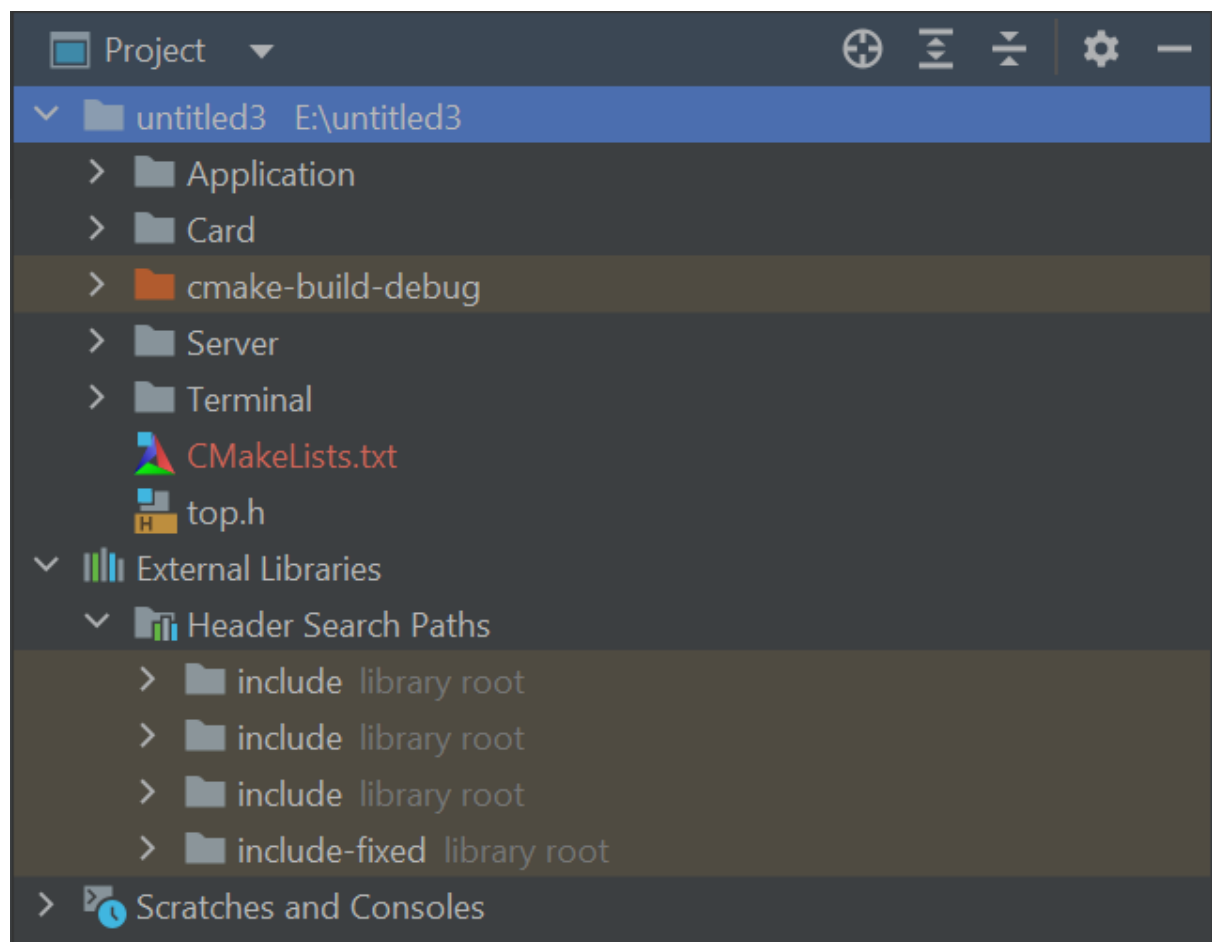
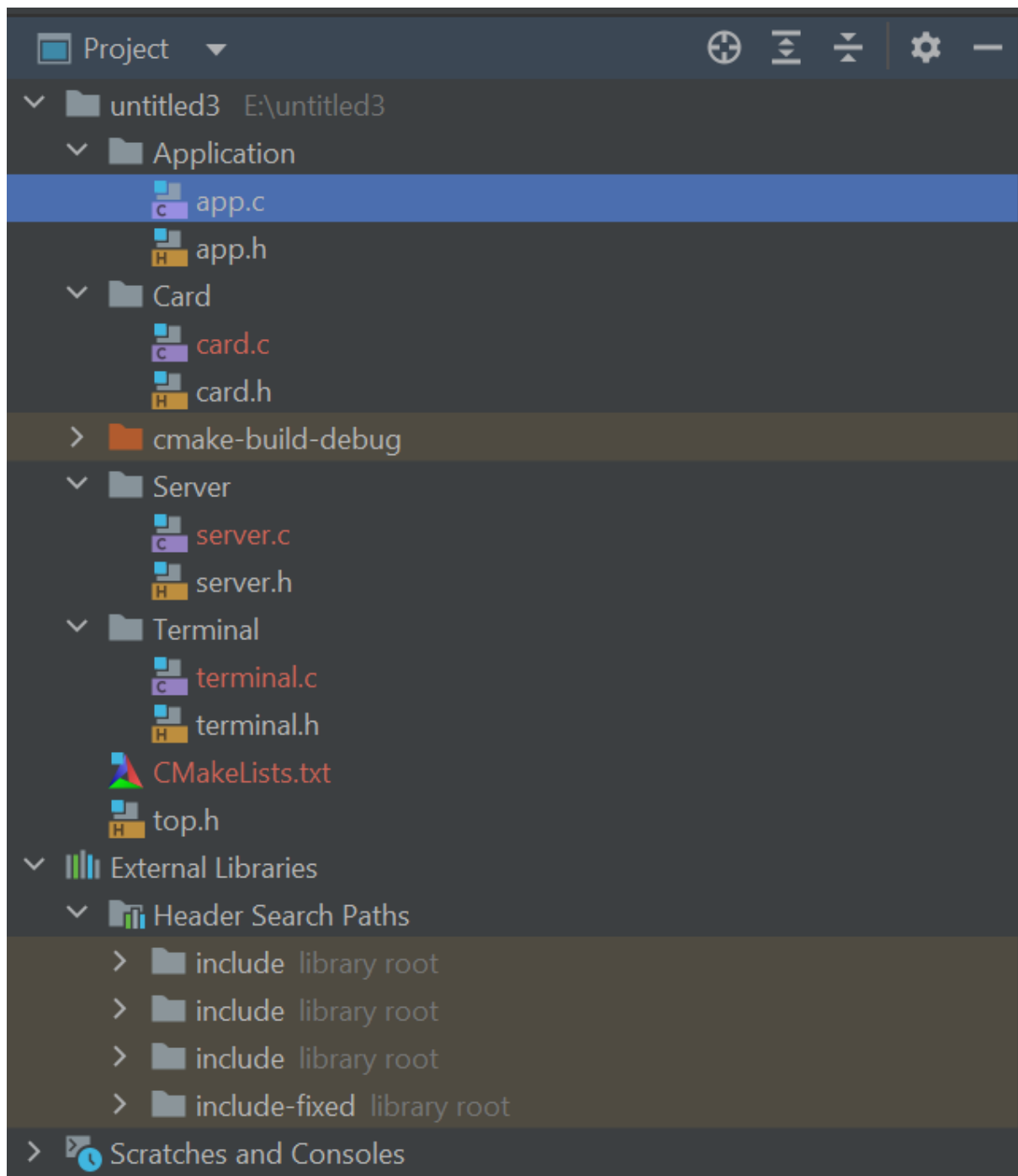


# Payment Application

George Welson





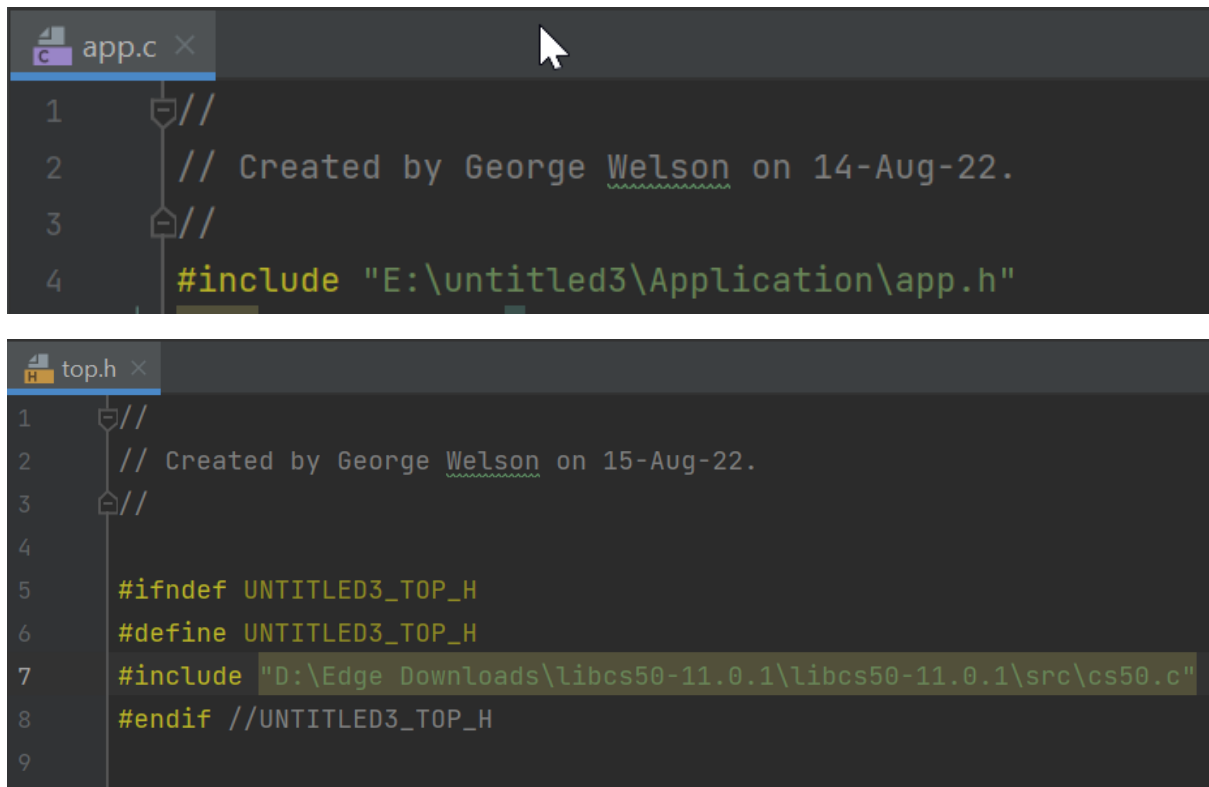
card.h

```
1 //
2 // Created by George Welson on 14-Aug-22.
3 //
4
5 #ifndef UNTITLED3_CARD_H
6 #define UNTITLED3_CARD_H
7
8 #include <stdint.h>
9 #include <string.h>
10 #include <math.h>
11 #include <stdio.h>
12 #include <ctype.h>
13 #include <stdlib.h>
14 #include "E:\untitled3\top.h"
15
100
101 #endif //UNTITLED3_CARD_H
102
```

```
terminal.h x
1 //
2 // Created by George Welson on 14-Aug-22.
