# 保留基板與出貨別紙管理

# 系統需求

### 前台功能規劃:

需求者可以新增、修改和維護作業進度與處理者備註。

## 後台功能規劃:

資料建立後,僅同部門同事與建立者可上傳資料、修改內容。

資料建立後,僅建立者可刪除整筆內容。

處理者只能針對備註內容進行新增,並修改作業進度。

#### Client / Browser

- HTML \ CSS
- JavaScript (jQuery/ES6)
- Ajax

### Web Server (IIS)

- FastCGI Module
- wfastcgi.py觸發python

### Backend (Flask)

- Router
- REST / JSON API
- 公司login API

#### Database

MySQL

jQuery

lottie.js (loading動畫)

DataTables .js (增強HTML表格,提供排序、過濾、翻頁和其他功能)

SweetAlert2.js (美觀的彈出提示)

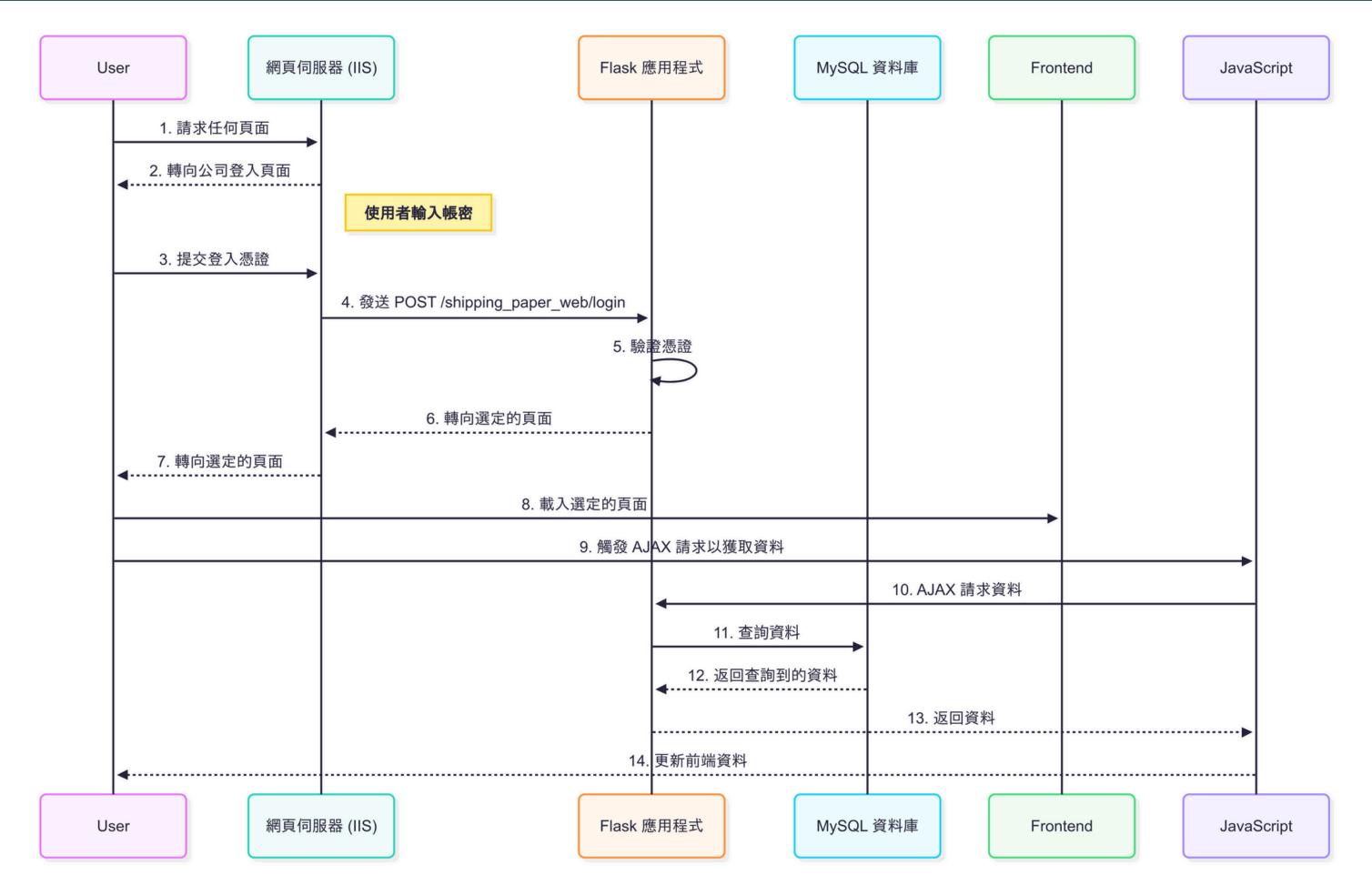
Air Datepicker (輕量級的日期選擇套件)

