Smart Food Saver: Avoid Needless Expenses - A Vision for Zero Waste

Divya Lakshmi K[1], Harini S[2], Janani R[3], Mr. F. Gnanapriyan[4]
[1,2,3]*Department of Artificial Intelligence and Data Science, Rajalakshmi Institute of Technology, Chennai, Tamil Nadu, India*
[4]*Assistant Professor, Department of Artificial Intelligence and Data Science, Rajalakshmi Institute of Technology, Chennai, Tamil Nadu, India*

***Abstract*—Food waste is a major problem in most households, often caused by poor awareness of existing grocery inventory. As a result, users tend to repurchase items they already have at home, leading to unnecessary expenses and wastage. The Smart Food Saver, a modular grocery management system that assists customers in monitoring their home products and avoids unnecessary spending, is proposed in this research work. The system allows customers to manually enter products and automatically update the list based on consumption and potential overbuying scenarios. The semi-automatic receipt scanning system of the proposed system employs optical character recognition (OCR) to reduce manual processing. Before the products are entered into the system, they can be verified by the customers.**

***Index Terms*—Optical Character Recognition, Food Waste Reduction, Grocery Management, Inventory Tracking, Flask Application, Overbuy Prediction, Household Inventory System.**

I. INTRODUCTION

Food waste is a major global issue, with households contributing significantly due to poor inventory awareness and unnecessary repurchasing [1], [2], [6]. This is not usually the case because people do not intend to waste food, but they do not know what they already have at home. People do not monitor what they eat and what they already have at home, so they go out and purchase food, packaged products, and other items.

Because of the rise in digital solutions, many applications have been developed to assist users with their grocery planning and inventory management. However, most of the current applications have some drawbacks. Some applications require too much input and are therefore time-consuming and hard to use. Most of the current applications are designed for business use in inventory management and are not simple enough to use at home. Thus, it is difficult for these applications to gain long-term acceptance by users.

There is a need to develop a system which is user-friendly and sustainable in the context of both manual control and intelligent control to address the constraints that are identified. The system to be developed will give users the ability to monitor the items they use, which in turn will give the users the maximum benefits out of their buying activities.

Smart Food Saver is a food management system that provides users with the opportunity to manage their inventory effectively. In addition, the module tracks the products that users purchase and store, which helps the user to be aware of situations where they are about to overbuy the products. Furthermore, the module utilizes Optical Character Recognition, which makes it easy to scan receipts, enabling customers to access their receipts and inquire about the goods in the grocery store.

Besides being a helpful tool for managing household groceries, Smart Food Saver aims to reduce food waste and encourage sustainable consumption via an accurate, scalable approach.

II. LITERATURE REVIEW

The problem of food wastage is a problem of great impacts, not only to society but also to the environment. In this context, the study done by Parfitt, Barthel, and Macnaughton indicates that a large percentage of food wastages emanates from the level of consumption. This is due to the fact that consumers handle food haphazardly. In this regard, consumers have the habit of discarding food not due to the fact that the food is bad, but due to the fact that they have forgotten that the food is already available at their respective homes. Research conducted by Gustavsson, Cederberg, and Sonesson on behalf of the Food and Agriculture Organization (FAO) reveals that if home planning and management are done properly, it can be of immense help in reducing unnecessary food waste.

Computerized systems have been investigated by researchers as a way to efficiently handle the inventory of household items. These systems assist people in monitoring their grocery buys and also keep them updated on their existing stock. These systems are very helpful. But, as evident from previous studies, they tend to have problems with long-term usability. Users tend to get tired of entering information manually again and again. This leads to an increase in errors, and eventually, users abandon the system altogether. This has resulted in an increase in research studies on ways to reduce user effort without affecting accuracy.

One of the way automations is done is through Optical Character Recognition (OCR) for digitizing receipts. Ray Smith, in his study on the Tesseract OCR engine, discovered that OCR works well for reading printed text in a controlled setting. OCR-based receipt processing accelerates data entry by reading item information directly from grocery receipts. But in practical applications, there are many challenges, such as tilted pages, varying fonts, low contrast, and background noise, that can affect accuracy in a home setting.

