



**CRESCENDO®**  
INTERNATIONAL  
COLLEGE

**DIPLOMA IN COMPUTER SCIENCE  
(SYSTEM DEVELOPMENT)**

**COMPUTING PROJECT REPORT  
DCS 5604**

**VOICE MEMO ASSISTANT**

**CHAU WAN QIAO  
DCS2309265  
051120-01-0570**

**May 2025**

## 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 diploma or award at Crescendo International College or other institutions.

A handwritten signature in black ink, consisting of a large, stylized 'C' followed by a smaller, more complex mark.

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

Student's Name: Chau Wan Qiao

## **ACKNOWLEDGEMENT**

I would like to express my sincerest gratitude to all those who have provided support and assistance in the completion of this project report.

Firstly, I would like to thank my supervisor, Mr Simon, for his guidance and encouragement throughout the completion of this report. These suggestions helped to enhance and ensure the quality and direction of this report.

In addition, I would also like to thank to my classmates Tai Xing Rou and Hiew Pei Qiao for their help in both the development and research of this project. To illustrate, there are always being some situations and problems that arise at the beginning when configuring the development environment, and their willingness to help was crucial to the contribution of this project.

Besides that, I would also like to thank the many community members who provided free open-source code to facilitate the progress of this project.

Finally, I would like to thank my family and friends for their support and encouragement throughout the project. They have served as my moral support in completing and continuing this project.

Once again, thank you all for your supporting.

## **ABSTRACT**

The purpose of system development is usually for satisfy the needs in the market. When the products or solutions in the market cannot fully address and satisfy the needs of the users, there is a need to create new or advanced products that can satisfy these needs. The aim of this project is to develop an efficient and advanced intelligent voice-memo-assistant to improve the functionality that lack in the market, and also to provide more choices to meet the needs of more people with specific needs. The theme of this project is to use the user's voice input and convert it into text, and then process and analyse the text in a specific way based on the natural language classification model of artificial intelligence. Then, the content is categorised into four different categories of memo types. These memos will be composed of intuitive headings and clear timings, allowing users to know what they need to accomplish. In short, this is a voice memo assistant, which aims to remind and record the user's must-do items. To illustrate, this project will also support English and Chinese language inputs, which makes this project more flexible. In addition, this project is composed of React Native as the framework, and PHP language and Flask as the back-end to deal with complex voice input and text recognition functions. As a conclusion, I hope the development and success of this project can benefit more users and make things more efficient.

# TABLE OF CONTENTS

## Contents

<b>ACKNOWLEDGEMENT</b> .....	ii
<b>ABSTRACT</b> .....	iii
<b>TABLE OF CONTENTS</b> .....	iv
<b>LIST OF FIGURES</b> .....	vi
<b>LIST OF TABLES</b> .....	viii
<b>CHAPTER 1</b> .....	1
<b>1.1 Problem Statement</b> .....	1
<b>1.2 Objective</b> .....	2
<b>1.3 Scope</b> .....	3
<b>1.4 Gantt Chart</b> .....	6
<b>1.5 Conclusion</b> .....	10
<b>CHAPTER 2</b> .....	11
<b>2.1 Project Motivation and Industry Challenges</b> .....	11
<b>2.2 Similar Existing System</b> .....	12
<b>2.2.1. Application 1- Apple Voice Memos</b> .....	12
<b>2.2.2. Application 2- Microsoft OneNote</b> .....	13
<b>2.2.3. Application 3- Microsoft To Do</b> .....	14
<b>2.2.4. Comparison of Existing Similar System</b> .....	15
<b>2.3 Conclusion</b> .....	16
<b>CHAPTER 3</b> .....	17
<b>3.1 Introduction</b> .....	17
<b>3.2 Phase 1: Planning</b> .....	19
<b>3.3 Phase 2: Design</b> .....	20
<b>3.4 Development</b> .....	27
<b>3.5 Testing and Evaluation</b> .....	29
<b>3.6 Deployment</b> .....	32
<b>3.7 Maintenance</b> .....	34
<b>3.8 Conclusion</b> .....	35
<b>CHAPTER 4</b> .....	36
<b>4.1. Introduction</b> .....	36
<b>4.2. New Mobile Application – Voice Memo Assistant App</b> .....	37
<b>4.3. Testing and Evaluation</b> .....	53
<b>4.4. Conclusion</b> .....	58
<b>CHAPTER 5</b> .....	59
<b>5.1. Summary Project</b> .....	59

<b>5.2.</b>	<b>Strength and Weakness.....</b>	<b>61</b>
<b>5.3.</b>	<b>Recommendation and Future Enhancement .....</b>	<b>64</b>
<b>5.4.</b>	<b>Conclusion .....</b>	<b>66</b>
<b>REFERENCES .....</b>		<b>68</b>

## LIST OF FIGURES

### Figure

Figure 1 Details of Gantt Chart 1 .....	7
Figure 2 Gantt Chart 1 .....	7
Figure 3 Details of Gantt Chart 2 .....	8
Figure 4 Gantt Chart 2 .....	8
Figure 5 Apple Voice Memos (Application 1) .....	13
Figure 6 Microsoft OneNote (Application 2) .....	14
Figure 7 Microsoft To-Do (Application 3) .....	15
Figure 8 Diagram for Waterfall model processes .....	18
Figure 9 Context Diagram .....	20
Figure 10 DFD diagram .....	22
Figure 11 Login Page .....	23
Figure 12 Sign up page .....	24
Figure 13 Home page .....	24
Figure 14 Confirm memo page (POP OUT) .....	25
Figure 15 Memo Display page .....	25
Figure 16 app.py file screenshot .....	28
Figure 17 Unit Testing example .....	29
Figure 18 Integration testing example .....	30
Figure 19 User testing example .....	31
Figure 20 api.js screenshot .....	32
Figure 21 Maintenance for code comment example .....	35
Figure 22 Sign Up Page 1 .....	38
Figure 23 User Information Database .....	38
Figure 24 Sign Up Page 2 .....	39
Figure 25 Login Page 1 .....	39
Figure 26 Home page 1 .....	40
Figure 27 Home Page 2 .....	41
Figure 28 Home Page 3 .....	41
Figure 29 Home Page 4 .....	42
Figure 30 Confirm memo page .....	43
Figure 31 Memo saved Alert .....	43
Figure 32 Memo Details 1 .....	44
Figure 33 Memo Details Page 2 .....	45
Figure 34 Daily category Page .....	45
Figure 35 Memo Details Page 3 .....	46
Figure 36 Memo Details Page 4 .....	46
Figure 37 Confirm Delete Alert .....	47
Figure 38 Confirm Delete Memo Page (Voice input) .....	48
Figure 39 The backend information .....	48
Figure 40 View Memo Page .....	49
Figure 41 Query_Today backend information .....	49
Figure 42 The Memo details result Page .....	50
Figure 43 Black Box testing .....	53

Figure 44 Time Extraction Problem.....54

Figure 45 Time Extracted logic in both language model.....55

Figure 46 Inaccurate Classification Result.....55

Figure 47 The Training text included.....56

Figure 48 Training model result .....56

Figure 49 Result of the memo categorization .....57

Figure 50 The connection on Plesk .....60

Figure 51 Voice Memo Assistant application poster .....67



## LIST OF TABLES

### Table

Table 1 Summary of Comparison System.....	15
Table 2 Context diagram entities .....	21
Table 3 Software requirements table .....	27
Table 4 Hardware requirement table.....	28

# CHAPTER 1

## INTRODUCTION

### 1.1 Problem Statement

#### i. Lack of intelligent categorisation

Currently there is no perfect and efficient intelligent classification of memos on the market, most of the systems only support voice input to text and do not have the intelligent classification function. The lack of memo classification and partition function makes users unable to quickly find the corresponding records when recording events. Manual sorting and classification will cause inefficiency and waste of time.

#### ii. Fragmentation of product content and functionality

Incomplete functionality of the system or the users need to switch between multiple applications will lead to user inconvenience and poor user experience. Current products on the market lack integration that require users to manually process and categorise memos. As a result, the user's needs cannot be fully satisfied, the dependence on the system and the need to decline.

#### iii. Unable to record quickly when busy

When users need to handle multiple tasks at the same time, they can't quickly use text input to record important things through the keyboard. For example, when driving or in a meeting, users will be distracted and focused on another task and often can't record ideas in time. This also leads to oversights and omissions. Example, misremembering the time of a meeting, which make the meeting conduct unsuccessfully.

## **1.2 Objective**

### **a) Refinement of deficiencies in intelligent categorisation**

Analysing the existing memo applications in the market reveals the inadequacies and lack of functionality of these systems. The goal of this project is to make it easier for users to categorise the things they need to do without having to do it manually. In this way, users do not have to spend time and effort to manually record reminders and classification, intelligent recognition of text can be automatically classified into the appropriate category, reducing the operating time. And the practicality and convenience of the memo have been improved.

### **b) Development of fully functional system**

In order for users to keep track of all the things that need to be done and the reminder function in one app, designing a smart voice memo with full functionality is needed. This will allow users to go through the voice input, and the text recognition and smartly categorise the memo. This ensures that the user is presented with the memo information in a more intuitive way for subsequent management. Benefit for realising this project is about improving the completion rate of tasks, reducing omissions and negligence whether it's in life or at work.

### **c) Improve the usability of the system in real world**

The main role of a memo is to be responsible for reminding the user of things that need attention or need to be done. And the user's dependence on memos is linked to the usefulness of the app. Therefore, the purpose of this project is to realise that memos can be used to record any matter in daily life, study and work. Since the system supports bilingualism (English and Chinese), it is possible to recognise and record memo even in different languages in real life, and the quick recording of important things like in daily life when the user is walking or driving can also improve the efficiency and convenience of the user. This optimises the user experience, especially for those with weak memories or the elderly, who can use memo to check what needs to be done.

### **1.3 Scope**

This section is to define the research and development direction of the project, discuss the core features and target audience included in the project. I will also list the features and content that are not included in this project to prevent the project from straying from its focus and direction.

#### **a) Project Scope Inclusion**

The core functionality of this project is based on voice input and converting text and using Nature Language Processing (NLP) to process and intelligently categorise memo categories to help users not have to manually partition and enter memos. The following are the frameworks and technologies used in the project:

##### **i. Speech input recognition and text replacement module**

In this project, I'll be using a React Native front-end to allow for voice recording functionality, and Flask as the back-end to process the text.

##### **ii. Natural Speech Processing and Classification Module**

Regarding the intelligent recognition of text entry memos, I used a combination of Python and SpaCy to train a text classification model, and then semantically analysed and categorised the text. These items are grouped into predefined categories, namely 'Daily', 'Work', "Study" and 'Others'.

##### **iii. Database and PHP Linking Relationships**

Regarding the user and memo data, I used PHP to link MySQL Workbench to store all the user data stored on the application for the system's functionality and interaction to work properly.

## **b) Project Scope Exclusion**

To ensure technical and time manageability, the following features and modules will not be included in this project. These additional features can be considered for future builds and extensions.

### **i. Multi-users collaboration and sharing capabilities**

Users can't share a memo to other users to view it together and edit it at the same time via connection or invitation. These features require data control and synchronisation between users, which involves complex front-end and back-end logic. This project is focusing on the user's own schedule, so this feature is not a requirement and is not in the development plan.

### **ii. Image recognition text function**

The main core content in this project is to record the memo by voice input text, so OCR text recognition and image functions are not supported in this project for the time being. However, if there is a need to expand more functions in the future, we can also add the image recognition function to enhance the convenience and practicality of the user.

### **iii. Link ability of third-party applications**

This project is more based on standalone application functionality, so connecting and integrating functionality with other applications or platforms is not considered. In addition, also because of technical and time issues, integration functions with other third-party platforms are not supported for the time being.

### **c) Target Audience of project**

#### **i. Primary Users**

This project is aimed to the use of students and office workers. Students can keep track of course assignments, revisions and deadlines of homework for better time management. This will avoid overdue or missed assignments. Besides that, for office workers or entrepreneurs, the app can be used to keep track of daily routines and schedules, such as meeting records and client meeting times. This way users can see their time and work schedule clearly.

#### **ii. Secondary Users**

The sub-user groups in this project are clustered around data analysts and IT technicians. The data analysts are responsible for analysing voice content entered by multiple users to gain a deeper understanding of user behavioural patterns, memo sorting content and user activity. This is to provide data support for subsequent project adjustments and optimisations. In addition, the IT technicians focus on software testing, system maintenance, database management and technical evaluation. This is to ensure that the application can be utilised properly and efficiently by the user community, and that the system can be fixed in a timely manner when there are bugs in the code.

## **1.4 Gantt Chart**

Gantt Chart is a visual project planning timeline primarily designed to help break down huge tasks into smaller for setting up short-term goals.

Gantt Chart is typed with specific tasks and times on the left vertical, and a visual representation of the project's period and duration on the right horizontal. This facilitates the user to be clear about what needs to be done in each time period. The elements in Gantt Chart including tasks, timelines and task bars, which allow all the participants of the project to follow the preordain scheduled, avoiding task overruns and omissions.

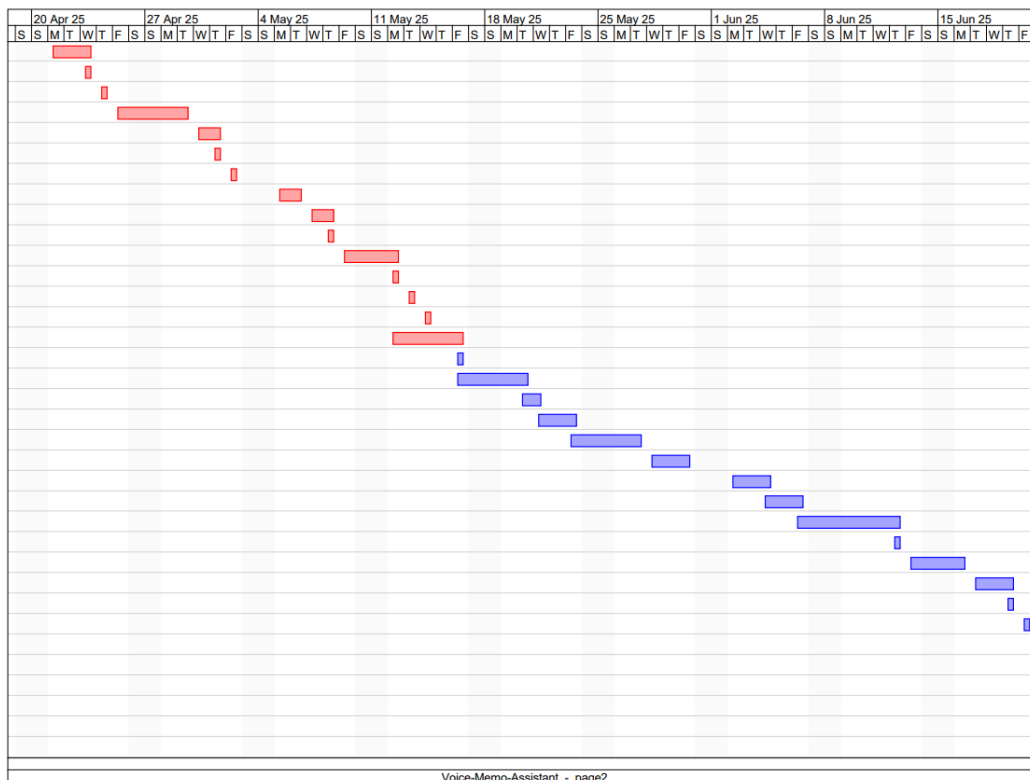
Besides that, another focus of Gantt Chart is to track the progress of the project. It displays indicators of work in progress, with tasks shaded in contrasting colours to indicate whether they are completed or not. Furthermore, each task can be detailed with allocated resources and the name of the person who performed it, for better planning and allocation of human resources.

In addition, Gantt Chart will indicate the dependencies between tasks. For example, you must complete task A before you can perform task B. The advantage of this is to avoid task conflicts and overlapping of resources used. It ensures that the tasks progress in a normal chronological order, avoiding the problems that can lead to unsuccessful development of the system if the tasks are disorder.

