國立臺北商業大學

資 訊 管 理 系

113資訊系統專案設計

**系統手冊**

**組 別：第113402組**

**題 目：租中自有黃金屋**

**指導老師：蒯思齊老師**

**組 長：11046008 郭育廷**

**組 員：11046016 廖芸珮 11046028 蘇櫟**

**11046030 黃薪橙 11046034 陳玉函**

**中華民國113年01月03日**

目錄

[第1章 前言 4](#_Toc161927504)

[1-1 背景介紹 4](#_Toc161927505)

[1-2 動機 4](#_Toc161927506)

[1-3 系統目的與目標 4](#_Toc161927507)

[1-4 預期成果 4](#_Toc161927508)

[第2章 營運計畫 4](#_Toc161927509)

[2-1 可行性分析 4](#_Toc161927510)

[2-2 商業模式－Business model 4](#_Toc161927511)

[2-3 市場分析－STP 4](#_Toc161927512)

[2-4 競爭力分析SWOT-TOWS 4](#_Toc161927513)

[第3章 系統規格 4](#_Toc161927514)

[3-1 系統架構 4](#_Toc161927515)

[3-2 系統軟、硬體需求與技術平台 4](#_Toc161927516)

[3-3 使用標準與工具 4](#_Toc161927517)

[第4章 專案時程與組織分工 4](#_Toc161927518)

[4-1 專案時程 4](#_Toc161927519)

[4-2 專案組織與分工 4](#_Toc161927520)

[第5章 需求模型 4](#_Toc161927521)

[5-1 使用者需求 4](#_Toc161927522)

[5-2 使用個案圖 4](#_Toc161927523)

[5-3 使用個案描述 4](#_Toc161927524)

[5-4 分析類別圖 4](#_Toc161927525)

[第6章 設計模型 4](#_Toc161927526)

[6-1 循序圖(Sequential diagram)或通訊圖(Communication diagram) 4](#_Toc161927527)

[6-2 設計類別圖 4](#_Toc161927528)

[第7章 實作模型 5](#_Toc161927529)

[7-1 佈署圖(Deployment diagram) 5](#_Toc161927530)

[7-2 套件圖(Package diagram) 5](#_Toc161927531)

[7-3 元件圖(Component diagram) 5](#_Toc161927532)

[7-4 狀態機(State machine) 5](#_Toc161927533)

[第8章 資料庫設計 5](#_Toc161927534)

[8-1 資料庫關聯圖 5](#_Toc161927535)

[8-2 表格及其Meta data 5](#_Toc161927536)

[第9章 程式 5](#_Toc161927537)

[9-1 元件清單及其規格描述 5](#_Toc161927538)

[9-2 其他附屬之各種元件 5](#_Toc161927539)

[第10章 測試模型 5](#_Toc161927540)

[10-1 測試計畫 5](#_Toc161927541)

[10-2 測試個案與測試結果資料 5](#_Toc161927542)

[第11章 操作手冊 5](#_Toc161927543)

[11-1 系統元件 5](#_Toc161927544)

[11-2 系統下載及安裝 5](#_Toc161927545)

[第12章 使用手冊 5](#_Toc161927546)

[12-1 使用者操作說明 5](#_Toc161927547)

[第13章 感想 6](#_Toc161927548)

[第14章 參考資料 6](#_Toc161927549)

[附錄一 會議記錄 6](#_Toc161927550)

[附錄二 評審問題回覆 6](#_Toc161927551)

**圖目錄**

[圖 2-1‑1 112年度大專院校人數租屋比例 9](file:///C:\Users\user\OneDrive\桌面\專題QQ\系統手冊%20更新版.docx#_Toc165551405)

[圖 3-1‑1系統架構圖 12](#_Toc165551406)

[圖 3-2‑1 2023年台灣行動作業系統占比 13](file:///C:\Users\user\OneDrive\桌面\專題QQ\系統手冊%20更新版.docx#_Toc165551407)

[圖 3-2‑2歷年來全球作業系統占比 13](file:///C:\Users\user\OneDrive\桌面\專題QQ\系統手冊%20更新版.docx#_Toc165551408)

