

## I. INTRODUCTION

### A. Background of the Study

TeX is both a program and a language. It is a program that uses the TeX typesetting language to create finished documents. [1] It was designed with two main goals in mind: first, to allow anybody to produce high-quality books using a reasonable amount of effort, and, second, to provide a system that would give the exact same results on all computers, now and in the future. [2]

LaTeX is a document markup language and document preparation system for the TeX typesetting program. LaTeX is intended to provide a high-level language that accesses the power of TeX. LaTeX essentially comprises a collection of TeX macros and a program to process LaTeX documents. Because the TeX formatting commands are very low-level, it is usually much simpler for end-users to use LaTeX. [3]

To publish something, authors give their typed manuscript to a publishing company. One of their book designers then decides the layout of the document (column width, fonts, space before and after headings, ...). The book designer writes his instructions into the manuscript and then gives it to a typesetter, who typesets the book according to these instructions. A human book designer tries to find out what the author had in mind while writing the manuscript. He decides on chapter headings, citations, examples, formulae, etc. based on his professional knowledge and from the contents of the manuscript. [4]

In a LaTeX environment, LaTeX takes the role of the book designer and uses TeX as its typesetter. But LaTeX is “only” a program and therefore needs more guidance. The author has to provide additional information to describe the logical structure of his work. This information is written into the text as “LaTeX commands.” This is quite different from the WYSIWYG approach that most modern word processors, such as MS Word or Corel WordPerfect, take. With these applications,

authors specify the document layout interactively while typing text into the computer. They can see on the screen how the final work will look when it is printed. [4]

LaTeX is based on the idea that authors should be able to focus on the meaning of what they are writing without being distracted by the visual presentation of the information. In preparing a LaTeX document, the author specifies the logical structure using familiar concepts such as chapter, section, table, figure, etc., and lets the LaTeX system worry about the presentation of these structures. It therefore encourages the separation of layout from content while still allowing manual typesetting adjustments where needed. This is similar to the mechanism by which many word processors allow styles to be defined globally for an entire document or the CSS mechanism used by HTML. [3]

A thesis is a document submitted in support of candidature for an academic degree or professional qualification presenting the author's research and findings. In some countries/universities, it is used as part of a bachelor's or master's course. [5]

The structure of the thesis or dissertation explains the purpose, the methods used and the findings of the project by including a series of chapters. A typical thesis has a title page, an abstract, a table of contents, a body, comprising the various chapters, and a bibliography or (more usually) a references section. They vary in their structure in line with the many different areas of study (arts, humanities, social sciences, technology, sciences, etc.) and the great differences between them. [5]

Degree-awarding institutions often define their own house style that candidates have to follow when preparing a thesis document. [5] As such, LaTeX would be very useful in enforcing uniformity given an institution's standards for thesis documents.

## **B. Statement of the Problem**

Since universities impose standards on the style and structure of thesis documents being submitted, observing proper format consumes time and distracts one from what he should be focusing on – the contents of the thesis document.

Students consulting regarding proper formatting and revising the document to comply with this, and defense panel members having to go over a document's format, are some of the overheads to be expected when preparing a thesis document. And given that these formats are currently applied manually, differing formats are also observed.

Using LaTeX aids one in creating well-structured documents, complying with set standards. But since it involves codes, using LaTeX alone is also not that simple. Similar to programming languages, TeX is written using commands which one will have to learn in order to create documents. There are also rules governing the use of these commands and they should be followed carefully to produce the desired output.

Since LaTeX has been around for quite some time now, a number of editors, such as TeXnicCenter, WinEdt, and TeXMaker, with enhanced TeX support can be found online which eases the process of creating documents. That means that TeX codes can be written manually, but these editors offer many possibilities to insert frequently-used code snippets (standard formatting commands, super- and subscripts and other constructs like formulas, tables, etc.) via buttons or menus. You can also run TeX and view the results with a couple of simple clicks. [1]

But most open source editors were designed to run on UNIX platforms. Editors meant to run on Windows are mostly shareware with the exception of some, such as TeXMaker and TeXnicCenter beta.

Although these editors provide buttons and menus to facilitate programming, people who are new to LaTeX would find it difficult to use the program. A lot of commands still need to be written manually, hence, you will have to learn the rules and commands first before you can produce a working document.

Compiling your code poses another problem. LaTeX compilers offer error detection but it is up to the user to determine the cause of the error by reading the

manuals provided by LaTeX. It is often difficult to trace the cause of the error and this can be time consuming.

Mainly, the disadvantage of using the existing LaTeX system is based on the difficulty of learning and remembering the LaTeX commands and the possibility of having syntax errors, and dealing with them.

### **C. Objectives**

To develop a stand-alone application to aid in standardizing the style and structure of UPM thesis documents. This application shall take advantage of LaTeX's features and provide the following functionalities:

- Provide standard formatting for the thesis document, relieving users of the need to observe and apply proper formatting manually.
- Provide a GUI interface on which users can work on so he'll be able to produce a working thesis document without having to learn LaTeX commands. The interface will provide a template where user can enter contents of the thesis document.
- The system should be able to process data entered by users through the template to produce LaTeX scripts which can be compiled to output a LaTeX-formatted thesis document.
- System should allow production of well-formatted thesis documents even with users being completely unknowledgeable on the formatting standards and on LaTeX programming. Hence, the system must be able to produce error-free scripts.

### **D. Significance of the Study**

Programming with LaTeX aids one in producing highly presentable and well-structured documents. It includes commands for properly structuring a document, displaying mathematical equations and figures, among others – possibly everything an author would need in creating any type of document.

Employing LaTeX's features and incorporating the university's standard, the UPM GUI LaTeX Thesis Editor will be a very effective tool in standardizing the structure and format of thesis documents. It would lessen the turn-around time and efforts related to formatting and verification thereof. By providing a GUI Editor, user can take advantage of these features without having to learn LaTeX programming and the standards set for formatting thesis documents.

### **E. Scope and Limitation**

1. The editor to be developed will not necessarily support the features of a typical editor described in Section C of the Theoretical Framework.
2. Packages to be included in the editor would only be those that are usually used by students. These packages should at least support the following:
  - Creation of the Table of Contents, Chapters, Bibliography, etc
  - Addition of figures, mathematical equations, formula, citations, etc
  - Apply proper pagination
  - Apply text styles and formats
3. Acceptable images are only those in PNG and JPG format.
4. The existing MikTeX compiler would be used in compiling the generated code.
5. The system will integrate with the existing TeXaide equation editor to aid in the creation of math equations.
6. Documents will be produced in PDF format.
7. Package upgrading will not be supported as they will not be added in the GUI editor.
8. Copying text from Word documents is not supported by the text pane used in the templates. As a work-around, user can copy contents first to Notepad, then recopy it to the templates.

### **F. Assumptions**

1. Version of compiler must be similar to the one used in developing the system so as to avoid unexpected results and errors.

2. All external programs needed by the system to work fully will be provided. These must be properly installed according to instructions given.
3. Images added to the system are not stored internally. This means that the file path user specified when adding the image will be used to retrieve and display the image. Removing the actual image from the file specified, the system will not be able to display the image properly. If the image is replaced while retaining its filename, the new image will be displayed by the system instead.

## II. REVIEW OF RELATED LITERATURE

The following journals were all taken from The PracTeX Journal which is dedicated to the exchange of practical experience and instruction of the use of TeX and other TeX-related or TeX-derived systems. [6]

*In My Opinion: LaTeX isn't for everyone but it could be for you* gives a background on what LaTeX is and the advantage and disadvantages of using it. While the merits of LaTeX the author points out are familiar to most PracTeX Journal readers, the follow-on comments from readers pointed out some of LaTeX's (and TeX's) weaknesses. [7]

*The Beginner's Forest of LaTeX* discussed how using LaTeX makes projects both beautiful and consistent, but how it can also be very time consuming. In particular, trying to input math equations by using Greek fonts, making tables, using some packages, and trying to solve issues by reading incomprehensible package documentation (some of which didn't even explain how to use the package, only how it was coded) were very time-consuming indeed. [8]

*Will TEX ever be WYSIWYG or the PDF synchronization story* discusses why editing plain, LaTeX or ConTeXt documents can be very difficult. It was said that dedicated text editors and environments are very handy, but we are far from the efficient graphical user interface of a modern word processor. In this article, some of the reasons why TeX has no WYSIWYG user interface has been pointed out and possible remedies have been discussed. One of them is the PDF synchronization implemented in the pdfsync package. This technology has been explained, with its benefits and its limitations. Routes towards a better user experience have been considered. The article is mainly concerned with LaTeX and PDF output. [9]

*ACIDE: An Integrated Development Environment Configurable for LaTeX* introduces the configurable integrated development environment ACIDE, which is an ongoing development project currently in alpha status. It is cross-platform, open-source, and free, and will be distributed under GPL. Although targeted to any

programming language environment, including compilers, interpreters, and database systems, in particular it is well-suited to tasks required for LaTeX and TEX document preparation systems. It manages projects, is useful in dealing with multifile documents, allows configurable menus, has a toolbar for executing commands and lexical tokens for syntax coloring, and even grammars to identify programming (syntactical) errors on the fly. [10]

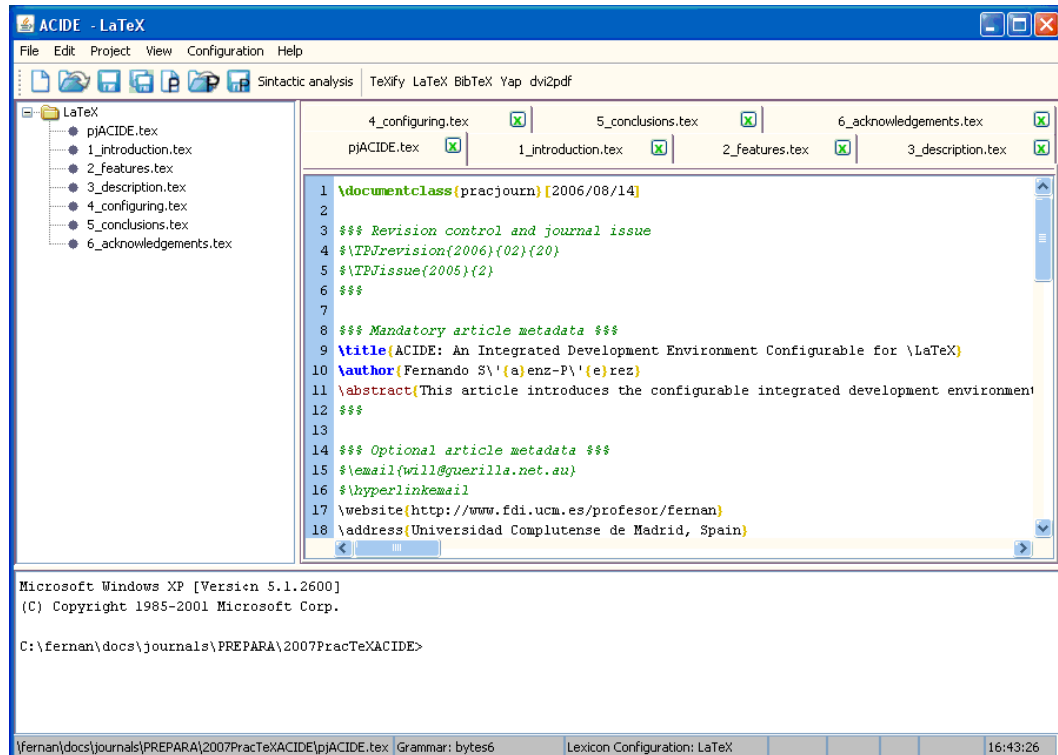


Figure 1: ACIDE Main GUI

Free and commercial LaTeX<sub>2 $\epsilon$</sub>  implementations for the Mac OS X are available on the internet. For people who have always used Linux and now find themselves in front of a brand new Mac OS X machine and have no time to learn new tools from scratch, their best option would be to use your familiar Linux applications on Mac OS X. A LaTeX<sub>2 $\epsilon$</sub>  “Linux-like” environment on Mac OS X aims to show how to install and configure a Linux-like LaTeX<sub>2 $\epsilon$</sub>  environment in the Mac OS X, using Fink, teTeX and Kile. [11]



The following are, so far, the only journals in The PracTeX Journal dealing with LaTeX editors:

*LaTeXing with TextMate* discusses the TextMate text editor and its many capabilities that make working with LaTeX documents a lot easier. Some of its features include syntax highlighting, various methods for automatic insertion of text (such as the begin-end blocks in environments and automatic labels for section commands), and lookup of labels and cite keys based on partial matches, as well as tools for dealing with large projects. TextMate is designed with the user in mind, so it is easy to customize it to your needs. During its short lifetime (about two and a half years) it has gained many supporters and has become a very popular text editor for the Mac OS X platform, and especially among LaTeX users, as can be seen from the exponential growth in the number of users, and the large number of LaTeX related questions on the TextMate mailing list. [12]

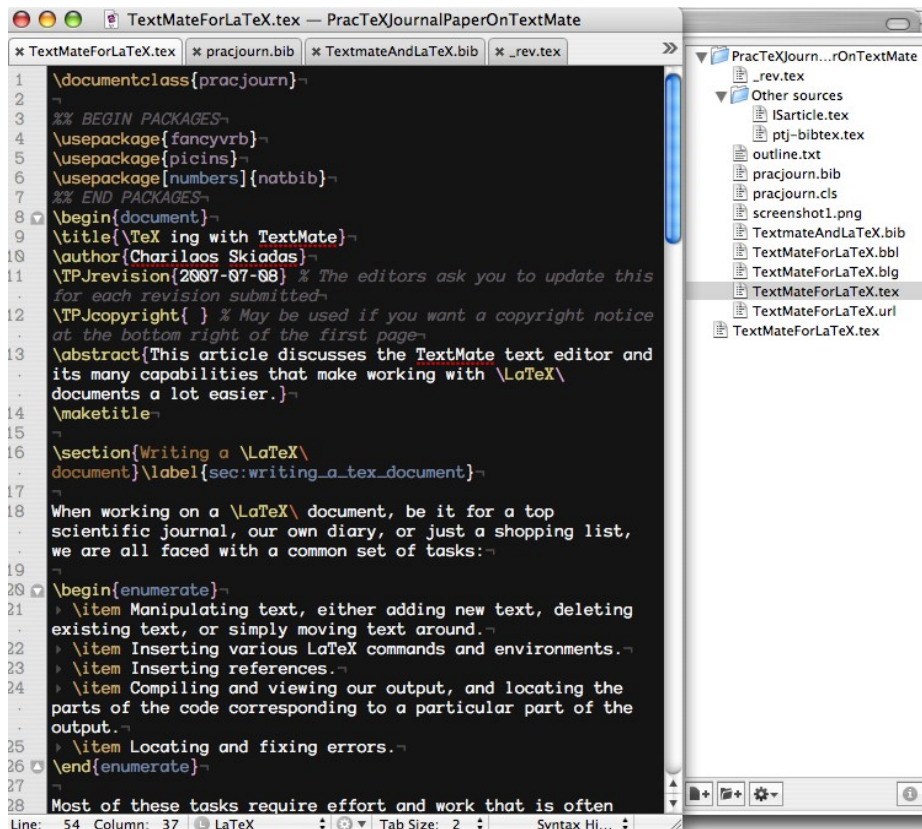


Figure 2: TextMate's Project Drawer, and Syntax Highlighting

*TpX* — *LaTeX-oriented graphical editor* describes *TpX*, a lightweight, easy-to-use graphical editor for Windows platform, presents guidelines for its use and discusses some features and limitations. [13]

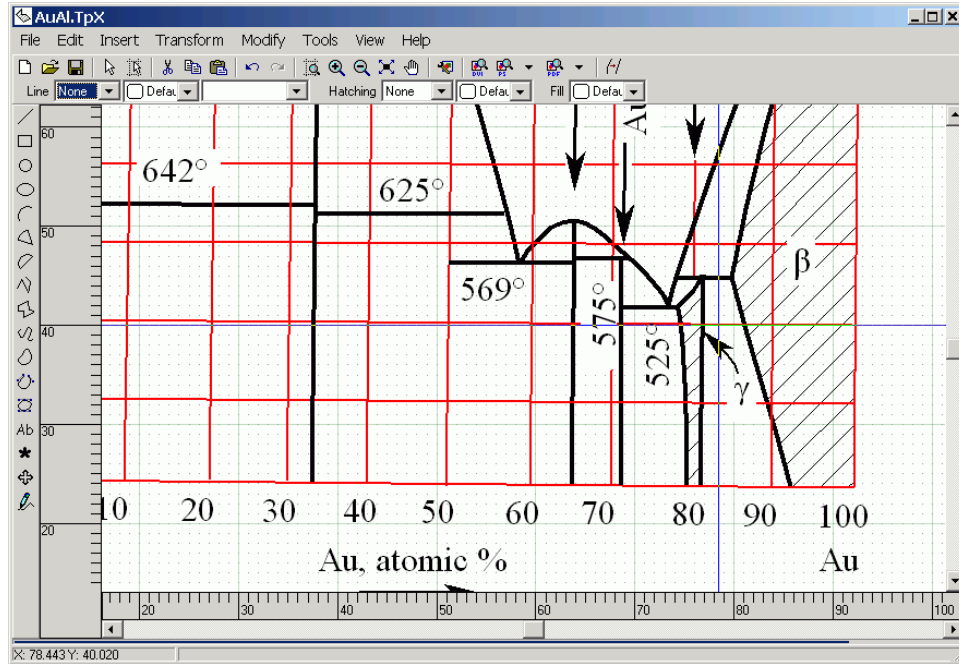


Figure 3: *TpX* screenshot

*IPE—a graphics editor with LaTeX integration* talks about author’s experiences with the IPE graphics editor. The main features of IPE are described, as well as some advanced usage. A similar editor, vrr, is also mentioned. [14]

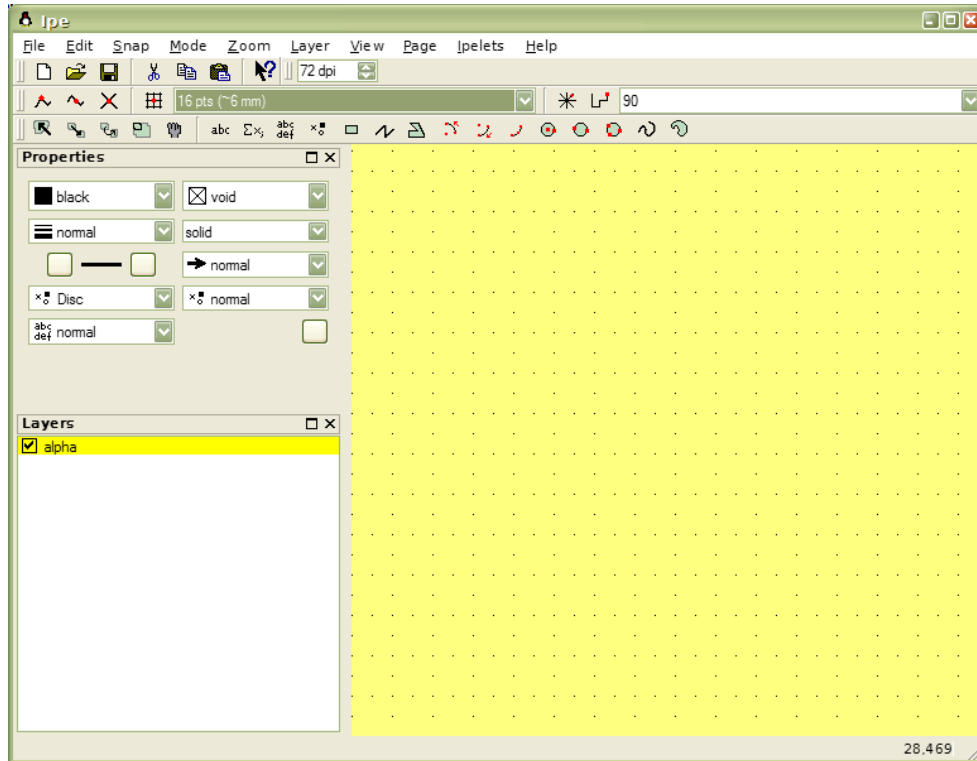


Figure 4: A new IPE window

Others such as *Minutes in Less Than Hours: Using LaTeX Resources* [15], *Tables in LaTeX2 $\epsilon$ : Packages and Methods* [16], *Creating Book Covers using PSTricks* [17], *Graphics in LaTeX* [18], and *A Tool for Logicians* [19] are just some of the articles dedicated to giving guidance on how explore the capabilities of LaTeX and how to improve your work by using other LaTeX packages and resources.

### III. THEORETICAL FRAMEWORK

#### A. TeX and LaTeX

TeX is both a program and a language. It is a program that uses the TeX typesetting language to create finished documents. [1]

LaTeX is a document preparation system. You use LaTeX to create documents for others to read. In that respect it is similar to Microsoft Word. But the similarities end there. Document preparation with LaTeX typically consists of using a text editor (such as Emacs, vi, or even Notepad) to edit a LaTeX source file, which has the extension .tex, and then running the LaTeX program to convert the source file to a document interchange format such as Postscript or PDF. Once the document is in a document interchange format, it can be previewed on the screen, sent to others, printed, etc. [20]

#### *The levels of TeX*

The word “TeX” can be used to refer to any of a myriad of items at any level. Starting at the top:

1. *Distributions*: MiKTeX, TeX Live, ... These are the large collections of TeX-related software to be downloaded and installed. When someone says “I need TeX on my system”, they're usually looking for a distribution.
2. *Front ends*: Emacs, TeXShop, TeXnicCenter, WinEdt, ... These editors are what you use to create a document file. Some (e.g., TeXShop) are devoted to TeX specifically, others (e.g., Emacs) can be used to edit any sort of file. TeX documents are independent of any particular editor; the actual TeX typesetting program itself does not include any sort of editor whatsoever. This is in stark contrast to programs such as Microsoft Word, where editing and typesetting are fundamentally intertwined.
3. *Packages*: geometry, lm, ... These are add-ons to the basic TeX system, developed independently, providing additional typesetting features, fonts,

documentation, etc. The CTAN web sites provide access to the vast majority of packages in the TeX world.

4. *Formats*: LaTeX, plain TeX, ... These are the TeX-based languages in which you actually write documents. When someone says “TeX is giving me this mysterious error”, they usually mean a format.
5. *Engines*: TeX, pdfTeX, XeTeX, LuaTeX, ... These are the actual executable binaries which implement the different TeX dialects. When someone says “TeX can't find my fonts”, they usually mean an engine. [21]

<pre>\documentclass[a4paper,11pt]{article} % define the title \author{H.~Partl} \title{Minimalism} \begin{document} % generates the title \maketitle % insert the table of contents \tableofcontents \section{Some Interesting Words} Well, and here begins my lovely article.  \section{Good Bye World} Good bye! \end{document}</pre>	<p style="text-align: center;">Minimalism</p> <p style="text-align: center;">H. Partl</p> <p style="text-align: center;">September 27, 2007</p> <p><b>Contents</b></p> <table><tr><td>1</td><td>Some Interesting Words</td><td>1</td></tr><tr><td>2</td><td>Good Bye World</td><td>1</td></tr></table> <p>1 Some Interesting Words</p> <p>Well, and here begins my lovely article.</p> <p>2 Good Bye World</p> <p>and the least well understood physical formula.</p>	1	Some Interesting Words	1	2	Good Bye World	1
1	Some Interesting Words	1					
2	Good Bye World	1					

Figure 5: Simple LaTeX command with corresponding output

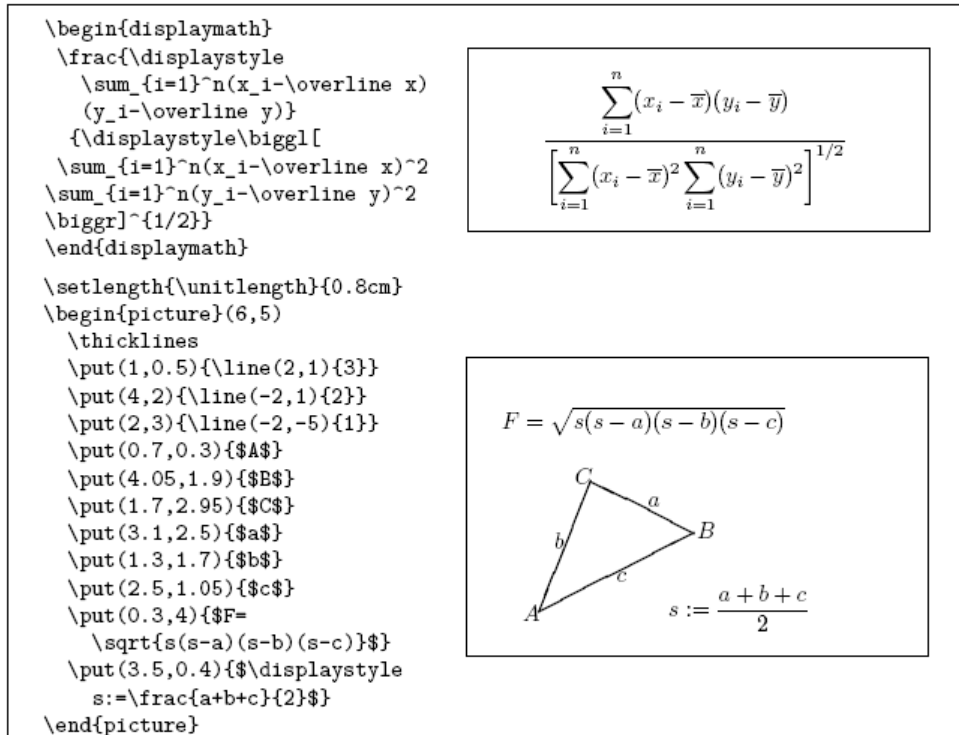


Figure 6: Simple LaTeX math commands with corresponding output

## B. Document Markup Language

A markup language combines text and extra information about the text. The extra information, for example about the text's structure or presentation, is expressed using markup, which is intermingled with the primary text. Originally, markup was used in the publishing industry in the communication of printed work between authors, editors, and printers. [22]

Document markup is the process of adding codes to a document to identify the structure of a document or the format in which it is to appear. Until the computerization of the printing industry, markup was primarily done by a copy editor writing instructions on a manuscript for a typesetter to follow ( Figure 1 ). Over a period of time, a standard set of symbols was developed and used by copy editors to communicate with typesetters.

As typesetting functions became computerized, text formatting languages were written. A typesetter would convert the copy editor's markup into the appropriate markup for the text formatting language being used. [23]

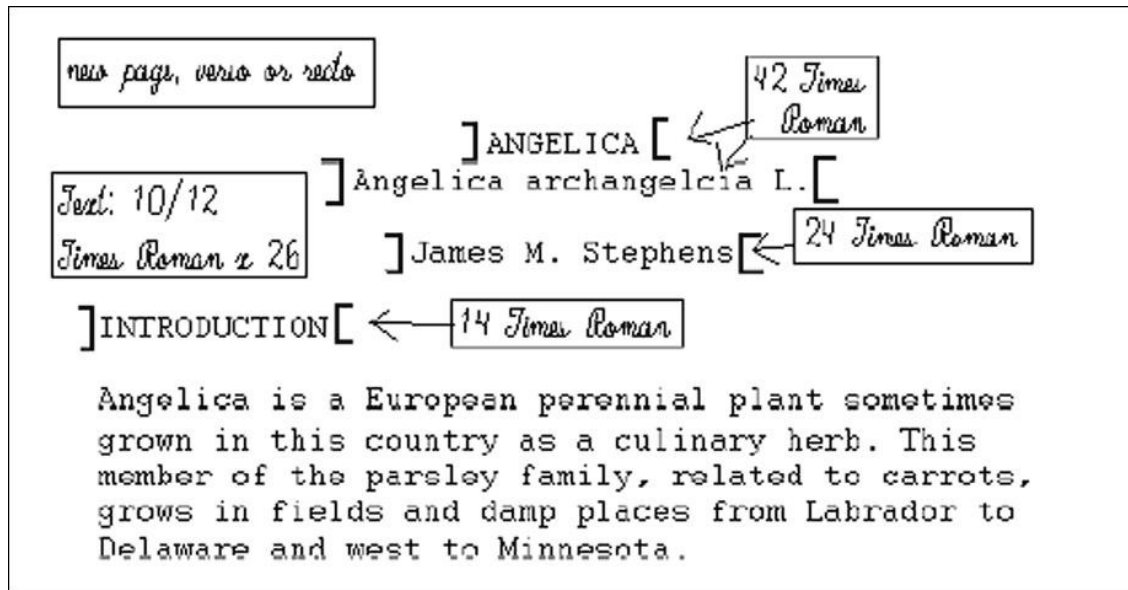


Figure 7. Portion of a marked-up page with instructions for a typesetter.

### C. Editor

A text editor is a type of program used for editing plain text files. Text editors are often provided with operating systems or software development packages, and can be used to change configuration files and programming language source code. [24]]

Typical features of an editor include:

- Search and replace - The process of searching for a word or string in a text file and optionally replacing the search string with a replacement string.
- Copy, cut and paste - Most text editors provide methods to duplicate and move text within the file, or between files.
- Text formatting - Text editors often provide basic formatting features like line wrap, auto-indentation, bullet list formatting, comment formatting, and so on.

- Undo and redo - Often—especially with older text editors—there is only one level of edit history remembered and successively issuing the undo command will only "toggle" the last change. Modern or more complex editors usually provide a multiple level history such that issuing the undo command repeatedly will revert the document to successively older edits. A separate redo command will cycle the edits "forward" toward the most recent changes. The number of changes remembered depends upon the editor and is often configurable by the user.
- Importing - Reading or merging the contents of another text file into the file currently being edited. Some text editors provide a way to insert the output of a command issued to the operating system's shell.
- Filtering - Some advanced text editors allow you to send all or sections of the file being edited to another utility and read the result back into the file in place of the lines being "filtered". This, for example, is useful for sorting a series of lines alphabetically or numerically, doing mathematical computations, and so on.



