**Health Service Management**

# Introduction

**Health Service Management**

Healthcare management is a healthcare manager is in charge of ensuring a healthcare facility is running smoothly and enabling to reach the goals of the facility’s practitioners and the needs of the community.

We have built healthcare management software that oversees the day-to-day operations of the facility.

# Functional Requirements

Doctors Dashboard

* Doctors Registration

Register doctor with specialization (Heart/Pediatrician, Gynecologists…), Personal information, availability, qualification/experience/awards

              Patients Dashboard

* Patients Registration

Register Patient with all symptoms and personal details

* Patients History

Old dieses history, reports -Blood, X-ray etc, and all details related to treatment, doctor advice and diet

* Patient Case – Treatment module

Current treatment, -doctor associated, - reports, medicines, detail description related to disease

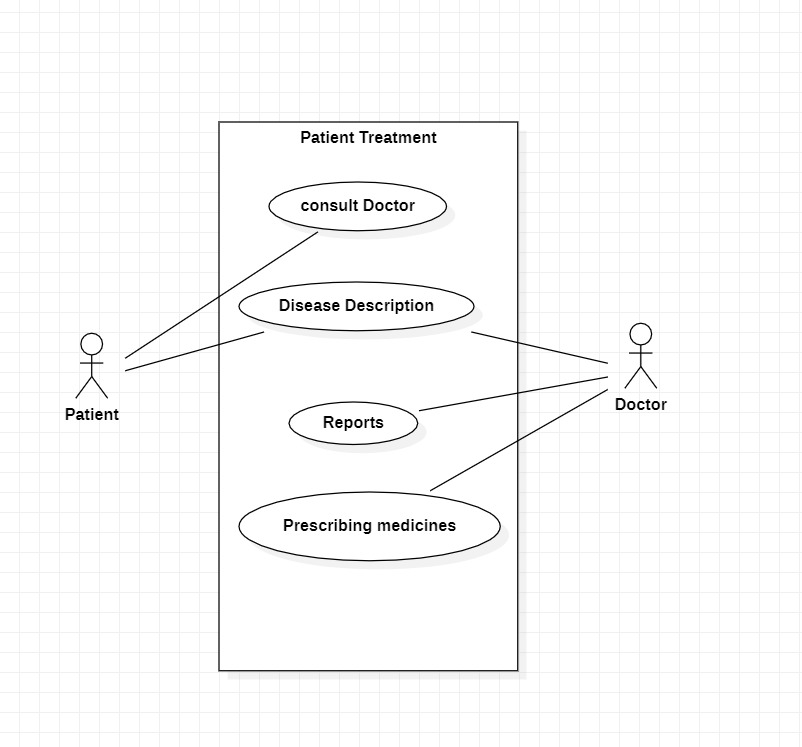
Admin Dashboard

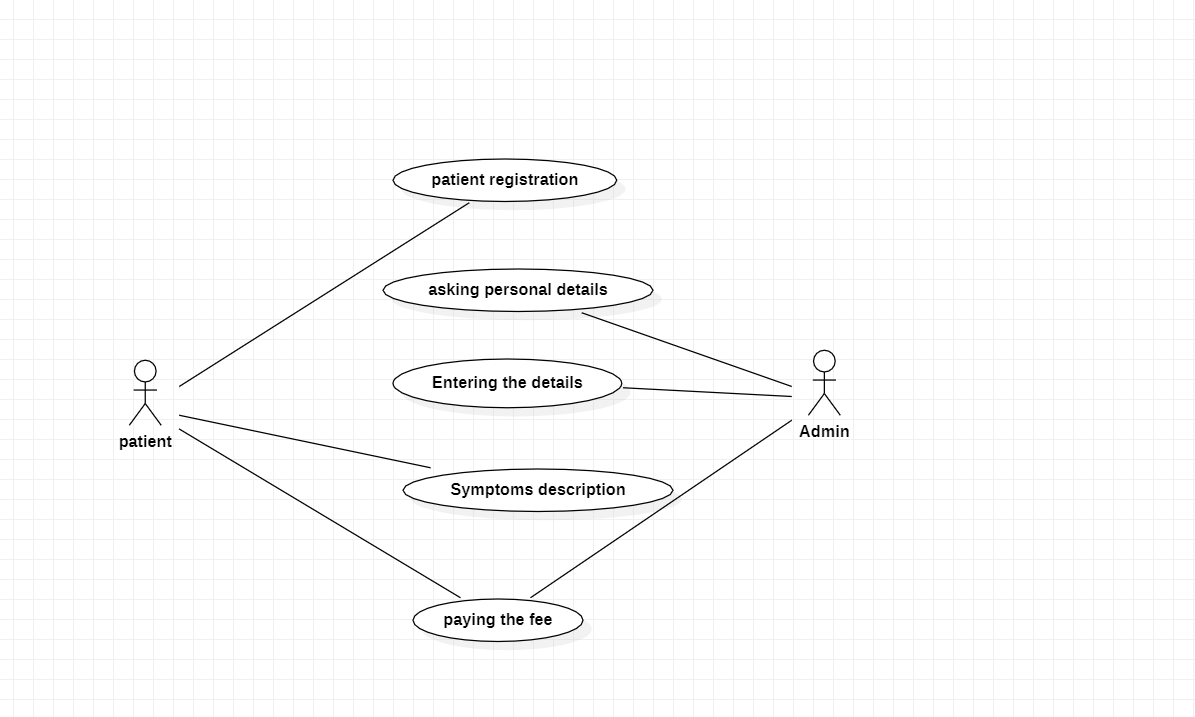
* Finance module- Doctors fees, Registration fees, Medicines amount
* Health Care Insurance Policy – can cover in policy, wanted to buy new policy

