BSc (Hons) in Information Technology

Object Oriented Concepts – IT1050

Assignment 2

Year 1, Semester 2 **2023-January**

Topic :Vehicle Insurance Management System

Group no : MLB_04.01_04

Campus : Malabe

Submission Date :11 /06/2023

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
IT22887108	KUMARAGE N D	0769963908
IT22316486	HEWAGAMA A D	0761252976
IT22336590	SIRIWARDANA B J A	0775920586
IT22326904	ALWIS D A C	0754538287
IT22338334	SHEHARA I G D S	0775265486

BSc (Hons) in Information Technology

Object Oriented Concepts – IT1050

Assignment 2

Year 1, Semester 2 2023-January

Description

SAFE travels is a vehicle insurance company which we created an online platform to simplify customers' vehicle life. There are two users in this system. Registered user can visit the insurance system through web browser .registered user can enter to the website.after, system needs correct login credentials to login to the system. Only one user can register only once. Registered users need to fill the vehicle details form and registered users can update vehicle information as he/she wishes. Every registered user in this system may have one or more vehicles and he/she can select any insurance type. The system can display insurance details to the registered user and registered users must make payments for their insurance policies by uploading payment details to the system and also he/she can edit the payment details. The system calculates the payments. Registered user can receive claim ,and for that process it needs claim details, provided by the customers and system display claim details to the customers. And also system provides claim amount and payment report after checking claim and payment details.

Policy manager is another user of this system. He/she can login to the system by using his/her login credentials. Policy manager can create policies, remove policies and system display policy details to the registered users.

Identified Classes

1. User
2. RegisteredUser
3. Policymanager
4. Claim
5. Insurance
6. Vehicle
7. Payment
8. Policy
9. Report

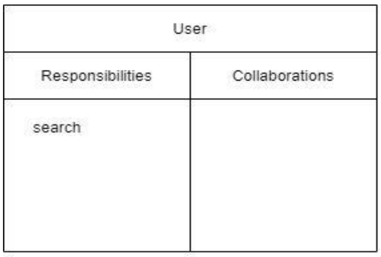
BSc (Hons) in Information Technology

Object Oriented Concepts – IT1050

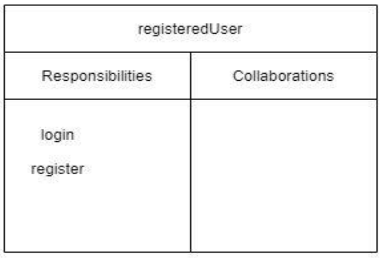
Assignment 2

Year 1, Semester 2 **2023-January**

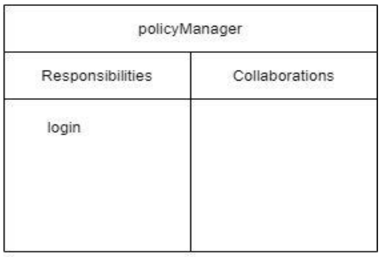
<u>CRC Cards</u>



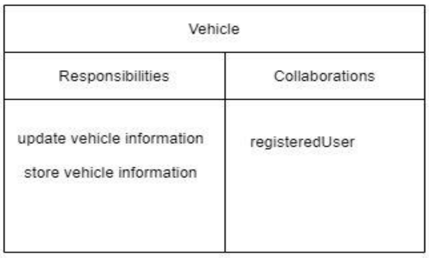
User	
Responsibilities	Collaborations
search	



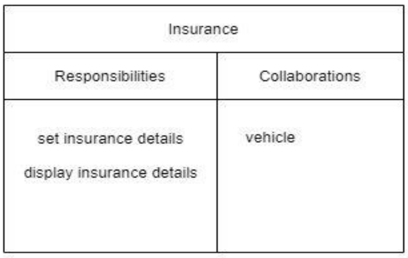
registeredUser	
Responsibilities	Collaborations
login	
register	



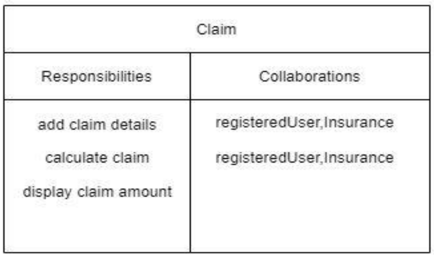
policyManager	
Responsibilities	Collaborations
login	



