FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Rozšíření nástroje pro vyhledávání osobních údajů
Student:	Tomáš Chvosta
Vedoucí:	Ing. Jiří Mlejnek
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2019/20

Pokyny pro vypracování

Seznamte se s bakalářskou prací Davida Skalského [1], která popisuje možnosti vyhledávání osobních údajů v informačních systémech. Na základě popisu uvedeného v této práci vytvořte návrh rozšíření existujícího nástroje pro vyhledávání osobních údajů. Zaměřte se na vyhledávání nových typů osobních údajů, jejichž vyhledávání dosud nebylo implementováno (telefonní čísla, biometrické údaje, informace o zdravotním stavu, apod.). Rozsah implementace konzultujte nejprve s vedoucím práce.
Dále proveďte návrh a implementaci řešení, které by umožnilo vyhledávat osobní údaje také v dokumentech ukládaných v databázi. Návrh proveďte se zaměřením na snadnou rozšiřitelnost o nové typy podporovaných formátů souborů.
Implementované řešení důkladně otestujte z pohledu výkonu. Na základě provedených testů identifikujte části procesu, které celkovou dobu, potřebnou pro prohledání databáze, nejvíce ovlivňují.

Seznam odborné literatury

[1] Skalský, David: *Vyhledávání osobních údajů v relačních databázích*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018

Ing. Michal Valenta, Ph.D.
vedoucí katedry

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

V Praze dne 29. ledna 2019

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Bakalářská práce

Rozšíření nástroje pro vyhledávání osobních údajů

Katedra softwarového inženýrství
Vedoucí práce: Ing. Jiří Mlejnek

15. května 2019

Poděkování

Velké poděkování patří vedoucímu práce Ing. Jiřímu Mlejnkovi za stálou odbornou pomoc a podporu v průběhu psaní této práce.

Prohlášení

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

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne 15. května 2019 ……

Odkaz na tuto práci

Chvosta, Tomáš. *Rozšíření nástroje pro vyhledávání osobních údajů.* Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2019.

Abstrakt

V této bakalářské práci jsou zkoumány osobní údaje podle GDPR (nařízení EU 2016/679 upravující náležitosti ohledně zpracování osobních údajů) z pohledu jejich obecné charakteristiky, ale také z pohledu uložení těchto údajů ve strukturovaných a nestrukturovaných datech. Dále je součástí návrh a implementace rozšíření již existujícího nástroje Winch pro vyhledávání osobních údajů a jejich anonymizaci. Implementační část rozšiřuje nástroj především v oblasti aktivního vyhledávání v nestrukturovaných datech, ale i ve strukturovaných datech, jako jsou databáze. Aplikační část využívá převážně technologii Java, konkrétně se jedná o jazyk Groovy. Pro uživatelské nastavení celého procesu vyhledávání je vytvořeno jednoduché grafické uživatelské rozhraní v jazyce C#. Poslední část je věnována testování funkčnosti a výkonu implementovaného řešení.

Klíčová slova GDPR, osobní údaje, Winch, nestrukturovaná data, rozšíření nástroje Winch, Groovy, C#, Microsoft SQL, Oracle SQL, Postgre SQL, QR kód

Abstract

This bachelor thesis is focused on research of the personal data under General Data Protection Regulation EU 2016/679 (GDPR) from the perspective of their general characteristics and also from the perspective of saving those informations in structuralized and unstructured data. Next part of the thesis is a suggestion and of the extension of already existing Winch tool for searching and anonymising of personal data. Implementation part extends the tool in areas of active search in unstructured data, but also in structuralized one i.e. database. Application part mostly uses Java technology, specifically Groovy language. For the user settings of the whole process of searching a simple graphic user interface is created in C# language. The last part of thesis is focused on testing of the functionality and performance of the implemented extension.

Keywords GDPR, personal data, Winch, unstructured data, Winch extension, Groovy, C#, Microsoft SQL, Oracle SQL, Postgre SQL, QR code

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Dne 27. dubna 2016 bylo schváleno Obecné nařízení o ochraně osobních údajů (zkráceně GDPR), plným názvem Nařízení Evropského parlamentu a Rady (EU) č. 2016/679 ze dne 27. dubna 2016 o ochraně fyzických osob v souvislosti se zpracováním osobních údajů, o volném pohybu těchto údajů a o zrušení směrnice 95/46/ES (obecné nařízení o ochraně osobních údajů). GDPR je v celé Evropské unii jednotně účinné od 25. května 2018. Samotné nařízení významně zvyšuje ochranu práv subjektů osobních údajů, kterými jsou zejména právo na přístup, opravu, anonymizaci, pseudonymizaci, omezení zpracování nebo vymazání. Ochrana fyzických osob v souvislosti se zpracováním osobních údajů je totiž základním lidským právem ukotveným v listinách základních práv a svobod EU i ČR.

Vedle posilování práv fyzických osob však toto nařízení současně stanovuje nové požadavky na zpracovatele osobních údajů (správce), kteří musí zajistit, aby zpracování osobních údajů fyzických osob bylo prováděno zákonným, spravedlivým a transparentním způsobem a rozsah tohoto zpracování byl nezbytně nutný a přiměřený pro dané účely a byl v souladu s obecně závaznými právními předpisy. Ve vztahu k GDPR potřebují společnosti zmapovat současný systém sběru dat i způsoby evidence osobních údajů v interních systémech a stanovit zásady pro správné zpracování osobních údajů. Negativní motivací k rychlému stanovení zásad práce s osobními údaji subjektů jsou pro správce údajů velmi tvrdé sankce obsažené v legislativě pro případy porušení stanovených povinností.

Legislativní změny tak kladou značné nároky na společnosti spravující osobní data, které musí ve velmi krátké době zajistit a změnit své postupy při řízení datových toků a správě osobních údajů. Pro usnadnění těchto kroků existuje nástroj Winch (byl vyvíjen v rámci jiných bakalářských prací na Fakultě informačních technologií na ČVUT v Praze) umožňující vyhledávání osobních údajů subjektů v databázích a následnou anonymizaci dat. Hlavním úkolem této práce je rozšíření funkcionality tohoto nástroje pro snazší identifikaci

osobních údajů ve strukturovaných datech v databázích, ale i v nestrukturovaných datech, jako jsou například dokumenty uložené v databázích. To by umožnilo menším a středním firmám rychlé uvedení zásad správného nakládání s osobními údaji subjektů do praxe v souladu s platnou legislativou.

Kapitola 1

Cíl práce

Cílem bakalářské práce je shrnout problematiku ochrany osobních údajů (Nařízení Evropského parlamentu a Rady (EU) č. 2016/679 ze dne 27. dubna 2016) se zaměřením na definici osobních údajů subjektů a jejich vyhledávání ve strukturovaných datech v databázích, ale i v nestrukturovaných datech, jako jsou například dokumenty obsahující smlouvu apod. Soubory s nestrukturovanými daty mohou být uloženy přímo v databázi nebo na filesystému.

Problematiku částečně řeší bakalářská práce Davida Skalského[4], která popisuje možnosti vyhledávání osobních údajů v informačních systémech. Cílem je tedy i seznámit se s touto prací a na základě popisu uvedeného v této práci vytvořit návrh rozšíření existujícího nástroje Winch pro vyhledávání osobních údajů.

Součástí práce je také vytvoření testovacích dat, na kterých bude samotné řešení otestováno z pohledu výkonu.

Kapitola 2

GDPR a definice základních pojmů

V této kapitole jsou shrnuty důležité základní pojmy týkající se této práce a dále je zde popsáno nařízení GDPR.

2.1 GDPR

„Nařízení (EU) 2016/679 (ÚOOÚ[1] dává k dispozici úplné znění GDPR[2] po zapracování opravy) představuje právní rámec ochrany osobních údajů platný na celém území EU, který hájí práva jejích občanů proti neoprávněnému zacházení s jejich daty a osobními údaji. GDPR přebírá všechny dosavadní zásady ochrany a zpracování údajů, na nichž unijní systém ochrany osobních údajů stojí a potvrzuje, že ochrana cestuje přes hranice současně s osobními údaji.“[5]

Současně s tím obecné nařízení rozvíjí a posiluje práva osob dotčených zpracováním. Konkrétně se jedná o právo získávat informace o tom, jaké jejich údaje jsou zpracovávány včetně důvodu zpracování, a domáhat se dodržování pravidel, včetně nápravy stavu. GDPR klade systematicky důraz na vymahatelnost práv lidí a povinností správců (subjektů odpovědných za zpracování). Obsahuje proto propracovanější a náročnější pravidla pro zvláštní kategorie údajů a zpracování a současně vynucuje od správců i zpracovatelů výrazně aktivnější přístup. V praxi se jedná zejména o povinnost posoudit vliv jednotlivých zpracování na ochranu osobních údajů (DPIA[3]) před zahájením nového zpracování a k tomu volit vhodné nástroje ochrany údajů. Klíčem k nastavení povinností pro správce je rizikovost, která je dovozována z rozsahu zpraco-

[1] Úřad pro ochranu osobních údajů
[2] General Data Protection Regulation
[3] Data Protection Impact Assessment

vání a obsahu případně typů zpracovávaných osobních údajů a používaných technologií.[5]

2.2 Základní pojmy

Osobní údaje: Osobní údaje jsou jakékoli informace o identifikovaném nebo identifikovatelném subjektu údajů. Identifikovatelnou fyzickou osobou je fyzická osoba, kterou lze přímo či nepřímo identifikovat, zejména odkazem na určitý identifikátor (jméno, číslo, síťový identifikátor) nebo na jeden či více zvláštních prvků fyzické, fyziologické, genetické, psychické, ekonomické, kulturní nebo společenské identity této fyzické osoby.[6] Mezi obecné osobní údaje řadíme jméno, pohlaví, věk a datum narození, osobní stav, ale také IP adresu a fotografický záznam. Vzhledem k tomu, že se GDPR vztahuje i na podnikající fyzické osoby, řadíme mezi osobní údaje i tzv. organizační údaje, kterými jsou například e-mailová adresa, telefonní číslo či různé identifikační údaje vydané státem. Mezi osobní údaje patří i zvláštní kategorie osobních údajů, jedná se o tzv. citlivé údaje.[7]

Popisné údaje: Do kategorie popisných neboli charakterizačních údajů se zahrnují údaje, které utváří hlubší a ucelenější obraz subjektů údajů.[8]

Citlivé údaje: Citlivé osobní údaje jsou speciální kategorií podle GDPR, která zahrnuje údaje o rasovém či etnickém původu, politických názorech, náboženském nebo filozofickém vyznání, členství v odborech, o zdravotním stavu, sexuální orientaci a trestních deliktech či pravomocném odsouzení osob. Tyto údaje mohou v případě zneužití subjekt údajů samy o sobě poškodit ve společnosti, v zaměstnání, ve škole či mohou zapříčinit jeho diskriminaci. Kategorie citlivých údajů GDPR nově zahrnuje genetické a biometrické údaje. Zpracování citlivých osobních údajů podléhá mnohem přísnějšímu režimu, než je tomu u obecných osobních údajů.[9]

Strukturovaná data: Strukturovaná data jsou data, která jsou specificky označena a strukturována a umožňují tak lépe pochopit, parsovat a interpretovat zdrojový kód. Například se jedná o organizaci pomocí relačních databázových systémů. Zde se používá hierarchie elementů od pole k záznamu, relaci až k databázi. V takto strukturovaných datech se lépe vyhledává a také se s nimi dále snáze pracuje (MARC[4], XML).[10]

Nestrukturovaná data: Nestrukturovaná data jsou data, která nejsou mezi sebou rozlišená, tzv. „tok bytů“ (prostý text) a lze v se v nich orientovat pouze pomocí fulltextového vyhledávání.[10]

[4]MAchine-Readable Cataloging (Strojově čitelná katalogizace)

Kapitola 3

Analýza rozšíření

Tato kapitola rekapituluje současný stav práce Davida Skalského[4] a analýzu nových osobních údajů, které dosud nebyly zpracovány nebo implementovány.

3.1 Analýza a shrnutí současného stavu práce

Jedním z úkolů této práce je seznámit se s bakalářskou prací Davida Skalského. Jmenovaný se v této práci zabýval výzkumem některých osobních a citlivých údajů, konkrétně se jednalo o jméno a příjmení, pohlaví, rodné číslo, adresu, poštovní směrovací číslo, datum narození, identifikační číslo osoby, daňové identifikační číslo, telefonní číslo, e-mailovou adresu, IP adresu, číslo občanského průkazu, GPS souřadnice, biometrické údaje, otisky prstů, rozpoznání obličeje, rasové a etnické údaje, státní příslušnost, náboženská víra a sexuální orientace. Všechny tyto údaje jsou popsané v textu jeho bakalářské práce a některé z nich se mu povedlo implementovat. Jedná se o vyhledávače DIČ, pohlaví, poštovního směrovacího čísla, IP adresy, čísla občanského průkazu a některých citlivých údajů.

Některé z výše uvedených údajů nejsou v jeho řešení vůbec implementovány, část údajů v textu bakalářské práce postrádá důležité informace pro správnou implementaci, například GPS souřadnice se mohou vyskytovat v databázích ve více formátech, než zmiňuje David Skalský. Dále například údaj sexuální orientace lze rozšířit více základními typy. Práce velmi dobře popisuje vyhledávání údajů v databázích. Vyhledávání osobních údajů v nestrukturovaných datech je sice v textu práce částečně rozebráno, v implementaci však značná část chybí.

Na základě těchto zjištění a po domluvě s vedoucím práce jsem se rozhodl provést analýzu a implementaci některých osobních údajů z práce Davida Skalského, ale i některých nových osobních a citlivých údajů, které v práci nejsou řešeny. Dále je třeba provést kompletní návrh a implementaci vyhledávání osobních údajů v nestrukturovaných datech.[4]

3.2 Analýza nových osobních údajů

V této kapitole jsou zpracovány osobní, citlivé a popisné údaje, které bude možné nově vyhledávat pomocí nástroje Winch. U každého údaje jsou zkoumány obecné vlastnosti, informace ohledně výskytu v databázích, tedy jaké jsou obvyklé názvy sloupců s daným údajem a jaká slova jsou často obsažena v komentářích sloupců. U každého údaje je popsán způsob, jakým ho lze ověřit, zda je třeba ho porovnávat se slovníkem nejčastějších hodnot nebo zda lze použít validační funkce nebo regulární výraz. Dále je zkoumáno, jestli existuje mezi jednotlivými údaji nějaká vazba. Veškeré tyto principy jsou popsané ve vztahu k databázi, jejíž názvy jsou převážně v českém jazyce případně základní angličtině.

Některé údaje, které jsou zpracovány v této kapitole, nemusí samy o sobě představovat osobní údaje, avšak ve spojení s jinými údaji či informacemi mohou umožnit identifikovat konkrétní osobu. V takových případech se podle zákonných definic společně s těmito dalšími údaji dají považovat za osobní údaje.

Následující údaje byly vybrány a schváleny na základě domluvy s vedoucím bakalářské práce.

3.2.1 GPS souřadnice

GPS[5] souřadnice jsou údaj udávající polohu bodu na zemi. Skládají se ze dvou číselných údajů, severní (N) či jižní (S) zeměpisné šířky (latitude) a západní (W) či východní (E) zeměpisné délky (longitude). Zeměpisná šířka určuje, na jaké rovnoběžce se dané místo nachází, a může nabývat hodnot 0° až 90°. Zeměpisná délka určuje, na jakém poledníku se místo nachází, a může nabývat hodnot 0° až 180°. Pro potřeby uživatelů GPS je nejčastěji užívaný geografický referenční systém WGS 84, známý také pod kódem EPSG:4326.

Nejrozšířenějšími formáty na území České republiky jsou první tři formáty z tabulky 3.1 (hddd.ddddd°, hddd°mm.mmm', hddd°mm'ss.s"). V databázích se může stát, že jednotlivé složky souřadnic (zeměpisná výška a šířka) budou uloženy v samostatném sloupci.[11]

Identifikace bude probíhat regulárním výrazem. Pro první tři formáty v tabulce lze použít následující regulární výrazy:

hddd.ddddd°: ^(N){0,1}(\\s*)((([0-9]|[0-8][0-9])\\.(\\d+))|(90|90.(0*)))(°){0,1}(\\s*)(N){0,1}(\\s*)(,|;){0,1}(\\s*)(E){0,1}(\\s*)((([0-9]|[0-9][0-9]|[0-1][0-7][0-9])\\.(\\d+))|(180|180.(0*)))(°){0,1}(\\s*)(E){0,1}\$

hddd°mm.mmm': ^(N){0,1}(\\s*)((([0-9]|[0-8][0-9])(°){0,1}(\\s*)([0-9]|[0-5][0-9])\\.(\\d+))|(90(°){0,1}(\\s*)(0|00)\\.(0*)))('|´){0,1}(\\s*)(N){0,1}(\\s*)(,|;){0,1}(\\s*)(E){0,1}(\\s*)((([0-9]|[0-9][0-9]|[0-1][0-7][0-9])

[5] Global Positioning system (Globální polohový systém)

(°){0,1}(\\s*)([0-9]|[0-5][0-9])\\d+))|(180(°){0,1}(\\s*)(0|00)\\.(0*)))(’|´){0,1}(\\s*)(E){0,1}\$

hddd°mm’ss.s” ^(N){0,1}(\\s*)((([0-9]|[0-8][0-9])(°){0,1}(\\s*)([0-9]|[0-5][0-9])(’|´){0,1}(\\s*)([0-9]|[0-5][0-9])\\.(\\d+))|(90(°){0,1}(\\s*)(0|00)(’|´){0,1}(\\s*)(0|00)\\.(0*)))(”|”|´´){0,1}(\\s*)(N){0,1}(\\s*)(,|;){0,1}(\\s*)(E){0,1}(\\s*)((([0-9]|[0-9][0-9]|[0-1][0-7][0-9])(°){0,1}(\\s*)([0-9]|[0-5][0-9])(’|´){0,1}(\\s*)([0-9]|[0-5][0-9])\\.(\\d+))|(180(°){0,1}(\\s*)(0|00)(’|´){0,1}(\\s*)(0|00)\\.(0*)))(”|”|´´){0,1}(\\s*)(E){0,1} \$

Tabulka 3.1: Nejznámější formáty GPS souřadnic[2]

Název formátu	**Příklad hodnoty v DB**
hddd.ddddd°	N 49.63117° E 014.05898°
hddd°mm.mmm’	N 49°37.870’ E 014°03.539’
hddd°mm’ss.s”	N 49°37’52.2” E 014°03’32.3”
Rakouská souřadnicová síť	M31 502478 499598
Holandská souřadnicová síť	780755 219015
Maďarská souřadnicová síť EOV	E289549 N488134
Finská souřadnicová síť	25566609 5580914
Německá souřadnicová síť	5432156 5499801
ITM	2186093 0557684
MGRS	33U VQ 32046 98049
RT 90	X 5500653 Y 1373837
SWEREF 99 TM	N5498049 E0432046
Souřadnicová síť US National	33UVQ 32046 98049
UTM UPS	33U 0432046 5498049
GARS	389MR34
Maidenhead	JN79AP71BL85
GEOREF	NKQE03533787

Názvy sloupce Očekávané názvy sloupce v českém jazyce jsou „GPS“, „SOURADNICE“, „GPS_SOURADNICE“, „POLOHA“, v angličtině pak „COORDINATES“, „GPS_COORDINATES“ a „LOCATION“

Komentáře sloupce Komentáře jsou prohledávány na částečnou shodu s frázemi „gps“, „souřadnice“, „poloha“, „coordinates“ a „location“.

