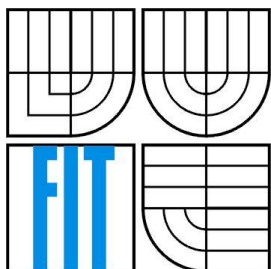


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

APLIKACE PRO TVŮRČÍ PSANÍ

AN APPLICATION FOR CREATIVE WRITING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PATRIK MAGDOLEN

VEDOUCÍ PRÁCE
SUPERVISOR

ING. MARTIN KRČMA

BRNO 2016

Abstrakt

Předmětem této bakalářské práce je návrh a implementace aplikace, která má za úkol umožnit uživateli pohodlné psaní prózy a nabídnout užitečné funkce jako zpráva kapitol, postav a poznámek. Dokáže automaticky rozpoznat názvy kapitol i jména postav v textu. Aplikace také nabízí další funkce napomáhající práci spisovatele včetně generátoru jmen a možností zadat cíle. Aplikace disponuje vícejazyčným multiplatformním moderním a personalizovatelným GUI, které dbá na estetičnost, praktičnost a přehlednost. Výsledky práce uživatelů se exportují do výstupního souboru, kterému je možné specifikovat formát.

Abstract

This Bachelor's thesis deals with the design and the implementation of the application, supposed to support comfortable writing of prose and offer helpful features like a management of chapters, characters and notes. It is able to automatically detect titles of chapters and names of characters in text. Application also offers other functions assisting the writer, including a name generator and an ability to set goals. The application includes multilingual multiplatform modern and personalizable GUI, which focuses on aesthetic, practicality and lucidity. Results of a usage of the application are exported into output file of various types.

Klíčová slova

Tvůrčí psaní, textový editor, generátor jmen, Java, JavaFX, Font Awesome, EPUB, PDF

Keywords

Creative writing, text editor, name generator, Java, JavaFX, Font Awesome, EPUB, PDF

Citace

Magdolen Patrik: Aplikace pro tvůrci psaní, bakalářská práce, Brno, FIT VUT v Brně, 2016

Aplikácia pre tvorivé písanie

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Martina Krčmy.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Patrik Magdolen

18. května 2016

Poděkování

Rád bych poděkoval Ing. Martinovi Krčmovi, za vedení bakalářské práce, odbornou pomoc, cenné připomínky a poskytnuté konzultace.

© Patrik Magdolen, 2016

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Úvod	2
1 Požiadavky a existujúce riešenia.....	3
1.1 Existujúce riešenia	3
1.1.1 Word	3
1.1.2 Writer.....	3
1.1.3 Ulysses.....	4
1.1.4 Scrivener	4
1.1.5 FocusWriter	5
1.1.6 Q10	6
1.1.7 OmmWriter.....	6
1.1.8 WriteMonkey.....	7
1.2 Zhrnutie poznatkov	8
2 Návrh aplikácie	9
2.1 Štúdium užívateľov.....	9
2.2 Špecifikácia.....	9
2.2.1 Kľúčové funkcie	10
2.3 Návrh užívateľského rozhrania.....	10
2.3.1 Úvodné okno.....	10
2.3.2 Rozhranie.....	11
2.4 Architektúra aplikácie.....	14
2.4.1 Dostupné technológie	14
2.4.2 Vybrané API.....	15
3 Implementácia.....	16
3.1 JavaFX framework.....	16
3.2 Implementovaný model	16
3.2.1 Trieda Configuration.....	17
3.2.2 Trieda Controller	17
3.2.3 Trieda EditorController.....	18
3.2.4 Správa pravého panelu.....	19
3.2.5 Triedy spravujúce okná.....	23
3.2.6 Triedy pre prácu so súbormi	24
3.2.7 Generátor mien	24
3.2.8 Animácie.....	25
3.3 Použité knižnice.....	25
3.3.1 RichTextFX	25
3.3.2 Epublib.....	25
3.3.3 Modul iText	25
3.3.4 Ikony Font Awesome.....	26
3.4 GUI.....	26
3.4.1 Rozhranie aplikácie.....	27
3.4.2 Okná aplikácie	28
4 Testovanie	30
4.1 Testovanie použiteľnosti.....	30
4.2 Výsledky testovania.....	30
5 Záver	32

Úvod

Krátko po rozmachu informačných technológií bolo zrejme, že éra písacích strojov skončila. Spisovatelia po celom svete teda rýchlo zaznamenali prechod na túto technológiu, nakoľko im neporovnateľne uľahčovala prácu. Vďaka neustálemu technickému vývoju sa zvyšoval výkon počítačov, čo umožnilo pohodlnejšiu prácu ako aj značné rozšírenie dostupných funkcií. Softvéru, ktorý spracováva text podľa požiadaviek užívateľa sa hovorí textový procesor a v súčasnosti ich existuje obrovské množstvo. Ich cena však nebýva najnižšia. Stali sa komplexnými, zahŕňujú množstvo funkcií, ktoré spisovateľ vôbec nevyužije. Iba malá časť z nich je však zameraná na tvorivé písanie. Tento problém sa ešte stupňuje, keď porovnáme podporu takýchto aplikácií pre českú a slovenskú lokalizáciu.

Hlavným cieľom tejto bakalárskej práce je vytvoriť práve takú aplikáciu, ktorá by obsahovala základné funkcie textového procesora a umožnila užívateľovi pohodlné písanie prózy. Za týmto účelom bude mať k dispozícii správu kapitol a možnosť spravovať postavy, ktoré budú v texte automaticky rozpoznávané. Nesmie tiež chýbať správa poznámok. Aplikácia bude rozšírená o generátor mien a niekoľko ďalších funkcií, ktoré napomôžu práci spisovateľa. Nakoľko je nutná lokalizácia, bude aplikácia dostupná vo viacerých jazykoch. Aplikácia tiež musí byť navrhnutá moderne a prakticky tak, aby spisovateľovi pri práci vytvorila nerušivé prostredie a to na rôznych platformách. Samozrejmosťou je aj export výslednej práce do rôzneho formátu.

Práca je rozčlenená do štyroch kapitol popisujúcich uvedenú problematiku a návrhy riešenia pre daný problém. V kapitole 1 sa nachádza prieskum obdobných aplikácií na trhu a rozoberá ich výhody a nevýhody pre tvorivé písanie. Kapitola 2 je zameraná najmä na návrh aplikácie, teda zobrazovanie jednotlivých interakcií, ktoré sa poskytujú užívateľovi a proces návrhu grafického užívateľského rozhrania (ďalej už iba GUI). Popisuje aj vybranú technológiu na ktorej bude aplikácia vyvíjaná a jej prvky z pohľadu vývojára. Kapitola 3 je venovaná implementácii aplikácie, detailnému popisu problémov a ich riešení, vrátane použitých knižníc. Kapitola 4 je zameraná na oblasť testovania, spracovanie výsledkov a zmien užívateľského rozhrania na základe spätných väzieb užívateľov. V kapitole 5 sa nachádza záver práce a stručné zhodnotenie výsledkov.

1 Požiadavky a existujúce riešenia

Aplikácie pre tvorivé písanie nie sú novinkou a preto dnešný trh ponúka hneď niekoľko. Táto kapitola podrobne rozoberá a analyzuje najpopulárnejšie už existujúce riešenia. Sú tu rozobrané ich výhody a nevýhody, ktoré následne ovplyvnia aj návrh vyvíjanej aplikácie. Ďalej sú zmienené funkcie, ktoré inšpirovali vývoj a taktiež bližšie špecifikované požiadavky na výslednú aplikáciu. Záver kapitoly obsahuje zhrnutie uvedených poznatkov a odôvodnenie vývoja aplikácie.

1.1 Existujúce riešenia

Pri hľadaní podobných riešení nebola nájdená žiadna aplikácia, ktorá by splňovala všetky požiadavky. Najpopulárnejšie aplikácie disponovali vyžadovanou funkcionalitou avšak kameňom úrazu pre nich bola nezanedbateľná cena či absencia lokalizácie. Analýza sa zaoberá tiež ich cenou a dostupnosťou na rôznych platformách ako aj zhodnotením dizajnu a celkového nadobudnutého dojmu pri práci.

1.1.1 Word

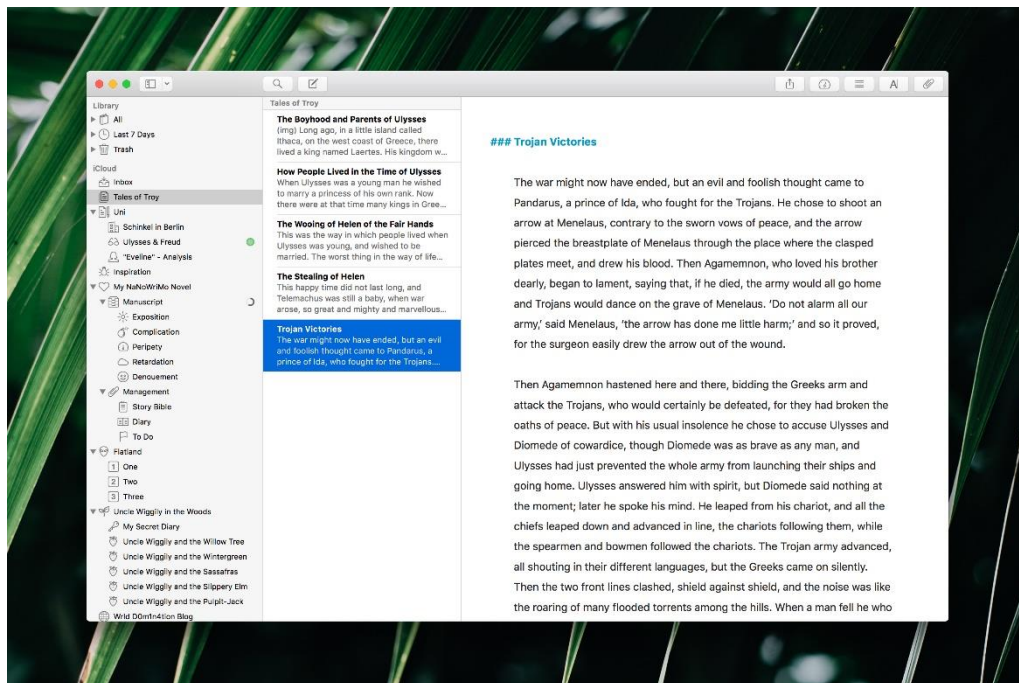
Komplexný textový procesor od firmy Microsoft, ktorý určite netreba predstavovať. Dnešnému trhu jednoznačne dominuje a stal sa štandardom, čo si vyslúžil hlavne vďaka svojej rozsiahlej funkcionalite. Umožňuje takmer neobmedzenú manipuláciu s textom, pokročilé formátovanie, automatické generovanie obsahu dokumentov, cloudové úložisko a pod. Disponuje príjemným a prepracovaným GUI, no vzhľadom na komplexnosť programu, určite nie prehľadným. Ponúka aj režim s minimalizáciou prvkov, no napriek tomu sa u spisovateľov veľkej popularite neteší. Word, ako súčasť kancelárskeho balíka Office, ktorého najnovšia verzia je Office 2016 má cenu v závislosti od verzie, od 189,99 Kč mesačne. Je možné zakúpiť trvalú verziu za 4 099,00 Kč. Dostupnosť je primárne pre operačný systém Microsoft Windows, no taktiež existuje verzia aj pre Mac OS. Pre autorov tvorivého písania však obsahuje príliš veľa nepotrebných funkcií a preto skôr siahnu po inej prívetivejšej voľbe. [1]

1.1.2 Writer

Obdoba Microsoft Word je publikovaný ako slobodný softvér s licenciou LGPLv3 na operačných systémoch Microsoft Windows, systémoch typu Unix a Mac OS. Opäť je to komplexný textový procesor, s mierne obmedzenejším rozsahom funkcií, od firmy Microsoft. Software je k dispozícii ako kancelársky balík OpenOffice od The Apache Software Foundation, tak aj LibreOffice od The Document Foundation, no líšia sa len v detailoch. Obrovskou výhodou je dostupnosť zdarma, aj keď za cenu značne menšieho rozsahu funkcií. Zrejme najväčším problémom je však nedokonalá kompatibilita s inými textovými procesormi, hlavne dominante trhu Microsoft Word. Taktiež sa však jedná o univerzálny textový procesor a teda jeho užívateľské prostredie určite nie je navrhnuté ako nerušivé, čoho následkom sa u spisovateľov veľkej popularite taktiež neteší. [2][3]

1.1.3 Ulysses

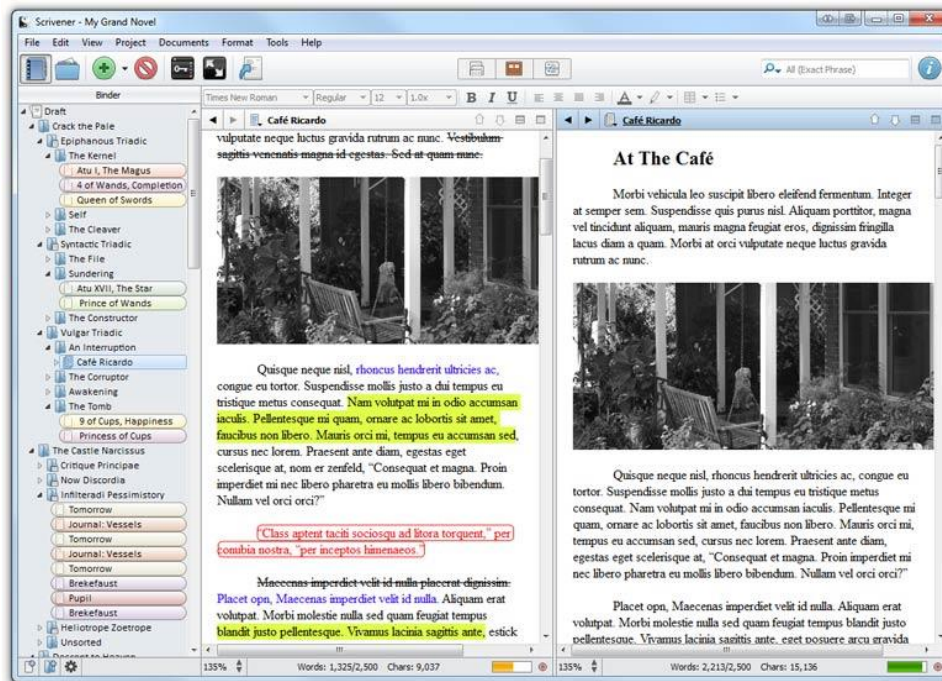
Táto aplikácia už nepatrí medzi univerzálne textové procesory. Je to editor priamo navrhnutý pre spisovateľov. Na prvý pohľad zaujme elegantné a užívateľsky prívetivé GUI a to aj vďaka režimu bez rušivých prvkov, prehľadnej organizácii dokumentov, kapitol a funkciou rozdelenia textov do tzv. „sheetov“, uľahčujúc tak ich záverečnú editáciu či reorganizáciu pomocou funkcie ťahaj a pusti. Ďalšou jeho výhodou je systém Markdown [4], značkovací jazyk využívajúc prefix „##“ pre označenie nadpisu, prípadne jeho opakovanie pre nadpisy nižšej úrovne, vďaka čomu užívateľ nemusí pustiť prsty z klávesnice. Poznámky autora, štatistiky, časovače a určovanie písacích cieľov sú samozrejmosťou. Napriek tomu všetkému si zachováva jednoduché a prehľadné užívateľské rozhranie s možnosťou prispôsobenia pomocou motívov. Jedná sa o jednu z najpopulárnejších aplikácií na trhu pre autorov tvorivého písania. Každý autor určite súčasne ocení aj pravidelné zálohovanie ich práce na vzdialených serveroch. Nevýhodou je dispozícia výhradne na platformu Mac a to za cenu 44,99 €. Ďalšou nevýhodou je nedostupná lokalizácia v češtine ako aj slovenčine. [5]



