Topic : Life Insurance Management System

Group no :MLB_10.01_13

Campus : Malabe

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**
IT21183072	H.M.Witharana	0761130417
IT21183454	P.A.D.S.L.Jayathilaka	0767416466
IT21157950	J.H.H.N.Karunarathna	0767388981

(01)User Requirements

- ✓ An unregistered user in a life insurance management system needs to first register providing personal details such as name,age,gender,contact no,email,marital status.
- ✓ Then the registered user can access into the system using their credentials.
- ✓ The registered user can update user details if there are any updates in the registration information.
- ✓ Clients can add details of dependents on their register application.
- ✓ Clients can get help from an Insurance agent regarding all their problems.
- ✓ After registration, the client can view the company description, rules and conditions.
- ✓ If registered user wants to apply for insurance, they can apply insurance cover.
- ✓ The client can make the payment for the insurance cover he/she has chosen.
- ✓ The system administrator notifies the client that the payment is successful once the client's payment has been validated.
- ✓ Registered user can request for claim.
- ✓ Clients can request for claim.
- ✓ The system has a staff that communicate directly with clients. Insurance agents and branch managers belong to that staff.
- ✓ Insurance agent and branch managers can create their profile on the system including their personal information such as Name, worked, contact no, salary, gender, address, email, NIC.
- ✓ Insurance agents can view client profile.
- ✓ Branch managers can guide insurance agents and clients.
- ✓ Claim department accepts claim requests.
- ✓ Branches store details of insurance agent, branch manager, customer, claim,claim department and insurance.

Identified classes using Noun Analysis

- Client
- Insurance
- Payment
- Claim
- Staff
- Insurance agent
- Branch manager
- Claim department
- Branches

CRC Cards for the online Airline Ticket Reservation system

Client	
Responsibility	Collaborators
Log in to the system	
Update/edit client's details	
Apply for insurance	insurance
Make the payment	payment
Request for claim	Claim

Insurance	
Responsibility	Collaborators
Store description of the insurance	
Add new insurance plan	
Store insurance owner's details	

Payment	
Responsibility	Collaborators
Store payment details	
Store payment report	
Release the claim payment	Claim

Claim	
Responsibility	Collaborators
Store claim request applications	
Send claim request to claim department	

Staff	
Responsibility	Collaborators
Communicate directly with clients	Client
Guide clients	Client
Provide reports to the branch	Insurance agent, Branch manager

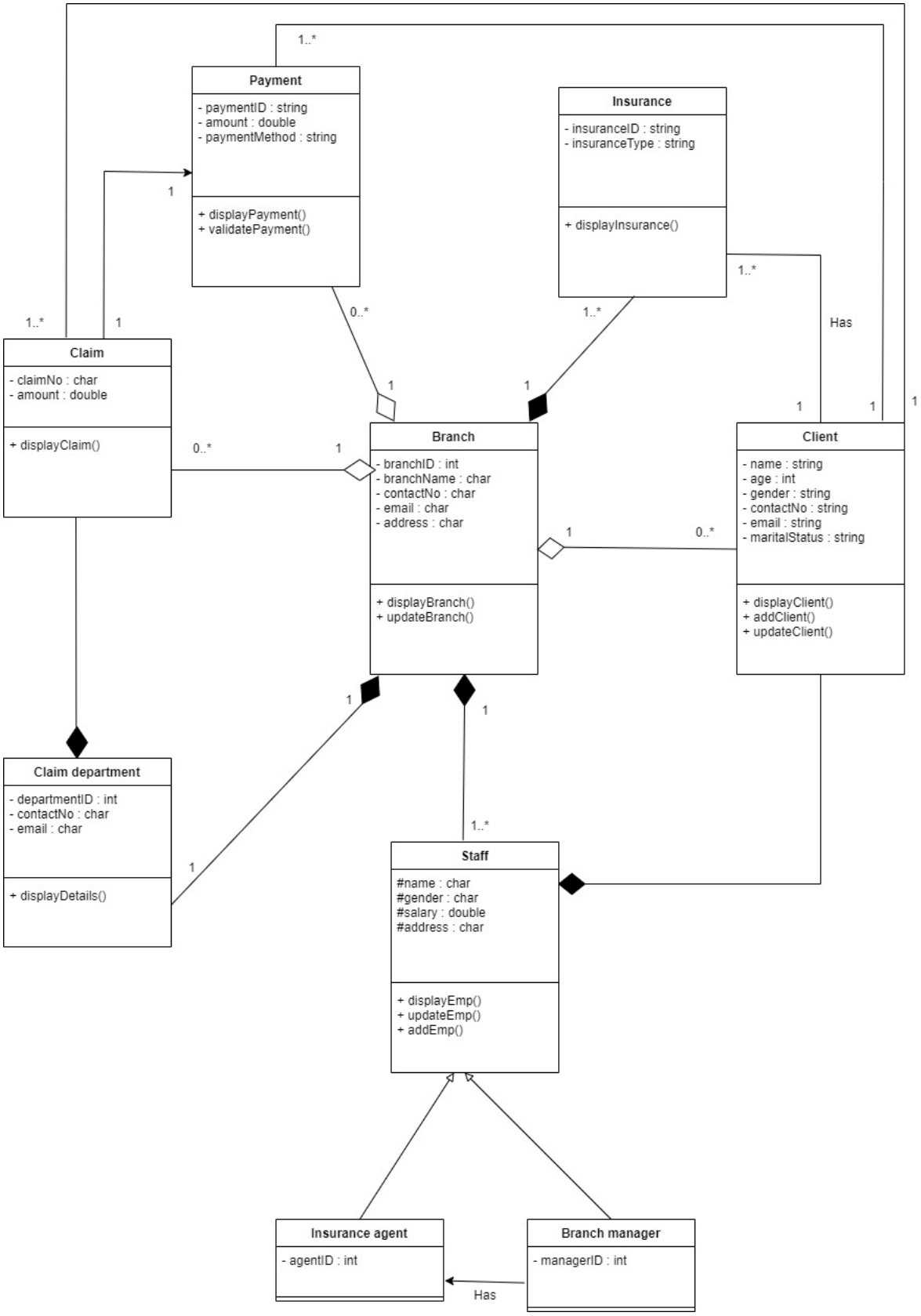
Insurance agent	
Responsibility	Collaborators
Log in to the system	
Update/edit agent's details	
Provide information about services for client	
Solve the client's problem	Branch manager

