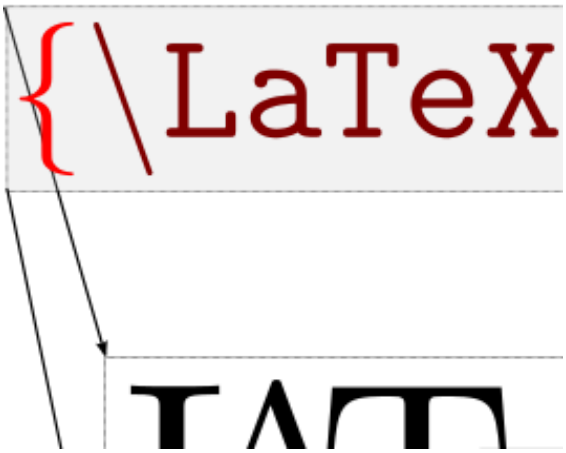
LatexEditor

Requirements Definition



VERSIONS HISTORY

Date	Version	Description	Author
3/19/2019	<1.0>	1st version of the requirements definition document	A. Zarras

TABLE OF CONTENTS

1 Introduction

This document defines the basic requirements for LatexEditor, an application that facilitates the writing and maintenance of Latex documents.

1.1 Purpose

Latex is a well known high quality document preparation markup language [1,2,3,4]. It provides a large variety of styles and commands that enable advanced document formatting. Typically, a Latex document is compiled with a tool like MikTex, Lyx, etc. to produce a respective formatted document in pdf, ps, etc. Formatting documents with Latex is like a programming process as it involves the proper usage of Latex commands which are embedded in the document contents. The objective of this project is to develop a simple Latex editor for inexperienced Latex users. The goal of the editor is to facilitate the usage of Latex commands for the preparation of Latex documents. One of the prominent features that distinguishes the LatexEditor from other similar applications is its multi-strategy version tracking functionalities that enable undo and redo actions.

1.2 Document Structure

The rest of this document is structured as follows. In Section 2 we focus on basic domain properties related to the project. In Sections 3 and 4 we provide, respectively, the "raw" functional and non-functional requirements that should be further analyzed to drive the design, implementation and testing of LatexEditor.

2 Domain Properties

The LatexEditor enables the creation and management of subsequent versions of a Latex document. A particular version of the Latex document is characterized by the following basic properties:

- The author.
- The date that was last modified.
- The copyright.
- A version ID.
- The contents of the document.

Typically the contents of the Latex document contain Latex commands that specify the title (\title{...}) of the document, the author (\author{...}), the (\date{}), and allow to structure the document properly (e.g., \chapter{..}, \section{...}, \subsection{...}, etc.).

LatexEditor allows the users to create different types of Latex documents, starting from corresponding **templates** that are extended and customized by the users during the document editing process. Each template specifies the basic document attributes and a skeleton document structure that serves as a starting point for the document editing process. In particular, we consider the following templates.

2.1 Report Template

```
\documentclass[11pt,a4paper]{report}

\usepackage{graphicx}

\begin{document}

\title{Report Template: How to Structure a LaTeX Document}
\author{Author1 \and Author2 \and ...}
\date{\today}
\maketitle

\begin{abstract}
Your abstract goes here...
...
\end{abstract}

\chapter{First Chapter}

\section{Section Title 1}
\section{Section Title 2}
\section{Section Title.....}

\chapter{....}

\chapter{Conclusion}


\chapter*{References}

\end{document}
```

2.2 Book Template

```
\documentclass[11pt,a4paper]{book}
\usepackage{graphicx}

\begin{document}

\title{Book: How to Structure a LaTeX Document}
\author{Author1 \and Author2 \and ...}
\date{\today}

\maketitle

\frontmatter

\chapter{Preface}

\mainmatter

\chapter{First chapter}

\section{Section Title 1}
\section{Section Title 2}
\section{Section Title.....}

\chapter{....}

\chapter{Conclusion}

\chapter*{References}

\backmatter

\chapter{Last note}

\end{document}
```

