Bil495 - Innovative Computer Application
NutriGame: AI-Based Nutrition Tracking and Recommendation System with Chatbot, Social Sharing, and Gamification

Submitted By

Betül Aydeğer 211101006
Duygu Tümer 211101048
Zeynep Yavuz 211101047
Nazlı Ceyda Ünsoy 211401007

Department of Computer Engineering
TOBB University of Economics & Technology

Spring 2025

ABSTRACT

In today's fast-paced world, maintaining a balanced diet is a significant challenge due to busy schedules, limited meal planning time, and lack of sustained motivation. Many existing nutrition applications suffer from high user drop-off rates, largely caused by time-consuming manual data entry, limited personalization, and poor user engagement. This project introduces NutriGame, an intelligent and gamified mobile application designed to simplify nutrition tracking and promote long-term healthy eating habits.

NutriGame leverages AI-powered food recognition, allowing users to log meals by simply taking a photo, and provides personalized meal recommendations based on individual goals and preferences. The app incorporates gamification elements such as points, achievements, and mini-games to maintain user motivation, alongside a supportive chatbot that offers daily tips in a friendly and encouraging tone. Social sharing features enable community engagement, while multilingual support ensures accessibility to a diverse user base.

The system is developed using Flutter for cross-platform compatibility on iOS and Android, with a FastAPI backend to ensure scalability and platform independence. By combining artificial intelligence, behavioral psychology, and user-centered design, NutriGame aims to improve user retention, reduce meal logging time, and contribute to both personal wellness and broader public health outcomes.

TABLE OF CONTENTS

1. INTRODUCTION

1.1. Project Purpose

In today's fast-paced world, maintaining a healthy diet has become increasingly challenging due to busy schedules, high stress levels, and limited time for meal planning and preparation. Research indicates that approximately 69% of users abandon health and nutrition apps within the first 90 days, and around 70% stop using them within 100 days [1]. The primary reasons identified for these high drop-off rates include time-consuming manual data entry, limited personalization, poor user experience, and lack of sustained motivation [1], [2].

While there are many nutrition apps available today, most of them have significant limitations. They typically rely on users to manually search for and enter every single item they eat, which can be time-consuming and inaccurate. Furthermore, these apps often lack the features that provide emotional and social support, which are critical for long-term motivation. They don't offer intelligent suggestions, adapt to the user's lifestyle, or create a sense of community and encouragement. This creates a clear market gap for a smarter, more engaging solution.

The primary purpose of this project is to develop **NutriGame**, an intelligent and user-friendly mobile application designed to address these challenges. Our motivation is to leverage the power of Artificial Intelligence (AI), social interaction, and gamification to make nutrition tracking simple, enjoyable, and sustainable. NutriGame is designed to be more than just a calorie counter; it aims to be a personal wellness companion that supports users on their journey to a healthier lifestyle.

The success of the project will be measured through clear and quantifiable indicators: achieving at least a 25% improvement in three-month user retention compared to benchmark nutrition apps, reducing the average meal logging time by 50% through AI-based automation, reaching a daily-to-weekly active user ratio above 40%, maintaining over 80% accuracy in AI-powered recognition for the top 100 most common foods in the target regions, and securing an average user satisfaction score of 4.2/5 or higher in pilot tests.

By combining technology, behavioral psychology, and community-driven motivation, NutriGame aims not only to improve individual health outcomes but also to contribute to broader public health by promoting long-term healthy eating behaviors.

1.2. Project Scope

NutriGame is a mobile application designed to make healthy eating easy, fun, and social. It acts as a personal wellness assistant that helps users track their daily nutrition, get smart food recommendations, and stay motivated on their health journey. By combining

Artificial Intelligence (AI) with gamification and social features, the app aims to turn the chore of calorie counting into an engaging and supportive experience, targeting at least a 25% improvement in 3-month user retention compared to benchmark nutrition apps

The core functionality of NutriGame will include the following features:

- Personalized Nutrition Planning: The app will create daily and weekly meal plans based on the user's goals (weight loss, gain, or maintenance), dietary preferences, and profile information.
- Intelligent Nutrition Tracking: Users can log their meals in three ways: 1. Manual Entry: Searching for food in the database. 2. Text-Based AI: Typing a description like "a bowl of oatmeal with berries" to get a calorie estimate. 3.Photo-Based AI: Uploading a photo of a meal for automatic food recognition and calorie estimation.
- AI-Powered Chatbot: NutriGame will include a smart chatbot that functions more like a personalized AI assistant. Unlike basic rule-based bots, this assistant will offer context-aware responses and adaptive suggestions based on user preferences and behavior. It can help users with nutrition-related questions, provide healthy recipe ideas based on available ingredients, and offer motivational tips tailored to the user's current goals or challenges. While it is still an automated system, its purpose is to simulate a helpful digital companion rather than just answering predefined questions. This assistant aims to support the user's health journey in a more natural, human-like way, making the experience both practical and emotionally supportive.
- Progress and Health Monitoring: The app will track key metrics, such as daily water intake, weight and body mass index (BMI) over time, mood and emotional well-being: Users can log their mood to see how their eating habits affect their feelings.
- Gamification and Motivation:
    - Points and Rewards: Users will earn points for completing healthy actions (e.g., logging a meal, drinking water).
    - Mini-Games: Points can be used to unlock fun, simple mini-games within the app.
    - Streak Tracking: The app will track the user's "streak" for consecutive days of meeting their goals, encouraging consistency.
- Social Sharing: Users will be able to share their achievements, favorite meals, or progress on social media to get encouragement from friends.
- History and Archive: The application will keep a detailed history of the user's logged meals, weight progress, and achievements, allowing them to review their journey over time.
- Shopping List Generation: Based on the weekly meal plan, the app will automatically generate a shopping list to make grocery shopping easier.

To keep the project realistic and manageable within the given timeframe, some advanced features will not be included in this version. NutriGame will not provide medical diagnoses or replace professional health advice. It will not connect to wearable devices like smartwatches, nor support direct food or grocery orders. Real-time coaching from

human experts will not be available, as the chatbot is AI-based. Additionally, the app will suggest recipes but will not include detailed cooking tutorials or videos.

If the core features of NutriGame are completed ahead of schedule, several advanced features may be explored as stretch goals to further enhance the app's functionality and user experience. One potential addition is a *barcode scanner*, which would allow users to quickly log packaged foods by scanning their barcodes and retrieving nutritional data automatically. Another feature could be *integration with wearable devices* such as Apple Health or Google Fit, enabling the app to import physical activity data and offer more accurate, personalized calorie recommendations. To build a stronger sense of community, we may also introduce an *in-app social feed* where users can share progress, post photos, and participate in group challenges. Finally, *the AI meal planner could be improved to adapt to user feedback*—learning preferences over time and tailoring suggestions based on individual likes, dislikes, and habits for a more personalized experience.

NutriGame's scope also specifies the intended user base, usage scenarios, and data boundaries to ensure clarity and compliance. The app is designed for individual users—such as students, busy professionals, and health-conscious individuals—who have basic smartphone literacy. It excludes children under 18 years old -for privacy, safety, and legal compliance reasons-, professional dietitians, healthcare institutions, and clinical practitioners, due to legal restrictions, privacy requirements, and differing functional needs.

Supported usage scenarios include daily nutrition tracking, meal planning, grocery list creation, and social sharing. Scenarios such as medical treatment planning, emergency health management, and multi-user family accounts are excluded, as they require more complex technical solutions and carry greater legal responsibilities.

Data collection will be limited to user-entered nutrition logs, information from the nutrition database, and basic health metrics such as weight, BMI, water intake, and mood. Sensitive medical data—such as medical history, genetic information, laboratory results, and insurance records—will not be collected, in order to protect user privacy and ensure legal compliance. External integrations will be limited to nutrition databases and built-in smartphone functions (e.g., camera, sharing tools), with no connection to third-party medical or insurance systems. This approach allows the project to focus on its primary goal—promoting healthy lifestyle habits—while maintaining a secure, manageable, and ethically responsible technical framework.

Scope Area	Included	Excluded
Features	- Personalized nutrition planning - AI-powered meal logging (manual, text, photo) - AI chatbot (recipes, motivation)	- Medical diagnosis / professional health advice - Wearable device integration - Direct grocery ordering - Human coaching

	- Progress tracking (water, weight, BMI, mood) - Gamification (points, rewards, mini-games, streaks) - Social sharing - History & archive - Shopping list generation	- Detailed cooking videos - Multi-user family accounts - Emergency health management
Users	- Individual users (students, professionals, health-conscious individuals)	- Children under 13 - Professional dietitians - Healthcare institutions - Clinical practitioners
Scenarios	- Daily nutrition tracking - Meal planning - Shopping list creation - Social sharing	- Medical treatment planning - Emergency health management - Multi-user scenarios
Data	- User-entered nutrition logs - Nutrition database information - Basic health metrics (weight, BMI, water intake, mood)	- Sensitive medical data (medical history, genetics, lab results, etc.) - Third-party medical/health system integrations

Table 1.1. Summary of the features

1.3. Definitions, Acronyms, and Abbreviations

Term	Definition
Semantic Versioning	Version numbering standard (MAJOR.MINOR.PATCH)
Static Analysis	Analyzing code without executing it
Regression Test	Testing to ensure that existing functionality is not broken by new changes
Data Scraping	Extracting data from websites

Label Encoding	Converting text labels into numerical values
Stop-word Removal	Removing insignificant words in natural language processing
Segmentation Mask	Labeled mask used to separate objects in an image
Bounding Box	Rectangular frame showing the position of an object in an image
Anonymization	Process of removing personal data from datasets
NutriGame	Mobile application for healthy living, meal tracking, and gamification
Meal Logging	Feature allowing users to record their meals in the system
AI Chatbot	AI-powered chatbot providing nutrition and diet advice
Gamification	System for increasing user motivation through points, badges, and goals
Streak	A series of consecutive days a user completes a certain task or goal. In NutriGame, it encourages consistency and habit-building.
BMI (Body Mass Index)	A number calculated from a person's weight and height. It is used to determine if someone is underweight, normal weight, overweight, or obese.

