



Sri Lanka Institute of Information Technology

Health Care System

Programming Applications and Frameworks (IT 3030)

2020

Project ID: GroupS1151.9

Final Project Report

Submitted By:

- IT16038806 Attanayaka J.A.E.P
- IT18171198 Abeykoon M.G.T.N.K
- IT17021944 Kaluarachchi T.T.D
- IT18012620 Samarasiri V.Y
- IT18152760 Geethan Madurange Perera L

Contents

1.	Members' Details and Workload Allocation	1
2.	Clickable link to GitHub Repository	2
3.	SE Methodologies/Methods	2
4.	Gantt chart	3
5.	Requirements.....	3
5.1	Requirement analysis.....	3
5.2	Stake Holder Analysis (Onion Diagram)	4
5.3	Requirements Modelling (Use Case Diagram)	5
6.	System's overall design	6
6.1	Overall Architecture	6
6.2	Class Diagram	7
7.	Individual Section.....	8
8.	System's integration details.....	30
9.	References	30
	Appendix 01	31
	Appendix 02	33
	Appendix 03	35

1. Members' Details and Workload Allocation

IT Number	Name	Web Service	Description of the Web Service
IT16038806	Attanayaka J.A.E.P	Hospital Service	<ul style="list-style-type: none"> • Add a new hospital • Update hospitals • Delete hospital • View all hospitals
IT18171198	Abeykoon M.G.T.N.K	Patient Service	<ul style="list-style-type: none"> • Add a new patient • Update a patient • Delete a patient • View all patients
IT17021944	Kaluarachchi T.T. D	Appointment Service	<ul style="list-style-type: none"> • Add a new appointment record • Update an appointment record • Delete an appointment record • View all appointment records
IT18012620	Samarasiri V. Y	Payment Management Service	<ul style="list-style-type: none"> • Add a new payment record • Update a payment record • Delete a payment record • View all payment records
IT18152760	Geethan Madurange Perera L	Doctor Management Service	<ul style="list-style-type: none"> • Add new prescriptions and operations • Update prescriptions and operations • Delete prescriptions and operations • View all prescriptions operations

2. Clickable link to GitHub Repository

<https://github.com/vidura21/HealthCareSystemPAF2020>

3. SE Methodologies/Methods

V-Model

The V-Model is a type SDLC model where processes are executed in a sequential manner. It is an extension of waterfall mode. It is also called as verification and validation model. It is associated with a testing phase for each development stage and development phase is directly associated with testing phase. Development of the next phase starts only after the completion of previous phase. By using this model we gained lots of benefits to our project because we can correct the errors, bugs and test it before we enter the next stage. We can identify the defects in the early stages. It is better to identify the errors and bugs early as possible because when it comes to end of the project the team members have to put more effort and time to find and manage these errors. We applied this methodology to our project and gained the maximum out of it by reducing the disadvantages as much as we can.

The usage of V- Model

- More suitable for small to medium size projects
- Better to use when Software requirements are clear and well defined

Advantages of V-Model

- Simple and easy to use
- Helps the project management to track progress of the work
- Highly disciplined model

Disadvantages of V-Model

- High risk and uncertainty
- Not suitable for complex projects
- Not suitable for projects where project requirements are not well defined

4. Gantt chart

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8
Hospital Management and Crud Operations								
Doctor Management and Crud Operations								
Patient Management and Crud Operations								
Appointment Management and Crud Operations								
Payment Management and Crud Operations								
Testing								
Intergartion								

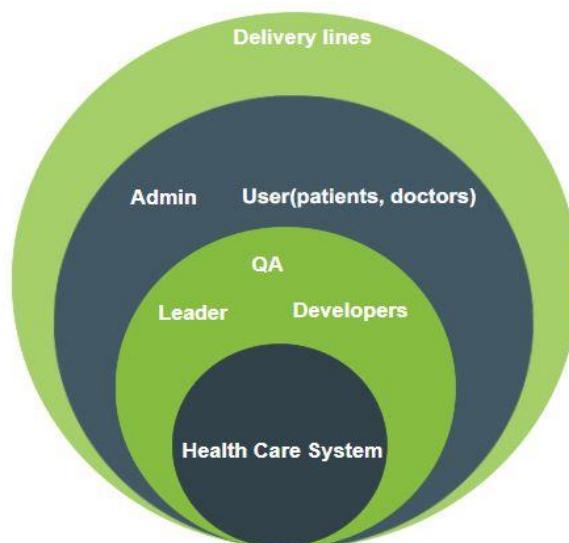
5. Requirements

5.1 Requirement analysis

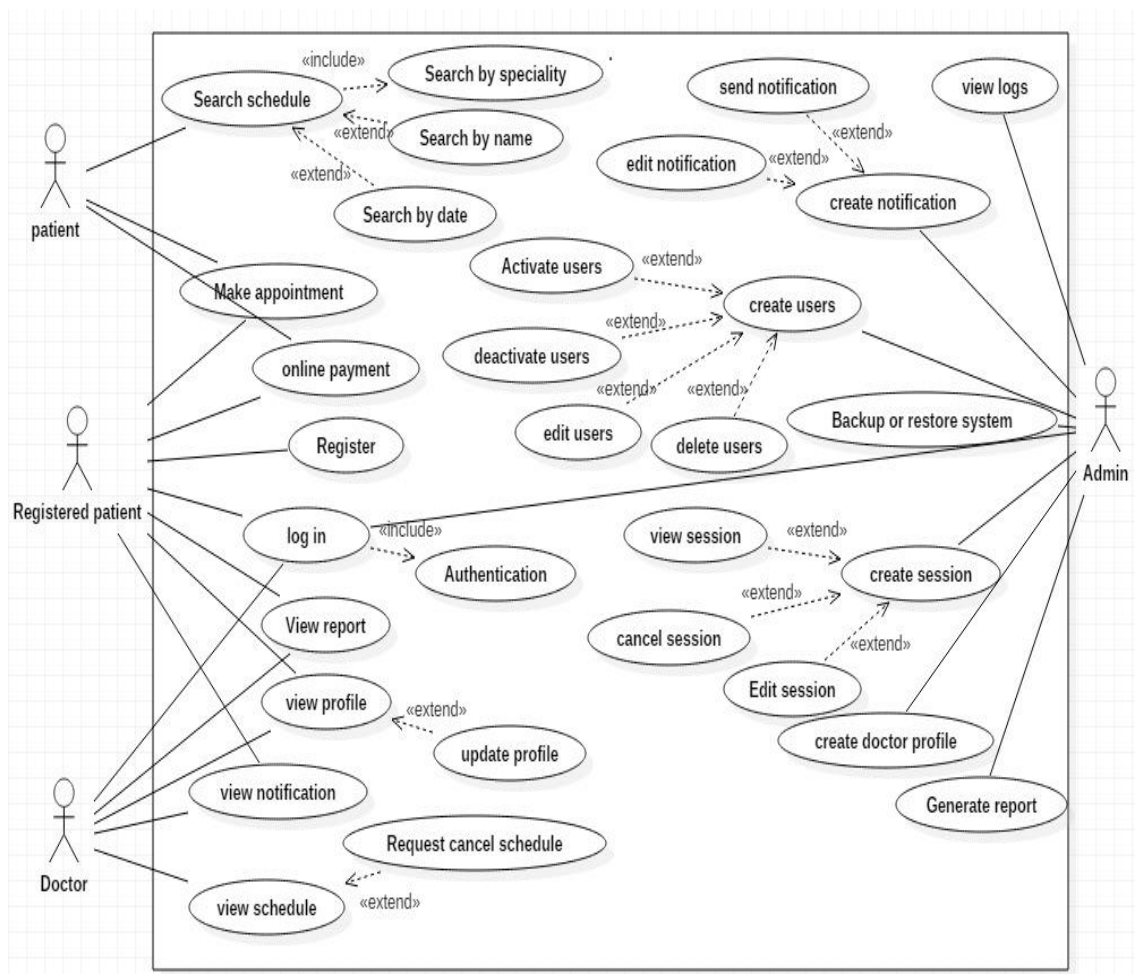
Micro-service	Functional Requirement	Non-Functional Requirement	Technical Requirement
Hospital Service	Add a new hospital Update a hospital Delete a hospital View all hospitals	Security, usability, performance	admin can add, update, delete, view hospitals
Patient Service	Add a new patient Update a patient Delete a patient View all patients	Security, usability, performance	User can enroll to the system and can update, delete, view patient details.
Appointment Service	Add a new appointment Update appointment Delete appointment View all appointment	Security, usability, performance	Receptionist can add, delete, View new appointment for patients. Patients can add a new appointment.

Doctor Service	<p>Add a new prescription and operations.</p> <p>Update a prescription and operations.</p> <p>Delete prescription and operations.</p> <p>View all prescriptions and operations.</p>	<p>performance, scalability, security, localization, usability</p>	<p>Doctors can add new prescriptions for their patients, and they can update, delete, and view all the prescriptions.</p> <p>Doctors can add new operations and can update, delete, and view all operations.</p>
Payment Service	<p>Add a new payment record</p> <p>Update a payment record</p> <p>Delete a payment record</p> <p>View the payment record</p>	<p>Security and Privacy</p> <p>Performance.</p> <p>Response in time</p> <p>Maintainability</p>	<p>User can add a new payment when checking out. User can, update, delete, view the payment details.</p>

