

# Recipe Collector — Product Requirements Document (MVP)

**Purpose:** Define the minimum viable product for a playful Android app that captures recipes from photos, auto-structures them, categorizes them, and stores them locally for browsing, editing, sharing, and basic analysis (baker's percentages + nutrition).

---

## Project Overview

**Goal:** Photo → Clean recipe in your library in under 30 seconds.

**Platform:** Android phone (single device), offline-first.

**Primary users:** Home cooks and hobbyist bakers.

**Non-goals:** Cloud sync, multi-user accounts, social features, complex nutrition labels, desktop/tablet UI.

---

## Level

Medium (mobile + OCR + deterministic parsing).

## Type of Project

Mobile app · On-device OCR · Data extraction & classification · Offline storage.

## Skills Required

- Android (Kotlin, Jetpack Compose, Coroutines)
  - ML Kit Text Recognition (on-device OCR)
  - Kotlin parsing/regex for structured extraction
  - Data modeling (recipes, ingredients, steps, categories)
  - Room/SQLite
  - Basic nutrition math & baker's percentages
  - UI/UX for playful, accessible interfaces
-

## Personas

1) **The Collector:** Snaps recipes from books, wants clean storage, quick edits, and email sharing. 2) **The Baker:** Cares about weights, hydration %, and scaling. 3) **The Tracker:** Wants basic calories/macros per serving.

---

## Success Metrics (MVP)

- **Capture conversion:**  $\geq 80\%$  of captured images yield a structured recipe without re-shoot.
  - **Edit friction:** Median time to correct a recipe  $< 2$  minutes.
  - **Findability:** Search returns a relevant recipe in  $\leq 2$  queries 90% of the time.
- 

## Key Features & Milestones (MVP only)

### Milestone 1 — Core Capture, Store, Browse

- Photo capture (camera or gallery import)
- On-device OCR (ML Kit)  $\rightarrow$  raw text
- Deterministic parsing to Recipe JSON (title, yields, ingredients, steps)
- Local database (Room/SQLite)
- Library view (grid/cards), detail view, edit view
- Favorites (toggle + filter)
- Category auto-tagging for broad groups (breads, desserts, chicken, beef, etc.)

**Acceptance criteria** - Capture  $\rightarrow$  Save flow in  $\leq 30$  seconds. - Searchable by title and ingredient.

### Milestone 2 — Formatting, Email Share, & Fun UI

- Clean formatting template for ingredients/steps
- One-tap share via email (plain-text body)
- Playful UI (microinteractions, gentle haptics)
- Basic search (title, ingredients)

**Acceptance criteria** - Email a formatted recipe from the detail screen in  $\leq 3$  taps.

### Milestone 3 — Baker's Percentages & Nutrition (Basic)

- Baker's % when flour is present (hydration, salt %, sugar %, fat %)
- Volume $\rightarrow$ weight conversions using a small internal density table
- Basic nutrition per serving (kcal, protein, fat, carbs) using a built-in mapping

**Acceptance criteria** - For flour-based recipes, hydration % and key baker's % display automatically with an "edit weights" panel. - Nutrition per serving appears with an "approximate" disclaimer.

---

## User Stories

- As a user, I can snap a photo of a recipe and get a clean, editable recipe in my library.
  - As a user, I can correct ingredient names, amounts, and steps before saving.
  - As a user, I can favorite recipes and filter by favorites.
  - As a user, I can email a recipe in a readable format.
  - As a baker, I can see baker's percentages for dough/batter recipes.
  - As a health-conscious user, I can see basic calories/macros per serving.
- 

## App Flows

1) **Capture:** Home → Capture → Camera → Crop/adjust → OCR → Parse → Review/Edit → Save → Library. 2) **Browse:** Library (filters: Favorites, Category) → Recipe Detail → Edit / Email / Favorite. 3) **Edit:** Detail → Edit (ingredients, steps, yields, categories) → Save → Recompute baker's % / nutrition. 4) **Share:** Detail → Share → Email client (plain text).

