

SOFTWARE ENGINEERING LAB

Laboratory Record

On

Clinic Management System

carried out as part of the course CSE CS3230(SE Lab) Submitted by

Aman Jain

209301581

&

Devansh Tiwari

209301586

VI-CSE-E

in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

In

Computer Science & Engineering

Under the supervision of

Dr. Rajat Goel



**MANIPAL UNIVERSITY
JAIPUR**

**Department of Computer Science & Engineering, School of
Computing and Computer Science,
Manipal University Jaipur,
16 April 2023**

INDEX s.no:	Assignment Name	Execution Date	Submission Date	Signature
01	Identifying the Requirements from Problem Statements.	14.FEB.2023	17.FEB.2023	
02	Do requirement analysis and develop Software Requirement Specification Sheet (SRS) for suggested system.	14.FEB.2023	19.FEB.2023	
03	Modelling UML Use Case Diagrams and Capturing Use CaseScenarios,	3.MAR.23	9.MAR.23	
04	Modelling (a) UML Class Diagrams and (b) Sequence diagrams	15.MAR.23	20.MAR.23	
05	Draw (a) E-R Model (b) <u>State- chart Diagram</u> (c) <u>Activity Diagram</u>	20.MAR.23	26.MAR.2023	
06	Estimation of Project Metrics (Effort Estimation)	27.MAR.23	3.APR.2023	
07	Estimation of Project Metrics (Cost Estimation)	27.MAR.23	3.APR.2023	
08	Test case development and implementation (of the project undertaken)	2.APR.23	6.APR.23	
09	Coding implementation. (Coding of Project)	2.APR.23	6.APR.23	

SOFTWARE ENGINEERING LAB(CS3230)
CLINIC MANAGEMENT SYSTEM

Under the supervision of Prof.
Rajat Goel

➤ **Problem Statement:**

The Clinical Management System is a windows-based software designed for registration and management of patient's records and easy access of the records. The system will be used to assist the register, doctors, lab technicians and chemists to store and manage patient records in a hospital or clinic for easier access and reference. All these activities are done routinely and would be cumbersome on the employees if done manually hence need of an efficient easy to use management software that will help ease the workload on employees in the clinic/hospital.

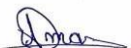
➤ **Description:**

- A Clinic Management System (CMS) is a software application that facilitates the management and organization of a healthcare clinic or hospital. The system is designed to streamline administrative tasks and enhance the efficiency of the medical practice, while also improving patient care and satisfaction.
- The system typically includes modules for managing patient appointments, patient records, electronic medical records (EMR), billing and payment processing, inventory management, and reporting. These modules are integrated and synchronized to ensure the accuracy and consistency of patient data across different functions and departments.
- With a CMS, healthcare providers can automate routine administrative tasks, such as appointment scheduling, prescription refill requests, and billing, freeing up more time for patient care. They can also access patient information in real-time, from any location, allowing them to make better-informed decisions and provide more personalized care.

➤ **Some functionalities may include:**

Here are some common functionalities for a Clinic Management System

- ✓ Patient Management: The system should allow healthcare providers to manage patient records, including personal information, medical history, appointment scheduling, and billing.
- ✓ Appointment Management: The system should allow the staff to schedule, modify or cancel appointments, and send reminders to patients via SMS, email or phone.
- ✓ Billing and Payment Management: The system should allow healthcare providers to generate bills for services, manage invoices, and accept online payments.
- ✓ Inventory Management: The system should allow the staff to manage the inventory of medicines, equipment, and other medical supplies and reorder them when the stock runs low.



Name: Aman Jain
Reg. No.: 209301581
B-tech CSE



Name: Devansh Tiwari
Reg. No.: 209301586
B-tech CSE

Software Requirements **Specification**

for

Clinic management system

Prepared by Aman Jain
Reg. no: 209301581

And

Devansh Tiwari
Reg. no: 209301586

Date Created: 19/02/2023

Table of Contents

Table of Contents	ii
Revision History	Error! Bookmark not defined.
1. Introduction.....	1
1.1 Purpose.....	1
1.2 Document Conventions.....	1
1.3 Intended Audience and Reading Suggestions.....	1
1.4 Product Scope	2
1.5 References	2
2. Overall Description.....	3
2.1 Product Perspective	3
2.2 Product Functions.....	3
2.3 User Classes and Characteristics.....	4
2.4 Operating Environment.....	4
2.5 Design and Implementation Constraints	4
2.6 User Documentation	5
2.7 Assumptions and Dependencies.....	5
3. External Interface Requirements	6
3.1 User Interfaces	6
3.2 Hardware Interfaces.....	7
3.3 Software Interfaces.....	7
3.4 Communications Interfaces	7
4. System Features.....	8
4.1 System Feature 1	Error! Bookmark not defined.
4.2 System Feature 2 (and so on).....	Error! Bookmark not defined.
5. Other Nonfunctional Requirements.....	8
5.1 Performance Requirements.....	8
5.2 Safety Requirements.....	10
5.3 Security Requirements.....	10
5.4 Software Quality Attributes.....	11
5.5 Business Rules	11
6. Other Requirements	11
Appendix A: Glossary.....	11
Appendix B: Analysis Models.....	12
Appendix C: To Be Determined List.....	16

1. Introduction

1.1 Purpose

The purpose of a Software Requirements Specification (SRS) document for a clinic management system is to clearly define and describe the functional and non-functional requirements of the software system that will be developed. The SRS document serves as a blueprint for the development team and stakeholders, outlining the scope of the project, the features and functions to be included, and the constraints and assumptions that must be taken into account.

Specifically, the SRS document for a clinic management system will:

1. Define the functional requirements of the software, such as the ability to schedule patient appointments, manage patient records, process billing and insurance claims, and generate reports.
2. Define the non-functional requirements of the software, such as its performance, reliability, security, usability, and compatibility with other systems.
3. Provide a clear and detailed description of the user interface and user experience (UI/UX) design, including wireframes, mock-ups, and prototypes.
4. Establish acceptance criteria and quality standards for the software, such as testing procedures, documentation requirements, and compliance with relevant regulations and standards.
5. Clarify any assumptions or constraints that may impact the development or use of the software, such as hardware and software dependencies, data privacy and security requirements, and user access permissions.

Overall, the SRS document for a clinic management system serves as a critical communication tool between the development team, stakeholders, and end-users, ensuring that everyone is aligned on the goals and requirements of the project, and that the resulting software meets the needs of its intended users.

1.2 Document Conventions

The following are the list of conventions and synonyms used in the document and project

Administrator:

- A log in ID representing the user with user administration privileges to software.

Client:

- Intend user of software. Doctors running clinics or Hospitals.
- Doctor: can see appointment and schedule appointment and can manage prescription and report of the patient.
- patient: can apply for appointment and can download report and prescription.

SQL:

- Used to perform operations on database.

1.3 Intended Audience and Reading Suggestions

The intended audience for the SRS (Software Requirements Specification) document for a clinic management system includes various stakeholders involved in the project, such as:

1. Project managers: Those responsible for overseeing the project and ensuring that it is completed within budget, on schedule, and meets the requirements of all stakeholders.
2. Developers: Those responsible for building and implementing the software system.
3. Testers: Those responsible for verifying that the software system meets the requirements specified in the SRS document.
4. Business analysts: Those responsible for analysing and documenting business requirements for the software system.
5. Healthcare professionals: Those who will be using the clinic management system to manage patient records, schedule appointments, and perform other clinical tasks.
6. Technical writers: Those responsible for creating user manuals, help guides, and other documentation related to the clinic management system.

Reading suggestions for the SRS document for a clinic management system would depend on the specific needs and interests of each stakeholder group. However, some general suggestions are:

1. Project managers should read the entire SRS document to ensure that the project meets business requirements, budget, and timeline.
2. Developers should read the functional and non-functional requirements sections to understand what needs to be built and the constraints involved.
3. Testers should read the acceptance criteria and quality standards sections to understand how to verify that the software system meets the requirements.
4. Business analysts should read the entire document to ensure that the business requirements are accurately captured.
5. Healthcare professionals should read the user interface and user experience (UI/UX) design section to ensure that the software system is easy to use and meets their needs.
6. Technical writers should read the entire document to understand how the software system works and what information needs to be documented.

Overall, the SRS document for a clinic management system is an important document that should be read and understood by all stakeholders involved in the project.

1.4 Product Scope

The product scope of a clinic management system software includes several key components, such as patient management, appointment scheduling, billing and payments, inventory management, and reporting and analytics. The software system should allow healthcare professionals to manage patient records, schedule appointments, generate invoices, manage inventory, and analyze data related to clinic operations. By providing these functionalities, the clinic management system can help healthcare organizations to streamline their operations, improve patient care, and optimize their revenue management.

1.5 References

- www.google.com
- www.wikipidea.com

- www.stackoverflow.com
- www.github.com

2. Overall Description

2.1 Product Perspective

The product perspective of a clinic management system software refers to how the software system fits into the larger context of the healthcare organization and the external environment. In other words, it describes the relationships between the software system and its stakeholders, including users, other software systems, and regulatory bodies.

From a user perspective, the clinic management system software is intended to be an all-in-one solution that helps healthcare professionals to manage patient records, appointments, billing, inventory, and other important aspects of clinic operations. The software system should be user-friendly and intuitive to use, with a clean and organized user interface that makes it easy for healthcare professionals to access and manage information.

Overall, the clinic management system software should be designed to meet the needs of healthcare organizations and their stakeholders, while also complying with technical standards and regulatory requirements. By providing an integrated solution for managing clinic operations, the software system can help healthcare organizations to improve patient care, streamline their workflows, and optimize their revenue management.

2.2 Product Functions

The product function of a clinic management system software refers to the specific capabilities and features that the software system provides to healthcare professionals to manage clinic operations. The following are some of the key functions of a clinic management system software:

1. **Patient management:** The software system should allow healthcare professionals to manage patient records, including personal information, medical history, and treatment plans. This includes the ability to create and update patient profiles, track patient visits and appointments, and view and share medical records.
2. **Appointment scheduling:** The software system should allow healthcare professionals to schedule and manage patient appointments, including sending reminders and notifications. This feature can help clinics to streamline their scheduling process and reduce missed appointments.
3. **Billing and payments:** The software system should allow healthcare professionals to generate invoices and manage payments. This includes the ability to create and send invoices, track payments, and manage insurance claims.
4. **Inventory management:** The software system should allow healthcare professionals to manage inventory of medicines, supplies, and other medical equipment. This feature can help clinics to keep track of their stock levels and ensure that they have the necessary supplies on hand.
5. **Reporting and analytics:** The software system should allow healthcare professionals to generate reports and analyse data related to patient visits, revenue, and other key performance indicators. This feature can help clinics to make data-driven decisions and improve their overall operations.

Overall, the product function of a clinic management system software is to provide healthcare professionals with an integrated solution for managing clinic operations. By providing these functionalities, the software system can help healthcare organizations to improve patient care, streamline their workflows, and optimize their revenue management.

2.3 User Classes and Characteristics

The user classes and characteristics of a clinic management system software refer to the different types of users who will interact with the software system and their unique characteristics. The following are some of the key user classes and their characteristics:

1. **Healthcare Professionals:** This includes doctors, nurses, medical assistants, and other healthcare staff who will use the software system to manage patient records, appointments, billing, and inventory. They should have a good understanding of medical terminology and procedures and be able to use the software system efficiently.
2. **Administrators:** This includes clinic managers, billing specialists, and other administrative staff who will use the software system to manage the clinic's operations. They should have a good understanding of the clinic's policies and procedures, as well as knowledge of accounting and billing practices.
3. **Patients:** This includes individuals who will use the software system to schedule appointments, view their medical records, and communicate with their healthcare provider. They may have varying levels of computer literacy and may require support to use the software system effectively.
4. **Technical Support:** This includes IT staff who will provide technical support and maintenance for the software system. They should have a good understanding of the software's technical specifications and be able to troubleshoot and resolve technical issues.

Overall, the user classes and characteristics of a clinic management system software should be considered when designing the software system's user interface and functionality. By understanding the needs and characteristics of different user classes, the software system can be designed to be intuitive and easy to use, while also providing the necessary functionalities for managing clinic operations.

2.4 Operating Environment

- *Windows*
- *Linux (Ubuntu 16.04)*
- *XAMPP*

2.5 Design and Implementation Constraints

Design and implementation constraints are limitations that may affect the design, development, and implementation of a clinic management system software. Some common design and implementation constraints for a clinic management system software include:

1. **Budget constraints:** The project may have limited financial resources, which could affect the ability to use advanced hardware and software technologies, or to hire a large development team.

2. **Time constraints:** The project may have strict deadlines for completion, which could affect the amount of time available for design, development, and testing.
3. **Resource constraints:** The project may have limited staff resources, which could affect the ability to allocate personnel to the project, or to obtain specialized skills required for the project.
4. **Technical constraints:** The project may have technical constraints, such as compatibility with existing systems, or the need to integrate with other software applications.
5. **Regulatory constraints:** The project may have regulatory constraints, such as compliance with government regulations regarding data privacy and security.

Design and implementation constraints should be considered during the planning and design phase of the project. The project team should identify potential constraints and develop strategies to mitigate their impact on the project. This may involve prioritizing features and functionalities, optimizing resource allocation, or seeking alternative solutions. By understanding and addressing design and implementation constraints, the clinic management system software can be developed and implemented successfully.

