

UNIVERZITA PARDUBICE

Fakulta elektroniky a informatiky

BAKALÁŘSKÁ PRÁCE

2022

Viktor Homolka

Univerzita Pardubice
Fakulta elektroniky a informatiky

Aplikační využití API k databázi ekonomických subjektů ČR
Bakalářská práce

2022

Viktor Homolka

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2019/2020

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Viktor Homolka**
Osobní číslo: **I17244**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Téma práce: **Aplikační využití API k databázi ekonomických subjektů ČR**
Zadávající katedra: **Katedra informačních technologií**

Zásady pro vypracování

Cílem bakalářské práce bude v teoretické části analyzovat využitelnost dostupných API, poskytujících data o ekonomických subjektech v ČR. Součástí práce bude i přehled rozhraní těchto API, identifikace jejich výhod a nevýhod.

Druhým cílem v praktické části bakalářské práce bude navrhnout a realizovat aplikaci využívající vybrané API, inovativním způsobem poskytující data uživateli dle zadaných kritérií (včetně lokality).

Rozsah pracovní zprávy: **35**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná**

Seznam doporučené literatury:

KYSELA, J., HORÁLEK, J., HOLÍK, F.: Measuring Information Quality of Geosocial Networks. In: New Trends in Intelligent Information and Database Systems. Springer, 2015. ISBN 978-3-319-16211-9. RUSSEL, M., KLASSEN, M.: Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Instagram, GitHub, and More. Sebastopol, USA: O'Reilly Media, 2019. ISBN 978-1-491-98504-5. KYSELA, J. Comparison of Web Applications Geolocation Services. In CINTI 2014 – 15th IEEE International Symposium on Computational Intelligence and Informatics: Proceedings. New York: IEEE (Institute of Electrical and Electronics Engineers), 2014. s. 449-453 s. ISBN 978-1-4799-5338-7.

Vedoucí bakalářské práce: **Ing. Jiří Kysela, Ph.D.**
Katedra informačních technologií

Datum zadání bakalářské práce: **15. listopadu 2019**
Termín odevzdání bakalářské práce: **7. května 2020**

L.S.

Ing. Zdeněk Němec, Ph.D. v.r.
děkan

Ing. Lukáš Čegan, Ph.D. v.r.
vedoucí katedry

Prohlašuji:

Práci s názvem Aplikační využití API k databázi ekonomických subjektů ČR 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., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, 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. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne

Viktor Homolka

PODĚKOVÁNÍ

Rád bych poděkoval svojí rodině za podporu a motivaci při studiu. Dále chci poděkovat programátorům Tomas Claudius Huber a Brian Lagunas jejichž videokurzy pro mě byly na počátku práce velice přínosné.

ANOTACE

Bakalářská práce se zaměřuje na aplikační využití API k databázi ekonomických subjektů ČR v praktickém využití. V práci jsou popsány výhody a nevýhody dostupných API. Hlavním těžištěm práce je programování aplikace fakturace a popis činností vedoucích k její výsledné podobě. Dále pak tvorba návrhu datových struktur a modelu aplikace. Stěžejní funkcionalitou je adresář odběratelů, který čerpá data z uvedeného API.

KLÍČOVÁ SLOVA

API, Databáze, XML, ARES, WPF, C#, Entity Framework

TITLE

Application use of API to the database of economic subjects in the Czech Republic

ANNOTATION

The bachelor thesis focuses on the application use of the API to the database of economic entities of the Czech Republic in practical use. The thesis describes the advantages and disadvantages of available APIs. The main focus of the work is the programming of the invoicing application and a description of the activities leading to its final form. Furthermore, the creation of data structures and application model. The main functionality is the customer directory, which draws data from the API.

KEYWORDS

API, Database, XML, ARES, WPF, C#, Entity Framework

OBSAH

ÚVOD	13
1. Seznámení s pojmy a technologiemi	15
1.1 API	15
1.2 XML	15
1.3 Formát XML	16
1.3.1 XML Dokument	16
1.3.2 Elementy a atributy	17
1.4 XML Parser	18
1.5 XPath	19
1.5.1 Výrazy XPath	19
1.5.2 Syntaxe výrazů XPath	20
1.5.3 Identifikátory osy	20
1.5.4 Podmínky	20
1.6 Relační databáze	21
1.7 MariaDB	22
1.7.1 Vytváření databází a tabulek	23
1.7.2 Vytváření databází	23
1.7.3 Vytváření tabulek	24
1.8 WPF	25
1.9 Entity Framework	26
1.10 LINQ	28
1.11 Aplikační vzor MVVM	29
1.12 REST API	31
1.13 Microsoft Graph API	32
2. Analýza registru ekonomických subjektů	33
2.1 ARES	33
2.1.1 Popis zdrojů aplikace	34
2.1.2 Přístup pomocí metody GET	34
2.1.3 Přístup pomocí metody POST	35
2.1.4 Popis poskytovaných služeb	35
2.2 ADIS	42
2.2.1 Služba pro zjišťování spolehlivosti plátců DPH a bankovních účtů	42
2.2.2 Operace getStatusNespolehlivyPlatce	42
2.2.3 Operace getSeznamNespolehlivyPlatce	44

2.2.4	Operace getStatusNespolehlivyPlatceRozsireny.....	44
3.	Výběr a návrh databáze	45
3.1	Výběr databáze	45
3.2	Návrh databázových tabulek.....	46
3.2.1	Tabulka Company	47
3.2.2	Tabulka CompanyName.....	47
3.2.3	Tabulka Address.....	48
3.2.4	Tabulka User	48
3.2.5	Tabulka Invoice.....	48
3.2.6	Tabulka InvoiceProduct	49
3.2.7	Tabulka Unit.....	49
3.2.8	Tabulka Product	50
3.2.9	Tabulka Currency	50
3.2.10	Ostatní tabulky	51
4.	Realizace aplikace	52
4.1	Dílčí projekty	53
4.2	Realizace databázového datového modelu	54
4.3	Realizace základních tříd aplikace.....	55
4.4	Realizace tříd pro komunikaci s databází	57
4.5	Přihlašovací okno aplikace	58
4.6	Hlavní okno aplikace	59
4.7	Adresář.....	60
4.8	Ceník.....	62
4.9	Fakturace.....	64
4.10	Tiskový report.....	66
4.11	Průzkumník obchodního prostředí.....	68
5.	Komerční řešení	69
5.1	Stormware – POHODA	69
5.2	Elisoft – EKONOM System	69
5.3	Ježek software – DUEL	70
	ZÁVĚR.....	71
	POUŽITÁ LITERATURA.....	73

SEZNAM ILUSTRACÍ A TABULEK

Obrázek 1: Příklad jazyka XML	17
Obrázek 2: Logo MariaDB Foundation	22
Obrázek 3: Entity Framework v modelu aplikace.....	26
Obrázek 4: Diagram vzoru MVVM	30
Obrázek 5: Propojenost Microsoft Graph	32
Obrázek 6: XML struktura odpovědi na dotaz pomocí služby Standard	37
Obrázek 7: Graf dostupnosti registru ARES	38
Obrázek 8: XML struktura odpovědi na dotaz pomocí služby Basic	40
Obrázek 9: Rozbalené elementy ROR a RRZ.....	41
Obrázek 10: Relační návrh v Oracle SQL DataModeller	46
Obrázek 11: UML Diagram základních tříd	55
Obrázek 12: Přihlašovací okno aplikace	58
Obrázek 13: Hlavní okno aplikace bez otevřených modulů	59
Obrázek 14: Detail subjektu v adresáři	61
Obrázek 15: Přehled ceníku	62
Obrázek 16: Detail ceníkové položky	63
Obrázek 17: Přehled faktur	64
Obrázek 18: Položky faktury.....	65
Obrázek 19: Tiskový report.....	66
Obrázek 20: Okno pro přihlášení k účtu Microsoft.....	67
Obrázek 21: Průzkumník obchodního prostředí.....	68

SEZNAM ZKRATEK A ZNAČEK

API	Application Programming Interface
ARES	Administrativní registr ekonomických subjektů
ADIS	Automatizovaný daňový informační systém
MFČR	Ministerstvo financí České republiky
XML	Extensible Markup Language
XAML	Extensible Application Markup Language
WPF	Windows Presentation Foundation
SQL	Structured Query Language
LINQ	Language Integrated Query
POCO	Plain Old CLR Object
DDL	Data Definition Language
MVVM	Model-View-ViewModel
ORM	Object-Relational Mapping
EF	Entity Framework
REST	Representational State Transfer
URI	Uniform Resource Identifier

ÚVOD

Informace jsou pro podnikání a uplatnění na trhu důležitým artiklem. Souvisí s tím rychlost jejich získávání a efektivního využití. Jeden z možných způsobů, jakým toho lze dosáhnout je využití již existujících dat, která jsou veřejně dostupná na internetu.

Shromažďováním dat se zabývá komerční i státní sektor. Informace komerční sféry jsou obvykle zpoplatněny, zatímco data státní správy jsou za pomoci veřejných registrů poskytována bezplatně. Se zvyšujícím se tempem digitalizace se bude objem těchto dat nadále navyšovat.

Na trhu existuje mnoho aplikací a řada z nich nevyužívají data státních registrů dostatečně. Tato data mohou posloužit například k rychlému zavedení dodavatelských firem do adresáře klientů. Není nutné je zadávat ručně neefektivním způsobem a zároveň zvyšovat riziko nesprávnosti informací.

Cílem této bakalářské práce bude navrhnout desktopový ekonomický fakturační software. Jeho hlavní náplní je evidence klientů, produktů, vytváření a evidence faktur včetně jejich položek. Dalším cílem bude získávat data o ekonomických subjektech ze služby ARES a využít je inovativním způsobem. Aplikace bude kromě standardních funkcí vybavena i funkcionalitou průzkumu obchodního prostředí a možností ukládat faktury na cloudové úložiště.

V první kapitole bude provedeno seznámení s pojmy a použitými technologiemi. V další kapitole bude provedena analýza registru ekonomických subjektů ARES, zhodnocení výhod a nevýhod jeho vybraných služeb. Následující kapitoly budou věnovány návrhu struktury databázových tabulek, popisu tříd a jednotlivých modulů aplikace.

1. Seznámení s pojmy a technologiemi

1.1 API

Aplikační rozhraní pro programování (Application Programming Interface) je obvykle sada funkcí a procedur, které umožňují vytváření aplikací, které přistupují k datům a funkcím jiných aplikací, služeb nebo operačního systému. Rozhraní API také usnadňují vývoj počítačových programů poskytnutím jeho komponent. [1] Jinými slovy, pokud chceme komunikovat s počítačem, systémem či webovou službou s úmyslem získání informací nebo provedení funkcí, API nám pomůže sdělit, co od rozhraní chceme.

1.2 XML

Rozšiřitelný značkovací jazyk (eXtensible Markup Language) definovaný W3C konsorciem. W3C (World Wide Web Consortium) je mezinárodní sdružení, jehož hlavním cílem je dohled nad vývojem webových technologií. Jazyk XML byl odvozen ze staršího standardu SGML (Standard Generalized Markup Language). Z SGML byl rovněž vyvinut známý jazyk HTML (Hypertext Markup Language). [2] XML je jednoduchý, velmi flexibilní textový formát navržený pro uchování a výměnu dat tak, aby byl čitelný pro počítač i člověka. [3] Slovo „rozšiřitelný“ znamená, že jazyk lze rozšířit na libovolný počet symbolů podle požadavku uživatele. Sady XML značek mohou být součástí definice XML dokumentu, specifikované odkazem nebo dohodnuty předem. [2]

Podobným značkovacím jazykem je HTML, který oproti XML používá předem definované značky a je určen pro prezentaci dat, nikoliv pro jejich přenos a uskladnění. XML je díky své flexibilitě a jednoduchosti využíván jinými standardy, jako například Microsoft Open XML, OpenDocument, XHTML či SVG. [4]

1.3 Formát XML

XML slouží pro reprezentaci a přenos obecných dokumentů. Autoři z konsorcia W3C se při návrhu XML řídili těmito principy:

- „formát XML musí být použitelný v rámci internetu,
- formát XML by měl podporovat širokou škálu aplikací,
- formát XML musí být kompatibilní s formátem SGML,
- musí být snadné vytvářet programy, které manipulují s dokumenty v XML,
- množství variant XML by mělo být minimální – nejlépe žádné,
- XML dokumenty by měly být čitelné a pochopitelné i pro člověka.“ [2]

S ohledem na stanovené principy navrhli definici XML, která má dvě části:

- XML dokument,
- XML procesory. [2]

Pro zápis a označení různých částí dokumentu slouží značky. Značky mají podobu zápisu do špičatých závorek. První částí je otevírací závorka (start-tag), například `<jazyk>`, a druhou částí je uzavírací závorka (end-tag), například `</jazyk>`. Mezi těmito částmi se nachází text. V případě, že je text mezi otevírací a uzavírací závorkou prázdný, je možné tuto dvojici nahradit prázdným elementem (empty-element), například `<jazyk/>`. [2]

1.3.1 XML Dokument

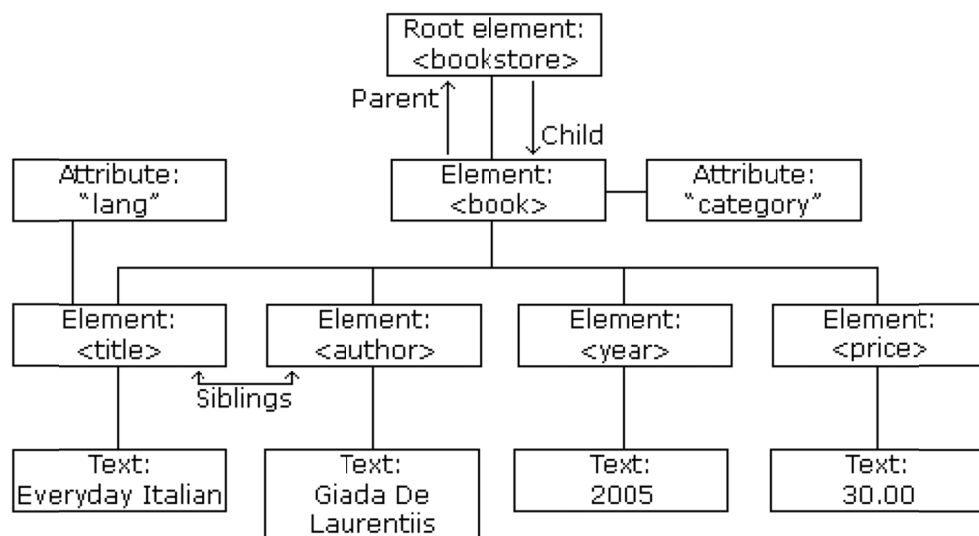
Představuje uspořádanou posloupnost prvku nazývaných entity. Každá entita může být vytvořena ze znaků jisté jazykové sady, implicitně se předpokládá Unicode. XML důsledně rozlišuje velká a malá písmena. Tím se značně liší například od jazyka HTML. XML značky určují logickou strukturu dokumentu a tím i jeho rozložení (layout).

První entita každého dokumentu se označuje jako kořen (root). Dokument se skládá z prologu, elementů, deklarací, komentářů a případně instrukcí pro zpracování jinými aplikacemi. Deklarace slouží pro kontrolu, rozpoznání entit a kontrolu správnosti dokumentu. Správně vytvořený XML dokument představuje stromovou datovou strukturu. [2]

1.3.2 Elementy a atributy

XML dokument tvoří jeden nebo více elementů ohraničených závorkami. Každý element je definován jménem v těchto závorkách. Pokud se nejedná o prázdný element, tvoří jeho obsah entity uzavřené mezi závorkami start-tag a end-tag. Element může obsahovat libovolný text, reference na jiné entity, sekci CDATA, instrukce pro zpracování aplikacemi či komentáře. Každý element může obsahovat atributy, které ho blíže popisují.

Atributy lze vkládat do otevíracích závorek v podobě dvojice hodnot. První z nich je klíč a druhá hodnota. Hodnoty atributů je nutné uvádět do dvojíých uvozovek nebo apostrofů. Formálním obsahem je řetězová hodnota, nicméně ji lze interpretovat také jako číslo. [2]



Obrázek 1: Příklad jazyka XML, převzato z [4]

Textový zápis pro výše uvedený obrázek bude následující:

V obrázku nejsou uvedeny hodnoty atributů, proto bude pro příklad použit řetězec "text".

```
<bookstore>
  <book category="text">
    <title lang="text">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
</bookstore>
```

1.4 XML Parser

XML parsery umožňují číst, zapisovat a manipulovat s XML dokumenty. Může se jednat o načtení XML dokumentu a jeho převodu do objektu stromové struktury, kterou lze jednoduše procházet. Rovněž kontrolují jeho konzistenci a zajišťují i jeho validaci. [5]

XML parsery lze dělit do tří kategorií podle toho, jak zpracovávají XML dokument:

- DOM-based (DOM – Document Object Model),
- SAX-based (SAX – Simple API for XML),
- Pull-model. [5]

DOM-based parsery jsou založeny na doporučení W3C konsorcia a jsou pravděpodobně jedny z nejvíce používaných. Někdy jsou také označovány jako tree-based parsery. Umožňují XML dokument číst, ale i upravovat. Rovněž dovolují přistupovat k jakýmkoliv prvkům ve stromové struktuře XML dokumentu. Pro musí být celý XML dokument nejdříve nahrán do paměti a až poté je možné s ním provádět různé akce. To sebou nese větší požadavky na paměť, což může být v některých případech jejich nevýhoda.

