

**ComiTranslate: Empowering Global Readership through Auto-Translated Comics and Manga**

**Goh Yee Fong**


**A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Software Engineering with Honours**

**Lee Kong Chian Faculty of Engineering and Science  
Universiti Tunku Abdul Rahman**

**May 2024**

**DECLARATION**

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature :   
\_\_\_\_\_

Name : Goh Yee Fong  
\_\_\_\_\_

ID No. : 2003255  
\_\_\_\_\_

Date : 17 May 2024  
\_\_\_\_\_

**APPROVAL FOR SUBMISSION**

I certify that this project report entitled “**ComiTranslate: Empowering Global Readership through Auto-Translated Comics and Manga**” was prepared by **Goh Yee Fong** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Science (Hons) Software Engineering with Honours at Universiti Tunku Abdul Rahman.

Approved by,

Signature : *Kai*

Supervisor : Dr Chia Kai Lin

Date : 17 May 2024

Signature : \_\_\_\_\_

Co-Supervisor : \_\_\_\_\_

Date : \_\_\_\_\_

The copyright of this report belongs to the author under the terms of the copyright Act 1987 as qualified by Intellectual Property Policy of Universiti Tunku Abdul Rahman. Due acknowledgement shall always be made of the use of any material contained in, or derived from, this report.

© 2024, Goh Yee Fong. All right reserved.

## ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Dr. Chia Kai Lin for her invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition, I would like to express my gratitude to my loving parents and friends who had helped and given me encouragement to pursue the development of this project.

## ABSTRACT

Due to the steep learning curve and time needed to translate the contents of the comic, readers are faced with significant challenges due to this language barrier. Conventional methods often include the laborious and time-consuming use of dictionaries or other language-learning aids, which can be discouraging. Therefore, this project aims to encourage readership of foreign literature by providing a solution which automates the translation process by the press of a button. The proposed system utilizes Python Image Library, Tesseract, Keras-OCR, Manga OCR, DeepL, and OpenCV to achieve the desired result as a solution. Kanban In Agile Methodology was practiced throughout the development phase as a way to manage the project's progress. The application detects all computer generated text, extracts them, translates, and then places the translated text back onto the image . Due to the application's superiority in its translation speed as compared to human translators and redrawers, it is able to make reading self-published works, which are often not translated by official sources, easier for readers as the application provides them the means of translating the comics. However, there are still flaws in its accuracy and improvements are planned for future iterations of this project. In conclusion, all objectives had been fulfilled through the development of this project.

## TABLE OF CONTENTS

DECLARATION .....	ii
APPROVAL FOR SUBMISSION .....	iii
ACKNOWLEDGEMENTS.....	v
ABSTRACT.....	vi
TABLE OF CONTENTS.....	vii
LIST OF TABLES .....	xi
LIST OF FIGURES .....	xii
LIST OF APPENDICES.....	xiii
<b>CHAPTER 1</b>	<b>1</b>
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 Introduction	1
1.2 Background of the Project	2
1.3 Problem Statement	3
1.4 Project Objectives	5
1.5 Project Solution	6
1.6 Project Approach	9
1.7 Project Scope	11
<b>CHAPTER 2</b>	<b>13</b>
<b>2 LITERATURE REVIEW</b>	<b>13</b>
2.1 Introduction	13
2.2 Review of Other Manga Translation Applications	13
2.2.1 ISMANGA Manga Translator	14
2.2.2 Scan Translator	17
2.3 Pillow	19
2.4 OpenCV	19
2.5 Keras-OCR	20
2.6 Review on Machine Translation Services	21

2.6.1 DeepL	21
2.6.2 Google Translate	21
2.6.3 Comparison of the Machine Translation Services	22
2.7 Review on OCR Tools	23
2.7.1 Tesseract	23
2.7.2 Amazon Textract	23
2.7.3 Google Document AI	24
2.7.4 Manga OCR	24
2.7.4.1 Comparison of the OCR Tools	25
2.7.4.2 Tesseract vs Manga OCR : Comparison on Japanese Text Accuracy	26
2.8 Summary of Literature Findings	27
<b>CHAPTER 3</b>	<b>28</b>
<b>3 METHODOLOGY AND WORKPLAN</b>	<b>28</b>
3.1 Introduction	28
3.2 Waterfall Model	28
3.3 Agile Methodology	30
3.3.1 Kanban In Agile	30
3.3.2 Scrum In Agile	31
3.4 Conclusion on Chosen Methodology	32
3.5 Proposed Workplan	33
3.5.1 Work Breakdown Structure	33
3.5.2 Gantt Chart	34
3.6 Development tools and technologies	35
3.6.1 Pillow	35
3.6.2 OpenCV	35
3.6.3 Keras-OCR	35
3.6.4 DeepL	35
3.6.5 Tesseract	36
3.6.6 Manga OCR	36
3.7 Summary of Project Methodology	36



<b>CHAPTER 4</b>	<b>38</b>
<b>4 PROJECT SPECIFICATIONS</b>	<b>38</b>
4.1 Introduction	38
4.2 Requirements Specification	38
4.2.1 Functional Requirements	38
4.2.2 Non-Functional Requirements	39
4.3 Use Case	40
4.3.1 Use Case Diagram	40
4.3.2 Use Case Description	41
4.4 User Interface	48
<b>CHAPTER 5</b>	<b>49</b>
<b>5 SYSTEM IMPLEMENTATION</b>	<b>49</b>
5.1 Overview	49
5.2 Image Translation Process	49
5.2.1 Image Import	49
5.2.2 Text Detection	50
5.2.3 Textbox Merging	51
5.2.4 Mask Drawing	52
5.2.5 Textbox Extraction	53
5.2.6 Text Extraction	54
5.2.7 Text Translation	55
5.2.8 Text Overlay	56
5.3 Reading Page Design	57
<b>CHAPTER 6</b>	<b>59</b>
<b>6 SYSTEM TESTING</b>	<b>59</b>
6.1 Introduction	59
6.2 Unit Testing	59
6.3 Integration Testing	60
6.4 Visual Testing	60

6.5 Usability Testing	61
<b>CHAPTER 7</b>	<b>64</b>
<b>7 CONCLUSION AND RECOMMENDATIONS</b>	<b>64</b>
7.1 Achievement	64
7.2 Limitations	65
7.3 Future Improvements	66
<b>REFERENCES</b>	<b>67</b>
<b>APPENDIX</b>	<b>69</b>

**LIST OF TABLES**

TABLE 2-1 THE COMPARISON BETWEEN DEEPL AND GOOGLE TRANSLATE	22
TABLE 2-2 THE COMPARISON BETWEEN DOCUMENT AI, TEXTRACT AND GOOGLE TESSERACT	25
TABLE 2-3 THE COMPARISON BETWEEN TESSERACT, AND MANGA OCR	26
TABLE 3-1 THE ADVANTAGES AND DISADVANTAGES OF WATERFALL MODEL	29
TABLE 3-2 THE ADVANTAGES AND DISADVANTAGES OF KANBAN IN AGILE	31
TABLE 3-3 THE ADVANTAGES AND DISADVANTAGES OF SCRUM IN AGILE	32
TABLE 3-4 THE COMPARISON BETWEEN KANBAN AND SCRUM	32
TABLE 6-1 UNIT TEST CASE	59
TABLE 6-2 INTEGRATION TEST CASE	60
TABLE 6-3 USABILITY TEST TABLE	61
TABLE 6-4 SATISFACTION SURVEY	62
TABLE 6-5 USABILITY TEST TABLE AVERAGE SCORE	62
TABLE 6-6 SATISFACTION SURVEY AVERAGE SCORE	63

## LIST OF FIGURES

FIGURE 1-1 PROPOSED SOLUTION WITH THE MODULES USED IN THE APPLICATION	6
FIGURE 1-2 KANBAN BOARD	9
FIGURE 2-1 ORIGINAL MANGA PAGE OF SAMPLE 1	14
FIGURE 2-2 ORIGINAL MANGA PAGE OF SAMPLE 2	15
FIGURE 2-3 ISMANGA TRANSLATED PAGE OF SAMPLE 1	15
FIGURE 2-4 ISMANGA TRANSLATED PAGE OF SAMPLE 2, ISMANGA STRUGGLES TO DETECT THE TEXTS	16
FIGURE 2-5 SCAN TRANSLATED PAGE OF SAMPLE 1	17
FIGURE 2-6 SCAN TRANSLATED PAGE OF SAMPLE 2	18
FIGURE 3-1 THE WATERFALL MODEL	28
FIGURE 3-2 THE KANBAN BOARD	30
FIGURE 3-3 GANTT CHART FOR PROJECT I	34
FIGURE 3-4 GANTT CHART FOR PROJECT II	34
FIGURE 3-5 SYSTEM ARCHITECTURE FOR THE PROPOSED PROJECT	37
FIGURE 4-1 USE CASE DIAGRAM	40
FIGURE 4-2 READING PAGE	48
FIGURE 5-1 IMPORTED IMAGE	49
FIGURE 5-2 IMPORTED IMAGE WITH BOUNDING BOX DRAWN TO VISUALIZE THE AREAS DETECTED BY KERAS-OCR	50
FIGURE 5-3 IMPORTED IMAGE WITH BOUNDING BOX DRAWN TO VISUALIZE THE RESULTANT MERGED BOXES	51
FIGURE 5-4 IMPORTED IMAGE WITH WHITE MASK DRAWN	52
FIGURE 5-5 EXTRACTED TEXT REGIONS FROM THE IMPORTED IMAGE	53
FIGURE 5-6 EXTRACTED TEXT FROM THE TEXT REGIONS	54
FIGURE 5-7 TRANSLATED TEXT	55
FIGURE 5-8 FINAL PRODUCT OF THE COMITRANSLATE APPLICATION	56
FIGURE 5-9 COMITRANSLATE READING PAGE	57
FIGURE 5-10 COMITRANSLATE READING PAGE DISPLAYING WARNING	58
FIGURE 6-1 NOISY IMAGE (LEFT SIDE) DENOISED IMAGE (RIGHT SIDE)	60

**LIST OF APPENDICES**

APPENDIX A : Sample panels to be used for the working prototype in FYP2	69
APPENDIX B : Supervisor and moderator comments on project plan	71

## CHAPTER 1

### INTRODUCTION

#### 1.1 Introduction

Comics and *manga* are captivating forms of visual storytelling that have delighted audiences with their unique blend of art and narrative. Comics feature dynamic panels and lively dialogue, spanning various genres from superhero adventures to thought-provoking graphic novels. *Manga*, from Japan, creates intricate stories through detailed art and emotive characters. Both mediums immerse readers in imaginative worlds, exploring complex subjects, relatable emotions, and fantastical adventures. However, a language barrier exists between the reader and the language used in foreign-language comics or *manga*. This language barrier poses a substantial difficulty for readers because of the high learning curve and the time required to translate the comic's contents. Traditional approaches frequently entail the time-consuming, arduous use of dictionaries or other language-learning tools, which may be frustrating and wasteful. Additionally, relying on external translation tools interferes with the immersive reading experience and makes it difficult for the reader to completely interact with the artwork and narrative. The problem statement emphasizes the need for a thorough and effective solution that makes translating comics easier, removes the learning curve, and makes it simple for viewers to access and comprehend comics written in other languages.

## 1.2 Background of the Project

In this day and age, especially with the digitization of media content on to the internet, more and more people are able to access a wide variety of materials for entertainment or for education. While highly accessible for many people, not all people from various countries of different languages can read it as comics come in various languages such as English, French, Swedish and many more and as for the case of *mangas* they are always written in Japanese. This means that readers will have to translate the contents in the comics or *mangas* by themselves if they wish to understand the contents in the comic or *manga*.

Alternatively, there are officially translated versions of comics and *mangas* in various languages for various readers in other countries, but this also comes at a cost as the localized versions of comics and *mangas* often cost 3 times as much as the original release version. Localized comics and *mangas* also take a long time to publish, this means that other fans of the series have to bear with the long wait time before they can enjoy the content they desire. In order to eliminate most of the downsides of localization of comics and *mangas*, a faster and cheaper solution is needed to resolve the issue.

### 1.3 Problem Statement

In this century, technology has been used to improve various things, such as raising human living standards or, in this case, producing smooth and convenient solutions for readers to overcome the language barrier in reading foreign comics and *mangas*. The solution needs to be easy to use and snappy in terms of its operations.

