Hospital Management System

A Practicum Report Submitted By

Md. Farhad Islam Shakil

ID No. 14203055

In Partial Fulfillment of the Requirements for the Award of

Bachelor of Computer Science and Engineering

Department of Computer Science and Engineering

College of Engineering and Technology

IUBAT– International University of Business Agriculture and Technology

Spring 2018

Hospital Management System

Md. Farhad Islam Shakil

A practicum report submitted in partial fulfillment of the requirements for the degree of Bachelor of Computer Science and Engineering (BCSE)

The practicum has been examined and approved,

_____

Prof Dr Md Abdul Haque
Chair and Professor
Dept. of Computer Science and Engineering
IUBAT – International University of Business Agriculture and Technology

_____

Dr Utpal Kanti Das
Coordinator and Associate Professor
Dept. of Computer Science and Engineering
IUBAT – International University of Business Agriculture and Technology

_____

Md. Mortuza Hossain
Dept. of Computer Science and Engineering
IUBAT – International University of Business Agriculture and Technology

Department of Computer Science and Engineering
College of Engineering and Technology

IUBAT – International University of Business Agriculture and Technology

Spring 2018

ABSTRACT

My project Hospital Management system includes registration of patients, storing their details into the system, and also computerized billing in the pharmacy, and labs. My software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. It includes a search facility to know the current status of each room. User can search availability of a doctor and the details of a patient using the id.

Admin having the rights to create doctor, employee, equipment details, appointments and user management. These details are secured and only admin can edit and update the details. There are a lot of benefits to the Hospital and doctor by placing the system at their registration and doctor appointments. At the same time the patients are also benefited using this system. They can get the work done within no time.

The Hospital Management System can be entered using a username and password. It is accessible either by an administrator or receptionist. Only they can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast.

LETTER OF TRANSMITAL

12 April, 2018

To

Md. Mortuza Hossain

Faculty, Department of Computer Science & Engineering.

IUBAT-International University of Business Agriculture & Technology

4Embankment Drive Road, Sector-10, Uttara Model Town, Dhaka-1230, Bangladesh

Subject: Submitting a report on "**Hospital Management System**".

Dear Sir,

As agreed we are submitting the report entitled "**Hospital Management System***"*.

By this project we can show our efficiency of technical work effectively. For your kind consideration we inform you that we tried our best to prepare the project for an effective System design for**"Hospital Management System***"*.

.We have immense pleasure to have the study on this which is effective in our practical field as well as our study sector. It was great opportunities for us to work on this report actualize our theoretical knowledge in the practical area.

We hope you find this report satisfactory. After that, if you have any query we will be available any time.

Sincerely yours,

……

Md. Farhad Islam Shakil

STUDENT'S DECLARATION

I hereby declare that this practicum report titled '**Hospital Management System**' is my original work. It has never been presented previously or concurrently for any other purpose, reward or degree at IUBAT University or any other institutions either by me or by any other student. I also declare that there is no plagiarism or data falsification and materials used in this report from various sources have been duly cited.

……..

Md. Farhad Islam Shakil

ID No. 14203055

ACKNOWLEDGEMENTS

At first I give to thanks Allah who is the most merciful and the most graceful. I am extremely grateful to Late Prof Dr M Alimullah Miyan, the Vice-Chancellor for taking me as an intern in IUBAT. And also giving thanks to Prof Dr.Abdur Rab, the present Vice-Chancellor of IUBAT. It is also an immense pleasure for me to thank for assigning me that project.

I am heartily thankful to my supervisor, Md. Mortuza Hossain, whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject.

Furthermore, I would like to thank Mr. Utpal Kanti Das, Md. Rashedul Islam and Md. Riaz Hasib Hossain because the knowledge and idea obtained from his previous courses helped me a lot to complete this project.

Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of the project.

SUPERVISOR'S CERTIFICATION

I certify that the student Md.Farhad Islam shakil (ID No. 14203055) carried out her practicum work '**Hospital Management System**' between January 7, 2018 and April 10, 2018. During this period she consulted me on regular basis as required by the department. I therefore recommend that her practicum report be accepted for oral examination.

_____

Md. Mortuza Hossain
Lecturer
Dept. of Computer Science and Engineering
IUBAT – International University of Business
Agriculture and Technology

DEPARTMENTAL CERTIFICATION

On behalf of the Department of Computer Science and Engineering of International University of Business Agriculture and Technology (IUBAT University) we, the undersigned, certify that this practicum report 'Developing e-Commerce System' for the award of Bachelor of Computer Science and Engineering (BCSE) degree was duly presented by Md. Farhad Islam Shakil (ID No. 14203055) and accepted by the department.

_____

Md. Mortuza Hossain
Supervisor

_____

Dr. Utpal Kanti Das
Coordinator

_____

Prof Dr Md Abdul Haque
Chaiman

TABLE OF CONTENT

Contents

LIST OF FIGURE

List Page No

LIST OF TABLE

LIST OF ABBREVIATION

Short Name	Abbreviation
DFD	Data Flow Diagram
ERD	Entity Relationship Diagram
RE	Requirement Engineering
CC	Customer Communication
FP	Function Point
EI	External Input
EQ	External Query
EO	External Output
ILF	Internal Logical File
FTR	File Type Reference
DET	Data Element Type
RET	Record Element Type
UFP	Unadjusted Function Point
AFP	Adjusted Function Point
RMMM	Risk Mitigation, Monitoring, and Management Plan

Chapter-1

The Organization

1.1. Organizational Overview:

MYSOFT IT is one of the leading IT companies in Bangladesh. We provide Secure, Scalable, On-Demand Application System and Data Solutions to help customers improving their business performance. It is also an IT Enabled Service (ITES) provider in the domain of Healthcare, Textiles, Manufacturing as well as e-Commerce Portal business. As a provider of outstanding technology solution and management consultancy through Information Technology, MYSOFT bridges the gap between performance and corporate vision. MYSOFT has achieved very positive response from market in providing business solution to renowned business houses. Due to its customer centric approach and strong post- sales services delivering business value, within a very brief period, MYSOFT achieved numerous clients (more than 60).

Its core with the highly qualified Designers and Developers having experience of more than 3 years in various and complex designs and development. MySoft It has satisfied the clients with the services like Web design and development, Mobile app design and development,
Software development,SEO and Social Media Designing & Development. We are customer centric and divert our efforts to act as a one-stop solution provider in the area of IT. In every area of our, operations we work hard in understanding the Client's requirement and providing the Sailor made solution.

We firmly believe in the philosophy of 'Our Asset Our People'. We take pride in a team of highly qualified, skilled and motivated Professionals who are encouraged to lead, innovate and excel. Our team consists of top professionals who share a common vision and passion, providing our clients with critical insights and advice to succeed in today's competitive environment. We believe in delivering Expertise, Excellence Services through our past Experience and providing the highest and best end use of services to our client.

MYSOFT has gained a notable recognition in Bangladesh. We have developed a wide range of software applications namely as Hospital Information System, Clinical Laboratory Information System, Radiology Information System, Endoscopy Image Processing and Reporting System, Picture Archiving and Communication System, Bidirectional Host Communication System, Clinical ERP, etc

1.2. Mission

Produce excellent service in the field of IT Service, Software Development, Website Design & Development, Apps Development, SEO (Search Engine Optimization), SMM (Social Media Marketing), Online Advertisement, e-database systems and banking home and abroad, E-commerce and Consultancy with maximum effort driven toward customer satisfaction.

- ✓ To achieve maximum customer satisfaction over the entire life cycle of our customer solution via our excellence of products and solutions.
- ✓ To consistently enhance our competitiveness and deliver profitable growth.
- ✓ To practice highest standards of corporate governance and be a financially sound company.
- ✓ To be a partner in nation building and contribute towards Bangladesh economic growth.
- ✓ To encourage ideas, talent and value systems and become the customers of choice.
- ✓ To earn the trust and confidence of all customers, exceeding their expectations.
- ✓ To uphold the guiding principles of trust, integrity and transparency in all aspects of interactions and dealings.

1.3. Vision

- ✓ To build a trusted IT Companies in Bangladesh.
- ✓ To be the largest Software Companies in Bangladesh.
- ✓ To be the largest online retailer.
- ✓ To be the best choice for people when they like to Apps Development.
- ✓ To be the largest SEO (search engine optimization) Marketing Company in Bangladesh.
- ✓ To be the largest IT company in World.

1.4. My Position in this Company

I am working as a Junior Software Developer in this company.

1.5 Service Provided

In today's Business world to get edge in competition, a robust IT strategy is critical for every industry to attain maximum flexibility, speed, and efficiency. Sailor Info Tech IT Services offering helps companies make the most of their IT investments – from providing System. Integration Solutions, Application development and Management Services, and Testing Solutions. See their expertise area below.

Sailor Info Tech is committed to deliver excellence and certainty across all of types of enterprise's IT needs. Sailor Info Tech offers services that cover:

- ✓ Web Design and Development
- ✓ Software Development
- ✓ Mobile Application Development
- ✓ Graphic Design
- ✓ Corporate Identity
- ✓ Interactive Design
- ✓ Online Marketing
- ✓ Domain Registration

1.6. Organizational Structure

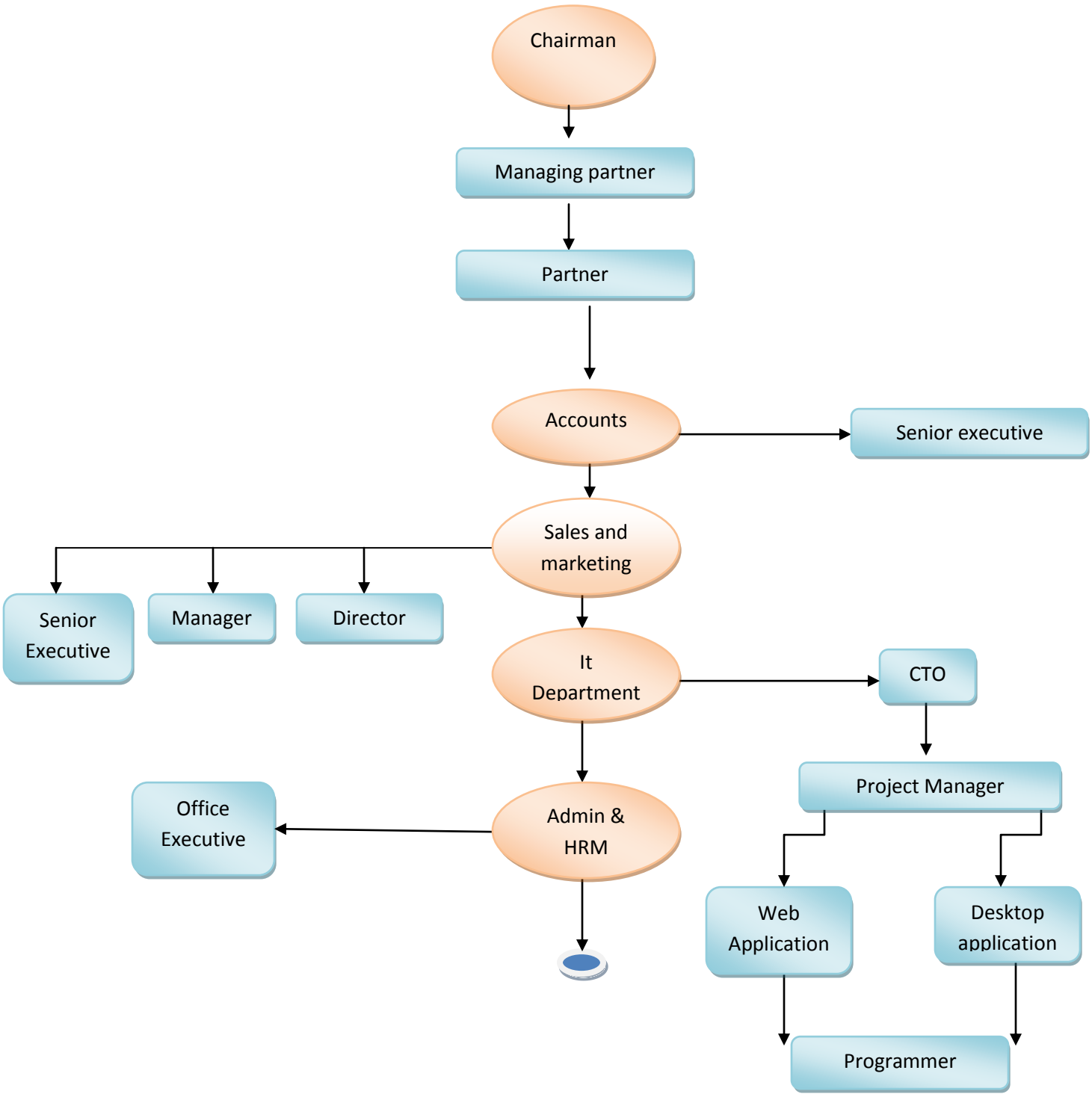


Figure 1.1: Organizational Structure

Chapter-2

The Project

The project name is "**Hospital Management System**".

The purpose of the project entitled as "HOSPITAL MANAGEMENT SYSTEM" is to computerize the Front Office Management of Hospital to develop software which is user friendly, simple, fast, and cost – effective. It deals with the collection of patient's information, diagnosis details, etc.

Traditionally, it was done manually. The main function of the system is to register and store patient details and doctor details and retrieve these details as and when required, and also to manipulate these details meaningfully System input contains patient details, diagnosis details; while system output is to get these details on to the CRT screen.

2.1 Broad Objective

This main objective of the Hospital Management System is to computerize the maintenance of the patient details and appointment section in the hospital. This system has been developed to form whole management system including Employees, doctors, patients, and appointments etc. It also includes the details about the laboratory equipments.

2.2 Specific Objective

- ✓ To minimize the length of work process.
- ✓ To maximize effectiveness of record.
- ✓ To eliminate the duplication of work.
- ✓ To eliminate manual work process.
- ✓ To provide maximum security.
- ✓ To check information by the management and take action.

2.3. System Benefit:

System benefit are given below:

2.3.1. Easy Access To Patient Data

A well-implemented Hospital Management System means readily available patient data to the care providers. It is only a matter of few clicks and all the requisite information about a patient, from various departments in the hospital, can be available on the screen. If the treating doctor needs to re-check the test reports of a patient, she need not go looking for the IPD file; logging into the HMS will give her instant access to those reports and timely treatment decisions ensue.

2.3.2. Cost Effective

HMS, when implemented well, cuts out on a lot of manual work that are essentially performed in hospitals, especially the ones where documentation and record keeping is required. It helps in cutting down manpower because a lot of work gets automated and does not require manual intervention to store or analyze the information. It also saves much on storage and the related costs.

2.3.3. Improved Efficiency

Processes automated using software would mean that the processes will be taken care of mechanically without any human intervention and this will instantly ensure improved efficiency. The software will not face human problems like fatigue, miscommunication or lack of focus; it will perform every task assigned to it with the same accuracy day in and day out.

2.3.4. Reduces Scope of Error

Because processes on HMS are automated and a lot of tasks are assigned to the software to perform with utmost accuracy with minimum human intervention, the scope of error is reduced dramatically. For instance, while billing patient for the drugs used with HMS, the bill can hardly go wrong because the drug the nurse indents is what is billed for until and unless there is a shortage in stock or change in drug order after the indent has been sent. Per unit rate of the drug is saved in the software as part of standard operating procedure of automation. Just selecting the drug name and the quantity will enable the software to calculate the amount due,accurately.

2.3.5. Increased Data Security & Retrieve-ability

Record keeping in hospitals is a mandatory bane with two challenges: keeping the data safe with only authorized personnel getting access to it and retrieving it in the minimum possible time. Add to these the perennial problems of space shortage, protection from natural elements and protection from pest damage etc.

HMS is the perfect solution for these problems. All the data is stored on the server or cloud, keeping it safe. Since HMS works on logins, data security is becomes a non-issue offering data access based on the role of the person – Receptionist, doctor, nurse, radiologist etc. Retrieve-ability of data stored on a server or cloud is only a matter of few clicks and the data will appear on the screen within seconds.

2.3.6. Improved Patient Care

Improved access to patient data and improved work efficiency means better and faster clinical decisions. In this age of evidence based medicine, the faster the clinician gets the diagnostic reports and the quicker her orders are implemented the faster is the patient recovery and the better it is on the patient care index. With automation, all departments in the hospitals are inter-connected and the faster information access further improves the quality of patient care and the resultant bed turnover in the hospital.

2.4. Methodologies

To develop this project first is to make the requirement analysis. Depending on the user requirements choose the Waterfall Process Model to develop the system and follow this model in entire project of development.

2.5. Software Development Process Model

Waterfall process model is appropriate in small project and my project is consider as a small project. It is very simple to understand and use. In a waterfall model, each phase must be completed fully before the next phase can begin. This type of software development model is basically used for the for the project which is small and there are no uncertain requirements. At the end of each phase, a review takes place to determine if the project is on the right path and

whether or not to continue or discard the project. In this model software testing starts only after the development is complete.

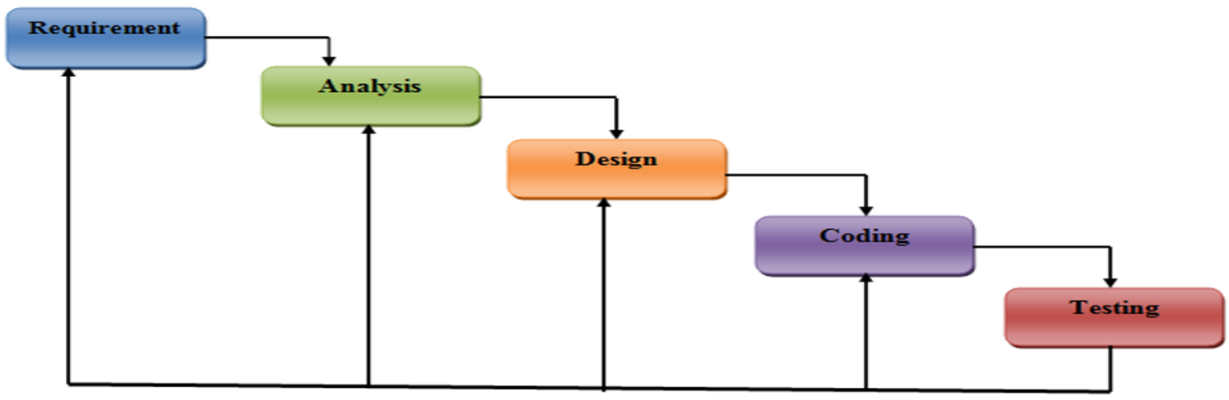


Figure 2.1: Waterfall Process Model

2.5.1. Reason For Choosing:

- This Model is used only when the requirements are very well known, clear and fixed.
- Product definition is stable
- Technology is understood.
- There are no ambiguous requirements.
- The project is short.

As these criteria match the case of this project, waterfall model is chosen to be process model.

2.5.2. Steps of Waterfall Model

- **Requirements Gathering:** all possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.
- **Analysis**: During this phase research is being conducted which includes brainstorming about the software, what is going to be and what purpose is it going to fulfill. This step is also the most important, because it involves gathering information about what the customer needs and defining, in the clearest possible terms, the problem that the product is expected to solve.

- **Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System design helps in specifying hardware and system requirements and also helps in defining overall system architecture.
- **Coding:** In this phase the source code of the program is written.
- **Testing:** After coding phase, all the units are integrated into system. Post integration the entire system is tested for any faults and failures. In this stage, both individual components and the integrated whole are methodically verified to ensure that they are error-free and fully meet the requirements outlined in the first step.

2.5.3. Benefit of Using Waterfall Model

- The project requires the fulfillment of one phase, before proceeding to the next. Therefore, if there is a fault in this software it will be detected during one of the initial phases and will be sealed off for connection.
- A lot of emphasis is laid on paperwork in this method as compared to the newer method. When new workers enter the project, it is easier for them to carry on the work from where it had been left. The newer methods don't document their developmental process which makes it difficult for a newer method and lets one know easily what stage is in progress.
- The waterfall method is also well known amongst the software developers therefore it is easy to use. It is easier to develop software through this method in short span of time.

2.6 Feasibility Study

Feasibility study determines whether that solution is feasible or achievable for the organization. There are three major areas of feasibility study.

- ✓ Technical feasibility
- ✓ Economic feasibility
- ✓ Operational feasibility

2.6.1. Technical Feasibility

Technical feasibility addresses concerns about hardware capability, reliability and availability and the skills of the development team. This study looks at the hardware and software available to perform the necessary steps for the proposed system. I have identified several software and hardware requirements for the implementation of our system. The hardware and software requirements are as follows:

- ✓ SQL Server
- ✓ Sublime Text
- ✓ Power backup for electricity failure.

2.6.2. Economic Feasibility

Economic feasibility determines to what extent a new system is cost effective. We are developed this system for Hospital Management System. Despite keeping records in physical document Hospital Mangement System would be financially feasible as user will be able to manage their entire payroll system by this system. Using this system will reduce their paper work and will minimize the length of their work process. Analyzing both from client and development team side we found that the proposed system will be cost effective to develop.

2.6.3. Operational Feasibility

Operational feasibility addresses concerns about user acceptance, management support, and the requirements of entities. The proposed system will be desirable with the existing way of their Hospital Management System. As the system will cover all the requirements of user it will become very easy for them to keep records. During our conversation with them we found that they are very much interested and eager to use this system.

.

Chapter-3

Requirement Engineering

The process of establishing the services that the customer requires from a system and the constraint under which it operates and is developed.

Requirement reflects the needs of customer for a system that serves a certain purpose such as controlling a device, placing an command or finding information.

The process of finding out, analyzing, documenting and checking these services and constrains is called requirement engineering (RE).Bridge to design and construction.

3.1. Methodology

For analysis the requirements we follow the 6 phases of requirement engineering. In each phase we do the specific job for requirement analysis.

3.2. Requirement Analysis

Requirements analysis is a software engineering task that bridges the gap between system engineering and system design. Requirements analysis allows the software engineer to define the software allocation and build the models of the data, functional and behavioral domains that will treat by software. Requirement analysis provides the software designer with a representation of information, function and behavior of the system.

Requirements analysis is the first stage in the software development process. It encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, such as beneficiaries or users.

Analysis the requirement is critical to the success of a development project. Requirements must be actionable, measurable, testable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design. Requirements can be functional and non-functional.

There are 6 phases of requirement analysis which is described hereby →

Requirement Initiation: I submit a proposal on the project entitled " Hospital Management System".

Requirement Elicitation:

Eliciting requirements is the task of communicating with customers and users to determine what their requirements are. This is sometimes also called requirements gathering.

New systems change the environment and relationships between people, so it is important to identify all the stakeholders, take into account all their needs and ensure they understand the implications of the new systems.

In this step a team from Sailor Info Tech goes to Mirai group, Baridhara for requirement gathering. We also try to understand the user's needs and constraints for the system. We analyze User's work process and their organizational structure.

In this phase we do mainly two works →

- Analyze the Requirement
- Recording User Requirements

Requirement Elaboration: is the process of collecting user's needs and constraints. How the entities of the system will interact with each other.

Requirement Negotiation: is the process of negotiating with the client about the software cost and other facilities will be provided with the system.

Requirement Specification:

A software requirements specification (SRS) is a complete description of the behavior of the system to be developed. It includes a set of use cases that describe all of the interactions that the users will have with the software.

In this stage we specified the manpower requirements, and technologies required for the deployment of the system.

Requirement Verification: We verify all the requirements of the user whether they need any modification or not.

3.3. User Requirement & System Requirement

User requirement is most important and first step of software. It decides how the software will behave. From there we get the specification for the software. The user requirement and system requirement of this Hospital Management System is given below:

User Management

User requirement:

- User can login, edit profile, change password.

System Requirement:

- First user login with email and password.
- If email and password is valid user can view his dashboard.
- From there he can view profile, edit profile, change password.

Role Management:

User Requirement:

- Admin can add menu, function, edit menu, edit function.
- Admin can create role, edit role , delete role.
- Admin can create user, edit user, delete user, block user and also deactivate any user.
- Admin can give access role to the user.

System requirement:

- After clicking menu admin will view menu.
- From menu admin click on add new to add new menu.
- Fill the menu form and submit and the page redirect to the view menu page.
- Admin can edit delete from this view page of the menu.
- After clicking function admin will view function.
- From function admin click on add new to add new function.
- Fill the menu form and submit and the page redirect to the view menu page.
- Admin can edit delete from this view page of the function.

- Admin click on role and view role.
- By clicking add new role. Admin can create new role. From there admin can add all menus and function how many access admin want to give this role.
- From menu view page admin can edit and delete role.
- Same way admin can add new user, edit user, delete user.
- After clicking user admin will view user.
- From user admin click on add new to add new user.
- Fill the user form and submit and the page redirect to the view user page.
- Admin can edit delete from this view page of the user.

Hospital Allocation management:

User requirements:

- Admin can view hospital details like name address, photo and other necessary data.
- Admin can edit those data

System requirements:

- Clicking on hospital registration admin can view all details of hospital.
- From this view page admin can edit anything admin need and by clicking submit, store the data into the database.

Hospital Building Management

User Requirement:

- Sub admin can add building details, edit building details, delete building details.
- Sub admin can add floor details, edit floor details, delete floor details.
- Sub admin can add room details with cost, edit room details, delete room.
- Sub admin can add bed details, edit bed details, delete bed details.

System Requirement:

- After clicking on building sub admin view all building details.

- Clicking on add new building sub admin can add new building details of hospital and after successful add system gives a successful message.
- From view page sub admin can edit and delete building details
- After clicking on floor sub admin view all floor details.
- Clicking on add new floor sub admin can add new floor details of hospital and after successful add system gives a successful message.
- From view page sub admin can edit and delete floor details
- After clicking on room sub admin view all room details.
- Clicking on add new room sub admin can add new room details of hospital and after successful add system gives a successful message.
- From view page sub admin can edit and delete room details
- After clicking on bed sub admin view all bed details.
- Clicking on add new bed sub admin can add new bed details of hospital and after successful add system gives a successful message.
- From view page sub admin can edit and delete bed details.

Human Resources Management

User Requirement:

- Sub admin can view, add, edit and delete department.
- Sub admin can view, add employee, edit employee, delete employee.
- Sub admin can view doctor details, add doctor details and delete doctor details.

