國立臺北商業大學

資 訊 管 理 系

109’資訊系統專案設計

系統手冊

組 別：第109009組

題 目：愛心傘租借系統

指導老師：林宏仁老師

組 長：10646003 黃峻彥

組 員：10646006 詹壹婷

10646007 李恩瑋

10646016 吳宇量

中華民國109年　05　月　10　日

# 第1章 背景與動機

## 1-1 簡介

近年來的科技蓬勃發展，人手一支的智慧型手機，改變了整個世界的運作方式，同時也讓人們幾乎不受限地快速輕鬆的獲取資訊；我們可以迅速知道今天的新聞有哪些、上班上學路上的交通狀況、火車的到站時間，以及天氣預報等資訊，來幫助我們做出行程的安排和準備。

但計畫總是趕不上變化，即使做好再周全的安排，一旦出現自身無法控制的紕漏，或者一個環節出現失誤，就可能打亂整個行程。每個人都應該有過類似的經歷，像是到了車站才發現嚴重誤點，或者因為車禍而造成的道路堵塞；在各種出乎意料的狀況中，最常遇到的應該就屬下雨。

氣象預報並非100%的準確，也不是每個人都有那個空間或者習慣去攜帶摺疊式的雨傘，如果出門在外沒有帶傘卻遇到一場突如其來的雨，大部分的人會選擇乾脆淋雨，或者在附近的便利商店買一把最便宜的傘。

可那種便利商店的傘，定位上就屬於應急用，耐用性絕對不高，經常風稍微大些、使用次數多了些，便出現損壞。再者，即使它足夠耐用，沒有帶傘習慣的人，下次還是可能遇到同樣的情況，而又陷入同樣的問題之中。

既然不一定會用到，卻又具有不定時的需求性，且只有出門在外會使用，我們認為，何不將其用共享的方式提供，讓人們在需要的時候不用怕忘記攜帶，或者使用不合其值的價位再去購買一個幾乎只會使用一次的傘。

這也符合我們在最初思考題目時的其中一個考量，『希望做出來的成果能夠解決我們自身的煩惱』，我們有個成員本身沒有觀看氣象預報也沒有帶傘的習慣，而我自己也是因為搭乘火車上下學的關係充分感受到手拿濕漉漉的傘在尖峰時期的火車捷運上有多麼不方便

如果出門時，我們可以不必在乎降雨機率，也不必在人擠人的大眾運輸上讓濕漉漉的傘將雨水抹上全身，更不用在乎幾乎空著的愛心傘桶是否難得有了一把正常的雨傘在裡面。

下雨時就只要到站點使用手機操作，便可以借取一把雨傘，免於變成落湯雞的命運，到達目的地時只要在使用手機操作，手上的傘就可以留在原地讓自己輕鬆進續行程，那是相當夢幻的一件事。

## 1-2 問題與機會

|  |  |
| --- | --- |
| S | W |
| 1. 操作簡單，流程方便快速  2. 站台構造簡單成本低，擴站方便 | 1. 每個站的傘數量少，難以應付短時間的大量需求 |
| O | T |
| 1. 北部區域大眾運輸交通工具發達，行人較多  2.現有市場競爭者稀少又不衝突 | 1.市場的進入門檻相當低  2.人們對於傘的需求是即時性質的 |

經過分析後我們認為應該採用WT策略，外部的威脅會對應到我們本身的劣勢產生共鳴，所以主要目標在於解決站台傘數可能不足以應付某些地區短時間內產生的大量需求，像是全台人流量最大的大眾運輸站點-台北火車站，我們可以設立較大型的站點或者擁有儲存大量雨傘能力的機台，以解決劣勢

即使現在幾乎每個人都備有雨傘，但人在外時身上沒有傘的情況還是經常發生，而在大眾運輸交通工具蓬勃發展的台灣，雨衣這種不好收納的雨具也並非一個好的選擇，又或者雨傘損壞時不知道去哪修理，只好掏錢再買一把。

雨傘的單價不高，不同於其他B2C的共享經濟，像是單車、機車之類，屬於價位高昂，但是又絕非時常會用到的東西，但雨傘具有比起其他的共享經濟還要高的必要性，尤其在北部捷運範圍內，交通極其發達，想去的地方或者上班上課的所在地幾乎都能使用捷運加上步行抵達，且在近年捷運公司的政策當中，這也是最經濟實惠的一種交通方式。