As mentioned once in the project background, localized reading materials come with higher costs and long wait times compared to the original material. Other than that, most of the content created by foreign artists remains to be untranslated as most publishing companies do not focus on independent work unless said work generates enough followers from readers. Still, there is no guarantee that the foreign works will be picked up by global publishers as there are still a lot of popular *mangas* that are still not been officially translated yet by the time of writing this report.

To address the issues of officially localized comics and *mangas*, there are fan translation groups available to translate the contents at a moment's notice. These fan translation groups are typically formed from a community of like-minded individuals. While cheaper and faster than the official translations, the quality of the translation may also vary depending on the groups which contributes to an unbalanced writing style between fan-translated versions on the same *manga* and comic. Also, this form of localization also runs a chance to be halted midway due to internal group fallout, groups losing interest or other personal reasons.

As readers without the ability to understand the language or the time to translate the comics or *mangas* of their liking, they are forced to rely on both mentioned parties to be able to enjoy the reading materials. Furthermore, the reader is ultimately stuck if both parties are out of the equation. As the reading materials remain untranslated, there will be fewer people who will know of its existence. As time passes by, the reading materials will fall into obscurity and thus become what is known as 'lost media'. Lost media are effectively media that has been completely lost in time or forgotten by the masses and this might impact the preservation of such content in general against the passage of time.



Thusly, an application with the function to translate comics and *mangas* has to be developed for readers to perform the translation work by themselves. Such applications will also help in generating and keeping reader interest in following the comic or *manga* of their choosing and this may also help in boosting the popularity of the series or franchise the comics and *mangas* are based on. To encourage the users to translate comics or *mangas*, the application will have to be autonomous in finding the words on the comics or *mangas* and output back the same page with the translated content in its place.

## 1.4 Project Objectives

This project is aimed to achieve the following objective stated below:

1. To reduce the time required to translate comics and *mangas*.
2. To aid non-skilled users in performing translation operations.
3. To translate foreign languages into the user's preferred language.
4. To develop an application that can be used to translate *mangas* and comics on the fly.

The developed application will increase the reader's interest in reading comics and *mangas* in foreign languages by allowing the operation to be done fast and without much manual input from the users.

The application will achieve this by automatically detecting the text on the uploaded images, converting the text images, and sending the converted text to a translation API (Application Programming Interface) (eg : DeepL, Google Translate).

## 1.5 Project Solution

This project aims to solve the problem of the fact that there are many more *mangas* and comics that remain to be untranslated. An application with an automated solution will be made as the solution to the problem. To achieve this goal, the application will use a back-end API translation service (eg: Deepl, Google Translate) to perform the translation work once the text on the image is converted into manipulatable text. Image manipulation tools and libraries such as PILLOW and OpenCV will be used to manipulate the image for the OCR (Optical Character Recognition) tool, Pyterreact for English text and Manga OCR for Japanese to detect the text image and convert it into manipulatable text. Even with all of the steps required to perform the task, this approach via the use of an application will still be faster than having the translation done the traditional way.

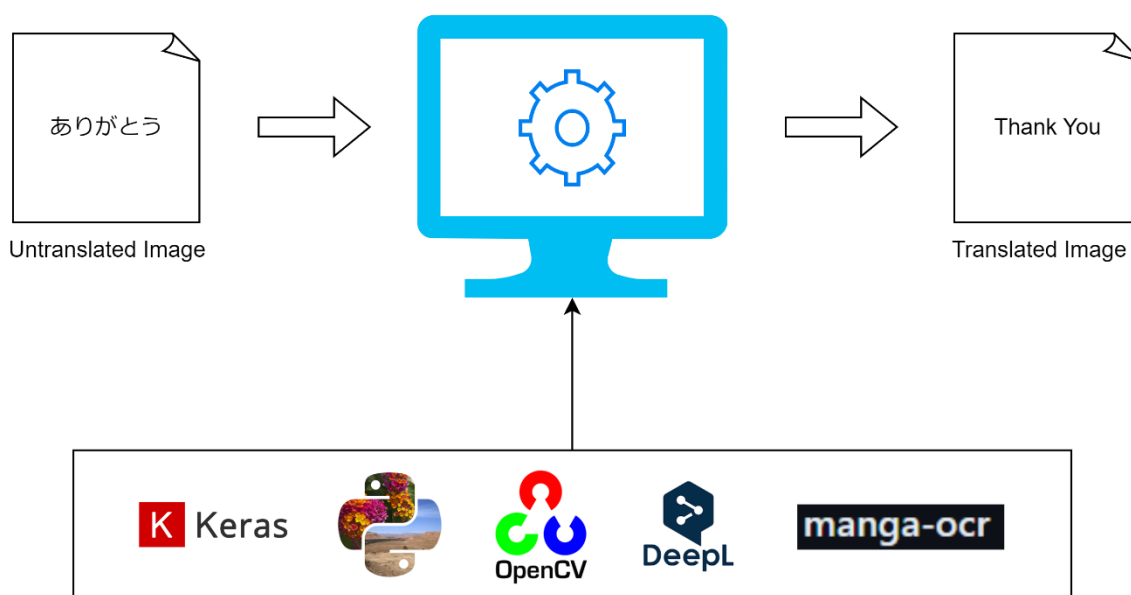


Figure 1-1 Proposed solution with the modules used in the application

The application in design usually won't require additional user input besides selecting the images of the comics and *mangas* to be translated. In order to fulfill this requirement, the use of python tool, Pillow will be used. Pillow is used to convert the image file for Manga OCR to process and perform some preliminary actions on clearing/touching up the image to make the image more easily readable for the OCR, Pillow will also be used to overlay the translated text onto the image.

OpenCV will be used as the component to import images as well as acting as the primary image pre-processing component, as well as to employ noise-reduction techniques to improve the accuracy of the OCR in text extraction if necessary, as some comics may contain background noise in the form of speckles, which may impair the OCR's ability to detect text accurately. Since some digital comics or *manga* may come with texts in varying colors, pre-processing techniques such as gray scaling and binarization of colors to which will be passed on to the OCR component. OpenCV will also be used to apply masks onto the image to prepare for the overlay of the translated texts back onto the place where the original text used to be.

Keras OCR will be used as the text detection component in this application. The purpose of this component would be to help in detecting text on the image to be further processed by OpenCV. Keras OCR by itself is powered with CRAFT detection which makes it easier to map out bounding boxes especially on comics and mangas in which groups of text are bunched together.

Pytesseract will be used as the OCR component in this application. The purpose of such a component is to extract text from the images given which will be used to pass onto the translation API which will be DEEPL.

Manga OCR will be used as the OCR component for translating Japanese text in this application. The purpose for a component with such a specific function is because of the complexity of how texts can come in various forms and styles in *mangas* as a way to exude the author's uniqueness and creativity which makes it hard for Pytesseract to detect the texts on the image.

The application will be using DEEPL for handling translation tasks as DEEPL is well-known for its excellent translation quality. It employs deep learning techniques, namely neural machine translation (NMT), to generate more accurate and natural-sounding translations than typical statistical machine translation systems. This achieves in providing much more accurate and high-quality translations.

## 1.6 Project Approach



Figure 1-2 Kanban board

The project will employ the use of the Kanban approach in the sense of agile principles built upon the philosophy of continuous development. This means that the project scope and the tasks will be presented in a visualized manner, the workflow of performing the project will be more on the preliminary side in the beginning phase to limit the work in progress and will be expanded on in the later stages upon approval, in the sense of pulling items out from the backlog of ideas to be improved in the later stages (Wiegand, 2018).

The Kanban Approach is separated into 3 different states in its most basic form, namely: Backlog, In Progress, and Done. The Backlog is where the tasks to be held are kept for later development in the project, The In-Progress state is where the currently undertaken tasks are held, while the Done state keeps the finished task. If the user feels the need to place tasks that he/she might not be confident might be done, the Waiting state can be used to indicate such tasks (Wiegand, 2018). This sort of task sorting can help to reduce the amount of task clutter the user has to deal with on a regular basis for the entire duration of the development.

As mentioned, the Kanban approach offers a great level of flexibility as the duration for the tasks is not mentioned, this may also help in reducing the mental strain on the user in the overall project cycle. The increased efficiency is also one of the pros this approach brings in reducing process wastes as the small or the beginning portion of the tasks are continually shown to the customer which allows for more opportunities to modify future iterations of the plan to the updated project requirements (Lynn, 2021).

For this project, Trello will be used as the base platform for any Kanban-based operations. This web application is chosen as it provides a minimal and user-friendly approach to the addition and deletion of tasks for free, albeit with limited functionality.

Additionally, the comic and *manga* panels that will be used to demonstrate the application will be acquired from X which was previously known as Twitter as to avoid stealing from official comic and *manga* publishers which are protected by anti-copyright policies.

## 1.7 Project Scope

With considerations regarding the nature of the final year project, the project scope for the system in development will have significantly narrowed requirements and functionality to focus on the university's other course modules. The system application will run on the Windows Platform on PC and is designed to focus on the function to detect text on the image, send the text to a translation API, receive translated text, and then overlay the text back onto the image. To achieve the desired outcome, the following functions will be used :

### **i. Image Import**

The user puts the images in the application to be loaded into the system to be translated. The user will have to place the images in series for the system to import the images into the system in the correct order. This can be done by naming the file from 1 to however pages the user requires.

### **ii. Text Detection and conversion**

The application will detect the text on the images automatically after performing some image alterations to make the text more easily detectable by the OCR used in the system, the system will display the effect of the alteration to the user to make sure that the image is calibrated to the state that makes text detection easy. Once the desired effect is reached, the system will convert the image into text.

### **iii. Translation**

The application will pass the text to an API for translation work to be done. This will be done by passing on the text as tokens to the API. The translated text will be passed back to the application.

### **iv. Text Overlay**

The application will lay the translated text onto the position where the text was originally detected on. This makes the translated image more easily readable for the user.



#### **v. Mask Drawing**

The application will draw a white box on the exact place where the texts are detected. This will create space for the translated text to be overlaid onto the final image.

The application is not without its limitations on aspects of which text the OCR can detect. One of the primary concern lies in the detection of sound effect texts, which contribute to the immersive experience of comic and *manga* reading. This is due to the nature of the shape of the sound effects as most of the sound effects do not come in a consistent format, which makes it hard for the OCR to detect. Besides, the the presence of the text by itself usually is enough to convey the message to the reader so this aspect of text would not be included in the scope of the project. Additionally, the application will only detect the text present on the comic or *manga* panels instead of detecting the speech bubbles. This is also due to the fact the speech bubbles possess different formats of shape and sizes which also makes it hard for the OCR to detect.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 Introduction

As the comic industry expand over the years, so does the demand for more translated mangas. As an example, currently there are still many popular mangas that remain to be untranslated for western audiences, one of which is the very popular *GUYVER: BIO – BOOSTER ARMOR* which sparked a massive cult following in the 1990s, but the fans of the series were left disappointed as the western localizer for the series, VIZ, dropped the translation for the series after the original went on a hiatus (Burton,2022). There were also efforts made by fan translation groups but were also quickly dropped for unknown reasons. Not only that, there still remains countless of popular manga remain to be untranslated such as *Sket Dance*, *Billy Bat*, *Hajime No Ippo*, *Kingdom*, *KochiKame*, *HolyLand*, *Kinnikuman*, and many more (Fox, 2023). With this in mind, in the absence of official and fan translated efforts, if ordinary users without knowledge on foreign language want to read their favorite, but unlocalized manga, an automated and computerized solution is needed to enable such. Hence, the project will be started with the intent to also meet the reading community's demands for automatically translated manga.

A literature study is essential for acquiring a greater understanding of the topics relevant to designing an application for an automated computer vision-powered manga/comic translation that can meet the requirements of its users. A series of reviews on the components needed for the translation application and its components were undertaken for this purpose. In addition, an analysis and comparison of various software development methodologies was conducted, and Agile in the form of the Kanban approach will be employed for this project.

#### 2.2 Review of Other Manga Translation Applications

There exist some other automated manga translation applications which aim to provide automated comic and manga translations for users via web browsers in the form of browser extensions. The applications in consideration for the reviews are ISMANGA and Scan manga Translators as there are the few highly rated applications with a free trial model.

### 2.2.1 ISMANGA Manga Translator

ISMANGA is an application from SHUHO SATO which is made to primarily translate mangas. The application in its web extension form provides a non-intrusive and user-friendly interface that facilitates ease of use. The application currently can translate comics and *mangas* of Japanese, Chinese, and Korean origins into English, German, French, Malay, and many more. The application has two versions, namely “Free” and “Premium” and the the two versions differ in the speed of the translation, the accuracy and the quality of the translation, and the frequency of translations allowed per week.