5.2 Stake Holder Analysis (Onion Diagram)

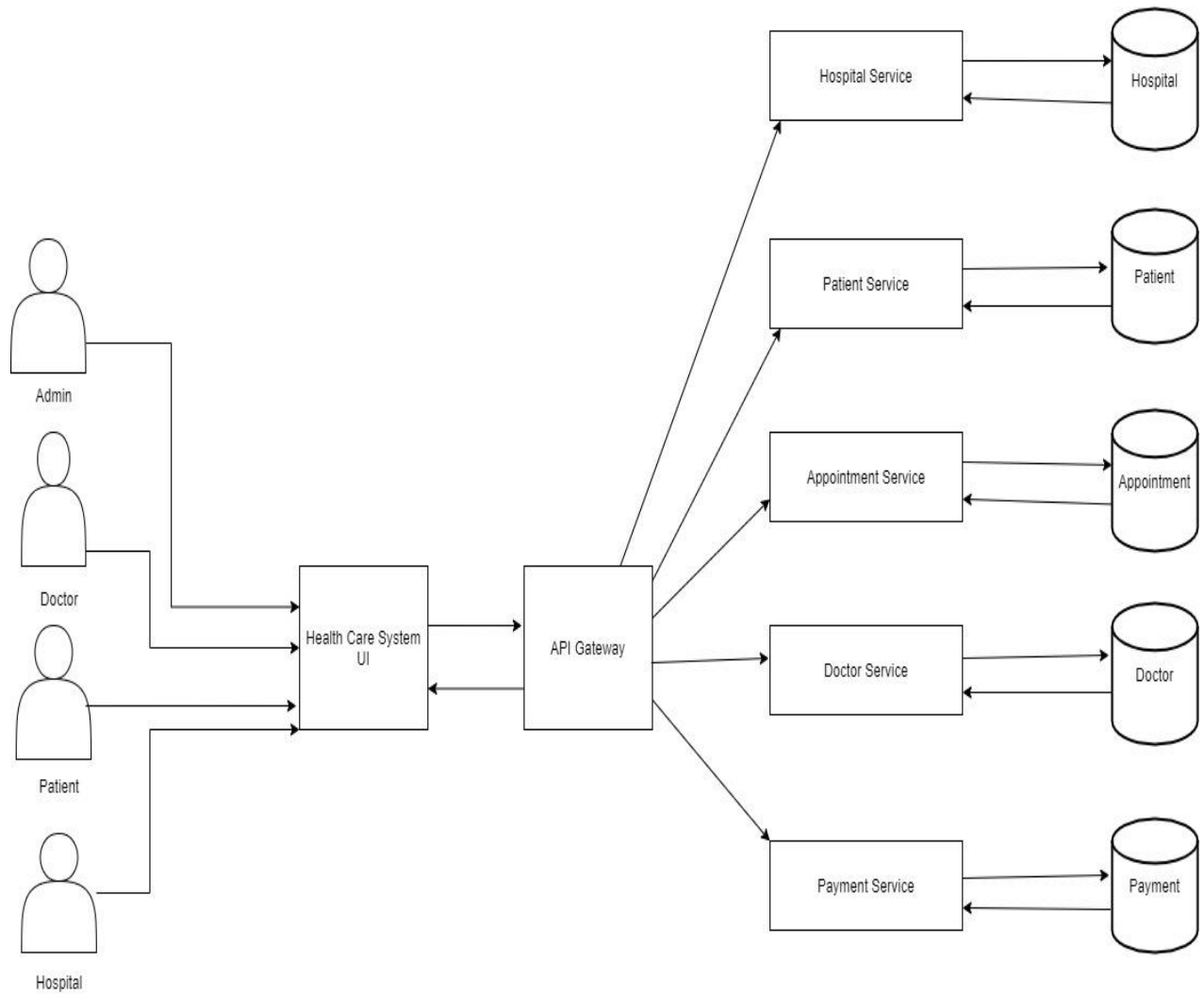


5.3 Requirements Modelling (Use Case Diagram)

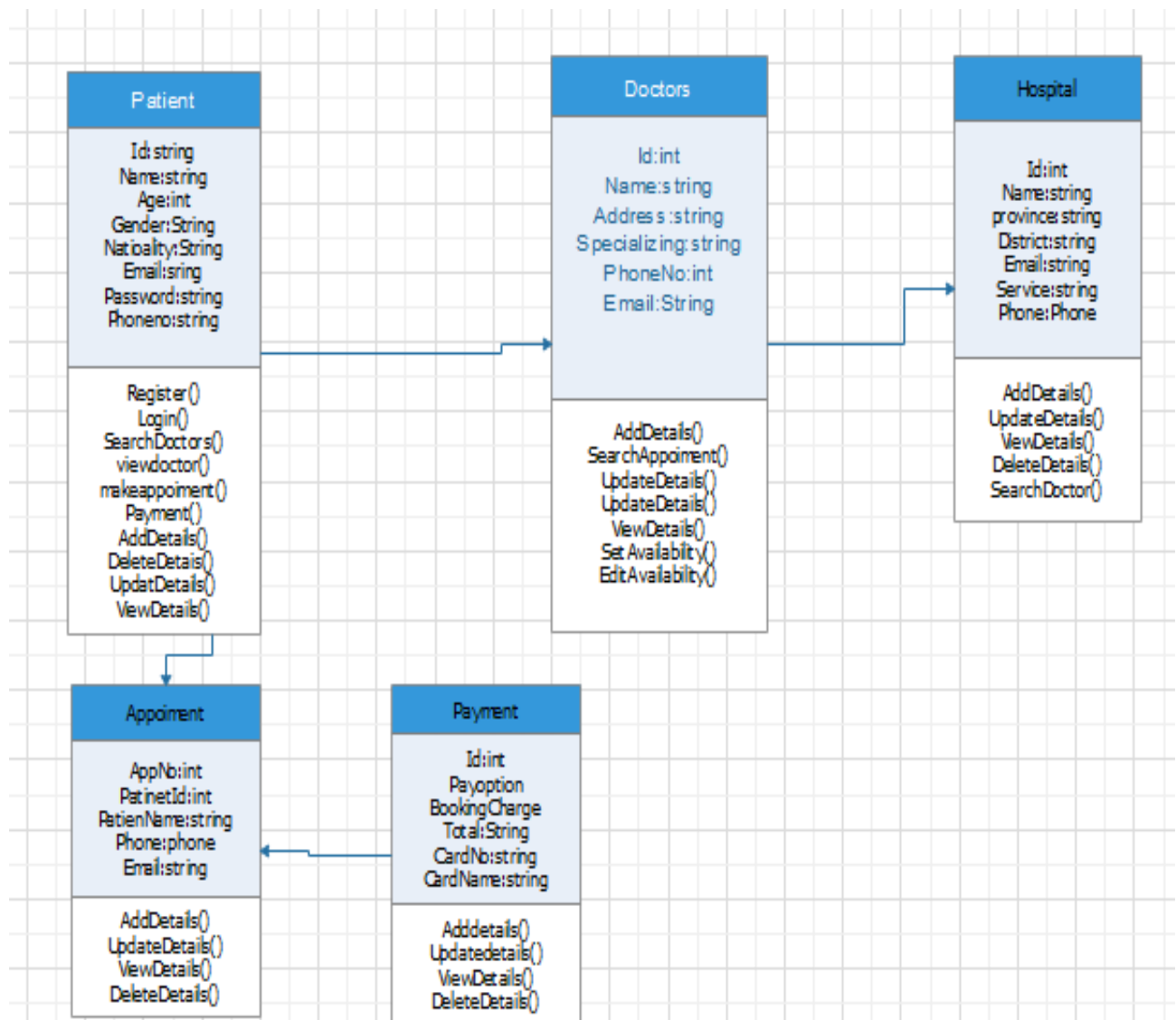


6. System's overall design

6.1 Overall Architecture



6.2 Class Diagram



7. Individual Section

IT Number: IT18012620

Name: V.Y Samarasiri

Description

Health care system is a system which can perform several tasks such as make an appointment, view hospitals and information about doctors. Making payments is an essential function for the system. The registered patients must have a way to do their payment online using the system. Our team has decided to implement the system based on micro services architecture, using RESTful web services.

API for get all the payment details in the database (GET Request)

URL:- http://localhost:8080/PAF_Project/PaymentServiceResource/Payments

Request : - {}

Response : - { paymentID = "auto generated integer value"}
 { firstname = "vidura" }
 { lastname = "yasiru" }
 { ammount = "5000.0" }
 { cardnumber = "222145667890" }
 { date = "12/4/20" }
 { cvv = "678" }

Request : - { }

Response : - { paymentID = "auto generated integer value "
 { firstname = "saman" }
 { lastname = "perera" }
 { ammount = "4000.0" }
 { cardnumber = "234511239876" }
 { date = "13/4/20" }
 { cvv = "345" }

API for insert a new payment to the database (POST Request)

URL:- http://localhost:8080/PAF_Project/PaymentServiceResource/Payments

Request :- { paymentID = "auto generated integer value " }
 { firstname = "kamal" }
 { lastname = "mendis" }
 { ammount = "7000.0" }
 { cardnumber = "567800003434" }
 { date = "3/4/20" }
 { cvv = "900" }
 Response :- { result = "Inserted successfully" }
 { paymentID = "auto generated integer value"}
 { error = "Error while inserting the payment" }

API for update a payment which is existing in database (PUT Request)

URL:- http://localhost:8080/PAF_Project/PaymentServiceResource/Payments

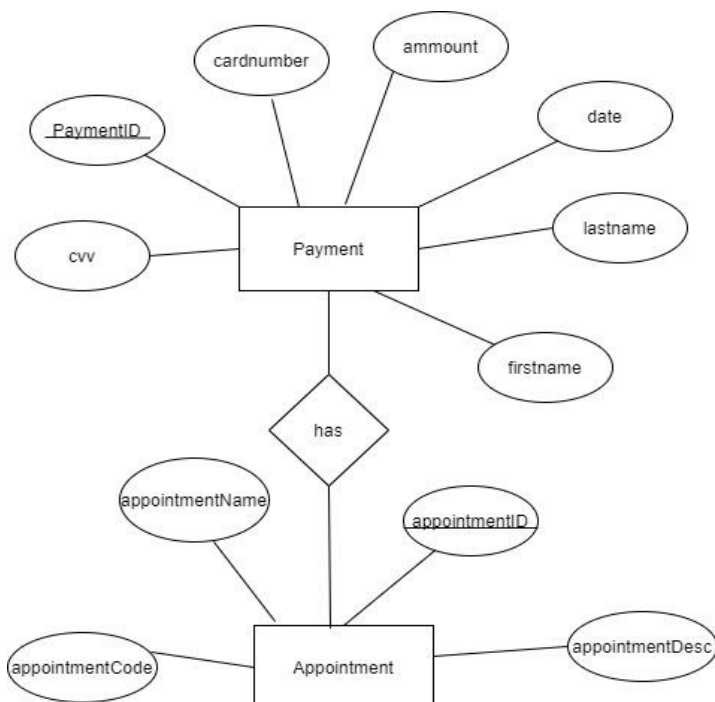
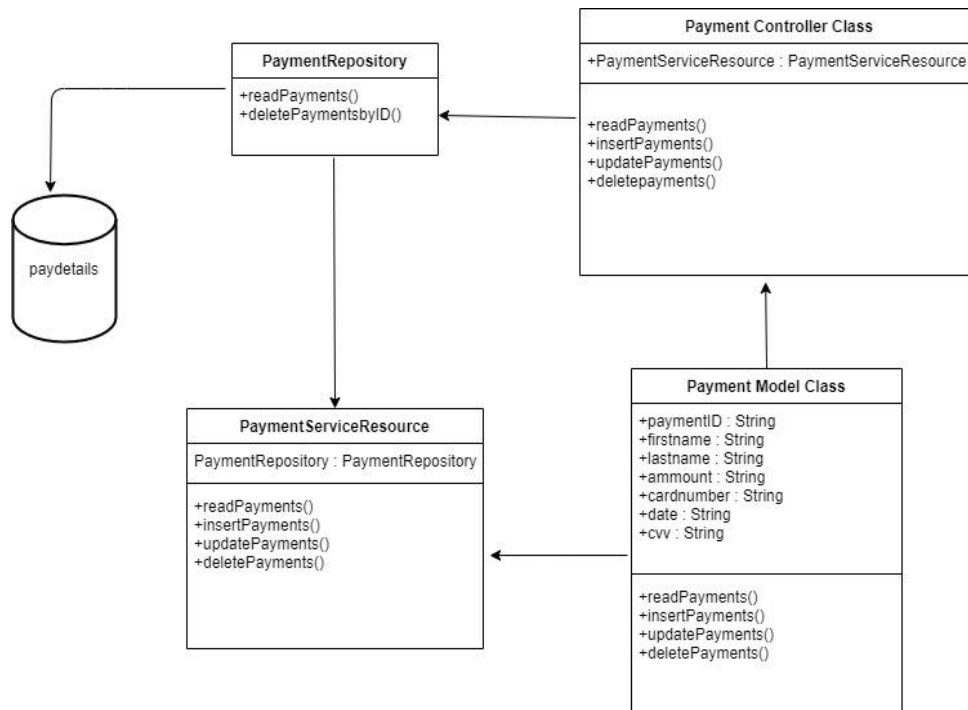
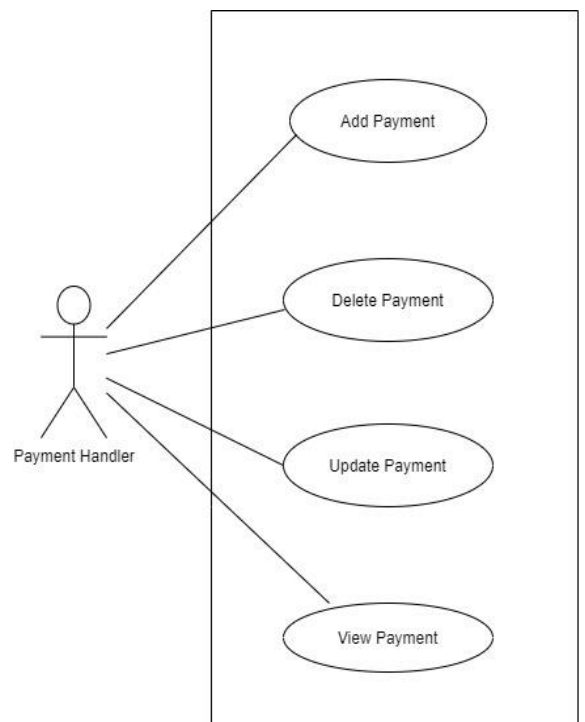
Request :- { paymentID = "auto generated integer value " }
 { firstname = "kamal" }
 { lastname = "mendis" }
 { ammount = "5500" }
 { cardnumber = "567800003434" }
 { date = "3/4/20" }
 { cvv = "900" }
 Response :- { result = "Updated successfully" }
 { error = "error while updating the payment" }

