國 立 臺 北 商 業 大 學

資 訊 管 理 系

112 資訊系統專案設計

**系統手冊**

****

**組 別：第112401組**

**題 目：智能門鎖兼門禁管理系統**

**指導老師：唐震老師**

**組 長：10946019 吳貫瑋**

**組 員：10946020 范紘齊 10946023 王信翰**

**10946042 林旻賢**

**中華民國112年11月22日**

**目錄**

[**第 1 章 前言 6**](#_heading=h.gjdgxs)

[1-1 背景介紹 6](#_heading=h.30j0zll)

[1-2 動機 7](#_heading=h.3znysh7)

[1-3 系統目的與目標 7](#_heading=h.2et92p0)

[1-4 預期成果 7](#_heading=h.tyjcwt)

[**第 2 章 營運計畫 8**](#_heading=h.3dy6vkm)

[2-1 可行性分析 8](#_heading=h.1t3h5sf)

[2-2 商業模式 — Business model 8](#_heading=h.4d34og8)

[2-3 市場分析 — STP 10](#_heading=h.2s8eyo1)

[2-4 競爭力分析 SWOT 分析 10](#_heading=h.17dp8vu)

[**第 3 章 系統規格 12**](#_heading=h.26in1rg)

[3-1 系統架構 12](#_heading=h.lnxbz9)

[3-2 系統軟、硬體需求與技術平台 12](#_heading=h.35nkun2)

[3-3 使用標準與工具 13](#_heading=h.44sinio)

[**第 4 章 專案時程與組織分工 14**](#_heading=h.z337ya)

[4-1 專案時程 14](#_heading=h.3j2qqm3)

[4-2 專案組織與分工 15](#_heading=h.4i7ojhp)

[**第 5 章 需求模型 17**](#_heading=h.3whwml4)

[5-1 使用者需求 17](#_heading=h.2bn6wsx)

[5-2 使用個案圖 18](#_heading=h.49x2ik5)

[5-3 使用個案描述 19](#_heading=h.2p2csry)

[5-4 分析類別圖 24](#_heading=h.147n2zr)

[**第 6 章 設計模型 25**](#_heading=h.3o7alnk)

[6-1 循序圖 25](#_heading=h.23ckvvd)

[6-2 設計類別圖 30](#_heading=h.32hioqz)

**第7章 資料模型**

7-1 實體關聯圖或資料結構圖

7-2 資料字典

**第八章 資料庫設計**

8-1 資料庫關聯圖：需註明參考關係及限制。

8-2 表格及其Metadata

**第九章 程式**

9-1 軟體架構與程式清單

9-2 程式規格描述

**第10章 測試模型**

10-1 測試計畫：說明採用之測試方法及其進行方式

10-2 測試個案與測試結果。

**第11章 操作手冊**

介紹系統之元件及其安裝及系統管理

**第12章 使用手冊**

介紹各畫面、操作之移轉，以類似State Transition Diagram之表示之

**第13章 感想**

說明感想與建議

**第14章 參考資料**

參考資料

**圖目錄**

[▲ 圖 2-2-1 商業模式圖](about:blank) 8

[▲ 圖 3-1-1 系統架構圖](about:blank) 12

[▲ 圖 5-2-1 使用個案圖](about:blank) 18

[▲ 圖 5-3-1 密碼輸入開鎖](about:blank) 19

[▲ 圖 5-3-2 指紋辨識開鎖](about:blank) 19

[▲ 圖 5-3-3 聲音辨識開鎖](about:blank) 20

[▲ 圖 5-3-4 影像辨識開鎖](about:blank) 20

[▲ 圖 5-3-5 設定門鎖密碼](about:blank) 21

[▲ 圖 5-3-6 查詢系統日誌](about:blank) 22

[▲ 圖 5-3-7 後台登入](about:blank) 22

[▲ 圖 5-3-8 查詢權限](about:blank) 23

[▲ 圖 5-4-1 分析類別圖](about:blank) 24

[▲ 圖 6-1-1 密碼輸入開鎖](about:blank) 25

[▲ 圖 6-1-2 指紋辨識開鎖](about:blank) 26

[▲ 圖 6-1-3 聲音辨識開鎖](about:blank) 27

[▲ 圖 6-1-4 影像辨識開鎖](about:blank) 27

[▲ 圖 6-1-5 後台登入](about:blank) 28

[▲ 圖 6-1-6 設定門鎖密碼](about:blank) 28

[▲ 圖 6-1-7 查詢系統日誌](about:blank) 29

[▲ 圖 6-1-8 查詢權限](about:blank) 29

[▲ 圖 6-2-1 設計類別圖](about:blank) 30

[▲ 圖 9-1-1](about:blank) 程式檔案架構-總覽 33

[▲ 圖 9-1-2](about:blank) 程式檔案架構-Arduino程式 33

[▲ 圖 9-1-3](about:blank) 程式檔案架構-ESP32開發板程式 33

[▲ 圖 9-1-4](about:blank) 程式檔案架構-門禁系統App 34

**表目錄**

[▼ 表 2-4-1 SWOT分析 10](#_heading=h.3rdcrjn)

[▼ 表 3-2-1 系統軟、硬體需求 12](#_heading=h.1ksv4uv)

[▼ 表 3-3-1 使用標準與工具 13](#_heading=h.2jxsxqh)

[▼ 表 4-1-1 時程甘特圖 14](#_heading=h.1y810tw)

[▼ 表 4-2-1 工作分配表 15](#_heading=h.2xcytpi)

[▼ 表 5-1-1 功能性需求表 17](#_heading=h.qsh70q)

[▼ 表 5-1-2 非功能性需求表 17](#_heading=h.1pxezwc)

▼表 9-1-1 程式清單 34

[▼ 表 9-2-1 程式](#_heading=h.1pxezwc)規格描述-Main.ino 37

[▼ 表 9-2-2 程式](#_heading=h.1pxezwc)規格描述-esp32\_connect.ino 40

[▼ 表 9-2-3 程式](#_heading=h.1pxezwc)規格描述-SecActivity.kt 43

[▼ 表 9-2-4 程式](#_heading=h.1pxezwc)規格描述-NewAddedUser.kt 46

[▼ 表 9-2-5 程式](#_heading=h.1pxezwc)規格描述-Adapter.kt 46

[▼ 表 9-2-6 程式](#_heading=h.1pxezwc)規格描述-AppUserRegister.kt 49

# 前言

## 背景介紹

現代社會中，居住安全是人們最關心的問題之一，然而傳統的門禁只採用磁扣、密碼等方式進行開門，這種方式有著許多缺陷，例如磁扣容易被仿冒，使用者容易忘記密碼等等。為了解決這些問題，人們轉而開始使用更加智能化的門禁系統，例如可使用指紋辨識、臉部識別等技術的門鎖。

而用Arduino來操控門鎖是基於Arduino開發板高度的靈活性和可定制性而設計的多因素驗證（multi-factor authentication）辨識功能，透過搭載指紋、聲紋及臉部辨識等模組，就能夠實現快速辨識身分並開門的功能，不再需要擔心忘記密碼或磁扣遺失的問題。而作為智能辨識功能發生故障時的保險，也可以採用輸入密碼的功能來進出入。同時也可以設定警報系統，例如一旦有人試圖撬門或強行進入，門鎖就會發出警報聲，提醒使用者注意安全等。

只有硬體門鎖肯定不夠，由於現代智慧型手機的普及，也有不少設計用來開門的門禁App被使用，以IoT的形式來運作，配合門鎖來運作的門禁管理系統可以大幅提升監控安全的方便性，用戶可透過App來查詢進出紀錄和門鎖狀況等即時資訊，並透過雲端連線來進行操作。

## 動機

Arduino開發板可以藉由裝上各式各樣的模組來達成很多有趣的應用，包括Wifi偵測、指紋、影像跟聲音的辨識等，也能夠用Arduino來控制門鎖。但網路上很少看到把門鎖同時結合影像、聲音及指紋開鎖，同時又兼具門禁管理系統App的應用，且組員有一部分都有學習過硬體的背景，因此決定試著開發這個整合許多開鎖功能的智慧門鎖。

而在學校的課程中有接觸過開發手機App的相關知識，決定學以致用，試著為這個智能門鎖專門開發一款監控App。

## 系統目的與目標

透過智慧門鎖系統實現一個智能、安全的門禁管理系統。首先以結合了影像、聲音、指紋等多種開鎖方式的門鎖來實現便捷且準確的開鎖，並且結合後台管理系統，讓用戶能夠在手機App上操作權限管理、系統日誌查詢、進出人員管制、報警設置等功能，對門禁系統進行全面管理和控制。

