



Topic : Life Insurance Management System

Group no : MLB_03.02_06

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
IT21491412	Ganepola U.I	0764918935
IT21561252	Liyanage L.S.T	076-9646152
IT21480218	D.M.L.M. Deheragoda	0703816915
IT21787072	D.A.S. Costa	0701683807
IT21572388	Arampath A.M.S.Y	0767702709

Requirements

1. A visitor has access to view policy details, if satisfied the visitor may register and will receive a confirmation letter via email.
2. The registered user logs into the user account page by entering the login credentials and the user will be able to manage profile and policies.
3. Purchasing policies can be done via online payments or bank transfers. A registered user has the ability to request a claim at any time using this system.
4. When a claim is requested, the agent will be notified and the request will be reviewed.
5. If the request follows the relevant guidelines, request will be passed to the manager.
6. Approval will be given after the managers inspection.
7. Once the claim request is approved the user gets the option to rate and give feedback regarding the service.
8. Manager can manage the transactions which happens inside the system.
9. Admin handles the database and updates the system.
10. Manager can manage the agents and policies.
11. Manager can view reports.

Nouns

1. Visitor
2. Letter
3. Email
4. Registered user
5. User account page
6. Login credentials
7. User
8. Profile
9. Policy
10. Payments
11. Bank transfer
12. Claim
13. Agent
14. Request
15. Guidelines
16. Manager
17. Rate
18. Feedback
19. Service
20. Transaction
21. Admin
22. Database

Verbs

Noun	Verb
visitor	View (policy) satisfied Register
Registered user	Receive (confirmation email) Log in Entering (login credentials) Manage (profile and policies) Purchasing (policy) Request (claim) Give(feedback)
agent	Notified (claim) Reviewed (claim)
claim	Requested Follows (guidelines) Passed Approved (manager)
manager	Approval(claim) Manage (transactions, agents , policies)
Admin	Handles (database) Updates (System)

Classes

1. Visitor
2. Registered user
3. Policy
4. Payment
5. Claim
6. Agent
7. Manager
8. Admin
9. Feedback
10. Report

CRC Cards

Class Name : Claim	
Responsibilities	Collaboration
Request claim Review Approval	Registered user Agent Manager

Class Name : Visitor	
Responsibilities	Collaboration
View policy details Register	Policy

Class name: Agent	
Responsibilities:	Collaborations:
Check for claims	
Manage FAQs	
Manage feedbacks	

Class name: policy	
Responsibilities:	collaborations
View policy	
Add policy	Manager
Update policy	Manager
Delete policy	Manager

Class Name - Admin	
Responsibilities	Collaboration
Handle the database	
Update system	

Class Name - Payment	
Responsibilities	Collaboration
Display payment methods	User Database
Add payment details	
Verify Payment	
Issue receipt	
Keep a record of payment details	

