Gymnázium Brno, třída Kapitána Jaroše

Vývoj aplikace na kontrolování změn na webových stránkách

Závěrečná maturitní práce

Vedoucí práce:
Mgr. Marek Blaha

Jiří Kalvoda

Blansko 2020

Chtěl bych poděkovat Mgr. Marku Blahovi za odborné vedení práce, věcné připomínky a podání cenných rad.

Prohlašuji, že jsem tuto práci vyřešil samostatně s použitím literatury, kterou uvádím v seznamu

V Blansku dne 19. února 2020 .....

Abstract

Kalvoda, J. Development an application for checking changes on web pages

This thesis describe developing and using an application monitoring changes on web pages. This app is produced in c++ language using Qt library. Therefore, the app can be used on multiple platforms. This app come under GNU LGPL licence and is available whit source code. This thesis contain description of its working and controlling, implementation and used software when it has been developed.

Abstrakt

Kalvoda, J. Vývoj aplikace na kontrolování změn na webových stránkách

Tato závěrečná práce popisuje vývoj a použití aplikace na monitorování změn na webových stránkách. Aplikace je vyvíjena v jazyce c++ pomocí knihovny Qt. Díky tomu se jedná o multiplatformní software. Je dostupná včetně zdrojového kódu pod licencí GNU LGPL. Tato závěrečná práce obsahuje popis jejího fungování, implementace a použitého softwaru při jejím vývoji.

Obsah

1 Úvod a cíl práce

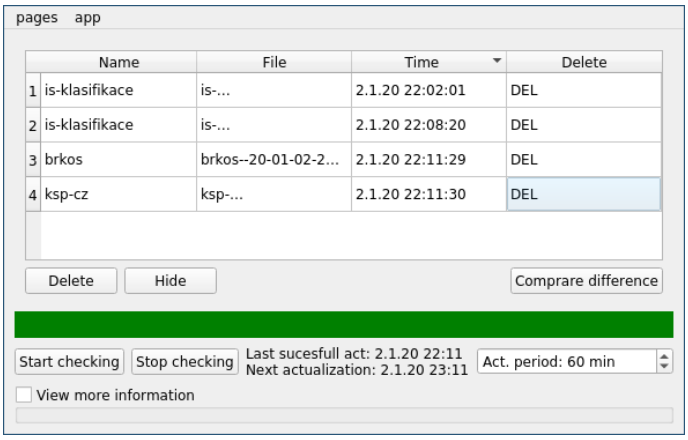
1.1 Úvod do problematiky

Webové stránky se mohou neustále měnit, proto je dobré automaticky monitorovat jejich aktualizace. Tato aplikace umožňuje automatizovat tento problém a tím uživateli ušetřit čas a eliminovat lidský chybový faktor.

Aplikace podporuje různé tolerance při načítání a porovnávání změn stránek. Například na stránce, kde se část neustále mění, mohu tuto část vypustit, nebo přímo porovnávat jen nějaké části nebo lze tolerovat prohazování prvků stránky. Díky práci s cookies je možné navázat i složitější spojení se serverem a provést definovanou sekvenci úkolů (např. přihlásit se a načíst nějaký soukromý obsah). Historie stránek se může ukládat a pak lze v ní vyhledávat a zjišťovat rozdíly mezi verzemi pomocí grafického porovnávání napojeného na uživatelův oblíbený prohlížeč.

1.2 Cíl práce

Cílem této práce je vyvinout funkční aplikaci umožnující zjišťování aktualizací, archivaci a porovnávání webových stránek (případně i jiných dokumentů) a publikovat ji cílovým uživatelům na různých operačních systémech. K aplikaci také bude vypracována rozsáhlá uživatelská i technická dokumentace, která umožní její další vývoj. Umožním tedy dalším programátorům tuto aplikaci pohodlně modifikovat a upravovat dle svých potřeb. Cílem této práce je také aplikaci rozšířit mezi skupinu testovacích uživatelů a použít jejich připomínky a problémy k dalšímu vývoji a stabilizaci aplikace.



Obrázek 1: Základní okno aplikace.

2 Popis fungování a ovládání aplikace

2.1 Instalace

Kompilace ze zdrojového kódu

Způsobem, jak aplikaci nainstalovat na většině používaných operačních systémů, je kompilace ze zdrojového kódu. Zdrojové kódy aktuální stabilní verze je možné stáhnout z `gitlab.com/JiriKalvoda/webupdatingindicator/tree/master`. V případě, že má uživatel zájem o aktuálně nejnovější funkce, je možné použít testovací verzi produktu dostupnou z `gitlab.com/JiriKalvoda/webupdatingindicator/tree/Test`. Soubory lze stáhnout pomocí webového rozhraní a nebo je lze naklonovat s použitím gitu. Aplikaci pak lze zkompilovat za použití Qt knihoven. Nejsnazší způsob je využít aplikace Qt Creator. Pomocí ní stačí otevřít soubor `WebUpdatingIndicator.pro` a v levém rohu aplikace kliknout na tlačítko pro kompilaci. Tímto způsobem by měla vzniknout samostatně spustitelná aplikace, kterou stačí umístit do požadované složky a v ní ji spouštět. Podporovaná by měla být libovolná verze Qt větší než 5.4. Pro vývoj se používá Qt 5.12.5.