SAX-based parsery nečtou celý XML dokument a nenačítají ho celý do paměti. Namísto toho prochází a čtou sériově celý XML dokument odshora dolů. Pokud najdou požadovaná data, vyvolají patřičnou událost, na kterou je možné reagovat. Jejich výhodou je vysoká rychlost a paměťová nenáročnost. SAX parsery jsou typu read-only, to znamená, že mohou data pouze číst. Nemůžeme pomocí nich provádět žádné úpravy. Jsou užitečné v případech, kdy je nutné procházet obsáhlé XML dokumenty. Tento typ parserů je rovněž označován jako event-based.

Pull-model parsery sekvenčně iterují skrze XML dokument. Používají návrhový vzor Iterator. Prochází každou část dokumentu, jako jsou elementy a atributy. Přitom lze zjistit typ položky a číst její atributy. [5]

Parsery také mohou být klasifikovány jako validující nebo nevalidující. Validující parsery ověřují XML dokument s určitými pravidly DTD (Document Type Definition). Říkají, zda je dokument validní a dobře formátován. Zatímco nevalidující parsery se ověřováním nezabývají. [5]

.NET Framework pro tento účel nabízí celou řadu programových tříd.

Patří mezi ně například:

- XmlReader (Pull-model),
- XmlWriter (SAX-based),
- XmlDocument – podporuje LINQ to XML (DOM-based),
- XmlDocument (DOM-based). [6]

1.5 XPath

XML Path Language je jazyk určený pro vyhledávání elementů a atributů v hierarchické struktuře XML dokumentu. Nahlíží na XML dokument jako na strom uzlů. Rozeznává uzly prvků, atributů a textů. Název uzlu je složen ze dvou částí, místní a oborů názvů, která nemusí být uvedena. Každá implementace XPath musí obsahovat základní sadu funkcí, které přijímá nula nebo více argumentů.

1.5.1 Výrazy XPath

Jednou z nejdůležitějších konstrukcí jazyka je cesta k XML uzlu. Obsahuje několik kroků, přičemž každý z nich sestává z identifikátoru osy, testu uzlu a podmínky. Povinný je ale pouze test uzlu. Jednotlivé kroky výrazu se pak spojují lomítky a vyhodnocení probíhá zleva doprava. Na počátku zpracování se vždy vychází od aktuálního prvku. Výsledkem pak může být libovolný typ uzlu. Množinu vybraných uzlů lze omezit podle testu uzlu, kterým může být například název elementu, přičemž se zohledňují dodatečné podmínky. Uzly, které vyhodnocením úspěšně prošly, postupují do dalšího kroku. V něm se pak budou postupně zpracovávat všechny uzly, které do něj postoupily. [7]

1.5.2 Syntaxe výrazů XPath

Řada vyhledávacích znaků je stejná jako u adresářové struktury.

Pro vyhledání žádaných elementů jsou definovány klíčové znaky:

- / – Odděluje jednotlivé kroky cesty, pokud je jedno lomítka na začátku výrazu, znamená to, že výraz je vztažen ke kořeni dokumentu.
- // – Dvě lomítka bezprostředně za sebou slouží k překonání víceúrovňové struktury. Pokud jsou dvě lomítka na začátku výrazu, opět se jedná o absolutní cestu od kořene dokumentu, takže lze snadno vybrat libovolný element odkudkoli.
- . – Nahrazuje odkaz na sama sebe.
- .. – Odkaz na rodiče aktuálního uzlu.
- @ – Vyhledání atributu, ekvivalentní zápis *attribute::*.
- * – Vybere všechny podřízené prvky uzlu. [7]

1.5.3 Identifikátory osy

Určuje směr procházení XML dokumentu a říká, odkud se budou uzly k vyhodnocení vybírat.

- child:: – přímí potomci aktuálního uzlu,
- descendant:: – všichni potomci aktuálního uzlu,
- descendant-or-self:: – aktuální uzel a všichni potomci,
- self:: – aktuální uzel,
- ancestor-or-self:: – aktuální uzel a všichni jeho předci,
- ancestor:: – všichni předci aktuálního uzlu,
- parent:: – rodič aktuálního uzlu. [8]

1.5.4 Podmínky

Podmínky lze zapisovat do hranatých závorek, které omezují výsledek předchozího vyhodnocení. Pokud místo podmínky zapíšeme do závorek pouze číslo, vybere se uzel na dané pozici.

- //oddeleni[@navez="abc"] – vyhledá všechny elementy oddělení s atributem název odpovídající výrazu,
- /[8] – vezme čtvrtého potomka aktuálního elementu.

1.6 Relační databáze

Databáze je uspořádaná struktura dat obvykle uložených v elektronické podobě. Je tvořena sadou tabulek, které mezi sebou mají určité vztahy. Každá tabulka je uspořádána jako sada sloupců a řádků. Tabulky slouží k uchování různých druhů dat. Každý sloupec má určitý typ, který určuje, jaká informace se v něm nachází (číslo, textový řetězec, čas, ...). Řádky v tabulce představují množinu souvisejících hodnot jednoho objektu nebo entity a jsou identifikovány pomocí jednoznačného údaje nazývaného se primární klíč. Tabulky jsou mezi sebou vzájemně provázány pomocí cizích klíčů. [9]

Relační databázový systém (RDBMS)

Relační systém správy databáze (Relational Database Management System).

Jedná se o software, který provádí dotazy na data včetně přidávání, aktualizace a vyhledávání hodnot. Stará se také o efektivní a spolehlivé ukládání informací, ochranu dat před ztrátou a schopnost poskytnout data v přijatelném čase. RDBMS může také poskytovat vizuální reprezentaci dat. Používají SQL jako jejich standardizovaný jazyk. [10]

Mezi nejznámější DBMS systémy patří:

- MySQL,
- Microsoft SQL Server,
- Oracle Database,
- MariaDB.

1.7 MariaDB

MariaDB je jednou z nejpopulárnějších relačních databází s otevřeným zdrojovým kódem. Počátek vývoje relační databáze MySQL sahá do roku 1995. Jejími zakladateli jsou Michael “Monty“ Widenius a David Axmark, kteří ve Švédsku založili společnost MySQL Ab (o rok později MySQL Inc). Software tvořili pod licencí GNU GPL. V roce 2008 došlo k převzetí společností Sun Microsystems a následně o rok později přestal být zdrojový kód otevřený.

Později Monty Widenius zakládá další firmu. K tomu došlo, protože byl nespokojen s odkoupením Sun Microsystems společností Oracle. Nová společnost se jmenovala Monty Program Ab a vyvinula softwarovou větev databázového systému s názvem MariaDB. V říjnu roku 2013 dochází k programovému sloučení s projektem SkySQL a celá společnost se přejmenovává na MariaDB Ab. Nyní je software spravován nadací MariaDB Foundation, která nemůže být odkoupena žádnou firmou. [11]

Databáze MySQL či MariaDB jsou často využívány webovými stránkami, které požadují rychlou a stabilní databázi. Tyto databáze využívají i populární weby jako Google, Amazon, Facebook či Wikipedia. [11] MariaDB je využívána ve většině cloudových řešení a je výchozí databázovou službou v celé řadě Linuxových distribucí, například RedHat Enterprise Linux či Fedora Linux. MariaDB je k dispozici pod licencí GPL v2.0 (General Public License).

MariaDB je nabízena ve dvou variantách. První z nich Community Server je komunitní verze, která je zdarma ke stažení. Druhá varianta je MariaDB Enterprise Server, který má nahradit proprietární (uzavřené) databáze, například Microsoft SQL či Oracle SQL. [12]



Obrázek 2: Logo MariaDB Foundation, převzato z [13]

1.7.1 Vytváření databází a tabulek

Abychom byli schopni manipulovat s daty, musí být jako první vytvořena databáze. Databáze je v podstatě kontejner pro tabulky. Vytváření tabulek je složitější a nabízí mnoho možností. Před vytvořením tabulek si musíme určit jejich datovou strukturu. Například: počet sloupců, datové typy jednotlivých sloupců, jak budou data indexována pro rychlejší přístup a další. [11]

Obvykle vycházíme z modelu datové analýzy, který specifikuje:

- Počet tabulek a jejich názvy,
- Názvy tabulkových sloupců,
- Typy jednotlivých datových atributů.

1.7.2 Vytváření databází

Vytvoření databáze je velice jednoduché. Pro všechny databázové operace slouží jazyk SQL (Structured Query Language). Pro vytvoření databáze používáme SQL příkaz CREATE DATABASE s parametrem názvu databáze. [11]

Pro vytvoření databáze s názvem `invoiceman`, je nutné provést následující příkaz:

```
CREATE DATABASE invoiceman
```

Případně se stejným výsledkem:

```
CREATE SCHEMA invoiceman
```

Vytvořením databáze nevznikají žádná uživatelská data, ale pouze místo pro budoucí tabulky. Při vytváření databáze můžeme přidat pár dalších možností, například: výchozí znakovou sadu, způsob řazení či porovnávání dat. Musíme také myslet na to, že každý příkaz, který je rozložený na dva a více řádků, je nutné ukončit středníkem. [11]

Příklad vytvoření databáze s názvem invoiceman:

```
CREATE DATABASE invoiceman
CHARACTER SET utf8mb4
COLLATE utf8mb4_0900_ai_ci;
```

Pro nastavení nově vytvořené databáze jako výchozí slouží následující příkaz:

```
USE invoiceman
```

Při vynechání tohoto příkazu bychom museli při každém SQL dotazu specifikovat, k jaké databázi se daný příkaz vztahuje.

1.7.3 Vytváření tabulek

Pro vytvoření tabulky používáme příkaz CREATE TABLE s parametrem názvu. Dále do závorek udáváme názvy jednotlivých sloupců spolu s jejich typy. Název sloupce může být dlouhý až 64 znaků. Názvem může být jakékoliv slovo kromě jistých rezervovaných klíčových výrazů. Tato rezervovaná slova lze přesto použít, pokud je uvedeme do uvozovek. Nicméně jejich použití není nedoporučeno, protože tyto názvy mohou být pro uživatele nepřehledné.

Sloupec, který bude uchovávat klíčové hodnoty, lze označit výrazem PRIMARY KEY. Primární klíč hraje důležitou roli při vytváření relací mezi tabulkami. Spolu s ním bývá často používán další klíčový výraz a tím je AUTO_INCREMENT. Jeho použitím bude docházet k automatickému navyšování hodnoty. [11]

Příklad vytvoření tabulky s názvem Address:

```
CREATE TABLE Address (
  Id INT AUTO_INCREMENT PRIMARY KEY,
  City VARCHAR(80),
  Street VARCHAR(80),
  Zip VARCHAR(16));
```


1.8 WPF

WPF (Windows Presentation Foundation) je softwarová struktura od společnosti Microsoft pro tvorbu formulářových aplikací. Je součástí balíčku .NET Framework od verze 3.0 vydané 21. listopadu 2006.

WPF disponuje mnoha formulářovými komponentami s širokou možností stylování. WPF zavádí také definici formulářů pomocí jazyka XML, konkrétně XAML (Extensible Application Markup Language). XAML je do určité míry podobný s HTML, které rovněž definuje vzhled uživatelského rozhraní. XAML přináší lepší oddělení prezentace a logiky. Navíc disponuje podporou tzv. bindingů, které umožňují propojit vlastnosti objektů přímo s ovládacími prvky formuláře. Určitou nevýhodou se může jevit nutnost znalosti dalšího jazyka. XAML umožňuje vytvářet graficky bohaté ovládací prvky. Můžeme například vkládat obrázky do tlačítek, comboboxů, zprůhledňovat prvky nebo dokonce jednoduše animovat pomocí tzv. storyboardů. WPF aplikace rovněž podporují dotyková gesta. [14]

Mezi hlavní přednosti technologie WPF patří:

Vysoký stupeň integrace – Dříve musel být vývojář pro tvorbu bohatých aplikačních oken s 3D, 2D, audiem a videem vybaven znalostí mnoha různých technologií. WPF tyto technologie integruje do konzistentního programového rozhraní tak, aby práce s nimi byla co možná nejjednodušší.

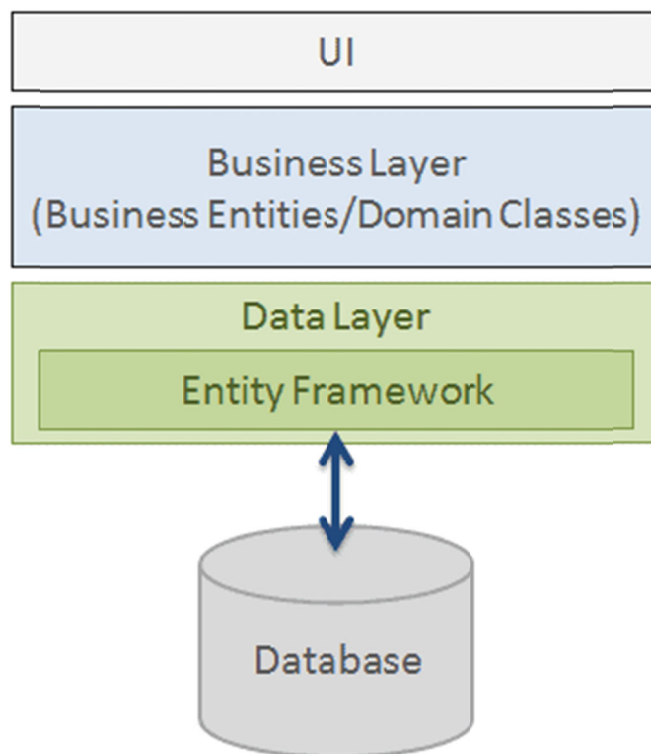
Nezávislost na rozlišení – Technologie WPF zavádí jednotku délky, tzv. DIP (Device Independent Pixel) a čistě vektorovou grafiku, aby výsledné zobrazení formulářů aplikace mělo odpovídající rozměry na každém zařízení. Tím se značně liší od starší technologie WF (Windows Forms), kde je problematické přizpůsobit okna aplikace rozlišení obrazovky a měřítku nastaveného v systému.

Hardwarová akcelerace – WPF používá pro vykreslování formulářů rozhraní Direct3D, jež je jednou z komponent sady DirectX. Ať už je obsah v aplikaci WPF 2D, 3D, grafika nebo text, dochází k výpočtu pomocí grafické procesorové jednotky GPU namísto jednotky CPU. Během zpracování grafického obsahu dochází k hardwarové akceleraci a tím je zajištěno rychlejší zpracování a svižnost aplikace.

Vysoká možnost přizpůsobení – Ovládací prvky WPF mohou být sestaveny způsobem, který ve starších technologiích nebyl možný. Například ComboBox naplněný animovanými tlačítky. Vývojáři nejsou nuceni psát velké množství kódu k přizpůsobení ovládacích prvků. Rovněž je podporována i funkce změny motivů (Themes). Podle motivu v systému může být upraven vzhled aplikace. [15]

1.9 Entity Framework

Entity Framework je multiplatformní relační mapovač ORM (Object Relational Mapper) od společnosti Microsoft s otevřeným zdrojovým kódem. Byl navržen tak, aby byl lehký, rozšiřitelný a byl podporován napříč platformami. Objektově relační mapování je technika, která nahlíží na data v podkladových databázových tabulkách jako na objekty tříd. Entity Framework dovoluje vývojářům pracovat s daty na vyšší úrovni abstrakce a mohou tak vytvářet a udržovat datově orientované aplikace s menším objemem kódu ve srovnání s tradičními aplikacemi.



Obrázek 3: Entity Framework v modelu aplikace, převzato z [16]

Podle výše uvedeného obrázku jsou ukládaná a načítaná data automaticky převáděna na objekty tříd, se kterými již aplikace pohodlně pracuje. Obchodní vrstva (Business Layer) popisuje vztahy mezi datovými entitami, které se v modelu nazývají obchodní entity (Business Entities).

Několik příkladů vztahů mezi entitami:

- faktura – položky faktury,
- faktura – měna,
- firma – název firmy,
- firma – adresa firmy,
- produkt – kategorie,
- položky faktury – produkt.

