
<LATEX EDITOR PROJECT >

OVERALL REPORT

VERSION <2.0>

<IOANNA PAPAGIANNI 2790, NIKOS PANTELIDIS 2787>

TABLE OF CONTENTS

Introduction	4
Design Recovery	4
Architecture	18
Detailed Design	5
Implementation	8
Quality Assessment	14
Re-engineered Design	18
Architecture	18
Detailed Design	19
Implementation	22

INTRODUCTION

This is the overall report for LatexEditor Project. The goal of the Latex editor is to facilitate the usage of Latex commands for the preparation of Latex documents. The novel idea of this LatexEditor, is its multi-strategy version tracking functionalities that enable undo and redo actions.

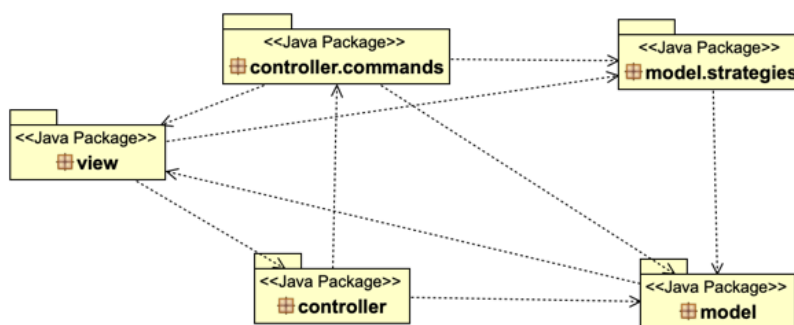
There are two parts, **design recovery** and **re-engineered design**. Design recovery, shows the design and logic of the original given project, it detects the collaborations among classes and eventually, spots functional and non-functional issues.

In re-engineered design part, our mission is to reengineer and extend an existing Java application. The first phase, is refactoring the application so that the internal structure and architecture are enhanced as well as fixing existing bugs in order to eliminate the deviations of the software from its expected delivery or service. The second phase is extending the application by adding new functionalities and respective tests.

