Technická dokumentácia projektu

(Textový editor obohatený o grafické prvky)

*Tímový projekt*

|  |  |
| --- | --- |
| **Autor:** | tím č.10 – Innovators |
| **Téma projektu:** | textový editor obohatený o grafické prvky (TrollEdit) |
| **Vytvorený:** | 02.10. 2011 |
| **Stav:** | predbežný |
| **Vedúci projektu:** | Ing. Peter Drahoš |
| **Vedúci tímu:** | Bc. Lukáš Turský |
| **Členovia tímu:** | Bc. Marek Brath  Bc. Adrián Feješ  Bc. Maroš Jendrej  Bc. Jozef Krajčovič  Bc. Ľuboš Staráček |
| **Kontakt:** | tp-team-10@googlegroups.com |

Obsah

[1 Úvod 4](#_Toc307515575)

[2 Analýza 5](#_Toc307515576)

[2.2 Analýza predchádzajúceho riešenia 5](#_Toc307515577)

[2.1 Analýza použitých technológií 5](#_Toc307515578)

[3 Špecifikácia požiadaviek 6](#_Toc307515579)

[3.1 Funkcionálne požiadavky 6](#_Toc307515580)

[3.2 Nefunkcionálne požiadavky 7](#_Toc307515581)

[4 Návrh riešenia 8](#_Toc307515582)

[5 Implementácia prototypu 12](#_Toc307515583)

[5.1 Popis prototypu 12](#_Toc307515584)

# 1 Úvod

Súčasne textové editory zdrojových kódov skoro vôbec nevyužívajú možnosti grafickej reprezentácie, čo je veľká škoda vzhľadom na to, že práve obohatenie editorov o grafické prvky by mohlo nielen sprehľadniť zdrojový kód, ale aj zjednodušiť a zefektívniť jeho tvorbu, údržbu a prezentáciu, a priniesť možnosť nového pohľadu na integráciu dokumentácie s programom.

Cieľom tohto projektu bude pokračovať vo vývoji multiplatformového editora „TrollEdit“ (ktorý bol riešení v roku 2009/10 tímom s názvom UFOPAK) pre úpravy zdrojových kódov, ktorý bude využívať grafické prvky na zjednodušenie a zefektívnenie práce programátora. Našim zameraním bude rozšírenie stávajúcej funkcionality do podoby vhodnej pre reálne nasadenie editora do praxe

Tento dokument obsahuje zhrnutie všetkých riešení nášho tímu na tomto projekte od analýzy až po implementáciu.

# 2 Analýza

Vzhľadom na to, že pokračujeme na projekt, ktorý bol vyvíjaný v rámci minuloročného tímového projektu bolo nutné vykonať analýzu predchádzajúceho riešenia.

//Sem by mohli isť aj nejaké porovnania súčasných editorov (Scite, JuffEed etc.) tak stručne //o editore klady a zápory a zhodnotenie.

//Nakoniec nejaká tabuľka kde budú porovnania funkcionality

## 2.2 Analýza predchádzajúceho riešenia

Čo by tam mohli byť:

Klady:

Zápory:

Analýza nájdených nedostatkov resp. chýb

Zhodnotenie

Vytvorí Lukáš a Adrián

## 2.1 Analýza použitých technológií

Aké použili aké technológie a prečo?

Klady a zápory:

Prečo ich aj my použijeme

Zhodnotenie

Vytvorí Ľuboš

# 3 Špecifikácia požiadaviek

Keďže vychádzame už z existujúceho multiplatformového textového editora obohateného o grafické prvky „TrollEdit“, popisujeme iba tie požiadavky na systém, ktoré chceme implementovať pripadne modifikovať v súčasnej verzií.

