LatexEditor

Release Report

<Women of Wisdom>
<Konstantinos Chondralis 3109, Argyrios
Theodoridis 2978 >

VERSIONS HISTORY

Date	Version	Description	Author
23/5/2019	2.0	Final Version	Konstantinos Chondralis 3109, Argyrios Theodoridis 2978

1 Introduction

This document provides information concerning the **<2nd>** release of the project.

1.1 Purpose

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

1.2 Document Structure

The rest of this document is structured as follows. Section 2 specifies the acceptance tests that have been employed for this release of the project. Section 3 specifies the main design concepts for this release of the project.

2 Tests

<For the user stories included in this release specify below corresponding tests using a typical tabular form.>

2.1 Tests for User Story **<1>**

Test ID	TC1
User Story	*US1*
Test Class	*US1TestCreation*
Description	*The purpose of the test is to test the correct creation of a template document.*

2.2 Tests for User Story **<2>**

Test ID	TC2
User Story	US2
Test Class	US2TestEditing
Description	The purpose of the test is to test whether it is possible to edit the contents of a document.

2.3 Tests for User Story **<3>**

Test ID	TC3
User Story	US3
Test Class	US3TestAddLatexCommand
Description	The purpose of the test is to test whether it is possible to add preexisting latex commands.

2.4 Tests for User Story **<4>**

Test ID	TC4
User Story	US4
Test Class	US4TestVersionKeeping
Description	The purpose of the test is to test the proper function of version control utility.

2.5 Tests for User Story **<5>**

Test ID	TC5
User Story	US5
Test Class	US5TestVersionChange
Description	The purpose of the test is to test whether it is possible to change the version control strategy in runtime.

2.6 Tests for User Story **<6>**

Test ID	TC6
User Story	US6
Test Class	US6Test
Description	The purpose of this test is to test if it is possible to keep a version while the version control utility is off.

2.7 Tests for User Story **<7>**

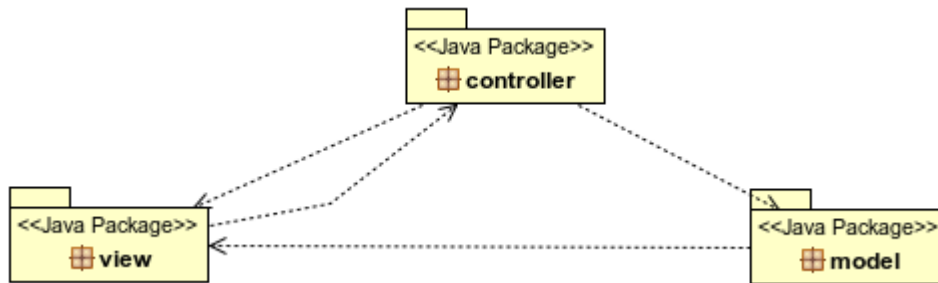| Test ID	TC7
User Story	US7
Test Class	US7TestRollback
Description	The purpose of the test is to test whether it is possible to rollback to a previous version of a document after creating a new version first.