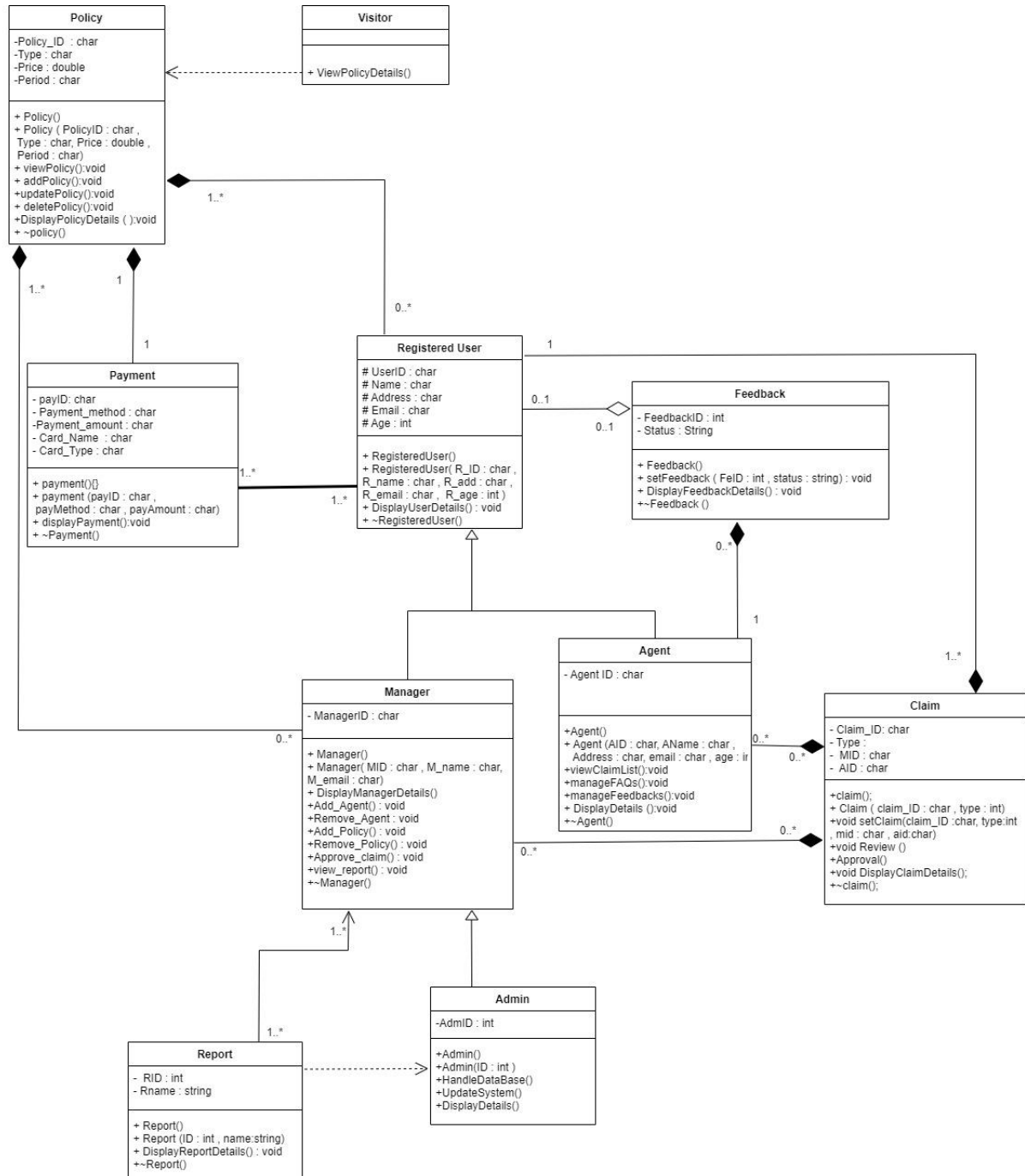
Class Name: Manager	
Responsibilities	Collaboration
Approval claim Manage the transactions Manage the agents and policies	Claim Agent, Policy

Class Name: Registered User	
Responsibilities	Collaboration
Log in user account Manage profile and policies Purchasing policies Request claim Give feedback	 Policy Claim Feedback

Class name: Report	
Responsibilities:	collaborations
Generate Reports	
Update Reports	
Delete Reports	Admin
View Reports	Manager

Class name: Feedback	
Responsibilities:	Collaborations:
Send Feedback	Registered User

Exercise 1



Exercise 2

Admin.cpp

//IT21572388 - Arampath A.M.S.Y

```
#include<iostream>
```

```
#include<string.h>
```

```
#include "admin.h"
```

```
using namespace std;
```

```
Admin::Admin(){
```

```
}
```

```
Admin::Admin(const char A_ID ){
```

```
    strcpy(AdmId , A_ID);
```

```
}
```

```
void Admin::HandleDataBase(){
```

```
}
```

```
void Admin::UpdateSystem(){
```

```
}
```

```
void Admin::DisplayDeatils()
{
    cout << "Admin ID : " << AdmID << endl;
    Manager::DisplayDeatils();
}
```

Admin.h

//IT21572388 - Arampath A.M.S.Y

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

```
class Admin: public Manager{
```

```
protected:
```

```
    char AdmID;
```

```
public:
```

```
    Admin();
```

```
    Admin(char A_ID );
```

```
    void HandleDataBase();
```

```
void UpdateSystem();  
  
void DisplayDeatils();  
  
}
```

Agent.cpp

```
//IT21787072_COSTA D.A.S  
  
#include<iostream>  
  
#include<string.h>  
  
#include"agent.h"  
  
using namespace std;  
  
agent::agent(){  
  
}  
  
agent::agent(int AID, char name, char address,char email , int age);  
  
  
void agent::viewClaimList() {  
  
  
  
}  
  
void agent::manageFAQs() {
```

```
}  
  
void agent::manageFeedbacks() {  
  
}  
  
void DisplayDetails(int AID)  
{  
  
    cout<<"agent ID:"<<AID <<endl; }  
agent::~~agent(){  
  
    cout<<"agent deleted" <<endl;  
  
}
```