While it provides readable quality translations, the use of the application is limited to items on internet browsers as it lacks the functionality to read files locally. While ISMANGA has an Android application available on Google Play, it possesses the same constraint as the web extension application as it cannot perform translation tasks on local files. The application also sometimes struggles with detecting large texts that are located outside of the boundaries of the speech bubbles which may shatter the immersion of the reading experience for the users.



Figure 2-1 Original manga page of sample 1



Figure 2-2 Original manga page of sample 2



Figure 2-3 ISMANGA translated page of sample 1



Figure 2-4 ISMANGA translated page of sample 2, ISMANGA struggles to detect the texts

### 2.2.2 Scan Translator

Scan translator is a *manga* translation application from an unnamed developer from the Chrome web store. The application comes with an easy-to-use user interface and can translate *manga* panels into more than 50 languages. The application also has a free and paid version in which the free version can only translate up to 20 panels every week. The application can detect words within and outside of the speech bubble which makes it a versatile tool for reading.

The downside to the application is that it is only available as a web browser extension, meaning it cannot be used to view locally stored manga panels. The application also has the tendency to overlay oversized white panels on the translated words that heavily obscures the panels which may diminish the reading experience for most readers. The use of Scan translator is also only limited to translating *mangas* as it does not read any languages other than Japanese. With this, the use of this application is also limited to some degree.

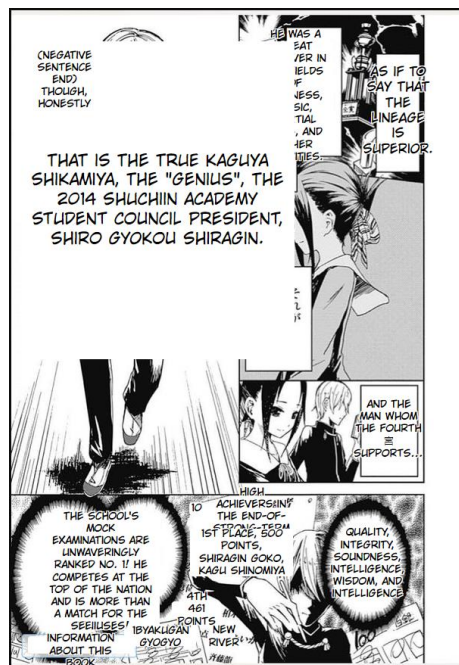


Figure 2-5 Scan translated page of sample 1



Figure 2-6 Scan translated page of sample 2

### 2.3 Pillow

Pillow, or PIL (Python Image Library) is an imaging library for Python that adds image processing capabilities to the Python interpreter. This library supports a wide range of file formats, possesses efficient internal representation, and has rather robust image processing features. The core image library was created to provide quick access to data contained in a few fundamental pixel formats. It should be a good starting point for a broad image processing program. Which in case, in order to establish a good starting point for image OCR, PIL is used to handle the opening and editing of photos in a variety of formats.

Here are the unique features that PIL offers:

- **Image Processing**
- **Image Segmentation**
- **Image Manipulation**

### 2.4 OpenCV

OpenCV is a computer vision library that is mostly used for image processing. It was initially a research project by Intel which was later made freely accessible under the Berkeley Software Distribution license, which is open source. OpenCV includes a number of tools for dealing with computer vision problems. It includes image processing operations at a low level as well as high-level algorithms for detecting objects, object matching, and tracking (Naveenkumar & Ayyasamy, 2016). OpenCV provides the following main image processing techniques:

#### i) **Image Filtering**

A method of changing or improving a picture. There are two kinds of image filtering. The first is linear image filtering, in which the value of an output pixel is a linear combination of the values of the pixels in its neighborhood. The second is non-linear image filtering, in which the output value is not a linear function of the input (Naveenkumar & Ayyasamy, 2016).



## ii) Image Transformation

Produces a new image from multiple sources that highlights specific traits or attributes of interest better than the original input images. The main image transformation methods it provides are:

- **Hough Transform** - Find lines in an image
- **Radon Transform** - Reconstruct images from fan-beam and parallel-beam projection data
- **Discrete Cosine Transform** - Compress image and videos
- **Discrete Fourier Transform** - Filter and analyse frequency
- **Wavelet Transform** - Perform discrete wavelet analysis, denoise, and fuse images

## iii) Feature Detection

A feature can be considered as an area of interest which is used as a starting point for running a computer vision algorithm. The technique of detecting such elements like lines, edges, or angles, is known as feature detection. It will be useful for making local decisions regarding the picture's local information contents such as detecting speech bubbles on a panel from a comic or *manga*.

## 2.5 Keras-OCR

Keras-OCR is a library built on top of TensorFlow and Keras that simplifies building and using OCR systems. Keras-OCR provides a user-friendly interface for building and utilizing OCR models. This makes it easier for developers to use OCR functionality without needing in-depth knowledge of the underlying deep learning techniques. Keras-OCR comes with pre-trained models that can be used for immediate text extraction from images. This eliminates the need for users to train their own models from scratch, saving time and resources. While pre-built models are available, Keras-OCR also allows for training custom models on specific datasets. This is beneficial for scenarios where the pre-trained models might not perform optimally due to variations in text fonts, styles, or languages. Keras-OCR is an open-source project, allowing anyone to contribute to its development and improvement. This fosters an active community that helps keep the library updated and address potential issues. Keras also provides similar

functionality where it can run feature detection on words. This allows Keras-OCR to provide prediction data on the location of words which may be much more accurate as compared to OpenCV.

## **2.6 Review on Machine Translation Services**

A proper translator is crucial to the project's success in terms of achieving the goal of providing quality translations to the users. There exist several machine translation services that can be selected and compared to determine the right translator. Two machine translation services such as DeepL and Google Translate will be considered and discussed in terms of the service's offering and performance.

### **2.6.1 DeepL**

DeepL is a machine translation service by DeepL SE that employs the use of artificial intelligence and deep learning techniques to produce highly accurate and contextual sensitive translations from one language to the other. DeepL uses neural networks, which is a form of AI model inspired by the human neural pathways with the intention to generate translations that are capable of mimicking human language patterns. DeepL uses complete parallel corpora, pairs of texts in multiple languages that are translations of each other to train their AI model. This data is then fed into the network so that relationships between words, phrases, and sentences can be found. DeepL's neural networks excel at understanding context, allowing it to capture nuances and nuances of language. This contextual awareness helps develop translations that go beyond simple word-by-word changes, resulting in consistent, natural-sounding translations.

### **2.6.2 Google Translate**

Google Translate is a widely used machine translation service by Google that employs the use of a combination of NMT (Neural Machine Translation), which is similar to DeepL, this method is also the successor to Google Translate's previous methods such as rule-based algorithms and SMT (Statistical Translation). This drastically improves the translator's understanding of the structure and context of sentences, enabling more contextually accurate translations. Google Translate's neural network is trained on multilingual text data. The

networks analyze the input, encode its meaning, and then produce the decoded content. Ongoing training and use of new data helps improve system performance over time.

### 2.6.3 Comparison of the Machine Translation Services

Despite both translation services using NMT to perform translation operations, each has its own advantages and drawbacks. A number of factors can be weighed:

Criteria	DeepL	Google Translate
Discontinuous form sequence accuracy	Good	Average
Continuous form sequence accuracy	Good	Good
Sentence Structure	Good	Bad

Table 2-1 The comparison between DeepL and Google Translate

As shown in the table above, DeepL has the advantage against Google Translate in terms of providing a more accurate form of discontinuous sentences and retaining the sentence structure of the translated text. According to a study conducted to test the translation capabilities between DeepL and Google Translate in Spanish, DeepL was able to detect the idiom in Spanish and was able to detect and translate the idiom and provide the appropriate equivalent in English while Google Translate was still able to detect and translate the idiom but often provided a crude translation of the idiom in English form, which cripples the sentence structure. Both translation services perform similarly in the continuous form of the sequence in terms of accuracy. (Hidalgo-Terner, 2021). In regards to translating from Japanese text to English text, DeepL produces the translated sentences with proper sentence structure and grammar and is also able to translate Japanese dialects to English whereas Google Translate sometimes omits crucial info and produces a much more crude translation which makes it harder for readers to understand. Overall, DeepL proves to be much more capable and will be chosen as the main translator in this project.

## **2.7 Review on OCR Tools**

The success of the project hinges on the quality of the application's ability to accurately recognize the characters on the images as the resulting texts will be used as the basis of how the rest of the output will result in. Therefore, choosing the best OCR solution among the abundance available is critical. Tesseract, Manga OCR, Google Document AI, and Amazon Textract are evaluated based on their distinctive features and efficiency. To achieve the best outcomes, the ultimate decision is to align the tool's capabilities with the project's needs.

### **2.7.1 Tesseract**

Tesseract is an OCR developed by Hewlett-Packard and was later extensively improved on by Google (Huda,2017). Tesseract is mostly used to convert printed or handwritten text into manipulatable digital text for computer analysis. Tesseract is also capable of multi-language support as it can detect text in multiple languages. Tesseract has also gained a reputation for having one of the best free OCRs with high accuracy as it employs advanced machine learning techniques which are always under continuous improvement to improve the accuracy. Tesseract is also listed as an open-source project, which means the source code is open to the public and this allows it to be actively maintained and improved by a community of enthusiasts. Tesseract also comes with a solid layout analysis function to detect the structure of the placements of text, lines, and text. Since Tesseract lacks advanced image preprocessing techniques it is best paired with a good image preprocessor to fully utilize Tesseract's capabilities.

### **2.7.2 Amazon Textract**

Amazon Textract is a Machine Learning service by Amazon that claims to be more than just an OCR software. Amazon Textract is used primarily to automatically extract text, handwriting, and data from documents from image format, document formats, and PDFs. Amazon also supports multiple languages which makes the use of this service versatile. Textract can be used to extract texts from tables while maintaining the table's structure with table recognition and also forms too with form recognition, this makes digitization of files easier with Textract. Textract also has the capability to recognize handwritten which expands the OCR capability of the service greatly. Textract also comes with layout analysis functionality to understand the layout and structure of the document. As Textract is part of Amazon Web Services, it benefits

from the added scalability and innate integration with other Amazon's Web Services cloud solutions to use the extracted text for other purposes.

### **2.7.3 Google Document AI**

Document AI is a Machine Learning service by Google with the capabilities of an OCR software. Document AI is largely used to automatically extract text, handwriting, and data from images, document formats, and PDFs. Document AI also supports several languages, making it more adaptable. Document AI can deconstruct documents into their component pieces, such as tables, headers, and footers. Document AI also comes with layout analysis functionality to understand the layout and structure of the document. As Document AI is a cloud-based service, it benefits from the added scalability to process large volumes of documents.

### **2.7.4 Manga OCR**

Manga OCR is an OCR developed by a developer of the username "kha-white" on GitHub which specializes in performing text extraction for Japanese text. This OCR is based on a custom end-to-end model built on the Vision Encoder Decoder transformer framework. In other words, this OCR is fine tuned with pre-trained model of various Japanese text from manga panels. Through this, Manga OCR is capable of delivering highly accurate Japanese text recognition through various scenarios which other OCR models may struggle against such as: Vertical and horizontal text in a single image, text with furigana writing, text overlaid on images, various font styles and even low-quality images (kha-white, 2023). This makes Manga OCR a really powerful tool as pre-processing of images may not be even needed for an OCR with this sort of performance.

### 2.7.4.1 Comparison of the OCR Tools

Mean error on:	Document AI	Textract	Tesseract
Coloured sources, 1932 words	1.3	2	2.6
Binary color sources, 1932 words	1.3	1.7	2.3
Sources with common image noises, 42504 words	8.2	13	23.3

Table 2-2 The comparison between Document AI, Textract and Google Tesseract

As shown in the table above, Document AI excels in every field which is followed by Textract and then Tesseract. For results that were retrieved from scanning over 1900 words, the world errors produced by the three OCR tools can be considered negligible. However, when it comes to images with added noise such as blur, weak ink, salt and pepper, watermark, scribbles, and ink stains, the error rate increased significantly and especially so for Tesseract. This may be a result of the fact that Document AI and Textract are server-based which means they have access to a more scalable infrastructure that allows both tools to acquire more resources for processing if required while Tesseract is limited to running on a local machine with limited resources for processing OCR tasks (Hegghammer, 2022). Despite the issues that arise from the introduction of image noises, this is generally not an issue as most digitally published works do not possess graphical noises. With these considerations in mind, Tesseract will be chosen as the main OCR component in the project as it is free to use and the sufficient performance it provides for simple OCR tasks such as scanning digital comics and *mangas*.

