

Final project documentation



**SNOW
COLLEGE**

Christopher Hardamek

CS1410

04.23.22

Inhaltsverzeichnis

Introduction	2
Tasks	2
<i>Task 1: Program does something fun or useful</i>	2
<i>Task 2: Program handles invalid input</i>	2
<i>Task 3: Logic and IO cleanly separated</i>	3
<i>Task 4: UI layer calls down into logic layer for processing</i>	4
<i>Task 5: Read and/or write data from one or more: file, database, network</i>	4
<i>Task 6: Use interfaces to define access to external services (e.g., your storage service, database, network, etc.)</i>	5
<i>Task 7: Use bogus/test implementations of those interfaces in your unit tests (e.g., do _not_ actually write to the filesystem or call the network from a unit test).</i>	5
Retrospective	5
<i>Retrospective: What you learned in the process of completing this project</i>	5
<i>Retrospective: Share which principles/patterns/technologies you enjoyed learning about and why you feel they're valuable/beneficial.</i>	5

Introduction

This documentary is about my final project in CS1410 Object-oriented Programming. The target of this final project is to know how to program and how to handle different situations. Here is my [Repository](#).

Tasks

Task 1: Program does something fun or useful

This program can create accounts. The user can create a saving account, make a deposit, withdraw, and get credit from the bank in the accounts.

Task 2: Program handles invalid input

On the Index page, there are some input fields. One of the Input fields is Name, it isn't possible to create an account with just numbers. This counts for Lastname and name. The other inputs are for the day, month, and year. It isn't possible to create a name as a date. And this program will check if the date exists. The Account number must be between 8 and 12 Digits, otherwise, it shows an error.

On screenshot 1, you can see there is an Error message because the date is not valid.

Please enter your Information

Create an account

Name:

Lastname:

birthday day:

birthday month:

birthday year:

Account number:

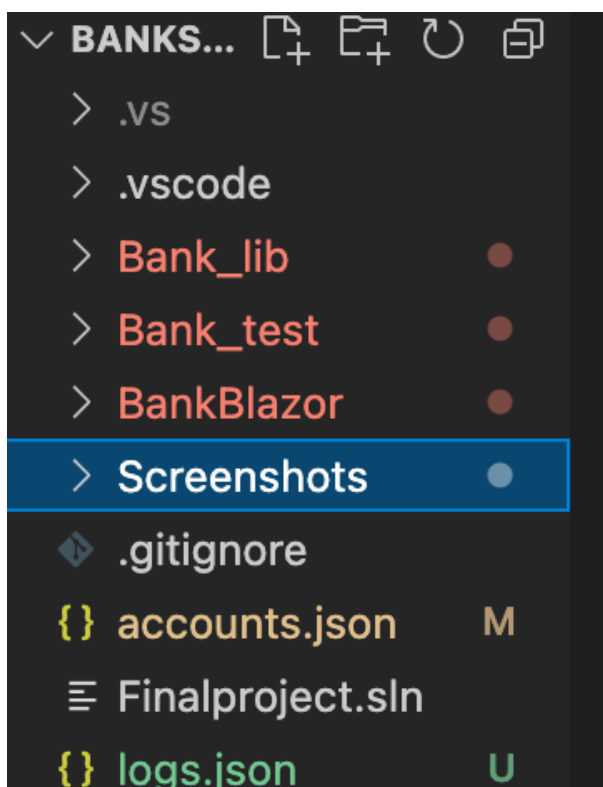
Create Customer

Year, Month, and Day parameters describe an un-representable DateTime.

[Test, Account](#)

Screenshot 1

Mehr darüberschreiben.



Screenshot 2

Task 3: Logic and IO cleanly separated

On Screenshot, there is 3 Folder. One folder Bank_lib. In this folder is the whole program. The folder BankBlazor, in this folder, is the Web application. The last folder is Bank_test. There is all the test inside.

In the Finalproject.sln there is the reference between the folder. It is necessary to have this because without the reference is not possible to use all folder and the logic.

```

78 Bank.SaveAccounts();
79 errorMessage = null;
80 } catch (AccountNumberMustBeBetween8And12Digits)
81 {
82     errorMessage = "Account Number must be between 8 and 12 digits";
83 } catch (AccountNameInvalid)
84 {
85     errorMessage = "Account Name must be minimum 3 characters";
86 } catch (Exception ex)
87 {
88     errorMessage = ex.Message;
89 }
90 }
91 public void LoadedAccount()
92 {
93     Bank.LoadAccounts();
94 }
95 }
96 public void GetAccountNumber()
97 {
98     customer.GetAccountNumber(Accountnumber);
99 }
100 }

```

Screenshot 3

Task 4: UI layer calls down into logic layer for processing

In screenshot 3 there is the index page. On line 96 is called the function

GetAccountNumber(Accountnumber). In

screenshot 4 there is the function of screenshot 3

```

239 Bank.SaveAccounts();
240 }
241 }
242 public void GetID()
243 {
244     ID = nextID++;
245 }
246 public void GetAccountNumber(int Accountnumber)
247 {
248     var AccountNumberLength = Accountnumber.ToString().Length;
249     if (AccountNumberLength >= 8 && AccountNumberLength <= 12)
250     {
251         AccountNumber = Accountnumber;
252     }
253     else
254     {
255         throw new AccountNumberMustBeBetween8And12DigitsException();
256     }
257 }
258 }
259 }
260 }
261 }

```

Screenshot 4

Task 5: Read and/or write data from one or more: file, database, network

```

1 {
2   {
3     "Owner": {
4       "Name": "Test",
5       "Lastname": "Account",
6       "Age": 0,
7       "Birthdate": "0001-01-01T00:00:00",
8       "CheckingBalance": 50,
9       "savingsaccount": false,
10      "Credit": false,
11      "SavingsBalance": 0,
12      "log": null,
13      "creditamount": 0,
14      "ID": 1,
15      "AccountNumber": 123455678
16    }
17  },
18  {
19    "Owner": {
20      "Name": "Try",
21      "Lastname": "Two",
22      "Age": 0,
23      "Birthdate": "0001-01-01T00:00:00",
24      "CheckingBalance": 50,
25      "savingsaccount": false,
26      "Credit": false,
27      "SavingsBalance": 0,
28      "log": null,
29      "creditamount": 0,
30      "ID": 2,
31      "AccountNumber": 12345678
32    }
33  }
34 }

```

Screenshot 5

In my project, I work with 2 files. The first file is responsible for the and the other one is responsible for the logs.

In screenshot 5 there are two accounts. One of them is Test Account and the other one is Try Two. They both have a different customerID. Test Account has ID 1 and "Try two" has ID 2. This number is unique.

Test Account has a balance of \$50 and the account number is 123455678. The Try Two account has a balance of \$50 and the account number is 12345678.

On the other hand, there is more information about the accounts. For example, they were both born on the 1st of January in the year 0001. Both accounts don't have a Credit. The saving balance has a balance of \$0 in both accounts. The Age function is not yet implemented.

Task 6: Use interfaces to define access to external services (e.g., your storage service, database, network, etc.)

Fixen

Task 7: Use bogus/test implementations of those interfaces in your unit tests (e.g., do `_not_` actually write to the filesystem or call the network from a unit test).

Retrospective

Retrospective: What you learned in the process of completing this project
I learned much about programming. I learned how to work with file, how to work with exception, Blazor, debuggen, Query expression and Test-driven development.

Retrospective: Share which principles/patterns/technologies you enjoyed learning about and why you feel they're valuable/beneficial.