1. **FlashFeed App - Requirements-Dokument (Prototyp)**

* **1. Projektzusammenfassung**

**Projektname: FlashFeed Prototyp  
Projekttyp: Funktionsfähiger Prototyp einer nachhaltigen Lebensmittel-Echtzeitangebots-Plattform  
Zielplattformen: Flutter Web (GitHub Pages Deployment)  
Entwicklungszeitraum: 5 Monate  
Zweck: Demonstration des Geschäftsmodells für praktische Arbeit  
Geschäftsmodell: Two-sided Market (B2B/B2C) mit Freemium-Monetarisierung**

* **Vision Statement**

**FlashFeed transformiert den deutschen Lebensmitteleinzelhandel durch eine bidirektionale Plattform, die Konsumentenbedürfnisse mit Händlerinteressen verbindet und dabei Food Waste reduziert.**

* **Prototyp-Scope**

**Dieser Prototyp demonstriert alle Kernfunktionalitäten mit simulierten Daten und Mock-Integrationen. Ziel ist die Validierung des Geschäftsmodells, nicht die Markteinführung.**

* **2. Stakeholder-Analyse**
* **2.1 Primäre Zielgruppen**

**B2C-Segment: Preissensitive Konsumenten**

* **Alter: 25-55 Jahre**
* **Einkommen: niedrig bis mittel**
* **Verhalten: Multi-Homing, Prospektanalyse, Mobilitätsbereitschaft**
* **Pain Points: Zeitaufwändige Angebotssuche, suboptimale Kaufentscheidungen**

