SUPERIOR UNIVERSITY LAHORE

Faculty of Computer Science & IT

Final Year Project

PROJECT REPORT (Part-1)

Hospital Management System

Project ID: **FYP-BSCS-F18-002**

Project Team

	STUDENT CONTACT DETAILS				
#	**Student ID**	**Student Name**	**Program**	**Contact No.**	**Email**
1	Bcsm-f15-317	**Amna Khalil**	BCS	03070110556	Amnakhalil07@gmail.com
2	Bcsm-f15-316	**Hafiz Muazam Ali Khan**	BCS	03056130521	Muazamkhan10@gmail.com
3	Bcsm-f15-330	**Sajjad Hussain**	BCS	03126147988	syedsajjad9718@gmail.com

Sir Ifraseab

Senior Lecturer

Project Report

Hospital Management System

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
	1.0		<Original Draft>	
			<Changes Based on Feedback from Supervisor>	
			<Changes Based on Feedback From Faculty>	
			<Added Project Plan>	
			<Changes Based on Feedback from Supervisor>	

APPROVAL

PROJECT SUPERVISOR

Comments: _____

_____

Name:_____

Date:_____ Signature:_____

PROJECT MANAGER

Comments: _____

_____

Date:_____ Signature:_____

HEAD OF THE DEPARTMENT

Comments: _____

_____

Date:_____ Signature:_____

Dedication

This work is dedicated to my family, friends and everyone who showered me with their love and support. I am especially thankful to those who presented challenges to my work and I pray for them because without their resistance I would never have been able to achieve what I did.

Acknowledgments

I am really thankful to my supervisor who has guided us well throughout the semester. As well as the people that provided us with their honest opinions and reviews. Who made time for us for conducting their interview from their busy schedules. I would also like to thank my friends and family that provided us with the opportunity and support for developing this project.

Executive Summary

Hospital Management System provides the benefits of streamlined operations, enhanced administration & control, superior patient care, strict cost control and improved profitability. The project 'Hospital Management System' is based on the database, PHP. As there are many areas where we keep the records in database for which we are using MY SQL software which is one of the best and the easiest software to keep our information. This project uses JAVA script as the front-end software which is an PHP and has connectivity with MY SQL.Hospital Management System is custom built to meet the specific requirement of the mid and large size hospitals across the globe. All the required modules and features have been particularly built to just fit in to your requirement. This package has been widely accepted by the clients in India and overseas. Not stopping only to this but they are highly satisfied and appreciating. Entire application is web based and built on 3 tier architecture using the latest technologies. The sound database of the application makes it more users friendly and expandable. The package is highly customize and can be modified as per the needs and requirements of our clients. Prolonged study of the functionalists of the hospital and its specific requirement has given it a wonderful shape both technically and usability wise. It covers all the required modules right from Patient Registration, Medicine details, Doctor, Wards, , Admin, Store, Patient appointment, bill payment, record modification, discharge details etc.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chapter 1: Introduction

This document describe the requirement specification for the software Hospital management system(HMS). HMS will mainly compose of five modules: Doctor, Patient, Pharmacy, Receptionist. Hospital Management system includes registration of patients, storing their details into the system, and also computerized billing for each patient and Room details. The main purpose of our system is to make hospital task easy and is to develop software that replaces the manual hospital system into automated hospital management system.

1.1. Background

Due to shortage, maintenance and owned hospital still use file systems to store their day to day activities and information; thereby putting at risk the loss of information should any of the files get missing. The company therefore required me to design software that is able to record all daily activities of their medical center. The medical center therefore suggest the name, hospital management software as the name of their software, therefore the implementation of Hospital Management Software will be called HMS from this point onward throughout this project.

1.2. Motivations and Challenges

HMS refers to the degree to which the information system service is perceived as greatly helpful to facilitate and improve the work efficiency and work speed. It refers to a hospital's capabilities of providing information system services that correctly delivery the service requested by users.

1.3. Goals and Objectives

- ✓ The computerized detail regarding patient detail and hospital detail.
- ✓ The automate the process of ward entities.
- ✓ The maintain records effectively.

- ✓ The manage current status of staff and doctor availability.
- ✓ Objectives Hospital Management Service. It is the user friendly application for Hospital which reduces the burden and helps to manage all sections of hospital like reception, inpatient/outpatient management and Billing etc., which improve the processing efficiency

1.4. Existing Solutions

The current systems require numerous paper forms, with data stores spreads throughout the hospital management infrastructure. Often information is incomplete, or does not follow management standard. Forms are often lost in transit between departments requiring a comprehensive auditing process to ensure that no vital information is lost. Multiple copies of the same information exist in the hospital and may lead to inconsistencies in data in various data stores.

1.5. Gap Analysis

According to our perception, gap analysis is the gap between the old hospital management system and the new hospital management system.

In old HMS you can only set appointment and check bill information. In our HMS you can also check your details online , the staff member easily check the assessment ,medicine detail ,ward details , Doctor details and you can easily consult a doctor through message.

1.6. Proposed Solution

- ✓ All these above problems we can resolve as we make record update, delete, add of patient, doctors and other staffs so we make an online system for patient for easily get appointment date from hospital.
- ✓ System will give information about patients regular treatment and admit in hospital.
- ✓ A discipline involves records keeping. Record keeping is an important aspect of every institution's day to day operations.
- ✓ The system will be used to assign doctors to patients in designated wards as need arises.
- ✓ Doctors will also use the system to keep track of the patients assigned to them.

- ✓ Receptionist will also use the system to keep track assigned to them.
- ✓ Administrator will also use the system to all data assigned to them.

1.7. Project Plan

1.7.1 Work Breakdown Structure

1. **Project Planning**
   - 1.1. Problem Statement
   - 1.2. Existing Systems review
   - 1.3. Scope
2. **Software requirement & analysis**
   - 2.1. Final Documentation Introduction
   - 2.2. Requirement Elicitation
   - 2.3. Requirements Analysis
3. **Design**
   - 3.1. Architecture design
   - 3.2. UML design
   - 3.3. Database design
   - 3.4. User Interface design
   - 3.5. Review all design
4. **Implementation**
   - 4.1. Coding & Debugging
   - 4.2. Change Management
5. **Testing**
   - 5.1. Unit Testing
   - 5.2. Integration Testing
   - 5.3. System Testing
6. **Deployment**
   - 6.1. System deployed

6.2. Presentation

7. System

7.1. Development Environment

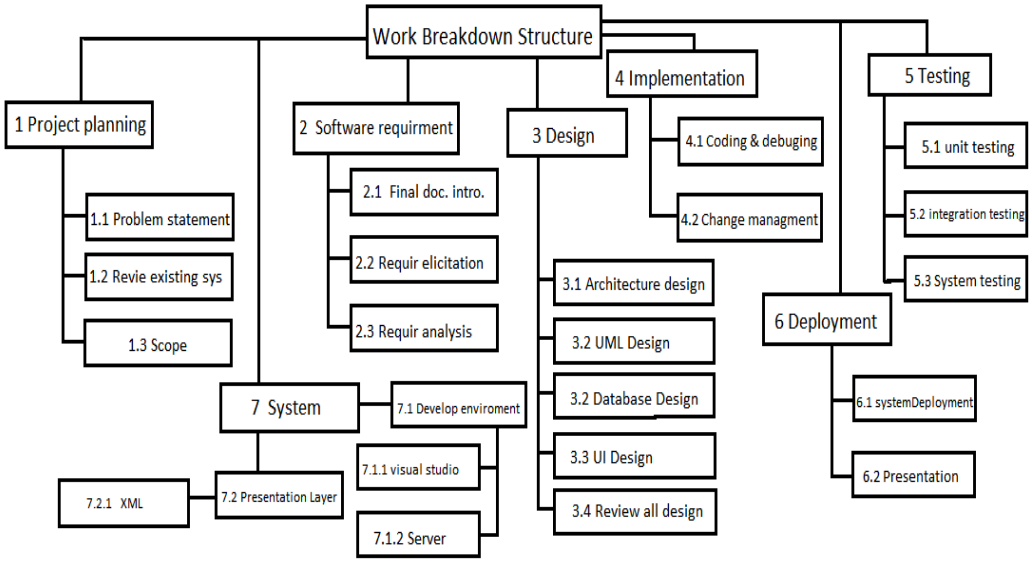
7.1.1. Visual Studio Code and Android Studio

7.1.2. Server

7.2. Presentation Layer

7.2.1. XML

1.7.1. Work Breakdown Structure



1.7.2. **Roles & Responsibility Matrix**

WBS #	WBS Deliverable	Activity #	Activity to Complete the Deliverable	Duration (# of Days)	Responsible Team Member(s) & Role(s)
1	Project Planing	1	Requirements problem statement,	10	Amna Khalil Muazam Ali

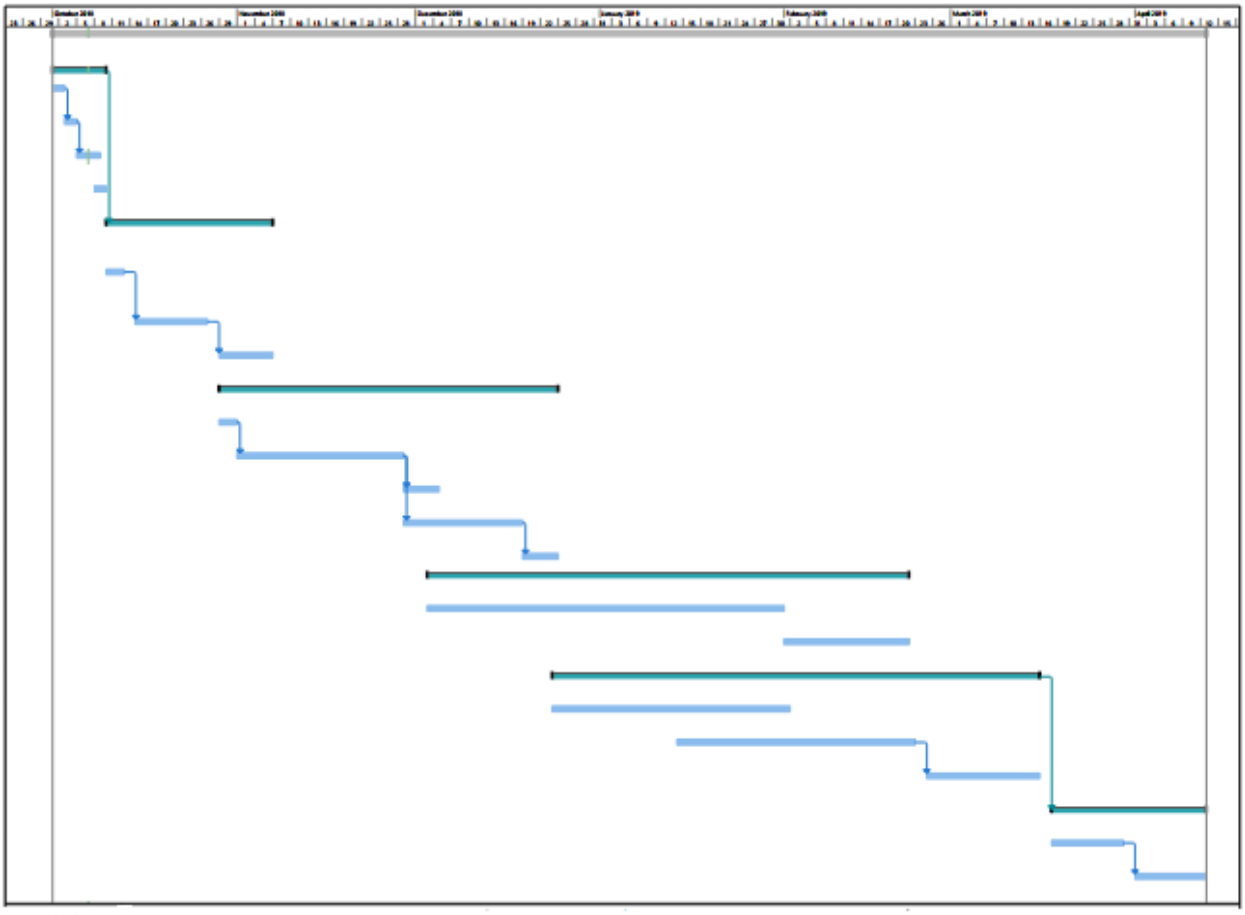
			Scope		Sajjad Hussian
2	Software Requirement Specification Document	2	Requirement Elicitation Requirement Analysis	12	Amna Khalil Muazam Ali Sajjad Hussian
3	Architecture Diagram	3	Architecture Design	5	Amna Khalil
4	Use Case Diagram	3	✧ UML Design	4	Amna Khalil
5	Fully Dress Use Case	3	✧ UML Design	3	Amna Khalil Sajjad Hussian
6	Domain Model	3	✧ UML Design	2	Muazam Ali
7	Class Diagram	3	✧ UML Design	2	Amna Khalil
8	Sequence Diagram	3	✧ UML Design	5	Amna Khalil Muazam Ali
9	Activity Diagram	3	✧ UML Design	4	Amna Khalil
10	State Machine Diagram	3	✧ UML Design	4	Muazam Ali
11	Entity Relationship Diagram	3	✧ Database Design	3	Sajjad Hussian
12	Software Design Document	3	✧ Architecture Design ✧ UML Design	30	Amna Khalil Muazam Ali Sajjad Hussian
13	User Interface	3	✧ User Interface	6	Amna Khalil

			Design		Muazam Ali Sajjad Hussian
14	HMS system	4 5 6	✧ Implementation Coding & Debugging Change Management ✧ Testing ✧ Deployment	135	Amna Khalil Muazam Ali Sajjad Hussian

1.7.3. Gantt Chart

WBS	Task Name	Start Date	Finish Date	Duration (Days)
1	**Project Planning**	**20-Oct-2018**	**28-Oct-2018**	**10**
1.1	Problem Statement	20-Oct-18	22-Oct-18	3
1.2	Existing System Review	23-Oct-18	24-Oct-18	2
1.3	Project Scope	25-Oct-18	26-Oct-18	2
1.4	Project Planning	27-Oct-18	28-Oct-18	2
2	**Software Requirements**	**29-Oct-2018**	**10-Nov-2018**	**12**
2.1	Final Doc. Introduction	29-Oct-18	31-Oct-18	3
2.2	Requirement Elicilation	1-Oct-18	04-Nov-18	4
2.3	Requirement Analysis	05-Nov-18	10-Nov-18	5
3	**Design**	**11-Nov-2018**	**20-Dec-2018**	**38**
3.1	Architecture Design	11-Nov-18	15-Nov-18	5
3.2	UML Design	16-Nov-18	09-Dec-18	24
3.3	Database Design	10-Dec-18	13-Dec-18	3
3.4	User Interface Design	14-Dec-18	20-Dec-18	6

4	**Implementation**	**21-Dec-2018**	**18-Feb-2019**	**60**
4.1	Coding And Debugging	21-Dec-18	08-Feb-19	50
4.2	Change Management	09-Feb-19	18-Feb-19	10
5	**Testing**	**19-Feb-2019**	**09-Apr-2019**	**50**
5.1	Unit Testing	19-Feb-19	10-Mar-19	20
5.2	Integration Testing	11-Mar-19	30-Mar-19	20
5.3	System Testing	01-Apr-19	10-Apr-19	10
6	**Deployment**	**11-Apr-2019**	**5-May-2019**	**25**
6.1	System Deployment	11-Apr-19	25-Apr-19	15
6.2	Deployment Testing	26-May-19	05-May-19	10



Chapter 2

Software Requirement Specifications

2.1. Introduction

2.1.1. Purpose

The purpose of HMS is the billing, insurance processing modules effectively process clinical data for efficient payout for health encounters. Main objectives of a Hospital Management System are:

Design a system for better patient care.

Provide details on demand to management for better decision making.

2.1.2. Document Conventions

Major headings are in bold 16pt font, and concurrent headings in bold 14 pt font. Sections are in the format of x.y, where x and y are real, positive integers.

2.1.3. Intended Audience and Reading Suggestions

This document is intended for use by current and future development teams and outside project advisers and stakeholders who are associated with HMS system. This document contains technical information regarding the performance, specifications, and functional requirements.

2.1.4. Product Scope

HMS used in any Hospital, Clinic for maintaining patient details and their test results. And will provide billing system, and also provide a information of patients, doctors, medicine etc. The system will be used to allocate beds to patients on a priority basis, and to assign doctors to patients in designated wards as need arises. Doctors will also use the system to keep track of the patients assigned to them. Nurses who are in direct contact with the patients will use the system to keep track of available beds, the patients in the different wards, and the types of medication required for each patient. This system implemented because the current system in use is a paper-based system. It is too slow and cannot provide updated lists of patients within a reasonable time frame. Doctors must make rounds to pick up patients treatment cards in order to know whether they have cases to treat or not.

2.1.5. References

https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database

https://www.google.com/search?q=srs+hospital+management+system&oq=srs+hosp&aqs=chrome.2.69i57j0l5.3242j0j7&sourceid=chrome&ie=UTF-8

2.2. Overall Description

2.2.1. Product Perspective

This project gives the procedural approach how a patient gets treatments, detail about routine checkup, registration , reports and how a patient gets appointment from front desk staff, other different criteria and how billing is calculated and card facilities etc. This HMS is a highly performance database system designed especially for keeping records for patients, doctors, details result for each patient, medical stock in medical store and medicine prescribe to each patient.

2.2.2. Product Functions

The data represented in hospital management application will perform following major function as:

- **Patients Details:** It includes in patients and out patient’ s details**.**
- **Task:** It includes different task which are use in hospital,
- **Billing Detail:** The software will help to calculate the bill much quicker and simple way. The enables the organization to keep information in efficient way.
- A **discipline involves records** keeping. Record keeping is an important aspect of every institution’ s day to day operations.
- The system will be used to **allocate Room to patients** on a priority basis, and to assign doctors to patients in designated wards as need arises.

2.2.3. User Classes and Characteristics

The software is developed such the total appearances of the product to make it more user friendly. The operation will provide with login id and password, General user with the basics computers skills, can use this system.

In this HMS users are:

Doctor:	The doctor will access and update every patient checkup data, prescribe medicine and see reports result.
Receptionist:	The receptionist handle various enquire about patient registration, Doctor visit schedule, enquiry of patient.
System Administrator:	System administrator will inspect the overall performance of the system. He will have the rights to grant permissions.
Pharmacist	He will update data about the patient medicine.

