

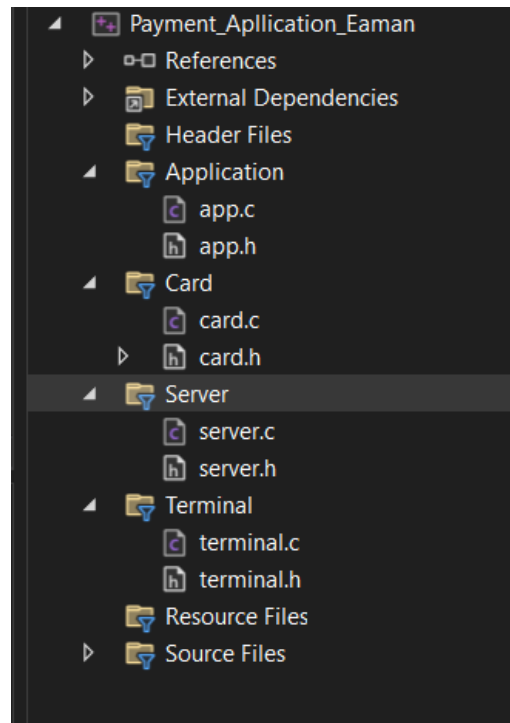


# **Payment Application Project**

**Eaman Mohamed Abdelkader Abdelghany Afan**

Date: 23/11/2022

## Folder structure



## Std.h

```
std.h  card_test_cardholder.cpp
Payment_Application_Eaman (Global Scope)
1  #ifndef STD_H
2  #define STD_H
3
4  #include <stdio.h>
5  #include <stdlib.h>
6
7
8  typedef unsigned char    uint8_t;
9  typedef unsigned short   uint16_t;
10 typedef unsigned int      uint32_t;
11
12 typedef signed char       sint8_t;
13 typedef signed short      sint16_t;
14 typedef signed int        sint32_t;
15
16
17 #endif // std types
```

# Card.h

```
std.h*  card.h  terminal.c*  server.c  card.c*  app.c
Payment_Application_Eaman  (Global Scope)

1
2  #ifndef CARD_H_INCLUDED
3  #define CARD_H_INCLUDED
4
5
6  #include "std.h"
7  #include <string.h>
8
9  typedef struct ST_cardData_t
10 {
11     uint8_t cardHolderName[25];
12     uint8_t primaryAccountNumber[20];
13     uint8_t cardExpirationDate[6];
14 }ST_cardData_t;
15
16 typedef enum EN_cardError_t
17 {
18     CARD_OK, WRONG_NAME, WRONG_EXP_DATE, WRONG_PAN
19 }EN_cardError_t;
20
21 EN_cardError_t getCardHolderName(ST_cardData_t* cardData);
22 EN_cardError_t getCardExpiryDate(ST_cardData_t* cardData);
23 EN_cardError_t getCardPAN(ST_cardData_t* cardData);
24
25 #endif
26
```

# terminal.h

```
terminal.h  std.h  card_test_cardholder.cpp
Payment_Application_Eaman  (Global Scope)

1
2  #ifndef TERMINAL_H_INCLUDED
3  #define TERMINAL_H_INCLUDED
4
5  #include "card.h"
6  #include "std.h"
7  #include <stdio.h>
8  #include <time.h>
9  #define MAX_TRANSACTION_AMOUNT 10000
10
11 typedef struct ST_terminalData_t
12 {
13     float transAmount;
14     float maxTransAmount;
15     uint8_t transactionDate[11];
16 }ST_terminalData_t;
17
18 typedef enum EN_terminalError_t
19 {
20     TERMINAL_OK, WRONG_DATE, EXPIRED_CARD, INVALID_CARD, INVALID_AMOUNT, EXCEED_MAX_AMOUNT, INVALID_MAX_AMOUNT
21 }EN_terminalError_t;
22
23 EN_terminalError_t getTransactionDate(ST_terminalData_t* termData);
24 EN_terminalError_t isCardExpired(ST_cardData_t* cardData, ST_terminalData_t* termData);
25 EN_terminalError_t getTransactionAmount(ST_terminalData_t* termData);
26 EN_terminalError_t isBelowMaxAmount(ST_terminalData_t* termData);
27 EN_terminalError_t setMaxAmount(ST_terminalData_t* termData, float maxAmount);
28 // EN_terminalError_t isValidCardPAN(ST_cardData_t* cardData); Optional
29
30 #endif

90 %  No issues found  Ln: 1  Ch: 1  SPC  CRLF
```