3 //
4
5 #ifndef UNTITLED3_TERMINAL_H
6 #define UNTITLED3_TERMINAL_H
7
8 #include <stdint.h>
9 #include <string.h>
10 #include <math.h>
11 #include <stdio.h>
12 #include <ctype.h>
13 #include <stdlib.h>
14 #include "E:\untitled3\top.h"
15
163
164 #endif //UNTITLED3_TERMINAL_H
165
```

```
server.h x
1  //
2  // Created by George Welson on 14-Aug-22.
3  //
4
5  #ifndef UNTITLED3_SERVER_H
6  #define UNTITLED3_SERVER_H
7
8  #include <stdint.h>
9  #include <string.h>
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131  #endif //UNTITLED3_SERVER_H
132
```

```
app.h x
1  //
2  // Created by George Welson on 14-Aug-22.
3  //
4
5  #ifndef UNTITLED3_APP_H
6  #define UNTITLED3_APP_H
7  #include "E:\untitled3\Card\card.h"
8  #include "E:\untitled3\Server\server.h"
9  #include "E:\untitled3\Terminal\terminal.h"
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71  }
72  #endif //UNTITLED3_APP_H
73
```



The image consists of two screenshots of a code editor. The top screenshot shows a file named 'app.c' with the following content:

```
1 //  
2 // Created by George Welson on 14-Aug-22.  
