<T>(Expression<Func<T, bool>> co) where T : class, new();
}
```

Obr. 15 – Rozhraní pro návrhový vzor Repository

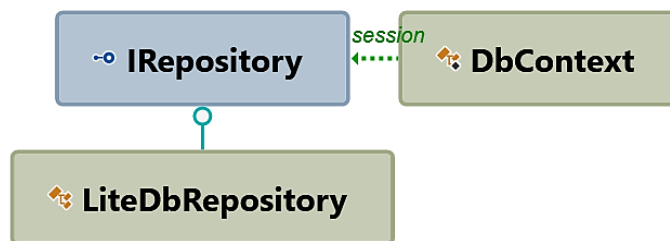
² Binární reprezentace formátu JSON s přidanou informací o datovém typu uchovávaných dat. Přestavuje rychlý a efektivní způsob serializace do binárního formátu.

Díky tomuto rozhraní je jedno jaký zdroj dat je v pozadí využíván a je snadné ho případně vyměnit za jiný. Třídou implementující výše zmíněné rozhraní je pak `LiteDbRepository`. Tato třída je již konkrétní implementací pro databázi LiteDB jakožto poskytovatele dat. Přístup k této třídě není přímý, nýbrž probíhá přes jeden statický globální kontext `DbContext`, díky čemuž je možné k databázi, respektive k poskytovateli dat přistupovat z libovolného místa aplikace. Ukázka přístupu je zobrazena na Obr. 16.

```
DbContext.Session.AddOrUpdate(doc);
```

Obr. 16 – Ukázka přístupu k poskytovateli dat

Schématické zobrazení knihovny `DataAccess` je na Obr. 17



Obr. 17 – Schématické zobrazení knihovny `DataAccess`

LiteDb je dokumentová databáze, která uchovává data v kolekcích. Mezi těmito kolekcemi není možné provádět klasický JOIN jako v případě databází relačních. Na místo toho je možné využít vložených dokumentů nebo vytvoření referenčního odkazu mezi těmito kolekcemi. Pro vytvoření tohoto odkazu je využita anotace `[BsonRef]` jak ukazuje Obr. 18.

```
[BsonRef]
public ObservableCollection<Document> DokumentyCollection
{
    get => _dokumentyCollection;
    set
    {
        if (Equals(value, _dokumentyCollection)) return;
        _dokumentyCollection = value;
        OnPropertyChanged();
    }
}
```

Obr. 18 – Využití referenčního odkazu mezi kolekcemi

Díky využití této vlastnosti je možné dokumenty rozdělit do více kolekcí. Tohoto se využívá především při větších objemech dat nebo častých změnách v kolekci dat. V tomto případě jsou soubory objednávky odděleny od hlavního dokumentu objednávky. To má za následek značné zmenšení nároků na paměť, protože při zobrazení objednávek v datagridu není zapotřebí deserializovat ještě všechny jejich dokumenty. Ty jsou z databáze načteny až ve chvíli, kdy je zobrazen detail objednávky. Využit je tzv. lazyLoading. S využitím anotace [BsonRef] je u každé objednávky uloženo pouze pole obsahující id a název kolekce kam se odkazovat. Názorná ukázka je na Obr. 19.

```
{
  "_id": 5481,
  "OrderId": "4b237874-cd2e-49fa-97db-c8544d2bfd17",
  "DokumentyCollection":
  [
    {
      "$id": 56,
      "$ref": "Document"
    },
    {
      "$id": 69,
      "$ref": "Document"
    }
  ],
  "Permissions":
  {
    "a41f8ac1-d7e3": "\u010cTEN\u00cd_Z\u00c1PIS",
    "a83fef49-1d56-891ca767e4d6": "\u010cTEN\u00cd"
  },
  "DatumPrijeti": {"$date": "2018-05-01T22:00:00.000000Z"},
  ...
}
```

Obr. 19 – Ukázka uchování objednávky ve formátu JSON

Přístup vložených dokumentů je použit například mezi dokumenty a jejich revizemi, protože se nepředpokládá, že by k jednomu souboru existovaly desítky či stovky revizí o nichž by musely být vedeny záznamy v podobě dokumentů v další kolekci databáze. Ukázka vloženého dokumentu je na Obr. 20.

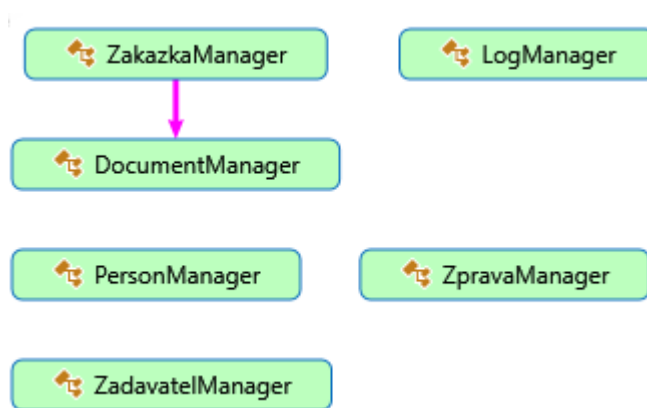

```

{
  "_id": 60,
  ...
  "TypeDocument": ".zip",
  "ListRevisions":
  [
    {
      "_id": 0,
      "NazevSouboruRevize": "Developers CollectionView.zip",
      "DatumVlozeni": {"$date": "2018-05-02T18:02:30.822000Z"},
      ...
    }
  ],
  ...
  "HasErrors": false
}

```

Obr. 20 – Ukázka vloženého dokumentu

Jako další vrstva, která je v aplikaci ISCOV obsažena a zajišťuje přístup k datům je vrstva manažerů. Manažery zajišťují business logiku nad modelovými třídami a zpřístupňují, modifikují a ukládají data z a do databáze. Všechna volání přistupující k databázi jsou tedy vedena přes manažery. Obr. 21 pak zobrazuje všechny manažer třídy obsažené ve vyvíjeném informačním systému ISCOV.



Obr. 21 – Manažery v aplikaci ISCOV

Ukázku použití ZakazkaManageru je možné vidět na Obr. 22.

```

ZakazkaManager m = new ZakazkaManager(zakazka);
m.UlozZakazku();

```

Obr. 22 – Ukázka použití manažeru

7.4 ViewModel v ISCOV

ViewModel je jednou z nejdůležitějších vrstev vyvíjené aplikace, která za pomoci manažerů zajišťuje načítání dat z databáze a jejich modifikaci. Zároveň drží stav aplikace. Tato vrstva je pomocí bindingu propojena s vrstvou View, do které promítá data. Stejně jako Modely musí i ViewModely pro správnou funkčnost bindingu implementovat rozhraní `INotifyPropertyChanged`. Jeho implementace byla provedena obdobně jako u modelových tříd v jednom společném předkovi, a to abstraktní třídě `BaseViewModel`. Kompletní schéma ViewModelu je dostupné v příloze 9.3.

Data získaná z manažerů jsou uchovávána v generických kolekcích typu `ObservableCollection<T>` kde `T` zastupuje konkrétní modelovou třídu. Ukázka tohoto volání je na Obr. 23. Tyto kolekce jsou využívány především kuli tomu, že dokáží vyvolat události při přidání případně odstranění prvku.

```
try
{
    OrdersCollection =
        new ObservableCollection<Order>(
            ZakazkaManager.DejVsechnyZakazkyProUzivatele((Person)
                Thread.CurrentPrincipal.Identity));
}
catch (Exception e)
{
    Dialogs.ShowDialog(e.Message + "\n" + e);
}
```

Obr. 23 – Ukázka získání objednávek pro přihlášeného uživatele

K tomu, aby mohl uživatel interagovat s aplikací je zapotřebí, aby už na úrovni ViewModelů byly implementovány metody pro jednotlivé ovládací prvky umístěné ve View. Tyto metody jsou ve ViewModelu reprezentovány vlastnostmi typu `ICommand` implementovány pomocí třídy `DelegateCommand`. Tato třída zároveň obsahuje i logiku pro povolení nebo zakázání vyvolání příkazů v podobě delegáta `canExecute`. Ukázka vytvoření vlastnosti je na Obr. 24.

```

public ICommand DetailZakazkyCommand { get; set; }

DetailZakazkyCommand =
    new DelegateCommand(o => Navigator.Navigate(new
        OrderDetailViewModel(SelectedOrder)));

```

Obr. 24 – Vlastnost ICommand pro navigaci na detail objednávky

ICommand vlastnosti jsou poté pomocí data bindingu přeneseny do View viz Obr. 25 a to na libovolný ovládací prvek implementující rozhraní ICommandSource. Zároveň může být z View předán zpět do ViewModelu libovolný object jako parametr commandu.

```

<Button
    Command="{Binding DetailZakazkyCommand}"
    Style="{DynamicResource MaterialDesignFlatButton}"
    IsEnabled="{Binding IsOrderSelected}"
    ToolTip="Detail zakázky" HorizontalAlignment="Left">
    <WrapPanel>
        <materialDesign:PackIcon Kind="Details" />
        <TextBlock Text="Detail zakázky" Margin="7,0,0,0" />
    </WrapPanel>
</Button>

```

Obr. 25 – Ukázka využití commandu ve View

Komunikaci mezi ViewModely zajišťuje třída Messenger. Každý ViewModel může díky této třídě komunikovat s jakýmkoli jiným ViewModelem, a to bez vzájemného vztahu, čímž je redukován počet spojení. V celé aplikaci existuje pouze jedna instance třídy Messenger, která je umístěná ve třídě BaseViewModel. Tím pádem ji mohou využívat všichni její potomci.

Následující ukázka (Obr. 26) zobrazuje registraci události „PridanaNovaZakazka“ pomocí výše zmíněné instance Messengeru v instanci MyOrderViewModel. Jako druhý parametr je zde odkaz na metodu, jež má být při této události vykonána.

```

MessengerInstance.Register<Order>(MessengerEventName.PridanaNovaZakazka,
    PridanaNovaZakazka);

```

Obr. 26 – Ukázka použití třídy Messenger pro komunikaci mezi MyOrderViewModelem a AddOrderViewModelem

Po zaregistrování dané události je jí možné v jiném ViewModelu, tady konkrétně v `AddOrderViewModel` vyvolat, jak ukazuje Obr. 27.

```
MessengerInstance.NotifyColleagues(MessengerEventName.PridanaNovaZakazka, order);
```

Obr. 27 – Ukázka odeslání notifikací při přidání nové zakázky

Událost dokonce může být zaregistrována ve více ViewModelech, pak jsou při použití metody `NotifyColleagues` informováni všichni.

7.5 Vrstva View v ISCOV

Vrstva View zajišťuje grafické uživatelské rozhraní aplikace. Byla navržena a napsána v deklarativním jazyce XAML. Tato vrstva je propojena s vrstvou ViewModel, jak již bylo poznamenáno výše, pomocí databindingu. Všechna View použitá v aplikaci jsou zobrazena na schématu Obr. 28.



Obr. 28 – View v aplikaci ISCOV

7.5.1 Navigace mezi View

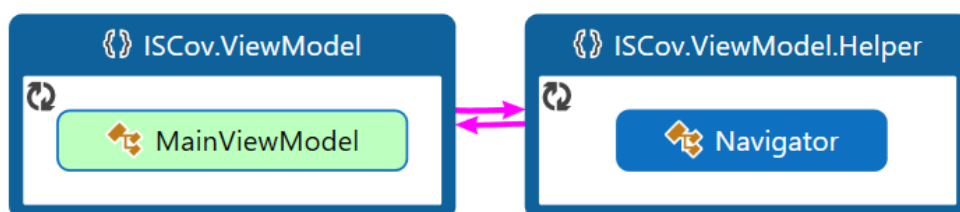
Uživatelské rozhraní je v aplikaci ISCOV navrženo tak, že jednotlivá View, která jsou reprezentována pomocí UserControl jsou načítána do komponenty ContentControl umístěné v pravé části MainView. Vlastnost Content v komponentě ContentControl je pomocí databindingu propojena s vlastností SelectedBaseViewModel a každé z View je v MainView definováno jako data template, k němuž je přiřazen patřičný ViewModel. Ukázka definování je na Obr. 29.

```
<DataTemplate DataType="{x:Type viewModel:DocumentViewModel}">
    <local:DocumentView />
</DataTemplate>
```

Obr. 29 – Definování data template v XAML

Díky tomu potom stačí změnit obsah vlastnosti SelectedBaseViewModel v MainViewModelu přiřazením jiného ViewModelu, čímž dojde k načtení nového obsahu v podobě View do ContentControl.

Aby bylo možné provádět navigaci mezi jednotlivými View vznikla třída Navigator. Jedná se o jednoduchou statickou třídu zpřístupňující metodu Navigate, z MainViewModelu, která jako parametr přijímá libovolného potomka BaseViewModelu, jehož instanci nastaví do vlastnosti SelectedBaseViewModel. Třída Navigator je schématicky zobrazena na Obr. 30.



Obr. 30 – Schématické zobrazení třídy Navigator

7.5.2 Databinding

Databinding prostupuje ve skutečnosti celou vrstvou View, každý ovládací prvek a každé pole je propojeno s patřičnou vlastností či commandem ve ViewModelu. Jednoduchá ukázka bindingu mezi ComboBoxem a odpovídající vlastností Zadavatel je na Obr. 31.