Obrázok 1.1: Rozhranie aplikácie Ulysses [6]

1.1.4 Scrivener

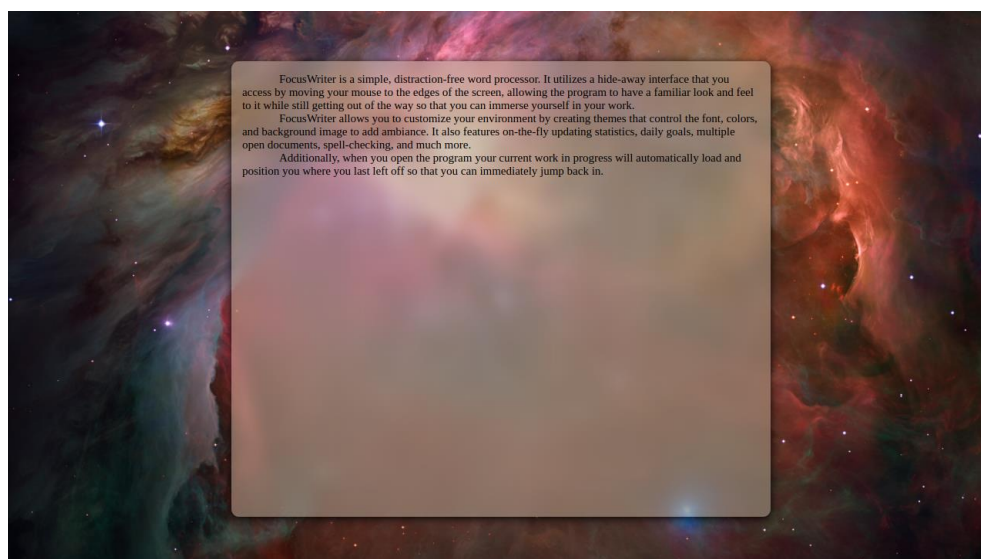
Komplexný program označený ako písacie štúdio. Je navrhnutý pre všetky druhy tvorivého písania, vhodný tak pre amatérov, ako aj profesionálnych žurnalistov, scenáristov či spisovateľov. Disponuje správou kapitol, nástenkou poznámok, možnosťou práce s niekoľkými sekciami textu súčasne, prehľadom štatistík, zadávaním cieľov a množstvom ďalších funkcií. Samozrejmosťou je aj nerušivý režim, kde užívateľ a všetky tieto funkcie nezaťažujú a môže sa sústrediť na písanie. Ponúka tiež management dokumentov, poznámok a obsahuje kolekciu šablón pre rôzne scenáre. Softwarové GUI však kvôli zmeneným funkciám výrazne stráca prehľadnosť a pre tvorivé písanie nemusí byť najvhodnejšie. Výhodou je pokrytie na rôzne operačné systémy: Microsoft Windows, Mac OS X aj systémy typu Unix, avšak jeho cena je 40,00 \$. Napriek svojmu rozšíreniu zostáva nedostupný v českej a slovenskej lokalizácii. [7]



Obrázok 1.2: Rozhranie aplikácie Scrivener [8]

1.1.5 FocusWriter

Jednoduchý textový procesor so základnými funkciami pre písanie. Minimalizuje počet rušivých elementov, automaticky skrýva nepoužívané prvky interface, čo v kombinácii s režimom na celú obrazovku vytvára pohodlné prostredie pre tvorbu. Obsahuje časovače, spisovateľské ciele a napriek svojej jednoduchosti aj kontrolu pravopisu. GUI aplikácie si zachováva prehľadnosť, no určite nepatrí medzi najelegantnejšie. Nevýhodami sú obmedzené možnosti prispôsobenia. Užívateľ nemôže modifikovať základné nastavenia, akými sú automatické skrytie prvkov GUI či zmenu typu písma ako aj obmedzený výber motívov. Veľkou výhodou je však to, že je voľne dostupný na všetky operačné systémy – Windows, Mac OS X aj Linux, vrátane českej a slovenskej lokalizácie. [9]



Obrázok 1.3: Rozhranie aplikácie FocusWriter [10]

1.1.6 Q10

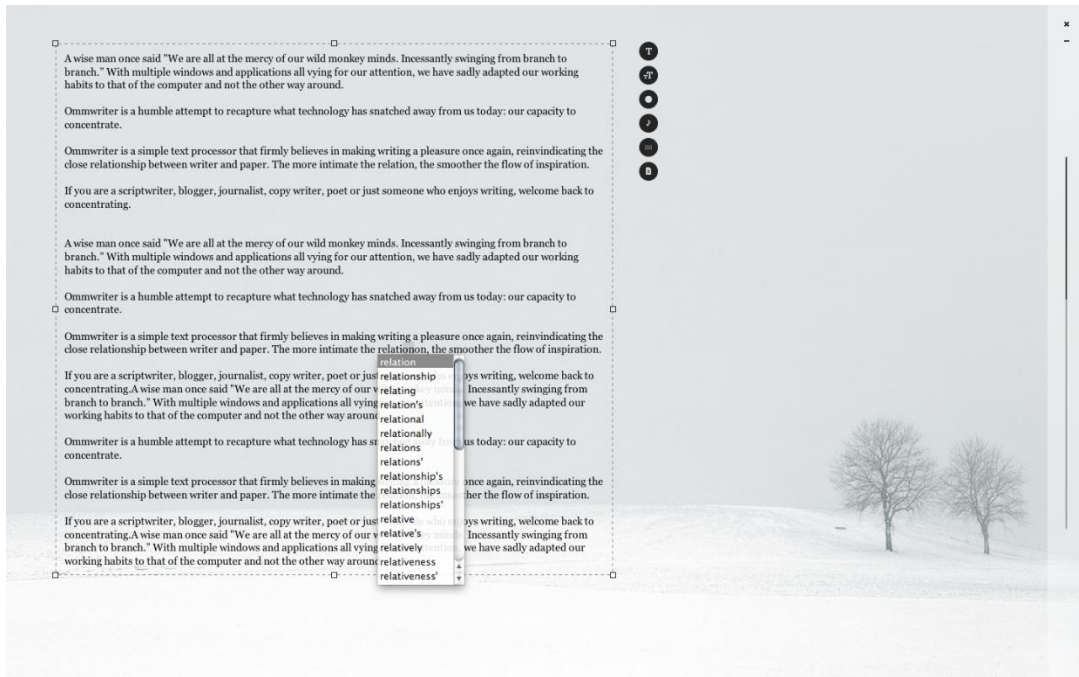
Táto aplikácia sa zameriava na minimalizáciu rušivých prvkov, čo dovedla do extrému a to povinným režimom na celú obrazovku bez akýchkoľvek prvkov interface. Tým vytvára absolútne nerušivé prostredie s jednoduchým a čiastočne prispôbitelným GUI. Obsahuje však najpotrebnejšie funkcie editora, avšak užívateľ k nim má prístup výhradne pomocou klávesových skratiek. Tento fakt z neho určite nerobí vhodnú voľbu pre menej technicky zdatných užívateľov. Napriek tomu si však drží základnú verných užívateľov k čomu prispieva aj fakt, že je voľne dostupný ale iba na operačnom systéme Microsoft Windows a iba v anglickom jazyku. [11]



Obrázok 1.4: Rozhranie aplikácie Q10 [12]

1.1.7 OmmWriter

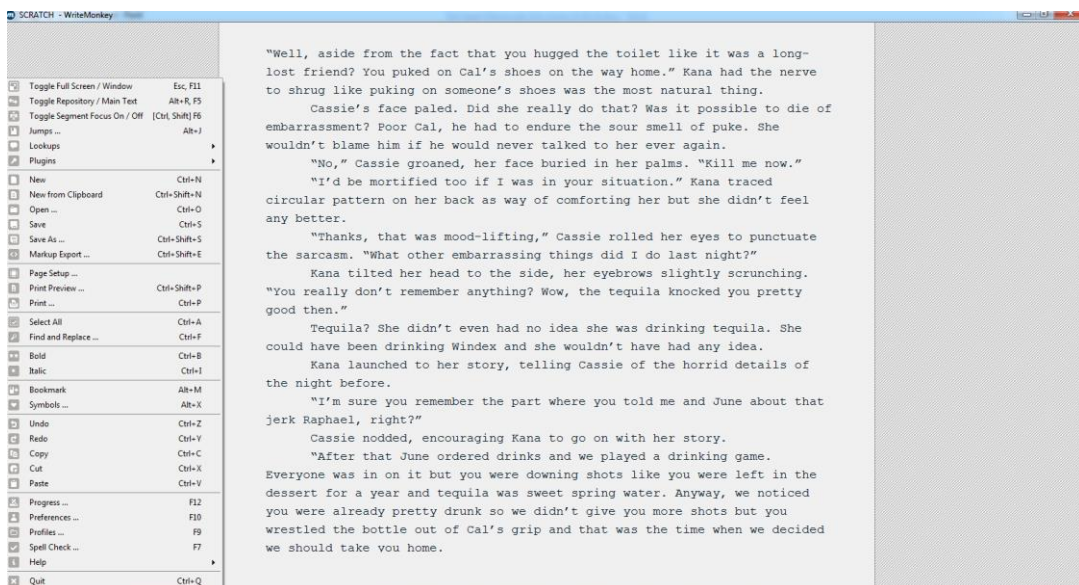
Ide o veľmi prehľadnú a nerušivú aplikáciu, pracujúcu tiež iba v režime na celú obrazovku. Obsahuje základnú sadu funkcií nevyhnutnú pre každý editor, vrátane správy kapitol či možnosti zadávania cieľov. Disponuje prepracovaným GUI, ktoré pôsobí moderne a elegantne avšak jeho prispôbenie je značne obmedzené. Čím ale aplikácia vyniká je dispozícia hudby, ktorá je zameraná na zvýšenie koncentrácie pri písaní. Celkovo tak vytvára veľmi pohodlné prostredie pre písanie. Je voľne dostupná na operačné systémy Mac OS a Microsoft Windows, avšak pri neplatenej verzii nie sú k dispozícii všetky funkcie. Tie sa postupne pridávajú podľa ponúkutej sumy, pričom tá sa pohybuje v rozmedzí niekoľkých dolárov či desiatok dolárov. Opäť však chýba česká a slovenská lokalizácia. [13]



Obrázok 1.5: Rozhranie aplikácie OmmWriter [12]

1.1.8 WriteMonkey

Ďalší jednoduchý, no elegantný textový procesor bez akýchkoľvek prvkov interface. Všetky jeho funkcie sú k dispozícii v kontextovom menu po kliknutí pravým tlačidlom myši. Hoci GUI aplikácia nepatrí medzi najelegantnejšie, vytvára nerušivé prostredie pre prácu. Umožňuje rozsiahle možnosti prispôsobenia, ktoré ale vyžadujú technicky zdatnejšieho užívateľa. K množstvu funkcií patrí správa kapitol a cieľov, ale aj kontrola pravopisu. Ponúka aj českú lokalizáciu. Opäť platí, že je voľne dostupný, ale nie všetky funkcie sú k dispozícii pri neplatenej verzii. Dostupný je však iba na operačnom systéme Microsoft Windows. [14]



Obrázok 1.6: Rozhranie aplikácie WhiteMonkey [15]

1.2 Zhrnutie poznatkov

Po zhodnotení podobne zameraných existujúcich aplikácií je možné zhrnúť niekoľko záverov. Existuje obrovské množstvo aplikácií určených na editáciu textu, z ktorých má užívateľ možnosť si vybrať. Niektoré z nich sú aj svojou funkcionalitou zamerané výhradne na tvorivé písanie. Nie je však možné povedať, že by existovalo riešenie, ktoré by plne spĺňovalo požiadavky vyvíjanej aplikácie.

Negatívom kvalitnejších aplikácií je cena, ktorá často krát nie je nízka a užívateľ si nemusí byť istý či mu daná aplikácia bude na prácu vyhovovať. Pre prípady jednoduchších aplikácií, ktoré sú k dispozícii zdarma, je typická veľmi obmedzená funkcionalita, nemoderný vzhľad či nedostatočná prispôsobivosť.

Ďalšou veľkou nevýhodou je obmedzenie na platformu, nakoľko množstvo aplikácií z uvedeného zoznamu je dostupných iba na jednu platformu. Výnimkami, ktoré sú dostupné na všetky platformy sú Scrivener, ktorý je zameraný skôr na žurnalistiku a FocusWriter, disponujúci veľmi obmedzenou prispôsobivosťou, nemoderným GUI či absenciou správy postáv.

Veľkým problémom sa ukázala častá neprítomnosť českej či slovenskej lokalizácie, čo môže odradiť mnohých spisovateľov. Dokonca aj u veľmi komplexných a rozšírených aplikácií akou je Ulysses tento nedostatok pretrváva.

Voľne dostupnou aplikáciu, spĺňajúcu všetky výsledné požiadavky teda aktuálne trh nedisponuje. Navyše funkcie, ako správa postáv či generátor mien, nie sú aktuálne zahrnuté v žiadnej známej aplikácii, čo zdôrazňuje potrebu vyvíjanej aplikácie.

2 Návrh aplikácie

Druhá časť práce je primárne venovaná procesu návrhu aplikácie. Ako prvé rozoberie cieľovú skupinu a následne aj na základe týchto poznatkov špecifikuje nároky ako na GUI aplikácie, tak aj na kľúčové funkcie, ktoré je nutné implementovať. Následne je pomocou schém uvedené rozhranie a jeho jednotlivé časti, ako aj ďalšie okná, ktorými aplikácia disponuje. Záver kapitoly sa zaoberá dostupnými API, a následne ich analýzou a výberom konkrétnej technológie pre vývoj. [16]

2.1 Štúdium užívateľov

Cieľovú skupinu aplikácie tvoria spisovatelia, od začiatočníkov až po profesionálov a to všetkých vekových kategórií. Je teda možné predpokladať, že užívateľ disponuje priemernými znalosťami, pokiaľ ide o prácu s počítačom, no zároveň však má množstvo skúsenosti s textovými procesormi. Nakoľko trh disponuje značným počtom podobných aplikácií, je možné usúdiť, že užívateľ je aspoň s niektorými z nich zoznámený a teda nemá veľký záujem zoznamovať sa s novou a komplikovanou aplikáciou. Vzhľadom na tieto fakty je nutné užívateľské rozhranie navrhnuť tak, aby ich na prvý pohľad zaujalo, no zároveň neporušilo konzistenciu programov, na ktoré sú užívatelia zvyknutí. Grafické rozhranie musí byť prehľadné s dôrazom na intuíciu, aby užívateľ po niekoľkých minútach práce získal prehľad o všetkých dostupných funkciách.

