# SMART CITATION MANAGER

BY

TEH ZHI YUN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF BUSINESS INFORMATION SYSTEM(HONS)

Faculty of Information and Communication Technology

(Kampar Campus)

JANUARY 2019

**UNIVERSITI TUNKU ABDUL RAHMAN**

# REPORT STATUS DECLARATION FORM

**Title**: Smart Citation Manager

**Academic Session**: May 2019

I Teh Zhi Yun

**(CAPITAL LETTER)**

declare that I allow this Final Year Project Report to be kept in

Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature)

(Supervisor's signature)

**Address**:
Lot 1409, Lorong A-13
81500, Pekan Nanas,
Johor

Supervisor's name

**Date**:

**Date**:

SMART CITATION MANAGER

BY

TEH ZHI YUN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF BUSINESS INFORMATION SYSTEM(HONS)

Faculty of Information and Communication Technology

(Kampar Campus)

JANUARY 2019

# DECLARATION OF ORIGINALITY

I declare that this report entitled “Smart Citation Manager” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____

Name : Teh Zhi Yun

Date : 8 April 2019

# ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my supervisor, Mr Yong Tien Fu for the guidance and assistance in this Smart Citation Manager Project. He gave me a lot of advices and encouragement whenever I have encountered the problems in completing the project. Besides that, he always provided thought-provoking feedback which further motivate me to accept difficulty and challenges faced.

My thanks and appreciation also go to my friends and family who are directly or indirectly support me in completing the projects and throughout the course.

# ABSTRACT

This project is a desktop-based reference management system named Smart Citation Manager that allow the user to easily organize and manage a large volumes of references during the process of literature review. However, there is no a single software that enable user to organize and plan their research. In order tackle the stated problem, Smart Citation Manager focused on providing improved solution which a single platform that allow user to draft their thoughts and literature. The goals to be achieved included (i) a reference management module to store, manage reference data and create citation and bibliography in formatted to a citation style required for publication. (ii) PDF viewer to render PDF document within the system environment. (iii) Mind mapping technique allow the mapping of ideas, its source and related reference item within single interface. (iv) Text extraction algorithm facilitate the retrieving text from PDF document to mind map. The project is developed using prototyping methodology which used to ensure constant user feedback to result in a useable system.

# TABLE OF CONTENT

# LIST OF TABLES

# LIST OF FIGURES

# CHAPTER 1 INTRODUCTION

Reference management is an essential research task for most researcher. It includes collecting and organizing the reference, creating citation and bibliographies and eventually integrating the reference into a manuscript (drafting, writing and citing). However, there is a lack of a single software tools in the market try to facilitate reference management process. For example, Endnote and Zotero help managing reference, Adobe Acrobat PDF Reader DC help reading and annotating PDF document, and Text document or Word document help to draft a paper. It is difficult to keep track everything that worked in distinct piece of software. Therefore, an improved solution is proposed to provide a single platform that support researcher in organizing, reading, citing, and drafting new literature by bundling several features:

1. A reference management module for managing reference, creating citation and bibliographies.
2. A PDF viewer for reading and annotating PDF literature such as creating comment, highlight on text and bookmarks.
3. A mind mapping module for drafting and providing a visual way to keep the information organized such as document, draft, and reference item.
4. Text extraction algorithm for promote retrieving text from PDF and place it in mind map using simple mouse drag-drop operation.

## 1.1 Motivation and Problem Statement

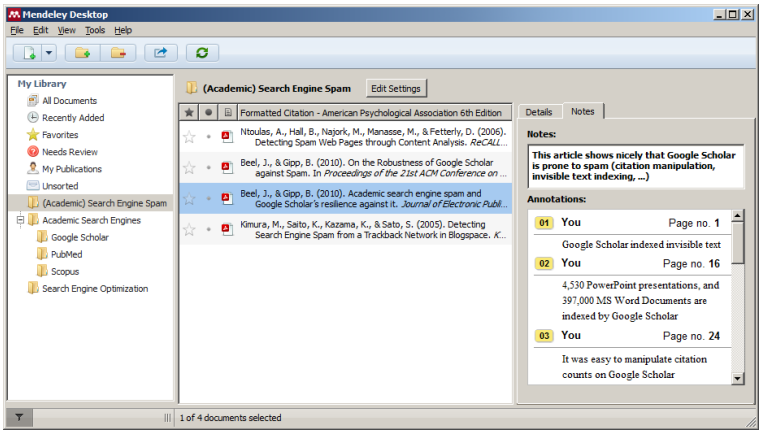



*Figure 1-1 Mendeley as an example of a classic three or four section user-interface for reference management*

**Use of folder, tag or keyword only is inadequate to keep dozen of electronic literature organized visually.**

Researchers who have reading and annotated large number of papers will easily lose track of what was written in which paper. However, the use of folder, tag or keyword to organize the references or PDFs could be ineffective when the user would like to search for particular item among dozens which listed in different categories. The user might need to remember the category, tag or keyword in order to find their interested item. Or the user would have to select each folder and scroll through all the items. It is not optimal because it takes a lot of time.

**Lack of single reference management system support drafting paper, creating in-text citation, keeping track document and ideas while reading PDFs.**

Research process could be a daunting undertaking and dominate life for months or years. Turning a collection of scholarly literature into final paper including keep track of related literature, take notes of interesting fact and idea being read, draft the paper, and also create bibliography. There are many system available to assist a researcher with keeping track of their bibliographic citations and organizing their thoughts. However, these programs are distinct pieces of system that do not always work well together. A visual diagram that grouping all information is required for paper-writing could help researcher to keep track their work along the research process.

## 1.2 Background Information

The rapid expansion of scientific literature has driven the development of reference management technique. There are many reasons to conduct literature reviews, including preparation of a journal article, completion of requirements for academic degrees, and description of the state of current knowledge. (Steele, 2008) When writing research literature reviews, the use of reference management system can be considered to promote accuracy in reference citations, manage large number of reference data and decrease time in formatting the reference to meet the style requirements.

In the old days, a common task to retrieve proper reference information and conform to the style for reference mandated by the journal manually is perceived to be time consuming and tedious by many researchers because the references were written on index cards and stored in boxes (Fenner, et al., n.d.). However, reference, technology, and instruction come together in the arena of research support.

The first RM system was developed in the 1980s and these system were initially for researchers to create online indexes of personal print-article collections (Lorenzetti & Ghali, 2013). The earliest system also able to manage the basic task of storing references and adding them to manuscripts. During the 1990s networked desktop access to major bibliographic databases became common and personal word processor packages were widely adopted. A large number of programs competed in this market but two programs became dominant – Endnote and Reference Manager. A couple of years back the programs like CiteULike, Connotea, Mendeley and Zotero are at the heart of this trend. They vary in functionality but are all free and have grown into the bibliographic world from the web world, whereas the more established products (Endnote, RefWorks) have gone the other way. In the last 15 years, there is a number of significant developments that have made author focus on building knowledge, rather than tedious job of managing reference (Fenner, et al., n.d.).

## 1.3 Project Scope

The proposed system is an offline desktop-based reference management system which will make user easy to search, organize and manage references, produce the bibliography and citation in the format required for a particular publication. Besides, this project explores the intriguing feature of providing single platform which allow user to draft and organize their thought and literature during research process.

Different from similar systems, the proposed system will be embedded with mind mapping tool and information extraction algorithm to empower the user during their research process. In this proposed system, a user is allowed to create a mind map to organize their ideas and thought without leaving the system. Beside, user can organize sentences from PDFs dedicated to the research in the mind map. Drag and drop interesting sentences from PDFs to mind map is supported while reading PDFs. The user able to link reference item to a node in mind map After drafting a mind map, it can be exported to text file or document file. A single mind map able to organize literature draft, bibliography citation, PDF document into a single interface which allow user to keep track the research process.

## 1.4 Project Objectives

There are the objectives of the project. These are:

- To analyse the current state of citation technology used during research process.
- To design a citation system to enhance research planning process by integrating mind mapping technique and text extraction for drafting research ideas.
- To evaluate the effectiveness of the mind-mapping as a tool in organizing research ideas.

## 1.5 Impact, Significance and Contribution

Different from similar system, this reference management system will blends mind-mapping tool, text extraction algorithm, PDF viewer and reference management into a single cohesive tool that can assists a researcher in almost every aspect of research process.

Despite there are many tools available to assist researcher in their research works, the separate and distinct pieces of program that do not always work well together remains a challenge problem to user's research planning and organizing process. Word processing system, PDF management system and reference management system are used separately to achieve the goal. This reference management system integrates these functions to seamless the work. By leveraging the power of mind mapping technique, this system empowers useful information structuring that is greater than most of the reference management system.

Smart Citation include several features to facilitate research process:

1. A reference management module for managing and creating citation and reference.
2. A PDF viewer for rendering and editing PDF document.
3. A mind mapping technique for drafting research ideas.
4. A text extraction algorithm is employed to facilitate retrieving text from PDF document.

## 1.6 Highlight of Achievement

The first achievement of this project is the function of generating citation and bibliography from data provided. The user able to create citation and bibliography from the data they have created and stored Smart Citation Manager. Other than generate output based on data stored in system, the user also able to create citation and bibliography instantly without storing the data. By having this function, the user is not necessary visit citation generator website or create data in advance to generate citation and bibliography just for one entry.

Next, Smart Citation Manager allows user to import PDF and organize it into the system. Unlike other similar systems, user able to render the imported PDF via the PDF viewer embedded in Smart Citation Manager. The PDF viewer provide editing, commenting, highlighting and other functions similar to Adobe Acrobat Reader which allow user to perform the actions within the system.

Moreover, Smart Citation Manager does provide mind mapping tool which allows user to draft their ideas for research. The interesting and relevant studies and ideas often result from large number of scholarly literatures. It is difficult to remember every interesting fact and ideas and their source. To overcome the problem, Smart Citation Manager provide a single interface that embeds PDF viewer and mind map panel which allow user to extract the sentences they interested from PDF and drag and drop it to mind map. The sentences from different source will be differentiated with difference background colour in mind map. After the user finish their draft, they able to export it in bulleted list to text file or document file. Besides that, the user able to know the source of ideas by creating a link to the reference item has been stored.

## 1.7 Report Organization

In Chapter 1, the general aspects of project will be discussed in details such as the project scope, objectives, significant and contribution of the project. Besides that, background information of the work, problem to be solved and motivation to pursuing this projects are described here.

In Chapter 2, the current and previous practice to solve the problems are described here. The similar systems will be researched and benchmarked in order to explore their strength and weakness in the particular area. By comparing the systems, the appropriate features that the system offered will be further improved and implemented into proposed system. Besides, solutions will be proposed in order to improve the weakness of current system.

In Chapter 3, the development of project is discussed in details. This chapter will include the key concept of techniques used, design specification and the several diagrams used to explain how the system is developed and work.

In Chapter 4, the process of developing project and the method used are briefly discussed here. Furthermore, the technology and tools used to develop the system such as programming language and software used. This chapter also provide instructions and requirement of the software installation. The issues faced during the project implementation and the timeline of developing the project are also included here.

In Chapter 5, this chapter mainly describes about the how the system to be tested and the test result produced.

Last but not least, Chapter 6 which is the last chapter that provide the summary of the overall project. Besides, the future work to enhance and improve the system will also be described here.

# CHAPTER 2 LITERATURE REVIEW

## 2.1 Literature Review / Related Work

### 2.1.1 System Review

The research landscape has changed tremendously since the first reference management (RM) system was introduced in the 1980s (Gilmour & Cobus-Kuo, 2011). The traditional RM system had the major features that we currently expect such as organizing references, creating bibliographies in various style and storing references. (Warling, 1992). However, in today world of open-source and advance technology, there is a corresponding emergence of RM system that are tailored to the different needs and expectations of users. Generally, most RM system carry out the same basic functions:

1. Insert citation in variety of styles
2. Creating bibliographies in variety of styles
3. Collect, organize, and annotate citations
4. Work with word processing software to facilitate in-text citation

The present study is to compare different prominent RM systems in term of superior features offered, their strength and weakness and possible algorithms can be deployed.

#### 2.1.1.1 EndNote


*Figure 2-1 Logo of EndNote software (Clarivate Analytics, 2017)*

**Introduction:** EndNote is system that installed on the personal computer. EndNote Web is also available as stand-alone system. EndNote can work compatible with Macs and Microsoft (Meredith, 2013). It is produced by Clarivate Analytics.

**Strength:** EndNote has the interesting feature that allow bibliography metadata to be extracted from PDF files after importing the PDF into its library (EndNote, n.d.). According to EndNote (n.d.), its feature 'cite while your write' allow user insert cited reference in the manuscript while simultaneously adding it to the bibliography list by just clicking on the endnote tool bar which integrated in word processing system.

**Weakness:** According to Hensley (n.d.), EndNote has no capability to work with PDF's within its environment although PDF metadata is extracted. PDF cannot be opened in EndNote application, highlighted and also annotated. Besides, although EndNote allow to access to references while writing in word processing system, but the user have to either highlight the reference in EndNote library first or search it in Endnote search bar, only then the relevant citations appear to be inserted.

### 2.1.1.2 Zotero

*Figure 2-2 Logo of Zotero software (Wikipedia, 2017)*

**Introduction:** Zotero is web-based and free open source reference management system which available for free to users around the world (Meredith, 2013). Zotero is a production of the Roy Rosenzweig Center for History and New Media, and was initially funded by the Andrew W. Mellon Foundation, the Institute of Museum and Library Services, and the Alfred P. Sloan Foundation. (Zotero, 2017)

**Strength:** Zotero allows extracting PDF metadata after importing PDF file to local library. Zotero's database has the features of robust folder system, each folder can contain many subfolders, and references can be dragged between folders or exist in more than one folder at a time. The organization of references can also rely on the tagging that allow the possibility of customized controlled classification. Moreover, Zotero allow a reference link to more than related reference and attach to URL or file in order to make the searching easy.

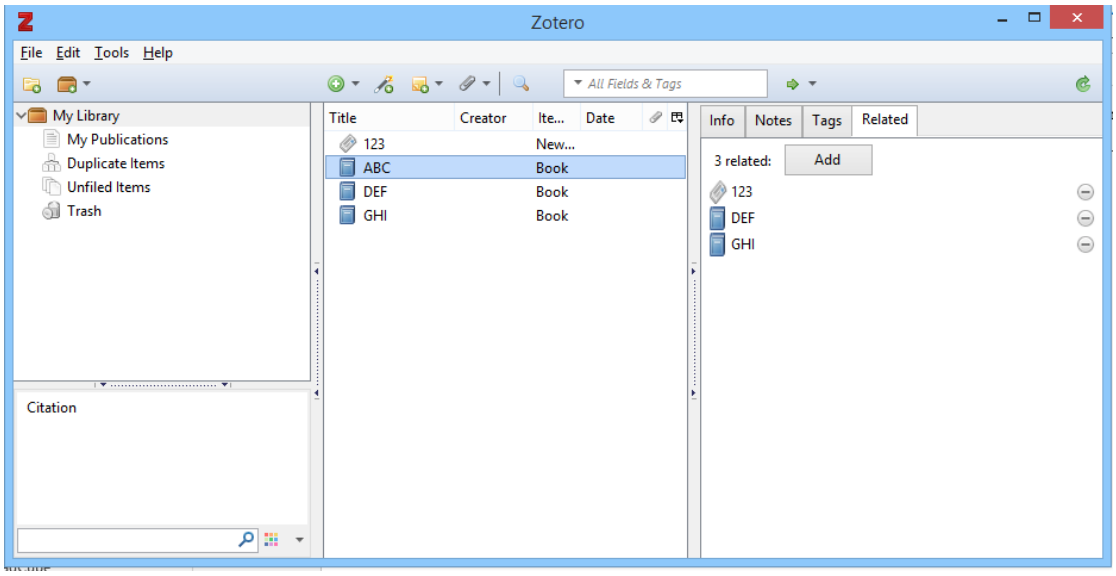



*Figure 2-3 Linking related references*

**Weakness:** Zotero cannot import PDF annotations to organize them in Zotero. Besides, Zotero only store the reference, not the entire PDF document. Moreover, Zotero do nothing on validating the data entered for each bibliography field and there is no hint for what format or type of data should be entered.

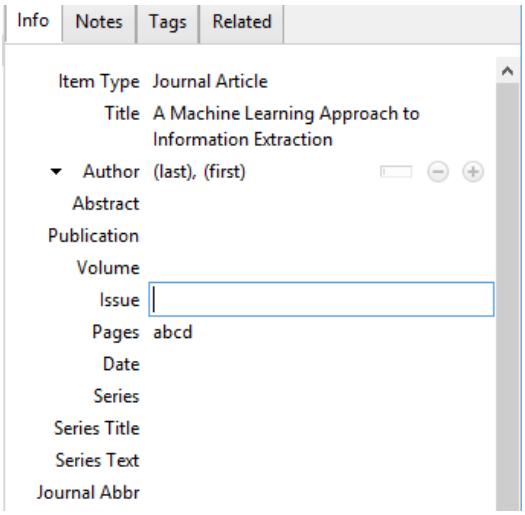



*Figure 2-4 Tab-based reference organization*

#### 2.1.1.3 Mendeley


*Figure 2-5 Logo of Mendeley software (Wikipedia, 2017)*

**Introduction:** Mendeley is a desktop and program produced by Elsevier for managing and sharing research papers. It combines Mendeley Desktop, a PDF and reference and reference management application (Fitzpatrick, 2009).

**Strength:** Mendeley incorporate PDF management and annotation features (Zaugg, et al., 2011) which allow importation of PDF metadata, automatic naming and filing of documents, opening of multiple PDF files within program that are navigable by tab achieved by dragging, and the ability to highlight and annotate PDF files within the application (Hensley, n.d.)

**Weakness:** No features for drafting documents, file management, or note taking.

#### 2.1.1.4 RefWorks


*Figure 2-6 Logo of RefWorks software (Wikipedia, 2017)*

**Introduction:** RefWorks is a web-based reference management system package and it is produced by RefWorks-COS, a business unit of ProQuest LLC. (LLC, 2008)

**Strength:** RefWorks allows the users to improve references from online databases and capture bibliographies information for webpages.

**Weakness**: RefWorks does not import metadata from the PDF while the PDF documents is uploaded. Moreover, it can only be used with web access

### 2.1.1.5 Docear

*Figure 2-7 Logo of Docear software (Docear, 2017)*

**Introduction:** Docear is a reference management system that integrates a PDF reader with Mind Mapping tool for those who want a visual way to keep their research organized (Kirby, 2017).

**Strength:** Mind Maps is the key to the unique approach for organization reference and PDFs in Docear (Docear, n.d.). Unlike options that display references as lists of citations or annotations, the mind maps allow the users to organize their literature. There are few killing features that distinguish Docear from all other: a) a single-section user interface that gives users a great overview of multiple PDFs and annotations in multiple categories at the same time b) users may sort single annotations independently from their parent PDFs which gives far more freedom in organizing information and c) Users can sort annotations within a PDF into categories which allows a far more detailed structure and overview (Docear, n.d.). Other than that, the highlighted sections and comments can be automatically extracted from the PDFs (Maps, 2012).

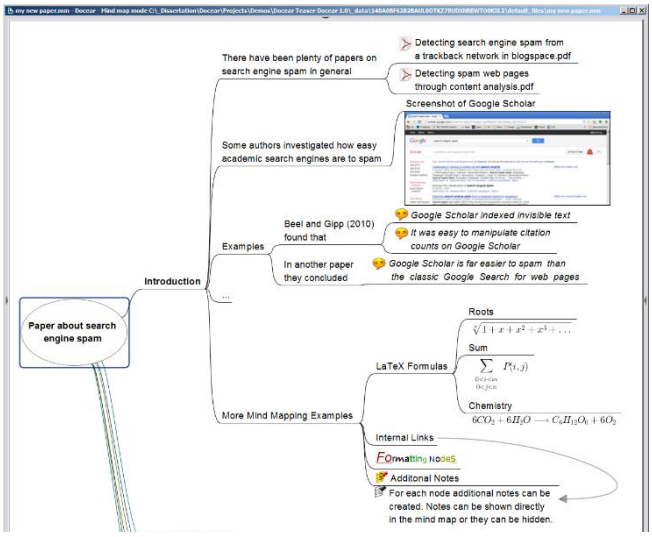



*Figure 2-8 Mind map in the Docear software (Docear, 2017)*

**Weakness**: A three (or four) section user-interface is what most RM system offer, and so does Docear. However the three section user-interface isn't neat and comfortable as other RM system.

