# AI生存小組期末大項目提案書：Line 智能聊天記帳系統 (進階版)

專案名稱 (Project Title): ChatBot-Finance：Line 智能聊天記帳系統

執行小組 (Team): AI 生存小組

日期 (Date): 2025 年 11 月 21 日

## 一、 構想與目標 (Concept and Objectives)

### 1.1 專案構想 (Core Concept)

本專案旨在徹底解決傳統記帳 App 輸入流程繁瑣（需解鎖手機、開啟 App、選擇分類、輸入金額）的痛點。透過結合 **Line Messaging API**、**Gemini 語音轉文字 (STT)** 技術與 **大型語言模型 (LLM)** 的深度語義分析，建立一個「無摩擦、零門檻」的記帳體驗。

使用情境範例：

用戶只需在買完東西後，拿起手機像傳訊息給朋友一樣，按住語音鍵說：「剛剛在 7-11 買了咖啡和麵包，總共 85 元。」

系統將自動完成：

1. **聽：** 接收語音檔。
2. **轉：** 轉成文字「剛剛在 7-11 買了咖啡和麵包，總共 85 元」。
3. **想：** 分析出 項目: 咖啡、麵包、金額: 85、地點: 7-11、分類: 餐飲/早餐。
4. **記：** 寫入資料庫並回覆用戶「已記錄：餐飲費 85 元」。

### 1.2 核心目標 (Key Objectives)

1. **語音輸入整合：** 成功接收 Line 語音訊息，並利用 API 準確轉譯成文字。
2. **LLM 智能分類：** 透過 LLM 對轉譯後的文字進行**情境判斷**，實現高階分類（例如：將「跟朋友吃午餐」自動歸類為「交際費」，而非單純的「餐費」）。
3. **高效率記帳：** 將記帳流程縮短至 5 秒內（僅需錄音時間）。
4. **資料結構化：** 成功解析用戶非結構化的口語輸入，轉化為 JSON 格式儲存（項目、金額、類別、時間）。

## 二、 執行步驟與技術棧 (Execution Steps and Tech Stack)

本專案採用現代化 Serverless 架構，確保低成本與高擴展性。

### 2.1 技術棧 (Technology Stack)

| **類別** | **技術/工具** | **詳細用途說明** |
| --- | --- | --- |
| **後端核心** | **Python (Flask)** | 輕量級 Web 伺服器，負責處理 Line Webhook 請求與邏輯串接。 |
| **通訊介面** | **Line Messaging API** | 使用 line-bot-sdk 處理用戶互動、訊息驗證 (Reply Token)、多媒體檔案下載。 |
| **AI 核心** | **Google Gemini API** | 一站式處理：  1. **STT**: 讀取音訊檔轉文字。  2. **NLP**: 進行意圖識別與實體擷取 (NER)。 |
| **資料儲存** | **Firebase Firestore** | NoSQL 資料庫，以 User ID 為 Document Key，儲存個人的記帳集合 (Collections)。 |
| **雲端部署** | **Render / Heroku** | 提供 HTTPS 服務端點，讓 Line Platform 能夠呼叫 Webhook。 |

### 2.2 詳細執行步驟 (Step-by-Step Implementation)

我們將開發過程細分為四個具體階段，每個步驟皆有明確的產出目標。

#### **Phase 1: 基礎建設與環境搭建 (Infrastructure Setup)**

* **Step 1.1 - Line Developer 設定：**
  + 申請 Line Developer 帳號，建立新的 Provider 與 Channel。
  + 取得 Channel Access Token 與 Channel Secret。
  + 開啟 Webhook 功能，並關閉 Line 官方預設的自動回應訊息。
* **Step 1.2 - 本地 Python 環境建置：**
  + 建立專案資料夾，設定虛擬環境 (venv)。
  + 安裝依賴套件：pip install flask line-bot-sdk google-generativeai firebase-admin。
* **Step 1.3 - 建立 Flask 伺服器 (Hello World)：**
  + 撰寫 app.py，建立 /callback 路由以接收 Line 請求。
  + 實作基本的文字回覆功能（Echo Bot），確保伺服器能運作。
* **Step 1.4 - 雲端部署：**
  + 將程式碼推送到 GitHub。
  + 連結 Render/Heroku 進行部署，取得 HTTPS URL。
  + 將 HTTPS URL 填回 Line Developer Console 的 Webhook URL 欄位並驗證成功。