3.2.2 Identifikační číslo vozidla

VIN[6] je mezinárodně jednoznačný identifikátor motorových vozidel. Číslo je tvořeno 17 písmeny a číslicemi, jeho formát je od roku 1983 určen normou (ISO 3779:1983).[12]

[6]Vehicle identification number (Identifikační číslo vozidla)

První tři znaky se označují jako WMI[7]. Z tohoto kódu lze jednoznačně určit světového výrobce vozidla. Pokud se jedná o malosériového výrobce, pak se jako třetí znak používá 9 a další tři znaky výrobce jsou na pozicích 13, 14, 15. Velkým výrobcům je naopak přiděleno více kódů. Část WMI je jedinou povinnou částí kódu. První znak kódu určuje region výrobce, druhý znak určuje stát, ve kterém výrobce působí:

Tabulka 3.2: Příklady světových regionů a výrobců

WMI	**Region**	**Příklady zemí**
A-H	Afrika	AA-AH (Jihoafrická republika)
J-R	Asie	KL-KR (Jižní Korea)
S-Z	Evropa	TJ-TP (Česká republika)
1-5	Severní Amerika	1, 4, 5 (USA)
6-7	Austrálie a Oceánie	7A-7E (Nový Zéland)
8-0	Jižní Amerika	9A-9E, 93-99 (Brazílie)

Znaky na pozicích 4 až 9 se označují jako VDS[8] a dají se z nich vyčíst informace o modelu vozidla. Tento kód si každý výrobce určuje sám podle vlastní volby.[13]

V některých zemích je číslice 9 vyhrazena jako kontrolní číslice. Ta zajišťuje, že špatně zapsaný VIN kód je odhalen jako neplatný. Pro výpočet kontrolní číslice se určí hodnota každého znaku (číslice používají svou vlastní hodnotu): A=1, B=2, C=3, D=4, E=5, F=6, G=7, H=8, J=1, K=2, L=3, M=4, N=5, P=7, R=9, S=2, T=3, U=4, V=5, W=6, X=7, Y=8, Z=9. Poté se hodnota každého znaku vynásobí vahou odpovídající pozici znaku ve VIN (s výjimkou samotné kontrolní číslice):

Tabulka 3.3: Váhy pozicí znaků ve VIN

1. 8x	5. 4x	10. 9x	14. 5x
2. 7x	6. 3x	11. 8x	15. 4x
3. 6x	7. 2x	12. 7x	16. 3x
4. 5x	8. 10x	13. 6x	17. 2x

Všechny součiny se sečtou a výsledný součet je vydělen číslem 11. Výsledný zbytek po dělení tvoří kontrolní číslici, pokud vyjde výsledek 10, použije se písmeno „X“[14].

Znaky na pozicích 10 až 17 se označují jako VIS[9] a tvoří pořadové výrobní číslo, které jednoznačně identifikuje konkrétní vozidlo. Přidělování těchto čísel je čistě v režii výrobce. Některé znaky VIS však mají speciální význam. Znak

[7]World Manufacturer Identifier (Světový kód výrobce)
[8]Vehicle Descriptor Section (Popisný kód vozidla)
[9]Vehicle identifier Selection

na 10. pozici se velmi často používá pro určení roku výroby. Znak A je rok 1980, B je rok 1981 atd. až po Y, což je rok 2000. Další roky jsou reprezentovány číslicemi 1 až 9. Pro rok 2010 se opět používá znak A, pro 2011 znak B atd. Znak na 11. pozici většinou určuje výrobní závod výrobce. Kupříkladu u vozů Škoda znak 0 označuje závod v Mladé Boleslavi, znak 5 továrnu v Kvasinkách, znak 7 výrobní závod Vrchlabí.[13]

Formát VIN je tedy 17 znaků, které mohou být tvořené číslem nebo velkým písmenem s výjimkou písmen „I“, „O“, „Q“, které lze snadno zaměnit za číslice 1, respektive 0. Je tedy možné použít následující regulární výraz:

- ^[A-Z0-9&&[^IOQ]]{17}$

Tento výraz však může identifikovat jako VIN i jiné kódy obsahující 17 znaků a splňující daná kritéria. Daleko vhodnější by bylo použít validační funkci, která by vypočetla hodnotu kontrolní číslice podle algoritmu popsaného výše a tuto hodnotu porovnala se znakem na 9. pozici. Tato funkce by však v praxi mohla vyloučit mnoho platných VIN, protože mnoho světových výrobců kontrolní číslici nepoužívá. Tato skutečnost byla ověřena na základě datasetu vozidel s 10000 reálnými záznamy, který pro tuto práci poskytla společnost zabývající se pojištěním majetku. Vzhledem k povaze dat není možné tento dataset publikovat. Jelikož se jednalo o reálná data, nebyly všechny údaje o vozidlech vyplněny korektně, mnoho VIN kódů u vozidel chybělo. Na těchto datech byla vyzkoušena validační funkce i regulární výraz uvedený výše. Regulární výraz odhalil všechny správně zapsané VIN, což odpovídalo zhruba 98% záznamů. Oproti tomu validační funkce odhalila pouze 41% správně zapsaných VIN.

Názvy sloupce Očekávané názvy sloupce v českém jazyce jsou „VIN“, „VIN_KOD“, v angličtině pak „VEHICLE_IDENTIFICATION_NUMBER“.

Komentáře sloupce Komentáře jsou prohledávány na částečnou shodu s frází „vin“.

3.2.3 Státní poznávací značka

Státní poznávací značka (zkratka SPZ nebo registrační značka RZ) je jednoznačně označení motorového vozidla nebo přívěsu či návěsu. V tomto označení se mohou vyskytovat písmena a číslice. V České republice jsou značky vydávány podle zákona č. 56/2001 Sb., o podmínkách provozu vozidel na pozemních komunikacích, účinný od 1. července 2001 a jejich podoba je předepsána vyhláškou č. 343/2014 Sb., o registraci vozidel, účinné od 1. ledna 2015.

U původních československých státních poznávacích značek byla skupina dvou až tří písmen a za ní dvě dvojice čísel, kdy skupiny byly odděleny pomlčkami. Skupina prvních dvou písmen označovala okres, třetí písmeno označovalo sérii. Každému okresu byla přidělena dvě písmena, více obydlené okresy

mohly mít více kombinací. Praha měla jen jedno písmeno A, další dvě písmena byla sériová. Každému okresu byla přidělena dvě písmena (například PB-Příbram), některé okresy s vyšším počtem obyvatel mohly mít dvě, nebo dokonce tři kombinace (například PA, PU).

Od roku 2001 se používají nové státní poznávací značky, u kterých se nerozlišuje okres, ale pouze kraj, ve kterém má majitel vozidla registrovánu adresu. Od začátku roku 2015 se nemění registrační značka, pokud je vozidlo dříve registrováno v jiném kraji. Kraje nahradily okresy zejména kvůli nedostatku kombinací pro jednotlivé okresy. Dnešní značky obsahují podle typu vozidla 5 až 7 znaků (číslice a písmena bez diakritiky s výjimkou písmen G, O, Q, W). Značky musí obsahovat alespoň jedno písmeno a jednu číslici. První písmeno zleva je vždy kód kraje. Písmena a číslice mohou být v různém pořadí např. 1AD 02A4, 6S9 945Z, 111 AB01 apod.

Od začátku roku 2016 existují tzv. značky na přání s vlastním textem. Na státní poznávací značce může být 7-8 znaků (písmena a číslice) obsahujících alespoň jednu číslici. Značky nemohou obsahovat písmena G, O, Q, W a také vulgární nebo hanlivé výrazy a názvy úřadů.

Od podzimu roku 2018 se používají ekologické registrační značky pro elektromobily a hybridy, díky kterým budou mít majitelé vozidel spoustu výhod (vyhrazené jízdní pruhy, snazší vjezd do městských center, bezplatný vjezd na dálnice a jiné). Poznávací značka by začínala dvojicí písmen EL a kombinací číslic.[15][16][17]

Tabulka 3.4: Tabulka kódů krajů v ČR[3]

Kód	**Kraj**	**Nejvyšší řada**
A	Praha	7AM
B	Jihomoravský kraj	9B9
C	Jihočeský kraj	8C5
E	Pardubický kraj	6E4
H	Královéhradecký kraj	7H3
J	Vysočina	6J6
K	Karlovarský kraj	4K5
L	Liberecký kraj	5L9
M	Olomoucký kraj	6M8
P	Plzeňský kraj	7P7
S	Středočeský kraj	9S4
T	Moravskoslezský kraj	6T5
U	Ústecký kraj	9U9
Z	Zlínský kraj	6Z5

SPZ je možné identifikovat pomocí validační funkce, která vyzkouší, zda se jedná o starý nebo nový formát státních poznávacích značek. V případě starého

formátu zkontroluje okres podle slovníku okresů a následně zbytek SPZ, zda odpovídá předepsanému formátu. Slovník okresů je možné vytvořit na základě seznamů, které je možné vyhledat na internetu.[18] V případě nového formátu zkontroluje kraj a následně může porovnat první část SPZ (první tři znaky) se současnou nejvyšší řadou. Pokud je vyšší, je SPZ vyhodnocena jako neplatná. Dále je samozřejmě zkontrolován zbytek SPZ, zda odpovídá předepsanému formátu.

Tato validační funkce však neodhalí nové značky na přání a ekologické registrační značky. Nicméně tyto značky se v praxi vyskytují minimálně a pro přesnější identifikaci je vhodné je z procesu validace vyloučit. Tato skutečnost byla ověřena na základě stejného datasetu vozidel, který byl použit i pro analýzu VIN kódu. Bylo zjištěno, že nové SPZ na přání a ekologické registrační značky se vyskytují v méně než 1% dat. Dá se tedy usoudit, že platnou SPZ identifikuje validační funkce s více než 99% pravděpodobností.

Názvy sloupce Očekávané názvy sloupce v českém jazyce jsou „SPZ“, „RZ“ „POZNAVACI_ZNACKA“, „STATNI_POZNAVACI_ZNACKA“, „REGISTRACNI_ZNACKA“ v angličtině pak „NUMBER_PLATE“, „VEHICLE_REGISTRATION_PLATE“, „LICENSE_PLATE“.

Komentáře sloupce Komentáře jsou prohledávány na částečnou shodu s frázemi „spz“, „rz“, „poznávací značka“, „státní poznávací značka“, „registrační značka“, „rozeznávací značka“, „rejstříková značka“, „evidenční značka“, „policejní značka“, v angličtině „vehicle registration plate“, „number plate“, „license plate“.

3.2.4 Zdravotní stav

V některých databázích a dokumentech se můžeme setkat s údaji, které se týkají zdravotního stavu člověka. Konkrétně se může jednat například o onemocnění, užívaný lék, léčbu a jiné. Všeobecná zdravotní pojišťovna České republiky poskytuje bezplatně na svých oficiálních webových stránkách číselníky týkající se zdravotní péče. Z těchto číselníků lze pro vyhledávání údajů týkajících se zdravotního stavu využít číselník diagnóz a číselník hromadně vyráběných léčivých přípravků a potravin pro zvláštní lékařské účely.[19]

Ideálním způsobem identifikace těchto údajů bude porovnání hledaného výrazu s hodnotami číselníků diagnóz a léků (hromadně vyráběných léčivých přípravků a potravin pro zvláštní lékařské účely). Tato metoda funguje spolehlivě pro identifikaci diagnóz. U léků je třeba brát ohled na to, že v databázích, lékařských zprávách a jiných dokumentech může být údaj uložen společně s informací o množství účinné látky obsažené v léku. Tyto informace se v číselníku nacházejí také, jsou však obsažené v jiném sloupci než název léku. Číselníky dostupné na stránkách Všeobecné zdravotní pojišťovny jsou seřazeny podle názvů léků, proto je nejefektivnějším řešením oddělit název léku od ostatních informací a ten následně hledat v číselníku léků.

Názvy sloupce Očekávané názvy sloupce v českém jazyce jsou „ZDRAVOTNI_STAV“, „ANAMNEZA“, „ONEMOCNENI“, „LECIVO“, „LEK“, „LEKY“, „NEMOC“, „DIAGNOZA“ v angličtině pak „HEALTH_CONDITION“, „ANAMNESIS“, „MEDICINE“, „DRUG“, „DISEASE“, „MEDICAMENT“, „ILLNESS“, „DIAGNOSIS“ .

Komentáře sloupce Komentáře jsou prohledávány na částečnou shodu s frázemi „zdravotní“, „zdravotní stav“, „anamnéza“, „lék“, „léky“, „léčivo“, „nemoc“, „onemocnění“, „diagnóza“, „choroba“ v angličtině „health“, „health condition“„anamnesis“, „medicine“, „drug“, „medicament“„disease“, „illness“, „diagnosis“.

3.2.5 Sexuální orientace

Sexuální orientace je osobní údaj, který označuje sexuální preference jednotlivce. V současné době existuje pouze několik málo typů sexuální orientace:

- Heterosexualita
- Homosexualita
- Bisexualita
- Pansexualita
- Asexualita
- Demisexualita
- Androsexualita
- Gynosexualita
- Androgynosexualita
- Omnisexualita
- Sapiosexualita
- Objectumsexualita
- Autosexualita
- Polysexualita

Informace a vysvětlení pojmů týkající se jednotlivých sexuálních orientací jsou uvedeny v příloze C.

K identifikaci tohoto údaje je možné využít slovník, ve kterém jsou uložené výše zmíněné sexuální orientace a také označení příslušníků těchto orientací. Tento způsob je ideální pro vyhledávání ve strukturovaných datech.

V nestrukturovaných datech je daleko výhodnější vyhledat všechny výskyty řetězce „sex“ a slova, ve kterých se nachází tento řetězec následně porovnat se slovníkem.[20]

Názvy sloupce Očekávané názvy sloupce v českém jazyce jsou „ORIENTACE“, „SEXUALNI_ORIENTACE“ v angličtině pak „ORIENTATION“, „SEXUAL_ORIENTATION“.

Komentáře sloupce Komentáře jsou prohledávány na částečnou shodu s frázemi „orientace“, „sexuální orientace“ v angličtině pak „sexual“, „orientation“, „sexual orientation“.

3.2.6 Číslo cestovního dokladu

Součástí cestovního pasu je tzv. datová strana, na které lze najít veškeré údaje o držiteli dokladu. Jednou z položek na základní datové straně je číslo pasu. U klasického cestovního pasu vydaného v České republice je číslo pasu složeno z 8 číslic a neobsahuje žádnou kontrolní číslici. Od 1. 9. 2006 bylo upraveno číslo pasu u diplomatických a služebních pasů. Číslo diplomatického pasu je sestaven z počátečního písmena D a 7 číslic. číslo služebního pasu je sestaven z počátečního písmena S a 7 číslic.

Každá země EU má vlastní formát čísla pasu. Například u dokladů, vydávaných Slovenskou republikou, obsahuje číslo pasu jeden znak (písmeno nebo číslice) následován sedmi číslicemi (žádné mezery ani oddělovače).[21][22]

V této práci se omezíme pouze na identifikaci čísla cestovních pasů vydávaných Českou a Slovenskou republikou. Formátem čísla je vždy 8 znaků, kdy první znak může být písmeno nebo číslice a zbylé znaky tvoří pouze číslice. Číslo neobsahuje žádný kontrolní znak, není tedy možné identifikovat číslo s naprostou jistotou. Pro identifikaci čísla je použitelný následující regulární výraz:

- ^[-A-Za-z0-9][0-9]{7}$

Názvy sloupce Očekávané názvy sloupce v českém jazyce jsou „CISLO_PASU“, „CISLO_CESTOVNIHO_DOKLADU“ v angličtině pak „PASSPORT_NUMBER“.

Komentáře sloupce Komentáře jsou prohledávány na částečnou shodu s frázemi „číslo pasu“, „číslo cestovního dokladu“, „pas“ v angličtině „passport number“, „passport“.

3.2.7 Telefonní číslo (MSISDN)

MSISDN[10] je osobní údaj, který jednoznačně identifikuje účastníka ve veřejné telefonní síti. Přidělování čísel je určeno standardem E. 164, který definovala

[10] Mobile Subscriber Integrated Services Digital Network Number

Mezinárodní telekomunikační unie. MSISDN ve veřejných sítích může dosáhnout 12 až 15 číslic. Tento číslovací plán definuje ve veřejných telefonních sítích v České republice maximálně 12 číslic. Struktura čísla je následující:

Obrázek 3.1: Struktura čísla MSISDN

Číslo země CC[11] může obsahovat 1 až 3 číslice. České republice byla přidělena hodnota 420. Číslo NDC[12] je národní směrové číslo, které určuje mobilní síť v příslušné zemi. Číslo SN[13] je účastnické číslo, podle kterého je možné určit konkrétního účastníka. V praxi se mnohdy před kódem země používá přestupný znak „+“ nebo „00“, tento znak však není součástí MSISDN. Příklady rozkladu MSISDN[23]:

Tabulka 3.5: Tabulka s příklady MSISDN

Země	MSISDN	CC	NDC	SN
Česká republika	420730123456	420	606	123456
Slovenská republika	421950123456	421	950	123456

V této práci se po konzultaci s vedoucím práce omezíme pouze na identifikaci telefonního čísla mobilních operátorů na území České a Slovenské republiky. Důvodem tohoto omezení je skutečnost, že systém Winch v současné době pracuje převážně s českými a slovenskými databázemi. Formát čísla tvoří 12 číslic, na začátku se navíc před číslem CC mohou vyskytovat symboly „+“ nebo „00“. Číslo CC může nabývat pouze hodnot 420 a 421, v praxi se však telefonní číslo často ukládá bez této hodnoty. Číslo NDC můžeme ověřit podle seznamů předvoleb mobilních operátorů. Seznam pro Českou republiku lze najít na stránkách PREDVOLBY.CZ[24], seznam pro Slovenskou republiku lze najít na stránkách slovenského Úřadu pro regulaci elektronických a poštovních služeb[25]. Tyto seznamy nejsou příliš obsáhlé, proto je možné identifikovat tento údaj pomocí následujícího regulárního výrazu:

- ”^((([+]|00){0,1}420){0,1}(60[1-8]|7(0[235]|20|3[0-4689]|7[0-9]|9[0-379]))|(([+]|00){0,1}421){0,1}(9(0[1-9]|[145][0-9])))[0-9]{6}\$”

[11] Country code
[12] National Destination Code
[13] Subscriber Number

Je však potřeba brát ohled na to, že nalezená čísla nemusejí být platným aktivním číslem českých mobilních operátorů a je třeba následně ověřit, zda se jedná skutečně o existující telefonní číslo.

Názvy sloupce Očekávané názvy sloupce v českém jazyce jsou „TELEFON“, „MOBIL“, „TELEFONNI_CISLO“ „MOBILNI_CISLO“, „MSISDN“ v angličtině pak „TELEPHONE“, „PHONE“, „CELLPHONE“, „TELEPHONE_NUMBER“, „PHONE_NUMBER“.

Komentáře sloupce Komentáře jsou prohledávány na částečnou shodu s frázemi „telefon“, „mobil“, „telefonní“, „mobilní“, „msisdn“ v angličtině „telephone“, „phone“, „cellphone“.

