LatexEditor

Release Report

TODO

ΣΑΚΚΟΣ ΣΤΑΥΡΟΣ ΑΜ:2532

**Versions History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 24/05/2019 | 1.2 | Second Release (Final) | Sakkos Stavros |

# Introduction

This document provides information concerning the first release of the project.

## 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.

## 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.

# Tests

## Tests for User Story US1

|  |  |
| --- | --- |
| **Test ID** | US-1 |
| **User Story** | UserStory1 |
| **Test Class** | CreateDocumentTest |
| **Description** | There are 5 acceptance tests : 1 for each template + 1 for an empty document.For each test,an –CreateCommand- object has been created and then executed.Finally, there is a comparison between the Object’s text and document’s text for each test.  (Note: This test should be run first) |

## Tests for User Story US2

|  |  |
| --- | --- |
| **Test ID** | US-2 |
| **User Story** | UserStory2 |
| **Test Class** | EditDocumentTest |
| **Description** | There are 2 acceptance tests : 1 for an empty document and 1 for a non empty document.For each test,an –EditDocument- Object has been created. Each Object edits the text inside the file during the execution. |

## Tests for User Story US3

|  |  |
| --- | --- |
| **Test ID** | US-3 |
| **User Story** | UserStory3 |
| **Test Class** | AddLatexCommandTest |
| **Description** | There are 8 acceptance tests : 1 for each given Latex command. The idea for implementing the tests is to create an AddLatexCommand that changes the contents of a document, execute it and subsequently get the new contents of the document (getContents()) and compare them against the contents that have been set. |

## Tests for User Story US5

|  |  |
| --- | --- |
| **Test ID** | US-5 |
| **User Story** | UserStory5 |
| **Test Class** | ChangeStrategyTest |
| **Description** | To test this story we need at least 2 acceptance tests, one for each different versions strategy. An idea for implementing the tests is to create a ChangeStrategyCommand, execute it, and check whether the strategy indeed changed. |

## Tests for User Story US6

|  |  |
| --- | --- |
| **Test ID** | US-6 |
| **User Story** | UserStory6 |
| **Test Class** | TrackingTest |
| **Description** | To test this story we will edit a Document 2 times.The first time tracking will be enabled but the second time it will be disabled.Then we will check whether the document has changed after each edit or not.  (In this case,tracking is boolean instead of a Command object which changes with a toggle method – JRadioButton) |

## Tests for User Story US7

|  |  |
| --- | --- |
| **Test ID** | US-7 |
| **User Story** | UserStory7 |
| **Test Class** | RollBackTest |
| **Description** | An idea for this test is to get the contents of the previous version of a document, create a RollbackCommand, execute it, and check whether the contents of the current document match with the contents of the previous version. |

## Tests for User Story US8

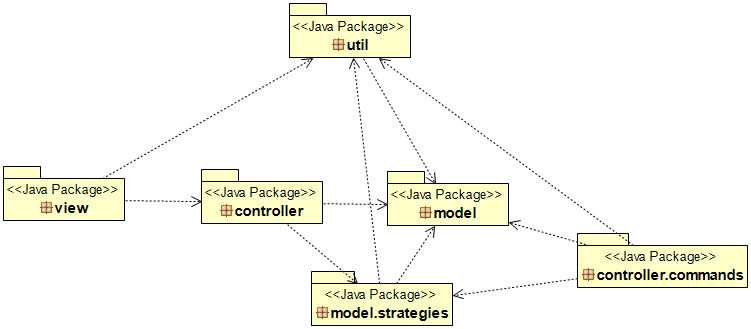
|  |  |
| --- | --- |
| **Test ID** | US-8 |
| **User Story** | UserStory8 |
| **Test Class** | SaveDocumentTest |
| **Description** | The idea for implementing the tests is to create an AddLatexCommandthat changes the contents of a document, execute it and subsequently get the new contents of the document (getContents()) and compare them against the contents that have been set. |

## Tests for User Story US9

|  |  |
| --- | --- |
| **Test ID** | US-9 |
| **User Story** | UserStory9 |
| **Test Class** | LoadDocumentTest |
| **Description** | The idea for this test is to create LoadCommand, execute it, get the contents of the current version of a document and check whether the contents match with the contents of the file that have been loaded from disk .(Note: CreateDocument Test has to run first in order to Load the files properly) |