## IV. DESIGN AND IMPLEMENTATION

### A. Context Diagram

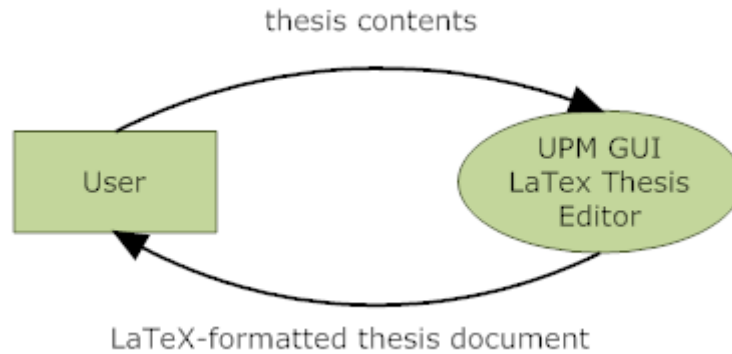


Figure 8: Context diagram of the system

### B. Data Flow Diagram

The UPM GUI LaTeX Thesis Editor to be developed is summarized by the Data flow Diagram in figure 9. To start, the user can either create or open a document. Once a document is opened or created, thesis contents entered using the templates can be saved to a GLP file, compiled to produce a TeX file, or previewed, creating a PDF file to be displayed afterwards.

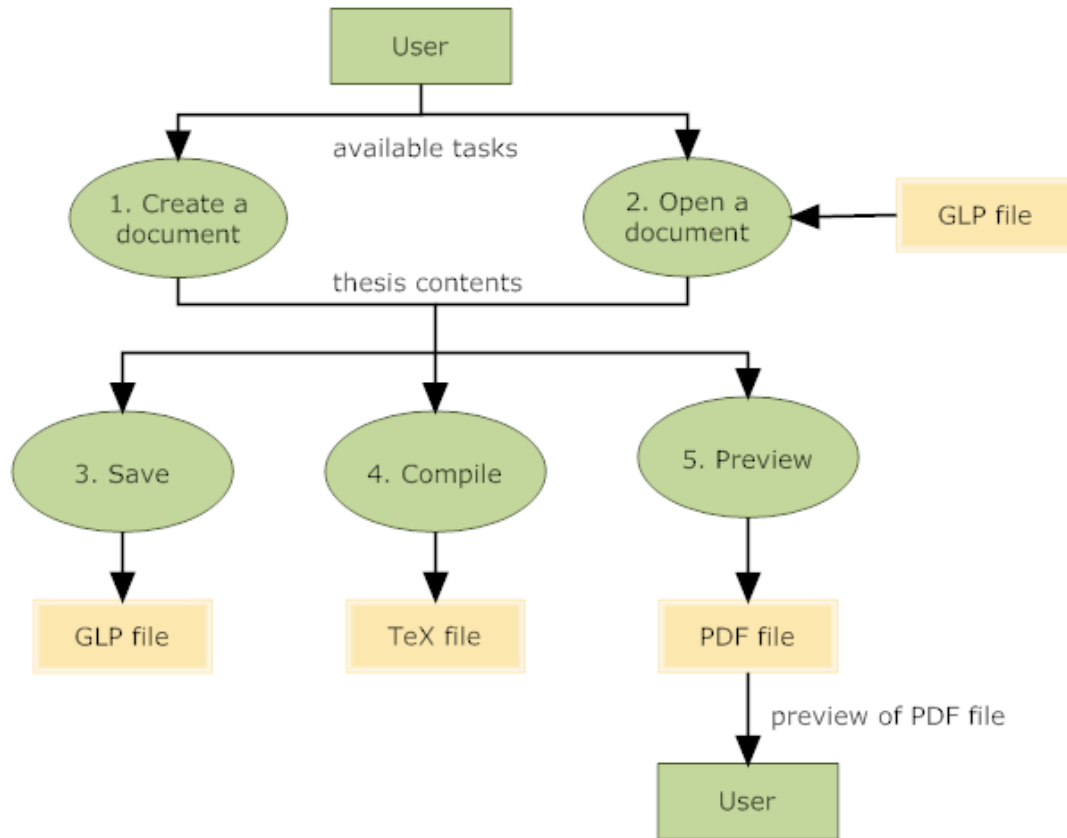


Figure 9: Data Flow Diagram of the system

## **C. Data Dictionary**

GUI (Graphical User Interface) - a type of user interface which allows people to interact with a computer and computer-controlled devices which employ graphical icons, visual indicators or special graphical elements called "widgets", along with text, labels or text navigation to represent the information and actions available to a user.

Typesetting – involves the presentation of textual material in graphic form on paper or some other medium.

LaTeX command – commands used to provide the desired attribute of the text

LaTeX environment – a set of commands used to apply multiple attributes on a series of text

Document Preparation System – a computer application used for the production (including composition, editing, formatting, and possibly printing) of any sort of printable material.

WYSIWYG – What You See Is What You Get, used in computing to describe a system in which content during editing appears very similar to the final product. It is commonly used for word processors, but has other applications, such as Web (HTML) authoring.

Package – a file in STY format which will contain the LaTeX commands and environments.

PDF format (Portable Document) – file format created by Adobe Systems document exchange. PDF is used for representing two-dimensional documents in a device-independent and display resolution-independent fixed-layout document format.

## V. RESULTS

The New Project page of the UPM GUI LaTeX Thesis Editor is shown in figure 10. This is the first page displayed upon opening the program.

The right side of the screen contains the Navigator, which is a list of the different parts of a standard thesis document, as required by our university. Clicking on an item will display its corresponding template. Below this pane are the buttons Add Subsection and Remove Subsection for adding and/or removing items from the Navigator list.

The middle panel contains the template for entering intended thesis document contents in plain text format. Contents of this panel changes depending on selected item in the Navigator Panel.

The right panel displays the common special characters available for input. Clicking on an item will insert an icon in the template.

The top portion of the page contains the menu bar for other user available actions.

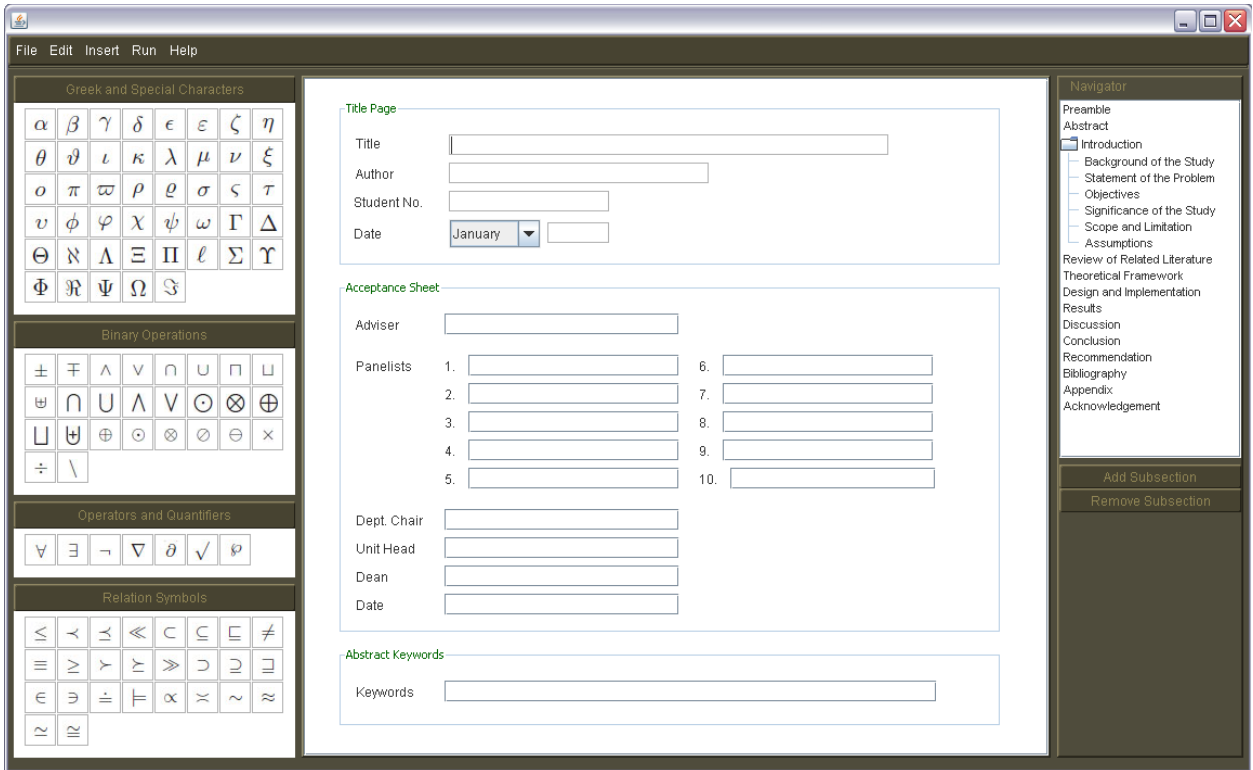


Figure 10: New Project page of the UPM GUI LaTeX Thesis Editor

Figures 11 to 14 shows additional forms for inserting items such as images, equations, and citations.

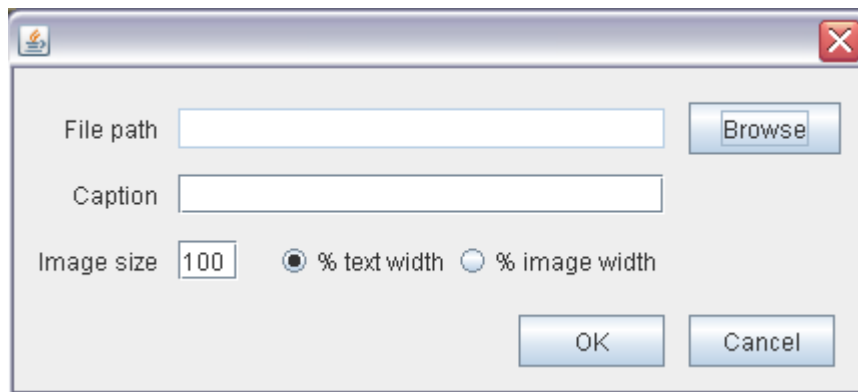


Figure 11: Insert Image form

Figures 12 and 13 show the form displayed for inserting an equation, and the main page of TeXaide's equation editor. Choosing to insert an equation will automatically open TeXaide. After creating equations here, user can copy its contents and paste it in the equation form in order to retrieve its corresponding LaTeX script.

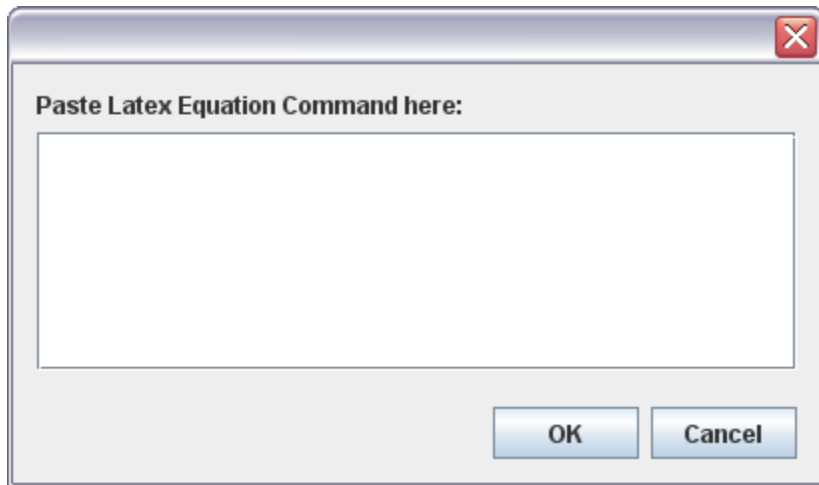


Figure 12: Insert Equation form

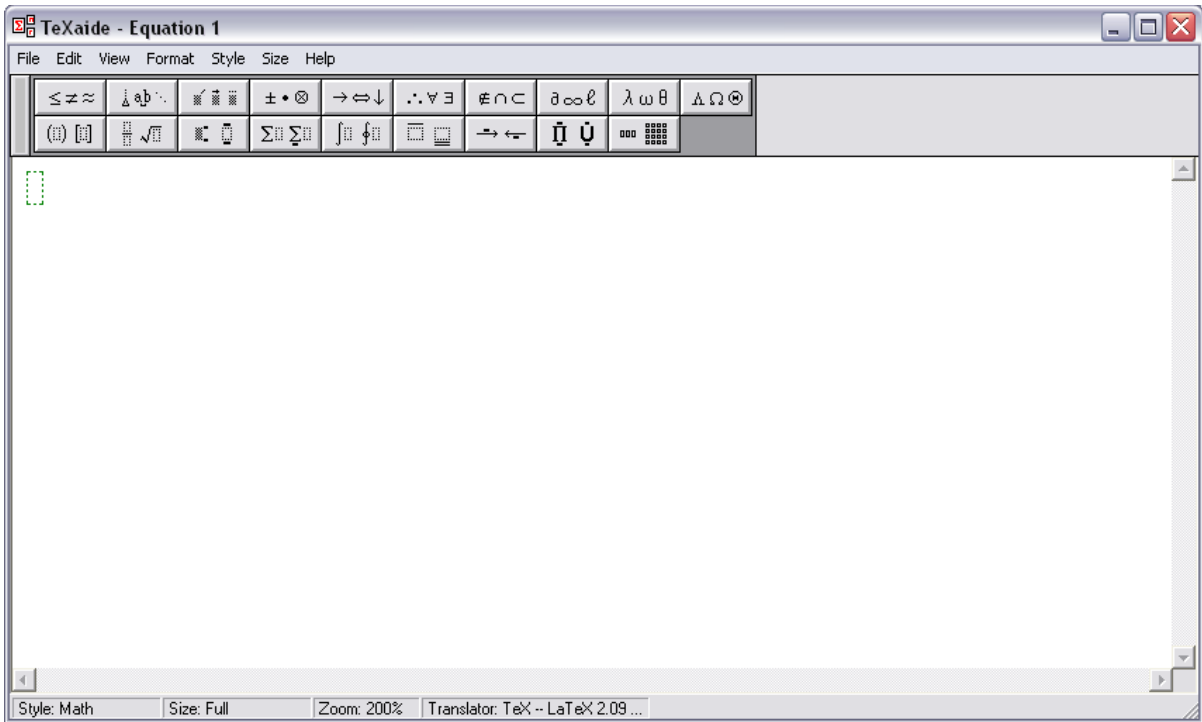


Figure 13: TeXaide equation editor

Figures 14 and 15 show two types of Insert Citation forms. The first form is displayed when no citations have been added yet, while the second form is displayed when user has already added at least one citation. Here, the user can choose to create a new citation, reuse an existing citation, or delete it.

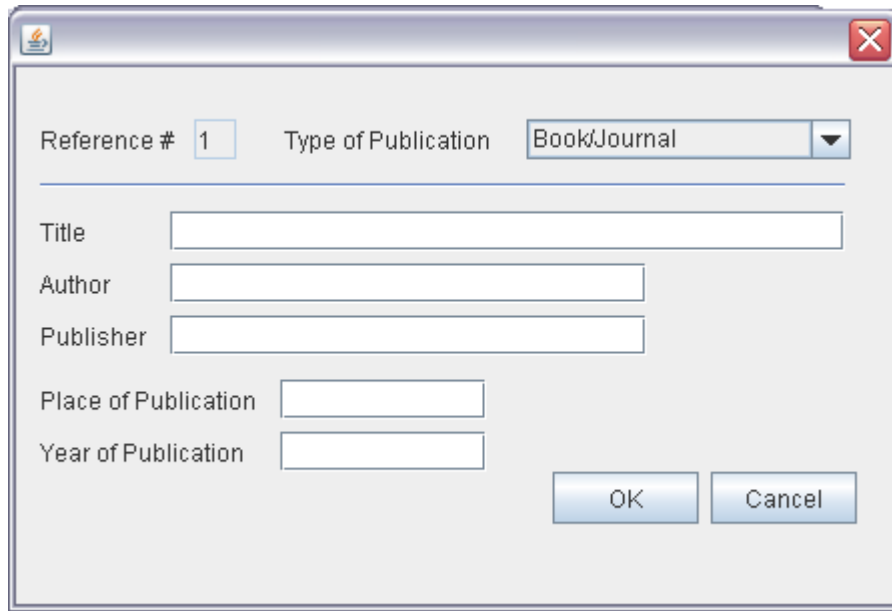


Figure 14: Insert Citation form – no existing citations

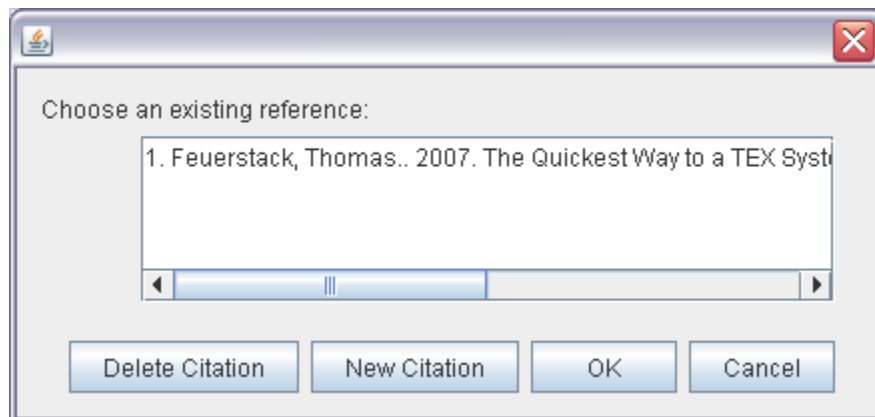


Figure 15: Insert Citation form – with existing citations

Refer to Appendix for a sample thesis document created using the program.



## **VI. DISCUSSION**

The UPM GUI LaTeX Thesis Editor is a tool developed to standardize the structure and format of thesis documents in the university by taking advantage of the capabilities and benefits of LaTeX programming. It has a GUI editor which provides templates and buttons allowing non-LaTeX programmers to create standard thesis documents without the need to learn LaTeX commands and syntax.

With a GUI editor capable of handling tasks related to document formatting and presentation, the UPM GUI LaTeX Thesis Editor provides users with the advantage of not having to worry over their document's structure, and focusing on its contents instead. At the same time, it allows defense panel members to skip format-related checking and also focus on the contents. As such, this system helps lessen turn-around time and streamlines the thesis document preparation process.

As the GUI editor only offers basic LaTeX features, there may be instances when user will have to find work-around solutions, or have to learn LaTeX syntax and switch to using the text editor in order to meet his needs. And while the templates provide a small room for flexibility of adding and modifying chapters, the default chapters provided for are only those normally used in Special Projects required from BS Computer Science students.

## **VII. CONCLUSION**

The UPM GUI LaTeX Thesis Editor streamlines the thesis document preparation process by providing means to standardize the style and structure of UPM thesis documents. This application uses LaTeX programming techniques and features, and incorporates thesis document formatting standards set by the university to automatically format thesis data entered by the user, relieving him of the need to observe and apply proper formatting manually.

The system provides a GUI interface on which the user can work on so he'll be able to produce a working thesis document without having to learn LaTeX commands. The interface consists of templates for the different parts of a thesis document. This data can then be compiled to create a LaTeX-formatted thesis document.

Even with users being completely unknowledgeable on the formatting standards and on LaTeX programming, the system is capable of converting data entered by user into error-free LaTeX scripts, and producing well-formatted thesis documents.

## **VIII. RECOMMENDATION**

The UPM GUI LaTeX Thesis Editor can be enhanced further to cater not only to students of the university but also to anyone who has the need to prepare well-formatted documents.

For one, the templates can be improved to provide more options and features to support requirements beyond those which are normally used in thesis documents in our university.

The system can also improve a lot in terms of the type of document it produces. More templates can be provided to support easy creation of other documents often used in the university such as letters, examination papers, etc.

A possible approach to handle the aforementioned enhancements is to make use of uploadable template files storing details of the formats to be applied, and the fields supposed to be filled up by users. The system can also be implemented using simpler tools which are easier to implement yet provide more flexibility such as Macro programming in Microsoft Excel.

## IX. BIBLIOGRAPHY

- [1] Feuerstack, Thomas. 2007. *The Quickest Way to a TEX System based on MiKTeX, TeXnicCenter, Ghostscript and GSview.*
- [2] TeX  
<http://en.wikiPEdia.org/wiki/TeX>
- [3] List of Document Markup Languages  
[http://en.wikiPEdia.org/wiki/List\\_of\\_document\\_markup\\_languages](http://en.wikiPEdia.org/wiki/List_of_document_markup_languages)
- [4] LaTeX  
<http://en.wikiPEdia.org/wiki/LaTeX>
- [5] Oetiker, Tobias et al. 2005. *The Not So Short Introduction to LaTeX2 $\epsilon$ .* Massachusetts: Free Software Foundation, Inc.
- [6] The PracTeX Journal  
<http://www.tug.org/pracjourn/2005-1/carnes-welcome/>
- [7] Roberts, Andy. 2007. *In My Opinion: LaTeX isn't for everyone but it could be for you.*  
<http://www.tug.org/pracjourn/2006-2/roberts/#RobertsArticle>
- [8] Song Loong, Theresa. 2007. *The Beginner's Forest of LaTeX.*  
[http://www.tug.org/pracjourn/2007-3/song\\_loong/song\\_loong.pdf](http://www.tug.org/pracjourn/2007-3/song_loong/song_loong.pdf)
- [9] Jérôme Laurens. 2007. *Will TEX ever be wysiwyg or the PDF synchronization story.*  
<http://www.tug.org/pracjourn/2007-3/laurens/laurens.pdf>
- [10] S'aenz-P'erez, Fernando. 2007. *ACIDE: An Integrated Development Environment Configurable for LaTeX.*  
[http://www.tug.org/pracjourn/2007-3/saenz\\_perez-acide/saenz\\_perez-acide.pdf](http://www.tug.org/pracjourn/2007-3/saenz_perez-acide/saenz_perez-acide.pdf)
- [11] Provenzano, Vinicius. *A LaTeX2 $\epsilon$  "Linux-like" environment on Mac OS X.*  
<http://www.tug.org/pracjourn/2007-3/provenzano/provenzano.pdf>
- [12] Skiadas, Charilaos, et al. *LaTeXing with TextMate.*  
<http://www.tug.org/pracjourn/2007-3/skiadas-textmate/skiadas-textmate.pdf>
- [13] Tsyplakov, Alexander. 2007. *TpX — LaTeX-oriented graphical editor.*

- <http://www.tug.org/pracjournal/2007-3/tsyplakov/tsyplakov.pdf>
- [14] Jan Hlavacek. 2006. *IPE—a graphics editor with LaTeX integration*.  
<http://www.tug.org/pracjournal/2006-2/hlavacek/hlavacek.pdf>
- [15] Hefferon, Jim. 2005. *Minutes in Less Than Hours: Using LaTeX Resources*.  
<http://www.tug.org/pracjournal/2005-4/hefferon/hefferon.pdf>
- [16] Mori, Lapo Filippo. 2007. *Tables in LaTeX2#: Packages and Methods*.  
<http://www.tug.org/pracjournal/2007-1/mori/mori.pdf>
- [17] Robbers, Yuri et al. 2007. *Creating Book Covers using PSTricks*.  
<http://www.tug.org/pracjournal/2007-1/robbers/robbers.pdf>
- [18] Beccari, Claudio. 2007. *Graphics in LaTeX*.  
<http://www.tug.org/pracjournal/2007-1/beccari/beccari.pdf>
- [19] Buchsbaum, Arthur. 2007. *A Tool for Logicians*.  
<http://www.tug.org/pracjournal/2007-3/buchsbaum/buchsbaum.pdf>
- [20] What the heck is LaTeX?  
<http://www.cs.berkeley.edu/~smcpeak/LaTeX/whatisLaTeX.html>
- [21] LaTeX vs. MiKTeX: The levels of TeX  
<http://www.tug.org/levels.html>
- [22] Brief History of Document Markup  
<http://edis.ifas.ufl.edu/AE038>
- [23] Markup Language  
[http://en.wikiPEdia.org/wiki/Document\\_markup\\_language](http://en.wikiPEdia.org/wiki/Document_markup_language)
- [24] Text editor  
[http://en.wikiPEdia.org/wiki/Text\\_editor](http://en.wikiPEdia.org/wiki/Text_editor)

## X. APPENDIX

```
/** Chapter.java*/
```

```
package upmguilte;
```

```
import java.awt.Insets;  
import java.awt.event.KeyEvent;  
import java.awt.event.KeyListener;  
import java.util.ArrayList;
```

```
import javax.swing.BorderFactory;  
import javax.swing.GroupLayout;  
import javax.swing.GroupLayout.Alignment;  
import javax.swing.JEditorPane;  
import javax.swing.JLabel;  
import javax.swing.JPanel;  
import javax.swing.JScrollPane;  
import javax.swing.LayoutStyle.ComponentPlacement;  
import javax.swing.ScrollPaneConstants;  
import javax.swing.undo.CannotRedoException;  
import javax.swing.undo.CannotUndoException;  
import javax.swing.undo.UndoManager;  
import javax.swing.undo.UndoableEdit;
```

```
class Chapter {  
    JPanel panel;  
    JEditorPane textPane;  
    JScrollPane scrollPane;  
    JLabel title;  
    int chapNum;  
    String chapTitle = "";  
    boolean hasSubsection = false;  
    UndoManager manager;  
    ArrayList<UndoManager> subManager;  
    ArrayList<JPanel> subsection;  
    ArrayList<JEditorPane> subTextPane;  
    ArrayList<String> subsecTitle;  
    UndoableEdit ue;
```

```
    public Chapter(int chapNum, String chapTitle){  
        this.chapNum = chapNum;  
        this.chapTitle = chapTitle;
```

```
        panel = new JPanel();  
        panel.setBackground(new java.awt.Color(224,  
216, 182));  
        textPane = new JEditorPane("text/html",  
"<body></body>");  
        textPane.setMargin(new Insets(10,10,10,10));
```

```
        textPane.addKeyListener(new KeyListener() {  
            public void keyTyped(KeyEvent e) {  
                if (e.paramString().contains("Enter")) {  
                    int caretPos = textPane.getCaretPosition();  
                    newProject.insertInto(textPane,"<input  
type=\"hidden\">", caretPos);  
                    String origText = textPane.getText();  
                    String newText =  
origText.replaceFirst("</li>\\s*</  
[ou]>\\s*(</li>\\s*</[ou]>\\s*)?<input  
type=\"hidden\">(\\s*<[ou]>\\s*<li>)?\\s*<[ou]>\\  
\\s*<li>", "</li><li>");
```

```
                if (!origText.equals(newText)) {  
                    textPane.setText(newText);  
                    textPane.setCaretPosition(caretPos);  
                } else {
```

```
                    textPane.setText(origText.replaceFirst("<in  
put type=\"hidden\">", "<br>"));  
                    textPane.setCaretPosition(caretPos);  
                }  
            } else if (e.paramString().contains("Tab")) {  
                int origCaretPos =  
textPane.getCaretPosition();  
                textPane.setText(textPane.getText().replace  
First("\\t", ""));  
                textPane.setCaretPosition(origCaretPos-1);  
            }  
        }  
        public void keyPressed(KeyEvent e) {}  
        public void keyReleased(KeyEvent e) {}  
    });
```

```
        manager = new UndoManager();  
        ue = new UndoableEdit() {  
            public void undo() throws  
CannotUndoException { }  
            public boolean canUndo() { return true; }  
            public void redo() throws CannotRedoException  
{ }  
            public boolean canRedo() { return true; }  
            public void die() { }  
            public boolean addEdit(UndoableEdit anEdit)  
{ return true; }  
            public boolean replaceEdit(UndoableEdit  
anEdit) { return true; }  
            public boolean isSignificant() { return true; }  
            public String getPresentationName() { return  
""; }  
            public String getUndoPresentationName()  
{ return ""; }  
            public String getRedoPresentationName()  
{ return ""; }  
        };
```

```
        textPane.getDocument().addUndoableEditListen  
er(manager);
```

```
        scrollPane = new JScrollPane();  
        scrollPane.setViewportView(textPane);  
        scrollPane.setHorizontalScrollBarPolicy(ScrollPan  
eConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED  
);  
        scrollPane.setBorder(BorderFactory.createEmpty  
Border(1, 1, 1, 1));
```

```
        title = new JLabel();  
        title.setForeground(new java.awt.Color(51, 102,  
0));  
        if ((chapNum != 0))  
            title.setText("Chapter " + chapNum + ": " +  
chapTitle);  
        else  
            title.setText(chapTitle);  
        title.setFont(new java.awt.Font("Georgia", 1,  
14));
```

```
        GroupLayout temp = new GroupLayout(panel);  
        panel.setLayout(temp);
```

```
        temp.setHorizontalGroup(  
            temp.createParallelGroup(Alignment.LEADING)  
                .addGroup(temp.createSequentialGroup()  
                    .addGap(20, 20, 20)
```

```

        .addGroup(temp.createParallelGroup(Alignment.LEADING)
        .addComponent(scrollPane,
        GroupLayout.DEFAULT_SIZE, 355,
        Short.MAX_VALUE)
        .addComponent(title)
        .addGap(21, 21, 21))
    );
    temp.setVerticalGroup(
    temp.createParallelGroup(Alignment.LEADING)
    .addGroup(temp.createSequentialGroup()
    .addGap(16, 16, 16)
    .addComponent(title)
    .addPreferredGap(ComponentPlacement.UNRELATED)
    .addComponent(scrollPane,
    GroupLayout.DEFAULT_SIZE, 332,
    Short.MAX_VALUE)
    .addGap(19, 19, 19))
    );
}

public void Subsection (String[] title) {
    hasSubsection = true;
    subsection = new ArrayList<JPanel>();
    subTextPane = new ArrayList<JEditorPane>();
    subsecTitle = new ArrayList<String>();
    subManager = new ArrayList<UndoManager>();
    if (title != null) {
        for (int i=0; i<title.length; i++) {
            addSubsection(0, title[i]);
        }
    }
}

public void addSubsection(int index, String title) {
    //append to last
    int last = subsection.size();
    if (index == 0) {
        subsection.add(new JPanel());
        subTextPane.add(new JEditorPane("text/html",
"<body></body>"));
        subManager.add(new UndoManager());
        subsecTitle.add(title);
        newSubSection(subsection.get(last),
subTextPane.get(last), last, title,
subManager.get(last));
        //insert section
    } else {
        subsection.add(index, new JPanel());
        subTextPane.add(index, new
JEditorPane("text/html", "<body></body>"));
        subManager.add(index, new UndoManager());
        subsecTitle.add(index, title);
        newSubSection(subsection.get(index),
subTextPane.get(index), index, title,
subManager.get(index));
    }
}

public void remSubsection(int index) {
    index = index%100-1;
    subsection.remove(index);
    subTextPane.remove(index);
    subManager.remove(index);
    subsecTitle.remove(index);
    if(subsection.size()==1) {
        hasSubsection = false;
    }
}
}

private void newSubSection(JPanel
subsection,final JEditorPane subTextPane, int
index, String subSechapTitle, UndoManager
manager) {
    JLabel subSecLabel = new JLabel();
    JScrollPane subScrollPane = new JScrollPane();
    JLabel chapLabel = new JLabel();

    chapLabel.setText("Chapter "+chapNum+":
"+chapTitle);
    chapLabel.setFont(new java.awt.Font("Georgia",
0, 11));
    chapLabel.setForeground(new
java.awt.Color(79, 64, 39));

    subTextPane.getDocument().addUndoableEditLi
stener(manager);
    subTextPane.addKeyListener(new KeyListener()
{
    public void keyTyped(KeyEvent e) {
        if (e.paramString().contains("Enter")) {
            int caretPos =
subTextPane.getCaretPosition();
            newProject.insertInto(subTextPane,"<input
type=\"hidden\">", caretPos);
            String origText = subTextPane.getText();
            String newText =
origText.replaceFirst("</li>\s*</
[ou]>\s*</li>\s*</[ou]>\s*"?<input
type=\"hidden\">(\s*<[ou]>\s*<li>)?\s*<[ou]>\
\s*<li>", "</li><li>");
            if (!origText.equals(newText)) {
                subTextPane.setText(newText);
                subTextPane.setCaretPosition(caretPos);
            } else {
                subTextPane.setText(origText.replaceFirst(
"<input type=\"hidden\">", "<br>"));
                subTextPane.setCaretPosition(caretPos);
            }
        } else if (e.paramString().contains("Tab")) {
            int origCaretPos =
subTextPane.getCaretPosition();
            subTextPane.setText(subTextPane.getText().
replaceFirst("\t", ""));
            subTextPane.setCaretPosition(origCaretPos-
1);
        }
    }
    public void keyPressed(KeyEvent e) {}
    public void keyReleased(KeyEvent e) {}
});

    subsection.setBackground(new
java.awt.Color(224, 216, 182));

    subSecLabel.setText("/*letter[index]+".
"+*/subSechapTitle);
    subSecLabel.setFont(new
java.awt.Font("Georgia", 1, 14));
    subSecLabel.setForeground(chapLabel.getForeg
round());

    subScrollPane.setViewportView(subTextPane);
    subScrollPane.setHorizontalScrollBarPolicy(Scroll
PaneConstants.HORIZONTAL_SCROLLBAR_AS_NEE
DED);
}
}