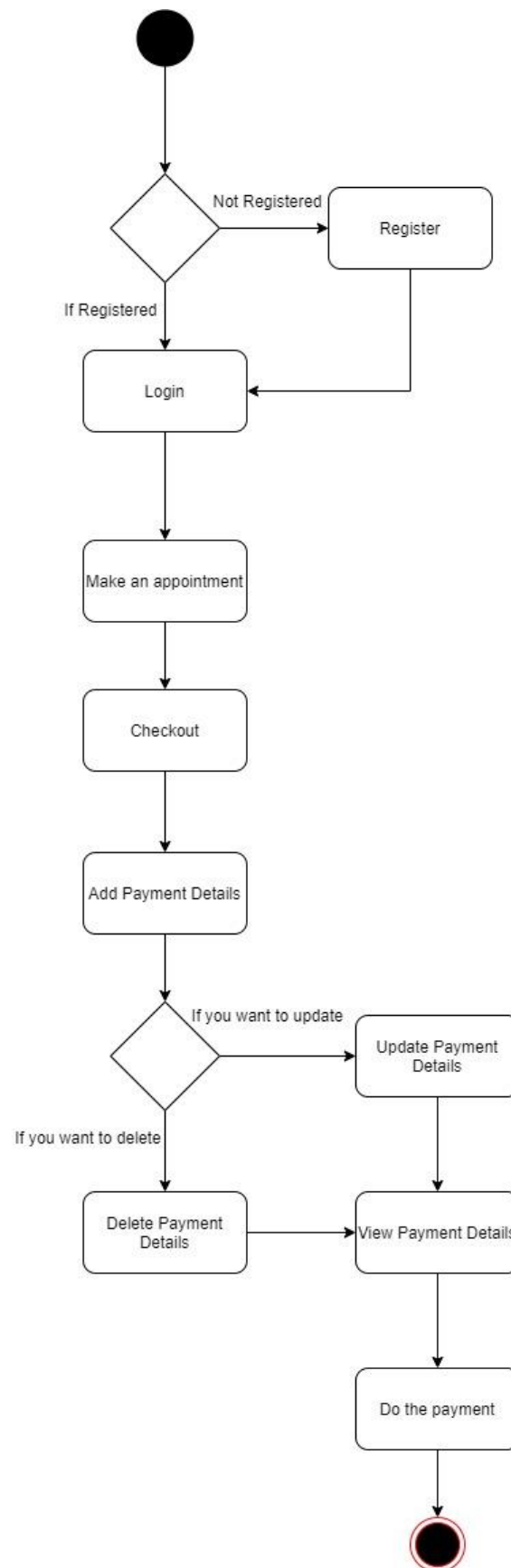
API for delete a payment which is existing in database (DELETE Request)

URL:- http://localhost:8080/PAF_Project/PaymentServiceResource/Payments

Request :- { paymentID = selected paymentID from the service }
 Response :- { result = "Deleted successfully" }
 { error = "Error while deleting the payment" }

Service Design

Class Diagram->**ER Diagram****Use case Diagram**

Activity Diagram->

IT Number: IT18152760

Name: Geethan Madurange Perera L

Description

Health care system is a system which can perform several tasks such as make an appointment, view hospitals and information about doctors, Operations, and prescriptions management. Doctor services are one of the main functions of the system, from doctor services doctor can manage Patient's Operations and Prescriptions. Our team has decided to implement the system based on micro services architecture, using RESTful web services. This is my part **Doctor services (Prescriptions management & operations management)**

**API for get all the Prescriptions & Operations details in the database
(GET Request)**

<u>Prescriptions</u>	<u>Operations</u>
URL:- http://localhost:8080/DoctorService/DoctorService/DoctorsPrescriptions	URL:- http://localhost:8080/DoctorService/DoctorService/DoctorsOperations
Request : - { } Response : - { presID= "auto generated integer value"} { presCode= "2020001" } { presName= "Saman" } { presAge= "35" } { presDate= "2020/01/05" } { presDesc= "Panadol , Paracetamol" } Request : - { } Response : - { presID= "auto generated integer value"} { presCode= "2020002" } { presName= "Kumara" } { presAge= "55" } { presDate= "2020/03/22" } { presDesc= "Vitamin C , Piriton" }	Request : - { } Response : - { operID= "auto generated integer value"} { operCode= "30300001" } { operName= "Leg Operation" } } { operDate= "2020/01/22" } { operDesc= "Inside leg finger" } } Request : - { } Response : - { operID= "auto generated integer value"} { operCode= "30300002" } { operName= "Hand Operation" } { operDate= "2020/03/13" } { operDesc= "veins operation" } }

**API for Insert the Prescriptions & Operations details to the database
(POST Request)**

<u>Prescriptions</u>	<u>Operations</u>
URL:- http://localhost:8080/DoctorService/DoctorService/DoctorsPrescriptions	URL:- http://localhost:8080/DoctorService/DoctorService/DoctorsOperations
Request :- <pre>{ presID= "auto generated integer value"} { presCode= "2020003" } { presName= "Jagath" } { presAge= "56" } { presDate= "2020/04/02" } { presDesc= "Vitamin D, Panadol" }</pre> Response :- <pre>{ result = "Inserted successfully" } { presID= "auto generated integer value"} { error = "Error while inserting" }</pre>	Request :- <pre>{ operID= "auto generated integer value"} { operCode= "30300003" } { operName= "Shoulder Operation" } { operDate= "2020/04/01" } { operDesc= "Shoulder Inside operation" }</pre> Response :- <pre>{ result = "Inserted successfully" } { presID= "auto generated integer value"} { error = "Error while inserting" }</pre>

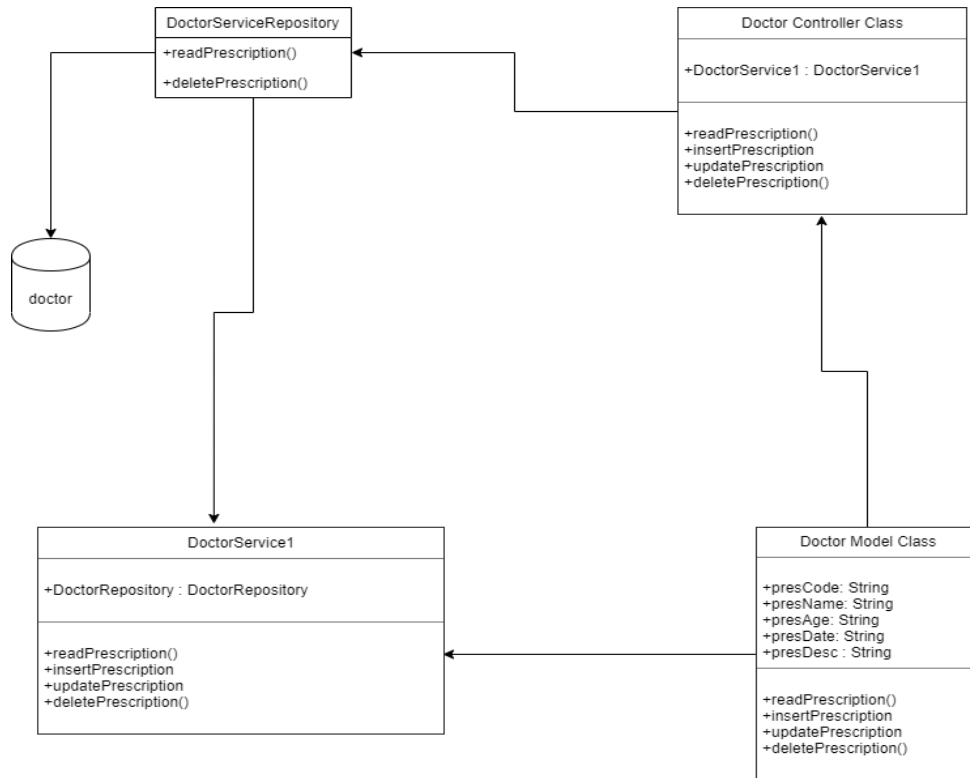
**API for Update the Prescriptions & Operations details Existing in database
(PUT Request)**

