Software Requirement Specification(SRS)
Hospital Management System

Project SRS submitted to : Dr. Varun Srivastava

Submitted by (2CS11) :

- Pareesh Sharma (102116092)
- Vimlendu Sharma (102166004)
- Ajayveer Dhillon (102166001)

1.1 - Purpose (Functional Requirements)

- The Software is for the automation of Hospital Management.
- It(Software) makes it easy for users and administrators to access and use the Hospital System to the fullest.

It maintains two levels of users:-

i. Administrator Level

ii. User Level

The Software includes:-

- ➡ Maintaining Patient details.
- ➡ Providing Prescription, Precautions and Diet advice.
- ➡ Providing and maintaining all kinds of tests for a patient.
- ➡ Billing and Report generation.

1.2 - Scope

It can be used in any Hospital, Clinic, Dispensary or Pathology labs for maintaining patient details and their test results.

1.3 - Technologies to be used

This project will be a desktop application to be developed in VS Code having Oracle/Php as backend.

- Database Design (Oracle/MySQL)
- Database Language (SQL)
- Coding Language (SQL, NodeJs, JavaScript, HTML)
- Testing (VS Code and Oracle)
- Reporting Tool (Data Report)

1.4 - Overview

Project is related to Hospital Management System.

The project maintains two levels of users:-

- Administrator Level-Doctor
- User Level-Data Entry Operator

Main facilities available in this project are:-

- Maintaining records of indoor/outdoor patients.
- Maintaining patients diagnosis details, advised tests to be done.
- Providing different test facilities to a doctor for diagnosis of patients.
    - ➡ X-Ray
    - ➡ Urine Test
    - ➡ Stool Test
    - ➡ Blood Test
    - ➡ Biochemistry Test

- Maintaining patient's injection entry records.
- Maintaining patient's prescription, medicine and diet advice details.
- Providing billing details for indoor/outdoor patients.
- Maintaining backup of data as per user requirements (between mentioned dates).
- If user forgets his/her password then it can be retrieved by hint question.
- Results of tests, prescription, precautions and diet advice will be automatically updated in the database.
- User or Administrator can search a patient's record by his/her name or their registration date.

Overall Description
2.1- Goals of proposed system
(Non Functional requirements)

1. **Planned approach towards working: -** The working in the organisation will be well planned and organised. The data will be stored properly in database, which will help in retrieval of information as well as its storage.
2. **Accuracy: -** The level of accuracy in the proposed system will be higher. All operations would be done correctly and it ensures that whatever information is coming from the centre is accurate.
3. **Reliability: -** The reliability of the proposed system will be high due to the above stated reasons.

Overall Description
2.1- Goals of proposed system

4. **No Redundancy: -** In the proposed system utmost care would be that no information is repeated anywhere, in storage or otherwise. This would assure economic use of storage space and consistency in the data stored.

5. **Easy to Operate: -** The system should be easy to operate and should be such that it can be developed within a short period of time and fit in the limited budget of the user.

6. **Responsive:-** The website/application should be responsive that means it should work fine with almost any browser or in any screen size/resolution.

Overall Description
2.2- Background

•A Hospital is a place where Patients come up for general diseases.
Hospitals provide facilities like:-

- Consultation by Doctors on Diseases.
- Diagnosis for diseases.
- Providing treatment facility.
- Facility for admitting Patients (providing beds, nursing, medicines etc.)
- Immunisation for Patients/Children.

•Various operational works that are done in a Hospital are:-

- Recording information about the Patients that come.
- Generating bills.
- Recording information related to diagnosis given to Patients.
- Keeping record of the Immunisation provided to children/patients.
- Keeping information about various diseases and medicines available to cure them.

Overall Description
2.3 – Project Requirements

Hardware Requirements

Processor	RAM	Disk Space
Any processor capable of doing basic things like Intel i3,i7 Mac M1 or higher.	1 GB or higher	256 GB or higher

Software Requirements

Operating System	Database
Win XP, Win 98 or higher . Linux or MacOS	Oracle or Php

Overall Description

2.4) User Characteristics

Every user should be:

- Comfortable of working with computer.
- He must have knowledge in medical field.
- He must also have basic knowledge of English too.

2.5) Constraints

- GUI is only in English.
- Login and password is used for identification of user and there is no facility for guest.

2.6- Definitions of problems

Problems with conventional system :-

1. **Lack of immediate retrievals: -**The information is very difficult to retrieve and to find particular information like- E.g. - To find out about the patient's history, the user has to go through various registers. This results in inconvenience and wastage of time.
2. **Lack of immediate information storage: -** The information generated by various transactions takes time and efforts to be stored at right place.
3. **Lack of prompt updating: -** Various changes to information like patient details or immunisation details of child are difficult to make as paper work is involved.
4. **Error prone manual calculation: -** Manual calculations are error prone and take a lot of time this may result in incorrect information. For example calculation of patient's bill based on various treatments.
5. **Preparation of accurate and prompt reports: -** This becomes a difficult task as information is difficult to collect from various registers.