## app.h

```
app.h  x terminal.h  std.h  card_test_cardholder.cpp
Payment_Application_Eaman  (Global Scope)

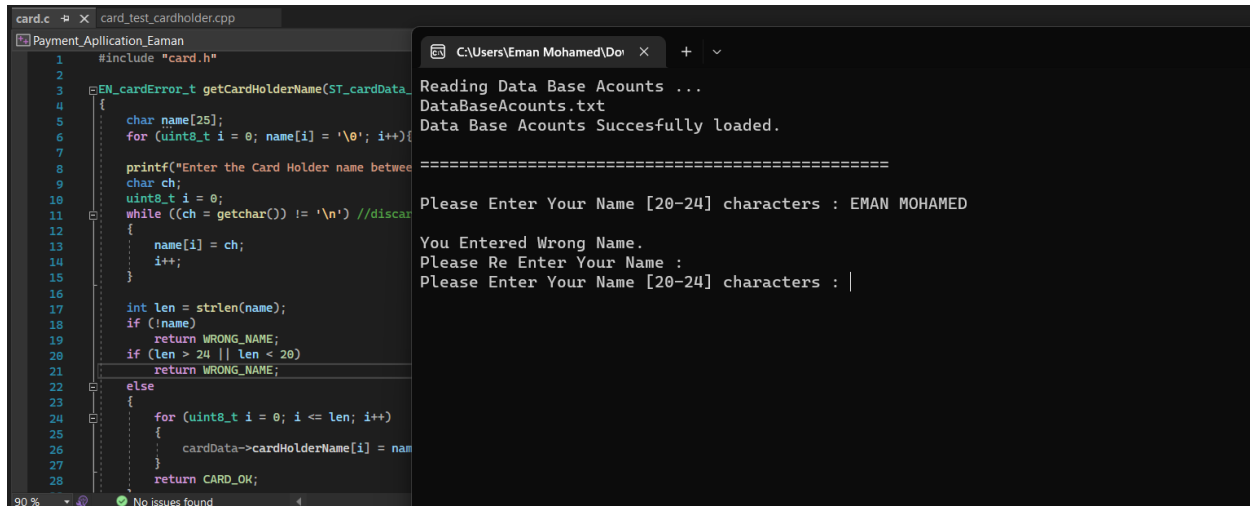
1  #ifndef APP_H_INCLUDED
2      #define APP_H_INCLUDED
3
4
5  #include "std.h"
6  #include "card.h"
7  #include "server.h"
8  #include "terminal.h"
9
10
11  void appStart(void);
12
13  #endif
14
```