```

```

subScrollPane.setBorder(BorderFactory.createE
mptyBorder(1, 1, 1, 1));

GroupLayout layout = new
GroupLayout(subsection);
subsection.setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(Alignment.LEADING
)
.addGroup(Alignment.TRAILING,
layout.createSequentialGroup())
.addGroup(layout.createParallelGroup(Alignm
ent.TRAILING)
.addGroup(Alignment.LEADING,
layout.createSequentialGroup())
.addContainerGap()
.addComponent(subScrollPane,
GroupLayout.DEFAULT_SIZE, 376,
Short.MAX_VALUE))
.addGroup(Alignment.LEADING,
layout.createSequentialGroup())
.addGroup(layout.createParallelGroup(Alignm
ent.TRAILING)
.addGroup(Alignment.LEADING,
layout.createSequentialGroup())
.addContainerGap()
.addComponent(chapLabel,
GroupLayout.DEFAULT_SIZE,
GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
.addGroup(Alignment.LEADING,
layout.createSequentialGroup())
.addGap(34, 34, 34)
.addComponent(subSecLabel,
GroupLayout.DEFAULT_SIZE, 131,
Short.MAX_VALUE))
.addGap(221, 221, 221)))
.addContainerGap())
);
layout.setVerticalGroup(
layout.createParallelGroup(Alignment.LEADING
)
.addGroup(layout.createSequentialGroup())
.addContainerGap()
.addComponent(chapLabel)
.addGap(13, 13, 13)
.addComponent(subSecLabel)
.addPreferredGap(ComponentPlacement.RELA
TED)
.addComponent(subScrollPane,
GroupLayout.DEFAULT_SIZE, 224,
Short.MAX_VALUE)
.addContainerGap())
);
}
}

```

**/\*\* Citation.java \*/**

```

package upmguilte;

public class Citation {
private int cnt;
private String citation;

public Citation(int cnt, String citation) {
this.cnt = cnt;
this.citation = citation;
}
}

```

```

public int getCitationCnt() {
return cnt;
}

public void setCitationCnt(int cnt) {
this.cnt = cnt;
}

public String getCitation() {
return citation;
}
}

/** compileAndRun.java */

package upmguilte;

import java.io.File;
import java.util.ArrayList;
import java.util.Map;
import java.util.Map.Entry;

import javax.swing.JEditorPane;
import javax.swing.JOptionPane;

class compileAndRun{
String tex = "";
String newline = "\n";
String title = "not set", author = "not set",
studNum = "not set", month = "not set", year =
"not set",
adviser = "notset", dean = "notset",
deptChair = "notset", date = "notset", unitHead =
"notset",
keywords = "notset";
ArrayList<String> panelists;

public compileAndRun() {}

public void convertProjectToTeX(){
tex = "\\documentclass[11pt, letterpaper,
oneside]{book}" + newline +
newline +
"\\usepackage{graphicx}" + newline +
"\\usepackage{multicol}" + newline +
"\\usepackage{url}" + newline +
"\\renewcommand{\\thesecion}
{\\Alph{section}.}" + newline +
"\\renewcommand{\\contentsname}{Table
of Contents}" + newline +
newline +
"\\pdfpagewidth 8.5in" + newline +
"\\pdfpageheight 11in" + newline +
newline +
"\\setlength\\topmargin{0in}" + newline +
"\\setlength\\oddsidemargin{0.5in}" +
newline +
"\\setlength\\textwidth{6in}" + newline +
"\\setlength\\textheight{8.5in}" + newline
+
newline +
"\\begin{document}" + newline +
"\\pagestyle{plain}" + newline +
"\\pagenumbering{roman}" + newline +
newline +
"\\nocite{*}" + newline +
"\\begin{titlepage}" + newline +
"\\center{" + newline +

```



```

" \textsc{" + newline +
" University of the Philippines
Manila" + newline +
" \\\\College of Arts and Sciences"
+ newline +
" \\\\Department of Physical
Sciences and Mathematics" + newline +
"}" + newline +
" \vspace{4cm}" + newline +
" \\\\ {\huge " + title + "}" + newline +
" \vspace{4cm}" + newline +
" {\small" + newline +
" \\\\A Special Problem in Partial
Fulfillment" + newline +
" \\\\Of the Requirements for the
Degree" + newline +
" \\\\Bachelor Of Science in
Computer Science " + newline +
"}" + newline +
" \vspace{4cm}" + newline +
" \\\\ {\Large " + author + "}" + newline +
" \\\\ {\small " + studNum + "}" + newline
+
" \vspace{2cm}" + newline +
" \\\\ " + month + " " + year + newline +
"}" + newline +
newline +
" \end{titlepage}" + newline +
newline +
" \begin{center}" + newline +
" \addcontentsline{toc}{chapter}
{Acceptance Sheet}" + newline +
" \textbf{ACCEPTANCE SHEET}" + newline
+
" \end{center}" + newline +
" \bigskip" + newline +
newline +
newline +
"The special problem entitled ``" + title + ""
prepared and submitted by " + author + " " +
" in partial fulfillment of the requirements
for the degree of Bachelor of Science in " +
" Computer Science has been examined and
is recommended for acceptance." + newline +
" \begin{multicols}{2}" + newline +
" \begin{center}" + newline +
" \begin{verbatim}" + newline +
newline +
newline +
newline +
newline +
newline +
" \end{verbatim}" + newline +
" -----\\\\" + newline +
" " + adviser + "\\\" + newline +
" \footnotesize Adviser\\\\" + newline +
" \end{center}" + newline +
" \end{multicols}" + newline +
" \medskip" + newline +
newline +
" \begin{multicols}{2}" + newline +
" \begin{center}" + newline +
" \textsc{EXAMINERS}" + newline +
" \end{center}" + newline +
newline +
" \begin{enumerate}" + newline;
for (String p: panelists) {
tex += " \item " + p + newline;

```

```

}
tex += " \end{enumerate}" + newline +
newline +
" \begin{multicols}{2}" + newline +
" \begin{center}" + newline +
" \textsc{APPROVED}\\\\" + newline;
for (String p: panelists) {
tex += " \bigskip" + newline +
" -----\\\\" + newline;
}
tex += " \end{center}" + newline +
newline +
" \begin{center}" + newline +
" \textsc{DISAPPROVED}\\\\" +
newline;
for (String p: panelists) {
tex += " \bigskip" + newline +
" -----\\\\" +
newline;
}
tex += " \end{center}" + newline +
" \end{multicols}" + newline +
" \end{multicols}" + newline +
newline +
" \begin{multicols}{2}" + newline +
" \begin{center}" + newline +
" \begin{verbatim}" + newline +
newline +
newline +
newline +
newline +
" \end{verbatim}" + newline +
" -----\\\\" + newline +
" DATE" + newline +
" \end{center}" + newline +
" \end{multicols}" + newline +
newline +
" Accepted and approved as partial
fulfillment of the requirements for the degree of "
+
" Bachelor of Science in Computer Science."
+ newline +
newline +
" \bigskip" + newline +
newline +
" \bigskip" + newline +
newline +
" \begin{multicols}{2}" + newline +
" \begin{center}" + newline +
" -----\\\\" +
newline +
" " + unitHead + "\\\" + newline +
" {\footnotesize Unit Head\\\\" +
newline +
" \medskip" + newline +
" Mathematical and Computing Science
Unit\\\\" + newline +
" Department of Physical Sciences
and Mathematics}\\\\" + newline +
" -----\\\\" +
newline +
" " + deptChair + "\\\" + newline +
" {\footnotesize Chair\\\\" + newline +
" \medskip" + newline +
" Department of Physical Sciences
and Mathematics}" + newline +
" \end{center}" + newline +
" \end{multicols}" + newline +

```

```

        "\\bigskip" + newline +
        newline +
        "\\begin{center}" + newline +
        " -----\\\\" +
newline +
        " " + dean + "\\\\" + newline +
        "  {\fontsize Dean\\\\" + newline +
        "  \\medskip" + newline +
        "  College of Arts and Sciences}" +
newline +
        "\\end{center}" + newline;
    }

    void endDoc() {
        tex += newline + "\\end{document}";
    }

    String getText() {
        return tex;
    }

    private String cleanText(String text) {
        text = text.replaceAll("\\s*<input
type=\\\"hidden\\\">\\s*", "");

        if (text.contains("&#8226;")) {
            text = text.replaceFirst("&#8226;",
"\\begin{itemize} \\\\item ");
            int pos = 0;
            while (text.contains("&#8226;")) {
                pos = text.indexOf("&#8226;");
                text = text.replaceFirst("&#8226;", "\\item
");
            }
            text = text.substring(pos).replaceFirst("<br>",
"\\end{itemize} \\\\item ");
        }
        text = text.replaceAll("&#8217;", "");
        text = text.replaceAll("&#8211;", "-");
        text = text.replaceAll("&#8212;", "-");
        text = text.replaceAll("&#8230;", "...");
        text = text.replaceAll("&#8220;", "");
        text = text.replaceAll("&#8221;", "");
        text = text.replaceAll("<i>", "\\itit{");
        text = text.replaceAll("</i>", "");
        text = text.replaceAll("<ul>",
"\\begin{itemize}");
        text = text.replaceAll("<ol>",
"\\begin{enumerate}");
        text = text.replaceAll("</li>", "");
        text = text.replaceAll("<li>", "\\item ");
        text = text.replaceAll("</ul>",
"\\end{itemize}");
        text = text.replaceAll("</ol>",
"\\end{enumerate}");

        text = text.replaceAll("<br>", "\\n\\\\item ");
        String text1 = "";
        String text2 = "";
        String img = "";
        String texImg = "";
        String path = "";
        String imgWidthType = "";
        String imgWidth = "";
        String caption = "";
        while (text.contains("<img>")) {
            text1 = text.substring(0,
text.indexOf("<img>"));
            text2 = text.substring(text.indexOf("<img>"));
            img = text2.substring(text2.indexOf("<img>"),
text2.indexOf(">"));
            path = img.substring(img.indexOf("src=")+13);
            if (path.contains("\\") {
                String tempPath = path;
                path = "";
                while (tempPath.contains("\\") {
                    path += tempPath.substring(0,
tempPath.indexOf("\\")+1);
                    tempPath =
tempPath.substring(tempPath.indexOf("\\")+1);
                }
                path += tempPath;
            }
            caption =
img.substring(img.indexOf("caption=")+9,
img.length());
            caption = caption.substring(0,
caption.indexOf("\\"));
            if (path.contains("equation")) {
                text = text1 + caption +
text2.substring(text2.indexOf(">")+1,
text2.length());
            } else if (img.contains("type=\\\"symbol\\\"")) {
                text = text1 + "\\(" + caption + "\\)" +
text2.substring(text2.indexOf(">")+1,
text2.length());
            } else {
                imgWidthType =
img.substring(img.indexOf("realwidth=")+11,
img.length());
                imgWidthType = imgWidthType.substring(0,
imgWidthType.indexOf("\\"));
                if (imgWidthType.charAt(0) == 'i')
                    imgWidth = "[scale=" +
imgWidthType.substring(1, imgWidthType.length())
+ "]" ;
                else
                    imgWidth = "[width=" +
imgWidthType.substring(1, imgWidthType.length())
+ "\\textwidth]";
                if (text2.indexOf(">")+1 != text2.length())
                    text2 = text2.substring(text2.indexOf(">")+1,
text2.length());
            }
            else
                text2 = "";
            if (caption.isEmpty())
                texImg = "\\begin{figure}[htbp]" + newline
+
                "      \\centering" + newline +
                "          \\includegraphics" +
imgWidth + "{" + path + "}" + newline +
                "\\end{figure}" + newline;
            else
                texImg = "\\begin{figure}[htbp]" + newline
+
                "      \\centering" + newline +
                "          \\includegraphics" +
imgWidth + "{" + path + "}" + newline +
                "          \\caption{" + caption + "}" +
newline +
                "\\end{figure}" + newline;
            text = text1 + newline + texImg + text2;
        }
        return text;
    }

```

```

void setChapter(String chapTitle, String text) {
    text = cleanText(text);
    if (chapTitle.equals("Abstract")) {
        tex += newline +
            "\\newpage" + newline +
            "\\begin{center}" + newline +
            "\\addcontentsline{toc}{chapter}
{Abstract}" + newline +
            "\\textbf{ABSTRACT}" + newline +
            "\\end{center}" + newline +
            "\\bigskip" + newline +
            newline +
            text.substring(text.indexOf("<body>")+6,
text.indexOf("</body>")) + "|||||" +
            newline +
            "\\textbf{Keywords:} \\textit{" +
this.keywords + "}" +
            newline + newline +
            "\\tableofcontents" + newline +
            "\\newpage" + newline +
            newline;
    } else {
        if (chapTitle.equals("Appendix"))|
chapTitle.equals("Acknowledgement")) {
            tex += newline +
                "\\addcontentsline{toc}{chapter}" +
chapTitle + "}" + newline +
                "\\chapter*" + chapTitle + "}" +
newline;
        } else {
            tex += newline +
                "\\chapter{" + chapTitle + "}" + newline;
        }
        if (chapTitle.equals("Introduction")) {
            tex += newline +
                "\\pagenumbering{arabic}" + newline
+ newline;
        }
        tex += text.substring(text.indexOf("<body>")
+7, text.indexOf("</body>"));
    }
}

void setBib(ArrayList<Citation> citations) {
    tex += newline +
        "\\addcontentsline{toc}{chapter}
{Bibliography}" + newline +
        "\\bibliographystyle{unsrt}" + newline +
        "\\begin{thebibliography}{99}" + newline;
    for (Citation c: citations)
        tex += " \\bibitem{" + c.getCitationCnt() +
"} " + c.getCitation() + newline;

    tex += "\\end{thebibliography}" + newline +
        newline;
}

void setSubSection(ArrayList<String>
subsecTitle, ArrayList<JEditorPane> subTextPane)
{
    String text = "";
    for (int i = 0; i < subTextPane.size(); i++) {
        text = subTextPane.get(i).getText();
        text = cleanText(text);
        text = text.substring(text.indexOf("<body>")
+6, text.indexOf("</body>"));
        tex += newline +

```

```

        "\\section{" + subsecTitle.get(i) + "}" +
newline +
        "\\setlength\\parskip{.15in}" + newline +
text;
    }
}

void setTitlePage(String titleValue, String
authorValue, String studNumValue, String
monthValue, String yearValue) {
    if (!titleValue.equals(""))
        this.title = titleValue;
    if (!authorValue.equals(""))
        this.author = authorValue;
    if (!studNumValue.equals(""))
        this.studNum = studNumValue;
    if (!monthValue.equals(""))
        this.month = monthValue;
    if (!yearValue.equals(""))
        this.year = yearValue;
}

void setAcceptanceSheet(String adviserVal,
String unitHeadVal, String deptChairVal,
String deanVal, String asDateVal,
ArrayList<String> panelists) {
    if (!adviserVal.equals(""))
        this.adviser = adviserVal;
    if (!unitHeadVal.equals(""))
        this.unitHead = unitHeadVal;
    if (!deptChairVal.equals(""))
        this.deptChair = deptChairVal;
    if (!deanVal.equals(""))
        this.dean = deanVal;
    if (!asDateVal.equals(""))
        this.date = asDateVal;

    this.panelists = panelists;
}

void setKeywords(String keywords) {
    if (!keywords.equals(""))
        this.keywords = keywords;
}

private String[] getEnvironmentVariables() {
    final Map<String, String> env =
System.getenv();
    final String[] result = new String[env.size()];
    final StringBuilder buf = new StringBuilder(100);
    int i = 0;
    for (Entry<String, String> e : env.entrySet()) {
        buf.setLength(0);
        result[i++] =
buf.append(e.getKey()).append('=').append(e.getV
alue()).toString();
    }
    return result;
}

public void run(File file, int viewFlag) {
    String fname = file.getName().replace(".tex",
""");
    File path = new
File(file.getPath().replace(file.getName(), ""));
    final Runtime r = Runtime.getRuntime();
    Process p;
    try {

```

```

        p = r.exec("cmd /C start \"%\" \"C:\\Program
Files\\MiKTeX 2.5\\miktex\\bin\\pdflatex.exe\"
-interaction=nonstopmode \" + fname + \".tex\",
getEnvironmentVariables(), path);
        if ( p.waitFor() != 0 ) {
            JOptionPane.showMessageDialog(null, "Error
encountered in LaTeX compiler", "Error!",
JOptionPane.ERROR_MESSAGE);
            return;
        }
        while (!(new File(path + "\\\" + fname +
\".toc\").exists()) { }
        while (!(new File(path + "\\\" + fname +
\".log\").exists()) { }
        while (!(new File(path + "\\\" + fname +
\".pdf\").exists()) { }
        while (!(new File(path + "\\\" + fname +
\".log\").exists()) { }
        while (!(new File(path + "\\\" + fname +
\".toc\").exists()) { }
        while (!(new File(path + "\\\" + fname +
\".aux\").exists()) { }

        deleteFiles(path, fname);
        if(viewFlag == 1)
        {
            p = r.exec("cmd /C start
C:\\Progra~1\\Ghostgum\\gsvie\\gsvie32.exe \"
+ fname + \".pdf\", getEnvironmentVariables(),
path);
            if ( p.waitFor() != 0 ) {
                JOptionPane.showMessageDialog(null,
\"Error encountered while trying to generate
preview.\", \"Error!\", JOptionPane.ERROR_MESSAGE);
                return;
            }
        }
        deleteFiles(path, fname);
    } catch (Exception err) {
        err.printStackTrace();
    }
}

void convertToPDF(File dviFile) {
    String fname = dviFile.getName().replace(\".dvi\",
\"");
    File path = new
File(dviFile.getPath().replace(dviFile.getName(),
\""));
    final Runtime r = Runtime.getRuntime();
    Process p;
    try {
        p = r.exec("cmd /C start
C:\\Progra~1\\MiKTeX~1.7\\miktex\\bin\\dvi2pdf.exe \"
+ fname + \".dvi\", getEnvironmentVariables(),
path);
        if ( p.waitFor() != 0 ) {
            JOptionPane.showMessageDialog(null, \"Error
encountered in creating PDF file.\", \"Error!\",
JOptionPane.ERROR_MESSAGE);
            return;
        }
        JOptionPane.showMessageDialog(null, \"PDF File
created in \" + path + "\\\" + fname + \".pdf\",
\"Converted!\", JOptionPane.PLAIN_MESSAGE);
        deleteFiles(path, fname);
        File f = new File(path + "\\\" + fname + \".dvi\");
        f.delete();
    } catch (Exception err) {

```

```

        err.printStackTrace();
    }
}

private void deleteFiles(File path, String fname) {
    File f = new File(path + "\\\" + fname + \".aux\");
    f.delete();
    f = new File(path + "\\\" + fname + \".log\");
    while (f.exists())
        f.delete();
    f = new File(path + "\\\" + fname + \".log\");
    f.delete();
}

}

/** createSymbolButtons.java */

package upmguilte;

import javax.swing.BorderFactory;
import javax.swing.ImageIcon;
import javax.swing.JButton;

class createSymbolButtons {
    JButton setButtonProperties(JButton
symbolsButton, String path) {
        symbolsButton.setBackground(new
java.awt.Color(255, 255, 255));
        symbolsButton.setIcon(new ImageIcon(path));
        symbolsButton.setBorder(BorderFactory.createE
tchedBorder());
        symbolsButton.setContentAreaFilled(false);
        symbolsButton.setFocusPainted(false);
        symbolsButton.setPreferredSize(new
java.awt.Dimension(30, 30));
        return symbolsButton;
    }
}

/** createSymbols.java */

package upmguilte;

import java.awt.Dimension;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.GroupLayout;
import javax.swing.JButton;
import javax.swing.JPanel;
import javax.swing.LayoutStyle;
import javax.swing.border.BevelBorder;
import javax.swing.border.MatteBorder;
import javax.swing.border.SoftBevelBorder;

import
org.netbeans.lib.awtextra.AbsoluteConstraints;
import org.netbeans.lib.awtextra.AbsoluteLayout;

public abstract class createSymbols{

    public createSymbols() { }

    public void initSymbols(JPanel rightSidePanel) {
        initComponents(rightSidePanel);
    }

    private void initComponents(JPanel
rightSidePanel) {

```

```

int xCoor, yCoor;
greekCharPanel = new JPanel();
binOpPanel = new JPanel();
opAndQuanPanel = new JPanel();
relSymPanel = new JPanel();
greekCharButton = new JButton();
binOpButton = new JButton();
opAndQuanButton = new JButton();
relSymButton = new JButton();
GCButton = new JButton[45];
BOButton = new JButton[26];
OAQButton = new JButton[7];
RSButton = new JButton[26];

newSymbolsPanel(greekCharPanel);
newSymbolsButton(greekCharButton, "Greek
and Special Characters");

greekCharButton.addActionListener(new
ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        greekCharButtonActionPerformed(evt);
    }
});
greekCharPanel.add(greekCharButton, new
AbsoluteConstraints(0, 0, 260, 25));

createSymbolButtons csb = new
createSymbolButtons();
for (int i=0; i<45; i++) {
    GCButton[i] = new JButton();
    GCButton[i] =
csb.setButtonProperties(GCButton[i],
getGCPath(GCName[i]));
    xCoor = 10 + ((i*30) % 240);
    yCoor = 30 + ((int) (Math.floor((i)/8)) * 30);
    greekCharPanel.add(GCButton[i], new
AbsoluteConstraints(xCoor, yCoor, -1, -1));
    GCButton[i].setName(GCName[i]);
    final String path = getGCPath(GCName[i]);
    final String name = GCName[i];
    GCButton[i].addActionListener(new
ActionListener() {
        public void actionPerformed(ActionEvent evt)
    {
        SymbolsButtonActionPerformed(evt, name,
path);
        }
    });
}

newSymbolsPanel(binOpPanel);
newSymbolsButton(binOpButton, "Binary
Operations");

binOpButton.addActionListener(new
ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        binOpButtonActionPerformed(evt);
    }
});
binOpPanel.add(binOpButton, new
AbsoluteConstraints(0, 0, 260, 25));

for (int i=0; i<26; i++) {
    BOButton[i] = new JButton();
    BOButton[i] =
csb.setButtonProperties(BOButton[i],
getBOPath(BOName[i]));

```

```

    xCoor = 10 + ((i*30) % 240);
    yCoor = 30 + ((int) (Math.floor((i)/8)) * 30);
    binOpPanel.add(BOButton[i], new
AbsoluteConstraints(xCoor, yCoor, -1, -1));
    BOButton[i].setName(BOName[i]);
    final String path = getBOPath(BOName[i]);
    final String name = BOName[i];
    BOButton[i].addActionListener(new
ActionListener() {
        public void actionPerformed(ActionEvent evt)
    {
        SymbolsButtonActionPerformed(evt, name,
path);
        }
    });
}

newSymbolsPanel(opAndQuanPanel);
newSymbolsButton(opAndQuanButton,
"Operators and Quantifiers");

opAndQuanButton.addActionListener(new
ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        opAndQuanButtonActionPerformed(evt);
    }
});
opAndQuanPanel.add(opAndQuanButton, new
AbsoluteConstraints(0, 0, 260, 25));

for (int i=0; i<7; i++) {
    OAQButton[i] = new JButton();
    OAQButton[i] =
csb.setButtonProperties(OAQButton[i],
getOAQPath(OAQName[i]));
    xCoor = 10 + ((i*30) % 240);
    yCoor = 30 + ((int) (Math.floor((i)/8)) * 30);
    opAndQuanPanel.add(OAQButton[i], new
AbsoluteConstraints(xCoor, yCoor, -1, -1));
    OAQButton[i].setName(OAQName[i]);
    final String path = getOAQPath(OAQName[i]);
    final String name = OAQName[i];
    OAQButton[i].addActionListener(new
ActionListener() {
        public void actionPerformed(ActionEvent evt)
    {
        SymbolsButtonActionPerformed(evt, name,
path);
        }
    });
}

newSymbolsPanel(relSymPanel);
newSymbolsButton(relSymButton, "Relation
Symbols");

relSymButton.addActionListener(new
ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        relSymButtonActionPerformed(evt);
    }
});
relSymPanel.add(relSymButton, new
AbsoluteConstraints(0, 0, 260, 25));

for (int i=0; i<26; i++) {
    RSButton[i] = new JButton();

```

```

        RSBUTTON[i] =
        csb.setButtonProperties(RSBUTTON[i],
        getRSPATH(RSNAME[i]));
        xCoor = 10 + ((i*30) % 240);
        yCoor = 30 + ((int) (Math.floor((i)/8)) * 30);
        relSymPanel.add(RSBUTTON[i], new
        AbsoluteConstraints(xCoor, yCoor, -1, -1));
        RSBUTTON[i].setName(RSNAME[i]);
        final String path = getRSPATH(RSNAME[i]);
        final String name = RSNAME[i];
        RSBUTTON[i].addActionListener(new
        ActionListener() {
            public void actionPerformed(ActionEvent evt)
            {
                SymbolsButtonActionPerformed(evt, name,
                path);
            }
        });
    }

    layout = new GroupLayout(rightSidePanel);
    rightSidePanel.setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(GroupLayout.Align
        ment.LEADING)
        .addComponent(greekCharPanel,
        GroupLayout.DEFAULT_SIZE, 260,
        Short.MAX_VALUE)
        .addComponent(binOpPanel,
        GroupLayout.DEFAULT_SIZE, 260,
        Short.MAX_VALUE)
        .addComponent(opAndQuanPanel,
        GroupLayout.DEFAULT_SIZE, 260,
        Short.MAX_VALUE)
        .addComponent(relSymPanel,
        GroupLayout.PREFERRED_SIZE, 260,
        GroupLayout.PREFERRED_SIZE)
        );
    layout.setVerticalGroup(
        layout.createParallelGroup(GroupLayout.Align
        ment.LEADING)
        .addGroup(layout.createSequentialGroup()
        .addComponent(greekCharPanel,
        GroupLayout.PREFERRED_SIZE, 220,
        GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(LayoutStyle.ComponentPla
        cement.RELATED)
        .addComponent(binOpPanel,
        GroupLayout.PREFERRED_SIZE, 160,
        GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(LayoutStyle.ComponentPla
        cement.RELATED)
        .addComponent(opAndQuanPanel,
        GroupLayout.PREFERRED_SIZE, 70,
        GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(LayoutStyle.ComponentPla
        cement.RELATED)
        .addComponent(relSymPanel,
        GroupLayout.PREFERRED_SIZE, 160,
        GroupLayout.PREFERRED_SIZE)
        );
    }

    private void
    greekCharButtonActionPerformed(ActionEvent evt)
    {
        int height = greekCharPanel.getHeight();
        if (height == 220)

