

Agentic AI-Powered Expiry Date Management System

Kale Shruti Sunil, Prof. Bangar A.P, Prof. Bhosale S.B., Prof. Karad V. A

Dept. of Computer Engineering
Jaihind College of Engineering, Pune, India

Abstract — Food and medicines that pass their expiry dates cause avoidable financial loss, health risk and environmental harm, yet most expiry-tracking tools are passive reminder lists that depend on manual data entry. This paper proposes an Agentic AI-Powered Expiry Date Management System in which a team of cooperating, tool-using AI agents autonomously captures product labels, extracts and validates expiry dates with optical character recognition (OCR), estimates shelf life when no date is printed, ranks items by spoilage risk, and acts on the result by sending timely alerts, suggesting recipes and routing surplus items to donation partners. A central orchestrator agent plans tasks, calls specialised agents and learns from user feedback, while a shared knowledge layer keeps inventory and food-safety rules consistent. We describe the architecture, the riskscoring model and the agent workflow, and report an illustrative simulation of 500 households over 60 days in which the agentic prioritisation policy reduced wasted items from 21.5% to 14.8% ($c = 0.7$) and 13.3% ($c = 0.9$). The modular design allows each agent to be upgraded or deployed independently in households, retail shelves and hospital pharmacies.

Keywords— Agentic AI, Expiry Date Management, Multi-Agent System, Optical Character Recognition, ShelfLife Prediction, Large Language Models, Food Waste Reduction..

I. INTRODUCTION

Roughly one-third of the food produced for human consumption is lost or wasted along the supply chain [1], and in 2022 households alone generated an estimated 631 million tonnes of food waste worldwide [2]. A large share of household waste arises because people forget what they have bought or cannot judge how long it will stay safe. The same problem affects over-the-counter medicines, cosmetics, and the stock of small shops, hostels and clinics, where expired items also create safety and compliance risks.

Existing expiry-tracking applications mostly act as digital lists: the user must scan or type every item, and the application only displays a countdown. Such tools are reactive, ignore items without a printed date, and do not help the user decide what to do with an item that is about to expire. Recent advances in large language models (LLMs) and tool-using agents [3], [4] make a more proactive design possible: an agent can perceive its environment (images, receipts, inventory), reason over goals, call tools and take actions with limited supervision.

This paper presents an Agentic AI-Powered Expiry Date Management System. Instead of a single monolithic model, the system uses several specialised agents coordinated by an orchestrator. The main contributions are: (i) a modular multi-agent architecture that covers the full life cycle from label capture to consumption or donation; (ii) a hybrid expiry extraction pipeline that combines OCR with rule-based validation and LLM-assisted parsing, falling back to learned

shelf-life estimation when no date is present; (iii) an urgency-based risk score that drives reminders and recipe suggestions; and (iv) a simulation-based evaluation of the prioritisation policy.

II. RELATED WORK

Food-loss research has quantified waste at scale [1], [2] and highlighted the role of date labelling and household planning. Computer-vision approaches detect printed dates using text detection and recognition networks; Tesseract-style engines and deep scene-text recognisers are widely used, but are sensitive to curved packaging, dot-matrix printing and low contrast [5], [6]. Object-detection models such as YOLO have been used to localise date regions before recognition [7]. Shelf-life prediction models typically use storage temperature, food category and packaging as inputs.

On the agent side, ReAct showed that interleaving reasoning with tool calls improves decision making [3]; retrieval-augmented generation (RAG) grounds model answers in external documents such as food-safety guidelines [8]; and multi-agent frameworks such as AutoGen let several LLM-based agents converse and divide work [4], [9]. Classical multi-agent theory provides the vocabulary of autonomy, reactivity and social ability [10]. To our knowledge, few systems combine these ideas specifically for expiry management with an action loop that ends in reuse or donation.

III. SYSTEM ARCHITECTURE

The system follows a layered, agent-oriented design shown in Fig. 1. The user interacts through a mobile or web application, which forwards requests and events to an orchestrator agent. The orchestrator decomposes goals (for example, "add today's groceries" or "what should I cook tonight?"), selects the appropriate specialised agents, merges their outputs and writes results to a shared knowledge layer.

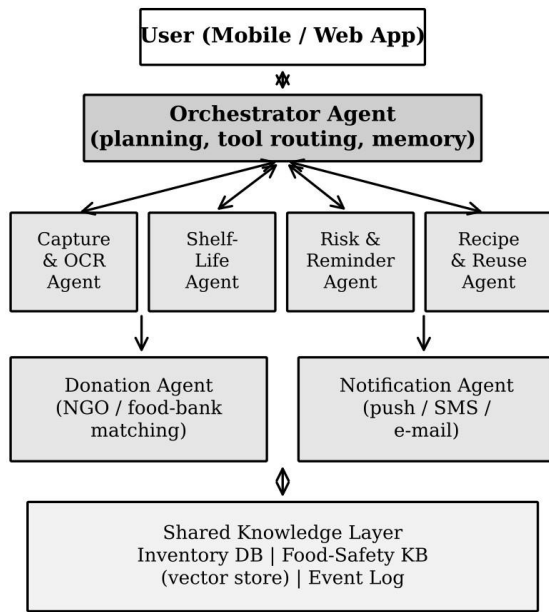


Fig. 1. Architecture of the agentic expiry date management system.

