# GUI Development Stage.

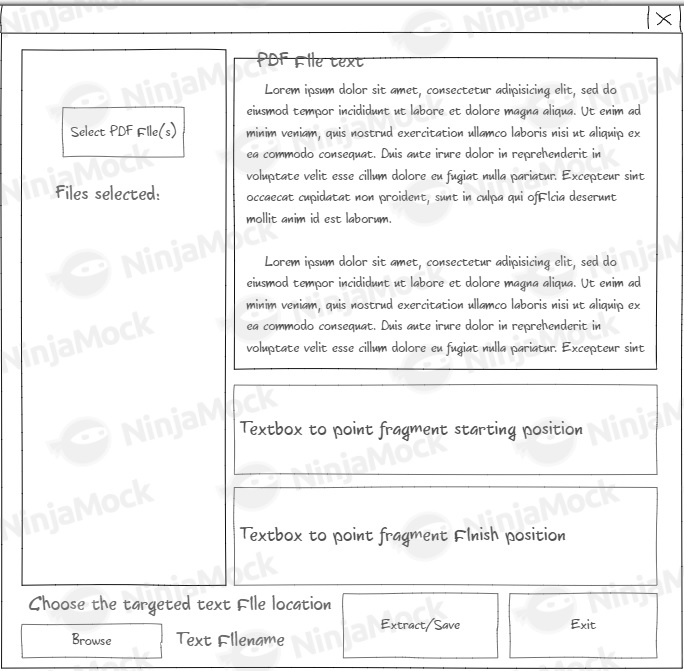
At this stage, I started developing application GUI interface and made some improving changes which I’m going to describe below.

Figure 1. Initial interface design.

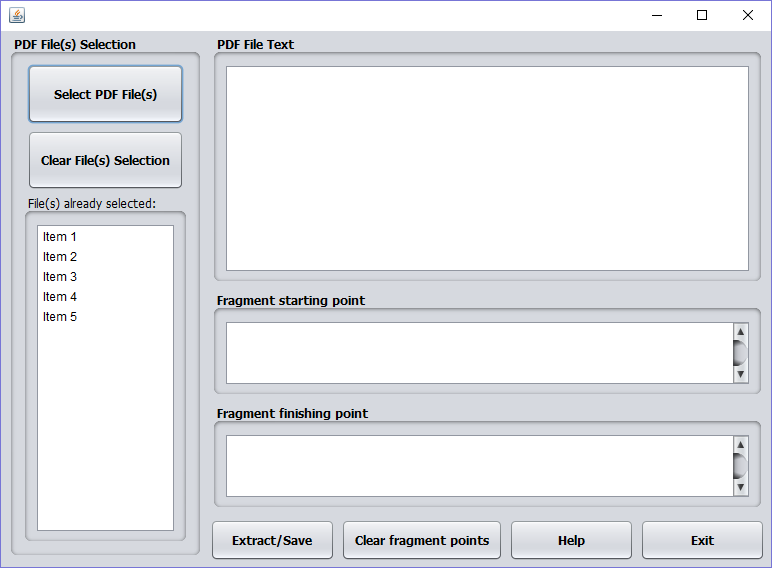


Figure 2. Interface design made in NetBeans.

Comparing to initial design “Browse” button was moved on a separate panel “PDF File(s) selection” and renamed into “Select PDF File(s)” this button brings a window (Figure 3) to select PDF file(s) for text extraction. “PDF File(s) selection” panel also has “Clear File(s) selection” button which clears out selection list and a list of selected PDF files.

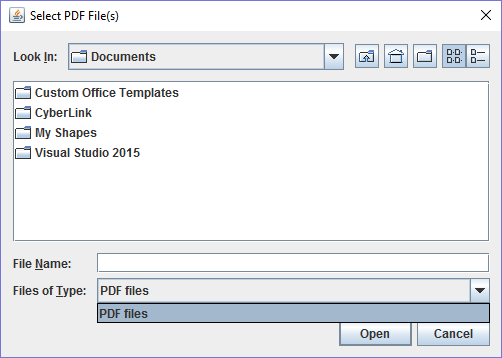
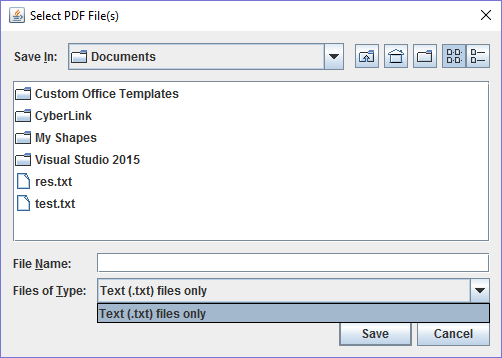


Figure 3. Window to select PDF file(s).

PDF File Text panel has a text area which displays text extracted from PDF file selected in the “File(s) already selected:” list of the PDF files.