2.6 User Documentation

User documentation for a clinic management system software provides instructions and guidance for users to understand and effectively use the software. User documentation typically includes the following components:

1. **User Manual:** The user manual provides a comprehensive guide on how to use the clinic management system software. It includes step-by-step instructions for various tasks, such as patient registration, appointment scheduling, and medical records management.
2. **Quick Reference Guides:** Quick reference guides provide a condensed version of the user manual for users who need a quick reference for specific tasks.
3. **Online Help System:** An online help system provides context-sensitive help for users. This means that users can access help information that is relevant to the task they are currently performing within the software.
4. **Tutorials:** Tutorials provide interactive instruction for users to learn how to use the clinic management system software. Tutorials may be in the form of videos, interactive simulations, or step-by-step instructions.
5. **Frequently Asked Questions (FAQs):** FAQs provide answers to common questions that users may have about the clinic management system software.

User documentation should be clear, concise, and easy to understand. It should be written from the user's perspective and use language that is familiar to the user. User documentation should also be regularly updated to reflect changes to the software, new features, or any bug fixes. By providing effective user documentation, users can quickly learn and effectively use the clinic management system software.

2.7 Assumptions and Dependencies

Assumptions and dependencies are factors that can impact the development and implementation of a clinic management system software. Some common assumptions and dependencies for a clinic management system software include:

1. Availability of hardware and software resources: The clinic management system software may require specific hardware and software resources to function properly. Assumptions regarding the availability and compatibility of these resources may impact the design and development of the software.
2. Availability of staff resources: The successful implementation of the clinic management system software may require the allocation of staff resources, such as system administrators, database administrators, and technical support personnel.
3. Access to data sources: The clinic management system software may depend on the availability and accuracy of data from external sources, such as laboratory information systems, electronic health records, and billing systems.
4. Compliance with regulatory requirements: The clinic management system software may need to comply with government regulations regarding data privacy and security, as well as industry standards for interoperability and data exchange.
5. Availability of training and support resources: Assumptions regarding the availability and quality of training and support resources may impact the ability of users to effectively use the clinic management system software.

Assumptions and dependencies should be identified and documented during the planning and design phase of the project. By understanding and addressing assumptions and dependencies, the clinic management system software can be developed and implemented successfully.

3. External Interface Requirements

3.1 User Interfaces

User interfaces of a clinic management system software refer to the graphical interfaces that enable users to interact with the software. The user interfaces should be intuitive, easy to use, and provide access to the functionalities and features of the software. Some common components of the user interface for a clinic management system software include:

1. Dashboard: A dashboard provides an overview of the system, including important metrics such as patient waiting times, appointment schedules, and staff availability.
2. Navigation: Navigation menus and buttons provide access to the various features and functionalities of the software, such as patient registration, medical records management, and appointment scheduling.
3. Forms: Forms are used to input and display information, such as patient demographic information, medical histories, and billing information.
4. Alerts and notifications: Alerts and notifications notify users of important events, such as patient arrivals, appointment cancellations, or medication reminders.
5. Reports: Reports provide data and statistics on the performance of the clinic, including patient satisfaction, revenue, and resource utilization.

User interfaces should be designed with the end user in mind, and should be tested and refined through user feedback. The design of the user interface should take into consideration the needs and preferences of the different user classes and characteristics, such as front office staff, clinical staff, and patients. By providing an intuitive and easy-to-use user interface, users can effectively use the clinic management system software to improve patient care and clinic efficiency.

3.2 Hardware Interfaces

Hardware interfaces of a clinic management system software refer to the physical devices that connect to the software, such as computers, printers, and scanners. The software should be compatible with a wide range of hardware devices and should be able to communicate with them effectively. The hardware interfaces should be documented, including details on hardware requirements and compatibility, to ensure that the software is compatible with the hardware devices used in the clinic. By providing robust hardware interfaces, the clinic management system software can seamlessly integrate with existing hardware infrastructure, enabling efficient and effective patient care.

3.3 Software Interfaces

Software interfaces of a clinic management system software refer to the other software applications and systems that the clinic management system software needs to interact with in order to function effectively. These interfaces are critical to ensuring that the clinic management system software can communicate with other systems and applications, such as electronic health records systems, laboratory information systems, and billing systems.

The software interfaces should be designed to facilitate seamless data exchange and communication between different systems, enabling the clinic management system software to access and utilize the data stored in these systems. The interfaces should be well-documented, including details on data formats, protocols, and APIs, to ensure that the software can effectively communicate with other systems.

In addition, the software interfaces should be designed with data security and privacy in mind, ensuring that sensitive patient information is transmitted securely and that data is properly encrypted during transmission. By providing robust software interfaces, the clinic management system software can effectively integrate with other healthcare systems, improving data accuracy, reducing errors, and enabling better patient care.

3.4 Communications Interfaces

External communication interface requirements for a clinic management system software refer to the requirements that govern how the software interacts with external parties, such as patients, healthcare providers, and insurance companies. These requirements include identifying the types of external parties the software needs to communicate with, specifying the communication protocols and data formats used for communication, and outlining any security and privacy considerations.

The external communication interface requirements should be well-documented, ensuring that the software can effectively communicate with external parties in a secure and efficient manner. By meeting the external communication interface requirements, the clinic management system software can enhance patient engagement, streamline healthcare operations, and improve healthcare outcomes.

4. System Features

System features of a clinic management system software refer to the capabilities and functionalities that the software offers to healthcare providers and patients. These features can include appointment scheduling, patient registration, electronic health record management, medical billing, and inventory management. The system features should be designed to meet the specific needs of healthcare providers and patients, providing a user-friendly interface that enables easy navigation and efficient workflows. The system features should also be well-documented, including detailed specifications and instructions for use. By providing robust system features, the clinic management system software can streamline healthcare operations, improve patient outcomes, and enhance the overall quality of care.

4.1 Appointment Scheduling:

One of the key system features of a clinic management system software is appointment scheduling. This feature enables healthcare providers to efficiently manage their schedules and book appointments with patients. The appointment scheduling feature should be designed to allow for easy navigation and should offer a user-friendly interface for both healthcare providers and patients. The feature should also allow healthcare providers to view their schedules and make changes in real-time, as well as offer automatic appointment reminders for patients. By providing a robust appointment scheduling system feature, the clinic management system software can help healthcare providers streamline their operations, reduce scheduling errors, and enhance the patient experience.

4.2 Electronic health record (EHR) management:

Another key system feature of a clinic management system software is electronic health record (EHR) management. This feature enables healthcare providers to easily manage and access patient health records in a secure and efficient manner. The EHR management feature should be designed to allow for the easy input and retrieval of patient data, including medical histories, diagnoses, and treatment plans. The feature should also allow healthcare providers to easily generate and share reports with other healthcare professionals. By providing a robust EHR management system feature, the clinic management system software can help healthcare providers deliver more effective and personalized care, while also improving healthcare outcomes and reducing errors in record-keeping.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Performance requirements for a clinic management system software refer to the criteria that must be met in order for the software to perform optimally. These requirements can include factors such as response time, processing speed, and system availability. The performance requirements should be designed to ensure that the software can handle the expected workload and user traffic without slowing down or crashing. The system should also be able to handle large amounts of data without compromising on speed or accuracy. By meeting the performance requirements, the clinic management system software can improve the efficiency of healthcare operations, enhance the patient experience, and improve healthcare outcomes.

5.1.1 Entity Relationship Diagram:

The E-R Diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is then used to organize data as a relation, normalizing relation and finally obtaining a relation database.

- **ENTITIES:** Which specify distinct real-world items in an application.
- **PROPERTIES/ATTRIBUTES:** Which specify properties of an entity and relationships.

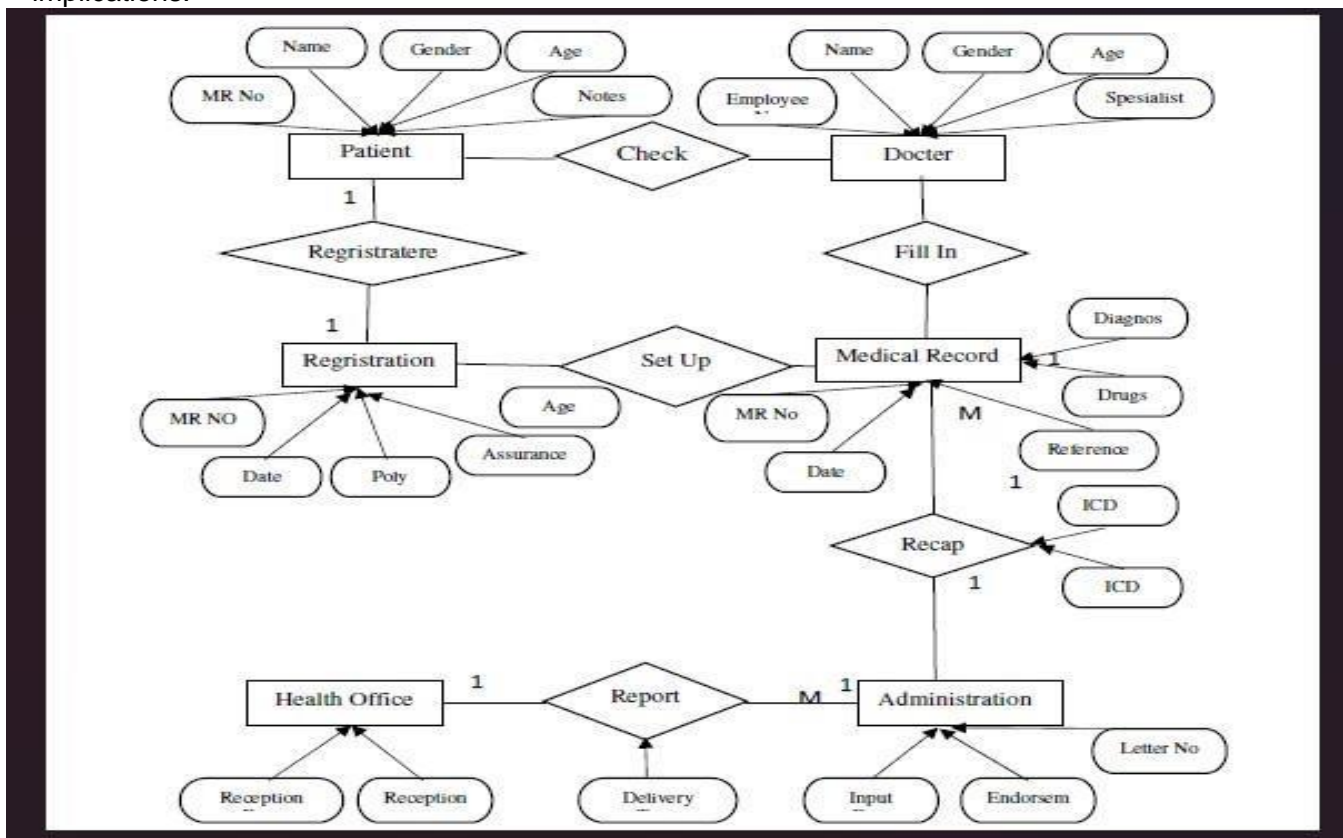
RELATIONSHIPS: Which connect entities and represent meaningful dependencies between them.

5.1.2 Normalization:

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

If a database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.

Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme. There are three different kinds of modifications of anomalies and formulated the first, second and third normal forms (3NF) is considered sufficient for most practical purposes. It should be considered only after a thorough analysis and complete understanding of its implications.



5.2 Safety Requirements

In the event of a catastrophic failure that results in significant damage to a large portion of the database, like a disk crash, the recovery method for the clinic management system software involves restoring a previously backed-up copy of the database that is stored in archival storage, usually on tape. This restoration process is followed by reconstructing a more recent version of the database by reapplying or redoing the committed transactions from the backed-up log that occurred until the time of the failure. By doing so, the recovery method ensures that the database returns to a consistent state, and all the committed transactions that occurred before the failure are recovered, thereby minimizing data loss and maintaining data integrity.

5.3 Security Requirements

Security requirements for clinic management software should include measures to ensure the confidentiality, integrity, and availability of patient and clinic data. This may involve implementing access controls to restrict user access to sensitive data, enforcing secure password policies, using encryption to protect data in transit and at rest, and implementing measures to prevent unauthorized modification or deletion of data. Additionally, the software should have a mechanism for logging and auditing user activity to detect and investigate any potential security breaches. Compliance with industry-standard security frameworks, such as HIPAA and GDPR, should also be considered. █

5.4 Software Quality Attributes

To ensure the effectiveness of the clinic management software, it is important to consider its maintainability, usability, and availability. The system administrator should take steps to keep the software up to date to ensure its continued functionality and optimal performance. The software should also be designed with usability in mind, with an intuitive interface that is easy to navigate and understand. Additionally, the software should be available for use at all times to promote reliability and minimize any potential downtime. By prioritizing these factors, the clinic management software can offer a high level of value and utility to its users.

5.5 Business Rules

- Patients must provide accurate personal and medical information before receiving treatment.
- Only authorized healthcare providers can access patient information.
- Payment for services must be made at the time of treatment.
- Prescriptions can only be issued by licensed healthcare providers.
- Patients have the right to access their own medical records.

