BSc (Hons) in Information Technology

Object-Oriented Concepts – IT1050

Assignment 2

Year 1, Semester 2 **2024 - February**

Topic: Life Insurance Management System

Group no: MLB_WD_04.02_08

Campus: SLIIT MALABE

Submission Date: 2024/5/19

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.

	Student Registration Number	Student Name
01.	IT23228276	Aaqib A.R.
02.	IT23223530	Thakkif Ahamed I.M.
03.	IT23220942	Thanis Ahamed A.R.
04.	IT23222236	Erathnage T.N.
05.	IT23222786	Budara V.P.R.

Table of Contents

1. Description of the Requirements

- Registered customers, Admin, Manager, and Agent are the users of this Life Insurance Management System.

- All the users can visit the website, look at insurance packages, visit terms and conditions, and the feedback of the users.

- Users can log in to the system using the username and the password.

- Registered customer must give their name and ID when registered.

- Registered customers can apply for insurance claims, and active insurance packages, give feedback, and make insurance package payments.

- Also, registered customers can pay for the insurance packages using credit card/debit cards and bank deposits.

- Admin has to give their name and ID when registered.

- Admin has to give his username and Password when he logs into the system.

- Admin can verify insurance claim details of users and Registered customers.

- Admin can add new user details for the database, remove user details from the database, and update user details in the database.

- Agent has to give his username and password when he logs into the system.

- Agent can add new insurance package to system.

- The Manager has to give their username and password when they log into the system.
- Manager can make insurance claims payments.
- Only the manager can approve insurance claims that registered customers apply.
- The manager can renew the feedback sent from customers.
- Also, the manager can generate the reports.

2. Classes Identified:

❖ User

❖ RegisteredCustomer

❖ Admin

❖ Manager

❖ Agent

❖ InsuranceClaims

❖ Packagers

❖ Reports

❖ Feedback

❖ Payment

3. CRC Card

User	
Responsibilities:	**Collaboration:**
Login to the system.	
View insurance packages.	

Registered User	
Responsibilities:	**Collaboration:**
Choose insurance packages.	Insurance Package
Make payment for packages.	Payment
Apply insurance claims.	Insurance Claim
Give feedback.	Feedback

Insurance Package	
Responsibilities:	**Collaboration:**
Store package details.	
Display package details.	
New Insurance Package	Agent

Feedback	
Responsibilities:	**Collaboration:**
Store feedback	
View feedback	

Payment	
Responsibilities:	**Collaboration:**
Store payment details.	
Display payment details.	
Validate payment	Manager

Insurance Claim	
Responsibilities:	**Collaboration:**
Store insurance claim details.	
Display insurance claim details.	
Validate claims	Manager

