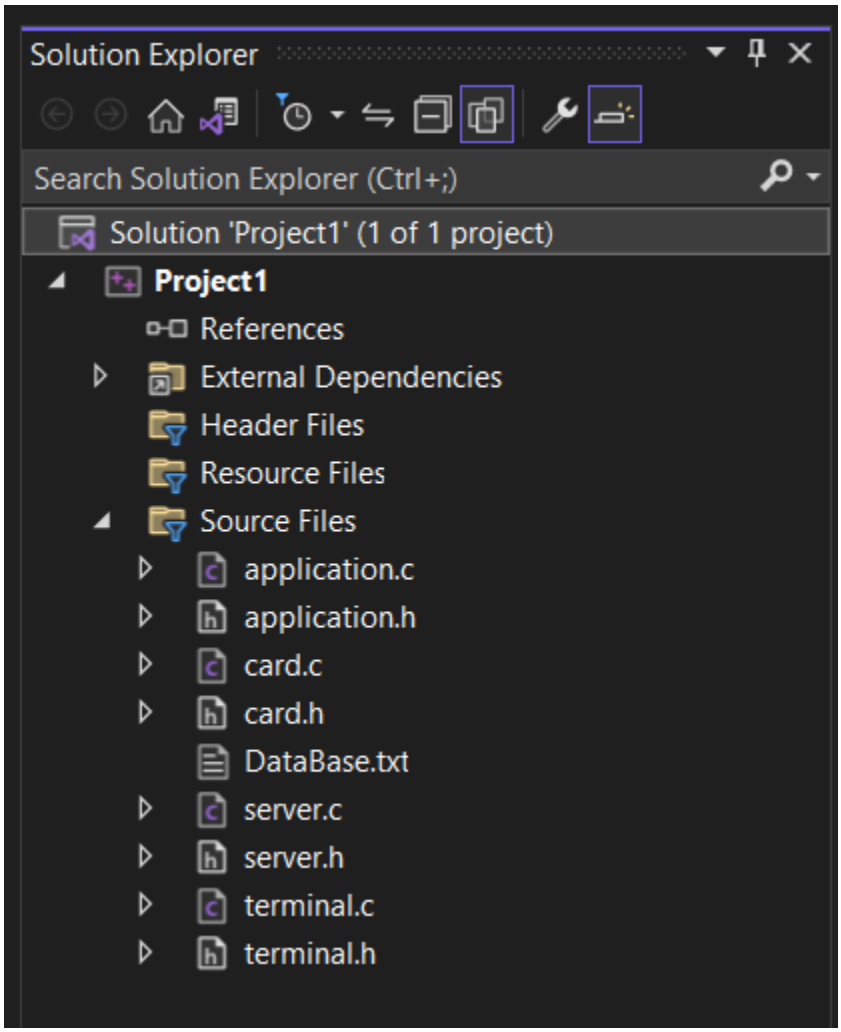
Embedded Systems Professional

Project 1

Submitted by:

Mostafa Ashraf Ebrahim Ali ELFEEL

Screenshot of the solution explorer clarified:



card.h file:

Project1 (Global Scope)

```
#ifndef card_h
#define card_h
#define _CRT_SECURE_NO_WARNINGS

#include<stdio.h>
#include<string.h>
#include<time.h>

typedef unsigned char uint8_t;
typedef signed char int8_t;
typedef unsigned short uint16_t;
typedef signed short int16_t;
typedef unsigned int uint32_t;
typedef signed int int32_t;
typedef unsigned long long uint64_t;
typedef signed long long int64_t;

typedef struct ST_cardData_t
{
    uint8_t cardHolderName[25];
    uint8_t primaryAccountNumber[20];
    uint8_t cardExpirationDate[6];
}ST_cardData_t;

ST_cardData_t cardData;

uint16_t expmonth;
uint16_t expyear;

uint16_t trnmonth;
uint16_t trnyear;

typedef enum EN_cardError_t
{
    OK, WRONG_NAME, WRONG_EXP_DATE, WRONG_PAN
}EN_cardError_t;

EN_cardError_t getCardHolderName(ST_cardData_t* cardData);
EN_cardError_t getCardExpiryDate(ST_cardData_t* cardData);
EN_cardError_t getCardPAN(ST_cardData_t* cardData);

#endif
```

terminal.h file:

roject1 (Global Scope)