	④	Name	Duration	Start	Finish	Predecessors	Resource Names
1		Decide the project title	3 days?	4/21/25 8:00 AM	4/23/25 5:00 PM		
2		Confirmation of the main f...	1 day?	4/23/25 8:00 AM	4/23/25 5:00 PM		
3		Project Approval and Ment...	1 day?	4/24/25 8:00 AM	4/24/25 5:00 PM		
4		Complete project proposal	3 days?	4/25/25 8:00 AM	4/29/25 5:00 PM		
5		Background research and ...	2 days?	4/30/25 8:00 AM	5/1/25 5:00 PM		
6		Functional requirements a...	1 day?	5/1/25 8:00 AM	5/1/25 5:00 PM		
7		Complete the Gantt Chart	1 day?	5/2/25 8:00 AM	5/2/25 5:00 PM		
8		Building an application fra...	2 days?	5/3/25 8:00 AM	5/6/25 5:00 PM		
9		Complete voice recording ...	2 days?	5/7/25 8:00 AM	5/8/25 5:00 PM		
10		Connect Flask as a backend	1 day?	5/8/25 8:00 AM	5/8/25 5:00 PM		
11		Complete the real-time co...	2 days?	5/9/25 8:00 AM	5/12/25 5:00 PM		
12		Create database in MySQL...	1 day?	5/12/25 8:00 AM	5/12/25 5:00 PM		
13		Using PHP connect to back...	1 day?	5/13/25 8:00 AM	5/13/25 5:00 PM		
14		Download and set the Spa...	1 day?	5/14/25 8:00 AM	5/14/25 5:00 PM		
15		Complete the report[20%]	5 days?	5/11/25 8:00 AM	5/16/25 5:00 PM		
16		Train the dual-language m...	1 day?	5/16/25 8:00 AM	5/16/25 5:00 PM		
17		Build the memo module	3 days?	5/16/25 8:00 AM	5/20/25 5:00 PM		
18		Using SpaCy connect back...	1.5 days?	5/20/25 8:00 AM	5/21/25 1:00 PM		
19		Complete the "Memo" func...	3 days?	5/21/25 8:00 AM	5/23/25 5:00 PM		
20		Create the To-Do list module	3 days?	5/23/25 8:00 AM	5/27/25 5:00 PM		
21		Create the "Today?" module	3 days?	5/28/25 8:00 AM	5/30/25 5:00 PM		
22		Complete the report[40%]	3 days?	5/31/25 8:00 AM	6/4/25 5:00 PM		
23		Complete the "Today?" fu...	3 days?	6/4/25 8:00 AM	6/6/25 5:00 PM		
24		Create the other services ...	5 days?	6/6/25 8:00 AM	6/12/25 5:00 PM		
25		Complementary functional ...	1 day?	6/12/25 8:00 AM	6/12/25 5:00 PM		
26		System Testing	2 days?	6/13/25 8:00 AM	6/16/25 5:00 PM		
27		Complete the report[60%]	3 days?	6/17/25 8:00 AM	6/19/25 5:00 PM		
28		Record the testing result	1 day?	6/19/25 8:00 AM	6/19/25 5:00 PM		
29		Optimization system	1 day?	6/20/25 8:00 AM	6/20/25 5:00 PM		
30		Debugging system and opt...	3 days?	6/21/25 8:00 AM	6/25/25 5:00 PM		
31		Generate evaluation report	2 days?	6/26/25 8:00 AM	6/27/25 5:00 PM		
32		Complete the report	5 days?	6/28/25 8:00 AM	7/4/25 5:00 PM		
33		Complete demonstration vi...	2 days?	7/4/25 8:00 AM	7/7/25 5:00 PM		
34		Complete PPT	1 day?	7/8/25 8:00 AM	7/8/25 5:00 PM		
35		Develop the introduction ...	4 days?	7/8/25 8:00 AM	7/11/25 5:00 PM		

Voice-Memo-Assistant - page1





Figure 1 Details of Gantt Chart 1



Voice-Memo-Assistant - page2

Figure 2 Gantt Chart 1



		Name	Duration	Start	Finish	Predecessors	Resource Names
36		Check all content and sour...	1 day?	7/12/25 8:00 AM	7/14/25 5:00 PM		
37		Submit all projects content	1 day?	7/15/25 8:00 AM	7/15/25 5:00 PM		
38		Presentation	1 day?	7/24/25 8:00 AM	7/24/25 5:00 PM		

Voice-Memo-Assistant - page5

Figure 3 Details of Gantt Chart 2

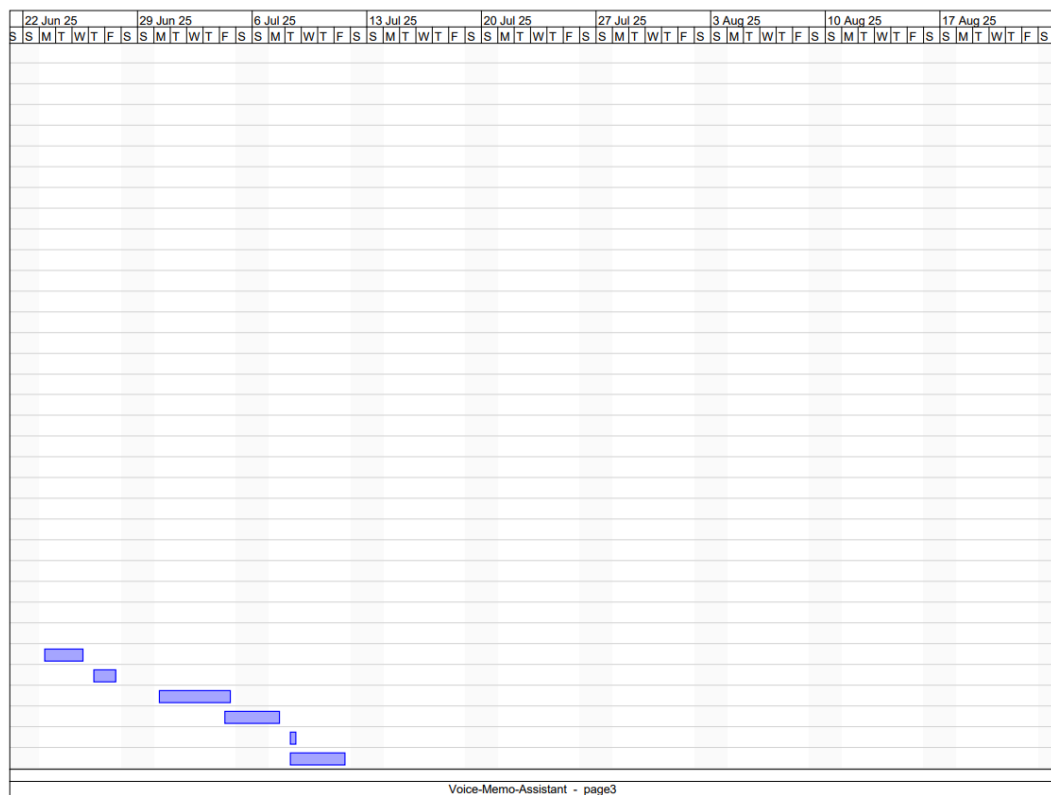


Figure 4 Gantt Chart 2

### **Steps to create a Gantt Chart:**

#### **1. Communicate and clarify the scope of this project**

Before the project is developed, I need to communicate with my lecture about the core content and tasks of the project, and clarify the functional scope of the project to avoid wasting resources.

#### **2. Break down and list all the activities and tasks to be completed**

Before I start the project, I need to break down the task into small tasks to set up short-term goals. The benefit of short-term goals is making people more motivated to complete them, and there is a greater sense of achievement when the small tasks are completed to support people to continue the tasks. Writing out all the tasks clearly helps me to know what I need to do.

#### **3. Identify dependencies between tasks**

Next, we need to think about each task, which task will affect the completion of another task and the order of task execution need to be understood and recorded. This allows me to prioritise and sequence tasks.

#### **4. Reasonable allocation of deadlines for each task**

I will allocate the task time according to the degree of difficulty for each project to ensure that this project can be completed before the deadline.

#### **5. Monitor task progress and update progress**

Since the Gantt Chart is made before the task is started, the time and progress of the task may vary at a later stage. So, it is very important to update the task progress and optimise it to give me a better understanding of the current task and what I should do subsequently.

As summary, by using Gantt Chart that makes the person in charge can roughly plan and understand exactly how to progress the work content before the project starts. By including all available resources and human resources, arranging them in a way that makes the project run smoothly and efficiently. Gantt Chart can also simplify team communication by eliminating the need to double-check their own work instead of using the Gantt Chart to view the details of their own work.

## **1.5 Conclusion**

In this chapter the problem statement, objectives, project scope and target audiences of the project are discussed, and the smart memo applications still not fulfil the needs of most users in the market that explored. By discussing the core features of the project, the direction and objectives of the project development are clarified. Audience groups and user needs are also identified in this section. In addition, I also used Gantt Chart to visualise the deadlines of the tasks, arrange the division of labour for the project and break down the tasks to be executed one by one. This is to ensure that the project will be completed before deadline. The in-depth analysis in this chapter builds the foundation for the subsequent development direction, clarifying the vision and execution steps of the project. The next chapter will discuss literature review, focusing on comparing and analysing existing memo application on the market. By combining and eliminating their strengths and weaknesses, aimed to develop a more perfect memo software.

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 Project Motivation and Industry Challenges**

Smart memo assistant is designed to enhance user productivity and allow users to keep track of sudden inspirations and things that need to be done in future. This helps to refresh and strengthen the user's memory. According to many literatures, people rely more on short-term memory in their daily lives, so if users often communicate verbally and need to follow up on tasks in their work, there is often needed a software that allows them to take quick notes.

The reason that inspired me to develop a voice memo assistant was that the shortcomings and taking notes manually led me to realise that I needed application that supported rapid and quickly note-taking. Paper notes are now becoming obsolete due to the fact that they need to be handwritten by human and don't distinguish well between each different category of memo. In addition, users can't search for keywords to find notes quickly, relying entirely on their annotation and memory to find them. This undoubtedly wastes the user's time and energy.

The emergence of voice memo can be a good solution to the current problem. It allows users to upload audio through verbal notes with text conversion and intelligently categorise the content. Like daily tasks can be differentiated from study tasks. The advantage is that users can quickly find the content according to the label classification, it helps to save time and energy. This feature also allows users to quickly record important content in a short period of time to avoid missing and omitted content. Lastly, voice memos also allow users to view and edit content on their phones at any time, which is a drawback that paper notes cannot compensate for.

The obvious benefit of voice memos is the quick recording and intelligent categorisation of content. This feature is suitable for busy office workers or students to record the planning of their work and studies. The same competitor applications on the market often lack the all-in-one feature, such as having the recording function but not supporting smart categorisation, or supporting smart categorisation but not being able to convert voice to text. These imperfections lead to users not being able to rely on the application.

The project I want to develop is to solve the lack of functionality of the current software. The current software on the market that supports speech-to-text is unable to achieve perfect accuracy and to deal with complex linguistic structures. This is due to the

fact that different people speak with different habits and grammar, which may cause the system to misrecognise the situation. In order to solve this problem, more diversified data sets should be provided to train more language models, so that the system can keep learning the language order intelligently to better understand what the user wants to express. In addition, background noise may also affect the recognition of text. For example, in a noisy crowd situation, the system may not be able to recognise the user's voice well. To solve this problem, the background noise can be reduced to highlight the user's voice so that the system can better recognise the text. Then, Natural Language Processing techniques are used to analyse the user's utterances and phrases of different users. As an example, 'I went to the supermarket to buy groceries today' and "Today, I bought groceries in supermarket". Although the meaning of these two sentences is same but the ordering of the phrases is different, which may affect the system's judgement error.

As a conclusion, under the current fast-paced life, most users not only need a software responsible for recording, they needed a memo assistant with intelligent classification and understanding of user needs. In the past, voice memos only allowed users to use the recording function, but could not focus on extracting and categorising the content. The project I want to develop, Voice-memo-assistant project which enable to provide the function of using NLP to intelligently recognise and classify memos after converting speech to text. It will execute smart categorisation function and time reminders for different types of tasks. This will greatly improve user efficiency and satisfy users who need to execute multitask at the same time.

For the future implementation and expectation of this project, I hope that it is not only a simple memo application, but also as an assistant responsible for managing the user's daily tasks, focusing on bringing convenience and efficiency to the user's application.

## **2.2 Similar Existing System**

In this chapter, I will compare the memorandum applications on the market that are often known and used by users. By analysing and comparing the core features, strengths and weaknesses, I will more be understanding the reasons that why user chosen these applications. In this way, I will be more clearly define about the purpose of the project and the user's needs when developing the project to fill the functionality gap of similar competing products in the market.

### **2.2.1. Application 1- Apple Voice Memos**

It is a voice memo software built-in IOS system, suitable for people who often attend the discussion meetings or lectures to record important contents. It provides recording function and allows users to edit the audio and upload it to iCloud directly in the software. Its

advantage is that the simple and user-friendly interface allows users to operate it directly, making it ideal for those who need to replay voice content multiple times. However, its disadvantages are also very obvious, it does not support the speech-to-text conversion function, and does not have the function of tagging and categorising memos. In addition, once there are too many files, the user cannot find the target file, but only relying to the file name and timestamps to find.

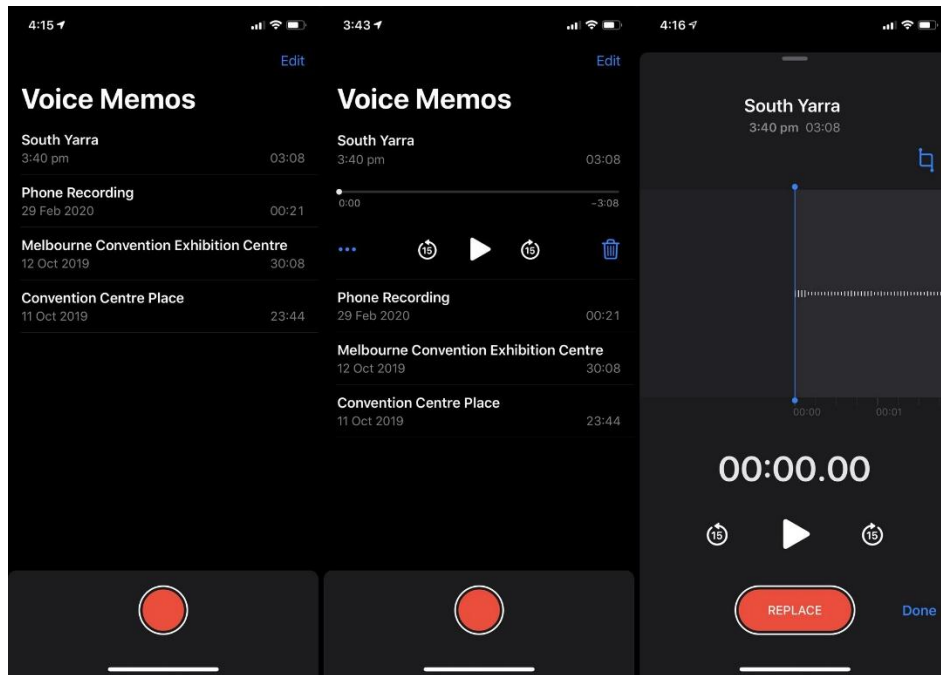


Figure 5 Apple Voice Memos (Application 1)

### 2.2.2. Application 2- Microsoft OneNote

OneNote is a memo software that most students and business people used. It is an expandable page form and flexible memo or note software. Unlike the Apple Voice Memos, its advantage not only supports audio recording, but also supports speech-to-text conversion function in some versions. In addition, it can also insert images and URLs for users to record different types of notes. As a memo application, it is considered the most recognised flexible memo software by the public. However, its weakness is it does not set up an intelligent classification system, it requires users to partition different content memos on their own. This also leads to some users who need to record notes quickly cannot be well satisfied. Other than that, it does not support the reminder function, which does not allow users to follow up on their daily tasks well.

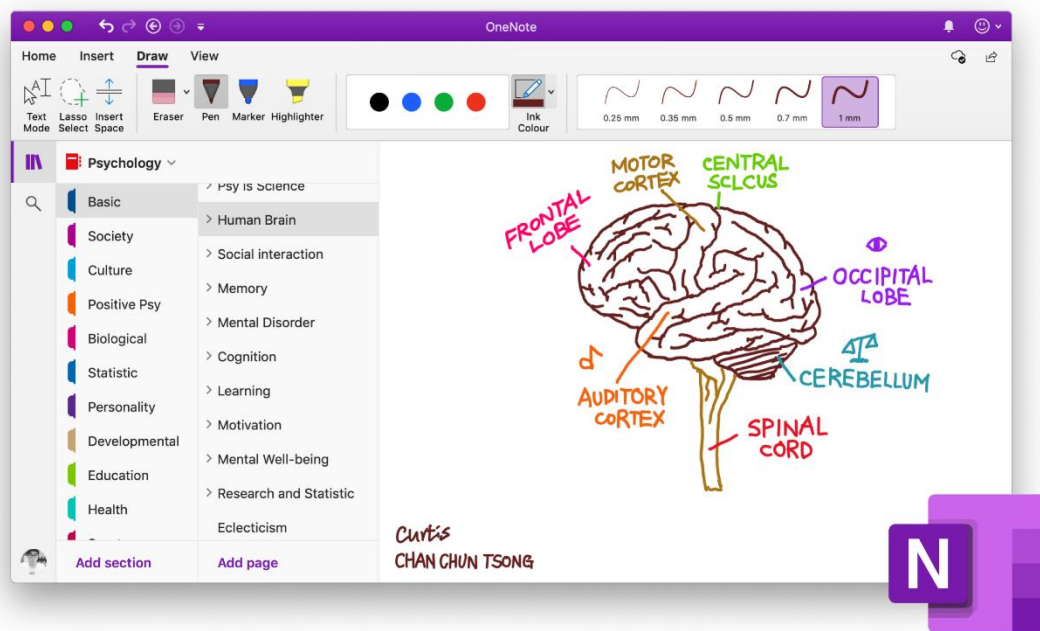


Figure 6 Microsoft OneNote (Application 2)

### 2.2.3. Application 3- Microsoft To Do

For Microsoft To-Do software, it is a list-based type of task management application than a note-taking type compared to the previous applications. Its able to set reminders and deadlines for tasks, as well as the ability to set recurring reminders. This feature is suitable for people who have to deal with heavy tasks on a daily basis. Examples, students and business people. These types of people need to plan their time and tasks tightly, so this software is more suitable to organise their daily schedule and time for better management. In addition, Microsoft To-Do supports task creation and task partitioning to make it easier for users to quickly find content through different categories. Although the advantages are outstanding and excellent, but it doesn't support voice input, speech-to-text and smart categorisation features. This is certainly not friendly and convenience to some working people who need to record content quickly and urgently, reducing usability during mobile scenarios and multitasking scenarios.

