

Topic : Online Fitness Trainer

Group no : MLB\_01.01\_06

Campus : Malabe

Submission Date :

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** |
| IT21147678 | Jayawardhana E.K | 0760365338 |
| IT21144462 | Ranasinghe T.R | 0770841275 |
| IT21191510 | Mendis A.S.P | 0765409121 |
| IT21195402 | Gunathilaka D.J.V | 0714911894 |
| IT21200588 | A.M.B.D Aththanayake | 0773299695 |

System requirement

1. A registered Customer/visitor can visit the online fitness trainer website.
2. A registered Customer/visitor can search product and services from the website.
3. To become a registered customer, a visitor should be registered by providing his/her personal information such as: Name, Address, contact number and email as mentioned in the login form.
4. Customers can request personalized videos based on their personal requirements from their personal trainer.
5. One order can consist with several products.
6. Customer can purchase the products with several payment methods. (Credit card/Debit/PayPal)
7. Payment should be validated and is given an order ID, once the Transactions is successfully confirmed.
8. Administrator can add new products and services, remove, update the inventory list.
9. Administrator can remove customers based on their past improper behaviors.
10. Store managers arrange the shipment according to the customers’ orders.

11. Customers can provide feedbacks upon receiving their services/products.

12. The financial records are updated according to the transactions. +++++++

13. System notifies the status of the order. +++++

14. Manager assign the staff members according to the order ID.

Noun / Verb analysis

(Nouns are in RED color and Verbs are in BLUE color)

1. A registered Customer/visitor can visit the online fitness trainer website.
2. A registered Customer/visitor can search product and services from the website.
3. To become a registered customer, a visitor should be registered by providing his/her personal information such as: Name, Address, contact number and email.
4. Customers can request personalized videos based on their personal requirements from their personal trainer.
5. One order can consist with several products.
6. Customer can purchase the products with several payment methods. (Credit card/Debit/PayPal)
7. Payment should be validated and is given an order ID, once the payment is successfully confirmed.
8. Administrator can add new products and services, remove, update the inventory list.
9. Administrator can remove customers based on their past improper behaviors.

10.Store managers arrange the shipment according to the customers’ orders.

11. Customers can provide feedbacks upon receiving their services/products.

12. The financial records are updated according to the refund or payment.

13. System notifies the status of the order.

14. Manager assign the staff members according to the order ID.

**Identified Nouns**

Customer - Class

Visitor – Redundant

Products - Class

Services - Class

Personal Information – Attribute of customer

Videos - Class

Personal Trainer - An actor of the system

Order - Class

Payment – Class

Refund - Class

Card - Class (Inherited from “Payment” class)

PayPal - Class (Inherited from “Payment” class)

Order ID - Attribute of order

Administrator - An actor of the system

Store Managers - An actor of the system

Shipment – An operation done by Store Manager

Feedbacks - An operation done by customer

Financial Records - Class

Manager - An actor of the system

Staff Members - An actor of the system

**Identified Classes**

1. Customer \*\*
2. Products \*\*
3. Services\*\*
4. Order\*\*
5. Payment\*\*
6. Card\*\*
7. PayPal\*\*
8. Financial record
9. Discount

**Identified Verbs**

**Customer**

* Visit
* Search Product and services
* Register
* Request videos
* Add products to cart
* Purchase products
* Select a payment method
* Give feedbacks

**Administrator**

* Add products and services
* Remove Inventory list
* Update Inventory list
* Remove customer

