

SpendWise

Kişisel Finans Yönetim Uygulaması

"Daha akıllı harca. Daha bilgece biriktir."

Geliştirici: SINA BAGHERZADEH KHIAVI

Tarih: Ocak 2025

Platform: Mobil Uygulama (Flutter) + Backend API (Flask)

Proje Planlama Raporu:

https://drive.google.com/file/d/1ZiwSxTRRA50gxMJeQ_msuYITxH1J_b5t/view

Frontend Repository (GitHub):

<https://github.com/Project-SpendWise/SpendWise-Client>

Backend Repository (GitHub):

<https://github.com/Project-SpendWise/SpendWise-API>

1. Proje Özeti

SpendWise, kullanıcıların kişisel finans yönetimini kolaylaştırmak için geliştirilmiş kapsamlı bir finansal takip uygulamasıdır. Uygulama, kullanıcıların banka hesap özetlerini PDF, XLSX, CSV ve DOCX formatlarında yükleyerek otomatik olarak işlemesini sağlar. Yapay zeka destekli analiz motoru sayesinde işlemler otomatik olarak kategorilere ayrılır, görsel grafiklerle harcama alışkanlıklarını gösterilir ve tasarruf edilebilecek noktalar hakkında içgörüler sunulur.

Proje, modern yazılım geliştirme prensipleri kullanılarak geliştirilmiştir. Backend tarafında Flask framework'ü ile RESTful API, frontend tarafında ise Flutter ile çapraz platform mobil uygulama geliştirilmiştir. Sistem, Clean Architecture prensiplerine uygun olarak modüler bir yapıda tasarlanmıştır.

2

Platform (Backend + Frontend)

5+

Dosya Formatı Desteği

8

Harcama Kategorisi

2

Dil Desteği (TR/EN)

2. Teknoloji Yığını

2.1. Backend Teknolojileri

Web Framework

- **Flask 3.0.0** - RESTful API geliştirme
- **Flask-SQLAlchemy 3.1.1** - ORM ve veritabanı yönetimi
- **Flask-JWT-Extended 4.6.0** - JWT tabanlı kimlik doğrulama
- **Flask-CORS 6.0.1** - Cross-origin istek desteği

Veritabanı

- **SQLite** - Geliştirme ortamı
- **PostgreSQL** - Üretim ortamı (yapılabilir)
- **SQLAlchemy 2.0.44** - Veritabanı araç seti

Güvenlik

- **bcrypt 4.1.2** - Şifre hashleme
- **JWT Tokens** - Erişim ve yenileme tokenları
- **Input Validation** - Girdi doğrulama ve sanitizasyon

Dosya İşleme

- **PDF, XLSX, XLS, CSV, DOCX** - Çoklu format desteği
- **ThreadPoolExecutor** - Asenkron dosya işleme
- **SHA-256** - Dosya hash doğrulama

2.2. Frontend Teknolojileri

Framework & Dil

- **Flutter SDK 3.9.2** - Çapraz platform geliştirme
- **Dart** - Programlama dili
- **Android, iOS, Web** - Platform desteği

State Management

- **flutter_riverpod 2.5.1** - Reaktif state yönetimi
- **Dependency Injection** - Bağımlılık yönetimi

Navigasyon & UI

- **go_router 14.2.0** - Tip-güvenli routing
- **shadcn_ui 0.2.4** - Modern UI bileşenleri
- **Material Design** - Tasarım sistemi

Veri Görselleştirme

- **fl_chart 0.69.0** - Grafik kütüphanesi
- **syncfusion_flutter_charts 31.2.4** - Gelişmiş grafikler

3. Mimari Tasarım

3.1. Backend Mimarisi

Backend, **Application Factory Pattern** kullanılarak modüler bir yapıda geliştirilmiştir. API, Blueprint'ler (modüler route grupları) kullanılarak organize edilmiştir:

- **auth_bp** - Kimlik doğrulama endpoint'leri (/api/auth/*)
- **files_bp** - Dosya yönetimi endpoint'leri (/api/files/*)
- **statements_bp** - Ekstre yükleme ve profil yönetimi (/api/statements/*)
- **transactions_bp** - İşlem CRUD operasyonları (/api/transactions/*)
- **budgets_bp** - Bütçe yönetimi (/api/budgets/*)
- **analytics_bp** - Finansal analitik (/api/analytics/*)

Asenkron İşleme

Banka ekstreleri **ThreadPoolExecutor** kullanılarak arka planda asenkron olarak işlenir. Kullanıcı dosyayı yükledikten sonra hemen yanıt alır, dosya arka planda işlenir ve durum güncellenir. Bu yaklaşım, kullanıcı deneyimini önemli ölçüde iyileştirir.