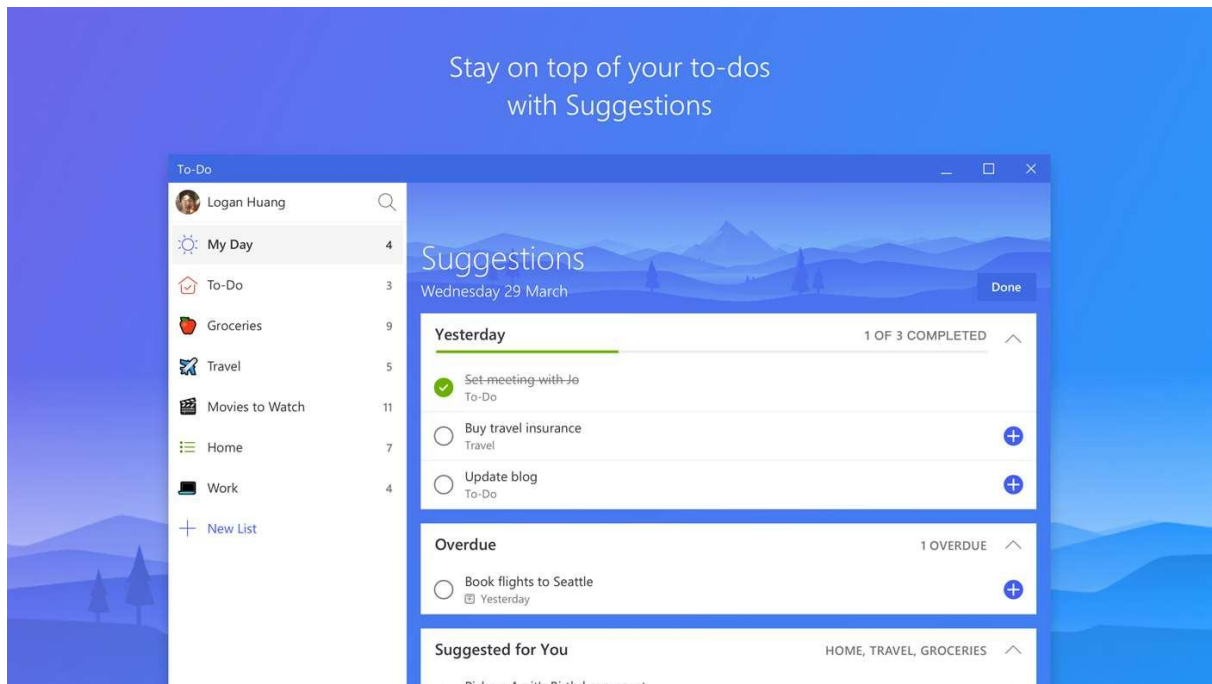


Figure 7 Microsoft To-Do (Application 3)

#### 2.2.4. Comparison of Existing Similar System

Table 1 Summary of Comparison System

No.	Features	Existing Similar System			New System (Voice-Memo-Assistant)
		Apple Voice Memos	Microsoft OneNote	Microsoft To Do	
1.	Audio Recording	✓	✓		✓
2.	Speech-to-Text		✓		✓
3.	Categorization		✓	✓	✓
4.	Reminders			✓	✓
5.	NLP-based Understanding				✓



### **2.3 Conclusion**

In this section, I explore the functional analysis of existing systems based on voice memo and task planning applications. Each of these existing applications has its own characteristics and salient features to attract different audience segments with their own strengths. Although these applications help users to improve their efficiency and productivity in different areas, but the lack of an integrated system does not allow most users to meet their needs. To illustrate, Apple Voice Memos lacks of tagging, speech recognition to text, and intelligent categorisation, making it impossible for users to find the target content quickly. For Microsoft OneNote that supports speech-to-text and content partitioning. However, the inability to intelligently categorise and the lack of task reminders are still a pain point. In addition, Microsoft To-Do also lacks of function for voice interaction although it supports task reminder and marking function. All in all, these apps lack complete functionality and only offer relatively single and specific features for users.

Instead, I wanted to develop a voice memo system to combine the features of the three existing systems above and integrate these features into a new system. This project is an all-in-one system that includes voice input, real-time speech-to-text, NLP recognition and classification, and task reminders, and is designed to be more efficient and time-saving for users. In order to fit the needs of a faster-paced and multi-tasking environment.

As a conclusion, after an in-depth analysis of problem identification and different applications, the next section will discuss about how to design and develop a more fully-featured voice memo system approach.

## **CHAPTER 3**

### **METHODOLOGY**

#### **3.1 Introduction**

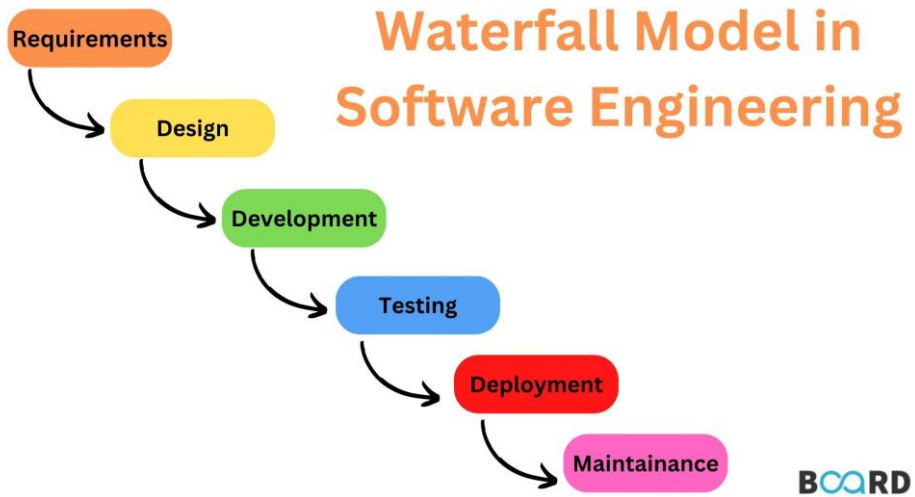
In this project I developed a voice-memo-assistant application, whose primary objective is to help users create and manage memos faster and easier, in order to making task reminders faster and intuitive that the users need to complete. This development uses SpaCy as a base for speech text recognition, Flask and NLP language to achieve the user from voice entry to the successful storage of the memo in the system and database.

To illustrate, I mainly use React Native to develop this Voice-Memo-Assistant application. Flask is used as the backend to receive and send user inputs to SpaCy for text classification. Furthermore, for storing user information and content in the database, I used PHP to connect MySQL workbench and the system for data storing.

Furthermore, in order to provide more accurate and intelligent memo classification, I also added two training models in two different languages to allow the system to better evaluate and analyse what users say.

Besides that, the main development tools used in this project are Atom as the code editor, Expo Go for real-time testing and MySQL workbench for managing and storing the data to shape and test this voice memo assistant.

Moreover, the methodology that I used in this project is the Waterfall model. Waterfall is a sequential approach to project management that works on the timing, requirements and outcomes of each phase to ensure that the user is aware of the tasks that need to be accomplished at each phase. Based on the fact that the system development phases have been defined clearly and carried out in a strict step-by-step and sequential manner, the Waterfall model is more suitable for this project as there are clear deliverables and schedules for each phase. Since the Voice Memo Assistant is a stable product with less changing requirements, the Waterfall model is more suitable for this project than other methodologies because of its structured approach.



*Figure 8 Diagram for Waterfall model processes*

### **3.2 Phase 1: Planning**

In the early stages of the development phase, the most important thing is to understand the target audience and user needs to design a perfect system that meets the conditions. In the planning stage, I will write down the tasks that need to be done in each period and strictly enforce the deadlines to complete the project.

To better understand the needs of my target users, I actively participated in online forums and discussion groups to get users' opinion and comments on the existing similar system listed in Chapter 2. This has helped me to conceptualise a more innovative and intelligent voice memo assistant that meets the users' needs. Based on this feedback, I found that users need a smart memo that can automatically categorise memos by using voice input, a feature that allows users to take notes anytime, anywhere and avoid omissions under the pressure of multitasking.

To illustrate, through requirements gathering and analysis, I identified the main functional and non-functional requirements in this project. The main functionality is ability to convert user's voice input to text and intelligently categorise memo categories using NLP language: Examples, Study, Daily, Work and Others. The non-functional requirements are some regular updates for efficiency and system usability, such as more intuitive display of tasks to be processed so that users have a clearer understanding of what they need to accomplish.

Furthermore, there are some limitations to this project. As an explanation, I will classify these factors into 3 types. Firstly, there are physical constraints, as the software development requires multiple pages and applications to be opened at the same time, which requires a highly configured laptop to support the development. Besides, the project also requires an android mobile phone to support the project testing and functionality testing. Secondly, for the financial constraints, I chose to use free and open-source services and tools for the development of this project. Due to my own limited ability and lack of experience in machine learning and mobile app development, I need to spend more time and effort in searching for information and related project development content on the internet.

Apart from that, due to the time and technical constraints, the scope of this project is rather limited, like some features such as third-party connectivity, image text recognition and multi-user editing are not involved in this project. The Speech-To-Text and memo categorisation are the main focuses of this project.

Finally, for the timeline of the project, I used Project Libre to create a Gantt Chart in order to have a clearer idea of what needs to be done in each step and the direction of the project. In the planning phase I also defined the expected results of the project. Examples, clearer and more intuitive UI design and accurate categorisation results. This is essential for the design phase to ensure that the project does not deviate from its goals.

### 3.3 Phase 2: Design

After planning stage, I understand more about the functional design and requirements that I need to fulfil in this project. Therefore, in the design stage, I will use the information that I have to plan the system architecture, data flow and UI design to really implement the system. This helps to visualise the core functionality and interaction of the system, which is very important for the actual development phase. In the design phase, I focus on the actual components of the system and how they interact with each other, so that the user can understand the overall flow of the system. Then, I also analyse and manage the small parts of each component to ensure that each component is working properly. I will use context diagram, data flow diagram and UI design to illustrate this information.

#### a) Context Diagram

The structure of this project is represented in the context diagram provided below, identifying the system boundaries, external actor and data flow. The purpose of the context diagram is to represent the flowchart of how the system exchanges data with users and external services.

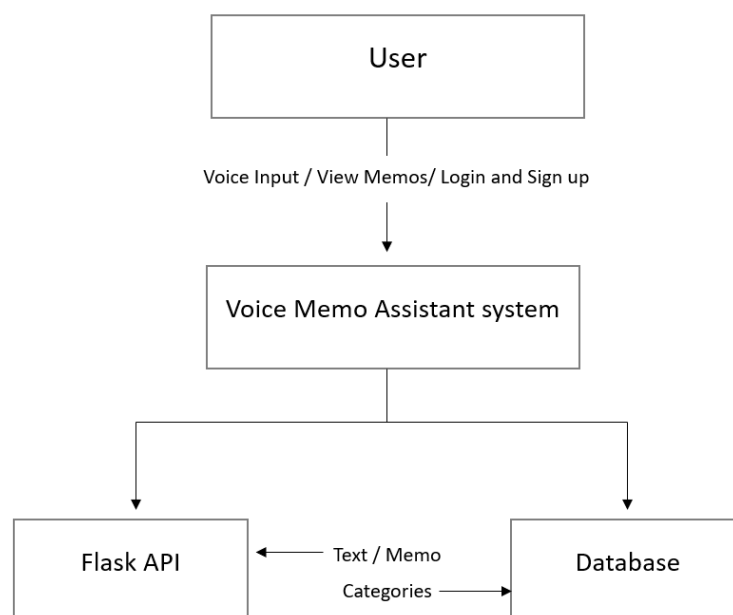


Figure 9 Context Diagram

*Table 2 Context diagram entities*

External Entities	Functions
User	Interact with system for inserting memos
Flask	Handles communication between the frontend and internal logic, and also the Speech-to-Text function
Database	Stores the user information, memos and the categories.

As an explanation, the main dataflow is that the user inputs speech then the application receives the user input and sends the text to the backend. Then SpaCy model gives the classification result and returns to backend. Memo's data is sent from backend to database and stored. Finally, the user can view the memo information with the data retrieved from database.

## b) Data Flow diagram

DFD is a graphical representation of data flow, used to show incoming data flow, outgoing data flow and data storage process. Its main purpose is to visualise the steps in a project.

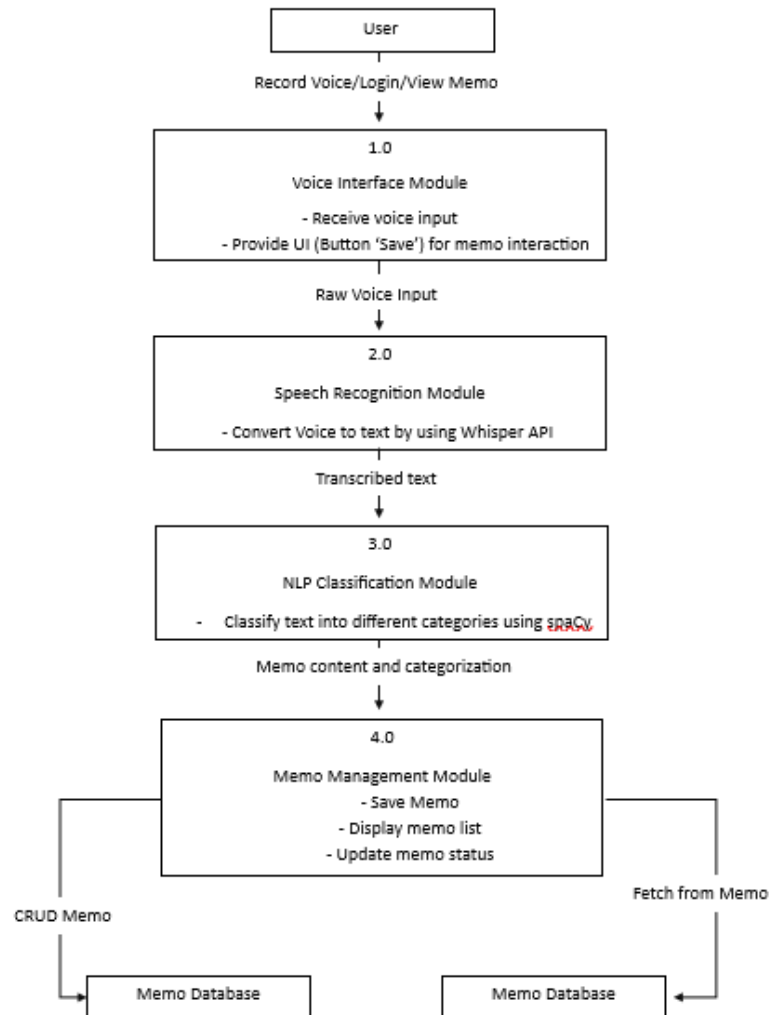


Figure 10 DFD diagram

The above diagram describes the flow of the project from the voice input to the successful storage of the memo in the database.

Firstly, after the user inputs the voice content, the Flask API sends the recorded file to the backend for analysis. Then, there is text recognition and intelligent classification by spaCy and NLP language to train and analyse the correct memo categories. In addition, for memo storage and management, I used two main databases tables (memos and user) to store all the relevant information. This information is presented graphically in terms of data flow and system architecture.

### c) UI design

UI design is the foundation of all software, it is the main functional display of user interaction with the system, helps the system to guide the user to use the part of the functions and give appropriate feedback. The perfect UI design will help users to get started easily, avoiding cumbersome or unclear buttons or interactions.

In this project, the goal of UI design is to make all users understand how to use the application. The minimalist and easy to use look and feel of the app allows for quick creation of memos and viewing of memos.

In order to improve user friendly, all buttons were designed to be colour coded and appropriately sized so that the user could quickly identify the interaction of the button. This helps to reduce the cognitive load to achieve the goal of quick function execution.

As conclusion, a suitable and aesthetically pleasing UI design not only improves usability and efficiency, but also enhances the user's dependence and liking for the system.

