



Topic : Online Auction System

Group no : MLB_06.01_07

Campus : Malabe

Submission Date : 20 May 2022

We declare that this is our own work and this Assignment does not incorporate without acknowledgment any material previously submitted by anyone else in SLIIT or any other university/Institute. And we declare that each one of us equally contributed to the completion of this Assignment.

Registration No	Name	Contact Number
IT21318320	Silva T.U.D	076 900 5515
IT21321054	Bandara D.M.E.K	076 944 7415
IT21319242	Ilangerathne R.A.C.W	071 543 1391
IT21318948	Dissanayake K.B	076 072 4229
IT21313684	Karunarathna D.T.S	071 161 1308

Description of the Requirements

- Users could search for the website using the URL and browse the services provided by the system.
- Users could contact support to resolve the issues related to services provided by the system.
- There are two types of users, unregistered users, and registered users. Guest users can search for products by product names and using available filters, and then users can view the product details and review them. Users should register to the system to access functionalities, including creating auctions and placing bids for products.
- Guest users should register to the system by providing required details, such as name, address, and phone number. After completing the registration, they can create the username and passwords for their profiles.
- Registered users could log in to the system by providing user credentials. Further registered users could reset passwords and edit their profiles.
- Registered users could add items for auctions by selecting the relevant product category and providing product details. The auctions could update and delete by the user based on the system restrictions.
- Registered users could place bids after reviewing the selected item, and users could update and delete the placed bid according to the system criteria.
- Auctioneers and bidders can communicate by sending messages to solve product-related issues.
- At the end of an auction, the winner will notify by a message. Then the registered users could make payments using bank cards, validate payments and receive the invoice.
- The seller will ship the item to the buyer with the help of the delivery system.
- Registered users could send feedback related to the service performances, and system admins could review the feedback and respond when needed.
- In addition, the system admin could review the auctions, bids and profiles and manage them by requesting to update and deleting them.

Classes

- Unregistered user
- Registered user
- Item
- Auction
- Category
- Bid
- Message
- Issue
- Payment
- Invoice
- Feedback
- Admin
- Customer Support

CRC Cards

- Message – IT21318320 – Silva T.U.D

Class name: Message	
Responsibilities	Collaborations
Send Message	Registered User
Save Message	
Display Message	

- Issue – IT21318320 – Silva T.U.D

Class name: Issue	
Responsibilities	Collaborations
Send Issue	Message
Save Issue Type	
Save Issue	Message
Display Issue	

- Feedback – IT21318320 – Silva T.U.D

Class name: Feedback	
Responsibilities	Collaborations
Send Feedback	Message
Save Feedback	Message
Save Rating	
Display Rating	

- Payment – IT21318320 – Silva T.U.D

Class name: Payment	
Responsibilities	Collaborations
Save Details	Item
Display Details	
Validate Details	
Authorize Payment	
Display Invoice	Invoice

- Invoice – IT21318320 – Silva T.U.D

Class name: Invoice	
Responsibilities	Collaborations
Save Invoice Details	Payment
Display Details	
Print Invoice	

- Admin – IT21321054 - Bandara D.M.E.K

Class name: Admin	
Responsibilities	Collaborations
View Feedback	Feedback
Manage Bid	Bid
Manage Auction	Auction
Manage Profile	Profile

- Customer Support – IT21321054 - Bandara D.M.E.K

Class name: Customer Support	
Responsibilities	Collaborations
Solve problems of users	Registered user

- Item – IT21319242 - Ilangarathne R.A.C.W

Class name: Item	
Responsibilities	Collaborations
Store details of item	
Search Item	
Select Item	
Add Item	Auction
Edit Item	

- Category – IT21319242 - Ilangarathne R.A.C.W

Class name: Category	
Responsibilities	Collaborations
Store details of category	
Update Category	
Delete Category	

- Registered user – IT21318948 - Dissanayake K.B

Class name: Registered user	
Responsibilities	Collaborations
Update their profile and reset password	
Add and view Item	Item
Update and delete auction	Auction
Make payment	Payment
Send Feedback	Feedback

- Auction – IT21318948 - Dissanayake K.B

Class name: Auction	
Responsibilities	Collaborations
Store auction details	
Update Auction	
Delete Auction	