---

## Architecture

- **UI:** Jetpack Compose + MVVM (ViewModel + immutable state)
  - **Domain:** Use-cases for capture, parse, save, compute
  - **Data:** Repository pattern over Room/SQLite
  - **OCR:** ML Kit Text Recognition (on-device)
  - **DI:** Hilt (or Koin)
  - **Images:** Coil
  - **Testing:** JUnit5, MockK, Robolectric, Paparazzi
- 

## Data & Storage

- Local database: **Room/SQLite** on the device (no sync).
- 

## Patterns & Frameworks

- **Language/UI:** Kotlin + Jetpack Compose
  - **Architecture:** Clean Architecture + MVVM + Repository pattern
  - **Storage:** Room/SQLite, DataStore (preferences)
  - **OCR:** Google ML Kit Text Recognition (on-device)
  - **Imaging:** Coil
  - **Testing:** JUnit5, MockK, Robolectric, Paparazzi
-

## Recipe Format

Adopt Schema.org/Recipe concepts in a compact JSON structure with baker's-percent and nutrition fields.

```
{
  "id": "uuid",
  "title": "Sourdough Bread",
  "source": {"type": "photo", "note": "Grandma's card"},
  "yields": "1 loaf",
  "servings": 12,
  "times": {"prep": 20, "rest": 180, "cook": 35, "total": 235},
  "categories": ["breads"],
  "tags": ["sourdough", "artisan"],
  "ingredients": [
    {"name": "bread flour", "qty": 500, "unit": "g", "grams": 500, "role": "flour"},
    {"name": "water", "qty": 350, "unit": "g", "grams": 350, "role": "water"},
    {"name": "salt", "qty": 10, "unit": "g", "grams": 10, "role": "salt"},
    {"name": "starter", "qty": 100, "unit": "g", "grams": 100, "role": "preferment"}
  ],
  "steps": [
    {"n": 1, "text": "Mix flour and water; autolyse 1 hr."},
    {"n": 2, "text": "Add starter and salt; fold every 30 min for 2 hrs."},
    {"n": 3, "text": "Shape, proof, and bake at 240°C for 35 min."}
  ],
  "photos": ["content://.../photo1"],
  "bakers": {"hydration_pct": 70, "salt_pct": 2, "starter_pct": 20},
  "nutrition": {"per_serving": {"kcal": 160, "protein_g": 5, "fat_g": 1, "carb_g": 33}},
  "created_at": 0, "updated_at": 0, "favorite": false
}
```

**Category taxonomy (seed)** - Breads; Desserts; Chicken; Beef; Pork; Seafood; Vegetarian/Vegan; Pasta/Grains; Soups/Stews; Salads; Breakfast; Sauces/Condiments

## Data Model (Room/SQLite)

- `recipes` (id PK, title, yields, servings, categories JSON, tags JSON, prep\_min, cook\_min, total\_min, favorite BOOL, created\_at, updated\_at)
- `ingredients` (id PK, recipe\_id FK, name, qty\_num, unit, grams\_num, role, food\_code TEXT)
- `steps` (id PK, recipe\_id FK, n INT, text)
- `photos` (id PK, recipe\_id FK, uri)
- `nutrition` (recipe\_id PK FK, kcal, protein\_g, fat\_g, carb\_g)

- `density_map` (name PK, grams\_per\_cup REAL, grams\_per\_tbsp REAL, grams\_per\_tsp REAL)
- 

## AI Extraction & Categorization

- On-device OCR → raw text blocks.
  - Deterministic parsing (regex + heuristics) to split Title, Yield/Servings, Ingredients, Steps.
  - Ingredient line parser (qty + unit + name) e.g., `1 1/2 cups bread flour`, `350 g water`.
  - Category classifier using keyword mapping to taxonomy.
- 