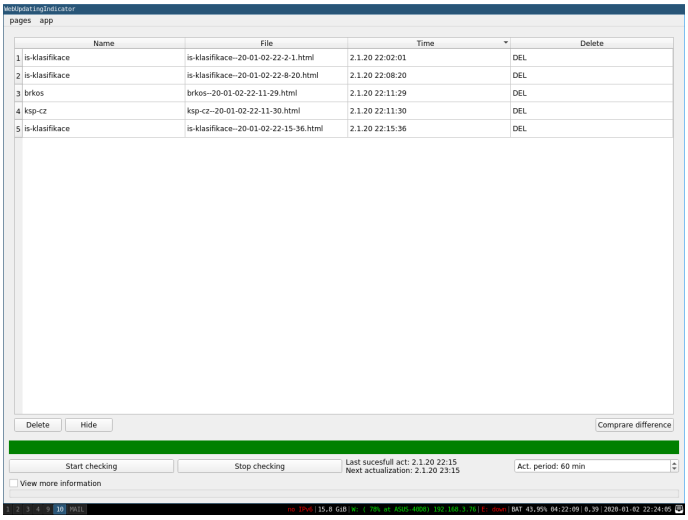
V případě, že uživatel nechce provádět kompilaci ze zdrojového kódu, pro základní architektury a operační systémy je možné využít již zkompilované varianty. Ty jsou dostupné na adrese `gitlab.com/JiriKalvoda/webupdatingindicator-install/tree/master`

Linux

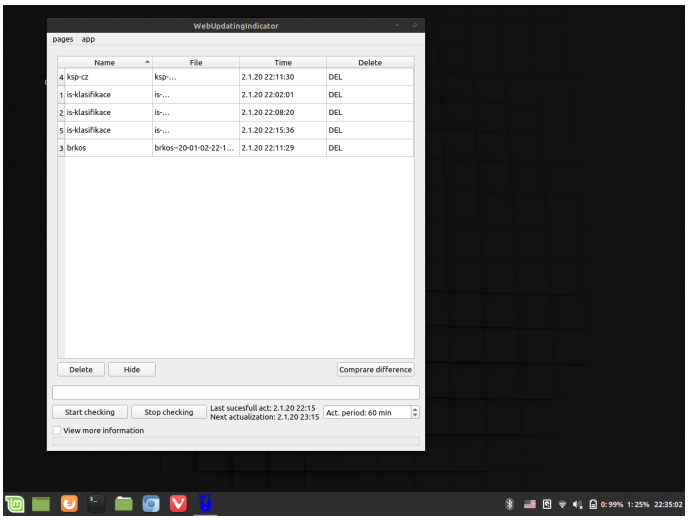
Na operačních systémech postavených na jádře Linuxu stačí pouze stáhnout a rozzipovat složku s programem do uživatelem zvoleného adresáře. Pracovní adresář aplikace je pak ten, ze kterého se aplikace spouští (nemusí tedy být shodný s adresářem, ve kterém je umístěna aplikace). Pro jednodušší spouštění je dobré vytvořit bash script, který bude obsahovat přepnutí polohy do pracovního adresáře a spuštění aplikace. Pro možnost spouštění aplikace z menu či pomocí přímého příkazu je možné tento skript umístit do adresáře `usr/bin`.

V případě užívání správce oken i3 je vhodné nastavit, aby se okna porovnávání stránek zobrazovala jako plovoucí. Toho lze docílit přidáním řádku `for_window [title="WebUpdatingIndicator compare"] floating enable` do konfiguračního souboru i3 umístěného v `~/.config/i3/config`. Pro snazší spouštění aplikace je také vhodné nadefinovat klávesovou zkratku. Případně je možně vyhradit aplikaci speciální pracovní plochu a definovat její spuštění a přepnutí na danou plochu pomocí příkazů (nastaví spuštění na `$mod+Shift,` a zobrazení na `$mod+,`):

```
bindsym $mod+Shift+comma workspace WUI;exec WebUpdatingIndicator.sh
bindsym $mod+comma workspace WUI
```