2.8 Tests for User Story **<8>**

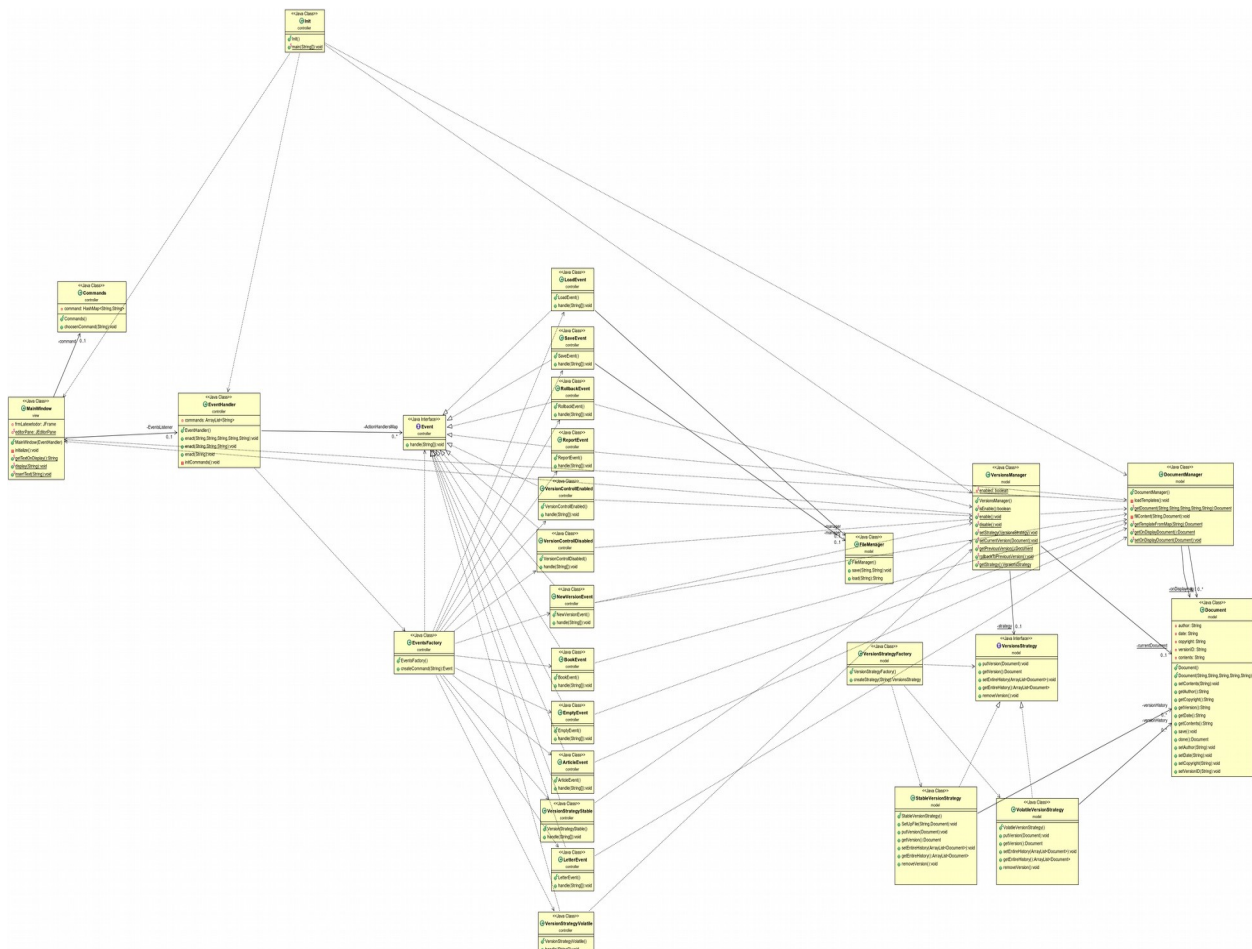| Test ID	TC8
User Story	US8
Test Class	US8TestSave
Description	The purpose of the test is to test if it is possible to save an onscreen document to file system.

2.9 Tests for User Story **<9>**

Test ID	TC9
User Story	US9
Test Class	US9TestLoad
Description	The purpose of the test is to test if it is possible to load a document from file system.

3 Design

3.1 Architecture

3.2 Design

Class Name: ArticleEvent.java	
Responsibilities:	**Collaborations:**
- To create an Article template document and display it on the main window - Implements Event interface	- MainWindow - DocumentManager - EventHandler

Class Name: BookEvent.java	
Responsibilities:	**Collaborations:**
- To create an Boko template document and display it on the main window - Implements Event interface	- MainWindow - DocumentManager - EventHandler

Class Name: ReportEvent.java	
Responsibilities:	**Collaborations:**
- To create an Report template document and display it on the main window - Implements Event interface	- MainWindow - DocumentManager - EventHandler

Class Name: LetterEvent.java	
Responsibilities:	**Collaborations:**
- To create an Letter template document and display it on the main window - Implements Event interface	- MainWindow - DocumentManager - EventHandler

Class Name: EmptyEvent.java

Responsibilities:	Collaborations:
▪ To create an Empty template document and display it on the main window ▪ Implements Event interface	▪ MainWindow ▪ DocumentManager ▪ EventHandler

Class Name: SaveEvent.java

Responsibilities:	Collaborations:
▪ To save the document displayed on screen to file system	▪ FileManager ▪ EventHandler

