System Analysis and Design Report

Extraction of Structured Information from Dictionary PDFs

Caleb Smith
Josephine Paculio
Joshua Dib
Roger Bernardo

Executive Summary

The structure will first provide the reader with an introduction to the dictionary extraction problem and the stakeholders involved. Following, there will be an in-depth look into the 12 functional and non-functional requirements identified briefly explaining their roles and conditions for a successful product. Moreover, the associated risks and constraints of the project such as PDF quality, package compatibility, and design considerations will be explored and the proposed solution to avoid errors from occurring. In addition, a collection of detailed diagrams will serve as an illustration for both the client and development team to follow as to how the system will interact internally with its own functions as well as externally with users and hardware. Finally, an extensive use case list and test plan will demonstrate the use of the application and assurance that everything has been checked and working as expected.

This report is paramount to all parties as it provides an in-depth understanding of the system being commissioned beyond the face value ensuring users and creators are completely satisfied with the project and a recognition of the relationships going on within.

Table of Contents

1. Introduction

As technology becomes more widespread and readily available, applications and processes have been put in place to improve quality of life and provide society with increased ease and efficiency for tasks. Linguistic analysis is one field to see the benefits technology can have through use of automation and algorithm application.

This document will provide the reader with an in-depth look into the project at hand and why it is a problem that requires a solution. Moreover, several proposed development plans will be detailed explaining the technologies each and their associated consequences both positive and potentially negative aspects. Furthermore, functionality will be defined for viewing and allow for a mutual understanding and expectation of what to the product will provide.

Following on from this document a better insight will be gained into the proposed solution going forward and agreement will be made on how to proceed with implementation.

2. Client details and Project background

Doctor Rachel Hendery is an Associate professor of Digital Humanities at Western Sydney University. She is a Linguist who works on how new technological development helps in finding new ways to study and research about language contact and change in the Pacific.

Her Doctor of Philosophy is about observing changes in relative clauses constructions cross-linguistically which is a project about historical typology. Moreover, her undergraduate degree was a Bachelor of Arts in Linguistics and German at University of Canterbury in New Zealand. Furthermore, Dr Hendery also got a Masters of Arts in Comparative Linguistics and German Medieval Literature at Johann-Goethe University in Frankfurt, Germany.

She supervises postgraduate projects about digital humanities such as data visualization, mapping, language, virtual reality, simulation and other topics such as typology, historical linguistics and contact linguistics.

She is currently working on her linguistic ARC project that deals with observing the relation of contact of different communities and the change in language or communication that might have been caused by the interaction between two or more communities.

3. Problem Statement

The linguistic ARC project involves working with large amount of PDF files that consist of different foreign language dictionaries. Optical character recognition (OCR) software is being used as a tool to enhance the quality and readability of these dictionaries. After going through the OCR software, the PDF files are being used as reference for searching words, phrases and translations.

This process is being done by using the search functionality that is featured in the Acrobat reader. However, the search functionality in Acrobat reader and other PDF viewers are only designed to work on a single document at a time.

This limitation makes the process longer since a single search must be repeatedly done to all different PDF files.

4. System Requirements

4.1 Functional Requirements

FR01 – Dynamic File upload	
Summary	The user will start the process by selecting a file to extract from their local system. This file will be used in the extraction and output of the data.
Justification	This application is a tool for research as a whole and not a specific dictionary itself, as such it needs to be compatible with accepting different files and formats
Success Conditions	• Upload requested file from the local storage • Accept multiple inputs at once
Failure Conditions	• Files must be hardcoded

FR02 – Extract dictionary entries	
Summary	Upon selecting a file and entering the search parameters the application will then extract the complete data from the file and begin splitting it up into individual entries and headings. These results are to be used in an onscreen output and eventual export.
Justification	The previous process involved hours of manual labour spent typing out the entries one by one, this is a large waste time and damaging to a user's workflow. Extracting and splitting up the entries automatically will allow more focus to be spent analysis
Success Conditions	• Outputs the entries correctly and applies error checking • Splits apart entries into categories • Generic formulas used to work on different formats • Dynamically OCR a file during upload
Failure Conditions	• Strict algorithms used to split up data doesn't work on various files • Entries incorrectly split

FR03 – Output formatted data	
Summary	Before exporting the extracted information, it should be presented in the application using tables and relevant packages. Doing so will allow the user to confirm the process went as expected and provides a chance to fix any errors in the

	moment rather than restarting the process or going back to edit it within the spreadsheet.
Justification	It is important for the user to have the chance to double check their result before exporting, it saves time and frustration to the alternative of clogging up storage with incorrect files until it is correct. Plus, the onscreen visuals provide a boost to user interaction and reflects a more positive experience using the system.
Success Conditions	• Functional presentation using stylesheets and frameworks • Provides ease of use when navigating data • Foundation allows for further features and manipulation
Failure Conditions	• Format does not promote readability

FR04 – Export formatted data	
Summary	After interpreting the selected data and outputting for the user to view and further interact with there will be an option to export the table to their local drive.
Justification	Exporting the state allows the user to revisit the data at a later date or transfer to another device, without this the user would have to repeat the process every time the dataset is used. Moreover, it allows for more powerful formulas to be applied in external programs such as excel.
Success Conditions	• Variety of file options provided (ie. csv, pdf, xml) • Maintains format of the onscreen display
Failure Conditions	• Unsuitable options of file type

FR05 – Data manipulation	
Summary	After outputting the structured data onto the application screen, the users will be able to apply search terms and filter out unnecessary entries.
Justification	It is important to allow this interaction before the data is exported as it ensures only the relevant information is saved and does not require the user to apply a new filter on the whole set when referring to a specific query
Success Conditions	• Filter list applicable to individual columns • Edit entries if incorrectly interpreted • Able to search of desired terms

