Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta

Nadstavba forenzního nástroje FTK Forensic Toolkit

Bakalářská práce

Pavel Sýkora

Vedoucí práce: Ing. Jaroslav Kothánek, Ph.D.

České Budějovice 2019

Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta

ZADÁVACÍ PROTOKOL BAKALÁŘSKÉ PRÁCE

Pavel Sýkora
Student: .....
(jméno, příjmení, tituly)

Aplikovaná informatika
Obor – zaměření studia: .....

Katedra: **Ústav aplikované informatiky, oddělení forenzních věd a kriminalistiky**

Ing. Jaroslav Kothánek, Ph.D.
Školitel: .....
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

Nadstavba forenzního nástroje FTK Forensic Toolkit
Téma bakalářské práce: .....

Úkoly a cíle práce :

1. Seznamte se s problematikou zásad forenzního zkoumání digitální techniky
2. Seznamte se s forenzním nástrojem FTK Forensic Toolkit
3. Proveďte analýzu reportů a vytvořte aplikaci, která bude v českém jazyce, umožní řazení dle jednotlivých použitých položek a jednoduché vyhledávání textových řetězců
4. Uvedenou aplikaci vytvořte tak, aby byla schopna pracovat samostatně bez instalace a plnila veškeré požadavky na důkazní řízení. Je nutné, aby nebylo nutné výsledný report instalovat a bylo s ním možno dynamicky pracovat.
5. Uvedenou aplikaci vytvořte v souladu s postupy vývoje softwarových aplikací (vytvoření dokumentace, návodu)
6. Aplikaci vyzkoušejte s minimálně 500 000 dokumentovanými položkami s jednou úrovní záložek v reportu a zhodnoťte využitelnost aplikace z časového hlediska

Základní doporučená literatura :

1. Fratepietro F., Rossetti P., DEFT User Guide, http://www.deftlinux.net/
2. Carrian B., File Systém Forensic Analysis, Addison Wesley Professional, ISBN: 0-32-126817-2
3. http://www.accessdata.com

Financování práce :.....

Vedoucí práce : **Ing. Jaroslav Kothánek, Ph.D.** podpis : .....

U externích vedoucích fakultní garant práce.....podpis : .....

Vedoucí oddělení **Ing. Jaroslav Kothánek, Ph.D.** podpis .....

Případný souhlas vedoucího ústavu AV .....podpis : .....

V Českých Budějovicích dne 1.2.2018

Převzal/a dne 8.2.2018 podpis : .....

Bibliografické údaje

Sýkora P., 2019: Nadstavba forenzního nástroje FTK Forensic Toolkit[FTK Forensic Toolkit report extension. Bc. Thesis, in Czech] - 43p., Faculty of Science, University of South Bohemia, České Budějovice, Czech Republic.

Abstrakt

Tématem bakalářské práce je vytvoření aplikace zpracovávající výstup z forenzního nástroje FTK Forensic Toolkit. Práce popisuje problematiku zásad forenzního zkoumání digitální techniky, forenzní nástroj FTK Forensic Toolkit a analyzuje výstupy z FTK. Bakalářská práce popisuje návrh, vývoj a vyhodnocení aplikace. Aplikace je tvořena pomocí programovacího jazyka Java.

Klíčová slova

FTK Forensic Toolkit, report, soubor, HTML, CSV

Abstract

The topic of this thesis is the creation of an application that processes the output of the forensic tool FTK Forensic Toolkit. The thesis describes the principles of forensic investigation of digital technology, forensic tool FTK Forensic Toolkit and analyzes outputs from FTK. Thesis describes the design, development and evaluation of the application. The application is created in Java programming language.

Keywords

FTK Forensic Toolkit, report, file, HTML, CSV

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích, dne 13. dubna 2019 Podpis .....

Poděkování

Rád bych poděkoval vedoucímu bakalářské práce Ing. Jaroslav Kothánkovi, Ph.D. za ochotu, trpělivost a cenné rady, které mi věnoval v průběhu tvorby této práce. Také děkuji svým nejbližším za podporu při jejím psaní i během celého studia.

Obsah

1. Úvod

V současné době, kdy jsou informační a komunikační technologie nedílnou součástí běžného osobního i pracovního života, tak roste i jejich zneužívání k trestné činnosti. Ovšem už dávno neplatí, že by se digitální důkazy vyhledávaly jen v případech takzvané počítačové nebo informační kriminality, protože moderní technologie uchovávají takové množství cenných informací, že i v případech nesouvisejících s elektronickými přístroji je nutné tato data vyhledávat. Protože každá naše interakce s elektronickým zařízením zanechává digitální stopu (poslaná zpráva, platba kartou, přihlášení k počítači, zachycení osoby na záznamu bezpečnostních kamer atd.). Tyto digitální stopy jsou data, která se dají použít ve vyšetřování trestné činnosti, stejně tak jako například otisky prstů nebo testy DNA. Proto existují forenzní vědy zaměřené právě na tato data. Zajišťování digitálních dat má svá pravidla a využívají se forenzní nástroje uzpůsobené na jejich vyhledávání, zkoumání a prezentování. Forenzních nástrojů je mnoho a můžeme je rozdělit podle jejich specializace - například na mobilní zařízení (mobilní telefony, tablety) nebo na počítače a notebooky.

Tato práce popisuje digitální forenzní analýzu a zaměřuje se na konkrétní nástroj Forensic Toolkit (FTK) od společnosti AccessData, který je považován za standard mezi forenzními nástroji. FTK dokáže zpracovat zajištěná data, procházet je, vyhledávat v nich a umožňuje nalezená data prezentovat v podobě reportů. Ovšem jedná se o relativně drahý software, a pokud prezentovaný report obsahuje velké množství informací, stává se nepřehledným. Součástí této práce je popsání práce s FTK, vytváření reportu, navrhnutí a vytvoření aplikace, která dokáže report opět načíst a umožní obsažené informace znovu procházet, řadit, vyhledávat v nich a bude schopna vytvářet nový report z nalezených informací.

1.1 Cíle práce

Práce obsahuje následující body:

- obeznámení s problematikou zásad forenzního zkoumání digitální techniky
- seznámení s forenzním nástrojem FTK Forensic Toolkit
- analýza reportu z FTK
- vytvoření aplikace zpracovávající report z FTK, který bude v českém jazyce, umožní jednoduché vyhledávání textových řetězců a řazení dle jednotlivých použitých položek
- schopnost aplikace pracovat samostatně bez instalace a plnění požadavků na důkazní řízení
- dokumentace a návod k aplikaci
- otestování aplikace s minimálně 500 000 dokumentovanými položkami
- zhodnocení aplikace z časového hlediska

2. Forenzní zkoumání digitální techniky

2.1 Forenzní vědy

Jedná se o vědy používané k zajišťování a analýze důkazů při vyšetřování trestné činnosti. Většina těchto věd vychází z přírodních vědních oborů jako jsou fyzika, chemie, biologie, matematika a psychologie. Patří sem například forenzní chemie, forenzní biologie, daktyloskopie, mechanoskopie, trasologie, balistika. Ovšem zásadním rozdílem mezi přírodními vědními obory a forenzními vědami jsou nejen specifická kritéria pro sběr a uchovávání vzorků, ale i výslednou prezentací zjištěných poznatků a závěrů.

2.2 Digitální forenzní analýza

Jak již bylo řečeno, většina forenzních věd má svého konkrétního "rodiče", což se ovšem o digitální forenzní analýze (DFA) říci nedá. DFA se stará o veškerá digitální data ze všech elektronických zařízení, jako jsou počítače, notebooky, tablety, mobilní telefony, paměťová média a mnoho dalších.

Základní kroky DFA:

1. vyhledávání a zajišťování digitálních dat
2. zkoumání a analyzování digitálních dat
3. dokumentace a prezentace získaných dat

Ovšem aby mohly být všechny tyto zajištěné digitální informace použité jako důkazy, musí se DFA řídit určitými zásadami.

2.3 Zásady forenzní analýzy

Tyto zásady nejsou nikde v ČR právně ani jiným způsobem definovány, jedná se o zásady vycházející z praxe a zahraničního doporučení. Pouze podjatost je specifikována zákonem o znalcích a tlumočnících jako možnost, pro kterou může být znalec ze zkoumání vyloučen.

První zásada *Legalita* - veškeré informace, stopy, vzorky, předměty, dokumenty atp., které slouží jako zdroj/vstup DFA, metody a způsoby zpracování, a tedy i výstupy DFA musí být získány, pořízeny a zhotoveny legálním způsobem.

Druhá zásada *Integrita* - vše, co bylo prováděno, veškeré způsoby práce se vstupními informacemi (stopy, vzorky...), musí být prováděno způsobem, ze kterého je jednoznačně jasné, že nemohlo dojít k úmyslné nebo neúmyslné manipulaci nebo změně, kdo, kdy, kde, jak a proč s nimi co dělal apod.

Třetí zásada *Opakovatelnost/přezkoumatelnost* - použití takových způsobů práce a jejich dokumentace tak, aby metody mohly být opakovaně provedeny stejným způsobem, čímž by se ověřilo, zda se dospěje ke stejným závěrům, nebo aby pomocí jiných ekvivalentních metod (pokud existují) mohla být správnost závěrů ověřena.

Čtvrtá zásada *Nepodjatost* - nezávislost subjektu provádějícího forenzní činnosti na zkoumaném předmětu nebo objektu.

Aby bylo možné prokázat nějaké závěry a doložit naplnění výše uvedených zásad, musí být vedena detailní dokumentace.[1]

2.4 Vyhledávaná data

Nejčastěji vyhledávaná data pomocí DFA:

2.4.1 Komunikace

Jedná se o data vytvářená při komunikaci, například pomocí e-mailu, sms nebo chatové konverzace. U těchto dat se vyhledávají údaje jako například kdo s kým komunikoval, kdy komunikace probíhala, obsah komunikace, přílohy.

2.4.2 Dokumenty

Jedná se o veškeré dokumenty v digitální podobě. U těchto dat se vyhledávají údaje jako například obsah dokumentu, autor dokumentu, časový údaj vytvoření dokumentu, časový údaj posledního přístupu k dokumentu, časový údaj posledního uložení.

2.4.3 Multimédia

Jedná se o veškeré obrázky, fotografie, videa a zvukové stopy. U těchto dat se vyhledávají údaje jako například obsah multimédia, časový údaj vytvoření, místo pořízení (GPS souřadnice u fotografií).

3. FTK Forensic Toolkit

Forensic Toolkit (FTK) je počítačový software umožnující provádět důkladné forenzní zkoumání zahrnující výkonné filtrování souborů, vyhledávání a přístup ke vzdáleným systémům v síti. Tento software napomáhá orgánům činným v trestném řízení, bezpečnostním složkám a IT znalcům vyhodnocovat a přistupovat k důkazům ze souborů, složek a počítačů.

K této bakalářské práci bylo použito FTK ve verzi 6.4 vydané 1. února 2018.

3.1 Získávání digitálních důkazů

Aby mohly být digitální důkazy použity u soudu, musí být zachována integrita zdrojových dat při zajišťování. Proto se při získávání digitálních důkazů musí vytvářet kopie těchto důkazů, aby se zabránilo jakékoliv úpravě nebo změně těchto dat. Tyto kopie se nazývají forenzní obrazy (forensic image). Pokud dojde ke zpochybnění těchto důkazů, dojde k porovnání původních dat s tímto obrazem pro dokázání integrity.

Vytvoření forenzního obrazu musí být provedeno takovým způsobem, aby bylo možné zaručit, že původní data nebyla nijak narušena ani pozměněna. Musí se jednat o tzv. bitovou kopii, kdy jsou původní data duplikována bit po bitu. Pro vytvoření takovéto kopie je možné použít software AccessData Imager.

Je velmi důležité chránit jak získané důkazy v podobě forenzních obrazů, tak i zaznamenané důkazy pomocí FTK. Proto FTK nabízí možnost šifrování pomocí AD Encryption.

3.2 Typy digitálních důkazů

Digitální důkazy jsou elektronická data jako například dokumenty a e-maily, které mohou být přenášeny a ukládány na elektronická zařízení jako jsou pevné disky, mobilní telefony nebo USB Flash disky.

Digitální důkazy se mohou rozdělit na tři druhy podle způsobu jejich zajištění.

