

---

---

<LATEX EDITOR PROJECT >

# OVERALL REPORT

VERSION <1.0>

<IOANNA PAPAGIANNI 2790, NIKOS PANTELIDIS 2787>

---

---

# TABLE OF CONTENTS

Introduction	4
Design Recovery	<b>Error! Bookmark not defined.</b>
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.

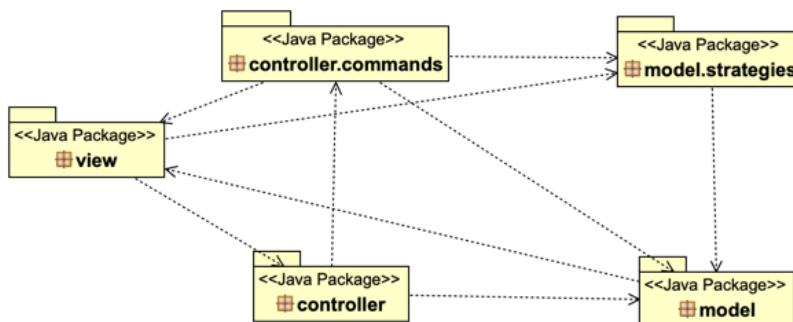
There are two parts, **design recovery** and **re-engineered design**. Design recovery, shows the design and logic of the original given project, it detects the collaborations among classes and eventually, spots functional and non-functional issues.

