UNIVERSITY OF OKLAHOMA
GRADUATE COLLEGE

IMPROVING NUTRITIONAL INTAKE TRACKING WITH AN INTELLIGENT SYSTEM: DESIGNING AN INTEGRATED NUTRITION FRAMEWORK AND PROTOTYPE

A THESIS
SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the
Degree of
MASTER OF SCIENCE

By FAVIO CESAR PEREZ GARCIA
Norman, Oklahoma
2025

IMPROVING NUTRITIONAL INTAKE TRACKING WITH AN INTELLIGENT SYSTEM: DESIGNING AN INTEGRATED NUTRITION FRAMEWORK AND PROTOTYPE

A THESIS APPROVED FOR THE
MICHAEL F. PRICE COLLEGE OF BUSINESS

BY THE COMMITTEE CONSISTING OF
Dr. Rajendra Singh (Chair)
Prof. Alexandra Durcikova
Prof. Heshan Sun

Table of Contents

ABSTRACT

Current nutrition technologies often use separate tools for food logging, recipe discovery, activity tracking, and dietary analytics. This separation makes it hard for users to see nutrition as a connected process, leading to confusion, inconsistent data, and low engagement over time. Inspired by enterprise resource planning (ERP) systems, this thesis suggests that nutrition should be managed as an integrated, lifecycle-based system instead of as isolated tasks. This study uses a Design Science Research approach and makes two main contributions. First, it introduces the Integrated Human Nutrition framework, which models nutrition management as three connected cycles: food production, food consumption, and activity. Each cycle follows clear Input, Process, and Output steps. Second, the study describes the design and creation of a working prototype. This web-based system brings together food acquisition, pantry management, recipe generation, meal logging, and progress tracking in one place. The prototype uses a React frontend, a FastAPI backend, and a PostgreSQL database, along with external nutrition and authentication services. The system is evaluated by assessing how well it matches the intended concepts, testing it in real-world scenarios, and comparing it with a commercial nutrition app. Results show that the prototype successfully implements the consumption cycle, reduces the fragmentation of nutrition tasks, improves users' awareness of their eating habits, and is consistent and easy to expand. While the prototype is not positioned as a commercial alternative, the evaluation indicates that an ERP-inspired, lifecycle-aware design provides clear structural advantages over siloed nutrition applications. This research lays the groundwork for future system improvements, real-world testing, and wider use of integrated information systems in nutrition informatics.

CHAPTER 1: INTRODUCTION

1.1 Background and motivation

Despite widespread access to nutritional information on food products and the availability of dietary apps, such as MyFitnessPal and Noom, young adults globally continue to experience inadequate dietary intake. This remains a primary immediate cause of malnutrition, as identified by the UNICEF framework, resulting in both nutrient deficiencies and obesity (Sablah, 2019). According to the World Health Organization, more than 2 billion people suffer from micronutrient deficiencies, while obesity tripled to over 650 million adults since 1975 (WHO, 2021). In the United States, adult obesity has reached 40.3% in 2023, with severe obesity affecting 9.4% of the population, increasing risks for type 2 diabetes, cardiovascular disease, and certain cancers (Emmerich et al., 2024). Concurrently, micronutrient deficiencies persist at high rates, with 95% of adults lacking vitamin D, 84% vitamin E, 46% vitamin C, 45% vitamin A, and 15% zinc, based on dietary intake data from the National Health and Nutrition Examination Survey (NHANES) spanning 2005–2016, a pattern that is not expected to improve amid increased intake of processed foods (Reider et al., 2020). This situation points to the limitations of current approaches to address malnutrition through fragmented solutions, which prevent accurate information flow across systems and therefore fail to improve population health.

These problems rarely stem from a single cause. Instead, scholars emphasize that malnutrition arises from system-level failures that shape food availability, quality, processing, selection, and consumption (Van Berkum et al., 2018; Fanzo et al., 2020). Governments have traditionally sought to address the underlying causes of malnutrition and poor dieting through poverty alleviation policies, food access, and educational programs. These include hospital-based nutrition programs, school feeding initiatives, and population-level dietary guidelines (Breda et al., 2019). While valuable, these interventions have little effect when disconnected from micro-level solutions that address the immediate causes of malnutrition, such as inadequate dietary intake. As a result, individuals are left to make dietary decisions with incomplete or inaccurate information, often relying on commercial diet apps that focus on market functionality and consumer preferences, and that capture only a portion of the eating process.

Most diet apps focus primarily on managing food consumption, which represents only one aspect of the broader food and nutrition system. Researchers have suggested that the food and nutrition system includes three subsystems, each with specific elements. The Producer Subsystem includes production, processing, and distribution elements; the Consumer Subsystem includes acquisition, preparation, and consumption elements; and the Nutritional Subsystem includes digestion, transport, and metabolism elements (Sobal et al.,1998). Some food intake and nutrition apps address inventory and preparation, while others track calories burned and integrate this data into calorie-tracking systems (Zečević et al., 2021). The proliferation of apps offering isolated services has led to user confusion and fails to address inadequate nutritional intake, as these tools do not connect all nine elements of the comprehensive food and nutrition system.

Although many food and nutrition software applications connect smart wearables to diet apps and provide activity-level reports, significant gaps remain in linking users to comprehensive food data and labeling. Few tools support users, governments, or producers in enhancing accuracy, transparency, or nutrient integrity in food production. Commercial apps that primarily depend on APIs and third-party datasets, whether public or paid, generally lack an integrated approach.

This disconnection results in reactive scenarios in which people monitor weight changes after excessive calorie intake, increase activity levels to compensate for weight imbalances, or switch food products on the shopping list. Given the complexity of food systems, an integrated, multi-layered approach is needed that connects:

1. *Food Production Layer* (nutrient variability, processing effects, accuracy of labels, guaranteeing transparency between producers, regulators, and consumers)
2. *Personal Consumption Layer* (planning meals, procuring groceries, managing home inventory, preparing food, and monitoring consumption with accurate data)
3. *Activity & Monitoring Layer* (linking energy expenditure, physical activity, and health programs)

According to systems thinking literature, challenges involving interconnected cycles and feedback loops necessitate integrated frameworks rather than isolated solutions (Meadows, 2008). The central question is how to organize these layers into a unified nutritional intake system that supports producers, informs policymakers, and guides consumers. Drawing inspiration from the comprehensive integration of enterprise resource planning (ERP) systems, this thesis proposes a conceptual model that unifies food system layers into a holistic nutrition ecosystem.

1.2 Problem Statement

Despite the widespread use of diet applications, users continue to face challenges in achieving healthy eating because these apps function in isolation and lack integration with the broader food system. For instance, inventory apps such as AnyList or Pantry Check assist in tracking home groceries but do not account for changes in food from farm to store, including nutrient loss during processing (Rickman et al., 2007). Calorie trackers like MyFitnessPal allow meal logging but rely on generalized USDA data, which can underestimate energy by 20–50% and overestimate sodium by up to 34 times, particularly for non-Western foods (Zečević et al., 2021; University of Sydney, 2024). Furthermore, only 11% of users maintain long-term engagement with these apps, as manual data entry often leads to user fatigue

(König et al., 2021).Recipe apps like Yummly or Mealtime provide meal ideas based on individual tastes, but they do not check what individuals have in their pantries or automatically update their calorie totals. Grocery budgeting apps such as Out of Milk or MealBoard suggest where to buy and track spending, yet none link to production details (such as how farming affects vitamins) or to your daily activity from wearables. This leaves big gaps across the nine food-nutrition stages: production/processing/distribution, acquisition/preparation/consumption, and digestion/transport/metabolism (Sobal et al., 1998). Users jump between 3-4 apps daily, creating confusion and wrong data that stops real diet improvement (Carter et al., 2013).

The real problem is that no current framework for designing these apps integrates all these pieces the way an ERP system integrates them in business organizations. Supermarket purchase data could be integrated into auto-pantries and smart meal plans (Jenneson, 2022), but the related apps stay focused on a single element (or a couple of elements) of the food and nutrition system. Without connecting production transparency, personal consumption cycles, and activity monitoring, rates of malnutrition and obesity persist despite significant app market revenues (WHO, 2021; Market.us, 2025). This study addresses this gap by proposing an integrated subsystem framework and a consumption-cycle prototype.

1.3 Research Motivation

Although diet applications claim to facilitate healthier eating, they often frustrate users due to siloed functionality and low retention rates, with 70% of users discontinuing within 2 weeks, primarily due to manual logging fatigue (Market.us, 2025). Inventory apps such as AnyList track pantry but skip production nutrient changes; calorie trackers like MyFitnessPal rely on USDA averages that can be inaccurate by 20–50%; recipe apps such as Yummly ignore your actual groceries; budgeting tools like Out of Milk recommend stores but do not provide information on the nutritional value of local produce. This fragmentation compels users to switch between multiple applications, thereby impeding proactive dietary planning from purchase to meal consumption. The primary motivation is building a framework that classifies food intake into cycles: production/processing/distribution, acquisition/preparation/consumption, digestion/transport/metabolism (Sobal et al., 1998); and links them like an ERP system. Current tools stay reactive (log after eating), ignoring upstream gaps like farming variability or cooking losses that fuel inadequate intake, the key malnutrition driver (Rickman et al., 2007; WHO, 2021). No app connects these for accuracy, especially with local foods regulated by country-specific standards.

The proposed framework bridges existing silos by guiding users through the consumption cycle, including automated pantry management, meal planning, and intake tracking, while preparing for future integration with production applications (such as geolocation-based nutrition) and wearable devices. The widespread adoption of wearable technology presents a timely opportunity to scale holistic nutrition ecosystems.

1.4 Research Objectives

This thesis designs an integrated framework for nutrition apps that connects food production, consumption, and activity cycles to better monitor dietary intake and visualize our eating habits. Drawing from the nine-stage food system (Sobal et al., 1998), it prototypes the consumption cycle while enabling future links to production transparency and wearables. The specific objectives are:

1. Design a conceptual framework classifying food-nutrition processes into three subsystems (production, consumption, activity) with defined data flows, analogous to ERP systems and the nine-stage food-nutrition system (Sobal et al., 1998).
2. Analyze weaknesses in current siloed apps (inventory, calorie tracking, recipes, budgeting) and food data infrastructures to justify holistic integration.

3. Develop a conceptual ERP model unifying data flows, stakeholders (users/producers/governments), and subsystems for proactive dietary management.
4. Develop a prototype app for the consumption cycle module: purchase data, auto-pantry, meal planning and suggestions, intake tracking, with design principles for usability and retention.
5. Demonstrate the functioning of the app through a Scenario-based approach, and discuss future extension for app developers, producers, and policymakers to scale the framework.

1.5 Research Questions

This thesis is guided by a set of research questions designed to explore how a unified, system-level model can be conceptualized. These questions reflect the multi-layered challenges identified in the background, problem statement, and motivation, and they structure the development of the proposed design artifact.

1. How can food systems cycles be structured into an integrated ERP framework for nutrition management?
2. What are the critical weaknesses in current nutrition tools?
3. How does an integrated consumption-cycle prototype support end-to-end control and visibility of dietary intake compared to siloed nutrition tools?
4. What design principles and architectural decisions are required to make the Integrated Human Nutrition framework feasible and internally consistent?
5. What are the implementation implications and scalability considerations for future integration of production-cycle and activity-cycle data into the broader Human Nutrition system?

1.6 Summary of Scope

This thesis develops a conceptual "Integrated Human Nutrition" framework classifying food processes into three subsystems: production (nutrient variability/traceability), consumption (acquisition/preparation/tracking), and activity (wearable energy balance); based on the nine-stage food-nutrition model (Sobal et al., 1998). The prototype implements selected consumption cycle components: grocery purchase import → auto-pantry database → meal planning and suggestions → intake tracking. Evaluation focuses on usability, conceptual analysis, and functionality through a simulated scenario using the OpenFoodFacts API. The design can be adaptable to local contexts (as the API records global data) for individual dietary management, with future scalability for producers/policymakers (Hevner et al., 2004).

1.7 Summary of Limitations

Several constraints shape this study's scope, ensuring feasible artifact development per Design Science Research guidelines:

- Exclusion of production/labelling systems: Conceptual model includes nutrient variability, but prototype uses free OpenFoodFacts API databases, regulatory supply-chain data beyond thesis resources.
- Exclusion of activity/metabolic monitoring: Framework recognizes energy wearables, but prototype excludes implementation (requires separate field beyond consumption focus).
- Partial consumption cycle implementation: Prototype demonstrates core components only, not full app/system.
- Data accuracy constraints: Relies on OpenFoodFacts API databases with known slow processing and inaccuracies, especially for non-Western foods.
- Generalizability limits: Scenario-based demonstration lacks cultural/socioeconomic diversity; no long-term field trials.

- Technical scope: No AI/geolocation, full automation, or multi-platform deployment due to time constraints.

1.8 Thesis Structure

This thesis has five chapters that build from malnutrition problems to a working nutrition framework prototype. It follows Design Science Research steps: identify gaps, design a solution, build a prototype, test it, and show impact (Hevner et al., 2004). This Chapter 1: Introduction sets up the background, problem, motivation, objectives, scope, and limits. The prototype focuses solely on the consumption cycle, from purchase to tracking, whereas the full framework also covers production and activity.

- **Chapter 2: Literature Review** – Discusses current literature on nutrition apps, food production changes, labelling issues, wearables, systems thinking, and ERP models. Finds gaps that need a unified framework (Sobal et al., 1998).
- **Chapter 3: Research Methodology** – Explains Design Science Research process—problem ID, objectives, design/build, demo, evaluation. Covers how we made the framework and prototype with rigor.
- **Chapter 4: System Design and Prototype** – Discusses "Integrated Human Nutrition" framework with data flows across production/consumption/activity. Details the prototype design
- **Chapter 5: Evaluation and Results** – This section presents the results by evaluating the prototype against three methods: scenario-based demonstration, conceptual analysis, and usability.
- **Chapter 6: Discussion** – Tests framework against research questions. Covers what it means for users, app makers, producers, and governments.
- **Chapter 7: Conclusion** – Lists limitations of the study and future work.

CHAPTER 2: LITERATURE REVIEW

2.1 Malnutrition and Inadequate Food Intake

Malnutrition, a critical global health issue, encompasses both nutrient deficiencies and excesses. According to the World Health Organization, more than 2 billion individuals experience micronutrient deficiencies, while the prevalence of obesity has tripled since 1975 to affect 650 million adults (WHO, 2021). In the United States, adult obesity reached 40.3% in 2023, with 9.4% classified as severely obese, thereby increasing the risk of diabetes, cardiovascular disease, and cancer (Emmerich et al., 2024). Concurrently, 95% of adults are deficient in vitamin D, 84% in vitamin E, and many others, according to NHANES data from 2005-2016, a trend that worsens with processed foods (Reider et al., 2020).

The UNICEF framework pins inadequate dietary intake as the main immediate cause, fed by food insecurity (underlying) and poverty/system failures (basic) (UNICEF, 2020). Young adults suffer most despite nutrition apps everywhere; poor diets cause both deficiencies and weight gain. Governments seek to address this through hospitals, school lunches, and dietary guidelines, supported by dietitians and clinics (Breda et al., 2019). These work for high-risk groups but skip everyday people making grocery and meal choices.

Applications such as MyFitnessPal increase user awareness but don't fix population habits, having 70% drop off fast due to bad databases, burdensome manual entry, and a lack of integration with food production data (Lieffers & Hanning, 2012; Chen et al., 2015). Ultra-processed foods, misleading labels, and a lack of transparency make choices (Franco et al., 2019). Scholars advocate for comprehensive data systems that encompass the entire food cycle, from production to physiological response, rather than relying on isolated tracking tools. This identified gap underscores the need for an "Integrated Human Nutrition" framework, as proposed in this thesis, to connect all stages and facilitate substantive change.

2.2 Limitations of Current Nutrition Tracking Tools

Digital nutrition apps like MyFitnessPal, Cronometer, and LoseIt let people log food, track calories, and watch macros. Studies show they help change habits when users stick with them (Chen et al., 2015). But research points out big problems that stop real results: data errors, user drop-off, and working alone instead of together.

- *Accuracy issues hurt most.* Apps depend on user input that's often wrong, repeated, or missing (Carter et al., 2013). Lieffers and Hanning (2012) tested popular apps and found nutrition numbers off from real lab data. Many grab labels as "truth," but labels miss by 20% due to farm differences, processing, storage loss, and cooking (Urban et al., 2011; Rickman et al., 2007). My earlier test showed USDA API lags 7 seconds per query with 20-50% errors on non-Western foods (Zečević et al., 2021; University of Sydney, 2024).
- *User retention is also a significant challenge.* Market data indicate that 70% of users discontinue use within two weeks, primarily due to the tedious nature of manual data entry (Market.us, 2025), and only 11% maintain long-term engagement (Carter et al., 2013).
- *Problem of fragmentation*. Apps split by function, and no full integration. No app covers full consumption cycle or links to production and activity. Here's how they break down:

Table 2.1: Nutrition App Silos

App Type	Examples	Strengths	Key Gaps
Calorie Tracking	MyFitnessPal, LoseIt, Cronometer	Good overall awareness; 90% wearable sync	USDA/OpenFoodFacts errors (20-50%); 7s API lag
Inventory/ Pantry	AnyList, Pantry Check, Out of Milk	Grocery lists, expiration alerts	No calorie/nutrient totals; ignores processing loss
Recipe Suggestions	Yummly, Mealime, Paprika	Taste-based ideas, 1M recipes	No pantry scan; generic nutrition (no local foods); manual add to tracker

Grocery/ Budget	Out of Milk, Basket, Flipp	Store deals, spending track	No nutrition scores; no meal planning link; label errors
AI Photo Log	SnapCalorie, FoodVision	Photo to calorie guess	25% error in international dishes; no ingredient breakdown

2.3 Food Production, Label Accuracy, and Traceability

No app or public tool perfectly tracks the nutritional content of food from farm to table for consumers. Governments, NGOs, and private companies built enterprise systems, but they are often locked behind paywalls or private networks with no free consumer access. For example, the IBM Food Trust uses blockchain for supply chains (Walmart, Nestlé), but starts at $100/month for small businesses to integrate these services, more than $10K per month, and no public API (Stanley, 2018; Dignan, 2018). FoodDocs and Craftybase do traceability for factories ($20-500/month); however, they only focus on B2B, rather than your everyday grocery app (Aung & Chang, 2014).

Public APIs fall short. USDA FoodData Central, with free features, limits 1,000 requests per hour per IP, and comes with a 7-second lag. In terms of accuracy, 20-50% of inaccuracies for non-US foods, and no local produce data (USDA API Guide, n.d.; Zečević et al., 2021). OpenFoodFacts can require up to 10min per search, frequent fails, and there is a need for premium in order to gain more reliability (Franco et al., 2019). Edamam free tier includes 500 hits trial before starting paying $29-499/month; and Nutritionix, at $299 per month in the starter package (Measurement Toolkit, n.d.). No API covers localized country, state, country nutrient variability from soil, ripeness, or regulations.

Real-world data gaps undermine the reliability of nutrition applications. Processing methods alone can reduce levels of vitamin C and B by 30–70%, with additional losses occurring during cooking and storage (Rickman et al., 2007). Food labels typically report average nutrient values across batches, yet they often fail to capture product-specific variability; for example, restaurant items have been shown to deviate by approximately 20%

from stated calorie values (Urban et al., 2011). Moreover, region-specific foods, such as Peruvian versus U.S. mango varieties, are frequently absent from standard datasets. Apps that rely on barcode scanning inherit these inaccuracies because many commercial nutrition databases are partly crowdsourced and lack rigorous verification mechanisms to identify discrepancies between labeled and actual nutrient content (Franco et al., 2019). Natural variability due to agricultural practices, processing techniques, and storage conditions further complicates nutrient estimation (Measurement Toolkit, n.d.; Urban et al., 2011). Although experts highlight the need for comprehensive and regularly updated national nutrient databases, their maintenance poses substantial operational and financial challenges (Franco et al., 2019).

Table 2.2: Food Data APIs Reality Check

API/System	**Access**	**Cost**	**Limits**	**Accuracy Issues**
USDA Food Data Central	Public	Free	IP; 7s lag	20-50% non-Western
OpenFoodFacts	Public	Free/ Premium	slow; fails often	User-generated errors
Edamam	Public	$29-499/month	500-10K hits/day	Generic averages
IBM Food Trust	Enterprise	$100-10K/month	B2B only	No consumer API
Govt/NGO Systems	Private	N/A	Nonpublic	Local foods missing

2.4 Activity Tracking and Health Technology

Activity tracking exploded with wearables, such as Fitbit, Apple Watch, Garmin lead with 80% US adoption (Piwek et al., 2016). Unlike production/consumption silos, this field matured fast, where smartphone health apps have evolved to integrate multiple data streams, including wearable device synchronization (Dennison et al., 2013; Piwek et al., 2016). Studies show synced apps boost awareness: users adjusting calories burned vs. intake, improving short-term weight loss, and their robust APIs enable real-time data synchronization with nutrition applications (Piwek et al., 2016).

Despite these advances, significant limitations persist. Energy-expenditure accuracy remains imperfect, with wrist-worn sensors showing approximately 15% error and frequently over- or underestimating caloric burn during mixed-activity routines (Shcherbina et al., 2017). More critically, these devices do not address the quality or accuracy of dietary intake data. Wearables can report that a user has "burned 500 calories," yet they cannot detect whether the corresponding intake values—often derived from nutrient databases with documented 20–50% inaccuracies—are themselves misestimated. They also do not adjust for nutrient variability; consuming calories from ultra-processed foods versus locally sourced produce is treated equivalently in most tracking systems (Zečević et al., 2021).

This produces a structural disconnect—activity-tracking systems have become highly refined, yet they remain isolated from upstream issues that shape dietary accuracy. Many applications compute simple energy-balance models (e.g., "eat 2000, burn 500 = deficit") without assessing whether the intake component is accurate or reflects the true nutrient composition. No mechanisms exist to incorporate production-level traceability or consumption-planning data into these calculations. As a result, such systems offer only partial benefits—they may support fitness-oriented users, but they do not address the underlying driver of malnutrition, namely inadequate intake quality (Chen et al., 2015).

Table 2.3: Activity Integration Reality

Feature	**Status**	**Strengths**	**Key Problems**
Wearable Sync	Mature (90% apps)	Instant data flow (Fitia, 2025)	No food accuracy check
Energy Accuracy	15% error	Good for trends (Shcherbina 2017)	No nutrient adjustment
Full Cycle Link	Missing	Steps→calories works	Ignores production/consumption gaps

User Adoption	High (80%)	Motivates tracking (Piwek 2016)	Drops when diet data fails

Activity tracking solves half the equation (output) but ignores inputs (food quality/quantity errors). An integrated framework needs all three cycles connected; wearables alone can't fix malnutrition.

2.5 Systems Thinking and Input–Process–Output Systems

This section examines the theoretical foundations underlying the "Integrated Human Nutrition" framework. Existing applications often fail to connect production, consumption, and activity, whereas systems thinking, enterprise resource planning (ERP) models, and design principles offer potential solutions. Systems thinking provides a conceptual basis for analyzing interdependent processes within complex environments. Scholars contend that many health and nutrition challenges persist because they are addressed through isolated interventions rather than integrated, system-level models (Meadows, 2008). Systems theory highlights the importance of cycles, feedback loops, and multi-level interactions, concepts essential to understanding human dietary behavior.