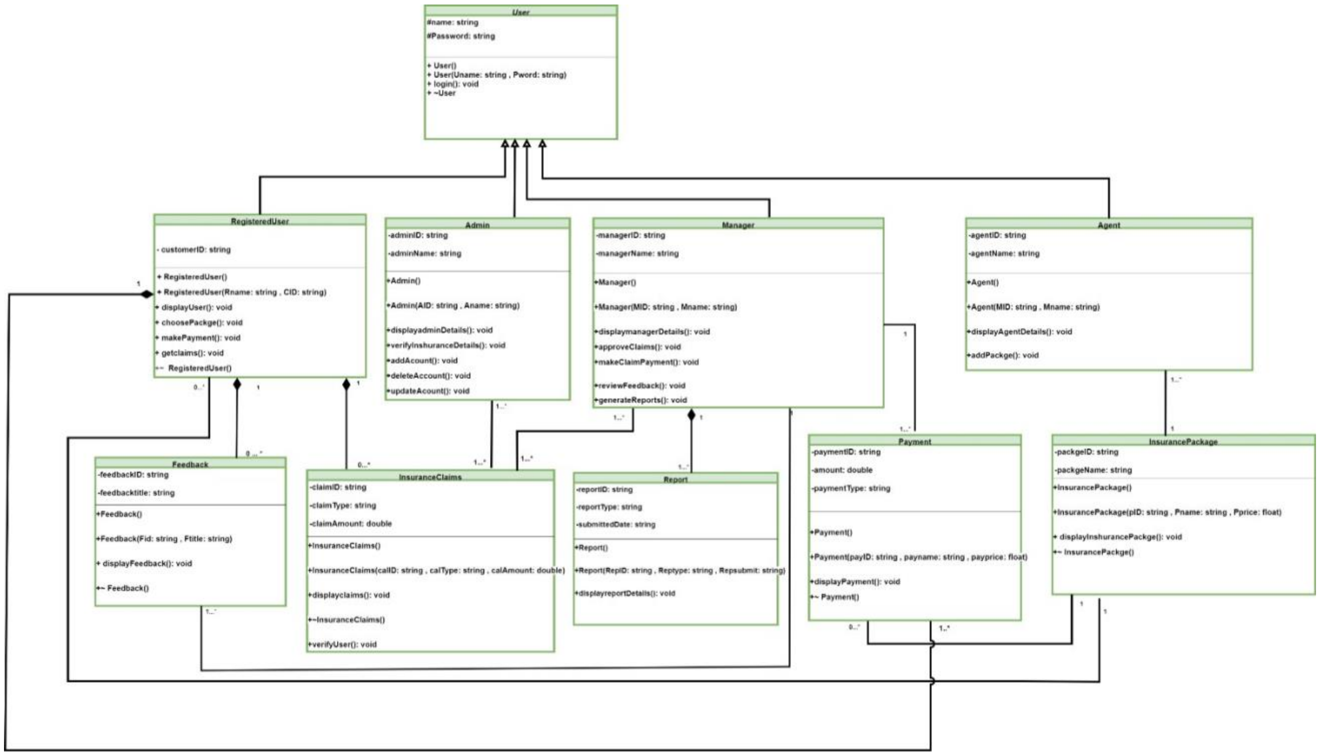
Admin	
Responsibilities:	**Collaboration:**
Verify insurance claim documents.	Insurance Claim
Manage user details	Registered user

Agent	
Responsibilities:	**Collaboration:**
Store agent information.	
Add new insurance package	Insurance package

Manager	
Responsibilities:	**Collaboration:**
Approve insurance claims.	Insurance Claim
Make insurance claim payments	Payment
Review feedback	Feedback
Generate reports	Reports

Report	
Responsibilities:	**Collaboration:**
Generate reports about packages	Insurance packages
Generate transaction Reports	Payment

4. Class Diagram



5. Code for classes.

user.h

```
#pragma once
#include<iostream> using
namespace std; class user
{
protected:
        string name; string
        password;

public:
        user();
        user(string Uname, string pwd);
        void login(string Uname , string pwd);
        ~user();

};
```

user.cpp

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

user::user(){}
user::user(string Uname, string pwd)
{
        name = Uname;
        password = pwd;
}

void user::login(string Uname, string pwd){} user :: ~user(){}
```

main.cpp

```
#include<iostream> #include"user.h"
#include"registeredUser.h"
#include"package.h"
#include"InsuranceClaim.h"
#include"Admin.h"
#include"Manager.h"
#include"Agent.h"
#include"payment.h"
#include"Report.h"
#include"feedback.h"
using namespace std;

int main ()
{
	/*Data insert to register user*/

	registeredUser* ruser = new registeredUser("jonny_sira", "1001" , "jonny siriwardhana" ,
"123","001","Hello","001","health",200000,"0001","transaction 1",50000);

	/*Display register user details*/ ruser-

	>displayUser();


//Data insert to Admin//
Admin* ad1 = new Admin("001", "Risini");

//Dispaly Admin details// ad1-
  >displayadminDetails();

//Data insert to InsuranceClaim//
InsuranceClaim * claim = new InsuranceClaim("001", "Helth", 20000.00, ad1);

//Dispaly claim details// claim-
>displayClaim();


ad1->verifyInsuranceDetails(claim);

//Data insert to InsuranceClaim//
Manager* m1=new Manager("123","Sandumini",claim); m1-
>displayManagerDetails();
m1->approveClaims(); m1-
>reviewFeedback(); m1-
>approveClaims(); m1-
>genaratereports();

//Dispaly claim details//
```

```
claim->displayClaim(); Agent agent;

  /*insertagentdetails*/

  agent.setAgentDetails(1, "John Doe", "john.doe@example.com", 123456789, 987654321);