3.2.8 IMEI

IMEI[14] je unikátní údaj, který identifikuje mobilní telefon svého vlastníka. V případě odcizení mobilního telefonu lze pomocí IMEI čísla blokovat mobilní síť. Jedná se o patnáctimístné číslo, které se skládá z těchto částí[26]:

- **TAC** - Type Approval Code (6 číslic z toho první dvě jsou kód země)
- **FAC** - Final Assembly Code (2 číslice - kód výrobce)
- **SNR** - Serial Number (6 číslic - sériové číslo telefonu)
- **SP** - Spare (1 číslice - kontrolní součet)

Poslední kontrolní číslice se vypočítá Luhnovým algoritmem. Kontrolní číslice x pro IMEI $n_{15}n_{14}n_{13}n_{12}n_{11}n_{10}n_9n_8n_7n_6n_5n_4n_3n_2x$ se vypočítá následovně (CS ve výpočtu značí ciferný součet)[27]:

$$x = (-\sum_{n=1}^{7} CS(2n_{2i}) - \sum_{n=1}^{7}(n_{2i+1}))mod10$$

IMEI lze snadno identifikovat pomocí validační funkce, která zkontroluje délku a kontrolní číslici podle algoritmu uvedeného výše.

Názvy sloupce Očekává se pouze jediný název sloupce v českém i anglickém jazyce „IMEI“.

Komentáře sloupce Komentáře jsou prohledávány na částečnou shodu s frází „imei“.

[14] International Mobile Equipment Identity

3.2.9 Číslo bankovního účtu

V některých případech je možné identifikovat osobu pomocí čísla bankovního účtu, proto se dá číslo účtu považovat za osobní údaj. Číslo účtu nebo také BBAN[15] zaručuje jednoznačnou identifikaci účtu v systému bankovního platebního styku. Číslo účtu se na základě vyhlášky č. 169/2011 Sb. používá buď v národním formátu nebo ve formátu IBAN.

Číslo účtu v národním formátu se skládá z identifikátoru účtu klienta a kódu banky. Identifikátor účtu je tvořen maximálně 16 číselnými znaky a dělí se na základní číslo účtu a předčíslí. Základní číslo účtu obsahuje maximálně 10 číslic, předčíslí obsahuje maximálně 6 číslic. Obvykle je číslo zleva doplněno „nevýznamovými nulami“ do délky maximálně 10 znaků, pokud je v daném čísle účtu použito předčíslí a zároveň délka základního čísla účtu není 10.[28]

Každé číslo v národním formátu lze zkontrolovat pomocí kontrolního součtu. Kontrolní součet čísla účtu ve formátu $n_{10}n_9n_8n_7n_6n_5n_4n_3n_2n_1$ musí být dělitelný 11 beze zbytku a vypočítá se následovně:

$$S = \sum_{n=1}^{10}((2^i mod 11) * n_i)$$

BBAN lze snadno identifikovat pomocí validační funkce, která zkontroluje, zda je kontrolní součet dělitelný 11 beze zbytku a zkontroluje kód banky například podle číselníku kódů, který lze najít na stránkách ČNB[16].[29]

Kromě BBAN existuje také IBAN[17]. IBAN je formát čísla účtu definovaný mezinárodní normou ISO 13616. Standard stanovuje mezinárodní čísla účtu tak, že může obsahovat pouze číslice a velká písmena. Na začátku jsou 2 znaky, které představují kód země, následují 2 znaky, které slouží jako kontrolní číslice. Dále obsahuje kód banky a číslo účtu (maximálně 30 číslic).

Standard IBAN definuje dvě kontrolní číslice, které se vypočítají systémem ISO 7064. Tento algoritmus pracuje následovně: Z IBAN se přesunou první 4 znaky na konec. Dále se všechna písmena nahradí čísly (A = 10, B = 11, …, Z = 35) a vypočítá se zbytek po dělení výsledného čísla číslem 97. Pokud je dané IBAN platné, musí být tento zbytek roven číslu 1. Pro výpočet kontrolních číslic stačí jako kontrolní číslice dosadit nuly, spočítat zbytek po dělení a ten odečíst od 98.[30]

Identifikaci IBAN bude ideální provést pomocí validační funkce. Tato funkce zkontroluje délku a ověří údaj podle algoritmu popsaného výše.

Názvy sloupce Očekávané názvy sloupce budou v českém jazyce jsou „UCET“, „CISLO_UCTU“, „BANKOVNI_UCET“ „BBAN“, „IBAN“ v angličtině pak „BANK“, „BANK_ACCOUNT“.

[15] Basic Bank Account Number

[16] Česká národní banka

[17] International Bank Account Number

Komentáře sloupce Komentáře jsou prohledávány na částečnou shodu s frázemi „účet“, „bankovní“, „číslo účtu“, „bban“, „iban“ v angličtině „bank“ a „bank account“.

3.2.10 Biometrické údaje

Biometrické údaje jsou osobní údaje technického charakteru zpracování fyzických či fyziologických znaků fyzické osoby, které umožňují její jednoznačnou identifikaci. Mezi biometrické údaje patří například otisk prstu, snímek obličeje, snímek oční sítnice a duhovky a podpis.[31]

Metoda identifikace otisku prstu a snímku obličeje je podrobně vysvětlena v práci Davida Skalského[4]. V této práci se zaměříme na zpracování snímku oční duhovky.

3.2.10.1 Snímek oční duhovky

Identifikace podle oční duhovky je založena na snímání lidské duhovky. Neexistuje jiná biometrická charakteristika člověka, která by poskytovala více rozlišovacích možností než právě oční duhovka.

Při snímání se používají konvenční CCD[18] kamery a nedochází k žádnému intimnímu kontaktu uživatele se snímacím zařízením. Po nasnímání oka je detekována oblast duhovky, která je pokryta sítí křivek. Na základě jasu bodů kolem těchto křivek je vytvořen kód (iriscode o velikosti od 256 B do 1024 B). Při tomto procesu je nutné normalizovat jas a velikost duhovky pro následné porovnání. K porovnání se využívá výpočet Hammingovy vzdálenosti[19] mezi dvěma kódy.[32]

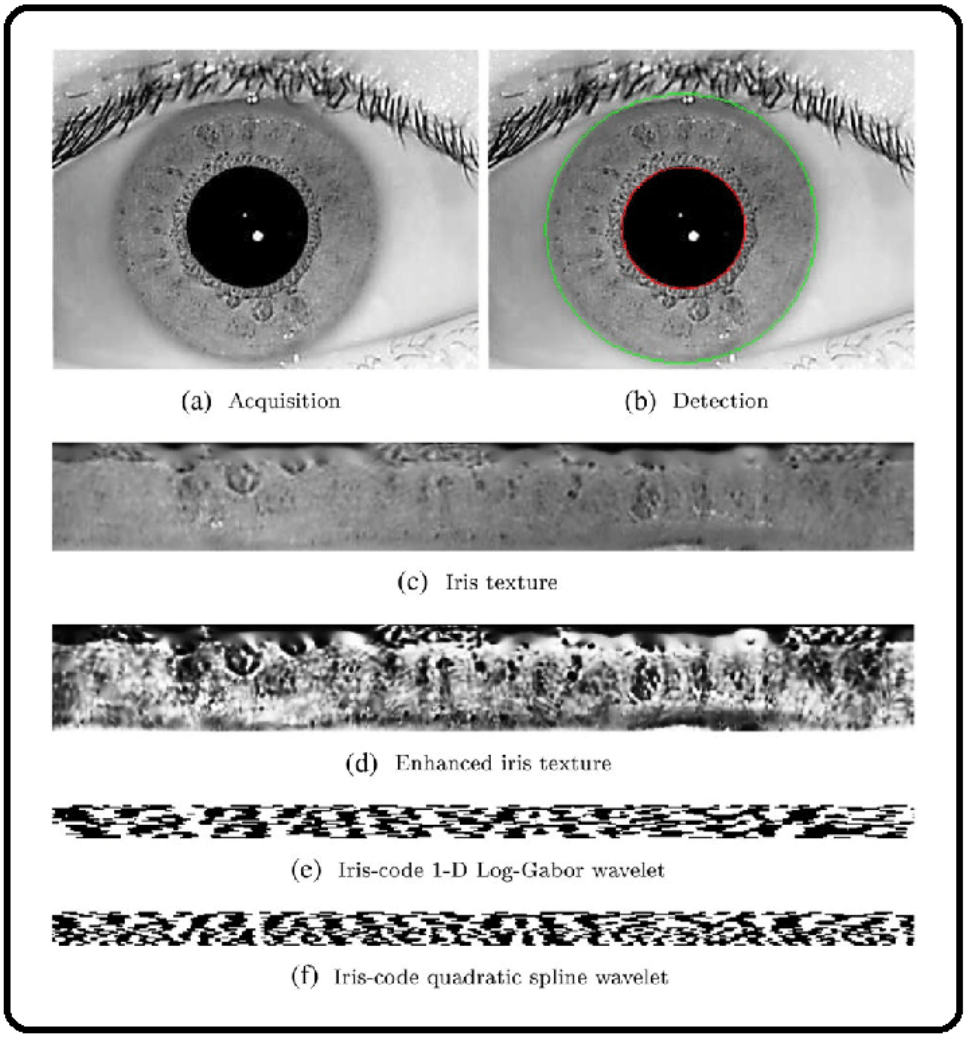
Na obrázku 3.2[1] je znázorněn proces rozpoznání duhovky a převedení do iriscodu. Nejprve jsou ostré snímky podrobeny analýze pro zjištění přítomnosti duhovky a její lokalizaci, což lze provést například pomocí matematických detektorů kruhových a obloukových hranic. Následně je detekovaná duhovka zarovnána pomocí Daugmanova modelu a pomocí Daugmanova algoritmu je poté převedena do iriscodu. Vytvořený kód je následně možné uložit přímo do databáze nebo ho využít k porovnání s jinými záznamy obsahující zpracované oční duhovky.[33]

Po zpracování snímku oční duhovky je několik možností, jakými lze informace o oční duhovce uložit. Jednou z možností je uložit přímo fotografii nasnímané oblasti duhovky nebo obrázek obsahující její iriscode. V takovém případě je téměř nemožné snímek identifikovat. Jiná možnost nabízí uložit iriscode duhovky v kódové formě. Obvykle se využívá iriscode, který obsahuje 2048 bitů (256 B), tedy 2048 symbolů „0“ nebo „1“. V některých případech jsou bity ukládány po dvojicích a jsou odděleny mezerou.[34]

[18] Charge-coupled device

[19] Počet bitů, v nichž se liší dvě sousední platné kódové kombinace

Obrázek 3.2: Proces rozpoznání duhovky[1]



Pro identifikaci iriscodu je tedy možné využít například následující regulární výraz:

- ^([01]{2048})|((00\\s|01\\s|10\\s|11\\s){1023}(00|01|10|11))\$

Tento regulární výraz sice rozpozná platný iriscode, je však třeba brát ohled na to, že může vyhodnotit jako platný iriscode kterýkoliv jiný binární kód o velikosti 2048 bitů, který v sobě nenese žádnou informaci o oční duhovce.

Názvy sloupce Očekávané názvy sloupce budou v českém jazyce jsou „DUHOVKA“, „SNIMEK_DUHOVKY“, „SKEN_DUHOVKY“ v angličtině pak „IRIS“, „IRIS_SCAN“.

Komentáře sloupce Komentáře jsou prohledávány na částečnou shodu s frázemi „duhovka“, „sken duhovky“, „snímek duhovky“ v angličtině „iris“, „iris scan“.

3.3 Vazba mezi údaji

Některé osobní údaje mohou mít vazbu k jiným osobním údajům a nabízí se tak možnost využít této vazby k jejich snazší identifikaci. Lze předpokládat, že dvojice osobních údajů, mezi kterými existuje vazba, se v databázích mohou vyskytovat na stejném řádku nebo se mohou vyskytovat ve stejném dokumentu.

3.3.1 Datum narození

Jedním z takových údajů je datum narození, které má vazbu k rodnému číslu. Popis rodného čísla a metody identifikace jsou podrobně popsány v bakalářské práci Davida Skalského[4].

Po nalezení rodného čísla v databázi je možné zkontrolovat, zda se na stejném řádku vyskytuje datum narození odpovídající nalezenému rodnému číslu a případně identifikovat sloupec, ve kterém se pravděpodobně nachází datum narození. Obdobný princip lze využít u nestrukturovaných dat. Pokud se v souboru vyskytuje rodné číslo, vyplatí se zkontrolovat, zda se v něm nevyskytuje také odpovídající datum narození.

3.3.2 DIČ

Daňové identifikační číslo je jednoznačná identifikace daňového subjektu. Obsahuje kód země a identifikátor. U fyzické osoby je identifikátorem obvykle jeho rodné číslo, u právnické osoby je to většinou jeho IČO[20].[35] IČO je podrobně popsáno v bakalářské práci Davida Skalského[4].

U DIČ lze použít stejný postup jako u data narození. Při detekci rodného čísla nebo identifikačního čísla osoby je možné zkontrolovat, zda se nenachází odpovídající DIČ na stejném řádku v databázi nebo v rámci stejného dokumentu.

3.4 Shrnutí

V této sekci se nachází tabulka, ve které jsou uvedeny všechny analyzované osobní údaje, jejich způsob identifikace a zda jsou tyto údaje v rámci této práce implementovány.

[20] Identifikační číslo osoby

Tabulka 3.6: Shrnutí analýzy osobních údajů

Údaj	**Identifikace**	**Implementováno**
GPS souřadnice	regulární výraz	ANO
VIN	regulární výraz	ANO
SPZ	validační funkce	ANO
Zdravotní stav - léky	porovnání se slovníkem	ANO
Zdravotní stav - diagnózy	porovnání se slovníkem	ANO
Sexuální orientace	porovnání se slovníkem	ANO
Číslo cestovního dokladu	regulární výraz	ANO
Telefonní číslo (MSISDN)	regulární výraz	ANO
IMEI	validační funkce	ANO
BBAN	validační funkce	ANO
IBAN	validační funkce	ANO
Snímek oční duhovky	regulární výraz	ANO
Datum narození	ověření vazbou	ANO
DIČ	ověření vazbou	ANO

Kapitola 4

Návrh řešení

Tato kapitola obsahuje popis řešení, které je použité ve výsledném programu. Jsou zde také popsány jednotlivé funkcionality, které jsou součástí řešení.

4.1 Požadavky

Nejprve je třeba stanovit, jakými funkcionalitami má program disponovat. Winch v současné době dokáže vyhledávat osobní údaje ve strukturovaných datech uložených v databázích. Pokud jsou v databázi uložena nestrukturovaná data, program je ignoruje a údaje v nich nevyhledává. Vyhledávání osobních údajů v nestrukturovaných datech uložených ve filesystému zatím nepodporuje.

Je tedy třeba rozšířit existující program o nové osobní údaje uvedené v sekci 3.2 a tyto údaje následně vyhledávat v databázích a nestrukturovaných datech. Řešení musí podporovat identifikaci údajů pomocí výskytu ve slovníku, regulárního výrazu, validační funkce a také pomocí vazby s jinými údaji.

U nestrukturovaných dat se nabízí více metod, které lze použít k vyhledávání osobních údajů. Je třeba rozhodnout, které metody jsou efektivnější, případně ve kterých situacích se hodí jednotlivé metody (více v sekci 4.3.1).

4.2 Vyhledávání v databázích

Nástroj Winch umožňuje vyhledávat osobní údaje v databázích Microsoft SQL, Oracle SQL a PostgreSQL. Způsoby extrakce dat z jednotlivých databází jsou detailně popsány v bakalářské práci Davida Skalského[4]. Data lze validovat pomocí slovníků nebo validační funkce. Identifikace osobních údajů pomocí regulárního výrazu je možná pouze v databázích OracleSQL a PostgreSQL. Nové řešení by mělo umožnit validovat osobní údaje pomocí regulárního výrazu i v databázích Microsoft SQL. Vedle toho je třeba, aby

program podporoval vyhledávání osobních údajů, které mají vazbu k jiným osobním údajům. Poslední funkcionalita rozšíření vyhledávání v databázích se týká nestrukturovaných dat, které se mohou vyskytovat v databázi například v podobě binárních dat. Program by měl tato data rozpoznat a zjistit, zda obsahují nějaké osobní údaje.

4.2.1 Rozšíření regulárních výrazů

Jak již bylo řečeno, stávající program není schopen v databázích Microsoft SQL ověřit údaje pomocí regulárních výrazů. Problémem je, že řešení, které bylo v minulosti implementováno, podporuje pouze jeden regulární výraz pro každý osobní údaj, což je vzhledem k odlišnosti syntaxe regulárních výrazů v databázích Microsoft SQL a Oracle SQL nevyhovující. Je tedy potřeba přidat do stávajícího řešení více regulárních výrazů, které mají odlišnou syntaxi v jednotlivých databázích.

4.2.2 Vazby mezi údaji

Další funkcionalita, kterou musí program mít, je vyhledávání osobních údajů na základě vazby k jiným již nalezeným údajům. Ideálním řešením je zkontrolovat celý řádek v databázi, zda neobsahuje údaje, které lze ověřit pomocí právě nalezeného osobního údaje. Například pokud program najde na daném řádku databáze rodné číslo, potom projde celý řádek a zkontroluje, zda se na něm nevyskytuje i datum narození nebo daňové identifikační číslo.

4.2.3 Binární data v databázích

U databází se velmi často stává, že kromě klasických hodnot, jako je string, integer a jiné, mohou obsahovat binární data uložená jako BLOB, CLOB, NCLOB, BFILE nebo v kterékoliv jiné podobě. Typickým příkladem jsou smlouvy nebo jiné dokumenty. Je třeba, aby si program taková data dokázal zpracovat a zkontroloval, zda se v nich nenacházejí osobní údaje. V případě nalezení binárních dat v databázi jsou dvě možnosti, jak dále postupovat.

První možnost je založena na předpokladu, že pokud program najde osobní údaje na stejném řádku, na jakém jsou uložena nestrukturovaná data, pak je pravděpodobné, že se tyto nalezené hodnoty budou vyskytovat i v těchto datech. Program se pokusí tyto hodnoty nalézt. Tato varianta by nemusela být příliš náročná, poněvadž se netestuje celý obsah dat, ale hledají se pouze konkrétní hodnoty, kterých většinou není mnoho. Je ovšem nutné počítat s tím, že v datech mohou být uloženy i další osobní údaje, jejichž vyhledávání tato metoda nepodporuje. Například program najde na stejném řádku databáze jméno, příjmení, rodné číslo, adresu bydliště a také nějaký dokument, který bude představovat smlouvu. Je velká šance, že nalezené údaje se budou vyskytovat i v této smlouvě, a proto program zkontroluje, zda se nevyskytují i v ní.

Může se však stát, že smlouva bude obsahovat i jiné osobní údaje, například telefonní číslo, které však touto metodou nebude vyhledáno.

Druhou možností je prohledat celá data a vyhledat všechny osobní údaje, které se v nich nacházejí. Tato metoda je sice spolehlivější, ale časově může být velmi náročná.

Ideálním řešením je vytvořit program, ve které si uživatel sám dle svých potřeb zvolí, které z těchto možností se budou provádět a které ne.

4.3 Vyhledávání v nestrukturovaných datech

Jednou z hlavních funkcionalit, kterou musí program zvládat, je vyhledávání osobních údajů v nestrukturovaných datech. Nejdříve je potřeba zjistit, o jaký typ dat se jedná, na jakém úložišti jsou uložena, a následně provést extrakci jejich obsahu, ve kterém se budou osobní údaje vyhledávat.

4.3.1 Metody vyhledávání

V této sekci jsou popsány metody, kterými lze osobní údaje v obsahu nestrukturovaných dat vyhledávat. K ověření údajů využíváme regulární výraz, validační funkci nebo slovník. Pro každý z těchto způsobů ověření se nabízí více metod vyhledávání a není zcela jasné, které z nich jsou nejefektivnější. Touto tematikou se zabývá kapitola 6.