2.2 Špecifikácia

Český autor René Nekuda, ktorý sa tiež venuje výučbe tvorivého písania vo svojej recenzii aplikácií Ulysses zdôrazňoval prvky kvalitného textového editora. Medzi kľúčové aspekty patrilo zaradenie medzi tzv. antirušivé textové editory, ktoré sa ako názov napovedá, zameriavajú na vytvorenie nerušivého prostredia pre zlepšenie koncentrácie spisovateľa. Ďalším dôležitým bodom je, aby všetky potrebné funkcie boli k dispozícii v jedinej aplikácii. K výhodám začleňuje aj použitie systému Markdown pre zvýrazňovanie nadpisov a kapitol. Export do rôznych formátov, vrátane formátu EPUB, je za samozrejmosťou. Na záver vyzdvihuje minimalistický design bez zbytočných kloniek, tlačidiel a ikon, no tiež možnosti prispôsobiť si štýl rozhrania. [17]

Hlavným cieľom implementovanej aplikácie je schopnosť pracovať ako stabilný textový procesor a teda disponovať základnou sadou funkcií, ktoré užívateľ očakáva. Medzi tie patrí ukladanie a otváranie projektov, ako aj ich export. Ďalej sem patrí prispôbenie rozmerov samotnej textovej plochy a špecifikovanie príslušného písma. Nevyhnutnými sú tiež možnosti vyhľadávania v texte a možnosť vrátiť sa späť o niekoľko krokov pri písaní v situáciách, ak by sa užívateľ pomýlil. Kľúčovou funkciou pre prácu každého spisovateľa je tiež automatická správa kapitol a pridávanie poznámok, či už k dokumentu alebo k jednotlivým kapitolám. Ďalšími overenými funkciami, ktoré zefektívňujú tvorbu autorov sú monitorovanie štatistík a tiež možnosť nastavovania cieľov a to, ako časových, tak aj zadávanie počtu slov, ktoré autor na zasadnutie napíše. Medzi pokročilejšie funkcie, ktoré robia vytváranú aplikáciu výnimočnou patrí správa postav, ktoré dávajú užívateľovi možnosť vytvárať vlastný zoznam postáv pre každé ich dielo. Ďalšou funkciou bude generovanie mien, na základne špecifikovaných požiadaviek, čo autorovi uľahčí vymýšľanie nových postáv alebo mu pomôže nadobudnúť inšpiráciu.

2.2.1 Kľúčové funkcie

Okrem uvedených všeobecných funkcií textového editora bude vyvíjaná aplikácia obsahovať aj niekoľko špecifických. Aplikácia pre tvorivé písanie je aplikáciou s konkrétnymi požiadavkami, ktoré vytváraná aplikácia naplňuje nasledujúcou funkcionalitou.

Kapitoly sú v texte vyhľadávané automaticky, buď na základe Markdown systému alebo pomocou aspoň dvoch prázdnych riadkov predchádzajúcim novej kapitoly. V texte nájdená kapitola následne vytvorí odkaz, ktorý pri kliknutí tlačidla myši zobrazí detail danej kapitoly. Každá kapitola má možnosť pridania komentára a tiež funkciu presunutia kurzoru na začiatok vybranej kapitoly.

Dostupný zoznam postáv umožňuje pridávanie, editovanie či odoberanie postáv, čo umožní spisovateľovi vytvárať vlastný zoznam postáv pre každé ich dielo. Každá postava v zozname je automaticky vyhľadaná v texte, čím rovnako ako v prípade kapitol vytvorí odkaz zobrazujúci detail postavy. Každá postava má povinný údaj meno, ktorý sa však na rozdiel od kapitol nemôže opakovať, ďalej obsahuje nepovinné údaje ako pohlavie, vek a tiež všeobecný popis postavy. Pri vypisovaní poľa s menom má užívateľ možnosť využiť generátor mien, ktorý je bližšie popísaný v kapitole 3.2.7.

Užitočnou funkciou každej aplikácie pre tvorivé písanie je možnosť zadávania špecifických cieľov. To nebude chýbať ani vo vyvíjanej aplikácii, čo spisovateľovi umožní zefektívniť svoju prácu. Aplikácia umožňuje zadávať dva druhy cieľov a to buď časové, kedy spisovateľ špecifikuje čas, počas ktorého sa bude venovať písaniu, alebo si určí počet slov, ktoré plánuje napísať. Kedykoľvek bude mať k dispozícii informácie, ako s daným cieľom pokročil, pričom vždy môže byť aktívny cieľ iba jeden. Pre prehľadnosť je v dolnej časti aplikácie umiestnená lišta, ktorá zobrazuje aktuálny pokrok v cieľoch ak je cieľ aktívny.

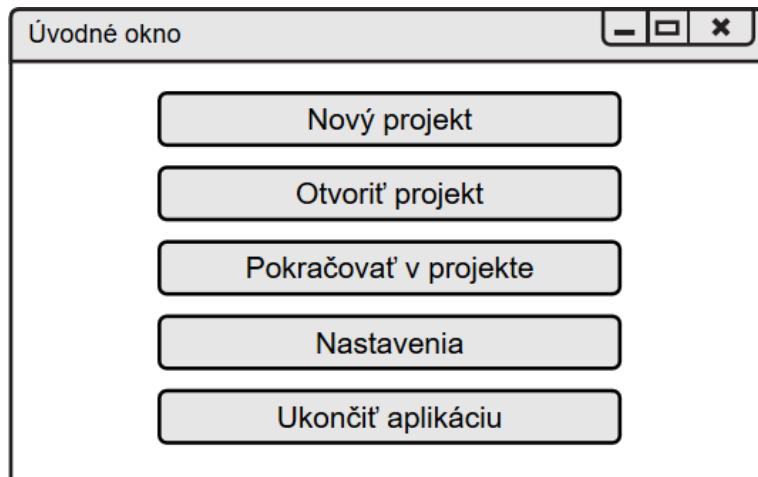
2.3 Návrh užívateľského rozhrania

Táto sekcia práce sa venuje návrhu GUI aplikácie vzhľadom na špecifikované požiadavky. Pre lepšie objasnenie sú v jednotlivých podkapitolách uvedené schémy návrhu, tzv. mockupy [18]. Tie boli vytvorené v nástroji Moqups [19]. Pre prehľadnosť je GUI rozdelené do niekoľkých okien, ktoré sú v nasledujúcich podkapitolách postupne opísané.

K požiadavkám spisovateľov, kvôli ktorej sa zvyknú vyhýbať populárnym textovým procesorom, akým je Microsoft Word, patrí aj nerušivosť prostredia. Strata koncentrácie výrazne znižuje efektivitu písania, čo je dôvod prečo každá zo zmienených aplikácií pre tvorivé písanie disponujú možnosťou prepnutia na režim celej obrazovky. Vzhľadom na poznatky zistené o cieľovej skupine užívateľov je prevažné množstvo grafických elementov navrhnutých v podobe zaužívaných ikon a tým je minimalizovaná potreba textových vysvetlení. Pri žiadaní užívateľa o komplexnejší vstup, napríklad práca s časom či výber farieb, budú k dispozícii moduly zjednodušujúce túto činnosť a vylučujú možnosť chyby.

2.3.1 Úvodné okno

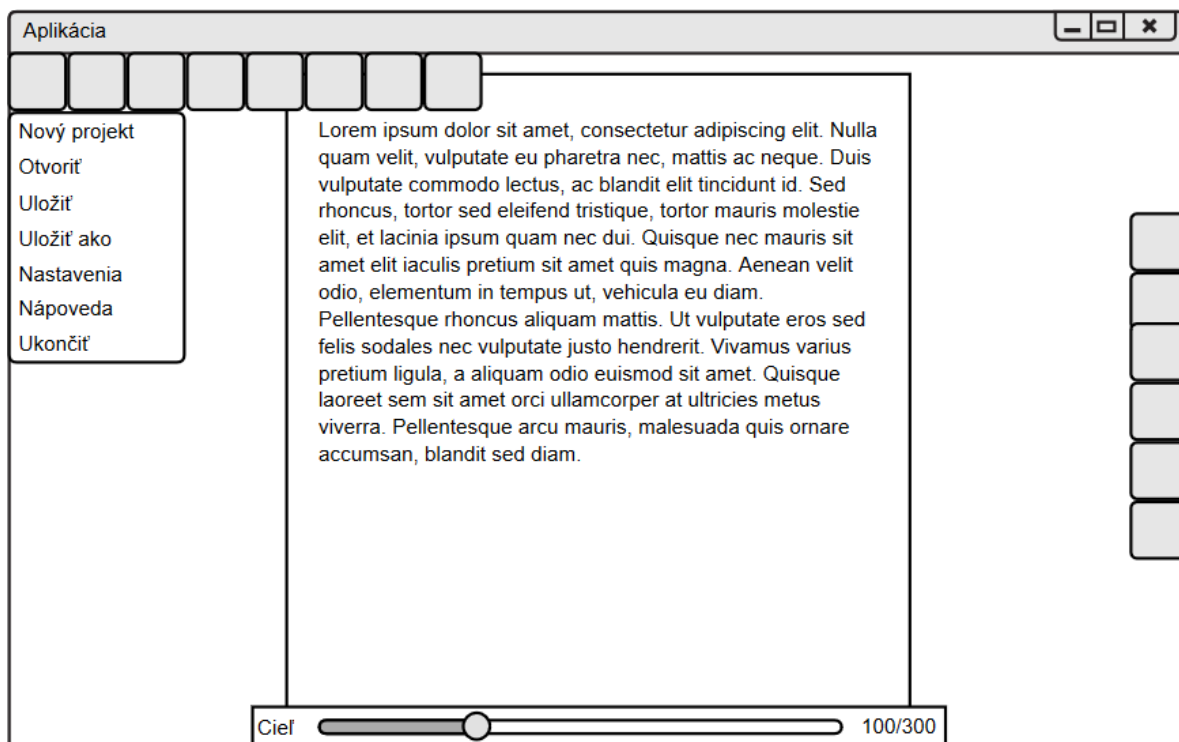
Úplne prvým oknom, ktoré sa užívateľovi naskytne po zapnutí aplikácie je úvodné okno, zobrazené na obrázku 2.1. To bude modulárne a teda neumožní užívateľovi prácu s aplikáciou, kým si nevyberie niektorú z možností. Okno obsahuje sériu tlačidiel pre začatie nového projektu, otvorenie existujúceho projektu, kde si užívateľ špecifikuje konkrétny súbor, ďalej je tu možnosť pokračovania v existujúcom projekte, ktorý spustí posledný projekt na ktorom užívateľ pracoval. Nechýbajú nastavenia pre zmenu jazyka, prípadne iných vlastností. Posledné tlačidlo celú aplikáciu ukončí.



Obrázok 2.1: Mockup úvodného okna

2.3.2 Rozhranie

Hlavná funkcionálna by mala byť rozdelená do troch častí. Prvou z nich je plocha, na ktorú bude užívateľ písať, umiestnená na pozadí v centre aplikácie. Tá bude vždy viditeľná a teda užívateľ má kedykoľvek možnosť nahliadnuť na napísaný text. Panel obsahujúci menu a niekoľko ďalších základných funkcií, ktoré nemanipulujú s obsahom, bude k dispozícii v pre ňu typickej, vrchnej časti aplikácie, ukotvenej v ľavom hornom rohu. Je dôležité, aby si aplikácia zachovala jednoduchosť a prehľadnosť, preto všetky pokročilé funkcie, ako aj funkcie pre manipulovanie s obsahom sú k dispozícii v druhom paneli. Ten bude rozdelený do sekcií prístupných jednotlivým funkciám, pričom užívateľ bude pracovať vždy iba s jednou sekciou. Tento panel je umiestnený v pravej časti aplikácie, vycentrovaný na stred. Na obrázku 2.3 je znázornená ukážka opísaného rozloženia. Nakoľko je pre aplikácie tohto typu esenciálna nerušivosť užívateľa, obidva spomínané panely sa budú automaticky skrývať a vysúvať podľa pohybu myši, umožňujúc tak užívateľovi sústrediť sa na tvorbu. Dôležitým aspektom sú možnosti prispôbenia, ktoré sprístupnia užívateľovi výber farieb, nastavenie automatického skrývania, či úplné vypnutie niektorých funkcií a budú k dispozícii v nastaveniach či v príslušajúcich paneloch.



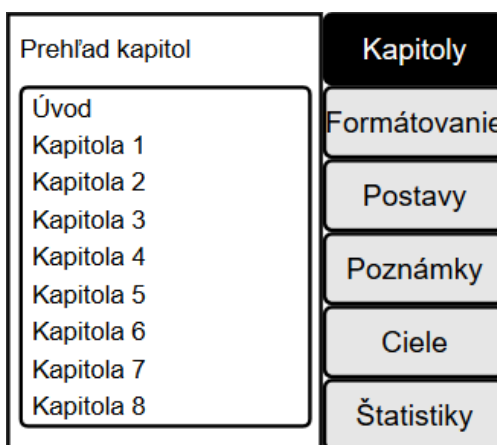
Obrázok 2.2: Mockup rozhrania

2.3.2.1 Horná lišta

Táto časť slúži pre manipuláciu s aplikáciou, pričom jej funkcie nijak neupravujú písaný text. Horná lišta je umiestnená v ľavej hornej časti obrázku 2.2. Obsahuje niekoľko tlačidiel zobrazujúcich menu aplikácie s funkciami pre otvorenie či uloženie projektu, prístup k nastaveniam, nápovede či možnosť ukončenia aplikácie. Panel tiež obsahuje tlačidlá pre vracanie krokov, prispôbenie rozmerov plochy s textom, vyhľadávania v texte a prepínanie s režimom na celú obrazovku.

2.3.2.2 Pravá lišta

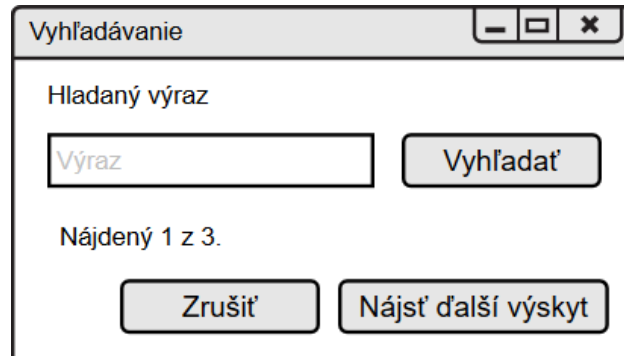
Táto lišta je umiestnená v pravej časti obrazovky a sprístupňuje funkcie pre prácu s textom. Za týmto účelom je rozdelená do niekoľkých sekcií, pričom rozhranie každej z nich je zobrazené na vysúvateľnom paneli. Užívateľ má vždy k dispozícii jednu sekciu a pri dvojkliku na zvýraznenú sekciu sa panel automaticky schová. Toto usporiadanie je zobrazené na obrázku 2.3. Zoznam sekcií tvorí prehľad kapitol, formátovanie, správa postáv, poznámky, ciele a štatistiky.



Obrázok 2.3: Mockup pravého panelu

2.3.2.3 Vyhľadávanie

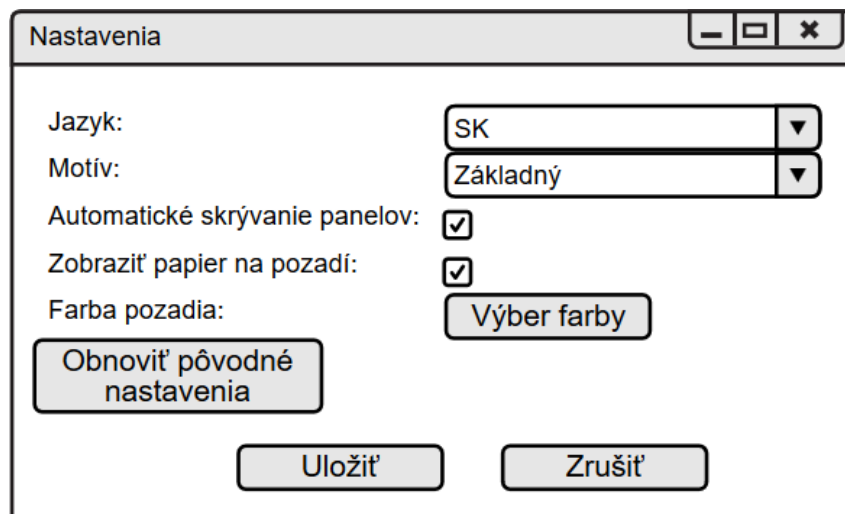
Okno pre vyhľadávanie slúži na nájdenie a označenie časti textu a je zobrazené na obrázku 2.4. Užívateľ zadá hľadaný výraz a postupne prechádza jednotlivé výskyty, ktoré presunú kurzor a označia nájdený text v hlavnom okne aplikácie.



