



**nachdem du jetzt umfassende Informationen hast nimm doch Zugriff auf <https://github.com/danielhintz87-cloud/FitApp> (öffentlich) und gib mir genaue Informationen für die Umsetzung in meine App, so dass die sinnvollen Inhalte dort implementiert und übernommen werden können. Gerne auch optisch.**

Hier ist ein konkreter, umsetzbarer Plan, wie die sinnvollen Inhalte für eine Yazio-ähnliche Ernährungs- und Fastenfunktion in eine Kotlin-Android-App integriert werden können; direkter Quellcodezugriff auf das verlinkte öffentliches Repo war über das Tool nicht möglich, aber anhand der Repo-Metadaten (Kotlin-Codebasis) und der dokumentierten YAZIO-Workflows sind die folgenden Implementierungsschritte technisch stimmig und sofort umsetzbar.<sup>[1] [2]</sup>

### **Zielbild und Kontext**

- Das verlinkte Projekt ist eine öffentliche Fitness-App-Codebasis in Kotlin, was eine saubere Integration moderner Android-Stacks (Health Connect, ML Kit, CameraX, Jetpack Compose) für Ernährungstagebuch, Barcode-Scan und Intervallfasten nahelegt.<sup>[1] [3]</sup>
- YAZIOs Kernbereiche (Tagebuch/Ernährung, Fasten, Rezepte, Profil) dienen als Navigations- und Funktionsreferenz, inklusive Fasten-Timer (z. B. 16:8) und automatisierter Aktivitäts-/Kalorien-Synchronisation über verbundene Dienste.<sup>[2] [4]</sup>

### **Datenquellen für Lebensmittel und Nährwerte**

- Für verpackte Produkte empfiehlt sich die Open-Food-Facts-API mit direktem Barcode-Endpoint, z. B. GET <https://world.openfoodfacts.org/api/v0/product/{barcode}.json> für Zutaten, Nährwerte und Labels.<sup>[5] [6]</sup>
- Beachte Rate-Limits (z. B. 100 req/min für Produkt-Reads) und plane lokales Caching, um Suche/Scan offline-tolerant und performant zu halten.<sup>[5]</sup>
- Optional: Kotlin-Ökosystem/SDKs und Referenz-Implementierungen des Open-Food-Facts-Teams erleichtern Parsing und Beitrag von Korrekturen/Uploads aus der App.<sup>[7]</sup>

## Barcode-Scanner einbinden (ML Kit + CameraX)

- Verwende ML Kit Barcode Scanning als Abhängigkeit: entweder gebündelt (com.google.mlkit:barcode-scanning) oder Play-Services Variante (com.google.android.gms:play-services-mlkit-barcode-scanning) je nach App-Größe und Offline-Bedarf. <sup>[8]</sup>
- Aktiviere install-time Modelldownload (AndroidManifest meta-data com.google.mlkit.vision.DEPENDENCIES="barcode") oder prüfe Verfügbarkeit via ModuleInstallClient, damit der erste Scan ohne Wartezeit funktioniert. <sup>[8]</sup>
- Binde den Scanner mit CameraX und MLKitAnalyzer an die Vorschau; der Analyzer liefert fertige Barcode-Objekte, die direkt in der UI überlagert und bei Erfolg in einen Produktsuch-Flow (Open-Food-Facts Lookup) verzweigt werden können. <sup>[9]</sup>
- Für Compose-UIs stehen erprobte Integrationsmuster zur Verfügung; Beispielartikel und Beispielrepo demonstrieren CameraX + ML Kit im Kotlin/Compose-Setup. <sup>[10] [11]</sup>