#### **Phase 2: 語音處理與 STT 串接 (Audio & Speech-to-Text)**

* **Step 2.1 - 接收音訊事件：**
  + 在 app.py 中增加對 MessageEvent 且類型為 AudioMessage 的判斷邏輯。
* **Step 2.2 - 下載音訊檔案：**
  + 使用 line\_bot\_api.get\_message\_content(message\_id) 取得二進位音訊流。
  + 將音訊暫存於本地 /tmp 目錄（格式通常為 m4a）。
* **Step 2.3 - 呼叫 Gemini API 進行 STT：**
  + 設定 Gemini API Key。
  + 使用 Gemini 的多模態功能，將音訊檔上傳並 Prompt：「請逐字轉錄這段語音的內容」。
  + 取得回傳的純文字字串。

#### **Phase 3: LLM 語意分析與資料庫寫入 (NLP & Database)**

* **Step 3.1 - 設計記帳專用 Prompt：**
  + 設計 System Prompt：「你是一個專業的記帳助手。請從使用者的輸入中擷取以下 JSON 資訊：{item: string, price: int, category: string}。如果沒有金額，請回覆 null。」
* **Step 3.2 - 整合 LLM 邏輯：**
  + 將 Step 2.3 轉譯出的文字（或使用者直接輸入的文字）丟給 Gemini 進行結構化分析。
  + 解析 Gemini 回傳的 JSON 資料。
* **Step 3.3 - Firestore 資料庫串接：**
  + 初始化 Firebase Admin SDK，下載私鑰 json。
  + 設計資料庫路徑：users/{user\_line\_id}/transactions/{transaction\_id}。
  + 將 LLM 解析後的 {date, item, price, category} 寫入 Firestore。
* **Step 3.4 - 回覆使用者：**
  + Bot 根據寫入結果回覆：「✅ 已記錄：早餐 - 三明治 ($45)」。

#### **Phase 4: 查詢功能與優化 (Query & Refinement)**

* **Step 4.1 - 實作查詢意圖識別：**
  + 當使用者輸入包含「查詢」、「花了多少」、「報表」等關鍵字時，觸發查詢邏輯。
* **Step 4.2 - 資料聚合 (Aggregation)：**
  + 從 Firestore 撈取該用戶當月/當週的數據。
  + 計算總金額與分類佔比。
* **Step 4.3 - 格式化輸出：**
  + 將統計結果組合成易讀的文字訊息回傳（例如：「本月累計：$5,200元，其中餐費佔 40%」）。

## 三、 產品特色 (Product Features)

1. **零摩擦語音記帳 (Voice Input & Zero-Friction):**
   * **直覺操作：** 支援用戶直接發送 Line 語音訊息，無需打字，長輩或忙碌時皆可輕鬆使用。
   * **混合輸入：** 系統同時支援文字與語音，依情境自由切換。
2. **LLM 智能情境分類 (Smart Contextual Categorization):**
   * **深度理解：** 透過 LLM 對輸入內容進行推論。
   * **自動標籤：** 範例：「買了兩張電影票和爆米花」→ 自動拆分為 娛樂費，而不會誤判為雜支。
3. **對話式 NLP 記帳 (Conversational Input):**
   * 支援多種口語化表達：「吃便當花了 120 元」、「去超商買飲料 60」，甚至模糊語句修正。
4. **即時回饋與視覺化 (Real-time Summary):**
   * 每次記帳後即時確認紀錄細節，防止記錯。
   * 提供簡易的文字報表，讓用戶隨時掌握財務狀況。

## 四、 商業價值與未來潛力 (Business Value and Future Potential)

### 4.1 核心商業價值 (Core Value Proposition)

* **極致用戶體驗：** 結合語音與 Line Bot，將記帳門檻降到最低，克服用戶對傳統記帳 App 的惰性，大幅提高資料輸入的完成率 (Retention)。
* **數據洞察基礎：** LLM 提供更精準的「情境」分類，這比傳統的「項目」分類更有價值，能分析出用戶的生活型態（Lifestyle）。
* **Line 生態優勢：** 利用 Line 龐大用戶基礎，無需額外下載 App，好友推薦即可使用，具備快速病毒式擴散的潛力。

### 4.2 未來擴展性 (Future Scalability)

* **進階語義分析與情緒判斷：** 讓 LLM 判斷語氣，例如：「今天心情不好，怒買了一件衣服 800」 -> 自動分類為「購物」並標註「情緒消費」，提供心理帳戶分析。
* **主動提醒與預算管理：** 整合行事曆或時間事件，當月還有 10 天但預算只剩 10% 時，主動 Line 推播提醒。
* **圖表化報表：** 整合 QuickChart API 或 Line Flex Message，輸出長條圖或圓餅圖，提供更直觀的視覺化財務報表。

**AI 生存小組** 堅信此專案能有效展示 Generative AI 在日常生活中的實際落地應用，不僅技術架構完整（STT + LLM + DB），更解決了真實世界的問題。