**表目錄**

[表 1-2‑1各租屋平台之優缺點 7](#_Toc165551505)

[表 2-2‑1 商業模式-Business model 10](#_Toc165551507)

[表 3-2‑1系統軟硬體需求-手機 14](#_Toc165551511)

[表 3-3‑1使用標準與工具表 15](#_Toc165551512)

[表 4-1‑1專案時程表 16](#_Toc165551513)

[表 4-2‑1專業組織與分工表 18](#_Toc165551514)

[表 4-2‑2專題成果工作內容與貢獻度表 19](#_Toc165551515)

# 前言

## 背景介紹

在這個快速變遷的社會中，租屋問題一直是許多學子關心的議題之一。隨著學生流動比率的增加，學生們尋找優質且適合的租屋處變得越來越具有挑戰性。

因此，我們以學生租屋的角度出發，旨在整合網路上的租屋平台及各種來源，為北漂學子提供更便捷、更優質的租屋資訊。我們將探討現有租屋平台的優缺點，分析學生租屋需求的特點，並提出相應的解決方案，讓同學們能夠更輕鬆地找到符合自身需求的理想租屋處，為北漂學子的生活品質帶來更多可能性。

## 動機

傳統的租屋方式往往是透過房屋仲介或是口耳相傳，但這些方式存在著信息不對稱、仲介費用高昂等問題，使得租屋過程充滿了不確定性和風險。本組所有成員皆是離開家鄉到台北求學的學生，所以我們深知找房子的迫切需求和困難之處。每到租房季，總是要提前好幾個月尋找合適的租屋處，需要花費大量的時間及心力。

目前市面上存在著許多租屋平台，每個平台都有自己的特點和優勢，但也同時存在著一些缺點和限制。例如：有些平台資訊不夠詳細，讓我們無法全面了解房屋情況；有些平台介面複雜，使用起來不夠直觀；還有些平台不會篩選刊登者，讓我們難以信任其中提供的資訊。

如何在眾多的選擇中找到最適合自己的租屋處儼然成為了新的挑戰，所以我們希望通過整合多個租屋平台的資訊和資源，提供一個更全面、更可靠的租屋平台，盡可能消除各平台之缺點，讓北漂學子能夠更輕鬆地找到符合需求的房子。透過整合，我們也可以更有效地管理和比較不同房源的價格、設施、地理位置等資訊，幫助使用者做出更明智的租屋決策。

表 1-2‑1各租屋平台之優缺點

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **591**  **房屋交易** | **好房網** | **樂屋網** | **租租通** |
| 是否有APP | V | V | V |  |
| 篩選功能完善 | V | V |  |  |
| 重複刊登 | V |  |  | V |
| 過濾房源 |  | V |  |  |
| 特色房源 |  |  |  | V |
| 房仲騷擾 | V |  | V |  |
| 房源數量 | 最多 | 次多 | 少 | 最少 |

## 系統目的與目標

本次專題目的在於提供一個更完善的租屋平台，從學生的角度出發，滿足租客對於租屋需求的多樣性和個性化。透過提供更全面的篩選條件，簡化租屋搜尋的過程，讓使用者能夠更快速、更有效地找到符合其條件和偏好的租屋選項。以滿足不同租客的需求。同時，我們將設計更直覺、易用的使用界面，減少使用者在搜索過程中的誤解和困惑，從而提高整體的使用者體驗。

## 預期成果

我們預期系統可以過濾重複刊登物件、比對是否為凶宅等功能，從而提高租客對於房源資訊的真實性和可信度。減少篩選器的限制，使篩選功能更完善，貼近租客需求。同時透過人工智能技術，結合LINE聊天機器人與ChatGPT，提供租客一個智能化的客服，通過對話方式幫助租客解決問題、提供建議，從而幫助他們做出更明智的租屋決策。

