Project Report
On
“Hospital Management System”

Submitted for partial fulfillment of requirement for the award of degree
Of

Kalyani Government Engineering College
Kalyani, Nadia, West Bengal, India. Pin-741235.

Session 2019-22 (4th Semester 2nd Year)

Guided by

Dr. Indrajit Bhattacharya

Submitted By

INDRAJIT PAL	**(10201019027)**
NEHA CHOWDHURY	**(10201019019)**
SAURABH DAS	**(10201019011)**
SUBRATA HALDER	**(10201019005)**

Kalyani Government Engineering College

CERTIFICATE

This is to certify that

Mr./ Ms. **INDRAJIT PAL** Enrollment No. **10201019027**, Mr./ Ms. **NEHA CHOWDHURY** Enrollment No**. 10201019019**, Mr./ Ms**. SAURABH DAS** Enrollment No**. 10201019011**, and Mr./ Ms. **SUBRATA HALDER** Enrollment No**. 10201019005** Branch **MCA** semester **4th** has satisfactorily completed the report.

INDEX

CHAPTER - 1
INTRODUCTION

Hospital Management System (HMS) is powerful, flexible, and easy to use system, which is very helpful in a hospital environment, in order to maintain a hospital efficiently. Hospital Management System designed for multi-specialty hospitals, to cover a wide range of hospital administration and management processes. It is an integrated end-to-end Hospital Management System (HMS) that provides relevant information across the hospital to support effective decision making for patient care, hospital administration and critical financial accounting, in a seamless flow.

In a HMS need to have entry of all the data related to patients, doctors, staffs, and administrator(s). Each patient admitted in the hospital needs to enter their personal information, which are accessible by authorized doctors and staffs. On the other hand, a patient should have access permission to download any medical report associated with him/her. Billing process should be included in the system, such that it can easily be update after each payment.

1.1 Purpose

The software is for the automation of Hospital Management. It maintains two levels of users

1. Administrator
2. Receptionist
3. User

This software includes maintaining patients' details, doctors' details, employees' details, providing prescriptions, providing and maintaining all kind of tests for a patient. And also provide billing and report generation.

1.2 Scope

- The Proposed Software product is the hospital Management System (HMS). The system will be used to get the information from the patients and then sorting the data for future usages.
- The current in use is a paper based system. It is too slow and cannot provide updated list of patients within a reasonable time-frame.
- The intention of the system are to reduced over-time pay and increase the number of patients that can be treated accurately.
- Requirements statement in this document are both functional & non-functional.

1.3 Overview

- This software requirements specification (SRS) is the requirements work product that formally specifies Hospital Management System (HMS).
- It includes the results of both business analysis and system analysis efforts various techniques where use to elicit the requirements and we have identified your needs, analyzed and refined them.
- The objective of this document therefore is to formally describe the system's high level requirements including functional requirements, non-functional requirements and business rules and constraints.

1.4 Constraints

- All the users must register into the system though the web interface.
- All the users must login into the web interface through userID (must be unique) and password.
- Patients should allow to get login him/her-self in order to collecting own information.
- On login into the system, a doctor must be able to see the patients list under his/her observation.
- One patient must not access the information of some other patients.
- All the patients must be able to download all the medical report and bill as per his .requirement by login into the system.
- Add, delete, and update must be reflect immediately in the system.

CHAPTER – 2
REQUIREMENTS

2.1. Functional Requirements

2.1.1. Description

❖ **Registration**

- **Add Patients:-**
  The **HMS** shall allow receptionist to add new patients to the system.

- **Assign Patient ID:-**
  The HMS shall allow receptionist to give each patient an ID and add it to the patient's record. This ID shall be used by the patient throughput his/her stay in hospital.

- **Delete Patient ID:-**
  The HMS shall allow receptionist to delete the ID of the patient from the system when the patient checks out.

- **Add Doctors:-**
  The administrative staff in the ward shall be allowed to add new doctors to the system.

- **Assign Doctor ID:-**
  The administrative staff in the ward shall be allowed to give each doctor an ID and add it to the doctor's record.

- **Delete Patient ID:-**
  The administrative staff in the ward shall be allowed to delete the ID of the Doctor from the system when the doctor resign.

- **Add Employee:-**

  The administrative staff in the ward shall be allowed to add new doctors to the system.

- **Assign Employee ID:-**

  The administrative staff in the ward shall be allowed to give each employee an ID and add it to the doctor's record.

- **Delete Employee ID:-**

  The administrative staff in the ward shall be allowed to delete the ID of the employee from the system when the employee resign.

- **Add to beds-available list:-**

  The HMS shall allow ----- to put the beds just evacuated in beds-available list.

❖ **<u>Report Generation</u>**

- **Patient information:-**

  The HMS shall generate report on patients about the following information: patient's name, patient's ph no., ward name and the doctor's name which was assigned.

- **Bed Availability:-**

  The HMS shall generate report on bed availability about the following information: ward name, bed number, occupied/unoccupied.

- **Patient Bill:-**

  The HMS shall generate bills of patient.

- **Lab Test Report:-**

  The HMS shall generate Lab test reports of patient.

❖ **<u>Database</u>**

- **Patient Mandatory Information:-**

  Each patient shall have the following mandatory information: first name, last name, phone number, date of birth, gender, address, city, pincode, patient ID.

- **Update Patient Information:**

  The HMS shall allow the receptionist to update any of the patient's information.

- **Doctor Mandatory Information:-**

  Each doctor shall have the following mandatory information: first name, last name, phone number, email, gender, address, city, pincode, doctor ID, department, specialist, cabin, salary

- **Update Doctor Information:**

  The administration staff in the ward shall be allowed to update any of the doctor's information.

- **Employee Mandatory Information:-**

  Each employee shall have the following mandatory information: first name, last name, phone number, email, gender, department, designation, address, city, pincode, patient ID.

- **Update Employee Information:**

  The administration staff in the ward shall be allowed to update any of the employee's information.

2.1.2. Technical issues

- **Database:-**
  The system shall use the MYSQL Database, which is open source and free.

- **Operating System:-**
  The Development environment shall be Windows.

- **Web-Based:-**
  The system shall be a Web-based application.

2.2. Non-Functional Requirements:

2.2.1. Security

- **Patient Identification:-**
  The system requires the patient to identify himself/herself using phone number.

- **Login ID:-**
  Any user who uses the system shall have a Login ID and Password.

- **Modification:-**
  Any modification (insert, delete, update) for the Database shall be synchronized and by the administrator in the ward.

- **Front Desk Staff (Receptionist) Rights:-**
  Receptionist shall be able to view all information in HMS, add new patients to HMS but shall not be able to modify any information in it except patient's information.

- **Administrator's Rights:**

  Administrators shall be able to view and modify all information in HMS.

2.2.2. Reliability

How general the form generation language is Simplicity vs. functionality of the form language = Speeds up form development but does not limit functional.

2.2.3. Availability

The system should be available all the time.

2.2.4. Safety

Humans are error-prone, but the negative effects of common error should be limited. E. g. user should realize that a given command will delete data, and be asked to confirm their intent or have the option to undo.

2.2.5. Software Quality

Good quality to the framework = product robust, bug free software which contains all necessary requirements Customer satisfaction.

2.2.6. Reusability

Is part of the code going to be used elsewhere = produces simple and independent code modules that can be reused.

2.2.7 Maintainability

- **Back Up:-**

  The system shall provide the capability to back-up the Data

- **Errors:-**

  The system shall keep a log of all the errors.

2.3 Hardware Requirements

- Standard pc
- Internet connection with good enough speed
- ATM
- Pentium IV 1.7 GHz class or better processor
- 128MB or more RAM (256 recommended)
- At least 500 MB Hard disk space.
- Smart mobile phone

2.4 Software Requirements

Prerequisites

- Install WAMP
- Any Editor (Preferably VS Code or Sublime Text)
- Any web browser with latest version

Languages and Technologies used

- HTML5/CSS3
- JavaScript (to create dynamically updating content)
- Bootstrap (An HTML, CSS, and JS library)
- WAMP (Windows Apache MySQL Php)
- PHP (PHP: Hypertext Preprocessor)
- MySQL (An RDBMS that uses SQL)
- TCPDF (to generate PDFs)

CHAPTER - 3
WATERFALL MODEL

The waterfall model is a sequential design process, often used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Analysis, Requirement Specification, Design, Implementation, Testing and Integration and Operation and Maintenance.

