
LatexEditor

Release Report

group5026

Βίκτωρ Μεγήρ 3026

Βαΐα Νταφοπούλου 3050

VERSIONS HISTORY

Date	Version	Description	Author
21/5/2019	2.0	Final Version	Victor Megir, Vaia Ntafopoulou

1 Introduction

This document provides information concerning the 2 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

1st release: US-1, US-1, US-3, US-4

2nd release: US-5, US-6, US-7, US-8 and US-9.

2.1 Tests for User Story US-1

Test ID	<i>TexDocManagerTest</i>
User Story	<i>US-1</i>
Test Class	<i>TexDocManagerTest.java</i>
Description	<i>Implement 5 acceptance tests that validate creation of TexDoc objects for 4 different templates and a no template option. Input: a Create Command. Expected result: the text from the JTextArea matches the template contents.</i>

2.2 Tests for User Story US-2

Test ID	<i>EditDocTest</i>
User Story	<i>US-2</i>
Test Class	<i>EditDocTest.java</i>
Description	<i>Check that the text of the JTextArea, when edited contains the changed text. Input: a window. Expected result: After a change in the window's text, the text of the window has changed.</i>

2.3 Tests for User Story **US-3**

Test ID	<i>LatexCommandTest</i>
User Story	<i>US-3</i>
Test Class	<i>LatexCommandTest.java</i>
Description	<i>Implement 8 acceptance tests that validate the the execution of the supported latex commands. Input: 8 LatexCommands and a JTextArea. Output: The text of the JTextArea contains the contents of each command.</i>

2.4 Tests for User Story **US-4**

Test ID	<i>VersionStrategyManagerTest</i>
User Story	<i>US-4</i>
Test Class	<i>VersionStrategyManagerTest.java</i>
Description	<i>Implements 2 acceptance tests that validate the execution of the different version tracking strategies. Input: A VersionStartegyManager, a TexDoc (document obejct) and a Commit command. Expected result: The contents of the TexDoc are stored according to each strategy.</i>

2.5 Tests for User Story **5**

Test ID	<i>ChangeVersionStrategyTest</i>
----------------	----------------------------------

User Story	<i>US-5</i>
Test Class	<i>ChangeVersionStrategyTest.java</i>
Description	<i>This test checks the correct changing of Version Startegies. We use 1 test for change from stable to volatile and 1 for the opposite.</i>

2.6 Tests for User Story 6

Test ID	<i>DisableVersionTrackingTest</i>
User Story	<i>US-6</i>
Test Class	<i>DisableVersionTrackingTest.java</i>
Description	<i>This test checks the correct enabling and disabling of Version Tracking mechanism. We have 1 test for disabling and one for enabling.</i>

2.7 Tests for User Story 7

Test ID	<i>RollbackCommandTest</i>
User Story	<i>US-7</i>
Test Class	<i>RollbackCommandTest.java</i>
Description	<i>This test checks the rollback mechanism. We have 1 test that changes the current TexDoc and rolls back to the previous version of the TexDoc.</i>