6. Other Requirements

- The system should comply with relevant data protection laws and regulations.
- The software must support multiple languages for international users.
- The system should be scalable to accommodate growth in data and user base.
- The software must be compatible with commonly used operating systems and browsers.
- The system should provide regular backups and disaster recovery options.

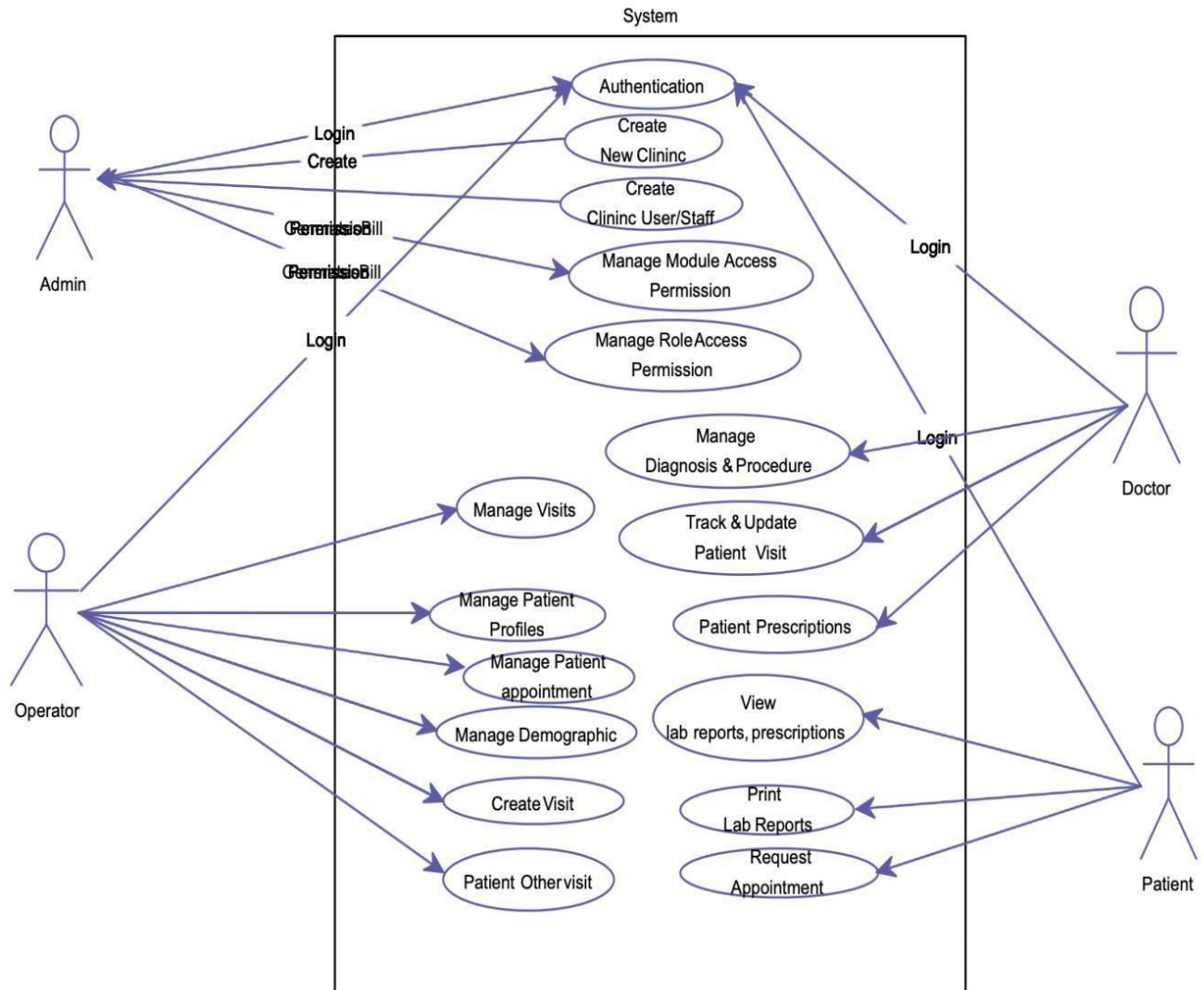
Appendix A: Glossary

some definitions for the glossary of clinic management system SRS are:

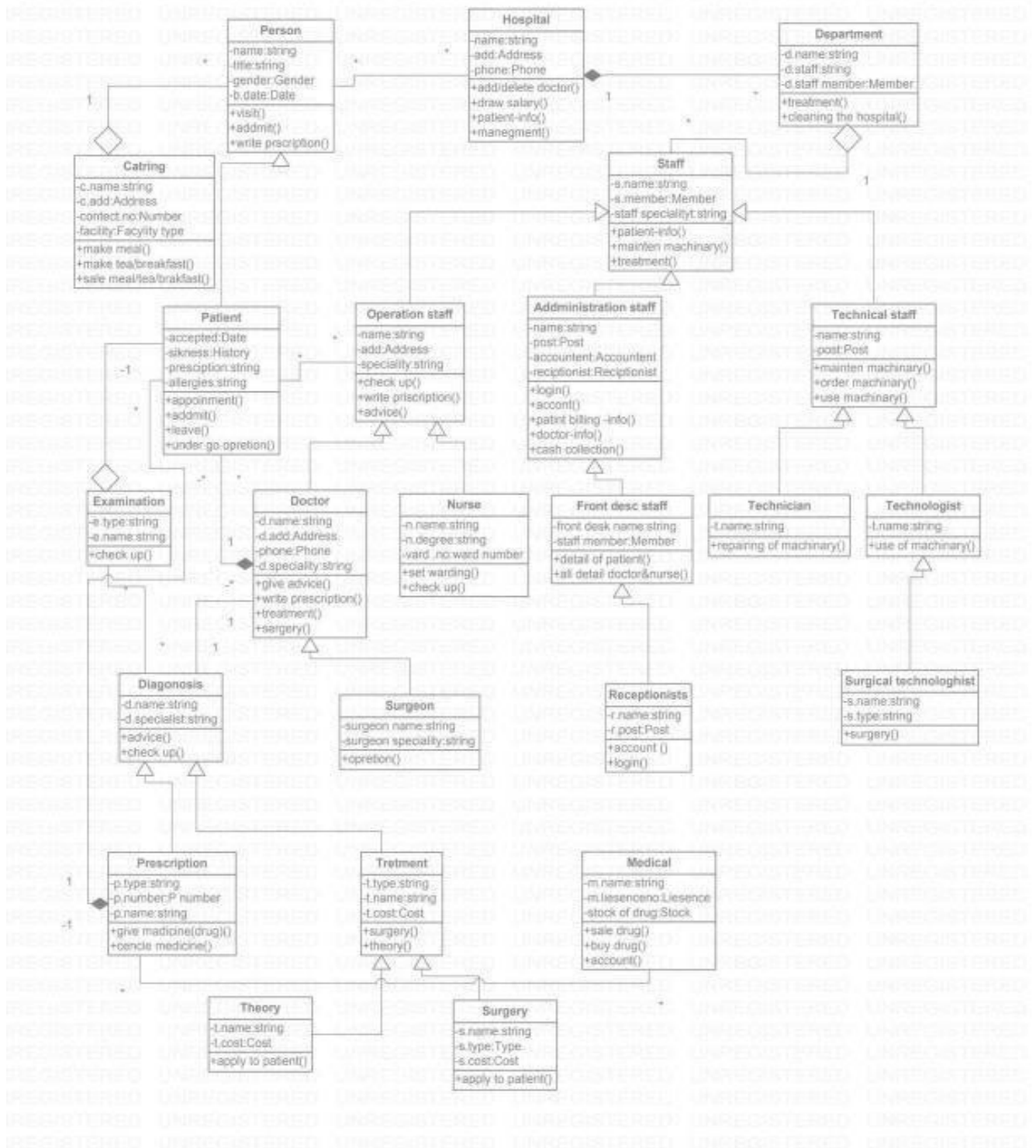
- Administrator: The person responsible for managing the clinic management system, including user accounts, security settings, and software updates.
- Electronic Health Record (EHR): A digital record of a patient's medical history, including diagnoses, treatments, medications, and other relevant information.
- HIPAA: The Health Insurance Portability and Accountability Act, which sets standards for the privacy and security of patients' medical information in the United States.
- Patient Portal: A secure online platform that allows patients to access their health records, communicate with their healthcare providers, and schedule appointments.
- Prescription Management: A feature of the clinic management system that allows healthcare providers to electronically prescribe medications for their patients and manage those prescriptions.
- Telemedicine: The use of technology, such as video conferencing, to provide healthcare services remotely.
- User: Any person who interacts with the clinic management system, including healthcare providers, administrative staff, and patients.

Appendix B: Analysis Models

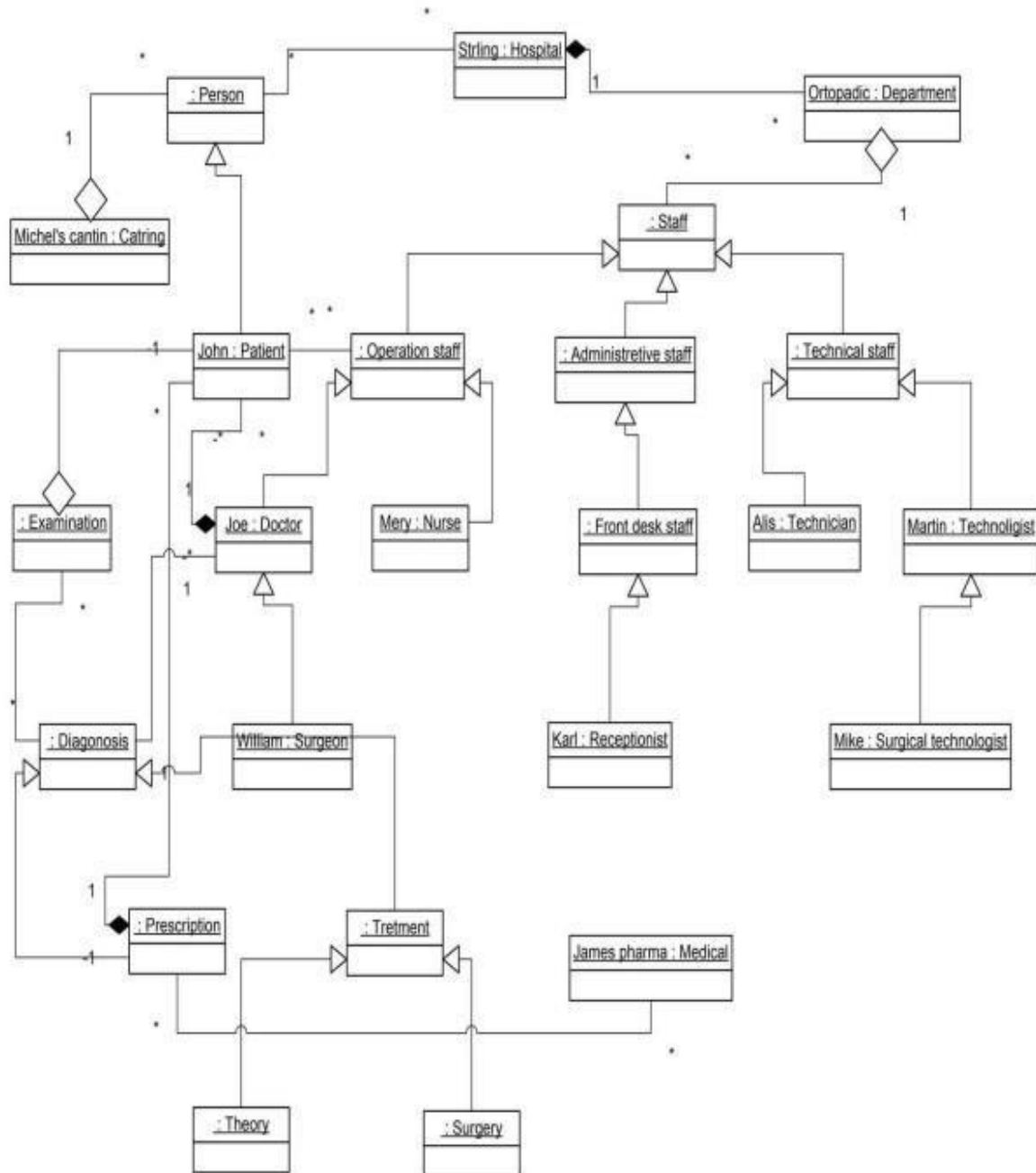
USE CASE DIAGRAM



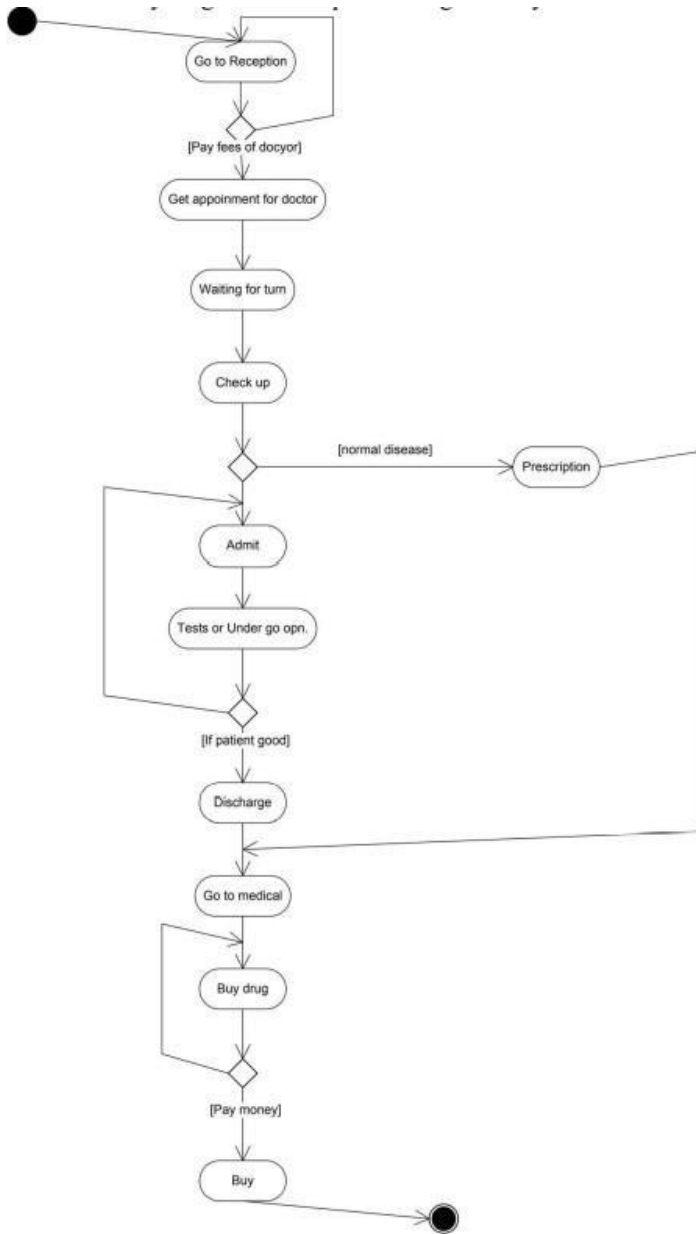
CLASS DIAGRAM:



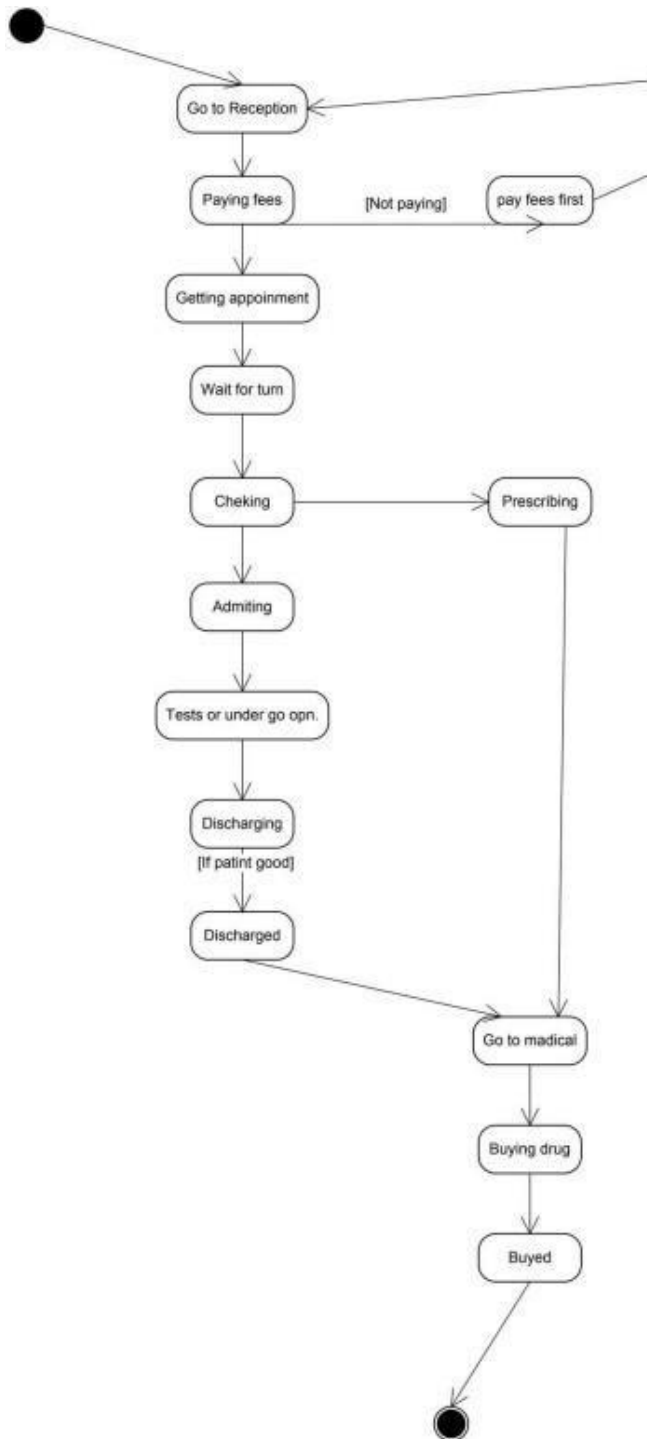
Object Diagram:



ACTIVITY DIAGRAM:

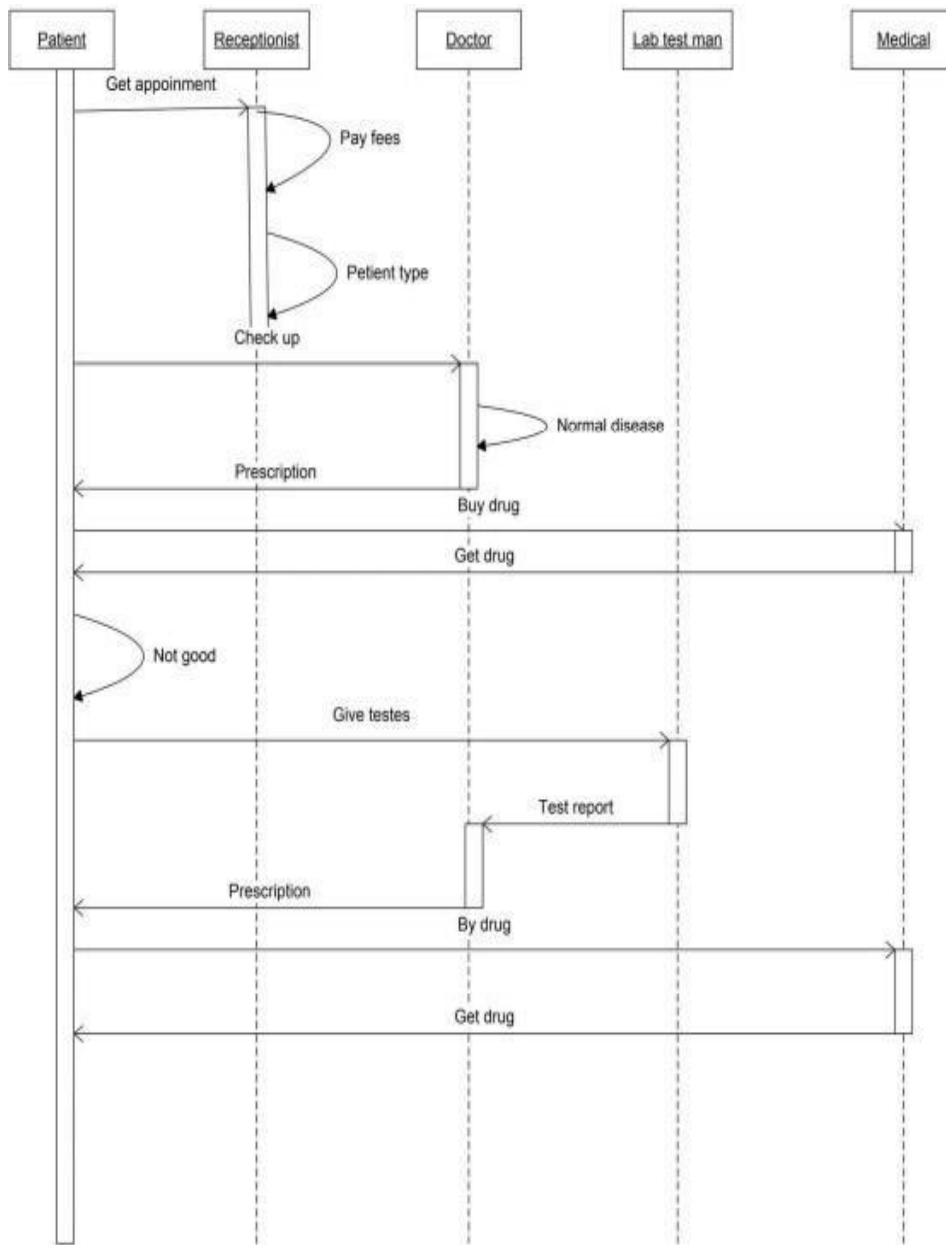


- **State Diagram:**

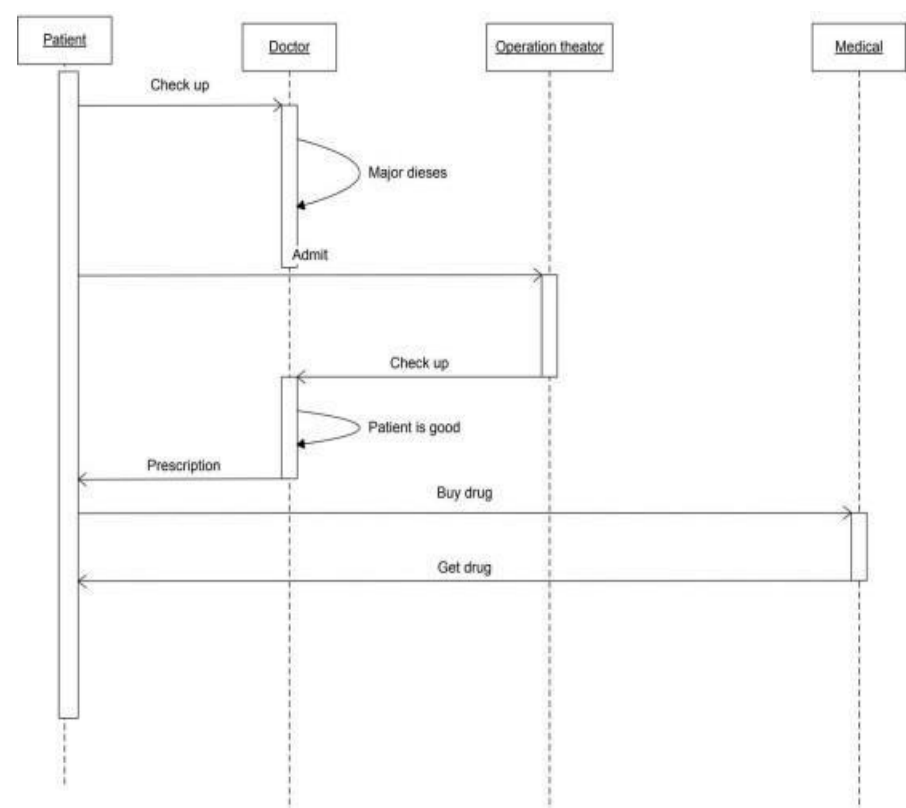


- **Sequence Diagram:**

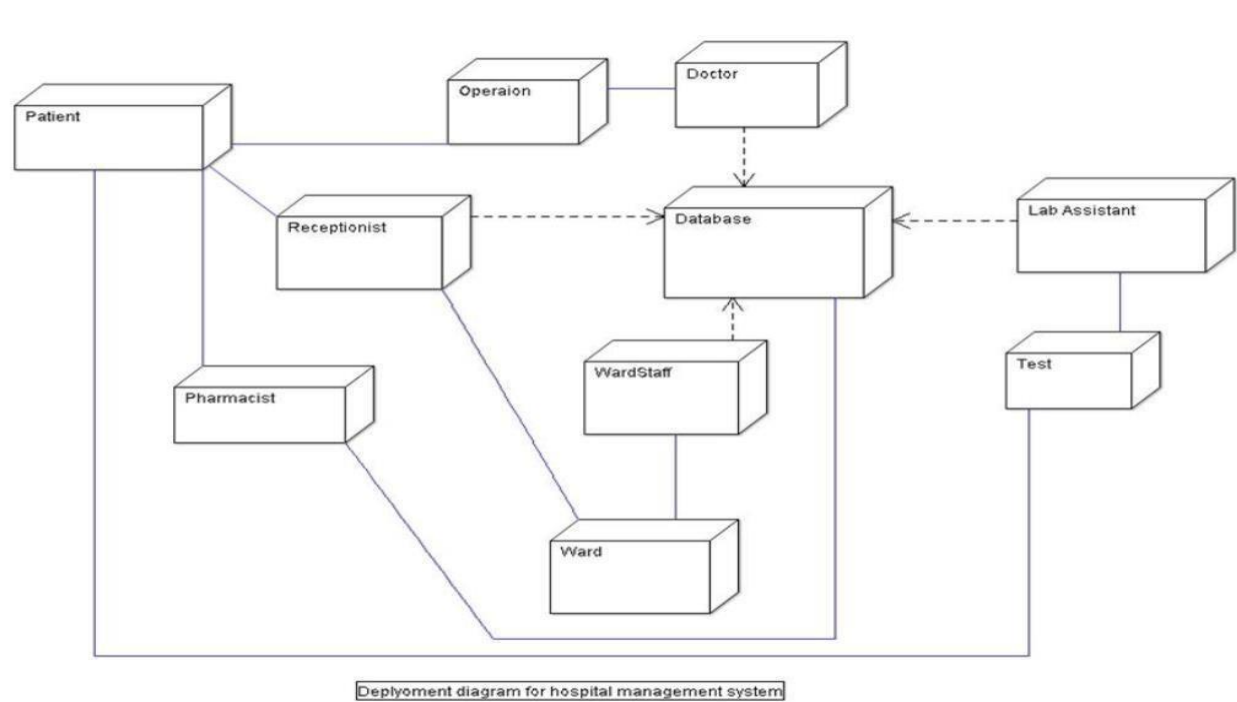
Sequence Diagram 1: Clinic management system