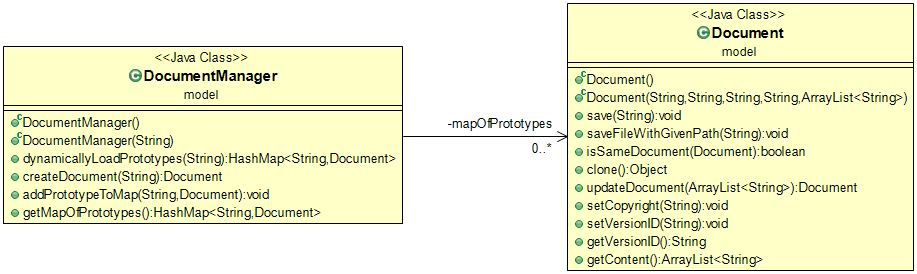
# Design

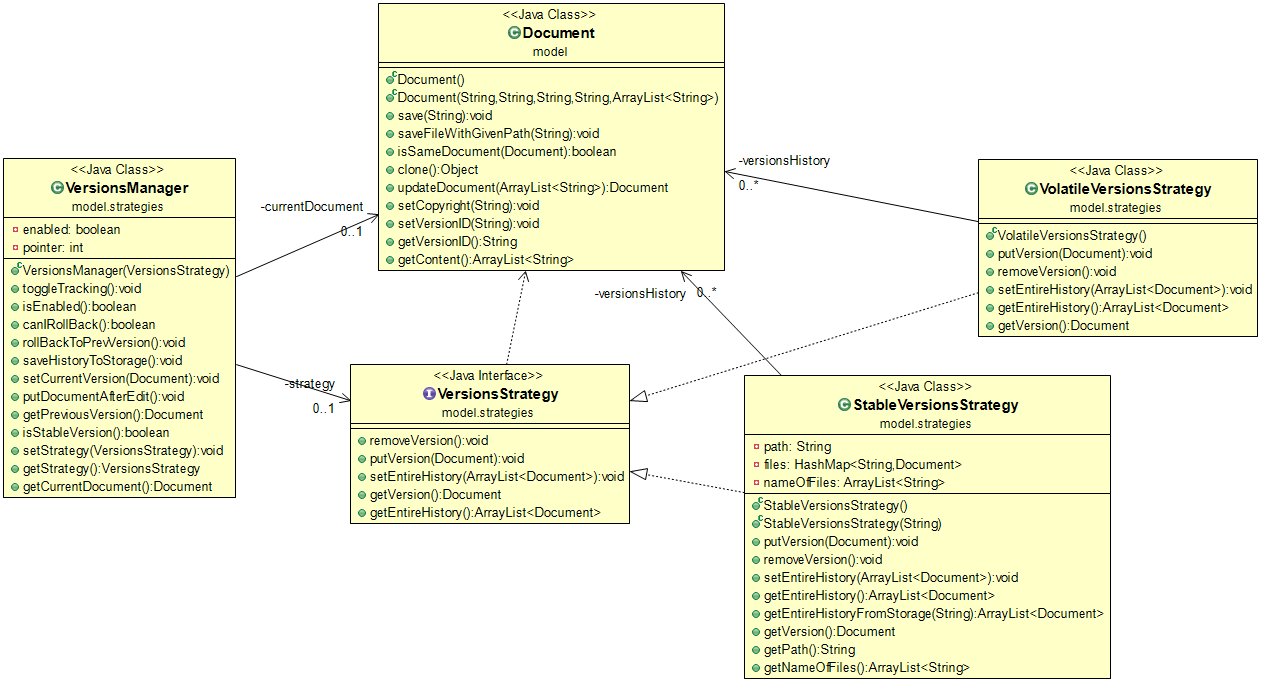
## Architecture

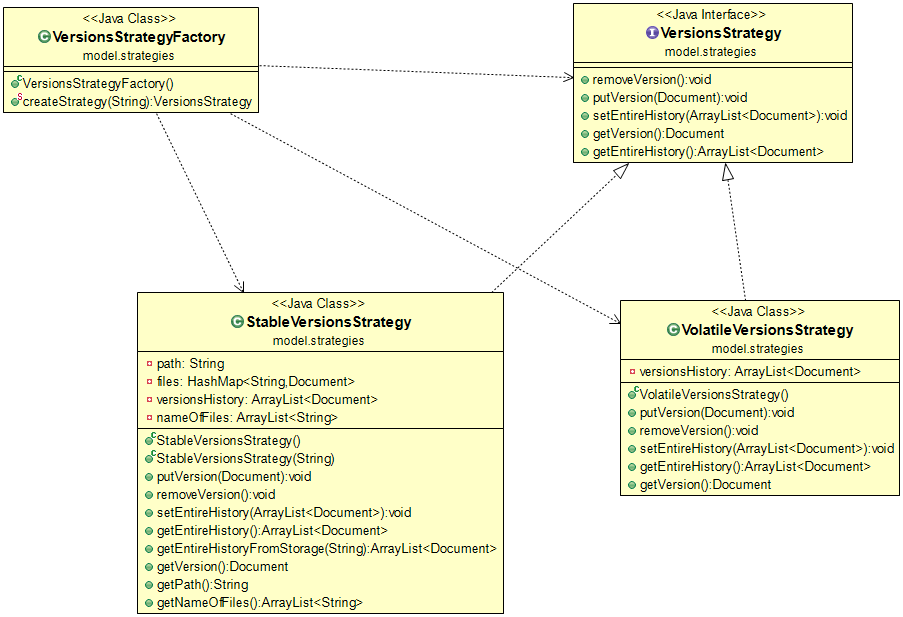