**B2B-Segment: Lebensmitteleinzelhändler**

* **Fokus: Multi-Filialen-Strukturen (EDEKA, REWE, Schwarz-Gruppe, ALDI)**
* **Bedürfnisse: Lagerdisposition, Abfallreduktion, Kundenakquisition**
* **Technologie: ERP-Integration, API-Kompatibilität**
* **3. Funktionale Anforderungen**
* **3.1 B2C-Funktionalitäten (Konsumenten-App)**
* **3.1.1 Authentifizierung & Onboarding**
* **REQ-3.1.1-1: Benutzerregistrierung mit E-Mail/Telefon**
* **REQ-3.1.1-2: Social Login (Google, Apple, Facebook)**
* **REQ-3.1.1-3: DSGVO-konforme Einverständniserklärung**
* **REQ-3.1.1-4: Tutorial für Hauptfunktionen**
* **REQ-3.1.1-5: Präferenz-Setup (Kategorien, Händler, Radius)**
* **3.1.2 Drei-Panel-Navigation (Haupt-UI-Struktur)**
* **REQ-3.1.2-1: Oberstes statisches Panel mit App-Name und Hamburger-Menü**
* **REQ-3.1.2-2: Drei Haupt-Panels als Tab-Navigation: Angebote, Karte, Echtzeit-Rabatte**
* **REQ-3.1.2-3: Linkes Panel (Angebote) ist standardmäßig aktiv und hat Fokus**
* **REQ-3.1.2-4: Overlay-Einstellungsmenü mit X-Button zum Schließen**
* **3.1.3 Panel 1: Multi-Händler-Angebotsvergleich**
* **REQ-3.1.3-1: Horizontale scrollbare Händler-Icon-Leiste mit Abo-Status**
* **REQ-3.1.3-2: Farbige Icons für aktive Ketten, graue für inaktive**
* **REQ-3.1.3-3: Freemium-Modell: 1 Kette kostenlos, weitere per Abo**
* **REQ-3.1.3-4: Produktgruppen-Grid unterhalb der Icon-Leiste**
* **REQ-3.1.3-5: Mapping verschiedener Händler-Kategorien auf einheitliche Produktgruppen**
* **REQ-3.1.3-6: Aggregierte Angebots-Anzeige pro Produktgruppe von allen aktiven Ketten**
* **REQ-3.1.3-7: Preisvergleich innerhalb Produktgruppen**
* **REQ-3.1.3-8: Ersparnis-Berechnung über mehrere Händler hinweg**
* **3.1.4 Panel 2: Karten-Ansicht**
* **REQ-3.1.4-1: Interaktive Karte mit Filialen der abonnierten Ketten**
* **REQ-3.1.4-2: Radius-Filter (einstellbar 1-20km)**
* **REQ-3.1.4-3: Händler-spezifische Pin-Farben/Icons auf der Karte**
* **REQ-3.1.4-4: Tap-on-Pin zeigt Filial-Details und aktuelle Angebote**
* **REQ-3.1.4-5: Navigation zu ausgewählter Filiale**
* **3.1.5 Panel 3: Echtzeit-Rabatte mit Indoor-Navigation**
* **REQ-3.1.5-1: Feed aller aktuellen Flash-Rabatte mit Countdown-Timer**
* **REQ-3.1.5-2: Produktname, Rabatt-Prozentsatz, ursprünglicher/neuer Preis**
* **REQ-3.1.5-3: Filial-Information mit genauer Adresse**
* **REQ-3.1.5-4: Indoor-Navigation via Bluetooth-Beacon-Integration**
* **REQ-3.1.5-5: Automatischer Lageplan-Download beim Betreten einer Partnfiliale**
* **REQ-3.1.5-6: Präzise Produktposition-Anzeige im Lageplan**
* **REQ-3.1.5-7: Alternative: Lageplan-Anzeige ohne Beacon (Fallback-Modus)**
* **REQ-3.1.5-8: Prototyp-Simulation: "Filiale betreten"-Button aktiviert Lageplan-Modus**
* **3.1.6 Location-based Services**
* **REQ-3.1.6-1: Automatische Standorterkennung (GPS)**
* **REQ-3.1.6-2: Manuelle Standorteingabe**
* **REQ-3.1.6-3: Radius-Definition für Angebote (1-20km)**
* **REQ-3.1.6-4: Turn-by-Turn-Navigation zu Geschäften**
* **REQ-3.1.6-5: Standort-basierte Push-Notifications beim Betreten der Filial-Nähe**
* **3.1.7 Push-Notification-System**
* **REQ-3.1.7-1: Echtzeit-Benachrichtigungen für neue Flash-Rabatte**
* **REQ-3.1.7-2: Standort-basierte Benachrichtigungen (Geofencing)**
* **REQ-3.1.7-3: Einkaufslisten-Erinnerungen**
* **REQ-3.1.7-4: "Angebot läuft ab"-Warnungen 30 Min vor Ende**
* **REQ-3.1.7-5: Benutzereinstellungen für Notification-Frequenz und -Art**
* **3.1.8 Indoor-Navigation-System**
* **REQ-3.1.8-1: Ein Bluetooth-Beacon pro Partnerfiliale am Eingang**
