



# SYSTEM DESIGN DOCUMENT

HOSPITAL MANAGEMENT SYSTEM

Roshani Thapa  
Nirjala Shrestha  
Leah K.C.  
Parash Neupane

## **Abstract**

The Hospital Management System (HMS) is a computerized system designed to manage the day-to-day operations and administrative tasks of a hospital. This system provides a platform for efficient and effective management of patient data, medical records, appointment scheduling, and inventory control. It also includes features for managing hospital finances, billing, and insurance claims processing. The HMS ensures seamless coordination between different departments, such as laboratory, pharmacy, and radiology, and facilitates communication among doctors, nurses, and other hospital staff. With the help of this system, hospital administrators can make informed decisions based on real-time data and analytics, improving patient outcomes and optimizing resource utilization.

## **Acknowledgement**

I would like to express my heartfelt gratitude to all those who have contributed to the successful completion of this Hospital Management System project. Firstly, I extend my sincere thanks to the hospital management and administration for providing me with the opportunity to work on this project. I also want to thank the doctors, nurses, and other medical staff who provided valuable insights and suggestions that helped shape the system. I am grateful to my colleagues who supported and encouraged me throughout the development process, as well as my family and friends who provided me with unwavering support and motivation. Lastly, I would like to acknowledge the open-source technologies and libraries that were used in building this system. Thank you all for your contributions and support towards the successful completion of this Hospital Management System project.

## Table of Contents

<b>Abstract .....</b>	<b>i</b>
<b>Acknowledgement.....</b>	<b>ii</b>
<b>List of figures .....</b>	<b>iv</b>
<b>1. Introduction .....</b>	<b>1</b>
1.1 Purpose:.....	1
1.2 Project Scope:.....	1
1.3 Document Conventions .....	1
1.4 Intended audience.....	1
1.5 Additional Information: .....	2
<b>2. Overall Description.....</b>	<b>3</b>
2.1 Product Perspective:.....	3
2.2 Product Features:.....	3
2.3 User Classes and Characteristics:.....	4
2.4 Operating Environment:.....	4
2.5 Design and Implementation Constraints: .....	4
2.6 Assumptions and Constraints .....	5
<b>3. Architecture .....</b>	<b>6</b>
3.1 Overview .....	6
3.2 Modules.....	7
<b>4. External Interface Requirements: .....</b>	<b>8</b>
4.1 User Interfaces .....	8
4.2 Hardware Interfaces .....	8
4.3 Software Interfaces.....	8
<b>5.Other Requirements .....</b>	<b>9</b>
<b>6. Non Functional Requirements.....</b>	<b>20</b>
6.1 Performance Requirements .....	20
6.2 Safety Requirements .....	20
6.3 Security Requirements .....	20
6.4 Software Quality Attributes .....	20
6.5 Business Rules .....	21
6.6 Gantt Chart .....	21
<b>References.....</b>	<b>22</b>

## List of figures

Figure 1:Architecture .....	7
Figure 2: class diagram.....	9
Figure 3:Use Case Diagram .....	10
Figure 4:ER diagram .....	11
Figure 5: Context Diagram .....	12
Figure 6: Level 1 DFD.....	12
Figure 7: Level 2 DFD.....	13
Figure 8: Sequence Diagram for Doctor Module .....	14
Figure 9: Patient Module Sequence diagram.....	15
Figure 10: Admin module.....	16
Figure 11: Activity Diagram for registration.....	17
Figure 12: Activity Diagram for patient Module .....	18
Figure 13: Activity Diagram for doctor module .....	19
Figure 14: Gnatt Chart .....	21

# **1. Introduction**

My project Hospital Management system includes registration of patients, storing their disease details into the system. My software has the facility to give a unique id for every patient and stores the details of every patient. The Hospital Management System can be used by entering respective username and password. It is accessible either by an administrator or receptionist. Only the respective person can add data in the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected and data processing is very fast, accurate and relevant. . Hospital Management System is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals

## **1.1 Purpose:**

A hospital management system is a software designed to manage all the areas of a hospital such as medical, financial, administrative and the corresponding processing of services.

## **1.2 Project Scope:**

Daily functions like patient registration, managing admission and overall management of various departments can be easily performed with higher accuracy after the installation of hospital software. The modules of hospital management software are user-friendly and easy to access.

## **1.3 Document Conventions**

The document is prepared using Microsoft Word 2010 and has used the font type 'Times New Roman'. The fixed font size that has been used to type this document is 14pt and for headings 18pt with 1.5 linespacing. It has used the bold property to set the headings of the document. All pages except the cover page are numbered, the numbers appear on the lower right-hand corner of the page. Every image and data table are numbered and referred to the in the main text.

## **1.4 Intended audience**

The intended audience of this document would be the client and specific employees like Manager and Receptionist, consultants and System Operators of the St Joseph Hospital, and project team, supervisor with the objective to refer and analyze the information. The SRS document can be used

in any case regarding the requirements of the project and the solutions that have been taken. The document would finally provide a clear idea about the system that is building.

### **1.5 Additional Information:**

Hospital Management System can be used by entering respective username and password. It is accessible either by an administrator or receptionist. Only the respective person can add data in the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected and data processing is very fast, accurate and relevant. A hospital management system is a software designed to manage all the areas of a hospital such as medical, financial, administrative and the corresponding processing of services.

## **2. Overall Description**

A hospital management system is a software designed to manage all the areas of a hospital such as medical, administrative and the corresponding processing of services. HMS is an abbreviation of hospital management system. The hospital management system (HMS) is an integrated software that handles different directions of clinic workflows. It manages the smooth healthcare performance along with administrative, medical, legal, and financial control. That is a cornerstone for the successful operation of the healthcare facility.

### **2.1 Product Perspective:**

This Hospital Patient Management System is a self- contained system that manages activities of the hospital as bed assignment, operations scheduling, personnel management, and administrative issues. Various stakeholders are involved in the hospital system.

### **2.2 Product Features:**

#### **Doctor Module:**

- Add patients report
- Delete patients report
- Display reports
- Give prescriptions
- Search reports of patient

#### **Receptionist Module:**

- Add patient's data
- Delete patient's data
- Display records
- Refer to different doctors
- Search the record of patients

#### **Patient Module:**

The different functionalities of the module are listed below:

- Search reports



- Search his record

### **2.3 User Classes and Characteristics:**

The system will be used in the hospital. The administrators, front-desk staff will be the main users. Given the condition that not all the users are computer-literate. Some users may have to be trained on using the system.

### **2.4 Operating Environment:**

The system is also designed to be user-friendly. The software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.

### **2.5 Design and Implementation Constraints:**

1. Anticipate difficulties and limitations regarding system upgrades and improvements due to the coordination required to stop clinical systems that require continuity of operation.
2. Be able to handle a significant number of transactions at any time.
3. Support a high rate of concurrent electronic transactions as different health professionals may have to enter new information or modify it.
4. Always log all transactions to be able to know what happened, allowing you to replay events, understanding bugs and ensuring the integrity of information.
5. Always ensure the integrity of the information, even in concurrent consultation.
6. Always make information accessible, even in concurrent consultation.
7. Guarantee a speed of data display, no matter how much information to look for in several different databases.

