Criterion B – Design

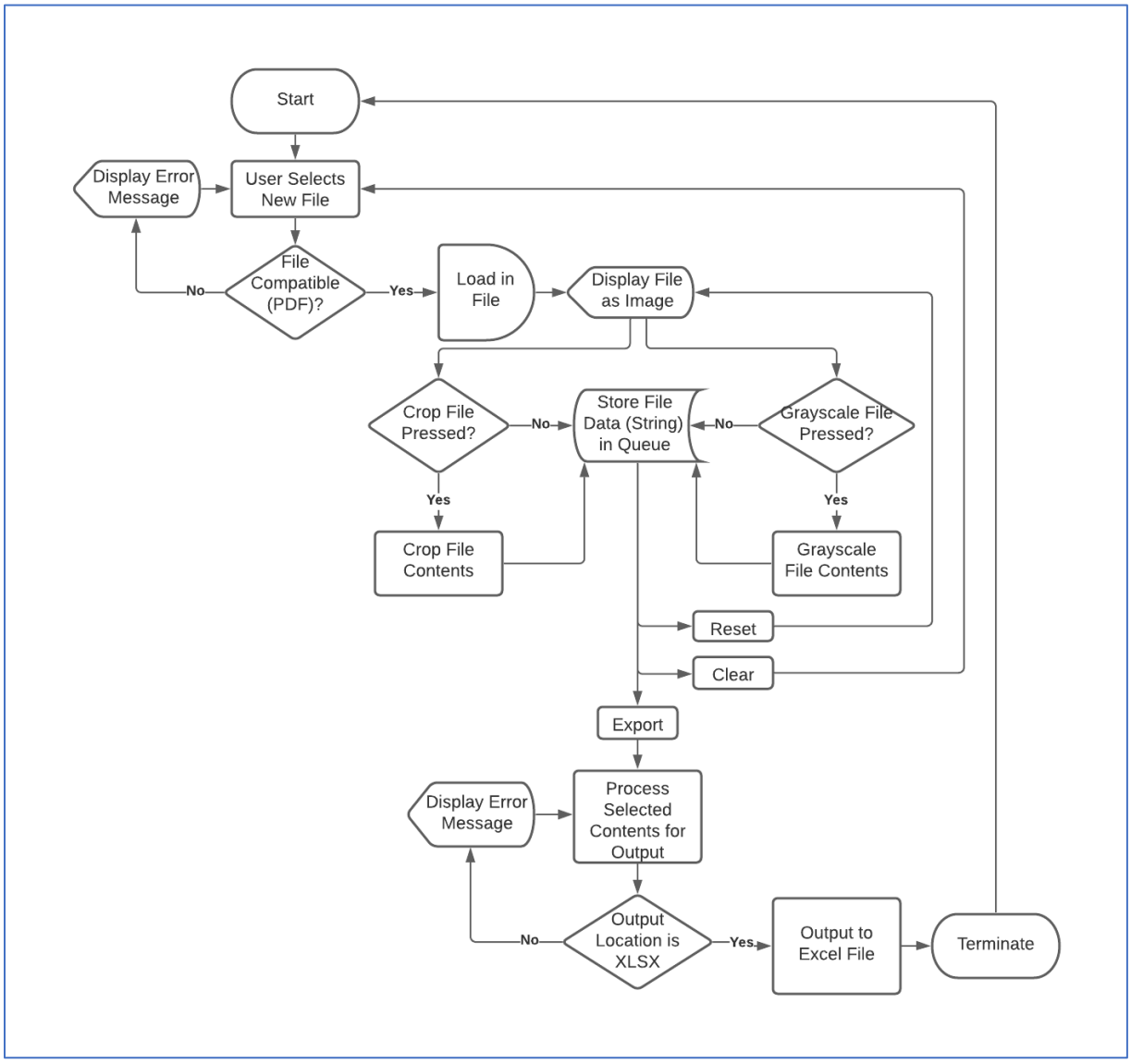
Test plan:

Test Type	Nature of Test	Example(s)
Functional GUI buttons, menus, and instructions	All processing functions are made clear to see and easy to use	"*Crop Button*" "*Grayscale Button*" "*Reset File Button*"
Functional file reading/writing	See if input to software and output work successfully	User selects PDF file: PDF file is displayed
Instructions manual is displayed every time the user opens the program; help menu can be accessed at any time under a *User Manual*	Open program and check if the instructions manual is present	User opens program: "*Welcome to the OCR Reader…*"+ instructions
Allow user to select PDF file	Use some of the sample files sent by the client and check if PDFs are compatible	"*Select PDF File"* option
Select file → allow to crop file area to the section they need	Crop the files sent by the client and observe the tools functionality	Presses crop button → opens new tab for cropping
Select file → allow for grayscale	When grayscale is pressed, image will become black and white	Presses grayscale button → image becomes grayscaled
Read contents, process them, and convert to text	When outputted, the tag should be properly read and in text form	User selects a tag → tag outputted upon export
Output to Excel file (as per modified user request in Appendix B)[1], with date, file contents, and name of file for organization	Check if tags output to an Excel file of the users choice	"*Export File*"
User causes an exception: program catches it and returns resolvent methods	Purposefully make mistakes and see if they ae handled	User does not select a file: "*Error, please select file*"
Provide brief page in instructions manual with plans for future development	User will see this page while cycling through instructions	"*Future dev ideas:*"

[1] See Appendix B (Mr.S, 2020, L. 11)

Diagram Representations

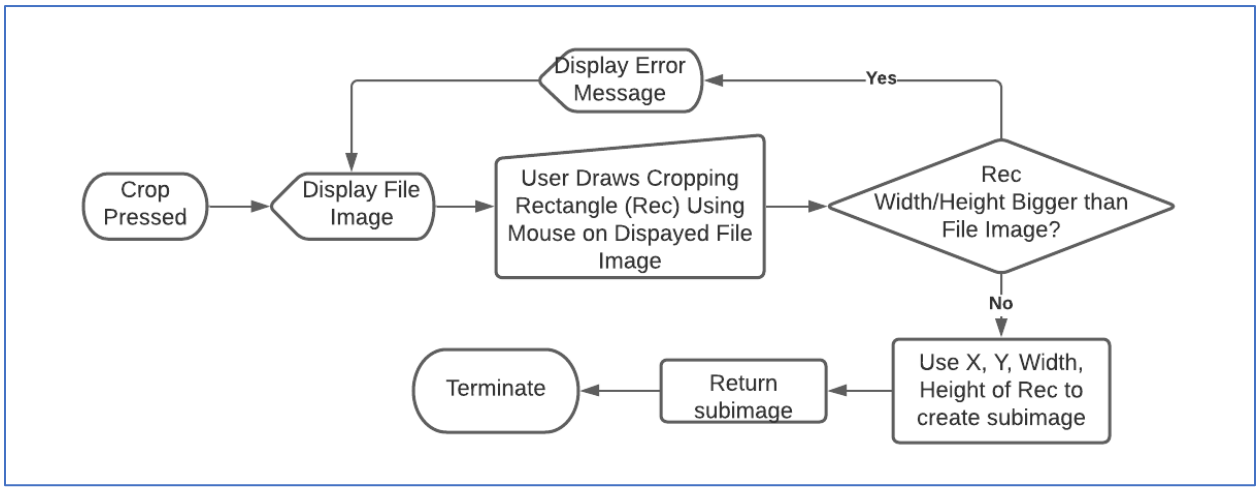
Main program algorithm (flowchart):



When the program is run, there will be an initial test (using an **if statement**) to see if the selected file contains ".pdf". If compatible, the file will be displayed, and its contents can be cropped and modified. The file name and data will then be **queued** and stored there until a valid export location is chosen. See appendix B for consultation.

