

FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY  
UNIVERZITA KOMENSKÉHO V BRATISLAVE

**Záverečná správa**

Anotátor pomenovaných entít  
(APE)

*Alexander Szendy*  
*Michal Klein*  
*2016/2017*

## Obsah

1	Úvod .....	5
1.1	Ciele Dokumentu .....	5
1.2	Rozsah projektu .....	5
1.3	Definície, skratky .....	5
1.4	Odkazy .....	5
1.5	Nasledujúce kapitoly .....	5
2	Všeobecný popis.....	6
2.1	Perspektíva projektu.....	6
2.2	Funkcie projektu .....	6
2.3	Charakteristika používateľov .....	6
2.4	Všeobecné obmedzenia.....	6
3	Špecifické požiadavky.....	7
3.1	Čistenie logu .....	7
3.2	Export logu.....	7
3.3	Práva používateľa .....	7
3.4	Anotovaný document .....	7
3.4.1	Načítanie dokumentu.....	7
3.4.2	Zmena obsahu anotovaného dokumentu .....	7
3.4.3	Zobrazovanie zmien v dokumente .....	7
3.4.4	Rozdelenie dokumentu .....	7
3.5	Anotácie.....	7
3.5.1	Pridávanie/mazanie/úprava anotácií.....	7
3.5.2	Prekryv anotácií.....	7
3.5.3	Referencie.....	7
3.5.4	Zmena referencií.....	8
3.5.5	Typy anotácií .....	8
3.5.6	Vyznačenie anotácií a referencií v texte .....	8
3.5.7	Zobrazovanie anotácií .....	8
3.6	Export .....	8
4	Diagramy.....	10
4.1	Entitno-relačný diagram .....	10
4.2	Use-case diagram .....	10
4.3	Stavový diagram .....	11
5	Používateľské rozhranie .....	12

5.1	Celé grafické rozhranie .....	12
6	Analýza technológií.....	14
6.1	Klient.....	14
6.1.1	Java a JavaFx.....	14
6.1.2	JavaFx CSS.....	14
6.1.3	RichTextFx .....	14
7	Dekompozícia .....	15
7.1	Diagram komponentov .....	15
8	Návrh .....	16
8.1	Úvod.....	16
8.2	Triedny diagram.....	16
8.3	Triedy .....	17
8.3.1	LogListView .....	17
8.3.2	SimpleLogRecord.....	17
8.3.3	SimpleLogger .....	17
8.3.4	TreeObject .....	17
8.3.5	DisplayedTreeObject.....	17
8.3.6	AnnotationObject .....	18
8.3.7	CategoryObject.....	18
8.3.8	CoreferenceObject.....	18
8.3.9	ColorObject .....	18
8.3.10	SerializableTreeObj .....	18
8.3.11	SerializableDisTreeObj .....	18
8.3.12	SerializableAnotObj .....	19
8.3.13	SerializableCatObj.....	19
8.3.14	SerializableRefObj.....	19
8.3.15	Contoller .....	19
8.4	Dátový model.....	21
8.4.1	Export Projektu.....	21
8.4.2	Export Logu.....	21
8.4.3	Uloženie projektu do súboru .....	21
9	Testovacie scenáre .....	22
9.1	Projekt .....	22
9.1.1	Načítanie dokumentu.....	22
9.1.2	Uloženie rozpracovaného projektu.....	22

9.1.3	Načítanie uloženého projektu .....	22
9.1.4	Export dokumentu .....	22
9.2	Anotácia .....	22
9.2.1	Zobrazovanie anotácií .....	22
9.2.2	Úprava kategórií .....	22
9.2.3	Úprava anotácie .....	23
9.3	Log .....	23
9.3.1	Zobrazenie popisu o vykonanej udalosti .....	23
10	Inštalačná príručka .....	24
10.1	Systémové požiadavky .....	24
10.2	Inštalácia .....	24
11	Používateľská príručka .....	25
11.1	Prvé spustenie Aplikácie .....	25
11.2	Načítanie textového súboru .....	25
11.3	Práca s textom (Anotovanie) .....	26
11.4	Export .....	29
11.5	Nastavenia textu .....	30
11.6	Log .....	30

# 1 Úvod

## 1.1 Ciele Dokumentu

Cieľom tohto dokumentu je čo najbližšie popísať požiadavky zadávateľa projektu bez bližšieho určenia technickej špecifikácie ako bude projekt APE implementovaný. Slúži ako zoznam funkcií, ktoré hotový produkt musí spĺňať a na základe ktorých bude aj vyhodnotený. Dokument je určený pre všetkých ľudí, ktorí prídu priamo alebo nepriamo do kontaktu s týmto systémom alebo jeho vývojom.

## 1.2 Rozsah projektu

Našou úlohou je vytvoriť offline verziu textového editoru pre jednoduchý text, v ktorom bude môcť používateľ anotovať jeden zvolený dokument.

## 1.3 Definície, skratky

- Anotácia – časť textu, ktorá bude vyznačená
- Typ/kategória anotácie – každá anotácia má určitý typ (mesto, osobu, štát, čas, ...)
- NE – named entity, pomenovaná entita, zvyčajne ide o konkrétnu osobu, čas, mesto, organizáciu, ...
- UI – umelá inteligencia

## 1.4 Odkazy

Nasledujú odkazy na podobný softvér:

- <https://gate.ac.uk/>
- <http://brat.nlplab.org/index.html>

Príklad textu, ktorý sa bude môcť anotovať:

- [http://sagadb.org/egils\\_saga.en](http://sagadb.org/egils_saga.en)

## 1.5 Nasledujúce kapitoly

V nasledujúcich kapitolách bližšie špecifikujeme funkcie nášho systému, ako export dokumentov.

## **2 Všeobecný popis**

### **2.1 Perspektíva projektu**

APE bude mať využitie najmä v oblastiach UI, kde je často nevyhnutné program najskôr niečo naučiť – zvyčajne od človeka, ktorý rieši ten istý problém (v našom prípade ide o tzv. named entity recognition – či dané slovo/slová sú NE a ak áno, treba správne určiť kategóriu).

Jedným z využití je napríklad také, kde anotáciu objemnejších textových dokumentov bude možné rozdeliť medzi viacero ľudí, čím sa zníži časová aj pracovná vyťaženosť človeka.

### **2.2 Funkcie projektu**

Aplikácia bude obsahovať jednu časť: klient. Primárna funkcia klienta (s grafickým rozhraním) bude samotná anotácia dokumentu a zobrazovanie zmien pri anotácii.

Podrobnejšie informácie sú uvedené v špecifických požiadavkách [3].

### **2.3 Charakteristika používateľov**

Používateľov nerozlišujeme, a teda máme iba jeden typ:

- normálny používateľ (ďalej len používateľ)

Používateľ používa klienta a jeho cieľom je anotovanie.