只是人們家中都還是多多少少備有雨傘，如何擴大客群，吸引更多的使用者，甚至瞄準那些本來出門就會帶傘的民眾也是一種挑戰。

## 1-3 相關系統探討

我們將以台灣的即享傘作為對比，在流程、站點、費用等不同的項目做比較。

|  |  |  |
| --- | --- | --- |
|  | i心傘 | 即享傘 |
| 站點 | 小傘架 | 類似販賣機 |
| 擴展難易度 | 以現有地標(ex:便利商店、車站)放置小型傘架，擴展方便 | 體積大，較難使用現有商家或車站作為據點 |
| 位置 | 以雙北為主 | 僅在中部 |
| 流程 | 下載APP->註冊->點選租借按鈕->掃描站台的QR code->租借 | 下載APP->註冊->設定綁定信用卡->點取借傘按鈕->將手機上的QR code給機器掃描->租借 |
| 費用 | 暫定免費，對於惡意使用者以封鎖等方式懲罰 | 第1-4小時收10元  第5-8小時再收10元  第9-12小時再收10元  第13-24小時再收10元。若自租借起五天未歸還，或者傘具出現非正常使用造成的損毀、販售、抵押、典當等行為，則將額外付出200元 |
| 付費方式 | 無 | 註冊時綁定的第三方信用卡 |

# 第2章 系統目標與預期成果

## 2-1 系統目標

### 改變人們的生活方式

出門不再需要帶傘，甚至不需要看天氣預報。一但下雨就走到最近的站點，快速的借取一把傘，安全乾爽的到達目的地之後也不用再占著一隻手拿著雨傘，或者煩惱著濕漉漉的折疊傘無法收進包包，只要將傘歸還，就能輕輕鬆鬆地繼續接下來的行程。

又或者忘記帶傘時偏偏遇上大雨，不需要花冤枉錢再去買一把只會使用一次的雨傘，或者繞道到幾乎不會有東西的愛心傘桶碰運氣，只要手機拿出來，簡單的操作，快速的借取一把雨傘後就能慢慢的走回家，而傘只需要在時間內歸還即可，或許就在下一次出門時。也不用擔心會忘記還傘導致帳號被加入黑名單，APP會推送通知讓你記得要歸還雨傘

而站台就廣泛的架設在便利商店門口或者捷運站出口、火車站出口或者公車站上，隨處即可見得。還能透過資料蒐集，觀察哪些站點的租借率或哪些區域傘的流動率特別高，彈性的調整站點的傘的總數，並調查歸還率較低的站點，做出適當的處置。並且在不佔用過多空間的前提下，增加一個站點內擁有的雨傘數量，減少人力補充。

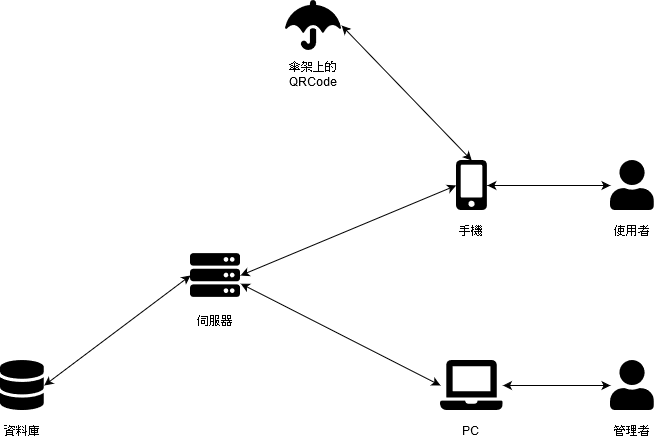
未來，希望能擴大設站的範圍，不僅限於首都和車站、地標，而是到社區門口設立，讓借還傘更加方便。雨傘毀損時使用者能夠透過回報系統回報，進行即時的更換並控制，也能透過懲罰惡意毀損行為，解決共享經濟中最常出現的問題:商品被不特定多數人所使用的不確定性狀況。也希望能將同樣的技術和模式運用在其他可以共享的物品上，幫助人們的生活更加便利簡單。

## 2-2 預期成果

吸引使用者使用此系統，嘗試不用帶傘也不怕的生活。註冊到借取能夠控制在一定的時間內，避免出現操作上過於麻煩而讓潛在客群止步的狀況。讓系統蒐集使用者的反應回饋進行分析處理，像是調整站點雨傘補充的頻率，或者對其他狀況作出相應的對策。推播通知使用者歸還期限，避免使用者忘記歸還，然後過濾出逾期不還者或惡意損毀盜用者，並對其做出懲處。