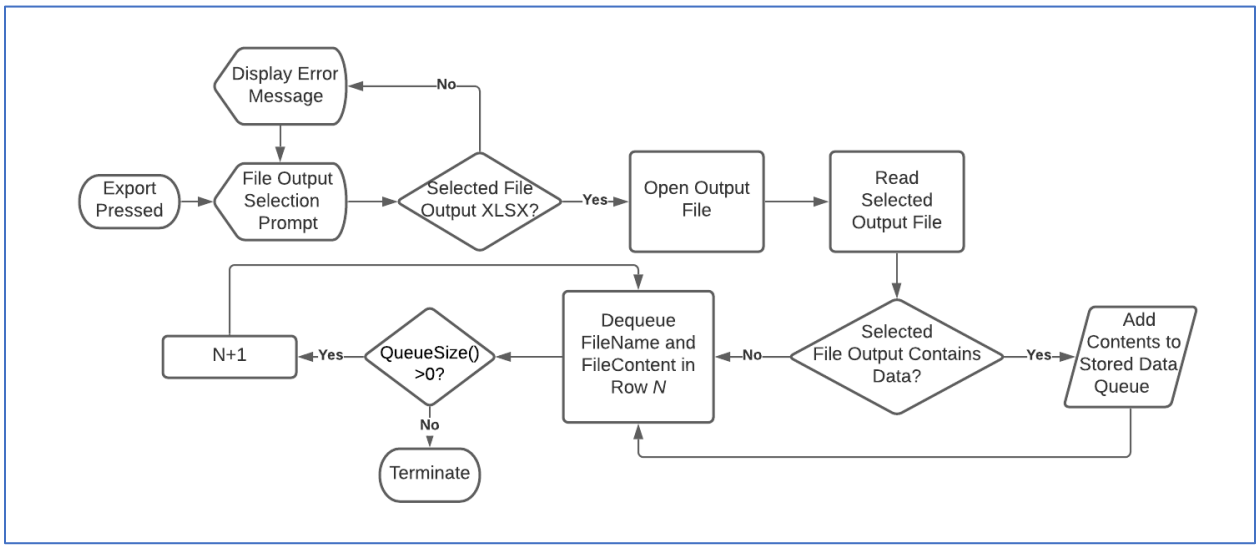
Candidate #: hpg293

Crop algorithm:



The selected file will be displayed and the user will be able to draw a rectangle around the tag (similar to the snipping tool). The rectangles parameters will be **updated** as the user moves.

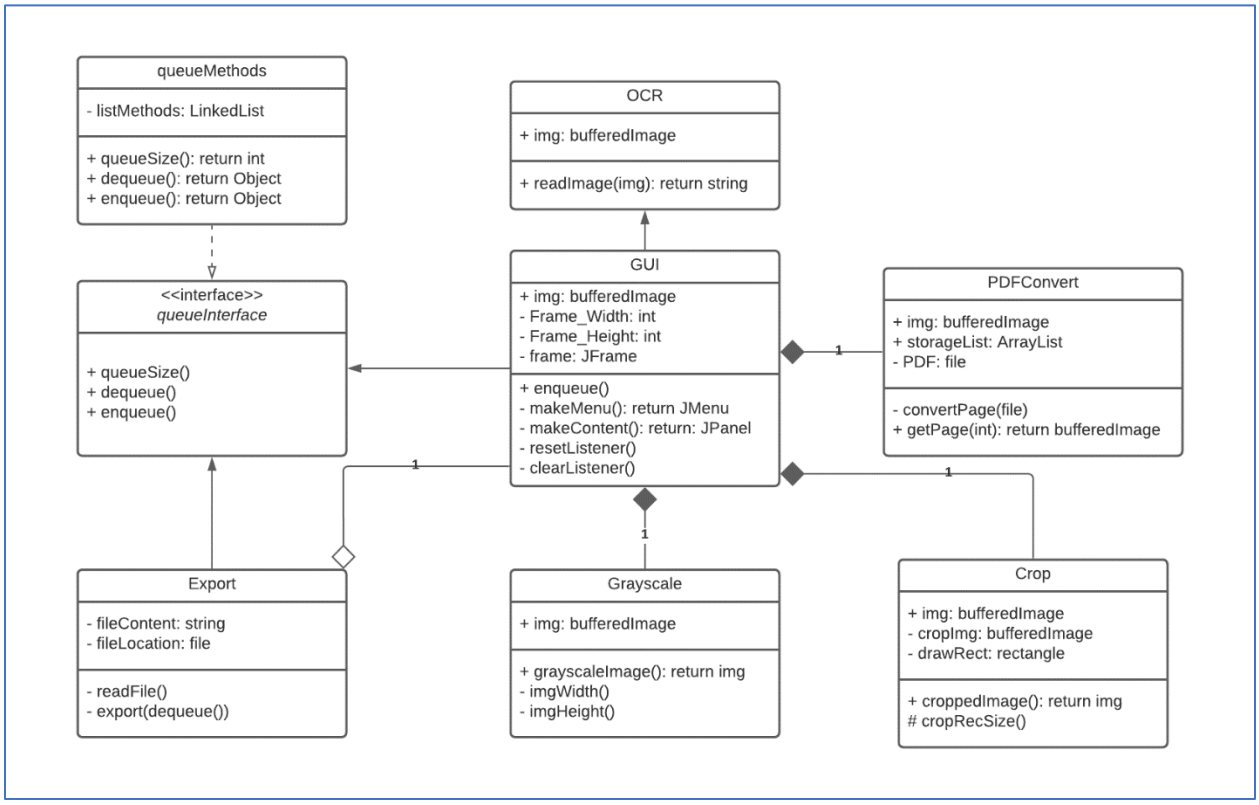
Output algorithm:



The output file is checked for validity then **read** to see if the location contains tags. If it does, these are added back to the queue and then outputted to the Excel document in appended order.

Candidate #: hpg293

Basic UML with interdependencies:



Advanced data dictionary:

Class Location	Data Item(s)	Data Type	Data Description	Validation/Other Notes
GUILayout1; cropClassMain; grayscaleFunc; PDFconverter;	img	bufferedImage	Converted PDF page in jpg format	public
GUILayout1	Frame_Width	int	Sets the default width of the main application	final; static; private
GUILayout1	Frame_Height	int	Sets the default height of the main application	final; static; private
GUILayout1	frame, smallFrame	JFrame	Used as frames to display the contents of the program	private
GUILayout1	imgLabel	JLabel	Used to display the bufferedImage in the program	public

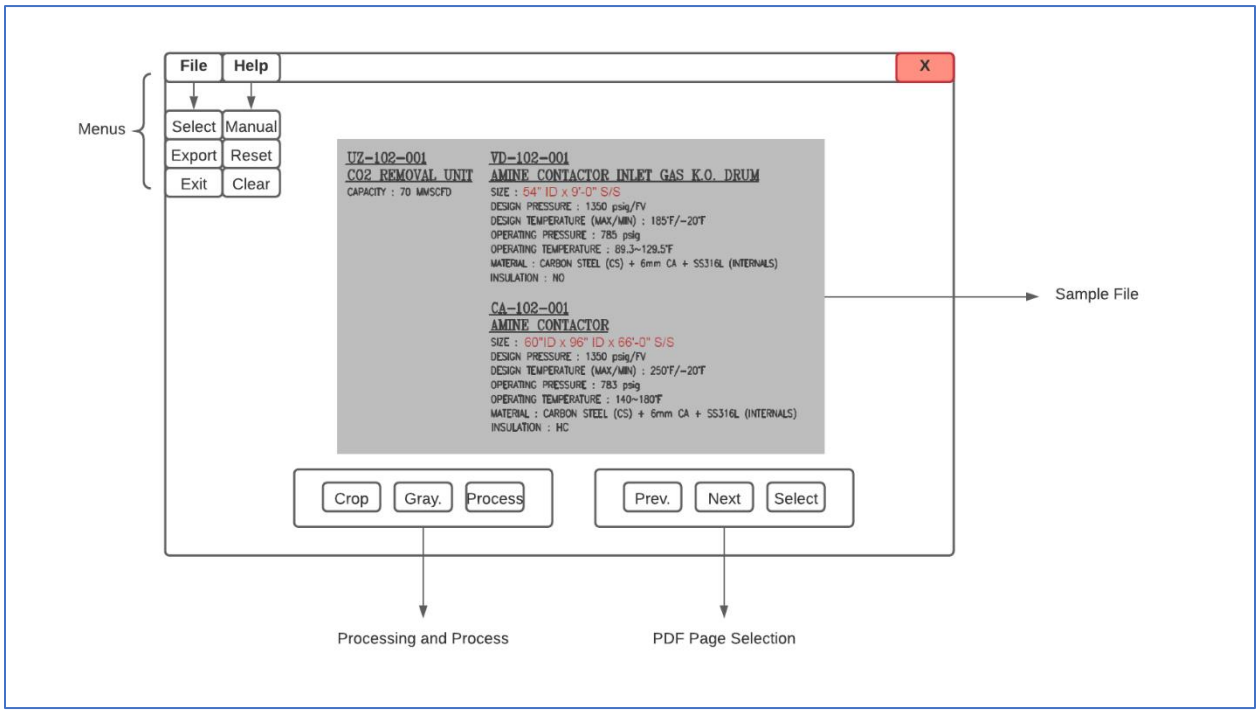
GUILayout1	crop, crop2, escape, grayscale, process, next, previous, select	JButton	Buttons for each function of the program	private
GUILayout1	isClicked	boolean	Determines if certain buttons have been clicked	private
GUILayout1	draftDraw	file	Used to find location of the file	private
GUILayout1	fileLocation, fileName	String	Used for identifying the file location and output the file name	public
GUILayout1	convert	PDFConverter()	Allows for the use of the PDFConverter class methods	public; object
GUILayout1	excel	exportTags()	Allows for the use of the exportTags class methods	public; object
GUILayout1	nextOrPrev	int	Used for returning certain pages by number	public;
GUILayout1; exportTags	queue	LinkedListQueue()	Allows for the use of a fully functional queue that stores processed tags	public; object; parameter; abstract data structure
exportTags	tagFile	String	File name	parameter; equal to fileName
exportTags	fileLocation	String	Location of the file	parameter; equal to fileLocation
exportTags	dataSize	int	Equal to the number of columns containing data in the output document	local
exportTags	storageQueue	LinkedListQueue()	Queue to store output	local; object; abstract data structure