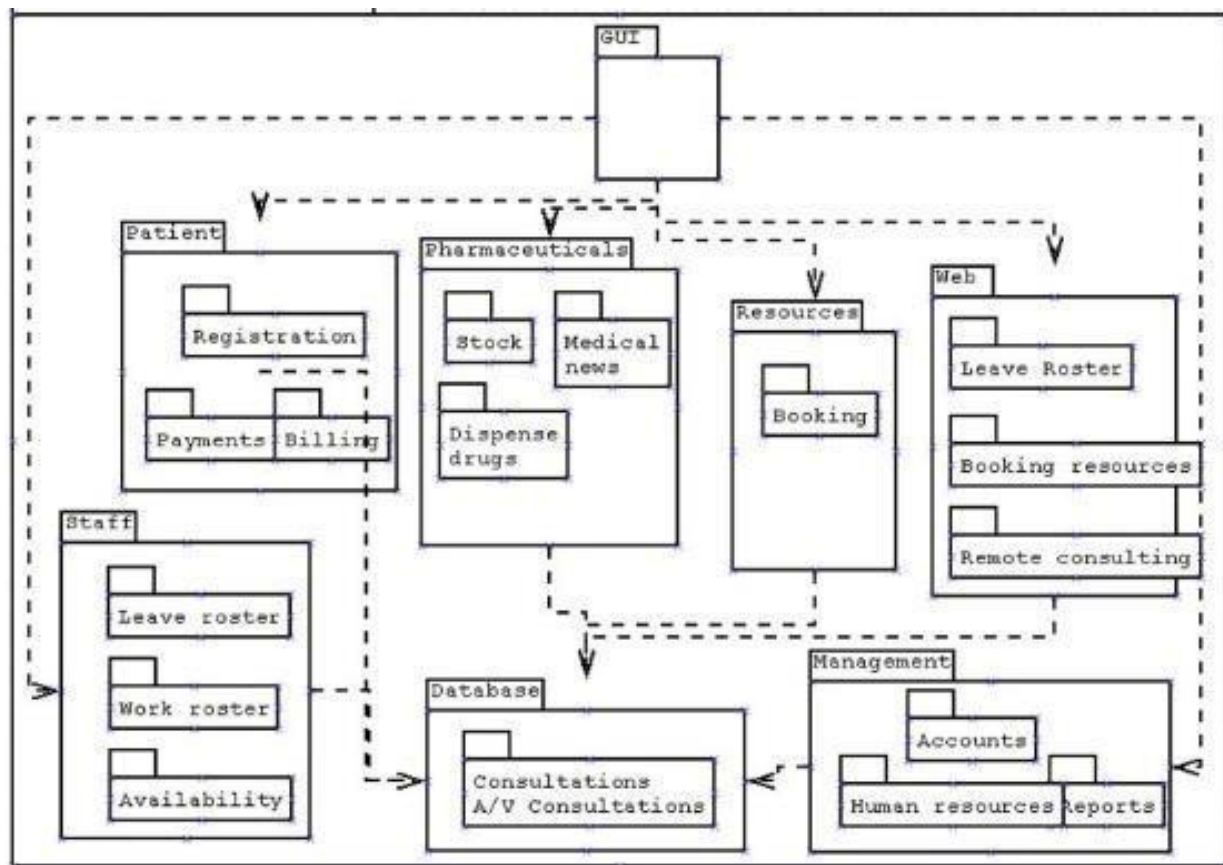
Sequence Diagram 2: Clinic management system



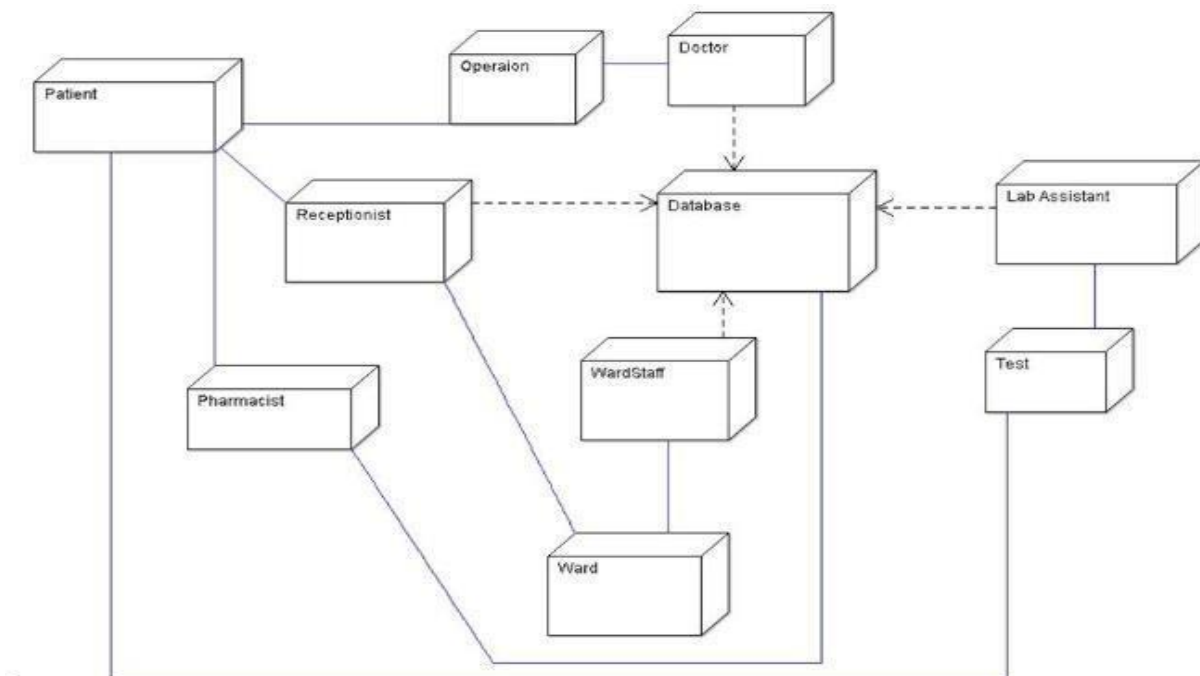
Component Diagram:



Package Diagram:



Deployment Diagram:



Appendix C: To Be Determined List

1. TBD - Security measures to be implemented for the protection of patient data.
2. TBD - Specific hardware requirements for the installation and operation of the system.
3. TBD - Detailed user documentation to be provided with the system.
4. TBD - Communication interfaces and protocols to be used for integration with external systems.
5. TBD - Specific performance metrics to be achieved by the system.
6. TBD - Methods for ensuring the safety and integrity of patient data in case of system failure.
7. TBD - Additional system features that may be added based on user feedback and requirements.
8. TBD - Business rules and workflows to be implemented in the system.
9. TBD - Additional constraints on the design and implementation of the system.
10. TBD - Dependencies on external systems or services that may affect the operation of the system.

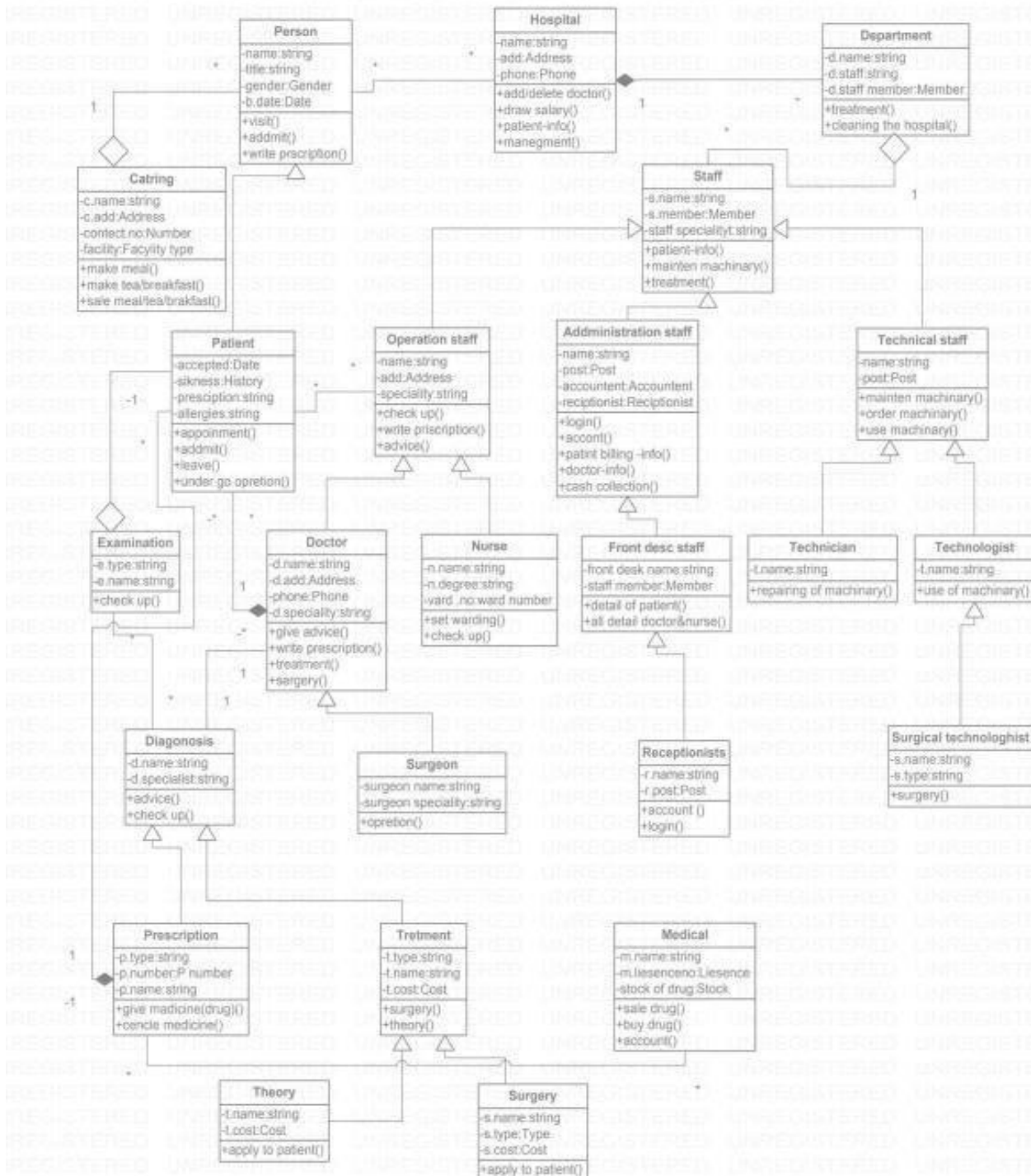
These TBD references must be resolved in order to finalize the requirements and specifications for the clinic management system.

