

Health Insurance Database Design

Harshitha Yerraguntla, Shreya Hagalahalli Shrinivas, Varun Potlacheruvu
Sanjose State University

Keywords Definition

- **Deductible**-The amount that has to be reached by the insurance holder within the policy/plan period before the insurance company starts sharing the bill/claim.
- **Co-Insurance**-Once the deductible is reached, the insurance holder pays a certain percentage of claim amount defined by co-insurance and the rest percentage is covered by the insurance company.
- **Out Of Pocket Limit** -This is the maximum out of pocket amount spent by the insurance holder for the policy/plan period. Once this amount is reached, the insurance company will pay the rest of the bill/claim amount.

Introduction

We are designing a Health Insurance Database System to showcase how an insurance company maintains records of its members based on their plans and corresponding hospital bills, claims from the provider.

A provider in a healthcare environment is someone or someplace that provides services to a patient. These services are billed to patients, the payments for which can be claimed from the insurance company based on the member's insurance plan.

Technologies: list of technologies are provided here

Database Engine : MySQL.

Versions: Wamp Server 3.1.9 , Php 5.6.40 , MySQL 5.7.26, Google chrome.

Block Diagram: Implementation of the health insurance application is displayed in Fig.1.

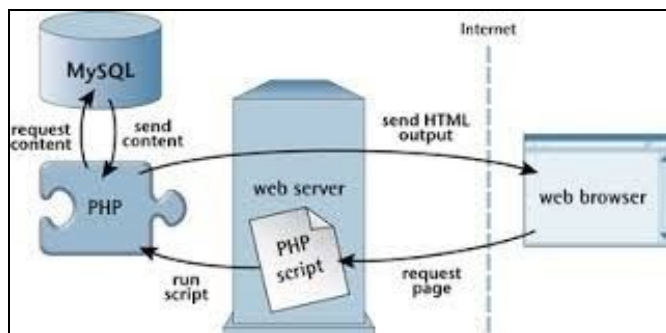


Fig.1. Health insurance application setup

Functionalities

- Members of the Insurance Company will be able to view and update their profile information ,view available plans and available in-network doctors associated with the insurance company.
- Members of the company are also able to view the details of their current plan consisting of their deductible amount reached , out of pocket amount reached, status of claims etc and raise claims if any
- Doctors generate claim to the members of the insurance company who seek treatment/service. The claim details are sent to the insurance provider for verification and approval/denial.
- Insurance provider will look at the claim information of the member he is assigned and process the claim .If claim is approved, the member record is updated and claim status is changed to approved from pending .If the claim is denied, the claim status is updated as denied in the claim table.
- Insurance provider is also able to update the plan information and in-network doctor information
- A statement for every claim is generated indicating the claim status and the amount covered by insurance company upon approval of the claim.

Entity Relationship Diagram:

The Entity Relationship diagram is highlighted to indicate small changes made. The Bill entity has been removed since most of its attributes were covered in the claim entity and hence redundancy has been avoided for a better Normalization. Fig. 2 displays ER diagram.

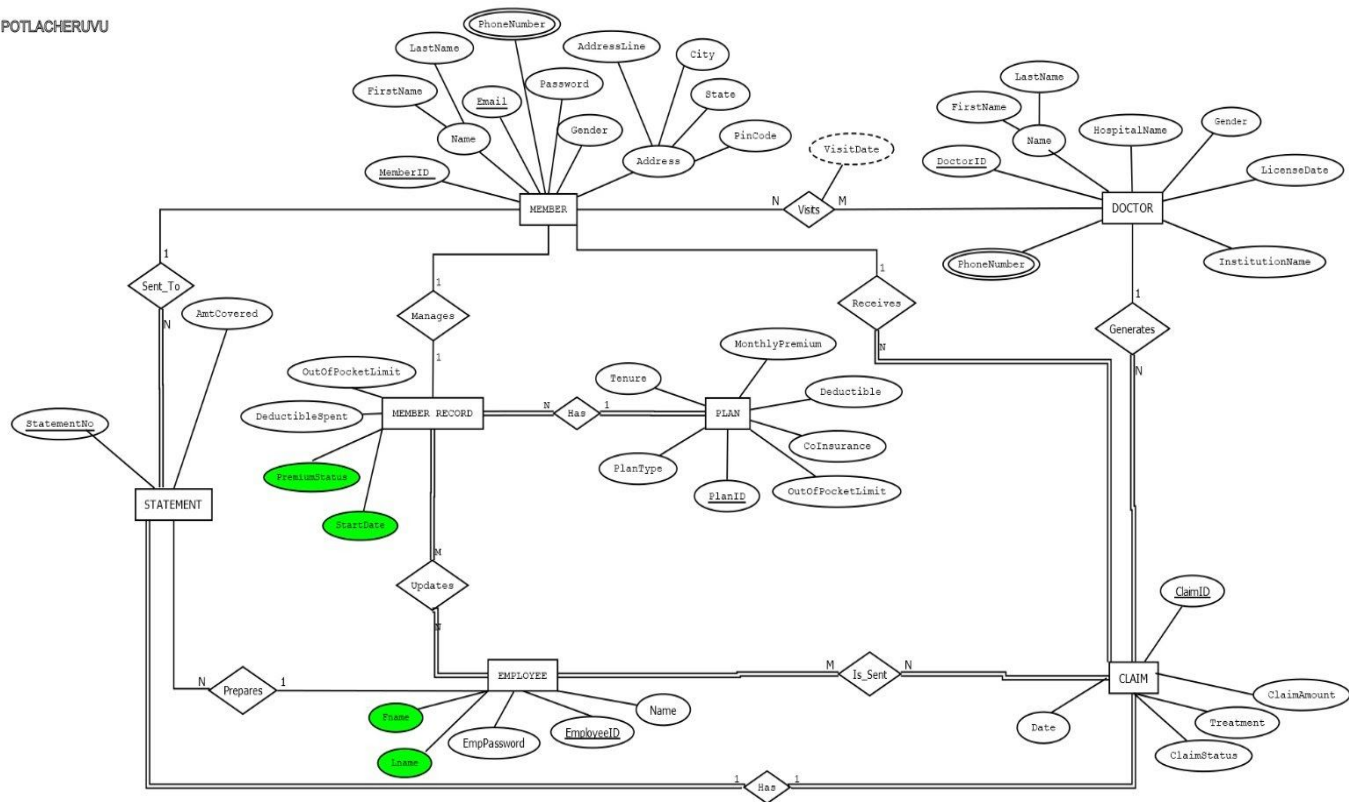


Fig.2. ERD for Health insurance database.

Specification of Database Objects (Tables and columns)

Normalization : All tables are upto 3rd NF

Member

This table contains information of all the members enrolled to suitable plans offered by the insurance company. This table enables the insurance company to have complete profile information of the members ,including name, address, contact, gender and date of birth. Each member is identified by the member Id.

