**Topic** : Online Land Sales System

**Group no.** : MET_WE_03_01

**Campus** : ~~Malabe /~~ **Metro** ~~/ Matara / Kandy / Kurunegala / Kandy / Jaffna~~

**Submission Date** : 20-05-2022

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 |
|---|---|---|
| IT21194030 | Evinda Hettiarachchi (Group Leader) | 077 700 7244 |
| IT21211232 | Binoosh Warakapola | 071 265 2553 |
| IT21375682 | Jayapalan Shandeep | 075 708 2659 |
| IT21375514 | Chanthirasegaran Sanjayan | 076 162 2337 |
| IT21298394 | Ashan Perera | 070 216 2498 |

# Table of Contents

## List of Figures

# 1. Identified Requirements

1. System should be available to Users 24 / 7 / 365
2. Guest users can overview the basic functionalities of the system.
3. Guest users can search the advertisements by using keywords
4. Guest users can register with the System by providing details such as their Name, NIC, Address, Contact No., and email.
5. Registered users can add advertisements to favourites
6. Registered users can pay for publishing advertisements through the website
7. Payment slip will be issued to the user
8. Users can filter the advertisements by area, price, district, province, and country.
9. People can compare the properties of a selected number of advertisements
10. Quality control should be able to view the advertisement which created by Seller and the payment status.
11. Quality control should be able to contact the seller via email or call if there are any issues regarding the advertisement
12. Quality control should be able to approve the advertisement
13. After approving, Quality control should be able to send confirmation to the Seller via email or message
14. Able to send feedback forms to the Seller
15. Quality control should be able to login using Quality control credentials like ID, password, email, verification code.
16. Quality control Should be able to check User requests and feedbacks.
17. After working on requests, Admin should be able to reply to the User via email.
18. Quality control should be able to view the dashboard.
19. System should indicate if there is any possible security breach.
20. A support service provider can login (using username and password), view and respond to the list of issues sent by customers.
21. A support service provider can send unsolved issues and receive solutions from an administrator.
22. After responding through customer support, the system requests and save feedback and ratings from customers.

# 2. Noun and Verb Analysis

## 2.1 Nouns

1. System should be available to Users 24 / 7 / 365
2. Guest users can overview the basic functionalities of the system.
3. Guest users can search the advertisements by using keywords
4. Guest users can register with the System by providing details such as their Name, NIC, Address, Contact No., and email.
5. Registered users can add advertisements to favourites
6. Registered users can pay for publishing advertisements through the website
7. Payment slip will be issued to the user
8. Users can filter the advertisements by area, price, district, province, and country.
9. People can compare the properties of a selected number of advertisements
10. Quality control should be able to view the advertisement which created by Seller and the payment status.
11. Quality control should be able to contact the seller via email or call if there are any issues regarding the advertisement
12. Quality control should be able to approve the advertisement
13. After approving, Quality control should be able to send confirmation to the Seller via email or message
14. Able to view feedback forms to the Seller
15. Quality control should be able to login using Quality control credentials like ID, password, email, verification code.
16. Quality control Should be able to check User requests and feedbacks.
17. After working on requests, Admin should be able to reply to the User via email.
18. Quality control should be able to view the dashboard.
19. System should indicate if there is any possible security breach.
20. A support service provider can login (using username and password), view and respond to the list of issues sent by customers.
21. A support service provider can send unsolved issues and receive solutions from an administrator.
22. After responding through customer support, the system requests and save feedback and ratings from customers.