#### 2.7.4.2 Tesseract vs Manga OCR : Comparison on Japanese Text Accuracy

Number of errors on :	Tesseract	Manga OCR
Monthly Girls' Nozaki-kun “お返事”	50	0

Table 2-3 The comparison between Tesseract, and Manga OCR

A comparison was conducted to try out the performance between the two OCR tools which are Tesseract and Manga OCR. From a test which was conducted by using the two OCR tools to extract the text boxes of a *manga* from chapter “お返事” from Monthly Girls' Nozaki-kun which can be found in the appendix, according to the data shown from the table, Manga OCR provided good results as there are no errors on the extracted text results whereas Tesseract provided subpar results with a staggering 50 word error. This is primary due to the reason that Tesseract isn't well tuned to detect Japanese texts with varying font styles, it is also limited to being able to use a single mode of scanning, namely “Japanese vertical (jpn\_vert)” and “Japanese(jpn)” which limits it to either just scan for vertical Japanese text or horizontal text at a time. Manga OCR is able to detect the fonts even with blurry graphics and is able to accurately extract text with various font styles. The only downside with the latter OCR tool being that it may be slow at times since it requires higher processing power to perform extraction operations.

## 2.8 Summary of Literature Findings

From analyzing the features of the existing comic/*manga* translation tools, both applications has the capability to detect texts on images although to a varying degree such that the Scan translator has a better feature detection algorithm which makes it possible to detect both text inside and outside of the text bubbles whereas ISMANGA failed to do so in detecting text outside of text bubbles. Scan translation excelled at text detection, but the overlaid translated text is way too big for reading comfort. The inability to download the translated comic also comes off as a detriment to the two applications as some users may prefer to re-read the comic at a later time. In this case, the proposed system will consider the following components:

1. **Pillow** – To import images into the application, overlay translated text onto the images.
2. **Keras OCR** – To detect words on the image.
3. **OpenCV** – To preprocess the images for OCR, mask over speech bubbles, perform save operations on translated images.
4. **Tesseract** – To perform OCR task to extract texts from the image.
5. **Manga OCR** – To perform OCR task to extract texts from the image.
6. **DeepL** – To perform translation operations on the extracted texts.

The system aims to improve on the functionality demonstrated in the review made on the existing systems by offering a solution to edit out the old texts on the translated image using OpenCV masking techniques to avoid obscuring the graphics such was the issue with Scan translator. The system should also possess two types of feature detection for pure text detection and speech bubble detection to rectify the issues demonstrated with ISMANGA translator. DeepL will be used in hopes of providing a much more natural sounding translation for the users. Tesseract will be used only for OCR tasks and Pillow is used to provide a more localized means of importing and saving images.



## CHAPTER 3

### METHODOLOGY AND WORKPLAN

#### 3.1 Introduction

The selection of a development methodology is critical to the success of a software project. The following methodology includes the Waterfall model, Scrum in Agile, and Kanban In Agile. The selected methodologies will be assessed based on their characteristics, advantages, and disadvantages.

#### 3.2 Waterfall Model

First Introduced in 1970 by Dr. Winston W. Royce, the Waterfall model is a software development methodology that separates the development tasks into different management phases, in which each phase requires a deliverable from the former phase to start operations (Hoory, 2023).

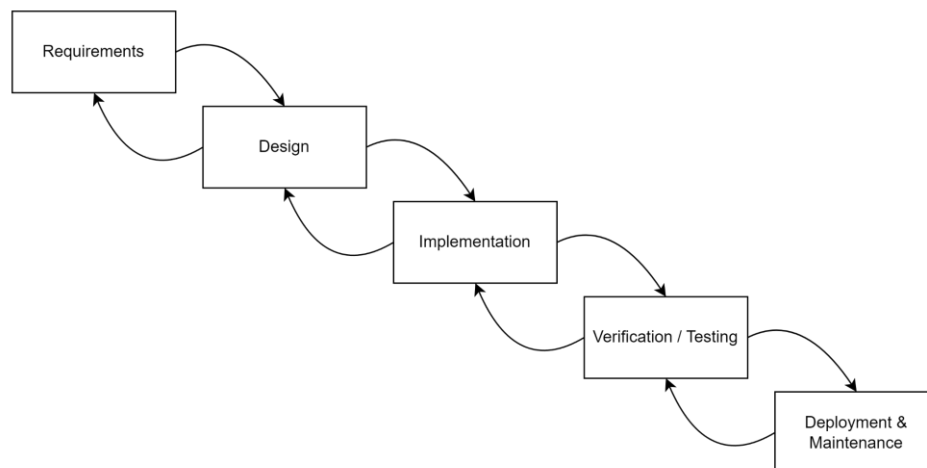


Figure 3-1 The waterfall model

According to the illustrated form of the Waterfall Model from Sommerville's (2010) description, the phases are commonly divided into 5 phases. :

1. **Requirements Phase** : Discussion with system users determines the system's services, constraints, and goals. They are then thoroughly specified and constitute a system specification.

2. **Design Phase :** By developing an overarching system architecture, the systems design process assigns requirements to either hardware or software systems. Identifying and determining the underlying software system concepts and their interactions is part of software design.
3. **Implementation Phase :** The design of the software is implemented as a collection of units.
4. **Verification/Testing Phase :** To ensure that the software requirements are satisfied, the different program modules are integrated and tested as a whole system.
5. **Deployment & Maintenance Phase :** The system has been installed and tested. Maintenance includes rectifying mistakes that were not identified earlier in the life cycle, improving system unit implementation, and increasing system services when new requirements are recognized.

<b>Advantages</b>	<b>Disadvantages</b>
Clients would not delay production by demanding more requirements	Lack of client involvement
Total cost can be estimated accurately	Does not cater for sudden changes in requirements
Design errors are captured early before implementation	Difficult to move back to the former phase

Table 3-1 The advantages and disadvantages of waterfall model

The Waterfall methodology's efficiency is determined by the volume and quality of work done on the front end, which includes meticulously documenting everything, including the user interface, storyboard, and all iterations and outcomes of the features. Since the majority of the research is completed ahead of time, predictions of the time necessary for each need are more precise, resulting in a more predictable delivery date. It is more difficult to change the direction of the moving phase on a Waterfall project if specifications change along the way during the duration of the project. Due to the nature of the project which involves constant communication and feedback between the project supervisor and the student, this development methodology might not be a good fit for this project.

### 3.3 Agile Methodology

The Agile methodology is a project management technique that emphasizes constant collaboration and improvement by splitting the project into stages. Teams go through a cycle of planning, executing, and evaluating. The agile methodology has a heavy focus on incremental and iterative development, Which allows members of the project to deliver results in smaller increments during any part of the project instead of providing a deliverable at the very end of a project, such as the Waterfall model methodology.

#### 3.3.1 Kanban In Agile

Initially conceived in the 1950s as a part of the JIT (Just-In-Time) production system for Toyota. The concept of Kanban revolves around the concept of making “only what is needed, when it is needed, and in the amount needed” (Toyota, 2023). The execution of the Kanban agile framework requires the use of a Kanban board to visualize the workloads in 3 different categories which is explained earlier in section **1.6 Project Approach**.

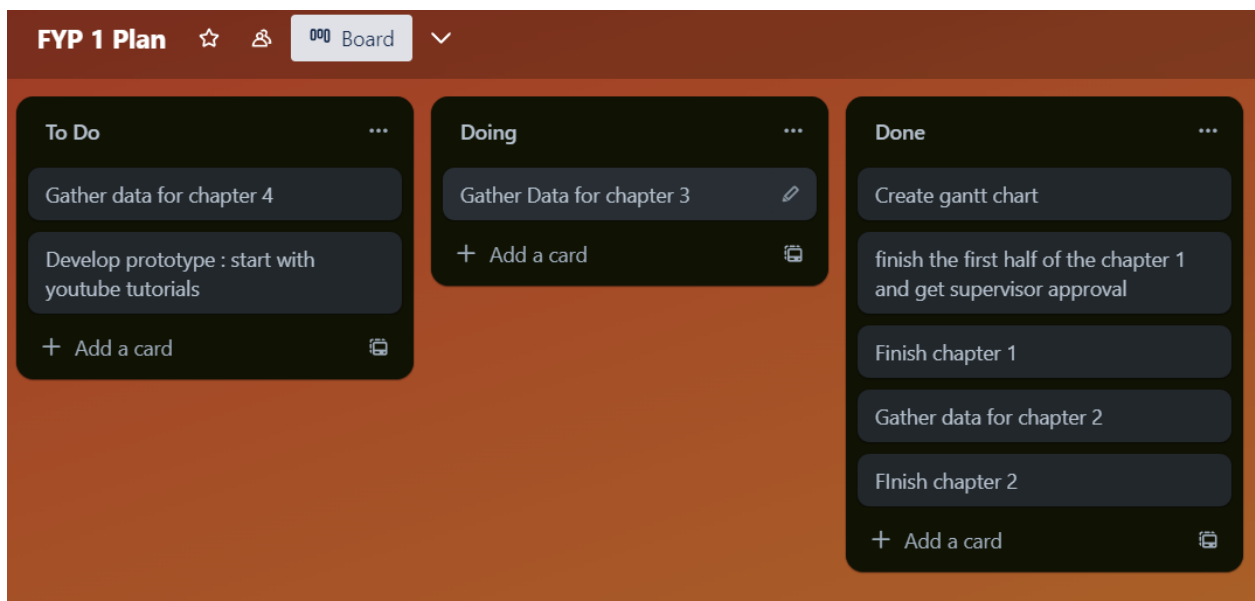


Figure 3-2 The kanban board

By separating and visualizing the tasks, the developers are able to standardize their workflow by simply referring to the Kanban board and it serves as the singular source of indicator for the developers’ workflow (Radigan, 2022).

Advantages	Disadvantages
Clarity on tasks at hand	Limited complexity
Burnout avoidance	Hard to predict project schedule
Improved efficiency	Incompatibility in a dynamic environment

Table 3-2 The advantages and disadvantages of kanban in agile

The Kanban framework takes a more flexible approach since it does not require specified roles or timed iterations. This makes Kanban ideal for projects which need processes that are tailor-made to fit specific contexts and requirements, which are usually smaller in scale. This is also one of Kanban's major flaws, since it is limited in large-scale projects when the timeline and roles are essential.

### 3.3.2 Scrum In Agile

First implemented in 1993 by Jeff Sutherland, John Scumniotales, and Jeff McKenna, this variation of agile framework outlines a set of values, principles, and practices that the scrum team must adhere to in order to provide a product or service. It describes the scrum team members and their responsibilities which defines the work required to create the deliverable product, and the scrum ceremonies that guide the scrum team through work. According to Drumond (2018), there are a total of 5 scrum ceremonies:

1. **Sprint Planning:** Done at the start of the sprint. Commits to tasks to be completed in the next sprint
2. **Daily Scrum:** Done every day during the sprint. Provide progress updates and synchronize on workflow.
3. **Sprint Review:** Done at the end of the sprint. Retrieve feedback from stakeholders.
4. **Sprint Retrospective:** Done at the end of the sprint. Inspect performance from the former sprint.
5. **Backlog Refinement:** Takes place continuously. Ensure the product backlog is well-defined to be included in the sprint.

<b>Advantages</b>	<b>Disadvantages</b>
Clear definition of each member's responsibilities	Mostly only successful with experienced members
Better task visibility	Subject to scope creep
Suitable for work in dynamic environment	Quality control can be hard to implement

Table 3-3 The advantages and disadvantages of scrum in agile

The Scrum framework is ideal for project teams desiring for a more defined framework for project management. This method allows teams to collect feedback early and frequently, which may lead to improved outcomes and increased customer satisfaction from the improved quality of the software from the adoption of the scrum ceremonies.

### 3.4 Conclusion on Chosen Methodology

Since the waterfall method has proven to not be effective in this type of project, only the Kanban and Scrum agile frameworks will be considered in this comparison.

<b>Kanban</b>	<b>Scrum</b>
Does not have predefined roles and responsibilities	Has predefined roles and responsibilities
High focus on cutting lead time	Lead time cutting is not emphasized
Emphasis on flexibility	Emphasis on continuous devotion to projects

Table 3-4 The comparison between kanban and scrum

Since the project will only be accomplished by a single person, the added flexibility helps in refining the workflow as patterns for solving the problems are discovered achieve the project objectives from section **1.6** in **Chapter 1**. There is minimal overhead with the Kanban approach as there is no requirement for specific developer roles which makes project management easier. There are also no timeboxing issues with Kanban since tasks can be changed without the need to synchronize with other team members. With these considerations in mind. The Kanban in agile methodology will be chosen as the mode of operation in the development of the project.