## 2.6 Assumptions and Constraints

### Assumptions:

- The hospital has adequate infrastructure and hardware resources to support the HMS, including servers, network, and storage devices.
- The hospital has trained personnel to operate and manage the HMS.
- The hospital has reliable internet connectivity to access and use cloud-based HMS solutions.
- The hospital has already established its workflows and processes, which can be integrated into the HMS.
- The hospital has an existing database of patient and staff records that can be imported into the HMS.

### Constraints:

- Budget constraints limit the selection of a suitable HMS solution.
- The hospital face regulatory constraints, such as compliance with HIPAA regulations.
- Integration with existing legacy systems pose technical constraints.
- Data security and privacy concerns impose constraints on data access and sharing.
- Language and cultural differences impose constraints on the usability and adoption of the HMS among hospital staff and patients.

## 3. Architecture

### 3.1 Overview

The architecture of a hospital management system (HMS) typically consists of several layers that work together to provide a comprehensive solution for managing the hospital's operations. Here is an overview of the typical architecture of an HMS:

- **Presentation Layer:** This layer is the user interface of the HMS and provides an interactive platform for hospital staff to access the system. The presentation layer can be web-based, mobile-based, or desktop-based and is designed to be user-friendly and intuitive.
- **Application Layer:** This layer contains the core logic and functionality of the HMS. It is responsible for processing user requests, managing data, and communicating with other layers of the system. The application layer can be divided into several modules, including patient management, staff management, billing and insurance, inventory management, laboratory management, radiology management, pharmacy management, and reporting and analytics.
- **Business Logic Layer:** This layer contains the business rules and logic that govern the behavior of the HMS. It is responsible for enforcing security policies, validating user input, and ensuring data integrity. The business logic layer interacts with the application layer to process user requests and generate responses.
- **Data Access Layer:** This layer is responsible for interacting with the database and managing data access. It provides an abstraction layer between the application layer and the database, allowing the application layer to interact with the database without worrying about the underlying data access logic.
- **Database Layer:** This layer contains the hospital's data and is responsible for storing, retrieving, and managing the data. The database can be relational, object-oriented, or document-oriented, depending on the specific requirements of the hospital.

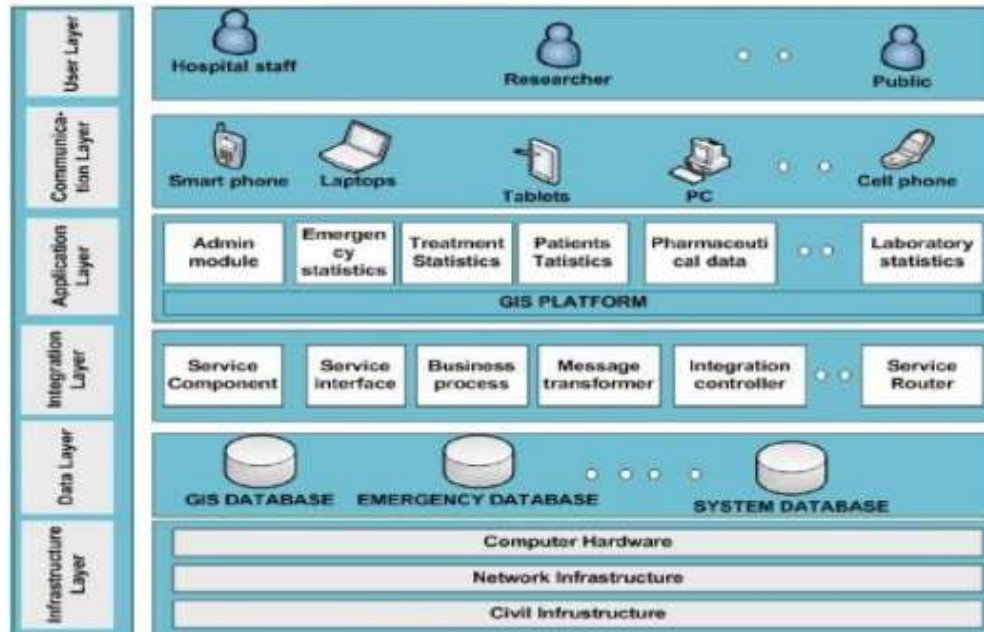


Figure 1:Architecture

## 3.2 Modules

**Patient Management Module:** This module helps manage patient records, including registration, admission, discharge, and transfer. It also provides features for managing appointments, scheduling visits, and generating reports.

**Doctor Management Module:** This module helps manage doctor records, including registration. It also provides features for managing appointments, scheduling visits, and generating reports of doctor.

**Staff Management Module:** This module helps manage hospital staff, including doctors, nurses, technicians, and administrative personnel. It includes features for managing their schedules, work shifts, and payroll.

## **4. External Interface Requirements:**

### **4.1 User Interfaces**

Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.

### **4.2 Hardware Interfaces**

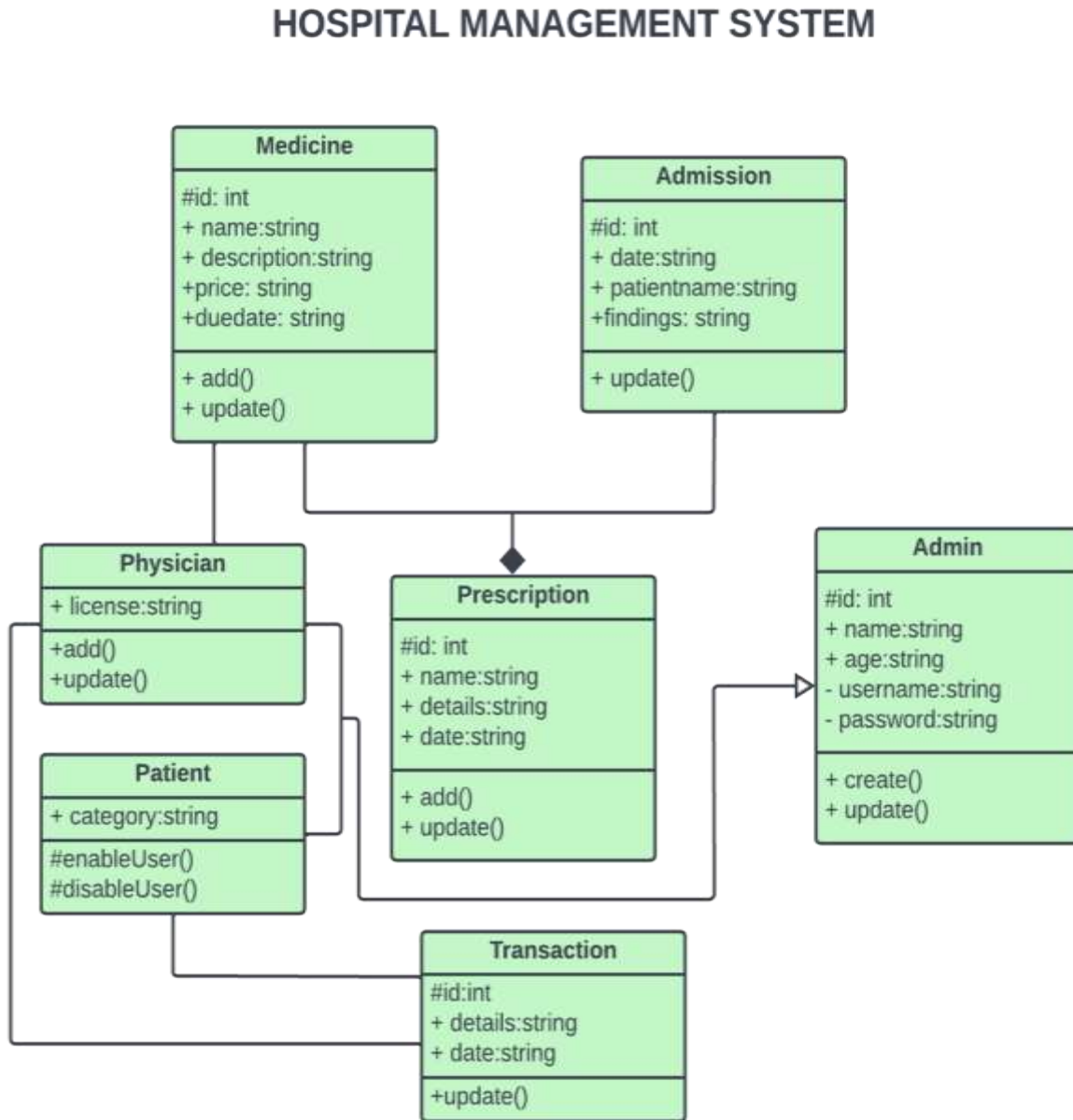
Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.