Metody ověření regulárním výrazem:

Metoda A: Program rozdělí obsah dat na jednotlivá slova (případně více slov) a každý prvek zkontroluje pomocí regulárního výrazu, zda odpovídá formátu daného údaje.

Metoda B: Program vyhledá údaje přímo pomocí regulárního výrazu.

Metody ověření validační funkcí:

Metoda A: Program rozdělí obsah dat na jednotlivá slova (případně více slov) a každý prvek zkontroluje pomocí validační funkce, zda odpovídá formátu daného údaje.

Metoda B: Program vyhledá potenciální kandidáty pomocí přibližného regulárního výrazu a tito kandidáti jsou poté zkontrolováni validační funkcí, zda odpovídají formátu daného údaje.

Metody ověření pomocí slovníku:

Metoda A: Program rozdělí obsah dat na jednotlivá slova a u každého prvku zkontroluje, zda se vyskytuje ve slovníku.

Metoda B: Program se pokusí vyhledat v obsahu dat jednotlivé hodnoty ze slovníku.

Metoda C: Program vyhledá potenciální kandidáty pomocí přibližného regulárního výrazu a následně zkontroluje, zda se nevyskytují ve slovníku.

Metoda D: Pokud se jedná o víceslovný údaj, vyhledá program pomocí přibližného regulárního výrazu potenciální kandidáty, které představují první slovo z údaje. Následně je každý záznam ve slovníku porovnán na částečnou shodu s každým kandidátem. V případě, že se potvrdí, že záznam ze slovníku obsahuje daného kandidáta, pokusí se program vyhledat záznam ze slovníku v obsahu dat.

V jednotlivých metodách se velmi často objevuje hledání tzv. potenciálních kandidátů pomocí přibližného regulárního výrazu. Účelem těchto metod je přeskakovat slova, u kterých je jisté, že neobsahují daný osobní údaj. Například pokud vyhledáváme křestní jména, můžeme využít pravidel českého pravopisu a pomocí regulárního výrazu najít pouze slova, která mají na začátku velké písmeno. Ve smyslu této práce jsou za potenciální kandidáty považovány právě tyto nalezené výrazy, kterých může být výrazně menší počet než počet všech slov v datech.

Metoda s potenciálními kandidáty vykazuje menší spolehlivost u osobních údajů ověřovaných pomocí slovníků, které mohou obsahovat více slov (metoda D u ověření pomocí slovníků). Typickým příkladem může být vyhledávání měst obsahujících až 5 slov (např. Rychnov u Jablonce nad Nisou). Potenciálních kandidátů by v tomto případě bylo příliš mnoho a efektivita vyhledávání by byla nízká. Nabízí se tak varianta vyhledat pomocí přibližného regulárního výrazu slova, která by mohla představovat první slovo v názvu města, tedy slova s velkým písmenem na začátku. Následně by se program pokusil najít tato slova v názvech měst ve slovníku. Pokud by některé město tuto část obsahovalo, ověřil by, zda se celý název vyskytuje v obsahu daných dat. Je potřeba počítat se skutečností, že s narůstajícím objemem dat může být tento proces velmi náročný a zdlouhavý. Pak je třeba zvážit, zda není výhodnější využít jiné metody.

Obecně je potřeba otestovat všechny tyto metody na optimálním vzorku dat a vyhodnotit, které řešení je nejefektivnější. Tomuto tématu se věnuje kapitola 6.

4.3.2 Způsoby uložení dat

Je mnoho způsobů, jakými lze nestrukturovaná data ukládat. Jedním ze způsobů je uložení přímo v databázi v podobě binárních dat. Dále mohou být uloženy ve filesystému v klasické podobě souborů nebo na nějakém vzdáleném datovém úložišti. Je potřeba, aby implementované řešení pro samotné

vyhledávání osobních údajů nebylo závislé na způsobu uložení dat a bylo tak snadno rozšiřitelné o nové typy úložišť.

Mezi nestrukturovaná data patří mimo jiné i obrázky, které také mohou obsahovat osobní údaje. Bohužel ve většině případů z nich nejsme schopni získat data, která potřebujeme. Jsou však obrázky, ze kterých lze velmi snadno získat obsah. Jedná se o QR kódy a různé čárové kódy.

4.3.3 QR a čárové kódy

QR a čárové kódy se používají především k rychlému, efektivnímu přenosu libovolné informace a jsou v dnešní době hojně využívané. V těchto kódech se může vyskytovat mnoho osobních údajů, velmi často v nich bývá uloženo telefonní číslo, email, ale i jiné údaje.

V následujících kódech jsou například uloženy osobní údaje týkající se autora této práce (QR kód byl vygenerován pomocí QRCodeMonkey[36], čárový kód pomocí TEC-IT[37]):

Obrázek 4.1: Jméno, telefonní číslo a město uložené v čárovém kódu

Obrázek 4.2: Jméno, telefonní číslo a město uložené v QR kódu

Součástí řešení by mělo být vyhledávání osobních údajů v QR kódech a čárových kódech. Program musí u dat rozpoznat, zda se jedná o obrázek, a poté zda obrázek obsahuje některý z uvedených kódů. Pokud ano, dekóduje tento kód a získá jeho obsah, ve kterém ověří, zda se v něm nacházejí některé osobní údaje. K vyhledávání údajů jsou použity metody ze sekce 4.3.1.

4.4 Návrh konfigurovatelnosti procesu

Nedílnou součástí řešení musí být možnost konfigurace procesu z pohledu uživatele. Uživatel by měl mít možnost nastavit parametry týkající se procesu vyhledávání osobních údajů ve strukturovaných a nestrukturovaných datech. Jedná se především o typy údajů, které se mají vyhledávat, metody vyhledávání u nestrukturovaných dat uložených v databázích, popřípadě typy dat a souborů, které se mají prohledávat.

Toto nastavení může v praxi velmi urychlit celý proces. Uživatel může eliminovat osobní údaje, u kterých si je jistý, že se v prohledávaných datech nenacházejí, nebo může volit méně náročné metody vyhledávání apod.

4.5 Návrh architektury

Nástroj Winch je třeba rozšířit v oblasti vyhledání osobních údajů ve strukturovaných a nestrukturovaných datech. Ve vztahu k vyhledávání ve strukturovaných datech, není třeba vytvářet nový návrh architektury, zde přidáváme jen některé funkcionality. Kompletní návrh však musíme udělat u vyhledávání v nestrukturovaných datech.

Cílem návrhu architektury by mělo být zajištění srozumitelnosti, rozšiřitelnosti a udržovatelnosti programu. Je tedy třeba zvážit, které části programu se mohou v budoucnu rozšiřovat. Jedná se především o podporované osobní údaje, metody vyhledávání a typy úložišť, ve kterých údaje vyhledáváme.

V první části je třeba provést extrakci obsahu nestrukturovaných dat. Pro extrakci jsou v návrhu vytvořena dvě rozhraní FileExtractor a DataExtractor. FileExtractor se stará o extrakci dat z jednotlivých úložišť. Toto rozhraní implementují třídy DatabaseExtractor pro extrakci dat získaných z databáze a FileSystemExtractor pro extrakci dat z filesystému. Rozhraní obsahuje metodu getFile, která vytvoří z dat soubor. Ten je pak následně předán rozhraní DataExtractor.

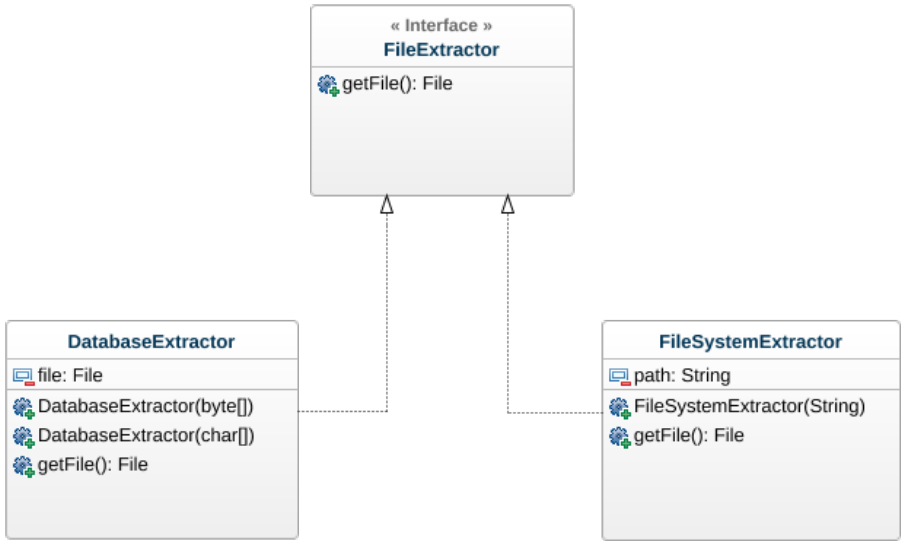
Toto rozhraní implementují třídy, které se starají o extrakci dat z jednotlivých typů souborů. Tento návrh je snadno rozšiřitelný o nové typy úložišť a uložených souborů.

Rozhraní DataExtractor je následně předáno třídě DataDiscoverer, která z něj díky metodě getContent může získat data bez ohledu na to, o který konkrétní typ extraktoru se jedná. Tato třída se stará o celý proces vyhledávání osobních údajů v nestrukturovaných datech. Kromě extraktoru má

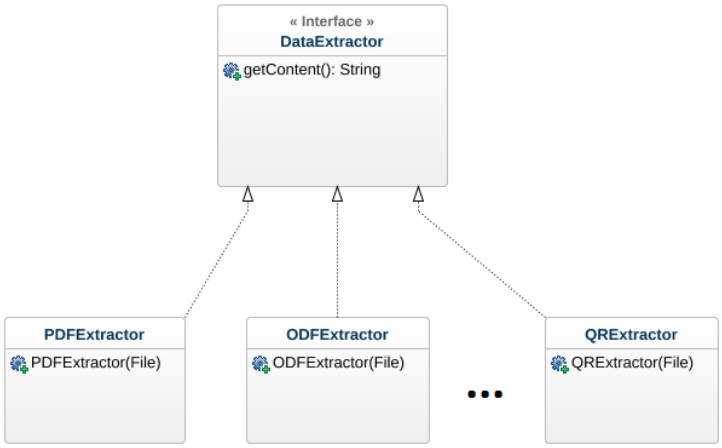
další dva vstupní parametry, kterými jsou parametry zadané uživatelem a případně osobní údaje nalezené v databázi. Uživatelské parametry určují, mimo jiné, které validátory třída DataDiscoverer využije. Validátory slouží k vyhledávání konkrétních osobních údajů. V návrhu je pro ně vytvořeno rozhraní s názvem ContentValidator, které obsahuje metodu getResults, která hledá a vrací nalezené osobní údaje, a metodu getContent, která vrací obsah, v němž jsou osobní údaje vyhledávány. Toto rozhraní implementují třídy AbstractValidator a ValueValidator podle toho, o jakou metodu vyhledávání se jedná. Třída ValueValidator slouží k práci s údaji, které již byly nalezeny v databázi a u kterých je pouze ověřován výskyt v souboru. Od třídy AbstractValidator dědí třídy RegularValidator, FunctionValidator, DictionaryValidator, LinkageValidator pro jednotlivé způsoby ověření údajů (pomocí slovníků, validačních funkcí apod.) a ty jsou dále rozšířeny třídami pro validaci konkrétních osobních údajů. Kromě zmíněných tříd se v modelu vyskytují třídy DictionaryController, ta slouží pro práci se slovníky, a Result, která reprezentuje nalezený osobní údaj v nestrukturovaných datech. Třída DataDiscoverer obsahuje metodu getResults, která vrací všechny osobní údaje nalezené v datech. V budoucnu lze řešení snadno rozšířit o nové typy osobních údajů a případně i o nové metody jejich vyhledávání.

Diagramy byly vytvořeny pomocí GenMyModel[38] a pro jejich vytvoření byl použit návrhový vzor Strategie. Tento vzor slouží k vyměňování různých implementací algoritmu za běhu programu. Tato záměna může proběhnout buď explicitně, nebo implicitně.

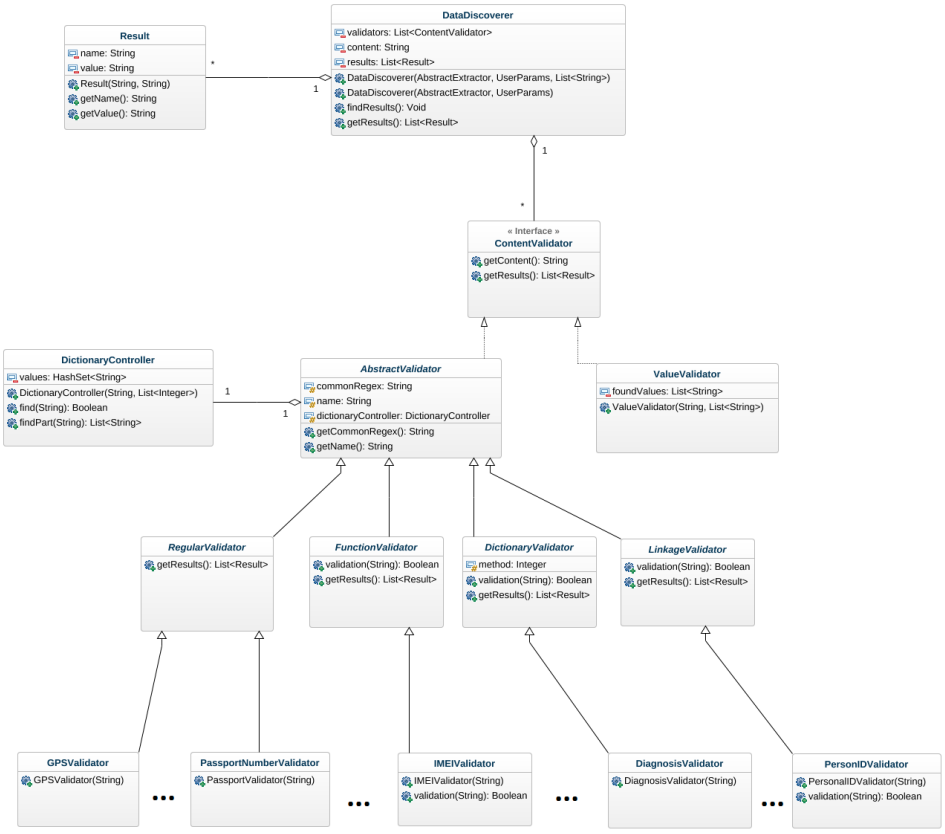
Obrázek 4.3: UML Class diagram - Extrakce dat 1. část



Obrázek 4.4: UML Class diagram - Extrakce dat 2. část



Obrázek 4.5: UML Class diagram - Validace dat



Kapitola 5

Implementace

Tato kapitola obsahuje popis samotné implementace, která rozšiřuje současný nástroj Winch. Je zde rozebráno vyhledávání osobních údajů v databázích a také v nestrukturovaných datech. Dále kapitola popisuje možnosti uživatele konfigurovat celý proces vyhledávání osobních údajů v nestrukturovaných datech. Celá implementace nástroje Winch je v jazyce Groovy.

5.1 Vyhledávání v databázích

Jednou z funkcionalit, kterou současný nástroj Winch nabízí, je vyhledávání osobních údajů v tabulkách databáze. Předchozí implementace podporovala několik základních osobních údajů a bylo potřeba rozšířit tuto implementaci o nové osobní údaje, které jsou rozebrány v sekci 3.2.

V nástroji Winch se o vyhledávání osobních údajů starají tzv. vyhledávače anonymizačních tříd. Základ těchto tříd tvoří abstraktní třída `AnonymizationClassDiscovery`, od které dědí všechny anonymizační třídy reprezentující jednotlivé typy osobních údajů, které se v databázi vyhledávají. Návrh těchto tříd nebyl proveden autorem této práce.

V rámci rozšíření této části bylo potřeba implementovat anonymizační třídy pro nové osobní údaje. Každá tato třída rozšiřuje abstraktní třídu `AnonymizationClassDiscovery`, která nabízí základní funkcionality jako například porovnání jména sloupců s očekávanými názvy osobních údajů, vyhledávání očekávaných klíčových slov v komentářích sloupců, kontrolu sloupců, zda obsahují relevantní datový typ pro daný osobní údaj a výpočet pravděpodobnosti, že se ve sloupci vyskytuje hledaný osobní údaj.

Do každé přidané anonymizační třídy bylo potřeba uvést název daného osobního údaje, očekávané názvy sloupců, očekávaná klíčová slova v komentářích sloupců, datové typy, pomocí kterých může být údaj uložen, minimální délku údaje a metodu `discover`, která provádí veškeré dotazy nad jednotlivými záznamy v databázi a také počítá výslednou pravděpodobnost výskytu

osobního údaje. U údajů ověřovaných validační funkcí bylo potřeba tuto funkci implementovat.

V případě osobních údajů, které se ověřují pomocí regulárního výrazu, bylo potřeba doplnit příslušný regulární výraz, který k validaci využívají podporované databáze. Ve všech databázích není možné použít všechny regulární výrazy uvedené v sekci 3.2. Použití těchto regulárních výrazů je v databázích Oracle SQL a Postgre SQL možné, avšak v Microsoft SQL tyto výrazy použít nelze, poněvadž tato databáze regulární výrazy vůbec nepodporuje. Nabízí pouze operátor `LIKE`, který umožňuje vyhledat řetězec textu na základě určitého paternu, avšak tento patern má oproti regulárním výrazům velmi omezené způsoby zápisu. Některé regulární výrazy u osobních údajů bylo možné upravit do tvaru, který je pro operátor `LIKE` přijatelný, jedná se konkrétně o číslo pasu a VIN. GPS souřadnice, telefonní číslo a kód oční duhovky mají příliš složitý regulární výraz, který takto převést nelze. Tyto údaje tedy v databázi Microsoft SQL není možné vyhledat.

V případě údajů, které se ověřují slovníky, obsahoval současný nástroj Winch abstraktní třídu `AbstractDictionary`, která se stará o parsování slovníků uložených v souborech v `.csv` formátu a také nabízí základní funkce pro vyhledávání konkrétních hodnot ve slovníku. Tuto třídu potom rozšiřují třídy pro jednotlivé slovníky a ty definují název slovníku, název souboru, ve kterém se slovník nachází, a seznam sloupců, ze kterých se mají hodnoty slovníku použít. V rámci rozšíření bylo potřeba vytvořit slovníky pro osobní údaje, které se pomocí nich ověřují, a také implementovat jednotlivé třídy rozšiřující třídu AbstractDictionary.

Konkrétně byly implementovány anonymizační třídy `GPSDiscoverer`, `VINDiscoverer`, `RegistrationPlateDiscoverer`, `MedicamentDiscoverer`, `DiagnosisDiscoverer`, `SexualOrientationDiscoverer`, `PassportNumberDiscoverer`, `PhoneNumberDiscoverer`, `IMEIDiscoverer`, `BBANDiscoverer`, `IBANDiscoverer`, `IrisDiscoverer` a dále třídy pro jednotlivé slovníky `MedicamentsDictionary`, `DiagnosisDictionary` `SexualOrientationDictionary`.

