

گردآورنده : علیرضا نوروزی مجرد

موضوع: توضیحات تکمیلی پروژه کیف پول آنلاین برای ساختار و قابلیت ها

تکنولوژی ساخت : SpringBoot-Java

## ساختار

هر شخص می تواند مالک چند حساب باشد

هر حساب می تواند دارای چندین صورت حساب .

هر حساب متعلق به یک شخص می باشد.

هر صورت حساب می تواند متعلق به یک حساب باشد

ساختار پایگاه داده:

موجودیت Person

	Name	Type
1	id	bigint
2	lastname	varchar(255)
3	date_of_birth	timestamp(6) without time zone
4	email	varchar(255)
5	first_name	varchar(255)
6	gender	varchar(255)
7	military_status	varchar(255)
8	mobile_number	varchar(255)
9	national_id	varchar(255)
10	user_status	varchar(255)

موجودیت Account

	Name	Type
1	id	bigint
2	account_number	varchar(255)
3	balance	double precision
4	creation_date	timestamp(6) without time zone
5	sheba_number	varchar(255)
6	person_id	bigint

موجودیت Bill (صورت حساب)

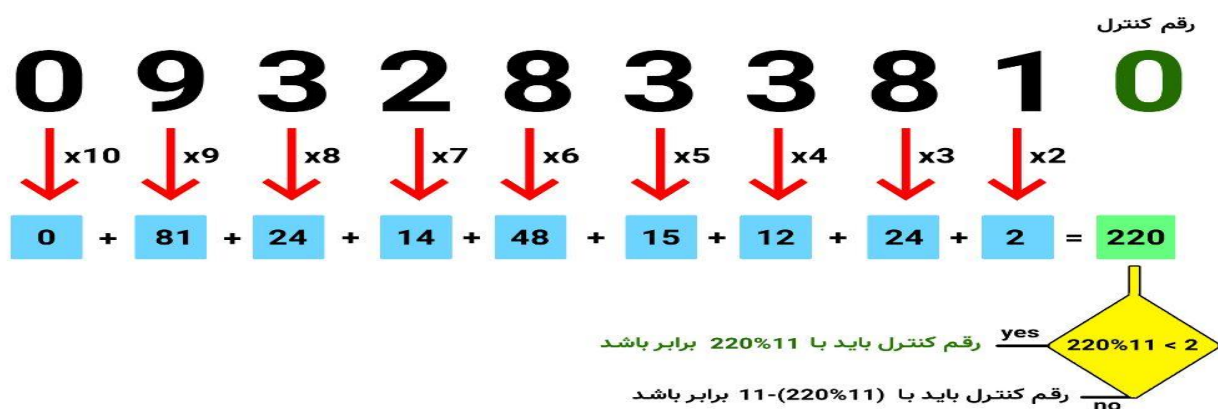
	Name	Type
1	id	bigint
2	bill_amount	double precision
3	bill_status	boolean
4	bill_type	varchar(255)
5	created_at	timestamp(6) without time zone
6	description	varchar(255)
7	account_id	bigint

فرایدهای پیاده شده

- عملیات crud برای موجودیت‌های شخص و حساب.
- عملیات واریز و برداشت از حساب
- مشاهده لیست تراکنشهای حساب اشخاص (هر شخص فقط میتواند تراکنشهای حساب خود را مشاهده کند، وقتی هر کاربر اطلاعات خود یا حساب خود را فراخوانی کند تراکنشهای حسابها به تفکیک قابل مشاهده خواهد بود)
- سیستم لاگین و مدیریت سطح دسترسی در دو سطح مدیر و کاربر

اعتبارسنجی پیاده سازی شده موجودیت اشخاص:

- اعتبارسنجی فیلد ایمیل فیلد کدملی: به صورت یک کلاس و annotation پیاده سازی شده است و از الگوریتم کد ملی استاندارد برای صحتسنجی استفاده می کند



