

# Ingredient Vision Recipe Finder

## MVP Technical Design Document

February 2026

### 1 Product Overview

The **Ingredient Vision Recipe Finder** is a web-based, mobile-friendly application that enables users to take a photo of available cooking ingredients and receive realistic, cookable recipe recommendations. The system leverages computer vision and AI-based reasoning to identify ingredients from images, normalize them into canonical food items, and match them against a recipe database.

#### MVP Goal

Convert a single image of ingredients into **3–5 usable recipes** in under **10 seconds**.

### 2 MVP Scope Definition

#### 2.1 In Scope

- Image upload via camera or file input
- Ingredient detection from images
- User confirmation and editing of detected ingredients
- Recipe matching and ranking
- Recipe detail view (ingredients and preparation steps)
- Deployed production MVP

#### 2.2 Out of Scope (Post-MVP)

- User accounts and profiles
- Grocery delivery integrations
- Nutrition tracking

- Meal planning and history
- Native mobile applications

### 3 System Architecture Overview

The system follows a client-server architecture with asynchronous AI processing and external API integrations.

- Frontend: User interface and image capture
- Backend: Image processing, AI orchestration, and recipe matching
- AI Services: Vision-based ingredient detection and text-based normalization
- External APIs: Spoonacular recipe data
- Storage: AWS S3 for image persistence

### 4 Core User Flow

1. User uploads or captures an image of ingredients
2. Backend sends the image to the vision model
3. Vision model returns detected ingredient labels
4. User confirms or edits the ingredient list
5. Backend matches ingredients against recipe data
6. Ranked recipes are returned to the frontend

### 5 AI Image Processing

#### 5.1 Selected Approach

- **OpenAI Vision API** for image-to-ingredient detection
- **GPT-based normalization** for canonical ingredient naming

#### 5.2 Example Output

```
["chicken breast", "onion", "garlic", "bell pepper"]
```

## 6 Recipe Matching Logic

### 6.1 Recipe Source

- Spoonacular API

### 6.2 Matching Algorithm (MVP)

1. Count exact ingredient overlaps
2. Penalize missing ingredients
3. Rank recipes by simplicity

$$\text{Score} = (\text{matched ingredients} \times 2) - \text{missing ingredients}$$

## 7 Technology Stack

### 7.1 Frontend

- Next.js
- Tailwind CSS
- Browser Camera and File APIs

### 7.2 Backend

- Python FastAPI
- RESTful APIs
- Asynchronous request handling

### 7.3 AI / ML

- OpenAI Vision API
- GPT for ingredient normalization

### 7.4 Infrastructure

- Frontend Hosting: Vercel
- Backend Hosting: Railway
- Object Storage: AWS S3

## 8 Development Timeline

- **Week 1 (Feb 9–13):** Planning, API design, repo setup, frontend scaffold
- **Week 2 (Feb 16–20):** Image upload, vision integration, ingredient extraction
- **Week 3 (Feb 23–27):** Recipe matching logic, ranking, UI components
- **Week 4 (Mar 2–6):** Error handling, UX polish, deployment
- **Week 5 (Optional):** Accuracy improvements, caching, UI animations

## 9 MVP Success Criteria

- Ingredient detection accuracy  $\geq 80\%$
- End-to-end recipe generation under 10 seconds
- Fully automated flow without manual intervention
- Publicly accessible deployed demo

## 10 API and Infrastructure Cost Analysis

Service	Estimated Monthly Cost
OpenAI Vision API	\$10–15
GPT (Text Normalization)	<\$1
Spoonacular API	\$0 (Free Tier)
AWS S3 Storage	<\$1
Backend Hosting (Railway)	~\$10
Frontend Hosting (Vercel)	\$0
<b>Total</b>	<b>\$20–25</b>

## 11 Future Enhancements

- User accounts and saved recipes
- Nutritional analysis
- Meal planning and grocery list generation
- Mobile application (React Native)

## 12 Resume-Ready Description

Built an AI-powered web application that transforms images of available ingredients into ranked, cookable recipe recommendations using computer vision, natural language processing, and cost-optimized API integrations.

## 13 User Interface Design Considerations

The user interface of the Ingredient Vision Recipe Finder is designed to be minimal, intuitive, and mobile-first, while remaining fully functional on desktop devices. The UI prioritizes speed, clarity, and user control, particularly given the probabilistic nature of AI-based ingredient detection.

The application follows a step-based interaction model, where each screen focuses on a single primary user action.

### 13.1 Design Principles

- **Simplicity:** Each screen presents a single, clear call-to-action.
- **Transparency:** Users are informed when AI processing is occurring.
- **User Control:** Detected ingredients are editable prior to recipe matching.
- **Mobile-First:** All layouts are optimized for small screens.

### 13.2 Primary UI Screens

#### 13.2.1 Image Upload Screen

The image upload screen serves as the primary entry point to the application. Users may capture a photo using their device camera or upload an existing image from local storage.

- Prominent camera and upload buttons
- Minimal visual distractions
- Supported image format guidance

This screen intentionally avoids secondary navigation to reduce cognitive load and guide users toward immediate action.

#### 13.2.2 Image Preview and Processing State

After image submission, users are presented with a preview of the uploaded image alongside a processing indicator. This state reassures users that the system is actively analyzing their input.

- Uploaded image preview

- Loading animation or progress indicator
- Informational messaging indicating analysis time

### **13.2.3 Ingredient Confirmation Screen**

The ingredient confirmation screen is a critical component of the user experience. Because computer vision outputs may contain inaccuracies, users are given full control to modify the detected ingredient list before proceeding.

- Editable ingredient chips
- Ability to remove incorrectly detected ingredients
- Manual addition of missing ingredients
- Explicit confirmation action to proceed

This screen increases user trust and improves overall recipe matching accuracy.

### **13.2.4 Recipe Results Screen**

The recipe results screen displays a ranked list of recipe recommendations based on the confirmed ingredient set.

- Recipe cards displaying recipe name and match quality
- Indicators for missing ingredients
- Clear call-to-action to view recipe details

The number of displayed recipes is intentionally limited to reduce decision fatigue.

### **13.2.5 Recipe Detail Screen**

The recipe detail screen presents all necessary information for meal preparation in a clear, structured format.

- Ingredient list with matched and missing indicators
- Step-by-step cooking instructions
- Optional external recipe source link

The layout prioritizes readability and usability on mobile devices.

### 13.3 UI States and Error Handling

The interface explicitly accounts for non-ideal scenarios to prevent user confusion and abandonment.

- Image upload failures
- AI processing timeouts
- No matching recipes found

In each case, users are provided with actionable feedback and the ability to retry or adjust inputs.

### 13.4 Future UI Enhancements

Post-MVP UI improvements may include:

- Ingredient confidence indicators
- Recipe match strength visualizations
- Subtle animations to mask processing latency