3.2.1 Static evidence (Statické důkazy)

Jedná se o data, která jsou vytvořena již před založením případu v podobě obrazu disku. Tato data jsou neměnná a zůstávají přístupná pro případ po celou dobu.
Obrazy disku jsou vytvářeny pomocí nástrojů, které se dělí na:

- Hardwarové - duplikují nebo klonují paměťové médium a podporují read-olny přístup, který chrání médium před nechtěným zápisem. Většinou se jedná o ruční zařízení.
- Softwarové - za použití softwarového nástroje se duplikuje paměťové médium a vytváří se jeho obraz.

3.2.2 Live evidence (Živé důkazy)

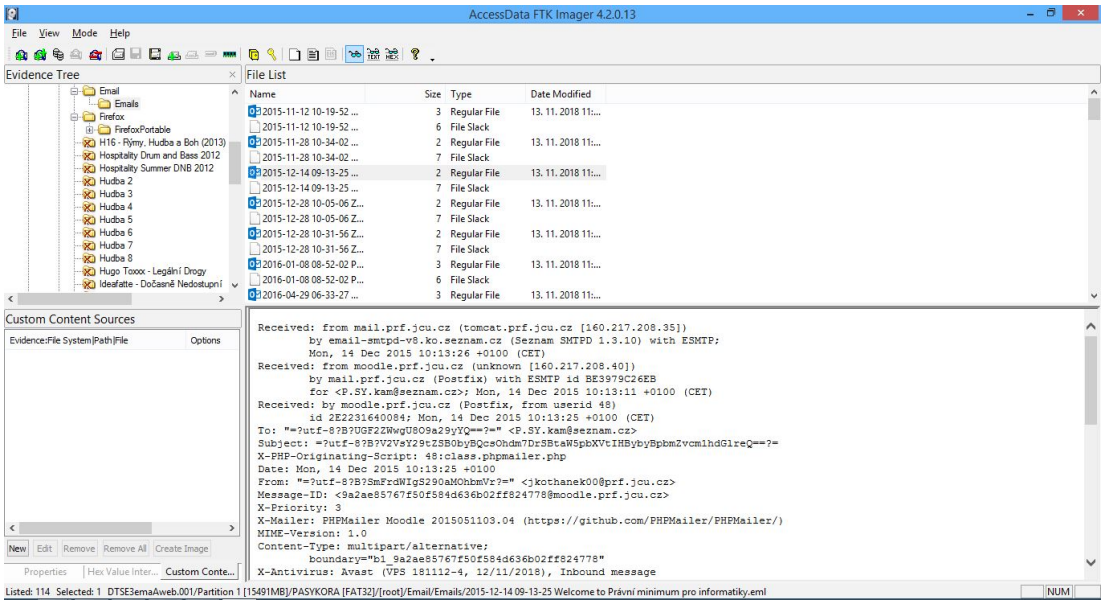
Tyto důkazy jsou získávány za běhu zdrojového přístroje, jedná se o disky nebo jiné přístroje, které jsou připojeny přímo k počítači vyšetřovatele. Veškeré propojování zařízení by mělo být provedeno za pomoci blokátoru[1].

Tuto možnost je třeba řádně zvážit, neboť s sebou nese velká rizika. Ovšem v některých případech nelze jinak. Například pokud je zkoumaný disk zašifrovaný, nebo pokud zkoumáme RAID pole, je nutné získat obraz disku ihned na místě zajišťování.

3.2.3 Remote evidence (Vzdálené důkazy)

Je možné získat živé důkazy z aktivních síťových počítačů včetně informací z paměti RAM. Tento způsob je označovaný jako vzdálený, protože není uložen v počítači vyšetřujícího, ale je přístupný skrze síť.

3.3 FTK Imager



Obrázek 3.1: Ukázka FTK Imager.

Imager je software pro vytváření forenzních obrazů zkoumaných médií. Vytváří identickou bitovou kopii zkoumaného média včetně file slacku[2], alokované a volné paměti. Imager umožňuje zvolit formát, kompresní úroveň a velikost datových segmentů.

Současně s Imagerem se doporučuje použít blokátor zápisu, protože některé operační systémy, jako je například Windows, zapisují na paměťové médium i pouze při čtení.

Je možné zpracovávat statické důkazy a získávat data z lokálních síťových zařízení a zobrazovat náhledy důkazů na vzdálených discích včetně CD a DVD.

Imager umožňuje rychlé zkoumání zajištěných dat, ovšem pro důkladnější analýzu je zapotřebí použít Forensic Toolkit.

[1] Blokátor = zařízení, které umožňuje připojení paměťového zařízení bez jakéhokoliv zápisu na připojené medium

[2] File slack = rozdíl mezi fyzickou a logickou velikostí souboru

3.4 Zkoumání digitálních důkazů

Zkoumání důkazů je proces vyhledávání a identifikace smysluplných dat pro vyšetřování a následně jejich shrnutí do přehledné a srozumitelné formy pro příslušné orgány.

Prvním krokem zkoumání je vytvoření případu (case). Vytvoření zahrnuje zadání obecných informací (vlastník případu, název případu, odkazy, popis případu), umístění složky případu, umístění databáze k případu a zvolení procesního profilu (Processing Profile).

Druhým krokem je vybrání všech zdrojů zkoumaných dat - to může zahrnovat obrazy pevných disků, CD nebo DVD jednotek, USB Flash disků, případně živé důkazy z jiných elektronických přístrojů. Dále pojmenování a popsání důkazů, možnost zařazení nebo vytvoření skupiny důkazů, zvolení časového pásma.

3.4.1 Processing Profiles

Procesní profily jsou uložené seznamy možností zpracování zkoumaných dat. FTK obsahuje pět předdefinovaných profilů (*Forensic procesing, eDiscovery prosessing, Summation processing, Basic assessment, Field mode*). Tyto profily slouží spíše jako předlohy pro vytvoření vlastního profilu pro případ. Vytvořený profil je možné uložit a použít na jakýkoliv další případ.

Mezi možnosti zpracování důkazů například patří:

- volba hashování
- rozbalení složených souborů (Zip, Rar, přílohy emailů)
- vytvoření náhledu grafických souborů
- zapnutí indexování (rychlejší vyhledávání, slovník pro lámání hesel)
- zahrnutí smazaných souborů
- vygenerování HTML nebo CSV souboru obsahujícího seznam všech souborů
- rozpoznání jazyka psaného textu
- Document Content Analysis DCA (shlukuje všechny dokumenty pro rychlejší přehled)

3.5 Hashování

Hashování je proces umožnující vytvoření jedinečné hodnoty ze souboru nebo souborů za použití speciálního algoritmu. Slouží k ověření integrity souborů, identifikaci duplicitních souborů a známých souborů.

Ověření integrity souborů je při dokazování trestné činnosti klíčové. Každý nalezený soubor, který chceme při dokazování použít musí mít vytvořený svojí hash. Pokud je nutné dokázat integritu předloženého souboru, musíme vypočítat jeho hash a porovnat jí s hashí původního souboru a tyto dvě hashe se musí shodovat, aby se prokázalo neporušení důkazu. Lze vytvářet i hashe obrazů celého disku a ověřovat tím integritu při manipulaci s důkazy během vyšetřování.

Je možné použít jednu ze tří hashovacích funkcí:

1. Message Digest 5 (MD5) - jde o algoritmus, který vezme libovolný vstup a vytvoří z něj 128 Bitový „otisk“. Předpokládá se, že je výpočetně nemožné, aby jiný vstup měl stejný „otisk“.[3]
2. Secure Hash Algorithms 1 (SHA-1) - vychází z principů použitých v MD5 a vytváří „otisk“ v délce 160 Bitů.
3. Secure Hash Algorithms 256 (SHA-256) - je stejný jako SHA-1, avšak číslo v názvu značí že vytváří „otisk“ v délce 256 Bitů.

Běžně jsou soubory hashovány a současně indexovány při přidávání do případu a poté jsou porovnány s nějakou známou databází hashů jako je například KFF.

3.6 Known File Filter

KFF neboli filtr známých souborů je nástroj obsažený v FTK, který porovnává hashe souborů se svojí databází známých souborů. Známé soubory mohou být standardní systémové soubory, které lze při vyšetřování ignorovat, nebo naopak soubory, o kterých je známo, že mohou obsahovat nelegální nebo nebezpečný materiál. Díky KFF je možné tyto soubory rychle vyloučit anebo vyhledávat přímo v nich.

3.7 Indexace

Indexace je proces, který každému slovu, řetězci nebo znaku přiřadí index, a vytvoří se tak seznam těchto informací. Tento seznam se používá k rychlému vyhledávání za pomocí indexů.

Nastavení indexace je možné si přizpůsobit při vytváření případu.

Nastavení obsahuje:

- možnost zvolení znaků, které je nutno indexovat a znaky, které indexovat netřeba
- seznam slov, které indexace ignoruje (například spojky a předložky)
- seznam znaků braných jako pomlčky, mezery nebo řídící znaky
- zvolení maximální délky slova pro indexování
- možnost indexování binárních souborů
- rozpoznávání datových údajů

3.8 Vyhledávání

Vyhledávat v datech lze pomocí indexového vyhledávání (Index Search) nebo živého vyhledávání (Live Search).

Index search při vyhledávání prochází již vytvořené tabulky indexů a velmi rychle vrací nalezené informace. FTK pro indexové vyhledávání používá nástroj dtSearch, který dokáže rychle projít terabyty dat. Ovšem v indexech se nemusí nacházet veškeré informace, a proto je občas nutné použít živé vyhledávání.

Živé vyhledávání prochází prohledávaná data bit po bitu, tzn. mnohem větší časovou náročnost oproti indexovému vyhledávání. Ovšem živé vyhledávání umožňuje vyhledávání nealfanumerických znaků, vzorů, regulárních výrazů a hexadecimálních hodnot.

3.9 Bookmarks

Bookmarks neboli záložky je způsob, kterým nalezená data evidovat. Je možné je rychle najít, odkazovat se na ně, přidávat k nim další nalezená data a soubory i takové, které nejsou přímo zpracovávány v daném případu (tato data se označují jako doplňková).

Vytváření záložky:

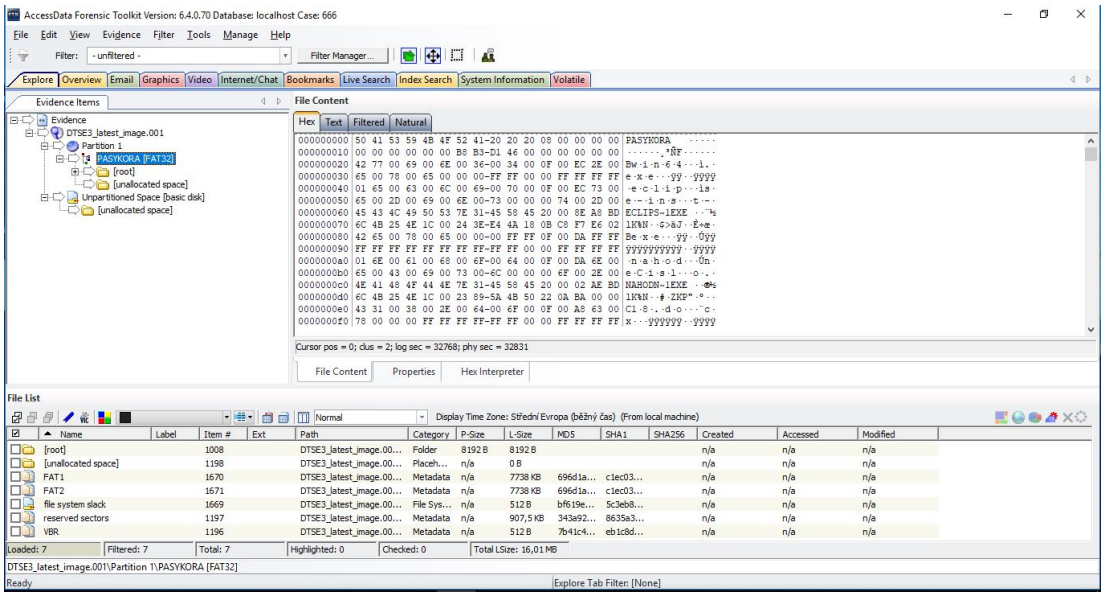
- pojmenování záložky
- popis záložky
- možnost přidání souborů (všechny zvýrazněné, všechny označené zaškrtávacím políčkem, všechny zobrazené nebo žádné)
- komentáře souborů
- doplňková data
- časová osa
- možnost zahrnutí (rodičovský soubor index.DAT, přílohy e-mailů, zdrojový e-mail pokud vybraný soubor je e-mailová příloha, aktuální zdrojový soubor)
- výběr rodiče záložky - Shared přístupné všem vyšetřovatelům anebo záložky pouze pro přihlášeného uživatele (jednotlivé záložky je možné i vnořovat, a tak vytvářet více úrovní)

Pro vytvoření záložky je povinný pouze název a vybrání rodičovské záložky.

