

MYE004 - SOFTWARE DEVELOPMENT II

L^AT_EX EDITOR PPROJECT (REFACTORED)

OVERALL REPORT

VERSION 1.0

Παναγιώτης Βουζαλής, AM 2653

TABLE OF CONTENTS

Introduction	3
Refactored Design	3
Architecture	4
Detailed Design	5
Implementation	11

INTRODUCTION

The goal of this project was to refactor a given program that has been developed by other developers. Going into detail, we discovered that this goal was composed of multiple challenging objectives. Objective no. 1 was to find instances where bad design patterns were used and try our best to fix them using proper design patterns.

Objective no. 2 was to find instances where bad coding practices were used and again try our best to fix them using proper coding practices. Objective no. 3 was to find bugs or misleading menus and buttons in the UX and rename them to something that makes more sense to the average user. Objective no. 4 was to showcase that the user stories of the original application were still working after the refactoring process. Finally, objective no. 5 was a critical objective that had to do with testing our refactored code and was tackled by creating proper jUnit tests. Throughout this project, our choices regarding the refactoring of the existing code and implementing the extensions that we were asked to do were done keeping in mind coding principles, design patterns, and refactoring techniques we were taught in class such as the Open-Closed Principle (OPC), the Model-View-Controller Design Pattern (MVC), the Singleton Pattern, and various refactorings such as Duplicate Code, Dead Code, Large Class/Method, Extract Class/Method, Middle Man, Message Chains and Displaced Responsibilities.

REFACTORED DESIGN

Demo video is located here (Google Drive link): [Click me!](#)

In the demo video:

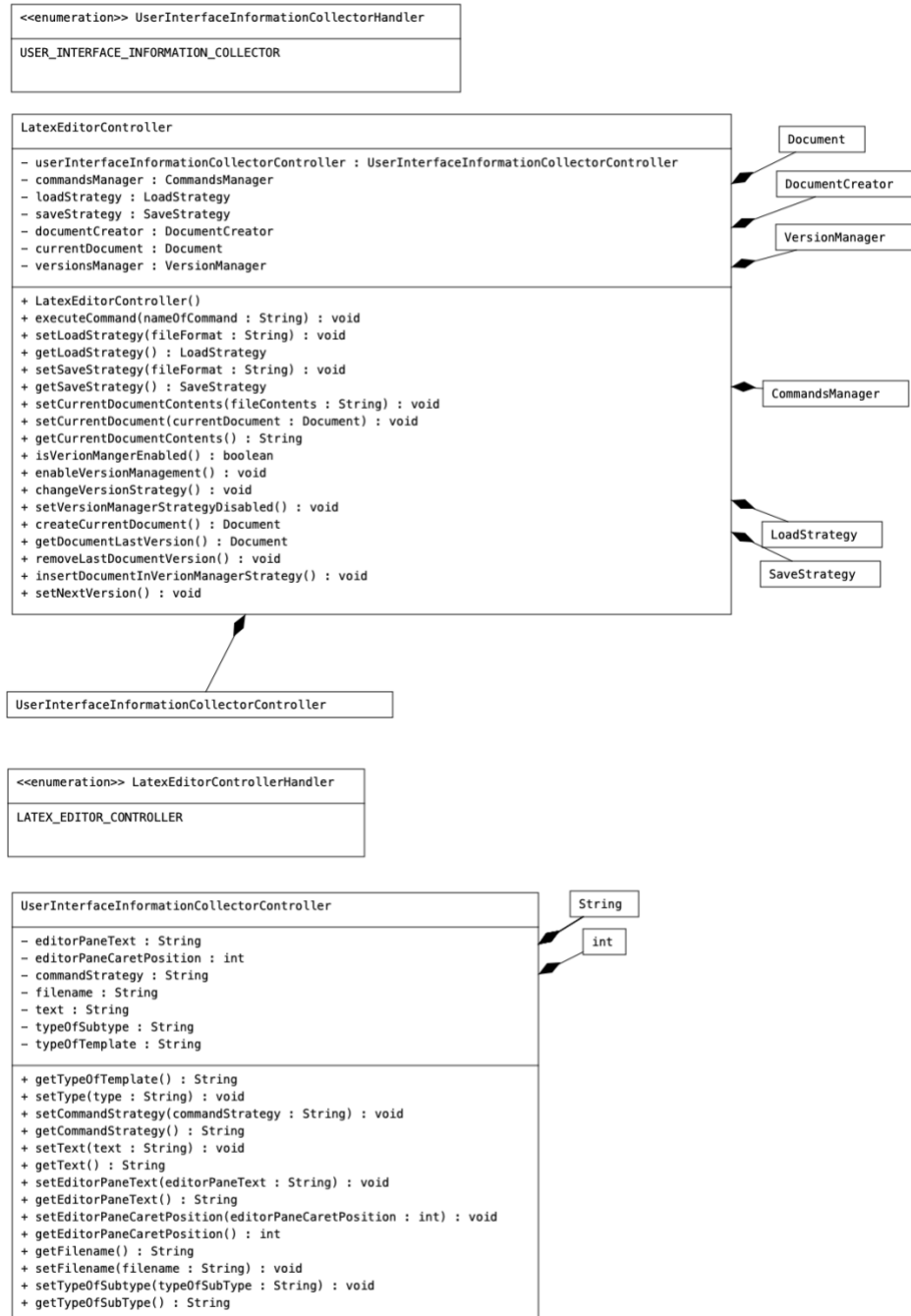
- We showcase that the user stories of the original application are still working after the refactoring (minutes 0:00 – 19:00)
 - We discuss how we addressed the different reengineering problems of the old design (minutes 19:00 – 33:00)
 - We demonstrate the execution of the jUnit tests we prepared for the project (minutes 33:00 – 39:00)