Field	Data Type		Description	PK/FK
MemberId	integer	Not- Null	This field defines the unique Id of each member	PK

FirstName	varchar(20)	Not- Null	Records the first name of the member	
LastName	varchar(20)	Not- Null	Records the last name of the member	
Email	varchar(50)	Not- Null	records the email Id of the member which is required field for user login	
Password	varchar(70)	Not- Null	records the password of the member which is a required field for member login	
Gender	varchar(10)	Null	Records the gender of the member	
AddressLine	varchar(30)	Null	Records the street address of the member	
City	varchar(20)	Null	Records the city in which the member resides	
State	varchar(2)	Null	Records the state in which the member resides	
PinCode	integer	Null	Records the pincode of the member address	
DOB	date	Not Null	Records the dob of the member	

FDs:MemberId -> FirstName,LastName,Email>Password,Gender,AddressLine,City,StatePinCode,DOB

Doctor

A doctor provides service or medical treatment to patients and generates a claim to be sent to the provider. The table consists of details of the doctor consisting of name, contact, gender, the hospital or clinic name and institution name. Each Doctor is identified by doctor ID.

Field	Data Type		Description	PK/FK
DoctorId	integer	Not- Null	This field defines the unique Id of each doctor	PK
FirstName	varchar(20)	Not- Null	Records the first name of the doctor	
LastName	varchar(20)	Not- Null	Records the last name of the doctor	
HospitalName	varchar(30)	Null	Contains the hospital/clinic name where he is currently working	
AddressLine	Varchar(30)		Contains the address of his place of practice	
Gender	Varchar(10)	Null	Records the gender of the doctor	
LicenseDate	Date	Not-Null	Contains the date of license owned by the doctor	

InstitutionName	varchar(40)	Null	Contains the name of institution from where the doctor graduated	
-----------------	-------------	------	--	--

FDs: DoctorId ->

FirstName,LastName,HospitalName,AddressLine,Gender,LicenseDate,InstitutionName

Plan

Plan table contains the details of the available plans in an insurance company .Each plan is identified by its plan Id.

Field	Data Type	Description	PK/FK
PlanId	integer	This field defines the unique Id of each plan	PK
PlanType	varchar(30)	Contains the name of the plan	
MonthlyPremium	decimal(10,2)	Contains the monthly premium amount	
Deductible	decimal(10,2)	Defines the deductible limit to be reached for the plan	
Co-Insurance	integer	Defines the co-Insurance percentage for the plan type	
OutOfPocketLimit	decimal (10,2)	Defines the maximum out of pocket limit for the plan type	

Tenure	integer	Defines the tenure period for which the plan is valid.	
--------	---------	--	--

FDs: PlanId -> PlanType,MonthlyPremium,Deductible,Co-Insurance,OutOfPOCKETLimit,Tenure

Employee

This table provides details of the employees in an insurance company.Each employee is identified by his employee Id.Employees of the insurance company verify and process the outstanding claims of members enrolled in a plan in an insurance company.

Field	Data Type	Description	PK/FK
EmployeeID	integer	This field defines the unique Id of each employee	PK
Fname	varchar(10)	Records the first name of the employee	
Lname	varchar(30)	Records the first name of the employee	
EmpPassword	varchar(100)	records the password of the member which is a required field for employee login	

FDs: PlanId -> EmployeeId,Fname,Lname,EmpPassword

Claim

The claim table defines claim tables consisting of claim claim Id,Status,Amount and treatment .Claims are sent by the doctor to the insurance provider of the member who seeks treatment.

Field	Data Type	Description	PK/FK
ClaimId	integer	This field defines the unique Id of each claim	PK
MemberId	integer	This field defines the Id of the member who upon which the claim has to be processed	FK
DoctorId	integer	This field defines the Id of the doctor who issues claim	FK
ClaimStatus	varchar(20)	Defines the status of the claim(Pending,approved,Denied)	
BillAmount	decimal(10,2)	Defines the amount billed for the service/Treatment	
Treatment	varchar(50)	Defines Name of the treatment	
ClaimDate	Date	Records the claim date	

FDs: ClaimId ->ClaimId,MemberId,DoctorID,ClaimStatus,BillAmount,Treatment,ClaimDate

MemberRecord

This table defines the current details of the member plan consisting of member's deductible amount reached,out of pocket amount reached,member's monthly premium status and start date of the plan.

Field	Data Type	Description	PK/FK
-------	-----------	-------------	-------

RecordNo	integer	Defines the record number of the members	PK
MemberId	integer	Defines the memberId	
PlanId	integer	Defined the plan Id on which the member is enrolled	FK
DeductibleSpent	decimal(10,2)	Records the deductible amount reached by the member	
OutOfPocketSpent	decimal(10,2)	Records the out of pocket amount reached by the member	
StartDate	Date	Defines the start date of enrollment to the plan	
PremiumPaymentStatus	Enum	Contains the monthly premium status(paid and unpaid)	

Fds:

RecordNo,MemberID->PlanId,DeductibleSpent,OutOfPocketLimit,StartDate,PremiumPaymentStatus

Statement

This table defines the statement to be given to the member consisting of information on the amount covered by insurance for the particular claim associated with member.

Field	Data Type	Description	PK/FK
StatementNo	Integer	Defines the unique number for each statement	PK
MemberId	Integer	Defines the unique Id of the member	FK

EmployeeId	Integer	Defines the unique Id of each employee	FK
AmountCovered	Decimal(10,2)	Records the amount covered from insurance provider for a particular claim	

Fds:StatementNo ->MemberId,EmployeeId,AmountCovered

IsSent

This table defines the unique combination of claim Id with corresponding employee Id

Field	Data Type	Description	PK/FK
ClaimId	integer	Contains the unique Id for each claim	PK,FK
EmpId	integer	Contains the unique employee Id for each employee	

Visits: This table defines the visit date of member with unique Id to the doctor with unique Doctor Id.

Field	Data Type	Description	PK/FK
MemberId	integer	Defines the unique Id for each member	PK,FK
DoctorId	integer	This field defines the unique Id of each doctor	
VisitDate	Date	Records the date of visit by to member to the doctor for a service/treatment	

FDs: -> MemberId,DoctorId -> VisitDate

Has: This table defines the unique statement number for each claim Id

Field	Data Type	Description	PK/FK
ClaimId	integer	Contains the unique Id for each claim	PK,FK
StatementNo	integer	Defines the unique number for each statement	

Updates

This table defines the member Id's with corresponding employee Id's.

Field	Data Type	Description	PK/FK
MemberId	integer	Defines the memberId	PK,FK
RecordNO	integer	Defines the record number of the members	
EmpId	integer	Contains the unique employee Id for each employee	