```

<ComboBox ItemsSource="{Binding ZadavatelCollection}"
SelectedItem="{Binding Zakazka.Zadavatel}"
IsTextSearchEnabled="True"
TextSearch.TextPath="NazevZadatele"
Style="{DynamicResource MaterialDesignFloatingHintComboBox}"
IsEditable="True"
Margin="2,15"

/>

```

Obr. 31 – Ukázka databindingu mezi View a ViewModelem

V tomto případě mohla být použita přímá vazba, protože položky `ItemsSource` nabindované z kolekce `ZadavatelCollection` jsou datového typu `Zadavatel` stejně jako položka `Zakazka.Zadavatel` nabindovaná jako `SelectedItem`. V aplikaci se ale vyskytují situace, kdy si dané datové typy neodpovídají v tomto případě je potom nutné použít tzv. konvertor.

7.5.3 Konvertory

Konvertor je malá jednoduchá třída implementující rozhraní `IValueConverter` a obsahující dvě metody. Metodu `Convert` a metodu `ConvertBack`. Implementace těchto metod zajišťuje správnou konverzi mezi danými datovými typy. Tyto metody jsou volány před zobrazením dat z `ViewModelu` případně před jejich odesláním zpět do `ViewModelu`. Tato konverze je automatická, stačí pouze ve `View` za deklarovat použití daného konvertoru pro danou datovou vazbu viz Obr. 32. V tomto případě se jedná o konverzi mezi koncovkou souboru a názvem k ní odpovídající ikony. Celou implementaci konvertoru je možné nalézt v 9.2.

```

<materialDesign:PackIcon Kind="{Binding TypeDocument,
Converter={StaticResource DocumentTypeIcon}}"

/>

```

Obr. 32 – Ukázka použití konvertoru

WPF, respektive XAML disponuje i tzv. multibindingem pomocí něhož je možné přenášet více než jednu informaci mezi `View` a `ViewModelem`. Díky multibindingu je tedy možné například zkombinovat obsah dvou polí (například: ve `View` budou dvě pole, jedno pro jméno druhé pro příjmení a pomocí konvertoru jsou obsahy zřetězeny a do `ViewModelu` přeneseny jako celé jméno).

K tomu, aby multibinding fungoval je zapotřebí implementovat konvertor s použitím rozhraní `IMultiValueConverter`. Ten se od klasického konvertoru liší především tím, že parametr metody `Convert` je pole typu `object` a parametr `targetType` metody `ConvertBack` je též typu pole. Liší se i jeho použití v XAML.

Následující ukázka kódu (Obr. 33) z aplikace ISCOV naznačuje využití multibindingu pro `command` parametr sloužící k přenosu dvou polí obsahující hesla zpět do `ViewModelu`.

```
<Button Content="ULOŽIT" Command="{Binding UlozitCommand}"
        HorizontalAlignment="Right" DockPanel.Dock="Bottom"
        VerticalAlignment="Bottom" Margin="0,5,5,5">

    <Button.CommandParameter>
        <MultiBinding Converter="{StaticResource
            MultipleParamConverter}">
            <Binding ElementName="HesloFTP" />
            <Binding ElementName="HesloEmail" />
        </MultiBinding>
    </Button.CommandParameter>

</Button>
```

Obr. 33 – Ukázka využití multibindingu

K tomuto řešení bylo přistoupeno především z toho důvodu, že přímé bindování hesel není z bezpečnostních důvodů vhodné.

7.5.4 Vlastní komponenty

Jazyk XAML umožňuje vytknout opakující se části do nových vlastních komponent. Toho bylo v aplikaci ISCOV využito pro vytvoření jednoduchého ukazatele činnosti tzv. loaderu. Tento loader obsahuje kruhový progresbar a vlastnost `IsWorking`, udávající, zda je loader viditelný či nikoli. K vytvoření nové komponenty je samozřejmě možné využít dědičnosti a daný prvek následně pouze rozšířit o požadovanou funkcionalitu.

Aby bylo možné přistupovat k vlastnosti `IsWorking` a měnit její hodnotu pomocí databindingu je zapotřebí založit v kontroleru loaderu `DependencyProperty`, dále vlastnost v tomto případě nazvanou `IsWorking` a obslužnou metodu, která bude volaná při každé změně hodnoty. Kompletní implementace je k nahlédnutí v příloze 9.4.

Použití takto vytvořené komponenty je de facto shodné jako použití komponenty systémové viz Obr. 34.

```
<helper1:LoadUserControl IsWorking="{Binding IsWorking}" />
```

Obr. 34 – Ukázka využití vlastní komponenty

7.6 Bezpečnost

V aplikaci ISCOV je pro autorizaci a autentizaci využita vlastní implementace tříd s využitím rozhraní `IIdentity` a `IPrincipal`. Rozhraní `IIdentity` zapouzdřuje totožnost uživatele a je implementováno modelovou třídou `Person`. Obsahuje pouze tři vlastnosti: `Name`, `AuthenticationType` a `IsAuthenticated`. Druhé rozhraní `IPrincipal` je implementováno třídou `Principal`. Ta drží identitu uživatele a umožňuje ověřit, zda se nachází v požadovaných rolích.

Přihlášení uživatele probíhá přes třídu `AuthUser` a její metodu `Login` ověřením jeho uživatelského jména a hesla oproti údajům v databázi. O tuto funkcionalitu se stará třída `AuthenticationService`, která porovná hash zadaného hesla s hashem uloženým v databázi pro dané uživatelské jméno. V případě, že se tyto hashe rovnají je vrácen objekt typu `Person` reprezentující aktuálně ověřeného uživatele, který je nastaven do vlastnosti `Identity` objektu `Principal`. Tímto krokem je přihlašování dokončeno.

Zabezpečení jednotlivých tříd je díky tomu velice jednoduché a je zajištěné pomocí třídy `PrincipalPermissionAttribute` použité jako anotace u dané třídy, případně metody, jak naznačuje následující ukázka (Obr. 35).