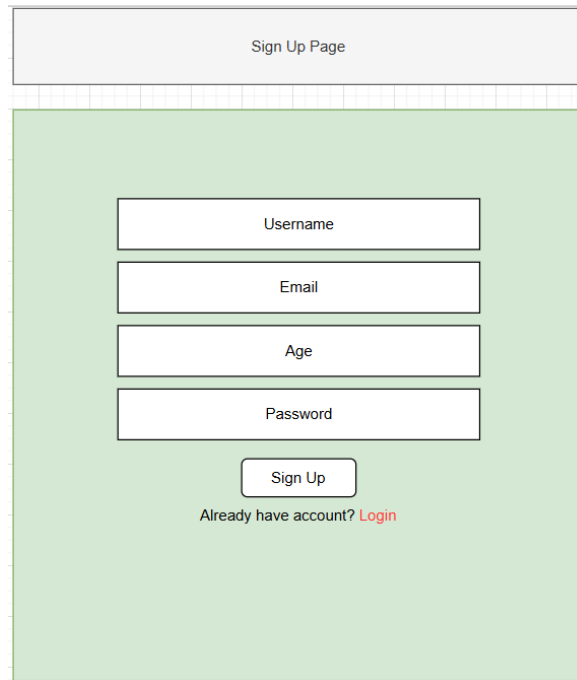
The below diagrams are the UI design before implementation in real scenario:

Figure 11 shows a wireframe of a login page. It features a light gray header bar at the top with the text "Login Page". Below this is a large light blue rectangular area representing the main content. Inside this area, there are two white rectangular input fields: the top one is labeled "Username" and the bottom one is labeled "Password". Below the password field is a blue text link that says "Forgot Password? Click Here". Further down is a white rounded rectangular button labeled "Login". At the very bottom of the blue area is the text "New user? Sign Up", where "Sign Up" is in red.

*Figure 11 Login Page*

The login page shown above including the "Username" and "Password". If username is corresponding to password, then the user can login successfully. The login page also includes the forgot password option for those who forget password can reset their password. For new user, the user can click the "Sign Up" link for go through the "Sign Up" page.

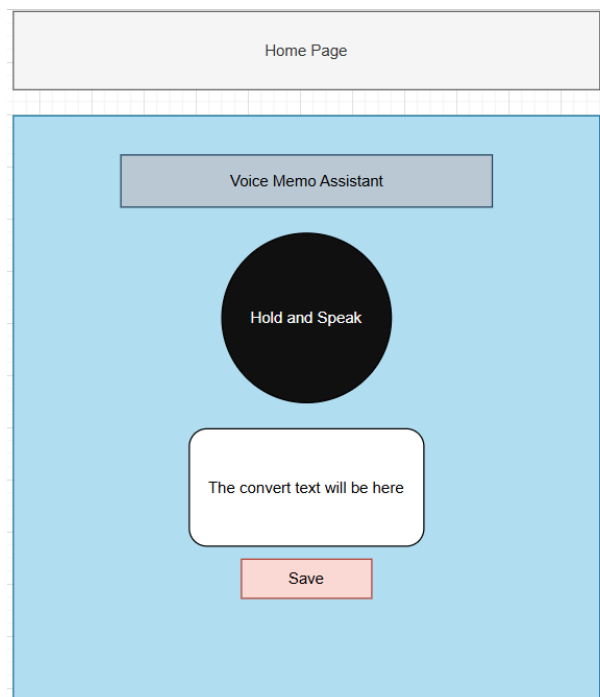




The image shows a wireframe of a 'Sign Up Page'. At the top is a light gray header bar with the text 'Sign Up Page'. Below this is a large green rectangular area containing the form. The form consists of four white rectangular input fields stacked vertically, labeled 'Username', 'Email', 'Age', and 'Password'. Below these fields is a white rounded rectangular button labeled 'Sign Up'. At the bottom of the green area, there is a link that says 'Already have account? Login', where 'Login' is in red text.

Figure 12 Sign up page

The sign up page is including the “Username”, “Email”, “Age” and “Password” as the required conditions for new user to sign up an account. After sign up successfully, user can go through the login page.



The image shows a wireframe of a 'Home Page'. At the top is a light gray header bar with the text 'Home Page'. Below this is a large light blue rectangular area. At the top of the blue area is a gray rectangular button labeled 'Voice Memo Assistant'. In the center is a large black circle with the text 'Hold and Speak' in white. Below the circle is a white rounded rectangular box containing the text 'The convert text will be here'. At the bottom of the blue area is a pink rectangular button labeled 'Save'.

Figure 13 Home page

For the home page, as the main function for the whole system. It including the recording button and the converted text will be shown below. After the user check the text is correct, user can click the save button to save the memo in database and system.

The diagram shows a 'Confirm memo page' which is a light gray rectangle at the top. Below it is a larger light blue rectangle representing the main content area. Inside this blue area, there is a gray button labeled 'Confirm your memo'. Below the button, the text 'Title: [user input]' and 'Time: [User set the time]' is displayed. Further down is a white input field labeled 'Category'. At the bottom of the blue area are two buttons: a red 'Save Memo' button and a white 'Discard Memo' button.

Figure 14 Confirm memo page (POP OUT)

After the user click the save memo button, the system will pop out the confirm page for user to verify the title and the time before save in the system. The system will analyse the text input and categorize the memo into corresponding memo category. Then, click the save memo button will be saving, or by clicking the Discard memo button will be return the home page.

The diagram shows a 'Memo List' page. At the top is a light gray header bar with a back arrow icon and the text 'Memo List'. Below the header is a table with a light blue header row. The table has two main sections: a left sidebar and a main content area. The sidebar has a 'Table' header and four rows: '1 Work', '2 Daily', '3 Study', and '4 Others'. The main content area has a tab labeled 'All' and a memo entry for 'Memo 1' with subtext 'Time Date Category' and a red 'Delete' button.

Table		All
1	Work	<input type="radio"/> Memo 1 Time Date Category <span>Delete</span>
2	Daily	
3	Study	
4	Others	

Figure 15 Memo Display page

After the confirm memo page completed, the user can see the memo details in the memo list page. User can view the different categories of memo by clicking the options such as "Work", "Daily", " Study" and "Others" that present in the left side navigation bar. Besides that, user can tick the memo as completed status and delete the memo by clicking the delete button.

As conclusion, for the UI design, I sketched the wireframes using Draw.io tool to quickly draw the layout core pages such as Login page, Sign Up page, Home page and the Memo list page.

For the above diagrams show the different pages in the project, the navigation flow is designed to be intuitive. At the beginning, user will be in the login page, if the user doesn't have account before, user can just click the sign up. After logging in, user will be directing to the home page as the voice recording page. Through the button at the middle, user can record the audio and save the memo by just click the button.

Besides that, user can click the memo page to view and manage the memo that saved before. I set for 4 choices button as the navigation at the left side to let the user click on. By clicking different categories of the memo, user can easy to view what should user do and the time will also display there.

As summary, this wireframe helps to visualize structure before go to the coding part. This helps to build a UI foundation for me before go through the coding part.

### 3.4 Development

The development phase is the core part of the project and aims to actually create the system. The features previously planned in the planning and design stages are implemented in a system through coding. By developing this project in Atom, I have adopted a modular approach to development, dividing the system's features into different sections to make it easier to manage. Then, I have downloaded many components like navigation bar through Expo Go.

Through the developing stages, I used the Android Studio app as simulator to test the system in real-time, to ensure the frontend and backend connectivity is stable. Furthermore, the system functions in this project is strictly follow the Data Flow Diagram and Context Diagram to ensure the project fulfil with the requirements.

#### a) Software Requirement

These tools include development environments, frameworks, API, database system and software used for documentation and diagramming. Each software tool plays a specific role for ensuring the successful implementation of the project's functionality and objectives.

*Table 3 Software requirements table*

Types of software	Software Purpose
Microsoft Word	Used for writing the documentation of the project.
Project Libre	To construct the Gantt Chart and manage project timeline.
Draw.io	Used to create Data Flow Diagram.
Expo Go (React Native)	Used for developing the cross-platform mobile application.
Flask (Python)	As the backend to access the voice input and memo management.
MySQL Workbench	As database to store the user data and memo data.
SpaCy	Used for categorizing memo content using a trained classification model.
Atom	Used for writing and manage coding part.
Android Studio	Used for testing the app function real-time.

The tools that listed above the table all are supporting and helping me in the full lifecycle of the application from planning stage to the maintenance stage, as the whole project development.

```

import spacy
import uuid
import requests
from pydub import AudioSegment
from flask_sqlalchemy import SQLAlchemy
import langdetect

nlp_zh = spacy.load("C:/Users/User/voice-memo-assistant/zh_text_categorizer_model")
nlp_en = spacy.load("C:/Users/User/voice-memo-assistant/en_text_categorizer_model")

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///memo.db'
db = SQLAlchemy(app)
CORS(app)

```

Figure 16 app.py file screenshot

For the diagram above, it shown the several libraries that import by using Flask as backend. As explanation, the whisper API is used to perform speech-to-text conversion from user voice input. Then, the spaCy is used to classify the text into appropriate memo category. Furthermore, the SQLALchemy is used for managing the interactions with MySQL workbench.

#### b) Hardware Requirement

In order to ensure that the system can run and develop smoothly, I need to ensure the performance of the hardware to provide a stable development environment. Based on voice memo assistant project, the hardware I need needs to support real-time speech processing, model classification, and database interaction services with the system.

Table 4 Hardware requirement table

Hardware	Specification
Laptop	Acer Predator
Operating System	64-bit Operating System, x64 based processor
Processor	13th Gen Intel(R) Core(TM) i7-13700HX 2.10 GHz
RAM	16.00 GB
Storage	1 TB
Display	16-inch screen
Connectivity	Stable internet connection

Whereas the hardware conditions listed above provide enough RAM for running and 1TB of storage, enable for downloading more applicable components and storing recording files for better subsequent testing. A stable internet connection is also an important factor in determining the ability to connect to APIs and back-end services. In addition, the 16-inch screen also facilitates multitasking for better project

development. Lastly, the i7 processor also improves the speed of development, that increasing project efficiency.

### 3.5 Testing and Evaluation

Software testing is designed to verify and validate that the application performs its functions properly and conforms to the expected operational processes. Sometimes the functionality designed in the planning phase is not suitable or does not work as expected when it is actually implemented. Therefore, the purpose of software testing is to identify errors or inappropriate design and selectively optimise and remove them in order to achieve functionality fitness of the software.

There are several steps and types of software testing to ensure that each direction and content is covered.

#### a. Unit Testing

In unit testing, the components of the software are tested individually to detect any errors and bugs in each component. Usually, these components are small pieces of code that corresponding for some functions, such as the 'delete' button deleting content from a database properly, to make sure that they are working as expected.

To illustrate, during the development of the project I used different speech content to test the accuracy of the converted text and provided different kinds of memos for spaCy to intelligently recognise. This ensures that the user can sort the memos properly when using the software.

```
Transcription result: 明天要上班
Text category probabilities: {'Work': 0.9945840239524841, 'Study': 0.00450518261641264, 'Daily': 0.0003661471710074693,
'Others': 0.000544683774933219}
Predicted category: Work, Probability: 0.9945840239524841
192.168.0.213 - - [19/May/2025 23:21:04] "POST /transcribe HTTP/1.1" 200 -
Received request: <Request 'http://192.168.0.17:5000/transcribe' [POST]>
Received file: recording.m4a
Uploaded file saved as: uploads\c164a5fa-5038-490d-ad64-5bada6b6124e.m4a
Converted to WAV: uploads\15594b81-9bbd-49bc-8201-dfbc9f7c347b.wav
Transcription result: 今天要呼吸中文
Text category probabilities: {'Work': 1.1930585060326848e-05, 'Study': 0.9999747276306152, 'Daily': 5.854849860043032e-06,
'Others': 7.496254966099514e-06}
Predicted category: Study, Probability: 0.9999747276306152
192.168.0.213 - - [19/May/2025 23:25:47] "POST /transcribe HTTP/1.1" 200 -
```

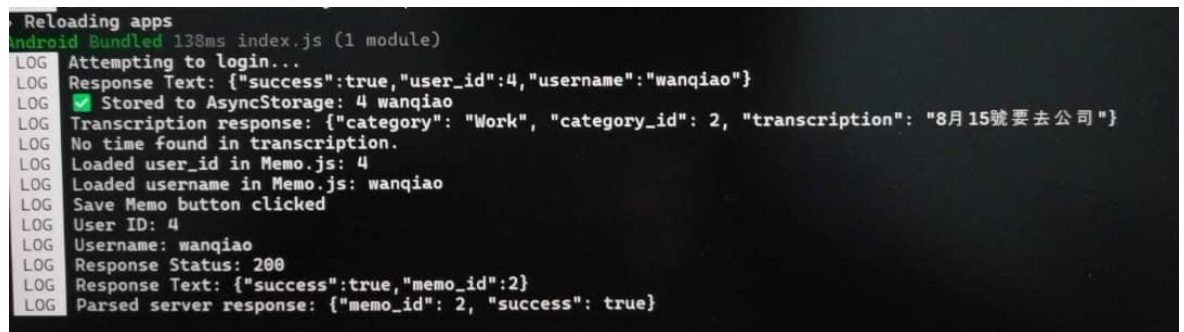
Figure 17 Unit Testing example

From the above diagram, there are several transcription results that show different categories of memo with the probability analysis. Through the probability analysis, I can clearly understand the accuracy of the spaCy for memo categorization. If the categorizations become wrong, I can add more training model to ensure the analysis is accurate. Then, those analysis results are true and successfully store in database.

## b. Integration Testing

Integration testing focuses on verifying the different functional modules in an application, such as testing whether the interaction and data exchange between these functions are working properly. The purpose of integration testing is to identify problems and errors that occur between the interaction functions of different components. To perform integration testing, I need to follow the application work flow step by step by clicking on the interaction functions and executing the flow in order to correctly verify and test that each function works properly and correctly without affecting the other functions.

As an example, to confirm that the React Native, Flask and MySQL database are working together and returning the correct feedback, I would test the app from voice input to converting text, then intelligently categorising it and storing it successfully in the database is executed correctly.

A screenshot of a mobile application's log output, likely from an Android Studio interface. The log shows the following sequence of events: 1. 'Reloading apps' and 'Android Bundled 138ms index.js (1 module)'. 2. 'Attempting to login...'. 3. 'Response Text: {"success":true,"user\_id":4,"username":"wanqiao"}'. 4. 'Stored to AsyncStorage: 4 wanqiao' (with a green checkmark icon). 5. 'Transcription response: {"category": "Work", "category\_id": 2, "transcription": "8月15號要去公司"}'. 6. 'No time found in transcription.'. 7. 'Loaded user\_id in Memo.js: 4'. 8. 'Loaded username in Memo.js: wanqiao'. 9. 'Save Memo button clicked'. 10. 'User ID: 4'. 11. 'Username: wanqiao'. 12. 'Response Status: 200'. 13. 'Response Text: {"success":true,"memo\_id":2}'. 14. 'Parsed server response: {"memo\_id": 2, "success": true}'.

```
Reloading apps
Android Bundled 138ms index.js (1 module)
LOG Attempting to login...
LOG Response Text: {"success":true,"user_id":4,"username":"wanqiao"}
LOG [✓] Stored to AsyncStorage: 4 wanqiao
LOG Transcription response: {"category": "Work", "category_id": 2, "transcription": "8月15號要去公司"}
LOG No time found in transcription.
LOG Loaded user_id in Memo.js: 4
LOG Loaded username in Memo.js: wanqiao
LOG Save Memo button clicked
LOG User ID: 4
LOG Username: wanqiao
LOG Response Status: 200
LOG Response Text: {"success":true,"memo_id":2}
LOG Parsed server response: {"memo_id": 2, "success": true}
```

Figure 18 Integration testing example

As the diagram shown above, it showing the flow from login, audio recording and text recognition, smart categorization between title and time and successfully store in database and the system.

## c. Performance Testing

Performance testing is generally used to evaluate the performance and scalability of a system. This is to test the speed of performance of the software under various constraints or conditions to ensure that the number of tasks can be processed properly.

In this project, the load on the computer is high because the use of AI models like spaCy to analyse the memo types. I tested the response time of the application and the results showed that the loading time performed on a computer with 16 RAM of operating memory would be within 10 seconds, and the best score is 3 seconds. However, since the unstable network can also affect the speed of text recognition, after making several adjustments, I make sure that the text transcription time was

limited to an acceptable range. This is to avoid the consequences of users not having the patience to wait, leading to distrust of the software.

#### d. User Testing

User testing accessed by involving different real users to test all the features of the application one by one to ensure the viability of the application. Then, collect feedback and suggestions from each user to improve or optimise the application.

