AdvancedText2SpeechEditor

Sprint Report

Πηνελόπη Ελευθεριάδη 3221
Αθανασία-Δανάη Τσαούση 3349

VERSIONS HISTORY

Date	Version	Description	Author
11/03/21	1	Οργανωση και πλανο εκτελεσης εργασιας	Ελευθεριάδη Πηνελόπη Αθανασία-Δανάη Τσαούση
16/03/21	2	Υλοποίηση πακέτων και οργάνωση κλάσεων	
23/03/21	3	Υλοποίηση TTSFacade και την τεσταρισμα με την `Text2SpeechEditorAppView` ,μετατρεποντας ενα String σε audio.	
6/04/21	4	Υλοποίηση κλάσεων για μορφη αρχείου και decoding.Επίσης,υλοποίηση κλάσεων για πακέτο commands.	
13/04/21	5		
19/04/21	6	Υλοποίηση κλάσεων για το πακέτο output(ExcelWriter,WordWriter)	
28/04/21	7	Υλοποίηση κλάσεων για το πακέτο output(AtBashDecorator, Rot13Decorator).	
3/05/21	8	Έλεγχος κλάσεων των πακέτων input,output.	
11/05/21	9	Αλλαγές στις κλάσεις του πακέτου commands και της κλάσης Document.	

17/05/21	10	Έλεγχος input μεσω του menu στην Text2SpeechEditorAppView.	
18/05/21	11	Έλεγχος input μεσω του menu στην Text2SpeechEditorAppView.	
20/05/21	12	Προσθήκη νέου παραθύρου όταν επιλεχθεί το button edit ώστε να γίνεται επεξεργασία του κειμένου.	
21/05/21	13	Προσθήκη νέου button στο παράθυρο της edit για την αποήκευση των αλλαγών.	
23/05/21	14	Επίλυση προβλημάτων στα πακέτα input,output.	
24/05/21	15	Επίλυση προβλημάτων στα πακέτα input,output.	
25/05/21	16	Επίλυση προβλημάτων στα πακέτα input,output.	
26/05/21	17	Προσθήκη μεθόδων για αναπαραγωγή και επανάληψη των περιεχομένων του αρχείου με τις μεθόδους play() και replayAudio στην κλάση Text2SpeechEditorAppView.	

Introduction

Με αυτό το report δίνουμε πληροφορίες για την εφαρμογή Text2SpeechEditorAppView.
Η εφαρμογή αυτη δίαβαζει αρχεία και τα μετατρέπει σε audio.Για το δίαβασμα και την εξαγωγή αρχείων υπάρχουν επιλογές για την μορφή του : Excel και Word.Επιπλέον, υπάρχει δυνατότητα κωδικοποίησης και αποκωδικοποίησης με δύο στρατηγικές : Rot13 και AtBash.

Scrum team and Sprint Backlog

<For the user stories included in this release specify below corresponding tests using a typical tabular form.>

Scrum team

Product Owner	Apostolos Zarras
Scrum Master	Πηνελόπη Ελευθεριάδη Αθανασία-Δανάη Τσαούση
Development Team	Πηνελόπη Ελευθεριάδη Αθανασία-Δανάη Τσαούση

Sprints

Sprint No	Begin Date	End Date	Number of weeks	User stories
1	**23/04/21**		**1**	**TTSFacade**
2	**06/04/21**		**3**	**Input package**
3	**19/04/21**		**1**	**commands**
4	**19/04/21**		**4**	**Output package**
5	**18/05/21**	**28/05/21**	**5**	**Text2SpeechEditorAppView**

Use Cases

<Use Case 1>

Use case ID	OpenDocument
Actors	User
Pre conditions	There is a file to open
Main flow of events	1. The user clicks on the OPEN button 2. The use case opens a window 3. User inputs the path of the file 4. The use case opens a window with choises(Rot13,AtBash,NoEncoding) 5. User chooses the decoding strategy
Post conditions	The content of the file appears on under the menu .