# 營運計畫

## 可行性分析

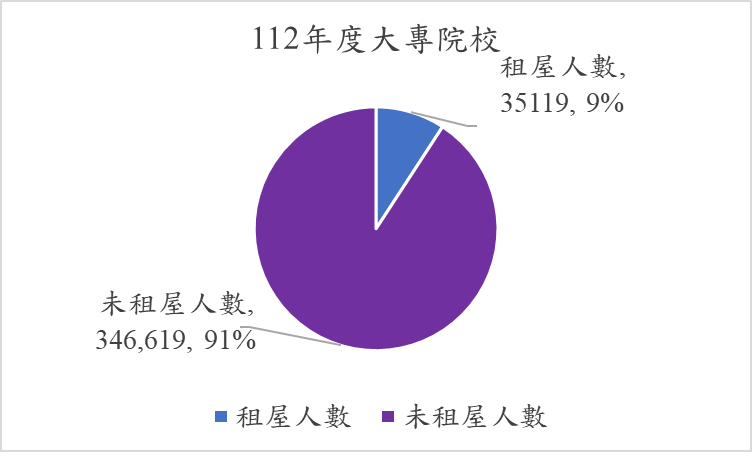
1. 時程可行性：因受學校專題課程時限，故本系統無法將系統功能開發到非常完善，於是先篩選出鄰近本校的幾個地區(中正區、大安區、萬華區)作為本次專題的目標。
2. 技術可行性：目前環境已存在的技術已足夠支援本系統之開發，例如：使用爬蟲將房屋的相關資訊存到資料庫、LINE Message API 開放使用者回答問題並且調用資料庫資料向使用者提供房屋推薦、使用Ckiptagger與Bert進行分詞與相似度分析、Microsoft Azure提供雲端資平台服務、利用Flutter框架以及Dart程式語言撰寫使用者介面。
3. 市場可行性：傳統的租屋方式如房屋仲介和口耳相傳存在諸多不便，如信息不對稱和高額的仲介費，這使得租屋過程變得不確定且風險較高。這些問題在台北尤為突出，特別是對於從其他地方來到台北求學的學生群體。所以我們特別根據在台北的大學生以及租房人數來評估市場可行性。

圖 2-1‑1 112年度大專院校人數租屋比例

根據網路上的資料，大家最喜愛用的平台排名以及數據

(表格)->收益可行性，租屋普及化，台北人口流動性高，房價貴，大部分人都是租屋

## 商業模式－Business model

表 2-2‑1 商業模式-Business model

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **關鍵合作夥伴**   * 房東 * 房仲 * LINE Message API | **關鍵活動**   * 平台維護 * 房源資訊更新 | **價值主張**   * 租屋知識 * 直觀簡潔的使用者介面 * LINE智能助手 * 節省尋找房屋的時間和精力 | | **顧客關係**   * 會員制 * 訂閱推送 * 客服服務 | **目標客戶**   * 有租屋需求之房客 * 學生 |
| **關鍵資源**   * 租屋房源管理 | **通路**   * 線上APP |
| **成本結構**   * 平台開發及維護費用 * 行銷費用 * 客服運營成本 | | | **收益流**   * 刊登收入 * 成交手續收入 | | |

根據表2-2-1 商業模式-Business model，本組的目標為透過有租房需求的租客以及學生，以訂閱推送、LINE智能助手為主要服務，吸引更多想租屋或者是租不到好房的租客來使用，提供更直觀簡潔的使用者介面，以節省尋找房屋的時間和精力，而本系統則以房東刊登費用和房屋成交手續費為主要來源收入。

## 市場分析－STP

Segmentation市場區隔

* 依照**地區**區分:台北市大安區/台北市中正區/台北市萬華區
* 依照**人生階段**區分:大專院校在學生/上班族
* 依照**租屋經驗**區分:有/無

Targeting 適合的目標市場

* 在台北無購置房屋之族群:本組將目標市場訂為北漂大專院校學生以及尚未有購置房屋能力之上班族群。
* 有出租需求之出租人:提供給房東一個平台，使他們的房員能夠更快出租。
* 喜歡簡潔的使用者介面:我們發現許多平台都存在著使用者介面不直觀的情形，所以設計更直觀簡潔的畫面，讓使用者節省時間及精力，能夠更快找到喜歡的物件。