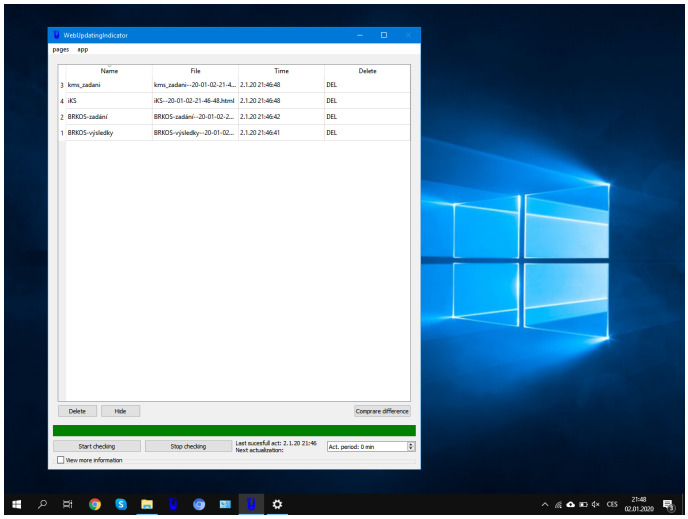
Obrázek 2: Aplikace otevřená v operačním systému Linux Mint 19 se správcem oken i3.



Obrázek 3: Aplikace otevřená v operačním systému Linux Mint 19 se správcem oken Cinnamon.

Windows

Pro Windows existuje alternativní instalační složka. Jejím stažením a rozzipováním do uživatelem zvolené složky vznikne spustitelná aplikace. Je dobré vytvořit zástupce, přes kterého se bude daný program spouštět. Při jeho vytváření se dá zvolit i pracovní adresář. Jako ikonku je možné nastavit `Logo.ico`. Zástupce je pak možné umístit například na plochu nebo do Start menu.



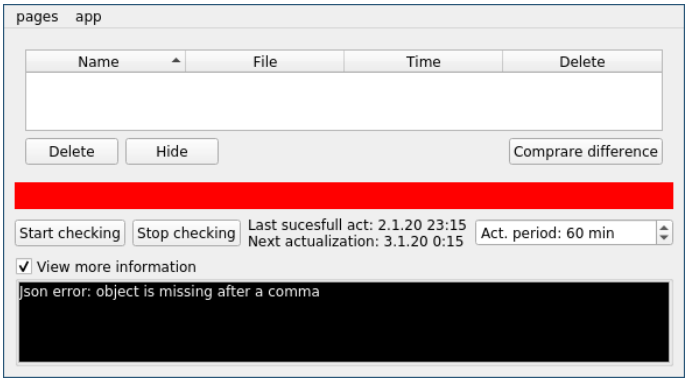
Obrázek 4: Aplikace otevřená v operačním systému Windows 10.

macOS

Na tomto operačním systému je momentálně možná instalace pouze pomocí kompilace ze zdrojového kódu.

2.2 Seznam stránek na kontrolu

Při spuštění si aplikace načte seznam stránek ke kontrole. Ten je obsažen v souboru `pages.json`, který musí být umístěn v adresáři aplikace (respektive v adresáři, kde se aplikace spouští). Soubor musí být validní json. V případě, že soubor neexistuje nebo není validní, aplikace vypíše upozornění. Očekává se, že soubor obsahuje pole struktur. Každá z nich obsahuje informace o jedné stránce, která se má kontrolovat.



Obrázek 5: Chybný vstupní soubor.

Adresa stránky a název

Každou stránku je nutné pojmenovat jednoznačným identifikátorem. Proto je nutné u každé stránky definovat položku označenou klíčem `name`. Toto jméno se pak zobrazuje v seznamu změn a také je nutné při vyhledávání v historii. Také se používá jako identifikátor v databázi i jako identifikátor souborů jednotlivých verzí stránky.

Dále je nutné u každé stránky definovat její umístění na webu, tedy její adresu. K tomu slouží položky `server`, `dir`, `file`. Ovšem není zapotřebí definovat všechny tyto položky. Pod klíčem `server` by měla být definována adresa serveru, na kterém je požadovaná stránka včetně přístupového protokolu. Tedy například `https://is.jaroska.cz` nebo pomocí ip adresy může zápis vypadat následovně: `http://195.178.65.1`. V případě, že není požadovaná stránka na serveru umístěna v jeho kořenové složce, je doporučeno název složky (popřípadě celou cestu několika zanořených složek) umístit do položky `dir`. `file` obsahuje jméno požadovaného souboru včetně přípony. V případě, že je požadován přístup na základní soubor (index) na dané adrese, není potřeba `file` uvádět. Dále zde také mohou být obsaženy parametry stránky, které se předávají metodou `GET`. Stačí je uvézt za otazník. Takový zápis může vypadat následovně: `index.php?akce=42&akcicka=0`.

V případě, že je nutné předat stránce informace pomocí protokolu `POST` (typicky při přihlašování na stránky), je možné v zápisu stránky užít klíče `post`.

Nejjednodušší způsob, jak zjistit tyto informace pro požadovanou stránku, je využít prohlížeč. Většina prohlížečů umožňuje zobrazit informace o navázaném spojení. Odsud stačí potřebné informace jen zkopírovat. V prohlížečích založených na jádře Chromium se lze pomocí klávesy `F12` dostat do vývojářského panelu. V záložce `Network` je možné najít příslušný soubor, jehož hlavičku je třeba použít.