If in the beginning of the project failures are detected, it takes less effort (and therefore time and money) for this error. In the waterfall model phases to be properly sealed first before proceeding to the next stage. It is believed that the phases are correct before proceeding to the next phase. In the waterfall model lay the emphasis on documentation.

It is a straightforward method. The way of working ensures that there are specific phases. This tells you what stage it is. One can use this method of milestones. Milestones can be used to monitor the progress of the project to estimate.

In our Project, all the requirements are clear and well known and the project is large. All the activities in our project are carried out in above mentioned phases of waterfall model.

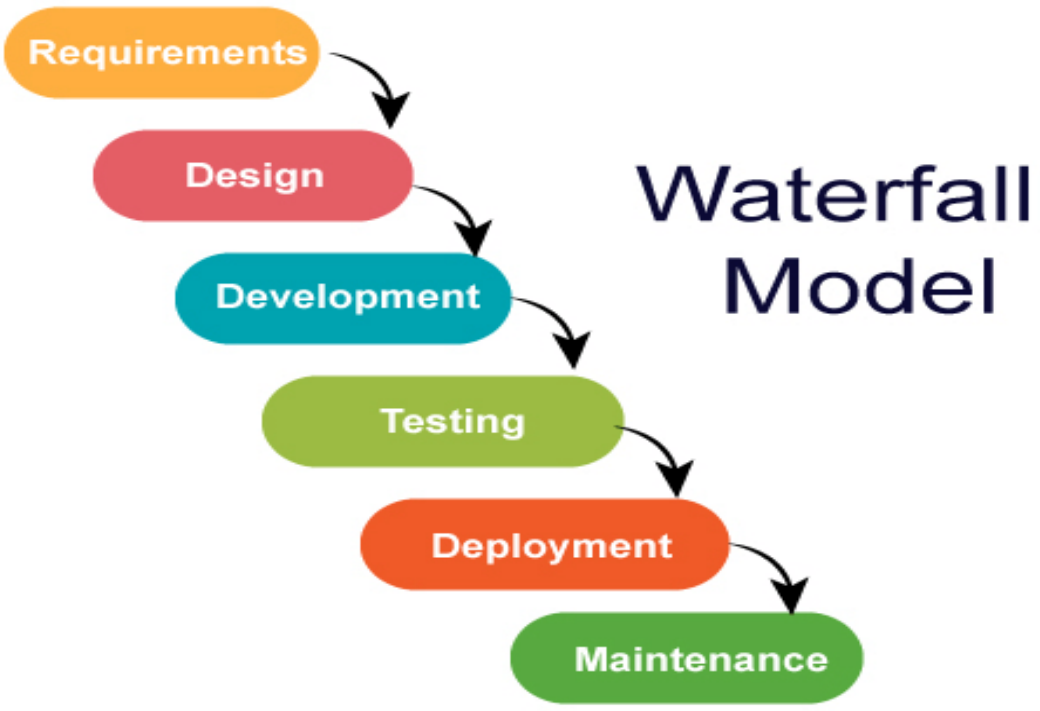


Fig: WATERFALL MODEL

CHAPTER - 4
FEASIBILITY STUDY

The prime focus of the feasibility is evaluating the practicality of the proposed system keeping in mind a number of factors. The following factors are taken into account before deciding in favor of the new system.

4.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require. The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.
- Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available, it gives an indication of the system is economically possible for development.

4.2 TECHNICAL FEASIBILITY

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed. Technical issues raised during the investigation are: Does the existing technology sufficient for the suggested one?
Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project is developed within latest technology.

Through the technology may become obsolete after some period of time, due to the fact that never version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using Java the project is technically feasible for development.

4.3 OPERATATIONAL FEASIBILITY

This includes the following questions: Is there sufficient support for the users? Will the proposed system cause harm? The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

4.4 SCHEDULE FEASIBILITY

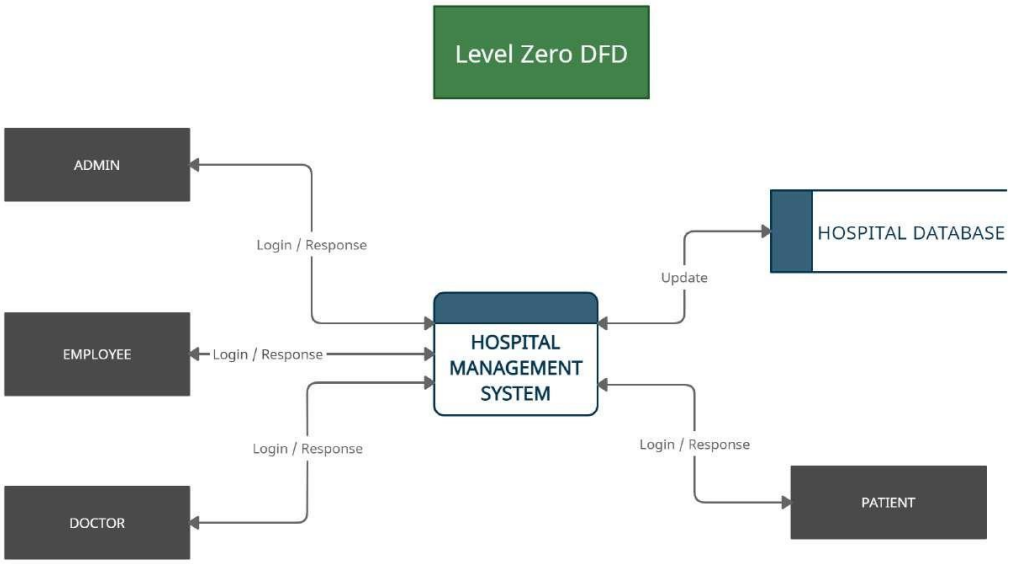
An evaluation of the time needed for the development of this project. The time schedule required for the development of this project is very important, since more development time effects machine time, costs and delays in the development of the other systems. So the project should be complete within affixed schedule time as far as the organization is concerned.

CHAPTER - 5
Data Dictionary

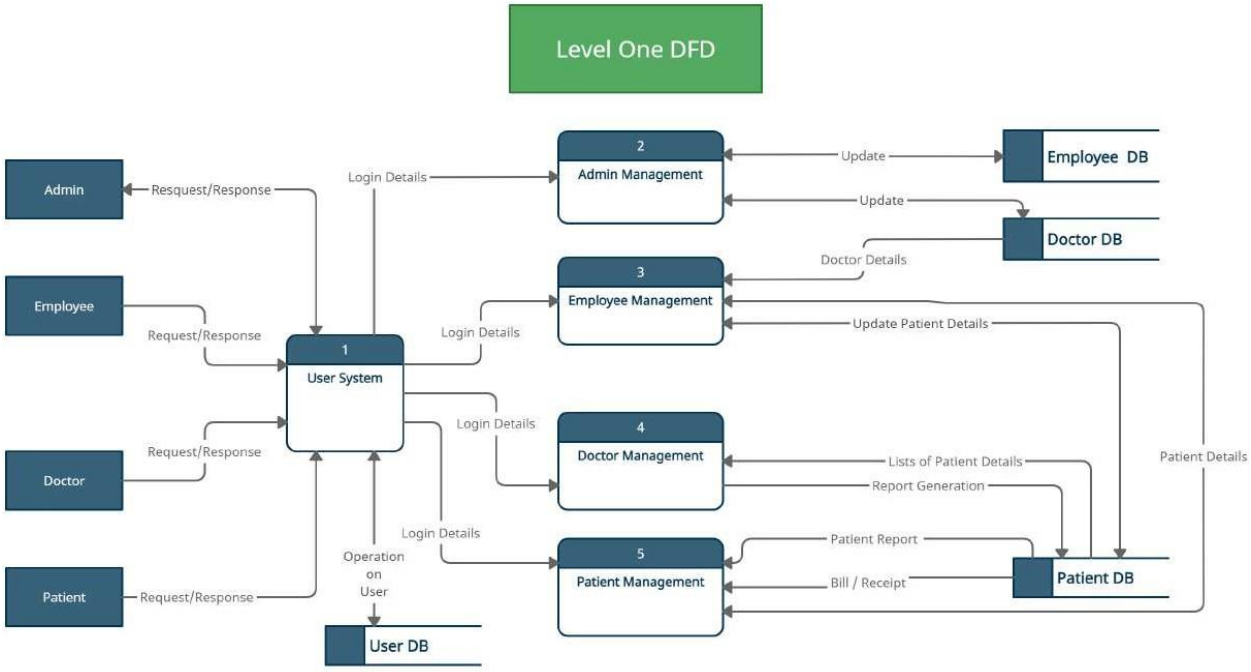
Data dictionary actor description instructions:

Name	Name of the actor EXACTLY as it appears on the use case diagram. It must be a noun or noun phrase with the first letter of the name capitalized.
Alternate Name(s)	Alternative names this actor may be referred to in the application domain. Providing these names helps the readers to understand this document.
Input Data	List of the inputs to the system that this actor provides. This section must contain a list of the use cases with which this actor interacts (has a line on the use case diagram) and provides input. For each use case, list the inputs this actor provides.
Output Data	List of the outputs from the system that this actor receives. This section must contain a list of the use cases with which this actor interacts (has a line on the use case diagram) and receives output. For each use case, list the outputs this actor receives.
Description	Brief description of the general purpose or role of this actor.
Comments	Any additional information that aid in the understanding of this actor.