## 預期成果

整合影像、聲音、指紋等多種開鎖方式，並建立後台管理系統讓用戶能夠管理、查閱資訊和進行控制，從而實現既智能又安全的門禁管理系統。

# 營運計畫

## 可行性分析

隨著人們安全意識的增強，會優先選擇能提供較高安全性的門鎖。而Arduino門鎖透過影像、聲音、指紋等多種開鎖方式能使門鎖具有較高的安全性，且Arduino門鎖的開發成本相對較低，價格也較傳統門禁系統更具競爭力，有機會吸引更多消費者。

市場上還存在許多機構和企業需要安全可靠的門禁系統，例如學校、辦公室、商場等，而這些機構也會需要後台管理系統以方便管理，因此具備後台管理系統的智慧門鎖在這些市場上有相當大的競爭力。

## 商業模式 — Business model

▲ 圖 STYLEREF 2 \s 2-2 SEQ ▲\_圖 \\* ARABIC \s 2 1 商業模式圖

1. 目標客層：目標客戶為住戶、辦公室、管制人員進出的場所及尋求安全性、方便性的人。
   1. 房客：房客需要門鎖保護自己的財產和隱私，但大多數房東不會更換門鎖或是門鎖老舊，因此他們可能會購買可自行安裝的Arduino門鎖來保護自己的安全。
   2. 管制人員進出的場所：例如辦公室可以使用Arduino門鎖為員工和訪客提供安全的門禁系統，同時也可以根據需求來操控訪問權限，方便管理員進行管理和控制。
2. 價值主張：主要價值在於其智能化、便利性和安全性，可以實現人臉、聲音或是用指紋辨識開門等功能，提高安全性和便捷性。
3. 通路：可以透過網絡和實體店面銷售。
4. 顧客關係：可以透過提供保固服務、產品功能升級等建立良好的客戶關係。
5. 收益流：主要收益來源是產品及相關配件的銷售收入，例如電池、鎖芯、外殼等。
6. 關鍵資源：產品設計、技術人才和生產設備。
7. 關鍵活動：售後服務和技術支援，包括產品使用指導、故障排除、維修保養等。
8. 關鍵合作夥伴：硬體供應商、軟體開發者、電商平台。
9. 成本結構：產品設計和開發、生產費用，硬體設計和軟體開發需要投入人力、時間和資金進行研發和測試，並保持產品更新和升級。

## 市場分析 — STP

1. 市場分割
   1. 住宅：公寓、別墅、透天厝、電梯等。
   2. 管制人員進出的場所：辦公室、飯店、商場等。
2. 目標市場

主要是居住在公寓、別墅、住宅區等環境中的個人和家庭及需要管制人員進出的場所，對於家庭安全十分關心的人，會更有意願購買能提供較高安全性的門禁系統以保障自身的人身和財產安全，且Arduino門鎖的簡單設計和低廉的價格會增加消費者的購買意願。需要管制人員進出的場所則是會因為後台管理系統能提供的各種管理功能而提高購買意願。

1. 市場定位
   1. 價格定位：較為低廉的價格可以吸引到更多的客戶。
   2. 安全性：強調Arduino門鎖的安全性可以吸引到尋求安全性的客戶。
   3. 功能性：後台管理系統可以吸引到需要管理門禁系統的客戶。

## 競爭力分析 SWOT 分析

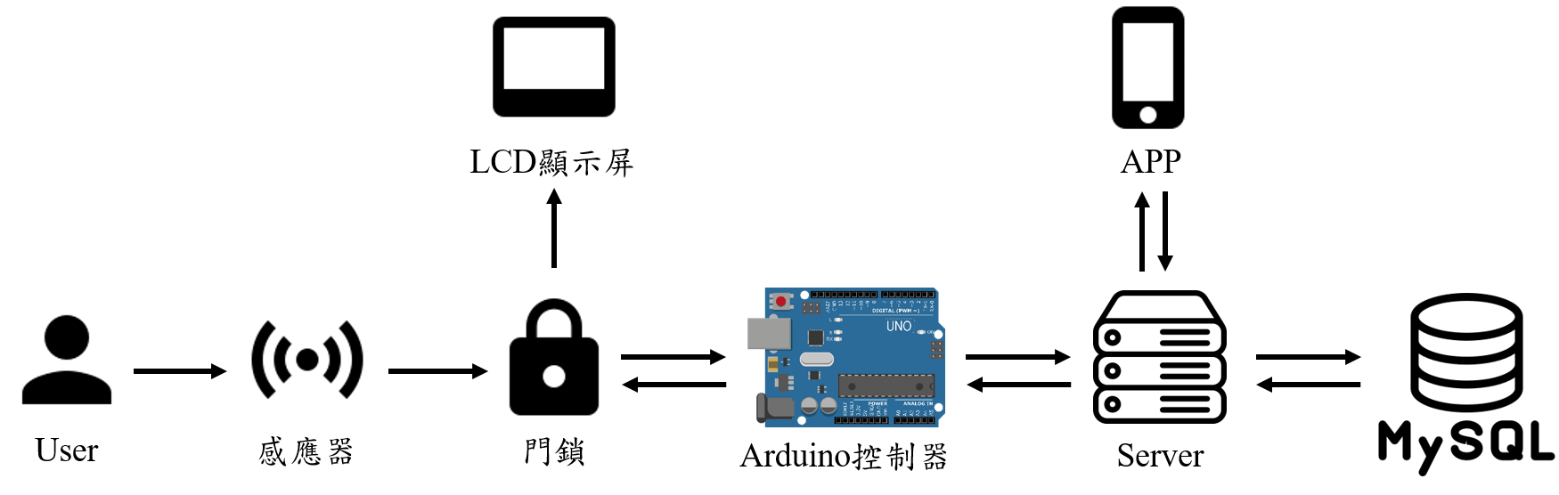
▼ 表 2-4-1 SWOT分析

|  |  |
| --- | --- |
| S 優勢 | W 劣勢 |
| 1. 整合影像、聲音、指紋等多種開鎖方式，提高門鎖的安全性。 2. 透過開鎖的生理訊號可以識別進入人員並在後台進行管理。 3. Arduino平台開發成本相對較低。 | 1. 相較於傳統的門鎖成本會較高。 2. 存在可能被駭客攻擊的風險。 3. 需要穩定的電源供應。 |
| O 機會 | T 威脅 |
| 1. 隨著人們安全意識的增強，會優先選擇能提供較高安全性的門鎖。 2. 隨著新技術的不斷發展，例如人工智慧、機器學習等，Arduino門鎖可以不斷更新和升級，提供更多的功能。 | 1. 市場上存在許多同類產品，競爭較為激烈。 2. 科技發展快速，Arduino門鎖需要不斷進步和升級，否則很有可能會被淘汰掉。 |

# 系統規格

## 系統架構

▲ 圖 STYLEREF 2 \s 3-1 SEQ ▲\_圖 \\* ARABIC \s 2 1 系統架構圖



[▲ 圖 3-1-1](about:blank) 系統架構圖

## 系統軟、硬體需求與技術平台

▼ 表 3-2-1 系統軟、硬體需求

|  |
| --- |
| 硬體需求 |
| 1. Arduino主板 2. ESP32 開發板模組，用Wifi連接至Server 3. 按鈕 / 鍵盤，用於輸入密碼 4. LCD顯示屏，可以顯示門鎖系統的運行狀態、錯誤信息等 5. 指紋辨識模組 6. 語音辨識模組 7. 攝影模組、攝影鏡頭 |
| 軟體需求 |
| 1. Windows / Mac / Linux 2. Arduino IDE 及其程式庫 3. Android手機 |

## 使用標準與工具

▼ 表 3-3-1 使用標準與工具

|  |  |
| --- | --- |
| 使用工具 | |
| 作業系統 | Windows 10 |
| 資料庫伺服器 | MySQL |
| 編輯器 | Arduino IDE、Android Studio |
| 程式語言 | Arduino、Java、Kotlin |
| 版本控制 | GitHub、Fork |
| 文件製作 | Microsoft Word |
| 簡報製作 | Microsoft PowerPoint |
| 專題Logo、海報繪製工具 | Krita |
| UML工具 | diagrams.net |

# 專案時程與組織分工

預期進度

實際進度

## 專案時程

▼ 表 4-1-1 時程甘特圖

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 工作項目  月份 | 2 月 | 3 月 | 4 月 | 5 月 | 6 月 | 7 月 | 8 月 | 9 月 | 10月 | 11月 | 12月 |
| 主題構思 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| 開發工具學習 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| 系統功能分析 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| 硬體組裝 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| 密碼開鎖、指紋辨識功能 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| 影像、聲音辨識功能 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| 硬體測試及整合 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| APP開發 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| Server架設 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| 資料庫開發 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| 系統測試及整合 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
| 系統手冊製作 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

