

# Specification and Proposed Design

<b>I. Summary of Proposal.....</b>	<b>2</b>
<b>1.1 Introduction .....</b>	<b>2</b>
<b>1.2 Summary of Current Research and Analyses .....</b>	<b>2</b>
1.2.1 Technology Principle .....	2
1.2.2 Framework Selection .....	3
1.2.3 Library.....	3
<b>1.3 Software Development Life Cycle .....</b>	<b>4</b>
<b>1.4 Project Constraints, Assumptions and Risks .....</b>	<b>4</b>
<b>1.5 Features and Software Specifications .....</b>	<b>5</b>
<b>II. Design.....</b>	<b>6</b>
<b>2.1 Application Design Documentation .....</b>	<b>6</b>
<b>2.2 Deliverables .....</b>	<b>6</b>
<b>2.3 Criteria for Evaluating the Success of a System.....</b>	<b>7</b>
<b>2.4 How to Evaluate These Criteria .....</b>	<b>7</b>
<b>2.5 Who Will Be Involved in the Evaluation .....</b>	<b>7</b>
<b>2.6 Conclusion that Expect from the Evaluation .....</b>	<b>7</b>
<b>III. Data Required.....</b>	<b>8</b>
<b>IV. Plan .....</b>	<b>8</b>
<b>References .....</b>	<b>9</b>

# I. Summary of Proposal

## 1.1 Introduction

In the process of scientific peer review, reviewers are required to generate a plain text review for a given PDF document instead of reading and taking notes on a printed one. However, the casual distribution of annotations on different pages brings troubles for reviewers to reread them sequentially and conveniently. Consequently, they have to laboriously type annotations to get a plain text document.

It's a conversation about content that anchors at the locus of the content. In the traditional method, the reviewer will create another document about the annotations after finishing reviewing. We can simplify and enhance our access to detailed information to enable reviewers to find the most helpful information.

Therefore, this project aims at implementing an easy-to-use, interactive web application, especially for scientific peer review. Advantages of this application are listed below.

1. It has the features of a basic PDF reader;
2. The specific position will be attached to each annotation;
3. Recognize handwritten/plain text notes to generate a plain text;
4. Automatically integrate all comments into a plain text file.

To maximize productivity, this project will produce a Web application. Meanwhile, cross-platform technology is used to facilitate the adaptation in different platforms and devices.

## 1.2 Summary of Current Research and Analyses

### 1.2.1 Technology Principle

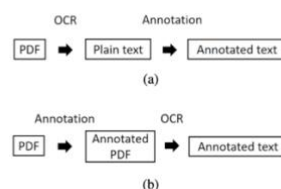


Fig. 1- General PDF Annotations

A recent research (Shindo, et al., 2018) has traced the history of general PDF annotations. As shown in Fig 1, there are two types of annotation procedures which

create annotated text from the PDF files: (a) convert the PDF to plain text or HTML and then annotate it by using a text annotation tool; (b) annotate the PDF file directly and then convert the annotated PDF file to plain text format.

If an annotated plain-text is eventually necessary, (b) still has two benefits. First, PDF is usually much more readable than plain text due to its better paragraph structure. Second, the annotations will become insensitive to OCR errors. We can switch among the OCR tools to convert the PDF to the plain text without modifying annotations.



Fig. 2 - A Struct of an Annotation Tool

In the GitHub community, many text document annotation tools, such as “PDF element”, “PDF Export”, “Nitro Pro” and “PDF Reader Pro”, have been developed so far (Kuchta et al., 2018). Additionally, according to *On the Electronic Documents: University, Finding, and Localizing Inconsistency Bugs in PDF Readers and Files* (Kuchta et al., 2018), all these tools basically follow method (b) as shown in fig. 1.

Their principle, almost all, is using “PDF.js” this library or based on it for secondary packaging. Using the Canvas technology of HTML5 provides a PDF client with a rendering method independent from the local system. However, the canvas element itself has no drawing capability; all the drawings must be done with JavaScript. Based on Fhala (2014), the basic principle is to render the PDF page as a fixed size bitmap, which is then loaded into the canvas element.