```
[PrincipalPermission(SecurityAction.Demand, Authenticated = true)]  
public class MyOrderViewModel : BaseViewModel  
{  
    ...  
}
```

Obr. 35 – Ukázka použití `PrincipalPermissionAttribute`

Jako parametr je možné uvést i roli ve které se má daný uživatel nacházet. V případě, že se přihlášený uživatel nebude v této roli nacházet či nebude vůbec přihlášen bude vyvolána výjimka.

Jako další bezpečností prvek, který je kromě autentizace a autorizace uživatelů, využit je šifrování. Celá databáze, ale i uživatelské údaje potřebné pro komunikaci se vzdálenými FTPS a SMTP servery jsou zašifrovány pomocí standartu AES256. Při práci s těmito údaji je pak využívána třída `System.Security.SecureString`, která reprezentuje text, který má být důvěrný a zajišťuje, že je zašifrován a odstraněn z paměti počítače ve chvíli, kdy již není potřeba.

Všechny metody pro šifrování, dešifrování a hashování jsou dostupné ze statické třídy `Encryption`.

8 Uživatelský manuál k aplikaci ISCOV

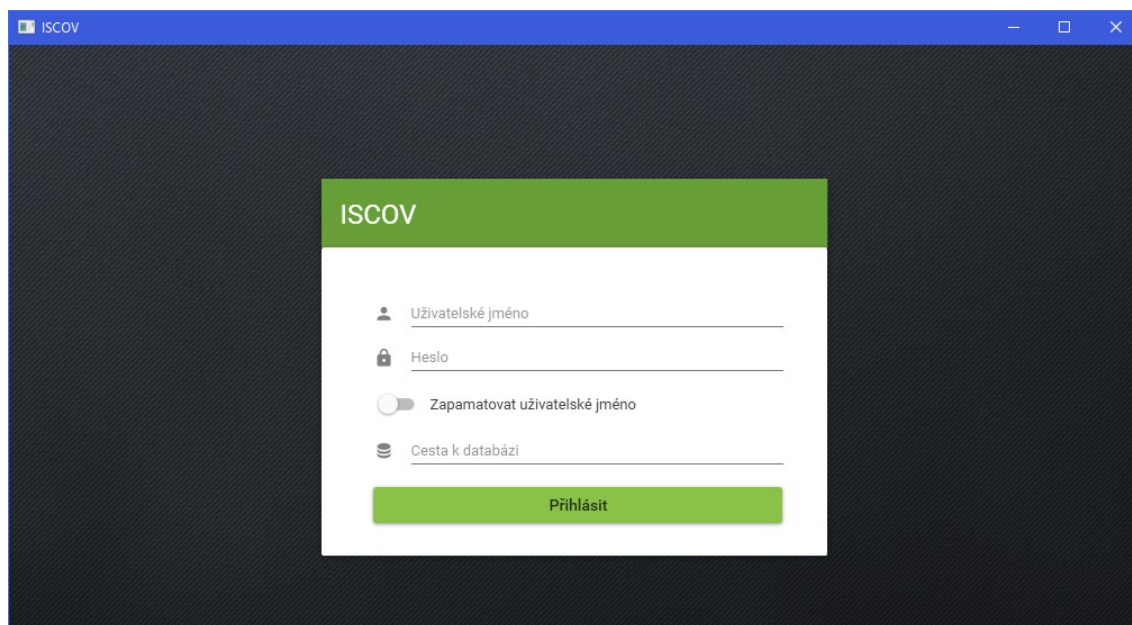
8.1 Instalace

Instalace aplikace probíhá pomocí klasického instalačního průvodce, pomocí kterého je vybrána cesta pro umístění souborů ISCOV. Dále je třeba nakopírovat soubor ISCOV_DB.db do sdíleného adresáře.

8.2 První spuštění

Při prvním spuštění (Obr. 36) je zapotřebí zadat do pole „Cesta k databázi“ celou cestu k databázovému souboru aplikace (ISCOV_DB.db) umístěném ve sdíleném adresáři a přihlásit se pomocí uživatelského jména a hesla pro správce.

- **Uživ. Jméno:** admin
- **Heslo:** admin



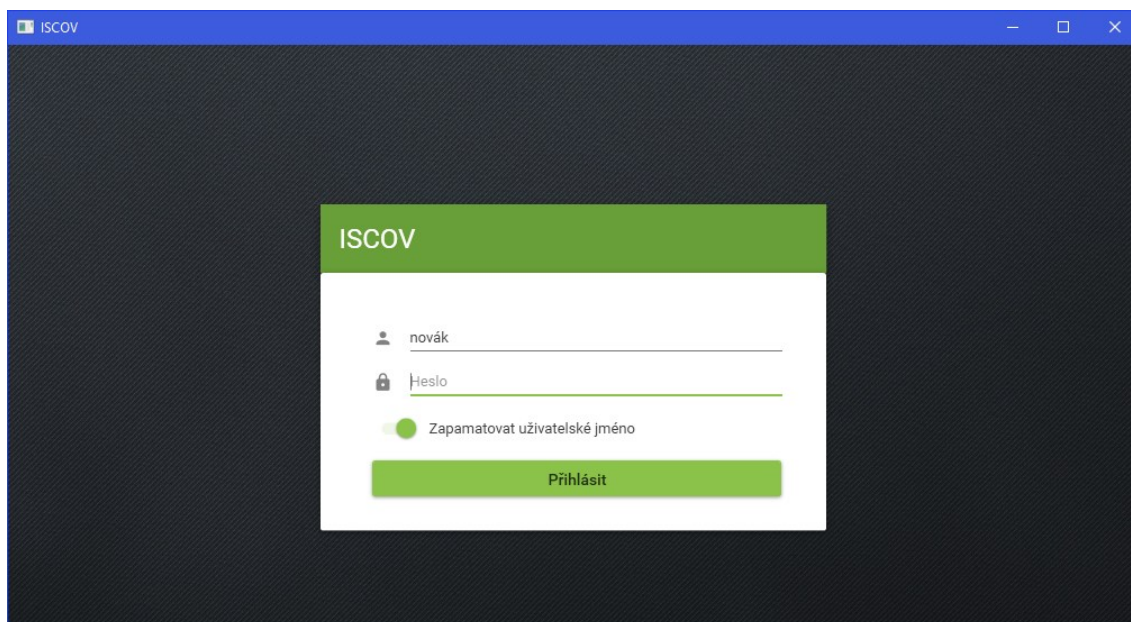
Obr. 36 – Obrazovka při prvním přihlášení do ISCOV

Toto jméno nebo alespoň heslo je doporučeno ihned změnit na obrazovce „Lidé“ viz kapitola 8.8.2 Editace stávajícího uživatele.

Dále je třeba ihned nastavit údaje pro komunikaci se servery FTPS a SMTP viz kapitola 8.9.

8.3 Přihlášení do ISCOV

Přihlašovací obrazovka je dostupná ihned po spuštění aplikace viz Obr. 37.



Obr. 37 – Přihlašovací obrazovka aplikace ISCOV

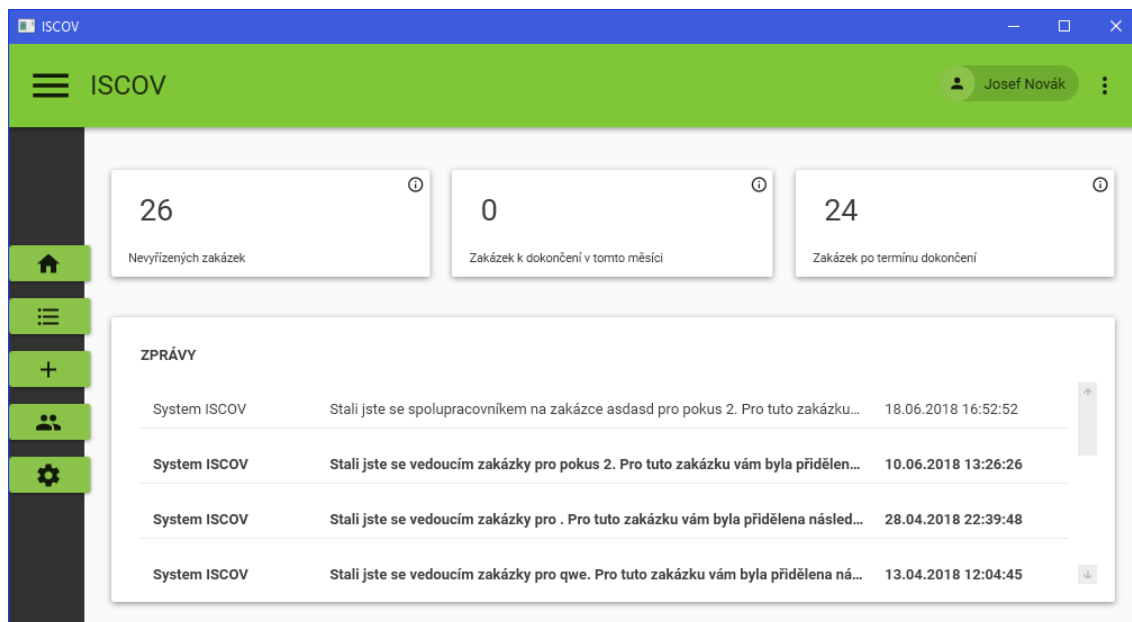
Přihlášení probíhá vyplněním přihlašovacích údajů v podobě jména, hesla a stisknutím tlačítka „Přihlásit“.

Zadané uživatelské jméno je možné uložit pomocí volby „zapamatovat uživatelské jméno“ nad tlačítkem přihlásit. Při příštím spuštění aplikace již nebude nutné uživatelské jméno znovu vyplňovat. V případě, že se jedná o první přihlášení nebo nebyla zaškrtnuta volba „zapamatovat uživatelské jméno“ budou obě pole prázdná.

8.4 Obrazovka „Dashboard“

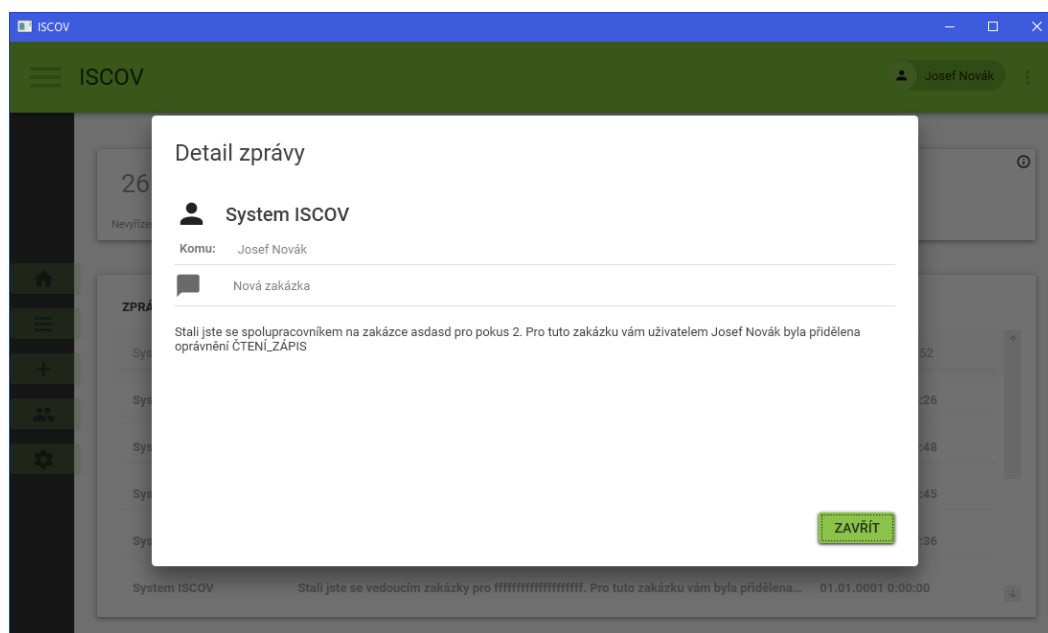
Obrazovka „Dashboard“ (Obr. 38) ukazuje ve vrchní části údaje o počtu zakázek. První box obsahuje počet nevyřízených zakázek pro aktuálně přihlášeného uživatele. Ve vedlejším boxu je počet zakázek, které mají být aktuálně přihlášeným uživatelem vyřízeny v průběhu aktuálního měsíce. Poslední box udává počet zakázek aktuálně přihlášeného uživatele, u nichž vypršel termín dokončení.

Ve spodní části okna jsou zprávy, které byly přihlášenému uživateli doručeny. Tučně jsou zvýrazněny zprávy, které ještě nebyly přečteny, respektive nebyly otevřeny v detailu.



Obr. 38 – Obrazovka "Dashboard"

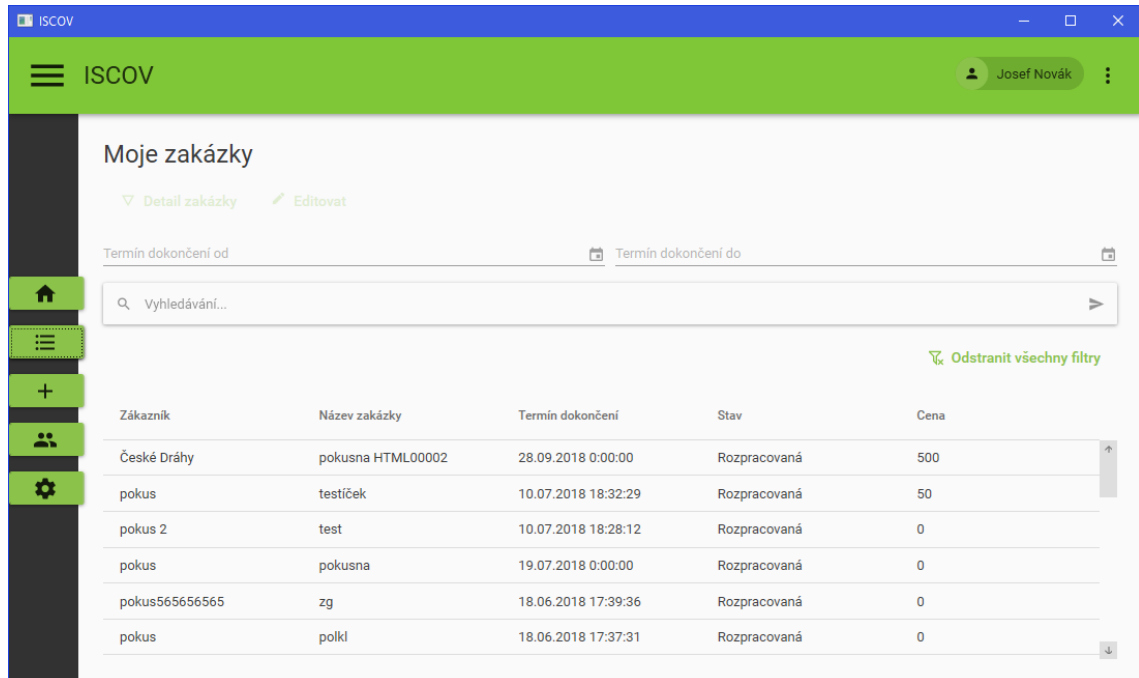
Dvojitým klikem na zprávu je možné zobrazit její detail viz Obr. 39.



Obr. 39 – Obrazovka "Detail zprávy"

8.5 Obrazovka „Moje zakázky“

Obrazovka „Moje zakázky“ (Obr. 40) zobrazuje zakázky, na kterých daný uživatel spolupracuje nebo je jejich vedoucím.



Moje zakázky

Detail zakázky Editovat

Termín dokončení od Termín dokončení do

Vyhledávání...

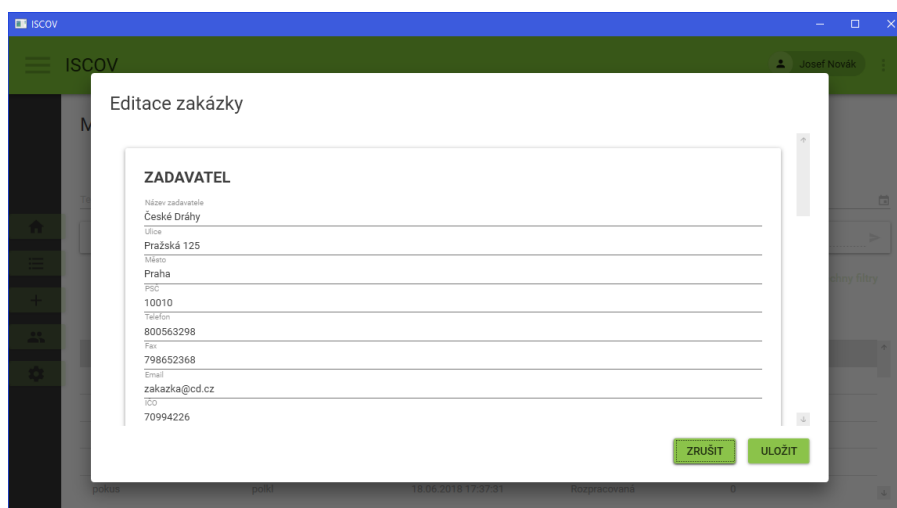
Odstranit všechny filtry

Zákazník	Název zakázky	Termín dokončení	Stav	Cena
České Dráhy	pokusna HTML00002	28.09.2018 0:00:00	Rozpracovaná	500
pokus	testiček	10.07.2018 18:32:29	Rozpracovaná	50
pokus 2	test	10.07.2018 18:28:12	Rozpracovaná	0
pokus	pokusna	19.07.2018 0:00:00	Rozpracovaná	0
pokus565656565	zg	18.06.2018 17:39:36	Rozpracovaná	0
pokus	polkl	18.06.2018 17:37:31	Rozpracovaná	0

Obr. 40 – Obrazovka "Moje zakázky"

Ve vrchní části této obrazovky jsou umístěna tlačítka pro editaci a zobrazení detailu zakázky. Tyto tlačítka jsou aktivní pouze v případě že je vybrána některá ze zakázek.

Tlačítko „Editovat“ poslouží k úpravám informací u vybrané zakázky (Obr. 41).



Editace zakázky

ZADAVATEL

Název zadavatele