ARCHITECTURE

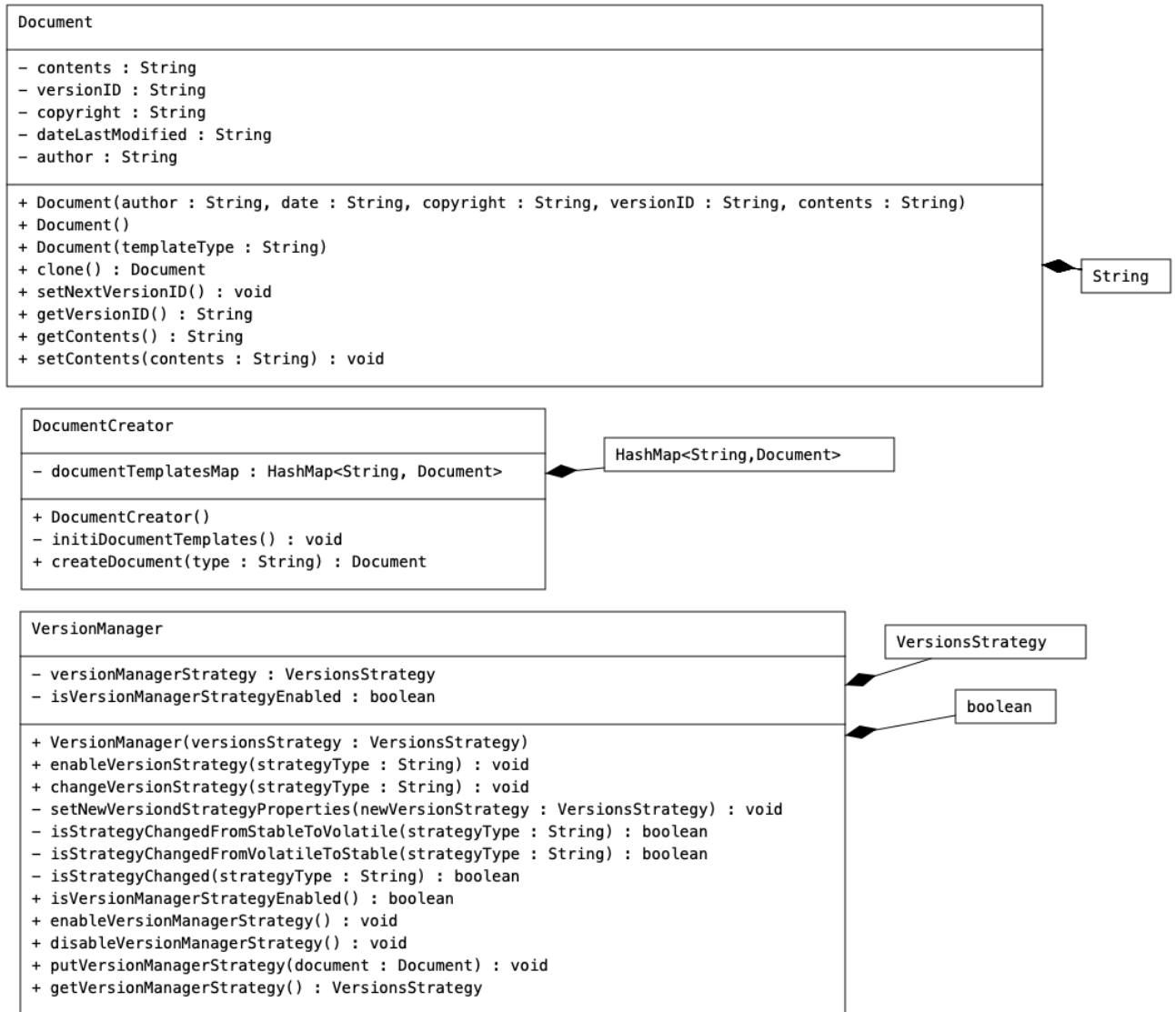
- Overall UML package diagram

DETAILED DESIGN

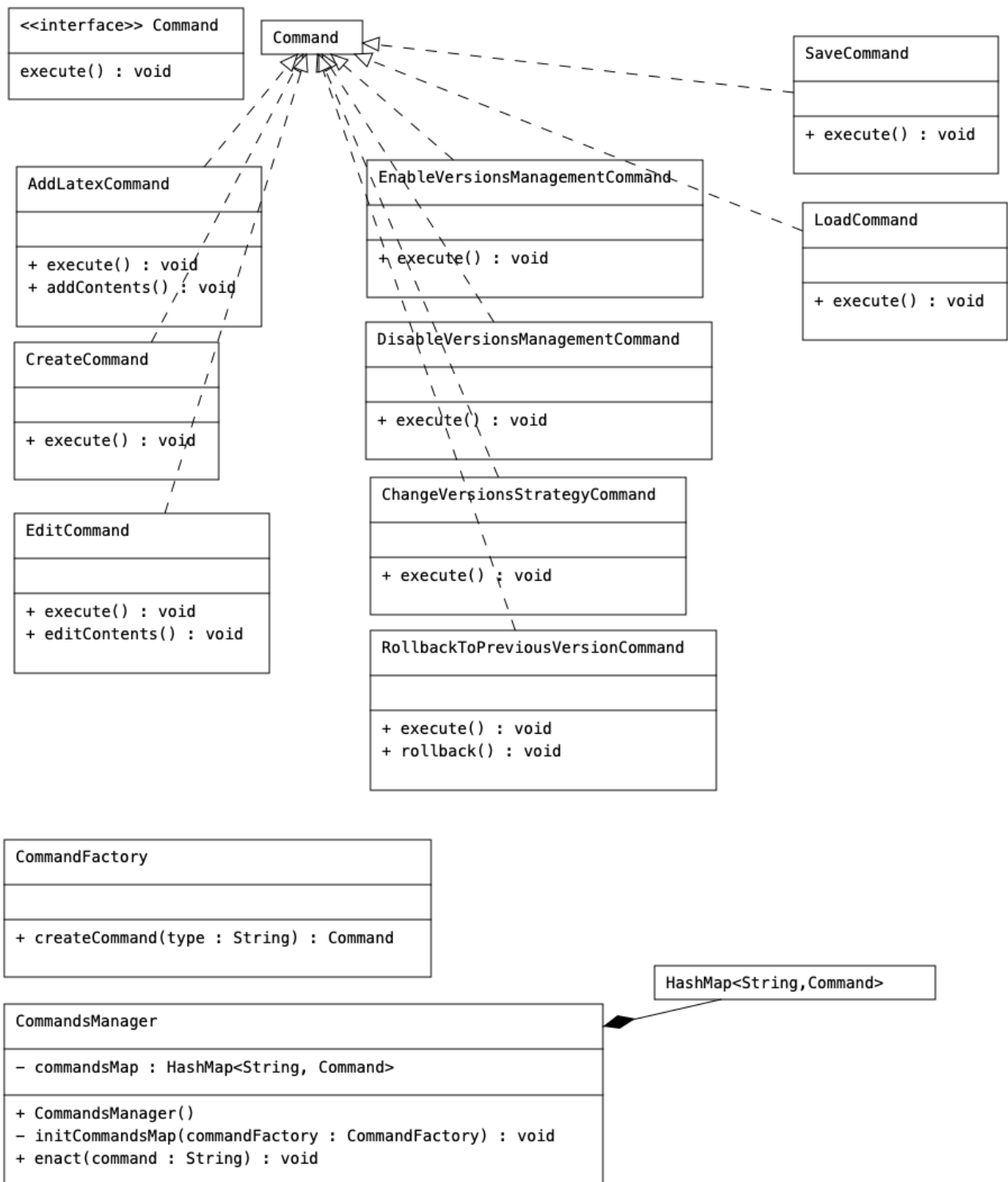