2.2.4. Operating Environment

The Hospital Management system using Microsoft word use for documentation, web service requires as Apache, SQL database running on the off a cloud windows, browser: Chrome etc.

2.2.5. Design and Implementation Constraints

- **Database:**

  The system shall use the MySQL database, which is open source and free.

- **Operating System:**

  Development environment shall be window 8.

- **Web-Based:**

Its is web-based application.

- ✧ The HMS system web server must be able to run PHP

2.2.6. User Documentation

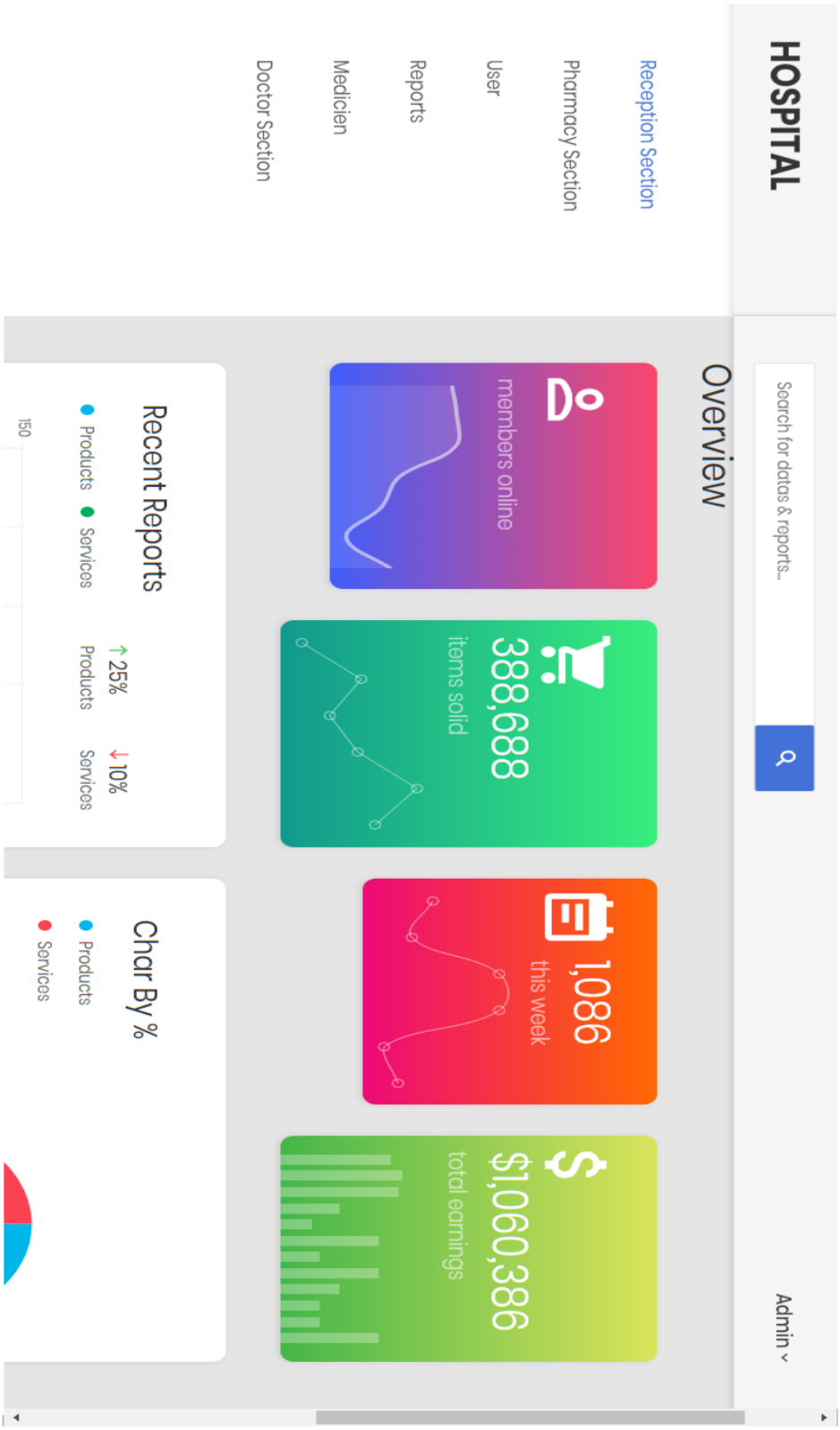
The application will come with an about which will allow users to access the offline and online HTML help manual. This manual will be updated with each new service pack. Other user documentation includes one user manual for lowest level users, one technical document describing the functionality of the sub-section in detail for use of technicians, one copy of documentation and link to current source for future contributors.

2.2.7. Assumptions and Dependencies

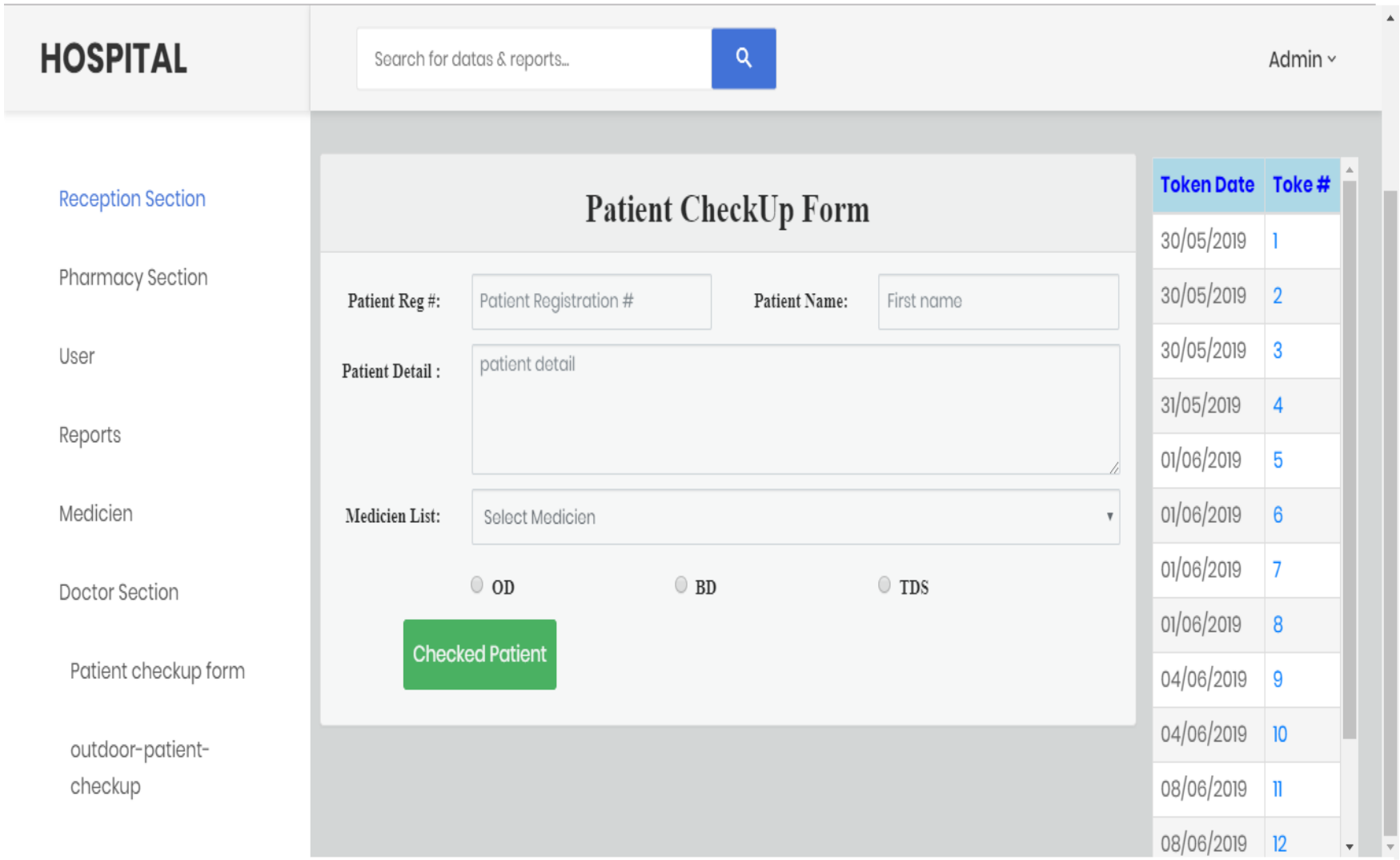
- ➢ It is assumed that 4 to 5 computer will available before the system installed and tested.
- ➢ Hospital have trained staff to take care of the system.
- ➢ All data entered will be correct and up to date.

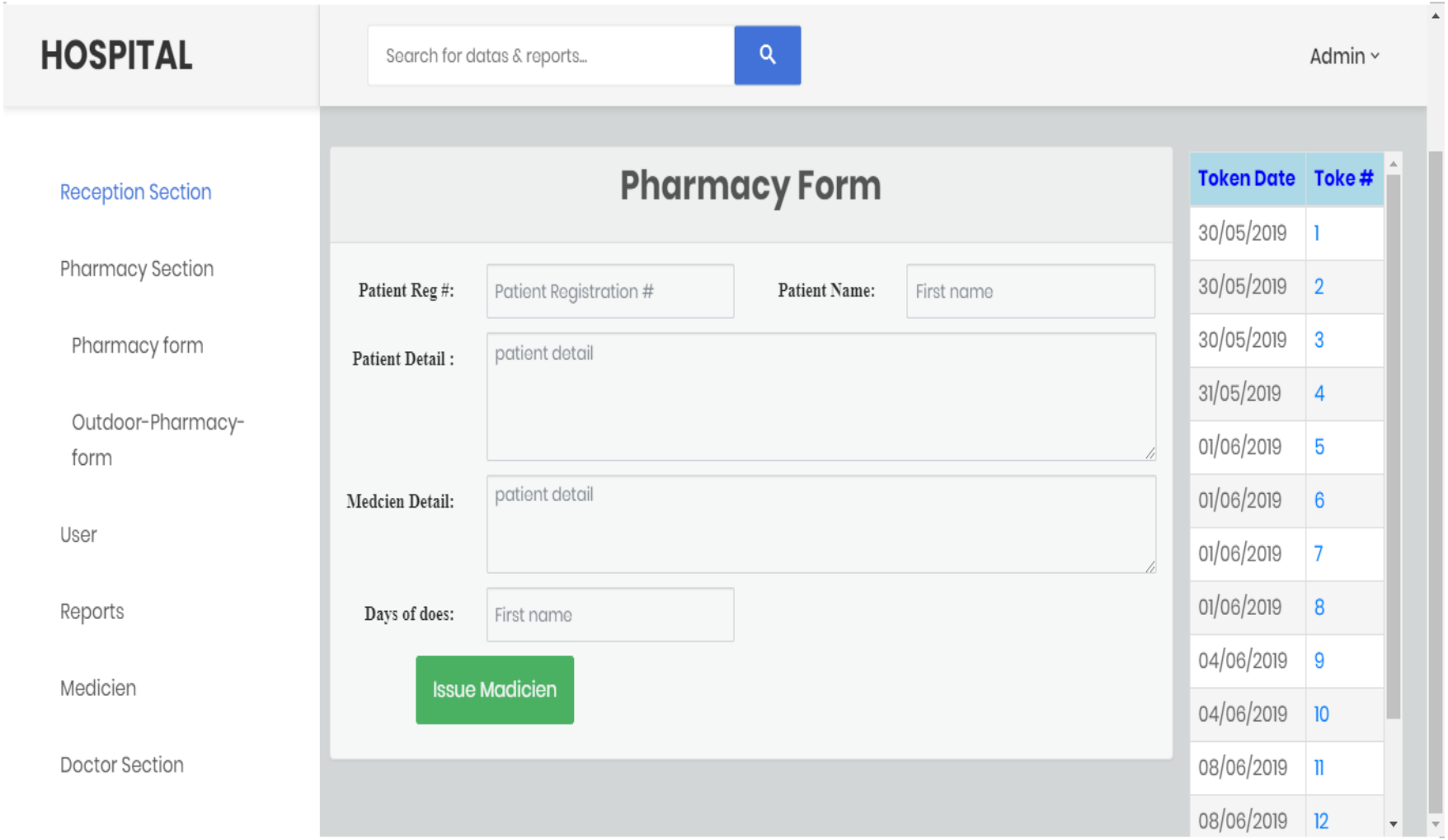
2.3. External Interface Requirement:

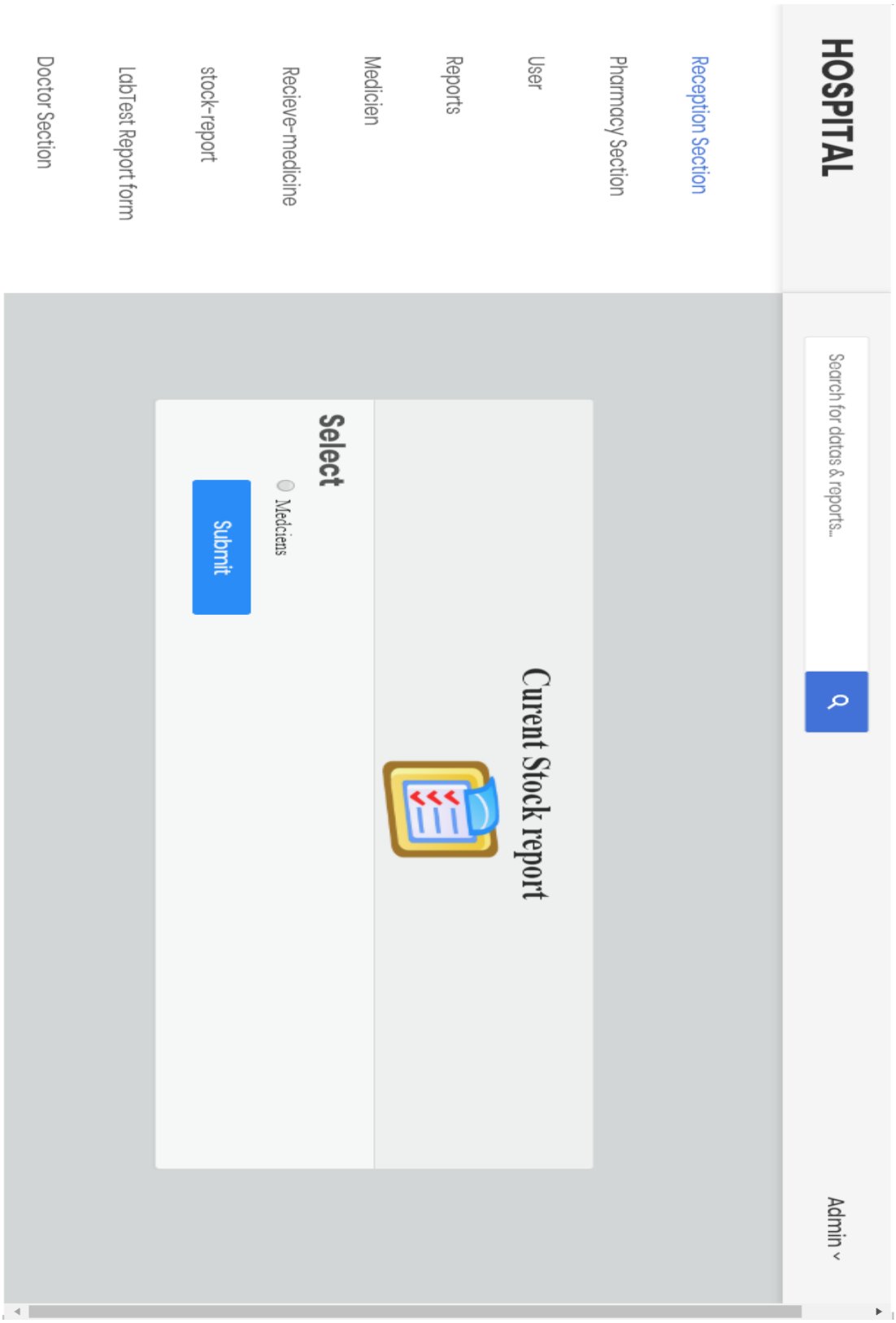
2.3.1. User Interfaces


HOSPITAL
Reception Section
Pharmacy Section
User
Reports
Medicien
Doctor Section
Search for datas & reports...
Admin
Overview
members online
388,688
items solid
1,086
this week
$1,060,386
total earnings
Recent Reports
Products
Services
25%
Products
10%
Services
150
Char By %
Products
Services

HOSPITAL
Search for datas & reports...
Admin
Reception Section
Pharmacy Section
User
Reports
Medicien
Doctor Section
Patient Registration Form
[Token Number]
14
First Name:
First name
Last Name:
Last Name
C.N.I.C:
Phone #:
Address :
Address
General
Private
Cash Received :
Enter Cash
Registar


HOSPITAL
Search for datas & reports...
Admin
Reception Section
Pharmacy Section
User
Reports
Medicien
Doctor Section
Patient checkup form
outdoor-patient-checkup
Patient CheckUp Form
Patient Reg #:
Patient Registration #
Patient Name:
First name
Patient Detail :
patient detail
Medicien List:
Select Medicien
OD
BD
TDS
Checked Patient
Token Date
Toke #
30/05/2019 1
30/05/2019 2
30/05/2019 3
31/05/2019 4
01/06/2019 5
01/06/2019 6
01/06/2019 7
01/06/2019 8
04/06/2019 9
04/06/2019 10
08/06/2019 11
08/06/2019 12


HOSPITAL
Search for datas & reports...
Admin
Reception Section
Pharmacy Section
Pharmacy form
Outdoor-Pharmacy-form
User
Reports
Medicien
Doctor Section
Pharmacy Form
Patient Reg #:
Patient Registration #
Patient Name:
First name
Patient Detail :
patient detail
Medcien Detail:
patient detail
Days of does:
First name
Issue Madicien
Token Date
Toke #
30/05/2019 1
30/05/2019 2
30/05/2019 3
31/05/2019 4
01/06/2019 5
01/06/2019 6
01/06/2019 7
01/06/2019 8
04/06/2019 9
04/06/2019 10
08/06/2019 11
08/06/2019 12


HOSPITAL
Search for datas & reports..
Admin
Reception Section
Pharmacy Section
User
Reports
Medicien
Recieve-medicine
stock-report
LabTest Report form
Doctor Section
Curent Stock report
Select
Medciens
Submit

2.3.2. Hardware Interfaces

HMS client side service requires a PC able to run windows, scanner, or printer etc

2.3.3. Software Interfaces

- HMS system integrates with multiple software including

- PHP

- MySQL

- Sublime text 3

- Apache

- HMS system also utilizes an Apache server with PHP functionality and both an offline and online Mysql database for inventory management.
- On the client side of the product HMS system must run on better OS.