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

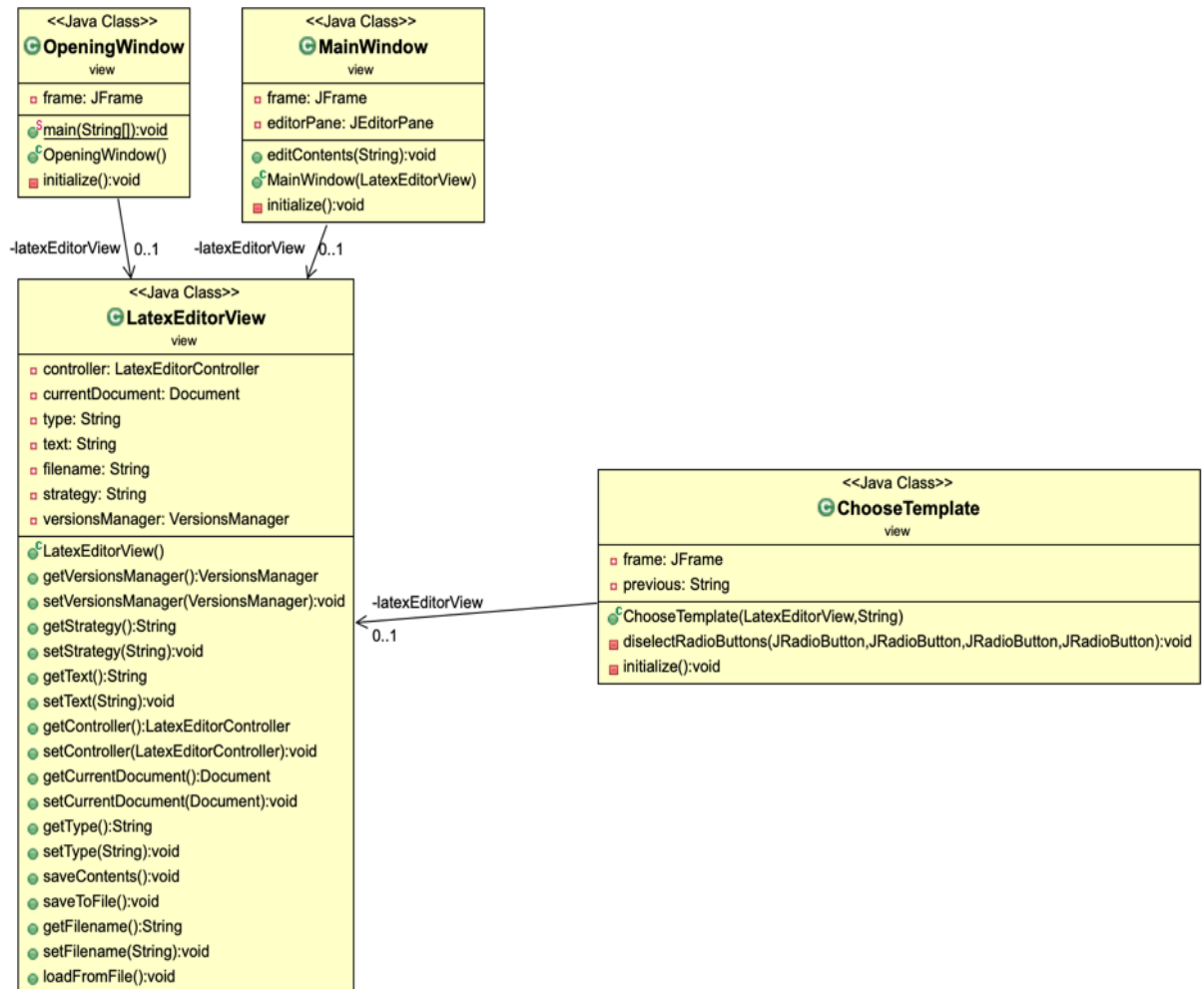
### ARCHITECTURE

packages UML

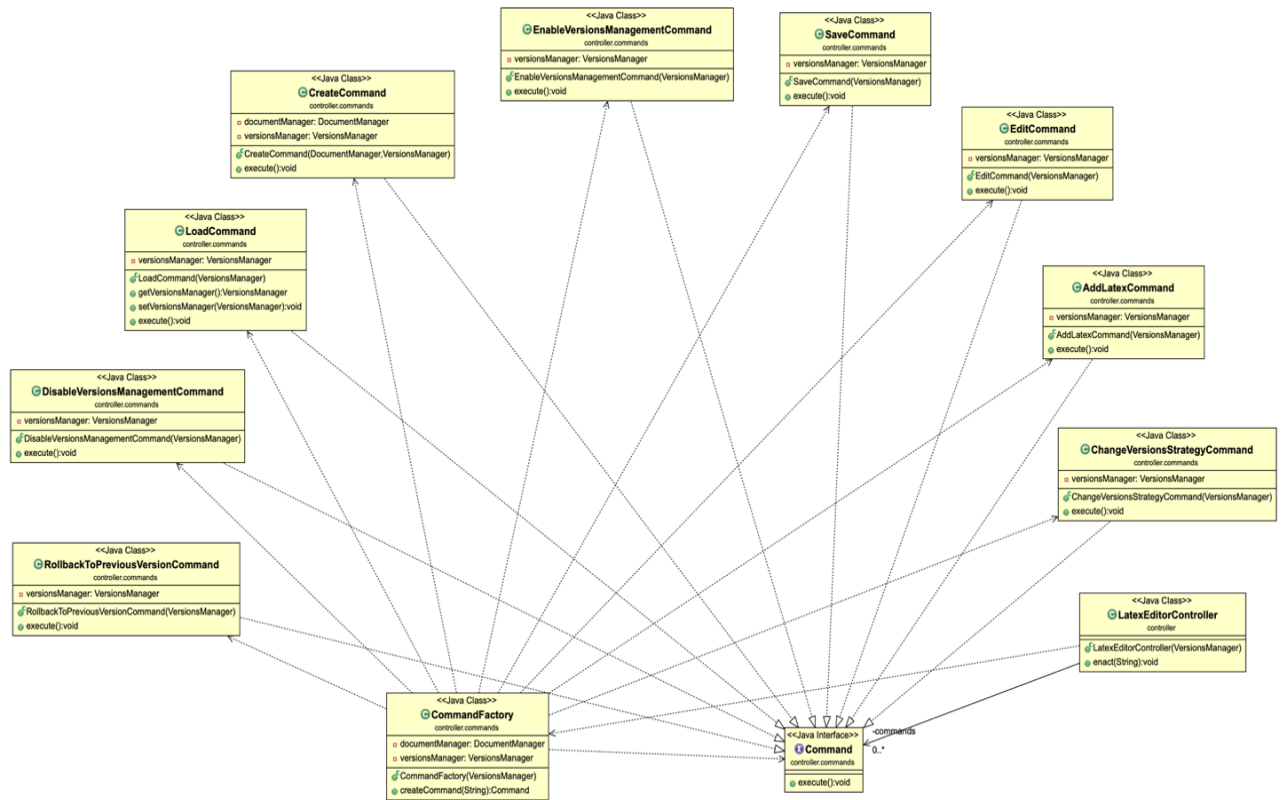


## DETAILED DESIGN

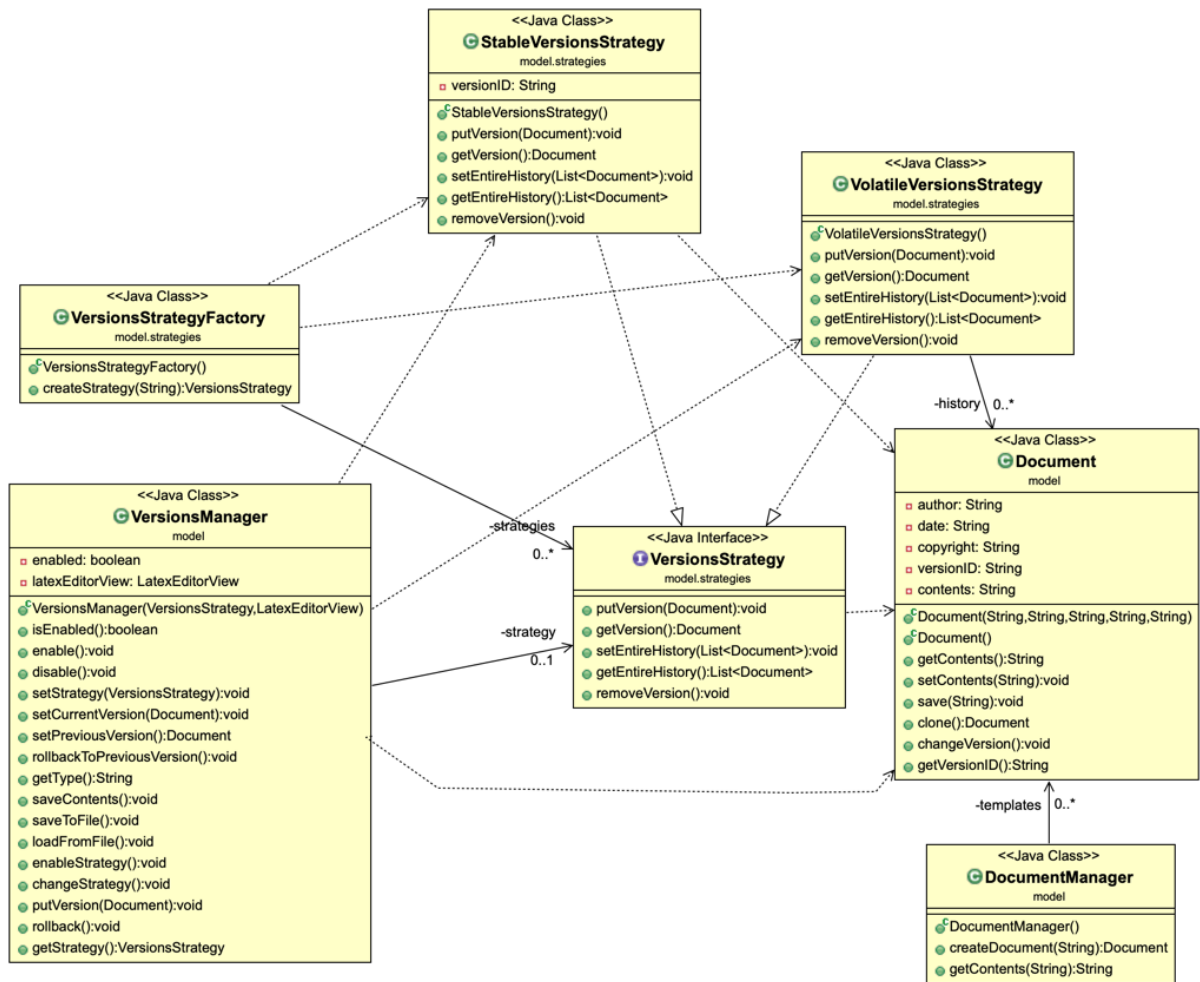
### view package UML



- controller package UML



- model package UML



## IMPLEMENTATION

<b>Class Name:</b> <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.	-----
<b>Class Name:</b> <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"> <li>• <code>Document.class</code></li> </ul>
<b>Class Name:</b> <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"> <li>• <code>LatexEditorView.class</code></li> <li>• <code>VersionsStrategy.class</code></li> <li>• <code>StableVersionsStrategy.class</code></li> <li>• <code>VolatileVersionsStrategy.class</code></li> <li>• <code>Document.class</code></li> </ul>



