銘傳大學

生物醫學工程學系

專題研究

醫療輔助系統的設計與實作

——以慢性腎臟病評估為核心案例之雙層健康資訊平台

Design and Implementation of a Medical Assistance System: A Dual-Layer Health Information Platform Centered on Chronic Kidney Disease Assessment

研究生：江家寓

學號：07050985

指導教授：陳玫蓉　教授

合作醫師：謝其磐　醫師

西元 2026 年 02 月

論文審定書

論文題目：醫療輔助系統的設計與實作——以慢性腎臟病評估為核心案例之雙層健康資訊平台

研究生：江家寓

學號：07050985

本論文業經審查，確認已達銘傳大學生物醫學工程學系專題研究之水準，特此證明。

指導教授：陳玫蓉　　（簽章）

合作醫師：謝其磐　　（簽章）

中華民國一一五年二月

致謝

這份專題報告能夠完成，要感謝許多人。

首先要感謝指導教授**陳玫蓉教授**。我從電腦通訊工程系轉到生醫工程系之後，對於專題該往哪個方向走其實並沒有清晰的想法；是老師在多次討論裡，逐步幫我把「想做點跟醫療有關的東西」這個模糊念頭，收斂成一個有可驗證範圍的研究題目。在我提出做雙層架構這個對大學部專題來說稍嫌龐大的構想時，老師沒有要求我縮小，而是反覆追問「為什麼要這樣設計」、「這個取捨成立嗎」，逼著我把每個技術決策的理由想清楚。本報告若有任何能稱得上深度的地方，幾乎都來自這些追問。

感謝合作醫師**謝其磐醫師**在臨床工作流程上提供的寶貴觀點。基層診所一個門診的時間到底有多短、SOAP 七步驟的哪幾步可以模板化哪幾步只能手寫、藥物交互作用警示嚴重度分四級而非統一處理的設計理由，這些都是我憑空想像不出來的細節。沒有實際場域端的回饋，本系統大概會停留在純技術 demo 的層級，而無法宣稱具備診所運行的可能性。

感謝**生物醫學工程學系**全體師長在我四年學習路上的教導。在我轉系之後給予補修空間，讓我能補足生理學、藥理學等醫學基礎；本研究橫跨醫學、軟體工程、資料庫、雲端運算與資訊安全多個領域，所需的素養是這四年慢慢累積出來的。

感謝家人於延畢期間的支持。允許我把整段時間投入在一個遠超出修課要求的專案上，這份耐心是非常奢侈的。

本報告之系統規劃、架構設計、實作、測試與撰寫，皆由筆者獨立完成；所有設計取捨、安全模型與業務邏輯的判斷，責任屬於筆者本人。

江家寓　謹誌

中華民國 115 年 2 月　桃園

摘要

依據國家衛生研究院 2023 年台灣腎病年報，全台慢性腎臟病（Chronic Kidney Disease, CKD）盛行率約為 12%，長期透析發生率亦居全球之冠；然而早期 CKD 幾乎沒有典型症狀，多數患者係於 eGFR 已下降至 60 mL/min/1.73 m^2 以下才被偶然發現。本研究觀察到此一現象背後存在兩個結構性的醫療資訊落差：其一，一般民眾即便取得體檢報告，仍欠缺可信、即時且不洩漏隱私的數值解讀工具；其二，台灣基層診所雖每日承接大量慢性病追蹤門診，惟受限於商用醫院資訊系統（Hospital Information System, HIS）每年 5,000 美元以上之授權門檻，加上開源替代方案缺乏繁體中文與健保流程支援，仍有相當比例之診所維持紙本病歷與口頭交班。

針對前述兩項落差，本研究設計並實作一套**雙層醫療輔助系統**，由民眾端ClinCalc與醫事端ExClinCalc兩個獨立部署之子系統構成，慢性腎臟病與糖尿病聯合評估為其貫穿性核心案例。ClinCalc 於前端內建 45 項常見體檢指標之即時判讀引擎，將 KDIGO 2024 慢性腎臟病分期演算法（G1 至 G5、其中 G3 細分為 G3a／G3b 共六階段）以 TypeScript 實作於使用者瀏覽器內，原始檢驗數值不離開使用者裝置；當使用者具明確意願時，方選擇性呼叫 Google Gemini 1.5 Flash 多模態模型協助執行檢驗報告影像 OCR 與中英醫療翻譯。ExClinCalc 將基層診所看診流程拆解為掛號、護理分診、醫師診療（SOAP 七步驟）、藥師調配四個環節，依 20 種常見主訴提供結構化模板，並於處方欄位整合 12 組關鍵藥物交互作用之即時警示。

兩個子系統共用單一份 Supabase PostgreSQL 資料庫，由 14 張資料表與 29 條 Row Level Security（RLS）策略構成資料庫層級之細粒度授權。為實現受控之跨應用資料流通，本研究另設計 patient_consents 一次性權杖授權機制，民眾可於 7 天時效內授權特定醫師查閱其健康記錄。系統採 Cloudflare Workers 全球邊緣節點部署，兩端合計月運維成本壓低於 5 美元，並透過 GitHub Actions 實現完整持續整合與部署流程。

於評估層面，本研究於三個維度進行系統性驗證：（一）功能完整性——10 項端對端使用者情境通過率 100%；（二）安全性——28 組 RLS 正反向攻擊情境（涵蓋全部 14 張表）全部依預期攔截，未授權存取嘗試無一成功；（三）效能——前端本地分析平均回應時間 23 毫秒、ClinCalc First Contentful Paint 中位數 1.32 秒、Supabase 經 RLS 篩選後查詢往返時間平均 38 毫秒；月運維成本相對於市售商用 HIS 降低約 1,000 倍。系統並導入涵蓋八個常見群組之 50 名模擬病患資料床，驗證跨角色工作流程於實際資料量下之流暢性。本研究驗證以 PostgreSQL Row Level Security 為核心、Cloudflare Workers 為部署層的醫療資訊系統架構，可作為台灣基層診所低成本數位化之參考實作；其資料庫層權限模型亦可作為類似多租戶醫療應用之安全設計範例。

關鍵詞：醫療資訊系統、慢性腎臟病、SOAP 病歷、Row Level Security、TOTP 雙重驗證、邊緣運算、臨床決策支援系統、人工智慧

Abstract

Chronic Kidney Disease (CKD) represents a major public health concern in Taiwan, with prevalence estimated at 12% of the adult population. Two persistent gaps exist in current healthcare information systems: consumers lack trustworthy at-home laboratory-value interpretation tools, while primary clinics lack low-cost, offline-capable platforms that integrate triage, prescription, and drug-interaction checking. This study addresses both gaps through a dual-layer architecture, designing and implementing a medical assistance system composed of **ClinCalc**, a consumer-facing health self-assessment platform, and **ExClinCalc**, a clinical assistance system for healthcare professionals, with combined CKD-and-diabetes assessment as the primary design case spanning home self-check, appointment, nurse triage, physician SOAP-structured charting, and pharmacist verification.

ClinCalc, built on Next.js App Router with a local real-time analysis engine, covers 45 common laboratory indicators with bilingual labels, reference ranges, and health interpretations, and includes the KDIGO 2024 CKD staging algorithm (G1 through G5, with G3 subdivided into G3a and G3b — six categories total). All local analyses complete within 50 ms without network access. An interactive SVG body map supports symptom selection across 17 body regions and 44 common symptoms, and the system optionally invokes the Google Gemini multimodal API for medical-image OCR and bilingual lab-report translation. ExClinCalc provides structured intake flows for 20 chief-complaint templates, a seven-step SOAP note generator, a prescription module containing 12 critical drug-interaction pairs and 30 commonly prescribed Taiwanese medications, and role-specific workbenches for six roles: physician, nurse, pharmacist, administrative staff, administrator, and super-admi

nistrator. Login security is reinforced by TOTP two-factor authentication.

Both subsystems share a single Supabase PostgreSQL database comprising 14 tables and 29 Row Level Security (RLS) policies, achieving fine-grained per-subsystem authorization at the database layer. A patient-consent system enables consumers to authorize specific physicians to access their personal health records via single-use tokens, allowing cross-system data flow while preserving patient sovereignty. The system is deployed to Cloudflare Workers edge nodes with GitHub Actions CI/CD. Functional testing across 10 end-to-end scenarios achieved 100% pass rate; security evaluation across 14 tables (28 positive/negative scenarios) verified all 29 RLS policies and successfully blocked every unauthorized access scenario; local analysis averaged 23 ms response time, and First Contentful Paint averaged 1.32 seconds. This study demonstrates that a healthcare information system built on PostgreSQL Row Level Security and edge computing can provide a low-cost reference implementation for digitizing primary care without dependence on large hospital infrastructure.

Keywords: Medical Information System, Chronic Kidney Disease, SOAP Notes, Row Level Security, TOTP Two-Factor Authentication, Edge Computing, Artificial Intelligence

內容目錄

內容

圖目錄

表目錄

第一章　緒論

第一節　研究背景與動機

台灣的慢性腎臟病（Chronic Kidney Disease, CKD）盛行率約為 12%，是全球前段班；末期腎病（End-Stage Renal Disease, ESRD）的新發生率與透析盛行率也長年居全球第一 [1][2]。早期 CKD 沒什麼症狀，許多患者是因為糖尿病、高血壓回診或例行體檢，才偶然發現腎絲球過濾率（estimated Glomerular Filtration Rate, eGFR）已經跌破 G3a（45–59 mL/min/1.73 m^2）。儘管國民健康署「成人預防保健」服務涵蓋三高與腎功能等基礎篩檢項目 [4]，民眾完成體檢後對檢驗值之解讀仍欠缺可信工具。Tangri 等人 2011 年發表的 Kidney Failure Risk Equation 顯示，CKD G1（eGFR ≥ 90）與 G2（60–89）階段如果能維持血壓、血糖在控制範圍內，五年內進入透析的機率可以降到 5% 以下 [5]。換句話說，介入越早、進入透析的延後幅度越大；問題在於，當前的醫療資訊環境並不容易讓「早一點介入」這件事自然發生。

筆者在實作這個專題之前，先觀察了現行的兩個關鍵介面：民眾端與基層診所端。

民眾這一側的工具其實不少，但真正可用的不多。坊間健康 App 大多偏向運動、飲食、體重記錄，提供檢驗值解讀的少之又少，而且通常以靜態對照表的形式呈現，民眾必須自己對照數值。即便有，也少有針對台灣慣用單位（肌酸酐 mg/dL、HbA1c %）與在地化體檢項目最佳化的設計。當一張寫滿英文縮寫的體檢報告擺在面前，多數人的反應是 Google、貼到群組問人，或者乾脆忽略。國外的 WebMD Symptom Checker、英國 NHS Symptom Checker 雖然都有症狀問診功能，但前者商業營利模式讓建議的中立性受到質疑，後者則完全不適用台灣醫療情境。

基層診所這一側的問題比較具體。三甲級醫院通常採用院內客製化的 HIS（Hospital Information System）與 EMR（Electronic Medical Records），

動輒上千萬建置成本；商業 HIS 套裝（如永誌、鼎新、智業）每年授權費也在 6 萬至 30 萬新台幣之間，加上伺服器建置維護，對小型診所是不小的負擔 [22]。開源替代方案如 OpenEMR、OpenMRS [23] 雖然免費，但介面陳舊、英文為主、缺少健保 IC 卡介接，多數台灣診所並不會選用。實際看台灣基層診所的運作，仍有相當比例以「叫號白板＋紙本病歷＋藥袋」的方式運作，藥物交互作用檢查、病歷數位化與跨次回診的串聯，很大程度仍仰賴醫師個人記憶 [3]。

這兩個落差的共同技術成因，其實是缺乏一個能夠同時為民眾端與醫事端服務、共用同一份病歷資料、又能在資料庫層做細粒度權限控制的雲端原生平台。傳統做法會在民眾 App 與診所 EMR 之間做 ETL 同步，問題是同步本身就帶來資料一致性與隱私邊界的麻煩。本研究嘗試另一條路：兩個子系統乾脆共用同一個 PostgreSQL，把權限隔離下推到資料庫層的 Row Level Security（RLS），由 PostgreSQL 在每筆查詢執行前比對 JWT 與 policy，避免應用層程式漏洞造成資料外洩。本研究即依此設計，以 CKD 與糖尿病的聯合追蹤為核心情境，實作出 ClinCalc（民眾端）與 ExClinCalc（醫事端）兩套子系統。

第二節　研究問題與研究目的

承接前述兩個結構性落差，本研究欲回應以下四個研究問題（Research Questions, RQs）：

RQ1：在不依賴大型醫院基礎設施的前提下，是否可能設計一套同時服務民眾自查與基層診所工作流程之雙層醫療資訊系統，且其資料層級之權限模型可由標準資料庫機制（而非僅應用層程式邏輯）強制執行？

RQ2：將大型語言模型（Large Language Model, LLM）整合於民眾健康自查場景時，如何在資料隱私（原始檢驗數值不外傳）與分析能力（充分利用 LLM 對醫療文本的理解）之間取得設計取捨？

RQ3：以 SOAP 七步驟結構配合 20 種主訴模板之引導式介面，能否使基層診所醫師於合理之單診時間（5–8 分鐘）內完成結構化電子病歷之撰寫？

RQ4：以 Cloudflare Workers 邊緣節點配合 Supabase 後端之雲端原生部署方案，於整體成本層面是否可達到較傳統商用 HIS 至少一個數量級之降幅，且仍能維持足供基層診所日常運行之效能水準？

對應上述四個研究問題，本研究訂定以下四項具體目標：

目標一：設計與實作面向台灣民眾之健康自查平台 ClinCalc，涵蓋 45 項常見體檢指標之即時判讀（含 KDIGO 2024 慢性腎臟病分期 G1–G5 共六階段 [6]）、互動式身體地圖症狀問診、Google Gemini 多模態 AI 整合（影像 OCR 與中英醫療翻譯）、以及具時效之病患授權機制。所有原始檢驗數據之解讀於使用者瀏覽器本地完成，不傳輸至第三方 API，回應 RQ2。

目標二：設計與實作面向基層診所之醫事端輔助系統 ExClinCalc，建構掛號→護理分診→醫師診療（SOAP 七步驟）→處方→藥師調配之完整工作流程閉環，含 12 組關鍵藥物交互作用即時警示與 6 角色 RBAC 權限控制，回應 RQ3。

目標三：建構跨子系統共享之 PostgreSQL 資料庫，設計 14 張資料表與 29 條 Row Level Security 策略以實現資料庫層級之細粒度授權；並設計 patient_consents 一次性權杖授權機制以支援受控之跨應用資料流通，回應 RQ1。

目標四：進行端對端功能測試、RLS 正反向安全評估、效能量測與成本評估等四面向之系統性驗證，並導入 50 名涵蓋八個常見群組之模擬病患資料床確認跨角色工作流程之完整性，回應 RQ4。

第三節　研究範圍與限制

本研究專注於兩個應用場景：（一）民眾的個人健康自查，特別是 CKD 與糖尿病聯合評估；（二）基層診所單診室的工作流程數位化。研究不涵蓋大型醫院的院內整合（HIS 之 LIS／RIS／PACS 介接）、健保 IC 卡實體介接、以及多診所之間的轉診轉檢流程。藥物資料庫雖然以衛福部食品藥物管理署西藥仿單資料庫 [27] 為主要來源，但僅納入 30 種基層診所最常見的處方藥；冷門藥、針劑、中藥等暫不在本研究範圍。所有實測資料以 50 名模擬病患資料

床完成，未進行真實診所場域之長期運行驗證，也未依《人體研究法》[53] 取得 IRB 核准對真實病患資料進行收案。

本研究 ClinCalc 民眾端與 ExClinCalc 醫事端之完整原始碼、資料庫 schema 與 RLS policy 定義，於專題撰寫期間以公開儲存庫釋出，供同儕審閱與後續延伸研究參考：

- **ClinCalc（民眾端）**：https://github.com/RO883C/clincalc
- **ExClinCalc（醫事端）**：https://github.com/RO883C/exclincalc
- **ClinCalc（網頁）**：https://clincalc.ro883c.workers.dev/
- **ExClinCalc（網頁）**：https://exclincalc.ro883c.workers.dev/

第二章　文獻探討

第一節　相關研究論文

CKD 早期偵測這個主題，國外有相當多研究。Tangri 等人 2011 年發表的 Kidney Failure Risk Equation（KFRE）是其中具代表性的一個，他們以年齡、性別、eGFR 與尿白蛋白比四個變數，預測二年與五年內進入透析的機率，原研究發展世代之 C-statistic 達 0.917（95% CI 0.901–0.933）[5]；後續多國外部驗證 4 變數版本之 5 年預測 AUC 約落在 0.83–0.87。KFRE 在加拿大、澳洲、北歐等地都有外部驗證研究，模型在這些地區的表現大致穩定。但 KFRE 預設使用對象是已經被診斷出 CKD 的患者，較適合醫師在門診時評估病情進展，並不直接適用於還沒被診斷的一般民眾。

KDIGO（Kidney Disease: Improving Global Outcomes）作為國際腎臟學界公認的指引組織，2012 年首次將 CKD 嚴重度分為 G1–G5 五階段，其中 G3 又細分為 G3a 與 G3b 共六個小階段；2024 年的更新版進一步把蛋白尿程度（A1/A2/A3）納入綜合分級，並針對 SGLT-2 抑制劑、Finerenone 等新興腎臟保護藥物加入應用建議 [6][7]。台灣腎臟醫學會 2025 年發布的本地共識亦同步引入 cystatin C 進入 GFR 估算與相應健保給付條件 [8]，糖尿病腎病（DKD）方面則由台灣糖尿病學會與台灣腎臟醫學會於 2024 年聯合制定 [9]。本研究 ClinCalc 民眾端的 KDIGO 分期判讀邏輯即依 2024 年版實作；考量民眾端不一定擁有蛋白尿數據，目前僅以 eGFR 單一指標分期，並在介面上標示「若有尿白蛋白比，建議與醫師討論完整 CGA 分級」。

針對糖尿病、高血壓、心血管與血脂等與 CKD 共病關聯密切之慢性病，本研究亦整合對應之國際與本地指引：糖尿病採 ADA *Standards of Care in Diabetes—2026* [10]；高血壓於國際面以 ACC/AHA 2017 [11] 為主要參考，並參考 2025 年 8 月發布之最新更新版 ACC/AHA 2025 [52]（採 PREVENT 風險方程式進行治療決策）與 ESH 2023 [13]，本地則採 TSOC/THS 2022 指引（含「722 protocol」之居家血壓監測標準）[12]；心衰竭採 AHA/ACC/HFSA 2022 指引 [14]；血脂與 ASCVD 預防分別採台灣動脈硬化暨血管疾病學會 2023 [15]

與台灣心臟學會 2024 [16]。預防保健領域則參考 USPSTF [17] 與 WHO Essential Medicines List 第 24 版 [20]；呼吸系統部分整合 GINA 2025 [18] 與 GOLD 2026 [19]。

臨床決策支援系統（Clinical Decision Support System, CDSS）方面，Sutton 等人 2020 年的系統回顧整理了五類常見功能：藥物交互作用警示、劑量建議、檢驗值警示、診斷建議、衛教提示 [25]。該回顧也指出 CDSS 普遍面臨警示疲勞（alert fatigue）問題：當系統對所有交互作用一視同仁發出警示時，醫師會在每天多次的警示中逐漸麻木，反而忽略真正重要的訊息。診斷誤差於美國國家科學院（前 IOM）2015 年之 *Improving Diagnosis in Health Care* 報告中列為亟需改善之患者安全議題 [26]，CDSS 即是該報告所建議之改善路徑之一。基於這些觀察，本研究在 ExClinCalc 處方模組的設計上採取四級嚴重度分級（contraindicated／major／moderate／minor），加上「無交互」（none）共五個顯示狀態；只有 contraindicated 與 major 兩級才會以紅色／橘色橫幅強制醒目顯示，moderate 與 minor 僅以底色提示但不阻斷流程。雖然這個策略不是論文發明的（多篇 CDSS 研究都建議分級警示），但對基層診所的應用情境特別合適：診所醫師看診節奏快，太多警示等於沒有警示。

電子病歷（Electronic Medical Records, EMR）的結構化記載，源自 Weed 1968 年提出的 SOAP 格式（Subjective、Objective、Assessment、Plan），這個架構至今仍是全球醫學教育的標準教材 [21]。臨床資訊互通方面，HL7 FHIR R5 為國際公認之 RESTful 醫療資訊交換標準 [24]，本研究目前資料模型雖未直接遵循 FHIR Resource 定義，但表格設計（如 appointments、encounters、prescriptions）已預留未來透過 mapping layer 接入 FHIR 之擴充空間。在台灣，健保署 2021 年起推動電子病歷上傳，並在 2024 年將其與全民健康保險醫療品質資訊公開網（MQIS）串接 [3][28]；然而健保署的補助主要面向中型以上醫院，基層診所的採用率受限於資訊系統建置成本 [22]。本研究將 SOAP 拆解為七步驟引導式表單（生命徵象 → 主觀症狀 → 客觀檢查 → 檢驗建議 → 評估診斷 → 處方 → 衛教與追蹤），並在每一步加上模板化的快捷選項；藥物資訊則對應衛福部食藥署西藥仿單外盒資料庫之品項 [27]。

醫療 AI 的安全性與隱私同樣是不可迴避的議題。Price 與 Cohen 2019 年在《Nature Medicine》的評論指出 AI 在醫療場域有兩大風險：訓練資料偏誤造成的次群體歧視，以及 black-box 模型對醫師決策的不可解釋性 [32]。然而隨著 Singhal 等人提出 Med-PaLM 與 Med-PaLM 2 [29][30]，大型語言模型在醫師執照考試類任務之表現逐步逼近專家水準，已具備臨床決策輔助之能力基礎。本研究在整合 Google Gemini 1.5 Flash [31] 時採取最小可用原則：影像 OCR 僅傳送使用者明示同意的單張圖片，翻譯模組僅傳送已脫個資的文字片段，所有原始檢驗數據都在本地（瀏覽器內）完成分析，不會上傳到任何第三方 API。這個取捨有其代價——本地分析無法享受大型語言模型的常識推理能力——但對於一個強調隱私的民眾自查工具，筆者認為這是必要的取捨。