Nevertheless, these mentioned tools do not specially output annotations in plain text format. Thus, they are not suitable for the scenario proposed by the present project.

### 1.2.2 Framework Selection

According to Timanovskyi and Plechawska-Wójcik (2020), the technology selection depends on different conditions, namely, the project scale, the developer’s knowledge, previous experience and so on. Therefore, though Angular, React, and Vue are similar, Vue is much more appropriate for this project for it applies MVVM architecture to separate the front-end and back-end strictly and has higher scalability.

### 1.2.3 Library

Based on Timanovskyi and Plechawska-Wójcik (2020), a comparison is conducted as

shown in Table 1, which indicates that vue.pdf is much more suitable for this project.

Table 1: Comparison of Different Tools

	Advantage	Disadvantage	Principle
Pdf.js	Good compatibility	Nonadaptation to the framework	JavaScript
iframe/ object/ embed	Easy-use	Change Dom, Waste of CPU	H5 label
Vue show pdf	Contains basic functions	Unable to customize components	Pdf.js
vue-pdf	Customizable	Fixed width to height ratio, need to modify the source code	pdf.js

### 1.3 Software Development Life Cycle

Waterfall Model will be used for this application. As shown in Fig. 3, there are five phases based on this model (Melonfire, 2006). Table 2 gives information about the particular content for each phase of this current project.

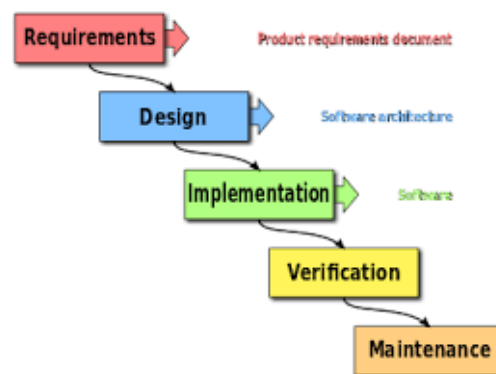


Fig. 3 – Five Phases of Waterfall Model.

Table 2: Particular Content of Each Phase Based on Waterfall Model.

<b>Requirement Analysis</b>	Based on project requirements;
<b>Design</b>	Perfect the design part based on the requirements;
<b>Implementation</b>	Produce fully functional software;
<b>Testing and Installation</b>	Test to confirm that the system work meets the requirements;
<b>Maintenance</b>	Maintain the system when new requirements arise.

### 1.4 Project Constraints, Assumptions and Risks

Two **Constraints** of this project are listed below.

- Due to the COVID-19, most meetings are held online.
- The project cannot be publicly tested, so there may be not adequate feedbacks.

The **Assumptions** of this project are listed below.

- a) Ensure the completion of project objectives and tasks;
- b) Send the design to the supervisor as soon as it is done;
- c) Prepare resources and development tools for this project;
- d) Allocate project schedule properly;
- e) There is no need to backtrack after the project is completed.

There may be some **Risks** arising during the project implementation; thus, corresponding strategies are considered to mitigate risks or minimize the impact.

1) Time shortage

Project impact: the product cannot be delivered on time.

Actions to mitigate: apply schedule charts, such as Gantt chart.

Actions to minimize impact: response the user requirements of lower priority lastly when there is not enough time.

2) Project advisor is unavailable when needed

Project impact: the project may not be on track.

Actions to mitigate: establish meeting time with the advisor in advance so that he can date it.

Actions to minimize impact: make another appointment.

3) Accidental loss of archive

Project impact: plan projects again.

Actions to mitigate: backup all the time.

Actions to minimize impact: continue with the latest backup.