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

<b>Class Name:</b> <a href="#">LatexEditorController.class</a> //package controller	
<b>Responsibilities</b>	<b>Collaborations</b>
This class stores the available commands in a HashMap and enacts a particular command.	<ul style="list-style-type: none"> <li>• <a href="#">CommandFactory.class</a></li> <li>• <a href="#">VersionsManager.class</a></li> <li>• <a href="#">Command.class</a></li> </ul>
<b>Class Name:</b> <a href="#">Command.class</a> //package controller.commands	
<b>Responsibilities</b>	<b>Collaborations</b>
This class specifies the interface for the commands.	-----
<b>Class Name:</b> <a href="#">CommandFactory.class</a> //package controller.commands	
<b>Responsibilities</b>	<b>Collaborations</b>
This class is responsible for the creation of the specified command object.	<ul style="list-style-type: none"> <li>• <a href="#">DocumentManager.class</a></li> <li>• <a href="#">VersionsManager.class</a></li> </ul>
<b>Class Name:</b> <a href="#">AddLatexCommand.class</a> //package controller.commands	
<b>Responsibilities</b>	<b>Collaborations</b>
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"> <li>• <a href="#">Command.class</a></li> <li>• <a href="#">VersionsManager.class</a></li> </ul>

<b>Class Name:</b> <a href="#">ChangeVersionsStrategyCommand.class //package controller.commands</a>	
<b>Responsibilities</b>	<b>Collaborations</b>
This class reverts the current VersionStrategy.	<ul style="list-style-type: none"> <li>• Command.class</li> <li>• VersionsManager.class</li> </ul>
<b>Class Name:</b> <a href="#">CreateCommand.class //package controller.commands</a>	
<b>Responsibilities</b>	<b>Collaborations</b>
This class creates a document according to a specific template.	<ul style="list-style-type: none"> <li>• Command.class</li> <li>• VersionsManager.class</li> <li>• documentManager.class</li> </ul>
<b>Class Name:</b> <a href="#">EditCommand.class //package controller.commands</a>	
<b>Responsibilities</b>	<b>Collaborations</b>
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"> <li>• Command.class</li> <li>• VersionsManager.class</li> </ul>
<b>Class Name:</b> <a href="#">DisableVersionsManagementCommand.class //package controller.commands</a>	
<b>Responsibilities</b>	<b>Collaborations</b>
This class disables the current VersionStrategy.	<ul style="list-style-type: none"> <li>• Command.class</li> </ul>

	<ul style="list-style-type: none"> <li>• VersionsManager.class</li> </ul>
<b>Class Name:</b> <a href="#">EnableVersionsManagementCommand.class</a> //package controller.commands	
<b>Responsibilities</b>	<b>Collaborations</b>
This class enables the current VersionStrategy.	<ul style="list-style-type: none"> <li>• Command.class</li> <li>• VersionsManager.class</li> </ul>
<b>Class Name:</b> <a href="#">LoadCommand.class</a> //package controller.commands	
<b>Responsibilities</b>	<b>Collaborations</b>
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"> <li>• Command.class</li> <li>• VersionsManager.class</li> </ul>
<b>Class Name:</b> <a href="#">RollBackToPreviousVersionCommand.class</a> //package controller.commands	
<b>Responsibilities</b>	<b>Collaborations</b>
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"> <li>• Command.class</li> <li>• VersionsManager.class</li> </ul>
<b>Class Name:</b> <a href="#">SaveCommand.class</a> //package controller.commands	
<b>Responsibilities</b>	<b>Collaborations</b>
This class saves the document to a file.	<ul style="list-style-type: none"> <li>• Command.class</li> </ul>