The challenges of receipt extraction in practical applications are evident from the ICDAR Robust Reading Competition, organized by Karatzas et al., which evaluates OCR algorithms on noisy and unstructured documents. Their results indicate that extracting structured information from receipts is still a challenge due to variations in layout and print quality. Consequently, fully automated OCR algorithms do not always work correctly without human intervention [3], [4], [7].

Apart from the digitization of receipts, there have been researches on predictive models that can interpret user behaviour in households. Rule-based predictive models rely on predetermined rates to predict the rate at which stock will be depleted. The models are quite easy to understand and do not need to be complex, making them the best for a household where the behaviour of the consumer can keep fluctuating. However, models like machine learning may need a lot of data, which is not possible in a typical household setting.

Another study makes it clear that the key to reducing food waste to a minimum relies heavily on user behaviour and the right decision support. It is not sufficient to simply show what is in the inventory. The value of systems increases when they are able to offer useful information, such as the current level of stock, the last purchase date, and the estimated life span of the product.

Computerized systems have been researched as a possible solution for efficiently managing household inventory. The computerized systems enable users to keep track of their grocery purchases and the current level of inventory. Previous research has revealed that the computerized systems have always had usability issues. The usability issues arise because of the potential to create user fatigue and frustration resulting from the repetitive process of manual data entry. In order to come up with a solution, it is vital to ensure that the product database is in good condition and verification processes by the users are well-received to guarantee accuracy and reliability of data. This will bring into focus the semi-automatic systems, which enable users to verify data whenever they may choose to [8], [9].

The different areas of research are inventory management, analysis of food wastage, scanning of receipts, and consumption tracking. However, the research does not offer a comprehensive solution where these aspects are incorporated together. It is quite clear from the literature study that the system needs to incorporate various modules such as effective inventory management, the application of OCR for the automation of the inventory, and the development of a smart assistance feature once sufficient details are obtained. Such research concepts were reflected as the basis of the Smart Food Saver system

III. SYSTEM METHODOLOGY

A. System Overview

The Smart Food Saver system assists households in managing groceries and prevents food waste. The system is based on inventory management, which needs very less human involvement and uses smart assistance for making decisions. The system combines manual and semi-automatic methods to make sure that it is practical in life.

The system allows users to develop new features in the future. The system maintains organized records of products, purchases, and inventory levels with a relational data model. With this, users can store data, retrieve data quickly, and calculate inventory levels precisely. The system's most basic features are ensured to be founded on accurate and trustworthy data by strong database.

A web interface that supports both manual and semi-automated operations can be used to control the system. In situations where receipts are not available, the manual functionality assists users in entering purchases. This reduces the chance that when inventory levels are changed, errors in quantity or date could occur.

To make the task of the users easier, the system also incorporates semi-automatic receipt scanning using Optical Character Recognition. The users validate the obtained receipt information before it is utilized in the inventory. This is done to ensure that the information is accurate even if there are multiple templates of the receipts. With information about the level of inventories, the system would be able to predict consumption patterns and notify users of possible options for over-purchase so that users can make well-informed purchase decisions.

One such future development in the system is the Smart Kitchen Assistant. After the required information is correctly fed into the inventory, this module in the system is able to provide information [10], [11].

B. System Architecture and Workflow

The Smart Food Saver system is based on a modular approach where users can interact, process, store, and even decide using a single module. The users of the Smart Food Saver system are able to interact with this application through a web interface where they are able to input the items manually or upload receipts from various supermarkets. All this data is sent to a backend, which uses Flask and checks if the quantity, date, and description are accurate before any processing is done. This is done to ensure that any false or misleading information is not included in the inventory processing.

The correct data is then transmitted to a SQLite database that links product data, purchase data, and inventory data. The inventory processing module periodically retrieves the purchase data stored in the database to determine the current status of the inventory and estimate the remaining inventory based on user-submitted usage rates. The estimates are then used to determine the expiration dates of the products and the times of overbuying. When uploading receipts, the OCR module transmits text data from images of bills to the backend for user validation, regardless of the type of receipt [16], [18].

The user is able to see the updated stock levels, purchase history, and notifications through the interface. The interface provides a smooth flow of information and ensures that all tasks are unique. The interface also provides scalability of the system. This means that it is able to handle new tasks such as automated consumption tracking and kitchen assistance with minimal modification.