3 //  
4 #include "E:\untitled3\Application\app.h"
```

The bottom screenshot shows a file named 'top.h' with the following content:

```
1 //  
2 // Created by George Welson on 15-Aug-22.  
3 //  
4  
5 #ifndef UNTITLED3_TOP_H  
6 #define UNTITLED3_TOP_H  
7 #include "D:\Edge Downloads\libcs50-11.0.1\libcs50-11.0.1\src\cs50.c"  
8 #endif //UNTITLED3_TOP_H  
9
```

Here you can check the contents of the library cs50.h, I had to use it to implement some of the functions:

[CS50 Library for C — CS50 Docs](#)

<https://github.com/cs50/libcs50>

```
server.h × app.h × app.c × card.h × terminal.h ×
1 //
2 // Created by George Welson on 14-Aug-22.
3 //
4
5 #ifndef UNTITLED3_CARD_H
6 #define UNTITLED3_CARD_H
7
8 #include <stdint.h>
9 #include <string.h>
10 #include <math.h>
11 #include <stdio.h>
12 #include <ctype.h>
13 #include <stdlib.h>
14 #include "E:\untitled3\top.h"
15 typedef struct ST_cardData_t
16 {
17     uint8_t cardHolderName[25];
18     uint8_t primaryAccountNumber[20];
19     uint8_t cardExpirationDate[6];
20 }ST_cardData_t;
21
22 typedef enum EN_cardError_t
23 {
24     OK_c, WRONG_NAME, WRONG_EXP_DATE, WRONG_PAN
25 }EN_cardError_t;
26
```



```
server.h x app.h x app.c x card.h x terminal.h x
Project
Commit
Pull Requests
25 }EN_cardError_t;
26
27 EN_cardError_t getCardHolderName(ST_cardData_t *cardData){
28     strcpy(cardData->cardHolderName ,get_string(NULL,"Enter Card Holder Name:\n"));
29
30
31     int n=strlen(cardData->cardHolderName);
32     for(int i=0;i<n;i++){
33         if(isalpha(cardData->cardHolderName[i])==0 && cardData->cardHolderName[i]!=' '){
34             return WRONG_NAME;
35         }
36     }
37     if(cardData->cardHolderName[0]!='\0')
38         return WRONG_NAME;
39     if( n<20 || n>24 ){
40         return WRONG_NAME;
41     }
42     else{
43         return OK_c;
44     }
45 }
46
```

```
server.h x app.h x app.c x card.h x terminal.h x
Project
Commit
Pull Requests
46
47 EN_cardError_t getCardExpiryDate(ST_cardData_t *cardData){
48     strcpy(cardData->cardExpirationDate ,get_string(NULL,"Enter Card Expiration Date:\n"));
49     int n=strlen(cardData->cardExpirationDate);
50     if(n!=5 || cardData->cardExpirationDate[0]!='\0') return WRONG_EXP_DATE;
51     int flag=0;
52     for(int i=0;i<n;i++){
53         if(i!=2 && isdigit(cardData->cardExpirationDate[i])==0){
54             flag =1;
55             break;
56         }
57         if(cardData->cardExpirationDate[2]!='/' ) {
58             flag =1;
59             break;
60         }
61     }
62     int date[2];
63     string str;
64     strcpy(str,cardData->cardExpirationDate);
65     char* token = strtok(str, "/");
66     int i=0;
67     while (token != NULL && i<2) {
68         date[i]= atoi(token);
69
70         token = strtok(NULL, "/");
71         i++;
72     }

```

```

70     token = strtok(NULL, "/");
71     i++;
72 }
73 if(date[0]>12 || date[0]<1) return WRONG_EXP_DATE;
74 if(date[1]>99 || date[1]<0) return WRONG_EXP_DATE;
75 if(flag ==1){
76     return WRONG_EXP_DATE;
77 }
78 else return OK_c;
79 }
80
81 EN_cardError_t getCardPAN(ST_cardData_t *cardData){
82     strcpy(cardData->primaryAccountNumber, get_string(NULL, "Enter Primary Account Number:\n"));
83     int n=strlen(cardData->primaryAccountNumber);
84
85     if(n<16 || n> 19 || cardData->primaryAccountNumber[0]!='\0'){
86         return WRONG_PAN;
87     }
88     int flag =0;
89     for(int i=0;i<n;i++){
90         if(isalnum(cardData->primaryAccountNumber[i])==0){
91             flag=1;
92             break;
93         }
94     }
95     if(flag==0)
96         return OK_c;
97
98 }
99
100 #endif //UNTITLED3_CARD_H

```

```
server.h × app.h × app.c × card.h × terminal.h ×
1 //
2 // Created by George Welson on 14-Aug-22.