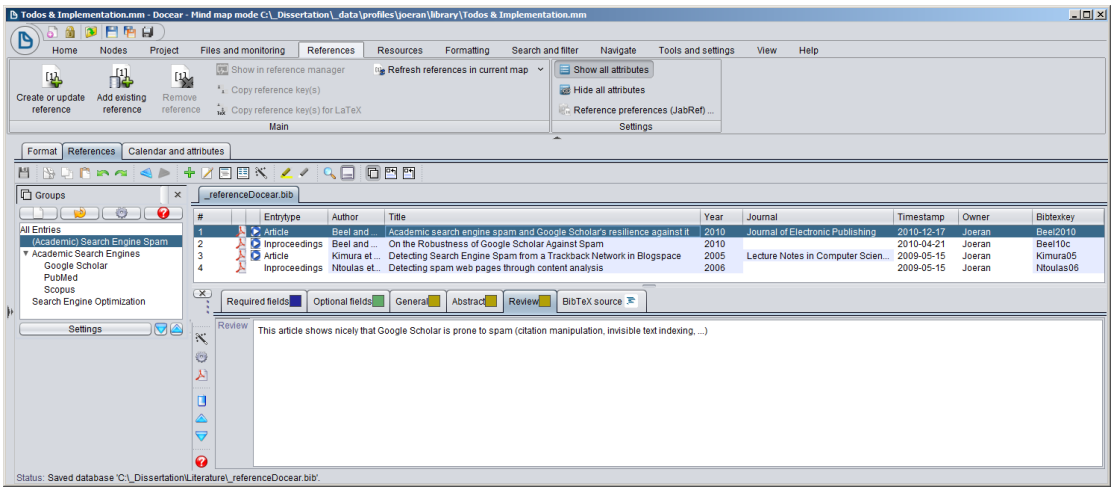



*Figure 2-9 Interface of Docear software (Docear, 2017)*

### 2.1.1.6 ReadCube


*Figure 2-10 Logo of ReadCube software*

**Introduction:** ReadCube is a desktop and browser-based program for managing, annotating, and accessing academic research articles (Wikipedia, 2017). ReadCube was created by Labtiva, a Boston-based company and a desktop client was publicly launched in October 2011. In November 2011, ReadCube Web Reader was integrated with the website of *Nature*.

**Strength:** ReadCube's Enhanced PDF viewer provides the ability to click in-line citations to immediately link to those references, click an author's name to browsehis/her other publications, see cited-by and altmetrics data for an article, make annotations and highlights, and browse figures.

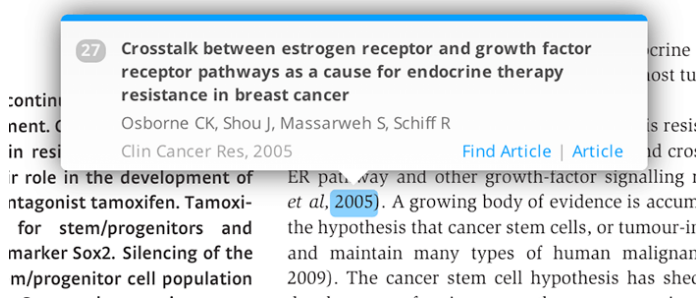



*Figure 2-11 Hyperlinked inline reference*

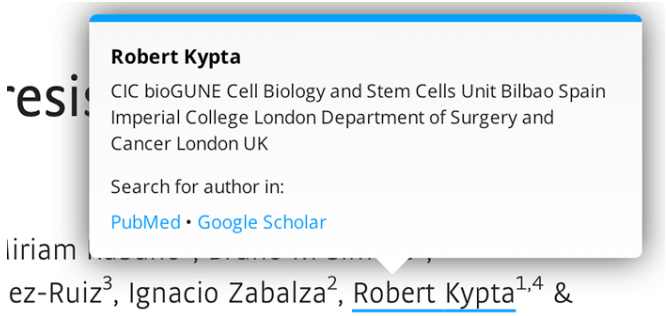



*Figure 2-12 Clickable author names*

**Weakness:** With respect to accessibility, ReadCube cannot be accessed without an internet connection, even when using the Desktop version. It prevents access to citations or articles while deprived of a decent internet connection (Academy, 2017)

### 2.1.2 PDF to Text Conversion

In order to allow user to easily retrieve the sentences they interested from PDF and organize in mind map, several PDF SDKs were studied. Adobe PDF Reader and Google PDFium provide the open-source software library that provide a control to render pdf on winform.

AxAcroPDF control from Adobe does render a pdf document in which it can allows the user perform the functions as same in Adobe Acrobat Reader such as viewing, commneting, highlighting and etc.

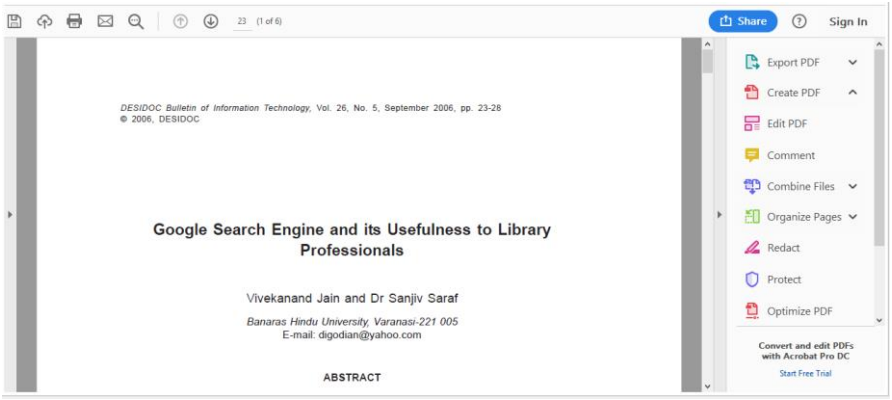



*Figure 2-13 PDF viewer from AxAcroPDF library*

PdfiumViewer from Google PDFium can render a pdf document and adds a toolbar with limited functionality.

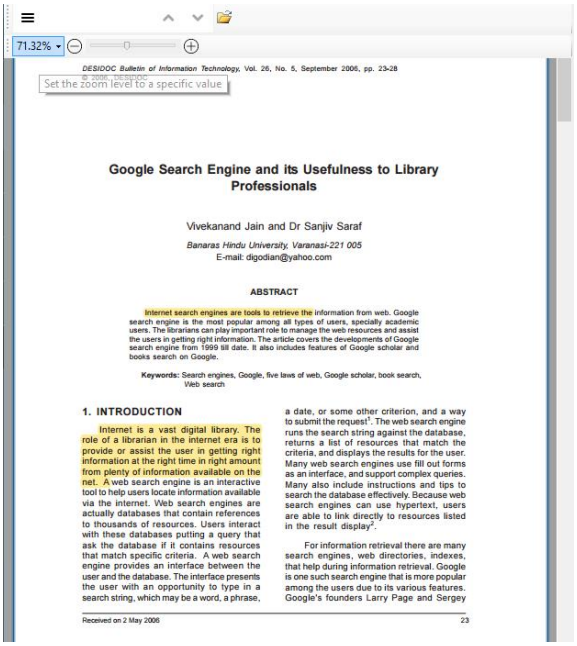



*Figure 2-14 PDF viewer from PDFium library*

| | AxAcroPDF | PdfiumViewer |
|---|---|---|
| Performance/ Processing Speed (viewing, printing, searching) | High | High |
| Rendering Capibilities (text, annotation, image, form) | Clear and Accurate | Clear and Accurate |
| Functionalities | Similar to Adobe Acrobat Reader | Limited |
| Function of detecting selected text? | No | Yes |

*Table 2-1 Comparison between AxAcroPDF and PDfium*

From the comparison table above, both library have the high performance and rich rendering capabilities. However, PdfiumViewer has limited functions compared to AxAcroPDF as it doesn't provide highlighting, commenting and other features. The significant limitation of AxAcroPDF is that it doesn't have the function of detecting the current selected text in order to be dragged to the mind map but PdfiumViewer does.

# CHAPTER 3 SYSTEM DESIGN

## 3.1 Key Concept

Smart Citation Manager will use the style written in Citation Style Language (CSL) for creating citation and bibliographies. CSL processor will be included to process data of an item to be cited which in the JSON format according the citation style which is in XML format.

### 3.1.1 Citation Style Language Overview

To generate references and citations in the desired style, description of each citation style in a language is required so that the computer can understand. Citation Style Language (CSL) is an open XML based language that describe the formatting of citations and bibliographies to simplify citing references (Anon., 2018). The following figure show how CSL works. (Zelle, 2016)

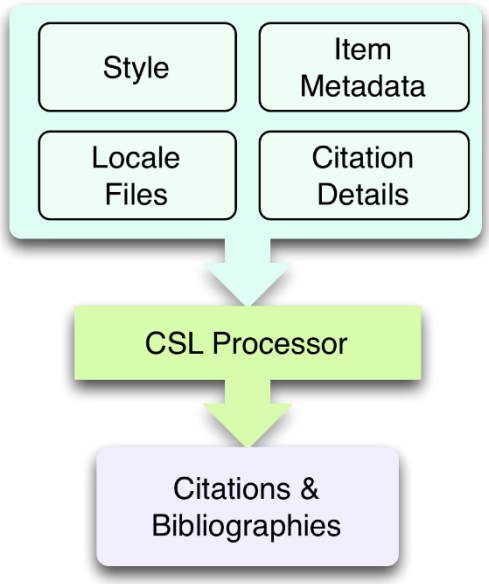



*Figure 3-1 CSL ecosystem*

### 3.1.1.1 Style (XML)

There are two types of CSL style which are independent styles and dependent styles. Independent CSL styles contain formatting instructions for citations and bibliographies while dependent CSL style only contain style metadata and simply refer to an independent CSL style to determine which citation format will be used so duplication of formatting instructions is not necessary. The styles are written in XML code as shown below.

```
<macro name="author">
  <names variable="author">
    <name and="text" delimiter-precedes-last="never" initialize-with=". " name-as-sort-order="all"/>
    <label form="short" prefix=" (" suffix=")"/>
    <et-al font-style="italic"/>
    <substitute>
      <names variable="editor"/>
      <names variable="translator"/>
      <choose>
        <if type="article-newspaper article-magazine" match="any">
          <text variable="container-title" text-case="title" font-style="italic"/>
        </if>
        <else>
          <text macro="title"/>
        </else>
      </choose>
    </substitute>
  </names>
</macro>
```

*Figure 3-2 Example of XML code in Style file* (Anon., 2017)

Style Structure

```
<?xml version="1.0" encoding="utf-8"?>
<style>
  <info/>
  <locale/>
  <macro/>
  <macro/>
  <citation/>
  <bibliography/>
</style>
```

*Figure 3-3 Style structure in XML format*

The figure above shown the basic structure of a style. The required <info> element stores the metadata about a style such as title of the style, author and contributor of the style and etc. The optional <locale> element is used to overwrite the locale data from

locales files. The optional <macro> elements are used to store CSL code that are used by <citation>, <bibliography> or other <macro> elements. The required <citation> and <bibliography> elements are used to define the format of citations and bibliography respectively.

```
<info>
  <title>Example Style</title>
  <id>http://www.zotero.org/styles/example</id>
  <link href="http://www.zotero.org/styles/example" rel="self"/>
  <link href="http://www.zotero.org/styles/apa" rel="template"/>
  <link href="http://www.example.com/style-guide/" rel="documentation"/>
  <author>
    <name>John Doe</name>
    <email>JohnDoe@example.com</email>
  </author>
  <contributor>
    <name>Jane Doe</name>
  </contributor>
  <contributor>
    <name>Bill Johnson</name>
  </contributor>
  <category citation-format="author-date"/>
  <category field="science">
  <updated>2014-10-15T18:17:09+00:00</updated>
  <rights license="http://creativecommons.org/licenses/by-sa/3.0/">This work is li
</info>
<locale xml:lang="en">
  <terms>
    <term name="no date">without date</term>
  </terms>
</locale>
```

*Figure 3-4 Example of XML code (part 1) in Style file* (Rintzr M, n.d.)

The <locale> elements is to overwrite the default translation as shown above. “without date” will be replace the default “no date” term in CSL locale file. The citation of a cited item without an issued date will be look like “(T. Yun, without date)” instead of “(T, Yun, no date)”.

```
<macro name="author">
  <names variable="author">
    <name initialize-with="."/>
  </names>
</macro>
<macro name="issued-year">
  <choose>
    <if variable="issued">
      <date variable="issued">
        <date-part name="year"/>
      </date>
    </if>
    <else>
      <text term="no date"/>
    </else>
  </choose>
</macro>
```

*Figure 3-5 Example of XML code (part 2) in Style file* (Rintzr M, n.d.)

The <macro> element are used to describe how the variable of cited item will be printed. The example above will print the author name in which the first name will be appear as intials followed by the ".". For the "issued-year" macro element, the year will be printed if the cited item has a date stored in "issued" date variable otherwise "no date" term will be printed.

```
<citation et-al-min="3" et-al-use-first="1">
  <sort>
    <key macro="author"/>
    <key macro="issued-year"/>
  </sort>
  <layout prefix="(" suffix=")" delimiter="; ">
    <group delimiter=", ">
      <text macro="author"/>
      <text macro="issued-year"/>
    </group>
  </layout>
</citation>
```

*Figure 3-6 Example of XML code (part 3) in Style file* (Rintzr M, n.d.)

The "et-al-min" and "et-al-use-first" describe the how the author name will be printed in citation. The example above specify that only the first name is printed, followed by the "et-al" term if an item has three or more authors. The <layout> element used to describe the format of each individual cite. The "delimiter" will separate the neighbouring cites with ";" whereas the <group delimiter> will separate the <text macro> element with ",".

```
<bibliography>
  <sort>
    <key macro="author"/>
    <key macro="issued-year"/>
    <key variable="title"/>
  </sort>
  <layout suffix="." delimiter=", ">
    <group delimiter=". ">
      <text macro="author"/>
      <text macro="issued-year"/>
      <text variable="title"/>
      <text variable="container-title"/>
    </group>
    <group>
      <text variable="volume"/>
      <text variable="issue" prefix="(" suffix=")"/>
    </group>
    <text variable="page"/>
  </layout>
</bibliography>
```

*Figure 3-7 Example of XML code (part 4) in Style file* (Rintzr M, n.d.)

The structure of <bibliography> element is similar to <citation> element but the <layout> is to define the format of individual bibliographic entry.

### 3.1.1.2 Locale Files (XML)

With locale file, the CSL styles can easily switch between different languages as it contain a set of localization data such as translations of common terms, localized date format and grammar option to support particular language dialect.

```
<?xml version="1.0" encoding="utf-8"?>
<locale xmlns="http://purl.org/net/xbiblio/csl" version="1.0" xml:lang="ko-KR">
  <info>
    <rights license="http://creativecommons.org/licenses/by-sa/3.0/">This work is licensed under a Creative Commons Attribution-ShareAlike
    <updated>2012-07-04T23:31:02+00:00</updated>
  </info>
  <style-options punctuation-in-quote="false"/>
  <date form="text">
    <date-part name="year" suffix="년"/>
    <date-part name="month" form="numeric" prefix=" " suffix="월"/>
    <date-part name="day" prefix=" " suffix="일"/>
  </date>
  <date form="numeric">
    <date-part name="year"/>
    <date-part name="month" form="numeric-leading-zeros" prefix="/"/>
    <date-part name="day" form="numeric-leading-zeros" prefix="/"/>
  </date>
  <terms>
    <term name="accessed">접근된</term>
    <term name="and">와/과</term>
    <term name="and others">and others</term>
    <term name="anonymous">anonymous</term>
    <term name="anonymous" form="short">anon</term>
    <term name="at">at</term>
    <term name="available at">available at</term>
    <term name="by">by</term>
    <term name="circa">circa</term>
    <term name="circa" form="short">c.</term>
    <term name="cited">cited</term>
    <term name="edition">
      <single>edition</single>
      <multiple>editions</multiple>
    </term>
```

*Figure 3-8 Example of XML code in Locale file* (Anon., 2015)

### 3.1.1.3 Item Metadata (JSON)

Item metadata is the bibliographic details of the item to be cited such as title, author name, date, and etc. It is stored in JSON-based format that can be understood by the CSL processor. The variable of the cited item can be refer to the Appendix.

```
{
    "id": "2656243/NWFAMK9T",
    "type": "book",
    "title": "Global Literary Theory",
    "publisher": "Routledge",
    "author": [
        {
            "family": "Lane",
            "given": "Richard J."
        }
    ]
},
```

*Figure 3-9 Example of item metadata in JSON format*

### 3.1.1.4 Citation Details (XML)

Citation detail contain context-specific information for specifying the way citation and bibliographies looks depend on the context in which the items are cited.

### 3.1.1.5 CSL Processors

CSL processor parse all the information (CSL styles, local files, item metadata, and citation details) and generate citations and bibliographies in correct format.

### 3.1.2 Citation Data Item

Each citation data items must be encoded in proper CSL-JSON format which is important to getting correct results from the CSL Processor. Each item consists of various field types and packaged into a container as a unit of citations.

```
"type": "book",
"title": "Science Fiction",
"author": [
    {
        "family": "Roberts",
        "given": "Adam"
    }
],
"accessed": {
    "raw": "2016-04-07"
}
```

*Figure 3-10 Example of item metadata in JSON format*

#### 3.1.2.1 Field Types

- *Type* Field

  The type field is a required and simple field containing string value. The values of the type field are constrained to a limited set of possible values which must be a valid CSL type. See the type in Appendix A.

- *Ordinary* Field

  In ordinary fields, it can contain a string or numeric value and only limited set of fields are recognized by the processor. Unrecognized fields will be ignored. For the fields available on each item type, see the listing of variable in Appendix B.

- *Name* Field

  A name field is an array of objects that lists authors, contributors, or creators, etc. There are two properties in name field: “family” and “given”. The “family” represents the person’s familial name and “given” represents the person’s name has been given. See the name field in Appendix C.

- *Date* Field

  Date field express in two different formats: array and raw string. Array format basically used to express date range while raw string represents the date in YYYY-MM-DD format. See the date field in Appendix D.

### 3.1.3 Mind Mapping technique

The mind map feature that provided in Smart Citation Manager is developed by several classes in C# language: GraphNode class, GraphPanel, and MindMap class. The GraphNode class basically describes the properties such as list of the nodes which connecting to its, and the connecting line and the functions to make connections with other nodes. GraphPanel class acts as a Panel which allow the nodes to be added on it. It contains a list of nodes which have been added, and the functions such as adding, editing and deleting node, moving node, drawing line between node and e.t. However, MindMap class is the mind map interface for the user to perform their task. In this class, the menu bar and contextual menu item events are defined in this class in order to trigger the relevant functions.

#### 3.1.3.1 GraphNode

```
namespace Citation_Manager
{
    [Serializable()]
    public class GraphNode : Label, ISerislizable
    {
        public List<Point> Edge = new List<Point>();
        public List<GraphNode> Node = new List<GraphNode>();
        public Point loc;
        public string link;
        public string item;
```

*Figure 3-11 Part of codes in GraphNode class (Part 1)*

The figure above show the part of codes in GraphNode class. A GraphNode instance will contains a list of connected nodes, the connection line positions, its current location, file path linked and the associated reference item.

```
//Create new connection
//p2 - point on graph
public void AddConnection(Point p2, bool isYes, GraphNode graph)...

//method which Draw Edges of current node,
//color and LineWidth
public void DrawEdges(Graphics g, Color color, int LineWidth)...

/// Method which draw edges of current node
public void DrawEdge(Graphics g, Color c, int LineWidth, bool isYes)...

/// Draw edge (Line) which connect two nodes, using 2 points
public void DrawEdge(Graphics g, Color c, int LineWidth, Point p1, Point p2, int ind)...
```

*Figure 3-12 Part of codes in GraphNode class (Part 2)*

The part of codes shown as above are to make a connection between nodes and store the locations of the line.

```
#region Serialization
//Simple Deserialization use only fields which we need
protected GraphNode(SerializationInfo info, StreamingContext context) : this()
{
    Type pGraphNode = typeof(GraphNode);
    typeof(Label).GetProperty("Name").SetValue(this, info.GetValue("Name", this.Name.GetType()), null);
    typeof(Label).GetProperty("Location").SetValue(this, info.GetValue("Location", this.Location.GetType()), null);
    typeof(Label).GetProperty("Text").SetValue(this, info.GetValue("Text", this.Text.GetType()), null);
    typeof(Label).GetProperty("ForeColor").SetValue(this, info.GetValue("ForeColor", this.ForeColor.GetType()), null);
    typeof(Label).GetProperty("BackColor").SetValue(this, info.GetValue("BackColor", this.BackColor.GetType()), null);
    typeof(Label).GetProperty("Font").SetValue(this, info.GetValue("Font", this.Font.GetType()), null);
    pGraphNode.GetField("link").SetValue(this, info.GetValue("link", typeof(string)));
    pGraphNode.GetField("item").SetValue(this, info.GetValue("item", typeof(string)));
    pGraphNode.GetField("Node").SetValue(this, info.GetValue("Node", typeof(List<GraphNode>)));
    pGraphNode.GetField("Edge").SetValue(this, info.GetValue("Edge", typeof(List<Point>)));
}
//Simple Serialization use only fields which we need
public virtual void GetObjectData(SerializationInfo info, StreamingContext context)
{
    info.AddValue("Name", this.Name);
    info.AddValue("Location", this.Location);
    info.AddValue("Width", this.Width);
    info.AddValue("Text", this.Text);
    info.AddValue("ForeColor", this.ForeColor);
    info.AddValue("BackColor", this.BackColor);
    info.AddValue("Font", this.Font);
    info.AddValue("link", this.link);
    info.AddValue("item", this.item);
    info.AddValue("Node", this.Node, typeof(List<GraphNode>));
    info.AddValue("Edge", this.Edge, typeof(List<Point>));
}
#endregion
```

*Figure 3-13 Part of codes in GraphNode class (Part 3)*

In order to save the mind map progress, each of the nodes has been created must be serialized into a stream of bytes to store the object to memory.