Assignment

Modelling: a) Class Diagrams b) Sequence diagrams

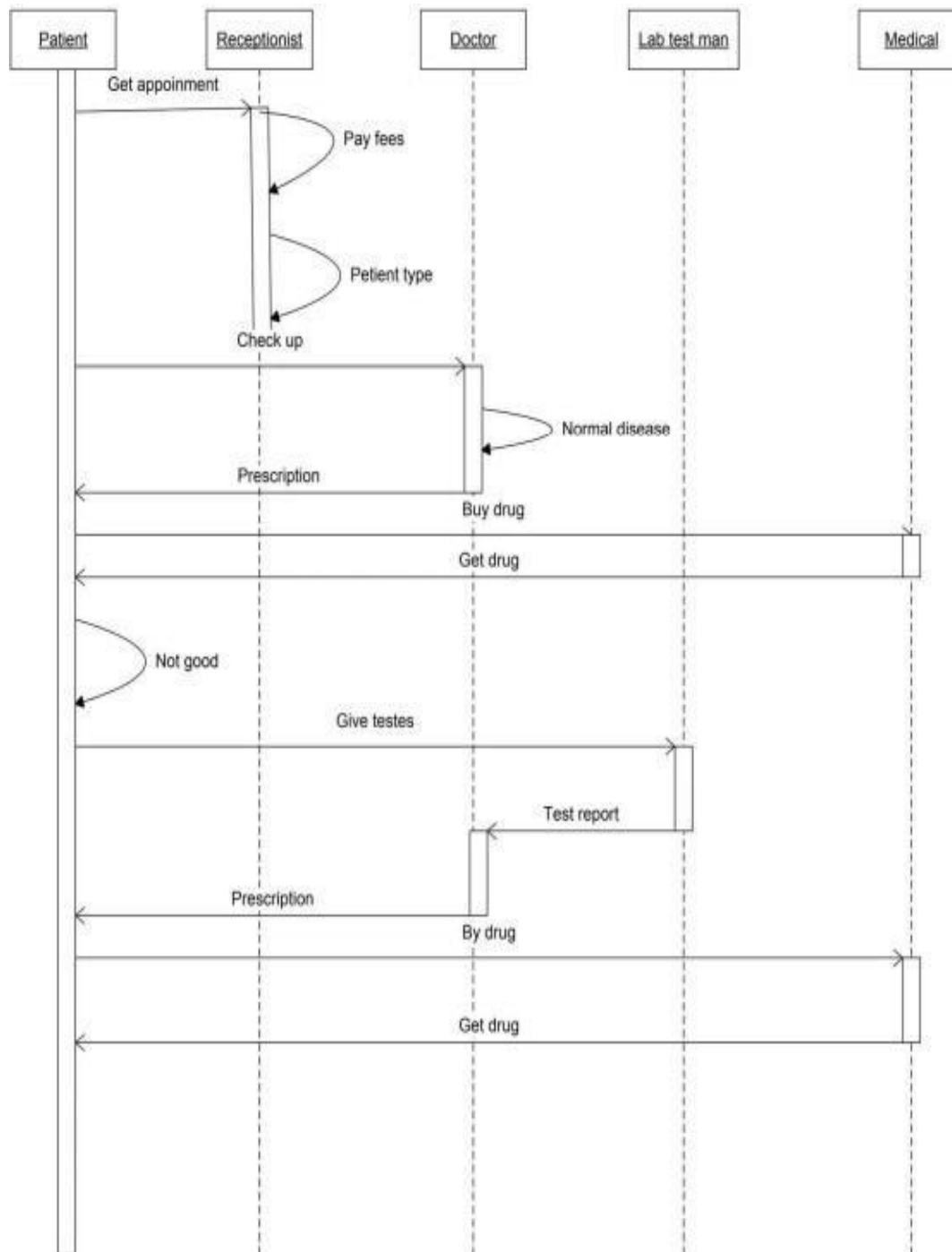
Clinic Management System

a) Class Diagrams

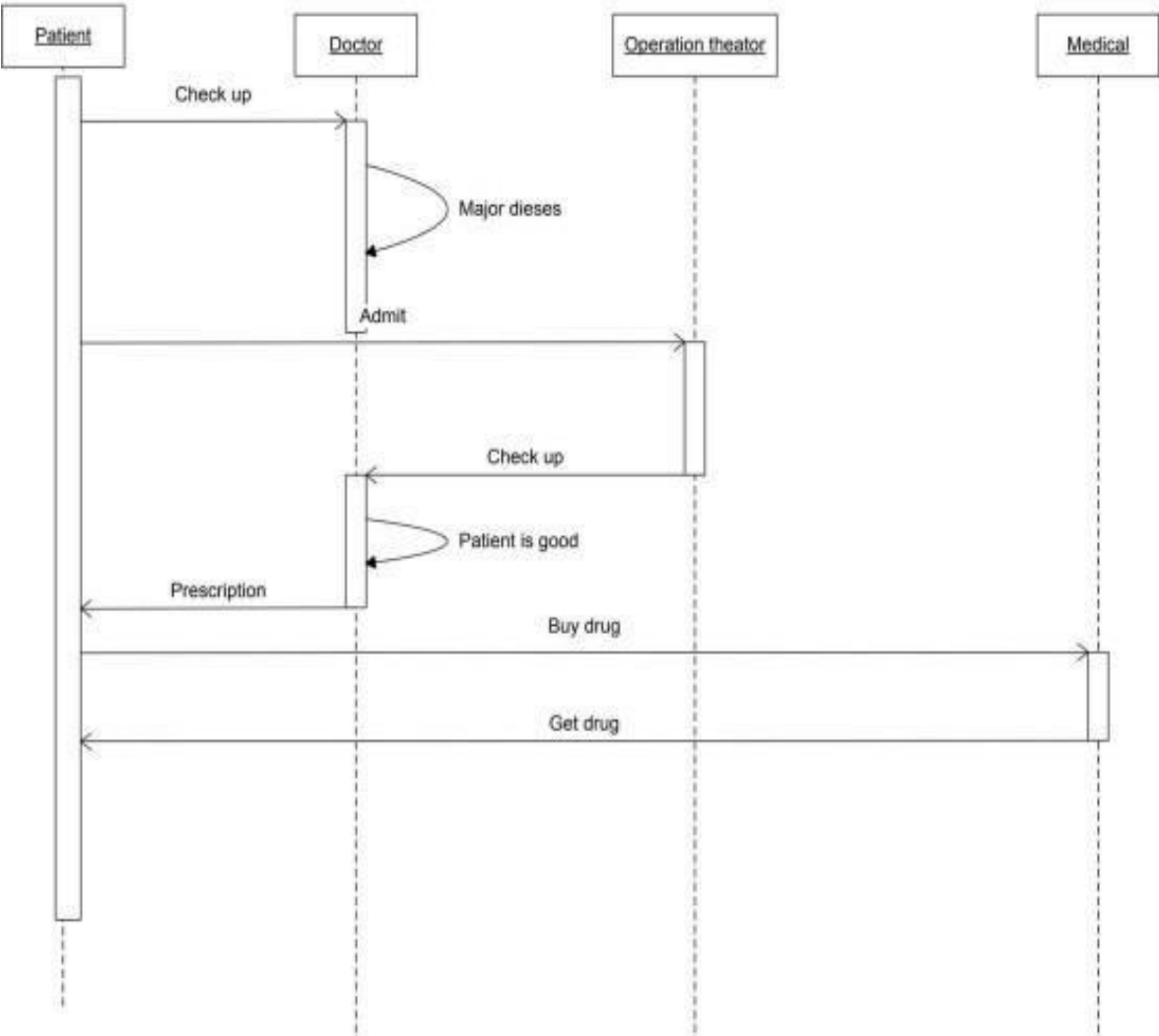


b) Sequence diagrams

Sequence Diagram 1: Clinic management system



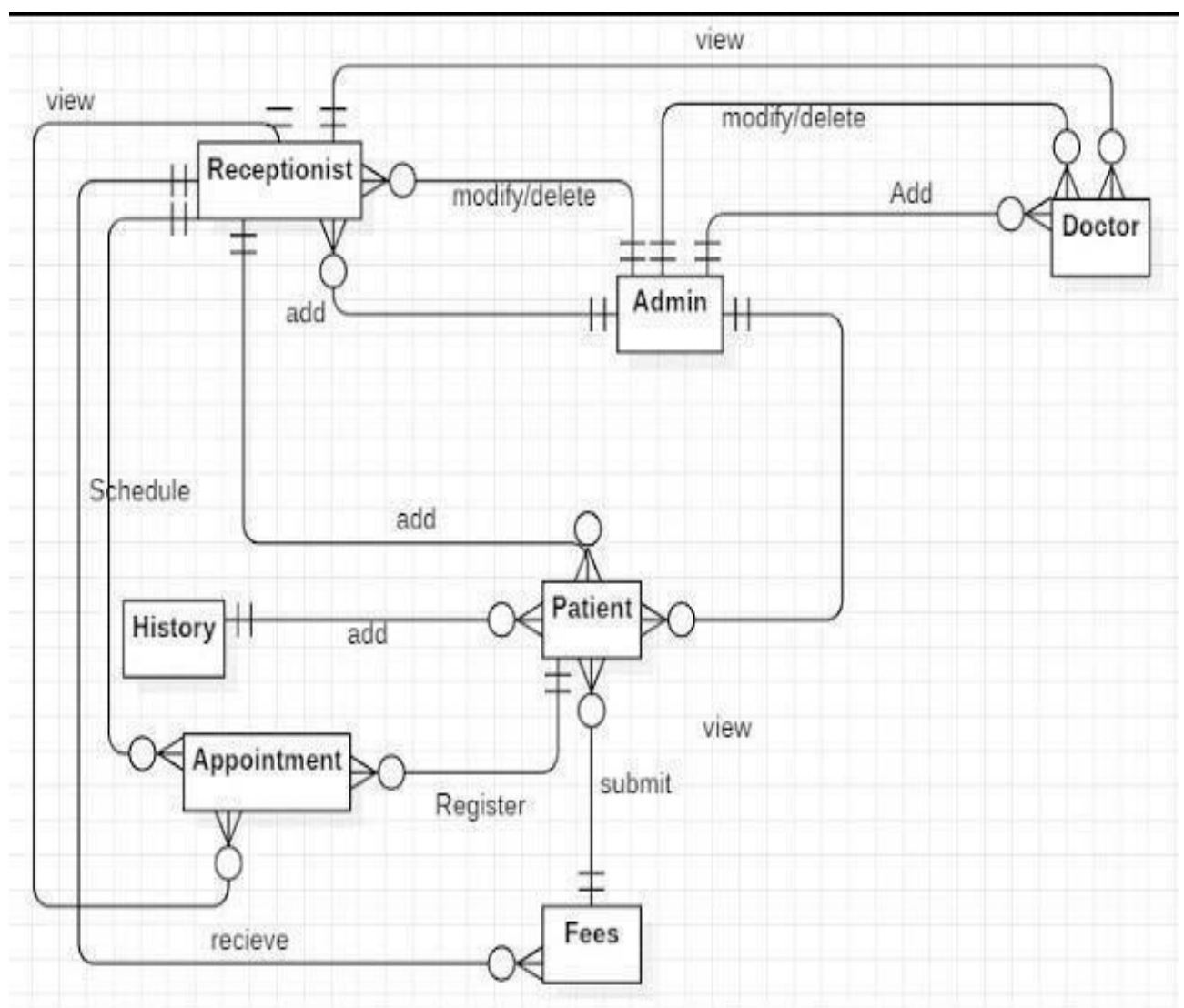
Sequence Diagram 2: Clinic management system



Assignment: E-R-Diagram

Clinic Management System

Note: Since, nowadays clinic contains mini lab to for instant result and also have facilities of bed and operation theatre area for small operations, so keeping those point in mind weare creating the clinic management system.