Front-end selection:

1. It must have a GUI that assists employees that are not from IT background.
2. Scalability and extensibility.
3. Flexibility.
4. Robustness.
5. According to the organisation requirement and the culture.
6. Must provide excellent reporting features with good printing support.
7. Platform independent.
8. Easy to debug and maintain.
9. Event driven programming facility.
10. Front end must support some popular back end like Oracle/MySQL.

-> According to the above stated features we selected HTML and JS as the front-end for developing our project.

Back-end Selection:

1. Multiple user support.
2. Efficient data handling.
3. Provide inherent features for security.
4. Efficient data retrieval and maintenance.
5. Stored procedures.
6. Popularity.
7. Operating System compatible.
8. Easy to install.
9. Various drivers must be available.
10. Easy to implant with the Front-end.

    -> According to above stated features we selected Oracle/MySQL as the backend.

Activity Diagram (Without swim lane) :-

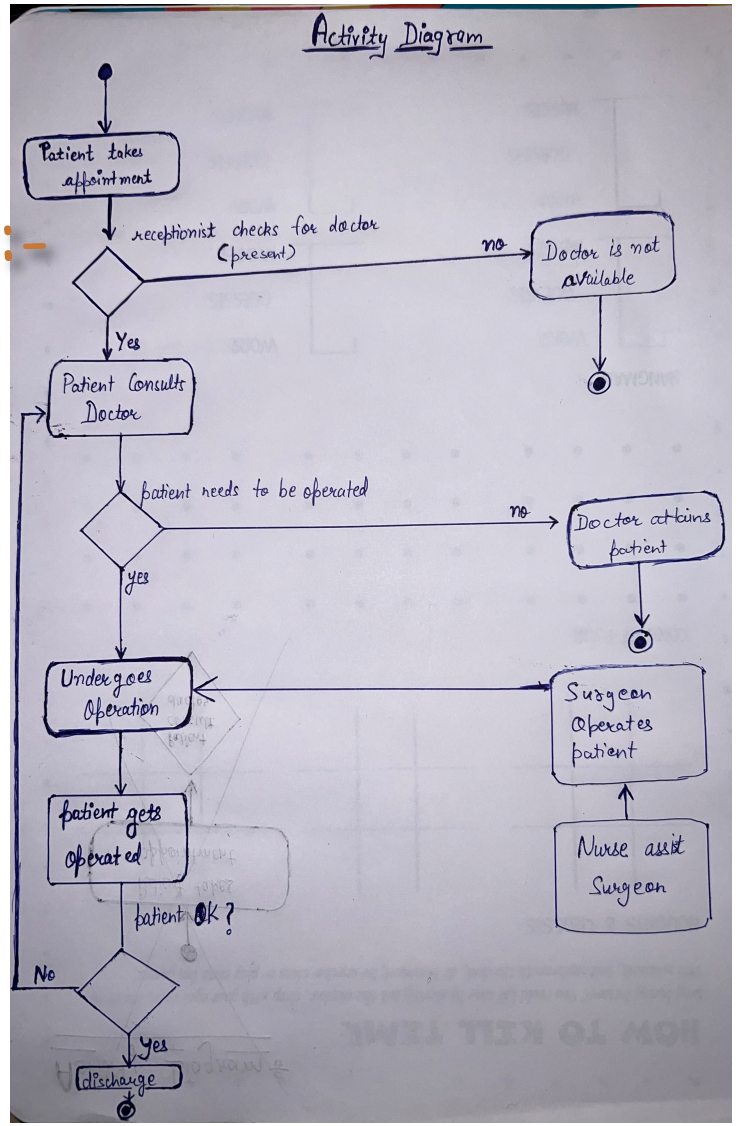
Representation :