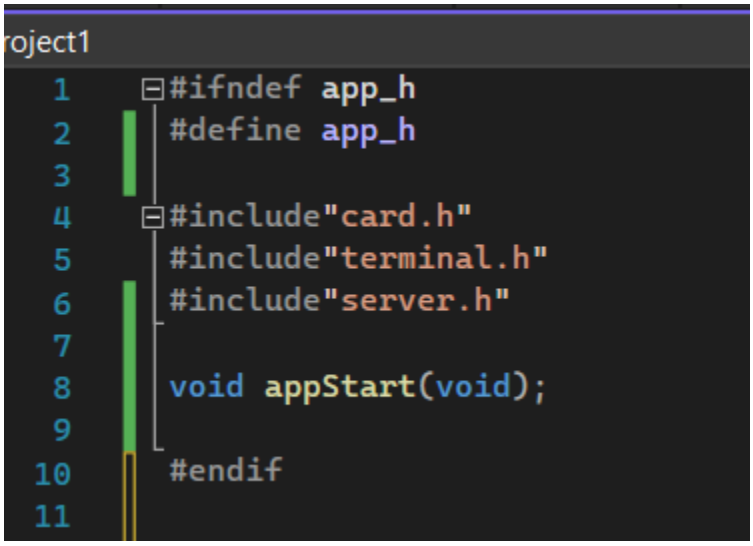
```
#ifndef terminal_h
#define terminal_h

#include "card.h"

typedef enum EN_terminalError_t
{
    OKEY, WRONG_DATE, EXPIRED_CARD, INVALID_CARD, INVALID_AMOUNT, EXCEED_MAX_AMOUNT, INVALID_MAX_AMOUNT
}EN_terminalError_t;

typedef struct ST_terminalData_t
{
    float transAmount;
    float maxTransAmount;
    uint8_t transactionDate[11];
}ST_terminalData_t;

ST_terminalData_t termData;

EN_terminalError_t getTransactionDate(ST_terminalData_t*termData);
EN_terminalError_t isCardExpired(ST_cardData_t *cardData, ST_terminalData_t*termData);
EN_terminalError_t getTransactionAmount(ST_terminalData_t*termData);
EN_terminalError_t isBelowMaxAmount(ST_terminalData_t*termData);
EN_terminalError_t setMaxAmount(ST_terminalData_t*termData);

#endif
```

server.h file:

roject1 (Global Scope)

```
#ifndef server_h
#define server_h

#include"card.h"
#include"terminal.h"
#include <math.h>
#define DataBaseSize 255
uint32_t PAN_validation;
uint32_t array_number;
uint32_t eofloop;
_Bool flag;

typedef enum EN_transState_t
{
    APPROVED, DECLINED_INSUFFECIENT_FUND, INTERNAL_SERVER_ERROR
}EN_transState_t;

typedef enum EN_serverError_t
{
    OKK, SAVING_FAILED, TRANSACTION_NOT_FOUND, ACCOUNT_NOT_FOUND, LOW_BALANCE, DECLINED_STOLEN_CARD
}EN_serverError_t;

typedef struct ST_accountsDB_t
{
    float balance;
    uint8_t primaryAccountNumber[20];
}ST_accountsDB_t;

ST_accountsDB_t serverData[DataBaseSize];

void ReloadDataBase(ST_accountsDB_t serverData[DataBaseSize]);
EN_serverError_t isValidAccount(ST_cardData_t* cardData, ST_accountsDB_t serverData[DataBaseSize]);
EN_serverError_t isAmountAvailable(ST_terminalData_t* termData, ST_accountsDB_t serverData[DataBaseSize]);
void saveTransaction(ST_accountsDB_t serverData[DataBaseSize]);