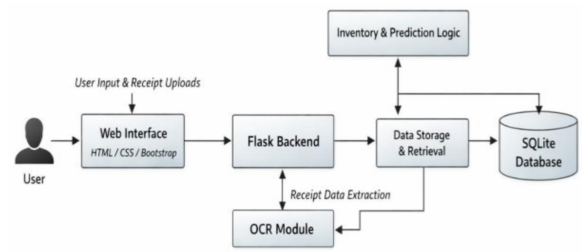

Figure 1. System Architecture of the Smart Food Saver Application.

Figure 1 describes the design of the Smart Food Saver system. The web interface will allow users to input their grocery information either manually or by scanning receipts. The web interface connects to the processing part of the Smart Food Saver system, which is the Flask backend. The validation of the input, inventory, consumption prediction, and overbuying predictions are the main tasks of the system's backend part. The system's backend is also responsible for storing all the information on the product, purchase, and inventory in an organized SQLite database.

The function of the OCR component in the system is the processing of the information from the receipts uploaded in the system and transmitting the processed information to the backend to validate and store the information. The system is made in a way that it offers a separation of user interaction in the system, processing in the system, automation in the system, and storage in the system. The system is made in a way and designed in a way to ease the development process. The system allows developers to create new features without affecting the existing ones. This is in preparation for future upgrades such as advanced prediction algorithms and kitchen automation.

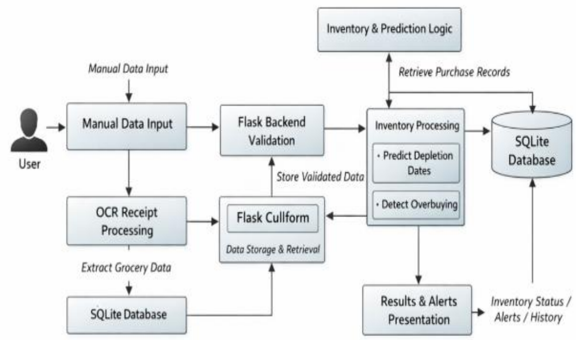

Figure 2. Data Flow diagram of the Smart Food Saver system illustrating user input, receipt processing, inventory computation, and result visualization.

The Data Flow Diagram of the Smart Food Saver system is shown in Figure 2. The data flow diagram shows the flow of data in the system. The flow of data is shown from the users to the system components and the database. The process begins at the user level where the data is inputted into the web interface either through manually inputting the product and purchase details or uploading a grocery receipt. At the manual process, the inputted data is received by the application layer where the data is verified by the Flask backend for the grocery receipt for its correct format, amount, and date. The data is then stored in the SQLite database in the form of correctly formatted product and purchase details. At the receipt upload process, the received image is processed by the OCR module where the details such as the names and amounts of the products are extracted. The inventory processing module is tasked with the mandate of fetching the purchase information from the database and determining the current level of the inventory. The system uses consumption estimates to determine when the products will be depleted and when there is a chance of overbuying. The processed information is then transmitted back to the presentation layer, where the system presents the information to the user. The flow of information ensures that input handling, processing, storage, and output are treated separately [19].

C. Experimental Setup

The project was implemented using Python, with Flask handling the backend processes of the project to ensure the smooth running of all the components of the system.

SQLite was used in the management of the structured data of products, purchases, and levels of inventory. The front-end of the project was implemented using HTML, CSS, Bootstrap, and Jinja2 templates to design a simple, mobile-friendly, and user-friendly interface.

The system was developed and tested on normal personal computers running Windows 10 or Windows 11 operating systems. The system was also tested on modern web browsers such as Google Chrome and Microsoft Edge. The hardware components of the system were computers running Intel Core i5 or i7 processors and 8 to 16 GB of RAM, which is normal in home and academic computers. There is no need for hardware implementation.

For the purpose of usability testing, a data set was created to simulate the grocery activity of a family. The data set included more than 120 different product entries, which included perishable products such as milk and vegetables, and non-perishable products such as rice and processed foods. More than 250 purchase entries were made with random dates.

Several validation tests were conducted during the usability testing to ensure the reliability of the system. Input validation tests allowed verification of the functionality of the system to process incorrect data entry, dates, missing data, and repeated entries. Database integrity tests allowed verification of the functionality of the system to process the consistency of relationships between products, purchases, and inventory entries after several transactions. The accuracy of the system to determine inventory was verified by manually checking the system-calculated values for selected inventory entries.