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

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

## 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:**

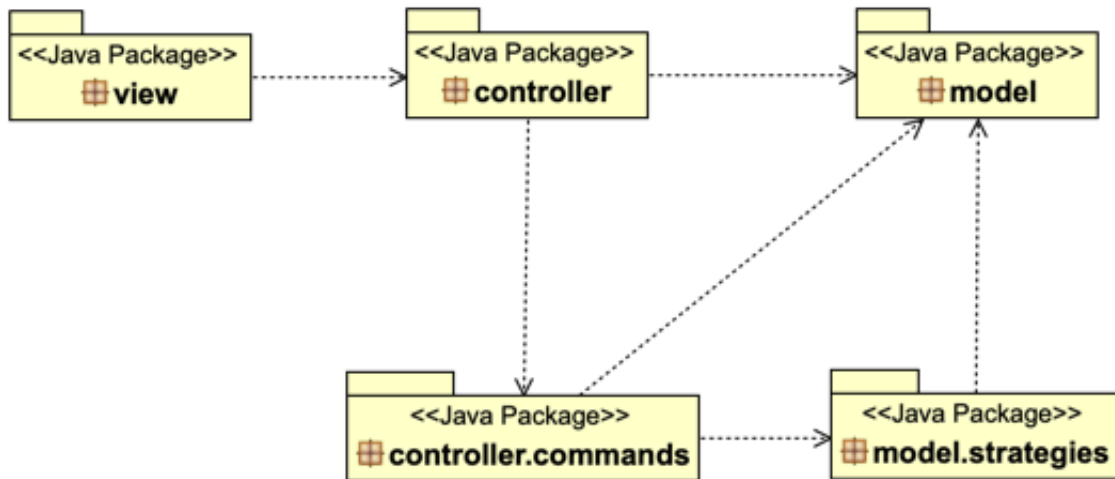
- Wrong-placed class: This class contains application logic that should be in model package.
- Misleading name: It has nothing to do with view.
- Too many responsibilities: It has many setter and getter methods that are already implemented in other classes and also saving and loading methods are implemented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



## General severe issues:

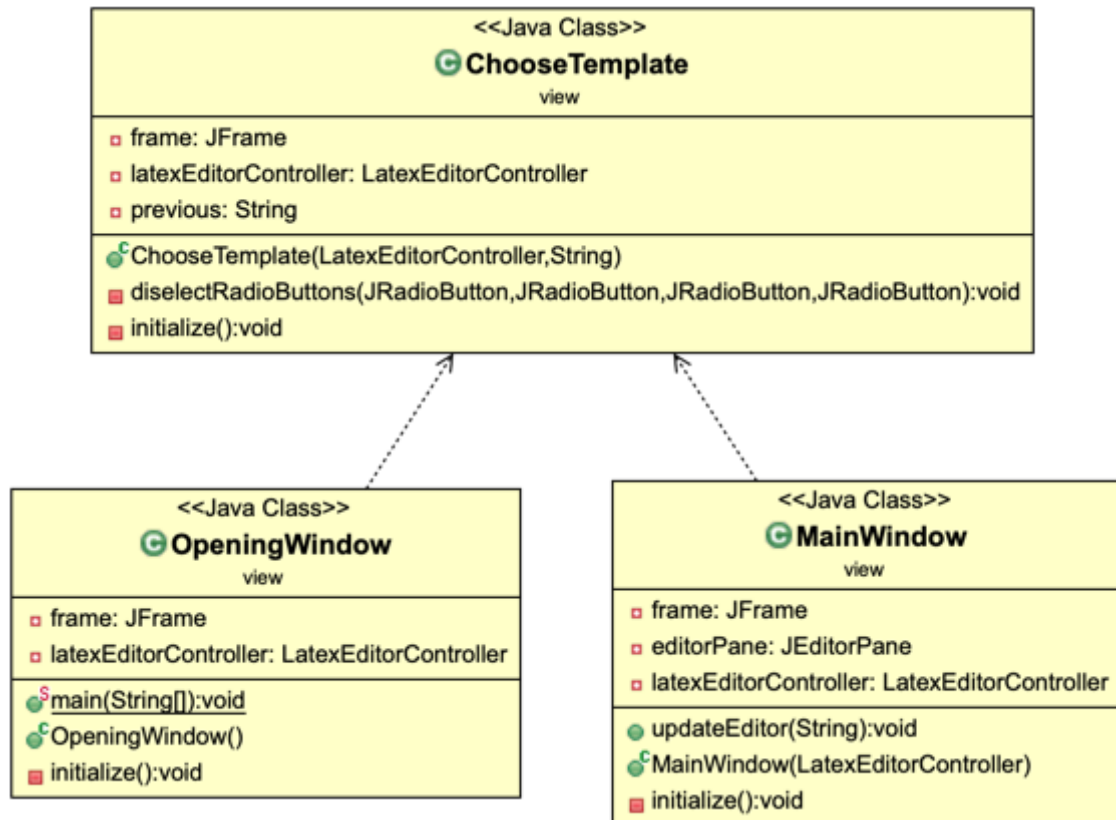
- There are many misleading methods and field names that could be altered.
- “Disable” button has a bug, it cannot be unmarked but it works.
- “Exit” button from the main bar (File-Exit) does not work.
- “Open Existing Document” button opens the file explorer but it cannot load any file.
- In order to 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