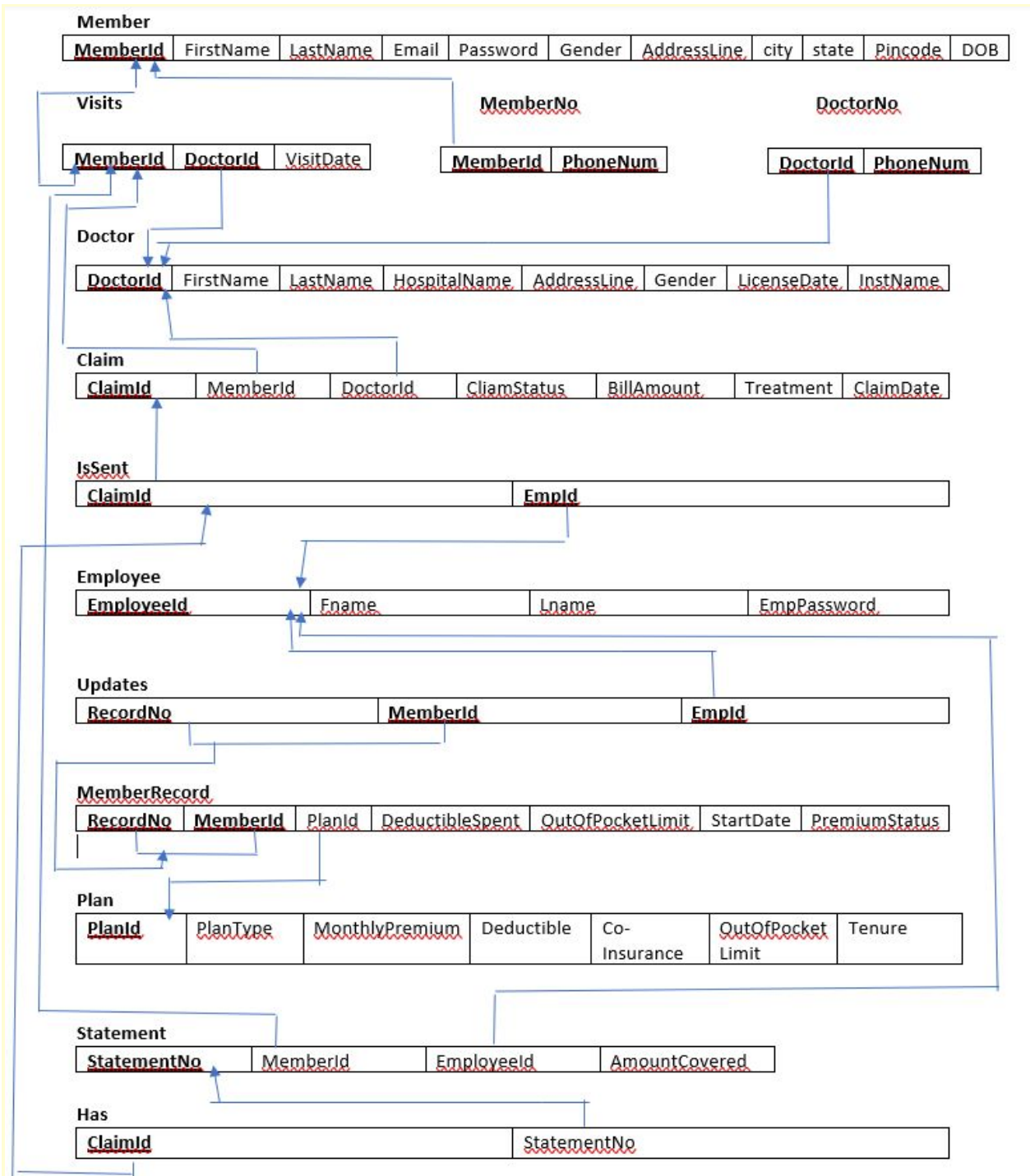
DoctorNo: This table defines the contact details for each member

Field	Data Type	Description	PK/FK
MemebrId	integer	Defines the unique Id for each member	PK,FK
PhoneNumber	integer	Contains the Contact Numbers of the doctor	

MemeberNo : This table defines the contact details of each doctor

Field	Data Type	Description	PK/FK
DoctorId	integer	This field defines the unique Id of each doctor	PK,FK
PhoneNumber	integer	Contains the Contact Numbers of the member	

Schema Diagram:



List of Tables and Views:

The screenshot shows a database management tool interface. On the left, a tree view displays the 'InsuranceCompany' database structure, including tables (claim, doctor, doctorNo, employee, has, issent, memberNo, memberRecord, members, plan, statement, updates, visits) and views (emp_claims_view). The main pane on the right displays a SQL script for creating a procedure named 'UpdateMemberRecord'.

```
1 • DROP PROCEDURE IF EXISTS UpdateMemberRecord;
2 DELIMITER $$
3
4 • CREATE PROCEDURE UpdateMemberRecord(IN var_claimID integer )
5 BEGIN
6
7     DECLARE var_empid INT;
8     DECLARE var_recordno INT;
9     DECLARE var_bill DEC(10,2) DEFAULT 0.0;
10    DECLARE var_memberid INT;
11    DECLARE var_oopSpent DEC(10,2) DEFAULT 0.0;
12    DECLARE var_deductibleSpent DEC(10,2) DEFAULT 0.0;
13    DECLARE var_approvedAmount DEC(10,2) DEFAULT 0.0;
14    DECLARE var_balanceOop DEC(10,2) DEFAULT 0.0;
15    DECLARE var_coInsurance DEC(10,2) DEFAULT 0.0;
16    DECLARE var_balanceDeductible DEC(10,2) DEFAULT 0.0;
17    DECLARE var_oopPlan DEC(10,2) DEFAULT 0.0;
18    DECLARE var_deductiblePlan DEC(10,2) DEFAULT 0.0;
19    DECLARE var_coInsurancePlan INT DEFAULT 0;
20    DECLARE var_tobePaidByCust DEC(10,2) DEFAULT 0.0;
```

SQL QUERIES:

Member Queries :

1. **Member Profile Query** : It fetches profile information such as email,address of a member.

The screenshot shows a database query result for the 'Member Profile Query'. The query is: `select email,addressline,city,state,pincode from members m where m.memberid = 101;`. The result is displayed in a table with 6 columns: email, addressline, city, state, pincode, and an empty column. The first row shows the data for member ID 101.

email	addressline	city	state	pincode	
isaac.m@email.com	109 Mammoth Street	San Jose	CA	95002	

2. **Current plan Query** : It fetches current plan details such as plan start date , end date, OOP Spent, Deductible spent, premium payment status.

```

1  select r.memberid, p.plantype, r.deductiblespent,r.outofpocketspent,r.startDate,
2  DATE_ADD(startDate, INTERVAL p.tenure MONTH) as end_date,r.premiumPaymentStatus
3  from memberrecord r, plan p
4  where p.planid = r.planid and r.memberid = 101;
5

```

100% 1:6

Result Grid Filter Rows: Search Export:

memberid	plantype	deductiblespent	outofpocketspent	startDate	end_date	premiumPaymentStatus
101	Bronze	550.00	550.00	2019-02-01	2019-08-01	Paid

3. Plans query : It fetches all the plans that are available for members.

```

1 • select planid,plantype,monthlypremium, deductible,coinsurance,outofpocketlimit,tenure from plan;

```

100% 97:1

Result Grid Filter Rows: Search Export:

planid	plantype	monthlypremium	deductible	coinsurance	outofpocketlimit	tenure
111	Bronze	150.00	7500.00	25.00	10000.00	6
222	Silver	200.00	6500.00	25.00	8600.00	12
333	Gold	250.00	5000.00	20.00	7000.00	12
444	Platinum	350.00	4000.00	15.00	6000.00	12
445	HSA Silver	100.00	2000.00	3.00	5679.00	12

4. Available Doctors query : It fetches list of doctors available along with their details.

```

1 • select doctorid, fname, lname, hospital, licensedate,institution,address, gender from doctor;
2
3
4