5.1 Data Flow Diagram

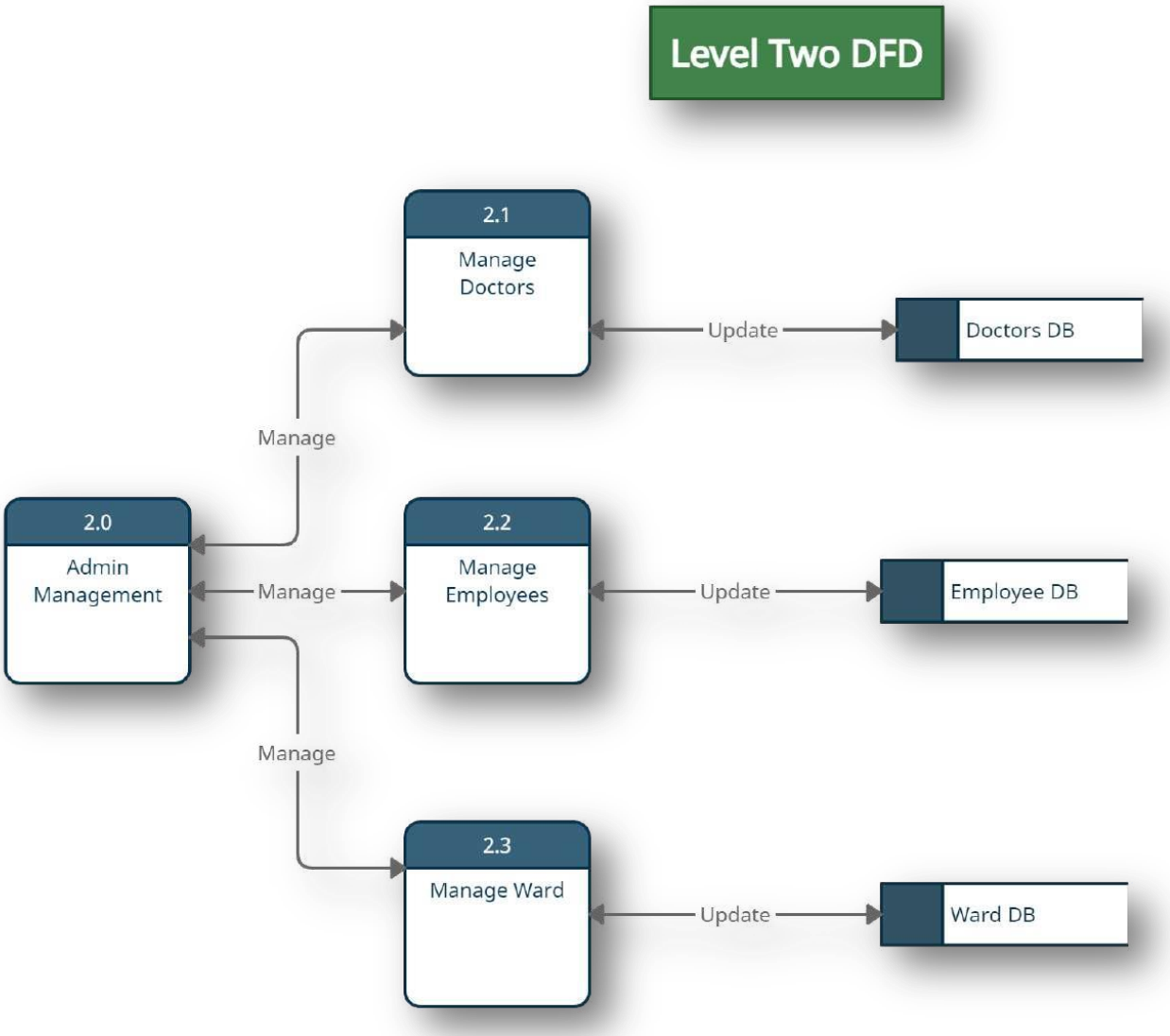


HMS Level Zero DFD (5.1.1)

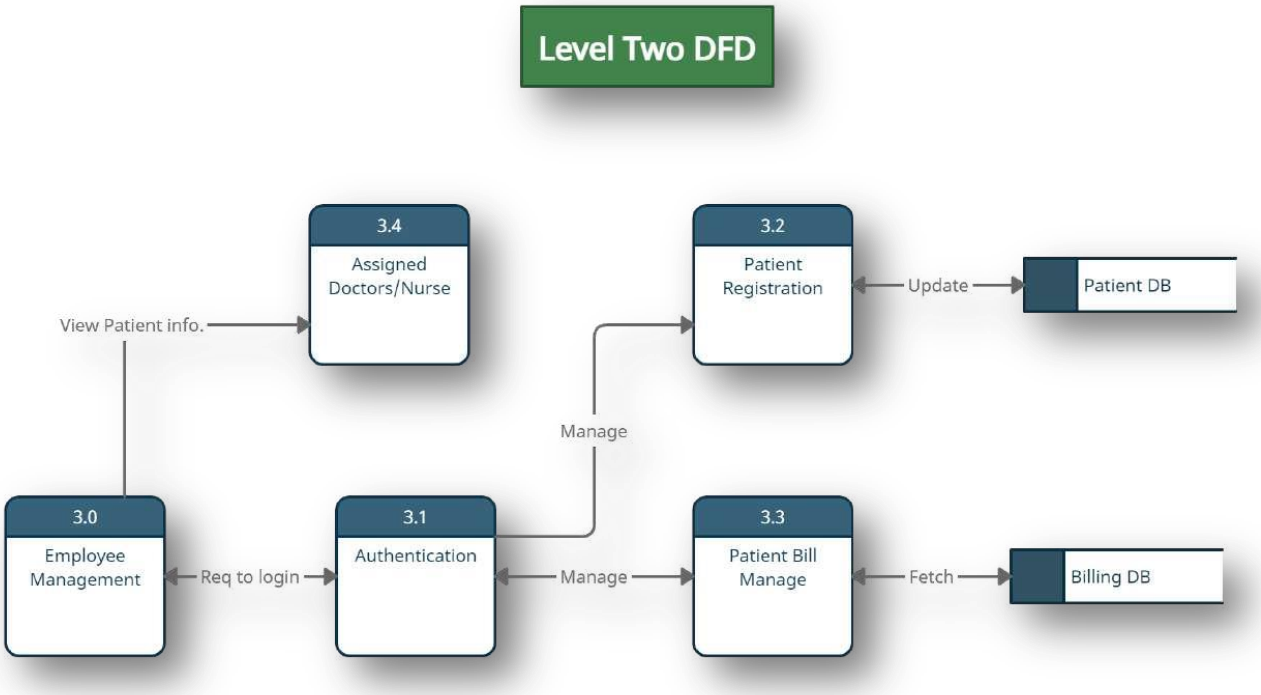


HMS Level One DFD (5.1.2)

