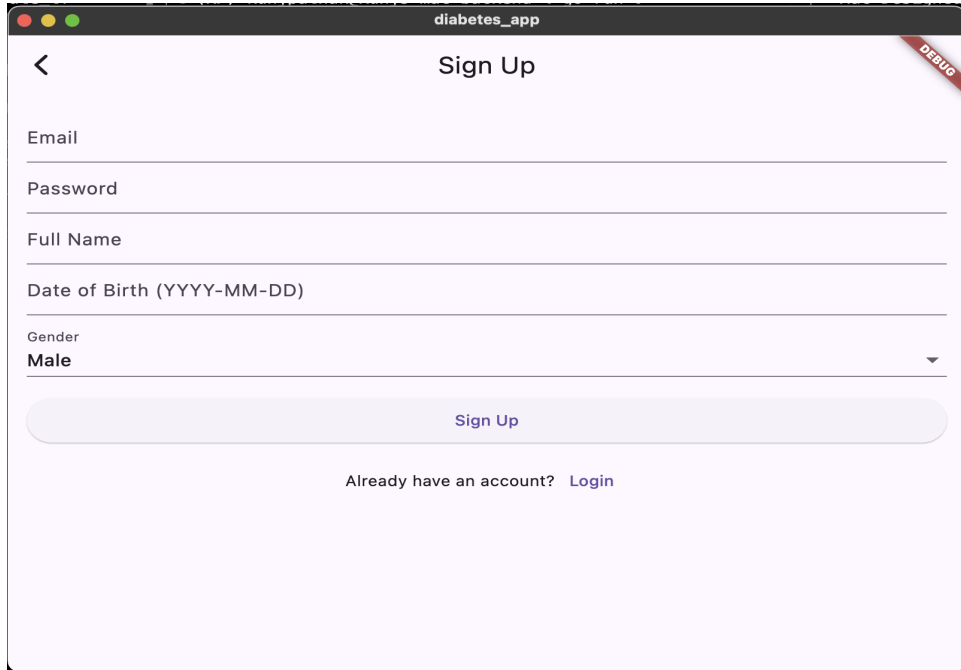


Diabetes Tracker Application

Project Presentation



The screenshot shows a mobile application window titled "diabetes_app" with a "Sign Up" screen. The screen has a light purple background and a dark purple header bar. The header bar contains a back arrow on the left and the title "Sign Up" in the center. A red "DEBUG" banner is visible in the top right corner. The form consists of several input fields: "Email", "Password", "Full Name", "Date of Birth (YYYY-MM-DD)", and "Gender". The "Gender" field is a dropdown menu currently showing "Male". Below the input fields is a large, rounded "Sign Up" button. At the bottom of the screen, there is a link that says "Already have an account? Login".

diabetes_app

< Sign Up

DEBUG

Email

Password

Full Name

Date of Birth (YYYY-MM-DD)

Gender

Male

Sign Up

Already have an account? [Login](#)