<Use Case 2>

Use case ID	EditDocument
Actors	User
Pre conditions	
Main flow of events	1.The user clicks on the EDIT button 2.The use case opens a window 3.User edits the content using the window that appeared 4.The use case hase a button “SAVE “ 5.If the user wants to save the edited contents clicks this button
Post conditions	The save windows appear.

<Use Case 3>

Use case ID	SaveDocument
Actors	User
Main flow of events	1.The user clicks on the SAVE button 2.The use case opens a window 3.User chooses the path of the file 4.The use case opens a window with choises(Rot13,AtBash,NoEncoding) 5.User chooses the encoding strategy
Post conditions	

<Use Case 3>

Use case ID	ExitDocument
Actors	User
Main flow of events	1.The user clicks on the EXIT button 2.The app closes .

<Use Case 4>

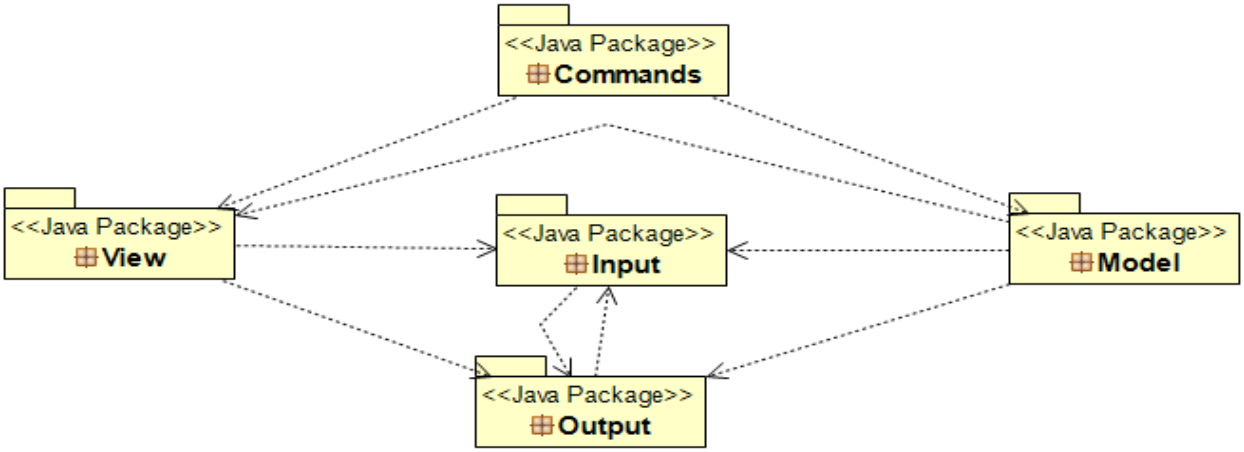
Use case ID	Audio
Actors	User
Pre conditions	A file has been opened.
Main flow of events	1.The user clicks on the AUDIO button 2.The use case plays the contents of the file.

<Use Case 5>

Use case ID	Replay
Actors	User
Pre conditions	A file has been opened.
Main flow of events	1.The user clicks on the REPLAY button 2.The use case replays the contents of the file.