## 專案組織與分工

▼ 表 4-2-1 工作分配表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 項目/組員 | | 10946019 吳貫瑋 | 10946020 范紘齊 | 10946023 王信翰 | 10946035 朱筱娟 | 10946042 林旻賢 | 10846025 呂映珊 |
| 後端開發 | 資料庫建置 | ● |  |  |  |  |  |
| 伺服器架設 |  |  |  |  | ● |  |
| 前端開發 | App介面設計 | ○ | ● |  |  | ○ |  |
| App主程式 | ○ | ● |  |  |  |  |
| App功能發想 | ● |  |  |  | ○ |  |
| 辨識工具 | 軟體 |  | ● |  |  |  |  |
| 硬體 |  | ● |  |  |  |  |
| 美術 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 | ● | ○ |  |  |  |  |
| 海報製作 |  | ● |  |  |  |  |
| 影片製作 |  | ● |  |  |  |  |

**專題成果工作內容與貢獻度表**

|  |  |  |  |
| --- | --- | --- | --- |
| 序號 | 姓名 | 工作內容 | 貢獻度 |
| 1 | 組長  吳貫瑋 | “智能門鎖兼門禁管理系統”專題中: App功能發想、Server開發、資料庫、簡報 | 20% |
| 2 | 組員  范紘齊 | “智能門鎖兼門禁管理系統”專題中:  Arduino UNO/ESP32等硬體辨識工具的採買、組裝、程式撰寫及整合；  門禁系統手機App的介面設計、程式撰寫；  本次專題的Logo、海報繪製、專題相關影片剪輯製作；  專題文件第1、9章；前半學期的專題組內會議的會議記錄人。 | 60% |
| 3 | 組員  王信翰 | “智能門鎖兼門禁管理系統”專題中: | 0% |
| 4 | 組員  林旻賢 | “智能門鎖兼門禁管理系統”專題中:  App功能發想、Server開發、資料庫、後半學期的專題組內會議的會議記錄人 | 20% |
|  | | | 總計:100% |

# 需求模型

## 使用者需求

* 1. 功能性需求

▼ 表 5-1-1 功能性需求表

|  |  |
| --- | --- |
| 功能性需求 | 使用案例 |
| 可以控制門鎖的開關和狀態 | 控制門鎖 |
| 可以透過指紋、影像、聲音辨識等開鎖方式驗證身份 | 身份認證 |
| 可以透過後台管理系統設定不同的使用者權限並進行管理，例如新增、修改、刪除使用者等操作 | 權限管理 |
| 可以透過後台管理系統查詢門鎖操作日誌 | 日誌查詢 |
| 可以記錄進出入的時間 | 進出紀錄 |
| 使用者可以設定門鎖密碼 | 設定密碼 |
| 可以透過 LCD顯示屏提示使用者身份驗證錯誤 | 錯誤提示 |

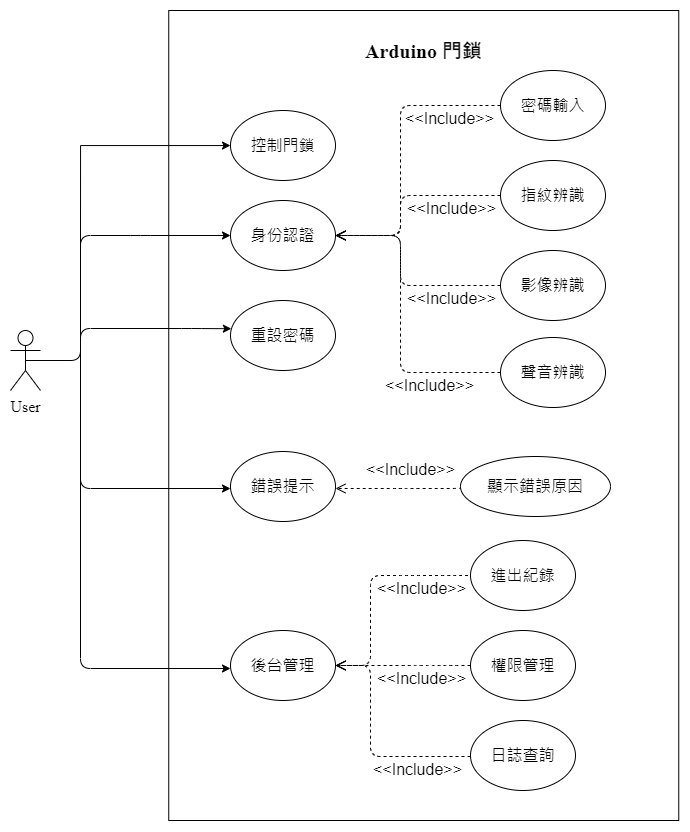
* 1. 非功能性需求

▼ 表 5-1-2 非功能性需求表

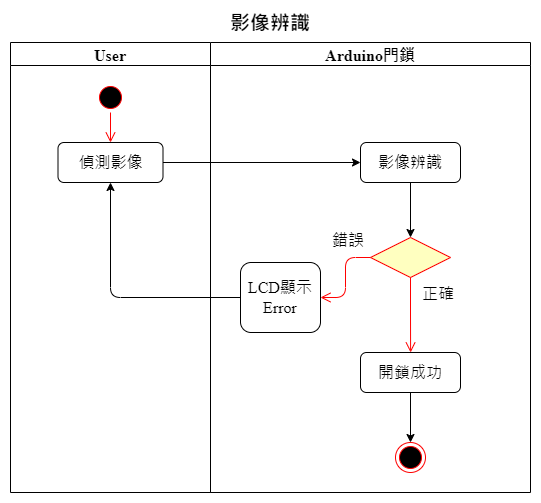
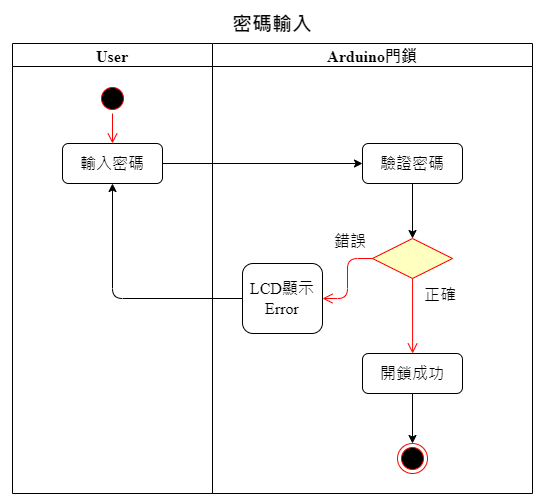
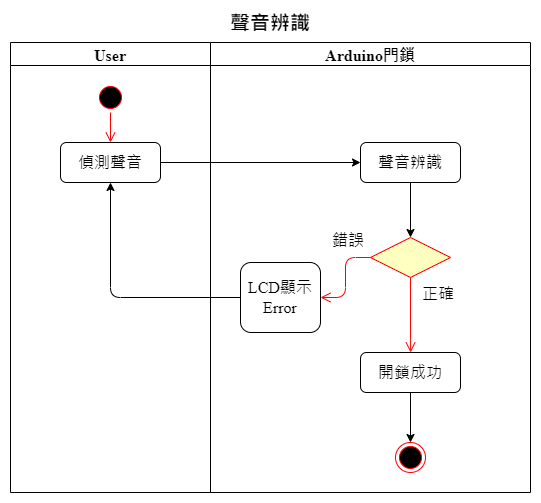
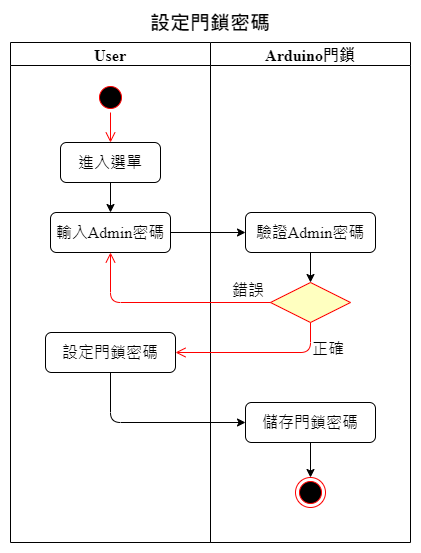
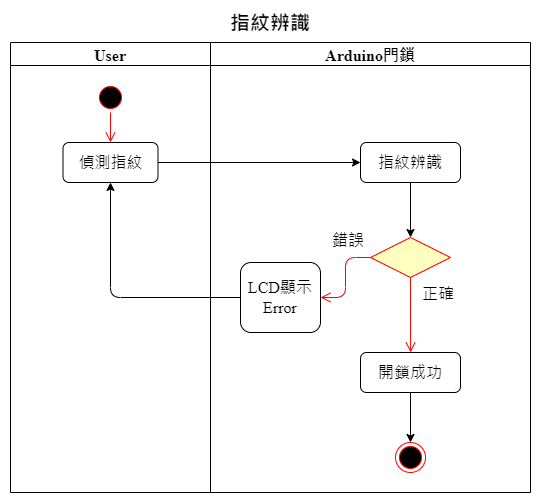
|  |  |
| --- | --- |
| 非功能性需求 | |
| 安全性 | Arduino門鎖提供足夠的安全性保護使用者的財產和隱私 |
| 易用性 | 易於使用，操作方式對使用者友好 |