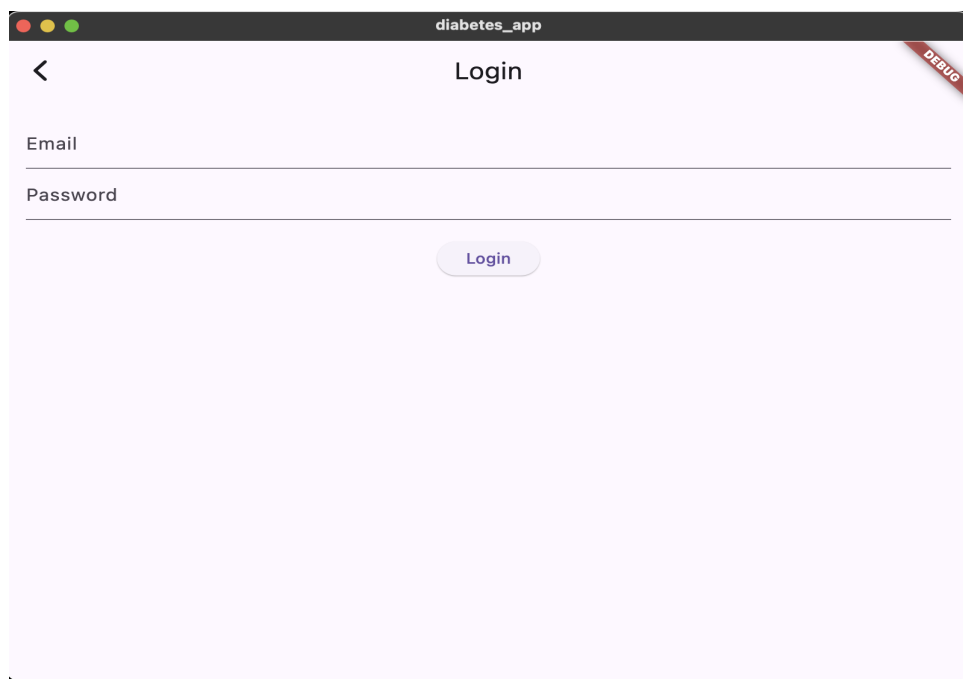
Diabetes Management System

Project Overview

The Diabetes Tracker Application is a comprehensive health management system designed to help individuals with diabetes monitor and manage their condition effectively. The application leverages modern technologies including machine learning for food image classification, personalized recommendations, and comprehensive health tracking.

Key Features

- Glucose Monitoring: Track blood glucose levels over time with visual analytics
- Dietary Management: Log meals and track nutritional intake
- Food Image Classification: Take photos of meals to automatically identify food items and their nutritional content
- Personalized Recommendations: Receive AI-powered dietary and lifestyle recommendations
- Appointment Scheduling: Manage healthcare appointments
- Medication Tracking: Monitor medication schedules and adherence
- Medical Profile Management: Store and manage medical information
- Cross-Platform Support: Available on iOS, Android, web, and desktop platforms



Application Features Visualization

Project Structure

The project is organized into three main components:

1. Backend (Go)

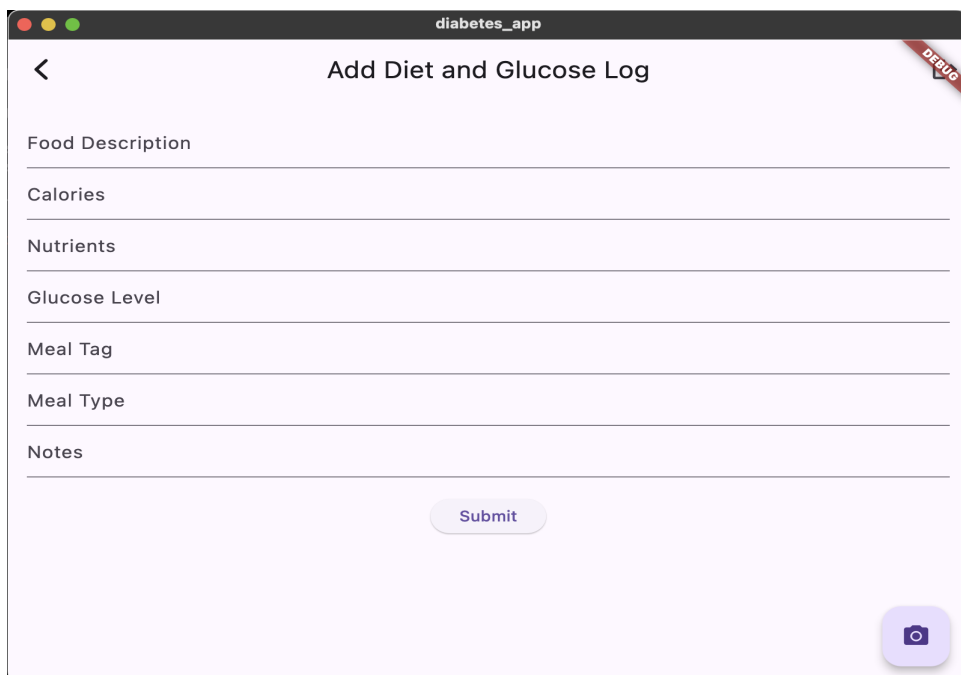
- RESTful API for handling client requests
- PostgreSQL database integration
- JWT authentication
- Integration with AI services

2. AI Services (Python)

- Food image classification using CNN with transfer learning
- Nutritional information extraction
- Glycemic index assessment
- Diabetes impact analysis