Způsoby kontroly změn

U některých stránek se často mění jejich část, i když se sledovaný obsah nezmění. Pro omezení kontrolování je proto vhodné užít v zápisu položky `diff`. Ta by měla obsahovat speciální strukturu popisující způsob ignorování změn[1]. Tato struktura může obsahovat následující položky:
Klíč `ignore` s libovolnou hodnotou znamená, že stránka se vůbec nebude kontrolovat na změny. Toto je vhodné například když se jedná pouze o přihlašovací stránku, která neobsahuje žádaná data.
Tag `ignoreSector` může obsahovat pole struktur. Každá z nich může obsahovat informace o jednom nutném vynechání pomocí tagů `start`, `end` a `countOfEnd`, které znamenají, že text od `start` po `countOfEnd`-tý výskyt řetězce `end` bude při porovnávání vynechán. Tagy `end` a `countOfEnd` nemusí být uvedeny. V takovém případě je `end` nastaveno na konec řádku a `countOfEnd` na 1.
Obdobným způsobem lze použít tag `onlySector`, který také může obsahovat pole struktur složených ze `start`, `end` a `countOfEnd`. Při použití tohoto tagu budou porovnávány pouze změny v těchto úsecích (budou ignorovány úseky od začátku k prvnímu výskytu, mezi nimi a od posledního na konec).
Další možností je využít klíče `permutation` s libovolnou hodnotou. Jeho použití znamená, že libovolné permutace znaků budou považovány za stejné. Toto je vhodné v případě, že se na stránce některé objekty náhodně prohazují.
Tato omezení porovnávání se provádí ve zde uvedeném pořadí.

Skupiny stránek, přihlášení a práce s cookie

Například v případě, že je nutné se pro získáni informací na nějakou stránku přihlásit, je možné využít v zápisu stránky položku `cookie`. Pomocí ní lze spojit více stránek do skupiny, ve které si stránky mezi sebou ukládají cookies. Stačí pouze u všech stránek vyplnit stejnou hodnotou tagu `cookie`. Když se některou ze stránek nepodaří načíst, v daném průchodu se nebudou načítat ani další stránky ze stejné skupiny.

2.3 Spuštění kontroly, tabulka změn, informační konzole

Po spuštění aplikace se zobrazí hlavní okno. Jelikož aplikace by měla běžet jako služba, okno neobsahuje tlačítko zavřít. V případě, že uživatel skutečné chce uzavřít aplikaci a tím i zabránit dalším automatickým kontrolám, je možné aplikaci ukončit z menu kliknutím na `app` → `quit`.

[1] Případně může obsahovat řetězec `"ignore"`, který má stejný efekt jako `{"ignore":1}`

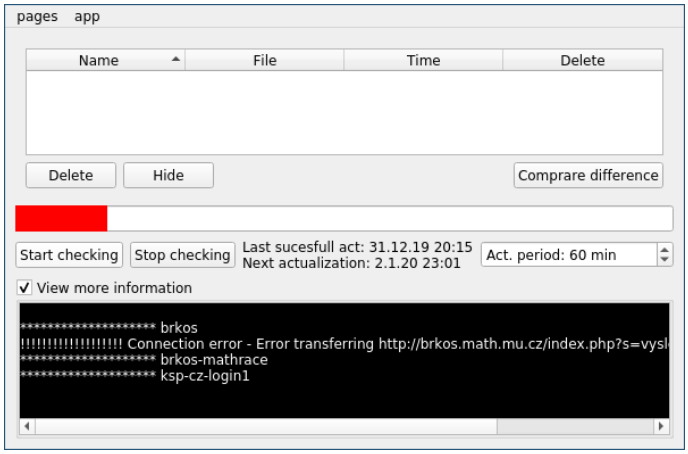
```
[
  {
    "server": "http://brkos.math.muni.cz",
    "dir":"mathrace",
    "file": "index.php",
    "name": "brkos-mathrace",
    "diff":{"ignoreSector":[{"start":"<div id=cas>"}]}
  },
  {
    "server": "https://is.jaroska.cz",
    "file": "login.php",
    "post": "formUsername=HugoKokoska&formPassword=Heslo",
    "name": "is-prihlaseni",
    "cookie": "is.jaroska.cz",
    "diff": "ignore"
  },
  {
    "server": "https://is.jaroska.cz",
    "file": "index.php?akce=650",
    "name": "is-klasifikace",
    "cookie": "is.jaroska.cz",
    "diff": {"permutation":1}
  }
]
```

Obrázek 6: Příklad validního souboru stránek na kontrolu.

Pro manuální spuštění kontroly je možné kliknout na tlačítko `Start checking`. V případě, že již kontrola běží, po kliknutí se ukončí a ihned začne znovu od začátku. Po kliknutí na `Stop checking` se prohledávání neprodleně ukončí.