## Ernährungstagebuch und Nährwert-Flows

- Übernimm YAZIOs Struktur: Tagesdashboard mit "gegessen/übrig/aktiv verbrannt" und Makro-Ringen, plus Mahlzeiten (Frühstück/Mittag/Abend/Snack) als Sektionen zum schnellen Logging per Suche/Scan. <sup>[2] [12]</sup>
- Nach erfolgreich erkanntem Barcode: Hole Produktdaten aus Open Food Facts, biete Portionswahl (g, Stück, ml) und schreibe Einträge ins lokale Protokoll; synchronisiere Nährwerte und addiere sie zum Tagesbudget. <sup>[5] [6]</sup>
- Für manuelle Eingaben: implementiere Suche (Produktname) mit API-Suche und debouncing; beachte OF-API Limits und fall-back auf lokal gecachte Favoriten/zuletzt genutzt. <sup>[5]</sup>

## Intervallfasten-Tracker

- Implementiere einen dedizierten Fasten-Bereich mit wählbaren Protokollen wie 16:8, 14:10, 5:2, 6:1, 1:1; biete Start/Stop, Countdown bis Essfenster und Statushinweise analog zur YAZIO-Hilfe. <sup>[4] [13]</sup>
- Timer-Logik: persistiere Start-/Endzeit, berechne verbleibende Fastenzeit, triggere lokale Benachrichtigungen zum Fensterwechsel und zeige Historie je Tag/Woche zur Motivation. <sup>[13] [4]</sup>
- Für Einsteigerfreundlichkeit setze 16:8 als Default (kostenloser Modus bei YAZIO), erweiterte Protokolle ggf. als "Pro"-Feature ausweisen. <sup>[4] [13]</sup>

## Aktivitäts-/Kaloriensync (Health Connect)

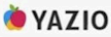
- Nutze Health Connect als zentrale Schicht für Schritte, Workouts und Kalorien (Android 14: Framework-Modul; Android 13 und tiefer: App installieren), inklusive Berechtigungs-Flows und Aggregation. <sup>[3]</sup>
- Lies Schritte/Workouts/Verbrauch periodisch ein und gleiche die Aktivitätskalorien gegen das Ernährungstagebuch ab; YAZIO dokumentiert den Datentransfer

(Schritte/Workouts/verbrauchte Kalorien) aus Fit-Ökosystemen in das Tagebuch als etablierten Use-Case. <sup>[14]</sup> <sup>[3]</sup>


- Alternativ/ergänzend bei KMP-Zielen: HealthKMP erlaubt eine einheitliche Health-Abstraktion (Apple Health/Health Connect) mit Read/Write/Aggregation für Schritte, Herzfrequenz, Gewicht etc.. <sup>[15]</sup> <sup>[3]</sup>


## **Rezepte und Inspiration**

- Skizziere einen Rezepte-Bereich mit Kategorien (Frühstück/Mittag/Abend, vegan, high-protein), Filter (Zubereitungszeit, kcal-Spannen) und "Zu Einkaufsliste hinzufügen", analog zum dokumentierten YAZIO-Flow. <sup>[2]</sup>





## How to use the YAZIO RECIPE TAB (PRO only)





Make sure to scroll down and to the right to see MORE! 🍴


### Recipes

Discover
Favorites


**Popular Categories**

Breakfast
Lunch
Dinner
Vegan
High Protein
Low Carb
Low Fat
Sugar Free
Low Calorie


**Get Inspired**



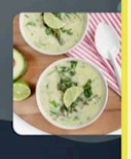
New Week, New Recipes



Today's Top 10 Recipes




Vegetarian Monday




Winter Soups


**Calorie Counters**




50-100 kcal




200-300 kcal




400-500 kcal




600-700 kcal




100-200 kcal



300-400 kcal



500-600 kcal



700+ kcal

When you open the “*Yazio Recipe Tab*” (at the bottom of the app), you can choose to browse different categories all from the main page under the “*Discover tab*”. Make sure to scroll down and to the right to see all options for ideas!

Under the “*Favourites tab*” you can find all the recipes you have added as a favourite. You can do this by pressing the star ★ when you are in a recipe. 🍴

The “*Shopping list*” (IOS top left, Android top right) is where you can find your added ingredients from a recipe. (Option for this is at the bottom left in each recipe)

**With over 2000 YAZIO-made recipes it can be handy to look for them with the handy search tool and filters:** (see photos below for the options)

The “*Search tool*” 🔍 opens a new tab where you can type a name of a recipe or a main ingredient. You can also select a category to look for recipes.