- controller package class diagram



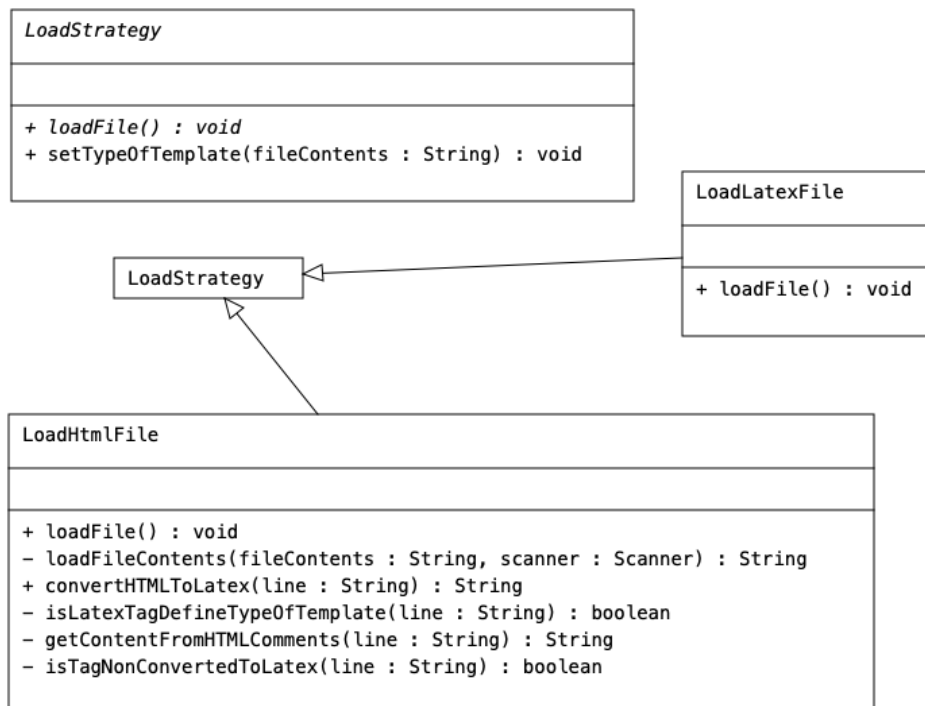
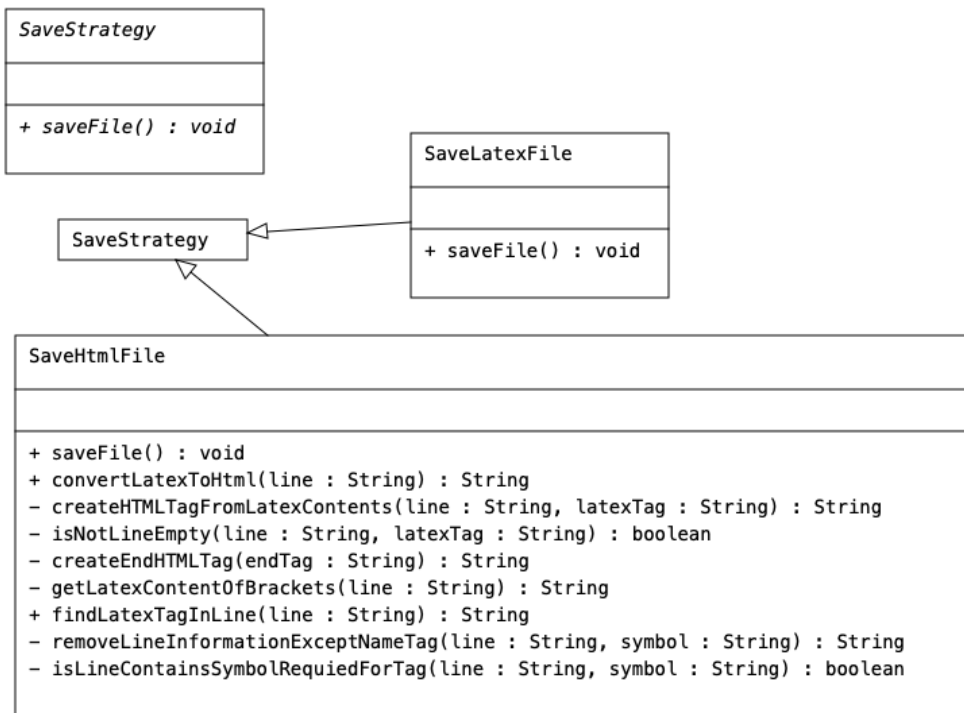
- model package class diagram