The Input–Process–Output (IPO) model, widely used in information systems research, offers a simple but powerful way to conceptualize how resources flow and transform within a system. Traditionally applied to organizational settings, IPO can also model food intake (input), digestion and metabolism (process), and energy or health outcomes (output) (Hackman & Morris, 1975).

Recent research advocating for food systems approaches highlights the need to integrate food production, dietary behavior, physiological responses, and environmental influences (Van Berkum et al., 2018; Fanzo et al., 2020). Viewing the human nutrition ecosystem through the lens of IPO and systems cycles reinforces the argument that fragmented apps cannot solve inadequate food intake. Instead, a comprehensive architecture is needed, analogous to ERP systems used in enterprises.

2.6 Enterprise Resource Planning (ERP) Inspired Design Principles

Enterprise Resource Planning (ERP) systems integrate business processes such as procurement, production, inventory, distribution, sales, finance, and analytics within a unified platform (Monk & Wagner, 2012). These systems connect previously separate processes, maintain shared master data, and provide real-time visibility across complex organizations. Prior to the adoption of ERPs, businesses relied on isolated systems characterized by limited traceability and inconsistent data, a situation analogous to the current fragmentation observed in nutrition applications, food data systems, and activity trackers (Davenport, 1998).

The analogy between ERP systems and human nutrition management is supported by strong theoretical foundations. Both domains involve cyclical, interdependent processes requiring coordination of inputs, transformations, and outputs. Systems-thinking literature endorses applying organizational frameworks to biological or behavioral contexts when underlying processes exhibit lifecycle and feedback characteristics (Meadows, 2008). Similarly, food-system scholars advocate for integrated models that link food production, procurement, preparation, consumption, and health outcomes (Van Berkum et al., 2018; Fanzo et al., 2020). However, despite these conceptual parallels, consumer nutrition technologies remain fragmented. Nutrition logging, recipe selection, food production traceability, and physical-activity data are managed in separate applications, which prevents users from developing a comprehensive understanding of their dietary behaviors.

This thesis applies ERP principles to develop the Integrated Human Nutrition framework, which structures the food lifecycle from production to consumption and incorporates activity tracking. Research in nutrition informatics demonstrates that proactive planning, utilizing inventory or purchase data, enhances adherence to dietary goals and reduces user effort compared to retrospective logging. For instance, supermarket transaction data can more accurately predict eating habits than self-reports and can reduce manual logging by up to 80% (Jenneson, 2022). Additionally, companies such as Hungryroot demonstrate that automated, inventory-based meal-planning algorithms can select up to 70% of groceries (Monteros, 2023). The Integrated Human Nutrition framework is grounded in enterprise-systems logic and takes a practical, testable form through its implementation in the prototype. The Human Nutrition Consumption System incorporates several ERP features:

1. *Shared data.* All modules, including the scanner, pantry, recipe generation, meal logging, and analytics, utilize a unified PostgreSQL database. Pantry items, logs, and goals function as primary records. This approach aligns with ERP master data management principles and eliminates inconsistencies throughout the process.
2. *Structured, multi-stage lifecycle flows.* The prototype employs defined Input-Process-Output steps to manage the consumption cycle. Each stage generates outputs that serve as inputs for subsequent stages, mirroring the management of business cycles such as procurement-to-pay or order-to-cash in ERP systems.
3. *Traceable information.* When a meal is logged, it is linked to its corresponding recipe, which is further connected to pantry items added through scanning or manual entry. This structure establishes a traceable chain, enabling both users and analytical tools to observe how specific ingredients contribute to dietary outcomes.
4. *Closed-Loop feedback.* Feedback loops are fundamental to systems thinking. In the prototype, consumption data are processed by analytics, which identify shortages, excesses, or inconsistencies. These insights inform subsequent recipe and pantry decisions. Consequently, user actions function as both inputs and outputs, establishing an operational closed loop.

Collectively, these features demonstrate that the Integrated Human Nutrition framework extends beyond a conceptual adaptation of enterprise systems. It constitutes a functional, testable system. The framework operationalizes ERP principles such as shared master data, lifecycle integration, traceability, integration speed, and feedback loops. These elements can be evaluated for consistency and compliance through scenario-based testing, thereby fulfilling Design Science Research criteria for translating theory into practical, testable solutions.

2.7 Health Tech Design Principles

- Simplicity: Minimize inputs (auto-pantry > manual)
- Personalization: Local foods and context (Dennison et al., 2013)
- Feedback loops: Accuracy flags (König et al., 2021)
- Habit formation: Proactive nudges beat reactive logs

Table 2.4: Theories and Framework Mapping

Theory	Key Idea	Nutrition App Gap	Framework Fix
Systems Thinking	Feedback loops	Siloed stages	ERP data flows
IPO Model	Input→Process→Output	Covers 1-2 stages	9-stage integration
ERP	Unified stakeholder data	Production blind	Producer to consumer links
Proactive Nutrition	Predict from purchases	Reactive logging	Auto-pantry planning
DSR	Build and test artifacts	No frameworks	Prototype and evaluation

2.8 Summary of Gaps the Proposed Artefact will Address

The literature identifies six critical gaps that existing nutrition technologies do not address, thereby justifying the development of the "Integrated Human Nutrition" framework. These gaps encompass the entire food-nutrition lifecycle, ranging from production-related blind spots to fragmented activity data, which collectively hinder effective malnutrition prevention.

Gap 1: No Integration Across Full Food Lifecycle

Digital tools fixate consumption logging, ignoring upstream production variability that alters nutrients by 30-80% through processing, storage, and cooking (Rickman et al., 2007). Labels and apps treat food as static, with no variation in value after processing methods, yet no consumer system captures this end-to-end (Urban, 2011; Aung & Chang, 2014).

Gap 2: Fragmentation Between Nutrition and Activity Ecosystems

Diet apps and wearables operate separately despite interdependence—energy balance requires accurate intake and expenditure, but syncs ignore food quality errors (20-50%) (Piwek et al., 2016; Shcherbina et al., 2017). No mainstream platform unifies logging, nutrient absorption estimates, calorie burn, and recovery metrics.

Gap 3: Static Databases Ignore Real-World Dynamics

USDA FoodData Central and OpenFoodFacts offer generalized averages that skip brands, batches, cooking (20-80% nutrient shifts), and storage; with no adaptation to user context (USDA, n.d.; Fukagawa et al., 2022).

Gap 4: Zero Consumer-Accessible Food Traceability

Enterprise food traceability systems focus on supply chain management for safety and recall purposes but remain isolated from consumer-facing applications (Aung & Chang, 2014). OpenFoodFacts tries but fails on reliability and full integration with personal diets.

Gap 5: Missing Unified Conceptual Model

Food systems research calls for holistic approaches bridging production, processing, distribution, and consumption (Van Berkum et al., 2018; Fanzo et al., 2020), yet traditional interventions remain fragmented and sectoral with narrowly defined focus (Van Berkum et al., 2018). Despite calls for integrated frameworks that address the full food system (Sobal et al., 1998), no practical unified model exists that connects production data with individual dietary behavior and metabolic responses in a manner analogous to enterprise resource planning systems.

Gap 6: Behavioral Design Lacks Systemic Context

Apps deliver calories without food quality flags, retention aids, or physiological feedback, where health tech needs accuracy and reliability for adherence, but ignores full-cycle dynamics (Topol, 2019; König et al., 2021).

Table 2.5: Gap Synthesis & Framework Solution

Gap	Core Problem	Lit Evidence	Framework Fix
1	Upstream blind	30-80% losses (Rickman 2007)	Production subsystem APIs
2	Diet/activity silos	15-50% errors (Shcherbina 2017)	ERP data flows
3	Static DBs	USDA Log zeros (USDA 2025)	Dynamic adjustment flags
4	No traceability	Enterprise-only (Aung 2014)	Future producer links
5	No model	Fragmented research with narrow focus (Van Berkum et al., 2018)	9-stage subsystems
6	Weak behavior	70% dropout (König 2021)	Proactive planning

These gaps demand a novel artifact. Our prototype demonstrates consumption-cycle integration as proof-of-concept, scalable to a full ERP uniting producers, consumers, and policymakers against malnutrition's immediate cause: inadequate intake.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Introduction to Design Science Research (DSR)

This research uses Design Science Research methodology for the systematic development and evaluation of this prototype, and it addresses seven fundamental guidelines to ensure the rigor and relevance of Design Science Research contributions (Hevner et al., 2004). These guidelines frame the present study as follows:

- The primary artifact constitutes a conceptual model: the Integrated Human Nutrition framework alongside a prototype of the consumption cycle subsystem.
- The research demonstrates direct relevance to the problem of inadequate dietary intake as the immediate cause of malnutrition, according to Chapter 2.
- The design is submitted for evaluation through conceptual analysis, scenario-based demonstration, and usability assessment.
- The development process adheres to concepts such as: systems thinking and ERP architectures.
- The artifact involved a structured and iterative process.
- The communication of results is comprehensive through this dissertation.
- The design shows scalability for future validation and industrial deployment.

The methodology confronts the fragmentation evident across contemporary nutrition technologies and food data infrastructures through the construction of a conceptual framework and functional prototype.

3.2 Research Process Overview

This thesis follows the six-stage Design Science Research process proposed by Peffers et al. (2007), adapted to the context of nutrition informatics.

In the first stage, problem identification, motivation, and gaps were identified through the literature review and system analysis: fragmentation between nutrition applications and activity tracking tools, inaccuracies and difficult access to nutrient databases, and misalignment between food production information and consumer systems.

In the second stage, five objectives were formulated to guide the design. These objectives included: mapping processes across the food production, consumption, and activity cycles; analyzing limitations in current digital tools and data; developing a unified conceptual model equivalent to an ERP system; designing compliant to Design Science Research guidelines; and demonstrating the model's performance through a working prototype of the food consumption cycle.

The third stage, involved the construction of the Integrated Human Nutrition framework through layered architecture, Input-Process-Output models, and decomposition of three cycles and nine stages from Sobal et al.'s food-system model. During this stage, cycle diagrams, architecture diagrams, and subsystem tables were produced to formalize the design.

In the fourth stage, the food consumption subsystem mentioned earlier turned into a functional web application, the Human Nutrition Consumption System. The prototype implements the conceptual design through modules for user authentication, pantry management, barcode scanning, recipe generation and logging, and progress analytics; and uses a three-tier architecture with a React frontend, FastAPI backend, and PostgreSQL data layer.

The fifth stage, is carried out in Chapter 4 through a scenario-based walkthroughs that includes screenshots of the main user interfaces and descriptions.

Finally, the sixth stage, assesses the prototype in terms of functionality, consistency with the conceptual model, and usability. Chapter 5 performs a combination of functional checks, conceptual alignment, and simulation walkthrough appropriate for this stage (Hevner et al., 2004; Gregor & Hevner, 2013).

3.3 Problem Identification and Objectives

The initial stage of the Design Science Research process, as delineated by Peffers et al. (2007), asks for identification of the problem and articulation of its importance. The problem of systemic fragmentation in current nutrition technologies is addressed through the synthesis of evidence and theoretical analysis presented in Chapters 1 and 2. Specifically, the literature review in Chapter 2 points out six deficiencies that undermine the efficiency of existing solutions in addressing inadequate dietary intake, identified by the UNICEF framework as the primary immediate cause of malnutrition (UNICEF, 2020). These deficiencies show the fragmentation between food production, nutrition logging and physical activity monitoring systems; dependence on nutrient databases unresponsive to variability in agricultural and processing practices, and preparation techniques; and lack of a cohesive model unifying all nutritional processes in the food nutritional system (Zečević et al., 2021; Rickman et al., 2007; Aung & Chang, 2014). Each deficiency receives empirical corroboration through quantitative indicators.

The second stage involves the definition of objectives that this prototype must satisfy to demonstrate both relevance and utility (Peffers et al., 2007). These objectives were formulated in Chapter 1 and validated through the literature review to establish an accurate evaluative criterion and ensure alignment between deficiencies and proposed solutions. The five specific objectives are:

- *Identification and Structuring of Key Processes*: To model the processes across food production, consumption, and activity cycles, and the nine-stage food-nutrition system by Sobal et al. (1998) into three interoperable subsystems.
- *Analysis of Deficiencies in Existing Technologies*: To examine weaknesses within current nutrition tools, food labeling and APIs, and monitoring technologies, establishing the theoretical justification for this architectural integration.

- *Development of Integrated Conceptual Model*: To build a comprehensive ERP-like design: the Integrated Human Nutrition framework, that represents the full nutrition ecosystem, promoting a proactive rather than reactive approach to dietary management.
- *Formulation of Design Principles and Considerations*: To formulate the system design principles according to Design Science Research guidelines (Hevner et al., 2004), encompassing usability, scalability, data transparency, and adaptability to future implementations.
- *Demonstration of Practical Utility*: To illustrate through prototype and scenario analysis how the proposed model enhances dietary decision-making and supports scalability for implementation by application developers, food producers, and public health policymakers.

3.4 Design & Development: Conceptual Framework

The conceptual framework developed in this study, the Integrated Human Nutrition framework, translates the complexity of nutrition management into three cycles: the Food Production Cycle, the Food Consumption Cycle, and the Activity-Level Cycle. Together, these cycles form a unified system architecture analogous to an enterprise resource planning system in business environments, where distinct modules serve different stakeholders while sharing coherent data (Davenport, 1998).

3.4.1 Subsystem Structure and Rationale

Building upon the nine-stage food and nutrition system proposed by Sobal et al. (1998), the framework reshapes the traditional producer, consumer, and nutritional subsystems as three interoperable cycles. The Food Production Cycle constitutes production, processing, and distribution activities, led mainly by producers, factories, and regulators. The Food Consumption Cycle focuses on acquisition, preparation, and consumption, led by individual users and households. The Activity-Level Cycle switches from digestion, transport, and metabolism to activity monitoring, physiological tracking, and behavioral changes as a component for balanced nutrition, led by individuals and, in future extensions, inviting nutrition professionals, gym coaches, and health programs.

Table 3.1 Food Nutrition System Table

Subsystem	Stages (Elements)	**Key Transformations/Processes**
Producer Subsystem	Production	Transforms resources into crops and commodities (e.g., farming, hunting, gathering).
	Processing	Transfers raw agricultural goods into food stuffs and foods (e.g., cleaning, milling, cooking, packaging).
	Distribution	Transfer of outputs from production and processing to places where food acquisition occurs (e.g., wholesale, retail, food service).
Consumer Subsystem	Acquisition	Procurement of food stuffs and foods from various outlets.
	Preparation	Transformation of raw food stuffs into consumable foods (e.g., cooking).
	Consumption	Eating, involving selection, serving, and ingesting food items.
Nutrition Subsystem	Digestion	Breakdown of foods into nutrients in the gastrointestinal tract, allowing absorption.
	Transport	Movement of absorbed food components to specific sites in the body for use or storage.
	Metabolism	Use of particular nutrients in physiological processes; this stage involves homeostatic mechanisms.

Each cycle is modelled using an Input–Process–Output structure to make it implementable in digital systems. Inputs include, for example, agricultural and batch data in the production cycle; grocery products, and user preferences in the consumption cycle; and intake records and wearable sensor data in the activity-level cycle. Processes include cultivation and processing, pantry updates and recipe generation, and estimation of energy use and feedback.

Outputs consist of labelled food products and nutrient profiles, updated pantry and meal records, and activity summaries and behavioral recommendations.

Table 3.2 Food-Production-Cycle IPO Table

Cycle	Stage	Input	Process	Output
Food Production	**Agricultural**	Soil Water Seeds	Growing Measuring quality Pesticides control	Raw products
	Processing	Raw foods Chemicals Processing instructions FDA/USDA rules Nutritional claims	Cutting, grinding, packaging Nutritional testing Shelf-life calculation Contaminant screening	Finished products Nutrition label Barcode or batch ID Verified nutritional values Passed quality checks
	Logistics	Packaged foods Temperature rules Expiration dates	Transport Tracking	Foods to stores

Table 3.3 Food-Consumption-Cycle IPO Table

Cycle	Stage	Input	Process	Output
Food Consumption	**Food Planning**	Goals and Budget Diet rules	Planning Selecting recipes Making a shopping list	Purchase plan Shopping List
	Food Acquisition	Choosing stores and budgets Buying groceries	Evaluating what is healthy, affordable, available Tracking pantry/fridge inventory Rotating food to prevent waste	Ingredients or raw materials Budget used
	Food Inventory Management	Raw foods	Storing Categorizing	Updated inventory Reduced waste

			Tracking expiration Noticing shortages	Efficient use of resources
	Food Preparation	Ingredients Recipes Tools Time & skills	Cooking (Washing, cutting, seasoning) Portioning meals	Meal portions (e.g., meal prep) Macro distribution (carbs/fat/protein)
	Food Intake	Meals Snacks Drinks	Calorie intake Digestion Nutrient absorption	Energy Changes in health metrics (weight, mood, glucose levels)
	Feedback, and Monitoring	Weight Nutritional intake data Energy levels Sleep Fitness performance Gut symptoms	Analyzing trends Identifying deficiencies Adjusting diet or exercise Re-planning groceries	Improved habits Better decisions Optimized personal health Loop back to Stage 1

Table 3.4 Activity-Level-Cycle IPO Table

Cycle	Stage	Input	Process	Output
Activity-Level	**Planning**	Goals Time Equipment Health conditions	Selecting and Scheduling	Workout planned
	Execution	Workout plan Equipment Body state	Performing	Muscle growth Weight changes
	Recovery	Sleep Stress Hydration	Muscle repair	Recovery Readiness for activity

3.5 Prototype Development: Human Nutrition Consumption System

The primary stage of this conceptual framework is a working prototype that implements the Food Consumption Cycle within the Integrated Human Nutrition framework. The prototype:

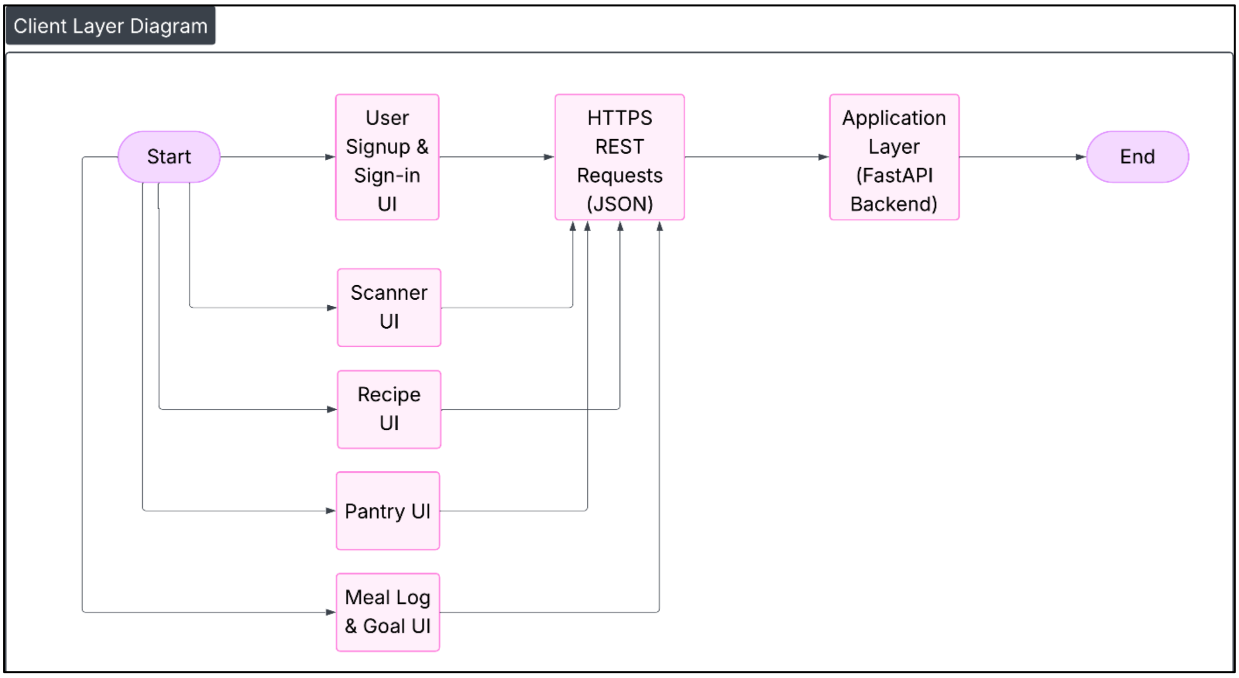
Human Nutrition Consumption System, portrays the positive impact of digitizing the acquisition, preparation, consumption, and monitoring stages through software modules. Its design follows the layered architecture and data flow outlined in the previous diagram.

3.5.1 System Architecture and Technology Stack

The system adopts a three-layer architecture consistent with web applications in health and nutrition informatics (Design Guidelines for Technology-Enabled Nutrition Education Tools).

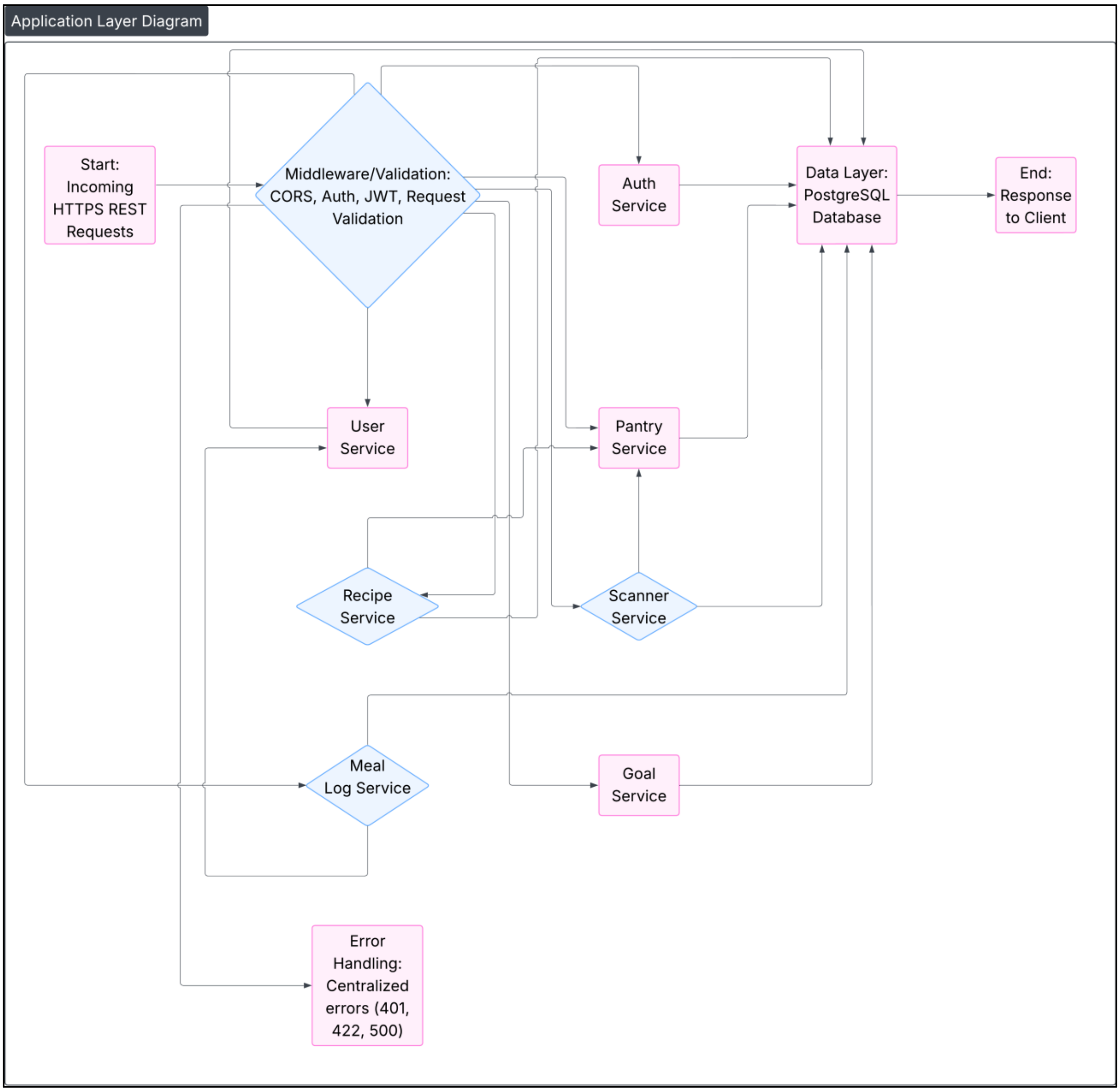
The client layer consists of a React web application that provides user interfaces for the main functional areas: Dashboard, Scanner, Pantry, Recipes, Progress and Insights, and Profile. These interfaces correspond directly to the stages of the consumption cycle by allowing users to scan or add items, manage inventory, generate meal suggestions and log them, and review trends and streaks.

