General information

• Database: MSSQLSERVER

• Language and framework: C# 10.0 and .NET 6

• Testing: Swagger, NUnit

• Other libraries: EF Core

Stories

1. As a cash register officer, I would like to create bill for customers in order to get better insights into customer buying habits

2. As a cash register officer, I would like to update bill data in order to be able to correct data

3. As a cash register officer, I would like to delete bill in order to be able to clean up old bills

4. As a cash register officer, I would like to create add/remove products to the bill in order to automatically calculate total costs of bill

5. As a cash register officer, I would like to get all bills in order to get better insights into customer buying habits

6. As a cash register officer, I would like to get bill by ID in order to find appropriate bill more efficiently

7. As a security officer, I would like to prevent entering invalid bill number in order to increase security of the application

8. As a cash register officer, I would like to verify credit card number, so that I can view basic information based on credit card number and prevent entering invalid credit card numbers

9. As a cash register manager, I would like to set upper limit for total amount of bill in order to get more control on daily turnover

10. As a cash register officer, I would like to set only distinct articles per bill in order to increase performances of the application

11. As an exchange officer, I would like to get EUR, USD, RSD according to exchange rate in order to be able to perform money conversion

Validation rules

Credit card verification – Luhn’s algorithm

1. Number can have 13, 15 or 16 digits.

2. Multiply every other digit by 2, starting with the number’s second-to-last digit, and then add those products’ digits together.

3. Add the sum to the sum of the digits that weren’t multiplied by 2.

4. If the total’s last digit is 0 (or, put more formally, if the total modulo 10 is congruent to 0), the number is valid!

Note:

• All MasterCard numbers uses 16 digits and starts with 51, 52, 53, 54 or 55.

• All VISA numbers uses either 13 or 16 digits and starts with 4.

• All American Express numbers uses 15 digits and start with either 34 or 37.

For example:

1. Visa: 4003600000000014

2. Multiply each of the underlined digits by 2

1•2 + 0•2 + 0•2 + 0•2 + 0•2 + 6•2 + 0•2 + 4•2

2 + 0 + 0 + 0 + 0 + 12 + 0 + 8

3. Sum those products’ digits together (not the products themselves)

2 + 0 + 0 + 0 + 0 + 1 + 2 + 0 + 8 = 13

4. Add that sum (13) to the sum of the digits that weren’t multiplied by 2 (starting from the end)

13 + 4 + 0 + 0 + 0 + 0 + 0 + 3 + 0 = 20

5. Credit card is valid

20 % 10 = 0

Verify bill number – ISO 7064 model 97

1. Bill number contains 3 parts: Identification Code (3 digits), bill number (13 digits) and control number (2 digits).

AAA-BBBBBBBBBBBBB-CC

252-0000000123456-85

2. Control number (CC) is used for verification. Multiply sequence of first 2 parts with 100.

2520000000123456 \* 100 = 252000000012345600

3. Divide product with 97 and reduce the rest of division from 98.

98 – (252000000012345600 % 97) = 98 – 13 = 85

4. Result represents control number of the bill (85).

Validation samples

Bill numbers

• 260-0056010016113-79

• 105-0081231231231-73

• 200-0000000075402-20

• More examples: https://www.nbs.rs/export/sites/NBS\_site/documents/platni-sistem/pregled\_racuna\_banka.pdf

Credit card numbers

• American Express (AMEX)

o 371449635398431

o 341126057925597

o 372164784723888

o 342989413867723

• MasterCard

o 5555555555554444

o 2720997972622890

o 5310691906973276

o 5514768347907230

• Visa

o 4111111111111111

o 4929464147377897

o 4532562104787989

o 4929949655254027478

• VISA Electron

o 4026584122256550

o 4026700311175175

o 4508708295928060