## Server.h

```
server.h  x app.h  terminal.h  std.h  card_test_cardholder.cpp
Payment_Application_Eaman  (Global Scope)

1  #ifndef SERVER_H_INCLUDED
2      #define SERVER_H_INCLUDED
3
4
5  #include "terminal.h"
6  #include "card.h"
7  #include "std.h"
8
9
10  typedef enum EN_transState_t
11  {
12      APPROVED, DECLINED_INSUFFICIENT_FUND, DECLINED_STOLEN_CARD, FRAUD_CARD, INTERNAL_SERVER_ERROR
13  }EN_transState_t;
14
15
16  typedef enum EN_serverError_t
17  {
18      SERVER_OK, SAVING_FAILED, TRANSACTION_NOT_FOUND, ACCOUNT_NOT_FOUND, LOW_BALANCE, BLOCKED_ACCOUNT, RUNNING_ACCOUNT
19  }EN_serverError_t;
20
21
22  typedef struct ST_transaction_t
23  {
24      ST_cardData_t cardHolderData;
25      ST_terminalData_t terminalData;
26      EN_transState_t transState;
27      uint32_t transactionSequenceNumber;
28  }ST_transaction_t;
29
30  typedef enum EN_accountState_t
31  {
32      RUNNING,
33      BLOCKED
34  }EN_accountState_t;
35
36
37  typedef struct ST_accountsDB_t
38  {
39      float balance;
40      EN_accountState_t state;
41      uint8_t primaryAccountNumber[20];
42  }ST_accountsDB_t;
43
44
45  EN_transState_t recieveTransactionData(ST_transaction_t* transData);
46  EN_serverError_t isValidAccount(ST_cardData_t* cardData, ST_accountsDB_t* accountReference);
47  EN_serverError_t isBlockedAccount(ST_accountsDB_t* accountReference);
48  EN_serverError_t isAmountAvailable(ST_terminalData_t* termData, ST_accountsDB_t* accountReference);
49  EN_serverError_t saveTransaction(ST_transaction_t* transData);
50
51  void listSavedTransactions(void);
52
53  #endif
```

# Test\_cardholder\_name



```
card.c  card_test_cardholder.cpp
Payment_Application_Eaman
1 #include "card.h"
2
3 EN_cardError_t getCardHolderName(ST_cardData_t cardData)
4 {
5     char name[25];
6     for (uint8_t i = 0; name[i] = '\0'; i++) {}
7
8     printf("Enter the Card Holder name between 20-24 characters : ");
9     char ch;
10    uint8_t i = 0;
11    while ((ch = getchar()) != '\n') //discard unwanted characters
12    {
13        name[i] = ch;
14        i++;
15    }
16
17    int len = strlen(name);
18    if (!name)
19        return WRONG_NAME;
20    if (len > 24 || len < 20)
21        return WRONG_NAME;
22    else
23    {
24        for (uint8_t i = 0; i <= len; i++)
25        {
26            cardData->cardHolderName[i] = name[i];
27        }
28        return CARD_OK;
29    }
30 }
```

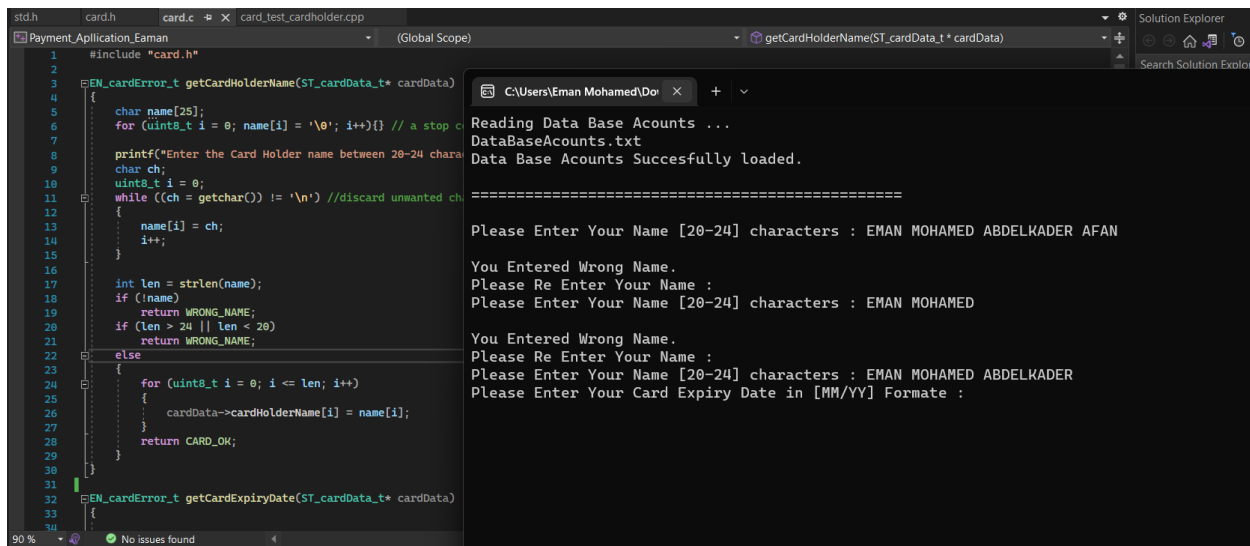
Reading Data Base Accounts ...  
DataBaseAccounts.txt  
Data Base Accounts Succesfully loaded.

