Habib University

Computer Science — Kaavish CS-491

SRS2: Block Diagram + Wireframe

SehatGuru

Team Members:

Hammad Malik (hm08298@st.habib.edu.pk)
Ahtisham Uddin (au08429@st.habib.edu.pk)
Arsal Jangda (aj08514@st.habib.edu.pk)
Sameer Kamani (sk08109@st.habib.edu.pk)

Date: November 12, 2025

Contents

1 System Design and Architecture

1.1 System Block Diagram

Figure 1 shows the high–level architecture of SehatGuru. The purpose of this diagram is to show how the mobile application, backend, AI subsystem, databases, and external services interact with each other.

1.2 Overview

SehatGuru is a Mobile App that interacts with its Backend. The Backend coordinates two main subsystems: **datastores** (for user data, meal logs and images) and **AI subsystems** (food recognition and chatbot). External services (Gemini API, Firebase Authentication, FCM) provide third-party features. The API Gateway is the boundary between client traffic and internal services.

1.3 Components

Mobile App: The app the user installs. It shows the dashboard, accepts meal images and manual entries, opens the chat interface, and receives push notifications.

API Gateway: HTTP endpoint layer that accepts requests from the mobile app and forwards them to the correct backend service. It centralizes routing, input validation.

Logic Layer / Backend Services: Implements business logic: logging a meal, converting portions to grams, calculating totals, making calls to AI subsystems, and returning responses to the app.

AI Subsystems:

- **Food Recognition:** Receives images and returns the best matching dish label.
- **Chatbot (SehatGuru):** Generates text responses for user questions and meal plan requests. It uses the Gemini API to generate responses.

Datastores:

- **User Profile & Auth Store:** Stores profile, goals, preferences and authentication identity references.
- **Food Log & Analytics Store:** Stores daily meal entries, timestamps, computed calories/macros, and aggregated analytics used by the dashboard.
- **Nutritional DB:** Stores nutrition facts (per 100 g) and portion presets for all supported dishes (200 items).
- **Image Storage:** Stores uploaded user images.
- **Chatbot Knowledge Base:** Small retrieval store with dish metadata, common Q&A and past conversation memory for personalization.

External Services: **Gemini API:** Provides high-quality LLM responses for the chatbot.

Firebase Authentication: Handles signup, login, OAuth and token validation.

FCM (Firebase Cloud Messaging): Sends push notifications to mobile clients.

Notification Service: Sends scheduled or event-driven notifications (e.g., logging reminders, meal plan ready).

1.4 Relationships and data flow

Below we describe how components connect and what data flows between them. Each arrow in the block diagram represents a request/response pair or a data push.

Client → API Gateway

- The Mobile App sends authenticated HTTP requests to the API Gateway for actions such as: login/signup, log meal, request chat, fetch analytics, and update profile.
- Authentication tokens (Firebase JWT or similar) are attached to each request and validated by the Gateway or Backend.

API Gateway → Backend Logic Layer

- The Gateway forwards requests to specific backend endpoints (e.g., `/log-meal`, `/chat`, `/user/profile`).
- The Backend performs validation, business logic, and calls other internal modules (AI subsystems, DB) as needed.

Backend ↔ Datastores

- **Read/Write Meal Logs:** When a meal is logged (manual or confirmed image result), the backend writes an entry to Food Log & Analytics and updates aggregated metrics used by the dashboard.
- **User Profile Access:** The backend reads/writes profile fields (weight, goals) to compute recommendations and personalize chatbot responses.
- **Nutritional Lookups:** The backend queries the Nutritional DB to compute calories/macros for selected portions.
- **Image Storage:** Images uploaded for recognition are persisted (temporary or permanent) and a storage URL is saved in the meal log entry.

Backend ↔ AI Subsystems

- **Food Recognition:** The backend sends the image (or storage URL) to the Food Recognition service which returns a {dish_id, confidence} pair. The backend passes this result to the mobile client and requires user confirmation before final logging.
- **Chatbot:** The backend prepares a chat context (user profile, recent logs, relevant dish facts from the Nutritional DB) and calls the Chatbot subsystem. The chatbot may itself call the Gemini API (external) if the answer requires LLM reasoning.

AI Subsystems ↔ External Services

- The Chatbot subsystem forwards LLM queries to Gemini API and receives generated text which is then post-processed (safety checks and personalization) before returning to the user.