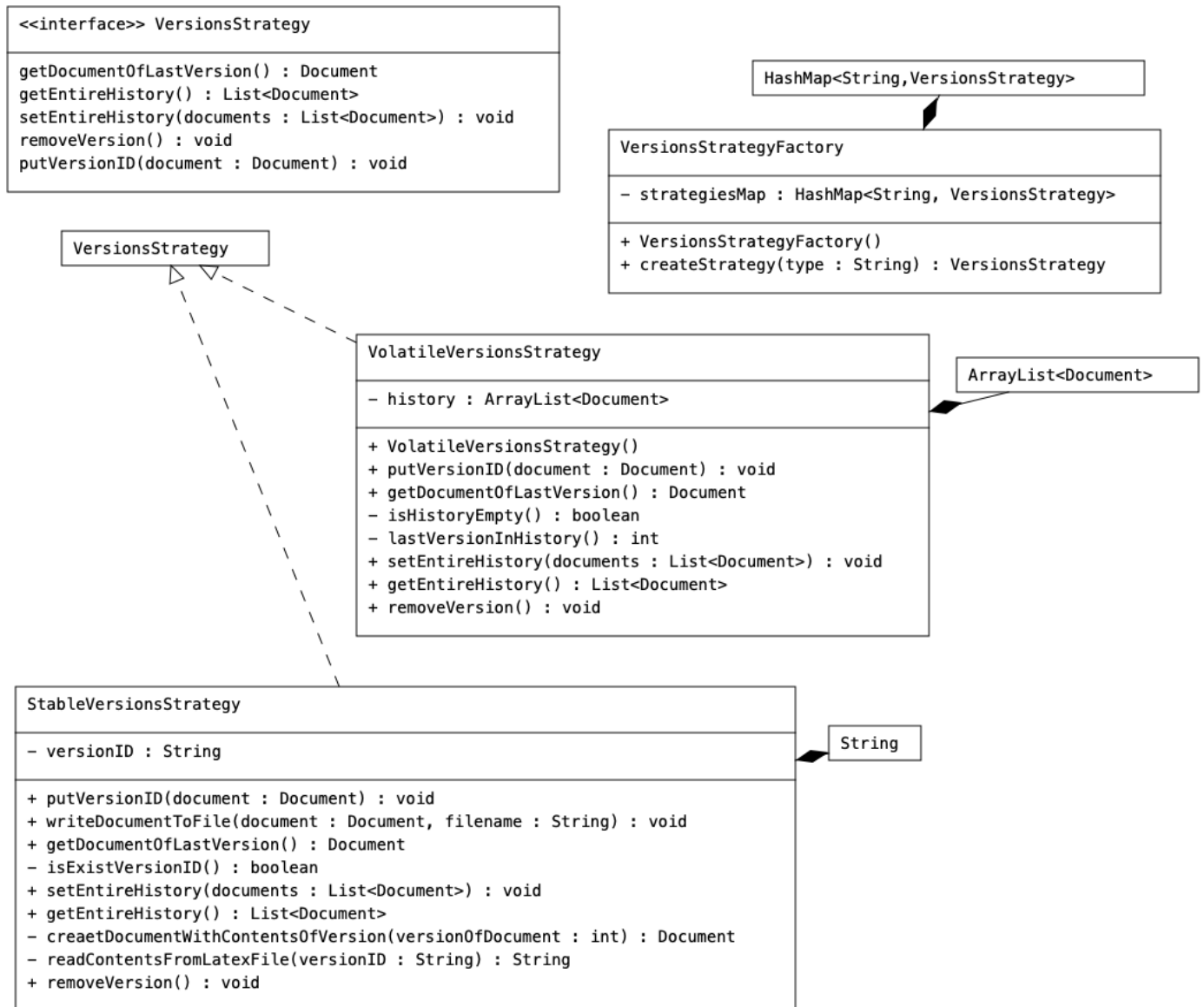
- model.commands package class diagram



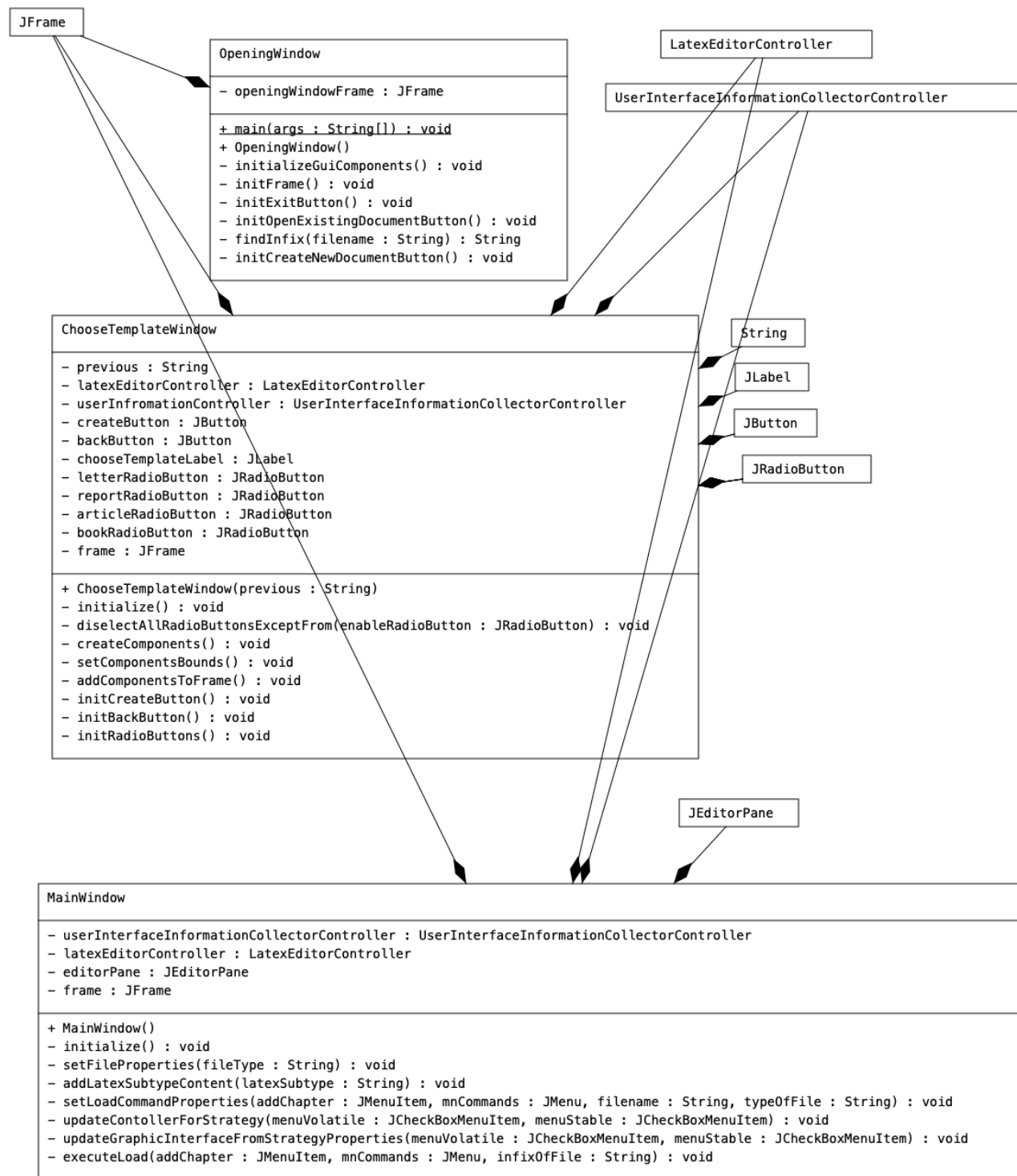
- model.save.load.strategies package class diagram



- model.version.strategies package class diagram



- view package class diagram



IMPLEMENTATION

- Class Responsibility Collaborations (CRC) Cards

(listed in alphabetical order as they are shown in the Eclipse IDE – packages first, classes second)

Package: constants - Class Name: StringConstants	
Responsibilities	Collaborations
<ul style="list-style-type: none">• This class contains all the string constants that are used throughout all classes.• This class is responsible for storing those constants and making them extremely easily accessible from any and every part of the code.• This class performs only one task: present whoever is calling upon it a static, immutable string constant.• The class is structured in such a way, so that when used properly, the developer will be able to write code cleanly and with fewer duplications , leading to a more robust and readable code structure.	<ul style="list-style-type: none">• Other classes that want to use (some of) the fields of this class simply import it.• Due to the nature of this class, it is independent but is associated with the classes that call upon its fields.• Dependants of this class: model.Document model.DocumentCreator model.commands.AddLatexCommand model.commands.CommandsManager model.save.load.strategies.Load model.save.load.strategies.SaveHtmlFile

Package: controller - Class Name: LatexEditorController (former LatexEditorView)	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This class serves as a middle layer that decouples the GUI from the model classes. Consequently, this class contains methods that aid in performing exactly this task. 	