MedOne Design

Use Case Diagram:

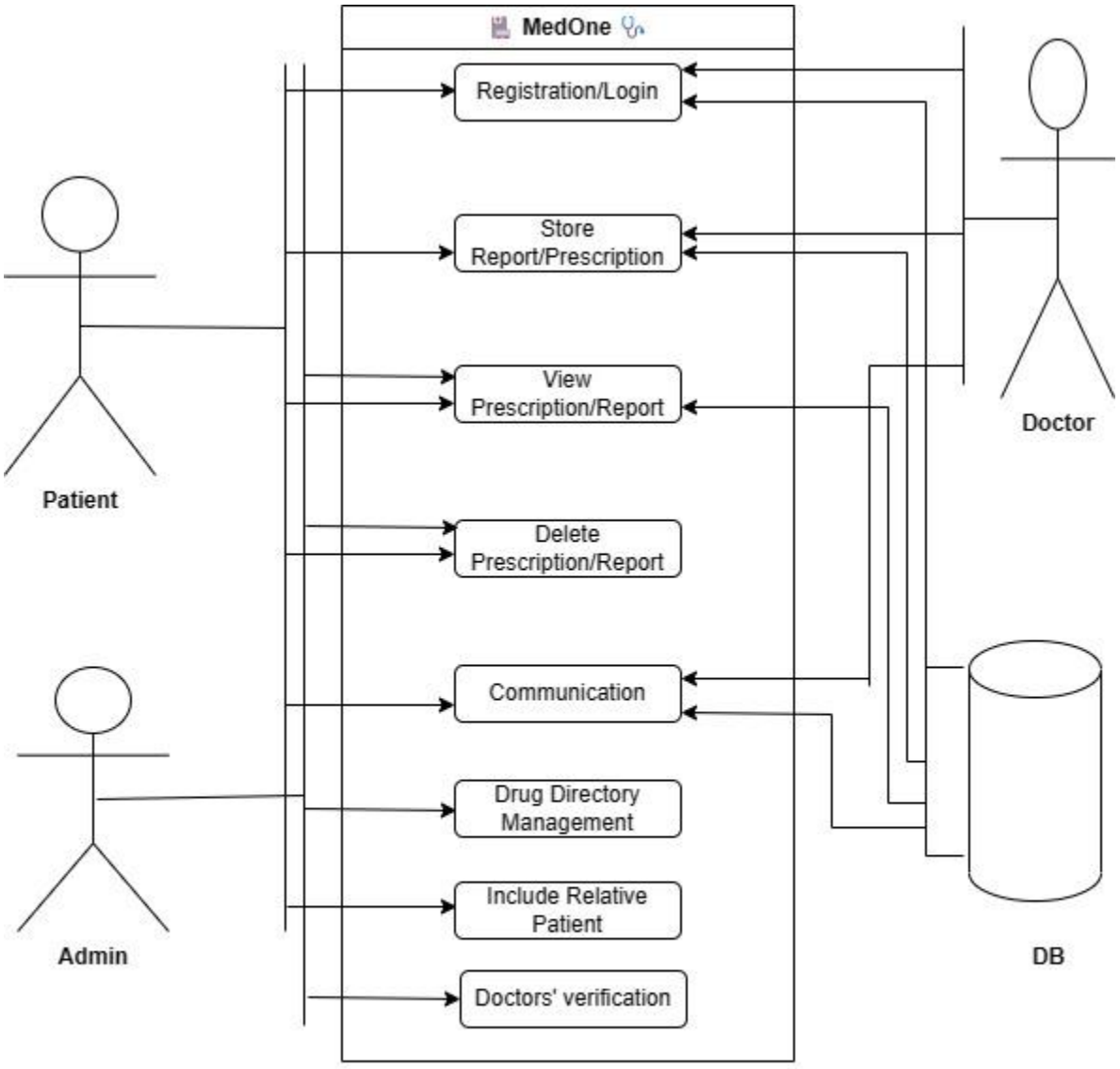
