Keto Tracker Android App

A comprehensive Android application for tracking ketogenic diet progress, built with modern Android architecture using Kotlin, Jetpack Compose, Room database, and MVVM pattern.

Features

🍎 Food Tracking

- Comprehensive food database with carbs and calories per 100g
- Daily food intake logging with quantity tracking
- Real-time carb limit monitoring (20g ketosis threshold)
- Visual ketosis gauge with color-coded feedback
- Food database management (add, edit, delete items)

🏥 Health Metrics

- Weight tracking with BMI calculation
- Blood glucose and ketone monitoring
- GKI (Glucose Ketone Index) calculation
- Blood pressure and pulse tracking
- Waist circumference measurement
- Daily notes for additional context

📊 Data Visualization

- Interactive charts for all health metrics
- Multiple time aggregations (daily, weekly, monthly)

- Weight and BMI trends
- Glucose/ketone correlation charts
- Blood pressure monitoring
- Carb intake tracking

⚙️ Settings & Data Management

- Customizable user profile (height, carb limits)
- Complete data export/import (CSV format)
- Food database backup and restore
- Theme preferences
- Data reset options

Technical Architecture

Architecture Pattern

- **MVVM (Model-View-ViewModel)** with Clean Architecture principles
- **Repository Pattern** for data abstraction
- **Dependency Injection** with Hilt

Tech Stack

- **Language**: Kotlin
- **UI Framework**: Jetpack Compose with Material Design 3
- **Database**: Room (SQLite)
- **Navigation**: Navigation Compose
- **Async Processing**: Coroutines + Flow
- **Dependency Injection**: Hilt
- **Charts**: MPAndroidChart (planned integration)

- **Data Export**: OpenCSV, Gson

Project Structure

```
app/src/main/java/com/example/ketotracker/
├── data/
│   ├── local/              # Room DAOs and Database
│   ├── model/              # Entity classes
│   └── repository/         # Repository implementations
├── di/                     # Dependency injection modules
├── ui/                     # Compose UI screens
│   ├── dashboard/          # Main dashboard
│   ├── food/               # Food tracking
│   ├── health/             # Health metrics
│   ├── charts/             # Data visualization
│   ├── settings/           # App settings
│   ├── navigation/         # Navigation setup
│   └── theme/              # UI theming
└── KetoTrackerApplication.kt
```

Installation Instructions

Prerequisites

- Android Studio Flamingo (2022.2.1) or later
- Android SDK API 24+ (Android 7.0)
- Kotlin 1.9.0+
- Gradle 8.0+

Setup Steps

1. **Clone/Extract the Project**
   `bash cd android_keto_tracker`

2. **Open in Android Studio**
   - Launch Android Studio
   - Select "Open an Existing Project"
   - Navigate to the `android_keto_tracker` folder
   - Wait for Gradle sync to complete

3. **Build the Project**
   `bash ./gradlew build`

4. **Run the App**
   - Connect an Android device or start an emulator
   - Click "Run" in Android Studio or use:
   `bash ./gradlew installDebug`

Build Variants

- **Debug**: Development build with debugging enabled
- **Release**: Production build (minified, optimized)

Usage Guide

Initial Setup

1. **Launch the app** - you'll see the Dashboard
2. **Set up your profile** in Settings (height for BMI calculation)
3. **Log your first meal** in the Food tab
4. **Record health metrics** in the Health tab

Daily Workflow

1. **Morning**: Log weight and health metrics
2. **Throughout the day**: Add food entries as you eat
3. **Evening**: Review charts and daily summary
4. **Monitor**: Keep carbs under 20g for ketosis

Key Features

- **Ketosis Gauge**: Visual indicator showing carb consumption vs. 20g limit
- **Quick Actions**: Dashboard shortcuts for common tasks
- **Data Export**: Backup your data regularly via Settings
- **Charts**: Analyze trends in the Charts tab

Data Management

Default Food Database

The app includes 13 keto-friendly foods:

- Walnüsse (7g carbs, 654 cal)
- Eier (0.6g carbs, 155 cal)
- Butter (0.1g carbs, 717 cal)
- Rindfleisch (0g carbs, 250 cal)
- Hähnchenbrust (0g carbs, 165 cal)
- Avocado (1.8g carbs, 160 cal)
- Brokkoli (4.4g carbs, 34 cal)
- Käse Cheddar (1.3g carbs, 403 cal)
- Olivenöl (0g carbs, 884 cal)
- Sahne 30% (3g carbs, 290 cal)
- Lachs (0g carbs, 208 cal)
- Magerquark (3.6g carbs, 67 cal)
- 10% Joghurt Natur (4.7g carbs, 110 cal)

Calculations

- **BMI**: Weight (kg) / Height (m)^2
- **GKI**: (Glucose mg/dL ÷ 18) ÷ (Ketones mmol/L) × 100
- **Net Carbs**: Total carbs from food database
- **Ketosis Threshold**: 20g carbs per day

Development

Key Dependencies

```
// Core Android
implementation("androidx.core:core-ktx:1.12.0")
implementation("androidx.activity:activity-compose:1.8.2")

// Compose UI
implementation("androidx.compose.material3:material3")
implementation("androidx.navigation:navigation-compose:2.7.6")

// Database
implementation("androidx.room:room-runtime:2.6.1")
implementation("androidx.room:room-ktx:2.6.1")

// Dependency Injection
implementation("com.google.dagger:hilt-android:2.48")
implementation("androidx.hilt:hilt-navigation-compose:1.1.0")

// Data Processing
implementation("com.opencsv:opencsv:5.8")
implementation("com.google.code.gson:gson:2.10.1")
```

Database Schema

- **FoodItem**: name, carbs_per_100g, calories_per_100g
- **DailyLog**: food_name, quantity, total_carbs, total_calories, date
- **HealthMetric**: date, weight, waist, glucose, ketones, bp_systolic, bp_diastolic, pulse, notes

Future Enhancements

Planned Features

- [ ] Chart implementation with MPAndroidChart
- [ ] Data import/export functionality
- [ ] Barcode scanning for food items
- [ ] Meal planning and recipes
- [ ] Progress photos
- [ ] Notifications and reminders
- [ ] Cloud sync and backup
- [ ] Advanced analytics and insights

Known Limitations

- Charts are placeholder implementations
- Import/export shows placeholder messages
- No barcode scanning yet
- No cloud synchronization

Troubleshooting

Common Issues

1. **Build fails**: Clean and rebuild project `bash ./gradlew clean build`
2. **Database errors**: Clear app data or reinstall
3. **UI issues**: Ensure target SDK 34 and compile SDK 34

Performance Tips

- The app is optimized for offline use
- Database operations are async with coroutines
- UI updates are reactive with StateFlow

Contributing

This project follows Clean Architecture and MVVM principles. When contributing:
1. Follow existing code style and structure
2. Add unit tests for new features
3. Update documentation for API changes
4. Test on multiple Android versions

License

This project is developed as a comprehensive keto tracking solution. All nutritional data should be verified with healthcare professionals.

Version: 1.0.0
Target Android Version: API 24-34 (Android 7.0 - 14)
Build System: Gradle with Kotlin DSL