Software Testing Document

for

iText Based PDF Viewer

Version 1.0

Prepared by

Dhruvkumar Patel
Jace Robinson
Pranav Pranav

CS 7140 Advanced Software Engineering
July 20th 2016

Table of Contents

1 Introduction

1.1 Purpose of The Test Plan Document

The Test Plan documents and tracks the necessary information required to effectively define the approach to be used in the testing of the project's product. The Test Plan document is created during the Planning Phase of the project. Its intended audience is the project manager, project team, and testing team. Some portions of this document may on occasion be shared with the client/user and other stakeholder whose input/approval into the testing process is needed.

2 Conformance Testing

2.1 Test Risks / Issues

Untimely program crashing can be considered as most prominent risk otherwise no major risk.

2.2 Items to be Tested / Not Tested

Item to Test	Test Description	Test Date	Responsibility
Functional requirements	All the desired features like, inserting and deleting annotations.	07-20-16	Team
UI/Display	Out of Scope, NOT to be tested	NA	NA

2.3 Test Approach(s)

Unit Tests are required to be written at modules level.

At the time of Integration all unit tests are required to be pass.
Critical requirements should be flag for Acceptance testing.

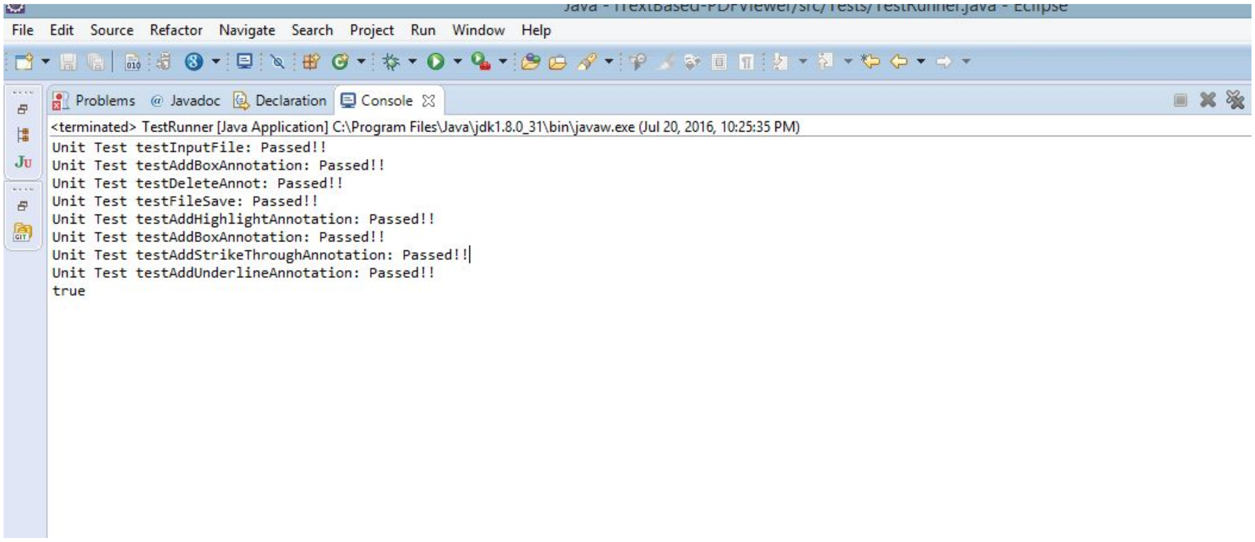
2.4 Test Pass / Fail Criteria

Whenever module fulfills the goal of producing the desired output and cross verified by assertions, it is considered as pass otherwise fail.

2.5 Test Entry / Exit Criteria

To satisfy the pre condition, all entry level variables should be provided to the unit test to check for the validity of the logic written in the module. Whenever it qualify the pass/fail criteria, it is considered good for exit.

2.6 Test Deliverables



```
<terminated> TestRunner [Java Application] C:\Program Files\Java\jdk1.8.0_31\bin\javaw.exe (Jul 20, 2016, 10:25:35 PM)
Unit Test testInputFile: Passed!!
Unit Test testAddBoxAnnotation: Passed!!
Unit Test testDeleteAnnot: Passed!!
Unit Test testFileSave: Passed!!
Unit Test testAddHighlightAnnotation: Passed!!
Unit Test testAddBoxAnnotation: Passed!!
Unit Test testAddStrikeThroughAnnotation: Passed!!
Unit Test testAddUnderlineAnnotation: Passed!!
true
```

2.7 Test Environmental / Staffing / Training Needs

Standard PC with any OS is required to create testing environment. Team is enough as staff. No training required

3 Functional Testing

3.1 Test Risks / Issues

Untimely program crashing can be considered as most prominent risk otherwise no major risk.

3.2 Items to be Tested / Not Tested

Item to Test	Test Description	Test Date	Responsibility
Annotation operations	Insertion and deletion functionality	07-20-16	Team
File operations	Input and processed output functionality	07-20-16	Team

3.3 Test Approach(s)

Same as above.

3.4 Test Pass / Fail Criteria

All the test of same category is required to pass in one batch. Otherwise same as above.

3.5 Test Entry / Exit Criteria

Same as above.

3.6 Test Deliverables

Same as above.

4 Performance Testing

4.1 Test Risks / Issues

Start with no lag and untimely crashing.

4.2 Items to be Tested / Not Tested

Item to Test	Test Description	Test Date	Responsibility
Program start	Product should start on click of exe.	07-20-16	Team
Program stop	Product should give proper output.	07-20-16	Team

4.3 Test Approach(s)

Run project with help of readme.txt

4.4 Test Pass / Fail Criteria

Manual verification.