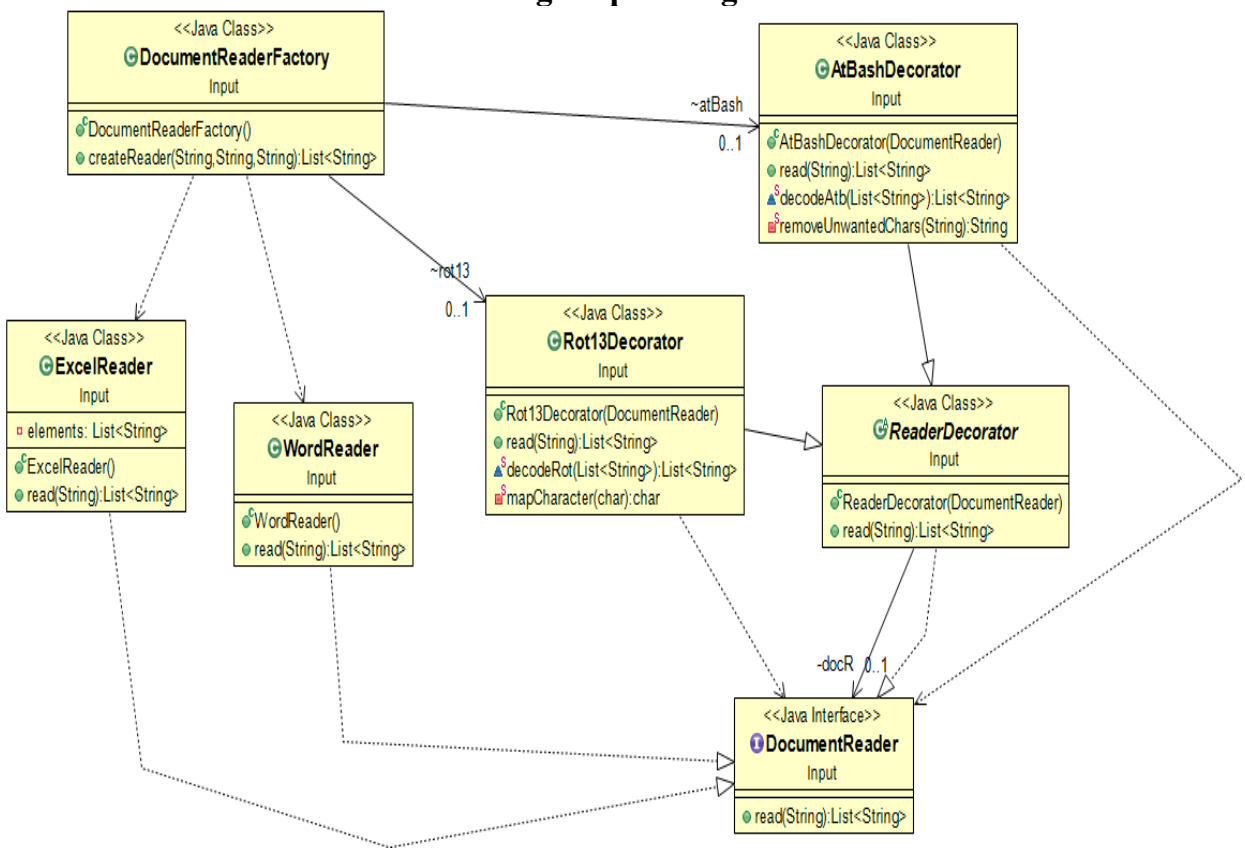
Design

Architecture



Design

Package Input Diagram



Class Name: DocumentReader	
Responsibilities: ▪ interface that is implemented by file opening strategies, which parse the file formats (.docx, .xlsx)	**Collaborations:**

Class Name: `ReaderDecorator`	
Responsibilities: ▪ allow to easily combine a decoding strategy with a particular file opening strategy ▪ `has different implementations, one for each different decoding strategy`	**Collaborations:** ▪ `DocumentReader`

Class Name: `AtBashDecorator`	
Responsibilities: ▪ Responsible for AtBash decoding.	**Collaborations:** ▪ `ReaderDecorator` ▪ DocumentReader