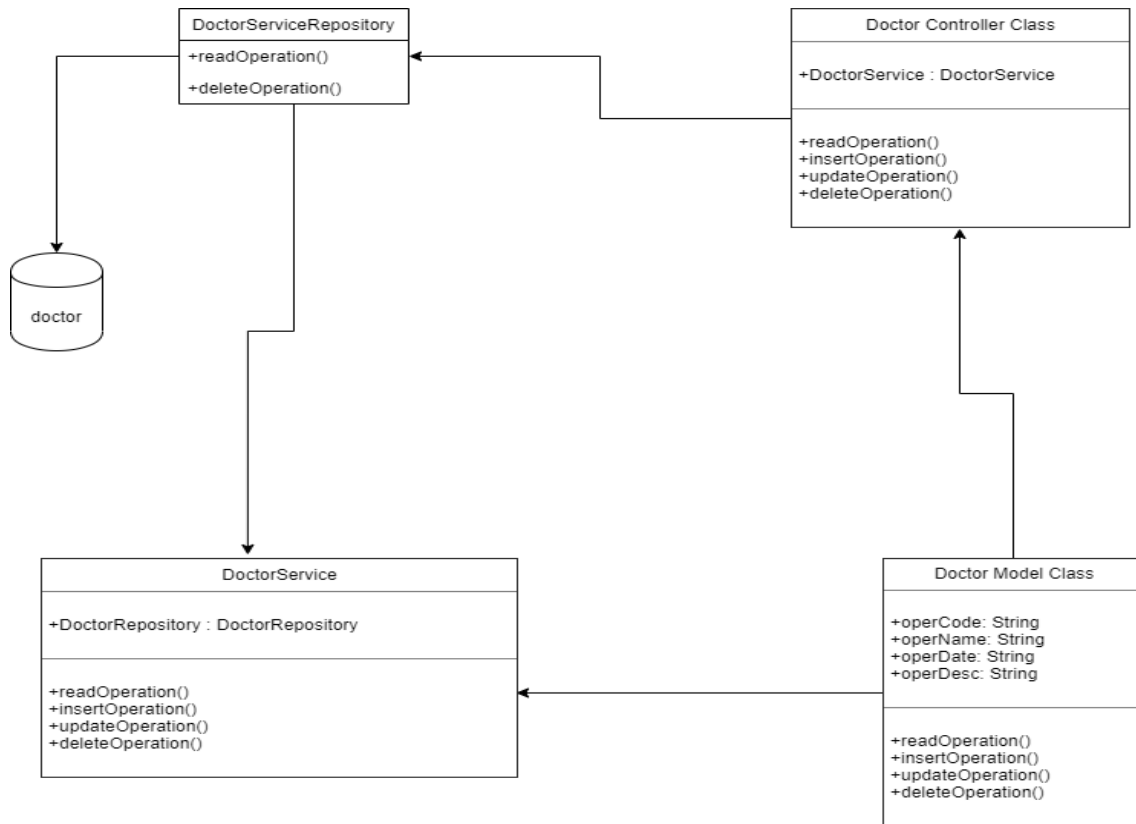
<u>Prescriptions</u>	<u>Operations</u>
URL:- http://localhost:8080/DoctorService/DoctorService/DoctorsPrescriptions	URL:- http://localhost:8080/DoctorService/DoctorService/DoctorsOperations
Request :- <pre>{ presID= "auto generated integer value"} { presCode= "2020005" } { presName= "K.L..Kumara" } { presAge= "49" } { presDate= "2020/04/08" } { presDesc= "Vitamin E, Paracetamol" }</pre>	Request :- <pre>{ operID= "auto generated integer value"} { operCode= "30300005" } { operName= "LEG operation" } { operDate= "2020/12/12" } { operDesc= "inside the leg" }</pre>

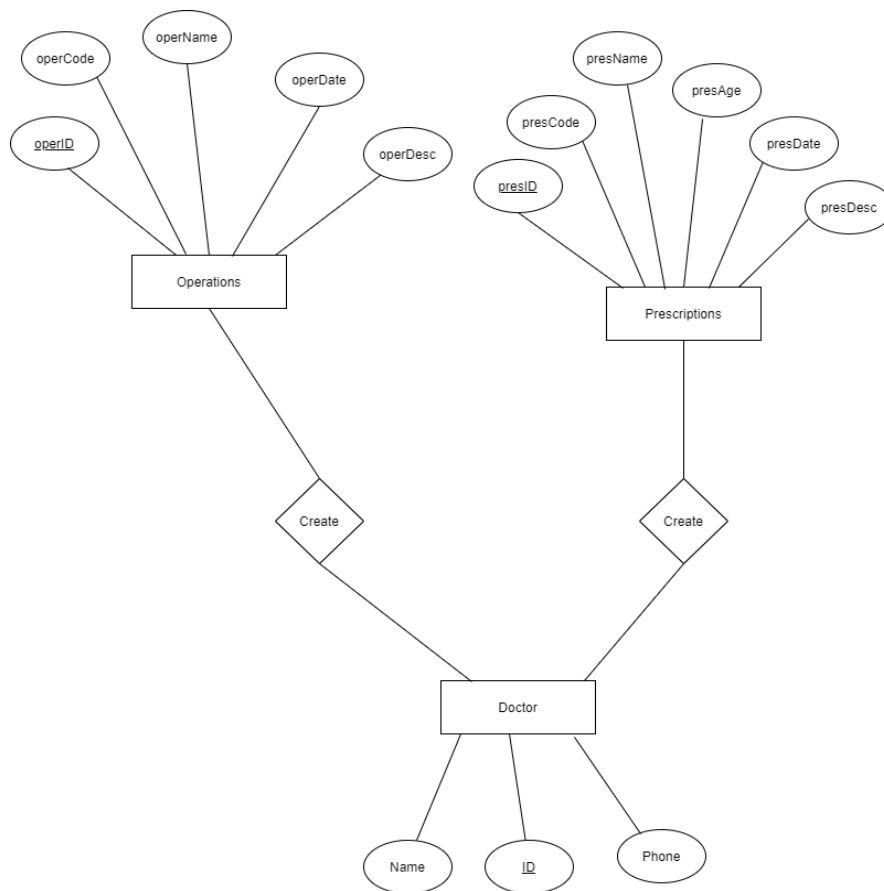
Response :- { result = "Updated successfully" } { error = "Error while updating!" }	Response :- { result = "Updated successfully" } { error = "Error while updating!" }
---	---

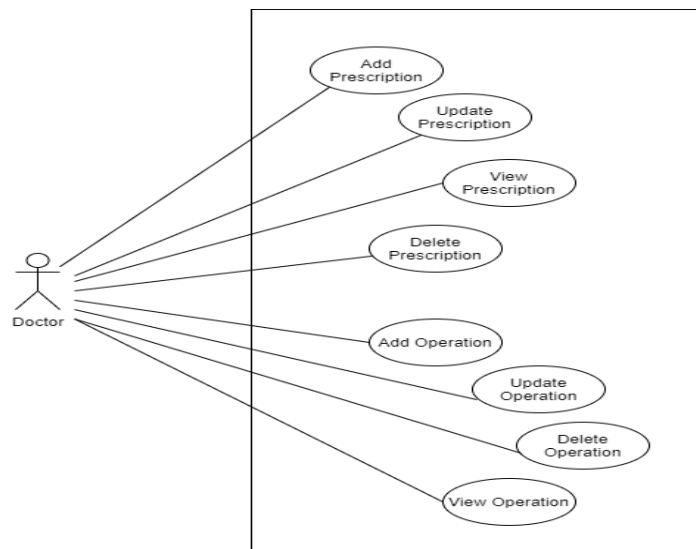
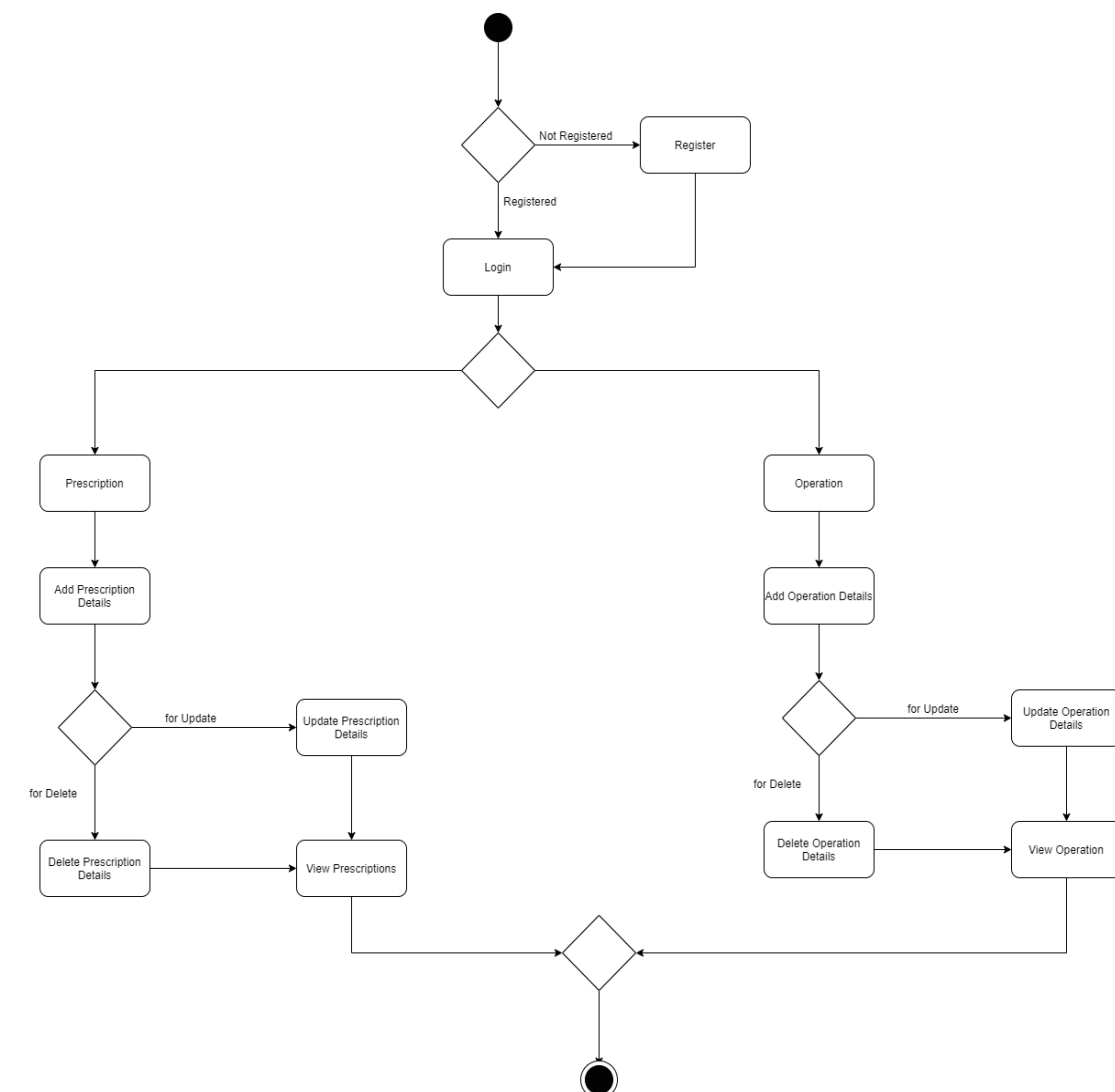
**API for Delete Prescriptions & Operations details in the database
(DELETE Request)**

<u>Prescriptions</u>	<u>Operations</u>
URL:- http://localhost:8080/DoctorService/DoctorService/DoctorsPrescriptions	URL:- http://localhost:8080/DoctorService/DoctorService/DoctorsOperations
Request :- { presID = selected presID from the service } Response :- { result = "Inserted successfully" } { presID= "auto generated integer value"} { error = "Error while inserting" }	Request :- { operID= selected presID from the service } Response :- { result = "Inserted successfully" } { operID= "auto generated integer value"} { error = "Error while inserting" }

Service Diagrams**Class Diagram (Prescription)->**

Class Diagram (Operation)->

ER Diagram->

Use case Diagram ->**Activity Diagram ->**

Registration Number: IT17021944

Name: T.T.D.kaluarachchi

Workload: Health care system

Group Name: S11 51.9

Description

Health care system is a system which can perform several tasks such as make an appointment. Making an appointment is a main function of a health care system. Users can register and login to the system and make their appointments such as add a new appointment, update an appointment, delete an appointment.