Positioning 定位

* App內建整合好的租屋知識提供給使用者，不需要自行到網路找大量資料。
* 結合LINE智能助手，提供一個24小時在線的智能服務。

## 競爭力分析SWOT-TOWS

優勢(Strengths)：

* 新增Line智能助手，改善用戶支持與互動。
* 提供訂閱推送功能。
* 較直觀的使用者介面。
* 針對特別族群提供更完善的篩選器。

劣勢(Weakness)：

* 由於開發資源限制，目前涵蓋的房源有限，僅大安區、中正區及萬華區。

機會(Opportunity)：

* 隨著租屋需求的增加，特別是位於都市區學生和年輕工作族群中，有機會擴大用戶基礎。

威脅(Threats)：

* 市場上已存在多個成熟的租屋平台，如591和好房網，需要在這些競爭對手中脫穎而出。
* 房地產市場的波動可能影響租屋市場的穩定性，進而影響平台的表現。

表 2-4‑1競爭力分析SWOT-TOWS

|  |  |  |
| --- | --- | --- |
| 內部 外部 | 優勢(Strengths) | 劣勢(Weakness) |
| 機會(Opportunity) |  |  |
| 威脅(Threats) |  |  |

# 系統規格

## 系統架構

大部分的租屋平台都有著不同的優缺點，而我們的系統旨在整合多個平台的優點和資訊加以強化，以提供一個全面而可靠的租屋服務。透過爬蟲技術從不同平台擷取資訊，並使用數據分析技術來評估房源。系統架構主要分為三個部分：

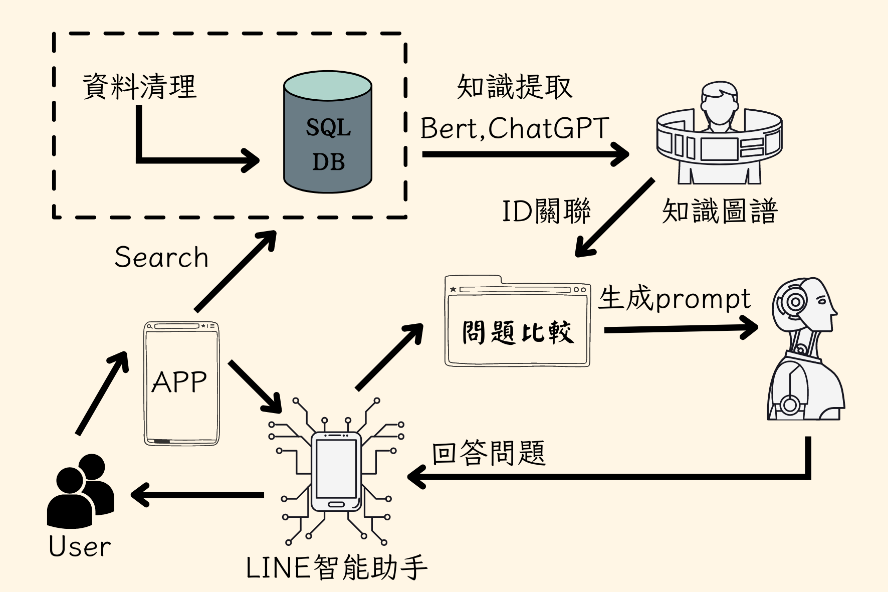
1. 前端介面：使用Flutter框架和Dart程式開發，提供用戶簡潔直觀的操作界面，讓用戶可以方便地瀏覽房源並進行篩選，節省時間以及精力。
2. 後端服務：以Node.js搭配Express框架建立，處理資料請求和系統邏輯。運用MySQL資料庫作為數據庫管理房源資料。
3. 數據處理與分析：利用Python進行數據清洗和分析，Ckiptagger和Bert用於處理自然語言，增強房源描述的分析準確性。