```

100% 94:1

Result Grid Filter Rows: Search Export:

doctorid	fname	lname	hospital	licensedate	institution	address	gender
142	Phoenix	Diaz	FlowerHill General Hospital	1994-01-05	Calgary Medical Institute	846 Oakwood Ave.	M
203	Colin	Benson	Riverside Clinic	2005-06-17	United Health Academy	281 Evergreen St.	M
405	Karissa	Patrick	FlowerHill General Hospital	2006-11-13	United Health Academy	281 Evergreen St.	F
416	Marc	Jenson	FlowerHill General Hospital	2013-01-08	National Institute for Medicine	732 Manor Ave.	M
591	Trace	Giles	Riverside Clinic	2015-10-20	John Hopkins University	8279 Bow Ridge St.	F
608	Alisa	Townsend	Mercy Vale Hospital	2001-05-14	John Hopkins University	8279 Bow Ridge St.	F

5.Statements Query: It fetches both approved and denied statements related to member

```

1 • select s.statementno, c.claimid, c.claimamount, c.claimstatus,c.claimdate, c.treatment, s.amou
2 from claim c, statement s ,has p
3 where c.claimid = p.claimid and s.statementno= p. statementno and s.memberid =101;

```

100% 84:3

Result Grid Filter Rows: Search Export:

statementno	claimid	claimamount	claimstatus	claimdate	treatment	amountcovered
1000	150	6000.00	Approved	2019-11-22	Bone Fracture	6000
1003	151	150.00	Approved	2019-10-13	Normal visit	150
1004	152	5000.00	Denied	2016-05-13	Scan	5000
1005	153	550.00	Approved	2019-08-13	Gynaecology	550

Employee Queries:

6.Members Fetch Query : Lists down members along with their details that are assigned to an employee.

```

1 • select m.memberid, p.plantype, m.fname, m.lname, r.deductiblespent,r.outofpocketspent
2 from employee e, memberrecord r, members m, updates u, plan p
3 where e.empid = u.empid and r.recordno = u.recordno
4 and r.memberid = m.memberid
5 and r.planid = p.planid
6 and e.empid = 4 order by m.memberid;
7
8
9
10

```

100% 24:9

Result Grid Filter Rows: Search Export:

memberid	plantype	fname	lname	deductiblespent	outofpocketspent
116	Gold	Rory	Jones	3000.00	3000.00
117	Gold	Chloe	Levine	3000.00	3000.00
118	Platinum	Armani	Haas	3900.00	3900.00
119	Bronze	Van	Robertson	7500.00	7500.00
120	Gold	Khalil	Ware	3000.00	3000.00

7.Claims List Query : It fetches the claim details of the members assigned to the employee.

```

1 • select claimid, memberid, empid, claimstatus, claimamount,paymentstatus from emp_claims_view
2 where empid = 1 order by claimid;

```

00% 39:2

Result Grid Filter Rows: Search Export:

claimid	memberid	empid	claimstatus	claimamount	paymentstatus
150	101	1	Approved	6000.00	Paid
151	101	1	Approved	150.00	Paid
152	101	1	Denied	5000.00	Paid
153	101	1	Approved	550.00	Paid
154	102	1	Pending	6500.00	Paid
155	103	1	Pending	550.00	UnPaid
156	104	1	Pending	150.00	Paid
157	105	1	Pending	3000.00	Paid
158	101	1	Pending	250.00	Paid
502	101	1	approved	9876.00	Paid

emp_claims_view 5

8. Member record update for approved claim : It is a stored procedure which runs whenever a claim is approved. It generates a statement with claim details and approved amount. It also updates member record with new values for deductible spent and out of pocket spent.

Transaction has been implemented as part of the stored procedure to maintain ACID properties. If one of the update/insert statements fail in the transaction then database roll back to previous state. Fig.1 illustrates the transaction performed during the member record update.

```
START TRANSACTION;
UPDATE memberrecord SET deductibleSpent = var_deductibleSpent,outOfPocketSpent= var_oopSpent WHERE memberId = var_memberid;
UPDATE claim SET claimStatus = 'approved' WHERE claimid = var_claimID;
INSERT INTO statement(memberid,empid,amountCovered) values (var_memberid, var_empid, var_approvedAmount);
SET out_param = LAST_INSERT_ID();
INSERT INTO has(claimid,statementNo) values (var_claimID,out_param);

IF var_rollback THEN
    ROLLBACK;
ELSE
    COMMIT;
END IF;
```

Fig.1 Transaction during member record update.

9. Denial Claim : Stored procedure performed when claim gets denied.It generates a statement with claim details and claim status. Transaction has been implemented as part of stored procedure such that if any of the query fails, database roll backs to the earlier state. Fig.2. Displays stored procedure for denial of the

```

CREATE PROCEDURE claimDenial(IN var_claimID integer )
BEGIN

    DECLARE var_rollback BOOL DEFAULT 0;
    DECLARE var_approvedAmount DEC(10,2) DEFAULT 0.0;
    DECLARE var_bill DEC(10,2) DEFAULT 0.0;
    DECLARE out_param INT;
    DECLARE var_memberid INT;
    DECLARE var_empid INT;
    DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET var_rollback = 1;

    SELECT claimamount,memberid INTO var_bill,var_memberid FROM claim where claimid=var_claimID;

    SELECT u.empid INTO var_empid FROM memberrecord mr, updates u
    where u.recordno = mr.recordno and u.memberid = mr.memberid and mr.memberID=var_memberid;

    START TRANSACTION;
    UPDATE claim SET claimStatus = 'denied' WHERE claimid = var_claimID;
    SET var_approvedAmount = 0.0;
    INSERT INTO statement(memberid,empid,amountCovered) values (var_memberid, var_empid, var_approvedAmount);
    SET out_param = LAST_INSERT_ID();
    INSERT INTO has(claimid,statementNo) values (var_claimID,out_param);

    IF var_rollback THEN
        ROLLBACK;
    ELSE
        COMMIT;
    END IF;
    SELECT var_claimID,var_bill,var_memberid,out_param,var_rollback;

```

10.