◆ - Decision Block

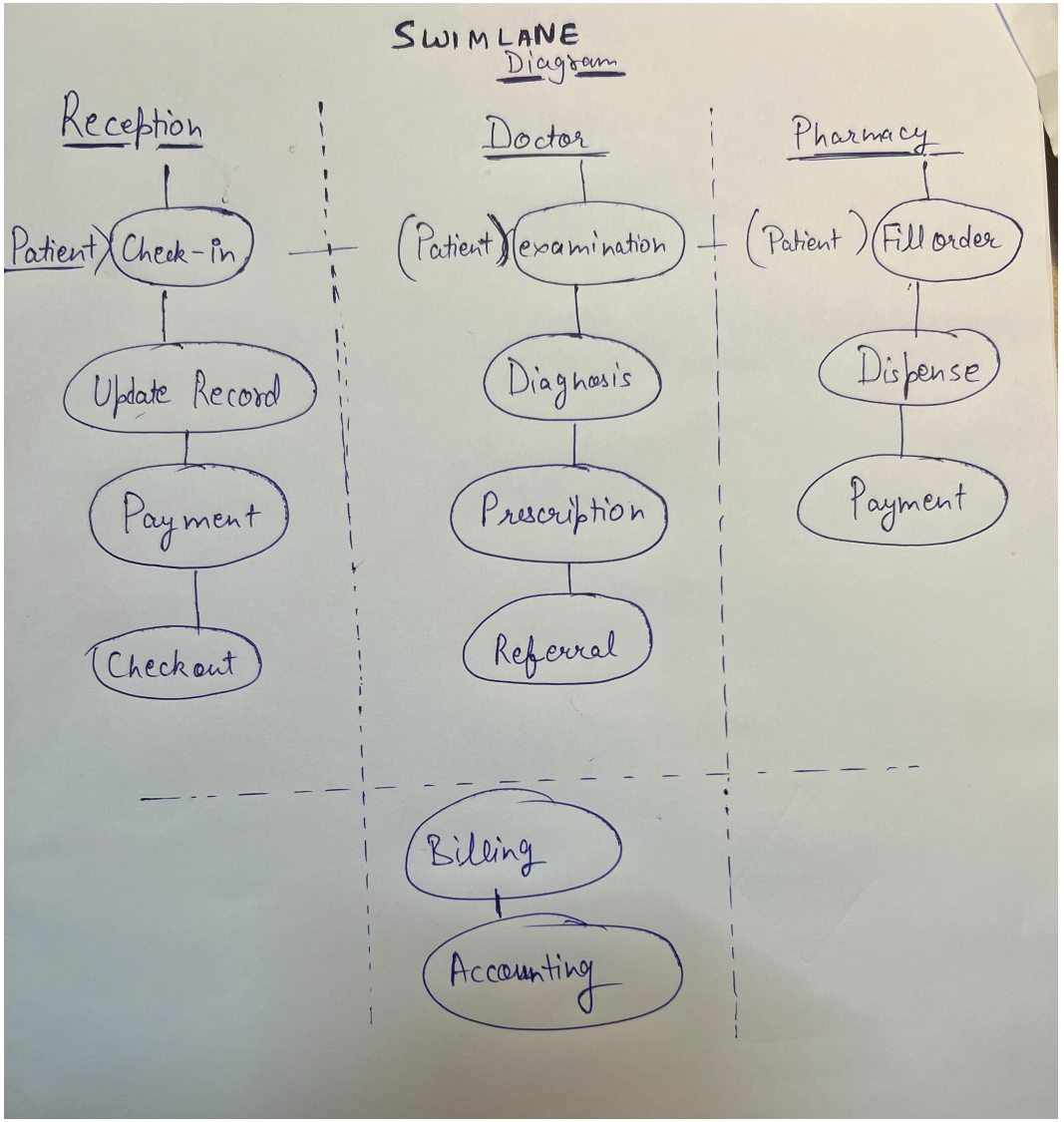
■ - Activity/Action block

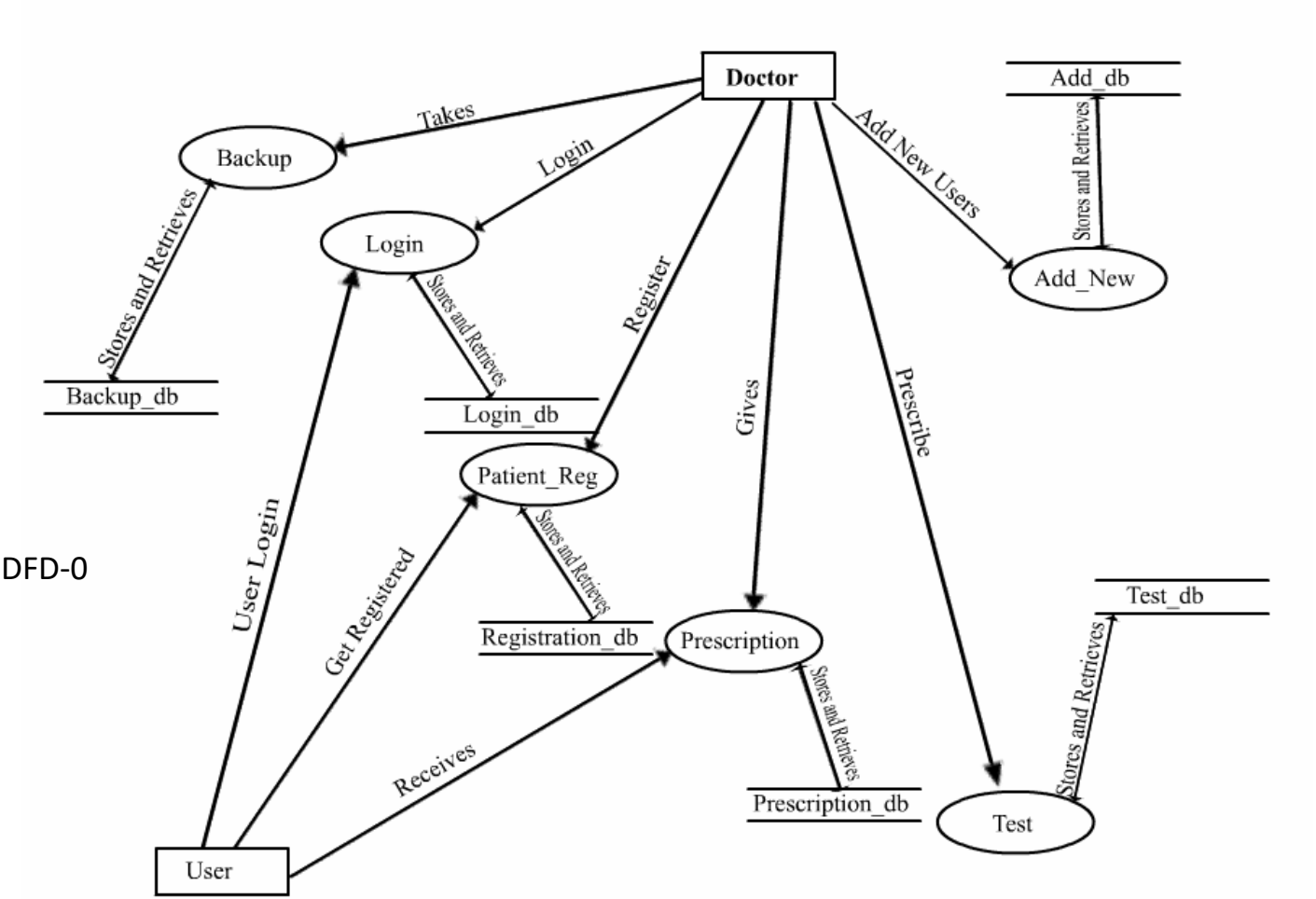
● - Initial state

◉ - Final State



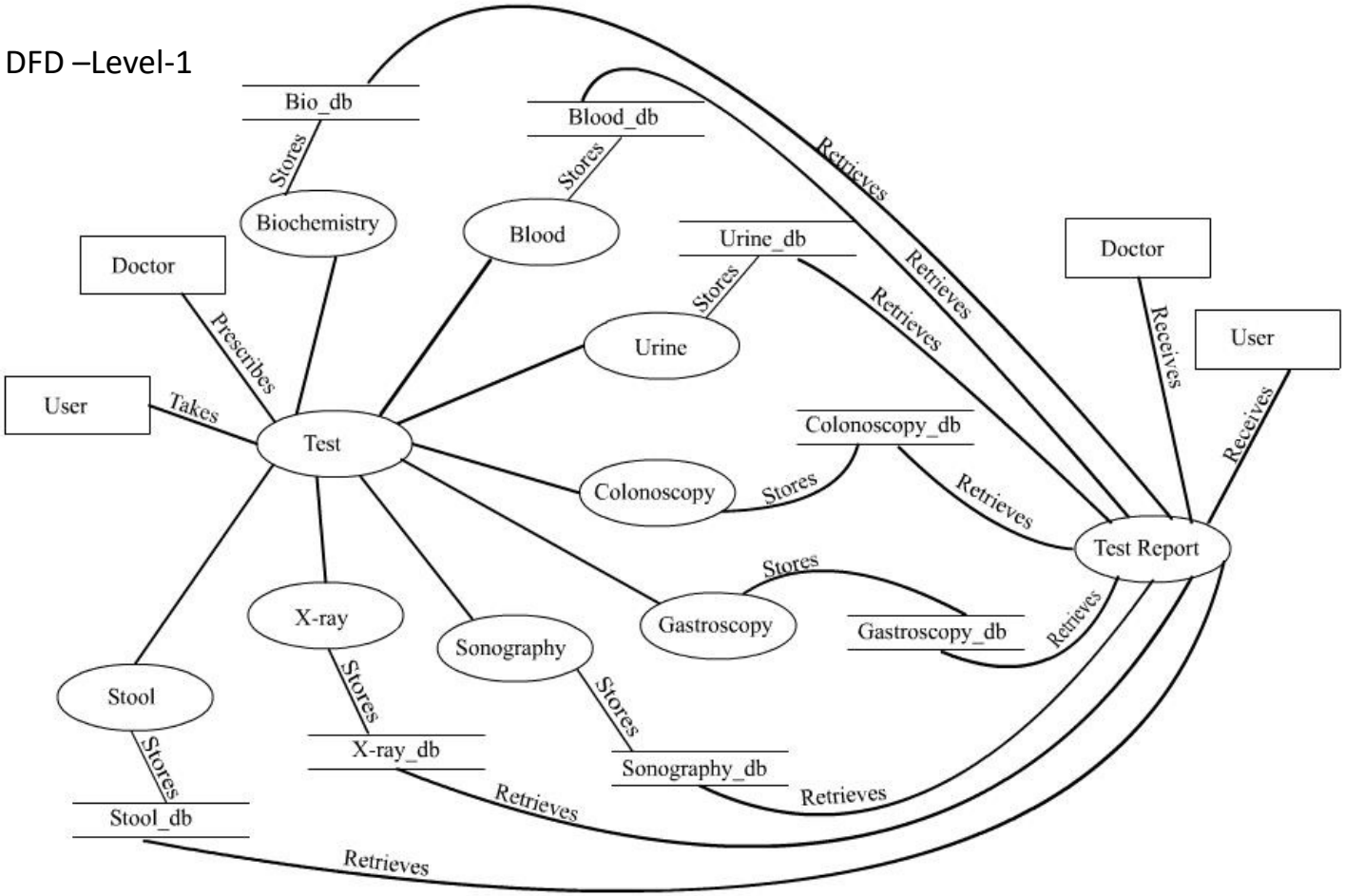