2.3.4. Communications Interfaces

- ✓ Client (Patient) on Internet will be using HTTP/HTTPS protocol.
- ✓ Client on Internet will be using HTTP/HTTPS protocol.

2.4. System Features

The users of the system should be provide the surety that their account is secure as:-

1. Login page:

Description and Priority:

The user open the application and with after every request, to enter their hospital employee ID. This a highly priority.

Stimulus/Response Sequences:

If ID and password is correct then open home screen.

2. Home page:

Description and Priority:

Home page is first page of this application. In which all information of application.

Stimulus/Response Sequences:

They shall then be prompted to enter the machine ID into the system. Upon selecting a valid machine they will be able to access all commands associated with its systems.

3. Admission:

Description and Priority:

It shall record the patient's symptoms and assign them a room and doctor based on their condition. It shall include also include search, edit, and print report features.

Stimulus/Response Sequences

The user must fill out the form and finalize all information. Once sent, the database will refresh with the newly entered information. Based on the patient conditions, the system will automatically assign them an appropriate doctor with the proper specialization to deal with the patient's ailment.

4. Registration:

Description and Priority:

This feature shall allow users to register a patient into the global database. It shall collect all of the patient's information and previous health records.

Stimulus/Response Sequences:

There shall be text fields for the patient's name, phone number, date of birth, age, address, social security number etc. The user will also have the option to import past health records if the patient has come from a different hospital.

5. Discharge:

Description and Priority:

This feature shall allow users to discharge a patient from the hospital. It shall check for their health care provider's approval first, and then print a packet of information about their visit, any medications they have been prescribed, and any other information necessary.

Stimulus/Response Sequences:

This feature shall be a screen that contains a list of all the documents in queue for the patient. The user shall be able to drag and drop any document into the queue to add to it.

2.4.1.1. Functional Requirements

Registration:

REQ-REG1: Add Patient: Register Patient with (password and username) So then the HMS shall allow front of the desk the new patient to the system.

REQ-REG2: Assign Id: This system provide id for every register patient.

REQ-REG3: The User Interface shall have data fields for the following patient information:

SSN, First Name, Last Name, Other Name, Age, Sex, Date of Birth, Phone, Street Address, City, State, Zip, Referring Doctor's Name, Referring Doctor's Phone number, Pharmacy Name, Pharmacy Location, Pharmacy Number

Record System:

REQ-RS1: The module enables the users to insert, delete, update the information of patients, doctors etc.

Data storage:

REQ-DS1: The system provides data storage capability. This is enhanced by database management system that is capable managing the storage requirements.

Patient Module:

REQ-PMDL1: Data Members: patient_id, name, age, room no, status, phone number.

Task Module:

REQ-TM1: Search bar: The module is used by system admin to mange the system as allow for the searching and of a record for example patient CNIC, etc.

REQ-TM2: User login with a username and password.

REQ-TM3: Display patients, doctors, nurses account details, order status etc.

Billing Module:

REQ-BM1: Total amount which gives the total amount which patient has to pay.

REQ-BM2: Patient room charges, other charges etc.

Database:

REQ-DB1: The **Add** every information when we new equipment add etc.

REQ-DB2: The **delete** information when we remove information.

REQ-DB3: The **module use to store** or procedure the patient reports, Doctors, Receptionist etc.

REQ-DB4: The **update** information after each task.

Searching:
REQ-SRC1: Records should retrieve records on click of a button without unnecessary delays are requested by the use.

REQ-SRC2: HMS provide search button for information get from HMS site.

Retrieval of data:
REQ-SF1-ROF1: It's should be easier to retrieve data from database through procedure design to ensure efficiently

Generate Report:
REQ-SF1-GP1: Patient information: The HMS shall generate report on patient about the following information like name, id, phone no, and doctor name which was assigned.

2.5. Other Nonfunctional Requirements

2.5.1. Performance Requirements

- ✧ The software should have high performance and low failure rates.
- ✧ The hardware and software should be able to receive data from databases with high baud rates, ranging from Mbps to Gbps. Machines should have all recent Windows updates installed, and have their security not compromised by viruses.
- ✧ Machines should solely be used for operation of the software, in order to maximize performance and security.

2.5.2. Safety Requirements

- ✧ The safety of the patients, all patients must wear lead vests for the body part being X-Ray.
- ✧ The X-Ray equipment must be regularly maintained, inspected, and used responsibly.

- ✧ The system shall not perform diagnoses. The User is responsible for performing diagnoses.
- ✧ The system shall only facilitate the display of information to aid the User in making quick and timely diagnoses; thus, the information must be displayed visibly and with as much clarity as the hardware infrastructure was capable of providing.

2.5.3. Security Requirements

- ✧ All data receiving and transmissions should be done higher encryption, in order to keep the patient's private medical and social security information out of the wrong hands.
- ✧ Before any user can access the system, they shall be required to input a company username, an ID number, and a password.
- ✧ Each password shall be required to be between 6-12 characters in length and shall be required to contain at least one capital letter, one number, and one special character.
- ✧ Passwords will need to be changed every half-year, with a unique password.

2.5.4. Software Quality Attributes

Flexibility, re-usability, robustness, and maintainability of the HMS system should be maximized, in order for clients to be able to deploy custom settings of the HMS to their individual hospital network needs.

2.5.5. Business Rules

Admin login with password and user.

Website will not responsible for damage of product.

User has to follow terms and conditions of website.

2.6. Other Requirements

A degrade mode of operation should be possible in which each system can operate independently of central scheduling. The software shall have failure and error code act as safety net.

Usability: This website easy to use and easy to learn how to

operate.

Portability: This software can be used on different devices like

tablets, mobiles, laptops etc.

Robustness: This software can handle many problems itself like if

user forgets his password so system gives it option to re-write name and user password.

Flexibility: The functionality of this software can be increased after development because documentation is available.

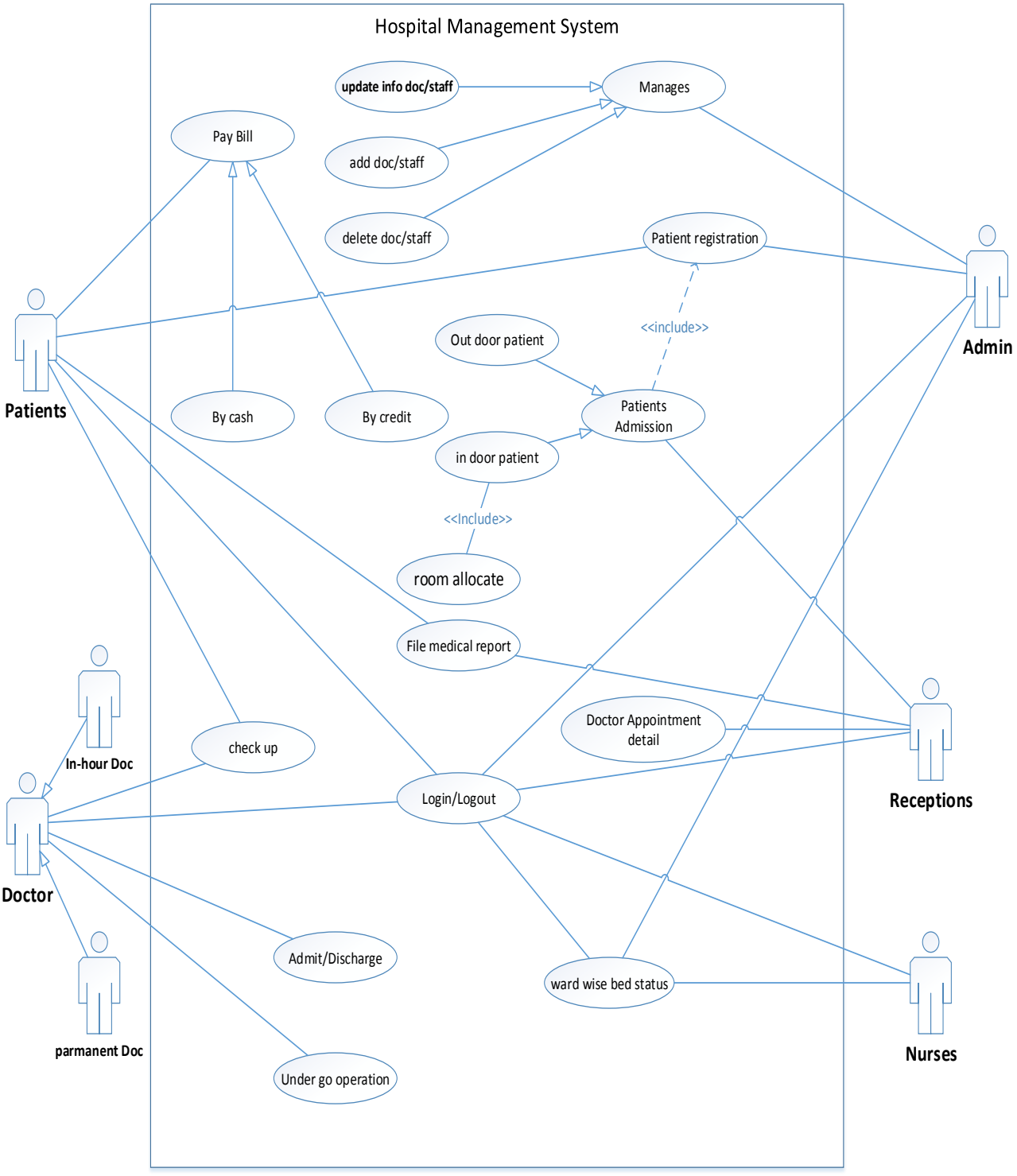
Chapter 3

Use Case Analysis

Chapter 3: System Analysis

Systems Analysis in which collecting and interpreting facts, identifying the problems, and decomposition of a system into its components as use-case and fully-dressed use-case. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose detail about relation between actors as (doctor, nurses, receptionist, patient) and use cases in hospital management system (HMS). In which detail about what Hospital management should do.

3.1. Use Case Model



3.2. Fully Dressed Use Cases

Fully dressed use case		
1.	**Use case name**	Registration
2.	**Scope**	Enter in hospital
3.	**Level**	User goal
4.	**Primary Actor**	Patient
5.	**Stakeholder**	Patient and admin
6.	**Precondition**	null
7.	**Post condition**	Added Patient
8.	**Success Scenario**	a) Field all data b) Correct all field c) Submit register button d) Display Registration successfully
9.	**Extension**	View login task.
10.	**Alternative scenario**	a) If patient want registered then add information. b) if correct then add otherwise no registered. c) If error then message: please filled correct field.

Fully dressed use case		
1.	**Use case name**	Login
2.	**Scope**	Open information page of hospital
3.	**Level**	User goal

4.	**Primary Actor**	Users
5.	**Stakeholder**	Users and receptionist
6.	**Precondition**	Register in hospital for check up
7.	**Post condition**	Login to system
8.	**Success Scenario**	a) User name and password b) If correct then login c) Display Login for information.
9.	**Extension**	View timing for doc.
10.	**Alternative scenario**	a) If user not added then message display Your password and user incorrect. b) Try again again enter name and password

Fully dressed use case		
1.	**Use case name**	Appointment detail
2.	**Scope**	Fix checkup time with doctor
3.	**Level**	User goal
4.	**Primary Actor**	Receptionist
5.	**Stakeholder**	Patient , doctor
6.	**Precondition**	Login to system
7.	**Post condition**	Allocate timing to patient.
8.	**Success Scenario**	a) Click search doctor available b) If find data, time then c) Allocate Appointment
9.	**Extension**	View schedule register patient

10.	**Alternative scenario**	a) If appointment not done so message display already done appointment

Fully dressed use case		
1.	**Use case name**	Admit in hospital
2.	**Scope**	Admit for checkup
3.	**Level**	User goal
4.	**Primary Actor**	Receptionist
5.	**Stakeholder**	Patient, receptionist
6.	**Precondition**	Allocate time for appointment
7.	**Post condition**	Admit in hospital
8.	**Success Scenario**	a) Check reports detail b) Clicked for admit if need then display c) Admit in hospital
9.	**Extension**	Checkup
10.	**Alternative scenario**	a) Check date and time for patient appointment if not available message no doctor available.

Fully dressed use case		
1.	**Use case name**	Checkup
2.	**Scope**	Doctor checkup of patient
3.	**Level**	User goal
4.	**Primary Actor**	Doctor

5.	**Stakeholder**	Patient, Doctor
6.	**Precondition**	Admit in hospital
7.	**Post condition**	Checkup patient
8.	**Success Scenario**	a) If all task done then store data b) Check patient id by search c) Then check from DB d) Checked patient
9.	**Extension**	Checkup patient
10.	**Alternative scenario**	a) If doctor not available system display message doctor not available

Fully dressed use case		
1.	**Use case name**	Admit/discharge
2.	**Scope**	If patient condition better then doctor will discharge patient other wise admit.
3.	**Level**	User goal
4.	**Primary Actor**	Doctor
5.	**Stakeholder**	Patient, Doctor
6.	**Precondition**	Checkup patient
7.	**Post condition**	Admit/discharge
8.	**Success Scenario**	a) Check reports detail search by ID b) Then check admit or discharge c) Admit/discharge option
9.	**Extension**	About condition of patient
10.	**Alternative scenario**	a) If patient not need treatment then discharge otherwise admit

Fully dressed use case		
1.	**Use case name**	Under go operation
2.	**Scope**	Doctor Assign operation if patient deserve .
3.	**Level**	User goal
4.	**Primary Actor**	Doctor
5.	**Stakeholder**	Patient, Doctor
6.	**Precondition**	Admit/discharge
7.	**Post condition**	Assign operation
8.	**Success Scenario**	a) Check detail of patient by id Assign operation b) Assign operation
9.	**Extension**	About condition of patient.
10.	**Alternative scenario**	Doctor check patient if no need for operation then save record patient is fine

Fully dressed use case		
1.	**Use case name**	Payment Bill
2.	**Scope**	Patient payment bill
3.	**Level**	User goal
4.	**Primary Actor**	Patient
5.	**Stakeholder**	Patient and accountant
6.	**Precondition**	Allocate slip to patient
7.	**Post condition**	Patient bill paid
8.	**Success Scenario**	a) Search detail expenses by ID of patient b) Collect all payment

		c) Generate bill
9.	**Extension**	Check bill slip and then receive it.
10.	**Alternative Scenario**	a) If patient no paid bill hospital expenses b) Message display unpaid

Fully dressed use case		
1.	**Use case name**	By credit
2.	**Scope**	Patient payment bill with credit card
3.	**Level**	User goal
4.	**Primary Actor**	Patient
5.	**Stakeholder**	Patient and accountant
6.	**Precondition**	Patient bill paid
7.	**Post condition**	Patient bill paid by credit
8.	**Success Scenario**	a) Payment checked b) Clicked on credit c) Submit payed d) Patient bill paid by credit
9.	**Extension**	Check bill slip and then receive it.
10.	**Alternative Scenario**	a) If patient paid bill by credit and credit card not work then message display unpaid

Fully dressed use case		
1.	**Use case name**	By debit
2.	**Scope**	Patient payment bill with debit card
3.	**Level**	User goal

4.	**Primary Actor**	Patient
5.	**Stakeholder**	Patient and accountant
6.	**Precondition**	Patient bill paid
7.	**Post condition**	Patient bill paid by debit
8.	**Success Scenario**	a) Payment checked b) Clicked on debit c) Submit payed d) Patient bill paid by credit Patient bill paid by debit e) done
9.	**Extension**	Check bill slip and then receive it.
10.	**Alternative Scenario**	a) If patient paid bill by debit and debit card not work then message display unpaid

Fully dressed use case		
1.	**Use case name**	Medical reports
2.	**Scope**	Receptionist handle medical reports of patients
3.	**Level**	User goal
4.	**Primary Actor**	Receptionist
5.	**Stakeholder**	Patient and receptionist
6.	**Precondition**	Patient bill paid
7.	**Post condition**	Completely work of reports
8.	**Success Scenario**	A. Payment checked B. Clicked on credit C. Submit payed D. Patient bill paid by credit E. Done
9.	**Extension**	Done medical reports of patient

10.	**Alternative scenario**	a) If not exist reports message display record is blank

Fully dressed use case		
1.	**Use case name**	Manages
2.	**Scope**	Admin handle register patient, add, update, delete staff information.
3.	**Level**	User goal
4.	**Primary Actor**	Admin
5.	**Stakeholder**	Patient, staff & Admin
6.	**Precondition**	Completely work of reports
7.	**Post condition**	Online all record manage
8.	**Success Scenario**	a) Search by id of patient b) Click Update/delete/add c) Successful Online all record manage
9.	**Extension**	All online work manage about update, delete, add etc.
10.	**Alternative Scenario**	a) If message does not add maybe already exist Message display ‘already exist’ b) If message does not deleted maybe not exist Message display ‘no exist’ c) If message does not edit maybe no exist Message display ‘no exist’

Fully dressed use case

1.	**Use case name**	Added
2.	**Scope**	Admin handle register patient, add staff information.
3.	**Level**	User goal
4.	**Primary Actor**	Admin
5.	**Stakeholder**	Patient, staff & Admin
6.	**Precondition**	Online all record manage
7.	**Post condition**	Added successfully
8.	**Success Scenario**	A. Search by id of patient B. Add detail of patient C. Click add D. Successful Online all record add
9.	**Extension**	All online work manage about add.
10.	**Alternative scenario**	A) If message does not add maybe already exist Message display 'already exist'

Fully dressed use case		
1.	**Use case name**	Update
2.	**Scope**	Admin handle update patient staff information.
3.	**Level**	User goal
4.	**Primary Actor**	Admin
5.	**Stakeholder**	Patient, staff & Admin
6.	**Precondition**	Online all record manage
7.	**Post condition**	update successfully
8.	**Success Scenario**	a) Search by id of patient b) Update detail of patient c) Click update d) Successful Online all record update

9.	**Extension**	All online work manage about update.
10.	**Alternative scenario**	a) If message does not edit maybe no exist Message display ‘no exist’

Fully dressed use case		
1.	**Use case name**	Delete
2.	**Scope**	Admin handle delete patient staff information.
3.	**Level**	User goal
4.	**Primary Actor**	Admin
5.	**Stakeholder**	Patient, staff & Admin
6.	**Precondition**	Online all record manage
7.	**Post condition**	Delete successfully
8.	**Success Scenario**	a) Search by id of patient b) delete detail of patient c) Click delete d) Successful Online all record Delete
9.	**Extension**	All online work manage about delete.
10.	**Alternative scenario**	a) If message does not deleted maybe no exist Message display ‘no exist’

Fully dressed use case		
1.	**Use case name**	Room allocate
2.	**Scope**	Room available for patient
3.	**Level**	User goal

4.	**Primary Actor**	Receptionist
5.	**Stakeholder**	Patient, receptionist
6.	**Precondition**	Bed allocate
7.	**Post condition**	Room available successful
8.	**Success Scenario**	a) Check detail patient by id b) Click on id search bar c) Find patient detail and select room d) Room allocate
9.	**Extension**	Patient get a room for treatment.
10.	**Alternative Scenario**	a) If no room add because error maybe room already booked.