Failure Conditions	• Does not provide a search or filter function • User cannot interact with the output data

FR06 – Error correction	
Summary	A combination of features will be applied such as character splitter, column count, and whitespace length will be set in the initial extraction to help the process provide an accurate interpretation. In addition, post-output editing features and a simple spell checker will be used to clean up the output in the moment.
Justification	Due to the variety of dictionary formats available in addition to the different levels of quality the extracted string is likely to have issue associated with it. By adding extra flags or methods to fix up the data in the application itself saves time further down the line double checking each entry after already being exported.
Success Conditions	• Able to manually edit information • Initial search parameters to fine tune the extraction
Failure Conditions	• Offers limited to no option for in application editing • Extraction options/process is rigid and doesn't accommodate fixes

4.2 Non-Functional Requirements

NFR01 - Project Documentation	
Summary	Files such as online help and troubleshooting help will be available and in depth to address any queries a user may have. Furthermore, the backend of the project will include thorough commenting explaining what each section of code does, what interactions it has.
Justification	The project will introduce a lot of change in many processes and documentation will be required to aid the transition. Computer literacy varies between users and so it must be accommodating to ensure every user understands how to use the product, otherwise workflow will be disrupted and the members may turn away from the product due to confusion. In addition, for the future development code commenting will address confusion as to methods use and their interactions.
Success Conditions	• Passes control groups in which a task is provided and the user is able to complete it given the documentation provided, conducted for staff and end user tasks

	• Members are able to identify which part of the code does what function and understand the dependencies and interactions
Failure Conditions	• Instruction documents are vague does not contain examples or relevant tips • Code comments are not thorough and introduces the uncertainty of how to integrate

NFR02 - User Accessibility	
Summary	The product should be compliant with industry accessibility standards to aid the users with disabilities. All elements and pages must keep this in mind and exercise practices such as image descriptions and easy to click items.
Justification	Product will be used by multiple users but not everyone has the same abilities. If a user feels ignored and outcast by their experience, they will not be willing to continue using the product.
Success Conditions	• Page navigation is possible using only a keyboard • Errors are labeled and provide an easy to understand fix
Failure Conditions	• Non-highlightable text is used preventing text to speech software • Design used does not account for vision impairment

NFR03 - Platform Capabilities	
Summary	The end user website will be designed to accommodate the users choice of web browser as well as platform of choice. It will not provide the user with any extra tasks to do such as downloading a compatible browser or plugin.
Justification	New devices, platforms, and operating systems are regularly released each with their own native browser and capabilities, failure to make the system universal will lead to faulty operation and member satisfaction
Success Conditions	• Same performance and capabilities on variety of browsers • Window screen does not alter the product design
Failure Conditions	• Functions are dependent on features that are not preinstalled

	• Resizing the window causes distortion and is hard to use

NFR04 - Operational Efficiency	
Summary	The program efficiency will focus on improving the primary features of data conversion and report generation. We aim to cut down operation times and artificial losses of efficiency such as manual data entry or entry formatting.
Justification	If the implemented solution does not increase the company efficiency the development process has not been worthwhile and will leave the client back where they started with a problematic system.
Success Conditions	• Workflow increases when in use • Time to perform tasks such as sheet generation and search queries perform better than the previous process
Failure Conditions	• Functions are not integrated with each other and possible improvements exist • System performance and design does not offer enhancement

NFR05 - Maintainability	
Summary	The system should be designed in such a way that allows for additions later in the life cycle without the need to redesign other functions and their interactions. Bugs that are identified can be addressed and further improvements to efficiency or usability are able to be easily integrated.
Justification	Research is a never-ending process and new requirement or focuses will appear over time, as such the technology should be well prepared to grow as needs change.
Success Conditions	• Identified bugs can be easily traced in code • Made compatible with new technology and systems
Failure Conditions	• System is left at launch version with minimal features • Difficulty in adding new feature due to poor infrastructure

NFR06 – Graphical user interface

Summary	The application will be presented via a graphical user interface rather than a script format run through the command line.
Justification	Using a command line is an intimidating task for users who are not experience which could alienate them and prevent the use and approval of the software. In addition, using a GUI allows for users to interact during the process such as viewing before export, data editing, filtering.
Success Conditions	• Intuitive design for new system users • Fits the clients and projects theme • Implements proper semantics and styling principles
Failure Conditions	• Poorly structured elements across the page • Confusing workflow and poor instructions

5. Risks and Constraints

5.1 Risks and Resolutions

Risks	**Resolution**	**Type of Risk**
The PDF dictionaries vary in quality, some dictionaries are scanned PDFs which we may have trouble using OCR technologies on them.	Extensive error checking and testing for after the initial version of the project has been released.	External
Project team has never worked together before – inexperienced team dynamic	Constant communication between team members throughout development.	Internal
OCR technology may not be 100% accurate going into the development process.	Substantial testing phase to be implemented.	External
Project deliverables may be affected by individual group member circumstances such as sickness or other serious misadventures that may affect group dynamics and progress.	Constant communication between team members throughout development to allow for coverage of workload.	Internal
Individual workload clashing with other university/work commitments	Constant communication between team members throughout development to allow for coverage of workload.	Internal

Reliance on university provided server as code repository	Set up Git so each member can work on specific development areas on their own and merge changes,	External

5.2 Design Considerations

Human computer interaction (HCI) principles aims to increase usability of software systems for users of all abilities. Moreover, good software interfaces improve interaction between users and systems by making interfaces as user friendly and receptive to the needs of the user. Specifically, one aim of the project is to create a simple, straight forward and user-friendly experience for users of all abilities. Users of the web application will have almost instant knowledge on how to properly use the main functions of the application for their needs through a well-designed form with a simple layout. Furthermore, instructions and validation will be present throughout the process which will guide the user on how to properly use the system if they make an error. If the user is still unsure on how to use the system, they can find text instructions and an instructional video in the navigation bar. Risks must be considered when designing an interface. Explicitly, designing an application that is aesthetically pleasing to all users is a difficult process as many users will not begin to use a system if they do not have a positive first impression (Interaction Design Foundation 2019). Therefore, finding design and a color scheme that is consistent and tranquil is important to gain the interest of the user and will reduce the risk of the user leaving and finding an alternative application to use. Thus, to reduce risks – it is important to learn from other similar successful web applications in design and to overall reduce the number of clicks and learning a user must perform to have a successful experience with the web application.