DESIGN RECOVERY

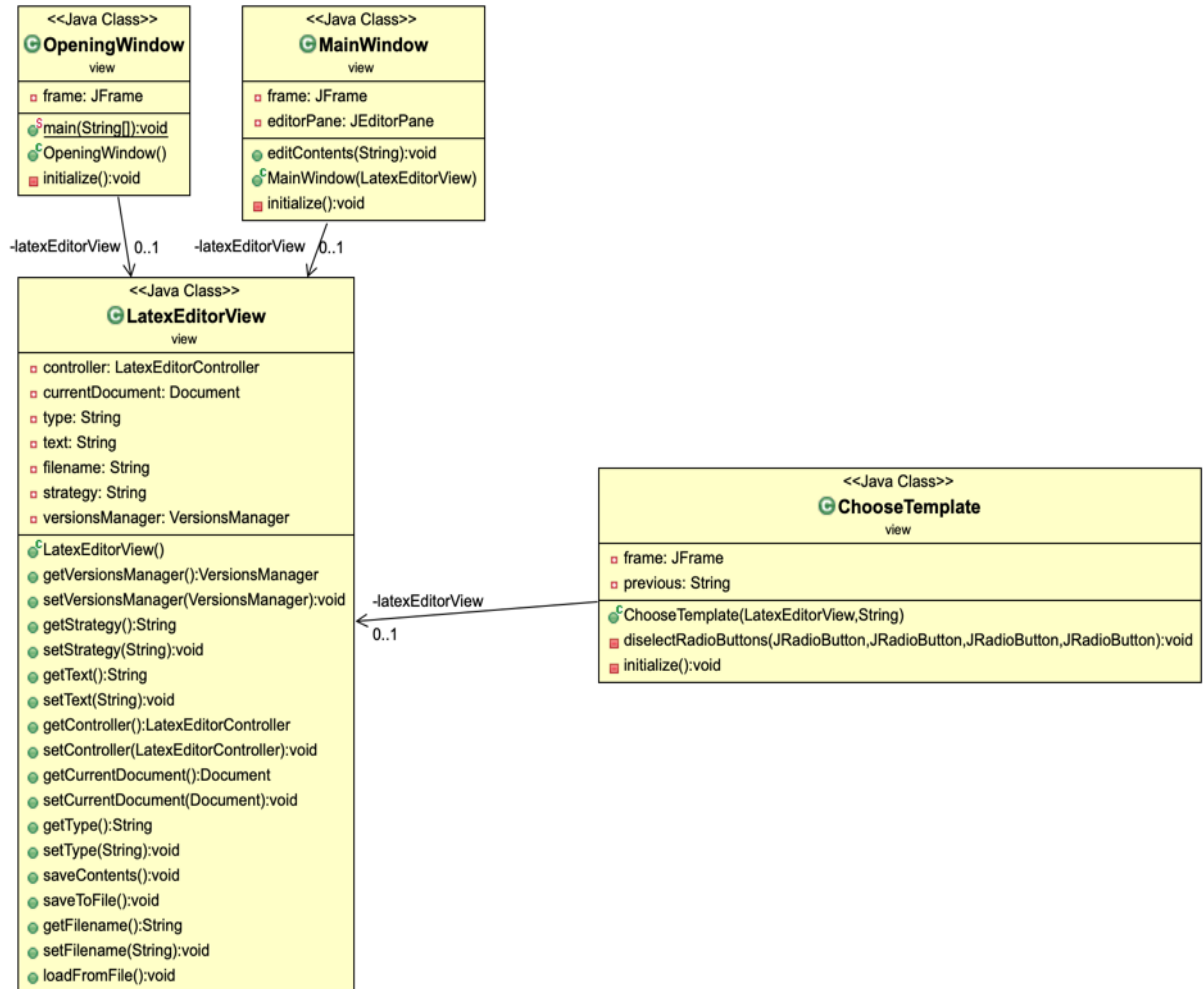
ARCHITECTURE

packages UML

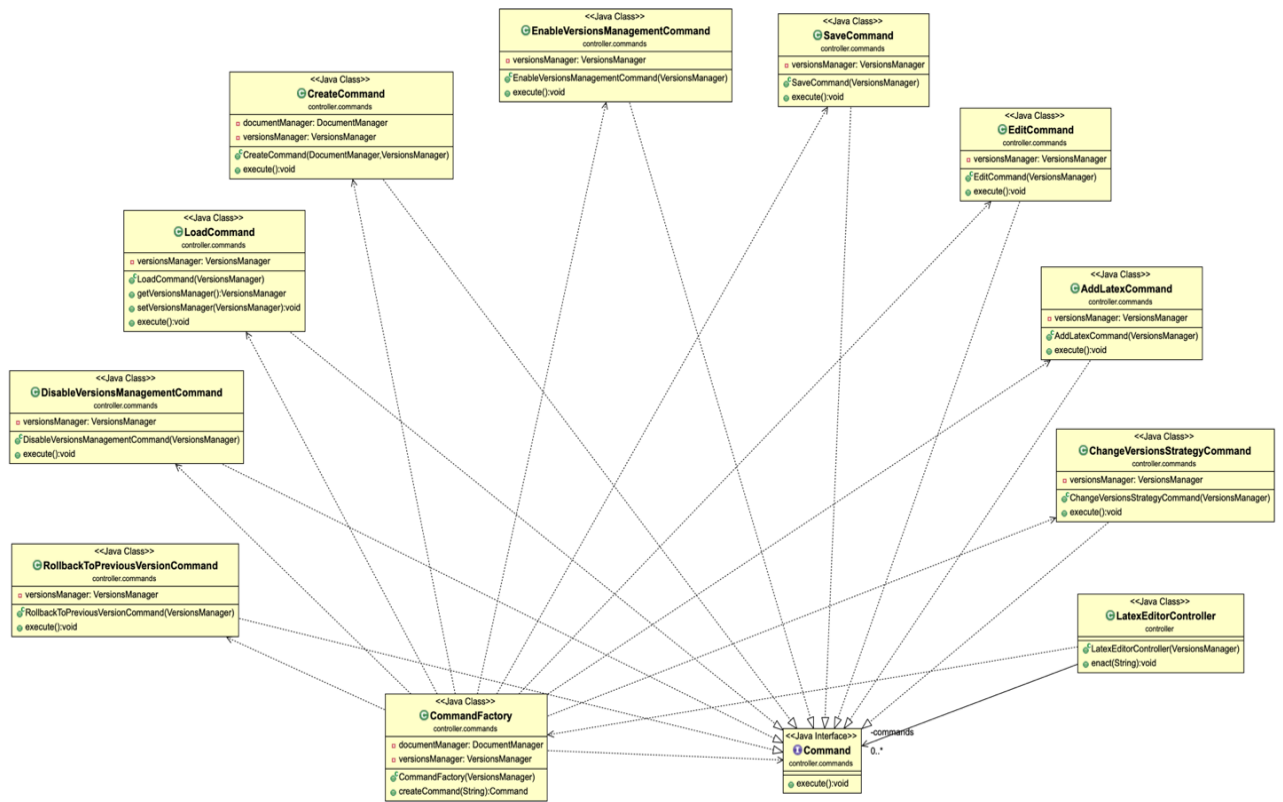


DETAILED DESIGN

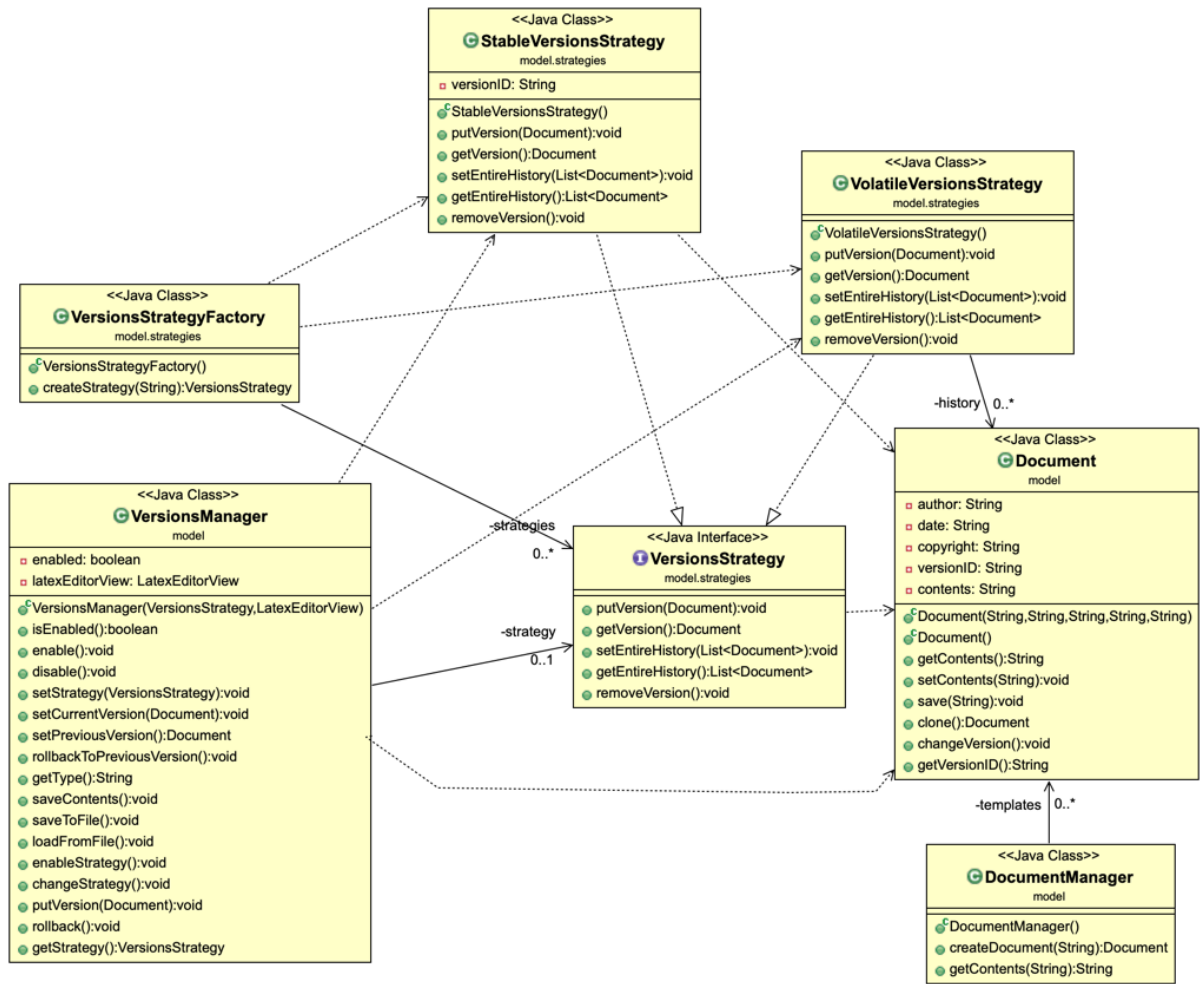
view package UML



- controller package UML



- model package UML



Class Name: <code>Document.class</code> <i>//package model</i>	
Responsibilities	Collaborations
This class gives us the basic properties of the Documents, allows us to configure them properly and save it to a specific file.	-----
Class Name: <code>DocumentManager.class</code> <i>//package model</i>	
Responsibilities	Collaborations
This class stores Latex templates in a HashMap and gives us a deep copy of a specific template.	<ul style="list-style-type: none"> • <code>Document.class</code>
Class Name: <code>VersionsManager.class</code> <i>//package model</i>	
Responsibilities	Collaborations
This class is responsible for the strategy management but it also uses methods from the <code>LatexEditorView</code> to load and store the different versions.	<ul style="list-style-type: none"> • <code>LatexEditorView.class</code> • <code>VersionsStrategy.class</code> • <code>StableVersionsStrategy.class</code> • <code>VolatileVersionsStrategy.class</code> • <code>Document.class</code>

Class Name: <code>VersionsStrategyFactory.class</code> <i>//package model.strategies</i>	
Responsibilities	Collaborations
This class creates the Volatile and Stable Version Strategies and returns the specified strategy.	<ul style="list-style-type: none"> • <code>VersionsStrategy.class</code> • <code>StableVersionsStrategy.class</code> • <code>VolatileVersionsStrategy.class</code>
Class Name: <code>VersionsStrategy.class</code> <i>//package model.strategies</i>	