API for get all the appointment details in the database (GET Request)

url :- `http://localhost:8080/Appoint/Appoint/Appointment`

Request :- { }

Response :- { id = "auto generated integer value " }
 { Appointment_code = "001" }
 { Appointment_name = "Thimira" }
 { Appointment_description = "patient description" }

Request :- { }

Response :- { id = "auto generated integer value " }
 { Appointment_code = "002" }
 { Appointment_name = ""dilshan" }
 { Appointment_description = "494,mampe north piliyandala,heshan" }

API for insert a new appointment details to the database (POST Request)

url :- `http://localhost:8080/Appoint/Appoint/Appointment`

Request :- { id = "auto generated integer value " }
 { Appointment_code = "003" }
 { Appointment_name = "hemantha" }
 { Appointment_description = "patient description" }

Response :- { result = "inserted successfully" }
 { id = "auto generated integer value" }
 { error = "Error while inserting items." }

API for update an appointment detail which is existing in database (PUT Request)

url :- `http://localhost:8080/Appoint/Appoint/Appointment`

Request :- { id = "auto generated integer value " }
 { Appointment_code = "004" }
 { Appointment_name = "deshan" }
 { Appointment_description = "patient details" }

Response :- { result = “updated successfully”
 { error = “Error while updating items” }

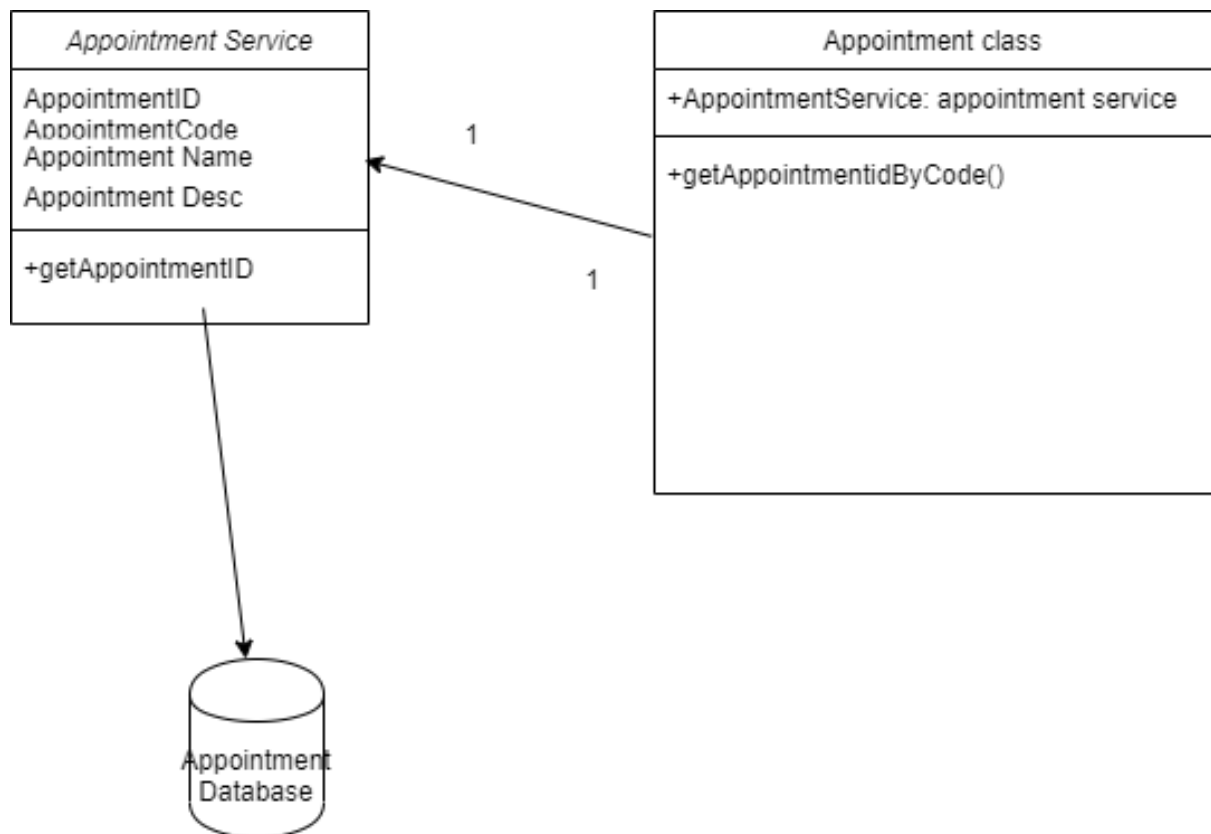
API for delete an appointment which is existing in database (DELETE Request)

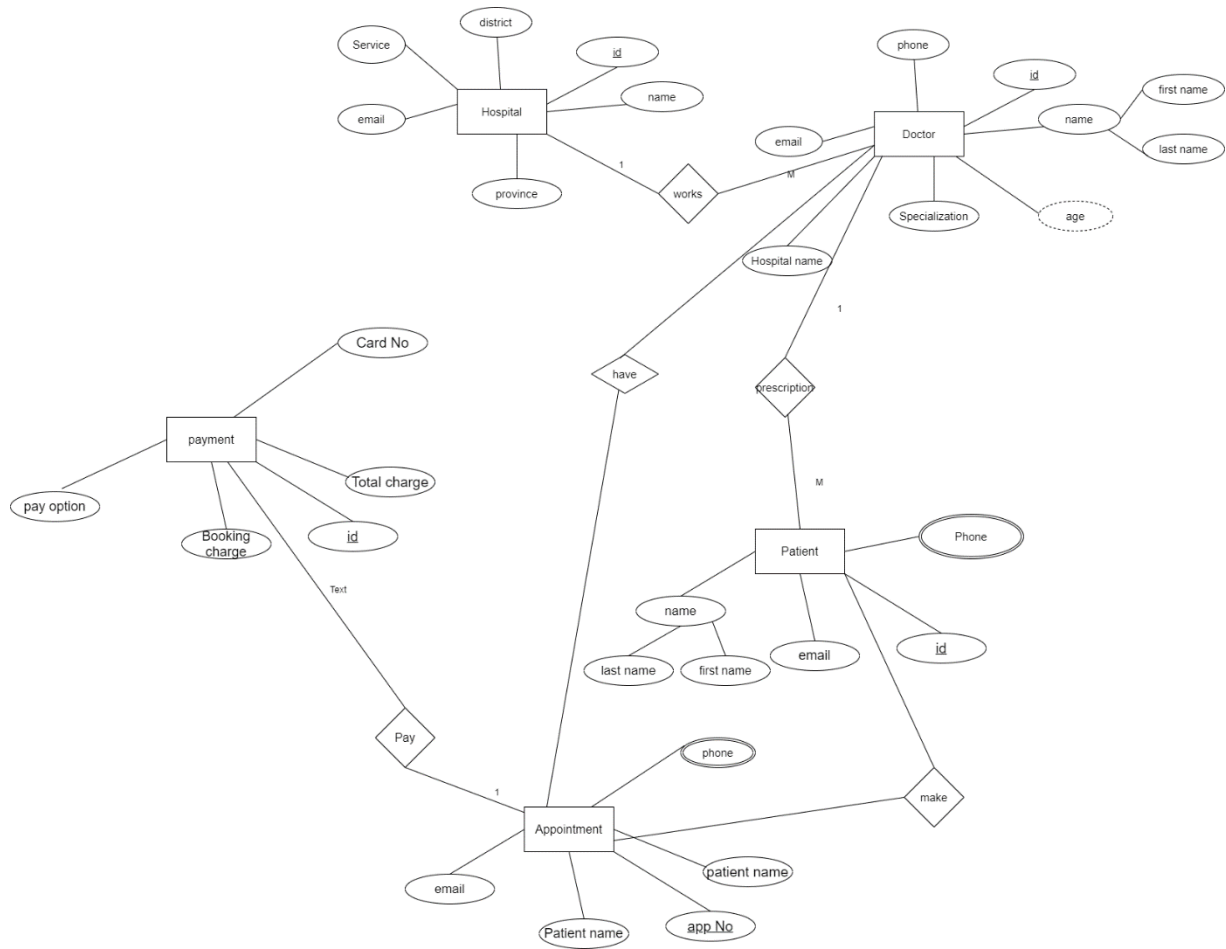
url :- http://localhost:8080/Appoint/Appoint/Appointment

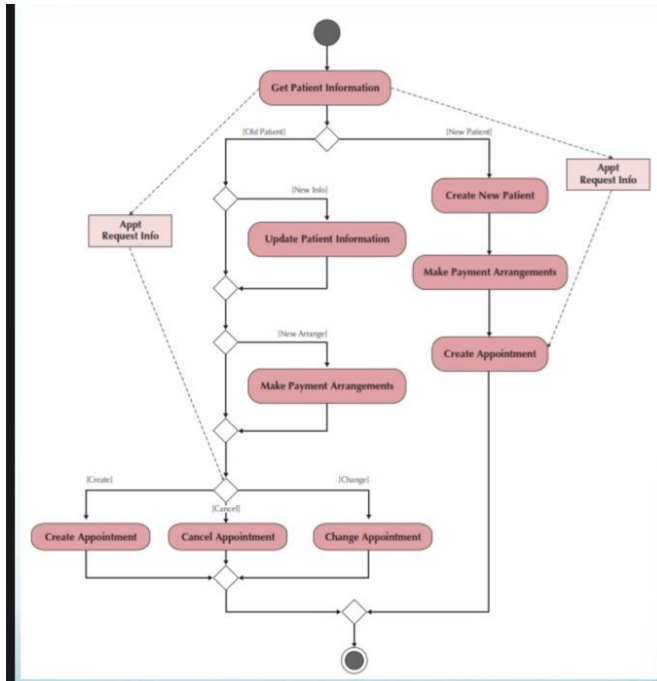
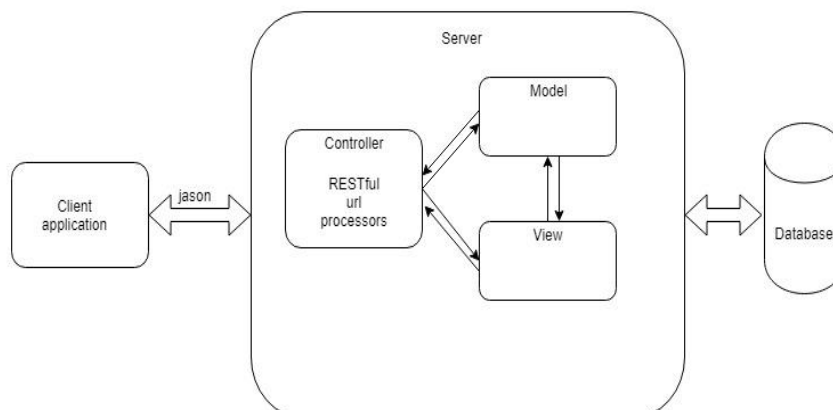
Request :- { id = selected id from the service }

Response :- { result = “deleted successfully” }
 { error = “Error while deleting items” }

Class Diagram->



ER Diagram->

Activity Diagram->**Internal Workflow->**

IT Number: IT18171198

Name: M.G.T.N.K.Abeykoon

Description

Health care system is a system which can perform several tasks such as make an appointment, patient registration, view hospitals and information about doctors. Registering patients is an essential function for the system. The registered patients can update, delete, and view their patient details

API for get all the patient details in the database (GET Request)

url :- `http://localhost:8050/PatientService/PatientService/patientRequest` :- { }