=====

Please Enter Your Name [20-24] characters : EMAN MOHAMED

You Entered Wrong Name.  
Please Re Enter Your Name :  
Please Enter Your Name [20-24] characters : |

characters less than 20



```
std.h  card.h  card.c  card_test_cardholder.cpp
Payment_Application_Eaman (Global Scope) getCardHolderName(ST_cardData_t* cardData)
1 #include "card.h"
2
3 EN_cardError_t getCardHolderName(ST_cardData_t* cardData)
4 {
5     char name[25];
6     for (uint8_t i = 0; name[i] = '\0'; i++) {} // a step c
7
8     printf("Enter the Card Holder name between 20-24 characters : ");
9     char ch;
10    uint8_t i = 0;
11    while ((ch = getchar()) != '\n') //discard unwanted characters
12    {
13        name[i] = ch;
14        i++;
15    }
16
17    int len = strlen(name);
18    if (!name)
19        return WRONG_NAME;
20    if (len > 24 || len < 20)
21        return WRONG_NAME;
22    else
23    {
24        for (uint8_t i = 0; i <= len; i++)
25        {
26            cardData->cardHolderName[i] = name[i];
27        }
28        return CARD_OK;
29    }
30 }
31
32 EN_cardError_t getCardExpiryDate(ST_cardData_t* cardData)
33 {
34 }
35 }
```

Reading Data Base Accounts ...  
DataBaseAccounts.txt  
Data Base Accounts Succesfully loaded.

=====

Please Enter Your Name [20-24] characters : EMAN MOHAMED ABDELKADER AFAN

You Entered Wrong Name.  
Please Re Enter Your Name :  
Please Enter Your Name [20-24] characters : EMAN MOHAMED

You Entered Wrong Name.  
Please Re Enter Your Name :  
Please Enter Your Name [20-24] characters : EMAN MOHAMED ABDELKADER  
Please Enter Your Card Expiry Date in [MM/YY] Formate :

characters exceed 24

# Test\_Expiry\_Date

```
std.h card.h card.c card_test_cardholder.cpp
Payment_Application_Eman (Global Scope)
25 {
26     cardData->cardHolderName[i] = name[i];
27 }
28 return CARD_OK;
29 }
30 }
31
32 EN_cardError_t getCardExpiryDate(ST_cardData_t* cardData)
33 {
34     char expDate[100];
35     uint8_t cardMonth, cardYear;
36     for (uint8_t i = 0; i < 100; i++)
37         expDate[i] = '\0';
38
39     printf("Please Enter Your Card Expiry Date in [MM/YY] Formate : *");
40     fseek(stdin, 0, SEEK_END);
41     FILE* stream;
42     if ((stream = fscanf("%s", expDate)) == NULL);
43         return;
44
45     if (!expDate)
46         return WRONG_EXP_DATE;
47
48     else if (expDate[6] != '\0' || expDate[3] == '\0')
49         return WRONG_EXP_DATE;
50
51     else if (expDate[2] != '/')
52         return WRONG_EXP_DATE;
53
54     cardMonth = (expDate[0] - '0') * 10 + (expDate[1] - '0');
55     cardYear = (expDate[3] - '0') * 10 + (expDate[4] - '0');
56
57     if ((cardMonth > 12 || cardMonth < 1) || (cardYear < 0 || cardYear > 99))
58         return WRONG_EXP_DATE;
59
60     cardData->cardExpirationDate[i] = expDate[i];
61
62     return CARD_OK;
63 }
```

Reading Data Base Accounts ...  
DataBaseAccounts.txt  
Data Base Accounts Succesfully loaded.