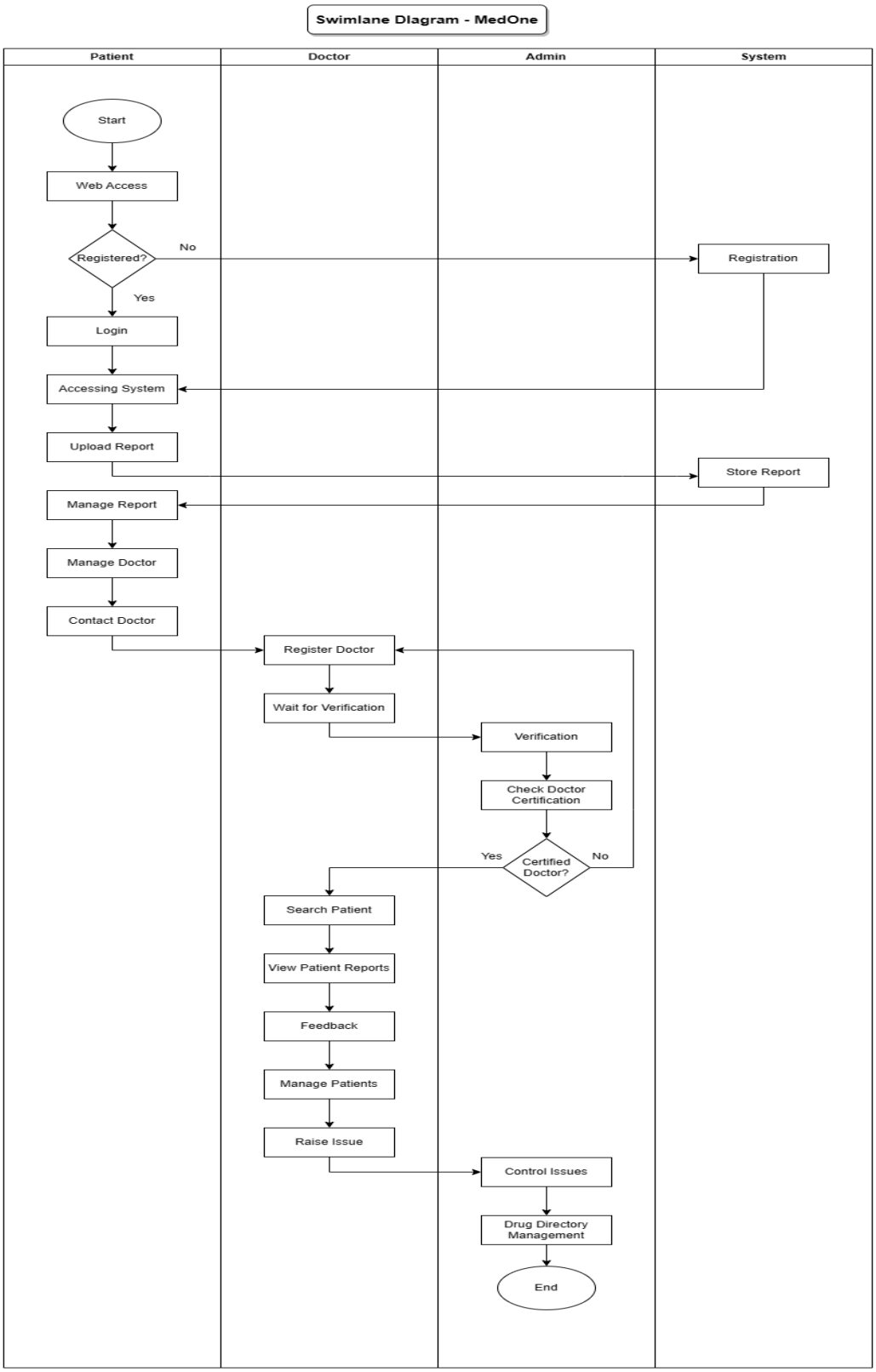


