

C端平台 MVP

文件版本: 1.0

發布日期: 2025年10月11日

作者: Ming, Via (AI 輔助生成)

1. 概述 (Overview)

1.1 產品願景與核心戰略

- 產品願景: twin3 旨在成為 GenAI 時代下, 領先的「真實人類體驗即服務 (HaaS - Human Experience as a Service)」市集。我們相信, 當 AI 自動化了內容生產後, 稀缺、可驗證的「人類體驗」本身將成為下一種高價值資產。我們的使命是將這種價值代幣化, 賦予每位用戶數據主權, 讓他們能將獨特的個人體驗變現。
- 核心戰略: 我們的核心護城河是 **Twin Matrix**——一個用於量化「真實性證明 (Proof-of-Authenticity)」的協議。本專案 (使用者端 MVP) 的戰略目標是驗證一個正向的成長飛輪 (**Growth Flywheel**): 透過有吸引力的任務獎勵, 驅動用戶完成「任務」, 在過程中自然地、有動機地貢獻數據, 從而豐富他們的 **Twin Matrix**; 一個更豐富的 **Twin Matrix** 又能解鎖更高價值的任務, 形成一個可持續的、自我強化的生態閉環。

1.2 MVP 階段目標

1. 驗證核心循環: 跑通「用戶發現任務 → 為解鎖任務而完善資料 → 執行任務 → 獲得獎勵與聲譽成長」的最小可行閉環。
2. 聚焦首個場景: 以「產品體驗測試 (**Product Experience Testing**)」, 特別是需要用戶螢幕錄影並上傳的任務類型, 作為首個灘頭堡, 以驗證我們處理非結構化數據的能力。
3. 建立使用者基礎: 在 Q4 2025 結束前, 達成 5,000+ 活躍使用者, 並獲得初步的產品市場契合度信號 (Initial Product-Market Fit signals)。

1.3 目標受眾 (Target Audience)

- 主要受眾 (**Persona**): "Ethan"
 - 身份: 20-35 歲的數位原生代, 活躍於 Web2 (Instagram, YouTube) 和 Web3 (Discord, X) 社群。可能是學生、上班族或自由工作者。
 - 動機:
 - 主要動機 (**Extrinsic**): 尋求利用零碎時間賺取額外收入的機會。對加密貨幣獎勵 (\$twin3 Token) 抱有期待。

- 次要動機 (Intrinsic): 樂於嘗試新產品, 並希望自己的觀點和數據能被量化、被認證, 從而建立一份獨特的數位履歷。
-
- 行為: 習慣於遊戲化的升級系統, 願意為了「解鎖」更高階的獎勵而付出努力。

2. 核心使用者故事 (Core User Stories)

- 核心總故事 (Epic):

「作為一個 twin3 平台的使用者, 我的主要目標是透過完成任務來賺取獎勵。我會策略性地完善我的 Twin Matrix, 因為我知道這是我解鎖更多、更高價值任務的唯一途徑, 也是我數位聲譽成長的證明。」

- 關鍵子故事 (User Stories):

- 探索與發現 (Discovery): 「我想要一個能讓我快速找到當前可接取的、獎勵最高的任務, 並能清晰地告訴我解鎖下一個等級的任務還需要什麼條件的市場。」
- 評估與決策 (Evaluation): 「我想要清晰地權衡任務的獎勵與完成它 (包括可能需要的額外驗證) 所需的時間和精力成本, 以做出最高投資回報率的決策。」
- 參與與執行 (Participation): 「我想要一個清晰、有引導的流程來完成任務並提交我的成果。」
- 追蹤與成長 (Tracking & Growth): 「我想要一個儀表板, 能讓我直觀地看到我的總收益, 並能清楚地展示我的 Twin Matrix 成長如何直接轉化為未來賺取更多收入的能力。」

3. 產品功能規格 (Functional Requirements)

3.1 資訊架構與導覽 (Information Architecture & Navigation)