3.2. Frontend Mimarisi

Frontend, **Clean Architecture** prensiplerine uygun olarak katmanlı bir yapıda organize edilmiştir:

- **Presentation Layer** - UI bileşenleri, ekranlar ve widget'lar
- **Data Layer** - API servisleri, veri modelleri ve dönüşümler
- **Core Layer** - Paylaşılan yardımcı fonksiyonlar, sabitler ve tema yapılandırması
- **State Management Layer** - Riverpod provider'ları ile uygulama durumu yönetimi

3.3. Veri Akışı

Uygulama, RESTful API üzerinden client-server mimarisi kullanır. Tüm veri işlemleri backend API üzerinden gerçekleştirilir. Frontend, JWT token tabanlı kimlik doğrulama kullanarak güvenli bir şekilde API ile iletişim kurar.

4. Veritabanı Şeması

Sistem, ilişkisel veritabanı yapısı kullanarak verileri organize eder. Ana tablolar ve ilişkileri:

Tablo	Açıklama	Ana Alanlar
users	Kullanıcı bilgileri	id, email, username, password_hash, first_name, last_name
statements	Banka ekstreleri (Profil sistemi)	id, user_id, file_name, profile_name, account_type, bank_name, status
transactions	Finansal işlemler	id, user_id, statement_id, date, description, amount, type, category
budgets	Bütçe tanımlamaları	id, user_id, category_id, category_name, amount, period, start_date
files	Yüklenen dosyalar	id, user_id, original_filename, file_path, file_type, processing_status

Profil Sistemi

Her yüklenen ekstre bir "**profil**" olarak organize edilir. Kullanıcılar birden fazla banka hesabı veya kredi kartı için ayrı profiller oluşturabilir. Her profil, özel isim, renk, ikon ve açıklama ile özelleştirilebilir. İşlemler profil bazında filtrelenebilir ve analiz edilebilir.

5. Temel Özellikler

5.1. Dosya Yükleme ve İşleme

- Çoklu Format Desteği:** PDF, XLSX, XLS, CSV, DOCX formatlarında dosya yükleme
- Otomatik İşlem Çıkarma:** Yapay zeka destekli analiz ile işlemlerin otomatik çıkarılması
- Asenkron İşleme:** Dosyalar arka planda işlenir, kullanıcı beklemez
- Durum Takibi:** İşleme durumu (processing, processed, failed) gerçek zamanlı takip

5.2. İşlem Yönetimi

- Otomatik Kategorizasyon:** İşlemler 8 ana kategoriye otomatik ayrılır:
 - Gıda, Ulaşım, Faturalar, Alışveriş, Eğlence, Sağlık, Eğitim, Diğer
- Gelir ve Gider Takibi:** Gelir ve gider işlemleri ayrı ayrı takip edilir
- Filtreleme:** Tarih, kategori, profil ve hesap bazında filtreleme
- Sayfalama:** Büyük veri setleri için sayfalama desteği

5.3. Finansal Analitik

Kategori Analizi

- Kategori bazında harcama dağılımı
- Yüzdelik oranlar
- Toplam tutarlar

Trend Analizi

- Günlük/haftalık/aylık trendler
- Gelir vs gider karşılaştırması
- Aylık karşılaştırmalar

İçgörüler

- Tasarruf oranı hesaplama
- En yüksek harcama kategorisi
- Günlük ortalama harcama
- Öneriler ve uyarılar

Görselleştirme

- Çizgi grafikleri
- Pasta grafikleri
- Çubuk grafikleri
- Alan grafikleri

5.4. Bütçe Yönetimi

- Kategori bazında bütçe tanımlama
- Aylık ve yıllık bütçe dönemleri
- Gerçek harcama vs bütçe karşılaştırması
- Bütçe aşımı uyarıları

5.5. Kullanıcı Deneyimi

- **Çoklu Dil Desteği:** Türkçe ve İngilizce dil desteği
- **Tema Sistemi:** Açık ve koyu tema desteği
- **Profil Yönetimi:** Kullanıcı profil bilgileri güncelleme
- **Güvenli Kimlik Doğrulama:** JWT token tabanlı güvenli giriş sistemi

6. API Endpoint'leri

Backend API, RESTful prensiplere uygun olarak tasarlanmıştır. Tüm endpoint'ler standart JSON formatında yanıt döner. Ana endpoint grupları:

Grup	Endpoint	Metod	Açıklama
Kimlik Doğrulama	/api/auth/register	POST	Kullanıcı kaydı
	/api/auth/login	POST	Kullanıcı girişи
	/api/auth/me	GET	Kullanıcı bilgileri
	/api/auth/refresh	POST	Token yenileme
	/api/auth/logout	POST	Çıkış yapma
	/api/auth/change-password	POST	Şifre değiştirme
Ekstre Yönetimi	/api/statements/upload	POST	Ekstre yükleme
	/api/statements	GET	Ekstre listesi
	/api/statements/{id}	GET/DELETE	Ekstre detay/silme
	/api/statements/profiles	GET	Profil listesi
İşlemler	/api/transactions	GET	İşlem listesi (filtrelenebilir)
	/api/transactions/{id}	GET/DELETE	İşlem detay/silme
Analistik	/api/analytics/categories	GET	Kategori dağılımı
	/api/analytics/trends	GET	Harcama trendleri
	/api/analytics/insights	GET	Finansal içgörüler
	/api/analytics/monthly	GET	Aylık trendler
Bütçe	/api/budgets	GET/POST	Bütçe listesi/oluşturma

Grup	Endpoint	Metod	Açıklama
	/api/budgets/{id}	GET/PUT/DELETE	Bütçe işlemleri

7. Güvenlik Özellikleri

7.1. Kimlik Doğrulama ve Yetkilendirme

- **JWT Token Sistemi:** Erişim tokenları (15 dakika) ve yenileme tokenları (30 gün)
- **Şifre Güvenliği:** bcrypt ile salt'lı şifre hashleme
- **Otomatik Token Yenileme:** Süresi dolan tokenlar otomatik yenilenir
- **Route Koruması:** Korunan endpoint'ler için JWT zorunluluğu

7.2. Veri Güvenliği

- **SQL Injection Koruması:** SQLAlchemy ORM ile parametreli sorgular
- **Input Validation:** Tüm kullanıcı girdileri doğrulanır ve sanitize edilir
- **Dosya Güvenliği:** Dosya tipi ve boyut kontrolü, benzersiz dosya isimleri
- **Kullanıcı İzolasyonu:** Tüm sorgular kullanıcı ID'sine göre filtrelenir

7.3. CORS Yapılandırması

Cross-Origin Resource Sharing (CORS) yapılandırması ile mobil ve web istemcilerinden güvenli erişim sağlanır.

8. Yapay Zeka Entegrasyonu

Sistem, banka ekstrelerinden işlem çıkarma için yapay zeka teknolojilerini kullanmaya hazırlıdır. Şu anda MVP aşamasında gerçekçi mock veri üretimi yapılmaktadır.

Mevcut Durum: Mock Veri Üretici

FakeStatementAnalyzer sınıfı, gerçekçi Türk finansal verileri üretir:

- Gelir işlemleri (maaş, freelance, bonus) - 17,000-25,000 TRY aralığında
- Gider işlemleri Türk hane halkı harcama kalıplarına göre kategorize edilir
- Kategori bazlı bütçe dağılımı (Gıda: %25-30, Ulaşım: %15-20, vb.)
- Seed tabanlı tutarlı üretim (aynı ekstre her zaman aynı verileri üretir)

Gelecek: Gerçek AI Entegrasyonu

Sistem, gerçek AI tabanlı işlem çıkarma için hazır. Desteklenen AI sağlayıcıları:

- Groq** (Önerilen) - 10-100x daha hızlı çıkışım, ücretsiz tier mevcut
- OpenAI GPT-4** - Yüksek doğruluk, ücretli
- Anthropic Claude** - Yüksek doğruluk, ücretli

AI entegrasyonu, mevcut analyzer arayüzü ile uyumludur ve kod değişikliği gerektirmez.

9. Kullanıcı Arayüzü Özellikleri

9.1. Ana Ekran (Dashboard)

- Genel Bakış Kartları:** Gelir, gider ve tasarruf özetini, aylık karşılaştırma göstergeleri
- Harcama Trendleri:** 7 günlük harcama çizgi grafiği
- Aylık Karşılaştırma:** Mevcut ay vs önceki ay karşılaştırması
- Hızlı İstatistikler:** Günlük ortalama harcama, en büyük gider, en çok kullanılan kategori
- Hızlı İçgörüler:** Finansal öneriler ve uyarılar
- Son İşlemler:** En son yapılan işlemlerin listesi