Figure 3.1 Human Nutrition System: Client Architecture



The *application layer* is implemented using a FastAPI backend that exposes REST endpoints and encapsulates the core business logic. It is organized into five principal services: an authentication service handling registration, login, Google OAuth, and password recovery; a scanner service managing barcode and image-based item capture and interaction with third-party nutrition APIs; a pantry service implementing create, read, update, and delete operations for inventory, including categorization and expiration tracking; a recipe service responsible for generating meal options from pantry items and user preferences and logging selected recipes as consumed; and a dashboard service that aggregates food logs into daily, weekly, and monthly metrics and computes streaks and badges.

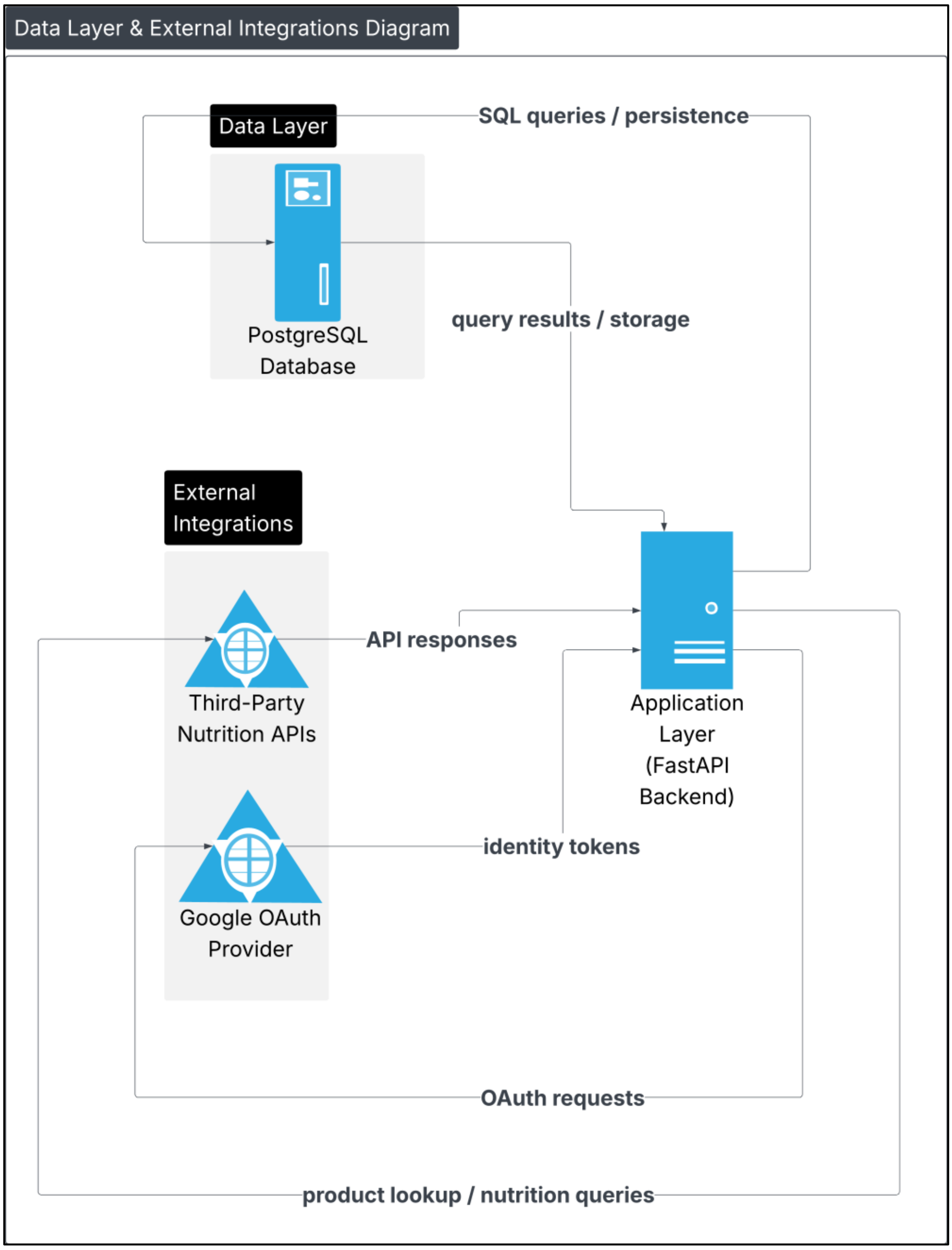
Figure 3.2 Human Nutrition System: Application Architecture



The *data layer* consists of a PostgreSQL relational database that stores user accounts, pantry items, recipe records, food logs, metrics, and user preferences. This layer reflects the data entities implied by the conceptual framework and supports persistence of the consumption-cycle stages.

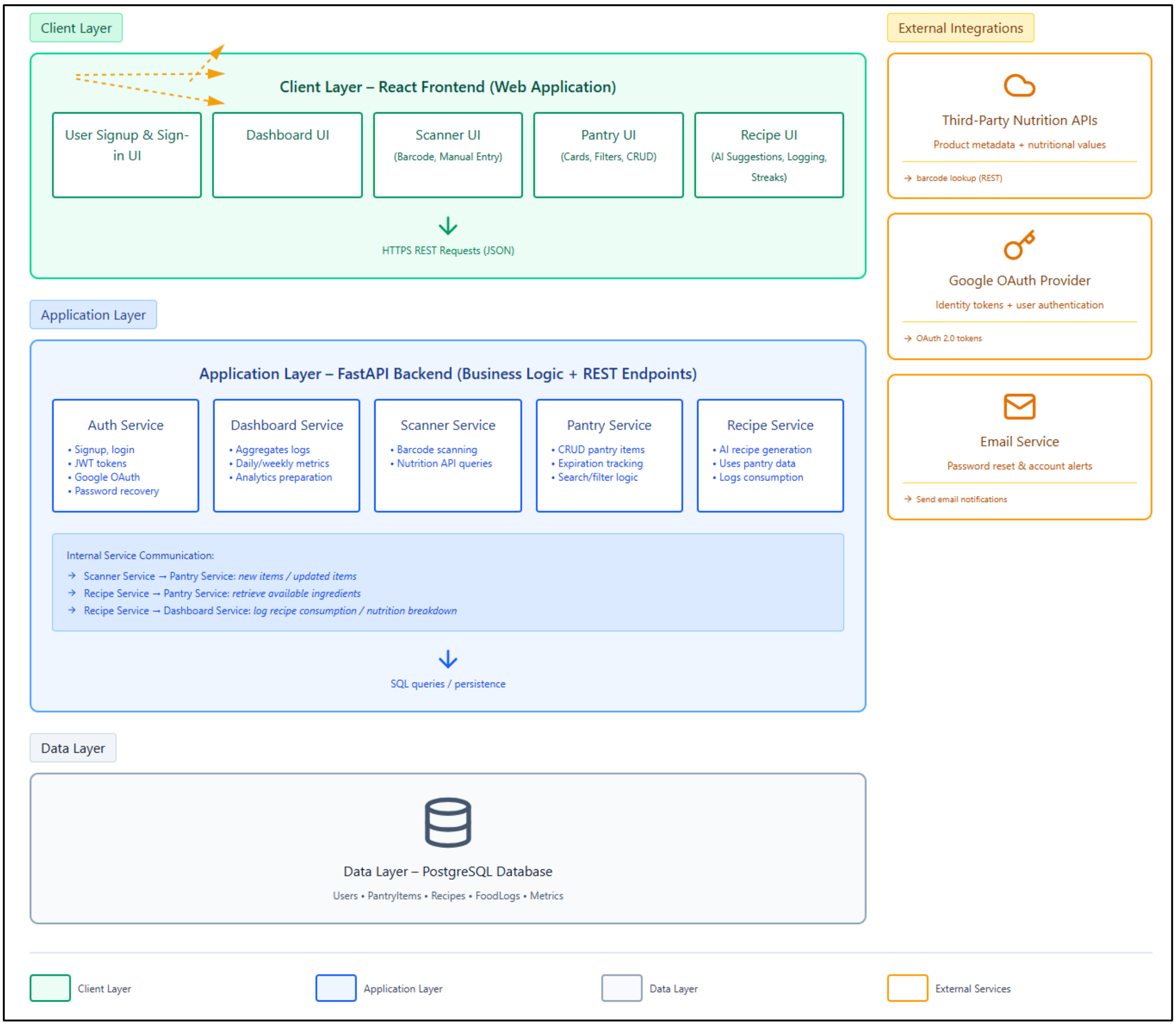
External integrations include nutrition information APIs for product metadata and nutrient values, Google OAuth for secure identity management, and an email service for account-related notifications.

Figure 3.3 Human Nutrition System: Data and External Integrations Architecture



The diagram below explicitly shows the flow of information across layers: user input enters through the client, goes through authenticated REST endpoints, activates backend services, and produces outputs based on database operations. Bidirectional arrows illustrate both data retrieval and state-changing operations, making the logic transparent and traceable.

Figure 3.4 Human Nutrition System Full Architecture



Within this architecture, the Food Consumption Cycle serves as the implemented subsystem for this thesis and maps its stages directly onto the modules represented in the prototype. This mapping shows how the conceptual flow from acquisition to monitoring is not only represented theoretically but incorporated into a functioning software system.

The Food Production and Activity-Level cycles remain conceptual in this phase of the research. Nevertheless, the prototype leaves room for integrating production data (nutrient data, processing nutritional change) and downstream activity data from wearables. Overall, the architecture diagram acts as both a communication tool and a validation mechanism. It shows that the conceptual Integrated Human Nutrition framework has been turned into a concrete and testable prototype.

3.5.2 Functional Modules

The prototype has five functional modules, each corresponding to one or more stages in the conceptual model:

1. Authentication and Profile Module: This module enables users to create accounts, authenticate via email and password or Google sign-in, recover forgotten passwords, and manage profile details and nutrition preferences. It is a foundation for all subsequent interactions and personalization through dietary preferences and goals.
2. Pantry Module: The pantry module implements inventory management. Users can view their current pantry interface as a card, add new items manually or through scanning, and edit and sort quantities and expiration. This module corresponds to the acquisition and inventory elements of the consumption cycle by maintaining an up-to-date pantry of available foods.
3. Scanner Module: The scanner module provides a camera and image interface for barcode capture. It forwards barcode data to the backend scanner service, which calls external nutrition APIs to retrieve product descriptions and nutrient information. Successfully resolved items are normalized and stored in the pantry. This module operates the input details of the acquisition stage.
4. Recipe Manager Module: The recipe manager allows users to generate meal suggestions based on pantry contents and other constraints such as dietary restrictions,

calories, and macronutrients. Generated recipes display ingredients, preparation, and nutritional breakdowns. Users can then log recipes as consumed in their dashboard. This module involves the planning and preparation stages of the consumption cycle.

5. Dashboard and Progress Module: The progress views aggregate logged intake into daily macronutrient summaries, weekly and monthly trend charts, macro distributions, and behavioral indicators such as streak length, and badges. This module represents the monitoring and feedback stage, translating logs into actionable insights for users, consistent with behavior-change recommendations in digital health technologies (Dennison et al., 2013; König et al., 2021).

3.5.3 Internal Data Flow

The internal data flow structure of the prototype follows the logical sequence where user acquires food items, either by scanning barcodes or by entering them manually, the scanner or pantry interface sends requests to the backend, which validates and enriches the data using external nutrition APIs and stores them in the pantry tables. The recipe service queries pantry data and user preferences to generate suggested meals, which are then presented in the recipe manager interface. When a user logs a meal as consumed, the system writes an entry to the tables, triggering the dashboard to recompute daily totals, weekly trends, macro breakdowns, and streak stats.

3.5.4 Design Science Deliverables

According to the Design Science Research expectations, the prototype yields several tangible outputs that substantiate the artifact’s viability (Hevner et al., 2004). These deliverables include a functioning web application covering the complete Food Consumption Cycle, a documented software architecture and data flow model, a relational database schema, integrated services for authentication, scanning, recipe management, and analytics, and scenario walkthroughs illustrated with user interface screenshots. Collectively, these outputs demonstrate that the conceptual framework can be translated into a concrete system that addresses the gaps in current nutrition tools and provides a basis for future extension into the production and activity-level cycles.

3.6 Evaluation Methods

The evaluation of the Human Nutrition Consumption System follows the principles for Design Science Research evaluation articulated by Hevner et al. (2004) and Peffers et al. (2007), which recommend assessing both the technical features and its alignment with the problem and objectives identified earlier. Given that it is an early-stage prototype focused on a single subsystem, the evaluation strategy combines a scenario-based demonstration, conceptual alignment, and benchmarking comparison analysis, rather than multiple empirical trials, to ensure feasibility and coherence in initial stage artifacts in the health and nutrition field (Gregor & Hevner, 2013; Nutrition informatics guidelines).

First, a *scenario-based demonstration* was used to evaluate the system from a process perspective. User journeys mirrored realistic situations, including stocking a pantry after a grocery trip, planning and preparing meals using available ingredients, logging consumption, and reviewing weekly and monthly nutrition trends. These scenarios were executed within the prototype, and the resulting screens, and data flows were documented in Chapter 5.

Second, a *conceptual alignment* analysis was performed to assess the correspondence between the prototype and the research objectives. Each objective was mapped to specific elements of the conceptual framework and prototype like the three-cycle model, software architecture, and functional modules. The analysis discussed whether the prototype addressed the fragmentation in current nutrition tools, whether the processes reflected the IPO structures outlined for the consumption cycle, and whether the design supported future integration with production and activity-level data sources. This alignment check provides evidence that the artifact is not only functional but also theoretically grounded and responsive to the gaps identified in the literature (Zečević et al., 2021; Van Berkum et al., 2018).

The third evaluation method is a benchmark comparison between the Human Nutrition Consumption System and an existing nutrition-tracking application, specifically MyFitnessPal, which serves as a widely used reference point in consumer nutrition technology. This comparison does not aim to measure superiority or performance outcomes but instead provides a structured way to assess how the prototype differs from a typical siloed application with respect to integration, friction, and visibility. The comparison evaluates a set of categories defined by the designer based on direct interaction, real-time testing of workflows, and examination of how each system supports common nutrition-related tasks. The intention is to assess how design choices manifest in practical functionality, rather than to make empirical claims about user outcomes. The benchmark focuses on the following categories: Friction and task complexity, Dependency on manual entry, Data consistency and structure, Integration gaps across lifecycle stages, Feedback immediacy and update latency, Visibility of behavior patterns. This scope is appropriate for a Design Science prototype that primarily aims to demonstrate feasibility and to lay the groundwork for subsequent empirical studies with broader and more diverse participant groups (Hevner et al., 2004; Gregor & Hevner, 2013)

3.7 Methodological Rigor, Limitations & Ethics

Methodological rigor is linked to the Design Science Research guidelines and in the systematic linkage between the identified problem, the conceptual framework, the implemented prototype, and the evaluation activities. Hevner et al. (2004) emphasize that rigorous Design Science Research requires clear problem relevance, reliance on a sound knowledge base, well-articulated design processes, and explicit evaluation of the resulting artifact. These criteria are addressed through the structured application of the Peffers et al.

(2007) process model, the extensive literature review in Chapter 2, and the use of recognized architectural patterns and technologies for web-based health applications.

First, the conceptual framework explicitly extends a food-systems model (Sobal et al., 1998) into three cycles, anchored in established theory. Second, all major design decisions, including the separation into production, consumption, and activity-level cycles and the use of Input–Process–Output representations, are documented and justified with reference to systems thinking, ERP analogies, and nutrition informatics best practices (Davenport, 1998; Meadows, 2008). Third, the prototype is implemented using well-documented technologies (React, FastAPI, PostgreSQL) and follows a three-tier architecture that allows replicability and facilitates future extension.

Despite these strengths, several limitations must be acknowledged. The most significant limitation concerns: the prototype initiates only the Food Consumption Cycle, while the Food Production and Activity-Level cycles remain conceptual. As a result, the current system does not yet integrate real-time production traceability data or wearable-based activity metrics.

A second limitation relates to evaluation depth. Due to time and resource constraints, the evaluation relies on functional checks, scenario-based demonstrations, and a user walkthrough rather than on empirical studies or controlled trials, therefore, conclusions remain provisional and require confirmation through future work, which is consistent with Design Science artefacts in health care domains (Gregor & Hevner, 2013).

A third limitation concerns data accuracy and external dependencies. The system currently depends on third-party OpenFoodFacts APIs, which sometimes exhibit variability and under- or overestimation of nutrient values, particularly for processed foods and non-Western dishes (Rickman et al., 2007; Zečević et al., 2021). Premium production-level data services were beyond the scope and budget of this thesis. As a result, while the framework is designed to accommodate higher-quality data sources, the current prototype cannot eliminate underlying inaccuracies in food composition information.

Finally, there are generalizability constraints. The system was designed and tested in a context reflecting the dietary patterns, food availability, and digital infrastructure typical of a high-income setting. Although the conceptual model intentionally allows contextual adaptation of the production cycle to different countries and regulatory environments, further work is required to validate the framework in low- and middle-income contexts where food systems, labeling practices, and technology access differ substantially (Fanzo et al., 2020; Franco et al., 2019). These limitations, while important, do not undermine the core contribution of the research, which is to provide a theoretically grounded and technically instantiated framework that demonstrates how the Food Consumption Cycle can be integrated and digitized, laying the foundation for extensions and more comprehensive evaluations.

CHAPTER 4: ARTIFACT DESIGN AND DEVELOPMENT

4.1 System Overview and Requirements

This chapter presents the design and implementation of the prototype developed to represent the consumption-cycle subsystem of the Integrated Human Nutrition framework introduced in Chapters 2 and 3. While the broader framework comprises three interconnected cycles: food production, food consumption, and activity-level monitoring, the implemented artifact focuses on the food consumption cycle, as this is the domain most directly actionable for individuals and the most feasible target for a first-stage Design Science Research prototype.

The prototype expands into five of the three stages of the consumption subsystem model: food acquisition, food storage, meal preparation, meal consumption, and nutritional monitoring. These stages are realized through a set of integrated modules for authentication, pantry management, product scanning using both barcode and image input, recipe generation, meal logging, and progress insights. Together, these modules provide a structured process that supports users in planning, managing, preparing, consuming, and reflecting on their food choices in a proactive, data-driven way, addressing the current fragmentation and reactive approach observed in current diet applications.

At a high level, the system enables users to create an account and define personal nutrition goals; to stock a virtual pantry with foods acquired via scanning or importing an image of their barcodes, or manual entry; to generate recipes based on available pantry items and individual preferences; build meal plans and record them; and to review daily and weekly progress through charts, summaries, and behavioral indicators such as streaks and consistency metrics.

These features form a self-contained consumption-cycle workflow that mirrors enterprise resource planning logic, in which inventory flows from entry to consumption, data are aggregated into meaningful insights, and decisions are informed by structured and timely information (Davenport, 1998).

4.2 Functional Requirements

The following functional requirements are derived from the objectives and informed by the limitations of existing tools mentioned in Chapter 2.

- *R1 – User authentication and personalization:* The system allows users to create an account, authenticate securely, and configure individual nutrition goals and preferences. This requirement is realized through email-and-password registration and login, Google OAuth support, and profile management interfaces.
- *R2 – Pantry inventory management:* Users must be able to add, edit, and remove items' details and expiration date in their pantry and view associated nutritional information. This requirement supports the inventory representation needed for recipe generation and consumption tracking.
- *R3 – Multi-input product acquisition:* The system supports multiple input methods for food items, including barcode scanning, importing images, and manual entry of items from pantry modules. This diversity of input mechanisms aims to reduce logging burden and records of purchase data, responding to adherence issues documented in the literature.
- *R4 – Recipe generation and meal-preparation support*: The system must provide recipe suggestions derived from current pantry contents and user preferences, improving the planning and preparation stages of the consumption cycle.
- *R5 – Meal logging and nutritional tracking:* Users would have access to meal plan suggestions for ease of logging, whether based on generated recipes or custom entries, and the system would record accurate intake summaries.

- *R6 – Progress and insights visualization:* The system must compute and display daily, weekly, and monthly progress, macronutrient totals, calories, streaks, and other indicators, in line with recommendations.

4.3 Non-Functional Requirements

In addition to functional capabilities, several non-functional requirements were defined to ensure that the prototype is reliable, usable, and extensible.

- *N1 – Usability:* Interfaces should be intuitive and easy to navigate. Screen layouts such as the pantry cards, dashboard summaries, and recipe-manager filters were designed with this goal in mind, drawing on usability principles highlighted in mobile health research.
- *N2 – Performance and responsiveness:* Barcodes as well as recipe generation, should return results promptly to keep engagement and prevent the drop-off commonly associated with other nutrition apps.
- *N3 – Data integrity and consistency:* Inventory records, food logs, and computed metrics must adhere to a schema and be synchronized across sessions and modules.
- *N4 – Security and privacy:* Sensitive information, including login credentials and dietary data, must be protected through hashed passwords, OAuth-based authentication, and secure communication with external APIs.
- *N5 – Extensibility:* The architecture should build a solid foundation for future improvements and potential integrations through interfaces with food production systems and activity systems without redesigning the core consumption modules.
- *N6 – Interoperability:* The system must interoperate with third-party services for nutritional lookup, barcode decoding, and email authentication.