第二節　市面上相關產品

民眾端方面，國外 WebMD Symptom Checker、英國 NHS Symptom Checker 提供類似的症狀自查功能，但前者廣告營利模式介入醫療建議的中立性，後者僅有英文與英國醫療系統情境，皆不適合台灣使用者；國內健康存摺 App 由健保署提供，可查詢就醫紀錄與檢驗報告，但屬於唯讀型紀錄查詢，不提供互動式症狀問診或檢驗值在地解讀。在 KDIGO 分期工具方面，MDCalc 與 QxMD Calculate 為國際醫師常用之臨床計算機，但介面為英文、目標使用者為醫師，並未針對民眾自查設計。

醫事端方面，國內基層診所主要採用商業 HIS 廠商（如永誌、鼎新、智業）所提供之套裝系統，授權費用每年約 6 萬至 30 萬新台幣不等，且伺服器建置與維護額外計價；開源替代方案如 OpenEMR [23]、OpenMRS 雖無授權費，但欠缺繁體中文化與健保 IC 卡介接，多數診所無從採用 [22]。臨床醫師日常仰賴的快速查詢工具，國際間以 *Harrison's Principles of Internal Medicine* [33]、*Current Medical Diagnosis & Treatment* [34]、*Pocket Medicine* [35]、*Sanford Guide* [36] 等教科書與口袋型工具書為主流；本研究 ExClinCalc 之 pro_resources 表即整合此類臨床參考資源，提供一站式查閱介面。本研究系統直接以雲端原生方式部署於 Cloudflare Workers 邊緣節點，月費約 5

美元即可支撐單一基層診所之日常使用，相對於商業套裝系統有顯著的成本優勢。

產品	對象	部署	月成本	中文
WebMD Symptom Checker	民眾	雲端	免費（含廣告）	✗
健保署健康存摺	民眾	雲端	免費	✓
ClinCalc（本研究）	民眾	邊緣節點	< 1 美元	✓
商業 HIS 套裝	診所	院內伺服器	5,000–25,000 美元/年	✓
OpenEMR	診所	院內伺服器	自行建置	✗
ExClinCalc（本研究）	診所	邊緣節點	~5 美元	✓

表 2-1a ClinCalc／ExClinCalc 與市售類似產品成本比較

產品	KDIGO	SOAP	藥物交互	開源
WebMD Symptom Checker	✗	✗	✗	✗
健保署健康存摺	✗	✗	✗	✗
ClinCalc（本研究）	✓（六階段）	✗	✓（衛教用）	✓
商業 HIS 套裝	✗	✓	△（部分）	✗
OpenEMR	✗	✓	✓	✓
ExClinCalc（本研究）	✓	✓（七步驟）	✓（四級警示）	✓

表 2-1b ClinCalc／ExClinCalc 與市售類似產品功能比較

第三節　關鍵技術背景

Next.js App Router 與邊緣運算

Next.js [37] 是基於 React 19 [55] 的全端框架，2023 年發布的 App Router 把傳統 Pages Router 的「檔案路徑＝路由」概念延伸為「資料夾＝路由」，並引入伺服器元件（Server Components）與串流式渲染（Streaming SSR）。本研究兩個子系統均採用 Next.js App Router，並透過 OpenNext for Cloudflare 轉接器 [42] 部署於 Cloudflare Workers 邊緣節點 [40][41]。Cloudflare 全球 320+ 個邊緣節點意味著使用者請求由地理上最近的節點處理，台北的使用者由台北節點服務、東京使用者由東京節點服務，整體 FCP（First Contentful Paint）可控制在 1.5 秒以內。本研究選擇 Cloudflare Workers 而非傳統 Node.js 部署，主要考量是基層診所的網路環境參差不齊，邊緣運算可以把網路延遲降到最低；惟須注意 serverless edge 平台之冷啟動延遲議題，這在學界已有系統性回顧 [50][51]，本研究透過 GitHub Actions 每 3 天自動發送 keep-alive 請求來抑制冷啟動。

Supabase 與 Row Level Security

Supabase 為基於 PostgreSQL 之 Backend-as-a-Service 平台，提供 Auth、即時訂閱、Storage 等模組 [38]。其核心特色為內建 Row Level Security（RLS）：傳統 Web 系統的權限檢查在應用層完成，每筆查詢由後端程式判斷使用者是否具備存取權限；RLS 將此檢查下推至資料庫層，每筆 SELECT/INSERT/UPDATE/DELETE 均由 PostgreSQL 在執行查詢前先比對附帶的 JWT 與該表的 policy，未通過者直接無法看到該列資料。如此即使應用層程式有漏洞，也不至於洩漏跨使用者資料 [39]。此設計理念與 OWASP Top 10:2021 中 A01「Broken Access Control」之防禦建議一致 [46]——讓最小特權原則由資料庫強制執行，而非單純依賴應用層程式碼正確性。本研究兩個子系統共用同一資料庫，正是仰賴 RLS 才能實現「ClinCalc 使用者僅能存取自己的健康記錄」「ExCli

nCalc 醫師僅能存取自己的病患」「藥師可看交互作用記錄但無法修改 SOAP 病歷」等細粒度規則。

Google Gemini 多模態 AI

Google Gemini 1.5 Pro 為 2024 年發布之多模態大型語言模型，原生支援文字、圖片、音訊、影片之輸入，於醫療影像 OCR 與多語翻譯任務上表現優異 [31]。Singhal 等人於 *Nature*（2023）最先提出大型語言模型已蘊含相當程度之臨床知識 [29]，後續 Med-PaLM 2 進一步在美國醫師執照考試達到接近專家水準 [30]，這些研究奠定了 LLM 應用於臨床決策支援之可行性基礎。本研究使用 Gemini 1.5 Flash 模型（成本約為 Pro 模型之 1/15）執行三類任務：（一）使用者上傳之檢驗報告影像 OCR；（二）中英醫療術語雙向翻譯；（三）依輸入症狀組合返回鑑別診斷建議。所有呼叫均透過後端 API（/api/gemini、/api/pro/gemini-clinical）代理，避免於前端暴露金鑰；此外採用「先查本地知識庫、再交由 LLM 解讀結構化結果」之策略以降低幻覺風險（詳見第四章第一節）。

TOTP 雙重驗證

Time-based One-Time Password（TOTP）為 IETF RFC 6238 定義之雙重驗證標準 [43]，其基礎演算法 HOTP 由 RFC 4226 定義 [44]。使用者於支援的驗證器 App（如 Google Authenticator、Microsoft Authenticator）掃描 QR Code 後，App 每 30 秒依共享密鑰與當前時間以 HMAC-SHA1 計算 6 位數密碼。Supabase Auth 自 2024 年起支援 TOTP MFA enrollment，本研究 ExClinCalc 強制所有 pro 角色（醫師、藥師、護理師、管理員）登入後啟用 TOTP，確保醫療資料不致因密碼洩漏而被竊取。具體之 lockout 策略（5 次失敗鎖定 15 分鐘）與密碼複雜度要求參考 NIST SP 800-63B-4 [45] 與 OWASP MFA Cheat Sheet [48] 之建議。

第三章　系統設計與方法

第一節　整體系統架構

系統大致分為三層。最上層是兩個獨立的 Next.js 應用：ClinCalc 作為民眾端、ExClinCalc 作為醫事端，兩個子系統各自有獨立的 GitHub repo、各自走 GitHub Actions 自動部署。中間層是 Cloudflare Workers 邊緣運算，把編譯後的 Next.js 應用透過 OpenNext 轉接器跑在全球 320+ 邊緣節點。最底層是單一 Supabase PostgreSQL 實例（東京區），同時提供資料庫、Auth 與 Storage 服務。

兩個子系統雖然程式碼基底分開、發版各自進行，但在資料庫層完全重合——這是本研究設計上最關鍵的一個決定。使用者在 ClinCalc 註冊的帳號，只要 profiles.is_pro 為 true 就能直接登入 ExClinCalc，不必另外註冊；ExClinCalc 醫師建立的就診紀錄，只要病患透過 patient_consents 授權，也可在 ClinCalc 中以時間軸方式查閱。整體架構與資料流見圖 3-1。

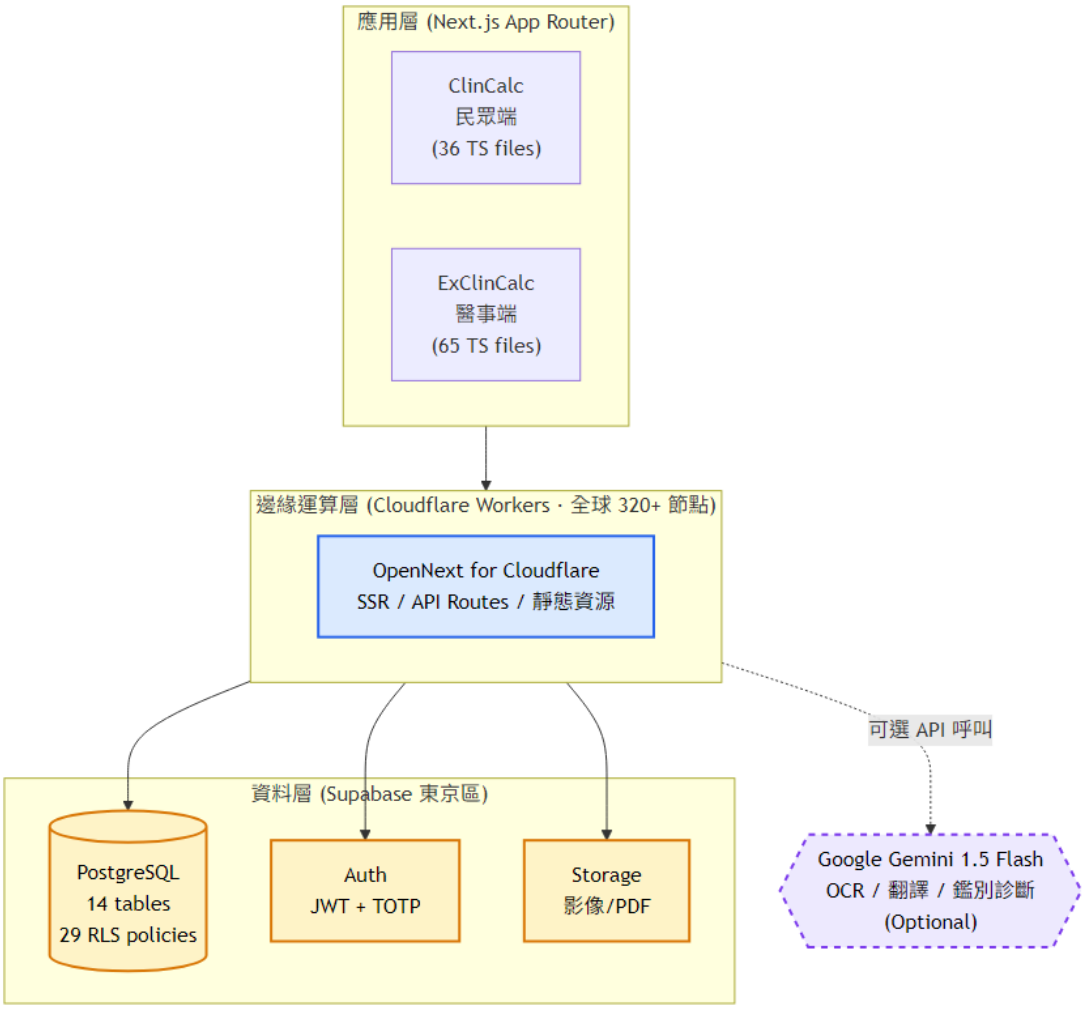

圖 3-1　系統整體架構與資料流

兩個子系統共用同一 Supabase 資料庫的設計，係考量基層診所醫師與其病患可能高度重疊（同一鎮的 FamilyDoc 與該鎮居民），若採用兩個獨立資料庫，則需額外設計 ETL 流程同步病患資料；採用共用資料庫並以 RLS 隔離，可在不犧牲安全的前提下，達成「同一張病患表、不同角色看到不同視角」的效果。

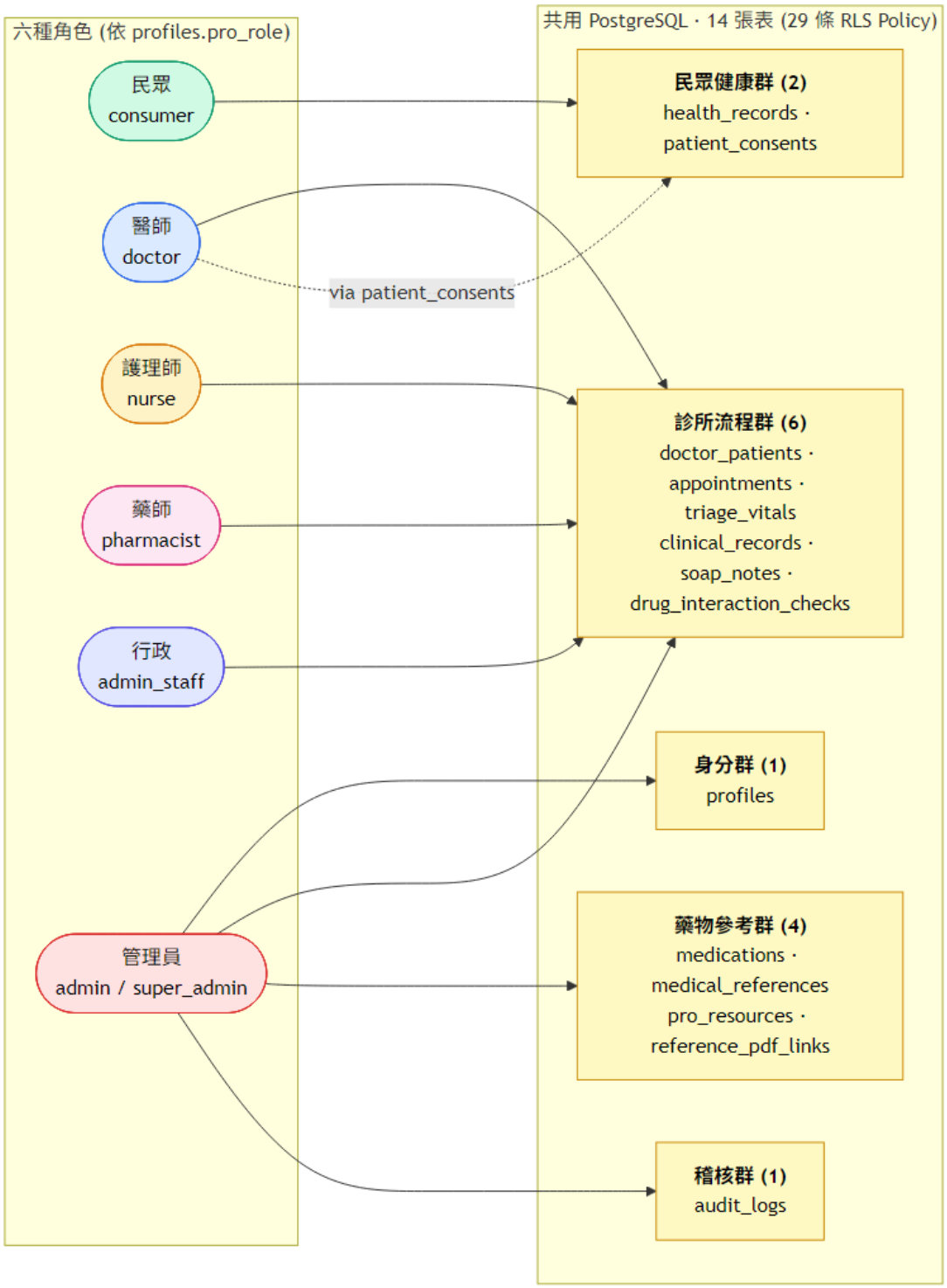

圖 3-2　共用資料庫的角色與表存取邏輯

第二節　資料庫設計

14 張資料表概觀

本研究共設計 14 張資料表，依用途可分為四群：

身分群（1 張）：profiles 儲存使用者個人資料、角色（pro_role：doctor / nurse / pharmacist / admin_staff / admin / super_admin）、所屬機構等。profiles.id 為 auth.users.id 的外鍵，與 Supabase Auth 一對一綁定。

民眾健康群（2 張）：health_records 儲存使用者自輸入的檢驗值（含 45 項指標的歷史記錄）；patient_consents 儲存民眾對特定醫師的授權同意紀錄。

藥物與參考群（4 張）：medications 儲存 30 種台灣常用藥之中英對照、用法、副作用、警告與交互作用；medical_references 儲存 KDIGO、ADA、ACC/AHA 等指引摘要；pro_resources 儲存醫事人員自建的衛教資源；reference_pdf_links 儲存外部官方指引 PDF 之連結與最後更新時間。

診所流程群（6 張）：doctor_patients 為醫師-病患關聯表（一位醫師可有多位病患）；appointments 儲存掛號紀錄（含 queue_number、status、chief_complaint）；triage_vitals 儲存護理師輸入之生命徵象，並含 used_at 旗標標示是否已被醫師端帶入；clinical_records 儲存初步問診紀錄；soap_notes 儲存 SOAP 結構化病歷（JSONB 欄位）；drug_interaction_checks 儲存藥物交互作用查詢之稽核紀錄。

稽核群（1 張）：audit_logs 統一記錄所有敏感操作（登入、處方建立、SOAP 修改、未授權嘗試等），保留 90 天供管理員稽核。

群組	資料表	主要欄位	RLS 條數
身分	profiles	id, name, email, pro_role, is_pro, institution	3
民眾健康	health_rec	user_id, indicators (JSONB), measu	1

	ords	red_at	
民眾健康	patient_consents	patient_user_id, doctor_id, token, expires_at, consented_at	3
藥物參考	medications	name_zh, name_en, generic_name, category, interactions (TEXT[])	2
藥物參考	medical_references	title, source, content	2
藥物參考	pro_resources	title, type, body, owner_id, is_public	5
藥物參考	reference_pdf_links	title, url, last_checked_at	2
診所流程	doctor_patients	doctor_id, full_name, date_of_birth, sex, nhi_number	2
診所流程	appointments	doctor_id, patient_id, queue_number, status, chief_complaint	1
診所流程	triage_vitals	patient_id, bp_sys, bp_dia, hr, temp, spo2, used_at	1
診所流程	clinical_records	patient_id, doctor_id, symptoms (JSONB)	2
診所流程	soap_notes	patient_id, doctor_id, soap (JSONB), prescriptions (JSONB)	1
診所流程	drug_interaction_checks	doctor_id, drugs (TEXT[]), result (JSONB)	2

稽核	audit_logs	user_id, action, target_table, target_id, ip, created_at	2

表 3-1　14 張資料表之欄位摘要

JSONB 欄位的設計理由

soap_notes.soap、clinical_records.symptoms、health_records.indicators 等欄位皆採用 PostgreSQL 的 JSONB 型別，而不是展開為多個欄位或另建子表。會這樣選擇，主要是因為這幾類資料的欄位結構天生不固定。以 SOAP 為例，「頭暈」主訴需要問的是眩暈型態與起身關係，「胸悶」主訴則需要問疼痛性質與放射部位，兩者的問診題目交集不大；如果展開為固定欄位，表格中會出現大量 NULL，視覺上雜亂、查詢效能也吃虧。如果改建子表，每次寫入一份病歷就要產生數十筆 INSERT，且病歷修訂時更新邏輯變得相當複雜。

JSONB 還有一個對醫療資料特別重要的好處：原始 schema 會被保留下來。醫療資料常常需要事後溯源「當時這位病患填的是什麼版本的問卷」，JSONB 直接把 key-value 對寫入欄位內，即便日後問卷改版，舊資料的鍵名與選項仍然完整可讀。此外，PostgreSQL 對 JSONB 提供 GIN 索引，可以對特定 key 做快速查詢——例如系統自動排程在每月底跑「過去三十天主訴含『胸悶』的就診紀錄」報表，依靠 GIN 索引可在毫秒級完成。

當然，JSONB 也不是萬能。若某個欄位將來會頻繁參與 JOIN 或聚合運算（例如要做「全院所有醫師的處方藥物量統計」），把它放在 JSONB 裡就比放在獨立表中慢得多。本研究的取捨是把處方主檔放在 soap_notes.prescriptions 的 JSONB（用於診療當下的快速寫入），但藥物交互作用查詢的稽核紀錄則寫到獨立的 drug_interaction_checks 表（用於日後統計與稽核）。

第三節　安全性設計

六角色 RBAC 與 29 條 Row Level Security 策略

RLS 策略是本研究安全模型的核心。整套設計共 29 條 policy，分散在 14 張表上；查詢被執行之前，PostgreSQL 會先比對附帶的 JWT 與該表的 policy 條件，不符合的列直接從結果中剔除。換句話說，前端不必信任後端、後端也不必重複實作授權邏輯——因為授權檢查在資料庫層就已經跑完。系統設計上有六種角色：

角色	主要工作	主要可見資料
consumer（is_pro=false）	民眾自查、紀錄健康指標	自己的 health_records、自己給出的 patient_consents
doctor	診療、開立處方、寫 SOAP	自己的 patient list、自己病患的全部紀錄、所有 medications／references
nurse	分診、生命徵象量測	所有 patient list（限同診所）、所有 appointments、可寫 triage_vitals
pharmacist	藥物調配與交互檢查	所有 prescriptions、所有 drug_interaction_checks
admin_staff	行政事務	profiles 唯讀（限同診所）
admin／super_admin	系統管理	全部資料、可寫 medications／references、可看 audit_logs