## Baker's Percentages — Spec

- **Flour weight (FW):** sum of ingredients with `role = flour`.
  - **Percent for ingredient i:**  $\text{pct\_i} = (\text{grams\_i} / \text{FW}) * 100$ .
  - **Hydration %:**  $(\text{total water grams} / \text{FW}) * 100$ .
  - Display hydration, salt %, sugar %, fat %, and per-ingredient % table when FW > 0.
- 

## Nutrition (Basic) — Spec

- Built-in mapping of common ingredients → kcal/protein/fat/carb per 100 g.
  - Map ingredients by fuzzy name; compute per serving.
  - Mark as approximate and allow per-ingredient override.
- 

## Privacy & Security

- All processing local on device (OCR, parsing, calculations). No cloud use.
- 

## Playful & Accessible UI


- Bright accent color, rounded cards, emoji/category glyphs.
  - Subtle haptics on save/favorite; confetti on first successful capture.
  - Cooking mode (large text, step-by-step), Dark mode.
- 

## Screen Mockups (Lo-Fi)


Wireframes are conceptual guides for the MVP screens. Elements shown are limited to MVP scope: capture, local store, browse/search, edit, email share, favorites, baker's %, and basic nutrition.

## 1) Home / Library (Empty → Grid)

Recipe Collector

 Search [    ]

(Empty State)

 Capture your first recipe

[Big Capture Button]

(After items exist)


Filters: [★ Favorites] [Categories ▾]

☐

Card:  Sourdough Bread    ★

Breads • 12 servings

☐

Card:  Brownies    ☆

Desserts • 9x9 pan

☐

Card:  Roast Chicken    ☆

Chicken • 4 servings



[Floating Action Button: 

## 2) Capture (Camera) → Crop/Adjust



Capture


[ Viewfinder with alignment grid ]

Hint: "Fill the page; avoid glare."

[Gallery  ]      [Shutter  ]      [Flash]

Crop & Adjust

[ Crop handles  ]    [Contrast  ]

[ Rotate  ]

[Retake]    [Continue ►]

### 3) Review & Parse (OCR → Structured)

Review & Parse
Tabs: [Text] [Structured ✓]
Structured Preview:
Title: _____ Yields: _____ Servings: _____
Category: [Breads ▾]
Ingredients
• qty unit name grams [🔗]
• 500 g bread flour 500 [🔗]
• 350 g water 350 [🔗]
Steps
1) ... [🔗]
2) ... [🔗]
[Back] [Save ▶]

### 4) Recipe Detail

Sourdough Bread	☆ (Favorite)
Breads • Yields: 1 loaf • Servings: 12	
Chips: 🇩🇪 Breads 🍷 Savory	
Ingredients	
• 500 g bread flour	
• 350 g water	
• 10 g salt	
• 100 g starter	
Steps	
1) Mix flour & water; autolyse 1 hr.	
2) Add starter & salt; folds.	
3) Shape; proof; bake 240°C 35 min.	
Baker's % (if flour present)	
Hydration 70% • Salt 2% • Starter 20%	
[View per-ingredient %]	

Nutrition (approx., per serving)  
160 kcal • 5 g P • 1 g F • 33 g C

Actions: [Edit] [Email] [Cooking Mode]

## 5) Edit Recipe

Edit Recipe

Title: [\_\_\_\_\_]

Yields: [\_\_\_\_\_] Servings: [\_\_\_] Category: [▼]

Ingredients


qty [\_\_\_] unit [▼] name [\_\_\_\_\_] grams [\_\_\_] [-]

[+ Add Ingredient]

Steps

1) [\_\_\_\_\_]

[+ Add Step]

[Cancel] [Save 

## 6) Cooking Mode

Cooking Mode


Step 1/3

Mix flour & water; autolyse 1 hr.

[◀ Prev] [Next ▶]

[Large text toggle A+] [Keep screen on]

## 7) Search & Filters

 Search: [sour...\_\_\_\_\_]

Filters: [★ Favorites] [Category ▼]



Results (title + snippet highlighting term)
• Sourdough Bread (Breads)
• Sourdough Pancakes (Breakfast)

---

## Risks & Mitigations

- **OCR inaccuracies:** Provide crop/contrast tools; quick in-line edits.
- **Ambiguous units:** Editable conversions; per-ingredient density table.
- **Nutrition mapping gaps:** Mark unknowns; allow manual mapping.
- **Performance:** Optimize image sizes; lazy lists in Compose.

