

MVP REQUIREMENTS DOCUMENT

ESG-Score App

Super SMARTMarket

Version 1.0 | Februar 2026

Autor: Mustafa Demir

Status: Draft | Klassifikation: Intern

*Dieses Dokument dient als technischer Bauplan
und kann direkt an Claude Code zur Entwicklung weitergegeben werden.*

1. Executive Summary

Die ESG-Score App ermöglicht Konsumenten, den Barcode eines Produkts zu scannen und sofort eine verständliche Nachhaltigkeitsbewertung (Environmental, Social, Governance) auf einer Skala von 0-10 mit Ampelsystem zu erhalten. Das MVP fokussiert sich auf die Kernfunktionalität: Scannen, Bewerten, Anzeigen.

1.1 Projektziel

Ein funktionsfähiger Prototyp, der den gesamten Flow abdeckt: Barcode scannen, Produktdaten abrufen, ESG-Score berechnen, Ergebnis anzeigen. Zielplattform: iOS (iPhone).

1.2 Abgrenzung: Was ist NICHT im MVP

Im MVP (Phase 1)	NICHT im MVP (Phase 2+)
Barcode-Scanner Open Food Facts API Anbindung E-Score Berechnung (regelbasiert) Ampel-Anzeige (Grün/Gelb/Rot) Produkt-Detailseite Einfache Suchfunktion Offline-fähiger Grundmodus	KI-Orchestrierung / NLP Bilderkennung (Google Vision) Personalisierung / ML Gamification & Community AR-Overlays User-Accounts & Profile S-Score & G-Score Vollberechnung Alternative Produktvorschläge

2. Technologie-Stack

Komponente	Technologie	Begründung
Frontend	Flutter (Dart)	Cross-platform, aber MVP nur iOS. Claude Code kann Flutter-Code direkt generieren.
Backend	Supabase (EU)	PostgreSQL + Auth + Storage. DSGVO-konform in EU gehostet. Kostenloser Tier reicht für MVP.
Produkt-API	Open Food Facts	Kostenlos, ~600.000 DE-Produkte, Eco-Score, Labels, CO2-Daten.
Barcode Scanner	mobile_scanner (Flutter)	Bewährtes Flutter-Package für Barcode/QR. Nutzt iOS-Kamera nativ.
State Management	Riverpod	Moderner Standard für Flutter. Gut dokumentiert, Claude Code kennt es.
Lokaler Cache	Hive / SharedPreferences	Für Offline-Fähigkeit und schnelle Wiederanzeige gescannter Produkte.
IDE	VS Code + Claude Code	Claude Code schreibt Features direkt. VS Code für Überblick.
Versionierung	GitHub	Code-Sicherung und Nachvollziehbarkeit. Später CI/CD möglich.

3. Screens & User Flow

Der MVP hat 5 Hauptscreens. Der User Flow ist linear: Öffnen, Scannen, Ergebnis sehen, Details lesen.

3.1 Screen-Übersicht

#	Screen	Beschreibung	Kern-Elemente
S1	Home / Scanner	Hauptscreen mit prominentem Scan-Button. Einstiegspunkt der App.	Scan-Button (zentral) Letzte Scans (Liste) Suchfeld (optional)
S2	Kamera / Scan	Kamera öffnet sich, Barcode wird erkannt. Automatischer Übergang zu S3.	Kamera-View Barcode-Overlay Abbrechen-Button
S3	Score-Ergebnis	Zeigt ESG-Gesamt-Score mit Ampel. Kompakte Übersicht aller Dimensionen.	Produktname + Bild Gesamt-Score (0-10) + Ampel E/S/G Einzelscores Zur Detailansicht
S4	Score-Details	Aufschlüsselung einer Dimension. Zeigt woher der Score kommt.	Sub-Scores mit Gewichtung Datenquellen-Anzeige Fehlende Daten markiert Erklärungstexte
S5	Produkt nicht gefunden	Fallback wenn Barcode nicht in Open Food Facts ist.	Fehlermeldung Manuell suchen Produkt melden