<ul style="list-style-type: none"> Dependencies: <ul style="list-style-type: none"> model.Document model.DocumentCreator model.VersionManager model.save.load.strategies. - abstract LoadStrategy - abstract SaveStrategy model.commands.CommandManager controller.UserInterfaceInformationCollectorController Dependants: <ul style="list-style-type: none"> controller.LatexEditorControllerHandler view.ChooseTemplateWindow view.MainWindow

Package: controller - Class Name: enum LatexEditorControllerHandler	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This class is responsible for being the single point of access to the LatexEditorController objects that are used by classes in the following packages: model.commands model.save.load.strategies view This class is a manifestation of the Singleton Design Pattern. 	<ul style="list-style-type: none"> This class creates instances of the LatexEditorController class. Thus it has the following dependencies: controller.LatexEditorController Dependants are classes that call upon a LatexEditorControllerHandler object. Namely: model.commands. - AddLatexCommand - ChangeVersionsStrategyCommand - CreateCommand - DisableVersionsManagementCommand - EditCommand EnableVersionsManagementCommand - LoadCommand - RollbackToPreviousVersionCommand - SaveCommand model.save.load.strategies. - LoadHtmlFile - LoadLatexFile - SaveHtmlFile - SaveLatexFile view.ChooseTemplateWindow view.MainWindow view.OpeningWindow

Package: controller - Class Name: UserInterfaceInformationCollectorController	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This class is responsible for collecting information from the view side of the MVC Design Pattern and pass this information to the controller.LatexEditorController class. Consequently, this class contains methods that aid in performing exactly this task. 	<ul style="list-style-type: none"> Dependencies: controller.LatexEditorController Dependants: view.ChooseTemplateWindow view.MainWindow

