Habib University
shaping futures

Habib University

Computer Science | Kaavish CS-491

Software Requirements Specification

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 5, 2025

Contents

1 Introduction

1.1 Purpose

SehatGuru is an intelligent nutrition tracking and advisory system designed for health-conscious desi individuals who want help maintaining their diets. It allows users to log their meals, either manually or through image recognition, and provides insights on calories and macronutrients.

With a database of over 200 traditional Pakistani dishes, SehatGuru offers personalized recommendations based on user preferences and goals. The app also features an AI-powered chatbot that provides nutritional advice and generates daily and weekly meal plans. SehatGuru helps users make informed eating decisions and stay on track with their health goals.

2 Scope

2.1 In-Scope Features

The following features are included in the SehatGuru application:

- **User Registration and Profile Setup:** Users can register, set up their profiles, and input personal details such as weight, height, dietary goals, and target calories.
- **Meal Logging:** Users can log meals manually or through CV based food recognition.
- **Food Recognition:** AI-driven food classification system for recognizing traditional Pakistani dishes from photos.
- **Nutritional metrics:** showing calories, macronutrients, and micronutrients for over 200 traditional Pakistani dishes.
- **Timeline View / Dashboard:** Visualization of daily meals in a timeline format to help users track their eating patterns and nutritional progress.
- **AI-Powered Chatbot:** A chatbot powered by the Gemini API that provides dietary advice and answers nutrition-related queries.
- **Personalized Meal Planning:** Generation of customized daily and weekly meal plans based on user preferences and logged meals.

2.2 Out-of-Scope Features

The following features are excluded from the initial version of SehatGuru:

- **Exercise Tracking:** Real-time tracking of physical activities such as workouts or steps is not included.
- **Desktop Application:** The app will only be available on mobile platforms (iOS and Android).
- **Integration with Wearables:** Integration with fitness devices or wearables (e.g., Fitbit, Apple Watch) will not be included.
- **Multilingual Support:** The app will support only one language (e.g., English) initially, with no multi-language support in the first release.

3 Functional Requirements

3.1 User Authentication and Profile Management

- The system shall allow users to sign up by entering their full name, email address and password.
- The system shall allow users to sign up using OAuth providers (e.g., Google) in addition to email/password.
- The system shall allow users to sign in using their registered credentials.
- The system shall use **Firebase Authentication** for secure login and signup processes.
- The system shall allow users to verify their email address after registration via a verification email.
- The system shall provide a mechanism for users who have forgotten their password to reset it via their email address.
- The system shall send a password reset email containing a unique, time-limited link/token.
- The system shall allow users to set a new password after following the reset link and confirming the new password.
- The system shall display appropriate success and error messages (e.g., “Password has been updated.”, “Email address not found.”, “Reset link is invalid or has expired.”).
- The system shall allow users to update their profile information including weight, height, activity level, fitness goals, target calories, and dietary preferences.
- The system shall store each user’s current and target calorie goals and related profile fields.
- The system shall provide an option for users to delete their account and associated personal data permanently.

3.2 User Profile and Health Goal Setup

- The system shall guide users through an initial setup flow upon first login to capture demographics and goals.
- The system shall collect user-provided inputs: age, weight (kg), height (cm), sex, activity level (Sedentary/Light/Moderate/Active), and health goal (Lose/Gain/Maintain).
- The system shall collect user preferences including dietary restrictions, disliked foods, budget level, and typical meal times.
- The system shall calculate system-internal daily calorie and macronutrient targets (BMR/TDEE-based) from the provided profile and store them.
- Post-condition: upon completion of setup the user shall be directed to the main app dashboard.

3.3 Nutritional Database

- The system shall provide a searchable database containing at least 200 Pakistani dishes with predefined nutritional information per standard unit (per 100 g).
- The system shall store for each food item, at minimum: Calories, Carbohydrates (g), Protein (g), and Fat (g).
- The system shall store portion presets for each dish (e.g., 1 Standard Bowl, 1 Cup, 1 Plate) with associated gram weights.
- The system shall mark a **Top 20 Pakistani Dishes** subset to be used by the camera identification feature.
- The system shall allow administrators to update and maintain the nutritional database entries.