4.5 Test Entry / Exit Criteria

Same as above.

5 Regression Testing

5.1 Test Risks / Issues

All modules should be interacting to each other.

5.2 Items to be Tested / Not Tested

Item to Test	Test Description	Test Date	Responsibility
All unit test	All unit test should pass	07-20-16	team

5.3 Test Approach(s)

Same as above.

5.4 Test Pass / Fail Criteria

Same as above.

5.5 Test Entry / Exit Criteria

Same as above.

5.6 Test Deliverables

Same as above.

6 Unit Testing

6.1 Test Risks / Issues

Each module should be working and throughput should be as per desired.

6.2 Items to be Tested / Not Tested

Item to Test	Test Description	Test Date	Responsibility
testInputFile	Unit test for checking the inputs	07-20-16	team
testFileSave	Unit test for checking the output	07-20-16	team
testAddUnderlineAnnotation	Unit test for checking UnderLine annotation are inserted or not	07-20-16	team

testAddHighlightAnnotation	Unit test for checking Highlight annotation are inserted or not	07-20-16	team
testAddStrikeThroughAnnotation	Unit test for checking Strikethrough annotation are inserted or not	07-20-16	team
testAddBoxAnnotation	Unit test for checking Box annotation are inserted or not	07-20-16	team
testDeleteAnnot	Unit test for checking any annotation are deleted or not	07-20-16	team

6.3 Test Approach(s)

Same as above.

6.4 Test Pass / Fail Criteria

Same as above.

6.5 Test Deliverables

Same as above.

7 User Acceptance Testing

7.1 Test Risks / Issues

Any kind of bugs at this stage should be prior identified.

7.2 Items to be Tested / Not Tested

Item to Test	Test Description	Test Date	Responsibility
Annotation operations	High priority tests	07-20-16	team

7.3 Test Approach(s)

All high priority tests are collected together and performed in front of customer including if he/she had demanded any. All should be pass.

7.4 Test Pass / Fail Criteria

Customer opinion matters. Apart from customer demanded tests, rest all will be same as above.

7.5 Test Entry / Exit Criteria

Same as above.

7.6 Test Deliverables

Along with customer demanded tests, rest all will be same as above.

7.7 Test Environmental / Staffing / Training Needs

Same as above.

8 Journals

Jace Robinson

Start Day 7-19-16

Today I will be continuing the implementation for the pdf viewer. Dhru has a good start available on github, https://github.com/Dhruvkumarpatel/iTextBased-PDFViewer, we will be working from that. The design document was "weak" in my opinion so some extra effort will be necessary for a good implementation.

As of today we are able to open and save pdfs, but are having troubles determining how to "display" the pdf to screen. For now I will develop functionality for adding and removing annotations and simply view the pdf outside of the program.

Start Day 7-20-16

Today we will continue working on PDF Viewer. My first goal of the day is functionality to add and remove annotations from within the hardcoded solution. I am ignoring implementing the viewer for now.

Given an (x,y,width,height, pageNumber), we are able to add a highlight annotation around that text. I will now look to do the same of underline, strikethrough, and boxing. I successfully completed this.

Dhru was able to get the viewer working! Yay! Now that are able to successfully view the pdf, we must new add annotations using GUIs. If we can identify "text positions" or positions based on user clicks, we should be able to add annotations.

Sadly the PDF-Renderer viewer we are using is UNABLE to show ANNOTATIONS. It does successfully display text. We attempted to implement alternatives but are having troubles. To proceed forward, we are completing the remaining functionality of adding and removing annotations. This will still be possible but the annotations will not display in our viewer. If you open the document in a different viewer such as xodo, the annotation will display.

We now have adding and removing annotations functionality complete. The accuracy of mouse clicks is off by a bit, but is at least on same page as user interacts. The underline and strikethrough are not created properly, they are added to the pdf file but do not actually display underline/strikethrough. This seems to be a problem with the iText implementation as the highlight and box work as expected.

Lastly is the wrap up of implementation and testing document. Dhruv and Pranav were able to create the unit testing, while I completed the doxygen and executable jar file.

All done!

Dhruv

Start Day 7-14-16

create a javafx project converted into maven and implement github repository and start implementation part with jace and pranav as a collaborator in Github repository. Try to learned iText 7 tutorials and try few things to get better idea. Thinking about pdf rendering and how to display pdf?

Start Day 7-16-16

Try to find pdfrenderer API to display pdf pages. Design basic functionality open existing pdf file and display inside javafx window using pdf renderer Api. Design next and previous button functionality to display pdf by pages and navigate in pdf through pages.try some annotations example using itext7 tutorials.

Start Day 7-17-16

write a review for requirement, specs and design documents.Try PDFBOx, JPedal and other things to display annotated pdf. but, we dont have that functionality for display. because we are depend on PdfRenderer features. start to developing annotation with jace and pranav.

start Day 7-19-16

developed highlights, underline, strikethrough,box annotations. Also developed remove annotation functionality

integrate all the code together. This project give me a good learning skills to

how to manage code while many people are working on same project inside same repository?

start Day 7-20-16

Atlast we developed SAVE functionality where user can able to save file wherever user wants inside local computer.we wrote the Testcases using JUnit and run successfully with all the testcases passed.

implementation document designed by Doxygen which is good tool for documentation.Add source code, readme file and make .exe file so one can open directly to use this tool. wrote Testing Document with all the testcases and screenshots.

Pranav

Start Day 7-16-16

Meet with dhruv discussed existing issues with rendering of pdf.

A couple of POC implemented.

Start Day 7-20-16

Group meeting: Task divided for the work on remaining of the implementations.
Worked a couple of PoC for Display and rendering.
To meet deadline, rescope goals.
Created unit testing and test report document.