

## Lap6

Check the validation conditions for accessing a bank account that I already have. Also, verify the validation process for adding a new card, balance information, and transferring funds between bank accounts.

**// verify for entry**

```
// verify for entry
// mus int idNational or userName
@NotNull(message = "ID number or username must be provided") no usages
@Pattern(regexp = "^[1-2]\\d{9}$", message = "ID number must start with 1 for citizens or 2 for residents and consist of 10 digits")
private String idNumber;

// Password
@NotNull(message = "Password cannot be null") no usages
@Size(min = 8, message = "Password must be at least 8 characters")
@Pattern(regexp = "^(?=.*[A-Z])(?=.*[a-z])(?=.*\\d.*\\d.*\\d.*\\d.*\\d.*\\d)[A-Za-z\\d]+$",
        message = "Password must contain at least one uppercase letter, one lowercase letter, and at least 6 digits")
private String bankPassword;

// mus int idNational or userName
@NotNull(message = "ID number or username must be provided") no usages
@Size(min = 4, max = 20, message = "Username must be between 4 and 20 characters")
@Pattern(regexp = "^(?=.*[A-Za-z])(?=.*\\d)[A-Za-z\\d]{4,20}$", message = "Username must contain both letters and numbers")
private String userName;
```

## //Add Card information

```
// add Card information
// Card Numbers
@NotNull(message = "Visa card number cannot be null") no usages
@Pattern(regexp = "^4[0-9]{17}$",
        message = "Visa card number must start with 4 and contain exactly 18 digits without letters or negative numbers")
private String visaCardNumber;

//expirationDate
@NotNull(message = "Expiration date cannot be null") no usages
@JsonFormat(pattern = "yyyy-MM-dd ")
@FutureOrPresent(message = "Expiration date must not be in the past")
private LocalDateTime expirationDate;

//cardholderName;
@NotNull(message = "Cardholder name cannot be null") no usages
@Pattern(regexp = "^[A-Z ]+$", message = "Cardholder name must only contain uppercase letters and spaces")
private String cardholderName;

// CVV
@NotNull(message = "CVV cannot be null") no usages
@Pattern(regexp = "[0-9]{3}$", message = "CVV must be exactly 3 digits")
private String cvv;
```

## //Transfer

```
// Transfer
@NotNull(message = "Sender account cannot be null") no usages
private String senderAccountNumber;

@NotNull(message = "Receiver account cannot be null") no usages
private String receiverAccountNumber;

@NotNull(message = "Amount cannot be null") no usages
@Positive(message = "Amount must be greater than zero")
@Max(value = 50000, message = "The transfer amount cannot exceed 50,000")
private Double amount;

// balance
@NotNull(message = "The account balance cannot be null") no usages
@DecimalMin(value = "0.0", inclusive = true, message = "The account balance must be greater than or equal to 0")
private BigDecimal balance;
```