Data Processing	Processing nutrition, health, and activity data
Healthy Recipe Dataset	Dataset containing healthy food recipes
Psychology Dataset	Dataset containing psychological counseling conversations
Image Segmentation Dataset	Dataset with food images annotated with segmentation masks
Image Classification Dataset	Dataset categorizing food images into different classes
Traceability Matrix	Table mapping requirements to test scenarios
P0 (Critical)	Blocks core functionality, must be fixed immediately before release.
P1 (High)	Major functionality issue, fix required in the next release cycle.
P2 (Medium)	Minor functionality issue, workaround available.
P3 (Low)	Cosmetic or non-critical issue.

Table 1.2. Definitions

Acronym	Description
AI	Artificial Intelligence
API	Application Programming Interface
CSV	Comma-Separated Values
GDPR	General Data Protection Regulation
KVKK	Turkish Personal Data Protection Law
OWASP	Open Web Application Security Project
PR	Pull Request
RCA	Root Cause Analysis
SQA	Software Quality Assurance
SQAP	Software Quality Assurance Plan
UAT	User Acceptance Testing
CI/CD	Continuous Integration / Continuous Deployment
UI	User Interface
UX	User Experience
DB	Database

ML	Machine Learning
NLP	Natural Language Processing

Table 1.3. Acronyms

Abbreviation	Description
AI	Artificial Intelligence
API	Application Programming Interface
CSV	Comma-Separated Values
GDPR	General Data Protection Regulation
KVKK	Turkish Personal Data Protection Law
OWASP	Open Web Application Security Project
PR	Pull Request
RCA	Root Cause Analysis
SQA	Software Quality Assurance
SQAP	Software Quality Assurance Plan
UAT	User Acceptance Testing

CI/CD	Continuous Integration / Continuous Deployment
UI	User Interface
UX	User Experience
DB	Database
ML	Machine Learning
NLP	Natural Language Processing

Task 1.4. Abbreviation

1.4. References

To position NutriGame within the existing landscape of nutrition-tracking applications, we conducted an analysis of two reputable sources: Healthline's Best Nutrition Apps list [3] and the App Store's Top Nutrition-Tracking Apps [4]. Through this review, we identified three applications that appeared in both lists—MyFitnessPal, Lifesum, and Ate Food Journal—and scoped our comparison to these apps as benchmarks for our development.

MyFitnessPal is widely recognized for its comprehensive food database and efficient calorie counting features, allowing users to track nutrients and macros through manual entry or barcode scanning. However, despite its strong tracking capabilities, MyFitnessPal primarily focuses on quantitative nutrition data without integrating emotional well-being or motivational elements. In contrast, NutriGame complements similar tracking features with AI-powered food recognition, mood tracking, and gamification elements such as rewards and mini-games, designed to keep users both informed and engaged.

Lifesum offers personalized diet plans that adapt to users' specific health goals, incorporating features like calorie tracking, health assessments, and macro analysis. While effective in structured nutrition guidance, Lifesum remains limited in interactive and motivational aspects, lacking features like AI-based meal logging or emotional tracking. NutriGame builds on Lifesum's personalized recommendations by introducing diverse logging methods (photo, text), social sharing capabilities, and an integrated chatbot that offers both dietary advice and motivational support.

Ate Food Journal emphasizes mindful eating, encouraging users to reflect on their meals through a visual food diary and understand the emotional context of their eating habits. While it successfully fosters mindfulness, Ate does not provide nutritional data analysis, calorie tracking, or personalized meal planning. NutriGame bridges this gap by combining Ate's visual journaling approach with nutritional estimation, emotional monitoring, and actionable insights, transforming reflective tracking into a more data-driven and supportive experience.

With these analyses, our aim in developing NutriGame is to create a holistic application that integrates the strongest features of these leading apps while introducing novel functionalities. By blending AI-powered tracking, emotional well-being support, gamification, and personalized nutrition planning, NutriGame aspires to offer a comprehensive, engaging, and sustainable tool for users seeking to improve their health and eating habits.

Beyond these applications, academic literature supports the integration of several key design strategies that NutriGame employs. Research on gamification in Mobile Health applications [5] demonstrates that features such as points, badges, and progress streaks significantly boost user engagement and retention. This directly informs NutriGame's design, which leverages gamified systems to make health tracking both enjoyable and habit-forming.

Moreover, studies on emotional eating [6] reveal that stress and negative emotions are strongly linked to unhealthy eating behaviors. This underscores the importance of integrating emotional support features within nutrition apps. NutriGame addresses this need by incorporating a motivational chatbot that offers both dietary guidance and emotional encouragement, helping users navigate challenges and setbacks.

Additionally, personalization and real-time feedback are critical for sustaining behavior change, as highlighted in a comprehensive review of weight management apps [7]. Many existing solutions lack these features, leading to short-term rather than lasting improvements. NutriGame is explicitly designed to overcome this limitation with personalized meal recommendations, progress tracking, and adaptive feedback loops.

The value of consistency tools like streaks and reminders is well-documented in mobile health app research [8]. NutriGame's habit tracking and streak-based motivation systems are directly inspired by these findings, reinforcing consistent engagement. Furthermore, the importance of social connectivity in enhancing app engagement has been highlighted, with findings indicating that emotional and social features significantly contribute to sustained long-term use [9]. Accordingly, NutriGame integrates social sharing capabilities, enabling users to share progress and achievements within their networks.

Through this combined analysis of commercial applications and academic research, NutriGame aims to deliver a comprehensive, adaptive, and engaging nutrition management experience. By synthesizing the best features of existing apps with

evidence-based strategies from behavioral science, NutriGame aspires to not only facilitate dietary tracking but also promote sustainable, long-term health behavior change.

1.5. Overview

This report provides a comprehensive overview of the NutriGame project, covering all phases from initial idea to final implementation. It begins with 1. INTRODUCTION that explains the purpose, scope, and core concepts behind the project. The 2. Overall Description section outlines how the system fits into existing technologies, what it aims to achieve, and who the intended users are. The 3. Specific Requirements chapter defines all functional and non-functional needs of the system, including interfaces, constraints, and software attributes. The 4. DATASET section explains how data is used in the project and any limitations involved. The 5. System Architecture and Technical Design section describes the structural components of the application, technologies used, and the logic behind implementation decisions. This is followed by 6. Testing and Quality Assurance, which details how the software is tested, monitored, and improved for reliability and security. Next, 7. Project Management and Risk Analysis presents the development timeline, team responsibilities, and strategies to handle potential challenges. The 8. Use Cases and User Stories chapter describes typical user interactions and expectations. Finally, the report concludes with an analysis of the 9. Social, Environmental, and Legal Impact, followed by references and supporting documents in the Appendix. This structure ensures a clear and detailed explanation of the NutriGame project from concept to delivery.

2. Overall Description

2.1. Product Perspective

NutriGame is a standalone mobile application designed to make nutrition tracking more efficient, engaging, and socially interactive. While many existing apps rely on manual logging and basic calorie counting, NutriGame integrates AI-powered food recognition, gamification, and emotional well-being tracking to create a more holistic user experience.

System Context and Relationships
Although self-contained, NutriGame connects to several external systems:

- **USDA FoodData Central API** for accurate nutritional data.
- **Cloud-hosted AI models** (e.g., Hugging Face, Gemini API) for photo-based recognition, text parsing, and chatbot interactions.
- **Cloud infrastructure** (MongoDB Atlas, Cloudinary, Render, Railway) for secure and scalable storage and deployment.
- **Mobile device features** (camera, microphone) for interactive logging methods.

Core Features and Use Cases
Key capabilities include multiple meal logging options (manual, text, photo), personalized meal planning, mood and progress tracking, gamified challenges, and social sharing. NutriGame can be used at home, in restaurants, or while shopping—serving as a personal health companion in daily life.

Objectives and Differentiation
The primary goal is to promote long-term healthy eating habits by reducing the effort of meal logging, increasing engagement through gamification, and providing personalized support. Unlike competitors, NutriGame combines emotional well-being tracking, AI-driven assistance, and rich gamification within a single platform, making it both practical and motivating for users.

2.2. Product Functions

Function	Acceptance Criterion	Metric	Target Value	Verification Method	Standard/ Reference
Personalized Meal Planning	The system must generate a daily or weekly meal plan within 5 seconds after the user updates their profile or goals.	Meal Plan Generation Time (sec)	≤ 5 sec	Black-box functional test, stopwatch measurement	ISO/IEC 25010 – Performance Efficiency

Intelligent Meal Logging	The AI must correctly identify at least 90% of common food items from a provided photo or text description.	Food Recognition Accuracy (%)	≥ 90%	Test dataset comparison, confusion matrix analysis	ISO/IEC 25010 – Functional Suitability
Calorie & Nutrition Estimation	Estimated calorie values must differ from USDA database values by no more than 10%.	Calorie Estimation Error (%)	≤ 10%	Black-box test, USDA data comparison	ISO/IEC 25010 – Accuracy, USDA FoodData Central
Health and Mood Tracking	The user must be able to log mood and health metrics within a maximum of 3 clicks from the main dashboard.	User Interaction Steps	≤ 3	Usability testing, task completion measurement	ISO 9241-210 – Usability
Gamification	The user must receive points or rewards within 1 second after completing a tracked action.	Reward Delay (ms)	≤ 1000 ms	Black-box timing test, event log analysis	ISO/IEC 25010 – Performance Efficiency
Social Sharing	Shared content must appear on the selected platform without data loss in at least 95% of attempts.	Share Success Rate (%)	≥ 95%	End-to-end integration testing	ISO/IEC 25010 – Interoperability
AI Chatbot Support	The chatbot must provide a contextually relevant and goal-specific response within 2 seconds.	Response Time (sec)	≤ 2 sec	Response timing test, scenario-based testing	ISO/IEC 25010 – Responsiveness

Shopping List Generation	The weekly shopping list must include 100% of the ingredients from the planned meals.	Ingredient Coverage (%)	100%	Black-box functional testing	ISO/IEC 25010 – Completeness
Data History	The user must be able to retrieve any past meal, mood, or achievement log within 2 seconds.	Retrieval Time (sec)	≤ 2 sec	Database query performance test	ISO/IEC 25010 – Time Behaviour
Data Privacy & Consent	Users must not be prompted to share data without viewing the consent form.	Consent View Rate (%)	≥ 95%	Analytics log analysis, A/B testing	GDPR, KVKK – Data Protection & Privacy

Table 2.1. Product Functions and Details

2.3. User Characteristics

NutriGame is designed for a broad audience interested in improving or maintaining healthy eating habits through an engaging, AI-powered mobile platform. The target users are defined as follows:

- **Age**: Primarily individuals aged **18–65**. Users under 18 are excluded for privacy and legal compliance reasons.
- **Gender**: Inclusive of all genders.
- **Geographic Region**: Primary target regions include English-speaking countries where USDA FoodData Central API data is relevant (e.g., USA, Canada, UK, Australia). Future localization may include other languages and regions.
- **Occupations & Lifestyle**: Students, busy professionals, freelancers, and anyone seeking to manage nutrition alongside a busy schedule.
- **Interests**: Healthy living, fitness, weight management, cooking, food photography, and the use of mobile technology for lifestyle improvement.
- **Technical Proficiency**: Basic familiarity with using smartphones and mobile applications; no advanced technical skills required.
- **Health Literacy**: A general understanding of nutrition and wellness concepts is beneficial but not mandatory, as the app offers educational and guidance features.