Obrázok 2.4: Mockup okna vyhľadávania

2.3.2.4 Nastavenia

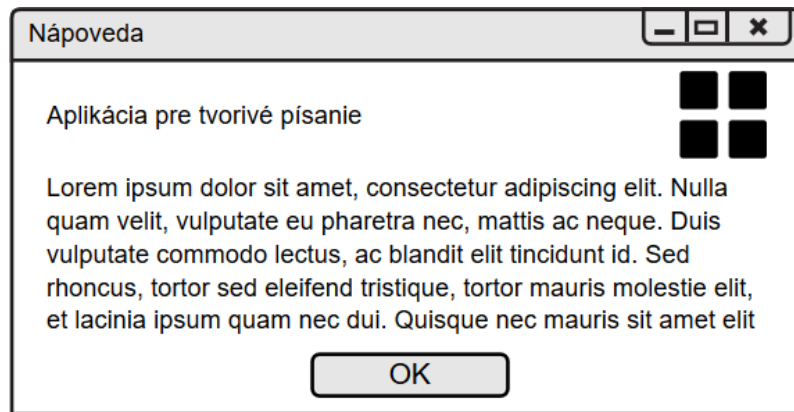
Pomocou tohto okna bude užívateľ môcť konfigurovať základné nastavenia aplikácie, ako aj niektoré funkcie prispôsobenia. Patrí k nim výber jazyka, motívu, prepínanie skrývania panelov či skrývanie pozadia za textovou plochou. Pre výber farby pozadia je k dispozícii modul pre výber farby klikom. Okno nastavení tiež disponuje tlačidlom pre obnovenie všetkých nastavení do pôvodného stavu. Rozloženie okna je zobrazené na obrázku 2.5.



Obrázok 2.5: Mockup okna nastavení

2.3.2.5 Nápoveda

Okno nápovedy zobrazuje stručný prehľad funkcií aplikácie. Taktiež sa tu nachádzajú informácie o projekte, autorovi a fakulte. Rozloženie okna je zobrazené na obrázku 2.6.



Obrázok 2.6: Mockup okna nastavení

2.4 Architektúra aplikácie

Táto časť práce je venovaná predstaveniu a porovnaniu najpoužívanejších existujúcich rozhraní pre programovanie aplikácií vhodných pre vývoj navrhutej aplikácie, tzv. API [16]. Ide o predom určený súbor funkcií, tried, knižníc, či platforiem ktoré môže programátor využívať. Pre tento projekt je možné použiť akúkoľvek technológiu, ktorá sa orientuje na desktopové GUI.

2.4.1 Dostupné technológie

Jednou z najpopulárnejších multiplatformových knižníc pre vytváranie programov s GUI je Qt. Využíva sa od roku 1999 a aktuálnu verziu 5.0 spravuje firma Nokia. Qt je knižnica primárne programovacieho jazyka C++. Výhodou Qt je prehľadne spracovaná dokumentácia a silná podpora technológie. Aplikácie nevyžadujú pre svoj beh inštaláciu žiadnych externých nástrojov či knižníc a sú k dispozícii na všetkých platformách. Používajú natívny vzhľad operačného systému takže vyvinutá aplikácia sa vždy prispôsobí užívateľskému prostrediu. [20]

Ďalšou je knižnica Swing pre platformu Java od firmy Oracle a podporuje tvorbu klasických užívateľských rozhraní. Java Swing patrí od vydania Javy 8, teda marca 2014 medzi zastaralé a pomaly sa nahrádza technológiou JavaFX. Medzi jej hlavné výhody patrí fakt, že ide o modernú technológiu a tiež zahŕňa dostupnosť na všetky platformy, keďže je od verzie 8 natívnou súčasťou Javy. Umožňuje tvorbu bohatých klientskych prostredí pracujúcich stabilne. K nevýhodám patrí nutnosť inštalovania Javy na cieľovom zariadení. [21]

Mocným nástrojom pre tvorbu užívateľských rozhraní je Windows Presentation Foundation, teda WPF od firmy Microsoft. Ide o podmnožinu .NET Frameworku od verzie 3.0 a nahrádza technológiu WinAPI. WPF používa značkovací jazyk XAML a programovací jazyk C#. Výhodou je podpora moderných návrhových modelov, pokročilej grafiky a animácií, či práce s médiami. Nevýhodou je podpora zameraná na platformu Microsoft Windows, aj keď firma v poslednom období propaguje svoju technológiu .NET Core, ktorá umožňuje rozšírenie WPF na ostatné platformy, stále ide o novú a pomaly sa rozširujúcu technológiu. [22]

Tkinter je štandardná knižnica pre tvorbu GUI programovací jazyk Python. Výhodou je dostupnosť na všetky platformy. K nevýhodám patria obmedzenejšie možnosti, ako ostatné už spomínané knižnice a vyžadovanie inštalácie na konečnom zariadení. [23]

2.4.2 Vybrané API

Všetky uvedené technológie v sebe zahrňujú požadovanú funkcionálnosť či dostupnosť a výsledná voľba je skôr otázkou rozšírenia, stability či preferencie. Okrem knižnice Qt sa u všetkých vyžaduje inštalácia. Za túto cenu však ponúkajú modernejšie a bohatšie možnosti vývoja. Aj pre tieto dôvody bol zvolený vývoj tejto aplikácie na platforme Java. S týmto rozhodnutím však prichádza aj nevýhoda, ktorá spočíva v požiadavke na cieľovú platformu, aby mala nainštalovanú Javu SE verziu 8. Programovací jazyk Java sa však za 20 rokov používania značne rozšíril medzi vývojárov a užívateľov a udáva 89% pokrytie desktopov v USA, pričom je možné predpokladať, že toto číslo ostáva vysoké či už v Českej a Slovenskej Republike. [24]

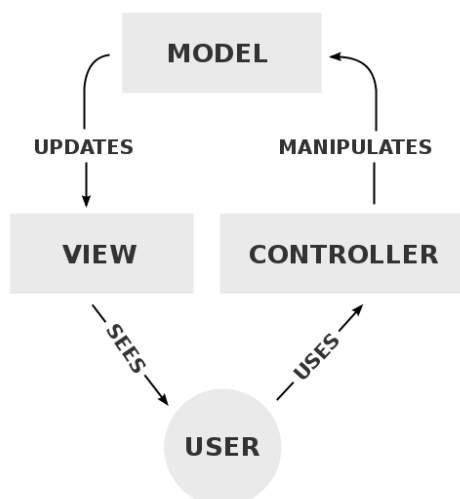
Samotný vývoj aplikácie prebiehal v IDE (vývojové prostredie z angl. integrated development environment) IntelliJ IDEA 2016 Community Edition od spoločnosti JetBrains. Je k dispozícii pod licenciou Apache (ASL) ktorá umožňuje softwarový vývoj zadarmo. Ďalej bol použitý nástroj JavaFX Scene Builder 2.0, ktorý slúži pre návrh GUI. [25]

3 Implementácia

V nasledujúcej kapitole sú popísané podstatné časti implementácie vyvíjanej aplikácie. Úvodná časť opisuje možnosti použitej technológie. Následne sú zobrazené aplikované diagramy tried a ich prepojenie s GUI. Diagramy sú v štandarde UML (skratka z angl. Unified Modeling Language) a boli vytvorené v nástroji Violet UML Editor [26].

3.1 JavaFX framework

Pre vývoj desktopových aplikácií v programovacom jazyku Java 8 je k dispozícii platforma JavaFX, využívajúca stromovú štruktúru, ktorá definuje užívateľské rozhranie zvané graf scény. Obsahuje viac ako šesťdesiat definovaných komponentov pre ľahší vývoj aplikácií. Štýlovanie grafických prvkov je možné pomocou kaskádových štýlov (v anglickom originálne Cascading Style Sheets so skratkou CSS). Pre popis rozloženia komponentov v GUI sa používa modifikovaný deklaratívny jazyk XML (skratka z anglického eXtensible Markup Language) teda rozšíriteľný značkovací jazyk uložený v súboroch FXML. Za interakciu s užívateľom zodpovedá trieda Controller. Aplikuje tak architektonický vzor MVC (z anglického Model View Controller). [21][27]

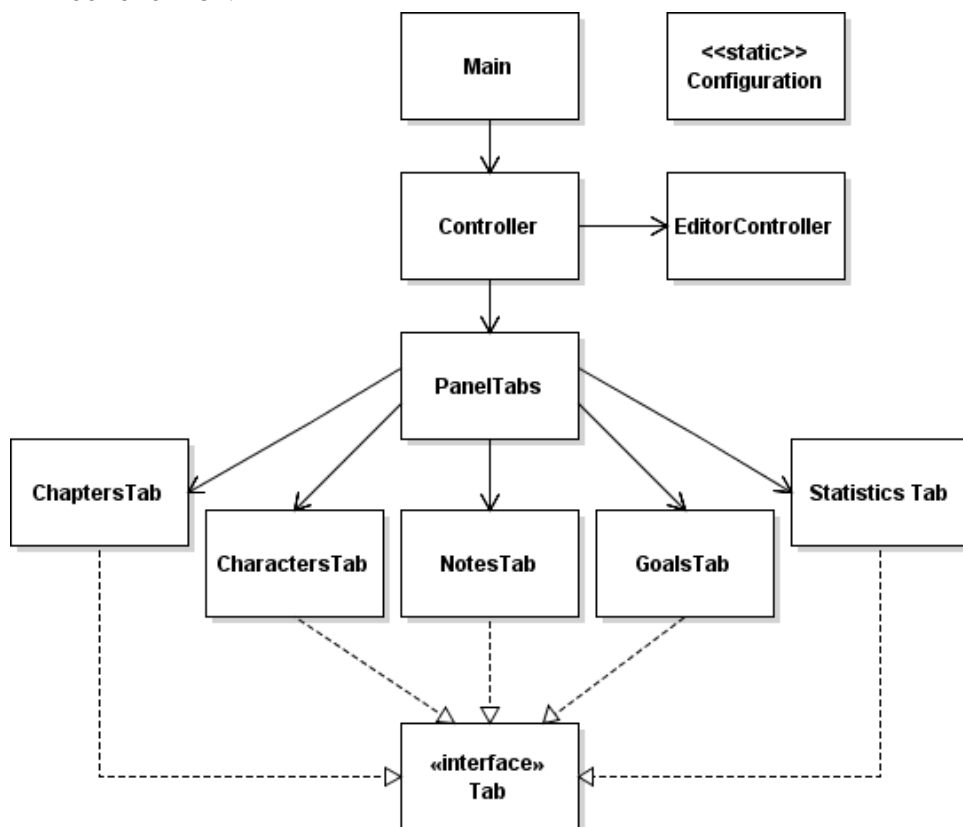


Obrázok 3.1: Schéma modelu MVC [28]

3.2 Implementovaný model

Po štarte aplikácie sa ako prvá spúšťa trieda `Main`, ktorá rozširuje triedu `Application`, čím implementuje potrebné metódy pre beh JavaFX aplikácie. Vytvára kontajner najvyššej úrovne zvaný `stage`, ktorý reprezentuje a riadi okno aplikácie. Pomocou metódy `loadView()` načíta rozloženie definované súborom FXML a následne metódou `setStage()` vytvorí kontajner spravujúci obsah v grafe scény. Tieto metódy sú definované samostatne, nakoľko pri modifikovaní niektorých nastavení dochádza k ich opätovnému volaniu, keďže je nutné tieto komponenty načítať nanovo. Trieda tiež zabezpečuje načítanie kaskádových štýlov, pridaných typov písma dostupných v aplikácii a tiež nastavuje názov a ikonu. Iniciuje triedu `Configuration`, ktorá spravuje nastavenia užívateľa.

Detailne je popísaná v podkapitole 4.2.1. Popísaným spôsobom trieda `Main` končí inicializáciu a správu predáva triede `Controller`.



Obrázok 3.2: UML návrhu tried

3.2.1 Trieda Configuration

Táto trieda spravuje a ukladá zmeny nastavení aplikácie užívateľom a za týmto účelom musí byť dostupná počas celého behu aplikácie z viacerých modulov. Z týchto dôvodov bola deklarovaná s modifikátorom `static`, čím je vždy k dispozícii iba jedna inštancia triedy a vytváranie nových nie je možné. Pre ukladanie je v adresári s aplikáciou vytvorený priečinok s názvom `Data` a v ňom súbor `RufusConfiguration.xml` kde sú tieto nastavenia uložené vo formáte XML dokumentu. Na ukladanie slúži metóda `saveConfiguration()` využívajúca modul `DocumentBuilder`, ktorý slúži na tvorbu formátu značkovacieho jazyka XML a je natívnou súčasťou jazyka Java. Túto konfiguráciu je možné obnoviť na pôvodné hodnoty, ktoré sú tiež aplikované pri prvom behu aplikácie. V prípade, že tento súbor nie je k dispozícii, je pri štarte aplikácie vytvorený, inak je načítaný pomocou funkcie `loadConfiguration()`. Ukážka súboru so všetkými ukladanými parametrami je k dispozícii v prílohe 2.

3.2.2 Trieda Controller

`Controller` v aplikáciách JavaFX zabezpečuje interakciu s užívateľom a to tak, že prepojuje grafické prvky a ich funkcie, definované v súbore FXML s volaním konkrétnych funkcií. Ako prvá sa volá metóda `initialize()`, ktorá nastaví fungovanie grafických komponentov podľa načítanej konfigurácie. K tomuto cieľu využíva štruktúru nazývanú `Listener`, ktorá umožňuje zachytávať a spracovávať rôzne udalosti. Nakoľko je v tejto časti nutné iniciovať iba prvky, ktoré sú k dispozícii počas celého behu aplikácie, aplikuje `setOnMouseEntered` pre vstup myši do oblasti

a `setOnMouseExited` pre výstup myši z oblasti do panelov s funkciou automatického skrývania. Správne fungovanie hlavného menu zabezpečuje `showingProperty`, ktorá upozorňuje na jeho zobrazenie a relevantné grafické prvky, tak prispôsobia svoju činnosť. V poslednom rade pridáva `setOnMouseClicked`, čím spracováva dvojkliky na zoznamy a umožňuje tak intuitívnejšiu prácu užívateľa.

Pri programovaní týchto štruktúr boli použité výrazy Lambda, ktoré pribudli v Jave od verzie 8. Lambda výraz je anonymná funkcia, umožňujúca písanie miestnych funkcií, ktoré je možné predať ako argument či návratovú hodnotu. Tie boli využívané pri vytváraní štruktúr zachytávajúce konkrétne udalosti v GUI. K ich výhodám tiež patrí stručnejší a prehľadnejší zápis, no často umožňujú tiež dosiahnuť kratší výpočtový čas. V nižšie uvedenej ukážke je zachytená zmena textu v editore. Zmenu zachytí `Listener`, ktorý by inak vyžadoval vytvorenie inštancie triedy `ChangeListener` a prepis jej metódy `changed()` pre zachytenie zmeny. Nutný by tak tiež bol `import` ďalšieho modulu do triedy. [29]

```
editor.textProperty().addListener((obs, oldText, newText) -> {
    computeHighlighting(newText.toString());
});
```

Trieda `Controller` tiež obsahuje referencie na triedy, ktoré spravujú jednotlivé časti aplikácie a ktorým predáva volanie funkcií. Každá z nich tak pracuje iba s konkrétnou skupinou grafických komponentov, ktoré sú pre ňu relevantné a tie predáva trieda `Controller` pomocou štruktúry nazývanej `Map`, umožňujúc mapovanie názvov týchto komponentov na referencie.

3.2.3 Trieda `EditorController`

Pre samotný textový editor má Java k dispozícii niekoľko komponentov. Prvým je `TextArea`, ktorý však podporuje iba základné funkcie pre prácu s textom, čo pre vyvíjanú aplikáciu nepostačuje. Druhým je `HTMLEditor`. Tento komplexný nástroj má množstvo funkcií avšak je možné ho prispôbiť iba čiastočne, čo sa pri pokuse o prepojenie s funkciami aplikácie ukázalo ako nedostatočné. Ďalšou nevýhodou je, že užívateľovi umožňuje pracovať s tagmi. Nakoniec bola pre prácu s textovou plochou použitá knižnica z tretej strany *RichTextFX* a jej komponenta `StyledTextArea`, nakoľko umožňuje vytvárať odkazy priamo v texte.