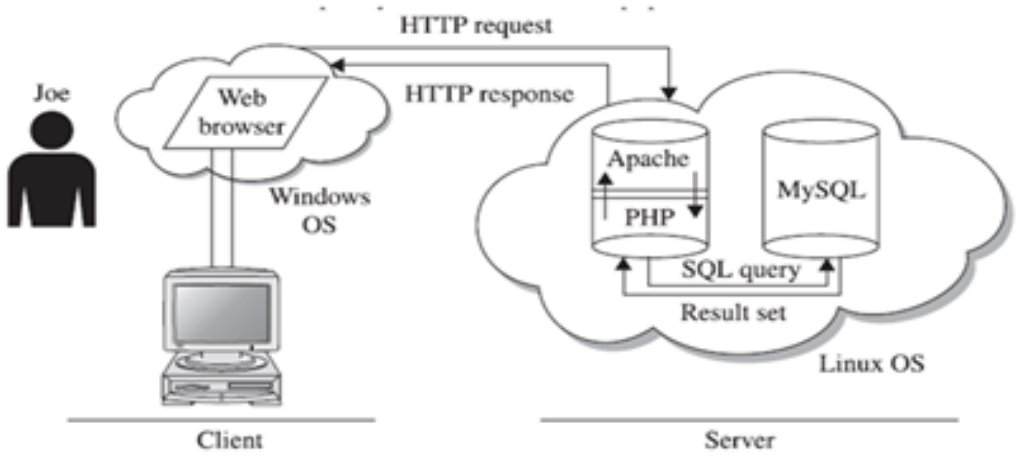
Chapter 4

System Design

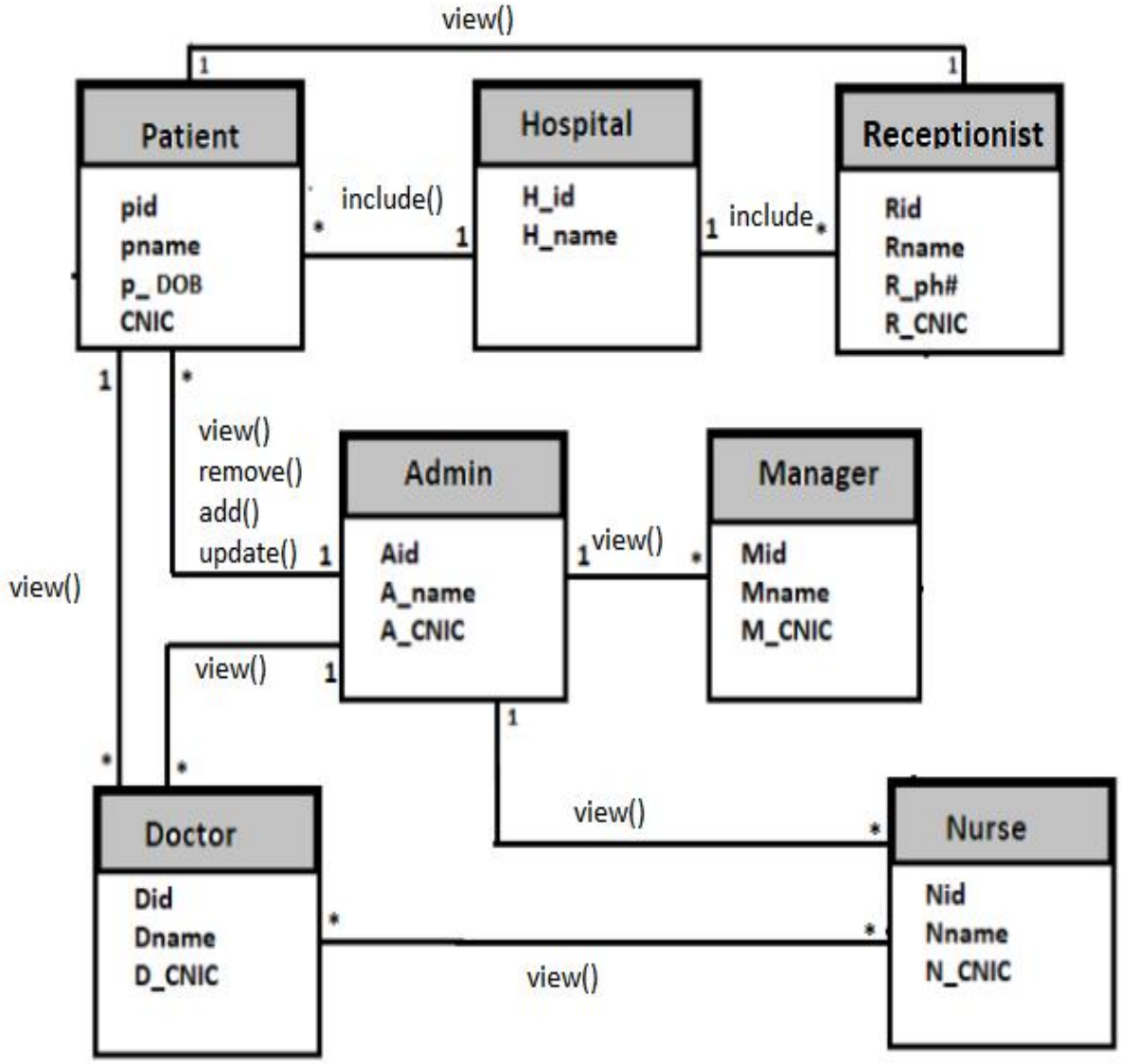
Chapter 4: System Design

In which we planning a new business system or replacing an existing system by defining its components or modules to satisfy the specific requirements as how we replace paper based hospital system to online hospital management system(HMS) and how we accomplish the objects of the Hospital management system through as architecture diagram, domain model and Entity Relationship Diagram etc.

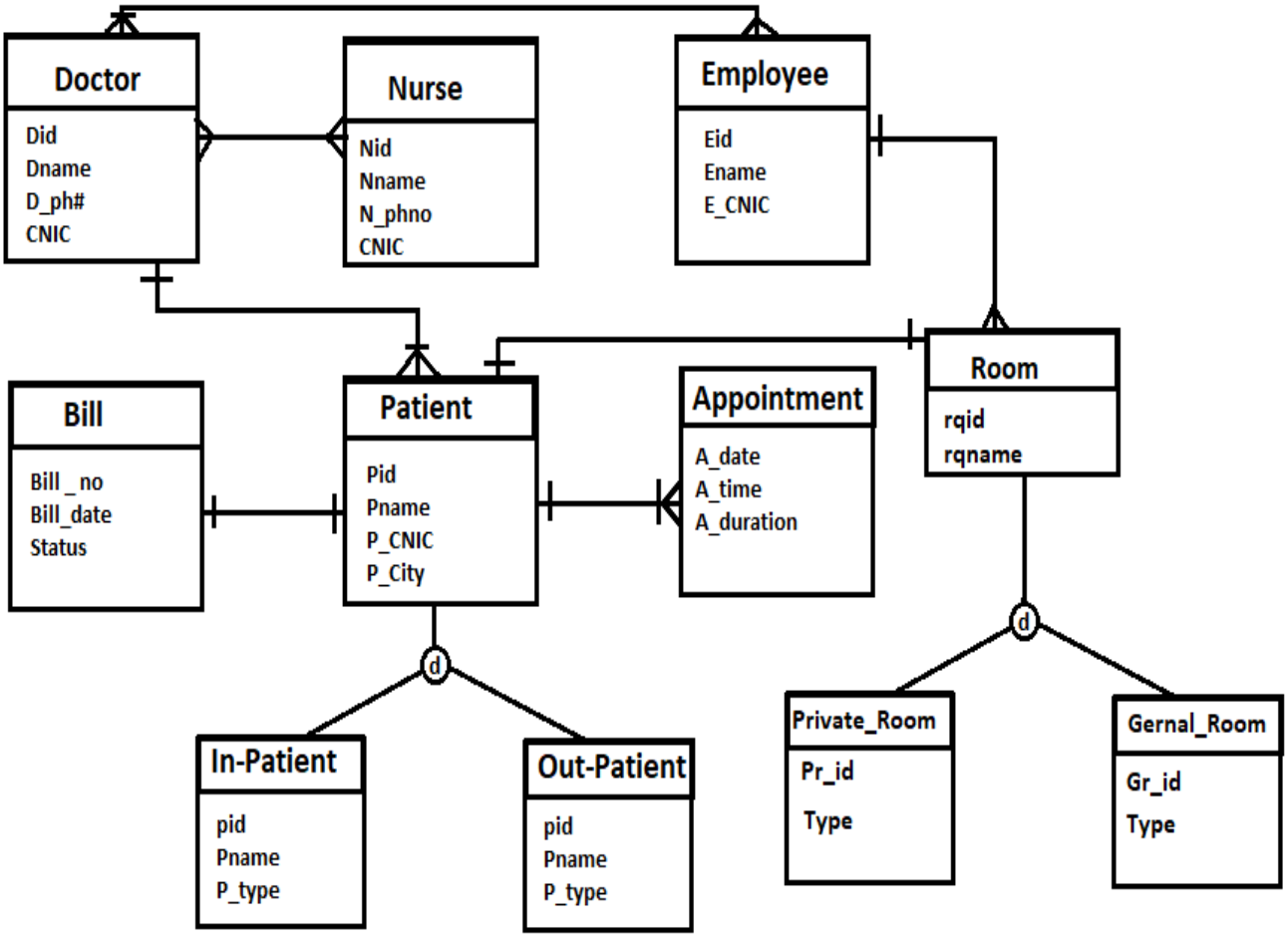
4.1. Architecture Diagram



4.2. Domain Model



4.3. Entity Relationship Diagram with data dictionary



Entities and their Attributes:

1. Admin

Attributes	Data type	Null able	Description
Aid(Primary key)	int	not	Id of this entity
User name	varchar		Name of this entity
Password	varchar		Pass of this entity

2. Doctor

Attributes	Data type	Null able	Description
Did(primary key)	Int	not	Identity of this entity
Dname	Varchar		Name of this entity
CNIC	Varchar		CNIC of this entity
Dph_no	Varchar		Phone no of this entity
DOB	Date		Date of birth of this entity

3. Out door Patient

Attributes	Data type	Null able	Description
pid(primary key)	Int	not	Identity of this entity
Pname	Varchar		Name of this entity

CNIC	Varchar		CNIC of this entity
Dph_no	Varchar		Phone no of this entity
DOB	Date		Date of birth of this entity

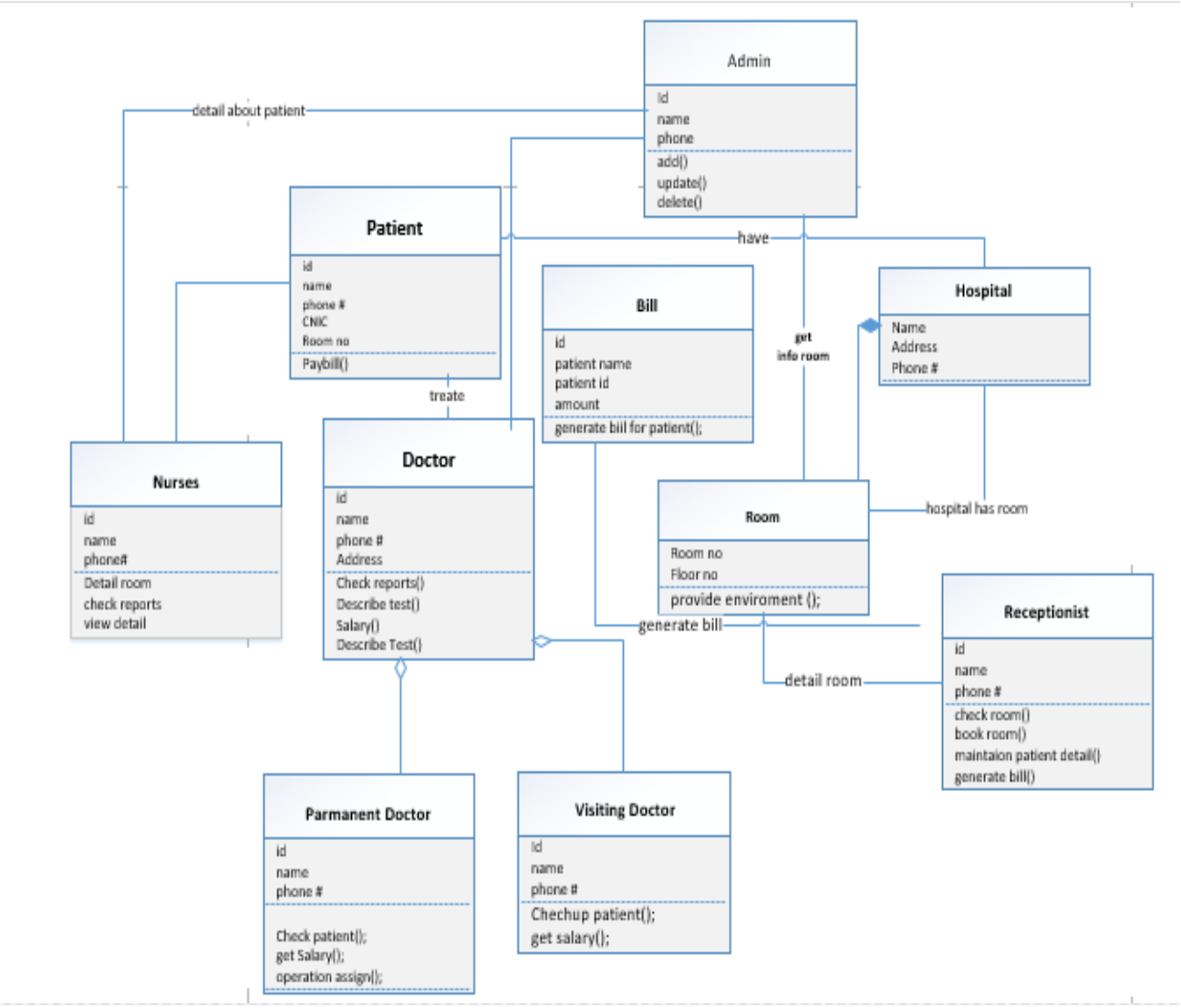
4. Indoor - Patient

Attributes	Data type	Null able	Description
pid(primary key)	Int	not	Identity of this entity
Pname	Varchar		Name of this entity
CNIC	Varchar		CNIC of this entity
Dph_no	Varchar		Phone no of this entity
DOB	Date		Date of birth of this entity
Room no	varchar		Room no
Patient type	varchar		Private or gernal

- In patient (pid, pname, type)

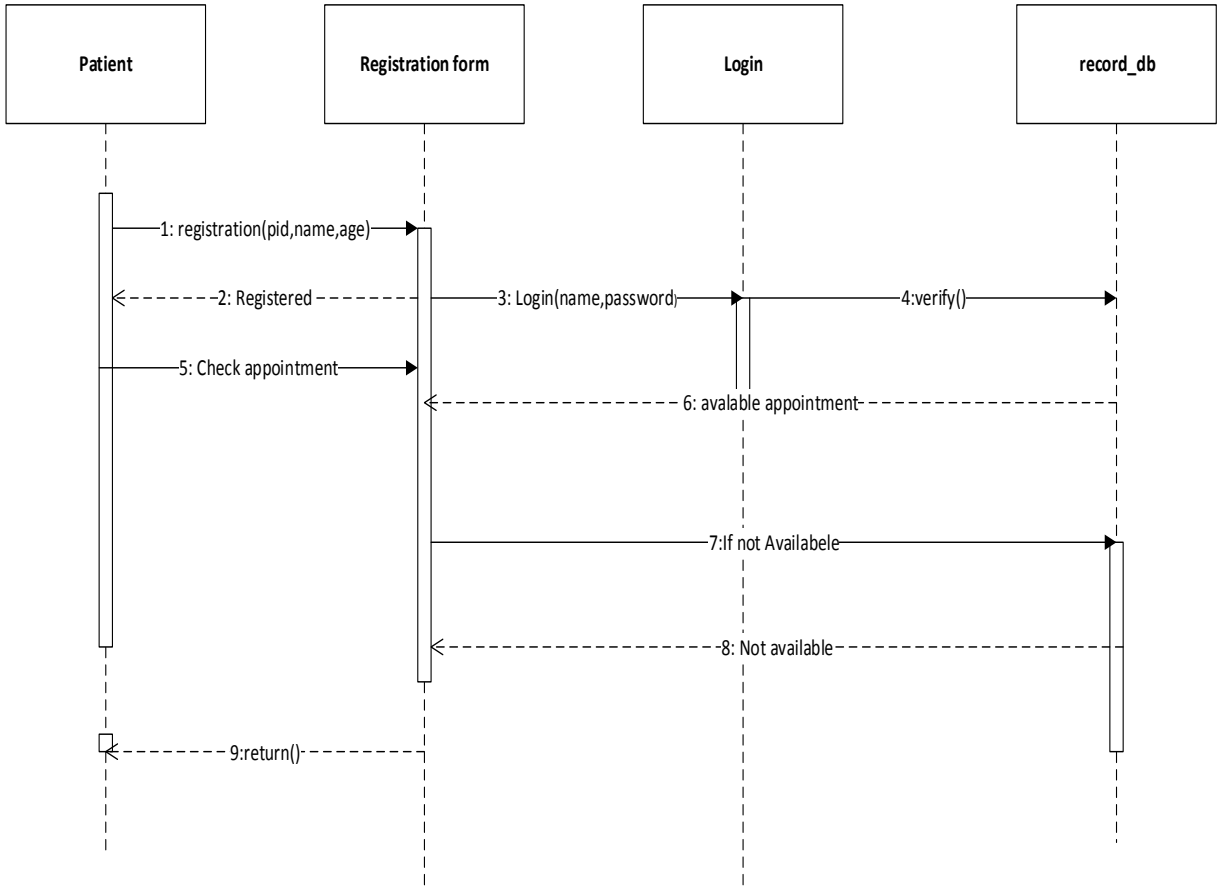
- Out patient (pid, pname, type)