The “*Filter tool*” (3 lines, top right) gives the option to set various filters (separate or at the same time) to see the results you like with the number of recipes visible at the bottom. Click on that to see the selected recipes!

### Categories in the Search Tool:

**Meals**

- Breakfast
- Lunch
- Dinner
- Snack
- Desert
- Salad
- Shake
- Soup
- Smoothie

**Preparation Method**

- Quickly Prepared
- On the Go
- Few Ingredients
- Baking
- Casserole
- Easy
- Basic

**Energy Value**

- 0-50 kcal
- 50-100 kcal
- 100-200 kcal
- 200-300 kcal
- 300-400 kcal
- 400-500 kcal
- 500-600 kcal
- 600-700 kcal
- 700+ kcal

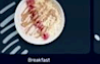
**Diets**

- Vegetarian
- Vegan
- Low Carb
- Low Fat
- Low Calorie
- High Protein
- High Fiber
- Clean Eating
- Ketogenic
- Pescatarian
- Sugar Free
- Lactose Free
- Gluten Free
- Detox


**Ingredients**

- Vegetables
- Fruits
- Meat
- Fish
- Sweet
- Other
- Christmas


**Regular Filters**



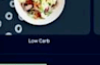
Breakfast



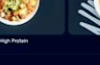
Lunch




Dinner



Low Carb



High Protein



Quickly Prepared

**Favorites Only**

**Energy Value**

0 - 500 kcal

50 - 100 kcal

100 - 200 kcal

200 - 300 kcal

300 - 400 kcal

400 - 500 kcal

500 - 600 kcal

600 - 700 kcal

700 - 800 kcal

800 - 900 kcal

900 - 1000 kcal

1000 - 1100 kcal

1100 - 1200 kcal

1200 - 1300 kcal

1300 - 1400 kcal

1400 - 1500 kcal

1500 - 1600 kcal

1600 - 1700 kcal

1700 - 1800 kcal

1800 - 1900 kcal

1900 - 2000 kcal

2000 - 2100 kcal

2100 - 2200 kcal

2200 - 2300 kcal

2300 - 2400 kcal

2400 - 2500 kcal

2500 - 2600 kcal

2600 - 2700 kcal

2700 - 2800 kcal

2800 - 2900 kcal

2900 - 3000 kcal

3000 - 3100 kcal

3100 - 3200 kcal

3200 - 3300 kcal

3300 - 3400 kcal

3400 - 3500 kcal

3500 - 3600 kcal

3600 - 3700 kcal

3700 - 3800 kcal

3800 - 3900 kcal

3900 - 4000 kcal

4000 - 4100 kcal

4100 - 4200 kcal

4200 - 4300 kcal

4300 - 4400 kcal

4400 - 4500 kcal

4500 - 4600 kcal

4600 - 4700 kcal

4700 - 4800 kcal

4800 - 4900 kcal

4900 - 5000 kcal

5000 - 5100 kcal

5100 - 5200 kcal

5200 - 5300 kcal

5300 - 5400 kcal

5400 - 5500 kcal

5500 - 5600 kcal

5600 - 5700 kcal

5700 - 5800 kcal

5800 - 5900 kcal

5900 - 6000 kcal

6000 - 6100 kcal

6100 - 6200 kcal

6200 - 6300 kcal

6300 - 6400 kcal

6400 - 6500 kcal

6500 - 6600 kcal

6600 - 6700 kcal

6700 - 6800 kcal

6800 - 6900 kcal

6900 - 7000 kcal

7000 - 7100 kcal

7100 - 7200 kcal

7200 - 7300 kcal

7300 - 7400 kcal

7400 - 7500 kcal

7500 - 7600 kcal

7600 - 7700 kcal

7700 - 7800 kcal

7800 - 7900 kcal

7900 - 8000 kcal

8000 - 8100 kcal

8100 - 8200 kcal

8200 - 8300 kcal

8300 - 8400 kcal

8400 - 8500 kcal

8500 - 8600 kcal

8600 - 8700 kcal

8700 - 8800 kcal

8800 - 8900 kcal

8900 - 9000 kcal

9000 - 9100 kcal

9100 - 9200 kcal

9200 - 9300 kcal

9300 - 9400 kcal