Trieda `EditorController` zapuzdruje všetky funkcie spojené s textom. Stará sa o zväčšovanie a zmenšovanie textovej plochy pomocou funkcií `increaseResolution()` a `decreaseResolution()`. V okamihu keď v texte nastane zmena obsahu, zachytí to `ChangeListener` a volá metódu `computeHighlighting()`. Tá najprv vynuluje akékoľvek štýly už aplikované nad textom a následne vytvorí regulárny výraz metódou `createRegex()`, ktorý pozostáva z formátu zápisu kapitol a tiež mien postáv. Zhody regulárneho výrazu sa následne cyklicky prechádzajú a podľa typu nálezu sa na danú sekciu textu vytvorí inštancia triedy `Hyperlink`, ktorá uchováva index kapitoly či meno postavy. Táto trieda je prepojená s `StyledTextArea` a pri vzniku automaticky aplikuje funkciu `applyToText()`, čím štylizuje sekciu obsahujúcu názov a kapitolu. Zároveň nastaví `onMouseClicked()` parameter, ktorý v panely zobrazí podrobné informácie či už o kapitole alebo postave.

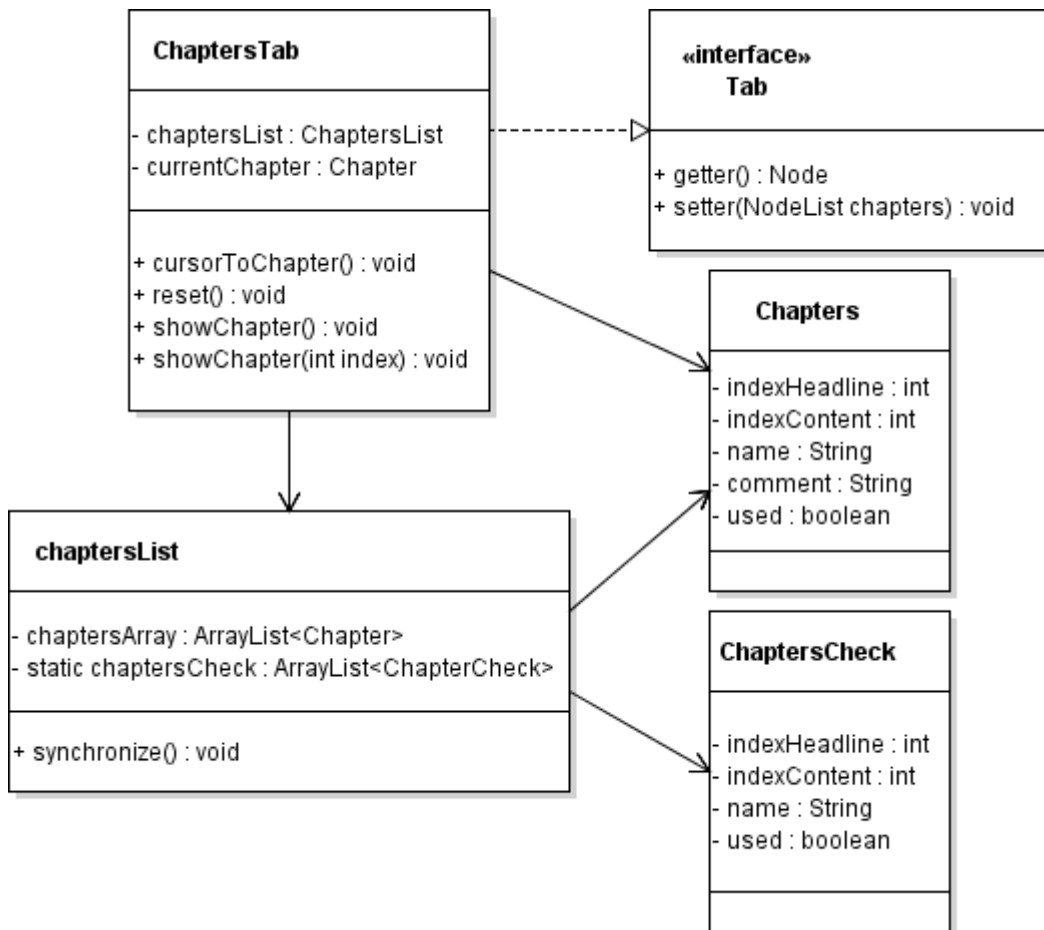
3.2.4 Správa pravého panelu

Fungovanie pravého panelu zabezpečuje trieda `PanelTabs`. Pravý panel obsahuje 5 kariet, pričom funkcie každého z nich spravuje jedna trieda, popísané v nasledujúcich podkapitolách. Okrem predávania volaní funkcií sa táto trieda stará o automatické skrývanie a správne fungovanie panela tak, aby vždy bola užívateľovi k dispozícii iba jedna karta. Za týmto účelom obsahuje definovaný výpočtový typ `Tab`, ktorý určuje vybranú kartu. Z metód spravujú interakciu `panelShow()` a `panelHide()` pre skrývanie a zobrazovanie panelu, ďalej `changeTab()` pre zmenu karty a `tabHide()` pre skrývanie kariet. Obsahuje tiež metódu `reset()`, ktorá resetuje obsah grafických komponentov, čo sa využíva pri otváraní nových projektov. Nakoľko všetky údaje patriace k projektu písaného užívateľom sú uložené buď v textovej ploche alebo v pravom paneli, obsahuje tiež trieda `PanelTabs` metódy `openProject()` pre načítanie obsahu, ktorý sa potom predá jednotlivým triedam spravujúcich karty a `saveProject()` pre výber obsahu z kariet a uloženie do súboru.

3.2.4.1 Kapitoly

Trieda zapudzujúca funkcionálnosť karty s kapitolami sa volá `ChaptersTab`. Aktuálny zoznam kapitol nájdených v texte je k dispozícii v zozname `ListView` pričom užívateľ sa môže presunúť na pozíciu kapitoly volaním funkcie `cursorToChapter()`, alebo zobrazíť detail kapitoly volaním `showChapter()`. Pre ukladanie a načítanie projektu sú k dispozícii metódy `getter()` a `setter()`, ktoré pracujú s XML štruktúrou dokumentu a trieda ako všetky implementuje interface `Tab`.

Samotný zoznam kapitol spravuje trieda `ChaptersList` a to tak, že má k dispozícii dva zoznamy. Prvý je zoznam inštancií triedy `Chapter`, ktoré reprezentujú reálne kapitoly, obsahujú názov, potrebné indexy a poznámky ku kapitole. Druhý je zoznam inštancií triedy `ChapterCheck` dočasný zoznam, ktorý sa vytvára pri zmene kapitol v texte. Oproti triede `Chapter` nezahŕňujú komentáre. Volaním funkcie `synchronize()` sa kontrolujú a dorovnávajú rozdiely. V prípade rovnakého počtu kapitol algoritmus aktualizuje indexy a upraví názvy kapitol. V prípade rozdielného počtu mohol užívateľ pridať, upraviť či zmazať aj niekoľko kapitol v rôznych častiach textu. Algoritmus preto najprv iteruje nad prvým zoznamom kapitol, hľadajúc ich ekvivalent a v prípade neúspechu kapitolu maže. Následne iteruje nad druhým zoznamom a pridáva chýbajúce kapitoly na správne indexy. To umožňuje uchovávanie komentárov k jednotlivým kapitolám aj pri dynamicky meniacom sa obsahu. Trieda `Chapter` obsahuje tiež atribút `used` typu `boolean`, ktorý sa resetuje na začiatku každého porovnávania kapitol a umožňuje tak aplikáciám pracovať s kapitolami s duplicitným názvom.



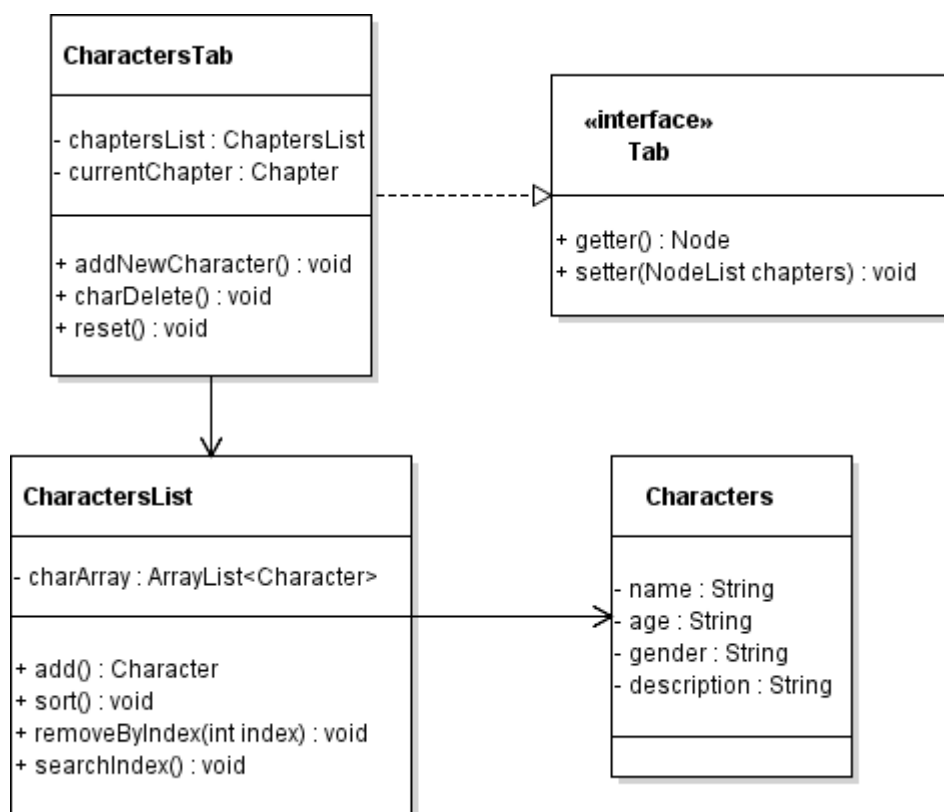
Obrázok 3.3: UML tried správy kapitol

3.2.4.2 Postavy

Postavy spravuje trieda `CharactersTab` a ich zoznam bude k dispozícii pomocou elementu `ListView`. Užívateľ môže pridávať nové postavy volaním metódy `addNewCharacter()`, ktorá zobrazí formulár pre pridávanie nových postáv obsahujúc meno, vek, pohlavie a popis. Na mazanie postavy slúži metóda `charDelete()`, ktorá zmaže označenú postavu zo zoznamu. Aj táto trieda obsahuje metódy `getter()` a `setter()` pre načítanie či ukladanie dát a metódu `reset()` pre vynulovanie zoznamu.

Zoznam postáv spravuje trieda `CharactersList()` obsahujúc zoznam inštancií triedy `Character`. Implementuje všetky potrebné funkcie zoznamu, ktoré vyžaduje trieda `CharactersTab` - pridávaním prvkov metódou `add()`, zoradením metódou `sort()`, mazaním podľa indexu prvkov metódou `removeByIndex()` či vyhľadávaním indexu postavy metódou `searchIndex()`.

Trieda `Character` obsahuje potrebné údaje k postave teda meno, vek, pohlavie a popis. Pri vytváraní inštancie vyžaduje konštruktor jedine meno, zadanie zvyšných atribútov teda nie je povinné. Z metód obsahuje gettery a settery pre atribúty a prekrýva metódu `equals()`, aby bolo možné radenie objektov.



Obrázok 3.4: UML tried správy postáv

3.2.4.3 Poznámky

Poznámky spravuje jednoduchá trieda `NotesTab`. Nakoľko ide o poznámky k celému dokumentu, karta obsahuje iba jedinú komponentu a to `TextArea`, do ktorého užívateľ priamo píše svoje poznámky. Samozrejme je opäť nutné zabezpečiť ukladanie a načítanie dát pomocou metód `getter()`, `setter()` a tiež vyčistenie poznámok metódou `reset()`.

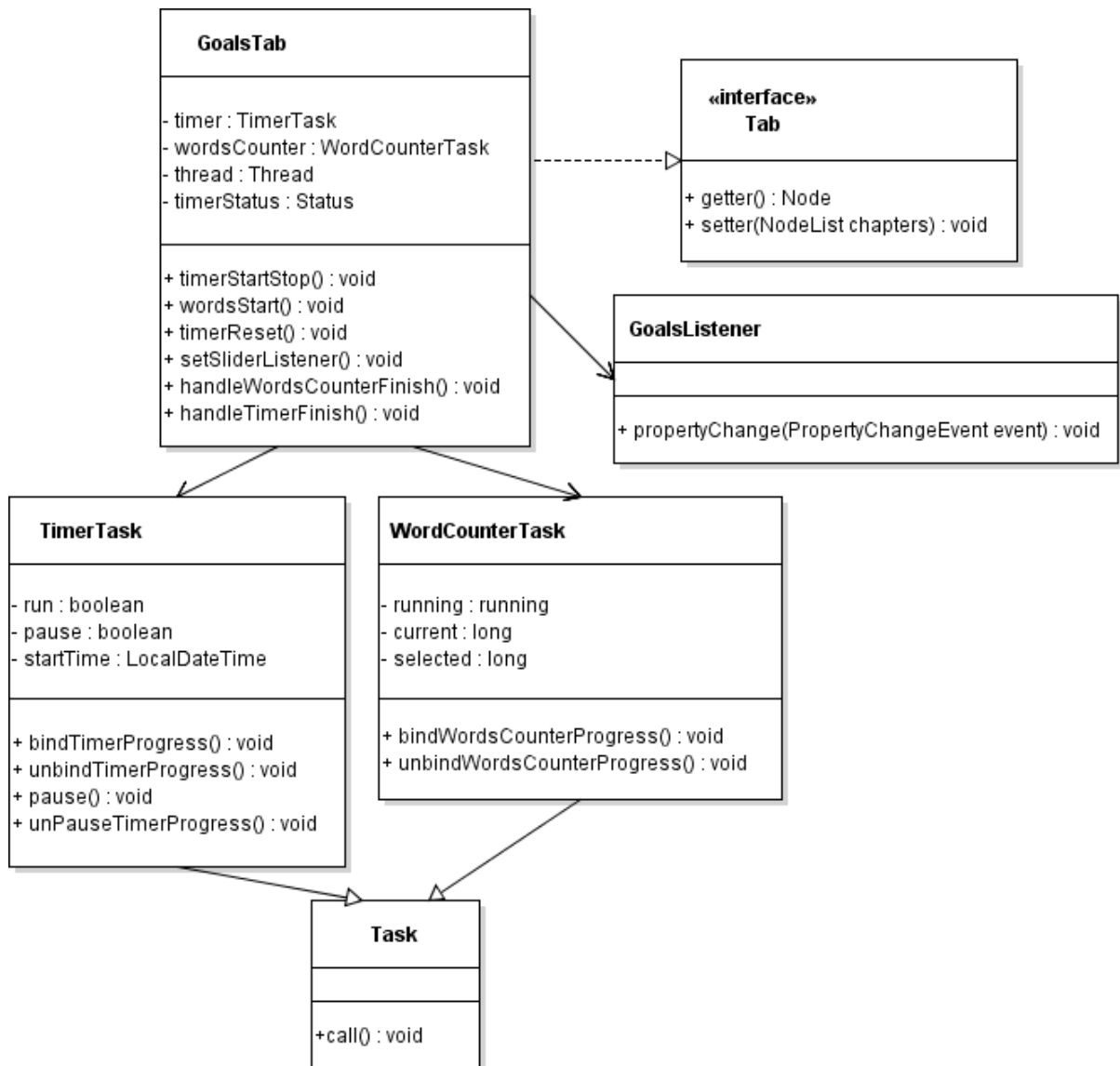
3.2.4.4 Ciele

Trieda spravujúca ciele sa volá `GoalsTab` a zabezpečuje funkcionálnosť tlačidiel metódami `timerStartStop()` a `wordsStart()` pre začatie cieľa a metódami `timerReset()` a `wordsReset()` pre ukončenie cieľa. Aktuálny pokrok je zobrazený vďaka metóde `setSliderListener()`, ktorá vytvorí binding, teda prepojenie medzi aktuálne bežiacim cieľom a relevantnými grafickými komponentami.