=====

Please Enter Your Name [20-24] characters : EMAN MOHAMED ABDELKADER  
Please Enter Your Card Expiry Date in [MM/YY] Formate : 23/23

You Entered Wrong EXP Date.  
Please Re Enter Your EXP Date :  
Please Enter Your Card Expiry Date in [MM/YY] Formate : |

Invalid month

```
std.h card.h card.c card_test_cardholder.cpp
Payment_Application_Eman (Global Scope)
34 char expDate[100];
35 uint8_t cardMonth, cardYear;
36 for (uint8_t i = 0; i < 100; i++)
37     expDate[i] = '\0';
38
39 printf("Please Enter Your Card Expiry Date in [MM/YY] Formate : *");
40 fseek(stdin, 0, SEEK_END);
41 FILE* stream;
42 if ((stream = fscanf("%s", expDate)) == NULL);
43     return;
44
45 if (!expDate)
46     return WRONG_EXP_DATE;
47
48 else if (expDate[6] != '\0' || expDate[3] == '\0')
49     return WRONG_EXP_DATE;
50
51 else if (expDate[2] != '/')
52     return WRONG_EXP_DATE;
53
54 cardMonth = (expDate[0] - '0') * 10 + (expDate[1] - '0');
55 cardYear = (expDate[3] - '0') * 10 + (expDate[4] - '0');
56
57 if ((cardMonth > 12 || cardMonth < 1) || (cardYear < 0 || cardYear > 99))
58     return WRONG_EXP_DATE;
59
60 for (uint8_t i = 0; i < 6; i++)
61     cardData->cardExpirationDate[i] = expDate[i];
62
63 return CARD_OK;
64 }
```

Reading Data Base Accounts ...  
DataBaseAccounts.txt  
Data Base Accounts Succesfully loaded.