Response :- { pid = "auto generated integer value " }
 { pid = "P001" }
 { name = "thilan" }
 { asdress = "kandy road" }
 { phoneNo = "075654" }
 { year = "23" }
 {email="neth"}

Request :- { }

Response :- { pid = "auto generated integer value " }
 { pid = "P002" }
 { name = "dananga" }
 { asdress = "kandy road" }
 { phoneNo = "0756545" }
 { year = "20" }
 {email="dana@g"}

API for insert a new patient to the database (POST Request)

url :- `http://localhost:8050/PatientService/PatientService/patient`

Request :- { pid = "auto generated integer value " }
 { pid = "P003" }
 { name = "kumara" }
 { address = "malabe" }
 { phoneNo = "07758984" }
 { year = "21" }
 {email="kumara@g"}

Response :- { result = "Inserted successfully" }
 { p id = "auto generated integer value" }
 { error = "Error while inserting the patient" }

API for update a patient which is existing in database (PUT Request)

url :- `http://localhost:8050/PatientService/PatientService/patient`

Request :- { pid = "auto generated integer value " }
 { pid = "P003" }
 { name = "kumara" }
 { address = "malabe" }

```

{ phoneNo = "07758984" }
{ year = "21" }
{ email="kumara@gmail.com"}
Response      :-      { result = "Updated successfully"
                       { error = "Error while updating the patient" }

```

API for delete a patient which is existing in database (DELETE Request)

url :- <http://localhost:8050/PatientService/PatientService/patient>

Request :- { p id = selected id from the service }

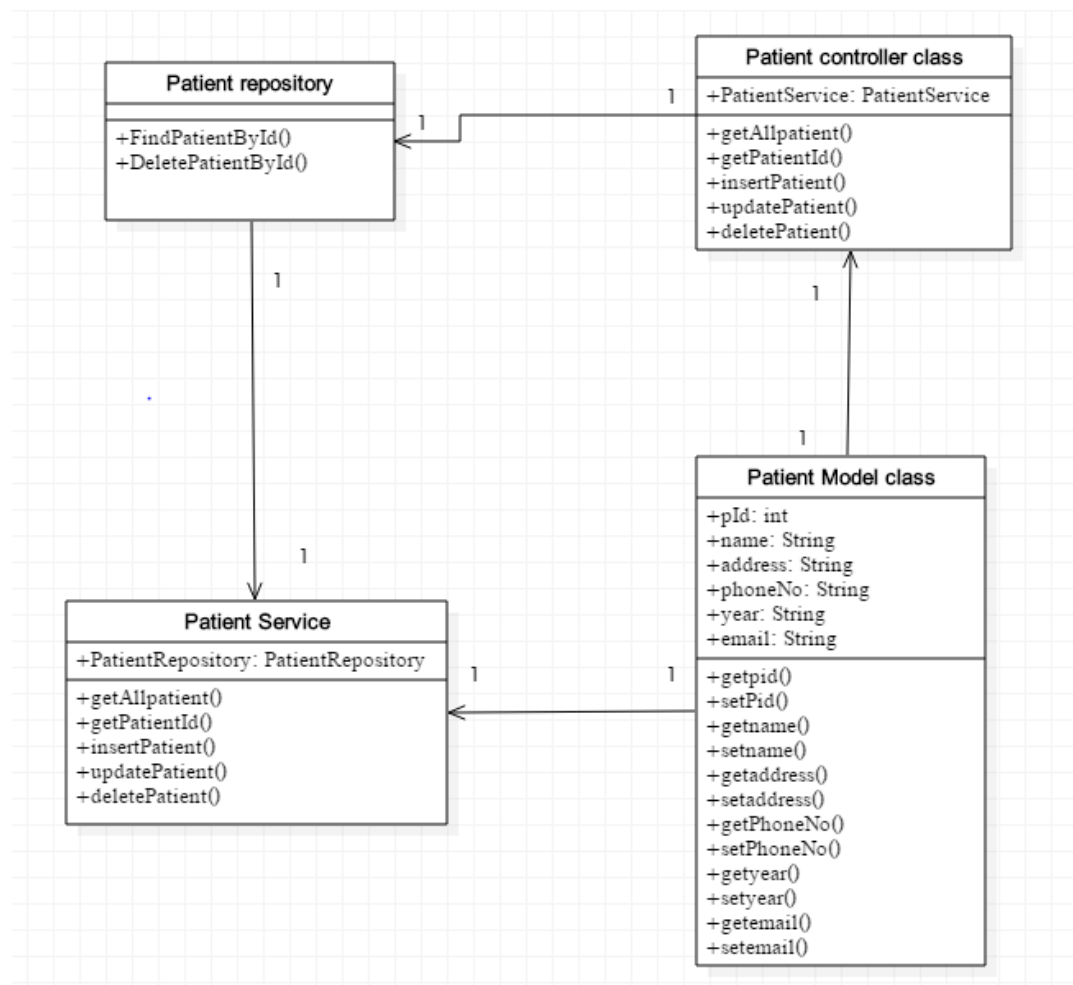
```

Response      :-      { result = "Deleted successfully"
                       { error = "Error while deleting the patient" }.

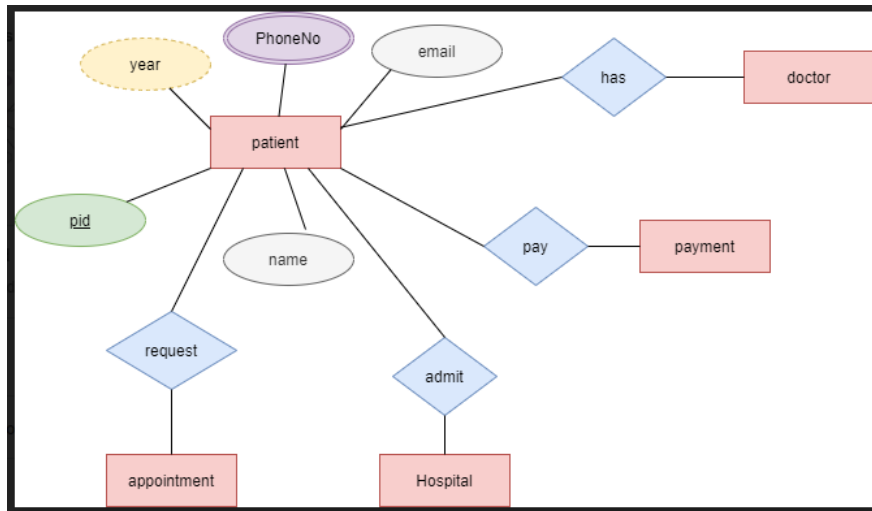
```

Service design

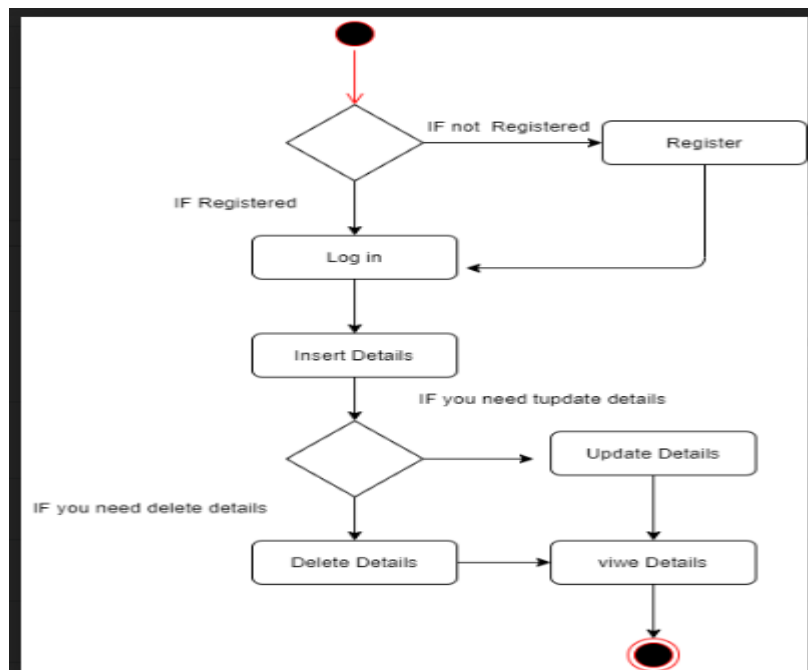
Class Diagram->



ER Diagram->



Activity Diagram->



IT Number: IT16038806

Name: Attanayaka.J.A.E.P.

API for get all the patient details in the database (GET Request)

url :- <http://localhost:8050/Hospital/ItemService/Hos>

```
Request      :-      { }
Response     :-      { HospitalID= "auto generated integer value " }
                  { HospitalCode= "Host3" }
                  { HospitalName= "Co-op" }
                  { HospitalAddress= "Kurunegala" }
                  { HospitalPhoNo= "372050348" }
```

```
Request      :-      { }
Response     :-      { HospitalID = "auto generated integer value " }
                  { HospitalCode = "Host4" }
                  { HospitalName = "Durdans" }
                  { HospitalAddress = "Kurunegala" }
                  { HospitalPhoNo = "372037521" }

                  { error = "Error while reading the Hospital." }
```

API for insert a new patient to the database (POST Request)

url :- <http://localhost:8050/Hospital/ItemService/Hos>

```
Request      :- { HospitalCode = "Host8" }
                  { HospitalName = "Peradeniya" }
                  { HospitalAddress = "Kandy" }
                  { HospitalPhoNo = "0375565958" }

Response     :- { result = "Inserted successfully" }
                  { HospitalID = "auto generated integer value" }
                  { error = "Error while inserting dataset" }
```

API for update a patient which is existing in database (PUT Request)