```

```

        height = 25;
        else
        height = 220;

        layout.setVerticalGroup(
        layout.createParallelGroup(GroupLayout.Align
        ment.LEADING)
        .addGroup(layout.createSequentialGroup()
        .addComponent(greekCharPanel,
        GroupLayout.PREFERRED_SIZE, height,
        GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(LayoutStyle.ComponentPla
        cement.RELATED)
        .addComponent(binOpPanel,
        GroupLayout.PREFERRED_SIZE,
        binOpPanel.getHeight(),
        GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(LayoutStyle.ComponentPla
        cement.RELATED)
        .addComponent(opAndQuanPanel,
        GroupLayout.PREFERRED_SIZE,
        opAndQuanPanel.getHeight(),
        GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(LayoutStyle.ComponentPla
        cement.RELATED)
        .addComponent(relSymPanel,
        GroupLayout.PREFERRED_SIZE,
        relSymPanel.getHeight(),
        GroupLayout.PREFERRED_SIZE)
        );
    }

    private void
    binOpButtonActionPerformed(ActionEvent evt) {
        int height = binOpPanel.getHeight();
        if (height == 160)
            height = 25;
        else
            height = 160;

        layout.setVerticalGroup(
        layout.createParallelGroup(GroupLayout.Align
        ment.LEADING)
        .addGroup(layout.createSequentialGroup()
        .addComponent(greekCharPanel,
        GroupLayout.PREFERRED_SIZE,
        greekCharPanel.getHeight(),
        GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(LayoutStyle.ComponentPla
        cement.RELATED)
        .addComponent(binOpPanel,
        GroupLayout.PREFERRED_SIZE, height,
        GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(LayoutStyle.ComponentPla
        cement.RELATED)
        .addComponent(opAndQuanPanel,
        GroupLayout.PREFERRED_SIZE,
        opAndQuanPanel.getHeight(),
        GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(LayoutStyle.ComponentPla
        cement.RELATED)
        .addComponent(relSymPanel,
        GroupLayout.PREFERRED_SIZE,
        relSymPanel.getHeight(),
        GroupLayout.PREFERRED_SIZE)
        );
    }

```

```

private void newSymbolsButton(JButton button,
String title) {
    button.setBackground(new java.awt.Color(71,
67, 49));
    button.setForeground(new java.awt.Color(145,
134, 86));
    button.setText(title);
    button.setBorder(new
SoftBevelBorder(BevelBorder.RAISED));
    button.setContentAreaFilled(false);
    button.setFocusPainted(false);
    button.setOpaque(true);
    button.setMargin(new java.awt.Insets(3, 14, 3,
14));
}

```

```

private void newSymbolsPanel(JPanel panel) {
    panel.setBackground(new java.awt.Color(255,
255, 255));
    panel.setMinimumSize(new Dimension(260,
180));
    panel.setPreferredSize(new Dimension(260,
180));
    panel.setLayout(new AbsoluteLayout());
    panel.setBorder(new MatteBorder(null));
}

```

```

private void
opAndQuanButtonActionPerformed(ActionEvent
evt) {
    int height = opAndQuanPanel.getHeight();
    if (height == 70)
        height = 25;
    else
        height = 70;

```

```

    layout.setVerticalGroup(
    layout.createParallelGroup(GroupLayout.Align
ment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(greekCharPanel,
GroupLayout.PREFERRED_SIZE,
greekCharPanel.getHeight(),
GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(LayoutStyle.ComponentPla
cement.RELATED)
            .addComponent(binOpPanel,
GroupLayout.PREFERRED_SIZE,
binOpPanel.getHeight(),
GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(LayoutStyle.ComponentPla
cement.RELATED)
            .addComponent(opAndQuanPanel,
GroupLayout.PREFERRED_SIZE, height,
GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(LayoutStyle.ComponentPla
cement.RELATED)
            .addComponent(relSymPanel,
GroupLayout.PREFERRED_SIZE,
relSymPanel.getHeight(),
GroupLayout.PREFERRED_SIZE)
        );
}

```

```

private void
relSymButtonActionPerformed(ActionEvent evt) {
    int height = relSymPanel.getHeight();
    if (height == 160)
        height = 25;

```

```

else
    height = 160;

    layout.setVerticalGroup(
    layout.createParallelGroup(GroupLayout.Align
ment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addComponent(greekCharPanel,
GroupLayout.PREFERRED_SIZE,
greekCharPanel.getHeight(),
GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(LayoutStyle.ComponentPla
cement.RELATED)
            .addComponent(binOpPanel,
GroupLayout.PREFERRED_SIZE,
binOpPanel.getHeight(),
GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(LayoutStyle.ComponentPla
cement.RELATED)
            .addComponent(opAndQuanPanel,
GroupLayout.PREFERRED_SIZE,
opAndQuanPanel.getHeight(),
GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(LayoutStyle.ComponentPla
cement.RELATED)
            .addComponent(relSymPanel,
GroupLayout.PREFERRED_SIZE, height,
GroupLayout.PREFERRED_SIZE)
        );
}

```

```

public String getGCPATH(String GCName) {
    String path = "C:\\LaTeX Symbols\\Greek and
Special Chars\\";
    if (GCName.length() == 1)
        return path + GCName + ".jpg";
    else if (GCName.equals("\\Re") ||
GCName.equals("\\Im"))
        return path + GCName.substring(1) + ".jpg";
    else if
(Character.isUpperCase(GCName.charAt(1)))
        return path + "UP-" + GCName.substring(1) +
".jpg";
    else
        return path + GCName.substring(1) + ".jpg";
}

```

```

public String getBOPATH(String BOName) {
    String path = "C:\\LaTeX Symbols\\Binary
Operations\\";
    return path + BOName.substring(1) + ".jpg";
}

```

```

public String getOAQPATH(String OAQName) {
    String path = "C:\\LaTeX Symbols\\Operators
and Quantifiers\\";
    return path + OAQName.substring(1) + ".jpg";
}

```

```

public String getRSPATH(String RSName) {
    String path = "C:\\LaTeX Symbols\\Relation
Symbols\\";
    return path + RSName.substring(1) + ".jpg";
}

```

```

public abstract void
SymbolsButtonActionPerformed(ActionEvent evt,
String bName, String bPath);

```

```

// Variables declaration - do not modify
private JButton greekCharButton;
private JPanel greekCharPanel;
private JButton GCBUTTON[];
private JButton binOpButton;
private JPanel binOpPanel;
private JButton BOBUTTON[];
private JButton opAndQuanButton;
private JPanel opAndQuanPanel;
private JButton OAQBUTTON[];
private JButton relSymButton;
private JPanel relSymPanel;
private JButton RSBUTTON[];
GroupLayout layout;
String GCName[] = new String[]
{"\alpha", "\beta", "\gamma", "\delta", "\epsilon", "\varepsilon", "\zeta", "\eta", "\theta", "\vartheta", "\iota", "\kappa", "\lambda", "\mu", "\nu", "\xi", "\omicron", "\pi", "\varpi", "\rho", "\varrho", "\sigma", "\varsigma", "\tau", "\upsilon", "\phi", "\varphi", "\chi", "\psi", "\omega", "\Gamma", "\Delta", "\Theta", "\aleph", "\Lambda", "\Xi", "\Pi", "\ell", "\Sigma", "\Upsilon", "\Phi", "\Re", "\Psi", "\Omega", "\Im"};
String BONAME[] = new String[]
{"\pm", "\mp", "\wedge", "\vee", "\cap", "\cup", "\sqcap", "\sqcup", "\uplus", "\bigcap", "\bigcup", "\bigwedge", "\bigvee", "\bigodot", "\bigotimes", "\bigoplus", "\bigsqcup", "\biguplus", "\loplus", "\cdot", "\otimes", "\oslash", "\ominus", "\times", "\div", "\setminus"};
String OAQNAME[] = new String[]
{"\forall", "\exists", "\neg", "\nabla", "\partial", "\uurd", "\wp"};
String RSNAME[] = new String[]
{"\leq", "\prec", "\preceq", "\ll", "\subset", "\subseteq", "\sqsubseteq", "\neq", "\equiv", "\geq", "\succ", "\succeq", "\gg", "\supset", "\supseteq", "\sqsupseteq", "\in", "\ni", "\doteq", "\models", "\propto", "\asymp", "\sim", "\approx", "\simeq", "\cong"};
// End of variables declaration
}

/** createTree.java */

package upmguilte;

import java.awt.Color;
import java.awt.Font;
import java.util.ArrayList;

import javax.swing.BorderFactory;
import javax.swing.GroupLayout;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTree;
import javax.swing.GroupLayout.Alignment;
import javax.swing.LayoutStyle.ComponentPlacement;
import javax.swing.border.BevelBorder;
import javax.swing.event.TreeSelectionEvent;
import javax.swing.event.TreeSelectionListener;
import javax.swing.tree.DefaultMutableTreeNode;
import javax.swing.tree.DefaultTreeCellRenderer;
import javax.swing.tree.DefaultTreeModel;
import javax.swing.tree.TreeSelectionModel;

public abstract class createTree implements
TreeSelectionListener {
protected JTree navigatorTree;
private JScrollPane navigatorScrollPane;
private JLabel navigatorLabel;
private JButton addSubsectionsButton;
private JButton remSubsectionsButton;
DefaultMutableTreeNode chapNode[];

public createTree(JPanel leftSidePanel, int
noOfChapters, String chapTitle[]) {
chapNode = new
DefaultMutableTreeNode[noOfChapters];
DefaultMutableTreeNode top = new
DefaultMutableTreeNode();
createNodes(top, chapTitle, noOfChapters);

navigatorTree = new JTree(top);
navigatorTree.setRootVisible(false);
navigatorTree.getSelectionModel().setSelection
Mode(TreeSelectionMode.SINGLE_TREE_SELECTIO
N);

ImageIcon leafIcon = new
ImageIcon("C:\\Users\\gomez\\Documents\\NetBea
nsProjects\\SP\\src\\images\\Binary
Operations\\ominus.jpg");
if (leafIcon != null) {
DefaultTreeCellRenderer renderer = new
DefaultTreeCellRenderer();
renderer.setTextSelectionColor(new
Color(145, 134, 86));
renderer.setBorderSelectionColor(Color.white)
;
renderer.setBackgroundSelectionColor(Color.
white);
renderer.setFont(new Font(Font.SANS_SERIF,
Font.PLAIN, 11));
renderer.setLeafIcon(leafIcon);
navigatorTree.setCellRenderer(renderer);
}
initComponents(leftSidePanel);

//Listen for when the selection changes.
navigatorTree.addTreeSelectionListener(this);
}

private void initComponents(JPanel leftSidePanel)
{
navigatorScrollPane = new JScrollPane();
navigatorLabel = new JLabel();
addSubsectionsButton = new JButton();
remSubsectionsButton = new JButton();

leftSidePanel.setBackground(new
java.awt.Color(71, 67, 49));
leftSidePanel.setBorder(BorderFactory.createBe
velBorder(BevelBorder.RAISED));

navigatorScrollPane.setViewportView(navigatorT
ree);

navigatorLabel.setForeground(new
java.awt.Color(145, 134, 86));
navigatorLabel.setText("Navigator");

addSubsectionsButton.setText("Add
Subsection");
}
}

```



```

        addSubsectionsButton.setBorder(new
javax.swing.border.SoftBevelBorder(BevelBorder.R
AISED));
        addSubsectionsButton.setBackground(new
java.awt.Color(71, 67, 49));
        addSubsectionsButton.setForeground(new
java.awt.Color(145, 134, 86));
        addSubsectionsButton.setFocusPainted(false);
        addSubsectionsButton.setOpaque(true);
        addSubsectionsButton.addMouseListener(new
java.awt.event.MouseAdapter() {
            public void
mouseClicked(java.awt.event.MouseEvent evt) {
                addSubsectionButtonMouseClicked(evt);
            }
        });
        remSubsectionsButton.setText("Remove
Subsection");
        remSubsectionsButton.setBorder(new
javax.swing.border.SoftBevelBorder(BevelBorder.R
AISED));
        remSubsectionsButton.setBackground(new
java.awt.Color(71, 67, 49));
        remSubsectionsButton.setForeground(new
java.awt.Color(145, 134, 86));
        remSubsectionsButton.setFocusPainted(false);
        remSubsectionsButton.setOpaque(true);
        remSubsectionsButton.addMouseListener(new
java.awt.event.MouseAdapter() {
            public void
mouseClicked(java.awt.event.MouseEvent evt) {
                remSubsectionButtonMouseClicked(evt);
            }
        });

        GroupLayout layout = new
GroupLayout(leftSidePanel);
        leftSidePanel.setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(Alignment.LEADING
)
                .addGroup(layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(navigatorLabel,
GroupLayout.PREFERRED_SIZE, 56,
GroupLayout.PREFERRED_SIZE)
                    .addGap(10, 10, 10)
                    .addComponent(navigatorScrollPane,
GroupLayout.DEFAULT_SIZE, 168,
Short.MAX_VALUE)
                    .addComponent(addSubsectionsButton,
GroupLayout.DEFAULT_SIZE, 168,
Short.MAX_VALUE)
                    .addComponent(remSubsectionsButton,
GroupLayout.DEFAULT_SIZE, 168,
Short.MAX_VALUE)
                );
        layout.setVerticalGroup(
            layout.createParallelGroup(Alignment.LEADING
)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(navigatorLabel,
GroupLayout.PREFERRED_SIZE, 14,
GroupLayout.PREFERRED_SIZE)
                    .addGap(10, 10, 10)
                    .addPreferredGap(ComponentPlacement.RELA
TED)
                    .addComponent(navigatorScrollPane,
GroupLayout.DEFAULT_SIZE, 84,
Short.MAX_VALUE)

```

```

                .addPreferredGap(ComponentPlacement.RELA
TED)
                .addComponent(addSubsectionsButton)
                .addComponent(remSubsectionsButton)
                .addGap(224, 224, 224)
            );
        }

        private void
createNodes(DefaultMutableTreeNode top, String[]
chapTitle, int noOfChapters) {
            chapNode[0] = new
DefaultMutableTreeNode(new
chapInfo(chapTitle[0], 0));
            top.add(chapNode[0]);
            for(int i=0; i<noOfChapters; i++) {
                chapNode[i] = new
DefaultMutableTreeNode(new
chapInfo(chapTitle[i+1], i+1));
                top.add(chapNode[i]);
            }
        }

        protected class chapInfo extends
DefaultMutableTreeNode {
            private static final long serialVersionUID = 1L;
            protected int index;
            private String title;

            public chapInfo(String chapTitle, int chapIndex)
        {
            index = chapIndex;
            title = chapTitle;
        }

            public String toString() {
                return title;
            }
        }

        public abstract void
valueChanged(TreeSelectionEvent e);
        public abstract void
addSubsectionButtonMouseClicked(java.awt.event.
MouseEvent evt);
        public abstract void
remSubsectionButtonMouseClicked(java.awt.event.
MouseEvent evt);

        void setTreeSubsections(int chapNum,
ArrayList<String> subsections) {
            DefaultMutableTreeNode subsec;
            for (int i=0; i<subsections.size(); i++) {
                subsec = new DefaultMutableTreeNode(new
chapInfo(subsections.get(i),
(chapNum)*100+i+1));
                chapNode[chapNum].add(subsec);
            }
            navigatorTree.expandRow(chapNum+1);
        }

        void addSubsecTree(int index, String subsection)
        {
            DefaultMutableTreeNode node =
(DefaultMutableTreeNode)navigatorTree.getLastSe
lectedPathComponent();
            addSubsecNode(index, subsection, node, false);
        }

```

```

void addSubsecNode(int index, String subsection,
DefaultMutableTreeNode node, boolean
nodeEmpty) {
    int chapNum = (index<100) ? index :
(int)Math.floor(index/100);
    if (nodeEmpty) {
        DefaultTreeModel model = (DefaultTreeModel)
navigatorTree.getModel();
        node = (DefaultMutableTreeNode)
model.getChild(model.getRoot(), chapNum+1);
    }
    if (node != null) {
        DefaultMutableTreeNode subsec;
        if (index<100) {
            subsec = new DefaultMutableTreeNode(new
chapInfo(subsection,
(chapNum)*100+node.getChildCount()+1));
            DefaultTreeModel model =
(DefaultTreeModel) navigatorTree.getModel();
            model.insertNodeInto(subsec, node,
node.getChildCount());
        } else {
            subsec = new DefaultMutableTreeNode(new
chapInfo(subsection, index+1));
            while (node.getNextSibling() != null) {
                node = node.getNextSibling();
                ((chapInfo)node.getUserObject()).index +=
1;
            }
            DefaultTreeModel model =
(DefaultTreeModel) navigatorTree.getModel();
            model.insertNodeInto(subsec,
((DefaultMutableTreeNode)node.getParent()),
index%100);
            navigatorTree.expandRow(chapNum+1);
            navigatorTree.updateUI();
        }
    }

void remTreeSubsections(int index) {
    DefaultMutableTreeNode node =
(DefaultMutableTreeNode)navigatorTree.getLastSe
lectedPathComponent();
    if (node != null) {
        DefaultMutableTreeNode parentNode =
(DefaultMutableTreeNode)node.getParent();
        while (node.getNextSibling() != null) {
            node = node.getNextSibling();
            ((chapInfo)node.getUserObject()).index -= 1;
        }
        parentNode.remove(index%100-1);
        navigatorTree.updateUI();
        navigatorTree.setSelectionRow((int)Math.floor(i
ndex/100)+1);
    }
}

void remChapSubsections(int chapNum) {
    DefaultTreeModel model = (DefaultTreeModel)
navigatorTree.getModel();
    DefaultMutableTreeNode node =
(DefaultMutableTreeNode)
model.getChild(model.getRoot(), chapNum+1);
    node.removeAllChildren();
    navigatorTree.updateUI();
}
}

```

```

/** editorComponents.java */

```

```

package upmguilte;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.InputEvent;
import java.awt.event.KeyEvent;
import java.io.File;
import java.util.ArrayList;

import javax.swing.BorderFactory;
import javax.swing.GroupLayout;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.KeyStroke;
import javax.swing.GroupLayout.Alignment;
import
javax.swing.LayoutStyle.ComponentPlacement;
import javax.swing.border.BevelBorder;
import javax.swing.undo.UndoManager;

public abstract class editorComponents extends
javax.swing.JFrame {
    public static File project;
    String filename;
    private JFileChooser fc;
    protected newProject np;
    protected createSymbols cs;
    protected createTree cnt;
    protected static UndoManager manager = new
UndoManager();
    readFile rf = new readFile();

    public editorComponents() {}

    public editorComponents(int flag) {
        setLocation(100, 30);
        filename = File.separator + "tmp";
        fc = new JFileChooser(new File(filename));

        initComponents();
        this.setVisible(true);
    }

    void setTreeSubsections(int chapNum,
ArrayList<String> subsections) {
        cnt.setTreeSubsections(chapNum, subsections);
    }

    void addTreeSubsections(int index, String
subsection) {
        cnt.addSubsecNode(index, subsection, null,
true);
    }

    void remTreeSubsections(int index) {
        cnt.remTreeSubsections(index);
    }

    void remChapSubsections(int chapNum) {
        cnt.remChapSubsections(chapNum);
    }

    private void initComponents() {
        framePanel = new JPanel();
    }
}

```

```

leftSidePanel = new JPanel();
mainPanel = new JPanel();
rightSidePanel = new JPanel();
toolBar = new javax.swing.JToolBar();
menuBar = new javax.swing.JMenuBar();
fileMenu = new javax.swing.JMenu();
newProjectCommand = new
javax.swing.JMenuItem();
openProjectCommand = new
javax.swing.JMenuItem();
saveCommand = new javax.swing.JMenuItem();
exitCommand = new javax.swing.JMenuItem();
editMenu = new javax.swing.JMenu();
undoCommand = new javax.swing.JMenuItem();
redoCommand = new javax.swing.JMenuItem();
jSeparator1 = new javax.swing.JSeparator();
cutCommand = new javax.swing.JMenuItem();
copyCommand = new javax.swing.JMenuItem();
pasteCommand = new javax.swing.JMenuItem();
jSeparator2 = new javax.swing.JSeparator();
selectAllCommand = new
javax.swing.JMenuItem();
insertMenu = new javax.swing.JMenu();
insertImageCommand = new
javax.swing.JMenuItem();
citationCommand = new
javax.swing.JMenuItem();
equationCommand = new
javax.swing.JMenuItem();
oListCommand = new javax.swing.JMenuItem();
uListCommand = new javax.swing.JMenuItem();
runMenu = new javax.swing.JMenu();
compileCommand = new
javax.swing.JMenuItem();
viewCommand = new javax.swing.JMenuItem();
helpMenu = new javax.swing.JMenu();
aboutCommand = new javax.swing.JMenuItem();
tutorialCommand = new
javax.swing.JMenuItem();

setDefaultCloseOperation(javax.swing.WindowC
onstants.EXIT_ON_CLOSE);
setMinimumSize(new java.awt.Dimension(1000,
710));

framePanel.setBackground(new
java.awt.Color(79, 76, 60));

leftSidePanel.setOpaque(false);

GroupLayout leftSidePanelLayout = new
GroupLayout(leftSidePanel);
leftSidePanel.setLayout(leftSidePanelLayout);
leftSidePanelLayout.setHorizontalGroup(
leftSidePanelLayout.createParallelGroup(Alignme
nt.LEADING)
.addGap(0, 206, Short.MAX_VALUE)
);
leftSidePanelLayout.setVerticalGroup(
leftSidePanelLayout.createParallelGroup(Align
ment.LEADING)
.addGap(0, 610, Short.MAX_VALUE)
);

mainPanel.setBackground(framePanel.getBackg
round());
mainPanel.setBorder(BorderFactory.createBevel
Border(BevelBorder.RAISED));

GroupLayout mainPanelLayout = new
GroupLayout(mainPanel);
mainPanel.setLayout(mainPanelLayout);
mainPanelLayout.setHorizontalGroup(
mainPanelLayout.createParallelGroup(Alignme
nt.LEADING)
.addGap(0, 610, Short.MAX_VALUE)
);
mainPanelLayout.setVerticalGroup(
mainPanelLayout.createParallelGroup(Alignme
nt.LEADING)
.addGap(0, 606, Short.MAX_VALUE)
);

rightSidePanel.setOpaque(false);

GroupLayout rightSidePanelLayout = new
GroupLayout(rightSidePanel);
rightSidePanel.setLayout(rightSidePanelLayout);
rightSidePanelLayout.setHorizontalGroup(
rightSidePanelLayout.createParallelGroup(Align
ment.LEADING)
.addGap(0, 158, Short.MAX_VALUE)
);
rightSidePanelLayout.setVerticalGroup(
rightSidePanelLayout.createParallelGroup(Align
ment.LEADING)
.addGap(0, 610, Short.MAX_VALUE)
);

toolBar.setBackground(framePanel.getBackgrou
nd());
toolBar.setBorder(BorderFactory.createEtchedBo
rder());
toolBar.setFloatable(false);
toolBar.setRollover(true);

GroupLayout framePanelLayout = new
GroupLayout(framePanel);
framePanel.setLayout(framePanelLayout);
framePanelLayout.setHorizontalGroup(
framePanelLayout.createParallelGroup(Alignme
nt.LEADING)
.addGroup(framePanelLayout.createSequential
Group()
.addGap(5, 5, 5)
.addComponent(leftSidePanel,
GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE)
.addPreferredGap(ComponentPlacement.RELA
TED)
.addComponent(mainPanel,
GroupLayout.DEFAULT_SIZE,
GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addPreferredGap(ComponentPlacement.RELA
TED)
.addComponent(rightSidePanel,
GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE)
.addGap(5, 5, 5))
.addComponent(toolBar,
GroupLayout.DEFAULT_SIZE, 1000,
Short.MAX_VALUE)
);
framePanelLayout.setVerticalGroup(
framePanelLayout.createParallelGroup(Alignme
nt.LEADING)

```

```

        .addGroup(Alignment.TRAILING,
framePanelLayout.createSequentialGroup()
        .addComponent(toolBar,
GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(ComponentPlacement.RELA
TED)
        .addGroup(framePanelLayout.createParallelGr
oup(Alignment.LEADING)
        .addComponent(rightSidePanel,
GroupLayout.DEFAULT_SIZE,
GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(leftSidePanel,
Alignment.TRAILING, GroupLayout.DEFAULT_SIZE,
GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(mainPanel,
Alignment.TRAILING, GroupLayout.DEFAULT_SIZE,
GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addGap(11, 11, 11)
    );

    menuBar.setBackground(new java.awt.Color(71,
67, 49));
    menuBar.setBorder(new
javax.swing.border.SoftBevelBorder(BevelBorder.R
AISED));

    fileMenu.setBackground(new java.awt.Color(71,
67, 49));
    fileMenu.setBorder(null);
    fileMenu.setForeground(new java.awt.Color(255,
255, 255));
    fileMenu.setText("File");
    fileMenu.setBorderPainted(true);
    fileMenu.setOpaque(true);

    newProjectCommand.setAccelerator(KeyStroke.
getKeyStroke(KeyEvent.VK_N,
InputEvent.CTRL_MASK));
    newProjectCommand.setBackground(new
java.awt.Color(57, 50, 21));
    newProjectCommand.setForeground(new
java.awt.Color(221, 208, 168));
    newProjectCommand.setText("New Project");
    newProjectCommand.addActionListener(new
ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            newProjectCommandActionPerformed(evt);
        }
    });
    fileMenu.add(newProjectCommand);

    openProjectCommand.setAccelerator(KeyStroke.
getKeyStroke(KeyEvent.VK_O,
InputEvent.CTRL_MASK));
    openProjectCommand.setBackground(newProjec
tCommand.getBackground());
    openProjectCommand.setForeground(newProjec
tCommand.getForeground());
    openProjectCommand.setText("Open Project");
    openProjectCommand.setBorder(null);
    openProjectCommand.setBorderPainted(true);
    openProjectCommand.setOpaque(true);
    openProjectCommand.addActionListener(new
ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            openProjectCommandActionPerformed(evt);
        }
    });

    saveCommand.setAccelerator(KeyStroke.getKey
Stroke(KeyEvent.VK_S, InputEvent.CTRL_MASK));
    saveCommand.setBackground(newProjectComm
and.getBackground());
    saveCommand.setForeground(newProjectComm
and.getForeground());
    saveCommand.setText("Save Project");
    saveCommand.setBorderPainted(true);
    saveCommand.setOpaque(true);
    saveCommand.addActionListener(new
ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            saveCommandActionPerformed(evt);
        }
    });
    fileMenu.add(saveCommand);

    exitCommand.setAccelerator(KeyStroke.getKeyS
troke(KeyEvent.VK_F4, InputEvent.ALT_MASK));
    exitCommand.setBackground(newProjectComm
and.getBackground());
    exitCommand.setForeground(newProjectComma
nd.getForeground());
    exitCommand.setText("Exit");
    exitCommand.setBorder(null);
    exitCommand.setBorderPainted(true);
    exitCommand.setOpaque(true);
    exitCommand.addActionListener(new
ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            exitCommandActionPerformed(evt);
        }
    });
    fileMenu.add(exitCommand);

    menuBar.add(fileMenu);

    editMenu.setBackground(fileMenu.getBackgroun
d());
    editMenu.setForeground(fileMenu.getForegroun
d());
    editMenu.setText("Edit");
    editMenu.setOpaque(true);

    undoCommand.setAccelerator(KeyStroke.getKey
Stroke(KeyEvent.VK_Z, InputEvent.CTRL_MASK));
    undoCommand.setBackground(newProjectComm
and.getBackground());
    undoCommand.setForeground(newProjectComm
and.getForeground());
    undoCommand.setText("Undo");
    editMenu.add(undoCommand);

    redoCommand.setAccelerator(KeyStroke.getKey
Stroke(KeyEvent.VK_Z, InputEvent.SHIFT_MASK |
InputEvent.CTRL_MASK));
    redoCommand.setBackground(newProjectComm
and.getBackground());
    redoCommand.setForeground(newProjectComm
and.getForeground());
    redoCommand.setText("Redo");
    editMenu.add(redoCommand);

    jSeparator1.setBackground(newProjectComman
d.getBackground());

```

```

jSeparator1.setForeground(newProjectCommand
.getForeground());
editMenu.add(jSeparator1);

cutCommand.setAccelerator(KeyStroke.getKeyS
troke(KeyEvent.VK_X, InputEvent.CTRL_MASK));
cutCommand.setBackground(newProjectComm
and.getBackground());
cutCommand.setForeground(newProjectComm
and.getForeground());
cutCommand.setText("Cut");
editMenu.add(cutCommand);

copyCommand.setAccelerator(KeyStroke.getKey
Stroke(KeyEvent.VK_C, InputEvent.CTRL_MASK));
copyCommand.setBackground(newProjectComm
and.getBackground());
copyCommand.setForeground(newProjectComm
and.getForeground());
copyCommand.setText("Copy");
editMenu.add(copyCommand);

pasteCommand.setAccelerator(KeyStroke.getKe
yStroke(KeyEvent.VK_V, InputEvent.CTRL_MASK));
pasteCommand.setBackground(newProjectCom
mand.getBackground());
pasteCommand.setForeground(newProjectCom
mand.getForeground());
pasteCommand.setText("Paste");
editMenu.add(pasteCommand);

jSeparator2.setBackground(newProjectComm
d.getBackground());
jSeparator2.setForeground(newProjectComm
and.getForeground());
editMenu.add(jSeparator2);

selectAllCommand.setAccelerator(KeyStroke.get
KeyStroke(KeyEvent.VK_A,
InputEvent.CTRL_MASK));
selectAllCommand.setBackground(newProjectCo
mmand.getBackground());
selectAllCommand.setForeground(newProjectCo
mmand.getForeground());
selectAllCommand.setText("Select All");
editMenu.add(selectAllCommand);

menuBar.add(editMenu);

insertMenu.setBackground(fileMenu.getBackgro
und());
insertMenu.setForeground(fileMenu.getForegrou
nd());
insertMenu.setText("Insert");

insertImageCommand.setBackground(newProjec
tCommand.getBackground());
insertImageCommand.setForeground(newProjec
tCommand.getForeground());
insertImageCommand.setText("Image");
insertImageCommand.addActionListener(new
ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        insertImageCommandActionPerformed(evt);
    }
});
insertMenu.add(insertImageCommand);