Responsibilities	Collaborations
This class gives us the interface for the different Version Strategies.	<ul style="list-style-type: none"> • <code>Document.class</code>
Class Name: <code>StableVersionsStrategy.class</code> <i>//package model.strategies</i>	
Responsibilities	Collaborations
This class is responsible for the file-based versioning type.	<ul style="list-style-type: none"> • <code>Document.class</code> • <code>VersionsStrategy.class</code>
Class Name: <code>VolatileVersionsStrategy.class</code> <i>//package model.strategies</i>	
Responsibilities	Collaborations
This class is responsible for the memory-based versioning type.	<ul style="list-style-type: none"> • <code>Document.class</code> • <code>VersionsStrategy.class</code>

Class Name: LatexEditorController.class //package controller	
Responsibilities	Collaborations
This class stores the available commands in a HashMap and enacts a particular command.	<ul style="list-style-type: none"> • CommandFactory.class • VersionsManager.class • Command.class
Class Name: Command.class //package controller.commands	
Responsibilities	Collaborations
This class specifies the interface for the commands.	-----
Class Name: CommandFactory.class //package controller.commands	
Responsibilities	Collaborations
This class is responsible for the creation of the specified command object.	<ul style="list-style-type: none"> • DocumentManager.class • VersionsManager.class
Class Name: AddLatexCommand.class //package controller.commands	
Responsibilities	Collaborations
This class updates the document changes and saves and creates a new version of the document when a version strategy is enabled.	<ul style="list-style-type: none"> • Command.class • VersionsManager.class

