**BSc (Hons) in Information Technology**

**IT1050 – Object Oriented Concepts Year 1, Semester 2 2022-June**

**Assignment 1**

**Group Details**

Group Number: PRO\_01

Project Title: Textile & Garment Management System

|  |  |
| --- | --- |
| **Student ID** | **Student Name** |
| IT18225716 | Ganganatha B. A. G. D |
| IT21015380 | Rajapaksh R.W.A.R |
| IT20148904 | Yapa .D.T.D |



**Table of Contents**

Contributions **2**

System requirements for Textile & Garment Management System 3

Identified Classes, Identified Attribute and Identified methods **5**

CRC Cards9

Class/UML Diagrams14

Coding Part15

**Contributions**

IT18225716 – Supervisor, workers and stock (CRC Cards, Class/UML Diagrams and Coding)

IT21015380 - Customer and outfits

IT20148904 - payment and Sales

**Exercise 1:**

**System requirements for Textile & Garment Management System**

01. Supervisor need permission to manage stock.

02. Supervisor only can view stock.

03. Workers need to follow their job role.

04. Supervisor can solve problem of workers.

05. Supervisor have power to control workers.

06. Supervisor can request to the supplier for buy items.

07. Supervisor can add, delete and update stock items.

08. Workers can view profile and they can send the inquiries via Email to the supervisor.

09. Supervisor can view profile and can view workers inquiries.

10. stock can view the available stock and can view damaged stock.

11. Sales data is created by the IDs of the sales

12. The manager has the ability to add, edit and delete the cells belonging to him, through the coustomer ID.

13. After the sales been made, the manager has the capability of adding payement details to the system

14. Once the payment is being made, I will be able to save, delete and update the payment details

NounVerb

01. Supervisor need permission to manage stock.

02. Supervisor only can view stock.

03. Workers need to follow their job role.

04. Supervisor can solve problem of workers.

05. Supervisor have power to control workers.

06. Supervisor can request to the supplier for buy items.

07. Supervisor can add, delete and update stock items.

08. Workers can view profile and they can send the inquiries via Email to the supervisor.

09. Supervisor can view profile and can view workers inquiries.

10. Supervisor can view the available stock and can view damaged stock.

**Identified Classes**

Supervisor

Workers

Stock

Outfits

Customer

payment

Sales

**Identified Attribute**

Supervisor ID

Supervisor Name

Email Address

Worker ID

Worker Name

Worker email

Stock ID

Stock Date

Stock type

Stock items

Stock details

Permission ID

Permission date

Role ID

**Identified methods**

add

delete

update

**CRC Cards**

|  |  |
| --- | --- |
| **Supervisor** |  |
| **Responsibilities:** | **Collaborations:** |
| Login to the system |  |
| System display “Welcome” |  |
| View profile |  |
| View Workers inquiries | Workers |
| Send a request to get the permission for stock managing. | Stock |
| Fill the online form | Stock |

|  |  |
| --- | --- |
| **Workers** |  |
| **Responsibilities:** | **Collaborations:** |
| Login to the system |  |
| System display “Welcome” |  |
| View profile |  |
| View job role |  |
| View the working hours |  |
| Send the inquiries via Email to the supervisor | Supervisor |

|  |  |
| --- | --- |
| **Stock** |  |
| **Responsibilities:** | **Collaborations:** |
| Login to the system | Supervisor |
| System display “Welcome” |  |
| View the available stock | Supervisor, Garments |
| View damaged stock | Supervisor, Garments |
| Add items | Supervisor |
| Delete items | Supervisor |
| Update items | Supervisor |
| Supplier details | Supervisor, Garments |
| Request for buy items | Supervisor, Garments |

|  |  |
| --- | --- |
| **Customer** | |
| **Responsibility** | **Collaborators** |
| Search, view outfits | outfits |
| Purchasing outfits | payments |
|  |  |
|  |  |

|  |  |
| --- | --- |
| **outfits** | |
| **Responsibility** | **Collaborators** |
| Store outfits details |  |
| Display details when barcode enteres |  |
| Add, remove outfits |  |
| Restoke outfits |  |

|  |  |
| --- | --- |
| **payment** |  |
| **Responsibilities:** | **Collaborations:** |
| Add Payment Details | Registered customer |
| Display Payment Details |  |

|  |  |
| --- | --- |
| **sales** |  |
| **Responsibilities:** | **Collaborations:** |
| Add sales Details | Registered customer |
| Display sales Details |  |
|  |  |

**Class/UML Diagrams**

-supervisor\_id: int

-supervisor\_name: string

-email\_address: string

-permission\_id: int

-permission\_date: int

-worker\_id: int

-worker\_name: string

-worker\_email: string

-role\_id: int

**Workers**

**Supervisor**

1..\*

1

+ViewProfile()

+ViewJobRole()

+SendInquiries()

+ViewProfile()

+ViewWorkersInquiries()

+SendRequest()

1

1..\*

**Stock**

-stock\_id : int

-stock\_items: string

-stock \_date: int

-stock\_type: string

-stock\_details: string

+AddStock()

+UpdateStock()

+DeleteStock()

+SerachStock()

+ViewAvailableStock()

+ViewDamagedStock()

+Request()