Pre zachytenie udalosti a dokončenie cieľa bola vytvorená trieda `GoalsListener`, ktorá za týmto účelom implementuje `PropertyChangeListener`, čo je interface jazyka Java. Pre fungovanie vyžaduje implementáciu metódy `propertyChange()`, ktorá triedu `GoalsTab` upozorní na dokončený cieľ a tá pomocou metód `handleWordsCounterFinish()` a `handleTimerFinish()` túto udalosť spracuje.

Časovač implementuje trieda `TimerTask` pričom je nevyhnuté, aby cieľ pracoval na nezávislom vlákne. JavaFX disponuje triedou `Task`, ktorej rozšírením je možné spracovať aj výpočtovo náročnejšie úlohy a prepojiť ich s GUI. Beh vlákna sa nachádza v prekrytej metóde `call()`, pričom ho tvorí cyklus, ktorý periodicky každú sekundu vypočíta uplynulý čas až kým sa cieľ nesplní, alebo užívateľ nevynútil zastavenie. Pre prácu s časom slúži modul `LocalDateTime`, ktorý je natívnou súčasťou Javy. Časovač je tiež možné pozastaviť a neskôr ho zase spustiť.

Druhým dostupným cieľom je `WordCounterTask`, ktorý taktiež rozširuje triedu `Task`. Uloží si aktuálny počet slov v texte a periodicky každú sekundu potom túto hodnotu prepočíta, až kým nie je dosiahnutý cieľ, alebo si užívateľ nevynútil zastavenie.



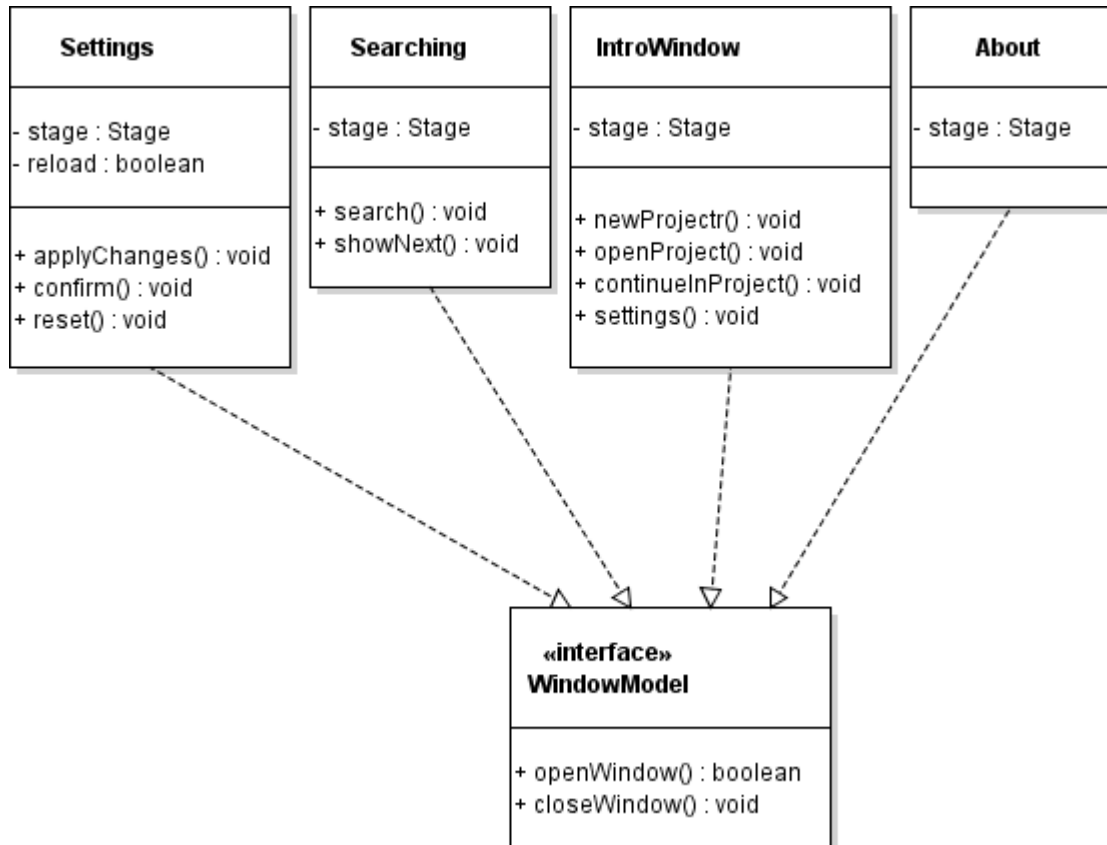
Obrázok 3.4: UML tried správy cieľov

3.2.4.5 Štatistiky

`StatisticsTab` je trieda, ktorá zapúzdruje kartu zobrazujúcu štatistiky. Spisovateľ má teda kedykoľvek možnosť nahliadnuť na aktuálny stav dokumentu a ako s písaním pokročil. Medzi zobrazované údaje patrí počet znakov, počet slov a počet normovaných strán, ktoré text zaberá. Nakoľko sú tieto výpočty nutné len počas zobrazenia karty, je tento výpočet implementovaný v triede `StatisticsTask`, ktorá rozširuje triedu `Task` a výpočet tak prebieha na samostatnom vlákne nezaťažujúc beh aplikácie. `StatisticsTab` je treťou a poslednou triedou využívajúc pre svoju činnosť vlákna. Pri maximálnej záťaži môže teda proces s aplikáciou pracovať až so štyrmi vláknami. Prepojenie s GUI zabezpečuje binding a interval výpočtu pomocou funkcie `countStatistics()` je nastavený na 2 sekundy.

3.2.5 Triedy spravujúce okná

Niektoré funkcie aplikácie vyžadujú pre prehľadnosť vlastné systémové okno. Tie sú implementované ako samostatné triedy so samostatným definovaným FXML rozložením. Pre vytvorenie okna sa vytvorí nová inštancia triedy `Stage` a jeho obsah vyplní `Scene`. Pre jednotnosť všetky tieto triedy implementujú rozhranie `WindowModel` a jeho funkciami `openWindow()` a `closeWindow()`.



Obrázok 3.5: UML tried zobrazujúcich okná

Prvým oknom sú nastavenia v triede `Settings`, kde má užívateľ možnosť si aplikáciu prispôbiť. Je rozdelená na dve časti. V prvej sú k dispozícii nastavenia jazyka a vzhľadu, teda farieb, pozadia či skrývanie panelov. Pre zmenu jazyka využíva aplikácia modul `ResourceBundle`, ktorý je natívnou súčasťou Javy a umožňuje obmieňať reťazce v jazykoch slovensky, česky a anglicky. Druhá časť nastavení pozostáva z voľby písma, jeho farby a tiež nastavenie zvyrazňovania postáv a kapitola v texte, ktoré si užívateľ tak môže kedykoľvek vyplniť.

Ďalším oknom aplikácie je vyhľadávanie implementované triedou `Searching`. Pomocou regulárneho výrazu prehľadáva text a nájde všetky výskyty, medzi ktorými sa užívateľ následne preklikáva. Tieto výskyty sú označené a zamerané kurzorom.

Po spustení aplikácie sa ako prvé užívateľovi zobrazí okno triedy `IntroWindow` a umožní mu tak vybrať prvotnú akciu. Väčšina funkcií je totožná s funkciami triedy `Controller` a preto obsahuje iba referencie na tieto funkcie. Výnimočná je funkcia pokračovania v poslednom projekte, kedy načíta cestu k poslednému rozpracovanému projektu uloženú v triede `Configuration`.

Posledným oknom je nápoveda, implementovaná v triede `About`. Účelom tohto okna je iba zobrazenie informácií a nedisponuje teda žiadnou výnimočnou funkcionalitou.

3.2.6 Triedy pre prácu so súbormi

Prácu so súbormi spravuje trieda `FileStream`. Uchováva cestu k súboru, formát pomocou vymenovaného typu `FileFormat` a potrebné dáta. Pre výber súboru slúži metóda `openDialog()` využívajúca triedu `FileChooser` pre zobrazenie systémového okna pre výber súborov. Obdobne pracujú metódy `saveDialog()` a `exportDialog()`, pričom každá zobrazuje iba relevantné formáty. Pre čítanie súboru slúži metóda `readFile()`, ktorá prečíta dáta súboru podľa špecifikovaného formátu a metóda `writeFile()` pre zápis vygenerovaného súboru.

Trieda `FileStream` následne pracuje s triedami `StreamXML`, `StreamPlain`, `StreamPDF` a `StreamEPUB`, pričom tie sú implementované tak aby spracovali svoj formát a bolo s nimi možné pracovať jednotne. Za týmto účelom implementujú rozhrania `IFileStream` a `IExportStream`. Pre vytváranie dokumentov formátu PDF bolo nutné použiť externú knižnicu `iText` a pre formát EPUB zase knižnicu `Epublib`.

3.2.7 Generátor mien

Pre generátor mien bolo nutné použiť istý matematický model, nakoľko takýto generátor musí byť schopný generovať reálne použiteľné mená. Jedným z matematických systémov používaných v modernej teórii pravdepodobnosti na modelovanie stochastických procesov je Markovov reťazec. Využíva sa v množstve aplikácií a ponúka dostatočne výsledky pre jednoduchý generátor mien. Markovove reťazce môžeme opísať ako náhodný proces, ktorý má Markovovskú vlastnosť, teda pravdepodobnosť navštívenia nasledujúcich stavov je určená len aktuálnym stavom a je nezávislá na predošlých stavoch. Pre generovanie mien som zostavil nasledujúci algoritmus na základe vyššie zmienených Markovových reťazcov. [30]

Algoritmus pracuje tak, že postupne generuje meno, pričom aké nasledujúce písmeno bude použité je vybrané na základe vygenerovaných pravidiel. Tie vznikajú postupným prechodom zoznamu existujúcich mien písmeno po písmene, a toto písmeno pridá do zoznamu. Pri generovaní nového mena vyberie náhodné písmeno z tohto zoznamu, pričom písmená sa môžu v zozname aj opakovať a v takýchto situáciách sú iba vybrané s väčšou pravdepodobnosťou, čo odpovedá reálnemu výskytu. Algoritmus je naďalej optimalizovaný tak, aby obmedzil opakovanú postupnosť samohlások či spoluhlások na 2 pre slovenské a české mená a 3 pre anglické mená, keďže sa takáto postupnosť vyskytuje len zriedka. Navyše, zaznamenáva aj symbol začiatku a konca mien, čím so zvýšenou pravdepodobnosťou umiestňuje zvyčajné počiatkové a koncové písmená mien. Napriek týmto optimalizáciám však generátor nezvykol generovať typické mená a preto bola pridaná mechanika, ktorá s 50% pravdepodobnosťou vyberie meno zo zoznamu, namiesto generovania nového. Striedajú sa tak tradičné mená s menami vygenerovanými.

K dispozícii je 150 členný zoznam najpopulárnejších českých mužských a 151 ženských mien podľa štatistických údajov Ministerstva vnútra Českej republiky k roku 2016 [31]. Zoznam slovenských mien obsahuje najpopulárnejšie mená novorodencov z roku 2013 a to 114 chlapčenských a 126 dievčenských podľa údajov Ministerstva vnútra Slovenskej republiky [32]. Vybrané počty mien bola volené podľa hraničnej úrovne popularity. Zoznam anglických mien obsahuje 100 chlapčenských a 100 dievčenských najpopulárnejších mien novorodencov z roku 2014 podľa údajov Úradu pre národné štatistiky Anglicka a Walesu [33].

Trieda implementujúca Markove reťazce sa nazýva `NameGenerator`. Obsahuje niekoľko zoznamom z uvedených zdrojov, z ktorých si užívateľ v komponentoch typu `ComboBox` špecifikuje krajinný pôvod a pohlavie. V okamihu zmeny sa volaním metódy `buildChains()` iterovaním nad vybraným zoznamom vytvorí pravidlá Markovových reťazcov pre generovanie mien. Pri kliknutí na

tlačidlo generovania sa volá metóda `generateName()`, ktorá tieto pravidlá aplikuje, pričom maximálna dĺžka vygenerovaného mena môže byť 12 znakov.

3.2.8 Animácie

Technológia JavaFX poskytuje množstvo nástrojov pre vytváranie animácií. Tie pracujú paralelne, aby maximálne uľahčili prácu programátorovi. Pre potreby aplikácie je použitých niekoľko vertikálnych a horizontálnych animácií, pričom sú umiestnené v triede `Animation`. Tie boli vytvorené kombinovaním funkcií pre presuny pomocou triedy `PathTransition` a plynulé zmeny viditeľnosti použitím triedy `FadeTransition`. Zabezpečuje tak plynulé prechody pre zobrazovanie a skrývanie líšt a panelov.

3.3 Použité knižnice

Pre popise implementácie bolo spomenutých niekoľko knižníc, ktoré boli použité v situáciách kedy daná funkcionálna nebola natívnou súčasťou programovacieho jazyka Java a jej implementácia by značne predĺžila prácu na vyvíjanej aplikácii. Tieto knižnice budú v nasledujúcich kapitolách bližšie opísané, pričom všetky sú podľa licencií voľne dostupné pre nekomerčné účely.

