



Project Management Report

Laura Kotalczyk
Manuel Mühlberger
Joana Silva
Eduardo Pinto
Nils Harrer

Group 4

January 13, 2026

1 Project Vision and Scope

The vision of this project is to develop a more accurate and user-friendly mobile application for tracking daily nutritional intake by combining visual and spoken input through AI, addressing the common pain points of current calorie-tracking tools, such as time-consuming manual entry, limited context in meal logging as well as calorie underestimation by existing AI-enabled calorie trackers.

Many users struggle to track self-cooked meals, mixed dishes, or food eaten outside the home (e.g., at restaurants or similar), because existing apps require users to manually search, estimate, or break down ingredients. Nowadays, some apps do leverage AI-enabled image recognition to streamline calorie tracking, promising greater ease and speed, however, these tools often fall short in accuracy, frequently over- or underestimating actual calorie count, which ultimately undermines the reliability of even the most convenient logging methods.

To address this gap, our app introduces an additional input to the AI-enabled estimation approach: by simply taking a photo of a meal and adding a brief voice description, such as “pancakes with milk, no sugar, 3 eggs, 2 cups of flour, topped with some maple syrup”, users can instantly generate a more accurate estimate of calories and nutrients. By expanding the vision modality with contextual natural language input, the app harnesses modern AI technologies to deliver not just convenience, but more accurate, data-driven insights. The goal is to reduce user effort and improve accuracy through its multi-modal approach, particularly for home-cooked and complex meals that are frequently misclassified or under-estimated in existing tools. The user can also browse a simple calendar view to display daily food intake and progress towards goals. Insights are provided to the user as well, so he can gain information on metrics such as average meal quality, longest logging streaks and many others.

This solution aims to make nutrition tracking simpler, smarter, and more accessible, helping users stay aware of their intake and make informed choices, without requiring expertise meticulous manual logging.

1.1 Key Deliverables and Defined Boundaries

1.1.1 Key Deliverables

Deliverable	Description
Mobile App Prototype	A basic app (Android/iOS or web-based) that lets users: take a photo of a meal and record a voice note (e.g., "pancakes with milk, no sugar, 3 eggs").
Voice-to-Text Feature	The app automatically turns the voice description into text using an AI audio transcription model such as Whisper.
AI Nutrient Analyzer	Uses a Vision Language Model (VLM) (e.g., GPT-4V or Qwen-VL) to analyze the photo and text, and returns estimated calories and nutrients (e.g., "250 kcal, 12g protein").
Simple Dashboard	Shows daily calorie and nutrient intake in a calendar view.
Test Data & Results	Uses 10–20 real example meals (e.g., homemade pancakes, homemade spaghetti Bolognese) with known nutrition values to test how accurate the app is.
Benchmarking Results	Uses the ground-truth real example meals for benchmarking the prototype against other competitor apps, evaluating its accuracy in estimation of nutrients and calories.
Project Report	A written report illustrating the design and functioning of the app, test results, and lessons learned from the project, in particular referring to the use of GenAI.

Table 1: Project Key Deliverables

1.1.2 Project Boundaries

To keep the project focused and manageable, we decided on the following project boundaries as well as mandatory and optional app features.

In Scope

- User can record one photo and voice note per meal
- Use of open-source AI models for voice transcription and nutrient estimation (e.g., OpenAI API, Hugging Face, available ones on GitHub)
- Output includes estimated calories and macronutrients (carbs, protein, fat)
- Simple, modern and intuitive user interface
- Testing using 10–20 example meals with known nutritional values (manually calculated)
- User has the possibility to see real-time insights considering his meal tracking activities

Out of Scope

- Support for multiple languages or complex dishes with many unknown ingredients
- Integration with wearables or fitness trackers
- Integration of a user's training activities to increase available calories
- Nutrient Database Integration for more accurate results
- Possibility for data import/export

