AdvancedText2SpeechEditor

Sprint Report

"Bug Catchers want to battle"
Papanikolaou Nikolaos 4145
Priftis Filippo 4162

VERSIONS HISTORY

Date	Version	Description	Author
26/5/2021	1.0v	Added all the information in the report.	Filippo Priftis

1 Introduction

1.1 Document Structure

The rest of this document is structured as follows. Section 2 describes out Scrum team and specifies the this Sprint's backlog. Section 3 specifies the main design concepts for this release of the project.

2 Scrum team and Sprint Backlog

User Story	Class	Comment
US1	OpenDocumentTest	Each method tests opening a different type of file or encryption.
US2	SaveDocumentTest	Each method tests saving a different type of file or encryption.
US3	SaveDocumentTest	We edit the contents of the file before saving them.
US4	TextToSpeechTest	Tested in the documentToSpeechTest() method.
US5	TextToSpeechTest	No different than US4 since we always give a string to Transform.
US6	TextToSpeechTest	Tested in the audioParametersTest() method.
US7	RecordingTest	Tested via the testStartRecording() method.
US8	TextToSpeechTest	Tested via replayRecodedAction().
US9	RecordingTest	Tested via the testEndRecording() method.

2.1 Scrum team

Product Owner	Apostolos Zarras
Scrum Master	Papanikolaou Nikolaos
Development Team	Papanikolaou Nikolaos, Filippo Priftis

2.2 Sprints

Sprint No	Begin Date	End Date	Number of weeks	User stories
1	**1/4/2021**	**15/4/2021**	**2**	**US1, US2, US3**
2	**16/4/2021**	**23/4/2021**	**1**	**US4, US5, US7**
3	**7/5/2021**	**21/5/2021**	**2**	**US6, US8, US9**

3 Use Cases

3.1 Open File

Use case ID	Open File
Actors	User
Pre conditions	The App is up and running.
Main flow of events	1. The UC starts when the user chooses the “Open File” option from the “File” menu. 2. The user chooses the encryption type of the file from the menu. 3. The File browser window opens. 4. While the desired file is not displayed in the files list of the file browser 4.1. The user selects a directory from the list. 4.2. The application displays the contents of the directory in the files list. 5. The user selects the desired file. 6. If the user has selected to open an encrypted file. 6.1 The file is decrypted according to the specified encryption type. 7. The file open dialog disappears. 8. The file is displayed.
Post conditions	The contents of the file are available to the user for further processing.

3.2 Save File

Use case ID	Save File
Actors	User
Pre conditions	There is an open file in the app
Main flow of events	1. The UC starts when the user chooses the “Save File” option from the “File” menu. 2. The user chooses the encryption type of the file from the menu. 3. The File browser window opens. 4. While the desired destination is not displayed in the files list of the file browser 4.1. The user selects a directory from the list. 4.2. The application displays the contents of the directory in the files list. 5. The user types the desired name. 6. If the user has selected to save an encrypted file. 6.1 The file is encrypted according to the specified encryption type. 7. The file save dialog disappears.
Alternative flow 1	1. The alternative flow starts after step 5. 2. The application indicates that a file with the desired name already exists in the directory. 3. A window pops up asking the user to choose whether he wants to override the file or to change the desired name. 4. The user chooses one of the options. 5. The window closes. 6. If the user chose to change the file name. 6.1 The use case continues from step 5 of the main flow. 7. The use case continues from step 6 of the main flow.

3.3 Transform File to Audio

Use case ID	Transform File to Audio
Actors	User
Pre conditions	There is an open file in the app.
Main flow of events	1. The UC starts when the user chooses the “Transform” option from the “Edit” menu. 2. A pop-up window appears asking the user if he wants to transform the whole file, only a part of it that he has selected (enabled only if there is text selected) or to cancel the operation. 3. If the user chooses to transform the whole file. 3.1 All the text in the file is transformed into audio. 4. If the user chose “Transform Selection. 4.1. The selected text is transformed into audio. 5. If the user chose “Cancel”. 5.1 The operation is canceled. 6. The pop-up window is closed. 7. A toolbar that handles the audio playback becomes enabled in the main window.
Post conditions	An audio transformation track is created.