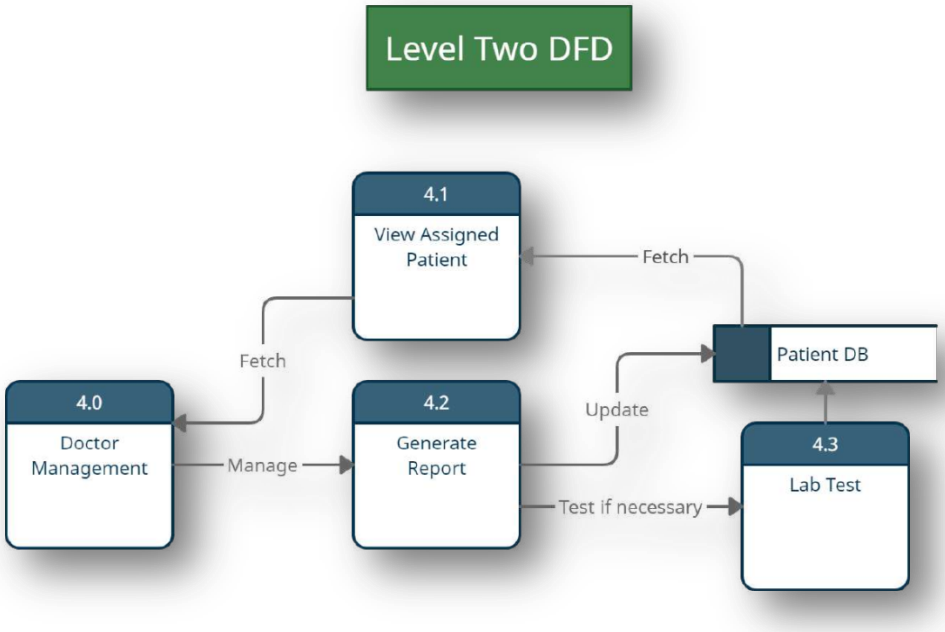
HMS Level two DFD (5.1.3)



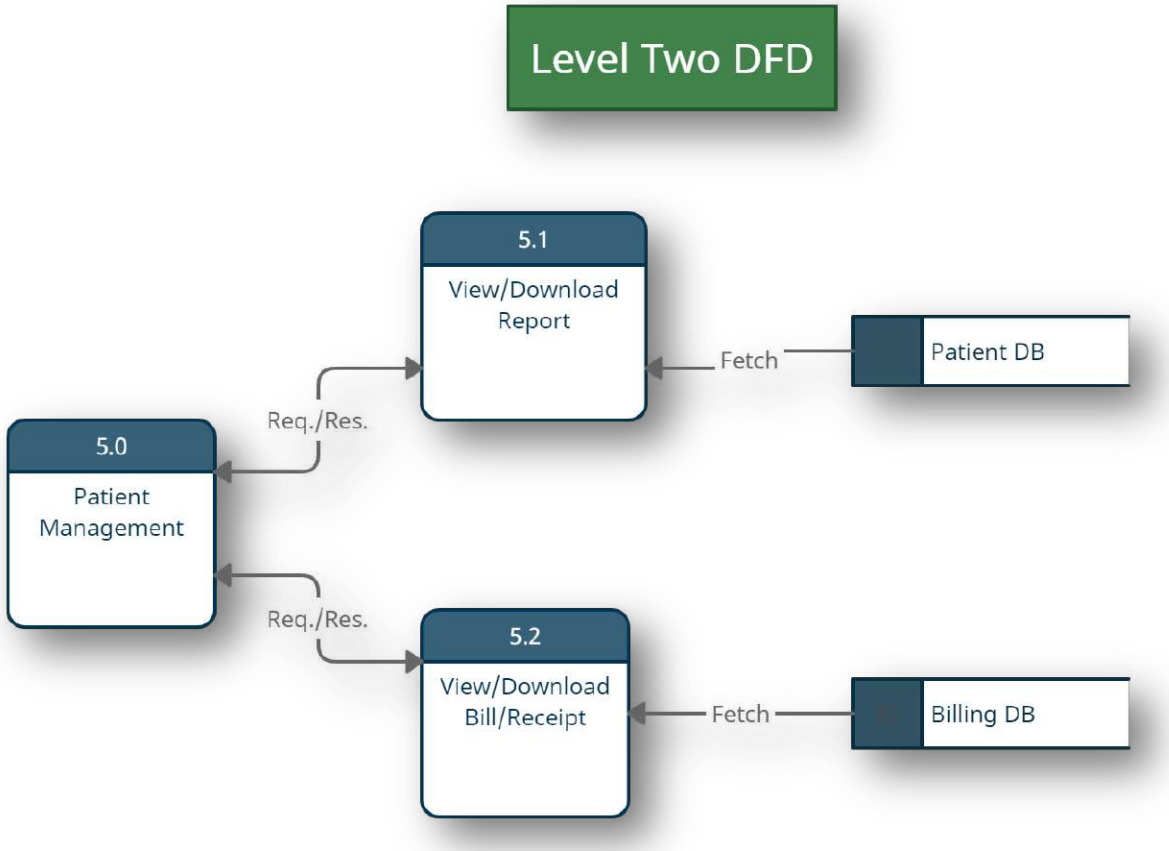
HMS Level Two DFD of Admin Management (5.1.3.1)



HMS Level Two DFD of Employee Management (5.1.3.2)



HMS Level Two DFD of Doctor Management (5.1.3.3)



HMS Level Two DFD of Patient Management (5.1.3.4)

CHAPTER-6
Design and Implementation

- The product is completely data oriented.
- Here, Administrator and Receptionist would input the various details of the patients, employees, doctors etc. for storing, updating, processing or retrieval of data from the database as per the instructions given and display an acknowledging message to the user.
- Login and password is used for identification and there is no facility for non-users to login.
- This system works only on a single server.
- GUI is only in English.
- Limited to HTTP/HTTPS protocols.
- When we consider the priority of this project it is mainly of medium cost, efficient to user access data, provides the required data, safe and secure one. we can know the details of patients whether it may be a outdoor patient or admitted patient.
- Overall view of the Hospital System: The overall view (design and implementation) of the Hospital System is as shown below:

6.1 UML Diagram

UML stands for Unified Modelling Language. It's a rich language to model software solutions, application structures, system behavior and business processes There are 14 UML diagram types to help you model these behaviors.

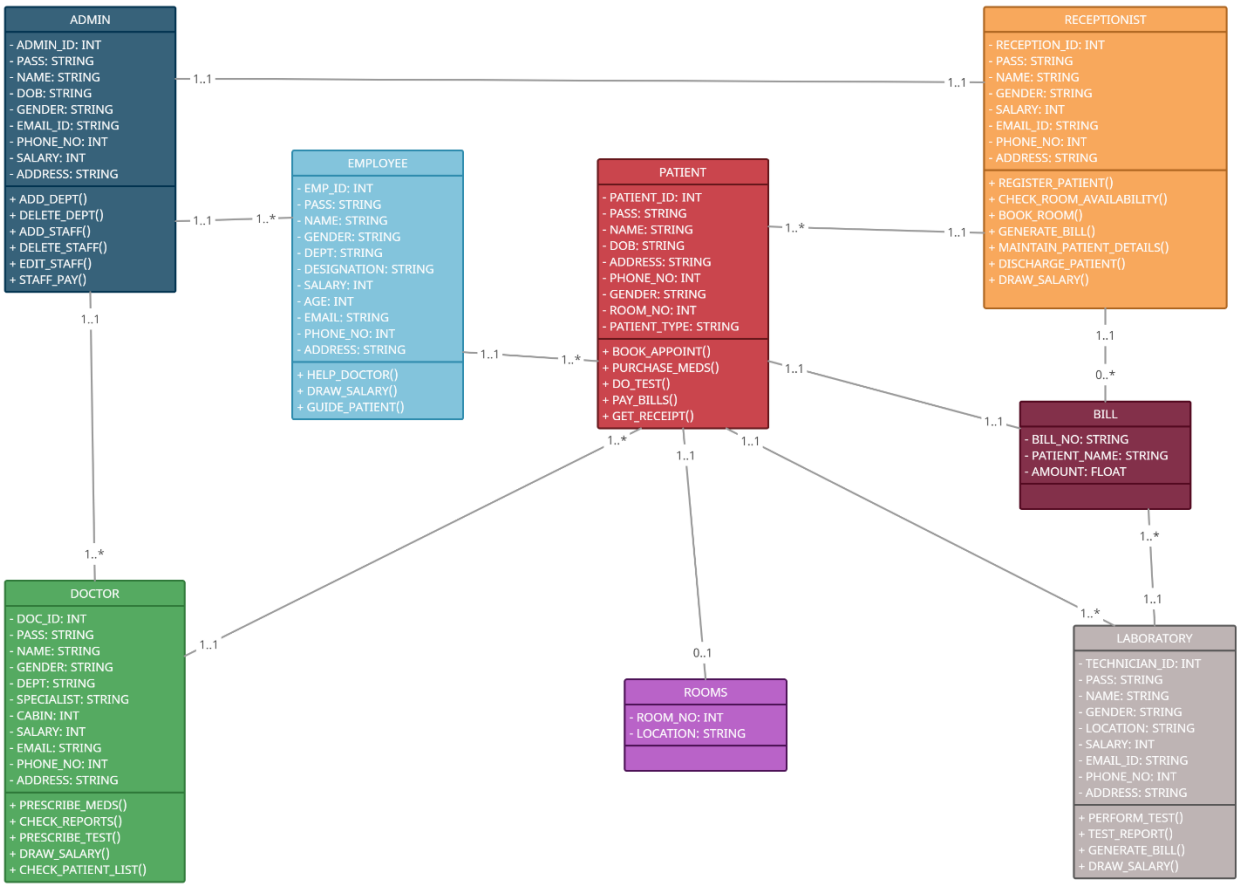
List of UML Diagram Types:

- Structure Diagrams
- Class Diagram
- Component Diagram
- Deployment Diagram
- Object Diagram
- Package Diagram
- Profile Diagram
- Composite Structure Diagram
- Behavioral Diagrams
- Use Case Diagram
- Activity Diagram
- State Machine Diagram

- Sequence Diagram
- Communication Diagram
- Interaction Overview Diagram
- Timing Diagram

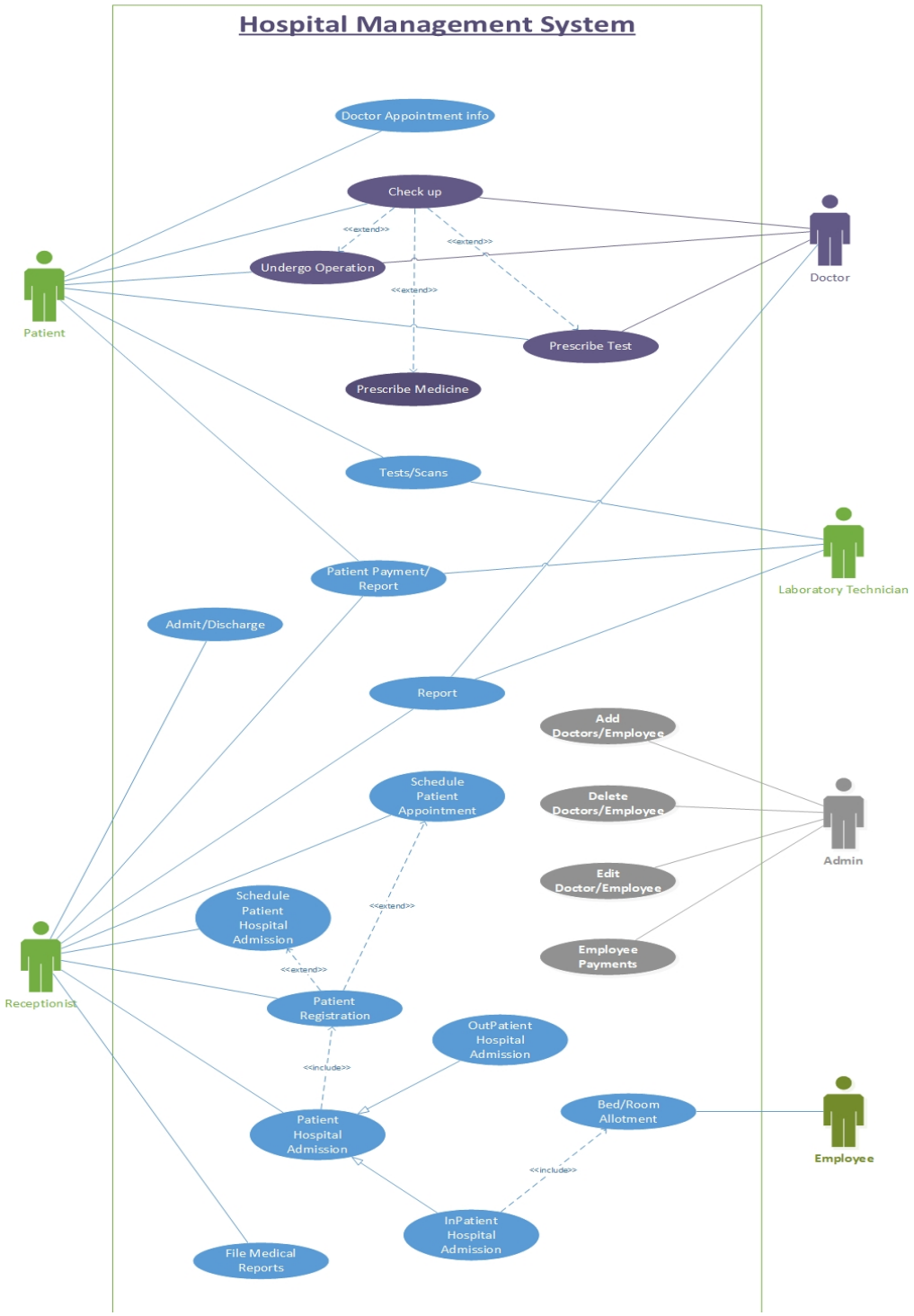
Here we have used Class Diagram, Use Case Diagram and Sequence Diagram to represent our Hospital Management System.

6.1.1. Class Diagram



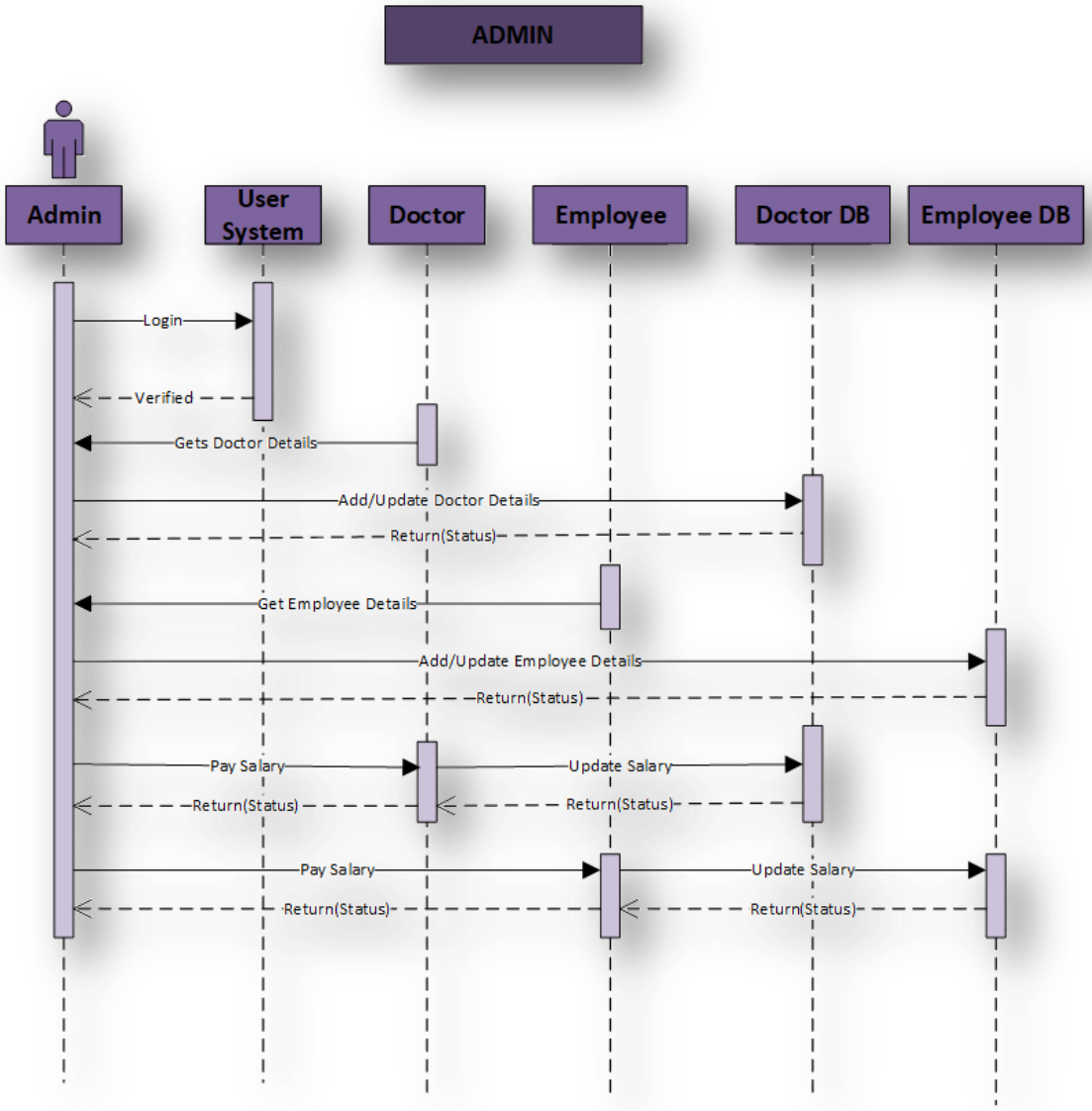
Here we have, Admin, Doctor, Employee, Patient, Receptionist, Rooms, Bill and Laboratory classes that have some data and methods to perform some operations on those data items. Like, Admin has Admin_id, Pass, Name and so on that are used to perform ADD_DEPT(), ADD_STAFF() like operations. Likewise, Doctor, Empolyee all the classes have their specific operations to preform that also pictorially defined in the Class Diagram.