```

```

citationCommand.setBackground(newProjectCo
mmand.getBackground());
citationCommand.setForeground(newProjectCo
mmand.getForeground());
citationCommand.setText("Citation");
citationCommand.addActionListener(new
ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        citationCommandActionPerformed(evt);
    }
});
insertMenu.add(citationCommand);

equationCommand.setBackground(newProjectC
ommand.getBackground());
equationCommand.setForeground(newProjectCo
mmand.getForeground());
equationCommand.setText("Equation");
equationCommand.addActionListener(new
ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        equationCommandActionPerformed(evt);
    }
});
insertMenu.add(equationCommand);

oListCommand.setBackground(newProjectComm
and.getBackground());
oListCommand.setForeground(newProjectComm
and.getForeground());
oListCommand.setText("Ordered List");
oListCommand.addActionListener(new
ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        oListCommandActionPerformed(evt);
    }
});
insertMenu.add(oListCommand);

uListCommand.setBackground(newProjectComm
and.getBackground());
uListCommand.setForeground(newProjectComm
and.getForeground());
uListCommand.setText("Unordered List");
uListCommand.addActionListener(new
ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        uListCommandActionPerformed(evt);
    }
});
insertMenu.add(uListCommand);

menuBar.add(insertMenu);

runMenu.setBackground(fileMenu.getBackgroun
d());
runMenu.setForeground(fileMenu.getForegroun
d());
runMenu.setText("Run");

compileCommand.setAccelerator(KeyStroke.get
KeyStroke(KeyEvent.VK_F3, 0));
compileCommand.setBackground(newProjectCo
mmand.getBackground());
compileCommand.setForeground(newProjectCo
mmand.getForeground());
compileCommand.setText("Compile");
compileCommand.addActionListener(new
ActionListener() {

```

```

        public void actionPerformed(ActionEvent evt) {
            previewCommandActionPerformed(evt);
        }
    });
    runMenu.add(compileCommand);

    viewCommand.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_F4, 0));
    viewCommand.setBackground(newProjectCommand.getBackground());
    viewCommand.setForeground(newProjectCommand.getForeground());
    viewCommand.setText("Preview");
    viewCommand.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            finalizeCommandActionPerformed(evt);
        }
    });
    runMenu.add(viewCommand);

    menuBar.add(runMenu);

    helpMenu.setBackground(fileMenu.getBackground());
    helpMenu.setForeground(fileMenu.getForeground());
    helpMenu.setText("Help");

    aboutCommand.setBackground(newProjectCommand.getBackground());
    aboutCommand.setForeground(newProjectCommand.getForeground());
    aboutCommand.setText("About");
    helpMenu.add(aboutCommand);

    tutorialCommand.setBackground(newProjectCommand.getBackground());
    tutorialCommand.setForeground(newProjectCommand.getForeground());
    tutorialCommand.setText("Tutorial");
    helpMenu.add(tutorialCommand);

    menuBar.add(helpMenu);

    setJMenuBar(menuBar);

    GroupLayout layout = new GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(Alignment.LEADING)
            .addComponent(framePanel, Alignment.TRAILING, GroupLayout.DEFAULT_SIZE, GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(framePanel, GroupLayout.DEFAULT_SIZE, GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );
    pack();
}

//open a Project File in GLP(GUI LaTeX Project)
format
private void
openProjectCommandActionPerformed(ActionEvent evt) {
    // Show open dialog; this method does not return until the dialog is closed
    // Start in current directory
    fc.setCurrentDirectory(new File("."));
    fc.addChoosableFileFilter(new ProjectFileFilter());
    int retval =
    fc.showOpenDialog(editorComponents.this);
    if (retval == JFileChooser.APPROVE_OPTION) {
        File file = fc.getSelectedFile();
        String fileType =
        fc.getTypeDescription(file);
        if (fileType.equals("GLP File")) {
            // Invoke the readfile method in this class
            String file_string = rf.readContents(file);
            if (file_string != null) {
                openProjectFromFile(file);
                project = file;
            } else {
                JOptionPane.showMessageDialog(null, "File is empty.", "Error!",
                JOptionPane.ERROR_MESSAGE);
            }
        } else {
            JOptionPane.showMessageDialog(null, "Invalid file. Choose GUI LaTeX Project files only.", "Error!",
            JOptionPane.ERROR_MESSAGE);
            return;
        }
    }
}

//save Project File in GLP(GUI LaTeX Project)
format
private void
saveCommandActionPerformed(ActionEvent evt) {
    projectFileChooser pfc = new projectFileChooser();
    if (pfc.chooseFile() == 0) {
        project = null;
        project = pfc.getFile();
        saveProjectToFile(project);
    }
}

private void
exitCommandActionPerformed(ActionEvent evt) {
    System.exit(0);
}

private void
insertImageCommandActionPerformed(ActionEvent evt) {
    showImageFrame dialog = new showImageFrame(new javax.swing.JFrame(), true);
    dialog.setVisible(true);
    if(dialog.getReturnStatus()==1)
        addImage(dialog.iCaptionVal.getText(), dialog.iPathVal.getText(), dialog.texSize, dialog.size);
}

```

```

private void
previewCommandActionPerformed(ActionEvent
evt) {
    CompileAndRunProject(0);
}

private void
finalizeCommandActionPerformed(ActionEvent evt)
{
    CompileAndRunProject(1);
}

private void
citationCommandActionPerformed(ActionEvent
evt) {
    addCitation();
}

private void
equationCommandActionPerformed(ActionEvent
evt) {
    final Runtime r = Runtime.getRuntime();
    Process p;
    try {
        p = r.exec("cmd /C start
C:\\Progra~1\\TeXaide\\Eqnedt32.exe");
        if ( p.waitFor() != 0 ) {
            JOptionPane.showMessageDialog(null, "Error
encountered while trying to open equation editor",
"Error!", JOptionPane.ERROR_MESSAGE);
            return;
        }
    } catch (Exception err) {
        err.printStackTrace();
    }
    insertEquation();
}

private void
oListCommandActionPerformed(ActionEvent evt) {
    insertBullets("ol");
}

private void
uListCommandActionPerformed(ActionEvent evt) {
    insertBullets("ul");
}

private void
newProjectCommandActionPerformed(ActionEvent
evt) {
    project = null;
    np = new newProject(0);
    this.dispose();
}

public abstract void addImage(String bName,
String bPath, String realWidth, Double width);
public abstract void insertBullets(String type);
public abstract void insertEquation();
public abstract void addCitation();
public abstract void saveProjectToFile(File file);
public abstract void openProjectFromFile(File file);
public abstract void CompileAndRunProject(int
flag);

private javax.swing.JMenuItem editCommand;
private javax.swing.JMenuItem citationCommand;

```

```

protected static javax.swing.JMenuItem
copyCommand;
protected static javax.swing.JMenuItem
cutCommand;
protected javax.swing.JMenu editMenu;
private javax.swing.JMenuItem
equationCommand;
private javax.swing.JMenuItem exitCommand;
private javax.swing.JMenu fileMenu;
private javax.swing.JMenuItem viewCommand;
private JPanel framePanel;
private javax.swing.JMenu helpMenu;
private javax.swing.JMenuItem
insertImageCommand;
private javax.swing.JMenu insertMenu;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JSeparator jSeparator2;
protected JPanel leftSidePanel;
protected JPanel mainPanel;
private javax.swing.JMenuBar menuBar;
private javax.swing.JMenuItem
newProjectCommand;
private javax.swing.JMenuItem oListCommand;
private javax.swing.JMenuItem
openProjectCommand;
protected static javax.swing.JMenuItem
pasteCommand;
private javax.swing.JMenuItem compileCommand;
protected static javax.swing.JMenuItem
redoCommand;
protected JPanel rightSidePanel;
private javax.swing.JMenu runMenu;
private javax.swing.JMenuItem saveCommand;
protected static javax.swing.JMenuItem
selectAllCommand;
private javax.swing.JToolBar toolbar;
private javax.swing.JMenuItem tutorialCommand;
private javax.swing.JMenuItem uListCommand;
protected static javax.swing.JMenuItem
undoCommand;
}

```

**/\*\* imageFileFilter.java \*/**

```

package upmguilte;

import java.io.File;

//create filter for TeX compatible images only
public class imageFileFilter extends
javax.swing.filechooser.FileFilter {

    public boolean accept(File f) {
        if (f.getName().toLowerCase().endsWith(".jpg"))
            return
f.getName().toLowerCase().endsWith(".jpg");
        else if
(f.getName().toLowerCase().endsWith(".jpeg"))
            return
f.getName().toLowerCase().endsWith(".jpeg");
        else if
(f.getName().toLowerCase().endsWith(".png"))
            return
f.getName().toLowerCase().endsWith(".png");
        return f.isDirectory();
    }

    public String getDescription() {
        return "TEX image(*.jpg, *.jpeg, *.png)";
    }
}

```

```

    }
}

/** newCitation.java */

package upmguilte;

import java.awt.Component;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.GroupLayout;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JSeparator;
import javax.swing.JTextField;
import javax.swing.GroupLayout.Alignment;
import javax.swing.LayoutStyle.ComponentPlacement;

public class newCitation extends
javax.swing.JDialog {

    private static final long serialVersionUID = 1L;
    private JLabel urlLbl = new JLabel();
    private JTextField urlVal = new JTextField();
    private String citation = "";
    private String texCitation = "";

    public static final int RET_CANCEL = 0;
    public static final int RET_OK = 1;

    public newCitation(java.awt.Frame parent,
boolean modal) {
        super(parent, modal);
        initComponents();
        urlLbl.setText("URL");
    }

    public int getReturnStatus() {
        return returnStatus;
    }

    void setCitationCnt(int citationCnt) {
        refNumVal.setText((citationCnt+1)+"");
    }

    private void initComponents() {
        okButton = new JButton();
        cancelButton = new JButton();
        refNumLbl = new JLabel();
        refNumVal = new JTextField();
        pubTypeLbl = new JLabel();
        pubTypeCombo = new JComboBox();
        pubLbl = new JLabel();
        authorLbl = new JLabel();
        titleLbl = new JLabel();
        authorVal = new JTextField();
        pubVal = new JTextField();
        titleVal = new JTextField();
        yearOfPubLbl = new JLabel();
        yearOfPubVal = new JTextField();
        jSeparator1 = new JSeparator();
        placeOfPubVal = new JTextField();
        placeOfPubLbl = new JLabel();

```

```

        addWindowListener(new
java.awt.event.WindowAdapter() {
            public void
windowClosing(java.awt.event.WindowEvent evt) {
                closeDialog(evt);
            }
        });

        okButton.setText("OK");
        okButton.addActionListener(new
ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                okButtonActionPerformed(evt);
            }
        });

        cancelButton.setText("Cancel");
        cancelButton.addActionListener(new
ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                cancelButtonActionPerformed(evt);
            }
        });

        refNumLbl.setText("Reference #");
        refNumVal.setEditable(false);
        pubTypeLbl.setText("Type of Publication");

        pubTypeCombo.setModel(new
javax.swing.DefaultComboBoxModel(new String[] {
"Book/Journal", "Online Journal", "Website" }));
        pubTypeCombo.addActionListener(new
ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                pubTypeComboActionPerformed(evt);
            }
        });

        pubLbl.setText("Publisher");
        authorLbl.setText("Author");
        titleLbl.setText("Title");
        yearOfPubLbl.setText("Year of Publication");
        placeOfPubLbl.setText("Place of Publication");

        GroupLayout layout = new
GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .add(layout.createParallelGroup(Alignment.LEADING)
                        .add(okButton)
                        .add(cancelButton)
                        .add(refNumLbl)
                        .add(pubTypeLbl)
                        .add(pubLbl)
                        .add(authorLbl)
                        .add(titleLbl)
                        .add(yearOfPubLbl)
                        .add(placeOfPubLbl))
                    .addContainerGap(), GroupLayout.PREFERRED_SIZE,
GroupLayout.PREFERRED_SIZE,
RELATED)
                .addGroup(layout.createParallelGroup(Alignment.LEADING)
                    .add(refNumVal)
                    .add(pubTypeCombo)
                    .add(authorVal)
                    .add(pubVal)
                    .add(titleVal)
                    .add(yearOfPubVal)
                    .add(jSeparator1)
                    .add(placeOfPubVal))
                    .addContainerGap(), GroupLayout.PREFERRED_SIZE,
RELATED)
                .addGroup(layout.createParallelGroup(Alignment.LEADING)
                    .add(layout.createSequentialGroup()
                        .add(layout.createParallelGroup(Alignment.LEADING)
                            .add(refNumVal)
                            .add(pubTypeCombo)
                            .add(authorVal)
                            .add(pubVal)
                            .add(titleVal)
                            .add(yearOfPubVal)
                            .add(jSeparator1)
                            .add(placeOfPubVal))
                        .addContainerGap(), GroupLayout.PREFERRED_SIZE,
RELATED)
                    .add(layout.createParallelGroup(Alignment.LEADING)
                        .add(okButton)
                        .add(cancelButton)
                        .add(refNumLbl)
                        .add(pubTypeLbl)
                        .add(pubLbl)
                        .add(authorLbl)
                        .add(titleLbl)
                        .add(yearOfPubLbl)
                        .add(placeOfPubLbl))
                        .addContainerGap(), GroupLayout.PREFERRED_SIZE,
RELATED)
                .addContainerGap(), GroupLayout.PREFERRED_SIZE,
RELATED)
        );
    }

```



```

        .addComponent(refNumLbl)
        .addGap(10, 10, 10)
        .addComponent(refNumVal,
GroupLayout.PREFERRED_SIZE, 21,
GroupLayout.PREFERRED_SIZE)
        .addGap(24, 24, 24)
        .addComponent(pubTypeLbl)
        .addPreferredGap(ComponentPlacement.
UNRELATED)
        .addComponent(pubTypeCombo,
GroupLayout.PREFERRED_SIZE, 162,
GroupLayout.PREFERRED_SIZE))
        .addGroup(Alignment.LEADING,
layout.createSequentialGroup())
        .addGroup(layout.createParallelGroup(Alig
nment.TRAILING, false)
        .addComponent(jSeparator1,
Alignment.LEADING)
        .addGroup(Alignment.LEADING,
layout.createSequentialGroup())
        .addGroup(layout.createParallelGroup(
Alignment.LEADING)
        .addComponent(pubLbl)
        .addComponent(authorLbl)
        .addComponent(titleLbl))
        .addPreferredGap(ComponentPlaceme
nt.RELATED)
        .addGroup(layout.createParallelGroup(
Alignment.LEADING)
        .addComponent(authorVal,
GroupLayout.PREFERRED_SIZE, 238,
GroupLayout.PREFERRED_SIZE)
        .addComponent(pubVal,
GroupLayout.PREFERRED_SIZE, 238,
GroupLayout.PREFERRED_SIZE)
        .addComponent(titleVal,
GroupLayout.PREFERRED_SIZE, 337,
GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(ComponentPlaceme
nt.RELATED)))
        .addPreferredGap(ComponentPlacement.
RELATED, GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        .addGap(19, 19, 19))
        .addGroup(layout.createSequentialGroup())
        .addGroup(layout.createParallelGroup(Align
ment.LEADING)
        .addComponent(yearOfPubLbl)
        .addComponent(placeOfPubLbl))
        .addPreferredGap(ComponentPlacement.RE
LATED)
        .addGroup(layout.createParallelGroup(Align
ment.TRAILING)
        .addComponent(placeOfPubVal,
Alignment.LEADING, GroupLayout.DEFAULT_SIZE,
103, Short.MAX_VALUE)
        .addComponent(yearOfPubVal,
Alignment.LEADING,
GroupLayout.PREFERRED_SIZE, 103,
GroupLayout.PREFERRED_SIZE))
        .addContainerGap(204,
Short.MAX_VALUE)))
    );

    layout.linkSize(javax.swing.SwingConstants.HOR
IZONTAL, new Component[] {cancelButton,
okButton});

    layout.setVerticalGroup(

        layout.createParallelGroup(Alignment.LEADING
)
        .addGroup(layout.createSequentialGroup())
        .addGap(26, 26, 26)
        .addGroup(layout.createParallelGroup(Alignm
ent.BASELINE)
        .addComponent(refNumVal,
GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE)
        .addComponent(refNumLbl)
        .addComponent(pubTypeLbl)
        .addComponent(pubTypeCombo,
GroupLayout.PREFERRED_SIZE, 20,
GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(ComponentPlacement.UNR
ELATED)
        .addComponent(jSeparator1,
GroupLayout.PREFERRED_SIZE, 8,
GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(ComponentPlacement.RELA
TED)
        .addGroup(layout.createParallelGroup(Alignm
ent.BASELINE)
        .addComponent(titleLbl,
GroupLayout.PREFERRED_SIZE, 12,
GroupLayout.PREFERRED_SIZE)
        .addComponent(titleVal,
GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(ComponentPlacement.RELA
TED)
        .addGroup(layout.createParallelGroup(Alignm
ent.BASELINE)
        .addComponent(authorLbl)
        .addComponent(authorVal,
GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(ComponentPlacement.RELA
TED)
        .addGroup(layout.createParallelGroup(Alignm
ent.BASELINE)
        .addComponent(pubLbl)
        .addComponent(pubVal,
GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createParallelGroup(Alignm
ent.LEADING)
        .addGroup(layout.createSequentialGroup())
        .addGap(58, 58, 58)
        .addGroup(layout.createParallelGroup(Align
ment.BASELINE)
        .addComponent(okButton)
        .addComponent(cancelButton)))
        .addGroup(layout.createSequentialGroup())
        .addGap(12, 12, 12)
        .addGroup(layout.createParallelGroup(Align
ment.BASELINE)
        .addComponent(placeOfPubLbl)
        .addComponent(placeOfPubVal,
GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(ComponentPlacement.RE
LATED)

```

```

        .addGroup(layout.createParallelGroup(Align
ment.BASELINE)
        .addComponent(yearOfPubLbl)
        .addComponent(yearOfPubVal,
 GroupLayout.PREFERRED_SIZE,
 GroupLayout.DEFAULT_SIZE,
 GroupLayout.PREFERRED_SIZE)))
        .addContainerGap(40, Short.MAX_VALUE)
    );

    pack();
}

/** Closes the dialog */
private void
closeDialog(java.awt.event.WindowEvent evt) {
    doClose(RET_CANCEL);
}

private void
pubTypeComboActionPerformed(ActionEvent evt)
{
    if (pubTypeCombo.getSelectedIndex() == 0) {
        authorLbl.setVisible(true);
        authorVal.setVisible(true);
        pubLbl.setVisible(true);
        pubVal.setVisible(true);
        urlLbl.setVisible(false);
        urlVal.setVisible(false);
        yearOfPubLbl.setVisible(true);
        yearOfPubVal.setVisible(true);
        placeOfPubLbl.setVisible(true);
        placeOfPubVal.setVisible(true);
        yearOfPubLbl.setText("Date of Publication");
        GroupLayout layout = new
 GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(Alignment.LEADING)
                .addGroup(layout.createSequentialGroup())
                .addContainerGap()
                .addGroup(layout.createParallelGroup(Align
ment.LEADING)
                    .addGroup(layout.createSequentialGroup())
                    .addGroup(layout.createParallelGroup(Align
ment.TRAILING)
                        .addGroup(layout.createSequentialGroup()
)
                            .addComponent(okButton,
 GroupLayout.PREFERRED_SIZE, 67,
 GroupLayout.PREFERRED_SIZE)
                            .addPreferredGap(ComponentPlacemen
t.RELATED)
                            .addComponent(cancelButton)
                            .addPreferredGap(ComponentPlacemen
t.RELATED))
                    .addGroup(Alignment.LEADING,
 layout.createSequentialGroup())
                        .addComponent(refNumLbl)
                        .addGap(10, 10, 10)
                        .addComponent(refNumVal,
 GroupLayout.PREFERRED_SIZE, 21,
 GroupLayout.PREFERRED_SIZE)
                        .addGap(24, 24, 24)
                        .addComponent(pubTypeLbl)
                        .addPreferredGap(ComponentPlacemen
t.UNRELATED)

```

```

        .addComponent(pubTypeCombo,
 GroupLayout.PREFERRED_SIZE, 162,
 GroupLayout.PREFERRED_SIZE))
        .addGroup(Alignment.LEADING,
 layout.createSequentialGroup())
        .addGroup(layout.createParallelGroup(A
lignment.TRAILING, false)
            .addComponent(jSeparator1,
 Alignment.LEADING)
            .addGroup(Alignment.LEADING,
 layout.createSequentialGroup())
            .addGroup(layout.createParallelGrou
p(Alignment.LEADING)
                .addComponent(pubLbl)
                .addComponent(authorLbl)
                .addComponent(titleLbl))
            .addPreferredGap(ComponentPlacem
ent.RELATED)
            .addGroup(layout.createParallelGrou
p(Alignment.LEADING)
                .addComponent(authorVal,
 GroupLayout.PREFERRED_SIZE, 337,
 GroupLayout.PREFERRED_SIZE)
                .addComponent(pubVal,
 GroupLayout.PREFERRED_SIZE, 337,
 GroupLayout.PREFERRED_SIZE)
                .addComponent(titleVal,
 GroupLayout.PREFERRED_SIZE, 337,
 GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(ComponentPlacem
ent.RELATED)))
        .addPreferredGap(ComponentPlacemen
t.RELATED, GroupLayout.DEFAULT_SIZE,
 Short.MAX_VALUE)))
        .addGap(19, 19, 19))
        .addGroup(layout.createSequentialGroup())
        .addGroup(layout.createParallelGroup(Align
ment.LEADING)
            .addComponent(yearOfPubLbl)
            .addComponent(placeOfPubLbl))
            .addPreferredGap(ComponentPlacement.R
ELATED)
            .addGroup(layout.createParallelGroup(Align
ment.TRAILING)
                .addComponent(placeOfPubVal,
 Alignment.LEADING, GroupLayout.DEFAULT_SIZE,
 103, Short.MAX_VALUE)
                .addComponent(yearOfPubVal,
 Alignment.LEADING,
 GroupLayout.PREFERRED_SIZE, 103,
 GroupLayout.PREFERRED_SIZE))
            .addContainerGap(204,
 Short.MAX_VALUE)))
    );
    layout.linkSize(javax.swing.SwingConstants.HO
RIZONTAL, new Component[] { cancelButton,
 okButton});

    layout.setVerticalGroup(
        layout.createParallelGroup(Alignment.LEADING)
            .addGroup(layout.createSequentialGroup())
            .addGap(26, 26, 26)
            .addGroup(layout.createParallelGroup(Align
ment.BASELINE)
                .addComponent(refNumVal,
 GroupLayout.PREFERRED_SIZE,
 GroupLayout.DEFAULT_SIZE,
 GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(refNumLbl)
        .addComponent(pubTypeLbl)
        .addComponent(pubTypeCombo,
GroupLayout.PREFERRED_SIZE, 20,
GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(ComponentPlacement.UN
RELATED)
        .addComponent(jSeparator1,
GroupLayout.PREFERRED_SIZE, 8,
GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(ComponentPlacement.REL
ATED)
        .addGroup(layout.createParallelGroup(Align
ment.BASELINE)
        .addComponent(titleLbl,
GroupLayout.PREFERRED_SIZE, 12,
GroupLayout.PREFERRED_SIZE)
        .addComponent(titleVal,
GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(ComponentPlacement.REL
ATED)
        .addGroup(layout.createParallelGroup(Align
ment.BASELINE)
        .addComponent(authorLbl)
        .addComponent(authorVal,
GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(ComponentPlacement.REL
ATED)
        .addGroup(layout.createParallelGroup(Align
ment.BASELINE)
        .addComponent(pubLbl)
        .addComponent(pubVal,
GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createParallelGroup(Align
ment.LEADING)
        .addGroup(layout.createSequentialGroup())
        .addGap(58, 58, 58)
        .addGroup(layout.createParallelGroup(Align
ment.BASELINE)
        .addComponent(okButton)
        .addComponent(cancelButton))
        .addGroup(layout.createSequentialGroup())
        .addGap(12, 12, 12)
        .addGroup(layout.createParallelGroup(Align
ment.BASELINE)
        .addComponent(placeOfPubLbl)
        .addComponent(placeOfPubVal,
GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(ComponentPlacement.R
ELATED)
        .addGroup(layout.createParallelGroup(Align
ment.BASELINE)
        .addComponent(yearOfPubLbl)
        .addComponent(yearOfPubVal,
GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE)))
        .addContainerGap(40, Short.MAX_VALUE)
);

```

```

    } else if (pubTypeCombo.getSelectedIndex() ==
1) {
        authorLbl.setVisible(true);
        authorVal.setVisible(true);
        pubLbl.setVisible(false);
        pubVal.setVisible(false);
        urlLbl.setVisible(true);
        urlVal.setVisible(true);
        yearOfPubLbl.setVisible(true);
        yearOfPubVal.setVisible(true);
        placeOfPubLbl.setVisible(false);
        placeOfPubVal.setVisible(false);
        yearOfPubLbl.setText("Date");
        GroupLayout layout = new
GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
layout.createParallelGroup(Alignment.LEADING)
G)
        .addGroup(layout.createSequentialGroup())
        .addContainerGap()
        .addGroup(layout.createParallelGroup(Align
ment.LEADING)
        .addGroup(layout.createSequentialGroup())
        .addGroup(layout.createParallelGroup(Align
ment.TRAILING)
        .addGroup(layout.createSequentialGroup()
)
        .addComponent(okButton,
GroupLayout.PREFERRED_SIZE, 67,
GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(ComponentPlacemen
t.RELATED)
        .addComponent(cancelButton)
        .addPreferredGap(ComponentPlacemen
t.RELATED))
        .addGroup(Alignment.LEADING,
layout.createSequentialGroup())
        .addComponent(refNumLbl)
        .addGap(10, 10, 10)
        .addComponent(refNumVal,
GroupLayout.PREFERRED_SIZE, 21,
GroupLayout.PREFERRED_SIZE)
        .addGap(24, 24, 24)
        .addComponent(pubTypeLbl)
        .addPreferredGap(ComponentPlacemen
t.UNRELATED)
        .addComponent(pubTypeCombo,
GroupLayout.PREFERRED_SIZE, 162,
GroupLayout.PREFERRED_SIZE))
        .addGroup(Alignment.LEADING,
layout.createSequentialGroup())
        .addGroup(layout.createParallelGroup(A
lignment.TRAILING, false)
        .addComponent(jSeparator1,
Alignment.LEADING)
        .addGroup(Alignment.LEADING,
layout.createSequentialGroup())
        .addGroup(layout.createParallelGrou
p(Alignment.LEADING)
        .addComponent(authorLbl)
        .addComponent(titleLbl))
        .addGap(14, 14, 14)
        .addGroup(layout.createParallelGrou
p(Alignment.LEADING)
        .addComponent(authorVal,
GroupLayout.PREFERRED_SIZE, 346,
GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(titleVal,
        GroupLayout.PREFERRED_SIZE, 346,
        GroupLayout.PREFERRED_SIZE)
        .addGroup(layout.createParallelGroup(
        Alignment.TRAILING, false)
        .addComponent(urlVal,
        Alignment.LEADING)
        .addComponent(yearOfPubVal,
        Alignment.LEADING, GroupLayout.DEFAULT_SIZE,
        103, Short.MAX_VALUE))))
        .addPreferredGap(ComponentPlacemen
        t.RELATED, GroupLayout.DEFAULT_SIZE,
        Short.MAX_VALUE))
        .addGap(21, 21, 21))
        .addComponent(yearOfPubLbl)
        .addComponent(urlLbl)))
    );

    layout.linkSize(javax.swing.SwingConstants.HO
    RIZONTAL, new Component[] {cancelButton,
    okButton});

    layout.setVerticalGroup(
    layout.createParallelGroup(Alignment.LEADIN
    G)
        .addGroup(layout.createSequentialGroup())
        .addGap(26, 26, 26)
        .addGroup(layout.createParallelGroup(Align
        ment.BASELINE)
        .addComponent(refNumVal,
        GroupLayout.PREFERRED_SIZE,
        GroupLayout.DEFAULT_SIZE,
        GroupLayout.PREFERRED_SIZE)
        .addComponent(refNumLbl)
        .addComponent(pubTypeLbl)
        .addComponent(pubTypeCombo,
        GroupLayout.PREFERRED_SIZE, 20,
        GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(ComponentPlacement.UN
        RELATED)
        .addComponent(jSeparator1,
        GroupLayout.PREFERRED_SIZE, 8,
        GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(ComponentPlacement.REL
        ATED)
        .addGroup(layout.createParallelGroup(Align
        ment.BASELINE)
        .addComponent(titleLbl,
        GroupLayout.PREFERRED_SIZE, 12,
        GroupLayout.PREFERRED_SIZE)
        .addComponent(titleVal,
        GroupLayout.PREFERRED_SIZE,
        GroupLayout.DEFAULT_SIZE,
        GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(ComponentPlacement.REL
        ATED)
        .addGroup(layout.createParallelGroup(Align
        ment.BASELINE)
        .addComponent(authorLbl)
        .addComponent(authorVal,
        GroupLayout.PREFERRED_SIZE,
        GroupLayout.DEFAULT_SIZE,
        GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(ComponentPlacement.REL
        ATED)
        .addGroup(layout.createParallelGroup(Align
        ment.BASELINE)
        .addComponent(yearOfPubLbl)