5.2 Vyhledávání v nestrukturovaných datech

Jednou z hlavních částí řešení této práce je vyhledávání osobních údajů v nestrukturovaných datech, jako jsou například různé dokumenty, smlouvy apod. Ty mohou být uloženy přímo v databázích v podobě binárních dat nebo mohou být uloženy ve filesystému nebo v kterékoliv jiné formě. Implementované řešení podporuje extrakci dat uložených v databázích a ve filesystému, ale lze jej snadno rozšířit o nové typy úložišť, neboť samotné vyhledávání osobních údajů je implementováno zvlášť a je nezávislé na způsobu uložení dat.

5.2.1 Binární data v databázích

Nejprve je třeba vymezit, jakým způsobem mohou být binární data uložená v jednotlivých databázích. V současné době podporuje Winch tři databázové systémy, kterými jsou Microsoft SQL Server, Oracle DB a Postgre SQL.

V **Microsoft SQL** databázi mohou být binární data uložena v následujících formátech[39]:

- `binary(`n`)` - datový typ s pevnou délkou n bytů (rozsah 1 až 8000) pro ukládání binárních dat (souborů, obrázků atp.)
- `varbinary(`n`)` - datový typ s variabilní délkou, kde n udává maximální kapacitu v bytech (rozsah 1 až 8000) - lze použít klíčové slovo MAX, které zaručí nastavení maximálního počtu bytů na $2^{31} - 1$
- `image` - v současné době málo využívaný datový typ, místo tohoto datového typu se doporučuje používat `varbinary`

V **Oracle DB** mohou být binární data uložena v následujících formátech[40]:

- `RAW` - řetězec binárních znaků s proměnlivou délkou, nejvýše 2000 bytů
- `LONG RAW` - řetězec binárních znaků s proměnlivou délkou, nejvýše 2 GB
- `BLOB` - nestrukturovaná binární data do velikosti 4 GB
- `CLOB` - nestrukturovaná znaková data do velikosti 4 GB
- `NCLOB` - sloupec typu CLOB podporující vícebytovou množinu znaků
- `BFILE` - binární soubor uložený vně databáze v souborovém systému do velikosti 4 GB

V **Postgre SQL** mohou být binární data uložena v následujících formátech[41]:

- `bytea` - tento typ drží data přímo v databázi, pojme až 1 GB
- `Large Object` - tento typ je velmi podobný datovému typu BLOB používaném v Oracle DB, maximální povolená velikost je 2 GB

5.2.2 Extrakce dat

Téměř všechna data, která jsou uložena pomocí těchto datových typů, je v Groovy možné získat pomocí JDBC driveru (s výjimkou `Large object` v Postgre SQL), který pro každý typ nabízí metody jako například `getBytes` nebo `getBLOB`. Tyto metody vrací pole bajtů, v případě datových typů `CLOB` a `NCLOB` pole charů.

Pro tato data je v řešení implementováno rozhraní `FileExtractor` obsahující metodu `getFile`, která vrací soubor vytvořený z dat. Toto rozhraní implementují třídy `DatabaseExtractor` a `FileSystemExtractor`. `DatabaseExtractor` se stará o data získaná z databází. Konstruktor této třídy má jeden vstupní parametr, kterým může být pole bajtů nebo pole charů. Tato pole jsou následně zpracována a převedena do dočasného souboru. Třída `FileSystemExtractor` se stará o data uložená ve filesystému, konstruktor třídy má jeden vstupní parametr, kterým je cesta k danému souboru.

Soubor, který tyto třídy vytvoří, je následně předán třídám implementující rozhraní `DataExtractor`, které obsahuje metodu `getContent`, a ta získá a vrátí textový obsah daného souboru. Výběr konkrétní třídy, které se má soubor předat, určuje nástroj Apache Tika, který umí rozpoznat, o jaký formát souboru se jedná. Implementace podporuje extrakci obsahu souborů Microsoft Office, dále také HTML, ODF, PDF, TXT, XML (vysvětlení jednotlivých zkratek obsahuje část Seznam použitých zkratek) a také extrakci obsahu QR kódů a čárových kódů. Pro každý tento typ souboru je vytvořena vlastní třída. Tato třída je následně předána jako parametr konstruktoru třídy `DataDiscoverer`, která si z ní pomocí metody `getContent` získá obsah souboru.

5.2.3 Vyhledávání a validace osobních údajů

Třída `DataDiscoverer` obstarává celý proces vyhledávání osobních údajů v daném obsahu. Má k dispozici metody `findResults` a `getResults`, které vyhledávají a vrací nalezené osobní údaje. Dále obsahuje třída atributy, kterými je obsah souboru, seznam nalezených osobních údajů a seznam validačních tříd pro jednotlivé údaje, které se mají vyhledávat.

Validační třídy vyhledávají a ověřují v daném obsahu konkrétní osobní údaje. Je pro ně vytvořeno rozhraní `ContentValidator`, které obsahuje metodu `getResults` pro vyhledání údajů. Toto rozhraní implementují třídy `AbstractValidator` a `ValueValidator`. Třída `ValueValidator` má na starosti vyhledávání těch údajů, které již byly nalezeny v databázi, a tyto nalezené údaje jsou třídě předány jako parametr konstruktoru.

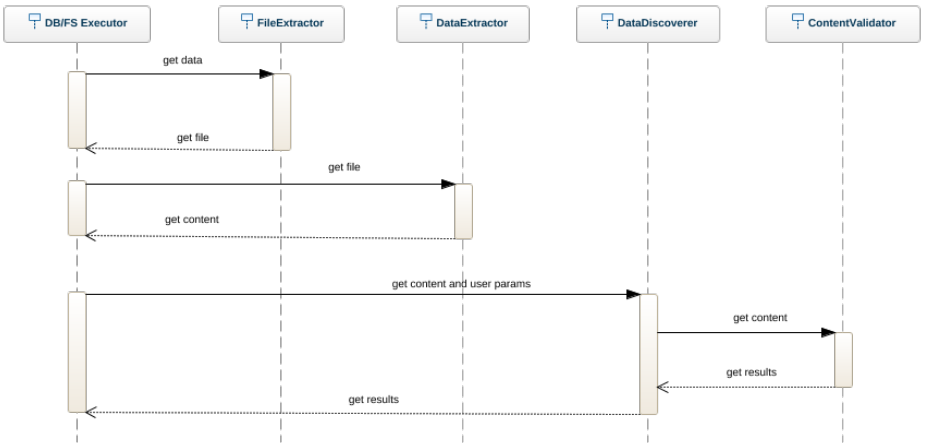
Třída `AbstractValidator` je abstraktní, jak už napovídá její název, a rozšiřují ji abstraktní třídy pro jednotlivé způsoby validace dat, kterými je validace validační funkcí (třída `FunctionValidator`), regulárním výrazem (třída `RegularValidator`) nebo slovníkem (třída `DictionaryValidator`) a také validace pomocí vazby (třída `LinkageValidator`). V těchto třídách se nachází konkrétní implementace metody `getResults`, která nabízí všechny varianty vyhledávání osobních údajů ze sekce 4.3.1. O tom, která varianta se použije, rozhoduje uživatel, více o této problematice je uvedeno v sekci 5.3.

Tyto abstraktní třídy rozšiřují třídy pro konkrétní osobní údaje. Název tříd se skládá ze dvou částí, první částí je název osobního údaje v angličtině, druhou částí je slovo „Validator“. Každá třída obsahuje konstruktor, ve kterém je definován přibližný regulární výraz, název údaje a metoda vyhledávání.

Obsah, ve kterém se údaje vyhledávají, je předán jako parametr konstruktoru. Pokud se jedná o osobní údaj, který ke svému ověření potřebuje slovník, je v konstruktoru definován také název slovníku a sloupce slovníku, které se mají použít. Pokud se údaj ověřuje pomocí funkce, obsahuje třída funkci `validation`, které je zadán ověřovaný řetězec znaků, a funkce následně vrátí hodnotu true nebo false podle toho, zda se jedná o daný osobní údaj.

Celá komunikace jednotlivých tříd během procesu extrakce dat a následně vyhledání osobních údajů je zobrazena na následujícím sekvenčním diagramu:

Obrázek 5.1: Sekvenční diagram pro popsané řešení



5.2.4 Vyhledávání osobních údajů s vazbou

Jednou z požadovaných funkcionalit řešení je podpora vyhledávání osobních údajů na základě vazby k jinému osobnímu údaji. V této práci se jedná konkrétně o daňové identifikační číslo a datum narození, které k ověření potřebují rodné číslo. Na rozdíl od ostatních osobních údajů nemají tyto údaje svoje vlastní třídy, ale jednu společnou, která rozšiřuje abstraktní třídu `LinkageValidator`. Třída nese název podle toho osobního údaje, pomocí kterého se další údaje ověřují, a obsahuje metodu `validation`, která ověřuje tento údaj. Dále obsahuje metodu `getBindingData`, která pro nalezený osobní údaj vrací seznam všech možných hodnot údajů, které s nalezeným osobním údajem souvisí.

Implementace metody `getResults` v abstraktní třídě `LinkageValidator` je udělána tak, že program nejprve nalezne osobní údaj, pomocí kterého se ověřují další údaje. Následně pro tyto nalezené hodnoty zavolá metodu `getBindingData` a všechny hodnoty ze seznamu, který tato metoda vrátí, se pokusí vyhledat v daném obsahu.

5.2.5 Ostatní třídy

Kromě tříd, které jsou uvedeny v předchozí sekci, obsahuje tato část řešení ještě třídy `DictionaryController`, `Result` a `UserParams`. Třída `DictionaryController` slouží pro práci se slovníkem. Obsahuje konstruktor, kterému jsou pomocí parametrů předány informace o slovníku, konkrétně jak se slovník jmenuje a v jakých sloupcích jsou hodnoty, které se mají použít. Konstruktor si vytvoří HashSet, do kterého vloží všechny hodnoty ze slovníku, a umožní tak vyhledávání v konstantním čase. Dále třída obsahuje metodu `find`, která vrací hodnotu true nebo false podle toho, zda se ve slovníku nachází hledaný prvek. Dále obsahuje metodu `findPart`, která projde všechny záznamy ve slovníku a vrátí seznam hodnot, u kterých alespoň část hodnoty představuje hledaný prvek. Velmi podobná třída pro práci se slovníky již v nástroji Winch existovala, jedná se o třídu `AbstractDictionary`. Tato třída obsahuje také metodu `find`, avšak tato metoda nevyhledává hodnoty tak efektivně jako metoda ve třídě `DictionaryController`. Dále musí mít zpracovávaný slovník pevně daný formát a neumožňuje tak například ponechat ve slovníku sloupce, u kterých zatím není jisté, zda se někdy využijí. Z tohoto důvodu nebyla použita třída `AbstractDictionary` a byla vytvořena nová třída `DictionaryController`, která je pro vyhledávání v nestrukturovaných datech vhodnější.

Třída `Result` slouží k uložení nalezených osobních údajů. Obsahuje dva atributy, kterými jsou název a konkrétní hodnota osobního údaje. O třídě `UserParams` pojednává kapitola 5.3.

5.3 Konfigurace procesu

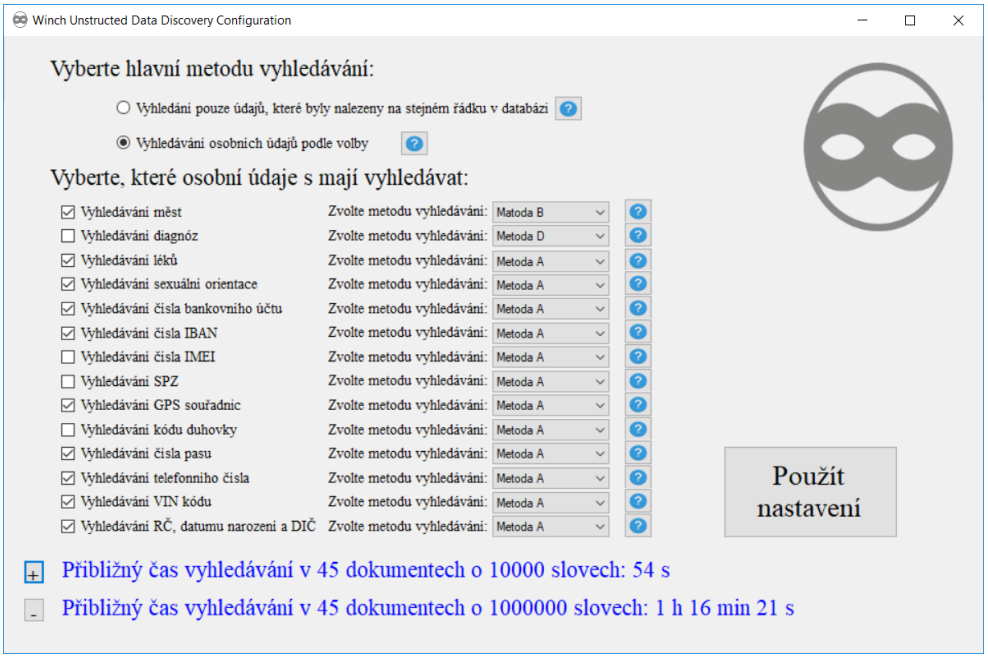
Jak již bylo řečeno, řešení obsahuje třídu `UserParams`, která obsahuje parametry zadané uživatelem. Tyto parametry se týkají celého procesu vyhledávání osobních údajů. Uživatel má možnost zvolit, které osobní údaje se budou vyhledávat a také které metody vyhledávání se k tomu mají použít. Pro toto nastavení byl vytvořen program s jednoduchým grafickým uživatelským rozhraním (obrázek 5.2). Program byl implementován v jazyce C# a týká se především vyhledávání v nestrukturovaných datech, je však možné využít tento program pro strukturovaná data.

V první části programu zvolí uživatel, zda se v nestrukturovaných datech budou vyhledávat všechny zvolené údaje nebo pouze ty, které již byly nalezeny v databázi. V druhé části nastaví uživatel, které údaje se mají vyhledávat, a u každého údaje metodu, kterou se údaj bude vyhledávat. Tyto metody se týkají pouze vyhledávání v nestrukturovaných datech, pro strukturovaná data lze aplikovat pouze volbu údajů, které se mají vyhledávat.

U každé metody vyhledávání je tlačítko s nápovědou, které uživateli popíše, o jakou metodu se jedná. Program také vypočte přibližnou délku trvání procesu vyhledávání v n dokumentech o 10000 a 1000000 slovech, kde parametr n si nastaví uživatel. Je ošetřeno, aby bylo možné nastavit pouze kladné

číslo. Časy pro tento výpočet byly získány na základě testování, které je rozebráno v sekci 6.3.2.

Obrázek 5.2: Program pro nastavení uživatelských parametrů



Po nastavení všech parametrů klikne uživatel na tlačítko „Použít nastavení“. Program uloží zadané parametry v JSON formátu do souboru `configuration.json`, v případě, že soubor neexistuje, vytvoří ho. Uložené parametry si pak načítá třída `UserParams`, která je následně předána třídě `DataDiscoverer` jako parametr konstruktoru. Může se stát, že některá zařízení nebudou program pro nastavení parametrů podporovat. V takovém případě je možné nastavit parametry přímo v souboru `configuration.json`. Zdrojové kódy a `.exe` soubor lze nalézt na přiloženém CD ve složce `configuration`.

5.4 Využité knihovny

Jazyk Groovy obsahuje rozsáhlou sadu základních knihoven, nicméně některé implementované funkcionality vyžadovaly využít externí knihovny. Pro práci se slovníky bylo potřeba využít vhodnou externí knihovnu, která usnadní práci se soubory v `.csv` formátu. Dále bylo třeba využít externí knihovnu, která podporuje co nejvíce formátů souborů a také dokáže sama rozpoznat, o který formát se jedná, a následně z těchto souborů dokáže získat jejich obsah v textové formě. Posledním požadavkem bylo nalézt externí knihovnu, která dokáže dekódovat QR kódy a čárové kódy. Jelikož Groovy je programovací jazyk pro

platformu Javy, lze využít knihovny pro jazyk Java. Implementované řešení využívá následující externí knihovny:

Apache Commons CSV Projekt Apache Commons se zaměřuje na tvorbu znovupoužitelných knihoven obsahujících často prováděné operace, které ve svém kódu opakovaně používá velký počet programátorů. Apache Commons CSV je knihovna usnadňující čtení a zápis CSV[21] souborů.[42]

Apache Tika Apache Tika je knihovna, která se používá pro detekci typu dokumentu a extrakci obsahu z různých formátů souborů. Mezi podporované formáty patří například .pdf, .odf, .txt, .html, .xml, .mp3 a mnoho dalších. Kromě extrakce obsahu umožňuje knihovna také extrahovat metadata souboru, ve kterých se také mohou vyskytovat osobní údaje.[43]

ZXing Knihovna umožňující načítání a dekódování jednoho nebo dvou dimenzionálních kódů. Podporuje širokou škálu kódů jako například UPC-A, Code 39, QR Code, UPC-E, Code 93, Data Matrix, EAN-8, Code 128, Aztec (beta), EAN-13, Codabar, RSS-14 a mnoho dalších.[44]

5.5 Rozšiřitelnost řešení

Řešení bylo navrženo a implementováno tak, aby bylo možné ho v případě potřeby snadno rozšířit. Implementace podporuje extrakci dat ze souborů uložených v databázi a ve filesystému, je však možné přidat třídu implementující rozhraní `FileExtractor`, které bude umožňovat extrakci dat z jiného typu úložiště. V případě, že je potřeba rozšířit nástroj o nový formát souboru, ze kterého se má extrahovat jeho obsah, stačí přidat třídu implementující rozhraní `DataExtractor`. Tato třída musí minimálně obsahovat metodu `getContent`, která vrací textový obsah souboru.

Jedním z nejočekávanějších rozšíření je přidání nových osobních údajů, které bude nástroj schopen vyhledat. V první řadě je potřeba si uvědomit, jakým způsobem se bude osobní údaj ověřovat, zda se k tomu využije validační funkce, regulární výraz, slovník nebo vazba k jinému údaji. Podle toho se vytvoří třída, která rozšíří abstraktní třídu `FunctionValidator`, `RegularValidator`, `DictionaryValidator` nebo `LinkageValidator`. V každé takto vytvořené třídě musí být v konstruktoru nastaveny atributy `commonRegex` (regulární výraz případně přibližný regulární výraz), `name` (název osobního údaje), `content` (obsah, ve kterém se údaj vyhledává, obvykle předaný jako parametr konstruktoru) a navíc je možné nastavit atribut `method` (číslo metody vyhledávání), o nastavení tohoto atributu se však stará třída `DataDiscoverer`. V případě, že je k validaci potřeba využít slovník, je třeba nastavit atribut `dictionaryController`, který si drží instanci třídy `DictionaryController` umožňující snadnou práci se slovníkem.

[21] Comma Separated Value

Kapitola 6

Testování

Součástí této práce je důkladné otestování všech hlavních funkcionalit a také otestování řešení z pohledu výkonu. Tato kapitola tedy popisuje proces testování implementovaného řešení a zhodnocení, které metody vyhledávání ovlivňují nejvíce celkovou dobu procesu.

6.1 Jednotkové testy

Jednotkový test obvykle testuje pouze jednu danou konkrétní jednotku. V ideálním případě by měl být každý testovaný případ nezávislý na ostatních. Tyto testy jsou vytvořeny pro každou anonymizační třídu v případě vyhledávání v databázích a v případě vyhledávání v nestrukturovaných datech jsou vytvořeny testy pro každou validační třídu a pro jednotlivé třídy, které se starají o extrakci dat ze souborů a také pro ostatní jednotlivé třídy.