Updating Doctors and Patient details

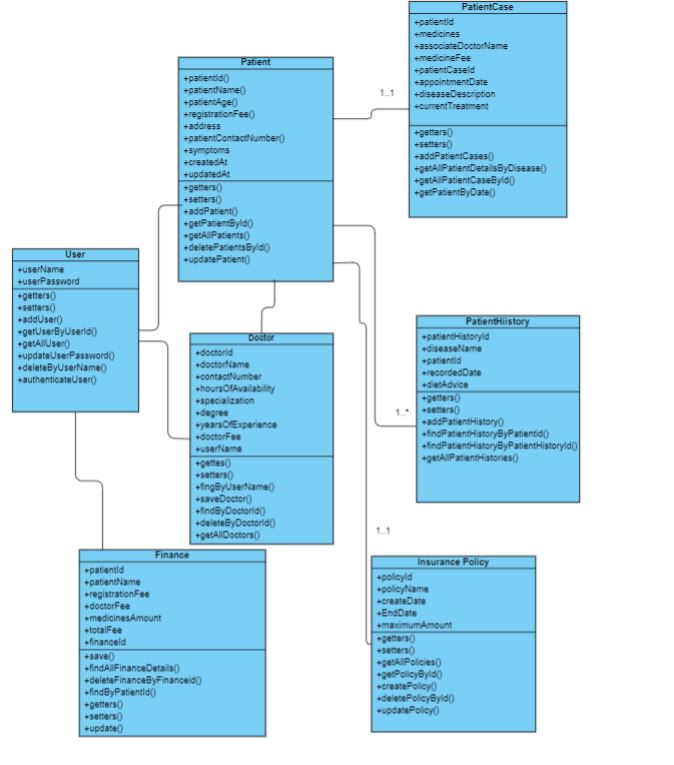
|  |  |
| --- | --- |
| **Team Members** | **Use Case Working on** |
| Allu Suma Sai Venkata Ratnam(TL) | Insurance Policy Module |
| Kapaganti Ratna Harika | Patient Module |
| Kalavakollu HariPriya | User Module |
| Shivangi Tirpathi | Finance Module |
| Gundabathula Sai Lakshmi Bhavani | PatientHistory |
| Akula Lakshmi | Doctor |
| Goripatri Renu Sree | PatientCase |