### 3.1.3.2 GraphPanel

```
// Draw vector from point1 to point2, using absolute values
private void DrawEdge(Color color, Point p1, Point p2)...

/// Draw all edges of the graph in the cicle
private void DrawAllEdges()...

//link to item
public void setItem(GraphNode graphNode,string item)...

//set node text color
public void setColor(GraphNode graphNode)...

//set node back color
public void setBackColor(GraphNode graphNode)...

//set node text font
public void setFont(GraphNode graphNode)...
```

*Figure 3-14 Part of codes in GraphPanel class (Part 1)*

The part of codes above show the functions that draw black line between connected nodes, add reference to a reference item and change the font and color of the node.

```
//get the node to export
public List<string> getGraphNode()...

/// Add node at the point
public GraphNode AddNode(Point location, bool drag, string text)...

/// Delets graph node and all incoming and outcoming connections (edges)
public void DeleteNode(GraphNode curG)...

//Get current selected edge
public SelectedGraphEdge GetSelectedEdge()...

//Delete current selected edge
public void DeleteEdge(SelectedGraphEdge currentgraph)...
```

*Figure 3-15 Part of codes in GraphPanel class (Part 2)*

The functions above are mainly used to get the nodes, create and delete node.

```
/// Mouse Down event for GraphNode control
private void graphNode_MouseDown(object sender, MouseEventArgs e)...

/// Mouse move event for GraphNode control
private void graphNode_MouseMove(object sender, MouseEventArgs e)...

/// Mouse up event for the GraphNode control
private void graphNode_MouseUp(object sender, MouseEventArgs e)...

//show file in location
public void graphNode_MouseDoubleClick(object sender, MouseEventArgs e)...

//text change event for GraphNode control
public void graphNode_TextChange(object sender, EventArgs e)...

//Save graph on the file
public void SaveGraphToFile(String fileName)...

//Load Graph From File
public void OpenGraphFromFile(String fileName)...

//create new graph
public void CreateNewGraph()...

/// Override on mouse event
protected override void OnMouseMove(MouseEventArgs e)...
```

*Figure 3-16 Part of codes in GraphPanel class (Part 2)*

The functions above are declared to handle the mouse events triggered by the user. Besides that, functions to create new mind map, save and load a mind map are also declared in the class.

### 3.1.3.3 MindMap

```
private void pnGraph_MouseClick(object sender, MouseEventArgs e)...
private void insertCitationToolStripMenuItem_Click(object sender, EventArgs e)...
private void colorToolStripMenuItem_Click(object sender, EventArgs e)...
private void backColorToolStripMenuItem_Click(object sender, EventArgs e)...
private void fontToolStripMenuItem_Click_1(object sender, EventArgs e)...
private void deleteToolStripMenuItem_Click(object sender, EventArgs e)...
private void addToolStripMenuItem1_Click_1(object sender, EventArgs e)...
private void showItemToolStripMenuItem_Click(object sender, EventArgs e)...
private void pnGraph_DragDrop_1(object sender, DragEventArgs e)...
private void pnGraph_DragEnter(object sender, DragEventArgs e)...
private void pdfViewer1_MouseDown_1(object sender, MouseEventArgs e)...
private void newToolStripMenuItem_Click(object sender, EventArgs e)...
private void saveToolStripMenuItem1_Click(object sender, EventArgs e)...
private void loadToolStripMenuItem_Click(object sender, EventArgs e)...
private void deleteToolStripMenuItem1_Click(object sender, EventArgs e)...
private void helpToolStripMenuItem_Click(object sender, EventArgs e)...
private void exportToTextFileToolStripMenuItem_Click(object sender, EventArgs e)...
```

*Figure 3-17 Part of codes in MindMap class*

The functions declared in MindMap class mainly used to handle the events triggered from menu item. MindMap class also handle the drag and drop function for extracting text from PDF.

### 3.1.4 Text Extraction

Text extraction technique allows the user extract the sentences from PDF and add to mind map seamlessly. This technique is supported by PDF library imported which provide viewer control to render the PDF document.

After user open the PDF via the PDF viewer inside Smart Citation Manager, they able to select sentences or paragraphs using mouse cursor and drag them directly to mind map panel. A node with the text selected will be automatically added into mind map. Smart Citation Manager will first detect any text is selected inside the PDF viewer. Once the user have select certain text, an event will be triggered to get the text. Whenever user mouse click the pdf viewer, activated mouse click event will check which mouse button has been clicked and the object. If the particular text is selected and dragged, a drag event will be triggered to copy the text. Drop event will be fired

in order to add a new node with the dragged text when the text is drop to mind map panel.

### 3.1.5 Generate Citation and Bibliography

There are several input required by the CSL Processor to produce citation and reference in selected style. The figure below show the steps that how CSL processor process data and style then generate citation and reference.

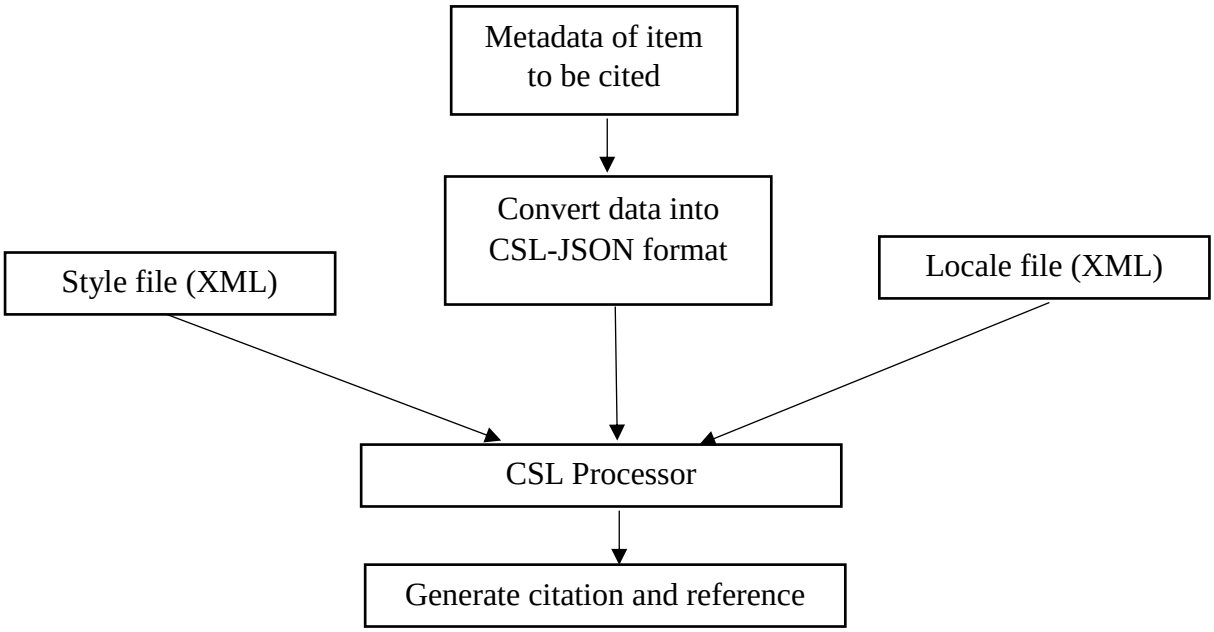



*Figure 3-18 Overview process of citation and reference generation*

Firstly, the processor will load and compile the style file. After that, the data in JSON format will be parsed to CSL-JSON which can be processed by the processor. The processor will generate a list of citation and reference based on the JSON-based data.

## 3.2 Design Specifications

### 3.2.1 General Overview

This design overview is to introduce and give a brief overview of the system design. Use case diagrams, flow chart diagram, use case description, and window navigation diagram are used to explain the functions and work flow of Smart Citation Manager.

Smart Citation System provide the functions of managing and generating citation and bibliography and also differs in the features offered among the reference management system. It implements the mind mapping technique, and text extraction. These features advancements will be accomplished by developing following components:

- Reference Management
- Mind Map
- PDF viewer
- Database

Smart Citation Manager is a reference management system which require to store the metadata of item and the pdf document so database is essential for it. Therefore, ERD model will be included to show the relationship of all entity used by Smart Citation Manager.

### 3.2.2 Design Consideration

#### 3.2.2.1 Assumptions

The user of the reference management system is aware of basic of a computer. The user also understand the standard terms for using the system.

#### 3.2.2.2 System Development

The system is accessible through any laptop and desktop with Microsoft Window OS, without any internet connection required. The system will be developed in Microsoft Visual Studio 2015 using C#.net. Third-party C# libraries are used for processing citation and reference, controlling PDF viewer, extracting text from PDF and exporting mind map to word document.

### 3.2.3 Development Tools

- Visual Studio 2015
- SQL Database Server
- C# language
- Third-party libraries.

## 3.3 Architecture

### 3.3.1 System Design

The block diagram below shows the principal parts of the system and their interactions.

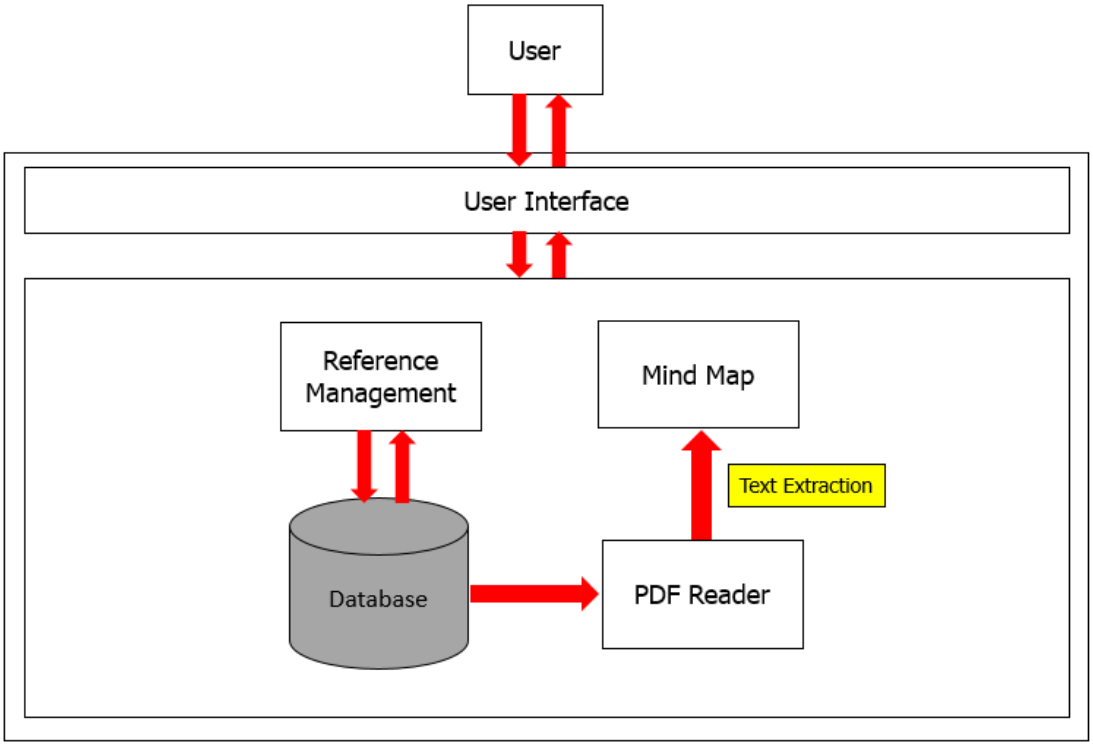



*Figure 3-19 System design block of Smart Citation Manager*

Figure above depicts the block diagram of the system. The system will be constructed from multiple components.

- User Interface – The user interface for user interacting with the whole system functionalities.
- Mind Map – It is an environment for drafting ideas with the support of text extraction from PDF document.
- Reference Management – A module for collect, store and organize references, generate references in appropriately formatted bibliographic style.
- PDF Reader – An interface for viewing and editing the PDF files.
- Text Extraction – An algorithm for extracting text from PDF file.
- Database – Data storage for storing, importing, exporting raw collected data required for the system to work.

### 3.3.2 Data Design

#### 3.3.2.1 Data Description

Microsoft SQL Server is used to communicate with the database that is installed locally on the server. The data will reside locally on the user device. The data created will exist for the time after the application is installed and will subsequently be destroyed when it is uninstalled.

#### 3.3.2.2 Tables

FileTable: Store the PDF documents.

ItemTable: Store the all items created.

Book: Store the information of book items.

Website: Store the information of website items.

JournalArticle: Store the information of journal articles items.