**Financial Report**

* **Store transaction details**

**Card**

* **Store card details**

**User**

* **Visit to the system**
* **Register the system**

**Service**

* **Display details of services**
* **Display trainer details**

**Order**

* **Store product amount**
* **Store order details**
* **Order type can store**

**Discount**

**Payment**

**Product**

* **Store product details**

**Paypal**

**Exercise 02:**

**CRC Cards for the System**

|  |  |
| --- | --- |
| **Customer** | |
| **Responsibility** | **Collaborators** |
| Search Product and services | Products/ Services |
| Register |  |
| Request videos | Videos |
| Add products to cart | Products / Cart |
| Purchase products | Products |
| Select a payment method | Payment |
| Give feedbacks |  |

|  |  |
| --- | --- |
| **Administrator** | |
| **Responsibility** | **Collaborators** |
| Add products and services | Products / Services |
| Remove/ update Inventory list | Products / Services |
| Remove customer | Customers |

|  |  |
| --- | --- |
| **Financial report** | |
| **Responsibility** | **Collaborators** |
| Store transaction details | Order |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Discount** | |
| **Responsibility** | **Collaborators** |
| Store discount |  |
| Display discount details |  |
|  |  |

|  |  |
| --- | --- |
| **Card** | |
| **Responsibility** | **Collaborators** |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Order** | |
| **Responsibility** | **Collaborators** |
| Store order details | Payment / Customer |
| Store order type |  |

|  |  |
| --- | --- |
| **Payment** | |
| **Responsibility** | **Collaborators** |
| Store payment details |  |
| Valid payment | Order |
| Display payment details | Card / Paypal |

|  |  |
| --- | --- |
| **Paypal** | |
| **Responsibility** | **Collaborators** |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **Product** | |
| **Responsibility** | **Collaborators** |
| Store product id |  |
| Store product price | Discount |
| Store product description |  |

|  |  |
| --- | --- |
| **Service** | |
| **Responsibility** | **Collaborators** |
| Display details of services |  |
|  |  |

|  |  |
| --- | --- |
| **User** | |
| **Responsibility** | **Collaborators** |
| View the system |  |
| Register to the system |  |
|  |  |

Diagram, engineering drawing

Description automatically generated

User.h

// student id : IT21147678

// Name : Jayawardhana E.K.

// MLB\_01.01\_06

// Malabe\_weekdays

class user

{

protected:

char Name[20];

string Dob;

char Gender[10];

public:

user();

user(const char uName[], string uDob, char const uGender[]);

void askQuestion();

void giveFeedback();

};

User.cpp

// student id : IT21147678

// Name : Jayawardhana E.K.

// MLB\_01.01\_06

// Malabe\_weekdays

#include "user.h"

#include<iostream>

#include<cstring>

using namespace std;

user::user()

{

strcpy\_s(Name, "");

Dob = "";

strcpy\_s(Gender, "");

}

user::user(const char uName[], string uDob, char const uGender[])

{

strcpy\_s(Name, uName);

Dob = uDob;

strcpy\_s(Gender, uGender);

}

void user::askQuestion()

{

}

void user::giveFeedback()

{

}

Administrator.h

// student id : IT21147678

// Name : Jayawardhana E.K.

// MLB\_01.01\_06

// Malabe\_weekdays

#include "customer.h"

class Administrator

{

private:

char name[20];

char adminId[10];

public:

Administrator();

Administrator(const char id[], const char adminName[]);

void removeCustomer(customer \*cus);

void replyToRegUser(customer \* cus);

~Administrator();

};

Administrator.cpp

// student id : IT21147678

// Name : Jayawardhana E.K.

// MLB\_01.01\_06

// Malabe\_weekdays

#include<iostream>

#include<cstring>

#include "Administrator.h"

using namespace std;

Administrator::Administrator()

{

strcpy(adminId, "");

strcpy(name, "");

}

Administrator::Administrator(const char id[], const char adminName[])

{

strcpy(adminId, id);

strcpy(name, adminName);

}

void Administrator::removeCustomer(customer \*cus)

{

}

void Administrator::replyToRegUser(customer\* cus)

{

}

Administrator :: ~Administrator()

{

};

Customer.h

// student id : IT21195402

// Name : Gunathilaka D.J.V.

// MLB\_01.01\_06

// Malabe\_weekdays

#include "user.h"

class customer : public user

{

private:

int RgId;

char telep\_no[20];

public:

customer(){};

customer(const char uName[], string uDob[], const char uGender[],int id, const char telepNum[]) : user(uName, uDob, uGender){};

void setWriteReview(){};

void setMakePayment(){};

int getRegUserId(){};

~customer(){};

};

Customer.cpp

// student id : IT21195402

// Name : Gunathilaka D.J.V.