3.4 Start Recording Transformation Operations

Use case ID	Start Recording Operations
Actors	User
Pre conditions	The App is up and running.
Main flow of events	1. The UC starts when the user chooses the “Start recording” option from the “File” menu. 2. The option “End Recording” is added to the “File” menu. 3. Every time the user executes an operation, it is saved to be played again.

3.5 End Recording Transformation Operations

Use case ID	End Recording Operations
Actors	User
Pre conditions	The App is up and running. The user has executed UC4.
Main flow of events	1. The UC starts when the user chooses the “End recording” option from the “File” menu. 2. The recorded Transformation Operations are deleted. 3. The audio controller toolbar becomes disabled in the main window. 4. The option “Start Recording” is added to the “File” menu.

3.6 Repeat Recorded Transformation Operation

Use case ID	Repeat Recorded Operation
Actors	User
Pre conditions	Requirements of Recorded Transformation. The user has executed UC3.
Main flow of events	1. The UC starts when the user chooses the “Replay Transformation” in the “File” menu. 2. A pop-up window appears with a list of all the recorded transformations. 3. The user selects an action. 4. The user clicks “OK”. 5. The selected Transformation is executed as in UC1, UC2 or UC3.

3.7 Play Transformed Audio

Use case ID	Play Transformed Audio
Actors	User
Pre conditions	There is an open file in the app. The audio is paused. There is a loaded audio track from UC3 or UC6.
Main flow of events	1. The UC starts when the user clicks the “Play” button in the audio player toolbar. 2. The “Play” button is then replaced with the “Pause” button. 3. The paused audio track resumes playing.

3.8 Pause Audio Playback

Use case ID	Pause Audio Playback
Actors	User
Pre conditions	There is an open file in the app. The user has executed UC7.
Main flow of events	1. The UC starts when the user clicks the “Pause” button in the audio player toolbar. 2. The “Pause” button is then replaced with the “Play” button. 3. The audio track pauses playing.

3.9 Change Audio Volume

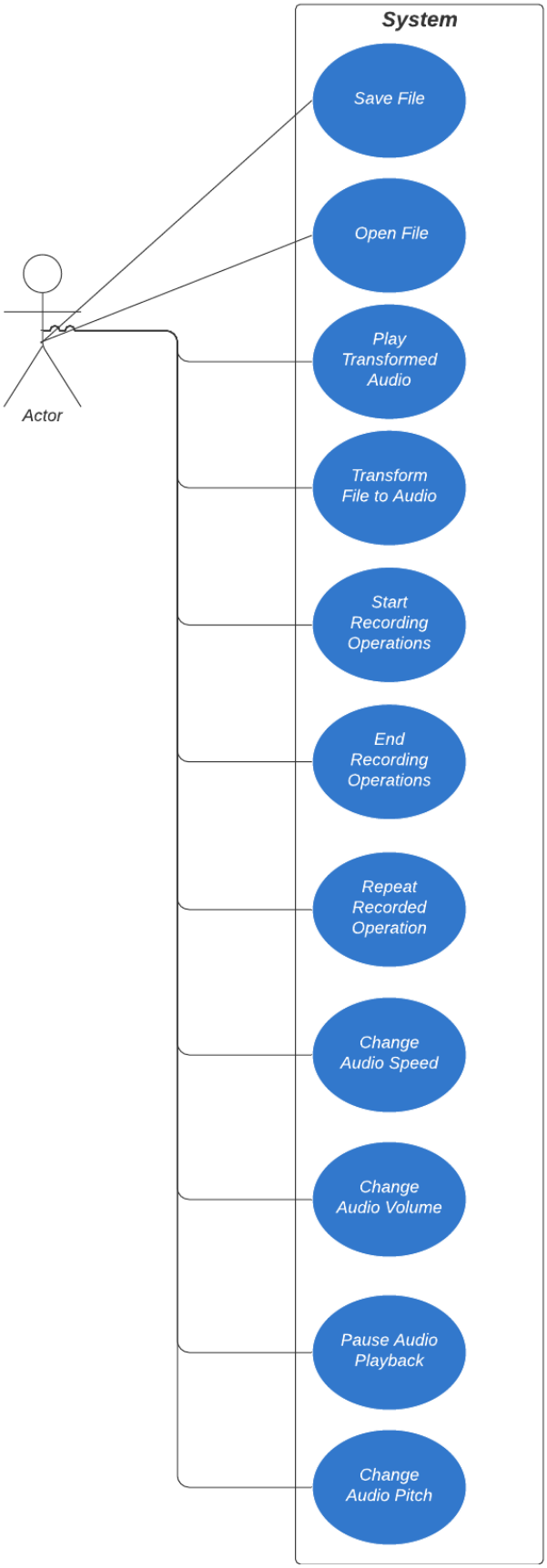
Use case ID	Change Audio Volume
Actors	User
Pre conditions	There is an open file in the app. There is a loaded audio track from UC3 or UC6.
Main flow of events	1. The UC starts when the user chooses the Volume Level of the audio from the slider in the audio player toolbar. 2. The volume changes according to the value of the slider.
Alternative Flow 1	1. The user clicks the “Mute” button from the audio player toolbar. 2. The sound volume changes to zero.