## 2.2 Verbs

1. System should be available to Users 24 / 7 / 365
2. Guest users can overview the basic functionalities of the system.
3. Guest users can search the advertisements by using keywords.
4. Guest users can register with the system by providing details such as their Name, NIC, Address, Contact No. and email.
5. Registered users can add advertisements to favourites.
6. Registered users can pay for publishing advertisements through the website
7. Payment slip will be issued to the user
8. Users can filter the advertisements by area, price, district, province, and country.
9. People can compare the properties of a selected number of advertisements.
10. Quality control should be able to view the advertisement which created by Seller and the payment status.
11. Quality control should be able to contact the Seller via email or call if there are any issues regarding the advertisement.
12. Quality control should be able to approve the advertisement.
13. After approving, Quality control should be able to send confirmation to the Seller via email or message.
14. Able to view feedback forms to the Seller.
15. Quality control should be able to login using Quality control credentials like ID, password, email, verification code.
16. Quality control should be able to check User requests and feedbacks.
17. After working on requests, Admin should be able to reply to the User via email.
18. Quality control should be able to view the dashboard.
19. System should indicate if there is any possible security breach.
20. A support service provider can login (using username and password), view and respond to the list of issues sent by customers.
21. A support service provider can send unsolved issues and receive solutions from an administrator.
22. After responding through customer support, the system requests and save feedback and ratings from customers.

# 3. Identified Classes

1. Guest User
2. Registered User
3. Seller
4. Payment
5. Quality Control
6. Land
7. Advertisement
8. Feedback

## 3.1 Reasons for rejecting other nouns

**Redundant:** Users, People, support services, customer service, customers

**An event or an operation:** send confirmation

**Outside scope of the system:** System

**Meta-language:**

**Attributes**: properties, payment status, emails, calls, message, ID, password, verification code

## 3.2 Methods

| | |
|---|---|
| 1. Guest User | - Provide details and register into the system |
| | - View advertisements |
| 2. Registered user | - Login to our system |
| | - View advertisements |
| | - Filtering the results to get preferred advertisements |
| | - Add advertisements to favorites |
| | - Compare properties |
| 3. Seller | - Publish advertisements |
| | - Pay for publishing |
| 4. Quality control | - View advertisements |
| | - Approve advertisements |
| | - Contact seller |
| 5. Payment | - Generate an ID |
| | - Make payments |
| | - Verify the details entered by publisher |
| | - Confirm the details |
| 6. Land | - Add details of Land |
| | - Update the existing details |
| | - Delete the existing details |
| | - Confirm Land Details |
| 7. Advertisement | - Generate an advertisement ID |
| | - Get details of land |
| | - Create advertisement |
| 8. Feedback | - Generate a feedback ID |
| | - Request for feedback |
| | - Forward to quality control |

# 4. CRC Cards

## 4.1 Guest User

| **Class Name: Guest User** | |
|---|---|
| **Responsibility** | **Collaborators** |
| Provide details and register into the system | |
| View advertisements | Advertisement |
| | |
| | |

## 4.2 Registered user

| **Class Name: Registered user** | |
|---|---|
| **Responsibility** | **Collaborators** |
| Login to our system | |
| View advertisements | Advertisement |
| Filtering the results to get preferred advertisements | Advertisement |
| Add advertisements to favourites | Advertisement |
| Compare properties | |

## 4.3 Seller

| **Class Name: Seller** | |
|---|---|
| **Responsibility** | **Collaborators** |
| Publish advertisements | Advertisement |
| Pay for publishing | Payment |
| | |

## 4.4 Quality Control

| **Class Name: Quality control** | |
|---|---|
| **Responsibility** | **Collaborators** |
| View advertisements | Advertisement |
| Approve advertisements | Advertisement |
| Contact seller | Seller |
| View feedback | Feedback |

## 4.5 Payment

| **Class Name: Payment** | |
|---|---|
| **Responsibility** | **Collaborators** |
| Generate an ID | |
| Make payments | Seller |
| Verify the details entered by publisher | Quality Control |
| Confirm the details | |

## 4.6 Advertisement

| **Class Name: Advertisement** | |
|---|---|
| **Responsibility** | **Collaborators** |
| Generate an advertisement ID | |
| Get details of land | Seller |
| Create advertisement | Seller |

## 4.7 Feedback

| **Class Name: Feedback** | |
|---|---|
| **Responsibility** | **Collaborators** |
| Generate a feedback ID | |
| Request for feedback | |
| Forward to quality control | Quality Control |
| | |