V případě anonymizačních tříd pro údaje, které se ověřují regulárním výrazem, je testována hlavně funkčnost tohoto regulárního výrazu. Jsou vytvořeny údaje, které regulární výraz musí vyhodnotit jako platné osobní údaje a také údaje, které musí vyhodnotit jako neplatné. Stejně tak je to u údajů, které se ověřují validační funkcí, zde je akorát testována tato validační funkce. U údajů, které jsou ověřovány pomocí slovníků, je testováno vyhledávání platných a neplatných hodnot ve slovnících importovaných do programu. Kromě těchto testů je testována také funkčnost tříd nad jednotlivými databázemi. Test projde dva sloupce databáze, jeden, který daný osobní údaj obsahuje, a druhý, který ho neobsahuje, a je testováno, zda je tato skutečnost správně vyhodnocena.

V případě vyhledávání údajů v nestrukturovaných datech je u každé třídy pro jednotlivé osobní údaje testována funkčnost přibližného regulárního výrazu a metody `getResults`, zda v jednoduchém textu vyhledá dané osobní údaje. Tato metoda nabízí více způsobů vyhledávání (viz sekce 4.3.1) a testy většinou testují všechny tyto způsoby s výjimkou těch, které jsou pro daný

osobní údaj nepoužitelné. Například u vyhledávání diagnóz se očekává většinou víceslovná hodnota, kterou metoda A nemůže vyhledat, protože tato metoda rozdělí text na jednotlivá slova, a ta pak vyhledává ve slovníku diagnóz. Pro údaje, které se ověřují validační funkcí, obsahují třídy navíc funkci `validation`, tato funkce je také zvlášť otestována, zda správně vyhodnotí platné i neplatné údaje.

Testy pro práci se slovníky nejsou testovány ve validačních třídách, ale je vytvořen test pro třídu `DictionaryController`. Ten vyzkouší, zda fungují metody `find` a `findPart`.

Pro třídy, které se starají o extrakci obsahu z jednotlivých souborů, jsou vytvořeny zvlášť soubory a je testováno, zda z nich třídy získají daný obsah.

6.2 Sada testovacích dat

O testování jednotlivých funkcionalit se starají jednotkové testy, je však potřeba otestovat řešení z pohledu výkonu. Za tímto účelem je zapotřebí vytvořit sadu testovacích dat, na kterých bude řešení otestováno. Obecně existují dvě možnosti, jak taková data vytvořit:

1. **Generování dat**

   Jednou z možností je vygenerovat data pomocí regulárních výrazů, které jsou v programu pro jednotlivé osobní údaje definovány, případně pomocí importovaných slovníků. Výhodou této metody je různorodost vygenerovaných dat. Nevýhodou je, že vygenerované osobní údaje neobsahují žádné chyby, které jsou mnohdy potřeba pro vytvoření reálných dat.

2. **Manuálně vytvořená data**

   Druhou možností je vytvořit data manuálně podle potřeby. V praxi se často postupuje tak, že je vytvořen malý reprezentativní vzorek, který je následně duplikován tolikrát, kolikrát je potřeba. Nevýhodou této metody je náročné generování dat a také jejich uniformita. Výhodou je, že data přesně odpovídají požadavkům autora.

Pro otestování řešení byla použita metoda manuálního vytvoření dat, poněvadž se nehledí na přesnost procesu vyhledávání, ale na množství dat, které je potřeba k otestování výkonu. Pro vyhledávání osobních údajů ve strukturovaných datech bylo v jednotlivých databázích vytvořeno několik desítek záznamů, které byly následně duplikovány do množství až 10000 záznamů. Pro vyhledávání osobních údajů v nestrukturovaných datech byly vytvořeny soubory, které obsahují kusy reálných dokumentů, mezi které jsou náhodně vložené potřebné osobní údaje. Textový obsah těchto souborů byl následně také duplikován a byly tak vytvořeny soubory o délce až milion slov.

6.3 Výsledek testování

V této sekci je rozebrán výsledek testování z pohledu časové náročnosti vyhledávání osobních údajů ve strukturovaný a nestrukturovaných datech.

6.3.1 Testování vyhledávání ve strukturovaných datech

Aby bylo možné otestovat řešení z pohledu výkonu, bylo potřeba vytvořit pro testované údaje tabulky o 100, 1000, 10000 záznamech. Tyto tabulky byly vytvořeny pomocí insert skriptů pro jednotlivé databáze (skripty lze nalézt na přiloženém CD). Jelikož byly výsledky pro databáze Oracle SQL a Postgre SQL velmi podobné, jsou v této sekci uvedeny výsledky pouze pro databáze Oracle SQL a Microsoft SQL. Testování proběhlo na virtualizovaném operačním systému Windows s dvoujádrovým procesorem Intel(R) Core(TM) i7-6500U CPU o frekvenci 2,5 GHz, s RAM pamětí 6 GB. Konkrétní naměřené hodnoty slouží pro srovnání rychlosti růstu času dle velikosti dat. Pro vybrané testované údaje byly naměřeny následující hodnoty:

Tabulka 6.1: Vyhledávání čísla pasu v databázích

Počet řádků	**Výsledný čas - MSSQL**	**Výsledný čas - Oracle**
100	406 ms	84 ms
1000	983 ms	110 ms
10000	1163 ms	156 ms

Tabulka 6.2: Vyhledávání VIN v databázích

Počet řádků	**Výsledný čas - MSSQL**	**Výsledný čas - Oracle**
100	304 ms	62 ms
1000	356 ms	109 ms
10000	447 ms	125 ms

Tabulka 6.3: Vyhledávání léků v databázích

Počet řádků	**Výsledný čas - MSSQL**	**Výsledný čas - Oracle**
100	2017 ms	250 ms
1000	4369 ms	321 ms
10000	5228 ms	409 ms

Tabulka 6.4: Vyhledávání diagnóz v databázích

Počet řádků	**Výsledný čas - MSSQL**	**Výsledný čas - Oracle**
100	27044 ms	1361 ms
1000	274493 ms	1422 ms
10000	2312853 ms	1597 ms

Tabulka 6.5: Vyhledávání sexuální orientace v databázích

Počet řádků	**Výsledný čas - MSSQL**	**Výsledný čas - Oracle**
100	297 ms	125 ms
1000	354 ms	172 ms
10000	412 ms	188 ms

Tabulka 6.6: Vyhledávání IMEI v databázích

Počet řádků	**Výsledný čas - MSSQL**	**Výsledný čas - Oracle**
100	115 ms	71 ms
1000	190 ms	245 ms
10000	465 ms	270 ms

Tabulka 6.7: Vyhledávání BBAN v databázích

Počet řádků	**Výsledný čas - MSSQL**	**Výsledný čas - Oracle**
100	187 ms	78 ms
1000	235 ms	109 ms
10000	578 ms	162 ms

Tabulka 6.8: Vyhledávání SPZ v databázích

Počet řádků	**Výsledný čas - MSSQL**	**Výsledný čas - Oracle**
100	152 ms	102 ms
1000	183 ms	108 ms
10000	570 ms	115 ms

Tabulka 6.9: Vyhledávání IBAN v databázích

Počet řádků	**Výsledný čas - MSSQL**	**Výsledný čas - Oracle**
100	267 ms	101 ms
1000	503 ms	122 ms
10000	1365 ms	177 ms

Tabulka 6.10: Vyhledávání GPS v databázích

Počet řádků	**Výsledný čas - Oracle**
100	78 ms
1000	344 ms
10000	562 ms

Tabulka 6.11: Vyhledávání telefonního čísla v databázích

Počet řádků	**Výsledný čas - Oracle**
100	78 ms
1000	109 ms
10000	183 ms

Během testování jsem došel k závěru, že časově nejnáročnější část procesu je vyhledávání diagnóz. V databází Microsoft SQL s tabulkou o 10000 záznamech trvá téměř 40 minut ověřit, zda se v daném sloupci vyskytuje diagnóza. Databáze Oracle SQL nejspíše využívají optimalizované algoritmy vyhledávání, a tak s narůstající velikostí dat narůstá časová náročnost pouze nepatrně. Stejný proces trval při měření zhruba 1,6 vteřiny. Druhý nejnáročnější údaj na vyhledávání jsou léky, tedy opět údaj ověřovaný slovníkem. Obecně lze říci, že nejrychleji probíhá vyhledávání pomocí validační funkce. Vyhledávání pomocí regulárního výrazu je také relativně rychlé, nicméně s narůstající složitostí zápisu regulárního výrazu narůstá výrazně čas vyhledávání, typickým příkladem je vyhledávání GPS souřadnic. Nejdéle trvá vyhledávání s ověřováním pomocí slovníků. Z testování si lze všimnout, že celý proces je výrazně rychlejší v databázi Oracle SQL.

6.3.2 Testování vyhledávání v nestrukturovaných datech

Pro vyhledávání v těchto datech bylo implementováno více metod (metody jsou popsány v sekci 4.3.1) a bylo třeba otestovat, které metody je efektivní použít a také, jak dlouho trvají tyto metody pro jednotlivé osobní údaje. Pro každou z nich byly testovány následující scénáře:

1. Soubor, který je prohledáván, neobsahuje žádné osobní údaje.

2. Soubor, který je prohledáván, obsahuje na každé stránce obvyklý počet osobních údajů (1-2 údaje na každé stránce).

3. Soubor, který je prohledáván, je z velké části tvořen osobními údaji (osobní údaje tvoří přibližně 40% obsahu souboru).

Pro všechny tyto scénáře a pro všechny implementované osobní údaje (s výjimkou snímku oční duhovky) byly vytvořeny 4 dokumenty s různou velikostí. Nejmenší obsahuje vždy kolem 500 slov a velikostí představuje běžný dokument. Zbylé tři dokumenty obsahují přesně 10000, 100000 a 1000000 slov a slouží k otestování rychlosti vyhledávání jednotlivých metod. U každé metody je změřen čas vyhledávání v jednotlivých dokumentech a také je určena úspěšnost podle toho, kolik osobních údajů metoda nalezne. Testování proběhlo na počítači s operačním systémem Windows se čtyřjádrovým procesorem Intel(R) Core(TM) i7-6500U CPU o frekvenci 2,5 GHz, s RAM pamětí 8 GB. Konkrétní naměřené hodnoty slouží pro srovnání rychlosti růstu času dle velikosti dat. Veškeré výsledky včetně závěrů pro jednotlivé osobní údaje naleznete v příloze D.

Následující tabulka obsahuje shrnutí, které popisuje, které metody vyhledávání jsou nejlepší pro jednotlivé osobní údaje. Jejich použití k vyhledávání je pouze doporučené, uživatel si může zvolit kteroukoliv jinou metodu. Zdůvodnění tohoto výsledku je detailně popsáno v příloze D.

Tabulka 6.12: Shrnutí testování vyhledávání osobních údajů

Údaj	**Metoda**
Léky	Metoda B
Diagnózy	Metoda B
Města	Metoda B
Sexuální orientace	Metoda A
BBAN	Metoda B
IBAN	Metoda A
IMEI	Metoda A,B
SPZ	Metoda B
GPS souřadnice	Metoda B
Číslo pasu	Metoda A
Telefonní číslo	Metoda A,B
VIN	Metoda A
Rodné číslo, datum narození, DIČ	Metoda A

Závěr

V této bakalářské práci jsem se zabýval analýzou osobních údajů, kterou se mi podařilo realizovat. Všechny nově přidané osobní údaje jsou v části zabývající se rešeršemi analyzovány z formálního pohledu a zároveň z pohledu uložení těchto údajů ve strukturovaných a nestrukturovaných datech. Nástroj Winch dovede nově vyhledat v tabulkách databází i v dokumentech osobní údaje, o které byl tento nástroj rozšířen.

Pro vyhledávání osobních údajů v nestrukturovaných datech byl vytvořen návrh řešení, který se povedlo úspěšně implementovat. Program nyní zvládá vyhledat osobní údaje v datech, která jsou uložena v databázích pomocí binárních dat nebo ve filesystému. V budoucnu je možné rozšířit implementaci o nové typy úložišť dat. Pro samotný proces vyhledávání je implementováno více metod, z nichž každá může mít uplatnění pro jiný typ osobních údajů. V případě rozšíření o nové osobní údaje jsou všechny tyto metody k dispozici. Skutečnou míru využitelnosti nástroje bude možné určit až při jeho reálném nasazení do rutinního provozu. Nelze vyloučit, že při práci s reálnými daty se objeví problémy, které mé testování neodhalilo, i přestože jsem volil postupy pro maximální možnou eliminaci chyb.

V budoucnu je možné rozšířit nástroj o vyhledávání dalších osobních údajů. Takovým údajem může být například biometrický podpis, který v dnešní době představuje velmi aktuální téma. Dále je možné rozšířit vyhledávání osobních údajů o osobní stav, číslo řidičského průkazu, dosažené vzdělání, příjem ze zaměstnání, trestní delikty, krevní skupinu, Rh faktor krve a jiné.

Literatura

[1] ResearchGate: Unlinkable improved multi-biometric iris fuzzy vault - Scientific Figure on ResearchGate [online]. duben 2019, [cit. 2019-04-14]. Dostupné z: `https://www.researchgate.net/figure/ris-recognition-processing-chain-a-Iris-image-b-Iris-detection-c-Normalized-iris_fig4_309659737`

[2] GARMIN: Position Formats Supported by Garmin Outdoor Devices [online]. duben 2019, [cit. 2019-04-03]. Dostupné z: `https://support.garmin.com/en-US/?faq=lvWzTYlPsx6BvUDTyKfqC8`

[3] server rz-nej.czweb.org: Tabulka nejvyšších spatřených registračních značek typu 101 [online]. duben 2019, [cit. 2019-04-03]. Dostupné z: `http://www.rz-nej.czweb.org/`

[4] Skalský, D.: Vyhledávání osobních údajů v relačních databázích, bakalářská práce. *Praha: České vysoké učení technické v Praze, Fakulta informačních technologií*, 2018.

[5] Úřad pro ochranu osobních údajů: GDPR (obecné nařízení) [online]. duben 2019, [cit. 2019-04-19]. Dostupné z: `https://www.uoou.cz/gdpr%2Dobecne%2Dnarizeni/ds-3938/p1=3938`

[6] Zákon č. 101/2000 Sb., o ochraně osobních údajů a o změně dalších zákonů.

[7] portál GDPR.cz: Osobní údaje [online]. duben 2019, [cit. 2019-04-03]. Dostupné z: `https://www.gdpr.cz/gdpr/heslo/osobni-udaje/`

[8] Hercík, B. J.: Ochrana soukromí a osobních údajů zaměstnanců v obchodní společnosti, diplomová práce. *Brno: Mendelova univerzita v Brně, Provozně ekonomická fakulta*, 2017.

[9] portál GDPR.cz: Citlivé osobní údaje [online]. duben 2019, [cit. 2019-04-03]. Dostupné z: https://www.gdpr.cz/gdpr/heslo/citlive-osobni-udaje/

[10] Sklenák, V.: *Data, informace, znalosti a Internet*. Praha: C.H. Beck, c.h. beck pro praxi vydání, 2001, ISBN 80-7179-409-0.

[11] Seznam.cz: Pokročilé vyhledávání a GPS souřadnice [online]. duben 2019, [cit. 2019-04-19]. Dostupné z: https://napoveda.seznam.cz/cz/mapy/vyhledavani/pokrocile-vyhledavani-gps-souradnice/

[12] International Organization for Standardization: Vehicle ID - ISO coding system paves the way for a smooth ride [online]. duben 2019, [cit. 2019-04-03]. Dostupné z: https://www.iso.org/news/2011/04/Ref1603.htmlUDTyKfqC8

[13] Auto.cz: VIN čísla. Víte, co znamenají a k čemu slouží? A kde je najdete? [online]. duben 2019, [cit. 2019-04-03]. Dostupné z: https://www.auto.cz/vin-cisla-vite-co-znamenaji-a-k-cemu-slouzi-a-kde-je-najdete-105874

[14] Roman Rak, M. P.: *Identifikace vozidel*. Praha: Mobil Data, první vydání, 1999, ISBN 80-238-4157-2.

[15] Marinov, P.: *České poznávací značky – historie, dnešek, budoucnost*. Kampe, Praha, první vydání, 2007, ISBN 978-80-902955-7-5.

[16] Filip Zelený, D. F.: *Poznávací značky v Čechách, na Moravě a ve Slezsku*. SAXI, Praha, první vydání, 2011, ISBN 978-80-904767-2-1.

[17] fdrive.cz: Schváleno: elektromobily a hybridy dostanou v ČR vlastní značku s písmeny EL [online]. duben 2019, [cit. 2019-04-06]. Dostupné z: https://fdrive.cz/clanky/schvaleno-elektromobily-a-hybridy-dostanou-vlastni-znacku-s-pismeny-el-2491

[18] Epřehledy.cz: SPZ - Státní poznávací značky vozidel - okresy [online]. duben 2019, [cit. 2019-04-16]. Dostupné z: https://eprehledy.cz/rz_spz_vozidel.php

[19] Všeobecná zdravotní pojišťovna České republiky: Číselníky [online]. duben 2019, [cit. 2019-04-06]. Dostupné z: https://www.vzp.cz/poskytovatele/ciselniky

[20] Mladá fronta: Sexuální identita. Kdo je kdo? [online]. duben 2019, [cit. 2019-04-12]. Dostupné z: https://www.euro.cz/light/sexualni-identita-kdo-je-kdo-1399090

[21] Microsoft: EU Passport Number [online]. duben 2019, [cit. 2019-04-09]. Dostupné z: https://docs.microsoft.com/en-us/office365/securitycompliance/eu-passport-number

[22] Ministerstvo zahraničních věcí České republiky: Identifikátor diplomatických pasů [online]. duben 2019, [cit. 2019-04-09]. Dostupné z: https://www.mzv.cz/jnp/cz/zahranicni_vztahy/vyrocni_zpravy_a_dokumenty/poskytnute_informace/identifikator_diplomatickych_pasu.html

[23] Český telekomunikační úřad: Telekomunikační Věštník [online]. září 2000, [cit. 2019-04-13]. Dostupné z: https://www.ctu.cz/sites/default/files/obsah/clanky/66429/soubory/telekomunikacni-vestnik-09-2000-1114013634.pdf

[24] PREDVOLBY.CZ: Předvolby mobilních operátorů [online]. duben 2019, [cit. 2019-04-19]. Dostupné z: https://www.predvolby.cz/inpage/mobilni-operatori/

[25] Úrad pre reguláciu elektronických komunikácií a poštových služieb: Pridelené čísla a kódy [online]. duben 2019, [cit. 2019-04-19]. Dostupné z: https://www.ezd.sk/dbpc/databazacisel.php

[26] Objevit.cz: Co je to IMEI telefonu a k čemu je dobré? [online]. duben 2019, [cit. 2019-04-13]. Dostupné z: https://www.objevit.cz/co-je-to-imei-telefonu-a-k-cemu-je-dobre-t177894

[27] Česká správa sociálního zabezpečení: Kontrola IČPE pomocí Luhnova algoritmu [online]. duben 2019, [cit. 2019-04-13]. Dostupné z: https://www.cssz.cz/cz/e-podani/pro-vyvojare/luhnuv-algoritmus-pro-kontrolu-icpe.htm

[28] Zlatá Koruna: Číslo účtu v ČR [online]. duben 2019, [cit. 2019-04-13]. Dostupné z: http://www.zlatakoruna.info/zpravy/ucty/cislo-uctu-v-cr

[29] Česká národní banka: Číselník kódů platebního styku v České republice (ČKPS) [online]. 2019, [cit. 2019-04-13]. Dostupné z: https://www.cnb.cz/cs/platebni_styk/ucty_kody_bank/download/kody_bank_CR.pdf

[30] Česká národní banka: IBAN – mezinárodní formát čísla účtu [online]. duben 2019, [cit. 2019-04-13]. Dostupné z: https://www.cnb.cz/cs/platebni_styk/iban/iban_mezinar_cislo_uctu.html

[31] portál GDPR.cz: Biometrické údaje [online]. duben 2019, [cit. 2019-04-14]. Dostupné z: https://www.gdpr.cz/gdpr/heslo/biometricke-udaje/

[32] Ondrůšek, R.: Identifikační biometrické prostředky. *Zlín: Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky*, 2006.