3.10 Change audio Pitch

Use case ID	Change audio Pitch
Actors	User
Pre conditions	There is an open file in the app. There is a loaded audio track from UC3 or UC6.
Main flow of events	1. The UC starts when the user chooses a new value for the audio pitch slider in the audio player toolbar. 2. The audio pitch changes according to the value of the slider.

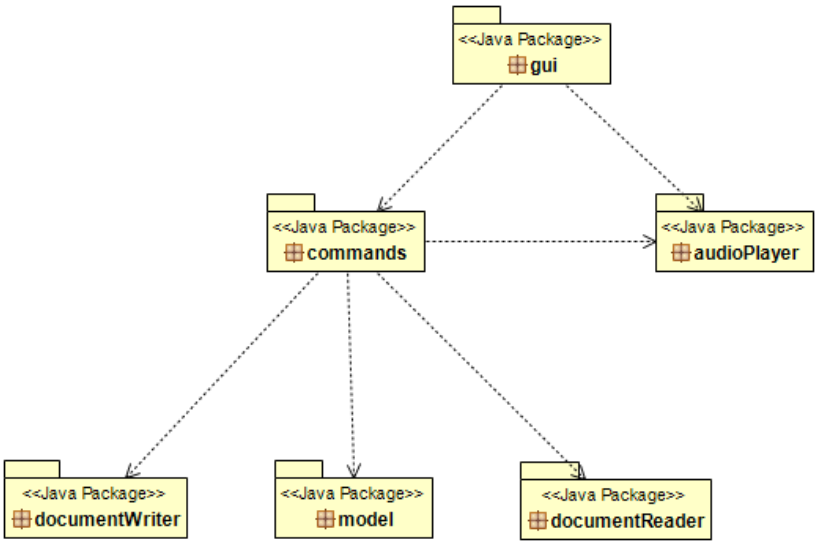
3.11 Change audio Speed

Use case ID	Change audio Speed
Actors	User
Pre conditions	There is an open file in the app. There is a loaded audio track from UC3 or UC6.
Main flow of events	1. The UC starts when the user clicks the plus or the minus button in the audio player toolbar. 2. The audio speed changes by 0.25(increasing or decreasing) according to the value of the slider.


System
Save File
Open File
Play Transformed Audio
Transform File to Audio
Start Recording Operations
End Recording Operations
Repeat Recorded Operation
Change Audio Speed
Change Audio Volume
Pause Audio Playback
Change Audio Pitch
Actor