url :- <http://localhost:8050/Hospital/ItemService/Hos>

```
Request      :- { HospitalID = "7" }
                  { HospitalCode = "Host10" }
                  { HospitalName = "Lanka" }
                  { HospitalAddress = "Kiribathgoda" }
                  { HospitalPhoNo = "0112568342" }

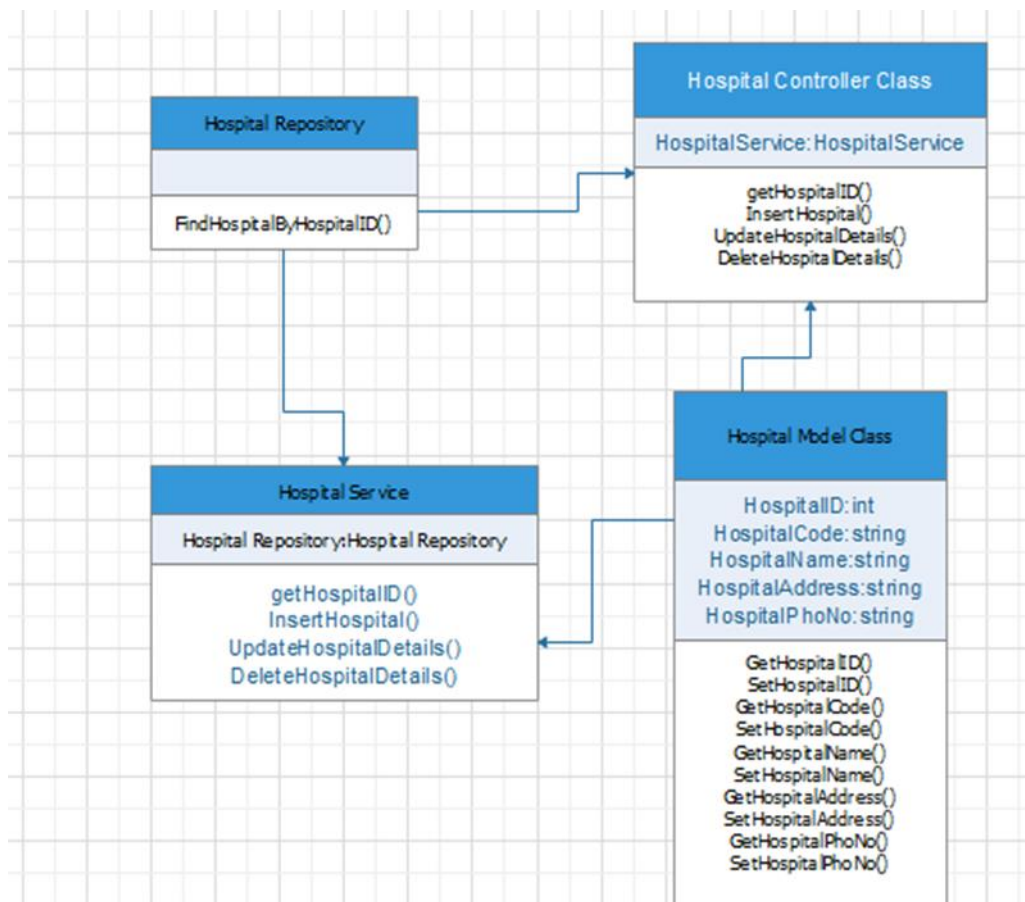
Response     :- { result = "Updated successfully" }
                  { error = "Error while updating the Hospital Dataset." }
```

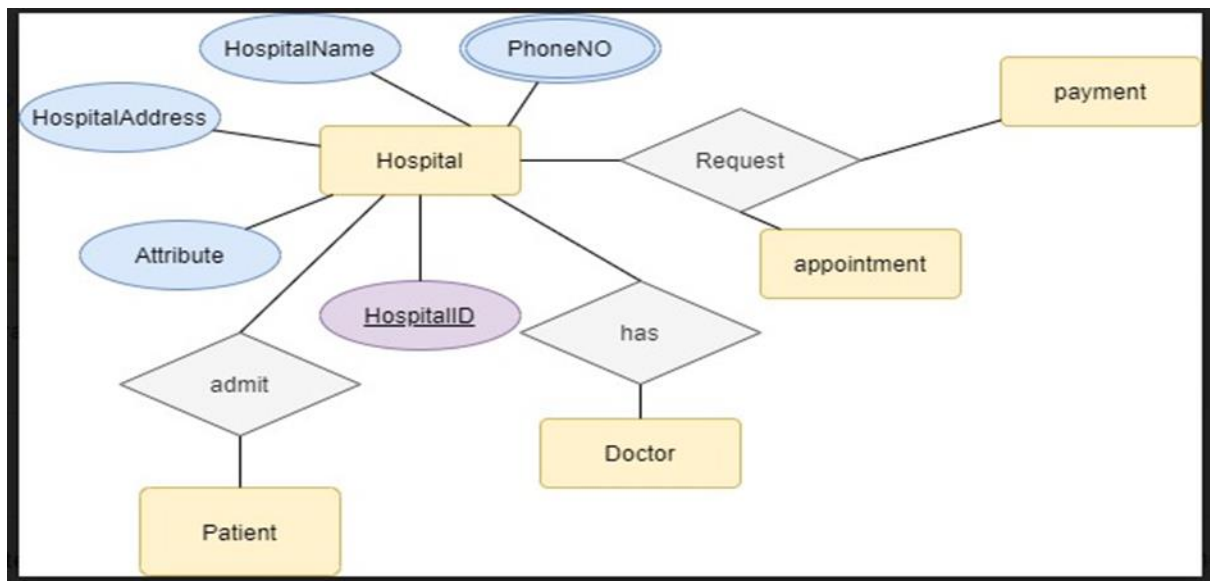
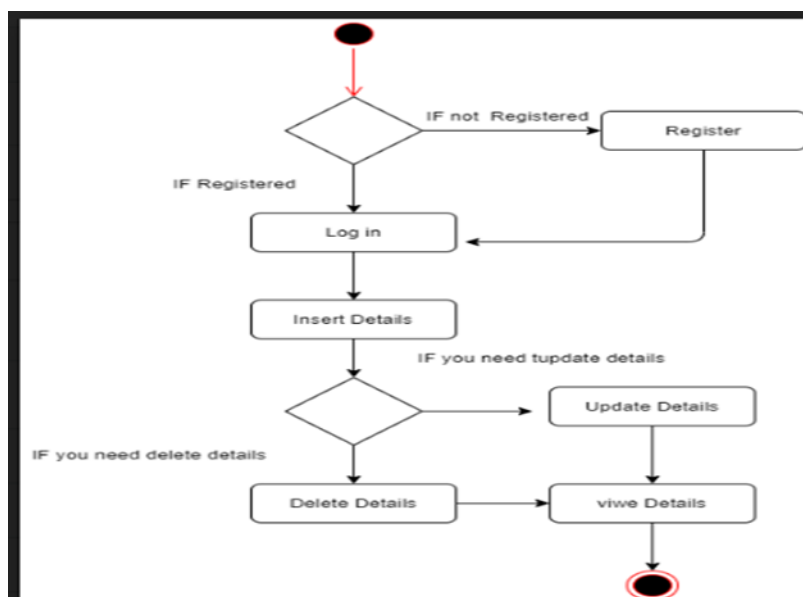
API for delete a patient which is existing in database (DELETE Request)

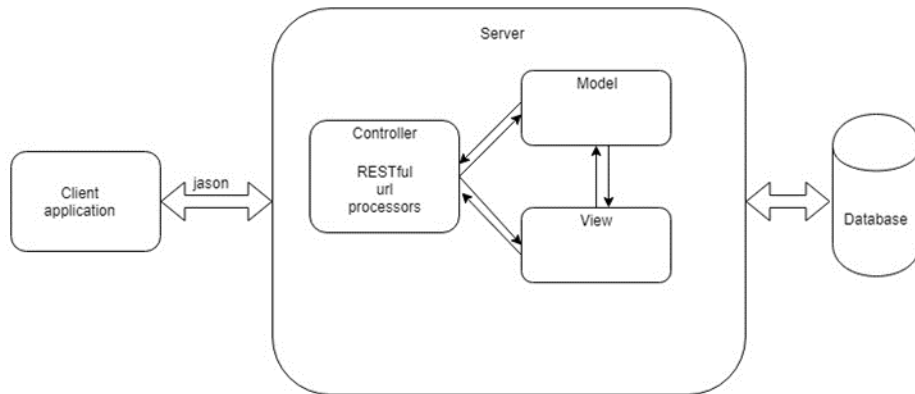
url :- <http://localhost:8050/Hospital/ItemService/Hos>

Request :- { HospitalID= 10}

Response :- { result = "Deleted successfully" }
 { error = "Error while deleting the hospital Dataset." }.

Class Diagram

ER Diagram->**Activity diagram->**



Software references

- Framework –J Query 3.2.1
- Scripting language –Java Script 1.6
- Database – MySQL 8.0
- Other- HTML, CSS v3.0

8. System's integration details

We developed the whole system as micro services and after the completion of micro services we integrated and combined our project using git hub. The integrated git hub repository is available in report. For testing purposes of the system, we used several tools such as selenium, SonarQube. Selenium is an ideal tool for testing web applications. Finally, we used postman tool to test the client components

9. References

<https://eclipse-ee4j.github.io/jersey/>

<https://www.javatpoint.com/jax-rs-example-jersey>

<https://www.vogella.com/tutorials/REST/article.html>

<https://www.mysql.com/>

<https://dev.mysql.com/downloads/connector/j/5.1.html>

Appendix 01

IT Number: IT18012620

POST ▼ http://localhost:8080/PAF_Project/PaymentServiceResource/Payments Send Save ▼

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies Code

☐ none ☐ form-data ☒ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	firstname	kamal			
<input checked="" type="checkbox"/>	lastname	mendis			
<input checked="" type="checkbox"/>	ammount	7000			
<input checked="" type="checkbox"/>	cardnumber	567800003434			
<input checked="" type="checkbox"/>	date	3/4/20			
<input checked="" type="checkbox"/>	cvv	900			
	Key	Value	Description		

Body Cookies Headers (3) Test Results Status: 200 OK Time: 72 ms Size: 130 B Save Response ▼

Pretty Raw Preview Visualize HTML ▼ 🔍

1 Inserted successfully

PUT ▼ http://localhost:8080/PAF_Project/PaymentServiceResource/Payments Send Save ▼

```
1 {  
2     
3   "paymentID": "9",  
4   "firstname": "kamal",  
5   "lastname": "mendis",  
6   "ammount": "5500",  
7   "cardnumber": "567800003434",  
8   "date": "3/4/20",  
9   "cvv": "900"  
10 }  
11 }
```

Body Cookies Headers (3) Test Results Status: 200 OK Time: 85 ms Size: 129 B Save Response ▼

Pretty Raw Preview Visualize 🔍

Updated successfully

GET http://localhost:8080/PAF_Project/PaymentServiceResource/Payments Send Save

<input checked="" type="checkbox"/>	firstname	kamal	
<input checked="" type="checkbox"/>	lastname	mendis	
<input checked="" type="checkbox"/>	ammount	7000	
<input checked="" type="checkbox"/>	cardnumber	567800003434	
<input checked="" type="checkbox"/>	date	3/4/20	
<input checked="" type="checkbox"/>	cvv	900	
	Key	Value	Description

Body Cookies Headers (3) Test Results Status: 200 OK Time: 51 ms Size: 1.36 KB Save Response

Pretty Raw Preview Visualize

First Name	Last Name	Ammount	Card Number	Date	Cvv	Update	Remove
vidura	yasiru	5000.0	222145667890	12/4/20	678	<button>Update</button>	<button>Remove</button>
saman	perera	4000.0	234511239876	13/4/20	345	<button>Update</button>	<button>Remove</button>
kamal	mendis	5500.0	567800003434	3/4/20	900	<button>Update</button>	<button>Remove</button>

DELETE http://localhost:8080/PAF_Project/PaymentServiceResource/Payments Send Save

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL ☐ XML Beautify

```
1 <paymentData>
2
3   <paymentID>
4     6
5   </paymentID>
6
7 </paymentData>
8
9
```

Body Cookies Headers (3) Test Results Status: 200 OK Time: 82 ms Size: 129 B Save Response