### **3.5 Proposed Workplan**

#### **3.5.1 Work Breakdown Structure**

##### **1. Project Preliminary Phase**

- 1.1 Conduct background study
- 1.2 Determine Problem Statement
- 1.3 Establish project objectives
- 1.4 Propose project solution
- 1.5 Determine project approach
- 1.6 Determine project scope

##### **2. Requirements Planning**

###### **2.1 Literature review**

- 2.1.1 Analyze existing comic translation applications
- 2.1.2 Analyze image library components
- 2.1.3 Analyze computer library components
- 2.1.4 Analyze online translation services
- 2.1.5 Analyze Optical Character Recognition components

###### **2.2 Determine system development approach**

- 2.2.1 Determine methodology for project development
- 2.2.2 Determine development tools and technologies

###### **2.3 Define system requirements**

###### **2.4 Create UML diagrams**

###### **2.5 Design Prototype**

##### **3. System Development Phase**

- 3.1 System analysis
- 3.2 Develop prototype
- 3.3 Prototype demonstration
- 3.4 System development
- 3.5 System integration

##### **4. Testing Phase**

- 4.1 Preparation for testing
- 4.2 Unit testing
- 4.3 Feature testing
- 4.4 User acceptance testing

##### **4. Deployment Phase**

- 4.1 Deploy PC version

### 3.5.2 Gantt Chart

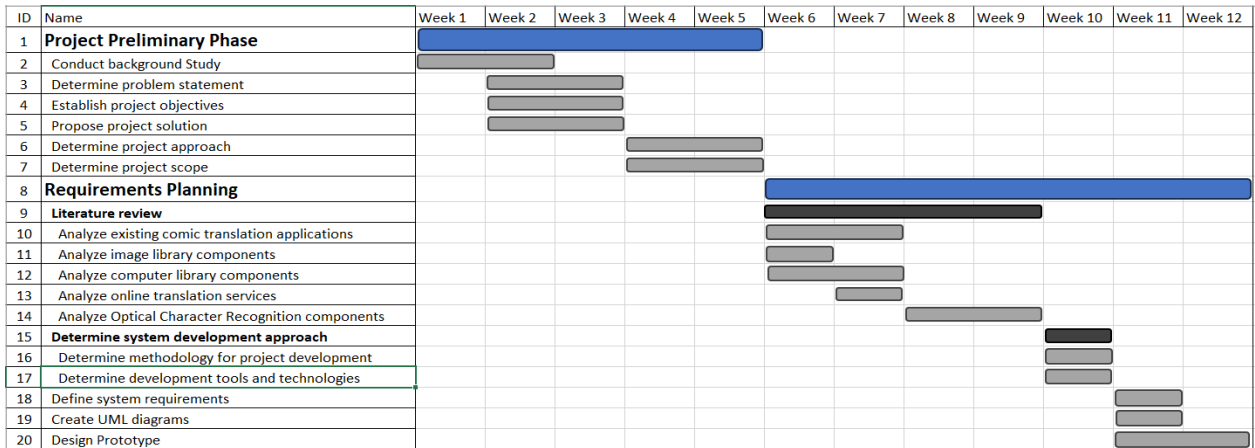


Figure 3-3 Gantt Chart for Project I

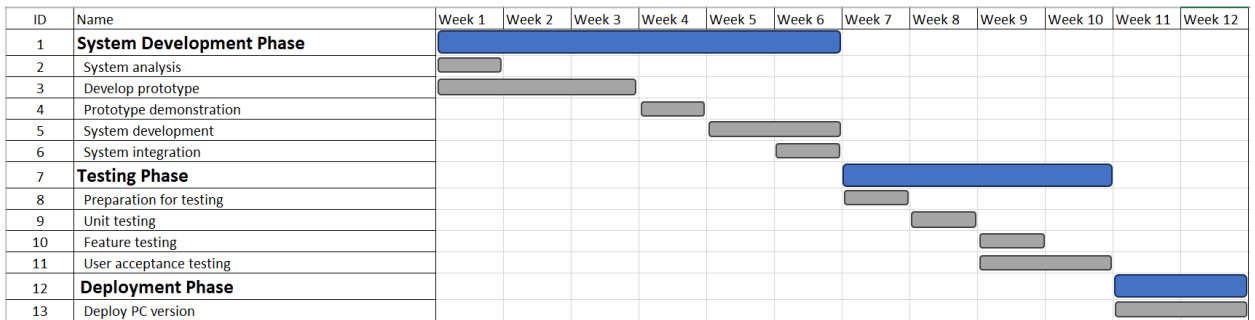


Figure 3-4 Gantt Chart for Project II

### 3.6 Development tools and technologies

The development tools and technologies listed below were employed during the proposed system's development phases.

#### 3.6.1 Pillow

Pillow is a Python imaging library that adds imaging processing capabilities to the Python interpreter. It offers a variety of features for opening, modifying, and saving pictures in different picture formats. Pillow fulfils the project's scope in **“Text Overlay”**. The inclusion of Pillow will be a key part of the project as an imaging processing component.

#### 3.6.2 OpenCV

OpenCV is a computer vision library that will be used in the project. This is because it can be used to open images and perform image modification, this will fulfill the project's scope in **“Image Import”**. It also possesses image filtering and transformation techniques which will be critical to the accuracy of the OCR component used in the project. OpenCV is also capable of overlaying various images on the detected areas on an image. OpenCv's role in feature detection tasks will play a major role too in the project as it is needed to detect the presence of text bubbles on the comic/manga panels to fulfill the project's scope in **“Text Detection and Conversion”** and **“Mask Drawing”**.

#### 3.6.3 Keras-OCR

Keras OCR is a Keras and Tensorflow derivative library for OCR systems and will be implemented in the project. Keras OCR sports a neural engine and an extensively trained model which will help it in providing accurate and satisfactory results in regards to detecting words to be processed for text extraction. This component will help in fulfilling the project's scope in **“Text Detection and Conversion”**.

#### 3.6.4 DeepL

DeepL is a neural network-powered AI machine translation service that will be used to provide the application the capability to translate the extracted text. DeepL capability to provide accurately translated sentences with proper sentence structure and grammar makes it a good fit for the project as it fulfills the project's scope in **“Translation”** and the project's objective in **“To translate foreign languages into the user's preferred language”**.



### 3.6.5 Tesseract

Tesseract is an OCR software that will be used to extract the texts from the comic/manga panels. Tesseract's ability to provide decently accurate OCR results while still being a native component is the reason it is chosen to be a key component in the project. Tesseract fulfills the project's objectives in **“To reduce the time required to translate comics and manga”** and **“To aid non-skilled users in performing translation operations”** and the project's scope in **“Text Detection and Conversion”**. Tesseract is also free to use which makes it the more economical solution for this project.

### 3.6.6 Manga OCR

Manga OCR is an OCR software specialized in extracting Japanese texts from manga panels. Manga OCR excels in providing exceptional OCR results is the reason its chosen to be the main Japanese text extractor. Manga OCR also fulfills the project's objectives in **“To reduce the time required to translate comics and manga”** and **“To aid non-skilled users in performing translation operations”** and the project's scope in **“Text Detection and Conversion”**. Manga OCR is also free to use which makes it the more economical solution for this project.

## 3.7 Summary of Project Methodology

Overall, this project will employ the use of Kanban In Agile methodology as the mode of development, this is due to the fact that the project touches on a field of software which is not widely popular and does not have surplus documentation to help determine on a concrete method to capture and OCR text. With this consideration in mind, the project development will be done by a single member and will likely involve adding multiple methodologies in the project backlog to be tried later upon experimentation on the different OCR methods.

As for the execution of the project application's intended purpose, the application will utilize Pillow solely to perform text overlay OpenCV will be used to import the image and perform image preprocessing, extraction of the text area, and image masking on the image. Keras will be used to perform text detection and provide the prediction data for OpenCV to extract the detected areas, Tesseract and Manga OCR will be used to convert the extracted image to a series of strings, DeepL will be used solely to translate the extracted text from

Tesseract, and lastly, the results will be overlaid on the previously masked areas on the image and shown as the final result to the user.

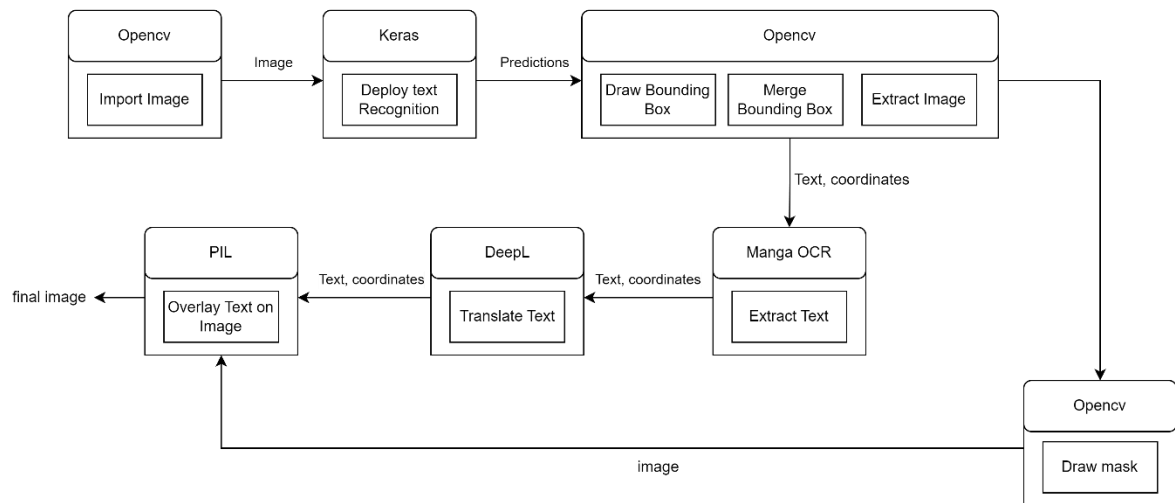


Figure 3-5 System architecture for the proposed project

## CHAPTER 4

### PROJECT SPECIFICATIONS

#### 4.1 Introduction

This chapter will discuss the requirements specification, use cases, and the layout of the User Interface for the application.

#### 4.2 Requirements Specification

##### 4.2.1 Functional Requirements

1. The system should allow the user to select the picture in the file manager for translation.
2. The system should allow the user to select the input language of the translation.
3. The system should allow the user to select the desired output language for the translation.
4. The system should be able to detect the texts present in the image.
5. The system should be able to output the translated image panels to the user.

## 4.2.2 Non-Functional Requirements

### 1. Usability

- a. The system shall be easy to use for users of all ages without training.
- b. The system UI shall be adaptable to laptop and desktop screen sizes.

### 2. Performance

- a. The system shall be able to finish translating an image in less than a minute on CPU-only processing.

### 3. Availability

- a. Since the core functionality of the system rely on external online API services such as the translation API, there is no concrete method to ensure the availability of the application.

### 4. Correctness

- a. The system shall be able to return the translated image in the right language specified by the user.

### 5. Security

- a. Due to the nature of the project which does not require user login, the use of security measures is redundant in this project.

### 4.3 Use Case

The interaction between users and the comic/*manga* translation system is depicted in the use case diagram below. Each use case is accompanied by a use case description, which explains how the use case can be triggered by the primary actor and the major sequence of events involved.