České Dráhy

Ulice
Pražská 125

Město
Praha

PSČ
10010

Telefon
800563298

Fax
798652368


Email
zakazka@cd.cz

ICD
70994226

ZRUŠIT ULOŽIT

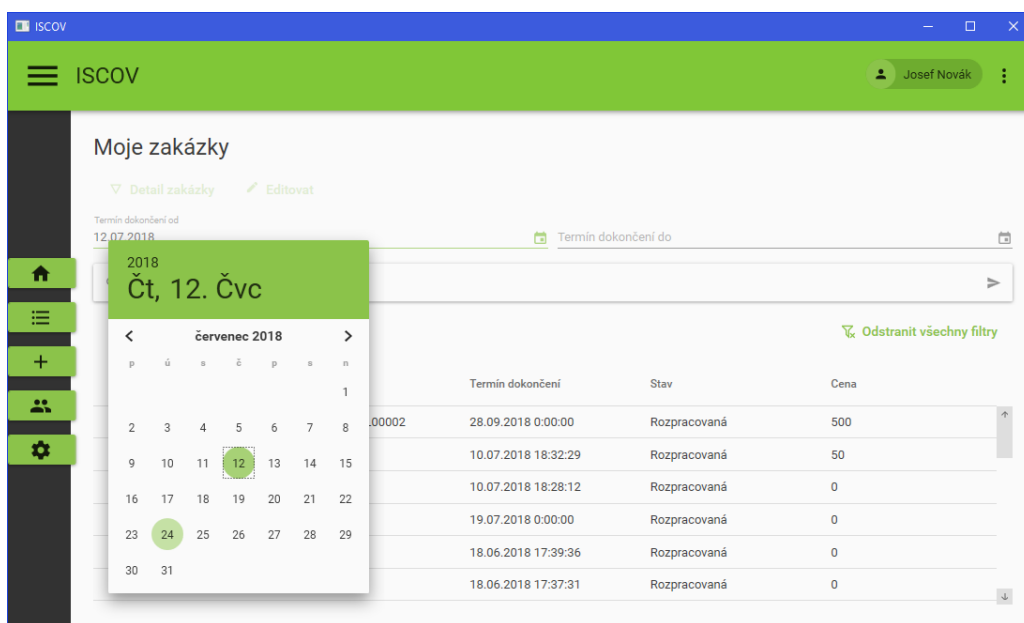
Obr. 41 – Ukázka "Editace zakázky"

Pomocí tlačítka „Detail zakázky“ je možné, pokud je některá zakázka vybrána, zobrazit její detail. Stejnou funkci má dvojitý klik na zakázku v tabulce ve spodní části okna. Ukázka okna je na Obr. 43.

Obrazovka „Moje zakázky“ umožňuje v zakázkách vyhledávat, a to buď na základě názvu zakázky případně názvu zadavatele. Hledaný výraz, který nemusí být kompletní, nemusí dokonce ani odpovídat začátku názvů je možné zadat do pole „Vyhledávání...“ a stisknout klávesu Enter, případně kliknout na ikonu „šipky“ . Ve spodní části okna poté budou zobrazeny výsledky hledání.

Další možností je vyhledávání na základě termínu dokončení. Ten může být stanoven jako rozmezí od do. Termín od, kterého se budou zakázky vyhledávat se zapíše do pole „termín dokončení od“ a termín dokončení do pak obdobně do pole „termín dokončení do“. V případě, že pole „termín dokončení do“ nebude vyplněno automaticky se doplní aktuální datum.

Obě data je možné vybrat z grafické komponenty (viz Obr. 42) kliknutím na ikonu kalendáře .



Obr. 42 – Ukázka výběru data

Výsledky budou opět zobrazeny ve spodní části okna po kliknutí na ikonu „šipky“.

Oba výše zmíněné filtry je možné kombinovat a provádět tak komplexnější vyhledávací dotazy typu: „zobraz pouze zakázky od 20.8.2017 do 30.8.2018 pro zákazníka České Dráhy“.

Zakázky je možné řadit kliknutím na záhlaví některého ze sloupců, a to jak vzestupně, tak sestupně, přičemž řadit lze vždy jen dle jednoho sloupce.

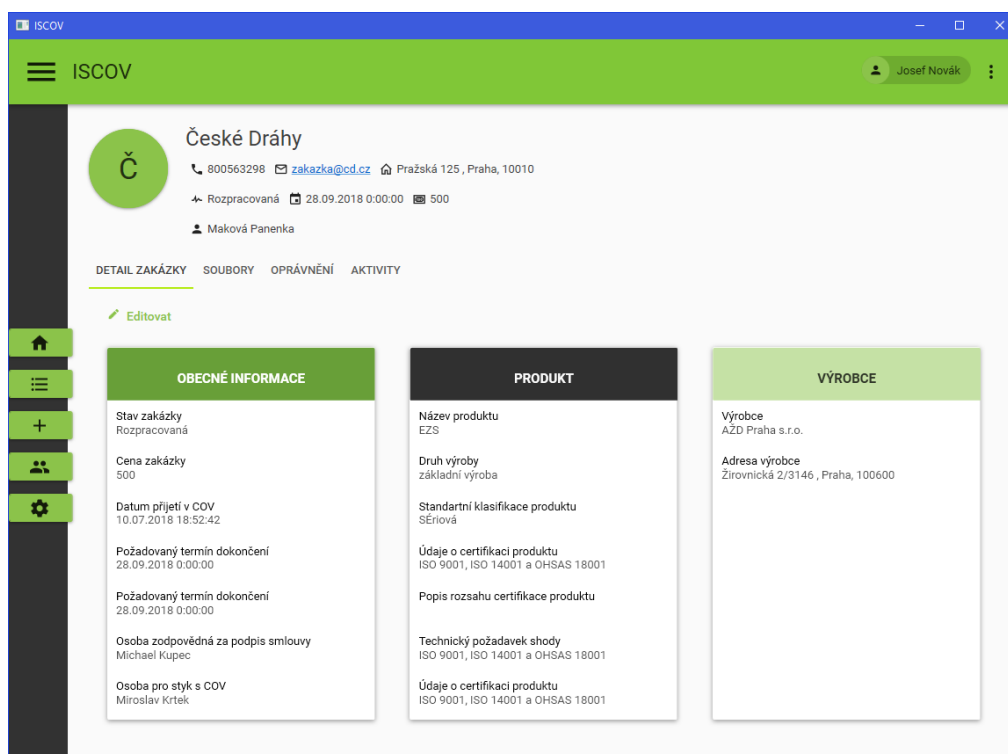
Všechny vyhledávací filtry včetně řazení je možné odstranit pomocí tlačítka „Odstranit všechny filtry“ v pravé části okna.

8.6 Obrazovka „Detail zakázky“

Obrazovka „Detail zakázky“ (Obr. 43) poskytuje komplexní informace o dané zakázce. Ve vrchní části je možné najít kontaktní informace na zadavatele jako je název, telefon, fax, email a adresu. Následují informace o samotné zakázce: jejím stavu, termínu dokončení, ceně a vedoucím.

8.6.1 Záložka „Detail zakázky“

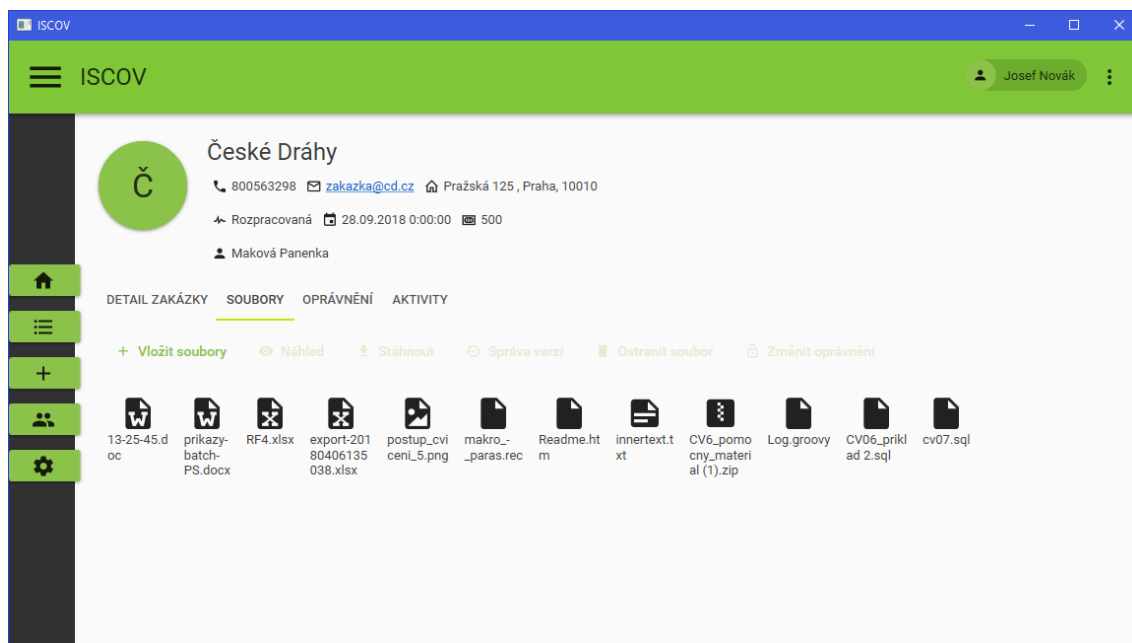
První záložka „Detail zakázky“ zobrazuje opět obecné informace o zakázce, podrobnější informace o produktu a informace o výrobci.



Obr. 43 – Obrazovka "Detail zakázky"

8.6.2 Záložka „Soubory“

Další záložka „Soubory“ (Obr. 44) zpřístupňuje pohled na soubory patřící k zakázce. Na této záložce je možné se soubory rovnou pracovat vkládat nové, zobrazovat náhled, stahovat, spravovat jejich revize, mazat a měnit k nim oprávnění. Vše za předpokladu dostatečného oprávnění přiděleného vedoucím zakázky nebo administrátorem systému.

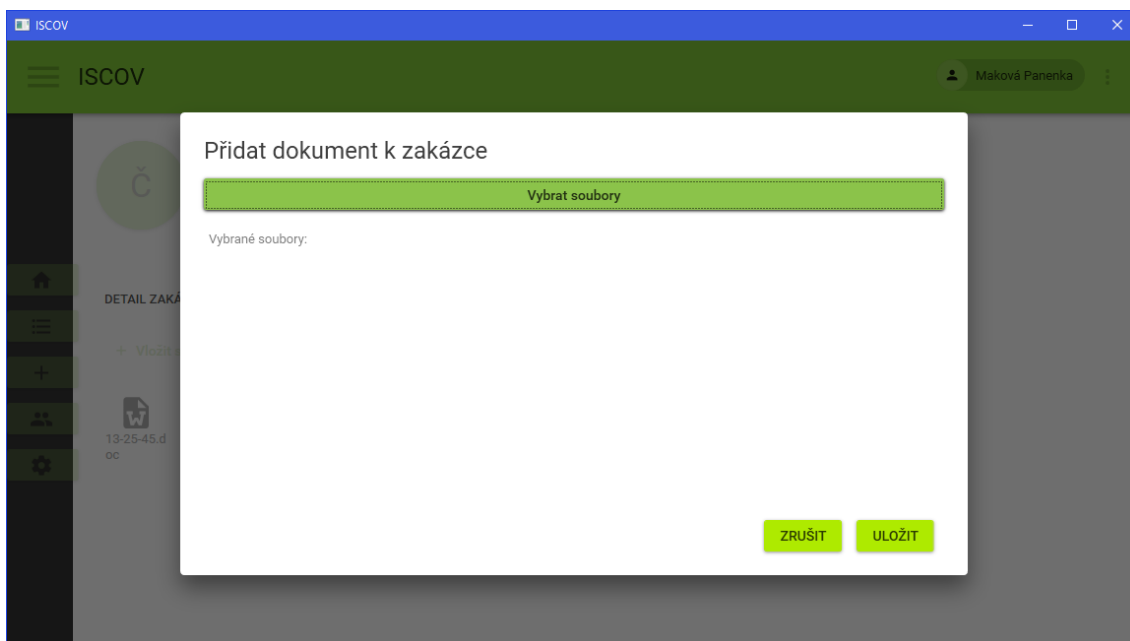


Obr. 44 – Obrazovka "Soubory"

8.6.2.1 Vložení nových souborů

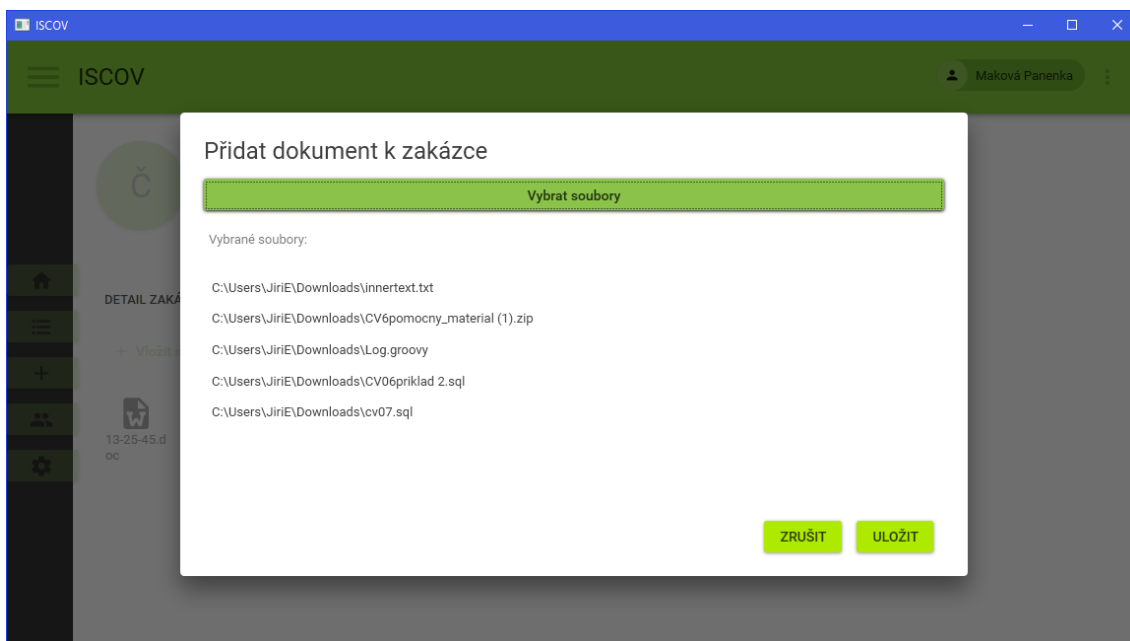
Vkládat nové soubory k zakázce mohou všichni uživatelé s právem alespoň pro čtení/zápis nastaveným na úrovni zakázky. V případě, že přihlášený uživatel nemá dostatečná oprávnění zobrazí se chybová hláška o tomto informující. Práva pak mohou být pro daného uživatele navýšena vedoucím zakázky nebo administrátorem systému. Více o tomto tématu v kapitole Změna oprávnění.

Vložení nových souborů je možné po kliknutí na tlačítko „Vložit soubory“, které spustí následující dialogové okno (Obr. 45).



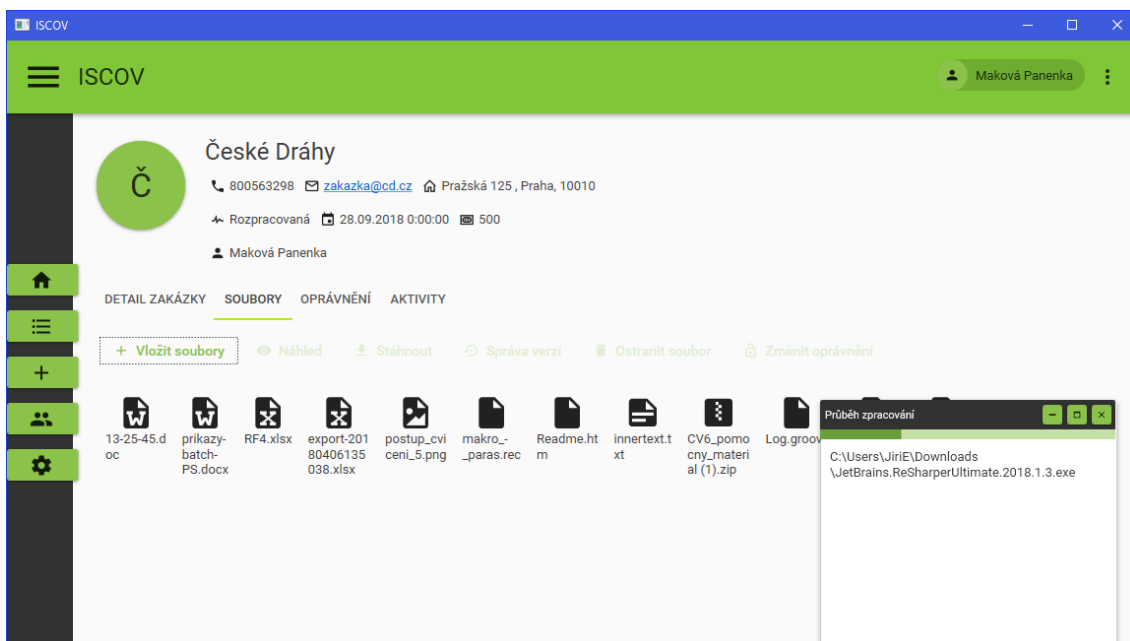
Obr. 45 – Ukázka výběru souborů

V tomto okně se nachází tlačítko „Vybrat soubory“, které po kliknutí spustí klasický systémový dialog pro výběr souborů. V tomto dialogu je možné vybrat jeden či více souborů, které budou následně zobrazeny uvnitř tohoto dialogového okna (Obr. 46).



Obr. 46 – Ukázka souborů přidávaných k zakázce

Stisknutím tlačítka „Uložit“ dojde k asynchronnímu nahrání souborů na FTPS server. Průběh tohoto nahrávání je možné sledovat v malém náhledovém okně umístěném v pravé dolní části hlavního okna (Obr. 47).



Obr. 47 – Ukázka nahrávání souboru

Po úspěšném nahrání jsou soubory ihned dostupné v hlavním okně aplikace.

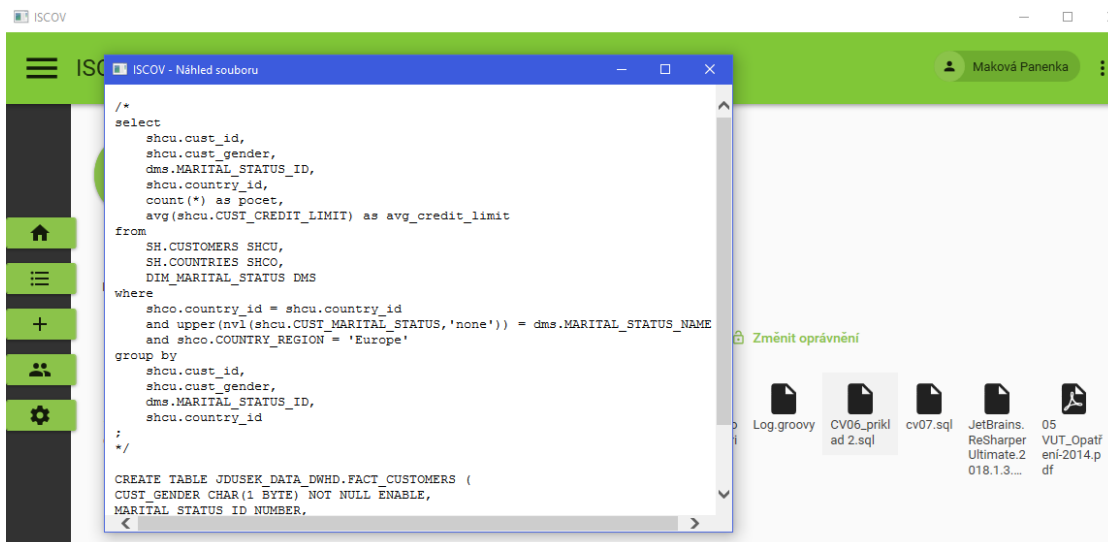
Tyto nově nahrané soubory mají uživatelská oprávnění zděděna ze zakázky. Pro jejich změnu je nutné se obrátit na vedoucího zakázky nebo na administrátora.