3.2 User Flow Diagramm

App öffnen (S1) --> Scan-Button drücken --> Kamera (S2) --> Barcode erkannt
 --> Produkt gefunden? --> JA --> Score-Ergebnis (S3) --> Details (S4)
 --> Produkt gefunden? --> NEIN --> Produkt nicht gefunden (S5)
 --> Zurück zu S1 (Scan-Historie wird aktualisiert)

4. Datenmodell

4.1 Produkt (lokal gecacht)

Feld	Typ	Quelle	Beschreibung
barcode	String	Scanner	EAN-13 Barcode des Produkts
product_name	String	OFF API	Produktname
brand	String?	OFF API	Markenname (optional)
image_url	String?	OFF API	Produktbild-URL

ecoscore_grade	String?	OFF API	A/B/C/D/E oder null
co2_total	double?	OFF API	kg CO2 pro kg Produkt
packaging_tags	List<String>	OFF API	Verpackungsmaterialien
origin_tags	List<String>	OFF API	Herkunftsländer
labels_tags	List<String>	OFF API	Siegel: eu-organic, fairtrade etc.
categories_tags	List<String>	OFF API	Produktkategorien
scanned_at	DateTime	Lokal	Zeitpunkt des Scans

4.2 ESG-Score (berechnet)

Feld	Typ	Beschreibung
e_score	double?	Environmental Score 0-10 (berechnet aus 5 Sub-Scores)
s_score	double?	Social Score 0-10 (MVP: nur basierend auf Labels/Siegeln)
g_score	double?	Governance Score 0-10 (MVP: Platzhalter, da Unternehmensdaten fehlen)
total_score	double?	Gewichteter Durchschnitt: E*0.40 + S*0.35 + G*0.25
data_completeness	double	0.0-1.0: Wie viel % der Datenfelder vorhanden waren
traffic_light	String	green / yellow / red basierend auf total_score

5. API-Integration: Open Food Facts

5.1 Endpunkt

GET <https://world.openfoodfacts.org/api/v2/product/{barcode}.json>

Kein API-Key erforderlich. Rate Limit: Kein offizielles Limit, aber User-Agent Header Pflicht.

Header: User-Agent: ESGScoreApp/1.0 (kontakt@esg-score.app)

5.2 Relevante Response-Felder

API-Feld (JSON Path)	Mapping	Beispielwert
product.product_name	product_name	Hafermilch Bio
product.brands	brand	Oatly
product.image_url	image_url	https://images.off.org/...
product.ecoscore_grade	ecoscore_grade	b
product.ecoscore_data.agribalyse.co2_total	co2_total	0.94
product.packaging_tags	packaging_tags	["en:tetra-pak"]
product.origins_tags	origin_tags	["en:germany"]

product.labels_tags	labels_tags	["en:eu-organic", "en:fairtrade"]
---------------------	-------------	-----------------------------------

Wichtig: Nicht alle Felder sind bei jedem Produkt vorhanden. Die App muss graceful mit null-Werten umgehen.

5.3 Fehlerbehandlung

HTTP Status	Bedeutung	App-Verhalten
200 + status=1	Produkt gefunden	Score berechnen, S3 anzeigen
200 + status=0	Produkt nicht in Datenbank	S5 anzeigen (Produkt nicht gefunden)
Timeout / Netzwerk	Keine Internetverbindung	Lokalen Cache prüfen, sonst Offline-Hinweis
429	Rate Limit (selten)	Retry nach 2 Sekunden, max 3 Versuche

6. Scoring-Logik (MVP-Version)

Das vollständige Scoring-Modell ist im separaten Dokument ESG-SCORING-MODELL-v1.md definiert. Hier die MVP-Vereinfachung:

6.1 MVP-Scope der Dimensionen

Dimension	MVP-Status	Was berechnet wird	Datenquelle
E-Score	VOLL	Alle 5 Sub-Scores: Eco-Score, CO2, Verpackung, Herkunft, Bio-Siegel	Open Food Facts API (alle Felder)
S-Score	TEILWEISE	Nur soziale Siegel (Fairtrade etc.) aus Labels	Open Food Facts labels_tags
G-Score	PLATZHALTER	Kein echtes Scoring. Anzeige: 'Noch keine Governance-Daten'	Keine (Phase 2)

6.2 E-Score Berechnungslogik (Pseudo-Code)

// Siehe ESG-SCORING-MODELL-v1.md für vollständige Tabellen

function calculateEScore(product):

```

subScores = []
if product.ecoscore_grade != null:
    subScores.add({score: mapEcoScore(grade), weight: 0.40})
if product.co2_total != null:
    subScores.add({score: mapCO2(co2), weight: 0.25})

```

```

if product.packaging_tags != null:
    subScores.add({score: mapPackaging(tags), weight: 0.15})
if product.origin_tags != null:
    subScores.add({score: mapOrigin(tags), weight: 0.10})
// Bio-Siegel aus labels_tags extrahieren
bioScore = mapBioLabel(product.labels_tags)
subScores.add({score: bioScore, weight: 0.10})
// Proportionale Neugewichtung bei fehlenden Daten
totalWeight = sum(subScores.map(s => s.weight))
return sum(subScores.map(s => s.score * s.weight / totalWeight))

```

6.3 Ampel-Zuordnung

Score-Range	Ampel	Anzeige
7.0 - 10.0	Grün	Gut bis sehr gut - dieses Produkt schneidet überdurchschnittlich ab
4.0 - 6.9	Gelb	Mittelmäßig - es gibt Verbesserungspotenzial
0.0 - 3.9	Rot	Schlecht - dieses Produkt hat deutliche Nachhaltigkeitsdefizite

7. Nicht-funktionale Anforderungen

Kategorie	Anforderung	Messkriterium
Performance	Score-Berechnung nach Scan < 3 Sekunden	Inkl. API-Call + Berechnung + Rendering
Offline	Bereits gescannte Produkte offline abrufbar	Lokaler Hive-Cache, min. 100 Produkte
DSGVO	Keine personenbezogenen Daten in Phase 1	Kein Login, keine Tracker, kein Analytics
Barrierefreiheit	Grundlegende Accessibility	Kontraste, Schriftgrößen, VoiceOver-Labels
Transparenz	Jeder Score zeigt seine Datenquellen	Quellenangabe auf Score-Detail Screen
Ehrlichkeit	Fehlende Daten nie schätzen oder verbergen	Klare Anzeige: 'Keine Daten verfügbar'

8. Entwicklungsplan (8 Wochen)

Basierend auf 6 Stunden pro Woche. Gesamtbudget: ~48 Stunden.

Woche	Stunden	Aufgabe	Ergebnis
W1	6h	Flutter + Xcode installieren, Hello World App, Projekt-Struktur anlegen	App startet im Simulator
W2	6h	Home Screen (S1) bauen, Navigation einrichten, Design-System (Farben, Fonts)	Navigierbarer Home Screen
W3	6h	Barcode-Scanner (S2) integrieren, Kamera-Permission, Scan-Ergebnis weiterleiten	Barcode wird erkannt
W4	6h	Open Food Facts API anbinden, Produkt-Daten abrufen und parsen	Produktdaten aus API geladen
W5	6h	E-Score Berechnungslogik implementieren (alle 5 Sub-Scores)	E-Score wird korrekt berechnet
W6	6h	Score-Ergebnis Screen (S3) + Score-Details (S4) bauen	Score wird visuell angezeigt
W7	6h	S-Score (vereinfacht), Fehlerbehandlung (S5), lokaler Cache (Hive)	App funktioniert end-to-end
W8	6h	Testing, Bug-Fixing, UI-Polish, Scan-Historie auf Home Screen	MVP fertig zum Testen