Notification Service → Mobile App (via FCM)

- Backend schedules or triggers notifications which are sent through FCM. These may be push reminders for logging, meal plan ready notices, or progress nudges.

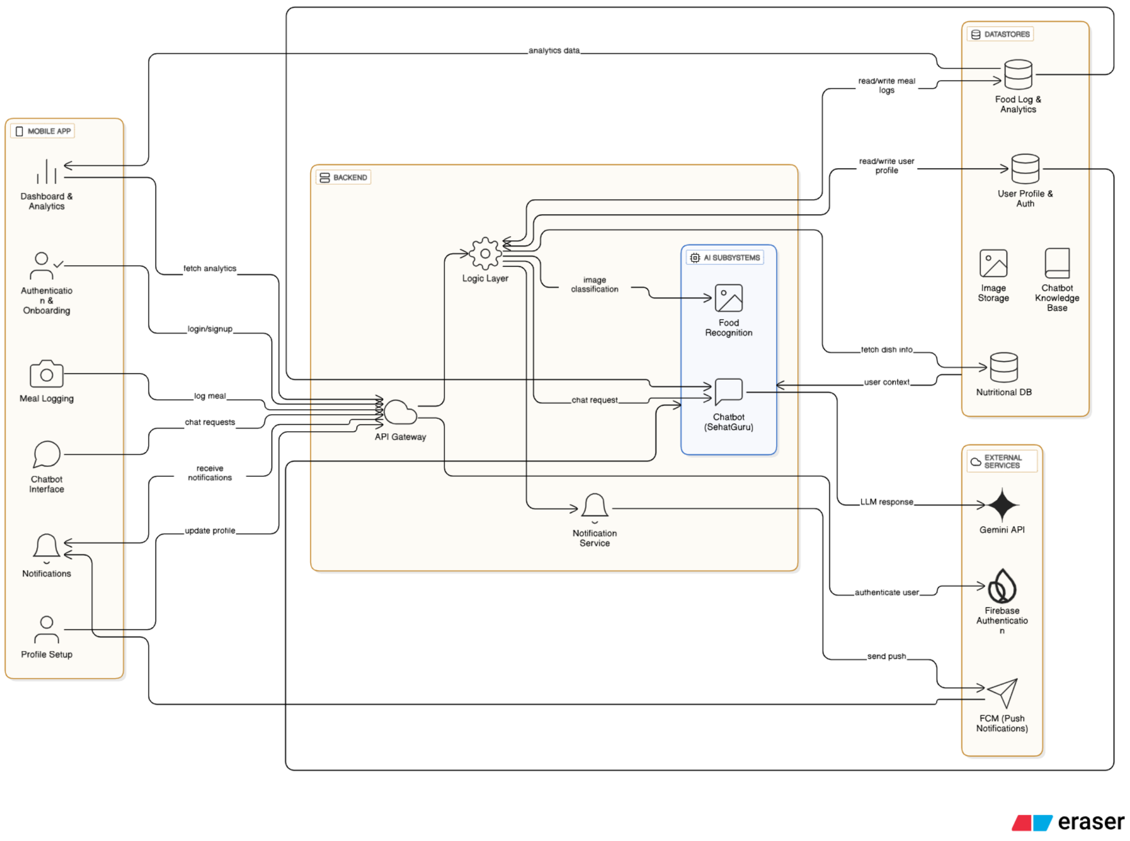


Figure 1: System Block Diagram of SehatGuru

2 Project Plan

2.1 Objectives

- Build a nutrition tracking app for Pakistani meals.
- Allow users to log meals manually or through image recognition.
- Provide their daily/weekly nutritional insights.
- Allow users to seek help/advice and create meal plans using an AI–based chatbot.

2.2 Team Roles

Team Member	**Responsibility**
Hammad Malik and Arsal Jangda	App infrastructure, Frontend and Backend, Connecting services
Ahtisham Uddin and Sameer Kamani	Food Recognition feature, Chatbot integration
Everyone	Data collection and verification

2.3 Timeline

Week 13–14	Data Collection and Classification + Literature review
Week 14–15	Frontend and Backend Integration(Prototype level)
Week 15–16	Testing and adding feedback accordingly

2.4 Risks and Mitigation

- **Risk:** Image recognition accuracy may be low. **Mitigation:** Allow manual logging fallback.
- **Risk:** Chatbot may return inconsistent responses.
  Mitigation Ideas:
  - Validate output by providing context to the prompt, so it tailors its response according to the nutritional needs of the user.
  - Store fixed responses in the case of basic questions.
- **Risk:** Time constraints. **Mitigation:** Prioritize MVP first: food recognition + simple chatbot.

