

Vybrat-sw-řešení-umožňující-dosažení-cíle-práce.objective1 (metody, výsledky)

Úvod -motivace (problém)

Nalezení informací o připravených programech je v důsledku nesystematického, či chybějícího zápisu, přinejlepším poměrně časově náročné.

(chybějící zápis) -Proč není zapsaný? ->

Protože lidé nevidí dostatečný smysl v zápisu možných informací. -Proč?->

Protože se jim buď zatím nepodařilo najít přijatelný způsob využití, nebo pro vytvořené záznamy prostě žádný důvod nevidí. -Proč->

Protože *aktuálně neexistuje žádný systém, který by říkal co, jak a kam zapisovat. A zároveň naopátku umožňoval v minimálním čase zobrazovat pouze specifické informace ani z malého, natož většího objemu záznamů.*

Úvod -cíl

Navrhnout systém pro sdílení *znalostí o připravených programech.*

Který by *uživatelům* poskytl možnost

efektivního (*rychle, správně*)

a zároveň

příjemného (*S co nejnižší bariérou, která je potřeba překonat pro použití novými uživateli.*)

vyhledávání v uloženém obsahu.

Spolu se zachováním

alespoň **stejně úrovně kvality uživatelské zkušenosti**

při zapisování, jako v aktuálním řešení.

Bez toho, aniž by navržený systém vyžadoval *více zdrojů na provoz a údržbu*, než sám ušetří při svém *využívání*.

Úvod -dílčí cíle

	a	b
1	Vybrat sw řešení umožňující dosažení cíle práce.	analýza literatury [] + komparace, miltikriteriární výběr
2	Vybrat výseč reality relevantní pro skautské programy.	analýza literatury [] + konceptuální modelování :UML-classDiagram []
3	Navrhnout schema pro databázi.	Z modelu pojmů podle doporučených postupů, na základě využití báze []
4	Ověřit úspěšnost dosažení výsledků.	experiment/implementac []

Úvod -struktura

Členění práce je provedeno podle doporučené struktury pro výzkumné práce IMRAD []

První následující kapitolou je Metodika, ta s využitím výše uvedeného schématu definuje metody využitě pro dosažení stanovených dílčích cílů.


Výsledky definované metodiky, jsou prezentovány v částech Teorie a Vypracování.

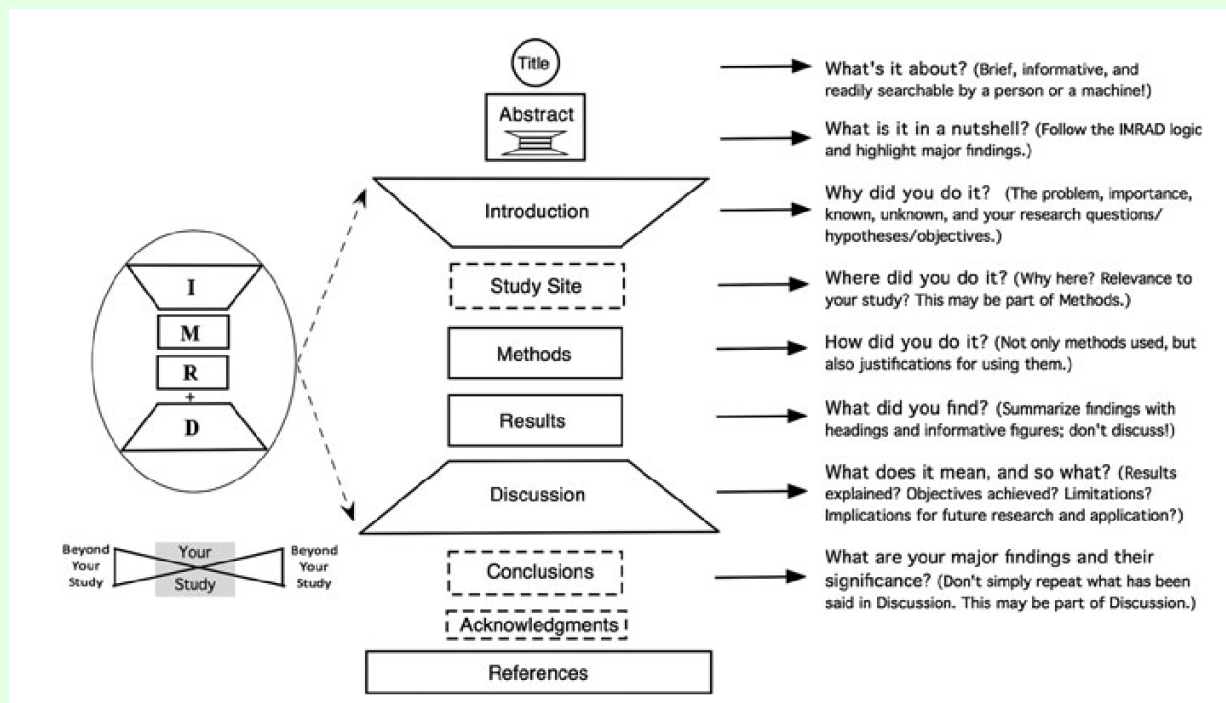
Část Teorie zahrnuje výsledky i interpretaci výsledků 'analýzy softwarových kandidátů', která je použita jako metoda pro 1. dílčí cíl. Z 'analýzy existujícíchází', metody zvolené pro dosažení 2. dílčího cíle, jsou však v teoretické části pouze její výsledky s nejstručnějším popisem.

Interpretace výsledků získaných 'analýzou existujícíchází' je totiž uskutečněno sestavením diagramu tříd vycházejícího ze standardu UML. A jelikož proces konceptuálního modelování, narozdíl od přímočarého porovnávání softwaru, nepředstavuje triviální záležitost a zejména pak proto, že je vytvořený model přímým vstupem pro 'návrh schématu báze' (3.dílčí cíl), jsou výsledky z tvorby modelu (interpretace výsledků získaných analýzou existujícíchází) podrobně popsány až v úvodu praktické části 'Vypracování'. Dále se v praktické části nachází popis navrženého schématu pro databázi, které bylo odvozeno z konceptuálního modelu pomocí nejlepších praktik pro modelování dat ve vybrané databázi, identifikovaných autory databáze.