## 4.8 Land

| **Class Name: Land** | |
|---|---|
| **Responsibility** | **Collaborators** |
| Add Land Details | Seller |
| Update Land Details | Seller |
| Delete Land Details | Quality Control, Seller |
| Verify Land Details | Quality Control |

# 5. Exercise 1 - Class Diagram (UML Notation)

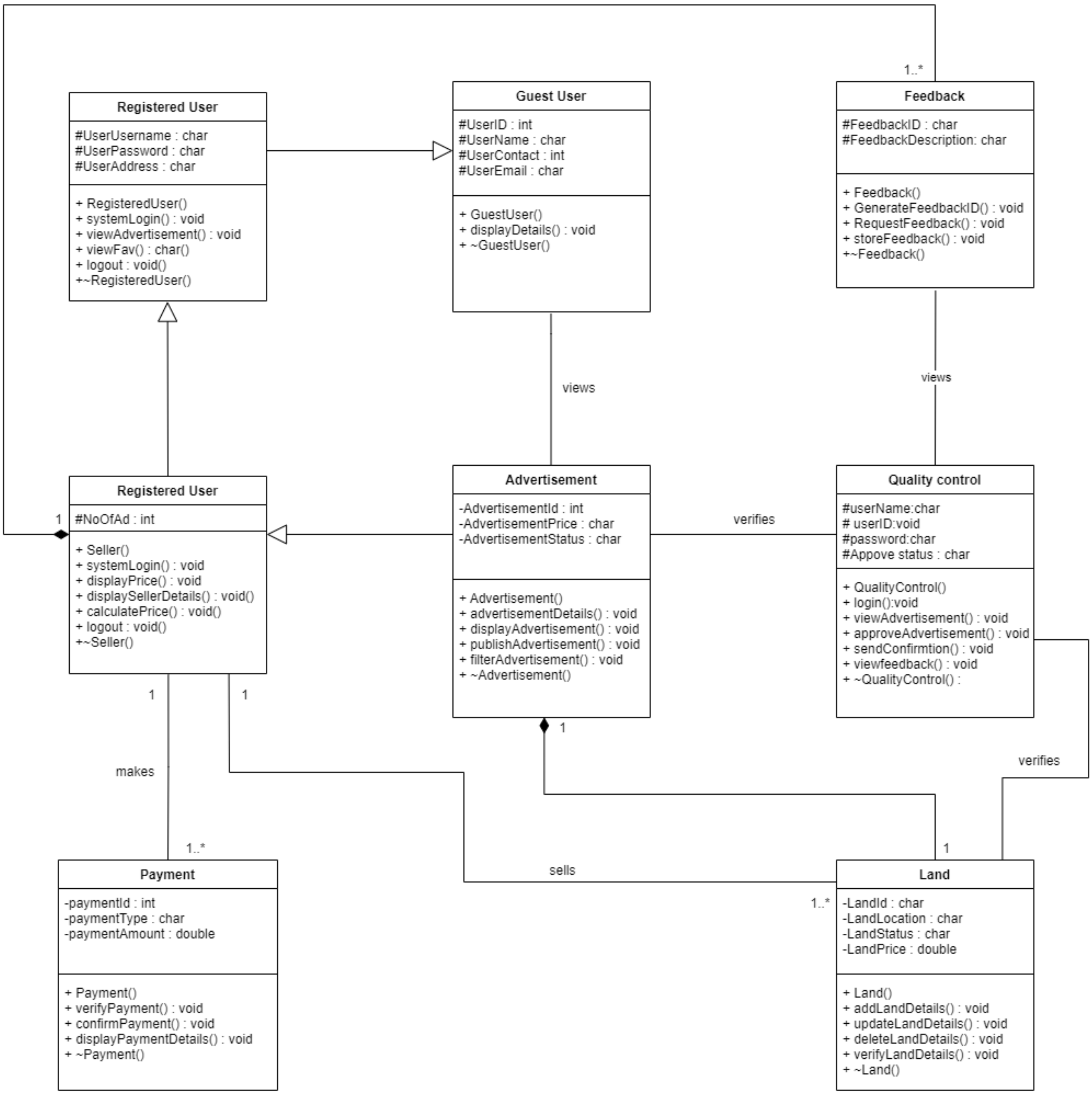

*Figure 1 – Class Diagram (UML Notation)*

# 6. Exercise 2 – Coding

## 6.1 Header Files

### 6.1.1 Guest User (IT21211232 – Binoosh Warakapola)

```
class GuestUser
{
protected:
	char UserID[10];
	char UserName[20];
	char UserContact[10];
	char UserEmail[20];
public:
	GuestUser();
	GuestUser(char inUserId[], char inUserName[], char inUserContact[],
char inUserEmail[]);
	void setUserID(char inUserId[]);//setter for user ID
	char getUserId(); //getter for user ID
	void setUserName(char inUserName[]);//setter for username
	char getUserName();//getter for username
	void setUserConatct(char inUserContact);//setter for user contact
	char getUserContact();//getter for user contact
	void setUserEmail(char inUserEmail);//setter for user email
	char getUserEmail();//getter for user email
	void UserRegister();

	~GuestUser();
};
```

### 6.1.2 Registered User (IT21211232 – Binoosh Warakapola)

```
class RegisteredUser :public GuestUser
{
protected:
	char username[20];
	char password[20];
public:
	RegisteredUser();
	RegisteredUser(char username[20], char password[20],
		int inUserId, char inUserName[],
			char inUserContact[], char inUserEmail[]);
		void setUserID(char inUserId[]);//setter for user ID
		char getUserId(); //getter for user ID
		void setUserName(char inUserName[]);//setter for username
		char getUserName();//getter for username
		void setUserConatct(char inUserContact);//setter for user contact
		char getUserContact();//getter for user contact
		void setUserEmail(char inUserEmail);//setter for user email
		char getUserEmail();//getter for user email

		void viewAdvertisements(int aid[]);
		void viewFav(int aid[]);
		void Systemlogin();