1. Agents and Responsibilities

Table I summarises the agents. Each agent has a narrow goal, a set of tools and a defined input/output schema, which isolates failures and lets agents be replaced independently.

Table 1: Agents, Tools and Outputs

Agent	Main tools	Output
Capture & OCR	Camera, barcode / receipt reader, OCR engine	Item name, raw date text
Shelf-Life	Category rules, regression / ML model, KB retrieval	Estimated expiry date, confidence
Risk & Reminder	Risk-score function, scheduler	Ranked list, alert plan

Recipe & Reuse	LLM with RAG over recipes	Recipes using nearexpiry items
Donation	Partner directory, route / time-window check	Donation offer or pickup request
Notification	Push, SMS, e-mail APIs	Delivered alert, user response
Orchestrator	LLM planner, memory, tool router	Task plan, final response

2. Shared Knowledge Layer

Three stores are shared by all agents: an inventory database (item, quantity, purchase date, storage location, expiry date and its source), a food-safety knowledge base (typical shelf life by category and storage condition, "useby" versus "best-before" semantics), and an event log used for audit and feedback learning. The knowledge base is indexed in a vector store so that agents can retrieve relevant rules through RAG [8].

IV. METHODOLOGY

1. Expiry Date Extraction

Fig. 2 shows the extraction workflow. The Capture & OCR agent pre-processes the image (denoising, deskewing, contrast normalisation), locates candidate date regions, and runs OCR.

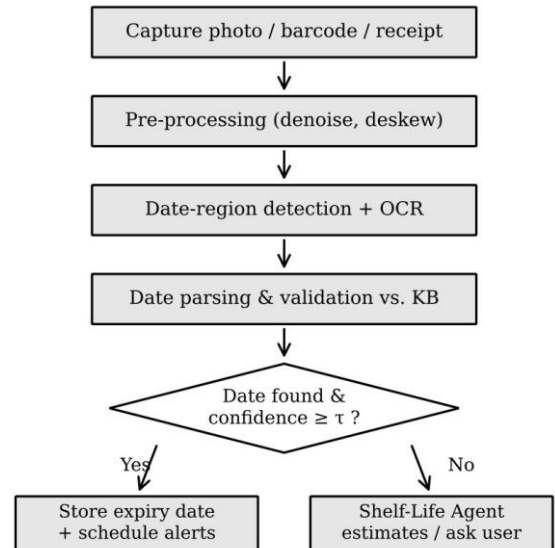


Fig. 2. Expiry date detection and fall-back workflow.

Raw strings such as "EXP 12 03 26", "BB: 03/2026" or "USE BY 12MAR26" are normalised by a parser that combines regular expressions with an LLM prompt constrained to return

a date in ISO format and a confidence value. A date is accepted only if it passes validation rules: it must be a real calendar date, later than the manufacture or packing date when present, and within a plausible range for the product category from the knowledge base. If recognition confidence is below a threshold τ , the agent re-captures the image from another angle or asks the user to confirm.

2. Shelf-Life Estimation

Loose produce, bakery items and home-cooked food have no printed date. For these items the Shelf-Life agent predicts remaining life from category, storage condition (room, refrigerator, freezer), packaging and days since purchase. The estimate is the category prior from the knowledge base, adjusted by a learned correction. In the prototype, the correction is updated from user feedback ("it was still good", "it spoiled early"), so the estimates become personalised over time.

3. Risk Scoring and Prioritisation

For every item i the Risk & Reminder agent computes an urgency-weighted score

$$R_i = w_1 \cdot \exp(-\Delta_i / \tau) + w_2 \cdot \pi_i + w_3 \cdot (1 - \kappa_i) \quad (1)_s$$

where Δ_i is the number of days left before expiry, τ is a time scale, π_i is the perishability class of the item (0 to 1), κ_i is the confidence of its expiry date, and w_1 – w_3 are weights. Items are sorted by R_i (an extension of the first-expired-first-out principle); an alert is issued when R_i crosses a threshold, and the top-ranked items are passed to the Recipe & Reuse agent.

4. Agentic Action Loop

The orchestrator follows a perceive–reason–act–learn loop. It (1) perceives new inventory events or a daily timer; (2) reasons over the ranked list and the user's context (preferences, allergies, household size); (3) acts through tools by sending a reminder, generating a recipe that uses two or three near-expiry items, or offering an item to a donation partner when it cannot be consumed in time; and (4) learns from the user's response by updating the shelf-life corrections and the alert timing. Irreversible or sensitive actions, such as sending a donation request, require explicit user confirmation, and food-safety rules from the knowledge base always override LLM suggestions.