Na prezentované výsledky navazuje kapitola Diskuze, kterážto zahrnuje kromě 'ověření úspěšnosti dosažení cílů' (4.dílčí cíl) také vysvětlení získaných výsledků. Rovněž se tato kapitola vyjadřuje k limitacím přijatým pro dokončení práce a z nich vyplývající potenciální budoucí práci na zdokonalování nyní dosaženého návrhu.

Poslední kapitolou je pak Závěr, který shrnuje a hodnotí dosažení hlavního cíle práce (v tomto případě: 'návrh báze znalostí skautských programů').

 _Zot..



--Page 1346

Metodika

V této kapitole jsou definovány metody využitě prací k dosažení dílčích cílů. Pro každý z cílů je nejprve definován a vysvětlen jeho účel, popsán a zdůvodněn konkrétní postup,

případně i obecný postup, pokud je takový aplikován, a na závěr jsou identifikovány části práce, ve kterých jsou výstupy konkrétních metod prezentovány.

Metodika analýzy sw kandidátů

Účel postupu

Jelikož tato práce neklade na navrhovaný systém zrovna nízké nároky (viz. Cíl), nestačí pouze určit, který obsah zaznamenávat a opomenout přitom řádný výběr softwarového nástroje nebo nástrojů pro uložení zaznamenávaného obsahu. Zároveň nelze tento krok ani vynechat, poněvadž bez volby alespoň konkrétního typu SW pro uložení dat (npř. RDB, GDB, ...), není možné navrhnout ani konkrétní strukturu nové báze. Co víc, toto rozhodnutí ovlivňuje mimo konceptuální model, jakožto prostředek nezávislý na konkrétní implementaci, všechnu budoucí práci na vývoji, údržbu nasazeného systému i šanci na to, aby byla navržená báze skutečně cílovými uživateli přijata a využívána. Proto je zvolen **dílčí cíl**: "Vybrat sw řešení umožňující dosažení cíle práce.". A účelem tohoto postupu proto je určení místa (SW nástroje) pro uložení záznamů navrhovanou bází, které by splňovalo požadavky definované cílem práce (viz. Cíl) a tím tak bylo odpovědí na otázku "Kam uložit záznamy v navrhované bází?".

Záměrem však není ani tak podrobná či kompletní analýza veškerých dostupných nástrojů, spíše jako porovnání několika softwarových zástupců (kandidátů) s odlišnými typy datových struktur, které se obvykle pro stavbu bází využívají, a zjištění, kteří kandidáti splňují požadavky stanovené v cíli (viz. Cíl).

Jako typy datových struktur k posouzení byly zvoleny dokumenty jakožto výchozí možnost, spreadsheets jakožto kompromis mezi databází a prostředím dokumentů, relační databáze jakožto prakticky standart při tvorbě znalostních bází a grafová databáze jakožto modernější varianta klasické relační DB. Konkrétními kandidáty posuzovanými na funkci uložení záznamů navrhované báze, reprezentující jednotlivé typy zvolených datových struktur, jsou gDocs za dokumenty, gSheets za spreadsheets strukturu, MySQL za RDB a nakonec databáze Neo4j jakožto reprezentace GDB a NoSQL.

První dva SW kandidáti byly zahrnuti do výběru také proto, že se jedná o nástroje aktuálně v našem oddíle využívané, což znamená, že cíloví uživatelé navrhovaného systému již mají určité standardy, na něž jsou zvyklí, že systém nabízí. A jak je prací stanoveno, tyto kvality by neměly být návrhem zredukovány, ba naopak by mělo být provedeno jejich rozšíření. Z toho důvodu jsou zahrnuti druzí dva SW kandidáti, jelikož disponují funkcemi, které v aktuálním řešení chybí.

Obecný postup

Pro dosažení prvního dílčího cíle byla zvolena metoda 'analýza literatury' dodržující postup popsany Bernedtssonem a spol. (berndtssonThesisProjectsGuide2008). Autoři knihy uvádějí, že účelem této metody, mimo získání konkrétních hledaných údajů, je přesvědčení čtenáře práce o tom, že bylo analyzováno dostatečné množství, dostatečně kvalitních a relevantních zdrojů.

Aby čtenář mohl udělat názor o tom, zda byly zdroje dostatečně relevantní, potřebuje znát konkrétní zamýšlený účel se kterým je analýza prováděna. Jasně definovaný účel je pak podle autorů klíčovým prvkem rovněž i pro autora analýzy, jelikož pokud si autor nebude konkrétního účelu vědom nebo ho bude přehlížet, jeho šance na přesvědčení čtenářů o validitě a přínosnosti prováděné analýzy značně klesá. Zároveň specifický účel k provedení odlišuje 'analýzu literatury' od 'rešerše literatury', kterážto má primární účel seznámit autora i čtenáře s obsahem literatury z dané oblasti. []

Autoři knihy dále identifikují následující kroky, které by jak v zájmu autora tak i čtenáře měly mít jasně definovanou strategii provedení.

[myDM/Zotero/LiteratureNotes/berndtssonThesisProjectsGuide2008](#) > [^ZIT8YCXSaNT5KVCQVp67](#)

- **Výběr literatury:** Zahrnuje vyhledávání relevantních zdrojů a literatury související s tématem projektu.

Jasně definovaná strategie pomůže, aby čtenář nepochyboval že byly provedeno adekvátní hledání zdrojů.

- **Hodnocení zdrojů:** Kritické posouzení zdrojů na základě jejich relevance a spolehlivosti.
..., aby čtenář nepochyboval o dostatečném objemu a dostatečné kvalitě zpracovaných zdrojů.
..., aby čtenář porozuměl proč byly některé zdroje vybrané a jiné vynechané.
- **Analýza obsahu:** Podrobné zkoumání a získávání informací z vybraných zdrojů.
..., aby měl čtenář šanci pochopit jak byly výsledky získány.
- **Interpretace výsledků:** Integrace zjištění do koherentního celku a formulace závěrů.
..., aby čtenář rozuměl co získané údaje reprezentují.