3. Frontend (Flutter)

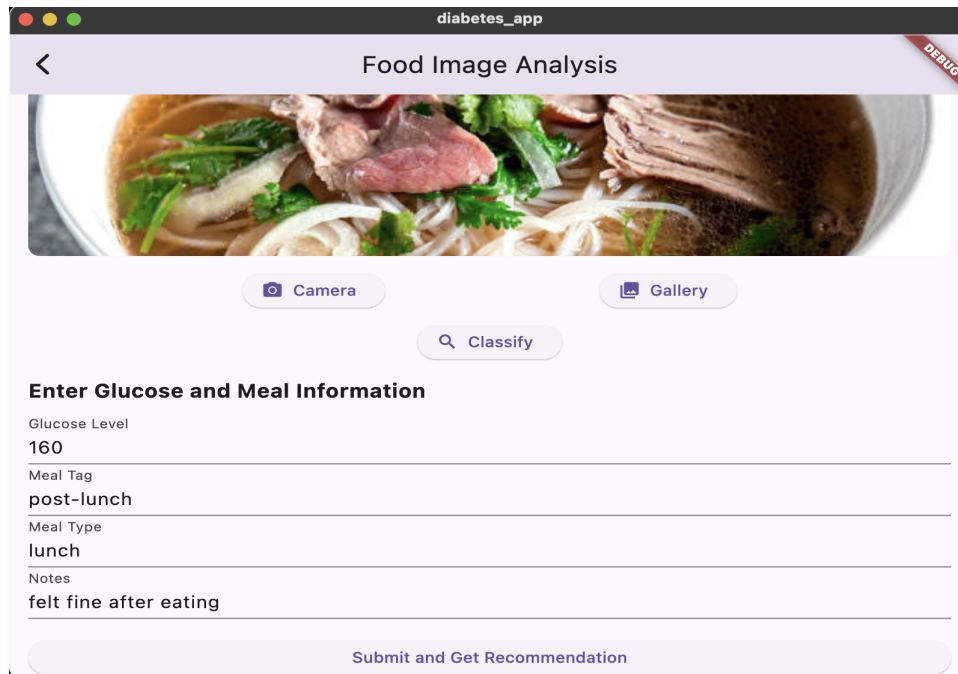
- Cross-platform mobile and web application
- User-friendly interface
- Real-time data visualization
- Camera integration for food image capture

A screenshot of a mobile application window titled "diabetes_app". The screen displays a form titled "Add Diet and Glucose Log". The form has a light purple background and a white border. It contains several input fields with labels: "Food Description", "Calories", "Nutrients", "Glucose Level", "Meal Tag", "Meal Type", and "Notes". Each label is followed by a horizontal line for text entry. At the bottom center of the form is a rounded rectangular button labeled "Submit". In the bottom right corner, there is a purple circular button with a white camera icon. A red "DEBUG" banner is visible in the top right corner of the app window.

Project Component Structure

System Architecture

The Diabetes Tracker Application follows a modern, microservices-oriented architecture with three primary components that work together to provide a comprehensive diabetes management solution.

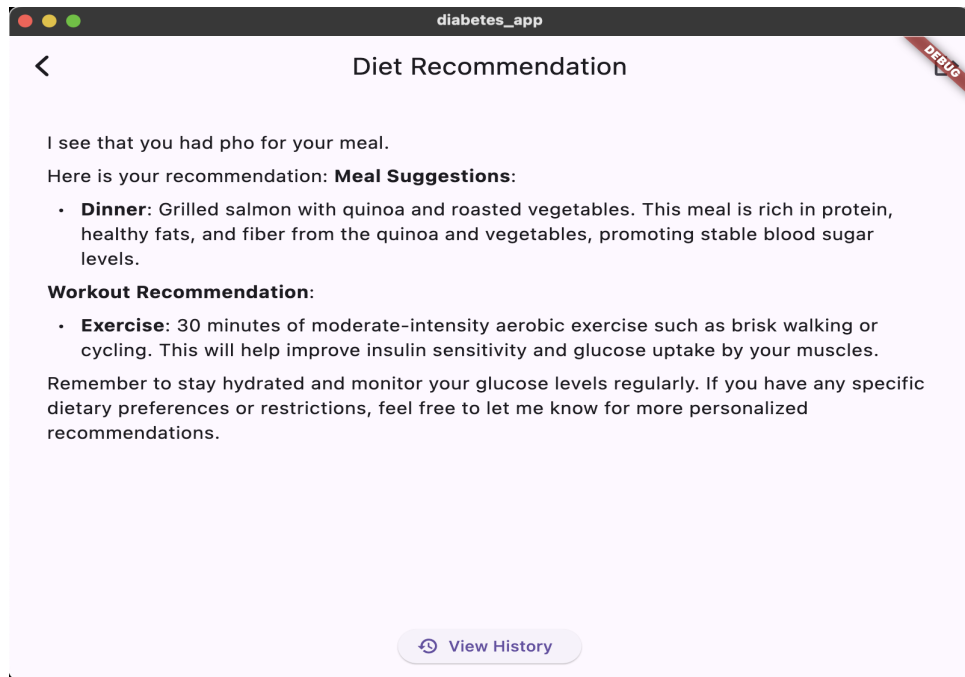


High-Level Architecture Diagram

Component Breakdown

Frontend Architecture (Flutter)