## 1.5 Features and Software Specifications

### a) Proposed Features

Upload file(local)
Directory
Bookmark
View (single page and double page)
Zoom in and zoom out
Downloadable
Enter an annotation
Recognize text (plain or handwritten)
Annotation point to the location of the text of the file
Merge all annotation and output (Content and position
Output all annotation as plain text files

## b) Software Specifications

Option	Software Requirement
Language	JavaScript
Formwork	VUE
IDE	VS Code
Development Environment	Mac OS big sur
Client Browser	Chrome Stable

## c) Technical Specifications

Vue
Axios
Vue-cli
Webpack
element-ui
PWA

# II. Design

## 2.1 Application Design Documentation

Designs	Design descriptions
Use case diagram	Characters, use cases and their relationships.
Sequence diagram	Clarifies the interacting process between objects and what will happen at a specific time in the process of system execution.
Class diagram	1. Inheritance of page components. 2. Inheritance of function components.
Web data flow diagram	Maps out the information flow of each process or system.
Database table relationship	Describes the structure of stored data.
Interface design	Overall design interaction, operation logic, and beautiful interface of the software.
Test design	Test plan and test results.

## 2.2 Deliverables

a) Application design documentation
b) Fully functional web PDF reader application
c) User manual and screenshots
d) Testing documentation
e) Final report

## 2.3 Criteria for Evaluating the Success of a System

Evaluation design

Two parts need to be taken into account to evaluate the success-degree of this project. First, whether the outcome match up with the project aim. Second, whether the project timeline and logbook are managed as planned.

## 2.4 How to Evaluate These Criteria

### Part 1

1. Whether the requirements are met;
2. Whether it is in line with the working scenario, considers the usage scenario and user mode.
3.Stability;
4. Efficient: whether it can display information and deal with problems simultaneously, for example, display pictures, send/ receive information and optimize the network at once;
5. Clear logic: Whether the function is simple and easy to use;
6. The UI design;
7. Platform support and compatibility;
8. Whether it is extensible.

### Part 2

Event plan management
Schedule management
Event handling capability
Document management

## 2.5 Who Will Be Involved in the Evaluation

1. Supervisor1
2. Supervisor2
3. Myself

## 2.6 Conclusion that Expect from the Evaluation

Review the project	What technology has been learned through this project; What extra features would be added if the project could be extended;
Professional and Academic Expertise	the value of professional and academic expertise which is earned throughout the learning process and carrying out this project;

limitation	Shortcomings of the current project;
Future study	Next steps in the future.

### III. Data Required

#### Ethical Statements

This project complies with the British Computer Society (BCS, 2021).

Data required and sources. This project requires a PDF file for testing from “data.uk.gov” The data is released by the UK government and is free to download and use by the public. Only a PDF file is used as a test in this project, and no data is analyzed.

Ethical use of data. Synthetic data or real non-human data, real human data will not be used in this project. Therefore, it does not require research ethics approval.

In Technology, if any third-party library is used, their open-source licenses will be complied.

### IV. Plan

The project’s overall schedule is based on the time standards of COMP702, and the details are shown in the Gantt Chart.

Important Date	Activity
<b>Monday 7 June 2021</b>	MSc projects officially begin ( <b>Week 1</b> ).
Weeks 1 and 2	Background reading and literature review.
Weeks 3, 4 and 5	Development of project design.
<b>Friday 9 July 2021 (5:00pm)</b> (to turn in Specification and Design report)	<b>Specification and Proposed Design</b> report (end of <b>Week 5</b> ). Oral presentation (by end of <b>Week 6 (July 12-16)</b> ).
Weeks 6, 7, 8, 9, 10, 11, and 12	Software implementation and testing.
<b>Friday 27 August 2021 (5:00pm)</b>	<b>Final presentation</b> + software demo when appropriate (by end of <b>Week 13 (Aug 30 - Sep 03)</b> ).
Weeks 11, 12, 13, and 14	Software experimentation and analysis of results.
Weeks 13, 14, and 15	Write-up of dissertation.
<b>Friday 24 September 2021 (5:00pm) (noon)</b>	<b>Dissertation</b> hand in ( <b>Friday in Week 16, Firm deadline</b> ).