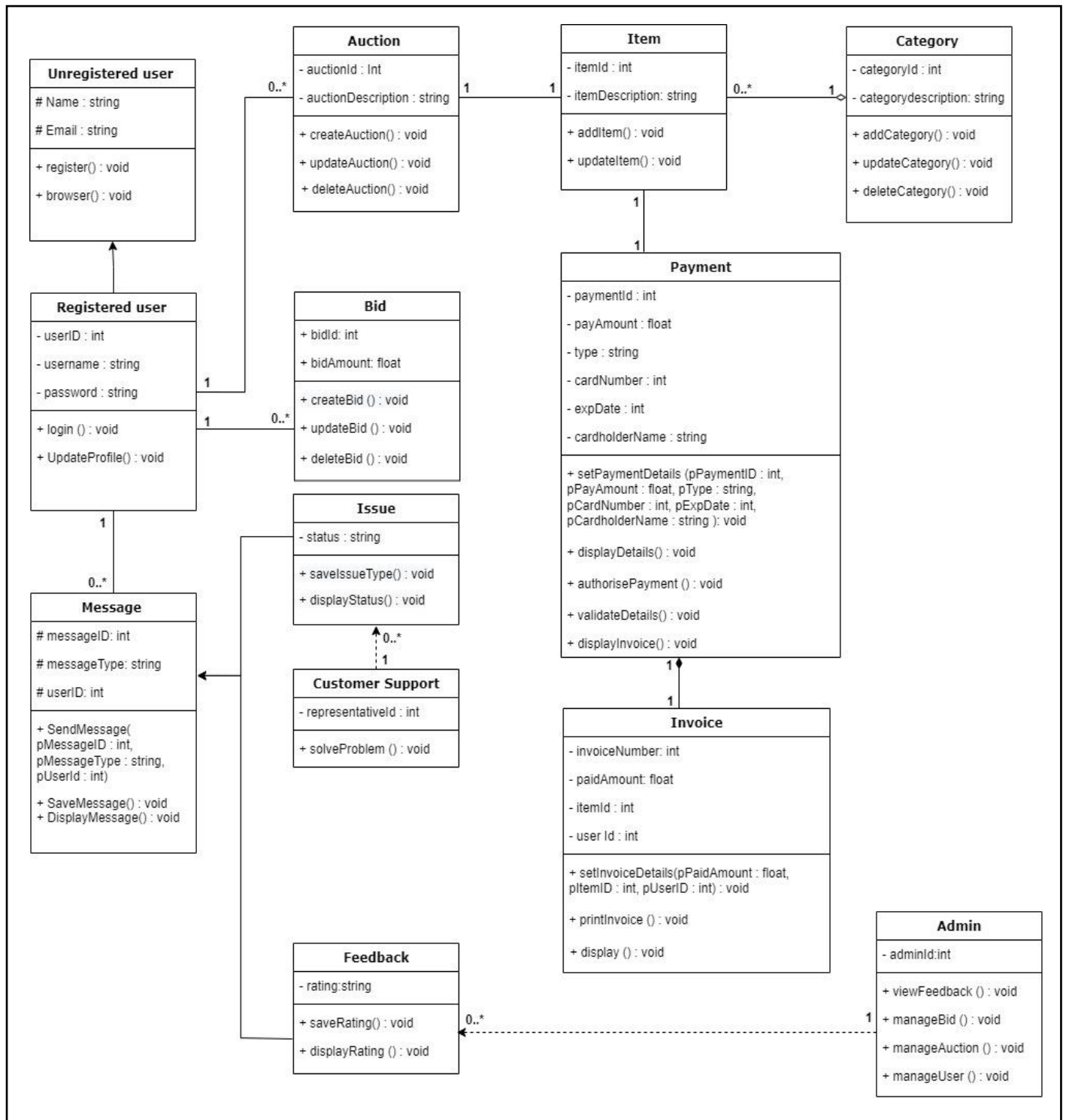
- Unregistered user – IT21313684 - Karunarathna D.T.S

Class name: Unregistered user	
Responsibilities	Collaborations
Provide the required details	
Create a username and password	

