



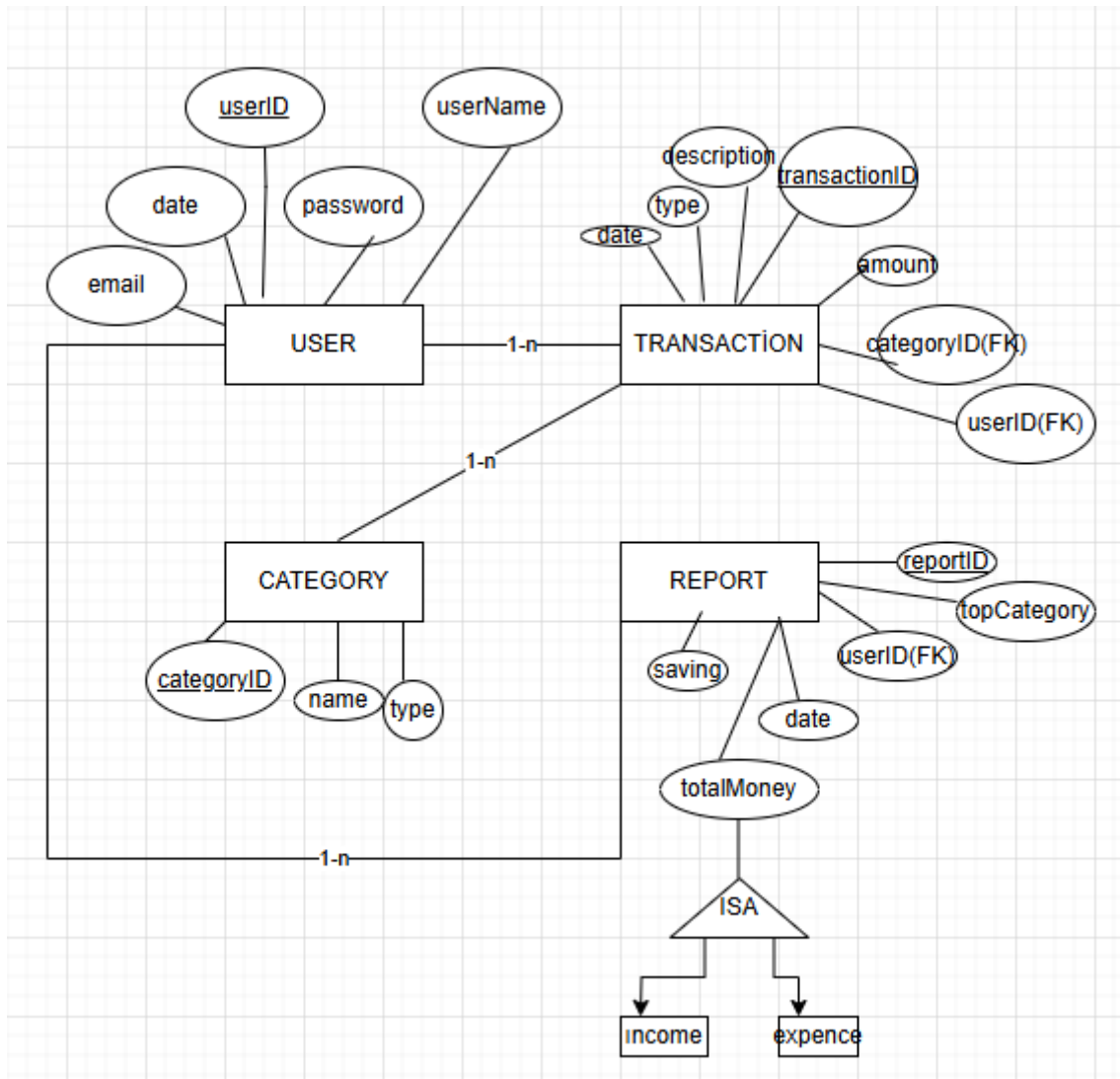
**GRADUATION PROJECT**

**MICRO FINANCE MONITORING SYSTEM**

**INSTRUCTOR: SENIOR HATİCE DAĞLI**

**STUDENT: ELİF SİNEM HAMEŞ**

## PERSONAL BUDGET TRACKER ER DIAGRAM



## USE SCENARIOS

### Scenario 1: User Registration

User sends name, surname, email and password to "/registration" endpoint

A successful response is returned.