* **REQ-3.1.8-2: Automatische Lageplan-Übertragung via Bluetooth**
* **REQ-3.1.8-3: Zentrales Lageplan-Management durch FlashFeed-Team**
* **REQ-3.1.8-4: Automatische Markierung aktueller Echtzeit-Rabatt-Positionen**
* **REQ-3.1.8-5: Dynamische Lageplan-Updates bei neuen Flash-Angeboten**
* **REQ-3.1.8-6: Offline-Lageplan-Cache in der App (Fallback bei Beacon-Problemen)**
* **3.1.9 Lageplan-Qualitätssicherung**
* **REQ-3.1.9-1: Filialleiter-Interface zur Meldung von Lageplan-Änderungen**
* **REQ-3.1.9-2: Kunden-Feedback-System nach Echtzeit-Rabatt-Nutzung**
* **REQ-3.1.9-3: Automatische Vertriebsbenachrichtigung bei negativem Feedback**
* **REQ-3.1.9-4: Zweigleisiges Update-System: proaktiv (Filialleiter) + reaktiv (Kundenfeedback)**
* **3.1.10 Intelligente Einkaufsliste**
* **REQ-3.1.10-1: Multimodale Eingabe (Text, Sprache, Barcode-Scan, Foto)**
* **REQ-3.1.10-2: Automatische Kategorisierung entsprechend Produktgruppen-Mapping**
* **REQ-3.1.10-3: Smart-Suggestions basierend auf aktuellen Angeboten der abonnierten Ketten**
* **REQ-3.1.10-4: Mengen- und Einheitenverwaltung**
* **REQ-3.1.10-5: Teilen von Einkaufslisten (Haushaltsmitglieder)**
* **REQ-3.1.10-6: Integration verfügbarer Rabatte direkt in Einkaufsliste**
* **REQ-3.1.10-7: Optimierungsvorschläge basierend auf Multi-Händler-Preisvergleich**
* **3.1.11 Personalisierung & Gamification**
* **REQ-3.1.11-1: FlashFeed Coins-System für App-Nutzung und erfolgreich genutzte Rabatte**
* **REQ-3.1.11-2: Tägliche/wöchentliche Challenges (z.B. "5 Flash-Rabatte nutzen")**
* **REQ-3.1.11-3: Level-System basierend auf Gesamtersparnis**
* **REQ-3.1.11-4: Personalisierte Angebote basierend auf Einkaufslisten-Historie**
* **REQ-3.1.11-5: Premium-Features-Freischaltung durch Coins (temporär)**
* **3.1.12 Community Features**
* **REQ-3.1.12-1: Angebotsbewertungen und -kommentare**
* **REQ-3.1.12-2: Produktqualitäts-Feedback nach Indoor-Navigation-Nutzung**
* **REQ-3.1.12-3: Teilen von Flash-Rabatt-Funden in sozialen Medien**
* **3.2 B2B-Funktionalitäten (Händler-Dashboard) - Prototyp mit Mock-Daten**
* **3.2.1 Händler-Onboarding (Simuliert)**
* **REQ-3.2.1-1: Mock-Registrierung für Testhändler (EDEKA, REWE, Netto, etc.)**
* **REQ-3.2.1-2: Vordefinierte Händlerprofile mit realistischen Filialdaten**
* **REQ-3.2.1-3: Demonstration der geplanten API-Integration**
* **REQ-3.2.1-4: Simulierte ERP-Plugin-Oberfläche**
* **3.2.2 Angebotsverwaltung (Demo-Funktionalität)**
* **REQ-3.2.2-1: Regelbasierte Rabatt-Erstellung ("Wenn MHD < 2 Tage, dann 15% Rabatt für 2h")**
* **REQ-3.2.2-2: Simulation von automatischen Angeboten basierend auf Algorithmen**
* **REQ-3.2.2-3: Mock-Lagerbestände mit simuliertem kritischem Bestand**
* **REQ-3.2.2-4: Vordefinierte Produktkategorien (Brot, Fleisch, Obst, Gemüse, etc.)**
* **REQ-3.2.2-5: Geolokalisierte Testfilialen für Location-based Services**
* **REQ-3.2.2-6: Preview-Funktionalität für Angebotspublikation**
* **3.2.3 Händler-ERP-Integration (Push-API-System)**
* **REQ-3.2.3-1: Push-basierte API-Architektur (Händler sendet Daten an FlashFeed)**
* **REQ-3.2.3-2: ERP-Plugin-Entwicklung für SAP, Microsoft Dynamics, Oracle**
* **REQ-3.2.3-3: Regelbasierte Rabatt-Engine ("Wenn MHD < 2 Tage, dann 15% für 2h")**
* **REQ-3.2.3-4: Benutzerfreundliche GUI für Filialleiter zur Regel-Konfiguration**
* **REQ-3.2.3-5: Automatische Datenpaket-Generierung (Produkt, Preis, Standort, Zeitlimit)**
* **REQ-3.2.3-6: Keine Rückübertragung sensibler Geschäftsdaten an FlashFeed**
* **REQ-3.2.3-7: One-way Datentransfer zur Gewährleistung der Datensicherheit**