## 使用個案圖

▲ 圖 STYLEREF 2 \s 5-2 SEQ ▲\_圖 \\* ARABIC \s 2 1 使用個案圖



## 使用個案描述



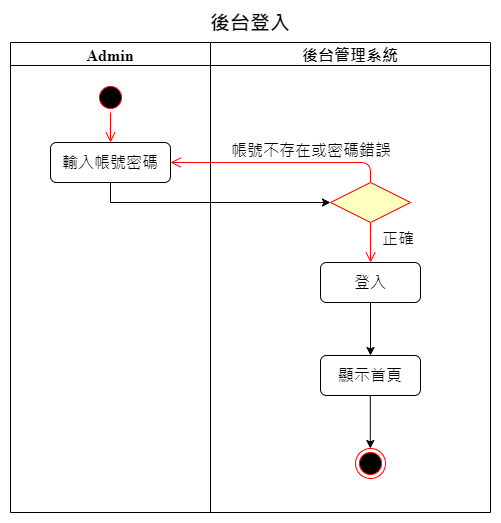
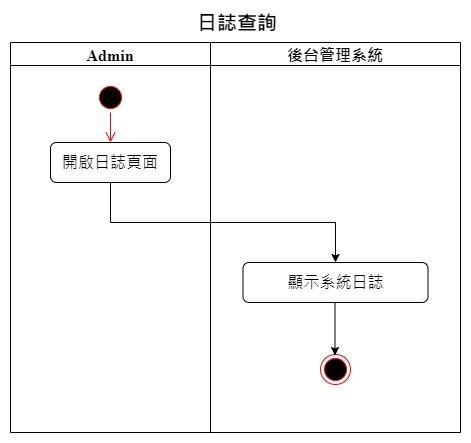
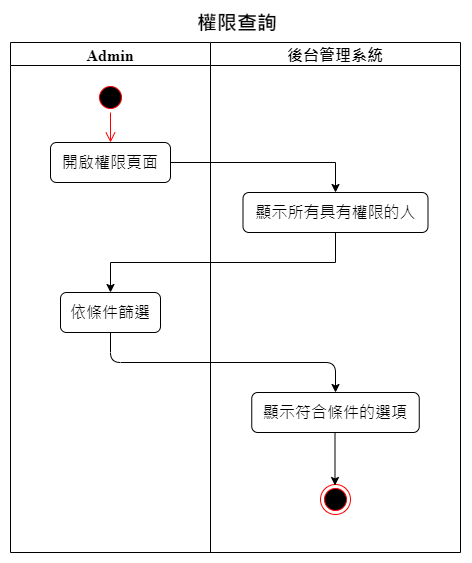
▲ 圖 STYLEREF 2 \s 5-3 SEQ ▲\_圖 \\* ARABIC \s 2 2 指紋辨識開鎖

▲ 圖 STYLEREF 2 \s 5-3 SEQ ▲\_圖 \\* ARABIC \s 2 4 影像辨識開鎖

▲ 圖 STYLEREF 2 \s 5-3 SEQ ▲\_圖 \\* ARABIC \s 2 1 密碼輸入開鎖

▲ 圖 STYLEREF 2 \s 5-3 SEQ ▲\_圖 \\* ARABIC \s 2 3 聲音辨識開鎖

▲ 圖 STYLEREF 2 \s 5-3 SEQ ▲\_圖 \\* ARABIC \s 2 5 設定門鎖密碼

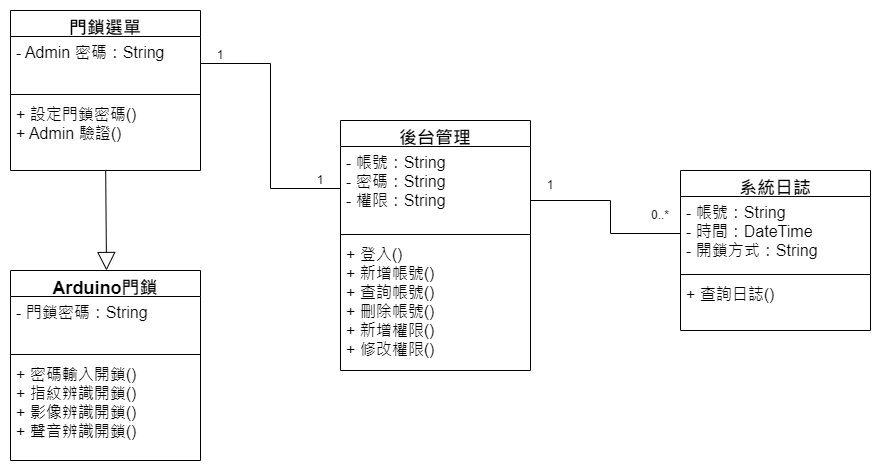


▲ 圖 STYLEREF 2 \s 5-3 SEQ ▲\_圖 \\* ARABIC \s 2 7 後台登入

▲ 圖 STYLEREF 2 \s 5-3 SEQ ▲\_圖 \\* ARABIC \s 2 6 查詢系統日誌

▲ 圖 STYLEREF 2 \s 5-3 SEQ ▲\_圖 \\* ARABIC \s 2 8 查詢權限

## 分析類別圖

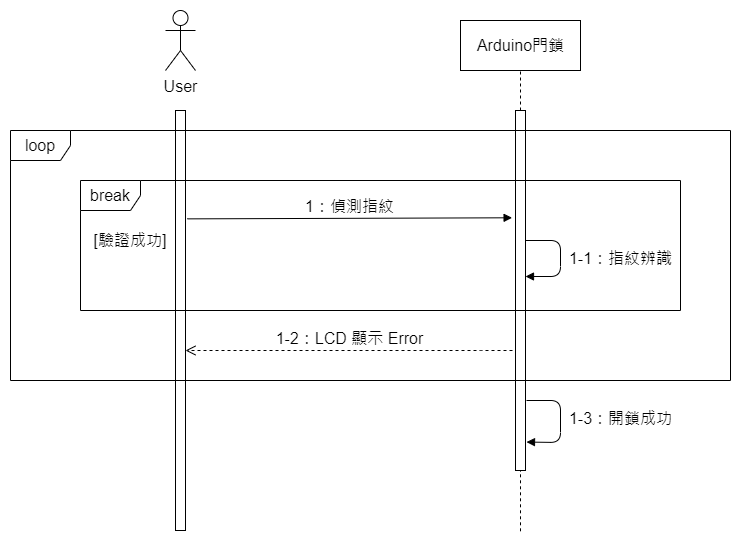
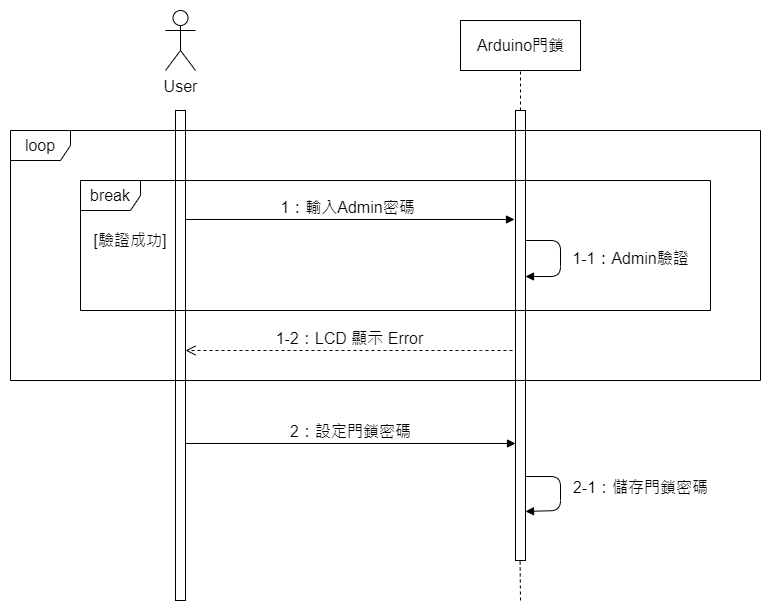
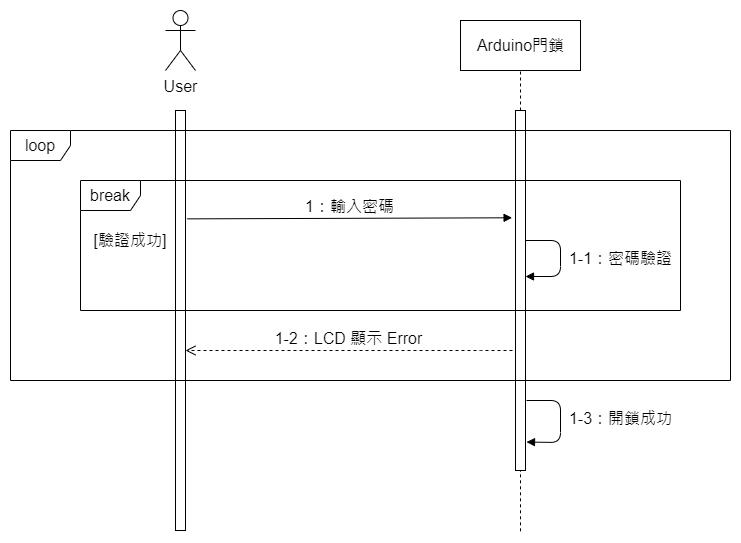
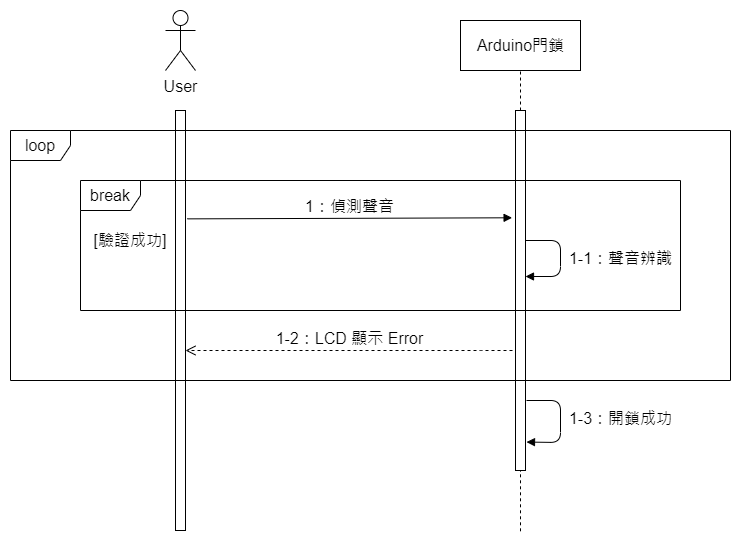