System Requirement:

- After clicking on department sub admin view all department details.
- Clicking on add new department sub admin can add new department details of hospital and after successful add system gives a successful message.
- From view page sub admin can edit and delete department details.
- After clicking on employee sub admin view all employee details.
- Clicking on add new employee sub admin can add new employee details of hospital and after successful add system gives a successful message.
- From view page sub admin can edit and delete employee details.

- After clicking on doctor sub admin view all doctor details.
- Clicking on add new doctor sub admin can add new doctor details of hospital and after successful add system gives a successful message.
- From view page sub admin can edit and delete doctor details.

Patient Management

User requirement:

- Receptionist can register new in-patient, edit in patient details.
- Receptionist also can allocate bed for inpatient and billing and discharge and report generate for in-patient billing.
- Receptionist can view available bed.
- Receptionist can register the outpatient details, edit details. And also can billing of outpatient.

System Requirement:

- Clicking on register in patient receptionist can register patient and in the mean time receptionist can allocate bed for patient by selecting bed.
- From clicking view receptionist can view inpatient details. And also can edit details if any there is any need.
- From available bed receptionist can view available bed.
- Clicking on outpatient registration receptionist can register outpatient and from view receptionist edit and delete outpatient.
- There is discharge menu from where all patient bills are viewed and after payment complete receptionist click on the discharge button then receptionist generate the billing report of the in patient.

Pharmacy Management

User requirement:

- Pharmacist can view, add, edit, delete product.
- Pharmacist can view add delete supplier.

- Pharmacist can purchase product and view the transaction of the purchase product. And also can pay the due payment.
- Pharmacist can sell the product and view the transaction of sell. And also can take the due payment. Pharmacist registers the sell record within inpatient registration number.
- Pharmacist can generate purchase and sell report and billing report.

System Requirement:

- After clicking on supplier sub admin view all supplier details.
- Clicking on add new supplier sub admin can add new supplier details of hospital and after successful add system gives a successful message.
- From view page sub admin can edit and delete supplier details.
- For purchase pharmacist will go to purchase and purchase product with product name expiry date, price and supplier. From their pharmacist can view transaction and view the status of transaction.
- By depending on status pharmacist will pay the due bill. And also can edit or delete from this view page.
- For sell pharmacist will go to sell and sell the product. . From their pharmacist can view transaction of sell and view the status of transaction.
- By depending on status pharmacist will pay the due bill. And also can edit or delete from the view page
- Pharmacist will generates report after every transaction by clicking on billing.

Blood Bank Management

User Requirement

- Lab operator can add, view, edit blood type with cost.
- Lab operator can add donor details, edit donor details and delete donor details.
- Lab operator can view, add, edit, delete, and blood transaction.
- Lab operator can view available blood.

System Requirement

- By clicking on blood bank type Lab operator can view blood type with cost and add new blood type.
- By clicking on blood donor Lab operator view blood donor, from there he/she can add new donor and also delete and edit from this view page.
- Lab operator view transaction by clicking on blood transaction. From this view page Lab operator can add new transaction, edit existence transaction and delete transaction if needed.
- In Available blood bank menu Lab operator can view available blood are in blood bank.

Manage laboratory

User Requirement:

- Lab operator can view test, add new test, edit test and also delete test.
- Lab operator can view inpatient test result and outpatient test result and also can edit, delete test result of patient.
- Lab operator can bill the lab test transaction and generate the billing report.
- Lab operator generate the rest report.

System Requirement:

- From add test menu Lab operator view all test with cost and from this view page Lab operator add new test details, edit test details, and delete test details.
- In Patient result menu clicking Lab operator view the Lab operator patient test result and add new test result and edit, delete new patient result details. From there Lab operator generate patient test result.
- In test billing Lab operator bill generate report for patient lab test.

Manage Shift:

User Requirement:

- Sub admin can view shift and add shift, edit shift, delete shift.

System Requirement:

- From shift menu, sub admin will add new shift change the shift by editing and delete the shift from the same page.

3.4 Functional Requirements:

- Admin can add user, role and also can give access to the role for user.
- Admin can view user, role and also edit and delete user and role.
- Admin can change his profile also
- Sub Admin can add employee, shift, bed and also can view, edit, delete employee, shift, bed.
- Receptionist can register patient and also view admitted and discharge patient.
- Receptionist can view available bed.
- Receptionist can discharge and taking the payment system of the patient.
- Lab operator can add test, test result of patient, blood type, blood quantity, donor and add transaction of outpatient test bill.
- Lab operator can view test, test result of patient, blood type, available blood quantity, transaction and also edit and delete them.
- Pharmacist can add purchase record, sells record, product record, supplier record, purchase transaction, sells transaction.

3.5. Non-Functional Requirements:

- ✓ Check Database Connection
- ✓ Prevent duplicate data
- ✓ Check required field.
- ✓ Check validation of data.
- ✓ Without login cannot access any menu any function.

3.6 System Design & Specification

The Unified Modeling Language (UML)

The Unified Modeling Language (UML), which has been chosen here to design the Hospital Management System, is the successor to the wave of analysis and design. Although it is not a method and a modeling language, but most methods consist, at least in principle, of both a modeling language and a process. Modeling language is mainly the graphical notation that methods use to express designs, whereas process is their advice on what steps to take in doing a design.

Although coding is the real point of software development, but diagrams of the user interfaces has great impact to the user who is going to execute software. UML consists of diagrams as well as precise description of those diagrams, but the fundamental reason for using UML is the communication. UML is used because it seems easier to communicate than alternatives. Natural language is always imprecise and makes it difficult for complex ideas. UML is such a modeling language, which can do a certain amount of precision and will never be lost in details. UML is invaluable because it helps to get an overall view of the system that aimed to develop. UML has various types of visual diagrams and each type of diagram's functionality is different also. In this project, several UML diagrams have been used for designing and specification of the Hospital Management System

3.6.1 Use Case Diagram of the System:

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

Figure 3.1 show the relationship between use case and actor. The use case diagram for the e-Commerce system is as under:

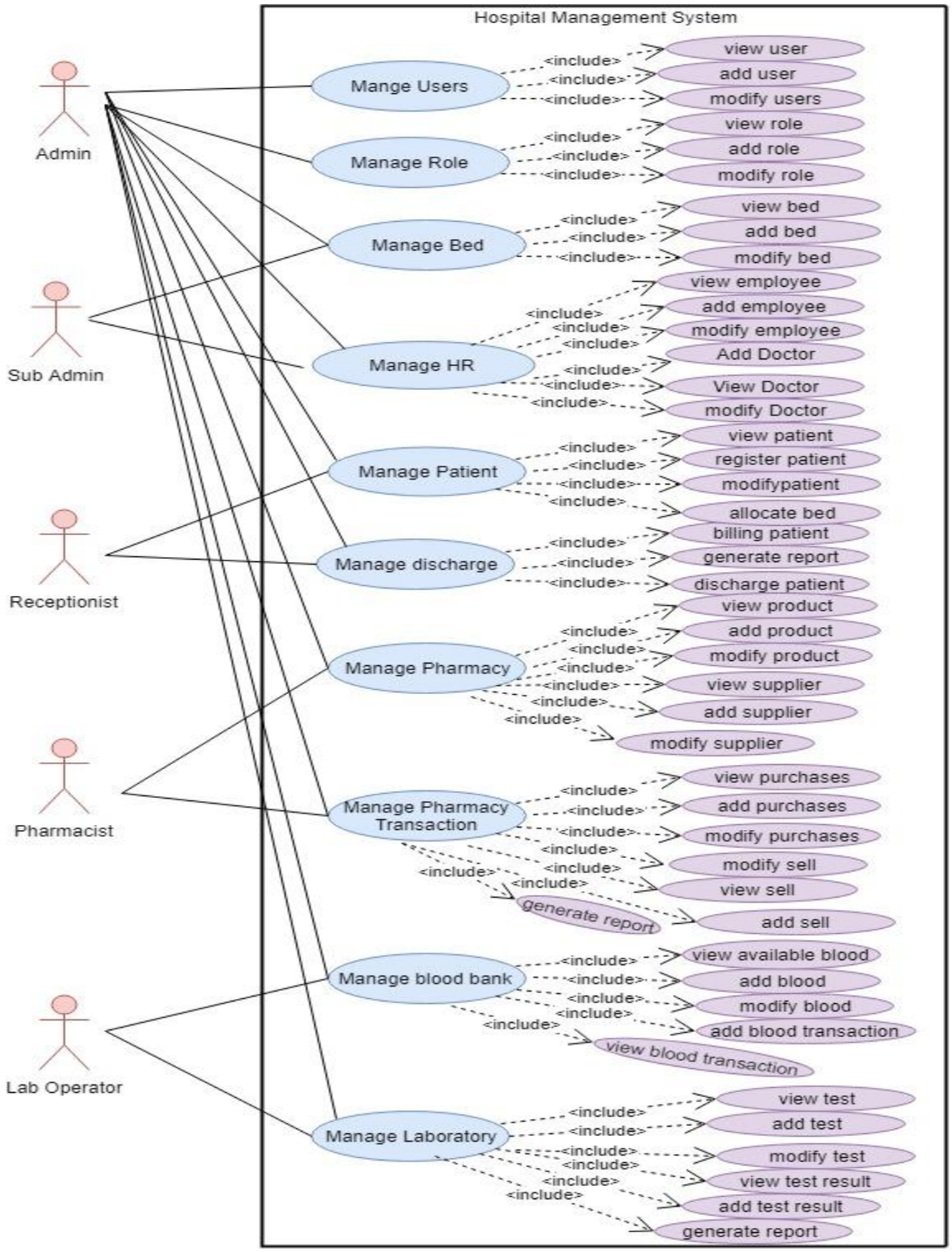


Figure 3.1: Use case Diagram

3.6.2 Use Case Text

Use Case Title: Role Management

Actor: Admin.

Description: I have to login first from the login panel. After successful login i can create role from role menu and can give role name and by clicking on checkbox I can give access of menu and function to this role. I can see my entire role in role page. From there I can edit and delete role

Use Case User Management

Actor: Admin.

Description: After successful login I can see all users by clicking on user menu. From this page I can edit delete. While I am creating user that I can give access of role to the users.

Use Case Title: Hospital Building Management.

Actor: Sub Admin

Description: Here I can add new floor, room, bed, to the system from the hospital building menu. I can see entire building floor, room, and be d information and edit and delete them

Use Case Title: HR Management.

Actor: Sub Admin

Description: Here from the HR menu I can add new employee. Here I can see all employees information ,If I want I can edit them and delete them.

Use Case Title: Patient management.

Actor: Receptionist.

Description: I can manage all inpatient and outpatient here from clicking on patient menu. There have all menu inpatient outpatient registration. By clicking on that menu I can complete patient registration. I can give bed allocation while I am admitting a patient. I can see available bed on clicking available bed. Here I can see all patient billing information and also generate a bill report and discharge patient.

Use Case Title: Laboratory management

Actor: Lab operator.

Description: I can add new test by clicking on add test. I can add test result of patient and generate test report. I can also add mange blood bank by adding donor, blood quantity. I can see available blood by clicking on available blood.

Use Case Title: Pharmacy

Actor: Pharmacist

Description: I can add new medicine, supplier, category, company, purchase and sell from pharmacy menu. From there I can see the entire information of supplier, purchases and sales, I can see the sue payment of customer and add new payment. And generate purchases and sales report.

Chapter-4

System Planning

In this Analysis and Development of Developing e-Commerce System many functions are working parallel. Functions of this proposed system are given below:

4.1 Functions of Proposed system

Table 4.1:Function of Proposed System

Manage Role	**[F1]**
Mange User	**[F2]**
Mange Hospital Building	**[F3]**
Manage HR	**[F4]**
Manage Shift	**[F5]**
Manage Patient	**[F6]**
Manage Doctor	**[F7]**
Manage Laboratory	**[F8]**
Manage Pharmacy	**[F9]**

4.2. System Project Planning

Software project planning commences with a set of activities that collectively called software project planning. Through the software project planning I estimate the work to be done, the resources that will be required, the time that will elapse from start to finish and finally I analyze the project to determine whether it is feasible or not.

The following activities of software project planning that have followed in this project are:

- ✓ Function Point Estimation of the project
- ✓ Process Based Estimation
- ✓ Task scheduling
- ✓ Project Schedule Chart
- ✓ Personnel requirements
- ✓ Resource requirements

- ✓ Estimation of the software cost

4.2.1 System Project Estimation

The accuracy of software project estimation reflects its impact on a number of things →
Properly estimated the size of the product to build the ability to translate the size estimation into human effort, calendar time and money. The degree to which the project plan reflects the abilities of the software team. The stability of the product requirements and the environment that supports the software engineering effort.

Software size estimation is the most important matter that I have to consider during the software project. If the software size not calculated properly, then this will cause various problems such as scheduling problems, budget problem etc. as the project goes on. Before estimating the software size, I have to confirm that software scope is bounded.

4.2.2 Scopes of the Project

- ➢ Manage Role
- ➢ Mange User
- ➢ Mange Hospital Building
- ➢ Manage HR
- ➢ Manage Shift
- ➢ Manage Patient
- ➢ Manage Doctor
- ➢ Manage Laboratory
- ➢ Manage Pharmacy

4.2.3. Function-Oriented Metrics

Function point based estimation focuses on information domain values rather that software values.
Function points are computed by comparing five information domain characteristics.

The Five Components of Function Points

Data Functions

- ✓ Internal Logical Files
- ✓ External Interface Files

Transactional Functions

- ✓ External Inputs
- ✓ External Outputs
- ✓ External Inquiries

Number of external inputs – Each user input that provides distinct application-oriented data to the software is counted inputs should be distinguished from inquires.

Number of external outputs – Each user output that provides application-oriented information to the user is counted.

Number of external inquires – An inquiry defined as an on-line input those results in the generation of some immediate software response in the form of an on-line output. Each distinct inquiry counted.

Number of Internal Logical files – Each logical internal file is a logical grouping of data that resides within the application's boundary and is maintained via external inputs.

Numbers of external interfaces – All machine-readable interfaces that used to transmit information to another system counted.

Functional Complexity - The first adjustment factor considers the Functional Complexity for each unique function. Functional Complexity is determined based on the combination of data groupings and data elements of a particular function. The number of data elements and unique groupings are counted and compared to a complexity matrix that will rate the function as low, average or high complexity. Each of the five functional components (ILF, EIF, EI, EO and EQ) has its own unique complexity matrix. The following is the complexity matrix for External Outputs.

After the components have been classified as one of the five major components (EI's, EO's, EQ's, ILF's or EIF's), a ranking of low, average or high is assigned. For transactions (EI's,

EO's, EQ's) the ranking is based upon the number of files updated or referenced (FTR's) and the number of data element types (DET's). For both ILF's and EIF's files the ranking is based upon record element types (RET's) and data element types (DET's). A record element type is a user recognizable subgroup of data elements within an ILF or EIF. A data element type is a unique user recognizable, non-recursive, field.

The value adjustment factor (VAF) is based on 14 general system characteristics (GSC's) that rate the general functionality of the application being counted. The degrees of influence range on a scale of zero to five, from no influence to strong influence.
Once these data has collected, a complexity value is associated with each count And the FP estimated is calculated by the equation

FP estimated=count-total*[0.65+0.01*∑Fi]

The following formula is used to estimate the function points →

- ✓ *FP count = {(opt+4*likely+pessimistic)/6}*weight*
- ✓ Complexity adjustment factor = [0.65 + 0.01 x Sum of Factor Values]
- ✓ *FP* estimated = count total x Complexity adjustment factor
- ✓ Function Point Estimation = Total FP estimated/No. of function point per day.

4.2.4.Function point Estimation DET & RET

Table 4.2: Function point Estimation DET & RET

Manage Role:

Functionality	Input	Output
Admin can add & edit role	Role_id,Role_name,menu_id,function_id	Display Successful & store data or error message
Admin can delete role	Delete role_id	Display success or error

Manage User:

Functionality	Input	Output
Admin can add & edit user	Name,user_name.role_id,reg_date	Display Successful & store data or error message
Admin can view user	View data	Display user information
Admin can delete user	Delete user	View rest of the user list

Manage Hospital Building:

Functionality	Input	Output
Sub-admin can add Bulding	Building_id.bulding_name	Display Successful & store data or error message
Sub-admin can add Floor	Flood_id,floor_name,building_id	Display Successful & store data or error message
Sub-admin can view floor	View floor	Display floor information
Sub-admin can add room	Room_id,room_name,floor_id,building_id	Display Successful & store data or error message
Sub-admin can view room	View data	Display room information
Sub-admin can add & edit bed	Bed_id,bed_name,room_name,building_name,floor_name	Display Successful & store data or error message
Sub-admin can view bed	View data	Display bed information

Manage HR:

Functionality	Input	Output
Sub_Admin can add employee	Employee_id,type_id,first_name,last_name,gender,date_of_journey,num,photo	Display Successful or error message , store data.
Sub_Admin can view employee information	View data	Display employee information
Sub_admin can delete employee	Delete employee	Delete the employee data

Manage shift:

Functionality	Input	Output
Sub_Admin can add shift	Shift_id,days,start_time,end_time, shift	Display Successful or error message , store data.

Sub_Admin can view employee shift information	View data	Display shift information

Manage Doctor:

Functionality	Input	Output
Sub_admin can add & edit doctor	Doc_id,designation_id,specialization_id,department_id,doc_name,address,phone,email,photo	Display Successful and user ID or error message, store data.
Sub_admin can view doctor information	View data	Display doctor information
Sub_admin can delete doctor	Delete doctor by id	Delete doctor information

Manage patient:

Functionality	Input	Output
Receptionist can add In patient	Patient_id,reg_no,name,reg_date,address,phone,email,depart_id,ref_doc_id,bed_id	Display Successful & store data or error message
Receptionist can view In patient	View data	Display in patient information
Receptionist can view available bed	View data	Display available bed information
Receptionist can view inpatient billing	View data	View patient billing details
Receptionist can discharge patient	Discharge patient	Generate billing report
Receptionist can add out patient	Out_patient_id,name,reg_no,phone,dep_id,ref_doc_id,amount	Display Successful & store data or error message

Manage laboratory:

Functionality	Input	Output
Lab-operator can add test	Test_id,test_name,unit,minimum_range,maxrenge,cost	Display Successful & store data or error message
Lab-operator can view test	View testing information	Display data
Lab-operator can add in_patient test	In_patient_test_result_id,test_result,test_id,description	Display Successful & store data or error message
Lab-operator can view	View data	Display in_patient list

in_patient test		
Lab-operator can add Out_patient test	out_patient_test,result_id,test_result,test_id,description	Display Successful & store data or error message
Lab-operator can view Out_patient test	View data	Display Out_patient list
Lab_operator can add & edit blood type	Blood_type_id,type_name,cost	Display Successful & store data or error message
Lab_operator can add blood qty	Id,blood_type_id,collection,address,qty,date	Display Successful & store data or error message
Lab_operator can add in_patient blood_transaction	Id,reg_no,date,blood_type_id,qty	Display Successful & store data or error message
Lab_operator can view in_patient blood_transaction	View data	Display transaction information
Lab_operator can view available blood	View data	View available blood information

Manage Pharmacy:

Functionality	Input	Output
Pharmacist Can view medicine	View data	Display medicine list
Pharmacist can add &edit purchase product	Id,supplier_id,purchase_id,product_id,ex_date,sell_price,buy_price,qty,total,sub_total,paid,due	Display Successful & store data or error message
Pharmacist can view purchase product transaction	View data	Display transaction information
Pharmacist can add &edit sell product	Sell_id,patient_id,medicine_price, quamtity,total,sub_total,paid,due	Display Successful & store data or error message
Pharmacist can view sell product transaction	View data	Display transaction information

4.2.5.Identifying Complexity:

Table 4.3: Identifying Complexity for Transaction function

Manage Role:

Transaction Function	Fields	FTRs	DETs
Admin can add & edit role(2*EI)	Fields:Role_id,Role_name,menu_id,function_id File:role,menu_access,function_access	3	4
Admin can delete role(EI)	Fields:Role_id,Role_name,menu_id,function_id File:role,menu_access,function_access	3	4

Manage User:

Transaction Function	Fields	FTRs	DETs
Admin can add & edit user(2*EI)	Fields:id,Name,email,user_name.role_id,reg_date,password,status File:user,role	2	8
Admin can view user(EQ)	Fields:Name,email,user_name.role_id,reg_date, status File:user,role	2	6
Admin can delete user(EI)	Fields:id,Name,email,user_name.role_id,reg_date,password,status File:user,role	2	8

Manage hospital Building:

Transaction Function	Fields	FTRs	DETs
Sub-admin can add Bulding(EI)	Fields: Building_id.bulding_name File:building	1	2
Sub-admin can add Floor(EI)	Fields:floor_id,floor_name,building_id File:building,floor	2	3
Sub-admin can view floor(EQ)	Fields:floor_name,building_id File:building,floor	2	2
Sub-admin can add room(EI)	Fields:Room_id,room_name,type,floor_id,building_id ,cost File: room,building,floor	3	6
Sub_admin can view room(EQ)	Fields:room_name,type,floor_id,building_id,cost File:room,building,floor	3	5
Sub-admin can add & edit bed(2*EI)	Fields:Bed_id,bed_name,room_name,building_name,floor_name,status File:bed,floor,room,building	4	6
Sub-admin can view bed(EQ)	Fields:bed_name,room_name,building_name,floor_name,status File:bed,floor,room,building	4	5

Manage HR:

Transaction Function	Fields	FTRs	DETs
Sub_Admin can add employee(EI)	Fields:Employee_id,type_id,first_name,last_name,gender,DOB,blood_group,date_of_journey,date_of_retirement,email,num,emergency_num,relation,qualification,age,photo File:employee,employee_type	2	16
Sub_Admin can view employee information(EQ)	Fields:type_id,first_name,last_name,gender,DOB,blood_group,date_of_journey,date_of_retirement,email,num,emergency_num,relation,qualification,age,photo File:employee,employee_type	2	15
Sub_admin can delete employee (EI)	Fields:Employee_id,type_id,first_name,last_name,gender,DOB,blood_group,date_of_journey,date_of_retirement,email,num,emergency_num,relation,qualification,age,photo File:employee,employee_type	2	16

Manage shift:

Transaction Function	Fields	FTRs	DETs
Sub_Admin can add shift(EI)	Fields: Shift_id,days,start_time,end_time,shift File:shift	1	5
Sub_Admin can view employee shift information(EQ)	Fields: days,start_time,end_time,shift File:shift	1	4

Manage Doctor:

Transaction Function	Fields	FTRs	DETs
Sub_admin can add & edit doctor(2*EI)	Fields:Doc_id,designation_id,specialization_id,department_id,doc_name,address,phone,consult_fee,shift_id,email,photo File:doctor,specialization,designation,shift,department	5	11
Sub_admin can view doctor information(EQ)	Fields:designation_id,specialization_id,department_id,doc_name,address,phone,consult_fee,shift_id,email,photo File:doctor,specialization,designation,shift,department	5	10
Sub_admin can delete doctor(EI)	Fields:Doc_id,designation_id,specialization_id,department_id,doc_name,address,phone,consult_fee,shift_id,email,photo File:doctor,specialization,designation,shift,department	5	11

Manage Patient:

Transaction Function	Fields	FTRs	DETs
Receptionist can add In patient (EI)	Fields:Patient_id,reg_no,name,reg_date,address,dob,phone,email,gender,age,diseases,depart_id,ref_doc_id,bed_id,status File:in_patient,department,bed,doctor	4	15
Receptionist can view In patient(EQ)	Fields:reg_no,name,reg_date,address,dob,phone,email,gender,age,diseases,depart_id,ref_doc_id,bed_id,status File:in_patient,department,bed,doctor	4	14
Receptionist can view available bed(EQ)	Fields:floor_name,room_name,bed_name File:floor,room,bed	3	3
Receptionist can view inpatient billing(EO)	Fields:reg_num,patient_name,day,bed_bill,test_bill,operation_bill,total_bill,ser_charge File:in_patient	1	8
Receptionist can discharge patient(EO)	Fields:reg_num,patient_name,day,bed_bill,test_bill,operation_bill,total_bill,ser_charge File:in_patient	1	8
Receptionist can add out patient(EI)	Fields:Patient_id,reg_no,name,reg_date,address,dob,phone,email,gender,age,diseases,depart_id,ref_doc_id,status File:out_patient,department,doctor	3	14

Manage Laboratory:

Transaction Function	Fields	FTRs	DETs
Lab-operator can add test(EI)	Fields:Test_id,test_name,unit,minimum_range,maxrenge,cost,description File:add_test	1	7
Lab-operator can view test(EQ)	Fields:test_name,unit,minimum_range,maxrenge,cost,description File:add_test	1	6
Lab-operator can add in_patient test(EI)	Fields:In_patient_test_result_id,test_result,test_id,description File:add_test_inpatient	1	7
Lab-operator can view in_patient test(EQ)	Fields:test_result,test_id,description File:add_test_inpatient	1	6
Lab-operator can add Out_patient test(EI)	Fields:In_patient_test_result_id,test_result,test_id,description File:add_test_outpatient	1	7
Lab-operator can view Out_patient test(EQ)	Fields:test_result,test_id,description File:add_test_outpatient	1	6
Lab_operator can add	Fields:blood_type_id,type_name,cost	1	3

& edit blood type(2*EI)	File:blood_type		
Lab_operator can add blood qty(EI)	Fields:id,blood_type_id,collection,address,date,qty File:blood_quantity	1	6
Lab_operator can add in_patient blood_transaction(EI)	Fields: Id,reg_no,date,blood_type_id,qty File:blood_transaction_inpatient	1	5
Lab_operator can view in_patient blood_transaction(EQ)	Fields: reg_no,date,blood_type_id,qty File:blood_transaction_inpatient	1	4
Lab_operator can view available blood(EQ)	Fields:blood_type,quantity File:available_blood	1	2

Manage Pharmacy:

Transaction Function	Fields	FTRs	DETs
Pharmacist Can view medicine(EQ)	Fields:Test_id,test_name,unit,minimum_range,maxrenge,cost,description File:product	1	10
Pharmacist can add &edit purchase product(2*EI)	Fields:Id,supplier_id,purchase_id,product_id,ex_date,sell_price,buy_price,qty,total,sub_total,paid,due File:purchase,purchase_transaction,product	3	13
Pharmacist can view purchase product transaction(EQ)	Fields:supplier_id,purchase_id,product_id,ex_date,sell_price,buy_price,qty,total,sub_total,paid,due File:purchase_transaction	1	11
Pharmacist can add sell product(EI)	Fields:Sell_id,patient_id,medicine,price,quamtity,total,sub_total,paid,due File:sales,sales_transaction	2	9
Pharmacist can view sell product transaction(EQ)	Fields:Sell_id,patient_id,medicine,price,quamtity,total,sub_total,paid,due File:sales	1	9