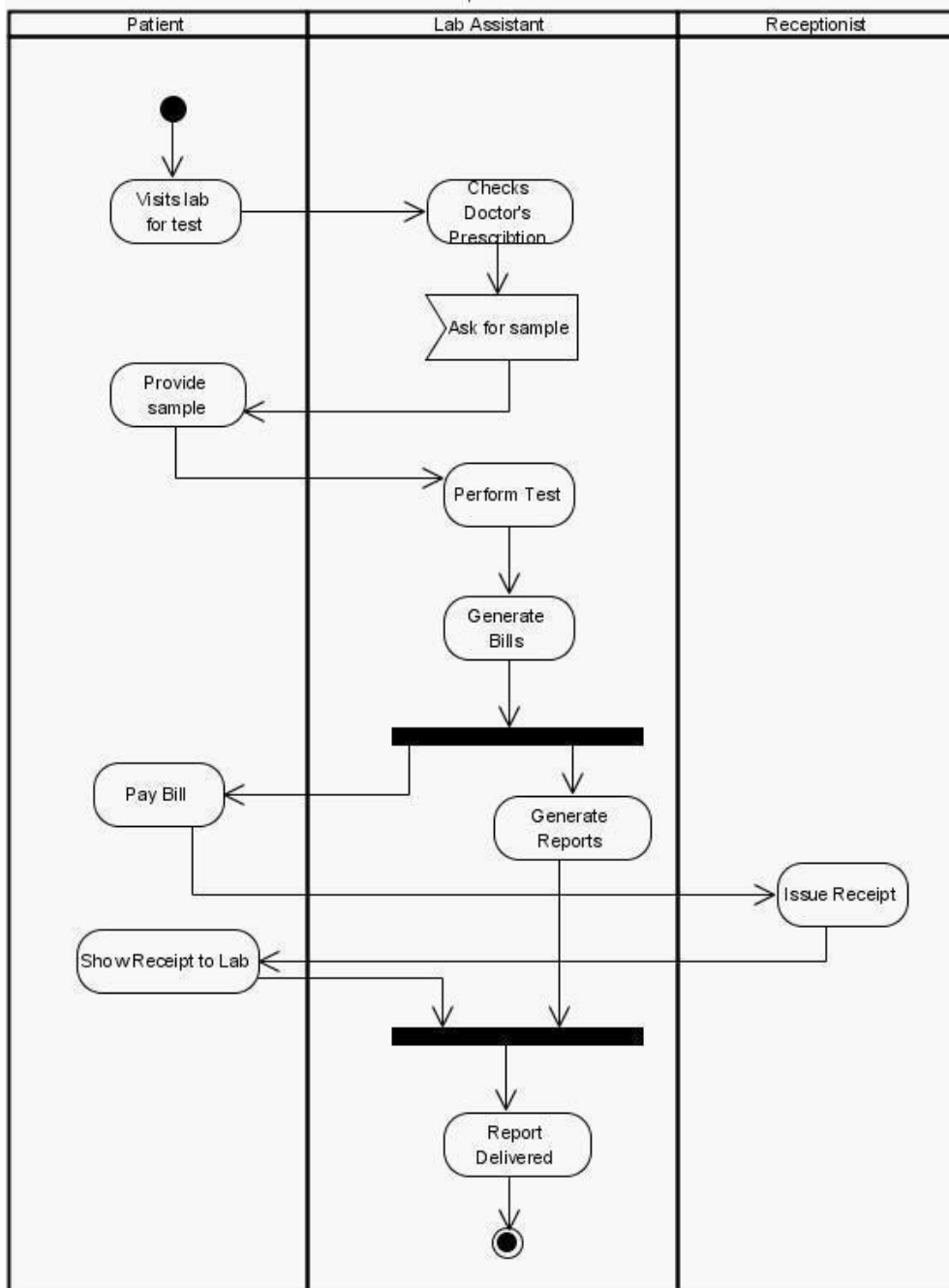


Assignment

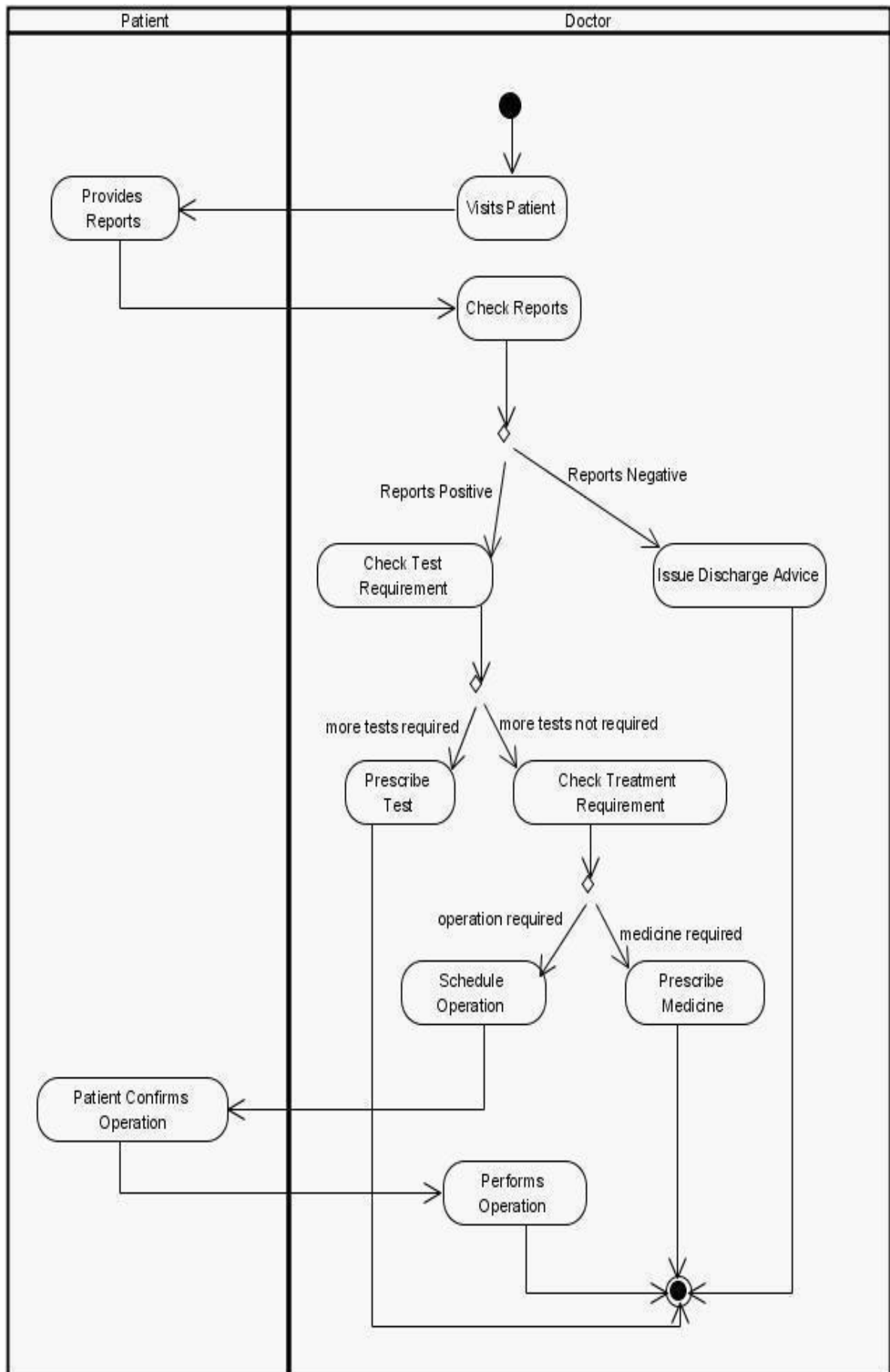
Activity Diagram for any two functionalities :Clinic Management System

Note: Since, nowadays clinic contains mini lab to for instant result and also have facilities of bed and operation theatre area for small operations, so keeping those point in mind we are creating the clinic management system.

A) Activity diagram between the patient and lab:



B) Activity diagram between the patient and doctor:



Assignment

State Transition Diagram: Clinic

Management System

Name: Aman Jain

Reg. No.: 209301581

Section: E

Batch: 2

Name: Devansh Tiwari

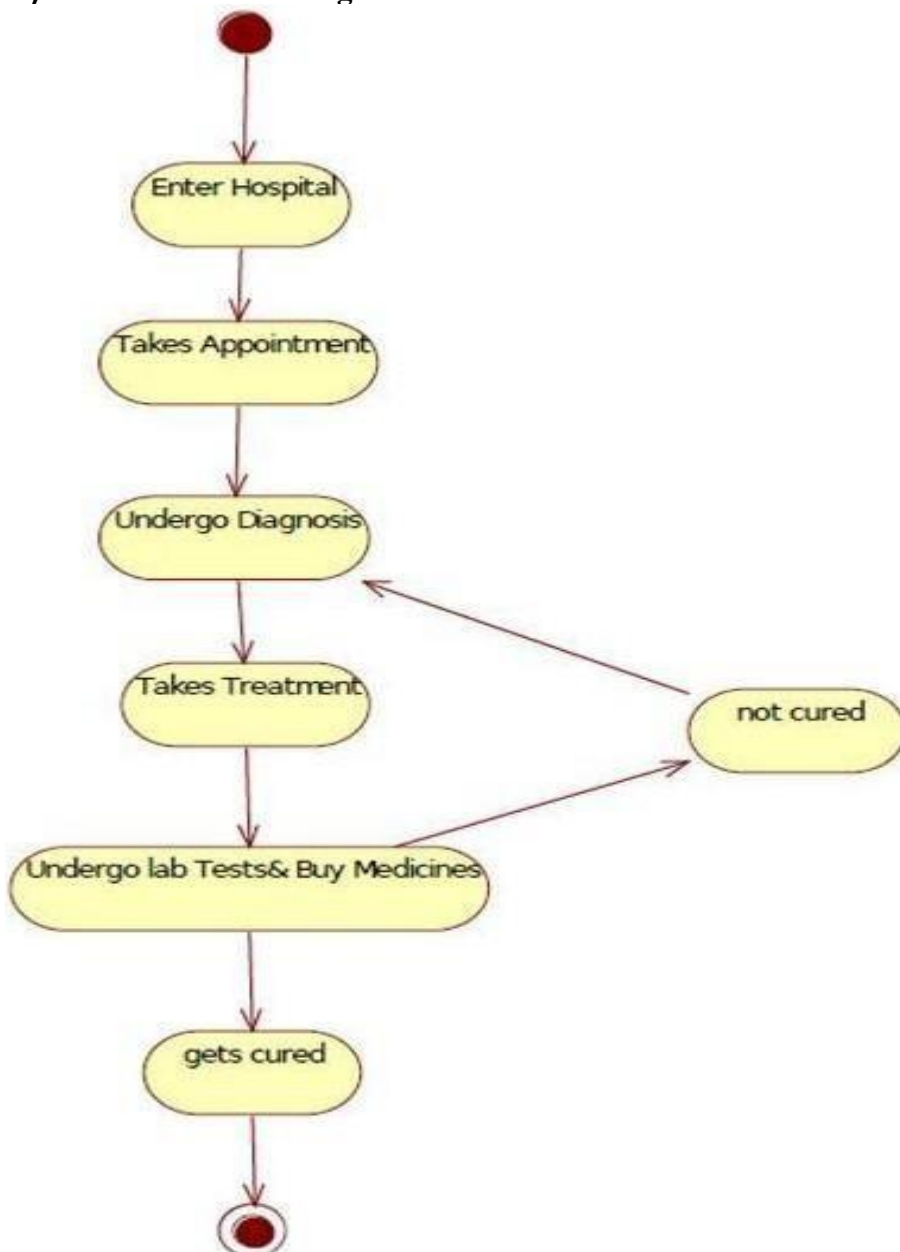
Reg. No.: 209301586

Section: E

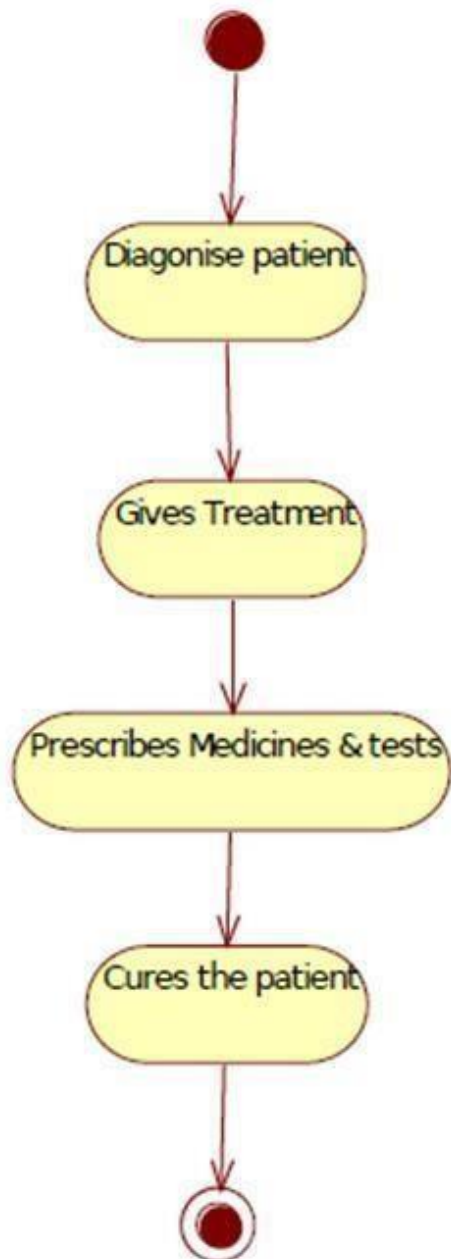
Batch: 2

Note: Since, nowadays clinic contains mini lab to for instant result and also have facilities of bed and operation theatre area for small operations, so keeping those point in mind weare creating the clinic management system.

A) State Transition Diagram :- Patient



A) State Transition Diagram :- Doct



Assignment

UML Diagram

Clinic Management System

Name: Aman Jain

Reg. No.: 209301581

Section: E

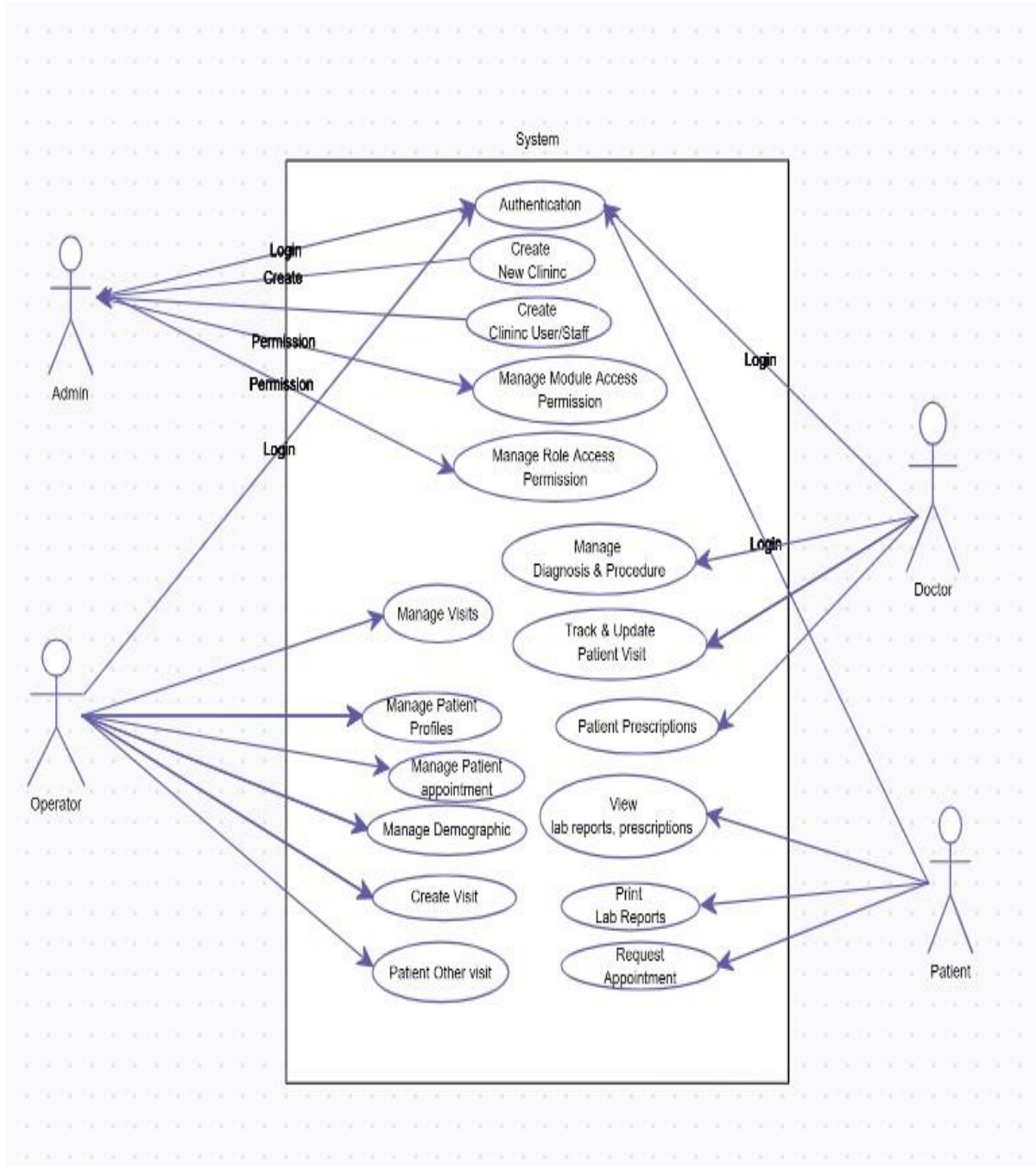
Batch: 2

Name: Devansh Tiwari

Reg. No: 209301586

Section: E

Batch: 2



Assignment

Effort

Estimation

Clinic Management System

Name: Aman Jain

Name: Devansh

Tiwari

Reg. No.: 209301581

Reg. No.: 209301586

Section: E

Section: E

Batch: 2

Batch: 2

Note: Since, nowadays clinic contains mini lab to for instant result and also have facilities of bed and operation theatre area for small operations, so keeping those point in mind weare creating the clinic management system.