Konkrétní postup

Pro nalezení odpovědí hledaných v rámci kroků 'hodnocení' i 'analýzy' jsou využity publikované zdroje k daným nástrojům, primárně pak dokumentace. Avšak platí, že tato analýza se nezabývá zdroji o nástrojích, ale nástroji samotnými. To znamená, že například v následující podkapitole 'Hodnocení zdrojů k analýze' jsou vnímány jako hodnocené zdroje samotné nástroje, nikoliv zdroje o nástrojích, ačkoliv právě ve zdrojích o jednotlivých nástrojích budou hledány odpovědi při prováděném hodnocení, není však prováděno žádné dodatečné systematické prohledávání či hodnocení dostupných zdrojů o nástrojích. Bylo tak rozhodnuto poněvadž je předpokládáno, že v rámci na webu dostupných informací o analyzovaných nástrojích, jakožto jasně definovaném softwaru, není významná šance, že by nalezené informace obsahovaly vyloženě nepravdivá tvrzení, zejména pak pokud budou při odpovídání upřednostněny oficiální zdroje k danému nástroji.

Výběr zdrojů

Co se prvotního výběru zdrojů (v tomto případě SW nástrojů využitelných jako uložiště báze znalostí) týče, ten je proveden bez rozsáhlejšího vhledávání, protože limitovaný rozsah práce neumožňuje adekvátní zpracování většího objemu variant zároveň spolu s dosažením stanoveného cíle. Konkrétní předvybraní zástupci posuzovaných datových struktur proto byly již, i s argumentací pro jejich výběr, představeni dříve v této kapitole. A protože i specifický účel pro analýzu byl vyjasněn, následující podkapitola se bude zabývat rovnou hodnocením předvybraných zdrojů a určením pro jakou podmnožinu ze SW kandidátů budou následně v rámci 'analýzy obsahu' zjišťovány odpovědi na stanovené analytické otázky.

Hodnocení zdrojů (sw) (čtení, zápis)

V této části budou všichni SW kandidáti v hodnocení ze dvou hledisek (možnosti zápisu a čtení uložených dat) odvozených z cíle práce (viz. Cíl). Každé z obou hledisek přitom bude vyhodnoceno zvlášť. To znamená, že jejich vyhodnocení vyprodukuje dvě sady výsledků, jednu pro každé hledisko.

Hodnocení nástrojů z hlediska zapisování

Podmínka pro zápis říká, že navržená báze má zachovat úroveň uživatelské zkušenosti, jako poskytuje aktuální řešení. Tím jsou gDocs, případně gSheets pro data jako seznam členů. K ani jedné z této variant, nepotřebuje uživatel téměř žádné nestandardní znalosti k tomu, aby informace zaznamenal. Pokud mu bude dán odkaz na dokument do kterého má zápis udělat, a pod jaký nadpis, mělo by to stačit každému. Jako kritérium pro vyhodnocení kandidátů z hlediska zápisu proto bude, zda vyžadují od uživatele znalost, jako například specifický jazyk k tomu, aby mohl zapisovat do uložiště. Pokud ano, nejsou pro návrh z hlediska zapisování do báze přijatelné.

Výsledkem tohoto hodnocení může být více nástrojů, jelikož není z pohledu cíle práce zapotřebí specifikovat jiná kritéria, než to jedno aktuálně definované.

Hodnocení nástrojů z hlediska čtení

☐ možná mírně rozvést nejkratší věty a dovysvětlit jejich význam

Z hlediska čtení, respektive možností prohledávání uloženého obsahu. Podmínka z cíle říká, že čtení má být efektivní. Měřeno rychlostí na získání výsledků a správností výsledků. To znamená absenci jak chyb kdy je zobrazen výsledek neobsahující hledaný obsah (false positive), tak chyb kdy obsah hledaný uživatelem není nalezen, i když ve skutečnosti je zaznamenán a měl by se zobrazit (false negative). Z kandidátů tedy budou vyřazeni ti, kteří toto nesplňují.

Pro zbývající pak bude vyhodnoceno stanovené kritérium příjemnosti využití. Definované jako co nejnižší bariera pro nové uživatele, kterou potřebují překonat aby mohli bázi prohledávat. Velikost bariéry pak bude měřena počtem znaků potřebných pro zadávání dotazů. A podle toho, zda k interakci s bází je v základu k dispozici webové grafické rozhraní.

Zbývající kandidáti tedy budou porovnání podle počtu znaků vyžadovaných k napsání dotazu, přítomnosti webového grafického rozhraní ve výchozí instalaci báze. Vyhodnoceno bude nejdříve pořadí kandidátů podle každého z těchto dvou kritérií zvlášť. Následně takto získané ~~tři~~ hodnoty pro každého z kandidátů budou sečteny a kandidát, který bude mít nejnižší výsledné číslo, protože se například podle každého kritéria umístnil na prvním místě, bude přijat jako možná část navrhované báze.

Výsledkem tohoto hodnocení je tak jediný nástroj, který podle sečtených výsledků z vyhodnocení kritérií hlediska zapisování (rychlost, správnost, příjemnost) vychází jako ten nejlepší.

Analýza vybraných zdrojů (sw) (uložení-DS, přístup-PA) -> (zdroje vyžadované na údržbu)

Pro zdroje splňující alespoň jedno hledisko hodnocení, jsou určeny jejich 'vnitřní datová struktura', spolu s jejich 'integrovatelností'.

'Vnitřní datovou strukturou' je myšlena struktura souborů skutečně uložených na disku. To znamená, že může být využita analytická otázka "Jak daný sw ukládá zaznamenané údaje na disk?".