- Bid – IT21313684 - Karunarathna D.T.S

Class name: Bid	
Responsibilities	Collaborations
Place bid	Registered user
Update and delete the placed bid	Registered user
Review the bids	Admin

Exercise 1



Exercise 2

- **Message – IT21318320 – Silva T.U.D**

//Class definition of the message class.

```
class Message
```

```
{
```

```
protected:
```

```
    int messageID;
```

```
    char messageType[50];
```

```
    Registered_user* Ruser;
```

```
public:
```

```
    Message();
```

```
    Message(int pmessageID, char pmessageType[], Registered_user *pRuser);
```

```
    void saveMessage();
```

```
    void diplayMessage();
```

```
    ~Message();
```

```
};
```

// Implementation of the methods of the class message.

```
Message::Message(){
```

```
    messageID = 0;
```

```
    strcpy(messageType, "");
```

```
}
```

```
Message::Message(int pmessageID, char pmessageType[], Registered_user *pRuser){
```

```

messageID = pmessageID;
strcpy(messageType, pmessageType);
Ruser = pRuser;
}

```

```

void Message::saveMessage() {}

```

```

void Message::diplayMessage() {}

```

```

Message::~Message() {}

```

- **Issue – IT21318320 – Silva T.U.D**

//Class definition of the issue class.

```

class Issue :public Message
{
private:
    char status[10];

public:
    Issue();
    Issue(int cMessageID, char cMessageType[], char pstatus[]);
    void saveIssueType();
    void displayStatus();
    ~Issue();
};

```

```
// Implementation of the methods of the class issue.
```

```
Issue::Issue() {
```

```
    messageID = 0;
```

```
    strcpy(messageType, "");
```

```
    strcpy(status, "");
```

```
}
```

```
Issue::Issue(int cMessageID, char cMessageType[], char pstatus[]) {
```

```
    messageID = cMessageID;
```

```
    strcpy(messageType, cMessageType);
```

```
    strcpy(status, pstatus);
```

```
}
```

```
void Issue::saveIssueType() {}
```

```
void Issue::displayStatus() {}
```

```
Issue::~~Issue() {}
```

- **Feedback – IT21318320 – Silva T.U.D**

```
//Class definition of the feedback class.
```

```
class Feedback :public Message
```

```
{
```

```
    private:
```

```
        char rating[20];
```

```

public:
    Feedback();
    Feedback(int cMessageID, char cMessageType[], char prating[]);
    void saveRating();
    void displayRating();
    ~Feedback();
};

// Implementation of the methods of the class feedback.
Feedback::Feedback() {
    messageID = 0;
    strcpy(messageType, "");
    strcpy(rating, "");
}

Feedback::Feedback(int cMessageID, char cMessageType[], char prating[]) {
    messageID = cMessageID;
    strcpy(messageType, cMessageType);
    strcpy(rating, prating);
}

void Feedback::saveRating() {}

void Feedback::displayRating() {}

Feedback::~Feedback() {}

```

- **Payment – IT21318320 – Silva T.U.D**

//Class definition of the payment class.

```
class Payment {  
    private:  
        int paymentId;  
        float payamount;  
        char type[20];  
        int cardnumber;  
        int expdate;  
        char cardholdername[50];  
        Item* item;  
        Invoice* invoiceNo[100];  
  
    public:  
        Payment();  
        Payment(int no1);  
        void setpaymentDetails(int ppaymentId, float ppayamount, char ptype[], int pcardnumber,  
int pexdate, char pcardholdername[]);  
        void displayDetails();  
        void validateDetails();  
        void authorizePayment();  
        void displayInvoice();  
        ~Payment();  
};
```

// Implementation of the methods of the class payment.

```
Payment::Payment() {  
    invoiceNo[0] = new Invoice(001);
```

```
    invoiceNo[1] = new Invoice(002);  
}
```

```
Payment::Payment(int no1){  
    invoiceNo[2] = new Invoice(no1);  
}
```

```
void Payment::setpaymentDetails(int ppaymentId, float ppayamount, char ptype[], int  
pcardnumber, int pexdate, char pcardholdername[]) {  
    paymentId = ppaymentId;  
    payamount = ppayamount;  
    strcpy(type, ptype);  
    cardnumber = pcardnumber;  
    expdate = pexdate;  
    strcpy(cardholdername, pcardholdername);  
}
```

```
void Payment::displayDetails() {}
```

```
void Payment::validateDetails() {}
```

```
void Payment::authorizePayment(){}
```

```
void Payment::displayInvoice(){  
    for(int count=0; count < 100; count++)  
    {  
        invoiceNo[count] -> display();  
    }  
}
```

```

Payment::~~Payment(){
    for(int count=0; count < 100; count++)
    {
        delete invoiceNo[count];
    }
}