▲ 圖 STYLEREF 2 \s 5-4 SEQ ▲\_圖 \\* ARABIC \s 2 1 分析類別圖

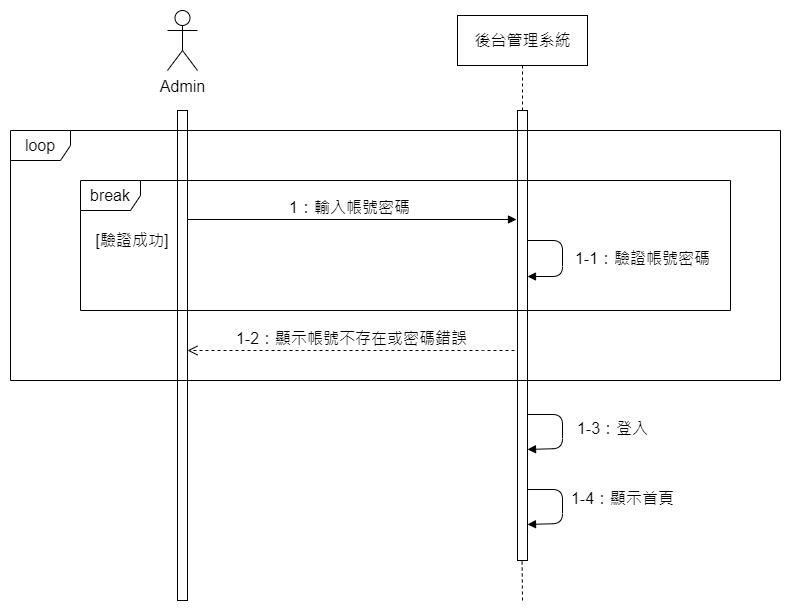
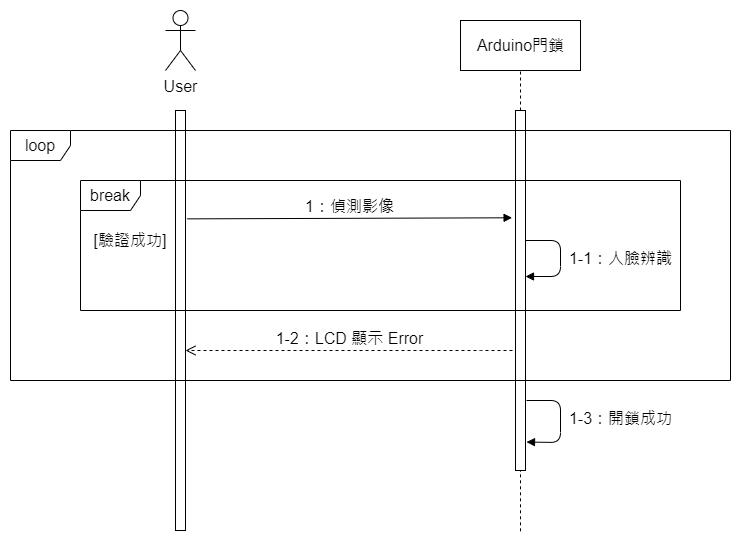
# 設計模型

## 循序圖

▲ 圖 STYLEREF 2 \s 6-1 SEQ ▲\_圖 \\* ARABIC \s 2 1 密碼輸入開鎖



▲ 圖 STYLEREF 2 \s 6-1 SEQ ▲\_圖 \\* ARABIC \s 2 2 指紋辨識開鎖

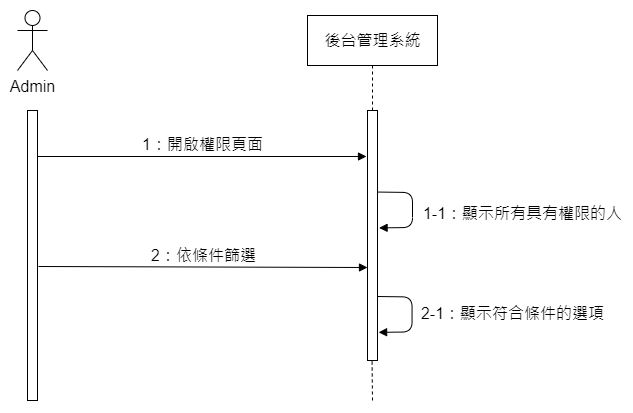
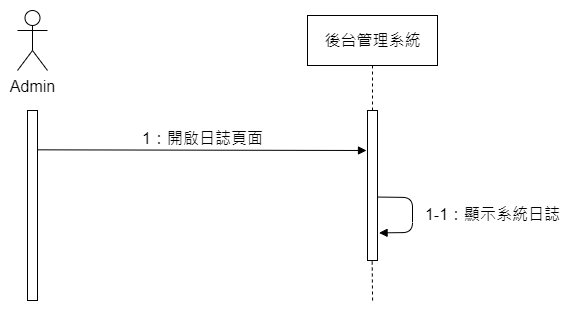


▲ 圖 STYLEREF 2 \s 6-1 SEQ ▲\_圖 \\* ARABIC \s 2 4 影像辨識開鎖

▲ 圖 STYLEREF 2 \s 6-1 SEQ ▲\_圖 \\* ARABIC \s 2 6 設定門鎖密碼

▲ 圖 STYLEREF 2 \s 6-1 SEQ ▲\_圖 \\* ARABIC \s 2 5 後台登入

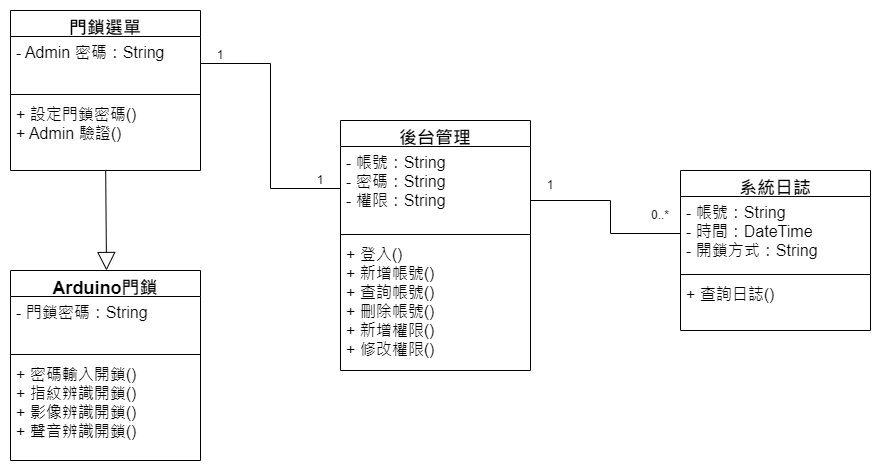
▲ 圖 STYLEREF 2 \s 6-1 SEQ ▲\_圖 \\* ARABIC \s 2 3 聲音辨識開鎖