Základní funkce Entity Framework

- Modelování – Vytvoří datový model na základě POCO (Plain Old CLR Object) entit.
- Dotazování – EF nám umožňuje používat dotazy LINQ (C#/VB.NET) k načítání dat z databáze. Zprostředkovatel připojení k databázi překládá dotazy do dotazovacího jazyka konkrétní databáze. EF nám také umožňuje spouštět SQL dotazy přímo.
- Sledování změn – EF sleduje změny hodnot (vlastností) entit.
- Ukládání – EF provádí příkazy INSERT, UPDATE nebo DELETE do databáze na základě provedených změn v entitách.
- Souběžnost – EF při kolizní manipulaci s daty používá metodu optimistické souběžnosti. Při konkurenční změně dat upozorní na změny provedené jiným uživatelem.
- Transakce – EF provádí automatickou správu transakcí.
- Ukládání do mezipaměti
- Migrace – EF poskytuje sadu příkazů pro migraci, což zahrnuje vytvoření kompletní databáze včetně tabulek a indexů, rovněž umožňuje provést tzv. Scaffolding (přegenerování databázových tabulek do POCO). [16]

1.10 LINQ

LINQ (Language INtegrated Query) je dotazovacím jazykem, který je přímo integrován do syntaxe jazyka C# již od verze 3.0 a .NET Framework 3.5. LINQ má podobný zápis jako jazyk SQL. Je zapisován deklarativně, to znamená, že vývojář popisuje, jaká data potřebuje a ne jak mají být získána.

LINQ může být použit k jednoduchému výběru dat z kolekcí, XML dokumentů, relačních databází a dalších zdrojů dat. Výhodou je kontrola syntaxe v době překladu programu. Pokud chceme využít jeho funkcionalit, musíme zahrnout sestavení System.Linq. Zpracování probíhá v novějších verzích ve více vláknech, což zvyšuje efektivitu. [17]

1.11 Aplikační vzor MVVM

MVVM (Model-View-ViewModel) je softwarový návrhový vzor, který je strukturován tak, aby oddělil logiku programu od vizuální části uživatelského rozhraní. Pomáhá organizovat kód, rozdělit programové komponenty do modulů, aby byl vývoj jednodušší a rychlejší. Uspodňuje testovatelnost, znovupoužitelnost kódu a zajišťuje datové vazby.

Architektura MVVM je doporučována společností Microsoft při použití technologie WPF. Výkonná podpora datových vazeb s pomocí rozhraní `INotifyPropertyChanged`, `ICommand` a `IValueConverter`. [18]

Model

Model představuje uživatelská data, například společnost či faktura. Definuje sadu tříd s vlastnostmi, která uchovávají data. Máme možnost vytvořit modelové třídy ručně nebo je nechat vygenerovat za pomoci ORM, například Entity Framework. Model je rovněž zodpovědný za správu dat a zapouzdření obchodní logiky. Modelové třídy obvykle poskytují mechanismy pro ohlášení změn dat, například za pomoci `INotifyPropertyChanged`. Pro ohlášení chyb používají rozhraní `IDataErrorInfo` nebo `INotifyDataErrorInfo`. [19]

View

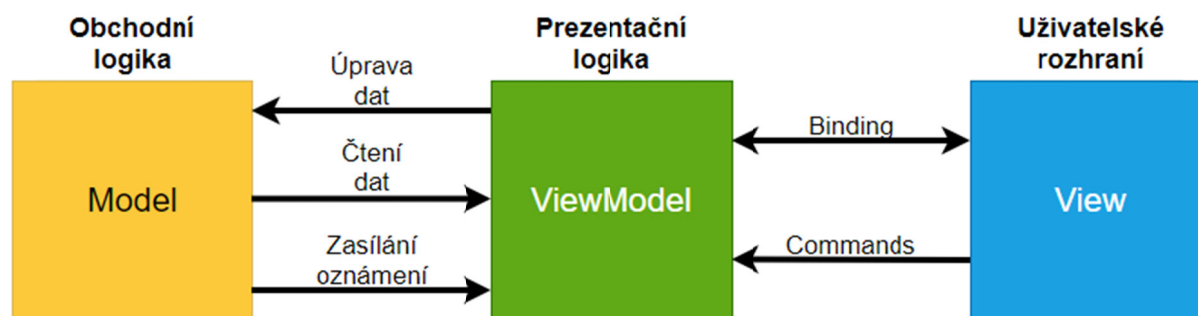
Představuje vizuální část uživatelského rozhraní aplikace. Shromažďuje veškeré prvky uživatelského rozhraní okna, navigační části, vlastní uživatelské prvky, soubory prostředků, stylů a motivů. Reprezentuje modelová data a interaguje s uživatelem (vstup z klávesnice, myši a dotyková gesta) a jejich zpracování předává viewmodelu prostřednictvím datové vazby.

View vystavuje vlastnost `DataContext`, která je nastavena na objekt `ViewModelu`. V code-behind (kód v pozadí uživatelského prvku) je obvykle pouze inicializace obsažených ovládacích prvků a žádný kód logiky. Jeden View může komunikovat pouze s jedním `ViewModelu`. [19]

ViewModel

`ViewModel` (model zobrazení) je nevizuální třída sloužící pro komunikaci mezi View a Model. `ViewModel` neví o zobrazení přímo, vazba probíhá prostřednictvím nějaké vlastnosti. Modelové třídy jsou k němu připojeny přímo.

Vystavuje data jako vlastnosti a akce jako metody (nebo příkazy) pro konkrétní pohled (view). ViewModel je závislý na View, nicméně na něj nesmí sám odkazovat. Vzhledem k nezávislosti na View může být jednoduše testován. Obvykle také implementuje rozhraní `INotifyPropertyChanged` poskytující ohlášení o změně dat. [19]



Obrázek 4: Diagram vzoru MVVM, zdroj: Autor

Důležitá rozhraní v .NET Framework, která pomáhají implementovat MVVM:

- `INotifyPropertyChanged` – rozhraní pro oznamování změny vlastnosti objektu,
- `INotifyCollectionChanged` – rozhraní pro oznamování změny položek kolekce (například odebrání, přidání a vymazání kolekce),
- `INotifyDataErrorInfo` – rozhraní pro ověření platnosti vstupních dat,
- `IDataErrorInfo` – starší verze výše uvedeného,
- `ICommand` – rozhraní zajišťující vykonání akce pro vizuální prvek (například tlačítko),
- `IValueConverter` – rozhraní poskytující převodní logiku hodnot při použití datových vazeb, může to být převod logické hodnoty na hodnotu výčtu `Visibility`.

1.12 REST API

REST (REpresentational State Transfer) je architektonické rozhraní navržené pro distribuované prostředí. Poprvé tento pojem definoval počítačový vědec Dr. Roy Fielding ve své doktorské práci.

Poskytuje vývojářům vysokou úroveň flexibility a svobody. Je používáné pro jednotný a snadný přístup k datovým zdrojům. Každý takový zdroj má svůj identifikátor URI.

Dnes je jedním z nejrozšířenějších způsobů přístupu k aplikacím či službám. Rozhraní je možné implementovat pomocí prakticky libovolného programovacího jazyka a podporuje různé formáty dat.

Při vývoji je jediným požadavkem, aby rozhraní odpovídalo následujícím šesti principům návrhu REST:

- Jednotné rozhraní – Všechny požadavky API na stejný zdroj by měly vypadat stejně, bez ohledu na to, odkud požadavek pochází.
- Oddělení klienta a serveru – Při návrhu REST API musí být na sobě klientské a serverové aplikace zcela nezávislé. Jedinou informací, kterou by klientská aplikace měla znát, je URI požadovaného zdroje.
- Bezstavovost – REST API jsou bezstavová, což znamená, že každý požadavek musí obsahovat všechny informace nezbytné pro jeho zpracování.
- Možnost ukládat do mezipaměti – Prostředky by měli být uložitelné do mezipaměti na straně klienta nebo serveru, pokud je to možné.
- Architektura vrstveného systému – Rozhraní REST API musí být navrženo tak, aby klient ani server nemohli zjistit, zda komunikují s koncovou aplikací nebo prostředníkem.
- Kód na vyžádání (volitelné) – Rozhraní REST API obvykle odesílají statická data. V určitých případech mohou odpovědi obsahovat také spustitelný kód (jako jsou aplety Java). V těchto případech by měl kód běžet pouze na vyžádání. [20]

Rozhraní REST API komunikují prostřednictvím požadavků HTTP a provádějí standardní databázové funkce, jako je vytváření, čtení, aktualizace a mazání záznamů (CRUD) v rámci prostředku. REST API by například použilo požadavek GET k načtení záznamu, požadavek POST k vytvoření záznamu, požadavek PUT k aktualizaci záznamu a požadavek DELETE k odstranění záznamu. [20]

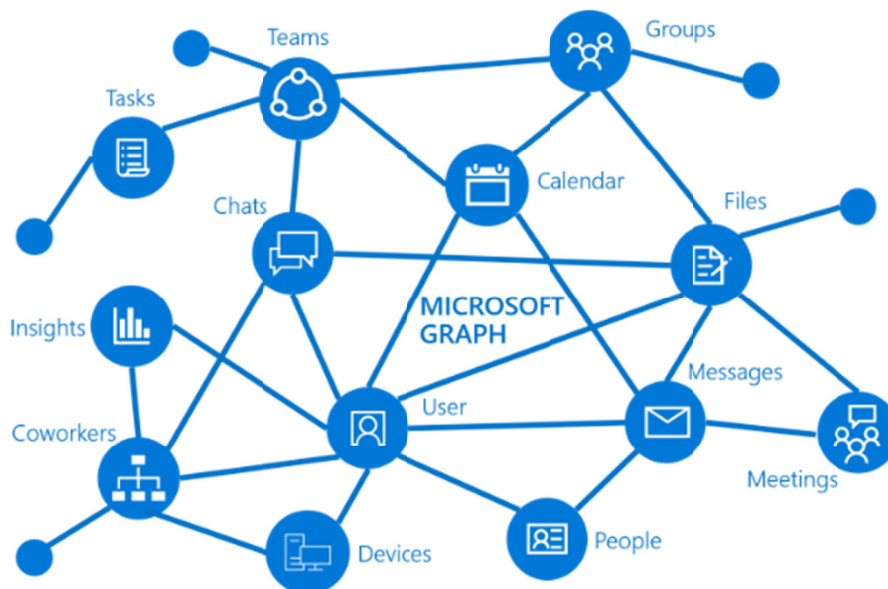
1.13 Microsoft Graph API

Microsoft Graph API je RESTful webové rozhraní API, které umožňuje přístup ke službám Microsoft Cloud. Po registraci aplikace a získání ověřovacích tokenů pro uživatele nebo službu je možné zadávat požadavky na rozhraní Microsoft Graph API. Pomocí něho můžeme vytvářet prostředí kolem kontextu uživatele, který mu pomůže být produktivnější. [21]

Může to být k užítku v aplikacích, kde chceme pomáhat uživateli zefektivňovat jeho práci. Obnášelo by to například pracovat s jeho kalendářem, dokumenty na OneDrive či zprávami.

Microsoft Graph zpřístupňuje REST API a klientské knihovny pro přístup k datům v následujících cloudových službách Microsoftu:

- Služby Microsoft 365: Delve, Excel, Microsoft Bookings, Microsoft Teams, OneDrive, OneNote, Outlook/Exchange, Planner a SharePoint.
- Služby podnikové mobility a zabezpečení: Advanced Threat Analytics, Advanced Threat Protection, Azure Active Directory, Identity Manager a Intune.
- Služby Windows 10: aktivity, zařízení, upozornění.
- Dynamics 365 Business Central. [21]



Obrázek 5: Propojenost Microsoft Graph, převzato z [21]

2. Analýza registru ekonomických subjektů

2.1 ARES

Administrativní registr ekonomických subjektů (ARES) je aplikace Ministerstva financí České republiky (MFČR). Jejím účelem je zpřístupnit údaje z informačních registrů veřejné správy o ekonomických subjektech. ARES přehledně zpřístupňuje údaje přebírané ze zdrojových registrů. Všechny údaje, které ARES zpřístupňuje, mají pouze informativní charakter.

Účelem aplikace je poskytnout rychlé a obecně dostupné informace o jednotlivých ekonomických subjektech. S ohledem na zabezpečení služby Ministerstvo financí stanovuje omezení a znemožní přístup uživatelům, kteří: [22]

- „odešlou k vyřízení více než 10.000 dotazů v době od 8:00 hod. do 18:00 hod.,
- odešlou k vyřízení více než 50.000 dotazů v době od 18:00 hod. do 8:00 hod. rána následujícího dne,
- se snaží o porušení bezpečnostní ochrany www serveru Ministerstva financí,
- opakovaně posílají nesprávně vyplněné dotazy,
- opakovaně posílají stejné dotazy,
- mají větší počet současně zadaných dotazů (pro automatizované XML dotazy),
- obcházejí povolené limity využíváním dotazování z většího množství IP adres,
- automatizovaně propátrávají databázi náhodnými údaji nebo generují většinu nesprávných dotazů“. [22]

ARES obsahuje údaje ze základních (majoritních) zdrojů, které jsou formou odkazů doplněny údaji z dalších zdrojů. Při zpracování se používají též kontrolní zdroje.

2.1.1 Popis zdrojů aplikace

Ares čerpá informace z celé řady registrů a rejstříků, které následně prováže a prezentuje.

„majoritní zdroje

- *Veřejný rejstřík (VR), vedený rejstříkovými soudy*
- *Živnostenský rejstřík (RŽP), vedený Ministerstvem průmyslu a obchodu ČR*
- *Registr ekonomických subjektů (RES), vedený Českým statistickým úřadem*
- *Registr církví a náboženských společností (RCNS), vedený Ministerstvem kultury ČR*
- *Národní registr poskytovatelů zdravotních služeb (NRPZS), vedený Ústavem zdravotnických informací a statistiky ČR*
- *Evidence zemědělského podnikatele (EZP), která je vedena Ministerstvem zemědělství ČR*
- *Seznam politických stran a hnutí (PSH), vedený Ministerstvem vnitra ČR*
- *Rejstřík škol a školských zařízení (RŠ), vedený Ministerstvem školství a tělovýchovy ČR*

další zdroje

- *Registr plátců daně z přidané hodnoty (DPH, SkDPH, údaje o nespolehlivém plátcu DPH), vedený Finanční správou ČR*
- *Registr plátců spotřební daně (SD), vedený Celní správou ČR*
- *Účelový registr organizací systému ARIS (RARIS), vedený Ministerstvem financí ČR*
- *Centrální evidence dotací z rozpočtu (CEDR), která je vedena Ministerstvem financí ČR*
- *Centrální evidence úpadců (CEU), která je vedena Ministerstvem spravedlnosti ČR*
- *Insolvenční rejstřík (IR), který je veden Ministerstvem spravedlnosti ČR*

kontrolní zdroj

- *Registr územní identifikace, adres a nemovitostí (RÚIAN), vedený Českým úřadem zeměměřickým a katastrálním“ [22]*

2.1.2 Přístup pomocí metody GET

„Následující seznam nabízí výběr aplikací, které poskytují data z ARES v XML pro dotazy metodou GET bez nutnosti digitálního podpisu. Název služby je současně příkladem dotazu do příslušné služby.

- *základní dotaz na výpis identifikačních údajů: Standard*
- *dotaz na seznam registrací subjektu ve zdrojích VR, RES a RŽP: Seznam_Reg*
- *základní výpis z více registrů: Basic*
- *výpis dat z Veřejného rejstříku: Vypis_OR*
- *výpis dat z Veřejného rejstříku: Vypis_VR*
- *výpis dat z Registru živnostenského podnikání: Vypis_RZP*
- *výpis dat ze statistického registru RES: Vypis_RES*
- *výpis dat z Registru církví a náboženských společností: Vypis_CNS*
- *výpis dat z Centrální evidence úpadců: Vypis_CEU*
- *přehled ekonomických subjektů*
- *dotaz na standardizovanou adresu*
- *přehled změn ekonomických subjektů*

- výpis dat z Národního registru poskytovatelů zdravotních služeb: *Vypis_RZZ*
- výpis dat ze zemědělského registru: *Vypis_SZR*
- výpis dat z Registru politických stran a hnutí: *Vypis_PSH*
- výpis dat z Rejstříku škol: *Vypis_SKO*
- elektronický opis Veřejného rejstříku: *Vypis_VREO*“ [22]

2.1.3 Přístup pomocí metody POST

„Automatizovaně lze k některým službám přistupovat metodou POST. Počet elementů <Dotaz> v jednom volání je omezen na maximální hodnotu 100.

- základní standardní dotaz: *Standard*
- základní výpis přes více registrů: *Basic*
- Veřejný rejstřík: *Vypis_OR*
- Veřejný rejstřík: *Vypis_VR*
- Registr živnostenského podnikání: *Vypis_RZP*
- statistický RES: *Vypis_RES*
- dotaz na seznam registrací subjektu ve VR, RES, RŽP: *Seznam_Reg*
- dotaz na standardizovanou adresu: *Stdadr*
- přehled změn ekonomických subjektů: *zm*“ [22]

2.1.4 Popis poskytovaných služeb

2.1.4.1 Služba Standard

Služba Standard poskytuje základní informace o subjektech. Údaje jsou získávány ze zdroje, který byl určen jako prioritní. Ve výsledku je uvedeno, z jakého registru data pocházejí. Základními údaji každé odpovědi jsou IČO, obchodní firma, právní forma, datum vzniku, adresa. Element <Datum_vzniku> má v závislosti na typu zdroje jiný význam. V RES a RCNS má význam vzniku či registrace subjektu, ve VR data zápisu a v RŽP význam data registrace 1. živnosti. Element <Adresa_ARES> zahrnuje kompletní specifikaci adresy z RÚIAN (např.: název ulice, číslo domu, PSČ). Pokud v tomto registru není, nepovažuje ji ARES za standardizovanou. Element <Adresa_UIR> obsahuje kompletní údaje z registru RÚIAN (např.: kód kraje, kód okresu).