#### 3.3.2.3 Entity Relationship Diagram (ERD)

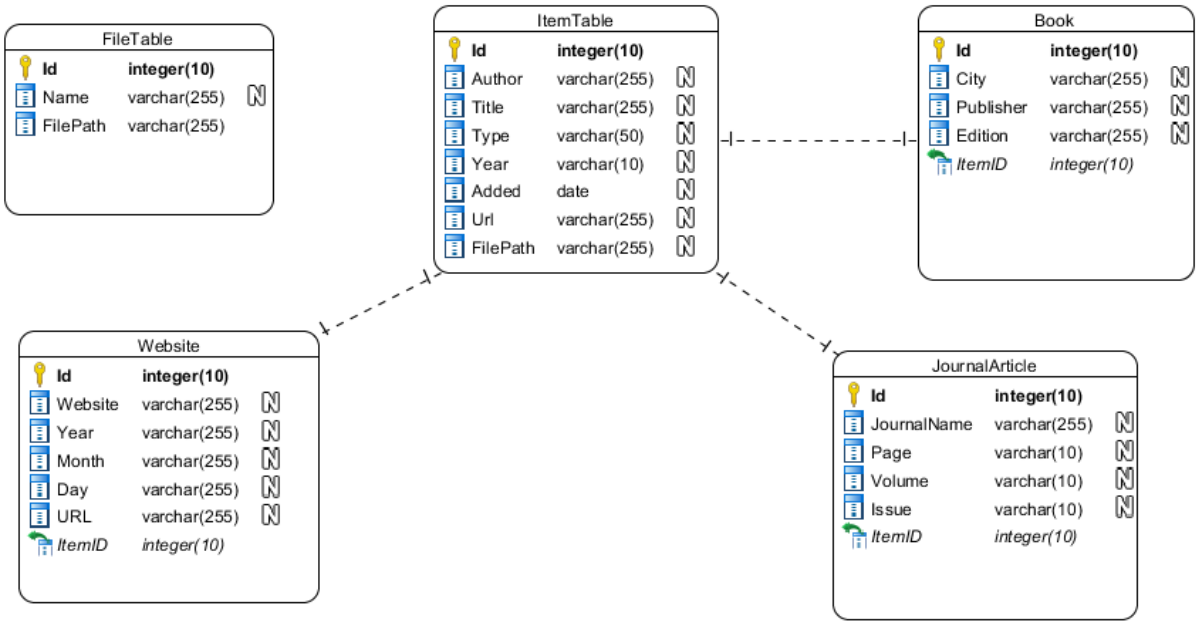



*Figure 3-20 Entity Relationship Diagram of database in Smart Citation Manager*

### 3.3.3 Component Design

#### 3.3.3.1 Use Case Diagram

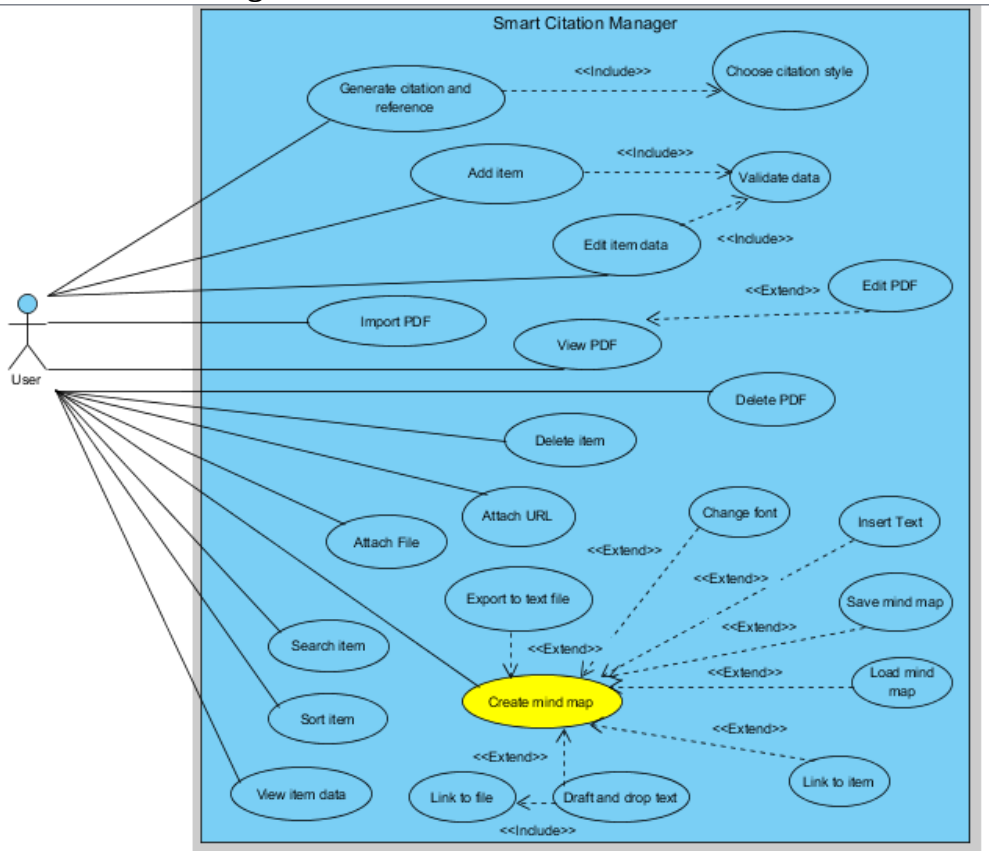



*Figure 3-21 Use Case Diagram of Smart Citation Manager*

### 3.3.3.2 System Flow Chart

a. Add Item

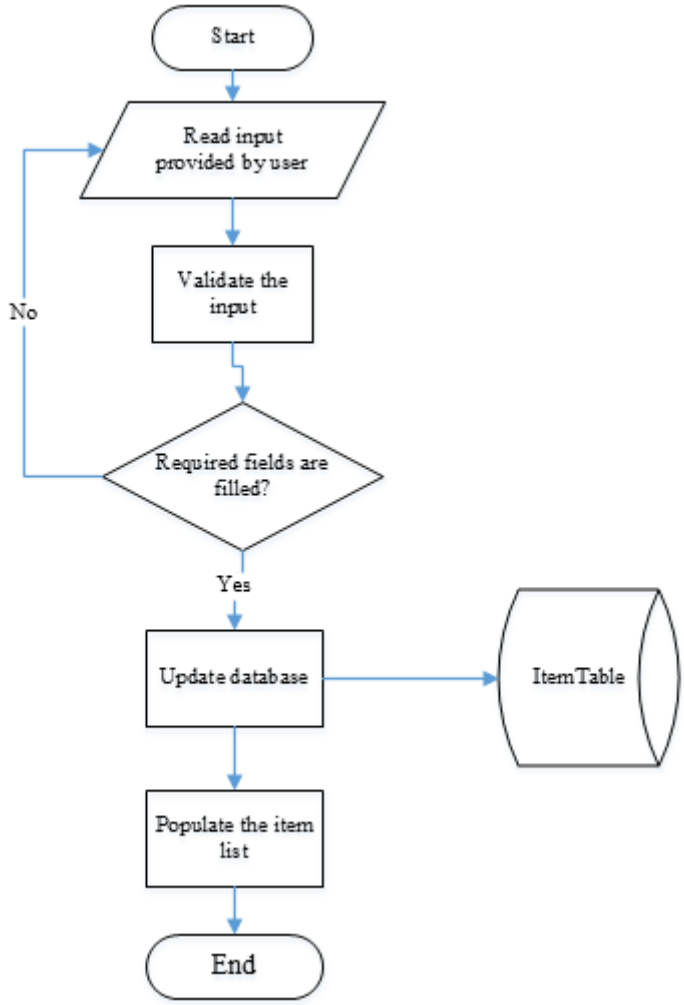



*Figure 3-22 Flowchart Diagram of Adding Item*

b. Edit Item

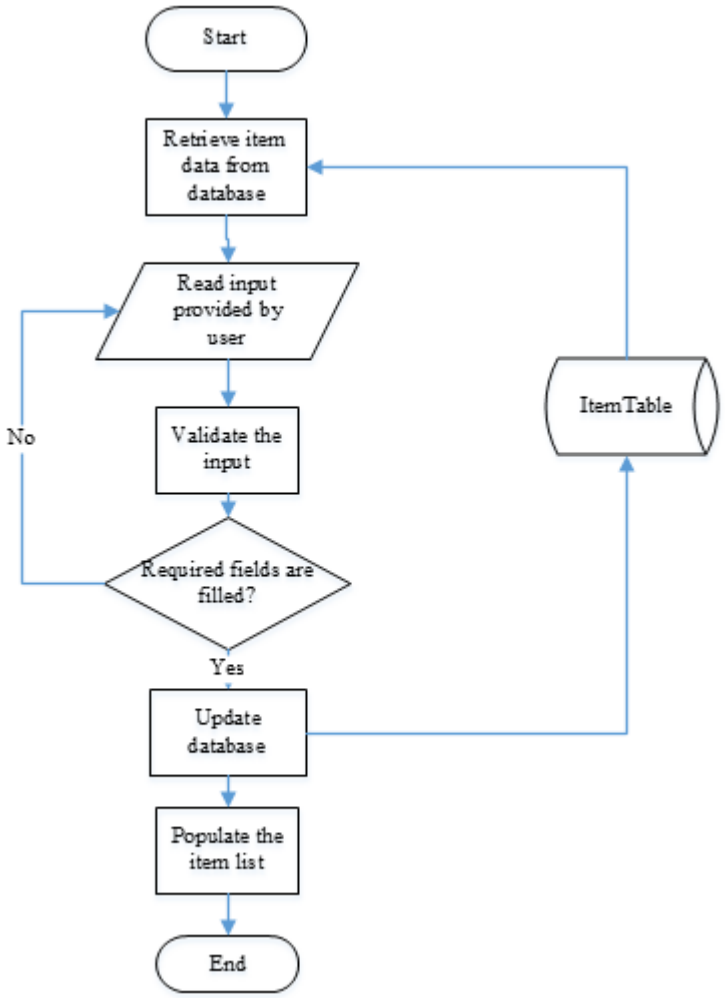



*Figure 3-23 Flowchart Diagram of Editing Item*

c. Search Item

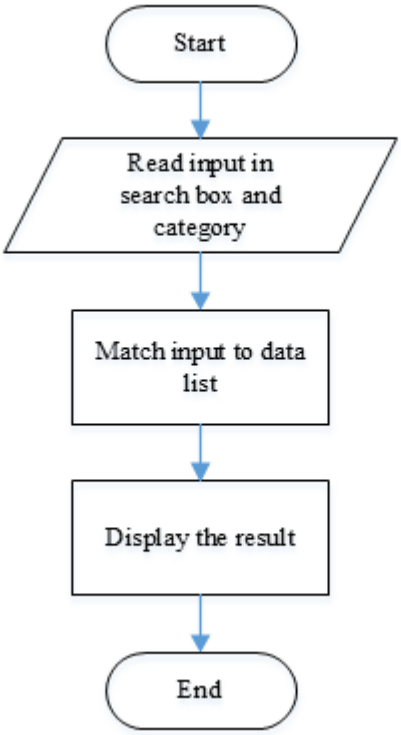



*Figure 3-24 Flowchart Diagram of Searching Item*

d. Sort Item

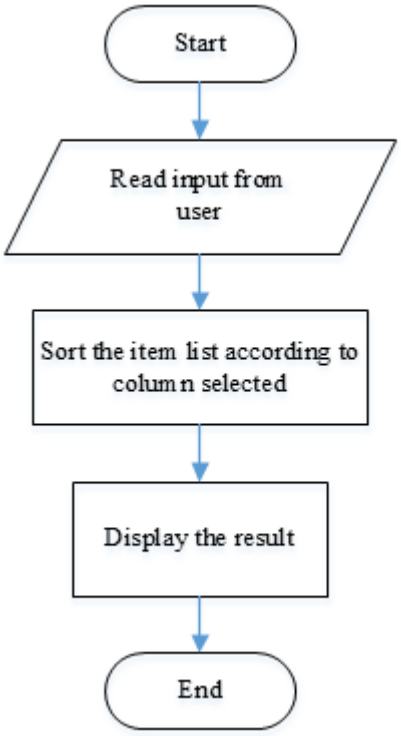



*Figure 3-25 Flowchart Diagram of Sorting Item*

e. Delete Item

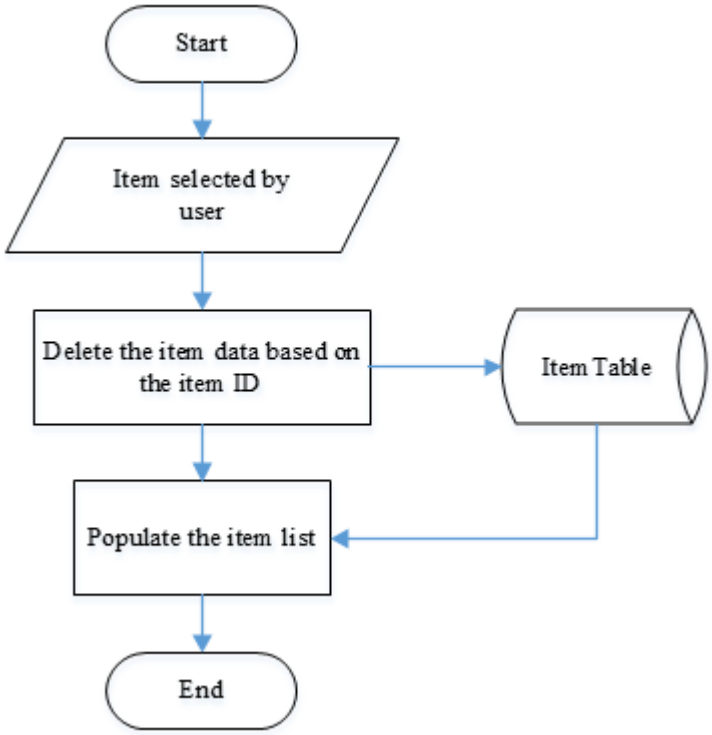



*Figure 3-26 Flowchart Diagram of Deleting Item*

f. Import PDF

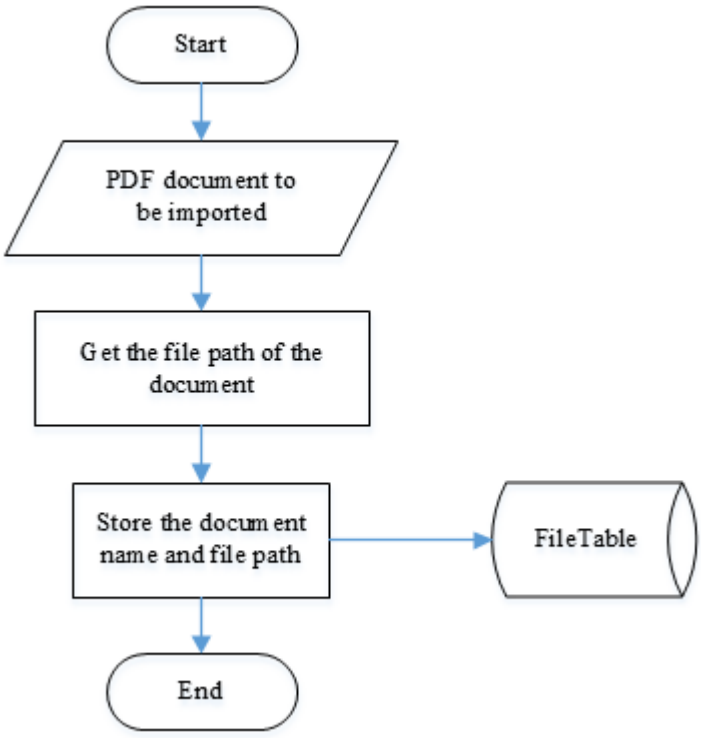



*Figure 3-27 Flowchart Diagram of Importing PDF*

g. View PDF

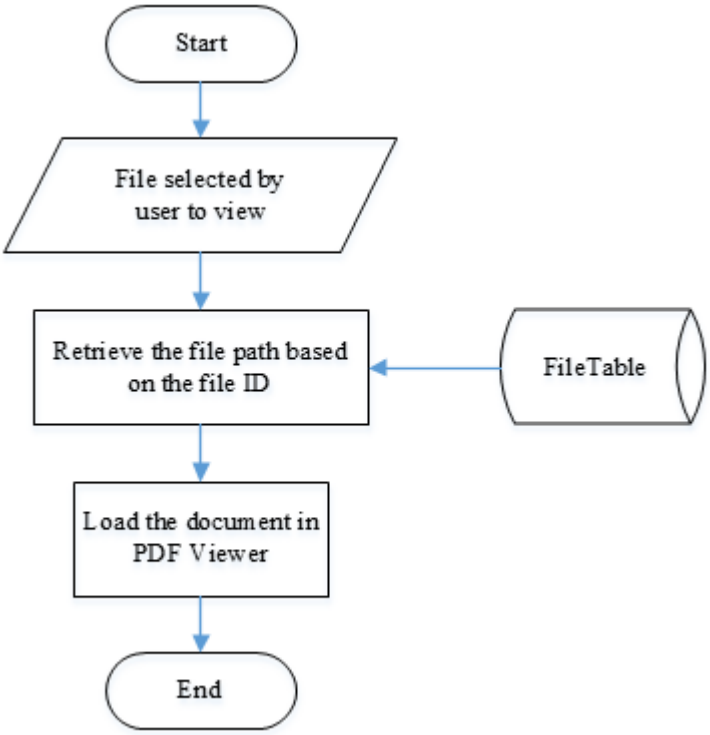



*Figure 3-28 Flowchart Diagram of Viewing PDF*

h. Delete PDF

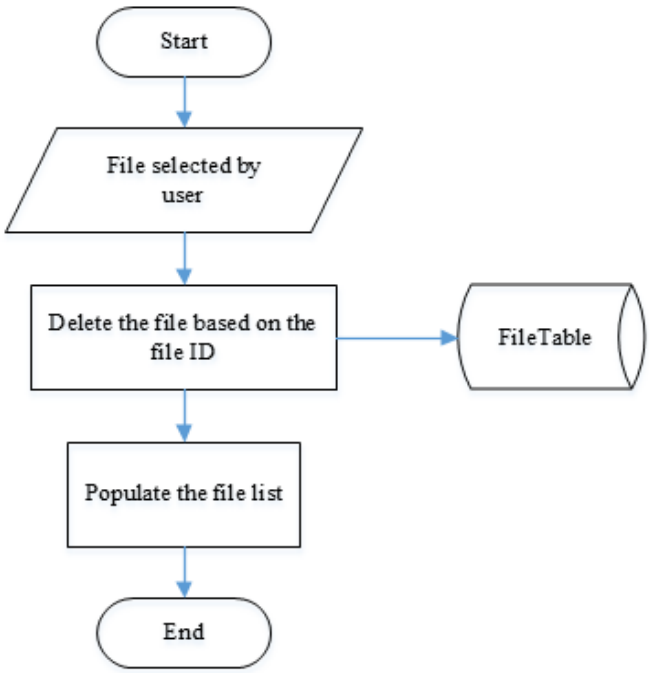



*Figure 3-29 Flowchart Diagram of Deleting PDF*

i. Generate Citation and Reference

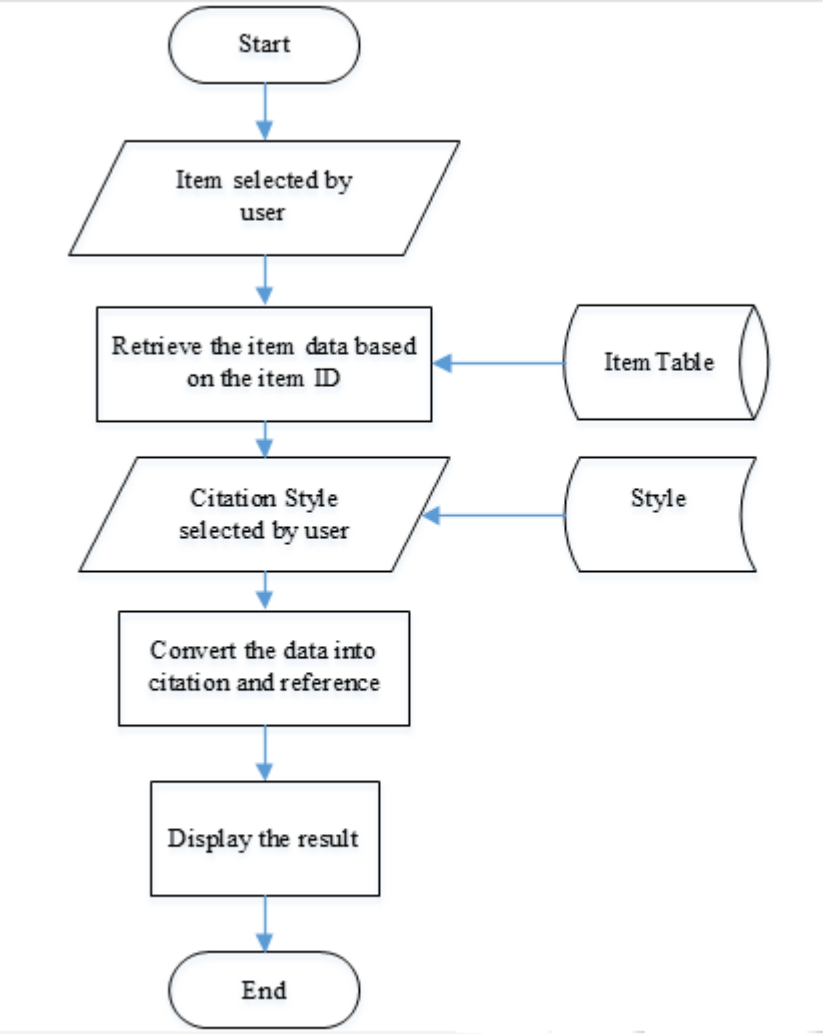



*Figure 3-30 Flowchart Diagram of Generating Citation and Reference*

j. Attach URI

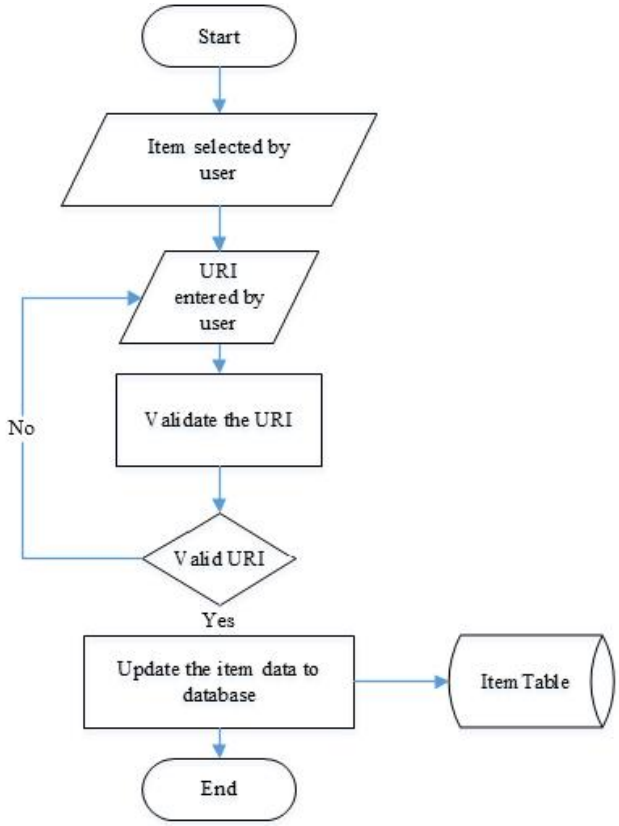



*Figure 3-31 Flowchart Diagram of Attaching URI*

k. Attach File

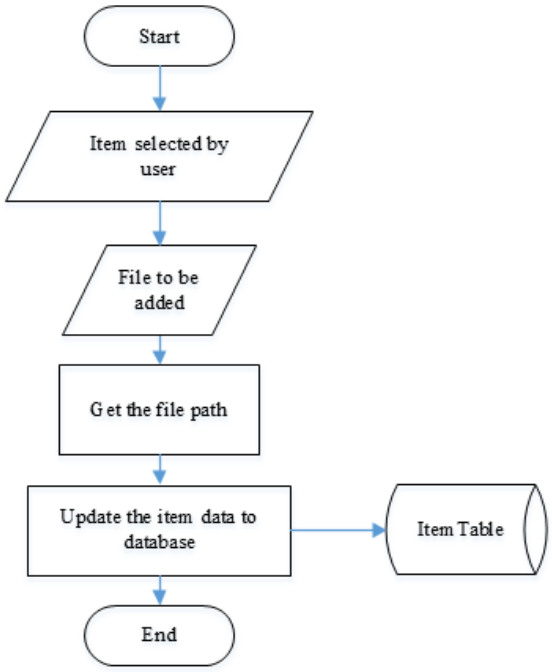



*Figure 3-32 Flowchart Diagram of Attaching File*

l. Create Mind Map

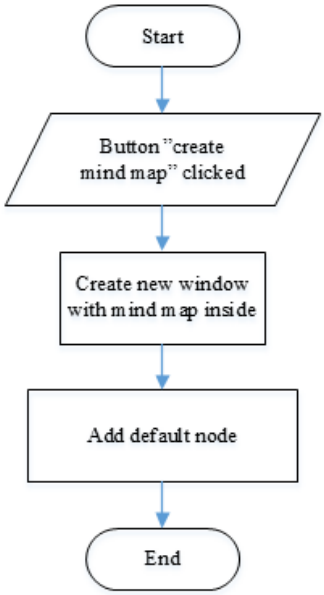



*Figure 3-33 Flowchart Diagram of Creating Mind Map*

m. Drag and Drop Text

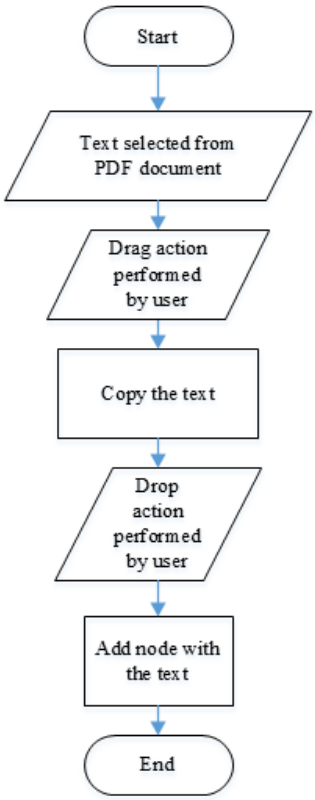



*Figure 3-34 Flowchart Diagram of Drag and Drop Text*

n. Export to File

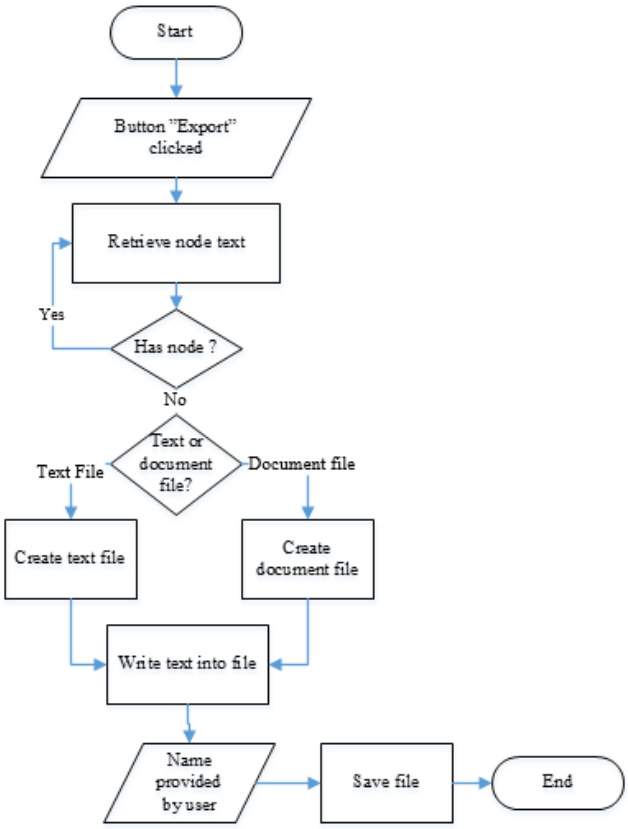



*Figure 3-35 Flowchart Diagram of Exporting to File*

l. Link to Item

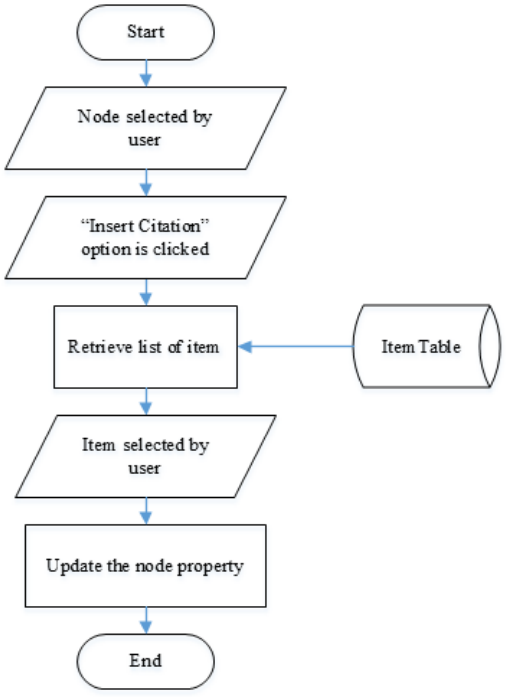



*Figure 3-36 Flowchart Diagram of Linking to Item*

### 3.3.3.3 Use Case Description

| Use Case Name: Add Item | ID :1 | Importance Level: High |
|---|---|---|
| Primary Actor: User | | |
| Summary Description:<br>User want to add an item in order to be stored and used in future | | |
| Relationship:<br>Association: User<br>Include: Validate Data<br>Extend:<br>Generalization: | | |
| Normal Flow of Events:<br>1. The user want to add an item in order to be stored and used in future<br>2. The user choose the type of item they want to add.<br>3. The user fill the information of the item<br>4. The system validates the required information provided by the user.<br>5. The system saves the item information in the database.<br>6. The system populate items in the main window. | | |
| Sub Flows:<br>Not Applicable | | |
| Alternate/Exceptional Flows:<br>Not applicable | | |

