# 楊桃美食網 AI 智慧食譜平台 - 系統開發規劃書

## 文件資訊

* **專案名稱**: YTower AI Smart Cooking Platform
* **版本**: V1.0
* **日期**: 2025/12/12
* **規劃者**: Gemini (資深系統設計規劃師)
* **技術棧**: Angular 17+, Node.js (NestJS), Firebase/PostgreSQL, OpenAI API/Vertex AI

# 1. 產品需求規劃書 (PRD)

## 1.1 產品願景

打造一個結合「內容 (Content)」、「社群 (Community)」與「商務 (Commerce)」的 AI 驅動型美食平台。透過 AI 解決使用者「不知道吃什麼」的痛點，並自然引導至相關廚具與食材的購買，實現從靈感到餐桌的一站式服務。

## 1.2 目標受眾 (Target Audience)

1. **家庭主婦/主夫**: 需要快速解決每日三餐，注重健康與營養。
2. **忙碌上班族**: 冰箱有剩材，需要 15 分鐘快速料理方案。
3. **料理新手**: 依賴詳細步驟與影片教學，對廚具不熟悉。
4. **電商潛在客戶**: 尋找特定廚具或被食譜燒到的消費者。

## 1.3 核心功能模組 (基於設計稿)

| **優先級** | **功能模組** | **功能描述** | **對應設計稿** |
| --- | --- | --- | --- |
| **P0** | **首頁與導航** | 搜尋霸、動態 Banner、推薦主題入口、購物網導流 | 0\_首頁.pdf |
| **P0** | **AI 食譜助手** | 輸入食材/風格，AI 生成食譜、推薦廚具、滿意度評分 | 2-1, 2-2 |
| **P0** | **食譜瀏覽系統** | 分類篩選 (時間/成本/器具)、列表展示、詳細頁 (影片/步驟/營養) | 3-1, 3-2, 3-3, 3-4 |
| **P1** | **最新消息 (News)** | 官方活動、食譜趨勢文章、AI 功能推廣 | 4-1, 4-2 |
| **P1** | **關於我們** | 品牌故事、願景、核心價值展示 | 1\_關於我們.pdf |
| **P2** | **會員系統** | 收藏食譜、AI 生成紀錄、評論互動 | - |

## 1.4 使用者體驗流程 (UX Flow) - AI 食譜生成

1. **入口**: 首頁點擊「AI 幫你想」或側邊欄進入 AI 助手。
2. **輸入**: 使用者輸入「冰箱現有食材」(如：豆腐、絞肉) 與「偏好」(如：川味、快速)。
3. **處理**: 系統顯示「AI 正在翻閱食譜...」的趣味加載動畫。
4. **結果**: 呈現完整食譜 (麻婆豆腐)，包含影片、步驟。
5. **導購**: 在「使用器具」區塊推薦 "Woody 不沾鍋"，點擊可跳轉購買。
6. **回饋**: 使用者給予星級評分，系統優化下次推薦。

# 2. 功能設計文件 (FDD)

## 2.1 系統資料流程圖 (DFD)

**Mermaid Context Diagram**

graph TD  
 User[使用者] -->|輸入食材與偏好| Frontend[Angular 前端]  
 Frontend -->|API 請求| Backend[後端 API Gateway]  
 Backend -->|Prompt| AI\_Service[AI 模型服務]  
 AI\_Service -->|生成食譜 JSON| Backend  
 Backend -->|查詢關聯商品| DB[(資料庫)]  
 DB -->|食譜與商品數據| Backend  
 Backend -->|回傳完整資料| Frontend  
 Frontend -->|渲染頁面| User  
 User -->|點擊購買| Ecommerce[外部購物車系統]

## 2.2 詳細頁面與功能規格 (Detailed Page & Functional Specifications)

### A. 首頁 (Home Page) - 參考檔：0\_首頁.pdf

* **導航列 (Navigation)**: 懸浮置頂，包含「關於我們」、「食譜專區」、「AI 食譜」、「最新消息」、「聯絡我們」。右側整合會員登入與購物車 icon。
* **Hero Section**:
  + **動態標語**: 顯示「今天吃什麼？」、「沒有想法？」，搭配打字機效果 (Typewriter Effect)。
  + **AI CTA**: 顯著按鈕「讓 AI 幫你想今晚吃什麼！」，點擊觸發 Dialog 或跳轉至 AI 頁面。
  + **搜尋霸**: 支援模糊搜尋食譜名稱或食材。
* **推薦區塊**:
  + **熱門食譜**: 橫向捲動展示 (Carousel)，如「蔬菜拼盤」等高評分食譜。
  + **廚具推薦**: 展示 "Cuisinart"、"Woody" 等品牌商品，標示優惠價格 (如「返場再殺 799 元」)。