Figure : Use case Diagram

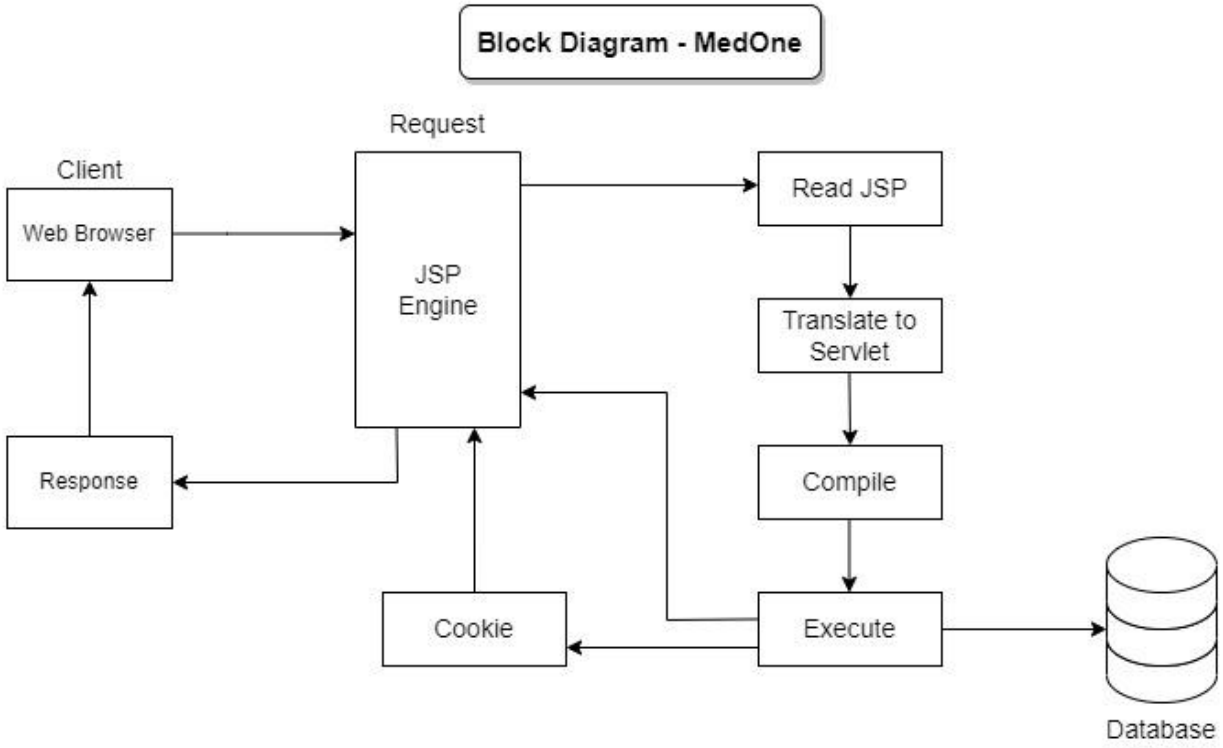
In this use case diagram, three actors (Patient, Doctor and Admin) access and use the services of the system in coordination with the database. The Patient can manage personal reports, communicate with doctors and include relatives. The Doctor can view patient reports, contact them, whereas the Admin controls the whole system's stability by controlling issues, managing drug directory and verifying doctors' certifications.

Swimlane Diagram:



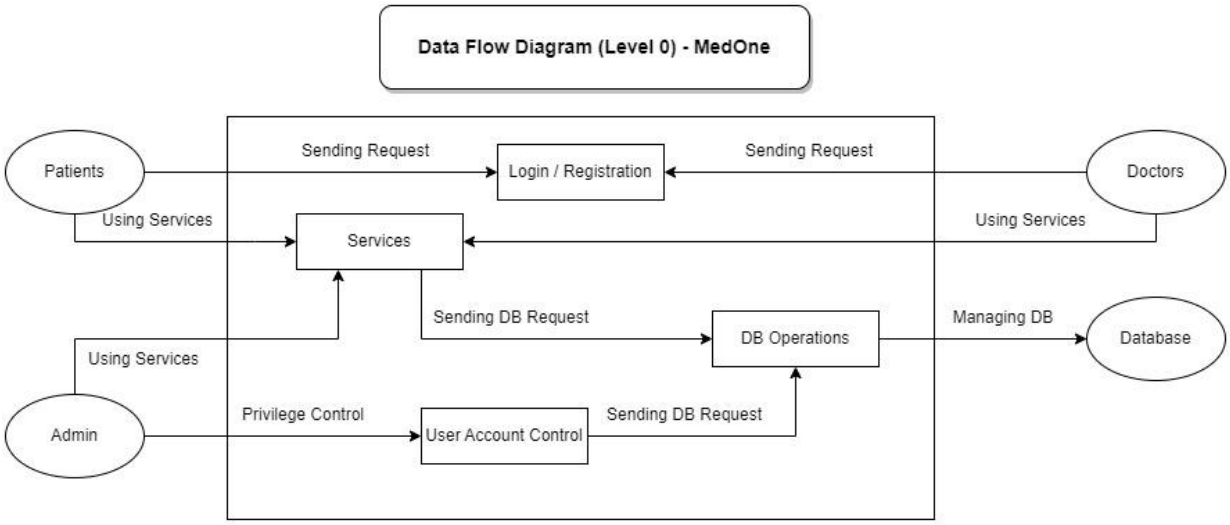
In this swimlane diagram, the activities of the actors and passing of control are shown. The patient after logging into the system, can manage their reports and contact their concerned doctors. The doctors who are registered with valid certifications can respond to their patients and provide necessary feedback. The verification of doctors' certification is handled by the Admin. The Admin also deals with the issues raised within the system by the users. The registration credentials and report management operations occur within the system.