3 //
4
5 #ifndef UNTITLED3_TERMINAL_H
6 #define UNTITLED3_TERMINAL_H
7
8 #include <stdint.h>
9 #include <string.h>
10 #include <math.h>
11 #include <stdio.h>
12 #include <ctype.h>
13 #include <stdlib.h>
14 #include "E:\untitled3\top.h"
15
16 typedef enum EN_terminalError_t
17 {
18     OK_t, WRONG_DATE, EXPIRED_CARD, INVALID_CARD, INVALID_AMOUNT, EXCEED_MAX_AMOUNT, INVALID_MAX_AMOUNT
19 } EN_terminalError_t;
20
21 int isleap(int y) {
22     if((y % 4 == 0) && (y % 100 != 0) && (y % 400 == 0))
23         return 1;
24     else
25         return 0;
```

```
server.h × app.h × app.c × card.h × terminal.h ×
19 }EN_terminalError_t;
20
21 int isleap(int y) {
22     if((y % 4 == 0) && (y % 100 != 0) && (y % 400 == 0))
23         return 1;
24     else
25         return 0;
26 }
27
28 int isValid(int d, int m, int y) {
29     if(y < 2000 || y > 2100)
30         return 0;
31     if(m < 1 || m > 12)
32         return 0;
33     if(d < 1 || d > 31)
34         return 0;
35     if( m == 2 ) {
36         if(isleap(y)) {
37             if(d <= 29)
38                 return 1;
39             else
40                 return 0;
41         }
42     }
```

```

43     if ( m == 4 || m == 6 || m == 9 || m == 11 )
44         if(d <= 30)
45             return 1;
46         else
47             return 0;
48     return 1;
49 }
50
51 EN_terminalError_t getTransactionDate(ST_terminalData_t *termData){
52     string expdate;
53     expdate=strcopy(termData->transactionDate,get_string(NULL,"Enter Transaction Date:\n"));
54     int n= strlen(termData->transactionDate);
55     if(termData->transactionDate[0]!='\0') return WRONG_DATE;
56     if (n!=10) return WRONG_DATE;
57     for(int i=0;i<n;i++){
58         if(i!=2 && i!=5 && isdigit(termData->transactionDate[i])==0)return WRONG_DATE;
59         if(i==2 || i==5){
60             if(termData->transactionDate[i]!='/') return WRONG_DATE;
61         }
62     }
63
64
65
66     int date[3];
67     char* token = strtok(expdate, "/");

```

```

65
66     int date[3];
67     char* token = strtok(expdate, "/");
68     int i=0;
69     while (token != NULL && i<3) {
70         date[i]= atoi(token);
71
72         token = strtok(NULL, "/");
73         i++;
74     }
75     for (int i=0;i<n;i++){
76         if(expdate[i]!='\0'){
77             expdate[i]='/';
78         }
79     }
80     if(isvalid(date[0],date[1],date[2])==0) return WRONG_DATE;
81     else return OK_t;
82 }
83
84 EN_terminalError_t isCardExpired(ST_cardData_t cardData, ST_terminalData_t termData){
85     int exp[2];
86     int trans[2];
87     char* tokene = strtok(cardData.cardExpirationDate, "/");
88     int i=0;

```

```
server.h × app.h × app.c × card.h × terminal.h ×
Project
Commit
Pull Requests
Structure
Bookmarks

85     int exp[2];
86     int trans[2];
87     char* tokene = strtok(cardData.cardExpirationDate, "/");
88     int i=0;
89     while (tokene != NULL && i<2) {
90         exp[i]= atoi(tokene);
91
92         tokene = strtok(NULL, "/");
93         i++;
94     }