圖 3-1‑1系統架構圖

為了豐富此APP的租屋資訊來源，我們設計定時器，每半年運用爬蟲獲取租屋平台的最新的房源資料，並將其存入資料庫。此外，系統利用自然語言處理(NLP)技術分析和比較房屋資訊的相似度，確保資料的準確性與相關性。使用者登入APP後，可以輕鬆管理和查看自己曾瀏覽過的房源。更進一步地，透過訂閱功能，系統能智能地將合適的房源推送給租客，提高房源的曝光率並加速租屋過程。這不僅為租客提供了個性化的房源推薦，也為房東帶來了更高的租屋效率。

系統流程圖

## 系統軟、硬體需求與技術平台

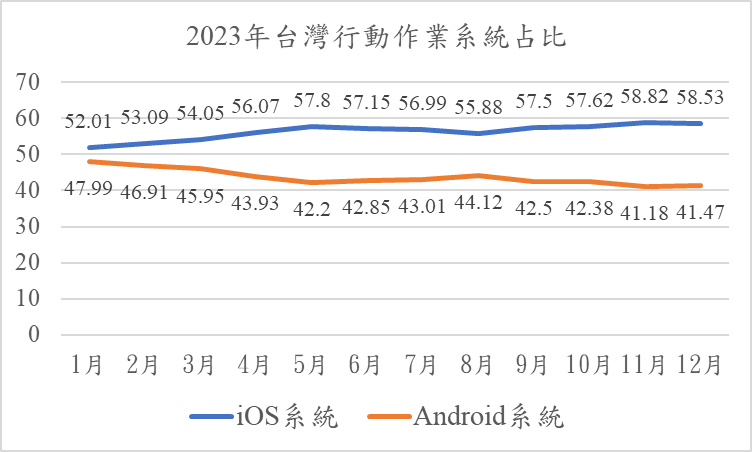
根據StatCounter網站流量分析工具統計，在[2023年台灣行動作業系統占比圖]中ios系統的比例高於Android系統，但礙於開發系統的限制，我們了解到ios系統只能於macOS 作業系統環境下開發，本組成員皆是使用Windows作業系統，因此我們選擇開發Android系統。

圖 3-2‑1 2023年台灣行動作業系統占比

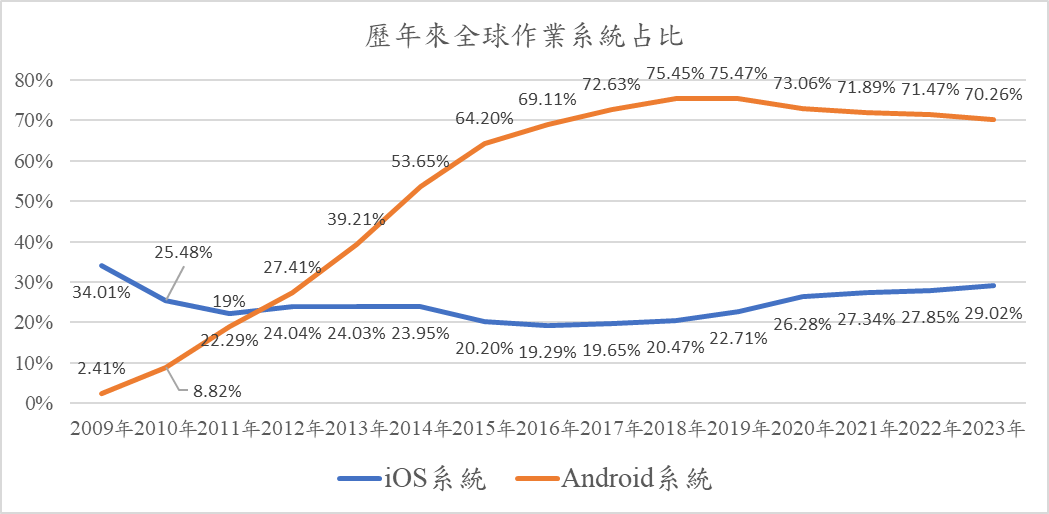
但在[歷年來全球作業系統占比圖]的統計數據來說，起初，iOS系統的市場佔有率高於Android系統，但隨著時間的推移，Android系統的增長逐年上升，並在2012年首次超越iOS系統。此後，Android系統持續擴大其市場份額，穩定在一個較高的水平。這一趨勢顯示了Android系統的廣泛接受度和使用率，其開放性和多樣化的硬體選擇使其能夠吸引全球更多的用戶，也揭示了消費者偏好和技術發展如何影響智慧手機操作系統的流行度，故本組選擇Android系統作為主要開發系統。