Pojmem 'integrovatelnost' jsou pak varianty skrze které je možné programově přistupovat k uloženému obsahu i ho stejným způsobem modifikovat. Odpovídající analytická otázka proto může být "Jaké možnosti programového přístupu k zaznamenaným údajům daný SW nástroj nabízí?".

A co se 'programového přístupu' týče, typickými příklady jsou například podpora prohledávání databáze pomocí HTTP dotazů, nebo s využitím programovacího jazyka s využitím některé dostupné, na volbě jazyka závislé, specifické knihovny implementující metody pro komunikaci. V obou případech se každopádně jedná o způsoby externí komunikace optimalizované pro využití v kódu konkrétního programu, proto zvolené označení.

Získání nejlepších praktik pro modelování dat v nejlépe prohledatelném nástroji

Navíc pro vyhodnocený nejlepší nástroj z hlediska možností čtení uloženého obsahu budou identifikovány nejlepší praktiky pro strukturování do něj ukládaných dat. Tento krok analýzy je důležitý z toho důvodu, že samotný výběr nástroje nezaručuje jeho správné využití a tím pádem nejvyšší šanci na splnění požadavků zohledňovaných při jeho výběru. Důvodem pro identifikaci nejlepších postupů pouze pro konkrétně jeden nástroj je to, že primární nedostatky aktuálního řešení jsou právě v možnostech prohledávání uložených záznamů, a proto práce klade důraz zejména na zdokonalení tohoto aspektu znalostníchází. V důsledku takto stanovených priorit a opět vzhledem k limitovanému rozsahu práce, je navrhována struktura pouze pro uložení s nejlepšími výsledky z hlediska čtení, i když v rámci návrhu bude pro dosažení Cíle práce využito více než jedno uložení.

Interpretace výsledků (sw)

Vycházejí z výsledků, získaných vyhodnocením předchozích dvou analytických otázek, bude určeno, zda je možné na základě vybraných sw kandidátů možné postavit bázi, která jak specifikuje cíl práce, nebude vyžadovat víc prostředků na provoz a údržbu než sama ušetří. To konkrétně bude provedeno pomocí nalezení způsobu jak eliminovat potřebu na ručně vykonávanou údržbu konzistence a aktuálnosti uloženého obsahu v bázi.

Při tomto hledání je vycházeno z předpokladu, že báze může ušetřit nějaký čas svým využitím, jen když nebude zároveň také vyžadovat čas na údržbu pro svojí správnou funkčnost. Rovněž by také neměla vyžadovat žádné finanční prostředky, jelikož vzhledem k neziskové povaze skauta není způsob, jak by se prostředky vydané na provoz takového systému mohly vrátit.

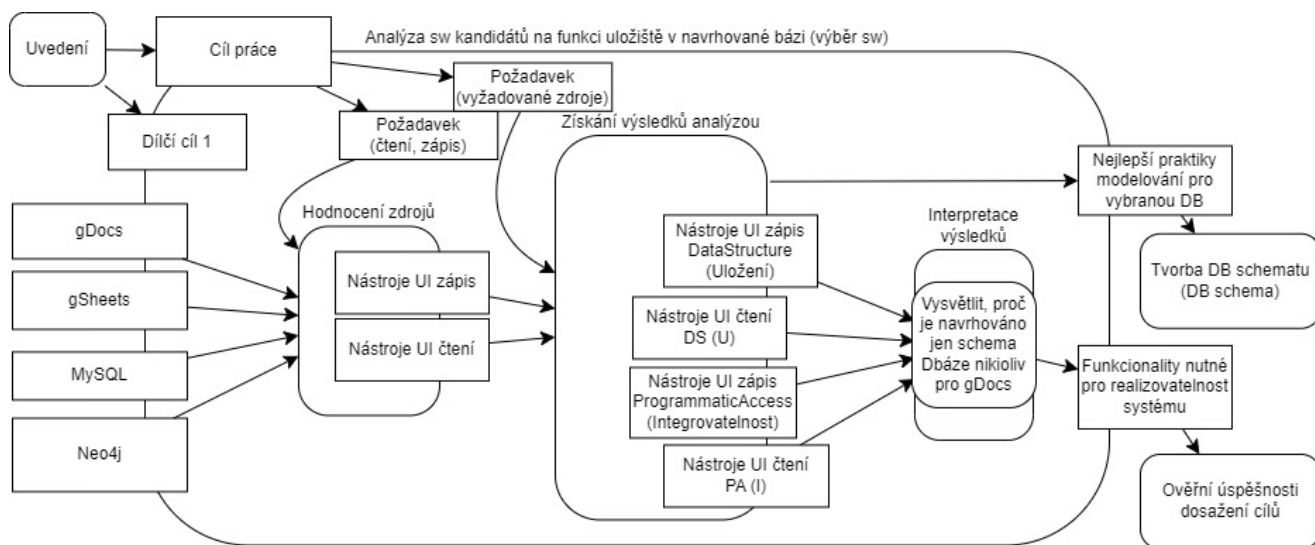
Základní otázkou, kterou se tedy tento krok analýzy bude snažit zodpovědět je otázka "Je možné eliminovat či alespoň zcela minimalizovat potřebu lidských i finančních zdrojů na údržbu konzistence údajů ve vybraných softwarech zaznamenaných?".

V případě nalezení takového způsobu, bude způsob popsán a následně sestaven výčet funkcionalit kritických pro funkcionalitu odpovídající požadavkům na navrhovanou bázi.

Výstupy

☐ DODĚLAT

Schema



Výsledky

V této kapitole budou prezentovány výsledky z výběru softwaru, který by umožnil návrh báze odpovídající podmínkám stanovených v cíli této práce.

Hodnocení zdrojů (sw)

Vyhodnocení hlediska zápisu pro všechny kandidáty

Vyhodnocení tohoto hlediska je velmi přímočaré. Vzhledem k tomu, že tento návrh klade velký důraz na minimalizaci nových nároků na uživatele. Zejména pak na nároky pro zapisování, jelikož pro navrhovanou bázi je klíčové, aby do báze uživatelé zapisovali a sdíleli tak své zkušenosti z připravených programů, čímž budou obohacovat prohledatelný obsah. Proto z tohoto hlediska budou vyřazeni kandidáti, kteří umožňují zapisování obsahu jen pomocí specifického jazyka.