- FR-1.0: 產品必須採用底部標籤導覽 (Bottom Tab Navigator) 作為主體架構。
- FR-1.1: 底部導航欄必須包含以下四個核心標籤:
 1. (儀表板): 預設首頁, 用於任務管理與狀態追蹤。
 2. (任務市場): 用於發現新任務。
 3. (我的代理): 用於管理個人化 AI 代理。
 4. (我的): 整合式的個人中心入口。

3.2 功能詳述: Dashboard (儀表板)

- FR-2.1: 必須作為用戶登入後的預設首頁。
- FR-2.2: 必須包含一個「我的狀態 (My Status)」元件, 聚合顯示用戶頭像、等級和總收益。此元件應作為進入 Profile 詳細頁的快捷入口。

- **FR-2.3:** 必須包含一個「進行中的任務 (In-Progress Quests)」列表, 展示用戶已接受但未完成的任務, 並提供「繼續任務」的入口。
- **FR-2.4:** 必須包含一個「成長預覽 (Matrix Growth Preview)」元件, 以視覺化方式(如進度條或迷你圖表)展示 Twin Matrix 的整體完成度。此元件應作為進入完整 Matrix 頁的快捷入口。

3.3 功能詳述: Quests (任務市場)

- **FR-3.1:** 必須以標準化的「任務卡片 (Task Card)」列表形式, 展示所有公開可見的任務。
- **FR-3.2:** 任務卡片上必須清晰地、優先地展示獎勵, 其次是任務標題和資格條件。
- **FR-3.3 (核心互動):** 當用戶與一個資格不符的任務卡片互動時, 系統必須提供一個清晰的引導流程, 告知用戶需要完成哪些授權或驗證(例如: 連接 Instagram)才能解鎖該任務。
- **FR-3.4:** 必須提供基礎的篩選(按任務類型)和排序(按獎勵最高、最新發布)功能。
- **FR-3.5:** 任務列表的排序應支持「贊助式推播 (Sponsored Quest)」, 即付費的任務可以置頂顯示。

3.4 功能詳述: Agents (我的代理)

- **FR-4.1:** 必須提供一個頁面, 讓用戶可以查看、管理自己擁有的「個人化 AI 代理」。
- **FR-4.2:** 每個代理應以一張獨立的卡片展示, 包含其名稱(如 "美食評論家 Agent")、狀態 (Active/Inactive)、核心技能標籤以及歷史表現記錄(如平均評分、完成任務數)。

3.5 功能詳述: Profile (我的)

- **FR-5.1:** 必須作為一個整合式的個人中心樞紐。
- **FR-5.2:** 必須提供通往以下四個核心子頁面的入口:
 1. 我的錢包 (Wallet): 查看詳細的交易記錄、獎勵明細和提現功能。
 2. 探索 Twin Matrix (Explore Matrix): 進入全功能的 Twin Matrix 視覺化探索頁面, 用戶可在此深入了解自己的數位 DNA。
 3. 任務歷史 (Quest History): 查看所有已完成和已過期的任務記錄。
 4. 帳號設定 (Settings): 編輯基本個人資料、管理已連接的第三方帳號、設定通知偏好。
-

3.6 功能詳述: 任務執行流程 (Quest Participation Flow)

- **FR-6.1 (MVP 核心場景):** 必須完整支持「產品體驗測試」任務類型。
- **FR-6.2:** 流程必須能夠引導用戶前往外部連結下載 App。
- **FR-6.3:** 流程必須清晰地指示用戶進行螢幕錄影。
- **FR-6.4:** 必須提供一個對話式 AI 互動介面, 作為用戶上傳已錄製影片的核心入口。

- **FR-6.5:** 影片上傳成功後, AI 必須給予即時的、鼓勵性的回饋, 並明確告知用戶的某項 `Twin Matrix Trait` 因此獲得了提升。
-

4. 範圍定義 (Scope Definition)