3.4 Food Logging (Camera Identification)

- The system shall allow users to capture a new photo or upload an existing image to identify a food item.
- The system shall process uploaded/captured images via a server-level food classification model.
- The classification model shall attempt to identify a single dish from the Top 20 subset and return the best single match when confidence is high.
- The system shall display clear error messages for cases such as dish not recognized.
- The system shall require the user to confirm or correct the identified dish before logging.
- Post-condition: on successful identification and confirmation, the user shall be navigated to the Portion Selection screen.

3.5 Food Logging (Manual Search)

- The system shall allow users to search the full dish database by text query (e.g., “chicken karahi”).
- The system shall return a list of matching food items or display a “No results found” message when none match.
- Post-condition: on selection, the user shall be directed to the Portion Selection screen.

3.6 Portion Size Selection

- The system shall require users to specify the quantity consumed before logging a selected food item.
- The system shall accept preset quantity selections (e.g., 1, 1.5, 2) combined with portion type (Standard Bowl, Cup, Plate), or accept a direct gram input (e.g., 250 g).
- The system shall compute the nutritional values for the selected portion and add the logged item to the user’s daily log upon confirmation.

3.7 Nutritional Calculation

- The system shall calculate estimated calories and macronutrients for a logged item based on the food database entry and the user-specified portion or grams.
- The system shall display in the user's log: estimated total calories, carbohydrates (g), protein (g), and fat (g).
- The system shall include the calculated values in daily/weekly totals and analytics.

3.8 Food Recognition and Classification Implementation Details

- The system shall run image classification at the server level to balance inference speed and client performance.
- The classification model family may include Convolutional Neural Networks (CNNs) and transformer-based image models; the final selection shall balance accuracy, latency, and compute cost.
- The system shall provide a fallback UI that allows manual selection when the classifier is uncertain or the dish is not in the Top 20.

3.9 AI Chatbot and Nutritional Assistance

- The system shall integrate a conversational chatbot ("SehatGuru") powered by the **Gemini API** and a Retrieval-Augmented Generation (RAG) database containing dish metadata and user-specific records.
- The system shall provide a text-based chat interface enabling users to ask nutrition-related queries and receive LLM-generated responses.
- The chatbot shall use the user's health profile, persisted preferences, and historical meal logs to produce context-aware answers.
- The system shall log chat interactions and user feedback for personalization and model refinement.
- The chatbot shall generate culturally-aware advice prioritizing traditional Pakistani meals and ingredients by looking at the dishes dataset of 200 Pakistani dishes database.
- The chatbot shall allow users to provide feedback on individual responses so it can remember his preferences.

3.10 Meal Planning and Meal Prep

- The system shall generate personalized daily and weekly meal plans via the chatbot, aligned to the user's calorie targets and dietary preferences, by selecting from the predetermined database of 200 Pakistani dishes to provide an optimal balance of macro- and micronutrients for the user's goals.
- The system shall allow users to preview, accept, modify, and add AI-generated meal plans directly into their food log or calendar.
- The system shall display aggregated calories and macronutrients for each planned meal and for the entire day/week.
- The system shall allow the user to replace or modify any dish in a generated plan; upon such changes, the chatbot shall automatically re-evaluate and regenerate the affected meal plan to maintain nutritional balance and adherence to the user's targets.

3.11 AI Memory and Personalization

- The system shall persist user-specific preferences stated in profile or during chat interactions (e.g., dislikes, allergies, preferred meal times) and make them available to the chatbot
- The chatbot shall adapt suggestions over time using persisted preferences and historic logs to refine future recommendations.
- The system shall periodically analyze recent logs (e.g., last 7 days) against goals to produce proactive, progress-based feedback.
- The system shall incorporate explicit user feedback on chatbot responses into personalization logic.

3.12 Progress Dashboard and Analytics

- The system shall provide a centralized dashboard displaying real-time nutritional intake against daily goals (calories and macronutrients).
- The dashboard shall list logged items organized by meal (Breakfast, Lunch, Dinner, Snacks) and show totals (e.g., "1500 / 2200 kcal").
- The system shall provide historical progress reports and charts (weekly/monthly) for calorie trends, macronutrient averages, and weight history.
- The system shall estimate time-to-goal for weight targets based on logged calorie trends and user progress.
- The system shall allow users to view summaries such as "You hit your protein goal X times this week".