```

application.h file:



getCardHolderName function:

```
EN_cardError_t getCardHolderName(ST_cardData_t* cardData) {
    printf("Please Enter Your Name (20 to 24 characters)...\n");
    gets(cardData->cardHolderName);
    if (strlen(cardData->cardHolderName) > 19 && strlen(cardData->cardHolderName) < 25) {
        puts("CORRECT_FORMAT\n");
        return OK;
    }
    else {
        puts("WRONG_NAME_FORMAT\n");
        return WRONG_NAME;
    }
}
```

getCardExpiryDate function:

```
EN_cardError_t getCardExpiryDate(ST_cardData_t* cardData) {
    printf("Please Enter Your Exp Date (MN/YR Format)...\n");
    gets(cardData->cardExpirationDate);
    expmonth = ((cardData->cardExpirationDate[0] - '0') * 10) + (cardData->cardExpirationDate[1] - '0');
    expyear = ((cardData->cardExpirationDate[3] - '0') * 10) + (cardData->cardExpirationDate[4] - '0');
    if (expmonth >= 1 && expmonth <= 12 && cardData->cardExpirationDate[2] == '/' ) {
        puts("CORRECT_FORMAT\n");
        return OK;
    }
    else {
        puts("WRONG_DATE_FORMAT\n");
        return WRONG_EXP_DATE;
    }
}
```

getCardPAN Function:

```
EN_cardError_t getCardPAN(ST_cardData_t* cardData) {
    printf("Please Enter Your Primary Account Number (16 to 19 characters)...\n");
    gets(cardData->primaryAccountNumber);
    if (strlen(cardData->primaryAccountNumber) > 15 && strlen(cardData->primaryAccountNumber) < 20) {
        puts("CORRECT_FORMAT\n");
        return OK;
    }
    else {
        puts("WRONG_PAN_FORMAT\n");
        return WRONG_PAN;
    }
}
```

getTransactionDate function:

```
EN_terminalError_t getTransactionDate(ST_terminalData_t* termData) {
    printf("Getting Time...");
    time_t rawtime = time(NULL);
    struct tm* ptm = localtime(&rawtime);
    strftime(termData->transactionDate, 11, "%d/%m/%Y", ptm);
    puts(termData->transactionDate);
    return OKEY;
}
```

isCardExpired function:

```
EN_terminalError_t isCardExpired(ST_cardData_t *cardData, ST_terminalData_t *termData) {

    trnmonth = ((termData->transactionDate[3] - '0') * 10) + (termData->transactionDate[4] - '0');
    trnyear = ((termData->transactionDate[8] - '0') * 10) + (termData->transactionDate[9] - '0');

    if (expyear > trnyear || (expyear == trnyear && expmonth >= trnmonth)) {
        printf("CARD_NOT_EXPIRED\n");
        return OKEY;
    }
    else {
        printf("EXPIRED_CARD\n");
        return EXPIRED_CARD;
    }
}
```

getTransactionAmount function:

```
EN_terminalError_t getTransactionAmount(ST_terminalData_t *termData) {
    printf("Please Enter Transaction Amount (<5000)...\n");
    scanf("%f", &termData->transAmount);
    if (termData->transAmount <= 0) {
        return INVALID_AMOUNT;
    }
    else {
        return OKEY;
    }
}
```

setMaxAmount function:

```
39  EN_terminalError_t setMaxAmount(ST_terminalData_t *termData) {
40      termData->maxTransAmount = 5000;
41      return OKEY;
42  }
```

isBelowMaxAmount function:

```
44
45  EN_terminalError_t isBelowMaxAmount(ST_terminalData_t *termData) {
46      if (termData->transAmount > termData->maxTransAmount) {
47          printf("DECLINED_AMOUNT_EXCEEDING_LIMIT\n");
48          return EXCEED_MAX_AMOUNT;
49      }
50      else {
51          printf("AMOUNT_ACCEPTED\n");
52          return OKEY;
53      }
54  }
```