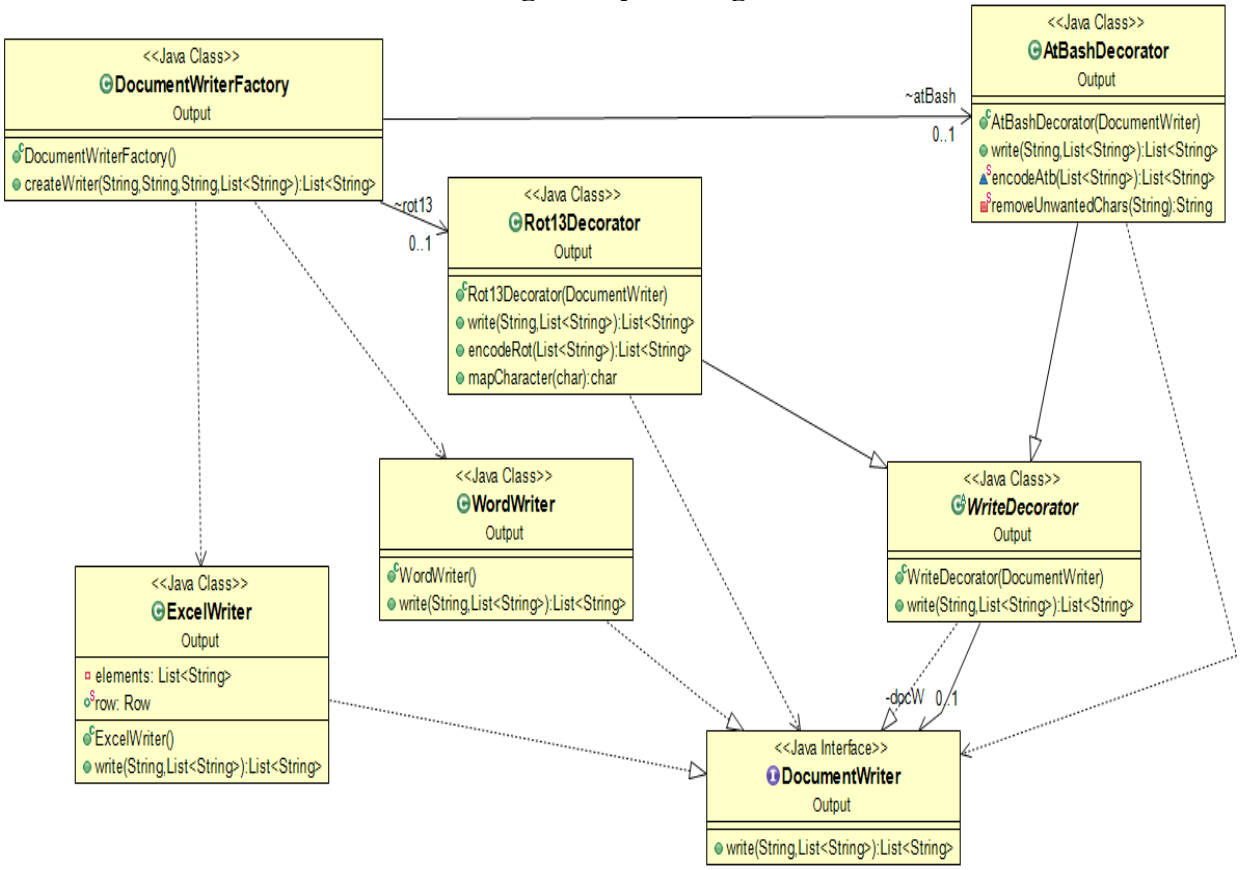
Class Name: `Rot13Decorator`	
Responsibilities: ▪ Responsible for Rot13 decoding.	**Collaborations:** ▪ `ReaderDecorator` ▪ DocumentReader

Class Name: `DocumentReaderFactory`	
Responsibilities: ▪ provides a parameterized factory method	**Collaborations:** ▪ AtBashDecorator ▪ Rot13Decorator ▪ WordReader ▪ ExcelReader

Class Name: `ExcelReader`	
Responsibilities: ▪ Reads .xlsx files	**Collaborations:** ▪ DocumentReader

Class Name: `WordReader`	
Responsibilities: ▪ Reads docx files	**Collaborations:** ▪ DocumentReader

Package Output Diagram



Class Name: DocumentWriter	
Responsibilities: ▪ interface that writes data to the file formats (.docx, .xlsx)	**Collaborations:**

Class Name: `WriterDecorator`	
Responsibilities: ▪ allow to easily combine a decoding strategy with a particular file writing strategy ▪ `has different implementations, one for each different decoding strategy`	**Collaborations:** ▪ `DocumentWriter`

Class Name: `AtBashDecorator`	
Responsibilities: ▪ Responsible for AtBash decoding.	**Collaborations:** ▪ `WriterDecorator` ▪ `DocumentWriter`