* **3.2.4 Analytics & Reporting (Mock-Dashboards)**
* **REQ-3.2.4-1: Demo-Dashboard mit realistischen, aber fiktiven KPIs**
* **REQ-3.2.4-2: Simulierte Reichweiten- und Engagement-Statistiken**
* **REQ-3.2.4-3: Beispiel-ROI-Berechnungen für Demonstrationszwecke**
* **REQ-3.2.4-4: CSV/PDF-Export von Mock-Reports**
* **REQ-3.2.4-5: Anonymisierte Wettbewerbsvergleiche (Testdaten)**
* **3.2.5 API-Architektur (Prototyp-Implementation)**
* **REQ-3.2.5-1: Dokumentierte REST-API mit Testendpoints**
* **REQ-3.2.5-2: Simulierte Webhook-Funktionalität**
* **REQ-3.2.5-3: Basis-Authentifizierung für Demo-Zwecke**
* **REQ-3.2.5-4: API-Versionierungskonzept (ohne Backward-Compatibility)**
* **3.3 Plattform-Features**
* **3.3.1 Matching-Algorithmus**
* **REQ-3.3.1-1: Intelligentes Matching zwischen Angeboten und Benutzerpräferenzen**
* **REQ-3.3.1-2: Machine Learning für Relevanz-Optimierung (simuliert)**
* **REQ-3.3.1-3: Standortbasierte Angebotspriorisierung**
* **REQ-3.3.1-4: Zeitbasierte Relevanz (Countdown-Integration)**
* **3.3.2 Content Management**
* **REQ-3.3.2-1: Admin-Dashboard für Plattform-Management**
* **REQ-3.3.2-2: Moderations-Tools für User-Generated Content**
* **REQ-3.3.2-3: Content-Genehmigungsworkflow für Händler**
* **REQ-3.3.2-4: Spam- und Missbrauchserkennung**
* **4. Nicht-funktionale Anforderungen**
* **4.1 Performance**
* **REQ-4.1-1: App-Startzeit unter 3 Sekunden**
* **REQ-4.1-2: Angebots-Updates in Echtzeit (< 5 Sekunden)**
* **REQ-4.1-3: 99.5% Uptime für kritische Services**
* **REQ-4.1-4: Skalierung auf 100.000+ parallele Nutzer**
* **4.2 Benutzerfreundlichkeit**
* **REQ-4.2-1: Intuitive Navigation (max. 3 Klicks zu Hauptfunktionen)**
* **REQ-4.2-2: Responsive Design für alle Bildschirmgrößen**
* **REQ-4.2-3: Barrierefreiheit nach WCAG 2.1 Level AA**
* **REQ-4.2-4: Mehrsprachigkeit (Deutsch, Englisch, Türkisch)**
* **4.3 Sicherheit & Datenschutz**
* **REQ-4.3-1: DSGVO-Compliance mit Privacy-by-Design**
* **REQ-4.3-2: End-to-End-Verschlüsselung für sensible Daten**
* **REQ-4.3-3: Zwei-Faktor-Authentifizierung für B2B-Konten**
* **REQ-4.3-4: Regelmäßige Sicherheitsaudits und Penetrationstests**
* **REQ-4.3-5: Anonymisierung von Nutzerdaten für Analytics**
* **4.4 Kompatibilität**
* **REQ-4.4-1: Flutter Web mit modernen Browsern**
* **REQ-4.4-2: Progressive Web App Standards**
* **REQ-4.4-3: Offline-Funktionalität für Einkaufslisten**
* **REQ-4.4-4: Cross-Platform-Konsistenz**
* **5. Technische Architektur (Prototyp)**
* **5.1 Frontend**
* **Framework: Flutter Web (für GitHub Pages Deployment)**
* **State Management: BLoC Pattern**
* **Local Storage: Browser LocalStorage für Mock-Daten-Persistierung**
* **Maps Integration: Google Maps JavaScript API**
* **5.2 Prototyp-Infrastruktur**
* **Hosting: GitHub Pages (kostenlos, automatische URL-Generierung)**
* **Deployment: GitHub Actions für automatischen Build-Prozess**
* **Zugang: QR-Code für direkte Browser-Öffnung auf Mobilgeräten**
* **Mock-Data: Lokal in der Web-App generiert und gespeichert**
* **5.3 Prototyp-spezifische Komponenten**
* **Mock-ERP-Simulator: Timer-basierte Rabatt-Generierung alle 2 Stunden**
* **Location-Services: GPS-Integration für echte Standorterkennung**
* **Push-Notification-System: Web Push Notifications für Browser**
* **Beacon-Simulator: "Filiale betreten"-Button aktiviert Lageplan-Modus**
* **Lageplan-Generator: Vordefinierte SVG-Lagepläne mit dynamischen Markern**
* **Produktgruppen-Mapper: Statische Zuordnungstabelle für Händler-Kategorien**