Swim lane Diagram :-



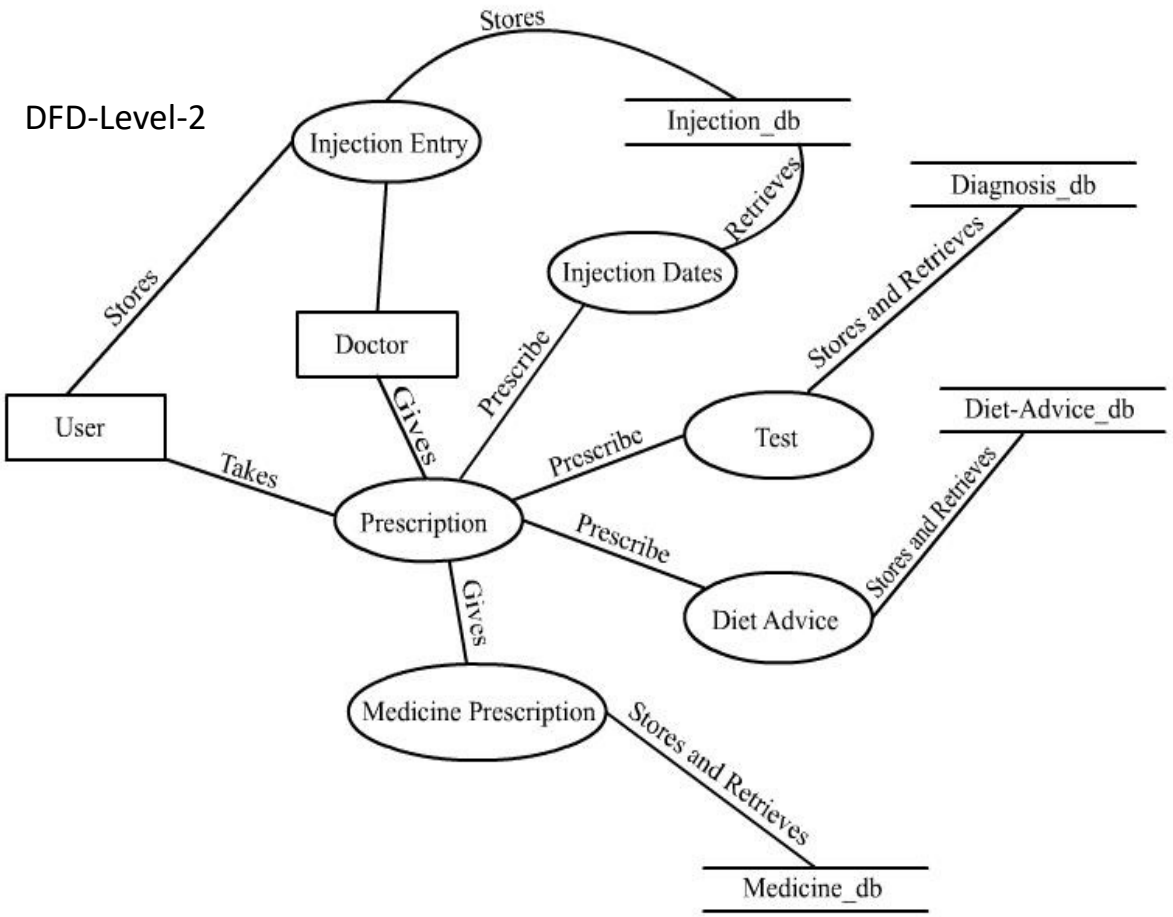

Doctor
Takes
Backup
Login
Add New Users
Add_db
Stores and Retrieves
Add_New
Stores and Retrieves
Backup_db
Login
Stores and Retrieves
Login_db
Register
Gives
Prescribe
Patient_Reg
User Login
Get Registered
Stores and Retrieves
Registration_db
Prescription
Stores and Retrieves
Prescription_db
Test_db
Stores and Retrieves
Test
Receives
User