4.4 Prototype Fit within the Integrated Human Nutrition Framework

Within the broader three-cycle Integrated Human Nutrition architecture, the prototype works as the operational core of the consumption cycle. It demonstrates how data can flow from acquisition, through storage and preparation, to consumption and monitoring for individual or household users. This shows how user behavior can be shaped through structured processes and feedback, how an adapted ERP structure can be applied at the micro-enterprise level of the household and individual, and how it takes a proactive approach to nutritional decision-making.

Although the Food Production Cycle and the Activity-Level Cycle remain conceptual at this stage, the prototype establishes the foundational logic through which these additional cycles can be integrated in future work. This prototype serves us as a practical tool not only for exploring improvements to dietary management but also as a first-stage artifact proposed by the Integrated Human Nutrition framework.

4.5 System Architecture

This system follows a modular architecture integrating a React-based client, a FastAPI backend, a PostgreSQL database, and several external services to support barcode decoding, nutritional lookup, authentication, and email delivery. The architecture is intentionally lightweight to support rapid prototyping, while maintaining a structure that can scale and expand in future iterations.

4.5.1 High-Level Architecture

At a high level, the system has three logical layers:

1. *Presentation layer (client).* A Next.js (React) web interface manages all user interactions.
2. *Application layer (backend API).* A FastAPI application implements the business logic. It comprises an API-routes layer, a request-validation layer using Pydantic

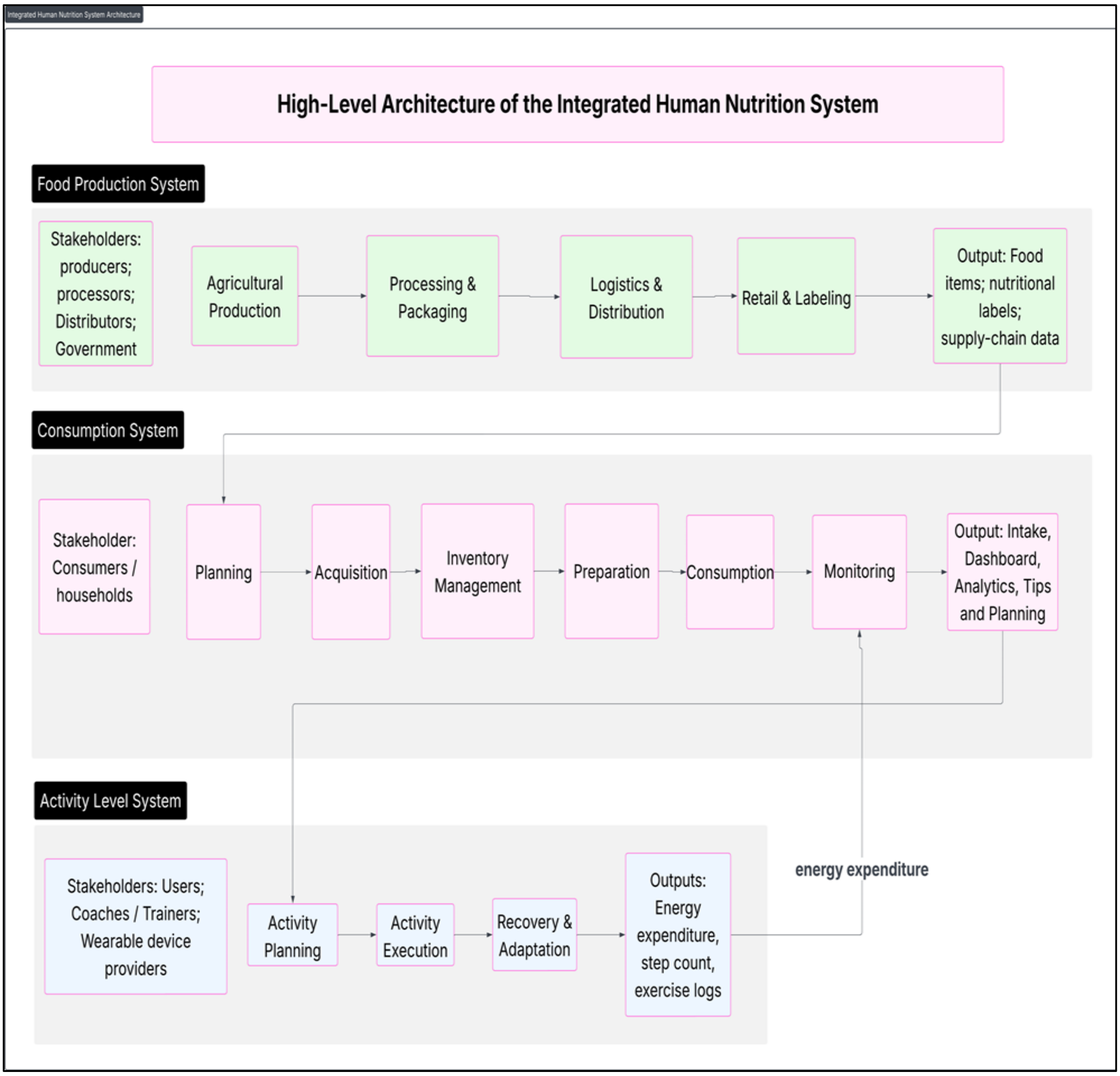
schemas, and a service layer that encapsulates domain operations: user management, barcode and image processing, recipe filtering, nutrient calculations, and analytics for progress and streaks.

3. *Data layer (PostgreSQL database)*. A relational schema stores users, goals, pantry items, recipes, meal logs, and derived metrics, represented through SQLAlchemy models.

Data typically flows in a loop through user actions, client user interface, backend services, database persistence and aggregation, analytics, and ending in visualization.

This loop reflects the cyclic nature of the consumption subsystem, in which each interaction updates both operational state (e.g., inventory and logs) and analytical summaries (e.g., daily macros and streaks).

Figure 4.1. System software architecture (client, backend, database, external services)



4.5.2 *Integrated Context and Data-Flow View*

To situate the prototype within its environment and clarify external dependencies, the architecture can also be described as an integrated context and data-flow model. The primary external actors and services are:

1. *User.* Interacts with all modules via the client, including the scanner, pantry manager, recipe manager, meal-logging views, profile settings, and progress dashboards.
2. *Authentication provider*. Google OAuth services validate identities and issue tokens that the backend uses to authenticate requests.

3. *Food-data APIs*. External services such as Open Food Facts provide nutrient values and product metadata based on barcode information, subject to the limitations discussed in Chapter 2.
4. *Email provider.* An SMTP or email service supports account verification, password-reset messages, and alert notifications.

Within the system boundary, we have the following core application modules: Authentication and Profile, Pantry Management, Scanner, Recipe Manager, Meal Logging, and Progress and Insights. They aim to communicate exclusively through the FastAPI backend and share a common data model in PostgreSQL. When a barcode is scanned, the client sends the decoded barcode to the Scanner endpoint; the backend calls the external food-data API, normalizes the response, and writes a pantry item record. The Pantry, Recipe Manager, and Meal Logging modules then operate on this shared inventory and log data, while the Progress and Insights module reads aggregated information to generate visual feedback.

4.5.3 Alignment with the Three-Cycle Architecture

The system architecture reflects the three-cycle Integrated Human Nutrition model described in Chapter 3. Conceptually, the implemented prototype occupies the food consumption cycle, while providing clear integration points for the production and activity-level cycles:

Table 4.5 ERP and Concept Breakdown

Cycle	Role in current prototype	Example system components
Food Production Cycle	Not implemented in code; referenced conceptually for future integration of traceability and batch-level nutrient data	Potential integration ports for production and labelling APIs
Food Consumption Cycle	Fully implemented	Scanner, Pantry Manager, Recipe Manager, Meal Logging, Progress and Insights
Activity-Level Cycle	Not implemented in code; reserved for future wearables and fitness imports	Hooks for fitness APIs and energy-expenditure modules

The architecture is therefore modular in a way that preserves ERP logic: each cycle can, in principle, operate independently under different stakeholders, yet all share a central datastore and a common user identity. This mirrors enterprise systems, such as SAP, where inventory, finance, and logistics modules are distinct yet integrated through a shared data model (Davenport, 1998).

4.5.4 Layer Responsibilities

Presentation layer: Next.js client

- Authentication screens (registration, login, password recovery).
- Pantry management interfaces presenting inventory cards with quantities, categories, and expiration status.
- Scanner screens for live camera capture and image upload.
- Recipe manager interfaces for setting constraints and selecting suggested meals.
- Meal logging forms and controls.
- Dashboard and Progress and Insights pages, including charts, macro breakdowns, streaks, and badges.
- Profile and goals configuration pages.

Application layer: FastAPI backend

- User authentication and authorization, including password hashing and token management.
- CRUD operations for all core resources (users, pantry items, recipes, food logs, metrics).
- The barcode process, which receives raw input, calls external services, and normalizes products into pantry records.
- Rule-based filtering and constraint handling for recipe generation.
- Macronutrient and calorie aggregation logic over daily and weekly windows.

- Calculation of behavioral metrics such as streak length and completion rates.
- Validation, transformation, and sanitization of incoming data.

Data layer: PostgreSQL schema

- Support one-to-many relationships, for example, between a user and their pantry items or meal logs.
- Reduce redundancy by separating ingredient-level nutrient information from specific consumption events.
- Preserve an audit trail through timestamps for item creation, updates, and log entries.
- Enable future extension to household-level or multi-user scenarios by associating multiple user accounts with a shared pantry or household identifier.

4.6 Module-Level Design

This section presents the internal design of the main modules that constitute the Human Nutrition Consumption System. For each module, the description covers its purpose within the consumption cycle, its Input–Process–Output structure, the principal backend logic, database interactions, representative user interface elements, and key design decisions. This level of detail supports transparency and reproducibility, in line with Design Science Research expectations that artifacts be grounded in explicit and inspectable design logic.

4.6.1 Authentication Module

Purpose: This module provides secure account creation and login, using both email and password credentials and Google OAuth. It ensures that all subsequent activities are link with a specific user identity and therefore, it can be personalized to individual goals and preferences.

Input–Process–Output structure:

- Inputs: Email address, password, optional Google OAuth token, and profile or goal-setting data for new users.

- Processes: Server-side password hashing, credential verification, token generation and validation, and onboarding flow for new accounts.
- Outputs: An authenticated session, error messages in the case of invalid credentials, and persisted user and goal records.

Storage: User records and associated goal configurations in the users and user_goals tables.

Backend logic: Passwords are hashed before storage using a secure algorithm, and login requests verify submitted credentials against the stored hash. Successful authentication results in the issuance of a JWT or equivalent token, which clients attach to subsequent requests. Google OAuth logins are supported by accepting tokens from the identity provider and, on first sign-in, creating a corresponding user record if one does not already exist. Newly registered users are directed through an onboarding sequence to specify daily calorie targets and macronutrient goals, which are stored in the user_goals table and later used by the analytics module.

Database interactions: The authentication module primarily interacts with:

- users table: email, hashed_password, name, profile_image, provider, provider_id, is_active, is_verified, reset_token, reset_token_expires, onboarding_completed, diet_preference, weight, height, age, gender, activity_level, health_goals, intolerances, allergies, id, created_at, updated_at
- user_goals table: user_id, daily_calories, daily_protein, daily_carbs, daily_fat, daily_fiber, weight_goal, id, created_at, updated_at

Each user has exactly one associated goals record, which simplifies later queries for personalized targets.

User interface elements: The user interface has registration and login screens, an option to sign in with Google, and an initial goal-setting page. These screens establish the user’s identity and ensure that subsequent views, such as the dashboard and progress metrics, are tailored.

Design decisions and rationale: Support for Google OAuth is intended to increase retention by removing the need for additional passwords during onboarding. Separating identity information from goal configuration increases flexibility, for instance, by allowing future extensions such as household or activity-cycle goals. Placing a brief onboarding flow immediately after account creation encourages users to articulate their objectives early, which can strengthen commitment and adherence in digital health tools.

4.6.2 Scanner Module

Purpose: This module enables users to log items in their pantry quickly through barcode scanning, image upload, or manual entry. By automating item capture wherever possible, the module addresses the burden of manual meal logging of diet-tracking applications.

Input–Process–Output structure

- Inputs: Barcode images, decoded barcode numbers, and manual item forms.
- Processes: Barcode decoding, calls to external nutrition APIs, validation, and normalization of items into a consistent internal format.
- Outputs: Pre-filled item forms for user confirmation of pantry records.

Storage: Pantry items persisted in the pantry_items table.

Backend logic: For barcodes, the backend decodes the supplied image or string, queries a nutrition API, and extracts product metadata and nutrient values from the response. The data are mapped to the internal schema and returned to the client as a structured object that can either be accepted or edited before saving. Manual entry allows users to create items when no barcode or reliable match exists, which is particularly important for culturally specific foods and home-prepared items.

Database interactions: The scanner module ultimately writes to the pantry_items table, which may include fields such as id, user_id, name, description, category, quantity, unit, barcode, fdc_id, brand_owner, a nutrients JSON field (storing calories and macro/micronutrients), expiry_date, and timestamps. Each record is linked to a specific user through user_id, reflecting the per-user pantry design.

User interface elements: The client provides a barcode scanner view with camera capture, an upload view for submitting images, and a manual entry form. These screens present pre-filled fields where possible, allowing users to confirm or adjust values before adding items to the pantry.

Design decisions and rationale: Providing multiple input pathways reflects the diversity of purchasing contexts and helps users gradually transition from manual entry toward more automated logging. Routing all acquisition modes through the same pantry schema maintains data consistency and simplifies subsequent processing. Allowing custom items ensures that the system can accommodate regional and culturally specific foods that may not be present in standard nutrition databases.

4.6.3 Pantry Module

Purpose: The pantry module acts as the user's personal inventory system, analogous to inventory management in ERP systems. It stores items acquired via the scanner or manual entry and supports the "food storage and preparation" segment of the consumption cycle by providing an up-to-date view of available ingredients.

Input–Process–Output structure:

- Inputs: Requests from the client to add, edit, or delete pantry items, as well as editing expiration dates, and searching and filtering inventory.
- Processes: Create, read, update, and delete operations; categorization and filtering; and sorting to expiration date or category.

- Outputs: Updated pantry lists, detailed item views, search results, and notifications about items approaching expiration.

Storage: The same pantry_items table used by the scanner module, ensuring a single source of truth for inventory data.

Backend logic: The pantry service exposes endpoints for listing a user's inventory, creating or updating items, and removing entries. It can filter items by category or tag and compute simple indicators such as "expiring soon" based on the current date and stored expiration values. Results may be ordered to prioritize items that should be used promptly, supporting both meal planning and reducing food waste.

Database interactions: The module operates over the pantry_items table, which is related to the users table via a one-to-many relationship. This design allows each user to maintain an independent inventory while still enabling future extension to shared household pantries through additional linking tables if required.

User interface elements: The pantry is presented as a card that displays item names, brands, quantities, key nutrient values, and expiration dates. Users can open item details to edit quantities or expiration and can filter or search to locate specific products or categories.

Design decisions and rationale: A card layout was selected to make it easy for the user to scan through key information, such as expiration status and nutrient highlights. Explicit expiration tracking aligns the system with sustainability goals by encouraging users to cook with items that are close to expiry. Treating the pantry as the main source for recipe generation supports proactive meal planning rather than purely reactive logging, which addresses limitations noted in existing diet-tracking applications.

4.6.4 Recipe Manager Module

Purpose: The recipe manager module supports the meal preparation stage of the consumption cycle by transforming pantry items and user preferences into concrete meal options. The module implements an inventory-driven approach in which available pantry items form the list of possible recipes, mirroring how ERP systems align production with inventory and supply.

Input–Process–Output structure:

- Inputs: Current pantry items, user preferences and dietary restrictions, and nutritional goals.
- Processes: Matching pantry items to recipes, generating and saving recipe options, applying filtering and ranking rules, and calculating nutritional information per portion.
- Outputs: A ranked list of recipe suggestions, detailed recipe views including ingredients and preparation steps, and associated nutritional breakdowns.

Storage: Optionally persisted recipes in a recipes table and corresponding entries in meal_logs when a recipe is logged as consumed.

Backend logic: The recipe suggestion process starts by retrieving the user's pantry items, then the backend queries an external recipe API using the available ingredients as input. Returned recipes are filtered according to rules such as matching a minimum number of pantry items, excluding ingredients the user has no preference for or is allergic to, and nutritional targets. For each candidate recipe, nutritional information is decomposed into calories and macronutrients, and portions are normalized to support consistent tracking across meals. Users may save selected recipes for later use or log them immediately. Logging a recipe creates an entry in the meal_logs table and triggers recalculation of the relevant analytics.

Database interactions: The module interacts with:

- recipes for storing a subset of frequently used or user-saved recipes.
- meal_logs for recording consumption of a recipe.

- pantry_items to determine the availability and suitability of ingredients.

In the current minimum viable product, recipe storage remains intentionally limited because recipes are primarily generated dynamically; a richer user-maintained recipe book is reserved for future iterations.

User interface elements: The interface presents a list of suggested recipes, detailed recipe views with ingredients and nutritional information, and controls for saving or logging a recipe. These screens connect pantry contents, user preferences, and consumption logging.

Design decisions and rationale: A lean storage strategy avoids unnecessary duplication while still allowing users to bookmark favorites. Using pantry-driven filtering differentiates the system from conventional fitness applications and encourages cooking with existing ingredients, which can reduce food waste. Automatic nutrient aggregation reduces the need for users to enter macronutrient data manually.

4.6.5 Meal Logging Module

Purpose: This module encompasses the meal consumption stage of the cycle by converting food choices into measurable nutrient intake records. It provides the main input to the Progress and Insights module.

Input–Process–Output structure:

- Inputs: A selected recipe with serving size information.
- Processes: Calculation of the nutrient profile, scaling by portion size, and macronutrients of the entry.
- Outputs: New entries in the meal log and updated metrics available to the dashboard and analytics module.

Storage: Records in the meal_logs table, which subsequently drive aggregated metrics.

Backend logic: The backend retrieves the nutritional profile, applies the selected serving size, and writes the resulting calorie and macronutrient values to meal_logs. After each log operation, the analytics service recalculates daily and weekly totals, progress toward goals, and streak information.

Database interactions: The module primarily uses the meal_logs table, which includes fields such as id, user_id, food_name, calories, protein, carbs, fat, meal_type, and timestamp. The table is append-only, preserving a historical record suitable for trend analysis and potential future research.

User interface elements: The user interface provides forms for meal planning and logging selected recipes. These views emphasize minimal interaction while maintaining clarity about what has been recorded.

Design decisions and rationale: Meal logging is designed to require as few actions as possible, in recognition of evidence that complex logging workflows reduce long-term use.

4.6.6 Dashboard and Progress & Insights Module

Purpose: This module supports the final stage of the consumption cycle. It aggregates meals into interpretable summaries, including daily calorie progress, weekly macronutrient trends, streaks and consistency metrics. These feedback mechanisms are central to behavior-change support in digital health interventions.

Input–Process–Output structure:

- Inputs: Meal-log records, user goals, and associated timestamps.
- Processes: Aggregation of nutrients over specified time windows, calculation of percentages relative to goals, derivation of streak and completion metrics, and preparation of data structures suitable for graphical visualization.
- Outputs: Dashboard views, charts, numerical summaries, and badges indicating milestones.

Storage: Aggregated values are primarily computed on demand from meal_logs and user_goals, with optional caching in a small metrics table.

Backend logic: Daily metrics are computed by summing calories and macronutrients for the current day and comparing these to the user's configured goals to obtain percentage completion values. Weekly trends are derived by grouping logs by day, calculating totals, and formatting the results as time-series data for visualization. Streak logic tracks whether a user has logged at least one meal per day; consecutive days with logs increase the streak, whereas a day without logs resets it. Badges or achievements are awarded at predefined thresholds such as seven, fourteen, or thirty consecutive days.

Database interactions: The module primarily reads from meal_logs and user_goals and can write summary statistics or streak information to a dedicated metrics table when caching is advantageous.

User interface elements: The dashboard presents summary tiles for daily calorie and macronutrient progress, charts displaying weekly trends, sections showing macro distributions, and panels for streaks and badges. The Progress and Insights view offers both weekly and monthly perspectives, enabling users to detect patterns over time.

Design decisions and rationale: Clear visual and immediate feedback is intended to support self-regulation for sustained change. Colored progress bars and simple charts enhance interpretability for non-expert users. The inclusion of streaks and badges leverages light gamification to encourage consistency without overwhelming users.

4.6.7 Profile and Settings Module

Purpose: This module allows users to manage their personal information, nutrition goals, dietary preferences, and account settings. It ensures that recipe suggestions and analytics are personalized and that the system can accommodate diverse cultural and dietary contexts.

Input–Process–Output structure:

- Inputs: Profile edits, goal updates, and changes to dietary preferences.
- Processes: Validation of data, updating of stored records, and adjust changes.

- Outputs: Updated profile information and recommendations and metrics.

Storage: Records in the users, user_goals, and preferences tables.

Backend logic: Profile endpoints modify basic demographic fields and contact details, while goal endpoints adjust daily calorie and macronutrient targets. Preference updates inform recipe generation logic by specifying diet types, intolerances, or disliked ingredients. The module interfaces with the authentication layer for password changes and account settings.

Database interactions: The module interacts with:

- users for core identity and contact information.
- user_goals for numeric targets used throughout the analytics pipeline.
- preferences for diet type, intolerances, and ingredient-level preferences.

User interface elements: The profile page presents account information, goal configuration forms, and preference settings.

Design decisions and rationale: Separating goals and preferences from basic identity information brings clarity and maintainability to the data model. Structured preferences also provide a foundation for more advanced personalization or AI-driven recommendation strategies in future work.