Class Name: <code>ChangeVersionsStrategyCommand.class //package controller.commands</code>	
Responsibilities	Collaborations
This class reverts the current VersionStrategy.	<ul style="list-style-type: none"> • Command.class • VersionsManager.class
Class Name: <code>CreateCommand.class //package controller.commands</code>	
Responsibilities	Collaborations
This class creates a document according to a specific template.	<ul style="list-style-type: none"> • Command.class • VersionsManager.class • documentManager.class
Class Name: <code>EditCommand.class //package controller.commands</code>	
Responsibilities	Collaborations
This class updates the document changes and saves and creates a new version of the document when a version strategy is enabled.	<ul style="list-style-type: none"> • Command.class • VersionsManager.class
Class Name: <code>DisableVersionsManagementCommand.class //package controller.commands</code>	
Responsibilities	Collaborations
This class disables the current VersionStrategy.	<ul style="list-style-type: none"> • Command.class

	<ul style="list-style-type: none"> • VersionsManager.class
Class Name: EnableVersionsManagementCommand.class <i>//package controller.commands</i>	
Responsibilities	Collaborations
This class enables the current VersionStrategy.	<ul style="list-style-type: none"> • Command.class • VersionsManager.class
Class Name: LoadCommand.class <i>//package controller.commands</i>	
Responsibilities	Collaborations
This class loads a document from a specific file and finds out its template type from the first line of the document.	<ul style="list-style-type: none"> • Command.class • VersionsManager.class
Class Name: RollBackToPreviousVersionCommand.class <i>//package controller.commands</i>	
Responsibilities	Collaborations
This class removes the current version of the document and set it to the previous version if the version strategy is enabled.	<ul style="list-style-type: none"> • Command.class • VersionsManager.class
Class Name: SaveCommand.class <i>//package controller.commands</i>	
Responsibilities	Collaborations
This class saves the document to a file.	<ul style="list-style-type: none"> • Command.class

	<ul style="list-style-type: none"> • VersionsManager.class
Class Name: MainWindow.class //package view	
Responsibilities	Collaborations
This class adds the Latex Commands, creates and initializes the MainWindow.	<ul style="list-style-type: none"> • LatexEditorView.class • Document.class • ChooseTemplate.class
Class Name: OpeningWindow.class //package view	
Responsibilities	Collaborations
This class creates and initializes the opening window.	<ul style="list-style-type: none"> • LatexEditorView.class • LatexEditorController.class • VersionsManager.class • VersionsStrategy.class • VolatileVersionsStrategy.class • ChooseTemplate.class
Class Name: ChooseTemplate.class //package view	
Responsibilities	Collaborations

<p>This class creates and initializes the chooseTemplate window.</p>	<ul style="list-style-type: none"> • MainWindow.class • LatexEditorView.class • OpeningWindow.class
<p>Class Name: LatexEditorView.class //package view</p>	
<p>Responsibilities</p>	<p>Collaborations</p>
<p>This class has a lot of setters and getters, saves and loads Documents.</p>	<ul style="list-style-type: none"> • LatexEditorController.class • Document.class • VersionsManager.class

QUALITY ASSESSMENT

model package:

VersionsManager class:

- Message chains: MiddleMan class, some methods of this class, call methods from LatexEditorView class without adding some additional functionality.

setCurrentVersion(), getType(), saveContents(), saveToFile(), loadFromFile(), enableStrategy(), rollback()

- Dead code: setPreviousVersion(), rollbackToPreviousVersion()

- Duplicate code: `enableStrategy()` method uses the same code with `changeStrategy()` method in else-if statements instead of calling `enableStrategy()`.
- Other issues: *enabled* field is not initialized in the constructor and as a result, `isEnabled()` method returns NULL if `enable()` or `disable()` method are not called yet.

`rollback()` method does what `rollbackToPreviousVersion()` should do.

DocumentManager class:

- Duplicate code: constructor and `getContent()` method repeat the same lines of code.
- Long method: `getContent()` method is too long, it hardcodes the latex templates' text.

model.strategies package

VersionsStrategyFactory class:

- “Dead class”: This class is never used.

controller package

LatexEditorController class:

- Duplicate code: Naïve way to fill the HashMap.

controller.commands package

AddLatexCommand class, EditCommand class:

- Duplicate code: These classes do exactly the same thing and they shouldn't. AddLatexCommand class should be responsible for latex command fields (add section etc.).

CommandFactory class:

- Duplicate code: createCommand() method

view package

MainWindow class:

- Duplicate code: editContents() method has hardcoded text and uses many if-statements that repeat lines of code.
- Wrong-placed method: editContents() is part of the application logic and should not be here.

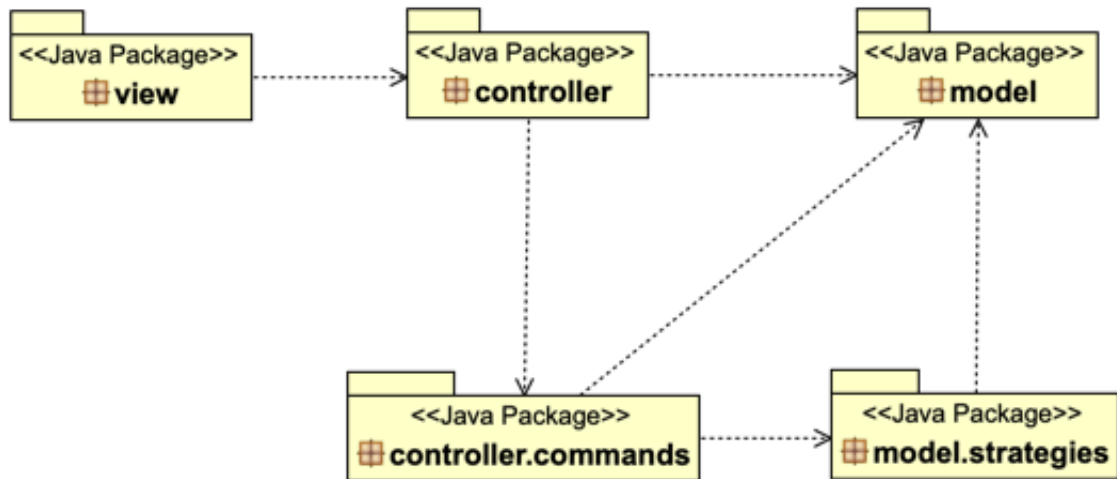
LatexEditorView class:

- Wrong-placed class: This class contains application logic that should be in model package.
- Misleading name: It has nothing to do with view.
- Too many responsibilities: It has many setter and getter methods that are already implemented in other classes and also saving and loading methods are implemented here
- Class with no purpose: This class could be avoided if we split its functionalities to the rightful classes.

General severe issues:

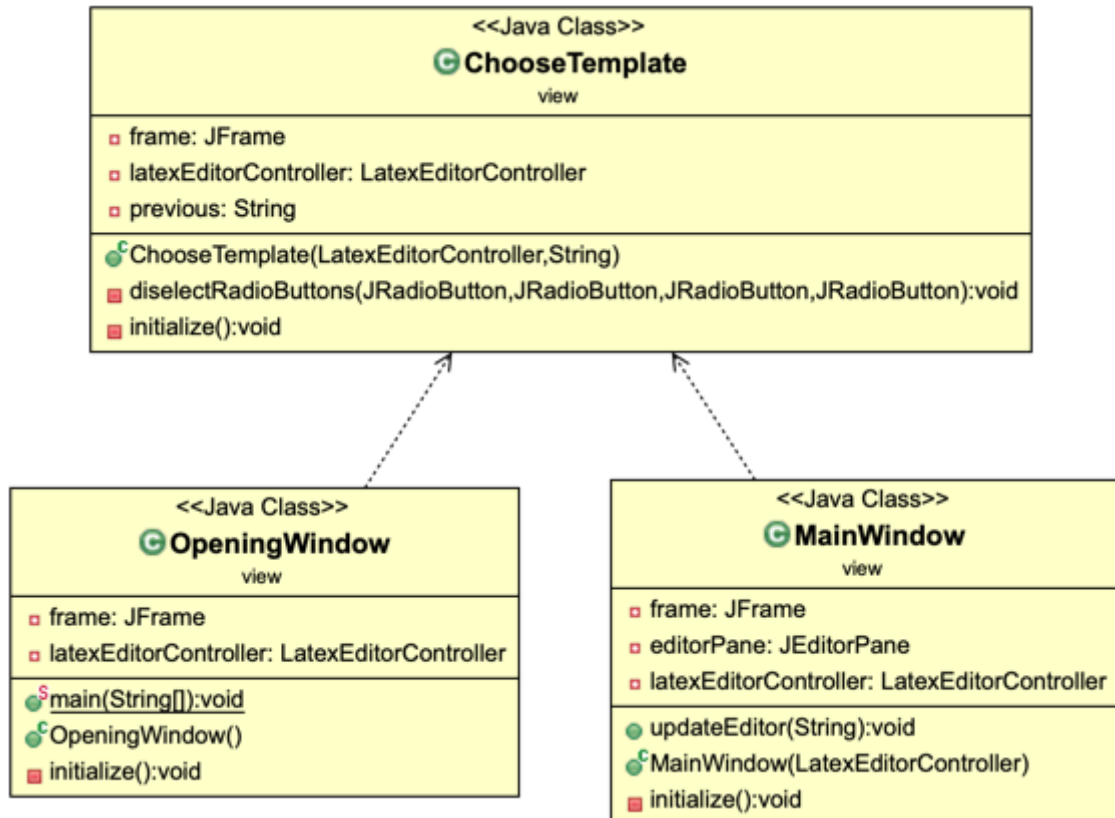
- There are many misleading methods and field names that could be altered.
- “Disable” button has a bug, it cannot be unmarked but it works.
- “Exit” button from the main bar (File-Exit) does not work.
- “Open Existing Document” button opens the file explorer but it cannot load any file.
- In order to save a file, due to complicated logic of the code, firstly is needed to press the “Save” button and then, “Save File” button. The only case where “Save” button is not needed first, is when you only add commands.
- There is no “Help” button to guide you through the program.