		void Systemlogout();

		~RegisteredUser();

};
```

### 6.1.3 Seller (IT21375514 – Chanthirasegaran Sanjayan)

```
#include "RegisteredUser"

class Seller :public RegisteredUser

{

	private:

		int noOfAd;

	public;

	   Seller();

	   Seller(const char pusername[],const char ppassword[],

		int inUserId,const char inUserName[],const char inUserContact[],

		const char inUserEmail[],int pnoOfAd);

	   void displayPrice();

	   void displaySellerDetails();
```

```
        void calculatePrice(int PnoOfAd,float price);
            void systemLogin();
        void systemLogout();
        ~Seller();
};
```

### 6.1.4 Quality Control (IT21375682 – Jayapalan Shandeep)

```
#include "Advertisement.h"
#include "Feedback.h"
#include "seller.h"


class QualityControl
{
private:

    char userName[20];
    char userID[10];
    char password[10];
    bool Approvementstatus;
private:
    QualityControl();
    QualityControl(const char IuserName[]);
    void login(const char IuserID[], const char Ipassword[]);
    void viewAdvertisement(Advertisement*view);
    void ApproveAdvertisement();
    void sendConfirmation(seller*seller1);
    void veiwFeedback(Feedback*Feed1);
    ~QualityControl();
};
```

### 6.1.5 Feedback (IT21375682 – Jayapalan Shandeep)

```
#include "seller.h"

class Feedback
{
private:
        char FeedbackID[10];
        char FeedbackDescription[100];

public:
        void Feedback();//default constructor
        void GenerateFeedbackID();
        void RequestFeedback(seller*seller1);
        void store Feedback();
        ~Feedback();


};
```

### 6.1.6 Payment (IT21298394 – Ashan Perera)

```
#include <iostream>

using namespace std;


class Payment{

private :

int PaymentId;

char PaymentType [10];

double PaymentAmount;
public :

Payment(); //default constructor

Payment(int pID, char pType [], double pAmo); //overloaded constructor

void verifyPayment();

void confirmPayment();

void displayPaymentDetails();

~Payment(); //destructor

};
```