```

- **Invoice – IT21318320 – Silva T.U.D**

//Class definition of the invoice class.

```

class Invoice {
    private:
        int invoiceNumber;
        float paidamount;
        int itemID;
        int userId;

    public:
        Invoice();
        Invoice(int no);
        void setInvoiceDetails(float ppaidamount, int pitemID, int puserId);
        void printInvoice();
        void display();
        ~Invoice();
};

```

// Implementation of the methods of the class invoice.

```

Invoice::Invoice(){

```



```
invoiceNumber = 0;
paidamount = 0;
itemID = 0;
userId = 0;
}
```

```
Invoice::Invoice(int no) {
    invoiceNumber = no;
}
```

```
void Invoice::setInvoiceDetails(float ppaidamount, int pitemID, int puserId){
    paidamount = ppaidamount;
    itemID = pitemID;
    userId = puserId;
}
```

```
void Invoice::printInvoice() {}
```

```
void Invoice::display() {}
```

```
Invoice::~~Invoice() {}
```

- **Admin – IT21321054 - Bandara D.M.E.K**

```
//Admin class
class Admin
{
private:
    int adminID;
public:
    Admin();
    Admin(int aId);
    void newFeedback(Feedback*F);
    void manageBid();
    void manageauction();
    void manageusers();
    ~Admin();
};

Admin::Admin() {

}

Admin::Admin(int aId) {
    adminID = aId;
}

void Admin::newFeedback(Feedback* F) {

}

void Admin::manageBid() {

}
```

```

void Admin::manageauction() {

}

void Admin::manageusers() {

}

Admin::~Admin() {

}

```

- **Customer Support – IT21321054 - Bandara D.M.E.K**

```

//Customer Support class
class Customer_Support
{
private:
    int representativeId;
public:
    Customer_Support();
    Customer_Support(int repID);
    void solveproblem(Issue*Is);
    ~Customer_Support();
};

Customer_Support::Customer_Support() {

}

```

```

Customer_Support::Customer_Support(int repID) {

    representativeId = repID;
}

void Customer_Support::solveproblem(Issue* Is) {

}

Customer_Support::~~Customer_Support() {

}

```

- **Item – IT21319242 - Ilangarathne R.A.C.W**

```

//Item class
class Item
{
private:
    int itemId;
    string itemDescription;
    Auction* Au;
public:
    Item();
    Item(int IId,string iDescription);
    void additem();
    void updateItem();
};

```

```

Item::Item() {

}

Item::Item(int IId,string iDescription,)
{

    itemId = IId;
    itemDescription = iDescription;
}

void Item::additem() {
    Au = new Auction;
}

void Item::updateItem() {
}

```

- **Category – IT21319242 - Ilangarathne R.A.C.W**

```

//Category class
class Category
{
private:
    int categoryId;
    string categorydescription;
    Item* item[2];
public:
    Category();
    Category(int caId, string cdescription,Item*it1, Item*it2);

```

```

void addcategory();
void updatecategory();
void delectcategory();
};

Category::Category() {

}

Category::Category(int caId, string cdescription, Item* it1, Item* it2) {
    categoryId = caId;
    categorydescription = cdescription;
    item[0] = it1;
    item[1] = it2;
}

void Category::addcategory() {

}

void Category::updatecategory(){

}

void Category::delectcategory() {

}

```

- **Registered user – IT21318948 - Dissanayake K.B**

// Registered users class

class Registered_users :public Unregistered_users

{

private :

int userid;

char username;

char password;

Bid* bid;

Message* msg;

public:

Registered_users();

Registered_users(int uid,char uname,char upassword, Bid* B, Message* M);

void login();

void updateProfile();

~Registered_users();

};