Primary User Segments:

1. **General Health Enthusiasts** – Individuals interested in maintaining or improving daily nutritional habits.
2. **Individuals with Fitness or Dietary Goals** – Including weight loss, maintenance, muscle gain, or adopting specific dietary preferences (e.g., vegetarian, low-carb).

3. **People Seeking Social and Emotional Support** – Users who value community motivation, chatbots, and shared progress tracking to stay engaged

2.4. Constraints

NutriGame's development and deployment are subject to several technical, legal, operational, and temporal constraints that define its scope, performance, and compliance requirements.

Technical Constraints

- Device Requirements: Requires a smartphone (Android/iOS) with a camera, minimum 4 GB RAM, and stable internet connection.
- Low-End Device Limitations: AI processing may be slower or less accurate on devices with limited processing power or outdated operating systems.
- Third-Party API Dependency: Relies on USDA FoodData Central API, Hugging Face, and Gemini API for core functionalities such as food recognition and nutritional data retrieval. API downtime or policy changes may directly affect app performance.
- Memory and Storage Limits: The app is designed to operate with ≤ 500 MB storage usage and ≤ 250 MB RAM usage during normal operation.
- Concurrent Usage Capacity: Targeted to support up to 3,000 concurrent users with initial cloud infrastructure, scalable in later updates.
- Node Capacity: Backend services are designed to operate on up to 3 active server nodes in the initial deployment phase.
- Expected Error Rates: AI recognition accuracy target ≥ 90%; acceptable false positive rate ≤ 5%.

Temporal Constraints

- Development Timeline: Total project duration is 3 months, with a 4-person development team.
- Month 1: Requirements analysis, architecture design, dataset preparation.
- Month 2: Core feature development (meal logging, AI recognition, meal planning).
- Month 3: Testing, bug fixing, UI polishing, and deployment preparation.
- Testing Phase: Minimum 2 weeks allocated for integration, system, and user acceptance testing.
- Recovery Time Objective (RTO): System recovery from a critical server failure should be completed within 15 minutes.
- Recovery Point Objective (RPO): Maximum acceptable data loss in the event of failure is < 5 minutes of user activity.

Legal Constraints

- Data Privacy Compliance: Must comply with GDPR and KVKK regulations, including explicit user consent for data collection.
- No Medical Advice or Diagnosis: The app must include clear disclaimers stating that it does not replace professional healthcare consultation.
- Data Storage Locations: All personal data stored in servers located in GDPR-compliant regions.

Operational Constraints

- AI Dataset Limitations: Food recognition models may underperform with culturally specific or rare dishes not present in the training dataset.

- Environmental Limitations: Image-based logging accuracy may drop in poor lighting or with low-quality photos.
- Feature Limitations at Launch: No real-time human coaching, wearable device integration, or voice-based logging in the initial release.
- API Rate Limits: USDA API daily request limit and Hugging Face model inference rate must be respected to avoid service interruptions.

2.5. Assumptions and Dependencies

- Assumptions
    - Users provide truthful and up-to-date personal information (age, weight, dietary restrictions, goals).
    - Users have basic smartphone literacy and can navigate mobile apps without advanced training.
    - Users have a general understanding of basic nutrition concepts or are willing to learn through the app.
    - Adequate lighting and functional smartphone cameras (≥ 8 MP) are available for accurate image-based meal logging.
    - Minimal background noise is present when using voice input features (in future versions).
    - Users have reliable internet access with a minimum speed of 2 Mbps for AI and database operations.
    - Food recognition datasets are sufficiently diverse to cover common international and regional dishes, with planned updates to reduce cultural bias.
- Dependencies
    - Continuous availability of GPU-enabled cloud platforms for AI hosting and processing.
    - Stable third-party APIs (USDA FoodData Central, Hugging Face, Gemini) remain accessible, with uptime above 99%.
    - Nutritional databases and datasets remain regularly updated for accurate calorie and macronutrient data.
    - App remains compatible with major Android and iOS updates.
    - Ethical AI chatbot responses are continuously reviewed to avoid misinformation and ensure relevance.

3. Specific Requirements

3.1. External Interface Requirements

3.1.1. User Interfaces

NutriGame is designed as a mobile application built using the Flutter framework, ensuring compatibility across both Android and iOS platforms. The user interface focuses on simplicity, interactivity, and accessibility to support a broad range of users.

Core Screens

- **Onboarding & Consent** – Guides new users through the app's purpose, collects necessary permissions, and ensures informed consent before any data collection.
- **Authentication** – Simple login and sign-up flow with no more than 2 steps to complete.
- **Home (Dashboard)** – Displays a daily summary of nutritional intake, water consumption, streaks, mood, and quick access to logging and chatbot features.
- **Meal Logging** – Three input methods: manual search, text-based natural language input, and AI-powered food photo recognition. Includes a confirmation and editing step before saving.
- **Chatbot Interface** – Provides personalized nutrition tips, motivational support, recipe ideas, and mood check-ins.
- **Progress Tracker** – Shows weight history, calorie trends, mood logs, and achievement milestones with filtering options.
- **Mini-Game Center** – Offers fun challenges and games unlocked through earned points.
- **User Profile & Settings** – Allows editing of personal data, notification preferences, language, and measurement units.
- **Error & Empty States** – Designed for all critical flows, with retry options and help links.

Measurable Acceptance Criteria

Requirement	Metric	Target Value	Test Method
Consent must be viewed before any data entry	Consent view rate	≥ 95%	UAT + analytics
Meal logging process	Steps to completion	≤ 3 steps	Usability testing
Dashboard load time	First paint time	≤ 1 second	Profiling tools
Search results load time	Time to first result	≤ 1.5 seconds	Integration testing
Photo logging accuracy	Correct detection rate	≥ 85%	AI model validation
Chatbot response time	Response latency	≤ 2 seconds	Black-box testing
Accessibility contrast	Color contrast ratio	≥ 4.5:1	WCAG audit

Table 3.1. Requirements and Criteria

Accessibility Compliance (WCAG 2.1)

- All interactive elements have a touch target ≥ 44×44 pt.
- Dynamic font scaling up to 200% without layout breakage.
- Logical focus order with visible focus indicators.
- Alternative text for all meaningful images.
- Error messages are linked to relevant form fields programmatically.
- Main landmarks (header, main, nav) are labeled for screen readers.

Performance & Reliability

- Maintain ~60 FPS during navigation and animations.
- Show skeleton loaders for operations exceeding 300 ms.
- Cache unsent meal logs locally for retry in case of network failure.
- Full localization for Turkish and English, with proper date/number formatting.

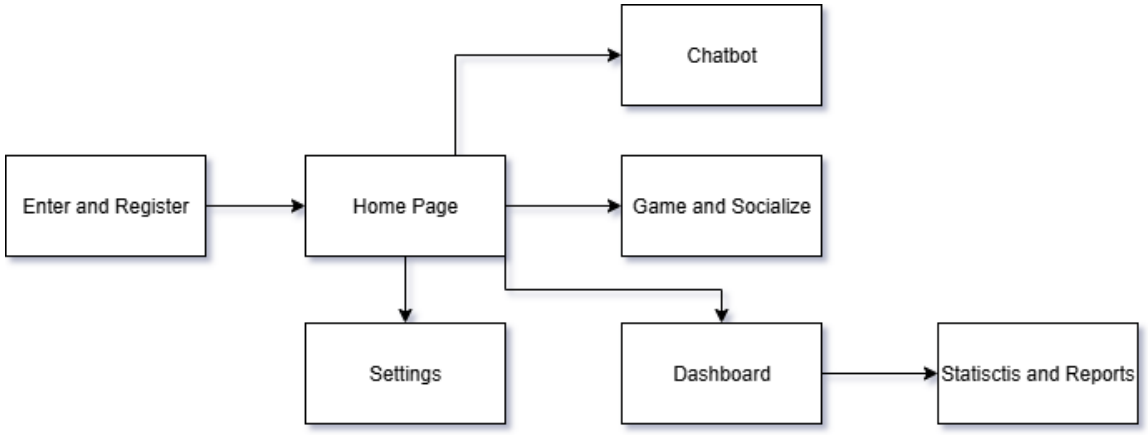

Figure 3.1.Application Flow Diagram

3.1.2. Hardware Interfaces

NutriGame leverages specific hardware capabilities of mobile devices to deliver its AI-powered nutrition tracking and engagement features.

- *Camera Access* – Required for capturing food images, which are then processed by the food recognition AI to estimate nutritional values.
    - Performance Target: Average image capture → cloud processing → result return time should be ≤ 3 seconds under stable internet connection.
- *Temporary Local Storage* – Captured images are stored temporarily in the device's local cache before being uploaded to the cloud for processing. Images are automatically deleted after processing to minimize storage usage and protect privacy.
    - Requirement: Maximum local cache allocation of 50 MB.