*Table 3-1 Use Case Description of Add Item*

| Use Case Name: Edit Item | ID : 2 | Importance Level: Middle |
|---|---|---|
| Primary Actor: User | | |
| Summary Description:<br>User want to modify the information of the item | | |
| Relationship:<br>Association: User<br>Include: Validate Data<br>Extend:<br>Generalization: | | |
| Normal Flow of Events:<br>1. The user want to modify the information of the item.<br>2. The user select the item to be edited from the main window.<br>3. The user edit the information of the item.<br>4. The user confirm the information being edited.<br>5. The system update the item information in the database.<br>6. The system populate items in the main window. | | |

| |
|---|
| Sub Flows:<br>Not Applicable |
| Alternate/Exceptional Flows:<br>Not applicable |

*Table 3-2 Use Case Description of Edit Item*

| Use Case Name: Search Item | ID : 3 | Importance Level: Low |
|---|---|---|
| Primary Actor: User | | |
| Summary Description:<br>User want to search for specific item based on the title, date, and author name. | | |
| Relationship:<br>Association: User<br>Include:<br>Extend:<br>Generalization: | | |
| Normal Flow of Events:<br>1. The user choose the field they want to search for.<br>2. The user enter the search term.<br>3. The system search the item based on the search term provided by user.<br>4. The system filter and display the item to the user. | | |
| Sub Flows:<br>Not Applicable | | |
| Alternate/Exceptional Flows:<br>Not applicable | | |

*Table 3-3 Use Case Description of Search Item*

| Use Case Name: Sort Item | ID :4 | Importance Level: Low |
|---|---|---|
| Primary Actor: User | | |
| Summary Description<br>User want to sort the items by the author name, title, year, date. | | |
| Relationship:<br>Association: User<br>Include:<br>Extend:<br>Generalization: | | |
| Normal Flow of Events:<br>1. The user click the header of the data column they want to sort.<br>2. The system rearrange the list of data. | | |
| Sub Flows:<br>Not Applicable | | |
| Alternate/Exceptional Flows:<br>Not applicable | | |

*Table 3-4 Use Case Description of Sort Item*

| Use Case Name: View Item Data | ID : 5 | Importance Level: Middle |
|---|---|---|
| Primary Actor: User | | |
| Summary Description<br>User want to view the information detail of item. | | |
| Relationship:<br>Association: User<br>Include:<br>Extend:<br>Generalization: | | |
| Normal Flow of Events:<br>1. The user want to view information detail of item.<br>2. The user select the item they want to view.<br>3. The system retrieve data from database.<br>4. The system display the item data in small window. | | |
| Sub Flows:<br>Not Applicable | | |
| Alternate/Exceptional Flows:<br>Not applicable | | |

*Table 3-5 Use Case Description of View Item Data*

| Use Case Name: Delete Item | ID : 6 | Importance Level: High |
|---|---|---|
| Primary Actor: User | | |
| Summary Description<br>User want to delete an item | | |
| Relationship:<br>Association: User<br>Include:<br>Extend:<br>Generalization: | | |
| Normal Flow of Events:<br>1. The user want to delete an item.<br>2. The user select the item they want to delete.<br>3. The user choose the option to delete the item.<br>4. The system delete the item data from the database.<br>5. The system populates items in main window. | | |
| Sub Flows:<br>Not Applicable | | |
| Alternate/Exceptional Flows:<br>Not applicable | | |

*Table 3-6 Use Case Description of Delete Item*

| Use Case Name: Import PDF | ID : 7 | Importance Level: High |
|---|---|---|
| Primary Actor: User | | |
| Summary Description:<br>User want to add PDF document in order to store in the system | | |
| Relationship:<br>Association: User<br>Include:<br>Extend:<br>Generalization: | | |
| Normal Flow of Events:<br>1. The user want to add PDF document in order to store it in the system<br>2. The user choose the document from the external environment (desktop).<br>3. The user confirm the document to be saved.<br>4. The system insert the document path into the database.<br>5. The system display the list of document stored. | | |
| Sub Flows:<br>Not Applicable | | |
| Alternate/Exceptional Flows:<br>Not applicable | | |

*Table 3-7 Use Case Description of Import PDF*

| Use Case Name: View PDF | ID :8 | Importance Level: Middle |
|---|---|---|
| Primary Actor: User | | |
| Summary Description:<br>User want to view the PDF that stored in the system. | | |
| Relationship:<br>Association: User<br>Include:<br>Extend: Edit PDF<br>Generalization: | | |
| Normal Flow of Events:<br>1. The user want to view the PDF document that stored in the system.<br>2. The user choose the PDF document that listed in the system.<br>3. The system verify the existence of the document.<br>4. The system render the PDF document in PDF viewer embedded. | | |
| Sub Flows:<br>Not Applicable | | |
| Alternate/Exceptional Flows:<br>Not applicable | | |

*Table 3-8 Use Case Description of View PDF*

| Use Case Name: Edit PDF | ID : 9 | Importance Level: Middle |
|---|---|---|
| Primary Actor: User | | |
| Summary Description<br>User want to comment, highlight, create bookmark and etc. in PDF document. | | |
| Relationship:<br>Association: User<br>Include:<br>Extend:<br>Generalization: | | |
| Normal Flow of Events:<br>1. The user want to edit PDF document.<br>2. Use Case ID 8 is performed.<br>3. The user use the tool available to PDF document within the viewer. | | |
| Sub Flows:<br>Not Applicable | | |
| Alternate/Exceptional Flows:<br>Not applicable | | |

*Table 3-9 Use Case Description of Edit PDF*

| Use Case Name: Delete PDF | ID : 10 | Importance Level: Middle |
|---|---|---|
| Primary Actor: User | | |
| Summary Description<br>User want to remove PDF document from the system | | |
| Relationship:<br>Association: User<br>Include:<br>Extend:<br>Generalization: | | |
| Normal Flow of Events:<br>1. The user want to delete PDF from the system.<br>2. The user select the PDF document in the list.<br>3. The user choose the option to delete PDF.<br>4. The system delete the record from database.<br>5. The system populate the PDF in the list. | | |
| Sub Flows:<br>Not Applicable | | |
| Alternate/Exceptional Flows:<br>Not applicable | | |

*Table 3-10 Use Case Description of Delete PDF*

| Use Case Name: Generate Citation and Reference | ID : 11 | Importance Level: High |
|---|---|---|
| Primary Actor: User | | |
| Summary Description:<br>User want to create citation and bibliography from the item. | | |
| Relationship:<br>Association: User<br>Include: Choose citation style<br>Extend:<br>Generalization: | | |
| Normal Flow of Events:<br>1. The user want to create citation and bibliography from the item.<br>2. The user choose the item which stored in the system to be cited.<br>3. The user choose the citation style for the cited item.<br>4. The user confirm the chosen option.<br>5. The system produce the citation and bibliography in the desired format. | | |
| Sub Flows:<br>Not Applicable | | |
| Alternate/Exceptional Flows:<br>2a. The user choose to generate citation and bibliography instantly.<br>3a. The user choose the type of item to be cited.<br>4a. The user choose enter the item information required.<br>5a. The user choose the citation style for the cited item.<br>6a. The user confirm the input provided.<br>7a. The system produce the citation and bibliography in the desired format | | |

*Table 3-11 Use Case Description of Generate citation and reference*

| Use Case Name: Attach URI | ID : 12 | Importance Level: Middle |
|---|---|---|
| Primary Actor: User | | |
| Summary Description<br>User want to attach URI to an item. | | |
| Relationship:<br>Association: User<br>Include:<br>Extend:<br>Generalization: | | |
| Normal Flow of Events:<br>1. The user want to attach URI to an item.<br>2. The user select an existing item.<br>3. The user choose the option to add URI. | | |

| |
|---|
| 4. The user enter the URI into field.<br>5. The system validate the URI entered by user.<br>6. The system update the item information to the database,<br>7. The system populate the item data in small window. |
| Sub Flows:<br>Not Applicable |
| Alternate/Exceptional Flows:<br>6a. The system display message “Invalid URI has been entered” to the user.<br>6b. The user re-enter the URI again.<br>6c. The system validate the URI entered by user. |

*Table 3-12 Use Case Description of Attach URI*

| Use Case Name: Attach File | ID : 13 | Importance Level: Middle |
|---|---|---|
| Primary Actor: User | | |
| Summary Description<br>User want to attach link to file to an item. | | |
| Relationship:<br>Association: User<br>Include:<br>Extend:<br>Generalization: | | |
| Normal Flow of Events:<br>1. The user want to attach link to file to an item.<br>2. The user select an existing item.<br>3. The user choose the option to add link to file.<br>4. The user choose the document from the external environment (desktop).<br>5. The user confirm the document to be saved.<br>6. The system insert the document path into the database.<br>7. The system populate the item data in small window. | | |
| Sub Flows:<br>Not Applicable | | |
| Alternate/Exceptional Flows:<br>Not applicable | | |

*Table 3-13 Use Case Description of Attach File*

| Use Case Name: Create Mind Map | ID : 14 | Importance Level: High |
|---|---|---|
| Primary Actor: User | | |
| Summary Description:<br>User want to create mind map in order to draft their research. | | |
| Relationship:<br>Association: User<br>Include:<br>Extend: Export to file, Change font, Insert text, Link to Item<br>Generalization: | | |
| Normal Flow of Events:<br>1. The user want to create mind map in order to draft their research.<br>2. The user choose the option to create new mind map.<br>3. The system create a new mind map inside a window. | | |
| Sub Flows:<br>Not Applicable | | |
| Alternate/Exceptional Flows:<br>Not applicable | | |

*Table 3-14 Use Case Description of Create Mind Map*

| Use Case Name: Export to File | ID : 15 | Importance Level: Middle |
|---|---|---|
| Primary Actor: User | | |
| Summary Description<br>User want to export their draft to text or document file. | | |
| Relationship:<br>Association: User<br>Include:<br>Extend:<br>Generalization: | | |
| Normal Flow of Events:<br>1. The user want to export their draft to text or document file.<br>2. The user click the option “Export” in menu bar.<br>3. The user select the option “Export to text file”.<br>4. The system pops out a save dialog box.<br>5. The user confirm the operation.<br>6. The system create a file to the external environment. | | |
| Sub Flows:<br>Not Applicable | | |
| Alternate/Exceptional Flows:<br>3a. The user select the option ‘Export to document file”. | | |

*Table 3-15 Use Case Description of Export to File*

| Use Case Name: Drag and drop text | ID : 16 | Importance Level: High |
|---|---|---|
| Primary Actor: User | | |
| Summary Description<br>User want to copy the text from PDF and add it to mind map. | | |
| Relationship:<br>Association: User<br>Include: Link to File<br>Extend:<br>Generalization: | | |
| Normal Flow of Events:<br>1. A mind map has been created.<br>2. A PDF document has been opened in PDF Viewer.<br>3. The user select text in the PDF document.<br>4. The user hold the right click mouse button and release when the mouse cursor dragged to the mind map.<br>5. The system copy the text and add a node with the text into mind map. | | |
| Sub Flows:<br>Not Applicable | | |
| Alternate/Exceptional Flows:<br>Not applicable | | |

*Table 3-16 Use Case Description of Drag and drop text*

| Use Case Name: Link to Item | ID : 15 | Importance Level: Middle |
|---|---|---|
| Primary Actor: User | | |
| Summary Description<br>User want to create link to reference item for an node, | | |
| Relationship:<br>Association: User<br>Include:<br>Extend:<br>Generalization: | | |
| Normal Flow of Events:<br>1. The user want to link a node to a reference item.<br>2. The user choose the option”insert citation” option.<br>3. The user select a reference item.<br>4. The system update the node property and create link event between them. | | |
| Sub Flows:<br>Not Applicable | | |
| Alternate/Exceptional Flows: | | |

*Table 3-17 Use Case Description of Link to Item*

### 3.3.3.4 Window Navigation Diagram

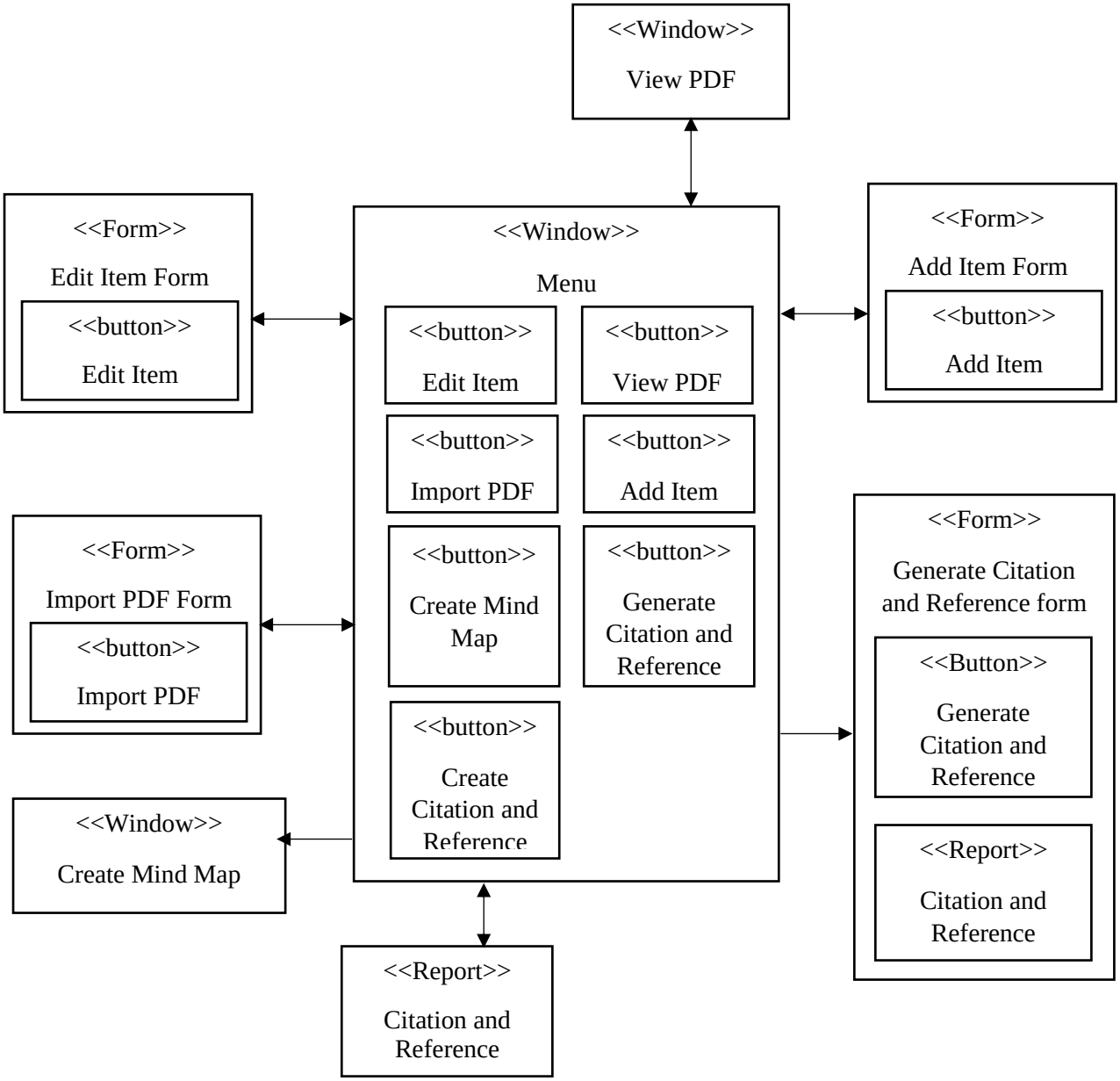



*Figure 3-37 Window Navigation Diagram of Smart Citation Manager*

### 3.3.4 Interface of Smart Citation Manager

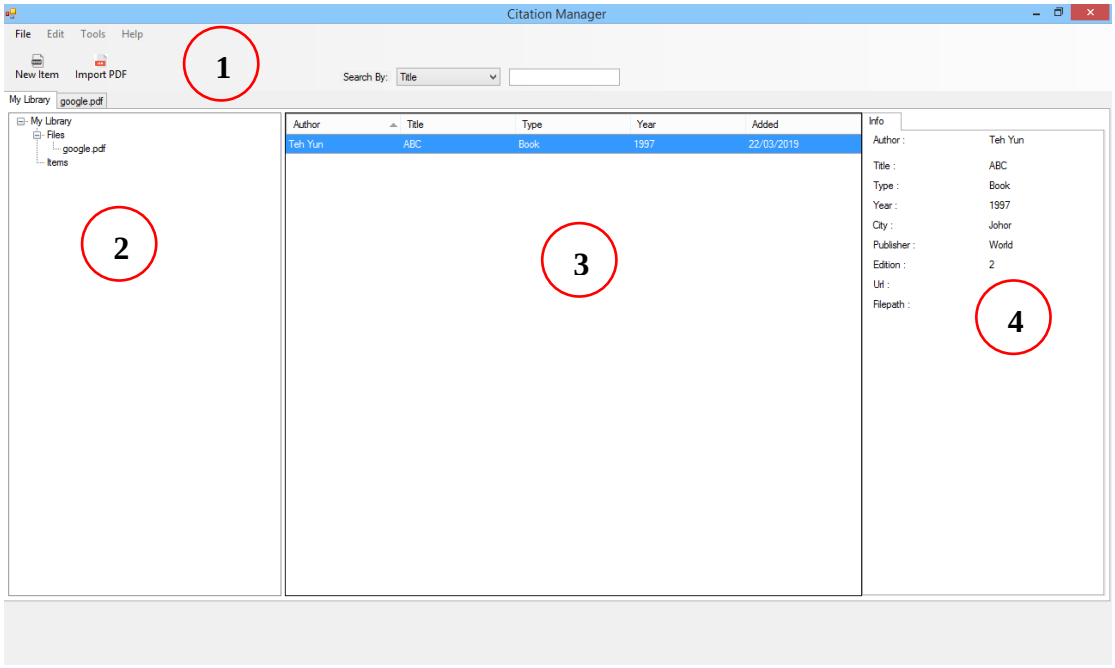



*Figure 3-38 Main Window of Smart Citation Manager*

The figure above show the main window of Smart Citation Manager.

1. The top part is the menu bar that provide functions such as importing new items, new file, and creating new mind map. Besides that, a search bar is available for user to search for the title, year or author name.
2. The left section is the area that list the PDF that have been imported.
3. This area show the list of items and the relevant data have been created.
4. The right section show the detailed information of a selected item.

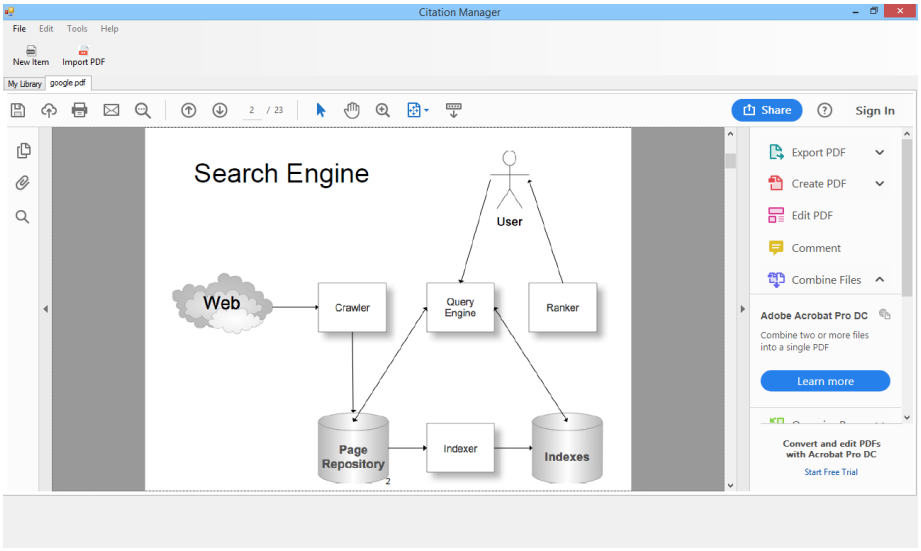



*Figure 3-39 PDF viewer embedded in Smart Citation Manager*

The figure above show the interface when the user shift to the tab which the pdf viewer embedded. The interface of PDF viewer are same with the Adobe Acrobat Reader software which provide the functions to view and manipulate files.