- Screens: User interface screens for different functionalities
- Widgets: Reusable UI components
- Services: Handle API communication with the backend
- Models: Data structures representing application entities
- State Management: Using BLoC pattern for managing application state



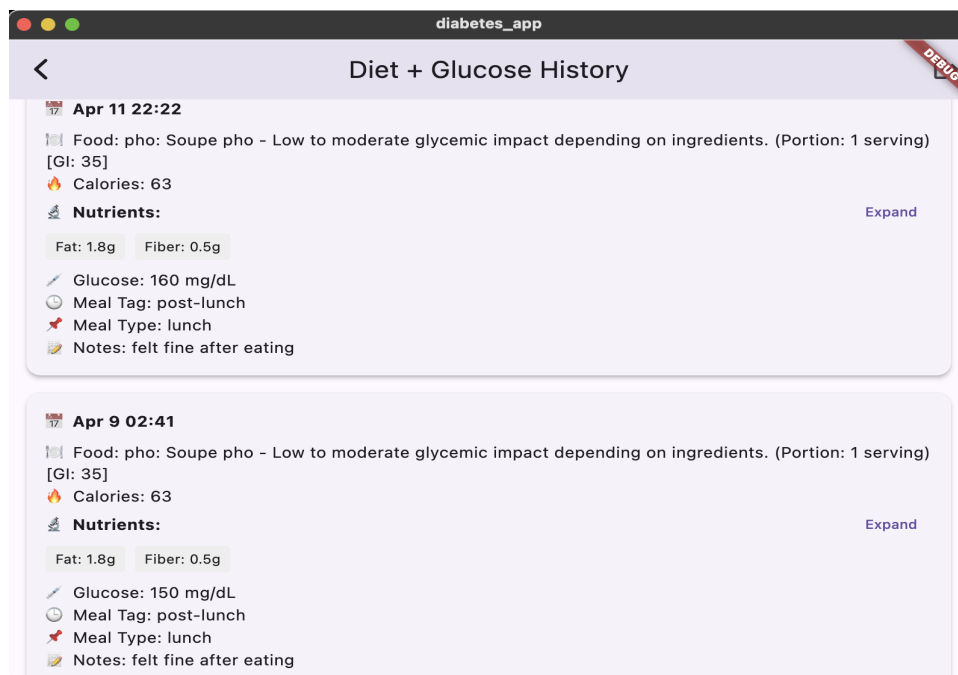
Frontend Component Architecture

Backend Architecture (Go)

- HTTP Router: Handles incoming HTTP requests and routes them to appropriate handlers
- Middleware: Handles cross-cutting concerns like authentication, logging, and error handling
- API Handlers: Process HTTP requests and return responses
- Business Logic Services: Implement application business logic
- Data Models: Define database schema and object relationships
- Database Access Layer: Manages database connections and operations
- AI Service Client: Communicates with the AI service for food classification

AI Service Architecture (Python/Flask)

- Flask API: Handles HTTP requests for food classification and model training
- API Endpoints: Define the service's API contract
- Food Classifier: Implements the classification logic
- CNN Model: Trained machine learning model for food image classification
- Image Processor: Preprocesses images for classification
- Nutrient Database: Stores nutritional information for classified foods



AI Service Component Architecture

Key Data Flows

User Authentication Flow

- User enters credentials in the frontend
- Frontend sends login request to backend
- Backend verifies credentials with database
- Backend generates JWT token
- Frontend stores token and shows dashboard

Food Classification Flow

- User takes food photo in the app
- Frontend processes and sends image to backend
- Backend forwards image to AI service
- AI service classifies image and returns results
- Backend processes results and returns food data
- User confirms and submits diet entry
- Backend stores entry and generates recommendations

Glucose Tracking Flow

- User enters glucose reading in the app
- Frontend sends data to backend
- Backend stores data and analyzes trends
- Frontend displays glucose trends to user

Technology Stack

Component	Technologies
Frontend	Flutter (Dart), BLoC pattern, Material Design
Backend	Go, Native HTTP/Gin, PostgreSQL, JWT
AI Service	Python, Flask, TensorFlow/Keras, OpenCV
Database	PostgreSQL

Future Improvements

- Enhanced AI Models: Implement more sophisticated CNN architectures
- Multiple Food Detection: Add support for identifying multiple food items in a single image
- Portion Size Estimation: Implement more accurate portion size estimation
- Integration with CGM Devices: Connect with continuous glucose monitoring devices
- Expanded Food Database: Add more food items and detailed nutritional information
- Advanced Analytics: Provide deeper insights into glucose patterns and dietary impacts
- Social Features: Add community support and sharing capabilities