To illustrate, I will also invite my friends and classmates to use the voice memo assistant application to record their daily work or tasks. Through the different types of memo content provided by different types of users, I can know and improve the accuracy of intelligent classification. For example, adding more new training text to improve the accuracy of the system's analysis. In addition, the users' real-life experience can also let me know the shortcomings of the project in order to improve and optimise it.

```
("看电影", {"cats": {"Work": 0.0, "Study": 0.0, "Daily": 1.0, "Others": 0.0}})
("和朋友吃饭", {"cats": {"Work": 0.0, "Study": 0.0, "Daily": 0.0, "Others": 1.0}})
("和朋友出门逛街", {"cats": {"Work": 0.0, "Study": 0.0, "Daily": 1.0, "Others": 0.0}})
("去工作", {"cats": {"Work": 1.0, "Study": 0.0, "Daily": 0.0, "Others": 0.0}})
("收拾行李", {"cats": {"Work": 0.0, "Study": 0.0, "Daily": 0.0, "Others": 1.0}})
("做project", {"cats": {"Work": 0.0, "Study": 1.0, "Daily": 0.0, "Others": 0.0}})
("学校小组开会", {"cats": {"Work": 0.0, "Study": 1.0, "Daily": 0.0, "Others": 0.0}})
("完成功课", {"cats": {"Work": 0.0, "Study": 1.0, "Daily": 0.0, "Others": 0.0}})
("汇报FYP进度", {"cats": {"Work": 0.0, "Study": 1.0, "Daily": 0.0, "Others": 0.0}})
("做早餐", {"cats": {"Work": 0.0, "Study": 0.0, "Daily": 1.0, "Others": 0.0}})
("做晚餐", {"cats": {"Work": 0.0, "Study": 0.0, "Daily": 1.0, "Others": 0.0}})
("做家务", {"cats": {"Work": 0.0, "Study": 0.0, "Daily": 1.0, "Others": 0.0}})
("打扫家里", {"cats": {"Work": 0.0, "Study": 0.0, "Daily": 1.0, "Others": 0.0}})
("收床单", {"cats": {"Work": 0.0, "Study": 0.0, "Daily": 1.0, "Others": 0.0}})
("整理床单", {"cats": {"Work": 0.0, "Study": 0.0, "Daily": 1.0, "Others": 0.0}})
("换床单", {"cats": {"Work": 0.0, "Study": 0.0, "Daily": 1.0, "Others": 0.0}})
("收衣服", {"cats": {"Work": 0.0, "Study": 0.0, "Daily": 1.0, "Others": 0.0}})
("吃饭", {"cats": {"Work": 0.0, "Study": 0.0, "Daily": 1.0, "Others": 0.0}}),
("冲凉", {"cats": {"Work": 0.0, "Study": 0.0, "Daily": 1.0, "Others": 0.0}}),
```

Figure 19 User testing example

Through the diagram shown above, I added more training text input to improve and enhance the accuracy of the spaCy model for analysis the accurate memo types. This helps for future when the users have different types of text that can determine the memo category in true.



### 3.6 Deployment

Once all the features have been tested and confirmed to be correct, it's time for the deployment phase, aimed at building the application in real situations and environments. This phase is designed to ensure that all components have been properly configured, integrated and are functioning properly. The goal of this phase is to deliver a fully functional system that is ready to be used by end users.

#### I. Environment Configuration

This project consists of a mobile front-end, Flask back-end and MySQL Workbench database. The front-end was developed using React Native, Atom, and the Expo Go and Android simulators were used to verify the compatibility and feasibility of the application. I also downloaded many Expo components and libraries to add more functionality to the system so that it would perform correctly. In addition, Flask is deployed locally using python 3.10, with project dependencies installed via pip. For examples, Whisper, spaCy, langdetect and others. There are a lot of contingencies when deploying the environment, such as the python version is not compatible with some of the dependencies that need to be reinstalled and removed, which also leads to the inefficiency of the project development. These experiences also helped me to identify the key issues and solve them quickly when I faced other unexpected situations. Furthermore, for database configuration I used PHP language and MySQL workbench to handle data creation, deletion, and retrieval functions. This allowed the system and users to create new users, memos, and category tables. The back-end API URLs were also managed using a proprietary configuration file, api.js, to allow for more efficient cross-platform communication.

```
const BASE_URL = "http://192.168.0.17:8000"; // 每次换 IP, 只改这里!
// 登录接口
export const login = async (username, password) => {
  try {
    const response = await fetch(`${BASE_URL}/login.php`, {
      method: "POST",
      headers: {
        "Content-Type": "application/json" // 确保是 JSON 格式
      },
      body: JSON.stringify({ username, password }) // 将数据转为 JSON 字符串
    });
    return await response.json();
  } catch (error) {
    console.error("Login error:", error);
    return { success: false, message: "Network error" };
  }
};
```

Figure 20 api.js screenshot

From the above diagram, it shown the code from api.js. It is to ensure the communication between the frontend with the backend is functioning. It used to define the base URL, allowing the application can change the different IP address based on the different servers.

## **II. Deployment Configuration**

During deployment, the PHP files such as login.php, signup.php, save\_memo.php were uploaded to the Crescendo International College's hosting environment under the htdocs/wanqiao/ directory. The MySQL database schema was exported from MySQL Workbench and imported into the college's phpMyAdmin system. The database configuration file (db.php) was updated to match the remote server settings. Through the changing of host, user, password and database name to connect with the cloud database for ensuring the system can correctly store the data in database.

To ensure seamless communication between the mobile front-end and the PHP back-end, an api.js file was used to manage API endpoints. To illustrate, the base URL was updated to point to the live deployment environment instead of the virtual environment base URL.

Finally, pilot testing was conducted to verify that all components such as front-end, back-end, and database were functioning correctly in the deployment environment. Users are able to sign up, log in, record voice memos, and save categorized memos into the cloud database successfully, representing a smooth transition from development to deployment. Besides that, it will also to be testing if the application under different network or server, it can be working and functioning properly or not. The successful execution of these workflows verified that the deployment was stable and that the system was ready for end-user interaction. This deployment configuration ensured that the system was production-ready and also laid the foundation for future maintenance and scalability, to avoid system be extended or hosted on other platforms.

### **3.7 Maintenance**

Regarding the maintenance phase, it is the last stage in every project, where the project needs to be monitored, updated and error fixed. To illustrate, maintenance needs to be done all the time in order to keep the project going in the future, specifically based on the product's lifecycle and the user's needs. In maintenance phase, developers need to make compatibility and security updates to improve user experience.

#### **a) Error Handling**

After deployment the app, there may be issues with unclear or laggy reception. These cases will be added to the Flask backend to explain and deal with the content of the error report to provide more accurate feedback. This can avoid the situation of not being able to find the key problem, enable more quickly and efficiently solve the problems faced.

In addition, in order to prevent some special memo content entry, such as: jumping three buildings, I added the 'Others' category of users to store such memos to ensure data consistency. This avoids inaccuracies during in spaCy categorization. In Layman's term, it provides flexibility in memo categorisation to improve fault tolerance.

#### **b) Compatibility Improvements**

As a part of maintenance, cross-platform and device-specific service features need to be improved. By changing the api.js configuration file to allow the application to work in different environments helps improve compatibility and stability.

#### **c) Future advance and maintenance**

Based on the modular design of this project, this feature allows future developers to modify some features without affecting other functions. This improves the maintainability and stability of the project, allows for system expansion and optimisation, and build the foundation for future updates and upgrades.

In addition, I also added some code comments for quickly finding related content code to avoid code confusion.

```
// ✅ tick as complete task function
const toggleComplete = async (id, currentStatus) => {
  try {
    const newStatus = !currentStatus;

    const updatedMemos = memos.map((memo) => {
      if (memo.memo_id === id) {
        return { ...memo, status: newStatus ? 1 : 0 };
      }
      return memo;
    });
    setMemos(updatedMemos);
  }
}
```

Figure 21 Maintenance for code comment example

As the diagram shown above, I added the comment to show that the tick function to change the memo status as completed. By viewing the comment, I can quickly understand the function for the part of coding, improving the efficiency and effectiveness.

### 3.8 Conclusion

As a conclusion, I have described in detail how the Waterfall model used in the development of this project in Chapter 3. From the initial requirement gathering and planning to the final maintenance phase, it was ensured that each phase of the project was strictly adhered to.

These processes ensured a seamless transition from the development of the application to its smooth operation, and provided a solid foundation for future maintenance and long-term operation. By adhering to each phase of the waterfall model, no functionality was left out or missing from the project development, which increased the feasibility and simplicity of the project's maintenance.

Lastly, in the next chapter, the performance of the system will be fully evaluated and the results will be discussed, including a description of how the system accomplished and achieved the desired goals.

## **CHAPTER 4**

### **RESULT AND DISCUSSION**

#### **4.1. Introduction**

In this chapter, I will look at the analysis of the project testing and presentation of the results, aim for ensuring whether the final results of the implementation for the voice memo assistant system are in line with the goals and expected results at the planning stage.

To illustrate, during the development of the voice-memo-assistant project, I implemented the core functions of voice recording, text recognition, memo saving and delete functions, memo viewing and voice control step by step, with presenting the screenshots and descriptive explanations of the results in the subsequent chapters. In addition, this chapter also discusses the various tests and debugging during the project process, by employing multiple testing methods to test each project's functionality to ensure that the project runs smoothly and as expected under different environments or conditions. Examples, unit testing, integration testing, performance evaluation and user testing. These tests verify that the project meets the desired goals and user expectations.

As a conclusion, this chapter also analyses the conclusion based on system's test results and provides an in-depth analysis of the performance, benefits and future scalability of the Voice Memo Assistant System. This chapter will mainly evaluate the functional coverage of the system, the accuracy of the classified text, and the user feedback in real scenarios. Through in-depth analyses, the functional effectiveness and application scenarios of the voice memo assistant are explored, and a foundation is made for the summary of the subsequent chapters and suggestions for future system enhancements.

#### **4.2. New Mobile Application – Voice Memo Assistant App**

Voice Memo Assistant application is a voice-driven memo application that aims to help add, manage, and view memos more efficiently. To illustrate, users can quickly save memos by using voice input in multitasking scenarios, making the whole process faster and more effective. This is suitable for those scenarios where users need to operate with both hands free, such as cooking, driving or working.

Besides that, the innovative and unique feature of this app is it integrates artificial intelligence to categorise the memos stored by the user. Voice Memo Assistant uses pre-trained NLP models to categorise the memos into four categories, namely 'Study', 'Work', "Daily", and " Others", allowing users to more efficiently see what tasks they need to do in different fields, enhancing the organisation of memos.

Furthermore, the Voice Memo Assistant system also supports English and Chinese dual language to reach out to a wider audience, it offers applicability in multilingual environments, especially in multilingual countries such as Malaysia.

The complete functional page display and the equipped page response screenshots below provide a better understanding of the core functionality of the Voice Memo Assistant application and the flow of user interaction. These interfaces have gone through a step-by-step development and optimisation process, including the flow of all the key steps of voice entry, intelligent recognition and sorting, memo query and deletion. Overall, Voice Memo Assistant leverages the power of voice and AI to provide a smarter and more efficient user experience.

## Screenshot for function page in Voice Memo Assistant

Figure 22 Sign Up Page 1

Before users start using the Voice Memo Assistant app, they will need to sign up for a new account to use all the features. The system allows users to register with username, email, password and age.

This user information is stored in a database to ensure that all fields are filled in correctly and duplicate usernames are not allowed. In addition, to ensure user privacy and data security, the passwords entered by users are not stored in plain text in the database, but are encrypted using the `password_hash()` function. This mechanism ensures that even if the database data is leaked, there will be no leakage or violation of user privacy.

	userID	username	age	email	password_hash	created_at
▶	3	Kai	20	kaiwong@gmail.com	\$2y\$12\$NRq2VqLMJIJBwgr1JCrMpeEhJTxN0I5...	2025-05-10 00:04:57
	4	wanqiao	20	wqiao@gmail.com	\$2y\$12\$IWm.xUnSEuyP9v2bmWVVhOyzWF0wn...	2025-05-10 00:09:43
	5	kahyan	20	kahyan@gmail.com	\$2y\$12\$uJFC5Tvd4vMht7FO8wAcne6./KVgAgD...	2025-06-24 23:23:31
	6	Shanie	20	shanie@gmail.com	\$2y\$12\$Oz110TGjNFQsDtiomF4fBe2pGYiQHzMS...	2025-06-24 23:24:52
•	NULL	NULL	NULL	NULL	NULL	NULL

Figure 23 User Information Database

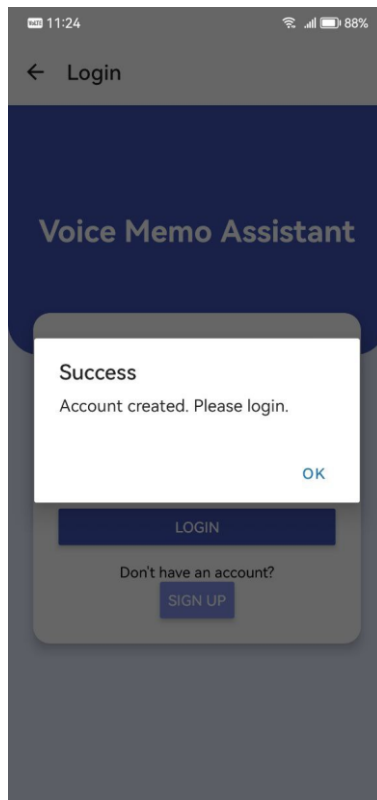


Figure 24 Sign Up Page 2

Once the user has successfully registered, the user will be passed directly to the login page so that the new user can log in directly.

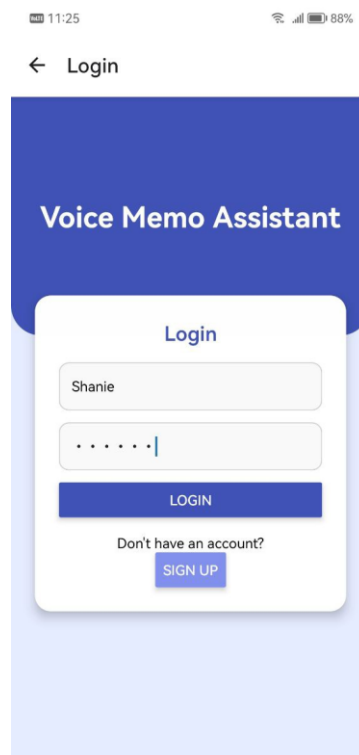
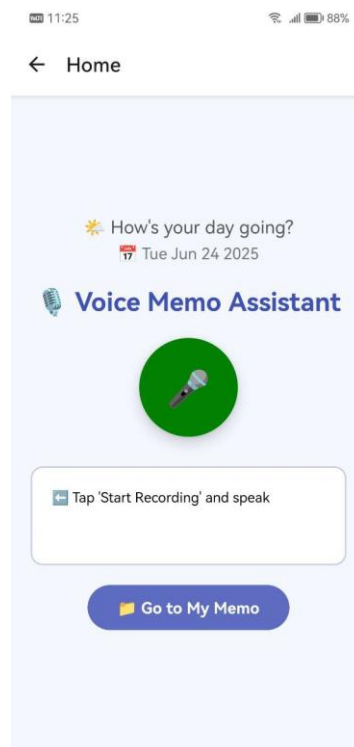


Figure 25 Login Page 1



After user successfully registers, they are automatically redirected to the Login Page to begin their session. This page serves as the gateway to access the core functionalities of the mobile application.

The user login with username and password filled, when the system determines from the database that the fields entered by the user are correct, the user is successfully logged in and redirected to the Home page.



*Figure 26 Home page 1*

After user successfully login, user will be redirected to home page. This page is the core features of the Voice Memo Assistant application, user will begin interacting with the system, including speech-to-text function, memo classification and memo management.

The header of the home page will have a welcome slogan and display the current date to provide user friendly functionality. The home page prominently displays a microphone button that allows the user to record their voice. When the green Record button is clicked, the text area below will alert that the system is recording voice and the user can say what they want, such as saving a new memo, deleting a specific memo, and retrieving a memo for a specific date. In addition, there is a 'Go to My Memo' button below the text area, when pressed it will incoming to the specified display of all memos corresponding to the logged in user.

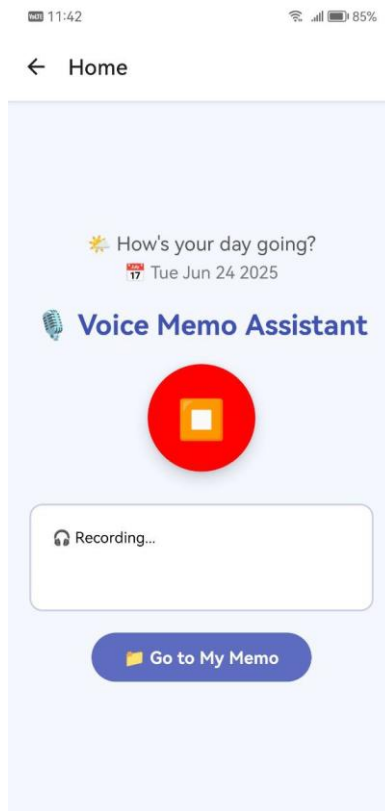


Figure 27 Home Page 2

After the user clicked the green record button, the button will changed status become red stop record button, it used to stop then save the user's voice input to conduct voice recognition and memo classification process.