- اعتبار سنجی فیلد ایمیل : با استفاده از کد زیر پیاده سازی شده است

```
@NotBlank(message = "Email is required")
>Email(message = "Email should be valid")
>UniqueEmail
>Column(unique = true)
private String email;
```

- اعتبار سنجی فیلد شماره موبایل: با استفاده از کد زیر پیاده سازی شده است

```
@NotBlank(message = "Mobile number is required")
>Column(unique = true)
>Pattern(regexp = "[0-9]{11}$", message = "Mobile number must
be 11 digits - 09123334444")
>UniqueMobileNumber(message = "Mobile number already exists")
private String mobileNumber;
```

- اعتبار سنجی فیلد وضعیت سربازی : این فیلد به صورت یک کلاس و annotation پیاده سازی شده است و صورتی سن آقا بالای 18 سال باشد باید توضیحاتی در فیلد وضعیت خدمت نوشته شود و در صورتی جنسیت شخص خانم باشد این فیلد هیچ الزامی ندارد.

- اعتبار سنجی یکتا بودن فیلد: تمامی فیلدهای که اعتبار سنجی شده اند یکتا هستند و با استفاده از یک کلاس و annotation پیاده سازی شده است در صورت ورود اطلاعات تکراری پیامی مبنی بر تکراری آن بودن آن فیلد نمایش خواهد داده خواهد شد

- اعتبارسنجی برای تمامی فیلدها : برای تمامی فیلدهای موجودیت شخص با استفاده از انوتیشن @Pattern اعتبار سنجی مشخصی در نظر گرفته شده که در موجودیت Person قابل مشاهده خواهد بود

## اعتبارسنجی حساب ها:

- اعتبارسنجی فیلد شماره حساب و شماره شبا : با استفاده از مکانیزم پیاده شده در موجودیت حساب وقتی درخواست ساخت



حسابی برای شخص خاصی داده می شود سیستم به صورت خودکار شماره حساب و شبا مشابه استاندارد بانکی مثل شکل زیر تولید خواهد کرد که غیر قابل تغییر خواهد بود.  
این مکانیزم توسط کد زیر در موجودیت حساب پیاده شده است.

```
@PrePersist
public void generateAccountAndShebaNumber() {
    String accountNumber =
        UUID.randomUUID().toString().replaceAll("[^0-9]", "");
    while (accountNumber.length() < 16) {
        accountNumber +=
            UUID.randomUUID().toString().replaceAll("[^0-9]", "");
    }
    this.accountNumber = accountNumber.substring(0, 16);

    String bankCode = "IR"; // کد بانک
    String branchCode = "000000"; // کد شعبه
    this.shebaNumber = bankCode + branchCode +
        this.accountNumber;
}
```

- اعتبارسنجی کف موجودی حساب 10000 ریال و سقف برداشت 10 میلیون ریال به صورت روزانه :

برای کنترل کف موجودی حساب از روش زیر به اضافی کنترل مقدار برداشت با حفظ کف موجودی در عملیات برداشت از حساب استفاده شده

```
@NotNull(message = "Balance is required")
@Min(value = 10000, message = "The initial transaction amount
must be 10,000")
private Double balance;
```

برای کنترل مقدار برداشت روزانه تا سقف 10 میلیون ریال از query زیر برای محاسبه مقدار برداشت روزانه یک حساب در صورتی که وضعیت برداشت ها موفق آمیز بوده باشد و کنترل شرط در عملیات برداشت مدیریت شده است .

```
@Query("SELECT COALESCE(SUM(b.billAmount), 0) " +
    "FROM Bill b " +
    "WHERE b.account.id = :accountId " +
    "AND b.billType = 'برداشت' " +
    "AND b.billStatus = true " +
    "AND b.createdAt >= :startOfDay " +
    "AND b.createdAt < :endOfDay") Double
sumWithdrawalsTodayByAccount(@Param("accountId") Long accountId,
    @Param("startOfDay")
    LocalDateTime startOfDay,
    @Param("endOfDay")
    LocalDateTime endOfDay);
```