5.3 Hardware Requirements

This program is compiled using Electron and Chromium architecture and requires no more than a standard browser. While it is possible to operate the application on different systems below details the recommended minimum for a smooth service.

Windows

To use Dictionary Extractor on Windows, you'll need:

- Windows 7, Windows 8, Windows 8.1, Windows 10 or later
- An Intel Pentium 4 processor or later that's SSE2 capable

Note: Servers require Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, or Windows Server 2016.

Mac

To use Dictionary Extractor on Mac, you'll need:

- OS X Yosemite 10.10 or later

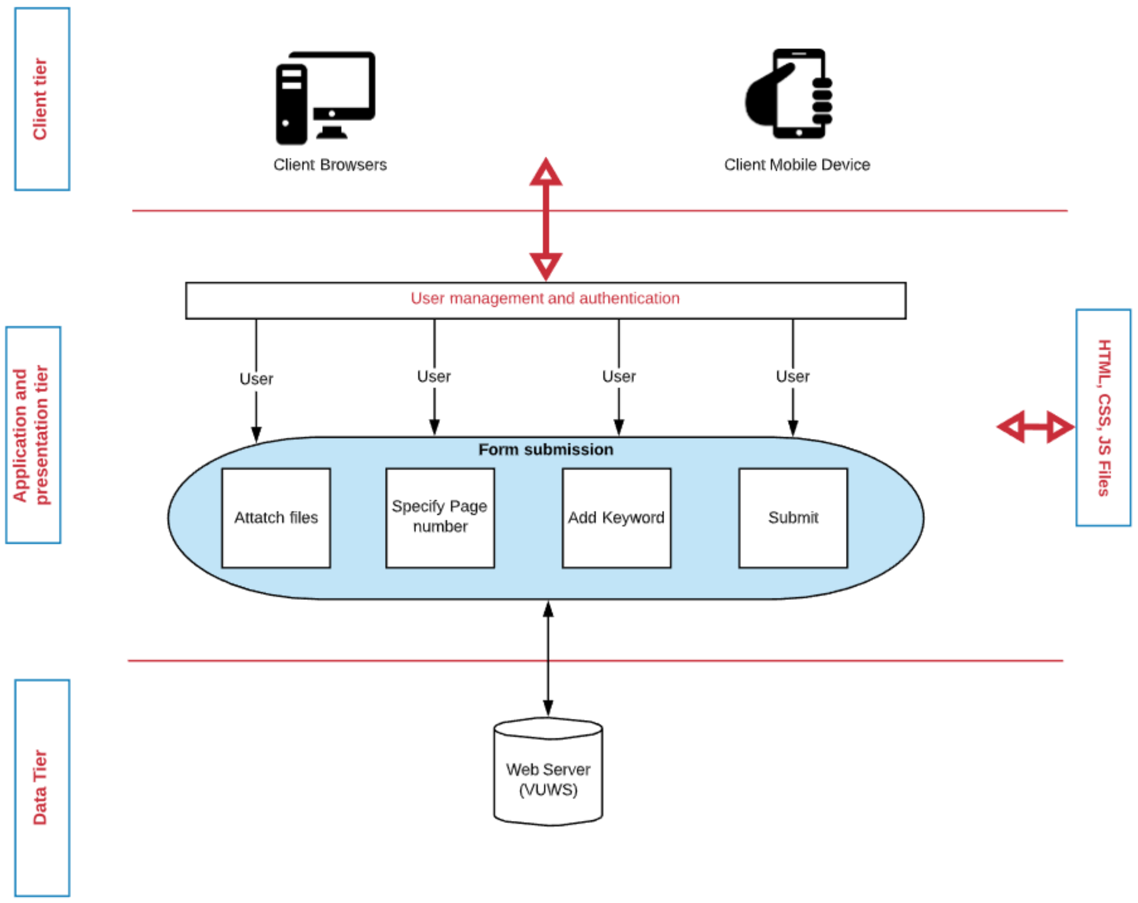
Linux

To use Dictionary Extractor on Linux, you'll need:

- 64-bit Ubuntu 14.04+, Debian 8+, openSUSE 13.3+, or Fedora Linux 24+

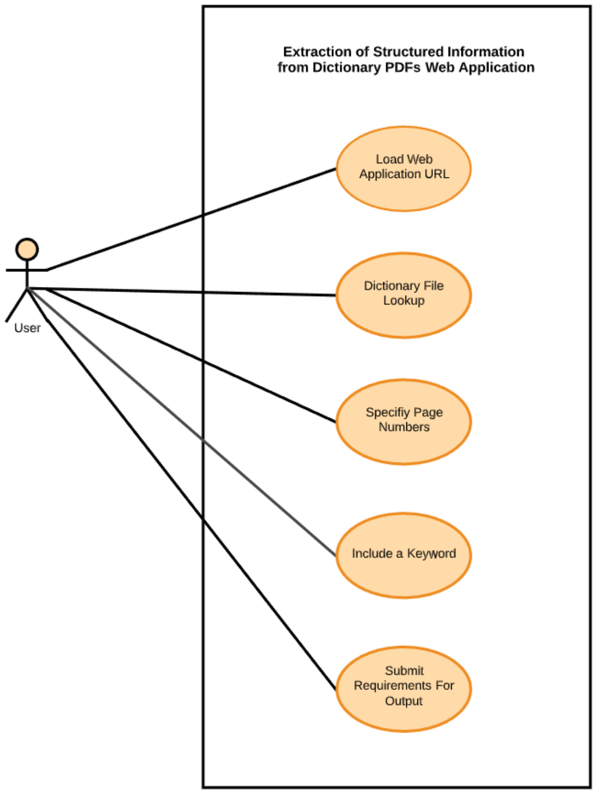
- An Intel Pentium 4 processor or later that's SSE2 capable

5.4 Software Architecture



6. Detailed System Design

6.1 Use Case Diagram


Extraction of Structured Information from Dictionary PDFs Web Application
User
Load Web Application URL
Dictionary File Lookup
Specifiy Page Numbers
Include a Keyword
Submit Requirements For Output

6.2 Expanded Use Cases

Use Case:	**Web application output sequence**
Category:	Core
Actors:	User
Description:	User loads web application URL through desktop icon. Dictionary PDF file lookup prompts file explorer where user can pass documents to web application. User specifies page numbers for the application to analyze. User creates a unique keyword for the application to find and process for output. User requirements are submitted for output and is printed to the screen.
Purpose:	Interaction user has with the web application system to gather their required information from linguist dictionaries.
Notes:	Use case steps 2, 3 and 4 do not need to be completed in that specific order as information is not processed until step number 5.
Pre-conditions(s)	User must have an internet connection and an internet browser.
Post-conditions(s)	Output information is printed to the screen in a table format.

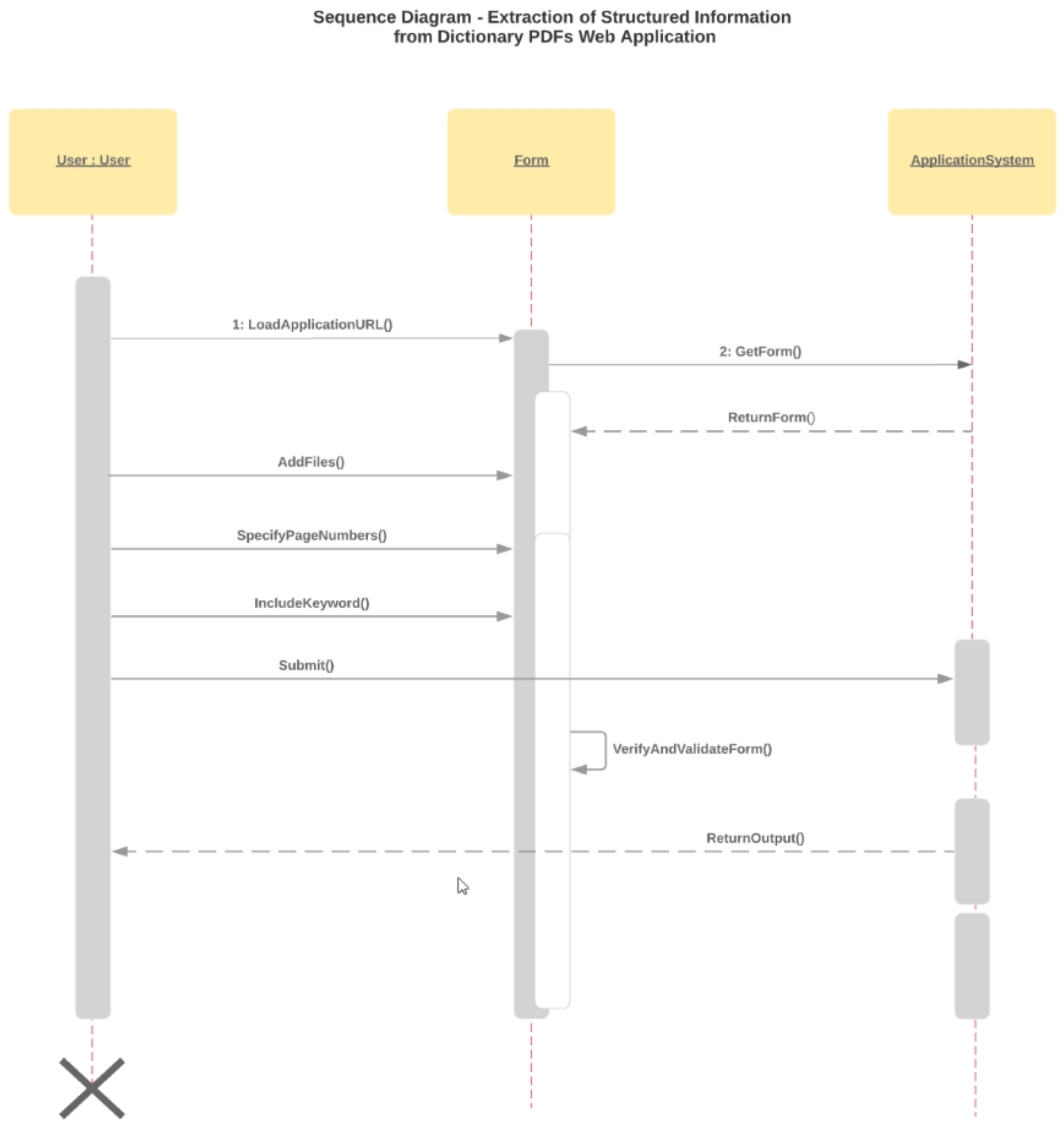
Typical Course of Events

#	Actor Action	System Event	#	System Response
1	Enter form details	Validate that there is a file attached	**2**	Output message “Please attach a valid PDF file”
3		Validate if page number input is numeric	**4**	Output message “Please enter a valid number”
5		Validate if keyword is not numeric	**6**	Output message “Please enter a valid keyword”
6		Validate that compulsory input has been entered	**8**	System will respond with validating entire form