# UML Diagrams

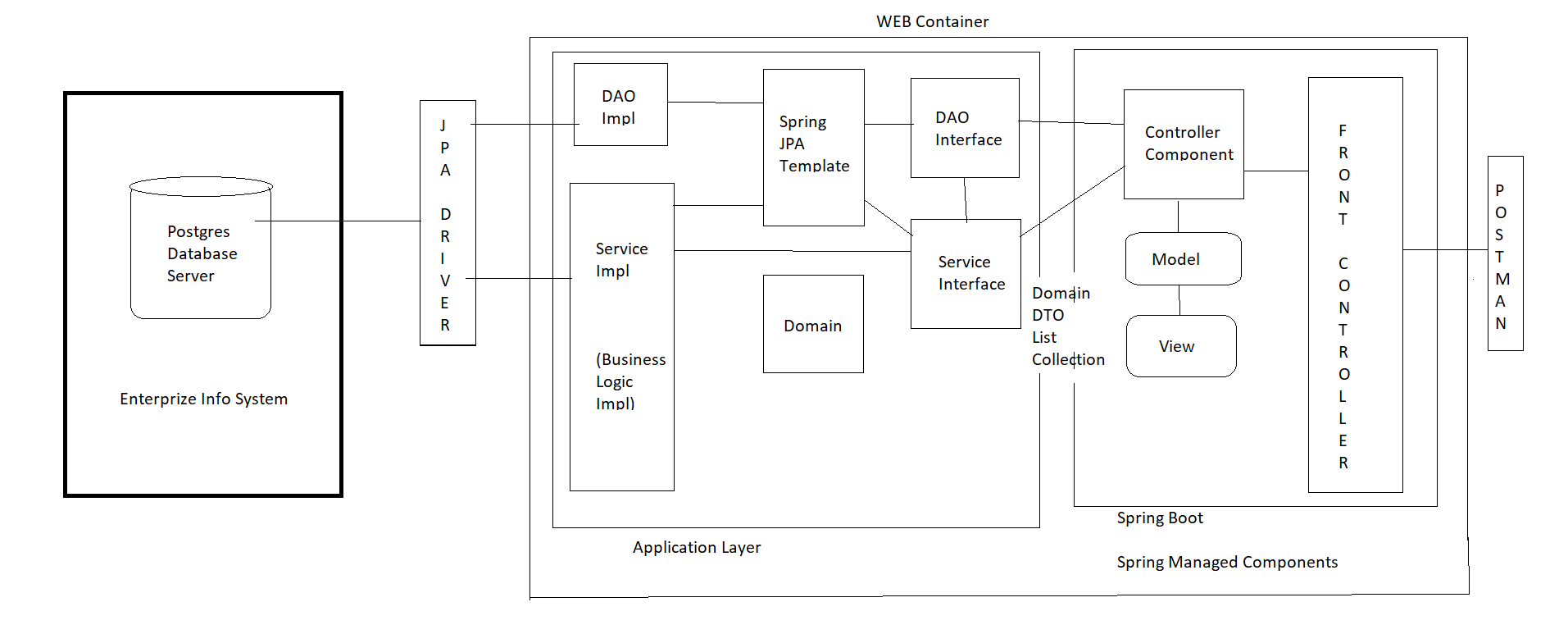




# Class Design



(5) Save, findAllFinanceDetails, deleteFinanceByFinanceid, findByPatientId, update



# Rest Service Specification

1. **Doctor Controller**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Request Method | URI | Input | Output | Description |
| POST | /api/doctors | Doctor Object | ResponseEntity object wrapping Doctor object | This API end point is to add Doctor record to database |
| GET | /api/doctors/alldoctors | None | List of Doctors | This API end point is to return list of Doctors |
| GET | /api/doctors/<doctorId> | doctorId | ResponseEntity object wrapping Doctor object | This API end point is to return Doctor object |
| DELETE | /api/doctors/<doctorId> | doctorId | String message | This API endpoint is to delete Doctor record from the database |
| PUT | /api/doctors | Doctor object | Doctor object | This endpoint receives Doctor object and returns Doctor object |

1. **Finance Controller:**
2. **Patient Controller**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Request Method** | **URI** | **Input** | **Output** | **Description** |
| POST | /api/finance | Finance Object | ResponseEntity object wrapping Finance object | This API end point is to add Patient record to database |
| GET | /api/finance/all | None | List of Finance objects | This API end point is to return list of Finance objects |
| GET | /api/finance/patientId | patientId | ResponseEntity object wrapping Finance object | This API end point is to return Finance object related to the Patient |
| DELETE | /api/finance/patientId | patientId | String message | This API endpoint is to delete Finance record related to patientId from the database |
| PUT | /api/finance/patientId | Finance  object | Updated Finance object | This endpoint receives patientId and returns updated Finance object |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Request Method** | **URI** | **Input** | **Output** | **Description** |
| POST | /api/patients/add | Patient Object | ResponseEntity object wrapping Patient object | This API end point is to add Patient record to database |
| GET | /api/patients/all | None | List of Patients | This API end point is to return list of Patients |
| GET | /api/patients/patientId | patientId | ResponseEntity object wrapping Patient object | This API end point is to return Patient object |
| DELETE | /api/patients/patientId | patientId | String message | This API endpoint is to delete Patient record from the database |
| PUT | /api/patients/patientId | Patient object | Patient object | This endpoint receives Patient object and returns Patient object |

1. **Insurance Policy Controller**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Request Method** | **URI** | **Input** | **Output** | **Description** |
| POST | /api/policies | Policy Object | Response Entity object wrapping Insurance Policy object. | This API end point is to add policy record to database |
| GET | /api/polcies/all | None | List of Insurance Policy objects | This API end point is to return list of Insurance Policy objects |
| GET | /api/polcies/policyId | Policy Id | Response Entity object wrapping Policy Object | This API end point is to return Finance object related to the Patient |
| DELETE | /api/policies /policyId | Policy Id | String message | This API endpoint is to delete Policy record related to  Related to p id from the database |
| PUT | /api/policies/policyName | Policy  object | Updated Policy object | This endpoint receives policy name and returns updated Policy object |