3.3.1 RichTextFX

Základné komponenty pre prácu s textom v platforme JavaFX nepostačovali, preto bol použitý modul `RichTextFX`, ktorého autorom je Čech Tomáš Mikula. Ponúka niekoľko komponentov pre prácu s textom a využívaných na rôzne účely. Pre túto aplikáciu bol použitý `StyledTextArea`, keďže ako jediný podporuje vytváranie dynamických odkazov v texte. K ďalším dostupným komponentom patria `CodeArea`, ktorý je vhodný pre prácu s programovacím kódom a `InlineStyleTextArea` obsahujúci množstvo funkcií pre rozličné spracovanie jednotlivých riadkov textu. Je k dispozícii pod duálnou BSD 2-Clause License a GPLv2 with the Classpath Exception licenciou a teda je k dispozícii ako slobodný softvér. [34]

3.3.2 Epublib

Knižnica `Epublib` umožňuje čítať, vytvárať ale aj manipulovať so súbormi formátu EPUB, teda formátu elektronických kníh. Pre potreby vyvíjanej aplikácie je použité iba vytváranie dokumentov pomocou funkcií pre priame vytváranie kapitol, pridávanie textu a tiež vyplňanie údajov o autorovi, ktoré sa využívajú pri exporte. Pri pridávaní obsahu funkcie knižnice vyžadujú vstup typu `Resource` a teda je buď nutné pracovať so samostatnými súbormi, alebo je nutná konverzia textu na typ `byte`, čo sa využíva aj vo vyvíjanej aplikácii. Autorom je Holanďan Paul Siegmann a k dispozícii je pod GNU Lesser General Public License teda opäť dostupný ako slobodný softvér. [35]

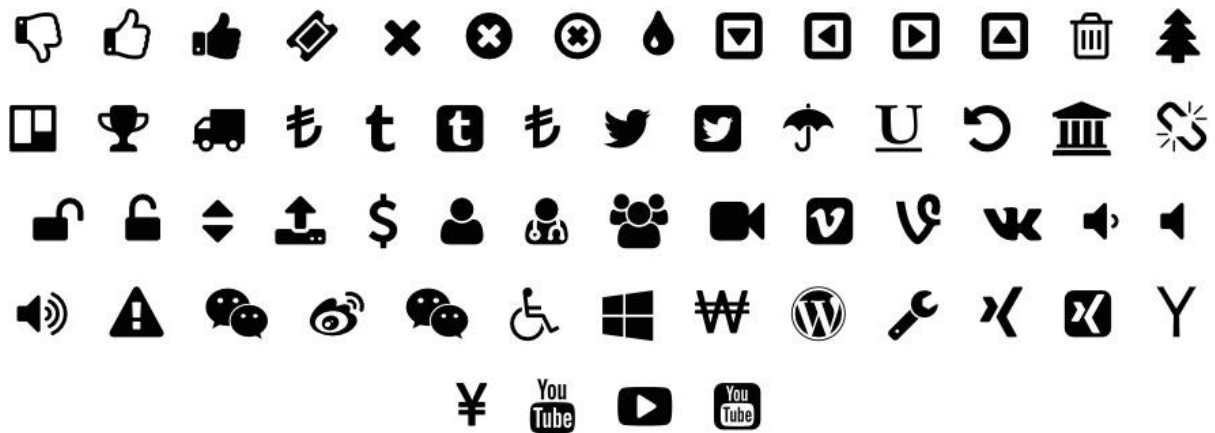
3.3.3 Modul iText

Pre zjednodušenie práce so súbormi formátu PDF, využívaných pri exporte, bola použitá knižnica `iText`. V Jave je pre prácu s formátom PDF dostupných niekoľko modulov, avšak obsahujú buď nedostatočnú funkcionálnu, alebo sú pre účely aplikácie príliš komplexné a implementáciu by tak mnohonásobne predĺžili. Niektoré vyžadovali aj zadávanie zlomu riadkov a strán, čo viedlo k rozhodnutiu použiť práve knižnicu `iText`, ktorá formátovanie textu rieši automaticky. Dokumenty

vytvára tak, že jednotlivé kapitoly pridáva ako Paragraph, so špecifikovaným nadpisom a obsahom. Tento modul od firmy iText Group NV je k dispozícii pod licenciou AGPL a teda voľne dostupný pre nekomerčné účely. [36]

3.3.4 Ikony Font Awesome

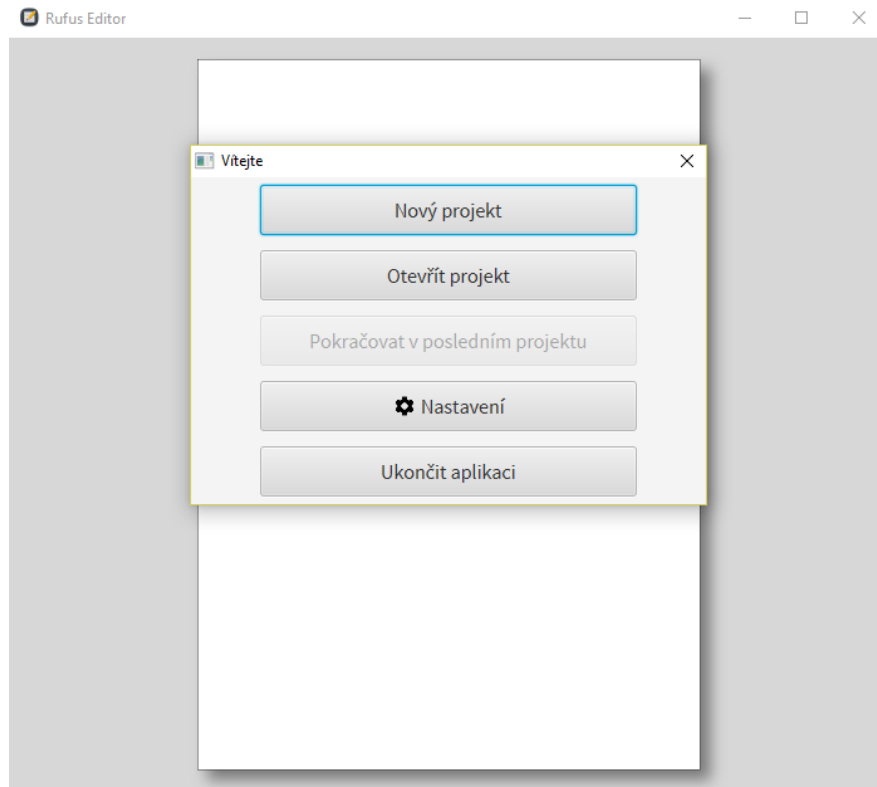
Pri návrhu aplikácie bolo špecifikované rozsiahle použitie ikon a tie teda bolo nutné zabezpečiť. Knižnica Font Awesome obsahuje súbor všeobecne známych a používaných ikon, pričom sú implementované vo forme vektorovej grafiky vďaka čomu vyzerajú ostro na rôznych monitoroch a rozlíšeníach. Aktuálna verzia obsahuje viac ako 600 takýchto ikon, pričom sem patria ikony používaných tlačidiel, symbolov, či značiek a na obrázku 3.6 sú ukážky niektorých z nich. Autorom je Dave Gandy a knižnica je dostupná pod licenciou SIL OFL 1.1 teda patrí medzi voľne dostupné písma. [37]



Obrázok 3.6: Ikony Font Awesome

3.4 GUI

Táto sekcia sa venuje implementácií GUI pomocou súborov FXML. Niektoré časti z návrhu v kapitole 2 budú následne zobrazené na obrázku, obsahujúci screenshot z výslednej aplikácie. Všetky nižšie uvedené obrázky zobrazujú aplikáciu v základných nastaveniach. Ako štýl je použitá *Modena*, ktorá je súčasťou platformy JavaFX, táto je mierne rozšírená v súbore `style.css`.



Obrázok 3.7: Screenshot spustenej aplikácie

3.4.1 Rozhranie aplikácie

Hlavné okno aplikácie tvorí množstvo elementov prekladaných cez seba. Ako kontajner bol teda zvolený `StackPanel`, ktorý túto hierarchiu podporuje. Na najnižšej vrstve sa nachádza komponent `Rectangle` tvoriaci papier na pozadí. Vo vyšších vrstvách je ľavý panel a následne panel v hornej časti aplikácie. Tlačidlá jednotlivých panelov sú zoradené pomocou komponentov `VBox` a `HBox` vďaka čomu si aplikácia zachováva rozloženie aj pri zmenách pomeru strán. Karty pravého panelu sú obsiahnuté v kontajneroch `AnchorPane`, pričom vždy je viditeľná práve jedna a užívateľ tak môže prepínať medzi kartami.



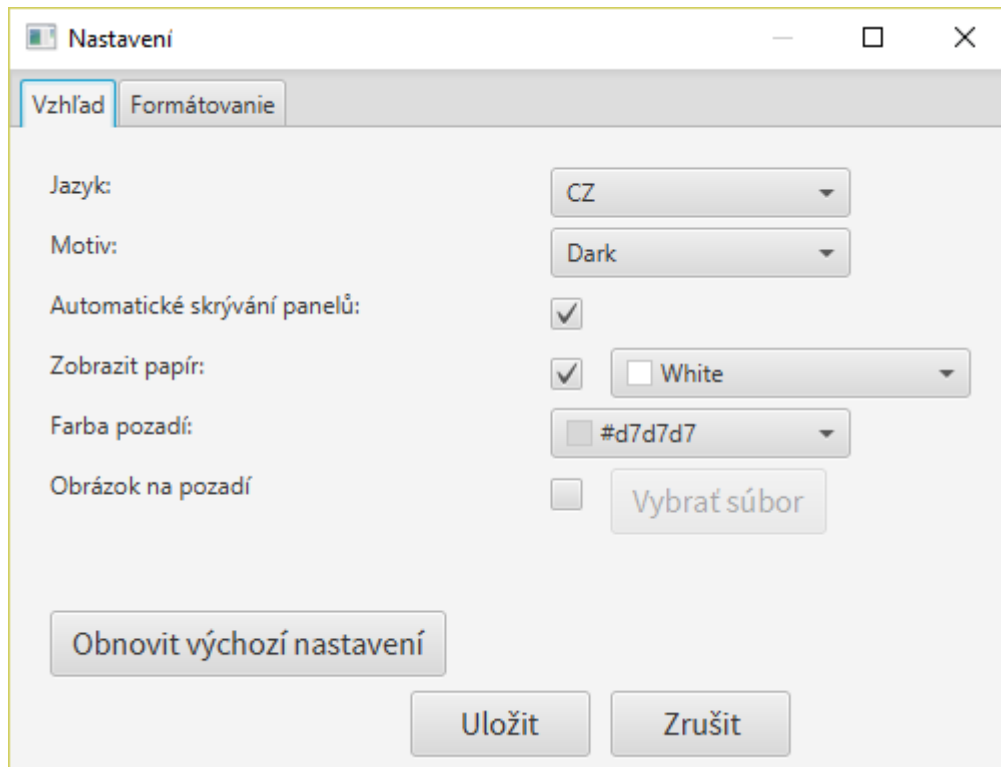
Obrázok 3.8: Screenshot rozhrania aplikácie

Pre správne fungovanie bolo tiež nutné niekoľko základných komponentov platformy JavaFX rozšíriť. Patrí sem tlačidlo `TopMenuButton` a `TopButton` pre hornú lištu a tiež `TabButton` pre pravú lištu, pridávajúc možnosť zopnutia tlačidla. Ďalším je tlačidlo `BasicButton` pre tlačidlá kariet pridávajúc napojenie na ikony Font Awesome. Okrem tlačidiel sem patrí `LimitedTextField`, obmedzujúc dĺžku vloženého textu na 40 znakov a `NumericTextField` zabraňujúc vkladaniu nečíselných znakov, čo sa využíva pri zadávaní veku postáv.

Pri návrhu aplikácie patrila k požiadavkám aj možnosť skryť či presunúť scrollbar pri textovej ploche, nakoľko patrí medzi rušivé elementy. V technológii JavaFX ide o jednoduchú úpravu, ktorou disponuje väčšina komponent pracujúcich s textom. Pri implementácii však bol použitý modul `StyledTextArea` čo prinieslo aj niekoľko nevýhod, vrátane chýbajúcej dokumentácie. Scrollbar je vytvorený pomocou štruktúry `VirtualizedScrollPane`, ktorá nie je prístupná a teda ju nie je ani možné modifikovať. Vo výsledku sa teda implementovať skrytie a presúvanie scrollbaru nepodarilo.

3.4.2 Okná aplikácie

Ostatné okná aplikácie tvorí kontajner `AnchorPane` a v ňom sú nadefinované pozície jednotlivých komponentov. Výnimkou je okno nastavení, ktoré je rozdelené na dve časti pomocou kontajnera `TabPane` zobrazeného na obrázku 3.9. V prvej časti si užívateľ definuje celkový vzhľad aplikácie a v druhej špecifikuje použité písmo aplikované na text.



Obrázok 3.9: Screenshot nastavení aplikácie

4 Testovanie

Táto kapitola sa zameriava na testovanie implementovanej aplikácie. Obsahuje teda subjektívne názory a spätné väzby niekoľkých užívateľov, ktorý sa testovania zúčastnili. Hlavným testovaným prvkom bolo užívateľské rozhranie, ktoré prešlo v priebehu vývoja niekoľkými rôznymi úpravami. V prvotných fázach sa počas implementácie niektoré časti návrhu mierne upravovali s cieľom dosiahnuť lepší výsledok. Ďalších niekoľko úprav pribudlo počas konzultácií s vedúcim práce, pričom na rozhodnutí uplatniť tieto zmeny sme sa zhodli.

4.1 Testovanie použiteľnosti

V prvej fáze testovania bolo oslovených niekoľko užívateľov pre otestovanie aplikácie, pričom ich činnosť bola sledovaná pri práci s aplikáciou. Dôraz sa kládol na ich prvé dojmy z aplikácie, pohyby myši a rýchlosť s akou užívatelia dokázali nájsť a použiť potrebné funkcie. Tohto testu sa zúčastnilo 5 užívateľov z rôznych odvetí a rôznych vekových kategórií, nakoľko je podľa expertov na testovanie je tento počet dostačujúci [38]. Počas tejto fázy obdržala aplikácia prevažne pozitívne reakcie, čím môžeme zhodnotiť návrh GUI aplikácie ako vydarený. Mierne úpravy však boli vykonané, ako napríklad presun možností formátovania textu z pravého panelu do nastavení.

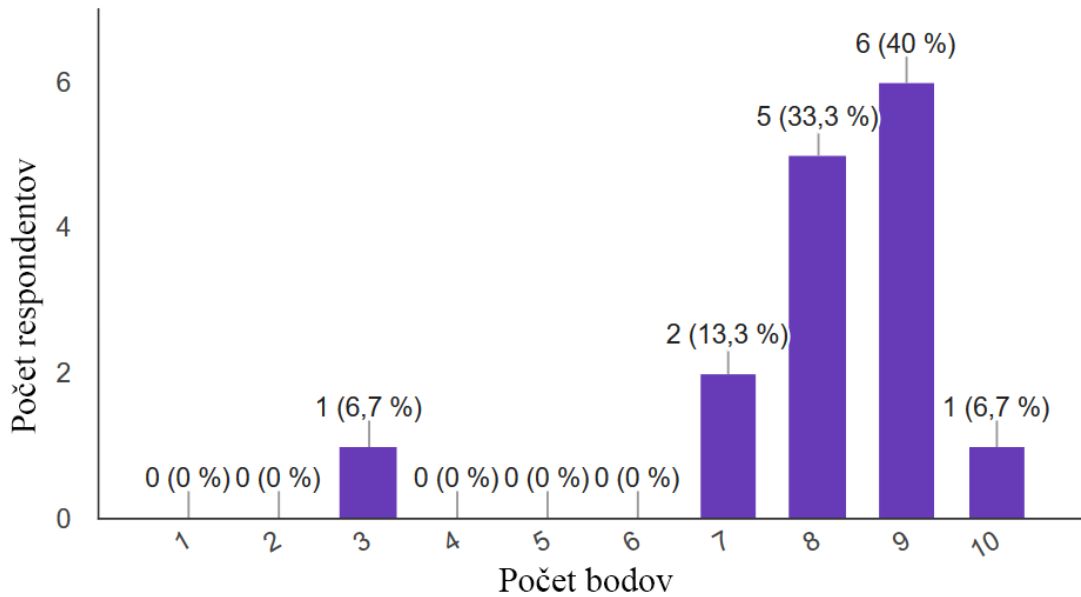
V druhej fáze mali užívatelia možnosť si aplikáciu vyskúšať samostatne. Bola zverejnená vo forme beta verzie na webovej stránke umiestnenej na školskom servere Eva [39]. Testerov tvorili ľudia s rôznou úrovňou znalosti domény, s rôznou úrovňou znalosti vývoja softwaru, s rôznym stupňom dosiahnutého vzdelania a rôzneho veku. Do testovania sa zapojili aj študenti žurnalistiky ako aj spisovatelia amatéri. Po vyskúšaní aplikácie vyplnili krátky dotazník, ktorý je dostupný v prílohe číslo 1 a jeho výsledky sú prezentované v nasledujúcej tabuľke. Formulár bol vytvorený pomocou nástroja Google Forms a maximum bodov, ktoré bolo možné v otázkach získať bolo 10 a testu sa zúčastnilo 15 účastníkov. Výsledky bodových otázok tohto formulára sú spriemerované v tabuľke 4.1. [40]

Dojem	7,26
GUI	8,06
Funkcie	8,33
Splnenie očakávaní	8
Celkové hodnotenie	8,06

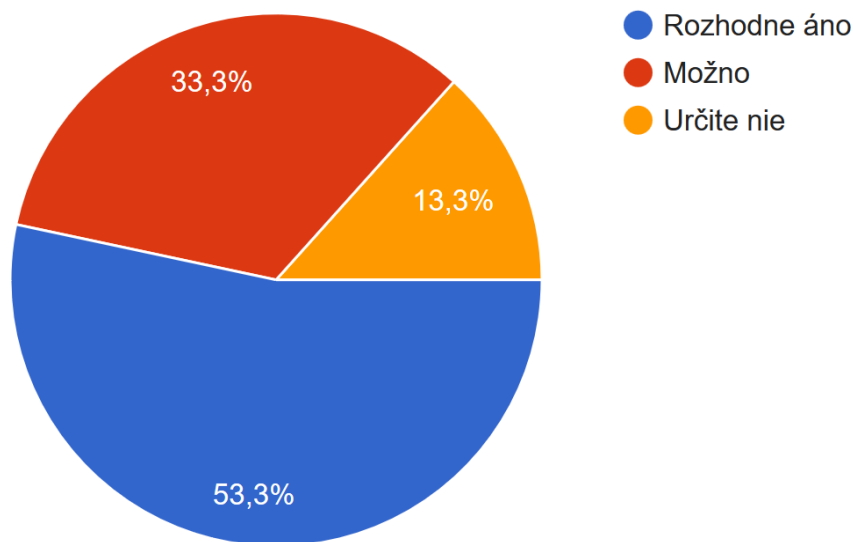
Tabuľka 4.1: Výsledky bodových otázok formulára

4.2 Výsledky testovania

Pri testovaní sa tiež kládol dôraz na správne zobrazenie a ovládanie. Napriek mottu Javy „raz napíš a spusti kdekoľvek“ sa na technológiu JavaFX vzťahuje len čiastočne a viaceré funkcie nepracovali korektne pri testovaní na rôznych platformách. Kľúčové z nich sa podarilo opraviť, avšak drobné chyby pri behu na niektorých platformách sa stále môžu vyskytnúť, nakoľko by bola oprava všetkých z nich príliš časovo náročná. Výskyt prípadných chýb by však nemal narušiť stabilitu programu.