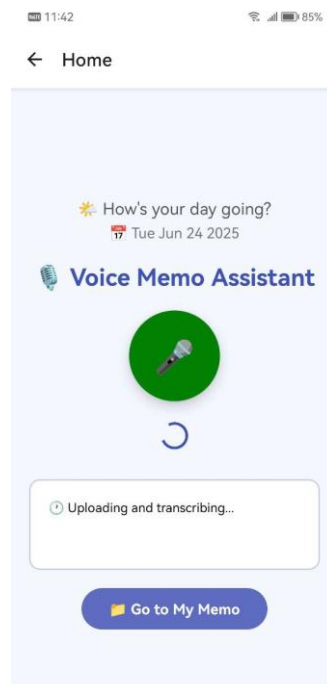


Figure 28 Home Page 3

After user stop recording, the system will start to analyse and conduct memo classification to continue the process later. The text area will show the text result for user to confirm.

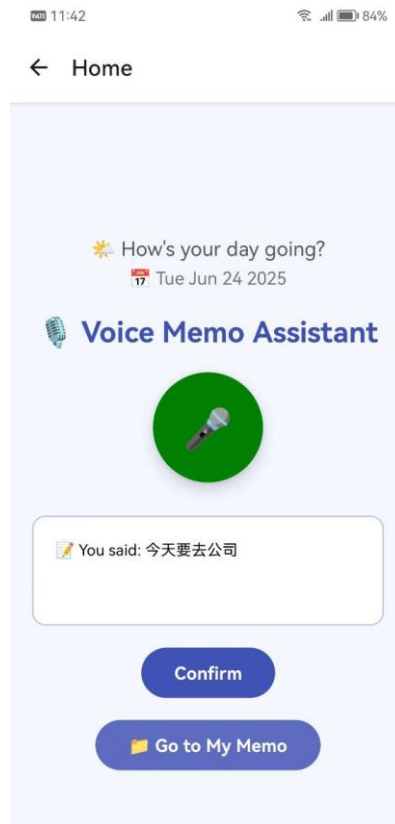


Figure 29 Home Page 4

The text area will show the text result for user to confirm when user say about memo saving function. Then, user can click the “Confirm” button to save the memo in database and system.

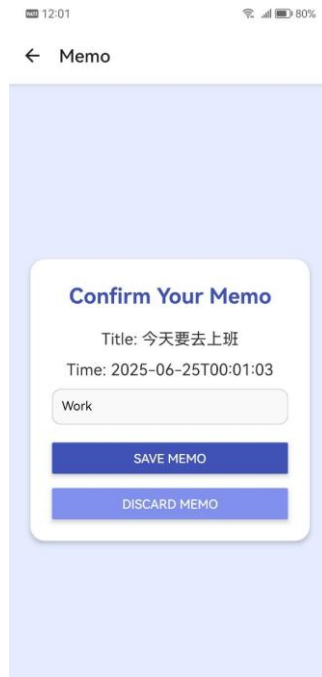


Figure 30 Confirm memo page

After user click the “Confirm” button, it will move to the confirm memo page for user to confirm all memo details to choose whether save or discard the memo. The system will show the title and the time said by the user, then show the memo classification type in the text area. If user ensure system correctly, then just click the “Save Memo” button to execute the save memo function. Otherwise, just click the “Discard Memo” button to cancel this memo and return to the home page.

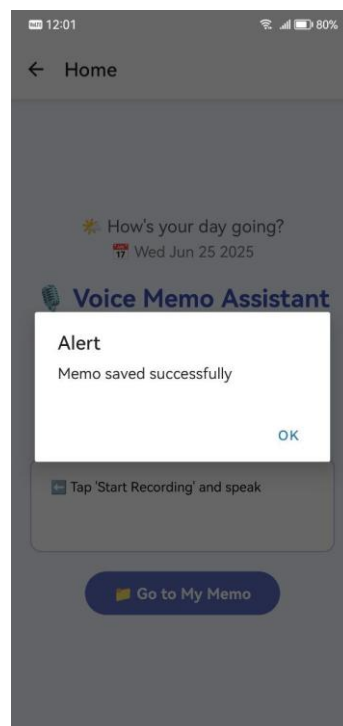


Figure 31 Memo saved Alert

After user saving the memo successfully, it will alert the memo saved successfully to user to achieve user friendly design.



Figure 32 Memo Details 1

By clicking on the 'Go to my memo' button on the home page, the user is transferred to the memo details page. In this page, unfinished memos will be displayed at the top and finished memos will be displayed at the bottom of the page. Each memo shows the title, memo type, time and date, and a red delete button on the right-hand side allows the user to delete the memo. Users can click the checkbox to mark the memo as completed and the system will automatically update its status in the database.

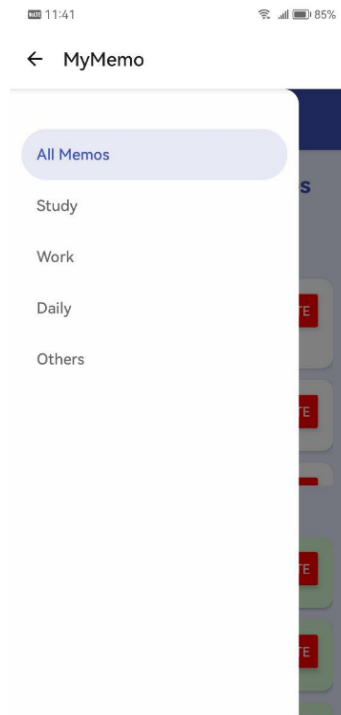


Figure 33 Memo Details Page 2

In addition, the sidebar navigation menu allows users to filter memos by category, such as all memo, study, work, daily and others. After selecting a category, the page displays only the memos belonging to that category, allowing users to focus on specific types of tasks. This is especially useful for students or professionals who need to prioritise different aspects of their daily responsibilities.



Figure 34 Daily category Page



Figure 35 Memo Details Page 3

The system sets two statuses of memo, which is completed and uncompleted. After user clicking the tick bar, the memo will automated become completed status and the background become green colour to alert this is a completed memo.



Figure 36 Memo Details Page 4

If user want to cancel the memo's completed status, users can click the tick bar here to switch the status, so that it can become an unfinished memo. This is suitable for some scenarios where the user touches it by mistake.

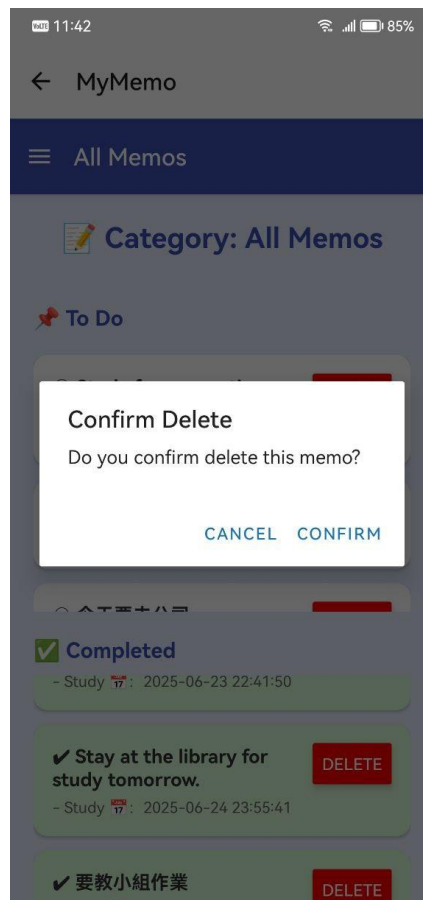


Figure 37 Confirm Delete Alert

After user clicked the delete button, it will pop out alert message to ensure user confirm to delete this memo. Then, the memo will be deleted from the system and database.





Figure 38 Confirm Delete Memo Page (Voice input)

This is the page that will pop out when user asked for delete the memo by specific title through voice input, then it will show the time range and title that user may want to delete. After the user confirmed the memo and click the “Confirm Delete” button, the memo will be removed from the system and database. It allows more user-friendly design for users can directly delete the memo they want with just using the voice input.

```
Converted to WAV: uploads\573d2dea-c619-4a30-ab71-17d2272504d9.wav
Transcription result: 帮我删除5月28日的事情
Text category probabilities: {'Work': 0.004440004471689463, 'Study': 0.00020690505334641784, 'Daily': 0.002348050009459257, 'Others': 0.012611379846930504, 'Query_Today': 0.0670710951089859, 'Query_Tomorrow': 0.0020616415422409773, 'Query_Custom': 0.08521980047225952, 'Delete_Specific': 0.013656623661518097, 'Delete_All': 0.8471465706825256}
Predicted category: Delete_All, Probability: 0.8471465706825256
Matched X月X号格式: 2026-05-28 00:00:00
Delete request | Type: Delete_All | Time: 2026-05-28 00:00:00
Time range for deletion: 2025-06-16 21:40:01.600001 to 2025-06-17 21:40:01.600001
192.168.0.135 - - [16/Jun/2025 21:40:01] "POST /transcribe HTTP/1.1" 200 -
```

Figure 39 The backend information

This is the Flask backend shown the user voice input text, it shown the user ask for delete the memo of 28 May, so the system will detect what user said and categorise it to “Delete\_All” category, and this message will send to frontend for delete this memo.

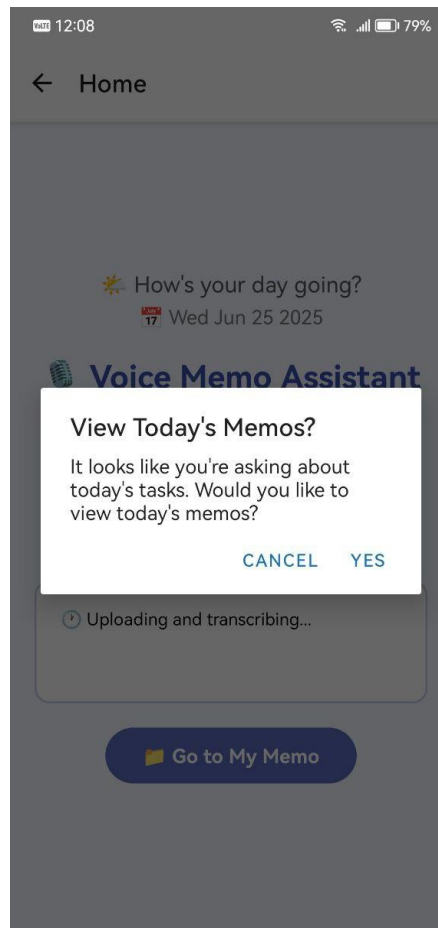


Figure 40 View Memo Page

The above diagram shown the page when user asked for what is today memo or other specific date, the system will detect the category and load to the memo details page. Then, user can just click the “Yes” button to show the memo today.

```

converted to WAV: uploads\3005cdca-0783-46ad-8835-22b7bf298ba6.wav
Transcription result: 帮我看一下今天的任务是什么
Text category probabilities: {'Work': 0.012741840444505215, 'Study': 0.0012550713727250695, 'Daily': 0.00046676487545482
314, 'Others': 0.0005735906306654215, 'Query_Today': 0.25713589787483215, 'Query_Tomorrow': 0.003524016123265028, 'Query
_Custom': 0.04156557470560074, 'Delete_Specific': 0.04325234144926071, 'Delete_All': 0.9943978786468506}
Predicted category: Delete_All, Probability: 0.9943978786468506
Time extraction result: [['今天', datetime.datetime(2025, 6, 13, 23, 48, 36, 594227)]]
Delete request | Type: Delete_All | Time: 2025-06-13 23:48:36
Time range for deletion: 2025-06-13 23:48:36.599226 to 2025-06-14 23:48:36.599226
192.168.0.134 - - [13/Jun/2025 23:48:36] "POST /transcribe HTTP/1.1" 200 -
  
```

Figure 41 Query\_Today backend information

This diagram shown the flow from user voice input that asking “What is the task for today?”, then the system detect and distribute to the “Query\_today” category to search all the memo about this date.



Figure 42 The Memo details result Page

As the result, after the user clicking the “Yes” button it will redirect to the memo details page. As you can see, there are showing the 25 June memos including the memo title, memo category, date and time. So, user can easily to view and know what should they do for that day through this function.

Through the series of interface screenshots and the functional descriptions, a workflow of the Voice Memo Assistant application is determined. From the user sign up and login process, and the voice recorded and transcribe to text, memo categorisation and memo manage function. The inclusion of features such as status marking, category filtering, and voice-driven memo deletion or querying help to enhance user experience, especially in multitasking scenarios. Each interface and logic flow has been designed to support both usability and performance, build a strong foundation for further deployment and real-world testing.

## System Evaluation

### 1. Project Requirements

The main target for this project is to develop a voice-driven memo assistant that allows users to manage their memo through voice input in multitasking scenarios. As the result shown above, the final system enable user to save, query, and delete memos through voice input, offering users a hands-free and efficient experience. Therefore, the system meets the original requirement of providing a more accessible and smart system for task management by avoiding the manual control.

### 2. Modifications during development

At the beginning, the backend logic is including the interaction with database, but based on my lecturer, Mr. Simon asked for the requirements, I changed the database interaction to PHP connection. So, it allowed a better management by using PHP to execute the database instructions. This architectural change reflects a critical development pivot that improved the system's robustness and maintainability.

### 3. Challenges faced during development

For the big challenge that I faced during the system development, is that Expo Go in mobile device update to new version that make a version mismatch with the development environment on the computer. As a result, the app failed to launch properly because all the version is conflict and incompatible, then caused the development process stagnation. To solve this issue, I need to reinstall the entire development environment which including all project dependencies and configuration files. To illustrate, this makes me wasted a lot of time to debug the python version and Expo Go version, but it also gives me a valuable experience in debugging cross-platform issues.

### 4. Initial Goals

Whether faced a lot of challenges through development process, but the Voice Memo Assistant application also achieved the initial goals in the project proposal. It enables more faster and more efficient memo saving compare to other similar system, especially suitable for users who are often needed multitasking. Furthermore, the auto memo classification function also helps to optimize the task management classification of memos without the manual control by users.

### 5. Preparation for further testing process

All main functions are implemented and tested across multiple use cases. I think the system is ready for deployment process and the real-world testing part. In the testing part, I will conduct the test including unit testing, integration testing and system testing.

### 4.3. Testing and Evaluation

System testing is the action of debugging the entire system to verify that the system works as expected. The purpose of the testing process is to reduce the risk of the project, because even if unit testing is performed, there will inevitably be problems of incompatibility or code conflict when each functional area is combined together for real scenario use. Therefore, the purpose of testing is to find potential bugs and errors and solve them completely. System testing is not only to test the functionality and design, but also to test whether the usability and smoothness of the whole system is matched to standard requirements.

Apart from unit testing and system testing, I also performed manual testing. Manual testing is an important part of software development, which relies on human beings to use the software manually to find errors and problems. In this project, I used manual testing to simulate the memo text that a real user would say when using the Voice Memo Assistant app. For example, 'I have a meeting today' would be distributed in the 'Work' category. This is to make sure that the system really converts the voice input into text and makes the correct memo categorisation.

To illustrate, I also conduct a black box testing where I entered the test text into the backend and extracted the results directly from the backend console log. In this way, I can directly view the information and probability of the categorization result returned by the system. With this backend information, I can add more test text to the specified classification categories to train the system more intelligently. So, the system will become more correct and accurate in making memo classification by using the new training model after debugging.

```
# 测试
text = "Send the invoice"
category, time = predict_and_extract_time(text)
print("Category:", category)
print("Time:", time)

# 测试
text = "delete the memo tomorrow"
category, time = predict_and_extract_time(text)
print("Category:", category)
print("Time:", time)

# 测试
text = "what should i do for 11 November"
category, time = predict_and_extract_time(text)
print("Category:", category)
print("Time:", time)
```

Figure 43 Black Box testing

Furthermore, I also put myself as the user in the role of multitasking in daily life. I simulated whether the user can just only use one hand to complete a series of processes from voice input to the successful storage of a memo, and the result is yes. This testing

method is also known as scenario testing, which means that the developer simulates the user's usage scenario. I believe that after conducting this series of tests, the Voice Memo Assistant application that I developed will be more approach with the needs of the project from the beginning.

### Problems found during testing

Although most of the bugs and system errors will be resolved in the development stage, there will still be a few slip-ups that go undetected. Here are some of the problems I found after a careful testing session and how they were resolved.

#### Problem 1:

This is the problem I found during testing part when I simulated the text the user would say in real scenario. The time input problem caused by the unclear time extraction settings in different languages. Example, the user said: 'Delete the memo of 28 May.' The system could not find the relevant memo in the database because it did not correctly get the time the user said.

The reason of this problem is that the system is unable to correctly extract the time in ISO format because it does not make a correct distinction between the incoming time functions, so the front-end is unable to correctly find the memo within the specified range, and the deletion and viewing functions are not correctly executed.