- *Cloud Infrastructure with GPU Support* – AI-powered services (food recognition, chatbot responses) operate on GPU-enabled cloud servers (e.g., Render, Railway) to ensure fast and accurate results.
    - Performance Target: AI chatbot responses should be returned within ≤ 2 seconds.
- *Voice Input Support (Optional)* – If enabled in future versions, the chatbot may use the device microphone for voice-based commands and queries.
    - Performance Target: Audio recording and transcription should complete within ≤ 2 seconds for a 10-second clip.
- *Minimum Device Requirements* – Android 10+ or iOS 14+, at least 3 GB RAM, 64-bit processor, and 200 MB of free storage space.
- *Excluded Hardware Integrations* – Integration with wearable devices or IoT systems (e.g., fitness bands, smart scales) is not included in the current version.

3.1.3. Software Interfaces

NutriGame integrates with multiple external and internal software services to deliver its core features. All software interfaces are accessed through secure HTTPS connections, adhering to TLS 1.3 standards. Authentication is managed through API keys, OAuth 2.0, or JSON Web Tokens (JWT) depending on the service.

1. Nutrition Database Interface

*Provider***:** USDA FoodData Central API

Purpose: Retrieve accurate macronutrient and caloric data for recognized or user-entered foods.

Input: Food name or description (string, UTF-8 encoded)
Optional filters for portion size or measurement unit

Output: JSON object containing nutrient details (calories, macros, micros) and portion size equivalents

Protocol: HTTPS (TLS 1.3), RESTful API

Authentication: API Key provided by USDA

2. Artificial Intelligence Services

a. Conversational AI

Provider: Gemini API or Hugging Face Transformers API

Purpose: Provide context-aware, emotionally supportive chatbot responses.

Input: User text query (string, UTF-8 encoded)
Optional context parameters (user goals, preferences)

Output: JSON object with chatbot response text and optional structured recommendations

Protocol: HTTPS (TLS 1.3), RESTful API

Authentication: OAuth 2.0 token or API key

b. Computer Vision for Food Recognition

Provider: Custom PyTorch model deployed via FastAPI on GPU-enabled cloud (Render or Railway)

Purpose: Identify food items in uploaded images and estimate nutritional values.

Input: Image file (JPEG/PNG, max 5 MB)

Output: JSON object with detected food labels, confidence scores, and estimated nutritional data

*Protocol***:** HTTPS (TLS 1.3), RESTful API

*Authentication***:** JWT

3. Database System Interfaces

*Provider***:** MongoDB Atlas (Cloud-based NoSQL)

*Purpose***:** Store user profiles, meal logs, mood entries, and gamification data.

Input: JSON documents (user data, activity logs, points)

*Output***:** JSON documents retrieved via query

*Protocol***:** MongoDB Atlas Driver over TLS 1.3

*Authentication***:** Encrypted connection with username/password or token

*Provider***:** Cloudinary

*Purpose***:** Store food photos and return optimized URLs for database reference.

*Input***:** Image file (JPEG/PNG, max 5 MB)

*Output***:** JSON object containing hosted file URL and metadata

*Protocol***:** HTTPS (TLS 1.3), RESTful API

*Authentication***:** API key + secret

4. Authentication Service Interface

Provider: Firebase Authentication (Google)

Purpose: Secure user registration, login, and session management.

Input: Email/password pair (string, encrypted)

Output: JSON Web Token (JWT) for session authentication

Protocol: HTTPS (TLS 1.3), RESTful API

Authentication: Managed internally by Firebase

3.1.4. Communication Interfaces

NutriGame employs secure and standards-based communication between its mobile client and backend services to ensure confidentiality, integrity, and availability of data. All interactions between the app and the server are transmitted over HTTPS using TLS 1.3, with support for HTTP/1.1 and HTTP/2 depending on device and network capabilities. Data exchanged between components is structured in JSON format, compliant with RFC 8259, while media uploads such as food images use multipart/form-data as defined in RFC 7578. UTF-8 encoding is enforced across all transmissions to guarantee compatibility and consistency.

For real-time interactions, particularly with the AI-powered chatbot, NutriGame utilizes secure WebSocket connections (wss://) following RFC 6455. These connections support events such as session initiation, user messages, bot responses, and heartbeat signals, enabling a natural and responsive conversation flow. In environments where WebSockets are unavailable, the system falls back to short-interval polling, maintaining a responsive user experience. The WebSocket lifecycle is illustrated in Figure 3.2**.** below, showing the connection states and transitions.

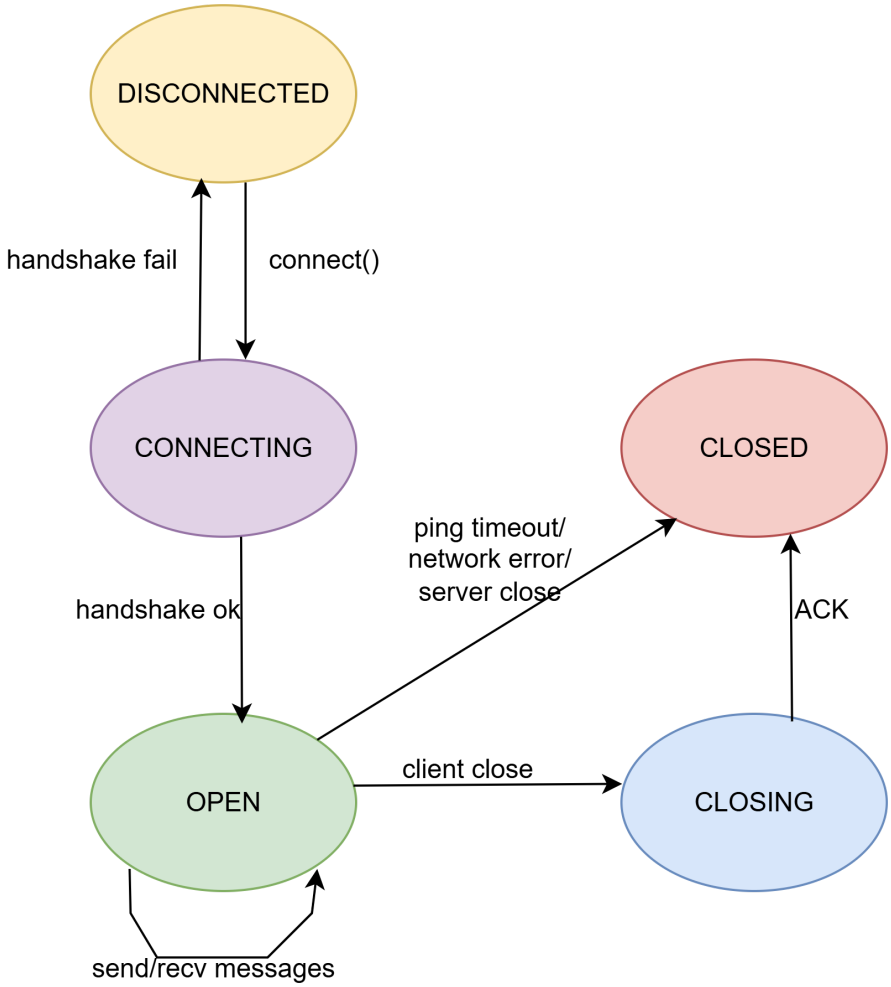

Figure 3.2. WebSocket State Diagram

Error handling is implemented according to standard HTTP status codes, providing both system-level diagnostics and user-friendly feedback. Failures trigger retries with exponential backoff and jitter to minimize service disruption. All API calls and streaming events are logged with correlation IDs for traceability and debugging purposes.

Security is enforced at multiple layers: all credentials are transmitted only over encrypted channels and stored in hashed form using bcrypt, authentication tokens follow the Bearer JWT format, and sensitive information is never stored or transmitted in plain text. The API gateway enforces HSTS policies, and secrets such as API keys are rotated according to policy to reduce exposure risks.

By combining robust transport protocols, well-defined data formats, and strong security controls, NutriGame's communication interfaces provide a reliable and standards-compliant foundation for all client-server interactions.

3.2. Functional Requirements

- **FR-01** – The system shall generate daily and weekly meal plans based on the user's goals.
- **FR-02** – The system shall allow users to log meals through manual search, natural language input, and photo upload methods.
- **FR-03** – The system shall estimate calories by recognizing food items from uploaded photos.
- **FR-04** – The system shall analyze food descriptions entered in natural language by mapping them to the USDA nutrition database.
- **FR-05** – The system shall track information related to water intake, weight changes, and mood.
- **FR-06** – The system shall provide gamification elements by awarding points when users achieve their daily goals.
- **FR-07** – The system shall store user profile information, including age, gender, goals, and dietary preferences.
- **FR-08** – The system shall archive users' historical data, such as daily meals, mood entries, weight records, and achievements.
- **FR-09** – The system shall access the mobile device’s camera to allow users to take and upload food photos.
- **FR-10** – The system shall allow users to interact with each other.
- **FR-11** – The system shall include healthy and diet-friendly recipes for users.
- **FR-12** – The system shall allow users to upload and share their own recipes.

These IDs will be used in section 8.2 to assign user cases.

3.3. Performance Requirements

- Under normal network conditions (≥10 Mbps), the system must return food image analysis and calorie estimation results within 10 seconds.
- Basic database queries must complete in ≤ 1 second.
- The architecture shall support at least 200 concurrent active users without performance degradation.
- The system shall handle up to 5,000 requests per day reliably.
- Monthly uptime shall be maintained at ≥ 99%.
- The acceptable API error rate shall not exceed 1% of total requests, excluding user input errors.
- In case of a system failure, automated recovery shall restore service within 5 minutes.
- The system must be able to process at least 10,000 food images per month without exceeding computational or storage limits.
- Image uploads of up to 5 MB must complete in under 3 seconds under standard broadband conditions.

3.4. Design Constraints

- All user data processing shall comply with **GDPR** and **KVKK** data protection regulations.

- The application shall adhere to **WCAG 2.1** accessibility guidelines and **OWASP Mobile Security** best practices.
- The system shall not provide any medical diagnoses or professional health advice and must display clear disclaimers to that effect.
- The application shall run on devices with **at least 2 GB RAM**, **Android 9.0 (API Level 28)** or later, and **iOS 14** or later, with a minimum of **200 MB free storage space**.
- Cloud API providers must support **GPU acceleration** to ensure real-time AI-powered features such as image recognition and chatbot responses.
- The mobile application shall be developed using **Flutter** with **Dart** as the primary programming language, following industry-standard clean architecture principles.
- All third-party libraries and APIs integrated into the project shall comply with their respective **open-source or commercial license agreements**, and license compatibility must be verified before integration.
- The system must support offline caching for limited functionality when the internet connection is unstable, but core AI features will require an active connection.