Bootstrap (前端框架)

Font Awesome (圖標字型)

SCSS

# 序列圖-登入



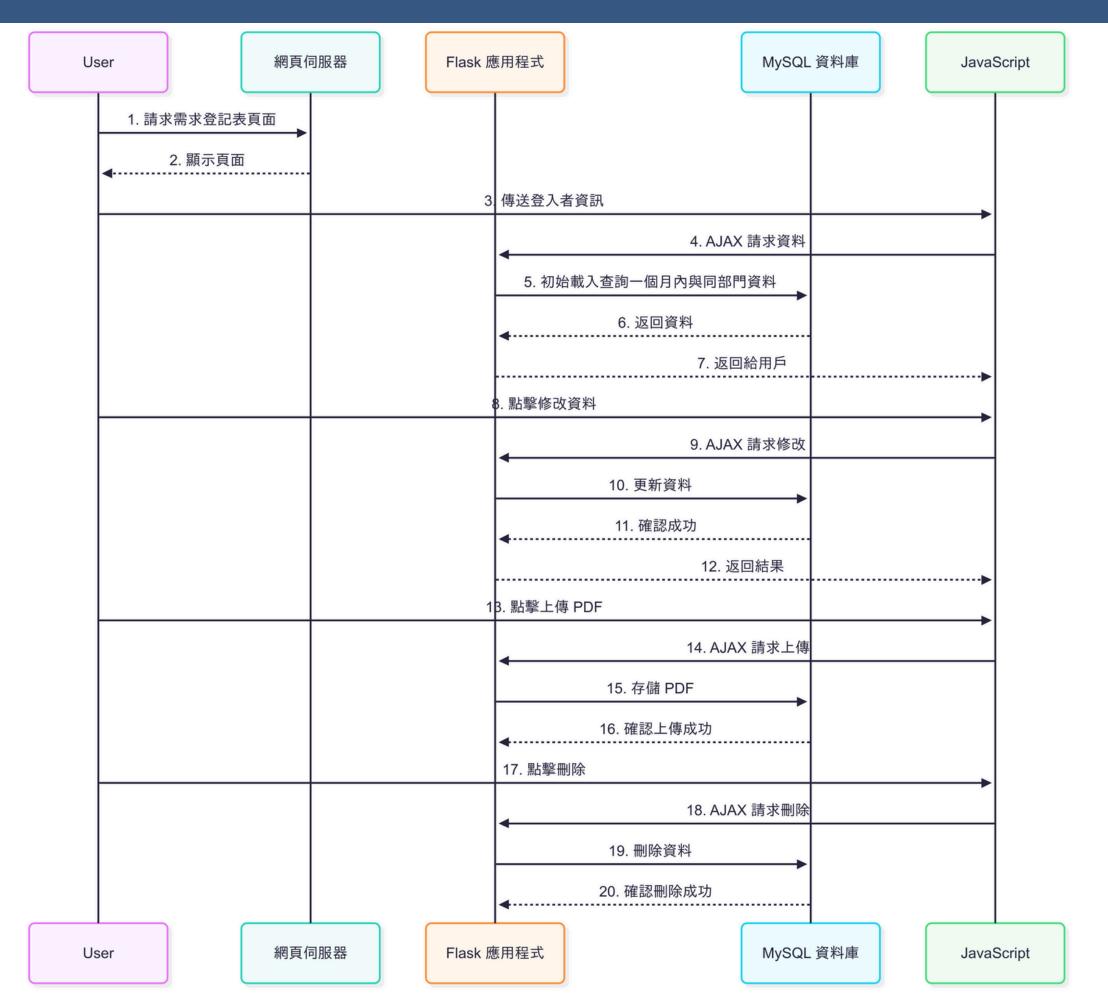
sequenceDiagram participant User participant WebServer as Web Server (IIS) participant FlaskApp as Flask Application participant MySQLDB as MySQL Database