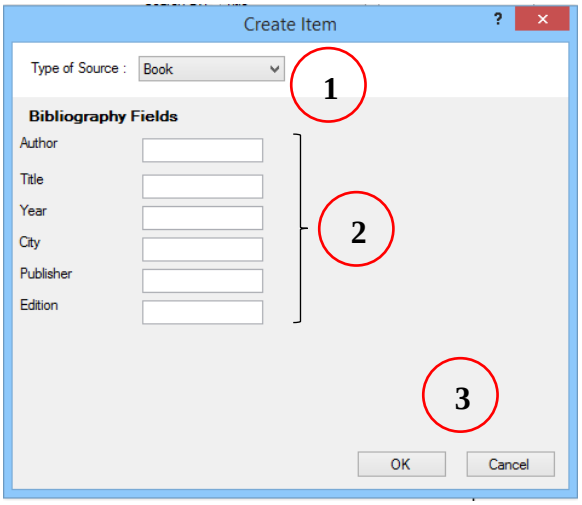



*Figure 3-40 A window prompt user to enter data for creating item*

The figure above show the create item form which user can create new item by filling the data.

1. A combo box that has a list of item type to be chosen.
2. User is required to enter the information for a new item. The fields will be different for each of types selected.
3. OK button is to confirm the data has been entered and create a new item. Cancel button is to close the window without saving the data.

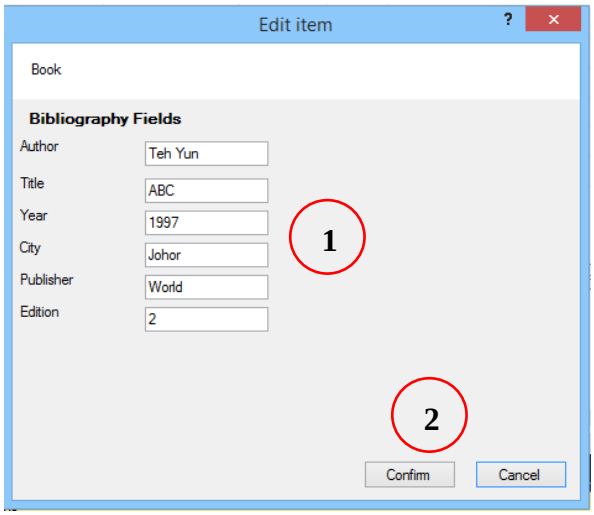



*Figure 3-41 A window prompt user to modify item data*

The figure above show the edit item window which has similar interface with create item table.

1. The data in the text box is previous inserted data that pulled from the database.
2. Confirm button is to confirm the change that has been made. Cancel button is to close the window without save the changes.

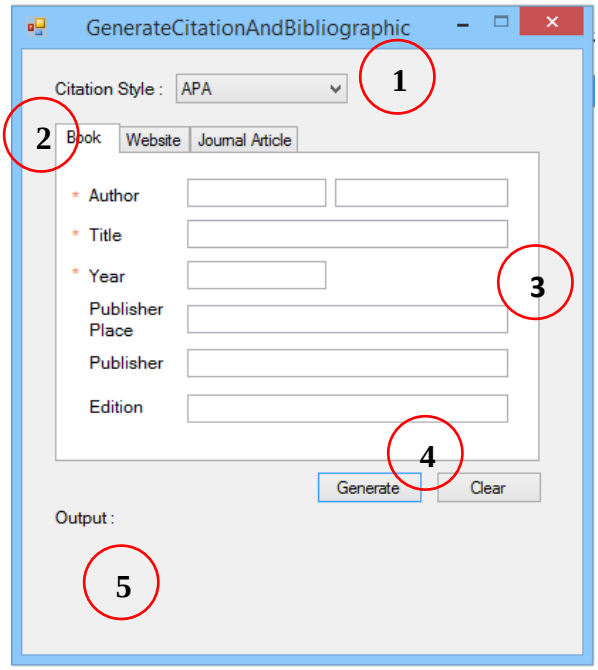



*Figure 3-42 A window for generate citation and reference instantly*

The figure above show a form which is used to create citation and reference instantly from the information provided by the user.

1. The combo box show the list of citation style that can be chosen.
2. The tab pages show the different type of item to be cited.
3. Its relevant field to be filled.
4. Generate button is to create the citation and reference based on the information provided in section 3.
5. The area that the result will be shown.

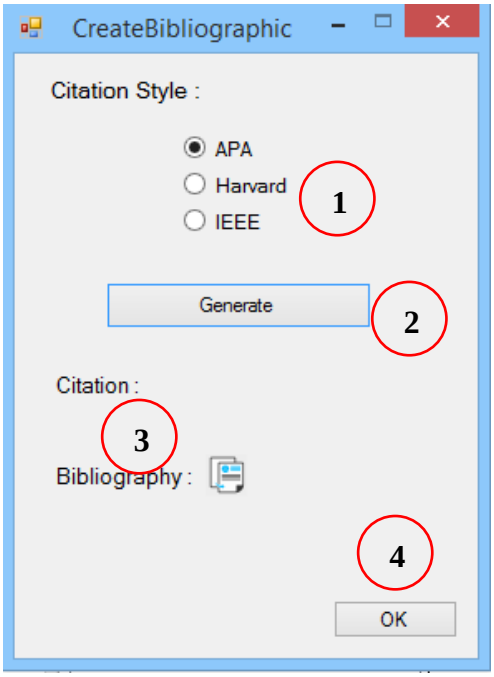



*Figure 3-43 A window for generate citation and reference from selected item*

The figure above show the window to create citation and reference from the item stored in system.

1. The citation styles to be chose to cite the item.
2. Generate button clicked to produce the citation and reference.
3. The area show the output.
4. OK button to close the window.

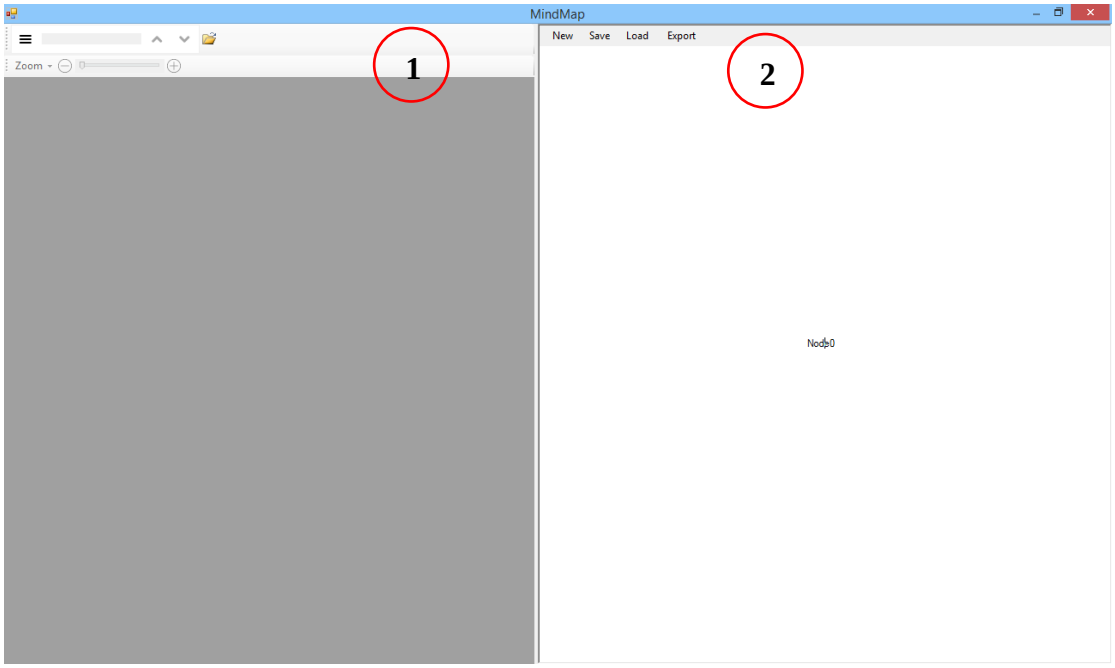



*Figure 3-44 A window where PDF viewer and mind map panel embedded*

The figure above show a mind map window. This window is separated into two sections. The first section is a PDF viewer to render the PDF

1. The user able to open PDF, search through the file, and zoom in and out the file in the PDF viewer.
2. In the mind map, user able to create new node, edit and change the font of the node text. Besides that, the node created is movable within the area and can be linked to another node.
3. The menu placed at the top right provide the user options to create new, load and save mind map to the computer local storage. If the user need to further elaborate their idea draft, they can export it to text document or word document.

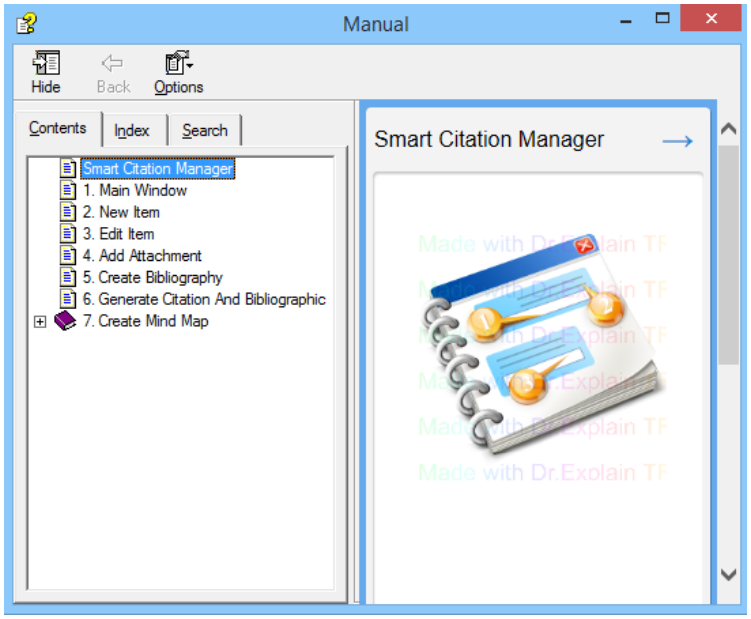



*Figure 3-45 A window where PDF viewer and mind map panel embedded*

A user manual is created to explain how to use Smart Citation Manager and what the program can do for the user. This manual can be opened by clicking ‘Help” within the program.

# CHAPTER 4 METHODOLOGY AND TOOL

## 4.1 System Development Life Cycle (SDLC) Methodology

According to Mohammed Sami (2012), SDLC is a series of phases that provide the understanding on activities of software development cycles. It describes how the software will be developed from the requirements elicitation phase to convert the requirements into necessary specifications until the features and functions satisfy the user requirement and achieve business goals. SDLC is a multistep, structured in methodical way which model a framework for both technical and non-technical activities to deliver a quality system (Innovativearchitects, n.d.).

SDLC activities can be broken down into 6 stages: planning, system analysis and requirement, system design, development, integration and testing, and maintenance. In planning phase, the scope of problem and solution are found out and the resource, cost, and time are considered here. In system analysis and requirement phase, teams should consider the functional requirement of the solution and analyse the need of end users to ensure their expectations are met. In system design phase, the necessary specifications, features and functions are described in detail. Building, integrating and testing the proposed system are carried out in development and testing phase. In the maintenance phase which is the last phase of SDLC, adaption, error correction and adding new functionality are usually involved.

The success of a projects is highly depend on choosing the right SDLC model according to project's specific requirement and concern. After careful consideration on time, project size, complexity and etc., prototyping methodology is adopted to develop Smart Citation Manager.

### 4.1.1 Prototyping Methodology

Prototyping model is one of the system development method in which a working replication of a system or product is built, tested and reworked until achieve an acceptable prototype from which final product or solution can be developed (Rouse, 2005).

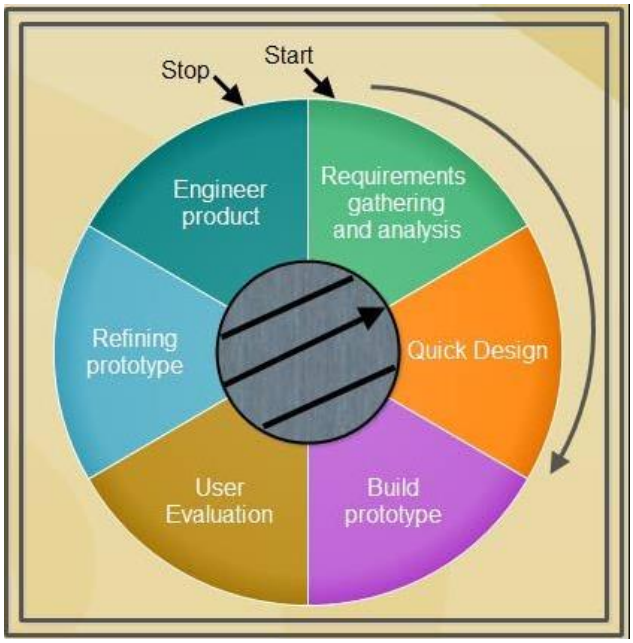



*Figure 4-1 Lifecycle of prototyping methodology* (Thakur, n.d.)

There are several steps carried out in prototyping model:

#### 4.1.1.1 Requirements Gathering and Analysis

During this phase, requirements are gathered during meeting with project supervisor. Besides that, problems of existing system are identified and evaluated in order to specify the project scope and objectives of Smart Citation Manager. Feasibility analysis are also performed to make sure the project development is realistic. The outcome of this phase is the preliminary report which consist of problem statement, project scope, project objectives and proposed solution of Smart Citation Manager.

#### 4.1.1.2 Quick Design

During quick design phase, functional and structural diagrams are used to describe the important aspects of the system. ERD model and CRC cards are used to design the database used by Smart Citation Manager. However, the idea of Smart Citation Manager is explained in use case, flow chart, and activity diagram.

#### 4.1.1.3 Build Prototype

During this phase, information gathered from quick design is modified to form the first prototype of Smart Citation Manager. It is a working model that can perform the core functions of the required system which provide the understanding of the proposed solution.

#### 4.1.1.4 User Evaluation

The prototype is then presented to the project supervisor and moderator for thorough evaluation to identify its strengths and weaknesses. Comments and suggestions for the system improvement are collected.

#### 4.1.1.5 Refine Prototype

During this phase, the current prototype of Smart Citation Manager is refined according to the suggestions gathered. The requirements provided will be evaluated by the developed and used to modify the prototype. Therefore, the new prototype will be evaluated just like the previous prototype and this process will loop until all the requirements specified are met. System testing will be conducted to ensure Smart Citation Manager free of defects and issues before being deliver to end user.

#### 4.1.1.6 Engineer Product

The final version of Smart Citation Manager will be developed on the basis of final prototype. The demonstration of Smart Citation Manager will be conducted in this phase so future enhancement can be achieved by collected the feedbacks.

## 4.2 Technology and Tool

There are several technologies and tools used to develop Smart Citation Manager: C# programming language. Microsoft SQL, Adobe PDF Reader, and PDFium.

### 4.2.1 C# Programming Language

C# programming language is an object-oriented programming language that offer ease to use, familiarity to C++ and java developer. Smart Citation Manager as computer software that is developed using C# language. The reasons to choose C# programming language as primary tool in building Smart Citation Manager is because C# is easier to use. C# is derived from C and C++ which Microsoft intend to build the language with number of objectives like simple and ease of use.

### 4.2.2 Microsoft SQL

SQL Server is a relational database management system and also object-relational database management system that developed by Microsoft. Microsoft SQL server is used to store and retrieve the data like item, file and other important information in this project. Microsoft SQL is selected as the project database is because it's native compatibility with .NET. It make the integration simpler for Window projects.

### 4.2.3 Adobe Acrobat Reader

There are several popular PDF reader such as Adobe Acrobat Reader, Foxit Reader, PDF-XChange Editor, and etc. Adobe PDF Reader is selected to be embedded in Smart Citation Manager because it provide free library: AcroPDFlib and AxAcroPDFLib which able to render PDF document in .NET application. The object has similar functionalities with Adobe Acrobat Reader which the user able to view, edit, make annotate the PDF within Smart Citation Manager.

### 4.2.4 Google PDFium

In order to enable user retrieve the interesting ideas in the PDF and organize them inside a mind map seamlessly, Google PDF software library, PDFium is open source PDF rendering engine for .NET application. PDFium is selected to be used because its object component has the ability to detect the selected text and perform drag and drop function easily.

## 4.3 Implementation and Requirements

In this session, the implementation of Smart Citation Manager and necessary software will be discussed. Besides that, the implementation requirements of the software are also described here.

### 4.3.1 Adobe Acrobat Reader

#### 4.3.1.1 System Requirement

- 1.5Hz or faster processor
- Windows Server 2008 (32 bit and 64 bit), 2012 (64 bit), Windows 7 (32 bit and 64 bit), Windows 8 (32 bit and 64 bit), Windows 10
- 1GB of RAM
- 380MB of available hard-disk space
- 1024x768 screen resolution
- Internet Explorer 11

#### 4.3.1.2 System Installation

Step 1: Download it from the Internet: https://get.adobe.com/reader/

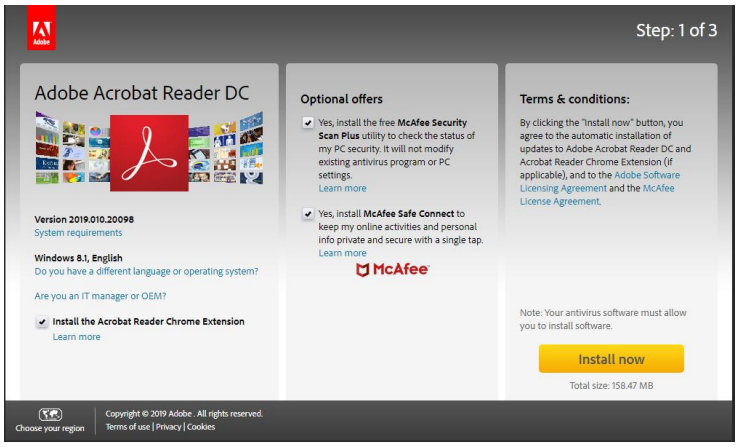



*Figure 4-2 Adobe Acrobat Reader Installation Guide (Part 1)*

Step 2: Launch the installer has been downloaded.

Step 3: Click the “Finish” button to finish the installation.

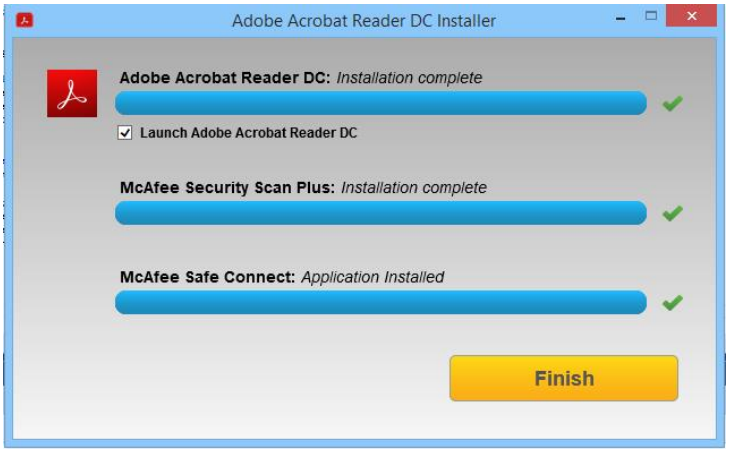



*Figure 4-3 Adobe Acrobat Reader Installation Guide (Part 2)*

### 4.3.2 Microsoft SQL Server

#### 4.3.2.1 System Requirement

- .NET Framework 4.6 or later
- Minimum 6GB of available hard-disk space
- Minimum x64 Processor: 1.4 GHz

#### 4.3.2.2 System Installation

Step 1: Download it from the Internet: https://www.microsoft.com/en-us/sql-server/sql-server-downloads

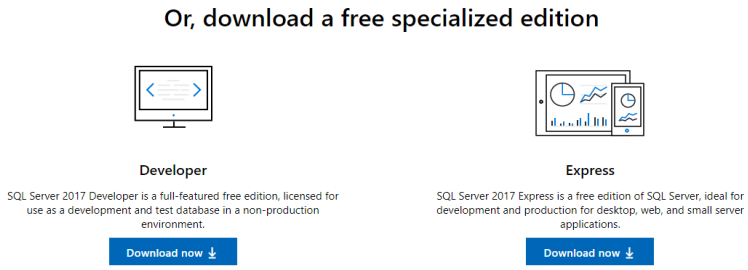



*Figure 4-4 Microsoft SQL Server Installation Guide (Part 1)*

Step 2: Launch the installer has been downloaded.

Step 3: Click the “Basic” installation type.