Vyhledávání na základě parametrů:

- ico,
- obchodni_firma.

Povinný je alespoň jeden z nich. Vyhledávání na základě specifikace obchodní firmy vyžaduje zadání minimálně 3 a maximálně 254 znaků. Lze také vyhledávat pomocí zástupných znaků ? a *.

„Je však nutné dodržet zásadu, že slovo musí obsahovat alespoň 3 po sobě jdoucí významné znaky písmena a číslice, nebo 4 tyto znaky celkem (je přípustné zadat vod, *ody, vo*dy, vo*dy*, dol?, není přípustné vo?y, vo*y). Nevhodné umístění zástupných znaků (např. na začátku textu) může značně zpomalit vyhledávání.*

Pokud název obchodní firmy obsahuje zástupné znaky (, ?), nesmí se tyto použít pro vyhledání dle názvu firmy. Při zadávání obchodního jména tyto znaky vynechejte.“ [22]*

- nazev_obce – Uplatňuje se jen v případě, kdy je zadán i parametr obchodni_firma.
- max_pocet – Číslo v intervalu 1 až 200, určuje maximální počet vrácených subjektů. Výchozí hodnota je 10.
- aktivni – Logická hodnota určuje, zda je subjekt aktivní (true – výchozí/false).
- diakritika – Určuje, zda je možno vyhledávat včetně diakritiky (true – výchozí/false).
- czk – Specifikuje kódování českých znaků v parametrech obchodni_firma a nazev_obce. Možné hodnoty parametru jsou „iso“ (znaková sada ISO-8859-2) a „utf“ (znaková sada UTF-8).

Vyhledávacích parametrů je více, nicméně pro běžné použití postačí výše uvedené.

Příklad odpovědi na dotaz do ARES API služby Standard ve formátu XML ze vstupní adresy s využitím parametrů obchodni_firma a nazev_obce. Červeně jsou vyznačeny elementy, které jsou v aplikaci využívány.

Adresa dotazu API pro metodu GET:

https://wwwinfo.mfcr.cz/cgi-bin/ares/darv_std.cgi?obchodni_firma=alza&nazev_obce=praha

```
▼<are:Ares_odpovedi xmlns:are="http://wwwinfo.mfcr.cz/ares/xml_doc/schemas/ares/ares_answer/v_1.0.1"
  xmlns:dtt="http://wwwinfo.mfcr.cz/ares/xml_doc/schemas/ares/ares_datatypes/v_1.0.4"
  xmlns:udt="http://wwwinfo.mfcr.cz/ares/xml_doc/schemas/udt_datatypes/v_1.0.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" odpoved_datum_cas="2021-07-24T14:12:56" odpoved_pocet="1"
  odpoved_typ="Standard" vystup_format="XML" xslt="klient"
  validation_XSLT="/ares/xml_doc/schemas/ares/ares_answer/v_1.0.0/ares_answer.xsl"
  xsi:schemaLocation="http://wwwinfo.mfcr.cz/ares/xml_doc/schemas/ares/ares_answer/v_1.0.1
  http://wwwinfo.mfcr.cz/ares/xml_doc/schemas/ares/ares_answer/v_1.0.1/ares_answer_v_1.0.1.xsd" Id="ares">
  ▼<are:Odpoved>
    <are:Pocet_zaznamu>1</are:Pocet_zaznamu>
    <are:Typ_vyhledani>FREE</are:Typ_vyhledani>
    ▼<are:Zaznam>
      ▼<are:Shoda_OF>
        <dtt:Kod>7</dtt:Kod>
        </are:Shoda_OF>
        <are:Vyhledano_dle>OF</are:Vyhledano_dle>
      ▼<are:Typ_registru>
        <dtt:Kod>2</dtt:Kod>
        <dtt:Text>OR</dtt:Text>
        </are:Typ_registru>
        <are:Datum_vzniku>2003-08-26</are:Datum_vzniku>
        <are:Datum_platnosti>2021-07-24</are:Datum_platnosti>
      ▼<are:Pravni_forma>
        <dtt:Kod_PF>121</dtt:Kod_PF>
        </are:Pravni_forma>
        <are:Obchodni_firma>Alza.cz a.s.</are:Obchodni_firma>
        <are:ICO>27082440</are:ICO>
      ▼<are:Identifikace>
        ▼<are:Adresa_ARES>
          <dtt:ID_adresy>209990078</dtt:ID_adresy>
          <dtt:Kod_statu>203</dtt:Kod_statu>
          <dtt:Nazev_okresu>Hlavní město Praha</dtt:Nazev_okresu>
          <dtt:Nazev_obce>Praha</dtt:Nazev_obce>
          <dtt:Nazev_casti_obce>Holešovice</dtt:Nazev_casti_obce>
          <dtt:Nazev_mestske_casti>Praha 7</dtt:Nazev_mestske_casti>
          <dtt:Nazev_ulice>Jankovcova</dtt:Nazev_ulice>
          <dtt:Cislo_domovni>1522</dtt:Cislo_domovni>
          <dtt:Typ_cislo_domovni>1</dtt:Typ_cislo_domovni>
          <dtt:Cislo_orientacni>53</dtt:Cislo_orientacni>
          <dtt:PSC>17000</dtt:PSC>
          ▶<dtt:Adresa_UIR>
            ...
          </dtt:Adresa_UIR>
        </are:Adresa_ARES>
      </are:Identifikace>
      <are:Kod_FU>13</are:Kod_FU>
      <are:Priznaky_subjektu>NAAANNNNNNNNNNNNNNNNNNNNNNNNNNNNN</are:Priznaky_subjektu>
    </are:Zaznam>
  </are:Odpoved>
</are:Ares_odpovedi>
```

Obrázek 6: XML struktura odpovědi na dotaz pomocí služby Standard, zdroj: Autor

Shrnutí služby Standard

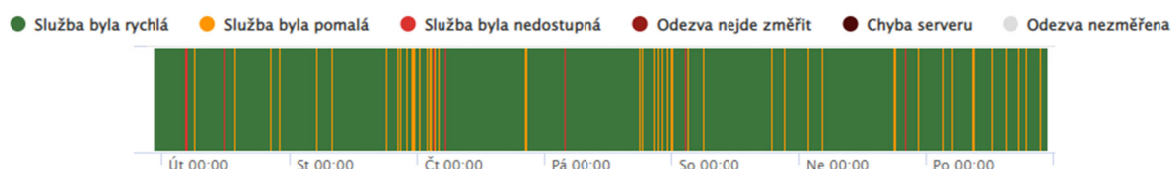
Výhody:

- Lze vyhledávat nejen pomocí IČO, ale i názvu obchodní firmy s upřesněním názvu obce, ve které má firma fakturační adresu.
- Téměř všechny důležité údaje o firmě v jedné odpovědi.
- Lze využívat zástupných znaků, v případě, že neznáme přesný název firmy.

Nevýhody:

- V případech, kdy chceme vyhledat podnikající fyzickou osobu s hojně rozšířeným jménem (například: Petr Novák), nelze dotaz realizovat. To i tehdy, pokud uvedeme do parametru max_pocet maximální možnou hodnotu 200. Následně jsme upozorněni zprávou „Na zadaný Dotaz by se vrátilo více odpovědí, než odpovídá parametru max_pocet. POZOR! Hrozí zablokování Vaší IP adresy!“ [22].
- Parametr max_pocet se může jevit jako nadbytečný, jelikož neomezuje počet vrácených záznamů, ale pouze to, zda služba vrátí data nebo chybu. Uživatel nemůže vědět, kolik výsledků služba vrátí, a proto nedává smysl zadávat jinou hodnotu než 200. Navíc služba musí dotaz v každém případě zpracovat a následně výsledek dle zadaného kritéria max_pocet buď zahodí, nebo vrátí.
- Dílčí nevýhoda může být spatřena v nutnosti dotázat se dalšího nebo jiného API v případě, kdy potřebujeme zjistit také DIČ daného subjektu.
- Doba odezvy je v průběhu dne proměnlivá. Během dne se pohybuje v řádu jednotek sekund. V nočních hodinách dochází k výraznému zpomalení odezvy. Pokud služba odpovídá rychle, doba odezvy je menší než 0,5 s. V případě, že je služba pomalá, doba se pohybuje v rozsahu od jednotek do desítek sekund.

Dostupnost Období od 06.12.2021 21:32 do 13.12.2021 22:30



Obrázek 7: Graf dostupnosti registru ARES, převzato z [23]

2.1.4.2 Služba Basic

Služba Basic poskytuje informace o subjektu ze zdroje, který byl určen jako prioritní. Ve výsledném dotazu jsou velmi podobné informace jako u služby Standard, nicméně jsou zde některé informace navíc, jako např. DIČ, registrace v OR, registrace v RŽP, rozsah počtu zaměstnanců, předměty podnikání, obory činností a několik dalších.

Vyhledávání na základě parametrů:

- ico – Jediný a zároveň povinný parametr, podle kterého lze vyhledávat je IČO.
- xml – Určuje výstupní formát dotazu (0 – xml – výchozí hodnota/(1, 2 – html).
- ver – Ovlivňuje délku názvů elementů. Hodnota „1.0.3“ (zkrácené názvy XML elementů), „1.0.2“ (verze s plnými názvy elementů).
- aktivni – Logická hodnota určuje, zda je subjekt aktivní (true – výchozí/false).

Vyhledávacích parametrů je více, nicméně pro běžné použití postačí výše uvedené.

Příklad odpovědi na dotaz do ARES API ve formátu XML ze vstupní adresy s využitím parametru ico. Červeně jsou vyznačeny elementy, které jsou v aplikaci využívány.

Adresa dotazu API: https://wwwinfo.mfcr.cz/cgi-bin/ares/darv_bas.cgi?ico=27082440

```
<are:Ares_odpovedi xmlns:are="http://wwwinfo.mfcr.cz/ares/xml_doc/schemas/ares/ares_answer_basic/v_1.0.3"
xmlns:D="http://wwwinfo.mfcr.cz/ares/xml_doc/schemas/ares/ares_datatypes/v_1.0.3"
xmlns:U="http://wwwinfo.mfcr.cz/ares/xml_doc/schemas/uvis_datatypes/v_1.0.3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
odpoved_datum_cas="2021-07-24T17:25:48" odpoved_pocet="1" odpoved_typ="Basic" vystup_format="XML" xslt="klient"
validation_XSLT="http://wwwinfo.mfcr.cz/ares/xml_doc/schemas/ares/ares_odpovedi.xsl"
xsi:schemaLocation="http://wwwinfo.mfcr.cz/ares/xml_doc/schemas/ares/ares_answer_basic/v_1.0.3
http://wwwinfo.mfcr.cz/ares/xml_doc/schemas/ares/ares_answer_basic/v_1.0.3/ares_answer_basic_v_1.0.3.xsd Id="ares">
  <are:Odpoved>
    <D:PID>0</D:PID>
    <D:VH>
      <D:K>1</D:K>
      </D:VH>
      <D:PZA>1</D:PZA>
    <D:UVOD>
      ...
    </D:UVOD>
    <D:VBAS>
      <D:ICO zdroj="OR">27082440</D:ICO>
      <D:DIC zdroj="DPH">CZ27082440</D:DIC>
      <D:OF zdroj="OR">Alza.cz a.s.</D:OF>
      <D:DV>2003-08-26</D:DV>
      <D:PF zdroj="OR">
        <D:KPF>121</D:KPF>
        <D:NPF>Akciová společnost</D:NPF>
      </D:PF>
      <D:AD zdroj="ARES">
        <D:UC>Jankovcova 1522</D:UC>
        <D:PB>17000 Praha</D:PB>
      </D:AD>
      <D:AA zdroj="ARES">
        <D:IDA>209990078</D:IDA>
        <D:KS>203</D:KS>
        <D:NS>Česká republika</D:NS>
        <D:N>Praha</D:N>
        <D:NCO>Holešovice</D:NCO>
        <D:NMC>Praha 7</D:NMC>
        <D:NU>Jankovcova</D:NU>
        <D:CD>1522</D:CD>
        <D:TCD>1</D:TCD>
        <D:CO>53</D:CO>
        <D:PSC>17000</D:PSC>
      <D:AU>
        ...
      </D:AU>
      <D:AA>
        <D:PSU>NAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA</D:PSU>
        <D:ROR>
          ...
        </D:ROR>
        <D:RRZ>
          ...
        </D:RRZ>
      <D:KPP zdroj="RES">2000 - 2499 zaměstnanců</D:KPP>
    <D:Nace>
      ...
    </D:Nace>
    <D:PPI>
      ...
    </D:PPI>
    <D:Obory_cinnosti>
      ...
    </D:Obory_cinnosti>
  </D:VBAS>
</are:Odpoved>
</are:Ares odpovedi>
```

Obrázek 8: XML struktura odpovědi na dotaz pomocí služby Basic, zdroj: Autor


```

▼<D:ROR>
  ▼<D:SZ>
    ▼<D:SD>
      <D:K>1</D:K>
      <D:T>Městský soud v Praze</D:T>
    </D:SD>
    <D:OV>B 8573</D:OV>
  </D:SZ>
  ►<D:SOR>
  ...
  </D:SOR>
</D:ROR>
▼<D:RRZ>
  ▼<D:ZU>
    <D:KZU>310007</D:KZU>
    <D:NZU>Úřad městské části Praha 7</D:NZU>
  </D:ZU>
  ▼<D:FU>
    <D:KFU>7</D:KFU>
    <D:NFU>Praha 7</D:NFU>
  </D:FU>
</D:RRZ>

```

Obrázek 9: Rozbalené elementy ROR a RRZ, zdroj: Autor

ROR – registrace v Obchodním rejstříku

RRZ – registrace v Registru živnostníků

Shrnutí služby Basic

Výhody:

- Oproti rozhraní Standard obsahuje Basic informace jako DIČ či kde byla obchodní firma registrována do OR (obchodního rejstříku) a RŽP (registr živnostenského podnikání).
- Poskytuje informaci o přibližném počtu zaměstnanců.
- Uvádí obory činností a předměty podnikání.

Nevýhody:

- Nelze vyhledávat podle názvu obchodní firmy, což nejprve vyžaduje vyhledat subjekt v rozhraní Standard a následně dle nalezeného IČO v rozhraní Basic.
- Je zde rovněž proměnlivá doba odezvy jako u služby Standard.

Informace o předmětech podnikání v elementu PPI a Obory_cinnosti nemá příliš velkou vypovídající hodnotu, jelikož řada firem registruje větší počet činností. Některými registrovanými obory se firma zabývá jen okrajově nebo vůbec.

2.2 ADIS

ADIS (Automatizovaný daňový informační systém) je aplikační rozhraní Finanční správy České republiky. ADIS se skládá z několika částí, jež každá z nich má svůj účel. Patří mezi ně registr daňových subjektů, zpracování daní (daňových přiznání, vyúčtování, hlášení) a systémy pro výběr daní. ADIS je provázán s dalšími státními systémy. Přebírá data z insolvenčního rejstříku, registru obyvatel, registru motorových vozidel a dalších. Poskytuje také informace ze systému státní sociální podpory vedeným ministerstvem práce a sociálních věcí. [24]

2.2.1 Služba pro zjišťování spolehlivosti plátců DPH a bankovních účtů

Daňová správa poskytuje rozhraní webové služby, které umožňuje on-line zjišťování spolehlivosti konkrétních plátců DPH a seznam bankovních účtů plátců DPH.

Služba je provozována na adrese:

<https://adisrws.mfcr.cz/adistc/axis2/services/rozhraniCRPDPH.rozhraniCRPDPHSOAP>.

Rozhraní pro dotazování do registru využívá metody **POST**.

2.2.2 Operace getStatusNespolehlivyPlatce

Základní operace, kterou rozhraní poskytuje.

Vstupní parametry:

- dic – DIČ subjektu, minimálně 1 a maximálně 100

Výstupem pro každé z dotazovaných DIČ je stav nespolehlivosti plátce, seznam evidovaných bankovních účtů společně s datem zveřejnění v registru a číslo finančního úřadu. Pokud je plátce nespolehlivý, je uvedeno i datum uveřejnění nespolehlivosti.

Výstupní hodnoty stavu nespolehlivosti plátce:

- ANO – plátce DPH je nespolehlivý,
- NE – plátce DPH je spolehlivý,
- NENALEZEN – DIČ nebylo nalezeno.