Table 4.4: Identifying Complexity for Data function

Data function	Fields	RETs	DETs
user(ILF)	Fields:id, Name,user_name.role_id,reg_date email,phone,password,status	**1**	**9**
bed(ILF)	Fields:Bed_id,bed_name,room_id,building_id,floor_id,status File:room,buiding,floor,bed	**4**	**6**

employee(ILF)	Fields:Employee_id,type_id,first_name,last_name,gender,DOB,blood_group,date_of_journey,date_of_retirement,email,num,emergency_num,relation,qualification,age,photo File:employee,employee_type	**2**	**16**
Doctor(ILF)	Fields:Doc_id,designation_id,specialization_id,department_id,doc_name,address,phone,consult_fee,shift_id,email,photo File:doctor,specialization,designation,shift,department	**5**	**11**
In_patient(ILF)	Fields:Patient_id,reg_no,name,reg_date,address,dob,phone,email,gender,age,diseases,depart_id,ref_doc_id,bed_id,status File:in_patient,department,bed,doctor	**4**	**15**
Out_patient(ILF	Fields:Patient_id,reg_no,name,reg_date,address,dob,phone,email,gender,age,diseases,depart_id,ref_doc_id,status File:out_patient,department,doctor	**3**	**14**
Available_blood(ILF)	Fields:blood_type,quantity File:available_blood	**1**	**2**
product(ILF)	Fields:pid,company_id,name,category_id,price,qty,ex_date,status File:product	**1**	**7**
Add_test(ILF)	Fields:Test_id,test_name,unit,minimum_range,maxrenge,cost,description File:add_test	**1**	**7**
Blood_transaction_inpatient(ILF)	Fields: Id,reg_no,date,blood_type_id,qty File:blood_transaction_inpatient	**1**	**5**

4.2.6.Unadjusted function point contribution

Table 4.5:Unadjusted function point contribution for Transaction Function

Manage Role:

Transaction Function	FTRs	DETs	Complexity	UFP
Admin can add & edit role(2*EI)	3	4	Average	8
Admin can delete role(EI)	3	4	Average	6
				Total-14

Manage User:

Transaction Function	FTRs	DETs	Complexity	UFP

Admin can add & edit user(2*EI)	2	8	Average	8
Admin can view user(EQ)	2	6	Average	4
Admin can delete user(EI)	2	8	Low	3
				Total-15

Manage hospital Building:

Transaction Function	FTRs	DETs	Complexity	UFP
Sub-admin can add Bulding(EI)	1	2	Low	3
Sub-admin can add Floor(EI)	2	3	Low	3
Sub-admin can view floor(EQ)	2	2	Low	3
Sub-admin can add room(EI)	3	6	High	6
Sub_admin can view room(EQ)	3	5	Low	3
Sub-admin can add & edit bed(2*EI)	4	6	High	12
Sub-admin can view bed(EQ)	4	5	Average	4
				Total-34

Manage HR:

Transaction Function	FTRs	DETs	Complexity	UFP
Sub_Admin can add employee(EI)	2	16	Average	4
Sub_Admin can view employee information(EQ)	2	15	Average	4
Sub_admin can delete employee (EI)	2	16	Average	4
				Total-12

Manage shift:

Transaction Function	FTRs	DETs	Complexity	UFP
Sub_Admin can add shift(EI)	1	5	Low	3
Sub_Admin can view	1	4	Low	3

employee shift information(EQ)				
				Total-6

Manage Doctor:

Transaction Function	FTRs	DETs	Complexity	UFP
Sub_admin can add & edit doctor(2*EI)	5	11	High	12
Sub_admin can view doctor information(EQ)	5	10	High	6
Sub_admin can delete doctor(EI)	5	11	High	6
				Total-24

Manage Patient:

Transaction Function	FTRs	DETs	Complexity	UFP
Receptionist can add In patient (EI)	4	15	High	6
Receptionist can view In patient(EQ)	4	14	High	6
Receptionist can view available bed(EQ)	3	3	Low	3
Receptionist can view inpatient billing(EO)	1	8	Low	4
Receptionist can discharge patient(EO)	1	8	Low	4
Receptionist can add out patient(EI)	3	14	High	6
				Total-29

Manage Laboratory:

Transaction Function	FTRs	DETs	Complexity	UFP
Lab-operator can add test(EI)	1	7	Low	3
Lab-operator can view test(EQ)	1	6	Low	3
Lab-operator can add in_patient test(EI)	1	7	Low	3
Lab-operator can view	1	6	Low	3

in_patient test(EQ)				
Lab-operator can add Out_patient test(EI)	1	7	Low	3
Lab-operator can view Out_patient test(EQ)	1	6	Low	3
Lab_operator can add & edit blood type(2*EI)	1	3	Low	6
Lab_operator can add blood qty(EI)	1	6	Low	3
Lab_operator can add in_patient blood_transaction(EI)	1	5	Low	3
Lab_operator can view in_patient blood_transaction(EQ)	1	4	Low	3
Lab_operator can view available blood(EQ)	1	2	Low	3
				Total-36

Manage Pharmacy:

Transaction Function	FTRs	DETs	Complexity	UFP
Pharmacist Can view medicine(EQ)	1	10	Low	3
Pharmacist can add &edit purchase product(2*EI)	3	13	High	12
Pharmacist can view purchase product transaction(EQ)	1	11	Low	3
Pharmacist can add sell product(EI)	2	9	Average	4
Pharmacist can view sell product transaction(EQ)	1	9	Low	3
				Total-25

Table 4.6:Unadjusted function point contribution for Data Function

Data function	RETs	DETs	Complexity	UFP
user(ILF)	**1**	**9**	**Low**	**7**
bed(ILF)	**4**	**6**	**Low**	**7**
employee(ILF)	**2**	**16**	**Low**	**7**

Doctor(ILF)	**5**	**11**	**Low**	**7**
In_patient(ILF)	**4**	**15**	**Low**	**7**
Out_patient(ILF	**3**	**14**	**Low**	**7**
Available_blood(ILF)	**1**	**2**	**Low**	**7**
product(ILF)	**1**	**7**	**Low**	**7**
Add_test(ILF)	**1**	**7**	**Low**	**7**
Blood_transaction_inpatient(ILF)	**1**	**5**	**Low**	**7**
				Total-70

Table 4.7: . Performance and Environmental Impact

Number	**Factor**	**Value**
1	Does the system require reliable backup and recovery?	4
2	Are specialized data communications required?	2
3	Are there any distributed processing functions?	0
4	Is performance critical?	1
5	Does the system run in existing operational environment?	2
6	Does the system require on-line data entry?	5
7	Input transaction over multiple screens	1
8	Are the ILFs updated on-line?	2
9	Are the input, output, files or inquiries complex?	1
10	Is the internal processing complex?	1
11	Is the code designed to be reusable?	4
12	Are conversation and installation included in the design?	2
13	Is the system designed for multiple installations?	3
14	Is the system designed to facilitate change and ease of use?	2
	Σ (Fi)	**30**

4.2.8. Function Point Estimation

UFP for TF = 195

UFP for DF = 70

Total UFP = 265

Value Adjustment Factor (VAF) = (0.65+ (0.01×30)) = 0.99 ~ 1

Adjusted Function Point (AFP) = UFP × VAF

= 265× 1

= 265

Efforts for PHP = AFP × Productivity for PHP

= 265× 15.5

= 4107.5 person hours

Work per day = 2945 / 8 person days

= 513.44 person days

Work per person = 513.44/4 person days

= 128.36 per person days = 4.2 month per person

= 4.2 ≈ 4 month needed for per person

4.2.9 Process Based Estimation

In process-based estimation, process is decomposed into a relatively small set of tasks and the effort required to accomplish each task is estimated. Process based estimation begins with a delineation of software functions obtained from the project scope.

Table 4.8: Process Based Estimation

Activity	**CC**	**Planning**	**Engineering**		**Construction**		**Imp.**	**Total**
Function			**Analysis**	**Design**	**Code**	**Test**		
F1	0.003	0.0155	0.12	0.15	0.34	0.12	0.06	
F2	0.001	1.0054	1.17	1.17	2.24	2.12	0.05	
F3	0.002	0.112	0.19	0.25	0.45	0.11	0.06	
F4	0.001	1.111	2.12	1.25	2.45	2.16	0.04	
F5	0.004	0.113	0.3	0.25	0.55	0.13	0.05	
F6	0.005	0.212	3.54	1.25	3.75	2.13	0.06	
F7	0.003	1.034	1.36	0.25	2.45	0.30	0.04	
F8	0.1	0.225	1.42	1.65	2.45	2.15	0.1	
F9	0.01	1.113	2.74	0.50	2.74	1.14	0.09	
Total	**0.129**	**3.91**	**13**	**8.72**	**16.75**	**10.49**	**0.55**	**54**
Effort	**.3%**	**6%**	**25.02%**	**16.5%**	**31%**	**20.2%**	**1%**	**100%**

4.2.10. Effort Based Estimation:

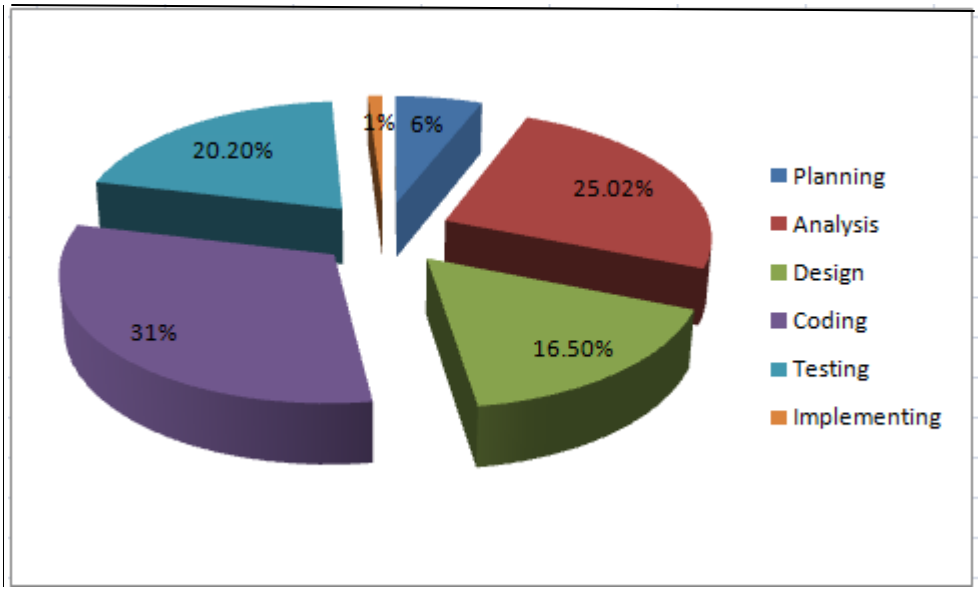


Figure 4.1: Effort Based Estimation

4.2.11 Project Schedule

Work Breakdown The following under noted work breakdown activities needs to be carried out for every products of the project:

a. Design & Specification: Design & specification phase activity deals with modeling the component/product.

b. Coding: Programming of the design for the component/product carries out in this phase.

c. Test Plans: In this phase test plans need to be developed to test each product/component and then integration of all the products/components.

d. Testing: Under the test plans, code needs to be tested at each component level and also when it is integrated with other components. Test results need to be evaluated at individual component level and at the level of integration also.

Project Scheduling Chart

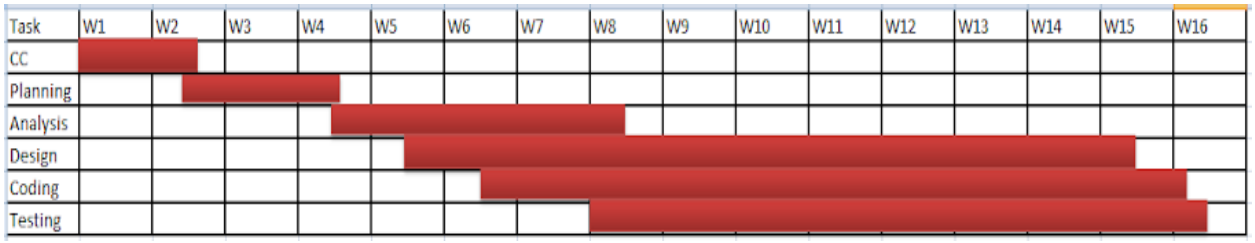


Figure 4.2: Project Scheduling Chart

4.2.12 Cost Estimation

Personnel cost

Number of days in a year = 365

Number of government holidays in a year =22

Number of weekly holidays in a year =52

Total number of working days in a year =365-(52+22) =291 days

Total number of working days per months in a year = 291 /12 =24.25 days ≈24 days

Organization working hours per day = 8 hours

Organization working hours per month=24*8= 192 hours

Table 4.9: Personnel cost

Position	**Salary/month**	**Salary/hour**
System Analyst	18,000.00	93.75
Designer	8000.00	41.7
Coder	10,000.00	52.1
Tester	8,000.00	41.7

Salary for the Technical Staffs Engaged

Job Title : System Analyst

Persons Required : One

Duration : 4 Months

Weeks : 16

Total working hours =24*8*4 = 768 hours

Total salary of Analyst in 4 months = TK 768*93.75 = TK 72000

Total Salary Of Designer in 4 months= TK 768*41.7 = TK 32000

Total Salary Of Coder in 4 months= TK 768*52.1 = TK 40012.8

Total Salary Of Tester in 4 months= TK 768*41.7 = TK 32000

Table 4.10: Personnel cost

Designation	**Person**	**Working Hours**	**Salary**	**Total Salary**
System Analyst	One	768	72000	176012.8
Designer	One	768	32000	
Coder	One	768	40012.8	
Tester	One	768	32000	

Hardware cost

Cost of a Laptop = 38000

Computer life = 3 years

Computer Usage=3+2+1=6

Year 1=3/6=0.5 Year 2 =2/6=.334 Year 3 =1/6=0.1667

Computer Cost in year 2=38000*.1667=6334.6 BDT

Software cost

Table 4.11:Software cost

Name	Price
MS Office 2010	8500
Windows 8.1	7500
Total	16000 BDT

Other Cost

Table 4.12:Other Cost

Name	Price
Transport	1200
House rent	2000
Electric bill	1500
Extra	500
Total	5200 BDT

Total Cost

Table 4.13:Total cost

Purpose	Amount
Salary	176012.8
Hardware	6334.6
Software	16000
Other	5200
Total	203547.4 BDT

Chapter-5

Risk Engineering

5.1 Risk Management

Risk analysis and management are a series of works that help a system development team to understand and manage uncertainty. Many problems can arise while developing a system. A risk is a potential problem. It may happen may not.

There are several steps to analyze and manage risks. The first step is risk identification. Next each risk is analyzed to determine the likelihood that it will occur and the damage that it will do if it does occur. Once this information is established risks are remarked. Finally, a plan is developed to manage those risks with high probability and impact.

There are different **Stages** of risks. They are:

- Risk Identification
- Risk Classification
- Risk Assessment
- Risk Analysis
- Risk Management Implementation

Risk Identification is the process of detecting potential risks or hazards through data collection. A range of data collection and manipulation tools and techniques exists. The team is using both automated and manual techniques to collect data and begin to characterize potential risks to Web resources. Web crawling is one effective way to collect information about the state of Web pages and sites.

Risk Classification is the process of developing a structured model to categorize risk and fitting observable risk attributes and events into the model. The team combines quantitative and qualitative methods to characterize

Risk Assessment is the process of defining relevant risk scenarios or sequences of events that could result in damage or loss and the probability of these events. Many sources focus on risk assessment. Rosenthal describes the characteristics of a generic standard for risk assessment as "transparent, coherent, consistent, complete, comprehensive, impartial, uniform, balanced, defensible, sustainable, flexible, and accompanied by suitable and sufficient guidance.

Risk Analysis determines the potential impact of risk patterns or scenarios, the possible extent of loss, and the direct and indirect costs of recovery. This step identifies vulnerabilities, considers the willingness of the organization to accept risk given potential consequences, and develops mitigation responses.

Risk Management Implementation defines policies, procedures, and mechanisms to manage and respond to identifiable risks. The implemented program should balance the value of assets and the direct and indirect costs of preventing or recovering from damage or loss.

To make comprehensive care of web based system we must consider the following points:

Hardware and software environment, including any upgrades to the operating system and web server, the installation of security patches, the removal of insecure services, use of firewalls etc.

Administrative procedures,including contracting with reputable service providers, renewing domain name registration etc.

Network configuration and maintenance, including load balancing, traffic monitoring, traffic management, usage monitoring etc.

Backup and archiving policies and procedure,including the choice of backup media, media replacement interval, number of backups made and storage location.

There are different categories of risk that should be considered in any software project. The following categories of risk have been considered in this project:

- Project Risks
- Technical Risk
- Business Risk

Project risk: These risks threaten the project plan. If these risks become real, it is likely that the project schedule will slip and that costs will increase. Project risks identify potential budgetary, schedule, personnel, resource, customer and requirement problems and their impact on the software project.

Technical risk: these risks threaten the quality and timeliness of the software to be produced. If a technical risk becomes a reality, implementation may become difficult or impossible. Technical risks identify potential design, implementation, interface, verification and maintenance problems. Moreover, specification ambiguity, technical uncertainly, technical obsolescence are also risk factors.

Business risk: These risks threaten the viability of the software to be built. The business risks can be market risks, strategic risks, changing business needs, management risks, budget risks etc.

Table 5.1: Business risks

RMMM Plane No. BR01		**Type of risk**
Description Scenario of the Risk	Not pay the entire installment of Software cost.	Business Risk
Impact	Disaster	
Prevention	We should make a good communication between customers and ensure that the entire installment will be completed.	
Cure	The only course of action available would be find out the reason and come in a solution.	
Status	We did not face this problem yet.	

RMMM Plane No. BR02		**Type of risk**
Description Scenario	Late Delivery of the project.	Business Risk
Impact	Disaster	
Prevention	Steps have been taken to ensure a timely delivery by determining the scope of project.	
Cure	The only course of action available would be to request	

	an extension to the deadline from the customer.	
Status	Our Project is completed in time.	

RMMM Plane No.BR03		Type of risk
Description Scenario of the Risk	End Users refuse to accept System	Project Risk
Impact	Catastrophic	
Prevention	The software will be developed with the end user in mind. The user-interface will be designed in a way to make use of the program convenient and pleasurable to the user.	
Cure	The program will be thoroughly examined to find the reasons that why this is so. Specifically the user interface will be investigated and if necessary we will come for alternative solution.	
Status	We are strictly monitoring this risk.	

Table 5.2 Technical risks

Technical risks - threaten product quality and the timeliness of the schedule.

RMMM Plane No. TR01		Type of risk
Description Scenario of the Risk	Computer Crash	Technical Risk
Impact	Disaster	
Prevention	We should take proper follow up of computers. We also can take regular data backup every day. We can use IPS to stop unexpected shutdown.	

Cure	If our computer has been creased then we will restore backup.	
Status	We are not facing such kind of problem yet.	

RMMM Plane No. TR02		**Type of risk**
Description Scenario of the Risk	Technology Does Not Meet Specifications	Technical Risk
Impact	Disaster	
Prevention	This ensures that the product we are producing, and the specifications of the customer are equivalent	
Cure	The customer should be immediately notified and whatever steps necessary to rectify this problem should be done. Preferably a meeting should be held between the development team and the customer to discuss at length this issue.	
Status	We are not facing such kind of problem yet.	

RMMM Plane No. TR03		**Type of risk**
Description Scenario of the Risk	Poor training skill in team members to train the client.	Technical Risk
Impact	Disaster	
Prevention	The training team should have a clear knowledge about the entire functionality of the software. System analyst need to ensure and monitor it while training session start.	

Cure	We should arrange a meeting with the train team and come to a point to solve this problem.	
Status	We are not facing such kind of problem yet.	

Table 5.3: Project risks

Project risks - threaten the project plan.

RMMM Plane No.PR 01		**Type of risk**
Description Scenario of the Risk	Lack of Development Experience.	Project Risk
Impact	Disaster	
Prevention	Each member of the team should watch and see areas where another team member may be weak.	
Cure	The members who have the most experience in a particular area will be required to help for overcome problem arises for this risk.	
Status	We are not facing such kind of problem yet.	

RMMM Plane No. PR 02		**Type of risk**
Description Scenario of the Risk	Poor Quality Documentation	Project Risk
Impact	Disaster	
Prevention	Meetings will be held routinely to offer documentation suggestions and topics. The progress on documentation will also have to monitor in each meeting.	

Cure	We will call a meeting and discuss about quality improvement of documentation. The addition of new topics or removal of unnecessary topics into the documentation will assigned to responsible person.	
Status	We are monitoring it.	

RMMM Plane No. PR03		**Type of risk**
Description Scenario of the Risk	Poor Comments in Code	Project Risk
Impact	Marginable	
Prevention	A formal written standard must be established to ensure quality of comments in all code.	
Cure	We should call a meeting with the development team to get rid of this problem and improve the quality of comments in code.	
Status	We are monitoring this risk.	

5.3 Risk Monitoring

The goal of the risk monitoring and management plan is to identify as many potential risks as possible. To help determine what the potential risks are evaluated using the checklists found. Hotel management system it needs to be concern that everything is alright or not if there is some problem in the system then customer will get the wrong information and they will not be satisfied for the services. For this kind of problem it turns on a bad impact on the reputation of hotel. The project will then be analyzed to determine any project-specific risks. When all risks have been identified, they will then be evaluated to determine their probability of occurrence, and ho will be affected if they do occur. Plans will then be made to avoid each risk, to track each risk to determine if it is more or less likely to occur, and to plan for those risks should they occur.

Chapter-6

Analysis & Designing

- To describe what the customer require
- To establish a basis for the creation of a software design
- To define a set of requirements that can be validated once the software is built.

6.1. Activity Diagram

Activity diagrams describe the workflow behavior of a system. Activity diagrams are similar to state diagrams because activities are the state of doing something. The diagrams describe the state of activities by showing the sequence of activities performed. Activity diagrams can show activities that are conditional or parallel.