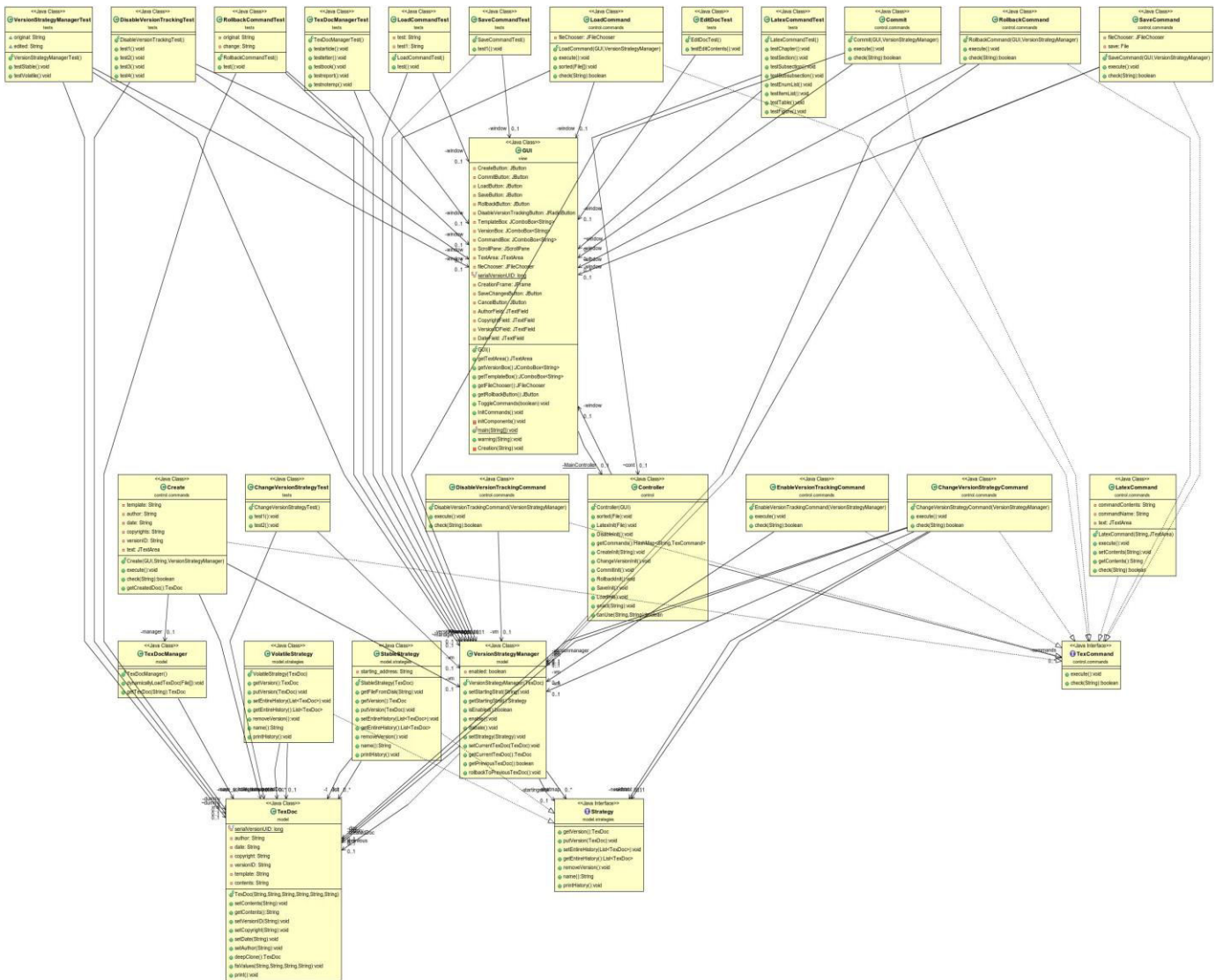
2.8 Tests for User Story 8

Test ID	<i>SaveCommandTest</i>
User Story	<i>US-8</i>
Test Class	<i>SaveCommandTest.java</i>
Description	<i>This test checks the saving of a document. We use 1 test that saves a TexDoc and checks if the contents are correct.</i>

2.9 Tests for User Story 9

Test ID	<i>LoadCommandTest</i>
User Story	<i>US-9</i>
Test Class	<i>LoadCommandTest.java</i>
Description	<i>This test checks the loading of a document. We use 1 test that saves a TexDoc then loads it and checks if the contents are correct. It also checks if the history of the saved TexDoc is loaded by executing a rollback command.</i>

3.1 Architecture



3.2 Design

Class Name: GUI	
Responsibilities: <ul style="list-style-type: none">▪ Implements the user interface	Collaborations: <ul style="list-style-type: none">▪ Controller

Class Name: Controller	
Responsibilities: <ul style="list-style-type: none">▪ Implements the communication between packages view and model	Collaborations: <ul style="list-style-type: none">▪ GUI▪ Invoker▪ VersionStrategyManager

Class Name: TexCommand	
Responsibilities: <ul style="list-style-type: none">▪ Interface for all the commands in package control.commands	Collaborations:

Class Name: LatexCommand	
Responsibilities: <ul style="list-style-type: none">▪ Implements TexCommand interface▪ Sets contents of a command on the GUI's text area	Collaborations: <ul style="list-style-type: none">▪ TexCommand

Class Name: Create	
Responsibilities: <ul style="list-style-type: none"> ▪ Implements TexCommand interface ▪ Creates a document by cloning an existing document 	Collaborations: <ul style="list-style-type: none"> ▪ TexDoc ▪ TexDocManager ▪ TexCommand ▪ VersionStrategyManager

Class Name: Commit	
Responsibilities: <ul style="list-style-type: none"> ▪ Implements TexCommand interface ▪ Saves a version of a document based on a version strategy 	Collaborations: <ul style="list-style-type: none"> ▪ GUI ▪ VersionStrategyManager ▪ TexCommand

Class Name: TexDoc	
Responsibilities: <ul style="list-style-type: none"> ▪ Keeps some meta-data for each document ▪ Allows other classes to clone a document 	Collaborations:

Class Name: TexDocManager	
Responsibilities: <ul style="list-style-type: none"> ▪ Creates a TexDoc ▪ Keeps a map of different TexDocs and clones them based on a template 	Collaborations: <ul style="list-style-type: none"> ▪ TexDoc

Class Name: VersionStrategyManager	
Responsibilities: <ul style="list-style-type: none"> ▪ Creates a version strategy ▪ Keeps a map of the supported version strategies ▪ Returns the previous version of a TexDoc ▪ Enables and disables version tracking 	Collaborations: <ul style="list-style-type: none"> ▪ TexDoc ▪ Strategy

Class Name: Strategy	
Responsibilities: <ul style="list-style-type: none"> ▪ Interface for all strategies ▪ Keeps a history of all the versions ▪ Inserts and removes a version from the history 	Collaborations:

Class Name: VolatileStrategy	
Responsibilities: <ul style="list-style-type: none"> ▪ Implements Startegy inteface ▪ Keeps an ArrayList of a document's versions 	Collaborations: <ul style="list-style-type: none"> ▪ Strategy ▪ TexDoc

Class Name: StableStrategy	
Responsibilities:	Collaborations:

<ul style="list-style-type: none"> ▪ Implements Strategy interface ▪ Saves a document's versions in the disk storage 	<ul style="list-style-type: none"> ▪ Strategy ▪ TexDoc
--	--

Class Name: ChangeVersionStrategyCommand	
Responsibilities: <ul style="list-style-type: none"> ▪ Implements the TexCommand interface. ▪ Changes the Version Tracking Strategy. 	Collaborations: <ul style="list-style-type: none"> ▪ TexCommand ▪ Strategy ▪ VersionStrategyManager ▪ TexDoc

Class Name: DisableVersionTrackingCommand	
Responsibilities: <ul style="list-style-type: none"> ▪ Implements the TexCommand interface. ▪ Disables Version Tracking mechanism. 	Collaborations: <ul style="list-style-type: none"> ▪ TexCommand ▪ VersionStrategyManager

Class Name: EnableVersionTrackingCommand	
Responsibilities: <ul style="list-style-type: none"> ▪ Implements the TexCommand interface. ▪ Enables Version Tracking mechanism. 	Collaborations: <ul style="list-style-type: none"> ▪ TexCommand ▪ VersionStrategyManager