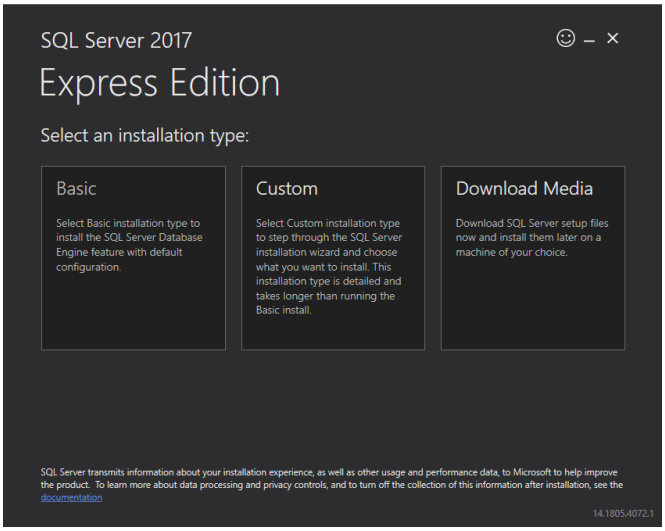



*Figure 4-5 Microsoft SQL Server Installation Guide (Part 2)*

Step 4: Click the “Accept” button after agreed on the license terms and privacy statement.

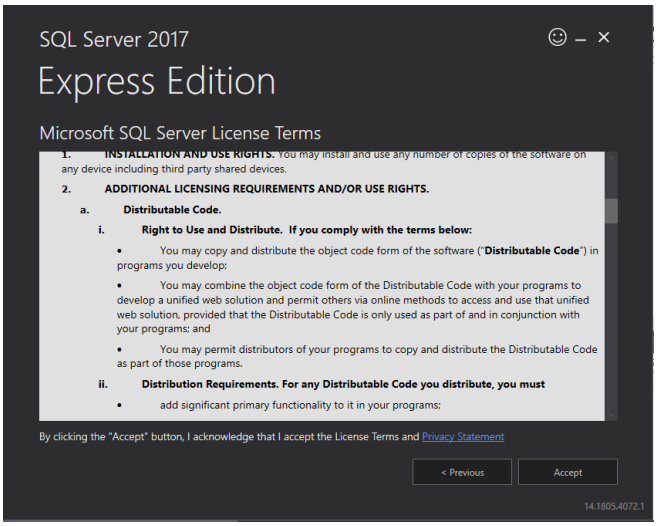



*Figure 4-6 Microsoft SQL Server Installation Guide (Part 3)*

Step 5: Choose the location for the system and click the “install” button.

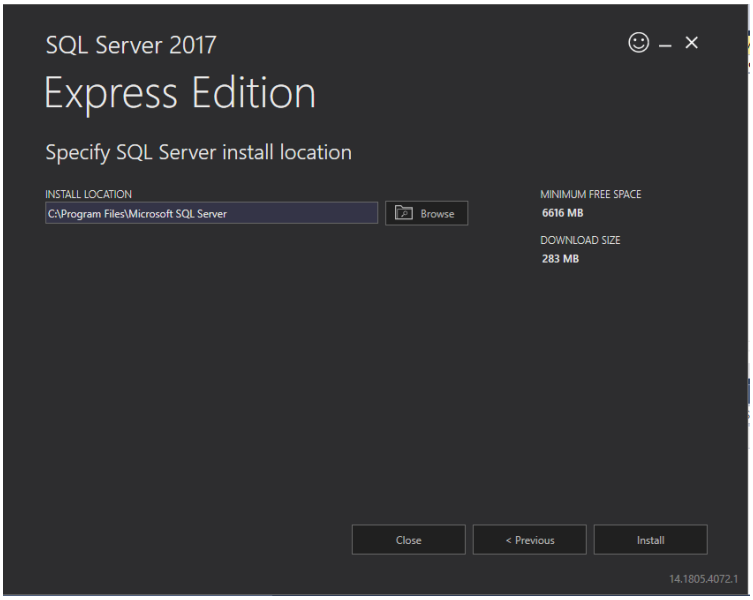



*Figure 4-7 Microsoft SQL Server Installation Guide (Part 4)*

### 4.3.3 Smart Citation Manager

#### 4.3.3.1 System Requirement

- Microsoft Windows XP or Windows 7 or Windows 8 or Windows 10
- 2GB of RAM
- 100 MB of available hard-disk space

## 4.4 Implementation Issues and Challenges

The first challenge of implementing the project is the database design for Smart Citation Manager. A well-designed database is essential in ensuring the data consistency, minimization of data redundancy, and high performance for application system. It is a challenge for this project as there are different type of item which contain different information to be stored in the system. However, most of them contain similar data such as author, title and year of the item. It is important to make a trade-off between data redundancy and efficiency of query execution. Besides that, the way the data to be stored should be took into consideration in order to ensure the efficiency and ease of data retrieval for processing.

Secondly, a .NET CSL library is used in Smart Citation Manager to format the data to citation and reference according to the citation style. The CSL style is XML-based format which describe how a citation is created and the CSL processor will process the data which in JSON format with a loaded style file. However, the CSL processor cannot be guaranteed that it can support the latest version of citation style format as the styles are constantly updated. Besides that, the CSL processor also take time to process the data. Few seconds will be delayed to generate the result.

## 4.5 Project Timeline

Final Year Project I:

- Determine the problems and weak of existing systems.
- Gather user requirement toward the solution.
- Analyse similar system and determine the technique and algorithm to solve the problem.
- Determine the scope of the project.
- Prepare preliminary system design.
- Create prototype for some functionalities.
- Prepare FYP1 documentation.
- Deliver prototypes to moderator.

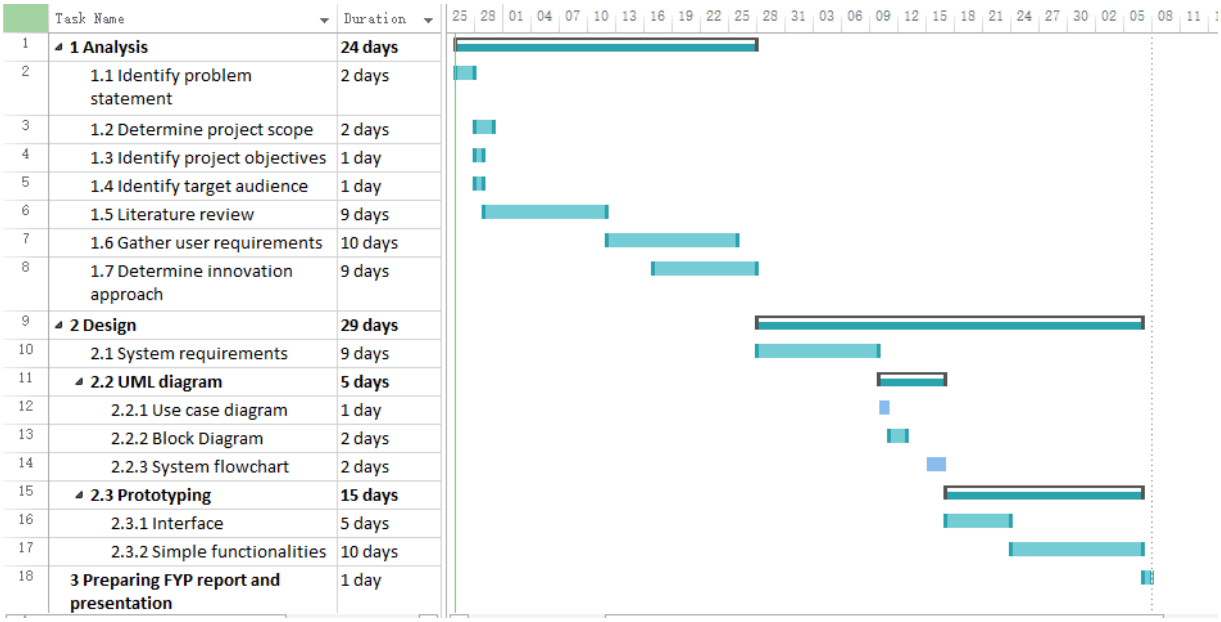



*Figure 4-8 Gantt Chart of Project 1*

Final Year Project II:

- Refine prototype to solicit additional requirements.
- Finalize the system design and component.
- Coding the complete system.
- Prepare various test plan and implement the test on the system
- Conduct user acceptance test plan
- Prepare complete system available to user
- Collect user feedback and document possible enhancement.
- Prepare FYP2 documentation.
- Deliver prototypes to moderator.

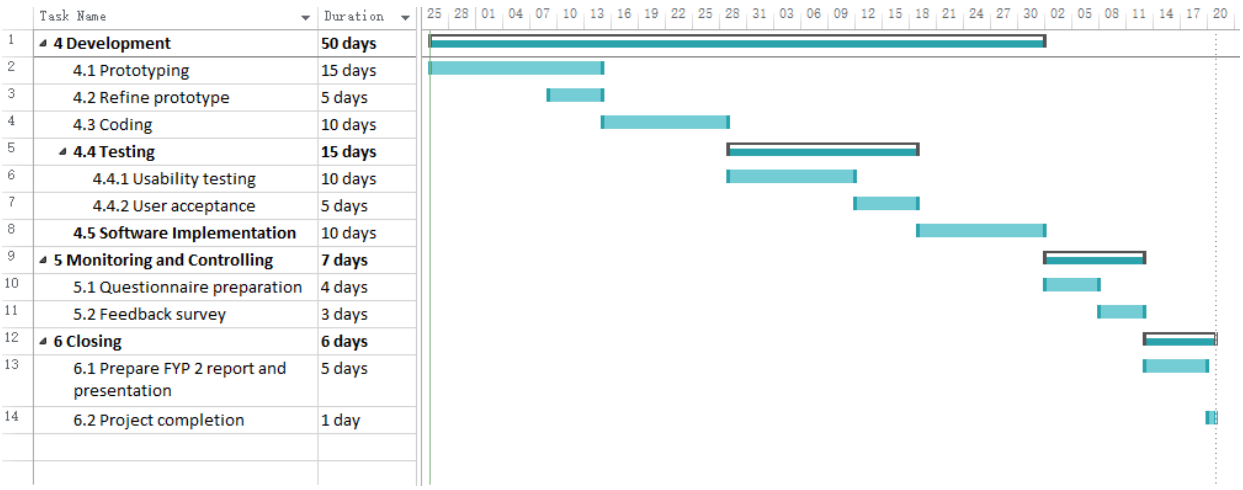



*Figure 4-9 Gantt Chart of Project 2*

# CHAPTER 5 SYSTEM TESTING

## 5.1 Unit Test

Unit testing is one of the software testing method where individual components of a software are tested. Unit test is conducted to validate each unit of Smart Citation Manager can performs as designed.

### 5.1.1 Insert Item Test Case

| Test Case | Test Case Description | Test Data/Test Steps | Expected Result | Actual Result | Pass/Fail |
|---|---|---|---|---|---|
| T1a | Check the response when all the fields are entered with data. | Type: Book<br>Author: Teh Yun<br>Title: ABC<br>Year: 1997<br>City: KL<br>Publisher: Pearson<br>Edition: 2 | The item should be created successfully. | Same as expected result | Pass |

*Table 5-1 Test Case of Insert Item*

### 5.1.2 Edit Item Test Case

| Test Case | Test Case Description | Test Data/Test Steps | Expected Result | Actual Result | Pass /Fail |
|---|---|---|---|---|---|
| T2a | Check whether the item shown in the form is the item selected to be edited. | Type: Book<br>Author: Teh Yun<br>Title: ABC<br>Year: 1997<br>City: Johor<br>Publisher: World<br>Edition: 2 | The data retrieved to be edited should same with the data of item selected. | Type: Book<br>Author: Teh Yun<br>Title: ABC<br>Year: 1997<br>City: Johor<br>Publisher: World<br>Edition: 2 | Pass |
| T2b | Validate the change when the item information is modified. | Type: Book<br>Author: Teh<br>Title: ABC<br>Year: 1997<br>City: Johor<br>Publisher: Hello<br>Edition: 2 | The change made should be successful. | Type: Book<br>Author: Teh<br>Title: ABC<br>Year: 1997<br>City: Johor<br>Publisher: Hello<br>Edition: 2 | Pass |

*Table 5-2 Test Case of Edit Item*

### 5.1.3 Import PDF Test Case

| Test Case | Test Case Description | Test Data/Test Steps | Expected Result | Actual Result | Pass /Fail |
|---|---|---|---|---|---|
| T3a | Check whether the type of file chosen only can be .pdf. | Click “Import file” button. | Only files with .pdf extension are allowed to be imported | Same as expected result | Pass |
| T3b | Check whether the imported file is same with the file selected to be imported. | RMS Comparison.pdf is imported. | A message is popped up to alert user. | Same as expected result | Fail |
| T3c | Check whether the file list is updated when a file is imported or deleted | RMS Comparison.pdf is deleted. | The list is updated. | Same as expected result | Pass |
| T3d | Check the response when there is an existing file. | A file is imported twice. | A message is popped up to alert user. | There is two file with same name appear in the list. | Fail |

*Table 5-3 Test Case of Import PDF*

### 5.1.4 Create Citation and Bibliography Test Case

| Test Case | Test Case Description | Test Data/Test Steps | Expected Result | Actual Result | Pass/Fail |
|---|---|---|---|---|---|
| T4a | Validate the output generated based on the selected item. | 1.Select an item and right click to select “create citation and bibliography” option 2.Click generate button with a style chosen. | The data and citation format are accurate. | Same as expected result. | Pass |
| T4b | Validate the output generated for each style chosen. | 1.Select an item and right click to select “create citation and bibliography” option 2.Click generate button for each citation style. | The format is change according to style chosen and accurate. | Same as expected result. | Pass |
| T4c | Check the response when copy button is | N/A | “Please fill in required field” | Same as expected result. | Pass |

| | | | | | |
|---|---|---|---|---|---|
| | clicked before the output is generated | | message should appear. | | |
| T4d | Validate the output that copied to the clipboard | N/A | The exact output is copied successfully. | Same as expected result. | Pass |

*Table 5-4 Test Case of Create Citation and Bibliography*

### 5.1.5 Generate Citation and Bibliography Instantly Test Case

| Test Case | Test Case Description | Test Data/Test Steps | Expected Result | Actual Result | Pass /Fail |
|---|---|---|---|---|---|
| T5a | Check the response when all the fields are entered with data. | Type: Book<br>Title: Hello World<br>Author: Zhi Yun<br>Year: 1999<br>Publisher Place: Perak<br>Publisher: ABC<br>Edition: 2 | A "Creating" message will be shown and the output is generated. | Same as expected result. | Pass |
| T5b | Check the response when only the required fields are filled with data. | Type: Book<br>Title: Hello World<br>Author: Zhi Yun<br>Year: 1999<br>Publisher Place:<br>Publisher:<br>Edition: | A "Creating" message will be shown and the output is generated. | Same as expected result. | Pass |
| T5c | Check the response when some or all the required fields are left with blank. | Type: Book<br>Title:<br>Author:<br>Year: 1999<br>Publisher Place:<br>Publisher:<br>Edition: | A "Please fill in required fields" is shown. | Same as expected result. | Pass |
| T5d | Validate the output based on the input provided and style selected | Style: APA, MLA<br>Type: Book<br>Title: Hello World<br>Author: Zhi Yun<br>Year: 1999<br>Publisher Place: Perak<br>Publisher: ABC<br>Edition: 2 | The output is generated based on the style selected. | APA:<br>Yun, Z. (1999) Hello World (2nd ed.). Perak: ABC.<br>Harvard<br>MLA:<br>Yun, Zhi. Hello World. 2nd ed., ABC, 1999. | Pass |

*Table 5-5 Test Case of Generate Citation and Bibliography Instantly*

### 5.1.6 Perform Actions on Item

| Test Case | Test Case Description | Test Data/Test Steps | Expected Result | Actual Result | Pass /Fail |
|---|---|---|---|---|---|
| T6a | Validate the information displayed when an item is selected | A Book Item is clicked<br>Title: ABC<br>Author: Teh<br>Year: 2000<br>City: Johor<br>Publisher: ABCD<br>Edition: 2 | The information being displayed should from the item selected. | A Book Item is clicked<br>Title: ABC<br>Author: Teh<br>Year: 2000<br>City: Johor<br>Publisher: ABCD<br>Edition: 2 | Pass |
| T6b | Check the response when selected item is deleted. | An item is selected to be deleted. | There is no record for the selected item displayed in the table. | Same as expected result | Pass |
| T6c | Attempt to attach file to an item | A note file. | There is a clickable link appear in the item information. | Same as expected result | Pass |
| T6e | Check the response when invalid URL is provided | Abcde | A message “invalid url” is shown | Same as expected result | Pass |
| T6f | Check the response when valid URL is entered | www.google.com | The url is added successfully. | Same as expected result | Pass |

*Table 5-6 Test Case of Perform Actions on Item*

### 5.1.7 Create Mind Map Test Case

| Test Case | Test Case Description | Test Data/Test Steps | Expected Result | Actual Result | Pass /Fail |
|---|---|---|---|---|---|
| T7a | Add node inside mind map | 1.Right click the panel<br>2.Click the “Add” option | New node is added successfully. | Same as expected result | Pass |
| T7b | Draw a line from one node to another | 1.Right click a node and hold it<br>2.Drag to link another node | A line is formed between two nodes. | Same as expected result | Pass |
| T7c | Delete a selected node | 1.Right click a node<br>2.Click the “Delete” option | The selected node is deleted successfully. | Same as expected result | Pass |
| T7d | Change the font of node text | 1.Right click a node<br>2.Click the “Font” option<br>3.Choose the font style | The text is changed based on the style chosen. | Same as expected result | Pass |
| T7e | Change the colour of text | 1.Right click a node<br>2.Click the “Text Colour” option<br>3.Choose the colour | The text colour is changed to the colour chosen. | Same as expected result | Pass |
| T7f | Change the back colour of node | 1.Right click a node<br>2.Click the “Back Colour” option<br>3.Choose the colour | The node colour is change to the colour chosen. | Same as expected result | Pass |
| T7g | Change the text of node | 1.Double click the node<br>2.Modify the text. | The node text is changed successfully. | Same as expected result | Pass |
| T7h | Click “New” button to create new mind map | N/A | The new mind map is created. | Same as expected result | Pass |
| T7i | Click “Save” button to save the mind map | N/A | The mind map is saved. | Same as expected result | Pass |
| T7j | Verify the saved state if a mind map is opened | N/A | The previous saved mind map is restored. | Same as expected result | Pass |
| T7k | Open a PDF in the PDF viewer | N/A | The selected PDF is rendered in the viewer. | Same as expected result | Pass |
| T7l | Drag a selected to the mind map | A sentence in the pfd. | The selected text appears as a node in mind map. | Same as expected result | Pass |

*Table 5-7 Test Case of Create Mind Map*

# CHAPTER 6 CONCLUSION

## 6.1 Project Review, Discussion, Conclusion

In conclusion, Smart Citation Manager is a desktop-based reference management system that allows user easy to create, search, manage their references, and create the bibliography and citation in the format required for a particular publication. Besides, Smart Citation Manager will be embedded with mind mapping tool and information extraction algorithm to empower the user during their research process.

With the help of this proposed system, the user able to store and manage their list of references for research papers. The user can easily create citation and bibliographies formatted to a citation style. In addition, Smart Citation Manager does provide PDF viewer that allow the user to open PDF document inside the system.

Besides, Smart Citation Manager allow the user to draft their paper in the mind map. Its embedded PDF viewer allow the user to capture the sentences they interested and add to the mind map while they are reading PDF documents. The user can keep track of their ideas, source of the ideas by creating link to the reference item stored, notes extracted from PDF, the related PDF document, and easily include them when mapping the structure of their paper. Any PDF document can be directly opened by the links embedded with text in the mind map. After finishing the mind map, it can be exported to text file or document file that allow user to further elaborate.

## 6.2 Novelties and Contribution of the Project has achieved

1. A reference management module for managing and creating citation and reference.
2. A document viewer for rendering and editing PDF document.
3. A mind mapping technique for drafting research ideas.
4. A text extraction algorithm is employed to facilitate retrieving text from PDF document.

## 6.3 Future Work

Currently, Smart Citation Manager is aim to enhance the research planning process by integrating the mind map and text extraction techniques inside the proposed system. In the future development, Smart Citation Manager will be improved to allow the user cite the ideas inside a mind map. A user can either insert citation from previous stored item or create it instantly. After the user finish their mind map, they able to generate a list of reference that associated with the cited item. By doing this, the user able to keep track the source of idea easily while drafting their research work.

Besides, metadata extraction from PDF document feature could be added into Smart Citation Manager. Once the user import PDF document from the local storage, the system able to identify and extract the title, author name and publication year from the PDF article. This can be accomplished by using Selenium library. Firstly, the text from PDF will be extracted and to verify whether it is the title of the article by using google search engine. When there is certain words like “cited by” are found, the text used for searching can be identified as the title of article. Selenium library is then used to scrap the name of authors, publication year by locating the CSS selector.

# REFERENCE

Academy, E., 2017. *ReadCube: Enhancing Scientific Discovery.* [Online]
Available at: https://www.enago.com/academy/readcube-enhancing-scientific-discovery/
[Accessed 22 11 2017].