Alternative events

#	Actor Action	System Event	#	System Response
A1	Enter form details	Validate new details after submission	**A2**	Update output details
A2		Refresh main page	**A3**	Clear output for new session

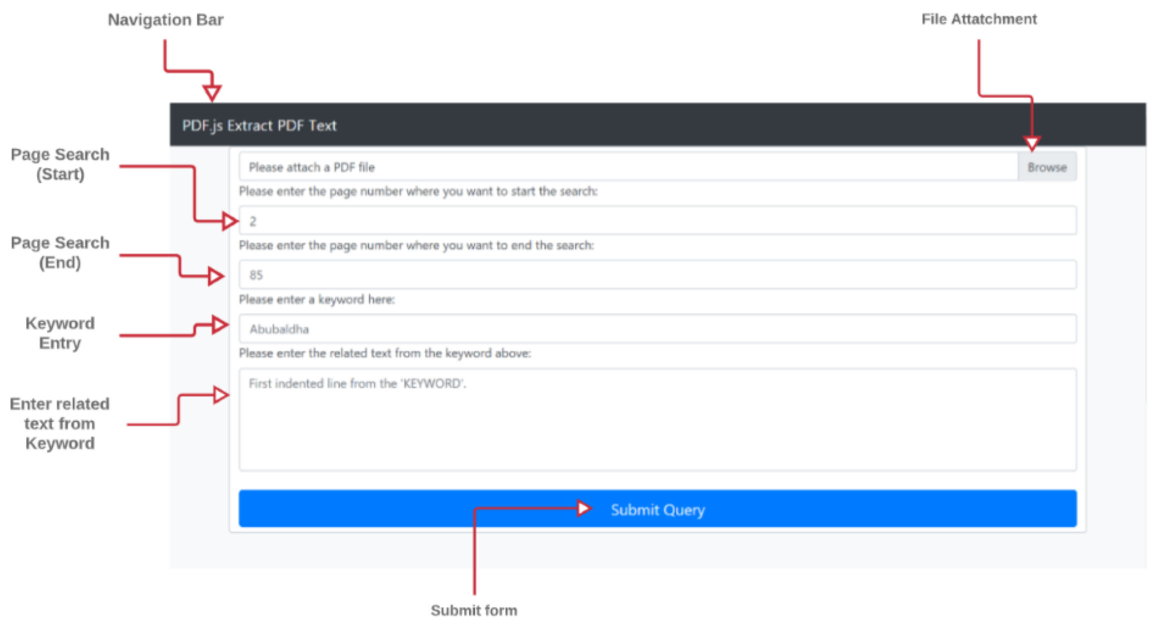
6.3 Sequence Diagrams



- Step 1: User boots application URL from desktop
- Step 2: Application returns form and web application main page to user
- Step 3: User adds dictionary files
- Step 4: User specifies page numbers parameters
- Step 5: User adds a keyword
- Step 6: User submits all details to web application
- Step 7: System validates and verify output is recognizable and correct
- Step 8: Application System returns output to screen

6.4 Screen Designs

Form screen



Please attach a PDF file

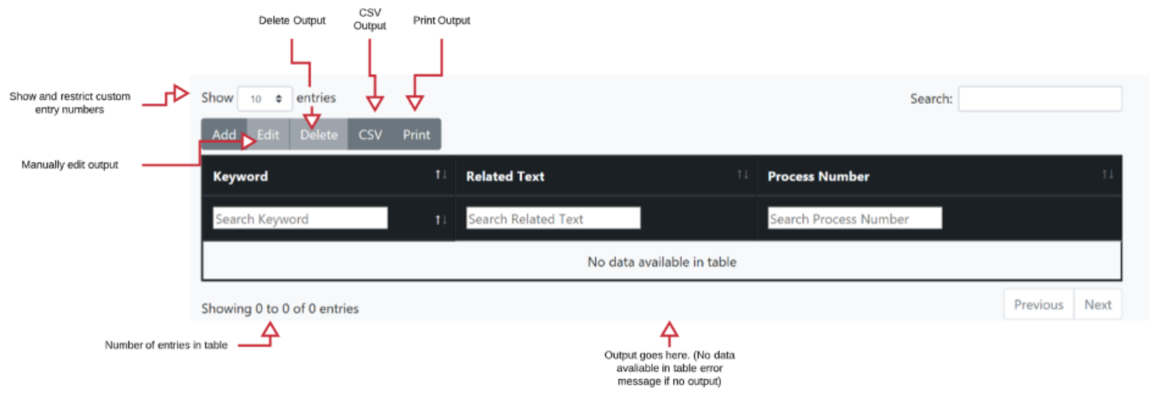
Please enter the page number where you want to start the search: 2

Please enter the page number where you want to end the search: 85

Please enter a keyword here: Abubaldha

Please enter the related text from the keyword above: First indented line from the 'KEYWORD'.