### **2.4 Všeobecné obmedzenia**

Anotovať bude možné len jednoduché textové dokumenty. Dokumenty budú môcť mať rozličné konce riadkov ako aj kódovania (minimálne akceptované budú UTF-8 a ASCII).

### **3 Špecifické požiadavky**

#### **3.1 Čistenie logu**

Log bude môcť používateľ aj zmazať. Log sa vždy viaže na aktívnu skupinu a pri jej zmene sa sám zmaže.

#### **3.2 Export logu**

Používateľ bude mať možnosť aktuálny log aj exportovať.

#### **3.3 Práva používateľa**

Používateľ má práva:

- anotovania – umožňuje pridávať anotácie
- reanotovania – umožňuje meniť/mazať anotácie
- editovania – umožňuje pridávať/meniť/mazať typy a podtypy anotácií

#### **3.4 Anotovaný document**

##### **3.4.1 Načítanie dokumentu**

Používateľ po spustení bude môcť otvoriť nejaký textový súbor, v akceptovanom formáte, ako aj zvoliť možnosť automatického načítania tohto súboru pri ďalšom spustení.

##### **3.4.2 Zmena obsahu anotovaného dokumentu**

Obsah dokumentu nemožno upravovať iným spôsobom ako anotovaním.

##### **3.4.3 Zobrazovanie zmien v dokumente**

Zmeny vykonané používateľom sa okamžite zobrazia v texte dokumentu.

##### **3.4.4 Rozdelenie dokumentu**

Dlhé texty, ktoré sa nezmestia na zobrazenie do okna, sa budú dať aj scrollovať.

#### **3.5 Anotácie**

##### **3.5.1 Pridávanie/mazanie/úprava anotácií**

Vyznačený text v dokumente môže používateľ anotovať stlačením kláves, kde si bude môcť vybrať napr.: typ anotácie (napr. “mesto”), popis, odkazy, .... Anotácie môžu byť zmenené alebo zmazané.

##### **3.5.2 Prekryv anotácií**

Ľubovoľná časť textu môže byť anotovaná aj viackrát.

##### **3.5.3 Referencie**

Používateľ bude môcť špecifikovať, či je daná anotácia referencia a na ktorú predošlú anotáciu sa vzťahuje.

### 3.5.4 Zmena referencií

Referenciám možno iba zmeniť objekt, na ktorý referujú (rodič), prípadne ich vymazať. Referencia dedí od rodiča všetky jeho vlastnosti, pri zmene rodiča je patričná zmena vykonaná na všetkých jeho referenciách.

### 3.5.5 Typy anotácií

Používateľ bude môcť vytvárať nové typy a podtypy anotácií, alebo ich iným spôsobom upravovať (farba, tag, názov,...) či mazať.

### 3.5.6 Vyznačenie anotácií a referencií v texte

Anotácie v textovom editore budú farebne vyznačené podľa daného typu/kategórie. Farbu rôznych kategórií si môže užívateľ zvoliť. Prekryv anotácií bude nejakým spôsobom rozlíšený (napr. konkrétnou farbou alebo zmiešaním farieb jednotlivých prekrývajúcich sa anotácií). Referencie budú mať vždy rovnakú farbu, ako anotácia, na ktorú odkazujú.

### 3.5.7 Zobrazovanie anotácií

Používateľ si bude môcť určiť, ktoré typy anotácií sa mu budú zobrazovať v texte (napr. sa rozhodne, že anotácie typu “osoba” nebudú momentálne vyznačené). Táto zmena je len lokálna – nezobrazuje sa ostatným používateľom v skupine.

## 3.6 Export

Používateľ bude môcť exportovať dokument svojej aktívnej skupiny. Každý riadok vyexportovaného dokumentu bude mať nasledovný formát:

- číslo riadku/poradie    slovo    X:Z,...    A-B:...    odkazy

Kde **X** bude buď písmeno ‘B’ (begin – označuje začiatok anotácie) alebo ‘I’ (pokračovanie anotácie) prípadne nič, **Z** bude zoznam tagov daného typu anotácie (napr. PER pre persons, tagy si môže meniť používateľ s editačnými právami, viaceré tagy budú oddelené čiarkou, referencie budú mať nemenný tag). **A:B** je interval  $[A, B)$ , ktorý určuje riadok slova alebo slov, na ktoré sa referencia vzťahuje, viacero intervalov je oddelených čiarkou. Pri anotáciách, ktoré nie sú referencie a pri neanotovanom texte je A aj B prázdne. Viaceré intervaly budú oddelené čiarkou. Nasledovať bude zoznam odkazov, pre danú anotáciu (pri viacсловných anotáciách je to vždy v riadku prvého anotovaného slova).

Interpunkcia je takisto na samostatnom riadku.

Samotná dvojbodka v 3. stĺpci označuje, že slovo nie je anotované a vo 4. stĺpci, že slovo nie je referencia (kvôli lepšiemu parsovaniu).

Nasledujú príklady riadkov exportovaného dokumentu:

- 2 Bratislava    B:CIT    :    en.wikipedia.org/wiki/Bratislava



Viacslovný anotovaný text:

- 3 Barack B:PER : [en.wikipedia.org/wiki/Barack\\_Obama](http://en.wikipedia.org/wiki/Barack_Obama)
- 5 Obama I :

Referencia:

- 7 He B:REF 1-2

Neanotované slovo:

- 11 foo : :

Viacrát anotované jedno slovo:

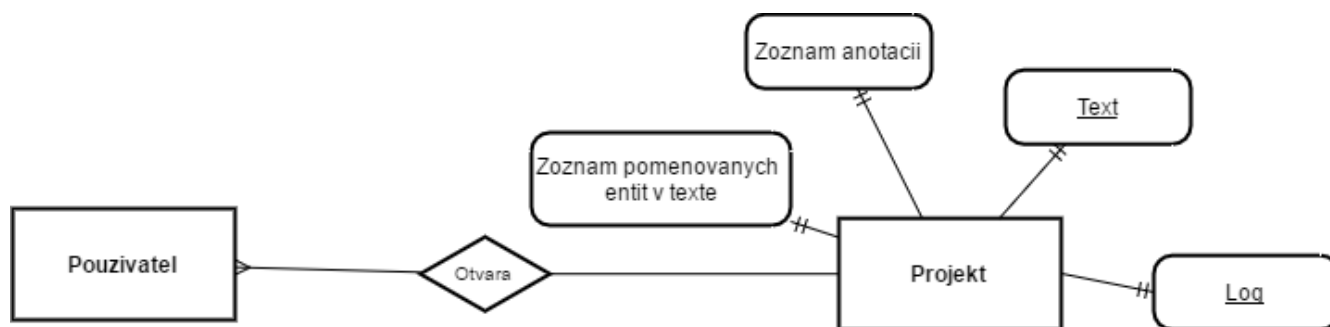
- 13 bar B:NAME1,NAME2 :