#### 4.3.1 Use Case Diagram

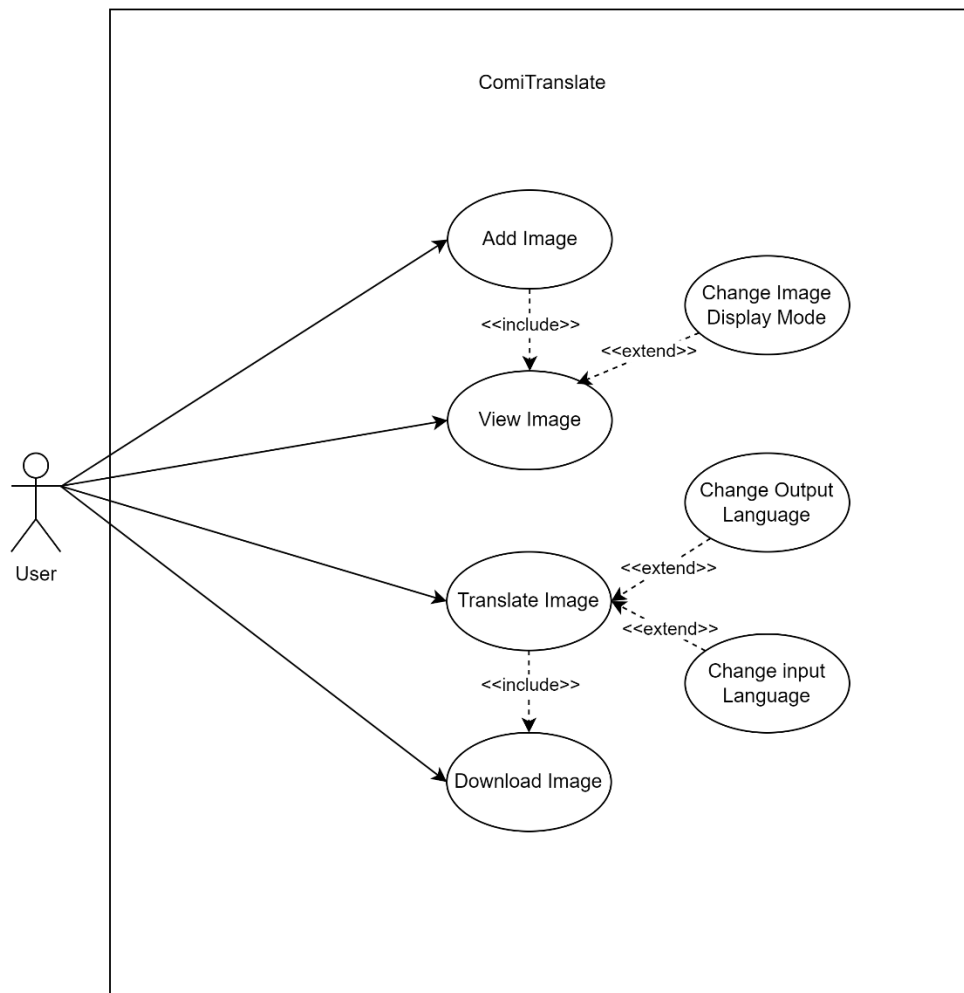


Figure 4-1 Use Case Diagram

### 4.3.2 Use Case Description

Use Case Name: Add Image	ID: UC01	Importance Level: High
Primary Actor: User	Use Case Type: Detail, Essential	
Brief Description: This use case describes how a picture is added into the system		
Trigger: User wants to add a picture into the system		
<b>Relationships:</b> Association : User Include : View Image Extend : N/A Generalization : N/A		
<b>Normal Flow of Events:</b> <ol style="list-style-type: none"> <li>1. User presses the “Browse” button in the top middle section of the Reading Page</li> <li>2. The system opens the file explorer for the user.</li> <li>3. User selects the desired picture</li> <li>4. The system imports the image.</li> <li>5. Perform UC02</li> </ol>		

<b>Use Case Name: View Image</b>	<b>ID: UC02</b>	<b>Importance Level: High</b>
<b>Primary Actor: User</b>	<b>Use Case Type: Detail, Essential</b>	
<b>Brief Description: This use case describes how the system shows the image to the user</b>		
<b>Trigger: User wants to view the image through the system</b>		
<b>Relationships:</b> <b>Association : User</b> <b>Include : N/A</b> <b>Extend : N/A</b> <b>Generalization : N/A</b>		
<b>Normal Flow of Events:</b>  <ol style="list-style-type: none"> <li>1. The system shows the user the picture in the Reading Page after UC01 is performed.</li> <li>2. If the user desires to view between the original image and the translated image, perform UC07.</li> </ol>		

Use Case Name: Translate Image	ID: UC03	Importance Level: High
Primary Actor: User	Use Case Type: Detail, Essential	
Brief Description: This use case describes how the user translates the pictures in the system		
Trigger: User wants to translate the images when reading		
<b>Relationships:</b> Association : User Include : N/A Extend : Change Input Language, Change Output Language Generalization : N/A		
<b>Normal Flow of Events:</b> <ol style="list-style-type: none"> <li>1. User wants to translate a comic/manga on the Reading page.</li> <li>2. User presses the “Proceed and Save button”</li> <li>3. The system translates the image.</li> <li>4. Perform UC04</li> <li>5. The system shows the processed image to the user.</li> <li>6. If user wants to make changes to the translation languages, continue to sub-flows 5.1 and 5.2.</li> </ol>		
<b>Sub-flows: {optional section}</b> <ol style="list-style-type: none"> <li>5.1 If user desires to make changes to the input language. <ol style="list-style-type: none"> <li>5.1.1 Perform UC05.</li> </ol> </li> <li>5.2 If user desires to make changes to the output language. <ol style="list-style-type: none"> <li>5.2.1 Perform UC06.</li> </ol> </li> </ol>		



<b>Use Case Name: Download Image</b>	<b>ID: UC04</b>	<b>Importance Level: High</b>
<b>Primary Actor: User</b>	<b>Use Case Type: Detail, Essential</b>	
<b>Brief Description: This use case describes how the user downloads pictures in the system</b>		
<b>Trigger: User wants to download the image after the system translates the picture.</b>		
<b>Relationships:</b> <b>Association : User</b> <b>Include : N/A</b> <b>Extend : N/A</b> <b>Generalization : N/A</b>		
<b>Normal Flow of Events:</b>  <ol style="list-style-type: none"> <li>1. User wants to download a translated image on the Reading page.</li> <li>2. User presses the “Proceed and Save” button on the Reading Page.</li> <li>3. The system downloads the image in the file explorer.</li> </ol>		

Use Case Name: <b>Change Input Language</b>	ID: <b>UC05</b>	Importance Level: <b>High</b>
Primary Actor: <b>User</b>	Use Case Type: <b>Detail, Essential</b>	
<b>Brief Description: This use case describes how the use changes the input language for the translation in the system</b>		
<b>Trigger: User wants to change the input language of the translation result in the Reading page</b>		
<b>Relationships:</b> <b>Association : User</b> <b>Include : N/A</b> <b>Extend : N/A</b> <b>Generalization : N/A</b>		
<b>Normal Flow of Events:</b>  <ol style="list-style-type: none"> <li>1. User wants to make changes to the input language on the Reading page.</li> <li>2. Continue to Sub-flows 2.1.</li> <li>3. The system makes the new changes done by the user and provides a new translation result based on the user's choice when user performs UC03.</li> </ol>		
<b>Sub-flows: {optional section}</b>  <b>2.1 If user desires to change the type of input language.</b> <ol style="list-style-type: none"> <li>2.1.1 User changes the language on the "From :" section to the desired language.</li> <li>2.1.2 return to normal flow 3</li> </ol>		

<b>Use Case Name: Change Output Language</b>	<b>ID: UC06</b>	<b>Importance Level: High</b>
<b>Primary Actor: User</b>	<b>Use Case Type: Detail, Essential</b>	
<b>Brief Description: This use case describes how the use changes the output language for the translation in the system</b>		
<b>Trigger: User wants to change the output language of the translation result in the Reading page</b>		
<b>Relationships:</b> <b>Association : User</b> <b>Include : N/A</b> <b>Extend : N/A</b> <b>Generalization : N/A</b>		
<b>Normal Flow of Events:</b>  <ol style="list-style-type: none"> <li>1. User wants to make changes to the output language on the Reading page.</li> <li>2. Continue to Sub-flows 2.1.</li> <li>3. The system makes the new changes done by the user and provides a new translation result based on the user's choice when user performs UC03.</li> </ol>		
<b>Sub-flows: {optional section}</b>  <b>2.1 If user desires to change the type of output language.</b> <ol style="list-style-type: none"> <li>2.1.1 User changes the language on the "To :" section to the desired language.</li> <li>2.1.2 return to normal flow 3</li> </ol>		

Use Case Name: <b>Change Image display mode</b>	ID: <b>UC07</b>	Importance Level: <b>High</b>
Primary Actor: <b>User</b>	Use Case Type: <b>Detail, Essential</b>	
<b>Brief Description:</b> This use case describes how the user changes between the two display modes on the Reading page, namely “Original” and “Translation”		
<b>Trigger:</b> User wants to navigate through the two display modes on the Reading page.		
<b>Relationships:</b> <b>Association :</b> User <b>Include :</b> N/A <b>Extend :</b> N/A <b>Generalization :</b> N/A		
<b>Normal Flow of Events:</b>  <ol style="list-style-type: none"> <li>1. User wants to switch between the two display modes on the Reading page. <ol style="list-style-type: none"> <li>1.1 Continue to Sub-flows 1.1 and 1.2.</li> </ol> </li> </ol>		
<b>Sub-flows: {optional section}</b>  <ol style="list-style-type: none"> <li>1.1 If user desires to view the original version of the image. <ol style="list-style-type: none"> <li>1.1.1 User presses on the “Original” button on the upper area of the translation tab.</li> <li>1.1.2 The system displays the unedited version of the picture to the user.</li> </ol> </li> <li>1.2 If user desires to view the translated version of the image. <ol style="list-style-type: none"> <li>1.2.1 User presses on the “Translated” button on the upper area of the translation tab.</li> <li>1.2.2 The system displays the translated version of the picture to the user.</li> </ol> </li> </ol>		

## 4.4 User Interface



Figure 4-2 Reading Page

The Reading Page is where the user reads the picture chosen by the user. The panel encased in **orange** lets the user select the picture to be translated and it triggers UC01. The panel encased in **purple** displays the images to the user as demonstrated in UC02. If the user desires to perform UC03 and UC04, they can click on the button encased in **yellow**. The elements encased in **red** triggers UC07 should the user desire so. The element encased in **blue** triggers UC05 and the **green** encased element below triggers UC06.

## CHAPTER 5

### SYSTEM IMPLEMENTATION

#### 5.1 Overview

The application was developed in stages of various time frames given the nature of Kanban technique where modules are only made when needed. Various prototypes of several modules were made to determine the best performing ones. Testing modules were added in the first iteration of the modules and then better versions of the modules were later made to replace the predecessor modules as needed. The final deliverable from the kanban board in this project was the creation of the UI for the whole backend code written in python via QT python.

#### 5.2 Image Translation Process

This section will provide an in-depth process on how the image translation is done on the system.

##### 5.2.1 Image Import

The final version of this module imports the image into the system via the use of OpenCV.



Figure 5-1 Imported image

### 5.2.2 Text Detection

The application utilizes Keras-OCR's pipeline to detect the text that are present on the picture. Areas which have detected to have a high probability of text will be taken and in the form of coordinates in four points, forming a rectangle around each detected text. This process usually leaves the text segmented in boxes.



Figure 5-2 Imported image with bounding box drawn to visualize the areas detected by Keras-OCR

### 5.2.3 Textbox Merging

In order to ensure that the OCR tool used by the application generates a full sentence from a single picture to avoid potential complications, bounding box merging techniques are employed to merge all nearby or overlapping boxes. The system achieves this effect by filtering out textboxes that may be too big, then boxes are expanded by the margin to increase the likelihood of detecting overlaps. The overlapping boxes will have its smallest rectangle enclosing the combining contour coordinates of the boxes drawn to form the merged version of the boxes via OpenCV. The merging of the boxes stops when there are no overlapping boxes within proximity.



Figure 5-3 Imported image with bounding box drawn to visualize the resultant merged boxes



### 5.2.4 Mask Drawing

Using OpenCV, masks are drawn in the form of white rectangles onto the image using the coordinates of the merged boxes to hide the original text and also to prepare for the overlay of the translated text back onto the image.



Figure 5-4 Imported image with white mask drawn

### 5.2.5 Textbox Extraction

With the coordinates gathered from the merged boxes, the text areas are then extracted using the coordinates of the merged boxes and will be passed onto the OCR to have the text extracted. The extracted boxes are also grayscaled to ensure higher text extraction accuracy when passed through the OCRs.