8.6.2.2 *Náhled souboru*

Náhled souboru (Obr. 48) je možné zobrazit kliknutím na požadovaný soubor a následně na tlačítko náhled nebo rovnou pomocí kontextového menu vyvolaného pravým tlačítkem myši nad vybraným souborem. Náhled je dostupný uživatelům s oprávněním alespoň pro čtení u daného souboru. Obsah souboru je následně zobrazen v novém okně.

Funkce náhled je dostupná pro následující formáty: ".svg", ".m4a", ".wav", ".htm", ".html", ".groovy", ".sql", ".gif", ".png", ".jpg", ".jpeg", ".pdf", ".zip", ".txt", ".doc", ".docx", ".ppt", ".pptx", ".mp3", ".wsd", ".xls" a ".xlsx" do velikosti 5 MB.

Pro zobrazení náhledu „pdf“ souboru je nutné mít nainstalovaný Acrobat Reader dostupný z <https://get.adobe.com/cz/reader/>. V opačném případě nemusí náhled pro „pdf“ fungovat správně, případně bude nutné soubor stáhnout a následně až zobrazit.




Obr. 48 – Ukázka náhledu souboru

8.6.2.3 Stažení souboru

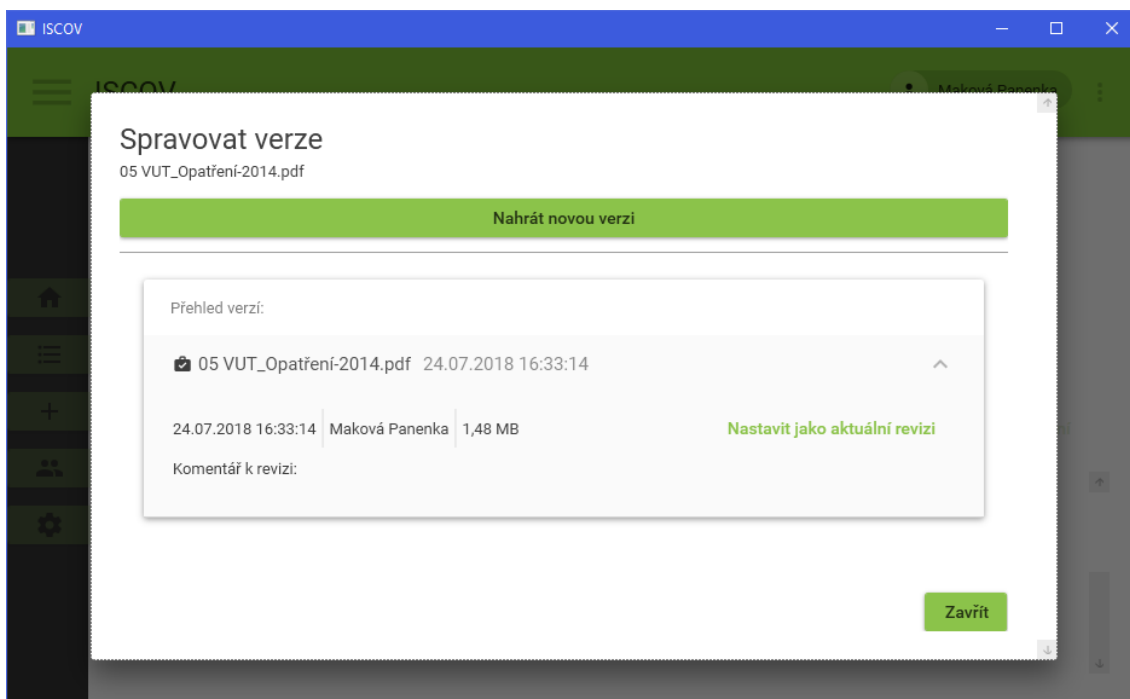
Obdobně jako u náhledu probíhá stažení souboru jeho vybráním a kliknutím na tlačítko „Stáhnout“ případně pomocí pravého kliku myši a volbou možnosti stáhnout. Po kliknutí bude zobrazen systémový dialog s možností výběru cesty. Soubor je umožněno stáhnout pouze uživatelů s minimálním oprávnění pro čtení na daném souboru

8.6.2.4 Správa verzí

Modul správa verzí (Obr. 49) je dostupný přes stejnojmenné tlačítko na záložce soubory, případně přes pravé tlačítko myši. V otevřeném dialogovém okně je v horní části umístěno tlačítko „Nahrát novou verzi“, po jehož stisknutí se otevře systémový dialog pro volbu souboru určeného k nahrání jako revize. Po vybrání souboru se otevře další dialogové okno, v němž je umožněno napsat komentář, který se stane součástí revize. Nově nahraný soubor se automaticky stává aktuální revizí.

Tento modul dále umožňuje vrátit se k předchozí verzi souboru pomocí jeho označení přes tlačítko „Nastavit jako aktuální revizi“. Aktuální revize je následně graficky označena pomocí ikony „kufříku“ .

Nahrávat revize je umožněno pouze uživatelům mající práva čtení/zápis, případně vedoucímu zakázky a administrátorovi IS.



Obr. 49 – Ukázka správy verzí

8.6.2.5 *Mazání souboru*

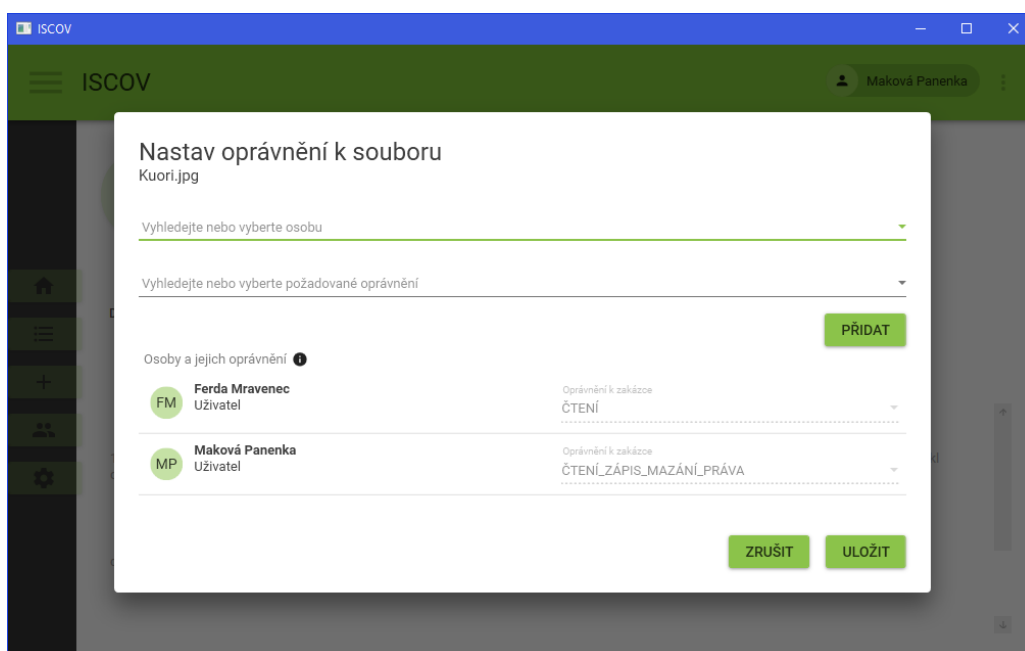
Mazání souborů ze zakázky je možné pomocí tlačítka „Odstranit soubor“ nebo přes pravé tlačítko myši. Tuto akci mají právo provádět pouze uživatelé s oprávněním čtení/zápis/mazání nastaveným na daném souboru, případně administrátor systému. Smazáním souboru dojde zároveň i k odstranění všech jeho revizí, a to jak z databáze, tak ze vzdáleného FTPS serveru. Soubor ani jeho revize již není možné obnovit.

8.6.2.6 *Změna oprávnění u souboru*

Změna oprávnění je dostupná přes tlačítko „Změnit oprávnění“ při vybraném souboru nebo pomocí pravého kliku myši. Po stisknutí tlačítka se zobrazí dialogové okno „Nastav oprávnění k souboru“ viz Obr. 50. Provádět změny oprávnění u souborů mají pouze uživatelé s právy čtení/zápis/mazání/práva pro daný soubor.

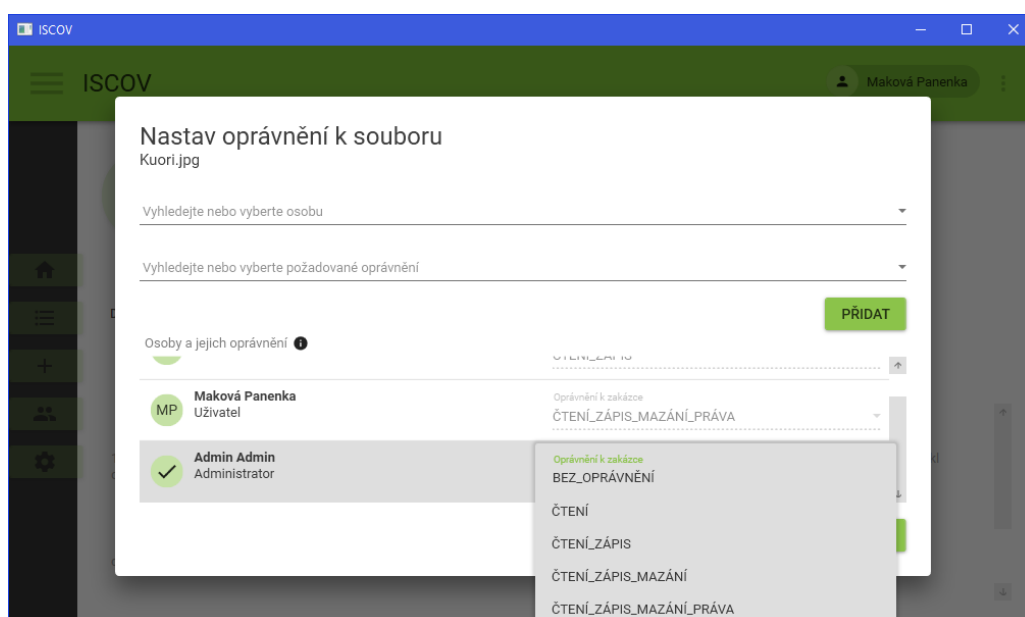
Ve vrchní části okna jsou k dispozici dvě výběrová pole. V prvním z nich je možné vybrat nebo vyhledat osobu pro kterou bude přidáno oprávnění ve druhém pak typ oprávnění. Kliknutím na tlačítko přidat dojde k přidání do seznamu „Osoby a jejich oprávnění“ v dolní části dialogového okna. Kliknutím na tlačítko uložit se provedené změny zapíše do databáze.

V případě nutnosti editovat existující oprávnění k zakázce je možné kliknutím na iniciálu uživatele odemknout možnost výběru jiného oprávnění (viz Obr. 51), zde vybrat nové oprávnění a kliknout na tlačítko „Uložit“.



Obr. 50 – Ukázka nastavení oprávnění

Při odstraňování uživatele ze seznamu oprávněných osob se postupuje obdobě jako při změně práv, přičemž při výběru oprávnění k zakázce je zvolena možnost „BEZ_OPRÁVNĚNÍ“. Provedené operace je nutné uložit kliknutím na tlačítko „Uložit“.

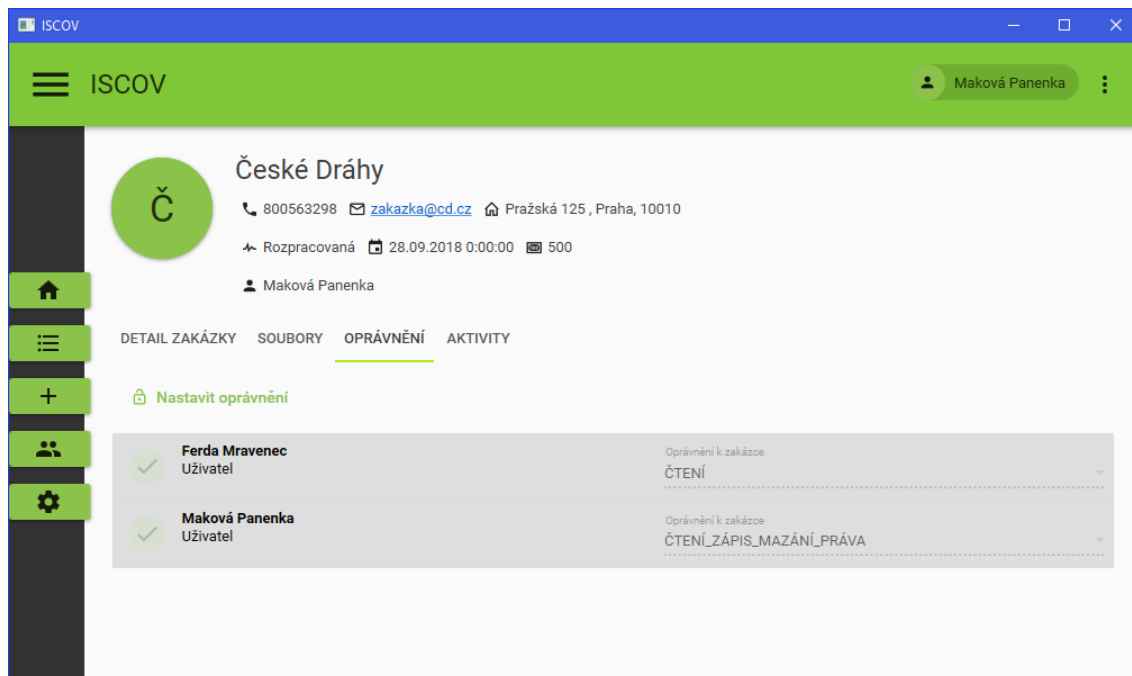


Obr. 51 – Ukázka editace existujících oprávnění k souboru

Provedené změny, které nemají být uloženy je možné zrušit tlačítkem „Zrušit“.

8.6.3 Záložka „Oprávnění“

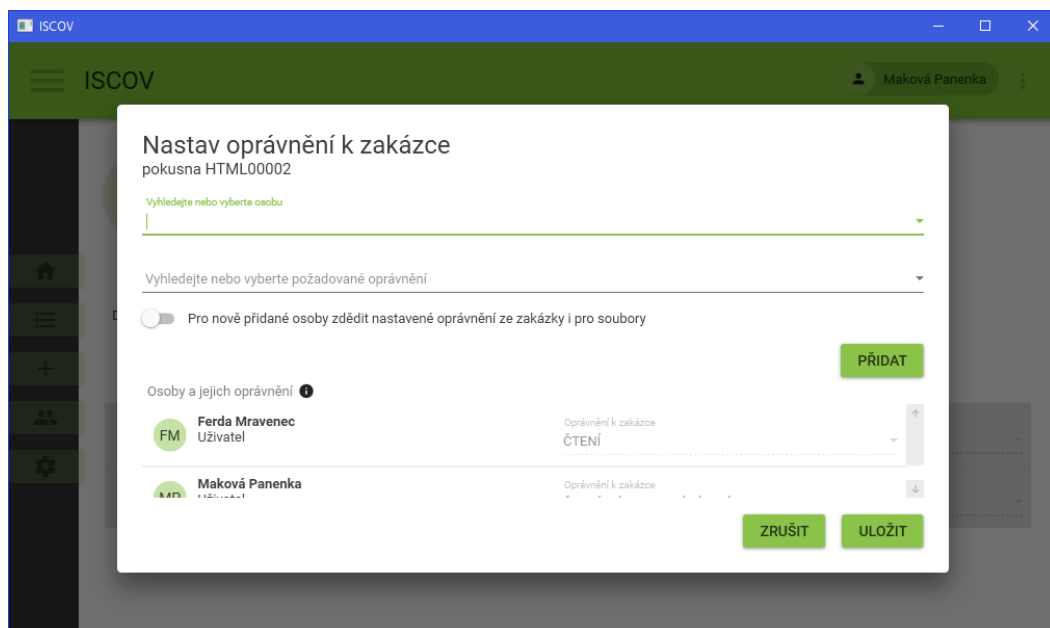
Na záložce „Oprávnění“ (Obr. 52) jsou vidět uživatelé a jejich oprávnění k zakázce.



Obr. 52 – Obrazovka "Oprávnění"