Hlavne ciele tohto projektu sú:

* Rozšíriť súčasne implementovanú funkcionalitu
* Modifikácia používateľského rozhrania GUI
* Vytvoriť kvalitný produkt, ktorý bude úspešný a mohol by presadiť aj v praxi

## 3.1 Funkcionálne požiadavky

Pre TrollEdit boli identifikovane funkcionálne požiadavky na základe dôkladnej analýzy predchádzajúceho riešenia a taktiež od nášho vedúceho tímu, ktoré sú spísané nasledujúcej tabuľke.

Tab. Funkcionálne požiadavky

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Požiadavka** | **Charakteristika** | **Priorita** |
| F01 | Možnosť Undo/Redo | možnosť vrátiť zmeny naspäť a opačne | vysoká |
| F02 | Podpora skratiek v editore (Shortcuts) | možnosť spustiť funkcie programu pomoc klávesových skratiek | vysoká |
| F03 | Dopytovanie sa do Lua |  | vysoká |
| F04 | Podpora paralelizmu | syntakticky strom by bežal pod vlastným vláknom a program pod vlastným | vysoká |
| F05 | 2 módy písania |  | vysoká |
| F06 | Nastavenie programu | možnosť rozšírených nastavení priamo v editore | stredná |
| F07 | Podpora intellisense | rozpoznávania bežných kľúčových slov programovacích jazykov ale aj najčastejšie používané bloky kódu, ako sú napríklad funkcie, cykly, podmienky | stredná |
| F08 | Rozšírenie funkcionality | možnosť rozširovať funkcionalitu pomocou zásuvných modulov | stredná |
| F09 | Vyhľadávanie | určitý druh fulltextového vyhľadávania s prípadnou optimalizáciou pre najčastejšie vyhľadávané výrazy | stredná |
| F10 | Export súborov | Možnosť exportovania súboru do iných formátov (napr. XML, WORD) | nízka |
| F11 | Podpora sw metrík | schopnosť detegovať určite ukazovatele v zdrojovo kóde ako index udržovateľnosti, cyklomatická zložitosť, hodnoty fan in a fan aut, ktoré by boli zobrazene v určitej tabuľke prípadne vizualizované v podobe grafov | nízka |
| **Legenda:**  Vysoká priorita – nevyhnutia funkcia systému, je základom funkcionality systému.  Stredná priorita – funkcia, ktorú možno implementovať neskôr netvorí základ funkcionality systému.  Nízka priorita – funkcionalita bude implementovaná v ďalších verziách | | | |

## 3.2 Nefunkcionálne požiadavky

Pre TrollEdit boli identifikovane nasledujúce nefunkcionálne požiadavky pre správne zabezpečenie fungovania programu.

Tab. Nefunkcionálne požiadavky

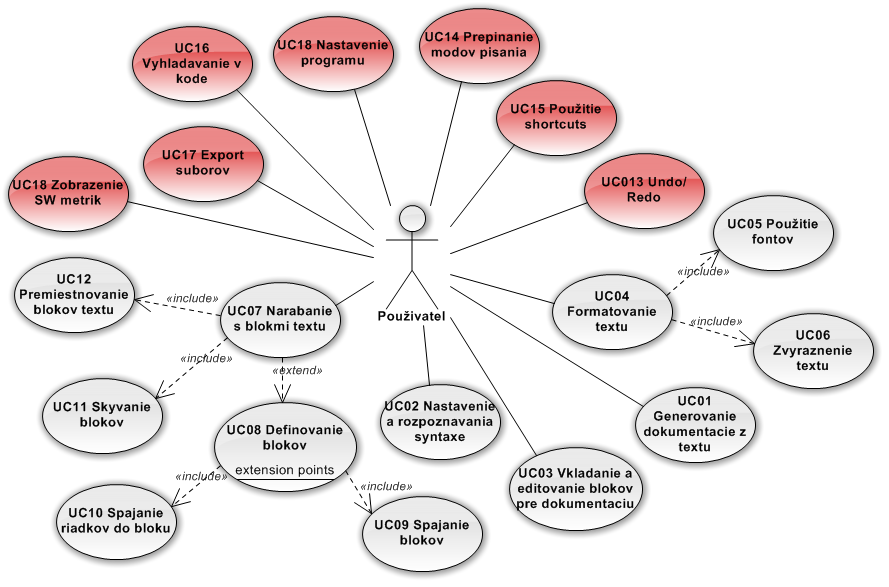
|  |  |  |
| --- | --- | --- |
| **ID** | **Požiadavka** | **Charakteristika** |
| N01 | Rýchlosť a spoľahlivosť | zrýchlenie programu hlavne čo sa týka parsovania, program by mal byť schopný pracovať aj na menej výkonnom hardvéri |
| N02 | Modulárnosť | možnosť rozširovania jeho funkcii pomocou dodatočnej implementácie nových modulov. Tým pádom nie je v zásade nutné zasahovať do samotnej implementácie systému pri rozširovaní jeho funkcionality |
| N03 | Redesign používateľského rozhrania GUI | musí byť jednoduché a prehľadné, pričom najčastejšie funkcie  systému by mali byť prístupné používateľovi bez náročného hľadania |