A nakoniec referencia odkazujúca na entitu, ktorá je na viacerých miestach:

- 14 It B:REF 123-124:2-3

## 4 Diagramy

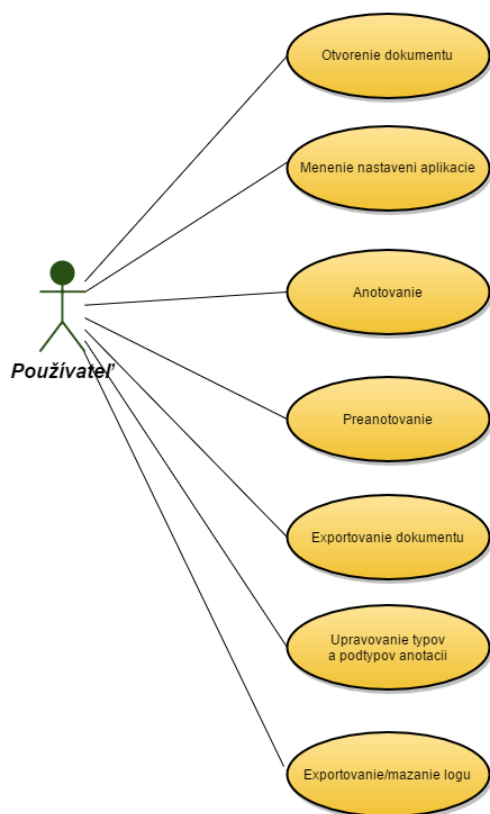
### 4.1 Entitno-relačný diagram



Obr. 4. 1 Entitno-relačný diagram

Entitno-relačný diagram (Obr. 4.1) zobrazuje entity nachádzajúce sa v projekte a relácie medzi nimi.

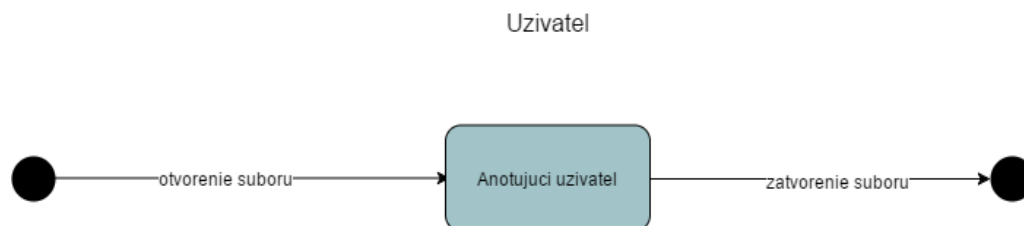
### 4.2 Use-case diagram



Obr. 4. 2 Use-Case diagram

Use-case diagram (Obr. 4.2) zobrazuje možnosti použitia projektu. Nachádza sa v ňom jedna entita používateľ'a, ktorá môže vykonávať rôzne funkcie.

### 4.3 Stavový diagram



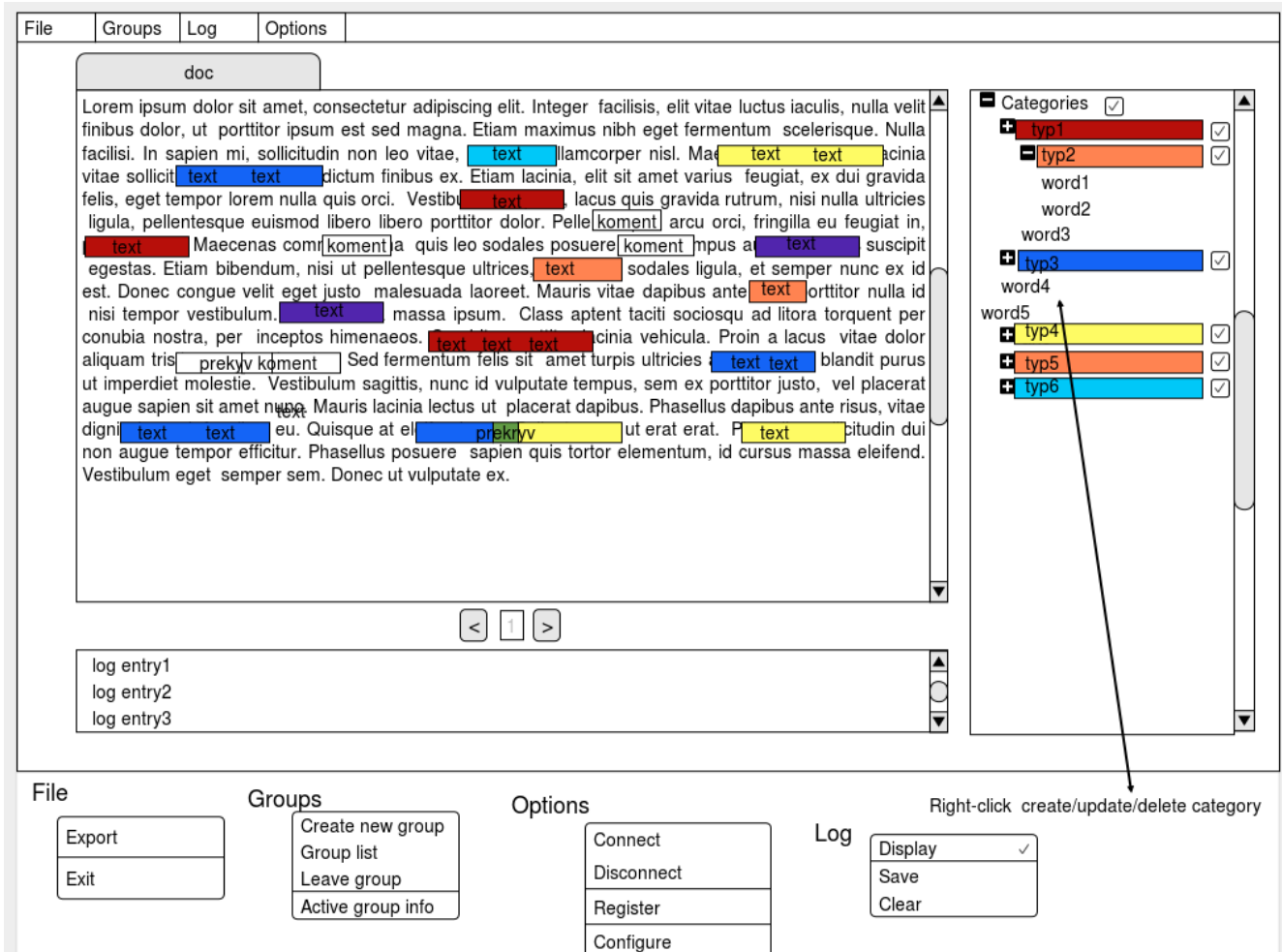
*Obr. 4. 3 Stavový diagram*

Stavový diagram (Obr. 4.3) zobrazuje stav, ktorý môže užívateľ nadobúdať po otvorení dokumentu. Počas tohto stavu užívateľ upravuje dokument anotovaním.