Agent.h

```
//IT21787072_COSTA D.A.S  
  
#include<registeredUser.h>  
  
#include<claim.h>  
  
#include<feedback.h>  
  
  
class agent
```

```
{  
private:  
    int AID;  
    int telNo;  
public:  
    agent();  
    agent(char name[],char address[],char email[],int age);  
    virtual void displayDetails();  
    void viewClaimList();  
    void manageFAQs();  
    void manageFeedbacks();  
    ~agent();  
};
```

Claim.cpp

//IT21491412 -Ganepola U.I

```
#include <cstring>
```

```
#include<iostream>
```

```
#include"claim.h"
```

```
using namespace std;
```

```
claim::claim() {
```

```
}
```

```
claim::claim(const char claimid[],int type) {
```

```
    strcpy(Claim_ID, claimid);
```

```
    Type = type;
```

```
}
```

```
void claim::setClaim(const char claimid[], int type, const char mid[], const  
char aid[]) {
```

```
    strcpy(Claim_ID, claimid);
```

```
    Type = type;
```

```
    strcpy(MID, mid);
```

```
    strcpy(AID, aid);
```

```
}
```

```
void claim::Review() {
```

```
}
```

```
void claim::Approval() {
```



```
}
```

```
void claim::DisplayClaimDetails() {  
    cout << "Claim ID : " << Claim_ID << endl;  
    cout << "Type : " << Type << endl;  
    cout << "Manager ID : " << MID << endl;  
    cout << "AID : " << AID << endl;  
}  
  
claim::~~claim() {  
    cout << "Claim deleted." << endl;  
}
```

Claim.h

```
//IT21491412 -Ganepola U.I
```

```
#pragma once
```

```
class claim {
```

```
private:
```

```
    char Claim_ID[10];

    int Type;

    char MID[6];

    char AID[6];


public:

    claim();

    claim(const char claimid[], int type);

    void setClaim(const char claimid[], int type, const char mid[], const
char aid[]);

    void Review();

    void Approval();

    void DisplayClaimDetails();

    ~claim();

};
```

Feedback.cpp

```
//IT21561252
```

```
//Liyanage.L.S.T
```

```
#include<iostream>
```

```
#include<cstring>
```

```
#include "feedback.h"
```

```
using namespace std;
```

```
Feedback::Feedback()
```

```
{
```

```
}
```

```
void Feedback::setFeedback(int ID, string status)
```

```
{
```

```
    feedbackID = ID;
```

```
    fstatus = status;
```

```
}
```

```
void Feedback::displayFeedbackDetails()
```

```
{
```

```
    cout << "ID : " << ID << endl;
```

```
    cout << "STATUS : " << status << endl;
```

```
}
```

```
Feedback::~~Feedback () {
```

```
cout << "Feedback deleted" << endl;  
}
```

Feedback.h

```
//IT21561252
```

```
//Liyanage.L.S.T
```

```
class Feedback{
```

```
private:
```

```
    int feedbackID;
```

```
    string fstatus;
```

```
Public:
```

```
    Feedback ();
```

```
    void setFeedback(int feID, string fstatus);
```

```
    void displayFeedbackDetails();
```

```
    ~Feedback ();
```

```
};
```

Manager.cpp

//IT21480218 - Deheragoda D.M.L.M

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

```
#include<cstring>
```

```
#include<iostream>
```

```
using namespace std;
```

```
Manager::Manager()
```

```
{
```

```
    Manager_ID = 0;
```

```
    strcpy(Name, "");
```

```
    strcpy(Email, "");
```

```
}
```

```
Manager::Manager(int M_ID, const char M_Name[], const char  
M_Email[])
```

```
{
```

```
    Manager_ID = M_ID;
```

```
    strcpy(Name, M_Name);
```

```
    strcpy(Email, M_Email);
```

```
}
```

```
void Manager::DisplayDetails()
{
    cout << "Manager ID: " << Manager_ID << endl;
    cout << "Manager Name: " << Name << endl;
    cout << "Manager Email: " << Email << endl;
}
```

```
void Add_Agent()
```

```
{
```

```
}
```

```
void Remove_Agent()
```

```
{
```

```
}
```

```
void Add_policy()
```

```
{
```

```
}
```

```
void Remove_policy()
```

```
{
```

```
}
```

```
void Approve_claim()
```

```
{
```

```
}
```

```
void view_report()
```

```
{
```

```
}
```

```
Manager::~~Manager()
```

```
{
```

```
    cout << "Manager Removed" << endl;
```

```
}
```