# 4 Návrh riešenia

V tejto kapitole je popísaný návrh programu TrollEdit podľa požiadaviek definovaných v predchádzajúcej kapitole. Funkcionálne požiadavky sa premietnu do diagramu prípadov použitia a nefunkcionálne do architektúry systému.

## 4.1 Diagram prípadov použitá

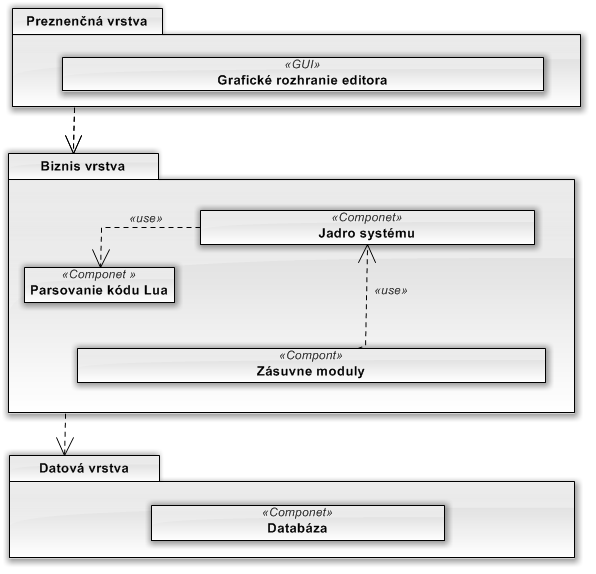
Na diagrame sú znázornené prípady použitia popisujúce funkcionalitu, ktorá je súčasne implementovaná v programe TrollEdit a taktiež novu funkcionalitu, ktorú sme identifikovali na základe analýzy. Nové prípady použitia sú odlíšené od tých starších červenou farbou.



Obr. Diagram prípadov použitia

## 4.2 Architektúra programu

Architektúra programu bude postavená na klasickom trojvrstvom princípe t.j. rozdelená na prezenčnú, biznis a dátovú vrstvu vid. .



Obr. Architektúra programu

V prezenčnej vrstve budú implementované triedy pre grafické rozhranie editora od hlavného menu až po nápovedu. Prezenčná vrstva bude komunikovať s biznis vrstvou v ktorej bude spracovaná logika programu.

Biznis vrstva sa bude skladať z troch komponentov jeden pre jadro systému kde bude implementovaná základná funkcionalita programu, druhy pre parsovanie zdrojového kódu kde sa bude vytvárať AST v skriptovacom jazyku Lua a tretí pre pridávanie novej funkcionality, ktorá nenaruší základnú funkcionalitu jadra programu.