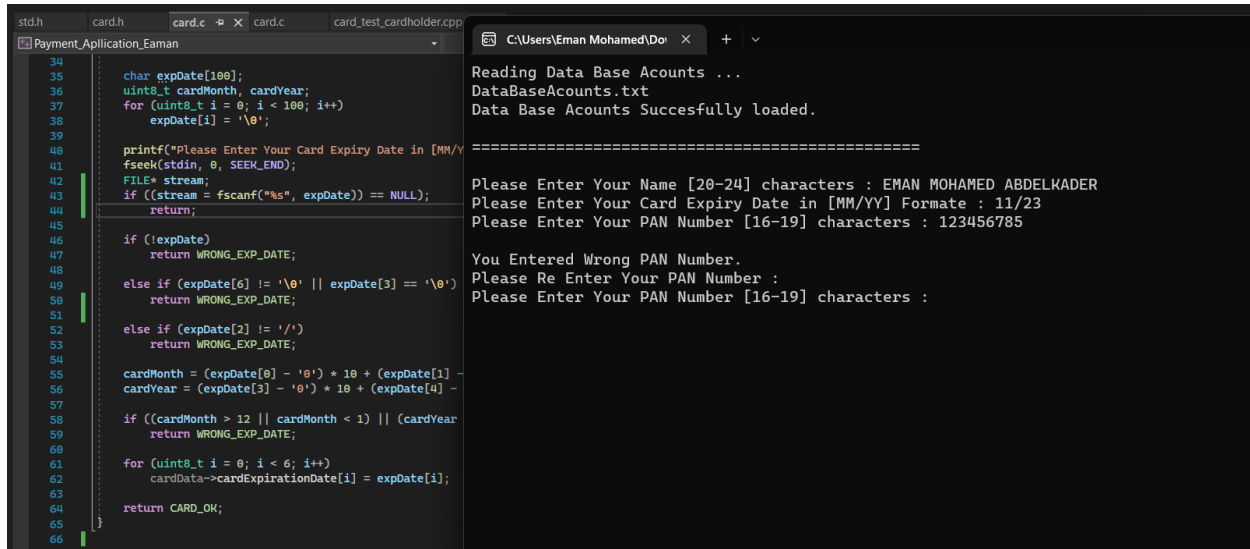
=====

Please Enter Your Name [20-24] characters : EMAN MOHAMED ABDELKADER  
Please Enter Your Card Expiry Date in [MM/YY] Formate : 5/-1

You Entered Wrong EXP Date.  
Please Re Enter Your EXP Date :  
Please Enter Your Card Expiry Date in [MM/YY] Formate : |

Invalid year

# Test\_PAN\_number



```
std.h  card.h  card.c  card_test_cardholder.cpp
Payment_Application_Eaman
34  char expDate[100];
35  uint8_t cardMonth, cardYear;
36  for (uint8_t i = 0; i < 100; i++)
37  {
38      expDate[i] = '\0';
39  }
40
41  printf("Please Enter Your Card Expiry Date in [MM/YY]
42  fseek(stdin, 0, SEEK_END);
43  FILE* stream;
44  if ((stream = fscanf("%s", expDate)) == NULL);
45  return;
46
47  if (!expDate)
48      return WRONG_EXP_DATE;
49
50  else if (expDate[6] != '\0' || expDate[3] == '\0')
51      return WRONG_EXP_DATE;
52
53  else if (expDate[2] != '/')
54      return WRONG_EXP_DATE;
55
56  cardMonth = (expDate[0] - '0') * 10 + (expDate[1] - '0');
57  cardYear = (expDate[3] - '0') * 10 + (expDate[4] - '0');
58
59  if ((cardMonth > 12 || cardMonth < 1) || (cardYear > 99 || cardYear < 0))
60      return WRONG_EXP_DATE;
61
62  for (uint8_t i = 0; i < 6; i++)
63      cardData->cardExpirationDate[i] = expDate[i];
64
65  return CARD_OK;
66
67
```

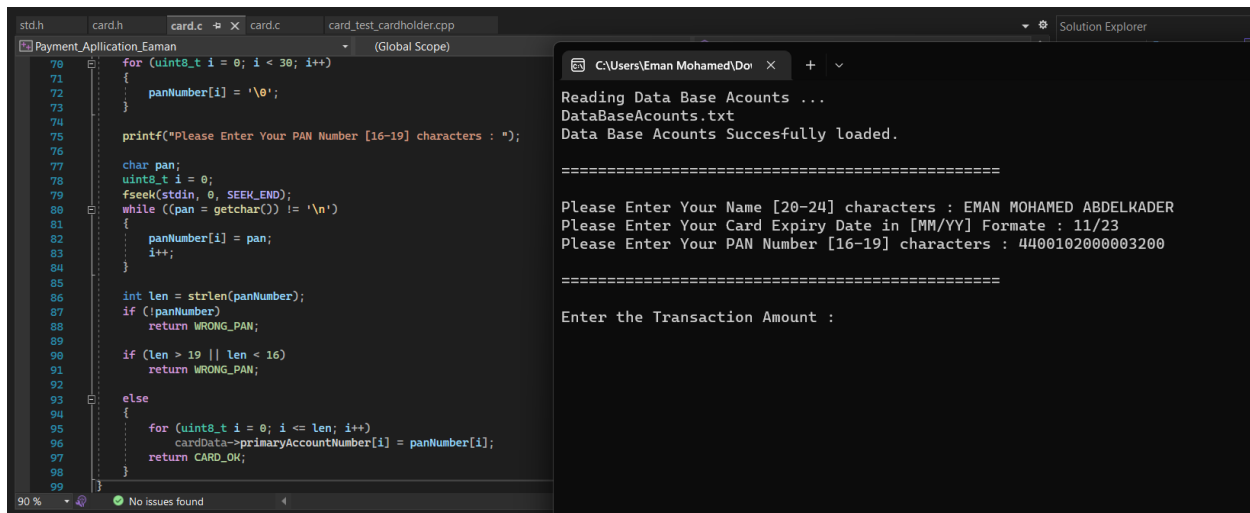
```
C:\Users\Eman Mohamed\Do...
Reading Data Base Accounts ...
DataBaseAccounts.txt
Data Base Accounts Succesfully loaded.

=====

Please Enter Your Name [20-24] characters : EMAN MOHAMED ABDELKADER
Please Enter Your Card Expiry Date in [MM/YY] Formate : 11/23
Please Enter Your PAN Number [16-19] characters : 123456785

You Entered Wrong PAN Number.
Please Re Enter Your PAN Number :
Please Enter Your PAN Number [16-19] characters :
```