---

## Implementation Notes & Estimates (MVP)

- Milestone 1: 1.5–2 weeks (capture, DB, list/detail/edit, favorites, categories)
- Milestone 2: 1 week (share, search, polish)
- Milestone 3: 1–1.5 weeks (baker's %, nutrition basics)

---

## Appendix A — Ingredient Roles (seed)

- Flour roles: all-purpose flour, bread flour, whole wheat flour, rye flour, spelt flour
- Water roles: water, milk
- Salt: fine sea salt, kosher salt
- Sugar: white sugar, brown sugar, honey
- Fat: butter, oil, shortening
- Preferment: starter, poolish, biga, levain

---

## Appendix B — Example Email Output (plain text)

Title: Sourdough Bread  
Yields: 1 loaf (12 servings)

### Ingredients

- 500 g bread flour
- 350 g water
- 10 g salt
- 100 g starter

### Steps

- 1) Mix flour and water; autolyse 1 hr.

- 2) Add starter and salt; fold every 30 min for 2 hrs.
- 3) Shape, proof, and bake at 240°C for 35 min.

Baker's %

- Hydration: 70%
- Salt: 2%
- Starter: 20%

Nutrition (approx., per serving)

- 160 kcal · 5 g protein · 1 g fat · 33 g carbs

## Recipe Collector — Product Requirements Document (MVP)

**Purpose:** Define the minimum viable product for a playful Android app that captures recipes from photos, auto-structures them, categorizes them, and stores them locally for browsing, editing, sharing, and basic analysis (baker's percentages + nutrition).

---

### Project Overview

**Goal:** Photo → Clean recipe in your library in under 30 seconds.

**Platform:** Android phone (single device), offline-first.

**Primary users:** Home cooks and hobbyist bakers.

**Non-goals:** Cloud sync, multi-user accounts, social features, complex nutrition labels, desktop/tablet UI.

---

### Level

Medium (mobile + OCR + deterministic parsing).

### Type of Project

Mobile app · On-device OCR · Data extraction & classification · Offline storage.

### Skills Required

- Android (Kotlin, Jetpack Compose, Coroutines)
- ML Kit Text Recognition (on-device OCR)
- Kotlin parsing/regex for structured extraction

- Data modeling (recipes, ingredients, steps, categories)
  - Room/SQLite
  - Basic nutrition math & baker's percentages
  - UI/UX for playful, accessible interfaces
- 

## Personas

1) **The Collector:** Snaps recipes from books, wants clean storage, quick edits, and email sharing. 2) **The Baker:** Cares about weights, hydration %, and scaling. 3) **The Tracker:** Wants basic calories/macros per serving.

---

## Success Metrics (MVP)

- **Capture conversion:**  $\geq 80\%$  of captured images yield a structured recipe without re-shoot.
  - **Edit friction:** Median time to correct a recipe  $< 2$  minutes.
  - **Findability:** Search returns a relevant recipe in  $\leq 2$  queries 90% of the time.
- 

## Key Features & Milestones (MVP only)

### Milestone 1 — Core Capture, Store, Browse

- Photo capture (camera or gallery import)
- On-device OCR (ML Kit)  $\rightarrow$  raw text
- Deterministic parsing to Recipe JSON (title, yields, ingredients, steps)
- Local database (Room/SQLite)
- Library view (grid/cards), detail view, edit view
- Favorites (toggle + filter)
- Category auto-tagging for broad groups (breads, desserts, chicken, beef, etc.)

**Acceptance criteria** - Capture  $\rightarrow$  Save flow in  $\leq 30$  seconds. - Searchable by title and ingredient.