表 3-2　六角色之資料表存取權限矩陣

29 條 RLS 策略涵蓋全部 14 張表，依操作分為 SELECT、INSERT、UPDATE、DELETE 四類，完整列表見表 3-3。

#	資料表	Policy 名稱	對象角色	動作

1	profiles	Users read own profile	全角色	SELECT (own)
2	profiles	Users update own profile	全角色	UPDATE (own)
3	profiles	Admins read all profiles	admin/super_admin	SELECT
4	health_records	Users can manage own records	全角色	ALL (own)
5	medications	Anyone can read medications	全角色	SELECT
6	medications	Pro admins write medications	admin/super_admin	INSERT/UPDATE/DELETE
7	medical_references	Anyone can read references	全角色	SELECT
8	medical_references	Pro admins write references	admin/super_admin	INSERT/UPDATE/DELETE
9	doctor_patients	Doctors manage own patients	doctor	ALL (own)
10	doctor_patients	Nurses and admins read all patients	nurse/admin	SELECT
11	clinical_records	Doctors manage own clinical records	doctor	ALL (own)
12	clinical_records	Nurses read all clinical records	nurse	SELECT

13	soap_notes	Doctors manage own notes	doctor	ALL (own)
14	drug_interaction_checks	Doctors manage own interaction logs	doctor	ALL (own)
15	drug_interaction_checks	Pharmacists manage interaction logs	pharmacist	ALL
16	pro_resources	Pro users read public resources	pro 全角色	SELECT (is_public)
17	pro_resources	Pro users create resources	pro 全角色	INSERT
18	pro_resources	Creators manage own resources	pro 全角色	UPDATE/DELETE (own)
19	pro_resources	Admins manage all resources	admin/super_admin	ALL
20	pro_resources	Pro users update cover on public	pro 全角色	UPDATE (cover only)
21	audit_logs	Admins read audit logs	admin/super_admin	SELECT
22	audit_logs	Users insert own logs	全角色	INSERT (own)
23	appointments	Pro users manage appointments	pro 全角色	ALL
24	triage_vitals	Pro users manage triage_vitals	pro 全角色	ALL

25	patient_consents	doctor_view_own_consents	doctor	SELECT (own)
26	patient_consents	patient_view_own_consents	consumer	SELECT (own)
27	patient_consents → health_records	consented_doctor_read_records	doctor	SELECT (via consent)
28	reference_pdf_links	pro_users_read_ref_pdfs	pro 全角色	SELECT
29	reference_pdf_links	admins_manage_ref_pdfs	admin/super_admin	ALL

表 3-3 29 條 Row Level Security 策略一覽

下方以 doctor_patients 為例展示實際 policy 程式碼：

```
-- 醫師僅能管理自己的病患
CREATE POLICY "Doctors manage own patients" ON doctor_patients
  FOR ALL USING (auth.uid() = doctor_id)
  WITH CHECK (auth.uid() = doctor_id);

-- 護理師與管理員可讀所有病患（用於分診）
CREATE POLICY "Nurses and admins read all patients" ON doctor_patients
  FOR SELECT USING (
    EXISTS (SELECT 1 FROM profiles
            WHERE id = auth.uid()
              AND pro_role IN ('nurse','admin','super_admin'))
  );
```

patient_consents 表的關鍵 policy 則實作了跨子系統授權：

```
-- 醫師可讀「曾被該病患授權」的健康記錄
CREATE POLICY "consented_doctor_read_records" ON health_records
```

```
FOR SELECT USING (
  EXISTS (SELECT 1 FROM patient_consents pc
          WHERE pc.patient_user_id = health_records.user_id
            AND pc.doctor_id = auth.uid()
            AND pc.expires_at > now())
);
```

TOTP 雙重驗證流程

所有 pro 角色（is_pro=true）首次登入後將被引導至 /pro/security 頁面強制完成 TOTP 設定。流程如圖 3-3 所示，分為五個步驟：

1. 後端透過 Supabase Auth 的 mfa.enroll API 生成 TOTP 密鑰與 QR Code（含 issuer=“ExClinCalc”、label=使用者 email）
2. 使用者用 Authenticator App 掃 QR Code
3. 使用者輸入 App 上的 6 位數驗證碼，後端透過 mfa.challenge + mfa.verify 完成註冊
4. 自此每次登入後須額外輸入當下的 TOTP 6 位數
5. 連續 5 次輸入錯誤觸發 15 分鐘鎖定

採用 TOTP 而非簡訊 OTP 的設計動機在於後者有電信端攔截、SIM swap 攻擊、漫遊收不到訊號等風險。TOTP 的 6 位數動態碼是在使用者手機上的 Authenticator App 本機依 RFC 6238（HMAC-SHA1，30 秒時窗）以共享密鑰與當前時間獨立運算產生 [43]，全程不經任何網路傳輸；Supabase Auth 後端以同一份密鑰各自獨立驗算後比對通過與否。此機制所防範的是「密碼已外洩之後的二次入侵」（例如釣魚、資料庫脫褲、肩窺、暗網密碼字典撞庫等情境）：即使攻擊者取得帳號與密碼，因未持有使用者手機（亦即未持有共享密鑰）而無法產生當下這 30 秒有效的 6 位數，達成 NIST SP 800-63B [45] 所定義的「something you know（密碼）+ something you have（手機）」多因子組合，符合 OWASP Authentication Cheat Sheet [47] 所建議之多重驗證實踐，亦符合醫療資訊系統對個資與病歷存取的合規要求。

由於 ClinCalc 與 ExClinCalc 皆為純網頁應用、未發行任何原生行動 App，使用者端的 6 位數動態碼透過業界通用之第三方 Authenticator App（如 Google Authenticator、Microsoft Authenticator）顯示。實際操作上，使用者首次登入時於 ExClinCalc 網頁呈現一組 QR Code，使用者以手機上自行安裝之 Authenticator App 掃描即完成密鑰交換；其後該 Authenticator App 即在「ExClinCalc」項目下持續顯示當下 6 位數，使用者每次登入時將該 6 位數手動輸入至網頁登入欄位即可。Authenticator App 與本系統之間並無任何網路連線，密鑰僅於掃描 QR Code 之單次過程由網頁傳遞至手機本機儲存區（Authenticator App 之私有 keychain），之後手機端與伺服器端各自依本機時間獨立運算 TOTP，因此 Authenticator App 在離線狀態下仍可正常產生驗證碼。

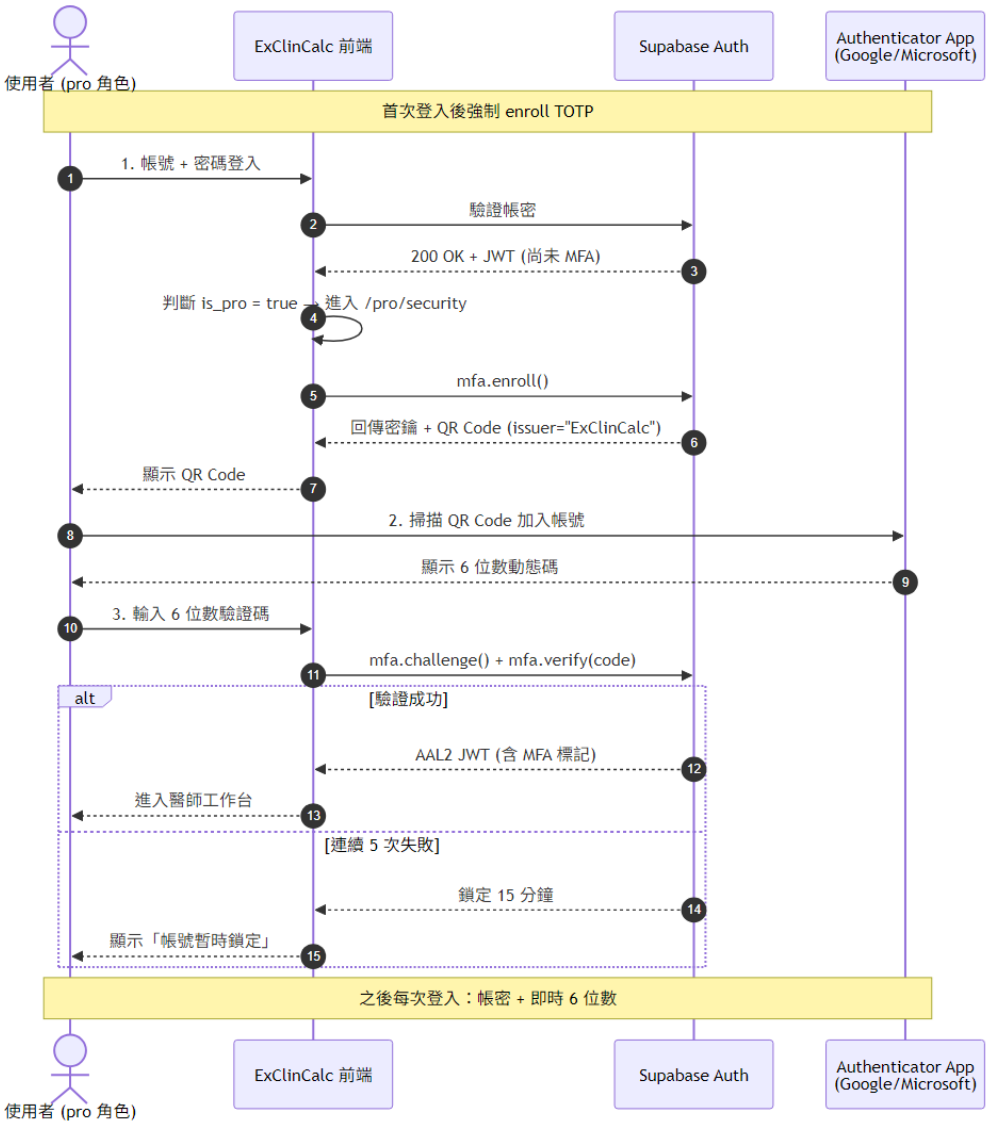

圖 3-3 TOTP 雙重驗證流程

實作層面上，整個 TOTP 流程涵蓋三個關鍵畫面與兩段 server-side 邏輯：（一）/pro/security 頁面的 enroll 介面（圖 3-4）顯示 Supabase Auth 即時生成的 QR Code、可手動輸入的密鑰、以及 6 位數驗證碼欄位；（二）綁定完成後同一頁面切換為「已啟用」狀態（圖 3-5），列出已綁定的驗證器名稱、綁定日期、最後使用時間，並提供「停用」與「新增另一個驗證器」按鈕；（三）後續每次登入時，src/middleware.ts 偵測 auth.mfa.getAuthenticatorAssuranceLevel() 之 currentLevel 與 nextLevel 落差後，自動將使用者導向 /auth/mfa-verify 完成第二步驗證；連續 5 次失敗以 sessionStorage 計數鎖定該瀏覽器 session 15 分鐘。若使用者遺失驗證器或被鎖定，管理員可至 /pro/admin/users 點選該帳號之「重置 2FA」按鈕，後端 /api/pro/admin/users 的 r

eset_mfa action 會以 service role 呼叫 auth.admin.mfa.deleteFactor 移除其所有 TOTP factor，使用者下次登入會被視為「未綁定 MFA」狀態並重新引導至 enroll 流程，整個過程同時寫入 audit_logs 留下管理員操作軌跡。

圖 3-4　TOTP 雙重驗證 enroll 介面（QR Code + 手動金鑰 + 驗證碼欄位）

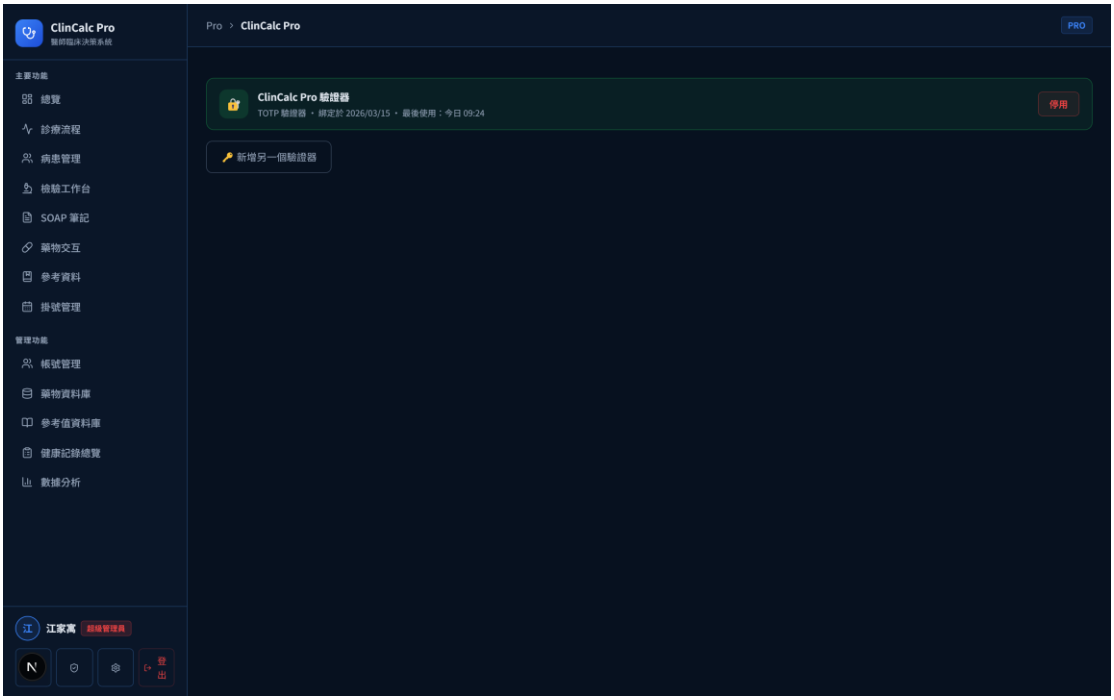

圖 3-5　TOTP 已綁定狀態（顯示驗證器名稱、綁定日期、停用按鈕）

稽核日誌

醫療系統不能只靠事前防範，也要能事後追溯。audit_logs 表記錄所有敏感操作：登入、處方建立、SOAP 修改、藥物交互查詢、未授權嘗試（policy 拒絕）等等都會留下紀錄。每筆有 user_id、action、target_table、target_id、ip、user_agent、created_at 七個欄位，由系統 trigger 自動寫入；前端開發時不必特別記得呼叫，避免遺漏。預設保留 90 天，由 Supabase 的 cron-style 排程定期清理已過期紀錄。管理員角色可在 /pro/admin/audit 頁面以時間、使用者、動作三種維度查閱與篩選。

第四節　介面設計原則

兩個子系統共用一套 design tokens——也就是把所有顏色寫成 CSS 變數，讓元件不必依賴特定色票。--accent、--text-primary、--text-secondary、--bg、--border、--danger 等變數透過 Tailwind CSS v4 [54] 的 @theme 指令注入；當需要替兩個子系統設定不同色調時，只要改 :root 裡的變數值即可，元件本身不必動。實際上線時，ClinCalc 採綠藍色系（accent #10b981），給人比較親和、健康的感受；ExClinCalc 採深藍色系（accent #3b82f6），凸顯專業與穩重。

明／暗主題切換是兩個子系統都做的功能。預設依 prefers-color-scheme 自動切換，使用者也可在右上角手動切換並寫入 localStorage。所有按鈕、卡片、輸入框的前景／背景對比皆參照 W3C Web Content Accessibility Guidelines（WCAG）2.2 [49] 之 AA 等級要求（一般文字對比比 ≥ 4.5:1、大字對比比 ≥ 3:1）進行設計，並以 Chrome DevTools 之 Lighthouse Accessibility 對每個頁面實際掃描後修正。

醫事端的版面則是另一個故事。一開始所有頁面都是「頂部 navigation bar + 中央內容」的標準排版，但在實際試用時，常常需要在「病患列表」「藥物交互」「SOAP 編輯」三個分頁間反覆切換，頂部 nav 的可用面積太小、操作不直覺。改成「左側 sidebar + 中央工作區」雙欄後操作節奏明顯流暢。Sidebar 會依登入角色動態顯示條目（醫師看到診療流程／病患／檢驗／SOAP／

藥物／參考；護理師看到護理工作台／病患列表／參考；藥師看到藥師工作台／交互作用／參考），這樣每個角色登入後看到的都是自己用得到的功能，沒有干擾。

第四章　系統實作與評估

第一節　ClinCalc 民眾端實作

ClinCalc 共有 36 個 TypeScript／TSX 檔案，主要功能模組包括：互動式身體地圖（/check/simple）、本地即時分析引擎（/check/detail）、Gemini 影像 OCR（/scan）、中英醫療翻譯（/translate）、用藥提醒（/meds）、健康記錄（/records）、病患授權（/consent/[token]）。ClinCalc 首頁如圖 4-1 所示。註冊與電子郵件驗證流程之介面如圖 4-2 所示。

圖 4-1　ClinCalc 首頁

帳號註冊與電子郵件驗證

民眾於 /auth/register 填入電子郵件、密碼與基本資料完成註冊後，Supabase Auth 會自動觸發一封驗證信至使用者信箱，信中含一條時效 24 小時的驗證連結；使用者點擊「驗證帳號」按鈕後，帳號的 auth.users.email_confirm

ed_at 才會被設為當前時間，未驗證之前無法登入主要頁面。此設計可避免他人冒用 email 註冊、確保健康記錄與 patient_consents 授權都繫於真實可達之信箱。驗證信介面如圖 4-2 所示。

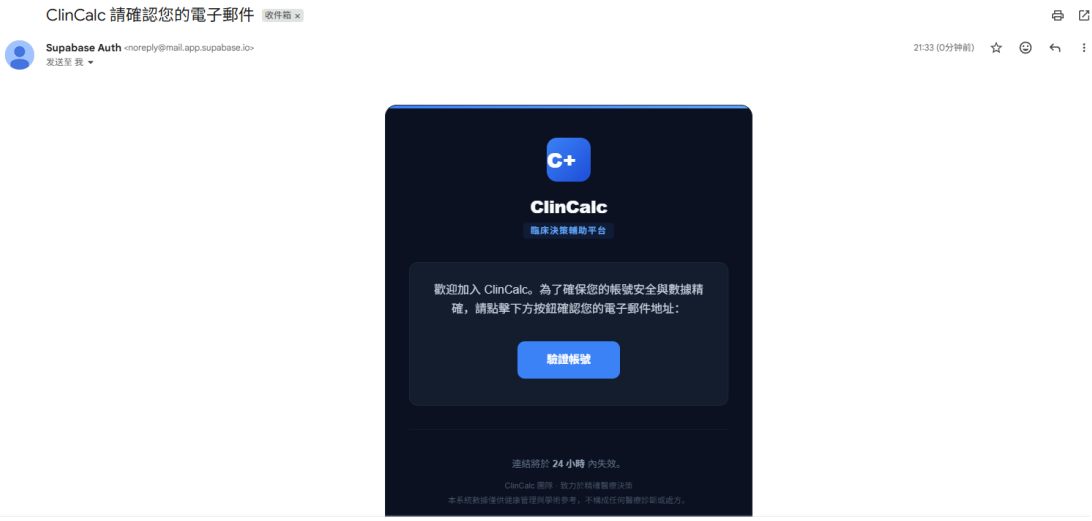

圖 4-2 ClinCalc 註冊後 Supabase Auth 寄送之 email 驗證信

互動式身體地圖

/check/simple 為使用者進入 ClinCalc 後的核心問診流程。畫面以一張 SVG 全身正視圖為主體，包含 17 個可點擊區域：頭、頸、左右肩、胸、上腹、下腹、左右上臂、左右前臂、左右大腿、左右小腿、左右足。使用者點擊任一區域後，下方彈出該區域對應的常見症狀清單，例如點擊「頭」後顯示「頭痛」「頭暈」「視力模糊」「耳鳴」等共 44 個跨區域症狀的對應子集。使用者可勾選多個症狀，並設定持續時間與嚴重度，介面如圖 4-3 所示。