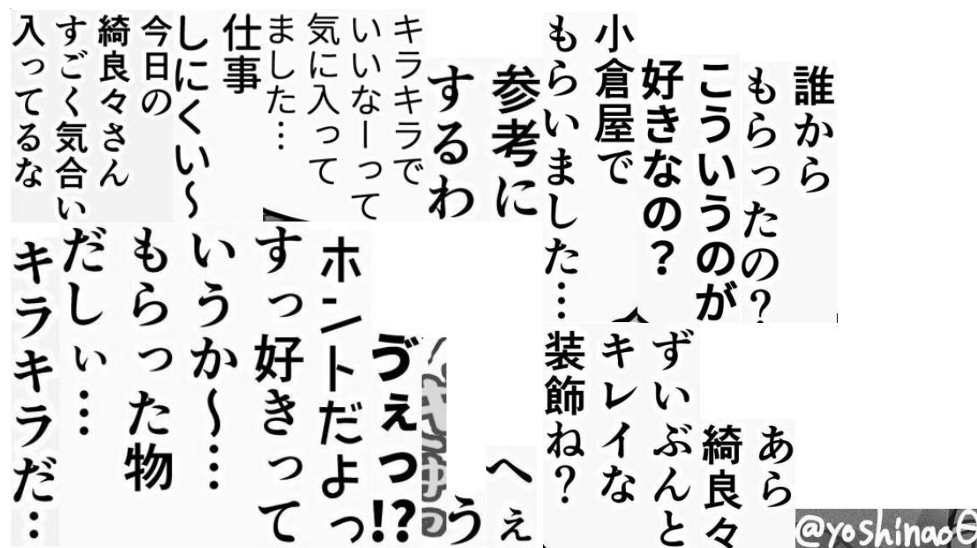


Figure 5-5 Extracted text regions from the imported image

### 5.2.6 Text Extraction

Using the one of the OCRs depending on the input languages chosen, Manga OCR in this case for Japanese, the system extracts the text from the extracted text regions.

```
Extracted text : ホントだよっ  
Extracted text : うえっ! ?  
Extracted text : そういえば、  
Extracted text : @yoskinash  
Extracted text : へえ  
Extracted text : キラキラだ:  
Extracted text : う  
Extracted text : 参考にするわ  
Extracted text : 仕事しにくい〜  
Extracted text : キラキラでいいなーって気に入ってました. . .  
Extracted text : 今日の綺麗々さんすごく気合い入ってるな  
Extracted text : 一応つけておこうかなって. . .  
Extracted text : すっ好きっていうか〜. . . もらった物だし. . .  
Extracted text : 誰からもらったの?  
Extracted text : 小倉屋でもらいました. . .  
Extracted text : こういうのが好きなの?  
Extracted text : これは〜. . . 別の所の荷物だよ. . .  
Extracted text : あら綺麗々  
Extracted text : ずいぶんとキレイな装飾ね?
```

Figure 5-6 Extracted text from the text regions

### 5.2.7 Text Translation

To translate the extracted text to the desired language, English in this case. The system will use DeepL to perform the task of translating the extracted text to the target language.

```
Translated Text: It's true.
Translated Text: Ew!
Translated Text: Speaking of which,
Translated Text: @y o s k i n a s h
Translated Text: eh?
Translated Text: Glitter:
Translated Text: う
Translated Text: Reference.
Translated Text: Hard to work with
Translated Text: I liked it because it sparkled ... .
Translated Text: Kira-san looks very enthusiastic today.
Translated Text: I thought I'd put it on just in case...
Translated Text: I like it so much..... It was a gift...
Translated Text: Who gave it to you?
Translated Text: I got it at Oguraya ... .
Translated Text: Do you like this kind of thing?
Translated Text: This is .... It's a package from another place ...
Translated Text: fine clothes
Translated Text: Pretty neat decor, isn't it?
```

Figure 5-7 Translated text

### 5.2.8 Text Overlay

To produce the final result image, PIL is used to perform text overlay, textboxes are drawn using the coordinates of the merged boxes and the text which might exceed the width of the textbox will be split and placed in a line below. The text are also scaled to 2% of the image to be overlaid on, this helps in making the fonts appear much more consistent and more easily readable for users.



Figure 5-8 Final product of the Comitranslate application

### 5.3 Reading Page Design

The figure below shows the Reading Page of Comitranslate, the page consists of a browse button to browse for images, original image and translated image buttons to switch between viewing the original image and the translated version of the image, two combo boxes to choose the input and output languages from English to Japanese, and finally the proceed and save button to start the image translation process and saving the translated image.

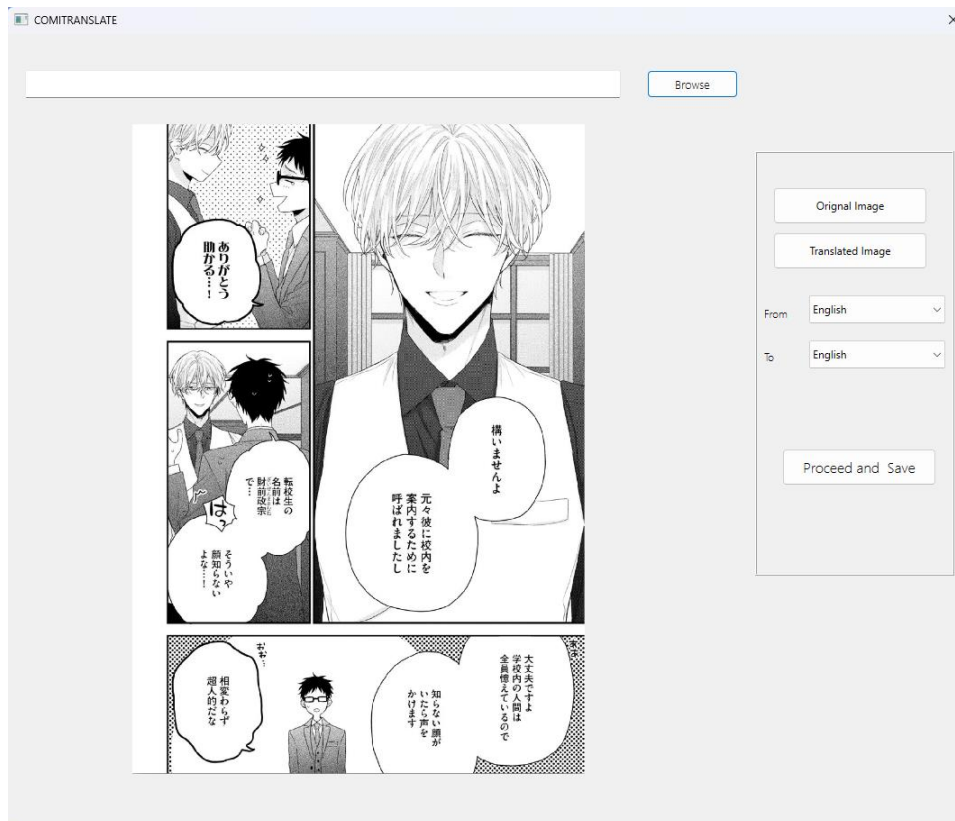


Figure 5-9 Comitranslate Reading Page

To safeguard the program against the possibility of the user loading in a non-image file, the application will return a warning in the center of the Reading Page

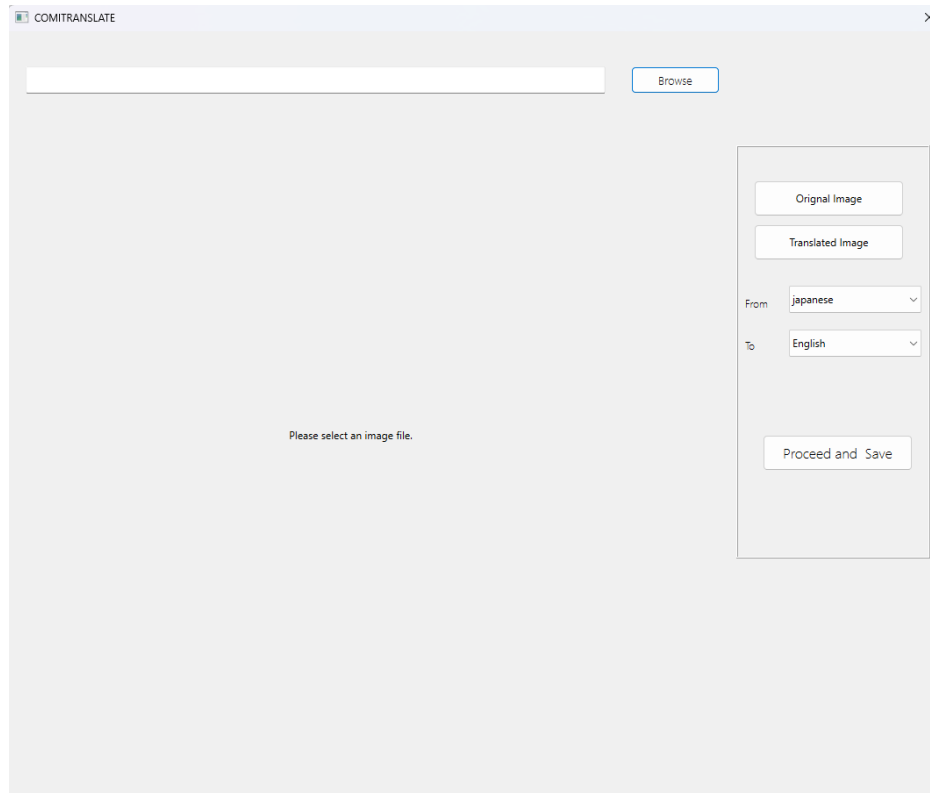


Figure 5-10 Comitranslate Reading Page displaying warning

## CHAPTER 6

### SYSTEM TESTING

#### 6.1 Introduction

This chapter will discuss the various types of testing involved in the project, emphasizing unit testing and integration testing. Automated testing plays a crucial role in ensuring that the image processing functions meet the project's requirements.

#### 6.2 Unit Testing

Python's "unittest" framework is used to write unit tests that validate the behavior of individual functions like "tup", "overlap", "grayscale", "getTextTesseract", "getTextMangaOCR", and "cleanImage". Tup, and overlap tests are needed to make sure that parts of the box merging function will work as intended. The testing of "grayscale" ensures that the output picture only has a 2D array which means it does not possess a colour channel. The testing of "cleanImage" will test if the output image has the same pixel resolution as the original to ensure preservation of image quality. The testing of "getTextTesseract", "getTextMangaOCR" will monitor if the output from the functions are in string.

Test case	Expected result	Status
Check Overlapping Boxes	The system should return true if boxes overlap and false if there is no overlap in boxes and bounds	Pass
Grayscale an image	The resultant image should possess a 2D array	Pass
Draw mask on image	The resultant image should retain the same pixel resolution as the original after masking	Pass
Extract English text	The resultant should be in a string	Pass
Extract Japanese text	The result should be in a string	Pass

Table 6-1 Unit Test Case



### 6.3 Integration Testing

Integration testing aims to determine if the collaborated functions works as intended, for this test, “getAllOverlaps” will be picked to determine if the combined function would return the right overlapped boxes to be merged.

Test case	Expected result	Status
Check Overlapping Boxes	The system should return the correct overlapped box coordinate.	Pass

Table 6-2 Integration Test Case

### 6.4 Visual Testing

Due to the nature of the project which produces images rather than human readable data, visual testing is conducted to check if the picture is processed such as the function “noise\_removal” which cannot be tested by automated means as it requires visual confirmation to check if the noise on the image is removed.

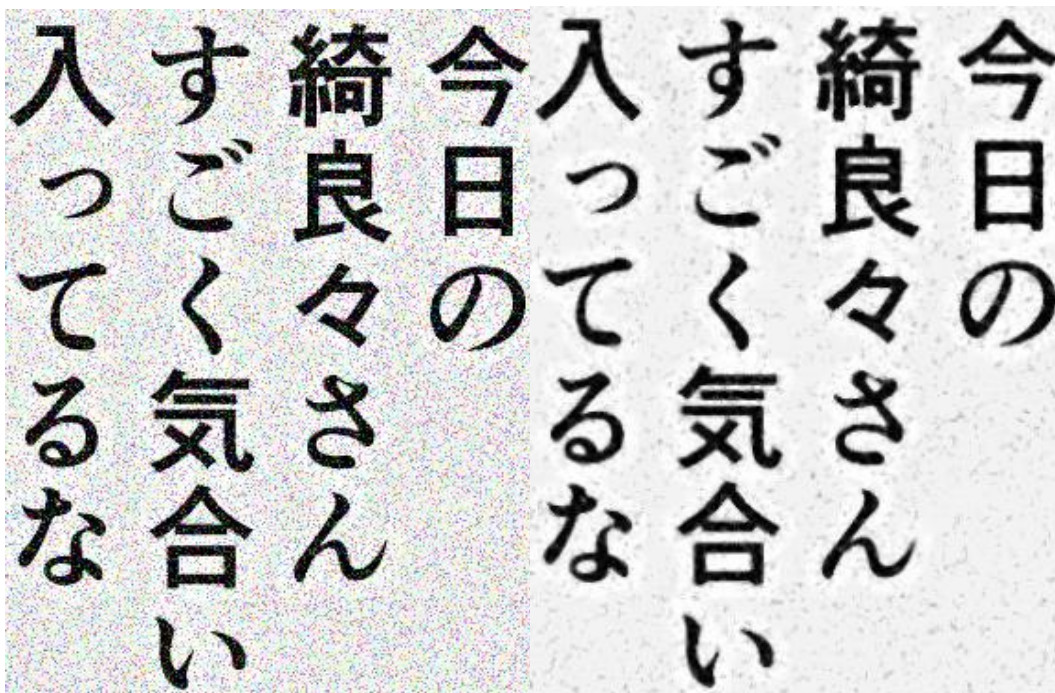


Figure 6-1 Noisy Image (left side) Denoised Image (right side)

As demonstrated in the figure above, most visible noises are removed and the result denoised image becomes easier to read. This demonstrates that the denoise function can indeed pass off as an actual denoising solution.