3.13 Personalized Notifications

- The system shall send push/in-app notifications for logging reminders when a user misses typical meal logging windows.
- The system shall send motivational messages based on progress (e.g., streaks, milestones).
- The system shall send inactivity alerts when the user has not used the app for a configurable period (e.g., 3 days).
- The system shall notify users when AI-generated meal plans or meal-prep schedules are available or updated.

3.14 Security and Data Management

- The system shall store user data securely in Firebase Firestore and Firebase Cloud Storage with rules limiting access to authenticated users.
- The system shall encrypt sensitive data at rest and in transit and follow best practices for authentication token handling.
- The system shall require user consent for data used in personalization and analytics and follow applicable data privacy best practices.

3.15 Use Case Diagram

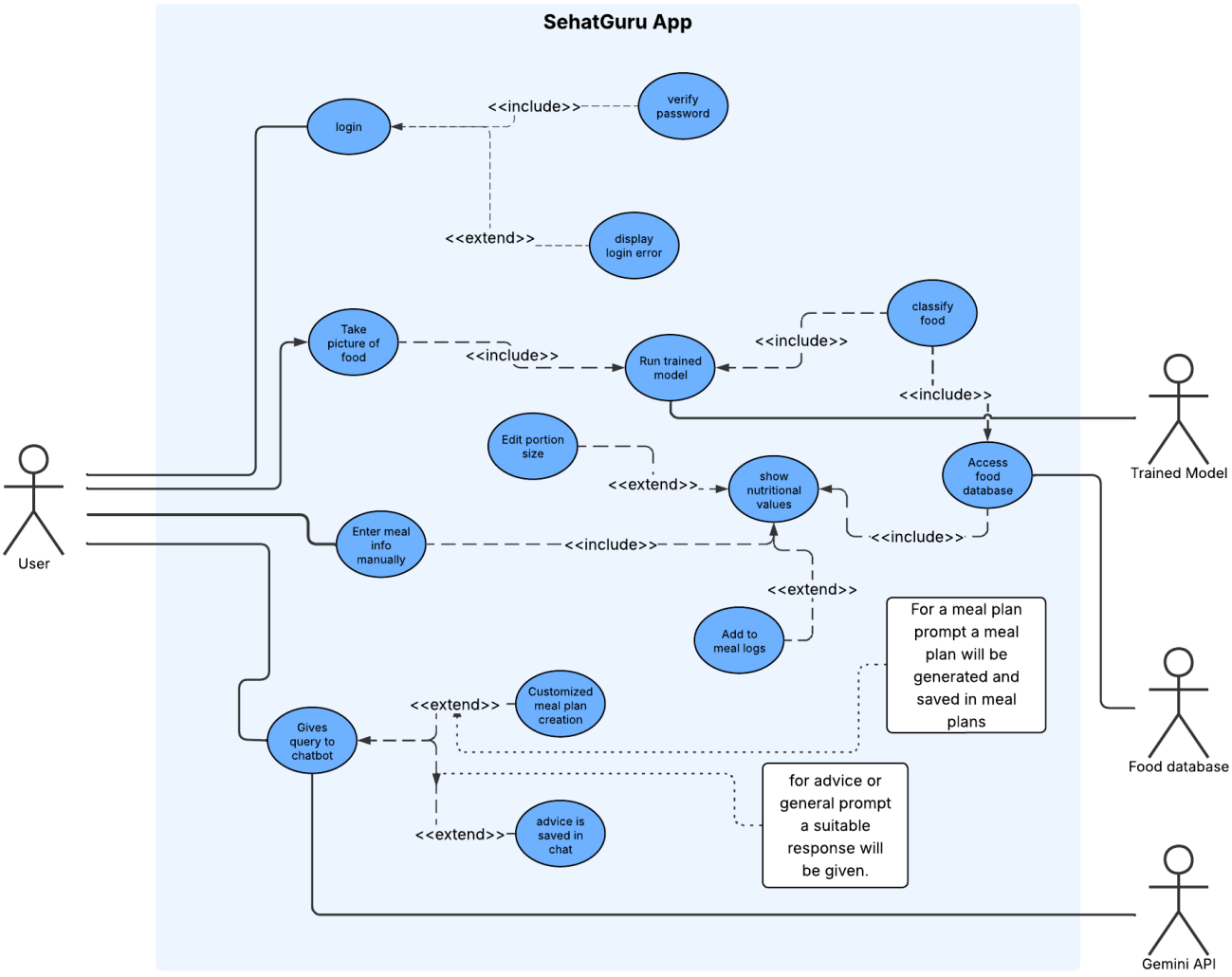

Figure 1: Use Case Diagram for SehatGuru

4 Non-Functional Requirements

4.1 Performance

- The system shall identify food items and display calorie estimates within 5 seconds under normal network conditions (4G/WiFi) for 90% of requests to enable quick meal logging without excessive wait times.
- The AI chatbot shall generate responses for meal preparation and nutritional advice queries within 8 seconds for typical user queries to ensure smooth conversational flow without frustrating delays.
- The nutrition tracking dashboard shall load and display user data (daily/weekly/monthly views) within 3 seconds to allow users to quickly check their daily progress.

4.2 Security

- The system shall protect user accounts with secure password requirements (minimum 6 characters) and email verification to ensure personal health data security and authorized access only.

- Password requirements of minimum 6 characters shall be enforced during account registration via Firebase Authentication.
- Email verification shall be required through a confirmation link sent to the user's registered email address upon account creation.
- Users must verify their email before gaining full access to the application features.
- All data transmitted between the mobile application and backend servers shall be encrypted using HTTPS protocol (TLS 1.2 or higher) to prevent interception of personal information.
- User credentials and sensitive health data (weight, height, goals) shall be stored securely using Firebase's built-in security features, including authentication tokens and database security rules.

4.3 Reliability

- The system shall maintain 90% uptime during the MVP testing period to ensure reliable meal tracking.
- If the image classification model fails or cannot identify a dish, the system shall automatically provide the option to redirect users to manual food search without application crash or data loss to ensure meal logging continuity.