2 Project Timeline & Milestones

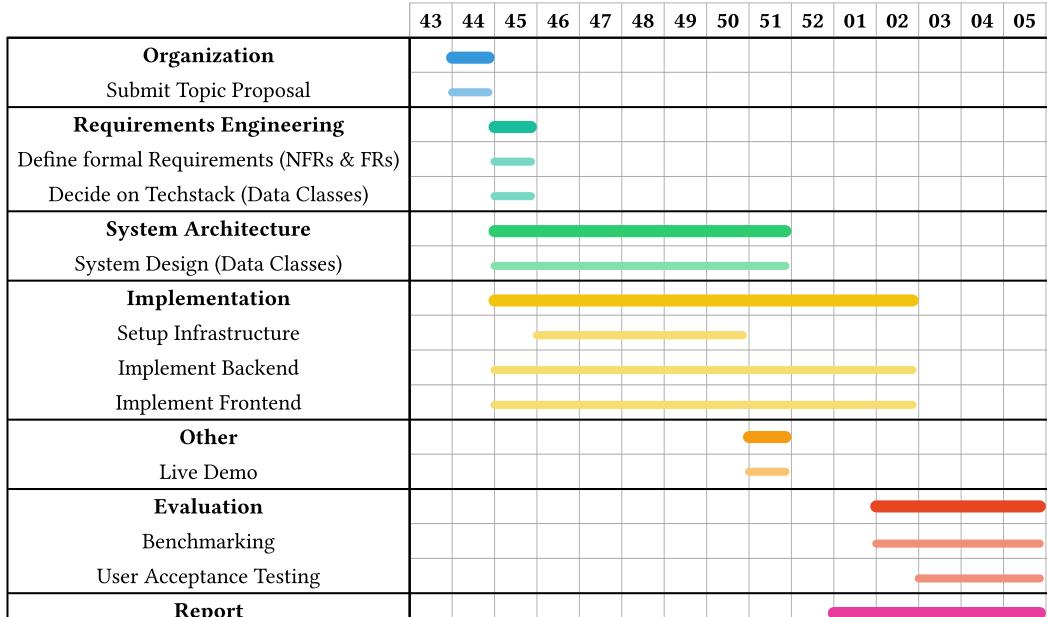


Figure 1: Project Timeline and Milestones

3 System Architecture and Technical Design

3.1 High-Level System Design

With respect to system design, we decided on a proxy-enabled server-client architecture, involving three different instances of servers, all of whom are locally hosted. First of all, there is the main "physical" server (1) acting as the database storing all user data. Secondly, server two (2) runs a dockerized reimplementation of OpenAI's Whisper model [2], that is used for transcribing a user's recorded voice note in the AI-assisted meal logging workflow. Thirdly, server three (3) is a dockerized proxy that forwards ...

By hosting all user data on our own locally managed server, we eliminate reliance on external service providers for storing sensitive user information. This allows us to maintain full control

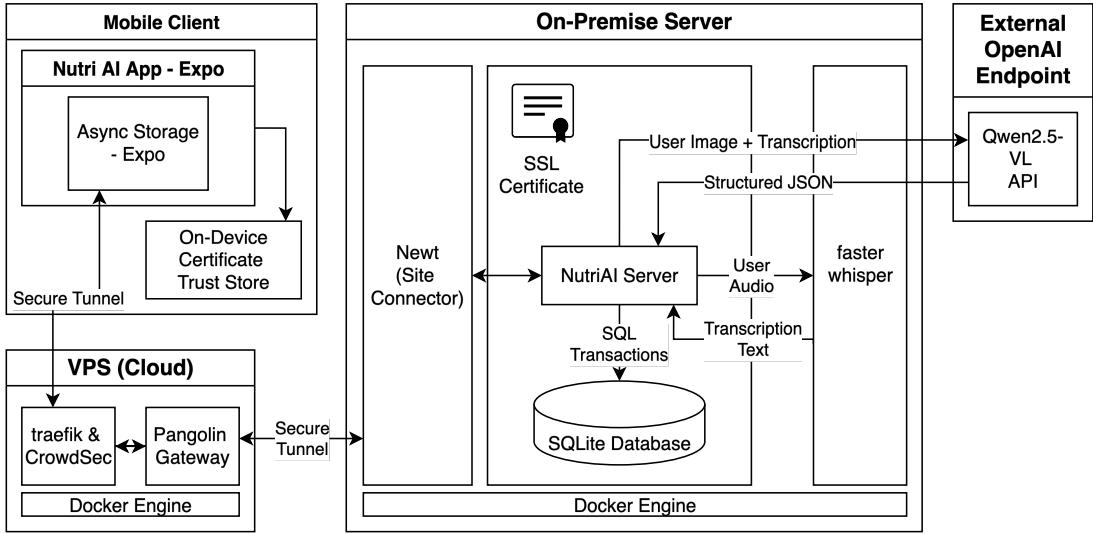


Figure 2: High-Level System Architecture of NutriAI

over the data, ensuring enhanced privacy and security. Furthermore, our locally hosted implementation of OpenAI’s Whisper model ensures that user voice data is never shared with external systems or used for AI training. Instead, audio recordings are securely processed within our private infrastructure, and only the resulting transcription, paired with the user’s meal photo, is sent to the externally hosted VLM. This approach guarantees that no sensitive voice information is exposed, and helps maintain privacy protections. Finally, our proxy solution,

- wie werden Passwörter abgelegt (mit Salt + Hash?)