Branch manager	
Responsibility	Collaborators
Log in to the system	
Update / edit manager's details	
Solve the client problems	
Solve the agent's problems	
Check agents performance	

Claim department	
Responsibility	Collaborators
Analyse the claim request	Claim
Accept/ reject claim	

Branch	
Responsibility	Collaborators
Store client details	Client
Store agent details	Insurance agent
Store staff details	Staff
Store claim details	Claim
Store payment details	Payment
Store insurance details	Insurance

Class diagram for the Life insurance management sysytem



C++ Cording

```
#include<iostream>
#include<cstring>
#include<string>
using namespace std;

class staff {
protected:
	char name[20];
	char gender[7];
	double salary;
	char address[20];

public:
	staff() {
		strcpy(name, "Malani");
		strcpy(gender, "Female");
		salary = 75000;
		strcpy(address, "Matara");
	}
	void updateEmp() {}
	void addEmp() {}
	void display() {
		cout << "This is staff class" << endl;
	}
	void displayEmp() {
		cout << name << "__" << gender << "__" << salary << "__" << address
<< endl;
	}
};

class InsuranceAgent : public staff {
protected:
	int agentid;

public:
	InsuranceAgent() {
		strcpy(name, "Nimal");
		agentid = 10021;
		strcpy(gender, "Male");
		salary = 45000;
		strcpy(address, "Hambantota");
	}
	void display() {
		cout << "This is InsuranceAgent class" << endl;
	}
	void displayEmp() {
		cout << name << "__" << agentid << "__" << gender << "__" << salary
<< "__" << address << endl;
	}
};

class BranchManager : public staff {
protected:
	int managerid;

public:
	BranchManager() {
		strcpy(name, "Sunimal");
		managerid = 420;
```

```
		strcpy(gender, "Male");
		salary = 55000;
		strcpy(address, "Galle");
	}
	void display() {
		cout << "This is BranchManager class" << endl;
	}
	void displayEmp() {
		cout << name << "__" << managerid << "__" << gender << "__" <<
salary << "__" << address << endl;
	}
};


//client class
class Client
{
private:
	string name;
	int age;
	string gender;
	string contactNo;
	string email;
	string martitalStatus;
public:
	Client();
	Client(string Name, int Age, string Gender, string ContactNo, string
Email, string MartitalStatus);
	void displayClient();
	void addClient();
	void updateClient();
	~Client();
};

//insurance class

class insurance {
private:
	string insuranceID;
	string insuranceType;

public:
	insurance();
	insurance(string insID, string insType);
	void displayClient();
	~insurance();
};



//payment class

class payment {
private:
	string paymentID;
	string paymentMethod;
	double amount;

public:
	payment();
	payment(string payID, string payMeth, double Amount);
	void displayClient();
```

```
	~payment();
};



//class Claim
#pragma once

#include "ClaimDepartment.h"

class Claim : public ClaimDepartment {
protected:
	char ClaimNo[50];
	double amount;

public:
	Claim();
	Claim(char cNo[50], double amnt);


	void displayClaim();

};


//class Branch
#pragma once

class Branch {
protected:
	int BranchID;
	char BranchName[50];
	char contactNo[15];
	char email[100];
	char address[100];

public:
	Branch();
	Branch(int bID, char bName[], char cNo[], char mail[], char adrs[]);
	void displayBranch();
};

//class claimDepartment
#pragma once

class ClaimDepartment {
protected:
	int departmentID;
	char contactNo[15];
	char email[100];


public:
	ClaimDepartment();
	ClaimDepartment(int depID, char cNo[], char mail[]);
	void displayDetails();
};



//method implementation
```

```
//client class method implementation

Client::Client()
{

}
Client::Client(string Name, int Age, string Gender, string ContactNo, string
Email, string MartitalStatus) {
	name = Name;
	age = Age;
	gender = Gender;
	contactNo = ContactNo;
	email = Email;
	martitalStatus = MartitalStatus;
}
void Client::displayClient()
{
	cout << " name: " << name << endl;
	cout << " age: " << age << endl;
	cout << "gender : " << gender << endl;
	cout << "contact NO: " << contactNo << endl;
	cout << "email: " << email << endl;
	cout << "martitalStatus: " << martitalStatus << endl;