### Milestone 2 — Formatting, Email Share, & Fun UI

- Clean formatting template for ingredients/steps
- One-tap share via email (plain-text body)
- Playful UI (microinteractions, gentle haptics)
- Basic search (title, ingredients)

**Acceptance criteria** - Email a formatted recipe from the detail screen in  $\leq 3$  taps.

### Milestone 3 — Baker's Percentages & Nutrition (Basic)

- Baker's % when flour is present (hydration, salt %, sugar %, fat %)

- Volume→weight conversions using a small internal density table
- Basic nutrition per serving (kcal, protein, fat, carbs) using a built-in mapping

**Acceptance criteria** - For flour-based recipes, hydration % and key baker's % display automatically with an "edit weights" panel. - Nutrition per serving appears with an "approximate" disclaimer.

---

## User Stories

- As a user, I can snap a photo of a recipe and get a clean, editable recipe in my library.
  - As a user, I can correct ingredient names, amounts, and steps before saving.
  - As a user, I can favorite recipes and filter by favorites.
  - As a user, I can email a recipe in a readable format.
  - As a baker, I can see baker's percentages for dough/batter recipes.
  - As a health-conscious user, I can see basic calories/macros per serving.
- 

## App Flows

1) **Capture:** Home → Capture → Camera → Crop/adjust → OCR → Parse → Review/Edit → Save → Library. 2) **Browse:** Library (filters: Favorites, Category) → Recipe Detail → Edit / Email / Favorite. 3) **Edit:** Detail → Edit (ingredients, steps, yields, categories) → Save → Recompute baker's % / nutrition. 4) **Share:** Detail → Share → Email client (plain text).

---

## Architecture

- **UI:** Jetpack Compose + MVVM (ViewModel + immutable state)
  - **Domain:** Use-cases for capture, parse, save, compute
  - **Data:** Repository pattern over Room/SQLite
  - **OCR:** ML Kit Text Recognition (on-device)
  - **DI:** Hilt (or Koin)
  - **Images:** Coil
  - **Testing:** JUnit5, MockK, Robolectric, Paparazzi
- 

## Data & Storage

- Local database: **Room/SQLite** on the device (no sync).
- 

## Patterns & Frameworks

- **Language/UI:** Kotlin + Jetpack Compose
- **Architecture:** Clean Architecture + MVVM + Repository pattern

- **Storage:** Room/SQLite, DataStore (preferences)
- **OCR:** Google ML Kit Text Recognition (on-device)
- **Imaging:** Coil
- **Testing:** JUnit5, MockK, Robolectric, Paparazzi

## Recipe Format

Adopt Schema.org/Recipe concepts in a compact JSON structure with baker's-percent and nutrition fields.

```
{
  "id": "uuid",
  "title": "Sourdough Bread",
  "source": {"type": "photo", "note": "Grandma's card"},
  "yields": "1 loaf",
  "servings": 12,
  "times": {"prep": 20, "rest": 180, "cook": 35, "total": 235},
  "categories": ["breads"],
  "tags": ["sourdough", "artisan"],
  "ingredients": [
    {"name": "bread flour", "qty": 500, "unit": "g", "grams": 500, "role": "flour"},
    {"name": "water", "qty": 350, "unit": "g", "grams": 350, "role": "water"},
    {"name": "salt", "qty": 10, "unit": "g", "grams": 10, "role": "salt"},
    {"name": "starter", "qty": 100, "unit": "g", "grams": 100, "role": "preferment"}
  ],
  "steps": [
    {"n": 1, "text": "Mix flour and water; autolyse 1 hr."},
    {"n": 2, "text": "Add starter and salt; fold every 30 min for 2 hrs."},
    {"n": 3, "text": "Shape, proof, and bake at 240°C for 35 min."}
  ],
  "photos": ["content://.../photo1"],
  "bakers": {"hydration_pct": 70, "salt_pct": 2, "starter_pct": 20},
  "nutrition": {"per_serving": {"kcal": 160, "protein_g": 5, "fat_g": 1, "carb_g": 33}},
  "created_at": 0, "updated_at": 0, "favorite": false
}
```