Package: controller - Class Name: enum <code>UserInfoInformationCollectorHandler</code>	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This class is responsible for being the single point of access to the <code>UserInfoInformationCollectorController</code> objects that are used by classes in the following packages: <code>model.commands</code> <code>model.save.load.strategies</code> <code>view</code> and in the <code>controller.LatexEditorController</code> class This class is a manifestation of the Singleton Design Pattern. 	<ul style="list-style-type: none"> This class creates instances of the <code>UserInfoInformationCollectorController</code> class. Thus it has the following dependencies: <code>controller.LatexEditorController</code> Dependants are classes that call upon a <code>UserInfoInformationCollectorControllerHandler</code> object. Namely: <code>controller.LatexEditorController</code> <code>model.commands.AddLatexCommand</code> <code>model.commands.EditCommand</code> <code>model.save.load.strategies</code>. - <code>LoadHtmlFile</code> - <code>LoadLatexFile</code> - abstract <code>LoadStrategy</code> - <code>SaveHtmlFile</code> - <code>SaveLatexFile</code> <code>view.ChooseTemplateWindow</code> <code>view.MainWindow</code> <code>view.OpeningWindow</code>

Package: model - Class Name: Document	
Responsibilities	Collaborations
<ul style="list-style-type: none"> • This class materializes the actual “document” object our LaTeX editor is supposed to edit (book, report, article, letter, blank document). • This class is responsible for keeping track of all the document-related fields such as a document’s author, latest modification date, copyright info, version id, and of course, its contents. Consequently, it contains methods that “set and get” those fields. 	<ul style="list-style-type: none"> • Dependencies of this class: constants.StringConstants. • This class is a vital part of the codebase and as such, it is associated with lots of other classes that call upon it. • Dependants of this class: controller.LatexEditorController model.DocumentCreator model.VersionManager model.commands. - RollbackToPreviousVersionCommand model.version.strategies. - StableVersionStrategy - interface VersionsStrategy - VolatileVersionsStrategy

Package: model - Class Name: DocumentCreator (former DocumentManager)	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This class, as its name suggests, is responsible for creating and returning the various “document” objects our LaTeX editor is supposed to edit (book, report, article, letter, blank document). 	<ul style="list-style-type: none"> This class either creates instances of another class or simply uses instances of other classes. Specifically it has the following dependencies: model.Document (creates) constants.StringConstants (uses) This class is also associated with other classes that call upon it. Dependants: controller.LatexEditorController

Package: model - Class Name: VersionManager	
Responsibilities	Collaborations
<ul style="list-style-type: none"> • The responsibility of this class is version management of the documents the LaTeX editor handles. • This class is responsible for enabling/disabling/switching between the different versioning strategies that are offered by the LaTeX editor. Consequently, it contains methods that perform exactly these tasks. 	<ul style="list-style-type: none"> • This class creates and uses instances of the following classes and thus has a dependency on them: model.version.strategies. - VersionsStrategy (interface) - VolatileVersionsStrategy - StableVersionsStrategy • This class is also associated with the class that creates it. • Dependants: controller.LatexEditorController

Package: model.commands - Class Name: AddLatexCommand	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This class is responsible for executing the command that adds new content to the document the LaTeX editor is currently handling. Consequently, it contains one method that performs exactly this task. 	<ul style="list-style-type: none"> This class uses instances of the following classes and thus has a dependency on them: constants.StringConstants controller. <ul style="list-style-type: none"> - LatexEditorControllerHandler - UserInterfaceInformationCollectorHandler This class is also associated with the class that creates it and the interface whose methods it implements. Dependants: <ul style="list-style-type: none"> model.commands.CommandFactory Interfaces this class implements: <ul style="list-style-type: none"> model.commands.Command

Package: model.commands - Class Name: ChangeVersionsStrategyCommand	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This class is responsible for executing the command that modifies the versioning strategy of the document the LaTeX editor is currently handling. Consequently, it contains one method that performs exactly this task. 	<ul style="list-style-type: none"> This class uses instances of the following classes and thus has a dependency on them: controller.LatexEditorControllerHandler This class is also associated with the class that creates it and the interface whose methods it implements. Dependants: model.commands.CommandFactory Interfaces this class implements: model.commands.Command

Package: model.commands - Class Name: interface Command	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This interface is solely responsible for offering methods that must be implemented from classes that are associated with it. Namely it offers the execute() method. 	<ul style="list-style-type: none"> This interface is associated with every other class that implements it. Classes that implement this interface: <ul style="list-style-type: none"> - AddLatexCommand - ChangeVersionsStrategyCommand - CreateCommand - DisableVersionsManagementCommand - EditCommand - EnableVersionsManagementCommand - LoadCommand - RollbackToPreviousVersionCommand - SaveCommand

Package: model.commands - Class Name: CommandFactory	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This class is a “factory” class. This class has only one responsibility: create and return various new objects to the classes that call upon it. 	<ul style="list-style-type: none"> This class creates and returns new instances of other classes Specifically it has the following dependencies: model.commands. <ul style="list-style-type: none"> - AddLatexCommand - ChangeVersionsStrategyCommand - CreateCommand - DisableVersionsManagementCommand - EditCommand - EnableVersionsManagementCommand - LoadCommand - RollbackToPreviousVersionCommand - SaveCommand This class is also associated with the class that creates it. Dependants: model.commands.CommandsManager

Package: model.commands - Class Name: CommandsManager (former LatexEditorController)	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This class is responsible for maintaining a map of the current commands that are supported by the LaTeX editor. It is also responsible for “enacting” these commands, i.e. “calling” each command when needed and ordering it to “execute” itself. 	<ul style="list-style-type: none"> This class creates and uses instances of another class, thus it has the following dependencies: model.commands.CommandFactory constants.StringConstants It also has a dependency towards every command class object that it stores on the map it maintains: model.commands. <ul style="list-style-type: none"> - AddLatexCommand - ChangeVersionsStrategyCommand - CreateCommand - DisableVersionsManagementCommand - EditCommand - EnableVersionsManagementCommand - LoadCommand - RollbackToPreviousVersionCommand - SaveCommand This class is also associated with the class that creates it. Dependants: controller.LatexEditorController