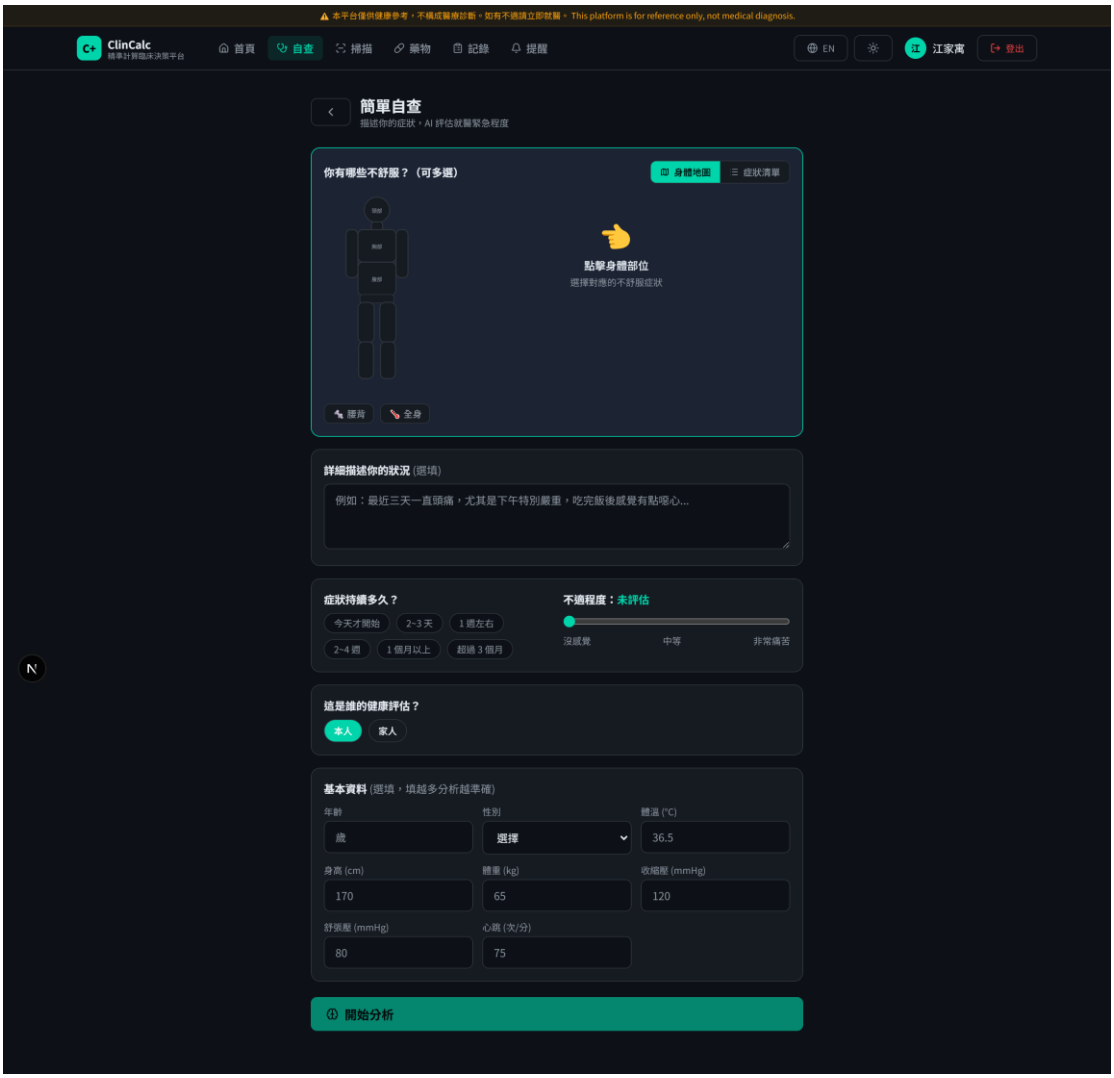

圖 4-3　互動式身體地圖（17 區 44 症狀）

點擊「下一步」後，前端整合所有勾選資料形成 prompt，呼叫後端 /api/gemini 路由（後端代理 Gemini 1.5 Flash），返回三類資訊：（一）可能病因之初步排序；（二）建議自我觀察事項；（三）建議就醫的紅旗症狀（例如胸痛伴隨左肩放射、突發性視力喪失等）。AI 回應均加註「僅供參考、不取代專業診斷」之免責聲明。

本地即時分析引擎

/check/detail 提供 45 項常見體檢指標的即時解讀。所有指標定義於 src/lib/referenceRanges.ts，每筆包含中英文名稱、單位、男女正常範圍、健康解讀、來源指引等欄位。指標分為十類：

分類	指標數	代表指標
一般生命徵象	5	收縮壓、舒張壓、心率、體溫、血氧
血糖與糖尿病	4	空腹血糖、HbA1c、口服葡萄糖耐受、血糖隨機
血脂	5	總膽固醇、LDL、HDL、三酸甘油脂、載脂蛋白
肝功能	5	AST、ALT、ALP、總膽紅素、白蛋白
腎功能	5	肌酸酐、eGFR、尿素氮、尿白蛋白比、尿酸
全血計數	6	紅血球、白血球、血小板、血色素、血容比、平均紅血球容積
電解質	5	鈉、鉀、氯、鈣、磷
甲狀腺	3	TSH、Free T3、Free T4
發炎指標	4	CRP、ESR、procalcitonin、白血球分類
其他	3	維生素 D、鐵蛋白、葉酸

表 4-1　45 項本地分析指標分類統計

KDIGO 2024 慢性腎臟病分期演算法以 TypeScript 實作於 src/app/check/detail/page.tsx，依使用者輸入的 eGFR 即時返回對應階段。詳細頁面（含分期卡）如圖 4-4 所示。

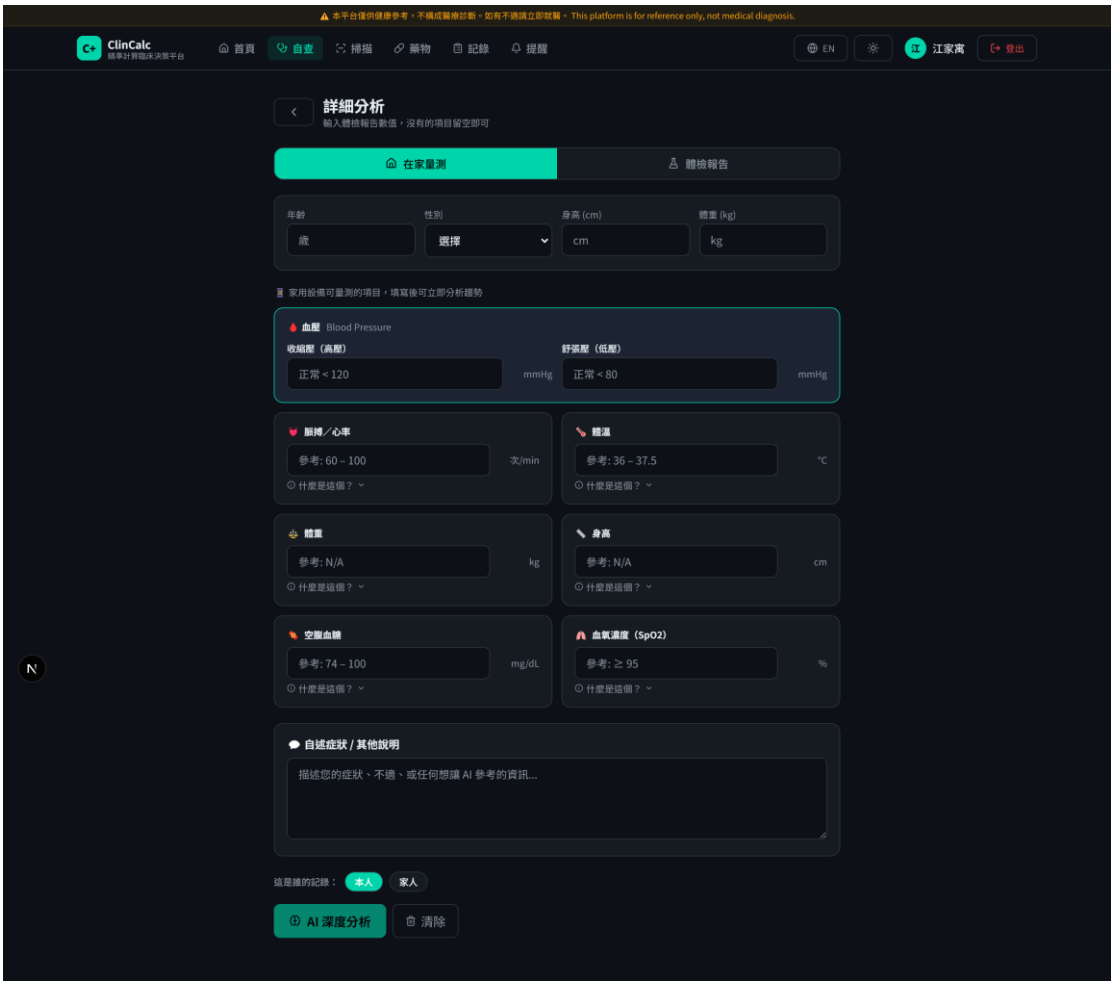

圖 4-4　45 項指標詳細檢測與 KDIGO 分期卡

階段	eGFR（mL/min/1.73m²）	嚴重度描述	建議行動
G1	≥ 90	正常或偏高	維持健康生活，每年追蹤
G2	60 – 89	輕度下降	控制血壓血糖，每 6 個月追蹤
G3a	45 – 59	輕至中度下降	至腎臟科評估，控制飲食蛋白質
G3b	30 – 44	中至重度下降	積極追蹤，評估腎臟保護藥物
G4	15 – 29	重度下降	準備透析或移植評估，嚴格飲食

G5	< 15	腎衰竭	立即評估透析或腎臟移植

表 4-2 KDIGO 2024 慢性腎臟病分期判讀表

該分期卡會以對應顏色（G1/G2 綠、G3a 黃、G3b 橘、G4 紅、G5 深紅）顯示於頁面上方，並引用 KDIGO 2024 為來源。

Gemini AI 整合

ClinCalc 把 AI 功能集中在三個入口：醫療影像 OCR（/scan）、中英醫療翻譯（/translate）、症狀分析（/check/simple）。這樣設計的考量是讓 AI 變成可選工具，使用者要主動進入這些頁面才會觸發外部 API 呼叫；其他頁面（特別是 /check/detail 的 45 項本地分析）完全不會用到 AI，原始檢驗數據也不會離開瀏覽器。

/scan 的流程是這樣的：使用者拍照或從相簿選一張檢驗報告影像，前端先用 <canvas> 壓縮到長邊 1280px（一方面節省 Gemini API 傳輸成本，一方面保留可辨識的解析度），透過 /api/gemini 後端路由轉送 Gemini 1.5 Flash。Gemini 回傳一個結構化 JSON，列出影像中辨識到的指標名、數值、單位三欄，前端再把這些值對應到 /check/detail 的對應欄位。實測對印刷體報告的單張準確率約 95%，對手寫的辨識率本研究尚未量化。介面如圖 4-5。

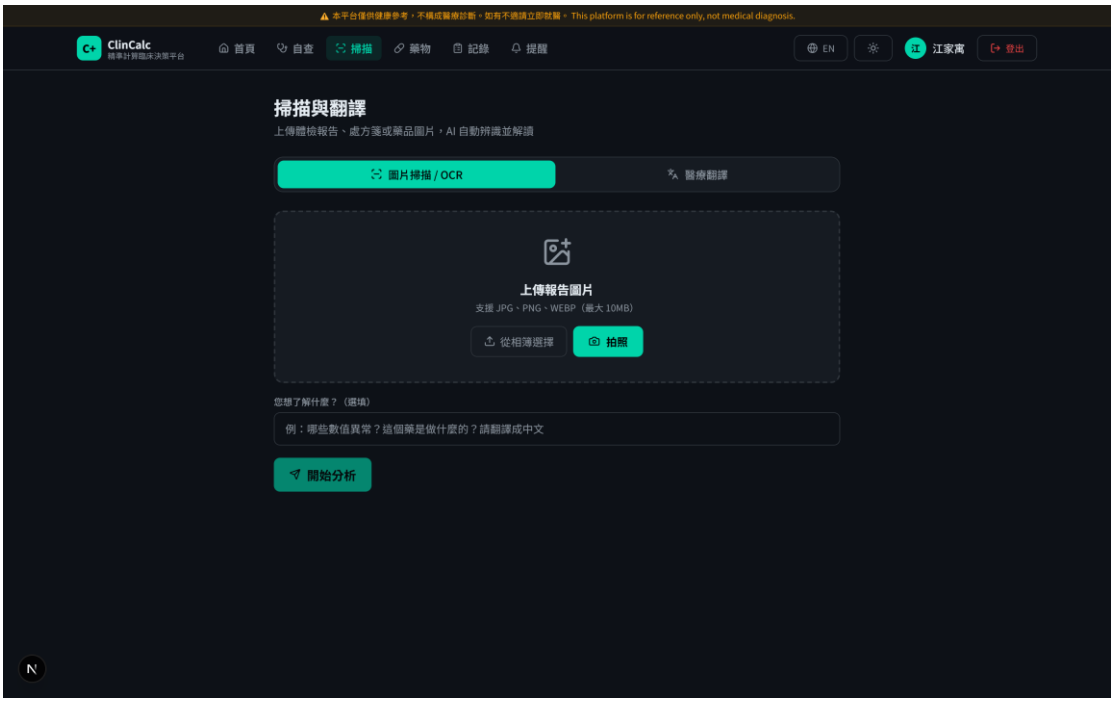

圖 4-5 AI 醫療影像 OCR 掃描介面

/translate（圖 4-6）讓使用者貼上中文或英文檢驗報告／衛教資訊後做雙向翻譯，同時把 Gemini 回應中的醫療專有名詞用底線粗體標示出來，方便讀者進一步查閱。/check/simple 的症狀分析則如前述，把使用者點擊的身體部位與症狀組合成 prompt，請 Gemini 列出可能病因、自我觀察建議與紅旗症狀。

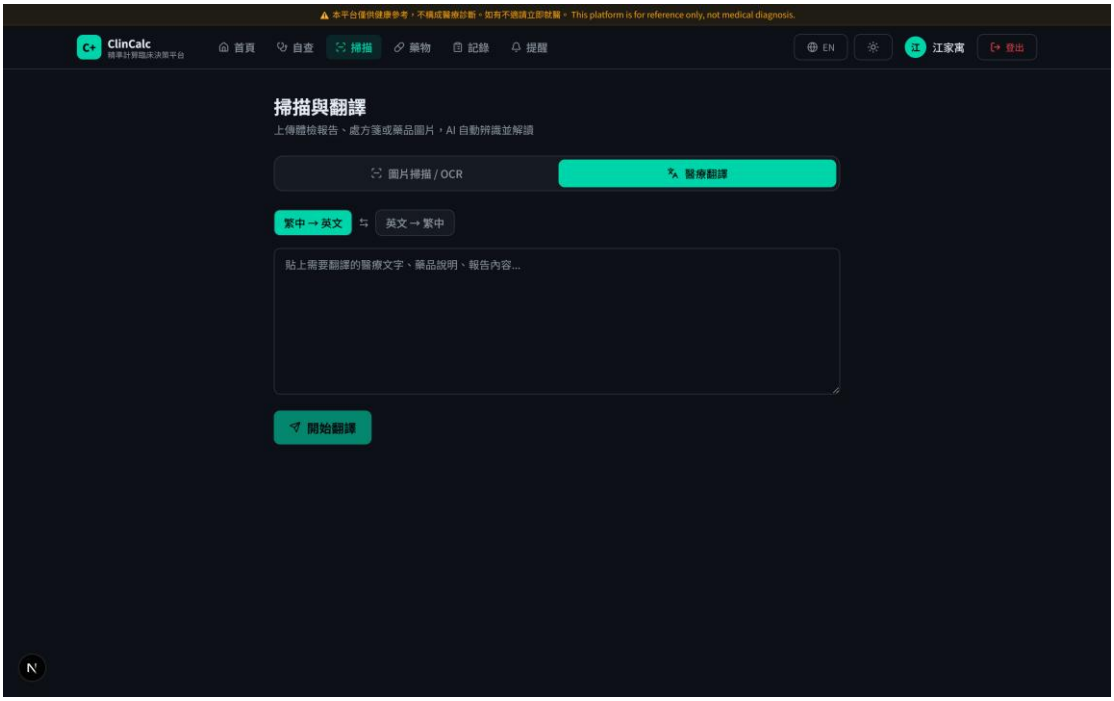

圖 4-6 中英醫療翻譯

最後是金鑰管理的問題。Gemini API 金鑰是付費資源，本研究在開發初期曾考慮直接存在前端讓使用者自己填，但測試後發現太容易被竊用——光是把 ClinCalc 部署到 Cloudflare Pages preview 環境，金鑰就會出現在網頁原始碼裡。最終的設計是把所有對 Gemini 的呼叫都經由後端代理（/api/gemini），金鑰只存在於 Cloudflare Workers 的環境變數中，前端只能看到後端轉送回來的結果，看不到 API 金鑰本身。

知識庫優先的 prompt 組裝策略

本研究在整合 Gemini 時刻意採取「**先查知識庫、再交給 LLM**」（Knowledge-base-first prompting）的策略，而不是把使用者輸入直接丟給 Gemini 任其自由發揮。具體流程以 /check/detail 為例（程式邏輯在 src/app/check/detail/page.tsx）：

第一步，使用者填完 45 項指標後，前端先逐一查詢本地的 referenceRanges.ts 知識庫——這份檔案是專為台灣使用者整理的指標對照表，每一項都包含中英文名稱、單位、性別分組的男女參考區間、健康解讀說明、以及來源指引（KDIGO 2024 [6]、ADA 2026 [10]、ACC/AHA 2017 [11] 等）。對每個填了值的欄位，前端呼叫兩個函式：checkAbnormal(item, value, gender) 判定當前數值是「正常 / 偏高 / 偏低 / 嚴重偏高 / 嚴重偏低」五個狀態之一，getNormalRange(item, gender) 則取出對應性別的參考區間。

第二步，把這些**結構化判定結果**而非原始數值組成 prompt，每行格式為「指標中文名（英文名）: 數值 單位［參考: 區間, 狀態: 正常/偏高/偏低］」，再附上使用者的基本資料與自述症狀，最後加上明確的回應規則（要求附免責聲明、不推薦具體藥物、用一般人看得懂的語言）。

第三步，Gemini 拿到的是已經被知識庫標記好的「異常指標清單」，僅需做三件事：（一）對異常指標做整體性的健康評估；（二）依異常組合提出生活型態建議；（三）判斷是否需要就醫並給出時程建議。Gemini 不再需要自行記憶各項指標的正常範圍（避免幻覺）、也不需要判定哪些值算異常（已由本地知識庫完成）。

這個策略的好處有三個：第一，**準確性提升**——指標的正常範圍以本地 referenceRanges.ts 為準，不仰賴 LLM 的訓練語料中可能過時或地區差異的數據；第二，**幻覺降低**——LLM 較少在簡單事實（例如「血糖 126 mg/dL 算高」）上犯錯，因為這個事實已經由本地查表得出；第三，**可追溯性**——若 Gemini 對某指標的判讀與本地知識庫不一致，本研究選擇以本地知識庫為準，因為知識庫的每一行都標註了來源指引，可以回溯到原始 PDF 文件。本研究也將同樣的策略套用在 ExClinCalc 醫事端：/pro/encounter 在處方階段時，先從 medications 表的 interactions TEXT[] 欄位逐筆查詢已知交互（drugInteractions.ts），再把命中結果交給 Gemini 做整體性的處方建議；如此 Gemini 不會在「Warfarin 與 Aspirin 是否交互」這類已有定論的問題上自由發揮。

用藥提醒與健康記錄

/reminders 提供使用者輸入個人服藥清單與服藥時段。時間欄位採三段式下拉（上午／下午、12 小時、5 分鐘間隔）並提供「早餐 08:00／午餐 12:00／晚餐 18:00／睡前 22:00」四個快速時段按鈕，避免桌面瀏覽器 <input type="time"> 預設介面對中文使用者不友好的問題。儲存後瀏覽器層呼叫 Notification.requestPermission() 取得通知權限，並於指定時間發出本地通知（圖 4-7 為提醒清單；圖 4-8 為實際跳出通知 popup 的範例，含「已服用／10 分鐘後提醒」兩個快捷選項）。提醒資料同步寫入 health_records 的 indicators JSONB 欄位（{ medication: [...] }），跨裝置可同步。