Pomocí tlačítka „Nastavit oprávnění“ je možné přidat nebo upravit uživatelská oprávnění k zakázce. Práva provádět změnu oprávnění a přidávat další spolupracovníky k zakázce mají jen osoby s oprávněním čtení/zápis/mazání/práva pro zakázku.

Princip je obdobný, jako tomu je u změny oprávnění u dokumentů. V horní části dialogového okna jsou umístěny dvě výběrová pole, v prvním z nich je možné vyhledat nebo vybrat osobu a ve druhém pak její oprávnění pro zakázku. Vybraná osoba s vybraným oprávněním se pomocí tlačítka „Přidat“ přesune do seznamu „Osoby a jejich oprávnění“. Nově přidané osoby nemají ve výchozím stavu možnost přistupovat k žádným souborům zakázky. V případě, že je potřeba nově přidaným uživatelům umožnit přístup k dokumentům zakázky, je možné použít přepínač pod druhým výběrovým polem. Díky tomu bude pro nově přidané osoby nastaveno zvolené oprávnění i pro všechny soubory zakázky. Ukázka nastavení oprávnění je na Obr. 53.

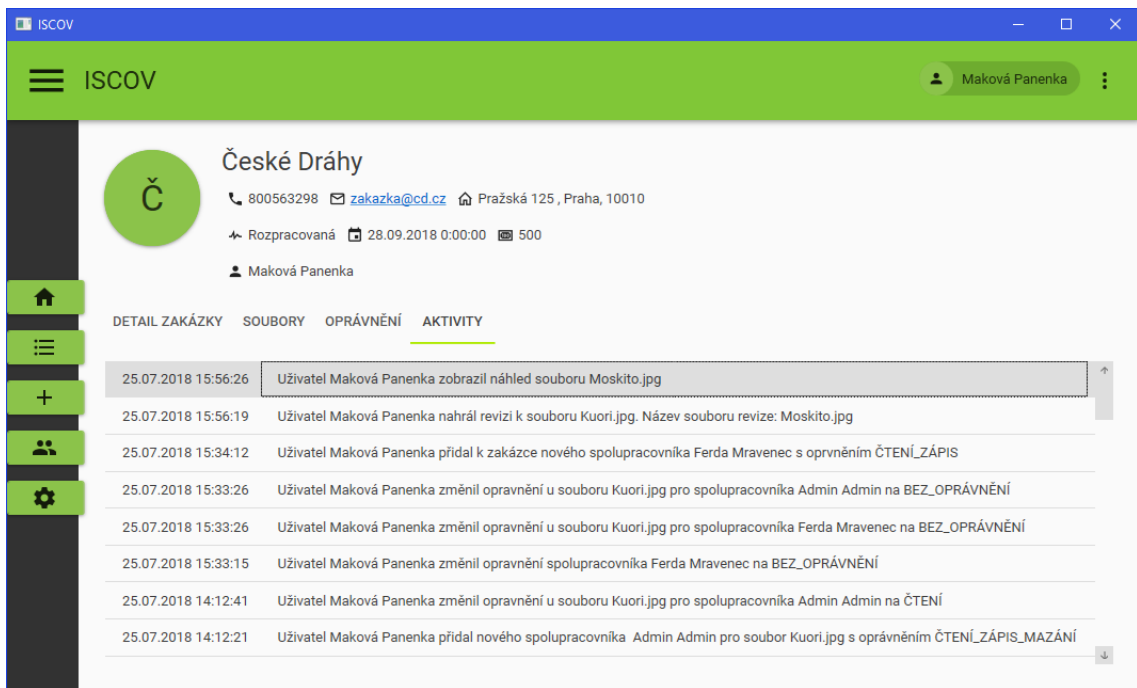


Obr. 53 – Ukázka nastavení oprávnění pro zakázku

8.6.4 Zložka „Aktivity“

Zložka „Aktivity“ (Obr. 54) zobrazuje informativní log o aktivitách prováděných na zakázce a k ní patřících souborech. Log je přístupný všem uživatelům mající alespoň práva pro čtení. Do logu není umožněno jakkoli zasahovat, a to ani uživatelům s rolí administrátor.

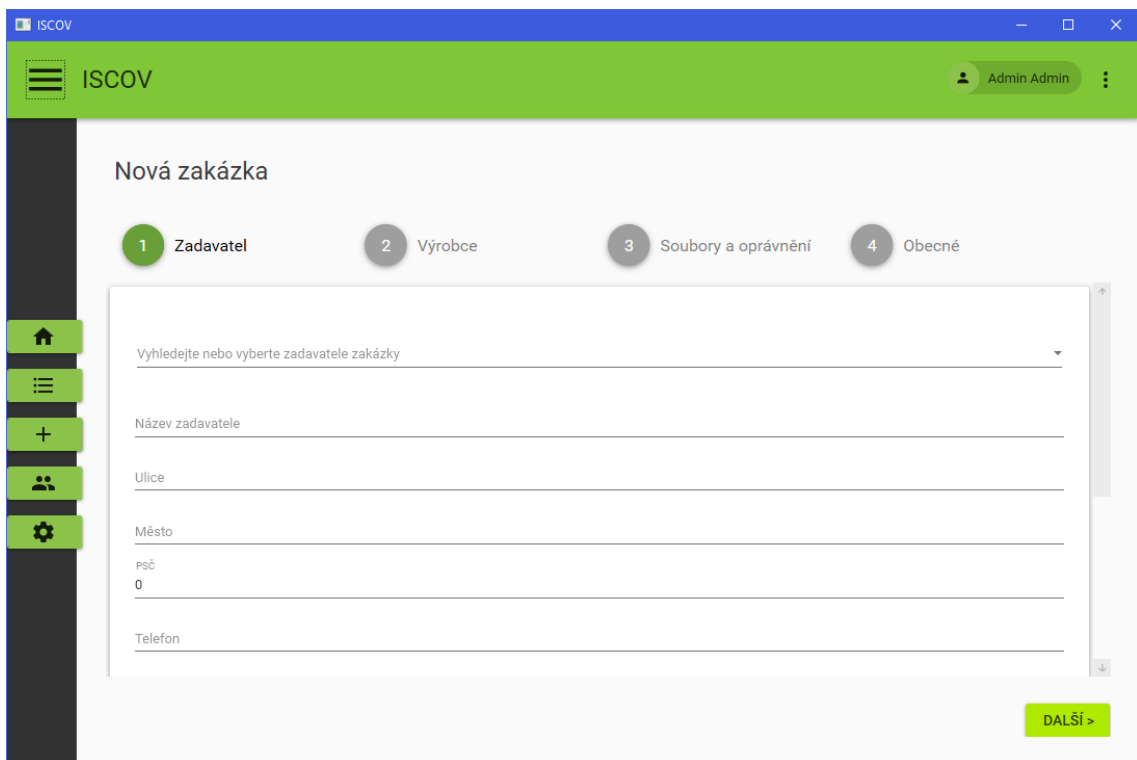
Zobrazeny jsou zde informace o změně oprávnění, a to jak k zakázce, tak k souboru, informace o nahrání nových souborů, přidání nového spolupracovníka, stažení, mazání souborů, náhledu, přidání nových revizí, informace o editaci zakázky. Vždy je doplněna informace o tom, kým byla daná akce provedena a kdy.



Obr. 54 – Záložka "Aktivity"

8.7 Obrazovka „Nová zakázka“

Obrazovka „Nová zakázka“ (Obr. 55) umožňuje uživatelů s rolí „Administrátor“ zakládat nové zakázky.



Obr. 55 – Obrazovka "Nová zakázka"

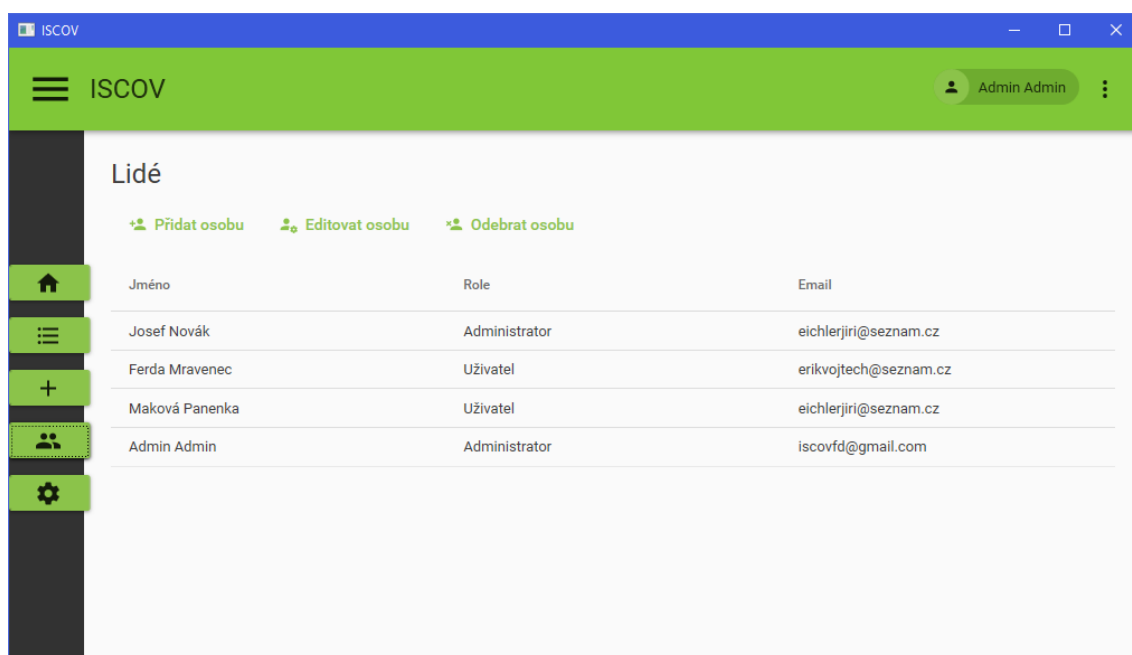
Tvorba nové zakázky probíhá mocí jednoduchého průvodce.

1. V prvním kroku je možné zadat údaje o zadavateli zakázky. Zadavatele je též možné vybrat nebo vyhledat v prvním výběrovém poli, přičemž po vybrání jsou údaje doplněny automaticky.
2. V dalším kroku následuje vyplnění údajů o výrobcí a výrobku, který má být certifikován.
3. Následující krok se věnuje přidání souborů k zakázce a nastavení oprávnění patřičným osobám. Přidání souborů je možné pomocí tlačítka „Vybrat soubory k zakázce“. Po kliknutí na toto tlačítko dojde k otevření systémového dialogu pro výběr souborů. Tyto se zobrazí v seznamu pod tlačítkem. Výběr osoby se provede kliknutím na její iniciálu a zvolením požadovaného oprávnění. Tyto osoby budou mít stejná oprávnění i ke všem přidaným souborům v tomto kroku.
4. V posledním kroku zadávání zakázky je zapotřebí zvolit cenu, datum dokončení, vedoucího zakázky a její název. Poslední dvě jmenované položky jsou zároveň jediná dvě povinná pole v celém procesu tvorby zakázky. Zakázka je uložena kliknutím na tlačítko „Uložit zakázku“, přičemž průběh ukládání zakázky, respektive asynchronní nahrávání souborů na FTPS je možné sledovat v automaticky otevřeném panelu průběhu. V průběhu tohoto je možné s aplikací normálně pracovat, není tedy nutné čekat na dokončení nahrávání.

8.8 Obrazovka „Lidé“

Obrazovka lidé (Obr. 56) slouží ke správě uživatelů v systému. K této funkci mají oprávnění přistupovat pouze osoby s rolí „Administrátor“.

Jedná se o velice jednoduchý modul umožňující základní operace jako je editace, přidávání a mazání.



Obr. 56 – Obrazovka "Lidé"

8.8.1 Přidání nového uživatele

Přidání nového uživatele probíhá kliknutím na tlačítko „Přidat osobu“ nebo pomocí pravého kliku myši a zvolením možnosti „Nový“, přičemž se otevře formulář, ve kterém je možné vyplnit základní údaje o sobě jako je jméno, příjmení, uživatelské jméno, heslo, email a role v ISCOV. Kliknutím na tlačítko „Uložit“ dojde k zapsání údajů do databáze.

8.8.2 Editace stávajícího uživatele

Editaci osoby je možné spustit vybráním osoby a kliknutím na tlačítko „Editovat osobu“ nebo pomocí pravého kliku myši nad vybranou osobou. Při editaci se otevře de facto stejný formulář jako tomu bylo u přidávání. Editovat je možné všechny údaje včetně hesla, které zde ale není zobrazeno (je uchovávan pouze hash). V případě, že zůstane pole heslo prázdné uživateli se heslo nezmění, v opačném případě bude heslo nahrazeno zadaným.