Při průchodu se nalezené změny ihned zobrazují do tabulky změn. Tedy i v případě, že je prohledávání ukončeno v průběhu, doposud nalezené změny budou uloženy a zobrazeny.

Aktuální stav průchodu je vidět na stavovém baru nad tlačítky. V případě, že bar má červenou barvu, alespoň jedna stránka nebyla při posledním průchodu úspěšně načtena. Po kliknutí na zaškrtávací políčko `View more information` se zobrazí záznam o provedených kontrolách a případně důvod jejich neúspěchu v informační konzoli. Všechny chybové hlášky jsou obsaženy na řádcích uvozených několika vykřičníky. Nejčastěji se může uživatel setkat s těmito chybami:
`Connection error - TIME OUT.` nastane v případě, že načítání stránky bylo moc pomalé a tedy byl překročen časový limit na přístup na jednu stránku.
`Cookie error - cookie not available.` se zobrazí v případě, že nastala chyba při načítání některé z předcházejících stránek ze stejné cookie skupiny.
`Connection error - Network access is disabled.` znamená nedostupnost síťového připojení.
`Connection error - Host not found.` informuje o nedostupnosti daného serveru. S největší pravděpodobností se jedná o chybně zadanou stránku nebo byla přesunutá.



Obrázek 7: Výpis chyby v informační konzoli.

V boxu vpravo od tlačítek je možno nastavit automatické spouštění kontrolování. Do boxu stačí napsat počet minut mezi automatickými kontrolami. Automatické kontroly lze vypnout napsáním `0` do políčka. Datum a čas poslední úspěšné (tedy takové, ve které se načetly všechny stránky) kontroly je vidět mezi tlačítky a boxem automatické aktualizace. V tomto místě je také vidět čas příští plánované kontroly.

Informace o poslední kontrole a periodě automatické kontroly si aplikace ukládá do souboru databáze v pracovním adresáři aplikace. Při spuštění si tento soubor načte (pokud existuje). Nastavení automatické kontroly tedy vydrží i vypnutí a zapnutí aplikace. V případě, že kontrola měla proběhnout v momentě, kdy byla aplikace vypnutá, proběhne neprodleně po jejím spuštění.

Když aplikace zaregistruje změnu stránky a není momentálně aktivní, bude se na změnu snažit upozornit. Provedení této funkce je částečně závislé na platformě. Aplikace pošle požadavek na vyskočení do popředí, což ve většině případů znamená červené rozblikání dané aplikace případně ikony dané pracovní plochy.

Všechny nalezené změny se zobrazují v tabulce změn umístěné v horní části hlavního okna aplikace. O každé změně se zobrazí řádek obsahující jméno stránky, čas detekování změny a jméno souboru, v němž je uložena aktuální verze. V případě, že uživatel chce data setřídit podle některé z těchto položek, může tak učinit kliknutím na hlavičku daného sloupce tabulky. Kliknutím na jméno souboru se ve výchozím prohlížeči otevře daná verze stránky. Aby bylo zajištěno správné fungovaní zobrazení, jsou všechny relativní odkazy (v rámci serveru) přepsány na absolutní. Před všechny odkazy v atributech `href` a `src`, které neobsahují absolutní cestu je tedy doplněn název serveru a složky. Díky tomu se při kontrole načítá pouze samostatná stránka, ale při zobrazení se načtou i obrázky, styly a další odkazované elementy.

V případě, že už si uživatel danou změnu prohlédl, kliknutím na buňku v sloupci `Delete` ji může z tabulky změn odstranit. Tímto odstraněním nedojde k smazání záznamu o změně ani odstranění souboru s danou verzí stránky. Změna se již nebude zobrazovat v tabulce změn. Jiným způsobem odstranění je vybrat jeden nebo několik řádků a pak kliknout na tlačítko `Hide` těsně pod tabulkou.

Vybráním řádků a kliknutím na tlačítko `Delete` dojde k smazání vybraných záznamů změn včetně souborů obsahujících dané verze stránek.

Všechny záznamy o změnách se ukládají do databáze a při spuštění aplikace se načtou všechny neskryté záznamy.

2.4 Historie změn a její procházení

Zobrazení již odstraněných záznamů z tabulky změn lze pomocí nástroje historie. Otevřít ho lze pomocí menu v hlavním okně kliknutím na `pages` → `history`. Po kliknutí se otevře další okno obsahující formulář na dotaz a pod ním prázdnou tabulku se stejnou strukturou jako původní tabulka změn.