2.3 Article Template

```
\documentclass[11pt,twocolumn,a4paper]{article}
\usepackage{graphicx}

\begin{document}

\title{Article: How to Structure a LaTeX Document}
\author{Author1 \and Author2 \and ...}
\date{\today}
```

```
\maketitle

\section{Section Title 1}

\section{Section Title 2}

\section{Section Title.....}

\section{Conclusion}

\section*{References}

\end{document}
```

2.4 Letter Template

```
\documentclass{letter}
\usepackage{hyperref}
\signature{Sender's Name}
\address{Sender's address...}
\begin{document}

\begin{letter}{Destination address....}
\opening{Dear Sir or Madam:}

I am writing to you .......


\closing{Yours Faithfully,}

\ps

P.S. text .....

\encl{Copyright permission form}

\end{letter}
\end{document}
```

3 Functional Requirements / User Stories

- [US-1] As a user, I want to create a new Latex document, based on a particular Latex document template. If I don't specify a template, the application should create an empty Latex document.
- [US-2] As a user, I want to edit the contents of the Latex document, via the application's user interface.

- [US-3] As a user, I want to add Latex commands in the Latex document automatically using the application user interface. A minimal set of Latex commands that I want to use is given in Table 1. Some of the commands are allowed only for specific types of Latex documents. The application should notify me if I try to add Latex commands that are not allowed in the Latex document.

- [US-4] As a user, I should be able to activate an automatic version tracking mechanism that keeps track of the document evolution history, at any time. The history consists of a sequence of subsequent versions of the Latex document. The mechanism should provide at least two alternative storage strategies for the document evolution history:

    - **Volatile (default strategy)**: for each document change the mechanism keeps the previous version of the document in a main memory list of subsequent document versions.

    - **Stable**: for each document change the mechanism keeps the previous version of the document on disk storage.

- [US-5] As a user, I should be able to change the storage strategy of the version tracking mechanism at any time.

    - If I change from Volatile to Stable the application should store the document evolution history on disk storage.

    - If I change from Stable to Volatile the application should load the document evolution history from disk storage to main memory.

- [US-6] As a user, I should be able to disable the version tracking mechanism at any time.

- [US-7] As a user, I should be able to rollback to a previous version of the Latex document based on the document evolution history. The application should notify me if I try to use the rollback action and the version tracking mechanism has not been enabled before.

- [US-8] As a user, I should be able to save the Latex document on disk storage.

- [US-9] As a user, I should be able to load the Latex document from disk storage.

Latex command	Description	Allowed document types
\chapter{...}	Add chapter	ALL except for letter and article
\section{}	Add section	ALL except for letter
\subsection{}	Add subsection	

\subsubsection{}	Add subsubsection	
\begin{itemize} **\item ...** **\item ...** **\end{itemize}**	Add enumeration list	
\begin{enumerate} **\item ...** **\item ...** **\end{enumerate}**	Add enumeration list	
\begin{table} **\caption{....}\label{...}** **\begin{tabular}{|c|c|c|}** **\hline** **... &...&...** **... &...&...** **... &...&...** **\hline** **\end{tabular}** **\end{table}**	Add a table	
\begin{figure} **\includegraphics[width=...,height=...]{...}** **\caption{....}\label{...}** **\end{figure}**		

4 Non Functional Requirements

[NF1] **Extensibility**: In software engineering, extensibility is a design principle where the implementation takes future growth into consideration. In the case of this project, extensibility concerns the supported Latex templates, commands, version tracking strategies. Specifically, the application should be designed such as the addition of new templates and strategies is as simple as possible and according to the well known open-closed principle. To this end, several GoF design patterns [3] can be considered (Factory, Command, Prototype, Strategy, etc.)

[NF2] **Usability**: In software engineering usability concerns the ease of use and learnability. In the context of this project the application should enable the patterns specification via a simple and user-intuitive interface. The application should also provide help, in the form of user guidelines, concerning its main functionalities, and the contents of the different pattern templates.

5 References

[1] General information about the Latex project https://www.latex-project.org/

[2] Download MikTex - Latex distribution - https://miktex.org/

[3] WinEdt - manipulation and compilation of Latex documents http://www.winedt.com/index.html

[4] LaTeX/Document Structure https://en.wikibooks.org/wiki/LaTeX/Document_Structure