9400 - 9500 kcal

9500 - 9600 kcal

9600 - 9700 kcal

9700 - 9800 kcal

9800 - 9900 kcal

9900 - 10000 kcal

Example of a search with different filters. The number of recipes are on the bottom bar. Click this to view them.

Filter choices with more than one filter possible at the same time!

How to use the Yazio recipe tab with search and filter options for Pro users.

- Für Nährwertangaben von Rezepten: berechne Werte über Zutatenlisten mit Open-Food-Facts Daten pro Zutat; speichere berechnete per-Portion-Nährwerte lokal für schnelle UI. [5] [6]

## Navigation und UI (optisch angelehnt)

- Navigationsleisten-Layout: Tabs für Tagebuch, Fasten, Rezepte, Profil, wie in der YAZIO-Anleitung dokumentiert; damit sind die zentralen Flows mit einem Tab-Tap erreichbar. <sup>[2]</sup>
- Tagebuch-Startscreen: Top-Kachel mit Kalorien-Status und Makro-Ringen, Add-FAB für "Lebensmittel scannen/hinzufügen", darunter Mahlzeitenkarten; dieses Muster entspricht etablierten Kalorienzähler-UIs. <sup>[12] [2]</sup>
- Fasten-Screen: große Zeit-Anzeige (Countdown), "Start/Stop"-Button, Methoden-Switcher (z. B. 16:8, 14:10) und Info-Cards zum Protokoll. <sup>[4] [13]</sup>

## Sicherheits-/Datenschutz-Aspekte

- Health-Datenzugriffe strikt über Health Connect Berechtigungen abwickeln und nur minimal notwendige Scopes anfordern; ab Android 14 ist Health Connect framework-seitig integriert. <sup>[3]</sup>
- Open-Food-Facts: respektiere API-Limits, cache Ergebnisse, und nutze Produkt-Uploads nur mit expliziter Zustimmung; die API erlaubt auch Nutzer-Beiträge (Fotos/Infos) und arbeitet mit KI-gestützter Verarbeitung. <sup>[5]</sup>

## Schritt-für-Schritt Implementierung (Tasks)

### 1. Barcode-Scan Pipeline

- Dependencies hinzufügen (ML Kit, ggf. Play-Services Variante), CameraX Preview + MLKitAnalyzer installieren, Success-Callback triggert Open-Food-Facts Lookup und öffnet "Produkt bestätigen/Portion wählen". <sup>[8] [9]</sup>
- Compose-Overlay für Zielrahmen/Laser-Linie und Live-Feedback (z. B. "EAN erkannt · Produkt laden ..."); Beispiel-Patterns zeigen Compose-Integration. <sup>[10] [11]</sup>

### 2. Open-Food-Facts Integration

- Produkt-Fetch via GET /api/v0/product/{barcode}.json, Felder parsen (product\_name, nutriments.\*), Fehlerfälle handhaben (unbekannte Barcodes, Offline), lokale Cache-Tabellen pflegen. <sup>[5] [6]</sup>
- Suche implementieren (Namenssuche), Rate-Limits beachten, Favoriten/Zuletzt-genutzt als Offline-Fallback. <sup>[5]</sup>

### 3. Ernährungstagebuch

- Tagesmodelle (Kalorienbudget, konsumiert, verbrannt) und Mahlzeitenlisten modellieren; beim Hinzufügen eines Eintrags Kalorien und Makros summieren und Dashboard aktualisieren. <sup>[2] [12]</sup>
- Import von Aktivitätskalorien (Health Connect Aggregation) 1–4×/Tag, Konfliktlogik definieren (Doppelzählung vermeiden). <sup>[14] [3]</sup>