5. Safety, Privacy and Reliability

Because LLMs can hallucinate, expiry dates are never accepted from free-form model output alone: every date is validated by deterministic rules and shown to the user with its source

(printed, estimated or manual). Images are processed on the device or stored encrypted, and only the minimum data are shared with partner organisations. Each agent call is logged to support auditing and error recovery.

V. EXPERIMENTAL EVALUATION

1. Simulation Setup

As a first step before a field study, we simulated the prioritisation policy of the system. Each of 500 simulated households was run for 60 days. Each day a Poisson distributed number of items (mean 3.0) was bought from five categories (dairy, bakery, produce, meat and packaged items) with category-specific shelf lives of 2–40 days, and a Poisson number of items (mean 2.2) was consumed. In the baseline, the household consumes a random unexpired item. In the agentic setting, with probability c (reminder compliance) the household follows the system's recommendation and consumes the item closest to expiry, otherwise it behaves as in the baseline. An item that expires unconsumed counts as waste. The "tracker only" case uses $c = 0.4$ to model passive reminders.

2. Results

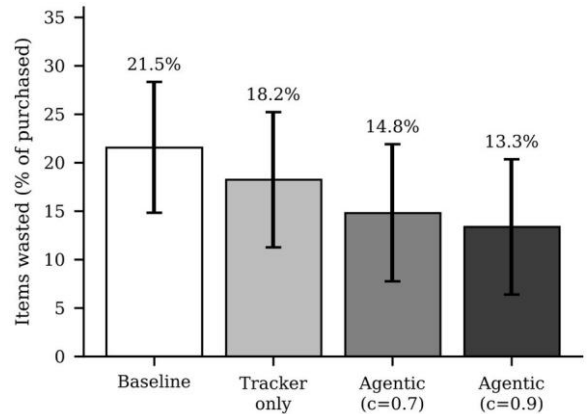


Fig. 3. Wasted items (% of purchased) under four policies (mean $\pm$ SD, 500 runs).

Fig. 3 shows that the baseline wasted 21.5% of purchased items, passive tracking reduced this to 18.2%, and the agentic policy reduced it to 14.8% at $c = 0.7$ and 13.3% at $c = 0.9$, i.e. relative reductions of 31% and 38%. The residual waste is mainly caused by over-purchasing, which the prioritisation policy alone cannot remove; this motivates the donation and purchase-planning features of the system.

3. Limitations

These results come from a synthetic simulation with simplified assumptions about purchasing and consumption behaviour, and they should be read as an illustration of the policy's effect, not as a measurement on real users. Field trials, OCR accuracy on a labelled image set, and a user study are planned as future work.

Applications and Future Scope

The same architecture can be applied to (i) households and hostels, (ii) retail and kirana stores, where shelf-edge scanning and markdown suggestions can reduce loss, (iii) hospital and community pharmacies, where expired medicines are a safety risk, and (iv) food banks and restaurants, where donation logistics are time-critical. Future work includes fine-tuning a date-recognition model on curved and dot-matrix labels, integrating IoT temperature sensors to refine shelf-life estimates, supporting regional languages in voice and text, and measuring behavioural impact through a controlled pilot study.

VI. CONCLUSION

We proposed an Agentic AI-Powered Expiry Date Management System that turns expiry tracking from a passive list into an autonomous workflow of capture, validation, prediction, prioritisation and action. The modular multi-agent design, deterministic validation around LLM components and human confirmation for sensitive actions aim to make the system practical and trustworthy. An illustrative simulation indicates that following the system's priority ordering can substantially reduce expired-item waste; real-world validation is the next step.

Acknowledgment

The authors thank their institution and colleagues for their guidance and support during this work.

REFERENCES

1. FAO, "Food Wastage Footprint: Impacts on Natural Resources – Summary Report," Food and Agriculture Organization of the United Nations, Rome, 2013.
2. UNEP, "Food Waste Index Report 2024," United Nations Environment Programme, Nairobi, 2024.
3. S. Yao et al., "ReAct: Synergizing Reasoning and Acting in Language Models," in Proc. Int. Conf. Learning Representations (ICLR), 2023.
4. Q. Wu et al., "AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation," arXiv:2308.08155, 2023.
5. R. Smith, "An Overview of the Tesseract OCR Engine," in Proc. 9th Int. Conf. Document Analysis and Recognition (ICDAR), 2007, pp. 629–633.
6. B. Shi, X. Bai and C. Yao, "An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 11, pp. 2298–2304, 2017.
7. J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788.
8. P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020.
9. J. S. Park et al., "Generative Agents: Interactive Simulacra of Human Behavior," in Proc. ACM Symp. User Interface Software and Technology (UIST), 2023.
10. M. Wooldridge, An Introduction to MultiAgent Systems, 2nd ed. Chichester, U.K.: John Wiley & Sons, 2009.