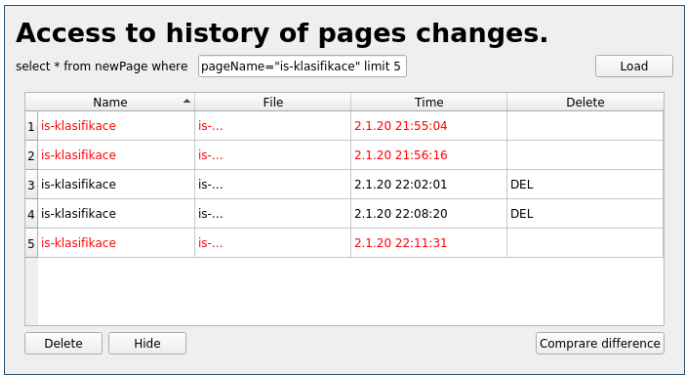
Dotazy do historie se zadávají formou dotazu do databáze. Do políčka lze doplnit podmínky hledání a po kliknutí na tlačítko `Load` se načte tabulka odpovídajících záznamů. Toto načtení je jednorázové, tedy v případě, že se například vygeneruje další záznam, tabulka historie se sama neobnoví. Obnovení lze vynutit opětovným stiskem tlačítka `Load`. Jedinou výjimkou jsou změny provedené v daném okně his-

torie – smazání a schování. Pod tabulkou historie jsou stejné ovládací prvky jako pod tabulkou změn. Tlačítko `Delete` smaže vybrané záznamy, ovšem tlačítko `Hide` funguje pouze na zatím neodstraněné záznamy z tabulky změn a to tak, že je schová, ovšem pouze v tabulce změn. V tabulce historie jsou schované záznamy zobrazeny červeně bez vyplněného sloupce `Delete`.

Do dotazů je možné psát libovolný SQL dotaz. Zdrojová tabulka by měla být `newPage` obsahující informace o všech změnách. Její struktura je:

- `id INTEGER PRIMARY KEY AUTOINCREMENT` – jednoznačný identifikátor záznamu
- `pageName VARCHAR` – identifikátor stránky (podle json zdrojového souboru)
- `time DATETIME` – čas zachycení dané změny
- `fileName VARCHAR` – jméno souboru obsahujícího danou změnu
- `del bit` – obsahuje 1 v případě, že stránka byla odstraněna z tabulky změn.

Dotaz by ovšem měl vracet tabulku o stejných sloupcích jako má `newPage`, aby mohl být správně graficky vykreslen. V opačném případě je chování aplikace nedefinované. Databáze je implementovaná pomocí SQLite. Podrobná dokumentace je uvedena na adrese `https://www.sqlite.org/lang_select.html`.



Obrázek 8: Okno přístupu do historie.

2.5 Grafické porovnávání verzí stránek

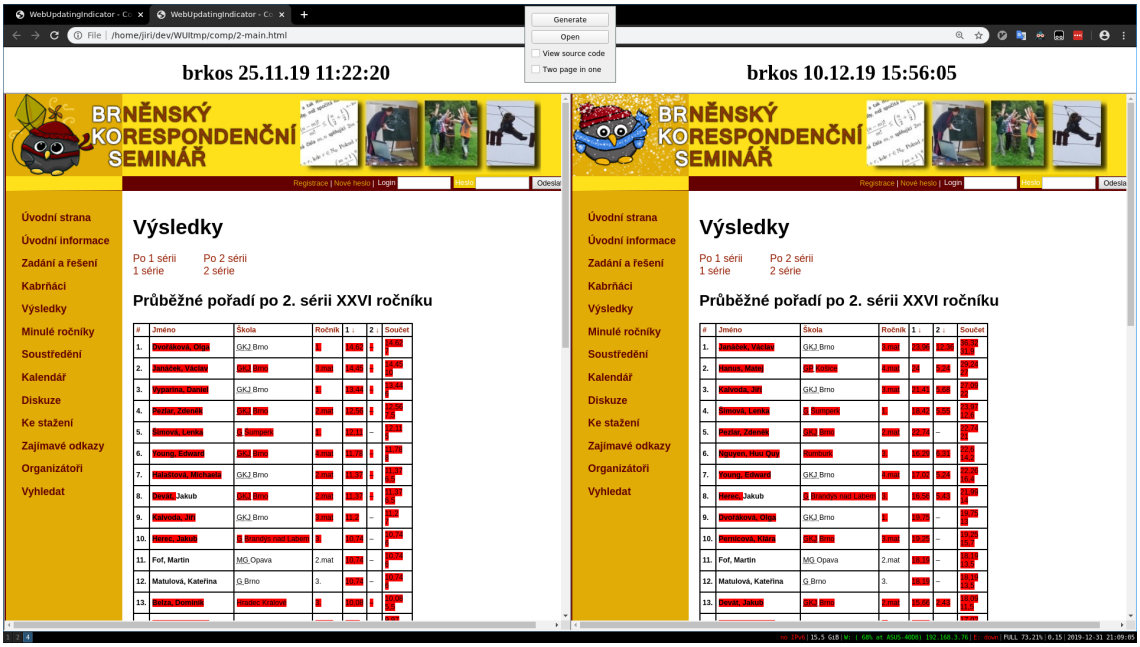
V případě, že uživatel chce vidět rozdíly mezi dvěma verzemi stránek (popřípadě i mezi dvěma stránkami) je možné využít nastroje porovnání. Tento nástroj lze spustit kliknutím na tlačítko `Compare difference` pod tabulkou změn nebo pod tabulkou historie. Před kliknutím je potřeba vybrat jeden nebo dva řádky v příslušné