- **IN SCOPE (本次 MVP 必須包含的功能):**
 - 完整的用戶註冊、登入流程。
 - 上述 FR-1 至 FR-6 的所有功能規格。
 - 完整的 Light Mode 和 Dark Mode 主題支持。
 - 「英雄」等級用戶的專屬主題色 (`Hero Theme`) 功能。
 - 基礎的 App 內推播通知功能。
 - **OUT OF SCOPE (本次 MVP 明確排除的功能):**
 - 複雜的 AI 任務個人化推薦演算法(MVP 階段使用基礎匹配邏輯)。
 - 完整的站內爭議處理系統(簡化為「聯繫客服」)。
 - 用戶之間的社交功能(如私信、組隊)。
 - `Agents` 的全自動化執行功能(MVP 階段 Agent 主要用於身份展示和任務匹配)。
 - 與外部 Bot (Telegram, Discord) 的任務推播整合。
-

5. 成功指標 (Success Metrics)

- 用戶留存率 (**Retention Rate**): 7 日留存率 > 50%。
- 任務完成率 (**Quest Completion Rate**): 成功完成的任務數 / 已接受的任務數 > 30%。
- 核心循環參與率 (**Core Loop Engagement**): 每週至少完成一次「授權 → 解鎖 → 完成任務」循環的用戶比例。
- 用戶滿意度 (**User Satisfaction**): App Store 評分或 NPS (淨推薦值) 調查。

系統需求文件 (SRD): twin3 使用者端平台 MVP v1.0

文件版本: 1.0

發布日期: 2025年10月11日

目標: 本文件為工程團隊提供了實現 twin3 使用者端 MVP 所需的技術架構、數據模型、API 接口、非功能性需求及開發規範。

1. 系統架構 (System Architecture)

1.1 核心技術棧 (Core Technology Stack)

- 前端框架 (Frontend): **React Native** (使用 **Expo** 框架)
 - 理由: 一次開發, 同時部署至 iOS 和 Android, 最大化 MVP 開發效率。
- 後端服務 (Backend): **Firebase**
 - **Authentication:** Firebase Auth (支援 Google/Apple 登入)。
 - **數據庫 (Database):** **Firestore** (NoSQL 文件數據庫)。
 - **伺服器邏輯 (Server-side Logic):** **Cloud Functions for Firebase** (Node.js 環境)。
 - 理由: 提供完整的後端即服務 (BaaS), 極大簡化了用戶認證、數據庫管理和後端邏輯的開發, 非常適合 MVP 階段的快速迭代。
- 導航 (Navigation): **React Navigation**
 - 理由: React Native 生態中最成熟、功能最強大的導航解決方案。
- 樣式 (Styling): **Tailwind CSS** (透過 **nativewind** 函式庫在 React Native 中實現)
 - 理由: 提供高效的原子化 CSS 開發體驗, 便於實現響應式設計和設計系統。

1.2 數據與區塊鏈架構

- 鏈下優先原則 (Off-Chain First):
 - 核心邏輯: 所有高頻、需要即時反饋的操作, 如 **Twin Matrix** 的 **Trait** 分數更新、任務歷史記錄、用戶評價等, 都必須在 **Firestore** 中進行。
 - 理由: 確保流暢、零延遲的使用者體驗, 並免除用戶在日常操作中需要支付的 Gas Fee。
- 鏈上錨定機制 (On-Chain Anchoring):
 - 實現: 透過一個定期的後台排程任務 (**Scheduled Cloud Function**), 將每個用戶 **Twin Matrix** 數據的哈希摘要 (**Hash Digest**), 錨定到 **Polygon** 區塊鏈上的 **Soulbound Token (SBT)** 中。
 - **Gas Fee** 處理: 此過程產生的所有 Gas Fee 均由平台方支付, 其成本從平台的商業收入 (如 B2B 抽成) 中覆蓋。

- 理由：在不犧牲 UX 的前提下，賦予 Twin Matrix 數據可驗證性、不可篡改性和去中心化特性。

2. Firestore 數據庫模型 (Data Schema)

- 2.1

- **Document ID:** `firebaseAuthUserId`
- **Fields:**
 - `email`: String (from Auth)
 - `displayName`: String
 - `photoURL`: String
 - `level`: String (e.g., 'Basic', 'Hero')
 - `createdAt`: Timestamp
-
- **Subcollections:**
 - : (Document ID: 'summary') 包含 `completionRate`, `humanityIndex`, `traits` (Map) 等。
 - : (Document ID: 'main') 包含 `balance` (Number), `transactions` (Subcollection)。
 - : (Document ID: `agentId`) 包含 `agentType`, `status`, `performance` (Map)。