Block Diagram:



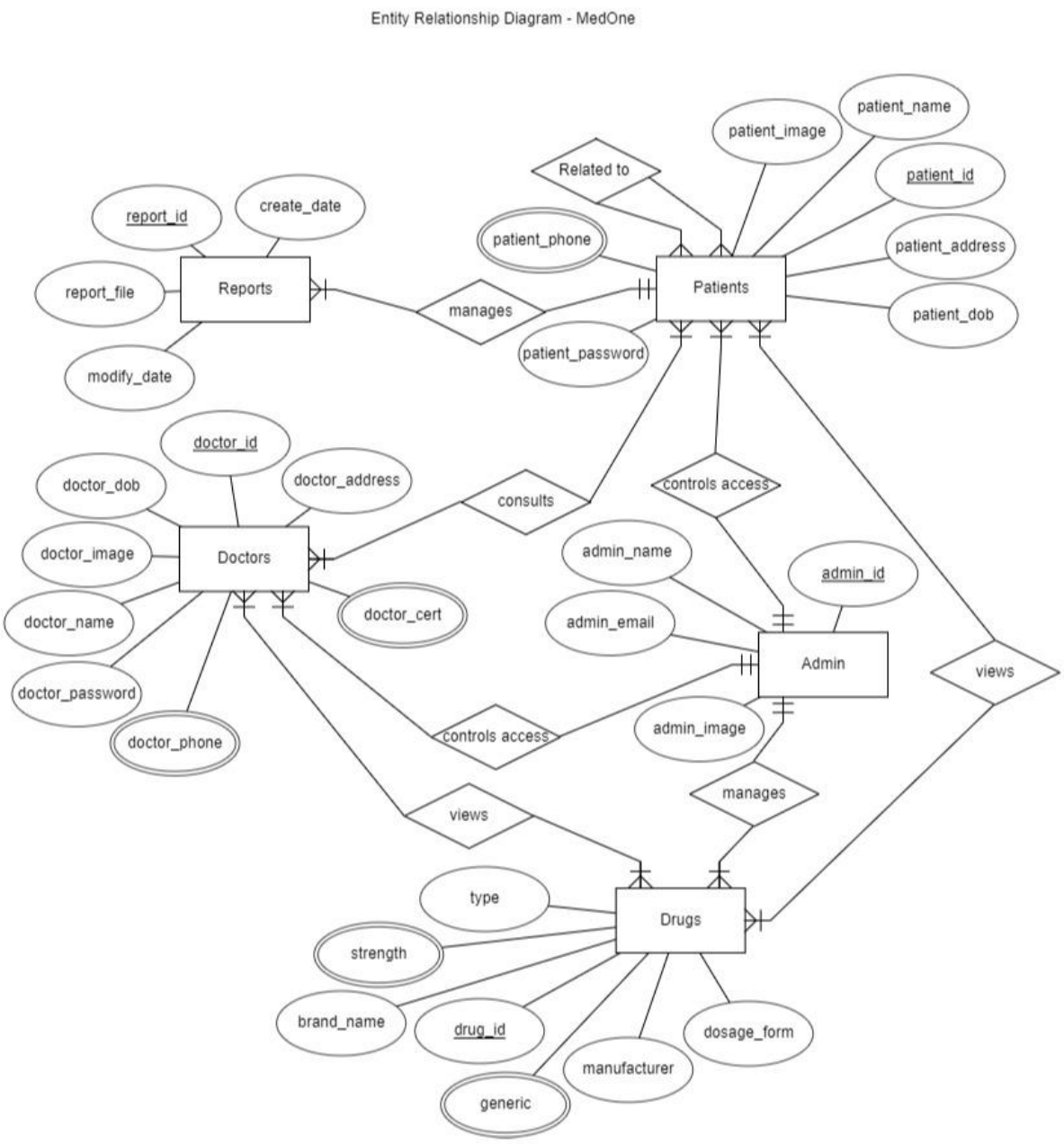
In the above block diagram, we can see that the client can send requests that are passed through the JSP Engine to the servlet where the commands are executed and necessary Database operations are performed. Then the response is again sent back through the JSP Engine to the client's web browser. Sometimes, for gaining better performance, the responses are passed through the Cookie to the JSP Engine.