4.7 Data Model and Database Schema

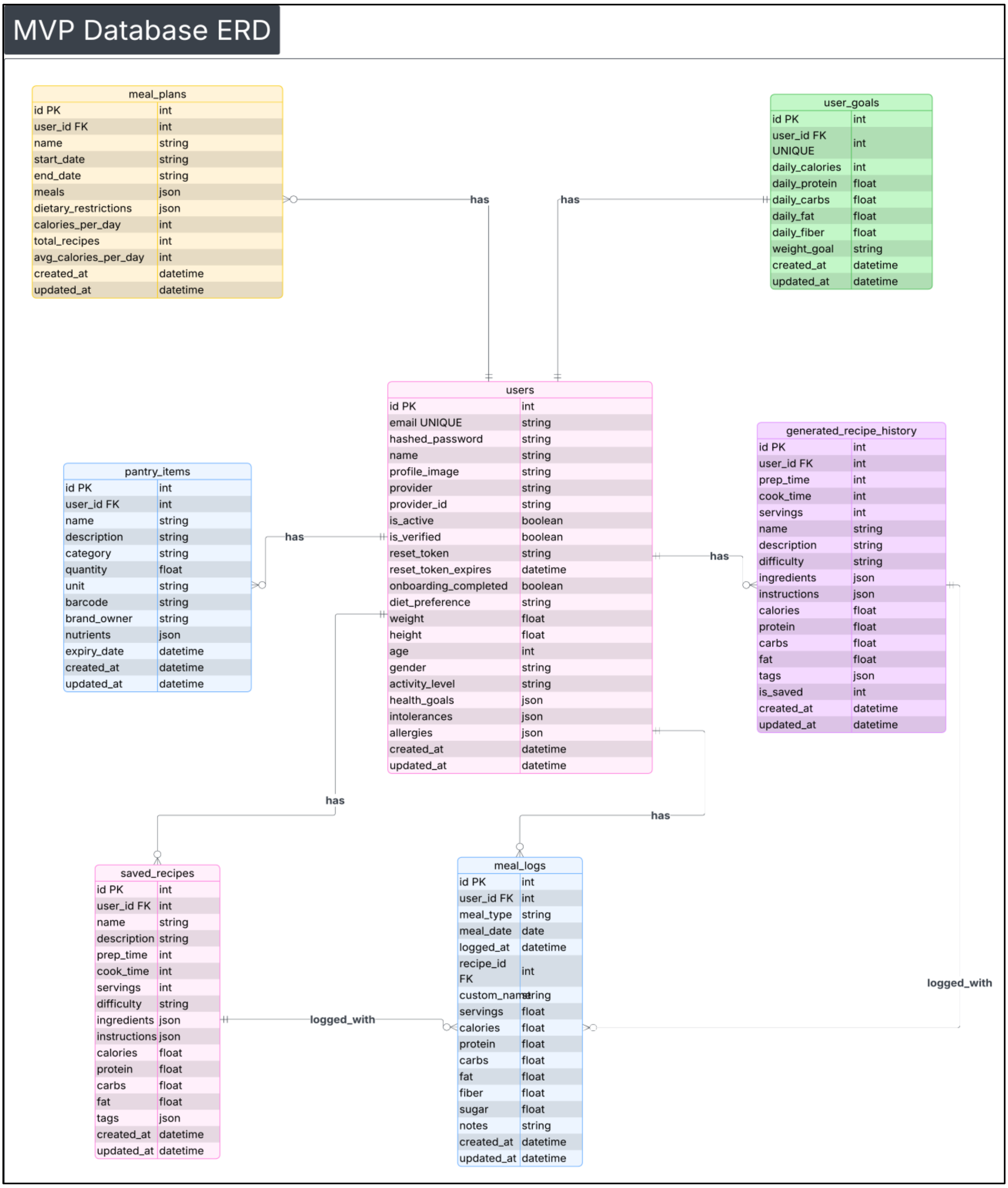
This section describes the PostgreSQL database schema that supports the consumption-cycle prototype of the Integrated Human Nutrition framework. The schema captures the core entities required for user management, pantry inventory, recipe generation, meal planning, meal logging, and progress analytics.

4.7.1 Overview Database Structure

The database consists of seven main entity types: Users, user_goals, pantry_items, saved_recipes, generated_recipe_history, meal_plans, and meal_logs

Together, these tables represent the minimum set of domain objects needed to operationalize the stages of the consumption cycle. All user-specific data are anchored on the users table. Other tables reference users.id as a foreign key, ensuring that privacy rules and access control can be enforced consistently across the system.

Figure 4.2. ERD and Database Schema



4.7.2 Core Tables

Users. The users table stores account-level identity, authentication, and high-level preference information. Key fields include a primary key id, unique email, hashed_password, OAuth fields (provider, provider_id), activation flags (is_active, is_verified), and password-reset tokens. Demographic and preference attributes such as weight, height, age, gender, activity_level, and JSON fields for health_goals, intolerances, and allergies are also stored here, alongside timestamps (created_at, updated_at). No plaintext passwords are stored, and all other tables that contain personal data reference users.id as a foreign key.

User goals. The user_goals table holds personalized numeric targets that drive the analytics layer, including daily_calories, daily_protein, daily_carbs, daily_fat, and daily_fiber, as well as a weight_goal. Each row references exactly one user via user_id, forming a one-to-one relationship. Keeping goals in a separate table from the core profile allows them to evolve independently and supports future extensions such as dynamic goal adjustment based on activity data.

Pantry items. The pantry_items table represents the user's inventory of foods available for meal preparation and recipe generation. Fields include user_id, name, description, category, quantity, unit, barcode, fdc_id (link to USDA FoodData Central where available), brand_owner, a nutrients JSON field, expiry_date, and timestamps. Each user may own many pantry items, producing a one-to-many relationship from users to pantry_items. Storing nutrient data inside a structured JSON field allows the system to keep full calorie and macro/micronutrient information without creating an overly wide table, while expiration dates support proactive food-waste reduction and recipe prioritization.

Saved recipes. The saved_recipes table stores recipes that users choose to retain in their personal library. Typical fields include user_id, name, description, prep_time, cook_time, servings, difficulty, JSON fields for ingredients and instructions, aggregated nutrients (calories, protein, carbs, fat), tags, and timestamps. This table represents stable recipes that users explicitly save, which can later be logged as meals or incorporated into meal plans.

Generated recipe history. The generated_recipe_history table records recipes produced by the AI/recipe engine, even if they are not saved permanently. It mirrors many of the fields in saved_recipes (user_id, name, description, times, nutrients, tags), and adds a boolean is_saved flag indicating whether a generated recipe was later promoted to the saved library. This history supports future analytics on recipe usage, suggestion quality, and user preferences without polluting the canonical saved-recipes set.

Meal plans. The meal_plans table represents higher-level planning artefacts spanning multiple days. Key fields include user_id, name, start_date, end_date, a JSON meals structure (for example, mapping days and meal types to recipe IDs), dietary_restrictions, calories_per_day, variety_preference, total_recipes, avg_calories_per_day, and timestamps. Although the current MVP focuses primarily on on-demand recipe generation and logging, this table provides a scaffold for future features such as weekly meal plans and automated grocery lists.

Meal logs. The meal_logs table stores all consumption events and is the primary input to daily and weekly nutritional analytics. Fields include user_id, meal_type (e.g., breakfast, lunch, snack), meal_date, logged_at, optional recipe_id (link to a saved or generated recipe), custom_name for free-text entries, servings, nutrient values (calories, protein, carbs, fat, fiber, sugar), notes, and timestamps. The table is treated as append-only: existing rows are never overwritten, only added or logically removed, which preserves a complete historical record suitable for time-series analysis, streak detection, and future research use.

4.7.3 Entity Relationships and Normalization

The principal relationships in the schema are:

- One-to-one from users to user_goals.
- One-to-many from users to pantry_items, saved_recipes, generated_recipe_history, meal_plans, and meal_logs.
- Optional links from meal_logs.recipe_id to saved_recipes.id (and, conceptually, to generated recipes) when the log corresponds to a structured recipe.

- Implicit relationships between meal_plans.meals (JSON structure) and saved_recipes or generated recipes, resolved at the application layer.

The schema is designed to approximate third normal form for core entities. Identity and authentication data reside solely in users; numeric goals are stored in user_goals; stable inventory data live in pantry_items; and individual consumption events are restricted to meal_logs. JSON fields (nutrients, ingredients, instructions, preference-related fields on users) are used deliberately to accommodate semi-structured data that may change over time without requiring frequent schema migrations. This balance keeps redundancy low while preserving flexibility for evolving nutrition ontologies and recipe formats.

4.8 Technology Stack, Privacy, and Security

This section describes the technologies, tools, and practices used to implement the consumption-cycle subsystem of the Integrated Human Nutrition framework. The implementation follows a three-layer structure: frontend, backend application programming interface, and database, supported by barcode integrations, enabling efficient food-acquisition workflows. Together, these technologies form a cohesive and scalable web application architecture.

- Frontend: React / Next.js. The client layer is built with React and Next.js, using functional components, API routing, and context management. Styling is handled through TailwindCSS, and the user interface includes dedicated screens for all modules This combination offers responsive, modular interfaces while rendering overhead manageable.
- Backend: FastAPI (Python). The application layer uses FastAPI, chosen for its high performance, OpenAPI documentation, and straightforward support for asynchronous operations and future integration with machine learning components. The backend is responsible for authentication and token management, nutritional calculations,

CREATE-READ-UPDATE-DELETE operations across pantry and meal logs, orchestration of external API calls, and aggregation of analytics such as daily metrics, weekly charts, and streak calculations.

- Database: PostgreSQL. The data layer uses PostgreSQL as a relational store supporting indexing, SQL support, and JSON fields where flexible structures are required. Its schema directly supports the Input–Process–Output chains across scanner, pantry, recipe, logging, and insights modules, enabling consistent computations for analytics and future integration with production- and activity-cycle data sources.
- APIs and barcode integrations. The scanner module relies on barcode-decoding tools to reduce manual effort in food acquisition. This feature extract item names and quantities from receipts or packaging that can then be resolved through food database OpenFoodFacts. These integrations substantially lower the logging burden, addressing a major barrier to sustained use of diet-tracking tools.

4.8.1 Environment Configuration and Deployment Considerations

Environment-specific configuration files allow separation of development and future production settings, including distinct credentials for databases, email servers, and external APIs. The backend is hosted on Render, and the frontend can be deployed via modern hosting platforms, in this case, it is hosted on Vercel. While PostgreSQL can be hosted on managed database services and managed serverless through Neon. This separation supports clean migration strategies and scalability if the system is later extended beyond a research prototype.

4.8.2 Privacy and Data Protection

Data collection is limited to fields required for functionality; no unnecessary personal identifiers or sensitive data is stored. Identity information is separated from activity data by maintaining user profiles in distinct tables from meal logs and preferences, and there is no sharing of personal data across users. All client-server communication occurs over HTTPS, and third-party API keys are stored in environment variables rather than exposed. The implementation incorporates standard security practices, such as hashed passwords. No plaintext passwords are logged, and OAuth identifiers prevent account duplication. Also, the prototype is designed to ensure users retain control over their data, which is used solely for personalized insights and not for sharing or commercial purposes. Furthermore, the prototype deliberately avoids collecting sensitive clinical measurements, focusing instead on dietary patterns and goals.

4.9 Scenario-Based Demonstration

This section presents a narrative walkthrough of the prototype in use, illustrating how a typical user progresses through the food consumption cycle using the implemented system. The scenarios explicitly connect interface steps to the conceptual stages introduced in Chapter 3 and trace the complete Input–Process–Output flow across modules, which is a recommended form of artifact demonstration in Design Science Research.

For illustration, the scenarios follow a single user, Maria, who seeks to improve her daily nutrition by planning meals more deliberately and increasing protein intake.

4.9.1 Step 1: Stocking the Pantry (Acquisition and Storage)

Maria begins by stocking her digital pantry, corresponding to the acquisition and storage stages of the consumption cycle. After purchasing groceries such as chicken breast, rice, broccoli, and eggs, she opens the scanner module and selects the barcode-scanning option. The barcode is decoded by the backend, which queries an external nutrition database and returns product metadata and nutrient values; the pantry item form is automatically populated for confirmation.

For items without barcodes, such as loose broccoli, Maria inputs the product manually. A service extracts text, and the backend parses it and, where possible, matches item names to known foods. Maria confirms quantities and inputs expiration dates before saving. As a result, her pantry inventory is updated, nutrient profiles are stored in the database, and expiration tracking is activated for perishable items.

Figure 4.4: Scanner view for barcode capture

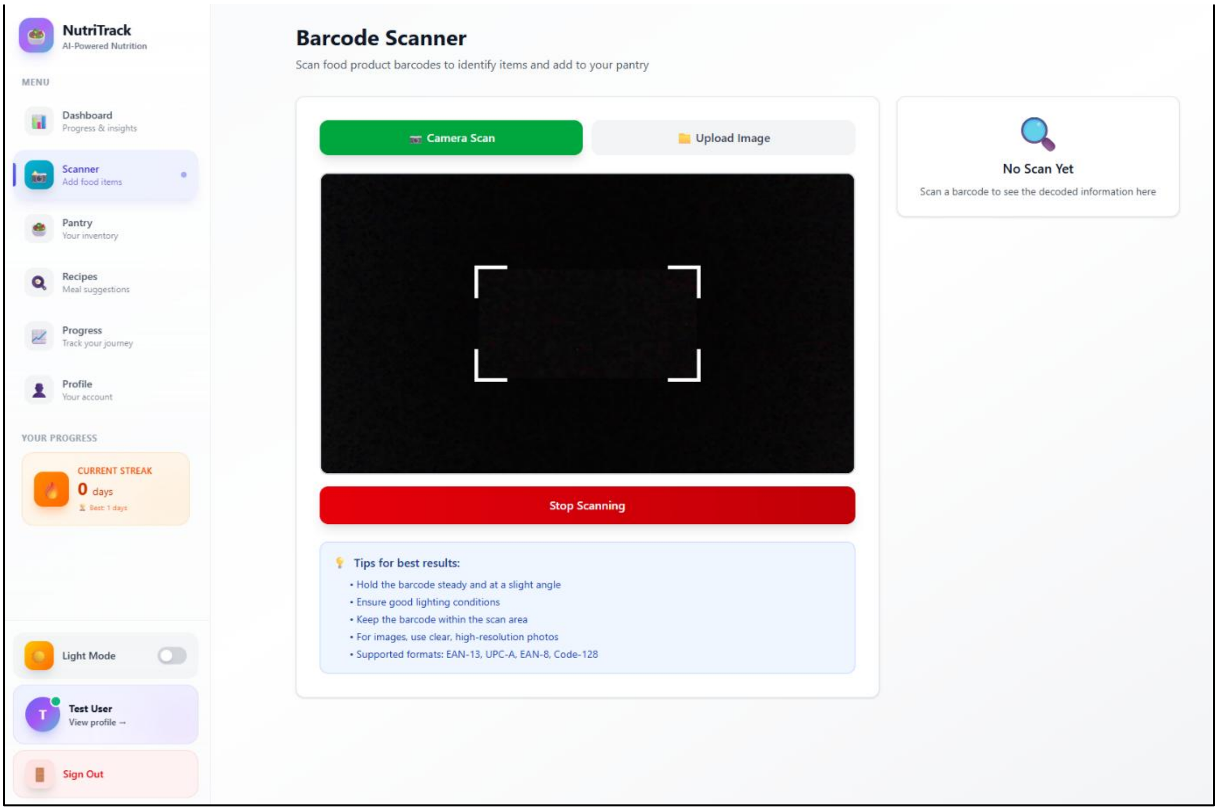
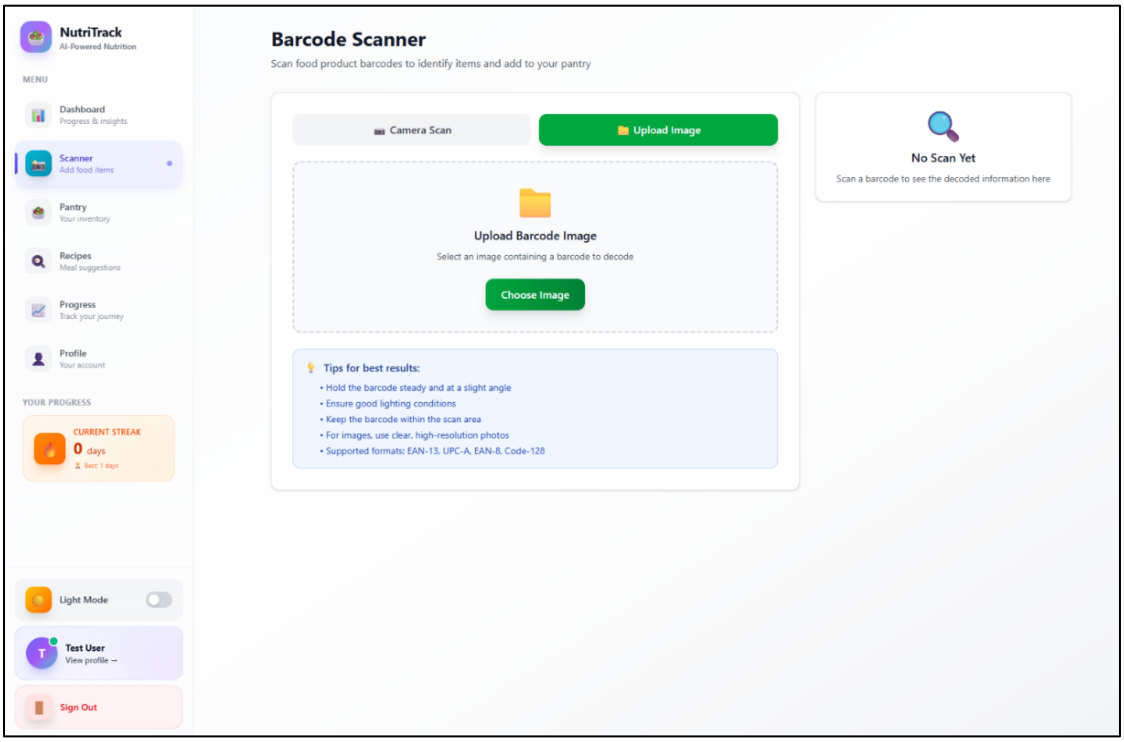


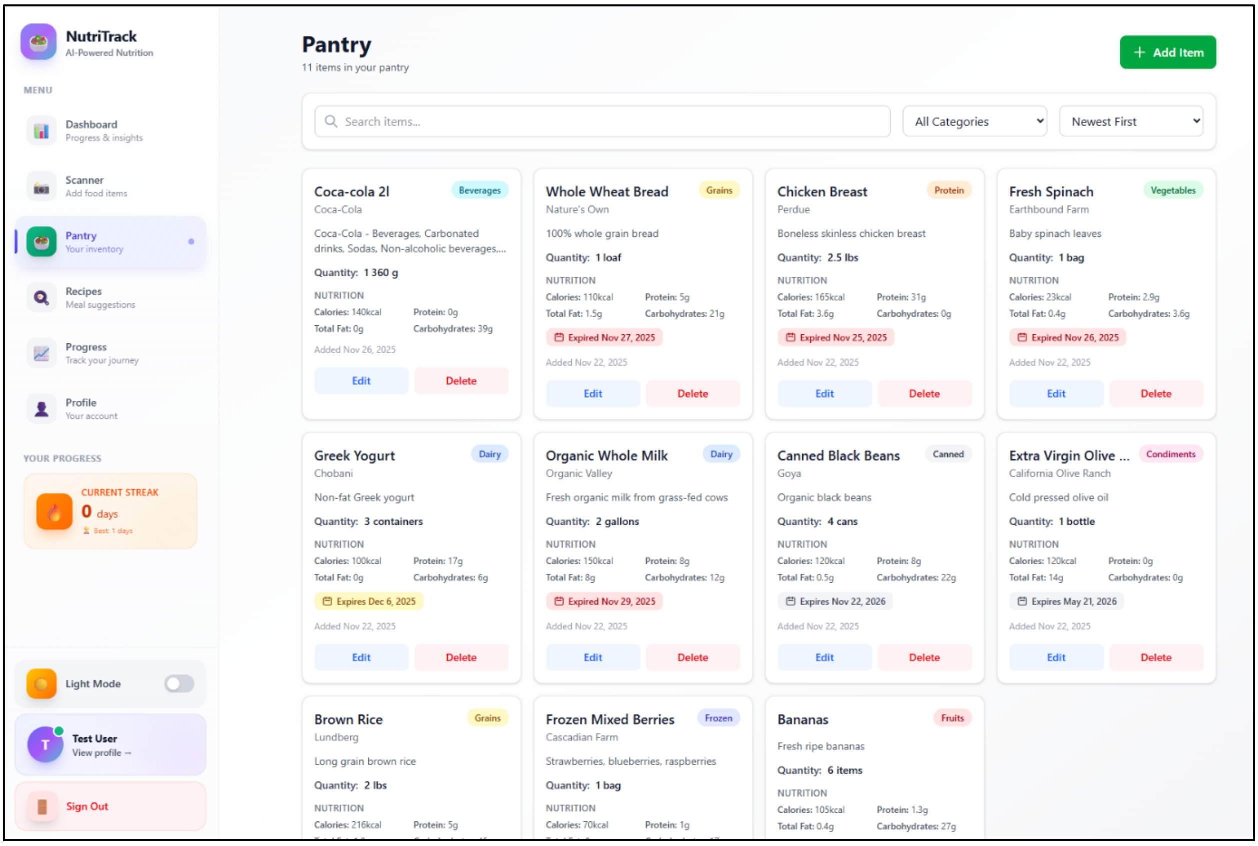
Figure 4.5: Image upload view



4.9.2 Step 2: Viewing and Managing the Pantry (Inventory Control)

Maria then navigates to the pantry view, which represents the storage and availability stage. Items are displayed as cards, such as protein sources, vegetables, grains, and dairy. Each card shows calories and macronutrients, quantity, expiration date, and brand, where available. The system highlights items approaching expiry, and Maria notices that her eggs will expire in two days. The pantry thus becomes both an overview of stock and the primary data source for subsequent recipe generation, mirroring inventory-control functions in enterprise systems.

Figure 4.6: Pantry overview with category filters; subset of items flagged as “expiring soon.”

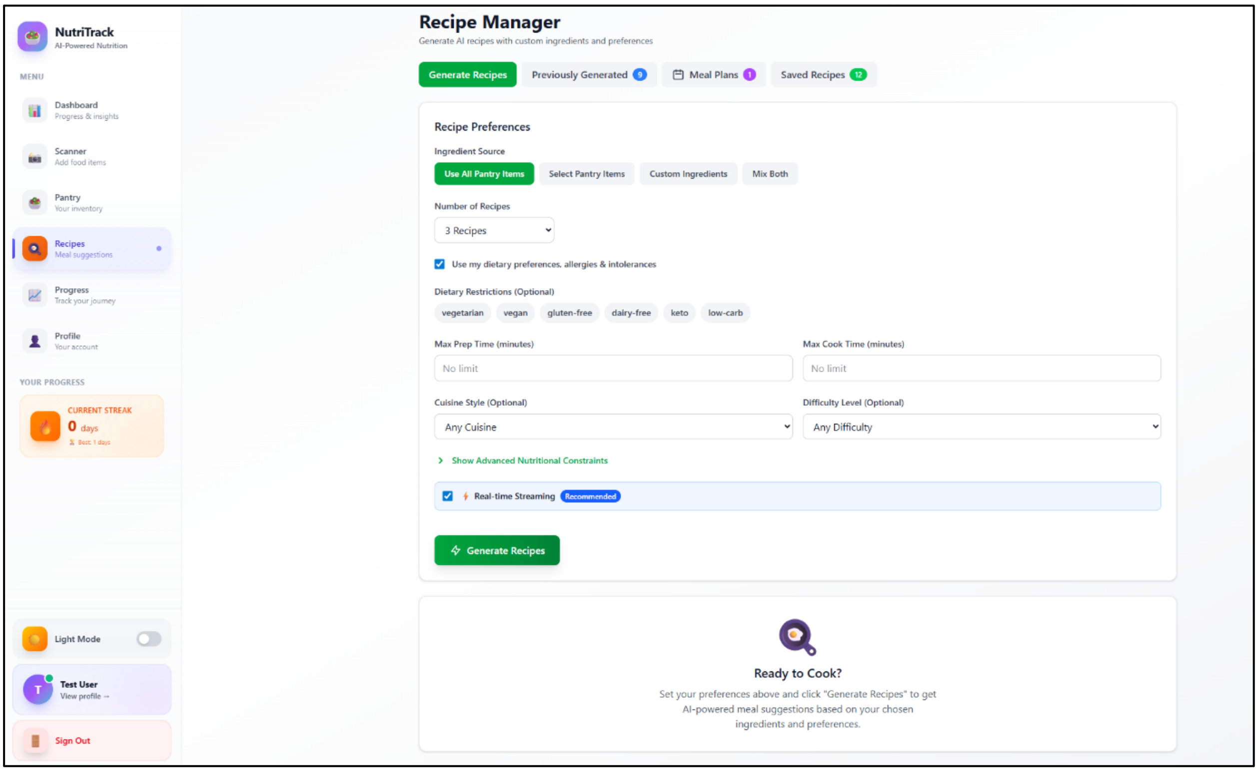


4.9.3 Step 3: Generating Recipes (Meal-Preparation Stage)

When planning lunch, Maria opens the recipe manager, which supports the meal-preparation stage by generating options based on real inventory. The module reads her pantry and identifies feasible combinations, such as chicken, rice, and broccoli. She applies filters to prioritize high-protein meals under a 600-calorie threshold.