Class Name: LoadEvent.java

Responsibilities:	Collaborations:
▪ To load an existing document from file system to main window.	▪ FileManager ▪ EventHandler

Class Name: Rollback.java

Responsibilities:	Collaborations:
▪ To rollback to previous version of a document.	▪ EventHandler ▪ VersionManager

Class Name: VersionControllEnabled.java

Responsibilities:	Collaborations:
▪ To enable version control utility	▪ EventHandler ▪ VersionManager

Class Name: VersionControllDisabled

Responsibilities:	Collaborations:
▪ To disable version control	▪ EventHandler

utility	▪ VersionManager

Class Name: NewVersionEvent	
Responsibilities:	**Collaborations:**
▪ To create a new version of the onscreen document.	▪ EventHandler ▪ VersionManager

Class Name: VersionStrategyStable	
Responsibilities:	**Collaborations:**
▪ To change the current versioning strategy to Stable.	▪ EventHandler ▪ VersionManager ▪ DocumentManager

Class Name: VersionStrategyVolatile	
Responsibilities:	**Collaborations:**
▪ To change the current versioning strategy to volatile.	▪ EventHandler ▪ VersionManager ▪ DocumentManager

Class Name: Init	
Responsibilities:	**Collaborations:**
▪ To initialize the main window, document manager, version manager and event handler	▪ EventHandler ▪ VersionManager ▪ DocumentManager ▪ MainWindow

Class Name: Commands	
Responsibilities:	**Collaborations:**
▪ To generate prefabricated latex commands and append them to the document contents.	▪ MainWindow

Class Name: Event	
Responsibilities: ▪ To provide classes with handle method to handle the main window's events.	**Collaborations:**

Class Name: EventHandler	

Responsibilities:
 	
   ▪ To perform actions that handle events generated by the main window.
 	
   ▪ To generate the supported events and subsequently, the supported commands.
	Collaborations:
	
	   ▪ MainWindow
	
	   ▪ All supported Event classes
	
	   ▪ EventFactory

Class Name: EventFactory	

Responsibilities:
 	
   ▪ To generate the supported events and subsequently, the supported commands.
	Collaborations:
	
	   ▪ EventHandler
	
	   ▪ All supported Event classes

Class Name: Doucment	

Responsibilities:
 	
   ▪ To represent the information needed for a latex document.
	Collaborations:
	
	   ▪ DocumentManager
	
	   ▪ VersionManager
	
	   ▪ Version Strategies

Class Name: DocumentManager	

Responsibilities:
 	
   ▪ To provide the supported templates for new document creation, and to track the current document on display.
	Collaborations:
	
	   ▪ Document
	
	   ▪ All document creation classes(article, report...)
	
	   ▪ Init
	
	   ▪ MainWindow

Class Name: FileManager	
Responsibilities:	**Collaborations:**
▪ To save and load documents to and from the file system.	▪ SaveEvent ▪ LoadEvent

Class Name: StableVersionStrategy	
Responsibilities:	**Collaborations:**
▪ Provide logic to implement keeping versions in file system.	▪ Documents ▪ VersionManager ▪ VersionFactory

Class Name: VolatileVersionStrategy	
Responsibilities:	**Collaborations:**
▪ Provide logic to implement keeping versions in memory.	▪ Documents ▪ VersionManager ▪ VersionFactory

Class Name: VersionsManager	
Responsibilities:	**Collaborations:**
▪ To create new versions, and to rollback to previous versions of a document depending on the current strategy.	▪ Documents ▪ VersionStrategy ▪ Init ▪ NewVersionEvent ▪ RollbackEvent ▪ VersionStrategyEnable/Disable ▪ VersionStrategyStable/Volatile

Class Name: VersionStrategyFactory	
Responsibilities:	**Collaborations:**
▪ To create the suported version strategies.	▪ VersionStrategyStable/Volatile ▪ VersionManager ▪ VersionStrategyEnabled

Class Name: VersionStrategy	
Responsibilities:	**Collaborations:**
▪ To provide classes with methods needed for implementing version control utilities	

Class Name: MainWindow	
Responsibilities:	**Collaborations:**
▪ To create the GUI and to pass information to event handler regarding the events generated by the user.	▪ EventHandler ▪ Init ▪ Commands ▪ DocumentManager ▪ VersionManager