3.5. Software System Attributes

3.5.1. Reliability

- The system shall maintain a monthly uptime of **at least 95%**, with a **Mean Time Between Failures (MTTF)** of **90 days or more**.
- In case of server or service failures, user data shall be fully recoverable, and the **Mean Time to Recovery (MTTR)** shall not exceed **30 minutes** for critical functions.
- Fault tolerance shall be achieved through redundant servers, automated failover mechanisms, and database replication.
- All transactions shall maintain **data consistency** even during unexpected shutdowns or partial failures.
- The system shall be resilient against unexpected conditions such as sudden traffic spikes, API downtime, or network instability, ensuring uninterrupted operation of core features.

3.5.2. Availability

- The system shall maintain a monthly uptime of **at least 99%**.
- The **Mean Time Between Failures (MTBF)** shall be **90 days or more**.
- The **Mean Time to Repair (MTTR)** for critical services shall not exceed **30 minutes**.
- Scheduled maintenance windows shall not exceed **2 hours per month** and will be announced to users at least **24 hours in advance**.
- Service interruptions shall be detected within **1 minute** through automated monitoring tools.

- Automatic recovery mechanisms shall restore core functionalities without manual intervention whenever possible.
- Redundant servers and load balancing shall be used to distribute traffic and prevent downtime during high-demand periods.
- Continuous system monitoring with real-time alerts shall be implemented to detect and respond to anomalies immediately.

3.5.3. Security

- All client-server communication shall be encrypted using **HTTPS with TLS 1.2+** to ensure confidentiality and prevent data interception.
- User authentication shall be handled via **secure login mechanisms** (e.g., email/password with bcrypt hashing, optional two-factor authentication) and **authorization** shall be enforced using **role-based access control (RBAC)** to restrict access to sensitive features.
- Data integrity shall be maintained through **checksum or hash validation** for critical transactions, ensuring no unauthorized modifications occur.
- User privacy shall be protected in compliance with **GDPR** and **KVKK**, with minimal data collection and explicit consent for sensitive information.
- An **audit logging** system shall record significant actions (logins, profile updates, data exports) for traceability and security incident investigation.
- Defense-in-depth strategies shall be applied, including **input validation**, **rate limiting**, and **firewall protections** to mitigate common attack vectors such as SQL injection, XSS, and brute force attacks.
- An **Intrusion Detection/Prevention System (IDS/IPS)** shall monitor unusual activities and trigger automated alerts.
- All third-party dependencies and server environments shall undergo **regular security patching and vulnerability scans** to prevent exploitation of known issues.

3.5.4. Maintainability

- The codebase shall be **modular**, enabling each component to be updated, replaced, or extended independently without impacting unrelated modules.
- All source code shall follow **clear naming conventions** and be **fully documented** to ensure understandability for new developers.
- The system shall be **fully testable**, including unit tests, integration tests, and automated regression testing to verify functionality after changes.
- Design patterns and **code reuse** practices shall be applied to minimize duplication and enhance maintainability.
- The architecture shall allow **easy modification** and **future adaptation** for new features without requiring major rewrites.
- All code shall comply with agreed **style guides and linting rules**, ensuring consistency across the project.
- **Version control** shall be maintained via Git, with standardized commit messages, branch naming policies, and code review processes.

- All external dependencies shall be managed through **dependency management tools** (e.g., npm, pip), with regular updates and vulnerability scans to ensure security and compatibility.

3.5.5. Portability

- The mobile application shall be developed using Flutter, ensuring compatibility with both Android and iOS platforms.
- Backend services shall be platform-independent and accessible via FAST APIs
- Data will be stored and transmitted using standard formats (e.g., JSON, UTF-8 encoding) to maintain compatibility across different environments. Additionally, the architecture will follow clean code and modular design principles, making it easier to extend the system to new platforms without a complete redevelopment.
- The application will support multiple language**s** from the start, including Turkish and English, with the flexibility to add more languages in the future. All text content will be stored in a way that allows easy localization without changing the core code.

3.6. Other Requirements

- Food photos uploaded by users shall be temporarily stored on the local device and automatically deleted after processing.
- The chatbot shall communicate in an empathetic and encouraging tone, avoiding judgmental or medically suggestive language.
- The application shall support both Turkish and English languages.
- A simple feedback form shall be included to collect user opinions and suggestions.

4. DATASET

4.1. Dataset Description

This project utilizes four types of datasets collected from various sources: healthy recipe data, psychological counseling data, image segmentation datasets, and image classification datasets. Each dataset serves a specific purpose in the system's development and evaluation.

1. Healthy Recipe Dataset

To build a dataset for nutritional analysis and healthy food recommendation:

- **Sources:**
  - https://fdc.nal.usda.gov/download-datasets
  - https://www.kaggle.com/datasets/utsavdey1410/food-nutrition-dataset
  - https://practicaldsc.org/final-project/datasets/recipes-and-ratings/#:~:text=Download%20the%20data%20here,download%20two%20CSV%20files
  - https://www.eatingwell.com/recipes/

- **Tools & Methods:**
    - Data scraping was performed using Python, BeautifulSoup, and Requests.
    - Structured data such as ingredients, cooking instructions, categories, calories and nutritional values were extracted.
    - Recipes from Kaggle and Hugging Face were downloaded in CSV/JSON format and merged with scraped data.
- **Ethical Considerations:**
    - All datasets were used for non-commercial academic purposes only.
    - No personal data was collected.

2. Psychology Dataset

To analyze psychological patterns related to eating disorders (e.g., anorexia, bulimia, binge eating disorder), various mental health datasets were used. Datasets format is question-answer.

- **Sources:**
    - https://huggingface.co/datasets/Amod/mental_health_counseling_conversations/viewer/default/train?row=8&views%5B%5D=train
    - https://huggingface.co/datasets/loaiabdalslam/counselchat
    - https://huggingface.co/datasets/nbertagnolli/counsel-chat
    - https://www.kaggle.com/datasets/zuhairhasanshaik/datacsv
    - https://www.kaggle.com/datasets/melissamonfared/mental-health-counseling-conversations-k
    - https://huggingface.co/collections/johua153/mental-therapy-datasets-662dbc1360df97baa82c07a9
- **Tools & Preprocessing:**
    - Python, Pandas, Scikit-learn
    - Text cleaning, label encoding, stop-word removal for NLP tasks
    - Data was anonymized and pre-labeled for supervised learning
- **Ethical Considerations:**
    - All data used was anonymized and suitable for non-commercial academic analysis

3. Image Segmentation Dataset

For food recognition tasks, we used datasets that include annotated images with segmentation masks and bounding box information.

- **Sources:**
  - https://mm.cs.uec.ac.jp/uecfoodpix/?
  - https://datasetninja.com/food-seg-103
  - Food Recognition 2022 - Dataset Ninja
  - https://www.kaggle.com/datasets/thezaza102/tray-food-segmentation

- **Tools:**
  - Data loaded using PyTorch.

- **Ethical Considerations:**
  - No personally identifiable visuals were used.

4. Image Classification

The datasets consist of various food categories, including Turkish and general cuisines, with corresponding image labels for classification tasks.

- **Sources:**
  - https://huggingface.co/datasets/yunusserhat/TurkishFoods-25
  - https://www.kaggle.com/datasets/dietapp/turkish-food
  - https://www.kaggle.com/datasets/bingolo/turkishcuisinenet
  - https://github.com/ChiShengChen/Food2K-TW101?tab=readme-ov-file
- **Tools:**
  - Data loaded using PyTorch.

- **Ethical Considerations:**
  - All images were publicly released.

  - No personally identifiable visuals were used.

4.2. Data Collection (If Applicable)

For this project, we did not create new data; instead, we utilized a wide range of free, open-source datasets that match our needs. Data was gathered from multiple sources, including USDA FoodData Central, Kaggle datasets, Hugging Face repositories, academic datasets, and public recipe/food image databases. Collection methods varied by dataset type:

- **Web scraping** of recipe data using Python, BeautifulSoup, and Requests to extract structured nutritional information.

- **Direct downloads** of CSV/JSON files from open repositories such as Kaggle and Hugging Face.

- **Loading and processing** of image datasets using PyTorch for segmentation and classification tasks.

- **Text data preprocessing** with Python, Pandas, and Scikit-learn, including cleaning, label encoding, and stop-word removal for NLP tasks.

All datasets were used solely for non-commercial academic purposes, with no personal or identifiable data collected.

4.3. Data Storage

NutriGame uses two main tools for storing data: a database and a media storage service. These tools are chosen to keep the app fast, organized, and secure.

First, we use MongoDB Atlas as the main database to store structured data. This includes user information (like age, weight, goals), meal logs, mood entries, and points earned through gamification. MongoDB is flexible and easy to use with changing data types, which is perfect for tracking different user habits. Thanks to the GitHub Student Developer Pack, we can use a higher storage limit for free during development.

Second, we use Cloudinary for storing images such as food photos. Instead of saving large image files directly in the database, we upload them to Cloudinary. The app then stores only the link (URL) of the image in MongoDB. This keeps the database lighter and improves performance. Cloudinary also automatically resizes and optimizes the images to make loading faster.

Although NutriGame is not a real-time system like a chat app, we use real-time data syncing with MongoDB to make sure that user data (like weight logs or mood entries) updates instantly in the app. This gives users a smooth and responsive experience.

We do not use large-scale data lakes because the project does not require storing raw, unstructured data from multiple sources. A structured NoSQL database is more suitable for our needs.

All data is protected using HTTPS encryption, and user passwords are securely hashed using bcrypt. This ensures privacy and security for all users.

4.4. Data Preprocessing

Since the datasets used in this project were collected from various online sources—such as Kaggle, Hugging Face, nutrition databases, and web scraping—comprehensive preprocessing was essential to ensure consistency, accuracy, and usability across different data types. Raw data often included inconsistencies, duplicates, missing values, and formatting issues that required cleaning and transformation before use.

For image classification tasks, all images were resized to a standard input shape, corrupted or unreadable files were removed, and class labels were reviewed and

standardized. In the case of image segmentation, image-mask pairs were carefully validated to ensure alignment, and low-quality or irrelevant samples (such as those not depicting food items) were excluded from the dataset.