// MLB\_01.01\_06

// Malabe\_weekdays

#include "customer.h"

#include<iostream>

#include<cstring>

using namespace std;

customer::customer()

{

RgId = 0;

strcpy(telep\_no, "");

}

customer::customer(const char uName[], string uDob[], const char uGender[], int id, const char telepNum[]) : user(uName, uDob, uGender)

{

RgId = id;

strcpy(telep\_no, telepNum);

}

void customer::setWriteReview()

{

}

void customer::setMakePayment()

{

}

int customer::getRegUserId()

{

return RgId;

}

customer::~customer()

{

};

Payment.h

// student id : IT21144462

// Name : Ranasinghe T.R

// MLB\_01.01\_06

// Malabe\_weekdays

#include<iostream>

#include "order.h"

#include "card.h"

#include "paypal.h"

using namespace std;

// #define SIZE 2

class payment

{

protected:

int paymentID;

finacialreport\* finrep;

public:

payment();

payment(int ppaymentid);

void getorder\_details(int oorder\_id, const char type[20], int oqty, double ot\_amount, int paymentID); // : order(int oorder\_id, const char type[20], int oqty, double ot\_amount);

void addcard();

void addpaypal();

virtual void validatePayment();

virtual void display();

~payment();

};

Payment.cpp

// student id : IT21144462

// Name : Ranasinghe T.R

// MLB\_01.01\_06

// Malabe\_weekdays

#include<iostream>

#include "payment.h"

using namespace std;

#define SIZE 2

payment::payment()

{

paymentID = 0000000;

}

payment::payment(int ppaymentid)

{

paymentID = ppaymentid;

}

void payment::getorder\_details(int oorder\_id, const char type[20], int oqty, double ot\_amount, int paymentID)

{

}

void payment::addcard()

{

}

void payment::addpaypal()

{

}

void payment::validatePayment()

{

}

void payment::display()

{

}

payment::~payment()

{

}

Paypal.h

// student id : IT21144462

// Name : Ranasinghe T.R

// MLB\_01.01\_06

// Malabe\_weekdays

#include<iostream>

#include "payment.h"

using namespace std;

#define SIZE 2

payment::payment()

{

paymentID = 0000000;

}

payment::payment(int ppaymentid)

{

paymentID = ppaymentid;

}

void payment::getorder\_details(int oorder\_id, const char type[20], int oqty, double ot\_amount, int paymentID)

{

}

void payment::addcard()

{

}

void payment::addpaypal()

{

}

void payment::validatePayment()

{

}

void payment::display()

{

}

payment::~payment()

{

}

Paypal.cpp

// student id : IT21144462

// Name : Ranasinghe T.R

// MLB\_01.01\_06

// Malabe\_weekdays

#include "paypal.h"

paypal::paypal()

{

strcpy(email, "");

}

paypal::paypal(char pemail[50])

{

email[50] = pemail[50];

}

void paypal::validatePayment()

{

}

void paypal::updatepaypalDetails(char new\_email[50])

{

email[50] = new\_email[50];

}

void paypal::display(double ot\_amount, char p\_email[50])

{

cout << "Rs : " << ot\_amount << " debitted from Your \*\*\*\*\* paypal account : " << p\_email[50];

}

paypal::~paypal()

{

//distroyed................

}

Card.h

// student id : IT21144462

// Name : Ranasinghe T.R

// MLB\_01.01\_06

// Malabe\_weekdays

#include "payment.h"

#include "order.h"

#include<iostream>

using namespace std;

class card :public payment

{

private:

int cardNo;

int cvv\_Code;

char expire\_Date[8];

public:

card();

card(int pcardNo, int pcvv\_Code, char pexpire\_Date[8]);

void validatePayment();

void updatecardDetails(int new\_cardNo, int new\_cvv\_Code, char new\_expire\_Date[8]);

void display(int c\_cardNo, double ot\_amount); // : order(double ot\_amount);

~card();

};