4.4. Class Diagram

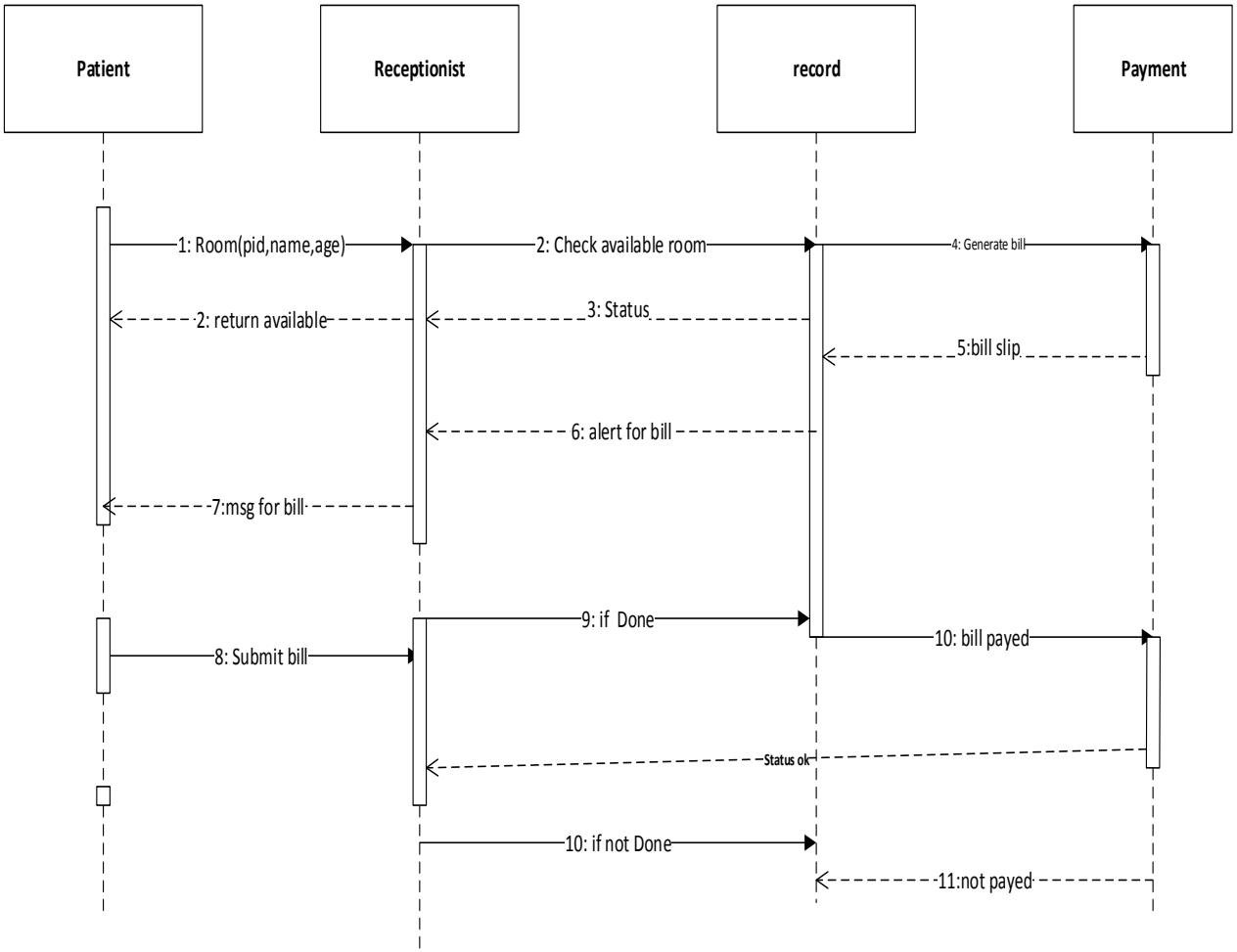


4.5. Sequence / Collaboration Diagram

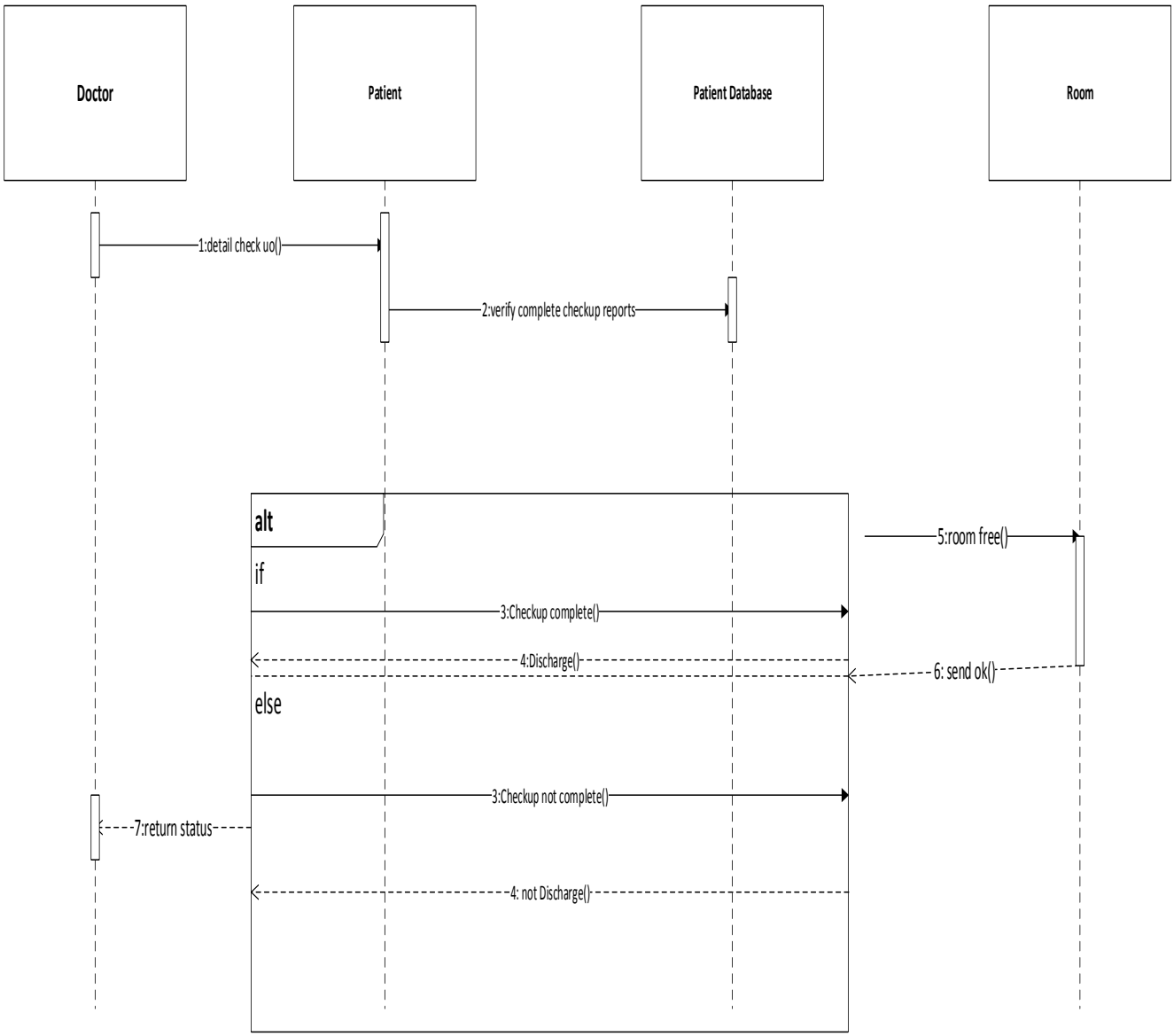
For Registration, Login, Appointment



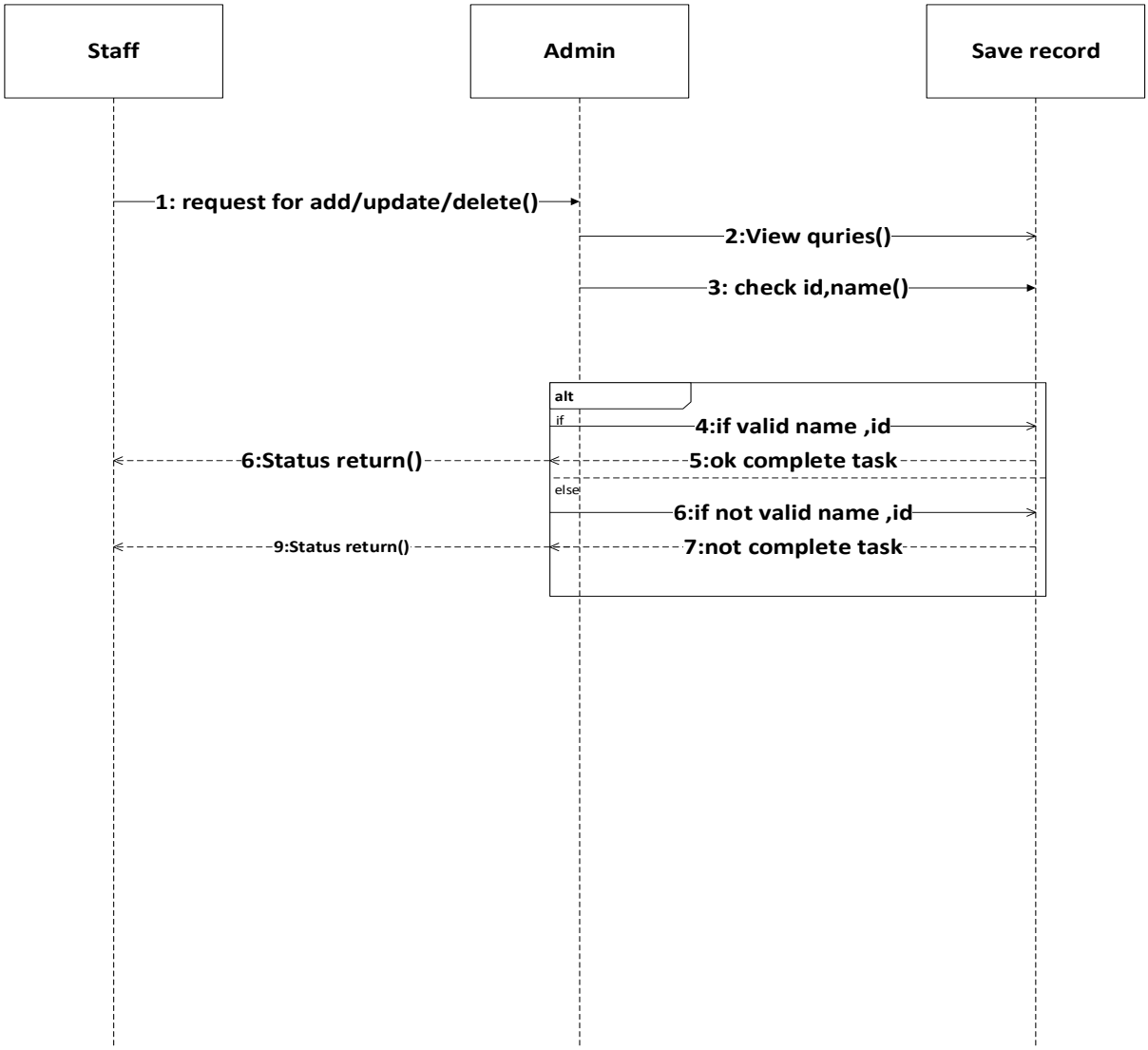
For Admit and room available, Payment



For Discharge


Doctor
Patient
Patient Database
Room
1:detail check uo()
2:verify complete checkup reports
alt
if
3:Checkup complete()
4:Discharge()
5:room free()
6: send ok()
else
3:Checkup not complete()
7:return status
4: not Discharge()

For Managing Task



4.6. Operation contracts

For Register

Operation Contract	
Operation name:	Register
Cross Reference:	Patient
Pre Condition:	Null
Post Condition:	**Object:** User **Attributes:**Username, password,email **Association:**between patient and register.

For Login

Operation Contract	
Operation name:	Login
Cross Reference:	Users
Pre Condition:	Registration
Post Condition:	**Object:** User **Attributes:**Username, password **Association:**between user and login

For Appointment

Operation Contract	
Operation name:	Appointment
Cross Reference:	Receptionist
Pre Condition:	Login
Post Condition:	**Object:** Receptionist **Attributes:**Pid, pname, doc_id **Association:**between receptionist and appointment

For Admit

Operation Contract	
Operation name:	Admit
Cross Reference:	Receptionist
Pre Condition:	Appointment
Post Condition:	**Object:**Receptionist **Attributes:**id,name **Association:**between receptionist and admit.

For checkup

Operation Contract	
Operation name:	Checkup
Cross Reference:	Doctor
Pre Condition:	Admit
Post Condition:	**Object:Doctor** **Attributes:**id,name,age **Association:**between Doctor and checkup

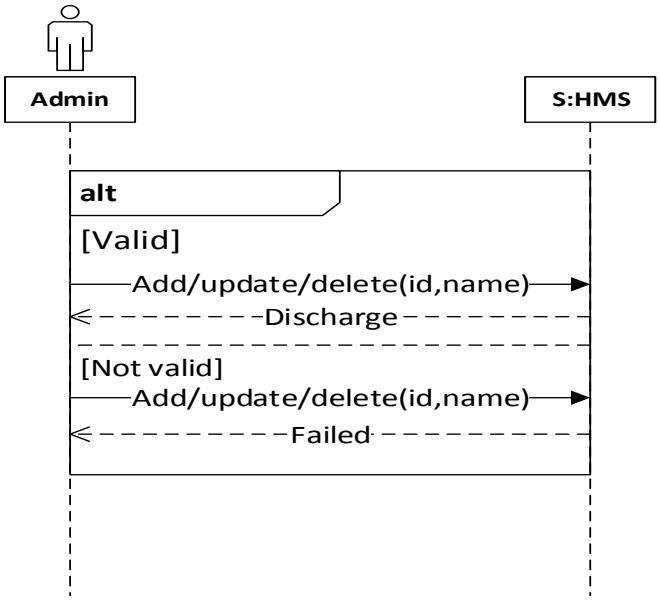
For Payment

Operation Contract	
Operation name:	Payment
Cross Reference:	Patient
Pre Condition:	Discharge
Post Condition:	**Object:**Doctor **Attributes:**id,name **Association:**between doctor and payment

For Discharge

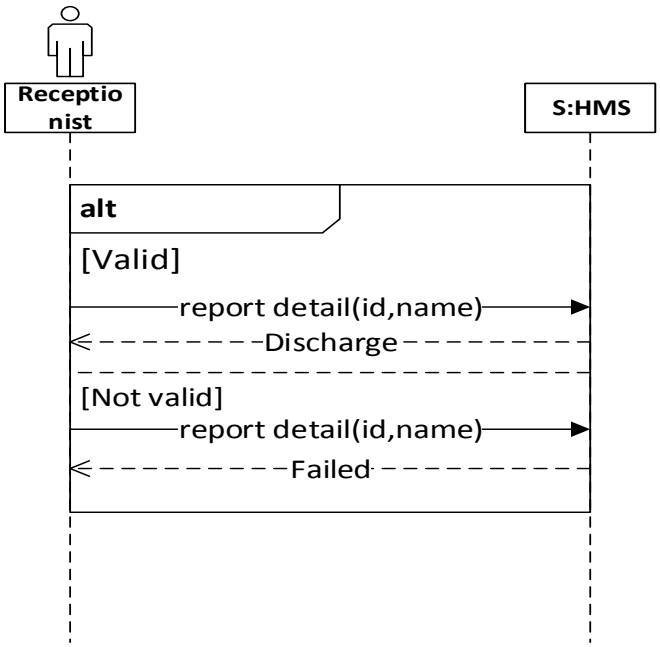
Operation Contract	
Operation name:	Discharge
Cross Reference:	Doctor
Pre Condition:	Checkup
Post Condition:	**Object:** Doctor **Attributes:** id,name **Association:** Between Doctor and Discharge

For Management



Operation Contract	
Operation name:	Manages
Cross Reference:	Receptionist
Pre Condition:	Checkup
Post Condition:	**Object:**Doctor **Attributes:**id,name **Association:**between Doctor and Discharge