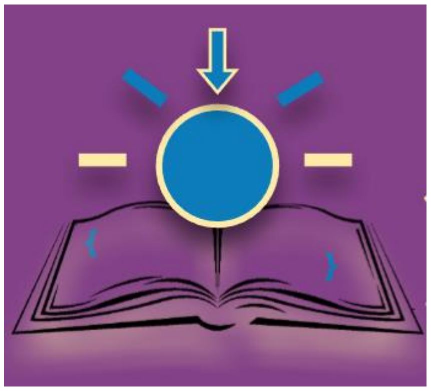
Application Output



Keyword	Related Text	Process Number
No data available in table		

Showing 0 to 0 of 0 entries

Application Logo



7. Test Plan

7.1 Features/use cases to be tested

Feature/Use case to be tested	Types of Testing	Pass fail criteria	Personnel	When and Where	Training	Risks	Contingencies
Form interface	Design requirement testing	Need work if not as per to the agreed design	Team Member: Caleb	Week 6 Western Sydney University – Kingswood campus	Not needed	The form might not satisfy or cover all requirements	Designate more resources to solve the problem
	Acceptance testing	Need work if not accepted and not as per to the agreed design	Client: Rachel	Week 6 Online Meeting	Needed provided prior to testing	The form might be too complicated for the users	Provide technical support during testing and demonstration
Form validation	Design requirement testing	As specified with the test data	Team Member: Caleb	Week 10 Western Sydney University – Kingswood campus	Not needed	Does not completely eliminate errors from the input	Designate more resources to solve the problem
Dynamic attachment of an OCRed PDF file	Design requirement testing	Need work if not as per to the agreed design	Team Member: Josephine	Week 6 Western Sydney University – Kingswood campus	Needed provided prior to testing	May not be able to read the right document	Clearly specify the document that the system received from the user
	Acceptance testing	Need work if not accepted and not as per to the agreed design	Client: Rachel	Week 6 Online Meeting	Needed provided prior to testing	May not be able to read the right document	Clearly specify the document that the system received from the user
Dynamic attachment of more than one PDF file	Design requirement testing	Need work if not as per to the agreed design	Team Member: Josephine	Week 12 Western Sydney University	Needed provided prior	May not be able to handle large amount	Make dynamic attachment of only one

				– Kingswood camp us	to testing	of PDF attachments	document available
	Acceptanc e testing	Need work if not accepted and not as per to the agreed design	Client: Rachel	Week 12 Online Meeting	Neede d provide d prior to testing	May not be able to handle large amount of PDF attachments	Make dynamic attachment of only one document available
Extraction of text from the attached OCRed PDF file	Design requirement testing	As specified with the test data	Team Member: Roger	Week 6 Western Sydney University – Kingswood camp us	Not needed	May not be able to accurately extract text with special characters	Allow the user to edit and correct the output of the system
Organize extracted text into structured format	Design requirement testing	Need work if not as per to the agreed design	Team Member: Roger	Week 13 Western Sydney University – Kingswood camp us	Not needed	May not be able to correctly organize the extracted text	Allow the user to edit and correct the output of the system
	Acceptanc e testing	As specified with the test data	Client: Rachel	Week 13 Online Meeting	Neede d provide d prior to testing	May not be able to correctly organize the extracted text	Allow the user to edit and correct the output of the system
Conduct error checking	Design requirement testing	Need work if not as per to the agreed design	Team Member: Roger	Week 13 Western Sydney University – Kingswood camp us	Not needed	May not be able to detect all possible errors in the output	Designate more resources to solve the problem
	Acceptanc e testing	As specified with the test data	Client: Rachel	Week 13 Online Meeting	Neede d provide d prior to testing	May not be able to detect all possible errors in the output	Designate more resources to solve the problem

Place organized text into Datatables	Design requirement testing	Need work if not as per to the agreed design	Team Member: Josh	Week 6 Western Sydney University – Kingswood campus	Not needed	May not be able to accurately place text into Datatables	Use the original extracted text if Datatables are not accurate
	Acceptance testing	As specified with the test data	Client: Rachel	Week 6 Online Meeting	Needed provided prior to testing	May not be able to accurately place text into Datatables	Use the original extracted text if Datatables are not accurate
Turn extracted text into JSON file/s	Design requirement testing	Need work if not as per to the agreed design	Team Member: Josh	Week 13 Western Sydney University – Kingswood campus	Not needed	May not be able to accurately place text into JSON file/s	Use the original extracted text if Datatables are not accurate
Save system's output to user's system	Design requirement testing	Need work if not as per to the agreed design	Team Member: Josh	Week 13 Western Sydney University – Kingswood campus	Needed provided prior to testing	May not be able save output due to system incompatibilities	Have a backup of the system's output until the user is able to save the output to user's system
	Acceptance testing	Need work if not accepted and not as per to the agreed design	Client: Rachel	Week 13 Online Meeting	Needed provided prior to testing	May not be able save output due to system incompatibilities	Have a backup of the system's output until the user is able to save the output to user's system
Apply word filter	Design requirement testing	Need work if not as per to the agreed design	Team Member: Josh	Week 13 Western Sydney University	Not needed	May not be able to filter words with	Show any related text from the word

				– Kingswood camp us		special characters	that was used as filter
	Acceptanc e testing	Need work if not accepted and not as per to the agreed design	Client: Rachel	Week 13 Online Meeting	Neede d provide d prior to testing	May not be able to filter words with special characters	Show any related text from the word that was used as filter
Convert PDF file/s into an image file	Design requirement testing	As specified with the test data	Team Member: Roger	Week 13 Western Sydney University – Kingswood camp us	Not needed	May not be able to convert some pdf files into image	Clearly specify to the user all the informatio n retained from the PDF file after converting it to an image file
Extract text from an image using Optical Character Recognition (OC R)	Design requirement testing	As specified with the test data	Team Member: Roger	Week 13 Western Sydney University – Kingswood camp us	Not needed	May not be able to accurately extract text from an image file	Clearly specify to the user all the information extracted from the image file after performing OCR

