Topic : Online Educational Games

Group no : KDY_11

Campus : Kandy

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**
IT21344824	Ekanayake T. E. M. A. P.	0770864008
IT21233630	Ismail M.A.U.	0760081256
IT21339424	Wickramarathna W.G.T.A.	0740236985
IT21220524	Bandara P.H.W.M.N.S.	0752719455
IT21333934	Vimansika A.P.	0767081034

Exercise 01

1. Requirements

1. A guest can register to the system.
2. Guests can view games and news.
3. User can login to the system.
4. Users can select game to purchase.
5. Users can view games and news.
6. Users can add games to their lobby.
7. User can make payments.
8. System can verify users.
9. System admin can manage the website.
10. Payment gateway can receive payments.
11. Payment gateway can verify transactions.

2. Noun-verb analysis

1. A guest can register to the ~~system~~. (Out of scope/Whole system is not a class)
2. ~~Guests~~ (Redundant) can game and news.
3. User can login to the ~~system~~. (Out of scope/Whole system is not a class)
4. ~~Users~~ (Redundant) can select ~~games~~ (Redundant) to purchase.
5. ~~Users~~ (Redundant) can view ~~games~~ and ~~news.~~ (Redundant).
6. ~~Users~~ (Redundant) can add ~~games~~ to their lobby.
7. ~~User~~ (Redundant) can make payments.
8. ~~System~~ can verify ~~users~~ (Out of scope/ Redundant).
9. System admin can manage the website.
10. ~~Payment gateway~~ (Redundant) can receive ~~payments~~ (Redundant).
11. ~~Payment gateway~~ (Redundant) can verify transactions.

3. Classes:

- Guest
- Game
- News
- User
- Payment
- Admin
- Lobby

Exercise 2

Class Name: Admin	
Responsibility	**Collaborations**
Manage website	
Editing game	Game
Editing news	News

Class Name: Game	
Responsibility	**Collaborations**
Store details of Games	Admin
Show games details for admin	Admin
Show games details for user	User
Show games details for guest	Guest

Class Name: News	
Responsibility	**Collaborations**
Store details of news	Admin
Show news about Games for admin	Admin
Show news about Games for user	User
Show news about Games for guest	Guest

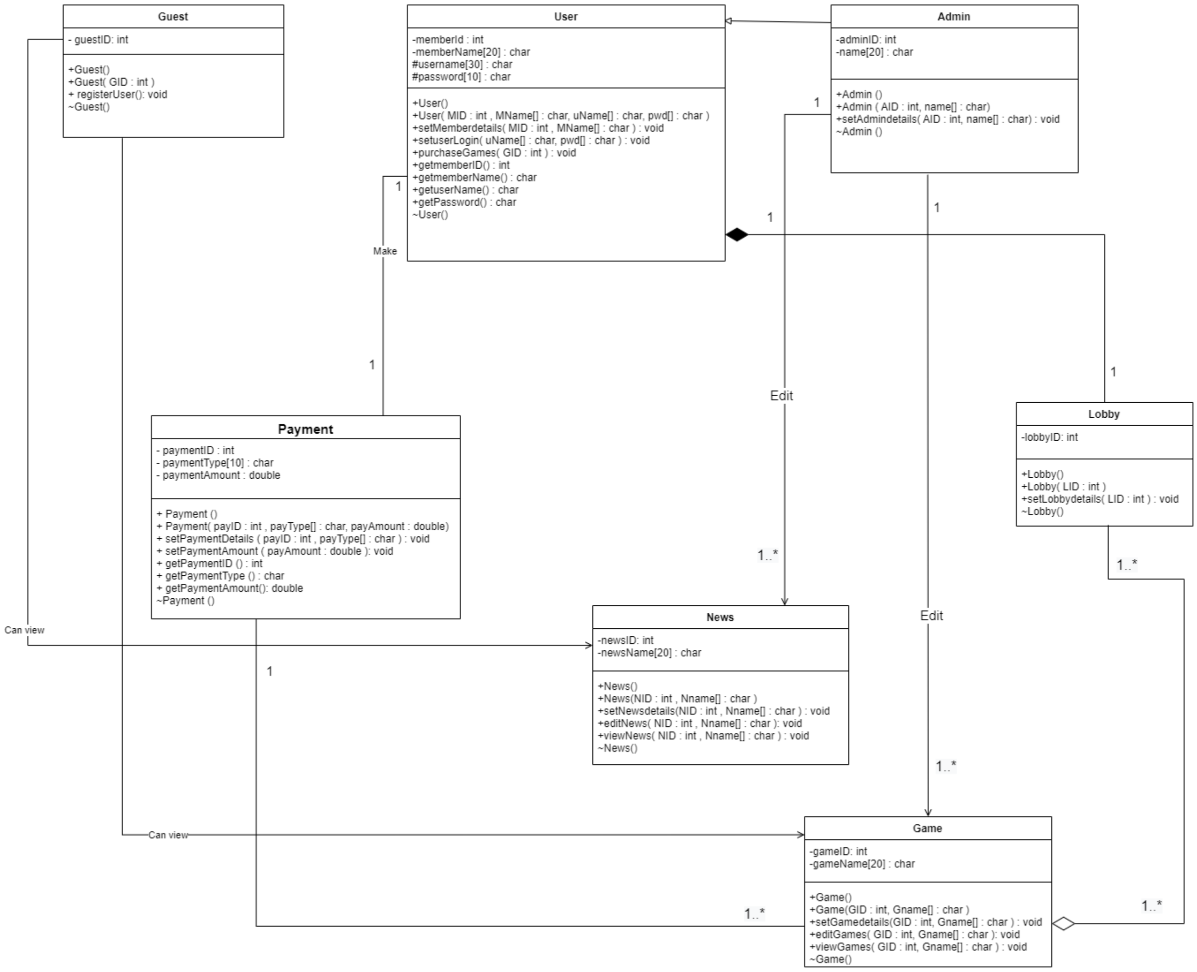
Class Name: Payment	
Responsibilities	**Collaborations**
Store details of payment information	
Provide credit card details	User
Purchasing game	Game