3 Wireframes

The wireframes represent the screens and user flow of the SehatGuru mobile application. They were created to visualize how users will interact with key features such as meal logging, AI chatbot, and dashboard tracking.

These wireframes illustrate:

- Login and onboarding flow
- Dashboard and analytics
- Meal logging (image recognition and manual search)
- Chatbot interaction and meal planning
- Profile and account management

The wireframe images below provide an overview of all primary screens, while the interactive Excalidraw version (linked below) shows the complete flow and screen-to-screen connections using arrows.

Click below to view the interactive wireframe flow:
`https://excalidraw.com/#json=VqrWsdaMk01yj-clSxpxL,DYbChIRpZ94nLvcFNsnLAg`

The wireframes helped us validate the interface structure and ensure smooth navigation across all modules of the application.

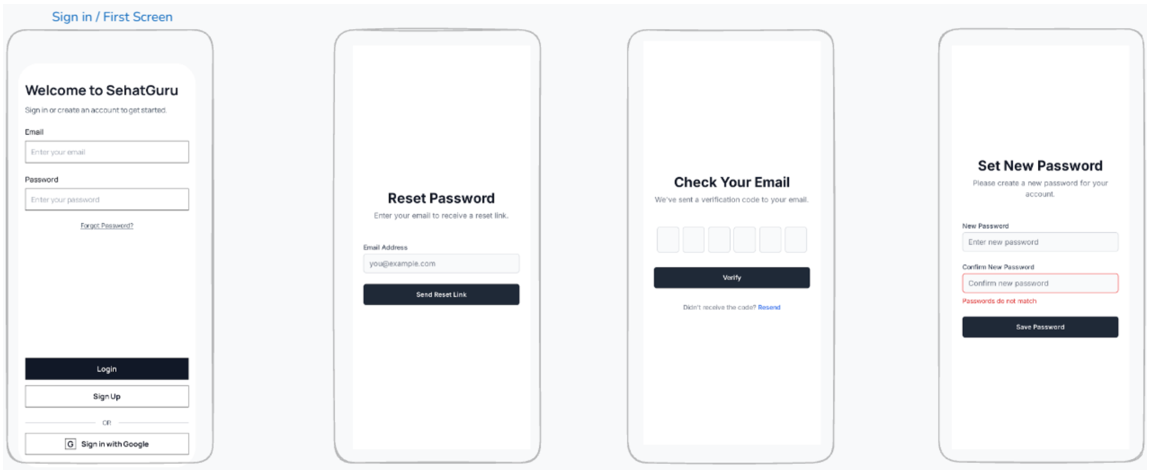
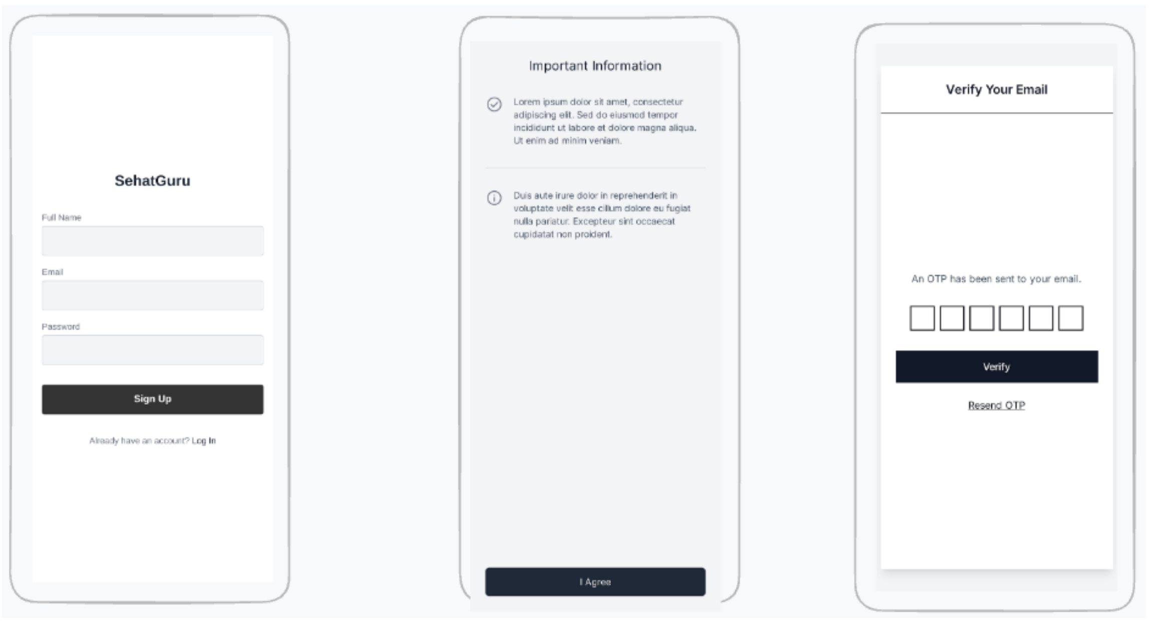