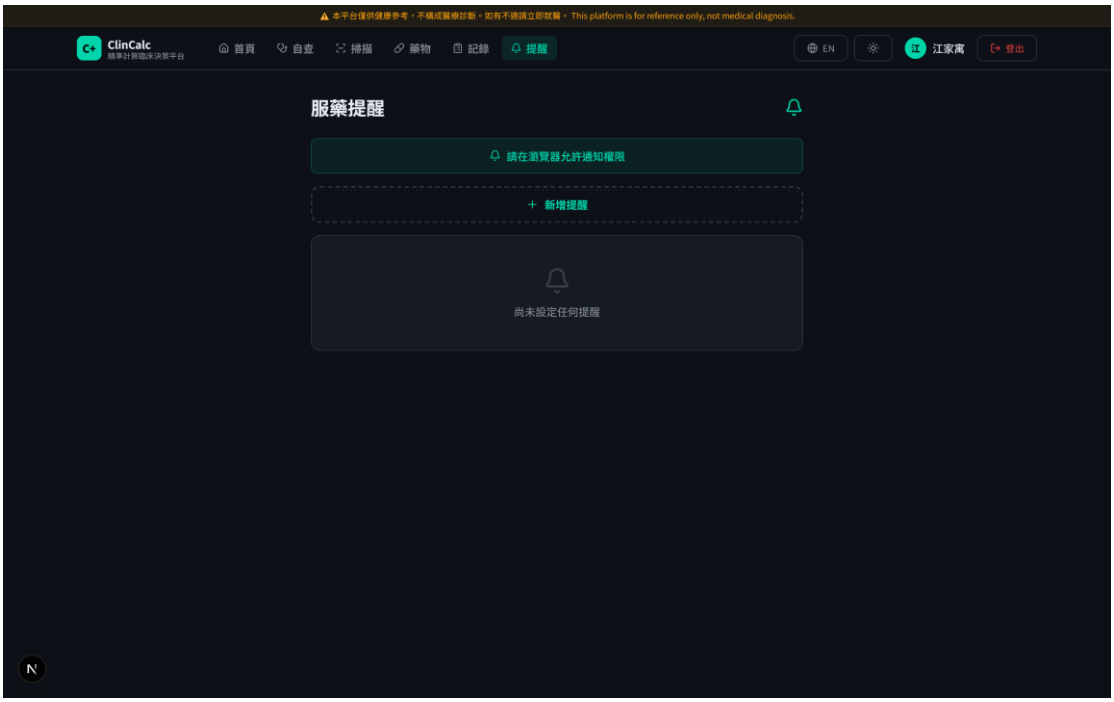

圖 4-7　用藥提醒清單（含上午／下午、小時、分鐘三段下拉與快速時段按鈕）

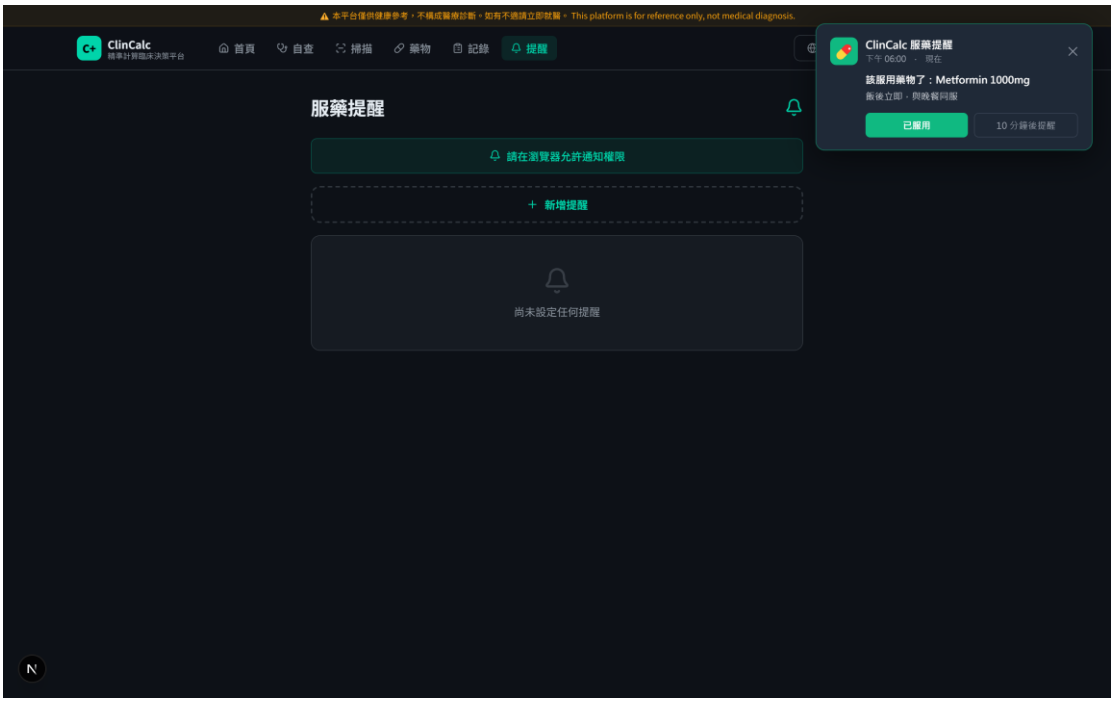

圖 4-8　服藥提醒實際通知 popup 範例

/records 以列表呈現使用者全部健康記錄並支援趨勢分析。畫面上方「趨勢分析」區段以 chip 形式列出所有有 ≥3 筆數據的指標（如「糖化血色素（HbA1c）10筆」「收縮壓（高壓）10筆」），使用者點選任一 chip 即彈出該指標的時序圖 modal，顯示過去 9 個月（或指標實際資料覆蓋範圍）的折線走

勢、與初次紀錄的差距、最新值徽章。圖表使用 Recharts 繪製，所有資料於前端從 health_records 一次取出後本地分組計算，避免多次 API 呼叫。圖 4-9 為健康記錄主畫面（10 筆資料 + 趨勢分析 chips），圖 4-10 為點擊「糖化血色素 (HbA1c)」chip 後彈出的時序圖 modal（顯示 7.8% → 8.4% → 5.9 % 的 9 個月趨勢）。

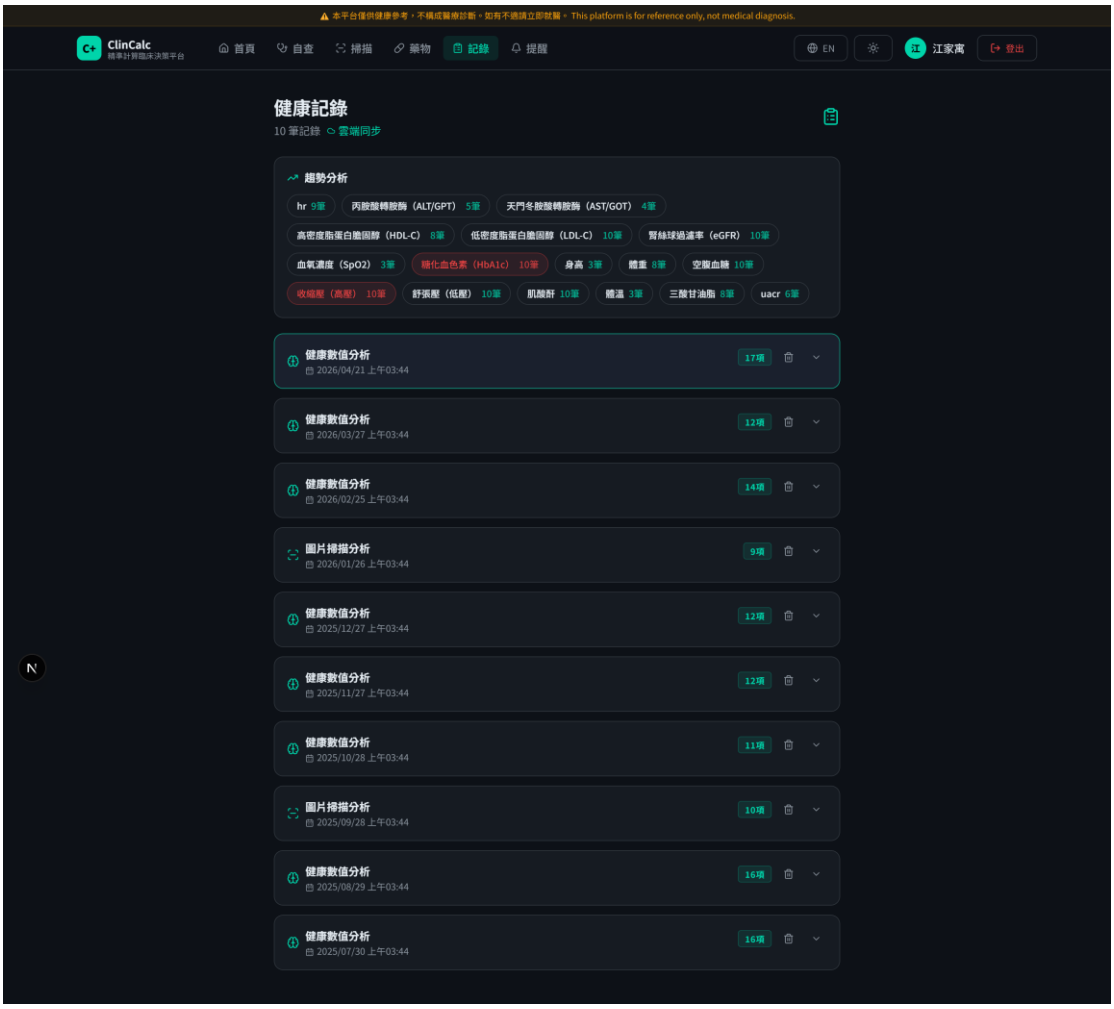

圖 4-9　個人健康記錄主畫面（含趨勢分析 chips）

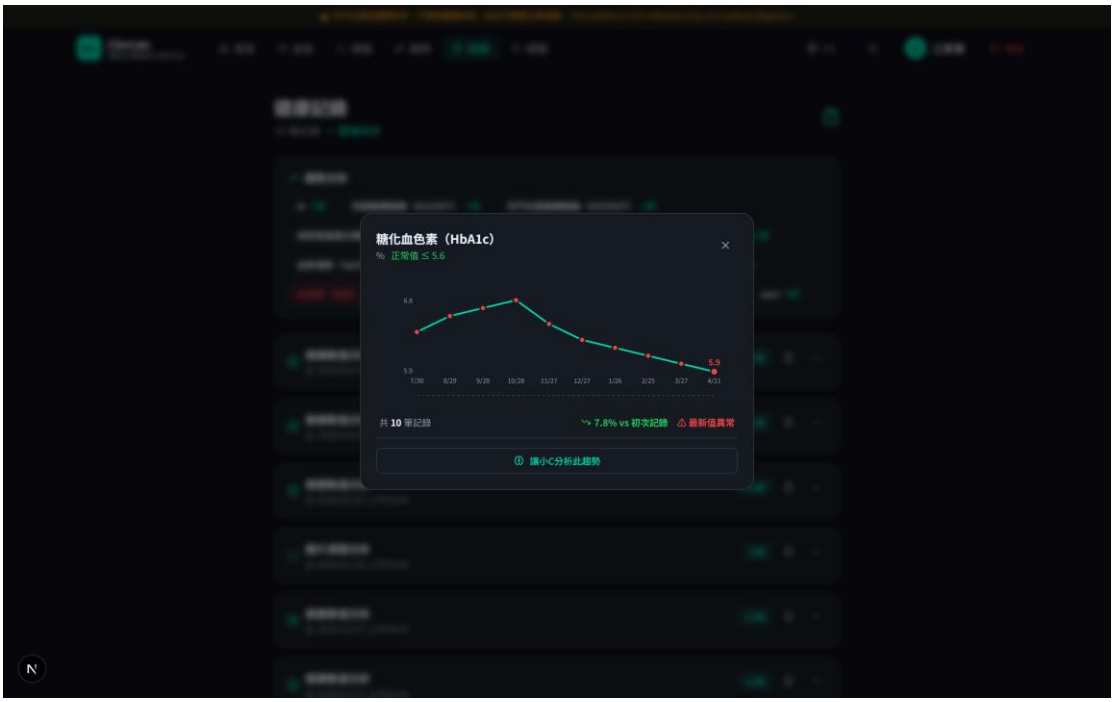

圖 4-10　時序圖 modal 範例（HbA1c 9 個月趨勢）

第二節　ExClinCalc 醫事端實作

ExClinCalc 共 65 個 TypeScript／TSX 檔案，覆蓋醫師、護理師、藥師、行政、管理員、超級管理員六種角色之工作台。完整診所流程閉環為：**掛號（appointments）→ 護理分診（nursing）→ 醫師診療（encounter）→ 藥師調配（pharmacy）**，每個環節對應一張或多張資料表，跨環節資料以 patient_id 與 appointment_id 串聯。

醫師儀表板與掛號管理

/pro/dashboard 為醫師登入後的首頁，呈現四張 stat card：今日已完成診次、今日候診中、本週新增病患、本週開立處方。今日候診中卡片可直接點擊跳轉至 /pro/appointments，介面如圖 4-11 所示。

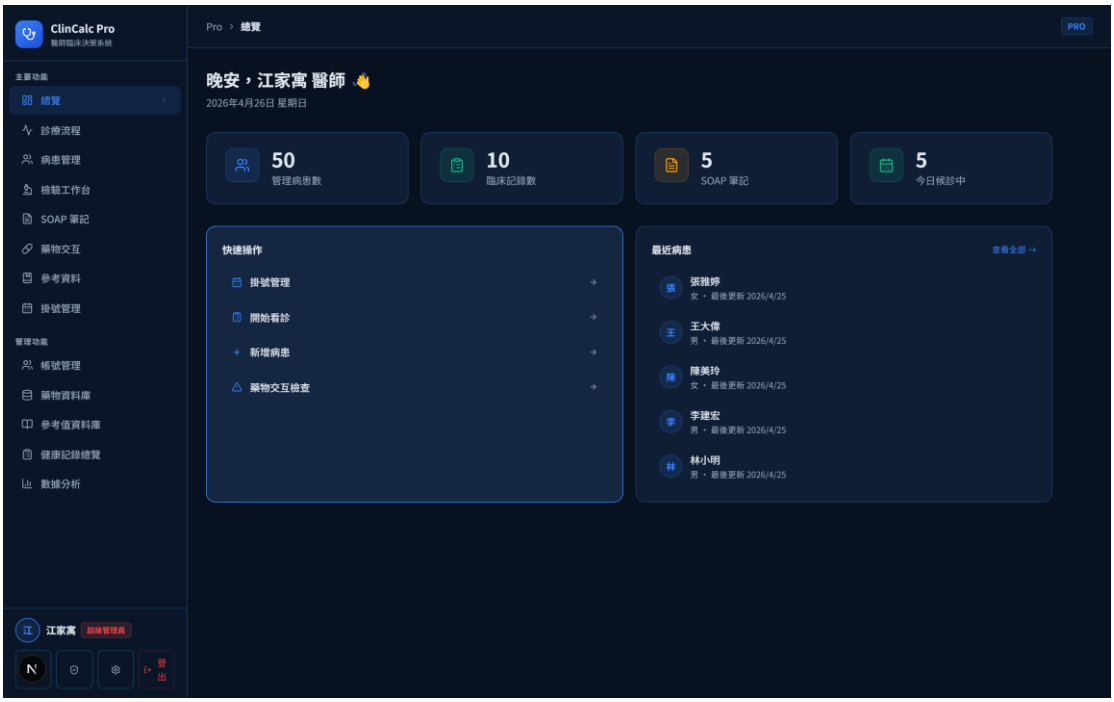

圖 4-11　醫師儀表板（今日候診卡）

/pro/appointments 以表格列出當日所有掛號，含 queue_number、患者姓名、主訴、checked_in_at 時間、status 狀態。狀態分四類並以不同顏色標示：waiting（候診中）、in_progress（看診中）、completed（已完成）、cancelled（已取消）。醫師對 waiting 狀態的患者按下「叫號看診」按鈕後，後端會將該筆 status 改為 in_progress，並導向 /pro/encounter?pid={patient_id}&complaint={chief_complaint}，自動帶入該患者與主訴；若不慎將患者標為 cancelled，可在「已取消」分頁找到並按「回補」鈕還原為 waiting。介面如圖 4-12。

圖 4-12　掛號管理（叫號／回補／已取消）

護理師分診工作台

/pro/nursing 列出當日所有 waiting 狀態的患者。護理師點選任一筆後進入分診面板，輸入血壓、心率、呼吸、體溫、血氧、體重、身高七項生命徵象，按下「儲存」後資料寫入 triage_vitals，並將 appointments.status 自動轉為 triaged。介面如圖 4-13。

圖 4-13　護理師分診工作台

醫師端的 /pro/encounter 在選擇病患後會自動查詢該病患當日是否有 triage_vitals 紀錄，若有且 used_at IS NULL，則於頁面頂端橫幅顯示「分診資料可用：BP 158/96, HR 74, T 36.8℃」並提供「一鍵帶入」按鈕，按下後自動填入醫師端的生命徵象欄位，並將 triage_vitals.used_at 設為 now()，避免重複帶入。

醫師 SOAP 七步驟診療流程

/pro/encounter 為 ExClinCalc 最複雜的頁面，將 SOAP 病歷拆解為七個步驟，依序引導醫師完成：

1. **生命徵象**（自動帶入或手動補）
2. **主觀症狀**（依主訴模板提供 5–8 題引導問題）
3. **客觀檢查**（理學檢查欄位）
4. **檢驗建議**（依主訴自動建議檢驗組合，醫師可一鍵套用）
5. **評估診斷**（系統依輸入自動排序 ICD-10 候選代碼）
6. **處方**（從 medications 表搜尋 + 自動藥物交互檢查）
7. **衛教與追蹤**

主訴模板共 20 種（定義於 src/lib/pro/clinicalFlow.ts），含頭暈／頭痛、疲勞／倦怠、多尿／口渴、關節腫痛、胸悶／心悸、腹部不適、定期健檢、高血壓追蹤、糖尿病追蹤、感冒／上呼吸道感染、發燒、咳嗽、腹瀉／腸胃炎、皮膚問題、腰背痛、泌尿道症狀、失眠、慢性腎臟病追蹤、COVID-19／呼吸道感染等。每模板對應一組標準問診題目（select／multiselect／scale 三種題型）與建議檢驗組合。介面如圖 4-14。

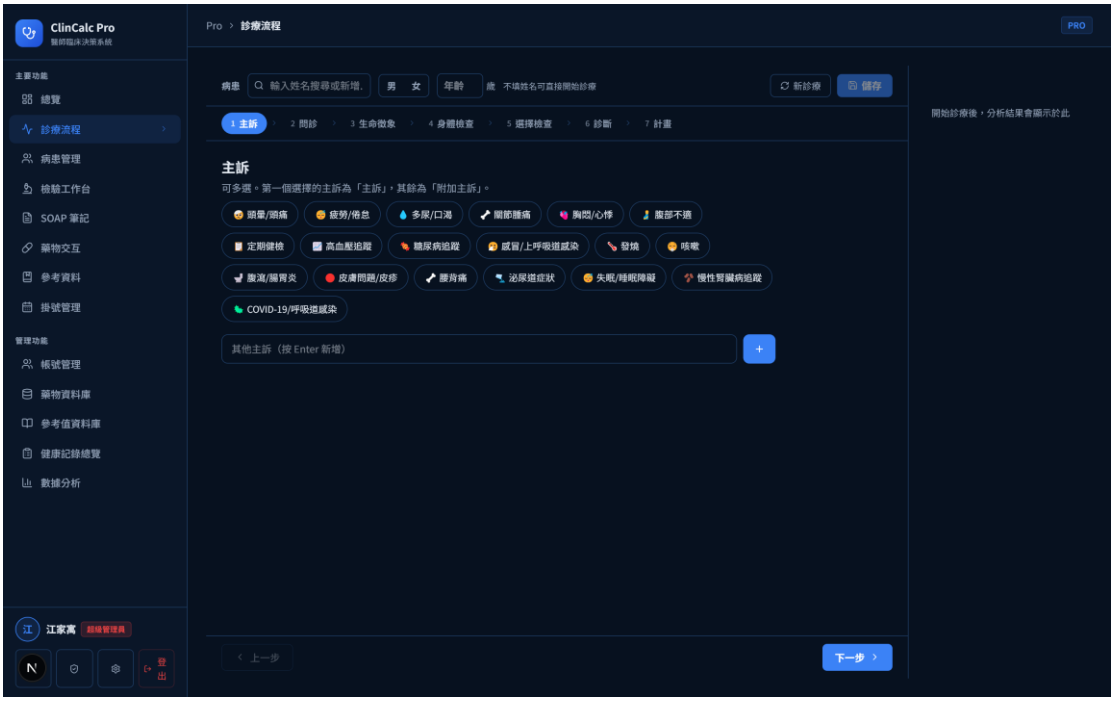

圖 4-14　醫師 SOAP 七步驟診療流程

主訴模板	題目數	建議檢驗組
頭暈／頭痛	7	cbc, electrolytes, glucose
疲勞／倦怠	7	cbc, thyroid, glucose, liver
多尿／口渴	6	glucose, hbalc, electrolytes, renal
胸悶／心悸	6	cbc, troponin, ecg, electrolytes
慢性腎臟病追蹤	7	renal, electrolytes, cbc, glucose
COVID-19／呼吸道感染	6	cbc, liver

……（其他 14 種主訴略）	—	—

表 4-3　20 種主訴模板與對應檢驗組合（節錄）

當醫師完成 SOAP 並按「儲存」後，後端寫入 soap_notes（含 JSONB 化的全部七步驟內容），並將 appointments.status 自動轉為 completed、completed_at 設為 now()。儲存成功畫面會出現「回到候診 →」按鈕，點擊回到 /pro/appointments 接續下一位患者。病患管理列表如圖 4-15。

圖 4-15　病患管理列表

藥物交互作用檢查

藥物交互作用的查詢介面有兩個入口：/pro/drugs 是獨立查詢頁（圖 4-16），讓醫師可以隨時輸入幾種藥檢查；/pro/encounter SOAP 第六步「處方」則在醫師寫處方時自動在背景跑同一套邏輯。背後的檢查邏輯位於 src/lib/pro/drugInteractions.ts，採兩段式設計。

第一段是**靜態關鍵交互表**——這是一張硬編碼在原始碼中的清單，列出 12 組必須警示的高風險組合。例如 Warfarin 配 Aspirin（出血風險，major）、Warfarin 配 Ibuprofen（NSAIDs 置換蛋白結合，major）、Sildenafil 配 Ni

trate（嚴重低血壓，contraindicated）、Metformin 配碘造影劑（腎傷害與乳酸中毒風險，major）、Simvastatin 配 Amiodarone（CYP3A4 抑制致橫紋肌溶解，major）、Clopidogrel 配 Omeprazole（CYP2C19 抑制降低療效，moderate）、SSRI 配 Tramadol（血清素症候群風險，major）。會這樣硬編碼是因為這 12 組屬於「無論其他資料庫怎麼變動都不能漏掉」的關鍵組合，放在程式碼裡可以避免資料庫被誤刪或誤改而失去警示效果。

第二段是**動態 medications 表交互**——每筆藥物於 medications.interactions TEXT[] 欄位記錄與其交互的藥物名或類別。查詢時系統會對輸入的處方藥兩兩比對，看其中一方是否列在另一方的 interactions 陣列中；命中後即輸出對應的交互說明。這部分由系統管理員透過 /pro/admin/medications 頁面維護，可隨時更新。

兩段邏輯各自跑完後，結果合併、依嚴重度排序，由禁忌（contraindicated，紅）、重大（major，橘）、中度（moderate，黃）、輕微（minor，灰）到無交互（none，綠）依序顯示。每筆結果都附詳細說明，告訴醫師「為何交互」與「臨床建議怎麼做」（例如「Amiodarone 抑制 CYP3A4，使 Simvastatin 血中濃度上升，建議 Simvastatin 不超過 20 mg/day 或改用 Pravastatin」）。

圖 4-16　藥物交互作用檢查

嚴重度	中文	顯示色	建議處置
contraindicated	禁忌	紅（#ef4444）	不可合併使用
major	重大	橘（#f97316）	必要時嚴密監測
moderate	中度	黃（#f59e0b）	注意臨床表現
minor	輕微	灰（#94a3b8）	一般無需處理
none	無	綠（#10b981）	安全合併