Odpověď dále obsahuje datum jejího vygenerování, atributy statusCode a statusText, jež mohou nabývat následujících hodnot:

- 0 – OK – Vše je v pořádku. Odpověď bude obsahovat informace k dotazovaným DIČ.
- 1 – OK – Příliš mnoho DIČ v požadavku. Vrátí prvních 100 dotazovaných subjektů.
- 2 – Technologická odstávka služby – odstávka probíhá v čase 0:00 až 0:10.
- 3 – Služba nedostupná – odstávka služby mimo čas 0:00 až 0:10.

Příklad volání služby getStatusNespolehlivyPlatce:

```
„<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <StatusNespolehlivyPlatceRequest xmlns="http://adis.mfcr.cz/rozhraniCRPDPH/">
      <dic>445522999</dic>
      <dic>8855229999</dic>
      <dic>7755229999</dic>
      <dic>010101010</dic>
    </StatusNespolehlivyPlatceRequest>
  </soapenv:Body>
</soapenv:Envelope>“ [25]
```

Příklad odpovědi služby getStatusNespolehlivyPlatce:

```
„<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <StatusNespolehlivyPlatceResponse xmlns="http://adis.mfcr.cz/rozhraniCRPDPH/">
      <status odpovedGenerovana="2013-03-07" statusCode="0" statusText="OK" />
      <statusPlatceDPH cisloFu="451" datumZverejneniNespolehlivosti="2013-03-02"
dic="445522999" nespolehlivyPlatce="ANO" />
      <statusPlatceDPH cisloFu="13" dic="8855229999" nespolehlivyPlatce="NE">
        <zverejneneUcty>
          <ucet datumZverejneni="2013-03-01">
            <standardniUcet cislo="123456789" kodBanky="0800" />
          </ucet>
          <ucet datumZverejneni="2013-02-01">
            <standardniUcet cislo="9876543210" kodBanky="0800" predcisli="158"/>
          </ucet>
          <ucet datumZverejneni="2013-02-01">
            <nestandardniUcet cislo="AL47212110090000000235698741" />
          </ucet>
        </zverejneneUcty>
      </statusPlatceDPH>
      <statusPlatceDPH cisloFu="451" dic="7755229999" nespolehlivyPlatce="NE" />
      <statusPlatceDPH dic="010101010" nespolehlivyPlatce="NENALEZEN" />
    </StatusNespolehlivyPlatceResponse>
  </soapenv:Body>
</soapenv:Envelope>“ [25]
```

2.2.3 Operace `getSeznamNespolehlivyPlatce`

Operace pro zjištění všech evidovaných nespolehlivých plátců.

Nemá žádné parametry.

Výstupem je seznam všech aktuálně evidovaných nespolehlivých plátců DPH. Struktura odpovědi je totožná jako pro operaci `getStatusNespolehlivyPlatce`. Hodnoty atributů `statusCode` a `statusText` jsou také stejné mimo chybějící stav 1.

2.2.4 Operace `getStatusNespolehlivyPlatceRozsireny`

Rozšířená operace rozhraní, která poskytuje více informací o nespolehlivém plátcí DPH.

Vstupní parametry:

- `dic` – DIČ subjektu, minimálně 1 a maximálně 100

Výstupem pro každé z dotazovaných DIČ je stav nespolehlivosti plátce, seznam evidovaných bankovních účtů společně s datem zveřejnění v registru a číslo finančního úřadu. Dále ještě rozšířeno o název subjektu a adresy ve strukturované podobě. Pokud je plátce nespolehlivý, je uvedeno i datum uveřejnění nespolehlivosti.

Adresa subjektu v odpovědi obsahuje hodnoty:

- ulice a číslo popisné,
- název části obce,
- název obce,
- PSČ,
- stát.

Atributy `statusCode` a `statusText` nabývají stejných hodnot jako v případě operace `getStatusNespolehlivyPlatce`.

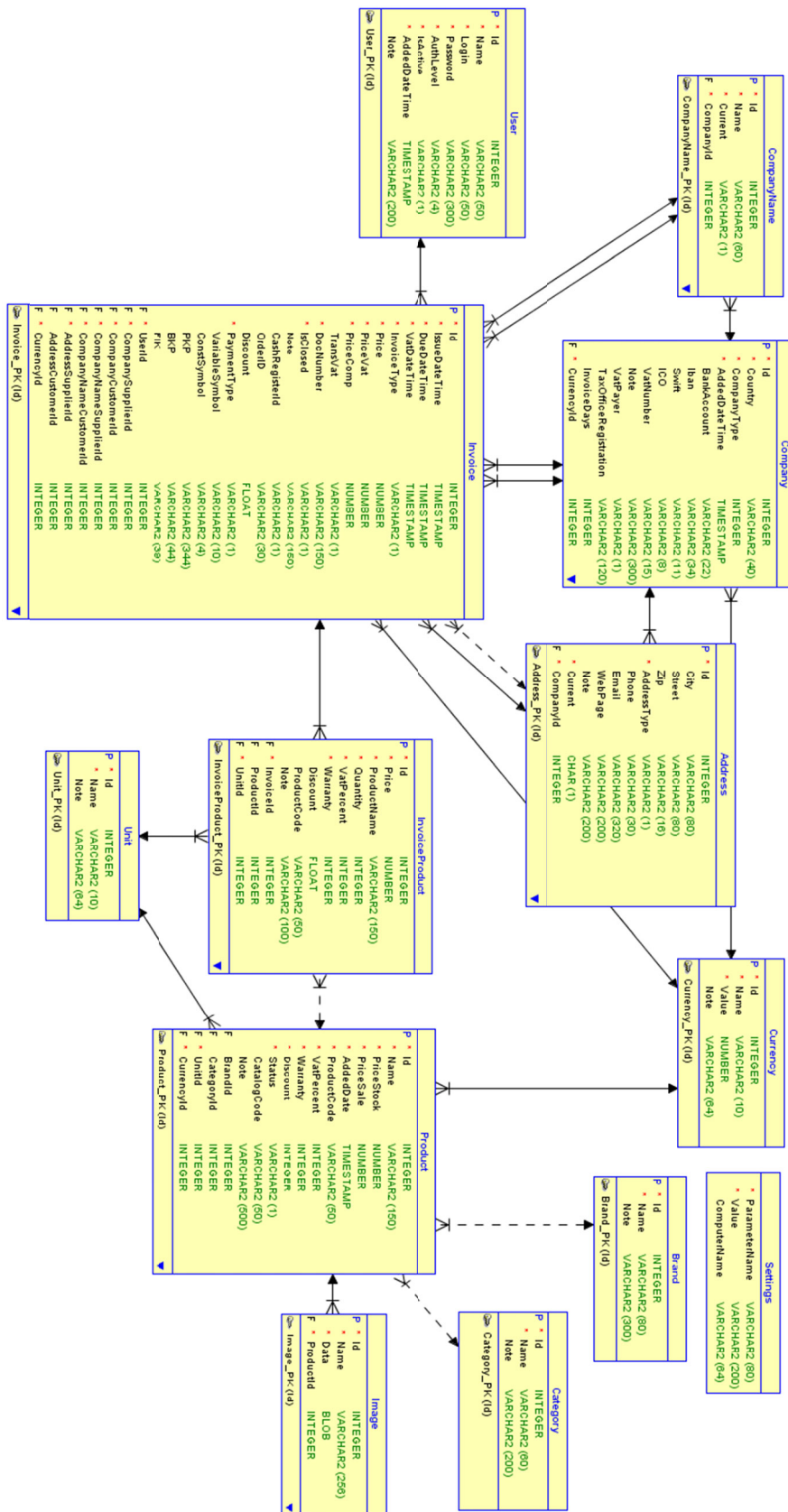
3. Výběr a návrh databáze

3.1 Výběr databáze

V první části návrhu bylo nutné stanovit, kde budou uložena data, se kterými bude aplikace pracovat. Ve hře byly dvě možnosti, buď data ukládat do lokální databáze běžící přímo na počítači jako služba, nebo k nim přistupovat za pomoci počítačové sítě. Druhá volba by nabízela větší flexibilitu a také schopnost s daty pracovat odkudkoliv. Vzhledem k záměru používat aplikaci více uživateli současně byla databáze v síti jasná volba. Na počátku byly stanoveny požadavky jako cenová dostupnost databáze a provoz nevyžadující velké nároky na implementaci.

Při tvorbě projektu nebyla k dispozici žádná již zakoupená serverová řešení, proto bylo nutné vyhledat jiné alternativy. Avšak dostupné bylo vlastní síťové úložiště NAS od Synology, kde lze volitelně instalovat softwarové balíčky. Na zařízení je nainstalován operační systém DSM 6.2.3, jehož jádrem je Linux Kernel. V aplikaci Centrum balíčků byly nalezeny balíčky MariaDB 5 a MariaDB 10. Pro tento projekt se nabízelo jako vhodné řešení databáze MariaDB 10. Vzhledem k faktu, že nebyla k dispozici žádná jiná databázová řešení než MariaDB, nebylo uvažováno o instalaci jiných jako například MySQL či PostgreSQL.

3.2 Návrh databázových tabulek



Obrázek 10: Relační návrh v Oracle SQL DataModeler, zdroj: Autor

3.2.1 Tabulka Company

Company je hlavní tabulkou komponenty Adresář, na níž jsou navázány další související tabulky. Vkládají se sem unikátní informace o firmě. Odsud jsou následně načítána různá data, například pro potřeby komponenty Fakturace.

Důležité sloupce:

- Id – primární klíč, cizím klíčem v tabulkách CompanyName, Address, Invoice,
- ICO – identifikační číslo subjektu (IČO),
- VatNumber – DIČ (daňové identifikační číslo),
- CompanyType – typ společnosti (evidovaná organizace, fyzická osoba, právnická osoba, zaniklá organizace),
- Country – zkratka státu dle ISO 3166 (dvě písmena),
- BankAccount – číslo bankovního účtu včetně předčíslí a kódu banky,
- AddedDateTime – datum a čas přidání společnosti do adresáře,
- InvoiceDays – počet dní splatnosti faktur, přičítá se jako počet dní k datu vystavení na faktuře,
- CurrencyId – cizí klíč z tabulky měn Currency.

3.2.2 Tabulka CompanyName

Tabulka sloužící pro ukládání jmen společností. Oddělení do samostatné tabulky bylo provedeno z důvodu, že společnosti mohou měnit svá jména. Zajišťuje, že po změně bude stále možné v případě potřeby vytisknout dříve vystavenou fakturu s původním názvem společnosti.

Důležité sloupce:

- Id – primární klíč, cizím klíčem v tabulce Invoice,
- Name – název společnosti,
- Current – označuje aktuální název firmy, pouze jeden název dané firmy může mít hodnotu aktuálnosti (0 nebo 1) nastavenou na 1,
- CompanyId – cizí klíč z tabulky Company.

3.2.3 Tabulka Address

Ukládají se zde různé adresy firem a klientů. Každá společnost nemusí mít adresu.

Důležité sloupce:

- Id – primární klíč, dvakrát cizím klíčem v tabulce Invoice, první vazba slouží pro adresu dodavatele a druhá pro adresu zákazníka,
- AddressType – uchovává typy adres: fakturační, doručovací nebo vyřazená. Poslední ze zmíněných slouží pro případ, že subjekt změní svoji adresu a bylo by nutné vytisknout fakturu se starou adresou,
- CompanyId – cizí klíč z tabulky Company.

3.2.4 Tabulka User

Jsou zde vedeni všichni uživatelé aplikace.

Důležité sloupce:

- Id – primární klíč, cizím klíčem v tabulce Invoice,
- Name – jméno a příjmení obsluhy,
- Login – uživatelské jméno, slouží pro přihlašování do aplikace,
- AuthLevel – úroveň oprávnění uživatele, někteří uživatelé nemusí mít zpřístupněné všechny komponenty aplikace, například nastavení,
- IsActive – říká, zda je činnost účtu uživatele pozastavena či nikoliv.

3.2.5 Tabulka Invoice

Jedna z nejdůležitějších tabulek aplikace. Určena pro ukládání hlavních údajů faktury. Dvě vazby 1:N na tabulku Company jsou ze z důvodu zaznamenání Id dodavatele a zákazníka. To samé platí pro vazbu do tabulky CompanyName a Address

Důležité sloupce:

- Id – primární klíč, cizím klíčem v tabulce InvoiceProduct,
- IssueDateTime – datum a čas vystavení faktury,
- DueDateTime – datum a čas splatnosti faktury, stanovuje se součtem IssueDateTime a počtu dní InvoiceDays odběratelské firmy,
- InvoiceType – typ faktury,
- Price – celková cena faktury bez DPH
- PriceVat – celková cena faktury s DPH,
- IsClosed – říká, zda je doklad uzavřen, pokud je uzavřen, nelze ho již editovat,

- UsedId – cizí klíč do tabulky User, říká, jaký uživatel vystavil fakturu,
- CompanySupplierId – cizí klíč do tabulky Company, odkazuje na firmu dodavatele,
- CompanyCustomerId – cizí klíč do tabulky Company, odkazuje na firmu zákazníka,
- CompanyNameSupplierId – cizí klíč do tabulky CompanyName, jméno dodavatele,
- CompanyNameCustomerId – cizí klíč do tabulky CompanyName, jméno zákaznického subjektu,
- AddressSupplierId – cizí klíč do tabulky Address, odkazuje na adresu dodavatele,
- AddressCustomerId – cizí klíč do tabulky Address, odkazuje na adresu zákaznického subjektu, nepovinný údaj.

3.2.6 Tabulka InvoiceProduct

Tabulka pro položky faktur. Ukládají se sem jak položky načtené z ceníku, tak i ručně vložené. Každá položka má proto údaj ceny, názvu, záruky, slevy, produktového kódu, sazby DPH. V případě, že jsou tyto údaje přeneseny z ceníku (z tabulky Product), nelze je následně dále editovat.

Důležité sloupce:

- Price – cena produktu, v případě ceníkového produktu je cena přenesena z tabulky ceníku Product,
- VatPercentage – procenta DPH dané položky, může být přeneseno z ceníku,
- Discount – procenta slevy, může být přeneseno z ceníků
- InvoiceId – cizí klíč do tabulky Invoice,
- ProductId – cizí klíč do tabulky Product, musí být uveden, pokud se jedná o produkt z ceníku,
- UnitId – cizí klíč do tabulky Unit, každá položka faktury může mít jinou jednotku.

3.2.7 Tabulka Unit

Tabulka jednotek, se kterými aplikace pracuje při specifikaci množství. Může se jednat například o kusy, metry, kilometry, hodiny.

Důležité sloupce:

- Id – primární klíč, cizím klíčem v tabulce InvoiceProduct a Product,
- Name – zkratka jednotky,
- Note – celý název jednotky.

3.2.8 Tabulka Product

Tabulka pro ukládání ceníkových produktů. Jsou zde uvedeny informace jako název produktu, kód produktu, katalogové číslo, ceny (skladová a prodejní), datum a čas přidání do ceníku, procenta DPH, záruka, status (vyřazeno, zařazeno, nedostupné), sleva a vazby do tabulek Currency, Brand, Category a Image.

Důležité sloupce:

- Id – primární klíč, cizím klíčem v tabulce InvoiceProduct,
- Name – název produktu,
- PriceStock – skladová cena,
- PriceSale – prodejní cena bez DPH,
- ProductCode – produktový kód,
- CatalogCode – katalogový kód,
- Status – status produktu,
- UnitId – cizí klíč do tabulky Unit, každý produkt může mít jinou jednotku (kusy, hodiny, kilometry, metry),
- CurrencyId – cizí klíč do tabulky Currency.

3.2.9 Tabulka Currency

Tabulka pro ukládání měn. Jsou zde uloženy měny, se kterými aplikace pracuje.

Důležité sloupce:

- Id – primární klíč, cizím klíčem v tabulkách Product,
- Name – mezinárodní zkratka měny,
- Value – hodnota měny vůči výchozí měně, hodnota výchozí měny je 1.

3.2.10 Ostatní tabulky

Tabulka Settings

Tabulka pro nastavení parametrů programu. Aktuálně není využita, ale může sloužit pro budoucí rozvoj aplikace. Uloženo by zde mohlo být označení pokladny, adresa EET, číslo certifikátu EET, maximální doba odezvy finančního portálu EET, označení provozovny pro EET, výchozí marže produktů či výchozí měna produktů.

Tabulka Brand

Tabulka značek produktů.

Tabulka Category

Tabulka kategorií produktů.

Tabulka Image

Tabulka obrázků produktů. Obsahuje název obrázku, data obrázku a cizí klíč produktu.

4. Realizace aplikace

Pro vývoj projektu byl zvolen programovací jazyk C# a vývojový nástroj Microsoft Visual Studio 2017. Vlastní realizace započala vytvořením projektu InvoiceMan s jeho následnou konfigurací. Ta spočívala v přidání balíčků knihoven.