## 5 Používateľské rozhranie

V tejto časti je zobrazený približný náčrt designu a jeho funkčnosti pre výslednú aplikáciu.

### 5.1 Celé grafické rozhranie



Obr. 5. 1 Grafické rozhranie aplikácie

Obr. 5. 1 zobrazuje náčrt celého grafického rozhrania zobrazeného po spustení aplikácie. Rozhranie obsahuje komponenty : Menu (Vľavo hore), Dokument (Vľavo v strede), Log (Vľavo dole), Kategórie (Vpravo).

Menu obsahuje záložku

- File
  - o Export – umožňuje exportovať dokument do textového súboru, podľa pravidiel uvedených v sekcii 3.6 tohto dokumentu
- do výslednej podoby tejto záložky boli pridané možnosti :
  - o Vytvorenia projektu, ktorá vytvorí prázdny dokument s textom,
  - o Otvorenia projektu, ktorá načíta už uložený projekt z pamäte.
  - o Uloženia projektu, ktorá uloží aktuálny stav projektu

- Groups – túto položku výsledná aplikácia neobsahuje
- Options
  - o Configure – nastavenie fontu a zobrazenia potvrdzovacích dialógov.
  - ostatné nie sú implementované
- Log
  - o Display- možnosť výberu zobrazenia – v okne/ pod dokumentom
  - o Save- možnosť ukladať históriu
  - o Clear- zmazanie histórie udalostí

A dialog box titled 'New category' with the following fields: 'Name', 'Tag', 'Color', and 'Subtype of' (a dropdown menu). At the bottom are 'Ok' and 'Cancel' buttons.

*Obr. 5. 2 New category*

A dialog box titled 'New annotation' with the following fields: 'Category' (a dropdown menu), 'Links', and 'Description' (a text area). At the bottom are 'Ok' and 'Cancel' buttons.

*Obr. 5. 3 New annotation*

- Annotations (Pridaná záložka do výslednej podoby rozhrania)
  - o New category (Obr. 5.2) – vytvorenie novej kategórie
  - o New annotation (Obr. 5.3)– vytvorenie novej anotácie textu
  - o New coreference – vytvorenie referencie na anotáciu

## **6 Analýza technológií**

### **6.1 Klient**

#### **6.1.1 Java a JavaFx**

Klient bude naprogramovaný v jazyku Java s využitím platformy JavaFx, ktorá slúži na tvorbu grafického rozhrania nielen pre desktopové aplikácie.

#### **6.1.2 JavaFx CSS**

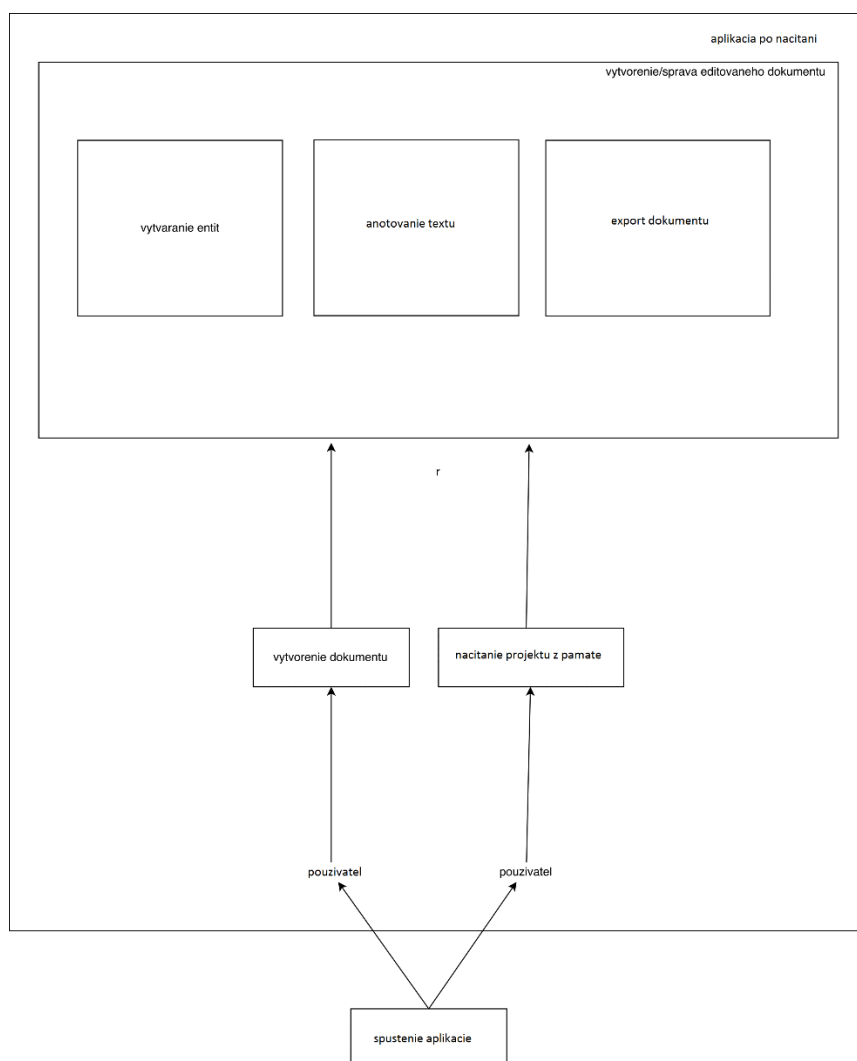
JavaFx CSS je založená na webovej verzii CSS. Používa sa na prispôsobenie vzhľadu aplikácií. V našom prípade je použitá na zvýraznenie anotácií ich zafarbením na príslušnú farbu.

#### **6.1.3 RichTextFx**

RichTextFx poskytuje API pre JavaFx a obsahuje rôzne užitočné funkcie, ako pridávanie štýlu k rôznym častiam textu, číslovanie riadkov, zobrazenie popup-ov, ... a bude tvoriť základ nášeho editora.