[33] Křístek, T.: Jednoduchý systém pro snímání duhovky. *Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství*, 2014.

[34] Jucheng Yang, Dong Sun Park, Sook Yoon, Yarui Chen, Chuanlei Zhang: *Machine Learning and Biometrics*. Books on Demand, 2018, ISBN 1789235901, 9781789235906, [Viz str. 11–18.].

[35] Kurzy.cz: DIČ - Daňové identifikační číslo, ověření DIČ firem a osob [online]. duben 2019, [cit. 2019-04-14]. Dostupné z: `https://www.kurzy.cz/dic/`

[36] RCODEMONKEY: [online]. duben 2019, [cit. 2019-04-23]. Dostupné z: `https://www.qrcode-monkey.com/#text`

[37] TEC-IT: Generate Free Barcodes Online [online]. duben 2019, [cit. 2019-04-23]. Dostupné z: `https://barcode.tec-it.com/en`

[38] GenMyModel: Class Diagram Online [online]. duben 2019, [cit. 2019-04-26]. Dostupné z: `https://www.genmymodel.com/class-diagram-online`

[39] Microsoft: binary and varbinary (Transact-SQL) [online]. květen 2019, [cit. 2019-05-04]. Dostupné z: `https://docs.microsoft.com/en-us/sql/t-sql/data-types/binary-and-varbinary-transact-sql?view=sql-server-2017`

[40] W3resource: Oracle Data Types [online]. květen 2019, [cit. 2019-05-04]. Dostupné z: `https://www.w3resource.com/oracle/oracle-data-types.php`

[41] The PostgreSQL Global Development Group: Storing Binary Data [online]. květen 2019, [cit. 2019-05-04]. Dostupné z: `https://jdbc.postgresql.org/documentation/head/binary-data.html`

[42] The Apache Software Foundation: Welcome to Apache Commons [online]. květen 2019, [cit. 2019-05-04]. Dostupné z: `https://commons.apache.org/`

[43] The Apache Software Foundation: Apache Tika - a content analysis toolkit [online]. květen 2019, [cit. 2019-05-04]. Dostupné z: `https://tika.apache.org/`

[44] OWEN, Sean and Tom TASCHE: Official ZXing („Zebra Crossing“) project home. GitHub [online]. květen 2019, [cit. 2019-05-04]. Dostupné z: `https://github.com/zxing/zxing/`

Příloha **A**

Seznam použitých zkratek

BBAN Basic bank account number

BLOB Binary large object

CC Country code

CCD Charge-coupled device

CLOB Character large object

CSV Comma-separated values

ČNB Česká národní banka

ČR Česká republika

ČVUT České vysoké učení technické v Praze

DIČ Daňové identifikační číslo

DPIA Data protection impact assessment

EAN European article number

EU Evropská unie

FAC Final assembly code

GDPR General data protection regulation

GPS Global positioning system

HTML Hypertext markup language

IBAN International bank account number

IČO Identifikační číslo osoby

IMEI International mobile equipment identity

JDBC Java database connectivity

JSON Javascript object notation

MARC Machine-readable cataloging

MSISDN Mobile station international subscriber directory number

NCLOB National character large object

NDC National destination code

ODF Open document format

PDF Portable document format

QR Quick response

RSS Rich site summary

RZ Registrační značka

SN Subscriber number

SNR Serial number

SP Spare

SPZ Státní poznávací značka

SQL Structured query language

TAC Type approval code

UML Unified modeling language

ÚOOÚ Úřad pro ochranu osobních údajů

UPC Universal product code

VDS Vehicle descriptor section

VIN Vehicle identification number

VIS Vehicle identifier selection

WGS World geodetic system

WMI World manufacturer identifier

XML Extensible markup language

Příloha B

Obsah přiloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `src`
  - `discovery` ..... zdrojové kódy discovery části nástroje Winch
  - `discoveryTest` ..... jednotkové testy
  - `configuration` .... zdrojové kódy programu pro nastavení konfigurace
- `testData` ..... testovací data použitá při testování výkonu
- `libraries` ..... využité externí knihovny
- `text` ..... text práce
  - `BP_Chvosta_Tomas_2019.pdf` ..... text práce ve formátu PDF
  - `src` ..... zdrojové soubory práce ve formátu LaTeX

Příloha C

Přehled sexuálních orientací

Heterosexualita: Náklonnost výhradně k osobám odlišného pohlaví. Člověk s touto sexuální orientací se nazývá „heterosexuál“.

Homosexualita: Náklonnost výhradně k osobám stejného pohlaví. Člověk s touto sexuální orientací se nazývá „homosexuál“.

Bisexualita: Náklonnost k osobám obou pohlaví. Člověk s touto sexuální orientací se nazývá „bisexuál“.

Pansexualita: Náklonnost k osobám jakékoliv sexuální orientace nebo identity. Tato orientace je často zaměňována s bisexualitou, ta je ale orientací pouze na muže nebo ženu. Člověk s touto sexuální orientací se nazývá „pansexuál“.

Asexualita: Orientace, při níž člověk necítí náklonnost k žádnému pohlaví. Člověk s touto sexuální orientací se nazývá „asexuál“.

Demisexualita: Orientace, při níž člověk necítí náklonnost k žádnému pohlaví do chvíle, kdy si vytvoří opravdu silné emocionální pouto. Člověk s touto sexuální orientací se nazývá „demisexuál“.

Androsexualita: Orientace na muže nebo spíše na maskulinní zjev. Člověk s touto sexuální orientací se nazývá „androsexuál“.

Gynosexualita: Orientace na ženy nebo spíše na ženský zjev. Člověk s touto sexuální orientací se nazývá „gynosexuál“.

Androgynosexualita: Orientace na muže i ženy bezpohlavního vzezření. Člověk s touto sexuální orientací se nazývá „androgynosexuál“.

Omnisexualita: Omnisexualita je velmi podobná pansexualitě, ale s tím rozdílem, že člověk je spíše přitahován všemi pohlavími, než že by jej pohlaví nezajímalo. Člověk s touto sexuální orientací se nazývá „omnisexuál“.

Sapiosexualita: Náklonnost k inteligenci protějšku. Člověk s touto sexuální orientací se nazývá „sapiosexuál“.

Objectumsexualita: Náklonnost k neživým objektům. Člověk s touto sexuální orientací se nazývá „objectumsexuál“.

Autosexualita: Orientace, při které člověk cítí sexuální přitažlivost jen k sobě. Člověk s touto sexuální orientací se nazývá „autosexuál“.

Polysexualita: Náklonnost k více než jednomu pohlaví nebo pohlavnímu vyjádření. Člověk s touto sexuální orientací se nazývá „polysexuál“.

Příloha D

Testování vyhledávání v nestrukturovaných datech

Vyhledávání a ověření pomocí slovníku

V první části jsou testovány osobní údaje, které se ověřují pomocí slovníků. Jedná se tedy o léky, diagnózy, sexuální orientace. Jelikož slovník sexuálních orientací obsahuje pouze kolem 16 řádků a slovníky léků a diagnóz jsou naopak příliš velké, bylo pro účel tohoto testování implementováno vyhledávání měst, které využívá středně velký slovník (kolem 500 záznamů). Během testování byly naměřeny tyto hodnoty:

Tabulka D.1: Testování metod vyhledávání léků 1. část

Výskyt údajů	**Počet slov**	**Metoda**	**Čas**	**Úspěšnost**
žádné údaje	500	A	0,025 s	100%
žádné údaje	500	B	0,050 s	100%
žádné údaje	500	C	0,036 s	100%
žádné údaje	500	D	0,058 s	100%
žádné údaje	10000	A	0,148 s	100%
žádné údaje	10000	B	0,225 s	100%
žádné údaje	10000	C	0,153 s	100%
žádné údaje	10000	D	0,897 s	100%
žádné údaje	100000	A	1,302 s	100%
žádné údaje	100000	B	2,864 s	100%
žádné údaje	100000	C	1,703 s	100%
žádné údaje	100000	D	28,479 s	100%

Tabulka D.2: Testování metod vyhledávání léků 2. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
žádné údaje	1000000	A	13,295 s	100%
žádné údaje	1000000	B	19,782 s	100%
žádné údaje	1000000	C	13,805 s	100%
žádné údaje	1000000	D	287,112 s	100%

Tabulka D.3: Testování metod vyhledávání léků 3. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
obvyklý	524	A	0,032 s	80%
obvyklý	524	B	0,046 s	100%
obvyklý	524	C	0,042 s	80%
obvyklý	524	D	0,059 s	100%
obvyklý	10000	A	0,109 s	80%
obvyklý	10000	B	0,146 s	100%
obvyklý	10000	C	0,158 s	80%
obvyklý	10000	D	0,852 s	100%
obvyklý	100000	A	0,971 s	80%
obvyklý	100000	B	1,966 s	100%
obvyklý	100000	C	1,362 s	80%
obvyklý	100000	D	20,418 s	100%
obvyklý	1000000	A	10,009 s	80%
obvyklý	1000000	B	12,412 s	100%
obvyklý	1000000	C	10,682 s	80%
obvyklý	1000000	D	200,214 s	100%

Tabulka D.4: Testování metod vyhledávání léků 4. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
nadměrný	604	A	0,030 s	80%
nadměrný	604	B	0,030 s	100%
nadměrný	604	C	0,022 s	80%
nadměrný	604	D	0,097 s	100%
nadměrný	10000	A	0,080 s	80%
nadměrný	10000	B	0,116 s	100%
nadměrný	10000	C	0,100 s	80%
nadměrný	10000	D	0,950 s	100%

Tabulka D.5: Testování metod vyhledávání léků 5. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
nadměrný	100000	A	0,678 s	80%
nadměrný	100000	B	1,850 s	100%
nadměrný	100000	C	1,044 s	80%
nadměrný	100000	D	24,854 s	100%
nadměrný	1000000	A	6,594 s	80%
nadměrný	1000000	B	9,490 s	100%
nadměrný	1000000	C	7,518 s	80%
nadměrný	1000000	D	244,222 s	100%

Ve všech částech testování byly nejrychlejší metody A a C. Tyto metody však hledají pouze jednoslovné osobní údaje a označení léků může obsahovat i více slov, proto ve druhé a třetí části testování mají metody A a C výrazně nižší úspěšnost. 100% úspěšnost mají metody B a D. Metoda B vyhledává všechny hodnoty ze slovníku v dokumentu a je výrazně rychlejší než metoda D, která hledá potenciální kandidáty, kterými je první slovo v názvu léku, a ty se pak pokouší najít v jednotlivých hodnotách slovníku, které pak následně hledá v dokumentu. Nejlepší metodu pro vyhledávání léků v nestrukturovaných datech je tedy metoda B.

Tabulka D.6: Testování metod vyhledávání diagnóz 1. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
žádné údaje	500	A	0,059 s	100%
žádné údaje	500	B	0,170 s	100%
žádné údaje	500	C	0,052 s	100%
žádné údaje	500	D	0,277 s	100%
žádné údaje	10000	A	0,173 s	100%
žádné údaje	10000	B	2,564 s	100%
žádné údaje	10000	C	0,186 s	100%
žádné údaje	10000	D	3,268 s	100%
žádné údaje	100000	A	1,482 s	100%
žádné údaje	100000	B	30,719 s	100%
žádné údaje	100000	C	1,760 s	100%
žádné údaje	100000	D	29,286 s	100%
žádné údaje	1000000	A	14,547 s	100%
žádné údaje	1000000	B	295,672 s	100%
žádné údaje	1000000	C	15,350 s	100%
žádné údaje	1000000	D	272,263 s	100%

Tabulka D.7: Testování metod vyhledávání diagnóz 2. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
obvyklý	524	A	0,056 s	50%
obvyklý	524	B	0,170 s	100%
obvyklý	524	C	0,055 s	50%
obvyklý	524	D	0,344 s	100%
obvyklý	10000	A	0,139 s	50%
obvyklý	10000	B	1,247 s	100%
obvyklý	10000	C	0,141 s	50%
obvyklý	10000	D	2,498 s	100%
obvyklý	100000	A	1,001 s	50%
obvyklý	100000	B	14,253 s	100%
obvyklý	100000	C	1,161 s	50%
obvyklý	100000	D	25,965 s	100%
obvyklý	1000000	A	10,199 s	50%
obvyklý	1000000	B	119,938 s	100%
obvyklý	1000000	C	10,397 s	50%
obvyklý	1000000	D	255,455 s	100%

Tabulka D.8: Testování metod vyhledávání diagnóz 3. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
nadměrný	604	A	0,073 s	10%
nadměrný	604	B	0,178 s	100%
nadměrný	604	C	0,088 s	10%
nadměrný	604	D	0,873 s	100%
nadměrný	10000	A	0,151 s	10%
nadměrný	10000	B	1,501 s	100%
nadměrný	10000	C	0,186 s	10%
nadměrný	10000	D	12,085 s	100%
nadměrný	100000	A	0,962 s	10%
nadměrný	100000	B	15,210 s	100%
nadměrný	100000	C	1,192 s	10%
nadměrný	100000	D	547,742 s	100%
nadměrný	1000000	A	9,779 s	10%
nadměrný	1000000	B	140,320 s	100%
nadměrný	1000000	C	9,984 s	10%
nadměrný	1000000	D	5452,136 s	100%

U vyhledávání diagnóz jsou opět nejrychlejší metody A a C. Jelikož se

však jedná o víceslovný údaj, jsou tyto metody opět nepoužitelné. Nejrychlejší metodou se 100% úspěšností je metoda B. Metoda D má také 100% úspěšnost, ale vyhledávání ve velkých dokumentech, které mají přes 100 stránek, je pro tuto metodu velmi časově náročné.

Tabulka D.9: Testování metod vyhledávání měst 1. část

Výskyt údajů	**Počet slov**	**Metoda**	**Čas**	**Úspěšnost**
žádné údaje	500	A	0,022 s	100%
žádné údaje	500	B	0,029 s	100%
žádné údaje	500	C	0,016 s	100%
žádné údaje	500	D	0,019 s	100%
žádné údaje	10000	A	0,141 s	100%
žádné údaje	10000	B	0,190 s	100%
žádné údaje	10000	C	0,155 s	100%
žádné údaje	10000	D	0,176 s	100%
žádné údaje	100000	A	1,295 s	100%
žádné údaje	100000	B	2,061 s	100%
žádné údaje	100000	C	1,494 s	100%
žádné údaje	100000	D	1,962 s	100%
žádné údaje	1000000	A	13,579 s	100%
žádné údaje	1000000	B	15,515 s	100%
žádné údaje	1000000	C	15,466 s	100%
žádné údaje	1000000	D	17,628 s	100%

Tabulka D.10: Testování metod vyhledávání měst 2. část

Výskyt údajů	**Počet slov**	**Metoda**	**Čas**	**Úspěšnost**
obvyklý	524	A	0,022 s	50%
obvyklý	524	B	0,033 s	100%
obvyklý	524	C	0,018 s	50%
obvyklý	524	D	0,019 s	100%
obvyklý	10000	A	0,103 s	50%
obvyklý	10000	B	0,117 s	100%
obvyklý	10000	C	0,105 s	50%
obvyklý	10000	D	0,159 s	100%

Tabulka D.11: Testování metod vyhledávání měst 3. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
obvyklý	100000	A	0,993 s	50%
obvyklý	100000	B	1,538 s	100%
obvyklý	100000	C	1,154 s	50%
obvyklý	100000	D	3,384 s	100%
obvyklý	1000000	A	9,726 s	50%
obvyklý	1000000	B	10,257 s	100%
obvyklý	1000000	C	9,689 s	50%
obvyklý	1000000	D	31,513 s	100%

Tabulka D.12: Testování metod vyhledávání měst 4. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
nadměrný	720	A	0,022 s	60%
nadměrný	720	B	0,033 s	100%
nadměrný	720	C	0,042 s	60%
nadměrný	720	D	0,053 s	100%
nadměrný	10000	A	0,148 s	60%
nadměrný	10000	B	0,151 s	100%
nadměrný	10000	C	0,141 s	60%
nadměrný	10000	D	0,308 s	100%
nadměrný	100000	A	1,287 s	60%
nadměrný	100000	B	1,815 s	100%
nadměrný	100000	C	1,625 s	60%
nadměrný	100000	D	9,917 s	100%
nadměrný	1000000	A	13,035 s	60%
nadměrný	1000000	B	13,868 s	100%
nadměrný	1000000	C	13,385 s	60%
nadměrný	1000000	D	95,497 s	100%

U vyhledávání měst se používá slovník, který není tak velký jako slovník diagnóz, proto má v tomto testu metoda B srovnatelné časy s metodami A a C, které jsou opět nejrychlejší, ale nedokáží vyhledat města, které mají v názvu více slov. Metodu D lze také využít, ale u velkých dokumentů je třeba počítat s velkou časovou složitostí.

Tabulka D.13: Testování metod vyhledávání sexuálních orientací 1. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
žádné údaje	500	A	0,021 s	100%
žádné údaje	500	B	0,023 s	100%
žádné údaje	500	C	0,020 s	100%
žádné údaje	500	D	0,020 s	100%
žádné údaje	10000	A	0,141 s	100%
žádné údaje	10000	B	0,184 s	100%
žádné údaje	10000	C	0,153 s	100%
žádné údaje	10000	D	0,156 s	100%
žádné údaje	100000	A	1,256 s	100%
žádné údaje	100000	B	1,563 s	100%
žádné údaje	100000	C	1,445 s	100%
žádné údaje	100000	D	1,356 s	100%
žádné údaje	1000000	A	13,080 s	100%
žádné údaje	1000000	B	12,795 s	100%
žádné údaje	1000000	C	13,325 s	100%
žádné údaje	1000000	D	13,214 s	100%

Tabulka D.14: Testování metod vyhledávání sexuálních orientací 2. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
obvyklý	524	A	0,020 s	100%
obvyklý	524	B	0,020 s	100%
obvyklý	524	C	0,024 s	100%
obvyklý	524	D	0,019 s	100%
obvyklý	10000	A	0,099 s	100%
obvyklý	10000	B	0,106 s	100%
obvyklý	10000	C	0,121 s	100%
obvyklý	10000	D	0,130 s	100%
obvyklý	100000	A	0,959 s	100%
obvyklý	100000	B	1,568 s	100%
obvyklý	100000	C	1,148 s	100%
obvyklý	100000	D	1,062 s	100%
obvyklý	1000000	A	10,090 s	100%
obvyklý	1000000	B	9,270 s	100%
obvyklý	1000000	C	10,063 s	100%
obvyklý	1000000	D	9,874 s	100%

Tabulka D.15: Testování metod vyhledávání sexuálních orientací 3. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
nadměrný	518	A	0,018 s	100%
nadměrný	518	B	0,020 s	100%
nadměrný	518	C	0,020 s	100%
nadměrný	518	D	0,025 s	100%
nadměrný	10000	A	0,095 s	100%
nadměrný	10000	B	0,106 s	100%
nadměrný	10000	C	0,115 s	100%
nadměrný	10000	D	0,127 s	100%
nadměrný	100000	A	0,879 s	100%
nadměrný	100000	B	1,294 s	100%
nadměrný	100000	C	1,379 s	100%
nadměrný	100000	D	1,181 s	100%
nadměrný	1000000	A	9,388 s	100%
nadměrný	1000000	B	8,681 s	100%
nadměrný	1000000	C	9,934 s	100%
nadměrný	1000000	D	10,075 s	100%

U vyhledávání sexuálních orientací mají všechny metody 100% úspěšnost, což je dáno tím, že se jedná o jednoslovný údaj, proto lze použít všechny metody. Při porovnání časů si můžeme všimnout velmi vyrovnaných výsledků. U středně velkých dokumentů, které obsahují od 10000 do 100000 slov, je nejrychlejší metoda A. U ostatních dokumentů nelze jednoznačně určit, která metoda je nejlepší, a je tedy možné použít všechny metody.