No.	stage	Start	prepare	finish	Actually finish	cost days
1	Generate ideas for report	2021/06/01	3	2021/06/03	2021/06/03	3 ✓
2	meeting	2021/06/02	1	2021/06/02	2021/06/02	1 ✓
3	background research	2021/06/04	7	2021/06/10	2021/06/15	12 ✗
4	meeting	2021/06/10	1	2021/06/10	2021/06/10	1 ✓
5	Ethical Statements	2021/06/11	1	2021/06/11	2021/06/11	1 ✓
6	Methodology	2021/06/11	1	2021/06/11	2021/06/11	1 ✓
7	Deliverables	2021/06/12	1	2021/06/12	2021/06/12	1 ✓
8	Features and Software Specifications	2021/06/15	2	2021/06/16	2021/06/17	3 ✗
9	Evaluate existing PDF annotation method	2021/06/18	7	2021/06/24	2021/06/23	6 ★
10	Specification and Proposed Design report	2021/07/12	3	2021/07/14	2021/07/14	3 ✓
11	Demand analysis	2021/06/25	2	2021/06/26	2021/06/26	2 ✓
12	System Design Architecture	2021/06/27	7	2021/07/03	2021/07/08	12 ✗
13	UI design	2021/07/05	10	2021/07/14	2021/07/21	17 ✗
14	Implementation	2021/08/24	1	2021/08/24		
15	Testing					
16	Critical evaluation					

## References

1. Shindo, H., Munesada, Y. and Matsumoto, Y., 2018, May. PDFAnno: a web-based linguistic annotation tool for pdf documents. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
2. Kuchta, T. *et al.* (2018) 'On the correctness of electronic documents: studying, finding, and localizing inconsistency bugs in PDF readers and files', *Empirical Software Engineering: An International Journal*, 23(6), p. 3187. doi: 10.1007/s10664-018-9600-2.
3. Fhala, B. (2012) *HTML5 graphing and data visualization cookbook. learn how to create interactive HTML5 charts and graphs with canvas, JavaScript, and open source tools*. Packt Pub. Available at: <https://search-ebscohost-com.liverpool.idm.oclc.org/login.aspx?direct=true&db=cat00003a&AN=lvp.b3069881&site=eds-live&scope=site> (Accessed: 29 June 2021).
4. Yehor Timanovskyi and Małgorzata Plechawska-Wójcik (2020) 'Performance analysis of selected tools for building a Single Page Application', *Journal of Computer Sciences Institute*, 14. doi: 10.35784/jcsi.1581.
5. McCormick, M., 2012. Waterfall vs. Agile methodology. *MPCS*, N/A.
6. (2021) Bcs.org. Available at: <https://www.bcs.org/membership/become-a-member/bcs-code-of-conduct/> (Accessed: 28 June 2021).

APPENDIX 1



DECLARATION OF ACADEMIC INTEGRITY

NAME (Print)	NIHAO DONG
STUDENT NUMBER	2014 7 66 06
MODULE TITLE/CODE	comp 702
TITLE OF WORK	Msc project

*This form should be completed by the student and appended to any piece of work that is submitted for summative assessment.*

Students should familiarise themselves with Section 9 of the Code of Practice on Assessment and Appendix L of the University's Code of Practice on Assessment which provide the definitions of academic malpractice and the policies and procedures that apply to the investigation of alleged incidents.

Students found to have committed academic malpractice are liable to receive a mark of zero for the assessment or the module concerned. Unfair and dishonest academic practice will attract more severe penalties, including possible suspension or termination of studies.

STUDENT DECLARATION

I confirm that I have read and understood the University's Academic Integrity Policy.

I confirm that I have acted honestly, ethically and professionally in conduct leading to assessment for the programme of study.

I confirm that I have not copied material from another source nor committed plagiarism nor fabricated data when completing the attached piece of work. I confirm that I have not previously presented the work or part thereof for assessment for another University of Liverpool module. I confirm that I have not copied material from another source, nor colluded with any other student in the preparation and production of this work.

I confirm that I have not incorporated into this assignment material that has been submitted by me or any other person in support of a successful application for a degree of this or any other University or degree awarding body.

SIGNATURE..... NIHAO DONG.....  
 DATE..... 2 / 7 / 2021.....