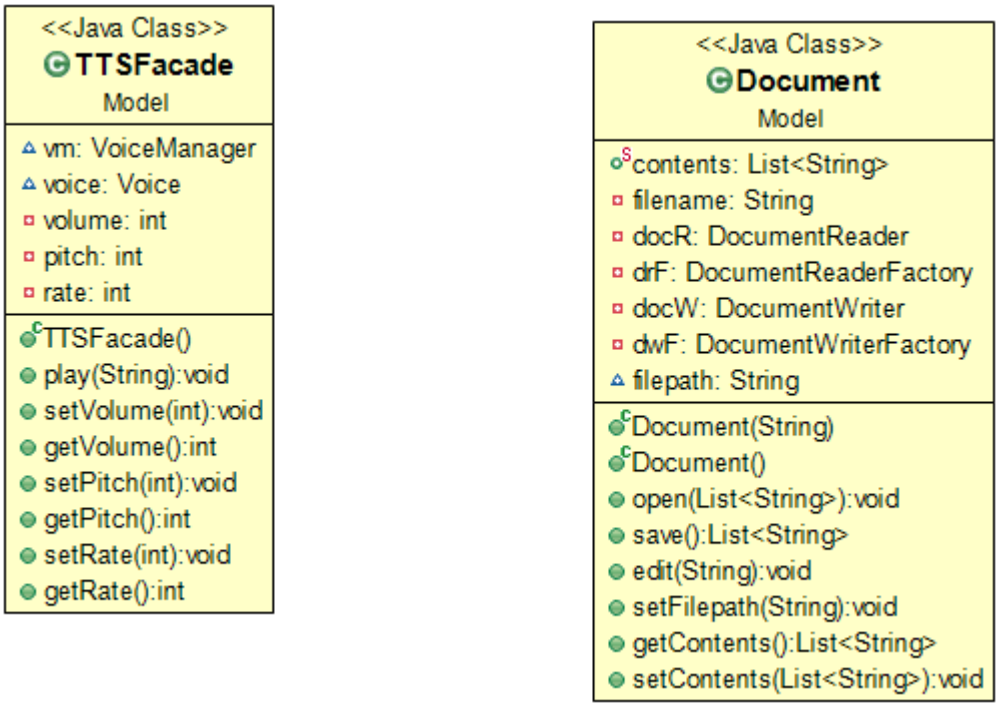
Class Name: `Rot13Decorator`	
Responsibilities: ▪ Responsible for Rot13 decoding.	**Collaborations:** ▪ `WriterDecorator` ▪ `DocumentWriter`

Class Name: `DocumentWriterFactory`	
Responsibilities: ▪ provides a parameterized factory method	**Collaborations:** ▪ AtBashDecorator ▪ Rot13Decorator ▪ `WordWriter` ▪ `ExcelWriter`

Class Name: `ExcelWriter`	
Responsibilities: ▪ Writes .xlsx files	**Collaborations:** ▪ `DocumentWriter`

Class Name: `WordWriter`	
Responsibilities: ▪ Writes .docx files	**Collaborations:** ▪ `DocumentWriter`