Z testování vyhledávání těchto 4 osobních údajů vyplývá, že pokud se jedná o víceslovný osobní údaj, který je nutné ověřit pomocí slovníku, není vhodné použít metody A a C i přesto, že jsou nejrychlejší. Metoda B se zdá být ideální volbou pro tyto případy. Metodu D lze také použít, ale oproti metodě B je tato metoda značně pomalejší. Obecně lze říci, že s narůstající velikostí slovníku narůstá časová náročnost těchto metod.

Pokud se nejedná o víceslovný osobní údaj a slovník neobsahuje mnoho osobních údajů, je možné využít všechny metody. V takovém případě se totiž nedá rozhodnout, která metoda je rychlejší. Pokud však velikost slovníku začne narůstat, je třeba počítat s tím, že začne narůstat i časová náročnost metod B a D. Tyto metody totiž procházejí všechny hodnoty ve slovníku a v těchto hodnotách buď hledají nějaký obsah, nebo tyto hodnoty vyhledávají v prohledávaném obsahu. Metody A a C jsou v těchto případech mnohem vhodnější, jelikož pouze vyhledávají konkrétní prvky ve slovníku a slovník je realizován pomocí HashSetu, který umožňuje vyhledat prvek v průměrně konstantním čase.

Vyhledávání a ověření pomocí validační funkce

V této části jsou testovány osobní údaje, které se ověřují pomocí validační funkce. Jedná se tedy o číslo bankovního účtu, číslo IBAN, číslo IMEI a státní poznávací značku. Během testování byly naměřeny tyto hodnoty:

Tabulka D.16: Testování metod vyhledávání BBAN 1. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
žádné údaje	500	A	0,027 s	100%
žádné údaje	500	B	0,021 s	100%
žádné údaje	10000	A	0,159 s	100%
žádné údaje	10000	B	0,155 s	100%
žádné údaje	100000	A	1,415 s	100%
žádné údaje	100000	B	1,621 s	100%
žádné údaje	1000000	A	14,535 s	100%
žádné údaje	1000000	B	13,427 s	100%

Tabulka D.17: Testování metod vyhledávání BBAN 2. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
obvyklý	524	A	0,024 s	100%
obvyklý	524	B	0,021 s	100%
obvyklý	10000	A	0,135 s	100%
obvyklý	10000	B	0,114 s	100%
obvyklý	100000	A	1,142 s	100%
obvyklý	100000	B	1,383 s	100%
obvyklý	1000000	A	11,437 s	100%
obvyklý	1000000	B	9,807 s	100%

Tabulka D.18: Testování metod vyhledávání BBAN 3. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
nadměrný	512	A	0,024 s	100%
nadměrný	512	B	0,034 s	100%
nadměrný	10000	A	0,151 s	100%
nadměrný	10000	B	0,129 s	100%
nadměrný	100000	A	1,386 s	100%
nadměrný	100000	B	1,786 s	100%
nadměrný	1000000	A	14,576 s	100%
nadměrný	1000000	B	12,357 s	100%

U vyhledávání čísla bankovního účtu je ve většině případů rychlejší metoda B. Pouze v případech, kdy má dokument okolo 100000 slov, je nepatrně rychlejší metoda A. Metoda A však funguje tak, že rozdělí daný obsah na jednotlivá slova a jedno po druhém kontroluje. Tedy ve chvíli, kdy se v čísle nachází mezera, číslo není rozpoznáno jako číslo bankovního účtu. Je tedy vhodnější využít metodu B, ta funguje v tomto ohledu spolehlivěji.

Tabulka D.19: Testování metod vyhledávání IBAN 1. část

Výskyt údajů	**Počet slov**	**Metoda**	**Čas**	**Úspěšnost**
žádné údaje	500	A	0,023 s	100%
žádné údaje	500	B	0,028 s	100%
žádné údaje	10000	A	0,190 s	100%
žádné údaje	10000	B	0,211 s	100%
žádné údaje	100000	A	1,460 s	100%
žádné údaje	100000	B	1,646 s	100%
žádné údaje	1000000	A	14,120 s	100%
žádné údaje	1000000	B	12,912 s	100%

Tabulka D.20: Testování metod vyhledávání IBAN 2. část

Výskyt údajů	**Počet slov**	**Metoda**	**Čas**	**Úspěšnost**
obvyklý	524	A	0,019 s	100%
obvyklý	524	B	0,021 s	100%
obvyklý	10000	A	0,108 s	100%
obvyklý	10000	B	0,106 s	100%
obvyklý	100000	A	1,136 s	100%
obvyklý	100000	B	1,313 s	100%
obvyklý	1000000	A	10,499 s	100%
obvyklý	1000000	B	9,519 s	100%

Tabulka D.21: Testování metod vyhledávání IBAN 3. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
nadměrný	287	A	0,018 s	100%
nadměrný	287	B	0,030 s	100%
nadměrný	10000	A	0,182 s	100%
nadměrný	10000	B	0,173 s	100%
nadměrný	100000	A	1,517 s	100%
nadměrný	100000	B	2,009 s	100%
nadměrný	1000000	A	15,544 s	100%
nadměrný	1000000	B	16,833 s	100%

U vyhledávání bankovního účtu ve formátu IBAN je tomu přesně naopak než u klasického čísla bankovního účtu. Ve většině případů je rychlejší metoda A, a poněvadž IBAN je běžně psán bez mezer, je vhodnější využít pro jeho vyhledání metodu A.

Tabulka D.22: Testování metod vyhledávání IMEI 1. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
žádné údaje	500	A	0,019 s	100%
žádné údaje	500	B	0,015 s	100%
žádné údaje	10000	A	0,151 s	100%
žádné údaje	10000	B	0,207 s	100%
žádné údaje	100000	A	1,332 s	100%
žádné údaje	100000	B	1,544 s	100%
žádné údaje	1000000	A	13,910 s	100%
žádné údaje	1000000	B	13,680 s	100%

Tabulka D.23: Testování metod vyhledávání IMEI 2. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
obvyklý	524	A	0,018 s	100%
obvyklý	524	B	0,022 s	100%
obvyklý	10000	A	0,118 s	100%
obvyklý	10000	B	0,108 s	100%
obvyklý	100000	A	1,200 s	100%
obvyklý	100000	B	1,235 s	100%
obvyklý	1000000	A	10,628 s	100%
obvyklý	1000000	B	9,602 s	100%

Tabulka D.24: Testování metod vyhledávání IMEI 3. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
nadměrný	432	A	0,020 s	100%
nadměrný	432	B	0,018 s	100%
nadměrný	10000	A	0,124 s	100%
nadměrný	10000	B	0,110 s	100%
nadměrný	100000	A	0,995 s	100%
nadměrný	100000	B	1,464 s	100%
nadměrný	1000000	A	10,132 s	100%
nadměrný	1000000	B	9,676 s	100%

U vyhledávání IMEI jsou časové náročnosti obou metod velmi podobné. U velkých dokumentů, které obsahují až milion slov, je rychlejší metoda B. V jiných případech je mnohdy rychlejší metoda A. Tyto rozdíly jsou však nepatrné a dají se tedy použít obě metody.

Tabulka D.25: Testování metod vyhledávání SPZ 1. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
žádné údaje	500	A	0,019 s	100%
žádné údaje	500	B	0,027 s	100%
žádné údaje	10000	A	0,161 s	100%
žádné údaje	10000	B	0,212 s	100%
žádné údaje	100000	A	1,430 s	100%
žádné údaje	100000	B	1,530 s	100%
žádné údaje	1000000	A	14,556 s	100%
žádné údaje	1000000	B	13,041 s	100%

Tabulka D.26: Testování metod vyhledávání SPZ 2. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
obvyklý	524	A	0,026 s	50%
obvyklý	524	B	0,022 s	100%
obvyklý	10000	A	0,118 s	50%
obvyklý	10000	B	0,109 s	100%
obvyklý	100000	A	1,066 s	50%
obvyklý	100000	B	1,359 s	100%
obvyklý	1000000	A	11,252 s	50%
obvyklý	1000000	B	9,533 s	100%

Tabulka D.27: Testování metod vyhledávání SPZ 3. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
nadměrný	689	A	0,032 s	60%
nadměrný	689	B	0,037 s	100%
nadměrný	10000	A	0,086 s	60%
nadměrný	10000	B	0,940 s	100%
nadměrný	100000	A	0,758 s	60%
nadměrný	100000	B	1,520 s	100%
nadměrný	1000000	A	8,119 s	60%
nadměrný	1000000	B	7,270 s	100%

U vyhledávání státní poznávací značky je rychlejší metoda A s výjimkou velký dokumentů, které mají okolo milionu slov. Je však třeba brát ohled na to, že textová hodnota SPZ bývá velmi často rozdělena na dvě části a uložena s mezerou. Takto uložené hodnoty metoda A nerozpozná. Je tedy lepší využít metodu B.

Z testování těchto údajů vyplývá, že při vyhledávání s ověřením pomocí validační funkce je rozhodně spolehlivější metoda B. Metoda A je v mnoha případech rychlejší, ale spoustu platných osobních údajů nevyhledá. Časové rozdíly těchto metod jsou však nepatrné, a proto je lepší pro tento typ vyhledávání použít metodu B.

Vyhledávání a ověření pomocí regulárního výrazu

V této části jsou testovány osobní údaje, které se ověřují pomocí regulárního výrazu. Jedná se tedy o GPS souřadnice, číslo pasu, telefonní číslo a VIN kód. Během testování byly naměřeny tyto hodnoty:

Tabulka D.28: Testování metod vyhledávání GPS 1. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
žádné údaje	500	A	0,021 s	100%
žádné údaje	500	B	0,027 s	100%
žádné údaje	10000	A	0,280 s	100%
žádné údaje	10000	B	0,158 s	100%
žádné údaje	100000	A	2,181 s	100%
žádné údaje	100000	B	1,643 s	100%
žádné údaje	1000000	A	21,611 s	100%
žádné údaje	1000000	B	13,997 s	100%

Tabulka D.29: Testování metod vyhledávání GPS 2. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
obvyklý	524	A	0,043 s	0%
obvyklý	524	B	0,038 s	100%
obvyklý	10000	A	0,194 s	0%
obvyklý	10000	B	0,111 s	100%
obvyklý	100000	A	1,780 s	0%
obvyklý	100000	B	1,235 s	100%
obvyklý	1000000	A	18,567 s	0%
obvyklý	1000000	B	10,398 s	100%

Tabulka D.30: Testování metod vyhledávání GPS 3. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
nadměrný	524	A	0,028 s	0%
nadměrný	524	B	0,020 s	100%
nadměrný	10000	A	0,193 s	0%
nadměrný	10000	B	0,110 s	100%
nadměrný	100000	A	1,806 s	0%
nadměrný	100000	B	1,408 s	100%
nadměrný	1000000	A	18,247 s	0%
nadměrný	1000000	B	10,028 s	100%

U vyhledávání GPS souřadnic je ve všech případech značně rychlejší i úspěšnější metoda B. Metoda A navíc není schopná tento údaj vyhledat, proto ji nelze použít.

Tabulka D.31: Testování metod vyhledávání čísla pasu 1. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
žádné údaje	500	A	0,027 s	100%
žádné údaje	500	B	0,025 s	100%
žádné údaje	10000	A	0,175 s	100%
žádné údaje	10000	B	0,183 s	100%
žádné údaje	100000	A	1,373 s	100%
žádné údaje	100000	B	1,590 s	100%
žádné údaje	1000000	A	14,256 s	100%
žádné údaje	1000000	B	13,265 s	100%

Tabulka D.32: Testování metod vyhledávání čísla pasu 2. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
obvyklý	524	A	0,020 s	100%
obvyklý	524	B	0,021 s	100%
obvyklý	10000	A	0,116 s	100%
obvyklý	10000	B	0,123 s	100%
obvyklý	100000	A	1,041 s	100%
obvyklý	100000	B	1,416 s	100%
obvyklý	1000000	A	11,206 s	100%
obvyklý	1000000	B	9,796 s	100%

Tabulka D.33: Testování metod vyhledávání čísla pasu 3. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
nadměrný	576	A	0,015 s	100%
nadměrný	576	B	0,023 s	100%
nadměrný	10000	A	0,089 s	100%
nadměrný	10000	B	0,104 s	100%
nadměrný	100000	A	0,750 s	100%
nadměrný	100000	B	1,321 s	100%
nadměrný	1000000	A	8,095 s	100%
nadměrný	1000000	B	7,454 s	100%

V případě vyhledávání čísla pasu je nepatrně rychlejší metoda A s výjimkou dokumentů, které obsahují kolem milionu slov. Obě metody jsou 100% úspěšné. Je tedy možné použít obě metody, ale vhodnější je využít metodu A.

Tabulka D.34: Testování metod vyhledávání telefonního čísla 1. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
žádné údaje	500	A	0,019 s	100%
žádné údaje	500	B	0,022 s	100%
žádné údaje	10000	A	0,180 s	100%
žádné údaje	10000	B	0,180 s	100%
žádné údaje	100000	A	1,666 s	100%
žádné údaje	100000	B	1,693 s	100%
žádné údaje	1000000	A	16,044 s	100%
žádné údaje	1000000	B	14,490 s	100%

Tabulka D.35: Testování metod vyhledávání telefonního čísla 2. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
obvyklý	524	A	0,023 s	100%
obvyklý	524	B	0,020 s	100%
obvyklý	10000	A	0,147 s	100%
obvyklý	10000	B	0,124 s	100%
obvyklý	100000	A	1,223 s	100%
obvyklý	100000	B	1,351 s	100%
obvyklý	1000000	A	12,854 s	100%
obvyklý	1000000	B	9,829 s	100%

Tabulka D.36: Testování metod vyhledávání telefonního čísla 3. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
nadměrný	483	A	0,027 s	100%
nadměrný	483	B	0,022 s	100%
nadměrný	10000	A	0,131 s	100%
nadměrný	10000	B	0,108 s	100%
nadměrný	100000	A	1,176 s	100%
nadměrný	100000	B	1,366 s	100%
nadměrný	1000000	A	11,815 s	100%
nadměrný	1000000	B	9,123 s	100%

U vyhledávání telefonních čísel jsou časové náročnosti obou metod velmi podobné a není možné jednoznačně určit, která metoda je lepší. Úspěšnost metod je opět 100% v obou případech.

Tabulka D.37: Testování metod vyhledávání VIN 1. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
žádné údaje	500	A	0,020 s	100%
žádné údaje	500	B	0,021 s	100%
žádné údaje	10000	A	0,151 s	100%
žádné údaje	10000	B	0,154 s	100%
žádné údaje	100000	A	1,373 s	100%
žádné údaje	100000	B	1,517 s	100%
žádné údaje	1000000	A	14,095 s	100%
žádné údaje	1000000	B	12,094 s	100%

Tabulka D.38: Testování metod vyhledávání VIN 2. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
obvyklý	524	A	0,027 s	100%
obvyklý	524	B	0,020 s	100%
obvyklý	10000	A	0,112 s	100%
obvyklý	10000	B	0,112 s	100%
obvyklý	100000	A	0,997 s	100%
obvyklý	100000	B	1,333 s	100%
obvyklý	1000000	A	10,644 s	100%
obvyklý	1000000	B	9,240 s	100%

Tabulka D.39: Testování metod vyhledávání VIN 3. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
nadměrný	367	A	0,019 s	100%
nadměrný	367	B	0,037 s	100%
nadměrný	10000	A	0,125 s	100%
nadměrný	10000	B	0,131 s	100%
nadměrný	100000	A	1,153 s	100%
nadměrný	100000	B	1,716 s	100%
nadměrný	1000000	A	11,510 s	100%
nadměrný	1000000	B	10,611 s	100%

U vyhledávání VIN kódů je nepatrně rychlejší metoda A. Pouze u velkých dokumentů je rychlejší metoda B. Úspěšnost obou metod je opět 100%.

Z testování těchto údajů vyplývá, že čím složitější je regulární výraz, tím je lepší použít metodu. Zároveň pokud údaj obsahuje mezery nebo jiné bílé znaky, není možné použít metodu A. Metodu B je tedy lepší použít pro vyhledávání GPS souřadnic. Metoda A je vhodnější pro vyhledávání čísla pasu a VIN kódu. Pro telefonní číslo lze využít obě metody.

Vyhledávání a ověření pomocí vazby

V této části jsou testovány osobní údaje, které se ověřují pomocí vazby. Jedná se tedy o datum narození a DIČ. K tomu, aby mohly být tyto údaje vyhledány, musí program nejprve najít rodné číslo, pomocí kterého tyto údaje vyhledá a ověří. Rodné číslo ověří program pomocí validační funkce, proto lze pro toto vyhledávání použít metody A a B, které jsou shodné jako metody pro vyhledávání údajů s ověřením pomocí validační funkce. Během testování byly naměřeny tyto hodnoty:

Tabulka D.40: Testování metod vyhledávání data narození a DIČ 1. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
žádné údaje	500	A	0,023 s	100%
žádné údaje	500	B	0,026 s	100%
žádné údaje	10000	A	0,169 s	100%
žádné údaje	10000	B	0,161 s	100%
žádné údaje	100000	A	1,459 s	100%
žádné údaje	100000	B	1,677 s	100%
žádné údaje	1000000	A	14,808 s	100%
žádné údaje	1000000	B	13,502 s	100%

Tabulka D.41: Testování metod vyhledávání data narození a DIČ 2. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
obvyklý	524	A	0,024 s	100%
obvyklý	524	B	0,020 s	100%
obvyklý	10000	A	0,127 s	100%
obvyklý	10000	B	0,116 s	100%
obvyklý	100000	A	1,657 s	100%
obvyklý	100000	B	2,741 s	100%
obvyklý	1000000	A	16,621 s	100%
obvyklý	1000000	B	20,643 s	100%

Tabulka D.42: Testování metod vyhledávání data narození a DIČ 3. část

Výskyt údajů	Počet slov	Metoda	Čas	Úspěšnost
nadměrný	616	A	0,020 s	100%
nadměrný	616	B	0,019 s	100%
nadměrný	10000	A	0,192 s	100%
nadměrný	10000	B	0,315 s	100%
nadměrný	100000	A	8,713 s	100%
nadměrný	100000	B	21,309 s	100%
nadměrný	1000000	A	87,619 s	100%
nadměrný	1000000	B	207,170 s	100%

Z testování vyplývá, že pro vyhledání rodného čísla je lepší využít metodu A, jelikož je ve většině případů rychlejší. Vyhledávání data narození a DIČ probíhá u obou metod stejným způsobem, tudíž není třeba provádět testo-

vání zvlášť. Z tabulek si můžeme všimnout, že s narůstajícím počtem údajů v dokumentu narůstá rapidně časová náročnost.