7.2 Candidate Test Cases/ Test data

Validation Test

Use Case / Feature	Attach PDF file/s – Validation Testing		
Interface Reference:	BF1		
Test Purpose	To check that all mandatory fields have been filled and that all fields follow the guidelines regarding content and lengths of characters		
Expected Results	See Individual Test Data Sets		
Success/Failure	Success		
Test Data			
Column Name	Set 1	Set 2	Set 3
Attached File (mandatory)	Wirangu - English Dictionary.pdf	djabugay_dictionary _quinnetal.pdf	NULL
Page number to start the search (mandatory)	0	8	0
Page number to end the search (mandatory)	1	8	1
Keyword (mandatory)	abubaldha	axe	NULL
Expected Results	Pass	Pass	Fail
Success/Failure			
Date Tested			
Actions to be taken			

Logic Test

Method Name	getPageContent – Logic Test			
Method Type	Public Function			
Parameters	pageNumber (Int) , pdfDocumentInstance (file)			
Return Value	Array of contents of a page			
Pseudo-code	1. Get a single page out of a pdf file 2. Get the details/contents of the single page from the pdf file 3. Return the contents of the single page from the pdf file 4. Return the page containing its contents from the single page from pdf file			
Test Data				
Column	Set 1	Set 2	Set 3	Set 4
pageNumber	1	0	8	0
pdfDocumentInstance	Wirangu - English Dictionary.pdf	Wirangu - English Dictionary.pdf	djabugay_dictionary _quinnetal.pdf	djabugay_dictionary _quinnetal.pdf
Expected Results	Return page 1 of Wirangu - English Dictionary.pdf	Error since the page numbering starts with 1 and not 0	Return page 1 of djabugay_dictionary _quinnetal.pdf and its contents	Error since the page numbering starts with 1 and not 0

	and its contents			
Actual Results				
Success/Failure				
Date Tested				
Actions to be taken				

Method Name	combinePagesToOne – Logic Test			
Method Type	Public Function			
Parameters	Contents (array of pages that contains page contents)			
Return Value	Array of all contents from all pages			
Pseudo-code	1. Loop trough the array of pages 1. Loop trough the contents of each pages 1. Copy all contents of a page in an array called items 2. Return array called items which contains the contents of all pages			
Test Data				
Column	Set 1	Set 2	Set 3	Set 4
Contents	Page 1 of PDF file	Page 1 to 2 of PDF file	All pages of PDF file	Page 0 of PDF file
Expected Results	Return all contents of page 1 of the pdf file	Return all contents of page 1 to 2 of the pdf file	Return all contents of all pages of the pdf file	Error since the page numbering starts with 1 and not 0
Actual Results				
Success/Failure				
Date Tested				
Actions to be taken				

Method Name	combineLineByLine – Logic Test			
Method Type	Public Function			
Parameters	Items (array of all contents from all pages)			
Return Value	Array of lines that are arranged line by line as it appears in the pdf document			
Pseudo-code	1. Loop trough the array of contents from all pages 1. Copy all contents on a single line to an array called lines 2. Then, move to the next line 2. Return array called lines which contains each line of string in the pdf document			
Test Data				
Column	Set 1	Set 2	Set 3	Set 4
Items	1	2	All contents of all pages of pdf file	0
Expected Results	Return an array of lines from the contents of page 1	Return an array of lines from the contents of page 1 to 2	Return an array of lines from the contents of all pages in the pdf file	Return an empty array because there is nothing to copy from

Actual Results				
Success/Failure				
Date Tested				
Actions to be taken				

Method Name	getIndentationDifference – Logic Test			
Method Type	Public Function			
Parameters	Keyword (string), lines (array of string lines from the contents of pages of the pdf file)			
Return Value	Indentation difference (double) between keyword and related text. The return value is rounded to the nearest 1 decimal place.			
Pseudo-code	1. Loop trough the array of lines of contents from the pdf file 1. If keyword is in the current line 1. Calculate indentation difference by subtracting the current line's x-coordinate to the keyword's x-coordinate 2. Round the indentation difference to the nearest 1 decimal place 3. Return indentation difference			
Test Data				
Column	Set 1	Set 2	Set 3	Set 4
PDF file	Wirangu - English Dictionary.pdf	djabugay_dictionary_quinnetal.pdf	lillie_girawa[1999]	Wirangu - English Dictionary.pdf
keyword	abubaldha	axe	aipar	NULL
lines	Lines from page 1 of the pdf file	Lines from page 9 of the pdf file	Lines from page 2 of the pdf file	Lines from page 1 of the pdf file
Expected Results	28.8	11.2	28.8	Error, since keyword to be searched is not provided
Actual Results				
Success/Failure				
Date Tested				
Actions to be taken				

Method Name	collectXCoordinate – Logic Test
Method Type	Public Function
Parameters	lines (array of string lines from the contents of pages of the pdf file), indentationDifference (double)
Return Value	Array of x-coordinates (double) of keywords
Pseudo-code	1. Loop trough the array of lines of contents from the pdf file 1. If the difference of x-coordinates of current line and the next line matches the indentationDifference 1. If the x-coordinate of the current line is not already in the xCoords array then, 1.1.1.1 Copy x-coordinate of the current line to an array called xCoords
Test Data	

Column	Set 1	Set 2	Set 3	Set 4
PDF file	Wirangu - English Dictionary.pdf	djabugay_dictionary_quinnetal.pdf	lillie_girawa[1999]	NULL
lines	Lines from page 1 of the pdf file	Lines from page 8 of the pdf file	Lines from page 2 of the pdf file	0
Indentation difference	28.8	11.2	28.8	NULL
Expected Results	89.76001, 324.4799	71.99982564500002, 304.558871375	90.04060000000001, 90.0406, 90.04060000000003, 333.04060000000004, 333.0406, 333.0405999999999	Error since there is no attached PDF file. Error since there is no indentation difference included to be used as comparison
Actual Results				
Success/Failure				
Date Tested				
Actions to be taken				