Manager.h

//IT21480218 - Deheragoda D.M.L.M

#pragma once

class Manager

{

private:

int Manager_ID;

char Name[100];

char Email[100];

public:

Manager();

Manager(int M_ID, const char M_Name[], const char M_Email[]);

void DisplayDetails();

void Add_Agent();

void Remove_Agent();

void Add_policy();

void Remove_policy();

void Approve_claim();

void view_report();


```
    ~Manager();  
};
```

Payment.cpp

```
//IT21572388 - Arampath A.M.S.Y
```

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

```
using namespace std;
```

```
Payment::Payment(){  
    strcpy(payID,"");  
    strcpy(Payment_method,"");  
    amount = 0;  
}
```

```
Payment::Payment(const char P_ID[] ,const char P_method[] , double  
P_amount){
```

```
strcpy(payID,P_ID[]);

strcpy(Payment_method,P_method);

amount = P_amount;

}


void Payment::displayPayment(){

    cout << "Payment ID : " << payID << endl;

    cout << "Payment method : " <<Payment_method << endl;

    cout << "Amount : " << amount << endl;

}


Payment::~~Payment()

{

    cout << "Deleting payment" << endl;

}
```

Payment.h

//IT21572388 - Arampath A.M.S.Y

```
class Payment
{
    private:

        char payID[10];

        char Payment_method[25];

        double amount;


    public:

        Payment();

        Payment(const char payID[] , const char
        Payment_method[] , double amount);

        void displayPayment();

        ~Payment();


};
```

Policy.cpp

```
//IT21787072_COSTA D.A.S
```

```
#include<iostream>
```

```
#include<string.h>
```

```
#include"policy.h"
```

```
using namespace std;
```

```
policy::policy(){
```

```
}
```

```
policy::policy(int PID, char Ptype, double Pprice,char Pperiod);
```

```
void policy::viewPolicy() {
```

```
}
```

```
void policy::addPolicy() {
```

```
}
```

```
void policy::updatePolicy() {
```

```
}
```

```
void policy::deletePolicy() {
```

```
}
```

```
void DisplayPolicyDetails(int PID,char Ptype,double Pprice,char Pperiod)
{
    cout<<"policy ID:"<<PID <<endl;
    cout<<"policy type:"<<Ptype <<endl;
    cout<<"policy price:"<<Pprice <<endl;
    cout<<"policy period:"<<Pperiod <<endl;
}

policy::~~policy(){
    cout<<"policy deleted" <<endl;
}
```

Policy.h

//IT21787072_COSTA D.A.S

#include<visitor.h>

#include<registeredUser.h>

#include<payment.h>

#include<manager.h>

class policy

{

private:

char policyID;

char type;

double price;

char period;

public:

policy();

policy(char policyID[],char type[],double price[],char
period[]);

void viewPolicy();

void addPolicy();

void updatePolicy();

void deletePolicy();

```
void DisplayPolicyDetails();  
  
~policy();  
  
};
```

Report.cpp

```
//IT21561252
```

```
//Liyanage.L.S.T
```

```
#include<iostream>
```

```
#include<cstring>
```

```
#include"report.h"
```

```
using namespace std;
```

```
Report::Report ()
```

```
{
```

```
}
```

```
void Report::setReport(int id, string name)
```

```
{
```

```
        rID = id;

        rName = name;

    }

    void Report::displayReportDetails()

    {

    }

    Report::~~Report ()

    {

    }
```

Report.h

//IT21561252

//Liyanage.L.S.T

```
class Report{

private:

    int rID;

    string rName;
```


Public:

```
    Report ();  
  
    void setReport(int id, string name);  
  
    void displayReportDetails();  
  
    ~Report ();  
  
};
```

User.cpp

//IT21480218 - Deheragoda D.M.L.M

```
#include "User.h"
```

```
#include<cstring>
```

```
#include<iostream>
```

```
using namespace std;
```

```
Registered_User::Registered_User()
```

```
{
```

```
    UserID = 0;
```

```
    strcpy(Name, "");
```

```
        strcpy(Address, "");  
        strcpy(Email, "");  
        Age = 0;  
    }
```

```
Registered_User::Registered_User(int R_ID, const char R_Name[], const  
char R_Address[], const char R_Email[], int R_Age)
```

```
{  
    UserID = R_ID;  
    strcpy(Name, R_Name);  
    strcpy(Address, R_Address);  
    strcpy(Email, R_Email);  
    Age = R_Age;  
}
```

```
void Registered_User::DisplayDetails()
```

```
{  
    cout << "Registered User ID: " << UserID << endl;  
    cout << "Registered User Name: " << Name << endl;  
    cout << "Registered User Address: " << Address << endl;  
    cout << "Registered User Email: " << Email << endl;  
    cout << "Registered User Age: " << Age << endl;  
}
```

```
}
```

```
Registered_User::~~Registered_User()
```

```
{
```

```
    cout << "Registered_User Removed" << endl;
```

```
}
```

