Payment.h

```cpp
#pragma once
class Payment
{
private:

        char username[30];
        char email[30];
        int contactNo;
        char accountNumber[20];
        double amount;
        int cardNo;
        char cardexpDate[20];
        int cardsecurityCode;


public:
        Payment();
        void setPaymentDetails(const char pusername[],const char pemail[],int
pcontactNo, const char paccountNumber[], double pamount, int pcardNo, const char
pexpDate[],
                                        int psecurityCode);
        void displayPayment();
        void validate(int scardNo, const char scardexpDate[], int scardsecurityCode);
        ~Payment();


};
```

Payment.cpp

```cpp
#include "Payment.h"
#include<iostream>
using namespace std;

Payment::Payment()
{
        strcpy(username,"");
        strcpy(email,"");
        contactNo=0;
        strcpy(accountNumber,"");
        amount=0;
        cardNo=0;
        strcpy(cardexpDate,"");
        cardsecurityCode=0;
}

void Payment::setPaymentDetails(const char pusername[], const char pemail[], int
pcontactNo, const char paccountNumber[], double pamount, int pcardNo, const char
pexpDate[],int psecurityCode)
{
        strcpy(username, pusername);
        strcpy(email, pemail);
        contactNo = pcontactNo;
```

```cpp
        strcpy(accountNumber,paccountNumber);
        amount = pamount;
        cardNo = pcardNo;
        strcpy(cardexpDate, pexpDate);
        cardsecurityCode = psecurityCode;
}


void Payment::displayPayment()
{
        cout << "Account number   : " << accountNumber << endl;
        cout << "Paid amount      : " << amount << endl;
        cout << "Email            : " << email << endl;

}

void Payment::validate(int scardNo, const char scardexpDate[], int
scardsecurityCode)
{
        cardNo = scardNo;
        strcpy(cardexpDate, scardexpDate);
        cardsecurityCode = scardsecurityCode;
}

Payment::~Payment()
{
        cout << "Destructor runs" << endl;
}
```

Receipt.h

```cpp
#pragma once
class Receipt
{
private:
        char accountNumber[20];
        char transactionID[20];
        char paymentDate[20];
        double amount;
public:
        void setReceiptDetails(const char saccountNumber[], const char
stransactionID[], const char spaymentDate[], double samount);
        void viewReceipt();
        string sendReceipt();
};
```

Receipt.cpp

```cpp
#include "Receipt.h"
#include<cstring>
#include<iostream>
using namespace std;

void Receipt::setReceiptDetails(const char saccountNumber[], const char
stransactionID[], const char spaymentDate[], double samount)
{
    strcpy(accountNumber, saccountNumber);
    strcpy(transactionID, stransactionID);
    strcpy(paymentDate, spaymentDate);
    amount = samount;
}

void Receipt::viewReceipt()
{
    cout << "Account number   : " << accountNumber << endl;
    cout << "Transaction ID        : " << transactionID << endl;
    cout << "Payment Date      : " << paymentDate << endl;
    cout << "Paid amount      : " << amount << endl;
    cout << "----------------------------------------" << endl;
    cout << "..Thank You.." << endl;
}
```

Contact.h

```cpp
#pragma once
class Contact
{
private:
    char customerID[20];
    char email[30];
public:
    Contact();
    void setContactDetails(const char pcustomerID[], const char pemail[]);
    ~Contact();

};
```

Contact.cpp

```cpp
#include "Contact.h"
#include<cstring>
#include<iostream>
using namespace std;

Contact::Contact()
{
        strcpy(customerID, "");
        strcpy(email, "");
}

void Contact::setContactDetails(const char pcustomerID[], const char pemail[])
{
        strcpy(customerID, pcustomerID);
        strcpy(email, pemail);
}

Contact::~Contact()
{
        cout << "Destructor runs" << endl;
}
```