Packages UML

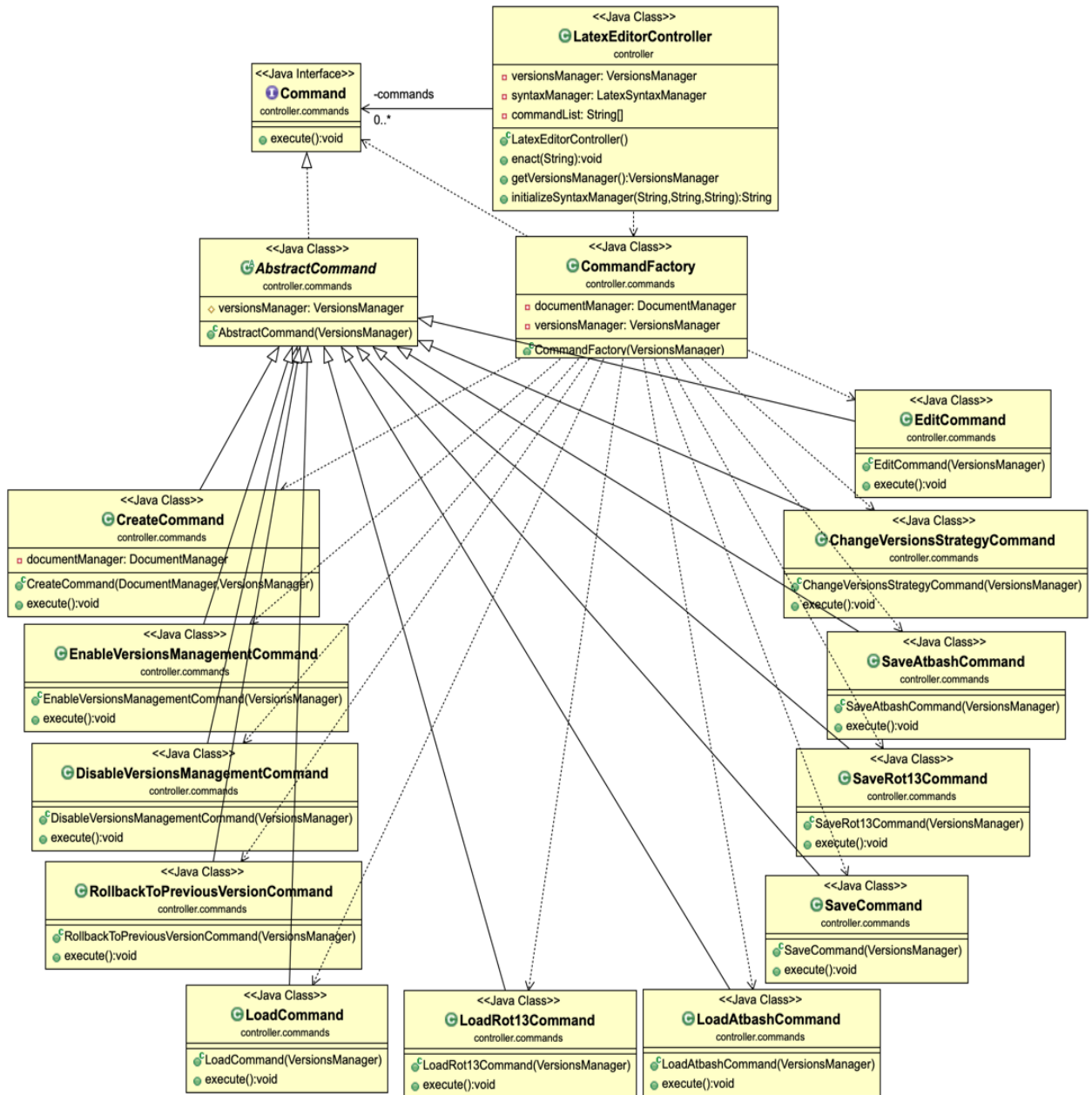


DETAILED DESIGN

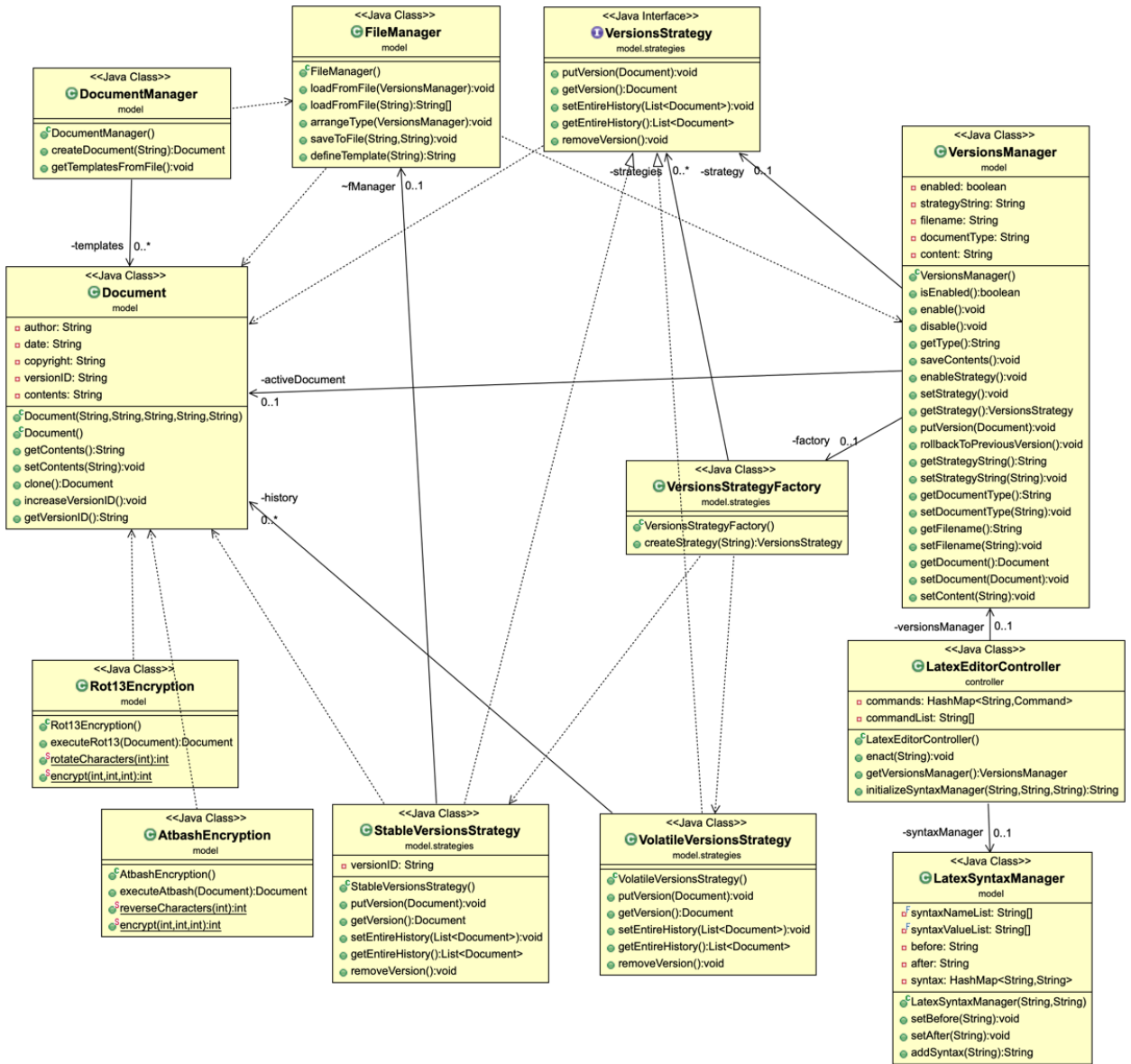
- view package UML



- controller package UML



- model package UML



IMPLEMENTATION

Class Name: <code>Document.class //package model</code>	
Responsibilities	Collaborations
This class gives us the basic properties of the Documents	-----
Class Name: <code>DocumentManager.class //package model</code>	
Responsibilities	Collaborations
This class takes Latex templates from files and stores them in a HashMap in order to give us a deep copy of a specific template.	<ul style="list-style-type: none"> • Document.class
Class Name: <code>VersionsManager.class //package model</code>	
Responsibilities	Collaborations
This class is responsible for active strategy management and document management	<ul style="list-style-type: none"> • VersionsStrategy.class • VersionsStrategyFactory.class • Document.class

Class Name: <i>FileManager.class //package model</i>	
Responsibilities	Collaborations
Added class that is responsible for saving to files and loading from files as well as recognizing the file type.	<ul style="list-style-type: none"> • VersionsManager.class • Document.class
Class Name: <i>LatexSyntaxManager.class //package model</i>	
Responsibilities	Collaborations
New class that is accountable for setting latex fields such as chapters, sections etc.	-----
Class Name: <i>AtbashEncryption.class //package model</i>	
Responsibilities	Collaborations
Extension class that is in charge of implementing Atbash Encryption(-Decryption)	<ul style="list-style-type: none"> • Documents.class
Class Name: <i>Rot13Encryption.class //package model</i>	
Responsibilities	Collaborations
Extension class that is in charge of implementing Rot13 Encryption(-Decryption)	<ul style="list-style-type: none"> • Documents.class

Class Name: <code>VersionsStrategyFactory.class</code> <i>//package model.strategies</i>	
Responsibilities	Collaborations
This class creates the Volatile and Stable Version Strategies and returns the specified strategy.	<ul style="list-style-type: none"> • VersionsStrategy.class • StableVersionsStrategy.class • VolatileVersionsStrategy.class