Skrze záložky se exportují data do výsledného reportu případu.

3.10 Examiner

Examiner je hlavní pracovní prostředí FTK, které se spustí po vytvoření/zvolení případu. Examiner umožňuje vyšetřovateli vybrat si z mnoha způsobů prohledávání dat za pomoci vybrané karty.



Obrázek 3.2: Ukázka FTK Examiner.

Výčet karet:

Explore karta slouží k procházení všech dat pomocí stromové struktury odpovídající struktuře uložení originálních dat.

Overview karta zjednodušuje vyhledávání díky rozdělení dat do skupin:

1. Evidence group - rozdělení podle skupiny důkazů
2. File items - podle položek (položky důkazů, označené a neoznačené položky pomocí zaškrtávacího políčka)
3. File extension - rozdělení dat podle koncovek souborů (exe, doc, eml, pdf, txt atd.)
4. File Category - rozdělení podle kategorie souboru (dokumenty, e-maily, grafické, spustitelné atd.)
5. File Status (špatná koncovka, zašifrovaná data, e-mailové přílohy, smazaná data atd.)
6. Email Status (e-mailové přílohy, všechny související soubory s e-maily, e-mailové odpovědi, přesměrované e-maily)
7. Labels - data je možné si „oštítkovat“ a vytvářet skupinu, která pak tato data zobrazuje
8. Bookmarks - záložky
9. Cluster topic - pracuje s daty vytvořenými přes Document Content Analysis
10. Document Content - čísla kreditních karet, e-mailové adresy, telefonní čísla, číslo sociálního zabezpečení (pouze v USA)

Pro přehlednost každá skupina a její položky obsahují přesný počet vyčtených souborů.

Email karta je uzpůsobená pro prohledávání e-mailových dat. Dokáže rozřadit e-maily podle příloh, odpovědí, přesměrování, data odeslaní a přijetí, adres odesílatelů a příjemců.

Graphics karta pro práci s grafickými soubory, umožňuje rychle procházet grafická data a zobrazovat náhledy.

Video karta pro práci s multimédii, jako jsou například videa nebo hudba.

Internet/Chat karta umožnující procházet soubory internetových prohlížečů, jako je historie, záložky nebo cookies.

Bookmarks karta slouží pro přehled a úpravu vytvořených záložek.

Live Search karta pro živé vyhledávání.

Index Search karta pro indexové vyhledávání.

System information karta zobrazující detailní informace o operačním systému daného obrazu disku (pokud se jedná o paměťové médium disponující nějakým operačním systémem). Je možné najít informace o nainstalovaných aplikacích, síťové informace, informace o vlastníku nebo připojovaná USB zařízení.

Volatile karta poskytuje nástroj pro zobrazování, hledání a porovnávání dat shromážděných pomocí live agent systému v síti. Jde především o data získaná jako memory dump (zaznamenaná data v případě zhroucení aplikace nebo systému).

3.11 Vytvoření reportu

Report je způsob, kterým lze prezentovat nalezená data, jenž mají nějaký význam pro vyšetřovaní. Report lze tvořit kdykoliv v průběhu analýzy dat anebo po dokončení celého zkoumání.

Možnosti reportu:

Case Information obsahuje informace o případu (agentura, vyšetřovatel, adresa, telefon, fax, e-mail, komentář).

Bookmarks obsahuje možnosti vybrání záložek, které se exportují do reportu, dále také možnost u jednotlivých záložek zahrnout e-mailové přílohy, exportovat soubory a připojit k nim odkazy.

Pomocí volby *Columns* neboli sloupce se otevře další dialogové okno, kde si můžeme vybrat jednu z předdefinovaných šablon, které určují jaké bude tabulka souboru v reportu obsahovat informace. Na výběr je mnoho specifických šablon, například šablony zaměřené na e-maily, chatové konverzace, internetová historie, mobilní zařízení a mnoho dalších. Je možné i vytvoření vlastní šablony.

Ve výchozím nastavení je zvolena šablona *Standard*, která obsahuje:

Tabulka 3.1: Šablona standard.

File Comments	Komentář k souboru
Name	Název souboru
Physical Size	Fyzická velikost
Logical Size	Logická velikost
Created date	Datum vytvoření
Modified Date	Datum pozměnění souboru
Accessed Date	Poslední přístup k souboru
Path	Cesta k souboru ve složce reportu

Avšak to je dost omezené množství informací, proto je dobré použít alespoň šablonu *Normal*, která navíc obsahuje informace:

Tabulka 3.2: Doplnění šablony standard.

Label	Štítek souboru
Item Number	Číslo přiřazené souboru při vytváření případu
Extension	Koncovka souboru
File Type	Typ souboru
MD5 Hash	
SHA1 Hash	
SHA256 Hash	

Graphics umožňuje zahrnutí sekce s grafickými soubory do reportu a možnost exportovat k náhledům grafických souborů i verze původních souborů v plné velikosti (pouze označené nebo všechny v případu).

Videos umožňuje zahrnutí sekce s multimediálními soubory do reportu, dále možnost exportování náhledů anebo vyrenderované MP4 soubory původních dat. Je také možné přiložit odkaz k původní verzi souboru v plné velikosti.

File Paths umožňuje zahrnout do reportu cesty umístění souborů ve zvolených kategoriích.

File Properties umožňuje zahrnout do reportu vlastnosti souborů ve zvolených kategoriích.

Registry Selection umožňuje zahrnutí souborů registrů, pokud jsou obsaženy v obrazu disku.

Screen Capture umožňuje zahrnout do reportu snímky obrazovky pořízené během analýzy dat.

Dalším krokem k vytvoření reportu je zvolení umístění, kam se má report uložit, zvolení jazyka reportu a časového pásma. Report může být vytvořen v 7 možných formátech (PDF, HTML, XML, RTF, WML, DOCX, ODT). K HTML formátu je možné nahrát vlastní logo a vlastní CSS styly. Je možné zaškrtnout volbu pro nahrazení názvu souborů identifikačními čísly pro zkrácení cest v reportu a volbu pro opravení špatných nebo chybějících koncovek souborů. [2]

4. Analýza reportu

FTK vytváří všechny formáty reportu pomocí souboru **Report.fo**. Jedná se o XML soubor, který se vytvoří při generování reportu a poté se konvertuje do potřebného formátu. Jedinou výjimkou je HTML report, který se z ničeho nekonvertuje, ale generuje se rovnou.

Reporty se ukládají vždy do své vlastní složky nehledě na formátu a tato složka obsahuje i podsložku Report_Files, ve které se nachází všechny přiložené soubory.

4.1 Textové reporty

Reporty v textových formátech mají stejnou strukturu a formátovaní, liší se pouze ve způsobu zobrazení. Slouží především k vytváření tištěné podoby reportu, která se používá jako dokumentace k případu.

DOCX, **ODT**, **RTF** jsou zobrazitelné v textových editorech.

PDF report lze zobrazit v prohlížeči PDF souborů. Z hlediska přenositelnosti je tento formát nejvhodnějším řešením.

4.2 XML Report

XML (Extensible Markup Language) je značkovací jazyk obsahující sadu pravidel pro definování sémantických značek, které rozkládají dokument do částí a identifikují různé části dokumentu. Je to meta-značkovací jazyk, který definuje syntaxi použitou k definování jiných značkovacích jazyků. Je uzpůsoben pro popisování dat, nikoliv pro zobrazování. Dovoluje uživateli podle daných pravidel vytvářet své vlastní značky, které nejlépe popisují používaná data.[3]

Report ve formátu XML se po vytvoření uloží do složky reportu do jednoho souboru *Report.xml* a je zobrazitelný ve webovém prohlížeči.

Listing 4.1: Ukázka jednoho záznamu v XML jazyce.

```
<fo:table-row>
 <fo:table-cell font-size="120%" font-weight="bold"
    padding-right="1pt" padding-left="1.5pt">
  <fo:block>Name</fo:block>
 </fo:table-cell>
 <fo:table-cell padding-left="3pt">
  <fo:block>1a.png</fo:block>
 </fo:table-cell>
</fo:table-row>
```

Na ukázce kódu 4.1 je znázorněn způsob, jak jsou data uložená v XML formátu. Data jsou ukládána do tabulek a konkrétně v této ukázce jde o záznam jednoho řádku (table-row) v tabulce, který obsahuje dvě buňky (table-cell) a každá buňka má svůj obsah (table-block), ve kterém jsou vypsané informace. Jak již bylo řečeno, reporty se generují ze souboru **Report.fo**, který je ale také napsán v XML jazyce, a tudíž se jedná

o totožnou kopii pouze s jinou příponou souboru. Kvůli tomu XML report obsahuje i údaje o formátování záznamu, které se využívají při konvertování do jiného formátu.

WML je stejný jako XML, ale jedná se o verzi pro Unixové systémy.

4.3 HTML Report

HTML report je zobrazitelný ve webovém prohlížeči a jako jediný report má i funkční prvky. Informace v něm jsou rozděleny pomocí záložek, které lze procházet. Slouží hlavně pro distribuování dokumentace elektronickou formou.

HTML (Hyper Text Markup Language) neboli hypertextový značkovací jazyk je stejně jako XML jazyk využívající značky, ale oproti XML jsou pevně definovány a není možné si vytvářet vlastní. HTML slouží hlavně pro zobrazování dat, a to především na webových stránkách.

HTML report je jako jediný rozdělen do více souborů a přistupuje se k němu skrze soubor *index.html* ve složce reportu. Ostatní soubory jsou uložené ve složce Report_Files. Záložky jsou rozdělené do souborů Bookmark_bk_IDx(y), kde x značí číslo záložky, a pokud záložka obsahuje více stran, má každá stránka svůj soubor a jsou očíslovány pomocí čísel v závorce poznačených jako y. Informace o případu jsou uložené v souboru *CaseInfo*, informace o zdroji důkazů jsou uložené v *EvidenceList* a výpis obsažených souborů je uložen ve *FileOverview*. Jestliže report obsahuje jakékoliv další informace, nalezneme je všechny ve svém vlastním souboru, pokud obsahují více stránek, jsou rozděleny stejně tak, jako je to provedeno u záložek. Procházení mezi všemi záložkami a ostatními informacemi je provedeno pomocí odkazů.

Listing 4.2: Ukázka jednoho záznamu v HTML jazyce.

```
<div class="row">
  <span class="bkmkColLeft bkmkValue labelBorderless clrBkgrnd"
    width="100%" border="1">Name
  </span>
  <span class="bkmkColRight bkmkValue">1a.png
  </span>
</div>
```

Na ukázce kódu 4.2 je znázorněn způsob, jak jsou data uložená v HTML formátu. Jednotlivá data jsou rozdělena do bloků (div) a každý blok obsahuje dva řádkové elementy (span), kde jsou vypsané informace.

File Comments:	
Name	1a.png
Physical Size	1581056 B
Logical Size	1580418 B
Created Date	12. 11. 2017 23:45:28 (2017-11-12 22:45:28 UTC)
Modified Date	15. 10. 2017 23:27:10 (2017-10-15 21:27:10 UTC)
Accessed Date	12. 11. 2017
Path	DTSE3emaAweb.001/Partition 1/PASYKORA [FAT32]/[root]/1a.png

Obrázek 4.1: Ukázka zobrazení HTML kodu z ukázky kódu v 4.2.

Obrázek 4.1 znázorňuje, jak vypadá celý záznam jednoho souboru a nachází se v něm i zobrazení kódu z ukázky kódu 4.2 a to konkrétně druhý řádek **Name** s hodnotou *1a.png*.

4.4 Porovnání reportů

V níže zobrazené tabulce je znázorněno porovnání časové náročnosti vytváření jednotlivých reportů a konečná velikost celého adresáře s hotovým reportem. Jedná se o reporty s naprosto totožnými vlastnostmi vytvořené na stejném hardwaru (Notebook Lenovo, Intel Core i3 2.30Ghz, 4Gb RAM).