95     char* token= strtok(&termData.transactionDate[3], "/");
96     int j=0;
97     while(token!=NULL&&j<2){
98         trans[j]=atoi(token);
99         token = strtok(NULL, "/");
100        ++j;
101    }
102    if((exp[1]+2000)<trans[1]){
103        return EXPIRED_CARD;
104    }
105    else if(exp[1]==trans[1]){
106        if(exp[0]<trans[0]){
107            return EXPIRED_CARD;
108        }
109    }
```

```
server.h x app.h x app.c x card.h x terminal.h x
Project
Commit
Pull Requests
Bookmarks
109     }
110     return OK_t;
111 }
112
113 EN_terminalError_t isValidCardPAN(ST_cardData_t *cardData){
114
115     int n = strlen(cardData->primaryAccountNumber);
116
117     int nSum = 0, isSecond = false;
118     for (int i = n - 1; i >= 0; i--) {
119         int d = cardData->primaryAccountNumber[i] - '0';
120         if (isSecond == true)
121             d = d * 2;
122
123         nSum += d / 10;
124         nSum += d % 10;
125
126         isSecond = !isSecond;
127     }
128     if(nSum % 10 == 0) {
129         return OK_t;
130     } else return INVALID_CARD ;
131 }
132
```

```
server.h × app.h × app.c × card.h × terminal.h ×
130     else return INVALID_CARD ;
131 }
132
133 EN_terminalError_t getTransactionAmount(ST_terminalData_t *termData){
134     float in= get_float("Enter Transaction Amount:\n");
135     if(in<=0) return INVALID_AMOUNT;
136     termData->transAmount=in;
137     return OK_t;
138 }
139
140 EN_terminalError_t isBelowMaxAmount(ST_terminalData_t *termData){
141     if(termData->transAmount>termData->maxTransAmount) return EXCEED_MAX_AMOUNT;
142     else return OK_t;
143 }
144
145 EN_terminalError_t setMaxAmount(ST_terminalData_t *termData){
146     float in= get_float("Enter Max Amount:\n");
147     if(in>0){
148         termData->maxTransAmount=in;
149         return OK_t;
150     }
151     return INVALID_MAX_AMOUNT;
152 }
153
154 #endif //UNTITLED3_TERMINAL_H
```



```
server.h × app.h × app.c ×
Project
Commit
Pull Requests
Structure
Bookmarks

1 //
2 // Created by George Welton on 14-Aug-22.
3 //
4
5 #ifndef UNTITLED3_SERVER_H
6 #define UNTITLED3_SERVER_H
7
8 #include <stdint.h>
9 #include <string.h>
10
11 typedef enum EN_transState_t
12 {
13     APPROVED, DECLINED_INSUFFECIENT_FUND, DECLINED_STOLEN_CARD, INTERNAL_SERVER_ERROR
14 } EN_transStat_t;
15
16 typedef enum EN_serverError_t
17 {
18     OK, SAVING_FAILED, TRANSACTION_NOT_FOUND, ACCOUNT_NOT_FOUND, LOW_BALANCE
19 } EN_serverError_t ;
20
21 typedef struct ST_accountsDB_t
22 {
23     float balance;
24     uint8_t primaryAccountNumber[20];
25 } ST_accountsDB_t;
26
27 typedef struct ST_terminalData_t
28 {
29     float transAmount;
30     float maxTransAmount;
```

```
25 }ST_accountsDB_t;
26
27 typedef struct ST_terminalData_t
28 {
29     float transAmount;
30     float maxTransAmount;
31     uint8_t transactionDate[11];
32 }ST_terminalData_t;
33
34 typedef struct ST_transaction_t
35 {
36     ST_cardData_t cardHolderData;
37     ST_terminalData_t terminalData;
38     enum EN_transState_t transState;