- The system shall display user-friendly, contextual error messages for all failure scenarios including network connectivity issues, authentication errors, image processing failures, and invalid data entries to guide users on appropriate next steps.

4.4 Usability

- New users shall be able to complete account registration, health profile setup, and log their first meal within 10 minutes without external assistance to ensure a smooth onboarding experience.
- The application shall provide a complete English language interface for all user-facing features, including menus, labels, error messages, and chatbot interactions.
- The application shall be compatible with and fully functional on Android devices running Android 8.0 (API level 26, Oreo) or higher versions.
- The application shall be compatible with and fully functional on iOS devices running iOS 16.0 or higher versions.
- The application shall follow standard mobile UI/UX patterns and design guidelines commonly used in popular Pakistani mobile applications to ensure intuitive navigation and minimal learning curve.

4.5 Compliance

- The application shall display a prominent disclaimer during initial setup stating that the nutritional advice and recommendations provided are for general informational and wellness purposes only and are not a substitute for professional medical advice, diagnosis, or treatment.
- The system shall obtain explicit user consent during the registration and onboarding process before collecting any health-related information (weight, height, age, health goals, dietary preferences) to ensure users maintain control of their personal data.

4.6 Data Integrity

- The system shall validate all user health profile inputs to maintain data accuracy, including:
  - Weight: 20 kg – 300 kg
  - Height: 100 cm – 250 cm
  - Age: 13 years – 100 years
  - Portion quantities: Greater than 0

  Invalid entries shall be rejected with clear error messages indicating the acceptable range.
- The food classification model shall achieve a minimum accuracy of 75% when identifying dishes from the "Top 20 Pakistani Dishes" subset in testing scenarios with clear, well-lit images.
- The system shall use a unified calculation algorithm for estimating calories and macronutrients regardless of input method, ensuring consistent results for identical food items and portion sizes.

4.7 Maintainability

- Core system modules (authentication, food logging, image classification integration, nutritional calculation, chatbot integration) shall include inline code comments and README documentation explaining functionality, dependencies, and setup procedures.
- The application shall follow a modular architecture pattern separating major functional components:
  - Authentication Module
  - Food Logging Module
  - AI Chatbot Module
  - Dashboard and Analytics Module
  - Notification Module

  Each module shall have clearly defined interfaces to minimize dependencies and enable independent development and testing.