The backend sends the selected ingredients and filters to the recipe service, which obtains candidate recipes, ranks them according to matching ingredients and goal alignment, and returns a list of suggestions. Each suggestion includes a full ingredient list, preparation instructions, and a nutritional breakdown per serving. This workflow demonstrates how the prototype shifts from reactive logging toward proactive planning, helping users understand what they can cook with what they already have.

Figure 4.7: Recipe suggestion list; recipe detail with nutrient summary and “log meal” controls.

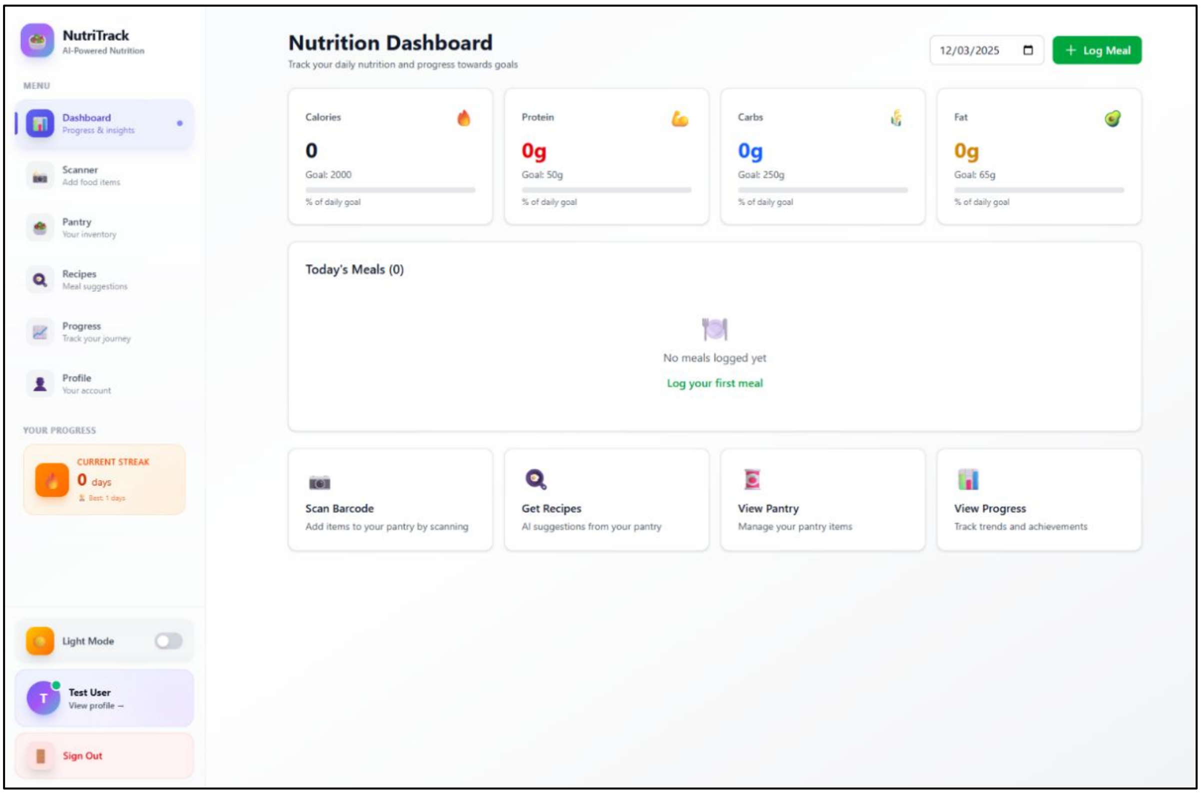


4.9.4 Step 4: Logging the Meal (Consumption Stage)

After cooking a grilled chicken bowl, Maria records it in the system. By selecting “log meal” on the dashboard page, she specifies the appropriate serving size or any meal from the recommendations or saved recipes. The backend retrieves the recipe’s nutrient profile, scales it by serving multiplier, and writes the resulting values to the meal_logs table together with the meal type and timestamp.

Immediately after logging, daily totals and weekly aggregates are recalculated, and the streak logic evaluates whether she has logged at least one meal on that day. This step implements the consumption stage of the cycle and supplies the analytics module with the raw events required for trend analysis.

Figure 4.8: Meal-logging confirmation; list of meals logged for the current day.



4.9.5 Step 5: Monitoring Progress (Reflection Stage)

After several days of consistent logging, Maria opens the Progress and Insights view, representing the monitoring and reflection stage. The dashboard presents her progress toward daily calorie and macronutrient goals, along with a bar chart of recent intake and a breakdown of macronutrient distribution. Weekly trend charts show how calories and macronutrients vary across days, and a streak counter displays the length of her current logging streak, with badges awarded at milestones such as seven consecutive days.

These visual summaries help Maria identify patterns, such as days when protein intake falls short, and provide positive reinforcement through streaks and achievements. The system thereby closes the feedback loop that is essential for supporting longer-term behavior change in nutrition-tracking tools.

Figure 4.9: Daily metrics dashboard; weekly progress chart; streak and badges panel - A

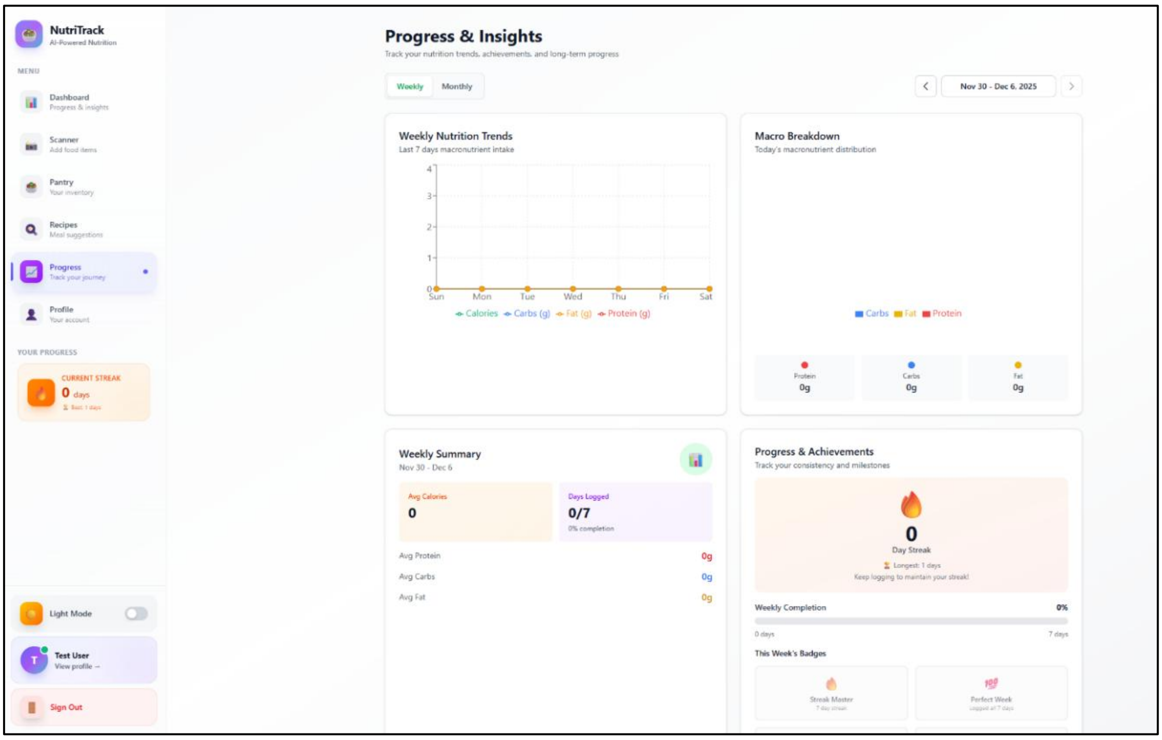
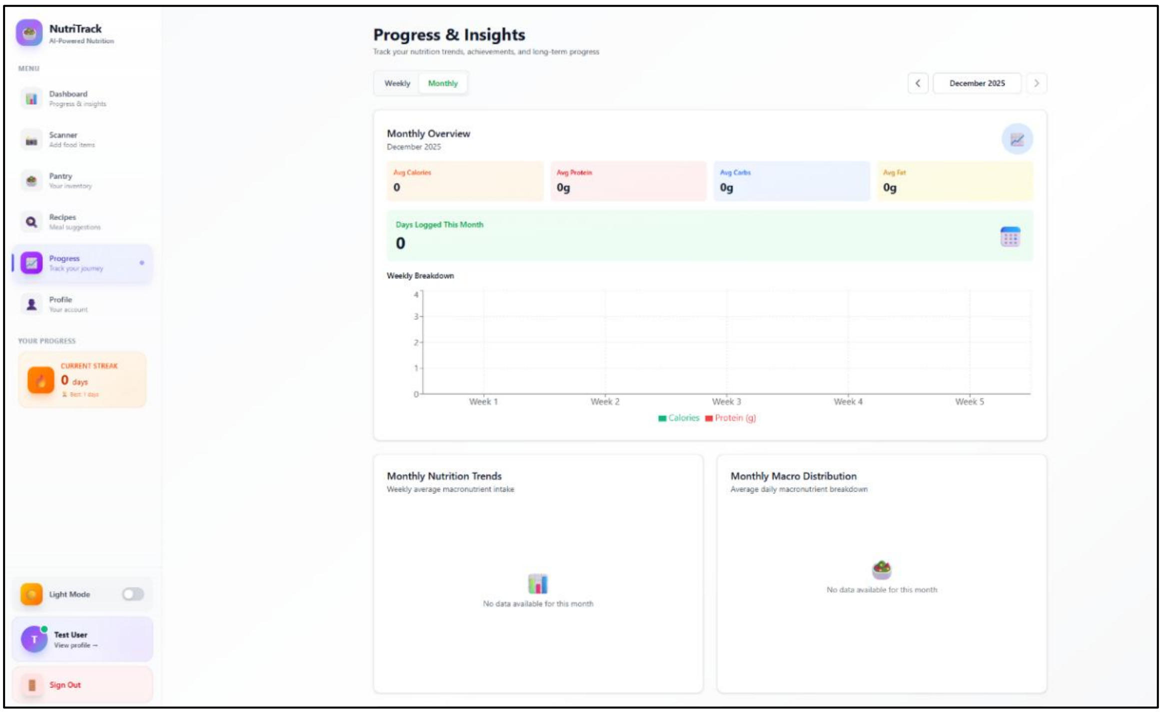
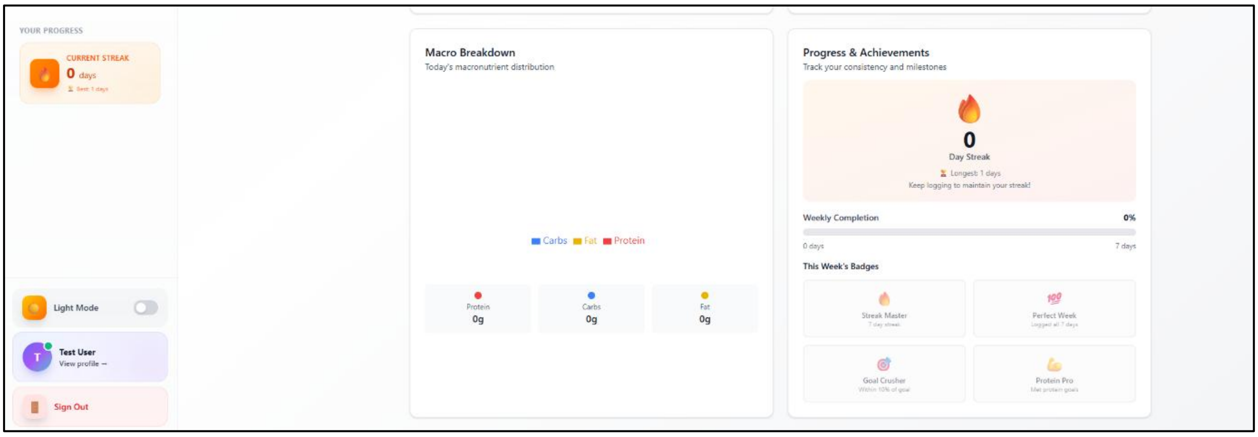


Figure 4.10: Daily metrics dashboard; weekly progress chart; streak and badges panel (B, C).





Taken together, the scenarios illustrate how the prototype implements the end-to-end consumption cycle. Acquisition and storage are realized through barcode and OCR-based stocking of the pantry; storage and availability are managed via the pantry module and its expiration tracking; preparation is supported by the recipe manager's pantry-aware suggestions; consumption is recorded through the meal-logging workflows; and monitoring is achieved through dashboards and streak visualizations.

This sequence can be summarized as an Input–Process–Output loop: food items enter the system through scanning and manual entry, are transformed into recipes, logs, and aggregated analytics, and finally produce dashboard insights that feed back into user decision-making. The scenario-based demonstration thus shows that the implemented artifact operationalizes the conceptual Integrated Human Nutrition framework for the consumption cycle in a coherent and practically usable manner.

CHAPTER 5: EVALUATION AND RESULTS

5.1 Introduction to Evaluation

Evaluation is a key component of Design Science Research, ensuring that an artifact is both rigorously constructed and relevant to the problem it addresses. Hevner et al. highlight that evaluation substantiates an artifact's utility, quality, and efficacy within its intended context, thereby validating its contribution to the knowledge base. In this thesis, evaluation is particularly significant because the prototype is designed to reorganize fragmented nutritional behaviors into a coherent, enterprise resource planning (ERP)-inspired consumption cycle. Accordingly, the assessment must determine whether this restructuring yields practical benefits for users and whether the system aligns with the conceptual framework established in Chapters 3 and 4.

This chapter utilizes a scenario-based evaluation, a conceptual alignment analysis, and a structured comparison with an existing nutrition-tracking application to provide an external benchmark for the prototype's performance. The scenario-based evaluation applies the user workflows presented in Chapter 4.6 as test cases to assess whether each stage of the consumption cycle—acquisition, storage, preparation, consumption, and monitoring—functions smoothly in practice. Furthermore, the conceptual alignment analysis examines the degree to which the implemented system reflects the Integrated Human Nutrition framework and the Input–Process–Output structures specified during design, including alignment with requirements R1–R6.

In addition to scenario-based and conceptual evaluations, this chapter presents a structured comparison with a widely used nutrition-tracking application (MyFitnessPal). This comparison serves as an external benchmark to assess the prototype's relative capacity to reduce friction, improve data consistency, enhance lifecycle integration, and increase the visibility of nutritional behaviors—factors frequently identified in the literature as barriers to effective nutrition management. By contrasting task workflows, reliance on manual entry, integration gaps, feedback delays, and coverage of consumption-cycle stages, the comparison provides further evaluative evidence and situates the prototype within the broader digital nutrition landscape. This approach strengthens the evaluation by demonstrating both the system's internal performance and its capacity to address limitations present in current commercial tools.

Taken together, these evaluation strategies, including the structured comparison with a commercial nutrition application, provide a comprehensive understanding of the artifact's effectiveness in addressing the research problem. They facilitate assessment of technical correctness, practical viability, and relative advantages over existing tools, as well as the potential to support more informed and health-conscious decision-making.

5.2 Evaluation Objectives

The evaluation of the consumption-cycle prototype is guided by five core objectives derived from the research questions and design goals established in Chapters 1 and 3. Together, they assess whether the implemented consumption-cycle system is conceptually sound, technically correct, and capable of improving the organization and visibility of nutritional behaviors.

- EO1 – Implementation of the consumption-cycle framework. Determine whether the prototype accurately implements the five stages of the consumption cycle, as defined within the Integrated Human Nutrition framework, and the Input–Process–Output structures defined in Chapter 3.
- EO2 – Assess usefulness in reducing fragmentation and friction in key nutritional tasks. The second objective examines whether the prototype meaningfully integrates tasks that are typically scattered across multiple nutrition tools, thereby reducing friction in logging, planning, and monitoring.

- EO3 – Enhanced visibility of nutritional behaviors. The third objective assesses whether the prototype improves users' ability to understand their dietary patterns. The Integrated Human Nutrition model assumes that better outcomes require more integrated and transparent information flows, including weekly intake, consistent logging, macronutrient balance, and the use of pantry items. This objective evaluates the extent to which the dashboard and insights components provide meaningful and comprehensible visibility into these behaviors.
- EO4 – Assess functional correctness and architectural consistency. These objective concerns whether core functions (scanning, pantry management, recipe generation, meal logging, and analytics) behave as intended when executed through realistic scenarios and whether the system maintains internal consistency with the architectural design requirements defined in Chapter 4. This includes adherence to modular boundaries, consistent data flow across modules, and support for future integration with the production and activity cycles.
- EO5 – Benchmarked comparison against existing nutrition-tracking tools. The fifth objective evaluates how the prototype performs relative to established applications, such as MyFitnessPal, along dimensions including friction, data consistency, reliance on manual entry, integration gaps, feedback latency, and visibility of behavioral patterns. This objective assesses whether the prototype offers measurable advantages over siloed tools and clarifies its contribution within the current digital-nutrition ecosystem.

5.3 Evaluation Methods

The evaluation of this prototype employs a combination of conceptual and empirical techniques rather than relying on a single method. Because the Human Nutrition Consumption System integrates a formal framework and process logic, the evaluation combines conceptual analysis, scenario-based assessment, and benchmark comparison. Taken together, these approaches examine internal coherence, alignment with the consumption-cycle framework, and overall system performance.

Conceptual evaluation focuses on theoretical alignment. It investigates whether the implemented system accurately reflects the Integrated Human Nutrition framework and the Input–Process–Output structures specified during design, including adherence to the functional requirements R1–R6. This involves reviewing the correspondence between modules and framework stages, checking the correctness of workflow sequences across scanner, pantry, recipe, logging, and analytics components, and assessing the completeness of the implemented features relative to the design. In Design Science Research, such conceptual validation is essential, since an artifact that does not embody its underlying theory cannot meaningfully address the problem it is intended to solve.

Scenario-based evaluation utilizes the workflows presented in Chapter 4.6 as practical test cases. Instead of re-enacting these scenarios in detail, the system is assessed against predefined success criteria for each stage of the consumption cycle, including the completeness of the flow from acquisition through monitoring, the accuracy of system outputs such as nutrient totals and inventory updates, and the clarity of processes from the perspective of user support. This method provides evidence that the system functions as intended under realistic conditions and that the stages are interconnected in accordance with the designed consumption-cycle logic.

Finally, benchmark comparison involves comparing the prototype against established nutrition tracking apps such as MyFitnessPal. This comparison evaluates several factors, including friction (number of steps per task), data consistency, the dependency on manual entry, integration gaps, feedback delays, and visibility of behavior patterns. The goal of this method is to assess whether the prototype offers measurable advantages over siloed tools and whether it aligns with or exceeds the performance of existing applications in addressing fragmented nutrition behaviors.

Combining conceptual evaluation, scenario-based assessment, and benchmark comparison enables the study to capture both the prototype's conceptual soundness and practical usability. This methodological triangulation enhances the validity of the findings and supports interpreting the artifact as both a functioning software system and a concrete instantiation of the Integrated Human Nutrition framework developed in this thesis.

5.3.1 Conceptual Alignment

The first evaluation method examines the conceptual alignment of the prototype with the Integrated Human Nutrition framework introduced in Chapter 3. In a Design Science Research project, this form of evaluation is important for determining whether the artifact system is both functional and theoretically grounded.

A primary criterion involves the degree of alignment between the prototype and the three-cycle Integrated Human Nutrition model. Although only the consumption cycle is implemented, the system must preserve the structural logic that would allow the food production and activity-level cycles to be added later. The architecture described in Chapter 4 maintains modular boundaries between subsystems, uses a shared data structure centered on the individual user, and organizes processes in a manner consistent with enterprise resource planning workflows, thereby enabling additional cycles to be integrated without fundamental reengineering.

A second criterion is the consistency of interactions among the core modules, particularly the sequence from scanner to pantry, recipe manager, meal logs, and progress insights. This sequence corresponds to the five functional stages of the consumption cycle, and conceptual evaluation assesses whether each module provides the inputs required by the next and exposes the expected outputs. The scanner and optical-character-recognition components generate structured pantry items; the pantry produces ingredient sets for recipe generation; the recipe manager outputs meals that can be logged; and logging events feed directly into the analytics that underpin the dashboard and Progress and Insights views. The Input–Process–Output structures specified in Chapter 4.3 demonstrate this coherence and make the lineage of data across the system explicit.

The evaluation also considers compliance with the functional requirements R1–R6 defined in Section 4.1. The prototype enables users to authenticate and define goals (R1), manage pantry inventory (R2), add items through multiple acquisition methods, including scanning and manual entry (R3), generate pantry-aware recipes (R4), log meals based on recipes or custom entries (R5), and review daily and weekly progress through analytical views (R6). While the dashboard does not yet provide long-term analytics or integration with activity-level expenditure, these omissions are consistent with the minimum viable scope of the current Design Science artifact and do not undermine the satisfaction of the stated requirements.

Overall, the conceptual evaluation indicates that the prototype constitutes a structurally sound and theoretically faithful instantiation of the Integrated Human Nutrition framework. Its design reflects the principles of modularity, traceable data flow, and lifecycle integration that motivated the conceptual model, and it offers a coherent foundation for future extensions into the production and activity-level cycles envisioned in the broader framework.

5.3.2 Scenario-Based Evaluation

The second evaluation method employs scenario-based assessment to examine how the prototype performs across realistic sequences of actions derived from the consumption cycle. In contrast to usability testing, which concentrates on user interaction and subjective experience, scenario-based evaluation focuses on the logical correctness and completeness of system behavior when executing full end-to-end workflows. The scenarios defined in Chapter

4.6 serve as reference cases, representing how a typical user progresses from food acquisition to monitoring outcomes through the Human Nutrition Consumption System.

Scenario-based evaluation applies predefined success criteria to each stage of these scenarios. The criteria assess whether the system processes inputs correctly, maintains data integrity across transitions, and generates outputs consistent with the conceptual model. During the acquisition stage, for example, success is defined as the accurate creation of structured pantry entries from scanner and optical-character-recognition input. In the preparation stage, success requires that the recipe manager uses pantry data and user preferences to produce viable recipe options. During the consumption and monitoring stages, success depends on whether nutritional values are aggregated correctly; meal logs are stored with appropriate timestamps, and progress indicators update promptly to reflect new entries.