Balíček Microsoft.EntityFrameworkCore pro manipulaci s databázovými daty. Balíček Pomelo.EntityFrameworkCore.MySql pro obstarání připojení do databáze a překlad LINQ dotazů. Balíček Autofac pro vyřešení závislostí tříd. Dále balíček Prism.Core pro manipulaci s příkazy a agregování událostí.

V aplikaci je dbáno na asynchronní zpracování všech dotazů do databáze tak, aby nebylo blokováno uživatelské rozhraní. K tomu účelu programovací jazyk poskytuje mechanismus asynchronního provádění metod pomocí klíčových slov async v předpisu volající metody a await na řádku volání.

Převážná část obrazovek aplikace je navrhována s ohledem na možnost úpravy velikosti s korektním zobrazením, což je podporováno technologií WPF. Vizuální část aplikace je definována pomocí souborů XAML. Jsou zde definována jak aplikační menu, tak veškerá okna a ovládací prvky. Styly uživatelského rozhraní se pro přehlednost nacházejí v samostatných souborech pojmenovaných dle typu komponent, které ovlivňují. V projektu jsou dodržovány konvence typické pro C# a WPF, což usnadňovalo orientaci během realizace.

Hodnotnou funkcí může být možnost mít moduly aplikace otevřené zároveň a jednoduše mezi nimi přepínat.

4.1 Dílčí projekty

Projektové řešení je rozděleno na dílčí projekty. Každý z nich má svoji specifickou úlohu.

- InvoiceMan.DataAccess – Knihovni projekt, v němž se nachází třídy zajišťující výměnu dat mezi aplikací a databází.
- InvoiceMan.Domain – Knihovni projekt služeb. Jsou zde umístěny třídy, například pro načítání dat z API ARES a registru spolehlivosti plátců DPH, služby pro výpočet faktur, hash hesel a podobně.
- InvoiceMan.Model – Projekt obsahující modelové třídy.
- InvoiceMan.Report – Projekt, ve které se nachází veškerá logika tiskového reportu, vizuální návrh a logika pro odesílání dat do cloudového uložiště.
- InvoiceMan.UI – Projekt spustitelné Windows aplikace. Obsahuje uživatelské formuláře, třídy datového kontextu pro uživatelské rozhraní, styly a další třídy pro potřeby uživatelského rozhraní.

Výstupem projektů DataAccess, Domain a Model jsou knihovny DLL (Dynamic Link Library), které jsou zahrnuty do projektu UI.

Aplikace se sestává z následujících částí:

- Přihlašovací obrazovka,
- Hlavní plocha aplikace s nabídkami k otevření jednotlivých modulů,
- Adresář,
- Fakturace,
- Položky faktur,
- Ceník,
- Tiskový report.

4.2 Realizace databázového datového modelu

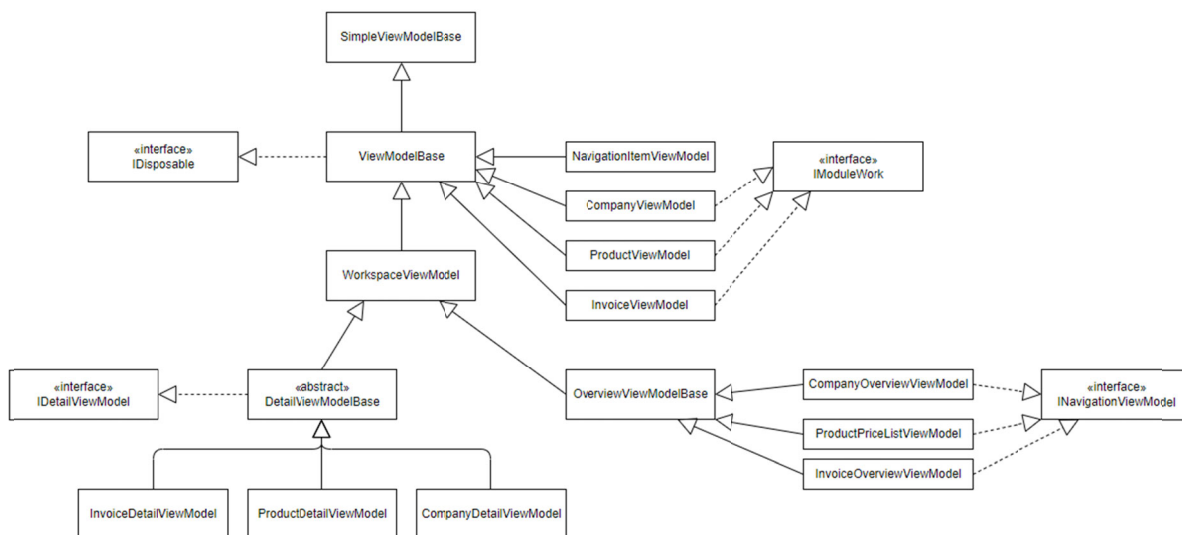
Projekt se sestává ze základních částí, které definuje architektura MVVM. V první fázi byla navržena datová logika databáze za pomoci nástroje Oracle SQL Developer Data Modeller. Výsledná logika byla následně přenesena do tříd (tzv. modelových tříd) projektu. Pro tvorbu databázového modelu existují dva možné postupy:

- Code-first – Modelové POCO třídy jsou včetně vzájemných vazeb převedeny pomocí mapovače do SQL DDL příkazů (CREATE, DROP, ALTER...) a následně migrovány do databáze.
- Database-first – Jako první je vytvořena struktura databázových tabulek. Návazně jsou tabulky za použití některého z nástrojů ORM, například Entity Framework pomocí techniky Scaffolding převedeny na třídy POCO.

Pro tento projekt byla zvolena metoda Code-first. Na základě databázového diagramu byly vytvořeny třídy POCO a ty následně pomocí mapovače EF migrovány do databáze.

4.3 Realizace základních tříd aplikace

Následující základní třídy se nachází v projektu InvoiceMan.UI ve jmenném prostoru ViewModel.



Obrázek 11: UML Diagram základních tříd, zdroj: Autor

SimpleViewModelBase

Nejzákladnější třída z koncepce ViewModel. Obsahuje jen nejzásadnější funkcionality. Implementuje rozhraní INotifyPropertyChanged (z knihovny .NET) vystavující událost PropertyChanged. Logika vyvolání události je obvykle implementována v metodě jménem OnPropertyChanged, tak je tomu i zde. Jejím účelem je v případě potřeby vystavit událost o změně vlastnosti (proměnné). Tato proměnná může být identifikována ručně tak, že uvedeme její název jako řetězec nebo neuvedeme nic a tím je doplněn automaticky podle názvu proměnné, ze které byla tato metoda vyvolána.

ViewModelBase

Dědí ze třídy SimpleViewModelBase. Implementuje rozhraní IDisposable (z knihovny .NET), ve které je poskytnut mechanismus pro uvolňování nespravovaných prostředků, například obslužných rutin událostí či otevřených souborů. Dále obsahuje metody pro nastavení delegáta (odkaz na metodu s danou signaturou) pro komunikaci mezi vlákny, konkrétně do hlavního vlákna uživatelského rozhraní.

WorkspaceViewModel

Dědí ze třídy `ViewModelBase`. Zajišťuje základní funkcionalitu pro oznámení pokusu o zavření daného formuláře. Tato třída je určena pro třídy pohledových datových kontextů a detailů.

OverviewViewModelBase

Základní třída datového kontextu pro přehledové formuláře, která dědí ze třídy `WorkspaceViewModel`. Obsahuje základní implementaci příkazů `DelegateCommand` pro otevření detailu, obnovení přehledu, vytvoření a odstranění položky. `DelegateCommand` je součástí NuGet balíčku `Prism.Core`. Její další součástí je kolekce přehledových položek pro otevření detailu, aktuálně vybraná položka a časovač. Časovač automaticky dle nastaveného intervalu obnoví položky přehledu. Výchozí hodnota intervalu je 10 minut.

DetailViewModelBase

Základní abstraktní třída pohledových detailů dědicí z `WorkspaceViewModel` a zároveň implementující rozhraní `IDetailViewModel`. Rozhraní specifikuje metodu `LoadAsync(id)` pro asynchronní načítání dat, vlastnosti `Id` a `HasChanges`. Dále jsou zde metody pro příkaz k odstranění a uložení s výchozí implementací.

NavigationItemViewModel

Základní třída přehledových navigačních položek dědicí z `ViewModelBase`. Nachází se zde proměnné `Id` a `DisplayMember`. Dále obsahuje základní implementaci příkazu `DelegateCommand` pro otevření položky, při které je vyvolána událost s parametrem `Id` položky a typem přehledu.

4.4 Realizace tříd pro komunikaci s databází

Operace CRUD

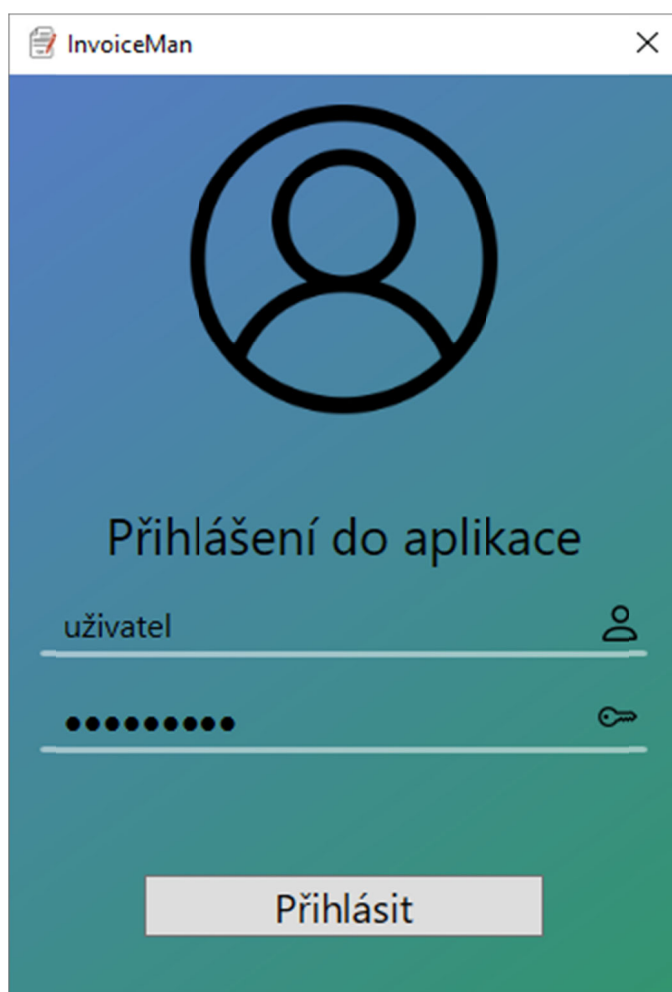
Pro zprostředkování datového přenosu byla definována základní třída `GenericRepository`, která dědí z `InvoiceManDbContext`, jehož předkem je `DbContext` (třída z balíčku `Microsoft.EntityFrameworkCore`). Objektu `DbContext` je znám databázový připojovací řetězec. `GenericRepository` obsahuje univerzální metody: `Add`, `GetByIdAsync`, `GetAllAsync`, `HasChanges`, `Remove`, `RemoveById`, `CountAsync` a `SaveAsync`. Od `GenericRepository` dědí například třídy zajišťující práci s detaily záznamů jednotlivých tabulek. Například: `CompanyRepository`, `ProductRepository` či `InvoiceRepository`. V nich jsou implementována případná rozšíření funkcionalit pro konkrétní potřeby daných typů záznamů.

Práce s daty jen pro čtení

K zobrazení přehledu slouží třída náhledů `LookupDataService`, která dodává fragmenty záznamů pouze pro čtení. Implementuje různá rozhraní, například: `ICompanyLookupDataService` či `IInvoiceLookupDataService`. Načítáním jen nutných dat je ušetřen zbytečný přenos těch, kterých by při zobrazení přehledu nebylo využito. Konkrétní detailní data jsou načtena až ve třídě `GenericRepository` nebo jedním z jeho potomků. Další výhodou následujícího přístupu je to, že načítaná data nejsou ukládána do mezipaměti. Dosaženo je tím pomocí zavolání metody `AsNoTracking()` na dotazovou metodu. Tím se zásadně liší od tříd `Repository`.

4.5 Přihlašovací okno aplikace

Přihlašovací obrazovka tvoří textová pole pro zadání uživatelského jména a hesla. Pole hesla je maskováno z bezpečnostních důvodů náhradními znaky. Po stisku tlačítka pro přihlášení je vložené uživatelské jméno a heslo porovnáno s hodnotou v tabulce uživatelů. Hesla v databázi jsou z bezpečnostních důvodů hashována. Použitou hashovací funkcí je PBKDF2, která dává možnost v algoritmu použít sůl. Po úspěšném ověření přihlašovacích údajů je uživateli otevřeno hlavního okno aplikace s nabídkami. V aplikaci je definován výchozí servisní uživatelský účet.



Obrázek 12: Přihlašovací okno aplikace, zdroj: Autor

4.6 Hlavní okno aplikace

Z hlavního okna jsou dostupné prostřednictvím lišty nabídek hlavní moduly aplikace. Ty jsou řešeny pomocí komponent Menu a MenuItem. Reakce na otevření modulu z nabídky je realizována pomocí implementace rozhraní pro příkazy ICommand, konkrétně DelegateCommand z balíčku Prism.Core. Každé tlačítko nabídky uvádí typ otevíraného modulu jako parametr při provedení příkazu. Tím je rozlišeno, jaký typ modulu má být otevřen. Nabídka „Artikl“ a její podnabídka s položkou „Ceník“ vyvolá obrazovku ceníku. V poslední nabídce „Další“ je umístěna položka „Průzkumník obchodního prostředí“.

Ve chvíli, kdy je otevřen kterýkoliv modul aplikace, je hlavní plocha překryta jeho formulářem. Otevřené moduly aplikace včetně jejich názvu jsou zobrazeny na levém vnitřním okraji okna jako záložky.



Obrázek 13: Hlavní okno aplikace bez otevřených modulů, zdroj: Autor

4.7 Adresář

Stěžejní částí aplikace je adresář, jenž využívá data veřejných registrů. Uživateli umožňuje velice rychle zadat novou firmu s minimálním úsilím. Adresář představuje obrazovku přehledu, ze které se následně přechází do detailu. Přehled obsahuje základní informace o firmě a v horní části jsou umístěny ikony pro vykonání akcí (obnovit, přidat, otevřít, odebrat). Je-li subjekt odběratelem či dodavatelem na faktuře, nelze jej odstranit. Při označení řádku přehledu je zobrazeno více informací subjektu. Jednotlivé položky jsou po dvojkliku otevřeny v dalších záložkách.

Během přidání nového subjektu je v detailu vytvořena nová položka názvu, kterou je nutné vždy vyplnit, jinak uživateli není umožněno záznam uložit. Každý subjekt musí mít alespoň jeden název. Pokud firma změní svůj název, není problém přidat další a nastavit ho jako aktuální. Klepnutím na ikonu zeleného plus, přidáme další název a zaškrtnutím políčka „Aktuální“ ho nastavíme jako aktuální. Nyní bude tento název zobrazován jako výchozí ve fakturaci pro daný subjekt.

Adresa subjektu je nepovinná pro všechny typy organizací, kromě typu „Evidovaná organizace“. Tento typ organizace představuje dodavatelský subjekt. U něho je nutné vyplnit alespoň telefon nebo email. Přidání nové adresy probíhá obdobně jako u přidání názvu.

V případě, kdy vytváříme nový záznam a známe název nebo IČO subjektu, volitelně i obec sídla, potom lze využít funkci vyhledání v registru ekonomických subjektů ARES. Pro vyhledávání je nutné dodržovat zásady výše uvedené v kapitole 2.1. Funkcionalita vyhledávání v ARES se objeví po klepnutí na ikonu šipky směřující dolů. Výsledky jsou zobrazeny v rozbalovacím menu. Z rozhraní ARES jsou přebírány následující informace: IČO, DIČ, název firmy, kód státu, kód právní formy, město, ulice, domovní číslo a PSČ. Výběr libovolného subjektu zvolíme zmačknutím tlačítka „Převzít“. Převzat je jak nalezený název, tak i jeho adresa, případně bankovní účet. Pokud je subjekt plátcem DPH, dojde k prověření na základě jeho DIČ v rejstříku spolehlivosti plátců DPH a výsledek je zobrazen pod políčkem pro vyhledávání. Každý formulář detailu, který obsahuje nevalidní datová pole, je označen červenou barvou. Při umístění kurzoru myši na takové pole je zobrazen popis dané chyby.

Kompletní formulář je uveden na následujícím obrázku.

InvoiceMan

Fakturace Adresář Artikl Další

Adresář

Firmy DIAMANT 2000 cz... X

DIAMANT 2000 cz, s.r.o.

Názvy

Název	Aktuální
DIAMANT 2000 cz, s.r.o.	<input checked="" type="checkbox"/>

Název Aktuální

Vyhledávání v ARES Výsledky

Subjekt je spolehlivý plátcem.