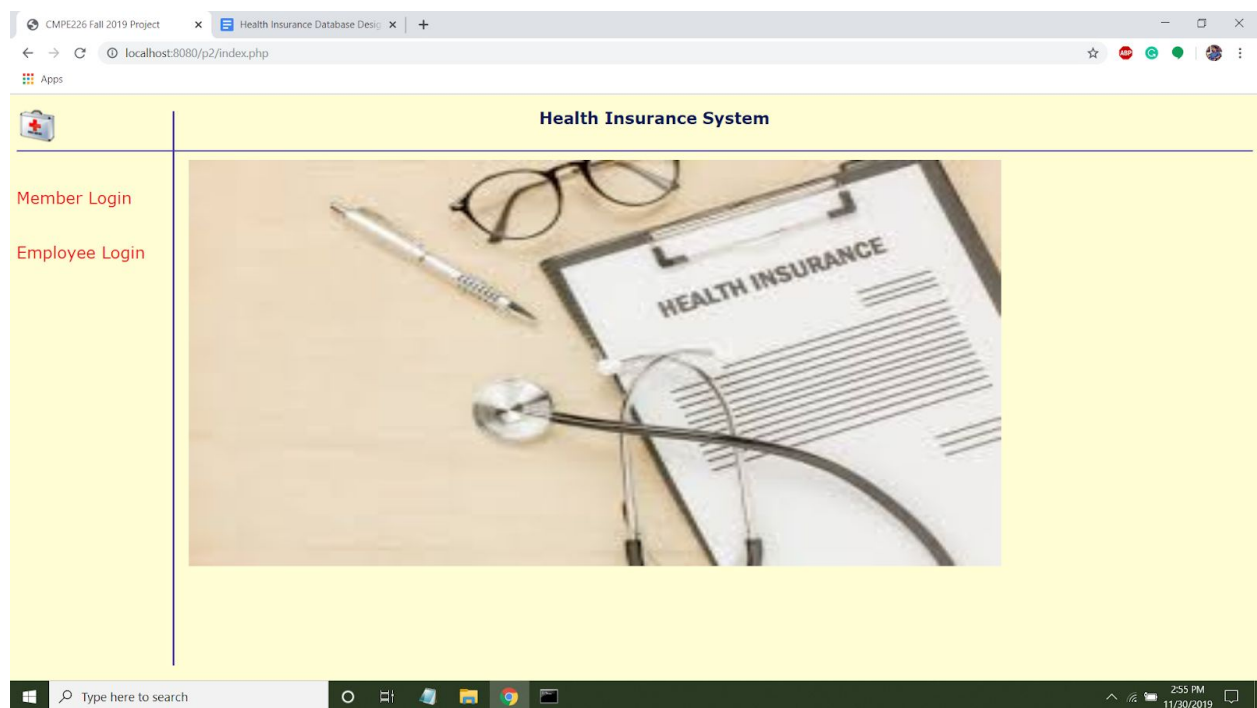
Application Execution

Starting the localhost server using Wamp Server

```
Command Prompt - php -S localhost:8080
Microsoft Windows [Version 10.0.18362.476]
(c) 2019 Microsoft Corporation. All rights reserved.

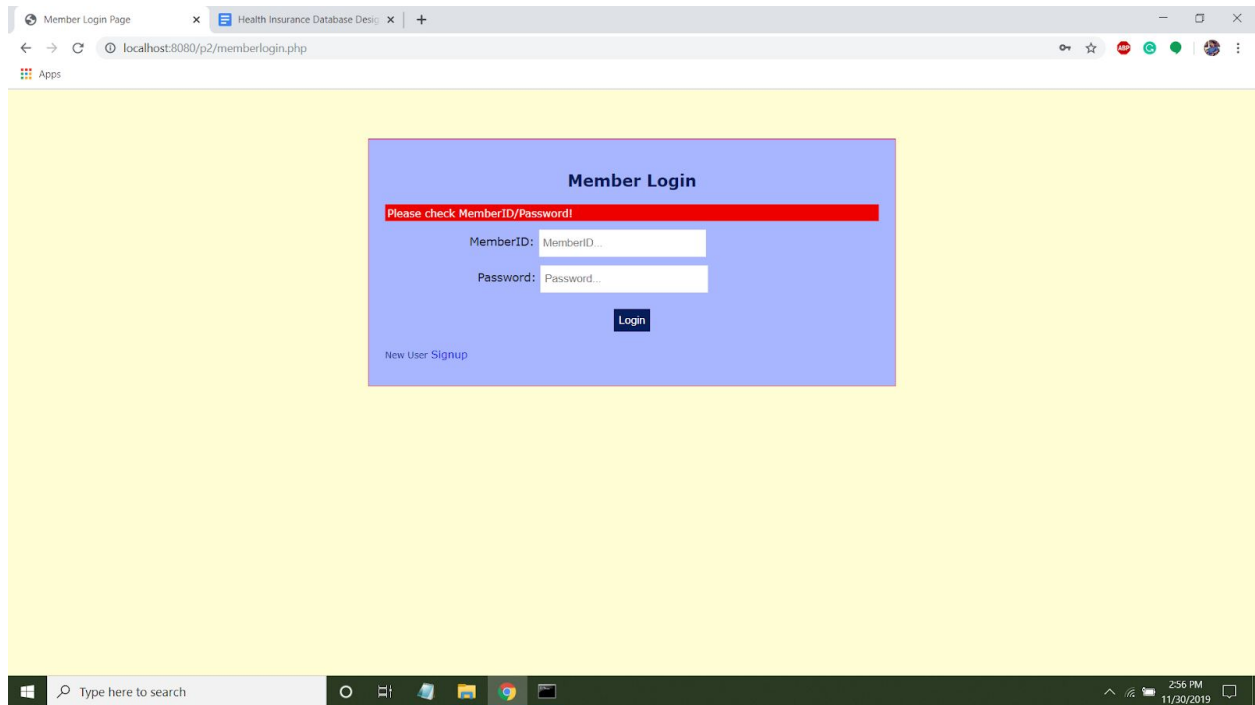
C:\Users\Varun>php -S localhost:8080
PHP 7.3.5 Development Server started at Sat Nov 30 14:54:08 2019
Listening on http://localhost:8080
Document root is C:\Users\Varun
Press Ctrl-C to quit.
[Sat Nov 30 14:54:08 2019] [::1]:59506 [200]: /p2/index.php
[Sat Nov 30 14:54:08 2019] [::1]:59505 [200]: /p2/css/style.css
[Sat Nov 30 14:54:08 2019] [::1]:59507 [200]: /p2/img/logo.png
[Sat Nov 30 14:54:08 2019] [::1]:59508 [200]: /p2/img/images.jpg
```

First Page of the Application (Member and Emp Login)



Login in as Member 104

Logging the Login Error in a text file




```
error - Notepad
File Edit Format View Help
December 1, 2019, 1:16:46:- MemberID 101 logged in
December 1, 2019, 1:19 pm*****Error Log closed*****
December 1, 2019, 1:19 pm*****Error Log opened*****
December 1, 2019, 1:20:25:- EmployeeID 1 logged in
December 1, 2019, 1:21 pm*****Error Log closed*****
December 1, 2019, 1:21 pm*****Error Log opened*****
December 1, 2019, 1:30 pm*****Error Log opened*****
December 1, 2019, 9:30:47:- Error connecting to database!
December 1, 2019, 1:31:13:- MemberID 101 logged in
December 1, 2019, 1:38 pm*****Error Log closed*****
December 1, 2019, 1:38 pm*****Error Log opened*****
December 1, 2019, 1:39:06:- EmployeeID 1 logged in
December 1, 2019, 2:03 pm*****Error Log opened*****
December 1, 2019, 2:03:14:- Please check MemberID/Password for 102!
December 1, 2019, 2:03:35:- Please check MemberID/Password for 102!
December 1, 2019, 2:03:47:- MemberID 102 logged in
December 1, 2019, 2:05:11:- EmployeeID 1 logged in
December 1, 2019, 2:05 pm*****Error Log opened*****
December 1, 2019, 2:05:37:- MemberID 102 logged in
December 1, 2019, 2:58:46:- EmployeeID 1 logged in
December 1, 2019, 3:02:27:- EmployeeID 1 logged in
December 1, 2019, 3:25 pm*****Error Log opened*****
December 1, 2019, 11:25:09:- Error connecting to database!
December 1, 2019, 11:25:53:- Error connecting to database!
December 1, 2019, 11:26:00:- Error connecting to database!
December 1, 2019, 3:26:22:- MemberID 101 logged in
December 1, 2019, 3:26 pm*****Error Log closed*****
December 1, 2019, 3:26 pm*****Error Log opened*****
December 1, 2019, 3:26:40:- EmployeeID 1 logged in
December 1, 2019, 3:26 pm*****Error Log closed*****
December 1, 2019, 3:26 pm*****Error Log opened*****
December 1, 2019, 3:27:03:- MemberID 101 logged in
December 1, 2019, 3:27 pm*****Error Log closed*****
December 1, 2019, 3:27 pm*****Error Log opened*****
December 1, 2019, 3:27:51:- MemberID 102 logged in
December 1, 2019, 3:28 pm*****Error Log closed*****
December 1, 2019, 3:28 pm*****Error Log opened*****
December 1, 2019, 3:28:41:- EmployeeID 1 logged in
December 1, 2019, 3:29 pm*****Error Log closed*****
December 1, 2019, 3:29 pm*****Error Log opened*****
December 1, 2019, 3:29:14:- MemberID 102 logged in
December 1, 2019, 3:44 pm*****Error Log opened*****
December 1, 2019, 3:45:00:- Please check MemberID/Password for 104!
```