Package: model.commands - Class Name: CreateCommand	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This class is responsible for executing the command that creates a new document for the LaTeX editor to handle. Consequently, it contains one method that performs exactly this task. 	<ul style="list-style-type: none"> This class uses instances of the following classes and thus has a dependency on them: controller.LatexEditorControllerHandler This class is also associated with the class that creates it and the interface whose methods it implements. Dependants: model.commands.CommandFactory Interfaces this class implements: model.commands.Command

Package: model.commands - Class Name: DisableVersionsManagementCommand	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This class is responsible for executing the command that disables the versioning strategy functionality of the LaTeX editor. Consequently, it contains one method that performs exactly this task. 	<ul style="list-style-type: none"> This class uses instances of the following classes and thus has a dependency on them: controller.LatexEditorControllerHandler This class is also associated with the class that creates it and the interface whose methods it implements. Dependants: model.commands.CommandFactory Interfaces this class implements: model.commands.Command

Package: model.commands - Class Name: EditCommand	
Responsibilities	Collaborations
<ul style="list-style-type: none"> • This class is responsible for executing the command that edits existing content on the document the LaTeX editor is currently handling. Consequently, it contains one method that performs exactly this task. • If the versioning strategy functionality of the LaTeX editor is enabled, then this class is also responsible for creating a new version of the document the LaTeX editor is currently handling and inserting it into the “stack” that holds the previous versions of the said document. 	<ul style="list-style-type: none"> • This class uses instances of the following classes and thus has a dependency on them: <ul style="list-style-type: none"> - LatexEditorControllerHandler - UserInterfaceInformationCollectorHandler • This class is also associated with the class that creates it and the interface whose methods it implements. • Dependants: <ul style="list-style-type: none"> model.commands.CommandFactory • Interfaces this class implements: <ul style="list-style-type: none"> model.commands.Command

Package: model.commands - Class Name: EnableVersionsManagementCommand	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This class is responsible for executing the command that enables the versioning strategy functionality of the LaTeX editor. Consequently, it contains one method that performs exactly this task. 	<ul style="list-style-type: none"> This class uses instances of the following classes and thus has a dependency on them: controller.LatexEditorControllerHandler This class is also associated with the class that creates it and the interface whose methods it implements. Dependants: model.commands.CommandFactory Interfaces this class implements: model.commands.Command

Package: model.commands - Class Name: LoadCommand	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This class is responsible for executing the command that loads a document residing in storage for the LaTeX editor to handle. Consequently, this class contains one method that performs exactly this task. 	<ul style="list-style-type: none"> This class uses instances of the following classes and thus has a dependency on them: controller.LatexEditorControllerHandler This class is also associated with the class that creates it and the interface whose methods it implements. Dependants: model.commands.CommandFactory Interfaces this class implements: model.commands.Command

Package: model.commands - Class Name: RollbackToPreviousVersionCommand	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This class is responsible for executing the command that rolls the document back to the previous version that is stored on memory. Consequently, this class contains one method that performs exactly this task. 	<ul style="list-style-type: none"> This class uses instances of the following classes and thus has a dependency on them: controller.LatexEditorControllerHandler This class is also associated with the class that creates it and the interface whose methods it implements. Dependants: model.commands.CommandFactory Interfaces this class implements: model.commands.Command

Package: model.commands - Class Name: SaveCommand	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This class is responsible for executing the command that saves the document the LaTeX editor is currently handling onto storage. Consequently, this class contains one method that performs exactly this task. 	<ul style="list-style-type: none"> This class uses instances of the following classes and thus has a dependency on them: controller.LatexEditorControllerHandler This class is also associated with the class that creates it and the interface whose methods it implements. Dependants: model.commands.CommandFactory Interfaces this class implements: model.commands.Command

Package: model.save.load.strategies - Class Name: LoadHtmlFile	
Responsibilities	Collaborations
<ul style="list-style-type: none"> • This class is responsible for loading an already existing html file (.html) from storage onto the application's memory so that the LaTeX editor can edit it. • Consequently, it contains methods that perform the conversion of a html file into a LaTeX document the editor can edit. 	<ul style="list-style-type: none"> • This class uses instances of the following classes and thus has a dependency on them: controller.LatexEditorControllerHandler controller.UserInterfaceInformationCollectorHandler • This class is also associated with the class that creates it. • Dependants: controller.LatexEditorController • Abstract class this class extends: model.save.load.strategies.LoadStrategy

Package: model.save.load.strategies - Class Name: LoadLatexFile	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This class is responsible for loading an already existing LaTeX file (.tex) from storage onto the application's memory so that the LaTeX editor can edit it. Consequently, it contains a method that performs exactly this task. 	<ul style="list-style-type: none"> This class uses instances of the following classes and thus has a dependency on them: controller.LatexEditorControllerHandler controller.UserInterfaceInformationCollectorHandler This class is also associated with the class that creates it. Dependants: controller.LatexEditorController Abstract class this class extends: model.save.load.strategies.LoadStrategy

Package: model.save.load.strategies - Class Name: abstract LoadStrategy	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This abstract class is responsible for offering a method that must be implemented from classes that extend it. Namely it offers the following method: loadFile() This class is also responsible for loading the correct template of a LaTeX document depending on the user's choice on the template selection window of the application. 	<ul style="list-style-type: none"> This abstract class is associated with every other class that extends it. Classes that extend this class: model.save.load.strategies. <ul style="list-style-type: none"> - LoadHtmlFile - LoadLatexFile Dependants: controller.LatexEditorController