User.>>WebServer: 1. Request any page
WebServer.>>User: 2. Redirect to company login page
User.>>WebServer: 3. Submit login credentials
WebServer: 3. Submit login credentials
WebServer.>>FlaskApp: 4. POST /shipping\_paper\_web/login
FlaskApp.>>>PlaskApp: 5. Validate credentials
FlaskApp.>>>WebServer: 6. Redirect to selected page
WebServer.>>Jesr: 7. Redirect to selected page
User.>>JavaScript: 9. Trigger AJAX request for data
JavaScript:>>FlaskApp: 10. AJAX request for data
MySQLDB->>FlaskApp: 11. Query for data
MySQLDB->>FlaskApp: 12. Return queried data
FlaskApp->>JavaScript: 3. Return data
JavaScript:>>>Juser: 14. Update front-end with data

sequenceDiagram participant User participant WebServer as 網頁伺服器 (IIS) participant FlaskApp as Flask 應用程式 participant MySQLDB as MySQL 資料庫

User>>WebServer: 1. 請求任何頁面
WebServer->>User: 2. 轉向公司登入頁面
User>>WebServer: 3. 提交登入憑證
WebServer->>FlaskApp: 4. 發送 POST /shipping\_paper\_web/login
FlaskApp->>FlaskApp: 5. 驗證憑證
FlaskApp->>WebServer: 6. 轉向選定的頁面
WebServer->>User: 7. 轉向選定的頁面
User>>>Frontend: 8. 載入選定的頁面
User>>>Frontend: 8. 載入選定的頁面
User>>>Frontend: 8. 載入選定的頁面
FlaskApp->>MySQLDB: 11. 查詢資料
FlaskApp->>MySQLDB: 11. 查詢資料
MySQLDB->>FlaskApp: 12. 返回查詢到的資料
FlaskApp->>JavaScript: 13. 返回資料
JavaScript->>FlaskApp: 12. 返回資料
JavaScript->>User: 14. 更新的端資料