Název obce (volitelně)

Typ

IČO DIČ Plátcem DPH

Registrace společnosti na úřadu

Stát Měna

Počet dní splatnosti faktur

Bankovní spojení

Bankovní účet

IBAN BIC/SWIFT

Poznámka

Adresy

Typ	Telefon	Email	Město	Ulice	PSČ	Aktuální
Fakturační		info@diamant.cz	Brno	Šikulova 193	63500	<input checked="" type="checkbox"/>

Typ Aktuální

Telefon

Email

Město

Ulice

PSČ

Webová stránka

Poznámka

Obrázek 14: Detail subjektu v adresáři, zdroj: Autor

4.8 Ceník

Ceník patří k dalším důležitým částem aplikace. Vedeny jsou zde všechny produkty, které lze uvést na fakturách. Produkty jsou zobrazeny v přehledu společně se všemi důležitými údaji každé položky. Důležitými údaji jsou: produktové číslo, název produktu, katalogové číslo, skladová cena bez DPH, prodejní cena bez DPH, prodejní cena s DPH, měna, sazba DPH, záruka a další. Stejně jako na každé obrazovce přehledu lze provádět standardní operace s položkami: obnovení, nová položka, otevřít a odstranit. Nelze odstranit ty ceníkové položky, které jsou svázané s položkami faktur.

V horní části modulu se nachází karty, které zobrazují přehled a každý z otevřených detailů. Otevřené produkty jsou uvedeny na záložkách svým zkráceným názvem.

Produktové číslo	Katalogov	Název	Záruka	Cena (skladov	Cena (prodej	Cena (prodejní	Cena (prodejní s DI	Měr	Sleva %	Sazb	St	Pc	D
GY50218984	25	Lenovo 530 Wireless Mouse (Platinum Grey) s bateri	24	219	259	259	313,39	CZK	0	21	Za	03	
CC-9011174-WV	24	Corsair iCUE 220T RGB Tempered Glass bílá	24	1513	1713	1713	2072,73	CZK	0	21	Za	03	
CC-9011181-WV	23	Corsair 275R Tempered Glass černá	24	1355	1485	1485	1796,85	CZK	0	21	Za	03	
CC-9011184-WV	22	Corsair 110Q černá	24	1140	1240	1240	1500,4	CZK	0	21	Za	03	
NH-U12S	21	NOCTUA NH-U12S	60	1368	1600	1120	1355,2	CZK	30	21	Za	03	
BX807010600K	20	Intel Core i5-10600K	36	4636	4836	4836	5851,56	CZK	0	21	Za	03	
S26361-F2222-L	18	Fujitsu NVIDIA Quadro P620 2GB	24	4514	4714	4714	5703,94	CZK	0	21	Za	03	
I199155C	19	ROLINE datový k HDD SATA 3.0. 1xHDD. 0.5m. zápac	24	56	65	65	7865	CZK	0	21	Za	03	
ST2000DM008	17	Seagate BarraCuda 2TB	24	982	1082	1082	1309,22	CZK	0	21	Za	03	
MZ-V7E500BW	16	Samsung 970 EVO 500GB	60	1291	1591	1591	1925,11	CZK	0	21	Za	03	
FQC-08926	15	Microsoft Windows 10 Pro CZ 64-bit (OEM)	24	3345	3545	3545	4289,45	CZK	0	21	Za	03	
580-ADG ²	14	Dell KB-216 černá - CZ	24	227	240	240	290,4	CZK	0	21	Za	03	
LF24T350FHRXE	13	24" Samsung F24T350	24	2071	2471	2471	2989,91	CZK	0	21	Za	03	
27S02EA#BCM	12	HP M01-F0000nc	24	9209	9909	9909	11989,89	CZK	0	21	Za	03	
90M80Z90-M0E	11	ASUS TUF Z390-PLUS GAMING (Wi-Fi)	36	2470	2770	2770	3351,7	CZK	0	21	Za	03	
KF432C16B#K2/	6	Kingston FURY 32GB KIT DDR4 3200MHz CL16 Beast	120	2221	2521	2521	3050,41	CZK	0	21	Za	03	
5W6-00019	4	Microsoft Surface Laptop 4 Black	24	32069	34069	34069	41223,49	CZK	0	21	Za	03	
GA401QC-K212:	10	Asus ROG Zephyrus G14 GA401QC-K2123T Eclipse C	24	26800	28800	28800	34848	CZK	0	21	Za	03	
82BJ006E#CK	9	Lenovo Yoga 7 15ITL5 Dark Moss + aktivní stylus Ler	24	25350	26456	26456	32011,76	CZK	0	21	Za	03	
Z1240005N	8	Macbook Air 13" M1 CZ Vesmírně šedý 2020	24	26000	27711	27711	33530,31	CZK	0	21	Za	03	
3A5H6EA#BCM	7	HP ProBook 450 G8	36	12750	13156	13156	15918,76	CZK	0	21	Za	03	
1646300	5	Bang & Olufsen Beoplay H9 3rd Gen. Matte Black	24	4562	4962	4962	6004,02	CZK	0	21	Za	11	
RZ01-03250100-	1	Razer Viper Mini	36	950	1081	1081	1308,01	CZK	0	21	Za	11	
GB3266QSU-B1	2	32" iiyama G-Master GB3266QSU-B1	36	7250	7796	7796	9483,16	CZK	0	21	Vy	11	
TN-9310-N2-72:	3	Dell XPS 13 (9310) Touch Silver	24	39450	41104	41104	49735,84	CZK	0	21	Za	11	
cizi	100	Cizí produkt	24	18	20	20	242	EUR	0	21	Za	11	

Obrázek 15: Přehled ceníku, zdroj: Autor

InvoiceMan

Fakturace Adresář Artikli Další

Fakturace X

Produkty Lenovo 530 Wirel... X Microsoft Surfac... X Corsair 275R Tem... X

Název produktu Corsair 275R Tempered Glass černá Status Zařazeno

Adresář X

Produktový kód (PN) CC-9011181-WW Katalogový kód 23

Značka (Žádná) Kategorie (Žádná) Záruka (měsíce) 24

Cenik X

Cena (skladová) bez DPH 1355 Cena (prodejní) bez DPH 1485 Cena (prodejní) s DPH 1796,85 Měna CZK Sazba DPH 21 % Sleva (%) 0 Měrná jednotka ks

Obrázek 16: Detail ceníkové položky, zdroj: Autor

4.9 Fakturace

Modul fakturace slouží k vytvoření faktury včetně jejich položek. Obrazovka modulu je tvořena dvěma záložkami. Na první z nich je umístěn přehled faktur včetně hlavičkových informací a podobně jako v obrazovce adresáře jsou v horní části umístěny ikony akcí. Druhá záložka obsahuje přehled položek včetně detailu.

V hlavičkách faktur lze volit dodavatele a odběratele na základě typu společnosti (CompanyType). Položka měna je přednastavována na základě měny zvoleného odběratele, lze ji však ještě změnit. To samé platí i pro položku datum splatnosti, kde je k aktuálnímu datu přičten počet dní splatnosti faktur odběratelské firmy.

Při procházení přehledu faktur jsou kompletní údaje záznamu načteny a promítnuty do detailu až po 500 ms z důvodu optimalizace přenosu dat. K tomuto účelu je využit časovač spouštěný v odděleném vlákne. Při rychlém procházení seznamu by docházelo ke zbytečnému načítání dat, se kterými uživatel ne zamýšlí pracovat.

The screenshot shows the InvoiceMan application interface. At the top, there are tabs for 'Fakturace', 'Adresář', 'Artikl', and 'Další'. The main window title is 'InvoiceMan'. Below the tabs, there is a search bar with the value '2112090001' and several action icons (print, save, refresh, add, delete). The main content area is divided into two sections: 'Faktury' and 'Položky faktury'. The 'Faktury' section displays a table of invoices with the following data:

Odběratel	Číslo dokladu	Částka bez DPH	Částka s DPH	Měna	Datum vystavení	Datum splatnosti	Platba	Uživatel
DIAMANT 2000 cz, s.r.o.	2112030002	23 509,60	28 446,62	CZK	24.08.2021 16:15:25	23.12.2021 14:16:44	Převodní příkaz	Viktor Homolka
Karel Novák	2111150001	2 335,91	2 826,45	USD	25.08.2021 19:00:20	23.12.2021 14:16:56	Převodní příkaz	Viktor Homolka
Karel Novák	2112060001	42 829,00	51 799,09	CZK	03.11.2021 0:00:00	17.11.2021 0:00:00	Převodní příkaz	Viktor Homolka
Eurometal Chotěboř spol. s r.o	2111110001	241 025,00	291 640,25	CZK	11.11.2021 20:57:58	25.11.2021 20:58:07	Převodní příkaz	Viktor Homolka
Eurometal Chotěboř spol. s r.o	2112030001	31,38	37,15	EUR	14.11.2021 16:46:18	28.11.2021 16:46:25	Hotově	Viktor Homolka
Karel Novák	2112090001	54 349,00	65 762,29	CZK	09.12.2021 14:21:04	23.12.2021 14:21:17	Převodní příkaz	Viktor Homolka

Below the table, there is a detailed view of the selected invoice (2112090001). It includes fields for 'Dodavatel' (Diatek), 'Odběratel' (Karel Novák), 'Měna' (CZK), 'Typ platby' (Převodní příkaz), 'Datum vystavení' (09.12.2021), 'Datum splatnosti' (23.12.2021), 'Datum zdanitelného plnění' (09.12.2021), 'Číslo dokladu' (2112090001), 'Číslo objednávky zákazníka' (56), 'Konstatní symbol', and 'Variabilní symbol' (12345678). At the bottom, there is a summary section showing 'Celkem cena' with 'bez DPH' (54 349,00 CZK) and 's DPH' (65 762,29 CZK).

Obrázek 17: Přehled faktur, zdroj: Autor

Položky faktur může uživatel zadávat ručně nebo je přebírat z ceníku. Produkty lze vyhledat na základě názvu položky, produktového či katalogového čísla. Potom, co uživatel přestane psát do vyhledávacího pole na déle než 500 ms a zároveň je počet znaků v poli větší nebo rovno 3, je provedeno samotné hledání. Výsledky jsou vráceny do vedlejší rozbalovací nabídky. Položka faktury se sváže s produktem po stisku tlačítka „Převzít“. Vazbu můžeme zrušit zmačknutím červeného křížku vedle názvu produktu. Převzetí produkt z ceníku způsobí, že již nebude možné upravovat data vybrané položky kromě množství.

Celková částka faktury je přepočítána vždy, když dojde k jedné z následujících akcí: přidání nové položky z ceníku, odstranění položky, změna částky, sazby DPH, množství či měny faktury.

Na fakturu lze přidávat i takové ceníkové položky, které mají jinou měnu, než je na faktuře. Přidáním ceníkové položky s odlišnou měnou dojde k automatickému přepočítání částky do cílové měny faktury. Kurz měn je stanoven vůči výchozí měně v CZK v tabulce Currency.

Přepočet částky ze zdrojové měny do cílové:

$\text{targetCurrencyValue} / \text{sourceCurrencyValue} * \text{productPrice}$

The screenshot shows the InvoiceMan application interface. At the top, there are tabs for 'Fakturace', 'Adresář', 'Artikl', and 'Další'. The main area displays a list of invoice items with columns for 'Produktové číslo', 'Název', 'Záruka [měsíc]', 'Množst', 'M', 'J. cena bez DP', 'Sleva [%]', 'Cena bez DP', 'Sazba DP', 'Celkem bez DP', and 'Celkem s DP'. Below the list, there is a detailed view of a selected item, 'HP ProBook 450 G8', with fields for 'Kód produktu', 'Název', 'Množství', 'Měrná jednotka', 'Vyhledat v ceníku', 'Produkty', 'Převzít', 'J. cena bez DPH', 'Sazba', 'Záruka [měsíce]', and 'Celkem cena'.

Produktové číslo	Název	Záruka [měsíc]	Množst	M	J. cena bez DP	Sleva [%]	Cena bez DP	Sazba DP	Celkem bez DP	Celkem s DP
3A5H6EA#BCM	HP ProBook 450 G8	36	3	ks	13 156,00	0	13156	21	39 468,00	47 756,28
FQC-08926	Microsoft Windows 10 Pro CZ 64-bit (OEM)	24	3	ks	3 545,00	0	3545	21	10 635,00	12 868,35
MZ-V7E500BW	Samsung 970 EVO 500GB	60	1	ks	1 591,00	0	1591	21	1 591,00	1 925,11
LF24T350FHRXEN	24" Samsung F24T350	24	1	ks	2 471,00	0	2471	21	2 471,00	2 989,91
	Instalace OS a software	24	2	h	0,00	0	0	21	0,00	0,00
	Doprava	24	46	krr	4,00	0	4	21	184,00	222,64

Kód produktu	Název	Ceníkový produkt	Množství	Měrná jednotka
3A5H6EA#BCM	HP ProBook 450 G8	X	3	ks kusy
Vyhledat v ceníku	Produkty	Převzít	J. cena bez DPH	Sazba
			13156 CZK	21 %
			Záruka [měsíce]	
			36	
Celkem cena		bez DPH	s DPH	
		54 349,00 CZK	65 762,29 CZK	

Obrázek 18: Položky faktury, zdroj: Autor

4.10 Tiskový report

Vyvolání tiskového reportu je aktivováno stiskem tlačítka „*Tisk faktury*“ označené ikonou tiskárny v přehledu faktur. Následně je zobrazeno modální okno s náhledem vybrané faktury dle vybraného formátu papíru, výchozím je A4. Tiskový report je implementován komponentou ReportViewer. Tisk je možné směřovat na tiskárnu nebo exportovat do formátu PDF, Microsoft Word či Microsoft Excel. Další možností je uložit fakturu ve formátu PDF na cloudové úložiště Microsoft OneDrive. Jedná se o modrou ikonu obláčku v nástrojové liště. Po jejím zvolení následuje zobrazení okna pro přihlášení pomocí účtu Microsoft. Pokud je uživatel úspěšně ověřen, bude na jeho datovém úložišti Microsoft OneDrive vytvořena složka s názvem aplikace (InvoiceMan) a do ní umístěn exportovaný PDF soubor.

Faktura - daňový doklad č. 2112090001

Variable symbol: 12345678

Dodavatel: Diatek Dolní Kouzov 465 28101 Velim IČO: 45654156 DIČ: CZ971211068 Telefon: 123456789 Email: info@diatek.cz Web: www.diatek.cz	Odběratel: Karel Novák Kolmá 123 53002 Pardubice
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------

Číslo účtu: 4165415164/2010 IBAN: 645246246282
Úřad XYZ Swifit: 14862

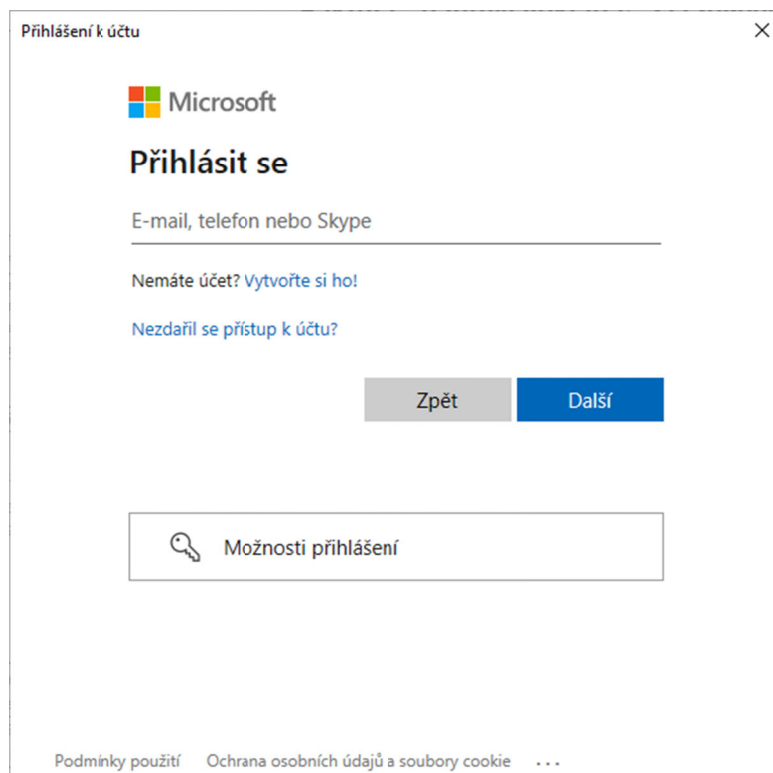
Datum vystavení: 09.12.2021
Datum splatnosti: 23.12.2021
Datum zdanitelného plnění: 09.12.2021
Forma úhrady: Převodní příkaz

Kód	Název	Záruka	J. cena	Počet	DPH %	Základ	Celkem
3A5H6EA#BCM	HP ProBook 450 G8	36	13 156,00	3 ks	21	8 288,28	47 756,28
FQC-08926	Microsoft Windows 10 Pro CZ 64-bit (OEM)	24	3 545,00	3 ks	21	2 233,35	12 868,35
MZ-V7E500BW	Samsung 970 EVO 500GB	60	1 591,00	1 ks	21	334,11	1 925,11
LF24T350FHRXEN	24" Samsung F24T350	24	2 471,00	1 ks	21	518,91	2 989,91
	Instalace OS a software	24	0,00	2 h	21	0,00	0,00
	Doprava	24	4,00	46 km	21	38,64	222,64
	Položka 15	24	100,00	1 ks	15	15,00	115,00

Vyčíslení DPH v CZK:			Celkem za doklad v CZK:	65 877,29
Sazba %	Základ	DPH		
15 %	100,00	15,00		
21 %	54 349,00	11 413,29		

Děkujeme za váš nákup

Obrázek 19: Tiskový report, zdroj: Autor



Obrázek 20: Okno pro přihlášení k účtu Microsoft, zdroj: Autor

4.11 Průzkumník obchodního prostředí

Účelem této aplikační komponenty je průzkum obchodního prostředí dle zadaných kritérií. Vyhledávat lze na základě řady kritérií: název, IČO, obec, ulice a číslo popisné. Ostatní filtrační parametry vybíráme pomocí rozbalovacím menu: ekonomická činnost, právní forma, finanční úřad. Můžeme rovněž zadat parametr třídění: IČO, obec, obchodní firma. Výsledky jsou zobrazeny v datové mřížce a v pravé části nad ní je uveden počet výsledků.