DFD-0

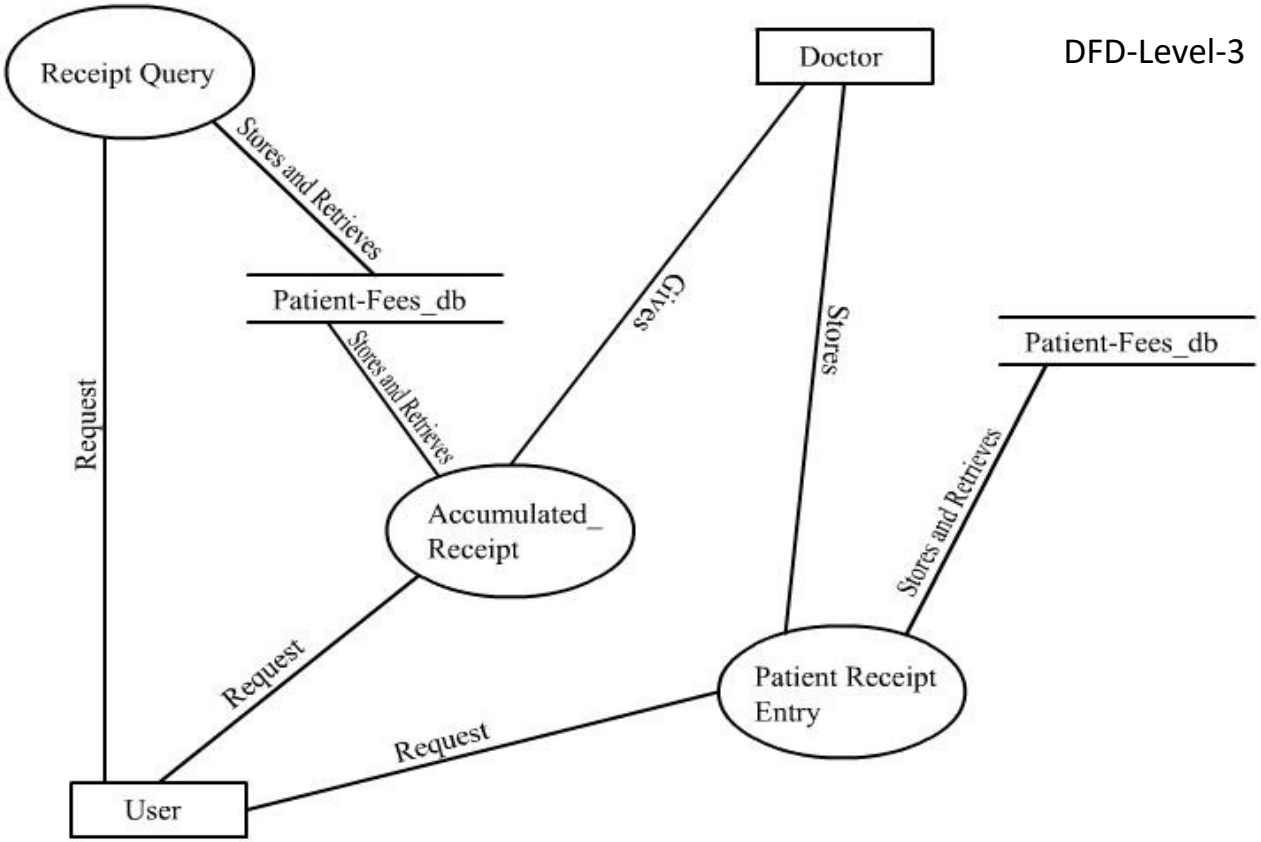
DFD –Level-1


Bio_db
Blood_db
Stores
Stores
Biochemistry
Blood
Urine_db
Doctor
Doctor
Stores
Prescribes
Urine
Retrieves
Retrieves
Retrieves
Receives
User
User
Takes
Test
Colonoscopy_db
Receives
Colonoscopy
Stores
Retrieves
Test Report
Stores
X-ray
Gastroscopy
Gastroscopy_db
Retrieves
Sonography
Stool
Stores
Stores
X-ray_db
Sonography_db
Stores
Retrieves
Retrieves
Stool_db
Retrieves