For Report



Operation Contract	
Operation name:	Report
Cross Reference:	Receptionist

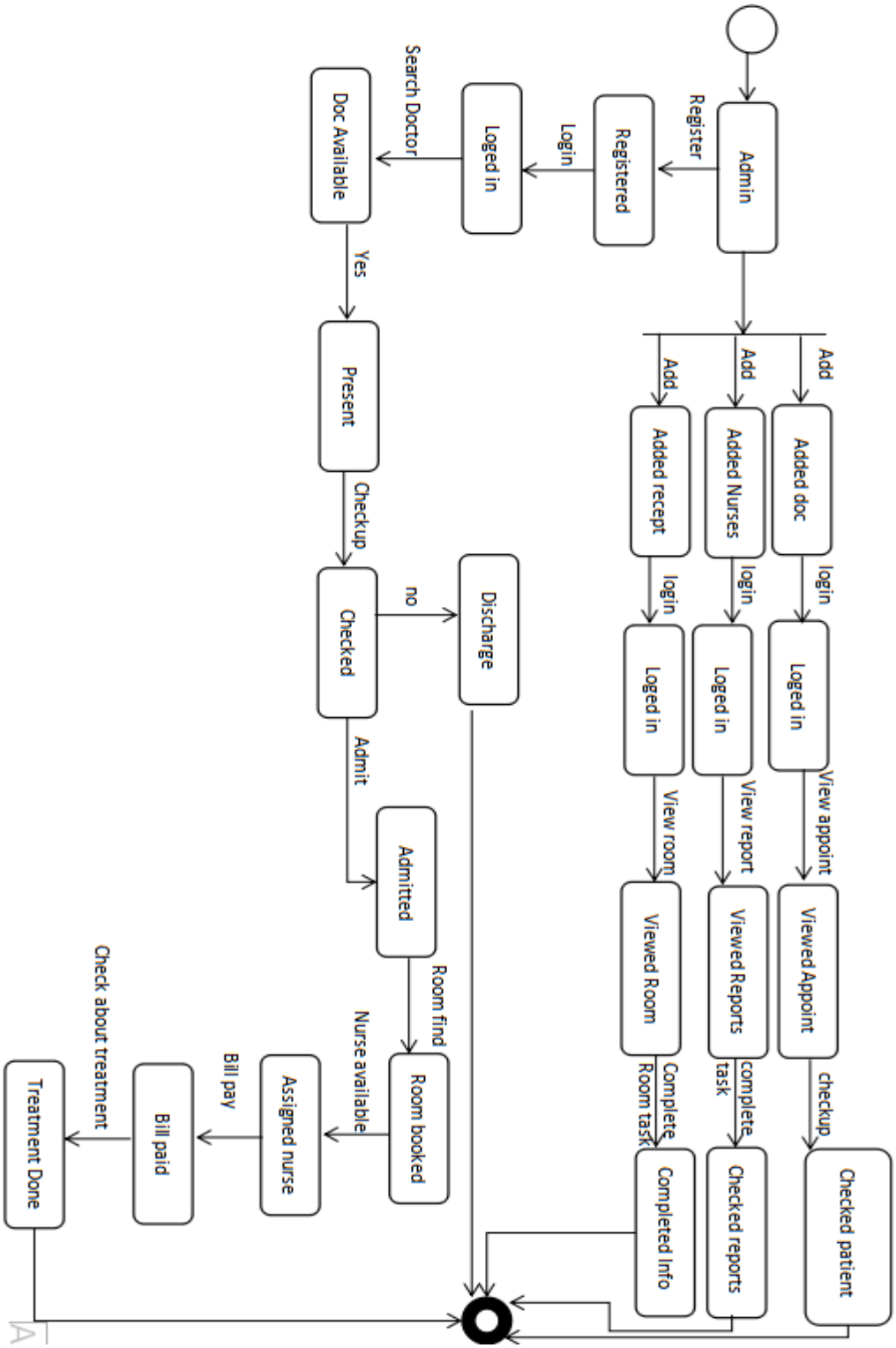
Pre Condition:	Discharge
Post Condition:	**Object:**Receptionist **Attributes:**id,name **Association:**between Receptionist and report

For Room Allocate

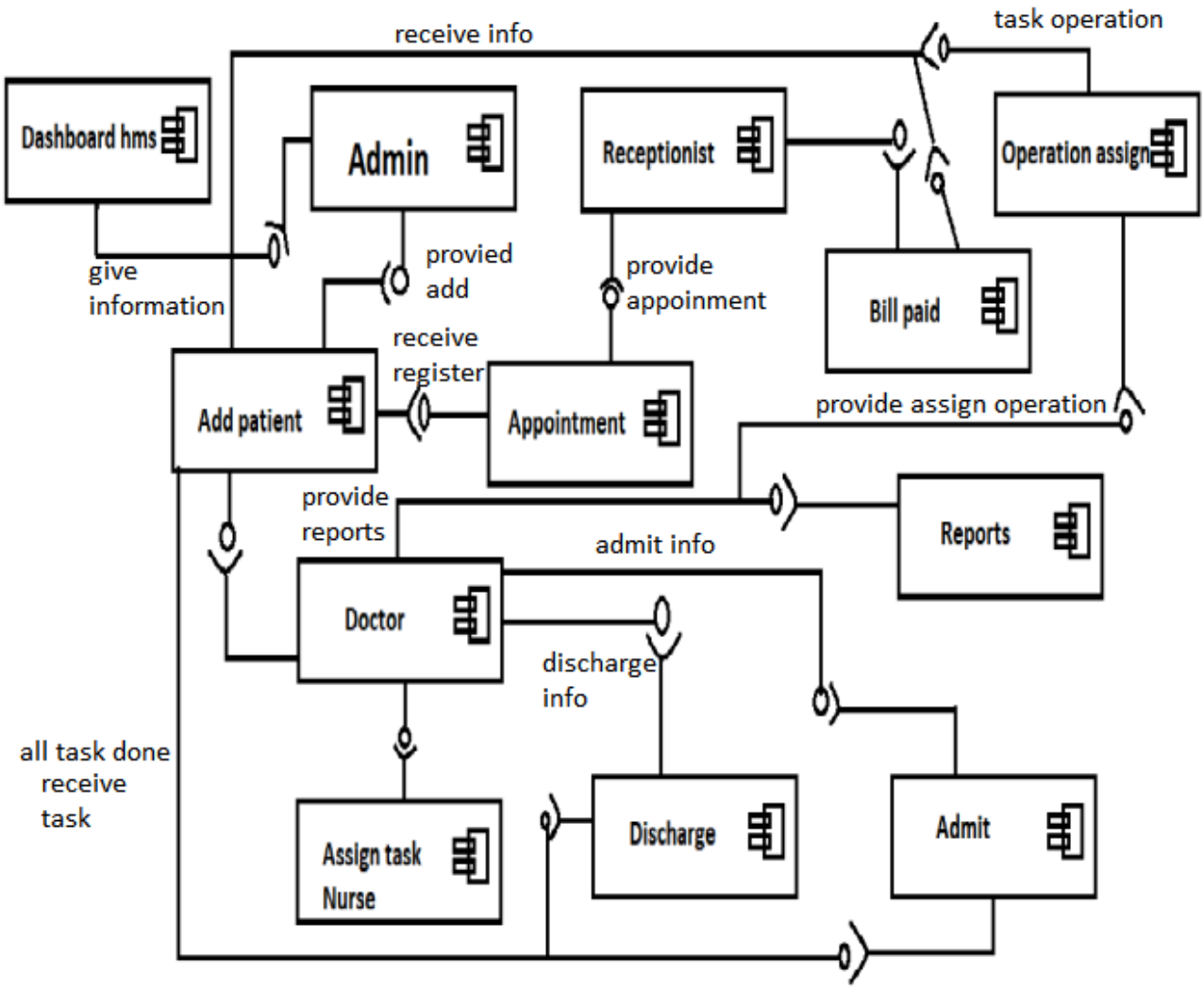
Operation Contract	
Operation name:	Rooms
Cross Reference:	Receptionist
Pre Condition:	Reports
Post Condition:	**Object:**Receptionist **Attributes:**id,name **Association:**between Receptionist and room