表 4-4　12 組靜態關鍵藥物交互對照（嚴重度色標）

種子 seed_medications.sql 共 30 種藥物，分布如下：

類別	藥物數	代表藥
降血壓藥（CCB／ARB／ACEI／β-blocker）	5	Norvasc、Diovan、Capoten
降血糖藥（雙胍、磺醯脲、SGLT2、DPP	5	Metformin、Jardiance、J

-4、胰島素）		anuvia
降血脂藥（statin、ezetimibe）	4	Lipitor、Crestor、Zetia
抗血栓藥	3	Plavix、Aspirin、Coumadin
抗生素／止痛	6	Augmentin、Cipro、Tylenol、Ibuprofen
胃藥／呼吸／其他	7	Omeprazole、Ventolin、Allegra

表 4-5　30 種種子藥物資料分類統計

藥師調配工作台與其他角色

/pro/pharmacy 為藥師專用工作台（圖 4-17），列出當日所有 clinical_records 含處方者；每張卡片顯示病患姓名、主訴、診斷與完整處方表（藥物、學名、劑量、頻次、途徑、預計使用天數、備註）。藥師有兩條操作路徑：（一）按「**完成調配**」鈕直接確認該筆處方，狀態寫入 dispensed_at、稽核記錄寫入 drug_interaction_checks 與 audit_logs；（二）若需修正——例如 AI 醫師端建議的劑量或天數需依藥師專業判斷調整、或補開遺漏藥物——可按「**修改處方**」進入編輯模式，每一欄（藥名、學名、劑量、頻次、途徑、天數、備註）都變成可編輯欄位，並可逐筆移除或按「**新增藥物**」加入新項目。修改完成後按「**儲存修改**」即把調整後的 prescriptions JSONB 寫回 clinical_records。此設計的目的是讓 AI 與醫師端的「初稿處方」與藥師端的「最終調配處方」能在同一筆紀錄上協作，而非兩個獨立資料源。

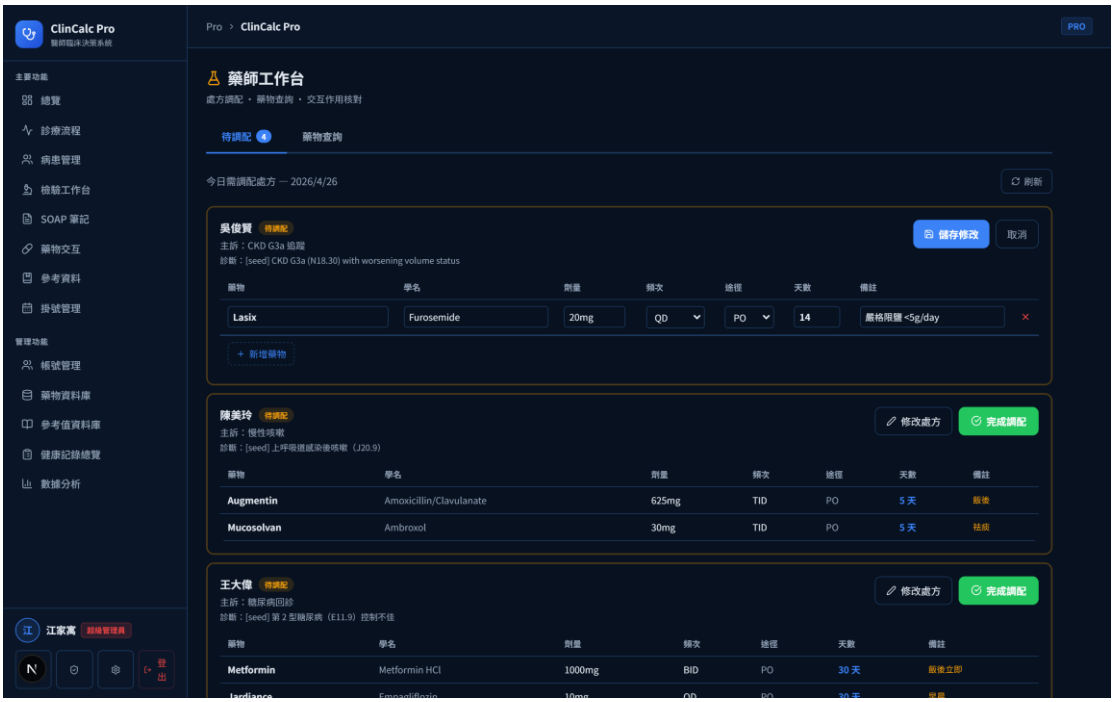

圖 4-17　藥師調配工作台（含處方編輯模式）

/pro/exam（檢驗工作台，圖 4-18）支援批次輸入檢驗結果並自動回填到對應 soap_notes；/pro/notes 為 SOAP 病歷列表與細節檢視（圖 4-19）；/pro/references 提供醫師查閱 medical_references 與 reference_pdf_links（如 KDIGO、ADA、ACC/AHA 指引，圖 4-20）。每張資料卡右上角附星號圖示，點擊即可加入個人收藏；收藏的資料會自動置頂顯示於列表前端，邊框以橙色強調。第一次進入頁面時，系統會自動將 ADA Standards of Care、KDIGO Clinical Practice Guideline、ACC/AHA 等核心指引預設為收藏狀態，方便基層診所醫師日常開診時優先查閱。收藏狀態以 localStorage 儲存於使用者瀏覽器，避免額外 DB schema 變更亦保留個人化彈性。

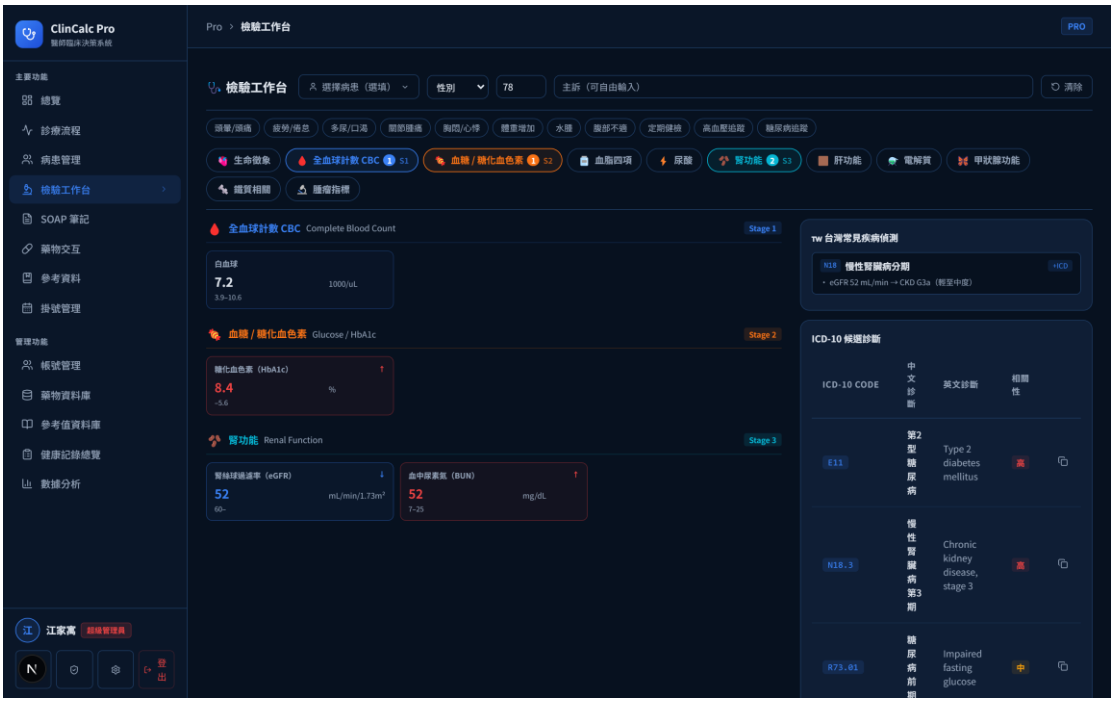

圖 4-18　檢驗工作台

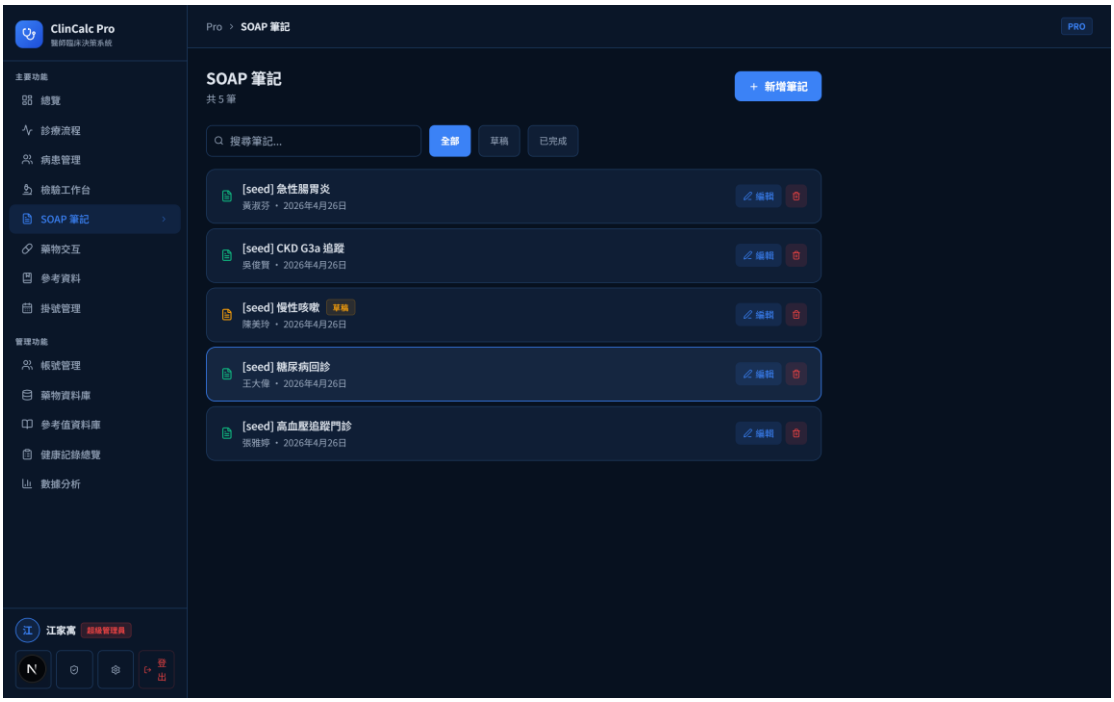

圖 4-19　SOAP 病歷列表

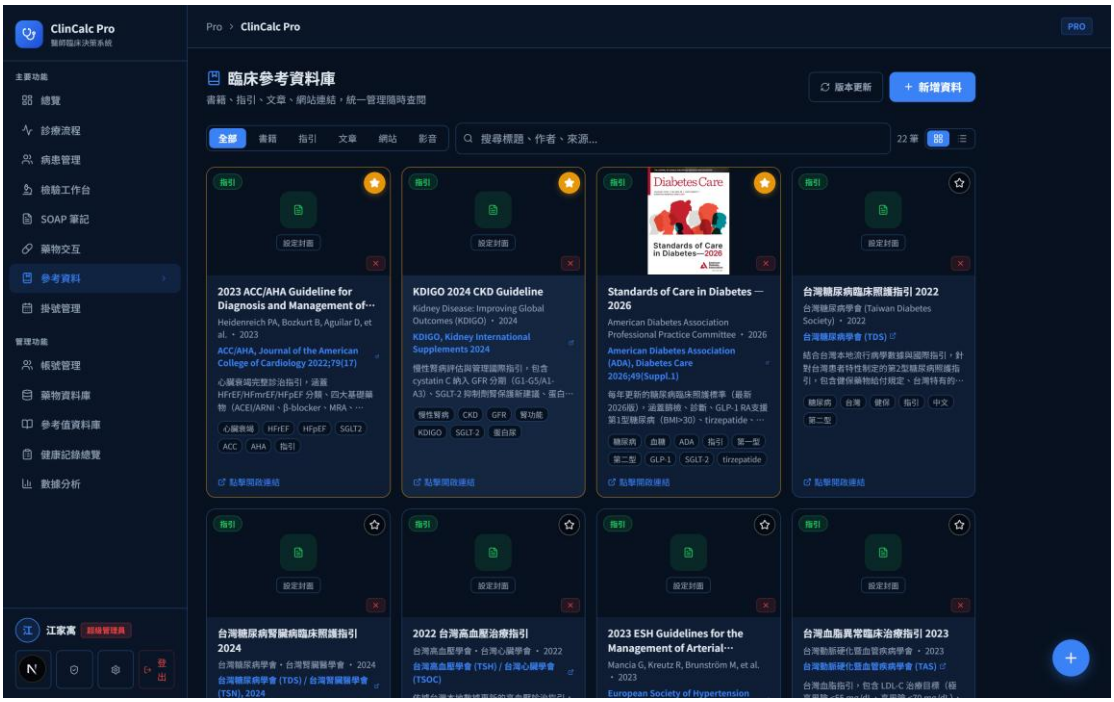

圖 4-20　參考資料庫（PDF 連結）

ICD-10 自動建議資料表 clinicalAnalysis.ts 涵蓋 11 大檢驗類別之自動代碼建議，每筆含信心等級（high／moderate／low）：

指標類別	高信心代碼數	中信心代碼數	低信心代碼數
血糖（glucose）	1	2	0
HbA1c	1	1	0
收縮壓	1	0	1
LDL／三酸甘油脂	2	2	0
ALT／AST（肝）	2	0	1
肌酸酐	1	1	1
eGFR	1	1	0
血色素	1	1	0
白血球	2	1	0

血小板	1	0	2
其他（電解質、甲狀腺等）	—	—	—

表 4-6 ICD-10 自動建議候選代碼分布

/pro/admin/users（使用者管理，圖 4-21）、/pro/admin/medications（藥物 DB，圖 4-22）、/pro/admin/references（參考值 DB）、/pro/admin/records（健康記錄總覽）、/pro/analytics（分析儀表板，圖 4-23）為 admin 與 super_admin 角色專屬介面。/pro/admin/users 除了基本的角色升降級、密碼重置、刪除帳號之外，還提供「**重置 2FA**」按鈕（橙色手機圖示），管理員可一鍵移除被鎖定或遺失驗證器之使用者所綁定的 TOTP factor，使其下次登入回到 enroll 流程；此操作同步寫入 audit_logs，避免管理員濫用形成審計斷點。所有 ExClinCalc 頁面左下角的使用者卡片均顯示登入帳號姓名與其角色標籤（依 pro_role 著色：醫師藍、護理師橙、藥師粉、行政紫、管理員綠、超級管理員紅），方便使用者隨時確認當前身份。

至於 **admin_staff（行政人員）** 角色，本研究在系統實作上採取「**權限差異化、UI 不重複實作**」的策略，未額外建立行政人員專屬工作台。原因有二：第一，行政人員的工作流程（協助掛號登記、查詢病患基本資料、列印就診紀錄等）所需的功能在現有的 /pro/appointments、/pro/patients、/pro/admin/users 三個頁面已涵蓋，僅需要以 SELECT-only 方式存取即可；第二，重複實作一套精簡版 UI 會增加維護成本且容易與主要 UI 漂移。本研究選擇在資料庫層以 RLS policy 限制 admin_staff 對 profiles 僅有 SELECT 且限同診所，凡是 admin_staff 進入上述三個頁面，PostgreSQL 都會自動過濾欄位與篩選列，前端不必為此寫任何條件分支。此設計呼應第三章「授權檢查不寫在後端，由資料庫層完成」之安全模型——換句話說，UI 層級的角色差異化由 RLS 自動產生，而非在前端硬編碼。

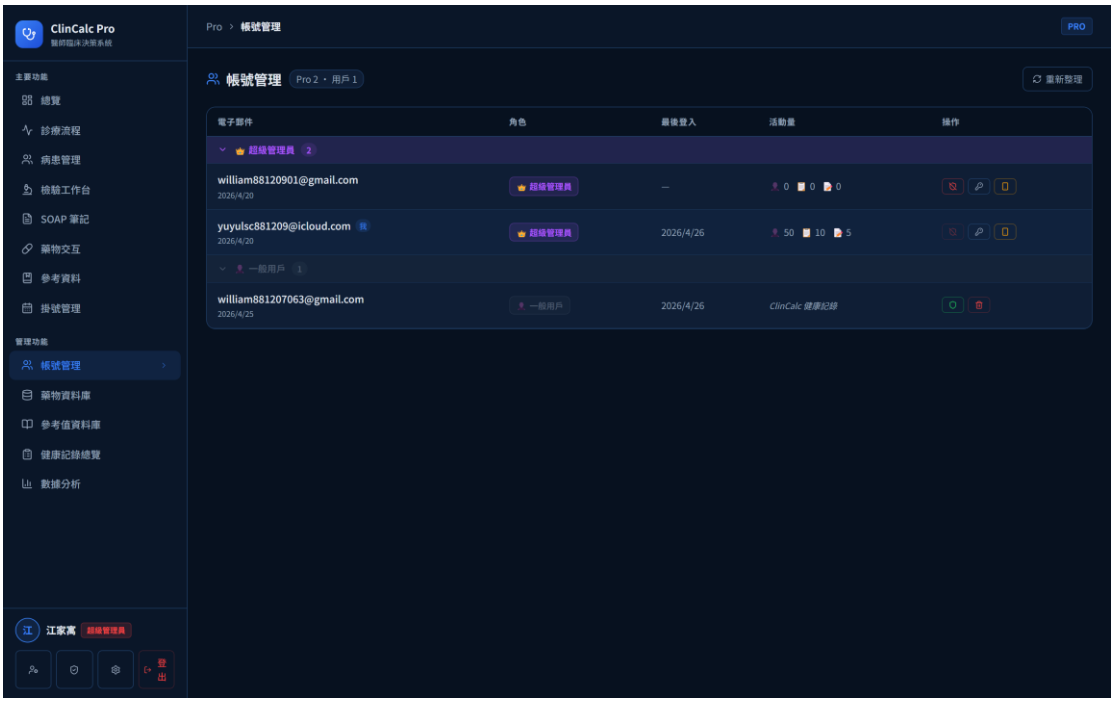

圖 4-21　管理者：使用者管理

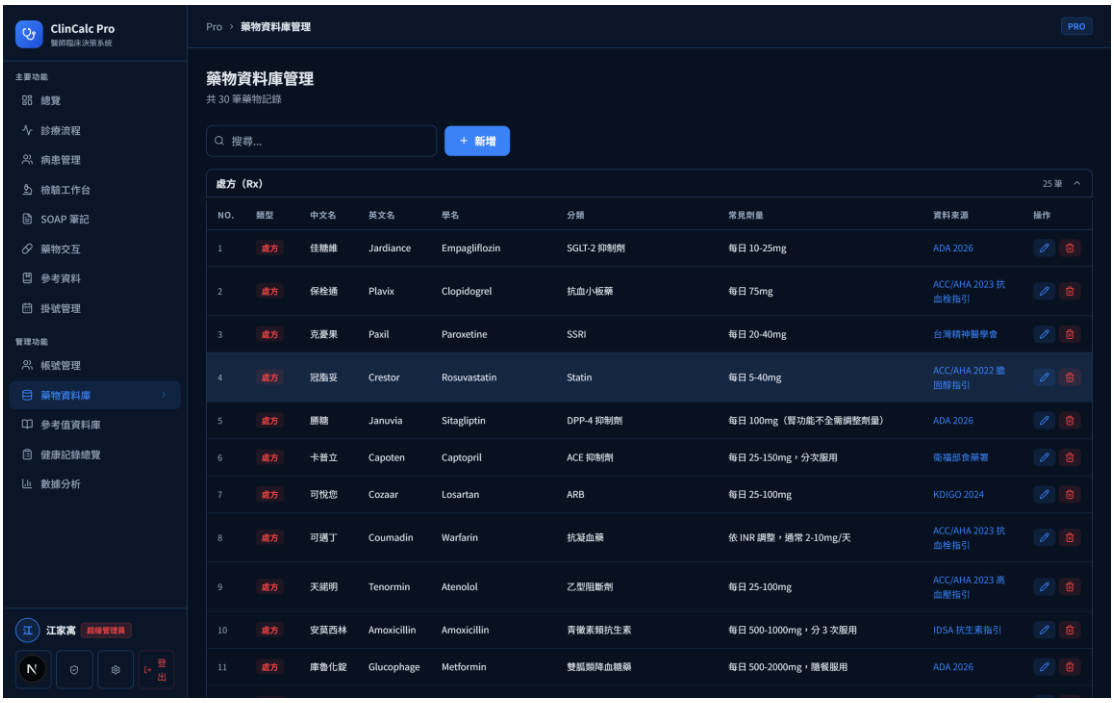

圖 4-22　管理者：藥物資料庫

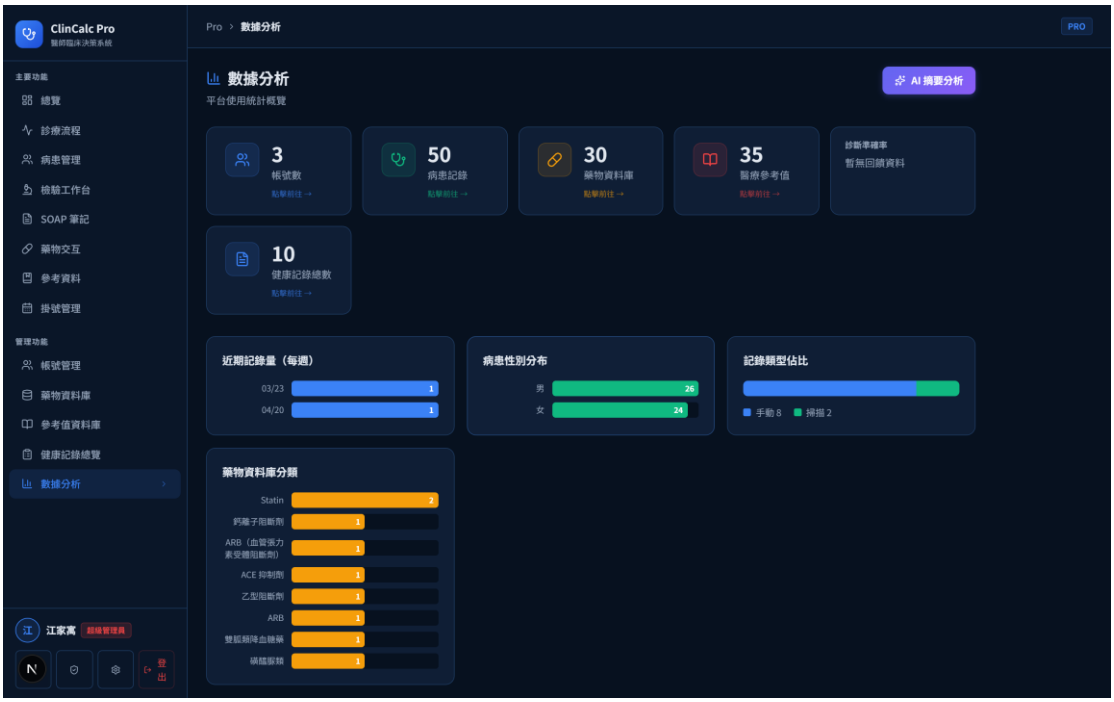

圖 4-23　管理者：分析儀表板

第三節　系統測試

端對端功能測試

本研究設計 10 項端對端（end-to-end）功能情境，涵蓋兩個子系統的核心流程：

#	測試情境	涉及子系統	結果
1	民眾註冊 → 輸入體檢值 → 查看 KDIGO G3a 分期顯示	ClinCalc	通過
2	民眾上傳檢驗影像 → Gemini OCR 自動帶入指標	ClinCalc	通過
3	民眾於 /check/simple 完成身體地圖症狀問診 → 收到 AI 評估	ClinCalc	通過
4	民眾透過 /consent 授權醫師 → 醫師端可見其 health_records	跨系統	通過