* **Analytics-Dashboard: Fake-Daten mit realistischen Wachstumskurven**
* **5.4 GitHub Pages Deployment**
* **Repository: Public GitHub Repository mit Flutter Web Build**
* **URL: https://[username].github.io/flashfeed-prototype**
* **QR-Code: Automatisch generiert für Professor-Zugang**
* **Responsive Design: Funktioniert auf Desktop, Tablet, Smartphone**
* **Offline-Fähigkeit: Service Worker für grundlegende Offline-Funktionalität**
* **6. Monetarisierungsmodell**
* **6.1 B2C Revenue Streams**
* **Freemium-Abonnements:** 
  + **Basic: Kostenlos (1 Kette)**
  + **Premium: 4,99€/Monat (unbegrenzte Ketten)**
  + **Family: 7,99€/Monat (bis zu 4 Accounts)**
* **In-App-Käufe:** 
  + **FlashFeed Coins-Pakete**
  + **Premium-Feature-Freischaltungen**
* **6.2 B2B Revenue Streams**
* **Transaktionsgebühren: 0,15€ pro publiziertem Flash-Rabatt**
* **Subscription Tiers:** 
  + **Starter: 99€/Monat (1 Filiale)**
  + **Professional: 299€/Monat (bis 10 Filialen)**
  + **Enterprise: Individuelle Preise**
* **Advertising Revenue:** 
  + **Banner-Werbung in Consumer App**
  + **Sponsored Product Placements**
* **7. Erfolgsmessung (KPIs)**
* **7.1 B2C Metriken**
* **Monthly Active Users (MAU)**
* **Session Duration**
* **Conversion Rate (Download → Registration)**
* **Premium Subscription Rate**
* **Customer Lifetime Value (CLV)**
* **7.2 B2B Metriken**
* **Händler-Akquisitionsrate**
* **API-Integration-Erfolgsrate**
* **Durchschnittliche Angebote pro Händler**
* **Food Waste Reduction (kg/Monat)**
* **7.3 Plattform Metriken**
* **Gross Merchandise Value (GMV)**
* **Take Rate (% der Transaktionen)**
* **Netzwerkeffekt-Messung**
* **Time-to-Market für neue Features**
* **8. Risiken und Mitigation**
* **8.1 Technische Risiken**
* **Risiko: Skalierungsprobleme bei schnellem Wachstum**
* **Mitigation: Cloud-native Architektur mit Auto-Scaling**
* **8.2 Marktrisiken**
* **Risiko: Wettbewerb durch etablierte Händler-Apps**
* **Mitigation: Fokus auf Unique Value Proposition (Food Waste Reduction)**
* **8.3 Regulatorische Risiken**
* **Risiko: Verschärfung von Datenschutzbestimmungen**
* **Mitigation: Privacy-by-Design und proactive Compliance**
* **9. Entwicklungsplanung**
* **Phase 1 (Monate 1-2): Core Prototyp**
* **B2C-App mit simulierten Echtzeit-Rabatten**
* **Mock-Händlerdaten und Testfilialen**
* **Basis Location Services mit GPS-Integration**
* **Push-Notification-System**
* **Phase 2 (Monate 3-4): Feature-Enhancement**
* **Intelligente Einkaufsliste mit Mock-Preisvergleichen**
* **B2B-Dashboard mit simulierten Analytics**
* **Gamification-System (FlashFeed Coins)**
* **ERP-Plugin-GUI-Simulation**
* **Phase 3 (Monat 5): Prototyp-Finalisierung**
* **Performance-Optimierung für Demo-Zwecke**
* **GitHub Pages Deployment mit QR-Code-Zugang**
* **Dokumentation der simulierten Integrationen**
* **User Testing mit realitätsnahen Szenarien**
* **Post-Prototyp: Markteinführungs-Vorbereitung**
* **Reale ERP-Integrationen entwickeln**
* **Händler-Akquisition beginnen**
* **Skalierungsarchitektur implementieren**
* **10. Prototyp-Limitationen & Ausblick**
* **10.1 Prototyp-Scope**
* **Demonstration aller Kernfunktionalitäten mit simulierten Daten**
* **Keine realen Händler-Integrationen oder Zahlungsabwicklung**
* **Fokus auf User Experience und Geschäftsmodell-Validierung**
* **10.2 Post-Prototyp Entwicklungsbedarfe**
* **Reale ERP-System-Integrationen**
* **DSGVO-konforme Datenverarbeitung**
* **App Store-Compliance und Zertifizierungen**
* **Skalierbare Cloud-Infrastruktur**
* **Händler-Akquisition und Partnerschaften**