Smart Fridge System: Intelligent Food Management and Health Integration

1st Axel Lundin
Acad. of Information Technology
College of Halmstad
Halmstad, Sweden
axelun22@student.hh.se

2nd Aleksander Janokovic
Acad. of Information Technology
College of Halmstad
Halmstad, Sweden
alejan22@student.hh.se

3rd Mohammed ISMAILI
Computer Science
Al Akhawayn University
Ifrane, Morocco
Mo.Ismaili@aui.ma

4th Hyunsuk Lee
Information Systems
Hanyang University
Seoul, South Korea
leehyunsuk2000@gmail.com

***Abstract*—The app integrates with a smart refrigerator connected via Ethernet and synced with a fitness app. Equipped with cameras and an internal scale, the fridge helps users automatically track the nutrition values of the food they consume throughout the day. By leveraging real-time data from both the food recognition system and weight measurements, the app provides an accurate calculation of daily nutritional intake.**

I. Introduction

A. Motivation

In an era where both health management and sustainability are increasingly critical, consumers are searching for solutions that can seamlessly integrate into their daily lives to help them achieve these goals. However, despite the proliferation of smart technologies in the home, existing smart fridges fall short in offering the tools necessary for comprehensive food management and nutritional tracking. These limitations make it difficult for users to manage their food consumption, track their caloric intake accurately, or reduce food waste effectively.

Our Smart Fridge System is designed to address these gaps by introducing a unique feature that current smart fridges lack: caloric intake tracking based on food weight. By integrating weight sensors and a nutritional database, our system enables users to track the exact amount of food consumed and automatically calculate the associated calories and macronutrients. This offers a precise and real-time method for managing daily food intake, which is especially valuable for users focused on personal health goals such as weight loss, maintenance, or muscle gain.

In addition to its nutritional tracking capabilities, our system is designed to simplify the process of meal planning. Through a user-friendly interface, the Smart Fridge can provide personalized meal recommendations based on the ingredients already available in the fridge and tailored to the user's dietary preferences (e.g., vegetarian, low-carb). This reduces the time and effort needed for meal preparation while promoting healthier food choices that align with the user's nutritional objectives.

One of the key challenges our system addresses is food waste, a growing environmental concern. Many households struggle to keep track of the food they have, leading to unnecessary waste as items expire unnoticed. Our Smart Fridge tackles this issue by sending alerts when food is nearing its expiration date and suggesting ways to use these ingredients before they spoil. This not only helps users make the most of the food they have but also contributes to the global effort to reduce food waste, promoting a more sustainable lifestyle.

The system is also designed to offer real-time insights into a user's nutritional intake, providing daily and weekly summaries of calorie consumption and macronutrient distribution. Users can set personalized nutritionalgoals and track their progress over time, receiving visual feedback through charts and progress indicators. These features make it easier for users to stay on top of their dietary goals without relying on manual tracking methods, which can often be tedious and imprecise.

Unlike more complex solutions that attempt to integrate multiple external devices, such as fitness trackers, our Smart Fridge keeps the focus on accurate calorie tracking through weight-based measurements and food inventory management. By prioritizing this core functionality, we ensure that the system is both practical and achievable within the project's timeframe, offering a real-world solution to the everyday challenges of food management and health tracking.

In summary, the Smart Fridge System provides a comprehensive and innovative approach to managing food consumption and nutritional health. By offering real-time calorie tracking, meal planning assistance, and food inventory management, the system enables users to make informed decisions about their diet while minimizing food waste. This project not only aligns with the growing demand for smarter, more sustainable household technologies but also addresses the need for more precise tools to help individuals achieve their personal health and fitness goals.

B. Problem Statement

1) Need for Comprehensive Food Management: Users require an efficient solution for tracking their food inventory

and nutritional intake. Current technologies do not provide the ability to automatically monitor food consumption or alert users about expiring items, resulting in significant food waste. A system that can intelligently track food items, manage expiration dates, and reduce waste is essential.

2) Desire for Integration with Health Management: Many individuals struggle to maintain a balanced diet and track their caloric intake due to the lack of integration between food management systems and health monitoring tools. Clients need a refrigerator that can synchronize with fitness apps and wearable devices, providing personalized meal recommendations based on their dietary goals and caloric burn.

3) Demand for Real-Time Nutritional Insights: Consumers often lack the tools to obtain real-time nutritional information about the food they consume. A system that can calculate calories and macronutrients based on actual food intake, using weight measurements and food recognition technology, is necessary for helping users meet their health goals.