圖 3-2‑2歷年來全球作業系統占比

考慮到市場上Android系統的普及和版本更新的趨勢，我們決定將手機的開發版本定為Android 8.0以上。此外，為了滿足現代通訊需求和提高用戶連接性，我們的設備將支持具備Wi-Fi、4G以及5G網路的用戶。這樣的配置將使我們的系統能夠適應快速變化的科技環境，同時滿足廣大用戶對高速網絡連接的需求。

表 3-2‑1系統軟硬體需求-手機

|  |  |
| --- | --- |
| 系統軟硬體需求 -手機 | |
| 手機版本 | Android 8.0以上 |
| 網路需求 | Wi-Fi/4G/5G |

## 使用標準與工具

以下列出了本組所選用的開發工具，並說明了每項工具的特點與選擇理由：

* Python：擁有簡潔且易於閱讀的語法，配備豐富的機器學習套件，特別適合進行高效能的爬蟲開發。我們選擇Python來構建爬蟲，以獲取並處理房源資料。
* My SQL：MySQL是一種免費的資料庫管理系統，用來儲存大量數據，我們使用此系統來儲存跟會員有關的相關資料以及房源資料。
* Flutter：為一個跨平台框架，利用Dart程式語言撰寫，能同時讓iOS與Android系統使用，也支援Web以及應用程式，為本組製作android APP的工具。
* Android Studio：提供免費的整合式開發環境(IDE)，專門為Android平台的應用開發而設計。我們使用內建提供的模擬器，讓我們能在多種Android設備和配置上測試應用程式，而不需要實際擁有這些設備。
* Microsoft Azure：作為微軟的公有雲端服務平台，Azure支持多種雲端運算服務，包括基礎設施即服務(IaaS)、平台即服務(PaaS)甚至是軟體及服務(SaaS)。它允許我們在雲端輕鬆建置虛擬機和資料庫，並且相對於最多人使用的AWS具有更低的建置成本。因此，我們使用Microsoft Azure來進行資料庫以及虛擬機的架設。
* Anaconda：是目前很普及的 Python 編輯器，只要安裝後就不用煩惱各種 Python 安裝以及編輯器的問題，且 Anaconda 豐富的資料套件，能夠輔助各種資料邏輯運算，是一套很適合開發者的 Python 編輯軟體。我們使用Anaconda來解決版本問題。
* GitHub：作為軟體開發的合作平台，GitHub提供了團隊協作功能。透過Fork，我們能夠有效管理團隊成員的進度和動態，同時也便於檔案的審查和記錄，大大提升團隊的工作效率，為本組的專案進度管理工具。

表 3-3‑1使用標準與工具表

|  |  |
| --- | --- |
| 系統開發環境 | |
| 作業系統 | Windows10 |
| 開發平台 | Python、Flutter |
| 程式開發平台 | |
| 前端 | Flutter |
| 後端 | Python、Microsoft Azure、Anaconda、My SQL |
| 文件美工工具 | |
| 文件 | Microsoft Word |
| 簡報 | Microsoft PowerPoint、Canva |
| 圖檔 | Visual Paradigm、Adobe Illustrator、Procreate |
| 專案管理平台 | |
| 專案管理 | GitHub |
| 檔案存放 | GitHub、 Google Drive |

# 專案時程與組織分工

## 專案時程

根據專案工作流程，我們把專案分成14個項目，將專案彙整成甘特圖如下：

表 4-1‑1專案時程表

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 113年 | | | | | | | | | | |
| 項目 月份 | 1月 | 2月 | 3月 | 4月 | 5月 | 6月 | 7月 | 8月 | 9月 | 10月 |
| Machine Learning 探討與實作 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 訂定主題 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 爬蟲探討與實作 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 文字資料預處理 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 文字相似度對比 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| UI/UX設計 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 頁面框架編寫 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 前端開發 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 前後端串接 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 資料庫建置 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 伺服器架設 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 文件製作 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 簡報製作 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 影片製作 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