### 6.1.7 Advertisement (IT21194030 – Evinda Hettiarachchi)

```
//IT21194030-Evinda Hettiarachchi-MET_WE_03_01

#include <iostream>

using namespace std;


class Advertisement{

private :

int AdvertisementId;

char AdvertisementPrice [20];

char AdvertisementStatus [50];




public :

Advertisement(); //default constructor
Advertisement (int aID, char aPr [], char aStat []); //overloaded constructor

void advertisementDetails();

void displayAdvertisement();

void publishAdvertisement();

void filterAdvertisement();

~Advertisement(); //destructor

};
```

### 6.1.8 Land (IT21194030 – Evinda Hettiarachchi)

```
//IT21194030-Evinda Hettiarachchi-MET_WE_03_01


#include <iostream>

using namespace std;


class Land{

private :

int LandId;

char LandLocation [100];

char LandStatus [20];

double LandPrice;




public :

Land(); //default constructor

Land(int lID, char lLoc [], char lStat [], double lPr); //overloaded constructor
void addLandDetails();

void updateLandDetails();

void deleteLandDetails();

void verifyLandDetails();

~Land(); //destructor

};
```

## 6.2 .cpp Files

### 6.2.1 Guest User (IT21211232 – Binoosh Warakapola)

```
#include "GuestUser.h"
#include <cstring>

GuestUser::GuestUser()
{
	strcpy(UserID, "");
	strcpy(UserName, "");
	strcpy(UserContact, "");
	strcpy(UserEmail, "");
}
GuestUser::GuestUser(char inUserId[],
	char inUserName[], char inUserContact[],
	char inUserEmail[])
{
	strcpy(UserID, inUserId);
	strcpy(UserName, inUserName);
	strcpy(UserContact, inUserContact);
	strcpy(UserEmail, inUserEmail);
}
void GuestUser::setUserID(char inUserId[])
{

}
char GuestUser::getUserId()
{

}//getter for user ID
void GuestUser::setUserName(char inUserName[])
{

}//setter for username
char GuestUser::getUserName()
{

}//getter for username
void GuestUser::setUserConatct(char inUserContact)
{
}//setter for user contact
char GuestUser::getUserContact()
{

}//getter for user contact
void GuestUser::setUserEmail(char inUserEmail)
{

}//setter for user email
char GuestUser::getUserEmail()
{

}//getter for user email
void GuestUser::UserRegister()
{

}

GuestUser::~GuestUser()
{

}
```

### 6.2.2 Registered User (IT21211232 – Binoosh Warakapola)

```
#include "RegisteredUser.h"
#include<cstring>
RegisteredUser::RegisteredUser()
{
	strcpy(username, "");
	strcpy(password, "");
}
RegisteredUser::RegisteredUser(char username[20], char password[20],
	int inUserId, char inUserName[],
	char inUserContact[], char inUserEmail[])
{

}

void RegisteredUser::setUserID(char inUserId[])
{
}//setter for user ID
char RegisteredUser::getUserId()
{

}//getter for user ID
void RegisteredUser::setUserName(char inUserName[])
{

}//setter for username
char RegisteredUser::getUserName()
{

}//getter for username
void RegisteredUser::setUserConatct(char inUserContact)
{

}//setter for user contact
char RegisteredUser::getUserContact()
{

}//getter for user contact
void RegisteredUser::setUserEmail(char inUserEmail)
{

}//setter for user email
char RegisteredUser::getUserEmail()
{

}//getter for user email

void RegisteredUser::viewAdvertisements(int aid[])
{

}
void RegisteredUser::viewFav(int aid[])
{

}
void RegisteredUser::Systemlogin()
{
}
void RegisteredUser::Systemlogout()
{

}

RegisteredUser::~RegisteredUser()
{

}
```