8.8.3 Odstranění uživatele

Odstranění uživatele probíhá pomocí tlačítka „Odstranit osobu“ nebo opět přes kontextové menu pravého tlačítka myši.

8.9 Obrazovka „Nastavení“

Obrazovka „Nastavení“ (Obr. 57), jak už napovídá její název, slouží pro nastavení informací potřebných pro komunikaci s FTPS a SMTP serverem. Tyto změny smí provádět pouze uživatel s rolí „Administrátor“. Nastavení jsou pak globálně uložena v databázi a jsou platná pro každého klienta.

The screenshot shows the 'Nastavení' (Settings) page in the ISCOV application. The interface features a green header with the ISCOV logo and a user profile 'Admin Admin'. A dark sidebar on the left contains navigation icons: home, list, plus, user, and settings (highlighted). The main content area is titled 'Nastavení' and contains two sections: 'FTP' and 'EMAIL'. The 'FTP' section has three input fields: 'Adresa FTP serveru' (127.0.0.1), 'Uživatelské jméno pro FTP server' (IS), and 'Heslo pro FTP server'. The 'EMAIL' section has one input field: 'Adresa SMTP serveru'. A green 'ULOŽIT' button is at the bottom right.

Obr. 57 – Obrazovka "Nastavení"

V první části je možné nastavit údaje pro FTPS jako je adresa, uživatelské jméno a heslo. V druhé části jsou uvedeny údaje pro SMTP server. Jeho adresa port, uživatelské jméno a heslo. V případě, že pole heslo zůstane prázdné, heslo nebude změněno.

Uložení nastavení proběhne po kliknutí na tlačítko „Uložit“.

Závěr

V úvodu této práce byl proveden náhled do problematiky elektronických dokument management systémů. Jsou zde definovány a popsány základní komponenty a vlastnosti, které musí tyto systémy splňovat. Následovalo srovnání existujících řešení. Do tohoto srovnání byly zařazeny produkty napříč nabízeným spektrem, a to jak placené, tak ty poskytované zdarma, případně jako open source. Kapitola 3 se poté zabývá legislativními požadavky na dokument management. Shrnuty jsou zde základní požadavky dle norem ČSN EN. Větší část této kapitoly je následně věnována evropskému nařízení pro ochranu osobních údajů tzv. GRPD. Po ní následuje představení Certifikačního orgánu pro výrobky při Fakultě dopravní ČVUT. V tomto představení je popsán obor činnosti a způsob zpracování zakázek od jejich přijetí vedoucím, až po závěrečná rozhodnutí o udělení či neudělení certifikátu. Poukázáno je také na fakt, že COV FD při své činnosti pracuje s rozličným množstvím dokumentů, které jsou na jedné straně tvořeny administrativními dokumenty a na straně druhé dokumenty patřícími k zakázce jako jsou plány a zdrojové texty pro certifikované výrobky. Tato kapitola obsahuje také popis stávajícího vedení dokumentace v laboratoři. V kapitole 5 byla posléze provedena analýza potřeb zkušební laboratoře. Ve spolupráci se zástupci COV FD a s přihlédnutím k legislativním nárokům a vnitřním předpisům, byly stanoveny požadavky na nově vznikající informační systém. Zároveň zde byly vymezeny uživatelské role a jejich oprávnění. Na základě výsledků vzešlých z analýzy požadavků však nebylo možné mezi testovanými dokument management systémy vybrat žádný, který by se mohl stát základem pro nový informační systém. Bylo tedy přistoupeno k implementaci zcela vlastního řešení informačního systému nazvaného ISCOV, postaveného „na míru“ laboratoři COV FD. Tento systém splňuje požadavky vznesené pracovníky COV FD v průběhu jeho tvorby. Nicméně teprve jeho praktické nasazení ukáže nutnost případných úprav či rozšíření a jistě i oprav bugů, které se v aplikaci nachází. V poslední kapitole práce je uveden podrobný uživatelský manuál aplikace. Systém ISCOV je možné rozšířit například o možnost centrálního přihlašování dostupného v laboratoři, zavést možnost automatických upozornění v případě nutnosti auditu nebo o možnost tvorby a tisku dokumentů přímo v prostředí systému. Dalším praktickým rozšířením by bylo zobrazovat komentáře, případně

aktivity vedené u souborů či zakázek v pravé části okna pro detail zakázky a mnoho dalších.

Veškeré hlavní cíle stanovené v zadání a úvodu práce, ke kterým patří analýza potřeb laboratoře, srovnání dostupných elektronických dokument management systémů, a především implementace vlastního informačního systému jakožto desktopové aplikace byly splněny.

9 Přílohy

9.1 Příloha A

Ukázka implementace metody Validate().