### اعتبار سنجی حداقل مبلغ برداشت

با استفاده از کد زیر از ذخیره یا برداشت شدن مبلغ کمتر 100.000 ریال جلوگیری می شود و در شروط عملیات برداشت عملیات با شکست مواجه خواهد شد.

```
@NotNull
@DecimalMin(value = "99999", message = "حداقل باید فاکتور مقدار 100000 باشد.")
// @DecimalMax(value = "10000000", message = "باید فاکتور مقدار 10000000 حداکثر باشد.")
private Double billAmount; // برای Double به نوع تغییر
بهتر اعتبارسنجی
```

### ویژگی های اضافه:

برای عملیات برداشت از حساب حتما باید به سامانه ورود کنید و هر شخص فقط می تواند از حساب های خود پول برداشت کند.

توضیحات تراکنش در صورت موفق یا شکست در فیلد توضیحات ذخیره خواهد شد

id	bill_amount	bill_status	bill_type	created_at	description	account_id
1	2	100000	<input checked="" type="checkbox"/>	واریز	7/15/1403 1:2	واریز موفقیت آمیز بود
2	3	100000	<input checked="" type="checkbox"/>	واریز	7/15/1403 1:3	واریز موفقیت آمیز بود
3	4	100000	<input checked="" type="checkbox"/>	واریز	7/15/1403 1:3	واریز موفقیت آمیز بود
4	5	100000	<input checked="" type="checkbox"/>	واریز	7/15/1403 1:4	واریز موفقیت آمیز بود
5	6	100000	<input checked="" type="checkbox"/>	برداشت	7/15/1403 1:4	برداشت موفقیت آمیز بود
6	7	1000000	<input type="checkbox"/>	برداشت	7/15/1403 1:5	خطا: موجودی کافی نیست
7	8	1000000	<input checked="" type="checkbox"/>	واریز	7/15/1403 1:5	واریز موفقیت آمیز بود
8	9	1000000	<input checked="" type="checkbox"/>	برداشت	7/15/1403 1:5	برداشت موفقیت آمیز بود
9	10	1000000	<input type="checkbox"/>	برداشت	7/15/1403 1:5	خطا: موجودی کافی نیست
10	11	10000000	<input type="checkbox"/>	برداشت	7/15/1403 2:1	خطا: موجودی کافی نیست
11	12	1000000	<input checked="" type="checkbox"/>	برداشت	7/15/1403 2:1	برداشت موفقیت آمیز بود
12	13	1000000	<input checked="" type="checkbox"/>	برداشت	7/15/1403 2:1	برداشت موفقیت آمیز بود
13	14	1000000	<input checked="" type="checkbox"/>	برداشت	7/15/1403 2:1	برداشت موفقیت آمیز بود
14	15	10000000	<input type="checkbox"/>	برداشت	7/15/1403 2:1	خطا: exceeded محدودیت برداشت روزانه
15	16	10000000	<input checked="" type="checkbox"/>	واریز	7/21/1403 7:1	واریز موفقیت آمیز بود

عملیات واریز به حساب توسط هر شخصی امکان پذیر هست و پیاده سازی شده است

سیستم لاگین برای تعیین و احراز هویت اشخاص و سطح تعیین سطح دسترسی به شکلی که هر شخص فقط به اطلاعات خود دسترسی داشته باشد و مدیر به تمامی api ها دسترسی تام دارد پیاده سازی شده است.

نمونه درخواست های json در حالت صحیح برای PostMan همراه فایل موجود هست.

### تست نرم افزار :

سعی شده است در قسمت تست SpringBoot برای تمامی api های تعریف شده در حالات مختلف تست صحیح نوشته شود و اجرا گرفته شود که در سورس برنامه قابل مشاهده هست.