### B. 關於我們 (About Us) - 參考檔：1\_關於我們.pdf

* **視覺體驗**: 採用滾動視差 (Parallax Scrolling) 技術，背景圖層隨捲動緩慢移動，營造沈浸感。
* **內容區塊**:
  + **品牌願景**: 展示「用美食連結彼此，用科技開啟未來」的核心理念。
  + **核心價值**: 三大支柱 icon 展示（深厚專業、科技應用、情感連結）。
  + **服務介紹**: 連結至購物網、食譜影音、AI 體驗與社群交流。

### C. AI 食譜助手 (AI Recipe Assistant) - 參考檔：2-1, 2-2.pdf

* **輸入介面**:
  + **食材輸入**: 支援 Tag 輸入法 (如 Instagram 標籤)，具備自動補全 (Autocomplete) 功能。
  + **風格選擇**: 下拉或按鈕選擇料理風格 (如：快速料理、健康飲食、家常菜)。
* **生成體驗**:
  + **Loading 動畫**: 畫面顯示鍋子烹煮動畫，提示語隨機變換 (如「正在翻閱百萬食譜...」、「正在挑選最適合的鍋具...」)。
* **生成結果**:
  + **食譜卡片**: 顯示菜名 (如：麻婆豆腐)、烹飪時間、星級。
  + **智能導購**: 自動在「使用器具」區塊推薦合適商品 (如：Woody 26cm 深炒鍋)，顯示「立即選購」按鈕。
  + **相似推薦**: 若使用者不滿意，底部提供「其他相似食譜」選項。

### D. 食譜瀏覽系統 (Recipe System) - 參考檔：3-1 ~ 3-4.pdf

* **總覽頁 (Overview)**:
  + **主題分類**: 網格展示「健康飲食」、「快速料理」、「節慶主題」、「親子學習」等分類入口。
  + **Top 10 排行**: 側邊欄顯示熱門食譜排行 (如：麻婆豆腐 NO.1)。
* **列表與篩選**: 支援多維度篩選條件 (時間/預算/器具/烹調方式)。
* **食譜內頁 (Detail)**:
  + **影音播放**: 頂部嵌入 YouTube/Vimeo 播放器，支援浮動視窗 (Picture-in-Picture) 方便邊看邊做。
  + **動態份量**: 切換人數 (1人/2人/4人)，食材清單份量即時計算更新。
  + **步驟檢核**: 每個步驟前有 Checkbox，勾選後該行文字變淡，方便追蹤進度。
  + **商城串接**: 在材料與器具旁顯示「楊桃購物網」對應商品，點擊開啟新分頁或加入購物車。

### E. 最新消息 (News) - 參考檔：4-1, 4-2.pdf

* **列表頁**: 以卡片式或時間軸展示最新活動、AI 功能上線通知。
* **文章內頁**:
  + **側邊導覽**: 右側固定顯示「人氣食譜排行榜」與「購物網人氣推薦榜」，增加交叉銷售機會。
  + **分享功能**: 整合 LINE、Facebook 分享按鈕。

### F. 互動微動畫 (Micro-interactions) 規格

* **Hover 效果**: 食譜卡片 Hover 時，圖片輕微放大 (Scale 1.05)，並播放 3 秒預覽影片 (如有)。
* **加入購物車**: 點擊購買按鈕時，商品圖片縮小並以拋物線路徑飛入右上角購物車 icon。
* **載入狀態**: 使用 Skeleton Screen (骨架屏) 搭配品牌色脈衝動畫，取代傳統 Loading Spinner。

# 3. 技術架構文件 (TAD)

## 3.1 技術選型

* **前端框架**: **Angular 17+**
  + **核心**: Standalone Components (減少 Module 複雜度), Signals (細粒度響應式更新)。
  + **樣式**: SCSS + Tailwind CSS (快速佈局) + **GSAP** (處理複雜動畫)。
  + **狀態管理**: NgRx 或 Angular Signals Store。
  + **SSR**: Angular Universal / AnalogJS (確保 SEO 排名)。
* **後端服務**:
  + **API**: NestJS (TypeScript 支援佳，架構嚴謹)。
  + **Database**: PostgreSQL (儲存結構化食譜), Redis (快取熱門搜尋)。
  + **AI Provider**: OpenAI GPT-4o-mini (性價比高) 或 Google Gemini API。

## 3.2 系統架構圖 (System Architecture)

graph TD  
 Client[Angular Client \n (Mobile/Desktop)]  
 CDN[CDN \n (Images/Videos/Assets)]  
   
 subgraph Cloud\_Infrastructure  
 LB[Load Balancer]  
 API\_Gateway[API Gateway]  
   
 subgraph Services  
 Auth[Auth Service]  
 Recipe[Recipe Service]  
 AI\_Handler[AI Integration Service]  
 Product[Product Service]  
 end  
   
 subgraph Data\_Layer  
 Redis[(Redis Cache)]  
 PrimaryDB[(PostgreSQL)]  
 VectorDB[(Vector DB for RAG)]  
 end  
 end  
   