Na **USB Flash disk** byl použit procesní profil *Forensic proccesing* a k reportu byly vyexportovány všechny nalezené soubory. Report obsahoval celkem `386` souborů.

Tabulka 4.1: Zkoumání flash disku.

Formát	Doba vytváření	Velikost reportu	Rozsah reportu
XML	1 minuta 12 sekund	435 MB	1 soubor
HTML	1 minuta 20 sekund	434 MB	31 souborů
PDF	1 minuta 22 sekund	435 MB	97 stran
DOCX	54 sekund	434 MB	89 stran

Na **Pevný disk** byl použit procesní profil *Forensic proccesing* a k reportu nebyly exportovány žádné soubory. Report obsahoval celkem `670 030` souborů.

Tabulka 4.2: Zkoumání pevného disku.

Formát	Doba vytváření	Velikost reportu	Rozsah reportu
XML	32 hodin 39 minut	6,4 GB	1 soubor
HTML	33 hodin 17 minut	2,17 GB	26 814 souborů
PDF	35 hodin 53 minut	4,61 GB	
DOCX	32 hodin 44 minut	3,36 GB	

Všechny reporty mají jednu společnou vlastnost, jsou postavené na velmi malé množství dat. Informace v nich jsou nejpřehlednější při desítkách záznamů, pokud už to jsou stovky záznamů, začínají být reporty dosti obsáhle a hůře se v nich hledají informace. A při několika tisíci a více záznamů se reporty stávají naprosto nepřehledné a nepraktické.

Z tabulky 4.1 lze vypozorovat, že u reportů s pár set záznamy se doby vytváření a velikost reportu nijak razantně neliší. U reportů z tabulky 4.2 s velkým obsahem dat se začínají lišit velikosti reportů a rozdíl v době vytváření reportu se zdá zanedbatelný. Zásadní je rozsah reportu, například u textových formátů jako je PDF, nebo DOCX u zkoumání Flash disku s 386 soubory jde o dokumenty se skoro sto stranami, ve kterých je velice náročné hledat a vyhodnocovat informace. U zkoumání pevného disku je DOCX report už tak velký, že Microsoft Word není schopen soubor zobrazit a PDF report s takovým množstvím dat se Forensic Toolkitu nepodařilo ani konvertovat ze souboru Report.fo. Z tohoto důvodu je časový údaj u PDF reportu v tabulce 4.2 velmi totožný s ostatními, protože konvertování selhalo a tím pádem není ve výsledném čase započítáno.

Zbylé dva formáty XML a HTML jsou schopné zpracovat jak malé reporty, tak i velké. Avšak zásadní nevýhoda XML spočívá v tom, že vše je uloženo v jednom souboru. U malých reportů to není problém, ovšem pokud report obsahuje několik set tisíc záznamů, stává se report extrémně nepřehledným a pomalým. Oproti tomu HTML veškeré informace rozděluje do různých souborů, které jsou velké maximálně pár set kilobitu.

Nevýhoda tohoto způsobu spočívá v tom, že například u reportu pevného disku zmíněného v tabulce se jedná o více jak 26 tisíc souborů, avšak soubory jsou systematicky pojmenovány a jsou propojeny odkazy.

Z důvodu nepřehlednosti reportů je úkolem této práce vytvoření aplikace, která usnadní procházení a vyhodnocování obsažených informací, a to bez nutnosti disponování samotným Forensic Toolkitem. Na základě výše uvedených informací bylo rozhodnuto, že vytvořená aplikace bude pracovat s HTML reporty.

5. Návrh aplikace zpracovávající HTML report

Navrhovaná aplikace by měla být schopna uživateli umožňovat dynamicky procházet a vyhledávat důležitá data pro vyšetřování. Měla by disponovat funkcemi, které toto vyhledávání usnadní a zrychlí. A nakonec by měla být schopna nalezená data přehledně prezentovat. Aplikace dostala pracovní název **Report Enhancer**.

Jakožto nadstavba pracuje aplikace s výstupem z FTK, bez těchto dat je aplikace nefunkční. FTK nabízí obrovské množství různých informací k různým souborům, které lze zahrnout do výstupního reportu, avšak aplikace musí dopředu vědět, jaké informace vyhledávat. Proto byl po konzultaci s vedoucím práce vytvořen seznam informací, které musí aplikace zvládat. Seznam je založen na zkušenostech z praxe a skládá se ze základních informacích o souborech + metadata dokumentů + všechny informace o e-mailových souborech.

Výčet informací:

Tabulka 5.1: Tabulka s výčtem informací.

Name	SHA256 Hash	Hidden Columns or Rows	CC
Label	Created date	Hidden Worksheets	BCC
Item Number	Accessed date	Last Printed	Submit Time
Extension	Modified date	Last Saved Time	Delivery Time
Path	Carved	Revision Number	Unread
Category	Deleted	Total Editing Time	Unsent
Physical Size	Author	Track Changes	Has Attachment
Logical Size	Last Saved By	Subject	Email Priority
MD5 Hash	Create Time	To	Email Account
SHA1 Hash	Embedded Comments	From	Src

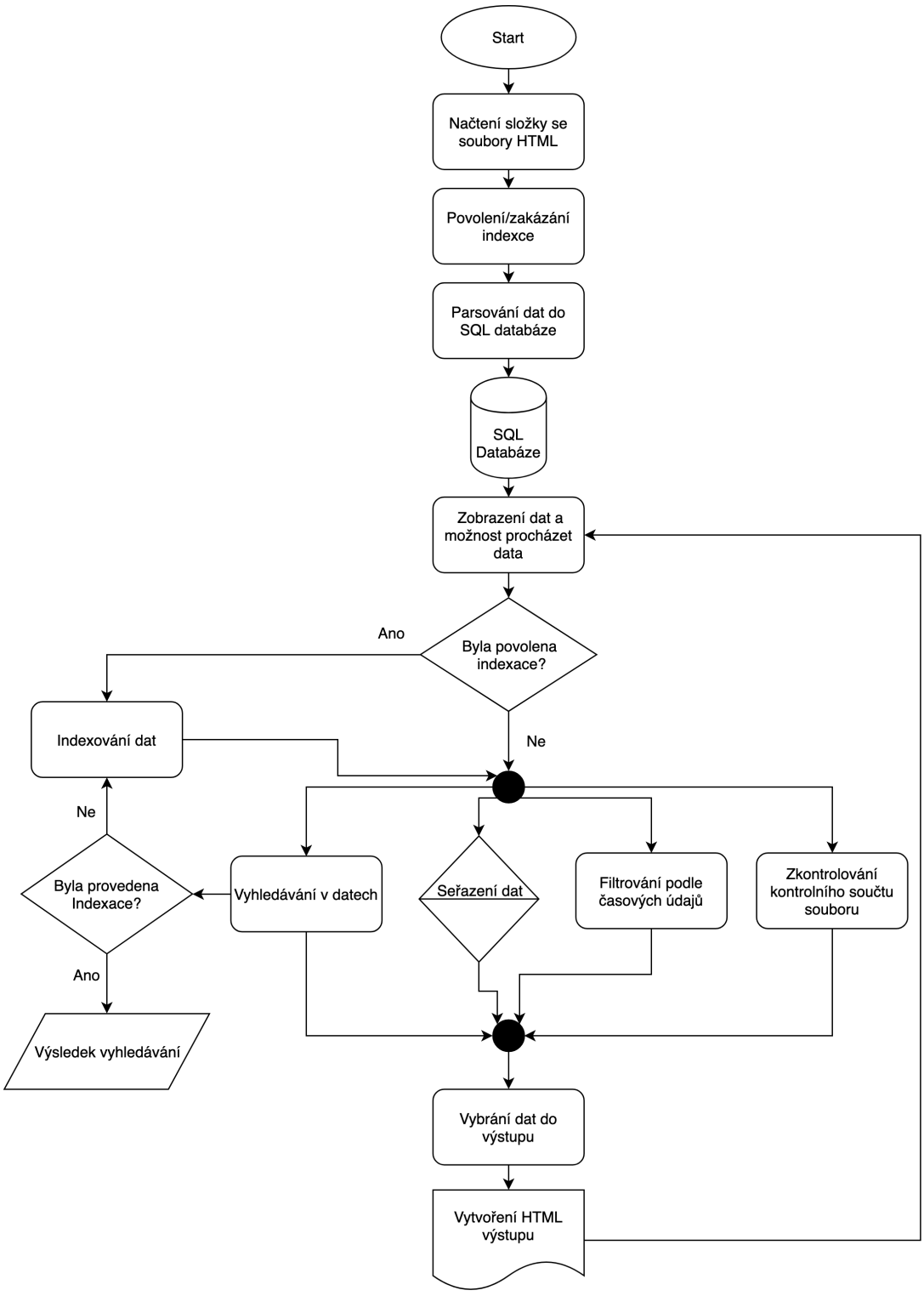
5.1 Funkce aplikace

- **Načtení reportu** - prvním krokem po spuštění aplikace je načtení potřebných dat, konkrétně složky Report_Files, kde se nachází soubory HTML reportu. Aby bylo možné s těmito daty dále pracovat, je zapotřebí načíst je do databáze (v tomto případě do SQLite databáze). Načítání dat do databáze musí projít všechny soubory Bookmark_bk_IDx(y) a informace v nich uložit do tabulky. Jelikož všechny záložky v reportu mají stejnou strukturu, lze ze všech dat vytvořit jednu tabulku. Dále dojde k načtení dat z *CaseInfo*, *EvidenceList*, *FileOverview*, které obsahují informace o důkazech a popis případu. Pokud budou součástí reportu i vyexportované soubory, musí být aplikace schopná s nimi pracovat.
- **Procházení načtených dat** - součástí aplikace musí být prostředí, které načte data z databáze do přehledné tabulky (jeden řádek tabulky = jeden konkrétní soubor z reportu) a umožní tyto data prohlížet a procházet. Data budou rozdělena do tabulek podle záložek, ale zároveň bude možné data třídit podle typu souboru (podle přípon souboru). Aplikace musí umožnit zobrazení dat, a to způsobem, že soubory se budou otevírat v programech, se kterými jsou asociovány na

daném zařízení (například soubory typu `jpg, png` se budou otevírat v prohlížeči obrázků).

- **Řazení dat** - aplikace musí umožňovat řazení načtených dat (vzestupně nebo sestupně vybraný sloupec tabulky), například podle názvu souboru, nebo velikosti.
- **Vyhledávání v datech** - aplikace musí umožnit dva druhy **vyhledávání**, první je vyhledávání v názvech souborů, druhé je vyhledávání jednoduchých textových řetězců uvnitř textových souborů. Pro vyhledávání v textových souborech je zapotřebí nejprve obsah těchto souborů indexovat. Možnost soubory indexovat je nabídnuta hned po načtení HTML reportu, avšak indexace je časově náročný proces, a proto ji lze při načítání dat vynechat a zavolat ji až v případě potřeby.
- **Filtrování dat podle časových údajů** - aplikace musí umožňovat filtrovat zobrazená data podle časových údajů, například zobrazit pouze soubory vytvořené od konkrétního data. Tento filtr lze použít pouze na sloupec tabulky, který obsahuje časové údaje.
- **Vypočítávání kontrolní sumy** - aplikace musí být schopna vypočítat kontrolní sumu (MD5 nebo SHA1) z přiloženého souboru u reportu a porovnat jí s hodnotou uvedenou v reportu a tím zkontrolovat integritu souboru.
- **Vytváření výstupu** - aplikace musí umožňovat označovat si důležité soubory, které chceme dále prezentovat a vytvořit z nich přehledný výstup. Tento výstup musí být ošetřen kontrolním součtem, aby bylo možné kontrolovat jeho integritu.

Na obrázku 5.1 na další straně je znázorněn návrh programu pomocí vývojového diagramu.



Obrázek 5.1: Vývojový diagram aplikace.

5.2 Použité technologie

Programovací jazyk

Program je napsán v Jazyce JAVA a grafická část pomocí frameworku JavaFX. Bylo použito starší JDK ve verzi 8, protože novější verze již neobsahují JavaFX.