▲ 圖 STYLEREF 2 \s 6-1 SEQ ▲\_圖 \\* ARABIC \s 2 8 查詢權限

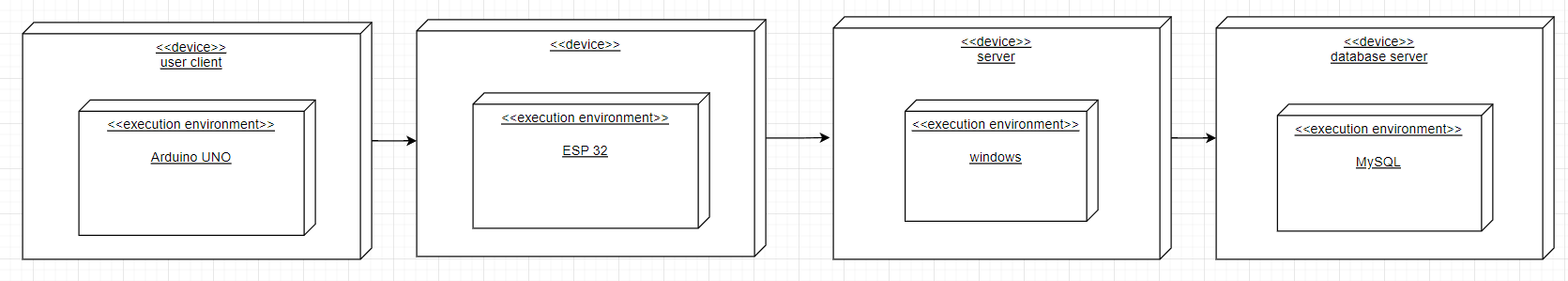
▲ 圖 STYLEREF 2 \s 6-1 SEQ ▲\_圖 \\* ARABIC \s 2 7 查詢系統日誌

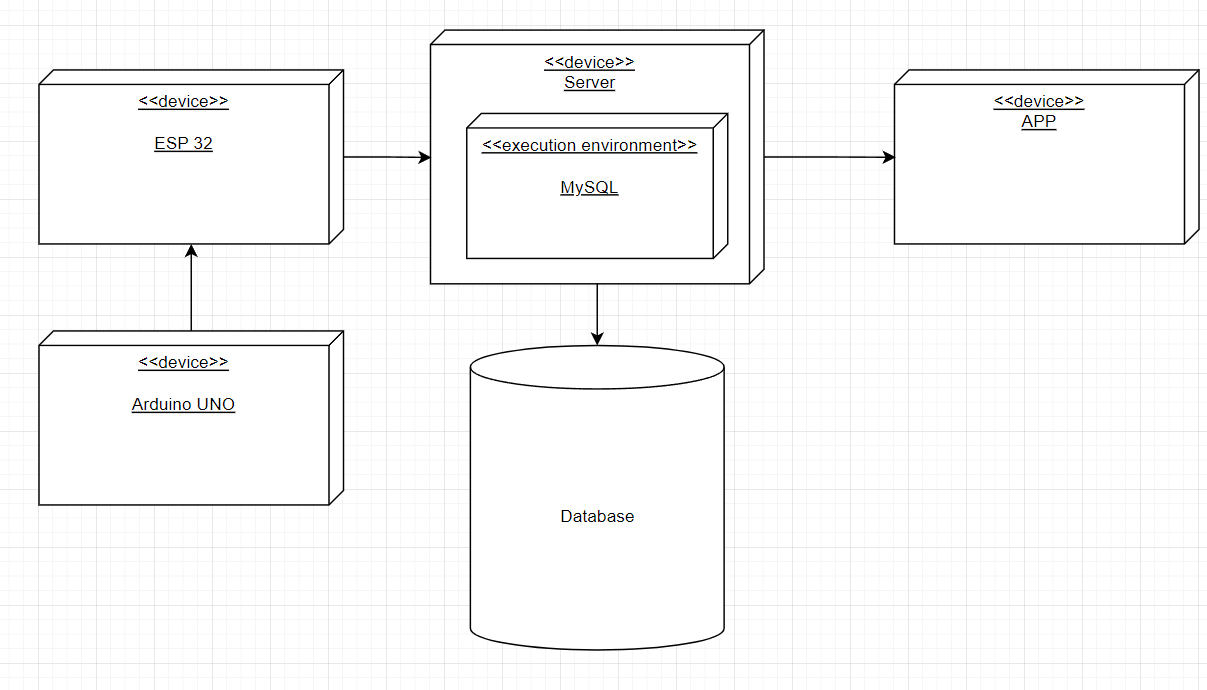
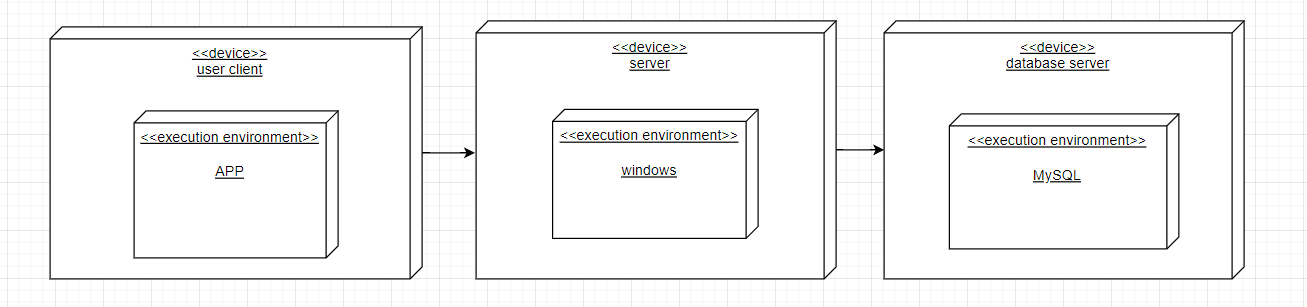
## 設計類別圖

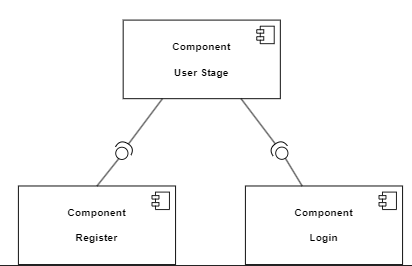


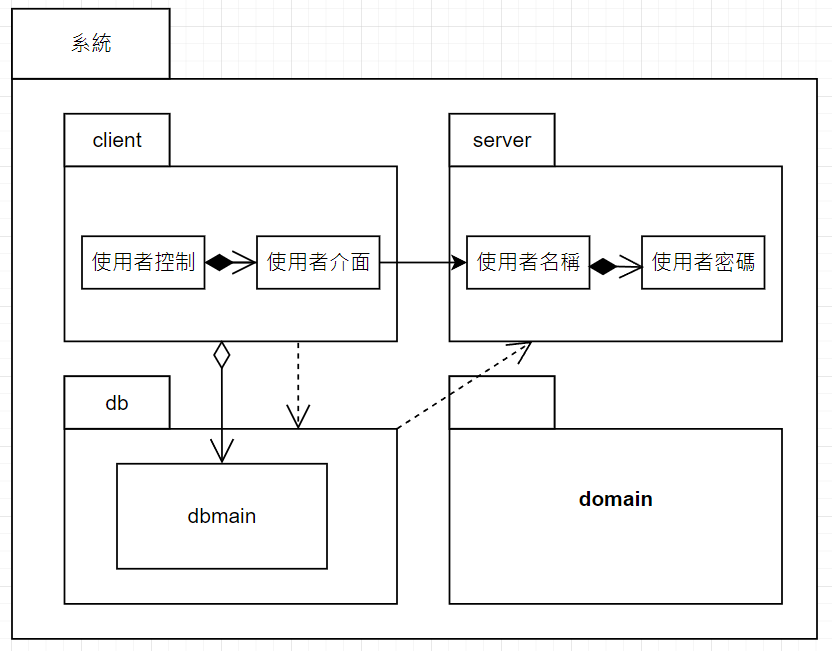
▲ 圖 STYLEREF 2 \s 6-2 SEQ ▲\_圖 \\* ARABIC \s 2 1 設計類別圖