Successful login of Member 104 (Kareem)

CMPE226 Fall 2019 Project

localhost:8080/p2/memberinfo.php

Apps

 **Health Insurance System**

Hi Kareem Salazar [Logout](#)

MY PROFILE

Email	AddressLine	City	State	ZipCode
kareem.s@email.com	97 1st Court	San Jose	CA	95037 Edit

QUICK LINKS

[Available Plans](#)

[Available Doctors](#)

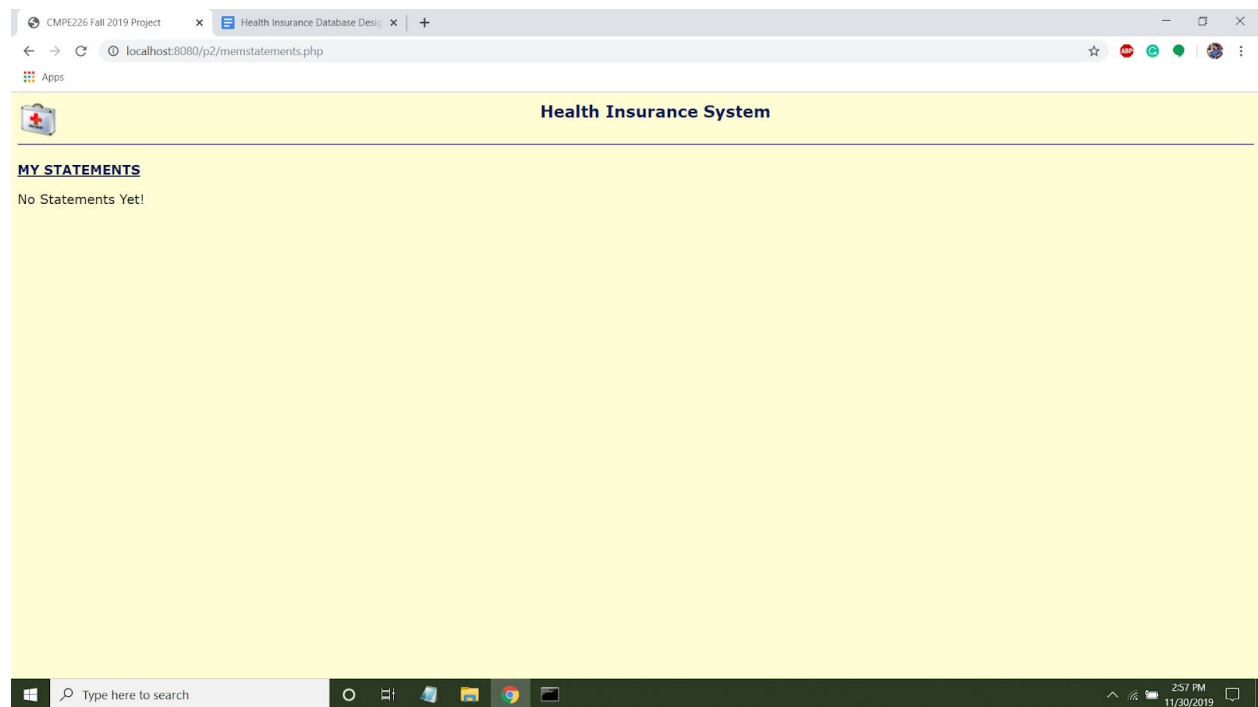
MY CURRENT PLAN

MemberID	PlanType	DeductibleSpent	OutOfPocketSpent	StartDate	EndDate	PremiumPayment
104	Bronze	400.00	400.00	2019-02-01	2019-08-01	Paid

[My Statements](#)

[Raise New Claim](#)

Currently no statements



Logging in as Employee 1 (Lauren) who manages members

CMPE226 Fall 2019 Project x +

localhost:8080/p2/employeeinfo.php

Apps

Health Insurance System

Hi Lauren Paul [Logout](#)

MY CURRENT MEMBERS

MemberID	PlanType	FirstName	LastName	DeductibleSpent	OutofPocketSpent
101	Bronze	Isaac	McMahon	6000.00	6000.00
102	Bronze	Amari	Miller	1000.00	1000.00
103	Bronze	Roberto	Medina	5000.00	5000.00
104	Bronze	Kareem	Salazar	400.00	400.00
105	Bronze	Jazmine	Kelley	300.00	300.00

MY TASKS

- [Manage Claims](#)
- [Manage Member Payments](#)
- [Manage Plans](#)
- [Manage Doctors](#)