tabulce. V případě, že jsou vybrány dva záznamy, provede se porovnání jejich souborů. V případě, že je vybrána pouze jedna stránka, provede se pouze porovnání mezi zvolenou verzí zvolené stránky a předchozí verzí dané stránky. Zobrazí se tedy rozdíly, které byly detekovány v daný moment.

Při použití porovnání se ve výchozím prohlížeči otevře stránka obsahující dva rámy obsahující obě porovnávané stránky. Dále se také otevře ovládací okno. V ovládacím okně je možno nastavit dva přepínače:
`View source code`, při jeho zaškrtnutí se v prohlížeči zobrazí místo stránek jejich zdrojové kódy. Přepínač `Two page in one` místo dvou rámů zobrazí jen jeden, do kterého zobrazí sjednocení obou stránek. Tato funkce je zatím pouze v testovací verzi a může dojít k nesprávnému propojení stránek. Stabilní je pouze při propojení se zobrazením zdrojového kódu. Pro vytvoření grafického výstupu do prohlížeče dle zadaných kritérií je nutné zmáčknout tlačítko `Generate`. K znovuotevření stránky v prohlížeči slouží tlačítko `Open`.

V jakékoliv verzi porovnávání se graficky zobrazují změny. V režimu se dvěma rámy se v každém z nich zobrazí přebývající text s červeným zvýrazněním. V případě režimu jednoho rámu se nový text zobrazí se zeleným podbarvením a odstraněný text červeně. V režimu zdrojového kódu se změny zobrazí na čemkoliv – textu i html elementu či jiném prvku stránky. Když je ovšem porovnání formou stránek, jsou zvýrazněny pouze změny v čistém textu.

Porovnávání stránek se provádí na úrovni slov a tagů. Aplikace se snaží najít jejich zobrazení na sebe s největším překryvem, tedy nejmenším počtem rozdílů.



Obrázek 9: Porovnávání stránek otevřené v aplikaci Chromium.

3 Implementace aplikace

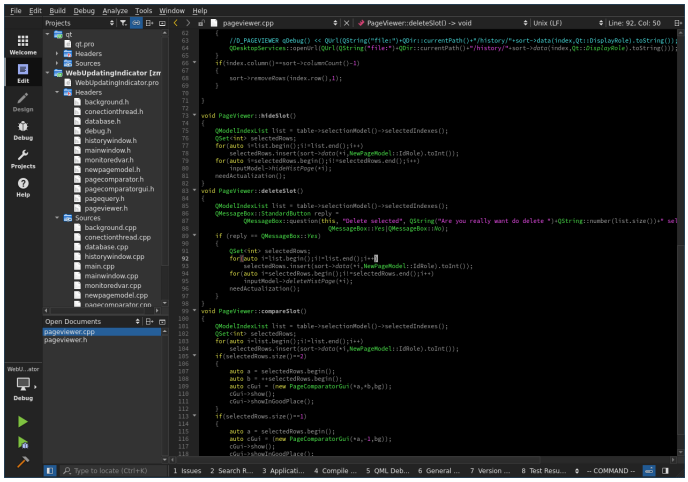
3.1 Použitý software

Qt

Qt je multiplatformní knihovna pro c++ umožnující nejen tvorbu grafických aplikací, ale i práci se soubory, obrázky či sítovým připojením. Byla nedílnou součástí vyvíjené aplikace. Knihovna podporuje běh samotného jádra aplikace – načítání webových souborů a také se stará o grafické zobrazování dat. Její funkce jsou tedy použité při většině výpočtů aplikace.

Qt Creator

Qt Creator je vývojové prostředí určené primárně pro vývoj aplikací využívajících knihovny Qt. Programátorovi nabízí zvýrazňování syntaxe a doplňování názvů. Umožňuje také kompilování, spouštění a debugování aplikace pomocí jednoduše ovladatelných nástrojů. Dále také umožňuje ovládání gitu pomocí zabudovaných nástrojů. V tomto prostředí vznikla většina zdrojových kódů aplikace.



Obrázek 10: Vývojové prostředí Qt Creator

Git

Git je systém na správu verzí využitý při vývoji této aplikace. Umožňuje oddělení jednotlivých úkonů při vývoji, jejich zdokumentování a možnost zobrazení jejich historie. Dále nabízí pomocí vzdálených repositářů jednoduchý způsob, jak aplikaci nahrát na web a zpřístupnit uživatelům.