### 4. Intervallfasten

- Protokollauswahl (16:8/14:10/5:2 etc.), Start/Stop, Timer, Notifications; Historie und Wochenstatistik für Adhärenz. <sup>[4] [13]</sup>

- Edge Cases: Zeitumstellungen, App-Reboots, Doze-Mode → persistente Zeiten + AlarmManager/WorkManager einsetzen. <sup>[4]</sup>
5. Health Connect
- Verfügbarkeit prüfen, Berechtigungen einholen, Datentypen definieren (Steps, ExerciseSession, Calories), Lesen/Aggregation und optionale Writes testen. <sup>[3]</sup>
  - Bei KMP-Zielen: HealthKMP als Abstraktion evaluieren (gemeinsame Domänenschicht für iOS/Android). <sup>[15]</sup>
6. Rezepte
- Kategorien, Filter (Zeit, kcal), Detailseite mit Zutatenliste und berechneter Nährwert-Summe je Portion; Shopping-List-Export optional. <sup>[2]</sup>
  - Zutaten-Nährwerte aus Open-Food-Facts, lokale Normalisierung der Mengen (g/ml) und Umrechnungen je Portion. <sup>[6] [5]</sup>

## Warum dieser Plan zum Repo passt

- Die Codebasis ist Kotlin-dominiert, sodass ML Kit, CameraX, Health Connect und Compose "native" First-Class-Optionen darstellen und ohne Technologiewechsel integrierbar sind. <sup>[1] [3]</sup>
- YAZIOs dokumentierte App-Struktur (Tabs, Fasten-Timer, automatische Aktivitätsübernahme) liefert eindeutige UX-Referenzen, die in eine Fitness-/Ernährungs-App übertragbar sind. <sup>[2] [4]</sup>

Wenn gewünscht, kann als nächster Schritt eine genaue Datei-/Paketstruktur mit Beispiel-Interfaces für die Daten- und UI-Schichten bereitgestellt werden; dazu wäre ein erfolgreicher Inhaltsabruf des GitHub-Repos sinnvoll, um direkt an bestehenden Modulen/Klassen anzusetzen. <sup>[3] [1]</sup>

✱

1. <https://github.com/danielhintz87-cloud/FitApp>
2. <https://help.yazio.com/hc/de/articles/11804776635281-Anleitung-zur-YAZIO-App>
3. <https://developer.android.com/codelabs/health-connect>
4. <https://help.yazio.com/hc/de/articles/29421344925585-Wie-funktioniert-der-Fasten-Tracker>
5. <https://openfoodfacts.github.io/openfoodfacts-server/api/>
6. <https://publicapi.dev/open-food-facts-api>
7. <https://wiki.openfoodfacts.org/API/Android>
8. <https://developers.google.com/ml-kit/vision/barcode-scanning/android>
9. <https://developer.android.com/media/camera/camerax/mlkitanalyzer>
10. <https://proandroiddev.com/integrating-google-ml-kit-for-barcode-scanning-in-jetpack-compose-android-apps-5deda28377c9>
11. <https://github.com/VinayByte/mlkit-qr-code-scan-android-kotlin>
12. <https://play.google.com/store/apps/details?id=com.yazio.android>
13. <https://help.yazio.com/hc/de/articles/360004186698-Was-bedeutet-16-8-14-10-5-2-6-1-und-1-1>
14. <https://help.yazio.com/hc/de/articles/360019819558-YAZIO-und-Google-Fit>