Package: model.save.load.strategies - Class Name: SaveHtmlFile	
Responsibilities	Collaborations
<ul style="list-style-type: none"> • This class is responsible for saving the document the LaTeX editor currently handles onto storage in the form of a html file (.html) • Consequently, it contains methods that perform the conversion of a LaTeX document into a html file. 	<ul style="list-style-type: none"> • This class uses instances of the following classes and thus has a dependency on them: constants.StringConstants controller.LatexEditorControllerHandler controller.UserInterfaceInformationCollectorHandler • This class is also associated with the class that creates it. • Dependants: controller.LatexEditorController • Abstract class this class extends: model.save.load.strategies.SaveStrategy

Package: model.save.load.strategies - Class Name: SaveLatexFile	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This class is responsible for saving the document the LaTeX editor currently handles onto storage in the form of a LaTeX file (.tex) 	<ul style="list-style-type: none"> This class uses instances of the following classes and thus has a dependency on them: controller.LatexEditorControllerHandler controller.UserInterfaceInformationCollectorHandler This class is also associated with the class that creates it. Dependants: controller.LatexEditorController Abstract class this class extends: model.save.load.strategies.SaveStrategy

Package: model.save.load.strategies - Class Name: abstract SaveStrategy	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This abstract class is solely responsible for offering methods that must be implemented from classes that extend it. Namely it offers the following methods: saveFile() 	<ul style="list-style-type: none"> This abstract class is associated with every other class that extends it. Classes that extend this class: model.save.load.strategies. <ul style="list-style-type: none"> - SaveHtmlFile - SaveLatexFile Dependants: controller.LatexEditorController

Package: <code>model.version.strategies</code> - Class Name: <code>StableVersionsStrategy</code>	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This class is responsible for implementing the “stable” versioning strategy that is offered by the LaTeX editor. This means that its job is to keep the previous version of the document on storage for each document change. Consequently, this class contains methods that aid in performing exactly this task. 	<ul style="list-style-type: none"> This class uses instances of the following classes and thus has a dependency on them: <code>model.Document</code> This class is also associated with the class that creates it and the classes that call upon it. Dependants: <code>model.version.strategies.VersionsStrategyFactory</code> <code>model.VersionManager</code> Interfaces this class implements: <code>model.version.strategies.VersionsStrategy</code>

Package: model.version.strategies - Class Name: interface VersionsStrategy	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This interface is solely responsible for offering methods that must be implemented from classes that are associated with it. <p>Namely it offers the following methods:</p> <ul style="list-style-type: none"> putVersionID() removeVersion setEntireHistory getEntireHistory getDocumentOfLastVersion 	<ul style="list-style-type: none"> This interface is associated with every other class that implements it. Classes that implement this interface: <ul style="list-style-type: none"> model.version.strategies.StableVersionsStrategy model.version.strategies.VolatileVersionStrategy

Package: model.version.strategies - Class Name: VersionsStrategyFactory	
Responsibilities	Collaborations
<ul style="list-style-type: none"> • This class is a “factory” class. • This class has only one responsibility: create and return various new objects to the classes that call upon it. 	<ul style="list-style-type: none"> • This class creates and returns new instances of other classes Specifically it has the following dependencies: model.version.strategies. - StableVersionsStrategy - VolatileVersionsStrategy

Package: <code>model.version.strategies</code> - Class Name: <code>VolatileVersionsStrategy</code>	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This class is responsible for implementing the “volatile” versioning strategy that is offered by the LaTeX editor. This means that its job for each document change is to keep the previous version of the document in a memory list of subsequent document versions. Consequently, this class contains methods that aid in performing exactly this task. 	<ul style="list-style-type: none"> This class uses instances of the following classes and thus has a dependency on them: <code>model.Document</code> This class is also associated with the class that creates it and the classes that call upon it. Dependants: <code>model.version.strategies.VersionsStrategyFactory</code> <code>controller.LatexEditorController</code> <code>model.VersionManager</code> Interfaces this class implements: <code>model.version.strategies.VersionsStrategy</code>

Package: view - Class Name: ChooseTemplateWindow (former ChooseTemplate)	
Responsibilities	Collaborations
<ul style="list-style-type: none"> • This class is responsible for presenting the template selection window of the application to the user. • Consequently, this class contains methods that initialize and interact with the GUI components of said window. 	<ul style="list-style-type: none"> • This class uses instances of the following classes and thus has a dependency on them: <ul style="list-style-type: none"> - LatexEditorControllerHandler - LatexEditorController - UserInterfaceInformationCollectorHandler - UserInterfaceInformationCollectorController • Dependants: <ul style="list-style-type: none"> view.MainWindow view.OpeningWindow

Package: view - Class Name: MainWindow	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This class is responsible for presenting the main window of the application to the user. Consequently, this class contains methods that initialize and interact with the GUI components of said window. 	<ul style="list-style-type: none"> This class uses instances of the following classes and thus has a dependency on them: controller. <ul style="list-style-type: none"> - LatexEditorControllerHandler - LatexEditorController - UserInterfaceInformationCollectorHandler - UserInterfaceInformationCollectorController Dependants: <ul style="list-style-type: none"> view.ChooseTemplateWindow view.OpeningWindow

Package: view - Class Name: OpeningWindow	
Responsibilities	Collaborations
<ul style="list-style-type: none"> This class is responsible for presenting the opening/starting window of the application to the user. Consequently, this class contains methods that initialize and interact with the GUI components of said window. 	<ul style="list-style-type: none"> This class uses instances of the following classes and thus has a dependency on them: controller. <ul style="list-style-type: none"> - LatexEditorControllerHandler - UserInterfaceInformationCollectorHandler Dependants: <ul style="list-style-type: none"> view.ChooseTemplateWindow