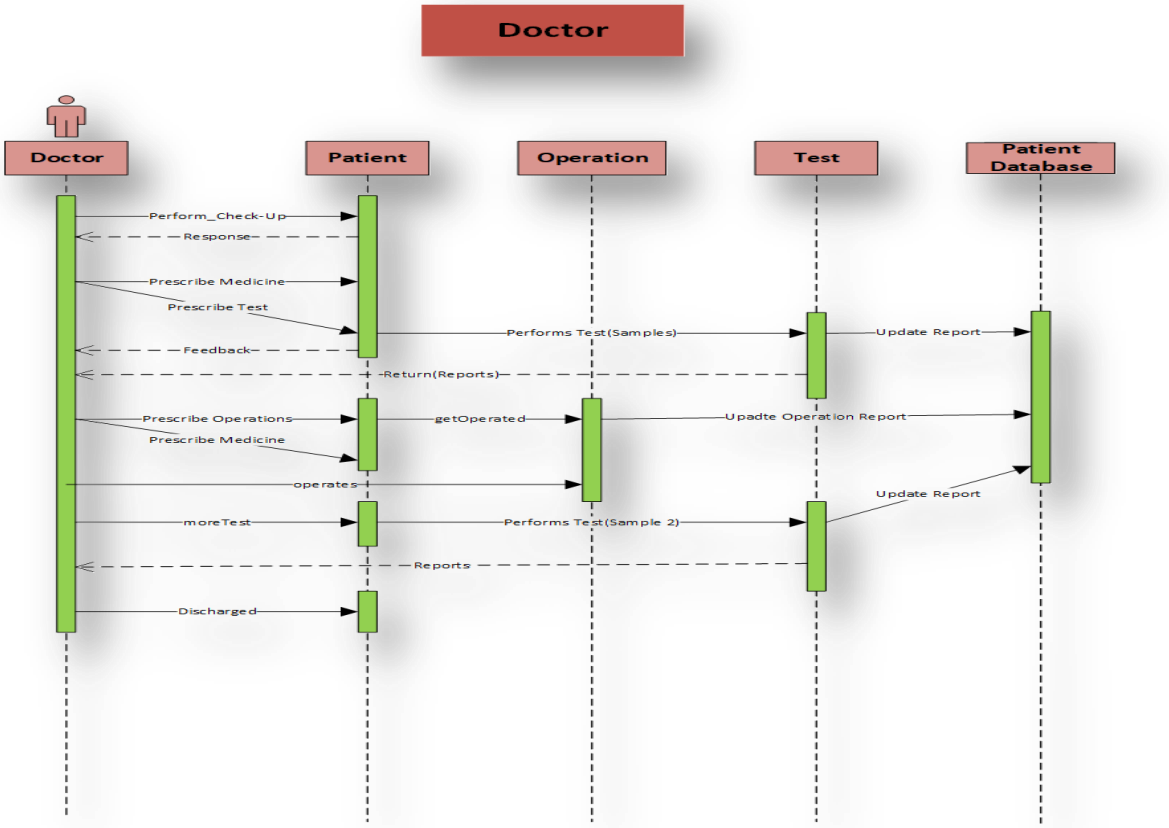
6.1.2. Use Case Diagram

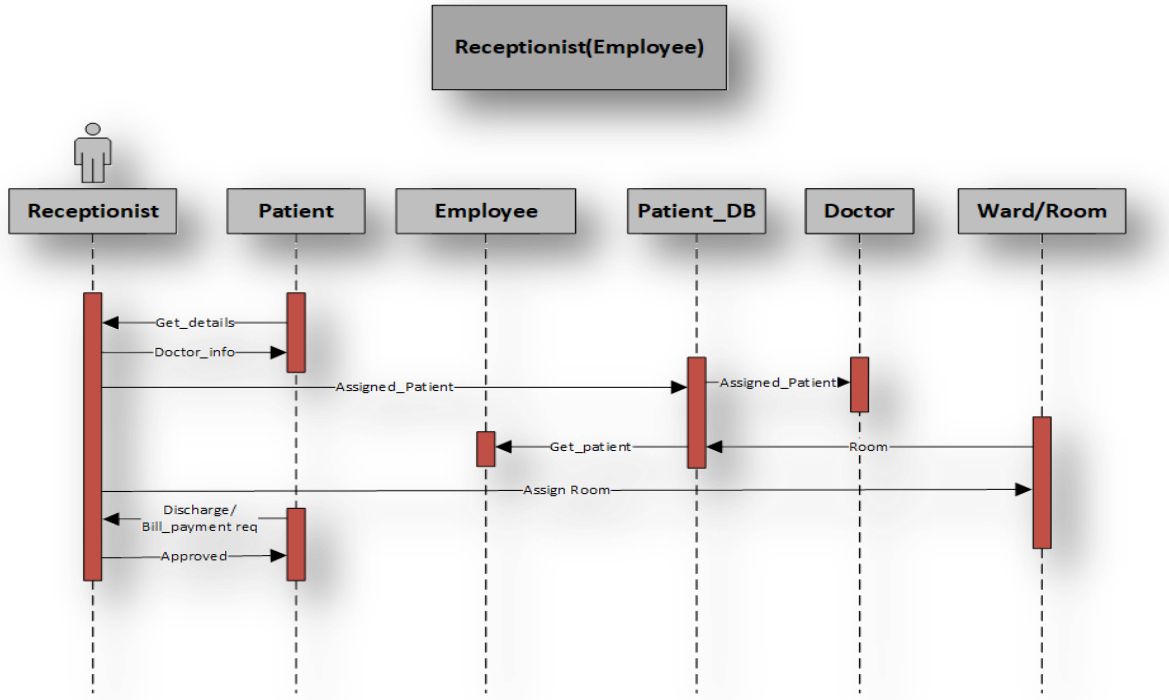


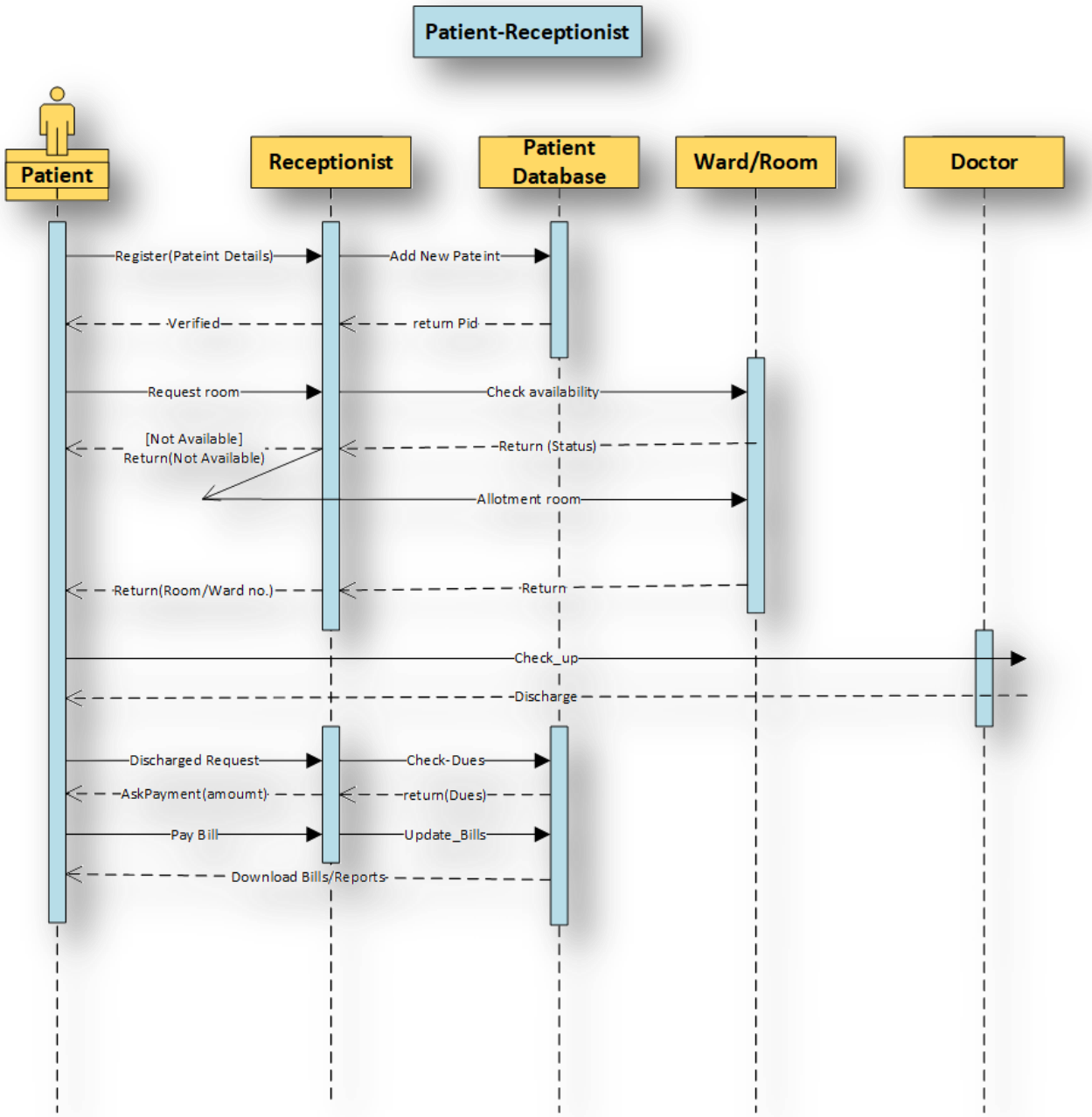
In the above Use Case Diagram, there are six actors named Doctor, Patient and so on. There are several use cases that represent the specific functionality of a Hospital Management System. Each actor interacts with a particular use case. Like, actor Patient interacts with Check_up, Tests, Payment that also interacts with actor Doctor and so on. Also each actor may have some use case that is specific to it. Like, Patient has Doctor_Appointment_Info, Receptionist has Admit/Discharge use case and Admin has all the use case specific to him.

6.1.3. Sequence Diagram




Doctor
Doctor
Patient
Operation
Test
Patient Database
Perform_Check-Up
Response
Prescribe Medicine
Prescribe Test
Performs Test(Samples)
Update Report
Feedback
Return(Reports)
Prescribe Operations
getOperated
Upadte Operation Report
Prescribe Medicine
operates
Update Report
moreTest
Performs Test(Sample 2)
Reports
Discharged


Receptionist(Employee)
Receptionist
Patient
Employee
Patient_DB
Doctor
Ward/Room
Get_details
Doctor_info
Assigned_Patient
Assigned_Patient
Get_patient
Room
Assign Room
Discharge/
Bill_payment req
Approved



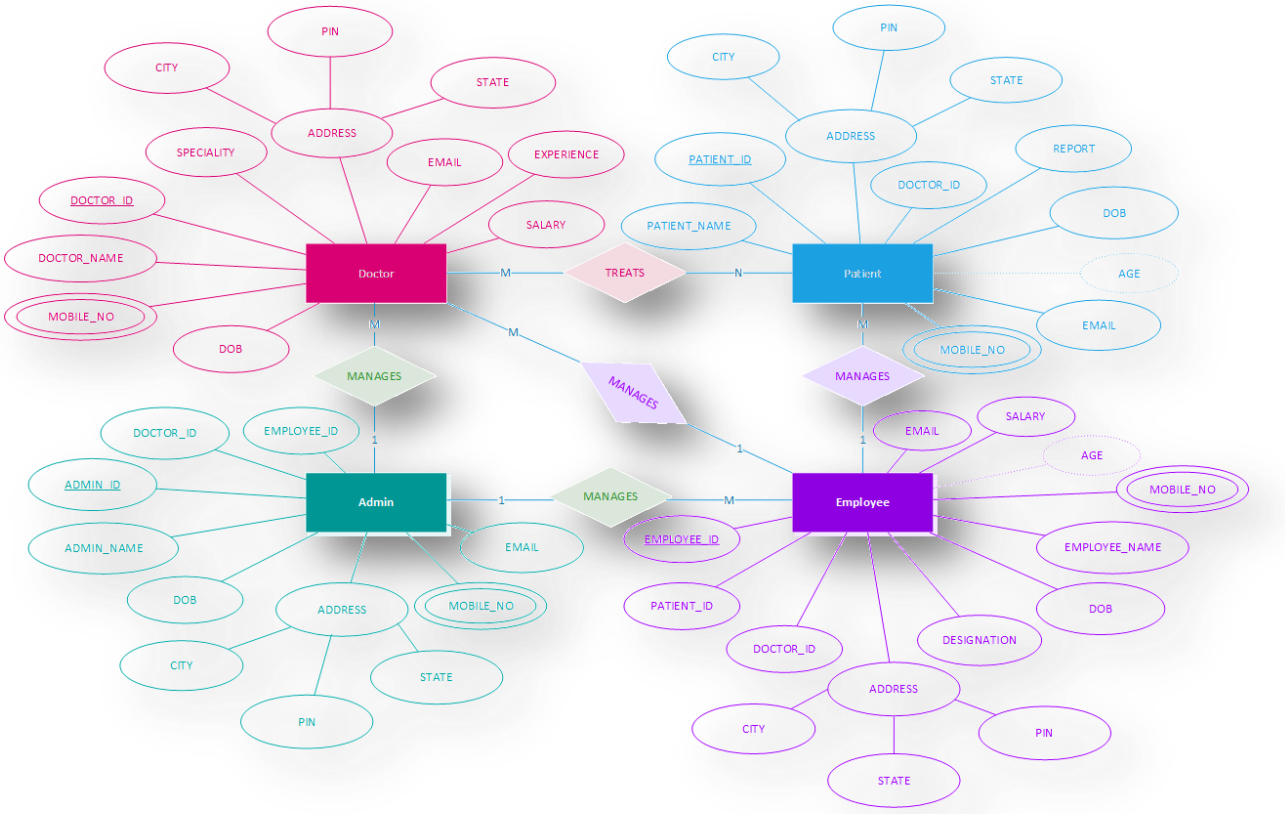
He we represent our whole Hospital Management System in five Sequence Diagram like Admin-Doctor-Employee, Doctor-patient, Employee-Patient, Patient-Receptionist Diagram.

6.2 Database Design

6.2.1. E-R [Entity-Relationship] Diagram

The database can be represented using the notations, and these notations can be reduced to a collection of tables.

In the database, every entity set or relationship set can be represented in tabular form



There are some points for converting the ER diagram to the table:

- Entity type becomes a table.

In the given ER diagram, ADMIN, DOCTOR, EMPLOYEE and PATIENT forms individual tables.

- All single-valued attribute becomes a column for the table.

In the ADMIN entity, ADMIN_NAME and ADMIN_ID form the column of ADMIN table. Similarly, DOCTOR_NAME and DOCTOR_ID form the column of DOCTOR table and so on.

- A key attribute of the entity type represented by the primary key.

In the given ER diagram, DOCTOR_ID, ADMIN_ID, EMPLOYEE_ID, and PATIENT_ID are the key attribute of the entity.

- The multivalued attribute is represented by a separate table.

In the ADMIN table, a MOBILE_NO is a multivalued attribute. So it is not possible to represent multiple values in a single column of ADMIN table. Hence we create a table USER_MOBILE with column name USER_ID and MOBILE_ID. Using both the column, we create a composite key. Similarly, same thing is applied with DOCTOR, EMPLOYEE and PATIENT table.

- Composite attribute represented by components.