# 第3章 系統規格

## 3-1 系統架構

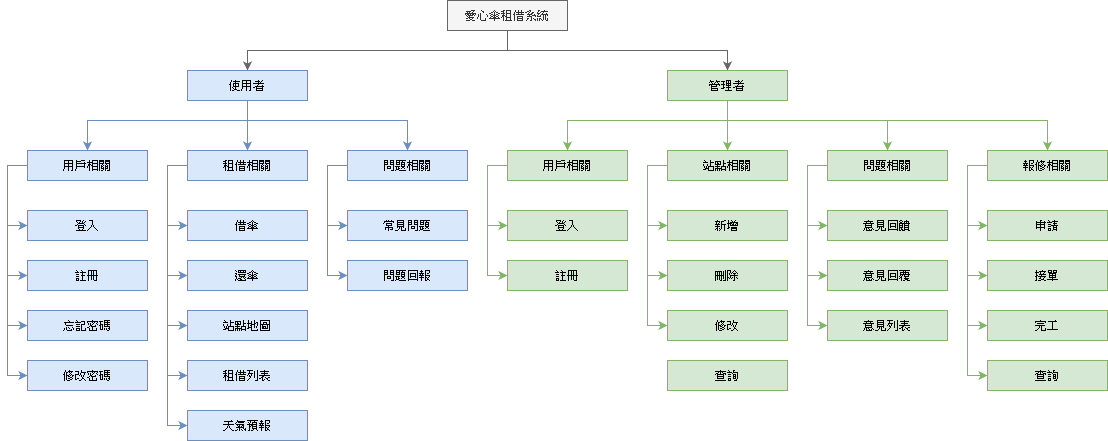


#### 圖 3-1-1 系統架構圖

要借傘時，使用者透過手機APP選取借傘功能，調出QRCode掃描器，掃描傘架上的QRCode取得傘架上的資料，接著APP會將使用者資料連同剛剛的站台資料發送至伺服器，伺服器會確認使用者是否擁有借傘的資格(沒有未歸還、惡意損毀…等紀錄)，確認後會發送解鎖訊號給傘架，請傘架執行開傘的動作，並讓使用者的帳號進入到借傘狀態。

還傘時，使用者再次執行同樣的動作，選取還傘按鈕後掃描傘架上的QRCode，等待APP將訊號發送至伺服器的期間，將雨傘放進傘架並將對準鎖頭，伺服器會確認使用者的帳號是否處於借傘的狀態，確認後會打訊號給傘架，把傘架鎖起來。

管理者則可以使用網頁瀏覽器進行網頁版的管理者介面瀏覽，可以瀏覽站台剩餘的雨傘數以及空著的站台數，並觀看現在借出的傘的總數以及相對應的借取者，以此對傘架的設置地點及雨傘的提供進行管理。



#### 圖 3-1-2 系統功能架構圖

## 3-2 系統軟、硬體需求與技術平台

#### 表 3-2-1 伺服器端規格表

|  |  |
| --- | --- |
| 伺服器端規格 | |
| 作業系統 | Windows Server 2012 R2 Datacenter |
| 開發環境 | Windows 10 |
| 整合式開發環境(IDE) | IntelliJ IDEA |
| 程式語言 | Java 11 |
| 框架 | Spring Boot 2.2.6 |
| 套件管理 | Gradle |
| 伺服器 | Apache Tomcat 9.0.33 Server |
| 資料庫 | MySQL |

#### 表 3-2-2 手機端規格表

|  |  |
| --- | --- |
| 手機端規格 | |
| 作業系統 | Android 10, iOS 13.4 |
| 開發環境 | macOS 10.15.4 |
| 整合式開發環境(IDE) | Android Studio 3.6.1, XCode 11.4 |
| 程式語言 | Dart 2.7.0 |
| 框架 | Flutter 1.12.13 |
| 套件管理 | Pub |

#### 表 3-2-3 網頁端規格表

|  |  |
| --- | --- |
| 網站端規格 |  |
| 作業系統 | Windows Server 2012 R2 Datacenter |
| 開發環境 | Windows 10 |
| 整合式開發環境(IDE) | Visual Studio Code |
| 程式語言 |  |
| 框架 |  |
| 套件管理 |  |
| 伺服器 |  |