Type here to search

3:47 PM 12/1/2019

Approving member 101 claim from pending to approved

CMPE226 Fall 2019 Project x +

localhost:8080/p2/claimslist.php

Apps

Health Insurance System

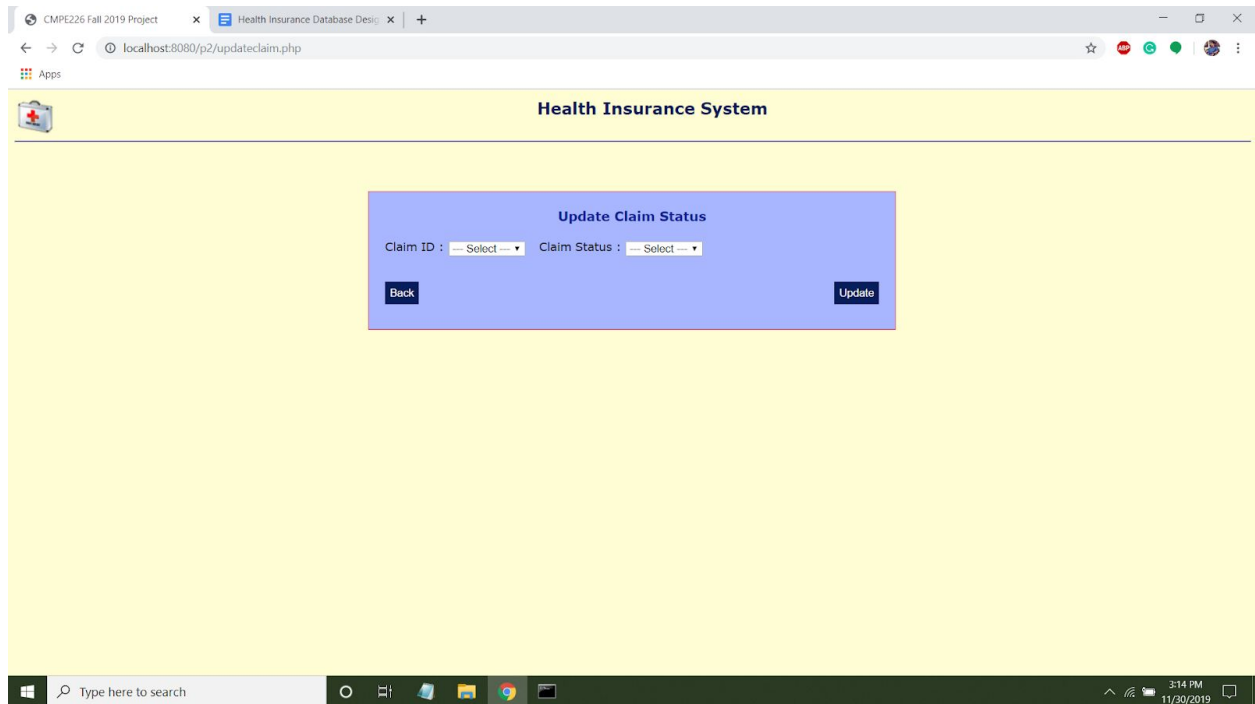
MY CLAIMS LIST

ClaimID	MemberID	ClaimStatus	ClaimAmount	PaymentStatus
150	101	Approved	6000.00	Paid
151	101	Approved	150.00	Paid
152	101	Denied	5000.00	Paid
153	101	Approved	550.00	Paid
154	102	Pending	6500.00	Paid
155	103	Pending	550.00	UnPaid
156	104	Pending	150.00	Paid
157	105	Pending	3000.00	Paid
158	101	Pending	250.00	Paid
502	104	Pending	7000.00	Paid

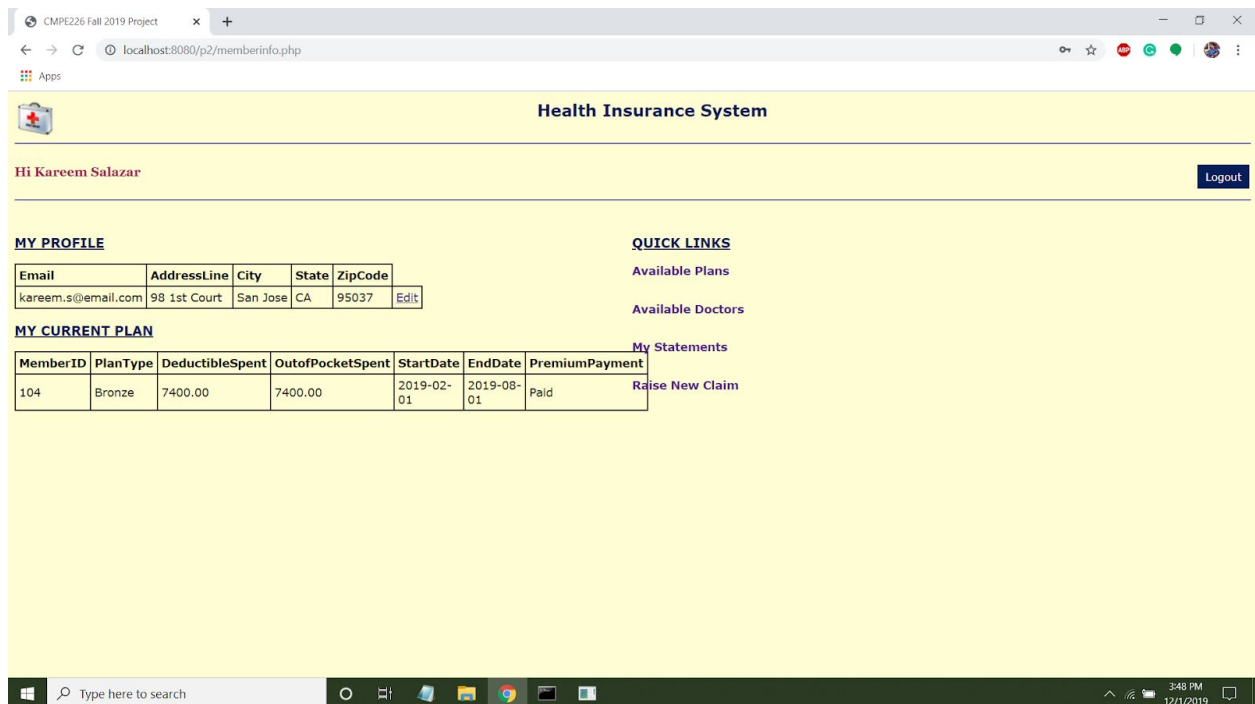
[Update Claim Status](#)

Type here to search

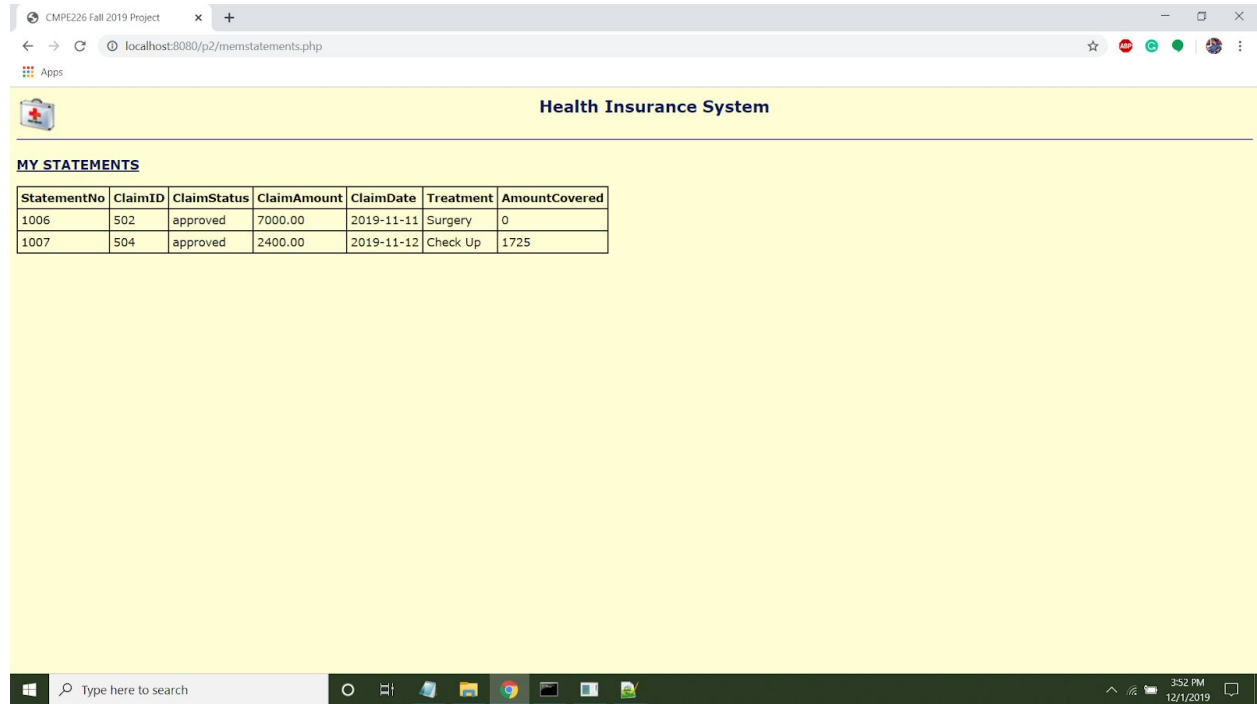
3:47 PM 12/1/2019



As you can see, once member 104 logs in again, he is able to view his updated record and a statement of how much the insurance company covered




When he raises a new claim for amount (2400), the record is updated again and he can view his statements which show that the insurance company covered \$1725.