第7章 實作模型

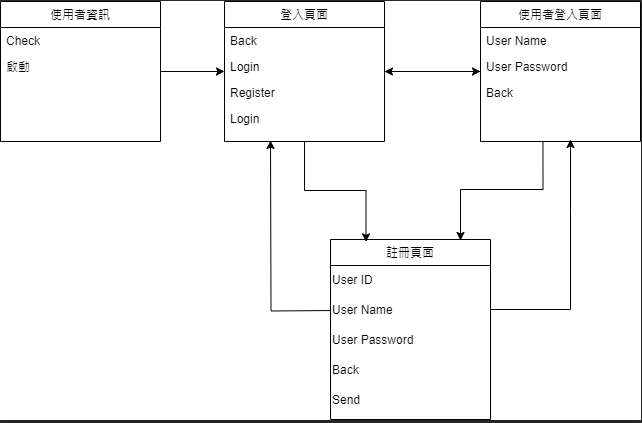






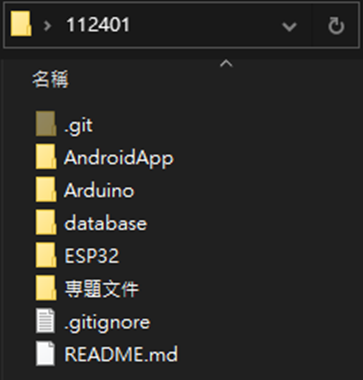


第8章 資料庫設計

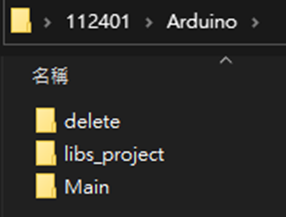


**第9章 程式**

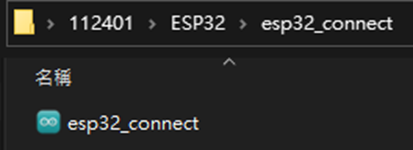
9-1 軟體架構與程式清單



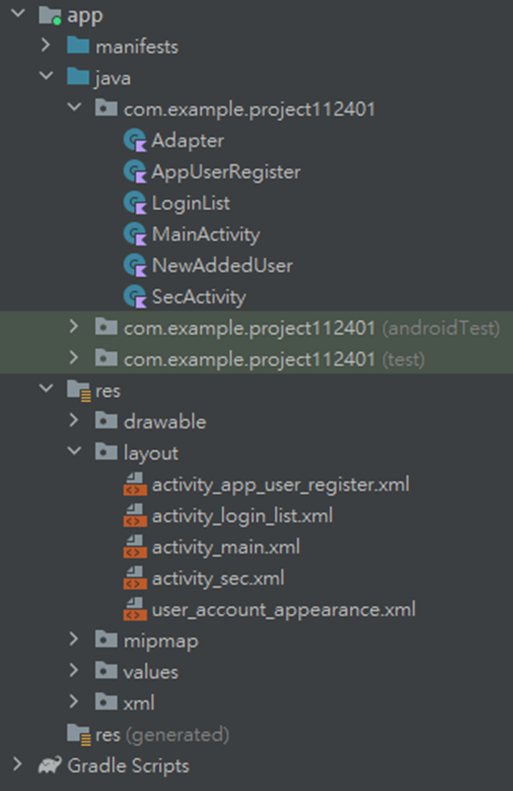
[▲ 圖 9-1-1](about:blank) 程式檔案架構-總覽



[▲ 圖 9-1-2](about:blank) 程式檔案架構-Arduino程式



[▲ 圖 9-1-3](about:blank) 程式檔案架構-ESP32開發板程式



[▲ 圖 9-1-4](about:blank) 程式檔案架構-門禁系統App

[▼ 表 9-1-1 程式清單](#_heading=h.1pxezwc)

|  |  |  |
| --- | --- | --- |
| 資料夾名稱:112401 | | |
| 編號 | 檔案名稱 | 功能 |
| 1-1 | Arduino | 資料夾—存放Arduino UNO相關程式 |
| 1-2 | ESP32 | 資料夾—存放ESP32開發板的程式 |
| 1-3 | AndroidApp | 資料夾—存放門禁系統用的手機App |
| 資料夾名稱: Arduino | | |
| 編號 | 檔案名稱 | 功能 |
| 2-1 | Main | 資料夾—存放Arduino UNO的主程式 |
| 2-2 | libs\_project | 資料夾—存放主程式導入的函式庫 |
| 資料夾名稱: Main | | |
| 編號 | 檔案名稱 | 功能 |
| 3-1 | Main.ino | Arduino UNO的主要辨識身分和指紋註冊功能 |
| 資料夾名稱: esp32\_connect | | |
| 編號 | 檔案名稱 | 功能 |
| 4-1 | esp32\_connect.ino | 尋找特定的Wifi，並傳遞Arduino UNO發出的訊息 |
| 資料夾名稱: project112401 | | |
| 編號 | 檔案名稱 | 功能 |
| 5-1 | SecActivity.kt | 顯示主要的操作選項，如登入、註冊等 |
| 5-2 | LoginList.kt | 顯示可登入的使用者身分 |
| 5-3 | NewAddedUser.kt | 用data class統一使用者的資料形式 |
| 5-4 | Adapter.kt | 資料跟顯示介面之間的橋樑 |
| 5-5 | AppUserRegister.kt | 註冊使用者的介面 |

9-2 程式規格描述

[▼ 表 9-2-1 程式](#_heading=h.1pxezwc)規格描述-Main.ino

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 3-1 | 程式名稱 | Main.ino |
| 目的 | 執行指紋識別功能 | | |
| 部分程式碼 | | | |
| finger\_status = getFingerprintID(); //迴圈中不斷檢查是否有指紋感應  //getFingerprintID()函式中的判斷  p = finger.fingerSearch();  if (p == FINGERPRINT\_OK) { //找到人  isUserFound = true;  Serial.print("Welcome back, User ");  Serial.println(finger.fingerID);  lcd.clear();  lcd.setCursor(0, 0);  lcd.print("Welcome, User");  lcd.print(finger.fingerID);  lcd.print("!");  }    //按下按鈕時  if(key){  pwDigit ++;  userEnter += key;    else if(pwDigit == password.length()){  //當輸入的密碼位數跟預設密碼的位數一樣時  if(userEnter == password){ //正確  Serial.println("\nWelcome!");  }  else{  Serial.println("\nIncorrect Password.");  Serial.println(userEnter);  lcd.print("Wrong password!");  }  }  } | | | |

[▼ 表 9-2-2 程式](#_heading=h.1pxezwc)規格描述-esp32\_connect.ino

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 4-1 | 程式名稱 | esp32\_connect.ino |
| 目的 | 連上程式內設定好的特定Wifi，並以硬體連線的方式傳遞Arduino UNO發出的訊息 | | |
| 部分程式碼 | | | |
| #define RXp2 16  //灰色線, ESP32板上的RX2(GIOP16), 接到Arduino板子上的1號  #define TXp2 17  //白色線, ESP32板上的TX2(GIOP17), 接到Arduino板子上的0號  #define ledPin 32 //GIOP32, 連至LED  bool ledStatus = false; //一開始先讓LED為暗  const char ssid[] = ""; //自家WiFi網路名稱  const char pwd[] = ""; //WiFi密碼    void setup() { //啟動程式時執行一次  Serial.begin(115200);  Serial2.begin(9600, SERIAL\_8N1, RXp2, TXp2);  //第二個Serial, 顯示從16,17腳位接收到的訊息  //第一個參數的數值需跟Arduino UNO板的baud值相同  WiFi.mode(WIFI\_STA); //設置WiFi模式  WiFi.begin(ssid,pwd); //啟動Wifi  //跳出迴圈時代表已成功連線  while(WiFi.status()!=WL\_CONNECTED){  delay(500);  }  Serial.println("\nConnected!");  }    void loop() { //執行後一直運行的動作  //如果Wifi斷掉連線就讓LED為暗,直到重新連上Wifi  if(WiFi.status()!=WL\_CONNECTED){…}  Serial.println(Serial2.readString());  //這裡的Serial顯示從Serial2讀取到的訊息  delay(1000); //設定每幾ms檢查一次從RX跟TX傳來的訊息  } | | | |

[▼ 表 9-2-3 程式](#_heading=h.1pxezwc)規格描述-SecActivity.kt

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 5-1 | 程式名稱 | SecActivity.kt |
| 目的 | 顯示登入跟註冊等按鈕的資訊，並顯示回傳的資料 | | |
| 部分程式碼 | | | |
| registerBtn.setOnClickListener **{***//設定按下按鈕後會觸發的動作* val intentToRegister = Intent(this,AppUserRegister::class.*java*)  startActivityForResult(intentToRegister, 112401002)  **}** loginBtn.setOnClickListener **{** val intentToLogin = Intent(this, LoginList::class.*java*)  startActivityForResult(intentToLogin, 112401003)  **}**    override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {  super.onActivityResult(requestCode, resultCode, data)  val userInfo = findViewById<TextView>(R.id.*userInfo*)  if(requestCode == 112401002){  *//requestCode傳回來時,根據resultCode的不同來顯示不同資訊*  userInfo.*text* =  if(resultCode == *RESULT\_OK*){  "Welcome, " + data?.getStringExtra("user\_Name")  }  else if(resultCode == *RESULT\_CANCELED*){  "Register canceled"  }  }  else if(requestCode == 112401003){  userInfo.*text* =  if(resultCode == *RESULT\_OK*){  "Welcome back, "+data?.getStringExtra("user\_Name")  }  else if(resultCode == *RESULT\_CANCELED*){  "Login canceled"  }  }  } | | | |