- **EFFORT ESTIMATION OF THE PROJECT FOR Clinic MANAGEMENT SYSYTEM**

I have used the COCOMO model for estimating the cost of the system. It is regarded as a semidetached system. Since this project is somewhat small, COCOMO estimate might be inaccurate. COCOMO is designed for use on system larger than 2 KDL. This model estimates the total effort in term of person-month of technical project staff. It does not include the cost of the secretarial staff that might be needed. The basic steps in this model are:

- (1) Obtain an initial estimate of the development effort from the estimate of thousands of delivered lines of source code (KDL).
- (2) Determine a set of multiplying factor from different attribute of the project.
- (3) Adjust the effort estimate by multiplying the initial estimate with the entire multiplying factor.

The initial estimate is determined by an equation of the form used in the static, single-variable modes, using KDL as measure of size. To determine the initial effort E_i in person-months the equation used is of the type

$$E_i = a * (KDL)^b$$

There are 15 different attributes, called cost driver attributes that determine the multiplying factors. These factors depend on product, computer, personal, and technology. All 15 factors are multiplied together to get the effort adjustment factor (EAF). The final cost estimate, E , is obtained by multiplying the initial estimate by the EAF.

$$E = EAF * E_i$$

Cost estimation

The size estimates for these in lines of code are.

$$5634 = 5.634 \text{ KDL}$$

Category of project is semidetached so constraint of a & b follows

$$A=3.0 \text{ \& b =1.12}$$

$$\text{So, } E_i = 3.0(5.634)1.12$$

$$= 3.0(6.932)$$

$$= 20.789$$

Rating of multiplier for different cost drivers.

The effort adjustment factor (EAF) is

$$EAF = 1.4 * 1.08 * 1.15 * 1.15 * .86 * 1 * .86 * .95 * .95 * 1.1 * 1$$

$$= 1.46$$

The initial effort of the project is $E = E_i * EAF$

$$= 1.46 * 20.789$$

$$= 30.52 \text{ PM}$$

Cost driver	Rating	Values
Software reliability	Very high	1.40
Data base size	High	1.08
Product complexity	High	1.15
Computer turn around time	Very high	1.15
Analyst capability	High	0.86
Application experience	Nominal	1.00
Programmer capability	High	0.86
Programming language exp.	High	0.95
Modern prog. Practice	High	0.95
Use of software tools	Low	1.10
Development schedule	Nominal	1.00

Assignment

Cost Estimation

Clinic Management System

**Name: Aman Jain
Tiwari**

Reg. No.: 209301581

Section: E

Batch: 2

Name: Devansh

Reg. No.: 209301586

Section: E

Batch: 2

Note: Since, nowadays clinic contains mini lab to for instant result and also have facilities of bed and operation theatre area for small operations, so keeping those point in mind we are creating the clinic management system.

- **COST ESTIMATION OF THE PROJECT FOR Clinic MANAGEMENT SYSTEM**

Method 1: Cost Estimation using COCOMO:

Average Labour Cost is 8,000 rupee Per month.

I have used the COCOMO model for estimating the cost of the system. It is regarded as a semidetached system. Since this project is somewhat small, COCOMO estimate might be inaccurate. COCOMO is designed for use on system larger than 2 KLOC. This model estimates the total effort in term of person-month of technical project staff. It does not include the cost of the secretarial staff that might be needed. The basic steps in this model are:

- (1) Obtain an initial estimate of the development effort from the estimate of thousands of delivered lines of source code (KLOC).
- (2) Determine a set of multiplying factor from different attribute of the project.
- (3) Adjust the effort estimate by multiplying the initial estimate with the entire multiplying factor.

The initial estimate is determined by an equation of the form used in the static, single-variable modes, using KDL as measure of size. To determine the initial effort E_i in person-months the equation used is of the type

$$E_i = a * (KDL)^b$$

There are 15 different attributes, called cost driver attributes that determine the multiplying factors. These factors depend on product, computer, personal, and technology. All 15 factors are multiplied together to get the effort adjustment factor (EAF). The final cost estimate, E , is obtained by multiplying the initial estimate by the EAF.

$$E = EAF * E_i$$

Cost estimation

The size estimates for these in lines of code are.

$$5634 = 5.634 \text{ KDL}$$

Category of project is semidetached so constraint of a & b follows

$$A = 3.0 \text{ \& } b = 1.12$$

$$\begin{aligned}
 \text{So, } E_i &= 3.0(5.634)1.12 \\
 &= 3.0(6.932) \\
 &= 20.789
 \end{aligned}$$

Rating of multiplier for different cost drivers.

Cost driver	Rating	Values
Software reliability	Very high	1.40
Data base size	High	1.08
Product complexity	High	1.15
Computer turn around time	Very high	1.15
Analyst capability	High	0.86
Application experience	Nominal	1.00
Programmer capability	High	0.86
Programming language exp.	High	0.95
Modern prog. Practice	High	0.95
Use of software tools	Low	1.10
Development schedule	Nominal	1.00

The effort adjustment factor (EAF) is

$$\begin{aligned}
 \text{EAF} &= 1.4 * 1.08 * 1.15 * 1.15 * .86 * 1 * .86 * .95 * .95 * 1.1 * 1 \\
 &= 1.46
 \end{aligned}$$

The initial effort of the project is $E = E_i * \text{EAF}$

$$\begin{aligned}
 &= 1.46 * 20.789 \\
 &= 30.52 \text{ PM}
 \end{aligned}$$

Using the average labour cost of 8000 rupees per month, we can estimate the total cost of the project as:

$$\text{Cost} = \text{Effort} * \text{Labour Cost} = 30.57 * 8000 = 2,44,560 \text{ rupees}$$

Therefore, the estimated cost of the project is 2,44,560 rupees and the estimated effort required to develop the system is 30.57 person-months. It's important to note that these are only rough estimates and the actual effort and cost required may vary depending on a variety of factors, such as the complexity of the system, the skill level of the development team, and the development methodology used.

Method2: (LOC) BASED ESTIMATION

Historical data obtained from the Metrics indicates the following Organizational Averages: Average Productivity is 620 LOC / Pm (Lines of Code Per Month)

Average Labour Cost is 8,000 rupee Per month.

Cost for a Line of Code can be calculated as follows (COST / LOC)

$$\text{COST / LOC} = (8000 / 620) = \$13$$

Total Estimated Project Cost and Project Effort can be calculated as: follows-

Considering that the Total LOC (\sum LOC) for the System is 5634 LOC

$$\text{Total Estimated Project Cost} = (5634 * 13) = 73,242 \text{ Rupee}$$

$$\text{Total Estimated Project Effort} = (5634 / 620) = \sim 9 \text{ Man Months}$$

TEST CASES
Clinic Management System

Name: Aman Jain
Reg. No.: 209301581
Section: E
Batch: 2

Name: Devansh Tiwari
Reg. No.: 209301586
Sec: E
Batch: 2

Note: Since, nowadays clinic contains mini lab to for instant result and also have facilities of bed and operation theatre area for small operations, so keeping those point in mind we are creating the clinic management system.

S.No	Test Type	Input	Expected Output	Actual Output	Result
1	Patient Registration	Name: John, Age: 45, Gender: Male	Patient registration successful	Patient registration successful	Pass
2	Patient Registration	Name: Jane , Age: 30, Gender: Female	Patient registration successful	Patient registration successful	Pass
3	Patient Registration	Name: Tom Brown, Age: 20, Gender: Male	Patient registration successful	Patient registration successful	Pass
4	Appointment Scheduling	Patient: John Doe, Doctor: Dr. tiwari, Date: 2023-04-20 10:00 AM	Appointment scheduled successfully for John Doe with Dr. Smith	Appointment scheduled successfully for John Doe with Dr. Smith	Pass
5	Appointment Scheduling	Patient: Jane Smith, Doctor: Dr. jain, Date: 2023-04-21 02:00 PM	Appointment scheduled successfully for Jane Smith with Dr. Lee	Appointment scheduled successfully for Jane Smith with Dr. Lee	Pass
6	Appointment Scheduling	Patient: Tom Brown, Doctor: Dr. Patel, Date: 2023-04-22 11:00 AM	Appointment scheduled successfully for Tom Brown with Dr. Patel	Appointment scheduled successfully for Tom Brown with Dr. Patel	Pass

7	Medical Record Retrieval	Patient: John Doe	Medical records retrieved successfully for John Doe	Medical records retrieved successfully for John Doe	Pass
8	Medical Record Retrieval	Patient: Jane Smith	Medical records retrieved successfully for Jane Smith	Medical records retrieved successfully for Jane Smith	Pass
9	Medical Record Retrieval	Patient: Tom Brown	Medical records retrieved successfully for Tom Brown	Medical records retrieved successfully for Tom Brown	Pass

Coding implementation (Coding of Project)

- Tow screenshot Back-end code:

```
1 if (process.env.NODE_ENV !== "production") {
2   | require("dotenv").config();
3 }
4
5 const express = require("express");
6 const path = require("path");
7 const mongoose = require("mongoose");
8 const ejsMate = require("ejs-mate");
9 const session = require("express-session");
10 const flash = require("connect-flash");
11 const methodOverride = require("method-override");
12 const passport = require("passport");
13 const LocalStrategy = require("passport-local");
14 const User = require("../models/user");
15 const Doctor = require("../models/doctor");
16 const Secretary = require("../models/secretary");
17 const mongoSanitize = require("express-mongo-sanitize");
18 const helmet = require("helmet");
19 const userRoutes = require("../routes/users");
20 const doctorRoutes = require("../routes/doctors");
21 const MongoDBStore = require("connect-mongo")(session);
22
23 // 'mongodb://localhost:27017/hospital'
24 const dbUrl = "mongodb://localhost:27017/hospital";
25 mongoose.connect(dbUrl, {
26   useNewUrlParser: true,
27   useCreateIndex: true,
28   useUnifiedTopology: true,
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** SQL CONSOLE

node

New version of nodemon available!
Current Version: 2.0.19
Latest Version: 2.0.22

Serving on port 3000
Database connected

```
app.use("/users", userRoutes);
app.use("/doctors", doctorRoutes);

app.get("/", async (req, res) => {
  | const doctors = await Doctor.find({});
  | res.render("home", { doctors });
});


app.use((err, req, res, next) => {
  | const { statusCode = 500 } = err;
  | if (!err.message) err.message = "Oh No, Something Went Wrong!";
  | res.status(statusCode).render("error", { err });
});

const port = process.env.PORT || 3000;
app.listen(port, () => {
  | console.log(`Serving on port ${port}`);
});
```

- **Front-End View:**

CinicManagementSystem

[Home](#)
[About](#)
[Facility](#)
[Review](#)
[Contact](#)
[Post](#)
[Login](#)
[Register](#)



stay safe, stay healthy.
caring for you.

[Learn More](#)

CinicManagementSystem

[Home](#)
[About](#)
[Facility](#)
[Review](#)
[Contact](#)
[Post](#)
[Login](#)
[Register](#)

make an appointment

choose a section

choose a doctor

dd-mm-yyyy

--:--

🕒

message (optional)

Make Appointment

CinicManagementSystem

[Home](#)
[About](#)
[Facility](#)
[Review](#)
[Contact](#)
[Post](#)
[Login](#)
[Register](#)

Login

Email

Password

Login

Not a member? [Sign up](#)

23°C Clear

Search

ENG IN

02:33 18-04-2023