Table 1. System Specifications for Implementation Environment

Specification Category	Description
Operating System	Windows 10/11
Processor	Intel Core i5 / i7
RAM	8–16 GB
Storage	Minimum 5 GB free space
Programming Language	Python 3.10+
Backend Framework	Flask
Database	SQLite (SQLAlchemy ORM)
Frontend Technologies	HTML, CSS, Bootstrap, Jinja2
OCR Library (Future Phase)	EasyOCR / Tesseract
Browser Used	Google Chrome / Microsoft Edge
Development Environment	Visual Studio Code
Additional Libraries	SQLAlchemy, Jinja2, Werkzeug
Hardware Requirement	Standard laptop/desktop (No GPU Required)

The criteria for evaluation have been developed to check the correctness, responsiveness, and usability of the system. Since the proposed work is related to the system, the criteria for evaluation are the reliability of data entry, correctness of inventory aggregation, responsiveness of critical operations, and usability feedback. The criteria for evaluation clearly show that the system is reliable in its normal household usage.

The performance of the system has been checked by calculating the response times for normal user operations like entry of products, entry of purchases, viewing of inventory, and uploading of images of receipts. The average response time was always less than one second, ensuring smooth interaction with the user without any delay. Further, the extreme conditions such as the number of purchases being high, the dates of purchases being old, and the products being entered repeatedly have also been tested to ensure the reliability of the system. The system has proved to be reliable, with the correct inventory information.

D. Evaluation Metrics

The criteria for evaluation have been set up to test the accuracy, responsiveness, and usability of the Smart Food Saver system. Since the project is mainly about the system, the criteria for evaluation will make sure that the data is entered correctly, the inventory is counted correctly, the system responds to critical operations in a predictable manner, and the system is user-friendly [13], [14], [17].

1) Data Recording Accuracy:

This verifies the product and purchase information entered by the user is recorded correctly and retrieved without any loss or damage to the data.

$$Accuracy(\%) = \frac{\text{Correctly Stored Entries}}{\text{Total Entries Submitted}} \times 100$$

2) Inventory Aggregation Consistency:

The consistency check of the inventory is to see if the estimated amount in the inventory table is consistent with the total purchases made in the database. This is to ensure that the mapping of product and purchase information, as well as the stock update process of the system, is consistent even with repeated entries and irregular purchases.

$$Stock_{system} = \sum_{i=1}^{n} Quantity_{purchased(i)}$$

3) System Response Time:

Response time is the time taken by the system to execute its operations while adding products, entering purchases, displaying inventory, determining the risk of overbuying, and uploading receipts. The system should be able to respond to operations normally.

$$T_{response} = T_{end} - T_{start}$$

4) Usability Feedback:

In order to establish the usability, there was an informal usability test conducted. The users were able to use the interface first before being asked to give their feedback about their own experience of the ease of use and the level of usefulness of the interface. The aim was to ensure that the interface is user-friendly so that just about anybody can use it to manage their groceries.

IV. RESULTS

The reliability and stability of the Smart Food Saver were initially tested and proved to work efficiently for the entire grocery inventory that is managed. This shows that all the entries for the particular product and subsequent purchase were properly stored and retrieved without any loss of data or errors. The performance of the aggregation of all the quantities contained within the inventory resulted in accurate checks for the relevant stock values. This further indicates that the algorithm for consumption estimation works efficiently to satisfy the consumption demands of the household. The overall performance of the overbuy detection function was effective in establishing whether there was enough in store in relation to the recent purchases and times taken for depletion. This further indicates that this particular function works perfectly and efficiently to meet the demands of the household. The validation functions performed well and ensured that errors arising from inaccurate and incomplete data entries were not reflected, showing that all information from the receipts upload was properly incorporated into the entire process

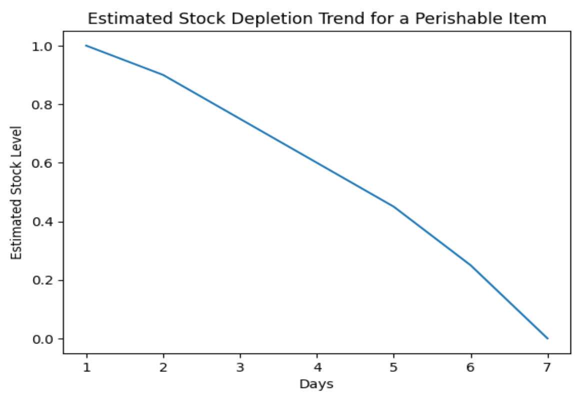

Figure 3. Estimated stock depletion trend for a perishable item based on recorded consumption.