## 7 Dekompozícia

### 7.1 Diagram komponentov



Obr. 7. 1 Diagram komponentov

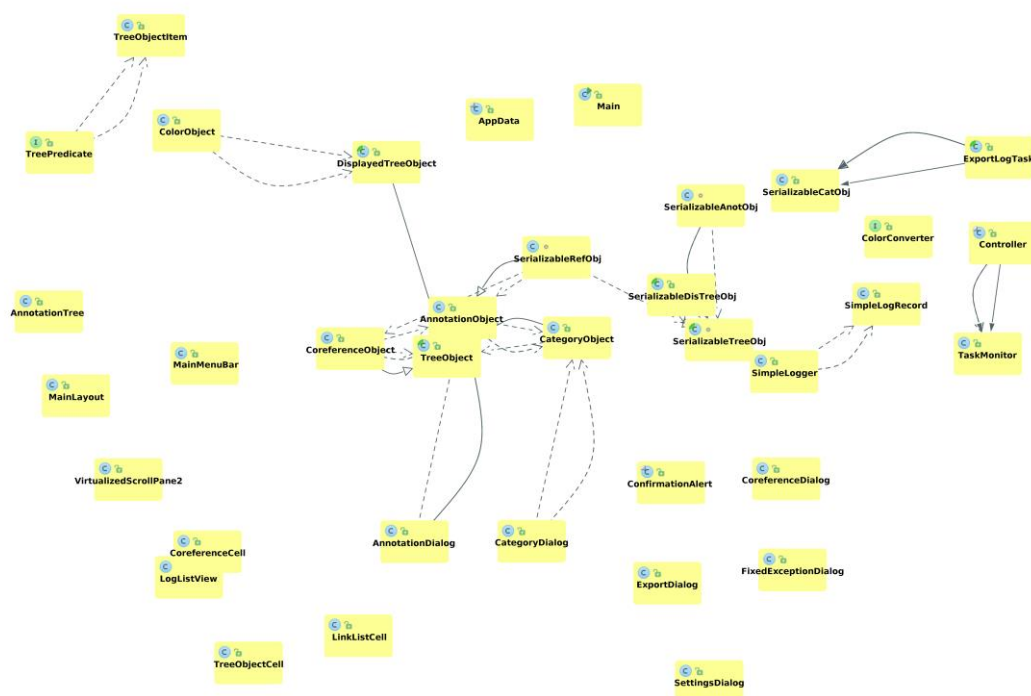
Obr. 7.1 obsahuje jeden komponent, ktorý slúži na prácu s dokumentom.

## 8 Návrh

### 8.1 Úvod

Systém je vyvíjaný v jazyku Java, ktorý používa grafické rozhranie JavaFx. JavaFx sa stará o všetky zobrazovania okien v tejto aplikácii. Ďalšiou využitou architektúrou je RichTextFx, ktorá upravuje zobrazenie textu dokumentu rôznymi štýlmi.

### 8.2 Triedny diagram



Obr. 8. 1 Class diagram

Tiedny diagram (Obr. 8.1) zobrazuje hlavné triedy, ktoré sú použité v našom systéme. Obsahuje napríklad triedy pre zobrazenie rôznych dialógových okien, Väčšina tried priamo súvisí so správnym fungovaním celého rozhrania- rôzne zobrazenia, rozloženia okien. Ďalšia časť tried pracuje s dátami, ako sú napríklad polia kategórií, anotácií, či samotný text dokumentu.



## 8.3 Triedy

### 8.3.1 LogListView

Slúži na zobrazenie Logu. asociovaná s triedou SimpleLogRecord.

SimpleDateFormat getFormatter() – vráti formát času

clearLog() – vymaže log

pause() – pozastaví animáciu logu

start() – pustí animáciu logu

stop() – zastaví animáciu logu

### 8.3.2 SimpleLogRecord

Trieda ukladajúca informácie logu.

Date getDate() – vráti čas vytvorenia logu

String getContext() – vráti typ vykonanej zmeny

String getMessage() – vráti popis zmeny

String toString() – vráti log reprezentovaný stringom

### 8.3.3 SimpleLogger

Trieda na prácu s logom. Asociovaná s SimpleLogRecord.

logNew(String message) – vytvorí SimpleLogRecord s contextom typu New

logUpdate(String message) – nový log s context Update

logDelete(String message) – nový log s contextom Delete

### 8.3.4 TreeObject

Abstraktná trieda obsahujúca strom Objektov.

long getId() – vráti id objektu

setId(int id) – nastaví id objektu

setName(string name) – nastaví meno objektu

String getTreeName() – vráti meno zoznamu, ktorom sa nachádza

String getName() – vráti meno objektu

changeParent(TreeObject<?> newParent) – zmení rodiča objektu

setParent(TreeObject<?> parent) – nastaví rodiča

TreeObject<?> getParent() – vráti rodiča

add(T item) – pridá dieťa

remove(TreeObject<?> treeObject) – odstráni zvolené dieťa

ObservableList<T> getChildren() – vráti zoznam detí

### 8.3.5 DisplayedTreeObject

Abstraktná trieda reprezentujúca označený text zdedená z TreeObject.

int getStart() – vráti začiatok objektu v texte

int getEnd() – vráti koniec objektu v texte

int getLineNum() – vráti číslo riadku

String getTreeName() – vráti objekt reprezentovaný stringom

### 8.3.6 AnnotationObject

Trieda anotácie zdedená od DisplayedTreeObject.

setUrls(List<String> urls) – nastaví URL adresy

List<String> getUrls() – vráti zoznam adries

String getDescription() – vráti popis anotácie

setDescription(String description) – nastaví popis anotácie

update(\$urls,\$description,TreeObject<?> newParent) – update Anotácie

### 8.3.7 CategoryObject

Zdedená trieda od DisplayedTreeObject.

update(String name, String tag, Color color, TreeObject<?> parent) – update premenných kategórie

Color getColor() – vráti farbu pre danú kategóriu

setColor(Color color) – nastaví farbu kategórie

String getTag() – vráti tag danej kategórie

setTag(String tag) – nastaví tag kategórie

### 8.3.8 CoreferenceObject

Zdedená trieda od DisplayedTreeObject.

update(TreeObject<?> parent) – nastaví anotáciu, na ktorú referuje

### 8.3.9 ColorObject

Trieda reprezentujúca zoznam zobrazených objektov. Asociovaná s DisplayedTreeObject.

ObservableList<DisplayedTreeObject<?>> getAllDisplayedObjects() – vráti zoznam všetkých zobrazených objektov

DisplayedTreeObject<?> getLastVisibleDisplayedObject – vráti posledný zobrazený objekt

DisplayedTreeObject<?>

removeDisplayedTreeObject(DisplayedTreeObject<?> displayedTreeObject) – odstráni \$displayedTreeObject zo zoznamu a vráti posledný objekt v zozname

### 8.3.10 SerializableTreeObj

Abstrakt, serializácia objektov

boolean isSelected() – vráti, či je Objekt zvolený.

long getId() – vráti Id objektu

### 8.3.11 SerializableDisTreeObj

Abstrakt, serializácia zobrazených objektov

List<Integer> getLineNumber() – vráti čísla riadkov

String toPosition() – vráti pozíciu formátu [cisloriadku, pozícia v riadku]

boolean isValid() – vráti, či je pozícia validná

boolean isStart(int i) – vráti, či objekt začína na pozícii \$i

boolean isEnd(int j) – vráti, či objekt končí na pozícii \$j  
SerializableTreeObj getParent() – vráti rodiča objektu  
setIsOkStart() – nastaví boolean startu objektu ako validný  
setIsOkEnd() – nastaví boolean konca objektu ako validný  
addLineNumber(int lineNumber) – pridá číslo riadku do zoznamu  
String toRecord(int start, Boolean outputId, Boolean outputDescription) – vráti reprezentáciu objektu pre export

### **8.3.12 SerializableAnotObj**

Zdedená od SerializableDisTreeObj, serializácia anotácie.