9.2. Analitik Ekranı

- Gelir vs Gider Grafiği:** Zaman içinde gelir ve gider karşılaştırması
- Aylık Trendler:** Son 12 ayın çubuk grafik görünümü
- Harcama Desenleri:** Haftalık harcama desenleri, gün bazlı analiz
- Harcama Tahmini:** Gelecek harcama tahminleri ve trend projeksiyonları
- Yıllık Karşılaştırma:** Mevcut yıl vs önceki yıl karşılaştırması

9.3. Yükleme Ekranı

- Dosya seçici ile PDF, XLSX, CSV dosyaları yüklemeye
- Yükleme ilerleme göstergesi
- Yüklenen dosyaların listesi ve durum takibi
- PDF görüntüleyici ile yüklenen dosyaları görüntüleme

9.4. Profil Ekranı

- Kullanıcı profil bilgilerini düzenleme
- Şifre değiştirme
- Tema değiştirme (açık/koyu)
- Dil seçimi (Türkçe/İngilizce)
- Profil seçimi (çoklu hesap yönetimi)

10. Proje Geliştirme Süreci

10.1. Geliştirme Aşamaları

Proje, planlı bir şekilde 10 haftalık bir süreçte geliştirilmiştir:

- Hafta 1-2:** Proje araştırması, planlama ve UI/UX tasarımı
- Hafta 3:** Flutter proje kurulumu ve temel yapılandırma
- Hafta 4:** PDF yükleme ve metin çıkarma özellikleri
- Hafta 5:** Harcama kategorizasyonu algoritması
- Hafta 6:** Veritabanı entegrasyonu (Firebase/SQLite)
- Hafta 7:** Veri görselleştirme ve grafikler
- Hafta 8:** Finansal içgörüler ve tasarruf önerileri
- Hafta 9:** Test ve optimizasyon
- Hafta 10:** Final raporu ve sunum

10.2. Kod Organizasyonu

Proje, bakımı kolay ve ölçülebilir bir yapıda organize edilmiştir:

- Modüler Mimari:** Her modül bağımsız olarak geliştirilebilir
- Separation of Concerns:** UI, iş mantığı ve veri katmanları ayrılmıştır
- Tekrar Kullanılabilirlik:** Ortak bileşenler ve yardımcı fonksiyonlar paylaşılır
- Test Edilebilirlik:** Birim testleri ve entegrasyon testleri için uygun yapı

10.3. Versiyon Kontrolü

Proje, Git ve GitHub kullanılarak versiyon kontrolü altındadır. Tüm kod değişiklikleri commit'lenir ve branch stratejisi kullanılarak yönetilir.

11. Gelecek Geliştirmeler

11.1. Kısa Vadeli Geliştirmeler

- Gerçek AI Entegrasyonu:** Groq/OpenAI/Claude ile gerçek işlem çıkarma
- Bildirim Sistemi:** Bütçe uyarıları ve faturalar için push bildirimleri
- Offline Desteği:** İnternet bağlantısı olmadan çalışabilme
- Veri Dışa Aktarma:** CSV/Excel formatında rapor dışa aktarma

11.2. Uzun Vadeli Geliştirmeler

- Doğal Dil İşleme (NLP):** Daha akıllı harcama sınıflandırması
- Banka API Entegrasyonu:** Gerçek zamanlı banka bağlantısı
- Çoklu Para Birimi:** Farklı para birimlerini destekleme
- Makine Öğrenmesi:** Harcama tahminleri ve öneriler için ML modelleri
- Topluluk Özellikleri:** Harcama karşılaştırmaları ve istatistikler

12. Sonuç

SpendWise projesi, modern yazılım geliştirme prensipleri kullanılarak başarıyla tamamlanmıştır. Proje, hem backend hem de frontend tarafında ölçeklenebilir, bakımı kolay ve kullanıcı dostu bir yapıya sahiptir.

Uygulama, kullanıcıların kişisel finans yönetimini kolaylaştmak için gerekli tüm temel özellikleri içermektedir: dosya yükleme, otomatik işlem çıkarma, kategorizasyon, görselleştirme, analitik ve bütçe yönetimi. Sistem, gelecekteki geliştirmeler için hazır bir altyapıya sahiptir.

Proje, Flutter ve Flask teknolojileri kullanılarak çapraz platform mobil uygulama geliştirme, RESTful API tasarımı, veritabanı yönetimi, güvenlik uygulamaları ve kullanıcı deneyimi tasarımları konularında kapsamlı bir deneyim sunmaktadır.

Tarih: Ocak 2025

Versiyon: 1.0