less than 16 numbers



```
std.h  card.h  card.c  card_test_cardholder.cpp
Payment_Application_Eaman  (Global Scope)
70  for (uint8_t i = 0; i < 30; i++)
71  {
72      panNumber[i] = '\0';
73  }
74
75  printf("Please Enter Your PAN Number [16-19] characters : ");
76
77  char pan;
78  uint8_t i = 0;
79  fseek(stdin, 0, SEEK_END);
80  while ((pan = getchar()) != '\n')
81  {
82      panNumber[i] = pan;
83      i++;
84  }
85
86  int len = strlen(panNumber);
87  if (!panNumber)
88      return WRONG_PAN;
89
90  if (len > 19 || len < 16)
91      return WRONG_PAN;
92
93  else
94  {
95      for (uint8_t i = 0; i <= len; i++)
96          cardData->primaryAccountNumber[i] = panNumber[i];
97      return CARD_OK;
98  }
99
```

```
C:\Users\Eman Mohamed\Do...
Reading Data Base Accounts ...
DataBaseAccounts.txt
Data Base Accounts Succesfully loaded.

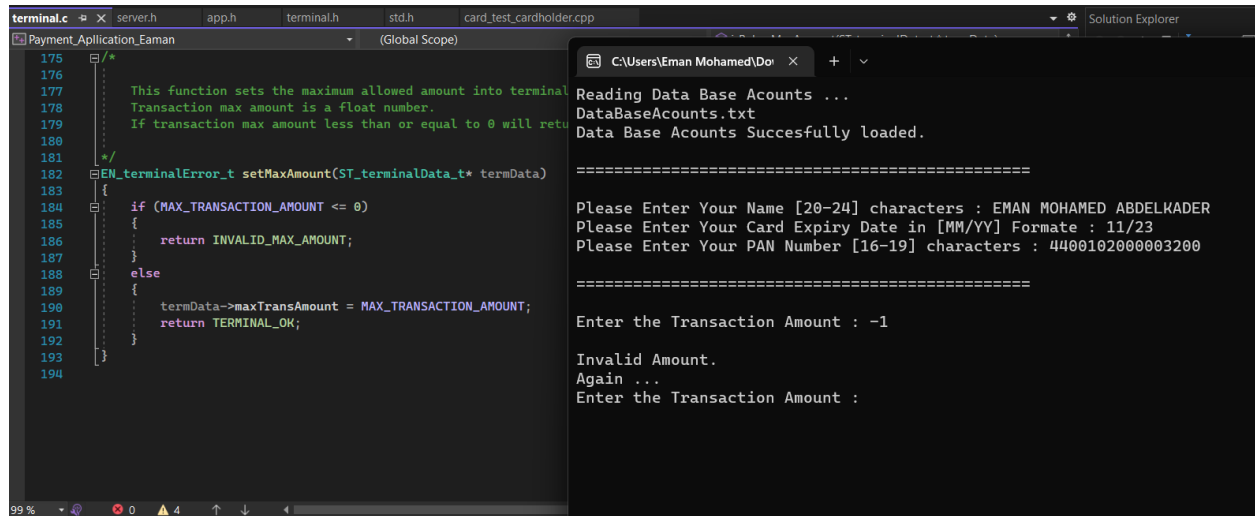
=====

Please Enter Your Name [20-24] characters : EMAN MOHAMED ABDELKADER
Please Enter Your Card Expiry Date in [MM/YY] Formate : 11/23
Please Enter Your PAN Number [16-19] characters : 4400102000003200

=====

Enter the Transaction Amount :
```