InvoiceMan

Fakturace Adresář Artid Další

Průzkumník o. prostředí

Název subjektu IČO Ekonomická činnost Právní forma Třídít dle
Vydávání softwaru | 582 (Nevybráno) netříděno

Obec Ulice Č. p. Finanční úřad Max. výsledků
Pardubice (Nevybráno) 200

Vyhledat Počet výsledků: 4

Název subjektu	IČO	DIČ	Město	Ulice	Forma
PRÁCE.TIPS, s.r.o.	25557696	25557696	Pardubice I	Zelené Předměstí, Masarykovo náměstí 1484	Společnost s ručením omezeným 112
NOVO-LEN správce s.r.o.	28858132	28858182	Pardubice I	Bílé Předměstí, Sakařova 1426	Společnost s ručením omezeným 112
Viacomp s.r.o. v likvidaci	27518191		Pardubice IV	Nemošice, Vodárenská 22	Společnost s ručením omezeným 112
BUSINESS INFORMATION SYSTEMS & SERVICES BISS, s.r.o.	25284746	25284746	Pardubice III	Studánka, Bartořova 932	Společnost s ručením omezeným 112

Fakturace

Obrázek 21: Průzkumník obchodního prostředí, zdroj: Autor

5. Komerční řešení

5.1 Stormware – POHODA

POHODA je ekonomický a účetní software vyvíjený společností Stormware působící na českém trhu již od roku 1994. Program je napsaný v programovacím jazyce C++. POHODA je nabízena v několika verzích, z nichž každá poskytuje různé funkce. Patří mezi ně verze Start, Mini, Lite, Jazz, Standard, Profi, Premium a Komplet. Liší se od sebe množstvím modulů a funkcionalit, mezi které může patřit například: daňová evidence, fakturace, skladová evidence a mzdová správa. Program je vhodný pro menší a střední firmy. Nabízí jednoduché a intuitivní uživatelské rozhraní, díky kterému je jeho obsluha snadná. Mimo řadu verzí desktopového programu POHODA je nabízena rovněž mobilní verze s názvem mPOHODA, která poskytuje jednoduché funkce mobilní pokladny a fakturace. Program POHODA si lze také zdarma a s omezenými funkcemi vyzkoušet ve verzi Start. Existují dvě možnosti. První z nich je stáhnout si program bez registrace s časovým omezením použití na 3 měsíce a omezením počtu záznamů v modulech na 10. Druhou možností je stáhnout program s registrací, kde je navýšen možný počet vedených záznamů v některých modulech na 200. [26]

Aplikace POHODA nedovoluje v žádné z nabízených verzí možnost exportu PDF faktury na cloud a rovněž neobsahuje podobnou funkci jako průzkumník obchodního prostředí. Není možné otevření několika aplikačních modulů současně. Vzhledem k použití starší technologie v kombinaci s již hotovým funkčním řešením by zapracování vyžadovalo velké množství času a peněz.

5.2 Elisoft – EKONOM System

Desktopový ekonomický software vyvíjený zlínsko-plzeňskou společností Elisoft již od roku 1998. K vývoji byl použit programovací jazyk Visual ForPro. EKONOM je nabízen v podobě programových sestav či jednotlivých modulů. Program je vhodný pro drobné podnikatele a malé firmy. Kromě desktopových verzí programu EKONOM je nabízena také cloudová aplikace EKONOM Online, mezi jejíž hlavní funkce patří fakturace a pokladna pro maloobchodní prodej. EKONOM je také dostupný ve zkušební verzi Start k vyzkoušení některých modulů s omezeným počtem pracovních záznamů. [27]

EKONOM System nabízí jednoduché uživatelské prostředí, avšak v některých ohledech se jeví málo přehledné.

V určitých případech je na obrazovce při práci zobrazeno i 6 modálních oken. Nevýhodou se také může jevit nemožnost načítat informace o podnikatelském subjektu z registru ekonomických subjektů ARES. Je zde však funkce pro načítání čísel bankovních účtů a ověření spolehlivosti plátce DPH z registru ADIS. Vzhledem k faktu, že aplikace EKONOM je vyvíjena již přes 20 let, používá se zde totožná koncepce jako v případě softwaru POHODA. Aplikace také nedisponuje stejnými funkcemi neobsaženými ani v programu POHODA.

5.3 Ježek software – DUEL

DUEL je aplikace pro vedení ekonomických agend vyvíjený firmou Ježek software. Pro vývoj programu je použit programovací jazyk Delphi a C#. DUEL je vyvíjen dlouhou dobu podobně jako výše uvedené aplikace, a to již více než 20 let. DUEL je prodáván jako jednotlivé aplikační moduly, například: Sklady, Mzdy, Účetnictví či Analýzy. Program je určen pro malé a střední firmy, živnostníky a účetní. DUEL je k dispozici také jako demoverze pro vyzkoušení různých modulů. Limitem jsou omezené počty záznamů v různých modulech. [28]

DUEL nabízí modernější a uživatelsky přívětivé aplikační prostředí. Mezi zajímavé funkce patří možnost otevřít více programových modulů současně. Aplikace nabízí možnost při zakládání nového záznamu do adresáře načíst informace o firmě z registru ekonomických subjektů ARES. Navíc dovoluje kontrolu DIČ v evropském systému VIES a report o firmě ze služby Cribis. Aplikace neumožňuje export PDF faktur na cloud a rovněž neobsahuje podobnou funkcionalitu jako průzkumník obchodního prostředí.

ZÁVĚR

Cílem této práce bylo analyzovat využitelnost dostupných API, poskytující data o ekonomických subjektech v ČR. Byla provedena analýza registru ekonomických subjektů ARES a jeho vybraných API. Identifikovány jsou výhody a nevýhody vybraných služeb, které byly v průběhu analýzy a vývoje aplikace zjištěny. Dalším cílem bylo navrhnout a realizovat aplikaci využívající vybrané rozhraní API inovativním způsobem. K tomuto cíli byla zvolena cesta vytvoření fakturační desktopové aplikace v moderním vývojovém prostředí s použitím nových technologií. Rovněž bylo záměrem, aby aplikace měla praktické využití a zefektivnila práci uživatele.

Jako první byla popsána struktura databázových tabulek a vztahů mezi nimi. Byly uvedeny důležité atributy tabulek včetně jejich účelu a následně byl uveden postup průběhu implementace těchto tabulek. Další část práce popisuje projektové rozvržení a k čemu každý dílčí podprojekt slouží. V neposlední řadě je představen každý modul aplikace.

Cíle práce se podařilo naplnit, jelikož jsou implementovány inovativní funkcionality v porovnání s komerčními řešeními, viz kapitola 5.

První z nich je průzkumník obchodního prostředí, v němž může uživatel vyhledávat údaje o podnikatelských subjektech z ARES dle zadaných kritérií (např.: město, ulice, ekonomická činnost, právní forma a další), viz kapitola 4.11. Uživatel má možnost rychle se orientovat v obchodním prostředí dané lokality, kde působí. Na základě toho může například upravit a cílit propagaci svých výrobků a služeb.

Aplikace poskytuje adresář subjektů, v němž je opět možné využít služeb ARES, viz kapitola 4.7. Práce v adresáři s využitím automatického získávání dat přináší řadu výhod. Zefektivňuje práci uživatele, vylučuje chybu obsluhy při ručním zadávání dat a poskytuje také informace o spolehlivosti daného subjektu (spolehlivý plátce DPH). Dílčí nevýhodou získávání dat z ARES se může jevit pomalejší odezva v nočních hodinách, viz kapitola 2.1.4.

Finálním produktem aplikace je výtisk faktury, který je zpracováván na základě tiskové šablony ve formátu XML. Tento přístup dovoluje operativně upravit vizuální stránku faktury dle požadavků uživatele bez nutnosti zásahu do kódu programu. Další z inovativních funkcí, která se vztahuje k tisku, je možnost vytvořený PDF dokument uložit na cloudové uložení, konkrétně Microsoft OneDrive, viz kapitola 4.10, a tím zjednodušit například sdílení dokumentů mezi vzdálenými uživateli.

Uživatelský komfort vylepšuje možnost současného otevření více aplikačních modulů a tím dovoluje uživateli jednoduše přenášet data mezi nimi. To platí i pro otevřené položky modulů.

Robustní návrh projektu byl sice časově náročný, avšak umožní budoucí rozvoj aplikace o další moduly a funkcionality. Plánovány jsou další moduly, například rozšíření o značky, kategorie produktů, správce měn a další.

POUŽITÁ LITERATURA

- [1] What is a REST API? *Red Hat*. [online]. Copyright © 2021 Red Hat, Inc. [cit. 8. 12. 2021]. Dostupné z: <https://www.redhat.com/en/topics/api/what-is-a-rest-api>
- [2] HOLUBOVÁ, Irena, Martin NEČASKÝ, Jaroslav POKORNÝ, Karel RICHTA, Kamil TOMAN a Vojtěch TOMAN. *XML Technologie: Principy a aplikace v praxi*. Praha: Grada Publishing., 2008. ISBN 978-80-247-6688-1.
- [3] XML Essentials - W3C. *World Wide Web Consortium (W3C)* [online]. [cit. 3. 8. 2021]. Dostupné z: <https://www.w3.org/standards/xml/core>
- [4] XML File Format. *File Format Docs* [online]. [cit. 3. 8. 2021]. Dostupné z: <https://docs.fileformat.com/web/xml/>
- [5] JOSHI, Bipin. *Beginning XML with C# 7: XML processing and data access for C# developers*. Second edition. New York, NY: Apress, [2017]. ISBN 978-1-4842-3105-0.
- [6] XML Processing Options | *Microsoft Docs*. [online]. Copyright © Microsoft 2021 30. 3. 2017 [cit. 3. 8. 2021]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/standard/data/xml/xml-processing-options>
- [7] XML Path Language (XPath). *World Wide Web Consortium (W3C)* [online]. Copyright © [cit. 3. 8. 2021]. Dostupné z: <https://www.w3.org/TR/1999/REC-xpath-19991116/>
- [8] BŘÍZA, Petr. Základy jazyka XPath | Interval.cz. *Interval.cz | Svět Internetu, Technologii a Bezpečnosti* [online]. 9. 4. 2009 [cit. 3. 8. 2021]. Dostupné z: <https://www.interval.cz/clanky/zaklady-jazyka-xpath/>
- [9] What is a Relational Database? – Amazon Web Services (AWS). *Cloud Computing Services - Amazon Web Services (AWS)* [online]. Copyright © 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved. [cit. 3. 8. 2021]. Dostupné z: <https://aws.amazon.com/relational-database/>
- [10] RDBMS (Relational Database Management System) Definition. *The Tech Terms Computer Dictionary* [online]. Copyright © 2021 Sharpened Productions. aktualizováno: 16. 12. 2017 [cit. 3. 8. 2021]. Dostupné z: <https://techterms.com/definition/rdbms>
- [11] DYER, Russell J. T. *Learning MySQL an MariaDB: Heading in the right direction with MySQL and MariaDB*. Sebastopol: O'Reilly Media, 2015. ISBN 978-1-449-36290-4.
- [12] RAVI, Saive. What is MariaDB? How Does MariaDB Work?. *Tecmint: Linux Howtos, Tutorials & Guides* [online]. Copyright © 2021. All Rights Reserved. 25. 9. 2020 [cit. 3. 8. 2021]. Dostupné z: <https://www.tecmint.com/what-is-mariadb-how-does-mariadb-work/>
- [13] MariaDB Foundation - MariaDB.org. *MariaDB Foundation - MariaDB.org* [online]. [cit. 3. 8. 2021]. Dostupné z: <https://mariadb.org/>

- [14] ČÁPKA, David. Lekce 1 - Úvod do WPF (Windows Presentation Foundation). *itnetwork.cz - Ajtácká sociální síť a materiálová základna pro C#, Java, PHP, HTML, CSS, JavaScript a další*. [online]. Copyright © 2021 itnetwork.cz. [cit. 3. 8. 2021]. Dostupné z: <https://www.itnetwork.cz/csharp/wpf/c-sharp-tutorial-wpf-uvod-a-prvni-formularova-aplikace/>
- [15] NATHAN, Adam. *WPF 4.5 Unleashed*. Indiana: Sams Publishing, 2013. ISBN 978-0672336973.
- [16] What is Entity Framework?. *Entity Framework Tutorial* [online]. [cit. 3. 8. 2021]. Dostupné z: <https://www.entityframeworktutorial.net/what-is-entityframework.aspx>
- [17] ČÁPKA, David. Lekce 7 - LINQ v C# .NET - Revoluce v dotazování. *itnetwork.cz - Ajtácká sociální síť a materiálová základna pro C#, Java, PHP, HTML, CSS, JavaScript a další*. [online]. Copyright © 2021 itnetwork.cz. Veškerý obsah webu [cit. 3. 8. 2021]. Dostupné z: <https://www.itnetwork.cz/csharp/kolekce-a-linq/c-sharp-tutorial-linq-dotazy>
- [18] KUMAR, Vishnujeet. MVVM (Model View ViewModel) Introduction: Part 1. *C# Corner - Community of Software and Data Developers* [online]. Copyright © 2021 [cit. 3. 8. 2021]. Dostupné z: <https://www.c-sharpcorner.com/UploadFile/0b73e1/mvvm-model-view-viewmodel-introduction-part-1/>
- [19] WEIL, Arnaud. *Learn WPF MVVM: XAML, C# and the MVVM pattern*. Lulu Press, 2017. ISBN 978-1-326-84799-9.
- [20] MASSÉ, Mark. *REST API Design Rulebook*. Sebastopol: O'Reilly Media, c2012. ISBN 978-1-449-31050-9.
- [21] Overview of Microsoft Graph - Microsoft Graph. *Microsoft Docs*. [online]. Copyright © Microsoft 2021 [cit. 8. 12. 2021]. Dostupné z: <https://docs.microsoft.com/en-us/graph/overview>
- [22] MFČR. *ARES – Administrativní registr ekonomických subjektu*. [online]. [cit. 3. 8. 2021]. Dostupné z: <http://www.info.mfcr.cz/ares/>
- [23] Hlídač ARES - ekonom.subjekty. *Hlídač státu*. [online]. [cit. 13. 12. 2021]. Dostupné z: <https://www.hlidacstatu.cz/statniweby/info/10174?h=3B49A8D0E30C9C366B280871C14D24AB>
- [24] ADIS. *GIST* [online]. Copyright © 2014 [cit. 3. 8. 2021]. Dostupné z: <https://www.gist.cz/cz/software/adis>
- [25] MFČR. Informace o webové službě pro zjišťování spolehlivosti plátců DPH a bankovních účtů plátců DPH – technické parametry. *Daňový portál – Elektronické služby Finanční správy České republiky*. [online]. [cit. 3. 8. 2021] Dostupné z: https://adisspr.mfcr.cz/adistc/adis/idpr_pub/dpr_info/ws_spdph.faces
- [26] Rady systému POHODA 2022. *POHODA - ekonomický a informační systém* [online]. Copyright © 2021 STORMWARE s.r.o. [cit. 16. 12. 2021]. Dostupné z: <https://www.stormware.cz/pohoda/rady/>

[27] Účetní a evidenční systém EKONOM System. *Účtujte jednoduše | Účetní a evidenční systém EKONOM System* [online]. Copyright © 2020 Všechna práva vyhrazena [cit. 19. 12. 2021]. Dostupné z: <https://ekonom-system.cz/>

[28] DUEL | Ježek software. *Ježek software* [online]. Copyright © 2021, Ježek software s.r.o. [cit. 20. 12. 2021]. Dostupné z: <https://www.jezeksw.cz/duel/>