Activity diagram for admin management

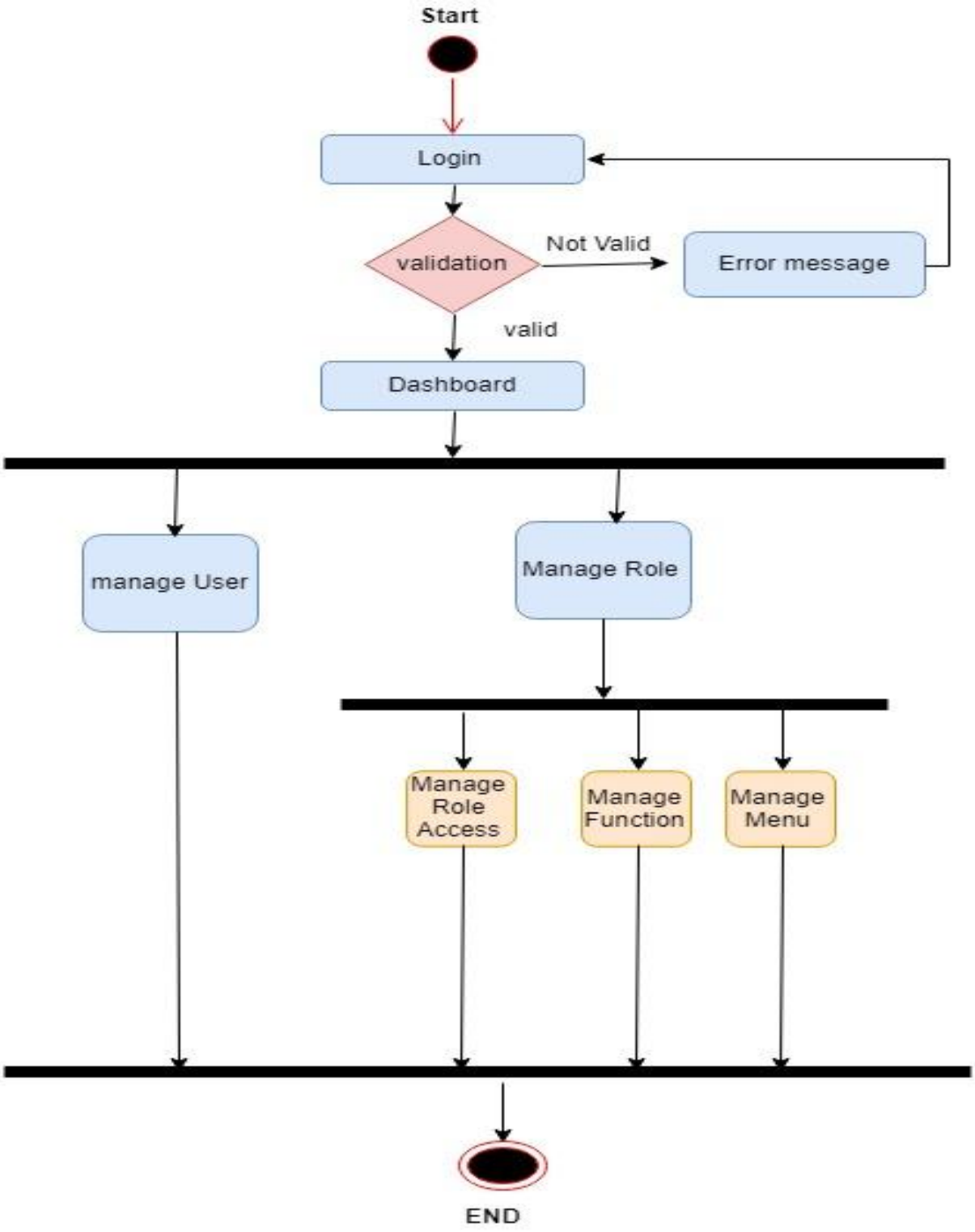


Figure 6.1: Activity diagram for Admin Management

Activity diagram for Sub-Admin Management

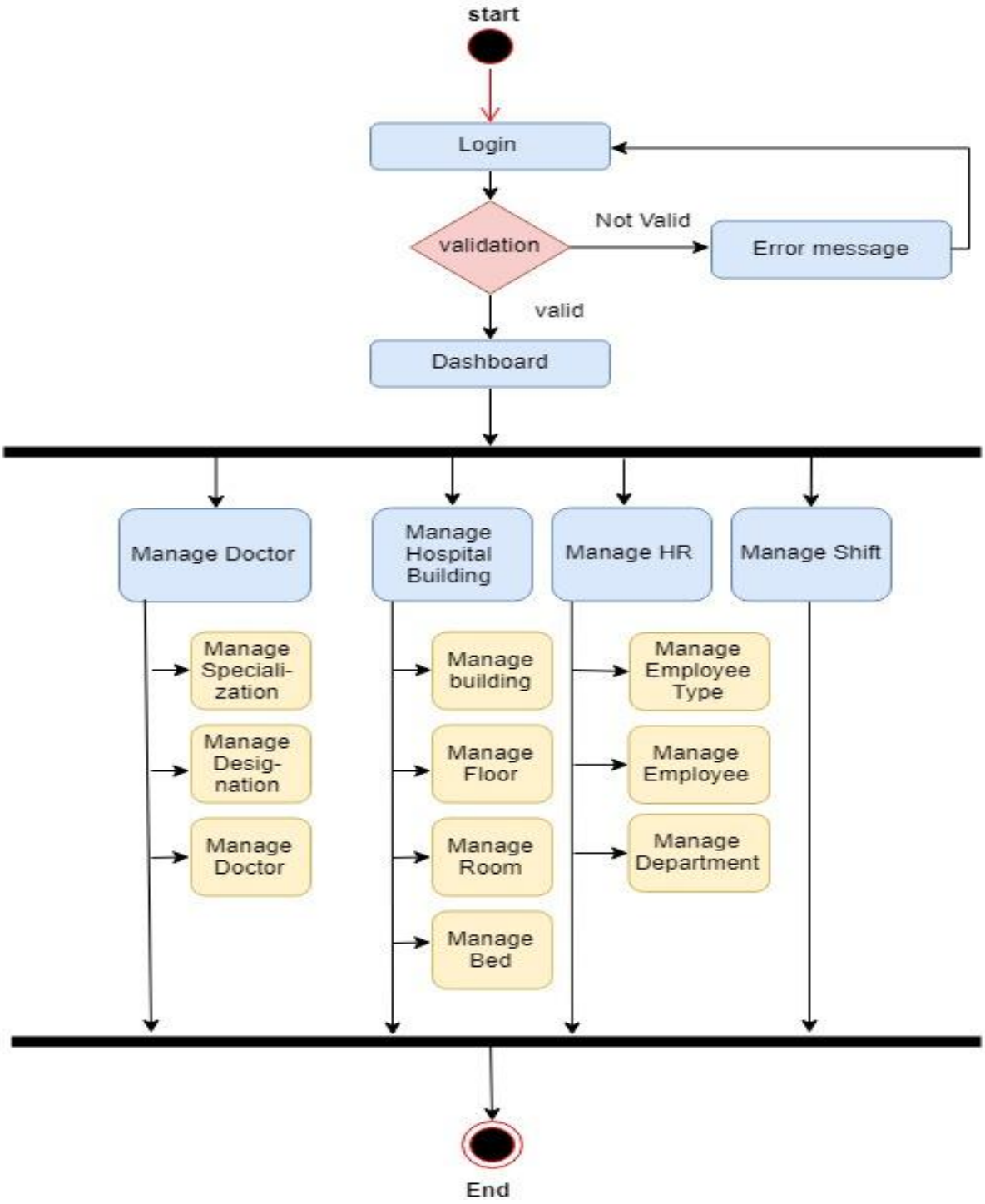


Figure 6.2: Activity diagram for Admin Management

Activity diagram for Receptionist Management

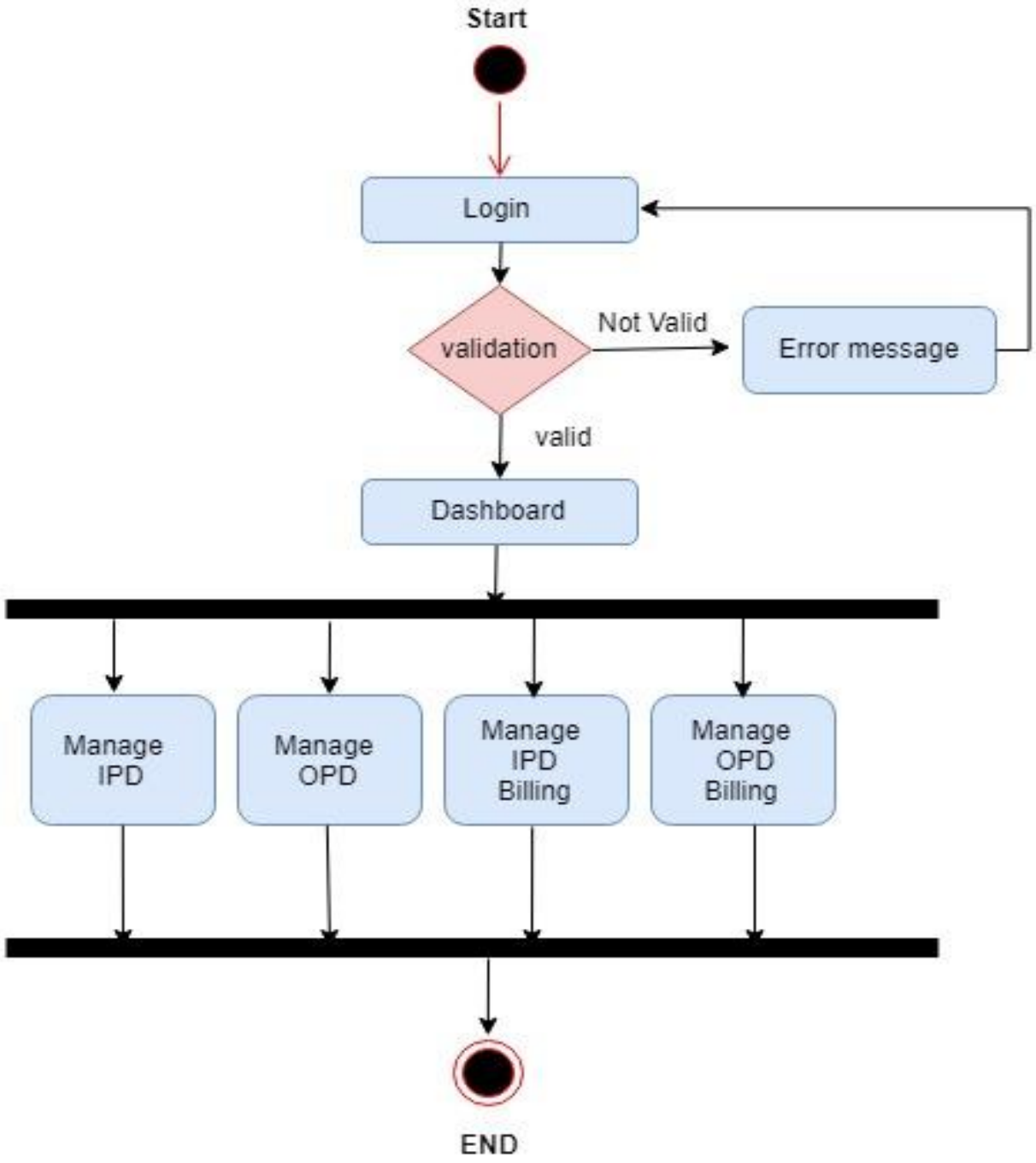


Figure 6.3: Activity diagram for Receptionist Management

Activity diagram Lab-Operator Management

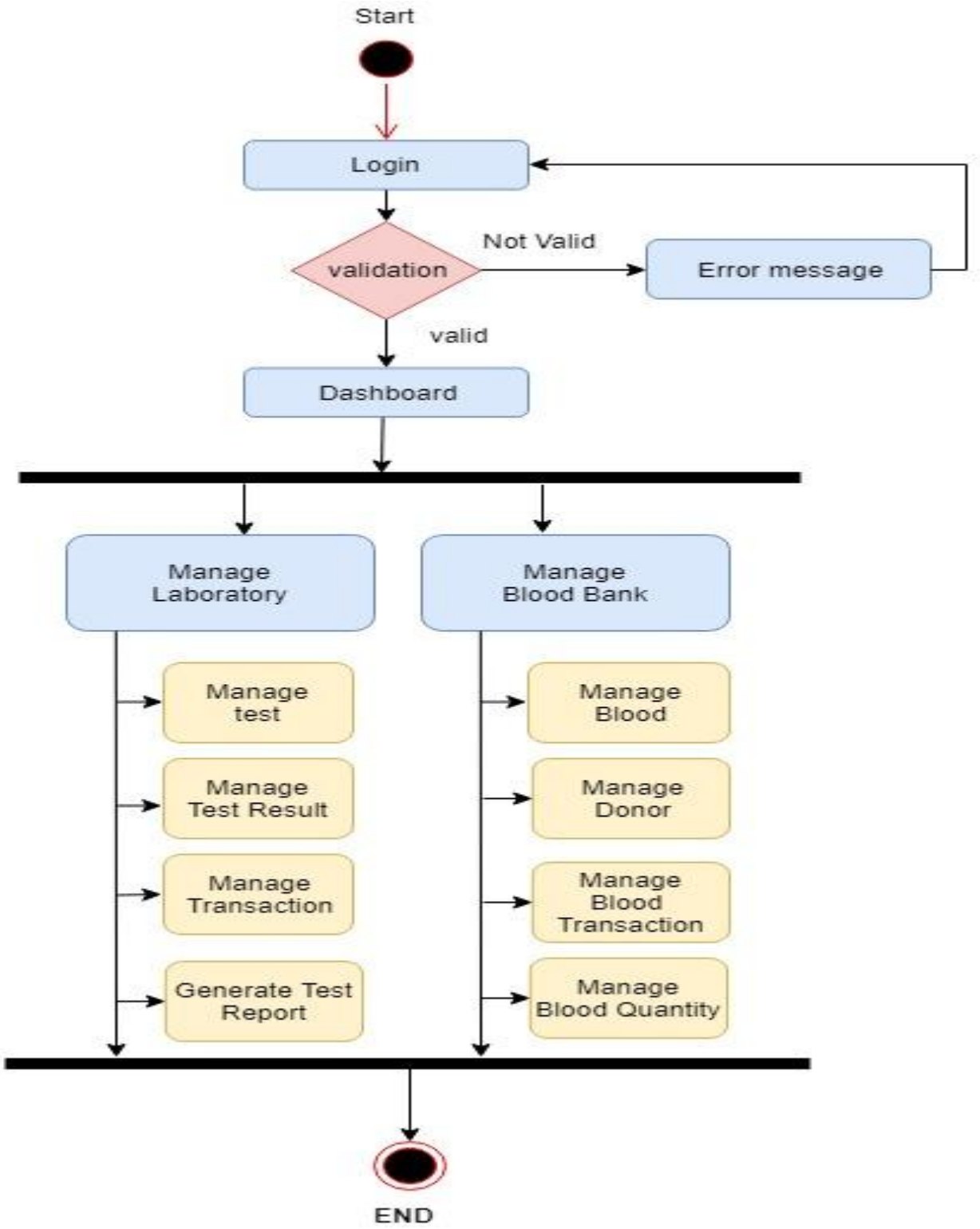


Figure 6.4: Activity diagram for Lab-Operator Management

Activity diagram for Pharmacist Management

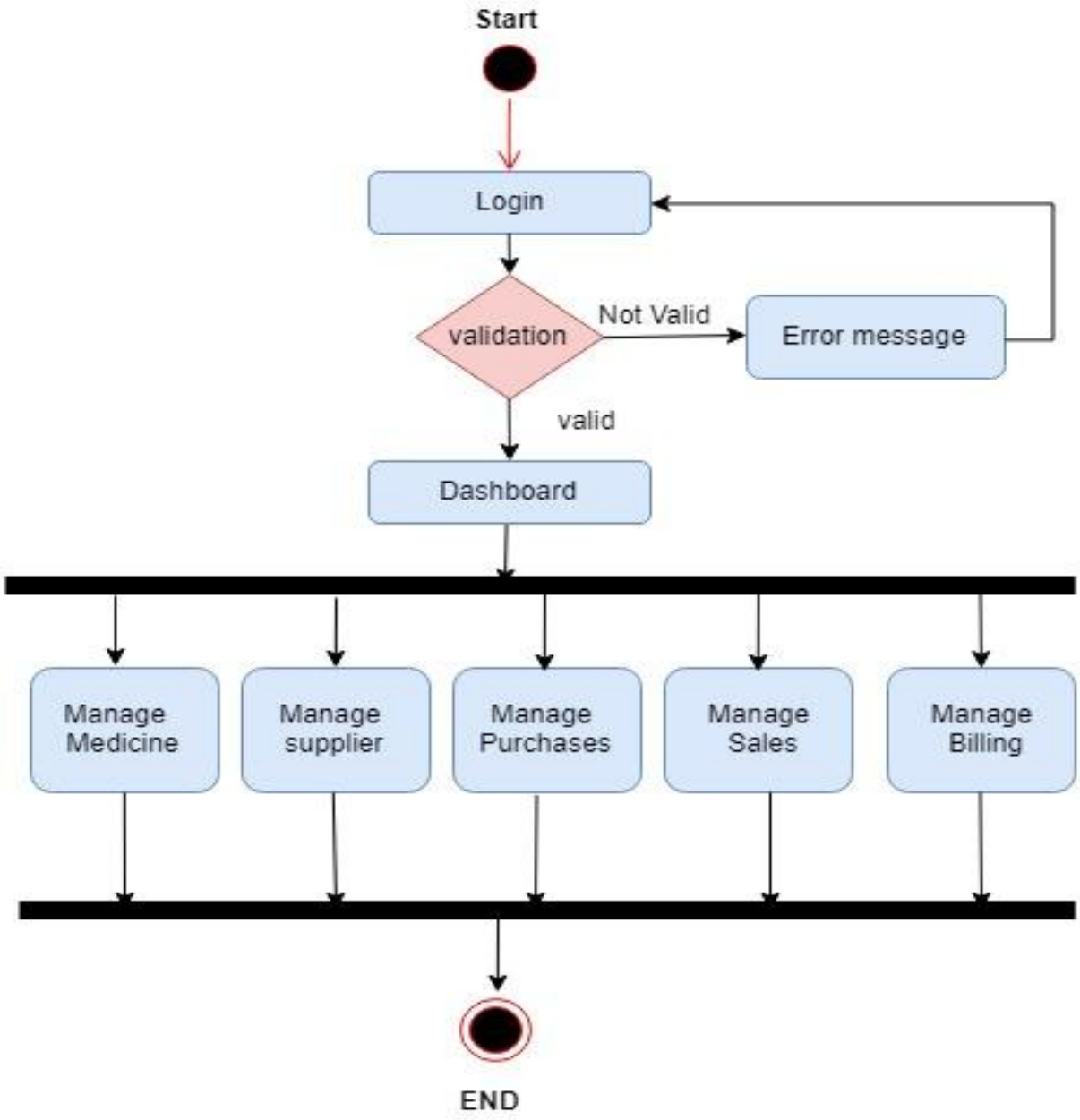


Figure 6.5: Activity diagram for Pharmacist Management

6.2. Swimlane Diagram

Swimlane diagram for Hospital Management

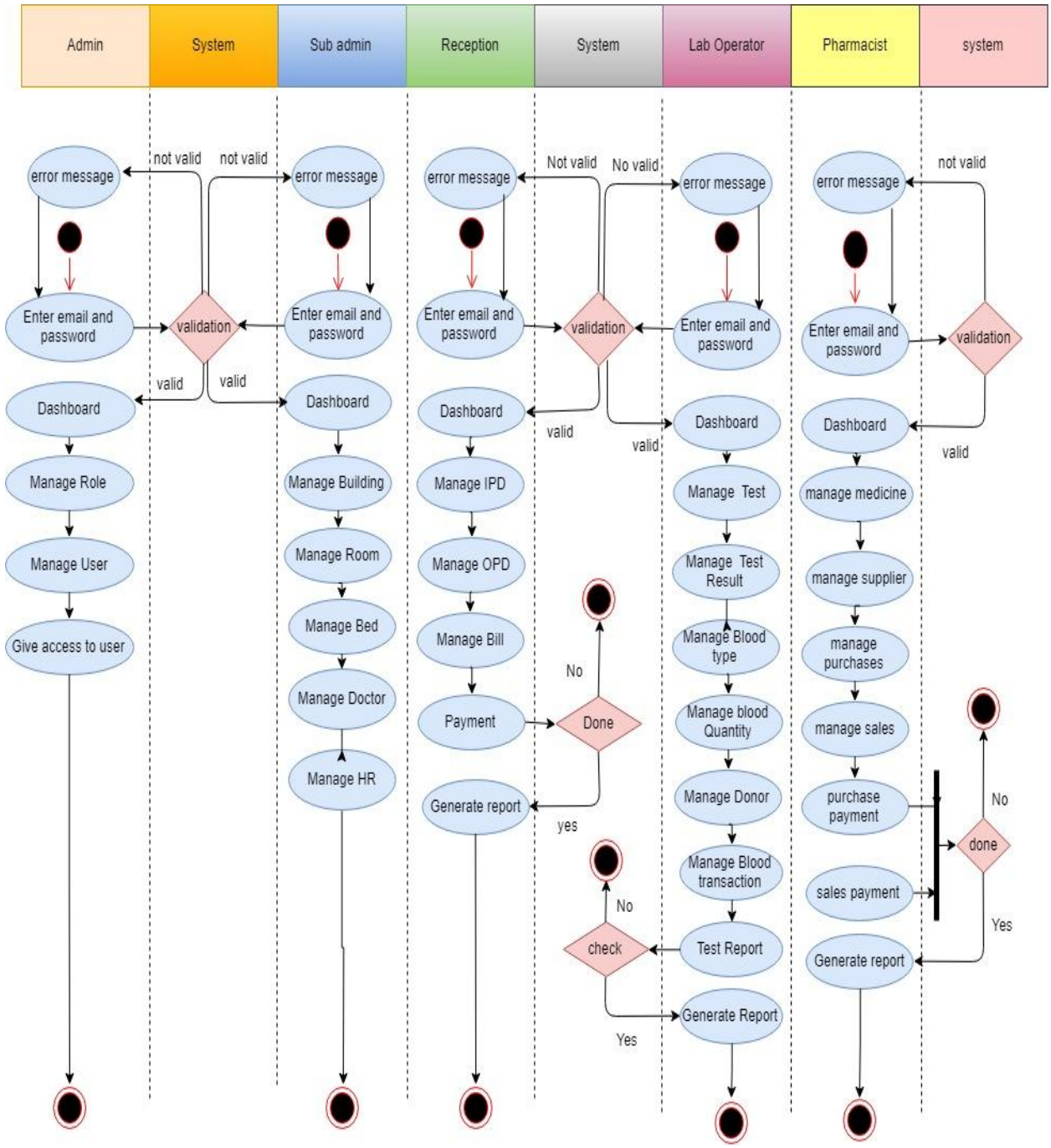


Figure 6.6: Swimlane diagram for Hospital Management

6.3:ERD

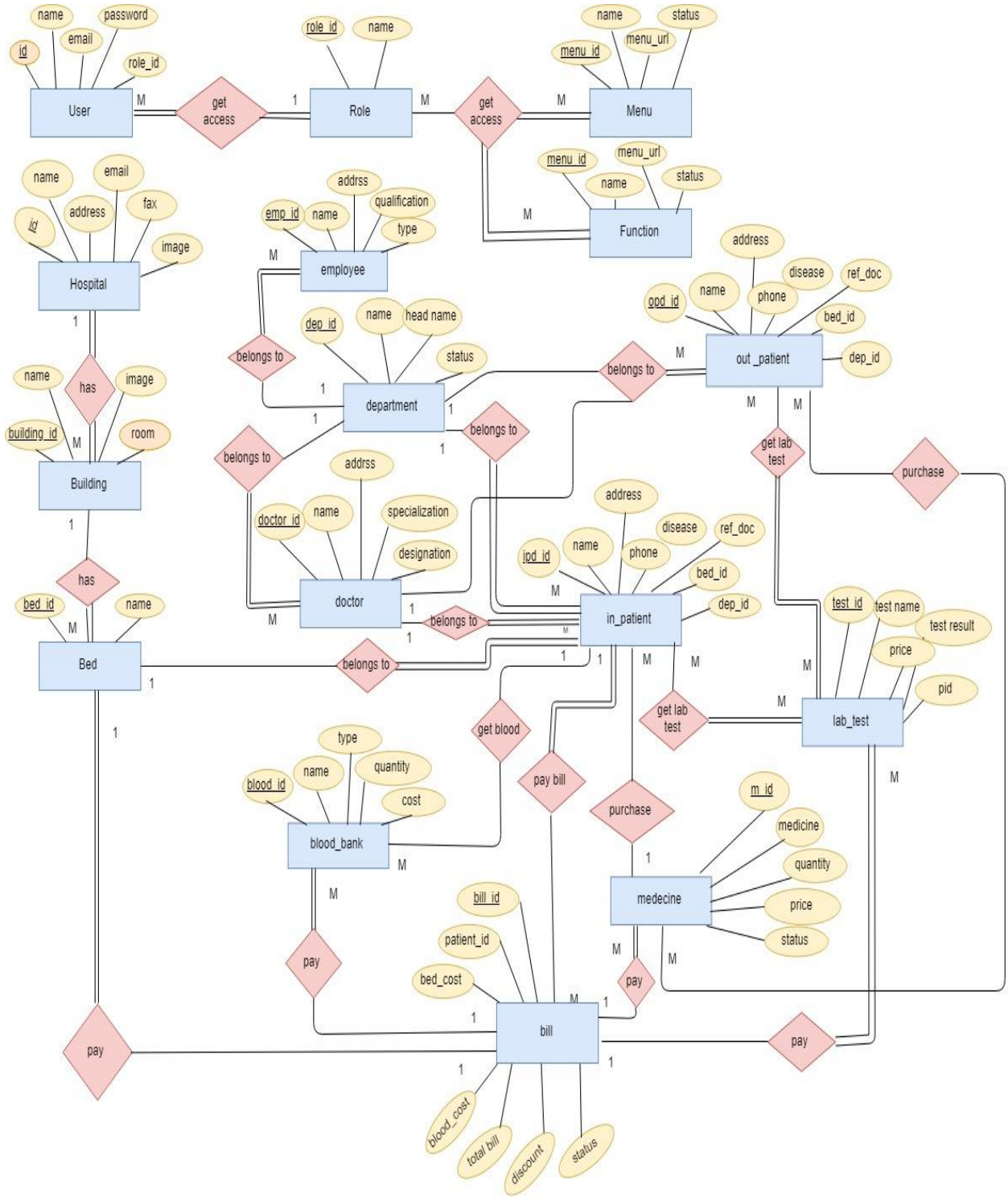


Figure 6.7:ERD of Proposed System

6.4:DFD

Context Level Diagram:

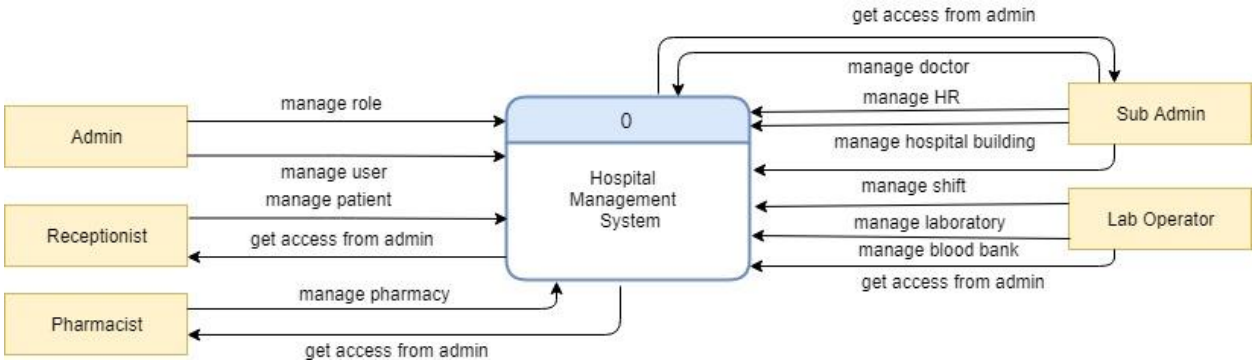


Figure 6.8:Context level Diagram

Level 1 Diagram:

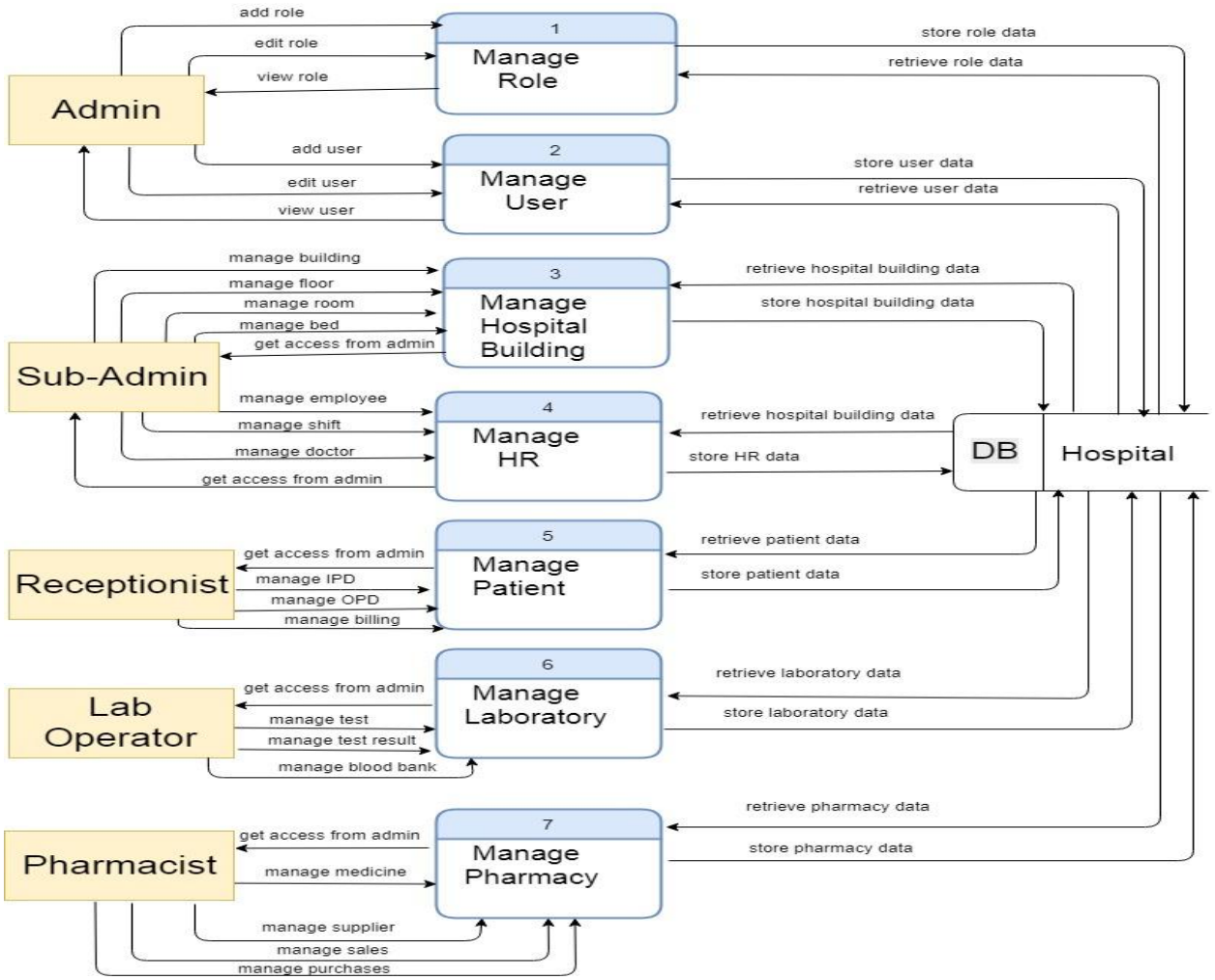


Figure 6.9:Level 1 Diagram

Level 2 diagram for process 1(Manage Role):

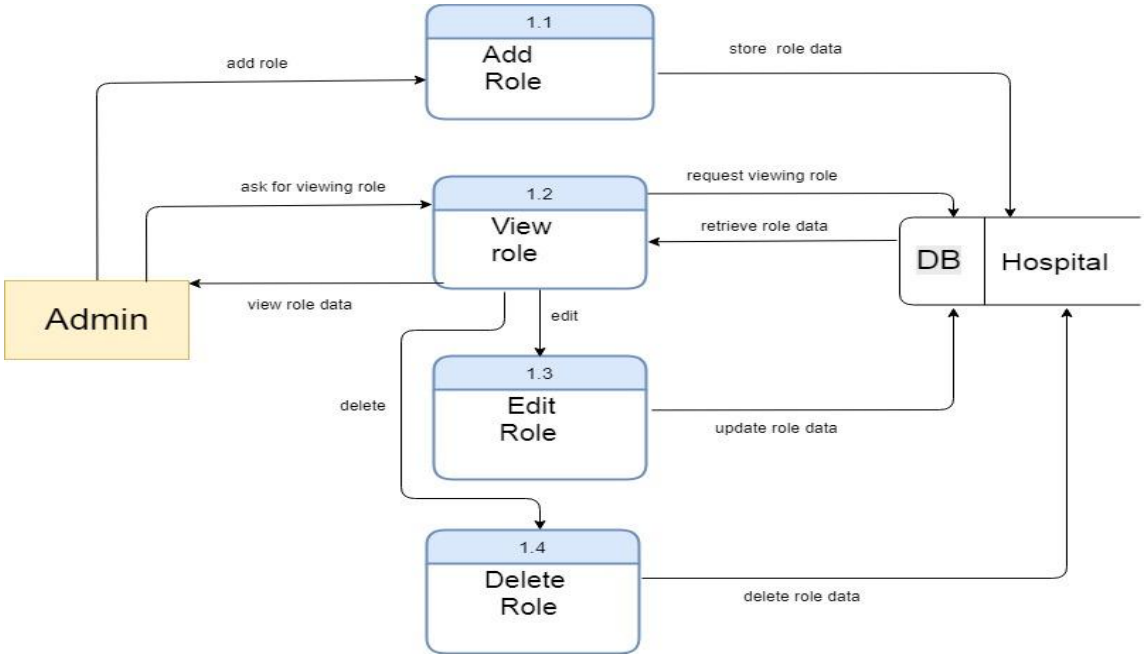


Figure 6.10: Level 2 diagram for process 1(Manage product)

Level 2 diagram for process 2(Manage User):

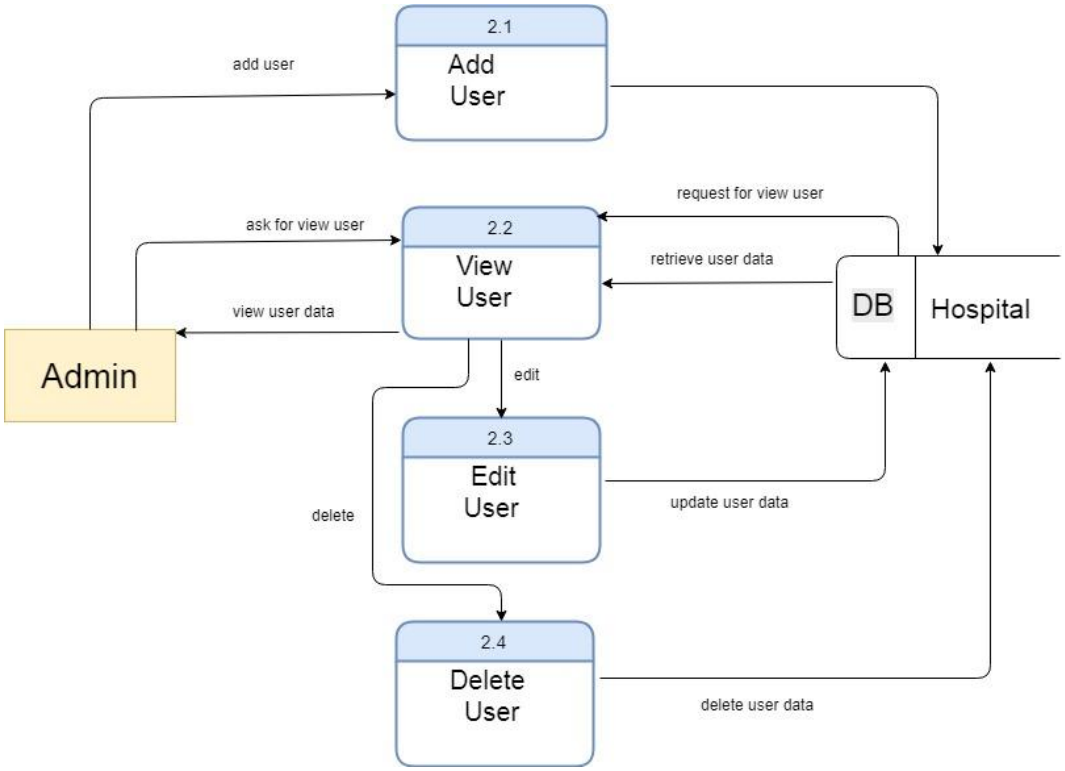


Figure 6.11: Level 2 diagram for process 1(Manage User)

Level 2 diagram for process 3(Manage Hospital Building):

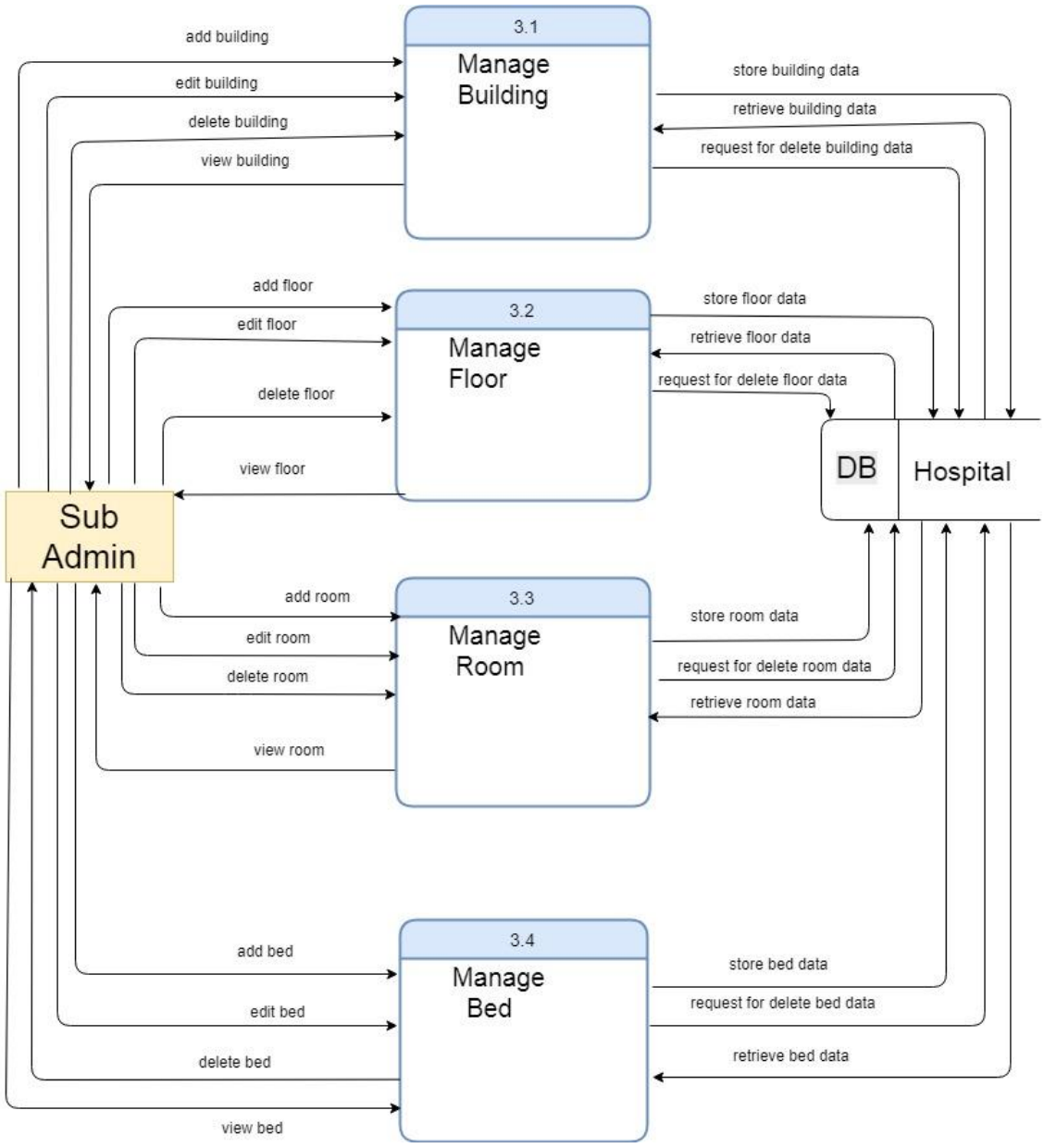
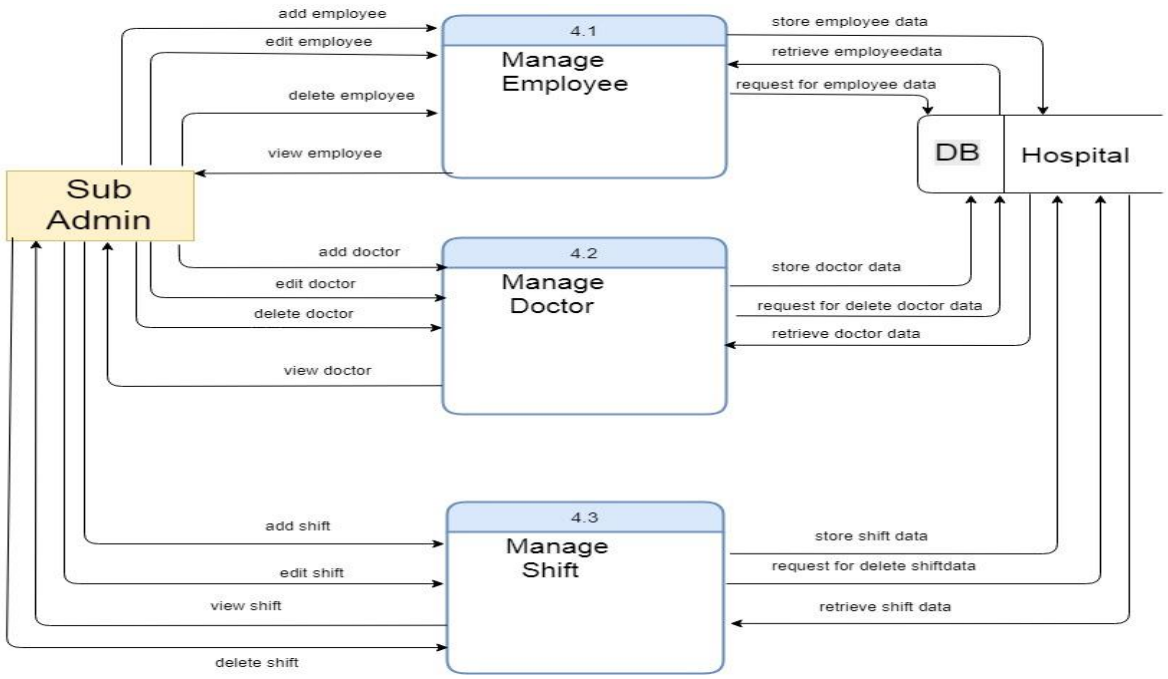


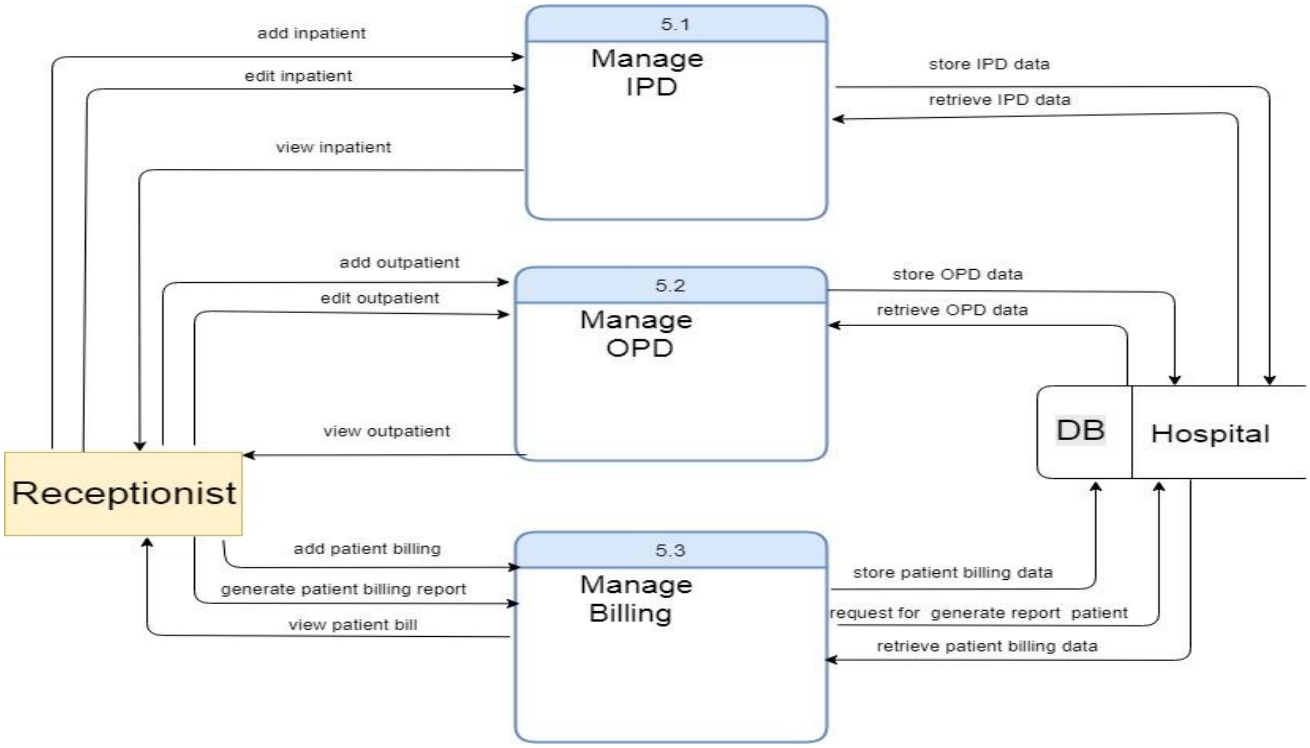
Figure 6.12: Level 2 diagram for process 3(Manage Hospital Building)

Level 2 diagram for process 4(Manage HR):

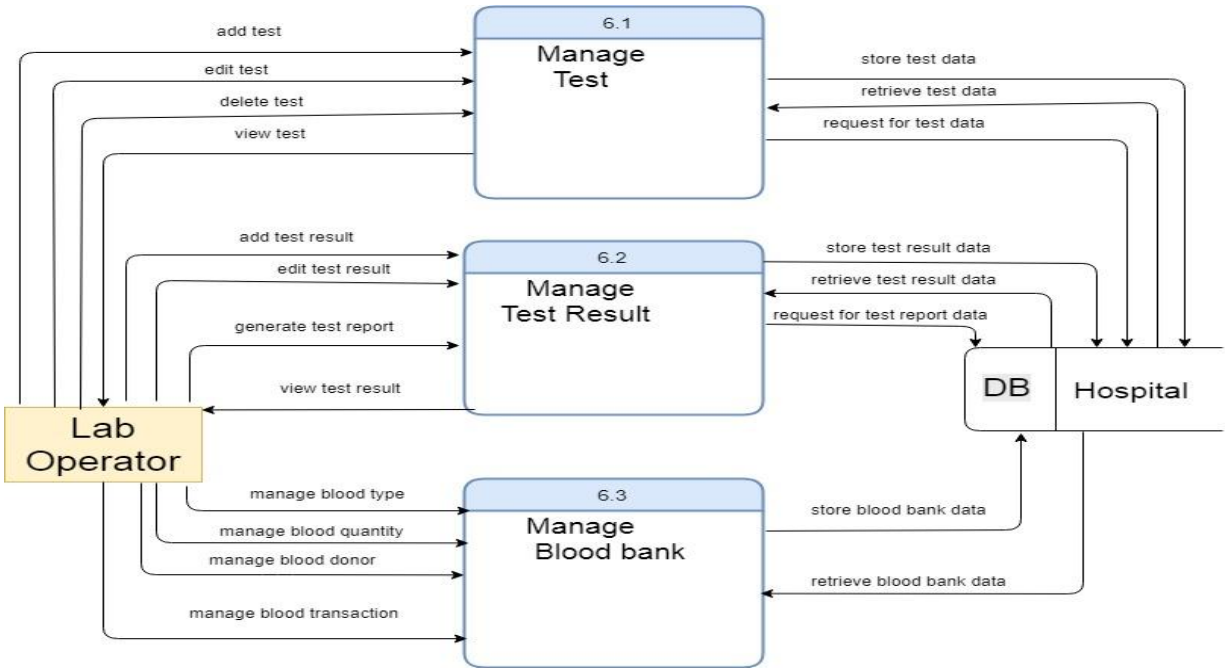


Level 2 diagram for process 4(Manage HR):

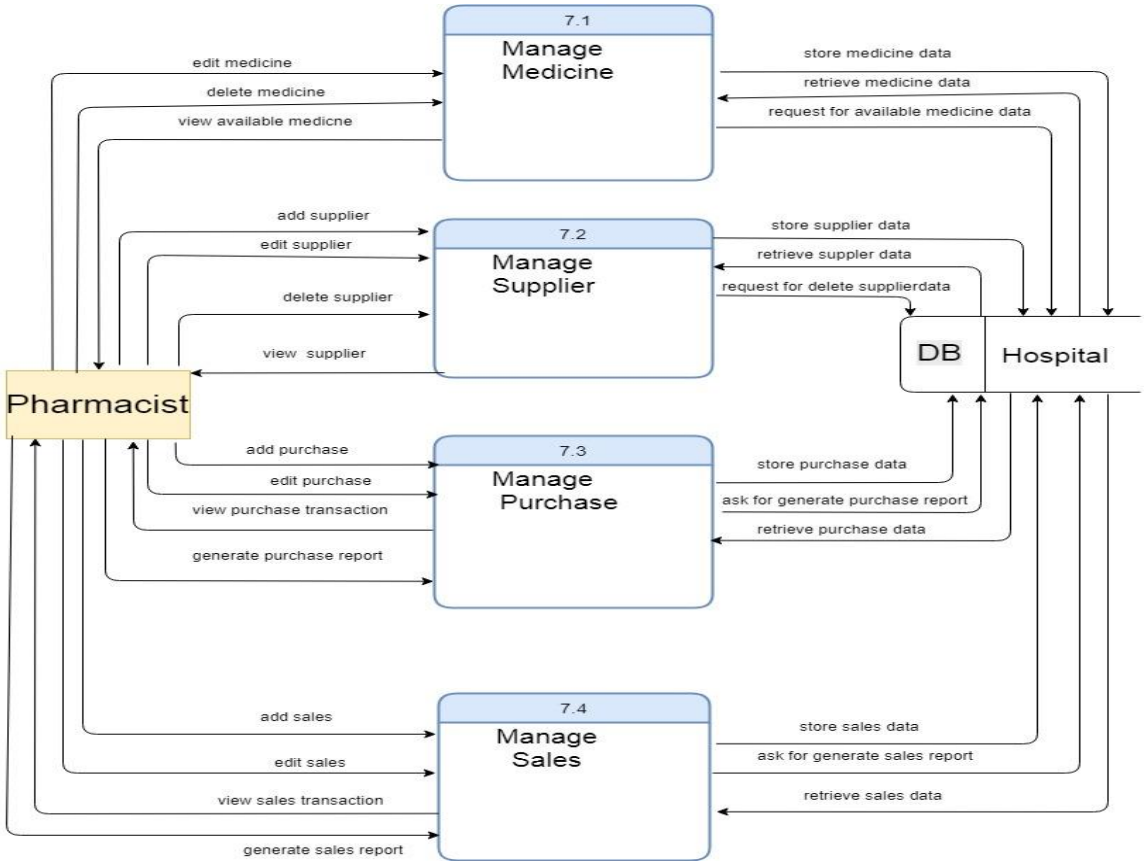
Level 2 diagram for process 5(Manage Patient):



Level 2 diagram for process 6(Manage Laboratory):



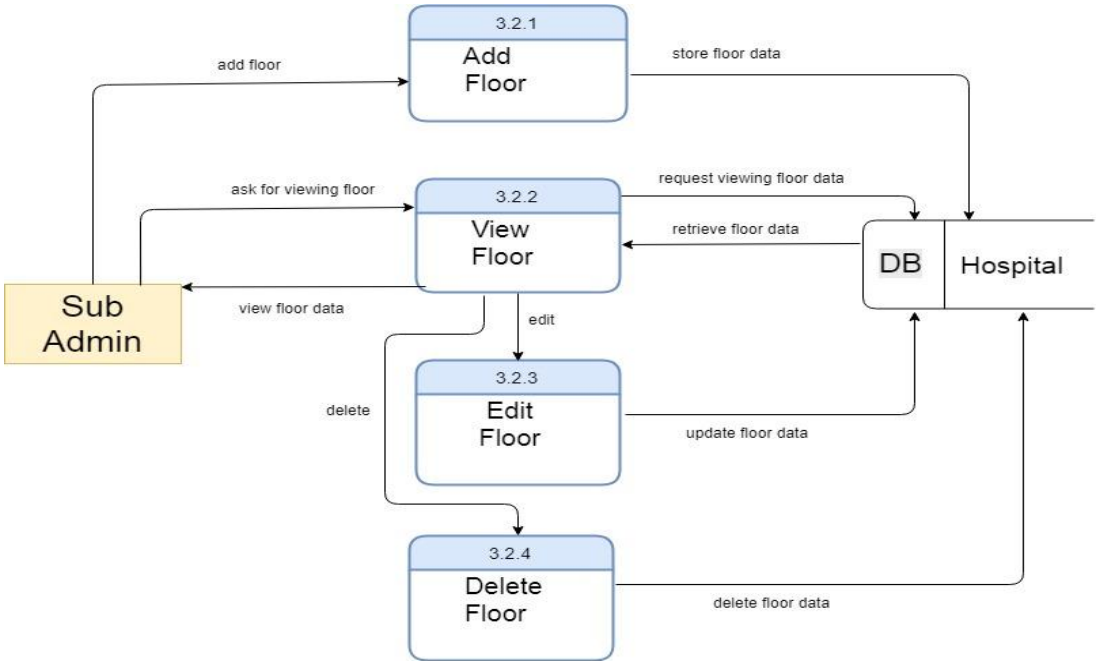
Level 2 diagram for process 7(Manage Pharmacy):



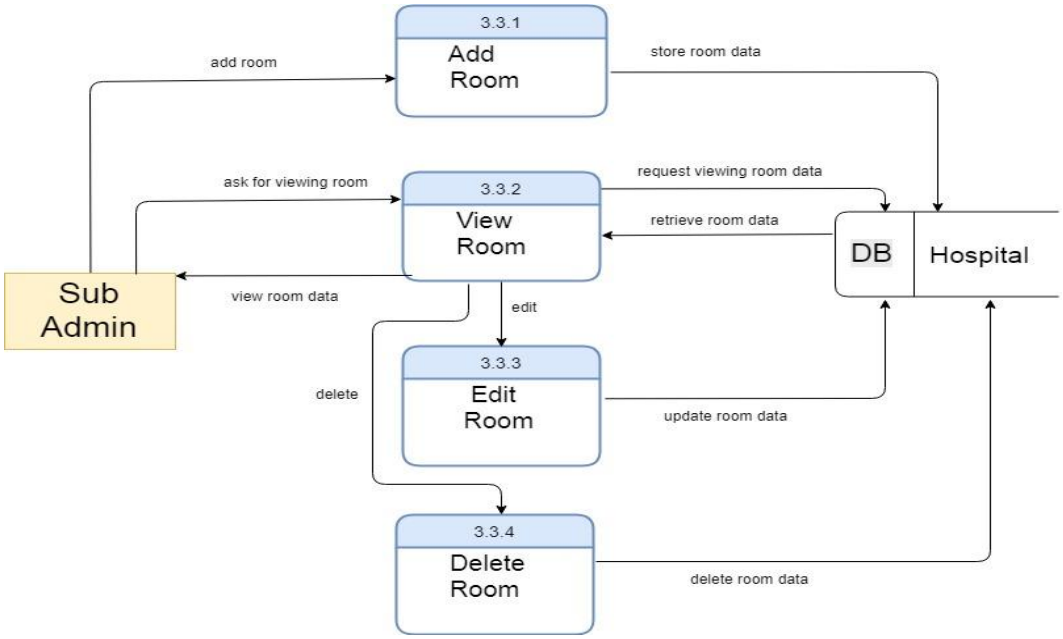
Level 3 diagram:

Level 3 diagram for process 3.1 (manage Building):

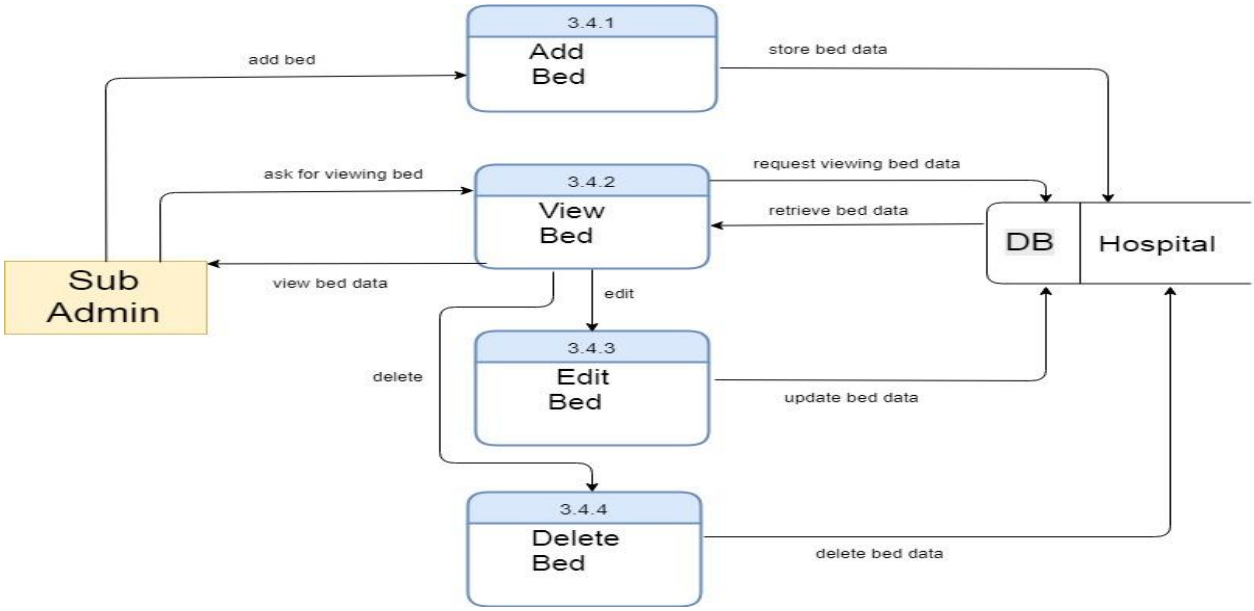
Level 3 diagram for process 3.2 (manage Floor):



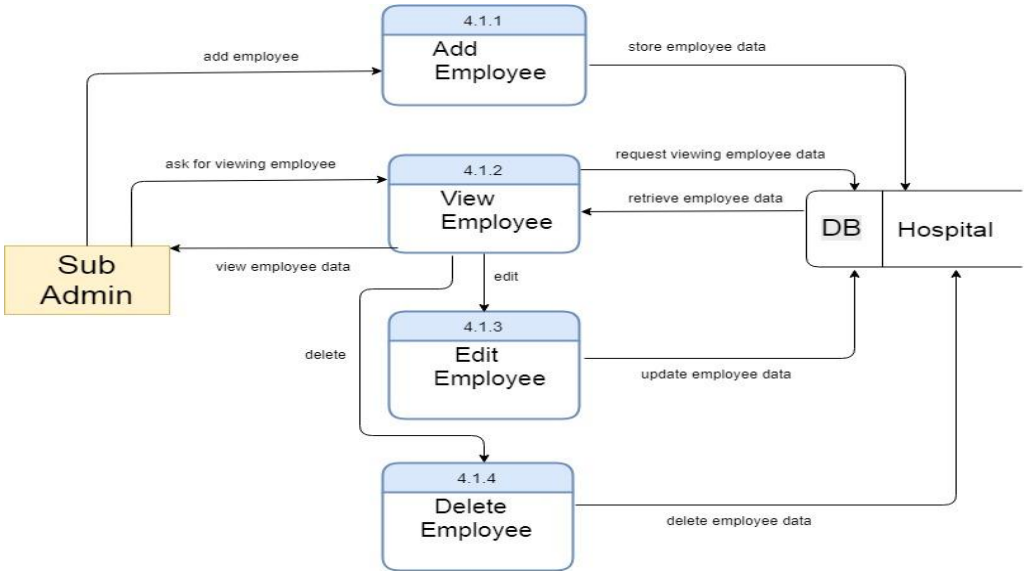
Level 3 diagram for process 3.3 (manage Room):



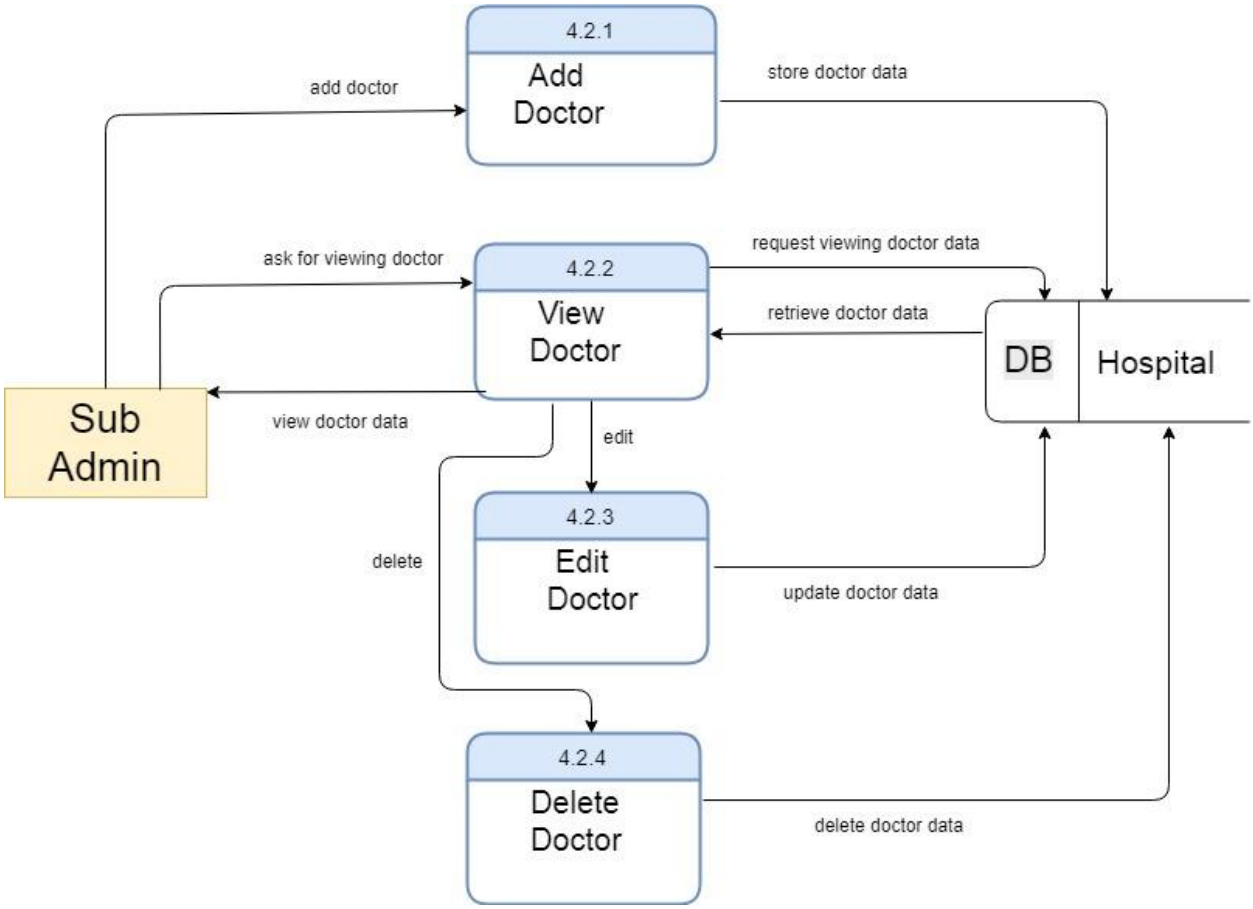
Level 3 diagram for process 3.4 (manage Bed):



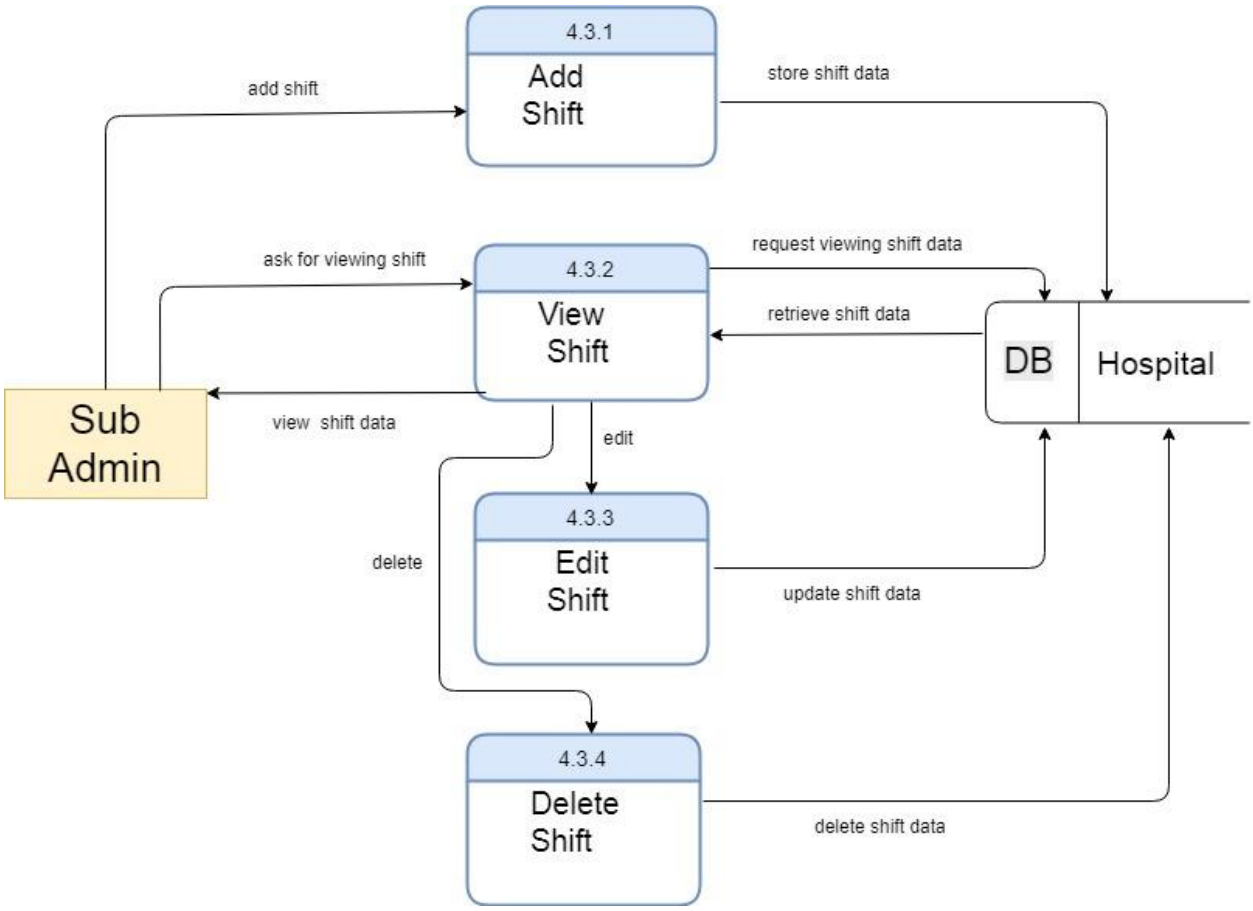
Level 3 diagram for process 4.1 (manage Employee):



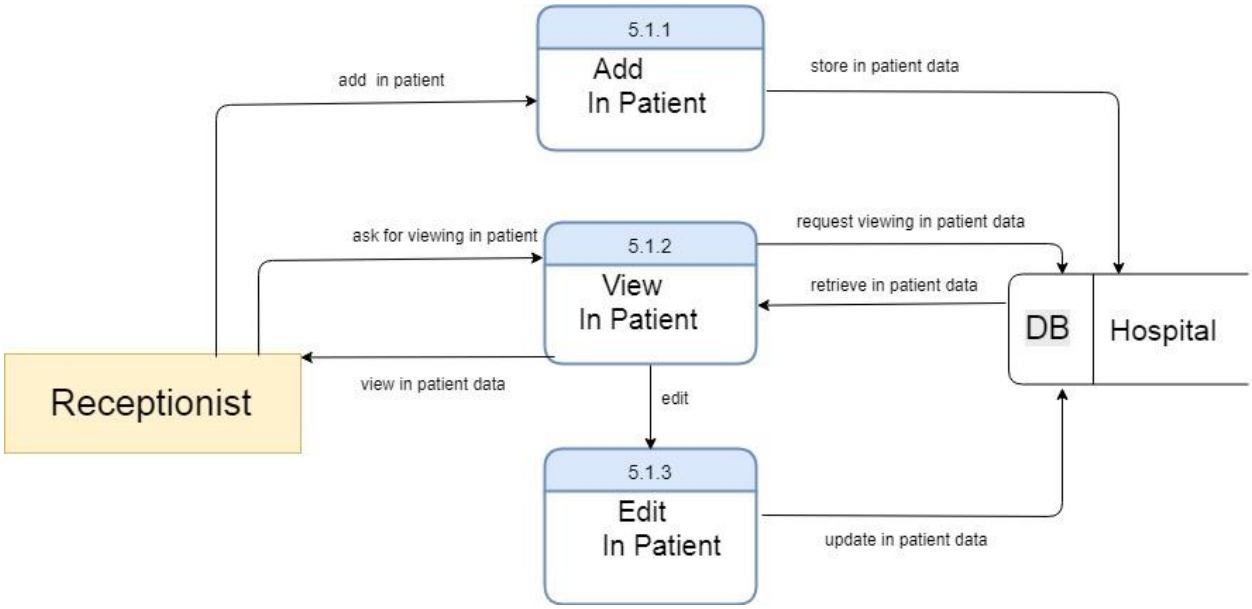
Level 3 diagram for process 4.2 (manage Doctor):



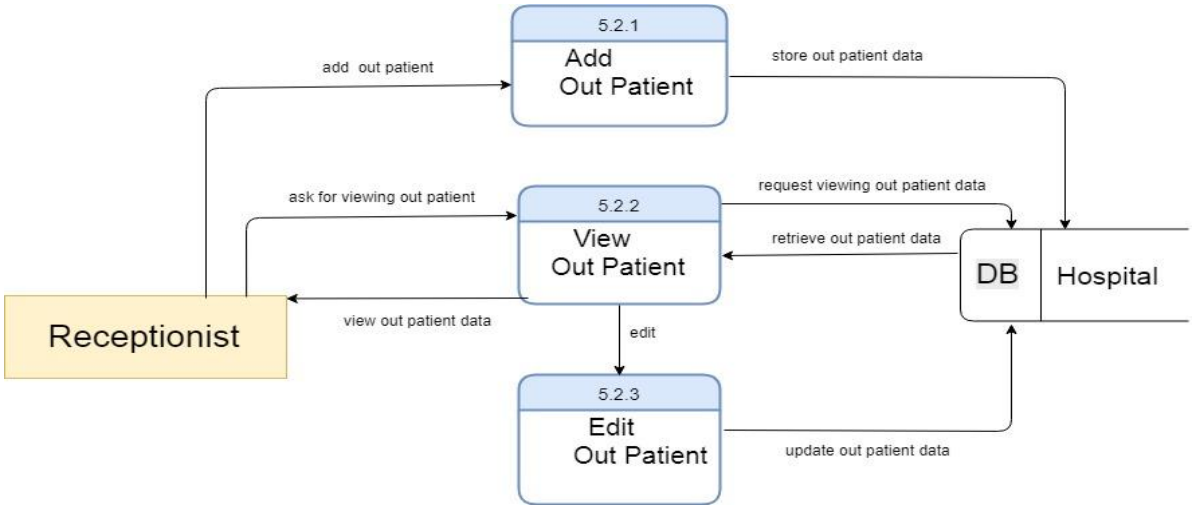
Level 3 diagram for process 4.3 (manage Shift)



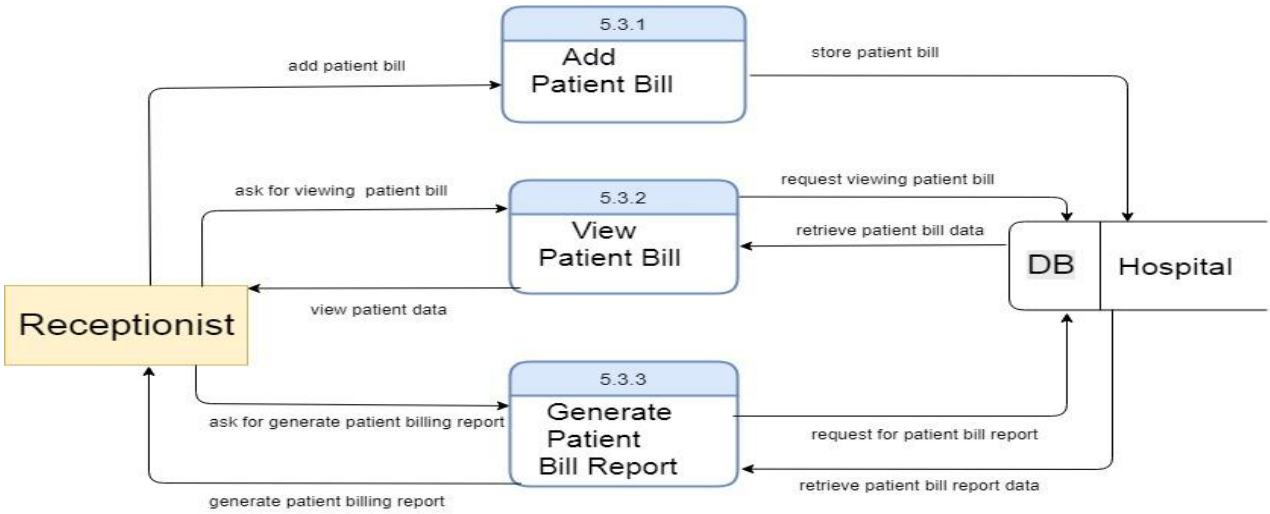
Level 3 diagram for process 5.1 (manage IPD):



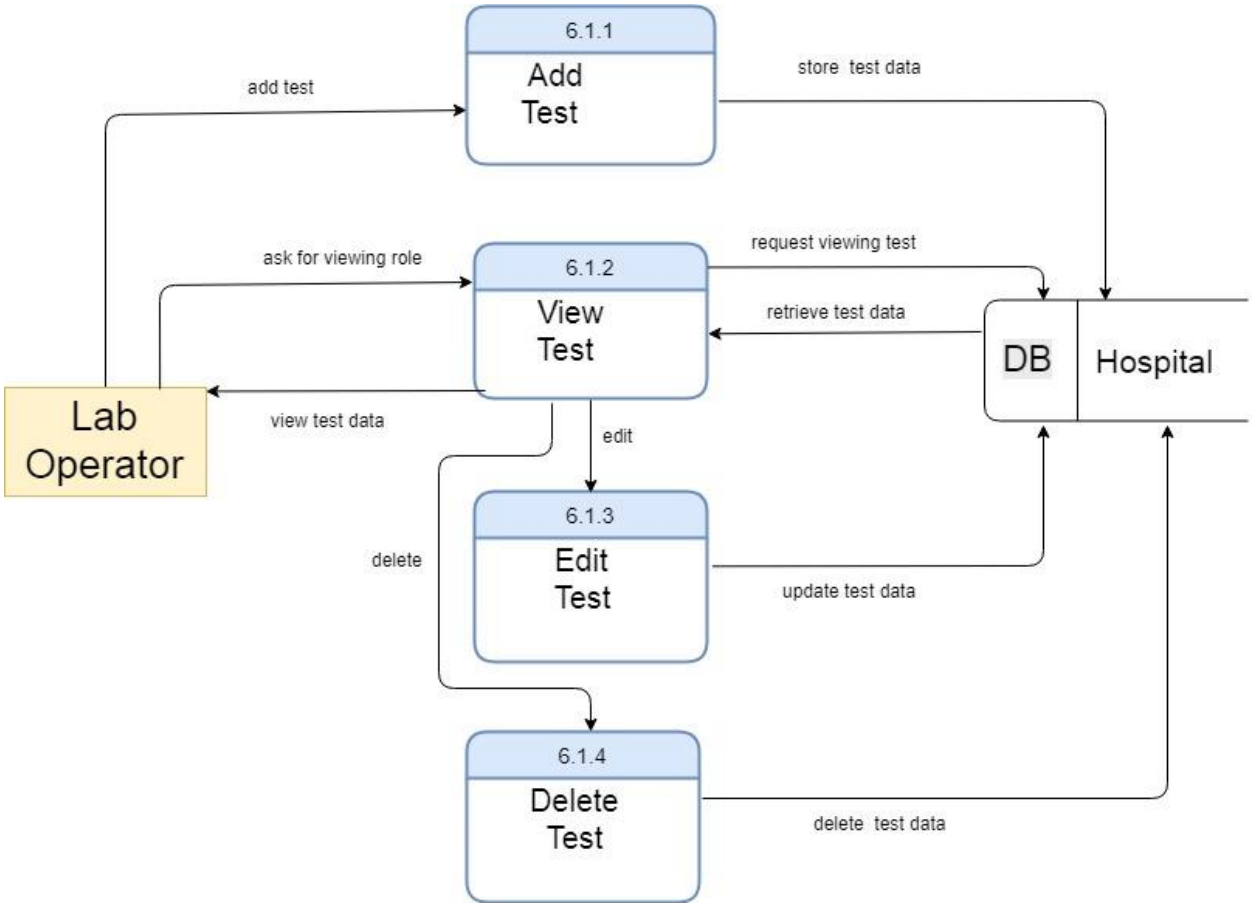
Level 3 diagram for process 5.2 (Manage OPD):



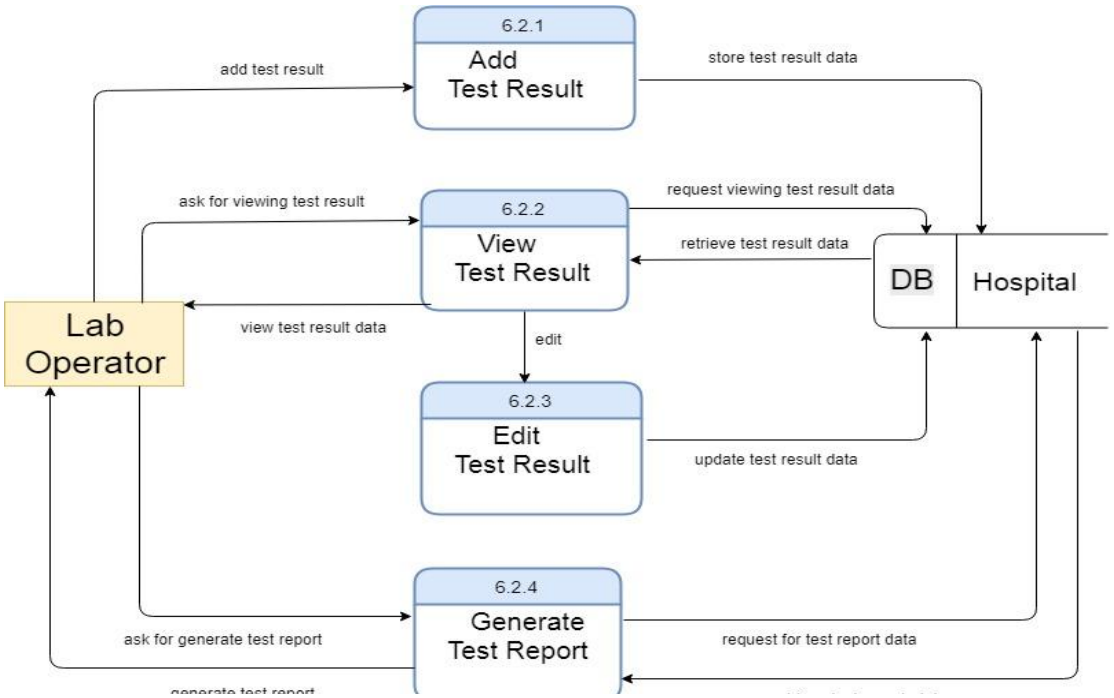
Level 3 diagram for process 5.3 (Manage Billing:



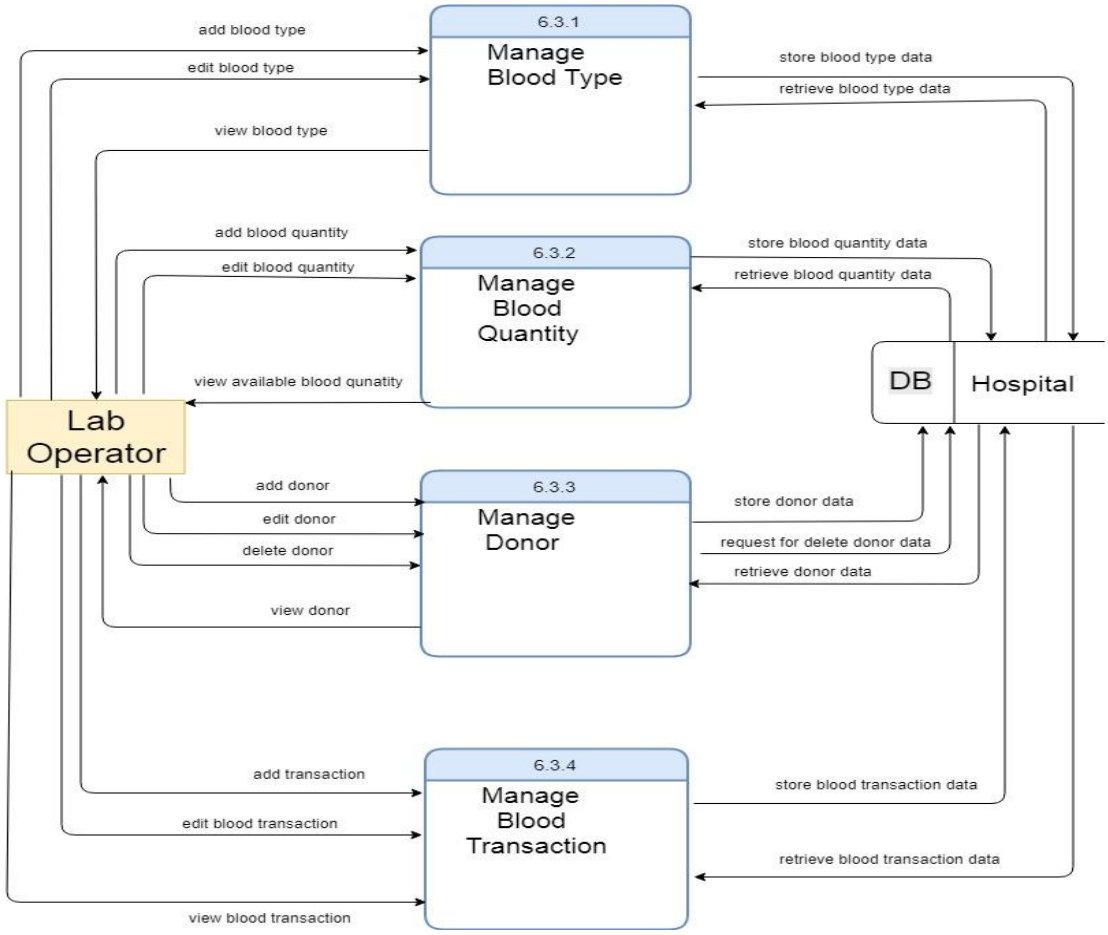
Level 3 diagram for process 6.1(Manage Test):



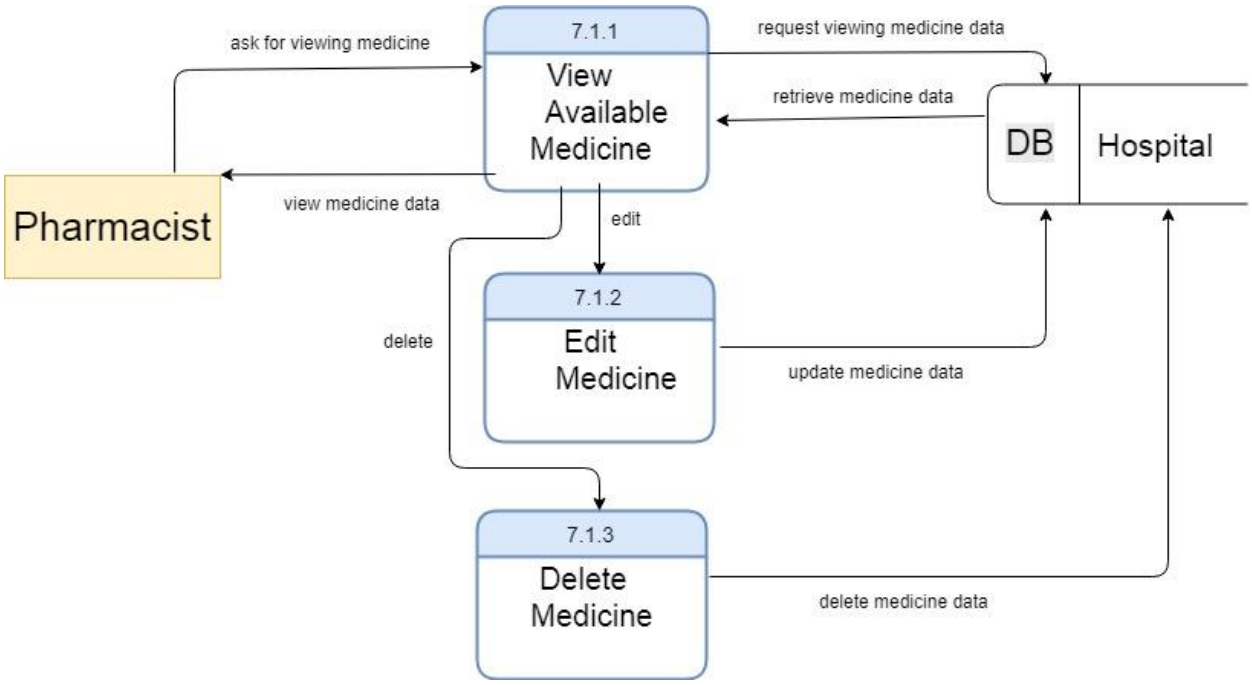
Level 3 diagram for process 6.2(Manage Test Result)



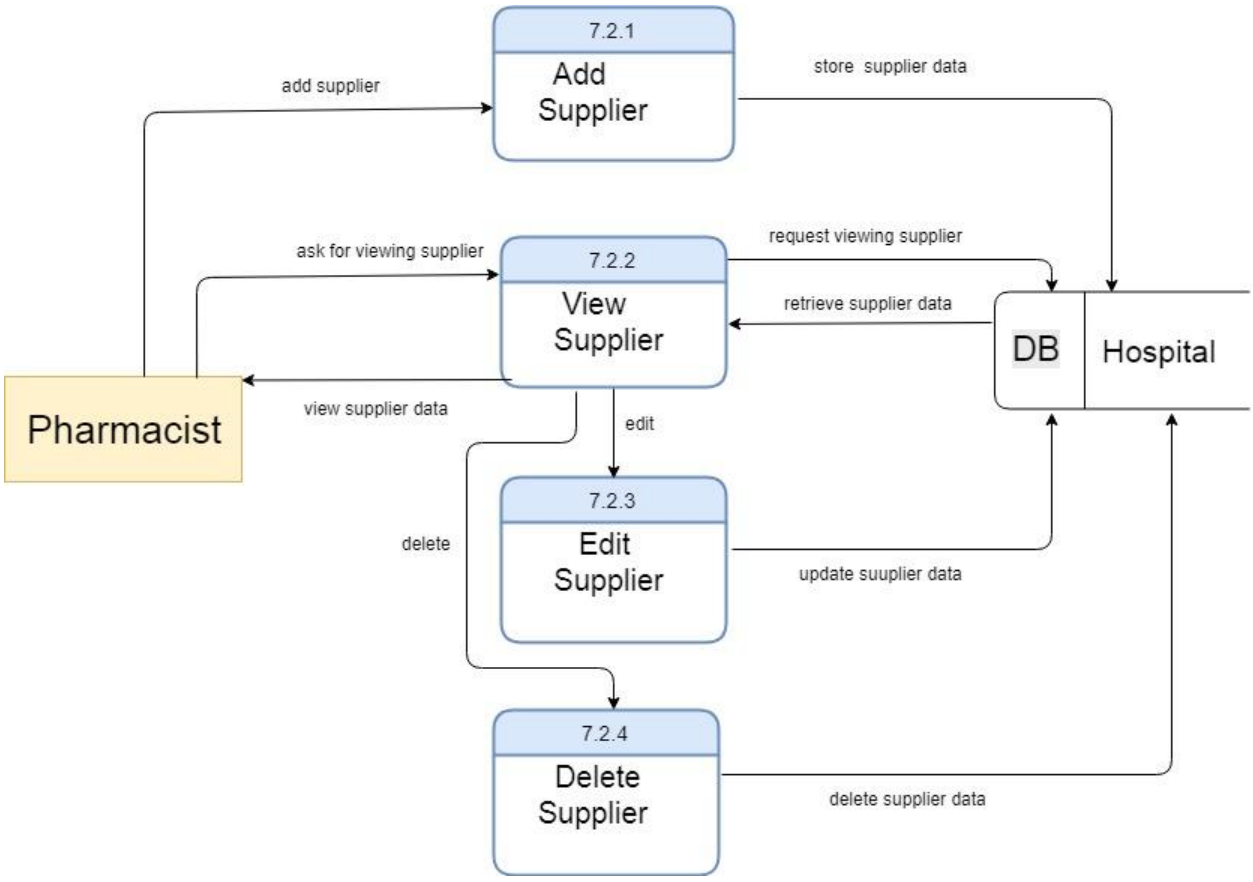
Level 3 diagram for process 6.3(Manage Blood Bank)



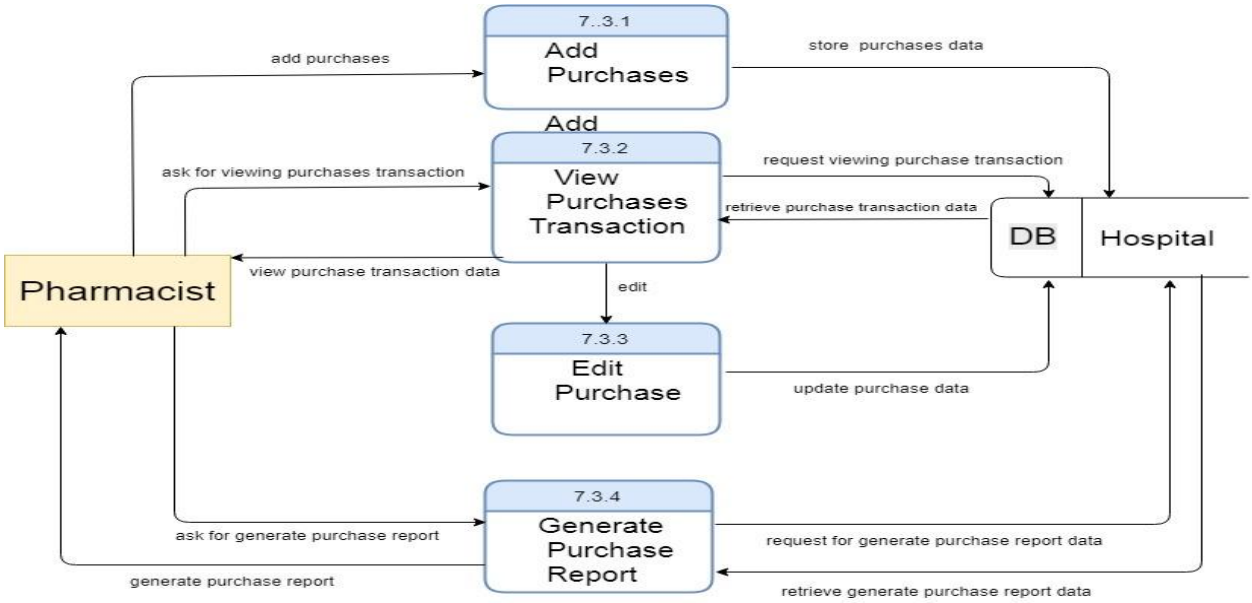
Level 3 diagram for process 7.1(Manage Medicine)



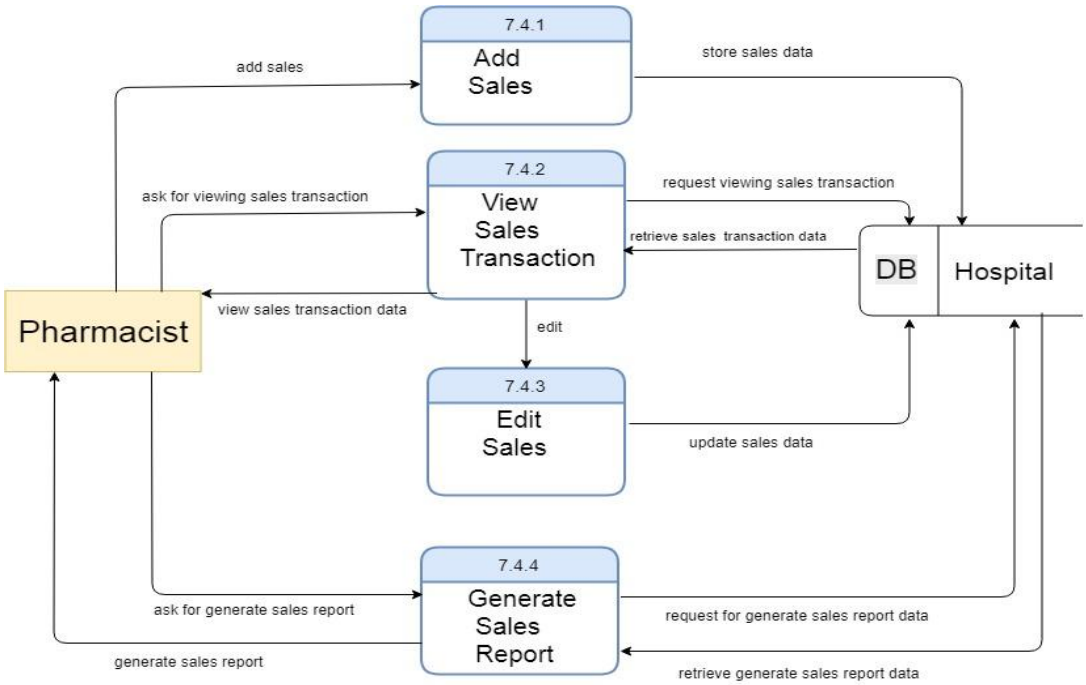
Level 3 diagram for process 7.2(Manage Supplier):



Level 3 diagram for process 7.3(Manage Purchase):



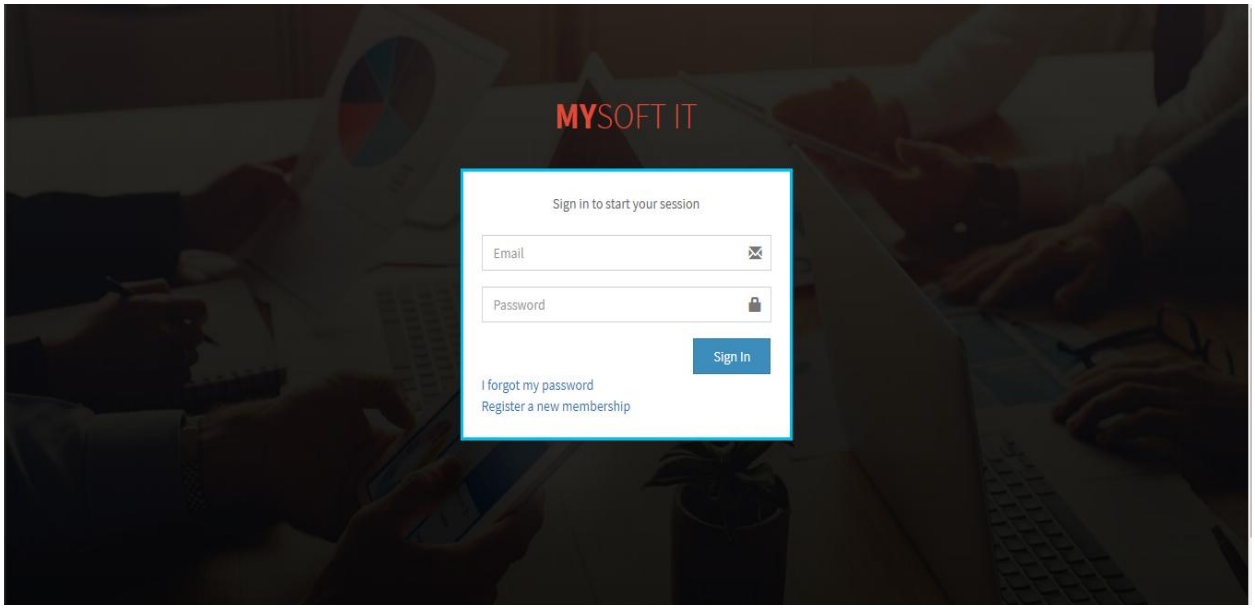
Level 3 diagram for process 7.4(Manage Sales):



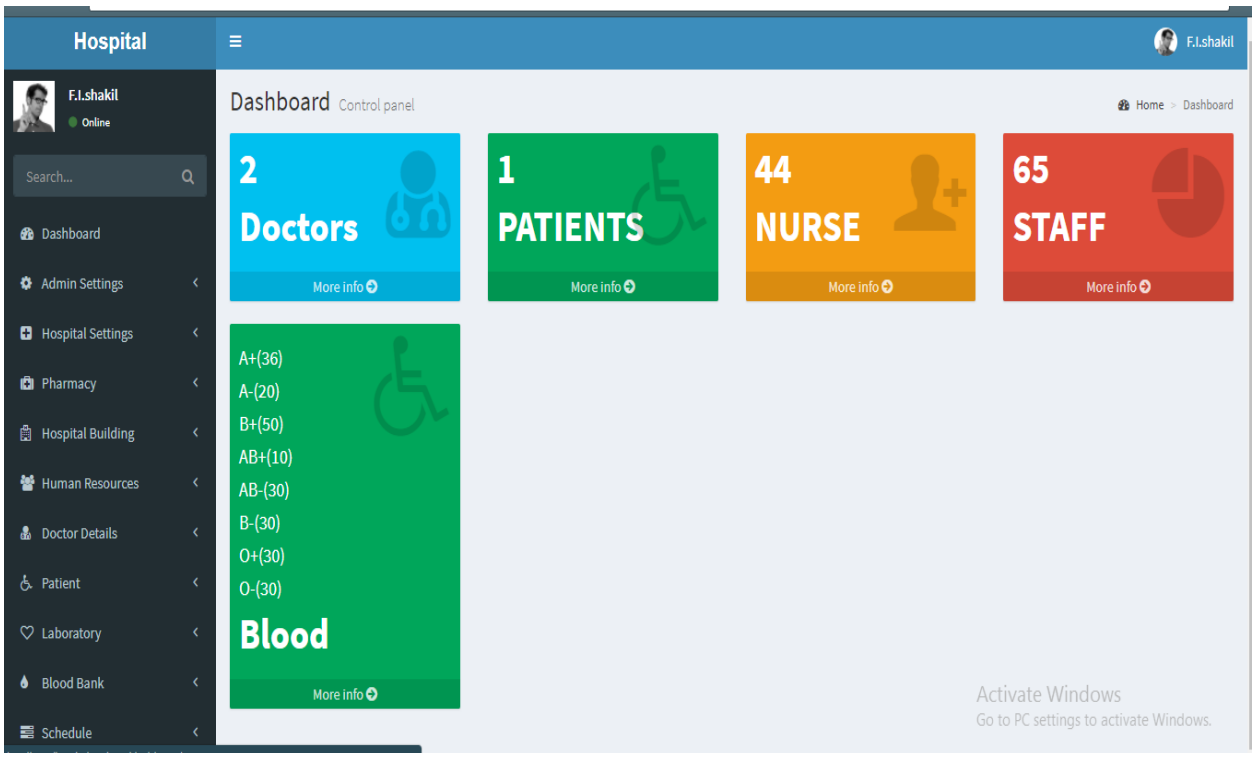
Text

6.5: Interface of the Project:

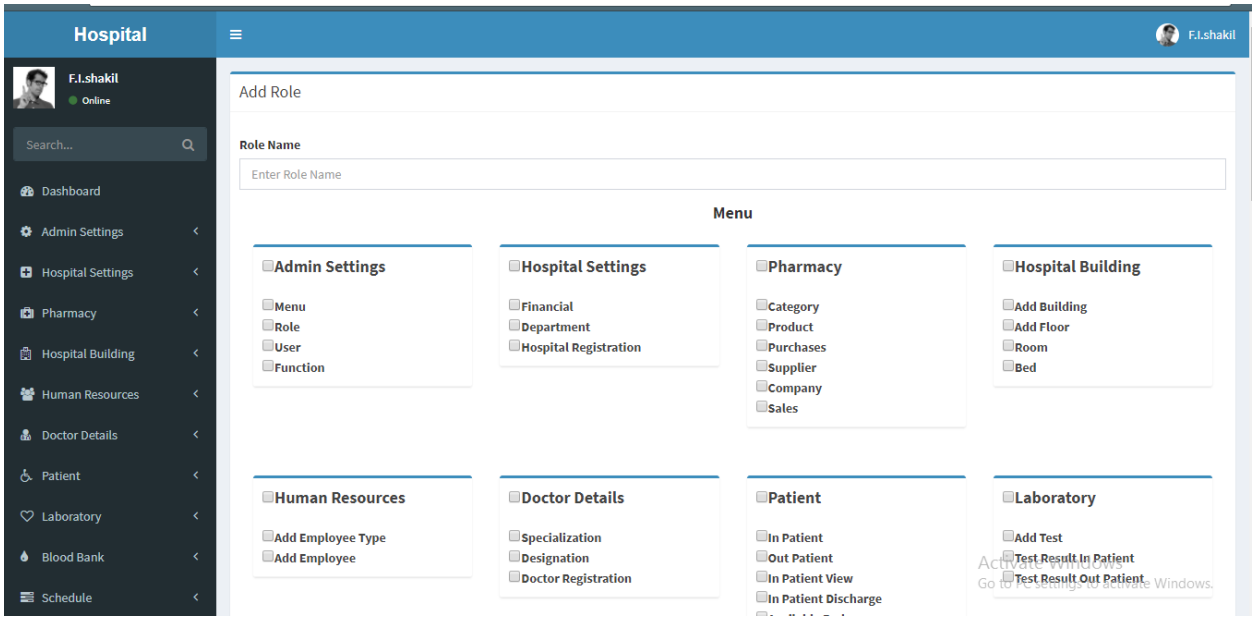
Login Page:



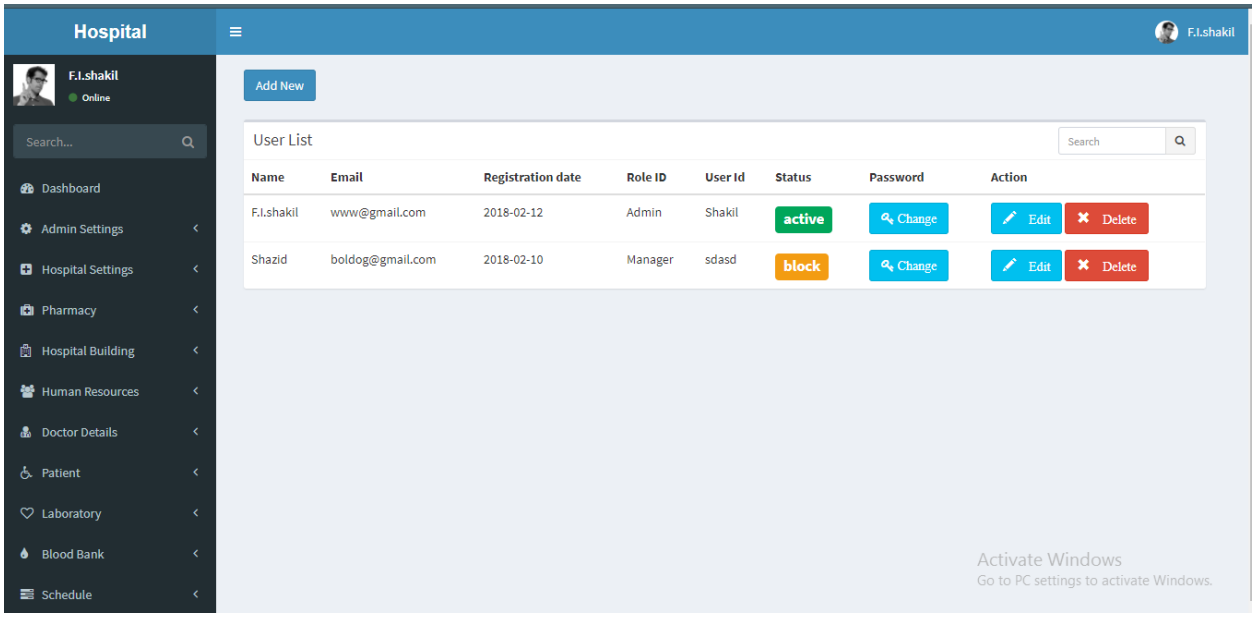
Dashboard:



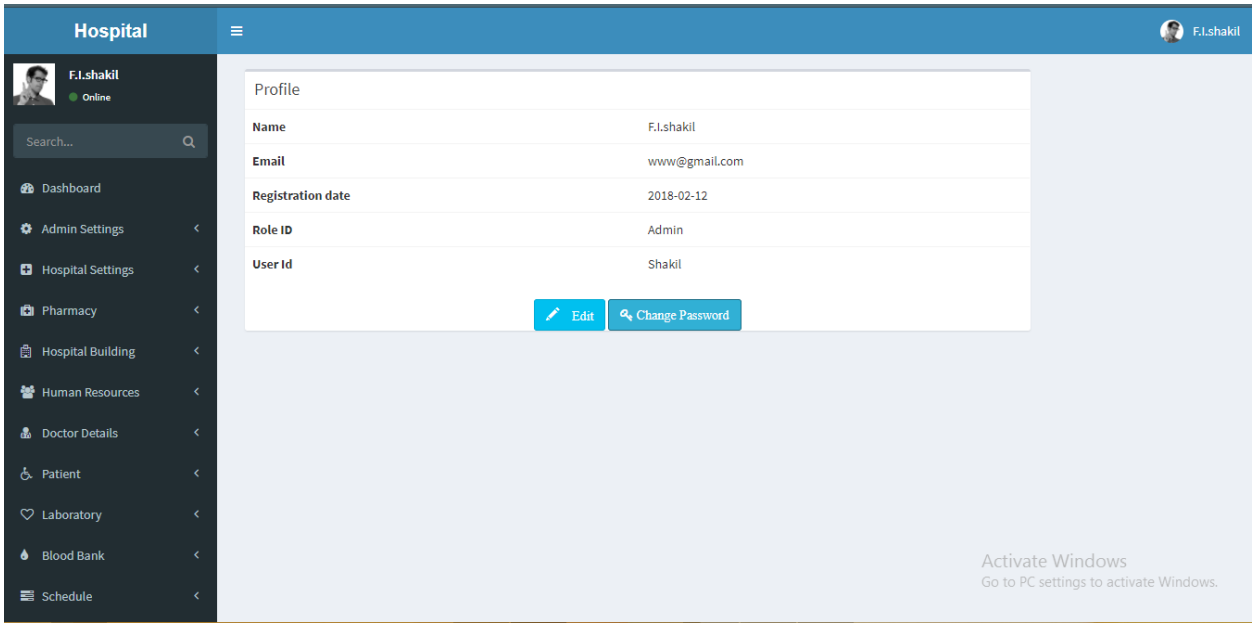
Role Access Page:



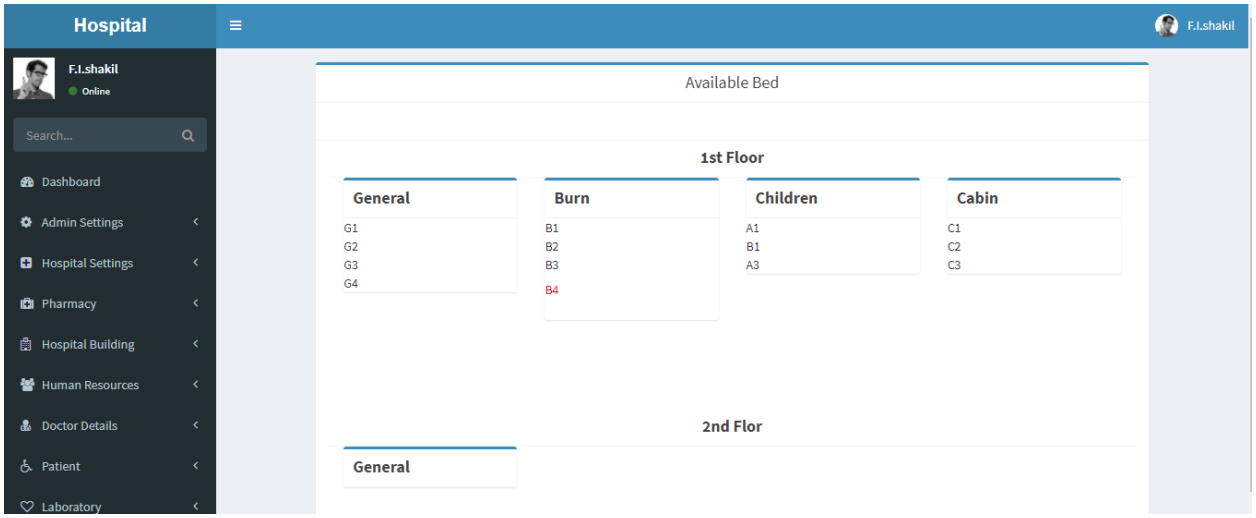
User View Page:



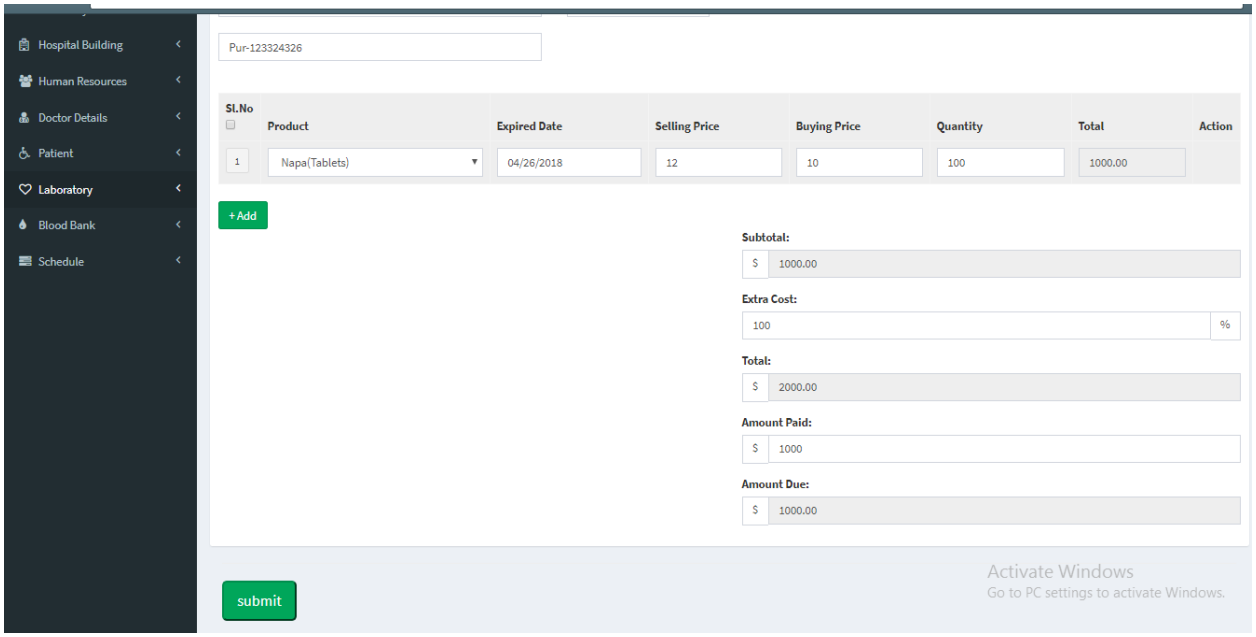
User Profile:



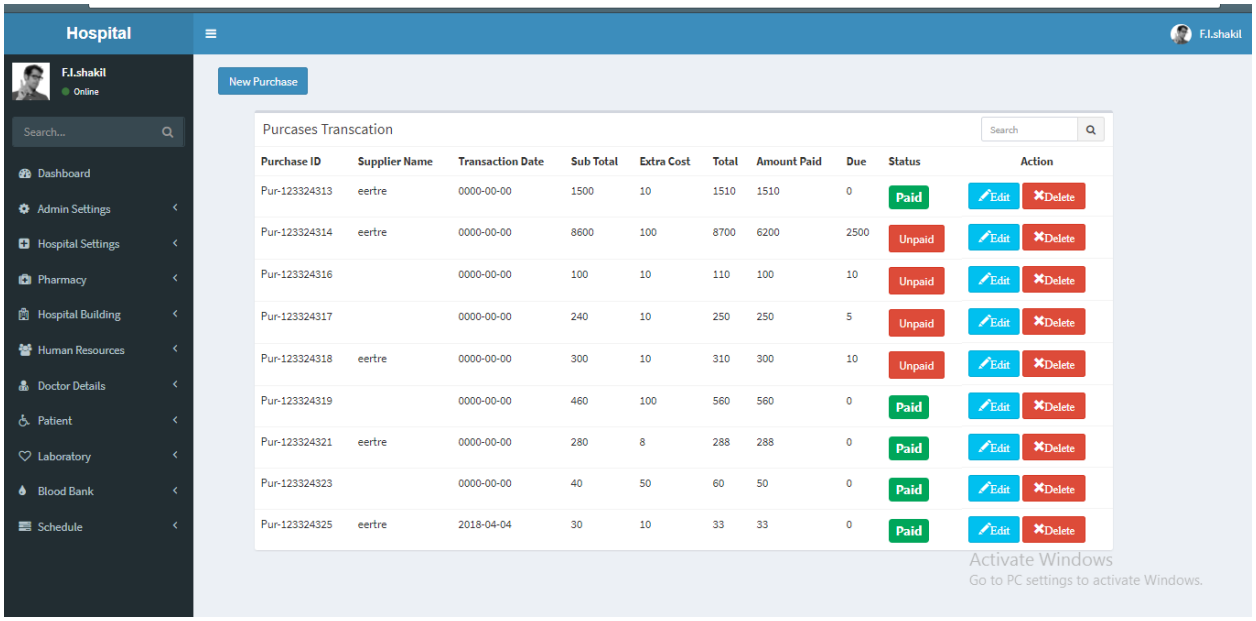
Available Bed:



Purchase Page:



Purchase Transaction List:



Available Blood Page:

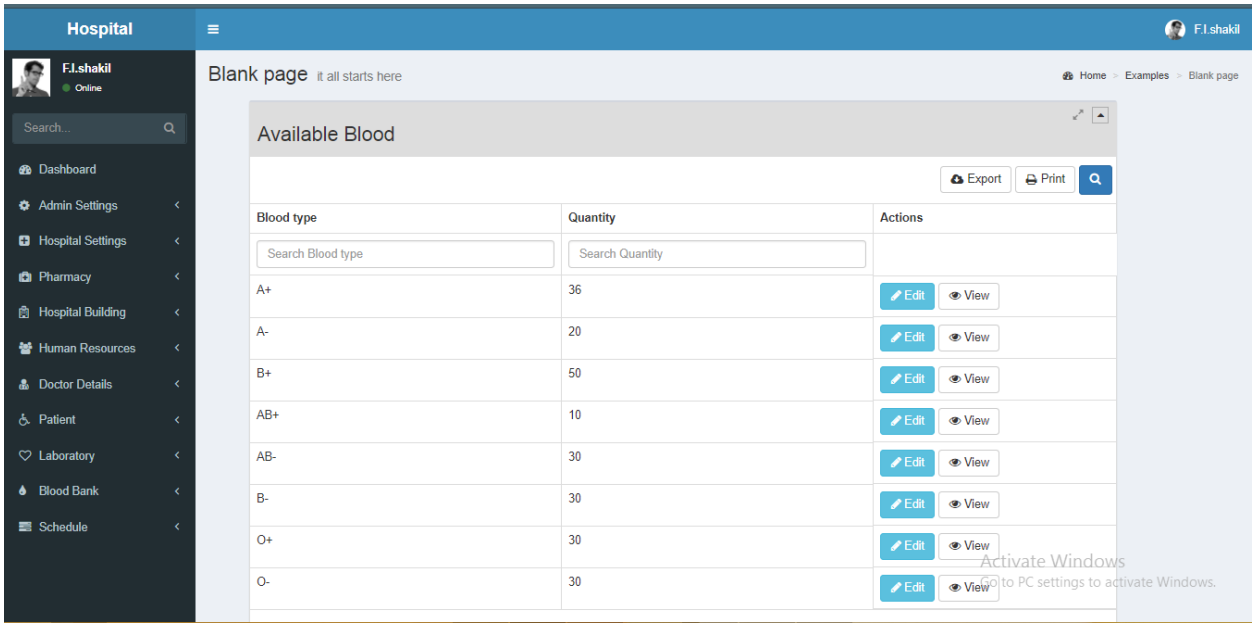


Figure 6. Interface of the Project

6.6: Database of the Project

User:

+ Options

	id	name	email	password	reg_date	role_id	user_id	status
Edit Copy Delete	17	F.I.shakil	www@gmail.com	202cb962ac59075b964b07152d234b70	2018-02-12	1	Shakil	active
Edit Copy Delete	18	Shazid	demo@gmail.com	202cb962ac59075b964b07152d234b70	2018-02-10	2	sdasd	block

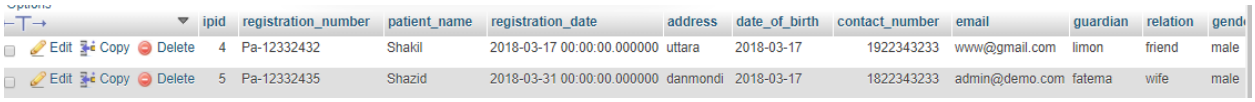
Role:

	role_id	role_name
Edit Copy Delete	1	Admin
Edit Copy Delete	2	Sub Admin

Bed:

	bed_id	bed_name	room_id	building_id	floor_id	status
Edit Copy Delete	1	B1	2	1	1	available
Edit Copy Delete	2	B2	2	1	1	available
Edit Copy Delete	3	B3	2	1	1	available
Edit Copy Delete	4	B4	2	1	1	booked
Edit Copy Delete	5	C1	4	1	1	available

Patient:



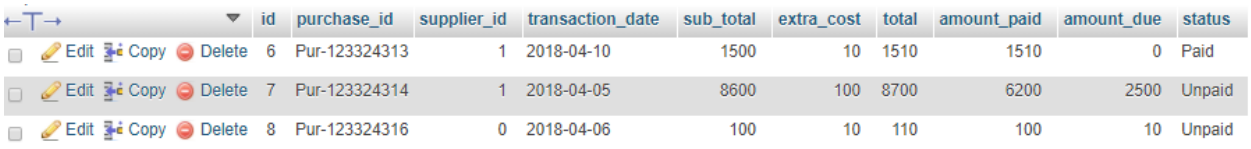
	ipid	registration_number	patient_name	registration_date	address	date_of_birth	contact_number	email	guardian	relation	gend
Edit Copy Delete	4	Pa-12332432	Shakil	2018-03-17 00:00:00.000000	uttara	2018-03-17	1922343233	www@gmail.com	limon	friend	male
Edit Copy Delete	5	Pa-12332435	Shazid	2018-03-31 00:00:00.000000	danmondi	2018-03-17	1822343233	admin@demo.com	fatema	wife	male

Medicine:

+ Options

	pid	company_id	name	catagory_id	price	quantity	expired_date	entry_user_id	entry_date	status
Edit Copy Delete	1	1	Napa	2	4	944	0000-00-00	dsdfdf	2018-10-10	active
Edit Copy Delete	3	2	Napa Extra	2	2	149	2018-03-31	www	2018-03-28	Active

Purchase Transaction



	id	purchase_id	supplier_id	transaction_date	sub_total	extra_cost	total	amount_paid	amount_due	status
Edit Copy Delete	6	Pur-123324313	1	2018-04-10	1500	10	1510	1510	0	Paid
Edit Copy Delete	7	Pur-123324314	1	2018-04-05	8600	100	8700	6200	2500	Unpaid
Edit Copy Delete	8	Pur-123324316	0	2018-04-06	100	10	110	100	10	Unpaid

Chapter-7

Quality Assurance & Testing

7.1 System Quality Management

A quality management software system that is automated and connects all departments is essential for a regulated or ISQ-compliant company. A QMS or a TQM (total quality management) system can connect each phase in a products development lifecycle with every department in a company. This gives everyone an opportunity to provide feedback. Automated, routing, with escalation, ensure the rapid responses to inputs needed from the department. By building quality into products as opposed to forcing QA to bear the burden of the responsibility, everyone wins, engineering, regulatory, QA, manufacturing, sales and marketing

The quality of software is assessed by a number of variables. These variables can be divided into external and internal quality criteria. External quality is what a user experiences when running the software in its operational mode. Internal quality refers to aspects that are code-dependent, and that are not visible to the end-user. External quality is critical to the user, while internal quality is meaningful to the developer only.

Some quality criteria are objective, and can be measured accordingly. Some quality criteria are subjective, and are therefore captured with more arbitrary measurement.

There are mainly two types of quality

- Internal quality:
- Test coverage
- Testability
- Portability
- Thread-safeness
- Conciseness
- Maintainability
- Documentation
- Legibility
- Scalability

- External quality
- Features
- Speed
- Space
- Network usage
- Stability
- Robustness
- Ease-of-use
- Determinism
- Back-compatibility
- Security
- Power consumption
-

7.1.1 Software Quality Management Process

- The aim of Software Quality Management (SQM) is to manage the quality of software and development and of its development process
- A quality product is one which meets its requirements and satisfies the user
- A quality culture is an organization environment where quality is viewed as everyone's responsibility

Some of the specific SQM processes defined in standard:

- **Quality assurance process**

Quality Assurance makes sure the project will be completed based on the previously agreed specifications, standards and functionality required without defects and possible problems. Its monitors and tries to improve the development process from the beginning of the project to ensure this.it is oriented to ‘’prevention’’.

- **Verification & validation process**

In software project management, software testing, and software engineering, verification and validation (V&V) is the process of checking that a software system meets specifications and that it full fill its intended purpose. It is normally the responsibility of software testers as part of the software development lifecycle.

In the verification, a client will either view the software, or see it implemented in a test situation. At this point it is imperative that the client who is needed of the software is able to ascertain that this software is hitting all the parameters initially requested or desired .only when this assurance is made should the next part of the verification and validation process be started. While this is not the last chance to "tweak" the software into doing the tasks required it is part of the last steps before a project is completed, and in being too quick to approve the software as this could cause problems later, and could also result in more money required for the software's later changes.

The next step of verifications &validation of software is simple. Client Company will approve the software and validate it as being what is required. This stage usually means a systematic checking off various requirements. While this might sound tedious, it is necessary part of the procedure to insure that again, the result is exactly to the specifications of all concerned. The entire verification and validation process is part of a normal sequence of quality control for software.

- **Review& Audit Process**

A software audit review, or software audit, is a type of software review in which one or more auditors who are not members of the software development organization conduct "An independent examination of a software product, software process, or set of software processes to assess compliance with specifications, standards, contractual agreements, or other criteria"

Five types of reviews or audits presented in the standard:

- **Management reviews**

The purpose of a management review is to monitor progress, determine the status of plans and schedules, confirm requirements and their system allocation, or evaluate the effectiveness of management approaches used to achieve fitness for purpose. This support decisions about changes and corrective actions that are required during a software project.

- **Technical reviews**

The purpose of technical review is to evaluate a software product to determine its suitability for its intended use. The objective is to identify discrepancies from approved specifications and standards. The result should provide management with evidence confirming (or not) that the product meets the specifications and adheres to standards and that changes are controlled". A technical review requires that mandatory inputs be in place in order to proceed:

- Statements of objectives
- A specific software product
- The specific project management plan

- **Inspections**

The purpose of an inspection is to detect and identify software product anomalies. Two important differentiators of inspections as opposed to reviews are as follows:

- An individual holding a management position over any member of the inspection team shall not participate in the inspection
- An inspection is too led by an impartial facilitator who is trained in inspection techniques.

The inspection exit must correspond to one of the following three criteria:

- Accept with no or at minor reworking
- Accept with rework verification
- Re inspect

Inspection meetings typically last a few hours, whereas technical reviews and audits are usually broader in scope and take longer.

- **Walk-through**

The purpose of a walk-through is to evaluate a software product. A walk-through may conduct for educating an audience regarding a software product. The major objectives are to:

- Find anomalies
- Improve the software product
- Consider alternative implementations
- Evaluate conformance to standards and specifications

The purpose of a software audit is to provide an independent evaluation of the conformance of software products and processes to applicable regulations, standards, guidelines, plans, and procedures. The audit is a formally organized activity, with participants having specific roles, such as lead auditor, another auditor, a recorder, or an initiator, and includes a representative of the audited organization. The audit will identify instances of nonconformance and produce a report requiring the team to take corrective action.

7.2 System Testing

Black box testing: You don't need to know the internal design in detail or have knowledge about the code for this test. It's mainly based on functionality and specification, requirements.

White box testing: This testing is based on detailed knowledge of the internal design and code. Tests are performed for specific code statements and coding styles.

Unit testing: The most micro scale of testing to test specific functions or code modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the

internal program design and code. Not always easily done unless the application has a well-designed architecture with tight code, may require developing test driver modules or test harnesses.

Incremental integration testing: Continuous testing of an application as new functionality is added. Requires that various aspects of applications functionality be independent enough to work separately before all parts of the programmer are completed or that test drivers be developed as needed. Done by programmers or by testers.

Integration testing: Testing of combined parts of an application to determine if they function together correctly. It can be any type of application which has several independent sub applications, modules.

Functional testing: Black box type testing to test the functional requirement of an application. Typically done by software testers but software programmers should also check if their code works before releasing it.

System testing: Black box type testing that is based on overall requirements specifications. Covers all combined parts of a system.

End to End testing: It's similar to system testing. Involves testing of a complete application environment similar to real world use. May require interacting with a database, using network communications, or interacting with other hardware, applications, or systems.

Sanity testing or smoke testing: An initial testing to determine if a new sw version is performing well enough to start for a major software testing. For example, if the new software is crashing frequently or corrupting database then it is not a good idea to start testing before all these problems are solved first.

Regression testing: Re-testing after software is updated to fix some problem. The challenge might to be determining what need to be tested, and all the interactions of the functions, especially near the end of the software cycle. Automated testing can be useful for this type of testing.

Accepting testing: This the final testing done based on the agreements with the customer.

Load / stress / performance testing: Testing an application under heavy loads. Such as simulating a heavy traffic condition in a voice or data network, or a web site to determine at what point the system start causing problems or fails.

Usability testing: Testing to determine how user friendly the application is. It depends on the user or customer. User interviews, surveys, video recording of user sessions, and other techniques can be used. Programmers and testers are usually not appropriate as usability testers.

Install / Uninstall testing: Testing of full, partial, or upgrade install / uninstall processes.

Recovery / failover testing: Testing to determine how well a system recovers from crashes, failures, or other major problems.

Security testing: Testing to determine how well the system protects itself against unauthorized internal or external access and intentional damage. May require sophisticated testing techniques.

Compatibility testing: Testing how well software performs in different environments. Particular hardware, software, operating system, network environment etc. Like testing a web site in different browsers and browsers and browsers versions.

Exploratory testing: Often taken to mean a creative, informal software test that is not based on formal test plans or test cases; testers may be learning the software as they test it.

Ad-hoc testing: Similar to exploratory testing, but often taken to mean that the testers have significant understanding of the software before testing it.

Context driven testing: Testing driven by an understanding of the environment, culture, and intended use of software. For example, the testing approach for life critical medical equipment software would be completely different than for a low cost computer game.

Comparison testing: Comparing software weakness and strengths to competing products.

Alpha testing: Testing of an application when development when development is nearing completion. Minor design changes may still be made as a result of such testing. Typically done by end users or others, not by programmers or testers.

Beta testing: Testing when development and testing are essentially completed and final bugs and problems need to be found before final release. Typically done by end users or others, not by programmers or testers.

Mutation testing: A method for determining if a set data or test case is useful, by deliberately introducing various code changes and retesting with the originals test data/ cases to determine if the defects are detected. Proper implementation requires large computation resources.

7.3. Testing

Table 7.1:Testing

Testing Scenario No:1	
Scenario	Login testing scenario of our system
Input's	Email and password of admin for login
Desired Output's	When enter email, password then get access level define.
Actual Output's	For login our system works properly.
Verdict	Getting result from desired outputs and actual outputs decided this system is successful for login.

Testing Scenario No:2	
Scenario	employee info insert testing scenario of our system
Input's	Sub_admin insert employee information
Desired Output's	Employee info will show for start or stop job
Actual Output's	We check this process and get actual outputs
Verdict	Our system is worked correctly and successfully.

Testing Scenario No:3	
Scenario	Subadmin insert bed information testing scenario of our system
Input's	Subadmin insert bed info for patient
Desired Output's	Give success message and store data
Actual Output's	We check this process and get actual outputs
Verdict	Our system is worked correctly and successfully.

Testing Scenario No:4	
Scenario	Lab operator add test testing scenario of our system
Input's	laboperator insert new information about test
Desired Output's	Test info will show for start or stop job
Actual Output's	Our desired output access to actual and successfully practical output. So, this is successfull
Verdict	Our system is worked correctly and successfully.

Testing Scenario No:5	
Scenario	View available blood info testing scenario of our system
Input's	Lab operator can view available blood info
Desired Output's	View the blood list
Actual Output's	Our desired output access to actual and practical output. So, this is successful.
Verdict	Our system is worked correctly and successfully.

Testing Scenario No:6	
Scenario	View patient billing info testing scenario of our system
Input's	Receiptionist can view patient billing info
Desired Output's	View the patient billing list
Actual Output's	Our desired output access to actual and practical output. So, this is successful.

Testing Scenario No:7	
Scenario	Subadmin insert doctor information testing scenario of our system
Input's	Subadmin insert doctor info for patient
Desired Output's	Give success message and store data
Actual Output's	We check this process and get actual outputs
Verdict	Our system is worked correctly and successfully.

Testing Scenario No:8	
Scenario	Subadmin insert patient information testing scenario of our system
Input's	Subadmin insert patient info for patient
Desired Output's	Give success message and store data
Actual Output's	We check this process and get actual outputs
Verdict	Our system is worked correctly and successfully.

Testing Scenario No:9	
Scenario	Subadmin edit patient information testing scenario of our system
Input's	Update patient information
Desired Output's	Give update information
Actual Output's	We check this process and get actual outputs
Verdict	Our system is worked correctly and successfully.

Chapter-8

Conclusion

8.1.Future Plan

- Add Account management system.
- Add Payroll System
- Add Schedule management System.

8.2.Limitation

- There is no account system.
- So many data input.
- Manual shifting system.

8.3.Conclusion

Now a days, Trade and commerce have become depended on online system. Many changed lifestyle has made online system services as a necessity. My project is a fundamental approach of these.

The four years undergraduate engineering studies gives a student many theoretical comprehensions and using that knowledge and observing live operational system, the practicum program clarifies those subject's matters to another level; blessed with practical working skills. Considering this fact, it gives me an immense pleasure to say that my practicum was a successful event.

During the internship period I have tried my level best to make my system efficient. I followed the lessons, methods, tools and techniques that I have learned during my study period at IUBAT. Successful software development is a blend of standard development practices, proper theoretical knowledge and developer's creativity

Chapter-9

References

Books and Handouts

- ✓ Kendall, E. & Kendall.(1999). *System Analysis and Design*. 4th Ed. New Delhi: Prentice Hall.
- ✓ Pressman, Roger S.(2004),*Software Engineering: A Practitioner's Approach*. 5th ed. Boston: McGraw Hill.

- ✓ Silberschattz, Abraham, Korth, Henry F., &Sudrashan S. (2002). *Database System Concepts*. 4th ed. Boston: McGraw Hill.
- ✓ S. Pressman, Roger (2008). *Software Engineering A Practitioner's Approach*. 6th edition.

✓Pitt, C. & Smith, J. (2012). *Pro PHP MVC*. Berkeley, CA: Apress, pp.3-4.

Websites

- ✓ Meyer, Bertrand. (1997), OOSC2: The Use Case Principle. [Online] Available at: http://www.elj.com/elj/v1/n2/bm/usecases/ (10th November, 2017).

✓Longstreet, David. (2005). Fundamentals of Function Point Analysis. [Online] Available at: http://www.softwaremetrics.com/fpafund.htm (05th December, 2017).

✓Tibro B. (2005). http://www.wikipidia.com(Jul 19 '14).