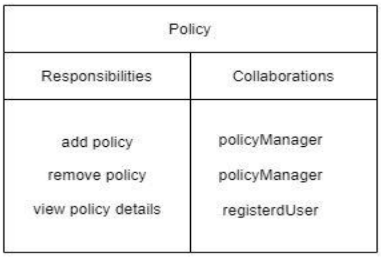
Vehicle	
Responsibilities	Collaborations
update vehicle information	registeredUser
store vehicle information	



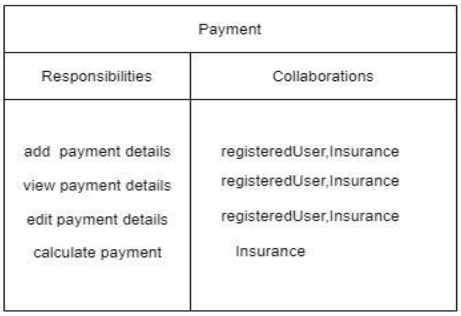
Insurance	
Responsibilities	Collaborations
set insurance details	vehicle
display insurance details	



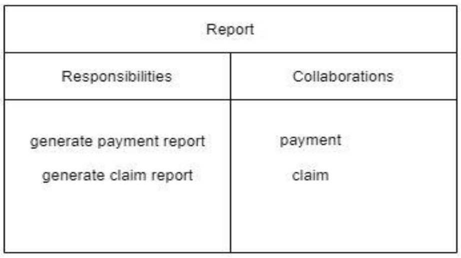
Claim	
Responsibilities	Collaborations
add claim details	registeredUser,Insurance
calculate claim	registeredUser,Insurance
display claim amount	



Policy	
Responsibilities	Collaborations
add policy	policyManager
remove policy	policyManager
view policy details	registerdUser



Payment	
Responsibilities	Collaborations
add payment details	registeredUser,Insurance
view payment details	registeredUser,Insurance
edit payment details	registeredUser,Insurance
calculate payment	Insurance



Report	
Responsibilities	Collaborations
generate payment report	payment
generate claim report	claim