Class Name: <code>VersionsStrategy.class</code> <i>//package model.strategies</i>	
Responsibilities	Collaborations
This class gives us the interface for the different Version Strategies.	<ul style="list-style-type: none"> • Document.class
Class Name: <code>StableVersionsStrategy.class</code> <i>//package model.strategies</i>	
Responsibilities	Collaborations
This class is responsible for the file-based versioning type.	<ul style="list-style-type: none"> • Document.class • VersionsStrategy.class • FileManager.class

Class Name: VolatileVersionsStrategy.class <i>//package model.strategies</i>	
Responsibilities	Collaborations
This class is responsible for the memory-based versioning type.	<ul style="list-style-type: none"> • Document.class • VersionsStrategy.class
Class Name: LatexEditorController.class <i>//package controller</i>	
Responsibilities	Collaborations
This class enacts a particular command and initializes the latex fields of the document.	<ul style="list-style-type: none"> • CommandFactory.class • Command.class • VersionsManager.class • LatexSyntaxManager.class
Class Name: Command.class <i>//package controller.commands</i>	
Responsibilities	Collaborations
This class specifies the interface for the commands.	-----

Class Name: <i>AbstractCommand.class //package controller.commands</i>	
Responsibilities	Collaborations
A new auxiliary class that provides VersionsManager as a field to the commands	<ul style="list-style-type: none"> • VersionsManager.class • Command.class
Class Name: <i>CommandFactory.class //package controller.commands</i>	
Responsibilities	Collaborations
This class is responsible for the creation of the specified command object.	<ul style="list-style-type: none"> • DocumentManager.class • VersionsManager.class • Command.class • ChangeVersionsStrategyCommand.class • CreateCommand.class • DisableVersionsManagementCommand.class • EditCommand.class • EnableVersionsManagementCommand.class • LoadCommand.class • RollbackToPreviousVersionCommand.class