Text-based data, particularly in the psychology domain, required natural language preprocessing. This included converting all text to lowercase, removing punctuation and stop words, and encoding categorical labels for use in machine learning models. Additionally, since eating disorder–specific conversations were not readily available, any synthetic text data generated using large language models (e.g., GPT) was also normalized and reviewed for consistency.

Structured data such as nutritional values and ingredient lists underwent parsing, unit standardization, and numeric normalization. Food names predicted by the model were matched with entries in official nutrition databases (e.g., TürKomp and USDA), and approximate portion sizes were used to estimate calorie values reliably.

In summary, a full-scale preprocessing pipeline was implemented to clean, filter, convert, and align data from diverse modalities. This ensures that each dataset is ready for accurate training, evaluation, and integration into the system.

4.5. **Data Limitations and Assumptions (If Applicable)**

In the psychology domain, we were unable to find datasets specifically focused on eating disorders. To overcome this, we plan to generate synthetic conversation data using generative AI models (ex. Gemini, GPT). This will help simulate relevant scenarios for model training, though it may introduce some bias.

In the image segmentation task, the available datasets are limited in size and lack specific examples of Turkish food. This may impact model performance and generalization, especially for culturally unique dishes.

5. System Architecture and Technical Design

This section details the methods and processes used to develop the project. It covers requirement gathering, system design, implementation strategies, and the selection of tools and technologies.

5.1. Architectural Design

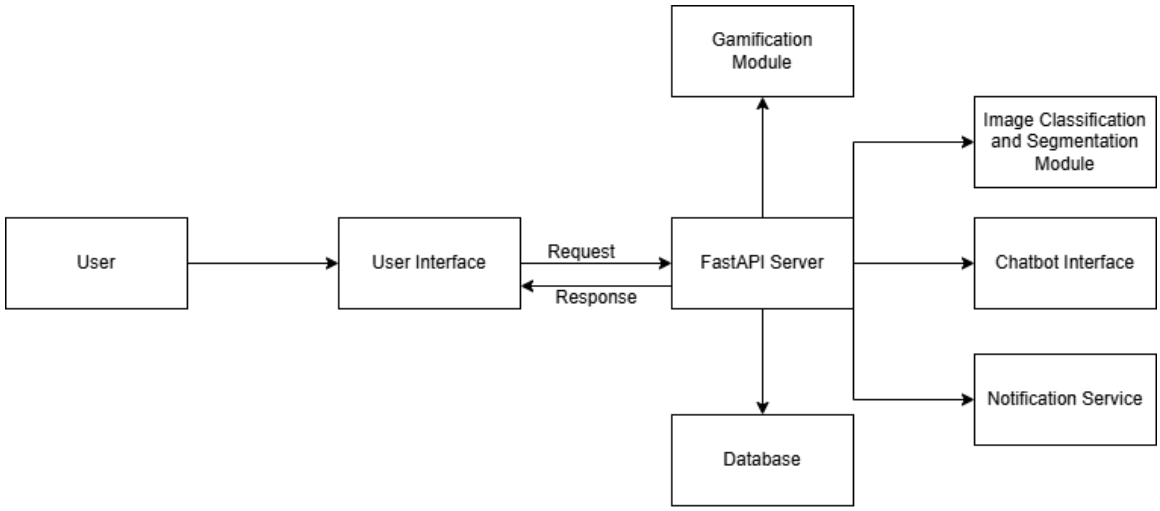

Figure 5.1. Architectural design of the project

5.2. Technology Stack

UI — Flutter
Backend — FastAPI
Database — PostgreSQL/MongoDB
AI — Gemini API, HuggingFace, Python
DevOps — Render/Railway

5.3. Implementation Details

The system will be developed as a cross-platform mobile application using Flutter for the frontend interface. The backend will be powered by FastAPI, a high-performance Python web framework, to serve RESTful APIs for communication between the client, AI modules, and the database.

1. Frontend

   Flutter will provide a responsive and modern user interface for users to:

   Log meals (manually or via photo),
   Communicate with the chatbot,
   Track calories, achievements, and goals,
   Interact with friends, streak and gaming leaderboard

2. Backend

   The backend will be implemented using FastAPI, a modern and high-performance Python framework. FastAPI will handle authentication, manage API requests and responses, store and retrieve data from the database, trigger notifications, and coordinate communication with the AI modules. Thanks to its asynchronous structure, FastAPI will be able to

efficiently handle tasks such as image processing without blocking other user requests.

3. Database

   Data will be stored in either PostgreSQL or MongoDB, depending on the data structure. PostgreSQL will be used for relational data such as user profiles and food logs, while MongoDB may be utilized for more flexible, document-based data like chat history or image metadata.

4. Chatbot Module

   The chatbot module will be powered by HuggingFace Transformers for natural language understanding and response generation. The chatbot will engage users with meal suggestions, motivational feedback, and responses to health and nutrition-related questions. Conversation history will be stored in the database to enhance user experience and allow context-aware interactions.

5. Gamification Module

   The gamification module will encourage user engagement by rewarding points, badges, and achievements for reaching calorie goals, completing daily tasks, or maintaining healthy habits. User progress will be managed through FastAPI and stored in the database, and visual feedback will be provided within the app to increase motivation and retention.

6. Image Classification and Segmentation

   When a user uploads a food photo, the image will be sent to an AI-powered image classification and segmentation service. This service may use pre-trained models from HuggingFace, TensorFlow. Once the food item is identified, it will be matched with entries from a food database to estimate the calories and nutritional values, which will then be returned to the user in real-time.

6. Testing & Quality Assurance

6.1. Software Quality Assurance Plan (SQAP)

The purpose of this Software Quality Assurance Plan is to ensure that the NutriGame application operates reliably, accurately, and sustainably, while fully meeting user needs and expectations. The scope of the plan covers all modules of the system, including meal logging, the AI-powered chatbot, gamification elements, data processing components, and the mobile user interface. By addressing each of these modules, the plan ensures comprehensive quality control across the entire project, from development to deployment, maintaining both functional performance and compliance with relevant standards.

- **Quality Objectives**
  - AI food recognition accuracy ≥ 90%
  - API response time ≤ 2 seconds
  - Application crash rate ≤ 1% per month
  - User data loss = 0
  - GDPR/KVKK compliance = 100%

- **Applicable Standards and Best Practices**
  - IEEE 730 (SQA Plan standard)
  - ISO/IEC 25010 (software quality model)
  - ISO 9241-210 (usability)
  - PEP8 (Python coding standard)
  - OWASP secure software development guidelines

- **SQA Activities**
  - Code review (PR review, static analysis)
  - Unit, integration, system, and acceptance tests
  - Data validation and bias checking
  - Process audits
  - Bug tracking and root cause analysis (GitHub Issues/Jira)

- **Responsibilities**
  - **SQA Lead**: Test strategy, reporting
  - **Backend Developer**: API validation, security testing
  - **ML Engineer**: Model validation, data cleaning
  - **Frontend Developer**: UI testing, usability checks
  - **Team**: Acceptance criteria, user testing

- **Verification and Validation**
  - **Verification**: Requirement-based testing, code coverage ≥ 80%
  - **Validation**: User acceptance tests, pilot user feedback

- **Bug and Change Management**
  - Bug tracking: Jira/GitHub Issues
  - Change approval: Pull request review + QA approval
  - Versioning: Git flow, semantic versioning

- **Test Plan and Criteria**
  - Unit test: ≥ 80% coverage for each module
  - Integration test: Backend–Frontend–AI integration
  - Acceptance test: Test scenarios for every item in the requirements list
  - Performance test: Stable operation with 20 concurrent users

- **Reporting and Record Keeping**
  - Weekly test report (passed/remaining bugs)
  - Model metrics (accuracy, F1-score, drift) tracked monthly
  - Traceability matrix (requirement → test case → result)

6.2. Verification and Validation (V&V)

In the NutriGame project, verification and validation processes will be defined to ensure that the software meets all requirements and operates correctly in real-world usage scenarios. The purpose of testing is to ensure that all modules (meal logging, AI chatbot, gamification, data processing, mobile interface) function reliably, accurately, and in compliance with legal requirements (GDPR, KVKK).

Success Criteria

- Success rate in all critical functions ≥ 95% in test scenarios
- AI food recognition accuracy ≥ 90%
- API response time ≤ 2 seconds
- Number of open P0/P1 issues = 0
- Test coverage (unit + integration) ≥ 80%

The methods to be used will include requirement-based testing, code reviews, static analysis, unit/integration/system testing, user acceptance testing, and security testing. Tests will be conducted on Android/iOS devices, using GPU-supported cloud infrastructure, MongoDB Atlas, and Cloudinary, with anonymized test data.

Test results will be recorded on Jira/GitHub Issues, and root cause analysis (RCA) will be performed for failed tests. Requirement IDs will be matched with test scenarios (Traceability Matrix). Performance and security reports will be evaluated separately.

Improvement Suggestions

- Retraining the AI model with additional data based on post-test findings
- UI/UX design updates to address usability issues
- Backend optimization if API response times are high
- Expanding test automation coverage

Test Participants

- **Test Supervisor**: Execution of test plans and scenarios
- **ML Developer**: AI model testing and metric evaluation
- **Backend Developer**: API integration testing
- **Frontend Developer**: UI testing, mobile device compatibility

Test ID	Requirement No	Test Type	Expected Result
T-001	FR-01	System Test	Generates daily/weekly meal plans correctly
T-002	FR-02	Integration Test	User can log meals via search, natural language, and photo upload
T-003	FR-03	AI/Functional Test	Correct calorie estimation from food photos
T-004	FR-04	Functional Test	Food descriptions map correctly to USDA DB

T-005	FR-05	System Test	Tracks water intake, weight, and mood
T-006	FR-06	Functional Test	Awards points for achieving goals
T-007	FR-07	Integration Test	Stores user profile info accurately
T-008	FR-08	System Test	Archives historical user data
T-009	FR-09	Device Integration	Accesses camera and uploads photos
T-010	FR-10	Functional Test	Enables user interaction
T-011	FR-11	Functional Test	Displays healthy and diet-friendly recipes
T-012	FR-12	Functional Test	Allows recipe upload and sharing

Table 6.1. Matrix of Matching Between Test Numbers and Requirement Numbers

6.3. Configuration and Change Management

In the NutriGame project, configuration management and change tracking will be carried out using a Git-based version control system (GitHub). All source code, documentation, and configuration files will be stored in a central repository and managed using the main development branch (*main*) and feature branches (*feature branches*). Changes will undergo code review through the Pull Request (PR) process before being merged, and at least one developer's approval will be required before integration into the main branch.