exportTags	workbook, input	XSSFWorkbook()	Workbooks to store the sheet for input and output	local; object; From Apache POI
exportTags	sheet, inputSheet	XSSFSheet()	Sheets that store the input and output data in a manner Excel can process	local; object; From Apache POI
exportTags	time	LocalDateTime()	Finds the current system time	local; object
exportTags	format	TimeFormatter()	Formats the time into a desired output	local; object
exportTags	tagData[][]	Object[][]	All tags are stored here before they are processed for output	local;
exportTags	rowNum, columnNum	int	Works with the iterator to place file contents in appropriate rows/columns	local;
cropClassMain	x,y,w,h	int	Equal to the parameters of the cropping rectangle (x,y, width, height), used for creating the subimage	private
cropClassMain	cropRec	Rectangle	Reference rectangle equal to the users mouse drawing	private
cropClassMain	image2	bufferedImage	Subimage of img created using x,y,w,h	public
PDFconverter	storageList	ArrayList<>	Stores all pages of a PDF file in image format	public
PDFconverter	pageCount	int	Used as a parameter for the conversion process	private
LinkedListQueue	lst	LinkedList	Used for queue implementation	private;

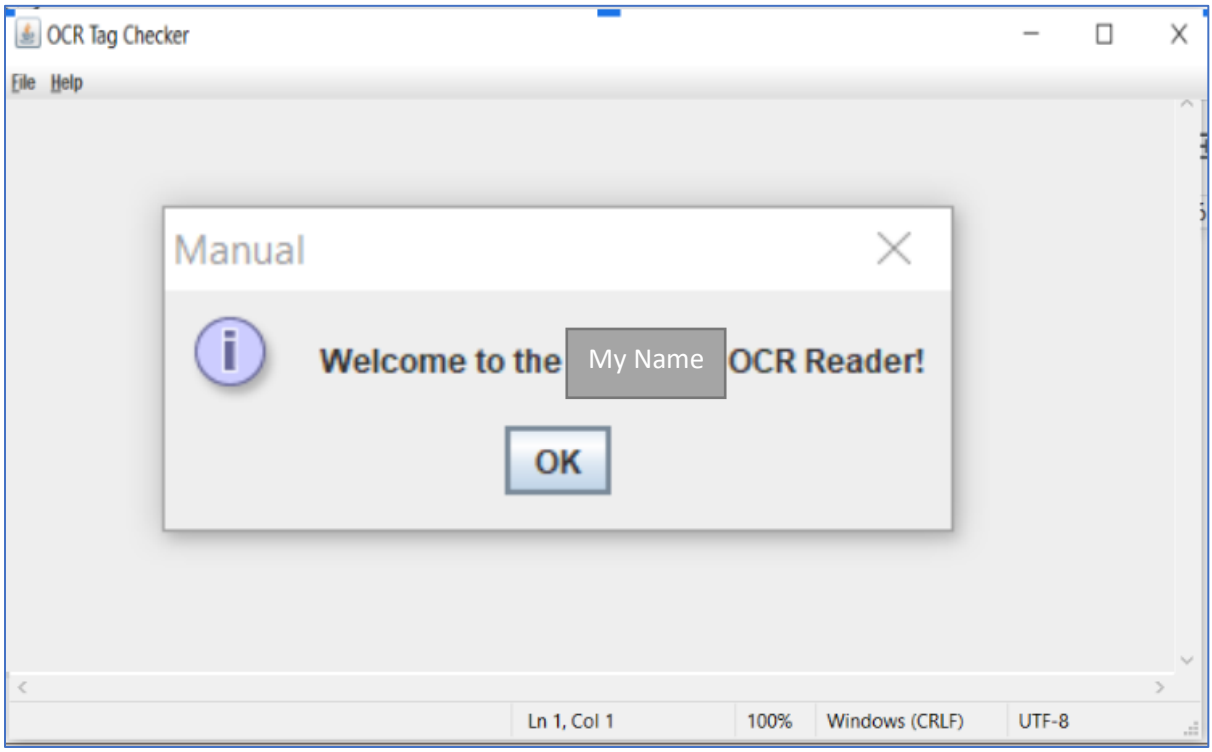
Candidate #: hpg293

Abstract Representations

Program with Loaded File

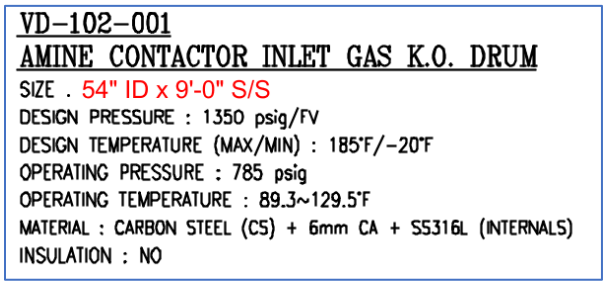


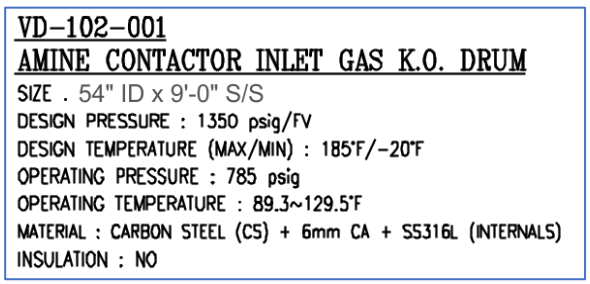
Early User Interface



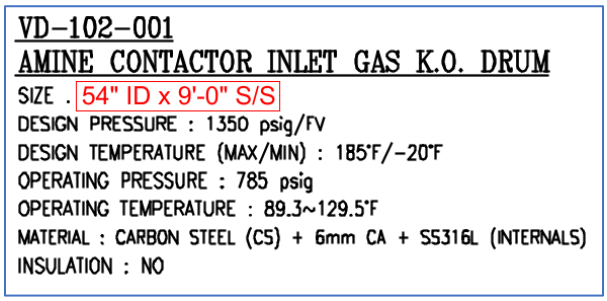
Candidate #: hpg293

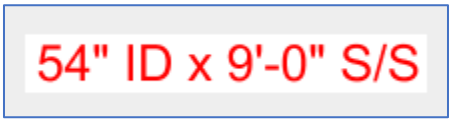
Grayscale (Before and After)





Crop (Before and After)





Excel Output

	A	B	C	D	E	F
1	File Name	Content	Time(opt.)			
2	tagFile.PDF	"alphanumeric tag"	21-Jun			
3	tagFile.PDF	"alphanumeric tag 2"	21-Jun			
4	tagFile2.PDF	"alphanumeric tag 3"	27-Jul			
5	tagFile3.PDF	"alphanumeric tag 4"	28-Jul			
6	tagFile.PDF	"alphanumeric tag 5"	08-Aug			
7						
8						
9						

Organised by time and date

Name of file for organisational purposes

Contents of scanned file are outputted

131 Words