Class Name: Guest	
Responsibilities	**Collaborations**
Should register to the system to buy games	
Should be able to view Games	Game
Should be able to view News	News

Class Name: User	
Responsibilities	**Collaborations**
Manage user profile	
Can view games	Game
Can view news	News
Can purchase games	payment

Class Name: Lobby	
Responsibilities	**Collaborations**
Adding the purchased game to lobby	Payment

Exercise 03


Guest
- guestID: int
+Guest()
+Guest(GID : int)
+ registerUser(): void
~Guest()
User
-memberId : int
-memberName[20] : char
#username[30] : char
#password[10] : char
+User()
+User(MID : int , MName[] : char, uName[] : char, pwd[] : char)
+setMemberdetails(MID : int , MName[] : char) : void
+setuserLogin(uName[] : char, pwd[] : char) : void
+purchaseGames(GID : int) : void
+getmemberID() : int
+getmemberName() : char
+getuserName() : char
+getPassword() : char
~User()
Admin
-adminID: int
-name[20] : char
+Admin ()
+Admin (AID : int, name[] : char)
+setAdmindetails(AID : int, name[] : char) : void
~Admin ()
1
1
1
1
Make
1
Edit
1
Lobby
-lobbyID: int
+Lobby()
+Lobby(LID : int)
+setLobbydetails(LID : int) : void
~Lobby()
Payment
- paymentID : int
- paymentType[10] : char
- paymentAmount : double
+ Payment ()
+ Payment(payID : int , payType[] : char, payAmount : double)
+ setPaymentDetails (payID : int , payType[] : char) : void
+ setPaymentAmount (payAmount : double): void
+ getPaymentID () : int
+ getPaymentType () : char
+ getPaymentAmount(): double
~Payment ()
1..*
1..*
Can view
Edit
News
-newsID: int
-newsName[20] : char
+News()
+News(NID : int , Nname[] : char)
+setNewsdetails(NID : int , Nname[] : char) : void
+editNews(NID : int , Nname[] : char): void
+viewNews(NID : int , Nname[] : char) : void
~News()
1
1..*
Can view
Game
-gameID: int
-gameName[20] : char
+Game()
+Game(GID : int, Gname[] : char)
+setGamedetails(GID : int, Gname[] : char) : void
+editGames(GID : int, Gname[] : char): void
+viewGames(GID : int, Gname[] : char) : void
~Game()
1..*
1..*

Exercise 4

Main.cpp

```
//header file
#include <iostream>
#include "User.h"
#include "Guest.h"
#include "Admin.h"
#include "Game.h"
#include "News.h"
#include "Payment.h"
#include "Lobby.h"
using namespace std;

int main() {

  //creating object
  User U1;
  User U2( 01, (char *) "Samith", (char *) "samithperera", (char *) "samith@2022" );

  Admin A1;
  Admin A2( 03, (char *) "Akidu" );

  Guest Gu1;
  Guest Gu2( 04 );

  Game G1;
  Game G2( 10, (char *) "COD4" );

  News N1;
  News N2( 12, (char *) "Populer games in 2022") ;

  Payment P1;
  Payment P2( 01344, (char *) "Visa", 3000.00 );

  Lobby L1;
  Lobby L2( 1003 );

  return 0;
  //end main
}
```