3.2 Objektový model, rozdělení problému

Samotná knihovna Qt je psaná objektově, a proto je vhodné navázat v objektové orientovaném programování i při vývoji. Program jsem tedy rozčlenil do několika vzájemně provázaných tříd. Pro každou třídu existují dva soubory obsahující hlavičky a samotný program.

Program jsem dále rozčlenil na téměř samostatné části. Třídy pozadí se starají o samotné načítání a zjišťování rozdílů na stránkách, práci s databází a porovnávání verzí stránek. Ostatní třídy se starají o správné předávání dat uživateli pomocí graficky vykreslených oken a také přijímají pokyny od uživatele a propagují je do příslušných tříd. Díky tomu je možné poměrně jednoduše přepsat celé uživatelské rozhraní například do podoby konzolové aplikace s využitím stávájících funkcí pozadí.

3.3 Pozadí aplikace

Hlavní třídou aplikace je `Background`. Ta se stará o inicializaci a propojení ostatních tříd pozadí. Dále si také ukládá a načítá seznam stránek na kontrolu.

Třída `ConnectionThread` se stará o samotné načítání a hledání rozdílů ve stránkách. Aby byl zajištěn hladký chod aplikace, je tato část programu vykonávána jako speciální vlákno. Díky tomu lze při načítání libovolně pracovat s aplikací. V této třídě je také implementováno ignorování částí změn na základě zadaných podmínek.

Pro usnadnění přístupu do databáze slouží třída `Database`. Ta přistupuje do souboru `database.db` pomocí knihovny SQLite. V databázi jsou uloženy záznamy o všech verzích stránek a také uživatelské nastavení.

O ukládání nových změn na stránce se stará `NewPageModel`. Tento tabulkový model je pak přímo napojen na zobrazování změn.

Třída `PageQuery` založená na `NewPageModel` obsahuje tabulky dotazu z databáze na historii změn.

Porovnávání verzí stránek

`PageComparator` umožňuje porovnávání verzí stránek. Jeho úkolem je najít nejlepší napojení stránek na sebe a následné vytvoření souborů, které zobrazují změny v prohlížeči.

Obě dvě verze stránky si rozdělí na jednotlivá slova a tagy. S nimi se dále pracuje jako se znaky. Poté hledá nejdelší shodný podřetězec. Na to využívá dynamického programování. Pro každou dvojici prefixů z obou řetězců si spočítá délku nejdelšího shodného podřetězce. K výpočtu ovšem využívá toho, že tuto hodnotu již zná pro kratší prefixy. Když je poslední slovo v obou prefixech stejné, tak oproti dvojici, v

niž jsou oba prefixy o jedno slovo kratší, může být délka podřetězce o jedna delší (toto odpovídá variantě, že poslední slovo přibude ve shodném podřetězci). Délka podřetězce ovšem může být stejná jako délka podřetězce v dvojicích, které mají jeden sufix o jedna kratší (toto odpovídá variantě, že shodný podřetězec zůstane stejný). Z těchto variant vybere maximum.

Poté již není problém zrekonstruovat tvar podřetězce. Stačí projít takto vytvořenou tabulku od prefixů obsahujících celé řetězce směrem odkud vzniklo maximum v dané buňce.

Při tomto průchodu si poznačí, které části vstupu jsou součástí shodného podřetězce a které nikoliv. Části, které nejsou součástí podřetězce pak zobrazuje jako změny, tedy je barevně zvýrazní obalením je do tagu `span`.

3.4 Grafické uživatelské rozhraní

Hlavní okno aplikace je implementováno pomocí třídy `MainWindow`. Toto je hlavní třída celého projektu. Z ní se teprve spouští inicializace pozadí aplikace včetně `Background`.

O správné vykreslování tabulek změn stránek včetně napojení jejich ovládacích prvků se stará `PageViewer`. `PageComparatorGUI` se stará o okno zobrazené při porovnávání stránek a jeho napojení na `PageComparator`. Okno historie změn zobrazuje `HistoryWindow`.

4 Seznam použité literatury

[The Qt Company, 2019] The Qt Company, 2019. Qt 5.12. *Qt* [online]. The Qt Company [Cit. 31.11.2019]. Dostupné z: `https://doc.qt.io/qt-5.12/index.html`.

[Hipp, 2000] Hipp, D. R. 2000. Documentation. *SQLite* [online]. SQLite [Cit. 31.11.2019]. Dostupné z: `https://www.sqlite.org/docs.html`.

[Watzke, 2010] Watzke, D. 2010. Seriál: Qt 4 - psaní grafických programů. *AbcLinuxu* [online]. AbcLinuxu [Cit. 31.11.2019]. Dostupné z: `http://www.abclinuxu.cz/serialy/qt-4-psani-grafickych-programu`.

[Chacon, 2009] Chacon, S. *Pro Git*. Praha: CZ.NIC, 2009, ISBN: 978-80-904248-1-4..