### **8.3.13 SerializableCatObj**

Zdedená od SerializableTreeObj, serializácia kategórie

### **8.3.14 SerializableRefObj**

Zdedená od SerializableDisTreeObj, serializácia referencie

### **8.3.15 Contoller**

Trieda slúžiaca na prepojenie celej aplikácie.

Init() – inicializuje Controller  
initTextArea() – inicializuje textový objekt, kde sa zobrazuje aktívny project  
setDefaultProperties() – nastaví všetky nastavenia na základné.  
Properties.getProperties() – vráti aktuálne nastavenia premenných  
double getWindowWidth() – vráti šírku okna  
double getWindowHeight() – vráti výšku okna  
String getFontFamily() – vráti štýl textu  
Int getFontSize() – vráti veľkosť textu  
Pattern getDelimetersPattern(Pattern pattern) – vráti oddeľovač  
setDelimetersPattern(Pattern pattern) – nastaví oddeľovač  
String getTreePos() – vráti pozíciu stromu  
setTreePos(String position) – nastaví pozíciu stromu  
String getLogPos() – vráti pozíciu logu  
setLogPosProperty(String position) – nastaví pozíciu logu  
String getStartFilePath() – vráti adresu súboru  
Boolean isPromptDel() - vráti, či je nastavené zobrazovania dotazovacích dialógov pre mazanie  
Boolean isPromptExit() - vráti, či je nastavené zobrazovania dotazovacieho dialógu pre ukončenie aplikácie  
updateStyleTextArea() – update štýlu textu dokumentu  
updateProperties(String fontFamily, Integer fontSize, Pattern pattern, String filename, String autoSelect, boolean isWrap, boolean isPromptExit, boolean isPromptDel) – update jednotlivých premenných

setShouldDisplayDialog(String type, Boolean value)- nastaví, či sa má dialóga daného typu zobrazovať  
deleteTreeObject(TreeObject<?>...treeObject) – vymaže objekty stromu  
newProject() – vytvorí nový project s načítaním súboru  
saveProject() – uloží projekt  
saveProjectAs() – uloží project do nového súboru  
openProject() – otvorí project zo súboru  
exportProjekt()- exportuje project  
exportLog() – exportuje log  
boolean canShutDown()- vráti, či používateľ zvolil kladnú možnosť v dialógu  
shutdown() – zatvorí aplikáciu

## 8.4 Dátový model

### 8.4.1 Export Projektů

Je vykonaný uložením do súboru v textovom formáte.  
Formát je bližšie špecifikovaný v časti **3.63.6Export**.

### 8.4.2 Export Logu

Je vykonaný zapísaním výpisu Logu do súboru v textovom formáte.  
Každý riadok v súbore predstavuje jeden riadok v Logu aplikácie.

Log je formátovaný nasledovne:

Čas[ ]TypVykonanejAkcije[ ]Objekt[ ]-[ ]PopisZmeny

Čas = Hodiny:Minúty:Sekundy

[ ] = medzera

TypVykonanejAkcije = typ akcie – New, Update, Delete

Objekt = objekt, na ktorom bola úprava urobená – kategória, anotácia, referencia...

PopisZmeny = rozpísaný popis zmeny

### 8.4.3 Uloženie projektu do súboru

Projekt sa ukladá do súboru pomocou Java Serializable.

Do súboru sú uložené všetky aktuálne premenné a celý stav aplikácie  
v momente uloženia.

## **9 Testovacie scenáre**

### **9.1 Projekt**

#### **9.1.1 Načítanie dokumentu**

- 1 Vstup:** výber textového dokumentu
- 2 Výstup:** zobrazenie textu dokumentu v okne
- 3 Stav:** otestované

#### **9.1.2 Uloženie rozpracovaného projektu**

- 4 Vstup:** výber adresára na uloženie súboru
- 5 Výstup:** súbor úspešne uložený
- 6 Stav:** otestované

#### **9.1.3 Načítanie uloženého projektu**

- 7 Vstup:** výber adresára na uloženie súboru
- 8 Výstup:** text s anotáciami a všetkými predošlými nastaveniami sa zobrazí v aplikácii
- 9 Stav:** otestované

#### **9.1.4 Export dokumentu**

- 10 Vstup:** výber adresára na uloženie súboru
- 11 Výstup:** súbor úspešne uložený v správnom formáte
- 12 Stav:** otestované

### **9.2 Anotácia**

#### **9.2.1 Zobrazovanie anotácií**

- 13 Vstup:** odznačená voľba na zobrazenie anotácie v texte
- 14 Výstup:** v texte nebudú farebne vyznačené anotácie danej kategórie
- 15 Stav:** otestované

- 16 Vstup:** označená voľba na zobrazenie anotácie v texte

- 17 Výstup:** farebné vyznačenie sa znovu zobrazí

- 18 Stav:** otestované

#### **9.2.2 Úprava kategórií**

- 19 Vstup:** správne vyplnený formulár pre vytvorenie novej/úpravu kategórie
- 20 Výstup:** kategória úspešne pridaná do stromu s kategóriami, alebo upravená
- 21 Stav:** otestované