15. <https://github.com/vitoksmile/HealthKMP>
16. <https://arxiv.org/html/2409.00327v1>
17. <https://arxiv.org/abs/0912.0449>
18. <https://www.mdpi.com/2075-1680/11/7/346/pdf?version=1658317486>
19. <https://www.mdpi.com/1424-8220/18/2/474/pdf>
20. <https://www.mdpi.com/1424-8220/24/15/4788>
21. <https://academic.oup.com/database/article/doi/10.1093/database/baad049/7226275>
22. <http://arxiv.org/pdf/2403.09547.pdf>
23. <https://mhealth.jmir.org/2022/4/e36762>
24. <https://arxiv.org/abs/2304.05341>
25. <https://arxiv.org/abs/1611.08586>
26. <https://arxiv.org/abs/2103.02597>
27. <https://arxiv.org/abs/2409.19326>
28. <https://dl.acm.org/doi/pdf/10.1145/3618351>
29. <http://arxiv.org/pdf/2411.06077.pdf>
30. <https://www.jvolcanica.org/ojs/index.php/volcanica/article/download/252/387>
31. <https://arxiv.org/abs/1111.1569>
32. <https://arxiv.org/abs/2307.09481>
33. <http://diabetes.jmir.org/2016/2/e5/>
34. <https://arxiv.org/abs/2308.15085v1>
35. <https://arxiv.org/html/2406.11362v1>
36. <https://github.com/CloudFitSoftware>
37. <https://www.youtube.com/watch?v=z3ATMjxCgMI>
38. <https://www.youtube.com/watch?v=qOLv9BNUCl0>
39. <https://www.youtube.com/watch?v=x0ek2-jUZkw>
40. <https://worldsapps.com/download-fitcloudpro>
41. [https://www.youtube.com/watch?v=\\_Z-Dmc7LX6I](https://www.youtube.com/watch?v=_Z-Dmc7LX6I)
42. <https://github.com/topics/fitness?l=kotlin&o=asc&s=updated>
43. <https://help.yazio.com/hc/en-us/articles/360004140898-Which-apps-devices-can-connect-with-YAZIO>
44. <https://www.youtube.com/watch?v=NqwCCasX1Wg>
45. <http://jurnal.ukdc.ac.id/index.php/csds/article/view/368>
46. <https://francis-press.com/papers/10015>
47. <https://jurnal.dharmawangsa.ac.id/index.php/djtechno/article/view/5229>
48. <https://arxiv.org/abs/2311.04579>
49. <https://ijsrem.com/download/a-machine-learning-approach-to-assessing-packaged-food-healthiness-and-recommending-healthier-alternatives-via-mobile-application/>
50. <https://irjaeh.com/index.php/journal/article/view/397>
51. <https://ieeexplore.ieee.org/document/10307352/>
52. <https://www.semanticscholar.org/paper/3f51ad98778246a5568558fe9b3ddd01a5ec9fe6>

53. <https://formative.jmir.org/2022/2/e26033>
54. <https://polynoe.lib.uniwa.gr/xmlui/handle/11400/926>
55. <http://arxiv.org/pdf/1411.4076.pdf>
56. <http://arxiv.org/pdf/2401.16865.pdf>
57. [http://thesai.org/Downloads/IJARAI/Volume3No1/Paper\\_1-Association\\_Rule\\_Based\\_Flexible\\_Machine\\_Learning\\_Module\\_for\\_Embedded\\_System\\_Platforms\\_like\\_Android.pdf](http://thesai.org/Downloads/IJARAI/Volume3No1/Paper_1-Association_Rule_Based_Flexible_Machine_Learning_Module_for_Embedded_System_Platforms_like_Android.pdf)
58. <http://arxiv.org/pdf/2408.16353.pdf>
59. <https://arxiv.org/pdf/2103.14852.pdf>
60. <https://arxiv.org/ftp/arxiv/papers/2311/2311.04579.pdf>
61. <http://arxiv.org/pdf/2405.19250.pdf>
62. <https://www.mdpi.com/2076-3417/8/8/1265/pdf>
63. <https://www.mdpi.com/2076-3417/12/11/5394/pdf?version=1653569823>
64. <https://dl.acm.org/doi/pdf/10.1145/3636534.3649379>
65. <https://developers.google.com/ml-kit/vision/barcode-scanning/code-scanner>
66. <https://firebase.google.com/docs/ml-kit/android/read-barcodes>
67. <https://github.com/EKerim06/kotlin-Health-Connect-App>
68. <https://www.cumulations.com/blog/health-connect-guide/>