```

```

        .addComponent(yearOfPubVal,
        GroupLayout.PREFERRED_SIZE,
        GroupLayout.DEFAULT_SIZE,
        GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(ComponentPlacement.REL
        ATED)
        .addGroup(layout.createParallelGroup(Align
        ment.LEADING)
        .addGroup(layout.createSequentialGroup())
        .addGap(22, 22, 22)
        .addGroup(layout.createParallelGroup(Align
        ment.BASELINE)
        .addComponent(okButton)
        .addComponent(cancelButton)))
        .addGroup(layout.createParallelGroup(Align
        ment.BASELINE)
        .addComponent(urlLbl)
        .addComponent(urlVal,
        GroupLayout.PREFERRED_SIZE,
        GroupLayout.DEFAULT_SIZE,
        GroupLayout.PREFERRED_SIZE))
        .addContainerGap(25, Short.MAX_VALUE))
    );

    } else if (pubTypeCombo.getSelectedIndex() ==
    2) {
        authorLbl.setVisible(false);
        authorVal.setVisible(false);
        pubLbl.setVisible(false);
        pubVal.setVisible(false);
        urlLbl.setVisible(true);
        urlVal.setVisible(true);
        yearOfPubLbl.setVisible(false);
        yearOfPubVal.setVisible(false);
        placeOfPubLbl.setVisible(false);
        placeOfPubVal.setVisible(false);
        GroupLayout layout = new
        GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
        layout.createParallelGroup(Alignment.LEADIN
        G)
        .addGroup(layout.createSequentialGroup())
        .addContainerGap()
        .addGroup(layout.createParallelGroup(Align
        ment.LEADING)
        .addGroup(layout.createSequentialGroup())
        .addComponent(refNumLbl)
        .addGap(10, 10, 10)
        .addComponent(refNumVal,
        GroupLayout.PREFERRED_SIZE, 21,
        GroupLayout.PREFERRED_SIZE)
        .addGap(24, 24, 24)
        .addComponent(pubTypeLbl)
        .addPreferredGap(ComponentPlacement.U
        NRELATED)
        .addComponent(pubTypeCombo,
        GroupLayout.PREFERRED_SIZE, 162,
        GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup())
        .addGroup(layout.createParallelGroup(Align
        ment.TRAILING, false)
        .addComponent(jSeparator1,
        Alignment.LEADING)
        .addGroup(Alignment.LEADING,
        layout.createSequentialGroup())
        .addGroup(layout.createParallelGroup(A
        lignment.LEADING)
        .addComponent(titleLbl)

```

```

        .addComponent(urlLbl)
        .addGap(27, 27, 27)
        .addGroup(layout.createParallelGroup(Alignment.LEADING, false)
            .addComponent(urlVal)
            .addComponent(titleVal, GroupLayout.DEFAULT_SIZE, 346, Short.MAX_VALUE)))
        .addPreferredGap(ComponentPlacement.RELATED, GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGroup(layout.createSequentialGroup())
        .addPreferredGap(ComponentPlacement.RELATED, 253, Short.MAX_VALUE)
        .addComponent(okButton, GroupLayout.PREFERRED_SIZE, GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(ComponentPlacement.RELATED)
        .addComponent(cancelButton))
        .addGap(21, 21, 21)
    );

    layout.linkSize(javax.swing.SwingConstants.HORIZONTAL, new Component[] {cancelButton, okButton});

    layout.setVerticalGroup(G)
    layout.createParallelGroup(Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(26, 26, 26)
            .addGroup(layout.createParallelGroup(Baseline.BASELINE)
                .addComponent(refNumVal, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
                .addComponent(refNumLbl)
                .addComponent(pubTypeLbl)
                .addComponent(pubTypeCombo, GroupLayout.PREFERRED_SIZE, 20, GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(ComponentPlacement.UNRELATED)
            .addComponent(jSeparator1, GroupLayout.PREFERRED_SIZE, 8, GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(Baseline.BASELINE)
                .addComponent(titleLbl, GroupLayout.PREFERRED_SIZE, 12, GroupLayout.PREFERRED_SIZE)
                .addComponent(titleVal, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(ComponentPlacement.RELATED)
            .addGroup(layout.createParallelGroup(Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGap(52, 52, 52)
                    .addGroup(layout.createParallelGroup(Alignment.BASELINE)
                        .addComponent(okButton)
                        .addComponent(cancelButton)))
            );

```

```

        .addGroup(layout.createParallelGroup(Alignment.BASELINE)
            .addComponent(urlLbl)
            .addComponent(urlVal, GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
        .addContainerGap(25, Short.MAX_VALUE)
    );
    pack();
}

private void
cancelButtonActionPerformed(ActionEvent evt) {
    doClose(RET_CANCEL);
}

private void
okButtonActionPerformed(ActionEvent evt) {
    if (pubTypeCombo.getSelectedIndex() == 0) {
        if (titleVal.getText().trim().equals("") ||
            authorVal.getText().trim().equals("") ||
            yearOfPubVal.getText().trim().equals("")) {
            JOptionPane.showMessageDialog(null, "Title, Author and Date are mandatory fields", "ERROR!", JOptionPane.ERROR_MESSAGE);
        } else {
            citation = authorVal.getText().trim() + ". " +
                yearOfPubVal.getText().trim() + ". " +
                "<i>" + titleVal.getText().trim() +
            ".</i> ";
            if (!placeOfPubVal.getText().trim().equals(""))
                if (!pubVal.getText().trim().equals(""))
                    citation += placeOfPubVal.getText().trim() + ": " +
                        pubVal.getText().trim() + ". ";
            else
                citation += placeOfPubVal.getText().trim() + ". ";
            else if (!pubVal.getText().trim().equals(""))
                citation += pubVal.getText().trim() + ". ";

            texCitation = citation.replace("<i>", "\\textit{");
            texCitation = texCitation.replace("</i>", "}");

            doClose(RET_OK);
        }
    } else if (pubTypeCombo.getSelectedIndex() == 1) {
        if (titleVal.getText().trim().equals("") ||
            authorVal.getText().trim().equals("") ||
            yearOfPubVal.getText().trim().equals("") ||
            urlVal.getText().trim().equals("")) {
            JOptionPane.showMessageDialog(null, "Title, Author, Date and URL are mandatory fields", "ERROR!", JOptionPane.ERROR_MESSAGE);
        } else {
            citation = authorVal.getText().trim() + ". " +
                yearOfPubVal.getText().trim() + ". " +
                "<i>" + titleVal.getText().trim() +
            ".</i> " +
                "<br><url>" + urlVal.getText().trim() +
            ".</url>";
            texCitation = authorVal.getText().trim() + ". " +
                yearOfPubVal.getText().trim() + ". " +
            "\\textit{" + titleVal.getText().trim() +
            "}.} \\|\\|\\| " +

```

```

        "\url{" + urlVal.getText().trim() +
        ".}";
        doClose(RET_OK);
    } else if (pubTypeCombo.getSelectedIndex() ==
2) {
        if (titleVal.getText().trim().equals("") ||
            urlVal.getText().trim().equals("")) {
            JOptionPane.showMessageDialog(null, "Title
and URL are mandatory fields", "ERROR!",
JOptionPane.ERROR_MESSAGE);
        } else {
            citation = "<i>" + titleVal.getText().trim() +
        "</i> " +
            "<br><url>" + urlVal.getText().trim()
+ "</url>";
            texCitation = "\\textit{" +
titleVal.getText().trim() + ".} \\url{" +
            "\url{" + urlVal.getText().trim() + ".}";
            doClose(RET_OK);
        }
    }
}

public String getCitation() {
    return citation;
}

public String getBibTex() {
    return texCitation;
}

private void doClose(int retStatus) {
    returnStatus = retStatus;
    setVisible(false);
    dispose();
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new
Runnable() {
        public void run() {
            newCitation dialog = new newCitation(new
javax.swing.JFrame(), true);
            dialog.addWindowListener(new
java.awt.event.WindowAdapter() {
                public void
windowClosing(java.awt.event.WindowEvent e) {
                    System.exit(0);
                }
            });
            dialog.setVisible(true);
        }
    });
}

private JLabel authorLbl;
private JTextField authorVal;
private JButton cancelButton;
private JSeparator jSeparator1;
private JButton okButton;
private JLabel placeOfPubLbl;
private JTextField placeOfPubVal;
private JLabel pubLbl;
private JComboBox pubTypeCombo;
private JLabel pubTypeLbl;
private JTextField pubVal;
private JLabel refNumLbl;
private JTextField refNumVal;

```

```

private JLabel titleLbl;
private JTextField titleVal;
private JLabel yearOfPubLbl;
private JTextField yearOfPubVal;
private int returnStatus = RET_CANCEL;
}

```

**/\*\* newProject.java \*/**

```

package upmguilte;

import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.FocusAdapter;
import java.awt.event.FocusEvent;
import java.awt.event.InputEvent;
import java.awt.event.KeyEvent;
import java.awt.event.MouseEvent;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.IOException;
import java.io.PrintStream;
import java.io.StringReader;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.logging.Level;
import java.util.logging.Logger;

import javax.swing.Action;
import javax.swing.BorderFactory;
import javax.swing.DefaultComboBoxModel;
import javax.swing.GroupLayout;
import javax.swing.Imagelcon;
import javax.swing.InputVerifier;
import javax.swing.JComboBox;
import javax.swing.JComponent;
import javax.swing.JEditorPane;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextField;
import javax.swing.KeyStroke;
import javax.swing.UIManager;
import javax.swing.GroupLayout.Alignment;
import
javax.swing.LayoutStyle.ComponentPlacement;
import javax.swing.event.TreeSelectionEvent;
import javax.swing.text.BadLocationException;
import javax.swing.text.DefaultEditorKit;
import javax.swing.text.Document;
import javax.swing.text.JTextComponent;
import javax.swing.text.html.HTMLEditorKit;
import javax.swing.tree.DefaultMutableTreeNode;
import javax.swing.undo.UndoManager;

public class newProject extends editorComponents
{

    private static final long serialVersionUID = 1L;
    protected int noOfChapters = 12;
    protected String chapTitle[] = new String[]
{"Preamble", "Abstract", "Introduction", "Review of
Related Literature", "Theoretical Framework",
"Design and Implementation", "Results",

```

```

"Discussion", "Conclusion", "Recommendation",
"Bibliography", "Appendix", "Acknowledgement"};
protected int bibIndex = 9;
protected JPanel activePanel;
protected static int activeIndex = -1;
protected static Chapter chapter[];
private UndoManager preambleManager = new
UndoManager();
private HashMap<Object, Action> actions;
private ArrayList<Citation> citations = new
ArrayList<Citation>();
private ArrayList<Citation> texCitations = new
ArrayList<Citation>();
private int citationCnt = 0;
int caretPosition = 0, i = 0;
boolean ulListFlag = false;
boolean olListFlag = false;
String lastKey = "";
private String equation;

public newProject() {}

public newProject(int flag) {
    super(flag);

    overRideActions();
    initComponents();

    titleValue.getDocument().addUndoableEditListen
er(preambleManager);
    authorValue.getDocument().addUndoableEditLis
tener(preambleManager);
    studNumValue.getDocument().addUndoableEdit
Listener(preambleManager);
    yearValue.getDocument().addUndoableEditListe
ner(preambleManager);

    actions=createActionTable(titleValue);
    cutCommand.setAction(actions.get(DefaultEdito
rKit.cutAction));
    copyCommand.setAction(actions.get(DefaultEdit
orKit.copyAction));
    pasteCommand.setAction(actions.get(DefaultEdi
torKit.pasteAction));
    selectAllCommand.setAction(actions.get(Default
EditorKit.selectAllAction));

    changeEditMenuCommands();

    //createChapters panels
    chapter = new Chapter[noOfChapters];
    for (i=0; i<noOfChapters; i++) {
        chapter[i] = new Chapter(i, chapTitle[i+1]);
    }
    chapter[1].Subsection(new String[]
{"Background of the Study", "Statement of the
Problem", "Objectives", "Significance of the Study",
"Scope and Limitations", "Assumptions"});

    chapter[bibIndex].textPane.setEditable(false);
    super.setTreeSubsections(1,
chapter[1].subsecTitle);
}

//The following two methods allow us to find an
action provided by the editor kit by its name.
private HashMap<Object, Action>
createActionTable(JTextComponent
textComponent) {

```

```

HashMap<Object, Action> tempactions = new
HashMap<Object, Action>();
Action[] actionsArray =
textComponent.getActions();
for (int i = 0; i < actionsArray.length; i++) {
    Action a = actionsArray[i];
    tempactions.put(a.getValue(Action.NAME), a);
}
return tempactions;
}

/** Returns an ImagemIcon, or null if the path was
invalid. */
protected static ImagemIcon
createImagemIcon(String path, String description) {
    if (path != null) {
        return new ImagemIcon(path, description);
    } else {
        System.err.println("Couldn't find file: " + path);
        return null;
    }
}

private void initComponents() {
    scrollPane = new JScrollPane();
    preamblePanel = new JPanel();
    aSheetPanel = new JPanel();
    abstractPanel = new JPanel();
    titlePagePanel = new JPanel();
    titleLabel = new JLabel();
    authorLabel = new JLabel();
    authorValue = new JTextField();
    dateLabel = new JLabel();
    studNumLabel = new JLabel();
    studNumValue = new JTextField();
    monthValue = new JComboBox();
    yearValue = new JTextField();
    titleScrollPane = new JScrollPane();
    titleValue = new javax.swing.JTextPane();
    adviserLbl = new JLabel();
    adviserVal = new JTextField();
    panelistsLbl = new JLabel();
    panel1Val = new JTextField();
    panel2Val = new JTextField();
    panel3Val = new JTextField();
    panel4Val = new JTextField();
    panel5Val = new JTextField();
    panel6Val = new JTextField();
    panel7Val = new JTextField();
    panel8Val = new JTextField();
    panel9Val = new JTextField();
    panel10Val = new JTextField();
    jLabel1 = new JLabel();
    jLabel2 = new JLabel();
    jLabel3 = new JLabel();
    jLabel4 = new JLabel();
    jLabel5 = new JLabel();
    jLabel6 = new JLabel();
    jLabel7 = new JLabel();
    jLabel8 = new JLabel();
    jLabel9 = new JLabel();
    jLabel10 = new JLabel();
    deptChairLbl = new JLabel();
    deptChairVal = new JTextField();
    unitHeadLbl = new JLabel();
    unitHeadVal = new JTextField();
    deanVal = new JTextField();
    deanLbl = new JLabel();
    asDateLbl = new JLabel();
}

```

```

asDateVal = new JTextField();
keywordsLbl = new JLabel();
keywordsVal = new JTextField();

preamblePanel.setBackground(new Color(255,
255, 255));

aSheetPanel.setBackground(new Color(79, 76,
60));
aSheetPanel.setBorder(BorderFactory.createTitl
edBorder(null, "Acceptance Sheet",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFIC
ATION,
javax.swing.border.TitledBorder.DEFAULT_POSITIO
N, new java.awt.Font("Tahoma", 0, 11), new
Color(0, 102, 0));
aSheetPanel.setAutoscrolls(true);
aSheetPanel.setOpaque(false);

abstractPanel.setBackground(new Color(79, 76,
60));
abstractPanel.setBorder(BorderFactory.createTitl
edBorder(null, "Abstract Keywords",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFIC
ATION,
javax.swing.border.TitledBorder.DEFAULT_POSITIO
N, new java.awt.Font("Tahoma", 0, 11), new
Color(0, 102, 0));
abstractPanel.setAutoscrolls(true);
abstractPanel.setOpaque(false);

titlePagePanel.setBackground(new Color(79, 76,
60));
titlePagePanel.setBorder(BorderFactory.createTitl
edBorder(null, "Title Page",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFIC
ATION,
javax.swing.border.TitledBorder.DEFAULT_POSITIO
N, new java.awt.Font("Tahoma", 0, 11), new
Color(0, 102, 0));
titlePagePanel.setOpaque(false);

titleLabel.setText("Title"); // NOI18N

authorLabel.setText("Author"); // NOI18N

authorValue.setBorder(BorderFactory.createEtche
dBorder());
authorValue.addActionListener(new
ActionListener() {
    public void actionPerformed(ActionEvent evt)
    {
        authorValueActionPerformed(evt);
    }
});

dateLabel.setText("Date"); // NOI18N

studNumLabel.setText("Student No."); // NOI18N

studNumValue.setBorder(BorderFactory.createE
tchedBorder());
studNumValue.addActionListener(new
ActionListener() {
    public void actionPerformed(ActionEvent evt)
    {
        studNumValueActionPerformed(evt);
    }
});

monthValue.setModel(new
DefaultComboBoxModel(new String[] { "January",
"February", "March", "April", "May", "June", "July",
"August", "September", "October", "November",
"December" }););
monthValue.setBorder(BorderFactory.createEmp
tyBorder(1, 1, 1, 1));
monthValue.setLightWeightPopupEnabled(false)
;

yearValue.setBorder(BorderFactory.createEtche
dBorder());
yearValue.addFocusListener(new FocusAdapter()
{
    public void focusGained(FocusEvent evt) {
        yearValueFocusGained(evt);
    }
});
yearValue.setInputVerifier(new
yearValueVerifier());

titleScrollPane.setBorder(null);

titleValue.setBorder(BorderFactory.createEtched
Border());

titleValue.getDocument().addUndoableEditListen
er(manager);

titleScrollPane.setViewportView(titleValue);

GroupLayout titlePagePanellayout = new
GroupLayout(titlePagePanel);
titlePagePanel.setLayout(titlePagePanellayout);
titlePagePanellayout.setHorizontalGroup(
titlePagePanellayout.createParallelGroup(Align
ment.LEADING)
.addGroup(titlePagePanellayout.createSequenti
alGroup()
.addGroup(titlePagePanellayout.createParallel
Group(Alignment.LEADING)
.addGroup(titlePagePanellayout.createSequ
entialGroup()
.addGap(10, 10, 10)
.addGroup(titlePagePanellayout.createPara
llegroup(Alignment.LEADING)
.addComponent(studNumLabel)
.addComponent(dateLabel)))
.addGroup(titlePagePanellayout.createSequ
entialGroup()
.addContainerGap()
.addComponent(authorLabel))
.addGroup(titlePagePanellayout.createSequ
entialGroup()
.addContainerGap()
.addComponent(titleLabel)))
.addGroup(titlePagePanellayout.createParallel
Group(Alignment.LEADING)
.addComponent(authorValue,
GroupLayout.PREFERRED_SIZE, 241,
GroupLayout.PREFERRED_SIZE)
.addGroup(titlePagePanellayout.createParallel
Group(Alignment.TRAILING, false)
.addComponent(studNumValue,
Alignment.LEADING)
.addGroup(Alignment.LEADING,
titlePagePanellayout.createSequentialGroup()

```



```

        .addComponent(monthValue,
GroupLayout.PREFERRED_SIZE, 85,
GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(ComponentPlacement.REL
ATED)
        .addComponent(yearValue,
GroupLayout.PREFERRED_SIZE, 58,
GroupLayout.PREFERRED_SIZE))
        .addComponent(titleValue,
GroupLayout.PREFERRED_SIZE, 407,
GroupLayout.PREFERRED_SIZE)
        .addContainerGap(99, Short.MAX_VALUE)
);
titlePagePanelLayout.setVerticalGroup(
titlePagePanelLayout.createParallelGroup(Align
ment.LEADING)
        .addGroup(titlePagePanelLayout.createSequent
ialGroup()
        .addContainerGap()
        .addGroup(titlePagePanelLayout.createParalle
lGroup(Alignment.BASELINE)
        .addComponent(titleLabel)
        .addComponent(titleValue,
GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
        .addGap(6, 6, 6)
        .addGroup(titlePagePanelLayout.createParalle
lGroup(Alignment.BASELINE)
        .addComponent(authorLabel)
        .addComponent(authorValue,
GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(ComponentPlacement.RELA
TED)
        .addGroup(titlePagePanelLayout.createParalle
lGroup(Alignment.BASELINE)
        .addComponent(studNumLabel)
        .addComponent(studNumValue,
GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(ComponentPlacement.RELA
TED)
        .addGroup(titlePagePanelLayout.createParalle
lGroup(Alignment.BASELINE)
        .addComponent(yearValue,
GroupLayout.PREFERRED_SIZE, 20,
GroupLayout.PREFERRED_SIZE)
        .addComponent(dateLabel)
        .addComponent(monthValue,
GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
        .addContainerGap())
);
adviserLbl.setText("Adviser");
panelistsLbl.setText("Panelists");
jLabel1.setText("1.");
jLabel2.setText("2.");
jLabel3.setText("3.");
jLabel4.setText("4.");
jLabel5.setText("5.");
jLabel6.setText("6.");
jLabel7.setText("7.");
jLabel8.setText("8.");
jLabel9.setText("9.");

```

```

jLabel10.setText("10.");
deptChairLbl.setText("Dept. Chair");
unitHeadLbl.setText("Unit Head");
deanLbl.setText("Dean");
asDateLbl.setText("Date");
keywordsLbl.setText("Keywords");

GroupLayout abstractLayout = new
GroupLayout(abstractPanel);
abstractPanel.setLayout(abstractLayout);
abstractLayout.setHorizontalGroup(
abstractLayout.createParallelGroup(Alignment.
LEADING)
        .addGroup(abstractLayout.createSequentialGro
up()
        .addContainerGap()
        .addGroup(abstractLayout.createParallelGrou
p(Alignment.LEADING)
        .addComponent(keywordsLbl,
GroupLayout.PREFERRED_SIZE, 70,
GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(ComponentPlacement.RELA
TED)
        .addGroup(abstractLayout.createParallelGrou
p(Alignment.LEADING, false)
        .addComponent(keywordsVal,
GroupLayout.PREFERRED_SIZE, 455,
GroupLayout.PREFERRED_SIZE))
        .addContainerGap())
);
abstractLayout.setVerticalGroup(
abstractLayout.createParallelGroup(Alignment.
LEADING)
        .addGroup(abstractLayout.createSequentialGro
up()
        .addContainerGap()
        .addGroup(abstractLayout.createParallelGrou
p(Alignment.BASELINE)
        .addComponent(keywordsLbl)
        .addComponent(keywordsVal,
GroupLayout.PREFERRED_SIZE, 20,
GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addContainerGap())
);

GroupLayout layout = new
GroupLayout(aSheetPanel);
aSheetPanel.setLayout(layout);
layout.setHorizontalGroup(
layout.createParallelGroup(Alignment.LEADING)
)
        .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(layout.createParallelGroup(Alignm
ent.LEADING)
        .addComponent(adviserLbl,
GroupLayout.PREFERRED_SIZE, 70,
GroupLayout.PREFERRED_SIZE)
        .addComponent(panelistsLbl)
        .addComponent(deptChairLbl)
        .addComponent(unitHeadLbl)
        .addComponent(deanLbl)
        .addComponent(asDateLbl))
        .addPreferredGap(ComponentPlacement.RELA
TED)
        .addGroup(layout.createParallelGroup(Alignm
ent.LEADING, false)
        .addComponent(adviserVal)

```

```

        .addComponent(deptChairVal)
        .addComponent(unitHeadVal)
        .addComponent(deanVal)
        .addComponent(asDateVal)
        .addGroup(layout.createSequentialGroup())
        .addComponent(jLabel5)
        .addPreferredGap(ComponentPlacement.RE
LATED)
        .addComponent(panel5Val))
        .addGroup(layout.createSequentialGroup())
        .addComponent(jLabel4)
        .addPreferredGap(ComponentPlacement.RE
LATED)
        .addComponent(panel4Val))
        .addGroup(layout.createSequentialGroup())
        .addComponent(jLabel3)
        .addPreferredGap(ComponentPlacement.RE
LATED)
        .addComponent(panel3Val))
        .addGroup(layout.createSequentialGroup())
        .addComponent(jLabel2)
        .addPreferredGap(ComponentPlacement.RE
LATED)
        .addComponent(panel2Val))
        .addGroup(layout.createSequentialGroup())
        .addComponent(jLabel1)
        .addPreferredGap(ComponentPlacement.RE
LATED)
        .addComponent(panel1Val,
 GroupLayout.PREFERRED_SIZE, 195,
 GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(Alignm
ent.LEADING)
        .addGroup(layout.createSequentialGroup())
        .addComponent(jLabel10)
        .addPreferredGap(ComponentPlacement.RE
LATED)
        .addComponent(panel10Val,
 GroupLayout.PREFERRED_SIZE, 190,
 GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup())
        .addComponent(jLabel9)
        .addPreferredGap(ComponentPlacement.RE
LATED)
        .addComponent(panel9Val,
 GroupLayout.PREFERRED_SIZE, 195,
 GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup())
        .addComponent(jLabel8)
        .addPreferredGap(ComponentPlacement.RE
LATED)
        .addComponent(panel8Val,
 GroupLayout.PREFERRED_SIZE, 195,
 GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup())
        .addComponent(jLabel7)
        .addPreferredGap(ComponentPlacement.RE
LATED)
        .addComponent(panel7Val,
 GroupLayout.PREFERRED_SIZE, 195,
 GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup())
        .addComponent(jLabel6)
        .addPreferredGap(ComponentPlacement.RE
LATED)
        .addComponent(panel6Val,
 GroupLayout.PREFERRED_SIZE, 195,
 GroupLayout.PREFERRED_SIZE))
        .addContainerGap()
        );
        layout.setVerticalGroup(
        layout.createParallelGroup(Alignment.LEADING
)
        .addGroup(layout.createSequentialGroup())
        .addContainerGap()
        .addGroup(layout.createParallelGroup(Alignm
ent.BASELINE)
        .addComponent(adviserLbl)
        .addComponent(adviserVal,
 GroupLayout.PREFERRED_SIZE, 20,
 GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(Alignm
ent.BASELINE)
        .addComponent(panelistsLbl)
        .addComponent(jLabel1)
        .addComponent(jLabel6)
        .addComponent(panel1Val,
 GroupLayout.PREFERRED_SIZE, 20,
 GroupLayout.PREFERRED_SIZE)
        .addComponent(panel6Val,
 GroupLayout.PREFERRED_SIZE, 20,
 GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(ComponentPlacement.RELA
TED)
        .addGroup(layout.createParallelGroup(Alignm
ent.BASELINE)
        .addComponent(jLabel2)
        .addComponent(jLabel7)
        .addComponent(panel2Val,
 GroupLayout.PREFERRED_SIZE, 20,
 GroupLayout.PREFERRED_SIZE)
        .addComponent(panel7Val,
 GroupLayout.PREFERRED_SIZE, 20,
 GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(ComponentPlacement.RELA
TED)
        .addGroup(layout.createParallelGroup(Alignm
ent.BASELINE)
        .addComponent(jLabel3)
        .addComponent(jLabel8)
        .addComponent(panel3Val,
 GroupLayout.PREFERRED_SIZE, 20,
 GroupLayout.PREFERRED_SIZE)
        .addComponent(panel8Val,
 GroupLayout.PREFERRED_SIZE, 20,
 GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(ComponentPlacement.RELA
TED)
        .addGroup(layout.createParallelGroup(Alignm
ent.BASELINE)
        .addComponent(jLabel4)
        .addComponent(jLabel9)
        .addComponent(panel4Val,
 GroupLayout.PREFERRED_SIZE, 20,
 GroupLayout.PREFERRED_SIZE)
        .addComponent(panel9Val,
 GroupLayout.PREFERRED_SIZE, 20,
 GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(ComponentPlacement.RELA
TED)
        .addGroup(layout.createParallelGroup(Alignm
ent.BASELINE)
        .addComponent(jLabel5)
        .addComponent(jLabel10)

```

```

        .addComponent(panel5Val,
        GroupLayout.PREFERRED_SIZE, 20,
        GroupLayout.PREFERRED_SIZE)
        .addComponent(panel10Val,
        GroupLayout.PREFERRED_SIZE, 20,
        GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addGroup(layout.createParallelGroup(Alignm
ent.BASELINE)
        .addComponent(deptChairLbl)
        .addComponent(deptChairVal,
        GroupLayout.PREFERRED_SIZE, 20,
        GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(ComponentPlacement.RELA
TED)
        .addGroup(layout.createParallelGroup(Alignm
ent.BASELINE)
        .addComponent(unitHeadLbl)
        .addComponent(unitHeadVal,
        GroupLayout.PREFERRED_SIZE, 20,
        GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(ComponentPlacement.RELA
TED)
        .addGroup(layout.createParallelGroup(Alignm
ent.BASELINE)
        .addComponent(deanLbl)
        .addComponent(deanVal,
        GroupLayout.PREFERRED_SIZE, 20,
        GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(ComponentPlacement.RELA
TED)
        .addGroup(layout.createParallelGroup(Alignm
ent.BASELINE)
        .addComponent(asDateLbl)
        .addComponent(asDateVal,
        GroupLayout.PREFERRED_SIZE, 20,
        GroupLayout.PREFERRED_SIZE))
        .addContainerGap())
    );

    GroupLayout preamblePanelLayout = new
    GroupLayout(preamblePanel);
    preamblePanel.setLayout(preamblePanelLayout)
    ;
    preamblePanelLayout.setHorizontalGroup(
    preamblePanelLayout.createParallelGroup(Align
ment.LEADING)
    .addGroup(Alignment.TRAILING,
    preamblePanelLayout.createSequentialGroup())
    .addGap(30, 30, 30)
    .addGroup(preamblePanelLayout.createParall
elGroup(Alignment.TRAILING)
    .addComponent(abstractPanel,
    Alignment.LEADING, GroupLayout.DEFAULT_SIZE,
    GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(aSheetPanel,
    Alignment.LEADING, GroupLayout.DEFAULT_SIZE,
    GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    .addComponent(titlePagePanel,
    Alignment.LEADING, GroupLayout.DEFAULT_SIZE,
    GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    .addGap(41, 41, 41))
    );
    preamblePanelLayout.setVerticalGroup(
    preamblePanelLayout.createParallelGroup(Align
ment.LEADING)
    .addGroup(preamblePanelLayout.createSequen
tialGroup())
    .addGap(20, 20, 20)