### 6.5 Usability Testing

To gather user's data on their experience on the application's features and functions, unmoderated remote usability testing was conducted by giving the participants the a set of tasks to be performed in the absence of supervision. The unmoderated remote usability test was done through online platforms via AnyDesk to access the application and the main communication channel used was through Discord. The table below contains the tasks and the participants were instructed to follow the items written in the table.

Usability Test Table				
No.	Task	Desc	Time taken (s)	Comments
1	Add image	Add in one of the preferred images of comic/manga into the system		
2	Translate Image	Select the preferred input and output language and start the translation		
3	Switch View	Try to switch the image view back to the original image		
4	Find Image	Find the translated image in the file explorer		

Table 6-3 Usability Test Table

A satisfaction questionnaire was also given to the user to gain more insight about their views on the application.

No.	Questions	Strongly disagree					Strongly agree				
		1	2	3	4	5	1	2	3	4	5
1	The system is easy to use										
2	I was able to complete all of the tasks in a reasonable time										
3	I felt confident in the actions I took when using the system										
4	I think that the system's functions are well integrated into the system										
5	The user interface elements feel consistent										
6	The Japanese-to-English translation is satisfactory										
7	The English-to-Japanese translation is satisfactory										
8	I am satisfied with the system										

Table 6-4 Satisfaction Survey

Around 10 participants were involved in the remote usability testing. By averaging the time taken by the participants, the scores in the tables below were procured:

Usability Test Table			
No.	Task	Desc	Average time taken (s)
1	Add image	Add in one of the preferred images of comic/manga into the system	5.62
2	Translate Image	Select the preferred input and output language and start the translation	38.79
3	Switch View	Try to switch the image view back to the original image	3.52
4	Find Image	Find the translated image in the file explorer	12.35

Table 6-5 Usability Test Table average score

No.	Questions	Average Score
1	The system is easy to use	4.2
2	I was able to complete all of the tasks in a reasonable time	4.0
3	I felt confident in the actions I took when using the system	3.2
4	I think that the system's functions are well integrated into the system	3.0
5	The user interface elements feel consistent	3.2
6	The Japanese-to-English translation is satisfactory	3.6
7	The English-to-Japanese translation is satisfactory	1.8
8	I am satisfied with the system	4.0
Total		3.38

Table 6-6 Satisfaction Survey average score

Based on the results procured from the usability test, it can be found that the translation time taken on average is 38.79 seconds, which is well within the requirement specified for the system. The total average score from the satisfaction survey is 3.38, which indicates that the users felt neutral about the system's capabilities.

## CHAPTER 7

### CONCLUSION AND RECOMMENDATIONS

#### 7.1 Achievement

Throughout the development cycle of the Final Year Project Phase 2, all of the functionalities and features made for “Comitranslate” had been written and designed according to the specifications in the project plan to develop solutions to the problem stated in the project. Kanban methodology was practiced to schedule the tasks, this practice was especially useful since multiple improvements of the same modules were constantly made to ensure the efficiency of the system. The system has achieved the following objectives :

1. To reduce the time required to translate comics and *mangas*.
2. To aid non-skilled users in performing translation operations.
3. To translate foreign languages into the user’s preferred language.
4. To develop an application that can be used to translate *mangas* and comics on the fly.

The system developed will contribute to the community primarily readers in making lesser-known comic and manga more discoverable by removing language barriers and promoting diversity in reading choices as the users do not need to understand the original language to read it using the automated translation system. "Comitranslate" may help mitigate the risk of exceptional foreign works becoming 'lost media' due to lack of translation. By making these stories accessible, we preserve valuable works from individuals like Twitter or Pixiv artists which may also help in building a bigger community for the artists.

## 7.2 Limitations

The system is not without its fair share of limitations which was brought upon by time constraints and also the lack of relevant information regarding the subject matter regarding this project.

The English OCR quality as received from the users from remote user accessibility testing was said to be terrible, this is due to the OCR's inherent inability to accurately detect texts of various fonts as it was pretrained with models with consistent font styles. As a result, some texts were mistakenly detected as numbers, such as extracting "70" from "To".

Even with MangaOCR's perfect OCR performance, the English translation proved to be unreliable at times. This is also due to the nature of the current translation technique which only takes each region of texts into consideration instead of putting texts stemming from a specific character as a whole into consideration. Some jokes also would not translate well using machine translation. Context on the characters in the literature is also needed in order to truly produce optimal translation results.

The UI, although simple, lacks polish and may prove to be confusing since there is no indicator when the system is running the translation process. But despite the UI's shortcomings it does support the display of all its needed features to the user.

The way the program overlays text onto the translated image needs rework since in its current form, it can only overlay text horizontally, the texts are also static which means the text overlay may exceed the original text bubble at times when there are way too many text squished into a small text bubble in the original manga panel.

In order to allow for text overlay, currently the program overlays blocks of white rectangles onto the place where the original text were detected. This method may prove to be annoying for some readers especially when there is graphic behind the texts, which is also another common way manga artists add texts onto manga panels.

### 7.3 Future Improvements

Potential advancements include expanding language support such as Korean and Chinese can be considered as it will allow “Comitranlate” the ability to translate Manshuas, which are similar to comics and mangas.

The UI can have some parts to be redesigned such as that it will make the intent of the buttons more clear to the user. Also, progress bars can be considered to indicate to the user that the system is underdoing the process of translating the image.

The performance of Pytesseract can use some upgrade, this can be done by extensively training a model on various comics in various fonts which helps in drastically improve the OCR accuracy of the texts.

The translation quality can be further improved by employing techniques to detect the character the texts came from, this can help in adding more context in the texts which may help in improving the translation quality, although translation of this kind will prove to be difficult as unique AI models will have to be made for it.

The program could benefit from machine image inpainting to redraw parts of the image where the original text is removed, this helps in preserving the original image’s artistic fidelity. This method will also require machine learning to achieve.

The text overlay for future iterations of this program can be upgraded to scale the text size according to the original image’s text size, also the future version of the text overlay module can be made to recognize the rotation of the original texts so that the translated text can be overlaid the same way as the original to preserve the original artist’s style of writing.

## REFERENCES

Burton, B. (2022, ). Guyver - Bring a full English translation to Western Audiences. Change.org. Available at: <https://www.change.org/p/guyver-bring-a-full-english-translation-to-western-audiences>

Drumond, C. (2018). Scrum - what it is, how it works, and why it's awesome. Atlassian. Available at: <https://www.atlassian.com/agile/scrum>.

Fox, J. (2023). 10 Best Manga That Still Don't Have Official English Translations. ScreenRant. Available at: <https://screenrant.com/best-manga-without-official-english-translations/>

Hidalgo-Tertero, C. M. (2021). Google Translate vs. DeepL. *MonTI. Monografías de Traducción e Interpretación*, 154–177. <https://doi.org/10.6035/MonTI.2020.ne6.5>

Huda, R. S., Wulaningrum, R., & Swanjaya, D. (2022). TESSERACT-OCR. *JOUTICA*, 7(2).  
 Naveenkumar, M., & Ayyasamy, V. (2016). OpenCV for Computer Vision Applications. *Proceedings of National Conference on Big Data and Cloud Computing (NCBDC'15), March 2015*, 52–56. Available at: [https://www.researchgate.net/publication/301590571\\_OpenCV\\_for\\_Computer\\_Vision\\_Applications](https://www.researchgate.net/publication/301590571_OpenCV_for_Computer_Vision_Applications)

Hegghammer, T. (2022). OCR with Tesseract, Amazon Textract, and Google Document AI: a benchmarking experiment. *Journal of Computational Social Science*, 5(1), 861–882. Available at: <https://doi.org/10.1007/s42001-021-00149-1>

Hoory, L. (2023) What is waterfall methodology? here's how it can help your project management strategy, Forbes. Available at: <https://www.forbes.com/advisor/business/what-is-waterfall-methodolog> (Accessed: 28 August 2023).

kha-white. (2023) Manga OCR. Available at: <https://github.com/kha-white/manga-ocr> (Accessed: 29 August 2023).



Lynn, R. (2021) Kanban system: Maximize time and improve efficiency, Planview. Available at: <https://www.planview.com/resources/guide/introduction-to-kanban/kanban-systemmaximize-time/> (Accessed: 10 July 2023).

Radigan, D. (2022). What is kanban? Atlassian. Available at: <https://www.atlassian.com/agile/kanban>.

Naveenkumar, M., & Ayyasamy, V. (2016). OpenCV for Computer Vision Applications. *Proceedings of National Conference on Big Data and Cloud Computing (NCBDC'15), March 2015*, 52–56. [https://www.researchgate.net/publication/301590571\\_OpenCV\\_for\\_Computer\\_Vision\\_Applications](https://www.researchgate.net/publication/301590571_OpenCV_for_Computer_Vision_Applications)

Sommerville, I. (2011). Software engineering / Ian Sommerville. — 9th ed. In Addison-Wesley (Vol. 9, Issues 978-0-13-703515–1).

Toyota (2023) Valuing the toyota production system and Lean Manufacturing, Toyota Forklifts. Available at: <https://www.toyotaforklift.com/resource-library/blog/toyota-solutions/valuing-the-toyota-production-system-and-lean-manufacturing> (Accessed: 28 August 2023).

Wiegand, S. (2018) Personal kanban part 1 - why Todo-lists don't work, HackerNoon. Available at: <https://hackernoon.com/personal-kanban-part-1-why-todo-lists-don-t-work3b5c6dc78708> (Accessed: 10 July 2023).

# APPENDIX

## APPENDIX A :Sample panels to be used for the working prototype in FYP 2



『月刊少女野崎くん』・©Izumi Tsubaki/SQUARE ENIX

Monthly Girls' Nozaki-kun “お返事”



Credit to @ppkhnKo98 from X.com/Twitter.com



Credit to @ppkhnKo98 from X.com/Twitter.com



Sample obscure manga



Obtained from the manga – Delicious In Dungeon



Credit to @Maaki\_12 from X.com/Twitter.com

## APPENDIX B :Supervisor and moderator comments on project plan

<b>Project title:</b>	ComiTranslate: Empowering Global Readership through Auto-Translated Comics and Manga	
<b>Student Name</b>	GOH YEE FONG	
<b>Supervisor</b>	Chia Kai Lin	
<b>Moderator</b>	Sneha Kanchan	
Key Assessment for Project Proposal	Supervisor Comments/Remarks	Moderator Comments/Remarks
<b>Project Description</b> - Is the problem or need to be addressed clearly presented? - Is the proposed approach or solution clearly presented and justified?		Yes. Need to be more clear on the type of model going to use for the project development.
<b>Project Scope and Objectives</b> - Is the scope of the project clearly defined? - Are the objectives of the project clearly specified? - Are the project scope and objectives appropriate for a final year project?		Yes.
<b>Literature Review / Fact Finding for Benchmarking / Verification of Project</b> - Are sources for literature review / fact finding appropriate? - Is information from literature review / fact finding relevant and adequate? - Is information from literature review / fact finding clearly presented and discussed?		Done
<b>Research/Development Methodology and Development Tools</b> - Is the methodology for the project clearly described and discussed? - Are the required development tools clearly described and discussed? - Are the stated methodology and development tools appropriate?		Need to work more upon this.
<b>Project Plan</b> - Are the phases and tasks of the project properly defined and planned? - Are the phases and tasks consistent with the methodology of the project?		Yes
<b>Initial Deliverables</b> - Are deliverables (e.g. use case diagrams and descriptions) of initial phases of the project plan included in the report?		Use case diagram needs to be more detailed.
<b>Report Structure and References</b> - Is the report organised in a logical structure? - Are references listed in accordance to Harvard format?		
<b>Language and Clarity of Writing</b> - Are the sentences concise and understandable? - Are there spelling and grammar issues?		