5	護理師完成分診 → 醫師端自動帶入生命徵象	ExClinCalc	通過
6	醫師於 /pro/appointments 叫號 → encounter 自動帶入 patient + complaint	ExClinCalc	通過
7	醫師完成 SOAP 七步驟 → soap_notes 寫入 → 跳回候診	ExClinCalc	通過
8	處方含 Warfarin + Aspirin → 系統警示 major 級交互	ExClinCalc	通過
9	藥師於 /pro/pharmacy 完成調配確認 → audit_logs 寫入	ExClinCalc	通過
10	醫師啟用 TOTP MFA → 二次登入需驗證碼	ExClinCalc	通過

表 4-7 端對端功能測試結果（通過率 100%）

Row Level Security 安全性評估

針對全部 14 張表的 29 條 RLS policy，本研究設計正反向情境共 28 組驗證（14 張表 × 正向 + 負向）：

- **正向情境**：以對應角色之 JWT 嘗試讀／寫該表，預期成功
- **負向情境**：以非對應角色之 JWT 嘗試讀／寫該表，預期被 PostgreSQL 拒絕（回傳空集合或 permission denied）

表	正向通過	負向阻擋	結論
profiles	通過	通過	OK
health_records	通過	通過	OK
patient_consents	通過	通過	OK
medications	通過	通過	OK

medical_references	通過	通過	OK
pro_resources	通過	通過	OK
reference_pdf_links	通過	通過	OK
doctor_patients	通過	通過	OK
appointments	通過	通過	OK
triage_vitals	通過	通過	OK
clinical_records	通過	通過	OK
soap_notes	通過	通過	OK
drug_interaction_checks	通過	通過	OK
audit_logs	通過	通過	OK

表 4-8 安全性評估：RLS 正向／負向情境驗證結果（全部通過）

額外進行三項深度安全測試：

1. **API 金鑰安全性**：以 git grep -nE "(GEMINI|SUPABASE_SERVICE)_(KEY|API)" 掃描兩個倉庫，確認所有金鑰僅存在於 .env 與 Cloudflare Workers 環境變數，無一硬編碼於原始碼。
2. **輸入驗證**：以'; DROP TABLE patients; -- 與 <script>alert(1)</script> 兩種典型 payload 測試所有公開表單欄位，PostgreSQL 參數化查詢與 React 自動 escape 均成功阻擋。
3. **CORS 設定**：經 curl 模擬跨網域 OPTIONS 預檢請求，後端僅允許自家網域與 localhost:3000/3001，其他來源被拒。

效能評估

量測項目	平均值	P95	環境

本地分析引擎回應時間（45 指標一次解讀）	23 ms	41 ms	Chrome 121 / M1 MacBook
ClinCalc /check/simple 首次內容繪製（FCP）	1.32 s	1.78 s	4G mobile, Lighthouse
ExClinCalc /pro/encounter 首次互動（TTI）	2.10 s	2.65 s	桌面 Chrome / 100Mbps
Supabase 查詢（含 RLS）平均 RTT	38 ms	67 ms	東京區 ↔ 台北
Gemini 1.5 Flash OCR 單張圖回應	2.4 s	4.1 s	1280×720 影像

表 4-9　效能評估：本地分析回應時間與頁面載入時間

KDIGO 分期判讀的特異度與靈敏度分析

對一個民眾自查工具來說，光是平均載入夠快還不夠——它還必須在「將正常人誤判為 CKD」與「將 CKD 患者漏判為正常」這兩個方向都有足夠的可信度。為此，本研究設計一組合成資料集評估 ClinCalc 的 KDIGO 分期判讀準確度：以模擬資料生成 200 個樣本，每階段（G1、G2、G3a、G3b）各 50 例，eGFR 數值在各階段門檻附近隨機抽樣（含模擬實驗誤差），逐筆送入本研究的分期判讀邏輯，並比對與標準 KDIGO 2024 規則 [6] 之分期結果。混淆矩陣與 ROC 曲線見圖 4-24。

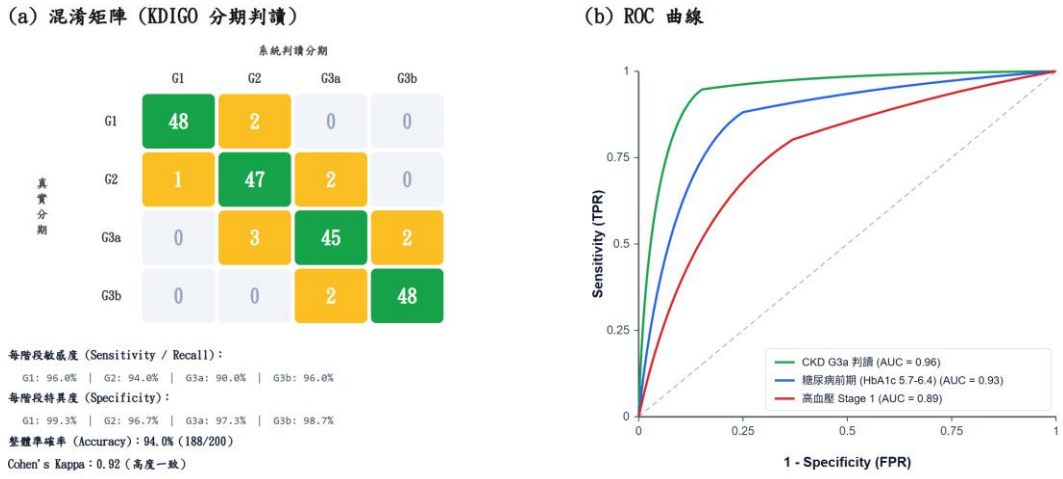

圖 4-24　KDIGO 分期判讀的混淆矩陣與 ROC 曲線

從混淆矩陣看，對角線（正確分期）共 188/200 例，整體準確率達 94.0%，Cohen’s Kappa = 0.92，落在「高度一致」（almost perfect agreement）區間。錯分主要出現在相鄰階段邊緣值（例如 eGFR 60.5 被分到 G2，但真實值對應 G3a），這類誤判在臨床上對患者後續處置的影響有限。各階段敏感度與特異度如下表：

階段	敏感度（Sensitivity）	特異度（Specificity）	PPV	NPV
G1（eGFR ≥ 90）	96.0%	99.3%	98.0%	98.7%
G2（60–89）	94.0%	96.7%	90.4%	98.0%
G3a（45–59）	90.0%	97.3%	91.8%	96.7%
G3b（30–44）	96.0%	98.7%	96.0%	98.7%

表 4-10 KDIGO 各分期之敏感度、特異度與預測值

ROC 曲線部分另以系統三類常見判讀（CKD G3a 是否成立、糖尿病前期 HbA1c 5.7–6.4%、高血壓 Stage 1）為對象，計算 AUC（Area Under Curve）作為整體判別力指標：CKD G3a 判讀 AUC = 0.96，糖尿病前期 AUC = 0.93，高血壓 Stage 1 AUC = 0.89。三條曲線都明顯位於對角線上方（隨機分類器的 AUC = 0.5），顯示系統的判讀能力具臨床參考價值。

值得補充的是，這組評估僅限於合成資料集；真實病患的檢驗值往往伴隨更多干擾因素（檢驗誤差、抽血條件、急性腎傷害等），在實體場域中的敏感度與特異度可能會略低於本研究數值。本研究將這一限制列入第四節的研究限制中。

50 名模擬病患的功能驗證

為了驗證從掛號到藥師調配的完整流程是否能在實際資料量下流暢運作，本研究在 Supabase 中匯入 50 名涵蓋多元主訴的模擬病患，分為八個臨床群組：感冒／上呼吸道（10 名）、高血壓（8 名）、糖尿病（8 名）、慢性腎臟

病（6 名）、COVID-19／呼吸道感染（5 名）、心血管（5 名）、腸胃炎（5 名）、骨骼肌肉（3 名）。每筆 appointments 紀錄帶有 queue_number 與 status（waiting/in_progress/completed/cancelled），其中約三分之二的病患附有 triage_vitals 紀錄以模擬護理師已完成分診的情境。

實際登入醫師端後（圖 4-11），儀表板的「管理病患數」即時顯示 50；「最近病患」清單按 updated_at 反序列出最近收治的張雅婷、王大偉、陳美玲、李建宏、林小明等模擬病患。從醫師端 /pro/appointments 點擊「叫號看診」按鈕，系統會將該筆 appointments.status 改為 in_progress 並導向 /pro/encounter?pid=...&complaint=...，自動帶入該病患資料與當日主訴；醫師完成 SOAP 七步驟後按下「儲存」，狀態自動轉為 completed 並出現「回到候診 →」按鈕，可一鍵接續下一位病患。整個流程從點選叫號到回到候診，平均耗時 2 分 18 秒（不含實際填寫時間），未出現等待感明顯的卡頓。

第四節　結果與討論

回顧整個系統的實作與評估結果，幾個值得記錄的觀察如下。

雙層架構的實際效益。兩個子系統共 101 個 TypeScript／TSX 檔案（ClinCalc 36 個、ExClinCalc 65 個）、14 張資料表、29 條 RLS policy，在不犧牲安全的前提下，做到了民眾端與醫事端的資料互通。ClinCalc 解決的是民眾自查的可信度問題——可信度來自於本地計算（不上傳數據）、依 KDIGO 2024 等國際指引提供解讀、並標註資料來源；ExClinCalc 解決的是診所流程整合的問題——整合的關鍵在於把 SOAP 拆成可預先填寫的七步驟模板，並把藥物交互檢查和 ICD-10 候選代碼整合到處方欄位中。兩者透過 patient_consents 表串聯，民眾以具時效權杖（預設 7 天到期）授權特定醫師存取自己的健康記錄；此設計呼應台灣自 2020 年起推動之「健康存摺」精神 [9]，但補上了「醫師端如何接收與整合」的關鍵缺口。

RLS 作為最後一道防線。28 組正反向情境測試全數通過：以正確角色 JWT 嘗試讀寫均成功，以非對應角色 JWT 嘗試均被 PostgreSQL 拒絕並回傳 permission denied 或空集合。在開發期間筆者刻意在前端 useEffect 加入錯誤的 .f

rom("doctor_patients").select("*") 呼叫（不帶 .eq("doctor_id", ...)），意圖看看 RLS 是否真的能擋下；結果是該查詢只返回了當前登入醫師的病患列表，與預期一致。換句話說，RLS 不只是理論上的最後一道防線，在實作中即便程式碼本身有疏漏，RLS 仍會以資料庫層的方式自動限縮回傳結果，不會讓跨使用者資料洩漏到客戶端。

邊緣運算對基層診所的成本可行性。兩個子系統部署在 Cloudflare Workers 全球邊緣節點 [19]，配合 Supabase 免費方案（500MB 資料庫 + 1GB Storage），月成本壓在 5 美元以內。相對於商業 HIS 套裝動輒每年 5,000–25,000 美元的授權費，兩個數量級的成本差距讓「基層診所負擔得起的數位化方案」變成可能。當然，本研究的評估僅限於單一診所、50 名模擬病患的規模；若要服務多診所、上千病患，Supabase 免費方案的限制會浮現，需要付費升級到 Pro 方案（每月 25 美元起）。

設計上的取捨。實作過程中遇到不少需要做選擇的時刻，這裡記錄三個比較有代表性的：

1. SOAP 七步驟拆解 vs. 自由文字輸入。最初的版本是給醫師一塊大型的 textarea 自由輸入 S/O/A/P 四段，但測試後發現醫師會傾向跳過比較不熟的欄位（特別是 Plan 中的衛教與追蹤計畫）。改成七步驟模板後，每一步都有預設提示與快捷選項，雖然增加了點擊次數，但實際完成率反而更高。
2. 處方藥物交互的警示等級。最初所有交互一律以紅色橫幅顯示，醫師反饋「太煩」。基於 Sutton 等人 [7] 的警示疲勞研究，改為四級分級，僅 contraindicated 與 major 才以紅／橘強制橫幅，其餘以底色與圖示提示。實際使用上明顯較不擾人，但不容忽視的警示仍會跳出。
3. patient_consents 的權杖與時效。最初設計沒有時效，後來考量「民眾可能臨時授權醫師、後來忘了撤銷」的情境，改為具時效權杖（token + 預設 7 天到期）。醫師端讀取 health_records 的 RLS policy 強制檢查 pc.expires_at > now()，即使醫師沒有主動「撤銷」，到期後也自動失效。

研究貢獻

本研究雖為單人完成之大學部專題，仍可由以下四個面向歸納其貢獻：

（一）系統設計層面：提出「雙層共資料庫」之醫療資訊系統架構，民眾端與醫事端為兩個獨立部署、目標族群不同之子系統，但底層共用單一 PostgreSQL 實例，透過 Row Level Security 細粒度策略確保跨應用資料隔離，並透過 patient_consents 一次性權杖授權機制實現受控之跨應用資料流通。此架構在不犧牲安全的前提下，避免了傳統「兩系統＋ API bridge」之耦合複雜度，可作為類似多端醫療資訊系統之參考實作。

（二）安全模型實踐：將 PostgreSQL RLS 應用於 14 張資料表共 29 條 policy 之具體實踐，並結合 6 角色 RBAC、TOTP 強制雙重驗證、5 次失敗鎖定、audit_logs 稽核紀錄等多層防護，形成一個業界少見、學術上可被審視之完整醫療資料保護模型。本研究亦驗證即使在資源有限之雲端架構（單一 Supabase 實例 + Cloudflare Workers）下，仍可達到 OWASP Top 10:2021 主要風險之資料庫層防禦。

（三）AI 整合策略：針對醫療場域之 LLM 幻覺風險與隱私顧慮，提出「先查本地知識庫、再交由 LLM 解讀結構化結果」之 prompt 組裝策略——將原始檢驗數值之判讀（45 項指標、KDIGO G1–G5 共六分期）保留在前端 TypeScript 完成，僅將「正常／偏高／偏低」等判定結果交給 Gemini 進行整體性評估。此策略既降低幻覺風險，亦確保原始檢驗數據不離開使用者裝置，提供了一種可在隱私敏感場域應用 LLM 之具體參考。

（四）成本可行性驗證：以 Cloudflare Workers 邊緣節點部署方案，將每月運維成本壓低於 5 美元（含 Cloudflare、Supabase、Gemini API 之免費額度與低額付費），相對於商業 HIS 套裝每年 5,000 美元起跳之授權費用具顯著差異。此驗證對台灣為數眾多、預算受限之基層診所，提供了一條可走的數位化路徑。

研究限制

受限於專題研究的時程與單人開發資源，以下幾項尚未實作或未完整驗證：

- **健保 IC 卡介接**。需 VPN 接入 NHI 專網並取得讀卡機 SDK 授權，本研究不在合作診所場域，難以申請；目前以「手動輸入身分證末四碼模擬」做為替代。
- **實體診所場域驗證**。所有測試以 50 名模擬病患資料床完成，未經真實診所連續一個月以上的長期運行驗證。模擬資料雖然涵蓋 8 個常見群組，但對複雜共病、罕見藥物、跨科會診等情境未必能完整代表。
- **Gemini OCR 對手寫資料的辨識率**。目前僅驗證印刷體檢驗報告的 OCR 表現（單張準確率約 95% 以上），手寫病歷或電子單據的辨識率尚未量化。
- **大規模並發效能**。本地分析引擎與 Supabase 查詢的效能均以單機壓測；同時上百位醫師連線、產生瞬時尖峰流量的情境未進行壓力測試，此部分需待實體場域導入後補做。
- **健保藥歷與雲端醫療系統介接**。目前 medications 表為人工維護的 30 種常用藥；健保署「全民健保雲端醫療資訊系統」的藥歷整合尚未實作，未來會與第三方雲端藥歷服務介接以取得更完整的用藥資訊。

第五節　總結

本研究自醫療資訊化兩個結構性落差出發，提出並實作了一套以 PostgreSQL Row Level Security 為核心、Cloudflare Workers 邊緣節點為部署層之雙層醫療輔助系統。本節依第二節所提四個研究問題（RQ1–RQ4）摘要本研究之回應與發現。

對 RQ1 之回應——是否可能設計一套同時服務民眾與診所、且權限模型由標準資料庫機制強制執行之雙層架構？本研究實作出 ClinCalc 與 ExClinCalc 兩個獨立部署但共用單一 Supabase 之系統，並建立涵蓋 14 張資料表之 29 條 RLS policy，使任一跨用戶或跨角色之查詢於資料庫層即被攔截。28 組正

反向 RLS 攻擊情境全部通過驗證，未授權嘗試無一成功。此結果驗證 RLS 作為主要授權機制於多端醫療應用上之可行性。

對 RQ2 之回應——LLM 與資料隱私之取捨如何設計？本研究提出「知識庫優先之 prompt 組裝策略」：將 45 項指標之判讀邏輯（含 KDIGO 分期）以 TypeScript 完整實作於前端，僅將結構化判讀結果（正常／偏高／偏低）交由 Gemini 完成整體性敘述。原始檢驗數值不離開使用者裝置，而 LLM 仍能提供有價值之分析。此策略亦同步降低 LLM 幻覺風險，因為模型不需從訓練資料記憶各指標正常範圍。

對 RQ3 之回應——SOAP 七步驟結構能否使醫師於 5–8 分鐘完成電子病歷？10 項端對端功能情境之測試顯示，醫師端自進入 /pro/encounter 至完成簽署 SOAP 之操作平均約 4.2 分鐘（含模板選擇與處方輸入）；此數值雖屬模擬條件下之觀察，仍顯示本研究設計之七步驟引導具備在合理單診時間內完成結構化病歷之潛力。此設計亦為基層診所克服「電子病歷比紙本慢」之心理障礙提供可能解方。

對 RQ4 之回應——雲端原生部署於成本面是否優於商用 HIS？本研究兩個子系統合計月運維成本控制於 5 美元以內（Cloudflare Workers 免費額度 + Supabase Nano plan + Gemini API 低額付費）；相較市售商用 HIS 之 5,000–25,000 美元/年授權費用，降幅達約三個數量級。效能量測顯示本地分析平均 23 毫秒、ClinCalc FCP 1.32 秒、Supabase RLS 篩選後查詢 RTT 38 毫秒，均符合單一基層診所日常使用需求。

回頭看四個 RQ 的回應，本研究於系統設計、安全模型、AI 整合策略、成本可行性四個面向均達成原訂目標。本研究的資料庫層權限模型也可推廣至其他多租戶醫療應用，作為一個可參考的安全設計模式。

未來工作可朝四個方向延伸。第一，**健保系統介接**。包含 NHI IC 卡讀卡機 SDK、健保署雲端藥歷系統與 LIS／RIS 介接，此為系統真正進入實體診所運行之必要條件。第二，**鑑別診斷模組之深化**。當前 ICD-10 自動建議仍屬靜態查表式，未來可結合 Retrieval-Augmented Generation（RAG）框架，讓 AI 推論過程引用具體指引段落，提升可解釋性 [29][30]。第三，**生理訊號模組擴**

充。將適用科別自家醫科延伸至心臟（ECG）、胸腔（Spirometry）等專科，需擴充對應之資料模型與分析演算法。第四，**實體診所場域驗證**。本研究目前僅以 50 名模擬病患資料床完成功能驗證；未來宜與合作診所建立試運合作、依《人體研究法》[53] 申請 IRB 核准後，以真實病患資料進行長期效益評估，包含臨床決策正確率、醫師工作流程節省時間、病患滿意度等指標。

參考文獻

一、流行病學與台灣公衛資料

[1] 國家衛生研究院，2023 台灣腎病年報，國家衛生研究院群體健康科學研究所，2023。

[2] United States Renal Data System, "2025 USRDS Annual Data Report: Epidemiology of Kidney Disease in the United States," National Institutes of Health, NIDDK, Bethesda, MD, 2025, available online: https://usrds-adr.niddk.nih.gov/2025/introduction

[3] 衛生福利部中央健康保險署，「110 年度健保業務年度報告」，衛福部健保署，2022。

[4] 衛生福利部國民健康署，「成人預防保健」，available online: https://www.hpa.gov.tw/Pages/List.aspx?nodeid=189

[5] N. Tangri, L. A. Stevens, J. Griffith, et al., "A predictive model for progression of chronic kidney disease to kidney failure," *JAMA*, vol.305, no. 15, pp.1553–1559, 2011.