Jak bylo řečeno v metodice, ani jeden z produktů společnosti Google toto kritérium

nesplňuje, zůstávají tedy jako přijatelné pro návrh. Oproti tomu, ani jedna z databází přes toto kritérium neprojde. Pro interakci s databází MySQL je totiž potřeba využít SQL (Structured Query Language) a v případě Neo4j se jedná pro změnu o 'Cypher', což je také jazyk, akorát uzpůsobený k prohledávání grafových struktur.

přijatelné pro zápis

- gDocs
- gSheets

Vyhodnocení hlediska čtení pro všechny kandidáty

Vyhodnocení druhého hlediska však bude již komplexnější, konkrétně tak, že pro získání výsledků využívá vícero kritérií, která jsou na závěr agragována do jednoho souhrnného vyhodnocení.

rychlost

První kritérium se zaměřuje na rychlost získání výsledků. Při následujícím hodnocení kandidátů nebude mít podstatnou roli. Nicméně nedostatečná rychlost vyhledávání v záznamech, pokud je možné jen manuální otevírání jednotlivých dokumentů podle jejich názvu a umístění ve složce, je primárním důvodem vzniku této práce. A proto není možnost manuálního prohledávání ani začleněna mezi kandidáty, kteří všichni toto kritérium splňují.

správnost

Další kritérium, absence chyb jak prvního, tak druhého typu, je však již relevantním pro hodnocené kandidáty.

Nejsnazší vyhodnocení tohoto kritéria umožňují kandidáti databázového typu, v jejich případě je totiž tato podmínka zahrnuta již v jejich podstatě jako databázích. Proto u nich absence chyb při vyhledávání nebude dále ověřována a bude předpokládáno, že toto kritérium splňují.

Pro vyhodnocení bezchybného vyhledávání v gSheets bude posuzována vestavěná funkce 'query', která nabízí podobné možnosti prohledávání tabulek v daném dokumentu gSheets, jako relační databáze svým SQL. Jelikož se tedy jedná opět o prohledávání přesně strukturovaných dat, pomocí exaktního algoritmu, bude předpokládáno, že tato funkce operuje bezchybně. Pro hodnocení vyhledávání v gDocs by mohla být využita funkce 'Najít' rozšiřitelná na 'Najít a nahradit'. Ta nicméně funguje pouze v rámci jednoho dokumentu. Což vzhledem k tomu, že by navrhovaná báze měla být v rámci škálovatelnosti rozdělena do více dokumentů případně na sebe odkazujících, nedělá z funkce 'Najít' vhodný způsob k hodnocení. Na základě předpokladu, že by báze neměla být zapsána v jediném dokumentu, ale spíše rozdělena do více dokumentů, bude k vyhodnocení kritéria bezchybného prohledání u gDocs využita funkcionality prohledávání služby gDrive. Prakticky se jedná o prohledávání obsahu na disku pomocí vyhledávacího řádku na vrchu webového grafického rozhraní služby gDrive. Ten nabízí, mimo možnosti vyhledávat podle názvu a typu souboru, i možnost zobrazit pouze ty dokumenty, které obsahují konkrétní slovo. Právě tato poslední možnost, vyhledávání dokumentů na základě textu v nich obsažených, bude hodnocena podle kritéria bezchybnosti vyhledávání. Toto hodnocení bude provedeno pomocí krátkého experimentu, jenž se bude skládat z vyzkoušení dvou případů.

1. Zapsání klíčového slova "test:." do nového dokumentu a následný pokus o vyhledání dokumentů podle toho zda obsahují klíčové slovo.
Aby byla zohledněna možnost, že systému trvá nějakou dobu, než provede indexování nově vytvořených dokumentů a umožní tak vyhledávání v nich. Bude vyzkoušena ještě následující situace.
2. Vybrání co nejunikátnějšího klíčového textu z libovolného souboru zapsaného na mém disku déle než měsíc, následované pokusem o nezezení daného souboru podle toho, že

obsahuje vybraný klíčový text.

Z výsledků těchto dvou experimentů vyplynulo, že tato varianta prohledávání jednoznačně není bezchybná. Jelikož oba pokusy o vyhledání dokumentu obsahujícího klíčový text byly neúspěšné, byly sice rychlé, avšak dokument z něhož byl získán klíčový text použitý k hledání, nebyl zobrazen ani v jednom případě. Postupujícími sw kandidáty k dalšímu hodnocení jsou tedy pouze gSheets a databáze MySQL a Neo4j. gDocs byly na základě experimentálně zjištěných výsledků vyhodnoceny jako nástroj nepřijatelný pro uživatelské rozhraní zprostředkující čtení obsahu navrhované báze.

příjemnost

Kritérium příjemnosti je ze všech kritérií zatím nejkomplexnější, proto i jeho vyhodnocení bude tomu odpovídat. Jak bylo stanoveno v metodice, bude nejprve vyhodnoceno pořadí zbývajících kandidátů podle každého z dílčích kritérií zvlášť. A tak získané dílčí výsledky budou následně využity k výběru nejvhodnějšího z kandidátů z hlediska příjemnosti prohledávání obsahu navrhované báze.

Následující dílčí kritéria budou vyhodnocena pro gSheets (funkce 'query'), MySQL (SQL) a Neo4j (Cypher). Využito bude primárně odhadů a dedukce.

1. počet znaků potřebných k napsání dotazu

Při hodnocení tohoto dílčího kritéria, je vycházeno z předpokladu, že funkce gSheets 'query', jakožto pouhá napodobenina funkcionality nabízené 'SQL', bude v případě jednoduchých dotazů možná i stejně úsporná na potřebné znaky jako jako SQL pro obdobný dotaz. Pro komplexnější dotazy, vyžadující například data z víc tabulek, je potom předpokládáno, že SQL bude, ve srovnání s funkcí query v gSheets, umožňovat díky své podstatně rozvinutější funkcionalitě způsob zapsání daného komplexnějšího dotazu s nižším počtem znaků, než funkce query.