Figure 2: Login and Password Recovery Screens

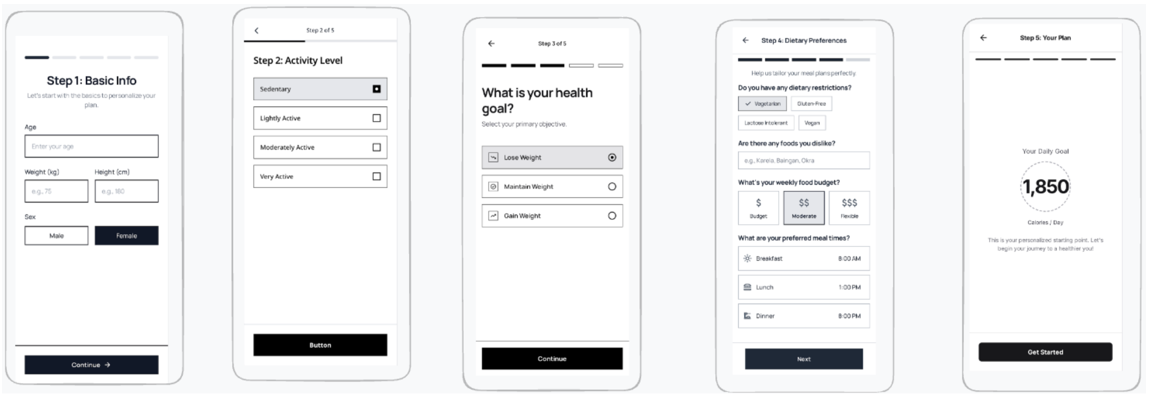


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident.

An OTP has been sent to your email.

Figure 3: Signup, Terms, and Email Verification Screens



Let's start with the basics to personalize your plan.

Enter your age

e.g., 75

e.g., 180

Select your primary objective.

Help us tailor your meal plans perfectly.

e.g., Karela, Baingan, Okra

This is your personalized starting point. Let's begin your journey to a healthier you!