The method also examines the completeness and continuity of the overall flow, that is, the extent to which one stage leads naturally into the next without missing information or contradictory results. Pantry items created during acquisition should immediately be available for recipe generation, and logging a meal should be reflected without delay in daily and weekly progress metrics. Scenario-based evaluation, therefore, assesses not only the correctness of individual modules but also the integrity of the workflow as a process-driven system aligned with the Integrated Human Nutrition model.

Finally, the evaluation considers the internal flow and clarity of outputs, including the consistency of nutritional totals, streak counts, progress indicators, expiration flags, and dashboard visualizations relative to the underlying log data. While the prototype does not aim to achieve clinical-grade precision, internal consistency within the system is treated as a key indicator of conceptual and functional soundness. By using full consumption-cycle scenarios as evaluation units, this method provides a structured way to validate that the minimum viable product functions cohesively and in accordance with the lifecycle logic specified in the framework.

5.3.3 Experimental Benchmark Comparison

The third evaluation method is a benchmark comparison between the Human Nutrition Consumption System and an existing nutrition-tracking application, specifically MyFitnessPal, which serves as a widely used reference point in consumer nutrition technology. This comparison does not aim to measure superiority or performance outcomes but instead provides a structured way to assess how the prototype differs from a typical siloed application with respect to integration, friction, and visibility.

The comparison evaluates a set of categories defined by the designer based on direct interaction, real-time testing of workflows, and examination of how each system supports common nutrition-related tasks. The intention is to assess how design choices manifest in practical functionality, rather than to make empirical claims about user outcomes. The benchmark focuses on the following categories:

1. Friction and task complexity: Measures the number of operational steps required to complete common tasks such as adding items, planning meals, and logging consumption. This helps identify where workflows become time-consuming or repetitive.
2. Dependency on manual entry: Evaluates the extent to which each system requires users to type information manually, versus drawing from structured data sources or automated processes.
3. Data consistency and structure: Examines whether information remains aligned across different parts of the system, and how each application handles updating, linking, or reusing data.

4. Integration gaps across lifecycle stages: Looks at which portions of the food lifecycle are covered by each system (e.g., acquisition, storage, preparation, consumption, monitoring) and whether they operate as connected processes or as isolated tasks.
5. Feedback immediacy and update latency: Observes how quickly each system reflects logged actions in dashboards or summaries, focusing on the smoothness of transitions between tasks.
6. Visibility of behavior patterns: Assesses how clearly each application presents weekly trends, nutritional balance, consistency patterns, and other indicators that help users understand their dietary habits.

This method supports Design Science Research by offering a practical, structured comparison that places the prototype within the context of existing tools. It highlights meaningful design differences that can be examined in the results chapter without drawing inferential or behavioral conclusions.

5.4 Evaluation Results

5.4.1 Conceptual Alignment Results

The conceptual evaluation examined how well the implemented prototype operationalizes the Integrated Human Nutrition framework and the stages of the consumption cycle defined in Chapter 3. Overall, the analysis indicates strong conceptual alignment, with the system faithfully reflecting the structured flow from acquisition, through storage, preparation, and consumption, to monitoring.

First, the prototype supports the Input–Process–Output logic that underpins the framework. Scanner inputs from barcodes, optical-character-recognition images, and manual forms are transformed into structured pantry items; pantry data then serves as inputs to recipe generation; and meal logs feed directly into analytics processes that compute progress metrics. Each module consumes and produces data in a manner consistent with the subsystem design specifications, demonstrating internal consistency across the architecture.

Second, the system maintains clear modular boundaries that mirror the framework's staged approach. The scanner components do not interact directly with recipe creation; the pantry functions as the central hub for inventory; and the Progress and Insights module operates downstream from logging. This separation of concerns is important for extensibility and confirms the conceptual soundness of the design.

Third, the prototype implements all functional requirements R1–R6 for the consumption-cycle minimum viable product. Users can authenticate and set goals, manage inventory, add items through multiple acquisition methods, generate recipes, log meals, and view daily and weekly progress. While some dashboard elements remain simplified; for example, long-term analytics and activity-level integration are not yet present.

Finally, the overall data-flow structure, particularly the consistent movement of structured nutritional values across modules, reflects the ERP-inspired principles of lifecycle management and integrated decision support. In summary, the conceptual evaluation validates that the artifact is theoretically coherent, aligned with Design Science Research principles, and correctly instantiates the designed consumption-cycle subsystem.

5.4.2 Scenario-Based Evaluation

Results Scenario-based evaluation assessed how the system performs across realistic workflows representative of the consumption cycle. Using the scenarios defined in Chapter 4.6 as test sequences, the prototype was evaluated against success criteria including flow completeness, process clarity, and output accuracy.

- Stage 1 – Acquisition. The scanner reliably converted barcode, and manual inputs into pantry items, meeting the acquisition criteria. Scanner performance varied with product brand, but the availability of manual entry ensured process continuity.

- Stage 2 – Storage (Pantry). Pantry operations behaved consistently with expectations. Items added during acquisition appeared immediately, and expiration tracking updated correctly. Editing, deleting, and filtering items functioned without errors, indicating stability of this stage and clear correspondence between interface behavior and user mental models of inventory.
- Stage 3 – Preparation (Recipe Generation). The recipe manager correctly utilized pantry contents to generate viable recipe options. Ingredient-matching logic reflected available items, and nutritional breakdowns were accurate with respect to the underlying data. Some advanced features could include personalization controls, such as stronger filtering of disliked ingredients, which represents a potential enhancement rather than a conceptual flaw.
- Stage 4 – Consumption (Meal Logging). Meal logging operated smoothly. Selecting a recipe and logging it as consumed updated daily totals immediately, and the success criteria for this stage accurate aggregation of calories and macronutrients and persistence of log entries were fully met. Logged meals appeared consistently in dashboard views without discrepancies.
- Stage 5 – Monitoring (Progress and Insights). Daily and weekly insights populated correctly, and streak counters updated as expected when meals were logged on consecutive days, with access to weekly and monthly insights.

Overall, the scenario-based evaluation shows that the system performs cohesively across the entire consumption cycle. Workflows follow the intended process sequence without broken links or major errors, and outputs at each stage align with the expectations of the Integrated Human Nutrition model.

5.4.3 Experimental Benchmark Comparison

To supplement the conceptual and scenario-based evaluations, a structured benchmark comparison was conducted between the Human Nutrition Consumption System and MyFitnessPal, a widely used commercial nutrition-tracking application. Both platforms were tested directly using the same user case, example products, and sequence of actions, including scanning items, logging meals, and reviewing insights. The objective was not to assess large-scale performance, but rather to evaluate how each system manages identical tasks and to determine whether the prototype reduces fragmentation and friction relative to an established industry standard.

The comparison evaluated twelve categories relevant to reducing fragmentation and friction, enhancing the visibility of nutritional behaviors, and improving integration. These categories included task friction, manual entry requirements, data consistency, integration across the consumption cycle, integration gaps, feedback speed, visibility of behavior patterns, insight coverage, pantry-driven planning, automation in food acquisition, user effort for logging, personalization, and traceability. The results are presented in Table 5.4.

Table 5.4 Benchmark Comparison Chart

Category	MyFitnessPal	Human Nutrition Consumption System
Task Friction (Steps Required)	**Moderate:** Logging a meal involves tapping “+”, searching or scanning an item, selecting a serving and a meal, then saving. Saved meals and recent items reduce steps for later.	**Low:** Logging a meal involves “+ Log meal” from saved recipes previously cooked or a custom entry
Dependency on Manual Entry	**High but mitigated:** Users rely heavily on manual search (premium paid versions allow barcode scanning)	**Low:** Users can scan the barcode and manually enter directly from the pantry module
Data Consistency Across Features	**Strong**: Foods logged are reflected in calorie and macro totals on the daily dashboard.	**Strong:** Item logs travel from scanning to the pantry and transform into meal logs, which update the dashboard.
Integration Across the Consumption Cycle	**Partial**: Covers logging meals, recipes, and macros only, and some monitoring, with newer features for premium.	**Full:** Covers logging products, recipes, meals, and analytics.

Integration Gaps Between Stages	**Gaps**: Only recipes flow into meal logging.	**Some gaps:** Covers the full cycle from acquisition to monitoring.
Feedback Delay / Update Latency	**Low**: Logs and views update almost immediately after logging, but sometimes they need to be refreshed or reopened.	**Medium**: Signing, scanning, recipe and meal viewing, and updates take between 2 and 4 seconds.
Visibility of Behavior Patterns	**Good:** Daily and weekly calorie/macro charts, weight and calorie trends, and consistency streaks.	**Good:** Provide daily/weekly/monthly charts, trends, and streaks
Coverage of Nutritional Insight	**Broad**: Shows macro distribution, calorie progress, trends, and personalized suggestions, with more options for premium tiers.	**Broad**: Shows macro distribution, calorie tracking, and progress.
Pantry-Driven Meal Planning	**Emerging**: Recently added meal planner with grocery lists.	**Implemented**: Filtering, searching, and categorization of the pantry.
Automation in Food Acquisition	**Medium**: Supports barcode scanning, meal scan, and recommended recipes, but does not automate purchasing.	**Low**: Barcode scanning, automated pre-filling, and recipe generation.
User Burden for Logging	**High**: Users must search, choose entries, and adjust servings.	**Medium**: Barcode scanning for entries and a manual entry option
User Preferences and Restrictions	**Strong**: Supports personalized calorie and macro goals, and allergy/preferences.	**Strong**: Personalized calorie and macro goals, allergies, preferences.
Traceability Across Modules	**Moderate**: Recipes and meals link back to ingredients and insights, but there is no full flow from acquisition through pantry to waste.	**Full**: Ingredients are linked to recipes and meals, and the update monitoring module.

The results indicate that the Human Nutrition Consumption System performs favorably compared to MyFitnessPal, particularly in lifecycle integration, structural coherence, and traceability. While MyFitnessPal demonstrates advantages in speed, data volume, and long-term analytics, the prototype distinguishes itself through comprehensive coverage of the consumption cycle, reduced task friction, automated acquisition, and robust data connections linking scanning, pantry management, recipes, and meal logs. Pantry-driven planning further enables users to make more proactive and sustainable choices. Notable limitations of the prototype include moderate latency, a smaller food database, fewer advanced insights, absence of automated grocery generation, and lack of large-scale usability testing. The primary objective is not commercial competition, but rather to demonstrate the value of an integrated, enterprise resource planning (ERP)-inspired approach to nutrition management. The findings suggest that this system provides superior lifecycle integration compared to mainstream applications, reduces friction in frequent tasks, and establishes a strong foundation for future development and research. Collectively, these results support the conclusion that the Integrated Human Nutrition model represents a promising and well-substantiated alternative to existing nutrition technologies.

CHAPTER 6: DISCUSSION

This discussion interprets the evaluation findings in relation to the five evaluation objectives (EO1–EO5) and the broader research aims established in Chapters 1 and 3. Collectively, the results demonstrate that the prototype operates as intended and offers substantive evidence for the feasibility and value of a structured, ERP-inspired approach to nutrition management. The integration of conceptual evaluation, scenario-based assessment, and benchmark comparison further shows that the system effectively operationalizes key components of the Integrated Human Nutrition framework in a technically consistent manner, indicating strong potential for real-world applications.

- **EO1 – Correct implementation of the consumption-cycle framework.** Findings from the conceptual, scenario-based, and benchmark evaluations provide strong support for EO1. The prototype effectively operationalizes the five stages of the consumption cycle: acquisition, storage, preparation, consumption, and monitoring; through coherent data flows, well-defined module interactions, and explicit Input–Process–Output structures. Cross-module traceability is maintained throughout the system: pantry items created by scanning are directly integrated into recipe generation, which then informs meal logs and subsequent analytics. This alignment confirms that the conceptual Integrated Human Nutrition model is both theoretically robust and technically implementable within a web-based system.
- **EO2 – Usefulness in reducing fragmentation and friction.** The evaluations consistently demonstrate that the prototype substantially reduces fragmentation across nutritional tasks that are typically distributed among multiple applications. By integrating scanning, inventory management, recipe generation, meal logging, and progress monitoring within a single platform, the system minimizes redundant processes and significantly decreases user workload. Scenario-based walkthroughs

confirmed seamless transitions between stages, eliminating the need for multiple tools or repeated data entry. The benchmark comparison with MyFitnessPal further substantiates this finding. The prototype demonstrates lower task friction for common actions such as meal logging and food acquisition, primarily due to its lifecycle-driven structure and pantry-based automation. These results support the central thesis that logging burden and tool fragmentation are significant contributors to low adherence in nutrition applications.

- **EO3 – Enhanced visibility of nutritional behaviors.** The dashboard and Progress & Insights modules deliver actionable visibility into calorie intake, macronutrient balance, weekly patterns, pantry utilization, and consistency streaks. During scenario execution, insights updated reliably, demonstrated internal accuracy, and accurately reflected logged behavior without data inconsistencies. Benchmark comparison also revealed that the prototype provides superior visibility compared to MyFitnessPal, particularly in modular traceability and pantry-driven insights. These dimensions are typically treated as separate features in commercial applications. Overall, the system demonstrates that integrated information flows substantially enhance users' ability to understand dietary behavior, even in the absence of full integration of production-cycle or activity-level data.
- **EO4 – Functional correctness and internal consistency.** Scenario-based evaluation confirms that the prototype behaves correctly across all core stages of the consumption cycle with their respective function (scanning, pantry, recipe, meal logs, analytics). These results validate that the prototype extends beyond conceptual representation and performs the concrete operational tasks necessary to support users throughout the entire consumption cycle.

- **EO5 – Extensibility and alignment with design requirements.** Our evaluation shows that the system satisfies the functional and non-functional requirements outlined in Chapter 4, particularly those related to modular architecture, clear data lineage, and lifecycle integration. The architectural design, ERD, and well-defined module boundaries collectively support straightforward extension into the remaining cycles of the Integrated Human Nutrition framework, including the food production subsystem, and the activity-level subsystem. The benchmark comparison further highlights extensibility as a distinguishing strength: while commercial nutrition applications may excel in usability or analytics, they rarely provide structured lifecycle integration or ERP-like traceability across modules. Because the prototype maintains clean separations of concern and relies on unified relational data structures, additional subsystems can be incorporated without reworking the core design. This demonstrates the system's scalability and fulfills the Design Science expectation that artifacts be capable of extension beyond their initial implementation scope.

6.1 Relation to research objectives.

The evaluation demonstrates that the project fulfills the primary goals of Design Science Research (DSR): it develops a practical conceptual model, constructs a functional artifact, and verifies its correctness within the defined scope. The prototype addresses the problem of fragmented nutrition tools by operationalizing the Integrated Human Nutrition framework as a practical consumption-cycle system. Scenario-based and benchmark testing confirm that the artifact performs as intended, maintains consistent data flow between modules, and adheres to the lifecycle logic established in the conceptual model.

The project aligns with essential DSR principles by producing a design artifact, addressing a significant problem, and upholding research rigor. The prototype implements the ERP-inspired framework as a web application, demonstrating the feasibility of shared master data, traceability, and structured multi-stage processes in consumer nutrition contexts. Evaluation results indicate reduced fragmentation and enhanced visibility of dietary behaviors, addressing persistent challenges in nutrition informatics. The design and evaluation processes are grounded in established theories, including ERP lifecycle structures, systems thinking, and documented issues in food data and digital health research.

This work further adheres to DSR principles by contributing new knowledge, employing an iterative development approach, and ensuring transparent communication. It presents a lifecycle-based model for organizing nutrition management and demonstrates the feasibility of implementing this model through modular architecture and integrated workflows. The development process involved iterative refinements to modules, data structures, and workflows based on testing outcomes and system constraints. The thesis provides a clear rationale for design decisions, outlines evaluation procedures, and discusses limitations, thereby meeting DSR standards for transparency and establishing a foundation for future research on production and activity-level cycles.

6.2 Limitations of Evaluation

While the evaluation offers meaningful evidence that the prototype is functionally correct, conceptually aligned, and feasible within the Integrated Human Nutrition framework, I recognize several important limitations that shaped both the process and the findings. These constraints emerged naturally from the realities of early-stage artifact development and the exploratory nature of Design Science Research. By reflecting these boundaries, I aim to provide a candid context for interpreting the evaluation and to highlight clear priorities for future work.

First, the scope of my evaluation methods was necessarily restricted. Due to time constraints and the decision to focus on system construction, I was unable to conduct formal usability testing with real users. Instead, I relied on conceptual validation and scenario-based functional assessments. These approaches helped me verify that the artifact performed as intended in controlled situations, but they did not allow for direct observation of user experiences, such as perceived usability, onboarding challenges, cognitive workload, or patterns of long-term engagement. As a result, while the prototype demonstrated coherence and technical feasibility, its true effectiveness in real-world settings remains to be established through structured usability studies and broader empirical research.

Second, I acknowledge that the prototype's scope is limited to the consumption cycle. The system does not yet incorporate the production and activity-level cycles that are integral to the broader Integrated Human Nutrition framework. Features like nutrient-variability modeling, production traceability, and wearable-based energy-expenditure integration were outside the scope of this artifact and, consequently, were not evaluated. This decision was intentional to ensure a focused and feasible project (as outlined in Chapter 1), but it does limit the extent to which the system's full lifecycle coherence can be assessed at this stage.

Third, my work depended on several external services, including barcode APIs, nutrition databases, and recipe providers, which introduced technical dependencies beyond my direct control. These external systems can sometimes introduce inaccuracies, latency, or missing data, all of which may affect the prototype's performance. While I included fallback mechanisms, such as manual entry, to reduce reliance on external calls, the evaluation could not fully disentangle the prototype's behavior from these dependencies. In real-world deployment, variability in third-party data quality and API reliability would demand more robust mitigation strategies.

Fourth, the evaluation was conducted over a short period and focused on scenario-based testing. This approach confirmed that the modules behaved correctly during immediate interactions, but it could not capture insights about long-term user adoption, dietary improvements, retention, or changes in nutritional literacy. Such outcomes would require longitudinal or field studies, which were outside the scope of this thesis. Therefore, while I was able to demonstrate technical correctness, conceptual alignment, and benchmarking to current solutions, I cannot yet speak to the system's behavioral or clinical impact.

Finally, conceptual evaluation inevitably involves a degree of interpretive judgement when assessing how closely the implemented system corresponds to the Integrated Human Nutrition framework. Although I used scenario-based checks and explicit Input–Process–Output structures to minimize bias, some subjectivity is inherent when determining conceptual coherence, flow completeness, or the positioning of this framework within the industry. In summary, this evaluation demonstrates that the prototype is functionally correct, conceptually aligned, and feasible within the consumption cycle. However, the findings should be interpreted in light of these acknowledged limitations, particularly the absence of user-based usability testing, the restricted scope, reliance on external data sources, and the short-term nature of the evaluation. These reflections directly inform my recommendations for refinement, expanded evaluation, and future development, as discussed in Chapter 7.

CHAPTER 7: CONCLUSION

7.1 Summary of Research Findings

This thesis addresses a significant limitation in contemporary nutrition technologies: the absence of an integrated, lifecycle-oriented system that unifies food acquisition, storage, preparation, consumption, and monitoring. Most existing applications address these stages in isolation, emphasizing calorie logging, recipe discovery, or habit tracking, which results in fragmented workflows, inconsistent data, and limited behavioral insight. By drawing on enterprise resource planning (ERP) theory and systems thinking, this research conceptualizes human nutrition as a sequence of interdependent processes that benefit from shared data structures, coordinated workflows, and real-time feedback, analogous to organizational operations.

Through the application of Design Science Research methodology, this study produced two principal contributions. First, it introduced the Integrated Human Nutrition framework, which organizes the nutrition system into three interconnected cycles: production, consumption, and activity, each defined by explicit Input–Process–Output stages. Second, the framework was instantiated through the Human Nutrition Consumption System, a working prototype that operationalizes food acquisition, pantry management, recipe generation, meal logging, and progress monitoring within a coherent, ERP-inspired architecture. These contributions directly address gaps identified in the literature, including systemic data fragmentation, lack of traceability, and limited transparency, which are recognized as barriers to both healthy behavior and the effectiveness of digital tools.

Evaluation findings confirm that the conceptual model is both technically feasible and practically implementable. Conceptual alignment checks verified that the prototype accurately represents the lifecycle logic of the consumption cycle and fulfills the functional requirements outlined in Chapter 4. Scenario-based assessments demonstrated consistent system performance across stages, maintenance of data integrity, and smooth transitions from acquisition to monitoring. Benchmark comparisons with MyFitnessPal provided further insight, indicating that the prototype reduces task friction in key workflows, enhances traceability across modules, and achieves more comprehensive lifecycle integration, while also identifying limitations related to update latency, data coverage, and long-term analytics.

Although only the consumption cycle was implemented, the architecture, shared master-data model, and module boundaries provide substantial evidence that this approach can be extended to the production and activity-level cycles outlined in the broader framework. Collectively, these findings reinforce the central claim that nutrition management can benefit significantly from ERP-style integration, lifecycle-aware design, and unified digital infrastructures. The prototype demonstrates the viability of this approach and establishes a foundation for future expansion, user-based evaluation, and real-world deployment.

7.2 Contributions

This research makes several contributions to information systems, nutrition informatics, and the design of public-health technologies.

First, it formalizes the Integrated Human Nutrition framework, conceptualizing nutritional behavior not as a set of isolated actions but as coordinated processes across three interrelated cycles. By structuring each cycle through explicit Input–Process–Output stages, the framework offers a systematic way to model how data, actions, and decisions propagate through personal and environmental food systems.

Second, the work advances a reinterpretation of the food and nutrition system using principles historically applied to large-scale organizations. The adaptation of enterprise resource planning concepts demonstrates that these ideas can enhance individual- and household-level nutrition practices as well as organizational ones.

Third, the study introduces an integrated, lifecycle-aware approach to nutrition-technology design. The implementation of a pantry-driven workflow that connects acquisition, storage, preparation, consumption, and monitoring serves as a design pattern for future nutrition-informatics systems, illustrating how modular subsystems can interact coherently to support user decision-making through structured data flows. Collectively, these contributions expand the conceptual vocabulary available for designing next-generation nutrition technologies and frame nutrition management as a systems-oriented challenge rather than solely a behavioral one.

7.3 Practical Impact

The thesis also offers several practical contributions relevant to developers, nutrition professionals, and health-technology practitioners.

First, the prototype demonstrates a low-friction approach to food logging, addressing a significant barrier to user adherence identified in prior work on diet-tracking applications. By integrating barcode and image-based scanning, automated pantry management, and one-click meal logging, the system simplifies daily tasks that are often experienced as burdensome.

Second, the pantry-centered architecture operationalizes a structured food-lifecycle workflow, enabling users to manage existing inventory, plan feasible meals, and track what they consume within a unified environment. This configuration extends beyond most existing applications, which generally support only a subset of these functions and thereby reinforce fragmented behaviors.