Registered_users::Registered_users() {

}

Registered_users::Registered_users(int uid, char uname, char upassword, Bid* B, Message* M)

{

userid = uid;

username = uname;

password = upassword;

bid = B;

```

    msg = M;
}
void Registered_users::login() {

}
void Registered_users::updateProfile() {

}
Registered_users::~Registered_users() {

}

```

- **Auction – IT21318948 - Dissanayake K.B**

```

//Auction class
class Auction
{
private:
    int auctionID;
    string description;
    int quantity;
    Registered_users* Ruser;
public:
    Auction();
    Auction(int aId,char adescription,int aquantity, Registered_users* Rsu);
    void creatauction();
    void updateauction();

```



```

    void deleteauction();
    ~Auction();
};

Auction::Auction() {

}

Auction::Auction(int aId, char adescription, int aquantity, Registered_users* Rsu) {

    auctionID = aId;
    description = adescription;
    quantity = aquantity;
    Ruser = Rsu;
}

void Auction::creatauction() {

}

void Auction::updateauction() {

}

void Auction::deleteauction() {

}

Auction::~~Auction() {

}

```

- **Unregistered user – IT21313684 - Karunarathna D.T.S**

```
// Unregistered user class
#include <iostream>
using namespace std;

class Unregistered_user
{
protected:
    char name[];
    char email[] ;
public:
    Unregistered_user();
    Unregistered_user(char unname[], char unemail[]);
    void singup();
    void browse();
    ~Unregistered_user();
};

Unregistered_user::Unregistered_user() {

}

Unregistered_user::Unregistered_user(char unname[], char unemail[])
{
    char name[] = char unname[];
    char email[] = char unemail[];
}

void Unregistered_user::singup() {
```

```

}
void Unregistered_user::browse() {

}
Unregistered_user::~Unregistered_user() {

}

```

- **Bid – IT21313684 - Karunarathna D.T.S**

```

//Bid class
class Bid
{
private:
    int bidId;
    float bidAmount;
    Registered_user* Ruser;
public:
    Bid();
    Bid(int bid,float bAmount,Registered_user*Rsu);
    void creatBid();
    void updateBid();
    void deleteBid();
    ~Bid();
};

```

```
Bid::Bid() {  
  
}  
Bid::Bid(int bid, float bAmount, Registered_users* Rsu)  
{  
    bidId = bid;  
    bidAmount = bAmount;  
    Ruser = Rsu;  
}  
void Bid::creatBid() {  
  
}  
void Bid::updateBid() {  
  
}  
void Bid::deleteBid() {  
  
}  
Bid::~~Bid() {  
  
}
```

- **Main Program**

```
#include <iostream>
using namespace std;

int main(){
    // Create object of class Unregistered_user.
    Unregistered_users unu1;

    // Create object of class Registered_users.
    Registered_users regUser1;

    // Create object of class Admin.
    Admin admin1;

    // Create object of class Custmor_Support.
    Custmor_Support support1;

    // Create object of class Category.
    Category category1;

    // Create object of class Auction.
    Auction auction1;

    // Create object of class Item.
    Item item1;

    // Create object of class Bid.
    Bid bid1;

    // Create object of class Message.
    Message msg1;
```

```
// Create object of class Issue.  
Issue issue1;  
  
// Create object of class Feedback.  
Feedback feedback1;  
  
// Create object of class Invoice inside the the class Payment.  
Payment *payment1;  
  
payment1 = new Payment(1);  
payment1 -> displayInvoice();  
  
return 0;  
}
```

Individual Contribution		
IT21318320	Silva T.U.D	Design and implement the, <ul style="list-style-type: none"> • Class Message • Class Issue • Class Feedback • Class Payment • Class Invoice
IT21321054	Bandara D.M.E.K	Design and implement the, <ul style="list-style-type: none"> • Class Admin • Class Customer Support
IT21319242	Ilangarathne R.A.C.W	Design and implement the, <ul style="list-style-type: none"> • Class Item • Class Category
IT21318948	Dissanayake K.B	Design and implement the, <ul style="list-style-type: none"> • Class Registered User • Class Auction
IT21313684	Karunarathna D.T.S	Design and implement the, <ul style="list-style-type: none"> • Class Unregistered User • Class Bid