```

```

        .addComponent(titlePagePanel,
        GroupLayout.PREFERRED_SIZE,
        GroupLayout.DEFAULT_SIZE,
        GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(ComponentPlacement.UNR
ELATED)
        .addComponent(aSheetPanel,
        GroupLayout.PREFERRED_SIZE,
        GroupLayout.DEFAULT_SIZE,
        GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(ComponentPlacement.UNR
ELATED)
        .addComponent(abstractPanel,
        GroupLayout.PREFERRED_SIZE,
        GroupLayout.DEFAULT_SIZE,
        GroupLayout.PREFERRED_SIZE)
        .addGap(20, 20, 20))
    );

    scrollPane.setViewportView(preamblePanel);

    GroupLayout framePanelLayout = new
    GroupLayout(super.mainPanel);
    super.mainPanel.setLayout(framePanelLayout);
    framePanelLayout.setHorizontalGroup(
    framePanelLayout.createParallelGroup(Alignme
nt.LEADING)
    .addComponent(scrollPane,
    GroupLayout.DEFAULT_SIZE,
    GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );
    framePanelLayout.setVerticalGroup(
    framePanelLayout.createParallelGroup(Alignme
nt.LEADING)
    .addComponent(scrollPane,
    GroupLayout.DEFAULT_SIZE, 508,
    Short.MAX_VALUE)
    );

    pack();
}

private void
authorValueActionPerformed(ActionEvent evt) { }

private void overrideActions() {
    cs = new createSymbols() {
        @Override
        public void
        SymbolsButtonActionPerformed(ActionEvent evt,
        String bName, String bPath) {
            if (activeIndex != -1) {
                addImage(bName, bPath, "symbol", 0.0);
            }
        }
    };
    cs.initSymbols(super.leftSidePanel);

    cnt = new createTree(super.rightSidePanel,
    noOfChapters, chapTitle) {
        @Override
        public void valueChanged(TreeSelectionEvent
e) {
            DefaultMutableTreeNode node =
            (DefaultMutableTreeNode)navigatorTree.getLastSe
lectedPathComponent();

            if (node == null) return;

```

```

Object nodeSelected = node.getUserObject();
if (node.isLeaf()) {
    chapInfo chap = (chapInfo)nodeSelected;
    if (chap.index == 0) {
        scrollPane.setViewportView(preamblePanel
);
        activePanel = preamblePanel;
        activeIndex = -1;
    }
    else if (chap.index <= noOfChapters){
        scrollPane.setViewportView(chapter[chap.i
ndex-1].panel);
        activePanel = chapter[chap.index-1].panel;
        activeIndex = chap.index-1;
    }
    else {
        scrollPane.setViewportView(chapter[(int)Ma
th.floor(chap.index/100)].subsection.get(chap.inde
x%100-1));
        activePanel =
chapter[(int)Math.floor(chap.index/100)].subsectio
n.get(chap.index%100-1);
        activeIndex = chap.index;
    }
}
else {
    chapInfo chap = (chapInfo)nodeSelected;
    if (chap.index == 0) {
        scrollPane.setViewportView(preamblePan
el);
        activePanel = preamblePanel;
        activeIndex = -1;
    }
    else if (chap.index < 10){
        scrollPane.setViewportView(chapter[chap.
index-1].panel);
        activePanel = chapter[chap.index-
1].panel;
        activeIndex = chap.index-1;
    }
}
changeEditMenuCommands();
}
@Override
public void
addSubsectionButtonMouseClicked(MouseEvent
evt) {
    if (activeIndex < 1 || (activeIndex > 6 &&
activeIndex < 100)) {
        JOptionPane.showMessageDialog(null,
"Cannot add subsection here.", "Not Allowed",
JOptionPane.ERROR_MESSAGE);
    } else if (activeIndex < 100) {
        String title;
        if ((title =
JOptionPane.showInputDialog("Please enter title for
this Subsection. ")) != null) {
            if (!chapter[activeIndex].hasSubsection) {
                chapter[activeIndex].Subsection(null);
            }
            chapter[activeIndex].addSubsection(0,
title);
            super.addSubsecTree(activeIndex, title);
        } else {
            String title;
            if ((title =
JOptionPane.showInputDialog("Please enter title for
this Subsection. ")) != null) {

```

```

        chapter[(int)Math.floor(activeIndex/100)].a
ddSubsection(activeIndex%100, title);
        super.addSubsecTree(activeIndex, title);
    }
}
@Override
public void
remSubsectionButtonMouseClicked(MouseEvent
evt) {
    if (activeIndex < 100) {
        JOptionPane.showMessageDialog(null,
"Please select a Subsection.", "Not Allowed",
JOptionPane.ERROR_MESSAGE);
    } else {
        if (JOptionPane.showConfirmDialog(null, "Are
you sure you want to remove this
Subsection?")==0) {
            chapter[(int)Math.floor(activeIndex/100)].re
mSubsection(activeIndex);
            super.remTreeSubsections(activeIndex);
        }
    }
}
};
}

private void addSubsection(int index, String title)
{
    int chapNum = (int)Math.floor(index/100);
    if (!chapter[chapNum].hasSubsection) {
        chapter[chapNum].Subsection(null);
    }
    chapter[chapNum].addSubsection(0, title);
    super.addTreeSubsections(chapNum, title);
}

private void remAllSubsections() {
    for (int chapCnt=1; chapCnt<7; chapCnt++) {
        int noOfSubsecs =
chapter[1].subsecTitle.size();
        for (int secCnt=noOfSubsecs; secCnt>1;
secCnt--) {
            int index = secCnt+100;
            chapter[chapCnt].remSubsection(index);
        }
        super.remChapSubsections(chapCnt);
    }
}

private void setBibContents() {
    String text =
chapter[bibIndex].textPane.getText();
    text = text.substring(text.indexOf("<body>")
+6, text.indexOf("</body>"));
    int bibCnt = 0;
    String[] bib;

    bib = text.split("<br><br>");
    for(String b: bib) {
        if (b.contains("(")) {
            bibCnt =
Integer.parseInt(b.substring(b.indexOf("(")+1,
b.indexOf(")")));
            b = b.substring(b.indexOf(")")+2);
            citations.add(new Citation(bibCnt, b));
            b = b.replaceAll("<i>", "\\textit{");
            b = b.replaceAll("</i>", "}");
            b = b.replaceAll("<br>", " \\\\n");

```

```

        b = b.replaceAll("<url>", "\\url{");
        b = b.replaceAll("</url>", "}");
        texCitations.add(new Citation(bibCnt, b));
    }
}

private void
studNumValueActionPerformed(ActionEvent evt)
{ }

private void yearValueFocusGained(FocusEvent
evt) {
    if (yearValue.getText().equals("YYYY"))
        yearValue.setText("");
    monthValue.setEnabled(false);
}

class yearValueVerifier extends InputVerifier {
    public boolean verify(JComponent input) {
        JTextField tf = (JTextField) input;
        char value[] = tf.getText().trim().toCharArray();
        for (int i=0; i<value.length; i++) {
            if (Character.isDigit(value[i]) == false) {
                JOptionPane.showMessageDialog(null,
                "Invalid year value.", "ERROR!",
                JOptionPane.ERROR_MESSAGE);
                return false;
            }
        }
        if (value.length != 4) {
            JOptionPane.showMessageDialog(null, "Year
value must consist of 4 digits.", "ERROR!",
            JOptionPane.ERROR_MESSAGE);
            return false;
        }
        monthValue.setEnabled(true);
        return true;
    }
}

public static void insertInto(JEditorPane textPane,
String htmlStr, int caretPos) {
    try {
        //assumes editor is already set to "text/html"
        type
        HTMLToolkit kit = (HTMLToolkit)
textPane.getEditorKit();
        Document doc = textPane.getDocument();
        StringReader reader = new
StringReader(htmlStr);
        kit.read(reader, doc, caretPos);
    } catch (IOException ex) {
        Logger.getLogger(newProject.class.getName())
.log(Level.SEVERE, null, ex);
    } catch (BadLocationException ex) {
        Logger.getLogger(newProject.class.getName())
.log(Level.SEVERE, null, ex);
    }
}

public void addImage(String iName, String iPath,
String realWidth, Double width) {
    ImageIcon ic = new ImageIcon(iPath, iName);
    if (activeIndex != -1) {
        if (activeIndex < 100) {
            if (realWidth.charAt(0) == 'i') {
                insertInto(chapter[activeIndex].textPane,
"<img src='file:\\\\\\\\\\" + iPath + " " +

```

```

        "caption='"+iName+"'" +
        "realWidth='"+realWidth+"'" +
        "width='" +
Math.round(ic.getIconWidth()*width) + "' " +
        "height='" +
Math.round(ic.getIconHeight()*width) + "'><br>",
chapter[activeIndex].textPane.getCaret
Position());
    } else if (realWidth.charAt(0) == 't'){
        insertInto(chapter[activeIndex].textPane,
"<img src='file:\\\\\\\\\\" + iPath + " " +
        "caption='"+iName+"'" +
        "realWidth='"+realWidth+"'" +
        "width='" + Math.round(width) + "' " +
        "height='" +
Math.round((width*ic.getIconHeight())/ic.getIconWi
dth()) + "'><br>",
chapter[activeIndex].textPane.getCaret
Position());
    } else {
        insertInto(chapter[activeIndex].textPane,
"<img src='file:\\\\\\\\\\" + iPath + " " +
        "caption='"+ iName + "' " +
        "type='symbol'>",
chapter[activeIndex].textPane.getCaret
Position());
    }
    } else {
        if (realWidth.charAt(0) == 'i') {
            insertInto(chapter[(int)Math.floor(activeInde
x/100)].subTextPane.get(activeIndex%100-1),
"<br><img src='file:\\\\\\\\\\" + iPath + " " +
+
        "caption='"+iName+"'" +
        "realWidth='"+realWidth+"'" +
        "width='" +
Math.round(ic.getIconWidth()*width) + "' " +
        "height='" +
Math.round(ic.getIconHeight()*width) + "'><br>",
chapter[(int)Math.floor(activeIndex/100
)].subTextPane.get(activeIndex%100-
1).getCaretPosition());
        } else if (realWidth.charAt(0) == 't'){
            insertInto(chapter[(int)Math.floor(activeInde
x/100)].subTextPane.get(activeIndex%100-1),
"<br><img src='file:\\\\\\\\\\" + iPath + " " +
+
        "caption='"+iName+"'" +
        "realWidth='"+realWidth+"'" +
        "width='" + Math.round(width) + "' " +
        "height='" +
Math.round((width*ic.getIconHeight())/ic.getIconWi
dth()) + "'><br>",
chapter[(int)Math.floor(activeIndex/100
)].subTextPane.get(activeIndex%100-
1).getCaretPosition());
        } else {
            insertInto(chapter[(int)Math.floor(activeInde
x/100)].subTextPane.get(activeIndex%100-1),
"<img src='file:\\\\\\\\\\" + iPath + " " +
        "caption='"+ iName + "' " +
        "type='symbol'>",
chapter[(int)Math.floor(activeIndex/100
)].subTextPane.get(activeIndex%100-
1).getCaretPosition());
        }
    }
}
}
}
}

```

```

public void insertEquation() {
    showEquationFrame sef = new
showEquationFrame(new javax.swing.JFrame(),
true);
    sef.setVisible(true);
    if (sef.getReturnStatus() == 1) {
        equation = sef.getEquation();
        if (activeIndex != -1) {
            if(activeIndex < 100) {
                insertInto(chapter[activeIndex].textPane,
                "<img src='file:\\\\C:\\LaTeX
Symbols\\equation.jpg' " +
                "caption=\"" + equation + "\">",
                chapter[activeIndex].textPane.getCa
retPosition());
            } else {
                insertInto(chapter[(int)Math.floor(activeInde
x/100)].subTextPane.get(activeIndex%100-1),
                "<img src='file:\\\\C:\\LaTeX
Symbols\\equation.jpg' " +
                "caption=\"" + equation + "\">",
                chapter[activeIndex].textPane.getCa
retPosition());
            }
        }
    }
}

```

```

public void insertBullets(String type) {
    if (activeIndex != -1) {
        if(activeIndex < 100) {
            ulListFlag = true;
            int curCaretPos =
chapter[activeIndex].textPane.getCaretPosition();
            insertInto(chapter[activeIndex].textPane,
            "<" + type + "><li></li></" + type + ">",
            chapter[activeIndex].textPane.getCaret
Position());
            chapter[activeIndex].textPane.setCaretPositi
on(curCaretPos+1);

```

```

        newProject.insertInto(chapter[activeIndex].te
xtPane,"<input type=\"hidden\">",
curCaretPos+1);
        String origText =
chapter[activeIndex].textPane.getText();
        origText = origText.replaceFirst("</li>\s*</
[ou]>\s*<" + type + ">\s*<li>\s*</li>\s*</" + typ
e + ">\s*<input type=\"hidden\">\s*<[ou]>",
"<" + type + "><li></li></" + type + "></li>");
        origText = origText.replaceFirst("</li>\s*</
[ou]>\s*<input
type=\"hidden\">\s*<" + type + ">\s*<li>\s*</li>\
\s*</" + type + ">\s*<[ou]>",
"<" + type + "><li></li></" + type + "></li>");
        chapter[activeIndex].textPane.setText(origTe
xt);
        chapter[activeIndex].textPane.setCaretPositio
n(curCaretPos+1);
    } else {
        int index = (int)Math.floor(activeIndex/100);
        int rem = activeIndex%100-1;
        int curCaretPos =
chapter[index].subTextPane.get(rem).getCaretPositi
on();

```

```

        insertInto(chapter[index].subTextPane.get(re
m), "<" + type + "><li></li></" + type + ">",

```

```

chapter[index].subTextPane.get(rem).
getCaretPosition());

```

```

chapter[index].subTextPane.get(rem).setCare
tPosition(curCaretPos+1);

```

```

        newProject.insertInto(chapter[index].subTextP
ane.get(rem),"<input type=\"hidden\">",
curCaretPos+1);

```

```

        String origText =
chapter[index].subTextPane.get(rem).getText();
        origText = origText.replaceFirst("</li>\s*</
[ou]>\s*<" + type + ">\s*<li>\s*</li>\s*</" + typ
e + ">\s*<input type=\"hidden\">\s*<[ou]>",
"<" + type + "><li></li></" + type + "></li>");
        origText = origText.replaceFirst("</li>\s*</
[ou]>\s*<input
type=\"hidden\">\s*<" + type + ">\s*<li>\s*</li>\
\s*</" + type + ">\s*<[ou]>",
"<" + type + "><li></li></" + type + "></li>");

```

```

chapter[index].subTextPane.get(rem).setText
(origText);

```

```

chapter[index].subTextPane.get(rem).setCare
tPosition(curCaretPos+1);

```

```

    }
}

```

```

public void removeKeyListener(JEditorPane
textPane) {
    textPane.removeKeyListener(textPane.getKeyLis
teners()[0]);
}

```

```

@Override
public void addCitation() {
    if (activeIndex != -1) {
        if(citationCnt == 0) {
            newCitation c = new newCitation(new
javax.swing.JFrame(), true);
            c.setCitationCnt(citationCnt);
            c.setVisible(true);
            if (c.getReturnStatus() == 1) {
                citationCnt = citationCnt+1;

```

```

                if(activeIndex < 100)
                    insertInto(chapter[activeIndex].textPane,
                    "[" + citationCnt + "]",
                    chapter[activeIndex].textPane.getCa
retPosition());
                else
                    insertInto(chapter[(int)Math.floor(activeInd
ex/100)].subTextPane.get(activeIndex%100-1),
                    "[" + citationCnt + "]",
                    chapter[(int)Math.floor(activeIndex/1
00)].subTextPane.get(activeIndex%100-
1).getCaretPosition());

```

```

                    updateBib(citationCnt, c.getCitation());
                    citations.add(new Citation(citationCnt,
c.getCitation());
                    texCitations.add(new Citation(citationCnt,
c.getBibTex());
                }
            } else {

```

```

        showCitationsFrame scf = new
showCitationsFrame(new javax.swing.JFrame(),
true);
        scf.setRefList(citations);
        scf.setCitationCnt(citationCnt);
        scf.setVisible(true);
        if (scf.getReturnStatus() == 1) {
            if(activelIndex < 100)
                insertInto(chapter[activelIndex].textPane,
                "["+scf.getCitationCnt()+"]",
                chapter[activelIndex].textPane.getCa
retPosition());
            else
                insertInto(chapter[(int)Math.floor(activelInd
ex/100)].subTextPane.get(activelIndex%100-1),
                "["+scf.getCitationCnt()+"]",
                chapter[(int)Math.floor(activelIndex/1
00)].subTextPane.get(activelIndex%100-
1).getCaretPosition());
        }
        else if (scf.getReturnStatus() == 2) {
            citationCnt = citationCnt+1;
            if(activelIndex < 100)
                insertInto(chapter[activelIndex].textPane,
                "["+citationCnt+"]",
                chapter[activelIndex].textPane.getCa
retPosition());
            else
                insertInto(chapter[(int)Math.floor(activelInd
ex/100)].subTextPane.get(activelIndex%100-1),
                "["+citationCnt+"]",
                chapter[(int)Math.floor(activelIndex/1
00)].subTextPane.get(activelIndex%100-
1).getCaretPosition());

            updateBib(citationCnt, scf.getCitation());
            citations.add(new Citation(citationCnt,
scf.getCitation()));
            texCitations.add(new Citation(citationCnt,
scf.getBibTex()));
        } else if (scf.getReturnStatus() == 4) {
            citationCnt = scf.getCitationCnt();

            //remove citation in textpanes
            for (Chapter chap: chapter) {
                while
(chap.textPane.getText().contains("[ "+citationCnt
+"]"))
                    chap.textPane.setText(chap.textPane.getT
ext().replace("[ "+citationCnt+"]", ""));
                if (chap.hasSubsection) {
                    for (JEditorPane sp: chap.subTextPane) {
                        while
(sp.getText().contains("[ "+citationCnt+"]"))
                            sp.setText(sp.getText().replace("[ "+cita
tionCnt+"]", ""));
                    }
                }
            }

            //change citation numbers displayed in
textpanes
            for (int cCnt=citationCnt+1;
cCnt<=citations.size(); cCnt++) {
                for (Chapter chap: chapter) {
                    while
(chap.textPane.getText().contains("[ "+cCnt+"]"))
                        chap.textPane.setText(chap.textPane.ge
tText().replace("[ "+cCnt+"]", "["+(cCnt-1)+"]"));
                if (chap.hasSubsection) {
                    for (JEditorPane sp: chap.subTextPane) {
                        while
(sp.getText().contains("[ "+cCnt+"]"))
                            sp.setText(sp.getText().replace("[ "+c
Cnt+"]", "["+(cCnt-1)+"]"));
                    }
                }
            }

            //remove citation from list
            citations.remove(citationCnt-1);
            texCitations.remove(citationCnt-1);

            //change citation number for citations in list
            int cTemp = 0;
            for (int cCnt=0; cCnt<citations.size(); cCnt+
+) {
                if ((cTemp =
citations.get(cCnt).getCitationCnt()) > citationCnt)
                    citations.get(cCnt).setCitationCnt(cTemp-
1);
            }
            updateBib(citationCnt, "");
            citationCnt = citations.size();
        }
    }
}

private void updateBib(int citationCnt, String
citation) {
    if (citation.equals("")) {
        String text = "";
        for (Citation c: citations) {
            text += "[" + c.getCitationCnt() + "]" +
c.getCitation() + "<br><br>";
        }
        chapter[bibIndex].textPane.setText(text);
    } else {
        String text = "[" + citationCnt + "]" + citation
+ "<br><br>";
        insertInto(chapter[bibIndex].textPane, text,
chapter[bibIndex].textPane.getCaretPosition());
    }
}

public void changeEditMenuCommands()
{
    if (activelIndex != -1) {
        if(activelIndex < 100) {
            manager = chapter[activelIndex].manager;
        } else {
            manager =
chapter[(int)Math.floor(activelIndex/100)].subMana
ger.get(activelIndex%100-1);
        }
    } else {
        manager = preambleManager;
    }

    undoCommand.setAction(UndoManagerHelper.g
etUndoAction(manager));
    redoCommand.setAction(UndoManagerHelper.g
etRedoAction(manager));
}

```

```

        undoCommand.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_Z, InputEvent.CTRL_MASK));
        redoCommand.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_Z, InputEvent.SHIFT_MASK | InputEvent.CTRL_MASK));
        cutCommand.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_X, InputEvent.CTRL_MASK));
        copyCommand.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_C, InputEvent.CTRL_MASK));
        pasteCommand.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_V, InputEvent.CTRL_MASK));
        selectAllCommand.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK_A, InputEvent.CTRL_MASK));
        cutCommand.setText("Cut");
        copyCommand.setText("Copy");
        pasteCommand.setText("Paste");
        selectAllCommand.setText("Select All");
    }

    public void saveProjectToFile(File file) {
        FileOutputStream out; // declare a file output object
        PrintStream p; // declare a print stream object
        try
        {
            String text="";
            text += "#NOOFCHAPTERS:" + noOfChapters
+ "\n" +
            "#CITATIONCNT:" + citationCnt + "\n" +
            "#BIBINDEX:" + bibIndex + "\n" +
            "#TITLE:" + titleValue.getText() + "\n" +
            "#AUTHOR:" + authorValue.getText() +
"\n" +
            "#STUDNUM:" + studNumValue.getText()
+ "\n" +
            "#MONTH:" +
monthValue.getSelectedIndex() + "\n" +
            "#YEAR:" + yearValue.getText() + "\n" +
            "#ADVISER:" + adviserVal.getText() +
"\n" +
            "#DEPTCHAIR:" + deptChairVal.getText()
+ "\n" +
            "#UNITHEAD:" + unitHeadVal.getText() +
"\n" +
            "#DEAN:" + deanVal.getText() + "\n" +
            "#DATE:" + asDateVal.getText() + "\n" +
            "#KEYWORDS:" + keywordsVal.getText()
+ "\n";

            if (!panel1Val.getText().trim().isEmpty())
                text += "#PANELIST:" + panel1Val.getText()
+ "\n";
            if (!panel2Val.getText().trim().isEmpty())
                text += "#PANELIST:" + panel2Val.getText()
+ "\n";
            if (!panel3Val.getText().trim().isEmpty())
                text += "#PANELIST:" + panel3Val.getText()
+ "\n";
            if (!panel4Val.getText().trim().isEmpty())
                text += "#PANELIST:" + panel4Val.getText()
+ "\n";
            if (!panel5Val.getText().trim().isEmpty())
                text += "#PANELIST:" + panel5Val.getText()
+ "\n";
            if (!panel6Val.getText().trim().isEmpty())
                text += "#PANELIST:" + panel6Val.getText()
+ "\n";
            if (!panel7Val.getText().trim().isEmpty())

```

```

                text += "#PANELIST:" + panel7Val.getText()
+ "\n";
            if (!panel8Val.getText().trim().isEmpty())
                text += "#PANELIST:" + panel8Val.getText()
+ "\n";
            if (!panel9Val.getText().trim().isEmpty())
                text += "#PANELIST:" + panel9Val.getText()
+ "\n";
            if (!panel10Val.getText().trim().isEmpty())
                text += "#PANELIST:" + panel10Val.getText()
+ "\n";

            text += "#ABSTRACT-----" + "\n";

            for (Chapter chap: chapter) {
                text += chap.textPane.getText();
                if (chap.hasSubsection) {
                    int ctr = 0;
                    for (JEditorPane sub: chap.subTextPane) {
                        text += "#SUB-----" +
chap.subsecTitle.get(ctr) + "\n";
                        text += sub.getText();
                        ctr++;
                    }
                    text += "#-----" + "\n";
                }
            }
            // Create a new file output stream connected to
            given file
            out = new FileOutputStream(file);
            // Connect print stream to the output stream
            p = new PrintStream( out );
            p.println( text);
            p.close();
            JOptionPane.showMessageDialog(null, "Project
saved to "+file.getPath(), "Project Saved",
JOptionPane.PLAIN_MESSAGE);
        }
        catch (Exception e)
        {
            JOptionPane.showMessageDialog(null,"Error
writing to file.", "",JOptionPane.PLAIN_MESSAGE);
        }
    }

    public void openProjectFromFile(File file) {
        // Invoke the readFile method in this class
        String line;
        //remove current subsections
        removeAllSubsections();

        try {
            FileReader in = new FileReader(file);
            BufferedReader dis = new BufferedReader(in);

            int panelCnt = 0;
            int chapCnt = 1;
            String chapText = "";
            while ((line = dis.readLine()) != null) {
                if (line.contains("#NOOFCHAPTERS:")
                    noOfChapters =
Integer.parseInt(line.substring(14));
                else if (line.contains("#CITATIONCNT:")
                    citationCnt =
Integer.parseInt(line.substring(13));
                else if (line.contains("#BIBINDEX:")
                    bibIndex =
Integer.parseInt(line.substring(10));
                else if (line.contains("#TITLE:")

```



```

        titleValue.setText(line.substring(7));
        else if (line.contains("#AUTHOR:"))
            authorValue.setText(line.substring(8));
        else if (line.contains("#STUDNUM:"))
            studNumValue.setText(line.substring(9));
        else if (line.contains("#MONTH:"))
            monthValue.setSelectedItem(line.substring(
7));
        else if (line.contains("#YEAR:"))
            yearValue.setText(line.substring(6));
        else if (line.contains("#ADVISER:"))
            adviserVal.setText(line.substring(9));
        else if (line.contains("#DEPTCHAIR:"))
            deptChairVal.setText(line.substring(11));
        else if (line.contains("#UNITHEAD:"))
            unitHeadVal.setText(line.substring(10));
        else if (line.contains("#DEAN:"))
            deanVal.setText(line.substring(6));
        else if (line.contains("#DATE:"))
            asDateVal.setText(line.substring(6));
        else if (line.contains("#KEYWORDS:"))
            keywordsVal.setText(line.substring(10));
        else if (line.contains("#PANELIST:")) {
            panelCnt++;
            switch (panelCnt) {
                case 1:
                    panel1Val.setText(line.substring(10)); break;
                case 2:
                    panel2Val.setText(line.substring(10)); break;
                case 3:
                    panel3Val.setText(line.substring(10)); break;
                case 4:
                    panel4Val.setText(line.substring(10)); break;
                case 5:
                    panel5Val.setText(line.substring(10)); break;
                case 6:
                    panel6Val.setText(line.substring(10)); break;
                case 7:
                    panel7Val.setText(line.substring(10)); break;
                case 8:
                    panel8Val.setText(line.substring(10)); break;
                case 9:
                    panel9Val.setText(line.substring(10)); break;
                case 10:
                    panel10Val.setText(line.substring(10)); break;
            }
        } else if (line.contains("#ABSTRACT-----")) {
            while (!(line =
dis.readLine()).contains("#-----")) {
                chapText += line;
            }
            if (line.contains("#-----")) {
                chapter[0].textPane.setText(chapText);
                chapText = "";
            }
            while ((line = dis.readLine()) != null) {
                if (chapCnt != noOfChapters) {
                    if ((line.contains("#-----")) ||
(line.contains("#SUB-----"))) {
                        chapter[chapCnt].textPane.setText(chap
Text);
                        chapText = "";
                        if (line.contains("#SUB-----")) { //
(chapCnt == 1) {
                            int subCnt = 0;
                            String subTxt = "";;
                            int index = chapCnt*100+subCnt+1;
                            addSubsection(index,
line.substring(10));

```

```

                    while (!(line =
dis.readLine()).contains("#-----")) {
                        if (line.contains("#SUB-----")) {
                            addSubsection(index,
line.substring(10));
                            chapter[chapCnt].subTextPane.get(s
ubCnt).setText(subTxt);
                            subTxt = "";;
                            subCnt++;
                        } else {
                            subTxt += line;
                        }
                        chapter[chapCnt].subTextPane.get(su
bCnt).setText(subTxt);
                    }
                    chapCnt++;
                } else {
                    chapText += line;
                }
            }
        }
        in.close();
        setBibContents();
    } catch (IOException e) {
    }
}

```

```

public void CompileAndRunProject(int viewFlag){
    int status = 1;
    compileAndRun car = new compileAndRun();
    car.setTitlePage(titleValue.getText(),
authorValue.getText(), studNumValue.getText(),
monthValue.getSelectedItem().toString(),
yearValue.getText());

```

```

        ArrayList<String> panelists = new
ArrayList<String>();
        if (!panel1Val.getText().trim().isEmpty())
            panelists.add(panel1Val.getText());
        if (!panel2Val.getText().trim().isEmpty())
            panelists.add(panel2Val.getText());
        if (!panel3Val.getText().trim().isEmpty())
            panelists.add(panel3Val.getText());
        if (!panel4Val.getText().trim().isEmpty())
            panelists.add(panel4Val.getText());
        if (!panel5Val.getText().trim().isEmpty())
            panelists.add(panel5Val.getText());
        if (!panel6Val.getText().trim().isEmpty())
            panelists.add(panel6Val.getText());
        if (!panel7Val.getText().trim().isEmpty())
            panelists.add(panel7Val.getText());
        if (!panel8Val.getText().trim().isEmpty())
            panelists.add(panel8Val.getText());
        if (!panel9Val.getText().trim().isEmpty())
            panelists.add(panel9Val.getText());
        if (!panel10Val.getText().trim().isEmpty())
            panelists.add(panel10Val.getText());

```

```

        car.setAcceptanceSheet(adviserVal.getText(),
unitHeadVal.getText(), deptChairVal.getText(),
deanVal.getText(), asDateVal.getText(),
panelists);

```

```

        car.setKeywords(keywordsVal.getText());

```

```

        car.convertProjectToTeX();

```

```

    for (Chapter chap: chapter) {
        if (chap == chapter[bibIndex]) {
            car.setBib(texCitations);
        } else {
            car.setChapter(chap.chapTitle,
                chap.textPane.getText());
            if (chap.hasSubsection)
                car.setSubSection(chap.subsecTitle,
                    chap.subTextPane);
        }
    }
    car.endDoc();

    String tex = car.getText();

    if (project == null) {
        projectFileChooser pfc = new
projectFileChooser();
        status = pfc.chooseFile();
        if (status == 1)
            return;
        project = pfc.getFile();
        saveProjectToFile(project);
    }

    saveToTeXFile sttf = new saveToTeXFile();
    status = sttf.saveAsTeX(project, tex);
    File TeXFile = sttf.getTeXFile();
    if (status == 1)
        return;
    car.run(TeXFile, viewFlag);
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new
Runnable() {
        public void run() {
            //Turn off metal's use of bold fonts
            UIManager.put("swing.boldMetal",
                Boolean.FALSE);
            new newProject(0).setVisible(true);
        }
    });
}

private JPanel aSheetPanel;
private JPanel abstractPanel;
private JLabel authorLabel;
protected JTextField authorValue;
private JLabel dateLabel;
protected JComboBox monthValue;
private JPanel preamblePanel;
private JScrollPane scrollPane;
private JLabel studNumLabel;
protected JTextField studNumValue;
private JLabel titleLabel;
private JPanel titlePagePanel;
private JScrollPane titleScrollPane;
protected javax.swing.JTextPane titleValue;
protected JTextField yearValue;
private JLabel adviserLbl;
private JTextField adviserVal;
private JLabel deptChairLbl;
private JTextField deptChairVal;
private JLabel jLabel1;
private JLabel jLabel2;
private JLabel jLabel3;
private JLabel jLabel4;
private JLabel jLabel5;