4.7. Activity Diagram

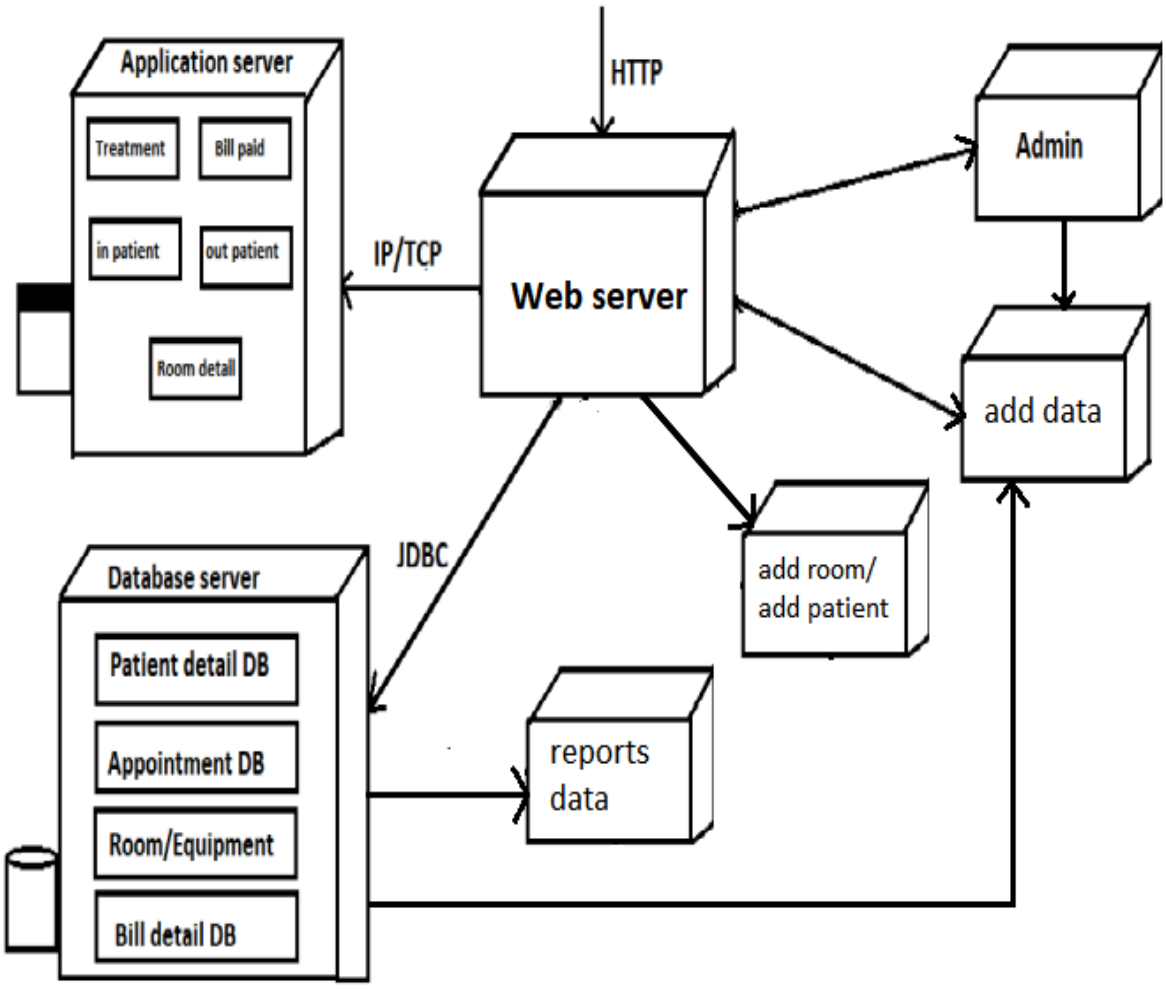
4.8. State Transition Diagram



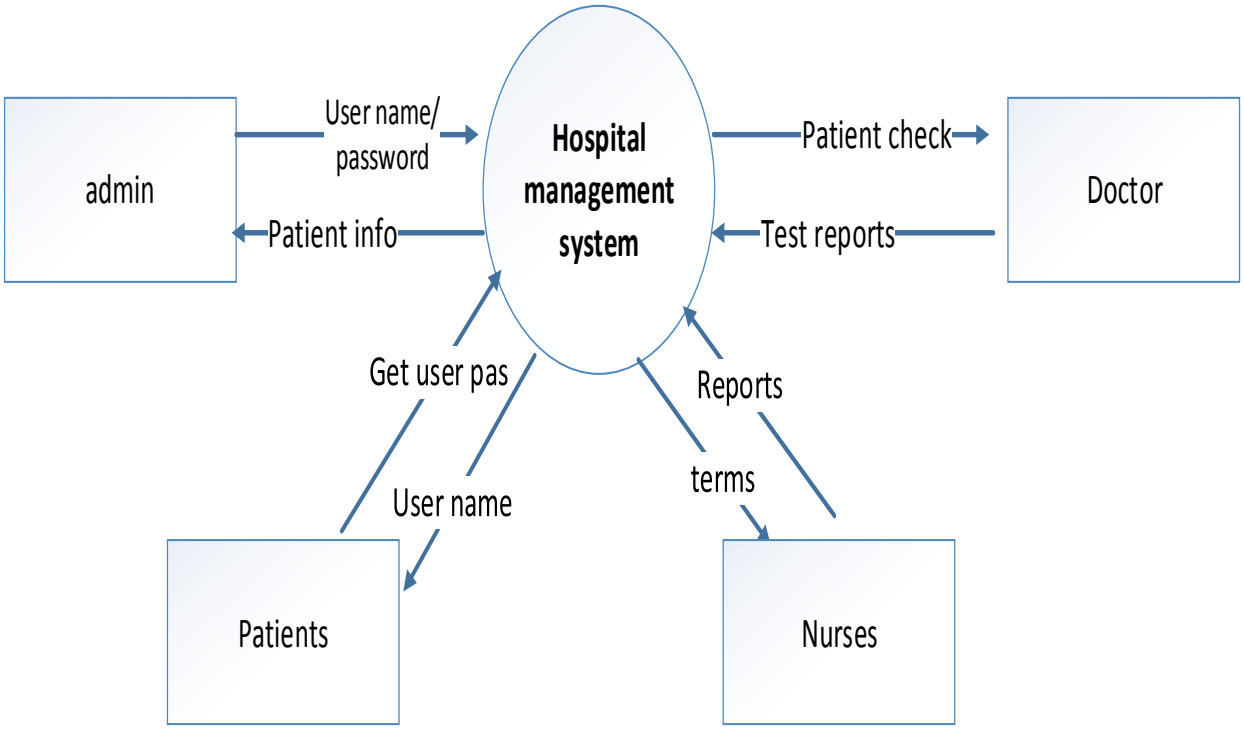
4.9. Component Diagram



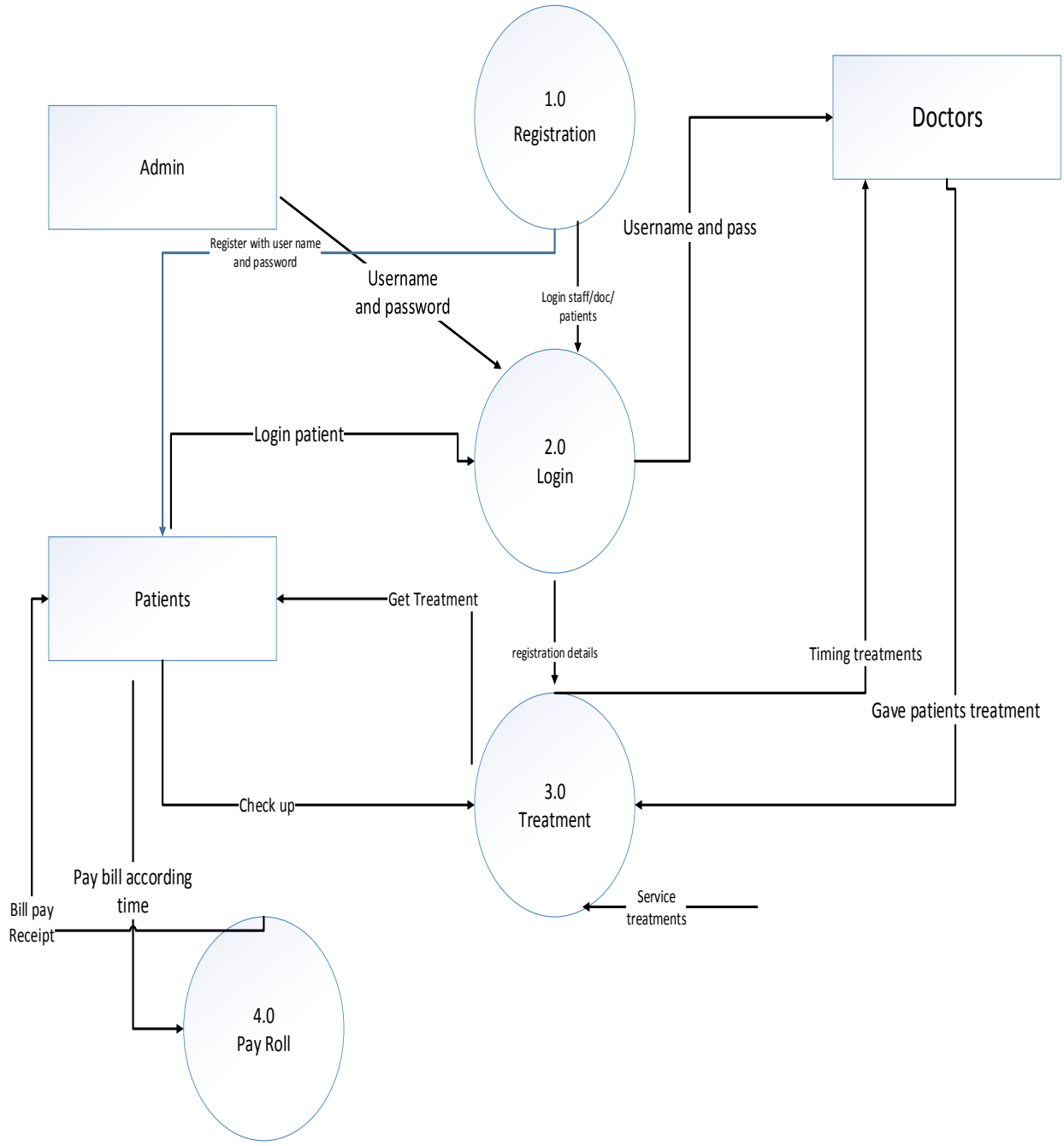
4.10. Deployment Diagram



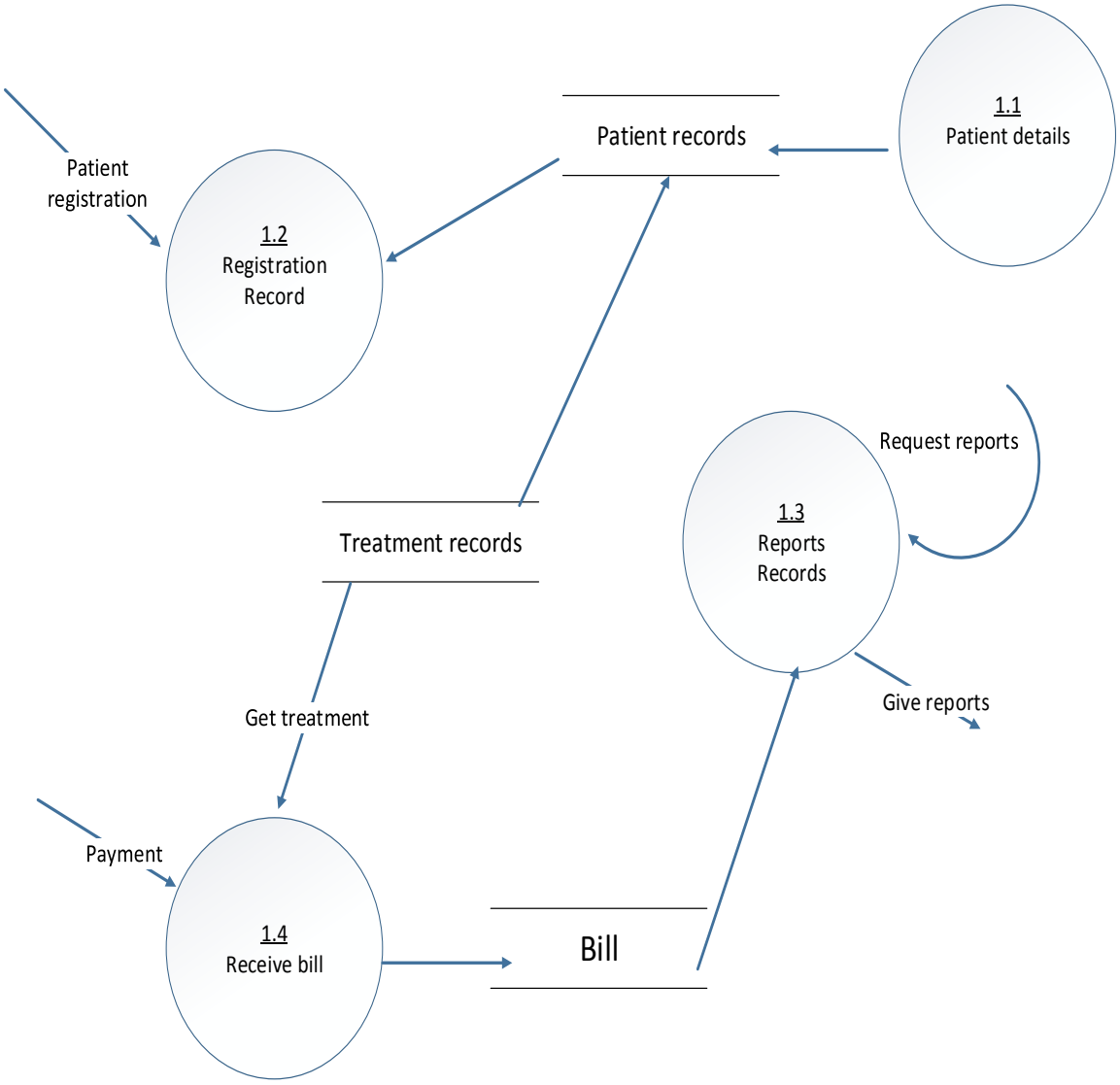
4.11. Data Flow diagram



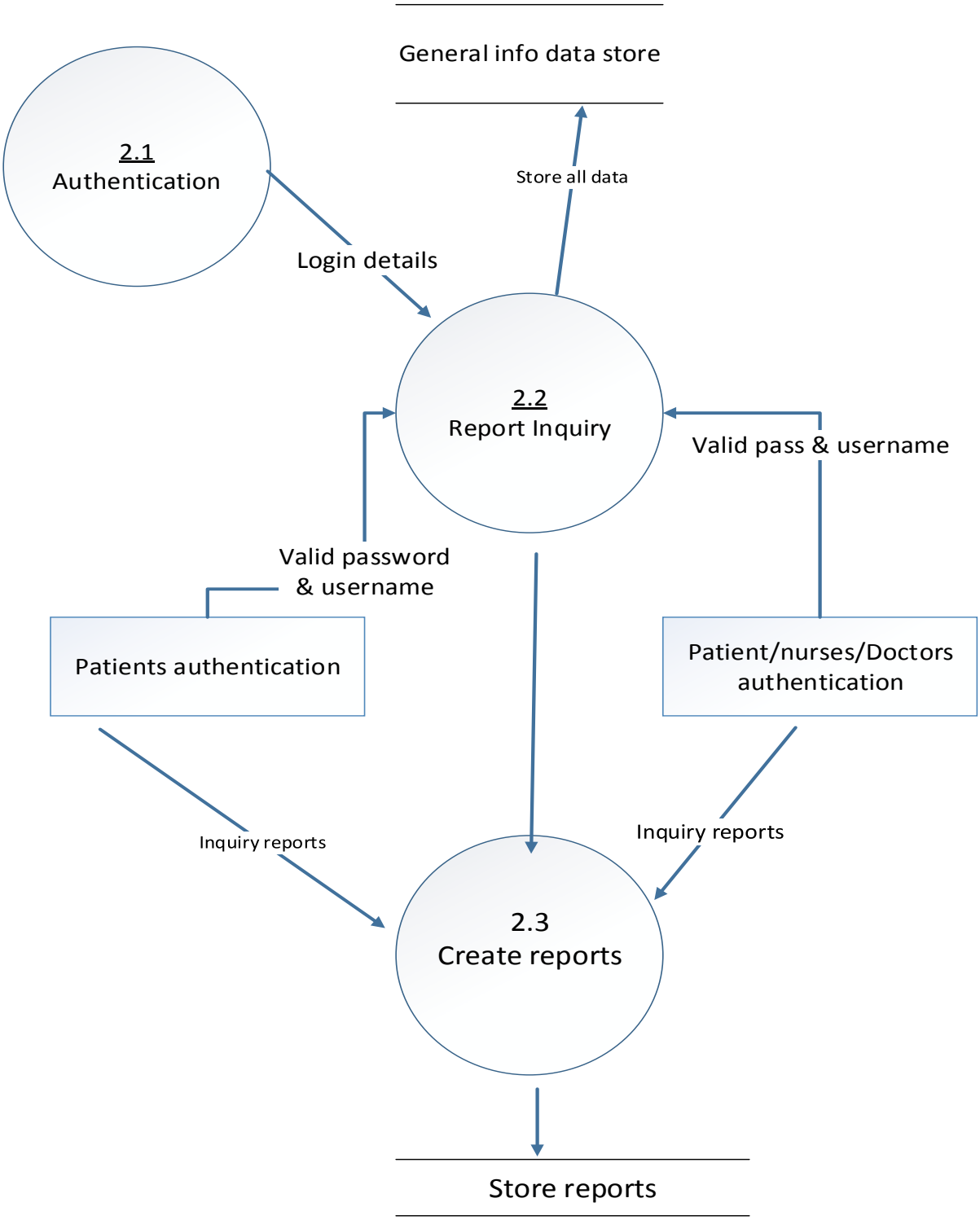
0-level Diagram



Level-1



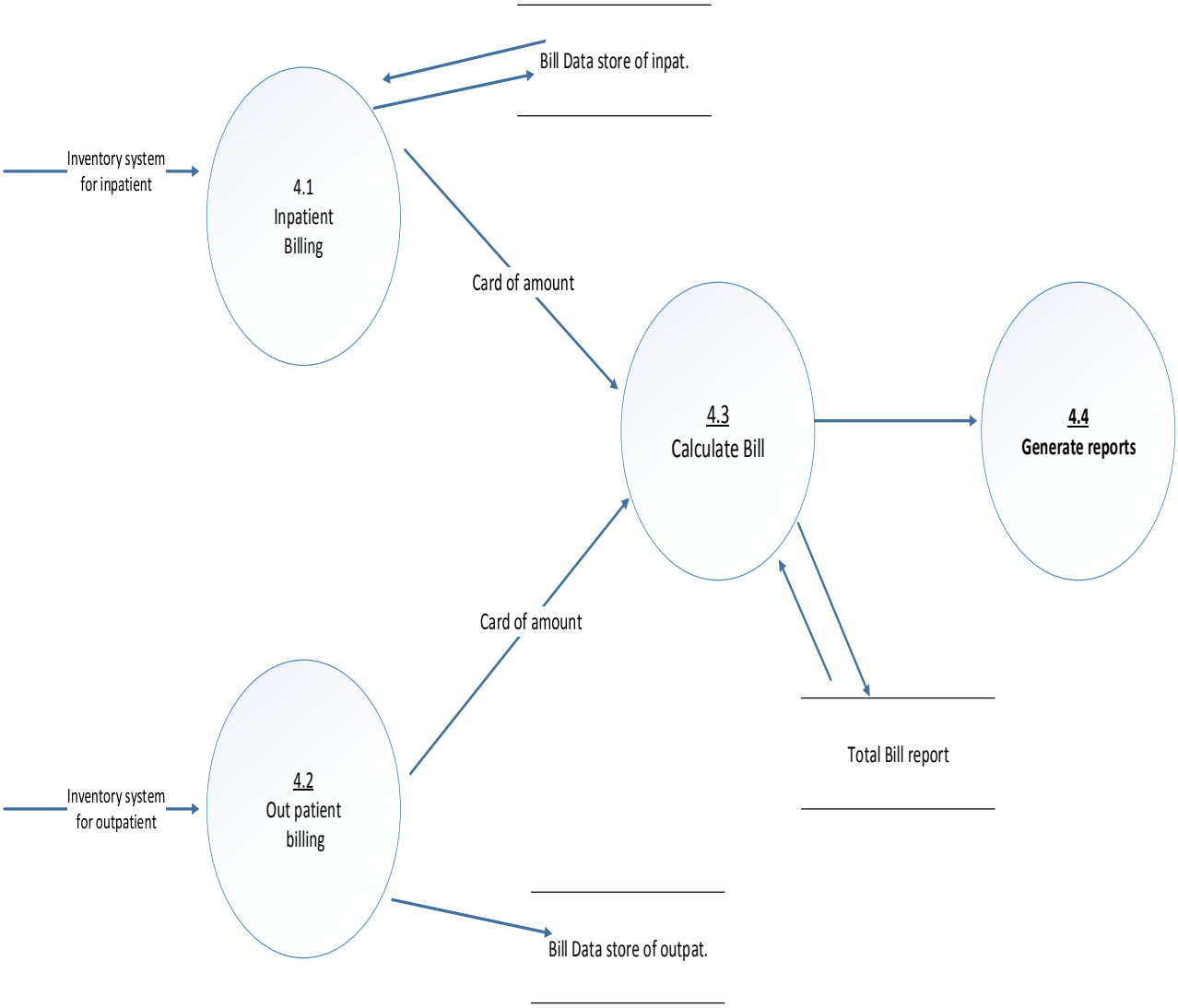
Level 1 cont…


General info data store
2.1
Authentication
Store all data
Login details
2.2
Report Inquiry
Valid pass & username
Valid password
& username
Patients authentication
Patient/nurses/Doctors
authentication
Inquiry reports
Inquiry reports
2.3
Create reports
Store reports

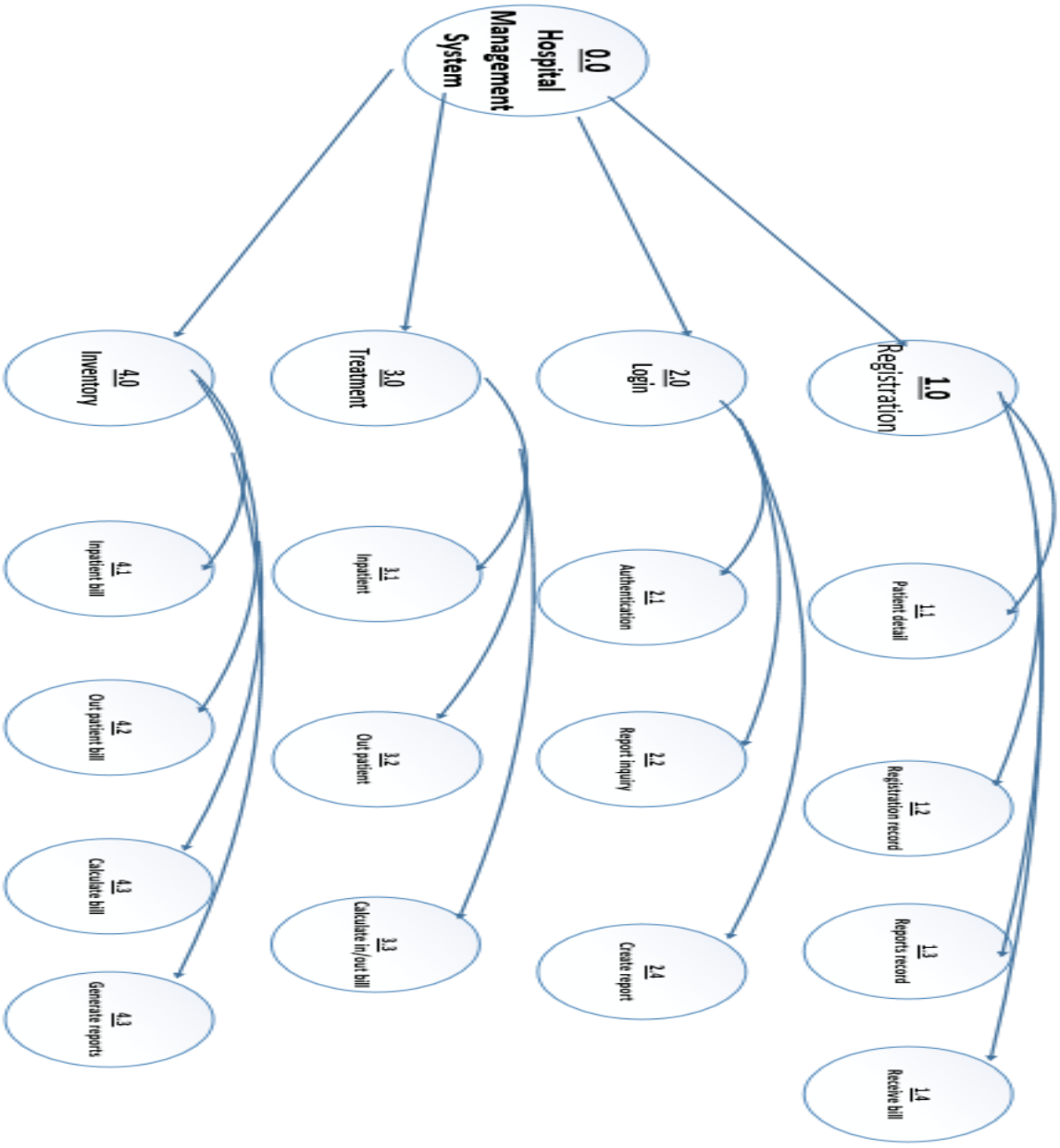
Level 1 cont….


3.1
Inpatients
Treatments
Admitted
patient
Inpatient data
store
3.3
Calculation in/out bill
3.2
Out patients
Treatments
Discharge
Patient
Inpatient data
store

Level 1 cont….


Bill Data store of inpat.
Inventory system for inpatient
4.1 Inpatient Billing
Card of amount
4.3 Calculate Bill
4.4 Generate reports
Card of amount
Total Bill report
Inventory system for outpatient
4.2 Out patient billing
Bill Data store of outpat.

Process Decomposition



Chapter 5

Implementation

Chapter 5: Implementation

RUP methodology use for the Implementation of this system. We get all the necessary information of all the development phases including inception, Elaboration, construction, construction and transition. In implementation we develop the platform and we check either the system fulfills the requirements.

Inception:	The idea for the project is stated. The development team determines if the project is worth pursuing and what resources will be needed.
Elaboration:	The project's architecture and required resources are further evaluated. Developers consider possible applications of the software and costs associated with the development.
Construction:	The project is developed and completed. The software is **designed**, **written**, and **tested**.
Transition:	The software is released to the public. Final adjustments or updates are made based on feedback from end users.

5.1. Important Flow Control/Pseudo codes

Pseudo code:

Login page front page HMS:

Start system

Open Login front page

User name and password

Database connect

Match with database user name and password

If (yes then okay)

Else(try again)

Add Patient after login:

If HMS Admin Login

then

Fill all detail of patient required

If all done

then

Registered patient successfully

Add record reports/test:

If patient add successfully

Open record current patient

If exits

then

Save reports result

Add successfully reports result

Editing record patient:

Click Edit button

Check information from database

If correct (if exit record)

then (okay)

Else

Delete * form table

If id exit

Else Message error ‘not exist’

Delete

Deleted information patient:

Check detail about patient

If correct

Then okay

Else If (any detail is incorrect)

then

Delete data from table

Message:”Deleted success full record”

Medicine add record:

If patient need medicine

Add medicine record

Else (ignore)

No need

Bill add info:

Check detail patient medicine, other expense

then

Generate Bill according to add price

If no receive bill

then

Status “Unpaid bill “

Else

Status “paid”

Room add record:

Check room information

If patient need room

Optional

1- Private

2- General

then Add room

"Generate expense"

5.2. Components, Libraries, Web Services and stubs

Languages and Libraries:

PHP, JavaScript, HTML, CSS, Bootstrap

Web services:

Rest API

5.3. Deployment Environment

Browser:

- ✧ Uc browser
- ✧ Chrome

Server:

- ✧ Bandwidth: 6GB
- ✧ Space: 500MB

5.4. Tools and Techniques

Development tool	Programming Languages	Graphic designing and Prototyping	Report Writing & Documentation
✓ XAMP	✓ PHP	✓ Vision software	✓ Ms Word

✓ Dream-weaver	✓ Java ✓ Bootstrap ✓ HTML		

5.5. Best Practices / Coding Standards

- ✓ Files only <? php and <? tags.
- ✓ Files only UTF-8 without PHP code.
- ✓ File Should either declare symbols (classes, functions, constants, etc.)
- ✓ Name-spaces and classes follow an “auto-loading”.
- ✓ Class names declared in Studly Caps.
- ✓ Class constants declared in all upper case with underscore separators.
- ✓ Method names declared in camel case.