# 序列圖-需求登記表



config: theme: redux-color

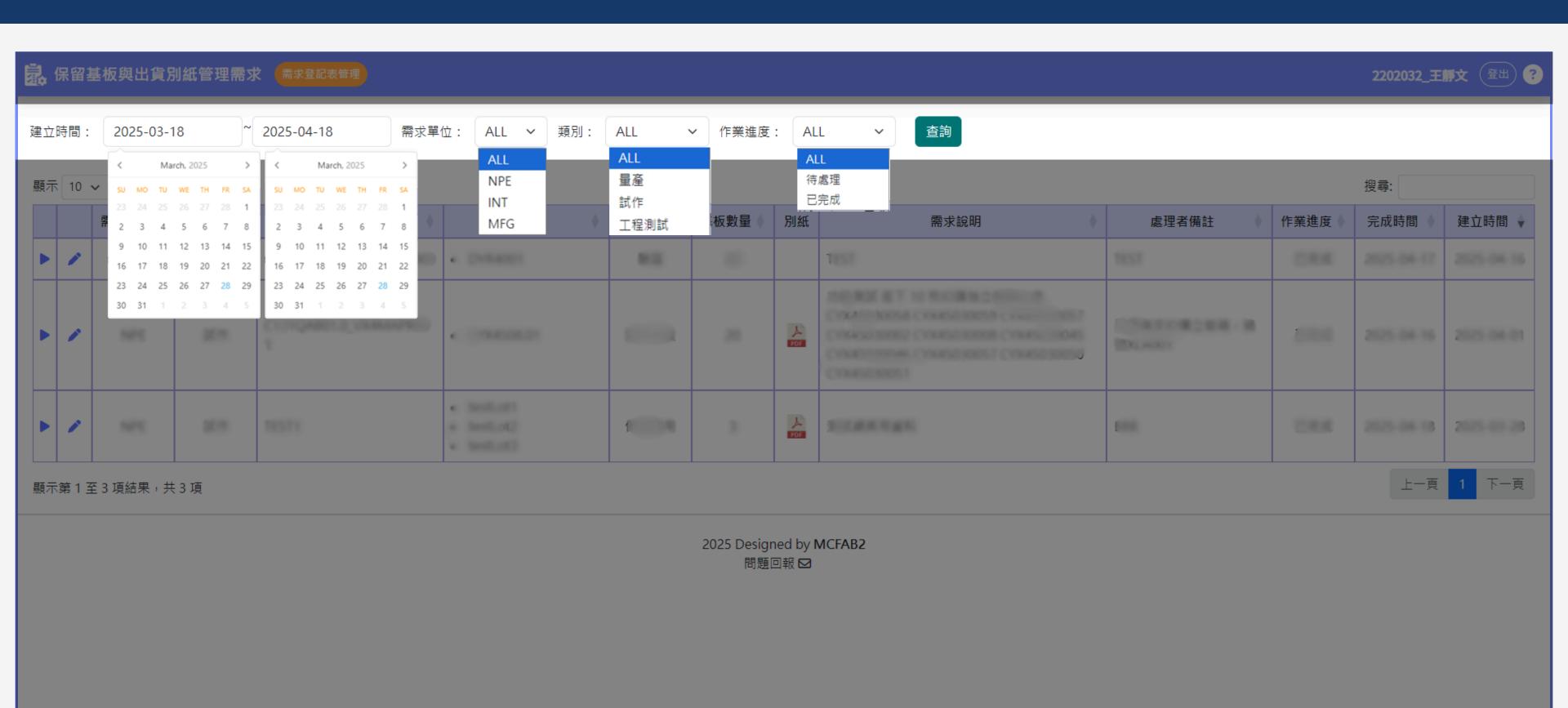
sequenceDiagram participant User participant WebServer as 網頁伺服器 participant FlaskApp as Flask 應用程式 participant MySQLDB as MySQL 資料庫 User->>WebServer: 1. 請求需求登記表頁面 WebServer-->>User: 2. 顯示頁面 User->>JavaScript: 3. 傳送登入者資訊 JavaScript->>FlaskApp: 4. AJAX 請求資料 FlaskApp->>MySQLDB: 5. 查詢同部門資料 MySQLDB-->>FlaskApp: 6. 返回資料 FlaskApp-->>JavaScript: 7. 返回給用戶 User->>JavaScript: 8. 點擊修改資料 JavaScript->>FlaskApp: 9. AJAX 請求修改 FlaskApp->>MySQLDB: 10. 更新資料 MySQLDB-->>FlaskApp: 11. 確認成功 FlaskApp-->>JavaScript: 12. 返回結果 User->>JavaScript: 13. 點擊上傳 PDF JavaScript->>FlaskApp: 14. AJAX 請求上傳 FlaskApp->>MySQLDB: 15. 存储 PDF MySQLDB-->>FlaskApp: 16. 確認上傳成功 User->>JavaScript: 17. 點擊刪除 JavaScript->>FlaskApp: 18. AJAX 請求刪除 FlaskApp->>MySQLDB: 19. 刪除資料