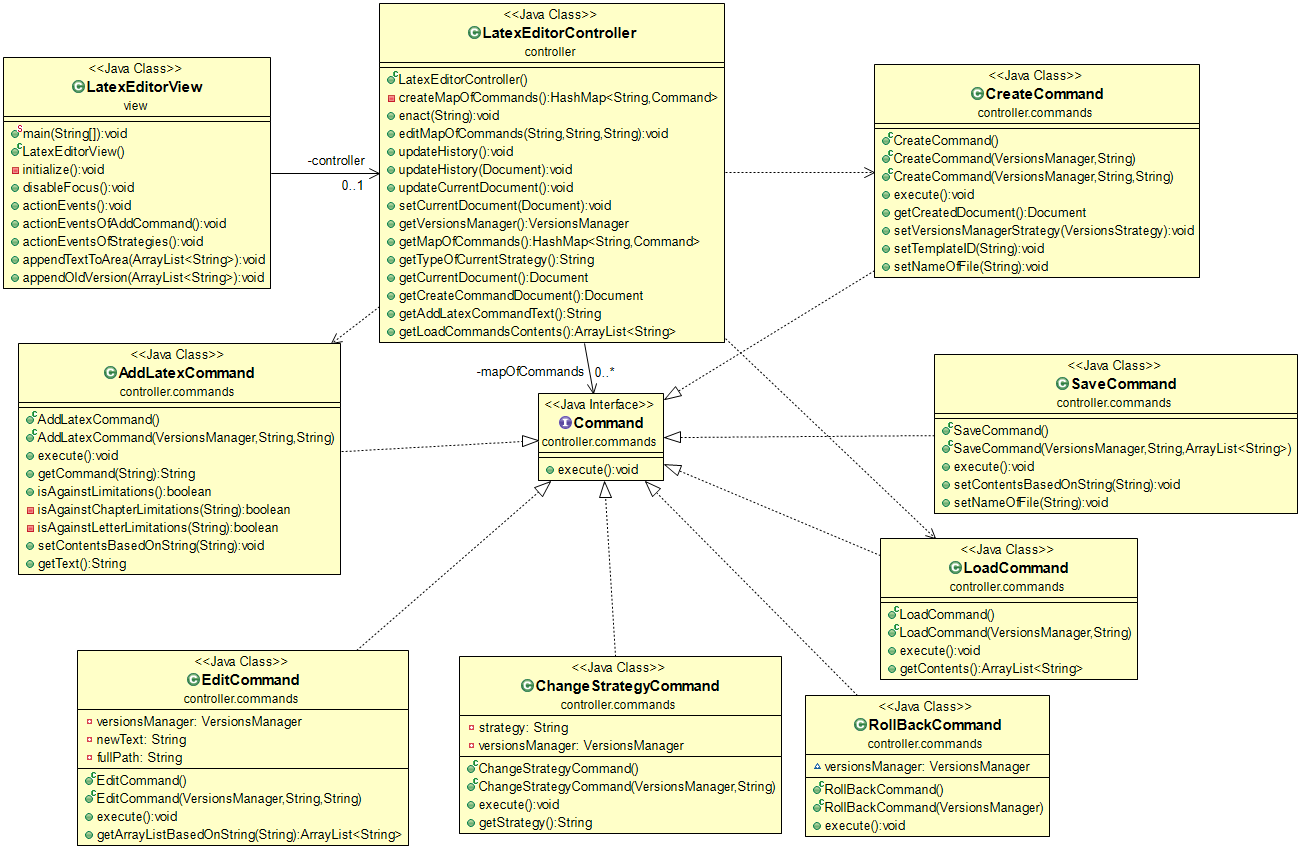
(Note: Update is done with the controller instead of the viewer.)

