

國立中央大學資訊管理系

軟體系統測試計畫書 (Test Plan Report)

專案名稱：Exchange Platform（卡片交換平台）

版本：1.0

測試階段：系統測試 / 驗收測試

報告日期：2025/12/12

第八組

114423020 陳欣妤

114423037 蕭筑云

114423051 張馨麒

114423068 廖承偉

指導教授：許智誠教授

目錄

1. 測試策略與範圍 (Strategy & Scope).....	3
1.1 總覽與風險評估.....	3
1.2 測試策略總則.....	3
1.3 測試層級規劃.....	3
A. 單元測試 (Unit Testing).....	3
B. 整合測試 (Integration Testing)	3
C. 系統與驗收測試.....	4
2. 核心功能測試設計 (Functional Testing).....	5
2.1 需求追溯矩陣 (RTM).....	5
2.2 測試模組矩陣 (精選).....	5
認證與帳戶	5
刊登與搜尋	5
交換提案與聊天	5
物流與送達	6
3. Web 架構與結構完整性測試 (Web Architecture).....	7
3.1 靜態結構測試.....	7
3.2 覆蓋率指標.....	7
3.3 Define-Use 測試路徑範例	7
4. 系統非功能性測試計畫 (Non-Functional).....	8
4.1 安全性測試 (Security).....	8
4.2 恢復力測試 (Recovery).....	8
4.3 壓力與效能測試 (Stress & Performance)	8
5. 測試執行流程手冊 (Execution Process)	9
5.1 執行階段 (Phases)	9
5.2 結束標準 (Completion Criteria)	9
5.3 角色責任	9

1. 測試策略與範圍 (Strategy & Scope)

1.1 總覽與風險評估

本系統架構為 Java Spring MVC/Spring Boot 後端，前端採用 JSP/Thymeleaf 模板與靜態資源，並包含 Python 物流追蹤工具與外部 API 整合。

✚ **主要模組：**使用者認證（含 OAuth）、刊登與搜尋、交換流程（提案/協商/出貨/追蹤/送達）、郵件通知。

✚ **風險重點：**

- **安全性：**表單輸入注入（SQL/XSS）、權限越權（ID 操弄）。
- **整合性：**JSP 與 Controller 資料綁定、外部服務（SMTP/OAuth）故障處理。
- **效能：**搜尋列表的 N+1 查詢問題、聊天室高併發 I/O。
- **穩健性：**長鏈路業務狀態機（提案至評價）的一致性與歷史資料修補。

1.2 測試策略總則

✚ **覆蓋面：**核心邏輯進行單元測試；API 與 JSP 綁定進行整合測試；系統層執行安全/效能測試；最終以 SRS 進行端到端驗收。

✚ **Mock/Stub 原則：**外部依賴（SMTP/Google OAuth/物流 HTTP）均以 Stub 取代；資料層測試使用 In-memory DB 或交易回滾。

✚ **覆蓋度度量：**核心服務分支覆蓋率 $\geq 80\%$ ；安全測試覆蓋 OWASP Top 10 高風險項。

1.3 測試層級規劃

A. 單元測試 (Unit Testing)

✚ **範圍：**ProposalService, ChatService, ShipmentService, DeliveryService 及共用 Utils。

✚ **重點案例：**

- 驗證禁止與自身物品交換 (ERR_SELF_SWAP_FORBIDDEN)。
- 聊天室封閉後禁止發送訊息。
- 物流查詢超時需啟動退避 (exponential backoff)。

B. 整合測試 (Integration Testing)

✚ **範圍：**前端 JSP ↔ 後端 Controller 資料綁定、REST API 權限與回應。

✚ **重點案例：**

- 搜尋頁面分頁參數綁定與模型屬性檢查。
- OAuth 登入拒絕授權或回調錯誤時的導向邏輯。

- 出貨事件序列逆序更新的阻擋驗證。

C. 系統與驗收測試

- ✚ **系統測試**：涵蓋安全性（SQLi/XSS）、效能（搜尋 P95 < 500ms）、壓力（物流服務故障注入）。
- ✚ **驗收測試 (E2E)**：覆蓋 UC-01 至 UC-08 端到端流程（如：使用者登入→建立刊登→提案→協商→出貨→送達）。

2. 核心功能測試設計 (Functional Testing)

本章節針對核心功能（UC-01~UC-08）設計詳細測試矩陣。

2.1 需求追溯矩陣 (RTM)

需求 ID	功能名稱	對應 UC	優先權
R1/R2	使用者登入/登出/OAuth	UC-01	High
R4/R5	刊登 CRUD 與搜尋篩選	UC-02/03	High
R6/R7	交換提案與協商聊天	UC-04/05	High
R8/R9	配送設定與物流追蹤	UC-06/07	High
R10	送達確認	UC-08	High

2.2 測試模組矩陣 (精選)

認證與帳戶

測試編號	情境	預期結果	測試方法
TC-AU01	成功登入 (帳密)	200; 導向首頁; 審計記錄登入	Use Case
TC-AU05	OAuth 成功	302 導向首頁; 建立綁定會話	Use Case
TC-AU10	安全性: 暴力嘗試	429 或鎖定策略; 審計告警	Security

刊登與搜尋

測試編號	情境	預期結果	測試方法
TC-LS03	非法圖片格式	400; 提示錯誤; 不儲存	等價劃分
TC-SR03	非法排序欄位	400; 拒絕非法欄位 (SQLi 防護)	Security