Java je objektově orientovaný jazyk vyvinutý společností Sun Microsystems. Jeho největší předností je přenositelnost mezi různými operačními systémy (Windows, Mac OS, Linux) a různými platformami (počítače, telefony, tablety). Toho je docíleno tím, že Java běží na virtuálním stroji Java Virtual Machine (JVM). Zdrojový kód je pomocí kompiléru převeden do bytecodu a až ten je poté převáděn interpreterem v JVM na strojový kód daného zařízení, na kterém je program spuštěn.[5]

JavaFX je grafická uživatelská rozhraní (GUI) nové generace Java, která umožňuje rychle vytvářet aplikace. JavaFX je vybudován od základů a využívá moderních grafických procesorů pomocí hardwarově akcelerované grafiky a poskytuje dobře navržené programovací rozhraní umožňující vývojářům kombinovat grafické, animační a uživatelské ovládací prvky. Nový program JavaFX 8 je čisté jazykové rozhraní pro programování aplikací jazyka Java (API).[6]

Vývojové prostředí

Jako vývojové prostředí bylo použito IntelliJ IDEA od společnosti Jet Brains ve verzi 2018.2.5.

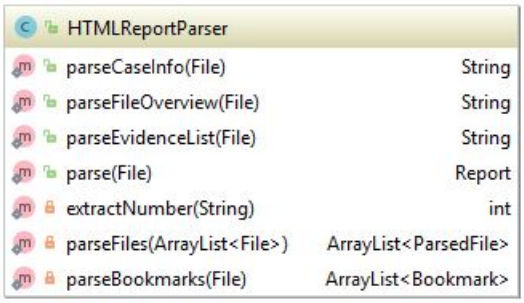
Databázový systém

Jako databázový systém bylo použito **SQLite**. Jedná se o databázový systém obsažený v malé knihovně, který nemá samostatný serverový proces, ale čte a ukládá data přímo z disku. Díky tomu je možné ho použít i u portable (přenositelné) aplikace. Umožňuje velikost databáze až 140 terabytů.[7]

Hlavním důvodem použití SQLite je soběstačnost a open-source licence.

6. Implementace aplikace zpracovávající HTML report

6.1 Třída pro zpracování HTML



Obrázek 6.1: Zjednodušený diagram třídy.

Jak již bylo zmíněno v návrhu, aplikace začíná načítáním dat z HTML. Jako první probíhá uložení dat z *CaseInfo*, *EvidenceList* a *FileOverview*.

Listing 6.1: Ukázka načtení informací o případu do objektu.

```
public static String parseCaseInfo(File dir) throws IOException {

    final File caseInfo =
     new File(dir.getPath() + File.separator + CASE_INFO_FILE);

    String caseInfoContent =
     new String(Files.readAllBytes(Paths.get(caseInfo.getPath())),
         StandardCharsets.UTF_8);

    Document caseInfoDoc = Jsoup.parse(caseInfoContent);

    caseInfoDoc.select("img").remove();

    return caseInfoDoc.toString();
}
```

Na ukázce 6.1 je znázorněna metoda `parseCaseInfo`, která načítá *CaseInfo* do objektu File. Metodě `parseCaseInfo` se předává cesta ke složce se soubory. Do Stringu `caseInfoContent` se uloží obsah HTML souboru a ten je poté pomocí `Jsoup.parse()` parsován. Pomocí `caseInfoDoc.select("img").remove();` jsou odstraněny obrázky v HTML souboru, které slouží pouze jako designový prvek. Metoda vrací String `caseInfoDoc`.

Jsoup je Javovská knihovna pro práci s HTML, která poskytuje rozhraní pro extrahování a manipulaci s daty.[8]

Soubory *EvidenceList* a *FileOverview* jsou zpracovány stejným způsobem.

Dalším krokem je načtení souboru Bookmarks.html, který obsahuje odkazy na všechny záložky v reportu. O práci s tímto souborem se stará metoda parseBookmarks(). Na ukázce 6.2 je zobrazen blok s odkazem na první stránku záložky *Documents*.

Listing 6.2: Odkaz na první stránku záložky HTML.

```
<tr><td>
      <a href="Bookmark_bk_ID8001(1).html">
            <span style="margin-left:30px">Documents
            </span></a></td><td></td>
</tr>
```

Soubor Bookmarks.html je parsován stejně jako předchozí soubory pomocí knihovny Jsoup. V ukázce 6.3 je kód, který z ukázky 6.2 extrahuje samotný odkaz na soubor, tedy Bookmark_bk_ID8001(1).html.

Listing 6.3: Kód pro extrakci odkazu.

```
Elements rows = doc.select("body > table:nth-child(5) tr td a");
```

Pokud záložka obsahuje větší množství dat, je rozdělena na více stran. V takovém případě obsahuje odkaz na další a poslední stránku. Aplikace kontroluje, jestli má záložka více stran a pokud ano, je nutné extrahovat odkaz na poslední stránku stejným způsobem jako v předchozím příkladu. Pomocí odkazu na poslední stránku aplikace zjistí, na kolik souborů je záložka rozdělena.

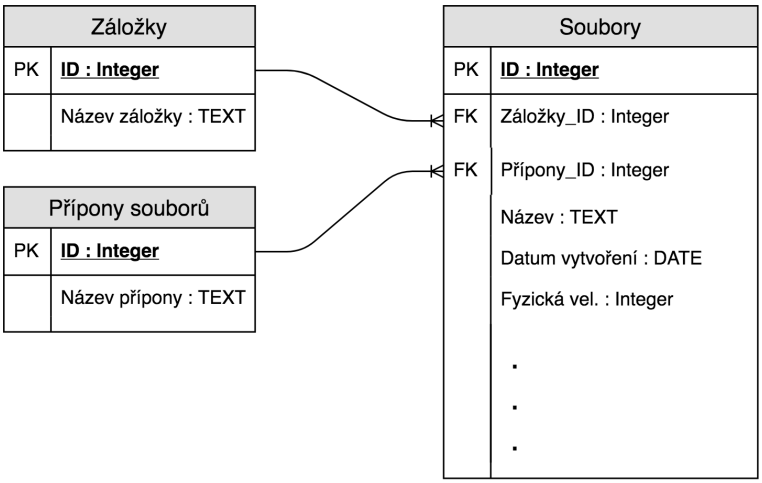
Podmínka na ukázce 6.4 projde, pokud záložka má více jak jednu stránku. Pomocí regulárních výrazů byl vytvořen vzorec, který z názvu souboru získá číslo označující počet souborů záložky. Například z Bookmark_bk_ID4001(8) získá číslo 8.

Listing 6.4: Kód pro extrakci čísla z názvu souboru.

```
if(lastBookmarkElement != null) {
    Pattern p = Pattern.compile("\\((\\d*?)\\)");
    Matcher m = p.matcher(lastBmkE.attr("href"));
    m.find();
    lastBookmarkIndex = Integer.parseInt(m.group(1));
            }
```

Metoda parseFiles() prochází záložky a vytváří souhrnný seznam (ArrayList) všech řádků ze všech souborů záložky. Tento seznam se pak prochází a hledají se klíčové informace jako „Name“, „Label“, „Extension“, „Path“ a další. Ty se poté ukládají do SQLite databáze.

6.2 Databáze



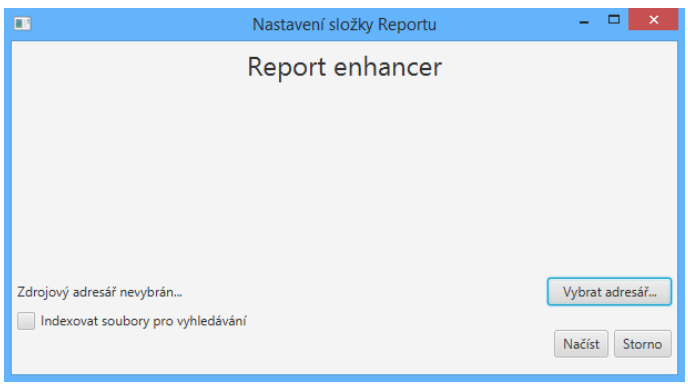
Obrázek 6.2: Diagram databáze.

Na obrázku 6.2 je znázorněn model použité databáze. Databáze obsahuje pouze tři tabulky.

1. `Záložky`, která obsahuje všechny záložky reportu.
2. `Přípony souborů`, která obsahuje koncovky souborů, které jsou získané z posledních 4 znaků názvu souboru. Avšak soubory mohou mít příponu neodpovídající typu souboru, tuto problematiku kontroluje FTK a lze do reportu zahrnout sloupec s odpovídající příponou.
3. `Soubory`, která obsahuje informace o souborech obsažených v reportu. Tabulka je znázorněna neúplná, protože obsahuje 44 sloupců. A to konkrétně všechny z tabulky 5.1 z kapitoly Návrh aplikace.

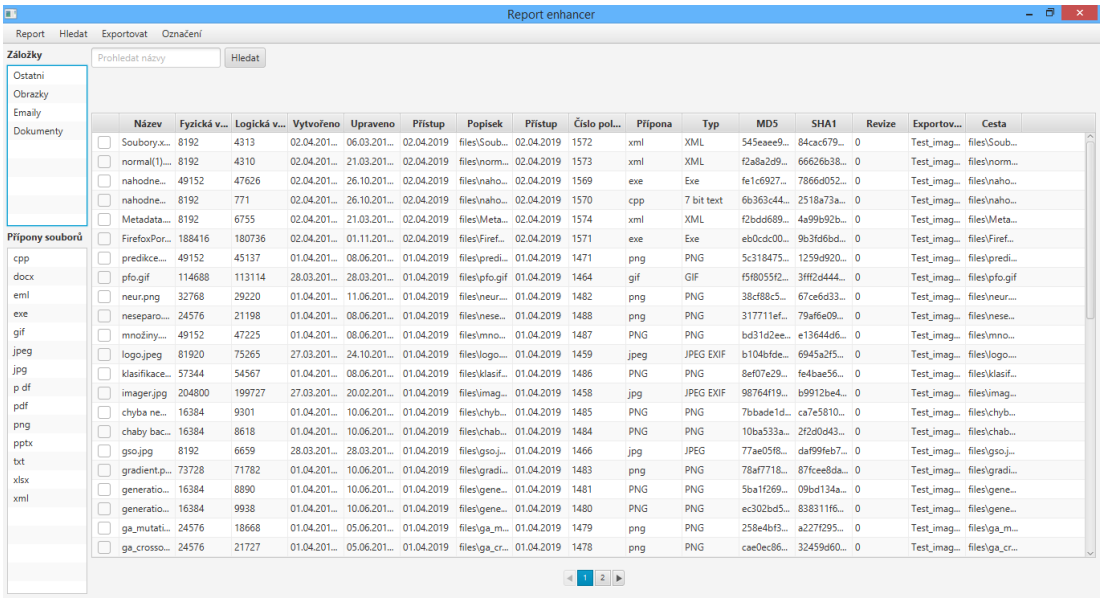
6.3 GUI

Uživatelské rozhraní aplikace se skládá ze dvou oken. První okno po spuštění aplikace se stará o nastavení cesty k adresáři Report_Files, kde jsou uloženy soubory HTML reportu a vyexportované soubory. Obsahuje ještě checkbox, ve kterém můžeme nastavit spuštění indexace hned po načtení dat.



Obrázek 6.3: Ukázka GUI načítání dat.

Druhým oknem aplikace je průzkumník načtených dat. Hlavní částí tohoto okna je tabulka, kde se zobrazují informace o souborech. V levé části se nachází rozdělení dat podle záložek, nebo podle koncovek souborů. Horní lišta obsahuje možnost zobrazení informací o reportu (přehled souborů, informace o případu, informace o zkoumaném médiu), prohledávání textových souborů, vyhledávání podle časového údaje a export vybraných dat.



Obrázek 6.4: Ukázka GUI průzkumníka dat.

Veškeré zobrazování dat v průzkumníku, řazení dat, vyhledávání v názvech souborů a filtrování podle časového údaje je provedeno pomocí SQL dotazů.

Listing 6.5: Příklad SQL.

```
SELECT * FROM FILES WHERE `bookmarks_id` = 3 AND
2  `extensions_id` = 18 ORDER BY created_date ASC
```

Na ukázce 6.5 je příklad jednoduchého SQL dotazu, který zobrazí data ze záložky „Dokumenty“, které mají příponu souboru „txt“ a seřadí je vzestupně podle data vytvoření.