Data Flow Diagram (Level 0):



In this data flow diagram, a bird's eye view of data flow among actors and their respective use cases are shown. The actors can send the login / registration requests by providing the necessary credentials to access the system. Then the services can be used by them according to their access control schemes. The Admin can control this user access and set necessary changes in the database for specific actors.

Entity Relationship Diagram:



In the above entity relationship diagram, all the possible entities that constitute MedOne, are shown along with their attributes. The relationships among individual entities are also shown. The patient can be related to another patient which is depicted by the self relationship. Also the patient can ask for consultation to doctors, view the drug directory, manage their reports. The doctor can provide consultation to patients, view the drug directory. The admin manages the drug directory, and controls user access for both patients and doctors. The diagram is completely normalized and loosely coupled.

State Diagram:

In the state diagram above, the change of system states along with their operations are shown in three different branches for three different actors (Patient, Doctor and Admin). The possible use cases are shown in each state along with their outputs on downward arrow signs. Each of the states is an oval containing the use case of the state in the upper part and its operation in the lower part.

Scenario:

Actor - Patient:

A patient firstly registers into the system, using important credentials such as Username, Password, Email, Date of birth, Contact number. Registered patients can login using contact number and password. The patients can upload / delete their medical reports which are stored in the database. They can also view their uploaded reports within the app in embedded format. Patients can also communicate to their concerned doctors for online consultation. The patient can also include their relatives who are patients. They can view information on drugs in the drug directory section.

Actor - Doctor:

A doctor registers into the system by providing contact number, password, other credentials along with certification. The system verifies the certification and gives access to the doctor. The doctor will be able to search for individual patients and review their reports. They can also provide feedback to patients online, view the drug directory and manage patient lists under their supervision.

Here, all the nouns and verbs are marked where every noun has the potential to become a class and every verb has the potential to become a function. There are mainly 7 categories a class can manifest into. Those are:

1. External entities
2. Things
3. Occurrences Or Events
4. Roles
5. Organizational Units
6. Places
7. Structures

Class Analysis:

The nouns / potential classes are shown below:

No.	Name	Satisfies No.	Status
01	Patient	1, 2, 4, 5, 7	Accepted
02	System	2, 4, 7	Accepted
03	Username	2, 5	Rejected
04	Password	2, 5	Rejected
05	Email	2, 5	Rejected
06	Date of Birth	2, 5	Rejected
07	Contact Number	2, 5	Rejected
08	Medical Report	2, 5, 7	Accepted
09	Database	2, 3, 7	Rejected
10	Doctor	1, 2, 4, 5, 7	Accepted
11	Online Consultation	3	Rejected
12	Relative	1, 2, 5, 7	Rejected
13	Drug	2, 5, 7	Accepted
14	Certification	2, 5	Accepted
15	Feedback	3, 7	Rejected

Only 6 of them are accepted as classes. The rejected ones fall into the accepted ones systematically.

Class Declaration Table:

1. Patient	
ID Name Address Contact No. Password Date of Birth Image	Init(): self Registration() Login() Logout() Upload() Delete() SearchDoctor() ViewDrug() Chat()

2. System	
Name Counter Status Timestamp	Init(): self Verify() Control() Authenticate()

3. Medical Report	
ID Create Date Modify Date Report File	Init(): self Modify() Upload() Delete() View()

4. Doctor	
ID Name Address Contact No. Password Date of Birth Image	Init(): self Registration() Login() Logout() SearchPatient() ViewReport() ViewDrug() Chat()

5. Drug	
ID Brand Name Type Strength Generic Manufacturer Dosage Form	Init(): self ViewDrug() AddDrug() DeleteDrug()

6. Certification	
ID Doctor Name Type Certification Date	Init(): self Verify() Check()