To illustrate, the date format used in the Chinese model is not accurate with the date format in English model. Examples, user may say '28 May' or 'May 28' which are two different time formats. Therefore, if the time input format of Chinese or English model is not properly set and differentiated, the system will not be able to execute the user's commands correctly.

```
Converted to WAV: uploads\573d2dea-c619-4a30-ab71-17d2272504d9.wav
Transcription result: 幫我刪除5月28日的事情
Text category probabilities: {'Work': 0.004440004471689463, 'Study': 0.00020690505334641784, 'Daily': 0.002348050009459257, 'Others': 0.012611379846930504, 'Query_Today': 0.0670710951089859, 'Query_Tomorrow': 0.0020616415422409773, 'Query_Custom': 0.08521980047225952, 'Delete_Specific': 0.013656623661518097, 'Delete_All': 0.8471465706825256}
Predicted category: Delete_All, Probability: 0.8471465706825256
Matched X月X号格式: 2026-05-28 00:00:00
Delete request | Type: Delete_All | Time: 2026-05-28 00:00:00
Time range for deletion: 2025-06-16 21:40:01.600001 to 2025-06-17 21:40:01.600001
192.168.0.135 - - [16/Jun/2025 21:40:01] "POST /transcribe HTTP/1.1" 200 -
```

Figure 44 Time Extraction Problem

From the above diagram, you can see that the text input is "Help me to delete the task for 28 May.", but the system extracted the time range is between 16 June to 17 June although the backend had been extracted the correct time.

So, I changed the return logic for the extracted time in Flask backend to make sure that the time stated by the user is returned and passed to the front end. In addition, I also added some conditional restrictions, such as automatically adding 1 year once the day and month

of the date of the user said have already passed. This action prevents the user from depositing a new memo that save it to last year, resulting in an inaccurate reminder.

```
zh_date_match = re.search(r"(\d{1,2})月(\d{1,2})[号日]?", text_cleaned)
if zh_date_match:
    print("✅ 明确中文日期:", zh_date_match.group(0))
    month = int(zh_date_match.group(1))
    day = int(zh_date_match.group(2))
    year = now.year
    parsed = datetime(year, month, day)

    # 🚫 删除类任务时不要跨年
    allow_next_year = not (
        "delete" in text.lower() or "删除" in text or "删除" in text
    )

    if parsed < now and allow_next_year:
        parsed = parsed.replace(year=year + 1)

    return parsed.strftime("%Y-%m-%d") if only_date else parsed.strftime("%Y-%m-%d %H:%M:%S")

# 检测英文日期格式 like "23 June" or "June 23"
en_date_match = re.search(r'(?:(\d{1,2})?(?:st|nd|rd|th)?\s+of\s+)?(January|February|March|April|May|June|July|A
if en_date_match:
    print("🖱️ 手动匹配英文日期:", en_date_match.group(0))
    day = int(en_date_match.group(1)) if en_date_match.group(1) else int(en_date_match.group(3))
    month_str = en_date_match.group(2)
    month = datetime.strptime(month_str[:3], "%b").month
    year = now.year
```

Figure 45 Time Extracted logic in both language model

As result, after fixing the time extraction code error, the system can correctly determine the date and time information entered by the users and successfully save into database.

## Problem 2:

The problem of inaccurate classification results also occurs often, because I trained the model myself, so inevitably there will be some keywords that are not added to it, and this kind of insufficient training text will lead to the problem of misclassification. In addition, the training text of some categories is too similar or too much can also lead to the problem of inaccurate classification results.

```
Converted to WAV: uploads\3005cdca-0783-46ad-8835-2257bf298ba6.wav
Transcription result: 帮我看一下今天的任务是什么
Text category probabilities: {'Work': 0.012741840444505215, 'Study': 0.0012550713727250695, 'Daily': 0.00046676487545482
814, 'Others': 0.0005735906306654215, 'Query_Today': 0.25713589787483215, 'Query_Tomorrow': 0.003524016123265028, 'Query
_Custom': 0.04156557470560074, 'Delete_Specific': 0.04325234144926071, 'Delete_All': 0.9943978786468506}
Predicted category: Delete_All, Probability: 0.9943978786468506
Time extraction result: [['今天', datetime.datetime(2025, 6, 13, 23, 48, 36, 594227)]]
Delete request | Type: Delete_All | Time: 2025-06-13 23:48:36
Time range for deletion: 2025-06-13 23:48:36.599226 to 2025-06-14 23:48:36.599226
192.168.0.134 - - [13/Jun/2025 23:48:36] "POST /transcribe HTTP/1.1" 200 -
```

Figure 46 Inaccurate Classification Result

From the above screenshot diagram of the backend return information, you can see that the user said: "Help me to see that what is the task for today.", but the system made a wrong categorisation. Originally it will be 'Query\_today' category, but the system predicted this is 'Delete\_All' request, so it will lead to the confirm delete page.



To solve this type of problem, I chose to add more training text to the classification of “Query” and “Delete” to help the system train and learn the differences between these two types of text. By adding a large amount of training text, the system was able to demonstrate the correct classification categories and title extraction in the later test results, helping the system meet the requirements.

```
read the cat at 7 PM", ("cats": {"Work": 0.0, "Study": 0.0, "Daily": 1.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("Tidy up the desk at 10 PM", ("cats": {"Work": 0.0, "Study": 0.0, "Daily": 1.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("Prepare lunch box for tomorrow", ("cats": {"Work": 0.0, "Study": 0.0, "Daily": 1.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("Buy the new perfume", ("cats": {"Work": 0.0, "Study": 0.0, "Daily": 1.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("Go for swimming today", ("cats": {"Work": 0.0, "Study": 0.0, "Daily": 1.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("Go grocery shopping tomorrow at 11 AM", ("cats": {"Work": 0.0, "Study": 0.0, "Daily": 1.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("Clean the fridge", ("cats": {"Work": 0.0, "Study": 0.0, "Daily": 1.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("Study the coding", ("cats": {"Work": 0.0, "Study": 1.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("Complete the assignment on 9 July", ("cats": {"Work": 0.0, "Study": 1.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("Submit the essay at 6 PM", ("cats": {"Work": 0.0, "Study": 1.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("I have to review chemistry tomorrow", ("cats": {"Work": 0.0, "Study": 1.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("I have to review the physics", ("cats": {"Work": 0.0, "Study": 1.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("Take notes on today's class", ("cats": {"Work": 0.0, "Study": 1.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("Review lecture slides on 5 January", ("cats": {"Work": 0.0, "Study": 1.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("Practice speaking for English oral exam", ("cats": {"Work": 0.0, "Study": 1.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("Must complete all the PMP coursework at 22 July", ("cats": {"Work": 0.0, "Study": 1.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("Work on the group assignment tomorrow", ("cats": {"Work": 0.0, "Study": 1.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("Have a group meeting for discuss the Digital course", ("cats": {"Work": 0.0, "Study": 1.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),

("11 November need to go to company", ("cats": {"Work": 1.0, "Study": 0.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("I need to go to work on 10 AM", ("cats": {"Work": 1.0, "Study": 0.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("What is the task for today", ("cats": {"Work": 0.0, "Study": 0.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 1.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("Help me to delete the task on 11 February", ("cats": {"Work": 0.0, "Study": 0.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 1.0})),
("Delete the task tomorrow", ("cats": {"Work": 0.0, "Study": 0.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 1.0})),
("I need to shower on 10 PM", ("cats": {"Work": 0.0, "Study": 0.0, "Daily": 1.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("Send invoice on 10 PM", ("cats": {"Work": 1.0, "Study": 0.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("What should i do for today", ("cats": {"Work": 0.0, "Study": 0.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 1.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("What should i do for 6 November", ("cats": {"Work": 0.0, "Study": 0.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 1.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),

("11 November need to go to company", ("cats": {"Work": 1.0, "Study": 0.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("I need to go to work on 10 AM", ("cats": {"Work": 1.0, "Study": 0.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("What is the task for today", ("cats": {"Work": 0.0, "Study": 0.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 1.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("Help me to delete the task on 11 February", ("cats": {"Work": 0.0, "Study": 0.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 1.0})),
("Delete the task tomorrow", ("cats": {"Work": 0.0, "Study": 0.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 1.0})),
("I need to shower on 10 PM", ("cats": {"Work": 0.0, "Study": 0.0, "Daily": 1.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("Send invoice on 10 PM", ("cats": {"Work": 1.0, "Study": 0.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("What should i do for today", ("cats": {"Work": 0.0, "Study": 0.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 1.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("What should i do for 6 November", ("cats": {"Work": 0.0, "Study": 0.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 1.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),

("11 November need to go to company", ("cats": {"Work": 1.0, "Study": 0.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("I need to go to work on 10 AM", ("cats": {"Work": 1.0, "Study": 0.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("What is the task for today", ("cats": {"Work": 0.0, "Study": 0.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 1.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
("Help me to delete the task on 11 February", ("cats": {"Work": 0.0, "Study": 0.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 1.0})),
("Delete the task tomorrow", ("cats": {"Work": 0.0, "Study": 0.0, "Daily": 0.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 1.0})),
("I need to shower on 10 PM", ("cats": {"Work": 0.0, "Study": 0.0, "Daily": 1.0, "Others": 0.0, "Query_Today": 0.0, "Query_Tomorrow": 0.0, "Query_Custom": 0.0, "Delete_Specific": 0.0, "Delete_All": 0.0})),
```

Figure 47 The Training text included

As the diagram shown above, a lot of different kinds of prediction type will be added into the training model.

```
C:\Users\User\voice-memo-assistant\whisper-api>cd C:\Users\User\voice-memo-assistant\
C:\Users\User\voice-memo-assistant>python trainzh.py
Iteration 0 Losses: {'textcat_multilabel': 1.8825588524341583}
Iteration 1 Losses: {'textcat_multilabel': 0.9672929543303326}
Iteration 2 Losses: {'textcat_multilabel': 0.2438951820076909}
Iteration 3 Losses: {'textcat_multilabel': 0.09264667494653622}
Iteration 4 Losses: {'textcat_multilabel': 0.00288749192418436}
Iteration 5 Losses: {'textcat_multilabel': 0.0009702094998829125}
Iteration 6 Losses: {'textcat_multilabel': 0.0005757329915923037}
Iteration 7 Losses: {'textcat_multilabel': 0.0004398736859911878}
Iteration 8 Losses: {'textcat_multilabel': 0.00036687664737655723}
Iteration 9 Losses: {'textcat_multilabel': 0.000317998699756572}
Iteration 0 Losses: {'textcat_multilabel': 12.797869747570587}
Iteration 1 Losses: {'textcat_multilabel': 5.177290588352179}
Iteration 2 Losses: {'textcat_multilabel': 2.4892177003679024}
```

Figure 48 Training model result

Through the multiple model trainings and text additions, it can be seen that the prediction category of the system became more accurately, and the error rate of the system is decreasing.



**Confirm Your Memo**

Title: 今天要去上班

Time: 2025-06-25T00:01:03

Work

SAVE MEMO

DISCARD MEMO

*Figure 49 Result of the memo categorization*

As a result, you can see at the above diagram, the user said: “Today need to go to work” will be extracted the correct date and time, the categorization also be correct as “Work”.

In conclusion, the testing and evaluation phase is very important for Voice Memo Assistant system to work as expected in various real-world scenarios. In this process, most of the problems will be solved by training the model, optimising the code logic and adding conditional restrictions. After successfully fixing these bugs, the system can meet the goals and functional requirements of the project and satisfy the users' experience. In my opinion, I think the Voice Memo Assistant system can start the deployment stage in order to officially release the system to the real environment.

#### **4.4. Conclusion**

As conclusion, a comprehensive overview of all testing phase and results of the Voice Memo Assistant system is presented in this chapter. Various test methods have been implemented to ensure that the system operates reliably under real-world conditions.

Throughout the testing process, I identified and resolved several key issues, such as errors in the extraction of datetime logic and inaccurate classification results in both the Chinese and English models. I successfully solved these problems by improving the Flask back-end and front-end connection code and extending the NLP training model. As a result, the Voice Memo Assistant system correctly processed the classification results and datetime extraction so that the system could operate correctly.

This phase validated the overall architectural stability and usability of the system, ensured that the system was approaching maturity phase. Therefore, in the next chapter, I will deploy the system in a real environment and summarise the results and process of the whole development, besides give the future extensions and suggestions of this system.

## **CHAPTER 5**

### **CONCLUSION AND RECOMMENDATION**

#### **5.1. Summary Project**

As the final section of this project report, I will describe the final progress and results of the project. The goal of the Voice Memo Assistant system is to enable users to interact with the system entirely by voice input. It allows the user to create and store memos, delete, or view memos by using voice input. In addition, the system also sorts each memo according to the model (English and Chinese trained models) entered by the user, and stores it in the corresponding database table. Furthermore, the user can also delete memos on a specific date depends on user input. Example, 'Delete the memo on 17 June'. Then the system will intelligently recognise "Delete" as the user purpose, '17 June' as the deletion condition object. Then the user can also provide the specified date to ask whether there are tasks that need to be completed on that day, and if there are corresponding memos found, the system will provide all the eligible memo information for the user to view.

In order to achieve all the above features, this project uses several technologies and tools for the development of Voice Memo Assistant system. To illustrate, Flask python is used as the backend of this project to handle the speech-to-text function and NLP linguistic intelligence classification, as well as to handle the user's input commands. Then, spaCy model was used as a custom training model tool to classify the content of the memo and analyse the user's intent. Examples, analysed whether the user wanted to save, view or delete the memo with the specified conditions. Other than that, PHP language was also used to process and manipulate the commands of the database and transfer the data back to the frontend. PHP was also responsible for acting as a bridge between React Native and JavaScript frontend to transfer data.

As an explanation, the current result of the project is close to a complete and fully functional system that implements all the function requirements that listed in the project proposal. It supports speech recognition of text, task classification results, NLP analysis time and database saved.

Inevitably, I will encounter some challenged or difficult problems during project development. The most critical issue during the development and testing process was to ensure the accuracy of the speech recognition and the correct analysis of the results, as this is the most important component of the Voice Memo Assistant system. Since the language model is custom-trained, I needed a diverse and numerous data to support various input modes and use cases, and to add specific restrictions to avoid errors and

bugs. To optimise this, I had to add as much test data as possible from the perspective of different types of users to ensure that both the English and Chinese training models covered and supported the user's speech input to the best of their ability and executed the commands correctly.

Another challenge was actually deploying Flask, PHP and database on live servers, and ensuring proper IP routing and connection stability. I searched the web for information and for deploying database and PHP on Plesk.

In conclusion, I think the Voice Memo Assistant system has achieved its intended goals and expected functionality, with many optimisations and improvements added to meet the user-friendly goals. The system now allows users to manage and store memos using voice input and provides real-time feedback. In my opinion, it meets the goals and expected results of a voice memo assistant that is suitable for multi-tasking environments, convenient and efficient.

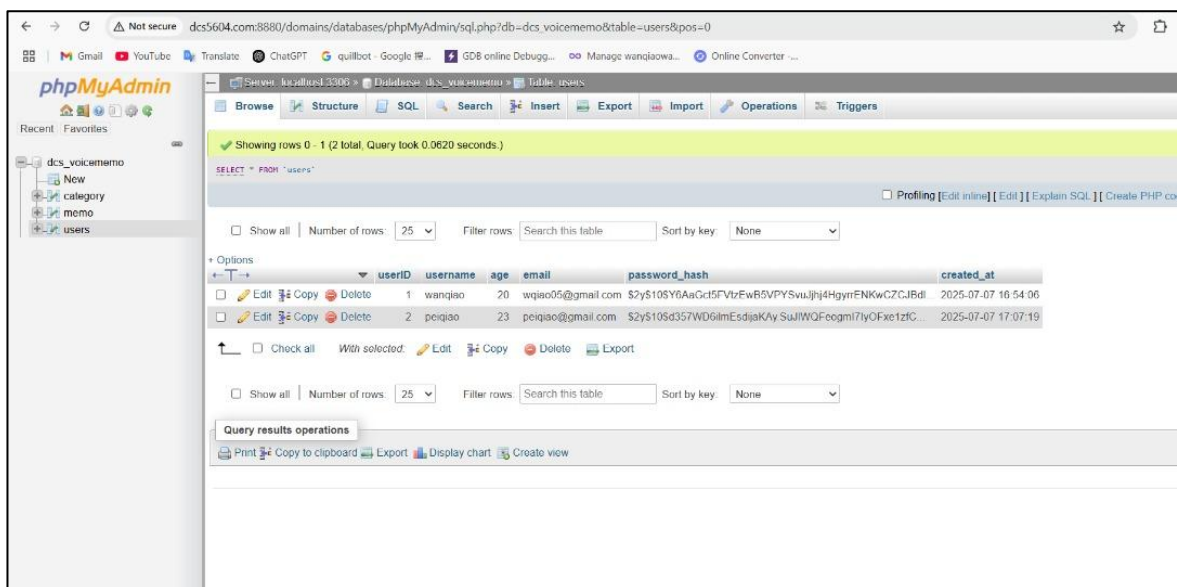


Figure 50 The connection on Plesk

## **5.2. Strength and Weakness**

In this section, I will describe the strengths and weaknesses of the Voice Memo Assistant project and explain more about the scope for future refinement or expansion.

### Strengths

The advantage of the Voice Memo Assistant system is that it allows the user to control all functions via voice input, such as adding, viewing, and deleting memos as mentioned in the above section, which helps to increase the accessibility and efficiency of the system for scenarios where the user needs to handle multiple events at the same time. Examples, when driving, going down stairs, or in a meeting scenario.