User.h

```
//IT21480218 - Deheragoda D.M.L.M
```

```
#pragma once
```

```
class Registered_User
```

```
{
```

```
private:
```

```
    int UserID;
```

```
    char Name[50];
```

```
    char Address[100];
```

```
    char Email[100];
```

```
    int Age;
```

public:

Registered_User();

Registered_User(int R_ID, const char R_Name[], const char
R_Address[], const char R_Email[], int R_Age);

void DisplayDetails();

~Registered_User();

};

Visitor.cpp

//IT21491412 -Ganepola U.I

#include <cstring>

#include<iostream>

#include"visitor.h"

using namespace std;

void visitor::viewPolicyDetails(){

```
}
```

Visitor.h

```
//IT21491412 -Ganepola U.I
```

```
#pragma once
```

```
class visitor {
```

```
private:
```

```
public:
```

```
void viewPolicyDetails() {};
```

```
};
```

Main.cpp

```
//IT21491412 -Ganepola U.I
```

```
//IT21561252 - Liyanage L.S.T
```

```
//IT21572388 - Arampath A.M.S.Y
```

```
//IT21787072_COSTA D.A.S
```

//IT21480218 - Deheragoda D.M.L.M

```
#include <iostream>
```

```
#include<cstring>
```

```
#include"claim.h"
```

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

```
#include "User.h"
```

```
#include "Payment.h"
```

```
#include "admin.h"
```

```
#include "feedback.h"
```

```
using namespace std;
```

```
int main() {
```

```
    claim *claim1 = new claim();
```

```
    claim1->setClaim("cl001", 3, "M001", "A001");
```

```
    claim1->DisplayClaimDetails();
```

```
    delete claim1;
```

```
//-----
```

```
    Manager* M1 = new Manager();
```

```
M1 = new Manager(1000, "Minosha Deheragoda",  
"minosha2000@gmail.com");
```

```
M1->DisplayDetails();
```

```
cout << endl;
```

```
delete M1;
```

```
//-----
```

```
Registered_User * R1 = new Registered_User();
```

```
R1 = new Registered_User(0001, "Shehan Gunasekara", "No 97,  
New Road, Kiribathgoda", "shehan2000@gmail.com", 21);
```

```
R1 -> DisplayDetails();
```

```
cout << endl;
```

```
delete R1;
```

```
//-----
```

```
Feedback *Feedback1 = new Feedback;
```

```
Feedback1 ->setFeedback(877,"In progress");
```

```
Feedback1 ->displayFeedbackDetails();
```

```
cout << endl;
```

```
//-----
```

```
Payment *Payment1 = new Payment;
```

```
Payment1->Payment("PM456" , "Cash" , 15000);
```

```
Payment1 ->displayPayment();
```

```
cout << endl;
```

```
//-----
```

```
admin *admin1 = new Payment;
```

```
admin1->Payment("AD456" , "Kamal" , "N0 5 , Temple road , Colombo"  
, "kamalad@gmail.com" , 36 );
```

```
admin1 ->DisplayDeatils();
```



```
cout << endl;
```

```
//-----
```

```
agent*agent1=new agent();
```

```
agent*a1=new agent("A001","Kevin");
```

```
agent*a2=new agent("A002","Sahan");
```

```
agent1->add agent(a1,a2);
```

```
delete a1;
```

```
delete a2;
```

```
agent*a3=new agent("A003","Sadun");
```

```
agent*a4=new agent("A004","Kasun");
```

```
agent1->add agent(a3,a4);
```

```
agent1->displayDetails();
```

```
cout << endl;
```

```
//-----  
policy *p1= new policy("p001","full","5000","12months");  
    visitor *v1 = new visitor("Sadun");  
  
    v1-> viewPolicyDetails();  
  
    v1-> displayPolicyDetails();  
  
cout <<endl;  
  
//-----  
  
}
```