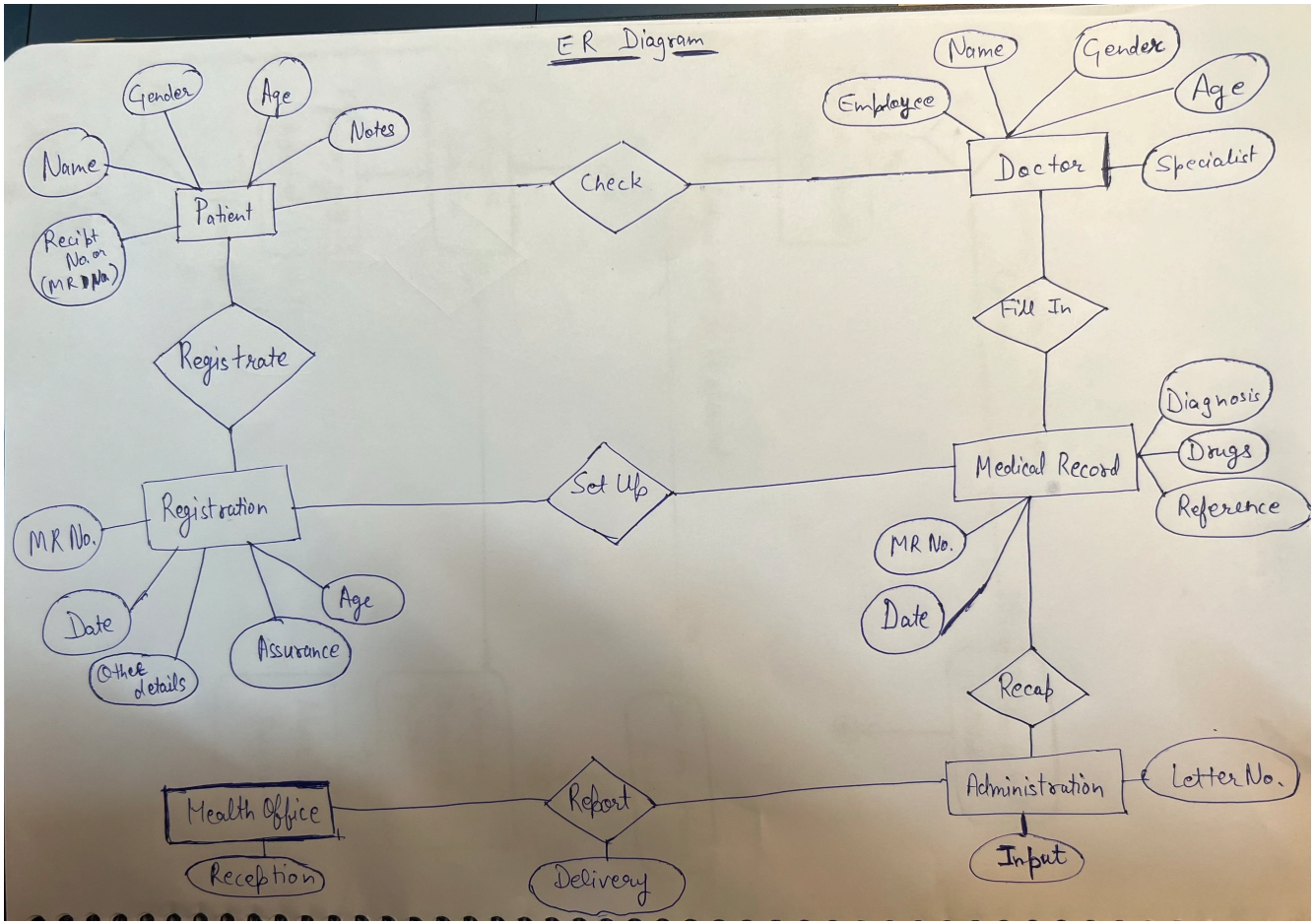
DFD-Level-2

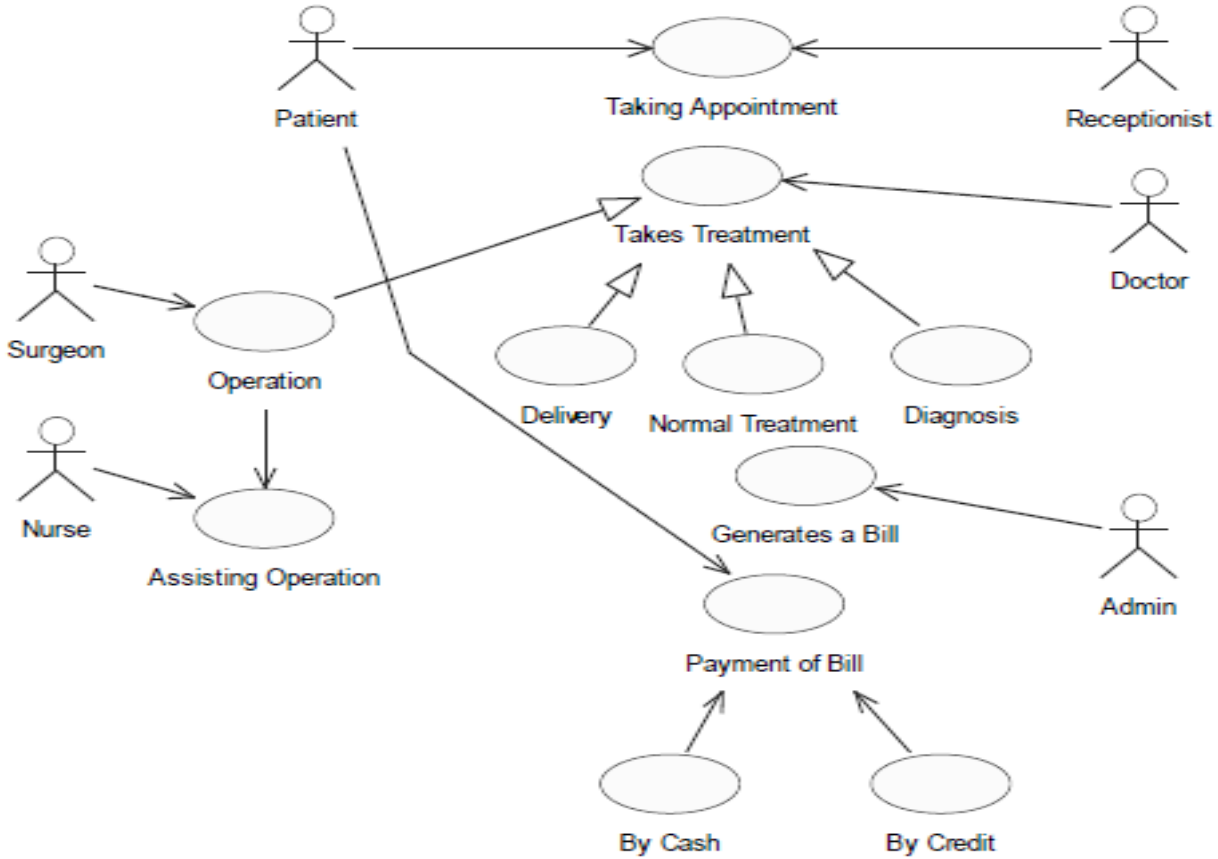

Stores
Injection Entry
Injection_db
Retrieves
Diagnosis_db
Stores
Injection Dates
Stores and Retrieves
Doctor
Prescribe
User
Gives
Test
Diet-Advice_db
Takes
Prescribe
Prescription
Prescribe
Stores and Retrieves
Gives
Diet Advice
Medicine Prescription
Stores and Retrieves
Medicine_db

DFD-Level-3


Receipt Query
Doctor
Stores and Retrieves
Patient-Fees_db
Gives
Stores
Patient-Fees_db
Request
Stores and Retrieves
Accumulated_
Receipt
Stores and Retrieves
Request
Patient Receipt
Entry
Request
User

ER Diagram :-

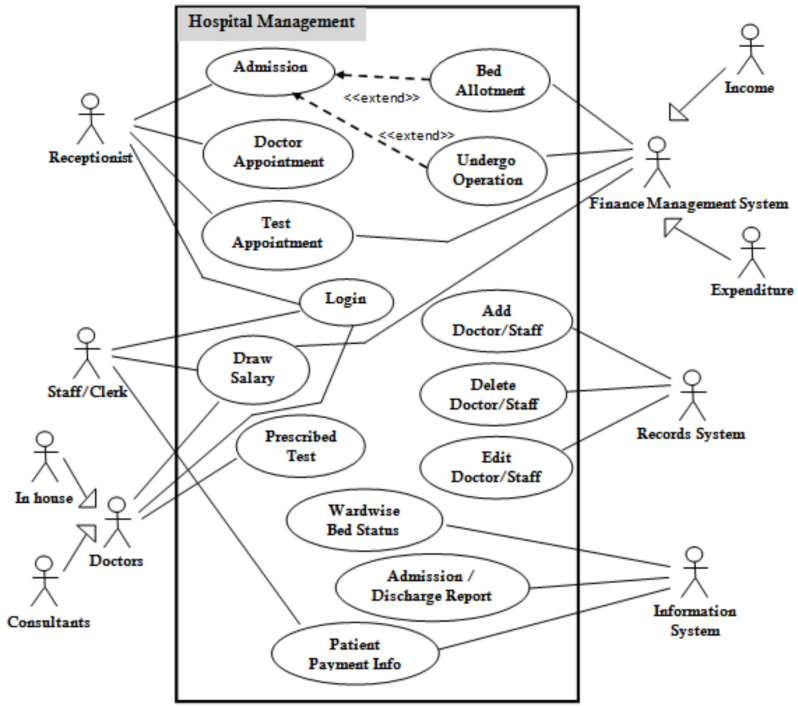

ER Diagram
Name
Gender
Age
Employee
Doctor
Specialist
Check
Patient
Notes
Name
Recipt No. or (MR No.)
Fill In
Registrate
Diagnosis
Drugs
Reference
Medical Record
Set Up
Registration
MR No.
MR No.
Date
Age
Date
Assurance
Other details
Recap
Letter No.
Administration
Report
Health Office
Input
Reception
Delivery



Use –Case Diagram

Use case scenario

The main difference b/w Use case diagram and use case scenario is that use case diagram provides top view of the system whereas use case scenario provides a detailed view of the system taken into consideration



Use Case Template

Use case name :	Schedule Appointment
Actors :	Patient, Receptionist, Doctor
Preconditions :	Patient must be registered in the system and the doctor's schedule must be available.

Use Case Template Continued..

Basic Flow:

1. The patient requests an appointment with a specific doctor.
2. The receptionist checks the doctor's schedule to find an available time slot.
3. The receptionist schedules the appointment for the patient.
4. The system sends a confirmation to the patient.

Postconditions: The appointment is scheduled and confirmed.

Alternate Flows:

- ➡ If the doctor is not available, the receptionist suggests another doctor or a later time.
- ➡ If the patient's preferred time is not available, the receptionist suggests an alternate time.

➡ Exceptions:

- ➡ If the patient is not registered in the system, the receptionist assists them with the registration process.
- ➡ If the doctor's schedule is not available, the receptionist informs the patient and suggests a later time.

Thank You!