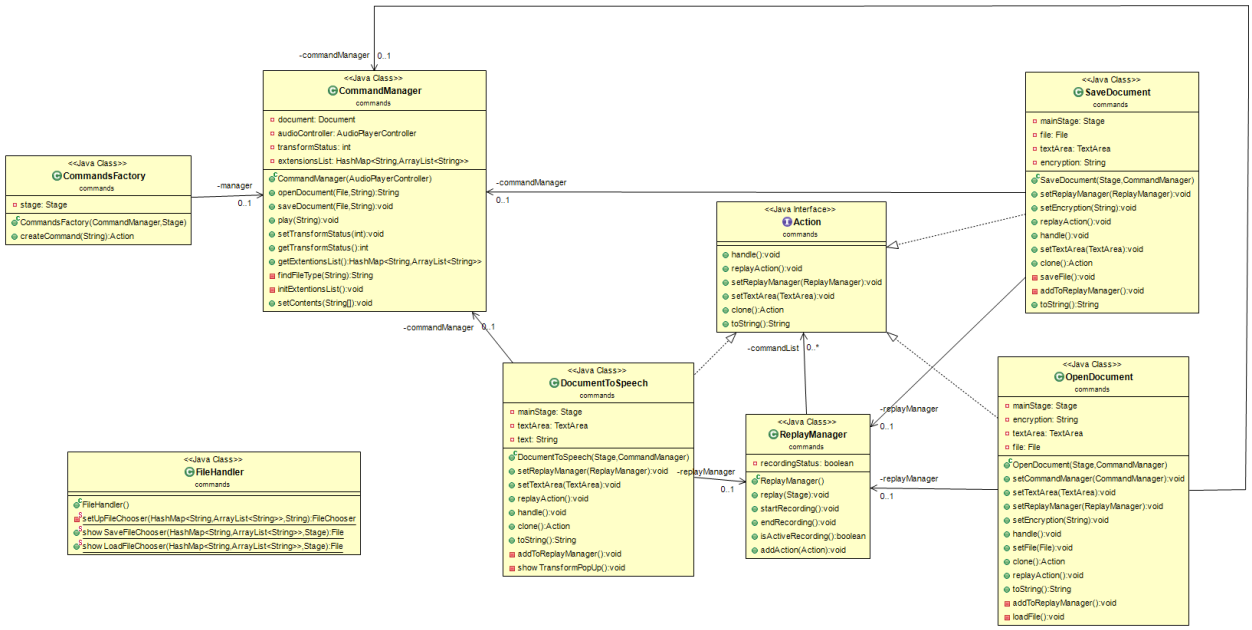
4 Design

4.1 Architecture

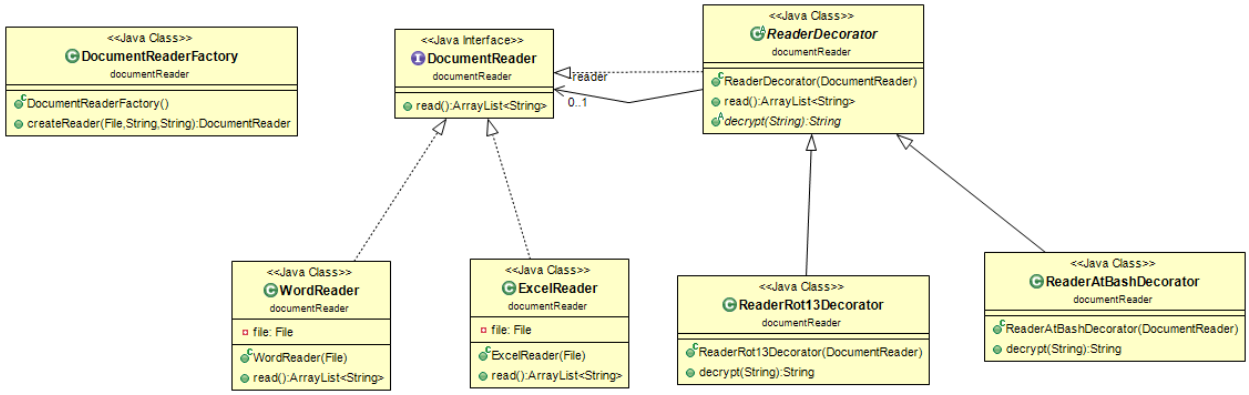


4.2 Design

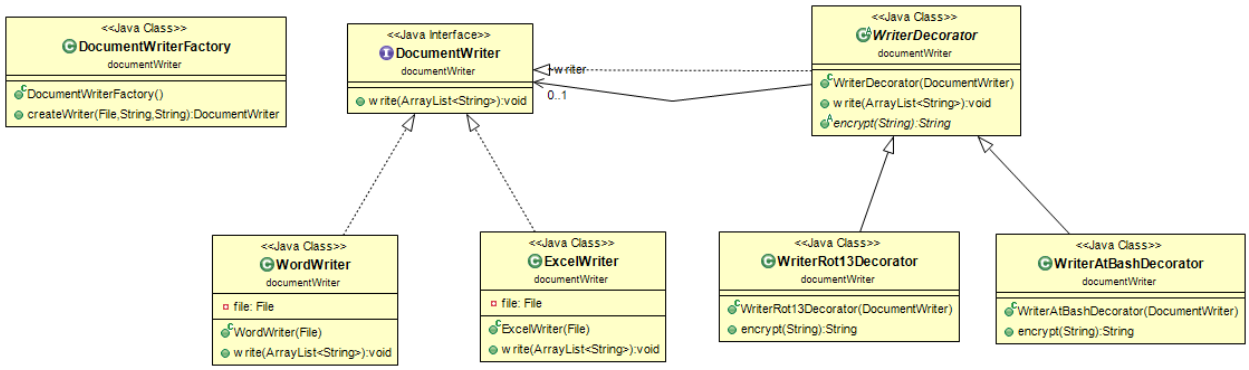
Commands Package



Document Reader Package



Document Writer Package



Class Name: Command Manager	
Responsibilities: ▪ Saves Document to File ▪ Opens Document from File ▪ Transforms Audio to Speech ▪ Calls Audio Manager to play audio ▪ Keeps compatible File Types	**Collaborations:** ▪ Document ▪ AudioPlayer ▪ DocumentToSpeech ▪ OpenDocument ▪ SaveDocument

Class Name: Command Factory	
Responsibilities: ▪ Creates new Command Objects	**Collaborations:** ▪ **----**

Class Name: Save Document	
Responsibilities: ▪ Saves Document to File ▪ Knows textArea ▪ Adds itself to ReplayManager	**Collaborations:** ▪ CommandManager ▪ Document ▪ ReplayManager

Class Name: Open Document	
Responsibilities: ▪ Opens Document from File ▪ Knows textArea ▪ Adds itself to ReplayManager	**Collaborations:** ▪ CommandManager ▪ Document ▪ ReplayManager

Class Name: Document to Speech	
Responsibilities: ▪ Transforms Audio to Speech ▪ Knows textArea ▪ Adds itself to ReplayManager	**Collaborations:** ▪ CommandManager ▪ Document ▪ ReplayManager

Class Name: Replay Manager	
Responsibilities: ▪ Keeps List of Actions if recording is enabled. ▪ Replays an action.	**Collaborations:** ▪ ----

Class Name: File Handler	
Responsibilities: ▪ Opens File Chooser to Save or Open a File	**Collaborations:** ▪ ----