交換提案與聊天

測試編號	情境	預期結果	測試方法
TC-PR02	自提案禁止	400/403; 錯誤碼 <code>ERR_SELF_SWAP</code>	等價劃分
TC-CH02	房間封閉	403; 拒絕送出訊息	業務邏輯
TC-CH04	XSS 嘗試	輸出編碼; 不執行腳本	Security

物流與送達

測試編號	情境	預期結果	測試方法
TC-TR01	追蹤事件序列	狀態機依序更新; 逆序拒絕	Use Case
TC-TR03	查詢超時重試	退避延遲遞增; 不阻塞主流程	恢復力
TC-DL01	單方先確認	狀態 awaiting-counterparty ; 觸發提醒	Use Case

3. Web 架構與結構完整性測試 (Web Architecture)

本章節專注於前端模板結構、連結有效性與資料流完整性。

3.1 靜態結構測試

- ✚ 孤兒頁面 (Orphan Pages)：掃描 `templates/*.html`，找出未被 `Controller` 映射或無內部連結指向的頁面。
- ✚ 失效連結 (Broken Links)：從首頁出發爬取所有 `a[href]` 與 `form[action]`，確保無 404/5xx 錯誤。
- ✚ 執行方式：建議結合 `MockMvc` + `jsoup` (Java) 或 Python 爬蟲進行自動化檢查。

3.2 覆蓋率指標

- ✚ 頁面覆蓋率 (Page Coverage)：100% 載入並驗證每一張模板頁。
- ✚ 連結覆蓋率 (Link Coverage)：100% 觸發站內連結與按鈕。
- ✚ 定義-使用覆蓋率 (Define-Use Coverage)：針對 Session/Model 變數 (如 `currentUser`, `proposalId`)，驗證其 "定義" 與 "使用" 路徑完整。

3.3 Define-Use 測試路徑範例

路徑 ID	流程描述	驗證點
DU-01	登入 (Define Session) → 造訪個人頁面 (Use Session)	確保使用者資料顯示正確
DU-03	設定交貨便代碼 (Define TrackingCode) → 觸發追蹤 (Use Code)	確保事件更新正確
DU-04	A 確認送達 (Define Status=A_confirmed) → B 未確認	確保 UI 顯示等待中

4. 系統非功能性測試計畫 (Non-Functional)

4.1 安全性測試 (Security)

- ✚ 未登入訪問：針對 `/ui/profile`, `/ui/chat` 等頁面，驗證是否導向登入頁或回傳 401/403。
- ✚ XSS 注入：於聊天訊息、刊登描述輸入 `<script>`，驗證輸出是否編碼。
- ✚ SQL 注入與排序：於 `/api/listings` 的 `sort` 參數輸入惡意字串，驗證白名單阻擋機制。

4.2 恢復力測試 (Recovery)

- ✚ DB 中斷：在寫入提案或訊息瞬間斷開 DB 連線，驗證事務回滾與錯誤頁面顯示。
- ✚ 外部服務故障：模擬 SMTP 失敗、`etracking.py` 超時，驗證重試策略與死信記錄。
- ✚ 狀態一致性：系統重啟後，檢查提案與物流狀態機是否無遺失。

4.3 壓力與效能測試 (Stress & Performance)

- ✚ 高併發登入：1000 併發請求 `/api/auth/login`，P95 響應時間需 $\leq 2s$ 。
- ✚ 聊天高頻訊息：每秒 1000 QPS 發送訊息，驗證系統穩定性與記憶體無洩漏。
- ✚ 首頁載入：`/ui/home` 的 TTFB $\leq 500ms$ ，整體載入 < 3 秒。
- ✚ 搜尋效能：針對多條件過濾與排序，驗證索引命中與 P95 < 2 秒。

5. 測試執行流程手冊 (Execution Process)

5.1 執行階段 (Phases)

1. 階段 1：單元測試 (Unit Testing)

- 目標：驗證核心服務與 Class 穩定性。
- 命令：`mvn -q -DskipITs=true test`
- 門檻：通過率 $\geq 95\%$ ，核心覆蓋率 $\geq 80\%$ 。

2. 階段 2：整合測試 (Integration Testing)

- 目標：驗證 API 正確性與 Controller-Template 綁定。
- 命令：`mvn -q -Dit.test=*IT verify`
- 重點：驗證未授權行為 (401/403) 與表單綁定。

3. 階段 3：系統測試 (System Testing)

- 內容：執行全套功能測試 (Ch.2)、Web 架構測試 (Ch.3) 與非功能測試 (Ch.4)。
- 產出：缺陷報告 (Bug Report) 與連結有效性報告。

4. 階段 4：回歸測試 (Regression Test)

- 目標：針對修復缺陷進行重測，確保無副作用 (Regression)。

5.2 結束標準 (Completion Criteria)

- ✚ 功能性：High Priority 案例 100% 通過。
- ✚ 覆蓋率：頁面覆蓋率達到 100%。
- ✚ 缺陷：系統中無 Severity 1 (關鍵) 缺陷殘留。
- ✚ 效能：首頁載入 $< 3s$ ，搜尋回應 $< 2s$ 。

5.3 角色責任

- ✚ 開發工程師：撰寫單元/整合測試，修復缺陷。
- ✚ 測試工程師 (QA)：執行系統測試，出具報告與度量，管理缺陷。
- ✚ 測試架構師 (TA)：制定策略、維護工具與審核覆蓋率。