6.4 Kontrolní součet

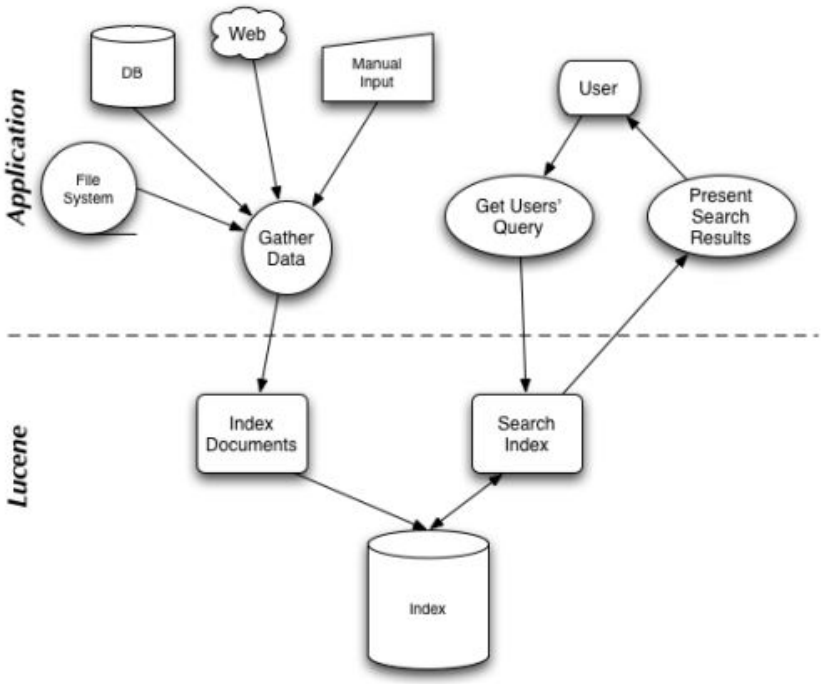
Na ukázce kódu 6.6 je znázorněna metoda starající se o výpočet a porovnání kontrolní sumy. Metodě se předá soubor, který chceme zkontrolovat a jeho hodnota MD5 z tabulky. Aplikace sama ze souboru vypočítá svojí hodnotu MD5 pomocí třídy `DigestUtils`. Nově vypočítaná hodnota se porovná s hodnotou v tabulce, a pokud se hodnoty shodují, tak je prokazatelné, že nebyla porušena integrita souboru. Obdobným způsobem je realizována i SHA1.

Listing 6.6: Kód výpočtu kontrolního součtu.

```
public void compare(String md5, File file){
    md5Label1.setText(md5);
    try (InputStream is =
            Files.newInputStream(Paths.get(file.getPath()))) {
      String countedMD5 =
            org.apache.commons.codec.digest.DigestUtils.md5Hex(is);
      md5Label2.setText(countedMD5);
      if(md5.equals(countedMD5)){
         md5ResultLabel.setText("Součty se shodují");
         md5ResultLabel.setTextFill(Color.web("#4caf50"));
      } else {
         md5ResultLabel.setText("Součty se neshodují");
         md5ResultLabel.setTextFill(Color.web("#f44336"));
      }
    }catch (Exception e){
    e.printStackTrace();
                }
     }
```

6.5 Fulltext vyhledávání

Pro vyhledávání textových řetězců uvnitř textových souborů využívá aplikace **Lucene**. Jedná se o výkonnou, rozšiřitelnou knihovnu pro vyhledávání informací. Umožňuje přidání indexace a vyhledávání do aplikace. Jde o open-source projekt implementovaný v Javě.[9]



Obrázek 6.5: Diagram spolupráce Lucene s aplikací.[9]

Vyhledávat v datech, lze pouze, pokud proběhla indexace a ta je možná provést jedině, pokud jsou k reportu vyexportované soubory. Proces indexace prochází adresář se soubory a vyhledává přijatelné soubory pro indexování. Seznam přijatelných souborů je obsažen ve třídě AcceptableFileFilter a jde o: *html, json, csv, pdf, docx, eml, txt.* Indexy se ukládají do souborů ve vytvořeném adresáři `Indexes`.

Lucene ovšem nedokáže správně vyhledávat v PDF a Microsoft dokumentech, proto bylo nutné ho rozšířit o metody getPDFFile() a getDocxFile(). Pro práci s PDF bylo rozšířeno pomocí knihovny PDFBox, jedná se o open-source Java knihovnu umožňující vytváření, manipulaci a schopnost extrakce informací z PDF dokumentů.[10] Pomocí této knihovny se data z PDF konvertují do textové podoby a až ta se poté indexuje. Stejným způsobem se pracuje s Microsoft dokumenty za pomoci Apache POI, což je rozhraní umožňující zapisování a čtení z dokumentů jako jsou MS Excel, MS Word atd. pomocí Javy.[11]

7. Zhodnocení aplikace zpracovávající HTML report

Aplikace splnila všechny funkční požadavky:

- nahrání HTML report
- umožnění procházení dat
- rozdělení dat podle záložek nebo podle přípon souborů
- řazení dat
- vyhledávání v názvech souborů i uvnitř textových souborů
- filtrování podle časových údajů
- zobrazování souborů jako jsou obrázky a dokumenty
- vytváření výstupu

Avšak v průběhu vývoje aplikace se ukázalo, že tento způsob není ideální. Hlavním problémem je samotná časová náročnost vytváření HTML reportu v FTK. Dále pak složité načítání dat z velkého množství souborů. Dalším problémem samotné aplikace je možnost načítání HTML reportů pouze s jednou úrovní záložek.

Zpracování HTML reportu v aplikaci nadále zůstane, ovšem bude zaměřeno na menší HTML reporty, které není časové náročné vytvořit ani zpracovat. Aplikace ale bude rozšířena o možnost načítání vhodnějších vstupních dat.

V kapitole 4 byly rozebrány jiné formáty reportu, avšak žádný z nich by stávající problém nevyřešil, proto je nutno zvolit jiný způsob exportu dat z FTK.

Pro rozšíření aplikace bylo nalezeno mnohem jednodušší a rychlejší řešení výstupu dat z FTK a to pomocí CSV výstupu.

8. CSV výstup

FTK neumožňuje výstupy pouze v podobě reportů, ale je možné exportovat i `File list informations`. Jedná se o vyexportování seznamu informací vybraných souborů. Zvolení informací k exportování funguje stejně jako u reportů - pomocí zvolení šablony sloupců.

Možnosti vybrání souborů: všechny zvýrazněné, všechny označené pomocí checkboxu, všechny zobrazené a úplně všechny soubory.

Seznam je možné uložit do textového souboru, nebo do CSV souboru. CSV - Comma-separated values neboli „Hodnoty oddělené čárkou“ je soubor, který ukládá údaje do tabulky a jako oddělovač používá čárku. Kvůli své jednoduchosti je vhodný pro načítání dat do databáze. Soubory CSV jsou zobrazitelné například pomocí Microsoft Excel. Ukázka tabulky je znázorněna na obrázku 8.1.

Name	Label	Item #	Ext	Path	Categor
!018-0~1.EML		1223	eml	DTSE3_latest_Imag	Zero Ler
!018-0~1.EML		1224	eml	DTSE3_latest_Imag	Zero Ler
!21995~3.SQL		1925	sql	DTSE3_latest_Imag	Zero Ler
!AVICO~2.SQL		1393	sql	DTSE3_latest_Imag	Zero Ler

Obrázek 8.1: Ukázka tabulky v CSV souboru.

Vytváření CSV exportu je mnohonásobně rychlejší, než vytváření reportů. Například CSV soubor s **500 147** záznamy se vyexportuje za **11 minut a 16 sekund**.

8.1 Export dat

CSV výstup obsahuje pouze informace o souborech a samotné soubory neumožňuje vyexportovat současně, jako to je u reportu. Avšak je možné vyexportovat soubory samostatně. Export umožňuje vybrat soubory, které chceme exportovat (všechny označené checkboxem, všechny zvýrazněné, všechny zobrazené, nebo všechny soubory zahrnuté v otevřeném případu). Dále je možnost exportovat všechny soubory do jedné složky tak, jak je to u reportů anebo vyexportovat data do původní adresářové struktury tak, jak byla uložena na původním médiu.

9. Návrh rozšíření aplikace o zpracování CSV souborů

Funkční požadavky na rozšíření zůstávají stejné jako u zpracovávání HTML reportu. Nicméně musí dojít k přepracování několika prvků:

- **Načítání dat**

  Načtení složky Report_Files pro HTML report zůstane a přibude další možnost a to konkrétně načtení dat z CSV souboru. K CSV souboru bude ještě možné načíst složku s vyexportovanými soubory, pokud je uživatel bude mít k dispozici.

- **Import hotové databáze a indexů**

  U velkých CSV souborů (například u zkoumání pevných disků, kde se jedná o milióny záznamů) se musí počítat s časovou náročností načítání dat do databáze, nebo indexace přiložených souborů. Z tohoto důvodu aplikace musí disponovat možností nahrát již hotovou databázi, nebo složku s indexy, aby nebylo nutné pokaždé data načítat znovu.

- **Načítání více CSV souborů**

  Další novou funkcí bude načítání více CSV souborů do jedné databáze. Tato funkce umožní procházet více CSV souborů najednou a tím usnadní a urychlí zkoumání dat.

9.1 Report Enhancer Reader

Samostatnou aplikaci Report Enhancer by měl využívat policejní znalec, který by v ní vytvářel výstupy v podobě PDF souborů a ty dále předával policejním orgánům. Díky přidané funkci načítání již hotové databáze se nabízí další možnost, jak zkoumaná data předávat. Z tohoto důvod je potřeba vytvoření druhé aplikace, která by umožňovala pouze načítání již hotové databáze, načítání složky indexů nebo načtení a indexování vyexportovaných souborů. Tato aplikace by se jmenovala Report Enhancer Reader a umožnila by dynamické procházení dat policejním orgánům.

10. Implementace rozšíření aplikace o CSV souborů

Při rozšiřování došlo k pozměnění, nebo přidání funkcionalit aplikace, avšak nějaké zůstaly totožné. Například **Fulltext vyhledávání** zůstává po rozšíření aplikace stejné jako u HTML reportu. Také průzkumník dat zůstává stejný se všemi svými funkcemi, pouze došlo k doplnění dalších.

Celá aplikace je nyní rozdělena do 4 částí, a to `Databáze`, `parsování dat`, `Report Enhancer` a `Fulltext vyhledávání`. Každá část se stará o určitou funkcionalitu aplikace.

10.1 Parsování dat

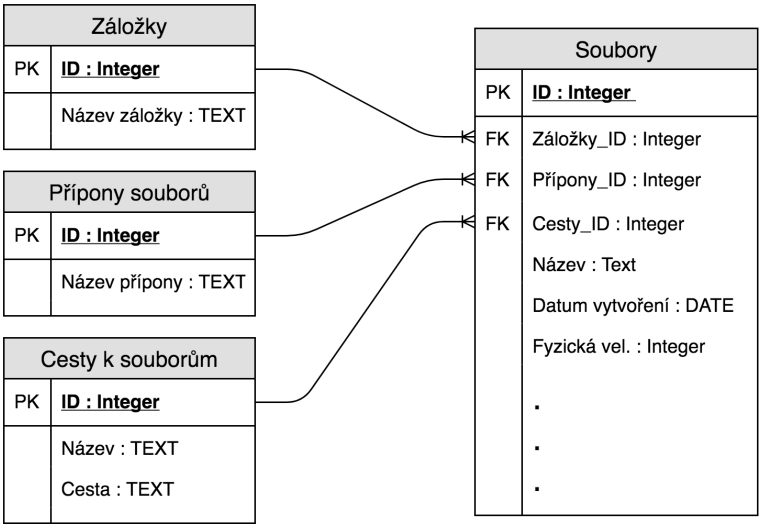
O parsování HTML reportu se stále stará třída `HRMLReportParser`. Aplikace byla rozšířena o metodu **CSVParseTask**, která se stará o parsování CSV souboru do databáze. O nahrávání dat ze vstupu do databáze se stará metoda `Call()`. Metoda prochází CSV soubor a ukládá pouze data ze sloupců obsažených v tabulce 5.1, ostatní sloupce ignoruje. O to se stará podmínka na ukázce kódu 10.1, která při prvním průběhu ukládá názvy sloupců z prvního řádku CSV souboru do ArrayListu `headers`. Při dalším průběhu prochází další řádky výstupu a ukládá jednotlivé hodnoty pod názvy sloupců do HashMapy. Klíčem HashMapy je název sloupce.