  /*displayagentdetails*/

  agent.displayDetails();

/*data insert to packge*/
  package* pack = new package("007", "ultimate", 25000.00,ruser);

  /*display packge details*/

  pack->displaypackage(); ruser-
  >ChoosePackge(pack);


/*data insert to report*/

Report report1("ID12345321", "E-Mail" , "19/05/2024");



Delete ruser;
delete ad1; delete
claim; delete m1;
delete pack;
```

RegisteredCustomer.h

```
#pragma once #include
"user.h"
#include"feedback.h"
#include"payment.h"
#include"InsuranceClaim.h"
#include<iostream>
using namespace std; class
package;
class registeredUser:protected user
{
private:
        string username; string
        customerID; feedback* fbk;
        InsuranceClaim* clm;
        payment* pay; package*
        pkg;

public:
        registeredUser();
        registeredUser(string Rname, string CID, string uname, string pwd);
        registeredUser(string Rname, string CID, string uname, string pwd , string fid, string
ftitle);
        registeredUser(string Rname, string CID, string uname, string pwd , string calID,
string calType, double calAmount);
          registeredUser(string Rname, string CID, string uname, string pwd , string payID,
         string payname, float payprice); registeredUser(string Rname, string CID, string
                                                                                uname,
string pwd, string fid, string ftitle , string calID, string calType, double calAmount, string payID, string
payname, float payprice);
        void displayUser();
        void ChoosePackge(package* pack); void
        makePayment(double amt); void getClaims();
        ~registeredUser();

};
```

RegisteredCustomer.cpp

```
#include "registeredUser.h" #include
"user.h" #include<string>
#include<iostream> #include"feedback.h"
#include"payment.h"
#include"InsuranceClaim.h"

using namespace std; registeredUser::registeredUser()
{
        fbk = new feedback("No id", "Not set");
        clm = new InsuranceClaim("No id", "Type not set", 0); pay = new payment("Not
        set","Not set", 0);
}

registeredUser::registeredUser(string Rname, string CID , string uname , string pwd , string
fid, string ftitle) : user(uname, pwd)
{
        username = Rname;
        customerID = CID; name
        = uname; password =
        pwd;
        fbk = new feedback(fid,ftitle);

}
registeredUser::registeredUser(string Rname, string CID, string uname, string pwd , string calID,
string calType, double calAmount) : user(uname, pwd)
{
        username = Rname;
        customerID = CID; name
        = uname; password =
        pwd;
        clm = new InsuranceClaim(calID,calType,calAmount);
```

```
}
registeredUser::registeredUser(string Rname, string CID, string uname, string pwd , string payID,
string payname, float payprice) : user(uname, pwd)
{
        username = Rname;
        customerID = CID; name
        = uname; password =
        pwd;
        pay = new payment(payID, payname, payprice);

}
registeredUser::registeredUser(string Rname, string CID, string uname, string pwd) : user(uname,
pwd)
{
        username = Rname;
        customerID = CID; name
        = uname; password =
        pwd;


}
registeredUser::registeredUser(string Rname, string CID, string uname, string pwd, string fid,
string ftitle, string callD, string calType, double calAmount, string payID, string payname, float
payprice)
{
        username = Rname;
        customerID = CID; name
        = uname; password =
        pwd;
        pay = new payment(payID, payname, payprice);
        clm = new InsuranceClaim(callD, calType, calAmount); fbk = new feedback(fid,
        ftitle);
}

void registeredUser::displayUser()
{
        cout << "Display registered user Name : " << username <<
endl;
        cout << "Display registered user ID : " << customerID
<<endl;
        cout << "Display name : " << name << endl;
        cout << "Display password : *****" << endl <<endl;
```

```
          fbk->displayfeedback(); clm-
          >displayclaims(); pay-
          >displayPayment();

}

void registeredUser::ChoosePackge(package* pack)
{
          pkg = pack;

}
void registeredUser::makePayment(double amt)
{

}
void registeredUser::getClaims()
{

}

registeredUser::~registeredUser()
{
          delete          fbk;
          delete          clm;
          delete          pay;
          delete pkg;
}
```

Admin.h