### **9.2.3 Úprava anotácie**

**22 Vstup:** označený text + vyplnený formulár pre pridanie/zmazanie/úpravu anotácie

**23 Výstup:** zobrazenie zmien v texte dokumentu a pridanie/ zobrazenie zmeny v strome kategórií

**24 Stav:** otestované

### **9.3 Log**

#### **9.3.1 Zobrazenie popisu o vykonanej udalosti**

**25 Vstup:** Ľubovoľná zmena v projekte

**26 Výstup:** zobrazenie popisu zmeny s aktuálnym časom

**27 Stav:** otestované pre rôzne vykonané zmeny

## **10 Inštalačná príručka**

### **10.1 Systémové požiadavky**

Java Runtime Environment  $\geq$  8, s JavaFx

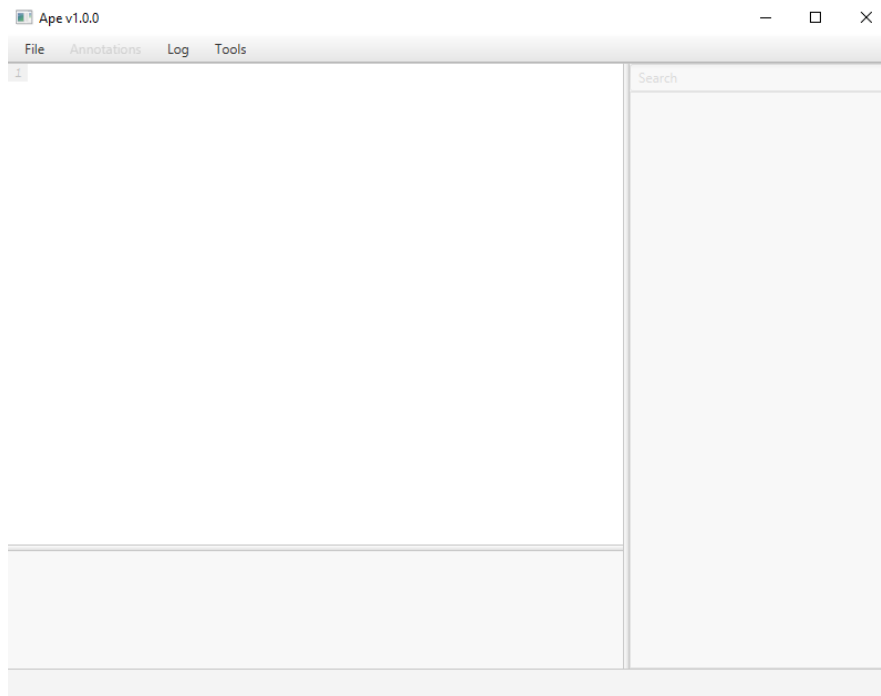
### **10.2 Inštalácia**

Stačí rozzipovať súbor ape.zip. Aplikácia sa spúšťa súborom ape.jar. Pre správne fungovanie by mal byť priečinok libs spolu s aplikáciou .jar v jednom priečinku.



## 11 Používateľská príručka

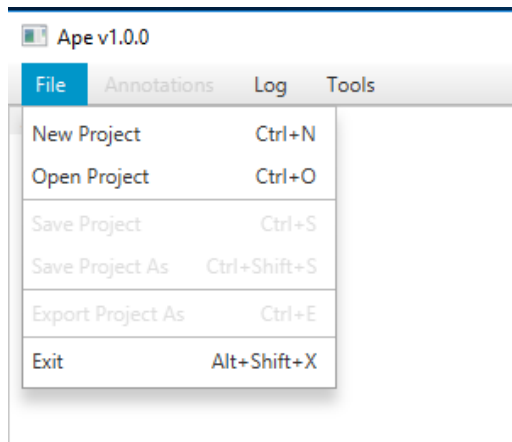
### 11.1 Prvé spustenie Aplikácie



Obr. 11. 1 Prázdné okno

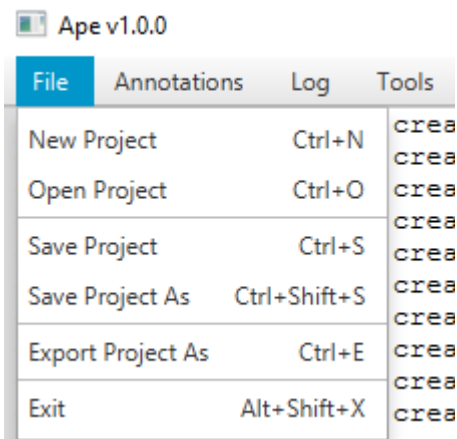
Obr. 11.1 zobrazuje ako vyzerá aplikácia po spustení, väčšina funkcií je zablokovaná. Pre viac funkcií je potrebné načítať súbor pomocou položky v menu: File -> New project (zobrazené na Obr. 11.2 nižšie).

### 11.2 Načítanie textového súboru



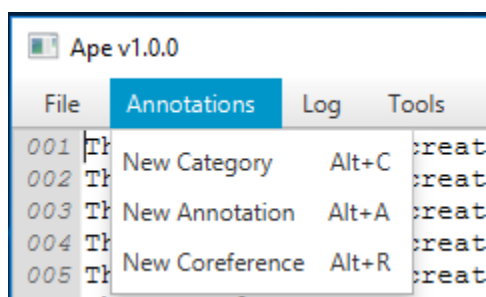
Obr. 11. 2 Kontextové menu File

Po kliknutí na New Project sa zobrazí nové okno, kde možno vybrať nejaký textový súbor (.txt) pre ďalšie spracovanie. Po potvrdení výberu sa text obsiahnutý vo vybratom súbore zobrazí na ploche nižšie a zároveň sa odomknú všetky funkcie, ktoré boli zablokované (viz Obr. 11.3).



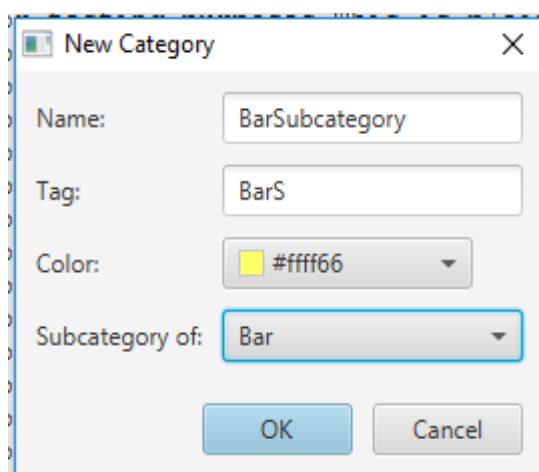
Obr. 11. 3 Odomknuté funkcie menu

### 11.3 Práca s textom (Anotovanie)



Obr. 11. 4 Anotácie

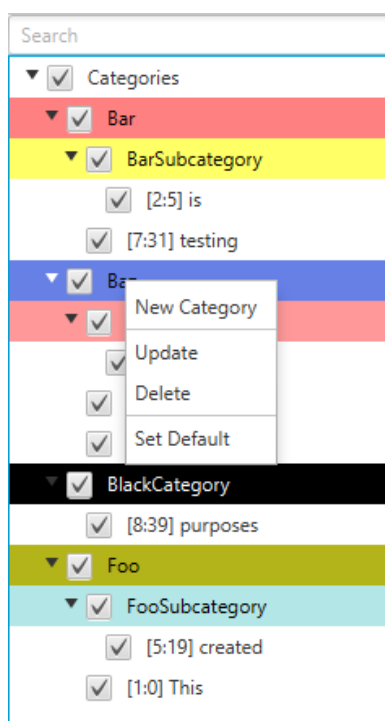
Obr. 11.4 zobrazuje obsah kontextového menu po kliknutí na voľbu Annotations.