As shown in Figure 3, it is observed that the system predicts product depletions based on the time elapsed against the recorded consumption rates. This serves to make users understand how the product has been used and for how long before depletion occurs.

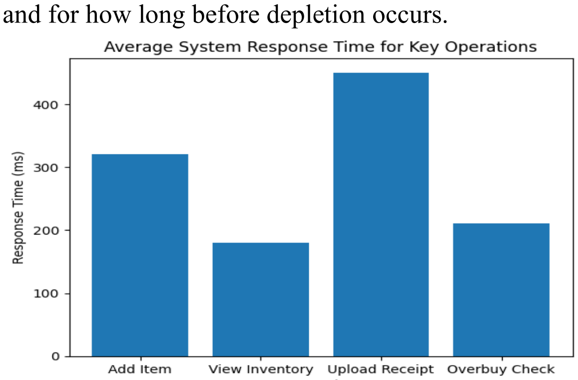

Figure 4. Average system response time for key user operations.

Figure 4 shows that all the important operations involving the addition of the item, viewing of the inventory, uploading, and overbuying are carried out in sub-second time. This confirms the smooth interface of the system, making it efficient for use in the household in real-time environments.

The system worked perfectly within all the test scenarios described. The period it took for the system to respond to critical scenarios such as adding items, making purchases, stock checking, and receipt uploading was always below a second. The system did not crash nor lead to any kind of corruption within it, even after thorough testing. The user feedback has confirmed that the system interface is user-friendly and easy to navigate, allowing users to easily understand their stock position and requirements for purchase. The test result clearly shows that the system is able to efficiently support home-based grocery management and can be used as a good starting point for further automation and intelligent assistance.

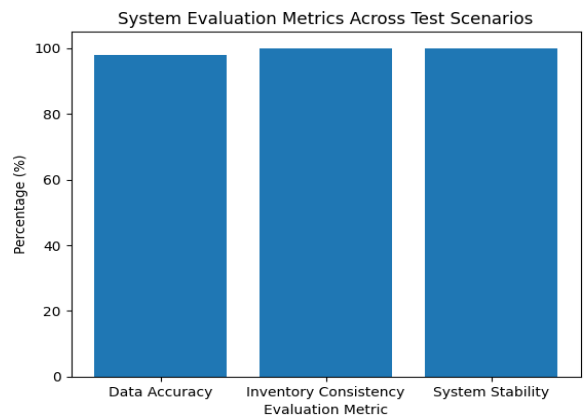

Figure 5. System evaluation metrics across test scenarios.

As shown in Figure 5, the Smart Food Saver system attains a high level of precision while recording the data, aggregating the inventory, and ensuring the stability of the system for carrying out the various test cases.

V. DISCUSSION

Analysis of the result shows that the Smart Food Saver system has the ability to handle the problems that arise from the processing of the groceries in the house. The scanning process and the validation of the information are sufficient to ensure that the rule applied is effective in dealing with the consumption pattern of the households, especially when the information is limited and the behaviour of the users is unpredictable.

The efficiency and fast response of the system show that the setting of the technology applied in the system is correct for light usage. The application of the Flask backend and the SQLite database ensures that the system is efficient without requiring any special hardware or infrastructure.

The addition of semi-automatic receipt processing is also a very good design element. Rather than simply using fully automatic receipt processing through OCR, which may not be completely accurate due to the different format of the receipt, the human-in-the-loop approach is a very good way of ensuring accuracy without having to manually process as much.

In addition, the logic system that is used in the overbuy detection is a very good way of showing how important it is to have inventory visibility and consumption prediction integrated. It gives the user feedback on consumption, such as the current amount and the time to depletion, which gives the user a reason to be more responsible with their purchasing and not waste money unnecessarily. Although the current system is based on fixed consumption rates, the modular system allows for the ability to improve these in the future.