```
#pragma once #include"user.h"
#include"InsuranceClaim.h"
#include<iostream>
using namespace std;

class InsuranceClaim;
class Admin : protected user
{
private:
        string adminID; string
        adminName; InsuranceClaim*
        claim;

public:
        Admin();
        Admin(string AId, string AName);

        void displayadminDetails();
        void verifyInsuranceDetails(InsuranceClaim* cal1); void addAccount();
        void deleteAccount(); void
        updateAccount();
        ~Admin();
};
```

Admin.cpp

```
##include "Admin.h"
#include "user.h"

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

Admin::Admin() {
        adminID ="";
        adminName = "";
}
```

```
Admin::Admin(string AId, string AName)
{
        adminID = AId;
        adminName=AName;


}

void Admin::verifyInsuranceDetails(InsuranceClaim* cal1)
{
        claim = cal1;


}
void Admin::addAccount() {
}
void Admin::deleteAccount() {
}
void Admin::updateAccount() {
}
void Admin::displayadminDetails() {
        cout << "Admin ID = " << adminID << endl
                << "Admin Name = " << adminName << endl;


}
Admin::~Admin(){
    delete claim;
}       cout << "Admin ID = " << adminID << endl
                << "Admin Name = " << adminName << endl;


}
Admin::~Admin(){
    delete claim;
}
```

Manager.h

```
#include<string> #include<iostream>
#include "InsuranceClaim.h" #pragma
once
using namespace std;

class InsuranceClaim; class
Manager
{
private:
        string managerID; string
        managerName; InsuranceClaim*
        claim;
public:
        Manager();
        Manager(string MID,string Mname,InsuranceClaim* cla1); void
        displayManagerDetails();
        void approveClaims(); void
        makeClaimPayment(); void
        reviewFeedback(); void
        genaratereports();
        ~Manager();

};
```

Manager.cpp

```
#include"InsuranceClaim.h" #include"Manager.h"

#include <iostream> class


InsuranceClaim;
```

```
Manager::Manager() { managerID = "";
        managerName = "";
}
Manager::Manager(string  MID, string Mname, InsuranceClaim* cla1)

{
        managerID = MID;
        managerName = Mname;
        claim = cla1;
}


       void Manager::approveClaims() {
                   cout << "Claims approved by Manager " << managerName <<
endl;
}
void Manager::makeClaimPayment() {
        cout << "Claim payment made by Manager " << managerName <<
endl;

}
void Manager::reviewFeedback() {
        cout << "Feedback reviewed by Manager " << managerName <<
endl;
}
void  Manager::genaratereports() {
        cout << "Reports generated by Manager " << managerName <<
endl;
}
void Manager::displayManagerDetails() {
        cout << endl << "Management ID = " << managerID << endl << "Manager Name = " <<
managerName << endl;
        claim->displayClaim();
}
Manager::~Manager()
{
        delete claim;
}
```

Agent.h

```
#pragma once

#include<iostream>
#include <string>

using namespace std;

class Agent{ private:
        intagent_id; string
        name; string email; int
        NIC;
        int contact_no;

public:
        void setAgentDetails(int id, string name, string email, int NIC, int contact_no);
        void displayDetails(); int
        getAgentId();
};

#endif // AGENT_H
```

Agent.cpp

```
#include "agent.h"

void Agent::setAgentDetails(int id, string username, string mail, int nic, int ct_no) {
        agent_id = id; name =
        username; email =
        mail; NIC = nic;
        contact_no = ct_no;
}

void Agent::displayDetails() {
        cout << "Agent ID: " << agent_id << endl; cout << "Agent Name:
        " << name << endl; cout << "Agent Email: " << email << endl;
        cout << "Agent NIC: " << NIC << endl;
        cout << "Agent Contact number: " << contact_no << endl;
}

int Agent::getAgentId() { return agent_id;
}
```

InsuranceClaims.h

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

class InsuranceClaim
{
private:
	string claimID; string
	claimType; string
	claimAmount;

public:
	InsuranceClaim();
	InsuranceClaim(string calID, string calType, double calAmount);
	void displayclaims();
	~InsuranceClaim(); void
	verifyuser();

};
```

InsuranceClaims.cpp

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

InsuranceClaim::InsuranceClaim(){}

InsuranceClaim::InsuranceClaim(string calID, string calType, double calAmount)
{
	claimID = calID; claimType =
	calType; claimAmount =
	calAmount;
}
```