CMPE226 Fall 2019 Project

localhost:8080/p2/memstatements.php

Apps

 **Health Insurance System**

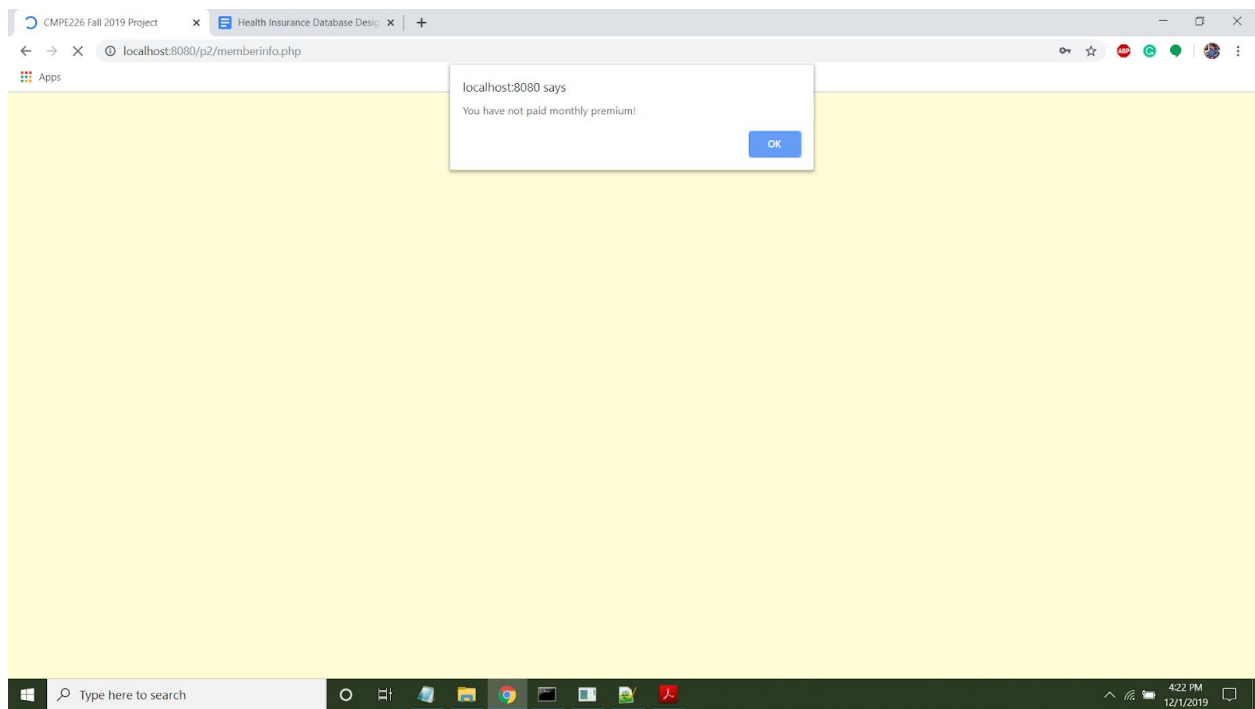
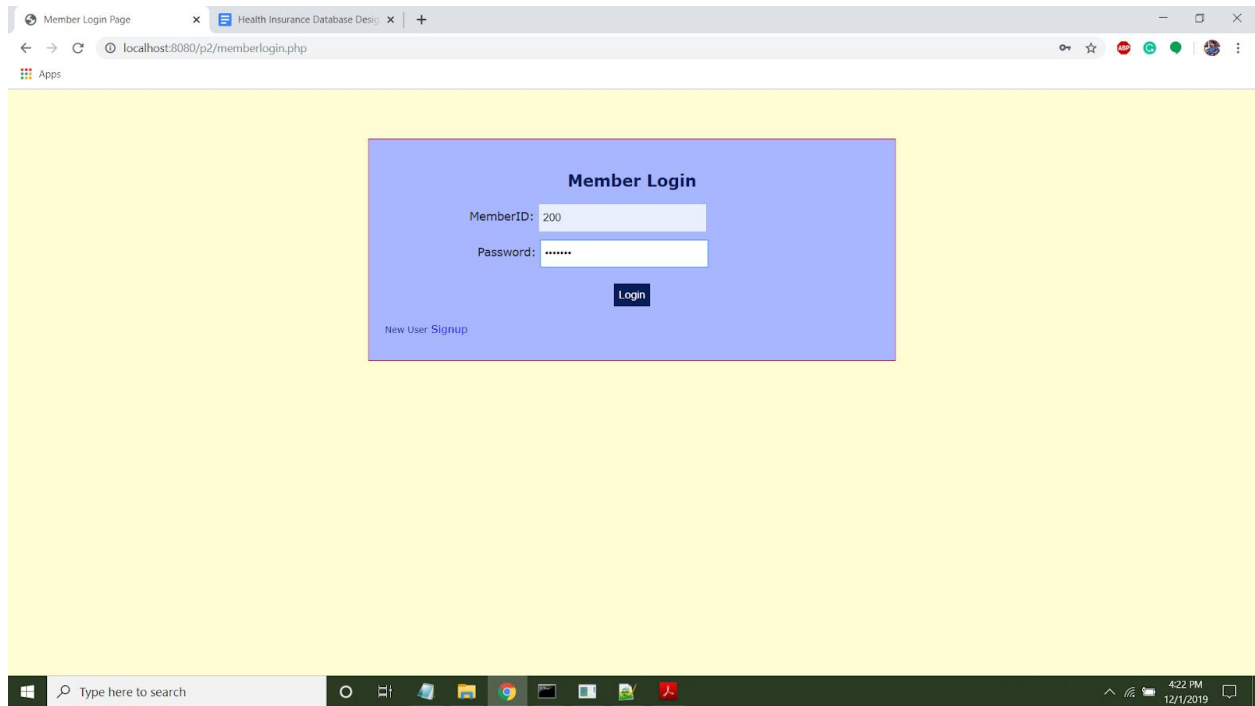
MY STATEMENTS

StatementNo	ClaimID	ClaimStatus	ClaimAmount	ClaimDate	Treatment	AmountCovered
1006	502	approved	7000.00	2019-11-11	Surgery	0
1007	504	approved	2400.00	2019-11-12	Check Up	1725

Type here to search

3:52 PM
12/1/2019

For member 200, there is a reminder that he has not paid his monthly premium upon login!




CMPE226 Fall 2019 Project

Health Insurance Database Desig

localhost:8080/p2/memberinfo.php

Apps



Health Insurance System

Hi Alex Anthony

Logout

MY PROFILE

Email	AddressLine	City	State	ZipCode
Alex@email.com	1235 E. Westwood Avenue	Sunnyvale	CA	95051

QUICK LINKS

Available Plans

Available Doctors

My Statements

Raise New Claim

MY CURRENT PLAN

MemberID	PlanType	DeductibleSpent	OutofPocketSpent	StartDate	EndDate	PremiumPayment
200	Gold	5000.00	7500.00	2019-10-24	2020-10-24	UnPaid

Type here to search



4:22 PM

12/1/2019