In conclusion, it is clear from the above discussion that Smart Food Saver is indeed an effective solution that emphasizes the need for reliability, usability, and improvement. The result has clearly shown that it is well worth implementing a modular approach that can transform from a manual system to an automated and smart help system.

VI. CONCLUSION

This project work introduces Smart Food Saver, a system that is capable of managing groceries at home. The proposed system aims to minimize food wastage and unnecessary costs. The system combines effective data management, inventory management, and decision support in a friendly web interface. The experimental results confirm the entry of data, accuracy of inventory calculation, and integrity of system functionality.

The fact that it uses a proper system that combines manual processing as well as some form of automation, the system is able to ensure accuracy in the data as well as minimize the input required by the user. The addition of the semi-automatic receipt scanning is an advantage that makes the system ready for automation in the future.

Smart Food Saver is a good start for food operation in the home. It shows the ability of system-driven answers in meeting the challenges of sustainability. Future improvements could improve automation and intelligence to make it as a useful tool.

REFERENCES

[1] FAO, *Global Food Losses and Food Waste – Extent, Causes and Prevention*. Rome, Italy: Food and Agriculture Organization of the United Nations, 2011.

[2] A. Papargyropoulou, R. Lozano, J. K. Steinberger, N. Wright, and Z. bin Ujang, "The food waste hierarchy as a framework for the management of food surplus and food waste," *Journal of Cleaner Production*, vol. 76, pp. 106–115, 2014.

[3] R. Smith, "An overview of the Tesseract OCR engine," in *Proc. Int. Conf. Document Analysis and Recognition (ICDAR)*, 2007, pp. 629–633.

[4] F. Simistira, M. Seuret, N. Eichenberger, A. Garz, M. Liwicki, and R. Ingold, "DIVA-HisDB: A precisely annotated large dataset of challenging medieval manuscripts," in *Proc. ICDAR*, 2016.

[5] E. Turban, R. Sharda, and D. Delen, *Decision Support and Business Intelligence Systems*, 10th ed. Pearson, 2014.

[6] L. Secondi, L. Principato, and T. Laureti, "Household food waste behaviour in EU-27 countries: A multilevel analysis," *Food Policy*, vol. 56, pp. 25–40, 2015.

[7] S. Impedovo, "Optical character recognition — A survey," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1154–1178, 1999.

[8] M. Z. Babai, A. A. Syntetos, and R. Teunter, "On the empirical performance of (T, s, S) inventory policies," *Eur. J. Operational Research*, vol. 202, no. 2, pp. 466–472, 2010.

[9] S. Nahmias and T. L. Olsen, *Production and Operations Analysis*, 7th ed. Long Grove, IL, USA: Waveland Press, 2015.

[10] I. Sommerville, *Software Engineering*, 10th ed., Pearson, 2016.

[11] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, 3rd ed. Addison-Wesley, 2013.

[12] Google, "ML Kit Text Recognition," [Online]. Available: https://developers.google.com/mlkit/vision/text-recognition. Accessed: Dec. 4, 2025.

[13] J. Nielsen, Usability Engineering, Morgan Kaufmann, 1994.

[14] A. Abran, A. Khelifi, W. Suryn, and A. Seffah, "Software quality measurement concepts and definitions," IEEE Software, vol. 19, no. 1, pp. 36–44, 2002.

[15] D. J. Power, Decision Support Systems: Concepts and Resources for Managers, Greenwood Publishing Group, 2002.

[16] S. Amershi et al., "Guidelines for human-AI interaction," Proc. CHI Conf. Human Factors in Computing Systems, 2019.

[17] ISO/IEC 25010:2011, Systems and Software Engineering—Systems and Software Quality Models.

[18] A. Dix, J. Finlay, G. Abowd, and R. Beale, Human–Computer Interaction, 3rd ed., Pearson, 2004.

[19] R. Pressman and B. Maxim, Software Engineering: A Practitioner's Approach, 8th ed., McGraw-Hill, 2015.

[20] M. Beaudouin-Lafon, "Designing interaction, not interfaces," Proc. ACM Advanced Visual Interfaces (AVI), 2004.