Pro porovnání počtu znaků vyžadovaných k napsání dotazu bázi Neo4j a MySQL je využito článku s názvem 'Use graph databases for complex hierarchies' []. Tento článek na modelovém příkladu dat, vyhodnocuje několik různých dotazů a porovnává jejich zápis a následný postup vyhodnocení v případě využití SQL oproti případu s využitím jazyka Cypher. V této práci jsou však zohledněny pouze porovnání zapsaných dotazů, nikoliv způsoby jejich vyhodnocování.

Z výsledků prezentovaných v článku vyplývá, že pro zapsání SQL dotazu je téměř v každém případě potřeba více znaků, než pro získání stejných výsledků pomocí jazyka Cypher. SQL přitom v některých případech vyžaduje až několikanásobně více znaku než ekvivalent zapsaný Cypherem. Proto navíc tato práce předpokládá i to, že ani pomocí funkce query v gSheets, není možné dosáhnout kratšího zápisu dotazů, než v případě Neo4j a Cypheru, jelikož odhadovaný počet znaků vyžadovaný funkcí query je v jednoduších případech podobný jako v případě SQL a ve složitějších situacích horší než SQL.

Výsledné pořadí tohoto dílčího kritéria proto je:

1. místo - Cypher
2. místo - SQL
3. místo - =query()

2. Nabízí v základu webové GUI?

Pro vyhodnocení druhého dílčího kritéria není třeba se uchylovat k odhadům, jelikož kandidáti buď budou nabízet možnost webového GUI v základní instalaci, nebo nebudou. Určení této binární hodnoty pro kandidáty, bude provedeno prohledáním internetových zdrojů s dotazem například "MySQL web GUI". A na základě nalezených výsledků bude určeno zda daný kandidát úspěšně splní toto kritérium.

Pro gSheets bylo vyhodnoceno, vzhledem k podstatě nástroje jako na cloudu založené služby, že vskutku nabízí ve svých základních možnostech zobrazení grafického rozhraní v prostředí prohlížeče, bez nutnosti lokální instalace čehokoliv jiného než samotného prohlížeče.

Pro MySQL z hledání vyplynulo, že grafická rozhraní umožňující přístup k této bázi jsou typicky povahy lokální instalace. Rovněž však existují i možnosti jako myPhpAdmin [], který je zdarma a nabízí webové GUI. Nicméně všechny z těchto variant jsou dodatečné nástroje, z nichž by bylo potřeba provést patřičný výběr, kdyby byly návrhem uvažovány

jako možnosti. Jelikož tedy žádné, v základní instalaci zahrnuté webové GUI není pro MySQL k dispozici, znamená to pro relační databázi podle tohoto kritéria druhé a zároveň poslední místo.

Pro Neo4j naopak bylo velmi snadné najít nástroj 'Neo4j Browser', jelikož to byl první výsledek po zadání dotazu "Neo4j web gui". Jednalo se o odkaz na oficiální dokumentaci Neo4j, kde bylo řečeno, že se jedná o výchozí rozhraní pro interakci s databází a zároveň je zahrnuto ve výchozí instalaci

[myDM/Zotero/LiteratureNotes/Neo4jBrowserDocsHomepage](#) > ^HBZ4U6B9aKFRES7KJ.

Výsledné pořadí tohoto dílčího kritéria proto je:

1. místo - gSheets, Neo4j
2. místo - MySQL

Finální pořadí kandidátů podle kritéria příjemnosti čtení jejich obsahu je proto následující.

1. místo - Neo4j (1+1)
2. místo - gSheets (1+3), MySQL (2+2)

A kandidátem vybraným v rámci hlediska čtení zapsaného obsahu se tak stává databáze Neo4j, jelikož umožňuje interakci pomocí dotazů s nejnižším počtem znaků z posuzovaných kandidátů a zároveň k interakci s ní není třeba žádný dodatečný software, který by nebyl zahrnut v základní instalaci.

Analýza vybraných zdrojů (sw)

Vybraným softwarem pro základ navrhované báze jsou tedy gDocs a gSheets jako uživatelské rozhraní pro zapisování údaje do báze a případnou modifikaci zapsaných údajů. Spolu s databází Neo4j slouží jako uživatelské rozhraní k prohledávání zapsaných údajů v navrhované bazi skautských programů. V této části budou představeny datové struktury využívané jednotlivými vybranými nástroji spolu s představením jejich možností vzájemné integrace.

datové struktury (uložení)

gWorkspace

Oba tyto nástroje z prostředí gWorkspace (gDocs, gSheets) mají jeden aspekt své struktury shodný. A to sice ten že v obou případech se jedná o soubory, uložené na disku google (gDrive). Každý ze souborů pak má přiřazené unikátní id, které je mimochodem součástí webové adresy (url) využitě pro zobrazení GUI editoru daného souboru. Pokud tedy v prohlížeči bude otevřen jeden konkrétní soubor tabulek z disku s identifikátorem ID, url zobrazované ve vyhledávacím řádku prohlížeče bude

<https://docs.google.com/spreadsheets/d/{ID}/edit#gid=0> []. Adresa funguje i v případě vynechání textu za posledním lomítkem, i v případě že je text "spreadsheets" nahrazen textem "docs" a je použito ID náležící dokumentu místo tabulek. Kromě id má také každý soubor přiřazený název, typ (gdocs,gsheets,pdf,...) a další. Navíc může být přiřazen například popis, který se zobrazuje v GUI gDrive i gDocs, ale i další volitelné atributy se kterými je však možno interagovat jen pomocí REST API [].

Vnitřní strukturu souborů už však mají oba nástroje specifickou.