In addition, Voice Memo Assistant features enable intelligent memo categorisation. Currently, similar memo apps on the market do not have automatic sorting functionality, requiring users to manually sort tasks. Voice Memo Assistant system can identify the characteristics of user input and analyse the results through custom-trained spaCy model, and the system can automatically classify the memo into the appropriate category. Example, when the user says 'I have to go to work tomorrow at 7:00 AM', the system recognises the focus on "go to work" and analyses it into the category of 'Work' for subsequent memo storing. This helps to improve data organisation and user experience, and the one-click automation system meets the user's needs for intelligent applications.

Next, the Voice Memo Assistant system is able to detect time-related expressions from the user's voice input. Since I have included many different expressions of time alignment in the training model to try for covering different users' speaking habits, the system is currently able to recognise the time stated by the user and intelligently assigns appropriate timestamps to the memos. If the user's time data is incomplete, the system can also intelligently store the appropriate time.

Apart from that, the Voice Memo Assistant system is bilingual which included Chinese and English to meet the needs of most users. Regardless of whether the user inputs in English or Chinese, the system recognises and returns the correct analysis results, and each training text is added equally to both the English and Chinese models, allowing the system to serve a wider audience.

Other than that, the Voice Memo Assistant system was developed with a modular back-end design, each functional area is separated to improve the efficiency of subsequent modifications or maintenance. To illustrate, Python Flask is responsible for the backend analysis and identification, while PHP is responsible for the database connection. This separation of functions improves the maintainability and flexibility of the system, and this

code structure allows developers to quickly find the corresponding code, reduces the possibility of code obfuscation, and generates more efficient and simplified code.

Lastly, the Voice Memo Assistant application interface has been designed to be both formal and intuitive, making it easy for even non-technical users to get started and interact with the Voice Memo Assistant system. The red pause record button with the text alert 'System is recording' makes it easy for users to understand how to use this feature. This improves accessibility and user-friendliness for a wider range of users. Then, when the user uploads the recorded audio to the system, the system provides feedback in the shortest possible time, displaying the corresponding transcription results and functions. This real-time interaction helps to maintain a smooth user experience with the Voice Memo Assistant application.

## Weakness

Besides that, I will objectively discuss the shortcomings and weaknesses of the current Voice Memo Assistant system and how to optimise or improve it in the future.

First of all, the Voice Memo Assistant system occupies a large amount of memory due to the inclusion of two language models. The spaCy model requires large memory capacity and long processing time, so the deploy system has the problem of slow response speed on servers or devices with limited resources. And for this problem, with my current access to resources cannot completely solve this problem, I can only try to compress the memory occupied by the two models. And in the future, if there are sufficient resources or more suitable servers can also be re-established for the system connection.

In addition, since the Voice Memo Assistant system needs to rely on an internet connection to communicate with Flask and Plesk. To illustrate, user login to their accounts, storing, deleting, or viewing memos, and performing any other functions rely on the servers to perform them successfully. Therefore, this limits its usability in areas with poor or no internet access.

Lastly, I think Voice Memo Assistant system is a standalone application because it lacks integration with third-party applications or services. To illustrate, it doesn't allow users to log in with a Google or Facebook account. This rather limits the user convenience as each user needs to create a new account separately in Voice Memo Assistant application. In this way, it is not possible to associate accounts with other platforms, or share account information and store passwords with Google. Currently, memo data is only stored in the backend of the application and it is not possible to share information with other applications or platforms. This rather limits collaboration features or usability in a cross-device environment.



### **5.3. Recommendation and Future Enhancement**

In this section, I will illustrate the future expandable features and directions of this system to optimize the current functional limitations. I will list some of the new features or enhancements that can be added to the Voice Memo Assistant system in the future, with the aim of optimising the user's experience and the system's suitability.

#### **1. Connect or cooperate with third-party platforms**

Currently, the Voice Memo Assistant system can only be used for account login and registration within the app, so it is rather inconvenient for some users. This is because they need to remember the username and password and cannot login with another platform account. So in the future, I hope to achieve a third-party login system such as Google accounts or Apple ID to simplify the authentication process to improve the user experience. In addition, connecting with third-party platforms can also extend the usability of the system. To illustrate, the Voice Memo Assistant system can be logged in on multiple devices to match the user's usage on a computer, tablet or different mobile phones, and memo messages can be exported to different devices. For explanation, memo information can be added to the Google Calendar to better remind the user of the tasks that need to be completed and execute. Other than that, users can also share memo information to designated people through other platforms. For example, you can use Google account to share tasks and deadlines at work. In this way, the Voice Memo Assistant system is not only limited to personal use, and also can be applied to a wider range of applications such as companies or between friends.

#### **2. Increase the display function of user's audio**

Currently, the system only focuses on categorising the results after converting the audio to text, which is not very conducive to user accessibility and fault tolerance. In the future, I would like to add a memo voice playback function, which is added the audio content of the user's voice input at the bottom of that memo. This would enable voice playback functionality to show more information or complete prompts. This would be a much more convenient feature for users, especially if they want to retrieve information or highlights. However, this is also dependent on the memory capacity to store the audio content of a very large number of users to the server.

#### **3. Intelligent Confirmation and Conflict Handling Functions**

Now, the project I am developing cannot be designed and developed more perfectly due to time and technical issues, such as some of the system prompts and intelligent recognition have not been done to deal with the particular perfection. To illustrate, sometimes users will

store the same content into the system repeatedly. This behaviour may lead to misunderstandings and errors in the subsequent work. So, a function can be designed to identify whether similar or identical memos have been stored in the system, in order to prevent the same task from being performed repeatedly. The memos with the same category but different work content at the same time will also intelligently remind the user whether they can be done at the same time, to avoid user negligence leading to the scheduling of two things at the same time, resulting in a conflict of tasks.

#### **5.4. Conclusion**

In conclusion, the development of the Voice Memo Assistant system has successfully achieved the system functionality and goals set out in the project proposal. The system allows the user to operate the main functions of the entire system by using voice input. From the creation and classification of memo to the subsequent storage of the system, basically it can rely on voice input to complete, thus freeing the user's hands and suitable for multi-tasking scenarios. Through speech recognition, NLP language classification and intelligent time parsing, the system provides a user-friendly and comfortable experience for driving, walking or other work scenarios.

The technical approach used in this project is included the Flask backend service, spaCy model and the modular separation of the PHP language for transporting data. This planned architectural design enables smooth interaction between the front-end, back-end and database functions. Then, the user interface of the Voice Memo Assistant system was designed using the React Native framework to ensure that users can easily and conveniently use the system's features.

Throughout the development process, I encountered a lot of challenges and issues. Some of them are mentioned in the above sections such as speech recognition results not meeting expectations, language trained model limitations, server deployment issues and python version component incompatibility issues. However, through my multiple and continuous system testing and improvements, all these issues were resolved. And after many tests, I also ensure that the system functioned as expected and avoided errors or bugs in the system.

To illustrate, the project included intelligent analytics, NLP processing and mobile application development to create a productivity tool. The Voice Memo Assistant system not only provides the voice memo function, but also acts as a smart assistant to help users record the tasks they need to do in different areas, as to help them manage all the tasks they need to do more efficiently.

All in all, the current project result of Voice Memo Assistant system is a good match with my expected target function. The system now supports the use of bilingual models and can intelligently extract the time and content of the user's voice input, then analyse the results and store them in the appropriate category. Users can also use voice input to perform the function of deleting and viewing the specified memo. So, I think Voice Memo Assistant system makes a deeper progress in the field of voice-assisted productivity tools. It demonstrates the advantages of using voice input interactions to manage daily tasks and

provides users with another memo application feature besides the traditional text input way. This provides a more efficient and convenient memo management tool.

From this project I have gained deeper knowledge and experience in the application of technology, which will provide me with a more solid foundation for my own future development and application in real-world environments. I also learnt more new knowledge that cannot get from the lesson, such as converting a development project into an APK file, deploying a project to a real environment and how to develop a mobile application. This project not only helped me to learn how to really develop an application, but also allowed me to apply what I learnt from the classroom to real examples.

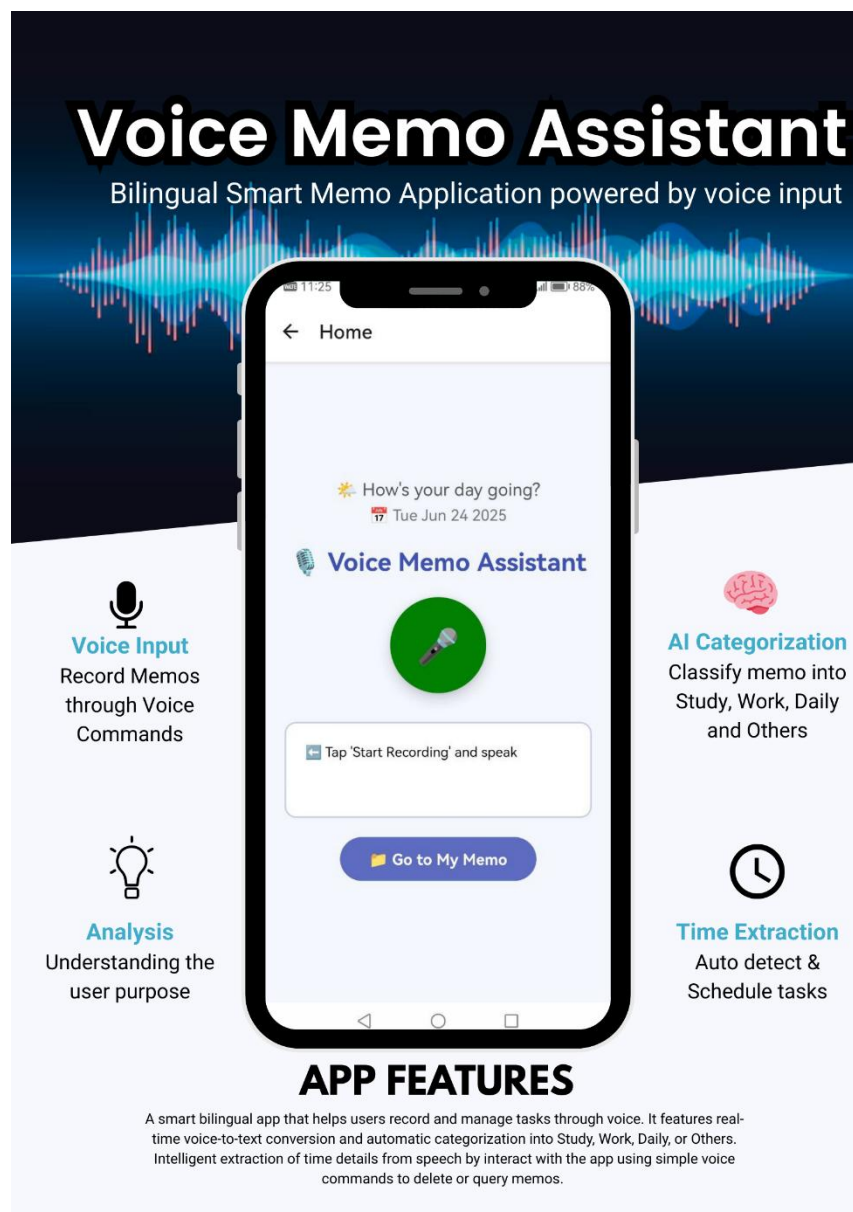


Figure 51 Voice Memo Assistant application poster

## REFERENCES

- [1] Apple Support. (n.d.). *Make a recording in Voice Memos on iPhone*. [online]  
Available at: <https://support.apple.com/guide/iphone/make-a-recording-iph4d2a39a3b/ios>.
- [2] Doctranslate.io. (2025). *Challenges Faced with Google Voice-to-Text and Their Solutions* -. [online]  
Available at: <https://blogs.doctranslate.io/2025/05/13/vi/challenges-faced-with-google-voice-to-text-and-their-solutions/> [Accessed 15 May 2025].
- [3] Flybynightgraphics. (2025). *Who are primary and secondary users? (2025)*. [online]  
Available at: [https://flybynightgraphics.com/articles/who-are-primary-and-secondary-users/1561#google\\_vignette](https://flybynightgraphics.com/articles/who-are-primary-and-secondary-users/1561#google_vignette) [Accessed 14 May 2025].
- [4] Kaushik, V. (2024). *Build a 'Speech-To-Text' React Native App with Whisper.rn*. [online] Medium.  
Available at: <https://medium.com/@vipulkaushik96/build-a-speech-to-text-react-native-app-with-whisper-rn-364439770728> [Accessed 12 May 2025].
- [5] LaPrad, L. (2024). *What is a Gantt Chart? Gantt Definitions & Uses*. [online] Teamgantt.  
Available at: <https://www.teamgantt.com/what-is-a-gantt-chart>.
- [6] PythonBasics (2021). *What is Flask Python - Python Tutorial*. [online] pythonbasics.org.  
Available at: <https://pythonbasics.org/what-is-flask-python/>.
- [7] Sentinelone (2024). *What is Systems Development? Key Concepts Explained*. [online] SentinelOne.  
Available at: <https://www.sentinelone.com/cybersecurity-101/cybersecurity/what-is-systems-development/>.
- [8] Shad, A.A. (2024). *Primary vs Secondary Users UX Design in SaaS*. [online] Thoughts about Product Adoption, User Onboarding and Good UX | Userpilot Blog.  
Available at: <https://userpilot.com/blog/primary-vs-secondary-users-ux-design/>.
- [9] Team, C. (2023). *Gantt Chart: A Complete Guide (Definition & Examples)* | Canva. [online] Canva.  
Available at: <https://www.canva.com/online-whiteboard/gantt-chart/>.

- [10] Team, I. (2025). *intellecs.ai*. [online] Intellects.ai.  
Available at: <https://www.intellecs.ai/blog/why-traditional-note-taking-is-dead-and-whats-replacing-it>.
- [11] Apple (2016). *Voice Memos*. [online] App Store.  
Available at: <https://apps.apple.com/us/app/voice-memos/id1069512134?see-all=reviews> [Accessed 7 Jun. 2025].
- [12] Gallagher, A., Dunleavy, J. and Reeves, P. (2019). The Waterfall Model: Advantages, disadvantages, and when you should use it. [online] IBM Developer.  
Available at: <https://developer.ibm.com/articles/waterfall-model-advantages-disadvantages/>.
- [13] GeeksforGeeks (2018a). Integration Testing Software Engineering. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/software-engineering-integration-testing/#what-is-integration-testing>.
- [14] GeeksforGeeks (2018b). Software Testing | Basics - GeeksforGeeks. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/software-testing-basics/>.
- [15] GeeksforGeeks (2019). Unit Testing | Software Testing. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/unit-testing-software-testing/>.
- [16] GeeksforGeeks (2020). What Is DFD(Data Flow Diagram)? [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/what-is-dfddata-flow-diagram/>.
- [17] GeeksforGeeks (2024). Waterfall Model - Software Engineering. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/waterfall-model/>.
- [18] Geeksforgeeks (2024). Context Diagrams. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/context-diagrams/>.
- [19] GeeksforGeeks. (2017). What is Software Testing? [online] Available at: <https://www.geeksforgeeks.org/software-testing-basics/#what-is-software-testing>.
- [20] <https://miro.com/>. (n.d.). Understanding the Differences Between Wireframe and UI Design. [online] Available at: <https://miro.com/wireframe/wireframe-vs-ui-design/>.
- [21] [www.linkedin.com](https://www.linkedin.com). (n.d.). How can maintenance phase improve user satisfaction in waterfall projects? [online]  
Available at: <https://www.linkedin.com/advice/0/how-can-maintenance-phase-improve-user>.

- [22] GeeksforGeeks (2018). *Software Engineering | Black box testing* - GeeksforGeeks. [online] GeeksforGeeks.  
Available at: <https://www.geeksforgeeks.org/software-engineering-black-box-testing/>.
- [23] Sharma, L. (2019). *What is System Testing? Its Objectives, Test Basics and Test Objects*. [online] TOOLSQA.  
Available at: <https://www.toolsqa.com/software-testing/istqb/system-testing/>.
- [24] TheKnowledgeAcademy (n.d.). *What is Scenario Testing in Software Testing?- A Complete Guide*. [online] www.theknowledgeacademy.com.  
Available at: <https://www.theknowledgeacademy.com/blog/scenario-testing-in-software-testing/>.
- [25] Computer science & Information technology 2024. (n.d.).  
Available at:  
<http://eprints.intimal.edu.my/1914/1/Guideline-Final%20Year%20Project%20Documentation.pdf>.
- [26] Linjanja, P. (2023). *Design Patterns for Modern Backend Development – with Example Use Cases*. [online] freeCodeCamp.org.  
Available at: <https://www.freecodecamp.org/news/design-pattern-for-modern-backend-development-and-use-cases/>.
- [27] Flask (2010). *Welcome to Flask — Flask Documentation (3.0.x)*. [online] Palletsprojects.com. Available at: <https://flask.palletsprojects.com/en/stable/>.