Card.cpp

// student id : IT21144462

// Name : Ranasinghe T.R

// MLB\_01.01\_06

// Malabe\_weekdays

#include "card.h"

#include<iostream>

#include<cstring>

card::card()

{

}

card::card(int pcardNo, int pcvv\_Code, char pexpire\_Date[8])

{

cardNo = pcardNo;

cvv\_Code = pcvv\_Code;

strcpy\_s(expire\_Date, pexpire\_Date);

}

void card::validatePayment()

{

}

void card::updatecardDetails(int new\_cardNo, int new\_cvv\_Code, char new\_expire\_Date[8])

{

cardNo = new\_cardNo;

cvv\_Code = new\_cvv\_Code;

expire\_Date[8] = new\_expire\_Date[8];

}

void card::display(int c\_cardNo, double ot\_amount)

{

cout << "Rs : " << ot\_amount << " debitted from Your \*\*\*\*\* card : " << c\_cardNo;

}

card::~card()

{

//destroyed...................

}

Order.h

// student id : IT21200588

// Name : Aththanayaka A.M.B.D.

// MLB\_01.01\_06

// Malabe\_weekdays

#pragma once

#include<iostream>

#include"order.h"

#define SIZE 4

using namespace std;

class order

{

private:

int order\_id;

int qty;

char type[20];

double t\_amount;

payment\* pay[SIZE];

public:

order();

order(int oorder\_id, const char type[20], int oqty, double ot\_amount);

void order::addpayment(payment\* pay1, payment\* pay2);

virtual void display();

~order();

};

Order.cpp

// student id : IT21200588

// Name : Aththanayaka A.M.B.D.

// MLB\_01.01\_06

// Malabe\_weekdays

#pragma once

#include<iostream>

#include"order.h"

#define SIZE 4

using namespace std;

class order

{

private:

int order\_id;

int qty;

char type[20];

double t\_amount;

payment\* pay[SIZE];

public:

order();

order(int oorder\_id, const char type[20], int oqty, double ot\_amount);

void order::addpayment(payment\* pay1, payment\* pay2);

virtual void display();

~order();

};

Product.h

// student id : IT21200588

// Name : Aththanayaka A.M.B.D.

// MLB\_01.01\_06

// Malabe\_weekdays

#pragma once

#include<iostream>

#include "order.h"

#include "Discount.h"

using namespace std;

class product : public order

{

private:

int p\_id;

int price;

char type[20];

char name[25];

char discription[50];

Discount \*d;

public:

product();

product(int no1);

void displayDiscount();

product(int pp\_id, int pprice, const char ptype[20], const char pname[25], const char pdiscription[50]);

void display();

~product();

};

Product.cpp

// student id : IT21200588

// Name : Aththanayaka A.M.B.D.

// MLB\_01.01\_06

// Malabe\_weekdays

#include"order.h"

#include<string.h>

#include "product.h"

product::product()

{

p\_id = 0;

price = 0;

strcpy(type, "");

strcpy(name, "");

strcpy(discription, "");

d = Dicount(1000);

}

product::product(int no1) {

d = Discount(no1);

}

product::void displayDiscount() {

d->display();

}

product::product(int pp\_id, int pprice, const char ptype[20], const char pname[25], const char pdiscription[50])

{

p\_id = pp\_id;

price = pprice;

strcpy(type, ptype);

strcpy(name, pname);

strcpy(discription, pdiscription);

}

void product::display()

{

}

product::~product()

{

cout << "Delete Product" << endl;

delete d;

cout << "Product is deleted" << endl;

}

Service.h

// student id : IT21195402

// Name : Gunathilaka D.J.V.

// MLB\_01.01\_06

// Malabe\_weekdays

#pragma once

#include<iostream>

#include"order.h"

class service

{

private:

int s\_id;

int price;

char name[20];

char discription[40];

public:

service();

service(int ss\_id, int sprice, const char sname[20], const char sdiscription[40]);

void display();

~service();

};

Service.cpp

// student id : IT21195402

// Name : Gunathilaka D.J.V.