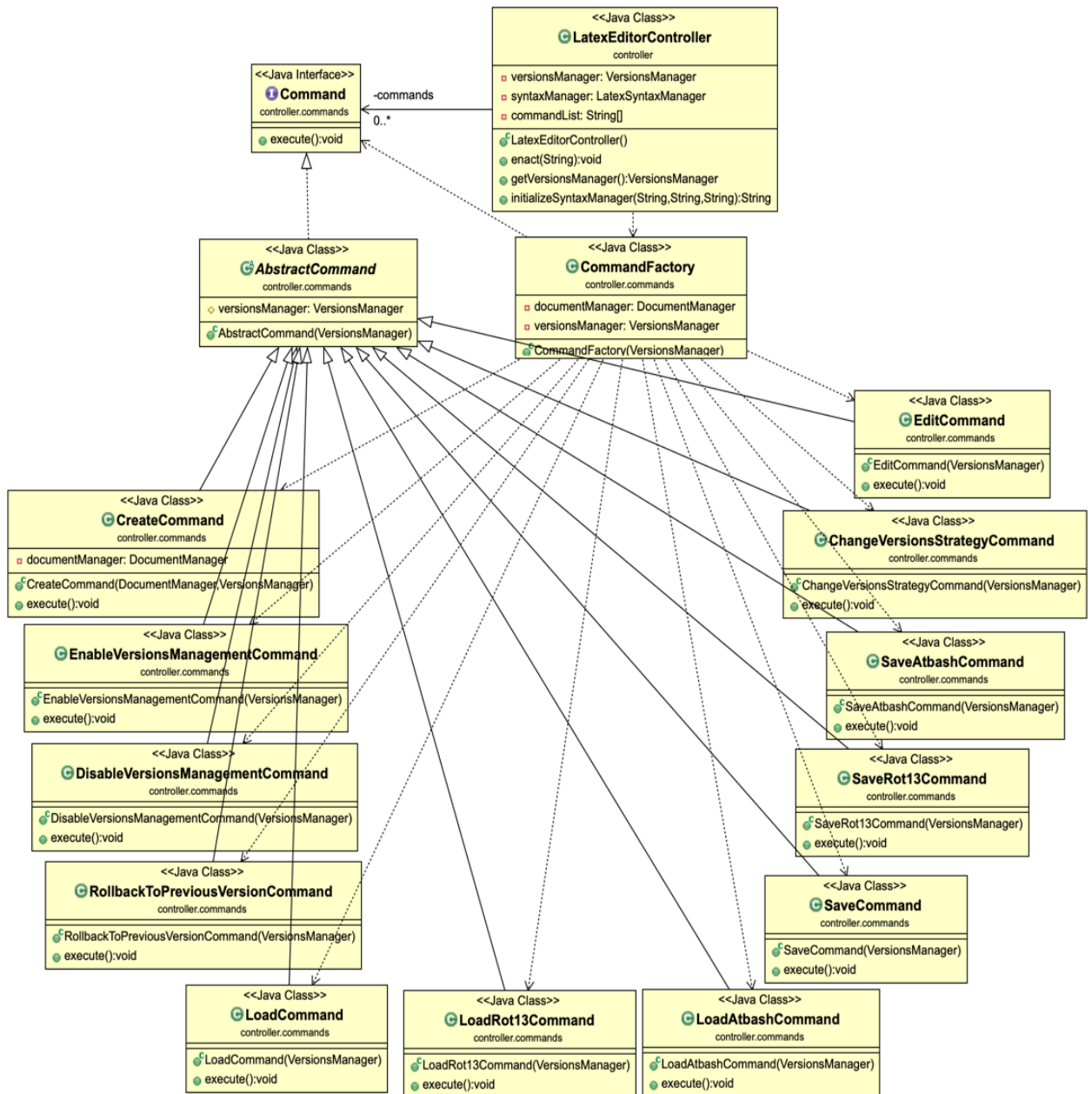


## DETAILED DESIGN

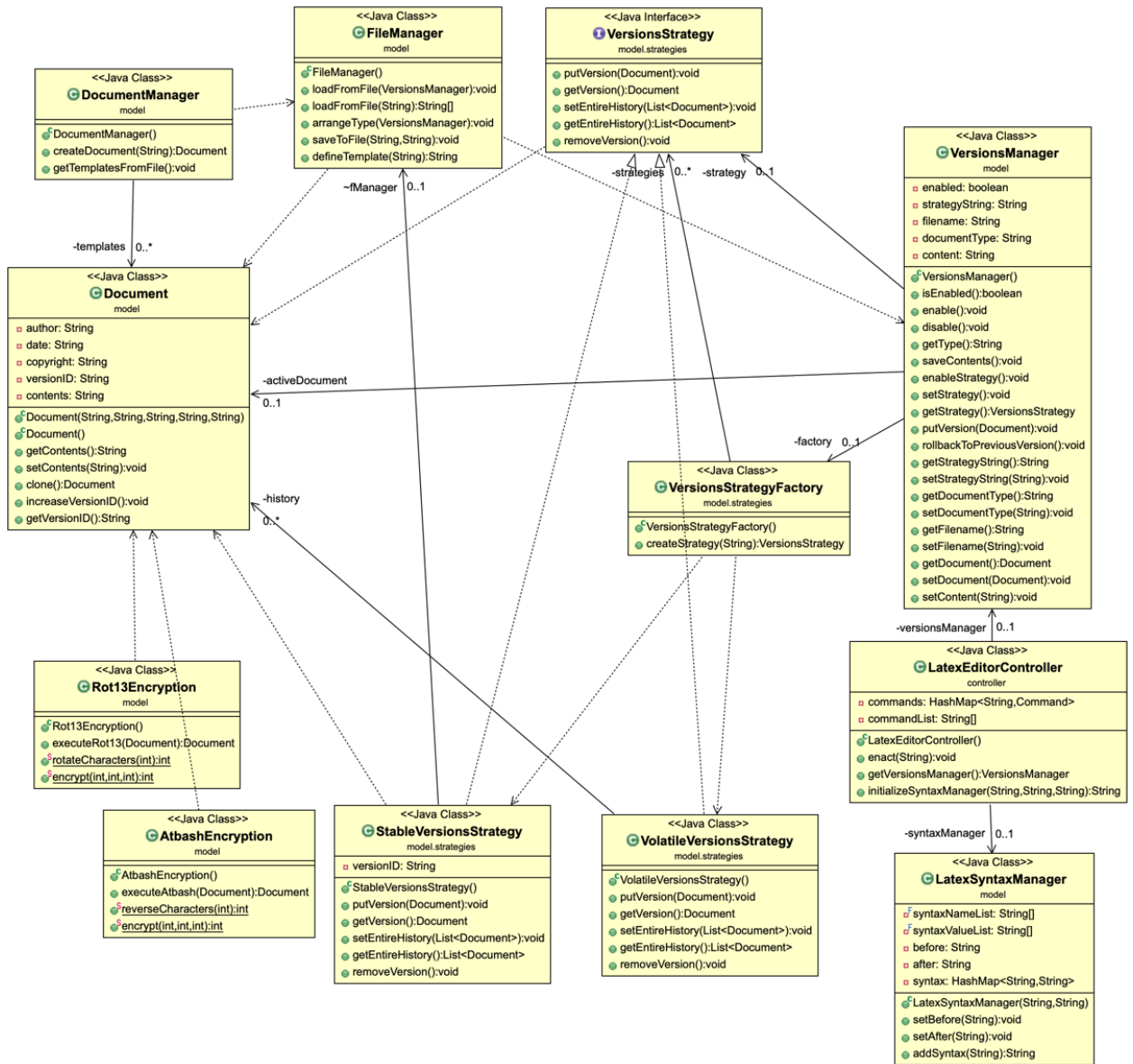
- view package UML



- controller package UML



- model package UML



## IMPLEMENTATION

<b>Class Name:</b> <i>Document.class //package model</i>	
<b>Responsibilities</b>	<b>Collaborations</b>
This class gives us the basic properties of the Documents	-----
<b>Class Name:</b> <i>DocumentManager.class //package model</i>	
<b>Responsibilities</b>	<b>Collaborations</b>
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"> <li>• Document.class</li> </ul>
<b>Class Name:</b> <i>VersionsManager.class //package model</i>	
<b>Responsibilities</b>	<b>Collaborations</b>
This class is responsible for active strategy management and document management	<ul style="list-style-type: none"> <li>• VersionsStrategy.class</li> <li>• VersionsStrategyFactory.class</li> <li>• Document.class</li> </ul>

<b>Class Name:</b> <i>FileManager.class //package model</i>	
<b>Responsibilities</b>	<b>Collaborations</b>
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"> <li>• VersionsManager.class</li> <li>• Document.class</li> </ul>
<b>Class Name:</b> <i>LatexSyntaxManager.class //package model</i>	
<b>Responsibilities</b>	<b>Collaborations</b>
New class that is accountable for setting latex fields such as chapters, sections etc.	-----
<b>Class Name:</b> <i>AtbashEncryption.class //package model</i>	
<b>Responsibilities</b>	<b>Collaborations</b>