39     uint32_t transactionSequenceNumber;
40 }ST_transaction_t;
41
42 ST_accountsDB_t accgloal[255]={
43     {
44         8000.0,"7502289947808108"
45     },{
46         755.0,"51714505293369193"
47     },{
48         4563.4,"1332184076269645697"
49     }
50 };
51
52 struct ST_transaction_t transactionglobal[255]={
53     {
54         {"0","0","0"},{0.0,0.0,"0"},0,0
```

```

server.h × app.h × app.c ×
52 struct ST_transaction_t transactionGlobal[255]={
53     {
54         {"0","0","0"},{0.0,0.0,"0"},0,0
55     },{
56         {"0","0","0"},{0.0,0.0,"0"},0,0
57     },{
58         {"0","0","0"},{0.0,0.0,"0"},0,0
59     },{
60         {"0","0","0"},{0.0,0.0,"0"},0,0
61     },{
62         {"0","0","0"},{0.0,0.0,"0"},0,0
63     },{
64         {"0","0","0"},{0.0,0.0,"0"},0,0
65     }
66 };
67
68 EN_serverError_t isValidAccount(ST_cardData_t *cardData){
69     for(int i=0;i<255;i++){
70         if(strcmp(cardData->primaryAccountNumber, accglobal[i].primaryAccountNumber)==0){
71             return OK;
72         }
73     }
74     return ACCOUNT_NOT_FOUND;
75 }
76
77 EN_serverError_t isAmountAvailable(ST_terminalData_t *termData, ST_accountsDB_t *accountsDb){
78     if(termData->transAmount<accountsDb->balance) return OK;
79     return LOW_BALANCE;
80 }
81

```

```

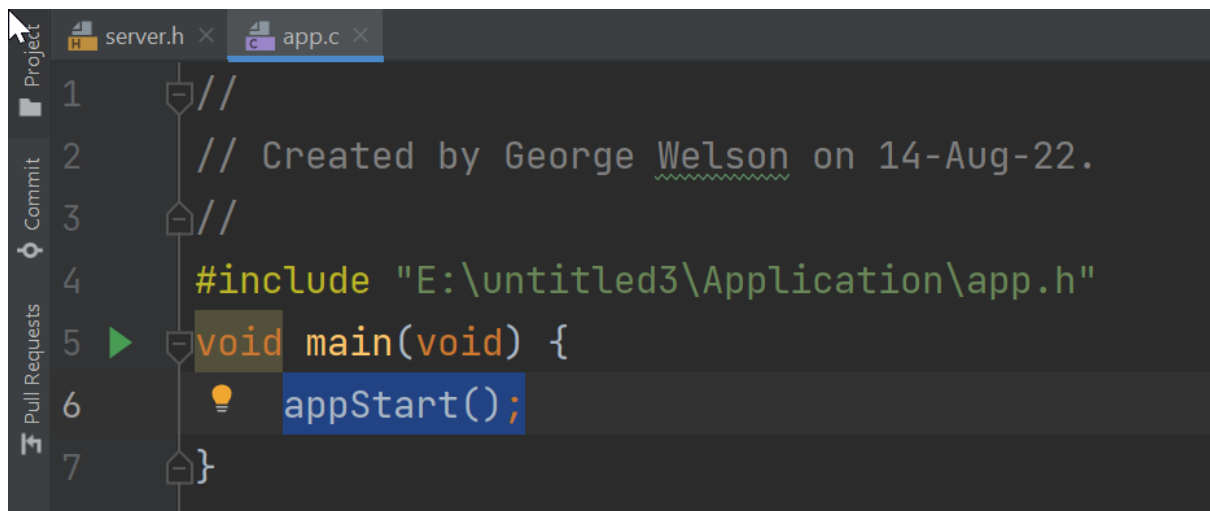
server.h × app.h × app.c ×
79     return LOW_BALANCE;
80 }
81
82 EN_serverError_t saveTransaction(struct ST_transaction_t *transData){
83     static int generator=1000;
84     generator++;
85     transData->transactionSequenceNumber=generator;
86
87     for(int i=0;i<255;i++){
88         if(transactionGlobal[i].transactionSequenceNumber==0){
89             transactionGlobal[i]=*transData;
90             return OK;
91         }
92     }
93     return SAVING_FAILED;
94 }
95
96 enum EN_transState_t receiveTransactionData(struct ST_transaction_t *transData){
97     if(isValidAccount(&transData->cardHolderData)==ACCOUNT_NOT_FOUND){
98         transData->transState=DECLINED_STOLEN_CARD;
99         return DECLINED_STOLEN_CARD;
100     }
101     for(int i=0;i<255;i++){
102         if (strcmp(transData->cardHolderData.primaryAccountNumber, accglobal[i].primaryAccountNumber) == 0) {
103             if (isAmountAvailable(&transData->terminalData, &accglobal[i]) == LOW_BALANCE) {
104                 transData->transState = DECLINED_INSUFFECIENT_FUND;
105                 return DECLINED_INSUFFECIENT_FUND;
106             } else {
107                 accglobal[i].balance -= transData->terminalData.transAmount;
108                 if (saveTransaction(transData) == SAVING_FAILED) {

```

```
server.h x app.h x app.c x
105         return DECLINED_INSUFFECIENT_FUND;
106     } else {
107         accglobal[i].balance -= transData->terminalData.transAmount;
108         if (saveTransaction(transData) == SAVING_FAILED) {
109             transData->transState = INTERNAL_SERVER_ERROR;
110             return INTERNAL_SERVER_ERROR;
111         }
112         return APPROVED;
113     }