Versioning will follow the **Semantic Versioning** format (**MAJOR.MINOR.PATCH**). Before each new release, automated tests, security scans, and builds will be executed through the CI/CD pipeline, and only successful builds will be deployed to the production environment.

Change requests will be recorded in Jira, linked to the corresponding requirement number and responsible person. An impact analysis will be performed for each change, and additional test scenarios will be created when necessary. Critical configuration files

(config, API keys, environment variables, etc.) will be stored in a secure manner with access limited to authorized personnel.

6.4. Risk and Defect Management

In the NutriGame project, risks and defects will be managed systematically to ensure software reliability and timely issue resolution.

Defect Management
All software defects will be tracked using Jira and GitHub Issues. Each defect will be assigned a unique ID, linked to the related requirement, and classified by severity:

- **P0 (Critical)** – Blocks core functionality, must be fixed immediately before release.
- **P1 (High)** – Major functionality issue, fix required in the next release cycle.
- **P2 (Medium)** – Minor functionality issue, workaround available.
- **P3 (Low)** – Cosmetic or non-critical issue.

Defects will follow a standard lifecycle: **New → In Progress → Fixed → Verified → Closed**. Root cause analysis (RCA) will be performed for recurring or high-severity issues, and regression tests will be applied before closure.

Risk Management
Potential risks (technical, operational, and compliance-related) will be identified during project planning and updated throughout development. Risks will be documented in a Risk Register, including probability, impact, mitigation strategy, and responsible owner. Key risk categories include:

- AI model accuracy falling below the target threshold
- Data privacy and GDPR/KVKK compliance breaches
- Performance degradation under high user load
- Security vulnerabilities in APIs or cloud infrastructure
- Delays in dataset preparation or integration

Mitigation strategies will be defined for each risk, and high-priority risks will be reviewed weekly during project meetings.

This structured approach ensures that both risks and defects are tracked, classified, resolved, and documented, enabling better quality control and reducing potential project delays.

6.5. Product Evaluation and Acceptance

The acceptance process for the NutriGame project is defined to verify that the software meets the requirements for performance, functionality, security, and usability.

Acceptance Criteria

- **Performance**: API response time ≤ 2 seconds, 100% uptime with 20 concurrent users
- **Functionality**: 100% of the features listed in the requirements implemented and operational
- **Security**: GDPR/KVKK compliance, no critical vulnerabilities (OWASP Top 10)
- **Usability**: ≥ 85% satisfaction score in User Acceptance Testing (UAT), ≥ 90% task completion rate
- **Error Handling**: No open P0/P1 defects; clear error messages and recovery steps provided to the user in case of failures

Acceptance Testing and Review Process

- Successful completion of all requirement-based test scenarios
- Passing all functional, performance, security, and usability tests
- Completion of peer code reviews
- Positive results from independent audits or evaluations if applicable
- Approval from the project supervisor

Once these criteria are met, the project will be considered “accepted” and granted approval for deployment to the production environment.

7. Project Management and Risk Analysis

7.1. Project Plan

Gantt Chart is provided with this link: GR8SLAYERS GANTT CHART.xlsx

7.2. Task Distribution

Task	Team Member(s)	Description
Dataset Review	Nazlı Ceyda Ünsoy	Reviewed data sources and defined labeling criteria.
Dataset Preprocessing	Nazlı Ceyda Ünsoy	Cleaned, formatted, and augmented the dataset.
UI/UX Wireframes	Duygu Tümer	Created wireframes and designed the main application screens.
Backend API Development	Betül Aydeğer, Zeynep Yavuz	Built the core backend API structure.
Authentication Module	Betül Aydeğer, Zeynep Yavuz	Implemented login, registration, and authentication features using JWT/OAuth.
Meal Logging APIs	Betül Aydeğer, Zeynep Yavuz	Developed APIs for meal logging (CRUD operations).
Flutter Frontend	Duygu Tümer, Betül Aydeğer	Developed the mobile app's core screens and navigation flow.
DB and Auth Setup	Betül Aydeğer, Zeynep Yavuz	Designed database schema and integrated authentication.
AI Chatbot & Food Recognition	Nazlı Ceyda Ünsoy	Integrated AI chatbot for dietary assistance and food image recognition.
Chatbot Training	Nazlı Ceyda Ünsoy	Trained the chatbot model using processed datasets.
Food Image Model	Nazlı Ceyda Ünsoy	Developed and fine-tuned the food classification model.
Gamification Features	Duygu Tümer, Betül Aydeğer	Added gamified features like badges and point system.

Frontend–Backend Integration	Betül Aydeğer, Zeynep Yavuz, Duygu Tümer, Nazlı Ceyda Ünsoy	Connected backend services to the frontend interface.
Unit & Integration Tests	Zeynep Yavuz	Performed unit and integration testing.
UX Testing	Duygu Tümer	Conducted usability testing and gathered user feedback.
Model Monitoring	Nazlı Ceyda Ünsoy	Monitored AI model performance in testing environment.
UI/UX Revisions	Duygu Tümer	Refined design based on test results and feedback.
Final Build & Deployment	Betül Aydeğer, Zeynep Yavuz	Packaged the final product and deployed it to production.

Table 7.1. Task Distibution

7.3. Risk Analysis and Mitigation Strategies

As NutriGame integrates AI-powered features such as food recognition, calorie estimation, and personalized recommendations, several technical and ethical risks must be addressed to ensure reliability, fairness, and trustworthiness.

Data Bias

AI models rely on training data that may not fully represent the diversity of foods from all cultures, regions, or homemade recipes. This lack of representation can lead to biased or inaccurate predictions, especially for users whose diets include less common or region-specific dishes. As a result, calorie estimates or food recognition may be less reliable for these groups, potentially reducing the app's overall effectiveness and user satisfaction.

Mitigation Strategy For Data Bias:

We will actively test the AI model across diverse food categories and user groups to evaluate performance gaps. User feedback mechanisms will allow users to flag incorrect classifications or suggest missing foods. In the long term, we aim to fine-tune the model using additional, diverse datasets, including user-contributed food images (with consent) to reduce bias.

AI Explainability

AI systems often behave like "black boxes," making it difficult for users to understand why a certain recommendation or calorie estimation was made. This can reduce trust in the system.

Mitigation Strategy For AI Explainability:

NutriGame will incorporate explainable AI (XAI) practices by showing users the reasoning behind certain outputs (e.g., “Estimated 250 kcal based on portion size and similar items in our database”). For recipe suggestions or calorie estimates, users will also be able to see the ingredient breakdown and data source to better understand the result.

Model Drift
Over time, changes in user behavior, food trends, or photo quality (e.g., new phone cameras) may cause the AI model’s accuracy to degrade a phenomenon known as model drift.

Mitigation Strategy For Model Drift:
We plan to monitor the performance of AI predictions continuously using real-world inputs and track accuracy trends. Periodic model retraining will be conducted using updated data, and a fallback mechanism will be available such as manual calorie input in case the model produces uncertain results.

Ethical Concerns
Using AI in health and wellness applications comes with ethical responsibilities. Users may become overly reliant on the app, develop unhealthy habits due to inaccurate information, or experience negative psychological effects related to body image or eating behavior.

Mitigation Strategy For Ethical Concerns:
NutriGame is designed as a supportive tool, not a medical advisor. Clear disclaimers will be included to indicate that it is not a substitute for professional dietary guidance. The chatbot will avoid triggering language and offer motivational support in a non-judgmental, encouraging tone. Additionally, privacy and consent will be prioritized in all data collection and use, aligning with GDPR and ethical design practices.

Economic Risks
The development and maintenance of AI-powered features depend on third-party APIs, cloud services, and infrastructure that may experience price fluctuations. Unexpected increases in API usage costs, cloud storage fees, or server expenses could strain the project budget. Additionally, delays in funding or resource allocation could impact development timelines.

Mitigation Strategy For Economic Risks:
We will establish a detailed budget with a contingency reserve to handle cost overruns. Long-term contracts or usage commitments with service providers will be considered to lock in favorable pricing. Alternative service providers and open-source tools will be evaluated to reduce dependency on any single vendor. Regular cost monitoring and reporting will ensure early detection of budget risks.

8. Use Cases and User Stories

8.1. Target Audience

The primary audience for NutriGame includes:

- Young Adults and University Students: Individuals juggling studies, social life, and health goals often neglect nutrition due to time constraints.
- Working Professionals: People with busy work schedules who struggle to plan meals and maintain consistent healthy eating habits.
- Fitness Enthusiasts: Users aiming to optimize their diet for muscle gain, fat loss, or athletic performance.
- People with Emotional Eating Patterns: Individuals who experience stress-related or emotional eating and need motivational support.
- Casual Users Seeking Healthier Lifestyles: Those looking for general wellness improvement without strict dieting.

These groups face challenges such as:

- Time-consuming manual meal logging.
- Lack of personalized diet recommendations.
- Difficulty maintaining motivation over time.
- Insufficient knowledge about balanced nutrition.
- Emotional triggers that affect eating behaviors.

8.2. User Stories

User 1 (U-1):

- As a user, I would like to log my meals by simply taking a photo so that I don't have to manually search and enter each food item.

User 2 (U-2):

- As a user, I would like the app to generate a shopping list based on my meal plan so that I can shop efficiently without missing any ingredients.

User 3 (U-3):

- As a user, I would like to track my mood alongside my diet so that I can understand how my eating habits influence my emotional well-being.

User 4 (U-4):

- As a user, I would like to receive daily motivational tips from the chatbot so that I stay encouraged to follow my nutrition plan.

User 5 (U-5):

- As a user, I would like to earn points and unlock mini-games when I meet my daily nutrition goals so that maintaining a healthy diet feels fun and rewarding.

User 6 (U-6):

- As a user, I want to share my progress on social media so that I can receive support and encouragement from my friends.