### **4.3 Software Interfaces**

Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.

## 5.Other Requirements

UML Class Diagram for Hospital Management System:



**UML CLASS DIAGRAM**

Figure 2: class diagram

## Use case Diagram

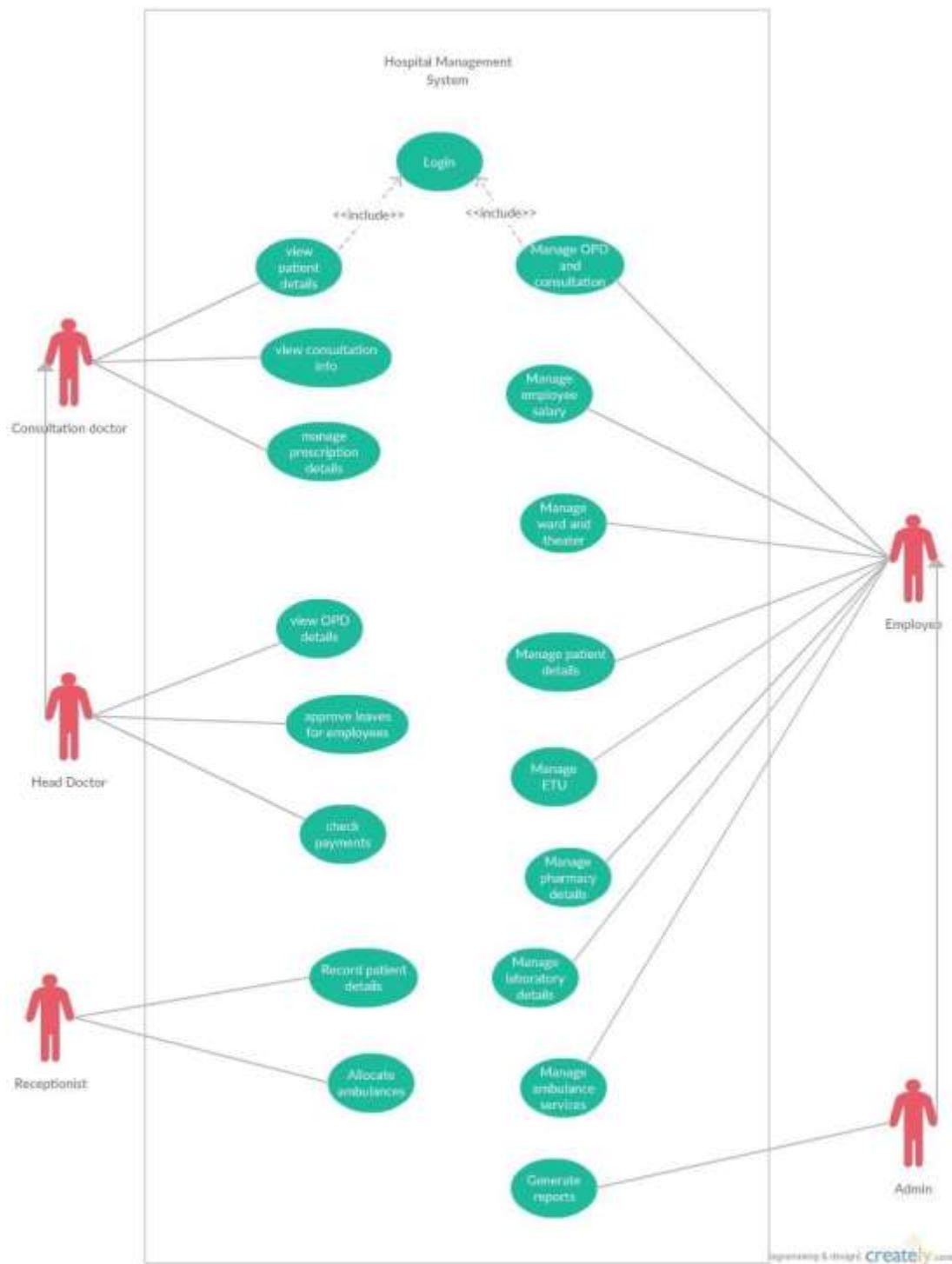


Figure 3: Use Case Diagram

ER diagram:

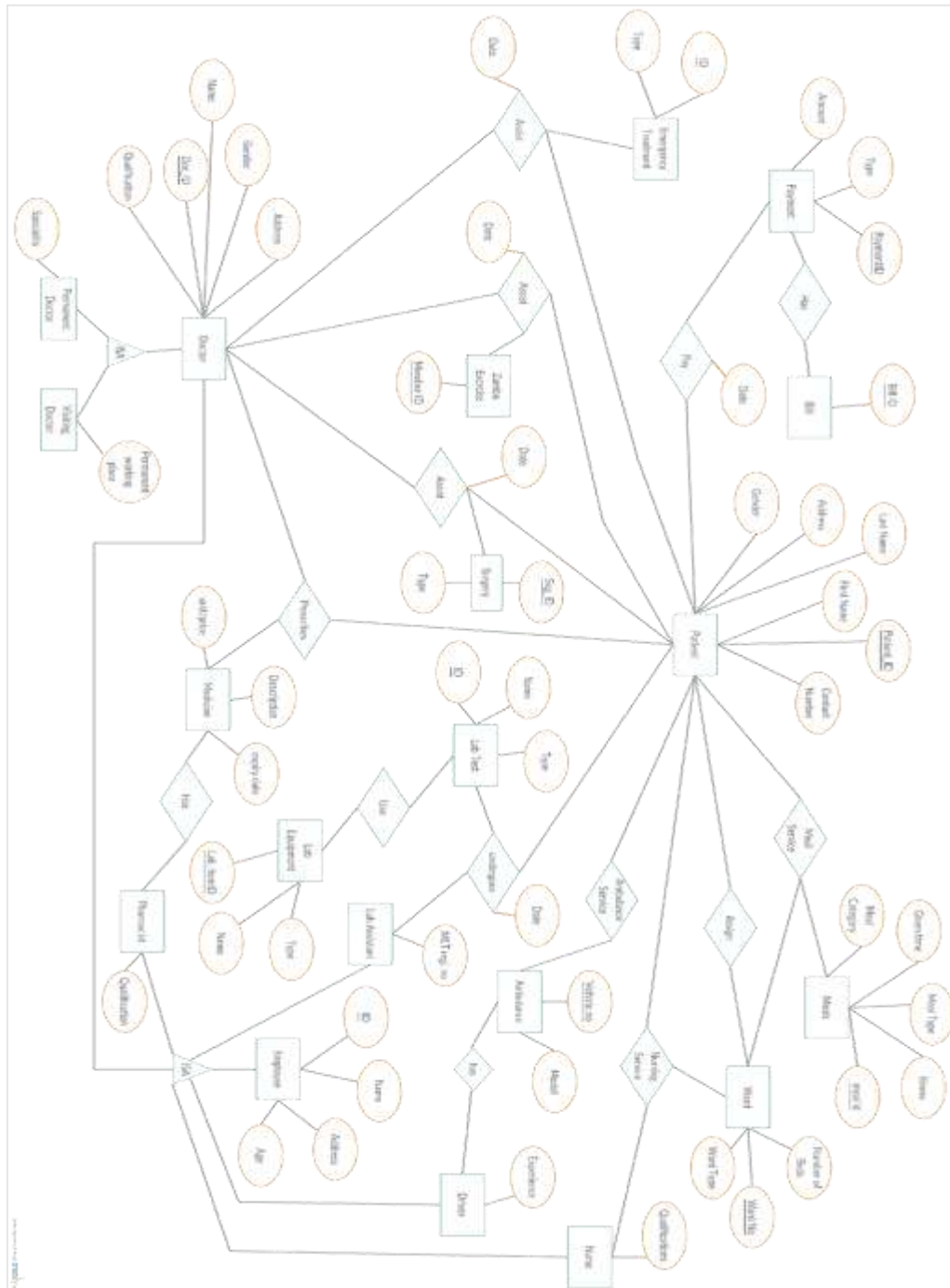


Figure 4:ER diagram



Context Level DFD:

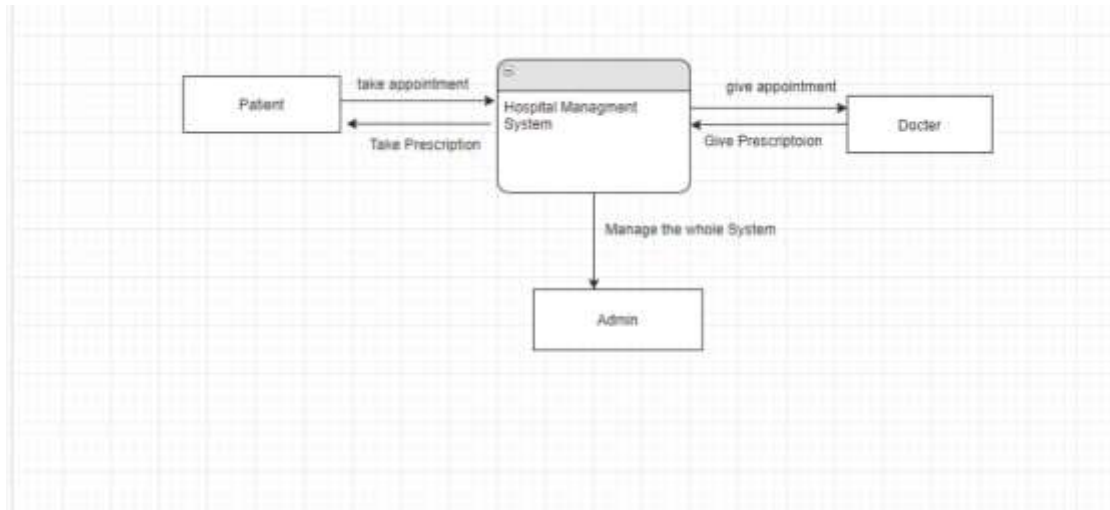


Figure 5: Context Diagram

Level 1 DFD:

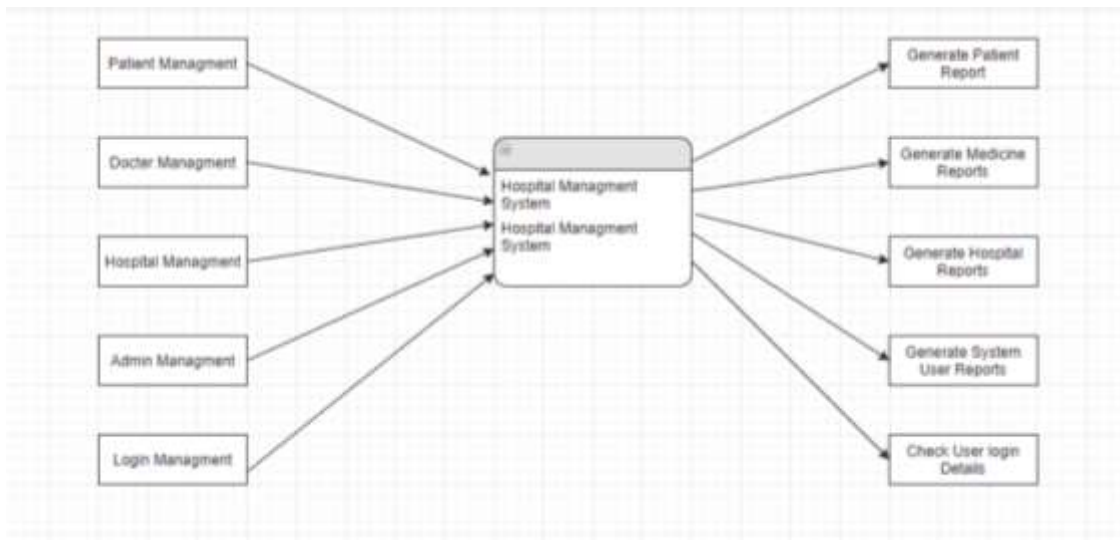


Figure 6: Level 1 DFD

Level 2 DFD:

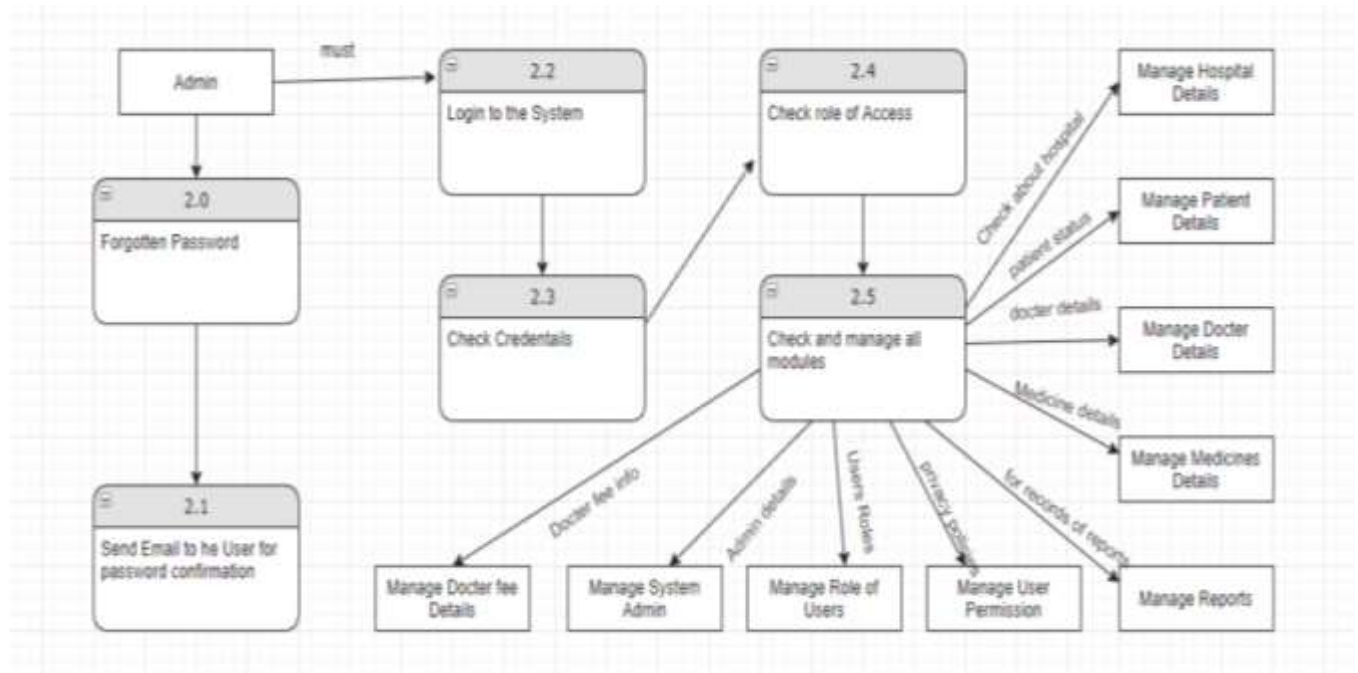


Figure 7: Level 2 DFD

Sequence diagram for Doctor Module:

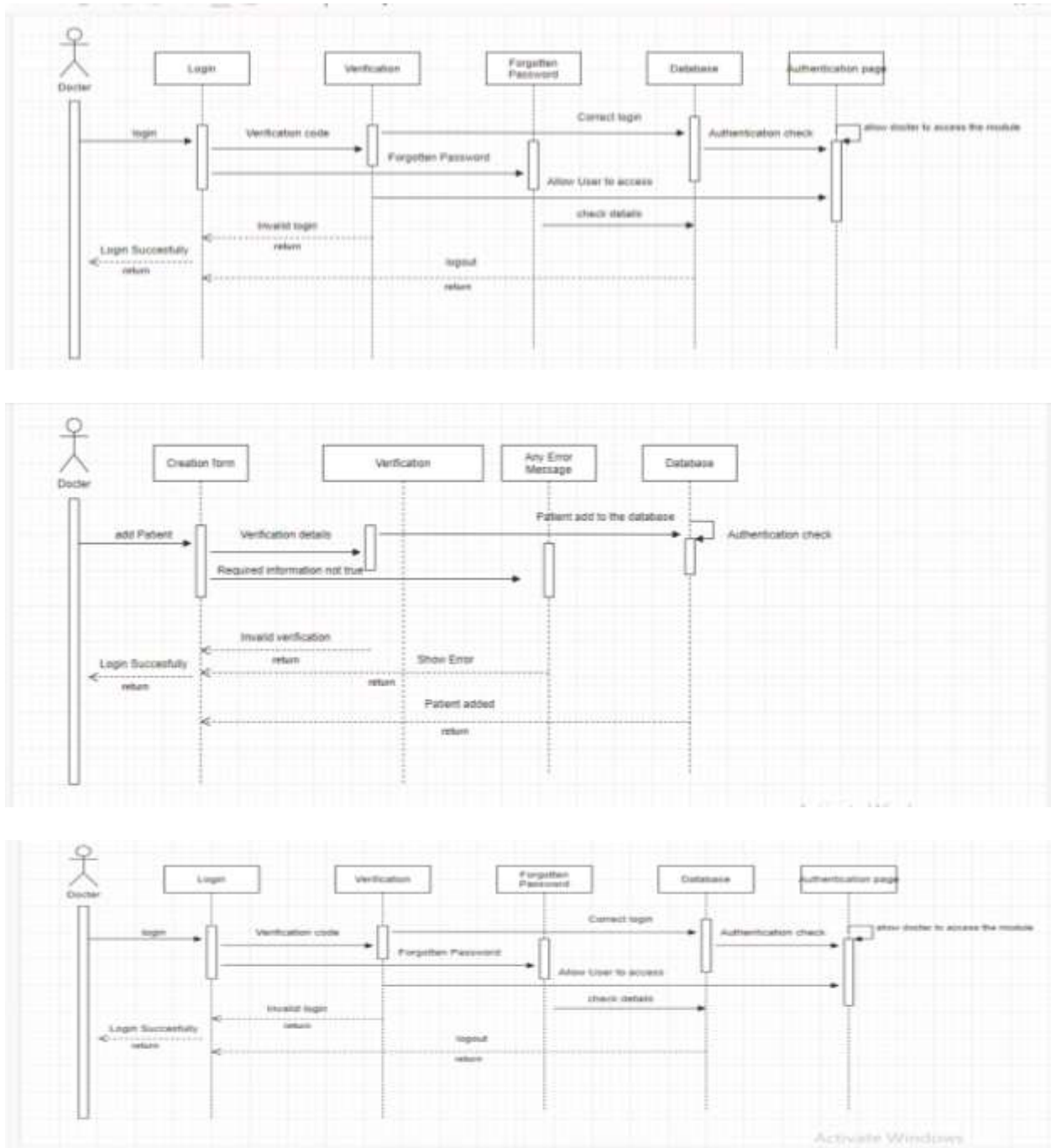
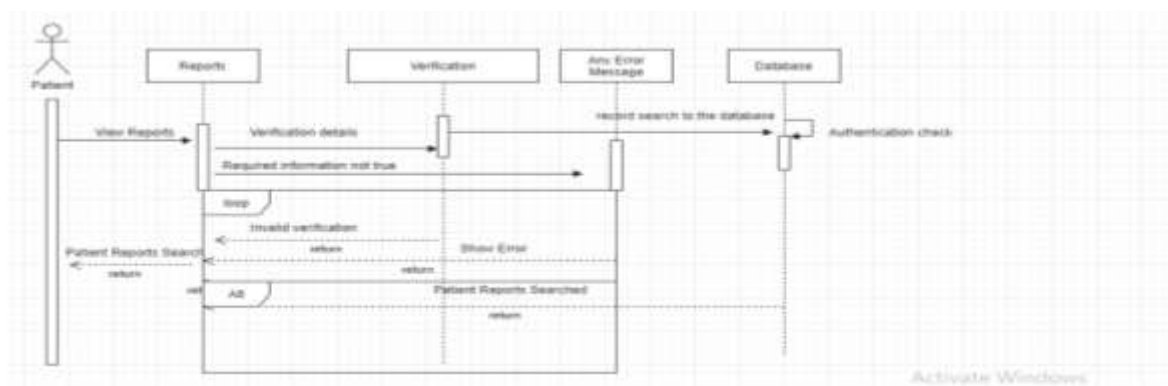


