

LowKcal MVP Prototype Documentation

Prototype Specification

1. Strategy

The LowKcal MVP prototype aims to address the increasing consumer demand for health-conscious dining options by establishing a user-centric platform. It tries to provide an intuitive and seamless experience for users to explore, customize, and order low-calorie meals from a thoughtful selection of restaurants.

2. Prototype Objectives

The primary objectives of the LowKcal MVP prototype are multifaceted.

Firstly, it seeks to empower users with the ability to effortlessly discover, personalize, and procure meals tailored to their dietary preferences and requirements.

Secondly, it aims to streamline administrative processes for restaurant owners and system administrators, facilitating efficient management of restaurant and meal information.

Lastly, the prototype aims to facilitate smooth order processing and fulfillment for both users and restaurants, ensuring a seamless dining experience.

3. Assumptions and Constraints

The development of the LowKcal MVP prototype operates under certain assumptions and constraints. It presupposes a keen interest among users in selecting low-calorie meal options, which informs the platform's focus on health-conscious dining. Additionally, the prototype acknowledges limitations in backend functionality during the MVP phase, necessitating a prioritization of essential features and functionalities.

4. End-of-Iteration Criteria

The success of the prototype iteration will be evaluated based on predefined criteria. These include the implementation and functionality of core features such as restaurant listing, meal customization, and order processing. Furthermore, the prototype aims to achieve operational viability by ensuring basic user interactions such as meal configuration and order placement are fully functional.

5. Prototype Scope

The scope of the LowKcal MVP prototype is decorated by its focus on foundational features essential for delivering a pleasant user experience. It implies functionalities aimed at facilitating meal discovery, customization, and ordering, while also prioritizing simplicity and usability across all user touchpoints.

6. Design Units

The design of the LowKcal MVP prototype is structured around distinct user interfaces tailored to supply the unique needs and interactions of customers, restaurants, and system administrators. These interfaces are meticulously crafted to ensure intuitive navigation and seamless interaction, fostering a positive user experience.

7. Major Functions

Meal Configurator

The meal configurator feature empowers users with detailed control over customizing their meal compositions. Through an intuitive interface, users can selectively add or remove ingredients, adjust quantities, and monitor calorie intake in real-time. A dynamic summary panel provides users with inclusive information regarding the nutritional composition of their customized meal, enhancing transparency and informed decision-making.

Restaurant Filtering

The restaurant filtering functionality serves as a foundation of the user experience, enabling users to discover dining locations based on various criteria. Utilizing sophisticated sorting algorithms, users can filter restaurant listings according to factors such as ratings, proximity, and personalized recommendations. This feature ensures that users can effortlessly find dining options that align with their preferences and requirements, enhancing overall satisfaction.

Order QR Code

Upon placing an order, users are provided with a unique QR code that serves as a digital token for order retrieval at the restaurant. This QR code encapsulates essential order details, including the order ID, meal summary, and payment information, streamlining the pickup process. By leveraging QR code technology, the prototype enhances operational efficiency for restaurants while offering users a seamless and convenient ordering experience.

Prototype Evaluation Report

1. Purpose

The prototype evaluation report serves as an inclusive assessment of the prototype's performance, identifying areas of strength and areas for improvement. It aims to distill insights gleaned from prototype evaluation activities, informing subsequent iterations and enhancements.

2. Observations

Through rigorous evaluation, several observations were noted regarding the prototype's functionality and user experience. These observations encompassed various facets, including user interaction, feature usability, and system performance.

3. Recommendations

Based on identified observations, actionable recommendations were formulated to address limitations and enhance the prototype's overall efficacy. Recommendations included refining the meal configuration interface to improve user control and feedback mechanisms, optimizing restaurant filtering algorithms to deliver more accurate and relevant recommendations, and expanding backend functionality to support additional features such as user authentication and profile management.

4. Form, Screen, and Report Layout Modifications

In light of evaluation findings and recommendations, modifications to form, screen, and report layouts were proposed to enhance usability and functionality. These modifications encompassed various aspects, including interface refinement, algorithm optimization, and backend integration. By implementing these modifications, the prototype aims to improve user engagement, streamline user workflows, and deliver a more cohesive and compelling user experience.