5.6. Version Control

- ✓ GIT

Chapter 6

Testing and Evaluation

Chapter 6: Testing and Evaluation

This Unit/Chapter test cases our HMS system to check the System work properly in various feature of our system.

6.1. Use Case Testing

6.1.1 Login Testing :

Test id	T001
Description	Verify that user is able to login with valid username and password.
Expected Result	It should be successful login.
Prerequisites	User should be Registered first
Actual Result	If we enter invalid pass or name must showing warning message.
Test Data	Username and password
Test Step	Valid username for doctor: amna Invalid username for doctor: aiman Valid Password:123 Invalid password: 124 Valid username for administrator: admin Invalid username for doctor: admiin Valid Password: admin Invalid password: amdmiin
Status	Pass
Created By	Amna
Date	5/05/2019

Executed By	Amna
Comment	No comments
Environment	Window 8 Browser Chrome

6.1.2 Logout Testing:

Test id	T002
Description	Verify that the logout screen
Prerequisites	User should be login
Expected Result	Dashboard of system must be logout
Actual Result	Logged out
Test Data	Logout field
Status	Pass
Created By	Amna
Date	5/5/2019
Executed By	Amna
Comment	No comments
Environment	Window 8 Browser: Chrome

6.1.3 Register Patient Testing:

Test Case id	TC003
Test Suit id	TS003
Description	To Verify when click on register button username, Password.
Prerequisites	-
Expected Result	If username, password, email valid then store user data by submit register button.
Actual Result	If name is valid the result is as expected

	If name is not valid then invalid message display If field empty then message display
Test Step	Fill field is Registered in form Fill valid name field Fill valid CNIC Fill valid Phone no Fill valid room detail (Private/gernal room) Enter Valid data in all field Submit Register Button Fill invalid name field Fill invalid CNIC Fill invalid Phone no Fill invalid room detail (Private/gernal room) Enter invalid data in all field
Status	Pass
Date	5/5/2019
Created By	Amna
Executed By	Amna
Comment	No comments
Environment	Window 8 Browser: Chrome

6.1.4 Doctor/Patient Register Testing:

Test Case id	TC004
Test Suit id	TS004
Description	To verify that delete a data is also add clicking Registered button.
Expected Result	Doctor will successfully registered with all field data.
Actual Result	Doctor should successfully registered with all field data.
Data steps	Name field fill. Address field fill. Phone no Field fill. Type field fill. Submit Registered button.

Status	Pass
Created By	Amna
Date	5/11/2018
Executed By	Amna
Comment	No comments
Environment	Window 8 Browser: Chrome

6.1.5 Update field Testing:

Test Case id	TC005
Test Suit id	TS005
Description	Verify that new details of new doctors can be update in the system.
Expected Result	Update Button working successfully. If enter data valid then update data by clicking update button. If field empty then display message. If given data are valid then display error message.
Actual Result	All add new record must be Update.
Test Step	All data fill then for update then update Click on update button in update field.
Status	Pass
Created By	Amna
Date	5/5/2019
Executed By	Amna
Comment	No comments
Environment	Window 8 Browser: Chrome

6.1.6 Delete Field Testing:

Test Case id	TC006
Test Suit id	TS006
Description	Verify that new details of new doctors can be Delete in the system.
Expected Result	Update Button working successfully. If enter data valid then Delete data by clicking update button. If field empty then display message. If given data are valid then display error message.
Actual Result	All add exist record must be Delete.
Test Step	All data fill then for Delete the Delete Click on update button in D field.
Status	Pass
Created By	Amna
Date	5/5/2019
Executed By	Amna
Comment	No comments
Environment	Window 8 Browser: Chrome

6.1.7 Search Field Testing:

Test Case id	TC007
Test Suit id	TS007
Description	Verify that details of patient through CNIC.
Expected Result	Data of Patients
Actual Result	Get Detail of Patient Record
Test Step	Field fill of CNIC of Patient And then click on CNIC

Status	Pass
Created By	Amna
Date	5/5/2019
Executed By	Amna
Comment	No comments
Environment	Window 8 Browser: Chrome

6.1.8 List Field Testing:

Test Case id	TC008
Test Suit id	TS008
Description	Verify that details of patient through id and date of patient.
Expected Result	Complete data of Patients.
Actual Result	Get Complete Detail of Patient Record
Test Step	Click on ID of patient with date
Status	Pass
Created By	Amna
Date	5/5/2019
Executed By	Amna
Comment	No comments
Environment	Window 8 Browser: Chrome

6.1.9 Room Testing:

Test Case id	TC009
Test Suit id	TS009

Description	Verify that detail of Patient assign room.
Expected Result	Data of room which is assign patient
Actual Result	Detail rooms assign Patient Record
Test Step	Field fill room id Field fill patient id Field fill Floor id(1^{st} , 2^{nd} , ground floor)
Status	Pass
Created By	Amna
Date	5/5/2019
Executed By	Amna
Comment	No comments
Environment	Window 8 Browser: Chrome

6.1.10 Dashboard Refresh Testing:

Test Case id	TC010
Test Suit id	TS010
Description	Verify dashboard information
Expected Result	Total indoor patient, Total doctor, outdoor patient, total earning complete information view
Actual Result	Total All information HMS
Test Step	After fill all information of system Refresh information
Status	Pass
Created By	Amna
Date	5/5/2019
Executed By	Amna

Comment	No comments
Environment	Window 8 Browser: Chrome

6.2. Equivalence partitioning

1. User name is alphabetic. (valid)
2. User name is not alphabetic. (invalid)
3. Password is 2 to 15 characters in length. (invalid)
4. Password include numeric value. (valid)
5. Password include '_'. (invalid)
6. Phone# only be number. (valid)
7. Phone# with alphabetic. (invalid)
8. City name is alphabetic. (valid)
9. City name with numeric values. (invalid)
10. Price only be numeric. (valid)
11. Price is alphabetic. (invalid)
17. Empty field (invalid)

6.3. Boundary value analysis

Sr.		**Partition 1**	**Partition 2**	**Partition 3**
1	Password	Less than 8 character	1 – 8 character	9 – 12
2	Price	Less than 0	Greater than 1	Less than
3	Email	<=0	1 - 11	>=12

6.4. Data flow testing

Relationship between one entity and another while performing a particular task in throughout information flow. Like between the sign in and registration method etc.

6.5. Unit testing

In unit testing we've got testified our totally different panel codes singly by performing arts different tests and by corporal punishment them singly, severally on totally different computers and that they were with success execute and that they performed well.

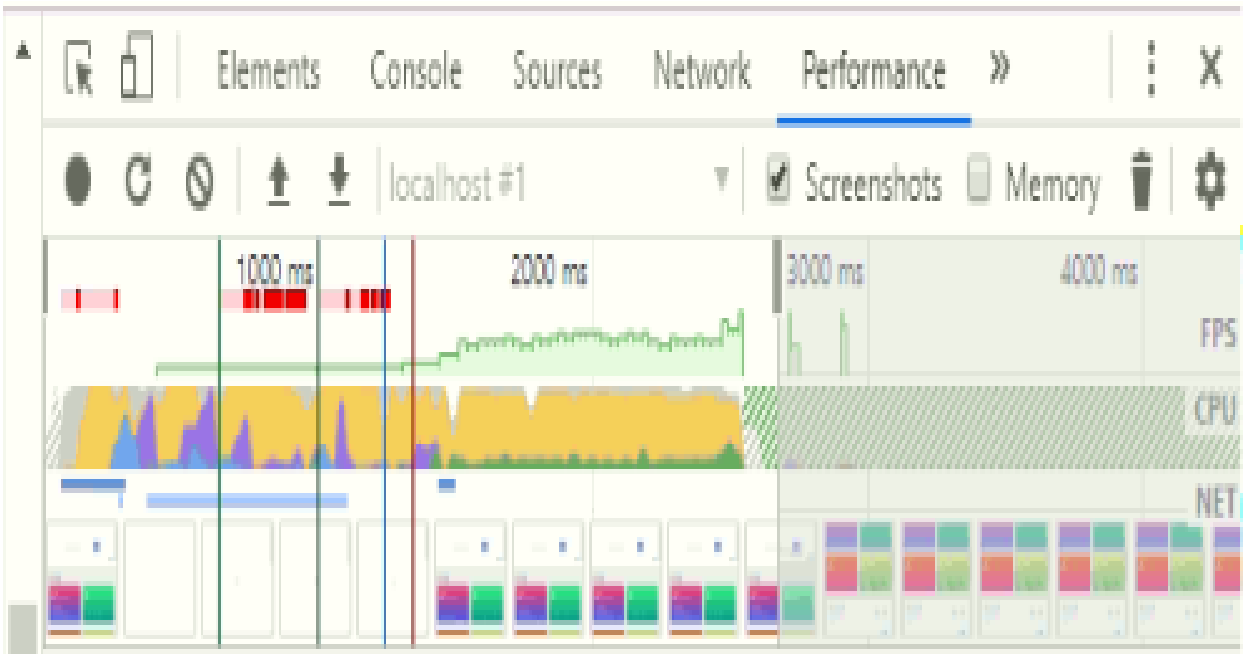
6.6. Integration testing

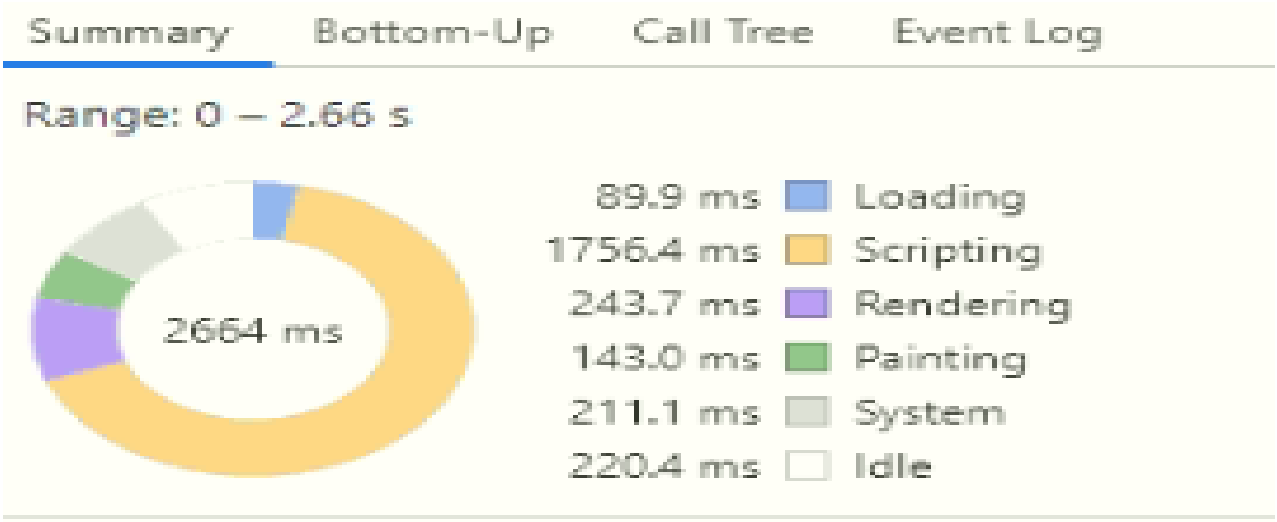
Data may be lost across Associate in interface, one module will have adverse afford on another, sub functions, once combined, might not manufacture the required major functions. Integration testing could be a systematic testing for constructing the program structure whereas at an equivalent time conducting tests to uncover errors related to within the interface. In integration testing step, all errors uncovered area unit corrected for future step.

Test no	Objective	Description	Result
1	Check link between the Login and Dashboard.	Enter username and password , click on the Login button	To be directed to the Dashboard.
2	Check the link between the Dashboard and View Patient registered and information.	Click on Registered button.	To be directed to the View Applied inserted and view patient details.
3	Link between Register and list view	Click on register button	All data will display with fetch and all data show in Doctor field
4	Check the link between the Dashboard and delete data.	Click on Delete data button.	To be directed to deleting and homepage open of the website.

5	Check the link between the Dashboard and update data.	Click on update data and update the options and click on Update button	To be directed to Dashboard with data updated.
6	Check the interface link between the Dashboard and Serach Bar.	Click on search button with the CNIC of patient.	It will open all data of patient through CNIC.
7	Check the interface link between the Login and logged out	Click on logged out	Then dashboard closed

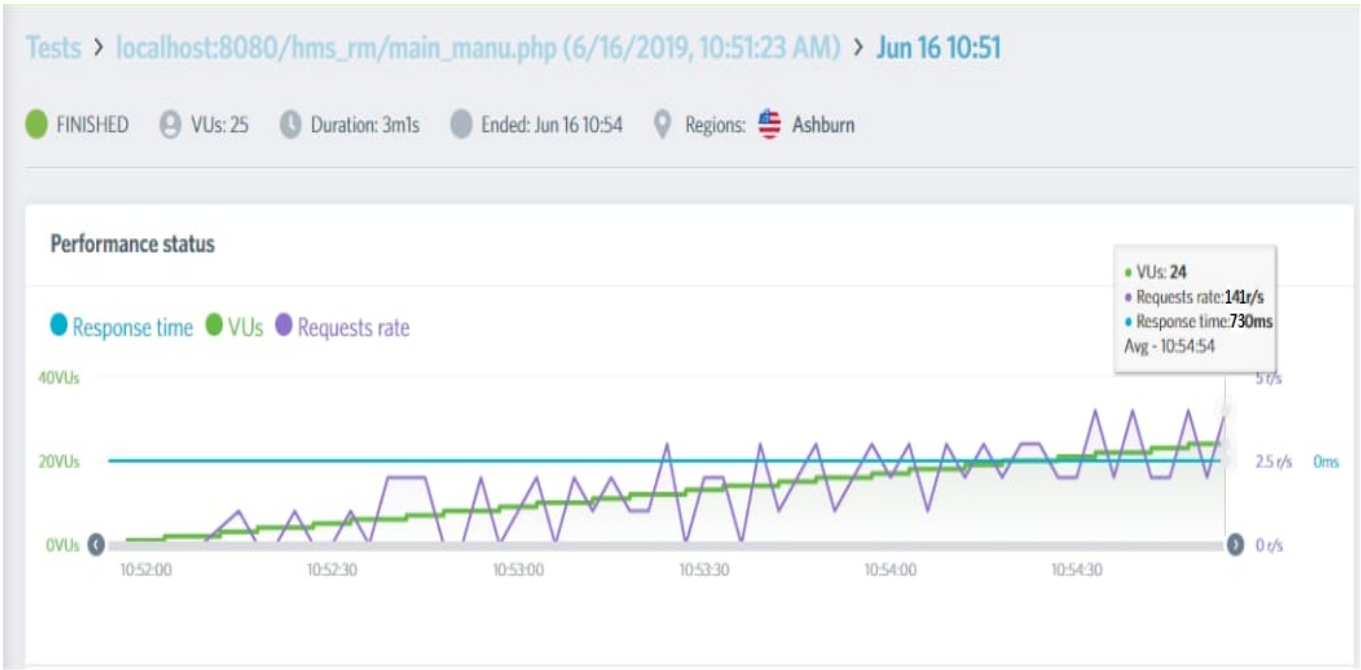
6.7. Performance testing





6.8. Stress Testing

Stress testing is done by 25 virtual users as number of user increase request rate and respone time of system increase.



Chapter 7

Summary, Conclusion and Future Enhancements

Chapter 7: Summary, Conclusion & Future Enhancements

7.1. Project Summary

This is a system that includes varied factors required for sleek operating and management of a hospital. This hospital management system permits administrator to store and manage varied merchandise and connected information. This includes the beds/normal accommodation as patient information. just in case a replacement patient arrives, the administrator could check for bed accessibility in varied hospital departments and allot it consequently as per the patient want. The system additionally stores the patient information during a well outlined manner. The patient details embody his name, number, address and varied health parameters. The software uses HTML, CSS, Java Script , Bootstrap as a forepart to style associate degree interactive GUI associate deg-reed an SQL server to store its side information. This side information includes accommodation details and even connected patient information for effectively managing a hospital.

7.2. Achievements and Improvements

The biggest achievements here are that we tend to we tend tore ready to compile all we learned in four years of finding out and apply it to the current project. We tend to learned computer code design style techniques, UML modeling, project management, testing and far additional, and were ready to apply it bushed this project. Ensuing massive action is that the things we tend to learn throughout this project. New languages, frameworks, libraries, totally different software's for diagrams, work with SQL information.

7.3. Critical Review

The crucial component of our system are: is that the developed system fast-paced, provide user friendly interface, not terribly expensive, fulfill the precise needs of the users at a run-time, has all the options that any institute will need. To manage of these aspects, it took heap of your time and exertions.

7.4. Lessons Learnt

This project sharpens our skills in PHP with SQL, and plenty of different tools and plenty of management ideas yet as the way to take care of drawback retardant and the way to stay for locating the answer of any problem till you fond. yet as technical skills this project conjointly enhances our personal development skills like team operating, dedication.

7.5. Future Enhancements/Recommendations

As it has been already same, there's invariably space for more improvement. And since we have a tendency to-conceive to launch this web site as our own start-up, enhancements can keep coming back. The system has been designed on react native, however at some purpose we have a tendency to work on its potency, accessibility, flexibility, we have a tendency to are positively reaching to scale it up that we have a tendency to believe ours is eventually reaching to become. we have a tendency to create it additional user friendly interface.

Reference and Bibliography

Reference and Bibliography

https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database

https://www.google.com/search?q=srs+hospital+management+system&oq=srs+hosp&aqs=chrome.2.69i57j0l5.3242j0j7&sourceid=chrome&ie=UTF-8

https://www.docsity.com/en/software-requirement-specification-for-hospital-management-system-computer-sciences-project-report/86558/

https://www.slideshare.net/kataria55/srs-for-hospital-management-system

https://medium.com/@datamateuaecrescent/hospital-management-system-features-objectives-62eeb13f4fc4