User 7 (U-7):

- As a user, I want to get meal recommendations based on my dietary preferences and current health goals so that I don't have to plan meals myself.

User 8 (U-8):

- As a user, I want to receive alerts when I am close to missing my water intake goal for the day so that I stay hydrated.

Additional Technical User Stories:

- (U-9): As a data scientist, I want to visualize the AI model's food recognition accuracy across different cuisines so that I can improve model generalization.
- (U-10): As an end-user, I want to be notified when the AI is unsure about a food item so that I can manually correct it for better tracking.

Requirement No	Requirement Description	Related User Story No
FR-01	The system shall generate daily and weekly meal plans based on the user's goals.	US-7
FR-02	The system shall allow users to log meals through manual search, natural language input, and photo upload methods.	US-1
FR-03	The system shall estimate calories by recognizing food items from uploaded photos.	US-1, US-10
FR-04	The system shall analyze food descriptions entered in natural language by mapping them to the USDA nutrition database.	US-1
FR-05	The system shall track information related to water intake, weight changes, and mood.	US-3, US-8
FR-06	The system shall provide gamification elements by awarding points when users achieve their daily goals.	US-5
FR-07	The system shall store user profile information, including age, gender, goals, and dietary preferences.	US-7
FR-08	The system shall archive users' historical data, such as daily meals, mood entries, weight records, and achievements.	US-3, US-5, US-8
FR-09	The system shall access the mobile device's camera to allow users to take and upload food photos.	US-1, US-10
FR-10	The system shall allow users to interact with each other.	US-6

FR-11	The system shall include healthy and diet-friendly recipes for users.	US-7
FR-12	The system shall allow users to upload and share their own recipes.	US-6

9. Social, Environmental, and Legal Impact

9.1. Societal Benefits

Our diet app supports society by making healthy living more accessible and sustainable. It addresses common challenges such as poor nutrition and obesity through intuitive, easy-to-use tools. For example, users can simply take a photo of their meals to instantly check calorie and nutrient content, or receive quick recipe suggestions based on ingredients they already have at home. This empowers them to make healthier food choices and prepare balanced meals with minimal effort.

The app also uses engaging games and progress tracking to keep users motivated, while mood-based insights help them understand and improve their relationship with food. Over time, this fosters lasting healthy habits. Additionally, by encouraging mindful eating and smarter grocery use, it reduces food waste, helping users save money and contributing to lower national healthcare costs in the long run.

9.2. Economic Constraints

The economic and environmental sustainability of our project depends on the careful management of its energy consumption and computational costs. The primary drivers of these costs are the AI-powered features: the interactive chatbot and the image-based calorie analysis.

Computational Costs and Processing Power: Running AI models, especially for a real-time chatbot and image recognition, requires significant processing power.

Chatbot: Instead of training a large language model from scratch, which is extremely expensive and energy-intensive, we plan to use a pre-trained, efficient model. We will fine-tune this model for our specific diet and wellness tasks. For user interactions (inference), we will choose a model optimized for speed and low computational load to handle many users at once without high costs.

Image Analysis: Similarly, the calorie estimation feature will use a lightweight, pre-trained computer vision model. Processing happens on-demand, meaning we only use computational resources when a user uploads a photo.

Data Storage: The main storage demand comes from user-uploaded images. To minimize costs and environmental impact, our strategy is to process images for nutritional analysis and then immediately discard them, rather than storing them long-term. User profile data and logs are text-based and require minimal storage space. For essential user data like profiles, streaks, and settings, we will use the free tier of a database service like Firebase or Supabase..

Optimization and Sustainability Strategies: To ensure our platform is efficient and environmentally friendly, we will implement several key strategies:

Use of Efficient Models: We will specifically select AI models that offer the best balance of performance and low resource usage.
Serverless Computing: We plan to use serverless infrastructure (like AWS Lambda or Google Cloud Functions). This means we only pay for a computation when it's actually running—for instance, when a user asks the chatbot a question or uploads a photo. This avoids the cost and energy use of an always-on server.
Smart Data Management: Our "process and discard" policy for images will drastically reduce our data storage needs and associated energy consumption.
Cost Optimization Approach: As a student-led project with limited funding, our priority is to use cost-effective or free solutions whenever possible, avoiding high-cost infrastructure or premium services unless absolutely necessary.
Use of Free and Open-Source Tools: We will prioritize free-tier cloud services (e.g., Firebase, Supabase), open-source AI models, and free developer tools to avoid licensing costs.
No Dedicated Hardware Purchases: We will use existing personal devices (laptops, smartphones) for development and testing, eliminating the need for hardware procurement.
Cloud Services on Free/Trial Plans: Hosting and computation will be handled through free-tier or trial accounts (e.g., AWS Educate, Google Cloud for Students).
AI Model Efficiency: Pre-trained lightweight models will be used to reduce processing costs and avoid expensive training processes.
Risk and Mitigation:
Unexpected Costs: If any API usage exceeds free-tier limits, we will switch to alternative free APIs or reduce usage by limiting certain features in testing phases.
Price Fluctuations: Since most tools will be on free tiers, the impact of price changes will be minimal. If needed, we will select backup providers offering free or discounted student plans.
Service Availability: In case a free service becomes unavailable, we will migrate to another free-tier service with minimal downtime.

9.3. Legal and Ethical Compliance

As developers handling sensitive user information, our project is built on a foundation of trust and safety. We are committed to following legal and ethical rules, with a focus on protecting our users.

1. Legal Compliance

Data Protection (GDPR/KVKK): We understand the importance of data privacy laws.
User Consent: We will not collect any data without clear and simple consent. Before using the app, users must agree to a straightforward Privacy Policy.
Data Minimization: We will only collect the data we absolutely need for the app to work (e.g., food entries, streaks). We will not collect unnecessary personal information.
Right to Delete: Users will have the ability to permanently delete their account and all associated data at any time, directly from the app.
Anonymity: The data from uploaded photos will be processed for calorie information and then immediately discarded. We will not store user photos on our servers.

Intellectual Property (IP): To avoid legal issues, we will only use open-source software, libraries, and pre-trained AI models that have licenses permitting their use in projects like ours. All the code we write will be our own original work.
Security Standards: We will follow best practices aligned with ISO/IEC 27001 (Information Security Management) and ISO/IEC 27701 (Privacy Information Management) [12]. All data transfers will use HTTPS/TLS encryption.

2. Ethical Principles

Responsible AI Use: Our AI design and deployment will follow OECD AI Principles [13], EU AI Act ethical guidelines [10], and IEEE Ethically Aligned Design recommendations [11]. These frameworks ensure our AI is transparent, fair, accountable, and respects human rights.
Clear Disclaimer: The chatbot will be programmed to state clearly: "I am an AI assistant, not a doctor. Please consult a healthcare professional for medical advice." It will refuse to diagnose conditions or give medical prescriptions.
Transparency: Users will be made aware that the calorie count from a photo is an estimate and not a lab-accurate measurement. We will be transparent about the limitations of our AI.
No Manipulation: The AI will be designed to be encouraging and supportive, not to manipulate user behavior or promote unhealthy, obsessive habits.
Privacy and Security: Protecting user data is our top priority.
Secure Practices: We will use standard security measures, like using secure connections (HTTPS) and free, trusted services like Firebase for user authentication and database management, which have their own strong security protocols.
Handling Sensitive Data: The "psychological observation" feature is for the user's eyes only. This data will be stored securely, and it will be made clear to the user that this information is private to them and is not monitored by us.

REFERENCES

[1] K. R. Schneider, J. J. Liu, and A. A. Voida, “Retention and Engagement in Mobile Health Apps: A Longitudinal Study of User Behavior,” Journal of Medical Internet Research, vol. 26, no. 1, p. e56897, Jan. 2024. [Online]. Available: https://www.jmir.org/2024/1/e56897

[2] C. R. Maher, K. R. Ferguson, and J. Olds, “When and Why Adults Abandon Lifestyle, Behavior, and Mental Health Apps: Qualitative Analysis of User Experiences,” JMIR mHealth and uHealth, vol. 10, no. 11, p. e38938, Nov. 2022. [Online]. Available: https://researchnow.flinders.edu.au/en/publications/when-and-why-adults-abandon-lifestyle-behavior-and-mental-health

[3] https://www.healthline.com/nutrition/top-iphone-android-apps

[4] https://apps.apple.com/us/story/id1680281199

[5] S. Sardi, A. Idri, J. Fernández-Alemán, and A. Toval, "The effectiveness of gamification in mHealth apps: A systematic review," Journal of Biomedical Informatics, vol. 71, pp. 31-48, 2017.

[6] E. Mattila, P. Saarni, S. Salminen, et al., "Emotional eating, depressive symptoms and self-compassion in university students: A population-based study," Appetite, vol. 68, pp. 66-72, 2013.

[7] G. F. Mateo, M. Granado-Font, A. Ferré-Grau, and J. M. Montaña-Carreras, "Mobile phone apps to promote weight loss and increase physical activity: A systematic review and meta-analysis," Journal of Medical Internet Research, vol. 17, no. 11, p. e253, 2015.

[8] J. M. Rabin and B. C. Bock, "Desired features of smartphone applications promoting physical activity," Telemedicine and e-Health, vol. 17, no. 10, pp. 801-803, Dec. 2011.

[9] L. Dennison, L. Morrison, G. Conway, and L. Yardley, "Opportunities and challenges for smartphone applications in supporting health behavior change: qualitative study," Journal of Medical Internet Research, vol. 15, no. 4, p. e86, 2013.

[10] European Commission. (2024). *Artificial Intelligence Act*. Retrieved from https://digital-strategy.ec.europa.eu/en/policies/european-approach-artificial-intelligence

[11] IEEE Standards Association. (2019). *Ethically Aligned Design: A Vision for Prioritizing Human Well-being with Autonomous and Intelligent Systems* (1st ed.). IEEE Standards Association. Retrieved from https://ethicsinaction.ieee.org

[12] ISO/IEC. (2019). *ISO/IEC 27701:2019 Security techniques — Extension to ISO/IEC 27001 and ISO/IEC 27002 for privacy information management — Requirements and guidelines*. International Organization for Standardization.

[13] OECD. (2019). *OECD Principles on Artificial Intelligence*. Retrieved from https://oecd.ai/en/ai-principles