### Scenario 2: Add Income

User enters "/transaction/add" endpoint and:

Amount: 5000

Category: Salary

Date: 2024-05-15

Description: May Salary

System records the transaction as belonging to the user

User's total income is updated

### Scenario 3: Receiving Monthly Report

User sends "/report/month?month=5" request

System:

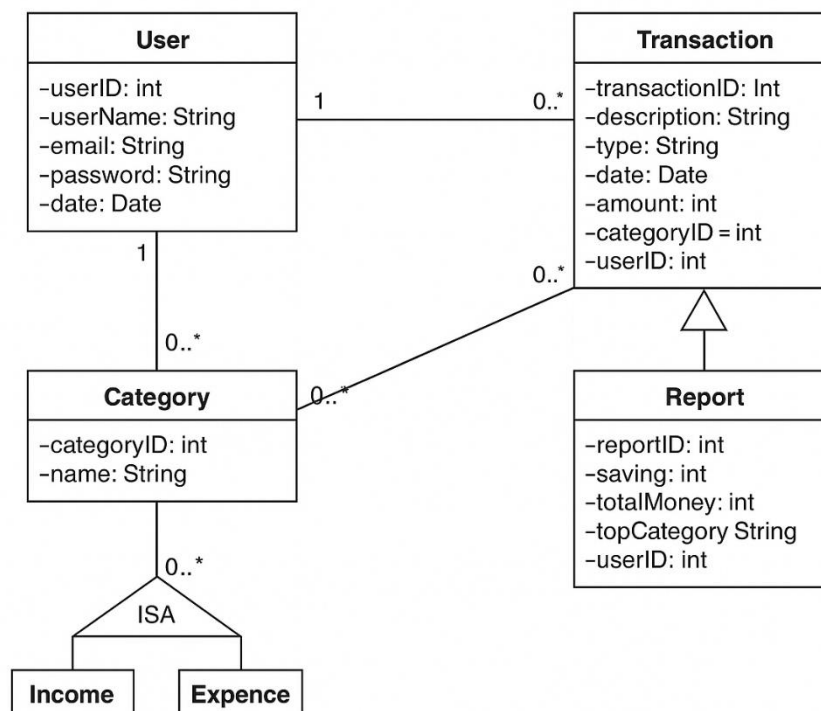
Retrieves user's transactions for May

Calculates income/expense balance

Determines the category with the highest spending.

Report returns in JSON format.

### CLASS DIAGRAM(UML)



## **API DOCUMENTATION (BASIC ENDPOINTS)**

### *UserController API Documentation*

#### **1. Kullanıcı Kaydı - POST /api/users/register**

Kullanıcı kaydı için kullanılır.

İstek Turu: POST

URL: /api/users/register

İstek Govdesi: JSON formatında User nesnesi (ornek: { "name": "...", "email": "...", "password": "..."} )

Dönen Değer: Kaydedilen User nesnesi

#### **2. Tüm Kullanıcıları Listeleme - GET /api/users/all**

Sistemde kayıtlı tüm kullanıcıları getirir.

İstek Turu: GET

URL: /api/users/all

Donen Deger: User nesnelerinden olusan bir liste.

#### **3. Kullanıcı ID ile Getirme - GET /api/users/{id}**

Belirtilen ID degerine gore kullanc bilgilerini getirir.

İstek Turu: GET

URL: /api/users/{id}

Parametre: id (Long turunde, path variable)

Donen Deger: Belirtilen ID'ye ait User nesnesi

### *TransactionController API Documentation*

#### **1. Gelir/Gider Ekleme - POST /transactions/add**

Yeni bir gelir veya gider işlemi ekler.