```
void InsuranceClaim::displayclaims()
{
        cout << "Claim Id : " << claimID << endl; cout << "Claim Type : " <<
        claimType <<endl;
        cout << "claim amount : " << claimAmount <<endl<< endl;

}
InsuranceClaim::~InsuranceClaim(){}

void InsuranceClaim::verifyuser()
{

}
```

Packagers.h

```
#pragma once #include<iostream>
#include"registeredUser.h" using
namespace std; class registeredUser;
class package
{
private:
       string packageID; string
       packgeName; double price;
       registeredUser* ruser;

public:
       package();
       package(string pid, string pname , double pprice , registeredUser* regUser);
       void displaypackage();
       ~package();




};
```

Packagers.cpp

```
#include "package.h"

#include<iostream> using
namespace std;

package:: package(){}
package::package(string pid, string pname, double pprice, registeredUser* regUser)
{
        packageID = pid;
          packgeName = pname;
          price = pprice; ruser =
          regUser;
}
void package::displaypackage()
{
        cout << "Package ID : " << packageID << endl; cout << "Package name : "
        << packgeName << endl; cout << "Package price : " << price << endl;

        ruser->displayUser();
}
package::~package()
{
        delete ruser;
}
```

Reports.h

```
#pragma once #include
<string>

using namespace std; class

Report
{
private:

                string reportID; string
                reportType; string
                submittedDate;

public:

                Report();
                Report(string RepID, string Reptype, string
Resubmit);
                void displayreportDetails();
};
```

Reports.cpp

```
#include           "Report.h"
#include          <iostream>
#include <string>

using namespace std; Report ::

Report () {}

Report :: Report(string RepID, string Reptype, string Resubmit)
{
        reportID = RepID; reportType =
        Reptype; submittedDate = Resubmit;
}

void Report :: displayreportDetails()
```

```
{
        cout << "Report ID : " << reportID << endl; cout << "Report Type : " <<
        reportType << endl;
        cout << "Submitted Date : " << submittedDate << endl;
}
```

Feedback.h

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

feedback::feedback(){} feedback::feedback(string fid, string ftitle)
{
        feedbackID = fid; feedback_title =
        ftitle;
}
void feedback::displayfeedback()
{
        cout << "Feedback ID : " << feedbackID << endl;
        cout << "Display title : " << feedback_title <<endl<<
endl;
}

feedback::~feedback(){}
```

Feedback.cpp

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


class feedback
{
private:
        string feedbackID; string
        feedback_title;

public:
        feedback();
        feedback(string fid, string ftitle); void displayfeedback();
        ~feedback();
};
```

Payment.h

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

payment::payment() {}
payment::payment(string payID, string payname, double payprice)
{
        paymentID = payID; amount =
        payprice; paymentType = payname;

}

void payment::displayPayment()
{
        cout << "Payment ID : " << paymentID << endl; cout << "Amount : "
        << amount << endl;
        cout << "payment type : " << paymentType << endl << endl;
}
payment::~payment(){}
```

06. Individual Contribution

IT23228276: Aaqib A. R

Designed the Classes

- Manager
- Payment

Coded for the above-mentioned Classes.
Created the CRC Cards for the relevant Classes.
Coded the relevant part in the main.cpp file.
Formatted and corrected the whole document.

IT23223530: Ahamed I.M. T

Designed the Classes

- User
- Report

Coded for the above-mentioned Classes.
Created the CRC Cards for the relevant Classes.
Coded the relevant part in the main.cpp file.

IT23220942: Thanis Ahamed A.R

Designed the Classes

- Agent
- Packagers

Coded for the above-mentioned Classes.
Created the CRC Cards for the relevant Classes.

IT23222236: Erathnage T. N

Designed the Classes

- Registered Customer
- Feedback

Coded for the above-mentioned Classes.
Created the CRC Cards for the relevant Classes.
Drew the class diagram using draw.io.

IT23222786: Budara V.P.R.

Designed the Classes

- Admin
- Insurance Claims

Coded for the above-mentioned Classes.
Created the CRC Cards for the relevant Classes.
Wrote the Description of the Requirements Section.