 Client --> LB  
 Client --> CDN  
 LB --> API\_Gateway  
 API\_Gateway --> Auth  
 API\_Gateway --> Recipe  
 API\_Gateway --> AI\_Handler  
 API\_Gateway --> Product  
   
 AI\_Handler -->|Retrieval Augmented Generation| VectorDB  
 AI\_Handler -->|LLM API Call| External\_AI[OpenAI / Gemini]  
   
 Recipe --> PrimaryDB  
 Recipe --> Redis

## 3.3 Angular 專案結構建議

src/  
 app/  
 core/ (單例服務: AuthService, ApiService)  
 shared/ (共用元件: Button, Card, LoadingSpinner)  
 features/ (功能模組)  
 home/  
 about/  
 recipe-detail/  
 ai-assistant/  
 components/  
 input-form/  
 result-card/  
 ai-assistant.component.ts  
 news/  
 layouts/ (MainLayout, AuthLayout)  
 app.routes.ts  
 app.config.ts

# 4. 介面設計規劃文件 (IDP) - 針對 Angular 開發

## 4.1 UI 設計規範 (Design System)

* **主色調**: 品牌陶土紅 (#C45D4A)、健康綠 (#4CAF50 - 針對健康分類)、深灰文字 (#333333)。
* **字體**: Noto Sans TC (確保繁體中文閱讀體驗)。
* **圓角**: 卡片 border-radius: 12px，按鈕 border-radius: 30px (Pill shape)。

## 4.2 動畫與互動細節 (Angular Animations / GSAP)

### 場景 1：AI 食譜生成等待 (Loading)

* **設計**: 畫面中央出現一個鍋子 icon，食材 icon (紅蘿蔔、肉、蔥) 依序跳入鍋中，鍋蓋蓋上後開始冒煙。
* **技術**: 使用 Lottie (JSON動畫) 或 GSAP Timeline。
* **Angular 實作**:  
  // 虛擬代碼  
  trigger('cookingState', [  
   state('cooking', style({ transform: 'rotate(0)' })),  
   transition('\* => cooking', animate('1s', keyframes([...])))  
  ])

### 場景 2：食譜卡片進入 (Page Entry)

* **設計**: 當使用者滾動到「推薦食譜」區塊時，卡片不應一次全部顯示，而是使用 Stagger (交錯) 效果，一張張向上浮現 (Fade In Up)。
* **技術**: Angular @query 與 @stagger。

### 場景 3：加入購物車/立即購買導購

* **設計**: 在食譜詳情頁點擊「購買此鍋具」時，商品圖片縮小並拋物線飛入右上角的購物車 icon。
* **目的**: 強烈的視覺回饋，提升轉化率。

## 4.3 響應式佈局策略 (RWD)

* **Desktop (>1024px)**: 3-4 列網格展示食譜，側邊欄顯示熱門排行。
* **Tablet (768px - 1024px)**: 2 列網格，側邊欄收合為底部推薦。
* **Mobile (<768px)**: 單列流式佈局 (Stream)，導航列變為 Bottom Tab Bar 或 Hamburger Menu。

## 4.4 Mermaid 類別圖 (Class Diagram) - 核心資料結構

classDiagram  
 class User {  
 +String id  
 +String name  
 +Array favorites  
 +login()  
 +saveRecipe()  
 }  
   
 class Recipe {  
 +String id  
 +String title  
 +String description  
 +Int cookingTime  
 +Int servings  
 +Array steps  
 +Array ingredients  
 +String videoUrl  
 +Array tags  
 }  
   
 class Ingredient {  
 +String name  
 +String amount  
 +String unit  
 }  
   
 class Kitchenware {  
 +String id  
 +String name  
 +String imageUrl  
 +Float price  
 +String purchaseLink  
 }  
   
 Recipe "1" \*-- "many" Ingredient : contains  
 Recipe "1" -- "many" Kitchenware : recommends  
 User "1" -- "many" Recipe : likes

## 5. 專案時程估算 (開發階段)

1. **Phase 1: 基礎建設 (Week 1-2)**
   * Angular 專案建置、Tailwind 設定、共用元件開發 (Header, Footer, Cards)。
2. **Phase 2: 核心數據與 CMS (Week 3-4)**
   * 食譜資料庫設計、後端 CRUD API、食譜列表與詳情頁前端。
3. **Phase 3: AI 助手整合 (Week 5-6)**
   * 串接 OpenAI API、Prompt 優化、前端聊天/輸入介面與動畫開發。
4. **Phase 4: 電商導購與互動優化 (Week 7-8)**
   * 商品關聯邏輯、微動畫 (GSAP) 實作、RWD 調整。
5. **Phase 5: 測試與發布 (Week 9)**
   * SSR 調優、效能測試 (Lighthouse)、部署。