	<ul style="list-style-type: none"> • SaveCommand.class • LoadRot13Command.class • SaveRot13Command.class • LoadAtbashCommand.class • SaveAtbashCommand.class
Class Name: ChangeVersionsStrategyCommand.class <i>//package controller.commands</i>	
Responsibilities	Collaborations
This class reverts the current VersionStrategy.	<ul style="list-style-type: none"> • AbstractCommand.class • VersionsManager.class
Class Name: CreateCommand.class <i>//package controller.commands</i>	
Responsibilities	Collaborations
This class creates a document according to a specific template.	<ul style="list-style-type: none"> • AbstractCommand.class • VersionsManager.class • documentManager.class • Document.class

Class Name: <code>EditCommand.class</code> <i>//package controller.commands</i>	
Responsibilities	Collaborations
This class updates the content changes of the document	<ul style="list-style-type: none"> • AbstractCommand.class • VersionsManager.class
Class Name: <code>DisableVersionsManagementCommand.class</code> <i>//package controller.commands</i>	
Responsibilities	Collaborations
This class disables the current VersionStrategy.	<ul style="list-style-type: none"> • AbstractCommand.class • VersionsManager.class
Class Name: <code>EnableVersionsManagementCommand.class</code> <i>//package controller.commands</i>	
Responsibilities	Collaborations
This class enables the current VersionStrategy.	<ul style="list-style-type: none"> • AbstractCommand.class • VersionsManager.class

Class Name: LoadCommand.class <i>//package controller.commands</i>	
Responsibilities	Collaborations
This class loads a document from a specific file and finds out its template type from the first line of the document.	<ul style="list-style-type: none"> • AbstractCommand.class • VersionsManager.class • FileManager.class
Class Name: LoadRot13Command.class <i>//package controller.commands</i>	
Responsibilities	Collaborations
This class loads Rot13 encrypted files, decrypts them and loads them to VersionsManager	<ul style="list-style-type: none"> • FileManager.class • VersionsManager.class • Rot13Encryption.class
Class Name: LoadAtbashCommand.class <i>//package controller.commands</i>	
Responsibilities	Collaborations
This class loads Atbash encrypted files decrypts them and loads them to VersionsManager	<ul style="list-style-type: none"> • FileManager.class • VersionsManager.class • AtbashEncryption.class

Class Name: SaveRot13Command.class <i>//package controller.commands</i>	
Responsibilities	Collaborations
This class saves the active Document in Rot13 encryption in disk.	<ul style="list-style-type: none"> • FileManager.class • VersionsManager.class • Document.class • Rot13Encryption.class
Class Name: SaveAtbashCommand.class <i>//package controller.commands</i>	
Responsibilities	Collaborations
This class saves the active Document in Atbash encryption in disk.	<ul style="list-style-type: none"> • FileManager.class • VersionsManager.class • Document.class • AtbashEncryption.class
Class Name: RollBackToPreviousVersionCommand.class <i>//package controller.commands</i>	
Responsibilities	Collaborations
This class removes the current version of the document and set it to the previous version if the version strategy is enabled.	<ul style="list-style-type: none"> • AbstractCommand.class • VersionsManager.class

Class Name: <i>SaveCommand.class //package controller.commands</i>	
Responsibilities	Collaborations
This class saves the document contents to a file.	<ul style="list-style-type: none"> • AbstractCommand.class • VersionsManager.class • FileManager.class
Class Name: <i>MainWindow.class //package view</i>	
Responsibilities	Collaborations
This class creates and initializes the MainWindow (GUI)	<ul style="list-style-type: none"> • LatexEditorController.class • ChooseTemplate.class
Class Name: <i>OpeningWindow.class //package view</i>	
Responsibilities	Collaborations
This class creates and initializes the opening window.	<ul style="list-style-type: none"> • LatexEditorController.class • ChooseTemplate.class
Class Name: <i>ChooseTemplate.class //package view</i>	
Responsibilities	Collaborations

<p>This class creates and initializes the chooseTemplate window.</p>	<ul style="list-style-type: none"> • LatexEditorController.class • OpeningWindow.class • MainWindow.class
--	--

ADDITIONAL CHANGES, FIXES AND EXTENSIONS

GUI FIXES

There were some bugs in the GUI of the first version. In this version we fixed them.

- Added functionality to the “Exit” button in the main window menu bar.
- Fixed the disable button bug in the main window menu bar.
- Removed the “Open existing Document” from the opening window.

EXTENSIONS

In this version we added some extra features.

- Rot13 file encryption and decryption.
- Atbash file encryption and decryption.

Now the user can use the encryption and decryption of his choice to keep his final files safe.

CHANGES

In this version we did some refactoring and some reengineering of the previous source code to improve the quality, the readability and the architecture. Now the project follows the MVC pattern. Additionally, we adjusted the previous tests according to our new design.

VersionsManager

- Moved rollback method to rollbackToPreviousVersion().
- Delete setPreviousVersion method.
- Initialized “enabled” to true. This variable was causing a problem in a test.
- EnableStrategy and changeStrategy were reengineered now we use fewer lines of code than before.
- Now volatile strategy is enabled by default.

DocumentManager

This class loads the templates from the disc now.

FileManger

This new class manages file loads and saves.

LatexSyntaxManager

This new class holds a hashmap with the available latex syntaxes.

LatexEditorController

- Created a list with the command names to load them in the hashmap.

AbstractCommand

- This new class gives an instance of the VersionsManager to the command classes.

LatexEditorView

We deleted this class and splitted its responsibilities to the other classes and adjusted the project to run without this class.

****These are the most import changes we made.**