Method Name	fillTable – Logic Test
Method Type	Public Function
Parameters	lines (array of string lines from the contents of pages of the pdf file), indentationDifference (double), xCoordinatesOfKeyword (array of doubles)
Return Value	Array of x-coordinates (double) of keywords
Pseudo-code	see below

```
1.  Get access to the table in the HTML page
2.  Loop trough the array of lines of contents from the pdf file
    1.  If the x-coordinate of the current line is in the array
    called xCoordinateOfKeyword then,
        1.  Get the keyword from the current line
        2.  Get the related text from the current line
        3.  Insert the keyword to the keyword section of the table
        4.  Insert the related text to the related text section of the
        table
    2.  Else, if the x-coordinate of the current line is not in the array
    called xCoordinateOfKeyword then,
        2.2.1 insert current line to the related text section of the table
```

Test Data

Column	Set 1	Set 2	Set 3	Set 4
PDF file	Wirangu - English Dictionary.pdf	djabugay_dictionary _quinnetal.pdf	lillie_girawa[1999]	NULL
lines	Lines from page 1 of the pdf file	Lines from page 8 of the pdf file	Lines from page 2 of the pdf file	0

Indentation difference	28.8	11.2	28.8	NULL
xCoordinatesOfKeyword	89.76001, 324.4799	71.99982564500002, 304.558871375	90.04060000000001, 90.0406, 90.04060000000003, 333.04060000000004, 333.0406, 333.0405999999999	NULL
Expected Results	Fill the table with correct keyword and related text	Fill the table with correct keyword and related text	Fill the table with correct keyword and related text	Error since there is no attached PDF file. Error since there is no indentation difference included to be used as comparison. Error since there is no x-coordinate to be used as comparison
Actual Results				
Success/Failure				
Date Tested				
Actions to be taken				

Method Name	isKeyword– Logic Test
Method Type	Public Function
Parameters	Index (int), Lines (array of string lines from the contents of pages of the pdf file), indentationDifference (double)
Return Value	True or false
Pseudo-code	1. If the next line is within the range of the array of lists then, 1. Calculate indentation difference by subtracting the current line's x-coordinate to the keyword's x-coordinate 2. Round the indentation difference to the nearest 1 decimal place 3. If the rounded indentation difference matches the indentationDifference from the parameters then, 1. Return true 2. Else if the next line is not within the range of the array of lists then, 2.1 Return false
Test Data	

Column	Set 1	Set 2	Set 3	Set 4
PDF file	Wirangu - English Dictionary.pdf	djabugay_dictionary _quinnetal.pdf	lillie_girawa[1999]	NULL
index	0	0	0	NULL
lines	Lines from page 1 of the pdf file	Lines from page 8 of the pdf file	Lines from page 2 of the pdf file	0
Indentation difference	28.8	11.2	28.8	NULL
Expected Results	Return true on lines with indentation difference that matches the indentationDifference given in the parameter. Return false on lines with indentation difference that does not matches the indentationDifference given in the parameter.	Return true on lines with indentation difference that matches the indentationDifference given in the parameter. Return false on lines with indentation difference that does not matches the indentationDifference given in the parameter.	Return true on lines with indentation difference that matches the indentationDifference given in the parameter. Return false on lines with indentation difference that does not matches the indentationDifference given in the parameter.	Return false since PDF file, index and indentation difference are not provided
Actual Results				
Success/Failure				
Date Tested				
Actions to be taken				

Method Name	isOnSameLine – Logic Test
Method Type	Public Function
Parameters	Item1 (content on a page that contains y-coordinates of the text/string of the object), Item2 (content on a page that contains y-coordinates of the text/string of the object)
Return Value	True or false
Pseudo-code	1. If y-coordinates of both item1 and item2 are not the same then, 1. false 2. else, if y-coordinates of both item1 and item2 are the same then, 2.1 return true

```
1.    If y-coordinates of both item1 and item2 are not the same then,
      1. false
2.    else, if y-coordinates of both item1 and item2 are the same then,
      2.1 return true
```

Test Data

Column	Set 1	Set 2	Set 3	Set 4
PDF file	Wirangu - English Dictionary.pdf	djabugay_dictionary _quinnetal.pdf	lillie_girawa[1999]	NULL

Item1	Lines[0]	Lines[0]	Lines[0]	NULL
Item2	Lines[1]	Lines[1]	Lines[1]	NULL
Expected Results	Lines[0]'s y-coordinate = 89.76001 Lines[1]'s y-coordinate = 89.76001 Since both y-coordinates is the same, the function will return true	Lines[0]'s y-coordinate = 683.721 Lines[1]'s y-coordinate = 696.8011 Since both y-coordinates is not the same, the function will return false	Lines[0]'s y-coordinate = 66 Lines[1]'s y-coordinate = 66 Since both y-coordinates is the same, the function will return true	Error since there is no attached PDF file. Error since item1 is not provided. Error since item2 is not provided.
Actual Results				
Success/Failure				
Date Tested				
Actions to be taken				

8. Conclusion

Upon concluding this document there are several key takeaways including a breakdown of the project's requirements, the possible development risks and how they will be dealt with, an insight into the applications processes and their related interactions with the surrounding environment, as well as an understanding of functions and how they are being tested for optimal performance.

Following on from here development will have been largely planned and documented from start till the final release barring any additional features, in turn allowing for full focus on application development ultimately resulting in the desired solution.