V případě dokumentů, je každý tvořen například záhlavím, zápatím a tělem dokumentu (nejedná se o kompletní výčet) []. Tělo dokumentu je pak dále členěno na jednotlivé elementy. V praxi je elementem každý nový řádek vytvořený stisknutím klávesy 'enter', případně vložený objekt jako třeba tabulka, nebo obrázek. Každý element pak může mít přiřazené hodnoty reprezentující jeho formátování, ale i konkrétní text zapsaný v daném elementu. Navíc, protože pro dokumenty je důležité pořadí zapsaných elementů, je s každým elementem asociován i identifikátor vyjadřující pořadí daného elementu v rámci těla dokumentu []. Je tak například možné získat na jakých pozicích, z hlediska pořadí v dokumentu, jsou nadpisy úrovně 1 a pomocí jednoduchých aritmetických operací získat na jakých pozicích v dokumentu začíná i končí elementy pod konkrétním nadpisem 1. úrovně.

V případě tabulek, jsou jednotlivé soubory organizovány do listů (stránek), s tím že každý list je tvořen tabulkovou strukturou ve které může být zapsáno i více tabulek. Konkrétní rozsahy v rámci listů mohou být také pojmenovány a reference na ně tak mohou být realizovány pomocí tohoto pojmenování []. Jelikož se však jedná spíše o sekundární rozhraní pro navrhovanou bázi, které je zamýšleno zejména na zapisování dalo by se říci konfiguračních údajů (dostupné materiály, seznam členů, výchovné cíle), popis struktury jeho souborů není rozebírán do větších podrobností.

Neo4j

Neo4j, vzhledem ke své podstatě databáze, má strukturu uložených údajů značně odlišnou. Struktura označovaná jako 'property graph' využitá Neo4j k uložení zapsaných dat, je na disku realizována pomocí několika odlišných souborů. To konkrétně znamená, že každá část uložené grafové struktury (nodes-vrcholy, relationships-vztahy, labels-popisky/štítky, properties-vlastnosti) je uložena v separátním souboru []. Všechny tyto čtyři soubory se přitom skládají ze záznamů o fixní délce bytů, dalo by se na ně tedy pohlížet jako na tabulky, ve kterých je možné velmi rychle přistupovat ke konkrétním záznamům, pokud známe pořadí ve kterém byly do souboru zapsány. Právě proto je databázi využita tato fixní struktura, jelikož je s její pomocí je možné efektivní propojení jednotlivých částí napříč čtyřmi separátními soubory. Například pokud je k vrcholu přiřazená vlastnost, bude ve vyhrazeném místě (bytech) pro zaznamenání přiřazených vlastností v daném záznamu v souboru vrcholů uvedeno pořadí ve kterém byla přiřazená vlastnost zapsána do souboru obsahujícího vlastnosti. Dalo by se tak říci, že pořadí zápisu jednotlivých záznamů do souborů, představují primární klíče pro jednotlivé "tabulky" a zachycení grafové struktury je dosaženo pomocí zápisu těchto klíčů k ostatním souvisejícím částem jako cizích klíčů.

Dále jsou v knize "Graph databases" od vydavatelství O'Reilly popsány i konkrétní struktury jednotlivých souborů. V rámci představení struktury databáze, jsou proto představeny i tato specifika. Popsaná struktura záznamu v souboru ukládajícím vrcholy je následující [].

- byte 1 - (in-use flag) Slouží bázi k určení, zda je daný záznam používán, či zda může být smazán a jeho pozice tak uvolněna.
 - bytes 2-5 - Reprezentují identifikátor prvního připojeného vztahu (odkaz realizovaný pořadím záznamu v souboru vztahů).
 - bytes 6-9 - Reprezentují identifikátor první připojené vlastnosti (odkaz realizovaný pořadím záznamu v souboru vlastností).
 - bytes 10-14 - Reprezentují odkazy na přiřazené 'labels', případně konkrétní štítky/popisky, pokud jich je přiřazeno pouze nízké množství.
 - byte 15 - Rezervován pro budoucí využití.
- Jak je řečeno v knize, jedná se tak prakticky jen o "hrstku odkazů, odkazujících do seznamů vztahů, popisků a vlastností" [].

Pro záznam v souboru ukládajícím vztahy je pak popsána následující struktura [].

- byte 1 - (in-use flag) Značí, zda záznam může být smazán a nahrazen novým.
- bytes 2-5 - Identifikátor prvního vrcholu v tomto vztahu. V případě směrovaného vztahu je tento vrchol počátkem.
- 6-9 - Identifikátor druhého vrcholu v tomto vztahu. V případě směrovaného vztahu je toto koncový vrchol.
- 10-13 - Identifikátor typu vztahu, který odkazuje na soubor obsahující seznam všech typů vztahů v databázi použitých.
- 14-17 - Identifikátor předchozího vztahu počátečního vrcholu.
- 18-21 - Identifikátor následujícího vztahu počátečního vrcholu
- 22-25 - Identifikátor předchozího vztahu koncového vrcholu
- 26-29 - Identifikátor následujícího vztahu koncového vrcholu
- 30-33 - Identifikátor první připojené vlastnosti.

- 34 - První v řetězu vztahů?