Class Name: Document Reader Factory

Responsibilities:	Collaborations:
▪ Creates new DocumentReader Objects	▪ ----

Class Name: Word Reader	
Responsibilities:	**Collaborations:**
▪ Reads contents of a word File	▪ ----

Class Name: Excel Reader	
Responsibilities:	**Collaborations:**
▪ Reads contents of an excel File	▪ ----

Class Name: Reader Decorator(Rot13 Decorator, AtBash Decorator)	
Responsibilities:	**Collaborations:**
▪ Decrypts contents of a File	▪ DocumentReader(WordReader, ExcelReader)

Class Name: Document Writer Factory	
Responsibilities:	**Responsibilities:**
▪ Creates new DocumentWriter Objects	▪ ----

Class Name: Word Writer	
Responsibilities:	**Collaborations:**
▪ Writes contents to a word File	▪ ----

Class Name: Excel Writer	
Responsibilities:	**Collaborations:**
▪ Writes contents to an excel File	▪ ----

Class Name: Reader Decorator(Rot13 Decorator, AtBash Decorator)	
Responsibilities: ▪ Encrypts contents of a File	**Collaborations:** ▪ DocumentWriter(WordWriter, ExcelWriter)

Class Name: Audio Player Controller	
Responsibilities: ▪ Plays audio. ▪ Pauses audio. ▪ Changes audio volume. ▪ Changes audio pitch. ▪ Changes audio speed.	**Collaborations:** ▪ TTSFacade

Class Name: TTS Facade	
Responsibilities: ▪ Plays audio. ▪ Pauses audio. ▪ Changes audio volume. ▪ Changes audio pitch. ▪ Changes audio speed.	**Collaborations:** ▪ AudioPlayerController