Figure 4: User Profile Setup and Goal Selection Screens

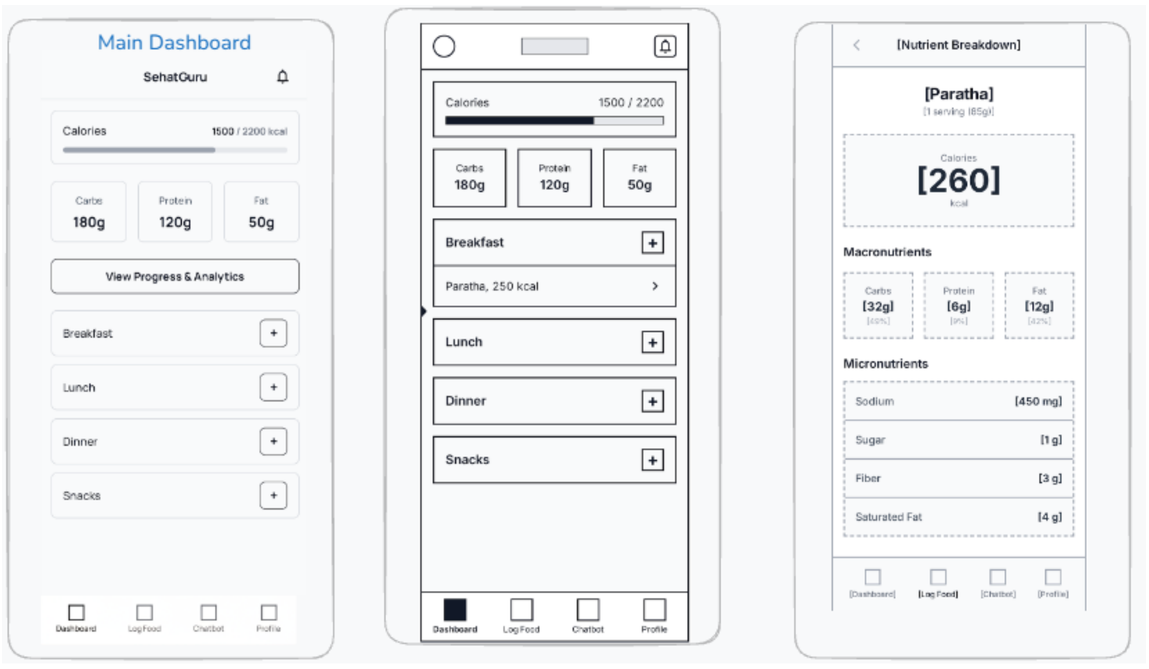


Figure 5: Main Dashboard and Nutritional Breakdown Screens

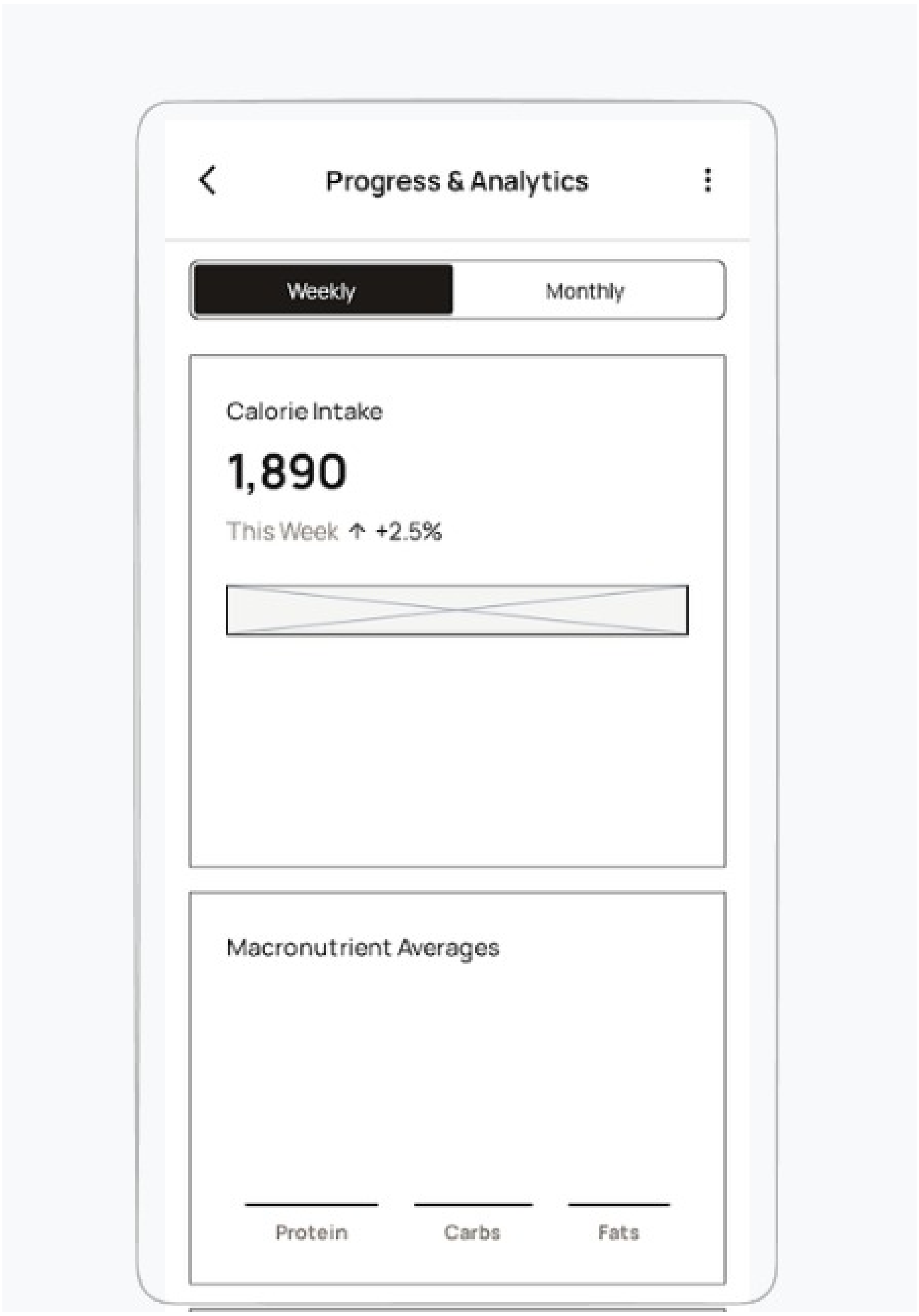


Figure 6: Progress and Analytics Overview Screen