In the given ER diagram, ADMIN address is a composite attribute. It contains CITY, PIN and STATE. In the ADMIN table, these attributes can merge as an individual column. Similarly, same thing is applied with DOCTOR, EMPLOYEE and PATIENT table.

- Derived attributes are not considered in the table.

In the PATIENT table, Age is the derived attribute. It can be calculated at any point of time by calculating the difference between current date and Date of Birth (DOB). Similarly, same thing is applied with EMPLOYEE table.

Using these rules, we can convert the ER diagram to tables and columns and assign the mapping between the tables. Table structure for the given ER diagram is as below:

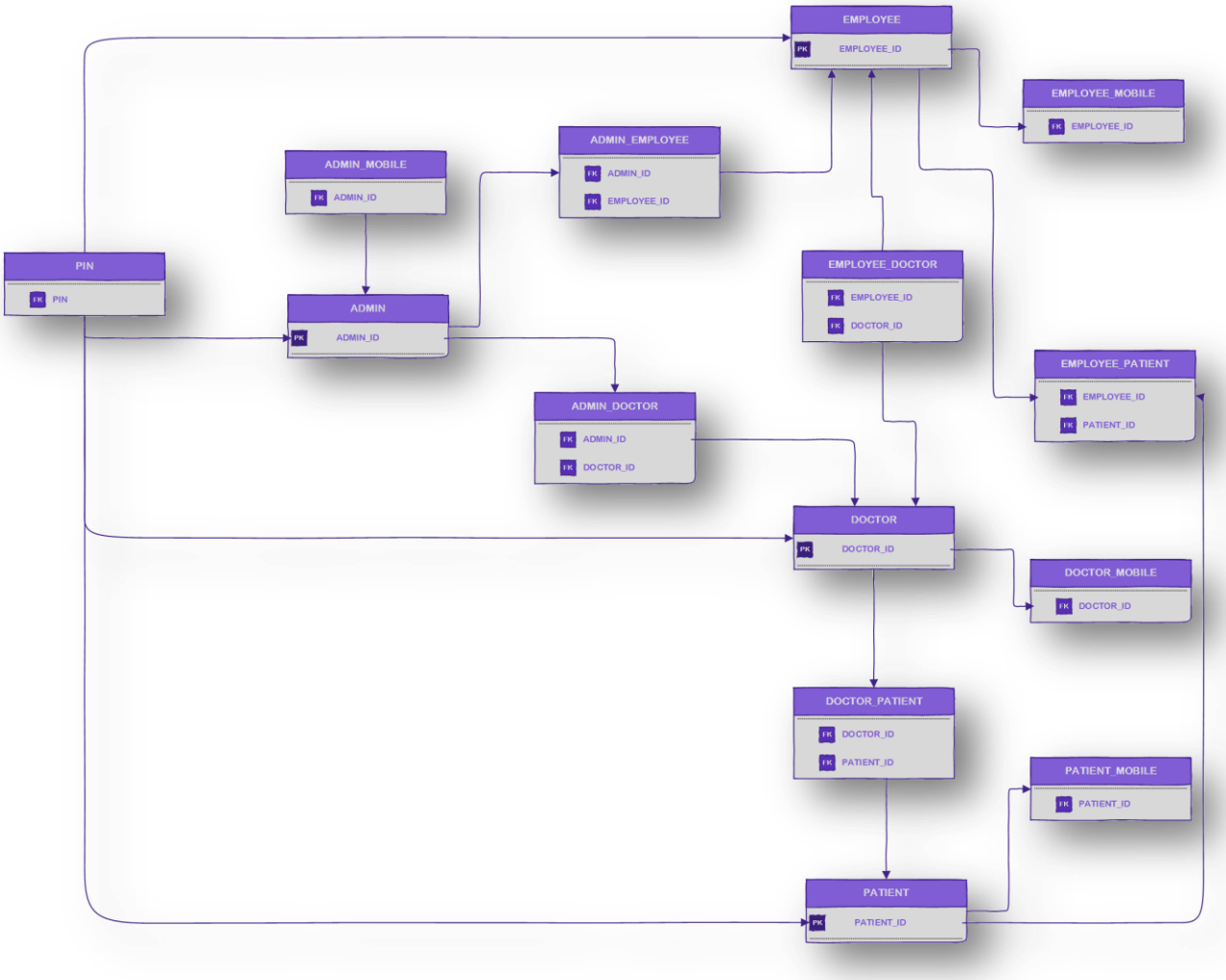


Figure: Table structure

In 1NF, A relation is in 1NF if it contains an atomic value.

In 2NF, A relation will be in 2NF if it is in 1NF and all non-key attributes are fully functional dependent on the primary key.

In 3NF, A relation will be in 3NF if it is in 2NF and no transition dependency exists.

Here in our table-structure, we have decomposed the following tables into 3NF.

CHAPTER-7
TESTING AND RESULT

The reason behind testing is to find errors. Every program or software has errors in it, against the common view that there are no errors in it if the program or software is working. Executing the programs with the intention of finding the errors in it is therefore testing; hence a successful test is one which finds errors. Testing is an activity; however it is restricted to being performed after the development phase is complete, but is carried parallel with all stages of system development, starting with requirement specification. A test case is a set of the data that a system will process as normal input. The software units developed in the system are modules and routines that are assembled and integrated to perform the required function of the system. Test results once gathered and evaluated, provide a qualitative indication of the software quality and reliability and serve as basis for design modification if required. The testing phase of the implementations works accurately and efficiently before live operation commences.

Unit Testing

The unit testing was done after the coding phase was done. The purpose of the unit testing was to locate errors on the module, independent of the other modules. Some changes in the coding were done during the testing. Finally all the modules were individually tested from bottom up starting with smallest and lowest modules and proceeding one at a time.

Black Box Testing

This method of software testing tests the functionality of an application as opposed to its internal structures or working. Specific knowledge of the internal structure and programming knowledge in general is not required. It uses external descriptions of the software, including specifications, requirements, and designs to derive test cases. The test designer selects valid and invalid inputs and determines the correct output.

White Box Testing

This method of software testing tests internal structures or workings of an Page 16 of 17 application, as opposed to its functionality (i.e. black-box testing). In white-box testing an internal perspective of the system, as well as programming skills, are required and

used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs.

Integration Testing

Once the unit was over, all the modules were integrated for integration testing. External and internal interfaces are implemented and work as per design, the performance of the module is not degraded.

Validation Testing

At the culmination of integration testing, software is said to be completely assembled as a package; interfacing errors have been uncovered and corrected. Then as a final series of software test, validation tests were carried out.

Acceptance Testing

This is the final stage in the testing process before the system is accepted for operational use. Any requirement problem or requirement definition problem revealed from acceptance testing are considered and made error free.

CHAPTER – 8
CONCLUSION

This Software Requirement Specification (SRS) document is used to give details regarding Hospital Management System. The project Hospital Management System is for computerizing the working in a Hospital. It is a great improvement over the manual system. The computerization of the system has speed up the process. In the current system, the front office managing is very slow. The Hospital Managing System was thoroughly checked and tested with dummy data and thus is found to be very reliable. The software takes care of all the requirements of an average hospital and is capable to provide easy and effective storage of information related to patients that come up to the hospital and also information related to doctors and employees who works in the Hospital. It generates test reports and also provide the facility for searching the details of the patient. It also provides billing facility on the basic of patient's status whether it is an indoor or outdoor patient. The system also provides the facility of backup as per the requirement.

Future enhancements:

The proposed system is Hospital Management System. We can enhance this system by including more facilities like pharmacy system for the stock details of medicine in the pharmacy.

Limitations:

- The size of the database increases day-by-day, increasing the load on the database backup and data maintenance activity.
- Training for simple computer operations is necessary for the users working on the system.

CHAPTER – 9
REFERENCES

- Fundamentals of Software Engineering 4th Edition by Rajib Mall
- Fundamentals of Software Engineering,2nd Ed. ,Ghezzi,PHI
- Software Engineering, Pankaj Jalote, PHI
- Classical and Object Oriented Software Engineering, Schach, TMH
- Software Engineering: Principles & Practice,Van Vliet,SPD/JOHN WILEY
- Software Engineering, K.K.Aggarwal & Yogesh Singh, New Age International
- HTML and CSS: Design and Build Websites Paperback by Jon Duckett
- HTML & CSS: The Complete Reference, Fifth Edition by Thomas Powell
- PHP: The Complete Reference Paperback by Steven Holzner
- MySQL(TM): The Complete Reference Paperback by Vikram Vaswani