Obr. 11. 5 Nová kategória

Pre neskoršie anotovanie je nutné vytvoriť rôzne kategórie, ktoré budú reprezentovať typ slova/ viacerých slov v texte. Tag slúži na reprezentovanie

danej anotácie pri exportovaní projektu do textového súboru, teda by mal byť tiež unikátny. Vybratie farby je tiež dôležité, pre rozlíšenie medzi ostatnými anotáciami priamo v texte. Je možné tiež vytvárať pod-kategórie už vytvorených kategórií. Všetky tieto veci sa dajú nastaviť pri kliknutí v menu na Annotations-> New Category, kde sa zobrazí nové okno. (Obr. 11.4 a Obr. 11.5)



Obr. 11. 6 Kontextové menu po kliknutí na jednu z kategórií

Zmena kategórie je možná po kliknutí pravým tlačítkom myši na danú kategóriu a zvolením možnosti Update.

(Obr. 11.6) V tomto kontextovom menu je tiež možnosť vytvoriť novú kategóriu, zmazať vybranú, alebo nastaviť vybranú kategóriu ako základnú, ktorá sa defaultne zobrazí pri anotácii.

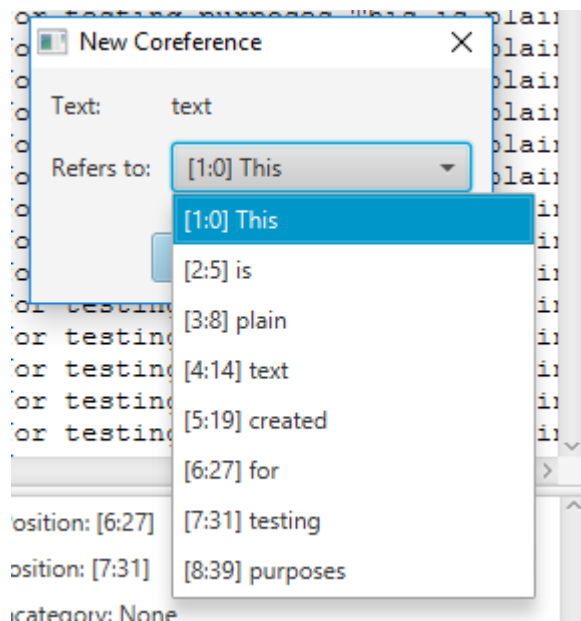
Obr. 11. 7 Nová anotácia

Vytvorenie anotácie je možné po vybratí textu a buď vybratí v menu (Obr. 11.4) možnosť New Annotation, alebo použitím klávesovej skratky Alt+A. Zobrazí sa formulár na vyplnenie (Obr. 11.7). Tu je možnosť vytvoriť novú kategóriu, alebo vybrať z existujúcich. Tiež sa dá vložiť URL, ktorá nejak súvisí s vybratým textom, a dá sa vložiť aj popis k danej anotácii.

Obr. 11. 8 Update anotácie

Možnosť upraviť už vytvorenú anotáciu môžete zobraziť po kliknutí pravým tlačítkom myši na anotáciu, ktorú chcete modifikovať (prípadne zmazať). (Obr. 11.8)

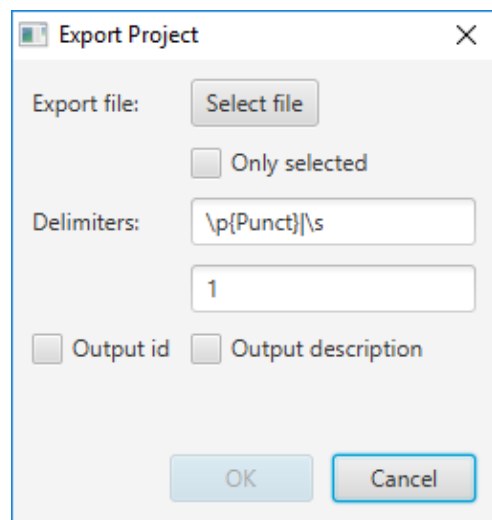
Rovnako ako pri vytváraní anotácie, je možné zadať všetky údaje znovu.



Obr. 11. 9 Referencia

Pridanie referencie sa vykonáva výberom textu a následným zvolením možnosti v menu Annotations->Coreference (Obr. 11.4), prípadne stlačením kláves Alt+R. Zobrazí sa nové okno, kde je možné zvoliť, na ktorú anotáciu daný text referuje.

## 11.4 Export

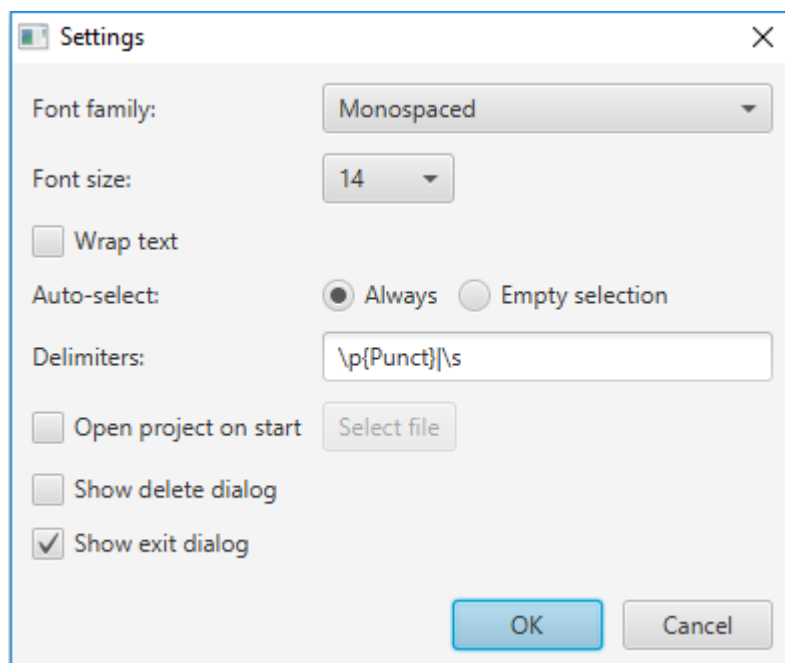


Obr. 11. 10 Export

Export dokumentu sa vykoná výberom v menu File->Export Project as (Obr. 11.3).

Zobrazí sa nové okno kde sa dá nastaviť formátovanie, ako aj výber miesta na disku, kam sa exportovaný dokument uloží.

## 11.5 Nastavenia textu



Obr. 11. 11 Nastavenia

Nastaviť rôzne formátovanie zobrazenia textu je možné v menu Tools->Options. Tiež je možné vybrať, ktoré potvrdzovacie dialogy sa budú zobrazovať.(Obr. 11.11)

## 11.6 Log

Po každej akcii sa do poľa vľavo dole pridá výpis druhu akcie s aktuálnym časom. Vymazanie, prípadne export tohto textu je možné vykonať zvolením menu položky Log.