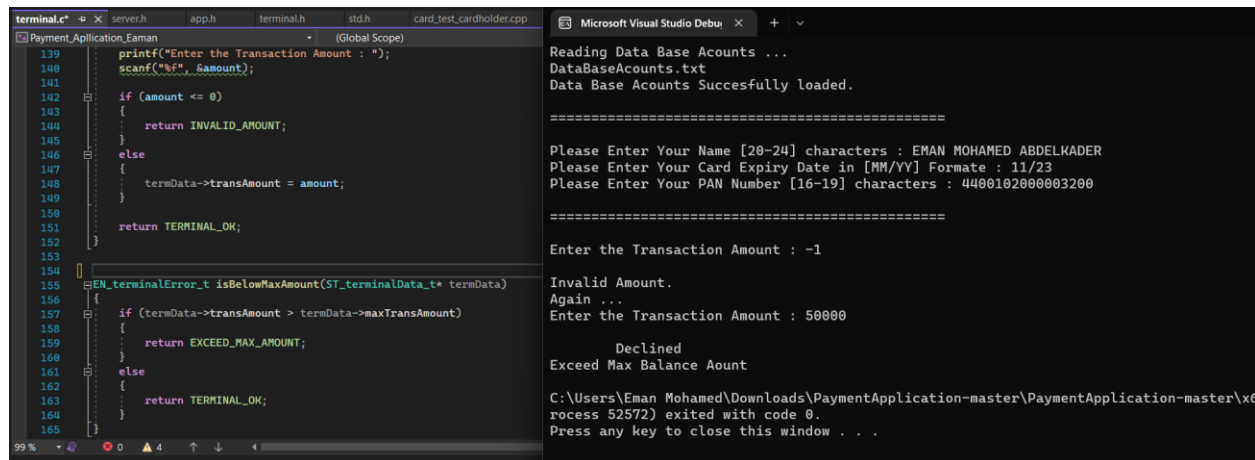
# Test\_Transaction\_Amount



The screenshot shows the Visual Studio IDE with the 'card\_test\_cardholder.cpp' file open. The code defines a function `setMaxAmount` that takes a `ST_terminalData_t` pointer and sets the maximum transaction amount. The function checks if the provided amount is less than or equal to zero; if so, it returns `INVALID_MAX_AMOUNT`. Otherwise, it sets `termData->maxTransAmount` to `MAX_TRANSACTION_AMOUNT` and returns `TERMINAL_OK`.

The output window shows the program's execution: it reads data from 'DataBaseAccounts.txt' successfully, prompts for user details (Name, Card Expiry Date, PAN Number), and then prompts for the transaction amount. When the user enters -1, the program outputs 'Invalid Amount. Again ...' and prompts for the transaction amount again.

less than 0



The screenshot shows the Visual Studio IDE with the 'card\_test\_cardholder.cpp' file open. The code defines a function `isBelowMaxAmount` that takes a `ST_terminalData_t` pointer and checks if the transaction amount is greater than the maximum allowed amount. If so, it returns `EXCEED_MAX_AMOUNT`. Otherwise, it returns `TERMINAL_OK`.

The output window shows the program's execution: it reads data from 'DataBaseAccounts.txt' successfully, prompts for user details, and then prompts for the transaction amount. When the user enters -1, the program outputs 'Invalid Amount. Again ...' and prompts for the transaction amount again. When the user enters 50000, the program outputs 'Declined Exceed Max Balance Amount'.

Exceed max transaction amount