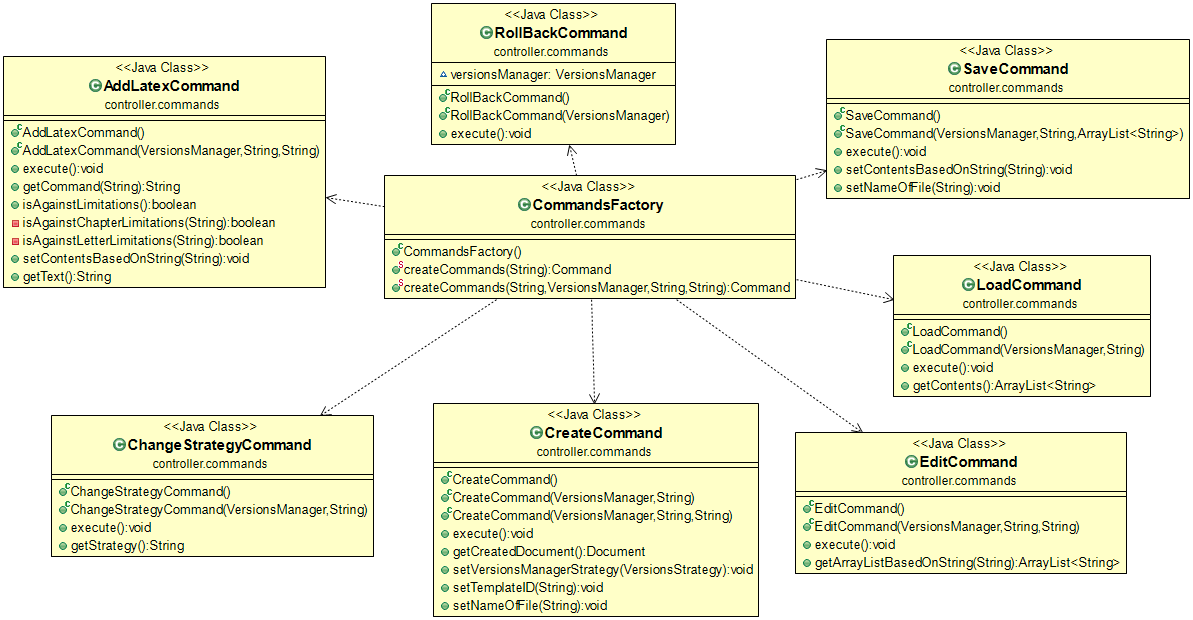
## Design











3.3 CRC cards for the most important classes

|  |  |
| --- | --- |
| **Class Name: DocumentManager** | |
| **Responsibilities:**   * Dynamic loading of prototypes * Document-Map creation * ... | **Collaborations:**   * Document * …. |

|  |  |
| --- | --- |
| **Class Name: Document** | |
| **Responsibilities:**   * Document creation * Exports Document * …. | **Collaborations:**   * FileParser * DocumentManager * VersionsStrategyManager * .... |

|  |  |
| --- | --- |
| **Class Name: FileParser** | |
| **Responsibilities:**   * File parser * Extracts text from files * Given X produces Y (String to ArrayList) | **Collaborations:**   * Document * .... |

|  |  |
| --- | --- |
| **Class Name: VersionsManager** | |
| **Responsibilities:**   * Document history management * Provides previous versions of the Documents. | **Collaborations:**   * VolatileVersionsStrategy * StableVersionsStrategy * LatexEditorController * …. |

|  |  |
| --- | --- |
| **Class Name: LatexEditorView** | |
| **Responsibilities:**   * Graphics * Provides data to controller * …. | **Collaborations:**   * LatexEditorController * …. |

|  |  |
| --- | --- |
| **Class Name: LatexEditorController** | |
| **Responsibilities:**   * Updates viewer’s data * Provides data to controller * …. | **Collaborations:**   * LatexEditorView * …. |