Obrázok 4.2: Graf celkového hodnotenia



Obrázok 4.3: Graf výsledkov formulára o ďalšom použití aplikácie

Výsledky testovania priniesli hneď niekoľko výhod. Poukázali na niekoľko drobných chýb v logike aplikácie, ktoré sa následne opravili. Dojmy užívateľov boli v celku pozitívne, ako aj ukazujú výsledky formulára v tabuľke 4.1 a na obrázku 4.2. Na základe spätnej väzby bol tiež pridaný popis použitia kapitol, ktorý zoznámí nových užívateľov s automatickým zvýrazňovaním textu. Na druhej strane výsledky súčasne poukazujú na nedostatky. V porovnaní s ostatnými otázkami obdržal prvý dojem aplikácie nižšie hodnotenie, čo značí že nový užívateľ potreboval istý čas aby sa v aplikácii zorientoval. Okrem toho však sú hodnotenia kladné a užívateľov zaujali najmä dostupné funkcie. Je teda možné usúdiť, že aplikácia bola navrhnutá dobre a špecifikované funkcie boli implementované úspešne. Navyše, ako je zobrazené v grafe na obrázku 4.3, vo formulári viac ako 53% zúčastnených odpovedali, že by mali záujem aplikáciu využívať aj namiesto už existujúceho riešenia, čo značí o možnom využití aplikácie.

5 Záver

Cieľom tejto bakalárskej práce bolo vytvorenie reálnej aplikácie pre tvorivé písanie, čo úzko súvisí s preskúmaním funkcií už existujúcich aplikácií a následne naštudovaním si princípov programovania v technológii JavaFX. Spolu s tým bolo taktiež potrebné preštudovať si a implementovať prácu s rôznymi knižnicami a súborovými formátmi. Výsledkom je aplikácia, ktorá úspešne implementuje všetky kľúčové funkcie z návrhu v kapitole 3, schopná pracovať ako plnohodnotný textový procesor pre spisovateľov. Ponúka správu kapitol, postáv, poznámok a taktiež disponuje štatistikami a cieľmi. Umožňuje export do niekoľkých formátov a obsahuje množstvom užitočných funkcií, akými je napríklad generátor náhodných mien. Aj napriek širokej škále možností, ktoré aplikácia ponúka, nechýba minimalistické viacjazyčné rozhranie s možnosťami prispôsobenia.

Finálna aplikácia je nazvaná Rufus. Toto meno poniesie z úctu k spisovateľom, akým bol slovenský autor Milan Rúfus, ktorý svojou tvorbou prispel ku kultúrnemu rozvoju krajiny. Tento autor aj v dnešnej dobe inšpiruje množstvo spisovateľov a bude vzorom pre ďalšie generácie.

Aj keď aplikácia spĺňa kladené požiadavky, jej priestor na vylepšenie je otvorený. Pri testovaní sa ukázalo, že viaceré funkcie nepracovali korektne pri testovaní na rôznych platformách. Kľúčové z nich sa podarilo opraviť, avšak drobné chyby pri behu na niektorých platformách sa môžu objaviť aj naďalej, nakoľko bola aplikácia vyvíjaná a testovaná primárne na operačnom systéme Windows. Na vyriešenie tohto problému by bolo nutné rozsiahle testovanie na čo najväčšom počte rôznych platforiem. Okrem tohto problému má však aplikácia značný potenciál a ďalší vývoj bude smerovať na predstaveniu aplikácie spisovateľom a jej vylepšeniu na základe spätnej väzby. Medzi ďalšie potenciálne rozšírenia patrí kontrola pravopisu, podpora viacerých jazykov a rozšírenie ponuky formátov pri exporte. Možnosti sú aj v pridaní ďalších volieb personalizácie, klávesových skratiek, či importu mien pre generátor. Prehľadnejšiu prácu s textom by umožnila funkcia skrytia odstavcov či kapitol, ktoré by sa tak aj vyčlenili z exportu. Spisovateľ by tiež mohol nájsť využitie importu postáv z iných diel.

Literatura

- [1] Microsoft Corporation. Volba Office [online]. 2016 [vid. 07.05.2016]. Dostupné z: <<https://products.office.com/cs-cz/compare-microsoft-office-products>>
- [2] Apache Software Foundation. Apache OpenOffice Writer [online]. 2016 [vid. 07.05.2016]. Dostupné z: <<https://www.openoffice.org/sk/product/writer.html>>
- [3] The Document Foundation. LibreOffice Writer [online]. 2016 [vid. 11.05.2016]. Dostupné z: <<https://www.libreoffice.org/discover/writer>>
- [4] BORBOËN Nicolas. Markdown [online]. 2013 [vid. 2016-05-13]. ISSN: 1420-7192. Dostupné z: <<https://infoscience.epfl.ch/record/183846/files/1-13-page5.pdf>>
- [5] Ulysses. The Ultimate App for Anyone Who Writes [online]. 2016 [vid. 08.05.2016]. Dostupné z: <<http://www.ulyssesapp.com>>
- [6] Ulysses. Z: UlyssesApp [online]. © 2016 Ulysses. [vid. 08.05.2016]. Dostupné z: <<http://www.ulyssesapp.com/features>>
- [7] Literature and Latte. Scrivener [online]. 2016 [vid. 11.05.2016]. Dostupné z: <<https://www.literatureandlatte.com/scrivener.php>>
- [8] Literature and Latte. Z: Scrivener [online]. © 2016 Scrivener. [vid. 11.05.2016]. Dostupné z: <<https://www.literatureandlatte.com/scrivener.php>>
- [9] Graeme Gott. FocusWriter [online]. 2016 [vid. 11.05.2016]. Dostupné z: <<http://gottcode.org/focuswriter>>
- [10] Graeme Gott. Z: FocusWriter [online]. © 2016 Graeme Gott. [vid. 11.05.2016]. Dostupné z: <<http://gottcode.org/focuswriter>>
- [11] BERNAL, Joaquín. Q10 [online]. 2016 [vid. 06.05.2016]. Dostupné z: <<http://www.baara.com/q10>>
- [12] Writershelpers. Z: Write In Peace With These Distraction-Free Editors [online]. © 2016 Writers Helpers. [vid. 06.05.2016]. Dostupné z: <<http://writershelpers.tumblr.com/post/40742549483/write-in-peace-with-these-distraction-free-editors>>
- [13] OmmWriter. Ommwriter [online]. 2016 [vid. 08.05.2016]. Dostupné z: <<http://www.ommwriter.com/en>>
- [14] WhiteMonkey. Whitemonkey [online]. 2016 [vid. 05.05.2016]. Dostupné z: <<http://writemonkey.com/index.php>>
- [15] WhiteMonkey. Z: Whitemonkey [online]. © 2016 Whitemonkey. [vid. 05.05.2016]. Dostupné z: <<http://writemonkey.com/index.php>>
- [16] Wikipédia. API [online]. 2016 [vid. 05.05.2016]. Dostupné z: <<https://cs.wikipedia.org/wiki/API>>
- [17] René Nekuda. Ulysses: nejlepší textový editor pro spisovatele [online]. 2016 [vid. 02.05.2016]. Dostupné z: <<http://www.renenekuda.cz/ulysses-nejlepsi-textovy-editor-pro-spisovatele>>
- [18] Wikipédia. Mockup [online]. 2016 [vid. 06.05.2016]. Dostupné z: <<https://en.wikipedia.org/wiki/Mockup>>
- [19] EverCoder Software SRL. Mockup [online]. 2016 [vid. 06.05.2016]. Dostupné z: <<https://moqups.com>>
- [20] The Qt Company. Qt Documentation [online]. 2016 [vid. 09.05.2016]. Dostupné z: <<http://doc.qt.io/qt-5/index.html>>
- [21] James L Weaver. Pro JavaFX 2 : a definitive guide to rich clients with java technology. Vyd. 1. Lexington : Apress, 2012, 620 s. ISBN 978-1-4302-6872-7

- [22] Wikipédia. Windows Presentation Foundation [online]. 2016 [vid. 06.05.2016]. Dostupné z: <https://en.wikipedia.org/wiki/Windows_Presentation_Foundation>
- [23] Python Software Foundation. Tkinter — Python interface to Tcl/Tk [online]. 2016 [vid. 08.05.2016]. Dostupné z: <<https://docs.python.org/2/library/tkinter.html>>
- [24] Oracle Corporation. Learn About Java Technology [online]. 2016 [vid. 05.05.2016]. Dostupné z: <<https://java.com/en/about>>
- [25] JetBrains. IntelliJ IDEA [online]. 2016 [vid. 03.05.2016]. Dostupné z: <<https://www.jetbrains.com/idea>>
- [26] Violet UML Editor. Violet UML Editor [online]. 2016 [vid. 02.05.2016]. Dostupné z: <<http://alexdp.free.fr/violetumleditor/page.php>>
- [27] Wikipédia. Model-view-controller [online]. 2016 [vid. 05.05.2016]. Dostupné z: <<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>>
- [28] Wikipédia. Z: Model-view-controller [online]. © 2016 Wikipédia. [vid. 05.05.2016]. Dostupné z: <<https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>>
- [29] Raoul-Gabriel Urma, Mario Fusco, Alan Mycroft. Java 8 in action : lambdas, streams, and functional-style programming. Vyd. 1. Shelter Island, NY : Manning, 2015, 394 s. ISBN 978-1-617291-99-9
- [30] MEYN, Sean. Markov chains and stochastic stability. Vyd. 1. Cambridge : Cambridge University Press, 2009, 594 s. ISBN 978-0-521-73182-9
- [31] Ministerstvo vnitra České republiky. Četnost jmen a příjmení [online]. 2016 [vid. 07.05.2016]. Dostupné z: <<http://www.mvcr.cz/clanek/cetnost-jmen-a-prijmeni-722752.aspx>>
- [32] Ministerstvo vnútra Slovenskej republiky. Najčastejším menom, ktorým rodičia pomenovali svoje dieťa v roku 2013, je Jakub [online]. 2016 [vid. 07.05.2016]. Dostupné z: <<http://www.minv.sk/?tlacove-spravy&sprava=najcastejsim-menom-ktorym-rodicia-pomenovali-svoje-dieta-v-roku-2013-je-jakub>>
- [33] Office for National Statistics. Baby Names - England and Wales: 2014 [online]. 2016 [vid. 07.05.2016]. Dostupné z: <<http://www.ons.gov.uk/peoplepopulationandcommunity/birthsdeathsandmarriages/livebirths/bulletins/babynamesenglandandwales/2015-08-17>>
- [34] MIKULA, Tomáš. RichTextFX [online]. 2016 [vid. 01.05.2016]. Dostupné z: <<https://github.com/TomasMikula/RichTextFX>>
- [35] SIEGMANN, Paul. Epublib [online]. 2016 [vid. 02.05.2016]. Dostupné z: <<http://www.siegmann.nl/epublib>>
- [36] iText Software. iText [online]. 2016 [vid. 03.05.2016]. Dostupné z: <<https://github.com/itext/itextpdf>>
- [37] Font Awesome. The iconic font and CSS toolkit [online]. 2016 [vid. 01.05.2016]. Dostupné z: <<http://fontawesome.io>>
- [38] VIRZI, Robert A. Refining the test phase of usability evaluation: how many subjects is enough? *Human Factors*. Vol. 34, vyd. 4, 1992, s. 547-568. ISSN 0018-7208
- [39] MAGDOLEN, Patrik. Rufus [online]. 2016 [vid. 03.05.2016]. Dostupné z: <<http://www.stud.fit.vutbr.cz/~xmagdo01/Rufus/index.html>>
- [40] Google. Formuláre Google [online]. 2016 [vid. 04.05.2016]. Dostupné z: <<https://www.google.com/intl/sk/forms/about>>

Seznam příloh

Příloha 1. Obsah DVD

Příloha 2. Testovací formulář

Příloha 1. Obsah DVD

Priložené DVD obsahuje následující strukturu:

- source zložka obsahující zdrojový kód
- Rufus.jar spustitelný soubor ve formátu JAR
- bp-xmagdo01.pdf text práce ve formátu PDF
- bp-xmagdo01.docx text práce ve formátu DOCX

Příloha 2. Testovací formulář

Testovanie vyvíjanej aplikácie

* Povinné

1. Ako by ste ohodnotili dojem, ktorý na vás aplikácia spravila pri prvom spustení? *

Označte iba jednu elipsu.

	1	2	3	4	5	6	7	8	9	10	
Vôbec ma nezaujala	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Veľmi ma zaujala

2. Ako na vás pôsobilo grafické užívateľské rozhranie? *

Označte iba jednu elipsu.

	1	2	3	4	5	6	7	8	9	10	
Nepraktické a rušivé	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Praktické a nerušivé

3. Ako ste boli spokojný s funkciami, ktorými aplikácia disponuje? *

Označte iba jednu elipsu.

	1	2	3	4	5	6	7	8	9	10	
Kľúčové funkcie chýbajú	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Obsahuje všetky potrebné

4. Splnila aplikácie vaše očakávania? *

Označte iba jednu elipsu.

	1	2	3	4	5	6	7	8	9	10	
Nesplnila	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Splnila

5. Mali by ste záujem túto aplikáciu používať aj napriek iným ktoré trh ponúka? *

Označte iba jednu elipsu.

- Rozhodne áno
 Možno
 Určite nie

6. Narazili ste na niečo čo vám prekážalo? Ak áno, popíšte čo to bolo?

.....
.....

7. Narazili ste na niečo čo vám chýbalo a uvítali by ste to v aplikácií? Ak áno, popíšte o aký prvok ide.

.....
.....

8. Koľkými bodmi by ste hodnotili celkovú aplikáciu? *

Označte iba jednu elipsu.

	1	2	3	4	5	6	7	8	9	10	
Minimum bodov	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Maximum bodov