Pretty Raw Preview Visualize HTML

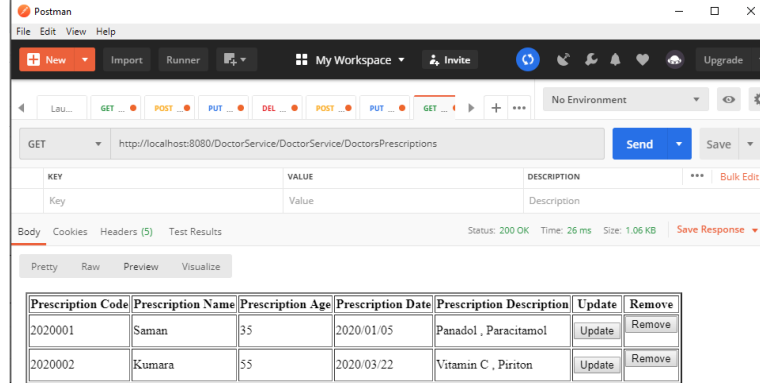
1 Deleted successfully

Appendix 02

IT Number: IT18152760

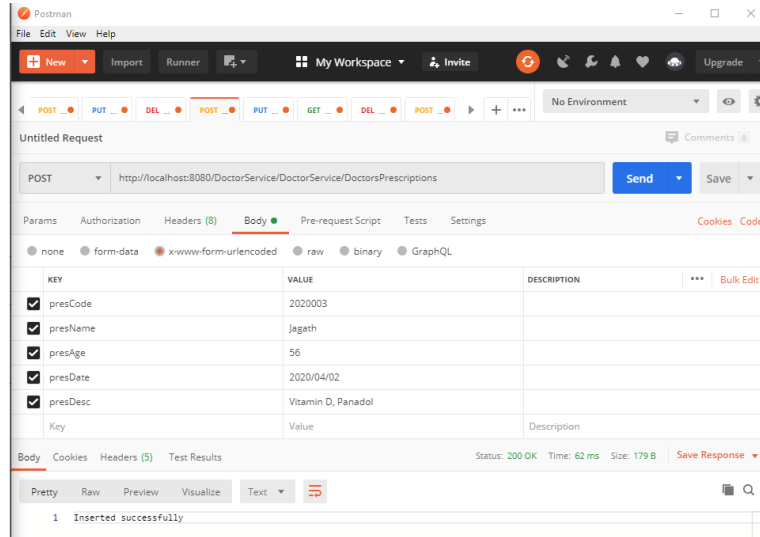
Prescriptions

View ->



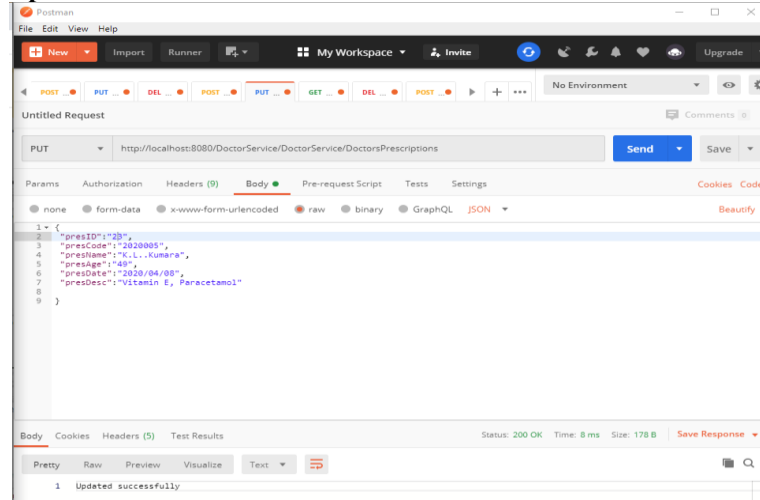
Prescription Code	Prescription Name	Prescription Age	Prescription Date	Prescription Description	Update	Remove
2020001	Saman	35	2020/01/05	Panadol , Paracetamol	Update	Remove
2020002	Kumara	55	2020/03/22	Vitamin C , Piriton	Update	Remove

Insert->



1 Inserted successfully

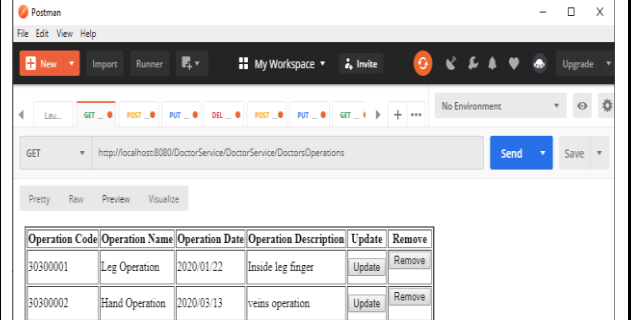
Update->



1 Updated successfully

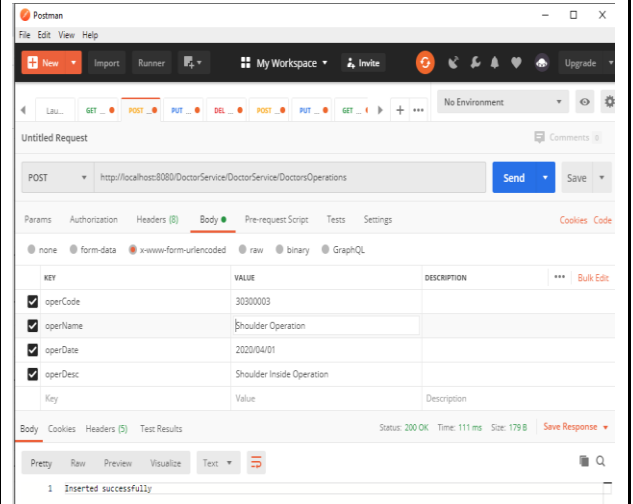
Operations

View ->



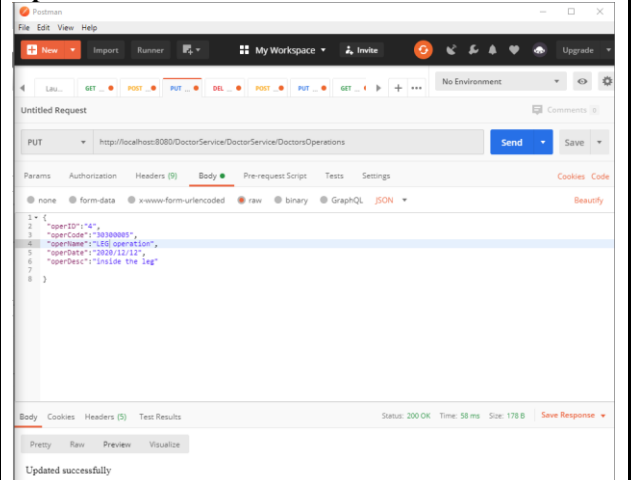
Operation Code	Operation Name	Operation Date	Operation Description	Update	Remove
30300001	Leg Operation	2020/01/22	Inside leg finger	Update	Remove
30300002	Hand Operation	2020/03/13	veins operation	Update	Remove

Insert->

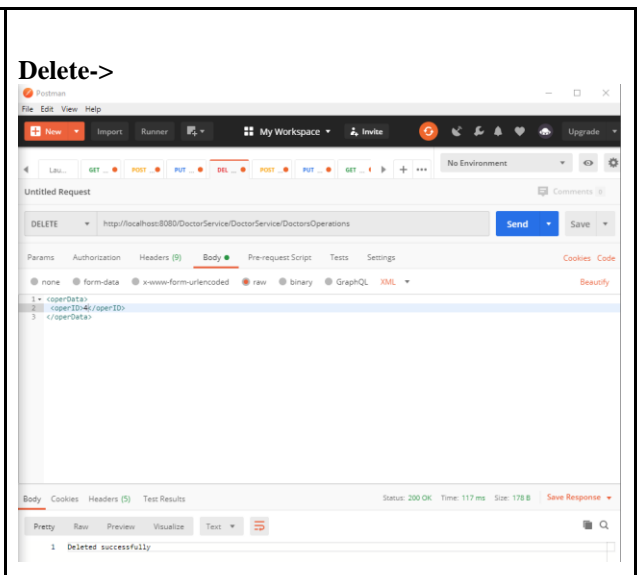
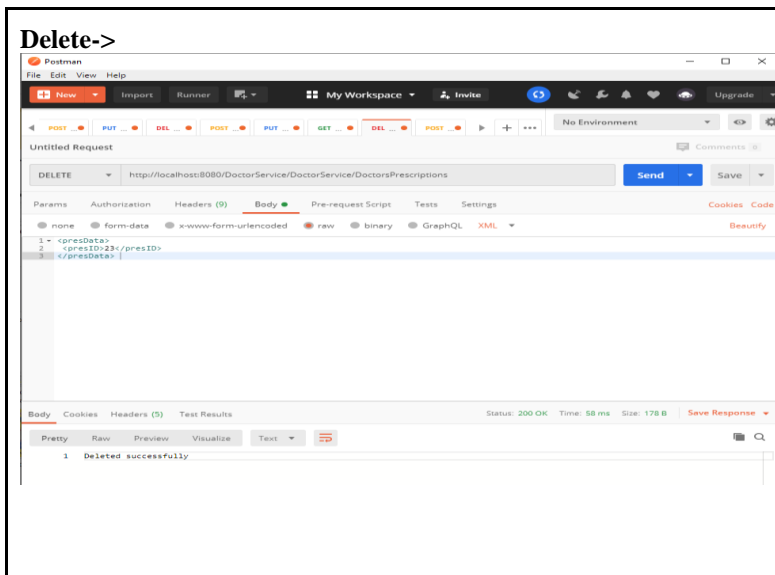


1 Inserted successfully

Update->



Updated successfully



Appendix 03

IT Number: IT16038806

GET http://localhost:8050... POST http://localhost:8050... PUT http://localhost:8050... DEL http://localhost:8050...

Untitled Request

GET http://localhost:8050/Hospital/ItemService/Hos

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize

HospitalCode	HospitalName	HospitalAddress	HospitalPhoNo	Update	Remove
Host10	Lanka	Kiribathgoda	112568342	<button>Update</button>	<button>Remove</button>
Host3	Co-op	Kurunegala	372050348	<button>Update</button>	<button>Remove</button>
Host4	Durdans	Kurunegala	372037521	<button>Update</button>	<button>Remove</button>
Host8	Peradeniya	Kandy	375565958	<button>Update</button>	<button>Remove</button>

GET http://localhost:8050... POST http://localhost:8050... PUT http://localhost:8050... DEL ht

Untitled Request

POST http://localhost:8050/Hospital/ItemService/Hos

Params Authorization Headers (8) Body Pre-request Script Tests Se

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE
<input checked="" type="checkbox"/> HospitalCode	Host8
<input checked="" type="checkbox"/> HospitalName	Peradeniya
<input checked="" type="checkbox"/> HospitalAddress	Kandy
<input checked="" type="checkbox"/> HospitalPhoNo	0375565958
Key	Value

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize Text

1 Inserted successfully

The image shows two screenshots of a REST client interface, likely Postman, demonstrating API interactions.

Top Screenshot: PUT Request

- Method:** PUT
- URL:** http://localhost:8050/Hospital/ItemService/Hos
- Body Type:** JSON
- Body Content:**

```
1 {  
2   "HospitalID": "7",  
3   "HospitalCode": "Host10",  
4   "HospitalName": "Lanka",  
5   "HospitalAddress": "Kiribathgoda",  
6   "HospitalPhoNo": "0112568342"  
7 }
```
- Response:** 1 Updated successfully

Bottom Screenshot: DELETE Request

- Method:** DELETE
- URL:** http://localhost:8050/Hospital/ItemService/Hos
- Body Type:** XML
- Body Content:**

```
1 <HospitalData>  
2 <HospitalID>10</HospitalID>  
3 </HospitalData>
```
- Response:** 1 Deleted successfully