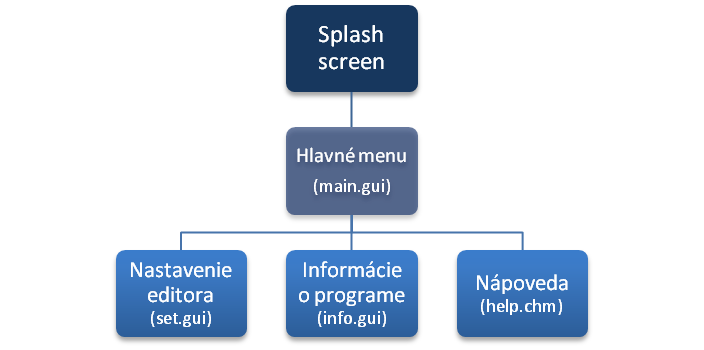
V dátovej vrstve budú dáta ktoré si bude program ukladať ako nastavenie programu dočasnú históriu zmien nad zdrojovým kódom a údaje o projekte.

//Refactoring použit nejake vzory ktore by sa nam tam hodily

//Nejaky diagram tried

## 4.3 Návrh GUI

Na základe analýzy sme identifikovali 5 okien, ktoré budú v programe implementovane, tieto okna budú mať hierarchicky význam t.j. okno na vyššej úrovni môže volať iba okna na nižšej úrovni nie však reverzne.



Obr. Hierarchické rozdelenie okien programu

**Splash screen -** úvodné modálne okno, ktoré sa zobrazí vždy pri spustení TrollEditu.

**Hlavne menu -** nemodálne okno, ktoré slúži ako hlavne menu programu, z tohto okna je možne volať iné okna.

**Nastavenie editora -** modálne okno, ktoré slúži pre detailne nastavenie editora.

**Informácie o  programe -**  modálne okno, ktoré zobrazuje informácie o programe ako popis programu, dátum vytvorenia, verzia programu.

**Napovedá -** nemodálne okno pre zobrazenie nápovedi programu štandardný súbor .chm.



Obr. Splash screen programu



Obr. Hlavné menu programu

# 5 Implementácia prototypu

Táto kapitola popisuje implementáciu systému t.j. prevedie návrhu do výsledného funkčného kódu.

## 5.1 Popis prototypu

# 

**Inkrementálny a iteratívny vývoj**

* Analýza, špecifikácia požiadaviek a hrubý návrh (zimný semester)
  + *Úvod - o čom je tento dokument, ciele, ohraničenia.*
  + *Analýza problému a špecifikácia riešenia*(pre tvorbu softvérového systému typicky zahŕňa tieto časti: Kontext systému, Špecifikácia funkcií systému (určí sa aj priorita pre jednotlivé funkcie), Špecifikácia údajov v systéme, Špecifikácia správania systému)
  + *Hrubý návrh riešenia*
  + *Ďalšie požiadavky a ohraničenia*
* Prototyp (zimný semester)
  + Cieľ prototypovania, dosiahnuté výsledky
  + Podľa dohody s pedagógom, odporúča sa používateľská príručka (pre celý systém)
* Produkt a dokumentácia k produktu (letný semester)
  + Stanoví sa podľa povahy projektu. Štandardne zahŕňa tieto časti:
    - *Používateľská príručka*
    - *Systémová príručka* (spolu s návodom na inštaláciu)
* Návrh, implementácia a overenie riešenia (letný semester)
  + *Zapracovanie nedostatkov špecifikácie a hrubého návrhu*
  + *Návrh systému*(pre tvorbu softvérového systému typicky zahŕňa tieto časti: Architektúra systému, Fyzický model údajov systému, Návrh algoritmov spracovania)
  + *Ohraničenia, zmeny špecifikácie, priority riešenia*
  + *Výber implementačného jazyka a prostredia*
  + *Opis realizácie (implementácie jednotlivých modulov, napr. zaujímavé veci, optimalizácia, doplnenia oproti návrhu,...)*
  + *Overenie výsledku (určenie spôsobu overenia výsledku, postup, testovacie údaje, ak sa zmenili oproti návrhu)*
  + *Záznam o používaní systému*
  + *Čo sme nestihli*
  + *Čo sme sa naučili*