4) Simplification of Meal Planning: Users find meal planning cumbersome and time-consuming, often leading to poor dietary choices. Clients require an intelligent appliance that can suggest meal ideas based on the ingredients available, dietary preferences, and fitness objectives, thus simplifying the meal preparation process.

5) Accessibility of Food Inventory Data: Consumers desire a user-friendly interface that allows them to access and manage their food inventory easily. An app that provides seamless updates on what is left in the refrigerator and calculates nutritional intake would enhance the user experience, making it easier for individuals to make informed food choices.

6) Reduction of Food Waste: As food waste is a significant global issue, users are increasingly seeking ways to minimize their environmental impact. A system that notifies them of expiring items, generates grocery lists, and encourages the consumption of available food before spoilage is critical in addressing this concern.

7) Support for Personal Health Goals: Clients are looking for a solution that not only tracks food but also aligns with their personal health and fitness goals, such as weight loss or muscle gain. Integrating food management with fitness tracking will enable users to achieve their nutritional objectives more effectively.

II. Requirements

A. User Authentication

1) Log-in: Users will be able to log in to the app using their registered credentials (email/username and password). The app will securely authenticate user details by checking them against stored credentials in the database.

2) Username: Each user will choose a unique username during the signup process. The username will be used to identify the user in the app, along with other personal details.
Sign-up
Users will create an account by providing details like email, username, and password. The app will validate the provided information (e.g., checking if the email is already in use) before saving the user's profile in the database.

3) Password recovery: If users forget their password, they can recover it via a password recovery process. The app will send a password reset link or OTP (One Time Password) to the registered email, allowing the user to set a new password.

4) User profile: Once logged in, users can manage their profile, which includes personal details like name, age, weight, height, and fitness goals (e.g., weight loss, muscle gain). This information helps the app calculate personalized nutrition recommendations. Users can also update their dietary preferences (e.g., vegetarian, keto) and other health details to further personalize meal recommendations.

B. Food Recognition & Measurement

1) Intergration with a scale to measure food weight: The app will determine the weight before and after to give an amount of used object. Combined with the food recognition system to calculate calorie content. When the user places food on the scale, the app will display the real-time weight and automatically update when the weight changes.

2) Camera/photo recognition for indenifying food items:
The app will track the weight of each item in the user's fridge or pantry. When the user takes food out of the fridge or puts it back, they will weigh the item before and after, allowing the app to track the difference in weight (how much of the item was consumed or added).
For example, if a user weighs a block of cheese before eating and then weighs it again afterward, the app will calculate the difference in weight and associate the corresponding calorie intake with the amount consumed.
This feature ensures more precise calorie tracking since it accounts for the exact amount of food consumed.

C. Calorie & Nutrition Tracking

1) Real-time calculation of calories based on food weight and type: The app will calculate calories in real-time based on the weight of food items detected by the refrigerator's internal scale and the type of food identified by the camera or manual input. The system will use a comprehensive food database to determine the calorie content of different types

of food and their respective portion sizes. As food is added or removed from the refrigerator, the app will automatically adjust the calorie count based on changes in weight.

2) Daily and weekly summaries of calorie intake and macro nutrients: The app will generate daily and weekly summaries that display the user's calorie intake, broken down into macronutrients: carbohydrates, proteins, and fats. These summaries will help users review their eating habits over time and make informed adjustments to their diet. The data will automatically update as new food is added or consumed.

3) Allow users to set calorie and macro nutrient goals (e.g., for weight loss, maintenance, or gain): Users can set personalized goals for daily calorie intake and macronutrient distribution (e.g., specific percentages for carbohydrates, proteins, and fats). The app will allow users to choose preset goals (e.g., for weight loss, maintenance, or muscle gain) or create custom targets based on their fitness objectives and dietary preferences. The system will compare the user's actual intake against their goals and provide feedback to help them stay on track.

4) Graphical representation of progress toward goals: The app will provide a graphical representation of the user's progress toward their daily and weekly calorie and macronutrient goals. Color-coded charts will differentiate between macronutrients (e.g., fats, carbs, proteins), showing users how close they are to meeting their targets. Users will also be able to track trends over time, helping them adjust their eating habits based on their progress. Alerts and notifications will be provided if the user is falling behind or exceeding their calorie or macronutrient goals.

