**BibTeX Management System**

**Software Design Specification**

**Author: Paulina Mibenge**

Table of Contents

[1. Introduction 3](#_Toc438409373)

[1.1. Purpose 3](#_Toc438409374)

[1.2. Scope 3](#_Toc438409375)

[1.3. Intended audience 3](#_Toc438409376)

[1.4. Glossary 3](#_Toc438409377)

[1.5. Document Overview 4](#_Toc438409378)

[2. Architectural Design 4](#_Toc438409379)

[2.1. High level component and their interaction 4](#_Toc438409380)

[2.1.1. Existing Components 4](#_Toc438409381)

[2.1.2. Components to be implemented 5](#_Toc438409382)

[2.1.3. Existing Interfaces 5](#_Toc438409383)

[2.1.4. Interfaces to be added to the system 6](#_Toc438409384)

[2.2. Design decision applied to whole application 6](#_Toc438409385)

[2.2.1. Object-oriented software development method 6](#_Toc438409386)

[3. Component and Detail Design 7](#_Toc438409387)

[3.1. Class diagram 7](#_Toc438409388)

[3.2. User interface design 7](#_Toc438409389)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 26-10-2015 | 0.1 | First draft | Paulina Mibenge |
| 08-11-2015 | 0.2 | Added description of architectural design and class diagram | Paulina Mibenge |
| 15-12-2015 | 0.3 | Updated architectural design | Paulina Mibenge |

1. Introduction
   1. Purpose

The purpose of this document is to outline software architecture and design of BibTeX Management System. It is targeted at developers to help them meet the client’s requirements in an efficient and effective way.

* 1. Scope

In this document, it will be specified how the requirements—both functional and non-functional—will be met. Needs of client are listed in the Analysis and Requirements Document. This specification will introduce developers to high level components and interfaces of the system, as well as algorithms.

* 1. Intended audience

This document is prepared mainly for the developers of the BibTex Management System.

* 1. Glossary

|  |  |
| --- | --- |
| **Term** | **Description** |
| Bibliography entry | Bibliography item containing specified parameters. Each entry has a type and a style. It also has a unique identification number. |
| File | File with extension ‘.tex’ or ‘.bib’. Those files are handled by the BibTeX Management System. |
| Style | Abbreviation style(trend) defining how an entry should look like. Each bibliography item can have different style and also appear as multiple variations considering its trend. |

* 1. Document Overview

The next chapter describes architectural design of the BibTex Management System. It lists the high-level components of the system and specifies their main tasks. It also describes the interfaces of these components, as well as the design decisions applied in the development process of the application.

1. Architectural Design
   1. High level component and their interaction
      1. Existing Components

* BibTeX Interpreter

Component responsible for interpreting the content of files with *bib* extension.

* LaTeX Interpreter

Component responsible for interpreting the content of files with *tex* extension.

* File Manager

Component which opens and reads all the content of selected file. It also has the ability to save the file.

* GUI

The key component in user interaction process. This is the user interface—it manages windows, their subcomponents, and results of actions performed on them.

* BibTex entry content

This component stores all the data of one bibliography entry. It has information about styles of each parameter as well as whether it is enabled in the compilation of bibliography.

* + 1. Components to be implemented
* Style Interpreter

This component interprets a style file.

* + 1. Existing Interfaces
* BibTeX Interpreter

|  |  |
| --- | --- |
| *getAllBibtexRecords* | The interpreter returns all the records that are present in the bibliography file. |

* LaTeX Interpreter

|  |  |
| --- | --- |
| *getAllBibtexReferences* | The interpreter returns all the references of bibliography entries in the analyzed text—content of open tex file. |

* File Manager

|  |  |
| --- | --- |
| *openTexFile* | The manager opens the file which is pointed by the path a user has specified. This method also reads all the content of the file. |
| *getContentOfTexFile* | The manager returns the content of the opened file. |
| *saveBibFile* | The manager saves specified data to a file with a specified path. |

* BibTeX entry content

|  |  |
| --- | --- |
| *addEntryContent* | Adds records to the entry. |
| *getEnabled* | It returns a collection of enable parameter for records. |
| *setEnabled* | It sets enable value for a specified record of the entry. |
| *getStyle* | It returns a collection of styles for records. |
| *setStyle* | It sets a specified style to a record. |

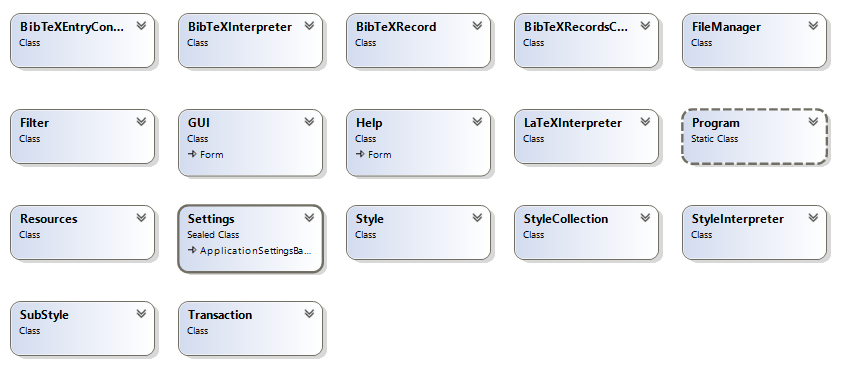
* + 1. Interfaces to be added to the system
* Style Interpreter

|  |  |
| --- | --- |
| *loadStyleCollection* | Reads the style file and collects all available styles. |
| *getStyledText* | Returns the text in a specified style. |

* 1. Design decision applied to whole application
     1. Object-oriented software development method

This method has been chosen to deliver a system which can be relatively fast further developed and easily maintained in case of requirement changes and add-ons.

1. Component and Detail Design
   1. Class diagram



* 1. User interface design

The user interface design is explicitly described in the GUI design document.