Extension class that is in charge of implementing Atbash Encryption(-Decryption)	<ul style="list-style-type: none"> <li>• Documents.class</li> </ul>
<b>Class Name:</b> <i>Rot13Encryption.class //package model</i>	
<b>Responsibilities</b>	<b>Collaborations</b>
Extension class that is in charge of implementing Rot13 Encryption(-Decryption)	<ul style="list-style-type: none"> <li>• Documents.class</li> </ul>

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



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

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

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

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

<b>Class Name:</b> <code>LoadCommand.class</code> <i>//package controller.commands</i>	
<b>Responsibilities</b>	<b>Collaborations</b>
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"> <li>• AbstractCommand.class</li> <li>• VersionsManager.class</li> <li>• FileManager.class</li> </ul>
<b>Class Name:</b> <code>LoadRot13Command.class</code> <i>//package controller.commands</i>	
<b>Responsibilities</b>	<b>Collaborations</b>
This class loads Rot13 encrypted files, decrypts them and loads them to VersionsManager	<ul style="list-style-type: none"> <li>• FileManager.class</li> <li>• VersionsManager.class</li> <li>• Rot13Encryption.class</li> </ul>
<b>Class Name:</b> <code>LoadAtbashCommand.class</code> <i>//package controller.commands</i>	