D. Meal Recommendations

Personalized meal suggestions based on user goals, dietary preferences, and daily progress. With personalized meal suggestions it allows the user to easily make meals depending on the fridge inventory that suits the users needs.

Option to filter recommendations (e.g., vegetarian, low-carb, high-protein), by providing an option to allow users to filter recipes it will make meal suggestions even more directed to the user

E. User Goals & Progress

Setting weight goals and tracking progress over time, this allows users to set personalized weight goals and continuously monitor progress through food consumption data and weight changes
Monitoring nutritional goals (calories, protein, carbs, fats), Provides insights into daily intake of calories and macronutrients, helping users stay on track with nutritional objectives.
Integration with wearable devices for activity tracking and calorie burn estimates, Syncs with fitness wearables to incorporate activity data, offering real-time calorie burn insights for a comprehensive view of intake versus expenditure.

F. Usability

Intuitive design for easy food logging and tracking:
Make a user-friendly app for logging and tracking food that stores it in a database for the user to see its progress Clear guidance for setting and achieving goals
The user should be able to set its goals such as weight.

III. Development Environment

A. Choice of software development platform

For this project we have chosen a platform, a development environment that fits into our aim of quick processing, Machine learning integration and compatibility with IoT devices.

Platform Selection
Linux (Ubuntu 20.04 LTS) - Given that we well appreciated the strong support from the Linux environment (especially for open-source development), we decided to use Linux as the primary development platform also because the vast majority of IoT and smart device projects are based on IoT solutions for Linux. We chose Linux for the flexibility, with better memory control, and larger community support. On the other hand, macOS Monterey is a second option since it provides a Unix-based environment that is compatible with most of the development frameworks and tools.

Programming Language
Out of these languages and based on our previous experience and understanding of language usage on specific domains, we chose Python as our primary language since it has a lot of libraries available for machine learning, image recognition and IoT integration. Some of the core libraries that give our project life include OpenCV for image processing, TensorFlow for machine learning just to name a few of them which help efficiently store nutritional information and personalized recommendations. JavaScript (Node.js) — Also for prospective web or mobile front-ends. Real-time interactions from the client side will be handled using Node.js which also has a lot of usage for backend development in IoT applications, making it an appropriate choice for any other web interface requirements that may be required.

Development Environment Details

Operating System: Ubuntu 20.04 LTS.
Software Versions:
Python 3.8+
OpenCV 4.x for image processing
TensorFlow 2.x or PyTorch 1.x for machine learning
Node.js 14+ for backend and web services (if needed)
Hardware Resources:
8GB+ RAM to support smooth data processing
256GB+ SSD storage for faster data access and storage
High-speed internet to handle real-time data syncing and potential cloud integrations.

Cost Estimation

Our selected development setup is mainly budget-friendly:
Development Software: No cost, utilizing Linux and free tools like Python, OpenCV, and TensorFlow.

Cloud Services: If additional storage and processing power are needed, we may consider using Amazon Web Services (AWS). This can provide scalability and security for real-time data processing but will vary in cost depending on usage.

B. Software in Use

Our Smart Fridge System uses some common software libraries and frameworks for food tracking, nutritional analysis, and meal recommendations. Here are the key components we are using:

1. OpenCV (Open Source Computer Vision Library)
OpenCV helps with image processing and identifying food. It enables the system to take and analyze pictures of food in the fridge to recognize them correctly. OpenCV is well-known for computer vision tasks and works with Python, making it suitable for our image recognition needs.

2. TensorFlow or PyTorch (Machine Learning Framework)
We utilize TensorFlow or PyTorch to create machine learning models that classify and analyze food items. These models can identify different types of food and estimate their nutrition based on training data. TensorFlow and PyTorch are widely used for deep learning, featuring pre-trained models and GPU support, which speeds up processing.

3. NumPy and Pandas (Data Manipulation and Analysis)
NumPy and Pandas assist with data handling and processing, allowing us to store food data, calculate nutritional intake, and track user progress. These libraries are fundamental for data manipulation in Python, providing fast, efficient data handling.

4. Dietary Database (e.g., USDA Food Composition Database)
We use a dietary database to provide accurate nutritional information, such as calories and macronutrients, for various foods. This database allows us to cross-reference identified food items and calculate their nutritional values, essential for accurate calorie and nutrient tracking.

5. Flask or Django (Web Framework for Backend Services)
Flask or Django enables us to create a lightweight backend service to manage data processing, communication with the fridge's components, and potentially a web interface. Flask is chosen for its simplicity, while Django offers a more robust feature set if additional functionality is needed.