// MLB\_01.01\_06

// Malabe\_weekdays

#include<string.h>

#include"order.h"

#include "service.h"

service::service()

{

s\_id = 0;

price = 0;

strcpy(name, "");

strcpy(discription, "");

}

service::service(int ss\_id, int sprice, const char sname[20], const char sdiscription[40])

{

s\_id = ss\_id;

price = sprice;

strcpy(name, sname);

strcpy(discription, sdiscription);

}

void service::display()

{

}

service::~service()

{

}

Financial report.h

// student id : IT21191510

// Name : Mendis A.S.P.

// MLB\_01.01\_06

// Malabe\_weekdays

class Financial\_report

{

private:

int fId;

Order\* odr;

public:

Financial\_report();

Financial\_report(int ofId, Order\* o);

void displayFinance();

};

Financial report.h

// student id : IT21191510

// Name : Mendis A.S.P.

// MLB\_01.01\_06

// Malabe\_weekdays

#include "Financial\_report.h"

Financial\_report::Financial\_report()

{

}

Financial\_report::Financial\_report(int ofId, Order\* o)

{

fId = ofId;

odr = o;

}

void Financial\_report::displayFinance()

{

cout << "fId" << fId << endl;

odr->displayOrder();

}

Discount.h

// student id : IT21191510

// Name : Mendis A.S.P.

// MLB\_01.01\_06

// Malabe\_weekdays

class Discount

{

private:

{

int discountId;

int discount;

}

public:

Discount();

Discount(int dId, int d);

calDiscount(int d)};

void display();

~Discount();

};

Discount.cpp

// student id : IT21191510

// Name : Mendis A.S.P.

// MLB\_01.01\_06

// Malabe\_weekdays

#include "Discount.h"

Discount::Discount()

{

}

Discount::Discount(int dId, int d)

{

discountId = dId;

discount = d;

}

Discount::calDiscount(int d)

{

if(10000 < pPrice)

{

d = pPrice - (pPrice\*10/100);

}

}

void Discount::display()

{

cout << "Discount Id" << discountId << endl;

cout << "Discount" << discount <<endl;

}

Discount::~Discount()

{

cout << "Delete Discount Id" << dicountId << endl;

cout << "Delete Discount" << discount << endl;

}

Main.cpp

#include <iostream>

#include "customer.h"

#include "card.h"

#include "Administrator.h"

#include "user.h"

#include "service.h"

#include "order.h"

#include "Discount.h"

#include "Financial\_report.h"

#include "payment.h"

#include "product.h"

#include "paypal.h"

using namespace std;

int main()

{

//---- Object creation ------

payment \*ppaypal = new paypal(); // Object payment class

payment \*pcard = new card(); // Object payment class

Discount \*dis = new Discount(); // Object Discount class

customer \*cus = new customer(); // Object customer class

Administrator\* admin = new Administrator(); // Object Administrator class

user \*puser[0] = new customer(); // Object user class

user \*puser[1] = new Administrator(); // Object user class

service \*serv[0] = new service(); // Object service class

order \*ord[0] = new order(); // Object order class

Financial\_report \*finrep[0] = new Financial\_report(); // Object Financial\_report class

product \*pro[0] = new product(); // Object product class

product \*pro[1] = new product(); // Object product class

//----Method Calling------

ppaypal->validatePayment();

ppaypal->display();

pcard->validatePayment();

pcard->display();

dis->calDiscount();

dis->display();

cus->setWriteReview();

cus->setMakePayment();

cus->getRegUserId();

admin->removeCustomer();

admin->replyToRegUse();

puser[0]->askquestion();

puser[0]->givefeedback();

puser[1]->askquestion();

puser[1]->givefeedback();

serv[0]->display();

ord[0]->display();

finrep[0]->displayFinance();

pro[0]->display();

//----Delete Dynamic objects------

delete ppaypal;

delete pcard;

delete dis;

delete cus;

delete admin;

delete puser[0];

delete puser[1];

delete serv[0];

delete ord[0];

delete finrep[0];

delete pro[0];

return 0;

}