[▼ 表 9-2-4 程式](#_heading=h.1pxezwc)規格描述-NewAddedUser.kt

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 5-3 | 程式名稱 | NewAddedUser.kt |
| 目的 | 建立data class作為新註冊的使用者資料的格式 | | |
| 程式碼 | | | |
| data class NewAddedUser(val name : String, val email : String, val password : String) *//User的data格式* | | | |

[▼ 表 9-2-5 程式](#_heading=h.1pxezwc)規格描述-Adapter.kt

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 5-4 | 程式名稱 | Adapter.kt |
| 目的 | 位於前後端之間，把data class的資料格式提供給顯示資料用的xml檔，並讓資料跟顯示介面之間能互動 | | |
| 部分程式碼 | | | |
| class Adapter(private val newUserList : ArrayList<NewAddedUser>) : RecyclerView.Adapter<Adapter.ViewHolder>() {  class ViewHolder(itemView : View) : RecyclerView.ViewHolder(itemView){  *//定義位於user\_account\_appearance.xml的物件*  val icon : ImageView = itemView.findViewById(R.id.*userList\_icon*)  val userName : TextView = itemView.findViewById(R.id.*userList\_userName*)  val userEmail : TextView = itemView.findViewById(R.id.*userList\_userEmail*)  val deleteBtn : ImageView = itemView.findViewById(R.id.*userList\_delete*)  }  override fun onBindViewHolder(holder: ViewHolder, position: Int) {  *//以position來逐一顯示資料*  val currentItem = newUserList[position]  holder.userName.*text* = currentItem.name  holder.userEmail.*text* = currentItem.email    holder.icon.setOnClickListener**{**  AlertDialog.Builder(holder.icon.*context*)  .setTitle("Login")  .setMessage("Login as this user?")  .setNegativeButton("Yes")**{**dialog, which **->**  *//傳送該使用者的名稱回SecActivity*  **}**  .setNeutralButton("No")**{**dialog, which **-> }**  .show()  **}**    holder.deleteBtn.setOnClickListener**{**  AlertDialog.Builder(holder.deleteBtn.*context*)  .setTitle("Delete the user")  .setMessage("Do you really want to delete this user?")  .setNegativeButton("Yes")**{**dialog, which **->**  newUserList.removeAt(position)  this.notifyDataSetChanged()  **}**  .setNeutralButton("No")**{**dialog, which **-> }**  .show()  **}** | | | |

[▼ 表 9-2-6 程式](#_heading=h.1pxezwc)規格描述-AppUserRegister.kt

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 5-5 | 程式名稱 | AppUserRegister.kt |
| 目的 | 提供註冊功能，並傳送用戶註冊後的資料至資料庫 | | |
| 部分程式碼 | | | |
| *//設定按下按鈕後會觸發的動作*  registerBtn.setOnClickListener **{**  *//檢查輸入資訊是否正確及適當*  if(userName.*text*.toString() == ""){  warningMessage.*text* = "Please Enter your name!"  }  else if(userEmail.*text*.toString() == ""){  warningMessage.*text* = "Please Enter your E-mail!"  }  else if(userPassword.*text*.toString() == ""){  warningMessage.*text* = "Please Enter your Password!"  }  else if(checkPassword.*text*.toString() == ""){  warningMessage.*text* = "Please Enter your Password again!"  }  else if(userPassword.*text*.toString() != checkPassword.*text*.toString()){  warningMessage.*text* = "Please Enter the same password twice!"  }  else if(userPassword.*text*.toString().length < 6){  warningMessage.*text* = "Your password was too short!"  }  else if(userPassword.*text*.toString().length > 16){  warningMessage.*text* = "Your password was too long!"  }  else{  intentBackToSec.putExtra("user\_Name", userName.*text*.toString())  setResult(*RESULT\_OK*, intentBackToSec)  finish()  }  **}** | | | |

**第10章 測試模型**

以使用者的視角去做APP測試，主要的測試內容為:

1.是否能夠正常註冊使用者資料

2.是否可以正常登入

3.資料輸入錯誤時會跳出警示視窗

4.登入後能夠進入各使用者的介面進行智慧鎖的操作

**第11章 操作手冊**

手機輸入特定網址下載APP，確認硬體模組的燈號是否亮起，如有亮起即可開始此用APP。

**第12章 使用手冊**

進入使用者介面，選擇登入或是註冊，如選擇註冊則輸入使用者名稱、信箱、密碼來完成註冊。如有帳號則選擇登入，並輸入使用者名稱及密碼進行登入，如完成則回到主介面並顯示歡迎資訊。

**第13章 感想**

**10946019 吳貫瑋**

通過這次的專題，我了解到組員之間的相互配合是一件重要的事情，組員之中有能力很強的也會幫忙其他組員，也有組員是較為散漫的，對於問題不會積極的去處理，也沒有回報，需要當面地去問才了解狀況，而我這次當了組長，在前半時期並沒有把這項問題處理好，直到後半時期才漸漸地有把情況改善，我也把這次的經驗當作一次教訓，以後在遇到類似的情形時就可以盡快的採取適當措施。再來，專題中會遇到以前沒有學過的東西，那就需要自己上網去查或是找對那方面比較有研究的同學或是朋友請教並學習，才能把問題解決。最後，我也在做專題期間體會到自己在策畫以及分配作業事項這方面較不擅長，需要再多磨練這方面，之後就可以更順利的把問題處理完。

我認為專題製作是一項可以磨練自己並確認自身有哪些不足的課程，並能在其中預先為出社會去工作時來做好準備，因為組員間的合作、策畫或是分配作業等事項都是在未來經常會遇到的事情，需要多多練習，才能夠熟練並準確完成，把事情做到最好。

**10946020 范紘齊**

進行了一年的專題，我個人最大的收穫是學習如何查資料自學，因為網路上的資源非常多，而例如為了要解決程式運作時的一個bug，雖然一查資料就很有可能解決，但在剛開始學習開發的時候也會因為網路上過多的資訊而導致不知道該相信哪種資訊。不過一當開發久了之後，在針對哪種問題查哪部分的資料這方面，我的思考能力有感覺越來越清晰，同時因為大多數的資料都是英文，也能增進我的英文閱讀能力。未來很多事情都得靠自己查閱自學，所以我也認為不斷地自學，吸收新資訊開發程式是一件既新鮮、有挑戰性，同時也是有著許多價值的事情。

而在進行這次專題的過程中，我也逐漸領悟到，進行專題製作時的態度就是未來在職場工作時的情況縮影，總是靠著被動接受事情的態度是無法在職場上好好生存的，所以我一直是主動去找事情做，主動跟其他成員討論並主動報告工作進度，盡可能讓所有人都清楚我負責的工作內容，就是不希望我們團隊一直處於後知後覺的狀態。

無奈中途有組員不負責任，一聲不說就失去聯絡，他負責開發的檔案我不斷提醒說要上傳到github，結果還是沒有上傳；還有拖延了好幾個月後還沒有任何進度的App，讓我逼不得已需要在僅剩兩三個月的時間內接手開發；加上不明確且散漫的分工跟進度管理，更讓我確信了如果想要把事情做好，自己必須要堅持主動思考跟積極進取的態度，並且清楚掌握自己的工作內容，同時把份內的工作想盡辦法做好，不能只是永遠等待著別人會來提出想法跟解決原本是自己負責的事情。即"堅持積極地做好自己，而不是消極地指望他人"。

**10946023 王信翰**

**10946042 林旻賢**

這一年的專題經驗使我深刻體會到團隊協作的重要性。我認識到，良好的配合和有效的溝通是確保專題順利進行的關鍵。與組員的互動讓我更加理解，不同角度和專業背景的觀點如何相互補充，促進專題發展的多元性，在分工合作的過程中，我學到了如何善用每個人的專長，以提高整體效能。在專題的進行中，我發現有些知識超出了課堂所學，因此我積極主動地上網搜尋相關資料，進行自主學習。同時，我也學會了善用團隊內外的資源，向同學、朋友或老師請教，以確保我們的專題能夠得到更全面的支持。這次專題經驗使我認識到自己的不足，並激發了我更積極主動地學習的意願，即便自己知道無法完全跟上某些方面，但可以透過與團隊的合作去幫助其他組員分擔一些自己可以完成的範圍，或是幫助其一起思考或尋找答案。這段經歷讓我對團隊協作、自主學習和積極參與工作有了更實際的認識，這對未來的學習和工作都具有實際的幫助。

**第14章 參考資料**

<https://www.programmingboss.com/2021/04/esp32-arduino-serial-communication-with-code.html#gsc.tab=0>

<https://stackoverflow.com/questions/60171641/any-solution-available-for-for-esp32-cam-brownout-detector-was-triggered-error>

<https://www.houseol.com.tw/service/news_view/6159.html>

<https://atceiling.blogspot.com/2021/06/android23-adapter.html>

<https://www.programmingboss.com/2021/04/esp32-arduino-serial-communication-with-code.html#gsc.tab=0>