- 2.2

- **Document ID:** (Auto-generated)
- **Fields:** `title`, `description`, `questType`, `reward` (Map), `status`, `targetAudience` (Map), `creatorId`, `createdAt` (Timestamp)。

- 2.3

- **Document ID:** (Auto-generated)
- **Fields:** `questId`, `userId`, `status` ('pending', 'approved'), `content` (Map, e.g., { `videoUrl`: '...' }), `submittedAt` (Timestamp)。

3. 資訊架構與路由 (Information Architecture & Routing)

- 3.1 導航結構 (React Navigation):

- 根導航: `BottomTabNavigator`
 - : `DashboardScreen`
 - : 一個 `StackNavigator`, 根畫面為 `QuestMarketplaceScreen`。
 - : `AgentsScreen`
 - : 一個 `StackNavigator`, 根畫面為 `ProfileScreen`。

- **3.2 路由表 (Route Map):**

- /dashboard: DashboardScreen
- /quests: QuestMarketplaceScreen
- /quests/:id: QuestDetailScreen
- /quests/:id/participate: QuestParticipationScreen
- /agents: AgentsScreen
- /profile: ProfileScreen
- /profile/wallet: WalletScreen
- /profile/matrix: MatrixScreen
- /profile/settings: SettingsScreen

4. UI 設計系統規格 (Design System Specification)

- **4.1 框架: Google Material Design 3 (M3)**

- 所有元件的開發必須基於 M3 的視覺和互動規範。

- **4.2 主題與色彩 (Theming & Colors):**

- **SR-4.2.1:** 系統必須實現一個主題提供者 (**ThemeProvider**), 能夠在整個應用程式中動態切換 `Light` 和 `Dark` 模式。
- **SR-4.2.2 (英雄主題):** 當已登入用戶的 `user.level` 屬性為 `'Hero'` 時, 系統的 M3 種子顏色 (Seed Color) 必須從預設的藍色 (`#137fec`) 動態切換為紫色 (`#9370DB`)。這個切換必須即時反映在所有使用主色的元件上。

5. 非功能性需求 (Non-Functional Requirements)

- **5.1 性能 (Performance):**

- 啟動時間: App 冷啟動到可互動的時間應小於 3 秒。
- 列表渲染: 所有包含長列表的頁面(如任務市場), 必須實現虛擬化 (Virtualization) 以保證滾動流暢。
- **API 回應:** 所有 Firestore 的讀取操作, 必須設定合理的索引 (Indexes), 確保 95% 的查詢在 500ms 內返回。

- **5.2 安全性 (Security):**

- 數據庫規則: 必須在 `firestore.rules` 中定義嚴格的安全規則, 確保:
 - 用戶只能讀取和寫入自己的 `user/{userId}` 文件及其子集合。
 - 用戶可以讀取 `quests` 集合, 但不能寫入。
 - 用戶只能創建屬於自己的 `submissions` 文件。
-

- **Cloud Functions:** 所有函式的進入點, 都必須驗證 `context.auth`, 確保調用者已通過身份驗證。
- **5.3 無障礙 (Accessibility - WCAG 2.1 AA):**
 - **SR-5.3.1:** 所有可點擊元件的最小觸控區域不得小於 **48×48dp**。
 - **SR-5.3.2:** 所有非純裝飾性的圖片, 都必須提供替代文字 (Accessibility Label)。
 - **SR-5.3.3:** 文字與背景的對比度必須符合 AA 等級要求。
- **5.4 可靠性 (Reliability):**
 - **離線支持:** MVP 階段暫不要求完整的離線功能, 但 App 在網路不穩定時, 應能優雅地處理錯誤, 並提供「重試」機制。