BSc (Hons) in Information Technology

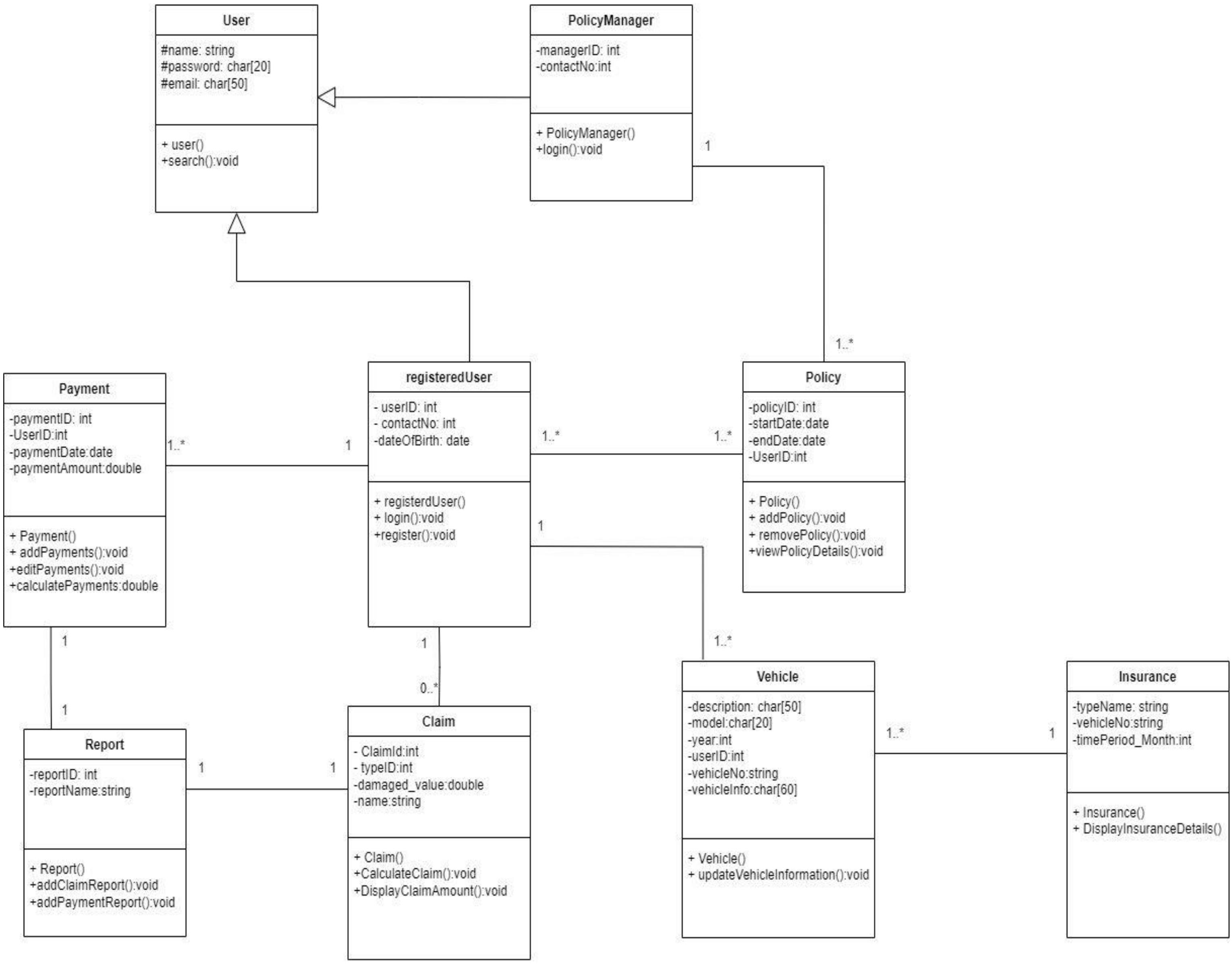
Object Oriented Concepts – IT1050

Assignment 2

Year 1, Semester 2 **2023-January**

Class Diagram

Vehicle Insurance Management System



BSc (Hons) in Information Technology

Object Oriented Concepts – IT1050

Assignment 2

Year 1, Semester 2 **2023-January**

Coding

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

class User;
class registerdUser;
class Policymanager;
class Claim;
class Insurance;
class Report;
class Payment;
class Vehicle;
class Policy;
```

1. User

```
class User {
protected:
  char name[45];
  char password[20];
  char email[50];

public:
  void search();

  User();
  User(char name[45], char password[20], char email[50]);
};

User::User() {
  strcpy(name, "");
  strcpy(password, "");
  strcpy(email, "");
}
```

2.Register User

```
class registeredUser : public User {
private:
  int userID;
  int contactNo;
  char dateOfBirth[10];
  Claim *cl[2]; // bi directional associaton with Claim
  Vehicle *veh; // bi directional associaton with Vehicle
  Payment *pay; // bi directional association with Payment
  Policy *pol;  // bi directional association with Policy

public:
  registeredUser();
  registeredUser(char name[45], char password[20], char email[50], int UserID,
                          char ContactNo[10], char dateOfBirth[10]);
  void Login(){};
  void Register(){};
};
registeredUser::registeredUser() {
  strcpy(name, "");
  strcpy(password, "");
  strcpy(email, "");
  userID = 0;
  contactNo = 0;
  strcpy(dateOfBirth, "");
}
```

3.Policy Manager

```
class Policymanager : public User {
private:
  int managerID;
  int ContactNo;

public:
  Policymanager();
  Policymanager(char name[45], char password[20], char email[50], int managerID,
               char ContactNo[10]);
  void Login(){};
};
Policymanager::Policymanager() {
  strcpy(name, "");
  strcpy(password, "");
  strcpy(email, "");
  managerID = 0;
  ContactNo = 0;
}
```

4.Claim