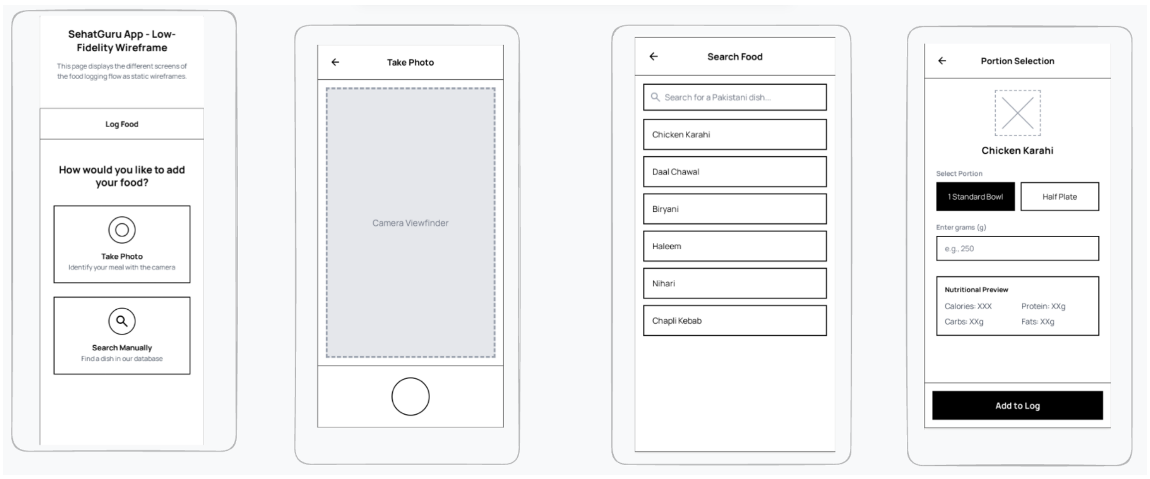


Figure 7: Food Logging Flow: Capture, Search, and Portion Selection Screens

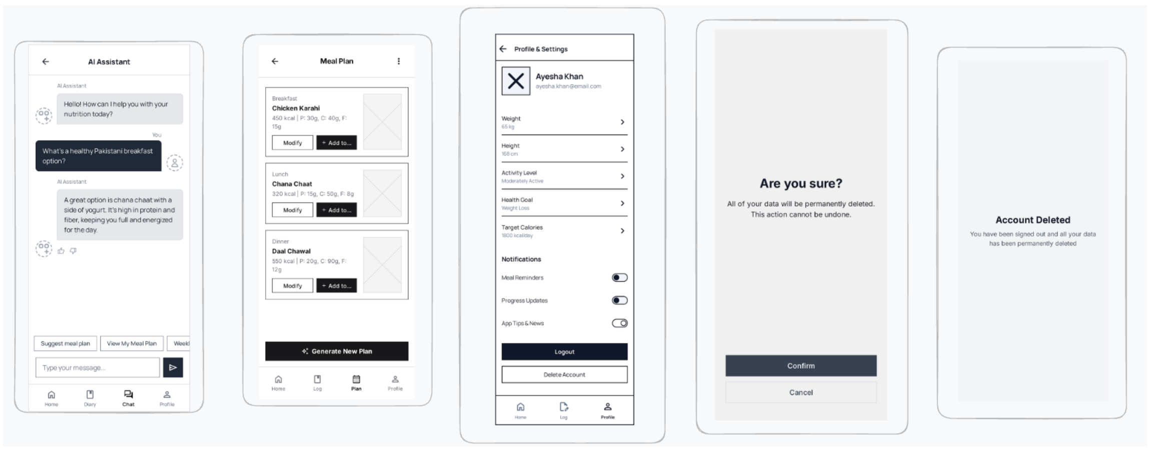


Figure 8: Chatbot, Meal Planning, Profile, and Account Management Screens