“Fragment starting point” text area to type a starting point of the fragments to extract from all PDF files selection if it’s left empty start of the file will be used as a starting fragment point. “Fragment finishing point” text area to type a finishing point of the fragments to extract from all PDF files selection if it’s left empty end of the file will be used as a finishing fragment point.

“Extract/Save” button brings a window (Figure 4) to select or type a name of the result text file and initiates the text extraction process and saves it into the selected text file.

Figure 4. Window to select text file to save extracted text fragment(s).

“Clear Fragment Points” button clears text fields “Fragment starting point” and “Fragment finishing point”.

Button “Help” brings a window with the application purpose help, GUI elements description and they functionality explanation.

Button “Exit” quits the application.

## 2.Files selection stage.

For the first developing stage, I’ll be working with “PDF File(s) selection” section. It should give a possibility to users select several PDF format files and display the filenames in the list below. Button “Clear file(s)selection” should clear all selected file from the list.

To get this done I am using a standard Java Library “File Chooser”. Copyrights are placed in a code file.

To get the selected files and to give a possibility to reuse these files from any other part of the program I made a class “PDF files”.

## 3.Developing converter.

Converter is going to be made with use of Apache Tikka v1.14 an open-source Java library downloaded from <https://tika.apache.org/download.html> .

Converter will be set in a separate class to use it from the button “Extract/Save” and from the list of selected pdf files.

## 4. “PDF File text” functionality development.

This area should be developed the way it will be displaying pdf file selected on the list of selected pdf files converted into the text.

## 5. “Save/Extract” button event code development.

Main part of the program that cuts the fragments from pdf files and saves them into the text file chosen by user or creates a new text file with name defined by user.

At this stage, I had difficulties with pdf file selection from the list of the files because code was returning to selected file value twice. Having checked the internet, I found that this was reported as a Java bug and explanation is given below:

” The first event is

delivered on mouse-press event and its valueAdjusting property is set to true. This denotes that the selection may not

be finished (when you move mouse pointer over another item in the list while holding button pressed then another event

of this type is delivered). When you release mouse button then a new event is generated (with valueAdjusting property

set to false) denoting that the selection was finalized.

If you don't want to react on temporary selections then your event handler should check whether the delivered event has

valueAdjusting property set to true and don't react otherwise.” (Jan Stola, 2008).

As a result, to solve this issue the piece of code was added:

“boolean adjust = evt.getValueIsAdjusting();

if ((!adjust) && (!lst\_PDFfiles.isSelectionEmpty())) {“

## 6. Finalizing and adding the rest of the GUI elements functionality.

On this stage, I’m developing:

1. Exit button functionality.
2. Help button functionality.
3. Clear fragment points button functionality.
4. Debugging the application as a finalized product.
5. Making a runnable jar file.

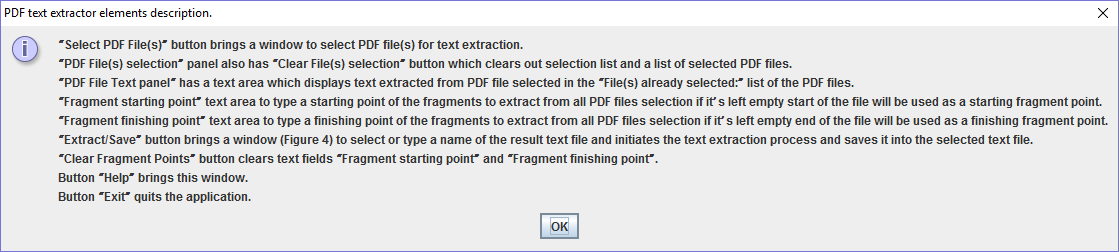


Figure 5. Help window dialog message.

When I was debugging the app, I realised that it proceeds the extraction code even when no files are on the list. The condition was added and appropriate message (Figure 6) made to direct user to “PDF file(s) selection” button.

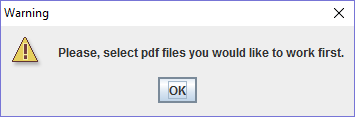


Figure 6. Warning message.

# Resources and materials.

The application developed on Java Platform (JDK) 8u121 in NetBeans IDE 8.2. Documentation for the project is made with the help of Microsoft Office 365, the diagrams are made in Microsoft Visio, for PDF file decoding an open source java library Apache Tikka v1.14 downloaded from <https://tika.apache.org/download.html>. JFileChooser Java library used, Copyrights are pointed in the code file PDFtextExtractorMainJFrame.java.

## Reference:

Jan Stola, (22 December 2008). NetBeans Bug 155789 – ListSelectionListener valueChanged method executed twice. Available at: <https://netbeans.org/bugzilla/show_bug.cgi?id=155789> (Accessed: 21 April 2017)