Figure 8: Sequence Diagram for Doctor Module

diagram:



15

Admin module:

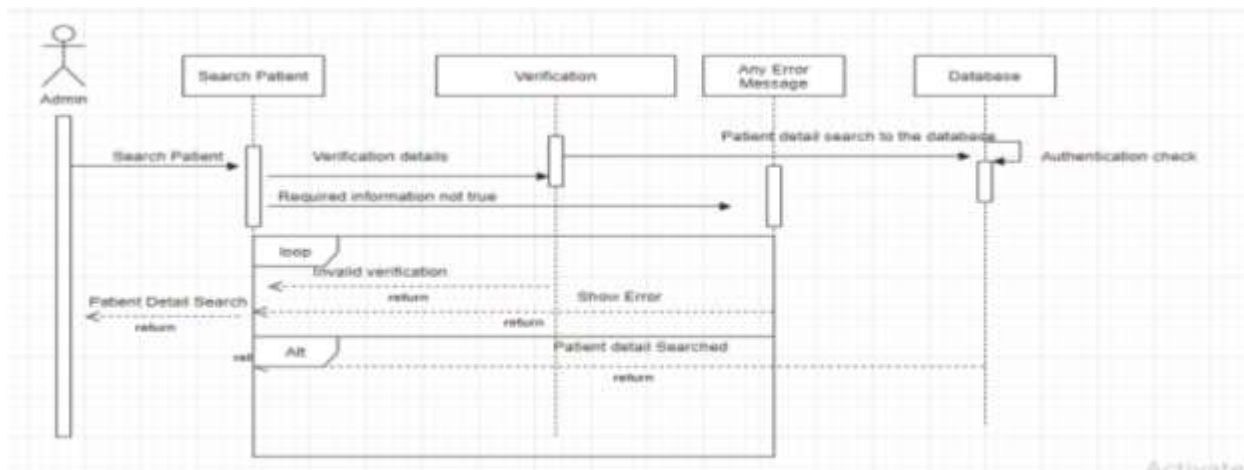
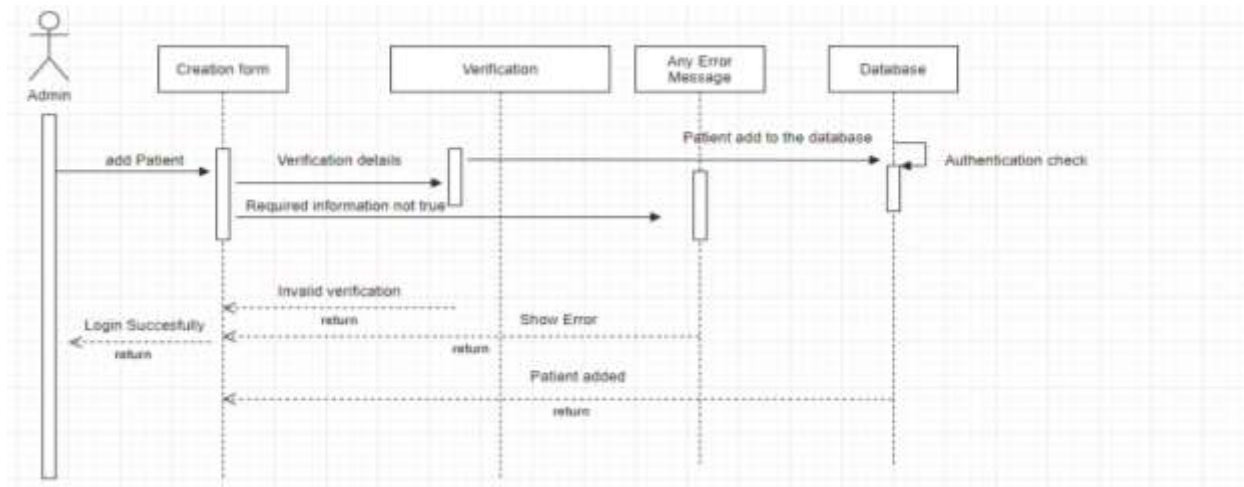
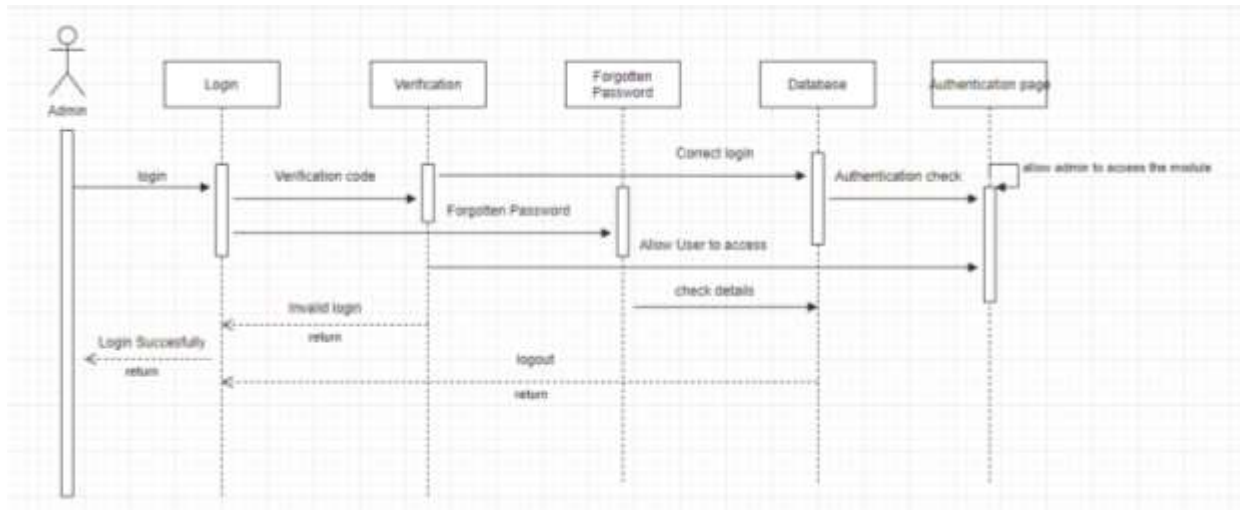


Figure 10: Admin module

## Activity Diagram

For Registration:

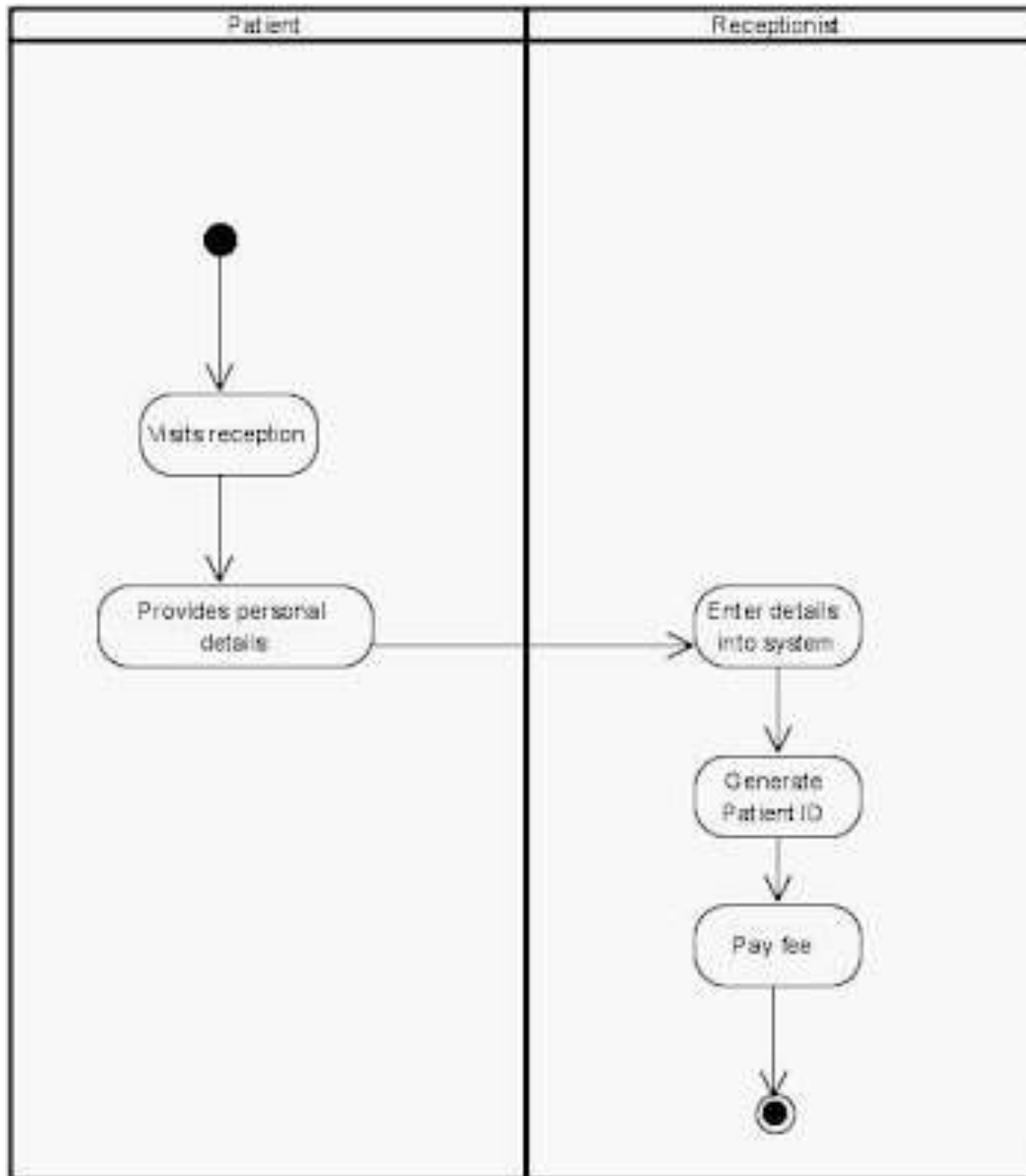


Figure 11: Activity Diagram for registration

Patient module:

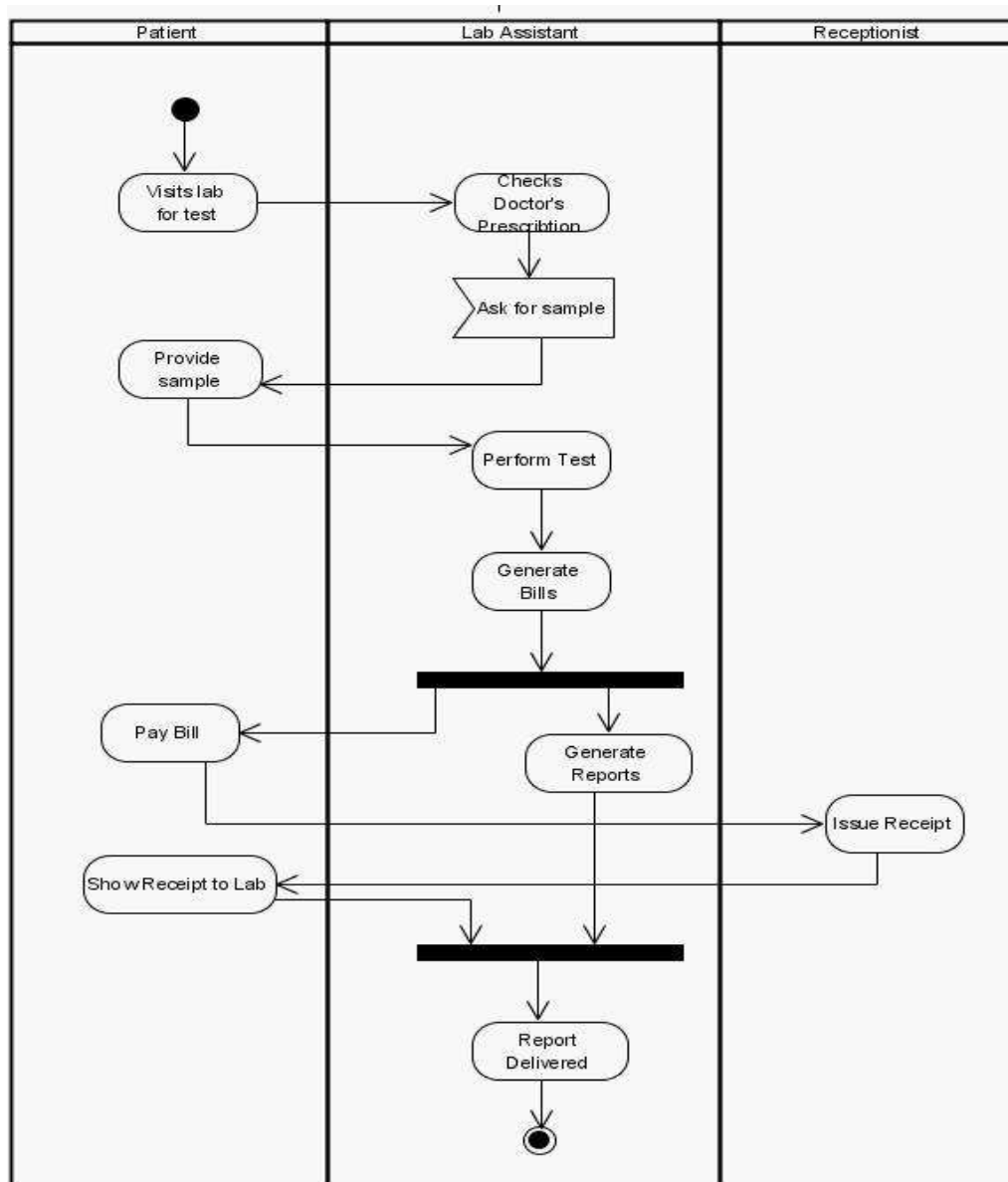


Figure 12: Activity Diagram for patient Module

## Doctor Module:

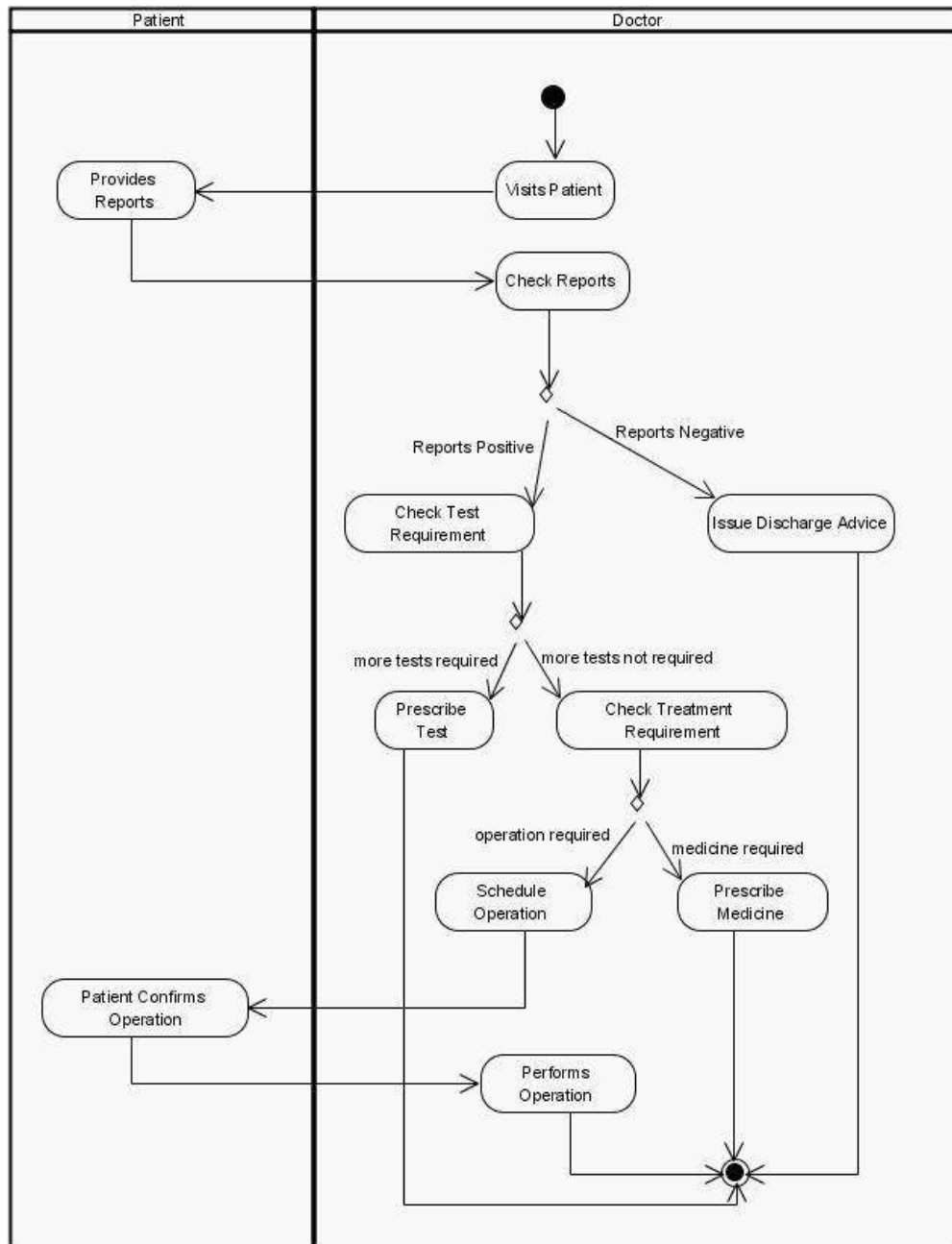


Figure 13: Activity Diagram for doctor module



## **6. Non Functional Requirements**

### **6.1 Performance Requirements**

- Response time-The system will give responses within 1 second after checking the patient information and other information.
- Capacity-The system must support 1000 people at a time
- User interface- User interface screen will response within 5 seconds.
- Conformity –The system must conform to the Microsoft accessibility

### **6.2 Safety Requirements**

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

### **6.3 Security Requirements**

All the administrative and data entry operators have unique logins so system can understand who is login in to system right now [1] [1] [1] no intruders allowed except system administrative nobody cannot change record and valuable data.

### **6.4 Software Quality Attributes**

- **Availability:** The system shall be available all the time.
- **Correctness:** A bug free software which fulfill the correct need/requirements of the client.
- **Maintainability:** The ability to maintain ,modify information and update fix problems of the system

- **Usability:** software can be used again and again without distortion.
- **Accessibility:** Administrator and many other users can access the system but the access level is controlled for each user according to their work scope.
- **Accuracy:** The reliability on the information/output. Can depend/be sure of the outcome.
- **Stability:** The system outcome/output won't change time to time. Same output will be given always for a given output.

## 6.5 Business Rules

- Want take the responsibility of failures due to hardware malfunctioning.
- Warranty period of maintaining the software would be one year.
- Additional payments will be analysed and charged for further maintenance
- If any error occur due to a user's improper use. Warranty will not be allocated to it.
- No money back returns for the software.
- Trust bond placement should be done before designing and coding. An advance or an agreement.

## 6.6 Gantt Chart

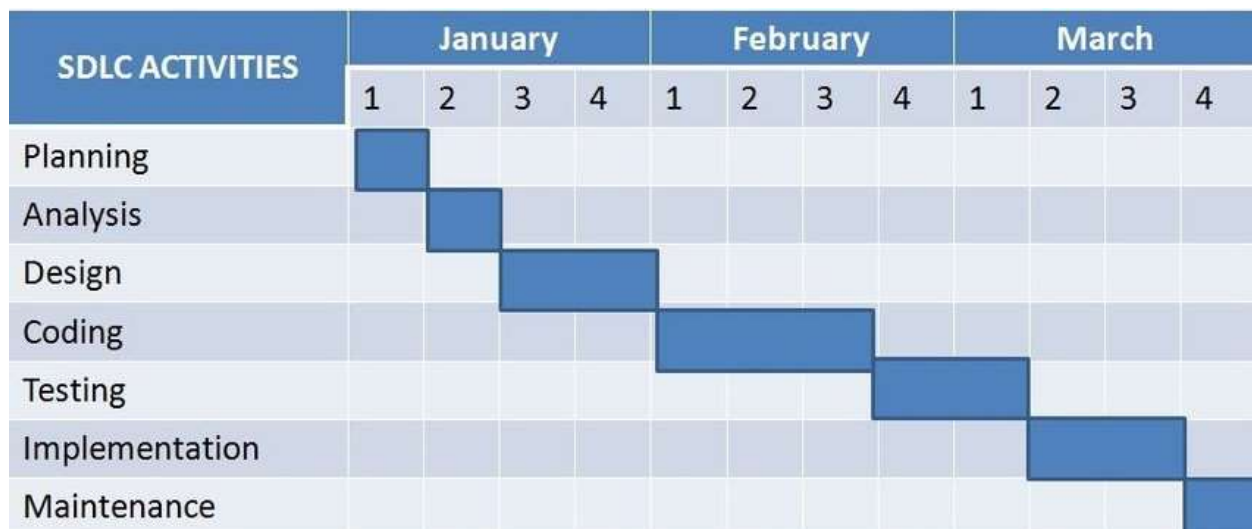


Figure 14: Gantt Chart

## References

- [1] Lauesen, S, (2003), *Task Descriptions as Functional Requirements*,  
IEEE Computer Society. <http://www.itu.dk/~slauesen/Papers/IEEEtasks.pdf>
- [2] <https://www.lucidchart.com>
- [3] <https://www.slideshare.net>
- [4] <https://www.scribd.com/doc>