	cout << endl;
}
void Client::addClient()
{

}
void Client::updateClient()
{

}
Client::~Client()
{



//insurance class method implementation


insurance::insurance()
{

}

insurance::insurance(string insID, string insType)
{
	insuranceID = insID;
	insuranceType = insType;
}

void insurance::displayClient()
{
	cout << "insuranceid" << insuranceID << endl;
	cout << "insurancetype" << insuranceType << endl;
	cout << "........................................." << endl;


}
```

```
insurance::~insurance()
{

}



//payment class method implementation


payment::payment()
{

}

payment::payment(string payID, string payMeth, double Amount)
{
	paymentID = payID;
	paymentMethod = payMeth;
	amount = Amount;


}

void payment::displayClient()
{
	cout << "paymentid" << paymentID << endl;
	cout << "paymentmethod" << paymentMethod << endl;
	cout << "amount" << amount << endl;
	cout << "....................................................." <<
endl;

}

//claim class method implementation
#include "Claim.h"

Claim::Claim() {

	strcpy(ClaimNo, "");
	amount = 0;

}
Claim::Claim(char cNo[], double amnt) {

	strcpy(ClaimNo, cNo);
	amount = amnt;

}
void Claim::displayClaim() {

	cout << "ClaimNo :" << ClaimNo << endl;
	cout << "amount :" << amount << endl;
	cout << endl;
}

//Branch class method implementation
#include "Branch.h"

Branch::Branch() {
	BranchID = 0;
```

```
	strcpy(BranchName, "");
	strcpy(contactNo, "");
	strcpy(email, "");
	strcpy(address, "");


}
Branch::Branch(int bID, char bName[], char cNo[], char mail[], char adrs[]) {

	BranchID = bID;
	strcpy(BranchName, bName);
	strcpy(contactNo, cNo);
	strcpy(email, mail);
	strcpy(address, adrs);

}
void Branch::displayBranch() {
	cout << "Branch ID : " << BranchID << endl;
	cout << "Branch Name : " << BranchName << endl;
	cout << "Branch contactNo : " << contactNo << endl;
	cout << "Branch email: " << email << endl;
	cout << "Branch address: " << address << endl;
	cout << endl;


}

//ClaimDepartment class method implementation
#include "ClaimDepartment.h"

ClaimDepartment::ClaimDepartment() {
	departmentID = 0;
	strcpy(contactNo, "");
	strcpy(email, "");

}
ClaimDepartment::ClaimDepartment(int depID, char cNo[], char mail[]) {

	departmentID = depID;
	strcpy(contactNo, cNo);
	strcpy(email, mail);

}
void ClaimDepartment::displayDetails() {
	cout << "Department ID : " << departmentID << endl;
	cout << "contactNo: " << contactNo << endl;
	cout << "email: " << email << endl;
	cout << endl;


}
```

```
#include "Branch.h"
#include "ClaimDepartment.h"
#include "Claim.h"


int main() {
	staff S1;
	InsuranceAgent A1;
	BranchManager M1;

	S1.display();
	S1.displayEmp();
	A1.display();
	A1.displayEmp();
	M1.display();
	M1.displayEmp();

//client

Client* c1 = new Client("puishpa", 60, "Female", "0767416466",
"pushpa@gmail.com", "Married");

Client* c2 = new Client("thilan", 25, "male", "0725643875", "thilan@gmail.com",
"single");

Client* c3 = new Client("Shanuka", 30, "Female", "0767416466",
"shanuka@gmail.com", "Married");

c1->displayClient();
c2->displayClient();
c3->displayClient();



//insurance
insurance* i1 = new insurance("100", "1");
insurance* i2 = new insurance("200", "2");

i1->displayClient();
i2->displayClient();

//payment

payment* p1 = new payment("100", "1", 2000);
payment* p2 = new payment("200", "2", 4000);

p1->displayClient();
p2->displayClient();



	//Branch
	Branch k(587, "pilimathalawa", "081248291", "587@gmail.com",
"327,pilimathalawa,kandy");
	k.displayBranch();

	//ClaimDepartment
	ClaimDepartment c(135, "081229944", "135@gmail.com");
	c.displayDetails();

	//Claim
```

```
        Claim m("5000", 8000);
        m.displayClaim();

}


return 0;

}
```

Individual Contribution – cording part

IT21183072 – Coded Staff class

Coded insurance agent class

Coded branch manager class

IT21183454-Coded client class

Coded insurance class

Coded payment class

IT21157950-Coded branch class

Coded claim department class

Coded claim class