User.h

```
//User class implementation

class User {

                    protected:

                        char username[30];

                        char password[10];

                    private:

                        int memberId;

                        char memberName[20];

                    public:

                        User();

                        User( int MID, char MName[], char uName[], char pwd[] );

                        void setMemberdetails( int MID, char MName[] );

                        void setuserLogin( char uName[], char pwd[] );

                        void purchaseGames( int GID );

                        int getmemberID();

                        char getmemberName();

                        char getuserName();

                        char getPassword();

                        ~User();

};
```

User.cpp

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

//defalt constructor
```

```
User::User()
{
        memberId = 0;
        strcpy( memberName , "" );
        strcpy( username , "" );
        strcpy( password , "" );
}
//overloaded constructor
User::User( int MID, char MName[], char uName[], char pwd[] )
{
        memberId = MID;
        strcpy( memberName , "MName" );
        strcpy( username , "uName" );
        strcpy( password , "pwd" );
}
//Destructor
User::~User()
{
        cout<< "User Destructor called" << endl;
}
```

Admin.h

```
//Admin class implementation
class Admin {
        private:
                int adminID;
                char name[20];
        public:
                Admin();
                Admin( int AID , char name[] );
                void setAdmindetails( int AID , char name[] );
                ~Admin();
};
```

Admin.cpp

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

//defalt constructor
Admin::Admin()
{
        adminID = 0;
        strcpy( name , "" );
}
//overloaded constructor
Admin::Admin( int AID , char name[] )
{
        adminID = AID;
        strcpy( name , "name");
}
//Destructor
Admin::~Admin()
{
        cout<<"Admin Destructor called" << endl;
}
```

Guest.h

```
//Guest class implementation
class Guest{
        private:
                int guestID ;

        public:
                Guest();
                Guest(int GID);
                ~Guest();
                void registerUser();
};
```

Guest.cpp

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

//defalt constructor
Guest::Guest()
{
        guestID = 0;
}
//overloaded constructor
Guest::Guest(int GID)
{
        guestID = GID;
}
//Destructor
Guest::~Guest()
{
        cout<<"Guest Destructor called" << endl;
}
```

Game.h

```
//Game class implementation
class Game{
    private:
        int gameID;
   char gameName[20];
 public:
    Game();
        Game( int GID , char Gname[] );
        void setGamedetails( int GID , char Gname[] );
        void editGames( int GID , char Gname[] );
        void viewGames( int GID , char Gname[] );
        ~Game();
};
```

Game.cpp

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

//defalt constructor
Game::Game()
{
    gameID = 0;
    strcpy( gameName , "" );
}
//overloaded constructor
```

```
Game::Game( int GID , char Gname[] )
{
        gameID = 0;
        strcpy( gameName , "Gname" );
}
//Destructor
Game::~Game()
{
        cout<<"Game Destructor called" << endl;
}
```

News.h

```
//News class implementation
class News{
        private:
                int newsID;
     char newsName[20];
  public:
        News();
                News( int NID , char Nname[] );
                void setNewsdetails( int NID , char Nname[] );
                void editNews( int NID , char Nname[] );
                void viewNews( int NID , char Nname[] );
                ~News();
};
```

News.cpp

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

//defalt constructor
News::News()
{
        newsID = 0;
        strcpy( newsName , "" );
}
//overloaded constructor
News::News( int NID , char Nname[] )
{
        newsID = 0;
        strcpy( newsName , "Nname" );
}
//Destructor
News::~News()
{
        cout<<"News Destructor called" << endl;
}
```

Payment.h

```
//Payment class implementation
class Payment{
	private:
		int paymentID;
		char paymentType[10];
		double paymentAmount;
	public:
		Payment();
		Payment( int payID , char payType[], double payAmount );
		void setPaymentDetails( int payID , char payType[] );
		void setPaymentAmount( double payAmount );
		int getPaymentID();
		char getPaymentType();
		double getPaymentAmount();
		~Payment();
};
```

Payment.cpp

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

//defalt constructor
Payment::Payment()
{
	paymentID = 0;
	strcpy( paymentType , "" );
	paymentAmount = 0;
}
//overloaded constructor
Payment::Payment( int payID , char payType[], double payAmount )
{
	paymentID = payID;
	strcpy( paymentType , "payType" );
	paymentAmount = payAmount;
}
//Destructor
Payment::~Payment()
```

```
{
	cout<<"Payment Destructor called" << endl;
}
```

Lobby.h

```
//Lobby class implementation
class Lobby{
	private:
		int lobbyID;
	public:
		Lobby();
		Lobby(int LID);
		void setLobbydetails(int LID);
		~Lobby();
};
```

Lobby.cpp

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

//defalt constructor
Lobby::Lobby()
{
	lobbyID = 0;
}
//overloaded constructor
Lobby::Lobby(int LID)
{
	lobbyID = LID;
}
//Destructor
Lobby::~Lobby()
{
	cout<<"Lobby Destructor called" << endl;
}
```