Listing 10.1: Podmínka s cyklem pro ukládání informací z CSV.

```
if (lineCounter == 0) {
    headers.addAll(Arrays.asList(line.split(",")));
} else {
 String[] s = line.split(",");
 HashMap<String, String> record = new HashMap<String, String>();
 for (int y = 0; y < s.length; y++) {
 if (y < headers.size()) {
 record.put(headers.get(y), s[y]);
        }
    }
```

10.2 Databáze

Rozšíření aplikace vyžaduje také rozšíření databáze o tabulku obsahující cesty k souborům. Jelikož CSV soubor obsahuje pouze původní cesty zkoumaných souborů z původního média a nedisponuje přímý odkaz na vyexportované soubory tak, jak je tomu u HTML reportu, protože zkoumané soubory se k CSV souboru musí exportovat z FTK zvlášť. Navíc soubory z FTK lze exportovat buď všechny do jedné složky, nebo podle struktury původního uložení souborů. Z tohoto důvodu je nutné adresář s vyexportovanými soubory projít průzkumníkem, který jej projde a uloží cestu ke každému nalezenému souboru do nové tabulky `Cesty k souborům`. Průzkumník je součástí metody `Call()` parsující CSV soubor. Struktura nové databáze je znázorněna na obrázku 10.1.



Obrázek 10.1: Nový diagram databáze.

10.3 Report Enhancer

Tato část je rozdělena podle MVC architektury. Aplikace využívá `Executor` rozhraní, které umožňuje pomocí příkazu `Executors.newSingleThreadExecutor();` rozdělovat úkony aplikace do samostatných vláken procesoru a tím bylo zamezeno „zamrzání“ aplikace a bylo docíleno větší plynulosti.

1. **Model** - se stará o vstupní data aplikace.

   Například časové údaje jsou v SQL databázi uložené pomocí datového typu `Long`, a proto je nutné je převést do čitelné podoby. O to se stará třída `EnhancerDate`. Pro převod byla použita třída `Calendar` a převod je zobrazen na ukázce kódu 10.2. Časové údaje se mohou v databázi vyskytovat ve třech formách a to: žádný časový údaj, den.měsíc.rok, den.měsíc.rok hodina:minuta:vteřina. Proto kód obsahuje podmínku, která tyto formáty rozlišuje.

Listing 10.2: Kód převádění časových údajů.

```
Calendar cal = Calendar.getInstance();
cal.setTime(this);
if (this.compareTo(new Date(0)) == 0){
 return "";
}
else if( (cal.get(Calendar.HOUR) == 0)
 && (cal.get(Calendar.MINUTE) == 0)
 && (cal.get(Calendar.SECOND) == 0) ){
 return new SimpleDateFormat("dd.MM.yyyy").format(this);
}
else {
 return new SimpleDateFormat
   ("dd.MM.yyyy HH:mm:ss").format(this);
}
```

2. **View** - obsahuje veškeré grafické prvky celé aplikace uložené v FXML souborech Javy FX
3. **Controler** - stará se o veškeré funkční prvky aplikace
    - SetupReportFolderControler - stará se o rozhraní importování dat do aplikace
    - FullTextSearchController - stará se o fulltextové vyhledávání textových řetězců
    - DataRangeSearch - stará se o vyhledávání pomocí časových údajů
    - CheckSumControler - stará se o výpočet kontrolních součtů souborů
    - ExportWizardController - stará se o vytváření výstupu z aplikace
    - GenerateCheckSumControler - stará se o generování kontrolní sumy vytvořeného výstupu
    - IndexFilesController - stará se o indexaci souborů
    - MainController - stará se o načítání dat do průzkumníka a následovné procházení těchto dat
    - CSVImportController - stará se o načítání dalších CSV souborů do databáze

10.4 Report Enhancer Reader

Jedná se o totožnou aplikaci jako je samotný Report Enhancer, pouze byla odstraněna většina možností importu dat kromě načítání souboru databáze, adresáře s indexy nebo načtení adresáře se soubory a možnost indexace těchto souborů.

10.5 Přenositelnost aplikace

Aby byla aplikace přenositelná (portable), nevyžadovala žádnou instalaci a nebylo nutné mít nainstalovanou Javu, musí k ní být přibalen adresář s JRE (Java Runtime Enviroment = prostředí pro spouštění Java aplikací). Toho bylo docíleno pomocí Launch4j - jedná se o multiplatformní nástroj pro balení Java aplikací v podobě `jar` souborů (formát používaný k distribuci Java aplikací) do spustitelných `exe` souborů. Dále umožňuje nastavování a přibalování JRE k aplikaci.[12]

K aplikaci je přibaleno starší JRE ve verzi 8, které ještě obsahuje JavaFX. Jelikož novější verze už JavaFX neobsahují, tak je v konfiguraci aplikace striktně nastavená nejvyšší možná verze JRE a to verze 8. Toto nastavení je důležité proto, že pokud se aplikace spouští na stroji, který má Javu nainstalovanou ve vyšší verzi, může dojít ke spuštění aplikace na ní a to by mohlo vést k chybám v aplikaci.

Složka aplikace tedy obsahuje jeden adresář s JRE a z důvodu dvou samostatných aplikací dva xml soubory konfigurace aplikace, dva jar soubory a dva spustitelné soubory aplikace.

Nevýhoda tohoto řešení je větší velikost celé aplikace (konkrétně 328 MB), a to kvůli přibalenému JRE, které má 196 MB.

11. Testování

Testování aplikace je rozděleno na tři části. První část je otestování načítání dat a funkcí aplikace. Druhá část je otestování přenositelnosti aplikace a třetí část je otestování časové využitelnosti aplikace.

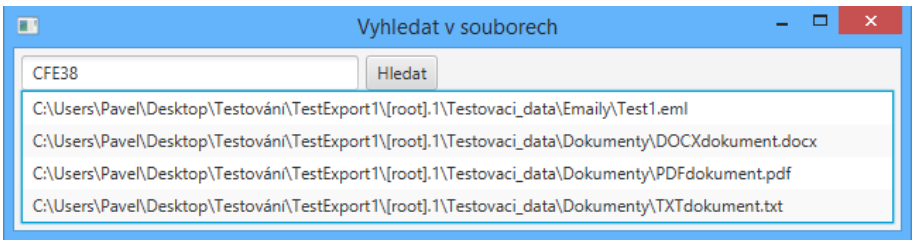
11.1 Funkčnost aplikace

Funkčnost aplikace je testována na vytvořených vzorových datech, které jsou obsažené v příloze práce. Vzorová data obsahují: jeden HTML report, dva CSV soubory a dvě složky s vyexportovanými zkoumanými soubory k CSV souborům.

Načítání
První možnost je načtení HTML reportu nebo CSV souboru. Aplikace zvládá načítaní menších HTML reportů, velké HTML reporty (například report celého pevného disku) načítat nezvládá. Z toho důvodu byla aplikace rozšířena o CSV soubory. Součástí první fáze je i volitelná indexace. Volitelná z toho důvodu, že je možné data indexovat až při potřebě vyhledávání textových řetězců v průběhu procházení dat, nebo vůbec. Druhá možnost je načítání vytvořené databáze a adresáře s indexy, které se vytváří pomocí první možnosti.

Vstupy pro import HTML, CSV a databáze jsou ošetřeny pomocí omezení vstupu pouze na dané formáty povolených vstupních souborů. Problém může nastat u vstupů adresáře se soubory, nebo indexy. Pokud do těchto vstupů uživatel vloží špatné adresáře, tak některé funkce aplikace budou nefunkční. Největší testovaný CSV soubor obsahoval více jak 3 miliony záznamů a byl úspěšně načten. Načítání menších HTML reportů, databází, CSV souborů nebo více CSV souborů do jedné databáze nevykazovalo při testování žádné problémy. Proto je načítání dat vyhodnoceno kladně. Součástí testovacích dat je i CSV soubor s 500 148 záznamy

Fulltext vyhledávání
Pro otestování vyhledávání textových řetězců byl vložen náhodný řetězec „CFE38“ do 4 různých souborů (txt, pdf, docx, eml) obsažených ve vzorových datech. Aplikace našla všechny 4 výskyty hledaného řetězce (obrázek 11.1 na další straně).



Obrázek 11.1: Testování fulltext vyhledávání.

11.2 Přenositelnost aplikace

Přenositelnost aplikace byla otestována spuštěním aplikace na několika počítačích s různou konfigurací a rozdílnými operačními systémy.

Tabulka 11.1: Tabulka testování přenositelnosti aplikace.

Hardware	Software	Stav aplikace
1. Notebook Lenovo, 4GB RAM, Intel Core i3	Windows 10	Funkční
2. Notebook HP, 6GB RAM, Intel Core i5	Windows 8.1	Funkční
3. Stolní počítač, 16GB RAM, AMD Ryzen 5	Windows 10	Funkční
4. Stolní počítač, 8GB RAM. Intel Core i5	Windows 7	Funkční
5. Notebook Asus, Intel Core i3, 4GB RAM	Ubuntu 19.04	Funkční

Na operačním systému Ubuntu je aplikace funkční, pouze je nutné jí spouštět pomocí JAR souboru a je nutná instalace Javy 8. Na všech ostatních strojích běžela aplikace bez jakéhokoliv problému, a proto je přenositelnost aplikace vyhodnocena kladně.

11.3 Časová využitelnost

Časová využitelnost musí být brána v kontextu se samotným FTK Forensic Toolkitem. Prvním faktorem je hardware, na kterém aplikace běží. Stejně tak jako je FTK závislé na hardwaru a podle toho se odvíjí výkon, je závislá i vyvinutá aplikace. Druhým faktorem je samotná časová náročnost práce s FTK, kdy se velké objemy dat mohou zpracovávat i několik dnů a dochází k prodlevám při procházení těchto velkých dat.

Malé (stovky až tisíce záznamů) HTML reporty a CSV soubory načítá aplikace v řádech vteřin a procházení těchto dat je plynulé bez čekání. Načítání již hotové databáze z těchto vstupů je okamžité.

Pro testování velkých CSV souborů (statisíce až miliony záznamů) byly vytvořeny dva testovací soubory - střední a velký. Tyto soubory byly načteny na prvním, druhém a třetím počítači z tabulky 11.1 a výsledky jsou znázorněny v tabulce 11.2. Z tabulky je názorně vidět, že doba načítání dat je velmi závislá na výkonu počítače.

Tabulka 11.2: Tabulka testování časové využitelnosti.

Export	Notebook HP	Notebook Lenovo	Stolní počítač
Střední (500 148 záznamů)	5 minut 40 sekund	6 minut 31 sekund	52 sekund
Velký (2 383 799 záznamů)	41 minut 25 sekund	51 minut 17 sekund	6 minut 15 sekund

U velkých importů dochází k menším prodlevám u procházení dat, tyto prodlevy jsou u takto velkého množství dat přijatelné. Práci s velkými importy velmi usnadňuje možnost nahrávání již hotové databáze, kde dochází pouze k načítání dat z databáze do tabulky průzkumníka.

Mezi časově náročné procesy patří i indexace. Ovšem indexace probíhá až po načtení dat a běží na pozadí, tudíž je možné současně s indexací data již procházet. Složku se soubory je nutné indexovat pouze při prvním načítání, poté je možné načítat složku s již hotovými indexy.

Z časové využitelnosti byla aplikace na základě výše uvedených informací vyhodnocena kladně.

11.4 Návrh na zlepšení

Během tvorby a testování aplikace bylo zjištěno několik možností, kterými by se dala aplikace vylepšit. Vytvořená aplikace ve vstupních datech vyhledává pouze konkrétní informace a ostatní ignoruje. Vylepšení by spočívalo v dynamickém vytváření databáze podle vstupních dat, a tím zahrnutí všech informací. Do budoucna by mohlo být vylepšeno i načítání HMTL reportů, aby bylo možné načítat i větší reporty a reporty s více úrovněmi záložek.

12. Závěr

Bakalářská práce v úvodní části obsahuje seznámení se zásadami forenzního zkoumání digitální techniky, popisuje forenzní nástroj FTK Forensic Toolkit a práci s ním. V další části práce byly analyzovány reporty FTK v různých formátech a byl vybrán nejlepší formát pro vytvářenou aplikaci. Další část práce popisuje návrh a implementaci aplikace, avšak v průběhu vývoje došlo k závěru, že vybraný formát reportu není vhodný. Došlo tedy k nalezení nového řešení, které bylo popsáno a následoval návrh nové aplikace a dále její implementace.