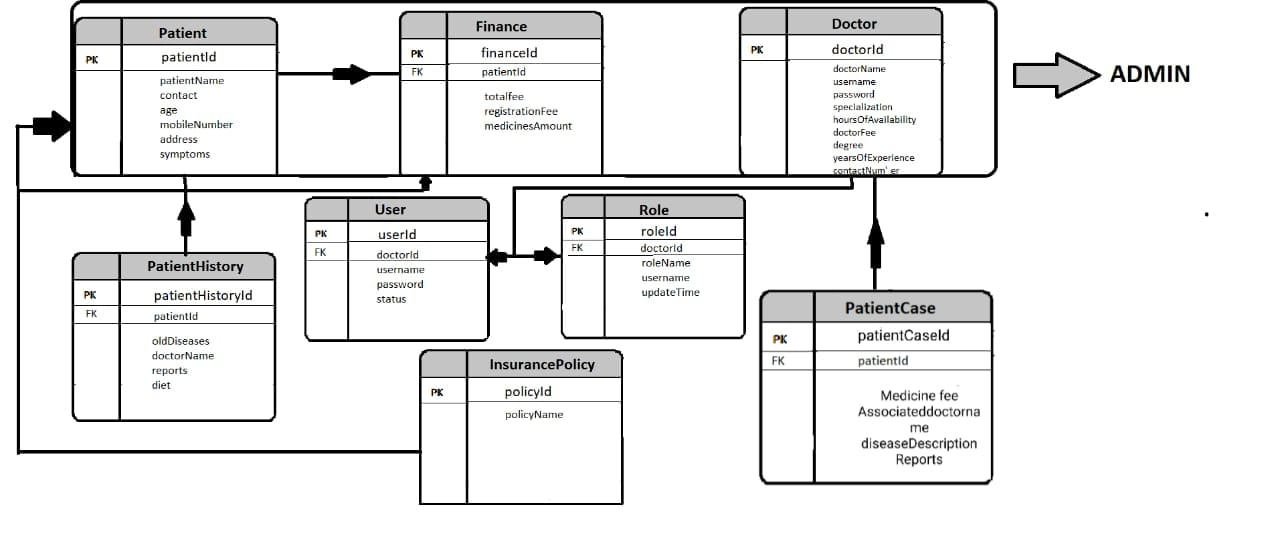
1. **Patient Case Controller**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Request Method** | **URI** | **Input** | **Output** | **Description** |
| POST | /api/patients | Patient Object | ResponseEntity object wrapping Patient object | This API end point is to add Patient record to database |
| GET | /api/patients/all | None | List of Patients | This API end point is to return list of Patients |
| GET | /api/patients/patientId | patientId | ResponseEntity object wrapping Patient object | This API end point is to return Patient object |
| DELETE | /api/patients/patientId | patientId | String message | This API endpoint is to delete Patient record from the database |

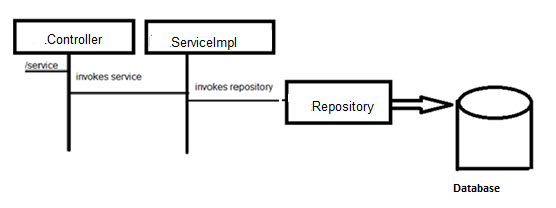
1. **Patient History Controller**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Request Method** | **URI** | **Input** | **Output** | **Description** |
| POST | /api/patients | Patient Object | ResponseEntity object wrapping Patient object | This API end point is to add  Patient record to database |
| GET | /api/patients/all | None | List of Patients | This API end point is to return  list of Patients |
| DELETE | /api/patients/patientId | patientId | String message | This API endpoint is to delete  Patient record from the database |
| PUT | /api/patients/patientId | Patient object | Patient object | This endpoint receives Patient object and returns Patient  object |

# ER- Diagram

****

# Process Flow Diagram



In this application, we have seven RestController classes each exposing API endpoint to perform CRUD operations on Doctor, Patient, Finance and User objects.

We have followed 3-tier layered architecture where the methods in Controller class will be calling the corresponding methods of Service class. The methods of Service class invoke the corresponding methods of Repository class. The methods of Repository class perform CRUD operations on the database.

# Overall Design Constraints

Healthcare management is an umbrella term that covers a wide variety of job titles. Clinical directors, healthcare supervisors, health coordinators, and nursing home facilitators often have degrees in healthcare management.

In our application we have covered only some important functionalities due to our project scope and time constraints.