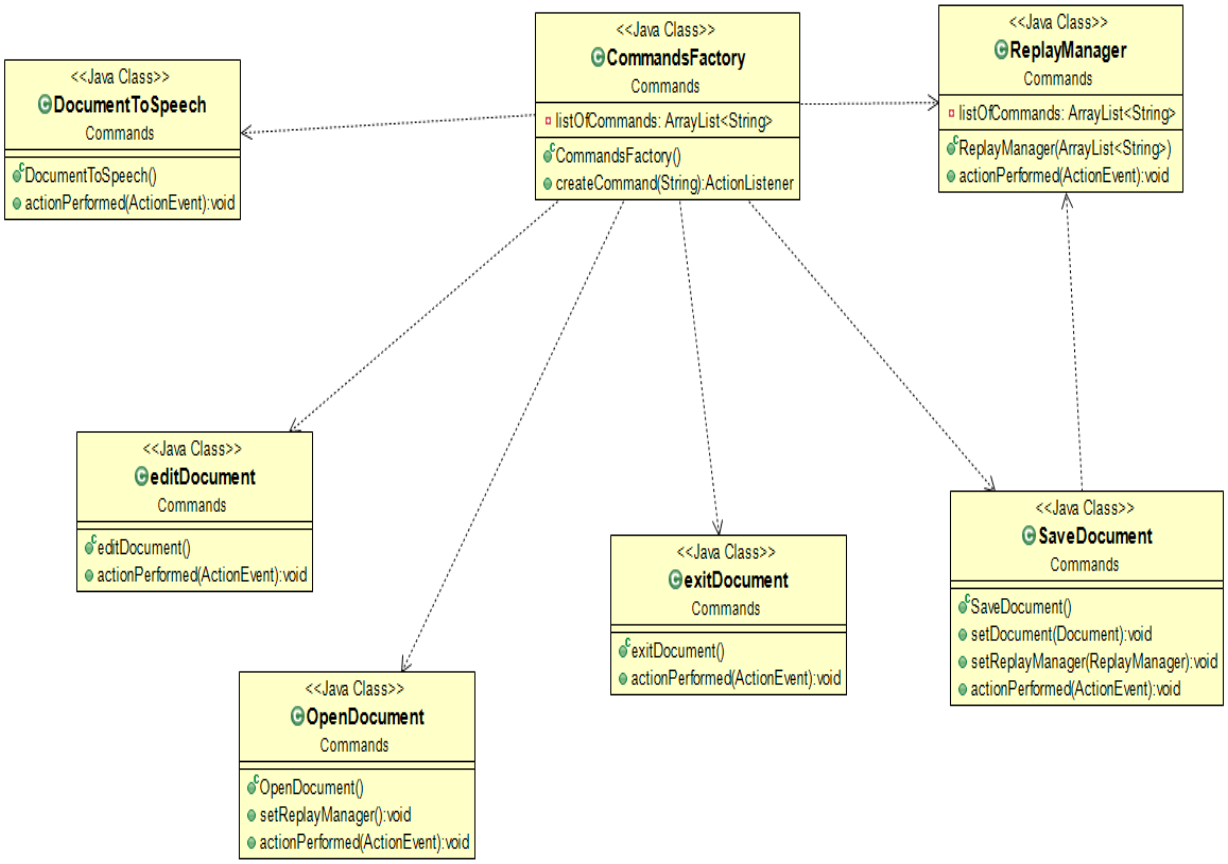
Package Model Diagram



Class Name: Document	
Responsibilities: ▪ Represents a document with contents.	**Collaborations:**

Class Name: TTSFacade	
Responsibilities: ▪ Responsible for converting text to audio.	**Collaborations:**

Package Commands Diagram



Class Name: `CommandsFactory`	
Responsibilities: ▪ provides a parameterized factory method when a button is pressed.	**Collaborations:** ▪ `editDocument` ▪ `DocumentToSpeech` ▪ `OpenDocument` ▪ `exitDocument` ▪ **ReplayManager** ▪ **SaveDocument**

Class Name: `editDocument`	
Responsibilities: ▪ calls Text2SpeechEditorAppView to edit the contents of the file	**Collaborations:**

Class Name: `DocumentToSpeech`	
Responsibilities: ▪ calls Text2SpeechEditorAppView to convert the contents of the file to audio.	**Collaborations:**