<b>Responsibilities</b>	<b>Collaborations</b>
This class loads Atbash encrypted files decrypts them and loads them to VersionsManager	<ul style="list-style-type: none"> <li>• FileManager.class</li> <li>• VersionsManager.class</li> <li>• AtbashEncryption.class</li> </ul>

<b>Class Name:</b> <code>SaveRot13Command.class</code> <i>//package controller.commands</i>	
<b>Responsibilities</b>	<b>Collaborations</b>
This class saves the active Document in Rot13 encryption in disk.	<ul style="list-style-type: none"> <li>• FileManager.class</li> <li>• VersionsManager.class</li> <li>• Document.class</li> <li>• Rot13Encryption.class</li> </ul>
<b>Class Name:</b> <code>SaveAtbashCommand.class</code> <i>//package controller.commands</i>	
<b>Responsibilities</b>	<b>Collaborations</b>
This class saves the active Document in Atbash encryption in disk.	<ul style="list-style-type: none"> <li>• FileManager.class</li> <li>• VersionsManager.class</li> <li>• Document.class</li> <li>• AtbashEncryption.class</li> </ul>
<b>Class Name:</b> <code>RollBackToPreviousVersionCommand.class</code> <i>//package controller.commands</i>	
<b>Responsibilities</b>	<b>Collaborations</b>
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"> <li>• AbstractCommand.class</li> <li>• VersionsManager.class</li> </ul>

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

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