ReloadDataBase function:

```
void ReloadDataBase(ST_accountsDB_t serverData[DataBaseSize]) {
    uint8_t RelaoadedData[35];
    uint32_t PANloop;
    uint32_t DBcounter = 0;
    uint32_t DecimalCounter=0;
    float Decimal;
    FILE* ReadDB;
    ReadDB = fopen("DataBase.txt", "r");
    for (DBcounter ; DBcounter < DataBaseSize; DBcounter++) {
        serverData[DBcounter].balance = 0;
    }
    for (eofloop = 0; fgets(RelaoadedData,34, ReadDB); eofloop++) {
        PANloop = 0;
        while(RelaoadedData[PANloop]!='-') {
            serverData[eofloop].primaryAccountNumber[PANloop] = RelaoadedData[PANloop];
            PANloop++;
        }
        PANloop++;
        for (PANloop; RelaoadedData[PANloop] != '.'; PANloop++) {
            serverData[eofloop].balance = (serverData[eofloop].balance * 10) +(RelaoadedData[PANloop]-'0');
        }
        PANloop++;
        for (PANloop; RelaoadedData[PANloop] != '-'; PANloop++) {
            DecimalCounter++;
            Decimal = (RelaoadedData[PANloop] - '0');
            serverData[eofloop].balance = (serverData[eofloop].balance)+ (Decimal/pow(10, DecimalCounter));
        }
    }
    fclose(ReadDB);
}
```

isValidAccount function:

```
EN_serverError_t isValidAccount(ST_cardData_t* cardData, ST_accountsDB_t serverData[DataBaseSize]) {
    printf("Returning to Server to Validate Your Account...\n");
    for (uint32_t loop = 0; loop < 255; loop++) {
        flag = 0;
        PAN_validation = strcmp(cardData->primaryAccountNumber, serverData[loop].primaryAccountNumber);
        if (PAN_validation == 0) {
            flag = 1;
            array_number = loop;
            break;
        }
    }
    printf("Your Account Is Being Validated and Verified\n");
    if (flag == 1) {
        flag = 0;
        printf("ACCEPTED(NOT_STOLEN_CARD)\n");
        return OKK;
    }else {
        printf("DECLINED(STOLEN_CARD)\n");
        return DECLINED_STOLEN_CARD;
    }
}
```

isAmountAvailable function:

```
57  EN_serverError_t isAmountAvailable(ST_terminalData_t* termData, ST_accountsDB_t serverData[DataBaseSize]) {
58      if (termData->transAmount > serverData[array_number].balance) {
59          printf("LOW_BALANCE_TRANSACTION_FAILED\n");
60          return LOW_BALANCE;
61      }
62      else {
63          serverData[array_number].balance = (serverData[array_number].balance) - (termData->transAmount);
64          printf("TRANSACTION_ACCEPTED\n");
65          return OKK;
66      }
67  }
```

saveTransaction function:

```
68
69  void saveTransaction(ST_accountsDB_t serverData[255]) {
70      FILE* WriteDB;
71      WriteDB = fopen("DataBase.txt", "w");
72      uint32_t loop = 0;
73      fprintf(WriteDB, "%s-%f-", serverData[loop].primaryAccountNumber, serverData[loop].balance);
74      for (loop = 1; loop < eofloop; loop++) {
75          fprintf(WriteDB, "\n%s-%f-", serverData[loop].primaryAccountNumber, serverData[loop].balance);
76      }
77      fclose(WriteDB);
78      printf("TRANSACTION_SAVED\n");
79  }
```

videos link:

https://drive.google.com/drive/folders/1eULdUeJsRnudqk2uZlun48x__3y_r5PR?usp=sharing