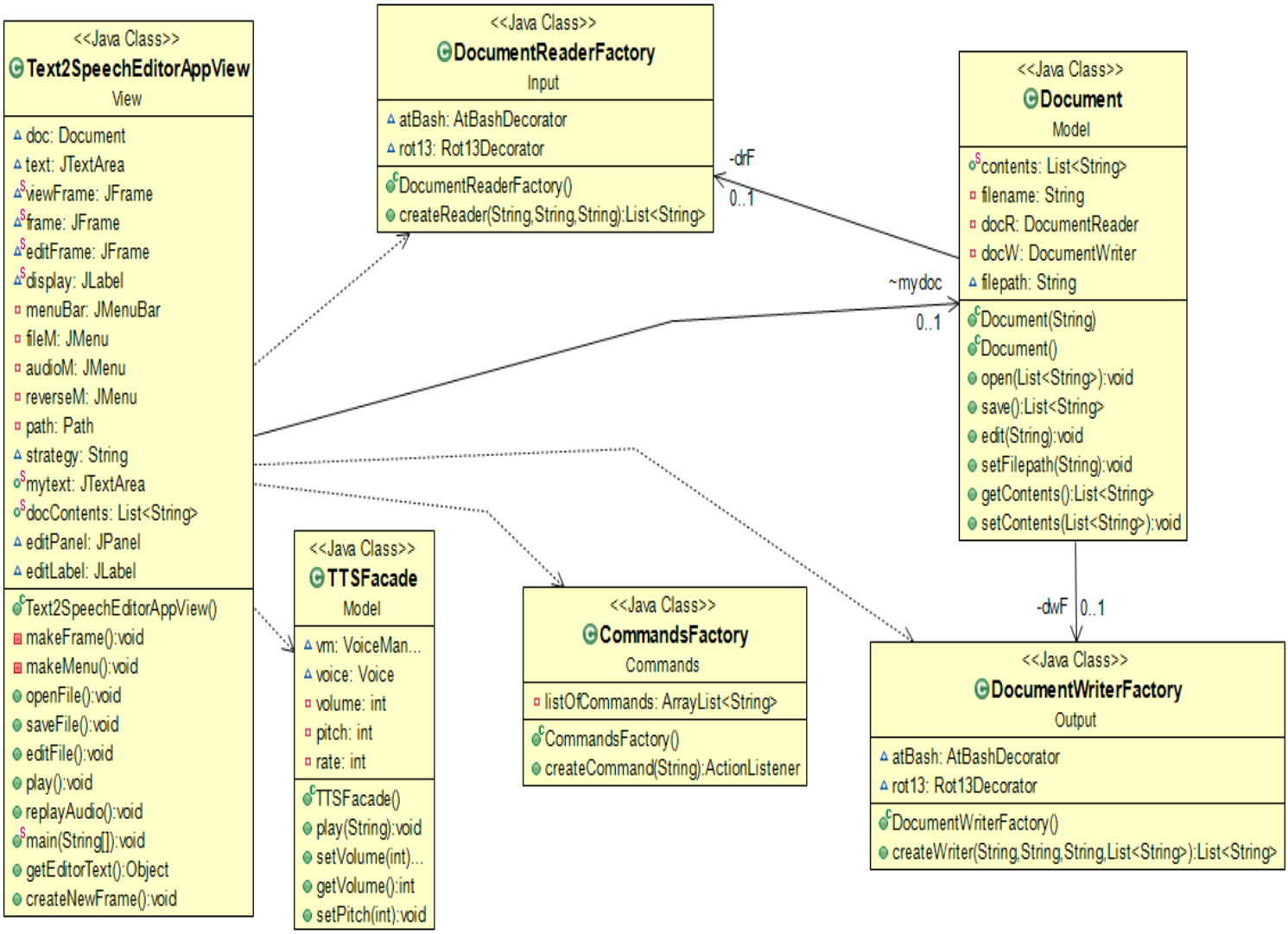
Class Name: `OpenDocument`	
Responsibilities: ▪ calls Text2SpeechEditorAppView to open the file.	**Collaborations:**

Class Name: `exitDocument`	
Responsibilities: ▪ calls Text2SpeechEditorAppView to exit the app.	**Collaborations:**

Class Name: ReplayManager	
Responsibilities: ▪ calls Text2SpeechEditorAppView to replay the contents of the file.	**Collaborations:**

Class Name: SaveDocument	
Responsibilities: ▪ calls Text2SpeechEditorAppView to save the contents of the file.	**Collaborations:**

Package View Diagram



Class Name: `Text2SpeechEditorAppView`	
Responsibilities: ▪ Has the User Interface(menu) ▪ Connects buttons with the correct ActionListeners	**Collaborations:** ▪ **DocumentReaderFactory** ▪ **DocumentWriterFactory** ▪ **Document** ▪ **CommandsFactory** ▪ **TTSFacade**