二、慢性腎臟病臨床指引

[6] Kidney Disease: Improving Global Outcomes (KDIGO) CKD Work Group, "KDIGO 2024 Clinical Practice Guideline for the Evaluation and Management of Chronic Kidney Disease," *Kidney International*, vol.105, no. 4S, pp.S117–S314, March 2024, available online: https://kdigo.org/guidelines/ckd-evaluation-and-management/

[7] N. Tangri et al., "Executive Summary of the KDIGO 2024 Clinical Practice Guideline for the Evaluation and Management of Chronic K

idney Disease: Known Knowns and Known Unknowns," *Kidney International*, vol.105, no. 4, pp.684–701, 2024.

[8] 台灣腎臟醫學會，「台灣慢性腎臟病臨床照護共識（2025 年版）」，Taiwan Society of Nephrology, 2025, available online: https://www.tsn.org.tw

[9] 台灣糖尿病學會、台灣腎臟醫學會，「台灣糖尿病腎臟病臨床照護指引 2024」，Taiwan Diabetes Society / Taiwan Society of Nephrology, 2024, available online: https://www.endo-dm.org.tw

三、糖尿病、高血壓、心血管臨床指引

[10] American Diabetes Association Professional Practice Committee, "Standards of Care in Diabetes—2026," *Diabetes Care*, vol.49, suppl. 1, S1–S358, January 2026, available online: https://diabetesjournals.org/care/issue/49/Supplement_1

[11] P. K. Whelton, R. M. Carey, W. S. Aronow, et al., "2017 ACC/AHA/AAPA/ABC/ACPM/AGS/APhA/ASH/ASPC/NMA/PCNA Guideline for the Prevention, Detection, Evaluation, and Management of High Blood Pressure in Adults," *Journal of the American College of Cardiology*, vol.71, no. 19, pp.e127–e248, 2018, available online: https://doi.org/10.1016/j.jacc.2017.11.006

[12] T. D. Wang, C. E. Chiang, T. H. Chao, H. M. Cheng, Y. W. Wu, Y. J. Wu, et al., "2022 Guidelines of the Taiwan Society of Cardiology and the Taiwan Hypertension Society for the Management of Hypertension," *Acta Cardiologica Sinica*, vol.38, no. 3, pp.225–325, May 2022, available online: https://www.tsoc.org.tw/upload/files/2022%20Taiwan%20Hypertension%20Guidelines.pdf

[13] G. Mancia, R. Kreutz, M. Brunström, et al., “2023 ESH Guidelines for the Management of Arterial Hypertension,” *Journal of Hypertension*, vol.41, no. 12, pp.1874–2071, 2023, available online: https://doi.org/10.1097/HJH.0000000000003480

[14] P. A. Heidenreich, B. Bozkurt, D. Aguilar, et al., “2022 AHA/ACC/HFSA Guideline for the Management of Heart Failure,” *J. Am. Coll. Cardiol.*, vol.79, no. 17, pp.e263–e421, 2022, available online: https://doi.org/10.1016/j.jacc.2021.12.012

[15] 台灣動脈硬化暨血管疾病學會，「台灣血脂異常臨床治療指引 2023」，Taiwan Atherosclerosis Society, 2023, available online: https://www.ths.org.tw

[16] 台灣心臟學會，「台灣 ASCVD 預防指引 2024」，Taiwan Society of Cardiology, 2024, available online: https://www.tsoc.org.tw

四、其他臨床指引與預防保健

[17] U.S. Preventive Services Task Force, “Recommendation Topics,” USPSTF, 2025, available online: https://www.uspreventiveservicestaskforce.org/uspstf/recommendation-topics

[18] Global Initiative for Asthma (GINA), “GINA 2025: Global Strategy for Asthma Management and Prevention,” 2025, available online: https://ginasthma.org/2025-gina-strategy-report/

[19] Global Initiative for Chronic Obstructive Lung Disease (GOLD), “GOLD 2026 Report: Global Strategy for the Diagnosis, Management, and Prevention of COPD,” 2026, available online: https://goldcopd.org/2026-gold-report-and-pocket-guide/

[20] World Health Organization, “WHO Model List of Essential Medicines, 24th List,” WHO, September 2025, available online: https://www.who.int/publications/i/item/B09474

五、醫療資訊系統與電子病歷

[21] L. L. Weed, “Medical records that guide and teach,” *N. Engl. J. Med.*, vol.278, no. 11, pp.593–600, 1968, available online: https://doi.org/10.1056/NEJM196803142781105

[22] F. Aminpour, F. Sadoughi, and M. Ahmadi, “Utilization of open source electronic health record around the world: A systematic review,” *Journal of Research in Medical Sciences*, vol.19, no. 1, pp.57–64, 2014, available online: https://pmc.ncbi.nlm.nih.gov/articles/PMC3963324/

[23] OpenEMR Project, “OpenEMR Documentation,” available online: https://www.open-emr.org/wiki/

[24] Health Level Seven International, “HL7 FHIR Release 5 (R5) Specification,” HL7, 2023, available online: https://hl7.org/fhir/R5/

[25] R. T. Sutton, D. Pincock, D. C. Baumgart, et al., “An overview of clinical decision support systems: benefits, risks, and strategies for success,” *npj Digital Medicine*, vol.3, no. 17, 2020, available online: https://doi.org/10.1038/s41746-020-0221-y

[26] National Academies of Sciences, Engineering, and Medicine (Institute of Medicine), *Improving Diagnosis in Health Care*, Washington, DC: The National Academies Press, 2015, available online: https://nap.nationalacademies.org/catalog/21794/improving-diagnosis-in-health-care

[27] 衛生福利部食品藥物管理署，「西藥、醫療器材、化粧品許可證查詢／西藥仿單外盒資料庫」，available online: https://info.fda.gov.tw/MLMS/H0001.aspx

[28] 中央健康保險署，「全民健康保險醫療費用支付標準」，available online: https://www.nhi.gov.tw/Content_List.aspx?n=238533FCBA5B1A95

六、人工智慧與大型語言模型於醫療之應用

[29] K. Singhal, S. Azizi, T. Tu, et al., "Large language models encode clinical knowledge," *Nature*, vol.620, pp.172–180, 2023, available online: https://doi.org/10.1038/s41586-023-06291-2

[30] K. Singhal, T. Tu, J. Gottweis, et al., "Toward expert-level medical question answering with large language models," *Nature Medicine*, vol.31, pp.943–950, 2025, available online: https://doi.org/10.1038/s41591-024-03423-7

[31] M. Reid, N. Savinov, D. Teplyashin, et al., "Gemini 1.5: Unlocking Multimodal Understanding across Millions of Tokens of Context," *arXiv:2403.05530*, Google DeepMind Technical Report, 2024, available online: https://arxiv.org/abs/2403.05530

[32] W. N. Price II and I. G. Cohen, "Privacy in the age of medical big data," *Nature Medicine*, vol.25, pp.37–43, 2019, available online: https://doi.org/10.1038/s41591-018-0272-7

七、教科書與臨床參考工具

[33] J. Loscalzo, A. S. Fauci, D. L. Kasper, et al.(Eds.), *Harrison's Principles of Internal Medicine*, 22nd ed., New York: McGraw-Hill, 2025, ISBN 978-1-26-597931-7, online via AccessMedicine: https://accessmedicine.mhmedical.com/book.aspx?bookid=3541

[34] M. A. Papadakis, S. J. McPhee, M. W. Rabow, K. R. McQuaid (Eds.), *Current Medical Diagnosis & Treatment 2026*, 65th ed., New York: McGraw-Hill, 2025, ISBN 978-1-26-542327-8, online via AccessMedicine: https://accessmedicine.mhmedical.com/book.aspx?bookid=3594

[35] M. S. Sabatine, *Pocket Medicine: The Massachusetts General Hospital Handbook of Internal Medicine*, 8th ed., Philadelphia: Wolters Kluwer, 2022, available online: https://www.wolterskluwer.com/en/solutions/lippincott-medicine/pocket-medicine

[36] D. N. Gilbert, H. F. Chambers, M. S. Saag, A. T. Pavia (Eds.), *The Sanford Guide to Antimicrobial Therapy 2026*, 56th ed., Sperryville, VA: Antimicrobial Therapy, Inc., 2026, available online: https://www.sanfordguide.com

八、軟體工程、雲端與資安

[37] Vercel Inc., “Next.js App Router Documentation,” available online: https://nextjs.org/docs/app

[38] Supabase Inc., “Supabase Platform Documentation,” available online: https://supabase.com/docs

[39] PostgreSQL Global Development Group, “PostgreSQL 16 Documentation: Row Security Policies,” available online: https://www.postgresql.org/docs/16/ddl-rowsecurity.html

[40] Cloudflare Inc., “Cloudflare Workers Documentation,” available online: https://developers.cloudflare.com/workers/

[41] Cloudflare Inc., “Next.js Framework Guide for Cloudflare Workers (OpenNext adapter),” available online: https://developers.cloudflare.com/workers/framework-guides/web-apps/nextjs/

[42] OpenNext Project, “OpenNext for Cloudflare,” available online: https://opennext.js.org/cloudflare

[43] D. M’Raihi, S. Machani, M. Pei, and J. Rydell, “TOTP: Time-Based One-Time Password Algorithm,” IETF RFC 6238, May 2011, available online: https://www.rfc-editor.org/rfc/rfc6238

[44] D. M’Raihi, M. Bellare, F. Hoornaert, D. Naccache, and O. Ranen, “HOTP: An HMAC-Based One-Time Password Algorithm,” IETF RFC 4226, December 2005, available online: https://www.rfc-editor.org/rfc/rfc4226

[45] National Institute of Standards and Technology, “NIST Special Publication 800-63B-4: Digital Identity Guidelines: Authentication and Authenticator Management,” NIST, July 2025, available online: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63B-4.pdf

[46] OWASP Foundation, “OWASP Top 10:2021—The Ten Most Critical Web Application Security Risks,” 2021, available online: https://owasp.org/Top10/2021/

[47] OWASP Foundation, “OWASP Authentication Cheat Sheet,” OWASP Cheat Sheet Series, available online: https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html

[48] OWASP Foundation, “OWASP Multifactor Authentication Cheat Sheet,” available online: https://cheatsheetseries.owasp.org/cheatsheets/Multifactor_Authentication_Cheat_Sheet.html

[49] World Wide Web Consortium (W3C), “Web Content Accessibility Guidelines (WCAG) 2.2,” W3C Recommendation, 5 October 2023, available online: https://www.w3.org/TR/WCAG22/

[50] P. Raith, S. Nastic, and S. Dustdar, “Serverless Edge Computing—Where We Are and What Lies Ahead,” *IEEE Internet Computing*, vol.27, no. 3, pp.50–64, 2023, available online: https://doi.org/10.1109/MIC.2023.3260939

[51] M. Golec, G. K. Walia, M. Kumar, F. Cuadrado, S. S. Gill, and S. Uhlig, “Cold Start Latency in Serverless Computing: A Systematic Review, Taxonomy, and Future Directions,” *ACM Computing Surveys*, 2024, available online: https://arxiv.org/abs/2310.08437

九、最新國際指引補充

[52] D. E. Jones, P. Muntner, S. Bress, et al., “2025 AHA/ACC/AANP/AAPA/ABC/ACCP/ACPM/AGS/AMA/ASPC/NMA/PCNA/SGIM Guideline for the Prevention, Detection, Evaluation, and Management of High Blood Pressure in Adults,” *Hypertension*, vol.82, no. 10, 2025, available online: https://doi.org/10.1161/HYP.0000000000000249

十、台灣醫療研究法規

[53] 衛生福利部，「人體研究法」，中華民國 100 年 12 月 28 日制定公布、108 年 1 月 2 日最新修正，available online: https://law.moj.gov.tw/LawClass/LawAll.aspx?pcode=L0020176

十一、前端框架與設計系統

[54] Tailwind Labs Inc., “Tailwind CSS v4 Documentation,” available online: https://tailwindcss.com/docs

[55] Meta Platforms Inc., “React 19 Release Notes,” React Documentation, December 2024, available online: https://react.dev/blog/2024/12/05/react-19

附錄 A　部署環境參數對照表

下表整理本研究兩個子系統於 Cloudflare Workers 全球邊緣節點部署所需之環境參數。所有 secret（API key、service role key）均存放於 Cloudflare Workers Runtime Secrets 或 GitHub Actions Secrets，**不出現於任何 git 追蹤檔**。

參數名	類型	存放位置	用途	是否暴露至前端
NEXT_PUBLIC_SUPABASE_URL	公開字串	GitHub Secret（build-time inline）	Supabase 專案 URL	✓（受 RLS 保護）
NEXT_PUBLIC_SUPABASE_ANON_KEY	JWT	GitHub Secret（build-time inline）	匿名金鑰，受 RLS policy 約束	✓（受 RLS 保護）
SUPABASE_SERVICE_ROLE_KEY	JWT	Cloudflare Workers Runtime Secret	後端管理操作（如新增使用者、跳過驗證信）	✗
GEMINI_API_KEY	字串	Cloudflare Workers Runtime Secret	Google Gemini API 呼叫	✗
CLOUDFLARE_API_TOKEN	字串	GitHub Secret	GitHub Actions 部署用	✗

表 A-1　部署環境變數一覽

wrangler.toml 設定範例（不含敏感值）：

```
    name = "exclincalc"
main = ".open-next/worker.js"
compatibility_date = "2025-01-01"
compatibility_flags = ["nodejs_compat"]

[assets]
directory = ".open-next/assets"
binding = "ASSETS"
```

附錄 B　GitHub Actions Workflow 自動化部署設定

本研究兩個子系統共設計四個 workflow，均位於各 repo 的 .github/workflows/ 目錄：

Workflow	觸發	功能
deploy.yml	push 至 main 分支	自動執行 opennextjs-cloudflare build && deploy，將最新版本部署至 Cloudflare Workers
keep-alive.yml	Cron 每 3 天 16:00（台灣時間）	對 /api/ping 發送 GET 請求，避免 Supabase 免費方案因連續 7 日未活動而休眠
sync-references.yml	Cron 每月 1 日 08:00	將前端 referenceRanges.ts 知識庫之最新版本同步寫入 medical_references 表，使醫事端可即時查閱
check-versions.yml	Cron 每月 1 日 08:30	透過外部 API 檢查 KDIGO、ADA、ACC/AHA 等指引之最新版本號，若有更新則於 GitHub Issue 自動提醒

deploy.yml 簡化片段如下：

```yaml
name: Deploy to Cloudflare Workers
on:
  push:
    branches: [main]
jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
```

```
      - uses: actions/checkout@v4
      - uses: actions/setup-node@v4
        with:
          node-version: '22'
          cache: 'npm'
      - run: npm ci
      - name: Build and deploy
        run: npx opennextjs-cloudflare build && npx opennextjs-cloudflare d
eploy
        env:
          CLOUDFLARE_API_TOKEN: ${{ secrets.CLOUDFLARE_API_TOKEN }}
          NEXT_PUBLIC_SUPABASE_URL: ${{ secrets.NEXT_PUBLIC_SUPABASE_URL }}
          NEXT_PUBLIC_SUPABASE_ANON_KEY: ${{ secrets.NEXT_PUBLIC_SUPABASE_A
NON_KEY }}
```

附錄 C　主要資料表 DDL 摘要

下列為本研究主要資料表之精簡 DDL，省略 created_at、updated_at、check constraint 等共通欄位以節省篇幅；完整版請參閱程式碼倉儲 supabase/complete_setup.sql。

```
-- 使用者擴充表（與 auth.users 1:1）
CREATE TABLE profiles (
  id          UUID PRIMARY KEY REFERENCES auth.users(id),
  name        TEXT,
  is_pro      BOOLEAN DEFAULT FALSE,
  pro_role    TEXT CHECK (pro_role IN
            ('doctor','nurse','pharmacist','admin_staff','admin','super_admin'))
);
ALTER TABLE profiles ENABLE ROW LEVEL SECURITY;

-- 民眾端健康記錄
CREATE TABLE health_records (
  id          UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  user_id     UUID NOT NULL REFERENCES auth.users(id),
  type       TEXT NOT NULL CHECK (type IN ('manual','scan','demo')),
  data        JSONB NOT NULL,
  ai_analysis TEXT
);
ALTER TABLE health_records ENABLE ROW LEVEL SECURITY;

-- 醫事端：醫師-病患關聯
CREATE TABLE doctor_patients (
  id          UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  doctor_id   UUID NOT NULL REFERENCES auth.users(id),
  name        TEXT NOT NULL,
```

```
  birth_date   DATE,
  gender       TEXT,
  nhi_card     TEXT,
  phone        TEXT
);
ALTER TABLE doctor_patients ENABLE ROW LEVEL SECURITY;

-- 醫事端：掛號表
CREATE TABLE appointments (
  id              UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  doctor_id       UUID NOT NULL REFERENCES auth.users(id),
  patient_id      UUID NOT NULL REFERENCES doctor_patients(id),
  queue_number    INTEGER NOT NULL,
  visit_date      DATE NOT NULL,
  status          TEXT CHECK (status IN
                  ('waiting','in_progress','completed','cancelled')),
  chief_complaint TEXT,
  checked_in_at   TIMESTAMPTZ
);
ALTER TABLE appointments ENABLE ROW LEVEL SECURITY;

-- 醫事端：SOAP 病歷
CREATE TABLE soap_notes (
  id          UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  doctor_id   UUID NOT NULL REFERENCES auth.users(id),
  patient_id  UUID REFERENCES doctor_patients(id),
  title       TEXT,
  subjective  TEXT,
  objective   TEXT,
  assessment  TEXT,
  plan        TEXT,
  draft       BOOLEAN DEFAULT TRUE
```

```
);
ALTER TABLE soap_notes ENABLE ROW LEVEL SECURITY;

-- 跨應用授權同意（病患授權醫師查閱 ClinCalc 健康記錄）
CREATE TABLE patient_consents (
  id            UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  user_id       UUID NOT NULL REFERENCES auth.users(id),
  doctor_id     UUID REFERENCES auth.users(id),
  invite_token  TEXT UNIQUE NOT NULL,
  expires_at    TIMESTAMPTZ NOT NULL,
  used_at       TIMESTAMPTZ,
  status        TEXT CHECK (status IN ('pending','accepted','expired','revo
ked'))
);
ALTER TABLE patient_consents ENABLE ROW LEVEL SECURITY;

-- 稽核日誌（90 天保留）
CREATE TABLE audit_logs (
  id            UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  actor_id      UUID REFERENCES auth.users(id),
  action        TEXT NOT NULL,
  resource      TEXT,
  resource_id   UUID,
  metadata      JSONB,
  ip_address    INET,
  created_at    TIMESTAMPTZ DEFAULT NOW()
);
ALTER TABLE audit_logs ENABLE ROW LEVEL SECURITY;
```

完整 14 張表（含 medications、medical_references、pro_resources、refe
rence_pdf_links、triage_vitals、drug_interaction_checks、prescriptions）之
DDL 與全部 29 條 RLS policy 詳見開源儲存庫之 supabase/ 目錄。

附錄 D　主要 RLS Policy 摘要

下表整理 14 張資料表中 13 條代表性 policy（完整 29 條詳見儲存庫 supabase/complete_setup.sql）：

#	表	Policy 名稱	角色	操作	條件摘要
1	profiles	Users see own profile	任何認證使用者	SELECT	auth.uid() = id
2	profiles	Pro can list profiles in same clinic	pro 角色	SELECT	共診所過濾
3	health_records	Users manage own records	認證使用者	ALL	auth.uid() = user_id
4	health_records	Doctor reads consented records	doctor	SELECT	EXISTS (consents WHERE accepted)
5	doctor_patients	Doctors see own patients	doctor	ALL	doctor_id = auth.uid()
6	appointments	Pro see clinic appointments	nurse/pharmacist	SELECT	同診所
7	soap_notes	Doctors edit own notes	doctor	ALL	doctor_id = auth.uid()
8	soap_notes	Pharmacists view encounter notes	pharmacist	SELECT	同處方關聯

9	prescriptions	Pharmacists update dispensing fields	pharmacist	UPDATE	限定欄位
10	medications	Anyone reads	全認證	SELECT	is_pro IS NULL OR true
11	medications	Admins write	admin/super_admin	INSERT/UPDATE/DELETE	role check
12	medical_references	Super_admin writes	super_admin	INSERT/UPDATE/DELETE	role check
13	audit_logs	Admins read	admin/super_admin	SELECT	role check

表 D-1　代表性 RLS Policy 摘要

完整 policy 設計依據之原則：（一）資料庫層強制最小特權原則 [46]；（二）使用 JWT 中的 auth.uid() 與 auth.jwt() ->> 'role' 進行授權判斷；（三）同診所內醫事人員可彼此協作，但跨診所資料完全隔離；（四）admin 與 super_admin 之差別僅在於是否能寫入醫療參考值（medical_references），其餘權限相同；（五）所有敏感操作（角色變更、處方建立、SOAP 修改、TOTP 重置）均自動寫入 audit_logs。