**Exercise 2:**

**Coding Part**

**Supervisor.h**

//supervisor class

class Supervisor

{

private:

int supervisor\_id;

string supervisor\_name;

string email\_address;

int permission\_id;

int permission\_date;

double salary;

public:

int ViewProfile();

Supervisor(int S\_ID, double sal, char sup\_name[]);

void ViewWorkersInquiries();

int SendRequest();

};

**Supervisor.cpp**

//Supervisor implementation

Supervisor::Supervisor() {

}

Supervisor::Supervisor(int S\_ID, double sal, char sup\_name[]) {

supervisor\_id = S\_ID;

salary = sal;

}

void Supervisor::ViewWorkersInquiries() { }

**Stock.h**

//stock class

class Stock {

protected:

int stock\_id;

string stock\_items;

int stock\_date;

string stock\_type;

string stock\_details;

public:

Stock(int stockID, string stock\_date, string stockItems, string stockType[]);

void AddStock();

void UpdateStock();

void DeleteStock();

void SearchStock();

void ViewAvailableStock();

void ViewDamagedStock();

int Request();

};

**Stock.cpp**

//stock implementation

Stock::Stock() {}

Stock::Stock(int stockID, string stock\_date, string stockItems, string stockType) {

stock\_id = stockID;

}

void Stock::AddStock() {}

void Stock::UpdateStock() {}

void Stock::DeleteStock() {}

void Stock::SearchStock() {}

void Stock::ViewAvailableStock() {}

void Stock::ViewDamagedStock(){}

**Workers.h**

//workers class

class Workers {

private:

int worker\_id;

string worker\_name;

string worker\_email;

int role\_id;

public:

void ViewProfile();

void ViewJobRole();

int SendInquiries();

Workers(int workerId, string workerName, string workerEmail);

};

**Workers.cpp**

//workers implementation

Workers::Workers() {}

Workers::Workers(int workerId, string workerName,string workerEmail) {

worker\_id = workerId;

worker\_name = workerName;

worker\_email = workerEmail;

void ViewProfile();

void ViewJobRole();

}

**Customer.h**

//class customer

{

protected:

string C\_ID;

string name;

string address;

string C\_Number;

Public:

Customer();

Customer(int C\_ID, char name, string C\_add, string Con\_number)

void setCID(int C\_ID);

void setname(string name);

void setaddress(string address);

void setContactnumber(string number);

string getCID();

string getname();

string getaddress();

string getContactNumber();

virtual void displayBasicdetails() = 0; };

**Customer.cpp**

//customer class implementation

Customer:: Customer(int C\_ID, char name, string C\_add, string Con\_number){

Customer ID = C\_ID;

Name = name;

Address = add;

Contact number = number;

void Customer::setC\_ID(int C\_ID)

{

Customer ID = C\_ID;

}

void Customer::setname(char name)

{

Name = name;

}

void Customer::setaddress(string add)

{

Address = add;

}

void Customer::setContactnumber(string number)

{

C\_Number = number;

}

string Customer::C\_ID()

{

return C\_ID;

}

char Customer::name()

{

Return name;

}

string Customer::getaddress()

{

return Address;

}

string Customer::getContactNumber()

{

return C\_Number;

}

**Outfits.h**

//outfits class implementation

class Outfit;

class Outfit

{

private:

string OutfitID;

float price;

Cart\* Car;

public:

Outfit(string pOutfitID, float pPrice, Cart\* pCar);

void displayOut();

};

Outfit::Outfit(string pOutfitID, float pPrice, Cart\* pCar)

{

OutfitID = pOutfitID;

price = pPrice; Car = pCar;

Car->addOutfit(this);

}

void Outfit::displayOut()

{

cout << " OutfitID : " << OutfitID << endl;

cout << " price : " << price << endl;

}

**Payment.h**

class Payment{

private:

int PaymentID;

char PaymentType[10];

double PaymentAmount;

int CustomerID;

public:

Payment(const int cPaymentID, char cPaymentType, double cPaymentAmount,

int cCustomerID);

void AddPaymentDetails();

void ViewPaymentDetails();

~Payment();

};

Payment.cpp

Payment::Payment(const int cPaymentID, char cPaymentType, double cPaymentAmount,

int cCustomerID)

{

stpcpy\_s(PaymentType, cPaymentType);

PaymentID = cPaymentID;

PaymentAmount = cPaymentAmount;

CustomerID = cCustomerID;

}

20

void Payment::AddPayDetails()

{

}

void Payment::ViewPayDetails()

{

}

Payment::~Payment()

{

}

**Sales.h**

class Sales{

private:

int SlaseID;

char Salesdescription[50];

int CustomerID;

public:

Slase(const int cSlaselID, char cSalesdescription[], int cCustomerID);

void AddSales();

void ViewSales();

~Salse();

};

**Slase.cpp**

Slase::Slase(const int cSalseID, char cSalsediscription[], int cCustomerID)

{

stpcpy\_s(Salsediscripition, cSalsediscription);

SalseID = cSalseID;

CustomerID = cCustomerID;

}

void Salse::AddSalse()

{

}

void Salse::ViewSalse()

{

}

Salse::~Salse()

{

}

**Main.cpp**

//Main Program

int main() {

Workers (1,"GG","abc");

Stock (01, 01 / 09, "clothes", "ribbons");

Supervisor (1, 10000, "smith");

return 0;

}