Anon., 2015. *Citation Style Language:locales.* [Online]
Available at: https://github.com/citation-style-language/locales/blob/master/locales-ko-KR.xml
[Accessed 15 March 2019].

Anon., 2017. *Citation Style Language: style.* [Online]
Available at: https://github.com/citation-style-language/styles/blob/master/harvard-cite-them-right.csl
[Accessed 16 March 2019].

Anon., 2018. *Citation Style Language (CSL) Editor: Making Citation Easier.* [Online]
Available at: https://www.enago.com/academy/citation-style-language-csl-editor-making-citation-easier/
[Accessed 12 August 2018].

Docear, n.d. *Details & Features.* [Online]
Available at:
http://www.docear.org/software/details/#b_Docear8217s_8220Literature_Suite8221_concept_Drafting_papers_theses
[Accessed 21 11 2017].

EndNote, n.d. *Getting Started Guide.* [Online]
Available at: http://endnote.com/training/guide/mac#tab-3
[Accessed 20 11 2017].

Fenner, M., Scheliga, K. & Bartling, S., n.d. *Reference Management.* [Online]
Available at: http://book.openingscience.org/tools/reference_management.html
[Accessed 23 11 2017].

Fitzpatrick, J., 2009. *Mendeley Manages Your Documents on Your Desktop and in the Cloud.* [Online]
Available at: https://lifehacker.com/5334254/mendeley-manages-your-documents-on-your-desktop-and-in-the-cloud
[Accessed 25 11 2017].

Gilmour, R. & Cobus-Kuo, L., 2011. *Reference Management Software: a Comparative Analysis of Four Products.* [Online]
Available at: http://www.istl.org/11-summer/refereed2.html?a%5C_aid=3598aabf
[Accessed 20 11 2017].

Hensley, M. K., n.d. A comparison of citation management software. *Citation management software: Features and Futures,* p. 4.

Innovativearchitects, n.d. *The Seven Phases of the System-Development Life Cycle.* [Online]
Available at: view-source:https://www.innovativearchitects.com/KnowledgeCenter/basic-IT-systems/system-development-life-cycle.aspx
[Accessed 17 March 2019].

Kirby, J., 2017. *Review: Docear.* [Online]
Available at: https://publish.illinois.edu/commonsknowledge/2017/08/31/review-docear/
[Accessed 31 8 2017].

LLC, P., 2008. *ProQuest Acquires RefWorks.* [Online]
Available at: https://www.proquest.com/about/news/2008/ProQuest-Acquires-RefWorks.html
[Accessed 22 July 2018].

Lorenzetti, D. L. & Ghali, W. A., 2013. *Reference management software for systematic reviews and meta-analyses: an exploration of usage and usability.* [Online]
Available at: https://bmcmedresmethodol.biomedcentral.com/articles/10.1186/1471-2288-13-141
[Accessed 23 11 2017].

Maps, D. —. L. M. a. W. w. M., 2012. *Organizing Creativity.* [Online]
Available at: http://www.organizingcreativity.com/2012/02/docear-literature-management-and-writing-with-mind-maps/
[Accessed 24 11 2017].

Meredith, S., 2013. EndNote. *Critical review of referencing software when used with OSCOLA*, 4(1).

Meredith, S., 2013. Zotero. *Critical review of referencing software when used with OSCOLA*, 4(1).

Mohamed, S., 2012. *Software Development Life Cycle Models and Methodologies.* [Online]
Available at: https://melsatar.blog/2012/03/15/software-development-life-cycle-models-and-methodologies/
[Accessed 17 March 2019].

Rintzr M, Z., n.d. *Primer-An Introduction to CSL.* [Online]
Available at: http://docs.citationstyles.org/en/stable/primer.html#cs-info-element
[Accessed 16 March 2019].

Rouse, M., 2005. *Prototyping Model.* [Online]
Available at: https://searchcio.techtarget.com/definition/Prototyping-Model
[Accessed 18 March 2019].

Steele, S. E., 2008. *Bibliographic Citation Management Software as a Tool for Building Knowledge*, p. 1.

Steele, S. E., 2008. Bibliographic Citation Management Software as a Tool for Building Knowledge. *Journal of Wound Ostomy & Continence Nursing*, 35(5), pp. 463-468.

Thakur, D., n.d. *Prototyping Model in Software Engineering.* [Online]
Available at: http://ecomputernotes.com/software-engineering/explain-prototyping-model
[Accessed 18 March 2019].

Warling, B., 1992. EndNote plus: enhanced reference database and bibliography maker. *Journal of Chemical Information and Computer Sciences*, pp. 755-756.

Wikipedia, 2017. *ReadCube.* [Online]
Available at: https://en.wikipedia.org/wiki/ReadCube
[Accessed 23 11 2017].

Zaugg, H., West, R. E., Tateishi, I. & Randall, D. L., 2011. Mendeley Features. *Mendeley: Creating*, p. 33.

Zelle, R. M., 2016. *Primer — An Introduction to CSL.* [Online]
Available at: http://docs.citationstyles.org/en/stable/specification.html#introduction
[Accessed 12 August 2018].

Zotero, 2017. *About.* [Online]
Available at: https://www.zotero.org/about/
[Accessed 26 11 2017].

# APPENDIX A TYPES

- article
- article-magazine
- article-newspaper
- article-journal
- bill
- book
- broadcast
- chapter
- dataset
- entry
- entry-dictionary
- entry-encyclopedia
- figure
- graphic
- interview
- legislation
- legal_case
- manuscript
- map
- motion_picture
- musical_score
- pamphlet
- paper-conference
- patent
- post
- post-weblog
- personal_communication
- report
- review
- review-book
- song
- speech
- thesis
- treaty
- webpage

# APPENDIX B ORDINARY FIELD

- abstract

Abstract of the item (e.g. the abstract of a journal article)

- annote

Reader's notes about the item content

- archive

Archive storing the item

- archive_location

Storage location within an archive (e.g. a box and folder number)

- archive-place

Geographic location of the archive

- authority

Issuing or judicial authority (e.g. "USPTO" for a patent, "Fairfax Circuit Court" for a legal case)

- call-number

Call number (to locate the item in a library)

- citation-label

Label identifying the item in in-text citations of label styles (e.g. "Ferr78"). May be assigned by the CSL processor based on item metadata.

- citation-number

Index (starting at 1) of the cited reference in the bibliography (generated by the CSL processor)

- collection-title

Title of the collection holding the item (e.g. the series title for a book)

- container-title

Title of the container holding the item (e.g. the book title for a book chapter, the journal title for a journal article)

- container-title-short

Short/abbreviated form of "container-title" (also accessible through the "short" form of the "container-title" variable)

- dimensions

Physical (e.g. size) or temporal (e.g. running time) dimensions of the item

- DOI

Digital Object Identifier (e.g. "10.1128/AEM.02591-07")

- event

Name of the related event (e.g. the conference name when citing a conference paper)

- event-place

Geographic location of the related event (e.g. "Amsterdam, the Netherlands")

- first-reference-note-number

Number of a preceding note containing the first reference to the item. Assigned by the CSL processor. The variable holds no value for non-note-based styles, or when the item hasn't been cited in any preceding notes.

- genre

Class, type or genre of the item (e.g. "adventure" for an adventure movie, "PhD dissertation" for a PhD thesis)

- ISBN

International Standard Book Number

- ISSN

International Standard Serial Number

- jurisdiction

Geographic scope of relevance (e.g. "US" for a US patent)

- keyword

Keyword or tag(s) attached to the item

- locator

A cite-specific pin pointer within the item (e.g. a page number within a book, or a volume in a multi-volume work). Must be accompanied in the input data by a label indicating the locator type (see the Locators term list), which determines which term is rendered by cs:label when the "locator" variable is selected.

- medium

Medium description (e.g. “CD”, “DVD”, etc.)

- note

(Short) inline note giving additional item details (e.g. a concise summary or commentary)

- original-publisher

Original publisher, for items that have been republished by a different publisher

- original-publisher-place

Geographic location of the original publisher (e.g. “London, UK”)

- original-title

Title of the original version (e.g. “Война и мир”, the untranslated Russian title of “War and Peace”)

- page

Range of pages the item (e.g. a journal article) covers in a container (e.g. a journal issue)

- page-first

First page of the range of pages the item (e.g. a journal article) covers in a container (e.g. a journal issue)

- PMCID

PubMed Central reference number

- PMID

PubMed reference number

- publisher

Publisher

- publisher-place

Geographic location of the publisher

- references

Resources related to the procedural history of a legal case

- reviewed-title

Title of the item reviewed by the current item

- scale

Scale of e.g. a map

- section

Container section holding the item (e.g. “politics” for a newspaper article)

- source

From whence the item originates (e.g. a library catalog or database)

- status

(Publication) status of the item (e.g. “forthcoming”)

- title

Primary title of the item

- title-short

Short/abbreviated form of “title” (also accessible through the “short” form of the “title” variable)

- URL

Uniform Resource Locator (e.g. “http://aem.asm.org/cgi/content/full/74/9/2766”)

- version

Version of the item (e.g. “2.0.9” for a software program)

- year-suffix

Disambiguating year suffix in author-date styles (e.g. “a” in “Doe, 1999a”)

# APPENDIX C NAME FIELD

- author

Author

- collection-editor

Editor of the collection holding the item (e.g. the series editor for a book)

- composer

Composer (e.g. of a musical score)

- container-author

Author of the container holding the item (e.g. the book author for a book chapter)

- director

Director (e.g. of a film)

- editor

Editor

- editorial-director

Managing editor (“Directeur de la Publication” in French)

- illustrator

Illustrator (e.g. of a children’s book)

- interviewer

Interviewer (e.g. of an interview)

- original-author
- recipient

Recipient (e.g. of a letter)

- reviewed-author

Author of the item reviewed by the current item

- translator

Translator

# APPENDIX D DATE FIELD

- accessed

Date the item has been accessed

- container
- event-date

Date the related event took place

- issued

Date the item was issued/published

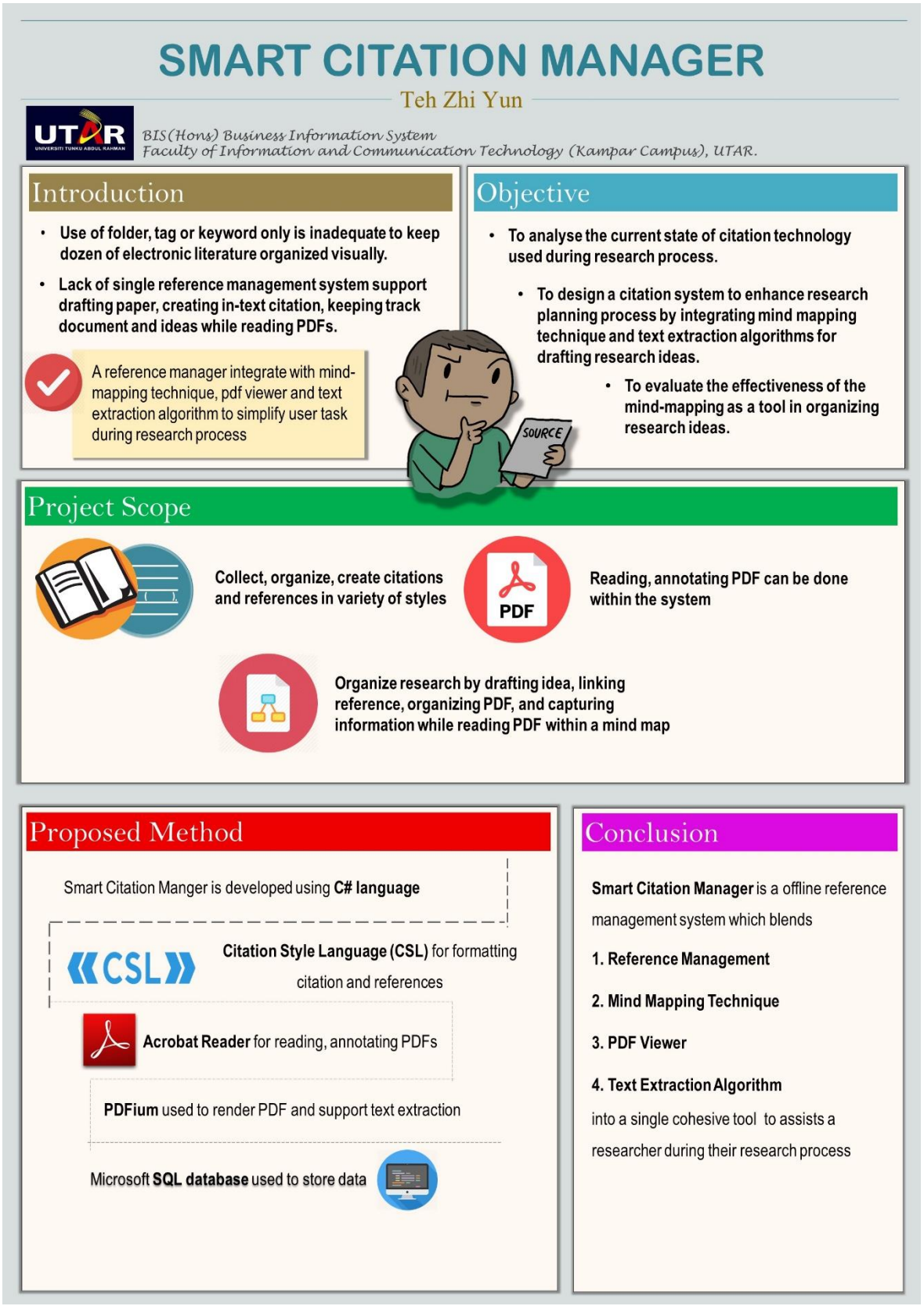
- original-date

(Issue) date of the original version

- submitted

Date the item (e.g. a manuscript) has been submitted for publication

# POSTER


SMART CITATION MANAGER
Teh Zhi Yun
BIS(Hons) Business Information System
Faculty of Information and Communication Technology (Kampar Campus), UTAR.
Introduction
Use of folder, tag or keyword only is inadequate to keep dozen of electronic literature organized visually.
Lack of single reference management system support drafting paper, creating in-text citation, keeping track document and ideas while reading PDFs.
A reference manager integrate with mind-mapping technique, pdf viewer and text extraction algorithm to simplify user task during research process
Objective
To analyse the current state of citation technology used during research process.
To design a citation system to enhance research planning process by integrating mind mapping technique and text extraction algorithms for drafting research ideas.
To evaluate the effectiveness of the mind-mapping as a tool in organizing research ideas.
SOURCE
Project Scope
Collect, organize, create citations and references in variety of styles
PDF
Reading, annotating PDF can be done within the system
Organize research by drafting idea, linking reference, organizing PDF, and capturing information while reading PDF within a mind map
Proposed Method
Smart Citation Manger is developed using C# language
CSL
Citation Style Language (CSL) for formatting citation and references
Acrobat Reader for reading, annotating PDFs
PDFium used to render PDF and support text extraction
Microsoft SQL database used to store data
Conclusion
Smart Citation Manager is a offline reference management system which blends
1. Reference Management
2. Mind Mapping Technique
3. PDF Viewer
4. Text Extraction Algorithm
into a single cohesive tool to assists a researcher during their research process

# PLAGIARISM CHECK RESULT

Smart Citation Manager (FYP2)

ORIGINALITY REPORT

| 14% | 12% | 4% | % |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | eprints.utm.my<br>Internet Source | 2% |
| 2 | eprints.utar.edu.my<br>Internet Source | 2% |
| 3 | www.cse.dmu.ac.uk<br>Internet Source | 1% |
| 4 | www.docear.org<br>Internet Source | 1% |
| 5 | docs.citationstyles.org<br>Internet Source | 1% |
| 6 | blogs.nature.com<br>Internet Source | 1% |
| 7 | www.slideshare.net<br>Internet Source | 1% |
| 8 | dspace.up.edu.ps<br>Internet Source | <1% |
| 9 | eprints.uny.ac.id<br>Internet Source | <1% |

| | | |
|---|---|---|
| 10 | Joeran Beel, Bela Gipp, Stefan Langer, Marcel Genzmehr. "Docear", Proceeding of the 11th annual international ACM/IEEE joint conference on Digital libraries - JCDL '11, 2011<br>Publication | <1% |
| 11 | link.springer.com<br>Internet Source | <1% |
| 12 | en.wikipedia.org<br>Internet Source | <1% |
| 13 | profesor-for-all.blogspot.com<br>Internet Source | <1% |
| 14 | Alex Hodgson, Lucas Schlager. "Closing the PDF gap: ReadCube's experiments in reader-focused design", Learned Publishing, 2017<br>Publication | <1% |
| 15 | digitalcommons.fiu.edu<br>Internet Source | <1% |
| 16 | Susan E. Steele. "Bibliographic Citation Management Software as a Tool for Building Knowledge :", Journal of Wound Ostomy and Continence Nursing, 09/2008<br>Publication | <1% |
| 17 | Peter Fernandez. "Zotero: information management software 2.0", Library Hi Tech News, 2011<br>Publication | <1% |

| | | |
|---|---|---|
| 18 | core.ac.uk<br>Internet Source | <1% |
| 19 | digitalcommons.lasalle.edu<br>Internet Source | <1% |
| 20 | forum.susana.org<br>Internet Source | <1% |
| 21 | labs.library.gvsu.edu<br>Internet Source | <1% |
| 22 | blog.bibsonomy.org<br>Internet Source | <1% |
| 23 | cdn-lnx1.nwu.ac.za<br>Internet Source | <1% |
| 24 | www.rivier.edu<br>Internet Source | <1% |
| 25 | www.opensourcebrain.org<br>Internet Source | <1% |
| 26 | www.emotiv.com<br>Internet Source | <1% |
| 27 | www.harptec.com<br>Internet Source | <1% |
| 28 | newton.cs.concordia.ca<br>Internet Source | <1% |
| 29 | www.knowdiv.org<br>Internet Source | <1% |

| | | |
|---|---|---|
| 30 | www.casellausa.com<br>Internet Source | <1% |
| 31 | neobots.blogspot.com<br>Internet Source | <1% |
| 32 | www.educba.com<br>Internet Source | <1% |
| 33 | www.intota.com<br>Internet Source | <1% |
| 34 | www.startertutorials.com<br>Internet Source | <1% |
| 35 | Pradeeka Seneviratne. "Chapter 3 Setting Up the Software Development Environment", Springer Nature, 2019<br>Publication | <1% |

| | | | |
|---|---|---|---|
| Exclude quotes | Off | Exclude matches | Off |
| Exclude bibliography | On | | |

Plagiarism Check Result

| Universiti Tunku Abdul Rahman | | | |
|---|---|---|---|
| **Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)** | | | |
| Form Number: FM-IAD-005 | Rev No.: 0 | Effective Date: 01/10/2013 | Page No.: 1of 1 |

**FACULTY OF INFORMATION AND COMMUNICATION**

| | |
|---|---|
| **Full Name(s) of Candidate(s)** | Teh Zhi Yun |
| **ID Number(s)** | 15ACB04334 |
| **Programme / Course** | Business Information System |
| **Title of Final Year Project** | Smart Citation Manager |

| **Similarity** | **Supervisor's Comments**<br>**(Compulsory if parameters of originality exceeds the limits approved by UTAR)** |
|---|---|
| **Overall similarity index: _____ %**<br>**Similarity by source**<br>Internet Sources: _____ %<br>Publications: _____ %<br>Student Papers: _____ % | |
| **Number of individual sources listed** of more than 3% similarity: _____ | |

**Parameters of originality required and limits approved by UTAR are as follows:**

**(i) Overall similarity index is 20% and below, and**

**(ii) Matching of individual sources listed must be less than 3% each, and**

**(iii) Matching texts in continuous block must not exceed 8 words**

Note: Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

***Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.***

Signature of Supervisor

Name: _____

Date: _____

Signature of Co-Supervisor

Name: _____

Date: _____

Check List

# UNIVERSITI TUNKU ABDUL RAHMAN

## FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

**CHECKLIST FOR FYP2 THESIS SUBMISSION**

| | |
|---|---|
| Student Id | 15ACB04334 |
| Student Name | Teh Zhi Yun |
| Supervisor Name | Mr Yong Tien Fui |

| TICK (√) | DOCUMENT ITEMS<br>Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item. |
|---|---|
| | Title Page |
| | Signed form of the Declaration of Originality |
| | Abstract |
| | Table of Contents |
| | List of Figures (if applicable) |
| | List of Tables (if applicable) |
| | List of Symbols (if applicable) |
| | List of Abbreviations (if applicable) |
| | Chapters / Content |
| | Bibliography (or References) |
| | All references in bibliography are cited in the thesis, especially in the chapter of literature review |
| | Appendices (if applicable) |
| | Poster |
| | Signed Turnitin Report (Plagiarism Check Result – Form Number: FM-IAD-005) |

*Include this form (checklist) in the thesis (Bind together as the last page)

| | |
|---|---|
| I, the author, have checked and confirmed all the items listed in the table are included in my report.<br><br>_____<br>(Signature of Student)<br>Date: | Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction.<br><br>_____<br>(Signature of Supervisor)<br>Date: |