Vlastnosti (properties) jsou potom uloženy následujícím způsobem. Opět je využita fixní velikost pro jednotlivé záznamy. S tím že každý záznam vlastnosti obsahuje odkaz na další související záznam. To je z důvodu, že vzhledem k fixní velikosti uložených vrcholů a vztahů, jsou k těmto částem grafu zaznamenány pouze odkazy na první přiřazenou vlastnost, a ostatní přiřazené vlastnosti jsou pak připojeny právě pomocí odkazu na následující záznam vlastnosti obsažený v záznamu první přiřazené vlastnosti k vrcholu, či vztahu. Podobný princip je pak aplikován i při procházení všech ostatních prvků v zaznamenaném grafu. Kromě tedy odkazu na další záznam v tomto 'řetězu vlastností', je u každého záznamu uveden datový typ dané vlastnosti (jakýkoliv primitivní typ podporovaný Java Virtual Machine, strings, arrays JVM primitivních typů), spolu s odkazem na soubor 'indexu vlastností', který obsahuje jména vlastností použitých v bázi. A na závěr je u každého záznamu vlastnosti buď uložena samotná hodnota vlastnosti, pokud je dostatečně malá, aby se vešla do fixně velkého záznamu. Nebo v případě, že velikost hodnoty přesahuje velikost místa poskytnutého záznamem fixní velikosti, je uložen pouze odkaz na zapsanou hodnotu vlastnosti, a samotná hodnota je uložena do speciálního dynamického uložistiště vlastností, které je realizováno opět samostatným souborem. Ve skutečnosti Neo4j disponuje dvěma těmito dynamickými uložistišti. První je optimalizováno pro uložení delších textů (stringů) a podporuje například full-text indexování a následné prohledávání těchto textů podle obsaženého textu. A druhé optimalizované pro uložení delších polí (arrays), typicky obsahujících čísla, v oblasti strojového učení a neuro sítí také označovány jako vektory, nebo tensors [].

Toto dynamické uložistiště polí proto může být využito například pro uložení takzvaných 'embedded' hodnot, která jsou získány "zakódováním" nějakého vstupu (text, obrázek,...) pomocí AI modelu. Takto získané hodnoty se následně využívají k 'Retrieve Augmented Generation', což prakticky znamená proces, ve kterém jsou nejdříve vyhledány záznamy, jejichž 'embedded' hodnota je vektorově podobná 'embedded' hodnotě uživatelem zadaného dotazu. Z nalezených vektorově podobných záznamů je následně vybráno vrchních x, a ty jsou poskytnuty některému z jazykových modelů spolu s uživatelským dotazem, aby na základě takto obohacených podkladů teprve vygeneroval odpověď zobrazenou nakonec uživateli.

integrovatelnost (přístup)

gWorkspace

Interakci s vybranými nástroji z gWorkspace pomocí programového kódu zprostředkovává googlem provozované REST API, to umožňuje pomocí http dotazů jak získávání obsahu jednotlivých dokumentů, tak i modifikaci jejich obsahu. Pro použití tohoto API je však potřeba každý dotaz adekvátně autorizovat []. Existuje nicméně ještě jedna možnost interakce s dokumenty pomocí kódu, která ale nevyžaduje explicitně autorizovat každý dotaz. Jedná se o interakci se službami v rámci gWorkspace pomocí služby google Apps Scripts, která je rovněž zahrnuta v gWorkspace. Samotné Apps Scripts představují službu, která umožňuje napsání téměř libovolného javascript kódu a jeho spuštění v rámci definovaných limitů zdarma. Scripty mohou být spuštěny buď časovačem, nebo přes "zavolání" url adresy přiřazené automaticky implementaci daného kódu napsaného v Apps Scripts. Hlavní výhodou při použití Apps Scripts je to, že rozhraní ostatních služeb (gDocs, gSheets, gDrive,...) není potřeba volat pomocí REST API a http dotazů, ale stačí rozhraní dané služby přidat jako knihovnu ke psanému scriptu. Jedinkrát při prvním spuštění je třeba odsouhlasit, že jako majitel účtu souhlasíte s tím, aby daný script měl přístup k vybrané službě, a tím starosti s autorizací požadavků končí []. Kromě denního limitu na počet spuštění, je bezplatné využití této služby vykoupeno ještě jedním podstatným omezením. A to sice, že není možné provádět "volání ven" ze skriptu (externí komunikaci) jinak než s využitím předdefinované funkce 'UrlFetch()'. Což zároveň znamená, že i pokud se podaří dostat do skriptu knihovnu například pro komunikaci s databází nebude tato knihovna fungovat [].

Neo4j

Možnosti programové interakce s databází Neo4j závisí na tom, která z implementací je využita. První varianta implementace Neo4j je cloud verze nabízená jako SaaS, spolu s poměrně dostatečným objemem zdrojů v rámci bezplatné úrovně účtu. Tato verze nicméně umožňuje programovou interakci, pouze pomocí knihoven, které jsou sice pro většinu nejběžnějších jazyků k dispozici, takže ve většině případů bude tato varianta nabízet dostatečnou konektivitu. Avšak v případě, jako dříve zmíněné Apps Scripts, které omezují možnosti externí komunikace pouze na http dotazy skrze předdefinovanou funkci, představuje absence podpory http komunikace v cloudové verzi Neo4j poměrně problém. Naštěstí existuje druhá varianta implementace, konkrétně takzvaná 'self-hosted' varianta, která může být například s využitím dockeru, nebo pomocí klasické instalace spuštěna na libovolné výpočetní instanci (počítači). A tato 'self-hosted' varianta umožňuje jak programovou interakci pomocí http tak pomocí knihoven pro konkrétní jazyky.

Získání nejlepších praktik pro modelování dat v nejlépe prohledatelném nástroji

☐ DODĚLAT

Interpretace výsledků analýzy (sw)

☐ DODĚLAT

Jak bylo stanoveno v metodice, v rámci této části bude popsán způsob implementace navrhované báze, který by nevyžadoval víc prostředků na údržbu, než sám ušetří. Konkrétně, vzhledem k tomu, že aktuálně není k dispozici způsob jak změřit ušetřený čas při využívání báze, je vycházeno z předpokladu, že pokud budou vyžadovány lidské, či finanční prostředky na to, aby byla prováděna jednosměrná synchronizace zapsaného obsahu v gDocs do efektivně prohledatelné databáze, nebude ušetřený čas větší, než ten vyžadovaný na údržbu. Proto bude popsána možnost automatizace synchronizačního procesu taková, která by nevyžadovala finanční prostředky na svůj provoz. Rovněž budou definovány konkrétní funkcionality, na kterých závisí proveditelnost popsaného způsobu. Na základě vybraných kandidátů a jim dostupných možností bylo určeno

automatizace jednosměrné synchronizace (zrcadlení)

nasazení databáze

funkcionality k ověření