114 }
115 }
116 }
117
118 EN_serverError_t getTransaction(uint32_t transactionSequenceNumber, struct ST_transaction_t *transData){
119
120     for(int i=0;i<255;i++){
121         if(transactionSequenceNumber==transactionglobal[i].transactionSequenceNumber) return OK;
122     }
123     transData->transactionSequenceNumber=transactionSequenceNumber;
124
125     return TRANSACTION_NOT_FOUND;
126 }
127
128 #endif //UNTITLED3_SERVER_H
129
```

```
server.h × app.h × app.c ×
E:\untitled3\Server\server.h
2 // Created by George Welson on 14-Aug-22.
3 //
4
5 #ifndef UNTITLED3_APP_H
6 #define UNTITLED3_APP_H
7 #include "E:\untitled3\Card\card.h"
8 #include "E:\untitled3\Server\server.h"
9 #include "E:\untitled3\Terminal\terminal.h"
10
11 void appStart(void){
12     ST_transaction transaction1;
13     if(getCardHolderName(&transaction1.cardHolderData)!=OK_c){
14         printf("Invalid Name\n");
15         return;
16     };
17     if(getCardExpiryDate(&transaction1.cardHolderData)!=OK_c){
18         printf("Invalid Expiry Date\n");
19         return;
20     };
21     if(getCardPAN(&transaction1.cardHolderData)!=OK_c){
22         printf("Invalid Primary Account Number\n");
23         return;
24     };
25     if(getTransactionDate(&transaction1.terminalData)!=OK_t){
26         printf("Wrong Date\n");
27         return;
28     };
29     if(setMaxAmount(&transaction1.terminalData)!=OK_t){
30         printf("Invalid Max Amount\n");
31         return;
32     };
33     if(isValidCardPAN(&transaction1.cardHolderData)!=OK_t) {
34         printf("Declined Invalid Primary Account Number\n");
35         return;
36     }
```

```
server.h × app.h × app.c ×
35     return;
36 }
37 if(isCardExpired(transaction1.cardHolderData,transaction1.terminalData)!=OK_t) {
38     printf("Declined Expired Card\n");
39     return;
40 }
41 if(getTransactionAmount(&transaction1.terminalData)!=OK_t){
42     printf("Declined Invalid Amount\n");
43     return;
44 };
45 if(isBelowMaxAmount(&transaction1.terminalData)!=OK_t) {
46     printf("Declined Amount Exceeding Limit\n");
47     return;
48 }
49 switch (receiveTransactionData(&transaction1)){
50     case DECLINED_STOLEN_CARD:
51         printf("Declined Stolen Card\n");
52         return;
53     case DECLINED_INSUFFECIENT_FUND:
54         printf("Declined Insufficient Fund\n");
55         return;
56     case INTERNAL_SERVER_ERROR:
57         printf("Internal Server Error\n");
58         return;
59     default:
60         printf("Approved\n");
61         printf("Approved\n");
62 }
63 switch (getTransaction(get_int("Enter Transaction Sequence Number\n"),&transaction1)){
64     case TRANSACTION_NOT_FOUND:
65         printf("Transaction Not Found\n");
66         return;
67     case OK:
68         printf("Transaction Found\n");
69         return;
70 }
71 }
72 #endif //UNTITLED3_APP_H
73
```



```
1 //  
2 // Created by George Welson on 14-Aug-22.  
3 //  
4 #include "E:\untitled3\Application\app.h"  
5 void main(void) {  
6     appStart();  
7 }
```