MySQLDB-->>FlaskApp: 20. 確認刪除成功

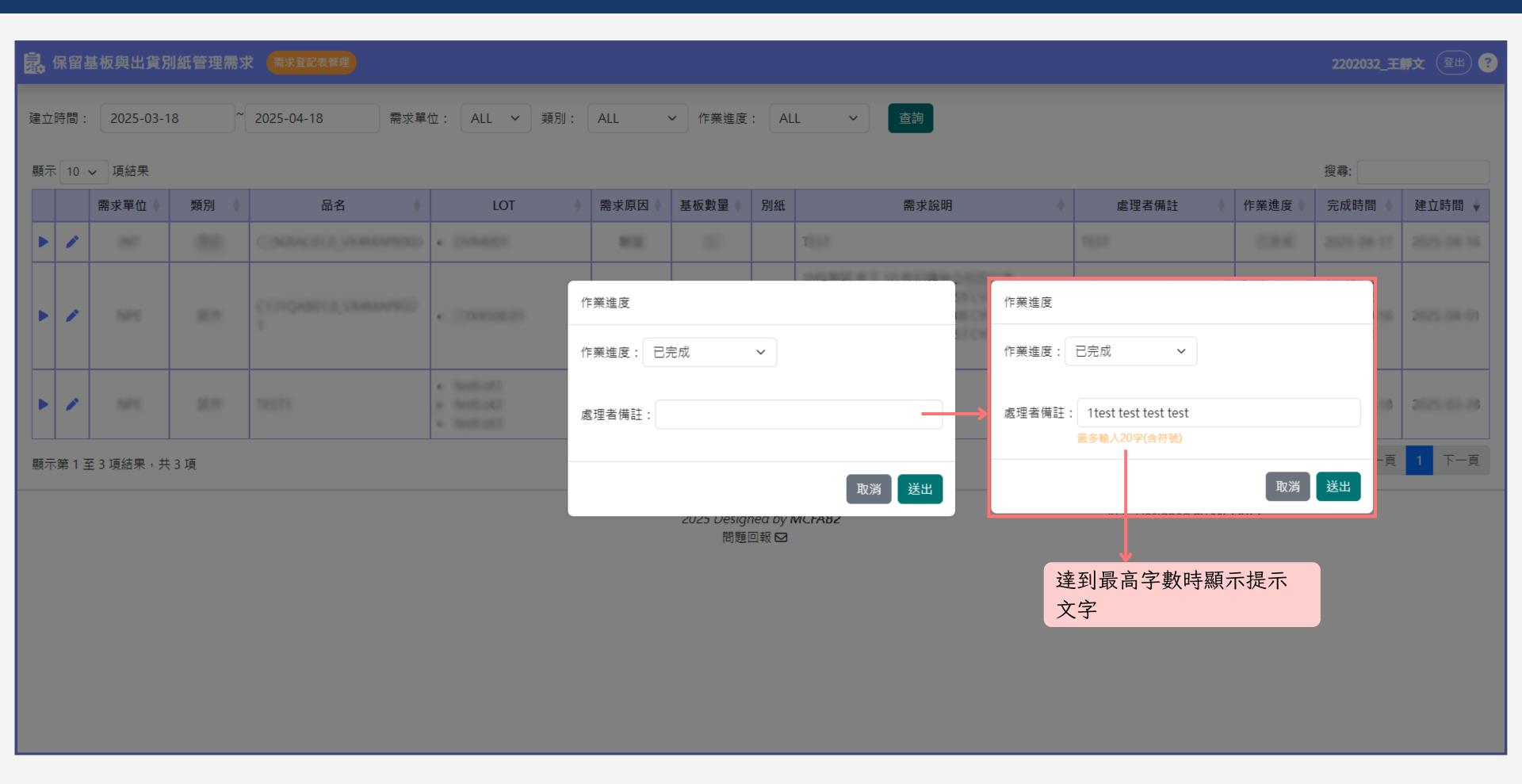
# 首頁登入後默認顯示前一個月登記表



## 查詢



# 更新作業進度



# 需求登記表管理



# 需求登記表管理權限限制



# 上傳別紙