9. Risiken & Mitigationen

Risiko	Schwere	Mitigation
Open Food Facts hat kein Eco-Score für viele DE-Produkte	HOCH	API vorab testen mit 20 typischen Barcodes. Fallback: nur verfügbare Daten zeigen, ehrlich kommunizieren.
CO2-Daten sind Kategoriedurchschnitte	MITTEL	Transparent machen: 'Basierend auf Durchschnitt der Kategorie X'. Nicht als exakt darstellen.
Flutter-Lernkurve unterschätzt	MITTEL	Claude Code generiert den Code. Fokus auf Verstehen, nicht auf Schreiben von Code.
Xcode/Simulator Probleme auf M5 Mac	NIEDRIG	Flutter + Xcode sind auf Apple Silicon stabil. Bei Problemen: Hot Reload als Fallback.
Scope Creep (zu viele Features)	HOCH	Strikt an dieses Dokument halten. Alles was nicht in Kapitel 3 steht, ist Phase 2.

10. Definition of Done (MVP)

Das MVP ist fertig, wenn folgende Kriterien erfüllt sind:

#	Kriterium	Status
1	App startet auf iPhone Simulator ohne Crash	Offen

2	Barcode-Scanner erkennt EAN-13 Barcodes zuverlässig	Offen
3	Open Food Facts API wird korrekt abgefragt	Offen
4	E-Score wird nach ESG-SCORING-MODELL-v1.md berechnet	Offen
5	Ampel-System zeigt korrekte Farbe für Score-Range	Offen
6	Alle 5 Screens (S1-S5) sind navigierbar	Offen
7	Fehlende Daten werden mit 'Keine Daten' angezeigt, nie geschätzt	Offen
8	Scan-Historie zeigt letzte 10 gescannte Produkte	Offen
9	App funktioniert offline für bereits gescannte Produkte	Offen
10	Score-Detail zeigt Datenquellen transparent an	Offen

11. Glossar

Begriff	Erklärung (einfach)
MVP	Minimum Viable Product - die einfachste Version der App, die trotzdem funktioniert und nützlich ist
Flutter	Ein Werkzeug von Google, mit dem man Apps für iPhone UND Android gleichzeitig bauen kann
Dart	Die Programmiersprache, die Flutter benutzt (ähnlich wie Python, aber für Apps)
Supabase	Eine Cloud-Datenbank mit Extras (wie Login, Dateispeicher). Wie Firebase, aber in der EU gehostet
Open Food Facts	Eine kostenlose Datenbank mit Infos über Lebensmittel - wie Wikipedia, aber für Produkte
API	Application Programming Interface - eine 'Tür', durch die unsere App Daten von Open Food Facts abruft
EAN-13	Der Standard-Barcode auf Produkten in Europa (13 Ziffern lang)
Eco-Score	Eine Umweltbewertung (A bis E) die auf Lebenszyklusanalysen basiert, berechnet von ADEME (Frankreich)
Riverpod	Ein Flutter-Werkzeug, das hilft Daten in der App zu verwalten (z.B. den gerade berechneten Score)
Hive	Ein lokaler Speicher auf dem Handy - damit die App auch ohne Internet funktioniert
Hot Reload	Eine Flutter-Funktion: Codeänderungen werden sofort in der App sichtbar, ohne Neustart
DSGVO	Datenschutz-Grundverordnung - EU-Gesetz das regelt wie Apps mit Nutzerdaten umgehen müssen
CSRD	Corporate Sustainability Reporting Directive - EU-Pflicht für Unternehmen, über Nachhaltigkeit zu berichten
LkSG	Lieferkettensorgfaltspflichtengesetz - deutsches Gesetz zu Menschenrechten in Lieferketten

Dieses Dokument ist der Bauplan für die ESG-Score App.

Es kann direkt an Claude Code übergeben werden als Entwicklungs-Briefing.

Alles was NICHT in diesem Dokument steht, ist Phase 2.