Third, the system generates clear, actionable analytics that help users understand their nutritional behaviors, identify patterns over time, and build consistency through feedback mechanisms such as trends and streaks. These features may support nutritionists and healthcare professionals who require tools that reinforce long-term dietary improvements rather than short-term tracking alone.

Finally, the Integrated Human Nutrition framework has practical implications for public-health technologies, offering a conceptual and architectural model that could ultimately interoperate with traceability systems, government nutrition programs, or clinical nutrition workflows, thereby linking individual-level tools to broader food-system initiatives.

7.4 Limitations of Research

While the research achieved its core objectives, several limitations should be acknowledged to contextualize the findings.

Scope limitation: Only the consumption cycle was implemented as a working system; the production and activity-level cycles remain conceptual and were not evaluated empirically. As a result, the complete Integrated Human Nutrition model is not yet instantiated, and conclusions about cross-cycle integration remain prospective.

Prototype constraints: The prototype depends on external services for barcode recognition, nutrient lookup, and recipe retrieval, and on optical-character-recognition performance that varies with image quality and environmental conditions. These dependencies introduce reliability concerns that lie partly outside the system's control and would require mitigation strategies in any production-grade deployment.

Short evaluation horizon: The study assessed short-term functional usability and correctness rather than long-term performance, behavior change, or health outcomes, which are central to nutrition management but require longitudinal evaluation designs. Consequently, the thesis does not make claims about sustained impact, only about initial feasibility.

7.5 Future Work and Research Directions

This thesis shows that the Integrated Human Nutrition framework works in practice and that its ERP-inspired design helps connect different nutrition tasks, improve traceability, and reduce fragmentation. However, the evaluation also points out areas for further development and more thorough testing. These areas include expanding the framework, improving system features, and increasing external evaluation.

One way to build on this work is to expand the framework beyond just the consumption cycle. For example, adding data from wearable devices like heart-rate monitors, step counters, GPS activity trackers, and sleep trackers could help measure energy intake and use more accurately. At the same time, including the production cycle would allow users to see where ingredients come from, how nutrient levels vary between batches, and other supply-chain details. These changes would align with current trends in nutrition informatics and demonstrate full integration across all cycles, as ERP principles suggest.

Another area for growth is expanding the system itself. The prototype could be updated to support households or multiple users, enabling shared pantries, group meal planning, and different access levels. Adding features such as grocery budgeting, cost-based meal planning, more cultural or regional recipe options, stricter allergy and preference filters, and automatic grocery lists would make the system more useful to more people. Connecting with local or regional nutrition databases would also improve the system's accuracy, especially for groups whose foods are underrepresented in global data.

Future developers and researchers could also improve how the system is tested and validated. The comparison in Chapter 5 shows the prototype’s strengths, such as its coverage of the full lifecycle, traceability, and consistent data across modules, but also points out issues, such as slow response times and a smaller food database. These results show the need for better evaluation methods. One plan is to create an Expert Evaluation Rubric in which faculty or specialists review the system using clear criteria for how well it matches the concept, how easy the interface is to use, how data flows, and how closely it follows the ERP-inspired model. This expert review gives an independent, theory-based opinion on whether the system truly applies the Integrated Human Nutrition framework.

Another future activity is to run a Usability Testing Study to measure task completion times, error rates, user understanding, and the system's ease of use, using tools such as the System Usability Scale (SUS). This testing is key to ensuring that the system makes things easier and improves the user experience compared to other tools. Below, we present two possible rubrics: the usability evaluation (Table 7.5.1) and the expert evaluation (Table 7.5.2). Such rubrics can help developers and researchers evaluate the usability of the Human Nutrition System and assess whether lifecycle integration brings real, measurable benefits.

7.5.1 Possible Usability Evaluation Rubric

Category	What It Measures	Why It Matters	Example
Task Completion Time	How long does it take to complete tasks (scan item, log meal, generate recipe)?	Indicates efficiency and friction reduction.	Seconds per task
Task Success Rate	Whether users can complete tasks without help.	Measures basic usability and clarity of UI flows.	% tasks completed correctly
Error Rate	Incorrect actions or failed attempts.	Reflects cognitive load and interaction problems.	Errors per task
Learnability	How quickly users improve after repeated trials.	Shows whether the system becomes easier to use over time.	Drop in task time after repetition
System Usability Scale (SUS)	Users' perceived usability.	Industry standard for comparing systems.	0–100 (SUS score)
User Satisfaction	How satisfied users are with features and workflows.	Measures acceptance likelihood.	Likert-scale responses

7.5.2 Possible Expert Evaluation Rubric

Category	What It Measures	Score
Conceptual Alignment	The degree to which the prototype reflects the Integrated Human Nutrition framework and lifecycle logic.	1–5
ERP Principles Applied	Use of shared master data, lifecycle integration, traceability, integration latency, and closed-loop feedback.	1–5
Module Interaction Coherence	Whether module outputs correctly feed into subsequent stages (scanner → pantry → recipes → meal logs → analytics).	1–5
Data Integrity & Lineage	Accuracy and consistency of data transitions across modules.	1–5
Interface Clarity & Usability	Visual design, navigation simplicity, and cognitive load.	1–5
Technical Implementation Quality	Code structure, modularity, API design, and database integrity.	1–5
Scalability & Extensibility	Ability to integrate production and activity cycles in future versions.	1–5
Feasibility for Real-World Use	Practicality for deployment, reliability, and external API dependency management.	1–5
Overall Assessment	Final judgment of design quality and contribution.	1–5

7.6 Final remarks

This thesis has argued that addressing fragmented nutrition management requires a systemic, lifecycle-oriented approach rather than a collection of isolated technological features. By introducing the Integrated Human Nutrition framework and implementing a consumption-cycle prototype, the research has shown that individual and household nutrition can be organized through structured processes analogous to those used in enterprise resource planning systems. Evaluation results indicate that the prototype is conceptually sound, practically usable, and capable of supporting more proactive, inventory-aware dietary management within its current scope.

While the work presented here focuses on a single subsystem, it demonstrates that integrating acquisition, storage, preparation, consumption, and monitoring into a coherent architecture is both feasible and beneficial. With continued development of the production and activity-level cycles and with further empirical study in diverse contexts, the Integrated Human Nutrition approach has the potential to reshape how individuals, households, and institutions understand and manage nutritional behavior, linking personal decision-making more closely to broader food-system dynamics and public-health objectives.

REFERENCES

Carter, M. C., Burley, V. J., Nykjaer, C., & Cade, J. E. (2013). Adherence to a smartphone application for weight loss compared to website and paper diary: Pilot randomized controlled trial. Journal of Medical Internet Research, 15(4), e32. https://doi.org/10.2196/jmir.2283

Emmerich, S. D., Fryar, C. D., Stierman, B., & Ogden, C. L. (2024). Obesity and severe obesity prevalence in adults: United States, August 2021–August 2023 (NCHS Data Brief No. 508). National Center for Health Statistics. https://www.cdc.gov/nchs/products/databriefs/db508.htm

Reider, C. A., Chung, R.-Y., Devarshi, P. P., Grant, R. W., & Hazels Mitmesser, S. (2020). Inadequacy of immune health nutrients: Intakes in US adults, the 2005–2016 NHANES. Nutrients, 12(6), 1735. https://doi.org/10.3390/nu12061735

Zečević, M., Mijatović, D., Kos Koklič, M., Žabkar, V., & Gidaković, P. (2021). User perspectives of diet-tracking apps: Reviews content analysis and topic modeling. Journal of Medical Internet Research, 23(4), e25160. https://doi.org/10.2196/25160

Sobal, J., Khan, L. K., & Bisogni, C. (1998). A conceptual model of the food and nutrition system. Social Science & Medicine, 47(7), 853-863. https://www.sciencedirect.com/science/article/pii/S027795369800104X

Sablah, M. (2019). Causes and impacts of undernutrition over the life course [Conference presentation]. United Nations Department of Economic and Social Affairs Expert Group Meeting on Population, Food Security, Nutrition and Sustainable Development, New York, NY, United States. https://www.un.org/en/development/desa/population/events/pdf/expert/30/presentations/Tuesday/Session4/Causes%20-%20Consequences%20of%20Undernutrition%20ICPD%20-%20UNICEF.pdf

König, L. M., Attig, C., Franke, T., & Renner, B. (2021). Barriers to and facilitators for using nutrition apps: Systematic review and conceptual framework. JMIR mHealth and uHealth, 9(6), e20037. https://mhealth.jmir.org/2021/6/e20037/

Market.us. (2025, March 14). Diet and nutrition apps statistics and facts (2025) [Report]. https://media.market.us/diet-and-nutrition-apps-statistics/

Jenneson, V. L., Pontin, F., Greenwood, D. C., Clarke, G. P., & Morris, M. A. (2022). A systematic review of supermarket automated electronic sales data for population dietary surveillance. Nutrition Reviews, 80(6), 1711–1722. https://academic.oup.com/nutritionreviews/article/80/6/1711/6425025

University of Sydney. (2024, August 29). AI food tracking apps need improvement to address accuracy, cultural diversity [Press release]. https://www.sydney.edu.au/news-opinion/news/2024/08/29/ai-food-tracking-apps-need-improvement-to-address-cultural-diversity.html

World Health Organization. (n.d.). Malnutrition [Fact sheet]. Retrieved December 5, 2024, from https://www.who.int/news-room/fact-sheets/detail/malnutrition

Rickman, J. C., Barrett, D. M., & Bruhn, C. M. (2007). Nutritional comparison of fresh, frozen and canned fruits and vegetables. Part 1. Vitamins C and B and phenolic compounds. Journal of the Science of Food and Agriculture, 87(6), 930–944. https://doi.org/10.1002/jsfa.2825

Rickman, J. C., Bruhn, C. M., & Barrett, D. M. (2007). Nutritional comparison of fresh, frozen, and canned fruits and vegetables II. Vitamin A and carotenoids, vitamin E, minerals and fiber. Journal of the Science of Food and Agriculture, 87(7), 1185–1196. https://doi.org/10.1002/jsfa.2824

UNICEF. (2020). Conceptual framework on the determinants of maternal and child nutrition. United Nations Children's Fund. https://www.unicef.org/media/113291/file/UNICEF%20Conceptual%20Framework.pdf

Breda, J., Castro, L. S. N., Whiting, S., Williams, J., Jewell, J., Engesveen, K., & Wickramasinghe, K. (2020). Towards better nutrition in Europe: Evaluating progress and defining future directions. Food Policy, 96, 101887. https://doi.org/10.1016/j.foodpol.2020.101887

Lieffers, J. R. L., & Hanning, R. M. (2012). Dietary assessment and self-monitoring: With nutrition applications for mobile devices. Canadian Journal of Dietetic Practice and Research, 73(3), e253-e260. https://doi.org/10.3148/73.3.2012.e253

Franco, R. Z., Fallaize, R., Lovegrove, J. A., & Hwang, F. (2019). Food identification by barcode scanning in the Netherlands: A quality assessment of labelled food product databases underlying popular nutrition applications. Public Health Nutrition, 22(10), 1780-1789. https://doi.org/10.1017/S1368980019000120

Urban, L. E., McCrory, M. A., Dallal, G. E., Das, S. K., Saltzman, E., Weber, J. L., & Roberts, S. B. (2011). Accuracy of stated energy contents of restaurant foods. JAMA, 306(3), 287-293. https://doi.org/10.1001/jama.2011.993

Measurement Toolkit. (n.d.). Food composition data and databases. Intake Center for Dietary Assessment. https://www.measurement-toolkit.org/diet/introduction/food-composition

Van Berkum, S., Dengerink, J., & Ruben, R. (2018). The food systems approach: Sustainable solutions for a sufficient supply of healthy food. Wageningen Economic Research. https://doi.org/10.18174/451505

Fanzo, J., Haddad, L., McLaren, R., Marshall, Q., Davis, C., Herforth, A., Jones, A., Beal, T., Tschirley, D., Bellows, A., & Miachon, L. (2020). The food system. In Sustainable food systems and nutrition in the 21st century: A report from the 22nd annual Harvard Nutrition Obesity Symposium. American Journal of Clinical Nutrition, 113(Suppl_1), 7S-33S. https://doi.org/10.1093/ajcn/nqaa318

Chen, J., Cade, J. E., & Allman-Farinelli, M. (2015). The most popular smartphone apps for weight loss: A quality assessment. JMIR mHealth and uHealth, 3(4), e104. https://doi.org/10.2196/mhealth.4334

Stanley, A. (2018, October 8). Ready to rumble: IBM launches Food Trust blockchain for commercial use. Forbes. https://www.forbes.com/sites/astanley/2018/10/08/ready-to-rumble-ibm-launches-food-trust-blockchain-for-commercial-use/

USDA, Agricultural Research Service. (n.d.). FoodData Central API guide. Retrieved December 5, 2024, from https://fdc.nal.usda.gov/api-guide

Dignan, L. (2018, October 8). IBM Food Trust blockchain network available, Carrefour joins retailer roster. ZDNet. https://www.zdnet.com/finance/blockchain/ibm-food-trust-blockchain-network-available-carrefour-joins-retailer-roster/

Aung, M. M., & Chang, Y. S. (2014). Traceability in a food supply chain: Safety and quality perspectives. Food Control, 39, 172–184. https://doi.org/10.1016/j.foodcont.2013.11.007

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. MIS Quarterly, 28(1), 75–106. https://doi.org/10.2307/25148625

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. Journal of Management Information Systems, 24(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Piwek, L., Ellis, D. A., Andrews, S., & Joinson, A. (2016). The rise of consumer health wearables: Promises and barriers. PLOS Medicine, 13(2), e1001953. https://doi.org/10.1371/journal.pmed.1001953

Fukagawa, N. K., McKillop, K., Pehrsson, P. R., Moshfegh, A., Harnly, J., & Finley, J. (2022). USDA's FoodData Central: What is it and why is it needed today? The American Journal of Clinical Nutrition, 115(3), 619–624. https://doi.org/10.1093/ajcn/nqab397

Davenport, T. H. (1998). Putting the enterprise into the enterprise system. Harvard Business Review, 76(4), 121–131. https://hbr.org/1998/07/putting-the-enterprise-into-the-enterprise-system

Hackman, J. R., & Morris, C. G. (1975). Group tasks, group interaction process, and group performance effectiveness: A review and proposed integration. Advances in Experimental Social Psychology, 8, 45–99. https://doi.org/10.1016/S0065-2601(08)60248-8

Meadows, D. H. (2008). Thinking in systems: A primer (D. Wright, Ed.). Chelsea Green Publishing. https://www.chelseagreen.com/product/thinking-in-systems/

Monk, E., & Wagner, B. (2012). Concepts in enterprise resource planning (4th ed.). Cengage Learning. https://www.cengage.com/c/concepts-in-enterprise-resource-planning-4e-monk-wagner/9781111820398/

Shcherbina, A., Mattsson, C. M., Waggott, D., Salisbury, H., Christle, J. W., Hastie, T., Wheeler, M. T., & Ashley, E. A. (2017). Accuracy in wrist-worn, sensor-based measurements of heart rate and energy expenditure in a diverse cohort. Journal of Personalized Medicine, 7(2), 3. https://doi.org/10.3390/jpm7020003

Topol, E. J. (2019). Deep medicine: How artificial intelligence can make healthcare human again. Basic Books. https://www.hachettebookgroup.com/titles/eric-topol/deep-medicine/9781541644632/

Dennison, L., Morrison, L., Conway, G., & Yardley, L. (2013). Opportunities and challenges for smartphone applications in supporting health behavior change: Qualitative study. Journal of Medical Internet Research, 15(4), e86. https://doi.org/10.2196/jmir.2583

Monteros, M. (2023, June 23). How AI meal recommendations made Hungryroot a $250 million business. Modern Retail. https://www.modernretail.co/technology/how-ai-meal-recommendations-made-hungryroot-a-250-million-business/

Abeltino, A., Riente, A., Bianchetti, G., Serantoni, C., De Spirito, M., Capezzone, S., Esposito, R., & Maulucci, G. (2024). Digital applications for diet monitoring, planning, and precision nutrition for citizens and professionals: A state of the art. Nutrition Reviews, 83(2), e574–e601.https://doi.org/10.1093/nutrit/nuae035

van der Haar, S., Raaijmakers, I., Verain, M. C. D., & Meijboom, S. (2023). Incorporating consumers' needs in nutrition apps to promote and maintain use: Mixed methods study. JMIR mHealth and uHealth, 11, e39515. https://mhealth.jmir.org/2023/1/e39515

Mozaffarian, D. (2016). Dietary and policy priorities for cardiovascular disease, diabetes, and obesity: A comprehensive review. Circulation, 133(2), 187-225. https://doi.org/10.1161/CIRCULATIONAHA.115.018585

APPENDICES

Table A1: Food Production Cycle

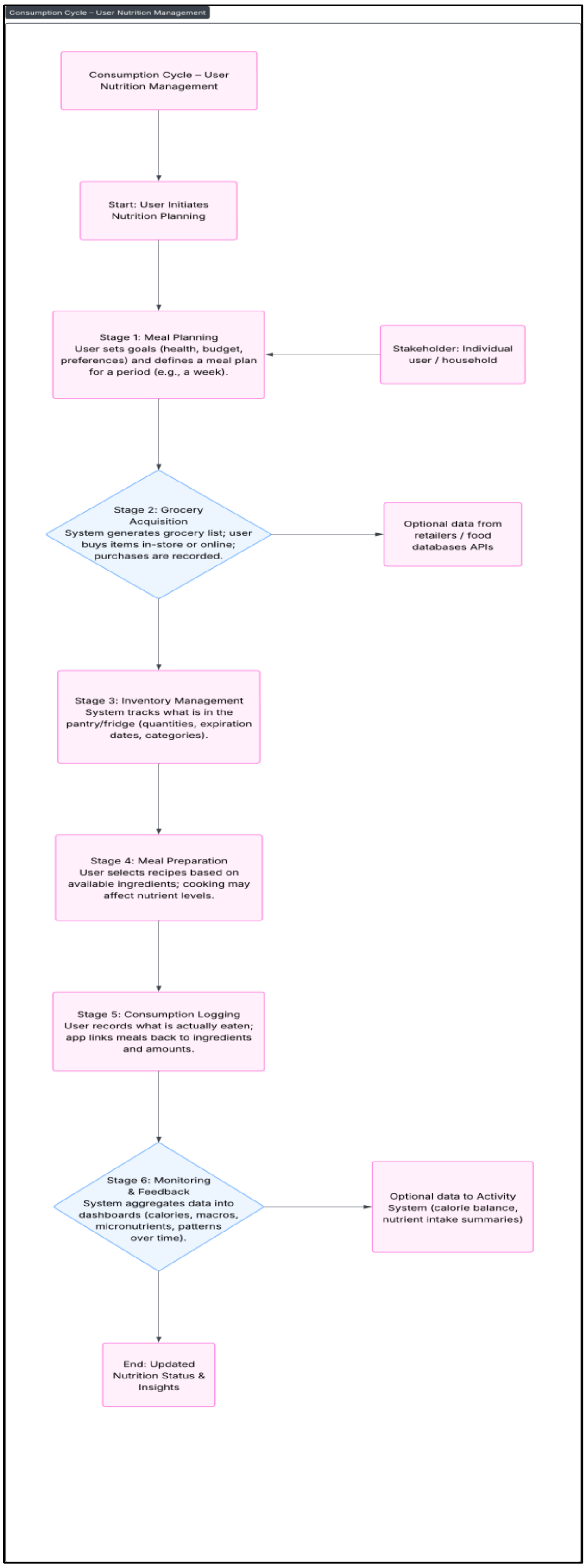


Table A2: Food Consumption Cycle

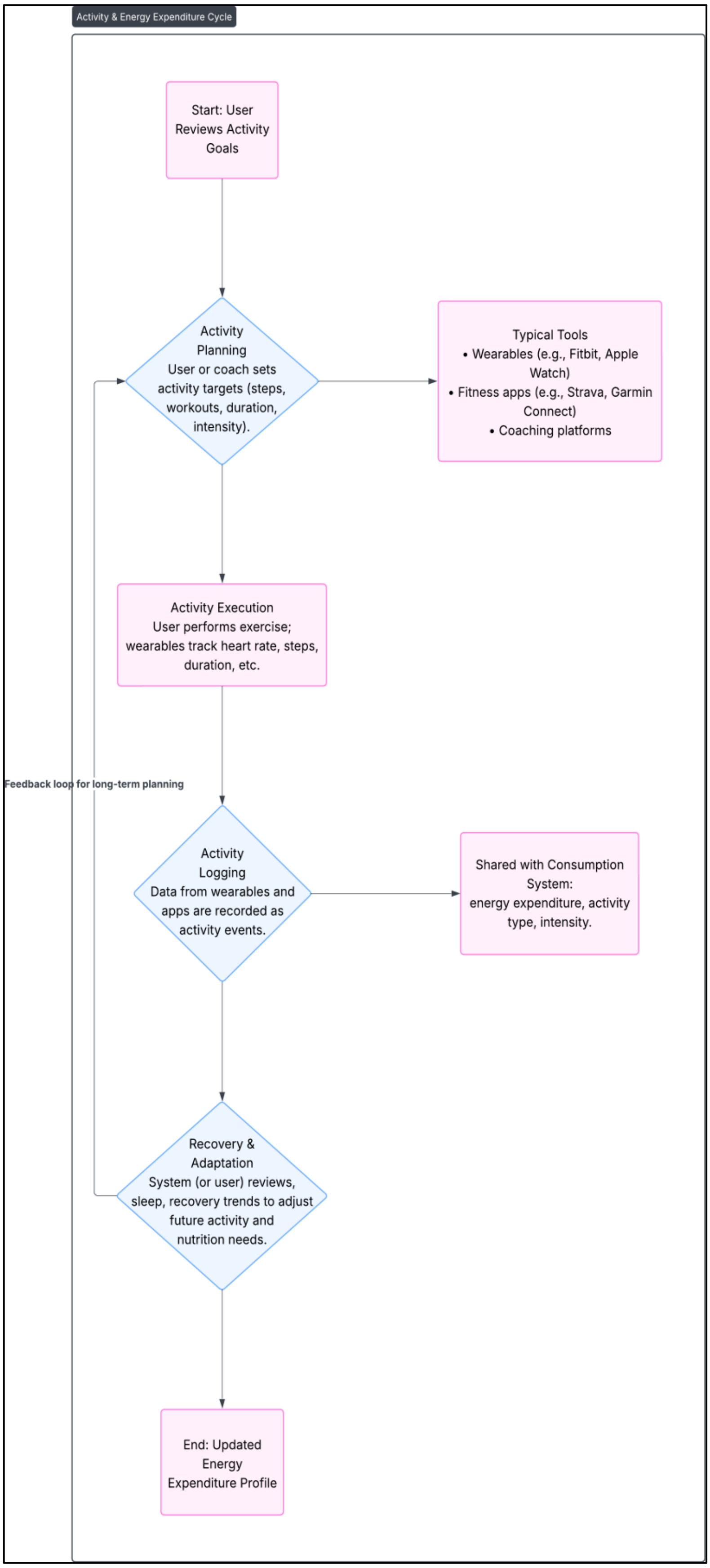


Table A3: Activity-Level Cycle