LowKcal MVP Architecture

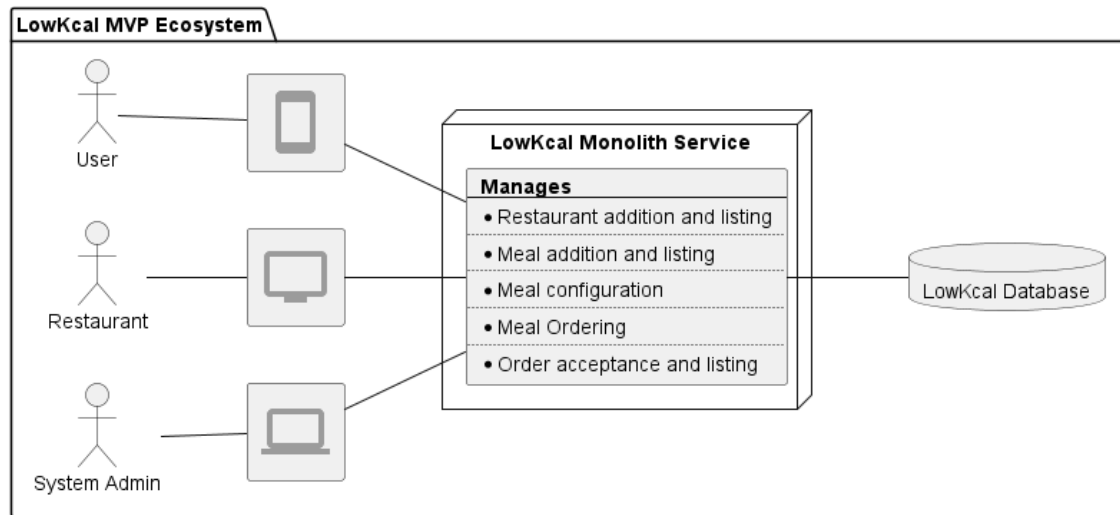


Figure 1 - LowKcal MVP HIGH LEVEL ARCHITECTURE DIAGRAM

1. System Overview

The LowKcal MVP prototype is supported by a monolithic backend service, known as the "LowKcal Monolith Service," designed to centralize core functionalities. This service interacts with a LowKcal Database to store and retrieve essential data related to restaurants, meals, and orders, facilitating seamless data management and access.

2. Key Points

The architecture of the LowKcal MVP prototype prioritizes rapid development and simplified deployment, leveraging a monolithic approach to streamline initial development efforts. While acknowledging potential scalability and maintainability concerns associated with monoliths, the prototype aims to mitigate these challenges through modular design principles and future-oriented considerations.

3. Backend Functionality

The LowKcal Monolith Service serves as the core of the prototype's backend functionality, encompassing critical features such as restaurant and meal management, order processing, and user interaction. This centralized service facilitates efficient data processing, seamless communication between system components, and robust backend operations.

4. Database Integration

Integration with the LowKcal Database forms the foundation of the prototype's data management strategy, ensuring secure and efficient storage of essential data entities. The database schema is meticulously designed to accommodate diverse data requirements, enabling seamless retrieval and manipulation of restaurant, meal, and order information.

Additionally, stringent data validation mechanisms are implemented to uphold data integrity and security standards.

LowKcal MVP Core Requirements

System Admins:

1. Restaurant Listing: System admins can list restaurants.
2. Restaurant Management: System admins can update or delete a restaurant.
3. Meal Listing: System admins can list meals in a restaurant.
4. Meal Management: System admins can update or delete a meal.
5. Restaurant Account Creation: System admins can create accounts for restaurants.

Restaurant:

1. Order Acceptance: Restaurants can accept or refuse an order.
2. Order Listing: The app shows the accepted order list to the restaurant.

User:

1. Restaurant Search: Users can search for a restaurant.
2. Meal Search: Users can search for a food.
3. Meal Configuration: Users can configure a meal.
4. Order Placement: Users can place an order.
5. Order Canceling: Users can cancel an order if the restaurant has not accepted it yet.
6. Register: Users can register.
7. Login: Users can log in.