```
public void Validate()
{
    lock (_lock)
    {
        var context = new ValidationContext(this, null, null);
        ICollection<ValidationResult> results = new List<ValidationResult>();
        Validator.TryValidateObject(this, context, results, true);

        foreach (var valuePair in _errors.ToList())
            if (results.All(r => r.MemberNames.All(x => x != valuePair.Key)))
            {
                List<string> outLi;
                _errors.TryRemove(valuePair.Key, out outLi);
                OnErrorsChanged(valuePair.Key);
            }

        var props = from validationResult in results
                    from x in validationResult.MemberNames
                    group validationResult by x
                    into grouping
                    select grouping
                    ;

        foreach (var prop in props)
        {
            var messages = prop.Select(r => r.ErrorMessage).ToList();

            if (_errors.ContainsKey(prop.Key))
            {
                List<string> outLi;
                _errors.TryRemove(prop.Key, out outLi);
            }

            _errors.TryAdd(prop.Key, messages);
            OnErrorsChanged(prop.Key);
        }
    }
}
```

9.2 Příloha B

Ukázka implementace konvertoru mezi koncovkou souboru a názvem ikony

```
public class DocumentTypeIcon : IValueConverter
{
    private static readonly Dictionary<string, string> Pripony = new
Dictionary<string, string>
    {
        {".jpg", "FileImage"},
        {".png", "FileImage"},
        {".bmp", "FileImage"},
        {".svg", "FileImage"},
        {".gif", "FileImage"},

        {".doc", "FileWord"},
        {".docx", "FileWord"},

        {".xls", "FileExcel"},
        {".xlsx", "FileExcel"},

        {".ppt", "FilePowerpoint"},
        {".pptx", "FilePowerpoint"},

        {".pdf", "FilePdf"},

        {".txt", "NoteText"},

        {".xml", "FileXml"},

        {".zip", "ZipBox"}
    };

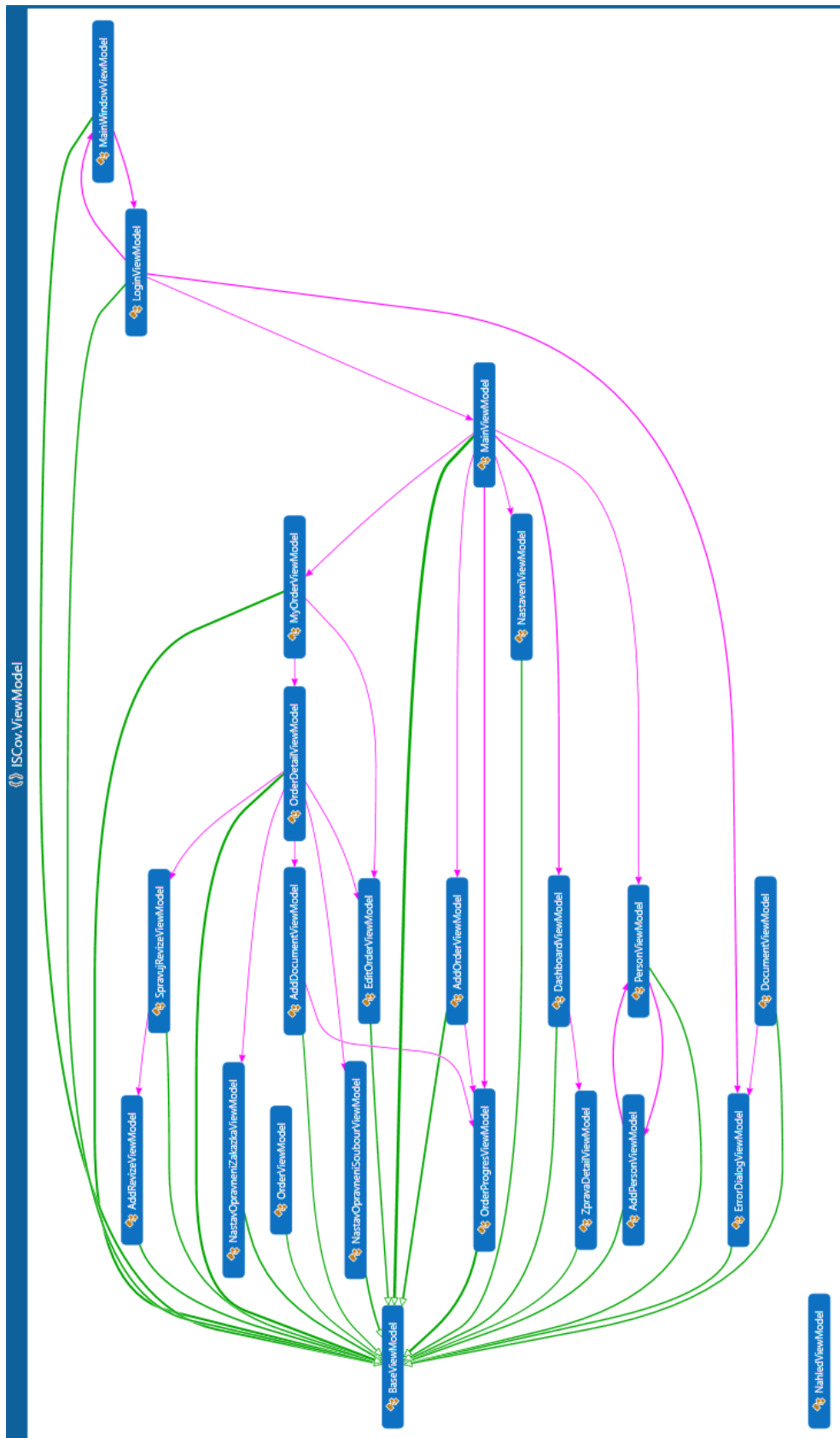
    public object Convert(object value, Type targetType, object parameter,
CultureInfo culture)
    {
        var path = (string) value;
        path = path.ToLower();

        return Pripony.ContainsKey(path) ? Pripony[path] : "File";
    }

    public object ConvertBack(object value, Type targetType, object
parameter, CultureInfo culture)
    {
        //není třeba ikona se nebude z GUI měnit
    }
}
```

9.3 Příloha C

Schéma ViewModelu aplikace ISCOV



9.4 Příloha D

Ukázka implementace kontroleru vlastní komponenty

```
public partial class LoadUserControl : UserControl
{
    public static readonly DependencyProperty IsWorkingInputProperty =
        DependencyProperty.Register("IsWorking", typeof(bool),
            typeof(LoadUserControl),
            new FrameworkPropertyMetadata(IsInputPropertyChanged));

    public LoadUserControl()
    {
        InitializeComponent();
    }

    public bool IsWorking
    {
        get => (bool) GetValue(IsWorkingInputProperty);
        set => SetValue(IsWorkingInputProperty, value);
    }

    /// <summary>
    ///     Metoda je volána při každé změně vlastnosti IsWorking
    /// </summary>
    private static void IsInputPropertyChanged(
        DependencyObject d, DependencyPropertyChangedEventArgs e)
    {
        var f = (LoadUserControl) d;
        var hodnota = (bool) e.NewValue;
        f.Visibility = hodnota == false ? Visibility.Collapsed :
            Visibility.Visible;
    }
}
```

10 Citovaná literatura

- [1] TVRDÍKOVÁ, Milena. *Aplikace moderních informačních technologií v řízení firmy: nástroje ke zvyšování kvality informačních systémů*. 1. vyd. Praha: Grada, 2008. Management v informační společnosti. ISBN 978-80-247-2728-8.
- [2] ADAM, Azad. *Implementing electronic document and record management systems*. Boca Raton: Auerbach Publications, 2008. ISBN 978-0849380594.
- [3] *SpringCM | When Your Documents Flow, Work Flows* [online]. 2018 [cit. 2018-08-11]. Dostupné z: <http://www.springcm.com/>
- [4] KRČÁL, Martin. Document management systems. *Inflow.cz* [online]. 2008 [cit. 2018-07-10]. Dostupné z: <http://www.inflow.cz/document-management-systems>
- [5] *Collaboration Software: Share Documents Online, Intranets, Web Databases* [online]. 2018 [cit. 2018-08-11]. Dostupné z: <http://www.hyperoffice.com/>
- [6] *Document Management System Software | LogicalDOC* [online]. 2018 [cit. 2018-08-11]. Dostupné z: <https://www.logicaldoc.com/>
- [7] *Free Document Management Software - Open source document management for PHP* [online]. 2018 [cit. 2018-08-11]. Dostupné z: <http://www.opendocman.com/>
- [8] *Capsa | Sdílejte data v týmu. Pracujte online. Archivujte.* [online]. 2018 [cit. 2018-08-11]. Dostupné z: <http://www.capsa.cz/>
- [9] *Alfresco Software and Services | ECM | BPM* [online]. 2018 [cit. 2018-08-11]. Dostupné z: <https://www.alfresco.com/>
- [10] EVROPSKÁ UNIE. Nařízení Evropského parlamentu a Rady (EU) 2016/679 ze dne 27. dubna 2016 o ochraně fyzických osob v souvislosti se zpracováním osobních údajů a o volném pohybu těchto údajů a o zrušení směrnice

- 95/46/ES (obecné nařízení o ochraně osobních údajů). In: *Úřední věstník Evropské unie*. Brusel, 2016, ročník 2016, L 119/1. Dostupné také z: <https://eur-lex.europa.eu/legal-content/CS/TXT/PDF/?uri=CELEX:32016R0679>
- [11] GDPR stručně: O ochraně osobních údajů stručně a jasně. *Úřad pro ochranu osobních údajů* [online]. 2013 [cit. 2018-07-19]. Dostupné z: <https://www.uoou.cz/gdpr-strucne/ds-4843/archiv=0&p1=3938>
- [12] *GDPR stručně a jasně* [online]. b.r. [cit. 2018-07-24]. Dostupné z: <https://gdpr.uoou.cz>
- [13] *ČSN EN ISO 9001: Systémy managementu kvality - Požadavky*. b.r.
- [14] *ČSN EN ISO 14001: Systémy environmentálního managementu – Požadavky s návodem pro použití*. b.r.
- [15] *ČSN EN ISO/IEC 17065: Posuzování shody - Požadavky na orgány certifikující produkty, procesy a služby*. 2013. b.r.
- [16] *ČSN EN ISO/IEC 27001: Informační technologie - Bezpečnostní techniky - Systémy řízení bezpečnosti informací - Požadavky*. b.r.
- [17] *COV FD ČVUT Příručka jakosti pro certifikaci výrobků*. In: . Praha, 2016, IN 2008/1.
- [18] *COV FD ČVUT Pravidla pro řízení dokumentů a záznamů*. In: . Praha, 2015, POSTUP IN 2008/17.
- [19] SOMMERVILLE, Ian. *Softwarové inženýrství*. 1. vyd. Brno: Computer Press, 2013. ISBN 978-80-251-3826-7.
- [20] TROELSEN, Andrew. *Pro C# 5.0 and the .NET 4.5 framework*. Sixth edition. New York, NY: Apress, 2012. ISBN 9781430242338.

- [21] NATHAN, Adam a Daniel. LEHENBAUER. *Windows presentation foundation unleashed*. Indianapolis, Ind.: Sams, 2007. ISBN 978-0672328916.
- [22] NATHAN, Adam. *XAML unleashed*. Indianapolis, IN: Sams Pub., 2014. ISBN 9780672337222.
- [23] DAVID, Mauricio. LiteDB: .NET NoSQL Document Store. *Mbdavid/LiteDB Wiki · GitHub* [online]. 2018 [cit. 2018-05-13]. Dostupné z: <https://github.com/mbdavid/LiteDB/wiki>
- [24] DAJBYCH, Václav. MVVM: Model-View-ViewModel. *DotNETPortal: Největší český web zaměřený na .NET framework* [online]. 2018 [cit. 2018-08-11]. Dostupné z: <https://www.dotnetportal.cz/clanek/4994/MVVM-Model-View-ViewModel>