**Category taxonomy (seed)** - Breads; Desserts; Chicken; Beef; Pork; Seafood; Vegetarian/Vegan; Pasta/Grains; Soups/Stews; Salads; Breakfast; Sauces/Condiments

## Data Model (Room/SQLite)

- `recipes` (id PK, title, yields, servings, categories JSON, tags JSON, prep\_min, cook\_min, total\_min, favorite BOOL, created\_at, updated\_at)
  - `ingredients` (id PK, recipe\_id FK, name, qty\_num, unit, grams\_num, role, food\_code TEXT)
  - `steps` (id PK, recipe\_id FK, n INT, text)
  - `photos` (id PK, recipe\_id FK, uri)
  - `nutrition` (recipe\_id PK FK, kcal, protein\_g, fat\_g, carb\_g)
  - `density_map` (name PK, grams\_per\_cup REAL, grams\_per\_tbsp REAL, grams\_per\_tsp REAL)
- 

## AI Extraction & Categorization

- On-device OCR → raw text blocks.
  - Deterministic parsing (regex + heuristics) to split Title, Yield/Servings, Ingredients, Steps.
  - Ingredient line parser (qty + unit + name) e.g., `1 1/2 cups bread flour`, `350 g water`.
  - Category classifier using keyword mapping to taxonomy.
- 

## Baker's Percentages — Spec

- **Flour weight (FW):** sum of ingredients with `role = flour`.
  - **Percent for ingredient i:**  $\text{pct\_i} = (\text{grams\_i} / \text{FW}) * 100$ .
  - **Hydration %:**  $(\text{total water grams} / \text{FW}) * 100$ .
  - Display hydration, salt %, sugar %, fat %, and per-ingredient % table when FW > 0.
- 

## Nutrition (Basic) — Spec

- Built-in mapping of common ingredients → kcal/protein/fat/carb per 100 g.
  - Map ingredients by fuzzy name; compute per serving.
  - Mark as approximate and allow per-ingredient override.
- 

## Privacy & Security

- All processing local on device (OCR, parsing, calculations). No cloud use.
- 

## Playful & Accessible UI

- Bright accent color, rounded cards, emoji/category glyphs.
- Subtle haptics on save/favorite; confetti on first successful capture.
- Cooking mode (large text, step-by-step), Dark mode.

---

## Risks & Mitigations

- **OCR inaccuracies:** Provide crop/contrast tools; quick in-line edits.
  - **Ambiguous units:** Editable conversions; per-ingredient density table.
  - **Nutrition mapping gaps:** Mark unknowns; allow manual mapping.
  - **Performance:** Optimize image sizes; lazy lists in Compose.
- 

## Implementation Notes & Estimates (MVP)

- Milestone 1: 1.5–2 weeks (capture, DB, list/detail/edit, favorites, categories)
  - Milestone 2: 1 week (share, search, polish)
  - Milestone 3: 1–1.5 weeks (baker's %, nutrition basics)
- 

## Appendix A — Ingredient Roles (seed)

- Flour roles: all-purpose flour, bread flour, whole wheat flour, rye flour, spelt flour
  - Water roles: water, milk
  - Salt: fine sea salt, kosher salt
  - Sugar: white sugar, brown sugar, honey
  - Fat: butter, oil, shortening
  - Preferment: starter, poolish, biga, levain
- 

## Appendix B — Example Email Output (plain text)

Title: Sourdough Bread  
Yields: 1 loaf (12 servings)

### Ingredients

- 500 g bread flour
- 350 g water
- 10 g salt
- 100 g starter

### Steps

- 1) Mix flour and water; autolyse 1 hr.
- 2) Add starter and salt; fold every 30 min for 2 hrs.
- 3) Shape, proof, and bake at 240°C for 35 min.

### Baker's %

- Hydration: 70%

- Salt: 2%
- Starter: 20%

Nutrition (approx., per serving)

- 160 kcal · 5 g protein · 1 g fat · 33 g carbs