3.2 Core Technical Components

Since we were aiming at building a mobile app that runs cross-platform, we opted for Expo, after some research on possible options. Expo is a full-stack React Native framework that offers a rich ecosystem of pre-built components and tools, known for its simple setup, fast prototyping, and smooth development workflow [1]. Its accessibility made it a strong fit for our team, particularly given the varying levels of experience among team members. In the following paragraphs, we will elaborate on each of our system’s components from a technical perspective, pointing out its role and interactions in the overall system.

3.2.1 Client

Frontend

Backend

3.2.2 Proxy

3.2.3 Server

- what is stored on the server and how

Audio Transcription with Whisper

Nutriant Estimation with Vision Language Model (VLM) Multi-modal Input (Transcription + Image)

3.3 Scalability and Security Considerations

4 Project Management Methodology

To organize our team and the development process of the mobile application properly, we decided on a "SCRUM-alike" agile work mode. We proceeded in increments of one-week long sprints, agreeing on tasks each of the team members should finish until the next week. During our weekly meeting slot in person, we then met to discuss our progress on the respective tasks, next steps and potential impediments. Apart from that we added new items and bug fix tickets to the product backlog, prioritized the product backlog and assigned tasks with respective deadlines to team members.

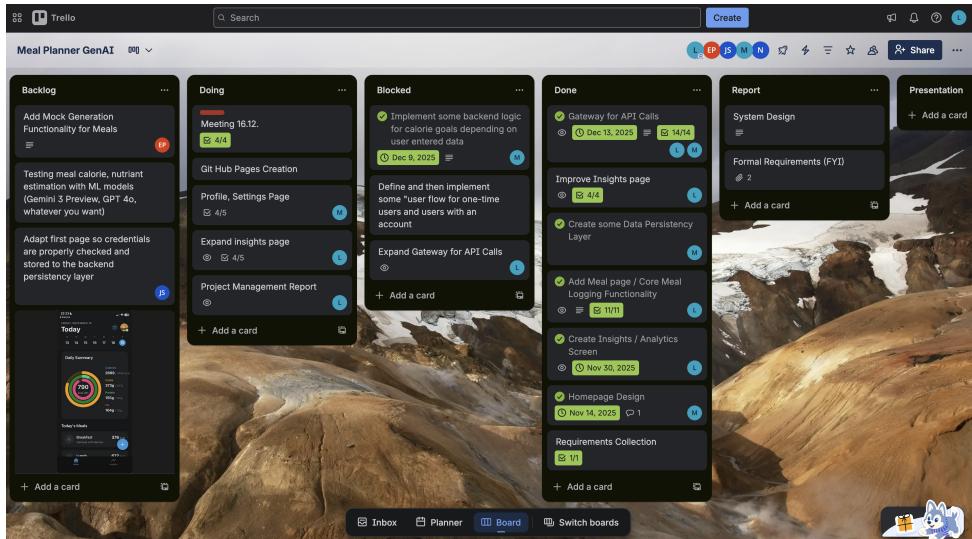


Figure 3: Project Management Board used for Development of NutriAI

To keep track of the tickets and overall tasks concerning our project, we used [Trello](#), a project management tool that uses a flexible, Kanban-style system of boards, lists, and cards to help individuals and teams organize tasks, workflows, and ideas, making it easy to track progress from "to-do" to "done". For a screenshot taken from our project's board, refer to Figure 3. Apart from using a project management tool, we also created a WhatsApp group with all the team members to agree on meeting dates, keep us updated and discuss immediate questions or provide advice quicklier.

4.1 Prompt Engineering Strategies

Key techniques used (e.g., role prompting, chain-of-thought), refinement process, and impact on GenAI performance

5 Team roles, responsibilities, and how collaboration was managed

6 Current Progress and Future Plans

Progress Status and Roadmap to Completion Current status (completed/in-progress/pending), key achievements, risks, and final steps

7 Project Pitch Video (Attached)

8 Usage of GenAI during the Document's Creation

Generative AI was used during the writing process to help with brainstorming and organizing initial notes. It was also used to assist with debugging certain problems occurring with LaTeX.

References

- [1] Inc. 650 Industries. *Everything you need to build apps*. URL: <https://expo.dev/> (visited on 01/06/2026).
- [2] LinuxServer.io. *linuxserver/faster-whisper*. URL: <https://docs.linuxserver.io/images/docker-faster-whisper/> (visited on 01/06/2026).