SMART EXPIRY DATE TRACKER WITH FOOD RECOMMENDATIONS

Mr Pradeep Nayak[1], Shreeraksha K S[2], Shreya[2], Shriya S[2], Sinchana[2]

[1]*Professor, ISE, Alva's Institute of Engineering & Technology, Moodbidri , India*

[2]*Student, ISE, Alva's Institute of Engineering & Technology, Moodbidri , India*

ABSTRACT

Food waste from forgotten or untracked perishable items is a major global problem. Systems that track expiry dates can reduce waste and improve safety; combining them with food-recommendation engines (recipe suggestions based on on-hand ingredients and imminent expiries) increases the likelihood of timely use. This paper reviews existing approaches to expiry detection and tracking (barcodes/QR, OCR, IoT sensors), surveys food-recommendation methods, and consolidates design choices into a practical mobile app implementation blueprint. Key challenges — OCR reliability on dot-matrix labels, sensor cost and integration, privacy, and user adoption — are discussed, and future research directions are proposed.

Keyword: *Expiry Tracker, Food Recommendation, IoT, Machine Learning, Mobile Application, Food Waste Reduction.*

1. INTRODUCTION

The drive to find innovative ways to reduce household and retail food waste has resulted in some approaches to better track the lifespan of food and notify the user that something has expired. Digital systems are based on a bar/QR code, OCR of printed date or IoT sensors that monitor storage conditions (temperature, humidity) with direct bearing on shelf life. The complementary area to this is food recommendation systems that convert mixture of available raw materials into meals (recipes, meal plans). Bringing them together (a) means you get notified of upcoming expiry and (b) starts suggesting what you can cook with them; which maximises consumption and reduces waste. Some recent mobile apps and smart-fridge projects have shown that this can be achieved but none so far have succeeded with a highly-interfaced fully integrated and robust solution at scale.

2. METHODS HOW LITERATURE WAS SEARCHED FOR

The search included: Academic surveys about food recommender systems and ingredient recognition.
More recent project papers and conference articles on smart expiry systems, IoT fridge solutions and mobile applications. Existing mobile apps and product descriptions (BEEP, Smart Expiry apps) were considered for inspiration. Search terms were focused to "expiry date tracker", "food expiry tracker IoT", "food recommender mobile app", "recipe recommendation systems", "smart fridge expiry". Systematic reviews and well-documented implementations were prioritised.

3. EVALUATION OF THE METHODS TO TRACK EXPIRY

3.1 Barcode / QR / manual input

Easiest to implement: displayed product barcodes, or reading the QR codes issued at purchase. There are many consumer apps that do barcode lookups along with either you manually entering the expiry date or providing suggested shelf-life defaults. This method is low cost and driven by the user but requires either manual expiry entry or extensive product database coverage.

3.2 Optical Character Recognition (OCR)

OCR attempts to read printed expiry dates from product packaging. Looks promising (no manual input needed) yet OCR fails at reading the expiry stamps due to dot-matrix fonts, low contrast, curved surfaces, and non-uniformity in date styles leading to abysmal error rates unless we train our datasets and model them specifically for expiry fonts as evidenced by numerous papers and project reports. Pre-processing (contrast enhancement, morphological ops) with regex filtering helps; the methods regularly proving to be insufficient.

3.3 IoT sensors and environmental monitoring

Objectively quantify the conditions that hasten spoilage with IoT sensors (temp, humidity, gas sensors, biosensors) - platforms that many times in conjunction with shelf-life models / ML forecasting can provide real remaining life instead of trusting printed dates. However, sensor deployment comes with added cost and hardware complexity.

3.4 Computer vision/image recognition for ingredient identification

Employ image classification and object detection (CNN) for identifying items inside of your fridge/pantry, enabling inventory management without barcode/OCR. Effective for distinctive food products, trouble lies in items packed, occlusion and similar-looking products. Combining vision with barcode / OCR increases accuracy.

4. FOOD RECOMMENDATION STRATEGIES REVIEW

4.1 Content-driven based on the similarity of ingredients

Recommend recipes that include the ingredients the user has in inventory (matching on ingredient tokens). These could be as simple as TF-IDF / cosine similarity on ingredient lists, and as lively as recent work on food Embedding models. These primarily work when ingredient labels are definitive.

4.2 Collaborative filtering / Personalization

Personalize recipe suggestions for the user based on user preferences, dietary constraints and interactions. Having a good number of users and interactions data is ideal; cold start handling are limited otherwise.

4.3 Image based recognition + recipe retrieval

Identify ingredients through the camera image, then query recipe databases for matches. Various mobile systems have also implemented real-time recognition and recipe retrieval.

4.4 Hybrid models & health-aware recommenders

Content (ingredients) + collaborative signals + health/diet constraints (calorie, allergy) for well-rounded suggestions. In recent years, systematic reviews have looked at the choices made in various works (of different types and in different domains)

5. INTEGRATION: EXPIRY TRACKING + RECS – SYSTEM PATTERNS

This is all about practical integration patterns:
1. See Barcode/DB + Recommender: Scan barcode → get product + default shelf life from DB → Add to inventory → recommender queries inventory for recipes. You will require a lot of database completeness; the cost of hardware is lower on this one.
2. OCR Read Exp. + CV + Recommender: Use image OCR for the expiry date and CV to identify the item as a fallback to manual entry in case OCR fails. Works offline but requires robust preprocessing and model training for dot-matrix expiry text.
3. Sensor-enhanced Predictive Model + Recs: Environmental sensors feed ML models that predict remaining shelf life; recommender prioritises recipes for items predicted to expire sooner. Most accurate but comes with more cost and complexity.