　▅ 預期進度 / ▅ 實際進度

自今年1月起，我們開始鑽研與Machine Learning相關之技術，在每週的會議中會有成員對其負責章節的詳細報告。到了3月，我們轉向學習Python爬蟲技術，同時在學習Machine Learning的過程中，我們也邊尋找有興趣的主題來發想題目，用於後續的系統開發。確定主題後，我們也開始進行App的外觀設計與初步開發工作。

## 專案組織與分工

表 4-2‑1專業組織與分工表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 項目 組員 | | 11046008  郭育廷 | 11036016  廖芸珮 | 11046028  黃薪橙 | 11046030  蘇櫟 | 11046034  陳玉函 |
| 後端開發 | 爬蟲-591資料 |  |  | ● |  |  |
| 資料庫建置- MySQL | ● |  |  |  |  |
| 資料分析-Ckiptagger |  |  |  | ● |  |
| 資料分析- Bert |  |  |  | ● |  |
| 伺服器建置 | ● |  |  |  |  |
| Line API |  |  | ● |  |  |
| Chat GPT API |  |  | ● |  |  |
| 前端開發 | UI/UX設計 |  | ● |  |  |  |
| App介面設計 |  | ● |  |  |  |
| Flutter頁面框架編寫 |  |  |  |  | ● |
| Dart程式開發 |  |  |  |  | ● |
| 美術設計 | 色彩設計 |  | ● |  |  |  |
| Logo設計 | ● |  |  | ● |  |
| 文件撰寫 | 統整 |  | ● |  |  |  |
| 第1章 前言 |  | ● |  |  |  |
| 第2章 營運計畫 |  | ● |  |  |  |
| 第3章 系統規格 |  | ● |  |  |  |
| 第4章 專題時程與組織分工 |  | ● |  |  |  |
| 第5章 需求模型 |  | ● |  | ● |  |
| 第6章 設計模型 | ● |  |  |  | ● |
| 第7章 實作模型 |  |  | ● | ● |  |
| 第8章 資料庫設計 | ● |  | ● |  |  |
| 第9章 程式 |  |  |  |  |  |
| 第10章 測試模型 |  |  |  |  |  |
| 第11章 操作手冊 |  |  |  |  |  |
| 第12章 使用手冊 |  |  |  |  |  |
| 報告 | 簡報製作 |  | ● |  |  |  |
| 影片製作 |  |  |  |  |  |

● 主要負責人 / ○ 次要負責人

在團隊討論完共同感興趣的主題後，我們將專題劃分為五大核心部分：後端開發、前端開發、美術設計、文件撰寫以及報告。為了更有效地進行專案管理，這五大項目進一步細分成多個小項目。我們採取專業分工的策略，根據每位組員的專長和強項來分配相應的工作，以確保每個階段的工作都能高效且專業地完成。不僅提高了工作效率，也讓本組成員能在其擅長的領域內發揮最大的潛力。

* 以下為Github各組員 commit的次數：

# 需求模型

## 使用者需求

## 使用個案圖

## 使用個案描述

## 分析類別圖

# 設計模型

## 循序圖(Sequential diagram)或通訊圖(Communication diagram)

## 設計類別圖

# 實作模型

## 佈署圖(Deployment diagram)

## 套件圖(Package diagram)

## 元件圖(Component diagram)

## 狀態機(State machine)

# 資料庫設計

## 資料庫關聯圖

## 表格及其Meta data

# 程式

## 元件清單及其規格描述

## 其他附屬之各種元件

# 測試模型

## 測試計畫

## 測試個案與測試結果資料

# 操作手冊

## 系統元件

## 系統下載及安裝

# 使用手冊

## 使用者操作說明

# 感想

# 參考資料

1. <https://data.gov.tw/dataset/54483>
2. <https://gs.statcounter.com/>
3. <https://reurl.cc/ZebrDV>

# 附錄一 會議記錄

# 附錄二 評審問題回覆