### 6.2.3 Seller (IT21375514 – Chanthirasegaran Sanjayan)

```
#include "RegisteredUser"


class Seller :public RegisteredUser


{


        private:


                int noOfAd;




        public;


          Seller();
          Seller(const char pusername[],const char ppassword[],

                int inUserId,const char inUserName[],const char inUserContact[],

                const char inUserEmail[],int pnoOfAd);

          void displayPrice();

          void displaySellerDetails();

          void calculatePrice(int PnoOfAd,float price);

                void systemLogin();

          void systemLogout();

          ~Seller();

};
```

### 6.2.4 Quality Control (IT21375682 – Jayapalan Shandeep)

```
#include "QualityControl.h"
#include <cstring>



QualityControl::QualityControl() {
    strcpy(userName,"");
    strcpy(password,"");
    Approvementstatus = false;
}

QualityControl::QualityControl(const char IuserName[]) {
    strcpy(username, "IuserName");

}
QualityControl::login(const char IuserID[],const char
Ipassword[]) {

}

QualityControl::viewAdvertisement(Adverstisement*view){

}

QualityControl::ApproveAdvertisement() {

}

QualityControl::sendConfirmation(seller*seller1) {

}

QualityControl::veiwFeedback(Feedback*Feed1) {

}

QualityControl::~QualityControl() {


}
```

### 6.2.5 Feedback (IT21375682 – Jayapalan Shandeep)

```
#include "Feedback.h"
#include<cstring>

Feedback::Feedback() {
	strcpy(FeedbackID, "");
	strcpy(FeedbackDescription,"");
}

Feedback::GenerateFeedbackID() {

}

Feedback::RequestFeedback(seller*seller1) {

}

Feedback::storeFeedback() {

}

Feedback::~Feedback() {

}
```

### 6.2.6 Payment (IT21298394 – Ashan Perera)

```
#include "Payment.h"

#include <iostream>

#include <cstring>

using namespace std;




Payment::Payment(int    pID,    char    pType    [],    double
pAmo)//overloaded constructor

{
```

```
PaymentId = pID;

strcpy(PaymentType, pType);

PaymentAmount = pAmo;

}



void Payment::verifyPayment(){



}



void Payment::confirmPayment(){



}



void Payment::displayPaymentDetails(){



}



Payment::~Payment()//destructor

{

cout << "Destructor runs";
```

```
}
```

### 6.2.7 Advertisement (IT21194030 – Evinda Hettiarachchi)

```
//IT21194030-Evinda Hettiarachchi-MET_WE_03_01


#include "Advertisement.h"

#include <iostream>

#include <cstring>

using namespace std;


Advertisement::Advertisement(int aID, char aPr [], char aStat []) //overloaded constructor

{

AdvertisementId = aID;

strcpy(AdvertisementPrice, aPr);

strcpy(AdvertisementStatus, aStat);

}


void Advertisement::advertisementDetails(){



}


void Advertisement::displayAdvertisement(){



}
```

```
void Advertisement::publishAdvertisement(){


}


void Advertisement::filterAdvertisement(){


}


Advertisement::~Advertisement()//destructor

{

cout << "Destructor runs";

}
```

### 6.2.8 Land (IT21194030 – Evinda Hettiarachchi)

```
//IT21194030-Evinda Hettiarachchi-MET_WE_03_01


#include "Land.h"

#include <iostream>

#include <cstring>

using namespace std;


Land::Land(int lID, char lLoc [], char lStat [], double lPr)//overloaded constructor

{

LandId = lID;

strcpy(LandLocation, lLoc);
}

strcpy(LandStatus, lStat);

LandPrice = lPr;

}


void Land::addLandDetails(){


}


void Land::updateLandDetails(){


}


void Land::deleteLandDetails(){


}


void Land::verifyLandDetails(){


}


Land::~Land()//destructor

{

cout << "Destructor runs";

}
```

## 6.3 main.cpp

```
#include <iostream>

#include "GuestUser.h"

#include "Seller.h"



using namespace std;



int main(void) {

	//creating static objects



	GuestUser g1;

	RegisteredUser r1;

	Seller s1;



	return 0;



}
```