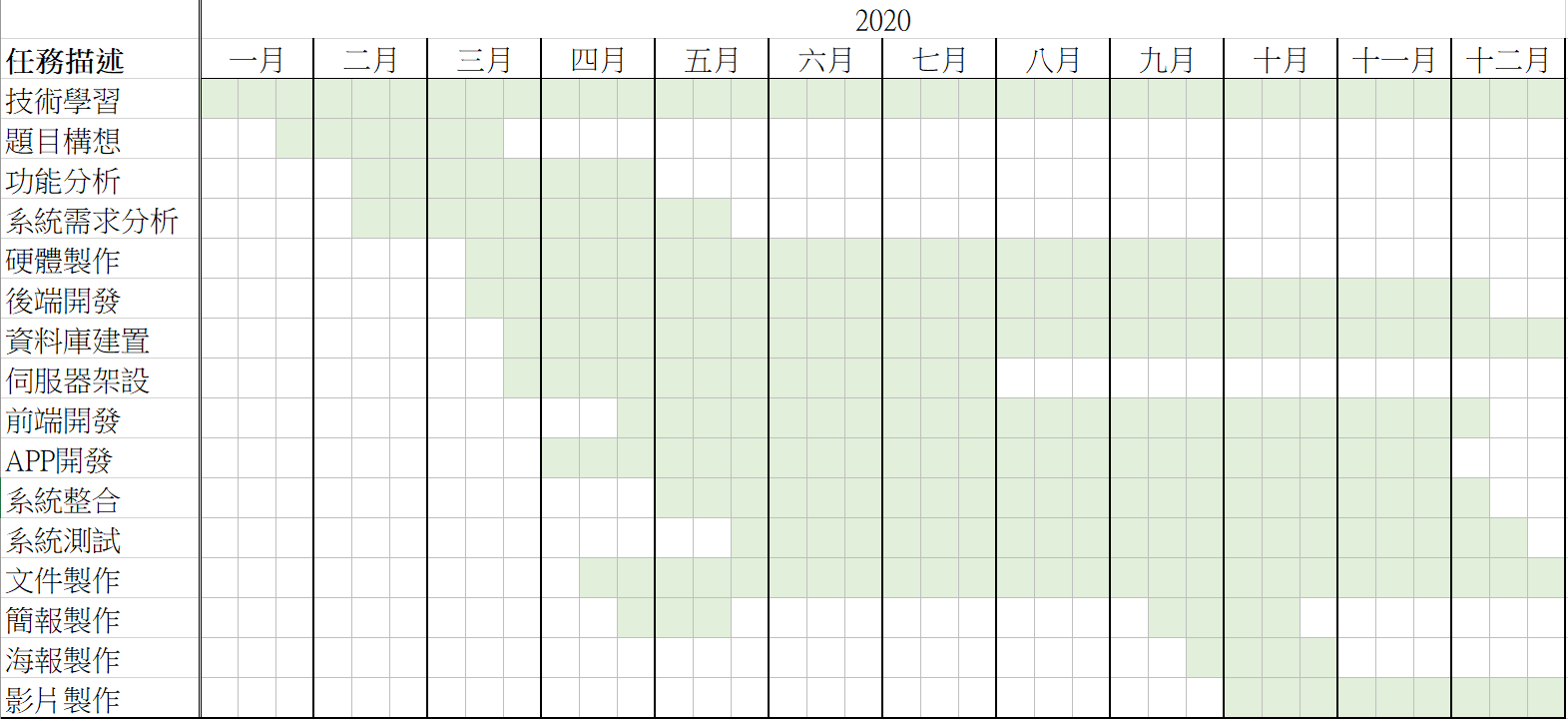
## 3-3 開發標準與使用工具

#### 表 3-3-1 使用工具表

|  |  |
| --- | --- |
| 開發輔助工具 | |
| 編輯器 | IntelliJ IDEA, Android Studio, XCode |
| 資料庫管理工具 | MySQL Workbench, DataGrip |
| 版本控制工具 | Git |
| API 測試工具 | Postman |
| API 文件 | Swagger |
| 文件撰寫工具 | |
| 文件製作 | Microsoft Word 2019 |
| 簡報製作 | Microsoft PowerPoint 2019 |
| UML 工具 | Visual Paradigm for UML 7.2 Enterprise Edition |
| 介面設計工具 | |
| 雛形設計 | Adobe XD |
| 製圖工具 | Adobe Photoshop, Adobe Illustrator |
| 專案管理工具 | |
| 溝通工具 | LINE, Telegram |
| 版本控管 | GitHub |
| 影片軟體 | |
| 剪輯軟體 | Adobe Premiere Pro |

# 第4章 專案時程與組織分工

## 4-1 專案時程：甘特圖或PERT／CPM圖



## 4-2 專案組織與分工

●：主要負責 ○：協助

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | 10646003  黃峻彥 | 10646006  詹壹婷 | 10646007  李恩瑋 | 10646016  吳宇量 |
| 企劃 | 主題構想 | ● |  |  | ○ |
| 需求功能分析 | ○ |  | ○ | ● |
| 系統分析 | ○ |  | ● | ○ |
| 美術 | 介面設計 | ● | ○ |  |  |
| Logo設計 | ● |  |  |  |
| 系統 | 網頁設計 |  | ● |  |  |
| 前端開發 |  | ● |  |  |
| 後端開發 |  |  | ● |  |
| 資料庫建置 |  |  | ● |  |
| 伺服器架設 |  |  | ● |  |
| APP設計 | ● |  |  |  |
| 硬體製作 |  |  |  | ● |
| 報告 | 文件撰寫 | ● |  | ○ | ○ |
| 簡報製作 |  | ● |  | ○ |
| 海報製作 |  |  |  |  |
| 影片製作 |  |  |  |  |