```

```

private JLabel jLabel6;
private JLabel jLabel7;
private JLabel jLabel8;
private JLabel jLabel9;
private JLabel jLabel10;
private JTextField panel1Val;
private JTextField panel2Val;
private JTextField panel3Val;
private JTextField panel4Val;
private JTextField panel5Val;
private JTextField panel6Val;
private JTextField panel7Val;
private JTextField panel8Val;
private JTextField panel9Val;
private JTextField panel10Val;
private JLabel panelistsLbl;
private JLabel asDateLbl;
private JTextField asDateVal;
private JLabel deanLbl;
private JTextField deanVal;
private JLabel unitHeadLbl;
private JTextField unitHeadVal;
private JLabel keywordsLbl;
private JTextField keywordsVal;
}

```

**/\*\* projectFileChooser.java \*/**

```

package upmguilte;

import java.awt.Component;
import java.io.File;
import java.io.IOException;

import javax.swing.JFileChooser;
import javax.swing.JOptionPane;

class projectFileChooser {
    private String filename;
    private JFileChooser fc;
    private readFile rf = new readFile();
    File file;

    public projectFileChooser() {}

    public int chooseFile () {
        filename = File.separator + "tmp";
        fc = new JFileChooser(new File(filename));
        fc.setCurrentDirectory(new File("."));
        fc.addChoosableFileFilter(new
ProjectFileFilter());
        int returnVal = fc.showSaveDialog(new
Component() {
            private static final long serialVersionUID =
1L;});

        if (returnVal == JFileChooser.APPROVE_OPTION)
        {
            file = fc.getSelectedFile();
            String file_string = rf.readContents(file);

            if (file_string == null) {
                try {
                    file = new File(file.getPath());
                    if(file.getPath().endsWith(".glp")) {}
                } else {
                    file = new File(file.getPath()+".glp");
                }
                file.createNewFile();
            }
        }
    }
}

```

```

        } catch (IOException ex) {
            JOptionPane.showMessageDialog(null, "Error
creating file.", "Error!",
JOptionPane.ERROR_MESSAGE);
            return 1;
        }
    }
    String fileTypeNames =
fc.getTypeDescription(file);

    if (fileTypeNames.equals("GLP File")) {
        return 0;
    } else {
        JOptionPane.showMessageDialog(null, "Invalid
file! Choose GUI-LaTeX Project files only.", "Error!",
JOptionPane.ERROR_MESSAGE);
    }
    } else {
        JOptionPane.showMessageDialog(null, "Saving
project
cancelled.", "", JOptionPane.PLAIN_MESSAGE);
    }
    return 1;
}

File getFile() {
    return file;
}
}

```

/\*\* ProjectFileFilter.java \*/

```

package upmguilte;

import java.io.File;

//create filter for GUI-LaTeX Projects only
public class ProjectFileFilter extends
javax.swing.filechooser.FileFilter {

    public boolean accept(File f) {
        return f.isDirectory() ||
f.getName().toLowerCase().endsWith(".glp");
    }

    public String getDescription() {
        return "GUI-LaTeX Project files(*.glp)";
    }
}

```

/\*\* readFile.java \*/

```

package upmguilte;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;

class readFile {
    public readFile() {}

    //read contents of the file
    public String readContents(File file) {
        StringBuffer fileBuffer;
        String fileString = null;
        String line;
        try {
            FileReader in = new FileReader(file);

```

```

        BufferedReader dis = new BufferedReader(in);
        fileBuffer = new StringBuffer();
        while ((line = dis.readLine()) != null) {
            fileBuffer.append(line + "\n");
        }
        in.close();
        fileString = fileBuffer.toString();
    } catch (IOException e) {
        return null;
    }
    return fileString;
}
}

```

/\*\* saveToTeXFile.java \*/

```

package upmguilte;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.PrintStream;

import javax.swing.JOptionPane;

class saveToTeXFile {
    File file;

    public saveToTeXFile() {}

    public int saveAsTeX(File file, String tex) {
        try {
            file = new File(file.getPath());
            if (file.getPath().endsWith(".glp")) {
                file = new File(file.getPath().replace(".glp",
".tex"));
            } else {
                file = new File(file.getPath().replace(".GLP",
".tex"));
            }
            file.createNewFile();
            this.file = file;
        } catch (IOException ex) {
            JOptionPane.showMessageDialog(null, "Error
creating file for preview.", "Error!",
JOptionPane.ERROR_MESSAGE);
            return 1;
        }
        try {
            saveToFile(file, tex);
            return 0;
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, "Error
creating file for preview.", "Error!",
JOptionPane.ERROR_MESSAGE);
            return 1;
        }
    }

    private void saveToFile(File file, String tex) {
        FileOutputStream out;
        PrintStream p;
        try {
            // Create a new file output stream connected to
given file
            out = new FileOutputStream(file);
            // Connect print stream to the output stream
            p = new PrintStream(out);
            p.println(tex);

```

```

        p.close();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null,"Error
writing to file.", "",JOptionPane.PLAIN_MESSAGE);
    }
}

public File getTeXFile() {
    return file;
}

public File getDVIFile() {
    return new File(file.getPath().replace(".tex",
".dvi"));
}
}

/** showCitationsFrame.java */

package upmguilte;

import java.awt.Component;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.util.Vector;

import javax.swing.AbstractListModel;
import javax.swing.GroupLayout;
import javax.swing.GroupLayout.Alignment;
import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JList;
import javax.swing.JOptionPane;
import javax.swing.JScrollPane;
import javax.swing.LayoutStyle.ComponentPlacement;
import javax.swing.ListSelectionModel;
import javax.swing.SwingConstants;

public class showCitationsFrame extends JDialog {

    private static final long serialVersionUID = 1L;
    public static final int RET_CANCEL = 0;
    public static final int RET_OK = 1;
    private int citationCnt = 0;
    private String citation = "";
    private String texCitation = "";

    public showCitationsFrame(java.awt.Frame
parent, boolean modal) {
        super(parent, modal);
        initComponents();
    }

    public int getReturnStatus() {
        return returnStatus;
    }

    public String getCitation() {
        return citation;
    }

    public String getBibTex() {
        return texCitation;
    }
}

```

```

int getCitationCnt() {
    return citationCnt;
}

void setCitationCnt(int citationCnt) {
    this.citationCnt = citationCnt;
}

void setRefList(ArrayList<Citation> citations) {
    Vector<String> v = new Vector<String>();
    String s = "";
    for (Citation c: citations) {
        s = c.getCitation().replaceAll("<i>", "");
        s = s.replaceAll("</i>", "");
        s = s.replaceAll("<br>", "");
        s = s.replaceAll("<url>", "");
        s = s.replaceAll("</url>", "");
        v.add(c.getCitationCnt() + ". " + s);
    }
    refList.setListData(v);
}

private void initComponents() {
    okButton = new JButton();
    cancelButton = new JButton();
    jScrollPane1 = new JScrollPane();
    refList = new JList();
    jLabel1 = new JLabel();
    newCitationButton = new JButton();
    delButton = new JButton();

    addWindowListener(new
java.awt.event.WindowAdapter() {
        public void
windowClosing(java.awt.event.WindowEvent evt) {
            closeDialog(evt);
        }
    });

    okButton.setText("OK");
    okButton.addActionListener(new
ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            okButtonActionPerformed(evt);
        }
    });

    cancelButton.setText("Cancel");
    cancelButton.addActionListener(new
ActionListener() {
        public void actionPerformed(ActionEvent evt) {
            cancelButtonActionPerformed(evt);
        }
    });

    refList.setModel(new AbstractListModel() {
        private static final long serialVersionUID = 1L;
        String[] strings = { "Item 1", "Item 2", "Item 3",
"Item 4", "Item 5" };
        public int getSize() { return strings.length; }
        public Object getElementAt(int i) { return
strings[i]; }
    });
    refList.setSelectionMode(ListSelectionModel.SIN
GLE_SELECTION);
    jScrollPane1.setViewportView(refList);

    jLabel1.setText("Choose an existing
reference:");
}

```

```

        newCitationButton.setText("New Citation");
        newCitationButton.addActionListener(new
ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        newCitationButtonActionPerformed(evt);
    }
});

        delButton.setText("Delete Citation");
        delButton.addActionListener(new
ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        delButtonActionPerformed(evt);
    }
});

        GroupLayout layout = new
GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(delButton)
                    .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(newCitationButton)
                    .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(okButton,
GroupLayout.PREFERRED_SIZE, 67,
GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(cancelButton))
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jScrollPane1,
GroupLayout.PREFERRED_SIZE, 347,
GroupLayout.PREFERRED_SIZE)))
            .addContainerGap(22, Short.MAX_VALUE));

        layout.linkSize(SwingConstants.HORIZONTAL,
new Component[] {cancelButton, okButton});

        layout.setVerticalGroup(
            layout.createParallelGroup(Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addComponent(jScrollPane1,
GroupLayout.PREFERRED_SIZE, 83,
GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(LayoutStyle.ComponentPlacement.RELATED, 19, Short.MAX_VALUE)

```

```

                .addGroup(layout.createParallelGroup(Alignment.BASELINE)
                    .addComponent(okButton)
                    .addComponent(cancelButton)
                    .addComponent(newCitationButton)
                    .addComponent(delButton))
                .addGap());

        pack();
    }

    private void
okButtonActionPerformed(ActionEvent evt) {
    if (refList.getSelectedIndex() == -1)
        JOptionPane.showMessageDialog(null, "Select
an entry from the list or create a new reference.",
"ERROR!", JOptionPane.ERROR_MESSAGE);
    else {
        citationCnt = refList.getSelectedIndex() + 1;
        doClose(RET_OK);
    }
}

    private void
cancelButtonActionPerformed(ActionEvent evt) {
    doClose(RET_CANCEL);
}

    /** Closes the dialog */
    private void
closeDialog(java.awt.event.WindowEvent evt) {
    doClose(RET_CANCEL);
}

    private void
newCitationButtonActionPerformed(ActionEvent
evt) {
    setVisible(false);
    newCitation c = new newCitation(new JFrame(),
true);
    c.setCitationCnt(citationCnt);
    c.setVisible(true);
    if (c.getReturnStatus() == 1) {
        citation = c.getCitation();
        texCitation = c.getBibTex();
        returnStatus = 2;
        doClose(returnStatus);
    } else {
        returnStatus = 3;
        setVisible(true);
    }
}

    private void
delButtonActionPerformed(ActionEvent evt) {
    if (refList.getSelectedIndex() == -1)
        JOptionPane.showMessageDialog(null, "Select
an entry to be deleted from the list.", "ERROR!",
JOptionPane.ERROR_MESSAGE);
    else {
        citationCnt = refList.getSelectedIndex() + 1;
        doClose(4);
    }
}

    private void doClose(int retStatus) {
    returnStatus = retStatus;
    setVisible(false);
}

```

```

        dispose();
    }

    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new
Runnable() {
            public void run() {
                showCitationsFrame dialog = new
showCitationsFrame(new JFrame(), true);
                dialog.addWindowListener(new
java.awt.event.WindowAdapter() {
                    public void
windowClosing(java.awt.event.WindowEvent e) {
                        System.exit(0);
                    }
                });
                dialog.setVisible(true);
            }
        });
    }

    private JButton cancelButton;
    private JButton delButton;
    private JLabel jLabel1;
    private JScrollPane jScrollPane1;
    private JButton newCitationButton;
    private JButton okButton;
    private JList refList;
    private int returnStatus = RET_CANCEL;
}

/** showEquationFrame.java */

package upmguilte;

import java.awt.Component;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.GroupLayout;
import javax.swing.GroupLayout.Alignment;
import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.LayoutStyle.ComponentPlacement;
import javax.swing.SwingConstants;

public class showEquationFrame extends JDialog {

    private static final long serialVersionUID = 1L;
    public static final int RET_CANCEL = 0;
    public static final int RET_OK = 1;
    private String equation;

    public showEquationFrame(java.awt.Frame
parent, boolean modal) {
        super(parent, modal);
        initComponents();
    }

    public int getReturnStatus() {
        return returnStatus;
    }
}

```

```

    public String getEquation() {
        return equation;
    }

    private void initComponents() {
        okButton = new JButton();
        cancelButton = new JButton();
        jLabel1 = new JLabel();
        jScrollPane1 = new JScrollPane();
        eqTextArea = new JTextArea();

        setDefaultCloseOperation(false);
        addWindowListener(new
java.awt.event.WindowAdapter() {
            public void
windowClosing(java.awt.event.WindowEvent evt) {
                closeDialog(evt);
            }
        });

        okButton.setText("OK");
        okButton.addActionListener(new
ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                okButtonActionPerformed(evt);
            }
        });

        cancelButton.setText("Cancel");
        cancelButton.addActionListener(new
ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                cancelButtonActionPerformed(evt);
            }
        });

        jLabel1.setText("Paste Latex Equation Command
here:");

        eqTextArea.setColumns(20);
        eqTextArea.setRows(5);
        eqTextArea.addMouseListener(new
java.awt.event.MouseAdapter() {
            public void
mouseClicked(java.awt.event.MouseEvent evt) {
                eqTextAreaMouseClicked(evt);
            }
        });
        eqTextArea.addKeyListener(new
java.awt.event.KeyAdapter() {
            public void keyTyped(java.awt.event.KeyEvent
evt) {
                eqTextAreaKeyTyped(evt);
            }
        });
        jScrollPane1.setViewportViewView(eqTextArea);

        GroupLayout layout = new
GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGroup(layout.createParallelGroup(Alignment.LEADING)
                        .add(okButton)
                        .add(cancelButton)
                        .add(jLabel1)
                    )
                    .addContainerGap(100, true)
                )
        );
    }
}

```

```

        .addComponent(okButton,
GroupLayout.PREFERRED_SIZE, 67,
GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(ComponentPlacement.RE
LATED)
        .addComponent(cancelButton))
        .addComponent(jLabel1)
        .addComponent(jScrollPane1,
GroupLayout.DEFAULT_SIZE, 380,
Short.MAX_VALUE))
        .addContainerGap()
    );

    layout.linkSize(SwingConstants.HORIZONTAL,
new Component[] {cancelButton, okButton});

    layout.setVerticalGroup(
        layout.createParallelGroup(Alignment.LEADING
    )
        .addGroup(Alignment.TRAILING,
layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel1)
        .addPreferredGap(ComponentPlacement.RELA
TED)
        .addComponent(jScrollPane1,
GroupLayout.PREFERRED_SIZE, 119,
GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(ComponentPlacement.RELA
TED, 18, Short.MAX_VALUE)
        .addGroup(layout.createParallelGroup(Alignm
ent.BASELINE)
        .addComponent(cancelButton)
        .addComponent(okButton))
        .addContainerGap()
    );

    pack();
}

private void
okButtonActionPerformed(ActionEvent evt) {
    if (eqTextArea.getText().trim().isEmpty())
        JOptionPane.showMessageDialog(null, "Please
paste the equation in the text area.", "ERROR!",
JOptionPane.ERROR_MESSAGE);
    else
        doClose(RET_OK);
}

private void
cancelButtonActionPerformed(ActionEvent evt) {
    doClose(RET_CANCEL);
}

private void
closeDialog(java.awt.event.WindowEvent evt) {
    doClose(RET_CANCEL);
}

private void
eqTextAreaMouseClicked(java.awt.event.MouseEve
nt evt) {
    equation = eqTextArea.getText();
    String[] lines = equation.split("\n");
    equation = "";

    for (String line: lines){
        if (!line.isEmpty())

```

```

        if (line.charAt(0) != '%')
            equation += line;
        }
    }
    eqTextArea.setText(equation);
}

private void
eqTextAreaKeyTyped(java.awt.event.KeyEvent
evt) { //GEN-FIRST:event_eqTextAreaKeyTyped
    equation = eqTextArea.getText();
    String[] lines = equation.split("\n");
    equation = "";
    for (String line: lines){
        if (!line.isEmpty())
            if (line.charAt(0) != '%')
                equation += line;
        }
    }
    eqTextArea.setText(equation);
}

private void doClose(int retStatus) {
    returnStatus = retStatus;
    setVisible(false);
    dispose();
}

public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new
Runnable() {
        public void run() {
            showEquationFrame dialog = new
showEquationFrame(new JFrame(), true);
            dialog.addWindowListener(new
java.awt.event.WindowAdapter() {
                public void
windowClosing(java.awt.event.WindowEvent e) {
                    System.exit(0);
                }
            });
            dialog.setVisible(true);
        }
    });
}

private JButton cancelButton;
private JTextArea eqTextArea;
private JLabel jLabel1;
private JScrollPane jScrollPane1;
private JButton okButton;
private int returnStatus = RET_CANCEL;
}

/** showImageFrame.java */

package upmguilte;

import java.awt.Component;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;

import javax.swing.GroupLayout;
import javax.swing.GroupLayout.Alignment;
import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JFileChooser;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

```

```

import javax.swing.JRadioButton;
import javax.swing.JTextField;
import
javax.swing.LayoutStyle.ComponentPlacement;
import javax.swing.SwingConstants;
import javax.swing.WindowConstants;

public class showImageFrame extends JDialog {

    private static final long serialVersionUID = 1L;
    public static final int RET_CANCEL = 0;
    public static final int RET_OK = 1;
    public Double size = 0.0;
    public String texSize = "";

    public showImageFrame(java.awt.Frame parent,
boolean modal) {
        super(parent, modal);
        initComponents();
    }

    showImageFrame() {
        throw new UnsupportedOperationException("Not
yet implemented");
    }

    public int getReturnStatus() {
        return returnStatus;
    }

    private void initComponents() {

        okButton = new JButton();
        cancelButton = new JButton();
        imgPathLbl = new JLabel();
        iInflationLbl = new JLabel();
        browseButton = new JButton();
        iPathVal = new JTextField();
        twRButton = new JRadioButton();
        iwRButton = new JRadioButton();
        iCaptionLbl = new JLabel();
        iCaptionVal = new JTextField();
        iInflationVal = new JTextField();

        setDefaultCloseOperation(WindowConstants.DO
_NOTHING_ON_CLOSE);
        setBackground(new java.awt.Color(79, 76, 60));
        setCursor(new
java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR
));
        addWindowListener(new
java.awt.event.WindowAdapter() {
            public void
windowClosing(java.awt.event.WindowEvent evt) {
                closeDialog(evt);
            }
        });

        okButton.setText("OK");
        okButton.addActionListener(new
ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                okButtonActionPerformed(evt);
            }
        });

        cancelButton.setText("Cancel");
        cancelButton.addActionListener(new
ActionListener() {

```

```

            public void actionPerformed(ActionEvent evt) {
                cancelButtonActionPerformed(evt);
            }
        });

        imgPathLbl.setText("File path");

        iInflationLbl.setText("Image size");

        browseButton.setText("Browse");
        browseButton.addActionListener(new
ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                browseButtonActionPerformed(evt);
            }
        });

        iPathVal.setDisabledTextColor(new
java.awt.Color(153, 135, 118));
        iPathVal.setEnabled(false);

        twRButton.setSelected(true);
        twRButton.setText("% text width");
        twRButton.addActionListener(new
ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                twRButtonActionPerformed(evt);
            }
        });

        iwRButton.setText("% image width");
        iwRButton.addActionListener(new
ActionListener() {
            public void actionPerformed(ActionEvent evt) {
                iwRButtonActionPerformed(evt);
            }
        });

        iCaptionLbl.setText("Caption");

        iInflationVal.setText("100");

        GroupLayout layout = new
GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(Alignment.LEADING
)
                .addGroup(layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addGroup(layout.createParallelGroup(Alignm
ent.TRAILING)
                        .addComponent(iCaptionLbl)
                        .addComponent(imgPathLbl)
                        .addComponent(iInflationLbl))
                    .addGap(10, 10, 10)
                    .addGroup(layout.createParallelGroup(Alignm
ent.TRAILING)
                        .addComponent(iCaptionVal,
Alignment.LEADING, GroupLayout.DEFAULT_SIZE,
234, Short.MAX_VALUE)
                        .addComponent(iPathVal,
Alignment.LEADING, GroupLayout.DEFAULT_SIZE,
234, Short.MAX_VALUE)
                        .addComponent(okButton,
GroupLayout.PREFERRED_SIZE, 67,
GroupLayout.PREFERRED_SIZE)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(iInflationVal)

```



```

        .addComponent(iInflationVal,
GroupLayout.DEFAULT_SIZE, 30,
Short.MAX_VALUE)
        .addGap(18, 18, 18)
        .addComponent(twRButton)
        .addPreferredGap(ComponentPlacement.RE
LATED)
        .addComponent(iwRButton))
        .addPreferredGap(ComponentPlacement.UNR
ELATED)
        .addGroup(layout.createParallelGroup(Alignm
ent.LEADING)
        .addComponent(browseButton)
        .addComponent(cancelButton))
        .addContainerGap()
    );

    layout.linkSize(SwingConstants.HORIZONTAL,
new Component[] {cancelButton, okButton});

    layout.setVerticalGroup(
        layout.createParallelGroup(Alignment.LEADING
)
        .addGroup(layout.createSequentialGroup()
        .addGap(17, 17, 17)
        .addGroup(layout.createParallelGroup(Alignm
ent.BASELINE)
        .addComponent(imgPathLbl)
        .addComponent(iPathVal,
GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE)
        .addComponent(browseButton))
        .addGap(10, 10, 10)
        .addGroup(layout.createParallelGroup(Alignm
ent.BASELINE)
        .addComponent(iCaptionLbl,
GroupLayout.PREFERRED_SIZE, 14,
GroupLayout.PREFERRED_SIZE)
        .addComponent(iCaptionVal,
GroupLayout.PREFERRED_SIZE, 20,
GroupLayout.PREFERRED_SIZE))
        .addGap(11, 11, 11)
        .addGroup(layout.createParallelGroup(Alignm
ent.BASELINE)
        .addComponent(iInflationLbl,
GroupLayout.PREFERRED_SIZE, 14,
GroupLayout.PREFERRED_SIZE)
        .addComponent(iwRButton)
        .addComponent(twRButton)
        .addComponent(iInflationVal,
GroupLayout.PREFERRED_SIZE,
GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
        .addGap(15, 15, 15)
        .addGroup(layout.createParallelGroup(Alignm
ent.TRAILING, false)
        .addComponent(cancelButton,
GroupLayout.DEFAULT_SIZE,
GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(okButton,
GroupLayout.DEFAULT_SIZE,
GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addContainerGap()
    );

    pack();
}

```

```

private void
okButtonActionPerformed(ActionEvent evt) { //GEN-
FIRST:event_okButtonActionPerformed
    try{
        if (this.iPathVal.getText().contains(" ")) {
            JOptionPane.showMessageDialog(null, "File
path must not contain spaces. \nMake sure parent
folders and filename does not contain spaces.",
"ERROR!", JOptionPane.ERROR_MESSAGE);
        } else if (iInflationVal.getText().isEmpty() ||
iPathVal.getText().isEmpty()) {
            JOptionPane.showMessageDialog(null, "Please
enter image path and inflation.", "Missing
information", JOptionPane.ERROR_MESSAGE);
        } else if (iInflationVal.getText().length()>3 ||
Integer.parseInt(iInflationVal.getText())==0) {
            JOptionPane.showMessageDialog(null,
"Inflation value invalid. Enter values between 0
and 999.", "Invalid input",
JOptionPane.ERROR_MESSAGE);
        } else {
            if (iwRButton.isSelected() ) {
                size =
Double.parseDouble(iInflationVal.getText())/100;
                texSize =
"i"+Double.parseDouble(iInflationVal.getText())/10
0;
            } else {
                size =
Double.parseDouble(iInflationVal.getText()) * 7;
                texSize =
"t"+Double.parseDouble(iInflationVal.getText())/10
0;
            }
            doClose(RET_OK);
        }
    } catch (java.lang.NumberFormatException e) {
        JOptionPane.showMessageDialog(null, "Inflation
value invalid. Enter values between 0 and 999.",
"Invalid input", JOptionPane.ERROR_MESSAGE);
    }
}

private void
cancelButtonActionPerformed(ActionEvent evt) {
    doClose(RET_CANCEL);
}

/** Closes the dialog */
private void
closeDialog(java.awt.event.WindowEvent evt) {
    doClose(RET_CANCEL);
}

private void
twRButtonActionPerformed(ActionEvent evt) {
    if(twRButton.isSelected())
        iwRButton.setSelected(false);
    else
        iwRButton.setSelected(true);
}

private void
iwRButtonActionPerformed(ActionEvent evt) {
    if(iwRButton.isSelected())
        twRButton.setSelected(false);
    else
        twRButton.setSelected(true);
}

```

```

    private void
    browseButtonActionPerformed(ActionEvent evt) {
        // Show open dialog; this method does not return
        until the dialog is closed
        // Start in current directory
        String filename = File.separator + "tmp";
        readFile rf = new readFile();
        JFileChooser fc = new JFileChooser(new
        File(filename));
        fc.addChoosableFileFilter(new imageFileFilter());
        int retval =
        fc.showOpenDialog(showImageFrame.this);

        if (retval == JFileChooser.APPROVE_OPTION) {
            File file = fc.getSelectedFile();
            if(fc.accept(file)){
                String fileTypeNames =
                fc.getTypeDescription(file);
                if (fileTypeNames.contains("JPG") ||
                fileTypeNames.contains("JPEG") ||
                fileTypeNames.contains("PNG")) {
                    // Invoke the readFile method in this class
                    String file_string = rf.readContents(file);
                    if (file_string != null) {
                        iPathVal.setText(file.getPath());
                    } else {
                        JOptionPane.showMessageDialog(null, "File
                        is empty!", "Error!",
                        JOptionPane.ERROR_MESSAGE);
                    }
                } else {
                    JOptionPane.showMessageDialog(null,
                    "Invalid file! Choose TEX compatible images(*.jpeg,
                    *.jpg, *.png) only.", "Error!",
                    JOptionPane.ERROR_MESSAGE);
                }
            } else {
                JOptionPane.showMessageDialog(null, "File
                not found. Check to see if the directory is correct.",
                "Error!", JOptionPane.ERROR_MESSAGE);
            }
        }
    }

    private void doClose(int retStatus) {
        returnStatus = retStatus;
        setVisible(false);
        dispose();
    }

    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new
        Runnable() {
            public void run() {
                showImageFrame dialog = new
                showImageFrame(new JFrame(), true);
                dialog.addWindowListener(new
                java.awt.event.WindowAdapter() {
                    public void
                    windowClosing(java.awt.event.WindowEvent e) {
                        System.exit(0);
                    }
                });
                dialog.setVisible(true);
            }
        });
    }
}

```

```

private JButton browseButton;
private JButton cancelButton;
private JLabel iCaptionLbl;
public JTextField iCaptionVal;
private JLabel iInflationLbl;
private JTextField iInflationVal;
public JTextField iPathVal;
private JLabel imgPathLbl;
private JRadioButton iwRButton;
private JButton okButton;
private JRadioButton twRButton;
private int returnStatus = RET_CANCEL;
}

```

**/\*\* UndoManagerHelper.java \*/**

```

package upmguilte;

import java.awt.Component;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;

import javax.swing.AbstractAction;
import javax.swing.Action;
import javax.swing.UIManager;
import javax.swing.undo.CannotRedoException;
import javax.swing.undo.CannotUndoException;
import javax.swing.undo.UndoManager;

class UndoManagerHelper {
    public static Action getUndoAction(UndoManager
    manager, String label) {
        return new UndoAction(manager, label);
    }

    public static Action getUndoAction(UndoManager
    manager) {
        return new UndoAction(manager, (String)
        UIManager.get("AbstractUndoableEdit.undoText"));
    }

    public static Action getRedoAction(UndoManager
    manager, String label) {
        return new RedoAction(manager, label);
    }

    public static Action getRedoAction(UndoManager
    manager) {
        return new RedoAction(manager, (String)
        UIManager.get("AbstractUndoableEdit.redoText"));
    }

    private abstract static class UndoRedoAction
    extends AbstractAction {
        UndoManager undoManager = new
        UndoManager();
        String errorMessage = "Cannot undo";
        String errorTitle = "Undo Problem";

        protected UndoRedoAction(UndoManager
        manager, String name) {
            super(name);
            undoManager = manager;
        }

        public void setErrorMessage(String newValue) {
            errorMessage = newValue;
        }
    }
}

```

```

public void setErrorTitle(String newValue) {
    errorTitle = newValue;
}

protected void showMessage(Object source) {
    if (source instanceof Component) {
        //JOptionPane.showMessageDialog((Component) source, errorMessage, errorTitle,
        JOptionPane.WARNING_MESSAGE);
    } else {
        System.err.println(errorMessage);
    }
}

public static class UndoAction extends
UndoRedoAction {
    private static final long serialVersionUID = 1L;

    public UndoAction(UndoManager manager,
String name) {
        super(manager, name);
        setErrorMessage("Cannot undo");
        setErrorTitle("Undo Problem");
    }

    public void actionPerformed(ActionEvent
actionEvent) {
        try {
            undoManager.undo();

```

```

        } catch (CannotUndoException
cannotUndoException) {
            showMessage(actionEvent.getSource());
            Toolkit.getDefaultToolkit().beep();
        }
    }

    public static class RedoAction extends
UndoRedoAction {
        private static final long serialVersionUID = 1L;

        public RedoAction(UndoManager manager,
String name) {
            super(manager, name);
            setErrorMessage("Cannot redo");
            setErrorTitle("Redo Problem");
        }

        public void actionPerformed(ActionEvent
actionEvent) {
            try {
                undoManager.redo();
            } catch (CannotRedoException
cannotRedoException) {
                showMessage(actionEvent.getSource());
                Toolkit.getDefaultToolkit().beep();
            }
        }
    }
}

```

## **XI. ACKNOWLEDGEMENT**