İstek Turu: POST

URL: /transactions/add

İstek Gövdesi: JSON formatında Transaction nesnesi (örnek: { "userId": 1, "amount": 500.0, "type": "income",  
... })

Donen Değer: Eklenen Transaction nesnesi

## **2. Kullanıcıya Ait İşlemleri Getir - GET /transactions/user/{userId}**

Belirtilen kullanıcı ID'sine ait tüm işlemleri döner.

İstek Turu: GET

URL: /transactions/user/{userId}

Parametre: userId (int, path variable)

Donen Değer: Transaction nesnelerinden oluşan bir liste

## **3. İşlemi ID ile Getir - GET /transactions/{id}**

Belirtilen işlem ID'sine ait işlemi getirir.

İstek Turu: GET

URL: /transactions/{id}

Parametre: id (int, path variable)

Donen Değer: Transaction nesnesi

## **4. İşlemi Güncelle - PUT /transactions/update/{id}**

Belirtilen ID değerine sahip işlemi günceller.

İstek Turu: PUT

URL: /transactions/update/{id}

Parametre: id (int, path variable)

İstek Gövdesi: Güncellenmiş Transaction nesnesi

Donen Değer: Güncellenmiş Transaction nesnesi

## **5. İşlemi Sil - DELETE /transactions/delete/{id}**

Belirtilen ID değerine sahip işlemi siler.

İstek Turu: DELETE

URL: /transactions/delete/{id}

Parametre: id (int, path variable)

Donen Değer: Silinen işlemin ID'si (Long)

## *ReportController API Documentation*

### **1. Aylık Özet Rapor - GET /reports/monthly-summary**

Belirli bir kullanıcı için aylık gelir, gider ve tasarruf özetini getirir.

İstek Turu: GET

URL: /reports/monthly-summary

Parametreler: userId (int), month (int), year (int) - query parametreleri

Dönen Değer: MonthlyReportDTO nesnesi

### **2. En Çok Harcama Yapılan Kategori - GET /reports/top-category**

Belirli bir kullanıcıya ait bir ay içerisindeki en çok harcama yapılan kategoriyi döner.

İstek Turu: GET

URL: /reports/top-category

Parametreler: userId (int), month (int), year (int) - query parametreleri

Donen Deger: Optional kategori bilgisi

### **3. Aylık Özet Raporu JSON Dosyası Olarak Dışarı Aktar - GET /reports/export**

Aylık özet raporu JSON formatında dışarı aktar (dosyaya yazar).

İstek Turu: GET

URL: /reports/export

Parametreler: userId (int), month (int), year (int) - query parametreleri

Donen Deger: İşlem sonucu mesaj (String)

### **4. Tüm Raporları TXT Dosyasına Yedekle - GET /reports/report/backup**

Tüm raporlar bir .txt dosyasına yedekler.

Istek Turu: GET

URL: /reports/report/backup

Parametre: Yok

Donen Değer: Yedekleme sonucu mesaj (String)

### *RegisterController API Documentation*

Yeni bir kullanıcı kaydı oluşturur.

#### ***Request Body (JSON):***

```
{  
  "name": "Kullanıcı Adı",  
  "email": "kullanici@example.com",  
  "password": "şifre123"  
}
```

#### ***Response (JSON):***

```
{  
  "id": 1,  
  "name": "Kullanıcı Adı",  
  "email": "kullanici@example.com",  
  "password": "şifre123"  
}
```

#### Açıklamalar:

- UserRequestDTO: Kullanıcıdan alınacak verileri temsil eder (DTO).
- User: Veri tabanında saklanacak kullanıcı modelidir.
- UserService: İş katmanında kullanıcı ekleme işlemini yapar.
- RegisterController: API isteğini karşılar ve servise yönlendirir
- "@Valid" anotasyonu ile gelen veriler kontrol edilir.