```
class Claim {
private:
  int ClaimId;
  int instypeID;
  double damaged_value;
  string username;
  registerdUser *user; // bi directional associaton with registerdUser
  Report *rep;

public:
  Claim();
  Claim(int cl_id, int ins_id, double value, string uname);
  void CalculateClaim();
  double DisplayClaimAmount();
  ~Claim();
};
// constructor implimantation
Claim::Claim() {
  ClaimId = 0;
  instypeID = 0;
  damaged_value = 0.00;
  username = " ";
}
void Claim::CalculateClaim() {}
double Claim::DisplayClaimAmount() {}
```

5.Insurance

```
class Insurance {
private:
  string typeName;
  string vehicleNo;
  int timePeriod_Month;
  Vehicle *veh; // Bi-directional with vehicle

public:
  Insurance();
  Insurance(string name, string ve_no, int timePeriod_Month);
  void DisplayInsuranceDetails();
  // Destructure
  ~Insurance();
};
// constructor implimentation
Insurance::Insurance() {
  typeName = " ";
  vehicleNo = " ";
  timePeriod_Month = 0;
}

void Insurance::DisplayInsuranceDetails() {}
```

6.Vehicle

```
class Vehicle {
private:
  // vehicle attributes
  char description[50];
  char model[20];
  int year;
  int userID;
  string vehicleNo;
  char vehicleInfo[60];
  Insurance *Ins[2];       // Bi- directional with Insurance
  registeredUser *user[5]; // Bi- directional with registerdUser

public:
  // Constructors
  Vehicle();
  Vehicle(const char description[], const char mdl[], int year,
          const char user[], int vehicleNo);
  // update vehicle information method
  void updateVehicleInformation();
};

Vehicle::Vehicle() {
  strcpy(description, "");
  strcpy(model, "");
  vehicleNo = "";
  strcpy(vehicleInfo, "");
  year = 0;
  userID = 0;
}
```

7.Payment

```
class Payment {
private:
  int paymentID;
  int userID;
  string paymentDate;
  double paymentAmount;
  registeredUser *user[5];

public:
  Payment();
  void addPayments();
  void editPayments();
  double calculatePayments();
};

Payment::Payment() {
  paymentID = 0;
  userID = 0;
  paymentDate = " ";
  paymentAmount = 0.00;
}
void Payment::addPayments(){

};
void Payment::editPayments(){

};
double Payment::calculatePayments(){

};
```

8.Policy

```
class Policy {
private:
  int policyID;
  string startDate;
  string endDate;
  int UserID;
  registeredUser *user[2];
  Policymanager *Pm;

public:
  Policy();
  void addPolicy();
  void removePolicy();
  void viewPolicyDetails();
};
Policy::Policy() {
  policyID = 0;
  startDate = " ";
  endDate = " ";
  UserID = 0;
}
void addPolicy(){

};
void removePolicy(){

};
void viewPolicyDetails(){

};
```

9.Report

```
class Report {
private:
  int reportID;
  string reportName;

public:
  Report();
  void addClaimReport();
  void addPaymentReport();
  Claim *Cl[1]; // bi-directional with Claim class
  Payment *pay; // bi - directional with payment
};

Report::Report() {
  reportID = 0;
  reportName = " ";
};
void Report::addClaimReport(){

};
void Report::addPaymentReport(){

};

int main() {

  User u1;
  registeredUser r1;
  Policymanager p1;
  Claim *claimobj = new Claim();
  Insurance *Insuranceobj = new Insurance();
  Vehicle *Vehicleobj = new Vehicle();
  Payment *Paymentobj = new Payment();
  Policy *Policyobj = new Policy();

  return 0;
}
```

BSc (Hons) in Information Technology

Object Oriented Concepts – IT1050

Assignment 2

Year 1, Semester 2 **2023-January**

Individual Contribution

ID_No	Name	Contribution
1.IT22887108	Kumarage N D	• CRC card(Policy,report) • Code segment of classes Policy and report
2.IT22316486	Hewagama A D	• CRC card(Payment) • Code segment of class Payment
3.IT22336590	Siriwardana B J A	• CRC card(User,registeredUser,policyManager) • Code segments of classes User,registeredUser and policyManager
4.IT22326904	Alwis D A C	• CRC card(Claim,Insurance) • Code segments of classes Claim and Insurance
5.IT22338334	Shehara I G D S	• Description • CRC card(Vehicle) • Code segments of class Vehicle