6. Docker (Containerization)
Docker is used to package the application and its dependencies into containers, simplifying deployment, testing, and running across different environments. This consistency is crucial for testing and deployment on various platforms, especially for IoT systems.

7. MySQL or PostgreSQL (Database Management System)
A database management system like MySQL or PostgreSQL stores user data, food inventory, and nutritional information, handling data storage and retrieval efficiently for easy access to food inventory and meal recommendations. PostgreSQL may be preferred for advanced data handling and analytics.

IV. Specifications

In this section, we outline the specific implementation details for the main features of the Smart Fridge System, including food recognition, nutritional tracking, meal recommendations, and user goal tracking. Each specification is designed to provide a comprehensive solution to user needs, as detailed below.

A. Food Recognition and Measurement

To accurately track food intake, our system employs a combination of image recognition and weight measurement:
1. Image Recognition:
Using OpenCV with a TensorFlow or PyTorch model, the system captures images of food items placed in the fridge. The image recognition model identifies the food item and categorizes it by type.
Implementation: A camera module within the fridge captures images, which are processed using OpenCV for initial image processing, such as resizing and normalization. The processed images are then input into a pre-trained machine learning model that classifies the food type with a specified confidence threshold.
Technical Details: The model is trained on a dataset of common food items, ensuring accuracy for frequently consumed foods. Post-processing adjusts the output to filter and confirm results.
2. Weight Measurement:
The fridge uses an internal scale to measure the weight of food items. Before and after each use, the weight is recorded to determine consumption.
Implementation: Weight data is processed through the application to calculate consumed portions. A food item's initial and final weights are compared, and the difference determines the amount consumed. Technical Details: Weight measurements are stored in the database with timestamps, and a smoothing algorithm accounts for minor fluctuations to ensure accuracy.

B. Nutritional Tracking

To provide accurate nutritional data and enable health tracking:
1. Caloric and Macronutrient Calculation:
Each identified food item is cross-referenced with a dietary database (e.g., USDA Food Composition Database) to retrieve nutritional values. Implementation: Once a food item is identified and its weight calculated, the system accesses the dietary database for calorie, protein, fat, and carbohydrate values. Nutritional values are then stored in a user's daily log. Technical Details: Calculations are adjusted for portion size

and weight, and data is updated in real time to reflect the user's cumulative daily intake.
2. User Goals and Progress Tracking:
Users can set goals for daily calorie intake and specific macronutrient distribution, such as high protein or low carbohydrate intake.
Implementation: The system tracks nutritional intake against these goals, providing alerts and progress reports. Data visualization tools display daily and weekly summaries, highlighting trends and goal adherence.
Technical Details: Data visualizations are created using libraries such as Matplotlib or Plotly, allowing users to easily track their progress toward nutritional targets.

C. Personalized Meal Recommendations

The system recommends meals based on available ingredients and user-defined dietary preferences:
1. Recipe Matching:
Using the identified inventory, the system suggests recipes aligned with user goals and available ingredients.
Implementation: An algorithm matches the food inventory against a preloaded recipe database, filtering results by user dietary preferences (e.g., vegetarian, low-carb).
Technical Details: Each recipe is tagged with dietary attributes, and an SQL query retrieves relevant options from the database based on user input.
2. Grocery List Generation:
The system generates a grocery list based on recipes chosen by the user and available inventory.
Implementation: When a recipe is selected, the system compares its ingredient list with the fridge's inventory and adds any missing items to the grocery list.
Technical Details: Inventory is updated automatically as items are used, and the grocery list is stored in the database for easy access.

D. Food Waste Reduction

To help users minimize waste, the system monitors food expiration dates and usage:
1. Expiration Alerts:
The system tracks the expiry dates of stored items, alerting the user when items are close to expiration.
Implementation: Users input expiration dates upon adding items, and the system sends reminders as the date approaches.
Technical Details: A daily background process checks for items within the expiration threshold, triggering notifications when needed. 2. Usage Suggestions for Expiring Items:
For items near expiration, the system suggests recipes to encourage usage before they spoil.
Implementation: When an item approaches expiration, the system searches for recipes that use the item and suggests them to the user. Technical Details: The algorithm prioritizes expiring items in the recipe search to minimize waste, drawing from the same recipe database.

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove the template text from your paper may result in your paper not being published.