Cílem této bakalářské práce bylo vytvoření aplikace zpracovávající výstupy z forenzního nástroje FTK Forensic Toolkit, která by umožnila dynamické procházení a vyhledávání důkazních dat. Vytvořená aplikace umožňuje tři způsoby importování dat - menší HTML reporty, CSV soubory (je možné načítat více CSV souborů do jedné databáze) a databázové soubory. Aplikace umožňuje načtená data zobrazovat, řadit, vyhledávat textové řetězce, filtrovat podle časových údajů nebo přípon souborů a kontrolovat integritu souborů. Následně lze vytvářet výstup z nalezených dat a dále je prezentovat. Byly vytvořeny dvě verze aplikace - jedna se všemi funkcemi a druhá omezená pouze na prohlížení databáze vytvořené pomocí první verze. Obě verze aplikace jsou plně přenositelné a není nutná instalace samotné aplikace ani žádných jiných prvků.

Vytvořená aplikace byla otestována z funkčního a časového hlediska. Veškeré funkce aplikace byly shledány funkčními. Z časového hlediska dopadla aplikace v porovnání se samotným FTK velmi dobře. Aplikace splnila veškeré požadavky zadání bakalářské práce.

Seznam obrázků

Seznam tabulek

Listings

Literatura

[1] SVETLÍK, Marián. Digitální forenzní analýza a bezpečnost informací. Data Security Management [online]. [cit. 2019-02-17]. Dostupné z: https://www.rac.cz/RAC/homepage.nsf/CZ/Clanky/\$FILE/DSM-Digit\%C3\%A1ln\%C3\%AD\%20forenzn\%C3\%AD\%20anal\%C3\%BDza-01-2010.pdf

[2] Forensic Toolkit (FTK) User Guide [online]. Lindon(USA): AccessData Group, 2018 [cit. 2019-02-28]. Dostupné z: https://support.accessdata.com/hc/en-us/articles/204056525-FTK-User-Guide

[3] RIVEST, R. The MD5 Message-Digest Algorithm [online]. MIT Laboratory for Computer Science and RSA Data Security, Inc., 1992 [cit. 2019-03-02]. Dostupné z: https://www.rfc-editor.org/pdfrfc/rfc1321.txt.pdf

[4] HAROLD, Elliotte Rusty. XML bible. Foster City, Calif.: IDG Books Worldwide, 1999. [cit. 2019-03-06] ISBN 07-645-3236-7

[5] FARRELL, Joyce. Java Programming. Seventh edition. Boston: Course Technology, 2017. [cit. 2019-03-10] ISBN 1-285-08195-1

[6] DEA, Carl. JavaFX 8: introduction by example. Second edition. New York, New York: Apress, [2014]. [cit. 2019-03-10] ISBN 978-1-4302-6460-6

[7] SQLite [online]. [cit. 2019-03-10]. Dostupné z: https://sqlite.org/draft/matrix/about.html

[8] Jsoup [online]. Jonathan Hedley, c2009-2018 [cit. 2019-03-13]. Dostupné z: https://jsoup.org/

[9] GOSPODNETIC, Otis a Erik HATCHER. Lucene in action. Greenwich, CT: Manning, [2005]. ISBN 1-932394-28-1

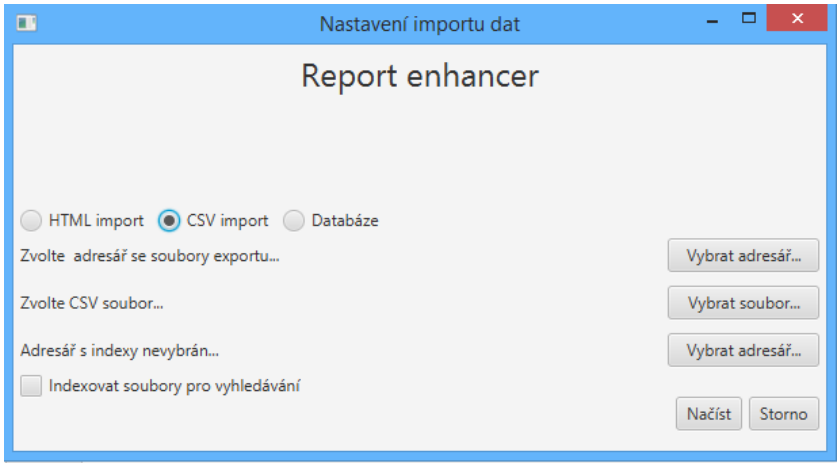
[10] PDFBox [online]. Wakefield, USA: Apache Software Foundation, c2009-2019 [cit. 2019-03-14]. Dostupné z: https://pdfbox.apache.org/

[11] POI [online]. Wakefield, USA: Apache Software Foundation, c2001-2019 [cit. 2019-03-14]. Dostupné z: https://poi.apache.org/

[12] Launch4j [online]. Grzegorz Kowal, c2005-2017 [cit. 2019-03-28]. Dostupné z: http://launch4j.sourceforge.net/

13. Přílohy A - Manuál k aplikaci

Spuštění aplikace
Pro zprovoznění aplikace je nutné pouze zkopírovat adresář *ReportEnhancer* z přiloženého disku do počítače. Aplikace se spouští pomocí exe souboru `ReportEnhancer` nebo `ReportEnhancer Reader` obsaženého v adresáři. S oběma aplikacemi se pracuje naprosto stejně, pouze `Reader` má omezené možnosti importu dat.



Obrázek 13.1: Formulář pro nastavení importu dat.

Import dat

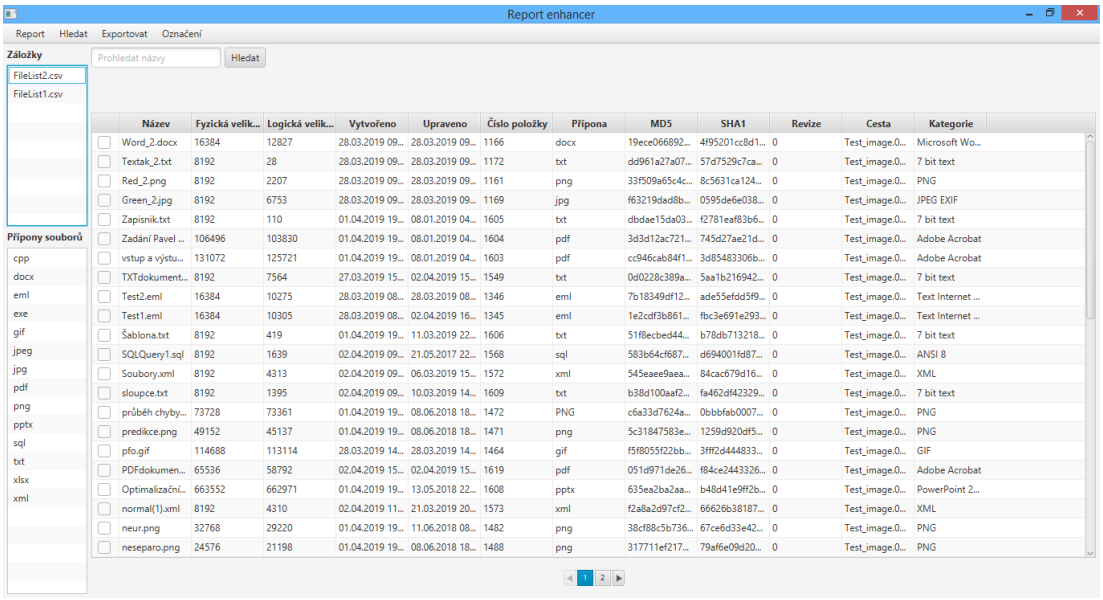
1. HTML import - vyžaduje vybrání adresáře `Report_files`. Pokud se nejedná o první načítání daného reportu a již máme vytvořené indexy, tak je možné vybrat adresář s indexy.

2. CSV import - vyžaduje vybrání CSV souboru. Pokud k CSV souboru máme vyexportované i zkoumané soubory v adresáři, můžeme jej vybrat. Pokud již máme vytvořené indexy pro dané zkoumané soubory, je možné načíst pouze adresář s indexy. Načtení dalších CSV souborů je možné v průzkumníku dat.

3. Databáze - vyžaduje vybrání souboru databáze, který se vytváří při prvním načítání daného importu. Databáze obsahuje pouze data z HTML reportu nebo CSV exportu, proto pokud chceme pracovat i s vyexportovanými zkoumanými soubory, tak je nutné zadat buď adresář, kde jsou soubory uloženy, nebo složku s indexy.

Všechny tři výše uvedené volby obsahují zaškrtávací políčko pro indexaci. Indexace je možná pouze, pokud je vyplněna cesta ke zkoumaným souborům (u HTML reportu jsou zkoumané reporty součástí složky `Report_files`).

Soubor databáze a adresář s indexy se vytváří v adresáři aplikace. Pokud tyto data chceme ponechat a znovu načítat, je nutné je manuálně zálohovat, jinak budou při dalším načítání přepsána.



Obrázek 13.2: Průzkumník dat.

Procházení dat

V levé části průzkumníka se nachází rozdělení dat. U HTML reportů jsou data v horním seznamu rozdělena podle záložek, u CSV exportu jsou data pouze v jedné úrovni tak, jak je tomu v samotném exportu. Druhý seznam rozděluje data podle přípon souborů. Jedno kliknutí myší na zvolenou koncovku nebo záložku zobrazí tabulku s daty, druhé kliknutí tabulku vyruší. Pokud je vybrána konkrétní záložka v HTML reportu a vybereme konkrétní příponu souboru, zobrazí se data pouze z vybrané záložky s vybranou příponou souboru.

Položka `Report` - obsahuje možnost zobrazení Case Information, File Overview, Evidencelist (pouze u HTML reportů) a také obsahuje možnost pro importování dalšího CSV souboru. K dalšímu CSV souboru je možné nahrát i složku s vyexportovanými zkoumanými soubory a indexovat je (indexovat tyto soubory lze pouze při importování).

`Řazení` - data se řadí kliknutím myší na daný sloupec, který chceme seřadit (vzestupně, sestupně, žádné řazení).

`Zobrazování souborů` - soubory lze otevřít dvojklikem myší, nebo pravým tlačítkem a vybrání možnosti *Otevřít*. Soubory se otevírají pomocí programů, se kterými jsou asociovány na daném počítači.

`Vyhledávání v názvech souborů` - je možné pomocí textového pole v horní části průzkumníka.

`Kontrolní součet` - pravé tlačítko myši na vybraném souboru v tabulce a zvolení volby *Kontrola součtu*.

Položka `Označení` - obsahuje možnost zrušit označení všech souborů a možnost označit všechny zobrazené soubory.

Vyhledávání v datech

Fulltext vyhledávání textových řetězců se nachází v horní liště průzkumníka pod položkou *Hledat* a volbou *Prohledat soubory*. Toto vyhledávání funguje pouze, pokud byla

provedena indexace souborů, nebo byly načteny již hotové indexy. Pokud tomu tak není, aplikace nabídne možnost indexování souborů.

Vyhledávání v časových údajích se nachází pod položkou *Hledat* a pod volbou *Vyhledat v časovém rozmezí*. Toto vyhledávání funguje pouze nad sloupci s časovým údajem obsažených v daném importu.

Export
Export dat nalezneme v horní liště pod položkou *Exportovat*.

1. Zvolení dat k exportu - označené, zobrazenou kategorii, vše.

2. Zahrnutí informací do reportu - Case Information, File Overview, Evidencelist je možné zahrnout pouze u HTML reportů a pouze pokud jsou součástí daného reportu. Zahrnutím obrázků, e-mailů a textových souborů dojde k zobrazení celého souboru v exportu.

3. Exportování - vytvoří se soubor HTML, který se uloží do adresáře aplikace. Soubor se automaticky otevře a nabídne možnost vytisknutí, nebo uložení do PDF formátu.

4. Kontrolní součet - pro kontrolu integrity při distribuci exportu je možné vygenerovat kontrolní součet exportu. Tuto možnost nalezneme v horní liště pod položkou *Exportovat* a volbou *Generovat kontrolní součet*. Tímto způsobem je možné vygenerovat kontrolní součet pro jakýkoliv soubor.

14. Přílohy B - Obsah přiloženého CD

- adresář s aplikacemi Report Enhancer a Report Enhancer Reader
- adresář s testovacími daty
- adresář se zdrojovými soubory aplikace ve formě projektu pro IntelliJ IDEA
- adresář s dokumentací projektu