6. MOBILE APP IMPLEMENTATION PROPOSAL (ARCHITECTURE & COMPONENTS)

A practical blueprint designed for a student/engineering project and scaled it for production below.

6.1 High-Level Architecture

Mobile app (Android/iOS) — camera UI, barcode/QR scan assist, manual entry UI, inventory list, recipe suggestions, notifications.
Edge modules (on-device) — barcode decoding, image preprocessing, offline OCR/CV (lightweight models), local cache.
Cloud backend — authentication, user inventory DB, product metadata (mapping barcode → product name), defaults of shelf life, recipe database, recommenders, notification scheduler.
Optional IoT gateway — process telemetry from refrigerator sensors to the cloud for shelf-life prediction.
External APIs — recipe APIs (Spoonacular /Edamam), barcode lookup APIs, push notification services.

6.2 Modules and Functions

Inventory module: create/update/delete item, store: product id, name, quantity, acquired date, expiry date (manual or computed), image, location (fridge/pantry).
Acquisition methods:
Barcode/QR scan → lookup product DB → suggest default shelf life → allow user edit.
Camera: CV to identify the item; OCR to extract printed expiry; ask user to confirm.
Manual entry.
IoT sensor input for items in smart containers (optional).
Expiry estimation & forecasting:
If printed expiry exists → save date.
If not → calculate expected shelf life using the ML model from product category + purchase date + sensor inputs (e.g., survival regression or RNN).
Recommender engine:
Primary tactic: match on content — recipes having near-expiry ingredients rank higher.
Personal independent layer: recipes matched with user diet/profile; collective signals as data grows.
Notifications & UI:
Local push notifications x days before expected expiry (configurable).
A "use soon" shelf in-app with recipe suggestions (adds recipe ingredients to shopping list in one tap).

6.3 Example Technology Stack

Mobile: React Native / Flutter(x-platform) | (native) Kotlin, Swift.
On-device CV/OCR: Tesseract (for OCR) with pre/post processing; TensorFlow Lite / PyTorch Mobile for basic image models. Note: Tesseract is challenging with dot-matrix expiry fonts unless trained.
Backend: Node.js / Python Flask; PostgreSQL as DB or Firebase for quick bits.
Recommender: TF-IDF + embeddings along with ranking service in Python (scikit-learn / TensorFlow).
Cloud: Firebase/GCP/AWS for storage, functions, push notifications.
Barcode lookup: Open Food Facts / (paid) barcode APIs.

6.4 UI/UX Recommendations

Make the process of adding items frictionless: scan → verify → save, in one tap.
Feature a "expires soon" card with a 1-tap recipe suggester. Ads in order of most urgent.
Preferences section: user edits lead time of notification, dietary filters, and pantry categories.
Accessibility: large fonts, voice prompts for the visually impaired (IJET demo project kept audio feedback in focus)

7. EVALUATION METRICS AND EXPERIMENTAL SETUP

To verify the system:

- OCR accuracy on expiry stamps (precision/recall) — test with a curated dataset of dot-matrix expiry prints.
- Inventory match rate: correctness of item identification (barcode/CV) vs ground truth.
- Recommendation utility: user acceptance rate of recommended recipes, time-to-use items.
- Waste reduction: measured change in expired items per household over test period.
- Pilots can be run with volunteer households; sensors can be introduced in a subset for comparison.

8. CHALLENGES AND OPEN ISSUES

- OCR reliability for expiry stamps: many studies and projects show poor out-of-the-box OCR performance on dot-matrix fonts — retraining or specialized preprocessing needed.
- Barcode DB coverage: not all items are in public DBs; user correction is often required.
- Sensor cost & deployment: while sensors improve prediction, the cost may prevent household adoption; retrofit solutions are needed.
- User behaviour and friction: adoption depends on how seamless item entry and suggestions are. Many consumer apps fail due to high manual effort.
- Privacy & security: inventory and consumption data are personal; secure storage and transparent data policies are required.
- Recipe-ingredient matching quality: matching granularity (e.g., ‘tomato’ vs ‘cherry tomato’) affects recommendations.

9. FUTURE DIRECTIONS

- Standardized expiry encoding on packaging (machine-readable) would greatly reduce friction.
- Edge AI models specialized for dot-matrix expiry fonts (public datasets) would improve OCR success.
- Integration with smart appliances (fridges with internal cameras) to auto-scan contents.
- Behavioural nudges and gamification to increase user follow-through on recommendations.
- Health-aware recommendation that considers diet, allergies, and nutrient balancing.

CONCLUSION

Combining expiry-tracking with a recipe recommender in a mobile app is feasible and promising for reducing household food waste. Practical implementations should use a hybrid acquisition strategy (barcode + CV + manual fallback), prioritize low-friction UX, and optionally supplement with sensors for higher accuracy. Key technical hurdles (OCR of expiry stamps, database coverage, sensor cost) must be addressed for robust, wide adoption. Several recent papers and prototypes confirm the viability of such systems and provide reusable building blocks for a student-scale mobile app implementation.

REFERENCES

1. Cloud-based Architecture of a Smart Expiry System with IoT Device (project paper).
2. IJET — "Expiry Date Tracker" (project report, Jan–Feb 2025) — discussion of OCR challenges and usability.
3. Bonde Vik et al., Systematic review on food recommender systems (2024) — survey of methods and evaluation.
4. Recipe IS / MDPI (2023) — ingredient recognition & recipe recommendation survey.
5. Mobile recipe recommendation using object recognition — Maruyama et al. (mobile CV → recipe).
6. BEEP and other consumer expiry-tracking apps (product pages).
7. IoT-based food inventory and spoilage detection project papers (IJIRT/IJARCCE samples).